Access 2000 professionell

Gerhard Brosius

Access 2000 professionell

Datenbank- Management
mit Office 2000

ADDISON-WESLEY

An imprint of Pearson Education

München • Reading, Massachusetts • Menlo Park, California
New York • Harlow, England • Don Mills, Ontario
Sydney • Mexico City • Madrid • Amsterdam

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Access 2000 professionell : Datenbank- Management mit Office 2000 / Gerhard Brosius. –
München; Reading, Mass. [u.a.] : Addison-Wesley-Longman, 1999
ISBN 3-8273-1541-7

© 1999 by Addison Wesley Longman Verlag,
ein Imprint der Pearson Education Deutschland GmbH
Martin-Kollar-Straße 10–12, D-81829 München/Germany
Alle Rechte vorbehalten

10 9 8 7 6 5 4 3 2 1
02 01 00 99
ISBN 3-8273-1541-7

Einbandgestaltung: Michael Learo, Köln,
unter Verwendung von Bildmaterial aus dem Film »Safety Last« von Fred Newmeyer und Sam Taylor (1923) mit freundlicher Genehmigung der Stiftung Deutsche Kinemathek, Berlin.
Lektorat: Rudolf Krahm, Bonn
Herstellung: Elisabeth Egger
Satz: Reemers EDV-Satz, Krefeld, gesetzt aus der Sabon 9,5/12 pt mit FrameMaker.
Druck: Bercker Graphischer Betrieb, Kevelaer

Die Informationen in diesem Produkt werden ohne Rücksicht auf eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.
Bei der Zusammenstellung von Abbildungen und Texten wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen, die in diesem Buch erwähnt werden, sind gleichzeitig eingetragene Warenzeichen oder sollten als solche betrachtet werden.

Umwelthinweis:
Dieses Produkt wurde auf chlorfrei gebleichtem Papier gedruckt. Die Einschrumpffolie – zum Schutz vor Verschmutzung – ist aus umweltverträglichem und recyclingfähigem PE-Material.

Inhaltsübersicht

Kapitel 1	Einleitung und Überblick	15
Kapitel 2	Für alte Bekannte: Neuerungen in Access 2000	19
Kapitel 3	Access installieren	23
Kapitel 4	Einführung in Access	29
Kapitel 5	Einführungsbeispiel: Eine einfache relationale Datenbank erstellen	63
Kapitel 6	Daten bearbeiten	123
Kapitel 7	Eine Datenbank konzipieren	167
Kapitel 8	Tabellen: Grundlagen	177
Kapitel 9	Tabellen importieren, exportieren und verknüpfen	241
Kapitel 10	Abfragen: Grundlagen	261
Kapitel 11	Auswahlabfragen	277
Kapitel 12	Komplexe Abfragen	319
Kapitel 13	Aktionsabfragen	353
Kapitel 14	Einführung in SQL	363
Kapitel 15	Formulare: Grundlagen	387
Kapitel 16	Standard-Steuerelemente in Formularen	433
Kapitel 17	ActiveX-Steuerelemente in Formularen	485
Kapitel 18	Haupt- und Unterformular	493
Kapitel 19	PopUp- und Dialogfeldformulare	505

Kapitel 20	Ausdrücke in Formularen	511
Kapitel 21	Berichte: Grundlagen	529
Kapitel 22	Komplexe Berichte	545
Kapitel 23	Ausdrücke in Berichten	555
Kapitel 24	Menüleisten, Symbolleisten und Kontextmenüs verwalten	559
Kapitel 25	Access benutzerdefiniert anpassen	567
Kapitel 26	Makros: Grundlagen	575
Kapitel 27	Ereignisse	601
Kapitel 28	Ereignisabhängige Makros in Formularen und Steuerelementen	611
Kapitel 29	VBA (Visual Basic for Applications): Grundlagen	625
Kapitel 30	Programmieren mit VBA	675
Kapitel 31	Datenzugriff mit DAO (Data Access Objects) und ADO (ActiveX Data Objects)	703
Kapitel 32	Datenmodellierung für Fortgeschrittene	731
Kapitel 33	Integration von Access 2000 in andere Office 2000-Anwendungen	783
Kapitel 34	Access im Inter- und Intranet	797
Kapitel 35	Arbeiten in einer Mehrbenutzerumgebung	837
Kapitel 36	Zugriffsrechte für Datenbank oder einzelne Objekte festlegen	845
Kapitel 37	Access-Projekte als Client/Server-Datenbank: Grundlagen	857
Kapitel 38	Access-Projekt erstellen und bearbeiten	873
Kapitel 39	Ausdrücke, Funktionen und Operatoren	903
Kapitel 40	Literaturverzeichnis	915
	Index	917

Inhalt

Kapitel 1		**Einleitung und Überblick**	**15**
Kapitel 2		**Für alte Bekannte: Neuerungen in Access 2000**	**19**
	2.1	Die beiden bedeutsamsten neuen Features: Datenzugriffsseiten und Access-Projekte	19
	2.2	Datenbankfenster	20
	2.3	Arbeiten mit Daten und Entwerfen von Datenbanken	20
	2.4	Formulare und Berichte	21
	2.5	Symbolleisten, Menüleisten und Kontextmenüs	21
	2.6	Schützen, Warten und Konvertieren von Datenbanken	22
	2.7	Was ist neu bei den Beispielanwendungen	22
Kapitel 3		**Access installieren**	**23**
	3.1	Access installieren	23
	3.2	Komponenten nachträglich installieren und deinstallieren	27
	3.3	Access-Datenbanken früherer Versionen verwenden und konvertieren	28
Kapitel 4		**Einführung in Access**	**29**
	4.1	Konzept	29
	4.2	Oberfläche	48
Kapitel 5		**Einführungsbeispiel: Eine einfache relationale Datenbank erstellen**	**63**
	5.1	Problemstellung	64
	5.2	Datenbankdesign	65
	5.3	Neue Datenbank erstellen	66
	5.4	Tabelle Personal definieren	68
	5.5	Personaldaten in Tabelle eingeben	83
	5.6	Tabelle Projekte definieren	89
	5.7	Beziehung zwischen den Tabellen definieren	98
	5.8	Projektdaten in die Tabelle eingeben	102
	5.9	Formular Personal erstellen	103
	5.10	Personaldaten in Formular eingeben	104
	5.11	Formular Personal und Projekte erstellen	105
	5.12	Position und Größe der Steuerelemente im Formular Personal und Projekte ändern	110

	5.13	Personal- und Projektdaten im Formular bearbeiten	115
	5.14	Abfrage: Unerledigte Projekte mit einem Auftragswert von mindestens 100.000 DM	116
	5.15	Datensätze filtern	120
	5.16	Abfrage und Formular: Projekte und Personal	121
Kapitel 6		**Daten bearbeiten**	**123**
	6.1	Objekte zur Bearbeitung von Datensätzen öffnen	123
	6.2	Hinzufügen von Datensätzen	124
	6.3	Bearbeiten von Datensätzen	125
	6.4	Speichern von Daten	136
	6.5	Datenblatt individuell gestalten	136
	6.6	Kopieren und Verschieben von Daten	144
	6.7	Löschen von Daten	147
	6.8	Suchen von Daten	148
	6.9	Suchen und Ersetzen von Daten	149
	6.10	Filtern von Datensätzen	149
	6.11	Sortieren	162
	6.12	Aktualisierung der Datenanzeige in Formularen und Datenblättern	163
Kapitel 7		**Eine Datenbank konzipieren**	**167**
	7.1	Welche Aufgaben?	167
	7.2	Welche Tabellen?	168
	7.3	Welche Felder in den Tabellen?	170
	7.4	Primärschlüsselfelder	171
	7.5	Welche Beziehungen zwischen den Tabellen?	173
Kapitel 8		**Tabellen: Grundlagen**	**177**
	8.1	Was ist eine Tabelle?	177
	8.2	Tabellen entwerfen	178
	8.3	Felder hinzufügen, löschen, kopieren, verschieben	185
	8.4	Felddatentypen festlegen	189
	8.5	Feldeigenschaften festlegen	193
	8.6	Nachschlagefeld erstellen	206
	8.7	Benutzerdefinierte Anzeigeformate	215
	8.8	Tabelleneigenschaften festlegen	223
	8.9	Primärschlüssel setzen, ändern und löschen	227
	8.10	Index erstellen	228
	8.11	Beziehungen zwischen Tabellen festlegen	230
	8.12	Tabelle drucken	238
Kapitel 9		**Tabellen importieren, exportieren und verknüpfen**	**241**
	9.1	Importieren von Tabellen	242
	9.2	Exportieren von Tabellen	252
	9.3	Verknüpfen von Tabellen	253
Kapitel 10		**Abfragen: Grundlagen**	**261**
	10.1	Was ist eine Abfrage?	261
	10.2	Wozu werden Abfragen verwendet?	262

	10.3	Wie Sie Abfragen erstellen	265
	10.4	Abfrageeigenschaften einstellen	267
	10.5	Voreinstellungen für neuen Abfrageentwurf festlegen	272
	10.6	SQL	274
	10.7	Abfrage drucken	275
Kapitel 11		**Auswahlabfragen**	**277**
	11.1	Felder hinzufügen und bearbeiten	277
	11.2	Sortieren	288
	11.3	Kriterien	290
	11.4	Duplikate unterdrücken	302
	11.5	Verknüpfungen bei Mehrtabellenabfragen	302
	11.6	Datenaktualisierung bei Ein- und Mehrtabellenabfragen	318
Kapitel 12		**Komplexe Abfragen**	**319**
	12.1	Parameterabfragen	319
	12.2	Abfragen mit Kriterien aus Steuerelementen in einem Formular (Query-by-Form)	323
	12.3	Aggregierte Berechnungen	334
	12.4	Kreuztabellenabfragen	343
	12.5	Abfragen mit SQL formulieren oder modifizieren	348
Kapitel 13		**Aktionsabfragen**	**353**
	13.1	Abfragen, die neue Tabellen erzeugen	353
	13.2	Löschabfragen	356
	13.3	Anfügeabfragen	358
	13.4	Aktualisierungsabfragen	360
Kapitel 14		**Einführung in SQL**	**363**
	14.1	Was ist SQL?	363
	14.2	Vorteile der Verwendung von SQL in Access	365
	14.3	Überblick über die möglichen SQL-Anweisungen	366
	14.4	Anweisungen, Operationen und Deklarationen zur Datenmanipulation	368
	14.5	Klauseln	377
	14.6	Unterabfragen	383
	14.7	Datumsangaben in Access-SQL-Anweisungen	384
Kapitel 15		**Formulare: Grundlagen**	**387**
	15.1	Was ist ein Formular?	387
	15.2	Wie Sie Formulare erstellen	390
	15.3	Formularansichten	393
	15.4	Das Formularfenster in der Entwurfsansicht	395
	15.5	Steuerelemente: Einführung und Überblick	404
	15.6	Steuerelemente markieren und bearbeiten	411
	15.7	Eigenschaften von Steuerelementen, Formular und Bereichen einstellen	421
	15.8	Ereignisprozeduren und -makros in Formularen	426
	15.9	Bereiche einfügen und entfernen	429
	15.10	Formular für den Druck einrichten	430
	15.11	Formulardefinition drucken	431

Kapitel 16 **Standard-Steuerelemente in Formularen** **433**
16.1 Textfeld 433
16.2 Bezeichnungsfeld 435
16.3 Kontrollkästchen, Optionsfeld und Umschaltfläche zur Darstellung von Ja/Nein-Werten 435
16.4 Optionsgruppe und Optionsfelder 437
16.5 Listenfeld und Kombinationsfeld 443
16.6 Mit einem Listen- oder Kombinationsfeld einen Datensatz im Formular suchen 452
16.7 Unterformular 457
16.8 Register 460
16.9 Diagramm 465
16.10 Gebundenes und ungebundenes Objektfeld 475
16.11 Bild 479
16.12 Linie 480
16.13 Rechteck 480
16.14 Seitenwechsel 480
16.15 Befehlsschaltfläche 481
16.16 Reihenfolge der Steuerelemente ändern 483

Kapitel 17 **ActiveX-Steuerelemente in Formularen** **485**
17.1 Überblick 485
17.2 ActiveX-Steuerelement einfügen 486
17.3 Registrieren eines ActiveX-Steuerelements 491

Kapitel 18 **Haupt- und Unterformular** **493**
18.1 Verfahren zum Erstellen von Haupt- und Unterformularen 493
18.2 Ein Unterformular durch Ziehen aus dem Datenbankfenster einfügen 494
18.3 Automatisches und manuelles Verknüpfen von Haupt- und Unterformular 495
18.4 Ansichten des Unterformulars 496
18.5 Hauptformular mit einem verknüpften Unterformular im verknüpften Unterformular 498
18.6 n:1-Formular erstellen 500

Kapitel 19 **PopUp- und Dialogfeldformulare** **505**

Kapitel 20 **Ausdrücke in Formularen** **511**
20.1 Ausdruck in Steuerelement eingeben 511
20.2 Beispiele 517

Kapitel 21 **Berichte: Grundlagen** **529**
21.1 Einführung und Überblick 529
21.2 Berichte und Formulare: Gemeinsamkeiten und Unterschiede 531
21.3 Neuen Bericht erstellen 532
21.4 Leeren Bericht erstellen und bearbeiten 543
21.5 Bericht drucken 544

Kapitel 22 **Komplexe Berichte** **545**
22.1 Mehrspaltenbericht 545
22.2 Sortieren 546
22.3 Gruppieren 547
22.4 Haupt- und Unterbericht 550

Kapitel 23 **Ausdrücke in Berichten** **555**
23.1 Zusammenfassende Funktionen 555
23.2 Textfeldeigenschaft Laufende Summe 557
23.3 Doppelte Feldinhalte ausblenden 558

Kapitel 24 **Menüleisten, Symbolleisten und Kontextmenüs verwalten** **559**
24.1 Das Konzept 559
24.2 Menüleisten, Symbolleisten und Kontextmenüs erstellen 560
24.3 Schaltflächen, Menüs und Befehle hinzufügen und löschen 562
24.4 Eigenschaften von Menüleisten, Symbolleisten und Kontextmenüs einstellen 563
24.5 Einbinden von Menüleisten, Symbolleisten und Kontextmenüs in Formulare, Berichte oder Steuerelemente 563
24.6 Bild oder Text von Schaltflächen und Menübefehlen bearbeiten 564

Kapitel 25 **Access benutzerdefiniert anpassen** **567**
25.1 Startoptionen für eine bestimmte Datenbank 568
25.2 Optionen für Access 571
25.3 Starten von Access mit Befehlszeilenoptionen 572

Kapitel 26 **Makros: Grundlagen** **575**
26.1 Einführung und Überblick 575
26.2 Makro erstellen 581
26.3 Makro ausführen 590
26.4 Makrogruppen bilden 592
26.5 Verfügbare Makroaktionen: Übersicht 594
26.6 Autoexec-Makro 597
26.7 Makrodefinitionen drucken 599
26.8 Makros kopieren 600

Kapitel 27 **Ereignisse** **601**
27.1 Was sind Ereignisse? 601
27.2 Liste sämtlicher Ereignisse nach Kategorien 603

Kapitel 28 **Ereignisabhängige Makros in Formularen und Steuerelementen** **611**
28.1 Befehlsschaltfläche 611
28.2 Gültigkeitsprüfung (Beispiel) 614
28.3 Steuerelementen Werte zuweisen (Beispiel) 615
28.4 Datensätze filtern (Beispiel) 617
28.5 Datensatz suchen (Beispiel) 619
28.6 Formulare synchronisieren 621

Kapitel 29 **VBA (Visual Basic for Applications): Grundlagen** **625**
29.1 Was ist VBA? 625
29.2 Prozeduren, Funktionen und Sub-Prozeduren: Sprachvereinbarung zur Klarheit 626
29.3 Für welche Zwecke lassen sich VBA-Prozeduren einsetzen? 627
29.4 Visual Basic-Entwicklungsumgebung 628
29.5 VBA-Prozedur erstellen und bearbeiten 641
29.6 Testen des Codes in Testfenstern 649
29.7 Funktionen und Sub-Prozeduren 656
29.8 Entscheidungs- und Schleifenstrukturen 664
29.9 Deutsche und englische Sprache 673

Kapitel 30 **Programmieren mit VBA** **675**
30.1 Das Access 2000 Objektmodell 675
30.2 Variablen und Konstanten 680
30.3 Objektvariablen 686
30.4 Programmieren mit Klassenmodulen 690
30.5 Fehlersuche 694
30.6 Fehlerbehandlung 696

Kapitel 31 **Datenzugriff mit DAO (Data Access Objects) und ADO (ActiveX Data Objects)** **703**
31.1 Strategiewechsel im Datenzugriffskonzept 703
31.2 Das DAO-Objektmodell 704
31.3 DAO-Objekte Objektvariablen zuweisen 706
31.4 Datenmanipulation mit DAO-Recordset-Objekten 713
31.5 Datenzugriff mit ADO 723

Kapitel 32 **Datenmodellierung für Fortgeschrittene** **731**
32.1 Grundlegende Überlegungen zur Datenmodellierung 731
32.2 Tabellen bestimmen 739
32.3 Felder bestimmen 741
32.4 Primärschlüssel und Indizes bestimmen 751
32.5 Das Redundanzproblem 757
32.6 Beziehungen bestimmen 771
32.7 Datenbankkonzept dokumentieren 779

Kapitel 33 **Integration von Access 2000 in andere Office 2000-Anwendungen** **783**
33.1 Überblick 783
33.2 Access 2000 als Client: Beispiel 791
33.3 Access 2000 als Server: Beispiel 794

Kapitel 34 **Access im Inter- und Intranet** **797**
34.1 Überblick 797
34.2 Hyperlinks 800
34.3 Eingeben einer Hyperlink-Adresse in der Formular- oder Datenblattansicht 803
34.4 Bearbeiten eines Hyperlinks 803
34.5 Hyperlink-Basis für eine Datenbank festlegen 803
34.6 Drei Arten von Webseiten in Access 2000 – Überblick 804

	34.7	Ex- und Importieren eines Access-Objekts im statischen HTML-Format oder als Active Server Page	805
	34.8	Datenzugriffsseiten	807
Kapitel 35		**Arbeiten in einer Mehrbenutzerumgebung**	**837**
Kapitel 36		**Zugriffsrechte für Datenbank oder einzelne Objekte festlegen**	**845**
	36.1	Das Sicherungskonzept	845
	36.2	Berechtigungsprüfung der Arbeitsgruppe ein- und abschalten	848
	36.3	Administrator-, Benutzer- und Gruppenkonten erstellen und verwalten	850
	36.4	Berechtigungen zuweisen und entfernen	853
	36.5	Kennwort ändern und löschen	854
	36.6	Besitzer ändern	855
	36.7	Datenbank verschlüsseln	856
Kapitel 37		**Access-Projekte als Client/Server-Datenbank: Grundlagen**	**857**
	37.1	Bedeutung und Funktionsweise von Microsoft Access-Projekten	857
	37.2	MSDE installieren, starten und nutzen	859
	37.3	Vergleich von Access-Datenbanken und Access-Projekten	864
	37.4	Kennenlernen eines Access-Projekts: Beispielprojekt NordwindCS.adp	865
Kapitel 38		**Access-Projekt erstellen und bearbeiten**	**873**
	38.1	Access-Datenbank mit dem Upsizing-Assistenten in Access-Projekt umwandeln	873
	38.2	Access-Projekt neu erstellen	879
	38.3	Tabellen und Datenbankdiagramme	885
	38.4	Sichten (Views) erstellen und bearbeiten	891
	38.5	Gespeicherte Prozeduren erstellen und bearbeiten	894
	38.6	Datenquellen und Abfrageparameter in Formularen und Berichten	898
	38.7	Sicherungskopien erstellen und wiederherstellen	899
	38.8	Access-Projekt portieren	900
	38.9	SQL-Datenbank löschen	901
Kapitel 39		**Ausdrücke, Funktionen und Operatoren**	**903**
	39.1	Komponenten eines Ausdrucks	903
	39.2	Wie Sie einen Ausdruck eingeben	904
	39.3	Operatoren	907
	39.4	Stellvertreterzeichen	908
	39.5	Arbeiten mit dem Ausdrucks-Generator	909
	39.6	Funktionen und Operatoren	913
Kapitel 40		**Literaturverzeichnis**	**915**
		Index	**917**

Kapitel 1

Einleitung und Überblick

Dem Titel entsprechend wendet sich dieses Buch vor allem an Leser, die Access professionell nutzen wollen. Erfahrungen mit dem Programm werden nicht vorausgesetzt. Vielmehr wird in jedes Gebiet so eingeführt, daß Sie es sich neu erschließen können. Viele werden ein Buch dieser Art nicht vom Anfang bis zum Ende zusammenhängend durcharbeiten wollen. Im allgemeinen werden Sie vermutlich selektiv vorgehen wollen, um sich ein Gebiet anzueignen, an dem Sie momentan besonders interessiert sind. Ich habe versucht, einem solchen Bedürfnis entgegenzukommen, indem ich die einzelnen Kapitel so bausteinartig wie möglich geschrieben habe. Auf der anderen Seite lassen sich gewisse Zusammenhänge nur entwickeln, wenn andere bereits bekannt sind. Daher setzen einzelne Passagen dieses Buches die Kenntnis anderer Teile voraus. Zur Orientierung über die 40 Kapitel seien diese im folgenden kurz charakterisiert.

Beispiel CD-ROM

Dem Buch ist eine Beispiel CD-ROM beigefügt. Sie enthält in komprimierter Form Datenbanken, die als Begleitmaterial zu diesem Buch dienen, das an vielen Stellen Bezug darauf nimmt. Diese Datenbanken erheben nicht den Anspruch kompletter Anwendungen. Dafür liefert Access mit den im Programmpaket enthaltenen Datenbanken *Nordwind.mdb* und *NordwindCS.adp* mit ihren vielen und gut ausgearbeiteten Tabellen, Abfragen, Formularen, Berichten und Makros sehr gute Beispiele. Auch auf diese Datenbanken wird vielfach Bezug genommen. Auf der anderen Seite ist *Nordwind.mdb* so groß und komplex, daß es nicht immer leicht fällt, die Zusammenhänge sofort zu durchschauen. Die Datenbanken der Beispiel CD-ROM konzentrieren sich dagegen jeweils bewußt auf wenige, von didaktischen Absichten bestimmte Aspekte, so daß die Struktur ihrer Tabellen, Formulare etc. leichter transparent werden sollte.

Im einzelnen dienen die drei Datenbanken den folgenden Zwecken: *Projekte.mdb* weist sehr einfache, aber für den Umgang mit Access grundlegende Lösungen auf. Der Kern dieser Datenbank wird in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, nachgearbeitet, darüber hinausgehende Lösungen dienen an anderen Stellen des Buches als Beispiele. *Adressen.mdb* ist in ihrer Tabellenstruktur sehr simpel, denn diese Datenbank verwendet nur eine Tabelle. An den Formularen, Makros und VBA-Prozeduren von *Adressen.mdb* können Sie vor allem lernen, wie Sie eine Adressendatei unter Einschluß von leistungsfähigen und bequemen Suchmöglichkeiten zum einen direkt mit Suchkriterien in einem Formular, zum zweiten mit Makros und zum dritten mit VBA-Prozeduren verwalten können. Dabei können Sie diese drei Arbeitsweisen unmittelbar vergleichen. Die dritte Datenbank, *Auftrag.mdb*, dient im wesentlichen zwei Zwecken. In ihr ist erstens eine m:n-Beziehung von Tabellen realisiert. Für die Verwaltung einer derartigen Beziehung bietet Access kein standardmäßiges Instrument an, so daß der Benutzer sich ein solches selbst entwickeln muß. Deshalb wird in *Auftrag.mdb* zweitens gezeigt, wie Sie eine solche Aufgabe mit Hilfe von Formularen und VBA-Prozeduren lösen können.

☞ Auf der Beispiel CD-ROM finden Sie außerdem den gesamten Text dieses Buches als Datei im *PDF*-Format. Dieser Service soll es Ihnen ermöglichen, elektronisch nach bestimmten Begriffen zu suchen. Außerdem können Sie an verschiedenen Stellen auf Hyperlink-Verweise klicken, wodurch Sie unmittelbar zu der betreffenden Stelle verzweigen.

Einführung in Access

Obgleich die Oberflächen von Windows-Programmen einander in wesentlichen Merkmalen wie z.B. der Fenstertechnik, dem Umgang mit der Maus etc. gleichen, hat doch jedes Programm seine eigene Oberflächenphilosophie. Dies gilt in ganz besonderem Maße für Access, denn seine Art der Benutzerführung ist unter sehr konsequenter Ausnutzung der Möglichkeiten graphischer Benutzeroberfläche auf den speziellen Umgang mit einem Datenbanksystem abgestellt, indem es dessen Hauptaufgaben zum strukturierenden Merkmal der Benutzeroberfläche gemacht hat. Dies macht den Umgang mit Access letztlich sehr bequem und einfach. Dennoch müssen Sie ihn zunächst erlernen. Dies geschieht am besten dadurch, daß Sie sich, ohne sich zunächst mit größeren sachlichen Datenbankproblemen zu belasten, blätternd, lesend, probierend etc. in einer fertigen Datenbank bewegen. Kap. 4, *Einführung in Access*, stellt mit Hilfe eines Spaziergangs durch die Datenbank *Nordwind.mdb* das Oberflächen- und Sachkonzept der Software Access vor. Ich empfehle Ihnen sehr, dieses Kapitel nachzuvollziehen, bevor Sie mit Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, eine eigene Datenbank erstellen. Der praktischen Einführung in Kap. 5 liegen die Erfahrungen zugrunde, die ich in zahlreichen Einführungskursen zu Access sammeln konnte. Es spricht nicht nur für die Teilnehmer

dieser Kurse, sondern vor allem für Access, daß es möglich ist, an einem Unterrichtstag eine komplette relationale Datenbank mit Formularen und Abfragen zu erstellen, und dies mit Kursteilnehmern, die sich mit Datenbanksystemen bis dahin überhaupt nicht beschäftigt hatten.

Die sechs Hauptgebiete

Access gliedert die gesamte Datenbankverwaltung – eine andere Bezeichnung ist DBMS für *Datenbank Management System* – in die sechs großen Gebiete *Tabellen*, *Abfragen*, *Formulare*, *Berichte*, *Makros* und *Module* (letztere für VBA-Prozeduren). Dieser Gliederung folgt auch die Benutzeroberfläche. Sie ist für das gesamte Programm derart prägend, daß es wenig sinnvoll wäre, in einem Buch einer anderen Gliederung zu folgen. Daher folgen die Kapitel 6 bis 30 den sechs Access-Hauptaufgaben, so daß Sie sich entsprechend orientieren können. Diese Kapitel beschreiben insgesamt den wesentlichen Inhalt des DBMS Access. Sie können sie größtenteils unabhängig voneinander lesen. Allerdings ist es nicht ratsam, sich beispielsweise gründlich mit Abfragen und Formularen zu beschäftigen, ohne sich zuvor über die wesentlichen Möglichkeiten und Beschränkungen von Tabellen informiert zu haben. Falls Sie sich Access systematisch Schritt für Schritt aneignen wollen (wohl besser: falls Sie sich dies zeitlich leisten können!), empfiehlt es sich, die Kapitel 6 bis 31 nacheinander zu lesen.

Weitere Beschreibungen

Weitere Informationen, die sich nicht systematisch in die genannten sechs Hauptgebiete integrieren lassen, werden in den Kapiteln 31 ff. behandelt. Diese Kapitel können Sie als Nachschlagekapitel betrachten, die nicht unbedingt systematisch aufeinander aufbauen. Sie werden die wesentlichen Inhalte dieser Kapitel anhand ihrer Überschriften identifizieren können, so daß an dieser Stelle nicht im einzelnen darauf eingegangen werden soll. Hervorgehoben seien aber zwei Gebiete: 1) Kap. 32, *Datenmodellierung für Fortgeschrittene*. Dieses mit 50 Seiten relativ umfangreiche Kapitel steht in der Gliederung nur deshalb so weit hinten, weil es viele Informationen enthält, die der Einsteiger in Access nicht sofort berücksichtigen muß. Für Leser mit bisher geringer Erfahrung mit Datenbanken sollten zunächst die Informationen des kurzen Kap. 7, *Eine Datenbank konzipieren*, ausreichen. Sobald Sie jedoch etwas anspruchsvollere Datenbanken entwickeln, sei Ihnen dringend die Lektüre von Kap. 32 empfohlen, in dem Sie auch die Begründung für diesen Rat finden: Dort wird u.a. deutlich gemacht, daß dem Prozeß der Datenmodellierung zentrale Bedeutung beim Entwickeln einer Datenbank zukommt, und daß die Nichtbeachtung gewisser Regeln bei der Umsetzung von Zusammenhängen der realen Welt in ein Datenbankmodell zu schwerwiegenden Widersprüchen und Disfunktionalitäten führen kann. 2) Kap. 37, *Access-Projekte als Client/Server-Datenbank: Grundlagen*, und Kap. 38, *Access-Projekt erstellen und*

bearbeiten. Diese beiden Kapitel erklären die in Access 2000 neue Möglichkeit, durch Verbindung einer Access-Anwendung mit einem Microsoft SQL Server eine echte Client/Server-Anwendung zu realisieren. Derartige Anwendungen werden *Access-Projekte* (*.adp*-Dateien) genannt, im Unterschied zu den bisherigen und in der praktischen Arbeit nach wie vor im Vordergrund stehenden *Access-Datenbanken* (*.mdb*-Dateien).

Online-Hilfe

Die Online-Hilfe von Access ist außergewöhnlich umfangreich und differenziert. Dort finden Sie außer Einführungen in thematisch geordnete Gebiete praktisch jede Einzelheit des DBMS Access detailliert beschrieben. Zusammen mit den Suchmöglichkeiten, die das System bietet, stellt die Hilfe ein sehr leistungsfähiges Nachschlagewerk dar. Diesen Sachverhalt habe ich auch beim Abfassen dieses Buches in Rechnung gestellt. Es ist unmöglich, eine so umfangreiche Software wie Access in einem einzigen Buch vertretbaren Umfangs in jeder Hinsicht vollständig zu dokumentieren. Daher müssen immer Kompromisse geschlossen werden. Ich habe mich im Zweifel stets für das Fortlassen von beschreibenden Einzelheiten zugunsten der Darstellung übergreifender Zusammenhänge und praktischer Beispiele entschieden, denn Einzelheiten können Sie in der Online-Hilfe nachschlagen.

Kapitel 2

Für alte Bekannte: Neuerungen in Access 2000

In den folgenden Punkten konzentriere ich mich auf die wichtigsten Neuerungen, vor allem auf solche, die neue Funktionalitäten mit sich bringen. Dagegen vernachlässige ich Neuerungen, die allein in einer geänderten Oberfläche bestehen.

2.1 Die beiden bedeutsamsten neuen Features: *Datenzugriffsseiten und Access-Projekte*

Diese beiden Features stellen die beiden überragenden Neuerungen in Access 2000 dar, gegen die alle anderen Neuerungen klein erscheinen.

- Die *Datenzugriffsseiten* können beispielsweise im Internet oder einem Intranet anderen Benutzern zur Verfügung gestellt werden, die sie mit dem *Microsoft Internet Explorer 5* öffnen und aktiv bearbeiten können, so daß z.B. Datenänderungen, die sie vornehmen, an der Quelle aktualisiert werden. Eine Datenzugriffsseite kann sogar eine PivotTable enthalten, die vom Benutzer interaktiv bearbeitet werden kann. Datenzugriffsseiten werden in diesem Buch in Kap. 34, *Access im Inter- und Intranet*, Punkt 34.8, Datenzugriffsseiten, behandelt.
- Access-Projekte sind in ihrer Bedeutung vermutlich noch weitreichender als die Datenzugriffsseiten, denn sie ermöglichen eine direkte Anbindung von Access an einen Microsoft SQL Server, so daß echte Client/Server-Anwendungen realisiert werden können. Dadurch wird Access in einem bisher nicht gekannten Ausmaß skalierbar. Diese Möglichkeit kann von jedem Benutzer, der über eine Access-Lizenz verfügt, genutzt werden, denn die Installations-CD-ROM enthält einen SQL Server in Form der MSDE (Microsoft Data Engine), die voll kompatibel mit dem Microsoft SQL Ser-

ver 7.0 ist. Diese neuen Möglichkeiten werden in diesem Buch in Kap. 37, *Access-Projekte als Client/Server-Datenbank: Grundlagen*, und Kap. 38, *Access-Projekt erstellen und bearbeiten*, behandelt.

2.2 Datenbankfenster

- **Organisieren von Datenbankobjekten in Gruppen**. Zusätzlich zu den integrierten Objektgruppen wie *Tabellen* oder *Formulare* können benutzerdefinierte Gruppen angelegt werden, in die beliebige Datenbankobjekte aufgenommen werden können. Eine Gruppe mit ihren Objekten erscheint im Datenbankfenster gleichberechtigt neben den integrierten Objektlisten.
- **Auswählen von Objekten durch Eingeben des Namens**. Beispielsweise können Sie, während Sie die Objektliste *Formulare* anzeigen, das Formular *Umsatzanalyse* auswählen, indem Sie einfach *u* oder *um* eingeben.

2.3 Arbeiten mit Daten und Entwerfen von Datenbanken

- **Verwenden der Sperrung auf Datensatzebene**. Microsoft Access-Datenbanken unterstützen jetzt zusätzlich zur Sperrung auf Seitenebene (bei der alle Datensätze auf einer Seite gesperrt werden) auch die Sperrung auf Datensatzebene. Sie aktivieren die Sperrungsebene mit der neuen Option *DB mit Sperrung auf Datensatzebene öffnen* (Menü Extras, Befehl *Optionen*, Registerkarte *Weitere*).
- **Suchen und Ersetzen**. Sie können sich jetzt frei zwischen den Dialogfeldern *Suchen* und *Ersetzen* sowie den Daten in der Sicht oder dem Fenster bewegen.
- **Unterdatenblätter**. Sie können ein Unterdatenblatt verwenden, um verwandte oder verknüpfte Daten im Datenblatt einer Tabelle, einer Abfrage bzw. eines Formulars oder in einem Unterformular anzuzeigen oder zu bearbeiten, jeweils in der gleichen Ansicht. Die Tabelle *Lieferanten* in der Beispieldatenbank *Nordwind* beispielsweise besitzt eine 1:n-Beziehung mit der Tabelle *Artikel*. Daher können Sie für jeden Datensatz in der Tabelle *Lieferanten* in der Datenblattansicht die verwandten Datensätze der Tabelle *Artikel* in einem Unterdatenblatt anzeigen und bearbeiten.
- **Automatische Korrektur von durch das Umbenennen von Feldern verursachten Fehlern**. *Objektnamen-Autokorrektur* korrigiert automatisch übliche Nebenwirkungen, die durch das Umbenennen von Formularen, Berichten, Tabellen, Abfragen, Feldern, Textfeldern oder anderen Steuerelementen ausgelöst werden.
- **Vorteile durch Unicode-Unterstützung**. Sie können in Ihren Daten die Schriftzeichen jeder Sprache verwenden, die *Unicode* unterstützt.
- **Verwenden des Euro-Symbols**. Um Eurobeträge zusammen mit anderen Währungen anzuzeigen, können Sie die Euro-Einstellung (€#.###,##) der *Format*-Eigenschaft verwenden. Alternativ können Sie das Euro-Symbol

(€) eingeben, indem Sie im numerischen Tastenblock bei gedrückter [Num⇩]-Taste [Alt]+0128 eingeben. Beim Einfügen oder Importieren von Daten, in denen das Euro-Symbol (€) enthalten ist, (z.B. aus Excel 2000 in Access 2000), speichert Access das Euro-Symbol unabhängig von dem in der Windows-Systemsteuerung unter *Ländereinstellungen* festgelegten Währungssymbol. Schließlich können Sie die *EuroConvert*-Funktion dazu verwenden, einzelne Währungen in andere zu konvertieren, indem sie den Euro als Zwischenwert benutzen.

- **Ausdrucken von Beziehungen**. Drucken Sie einen Bericht mit allen Beziehungen Ihrer Access-Datenbank, so wie sie im Fenster *Beziehungen* angezeigt werden.
- **Verwalten von Beziehungen mittels der Tastatur**. Beziehungen und Verknüpfungen können Sie auch über die Tastatur erstellen, bearbeiten oder löschen.
- **Verwendung von Microsoft ActiveX Data Objects (ADO)**. Mit Microsoft ActiveX Data Objects (ADO) können Sie auf Daten in einem Datenbankserver über beliebige OLE DB-Anbieter zugreifen und sie bearbeiten.

2.4 Formulare und Berichte

- **Gruppieren von Textfeldern und anderen Steuerelementen**. Verwenden Sie den Befehl *Gruppieren* im Menü *Format*, um zusammengehörige Textfelder oder andere Steuerelemente in einem Formular oder Bericht zu gruppieren.
- **Definieren von Regeln zur bedingten Formatierung für Textfelder und andere Steuerelemente**. Mit dem Befehl *Bedingte Formatierung* im Menü *Format* können Sie die Schriftfarbe, Schriftgröße, Hintergrundfarbe und andere Auszeichnungen definieren, die eine Reaktion auf Benutzereingaben in Formulare signalisieren.
- **Verteilen von Berichten an Benutzer, die nicht über Microsoft Access verfügen**. Access-Berichte können Sie in das Snapshot-Dateiformat (.snp) für Berichte exportieren. Mit dem Snapshot Viewer können Sie Berichts-Snapshots anzeigen, ausdrucken und als E-Mail-Nachricht versenden.

2.5 Symbolleisten, Menüleisten und Kontextmenüs

- **Verwendung personalisierter Menüs und Symbolleisten**. Sie können Menüs so erweitern, daß sie alle Befehle anzeigen, oder bestimmte Befehle auswählen, die Ihrem persönlichen Menü hinzugefügt werden sollen. Access 2000 richtet Menüs und Symbolleisten entsprechend Ihrer Arbeitsweise ein, so daß dort nur die Befehle und Symbolleistenschaltflächen angezeigt werden, die Sie am häufigsten verwenden.

- **Anordnung der Symbolleisten nebeneinander.** Klicken Sie auf *Weitere Schaltflächen*, falls auf der Symbolleiste kein Platz mehr für die gewünschte Schaltfläche ist. Sie können die Symbolleiste vergrößern, wenn Sie noch mehr Platz für weitere Schaltflächen benötigen. Wenn Sie auf eine Schaltfläche geklickt haben, wird sie in der Symbolleiste mit den zuletzt verwendeten Schaltflächen angezeigt.
- **Zuweisen von Hyperlinks zu Symbolleistenschaltflächen oder Menübefehlen.** Sie können einen Hyperlink einer Symbolleistenschaltfläche oder einem Menübefehl zuweisen und so auf Speicherorte auf Ihrem Computer, im Netzwerk, einem Intranet oder im Internet zugreifen.

2.6 Schützen, Warten und Konvertieren von Datenbanken

- **Schützen des Codes mit dem Visual Basic für Applikationen-Kennwortschutz.** Module sowie hinter Formularen und Berichten stehende Module werden jetzt durch ein VBA-Kennwort geschützt, das Sie im Visual Basic-Editor erstellen. Sie werden nicht mehr durch Datensicherheit auf Benutzerebene geschützt.
- **Verbessertes Komprimierungs-Dienstprogramm.** Komprimieren Sie Microsoft Access-Datenbanken und Microsoft Access-Projekte mit einem verbesserten Dienstprogramm, das Komprimierung und Reparatur in einem einzigen Vorgang durchführt und sicherer und effektiver arbeitet.
- **Automatische Komprimierung.** Aktivieren Sie im Dialogfeld *Optionen*, Registerkarte *Allgemein*, die Option *Beim Schließen komprimieren*, um eine Access-Datenbank oder ein Access-Projekt beim Schließen automatisch zu komprimieren.
- **Konvertieren von Datenbanken in Microsoft Access 97-Format.** Access 2000-Datenbanken können Sie in das Access 97-Dateiformat konvertieren.

2.7 Was ist neu bei den Beispielanwendungen

- **Nordwind GmbH.** Die Beispielanwendung *Nordwind* wurde für Benutzer entwickelt, die noch nicht mit Microsoft Access-Datenbanken gearbeitet haben. *Nordwind* enthält Daten, die Sie ändern können, sowie Formulare, Berichte, Datenzugriffsseiten und andere Objekte, die Sie als Modelle für Ihre eigenen Datenbanken verwenden können.
- **NordwindCS.** Die Beispielanwendung *NordwindCS* enthält ein SQL-Skript, das die *Nordwind*-Datenbank im Microsoft SQL Server erstellt, sowie ein Microsoft Access-Projekt, das mit der *Nordwind*-Datenbank verbunden wird. Das Access-Projekt enthält Formulare, Berichte, Datenzugriffsseiten und andere Objekte, die Sie als Modelle für Ihre eigenen Access-Projekte verwenden können.

Kapitel 3

Access installieren

In diesem Kapitel wird beschrieben, wie Sie Access 2000 als Einplatzversion auf einem PC installieren, so daß Sie mit Access dann ohne Netzwerk arbeiten können. Weiterhin wird davon ausgegangen, daß Sie Access 2000 als Teil von Office 2000 installieren.

3.1 Access installieren

Gehen Sie folgendermaßen vor:

- Starten Sie Windows 9x, Windows NT oder Windows 2000.
- Legen Sie die Installations-CD-ROM in das CD-ROM-Laufwerk ein.
- Starten Sie die Datei *Setup.exe* im Basisverzeichnis der CD-ROM. Dies können Sie mit dem Explorer von Windows oder mit dem Befehl *Ausführen* aus dem Menü *Start* bewerkstelligen. Im letzteren Falle müssen Sie den Befehl in der Form *d:setup* (für ein anderes Laufwerk entsprechend anders) angeben.

Bild 3.1: Dieses Menü wird angezeigt, wenn Sie auf den Pfeil am rechten Rand klicken.

Der gesamte Installationsprozeß ist narrensicher menügesteuert. Sie müssen sich entscheiden, ob Sie eine Standard- oder eine benutzerdefinierte Installation durchführen wollen. Im letzteren Falle können Sie nicht nur bestimmen,

welche Office-Komponenten Sie installieren wollen, sondern auch, wie dies geschehen soll, denn Sie können für jede installierbare Komponente wie beispielsweise die *Standardassistenten*, die *Beispieldatenbanken* oder den *Snapshot-Viewer* zwischen den Optionen wählen, die in dem in Bild 3.1 wiedergegebenen Menü dargestellt sind. Sie schlagen dieses Menü durch Klicken auf das links nebenstehend wiedergegebene Symbol auf, vgl. auch Bild 3.2. Wenn Sie die betreffende Komponente nicht installieren wollen, wählen Sie die Option *Nicht Verfügbar*. Mit *Bei der ersten Verwendung installiert* wird die Komponente zwar registriert, die zugehörigen Dateien werden jedoch nicht auf die Festplatte kopiert. Dies geschieht erst bei der ersten Verwendung dieser Komponente, weshalb Sie dann auch die Installations-CD-ROM einlegen müssen. Dies müssen Sie auch, wenn Sie die Option *Von CD starten* wählen. Mit *Vom Arbeitsplatz starten* werden alle für die Komponente erforderlichen Dateien während des aktuellen Installationsvorgangs auf die Festplatte kopiert. Mit dieser Option wählen Sie sicherlich die bequemste, wenngleich auch speicherintensivste Arbeitsweise.

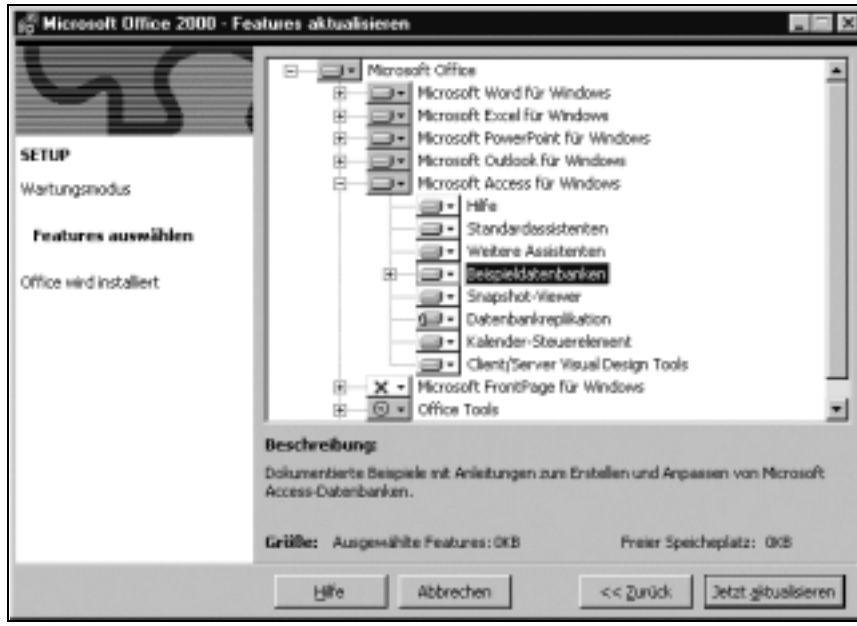


Bild 3.2: Dialogfeld des Office 2000-Setup-Assistenten zur Bestimmung der zu installierenden Komponenten, hier beim Nachinstallieren von Komponenten

Bei der Auswahl der zu installierenden Komponenten sollten Sie beachten, daß einige Access 2000-Features nur vollständig funktionieren, wenn auch zumindest einige der Office-Tools installiert sind. So setzt die volle Funktionsfähigkeit der *Datenzugriffsseiten* die Installation der *Office Webkomponenten* voraus.

Im allgemeinen fahren Sie auch gut, wenn Sie die Standardinstallation wählen. Die folgende Übersicht, die der Online-Hilfe entnommen ist, informiert Sie darüber, welche Komponenten mit welcher Installationsoption bei einer Standardinstallation installiert werden und welche unberücksichtigt bleiben.

Installation (Standard)

Feature	Position im Installationsprogramm
ClipArt-Gallery	Office-Tools; ClipArt
Dateien für Core-Support, Internationaler Support	Office-Tools; Internationaler Support
Encapsulated Postscript	Konverter und Filter; Grafikfilter
GIF (Graphic Interchange Format)	Konverter und Filter; Grafikfilter
Hilfe zu Microsoft Access	Microsoft Access für Windows
Hilfe zu Microsoft Graph	Office-Tools; Microsoft Graph
Indexerstellung-Systemsteuerung	Office-Tools
JPEG Interchange Format	Konverter und Filter; Grafikfilter
Macintosh PICT-Import	Konverter und Filter; Grafikfilter
Microsoft Access-Programmdateien	Microsoft Access für Windows
Microsoft Graph	Office-Tools
Microsoft Office Shortcut-Leiste	Office-Tools
MS Info	Office-Tools
Office Web Components	Office-Tools
Office-Assistent (ein Assistent)	Office-Tools; Office-Assistent
Rechtschreibprüfung	Office-Tools; Überprüfungstools
Standardassistenten	Microsoft Access für Windows
TIF-Dateiformat-Import	Konverter und Filter; Grafikfilter
Tool für Spracheinstellungen	Office-Tools
Typische Designs	Office-Tools; Designs
Webdiskussionen	Office-Tools
Webpublishing	Office-Tools
Windows Metafile-Import	Konverter und Filter; Grafikfilter

Installation bei erstmaliger Verwendung (Standard)

Feature	Position im Installationsprogramm
Bidirektionaler Support	Office-Tools; Internationaler Support
Chinesische Schriftzeichen (Taiwan)	Office-Tools; Internationaler Support
Chinesische Schriftzeichen (VR China)	Office-Tools; Internationaler Support

Feature	Position im Installationsprogramm
Computer Graphics Metafile (.CGM)-Import	Konverter und Filter; Grafikfilter
CorelDRAW Import	Konverter und Filter; Grafikfilter
Datenbankreplikation	Microsoft Access für Windows
Digitale Unterschrift für VBA-Projekte	Office-Tools
Enhanced Metafile (.EMF)-Import	Konverter und Filter; Grafikfilter
FPX-Import	Konverter und Filter; Grafikfilter
Hilfe zu Visual Basic	Office-Tools
Japanische Schriftzeichen	Office-Tools; Internationaler Support
Kodak Foto-CD-Import	Konverter und Filter; Grafikfilter
Koreanische Schriftzeichen	Office-Tools; Internationaler Support
Microsoft Query	Office-Tools
Nordwind Datenbank	Microsoft Access für Windows; Beispieldatenbanken
Office-Assistent (sieben weitere Assistenten)	Office-Tools; Office-Assistent
Organigramm	Office-Tools
PC Paintbrush PCX-Import	Konverter und Filter; Grafikfilter
PNG-Dateiformat-Import	Konverter und Filter; Grafikfilter
Scanner- und Camera-Add-In	Office-Tools
Snapshot Viewer	Microsoft Access für Windows
Visuelle Client/Server-Designtools	Microsoft Access für Windows
Webskripting (Microsoft Skript-Editor)	Office-Tools; HTML-Quellcode-Bearbeitung
Weitere Assistenten	Microsoft Access für Windows
Weitere Designs	Office-Tools; Designs
Windows Bitmap-Import	Konverter und Filter; Grafikfilter
WordPerfect-Grafikfilter	Konverter und Filter; Grafikfilter

Keine Installation (nur bei benutzerdefinierter Installation)

Feature	Position im Installationsprogramm
Erweiterte Supportdateien, Internationaler Support	Office-Tools; Internationaler Support
Formel-Editor	Office-Tools
Kalender-Steuerelement	Microsoft Access für Windows
Lotus VIM Mail-Support	Office-Tools

Feature	Position im Installationsprogramm
Microsoft Foto-Editor	Office-Tools
Nordwind SQL-Projektdatei	Microsoft Access für Windows; Beispieldatenbanken
Universelle Schriftart	Office-Tools; Internationaler Support

Sofern Sie keine benutzerdefinierte Installation durchführen, installieren die Setup-Programme von Microsoft Office 2000 und Microsoft Access 2000 den *Microsoft Internet Explorer 5*. Wenn Sie eine benutzerdefinierte Installation durchführen, können Sie auf die Installation des Internet Explorer verzichten. *Microsoft Internet Explorer 5* ist jedoch erforderlich, um Datenzugriffsseiten in Access 2000 verwenden zu können.

3.2 Komponenten nachträglich installieren und deinstallieren

- Rufen Sie dazu das Setup-Programm, wie im vorigen Punkt beschrieben, erneut auf. Alternativ: Rufen Sie es mit dem Tool *Software* aus der *Systemsteuerung* auf. Dann wird das in Bild 3.3 wiedergegebene Dialogfeld *Microsoft Office 2000-Wartungsmodus* angezeigt.

Bild 3.3: Dialogfeld »Microsoft Office 2000-Wartungsmodus«. Klicken Sie auf die Schaltfläche »Features h*inzufügen/ entfernen«.*

- Aktivieren bzw. deaktivieren Sie im Dialogfeld *Features aktualisieren* (vgl. oben Bild 3.2.) die zu installierenden bzw. zu deinstallierenden Komponenten, und bestätigen Sie mit *Jetzt aktualisieren*.

3.3 Access-Datenbanken früherer Versionen verwenden und konvertieren

Sie können mit Access-Datenbanken früherer Versionen in Access 2000 arbeiten. Allerdings können Sie deren Objekte dann nicht verändern, so daß Sie beispielsweise ein Formular nicht in der Entwurfsansicht bearbeiten können. Hierauf weist Access nach dem Öffnen einer derartigen Datenbank hin.

Daher sollten Sie im allgemeinen frühere Access-Datenbanken in das Format Access 2000 konvertieren. Diese Möglichkeit bietet Access an, wenn Sie eine Datenbank einer früheren Version öffnen wollen. Sie können dies aber auch unabhängig vom Öffnen im Menü *Extras* mit dem Befehl *Datenbank-Dienstprogramme*, Unterbefehl *Datenbank konvertieren* veranlassen. Die Option zum Konvertieren in die aktuelle Datenbankversion steht allerdings nur zur Verfügung, wenn keine Datenbank geöffnet ist. Dafür können Sie eine geöffnete Datenbank in ein Format der vorhergehenden Access-Version, also nach Access 97, konvertieren. Für die konvertierte Datenbank müssen Sie einen anderen Namen als den (bestehenden) der zu konvertierenden angeben. Daher verfügen Sie nach der Konvertierung in jedem Falle auch noch über die alte Version.

Kapitel 4

Einführung in Access

Der Umgang mit einer Software läßt sich im allgemeinen am besten an Hand von Beispielen, die am Bildschirm verfolgt werden können, erlernen. Dieser Regel folge ich auch im vorliegenden Buch. Gerade im Falle eines Datenbanksystems empfiehlt es sich jedoch, sich in allgemeiner Weise über grundlegende und strukturierende Begriffe und konzeptionelle Linien zu informieren, damit Sie die verschiedenen Einheiten des Systems als aufeinander bezogene erkennen können.

4.1 Konzept

4.1.1 Access - ein Datenbank-Managementsystem (DBMS)

Eine Datenbank ist eine geordnete und strukturierte Zusammenstellung von Daten. Bei Access können die Daten selbst sehr unterschiedlicher Art sein: Texte, Zahlen, Wahrheitswerte, Hyperlinks, Bilder. Einfache Datenbanken, die aus nur einer Tabelle bestehen, lassen sich im allgemeinen auch sehr einfach verwalten (Eingabe und Zugriff auf die Daten). Daher verfügen sogar verschiedene Tabellenkalkulationsprogramme (vgl. z.B. Microsoft Excel) über Möglichkeiten, einfache Datenbanken anzulegen und zu verwalten. Im allgemeinen besteht eine den Erfordernissen der Praxis genügende Datenbank dagegen aus mehreren Tabellen, deren Daten miteinander verknüpft sind. Eine solche Datenbank wird als *relationale Datenbank* bezeichnet. Die Datenhaltung in relationalen Datenbanken kann und sollte frei von *Redundanz* sein: Dieselbe Information soll nur einmal (an nur einer Stelle) gespeichert werden. Redundanzfreiheit ist vorteilhaft für Speicherplatz, vor allem aber für die Widerspruchsfreiheit der Informationen in der Datenbank (vgl. dazu genauer Kap. 7, *Eine Datenbank konzipieren* sowie Kap. 32, *Datenmodellierung für Fortgeschrittene*). Die Organisierung der Datenzugriffe in relationalen Datenbanken ist derart komplex, daß diese Aufgabe nicht mehr

quasi nebenbei von einem Tabellenkalkulationsprogramm gelöst werden kann, sondern nur noch von einer speziellen Software - einem *Relationalen Datenbank-Managementsystem* (*RDBMS*). Access ist ein RDBMS.

4.1.2 Tabellen: Speicherung der Daten

Die Daten einer Access-Datenbank werden in Tabellen gespeichert. Tabellen sind daher die Grundlagen einer Datenbank. Wie viele Tabellen eine Datenbank umfaßt und in welcher Weise die Daten der einzelnen Tabellen miteinander verknüpft sind, läßt sich nicht allgemein sagen. Diese Aufgabe muß bei den Überlegungen zum Datenbankdesign gelöst werden. Jede einzelne Tabelle ist dagegen in gleicher Weise aufgebaut.

Personal : Tabelle

PersonalCode	Nachname	Vorname	Sex	Anrede	Geburtsdatum	Einstellungsdatum	Straße
BELG	Biedermann	Sieglinde	w	Frau	22.06.63	01.07.83	Schillerstr. 7
GERSD	Gersdorff	Hubertus	m	Herr Dr.	04.05.63	01.08.85	Weidenstieg 88
LAUSE	Lauser	Hannelore	w	Frau Dr.	02.11.43	01.05.90	Fröbelstr. 12
LEJEU	Lejeune	Micaela	w	Frau	24.12.68	01.11.92	
MAHLM	Mahlmann	Hermann	m	Herr	02.02.68	01.01.96	Fritz-Reuter-Weg
THIED	Thiedemann	Klaus	m	Herr	12.04.41	01.09.70	Tallinner Str. 1
						01.06.99	

Bild 4.1: Tabellenausschnitt

Tabellen sind in Zeilen und Spalten organisiert. Jede Zeile stellt einen *Datensatz* dar, jede Spalte ein *Feld*. So sind im Tabellenausschnitt in Bild 4.1 folgende Felder zu sehen: *PersonalCode*, *Nachname*, *Vorname*, *Sex*, *Anrede*, *Geburtsdatum*, *Einstellungsdatum* und *Straße*. Ein Feld enthält daher so viele Werte, wie eine Tabelle Datensätze aufweist. Das Feld *Name* enthält z.B. (soweit im Bild wiedergegeben) die sechs Werte *Biedermann*, *Gersdorff*, *Lauser*, *Lejeune*, *Mahlmann* und *Thiedemann*. Obwohl die Werte eines Feldes sich im allgemeinen inhaltlich unterscheiden, sind sie doch von gleicher Art. So enthält das Feld *Name* lauter Textwerte, während das Feld *Geburtsdatum* nur Datumswerte aufweist. Für eine effiziente Datenbankverwaltung ist es erforderlich, daß bei der Definition einer Tabelle für jedes Feld sein Datentyp festgelegt wird, vgl. genauer Kapitel 8, *Tabellen: Grundlagen*.

4.1.3 Abfragen: Auswählen von Daten

Stellen Tabellen die Informationsgrundlage einer Datenbank dar, so sind *Abfragen* das wichtigste Instrument der Datenverwaltung. Mit einer Abfrage stellen Sie aus der Gesamtheit der Daten diejenigen Informationen zusammen, die Sie für einen bestimmten Zweck benötigen. So können Sie beispielsweise aus zwei Tabellen, von denen die eine Personaldaten und die andere die von den einzelnen Mitarbeitern betreuten Projekte enthält, alle Projekte zusammenstellen lassen, deren Auftragsvolumen im letzten Jahr 100.000 DM überschritt; vgl. dazu Bild 4.2 und 4.3. Mit einer Abfrage können Sie aber ebensogut eine einzelne Information (z.B. die Projektbezeich-

nung des Projekts mit der *ProjektNr 5*) oder eine aggregierte Information (z.B. das gesamte Auftragsvolumen der noch nicht abgeschlossenen Projekte) ermitteln lassen.

Bild 4.2: Tabelle »Personal« enthält Mitarbeiterangaben, Tabelle »Projekte« Angaben zu Projekten, die von den einzelnen Mitarbeitern betreut werden. Beide Tabellen sind über die Personalnummer miteinander verknüpft.

Abfrage1 : Auswahlabfrage

Nachname	Vorname	Projektbezeichnung	Auftragswert
Mahlmann	Hermann	Sportplatz Grundschule Heisterbach	1.000.000,00 DM
Mahlmann	Hermann	Neubau Einfamilenhaus Ottingen	370.000,00 DM
Gersdorff	Hubertus	Reparatur Stallungen Meierhof Ebbingen	150.000,00 DM
Thiedemann	Klaus	Anlage des Golfplatzes Peiner Hag	3.500.000,00 DM
Mahlmann	Hermann	Grundschule Hermannsdorf: Toilettenneubau	120.000,00 DM

Datensatz: 1 von 5

Bild 4.3: Die Abfrage stellt ausgewählte Mitarbeiter- und Projektangaben für alle Projekte zusammen, deren Auftragswert mehr als 100.000 DM beträgt.

Da die interne Datenorganisation (wie auch die äußere Form) von Abfrageergebnissen derjenigen in Tabellen entspricht, können Sie Abfragen annähernd so einsetzen wie Tabellen. Beispielsweise können Sie ein Formular oder einen Bericht ebensogut auf einer Abfrage basieren wie auf einer Tabelle.

Abfragen können in Access auf sehr bequeme Weise erstellt werden (vgl. genauer Kap. 10, *Abfragen: Grundlagen*). Sie können Abfragen zusammen mit der Datenbank speichern. Im allgemeinen enthält eine Datenbank eine

Vielzahl von gespeicherten Abfragen, die dann jeweils bei Bedarf aufgerufen oder dauerhaft als Datenherkunft von Formularen oder Berichten dienen. Obwohl das Abfrageergebnis große Mengen von Daten beinhalten kann, verbraucht eine gespeicherte Abfrage nur sehr wenig Speicherplatz auf der Festplatte. Insbesondere ist der benötigte Speicherplatz unabhängig von der im Abfrageergebnis vorhandenen Datenmenge. Dies ist darin begründet, daß Access nur die Abfrage*formulierung* speichert, nicht jedoch das Abfrage*ergebnis*. Die Folge dieses Konzepts ist natürlich, daß das Abfrageergebnis jedesmal neu ermittelt werden muß, wenn die Abfrage benötigt wird. Das kostet mehr oder weniger viel Zeit, hat jedoch den unschätzbaren Vorteil, daß Abfrageergebnisse stets dem neuesten Datenmaterial der Tabellen entsprechen.

4.1.4 Formulare: Übersichtliche Darstellung und Eingabe von Daten

Tabellen und Abfragen stellen die Daten in tabellarischer, standardisierter Form dar. Diese Art der Darstellung erweist sich als vorteilhaft, wenn Sie mehrere Datensätze gleichzeitig auf dem Bildschirm sehen und ggf. bearbeiten wollen. Der Benutzer hat allerdings kaum Möglichkeiten, die Art der Darstellung zu beeinflussen. In Formularen kann der Benutzer dagegen auf vielfältigste Weise bestimmen, wie die Daten auf dem Bildschirm dargeboten werden. Dies geschieht durch Anordnung und Gestaltung von Eingabefeldern (bei Access heißen diese *Textfelder*) und ihren *Bezeichnungen*, *Auswahlfeldern*, *Optionsgruppen*, *Unterformularen*, *Registern*, *grafischen Elementen* etc. Formulare erlauben eine derartige Gestaltung Ihrer Datenbank, daß sie ohne weiteres von Dritten, die im übrigen keinerlei Access-Kenntnisse zu besitzen brauchen, bearbeitet werden kann, lesend wie schreibend.

Bild 4.4: Formular mit Unterformular zur Bearbeitung der Daten aus den beiden Tabellen »Personal« und »Projekte« (vgl. oben, Bild 4.2)

Access stellt Ihnen mit Möglichkeiten wie *AutoFormular* oder *Formular-Assistent* Werkzeuge zur Verfügung, mit denen Sie auf sehr einfache und bequeme Weise verschiedene Standardformulare erzeugen können. Anspruchsvollere Formulare, die z.B. eine individuelle Gestaltung der Formularelemente (*Eingabefelder*, *Listenfelder* etc.) umfassen mögen, können und müssen Sie manuell erzeugen. Auch dies können Sie jedoch, da Sie konsequent die Maustechniken einsetzen können, auf recht einfache Weise bewerkstelligen, insbesondere dann, wenn Sie sich mit dem Formular-Assistenten ein bereits voll funktionierendes Formular erstellen lassen, das Sie dann anschließend nur noch Ihren Bedürfnissen entsprechend abändern.

4.1.5 Berichte: Drucken von Daten in übersichtlicher und gruppierter Form

Berichte haben eine große Ähnlichkeit mit Formularen, denn mit ihnen läßt sich die Wiedergabe von Daten in ähnlicher Weise gestalten wie mit diesen. Sie unterscheiden sich von Formularen vor allem in zwei Punkten: 1. Berichte sind nicht für den Bildschirm, sondern für die Druckausgabe gedacht. 2. In Berichten können Sie Daten in gruppierter Weise ausgeben und berechnen lassen, wie dies in Formularen nicht möglich ist. Ein gruppierter Bericht ist beispielsweise in Bild 4.5 wiedergegeben. Berichte setzen Sie besonders vorteilhaft ein, wenn Sie große Datenmengen in übersichtlicher Form auf dem Drucker ausgeben lassen wollen.

Bild 4.5: Gruppierter Bericht (Ausschnitt)

Auch Berichte können als *AutoBerichte* mit dem Berichts-Assistenten erzeugt werden. Bezüglich Standardisierung und individueller Gestaltung gilt hier das gleiche, was im letzten Punkt über Formulare angeführt wurde.

4.1.6 Makros: Benutzerdefinierte Programmsteuerung ohne Programmieren

Mit Makros können Sie die wichtigsten Access-Aktionen, die Sie sonst manuell mittels Ausführen von Menübefehlen oder durch Anklicken von Symbolschaltflächen veranlassen, automatisch ausführen lassen. Der Einsatz von Makros erweist sich als besonders wirkungsvoll, wenn Sie diese in Verbindung mit der Fähigkeit von Formularen und Berichten verbinden, ein Makro anläßlich eines bestimmten Formular- oder Berichts*ereignisses* zum Ausführen aufzurufen (zur Bedeutung von Ereignissen in Access vgl. in diesem Kapitel Punkt 4.1.8, *Ereignisse: Abläufe bedingt und automatisch steuern*). So können Sie beispielsweise ein Makro erstellen, welches das Formular *Bestellungen* öffnet und anzeigt. Dieses Makro können Sie mit dem Ereignis *Beim Klicken* einer Befehlsschaltfläche, die Sie im Formular *Kundenadressen* untergebracht haben, koppeln, so daß das Formular *Bestellungen* geöffnet wird, wenn Sie auf diese Befehlsschaltfläche klicken. Es lassen sich auch wesentlich komplexere automatische Abläufe und Ereigniszusammenhänge bilden, vgl. Bild 4.6.

Makros werden auf sehr bequeme Weise in tabellarisch angeordneten Makrofenstern erstellt. Auf diese Weise können Sie Programmabläufe automatisieren, ohne programmieren zu müssen.

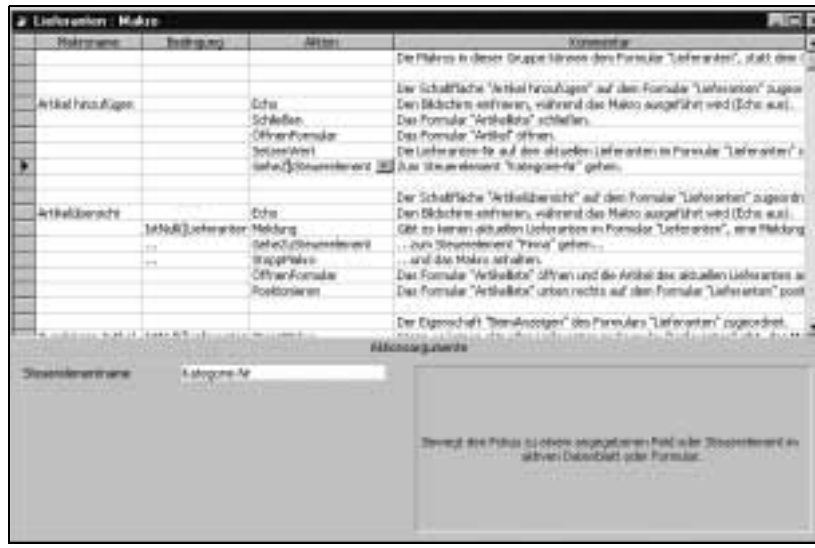


Bild 4.6: Das Makrofenster »Lieferanten« aus der Datenbank Nordwind.mdb enthält viele Makros, von denen hier die beiden Makros »Artikel hinzufügen« und »Artikelübersicht«, die jeweils eigenständig ausgeführt werden können, zu sehen sind. In die Kommentarspalte können ausführliche Kommentare geschrieben werden. In der Spalte »*Bedingung*« können Bedingungen angegeben werden, von deren Zutreffen die Ausführung einzelner Makroanweisungen abhängig ist.

4.1.7 Module: Programmieren mit VBA

Obwohl die mit Makros erzielbaren Automatisierungen sehr weitreichend sind, werden sie vielen Anwendern, vor allem den fortgeschrittenen, nicht ausreichen. Mit VBA (= *Visual Basic for Applications*) stellt Access 2000 eine mächtige Programmiersprache zur Verfügung, mit der Sie alle Möglichkeiten, die moderne Programmiersprachen üblicherweise haben, realisieren können. VBA ist die anwendungsübergreifende Programmiersprache von Microsoft, die in allen Office 2000-Anwendungen sowie in MS Visual Basic selbst gleichermaßen definiert und verfügbar ist. Aus diesem Grunde können Sie VBA-Code, den Sie in Access 2000 erstellt haben, prinzipiell auch in den anderen genannten Anwendungen verwenden. (Einschränkungen können sich ergeben, wenn Sie Objekte oder Funktionalitäten ansprechen, die nur in Access verfügbar sind; jedoch läßt sich auch dieser Fall lösen, vgl. genauer Kap. 33, *Integration von Access 2000 in andere Office 2000-Anwendungen*.) Der Einsatz von VBA ist besonders nützlich, wenn Sie selbst Funktionen schreiben wollen, weil Sie vielleicht eine spezielle Berechnung, die Sie immer wieder benötigen, nicht mit den Standardfunktionen von Access ausführen können. Darüber hinaus weist VBA viele Sprachelemente auf, die speziell dem Programmieren von Datenbankobjekten dienen. Die in VBA geschriebenen Funktionen und Prozeduren können genau so an Ereignisse der Formulare und Berichte gebunden werden wie die Makros. Daher läßt sich mit Hilfe von VBA praktisch jede Art von Automatisierung erzielen, die Sie wünschen.

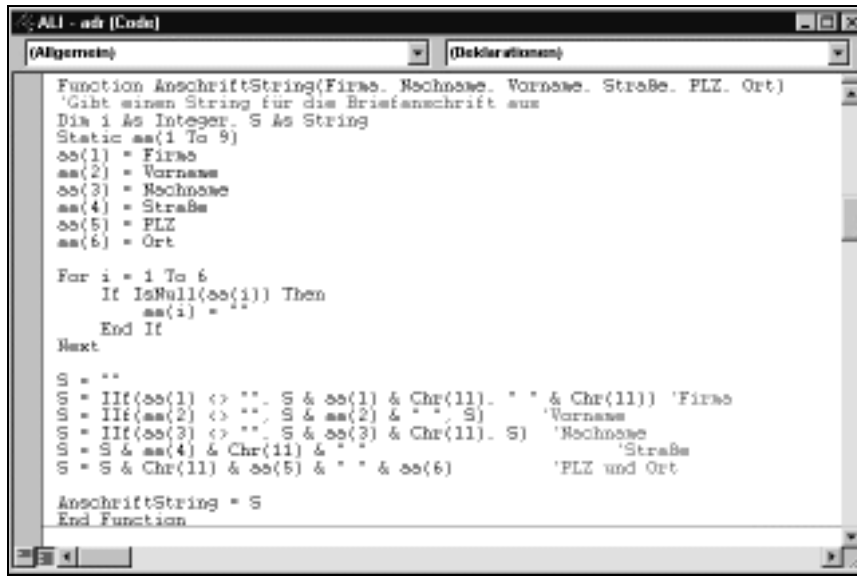


Bild 4.7: Modulfenster zum Bearbeiten von Access Basic-Prozeduren. Hier ist die benutzerdefinierte Funktion »AnschriftString« zu sehen, welche aus vier Adreßangaben einen String mit geeigneten Zeilenumbrüchen für das Anschriftenfenster im Brief erzeugt und ausgibt.

4.1.8 Ereignisse: Abläufe bedingt und automatisch steuern

In Access sind für die unterschiedlichsten Objekte *Ereignisse* definiert. Ein Ereignis ist eine Eigenschaft des jeweiligen Objekts. Für ein Formular beispielsweise existiert das Ereignis *Beim Anzeigen*. Dieses Ereignis tritt ein, wenn das Formular angezeigt wird, was z.B. beim Öffnen oder beim Einblenden eines bereits geöffneten Formulars zutrifft. Ein anderes Beispiel für ein Ereignis ist ein Textfeld, in das Sie in einem Formular Daten eingeben können. Für Textfelder beispielsweise ist u.a. das Ereignis *Vor Aktualisierung* definiert. Ereignisse sind im allgemeinen das Ergebnis einer Benutzeraktion. Die besondere Bedeutung von Ereignissen liegt darin, daß diesen Makros oder VBA-Prozeduren zugeordnet werden können, die ausgeführt werden, wenn das Ereignis eintritt. So können Sie beispielsweise dem eben angeführten Formularereignis *Beim Anzeigen* ein Makro oder eine VBA-Prozedur zuordnen, welche dafür sorgt, daß die Datenbasis des Formulars aktualisiert wird, weil in der Zwischenzeit möglicherweise Datenänderungen vorgekommen sind. Access kann auf eine große Zahl von Ereignissen reagieren. Für ein Formular z.B. sind 32 Ereignisse definiert, vgl. Bild 4.8, für ein Textfeld 15.

Bild 4.8: Ereignisse eines Formulars, angezeigt im Eigenschaftenfenster eines Formulars

4.1.9 Objekte: Bequeme Bezugnahme auf die Elemente des DBMS

Die verschiedenen selbständigen Elemente, mit denen das DBMS Access die Datenbankverwaltung organisiert, und über die Sie verfügen können (z.B. Tabellen, Formulare, Module, aber auch einzelne Formularelemente wie Textfeld, Listenfeld etc.), werden als *Objekte* bezeichnet. Jedes Objekt hat einen Namen, beispielsweise den Namen *Kundenadressen* für eine Tabelle oder *Mehrwertsteuer* für ein Textfeld in einem Formular. Sie beziehen sich auf ein Objekt, indem Sie seinen Namen anführen.

Ein Objekt hat im allgemeinen eine oder mehrere Eigenschaften. So besitzt ein Formular unter seinen vielen Eigenschaften auch die Eigenschaft *Datenherkunft*, für die Sie den Namen einer Tabelle oder Abfrage angeben können. Eine andere Formulareigenschaft ist z.B. *Datensatzmarkierer*, wofür Sie *Ja* oder *Nein* angeben können, je nachdem, ob im betreffenden Formular der Datensatzmarkierer (eine Leiste am linken Formularrand) angezeigt werden soll oder nicht. Die Eigenschaften der verschiedenen Objekte lassen sich im allgemeinen sehr leicht zuweisen (bzw. im Sprachgebrauch von Access: einstellen), weil Sie dafür das sogen. Eigenschaftenfenster benutzen können, welches genau die für das jeweils markierte Objekt verfügbaren Eigenschaften anzeigt und, wo immer dies möglich und sinnvoll ist, die möglichen Einstellungswerte auflistet, aus denen Sie dann wählen können.

Die konsequente Organisierung der Access-Elemente als mit Namen versehene und einstellbare Eigenschaften besitzende Objekte macht die Struktur selbst so komplexer Objekte wie Formulare, Tabellen oder Berichte transparent und erhöht die Verständlichkeit von Formeln und anderen Ausdrücken, in denen Access-Elemente angeführt und verarbeitet werden, ungemein.

Objekte lassen sich im allgemeinen mit den üblichen Windows-Techniken kopieren. Das gilt auch für ganze Tabellen, Formulare etc.:

- Markieren Sie das Objekt, klicken Sie auf die links nebenstehende Symbol-Schaltfläche *Kopieren* oder wählen Sie den Befehl *Kopieren* aus dem Menü *Bearbeiten* (bzw. Tastenkombination Strg+C).

- Klicken Sie anschließend auf die Symbol-Schaltfläche *Einfügen* (vgl. links nebenstehend) oder wählen Sie den Befehl *Einfügen* aus dem Menü *Bearbeiten* (bzw. Tastenkombination Strg+V). Je nach Objekt werden Sie beim Einfügen aufgefordert, weitere Angaben (z.B. einen neuen Objektnamen anführen) zu machen.

Alle Objekte einer Datenbank werden in einer Datei, der Datenbankdatei, gespeichert. Daher können Sie sicher sein, daß sämtliche Elemente, die Sie für eine bestimmte Datenbank erstellt haben, mit dem Öffnen der entsprechenden Datenbankdatei zur Verfügung stehen.

Import- und Exportmöglichkeiten von Tabellen machen es aber auch möglich, diese auszulagern. Weiterhin können Sie Tabellen aus anderen Anwendungen, auch fremder Formate (z.B. Btrieve, Paradox, dBase, Excel, sogar

Textformat), in eine Access-Datenbank einbinden. Eingebundene Tabellen werden wie eigene verwaltet, so daß sie gelesen, aber auch beschrieben werden.

Wenn Sie mit VBA programmieren, stehen für die meisten Objekte auch *Methoden* zur Verfügung, die deren Manipulation einfach macht, denn VBA ist eine *objektorientierte Programmiersprache*. Auf diese Weise wird auch erreicht, daß zwischen der »normalen« Ausführung von Access, wie sie jeder Benutzer ohne VBA-Prozeduren kennt, und der Steuerung mit Hilfe von VBA-Prozeduren kein konzeptioneller Unterschied besteht, so daß kein prinzipielles Umdenken erforderlich ist, wenn Sie zwischen den beiden Ebenen wechseln.

4.1.10 Datensätze: Automatisches Speichern

Wenn Sie Daten in ein Formular, eine Tabelle oder in das Datenblatt eines Abfrageergebnisses eingeben oder vorhandene Daten dort bearbeiten, speichert Access die Veränderung automatisch, sobald es sie als abgeschlossen erkennt. Wenn Sie beispielsweise einzelne Felder eines Datensatzes in einem Formular bearbeiten, werden die Veränderungen automatisch und ohne Meldung gespeichert, sobald Sie zu einem anderen Datensatz blättern. Diese Verfahrensweise hat den großen Vorteil, daß Sie sich als Benutzer (und ggf. auch als Programmierer) um das Speichern von Daten im allgemeinen überhaupt nicht zu kümmern brauchen. In einer Mehrbenutzerumgebung sind dadurch veränderte Datensätze sofort für andere Benutzer aktualisiert. Ferner haben Sie eine relativ große Datensicherheit gegenüber Systemabstürzen, weil ungesicherte Datenänderungen praktisch nicht vorkommen.

Beim Testen einer bereits mit Daten versehenen Datenbank müssen Sie andererseits vorsichtig vorgehen, denn anders, als Sie es vielleicht bisher von anderen Datenbankprogrammen gewohnt sind, werden auch Datenänderungen, die Sie lediglich zu Testzwecken vornehmen wollen, sofort permanent gemacht.

4.1.11 Die Jet-Datenbank-Engine

Wie die vorangehenden Punkte dieses Kapitels gezeigt haben, handelt es sich bei Access um eine Anwendung, die es Ihnen ermöglicht, Daten beinahe beliebig komplexer Art komfortabel und sicher zu organisieren und zu verwalten. Der Komfort wird dabei vor allem durch Werkzeuge wie Formulare, Datenblattansichten oder Berichte, die mit Mitteln der grafischen Benutzeroberfläche (vor allem Einsatz der Maus) gesteuert werden, gewährleistet. Für die Datensicherheit sorgt dagegen ein Programm-Modul, das im Hintergrund arbeitet: die *Jet-Datenbank-Engine*[1]. Aufgabe dieser »Datenbank-Maschine« ist es, eine relationale Datenbank nach allen Regeln der Kunst – und deren grundlegende sind mittlerweile international quasi standardisiert – zu verwalten. Dazu gehören vor allem Regeln der *Definition*, *Organisation*, *Manipulation* und *Sicherheit* der Daten. Diese Regeln müssen in allen

relationalen Datenbank-Managementsystemen in gleicher Weise aktiv und passiv durchgesetzt werden. Das Programm-Modul *Jet-Datenbank-Engine* ist konsequent in das Konzept der Objekt-Organisation integriert: Microsoft hat das Objekt *Data Access Object* – im allgemeinen und auch hier im folgenden als *DAO* bezeichnet – entwickelt, welches die *Jet-Datenbank-Engine* als ein Objekt enthält. Genau genommen ist daher nicht die *Jet-Datenbank-Engine*, sondern das Objekt *DAO* ein eigenes Programm-Modul. Dieses Modul wird eigenständig auch mit den anderen Anwendungen des Office-Pakets sowie mit Visual Basic ausgeliefert. Daher ist es auch möglich, über die volle Funktionalität des Relationalen Datenbank-Mangamentsystems z.B. von Excel aus zu verfügen. Auf diese Weise werden Office-Anwendungen hinsichtlich der Datenverwaltung integriert.

4.1.12 Assistenten und Auto-Objekte

Zur Vereinfachung häufig vorkommender komplexer Aufgaben stellt Access mehrere Assistenten und die Möglichkeit automatisch erzeugbarer Objekte zur Verfügung. Die Assistenten führen Sie Schritt für Schritt weiter und bieten Ihnen dabei geeignete Auswahlalternativen und Hinweise an. Wenn Sie alle Schritte ausgeführt haben, hat der Assistent eine fertige Tabelle, ein fertiges Formular etc. erstellt. Die wichtigsten Assistenten und Auto-Objekte sind die folgenden:

- *Tabellen*. Mit dem Tabellen-Assistenten können Sie eine neue Tabelle definieren und dabei aus einer Vielzahl von Beispieltabellen und praxisgerechten Beispielfeldern auswählen.
- *Formular*. Mit dem Formular-Assistenten können Sie verschiedene Formulare – einschließlich Haupt- mit Unterformular – in einer standardisierten Form erzeugen. Außer dem Formular-Assistenten können Sie auch verschiedene Formen von *AutoFormular* wählen, dann erstellt der Formular-Assistent für die Tabelle oder Abfrage, die (oder deren Symbol im Datenbankfenster) gerade aktiv ist, ein neues Formular vollständig automatisch. In vielen Fällen genügt ein derartiges Formular den praktischen Ansprüchen. Darüber hinaus stehen Ihnen ein Diagramm-Assistent sowie ein Assistent zum Erzeugen von Pivot-Tabellen zur Verfügung; für letztere muß auf Ihrem PC MS Excel verfügbar sein.

1. Mit Access 2000 wird es möglich, die Tabellen und Abfragen statt von der Jet-Datenbank-Engine vom Microsoft SQL-Server verwalten zu lassen. Diese Arbeitsweise, von der ich hier abstrahiere, wird ausführlich weiter unten beschrieben, vgl. Kap. 37, *Access-Projekte als Client/Server-Datenbank: Grundlagen*, sowie Kap. 38, *Access-Projekt erstellen*.

Bild 4.9: Der Tabellen-Assistent zum Erstellen einer neuen Tabelle läßt Sie aus einer Vielzahl von Tabellen und jeweils dazugehörigen Tabellenfeldern mit wichtigen Eigenschaften wählen.

Bild 4.10: Mit den Formular-Assistenten können Sie mehrere verschiedene Formulararten erzeugen.

- *Bericht*. Mit dem Berichts-Assistenten können Sie verschiedene Berichte – einschließlich gruppierte Berichte – in einer standardisierten Form erzeugen. Wenn Sie einen *AutoBericht* wählen, erstellt der Berichts-Assistent für die Tabelle oder Abfrage, die (oder deren Symbol im Datenbankfenster) gerade aktiv ist, einen neuen Bericht vollständig automatisch. Hier gilt nach meiner persönlichen Einschätzung weniger als für die AutoFormulare, daß ein derartiger Bericht den praktischen Ansprüchen genügt, so daß Sie einen vollständig automatisch erzeugten Bericht vermutlich öfter überarbeiten müssen.
- *Abfrage*. Abfrage-Assistenten unterstützen Sie beim Erstellen von vier verschiedenen Arten komplexerer Abfragen.

Bild 4.11: Mit den Berichts-Assistenten können Sie verschiedene Berichtsarten erzeugen.

Bild 4.12: Sie können diese vier verschiedenen Abfragearten vom Assistenten erzeugen lassen.

Bild 4.13: Erstes Dialogfeld des Steuerelement-Assistenten für ein Kombinationsfeld

- *Steuerelement*. Wenn Sie in ein Formular oder einen Bericht ein neues Steuerelement (z.B. ein Kombinationsfeld oder eine Befehlsschaltfläche) einfügen, können Sie sich von einem für das jeweilige Steuerelement spezifischen Assistenten unterstützen lassen, die verschiedenen Steuerelementeigenschaften einfach und sinnvoll auszufüllen.

Die Assistenten insgesamt

Die folgende Übersicht zeigt Ihnen insgesamt, für welche Aufgaben Assistenten zur Verfügung stehen.

Assistent	Beschreibung
AutoWähler-Assistent	Ermöglicht Ihnen das Festlegen von Modeminformationen, wenn Sie in der Formularansicht auf eine AutoWähler-Schaltfläche klicken.
AutoFormular-Assistent	Erstellt ein Formular automatisch.
AutoFormat-Assistent	Wendet ein vordefiniertes Format auf ein Formular oder einen Bericht an.
AutoSeiten-Assistent	Erstellt eine Datenzugriffsseite automatisch.
AutoBerichts-Assistent	Erstellt einen Bericht automatisch.
Diagramm-Assistent	Fügt ein Diagramm zu einem Formular oder Bericht hinzu, das auf den Daten in einer Tabelle oder Abfrage basiert.
Kombinationsfeld-Assistent	Erstellt in einem Formular ein Kombinationsfeld-Steuerelement.
Befehlsschaltflächen-Assistent	Erstellt in einem Formular ein Befehlsschaltflächen-Steuerelement.
Assistent zur Behebung von Replikationskonflikten	Löst Konflikte, die beim Synchronisieren zwischen replizierten Datenbanken auftreten.
Abfrage-Assistent für Kreuztabellen	Erstellt eine Abfrage, die Daten in einem kompakten Format, das einer Kalkulationstabelle ähnelt, zusammenfaßt.
Assistent zur Datenbankaufteilung	Teilt Datenbanken in eine Backend- und eine Frontend-Datenbank, so daß ein oder mehrere Benutzer lokale Kopien der Frontend-Datenbank einsetzen können, die mit den Daten (in der Backend-Datenbank) auf einem Server verbunden sind.
Datenbank-Assistent	Erstellt eine völlig neue Datenbank für eine Vielzahl unterschiedlicher Einsatzmöglichkeiten. Dabei stehen 2210 vordefinierte Modelle zur Verfügung.

Assistent	Beschreibung
Dokumentierer	Erstellt einen Access-Bericht, in dem die Entwurfsmerkmale von Datenbankobjekten angezeigt werden.
Textexport-Assistent	Exportiert Daten in eine Textdatei.
Abfrage-Assistent zur Duplikatsuche	Erstellt eine Abfrage zum Abrufen von Datensätzen, die Duplikatwerte aufweisen und sich in einer einzelnen Tabelle oder Abfrage befinden.
Abfrage-Assistent zur Inkonsistenzsuche	Erstellt eine Abfrage zum Abrufen von Datensätzen in einer Tabelle, für die es in einer verknüpften Tabelle keine Entsprechung gibt.
Formular-Assistent	Erstellt ein neues Formular.
Exchange/Outlook-Import-Assistent	Importiert einen Exchange- oder Outlook-Ordner in eine Tabelle einer Microsoft Access-Datenbank.
HTML-Import-Assistent	Importiert HTML-Tabellen und -Listen aus dem Internet oder einem Intranet in eine Microsoft Access-Tabelle.
Import-Assistent für Kalkulationstabellen	Importiert eine Microsoft Excel- oder eine andere Tabellenkalkulationstabelle in eine Microsoft Access-Tabelle.
Textimport-Assistent	Importiert eine Textdatei in eine Microsoft Access-Tabelle.
Eingabeformat-Assistent	Erstellt ein Eingabeformat für ein Feld, das Sie in einer Tabelle auswählen.
Etiketten-Assistent	Erstellt Adressetiketten in Standardgrößen und benutzerdefinierten Größen.
Exchange/Outlook-Verknüpfungs-Assistent	Verknüpft einen Exchange- oder Outlook-Ordner mit einer Tabelle in einer Microsoft Access-Datenbank.
HTML-Verknüpfungs-Assistent	Verknüpft eine HTML-Tabelle oder -Liste auf dem Internet oder einem Intranet in einer Microsoft Access-Tabelle.
Verknüpfungs-Assistent für Kalkulationstabellen	Verknüpft Tabellenkalkulationsdaten in einer Microsoft Access-Tabelle.
Textverknüpfungs-Assistent	Verknüpft eine Textdatei in einer Microsoft Access-Tabelle.
Tabellenverknüpfungs-Manager	Verwaltet Tabellenverknüpfungen zwischen Datenbanken.

Assistent	Beschreibung
Listenfeld-Assistent	Erstellt in einem Formular ein Listenfeld-Steuerelement.
Nachschlage-Assistent	Erstellt in einer Tabelle eine Nachschlage-Spalte, in der eine Liste von Werten angezeigt wird, aus der der Benutzer auswählen kann.
Assistent zur Konvertierung eines Makros in ein Modul	Konvertiert Makros in Visual Basic-Code.
Microsoft SQL Server Datenbank-Assistent	Erstellt eine neue Microsoft SQL Server-Datenbank, mit der ein neues Microsoft Access-Projekt verbunden ist.
Microsoft Word-Seriendruck-Assistent	Verwaltet Seriendruckoperationen unter Verwendung von in Microsoft Word gespeicherten Briefen und in Microsoft Access gespeicherten Adressen.
Optionsgruppen-Assistent	Erstellt in einem Formular eine Gruppe von Optionsschaltflächen.
Seitenkombinationsfeld-Assistent	Erstellt ein Dropdownfeld auf einer Datenzugriffsseite.
Seitenbefehlsschaltflächen-Assistent	Erstellt eine Befehlsschaltfläche auf einer Datenzugriffsseite.
Seitenlistenfeld-Assistent	Erstellt ein Listenfeld auf einer Datenzugriffsseite.
Seiten-Assistent	Erstellt eine neue Datenzugriffsseite.
Teilreplikations-Assistent	Erstellt oder ändert ein Teilreplikat einer Replikatsdatenbank.
Assistent zur Leistungsanalyse	Analysiert die Leistung einer Datenbank und macht Vorschläge zur Verbesserung der Leistung.
PivotTable-Assistent	Erstellt in einem Microsoft Access-Formular eine Microsoft Excel-Pivot-Tabelle.
Assistent zum Drucken von Beziehungen	Erstellt einen Bericht, in dem die Beziehungen in einer Microsoft Access-Datenbank mittels eines Diagramms dargestellt werden.
Berichts-Assistent	Erstellt einen Bericht, der auf einer Tabelle oder Abfrage basiert.
Auswahlabfrage-Assistent	Erstellt eine Auswahlabfrage, die auf den von Ihnen ausgewählten Feldern basiert.

Assistent	Beschreibung
Feldverknüpfungs-Assistent für Unterformulare/-berichte	Verknüpft Felder in einem Hauptformular und einem Unterformular oder in einem Hauptbericht und einem Unterbericht.
Unterformular-/Unterberichts-Assistent	Erstellt in einem Formular oder Bericht ein neues Unterformular oder einen Unterbericht.
Übersichts-Manager	Erstellt und verwaltet Übersichtsformulare für Anwendungen.
Tabellenanalyse-Assistent	Teilt eine Tabelle mit vielen Duplikatwerten in verknüpfte Tabellen auf, die effektiver gespeichert werden können.
Tabellen-Assistent	Erstellt eine neue Tabelle.
Upsizing-Assistent	Paßt die Größe einer Microsoft Access-Datenbank an eine Microsoft SQL Server-Datenbank an.
Benutzer-Datensicherheits-Assistent	Erstellt aus einer vorhandenen Datenbank eine neue, verschlüsselte Datenbank mit geregeltem Benutzerzugriff. Reguliert den Benutzerzugriff auf die aktuelle Datenbank und erstellt eine ungesicherte Sicherungskopie der Datenbank.

4.1.13 Generatoren

Für viele – meist kleinere, wenngleich manchmal schwierige – Aufgaben stellt Access ein Hilfswerkzeug zur Verfügung, das *Generator* genannt wird. Beispielsweise gibt es einen Ausdrucks-Generator, einen Farb-Generator, einen Feld-Generator etc., um Ausdrücke zu erzeugen, Farben zuzuweisen oder Eigenschaften von Tabellenfeldern einzustellen.

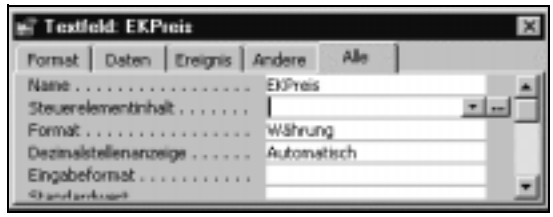


Bild 4.14: Die Generator-Schaltfläche mit den drei Punkten erscheint, wenn ein Bearbeitungsfeld aktiv ist, zu dem ein Generator geöffnet werden kann. In der wiedergegebenen Situation ist das Bearbeitungsfeld für die Eigenschaft »Steuerelementinhalt« eines Textfeldes aktiv.

Einen Generator rufen Sie allgemein auf, indem Sie im Eigenschaftenfenster auf die kleine Schaltfläche mit drei Punkten, die kontextbezogen eingeblendet wird, klicken. Wenn Sie beispielsweise eine Eigenschaft eines Steuerelements in einem Formular bearbeiten, erscheint am rechten Rand des Eigenschaften-Bearbeitungsfeldes die Generator-Schaltfläche. In Bild 4.14 sehen Sie diese Schaltfläche am rechten Rand des Bearbeitungsfeldes für die Eigenschaft *Steuerelementinhalt*, weil dieses gerade bearbeitet wird. Wenn Sie in diesem Falle auf die Generator-Schaltfläche klicken, wird der Ausdrucks-Generator geöffnet, dessen Dialogfeld Sie in Bild 4.15 sehen können. Der dort erkennbare Ausdruck

```
= Formulare![Artikel]![Einzelpreis] * Formulare![Bestellungen
Unterformular]![Rabatt]
```

wurde erzeugt, indem nacheinander verschiedene Elemente aus dem unteren Teil des Dialogfeldes ausgewählt wurden.

Bild 4.15: Dialogfeld des Ausdrucks-Generators

Die Generatoren insgesamt

Die folgende Übersicht zeigt Ihnen insgesamt, für welche Aufgaben Generatoren zur Verfügung stehen.

Generator	Beschreibung
Farb-Generator	Zeigt eine Farbpalette zum Erstellen benutzerdefinierter Farben an.
Ausdrucks-Generator	Erstellt Ausdrücke für Makros, Abfragen und Eigenschaftenfenster.
Feld-Generator	Erstellt Felder in Tabellen.

Generator	Beschreibung
Generator für ODBC-Verbindungszeichenfolgen	Erstellt die korrekte Syntax für eine Verbindung zu einer ODBC-Datenbank.
Bild-Generator	Erstellt Bitmap-Bilder für Formulare und Berichte.
Abfrage-Generator	Erstellt die korrekte Syntax für eine Abfrage.

4.1.14 Beispieldatenbanken

Im Lieferumfang von Access 2000 eingeschlossen sind mehrere Beispieldatenbanken. Die größte Bedeutung hat dabei die Datenbank *Nordwind.mdb*[2], denn sie enthält in den verschiedenen Tabellen, Formularen, Abfragen und Berichten viele Lösungsmuster, von denen Sie beim Erstellen einer eigenen Datenbank lernen können. Viele der Lösungen lassen sich direkt auf eigene Anwendungen übertragen. Die Online-Hilfe verweist an sehr vielen Stellen auf Beispiele in der Datenbank *Nordwind*, so daß diese praktisch Bestandteil der Erklärungstexte ist. Auch dieses Buch zieht vielfach Beispiele aus *Nordwind* zur Erklärung heran. Falls Sie diese Datenbank nicht bereits mit der Erstinstallation von Access auf die Festplatte kopiert haben, sollten Sie dies auf alle Fälle nachholen, indem Sie das Setup-Programm erneut starten und dabei die entsprechende Option wählen. Diese wie auch die anderen von Access mitgelieferten Beispieldatenbanken werden im Ordner *Beispiel* installiert, einem Unterordner zum Office- oder Access-Programmordner.

Bei den anderen mitgelieferten Beispieldatenbanken handelt es sich um *Adressbuch.mdb*, *Kontaktverwaltung.mdb* und *Haushaltsinventar.mdb*. Diese Datenbanken sind laut Access-Hilfe dazu gedacht, unmittelbar von Ihnen für die entsprechenden Zwecke verwendet zu werden, also für eine eigene Adress-, Kontakt- und Inventarverwaltung eingesetzt zu werden. Gleichwohl können Sie auch aus diesen Anwendungen Anregungen für eigene Datenbankentwürfe entnehmen.

4.1.15 Online-Hilfe

Die Online-Hilfe von Access ist außergewöhnlich umfangreich. Sie enthält beispielsweise das gesamte Sprachverzeichnis (d.h. Erklärungen zu allen Funktionen, Eigenschaften, Aktionen, Ereignissen, Methoden und Objekten von VBA) und wesentliche Teile der übrigen Handbuch-Informationen. Das

2. Die Nordwind-Datenbank wird unter dem Namen *NordwindCS* auch noch als Client/Server-Beispieldatenbank angeboten. *NordwindCS* enthält ein SQL-Skript, das die Nordwind-Datenbank im Microsoft SQL Server erstellt, sowie ein Microsoft Access-Projekt, das Sie mit der Nordwind-Datenbank verbindet.

Hilfe-System ist in der für Windows-Programme üblichen Weise organisiert. Es enthält daher auch die Möglichkeit, nach bestimmten Begriffen und Themen suchen zu lassen. Da die Suchbegriffe äußerst detailliert sind, werden Sie relativ selten enttäuscht sein, wenn Sie nach einem bestimmten Begriff suchen lassen. Ein intensiver Gebrauch der Access-Hilfe kann nur empfohlen werden.

4.2 Oberfläche

4.2.1 Symbolleisten, Menüleisten, Kontextmenüs, Tastenkombinationen

Wie bei modernen Windows-Programmen von Microsoft üblich, erfolgt die Steuerung des Systems mittels Menüs in Menüleisten, Kontextmenüs (aufschlagbar durch Klicken mit der *rechten* Maustaste auf das zu bearbeitende Objekt, das zu bearbeitende Feld etc.), Schaltflächen in Symbolleisten sowie Drücken von Tastenkombinationen. Darüber hinaus gibt es aber eine für Access spezifische Steuerung über das Datenbankfenster, worauf im nächsten Abschnitt eingegangen wird.

Der Umgang mit den Steuerungsinstrumenten Menü, Symbol-Schaltfläche und Tastenkombination wird hier als prinzipiell bekannt vorausgesetzt. Im übrigen können Sie sich in der Online-Hilfe leicht auf die folgende Weise über die Bedeutung eines Menübefehls oder einer Symbol-Schaltfläche informieren:

- Drücken Sie die Tastenkombination ⇧+F1; dann wandelt sich der Mauszeiger zu einem Fragezeichen-Mauszeiger, vgl. links nebenstehend.
- Wählen Sie mit dem Fragezeichen-Mauszeiger einen Befehl aus, oder klikken Sie auf die erklärungsbedürftige Symbol-Schaltfläche. Dann erhalten Sie einen Hilfetext zu dem Befehl bzw. zu der Symbol-Schaltfläche.

Über die verschiedenen Tastenkombinationen informiert die Online-Hilfe unter dem Stichwort *Tastenkombinationen*.

Eine Kurzerklärung zu Symbol-Schaltflächen bekommen Sie auf die folgende Weise:

- Fahren Sie den Mauszeiger auf die Symbol-Schaltfläche, und lassen sie ihn dort ggf. wenige Sekunden stehen; dann erscheint unmittelbar neben der Symbol-Schaltfläche ein kurzer Hilfetext, der die Schaltfläche erklärt.

4.2.2 Datenbankfenster

Wenn Sie in Access eine neue Datenbank öffnen, erhalten Sie den in Bild 4.16 wiedergegebenen Bildschirm. Sein beherrschendes Element ist das Datenbankfenster, das im wiedergegebenen Falle die Überschrift *NORDWIND: Datenbank* trägt. Das Datenbankfenster dient dazu, einzelne Tabel-

len, Abfragen, Formulare, Berichte, Seiten, Makros oder Module auszuwählen, um sie zu betrachten oder Änderungen an ihnen vorzunehmen. Darüber hinaus können Sie aus dem Datenbankfenster heraus neue Objekte erstellen, Objekte bestehenden Gruppen zuordnen und neue Gruppen definieren.

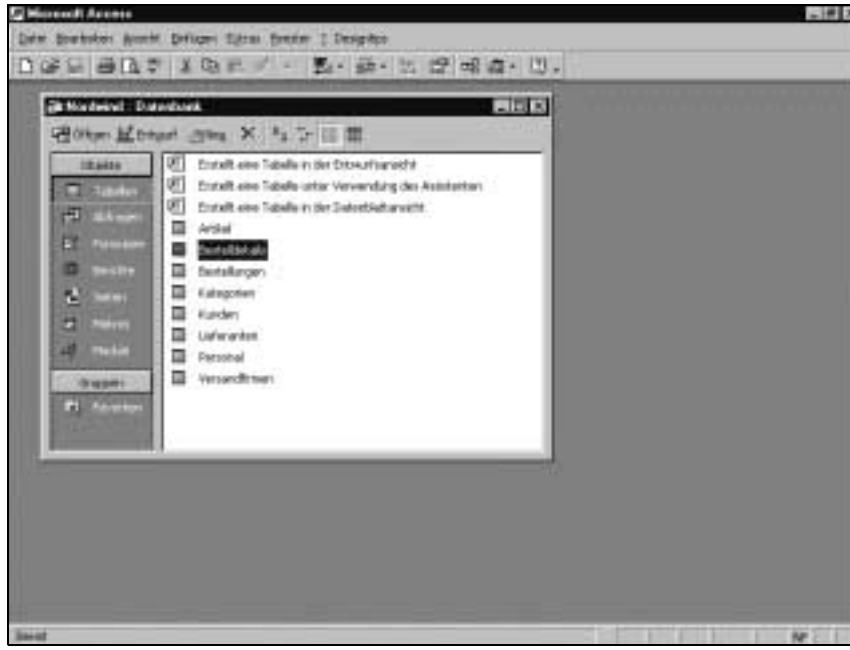


Bild 4.16: Eröffnungsbildschirm unmittelbar nach dem Öffnen der Datenbank Nordwind.mdb

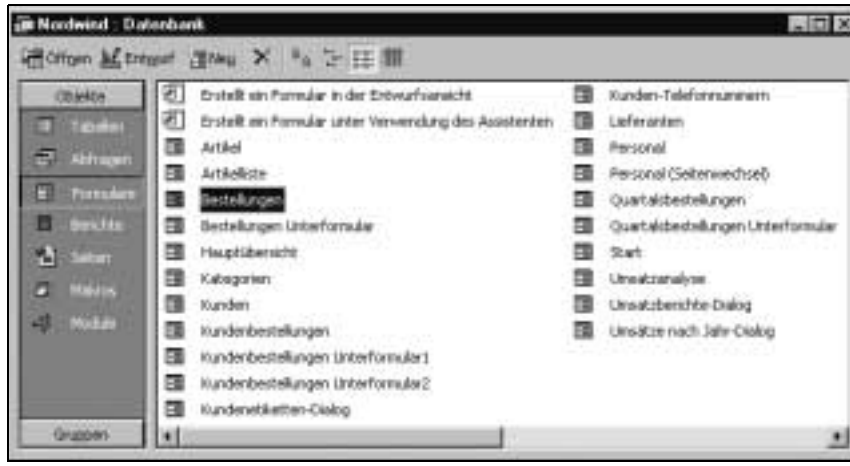


Bild 4.17: Datenbankfenster mit aktivierten Objekten »Formulare«

Um ein einzelnes Objekt (Tabelle, Abfrage etc.) zu öffnen, gehen Sie folgendermaßen vor:

- Klicken Sie auf eines der Symbole in der am linken Rand des Datenbankfensters angeordneten Objektsymbolleiste. Dann erscheint die Liste mit den für das angeklickte Objekt-Symbol vorhandenen einzelnen Objekten. In Bild 4.16 sehen Sie die Liste der in Nordwind.mdb vorhandenen Tabellen mit den Namen *Artikel*, *Bestelldetails*, *Bestellungen* ... *Versandfirmen*. Sie können auch erkennen, daß das Tabellen-Symbol gedrückt ist. Wenn Sie beispielsweise ein Formular öffnen wollen, müssen Sie zuvor im Datenbankfenster auf das Formular-Symbol (oder ein Gruppensymbol, dem Formulare zugeordnet sind) klicken; dann erhalten Sie die Liste der verfügbaren Formulare. Eine solche Situation ist in Bild 4.17 dargestellt.
- Markieren Sie das gewünschte Objekt (Tabelle, Abfrage, Formular ...) in der Liste im Datenbankfenster, indem Sie darauf klicken. Sie können auch den ersten Buchstaben des Objektnamens eingeben, dann springt die Markierung zum ersten Objekt, dessen Name mit dem getippten Buchstaben beginnt.
- Klicken Sie auf eine der Schaltflächen *Öffnen* oder *Entwurf*, je nachdem, für welchen Bearbeitungszweck Sie das Objekt öffnen wollen (vgl. dazu ausführlich die weiteren Kapitel). Wenn Sie auf ein Objekt-Symbol doppelklicken, ist dies gleichbedeutend mit Markieren und anschließendem Klicken auf die Schaltfläche *Öffnen*.

F11 Wenn Sie mehrere Objekte (Tabellen, Abfragen, Formulare, Berichte ...) geöffnet haben, werden die jeweils anderen vom aktiven Objektfenster verdeckt sein, Wenn Sie z.B. ein Formular geöffnet haben, wird dessen Fenster im allgemeinen die Fenster der anderen geöffneten Objekte (z.B. anderer Formulare oder Tabellen etc.) verdecken. Um in dieser Situation ein anderes bereits geöffnetes Objekt in den Vordergrund zu bringen, schlagen Sie das Menü *Fenster* auf und wählen in dessen Fensterliste das gewünschte durch Klicken aus. Zu den verdeckten Fenstern zählt oft auch das Datenbankfenster. Sie bringen es ebenfalls mit Hilfe der Fensterliste des Menüs *Fenster* in den Vordergrund. Die Tastenkombination, um das Datenbankfenster, das Sie sehr oft benötigen werden, in den Vordergrund zu bekommen, lautet: Taste F11.

Wenn Sie das Datenbankfenster schließen, wird mit ihm auch die geöffnete Datenbank geschlossen. Falls Sie das Datenbankfenster stört, können Sie es jedoch, wie jedes andere Fenster, zum Symbol verkleinern oder ausblenden.

4.2.3 Navigationsschaltflächen

Neben mehreren Menübefehlen (beispielsweise *Suchen* oder *Gehe zu* im Menü *Bearbeiten*) bietet Access vor allem die bequem handhabbaren Navigationsschaltflächen an, um zwischen den Datensätzen eines Formulars, einer Tabelle oder einer Abfrage zu blättern. Die Navigationsschaltflächen

befinden sich jeweils im linken Teil der waagerechten Bildlaufleiste am unteren Rand des entsprechenden Fensterausschnitts. Ihre Bedeutung erkennen Sie in Bild 4.18.

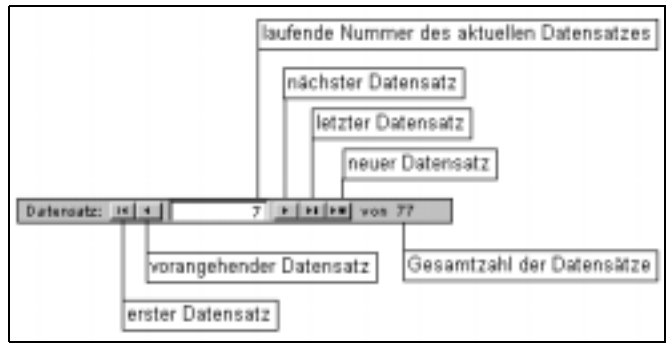


Bild 4.18: Navigationsschaltflächen zum Bewegen zwischen Datensätzen. Sie befinden sich jeweils im linken Teil der waagerechten Bildlaufleiste am unteren Rand des entsprechenden Fensterausschnitts.

4.2.4 Dialogfeld *Zoom*

Eingabefelder in Tabellen, Abfragen, Formularen etc. – deren Entwurfsansichten eingeschlossen – werden aus Gründen der Übersicht oftmals nicht sehr breit dargestellt. Dann können Sie eine darin enthaltene lange Zeichenfolge (z.B. einen Text oder einen Ausdruck) nur lesen, wenn Sie diese mit einer der Pfeiltasten rollen. Das ist sehr unbequem. Entsprechendes gilt für die Eingabe längerer Zeichenfolgen. In diesen Fällen empfiehlt es sich, das Dialogfeld *Zoom* zu öffnen, um die Zeichenfolge des Eingabefeldes zu lesen oder zu bearbeiten:

- Klicken Sie in das zu bearbeitende Eingabe- oder Bearbeitungsfeld, um es zu aktivieren.
- Öffnen Sie das Dialogfeld *Zoom*, indem Sie die Tastenkombination ⇧+ F2 drücken.
- Lesen Sie die Zeichenfolge, bearbeiten Sie diese oder geben Siesie neu ein. Bestätigen Sie mit OK oder brechen Sie ab, um das Dialogfeld *Zoom* zu schließen.

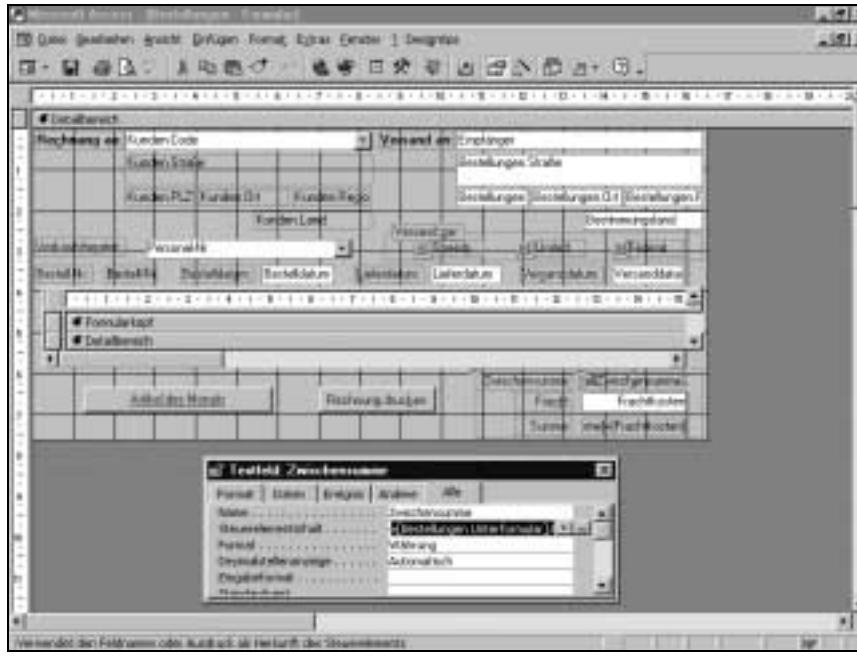


Bild 4.19: Formular »Bestellungen« der Datenbank Nordwind.mdb in der Entwurfsansicht. Das Textfeld »Zwischensumme« ist markiert. Im eingeblendeten Eigenschaftenfenster ist das Bearbeitungsfeld für die Eigenschaft »Steuerelementinhalt« aktiv. Der darin enthaltene Ausdruck ist länger, als in der aktuellen Breite darstellbar. Daher empfiehlt es sich, die Bearbeitung mit dem Dialogfeld »Zoom« vorzunehmen. Dieses ist in Bild 4.20 wiedergegeben.

Bild 4.20: Dialogfeld »Zoom«, geöffnet für das in Bild 4.19 als aktiv erkennbare Bearbeitungsfeld für die Eigenschaft »Steuerelementinhalt«

4.2.5 Beispiel: Ein Spaziergang durch die Datenbank Nordwind.mdb.

Um die Access-Oberfläche auch praktisch kennenzulernen, sollten Sie sich einmal verschiedene Formulare, Tabellen und Abfragen der Beispieldatenbank Nordwind.mdb ansehen und darin blättern. Wenn Sie die folgenden Punkte nachvollziehen, wird Ihnen die Orientierung in der Access-Oberfläche leicht fallen.

Hinweis. Die Datenbank *Nordwind.mdb* ist Teil des Access 2000- bzw. Office 2000-Pakets. Wenn Sie Access vom Office 2000-Paket mit Standardinstallation installiert haben, ist auch Nordwind.mdb mit installiert worden. Wenn vorhanden, sollte sich die Datei Nordwind.mdb im Ordner *\Office\Samples* befinden. Falls Sie die Datei *Nordwind.mdb* dort nicht finden, könnten Sie nach ihr auch noch im Explorer mit dem Befehl *Suchen* aus dem Menü *Extras* suchen lassen. Falls Sie auch dabei nicht fündig werden, müssen Sie das Office-Setup erneut ausführen, um die Beispieldatei Nordwind.mdb nachträglich zu installieren.

Access starten

Rufen Sie Access auf, indem Sie z.B. das Start-Menü der Task-Leiste von Windows aufschlagen, dort den Befehl *Programme* und aus dessen Unterbefehlen *Microsoft Access* wählen. In der Online-Hilfe erhalten Sie Informationen über mögliche Startoptionen, die Sie beim Aufrufen von Access angeben können; suchen Sie nach *Starten von Microsoft Access mit Befehlszeilenoptionen*.

Datenbank Nordwind.mdb öffnen

Access meldet sich mit dem in Bild 4.2.1 wiedergegebenen Eröffnungs-Dialogfeld, das bei Ihnen natürlich etwas anders aussieht als hier wiedergegeben, weil die Liste der vier zuletzt geöffneten Datenbanken benutzerspezifisch ist.

- Markieren Sie im Eröffnungs-Dialogfeld (vgl. Bild 4.2.1) den Eintrag *Weitere Dateien...*, und bestätigen Sie mit OK, wenn Sie Access gerade gestartet haben. Falls das Eröffnungs-Dialogfeld bereits geschlossen ist, klicken Sie auf die Symbol-Schaltfläche *Datenbank öffnen* (vgl. links nebenstehend) oder wählen Sie den gleichnamigen Befehl aus dem Menü *Datei*. Für den letzteren Fall müssen Sie möglicherweise zunächst das Datenbankfenster aktivieren, was Sie beispielsweise mit der Tastenkombination F11 bewirken können.
- Wechseln Sie im Dialogfeld *Öffnen* zu dem Verzeichnis, in dem sich Nordwind.mdb befindet, vgl. dazu den *Hinweis* anfangs dieses Punktes.

Bild 4.21: Mit diesem Dialogfeld meldet sich Access unmittelbar nach dem Start des Programms.

- Markieren Sie den Eintrag *Nordwind* im Listenfeld, indem Sie darauf klicken, und bestätigen Sie mit OK.

Nach dem Öffnen der Datenbank sollte Ihr Bildschirm so aussehen wie der oben in Bild 4.16 wiedergegebene, wenngleich bei Ihnen eine andere Objektliste des Datenbankfensters aktiviert sein mag, weil Access die in der letzten Sitzung zuletzt aktive Objektliste beim nächsten Öffnen erneut aktiviert.

Formular *Bestellungen öffnen*

- Aktivieren Sie die Objektliste *Formulare* im Datenbankfenster. Dann wird die Liste aller in der Nordwind-Datenbank verfügbaren Formulare angezeigt.
- Markieren Sie in der Liste den Eintrag *Bestellungen*, indem Sie darauf klicken. Klicken Sie anschließend auf die Schaltfläche *Öffnen*. Statt Markieren und anschließend *Öffnen* könnten Sie auch auf den Listeneintrag *Bestellungen* doppelklicken. In jedem Falle sollten Sie dann das in Bild 4.22 dargestellte Formular *Bestellungen* sehen.

Blättern zwischen Datensätzen

- Springen Sie zum letzten Datensatz des Hauptformulars. Klicken Sie dazu auf die Navigationsschaltfläche *Letzter Datensatz* (vgl. links nebenstehend) am unteren linken Formularrand. Sie sollten dann im Formular einen Datensatz angezeigt bekommen, der u.a. ausweist, daß die Rechnung an *Wolski Zajazd* geht. Außerdem erkennen Sie, daß die Bestellung in diesem Falle einen Artikel umfaßt: Im Unterformular, das die Artikel in Datenblattansicht wiedergibt, ist nur eine Zeile für einen Datensatz zu sehen; die Bestellung beläuft sich einschließlich Frachtkosten auf 600,32 DM.

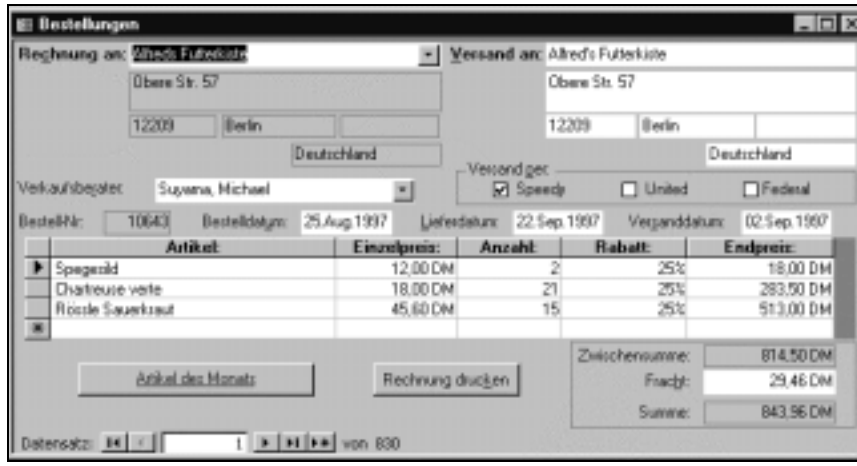


Bild 4.22: Formular »Bestellungen« der Nordwind-Datenbank, Formularansicht

- Blättern Sie einen Datensatz zurück, indem Sie auf die Navigationsschaltfläche *Vorheriger Datensatz* (vgl. links nebenstehend) am unteren linken Formularrand klicken. Dieser Datensatz bezieht sich ebenfalls auf *Wolski Zajazd*, weist aber für diese Bestellung vier verschiedene Artikel mit einer Summe von 706,31 DM auf.

- Blättern Sie zum ersten Datensatz des Bestellformulars zurück, indem Sie auf die Navigationsschaltfläche *Erster Datensatz* (vgl. links nebenstehend) am unteren linken Formularrand klicken. Er bezieht sich auf die Bestellung von *Alfreds Futterkiste*.

Die Artikelliste wird mit Hilfe eines Formulars in einem Formular, einem *Unterformular*, wiedergegeben. Dieses Unterformular befindet sich in der *Datenblattansicht*, einer tabellarischen Wiedergabe von Datensätzen. Dies bedeutet, daß jeder Datensatz in einer Tabellenzeile dargestellt wird. Auf diese Weise kann man mehr als einen Datensatz gleichzeitig sehen. Hier können Sie mit Hilfe der senkrechten Bildlaufleiste blättern.

Abfrage *Quartalsbestellungen* öffnen

Um ein weiteres Objekt (Tabelle, Abfrage, Bericht, anderes Formular etc.) zu öffnen, müssen Sie zunächst das Datenbankfenster in den Vordergrund bringen.

- Schlagen Sie das Menü *Fenster* auf, und wählen Sie darin NORDWIND: *Datenbank*. Alternativ: Drücken Sie die Taste F11. Falls Sie einen Zipfel des Datenbankfensters sehen (nur möglich, wenn sich das aktive Fenster nicht in Vollbilddarstellung befindet), brauchen Sie einfach auf einen beliebigen Punkt des Datenbankfensters zu klicken.

- Aktivieren Sie die Objektliste *Abfragen* im Datenbankfenster. Dann wird die Liste aller verfügbaren Abfragen angezeigt.
- Markieren Sie darin den Eintrag *Quartalsbestellungen*, und klicken Sie anschließend auf *Öffnen*. Alternativ: Doppelklicken Sie auf den Eintrag *Quartalsbestellungen*. Dann sehen Sie das Abfrageergebnis – also lauter Datensätze – zur Abfrage *Quartalsbestellungen*, vgl. Bild 4.23.

Quartalsbestellungen : Auswahlabfrage

Kunden-Code	Firma	Ort	Land
ALFKI	Alfreds Futterkiste	Berlin	Deutschland
ANATR	Ana Trujillo Emparedados y helados	México D.F.	Mexiko
ANTON	Antonio Moreno Taquería	México D.F.	Mexiko
AROUT	Around the Horn	London	Großbritannien
BERGS	Berglunds snabbköp	Luleå	Schweden
BLAUS	Blauer See Delikatessen	Mannheim	Deutschland
BLONP	Blondel père et fils	Strasbourg	Frankreich
BOLID	Bólido Comidas preparadas	Madrid	Spanien
BONAP	Bon app'	Marseille	Frankreich
BOTTM	Bottom-Dollar Markets	Tsawassen	Kanada
BSBEV	B's Beverages	London	Großbritannien
CACTU	Cactus Comidas para llevar	Buenos Aires	Argentinien
CHOPS	Chop-suey Chinese	Bern	Schweiz
COMMI	Comércio Mineiro	São Paulo	Brasilien
CONSH	Consolidated Holdings	London	Großbritannien
DRACD	Drachenblut Delikatessen	Aachen	Deutschland

Datensatz: 1 von 86

Bild 4.23: Abfrage »Quartalsbestellungen«, Datenblattansicht

Auch in der Datenblattansicht einer Abfrage können Sie sich zwischen den Datensätzen mit Hilfe der Navigationsschaltflächen bewegen. Wenn Sie beispielsweise zum letzten Datensatz blättern, sehen Sie wiederum die Bestellung der Firma *Wolski Zajazd*.

Abfrage *Quartalsbestellungen in der Entwurfsansicht*

Eine Abfrage gibt im allgemeinen eine Auswahl von Datensätzen und -feldern einer oder mehrerer zugrundeliegender Tabellen wieder. Die Abfrage selbst wird in der sogen. *Entwurfsansicht* formuliert. Sie können jederzeit zwischen der *Datenblattansicht* und der *Entwurfsansicht* einer Abfrage wechseln (dasselbe gilt für Formulare, Tabellen und Berichte), worauf im einzelnen in den folgenden Kapiteln eingegangen wird. Hier soll nur, am Beispiel einer Abfrage, die Oberflächenseite des Sachverhalts demonstriert werden:

- Klicken Sie, während das Fenster der Abfrage *Quartalsbestellungen* aktiviert ist, auf die Symbol-Schaltfläche *Entwurfsansicht* (vgl. links nebenstehend); diese sollte als Element der Symbol-Schaltfläche *Ansicht*, einer Dropdown-Liste, am linken Rand der Symbolleiste angezeigt werden. Sie sollten einen Bildschirm ähnlich wie in Bild 4.24 dargestellt sehen.

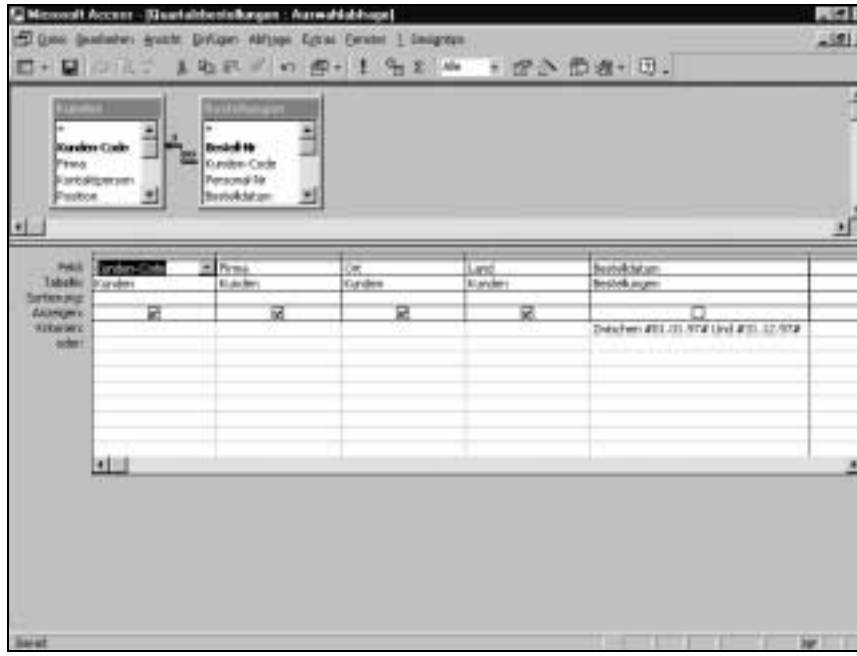


Bild 4.24: Abfrage »Quartalsbestellungen«, Entwurfsansicht. Die Breite der Spalte »Bestelldatum« wurde so verbreitert, daß der Eintrag für das Kriterium vollständig lesbar ist.

Wir können an dieser Stelle nicht auf Details eingehen, nur so viel sei zum Inhalt des Abfrageentwurfs gesagt: Man erkennt, daß die Abfrage Datensätze aus den beiden Tabellen *Kunden* und *Bestellungen* auswählt, denn beide Tabellen sind im oberen Teil des Abfrageentwurfs jeweils als grafisches Objekt angeführt. Ferner ist zu sehen, daß für die Abfrage zum Feld *Bestelldatum* ein Kriterium angegeben wurde, weil die Zeile *Kriterien* für dieses Feld den Eintrag »Zwischen #01.01.97# Und #31.12.97#« enthält. Dieses Kriterium bewirkt, wie unschwer zu deuten ist, daß nur Datensätze ausgewählt und angezeigt werden, deren Bestelldatum zwischen dem 01.01.97 und dem 31.12.97 liegt.

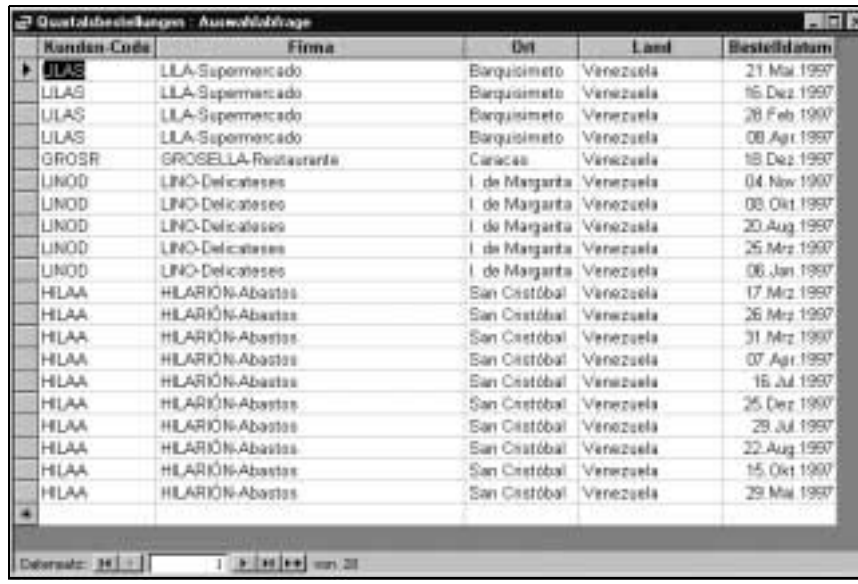
Sie könnten an diesem Abfrageentwurf eine Menge von Änderungen vornehmen, beispielsweise eine weitere Tabelle heranziehen, aus den angeführten Tabellen andere Felder berücksichtigen oder die Datensätze zusätzlich zur zeitlichen Beschränkung auf solche eines bestimmten Landes beschränken. Probieren Sie diese letzte Änderung einmal, indem Sie nur die Datensätze abfragen lassen, die sich auf *Venezuela* beziehen. Sorgen Sie ferner dafür, daß auch das Feld *Bestelldatum*, das bisher ausgeblendet ist (vgl. oben, Bild 4.23), angezeigt wird. Schließlich sollen die Datensätze nach dem Ort in aufsteigender Richtung sortiert werden. Gehen Sie dazu folgendermaßen vor:

- Schreiben Sie für das Feld *Land* in die Zeile *Kriterien* den Text *Venezuela*.
- Klicken Sie für das Feld *Bestelldatum* auf das Kontrollkästchen in der Zeile *Anzeigen*, so daß es eingeschaltet ist.
- Klicken Sie für das Feld *Ort* in die Zeile *Sortierung*, schlagen Sie die Dropdown-Liste auf und wählen Sie den Eintrag *Aufsteigend*.

Der Abfrageentwurf sollte jetzt aussehen wie in Bild 4.25 gezeigt.

Bild 4.25: Abfrage »Quartalsbestellungen«, Entwurfsansicht, nach Durchführung der drei Änderungen am Abfrageentwurf

- Wechseln Sie in die Datenblattansicht, um sich das Ergebnis dieser Änderung anzuschauen. Klicken Sie dazu in der Symbolleiste auf die Symbol-Schaltfläche *Datenblattansicht*. In der Datenblattansicht sollten Sie jetzt nur 19 Datensätze (statt vorher 84) sehen, die sich alle auf *Venezuela* beziehen, vgl. Bild 4.26.



Kunden-Code	Firma	Ort	Land	Bestelldatum
LILAS	LILA-Supermercado	Barquisimeto	Venezuela	21.Mai.1997
LILAS	LILA-Supermercado	Barquisimeto	Venezuela	16.Dez.1997
LILAS	LILA-Supermercado	Barquisimeto	Venezuela	28.Feb.1997
LILAS	LILA-Supermercado	Barquisimeto	Venezuela	08.Apr.1997
GROSR	GROSELLA-Restaurante	Caracas	Venezuela	18.Dez.1997
LINOD	LINO-Delicateses	I. de Margarita	Venezuela	04.Nov.1997
LINOD	LINO-Delicateses	I. de Margarita	Venezuela	08.Okt.1997
LINOD	LINO-Delicateses	I. de Margarita	Venezuela	20.Aug.1997
LINOD	LINO-Delicateses	I. de Margarita	Venezuela	25.Mrz.1997
LINOD	LINO-Delicateses	I. de Margarita	Venezuela	06.Jan.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	17.Mrz.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	26.Mrz.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	31.Mrz.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	07.Apr.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	16.Jul.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	25.Dez.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	29.Jul.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	22.Aug.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	15.Okt.1997
HILAA	HILARIÓN-Abastos	San Cristóbal	Venezuela	29.Mai.1997

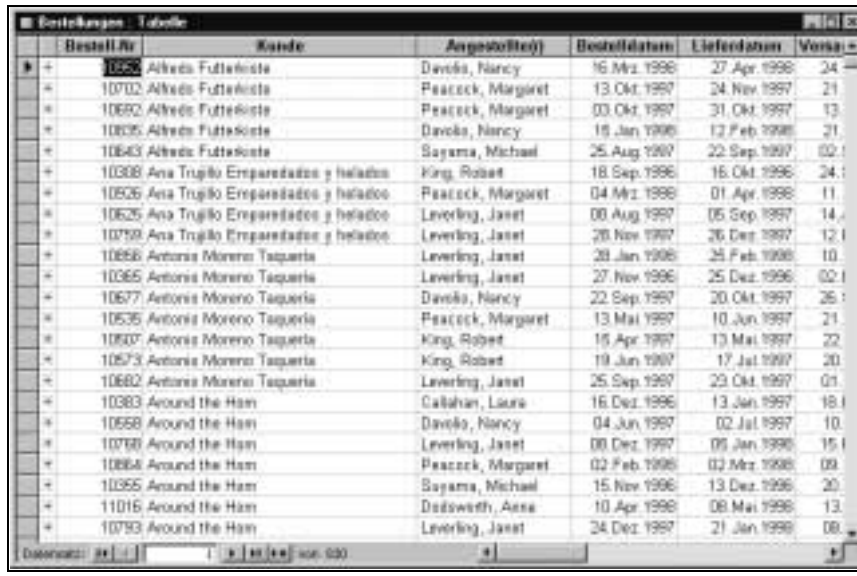
Bild 4.26: Ergebnis der Abfrage »Quartalsbestellungen« mit dem (zusätzlichen) Kriterium »Venezuela« für das Feld »Land«, eingeblendetem Feld »Bestelldatum« und aufsteigend sortiert nach dem Feld »Ort«

Hinweis: Wenn Sie das Abfragefenster schließen, sollten Sie die Frage, ob Änderungen gespeichert werden sollen, verneinen, denn sonst würden die von mir nur als temporär gedachten Änderungen permanent werden!

Tabelle *Bestellungen* öffnen

Um die Tabelle *Bestellungen* zu öffnen, müssen Sie zunächst wiederum zum Datenbankfenster wechseln. Gehen Sie im einzelnen folgendermaßen vor:

- Drücken Sie die Taste [F11], um das Datenbankfenster zu öffnen.
- Schlagen Sie die Objektliste *Tabellen* im Datenbankfenster auf. Dann wird die Liste aller verfügbaren Tabellen angezeigt.
- Markieren Sie darin den Eintrag *Bestellungen* und klicken Sie anschließend auf *Öffnen*. Alternativ: Doppelklicken Sie auf den Eintrag *Bestellungen*. Dann sehen Sie die Tabelle *Bestellungen* auf dem Bildschirm, vgl. Bild 4.27.



Bestell.Nr	Kunde	Angestellte(r)	Bestelldatum	Lieferdatum	Versa
10626	Alfreds Futterkiste	Davolio, Nancy	16.Mrz.1998	27.Apr.1998	24.
10702	Alfreds Futterkiste	Peacock, Margaret	13.Okt.1997	24.Nov.1997	21.
10692	Alfreds Futterkiste	Peacock, Margaret	03.Okt.1997	31.Okt.1997	13.
10835	Alfreds Futterkiste	Davolio, Nancy	15.Jan.1998	12.Feb.1998	21.
10643	Alfreds Futterkiste	Suyama, Michael	25.Aug.1997	22.Sep.1997	02.
10308	Ana Trujillo Emparedados y helados	King, Robert	18.Sep.1996	16.Okt.1996	24.
10926	Ana Trujillo Emparedados y helados	Peacock, Margaret	04.Mrz.1998	01.Apr.1998	11.
10625	Ana Trujillo Emparedados y helados	Leverling, Janet	08.Aug.1997	05.Sep.1997	14.
10759	Ana Trujillo Emparedados y helados	Leverling, Janet	28.Nov.1997	26.Dez.1997	12.
10856	Antonio Moreno Taquería	Leverling, Janet	28.Jan.1998	25.Feb.1998	10.
10365	Antonio Moreno Taquería	Leverling, Janet	27.Nov.1996	25.Dez.1996	02.
10677	Antonio Moreno Taquería	Davolio, Nancy	22.Sep.1997	20.Okt.1997	26.
10535	Antonio Moreno Taquería	Peacock, Margaret	13.Mai.1997	10.Jun.1997	21.
10507	Antonio Moreno Taquería	King, Robert	15.Apr.1997	13.Mai.1997	22.
10573	Antonio Moreno Taquería	King, Robert	19.Jun.1997	17.Jul.1997	20.
10682	Antonio Moreno Taquería	Leverling, Janet	25.Sep.1997	23.Okt.1997	01.
10383	Around the Horn	Callahan, Laura	16.Dez.1996	13.Jan.1997	18.
10558	Around the Horn	Davolio, Nancy	04.Jun.1997	02.Jul.1997	10.
10768	Around the Horn	Leverling, Janet	08.Dez.1997	05.Jan.1998	15.
10864	Around the Horn	Peacock, Margaret	02.Feb.1998	02.Mrz.1998	09.
10355	Around the Horn	Suyama, Michael	15.Nov.1996	13.Dez.1996	20.
11016	Around the Horn	Dodsworth, Anne	10.Apr.1998	08.Mai.1998	13.
10793	Around the Horn	Leverling, Janet	24.Dez.1997	21.Jan.1998	08.

Bild 4.27: Tabelle »Bestellungen« in der Datenblattansicht

Unterdatenblatt mit Erweiterungssymbol einblenden lassen

In der Datenblattansicht der Tabelle *Bestellungen* (vgl. Bild 4.27) befindet sich am linken Rand jeder Datensatzzeile das Erweiterungssymbol (Zeichen +). Diese Erweiterungssymbole werden für Tabellen angezeigt, die zu einer anderen Tabelle die Stellung einer sogen. *Mastertabelle* haben, wobei die andere Tabelle dann als *Detailtabelle* bezeichnet wird (vgl. dazu genauer im folgenden Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, Punkt 5.2.2, *Soll eine Beziehung zwischen den*

Tabellen bestehen?). Die Tabelle *Bestellungen* ist beispielsweise Mastertabelle für die Tabelle *Bestelldetails*, die daher in dieser sogen. 1:n-Beziehung Detailtabelle ist. Das Verhältnis von Master- und Detailtabelle ist u.a. dadurch charakterisiert, daß jedem Datensatz der Mastertabelle mehrere Datensätze der Detailtabelle zugeordnet sind, wobei das Wort *mehrere* auch kein oder ein Datensatz bedeuten kann. Durch Klicken auf das Erweiterungssymbol zu einem bestimmten Datensatz werden in einem Unterdatenblatt die Datensätze der Detailtabelle eingeblendet, die dem betreffenden Datensatz der Mastertabelle zugeordnet sind. Um beispielsweise die Datensätze der Tabelle *Bestelldetails* in einem Unterdatenblatt anzuzeigen, die dem vierten in Bild 4.27 angezeigten Datensatz der Tabelle *Bestellungen* zugeordnet sind, verfahren Sie wie folgt:

- Klicken Sie auf das Erweiterungssymbol des vierten Datensatzes der Tabelle *Bestellungen*. Dann werden die drei Datensätze der Tabelle Bestelldetails, die diesem Datensatz der Tabelle Bestellungen zugeordnet sind, in einem Unterdatenblatt angezeigt, vgl. Bild 4.28.

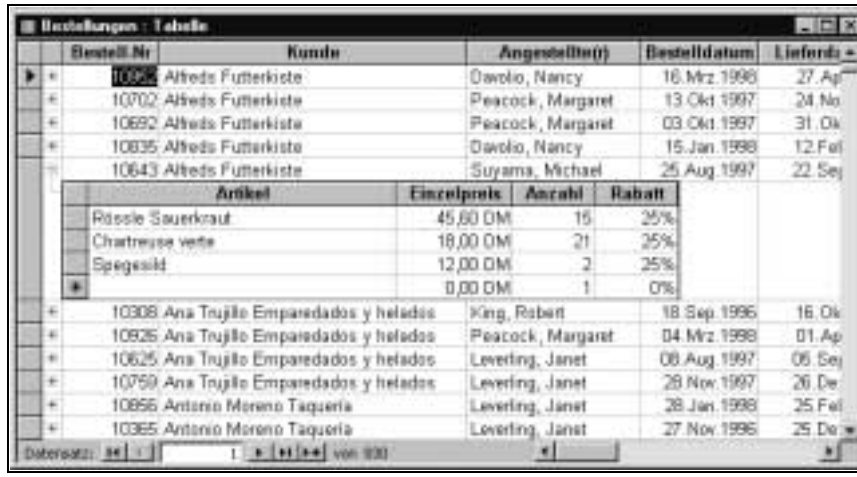


Bild 4.28: Für den vierten Datensatz der Mastertabelle werden die Datensätze der Detailtabelle in einem Unterdatenblatt angezeigt.

Objektfenster schließen

Sie haben jetzt drei Objektfenster geöffnet: Die Tabelle *Bestellungen*, die Abfrage *Quartalsbestellungen* und das Formular *Bestellungen*. Wenn Sie das Menü *Fenster* aufschlagen, werden Sie sehen, daß alle drei sowie das Datenbankfenster dort angeführt werden. Um jetzt alle drei Objektfenster zu schließen, müssen Sie nacheinander jedes einzeln schließen; es gibt keine Möglichkeit, mehrere (oder gar alle) Objektfenster mit einem Mal zu schließen, es sei denn, sie schließen die ganze Datenbank. Gehen Sie im einzelnen folgendermaßen vor:

- Bringen Sie, falls dies nicht bereits zutrifft, eines der drei Fenster (Abfrage *Quartalsbestellungen*, Formular *Bestellungen* oder Tabelle *Bestellungen*) in den Vordergrund, indem Sie es im Menü *Fenster* auswählen (Tastaturalternative: Strg+F6).

- Klicken Sie auf die Symbol-Schaltfläche *Schließen* am rechten oberen Fensterrand des betreffenden Objekts, vgl. links nebenstehend.
- Führen Sie die beiden vorangehenden Punkte jeweils für die beiden anderen Fenster aus.

Access beenden

Beenden Sie Access, indem Sie den Befehl *Beenden* aus dem Menü *Datei* ausführen. Alternativ: Klicken Sie auf die Symbol-Schaltfläche *Schließen* am rechten oberen Rand des Anwendungsfensters.

Kapitel 5

Einführungsbeispiel: Eine einfache relationale Datenbank erstellen

In diesem Kapitel möchte ich Ihnen zeigen, wie Sie eine relativ einfache, gleichwohl nicht triviale Datenbank erstellen. Dies läßt sich nur mit Hilfe eines Beispiels ausführen. Wenn Sie alle Schritte des Kapitels nachvollzogen haben, wird Ihre neue Datenbank derjenigen gleichen, die Sie auf der Beispiel-CD-ROM unter dem Namen *Projekte.mdb* finden können. Daher können Sie die Ergebnisse Ihrer Arbeit mit der fertigen Datenbank der Beispiel-CD-ROM vergleichen. Vielleicht wollen Sie die einzelnen Schritte aber auch gar nicht praktisch nachvollziehen, sondern nur theoretisch (dies sei allerdings nur Benutzern empfohlen, die bereits mit der Oberfläche von Access gut vertraut sind). Dann können Sie *Projekte.mdb* öffnen und sich die verschiedenen Tabellen, Abfragen und Formulare im einzelnen betrachten.

Der Sinn dieses Kapitels liegt darin, Ihnen die *wesentlichen* Schritte, die beim Erstellen einer praktisch einsetzbaren Datenbank zu bewältigen sind, *im Zusammenhang* zu zeigen. Obwohl Sie lernen, zwei Tabellen zu erstellen, eine *Beziehung* zwischen ihnen zu definieren, eine Abfrage und mehrere Formulare (ein verschachteltes Formular eingeschlossen) zu entwerfen, werden hier keinesfalls alle Einzelheiten dieser Gebiete angesprochen. Dazu dienen vielmehr die folgenden Kapitel dieses Buches.

5.1 Problemstellung

Stellen Sie sich eine Firma vor, die Änderungen an und Neuanlagen von Landschaftsprojekten sowie Hoch- und Tiefbauten unterschiedlichster Art (beispielsweise die Neuanlage eines Gartens, den Bau eines Schulsportplatzes, die Reparatur eines Gebäudes etc.) herstellt. Für jedes einzelne Projekt sollen die folgenden Informationen verfügbar sein:

Projektinformationen
Laufende Projektnummer
Projektbezeichnung
Auftragswert
Datum des Projektbeginns
Ist das Projekt bereits abgeschlossen oder noch nicht?
Von welchem Firmenmitarbeiter wird das Projekt verantwortlich betreut?

Gleichzeitig sollen in der Datenbank Informationen über jeden Mitarbeiter verfügbar sein, und zwar zu folgenden Merkmalen:

Personalinformationen
Identitätscode des Mitarbeiters
Name
Vorname
Geschlecht
Geburtsdatum
Einstellungsdatum
Straße
Postleitzahl
Ort
Telefon
Bemerkungen

Obwohl jedes Projekt von nur einem Mitarbeiter verantwortlich betreut wird, gilt das Umgekehrte nicht: Jeder Mitarbeiter betreut im allgemeinen mehrere Projekte gleichzeitig.

Die Bearbeitung der Projekt- und Personaldaten (Eingabe, Veränderung, Lesen) soll auf möglichst einfache und sichere Weise erfolgen, und zwar auch von Mitarbeitern, die keine besonderen Kenntnisse über Access haben, sich jedoch mit der Windows-Oberfläche auskennen. Diese Anforderung läuft darauf hinaus, daß die Bearbeitung der Daten mit Hilfe von Formularen erfolgt, weil diese im allgemeinen die übersichtlichste, bequemste und sicherste Form der Datenbearbeitung ermöglichen.

5.2 Datenbankdesign

Bevor Sie damit beginnen, einzelne Tabellen zu definieren, sollten Sie sich Klarheit darüber verschaffen, wie viele verschiedene Tabellen Sie überhaupt zur Lösung der o.a. Problemstellung benötigen, und welche Beziehungen die Tabellen zueinander aufweisen. Obwohl es auch möglich ist, nachträglich Änderungen an den Strukturen von oder den Beziehungen zwischen Tabellen vorzunehmen, kann dies doch, vor allem, wenn bereits Daten eingegeben wurden, mit besonderen Schwierigkeiten verbunden sein. Daher empfiehlt es sich stets, gründlich über Anzahl, Struktur und Beziehungen der Tabellen nachzudenken, bevor Sie mit der praktischen Realisierung am PC beginnen. Dagegen können Sie die Frage, welche Abfragen, Formulare oder Berichte Sie verwenden werden, getrost auf einen späteren Zeitpunkt verschieben, weil die Datenstruktur von diesen Objekten nicht beeinflußt wird.

5.2.1 Eine oder zwei Tabellen?

Im vorliegenden Falle ist zunächst die Frage zu beantworten, ob alle Projekt- und Personaldaten in einer oder in verschiedenen Tabellen gespeichert werden sollen. Für nur eine Tabelle spräche die daraus resultierende verhältnismäßig einfache Datenbankstruktur, die einen leichteren Datenbankentwurf ermöglichen würde. Mehrere Gründe sprechen aber dafür, für die Projektdaten eine Tabelle und für die Personaldaten eine andere anzulegen:

- Wenn Projekt- und Personaldaten in einer einzigen Tabelle untergebracht werden, müßten *redundante* (doppelt oder mehrfach dieselben) Informationen gespeichert werden. Dies ergibt sich im vorliegenden Falle aus der Forderung, daß jedes Projekt einem Mitarbeiter zugeordnet sein soll. Jedem Projekt entspricht ein Datensatz, und für jeden Datensatz müßten dann alle Informationen, die den Mitarbeiter betreffen, erneut eingegeben werden. Dies bedeutet nicht nur unnötig vielfache Eingabearbeit. Die Konsequenz solch redundanter Datenorganisation führt oft auch zu Widersprüchen. Beispielsweise würde ein Schreibfehler im Namen eines Mitarbeiters (z.B. »Hans Heinrich Meier« statt »Hans-Heinrich Meier«) im allgemeinen für die Verwaltung der Datensätze bedeuten, zwei verschiedene Namen anzunehmen. Auch Änderungen (z.B. der Telefonnummer des Mitarbeiters) müßten, sollen Widersprüche vermieden werden, stets an allen Datensätzen der Projekte, denen der Mitarbeiter zugeordnet ist, vorgenommen werden.
- Jede der beiden sachlich verschiedenen Datenzusammenstellungen – Projekt- und Personalinformationen – ist für sich sinnvoll. Möglicherweise wollen Sie später einmal eine Änderung Ihrer Datenbank vornehmen. Beispielsweise könnte es sich als notwendig erweisen, den Fahrzeugpark in die Datenbank einzubeziehen und dabei die Fahrzeuge ebenfalls einzelnen Mitarbeitern zuzuordnen. Eine solche Erweiterung läßt sich sehr einfach einrichten, wenn Sie die Personaldaten in einer eigenen Tabelle gespeichert haben und nicht zusammen mit den Projektdaten.

- Im vorliegenden Beispiel soll neben anderen auch ein Formular mit folgender Leistung erstellt werden: Sie blättern im Formular zu einem bestimmten Mitarbeiter und erhalten dann im selben Formular eine Tabelle mit allen von diesem Mitarbeiter betreuten Projekten, deren Merkmale Sie betrachten, aber auch bearbeiten können. Solch ein Formular läßt sich ohne weiteren Aufwand nur erstellen, wenn Personal- und Projektdaten in verschiedenen Tabellen, zwischen denen allerdings eine Beziehung bestehen muß, gehalten werden.

5.2.2 Soll eine Beziehung zwischen den Tabellen bestehen?

Wenn Sie einerseits wünschen, daß eine Zuordnung von Datensätzen der Personaltabelle zu Datensätzen der Projekttabelle erfolgt und andererseits möchten, daß die Verwaltung dieser Zuordnung automatisch durch das DBMS – also von Access – vorgenommen wird, müssen Sie eine Beziehung (Relation) zwischen den beiden Tabellen definieren. Sie konzipieren damit eine *relationale* Datenbank.

Beziehungen zwischen Tabellen können als 1:n-Beziehung, 1:1-Beziehung oder als m:n-Beziehung definiert werden, vgl. genauer Kap. 7, *Eine Datenbank konzipieren*. In unserem Beispiel muß zwischen der Personal- und der Projekttabelle eine 1:n-Beziehung definiert werden, weil *ein* (daher 1) Mitarbeiter *mehrere* (daher n) Projekte betreut. Eine 1:n-Beziehung läßt auch den Fall zu, daß einem Mitarbeiter kein Projekt zugeordnet ist, während das Umgekehrte nicht zulässig ist: Es können nur Projekte eingegeben werden, denen jeweils ein Mitarbeiter zugeordnet ist.

5.3 Neue Datenbank erstellen

Nach den allgemeinen und abstrakten (gleichwohl notwendigen) Vorüberlegungen der vorangehenden Punkte soll es nun an die praktische Seite gehen, die Datenbank einzurichten. Jede Access-Datenbank bekommt einen Dateinamen, unter dem sie gespeichert und geöffnet wird. Auf der Beispiel-CD-ROM hat die hier entwickelte Datenbank den Namen *Projekte.mdb*. Sie sollten sich einen anderen Namen wählen, um Namensverwechslungen zu vermeiden. Ich gehe im folgenden davon aus, daß Sie den Namen *Proj.mdb* verwenden; jeder andere Name, welcher den Konventionen über Dateinamen genügt, ist aber ebenso gut geeignet. Um die neue Datenbank *Proj.mdb* anzulegen, gehen Sie folgendermaßen vor:

- Starten Sie ggf. Access. Sofern Access bereits läuft und eine andere Datenbank geöffnet ist, schließen Sie diese Datenbank am besten vor dem nächsten Schritt. Access schließt eine geöffnete Datenbank ansonsten von sich aus, wenn Sie eine neue erstellen, denn es kann für eine Access-Sitzung jeweils nur eine Datenbank zur Zeit geöffnet sein. (Allerdings ist es möglich, mehrere Access-Sitzungen gleichzeitig laufen zu haben, wobei in jeder Sitzung eine andere Datenbank geöffnet sein kann.)

- Aktivieren Sie ggf. das Datenbankfenster über das Menü *Fenster* oder mit der Tastenkombination F11.

- Wählen Sie den Befehl *Neu...* aus dem Menü *Datei* oder klicken Sie auf die Symbol-Schaltfläche *Neu*, vgl. links nebenstehend. Dann erscheint das Dialogfeld *Neu,* vgl. Bild 5.1.
- Wählen Sie aus dem Register *Allgemein* die Datenbankvorlage *Leere Datenbank*, und bestätigen Sie mit OK. Dann erscheint das Dialogfeld *Neue Datenbankdatei*, vgl. Bild 5.2.

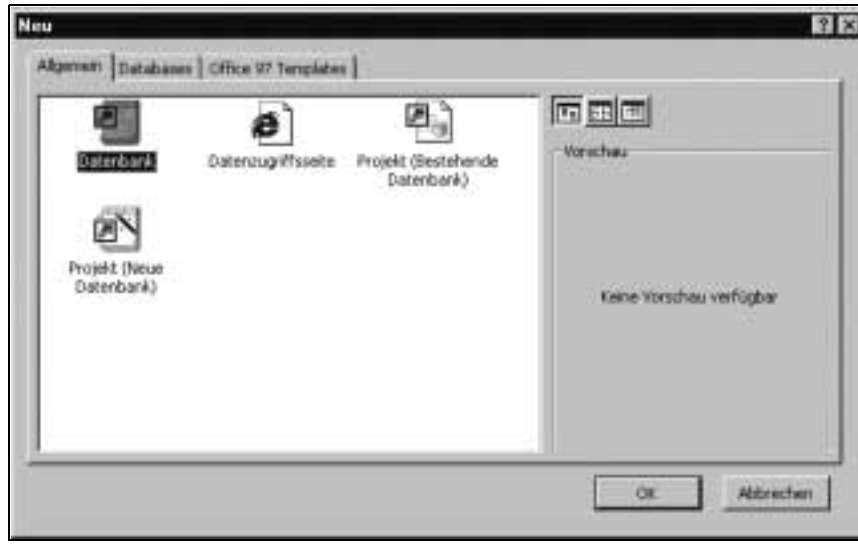


Bild 5.1: Dialogfeld »Neu« zur Auswahl einer Datenbankvorlage. Wechseln Sie im Dialogfeld »Neue Datenbankdatei« zu dem Ordner, in dem Sie die neue Datenbank speichern möchten.

- Geben Sie im Dialogfeld *Neue Datenbankdatei* in das Eingabefeld *Dateiname*: den Namen *Proj* ein; Access fügt dann automatisch die Dateinamenserweiterung .mdb (für Microsoft DataBase) an.
- Bestätigen Sie das Dialogfeld *Neue Datenbankdatei* mit *Erstellen*.

Es existiert jetzt die Datenbank Proj.mdb, und zwar nicht nur im Arbeitsspeicher, sondern auch auf der Festplatte. Diese zunächst noch völlig leere (d.h. ohne Tabellen, Formulare etc.) Datenbank belegt bereits einen Speicherplatz von ca. 96 KB! Das Datenbankfenster zeigt jetzt in seiner Titelleiste den Namen der Datenbank an.

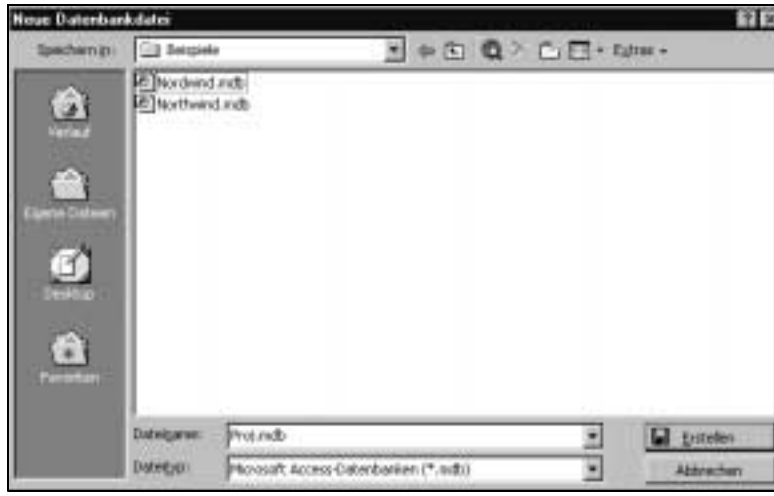


Bild 5.2: Wenn Sie in dieser Situation mit OK bestätigen, wird die neue Datenbank unter dem Namen Proj.mdb im Ordner »Beispiele« angelegt.

5.4 Tabelle *Personal definieren*

5.4.1 Überlegungen zum Tabellendesign

Im Prinzip ist es gleichgültig, ob Sie zuerst die Personaltabelle und dann die Projekttabelle definieren oder in umgekehrter Reihenfolge vorgehen. Nur für den Fall, daß Sie bereits Daten eingeben wollen (sei es zum Testen, sei es aus anderen Gründen), bevor beide Tabellen, einschließlich der 1:n-Beziehung, definiert sind, ist es empfehlenswert, mit der Personaltabelle anzufangen. Falls Sie nämlich Daten in die Projekttabelle eingeben und erst danach die 1:n-Beziehung zur Personaltabelle herstellen, könnten sich Probleme ergeben, die mit der sogen. *referentiellen Integrität* (vgl. dazu in diesem Kapitel unten, Punkt 5.7, *Beziehung zwischen den Tabellen definieren*) zusammenhängen.

Nicht zuletzt aus diesen Gründen soll zunächst die Personaltabelle definiert werden. Aus der Problemstellung (vgl. oben in diesem Kapitel, Punkt 5.1, *Problemstellung*) ist vorgegeben, daß diese Tabelle zwölf Felder enthalten muß. Zwei Punkte sind für die Definition eines Feldes *obligatorisch*:

- Jedes Feld muß einen Namen bekommen.
- Für jedes Feld müssen Sie einen Datentyp festlegen.

Die Angabe von Feldeigenschaften und Tabelleneigenschaften ist wahlfrei, d.h. Sie können dafür Werte angeben, dies muß jedoch nicht geschehen. Statt dessen können Sie ggf. die Voreinstellungen übernehmen. Für die Personaltabelle sollen die folgenden Feldnamen, Felddatentypen, Feldeigenschaften und Tabelleneigenschaften gelten:

5.4.2 Entwurf der Tabelle *Personal*

Tabellenfelder

Feldname	Felddatentyp	Feldeigenschaften
PersonalCode	Text	*Feldgröße: 5* *Eingabeformat:* >LLLLL *Eingabe erforderlich*: Ja
Nachname	Text	*Eingabe erforderlich*: Ja
Vorname	Text	
Sex	Text	*Feldgröße*: 1 *Gültigkeitsregel*: »m« oder »w«; *Gültigkeitsmeldung*: Geben Sie m oder w ein! *Eingabe erforderlich*: Ja
Anrede	Text	Feldgröße: 30 Nachschlagefeld als Kombinationsfeld
Geburtsdatum	Datum/Zeit	*Eingabe erforderlich*: Ja
Einstellungsdatum	Datum/Zeit	*Standardwert*: 1. des aktuellen Monats *Eingabe erforderlich*: Ja
Straße	Text	*Feldgröße:* 100
PLZ	Zahl	*Feldgröße:* Long Integer *Eingabeformat:* 00000 *Standardwert*: ohne Wert
Ort	Text	*Feldgröße:* 100
Telefon	Text	*Feldgröße*: 30
Bemerkungen	Memo	

Tabelle 5.1: Feldnamen, Datentypen und Feldeigenschaften der Tabelle »Personal«

Sinn und Bedeutung dieser Angaben seien im folgenden kurz erklärt.

Feldnamen

Namen – Feldnamen sowie allgemein Namen für Objekte – dürfen in Access bis zu 64 Zeichen enthalten und können aus einer beliebigen Kombination von Buchstaben, Zahlen, Leerzeichen und Sonderzeichen, mit Ausname von Punkten (.), Ausrufezeichen (!), Akzentzeichen (') und eckigen Klammern ([ ]) bestehen. Außerdem dürfen Namen nicht mit Leerzeichen beginnen und keine Steuerzeichen (ASCII-Werte 0 bis 31) enthalten. Die obigen Feldnamen entsprechen dieser Konvention. Sie haben darüber hinaus den Vorteil, einerseits sprechend (d.h. auf den Feldinhalt schließen lassend), andererseits kurz zu sein.

Felddatentypen

Für jedes Feld muß ein Datentyp festgelegt werden. Dies hat zwei Konsequenzen:

- In ein Feld können nur solche Werte eingegeben werden, die dem festgelegten Datentyp entsprechen. So können Sie für ein Feld, dem Sie den Datentyp *Zahl* zugewiesen haben, nur Zahlenwerte eingeben, nicht jedoch Text. Dies mögen Sie als unnötige Einschränkung empfinden. Beachten Sie aber, daß eine klare Datentypstruktur auch zu einer klaren Datenstruktur beiträgt.
- Das Festlegen von Datentypen für die Felder einer Tabelle ermöglicht Access eine wesentlich effizientere Datenverwaltung. Es ist nicht zuletzt aus diesen Gründen auch bei anderen DBMS üblich.

Die für den Entwurf der Personaltabelle von mir vorgeschlagenen Datentypen erklären sich im einzelnen folgendermaßen:

Text. Dies ist der allgemeinste Datentyp. In Felder des Datentyps *Text* können Sie jede beliebige Zeichenfolge eingeben, sofern sie 255 Zeichen nicht übersteigt. Diese Freiheit hat andererseits ihren Preis: Auch wenn Sie Zahlenwerte in ein Feld mit dem Datentyp *Text* eingeben, werden diese als Texte und nicht als Zahlen angenommen und interpretiert. Daher können Sie beispielsweise mit Feldern des Datentyps *Text* nicht rechnen, jedenfalls nicht unmittelbar.

Datum/Uhrzeit. Da Access den besonderen Datentyp *Datum/Zeit* bereitstellt, versteht es sich von selbst, ihn für die beiden Felder *Geburtsdatum* und *Einstellungsdatum* zu verwenden.

Memo. Felder des Datentyps *Memo* gleichen denen des Datentyps *Text* mit einem Unterschied: In ein *Text*-Feld können Sie maximal 255 Zeichen eingeben, in ein *Memo*-Feld dagegen bis zu 65.535 Zeichen. Für längere Bemerkungen ist dieser Datentyp daher erforderlich.

Über diese Felddatentypen hinaus kennt Access noch weitere; insgesamt werden neun Felddatentypen unterschieden, vgl. genauer Kap. 8, *Tabellen: Grundlagen*, oder die Online-Hilfe zum Stichwort *DataType-Eigenschaft*.

Feldeigenschaften

Zu jedem Feld können Sie bestimmte Eigenschaften festlegen (oder, im Sprachgebrauch von Access: *einstellen*). Um den Umgang mit und die Wirkung von Feldeigenschaften zu demonstrieren, schlage ich für die Personaltabelle die oben in Tabelle 5.1 wiedergegebenen beispielhaft vor. Diese werden im folgenden kurz erläutert.

Eingabe erforderlich. Mit der Eigenschaft *Eingabe erforderlich* legen Sie fest, ob in einem Feld ein Wert erforderlich ist oder nicht. Wenn diese Eigenschaft auf *Ja* eingestellt ist, müssen Sie bei der Eingabe von Daten in einen Datensatz einen Wert in das Feld (oder das Steuerelement eines Formulars, das an dieses Feld gebunden ist) eingeben.

Feldgröße. Die Eigenschaft *Feldgröße* steht nur für die beiden Felddatentypen *Text* und *Zahl* zur Verfügung. Bei einem *Text*-Feld können Sie durch eine entsprechende Angabe Platz für längere Texte (voreingestellt sind 50 Zeichen) schaffen oder Speicherplatz sparen. Letzteres trifft in unserem Beispiel für die Felder *Sex* und *Telefon* zu. Bei einem *Zahl*-Feld geben Sie als Feldgröße keine Zahl an, sondern im Grunde einen weiter spezifizierten Datentyp. In unserem Beispiel soll für das *Zahl*-Feld *PLZ* als Feldgröße *Long Integer* eingestellt werden. Damit wird festgelegt, daß dieses Feld nur ganze Zahlen (Integer bedeutet Ganzzahligkeit) annimmt, die eine Größe von bis zu 2.147.483.647 haben dürfen. Die Feldgröße *Integer* reicht für fünfstellige Postleitzahlen nicht aus, weil sie nur Zahlen bis zu 32.767 zuläßt.

Gültigkeitsregel. Für das Feld *Sex* soll eine Gültigkeitsregel festgelegt werden. Wenn Sie für ein Feld eine Gültigkeitsregel festlegen, können Sie in dieses Feld nur solche Werte eingeben, die der Gültigkeitsregel entsprechen. Eine Gültigkeitsregel schränkt den bereits durch den Felddatentyp vorgegebenen Rahmen weiter ein. Für das Feld *Sex* soll als Gültigkeitsregel definiert werden, daß nur die Zeichen *m* und *w* zugelassen sind. Dann akzeptiert das Feld *Sex* der Personaltabelle als Eingabe ausschließlich einen dieser beiden Buchstaben. Wie Sie diese Gültigkeitsregel praktisch formulieren, wird weiter unten erklärt.

Gültigkeitsmeldung. Wenn Sie eine Eingabe vornehmen, die gegen eine festgelegte Gültigkeitsregel verstößt, blendet Access eine Warnmeldung ein. Der Standardtext dieser Meldung würde im Falle einer Verletzung der genannten Gültigkeitsregel für das Feld *Sex* folgendermaßen lauten (vgl. Bild 5.3):

Bild 5.3: Standardmeldung bei einer Dateneingabe, die gegen eine benutzerdefinierte Gültigkeitsregel verstößt, hier für den Fall des Feldes »Sex«

Da dieser Standardtext dem Benutzer meistens keine hinreichenden positiven Hinweise liefert, wie er die Daten richtig einzugeben hat, sollten Sie die Meldung durch einen eigenen Text ersetzen. Dies geschieht dadurch, daß Sie für die Eigenschaft *Gültigkeitsmeldung* einen Text angeben, der dann den

Standardtext ersetzt. Beispielsweise können Sie für Verstöße gegen die Gültigkeitsregel des Feldes *Sex* den folgenden Text einblenden lassen: »Geben Sie m oder w ein!«.

Standardwert. Wenn Sie für ein Feld einen Standardwert festlegen, erscheint das Feld beim Eingeben eines neuen Datensatzes mit diesem Wert vorbesetzt. Diesen Wert können Sie dann übernehmen oder überschreiben. Für das Feld *Einstellungsdatum* soll als Standardwert der erste Tag des aktuellen Monats festgelegt werden. Dem liegt die Überlegung zugrunde, daß Einstellungen häufig zum 1. eines Monats erfolgen und die Eingaben für den neuen Mitarbeiter in die Personaltabelle im allgemeinen zwar im selben Monat, jedoch ein paar Tage nach der Einstellung erfolgen. Falls diese Unterstellung zutrifft, kann der Standardwert übernommen werden, falls nicht, ist er mit einem anderen Datum zu überschreiben.

Eingabeformat. Die Einstellung der Eigenschaft *Eingabeformat* bestimmt, wie Daten in ein Feld eingegeben werden. Wenn Sie die Eigenschaft z.B. auf »000-00-0000« setzen, werden die Bindestriche wie angegeben und jede Null als Unterstrich (_) angezeigt. Für das Feld *Personalcode* soll als Eingabeformat >LLLLL angegeben werden; dann werden nur die Buchstaben A – Z akzeptiert (dies bewirken die fünf L-Zeichen), und Klein- werden in Großbuchstaben umgewandelt (dies wird durch das Zeichen > festgelegt). Für das Feld *PLZ* soll als Eingabeformat 00000 angegeben werden; dann werden nur fünfstellige Zahlen akzeptiert, wie dies dem deutschen Postleitzahlsystem entspricht. Ein Eingabeformat wirkt wie eine Maske, welche die Daten filtert, bevor diese in der Tabelle gespeichert werden. In gewisser Weise hat ein *Eingabeformat* auch den Charakter einer *Gültigkeitsregel* und kann eine solche oftmals ersetzen.

5.4.3 Praktisches Festlegen der Feldnamen, Felddatentypen und Feldeigenschaften

Nachdem Sie die neue Datenbank *Proj.mdb* angelegt haben (vgl. in diesem Kapitel oben, Punkt 5.3, *Neue Datenbank erstellen*) und im vorangehenden Punkt ausführliche Überlegungen zum Tabellendesign nachvollzogen haben, können Sie nunmehr die Personaltabelle erstellen. Zu diesem Zweck erzeugen Sie zunächst eine neue Tabelle. Gehen Sie dazu folgendermaßen vor:

- Wechseln Sie ggf. zum Datenbankfenster (Menü *Fenster* oder Tastenkombination F11).
- Vergewissern Sie sich, daß in der Objekt-Symbolleiste des Datenbankfensters das Symbol *Tabellen* aktiviert ist; klicken Sie sonst darauf.
- Doppelklicken Sie im Datenbankfenster auf den Eintrag *Erstellt eine Tabelle in der Entwurfsansicht*. Dann erhalten Sie ein Tabellenfenster in der Entwurfsansicht, vgl. Bild 5.4. Die neue Tabelle trägt den vorläufigen Namen *Tabelle1*. Dieser soll später von Ihnen durch den Namen *Personal* ersetzt werden.

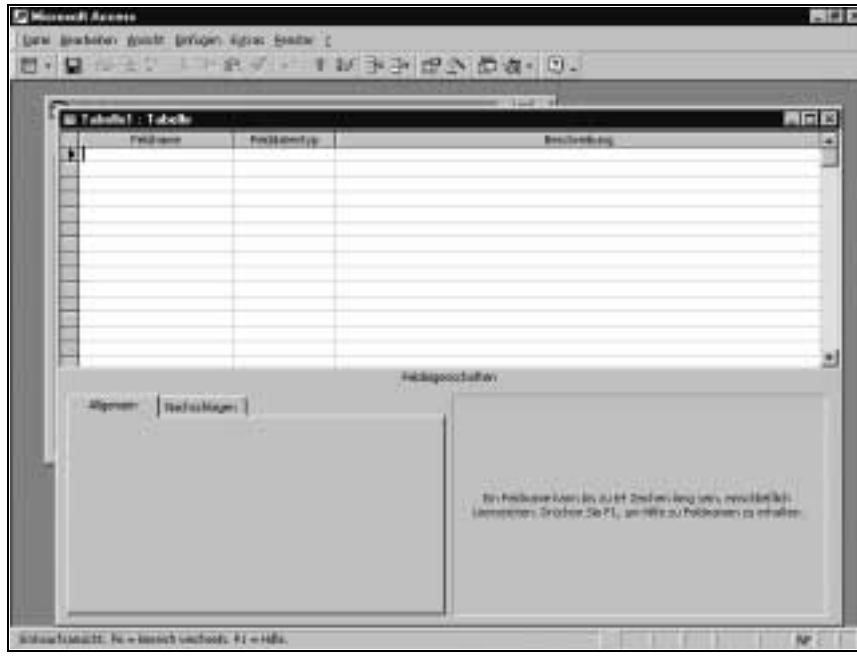


Bild 5.4: Tabellenfenster im Access-Bildschirm unmittelbar nach dem Anfordern einer neuen Tabelle in der Entwurfsansicht

Jetzt können Sie die Feldnamen, Datentypen und Feldeigenschaften eingeben. Dabei können Sie prinzipiell verschiedenen Strategien folgen: Manche Benutzer ziehen es vor, zunächst alle Feldnamen einzugeben und erst anschließend die Datentypen und Eigenschaften festzulegen. Andere definieren Feld für Feld jeweils mit Datentyp und Eigenschaften. Ich demonstriere hier den ersten Weg. Geben Sie daher zunächst alle Feldnamen in die Spalte *Feldname* ein. Gehen Sie dazu im einzelnen folgendermaßen vor:

Feldnamen festlegen

- Klicken Sie auf die erste Zeile der Spalte *Feldname*.
- Schreiben Sie den Namen *Personalcode*.
- Klicken Sie auf die zweite Zeile der Spalte *Feldname* (oder drücken Sie die Richtungstaste [↓]). Dann erscheint in der ersten Zeile der Spalte *Felddatentyp* der Eintrag *Text*, der von Access als Standard-Datentyp angenommen wird. Sie werden diesen Typ in der nächsten Runde ggf. ändern.
- Schreiben Sie den Namen *Nachname*.
- Wiederholen Sie die beiden vorangehenden Schritte entsprechend, bis Sie alle Namen eingegeben haben.

Ihre Tabelle sollte jetzt so aussehen wie diejenige in Bild 5.5.

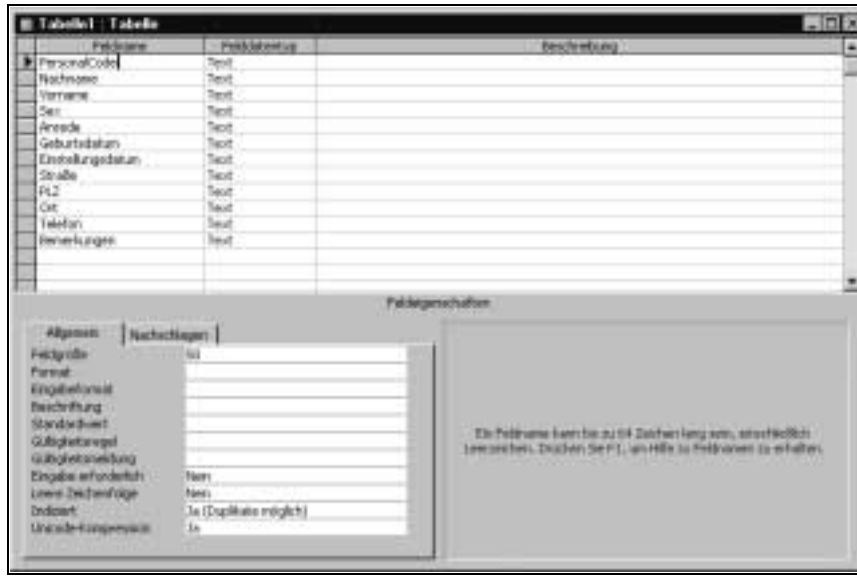


Bild 5.5: Personaltabelle in der Entwurfsansicht nach Eingabe aller Feldnamen

Felddatentypen festlegen

Legen Sie jetzt die Felddatentypen fest. Für die Felder *PersonalCode*, *Nachname*, *Vorname*, *Sex*, *Anrede*, *Straße*, *Ort* und *Telefon* brauchen Sie keine Änderung vorzunehmen, weil diese Felder den Datentyp *Text* bekommen sollen, der von Access bereits voreingestellt ist. Am Beispiel des Feldes *Geburtsdatum* demonstriere ich, wie Sie die Felddatentypen der anderen Felder bestimmen:

- Klicken Sie in die Spalte *Felddatentyp* des Feldes *Geburtsdatum*. Dann erscheint am rechten Rand des Tabellenfeldes der Pfeil zum Aufschlagen einer Dropdown-Liste.
- Schlagen Sie das Dropdown-Listenfeld auf, indem Sie auf den Pfeil klikken. In Bild 5.6 sehen Sie das aufgeschlagene Dropdown-Listenfeld mit den zur Verfügung stehenden neun Felddatentypen als Einträgen.
- Klicken Sie auf den hier erforderlichen Datentyp *Datum/Uhrzeit*. Dann wird dieser Datentyp in das Tabellenfeld übernommen.
- Verfahren Sie für die Felder *Einstellungsdatum*, *PLZ* und *Bemerkungen* entsprechend: Klicken Sie in die Spalte *Felddatentyp* der jeweiligen Tabellenzeile, die den betreffenden Feldnamen enthält, schlagen Sie das Dropdown-Listenfeld auf und wählen Sie den erforderlichen Felddatentyp aus. Danach sollte Ihr Bildschirm aussehen wie in Bild 5.7.

Bild 5.6: Aufgeschlagenes Dropdown-Listenfeld mit den Felddatentypen

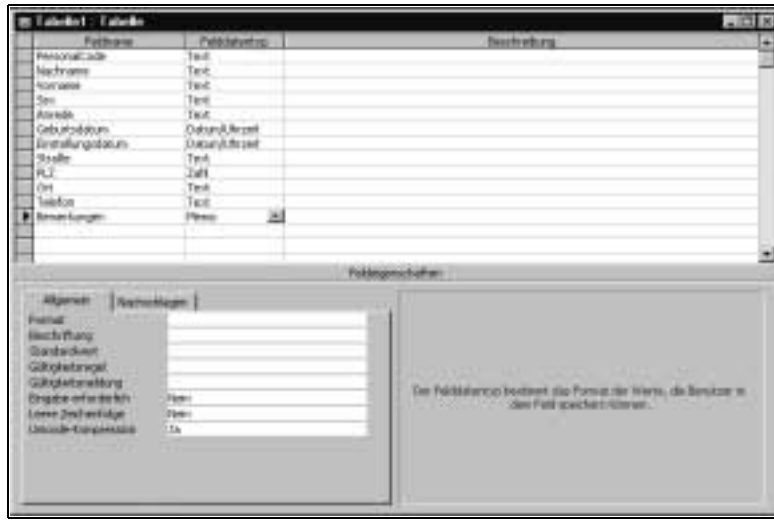


Bild 5.7: Personaltabelle mit für jedes Feld festgelegtem Felddatentyp

Feldeigenschaften festlegen

Das Festlegen der Feldeigenschaften ist im allgemeinen aufwendiger als die Angabe der Feldnamen und -datentypen, weil Sie für jedes Feld mehrere Eigenschaften angeben können. Die Feldeigenschaften geben Sie in die Eingabefelder unter *Feldeigenschaften*, links unten im Tabellenfenster, an (vgl. Bild 5.7). Die dort verfügbaren Eigenschaften sind kontextabhängig: Es werden jeweils die Eigenschaften eingeblendet, die dem Felddatentyp des Feldes mit dem *Fokus* entsprechen. Umgekehrt gilt: Wenn Sie Eigenschaften für ein Feld festlegen oder ändern wollen, müssen Sie diesem Feld zunächst den Fokus zuweisen. Welches Feld den Fokus hat, d.h. welche Zeile der Tabelle aktiv ist, erkennen Sie an dem kleinen schwarzen Dreieck am linken Rand

einer Zeile. In Bild 5.5 beispielsweise hat die erste Zeile, also das Feld *PersonalCode*, den Fokus, in Bild 5.7 dagegen das Feld *Bemerkungen*, also die letzte Zeile. Es bekommt jeweils das Feld (die Zeile) den Fokus, in dessen Zelle der Cursor steht. Sie ändern den Fokus also einfach, indem Sie auf irgendeine Zelle oder auf den linken Rand der entsprechenden Zeile klicken (oder die Markierung entsprechend mit den Richtungstasten bewegen).

Viele Feldeigenschaften sind bereits von Access voreingestellt worden. Die Eigenschaft *Eingabe erforderlich* beispielsweise ist für alle Felder (soweit verfügbar) auf den Wert *Nein* eingestellt worden. Ähnliches gilt für die Eigenschaft *Indiziert*, die ebenfalls generell auf *Nein* eingestellt wurde. Für andere Feldeigenschaften wurden dagegen keine Einstellungswerte vorgegeben, beispielsweise für *Format*. Im folgenden sollen die Feldeigenschaften nur in den Fällen geändert werden, in denen sich dies aus der o.a. Tabelle 5.1, *Feldnamen, Datentypen und Feldeigenschaften der Tabelle »Personal«*, ergibt. Ansonsten sollen die Voreinstellungen unverändert übernommen werden. Gehen Sie im einzelnen folgendermaßen vor:

PersonalCode

- Klicken Sie in eine beliebige Zelle der Zeile, die das Feld *PersonalCode* beschreibt; dann bekommt diese Zeile den Fokus, kenntlich an der schwarzen Dreiecksmarkierung am linken Zeilenrand. Im linken unteren Teil des Tabellenfensters erscheinen jetzt die Bearbeitungsfelder für die elf Eigenschaften, die für ein Feld mit dem Datentyp *Text* verfügbar sind.
- *Feldgröße*. Zunächst soll die Feldgröße auf 5 eingestellt werden. Klicken Sie zu diesem Zweck auf das Bearbeitungsfeld der Eigenschaft *Feldgröße* (Tabellenfenster links unten), löschen Sie die darin stehende Zahl 50, und geben Sie statt dessen die Zahl 5 ein.
- *Eingabeformat*. Schreiben Sie in das Bearbeitungsfeld *Eingabeformat* die Zeichenfolge

  >LLLLL

  Dieser Ausdruck sorgt dafür, daß nur die Buchstaben A-Z akzeptiert werden und daß davon genau fünf eingegeben werden müssen.
- *Eingabe erforderlich*. Klicken Sie auf das Bearbeitungsfeld der Eigenschaft *Eingabe erforderlich*. Dann erscheint an dessen rechtem Rand der Pfeil eines Dropdown-Listenfeldes, vgl. Bild 5.8, das auch die anderen für das Feld *PersonalCode* festgelegten Eigenschaftseinstellungen wiedergibt.

Schlagen Sie das Dropdown-Listenfeld auf und wählen Sie darin den Eintrag *Ja*. Alternativ könnten Sie auch das Wort *Nein* mit dem Wort *Ja* direkt überschreiben. Darüber hinaus können Sie auch auf den Eintrag doppelklicken; dann nimmt das Dropdown-Listenfeld den nächsten Wert aus der Liste an. In diesem Falle weist die Liste nur die beiden Werte *Ja* und *Nein* auf, daher wird durch Doppelklicken aus *Ja Nein* und umgekehrt. Diese angeführten Möglichkeiten der Bearbeitung einer Dropdown-Liste gelten im übrigen generell für derartige Eigenschaftenfelder.

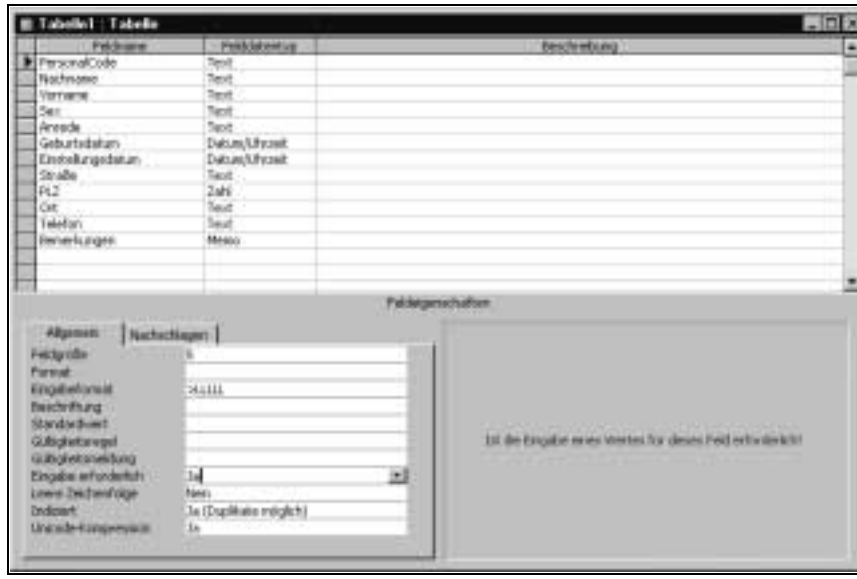


Bild 5.8: Der Pfeil des Dropdown-Listenfeldes erscheint im Bearbeitungsfeld der Eigenschaft »Eingabe erforderlich«, weil dieses aktiv ist.

Nachname

- Klicken Sie in eine beliebige Zelle der Zeile, die das Feld *Nachname* beschreibt; dann bekommt diese Zeile den Fokus.
- *Eingabe erforderlich*. Klicken Sie auf das Bearbeitungsfeld der Eigenschaft *Eingabe erforderlich*, so daß an dessen rechtem Rand der Pfeil des Dropdown-Listenfeldes erscheint. Schlagen Sie das Dropdown-Listenfeld auf, und wählen Sie darin den Eintrag *Ja*, oder überschreiben das Wort *Nein* mit dem Wort *Ja* direkt.

Sex

Für das Feld *Sex* soll die *Feldgröße* auf 1 eingestellt werden, als *Gültigkeitsregel* soll angegeben werden, daß nur die Buchstaben *m* und *w* bei der Eingabe in die Tabelle akzeptiert werden, und als *Gültigkeitsmeldung* soll der Text *Geben Sie m oder w ein!* ausgegeben werden. Gehen Sie bitte folgendermaßen vor:

- Klicken Sie auf eine beliebige Zelle der Zeile, die das Feld *Sex* beschreibt; damit diese Zeile den Fokus bekommt.
- *Feldgröße*. Klicken Sie auf das Bearbeitungsfeld der Eigenschaft *Feldgröße* (Tabellenfenster links unten), und überschreiben Sie die darin stehende Zahl 50 mit der Zahl 1.
- *Gültigkeitsregel*. Legen Sie jetzt die Gültigkeitsregel fest. Klicken Sie dazu in das Bearbeitungsfeld der Eigenschaft *Gültigkeitsregel*, und geben Sie die folgende Zeichenfolge ein:

  In (»m«;"w«)

Dies ist ein *Ausdruck*, der festlegt, daß nur Angaben akzeptiert werden, die in der Liste zwischen den Klammern () angegeben sind, hier also die Buchstaben *m* und *w*, die als Text in einem Ausdruck zwischen Anführungszeichen angegeben werden müssen. Access erlaubt eine große Vielfalt von Ausdrücken. Die Regeln zum Formulieren von Ausdrücken werden gesondert ausführlich in Kap. 39, *Ausdrücke, Funktionen und Operatoren*, beschrieben. Statt des Ausdrucks

In (»m«;"w«)

- hätten Sie übrigens auch den Ausdruck

  »m« Oder »w«

- angeben können, da dieser das gleiche leistet.
- *Gültigkeitsmeldung*. Legen Sie die Gültigkeitsmeldung fest. Klicken Sie zu diesem Zweck in das Bearbeitungsfeld für *Gültigkeitsmeldung,* und schreiben Sie den Text *Geben Sie m oder w ein!* (in diesem Falle übrigens ohne Anführungszeichen!).
- *Eingabe erforderlich*. Stellen Sie schließlich die Eigenschaft *Eingabe erforderlich* auf den Wert *Ja* ein.

Anrede

- *Feldgröße*. Klicken Sie auf das Bearbeitungsfeld der Eigenschaft *Feldgröße*, und überschreiben Sie die darin stehende Zahl 50 mit der Zahl 30.
- *Nachschlagefeld*. Für ein Tabellenfeld mit einem der Datentypen *Text*, *Zahl* oder *Ja/Nein* können Sie ein Nachschlagefeld festlegen. In unserem aktuellen Beispiel eines Feldes für die Anrede soll dieses Nachschlagefeld die Form einer Dropdown-Liste haben, die beim Aufschlagen die vier Einträge

  Frau
  Herr
  Frau Dr.
  Herr Dr.

  zum Auswählen anzeigt. Zum Einrichten eines derartigen Nachschlagefeldes gehen Sie folgendermaßen vor:

- Klicken Sie bei den Feldeigenschaften auf die Registerkarte *Nachschlagen*.
- Schlagen Sie die Dropdown-Liste *Steuerelement anzeigen* auf, und wählen Sie darin den Eintrag *Kombinationsfeld*. Daraufhin erweitert sich das Eigenschaftenfenster für die Eigenschaften, die ein Kombinationsfeld (das ist eine Dropdown-Liste) besitzt, vgl. Bild 5.9.
- Wählen Sie aus der Dropdown-Liste zu *Herkunftstyp* den Eintrag *Werteliste*. Damit bestimmen Sie, daß die Werteliste des Dropdown-Listenfeldes direkt als Einstellung der Eigenschaft *Datensatzherkunft* angegeben werden kann (und muß). (Bei der Definition der Tabelle *Projekte* werden Sie erfahren, wie die Einträge eines derartigen Nachschlagefeldes durch die Datensätze einer anderen Tabelle bestimmt werden.)

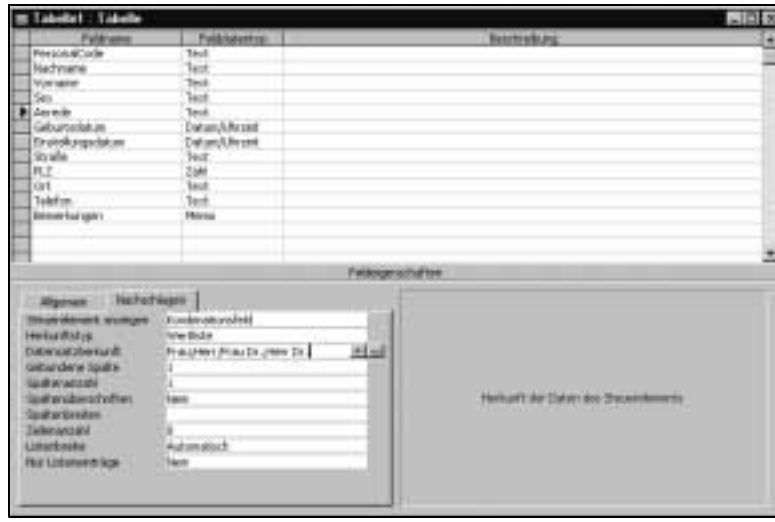


Bild 5.9: Eigenschaften zur Einstellung eines Nachschlagefeldes, hier eines Kombinationsfeldes, für das Tabellenfeld »Anrede«

- Tragen Sie zur Eigenschaft *Datensatzherkunft* die vier vorgesehenen Anreden ein. Trennen Sie dabei jede Anrede (d.h. jeden Wert der Werteliste) vom anderen durch ein Semikolon, so daß die Werteliste in der folgenden Form angegeben wird:

  Frau;Herr;Frau Dr.;Herr Dr.

Geburtsdatum

- Wählen Sie bei den Feldeigenschaften ggf. wieder das Register *Allgemein*.
- *Eingabe erforderlich*. Stellen Sie die Eigenschaft *Eingabe erforderlich* auf den Wert *Ja* ein.

Einstellungsdatum

- *Standardwert*. Sie erhalten als Standardwert den 1. des aktuellen Monats, wenn Sie in das Bearbeitungsfeld für *Standardwert* die folgende Zeile eingeben:

  =Datum()-Tag(Datum())+1

  Das vorangestellte Gleichheitszeichen ist wichtig, damit der nachfolgende Ausdruck, der die beiden Access-Funktionen *Datum*() und *Tag*() enthält, ausgewertet wird. Der wiedergegebene Ausdruck ist folgendermaßen zu verstehen: Die Funktion *Datum*() gibt das Tagesdatum der PC-Uhr wieder. Da Sie mit Access-Datumswerten rechnen können (zum Datumskonzept vgl. genauer z.B. Kap. 39, *Ausdrücke, Funktionen und Operatoren*), wird vom Tagesdatum die Zahl des laufenden Tages im aktuellen Monat abgezogen, welche mit dem Funktionsausdruck *Tag(Datum())* ermittelt wird; für den 18.11.1999 beispielsweise wäre dies die Zahl 18. Damit man nicht haargenau daneben trifft (nämlich den Monatsletzten des vorangehenden Monats statt des Monatsersten des aktuellen Monats), muß noch die Zahl 1 addiert werden.

Straße

- *Feldgröße*. Um die Feldgröße auf 100 einzustellen, klicken Sie auf das Bearbeitungsfeld *Feldgröße* und geben den Wert 100 ein.

PLZ

- *Feldgröße*. Um die Feldgröße als *Long Integer* festzulegen, klicken Sie auf das Bearbeitungsfeld *Feldgröße* und wählen aus dem Dropdown-Listenfeld den Typ *Long Integer* aus. Allerdings sollte Access dies bereits als voreingestellten Wert angeboten haben.
- *Eingabeformat*. Schreiben Sie in das Bearbeitungsfeld *Eingabeformat* fünfmal die Ziffer 0, also

  00000

  Dieser Ausdruck stellt sicher, daß nur Werte zwischen 00000 und 99999, die Grenzen eingeschlossen, akzeptiert werden.
- *Standardwert*. Löschen Sie den Voreinstellungswert 0, so daß das Bearbeitungsfeld leer ist.

Ort

- *Feldgröße*. Um die Feldgröße auf 100 einzustellen, klicken Sie auf das Bearbeitungsfeld *Feldgröße* und geben den Wert 100 ein.

Telefon

- *Feldgröße*. Um die Feldgröße auf 30 einzustellen, klicken Sie auf das Bearbeitungsfeld *Feldgröße* und geben den Wert 30 ein.

Sämtliche bisher besprochenen Punkte zur Tabellendefinition sind jetzt erledigt, denn alle Feldnamen, Felddatentypen und Feldeigenschaften wurden festgelegt. Im Prinzip könnten Sie nun mit der Dateneingabe beginnen. Wenn Sie aber in die dafür erforderliche Datenblattansicht wechseln würden, würden Sie von Access zum Speichern der Tabelle aufgefordert werden, denn Daten können nur eingegeben werden, wenn die letzten Änderungen, die in der Entwurfsansicht einer Tabelle vorgenommen wurden (also Änderungen, welche die Tabellenstruktur betreffen), gespeichert wurden, vgl. Bild 5.10.

Bild 5.10: Access blendet diese Meldung regelmäßig ein, wenn Sie nach Änderungen der Tabelle in der Entwurfsansicht in die Datenblattansicht wechseln wollen, ohne die Änderungen zuvor gespeichert zu haben.

Im vorliegenden Falle, in dem die Tabelle ja noch nie gespeichert wurde, kommt noch ein zweiter Punkt hinzu: Access empfiehlt dringend, für jede Tabelle einen *Primärschlüssel* festzulegen. Wenn Sie dies nicht getan haben, werden Sie vor dem erstmaligen Speichern gefragt, ob ein solcher von Access automatisch erstellt werden soll, vgl. die in Bild 5.11 wiedergegebene Meldung.

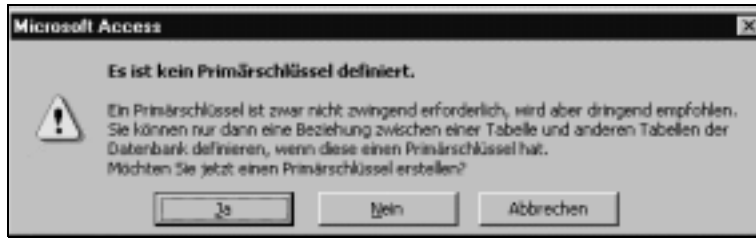


Bild 5.11: Wenn Sie eine Tabelle speichern wollen, für die kein Primärschlüssel definiert ist, bietet Access an, einen solchen automatisch zu erstellen.

Da unsere Personaltabelle aber ohnehin einen Primärschlüssel benötigt, soll dieser vor dem Speichern von uns festgelegt werden.

5.4.4 Primärschlüssel setzen und Tabelle speichern

Primärschlüssel setzen

Wenn Sie einem Feld den Primärschlüssel zuweisen, sorgt Access dafür, daß jeder Datensatz in diesem Feld einen eindeutigen Wert besitzt, d.h. in der gesamten Tabelle in diesem Feld nur einmal vorkommt. Anders ausgedrückt: Für das Feld mit dem Primärschlüssel können zwei Datensätze nicht denselben Wert haben. Der Primärschlüssel macht die Verwaltung einer Tabelle effizienter, Sie sollten daher im allgemeinen ein Feld als Primärschlüssel vorsehen. Wenn Sie zwischen zwei Tabellen eine 1:n-Beziehung definieren wollen, *muß* die Tabelle der 1-Seite (die sogen. Mastertabelle) einen Primärschlüssel besitzen, und zwar für das Feld, das die Verknüpfung (den Schlüssel) zur Tabelle der n-Seite herstellt. Im vorliegenden Beispiel ist dies das Feld *PersonalCode*. Setzen Sie daher den Primärschlüssel für das Feld *PersonalCode*. Gehen Sie dazu folgendermaßen vor:

- Geben Sie dem Feld *PersonalCode* den Fokus, indem Sie auf irgendeine Zelle der entsprechenden Zeile klicken.
- Klicken Sie auf die Symbol-Schaltfläche für *Primärschlüssel*, dem kleinen Schlüssel in der Symbolleiste. Danach sollte die Zeile des Feldes *PersonalCode* mit einem Schlüssel gekennzeichnet sein, vgl. die folgende Abbildung.

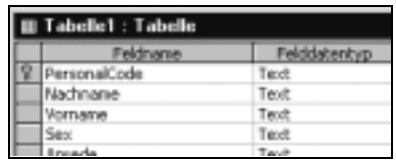


Bild 5.12: »Primärschlüssel« wurde eingefügt.

Falls Sie den Primärschlüssel versehentlich einer falschen Zeile zugewiesen haben, brauchen Sie diese Zuweisung nicht eigenständig aufzuheben, bevor Sie den Primärschlüssel der richtigen Zeile zuweisen: Geben Sie der Zeile *PersonalCode* den Fokus und klicken Sie auf die Symbol-Schaltfläche *Primärschlüssel*. Dann verliert das Feld, das zuvor den Primärschlüssel besaß, denselben, und das neue bekommt ihn zugewiesen. Dies liegt daran, daß jeweils nur ein Feld einen eigenen Primärschlüssel besitzen darf (obgleich mehrere Felder *zusammen einen* Primärschlüssel bilden können, vgl. genauer Kap. 8, *Tabellen: Grundlagen*).

Tabelle speichern

Die Personaltabelle soll unter dem Namen *Personal* gespeichert werden. Gehen Sie dazu folgendermaßen vor:

- Wählen Sie den Befehl *Speichern* aus dem Menü *Datei,* oder drücken Sie die Tastenkombination [Strg]+[S] oder klicken Sie auf die Symbol-Schaltfläche *Speichern* .
- Geben Sie in das Dialogfeld *Speichern unter* den Namen *Personal* ein, und bestätigen Sie mit OK. Danach erscheint der neue Name auch in der Titelleiste des Tabellenfensters. Ab jetzt wird die Tabelle *Personal* in der Objektliste für *Tabellen* im Datenbankfenster (und anderen derartigen Listen) aufgeführt.

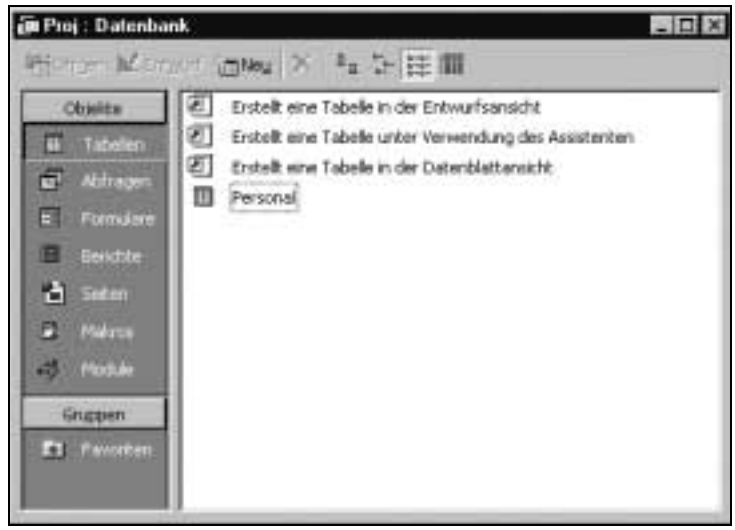


Bild 5.13: Die Tabelle wurde unter dem Namen »Personal« gespeichert und wird seitdem im Datenbankfenster als Tabellenobjekt angezeigt.

5.5 Personaldaten in Tabelle eingeben

Obwohl in den weiteren Punkten dieses Kapitels für die Beispieldatenbank Formulare entwickelt werden, die eine bequeme Dateneingabe ermöglichen, sollten Sie dennoch einmal probieren, wie die Dateneingabe ohne Formular, also direkt in die Tabelle, vor sich geht. Zu diesem Zweck müssen Sie von der Entwurfsansicht in die Datenblattansicht der Tabelle *Personal* wechseln.

In die Datenblattansicht wechseln Sie folgendermaßen:

- Klicken Sie auf das Symbol für *Datenblattansicht* in der Symbolleiste, oder wählen Sie den Befehl *Datenblattansicht* aus dem Menü *Ansicht*. Falls Sie seit der letzten Speicherung Änderungen am Tabellenentwurf vorgenommen haben – und sei es nur versehentlich –, würden Sie vor dem Wechsel in die Datenblattansicht zum Speichern der Tabelle aufgefordert werden, vgl. oben Bild 5.10. Falls die Tabelle gar nicht geöffnet ist, können Sie diese unmittelbar aus dem Datenbankfenster öffnen, indem Sie auf das Symbol für die Tabelle *Personal* doppelklicken oder auf die Schaltfläche *Öffnen*.

Bild 5.14: Datenblattansicht der Tabelle »Personal«, in die noch kein Datensatz eingegeben wurde, die jedoch alle Festlegungen für Feldnamen, Datentypen und Feldeigenschaften aufweist.

Nach dem Wechsel in die Datenblattansicht sollte Ihr Tabellenfenster aussehen wie in Bild 5.14 dargestellt: Die Tabelle ist im Prinzip leer, d.h. ohne einen Datensatz. Dieser Feststellung scheint zu widersprechen, daß eine Datensatz-Zeile zu sehen ist und daß am linken unteren Tabellenrand zwischen den Navigationsschaltflächen die Information *Datensatz: 1 von 1* angezeigt wird. Gleichwohl enthält die Tabelle noch keinen Datensatz. Dies

erkennen Sie u.a. am Inhalt des Feldes *PersonalCode*: Für dieses Feld ist ja der Primärschlüssel festgelegt, und Primärschlüsselfelder dürfen nicht leer sein, wenn ein Datensatz gespeichert wird. Für das Feld *Einstellungsdatum* wird in der einzigen Tabellenzeile der über einen Ausdruck als Standardwert festgelegte Datumswert angezeigt, in Bild 5.14 ist dies das Datum 01.06.99.

In der Datenblattansicht können Sie neue Datensätze eingeben, vorhandene Werte betrachten oder bearbeiten. Die Tabelle *Personal* soll am Ende des Einführungsbeispiels die Angaben zu sechs Mitarbeitern enthalten. Diese Angaben finden Sie in der nachfolgenden Tabelle 5.1. (Es steht Ihnen natürlich frei, andere Informationen als die in Tabelle 5.1 angeführten einzugeben, dann werden jedoch Ihre Bildschirmansichten im folgenden nicht immer mit den hier wiedergegebenen übereinstimmen.)

Inhalt der Tabelle *Personal*

Personal-Code	Nachname	Vorname	Sex	Anrede	Geburtsdatum	Einstellungsdatum
MAHLM	Mahlmann	Hermann	m	Herr	02.02.1958	01.01.1996
LAUSE	Lauser	Hannelore	w	Frau Dr.	02.11.1943	01.05.1990
BIEDE	Biedermann	Sieglinde	w	Frau	22.06.1963	01.07.1983
THIED	Thiedemann	Klaus	m	Herr	12.04.1941	01.09.1970
LEJEU	Lejeune	Micaela	w	Frau	24.12.1958	01.11.1995
GERSD	Gersdorff	Hubertus	m	Herr Dr.	04.06.1963	01.08.1985

Inhalt der Tabelle *Personal*

Personal-Code	Straße	PLZ	Ort	Telefon	Bemerkungen
MAHLM	Fritz-Reuter-Weg 9a	25497	Prisdorf	04101/3 44 21	Herr Mahlmann war vorher 10 Jahre lang bei der KRAWAG beschäftigt. Auf Empfehlung von Dr. Meierbeer zu uns gekommen.
LAUSE	Fröbelstr. 12	20146	Hamburg	040/12 34 77	

Personal-Code	Straße	PLZ	Ort	Telefon	Bemerkungen
BIEDE	Schillerstr. 7	27374	Visselhövede	04262/662	
THIED	Tallinner Str. 1	19063	Schwerin	0389/19 75 44	Herr Thiedemann ist sportinvalider Schwerbehinderter (65%). Ihm stehen gem. Betriebsvereinbarung jährlich 5 Tage Sonderurlaub zu.
LEJEU			Hoisdorf		Anschrift vervollständigen!
GERSD	Weidenstieg 89	20259	Hamburg	040/41 72 39	

Geben Sie jetzt den ersten Mitarbeiter-Datensatz, d.h. die Angaben zu *Hermann Mahlmann*, in die Tabelle *Personal* ein:

- Klicken Sie in die Zelle der Spalte *PersonalCode*, und schreiben Sie den *PersonalCode MAHLM*. Sobald Sie zu schreiben beginnen, erzeugt Access eine neue Datensatzzeile, in die Sie den zweiten Datensatz eingeben könnten. Die aktuelle, in der Bearbeitung befindliche Zeile ist am linken Rand mit einem Bleistift gekennzeichnet, vgl. links nebenstehend sowie Bild 5.15. Dieses Symbol zeigt an, daß der betreffende Datensatz noch nicht gespeichert wurde, sondern noch bearbeitet wird. Solange dieses Bleistiftsymbol noch angezeigt wird, können Sie alle Bearbeitungen des Datensatzes durch (ggf. mehrfaches) Drücken der [Esc]-Taste wieder rückgängig machen.

Bild 5.15: Der erste Datensatz wird bearbeitet. Beachten Sie das Bleistiftsymbol am linken Rand des in Bearbeitung befindlichen Datensatzes, das diesen Modus anzeigt.

- Beachten Sie, daß Sie die Eingabe in das Feld *PersonalCode* auch in Kleinbuchstaben vornehmen können: Access wandelt die Zeichen in die entsprechenden Großbuchstaben um. Dies ist auf das von uns für das Feld *PersonalCode* festgelegte Eingabeformat >LLLLL zurückzuführen. Dieses Eingabeformat ist auch dafür verantwortlich, daß Sie als *PersonalCode* exakt fünf Zeichen eingeben müssen

- Klicken Sie in die Zelle der Spalte *Nachname*, und geben Sie den Nachnamen ein.
- Verfahren Sie entsprechend für die weiteren Felder des ersten Datensatzes. Für die Anrede können Sie sich einen Eintrag aus der Dropdown-Liste wählen.

5.5.1 Eingabe rückgängig machen, automatisch speichern lassen oder Datensatz löschen

Falls Sie die Eingabe in ein Feld oder die Eingaben für den ganzen Datensatz unwirksam machen wollen, erreichen Sie dies durch (ggf. mehrfaches) Drükken der [Esc]-Taste. Umgekehrt: Der eingegebene Datensatz wird von Access automatisch gespeichert, wenn Sie seine Zeile mit dem Cursor verlassen. Praktisch bedeutet dies im allgemeinen, auf eine andere Datensatzzeile zu klicken. Dann wird der Datensatz der davor bearbeiteten Datensatzzeile auf der Festplatte gespeichert. Solange eine Änderung noch nicht gespeichert wurde, erscheint am linken Rand der in Bearbeitung befindlichen Datensatzzeile das bereits oben dargestellte Bleistiftsymbol, vgl. Bild 5.15. Nach dem Speichern eines Datensatzes können Sie einzelne Felder eines Datensatzes jederzeit wieder bearbeiten.

Sie können auch den ganzen Datensatz löschen:

- Markieren Sie den zu löschenden Datensatz, indem Sie auf das kleine graue Kästchen am linken Rand der Datensatzzeile klicken. Zum Markieren können Sie auch den Befehl *Datensatz auswählen* aus dem Menü *Bearbeiten* ausführen, der den Datensatz, der gerade den Fokus hat, markiert. Wenn Sie eine Datensatzzeile markiert haben, sind alle ihre Zellen markiert, werden also mit schwarzem (statt hellem) Hintergrund dargestellt.

Bild 5.16: Der erste Datensatz wurde markiert und kann jetzt z. B. gelöscht oder kopiert werden.

- Drücken Sie die [Entf]-Taste oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*. Bestätigen Sie die von Access zur Sicherung gegen versehentliches Löschen eingeblendete Warnmeldung.

5.5.2 Verstoß gegen eine Gültigkeitsregel, ein Eingabeformat und *Eingabe erforderlich*

Verstoß gegen Gültigkeitsregel

Verstoßen Sie aus Testgründen einmal gegen die Gültigkeitsregel des Feldes *Sex*, indem Sie dort beispielsweise den Buchstaben *a* eingeben. Sobald Sie das Feld nach der Eingabe verlassen, sollte der von Ihnen für diesen Fall vorgesehene Text *Geben Sie m oder w ein!* auf dem Bildschirm eingeblendet werden, vgl. Bild 5.17. Bestätigen Sie dann diese Meldung mit OK und korrigieren Sie den falschen Wert. Sie kommen nur weiter, wenn Sie einen der Gültigkeitsregel entsprechenden oder gar keinen Wert eingeben – letzteres jedoch auch nur, wenn für das betreffende Feld die Eigenschaft *Eingabe erforderlich* auf den Wert *Nein* eingestellt ist, was für das Feld *Sex* nicht zutrifft. Um die Eingabe eines unzulässigen Wertes abzubrechen, drücken Sie die Esc-Taste.

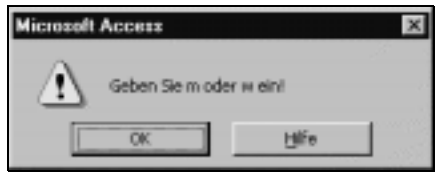


Bild 5.17: Diese benutzerdefinierte Gültigkeitsmeldung wird eingeblendet, weil bei der Eingabe eines Wertes für das Feld »Sex« gegen die Gültigkeitsregel verstoßen wurde.

Verstoß gegen Eingabeformat

Für das Feld *PLZ* haben Sie das Eingabeformat 00000 festgelegt. Dies bedeutet, daß Sie, *wenn* Sie eine Postleitzahl angeben, diese als fünfstellige Zahl eingeben müssen. (Da die Eigenschaft *Eingabe erforderlich* für dieses Feld auf *Nein* eingestellt ist, brauchen Sie indes nicht unbedingt einen Wert einzugeben.) Wenn Sie beispielsweise nur eine vierstellige Zahl eingeben und dann den Fokus auf ein anderes Feld oder eine andere Datensatzzeile setzen, erscheint die in Bild 5.18 wiedergegebene Meldung.

Bild 5.18: Meldung über den Verstoß gegen das festgelegte Eingabeformat für das Feld »PLZ«

Verstoß gegen *Eingabe erforderlich*

Für mehrere Felder der Tabelle *Personal* haben Sie die Eigenschaft *Eingabe erforderlich* auf den Wert *Ja* eingestellt, beispielsweise auch für das Feld *Geburtsdatum*. Wenn ein Datensatz gespeichert werden soll, ohne daß Sie für dieses Feld einen zulässigen Wert eingegeben haben, erscheint die in Bild 5.19 wiedergegebene Meldung. Der Hinweis auf den *Null-Wert* erklärt sich folgendermaßen: Leere Felder eines Datensatzes bekommen den speziellen Wert *Null*. Wenn Sie die Eigenschaft *Eingabe erforderlich* (*Required*) auf den Wert *Ja* eingestellt haben, ist dies gleichbedeutend mit der Anweisung, daß das betreffende Feld keinen Null-Wert enthalten darf.

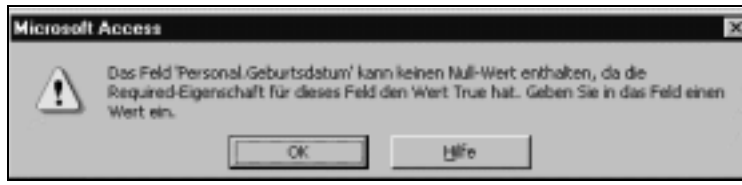


Bild 5.19: Für das Feld »Geburtsdatum« der Tabelle »Personal« ist die Eigenschaft »Eingabe erforderlich« (»Required«) auf »Ja« eingestellt. Es wurde versucht, einen Datensatz zu speichern, ohne daß in dieses Feld ein Wert eingegeben wurde.

5.5.3 Unerwartete Schwierigkeiten bei der Dateneingabe

Mit einer nicht geringen Wahrscheinlichkeit wird bei der Dateneingabe nicht alles so klappen, wie Sie es nach der Lektüre der vorliegenden Abschnitte erwarten. Kompliziert und in ihrer Wirkung sehr restriktiv können vor allem die Gültigkeitsregeln, Datentypen und Eingabeformate sein. Deren prinzipiell erwünschte Einschränkungen für die Dateneingabe können sich fatal auswirken, wenn Sie eine Gültigkeitsregel oder ein Eingabeformat fehlerhaft angegeben haben. In solchen Fällen gehen Sie zur Beseitigung der Schwierigkeit oder unerwarteten Effekts bei der Dateneingabe folgendermaßen vor:

- Wechseln Sie zur Entwurfsansicht der Tabelle, indem Sie auf das entsprechende Symbol klicken oder den Befehl *Entwurfsansicht* aus dem Menü *Ansicht* wählen. Sollte dies nicht sofort möglich sein, weil etwas »hakt«, drücken Sie, ggf. mehrmals, die Esc-Taste. Bei geschlossener Tabelle können Sie diese auch direkt aus dem Datenbankfenster in der Entwurfsansicht öffnen: Markieren Sie das Symbol für die Tabelle *Personal* in der Objektliste *Tabellen*, und klicken Sie dann auf die Schaltfläche *Entwurf*.
- Geben Sie der Zeile des Feldes, welches die Schwierigkeiten bei der Dateneingabe bereitete, den Fokus, indem Sie auf eine seiner Zellen klicken.
- Prüfen Sie, ob der Felddatentyp und die Feldeigenschaften, vor allem die Gültigkeitsregel und das Eingabeformat, den Angaben bzw. Ihren Wünschen entsprechen, und korrigieren Sie entsprechend.

- Wechseln Sie nach erfolgter Korrektur wieder in die Datenblattansicht. Wenn Sie in der Entwurfsansicht eine Änderung vorgenommen haben, werden Sie von Access aufgefordert, die Tabelle beim Wechsel zur Datenblattansicht zu speichern. Bestätigen Sie die Aufforderung, sonst gelangen Sie nicht in die Datenblattansicht.
- Versuchen Sie erneut, in das Feld, bei dem Sie Schwierigkeiten feststellten, einen Wert einzugeben.

Wenn Sie eine Tabelle neu konzipieren oder später strukturelle Änderungen an ihr vornehmen, ist ein ständiger Wechsel zwischen Datenblatt- und Entwurfsansicht normaler Bestandteil der Arbeit. (Entsprechendes gilt auch für das Entwerfen von Formularen, Abfragen und Berichten.)

5.5.4 Tabellenfenster schließen

Bevor Sie die Projekttabelle definieren, sollten Sie aus Gründen der Übersichtlichkeit die Tabelle *Personal* schließen. Gehen Sie dazu folgendermaßen vor:

- Sorgen Sie ggf. dafür, daß das Fenster der Tabelle *Personal* aktiviert ist. Sie aktivieren es z.B. über das Menü *Fenster.*

- Klicken Sie auf die Symbol-Schaltfläche *Schließen*, vgl. links nebenstehend. Alternativ: Wählen Sie im Menü *Datei* den Befehl *Schließen*. Falls die Tabelle bisher noch nicht gespeicherte Änderungen am Tabellenentwurf enthält, werden Sie gefragt, ob Sie die Tabelle speichern wollen, vgl. z.B. die Meldung oben in Bild 5.10. Änderungen der Daten in der Datenblattansicht werden dagegen stets automatisch gespeichert, so daß Sie insoweit nie entsprechend zum Speichern aufgefordert werden.

5.6 Tabelle *Projekte definieren*

5.6.1 Tabellenfelder

Nach den ausführlichen Erklärungen zum Entwurf der Tabelle *Personal* sollen die Ausführungen bezüglich der Tabellenfelder für die entsprechende Arbeit zur Tabelle, die die Projektdaten speichert, relativ knapp gehalten werden.

Für die Tabelle *Projekte* sollen die Feldnamen, Felddatentypen und Feldeigenschaften festgelegt werden, wie sie in der Übersicht der nachfolgenden Tab. 5.2 angegeben sind.

Tabellenfelder

Feldname	Felddatentyp	Feldeigenschaften
ProjektNr	AutoWert	
PersonalCode	Text	*Feldgröße*: 5 *Eingabe erforderlich*: Ja Nachschlagefeld als Kombinationsfeld
Projektbezeichnung	Text	*Feldgröße: 255* *Eingabe erforderlich*: Ja
Auftragswert	Währung	*Standardwert*: 0; *Gültigkeitsregel*: >= 0
Projektbeginn	Datum/ Uhrzeit	*Eingabe erforderlich*: Ja
Projektabschluß	Datum/ Uhrzeit	
Storniert	Ja/Nein	*Standardwert*: Nein

Tabelle 5.2: Feldnamen, Datentyen und Feldeigenschaften der Tabelle »Projekte«

Die sieben Felddefinitionen für die Projekttabelle ergeben sich weitgehend von selbst, wenn Sie die Vorgaben berücksichtigen, die dafür oben im Punkt 5.1, *Problemstellung*, zu Beginn dieses Kapitels formuliert wurden. Daher seien nur wenige Punkte hervorgehoben.

Fremdschlüsselfeld *PersonalCode*. Entsprechend unseren obigen Überlegungen zum Design der Datenbank *Projekte* sollen die beiden Tabellen *Personal* und *Projekte* durch eine 1:n-Beziehung miteinander verknüpft werden (vgl. Punkt 5.2, *Datenbankdesign*). Für diesen Zweck ist es erforderlich, daß das Feld, welches in der Mastertabelle (der 1-Seite) *Personal* als Primärschlüssel definiert ist, in der Detailtabelle (der n-Seite) *Projekte* ebenfalls erscheint, als sogen. *Fremdschlüssel*. Im vorliegenden Beispiel ist dies das Feld *PersonalCode*. Das Fremdschlüsselfeld muß vom selben Datentyp mit derselben Feldgröße sein wie das entsprechende Primärschlüsselfeld.

***AutoWert*.** Für ein Feld mit diesem Datentyp erzeugt Access für jeden neuen Datensatz eine fortlaufende ganze Zahl, beginnend mit 1 für den ersten Datensatz. Der Benutzer kann in ein Feld mit dem Datentyp *AutoWert* keinen Wert eingeben noch kann er ihn verändern. Er kann sich dessen Werte nur anschauen. Access sorgt bei einem Feld vom Typ *AutoWert* automatisch dafür, daß derselbe Wert in diesem Feld nur einmal vorkommt. Weil für die Projektnummern Eindeutigkeit sehr wichtig ist, andererseits auf bestimmte Projektnummern hier kein Wert gelegt wird, wurde die automatisch eindeutige Vergabe einer Projektnummer mit dem Festlegen des Datentyps *AutoWert* dem System Access übertragen.

Felddatentyp *Ja/Nein*. In ein Feld dieses Datentyps können Sie nur *Ja* oder *Nein* eingeben. (Zulässig sind auch die Werte WAHR und FALSCH oder -1 und 0). Da ein Projekt nur den Zustand *storniert* oder *nicht storniert* haben

kann, empfiehlt sich dieser Datentyp für das Feld *Storniert*. Der Felddatentyp *Ja/Nein* spart nicht nur Speicherplatz, sondern er erlaubt in Formularen auch die Verwendung eines ein- und ausschaltbaren Kontrollkästchens, was zur einfachen und sicheren Dateneingabe beiträgt.

Gültigkeitsregel für Auftragswert. Negative Auftragswerte sollen nicht erlaubt sein. Geben Sie als Gültigkeitsregel einfach die Zeile

>=0

ein. *Hinweis*: Anders als in mathematischer Notation üblich wird der Vergleichsoperator größer/gleich in Access, wie auch in anderen PC-Programmen, nicht als das eine Zeichen ≥, sondern als die *zwei* aufeinander folgenden Zeichen >= eingegeben.

Standardwert. Geben Sie die beiden vorgesehenen Standardwerte einfach so ein, wie sie in Tab. 5.2 angegeben sind: Schreiben Sie in das Bearbeitungsfeld für den Standardwert von *Auftragswert 0* und für denjenigen von *Storniert Nein*.

Nachschlagefeld. Für das Feld *PersonalCode* soll ein *Nachschlagefeld* als Kombinationsfeld festgelegt werden. Das Feld *PersonalCode* fungiert in der Tabelle *Projekte* als Fremdschlüssel, darf also nur Werte enthalten, die auch im Primärschlüsselfeld *PersonalCode* der Mastertabelle *Personal* vorkommen. Bei der Eingabe von Werten in das Feld *PersonalCode* müssen dem Benutzer also die Werte des Primärschlüsselfeldes der Mastertabelle bekannt sein, wenn die Dateneingabe in ein Datenblatt der Tabelle erfolgt. Dieses Problem kann mit einem Nachschlagefeld gelöst werden: Dieses kann in einer Liste alle Werte des Primärschlüsselfeldes der Mastertabelle anbieten, so daß einer davon zur Eingabe in das Fremdschlüsselfeld der Detailtabelle gewählt werden kann.

5.6.2 Gültigkeitsregel für die Tabelle

Sie haben Gültigkeitsregeln (und die zugehörigen Gültigkeitsmeldungen) als Eigenschaften einzelner Tabellenfelder kennengelernt. Diese Regeln sind sehr leistungsfähig, denn sie können als beinahe beliebig komplexe Ausdrücke formuliert werden. Sie unterliegen jedoch einer wichtigen Beschränkung: Ein Ausdruck, der als Gültigkeitsregel für ein Tabellenfeld angegeben wird, darf sich nicht auf ein anderes Feld beziehen. Damit ist ein Gültigkeitsvergleich zweier Felder miteinander auf diese Weise nicht möglich. In der Tabelle *Projekte* soll jedoch die Gültigkeit von Werten zweimal in Abhängigkeit vom Wert, der für ein anderes Feld eingegeben wurde, überprüft werden:

1. Wenn ein Projekt storniert ist (das Feld *Storniert* hat dann für den betreffenden Datensatz den Wert *Ja*), darf kein Datum für den *Projektabschluß* eingegeben werden, weil dies widersprüchlich wäre.
2. Das Datum für den *Projektabschluß* darf niemals vor demjenigen von *Projektbeginn* liegen.

Eine Gültigkeitsregel, die Werte von Tabellenfeldern miteinander vergleicht, kann nur als *Tabellen*eigenschaft *Gültigkeitsregel* angegeben werden. Diese Regel wird überprüft, wenn ein Datensatz gespeichert werden soll. Die *Gültigkeitsregel* eines Tabellen*feldes* wird dagegen bereits überprüft, wenn Sie einen Wert in ein Tabellenfeld eingegeben haben und einem anderen Feld desselben Datensatzes den Fokus geben, ohne daß der Datensatz schon gespeichert werden müßte. Es ist nicht schwer einzusehen, warum Access eine Gültigkeitsregel, die Tabellenfelder miteinander vergleicht, nur als Tabelleneigenschaft zuläßt und die Überprüfung erst beim Speichern des Datensatzes vornimmt: Ein Vergleich zweier oder mehrerer Felder kann erst vorgenommen werden, wenn sämtliche Werte der betreffenden Felder eingegeben wurden, und dieses trifft erst dann mit Sicherheit zu, wenn der gesamte Datensatz gespeichert werden soll.

Tabelleneigenschaften

Eigenschaft	Einstellung
Beschreibung	Enthält Projektdaten
Gültigkeitsregel	(([Storniert] Und IstNull([Projektabschluß])) Oder [Storniert]=Falsch) Und ([Projektabschluß]>=[Projektbeginn] Oder IstNull([Projektabschluß]))
Gültigkeitsmeldung	Wenn Sie ein Projekt storniert haben, darf kein Datum für den Projektabschluß angegeben werden! Oder: Das Datum für den Projektabschluß darf nicht vor dem Datum des Projektbeginns liegen!

Tabelle 5.3: Tabelleneigenschaften der Tabelle »Projekte«

Die oben unter a) und b) formulierten Gültigkeitsanforderungen, die im Grunde jede für sich eine eigene Regel darstellen, müssen als *eine* Gültigkeitsregel formuliert werden, denn Sie können die Gültigkeitsregel als Tabelleneigenschaft nur mit *einem* Ausdruck einstellen. In Tab. 5.3 ist der entsprechende sehr komplexe Ausdruck angegeben, der dieses leistet. Darüber hinaus finden Sie dort auch einen Textvorschlag für die Tabellenbeschreibung und die Gültigkeitsmeldung.

5.6.3 Projekttabelle praktisch erstellen

Tabellenfelder definieren

Um die Projekttabelle zu definieren, verfahren Sie zunächst völlig analog zum Vorgehen bei der Tabelle *Personal*. Dort können Sie im Zweifel Einzelheiten nachlesen. Definieren Sie zunächst die Felder mit Namen, Datentypen und Feldeigenschaften entsprechend den Angaben oben in Tab. 5.2. Im folgenden gebe ich nur die großen Schritte an, die Sie zu berücksichtigen haben:

- Erstellen Sie ein neues Datenbankobjekt *Tabelle*. Dies können Sie mit Hilfe der Symbol-Schaltfläche *Neues Objekt* – einer Dropdown-Liste, vgl. links nebenstehend – oder vom Datenbankfenster aus bewerkstelligen. Für den letzteren Weg bringen Sie das Datenbankfenster in den Vordergrund (Taste F11 oder über das Menü *Fenster*) klicken Sie im Datenbankfenster auf das Objektsymbol *Tabellen*, um dieses zu aktivieren, und klicken dann auf das Symbol *Erstellt eine neue Tabelle in der Entwurfsansicht*. Sie erhalten eine neue Tabelle in der Entwurfsansicht.
- Geben Sie die Feldnamen ein.
- Legen Sie die Felddatentypen fest.
- Legen Sie die Feldeigenschaften fest.
- Weisen Sie dem Feld *ProjektNr* den Primärschlüssel zu.

Nachschlagefeld erstellen

Entsprechend der oben formulierten Aufgabe soll für das Feld *PersonalCode* ein Nachschlagefeld erstellt werden. Die Bedeutung von Nachschlagefeldern ist sehr weitreichend, daher wird an anderer Stelle auch ausführlich darauf eingegangen, vgl. Kap. 8, *Tabellen: Grundlagen*, Punkt 8.6, *Nachschlagefeld erstellen*. Hier werde ich mich aus Platzgründen sehr knapp fassen. Gehen Sie folgendermaßen vor, um für das Feld *PersonalCode* der Tabelle *Projekte* ein Nachschlagefeld zu erstellen:

- Geben Sie in der Entwurfsansicht der Tabelle *Projekte* dem Feld *PersonalCode* den Fokus.
- Aktivieren Sie im (unteren) Fensterteil *Feldeigenschaften* die Registerkarte *Nachschlagen*.
- Wählen Sie in der Dropdown-Liste zur Eigenschaft *Steuerelement anzeigen* den Eintrag *Kombinationsfeld*. Dann sehen die in der Registerkarte *Nachschlagen* angezeigten Eigenschaften aus, wie in Bild 5.20 wiedergegeben.

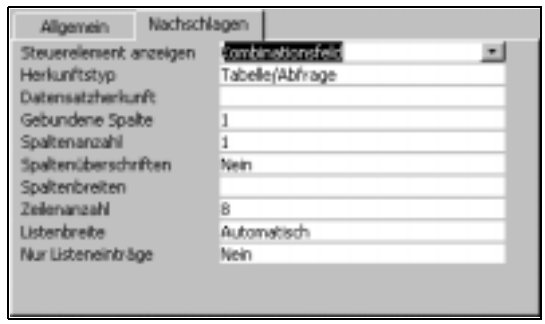


Bild 5.20: Voreinstellungen der Eigenschaften zu einem Kombinationsfeld-Nachschlagefeld

- Schlagen Sie die Dropdown-Liste zur Eigenschaft *Datensatzherkunft* auf, und wählen Sie darin die Tabelle *Personal*. Damit legen Sie fest, daß diese Tabelle die Datenquelle für die Werteliste des Nachschlagefeldes ist.
- Tragen Sie für die Eigenschaft *Spaltenanzahl* die Zahl 3 ein, damit die ersten drei Felder der Tabelle *Personal* als drei Spalten im Nachschlagefeld berücksichtigt werden.
- Belassen Sie es bei dem Wert 1 für die Eigenschaft *Gebundene Spalte*. Dies hat zur Folge, daß der Wert der ersten Spalte des Nachschlagefeldes an das Feld *PersonalCode* der Tabelle *Projekte* weitergegeben wird, während die anderen beiden Spalten zwar im Nachschlagefeld angezeigt, jedoch mit ihren Werten nicht gespeichert werden.
- Tragen Sie für die Eigenschaft *Listenbreite* die Zahl 10 ein, damit die aufgeschlagene Liste des Nachschlagefeldes 10 cm breit wird; dies ist in unserem Beispiel ausreichend, um die ersten drei Felder der Tabelle *Personal* gut lesbar in drei Spalten auszugeben.

Nach diesen Änderungen sollten die Einstellungen in der Registerkarte *Nachschlagen* für das Feld *PersonalCode* aussehen, wie in Bild 5.21 wiedergegeben.

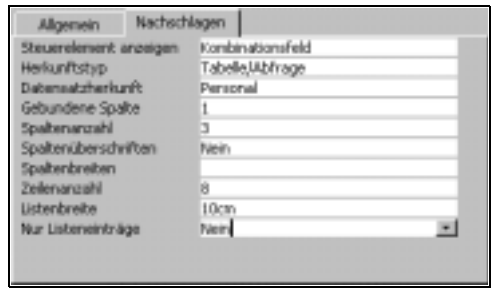


Bild 5.21: Eigenschaften-Einstellungen für das Nachschlagefeld zum Feld »PersonalCode« der Tabelle »Projekte«

Wenn Sie nach dieser Definition des Nachschlagefeldes in die Datenblattansicht der Tabelle *Projekte* wechseln, können Sie für jeden Datensatz des Feldes *PersonalCode* das Nachschlagefeld aufschlagen. Seine Werteliste bietet in drei Spalten die Werte der Felder *PersonalCode*, *Nachname* und *Vorname* der Tabelle *Personal* an, weil diese drei Felder die ersten in der Tabelle sind, vgl. Bild 5.21.

Tabelleneigenschaften einstellen

Die Tabelleneigenschaften werden im *Eigenschaftenfenster* eingestellt. Sie müssen es in der Entwurfsansicht der Tabelle einblenden, falls es nicht bereits sichtbar ist. Um das *Eigenschaftenfenster* der Tabelle einzublenden, gehen Sie folgendermaßen vor:

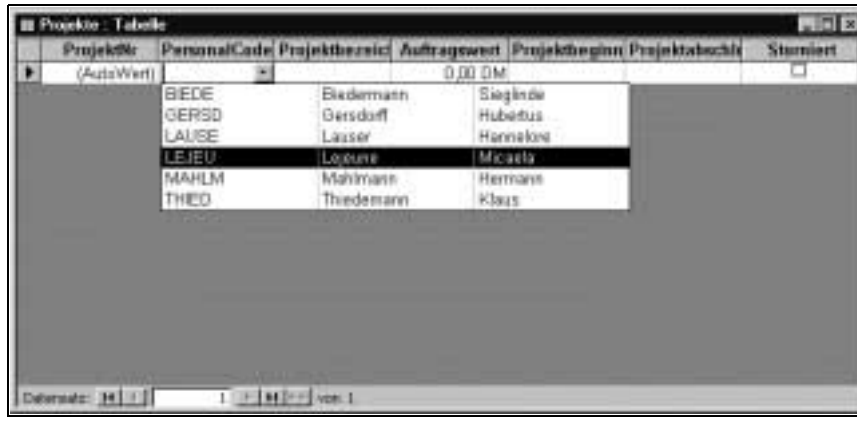


Bild 5.22: Aufgeschlagenes Nachschlagefeld für das Feld »PersonalCode«

- Klicken Sie auf die Symbol-Schaltfläche *Eigenschaften* (vgl. links nebenstehend) oder wählen Sie den Befehl *Eigenschaften* aus dem Menü *Ansicht*. Die Symbol-Schaltfläche bzw. der Befehl wirkt als Ein-/Ausschalter. Sie sehen das Eigenschaftenfenster der Tabelle, für deren Eigenschaften noch nichts eingetragen ist, in Bild 5.23.

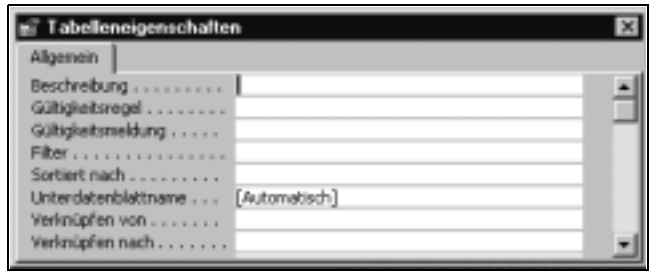


Bild 5.23: Eigenschaftenfenster einer Tabelle

Die oben in Tab. 5.3 angegebenen Tabelleneigenschaften geben Sie entsprechend wie die Feldeigenschaften ein. Wegen der sehr langen Zeichenfolgen für Gültigkeitsregel und Gültigkeitstext empfiehlt sich hier in besonderem Maße, mit dem Dialogfeld *Zoom* zu arbeiten. Sie öffnen dieses, während das betreffende Bearbeitungsfeld den Fokus hat, mit der Tastenkombination ⇧+F2.

Gültigkeitsregel für die Tabelle *Projekte*. Die Gültigkeitsregel geben Sie mit Hilfe des Dialogfeldes *Zoom* auf die folgende Weise ein:

- Klicken Sie in das Bearbeitungsfeld der Eigenschaft *Gültigkeitsregel* im Fenster *Tabelleneigenschaften*.
- Drücken Sie die Tastenkombination ⇧+F2. Dann öffnet sich das Dialogfeld *Zoom*.

- Geben Sie den Ausdruck für die Gültigkeitsregel in das Dialogfeld *Zoom* ein. Beachten Sie dabei sehr sorgfältig jedes Zeichen. Insbesondere dürfen runde und eckige Klammern (letztere erhalten Sie bei gedrückter [AltGr]-Taste) weder fortgelassen noch verwechselt werden. Den fertig eingegebenen Ausdruck sehen Sie im Dialogfeld *Zoom* von Bild 5.24.

Bild 5.24: Dialogfeld »Zoom« mit dem Ausdruck für die Gültigkeitsregel der Tabelle »Projekte«

Gültigkeitsmeldung für die Tabelle *Projekte*. Die Gültigkeitsmeldung geben Sie mit Hilfe des Dialogfeldes *Zoom* auf die folgende Weise ein:

- Klicken Sie auf das Bearbeitungsfeld der Eigenschaft *Gültigkeitsmeldung* im Fenster *Tabelleneigenschaften*.
- Drücken Sie die Tastenkombination [⇧]+[F2]. Dann öffnet sich das Dialogfeld *Zoom*.
- Geben Sie den Text für die Gültigkeitsmeldung in das Dialogfeld *Zoom* ein. Es empfiehlt sich, vor und nach dem Wort *oder* je zwei Zeilenschaltungen einzufügen, damit die beiden sachlich verschiedenen Meldungstexte gut voneinander unterscheidbar lesbar sind. Sie erzeugen eine Zeilenschaltung im Dialogfeld *Zoom* (wie auch allgemein in Bearbeitungsfeldern) mit der Tastenkombination [Strg]+[↵]. Das Dialogfeld *Zoom* mit dem umbrochenen Gültigkeitstext sehen Sie in Bild 5.25, die Meldung, die er ggf. erzeugt, in Bild 5.26.

Tabelle *Projekte speichern*. Wenn Sie alle Feld- und Tabellendefinitionen eingegeben haben, sollten Sie die Tabelle unter dem Namen *Projekte* speichern:

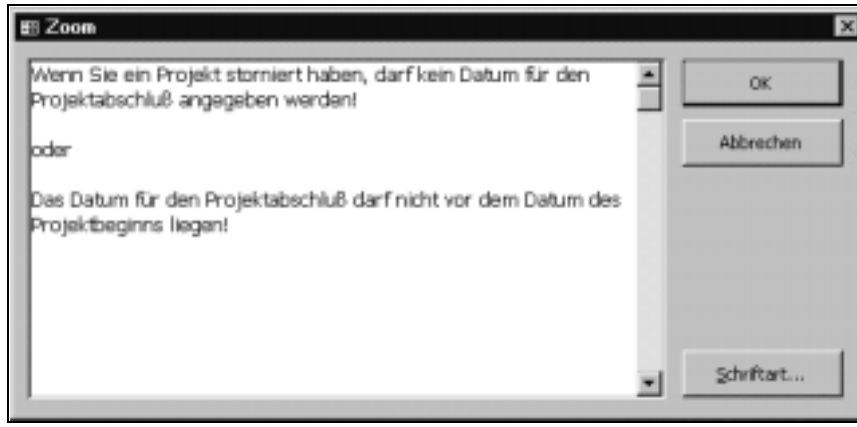


Bild 5.25: Dialogfeld »Zoom »mit dem Ausdruck für die Gültigkeitsmeldung der Tabelle »Projekte«. Die Zeilenschaltungen vor und nach dem Wort »oder« werden mit der Tastenkombination Strg+↵ erzeugt.

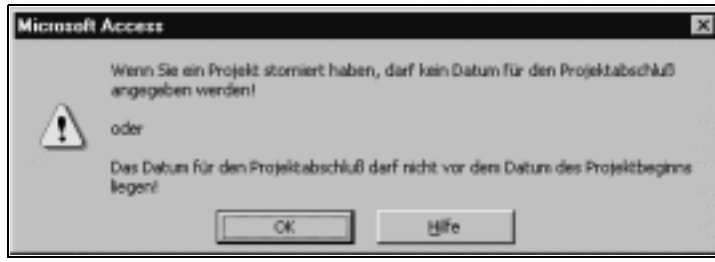


Bild 5.26: Gültigkeitsmeldung der Tabelle »Projekte« mit Zeilenschaltungen vor und nach dem Wort »oder«

- Geben Sie keinen Datensatz ein. Dies empfiehlt sich wegen der im folgenden Abschnitt festzulegenden Beziehung zwischen den beiden Tabellen: Für das Feld *PersonalCode* der Tabelle *Projekte*, das als Fremdschlüssel eine Verknüpfung zum Primärschlüssel *PersonalCode* der Tabelle *Personal* herstellen wird, dürfen dann nur Werte existieren, die bereits in *PersonalCode* von *Personal* existieren. Andernfalls würde Access sich weigern, eine entsprechende Beziehung festzulegen.
- Speichern Sie die Tabelle unter dem Namen *Projekte* mit der Tastenkombination Strg+S oder dem Befehl *Speichern* aus dem Menü *Datei*.
- Schließen Sie die Tabelle *Projekte*: Sorgen Sie ggf. dafür, daß sie aktiviert ist, und klicken Sie auf die Symbol-Schaltfläche *Schließen*, oder wählen Sie den Befehl *Schließen* aus dem Menü *Datei* oder dem *Systemmenü*.

5.7 Beziehung zwischen den Tabellen definieren

5.7.1 Praktisches Vorgehen

Bei den Überlegungen zum Datenbankdesign (vgl. vor allem den Punkt 5.2.2, *Soll eine Beziehung zwischen den Tabellen bestehen?*) wurde erklärt und entschieden, daß zwischen der Personal- und der Projekttabelle eine 1:n-Beziehung definiert werden soll. Diese Beziehung soll jetzt festgelegt werden. Gehen Sie dazu folgendermaßen vor:

- Schließen Sie ggf. die beiden Tabellen *Personal* und *Projekte*, denn zum Festlegen einer Beziehung darf keine der beteiligten Tabellen geöffnet sein, weder in der Datenblatt- noch in der Entwurfsansicht.
- Aktivieren Sie das Datenbankfenster (z.B. mit Taste F11).

- Wählen Sie den Befehl *Beziehungen...* aus dem Menü *Extras,* oder klicken Sie auf die Symbol-Schaltfläche *Beziehungen*, vgl. links nebenstehend. Wenn Sie diesen Befehl später, nachdem Sie bereits eine Beziehung festgelegt haben, erneut ausführen, wird sofort das Fenster *Beziehungen* eingeblendet werden. Da für diese Datenbank jedoch bisher keine Beziehung festgelegt wurde, blendet Access das Dialogfeld *Tabelle anzeigen* ein, damit Sie dem Fenster *Beziehungen* die beiden Tabellen hinzufügen können, vgl. Bild 5.27.

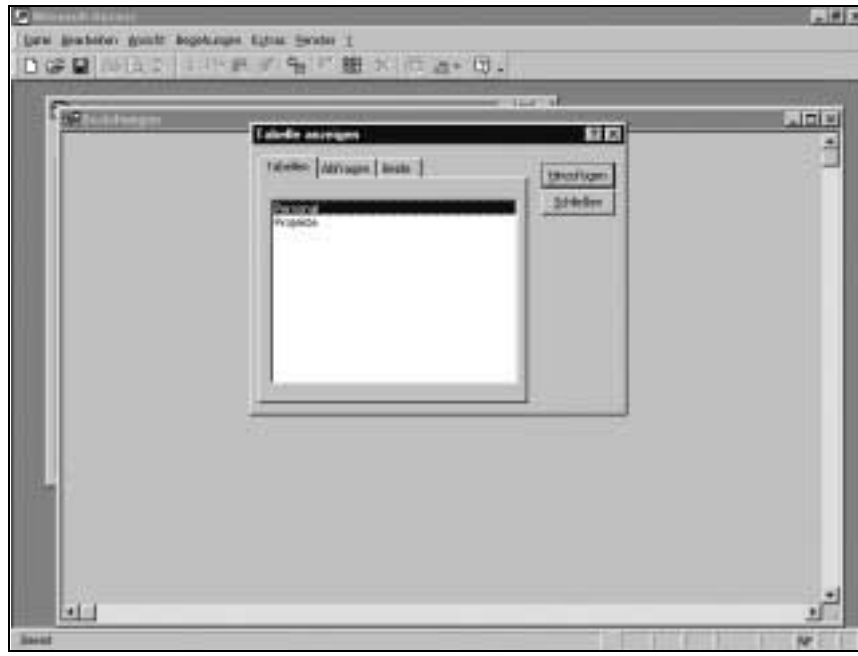


Bild 5.27: Fenster »Beziehungen« mit eingeblendetem Dialogfeld »Tabelle anzeigen«

- Fügen Sie dem Fenster *Beziehungen* die beiden Tabellen *Personal* und *Projekte* hinzu, indem Sie jeweils den entsprechenden Tabellennamen im Listenfeld des Dialogfeldes *Tabelle anzeigen* doppelklicken. Schließen Sie dann das Dialogfeld *Tabelle anzeigen*.

  Hinweis: Falls Sie das Dialogfeld *Tabelle anzeigen* versehentlich geschlossen haben, bevor Sie dem Fenster *Beziehungen* beide Tabellen hinzugefügt haben, können Sie es wieder einblenden. Wählen Sie dazu den Befehl *Tabelle anzeigen...* aus dem Menü *Beziehungen* oder klicken Sie auf die entsprechende Symbol-Schaltfläche, vgl. links nebenstehend.

  Nach dem Schließen des Dialogfeldes *Tabelle anzeigen* sollte das Fenster *Beziehungen* aussehen, wie in Bild 5.28 dargestellt. Beachten Sie, daß bisher nur Tabellen in das Fenster *Beziehungen* eingefügt, Beziehungen zwischen diesen jedoch noch nicht festgelegt wurden.

Bild 5.28: Fenster »Beziehungen« nach dem Einfügen der beiden Tabellen »Personal« und »Projekte«. Es wurden noch keine Beziehungen festgelegt. Dies erkennen Sie daran, daß zwischen den beiden Tabellen keine Verbindungslinie gezeichnet ist.

- Stellen Sie zwischen den beiden Tabellen eine Beziehung her. Ziehen Sie dazu im Fenster *Beziehungen* das Feld *PersonalCode* aus der Tabelle *Personal* auf das Feld *PersonalCode* in der Tabelle *Projekte*. Dann zeigt sich das Dialogfeld *Beziehungen*, vgl. Bild 5.29.

  Das Feld *PersonalCode* dient als Schlüssel, um die Beziehung zwischen den beiden Tabellen zu ermöglichen. Das Feld, welches als Schlüssel dient, muß für die Mastertabelle zwingend als *Primärschlüssel* definiert sein. Für die Detailtabelle wird das als Schlüssel fungierende Feld *Fremdschlüssel* genannt.

- Klicken Sie auf das Kontrollkästchen *Mit referentieller Integrität* (Bedeutung wird gleich erklärt), so daß es angekreuzt ist. Dann werden auch die bisher deaktivierten Steuerelemente im unteren Teil des Dialogfeldes aktiviert.

Bild 5.29: Dialogfeld »Beziehungen«, eingeblendet unmittelbar nach dem Ziehen des Feldes »PersonalCode« aus der Tabelle» Personal« auf das Feld »PersonalCode« der Tabelle »Projekte«

- Klicken Sie jeweils auf die Kontrollkästchen *Aktualisierungsweitergabe an Detailfeld* sowie *Löschweitergabe an Detaildatensatz* (Bedeutung wird auch hierfür gleich erklärt), so daß beide Kontrollkästchen angekreuzt erscheinen. Das Dialogfeld *Beziehungen* sollte jetzt so aussehen, wie in Bild 5.30 wiedergegeben.

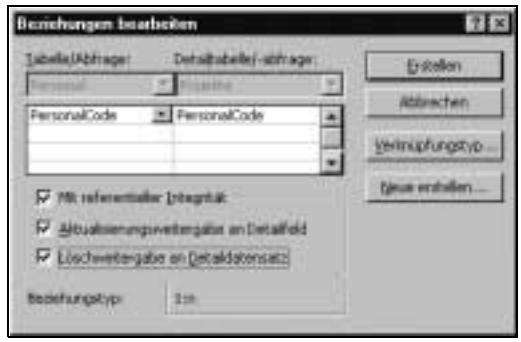


Bild 5.30: Dialogfeld »Beziehungen« nach dem Festlegen der referentiellen Integrität

- Klicken Sie auf die Schaltfläche *Erstellen*. Dann wird das Dialogfeld *Beziehungen* geschlossen, und das Fenster *Beziehungen* ist wieder sichtbar. Die festgelegte 1:n-Beziehung wird als Verbindungslinie zwischen den beiden Tabellen *Personal* und *Projekte* dargestellt. An den Beschriftungen dieser Verbindungslinie *1* bzw. ¥ können Sie erkennen, daß die Tabelle *Personal* die 1-Seite und *Projekte* die n-Seite der Beziehung ist.
- Schließen Sie das Fenster *Beziehungen* durch Klicken auf die Schaltfläche *Schließen*. Dabei werden Sie gefragt, ob Layoutänderungen an den Beziehungen gespeichert werden sollen. Bejahen Sie diese Frage, weil andernfalls zwar nicht die Definition der Beziehung, jedoch deren grafische Darstellung im Fenster *Beziehungen* verloren ginge.

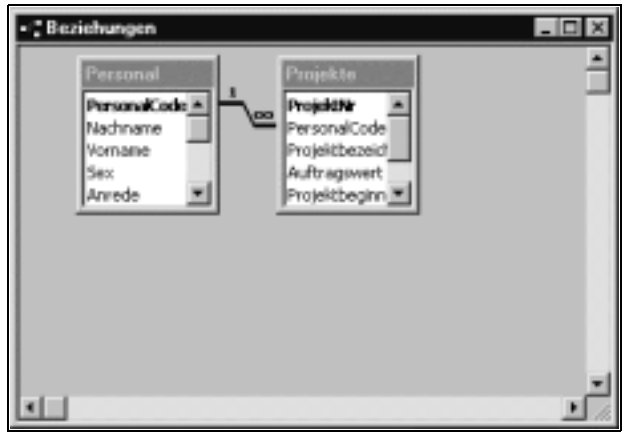


Bild 5.31: Die 1: n-Beziehung wird als Verbindungslinie zwischen den beiden Tabellen »Personal« und »Projekte« dargestellt. An den Beschriftungen »1« und »∞« erkennen Sie die 1-Seite und die n-Seite der Beziehung.

- *Hinweis*: Falls Sie die Layoutänderungen an den Beziehungen versehentlich nicht gespeichert haben, können Sie diese auch nachträglich anzeigen lassen. Gehen Sie dazu wie folgt vor: a) Öffnen Sie bei aktiviertem Datenbankfenster das Fenster *Beziehungen* mittels der Symbol-Schaltfläche *Beziehungen* oder dem Befehl *Beziehungen...* aus dem Menü *Extras*. b) Wählen Sie bei aktiviertem Fenster *Beziehungen* den Befehl *Direkte Beziehungen anzeigen* aus dem Menü *Beziehungen* oder klicken Sie auf die gleichnamige Symbol-Schaltfläche, vgl. links nebenstehend.

5.7.2 Bedeutung der referentiellen Integrität

Referentielle Integrität

Das Ankreuzen des Kontrollkästchens *Mit referentieller Integrität* und das damit verbundene Festlegen einer 1:n-Beziehung hat für die beiden Tabellen eine weitreichende Wirkung: In die Tabelle *Projekte* können dann nur Datensätze eingegeben werden, deren Feld *PersonalCode* einen Wert hat, der auch in der Mastertabelle *Personal* vorkommt. Umgekehrt: Sie können einen Datensatz aus der Tabelle *Personal* nicht löschen, ohne den oder die damit verknüpften Datensätze in der Tabelle *Projekte* ebenfalls zu löschen. Die Konsequenz ist die beabsichtigte und zugleich wesentliche Wirkung referentieller Integrität: In der *Detailtabelle* (das ist die n-Seite der Beziehung) sollen niemals verwaiste Datensätze existieren, d.h. keine Datensätze, für deren Fremdschlüsselfeld keine Entsprechung im Primärschlüsselfeld der *Mastertabelle* vorhanden ist.

Löschweitergabe an Detaildatensatz. Die Wahl dieser Option hat die folgende Wirkung: Wenn Sie einen Datensatz der *Mastertabelle* löschen, werden alle Datensätze der Detailtabelle, die mit ihm verknüpft sind, ebenfalls gelöscht. Wenn die Option nicht gewählt wurde, können Sie bei referentiel-

ler Integrität einen Datensatz der *Mastertabelle* nur dann löschen, wenn mit diesem kein Datensatz der Detailtabelle verknüpft ist. Zum Löschen eines solchen Datensatzes in der Mastertabelle müßten Sie dann zuvor die mit diesem verknüpften Datensätze in der Detailtabelle löschen.

Aktualisierungsweitergabe an Detailfeld. Diese Option betrifft die Aktualisierung (d.h. Änderung) des Wertes für das Primärschlüsselfeld der *Mastertabelle*. Wenn die Option nicht gewählt ist, kann der Wert dieses Feldes für solche Datensätze nicht mehr geändert werden, für die verknüpfte Datensätze in der Detailtabelle existieren. Wenn die Option dagegen gewählt wurde, können solche Werte geändert werden, und der neue Wert wird im Fremdschlüsselfeld der Detaildatensätze von Access entsprechend geändert.

5.8 Projektdaten in die Tabelle eingeben

Obwohl Sie die Projektdaten im allgemeinen Falle über ein Formular eingeben sollten, empfehle ich Ihnen, die Erfahrung zu machen, wie sich Access bei der Dateneingabe in eine Tabelle der n-Seite einer 1:n-Beziehung mit referentieller Integrität verhält. Um einen Datensatz in die Tabelle Projekte einzugeben, gehen Sie folgendermaßen vor:

- Öffnen Sie die Tabelle *Projekte* in der Datenblattansicht: Wechseln Sie dazu zunächst zum Datenbankfenster (Taste F11), klicken Sie dort auf das Register *Tabellen*, so daß dieses aktiviert ist, markieren Sie die Tabelle *Projekte* in der Tabellenliste, und klicken Sie auf die Schaltfläche *Öffnen* (oder doppelklicken Sie auf das Objekt *Projekte*).
- Versuchen Sie, einen neuen Datensatz einzugeben, indem Sie sich ein Projekt ausdenken. Geben Sie als *PersonalCode* einen Code ein, der in der *Mastertabelle Personal* nicht als *PersonalCode* existiert, z.B. AAAAA, und geben Sie die weiteren Werte für den ersten Datensatz ein.
- Klicken Sie in die nächste Datensatzzeile, um Access zu veranlassen, den Datensatz zu speichern. Sie werden die Meldung erhalten, die in Bild 5.32 wiedergegeben ist, weil Ihre Dateneingabe gegen die Regeln der referentiellen Integrität verstieß.
- Drücken Sie, ggf. mehrmals, die Esc-Taste, um die Dateneingabe aufzuheben.
- Schließen Sie die Tabelle *Projekte*.

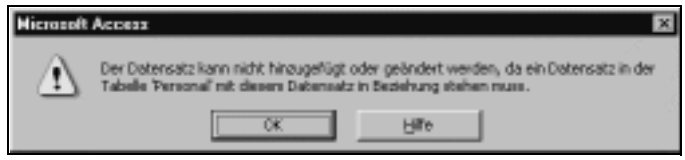


Bild 5.32: Weil mit dem PersonalCode »AAAAA« ein in der Mastertabelle nicht vorhandener PersonalCode eingegeben wurde, weigert sich Access, den Datensatz für die Detailtabelle zu übernehmen und macht mit dieser Meldung darauf aufmerksam.

5.9 Formular *Personal erstellen*

In die Tabelle *Personal* können neue Datensätze ohne Rücksicht auf Datensätze in der Tabelle *Projekte* eingegeben werden. Aus diesem Grunde ist es auch sinnvoll, dafür ein eigenes Formular zu erstellen. In diesem Abschnitt wird gezeigt, wie Sie ein entsprechendes Formular mit Hilfe des Formular-Assistenten erzeugen lassen können.

Hinweis: Im übernächsten Punkt (vgl. 5.11, *Formular Personal und Projekte erstellen*) wird gezeigt, wie Sie ein Formular erzeugen können, das gleichermaßen für die Eingabe von Personal- und Projektdaten geeignet ist. Daher wäre es eigentlich unter dem Gesichtspunkt der Datenbankgestaltung überflüssig, ein eigenständiges Personalformular zu erzeugen. Der Schritt ist aber aus didaktischen Gründen geboten: Das Formular *Personal und Projekte* besteht aus einem Haupt- und einem Unterformular und ist damit ein relativ komplexes Formular. Bevor Sie ein solches Formular erzeugen, sollten Sie zunächst ein einfaches erstellen. Dies trifft für das Formular *Personal* zu.

Um das Formular *Personal* zu erstellen, gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11.
- Aktivieren Sie die Objektliste *Formulare*, und klicken Sie anschließend auf die Schaltfläche *Neu*. Alternativ schlagen Sie die Dropdown-Liste der Symbol-Schaltfläche *Neues Objekt* auf, und wählen Sie darin das Objekt *Formular*. Dann zeigt sich das Dialogfeld *Neues Formular*, vgl. Bild 5.33.

Bild 5.33: Dialogfeld »Neues Formular« zum Erstellen eines neuen Formulars

- Wählen Sie im Listenfeld den Eintrag *Auto-Formular: Einspaltig*. Damit bestimmen Sie, daß der Formular-Assistent vollständig automatisch ein Formular erstellen wird.
- Schlagen Sie das Dropdown-Listenfeld auf und wählen Sie daraus den Tabellennamen *Personal*. Damit bestimmen Sie, daß ein Formular für die Felder dieser Tabelle erstellt wird.

- Bestätigen Sie das Dialogfeld *Neues Formular* mit OK. Daraufhin erstellt der Formular-Assistent das Formular, das in Bild 5.33 wiedergegeben ist.

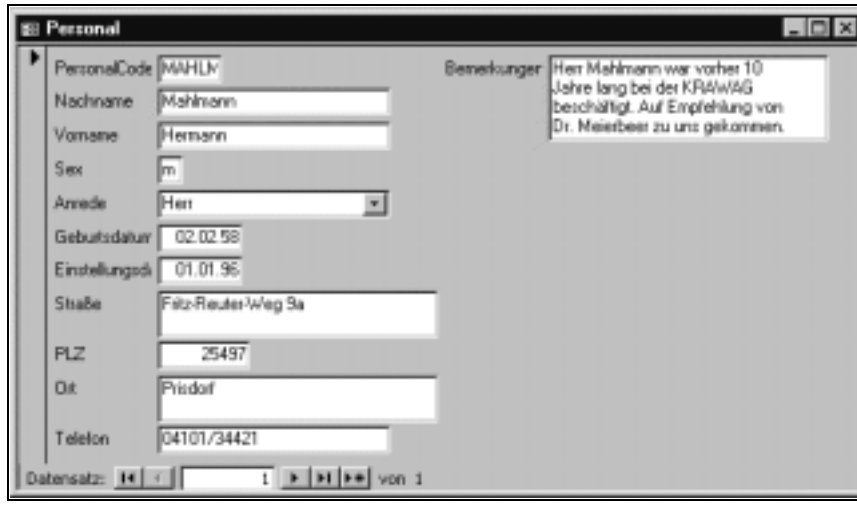


Bild 5.34: Formular »Personal«, das vom Formular-Assistenten als »Auto-Formular: Einspaltig« erstellt wurde

Das neue Formular existiert bis jetzt nur im Arbeitsspeicher. Um es dauerhaft zu sichern, gehen Sie folgendermaßen vor:

- Klicken Sie, während das Formular aktiv ist, auf die Symbol-Schaltfläche *Speichern*, vgl. links nebenstehend oder wählen Sie den Befehl *Speichern* aus dem Menü *Datei* oder drücken Sie die Tastenkombination Strg + S.
- Geben Sie in das Dialogfeld *Speichern unter* als Formularnamen *Personal* ein (Access sollte dort bereits den Namen *Personal* anbieten, so daß Sie wahrscheinlich nur zu bestätigen brauchen), und bestätigen Sie das Dialogfeld mit OK. Ab jetzt wird das Formular in der Objektliste *Formulare* des Datenbankfensters mit diesem Namen aufgeführt.

5.10 Personaldaten in Formular eingeben

Geben Sie ein paar der oben in Tabelle 5.1 wiedergegebenen Mitarbeiterinformationen als Datensätze ein, oder erfinden Sie selbst neue Mitarbeiterangaben. Gehen Sie dazu folgendermaßen vor:

- Blättern Sie ggf. zu einem neuen, leeren Datensatz. Dies können Sie auf verschiedene Weisen erreichen (z.B. mit dem Befehl *Gehe zu* aus dem Menü *Bearbeiten*), am schnellsten geht es jedoch durch Klicken auf die Symbol-Schaltfläche *Neuer Datensatz* in der Symbolleiste, vgl. links nebenstehend.

- Um Daten in ein einzelnes Feld einzugeben, klicken Sie auf dieses Feld (oder bewegen Sie sich mit der [⇆]-Taste von Feld zu Feld), und schreiben die gewünschte Information. Versuchen Sie einmal, gegen eine der Gültigkeitsregeln zu verstoßen: Auch im Formular werden diese erkannt! Wenn Sie in das Feld *Bemerkungen* klicken, werden Sie sehen, daß sich dann an dessen rechtem Rand eine kleine Bildlaufleiste zeigt, mit der Sie lange Texte rollen können. Der Formular-Assistent hat auch hier angemessen darauf reagiert, daß das Tabellenfeld *Bemerkungen* den Felddatentyp *Memo* besitzt.

Am linken Formularrand befindet sich eine senkrechte Leiste. Dies ist der *Datensatzmarkierer*. Wenn Sie darauf klicken, wird der gesamte Datensatz markiert, den Sie dann z.B. löschen könnten. Im Datensatzmarkierer wird auch der Bearbeitungszustand des aktuell sichtbaren Datensatzes angezeigt. Wenn Sie beispielsweise eine Änderung vorgenommen haben und diese noch nicht gespeichert wurde, zeigt sich im oberen Teil des Datensatzmarkierers das Bleistiftsymbol, das Sie bereits aus der Datenblattansicht der Tabelle kennen.

Die in das Formular eingegebenen Daten werden in der Tabelle, der das Formular zugeordnet ist (hier also Tabelle *Personal*), gespeichert, sobald Sie im Formular von einem Datensatz zum nächsten oder zu einem neuen blättern, gleichgültig, mit welcher Methode Sie das Blättern veranlassen. Sie verhindern das Speichern von Änderungen, indem Sie vor dem Blättern die [Esc]-Taste, ggf. mehrmals, drücken. Sie können vorgenommene Änderungen auch explizit speichern, ohne zu einem anderen Datensatz zu blättern: Klikken Sie auf den Datensatzmarkierer, während er das Bleistiftsymbol zeigt, dann wird der in Bearbeitung befindliche Datensatz gespeichert.

Wenn Sie Ihre Eingabearbeit am Formular abgeschlossen haben, sollten Sie das Formular *Personal* schließen. Beim Schließen werden alle noch nicht gesicherten Datensatzänderungen automatisch, ohne Rückfrage, gespeichert.

5.11 Formular *Personal und Projekte erstellen*

Das Formular *Personal und Projekte* soll es ermöglichen, *gleichzeitig* Personal- und Projektdaten einzugeben. Dies können Sie mit einem Formular erledigen, welches aus einem Haupt- und einem Unterformular besteht. Als Datenbasis für das Hauptformular dient dabei die *Mastertabelle*, hier also *Personal*, und für das Unterformular die *Detailtabelle*, hier also *Projekte*. Mit dem Formular-Assistenten ist diese Aufgabe sehr einfach zu bewerkstelligen. Gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste [F11].
- Aktivieren Sie die Objektliste *Formulare*, und klicken Sie anschließend auf die Schaltfläche *Neu*. Alternativ schlagen Sie die Dropdown-Liste der Symbol-Schaltfläche *Neues Objekt* auf, und wählen Sie darin das Objekt *Formular*. Dann zeigt sich das Dialogfeld *Neues Formular*, vgl. weiter oben Bild 5.33.

- Wählen Sie im Dialogfeld *Neues Formular* den Listeneintrag *Formular-Assistent*, und bestätigen Sie mit OK. Dann zeigt sich das erste Dialogfeld des Formular-Assistenten, vgl. Bild 5.35.

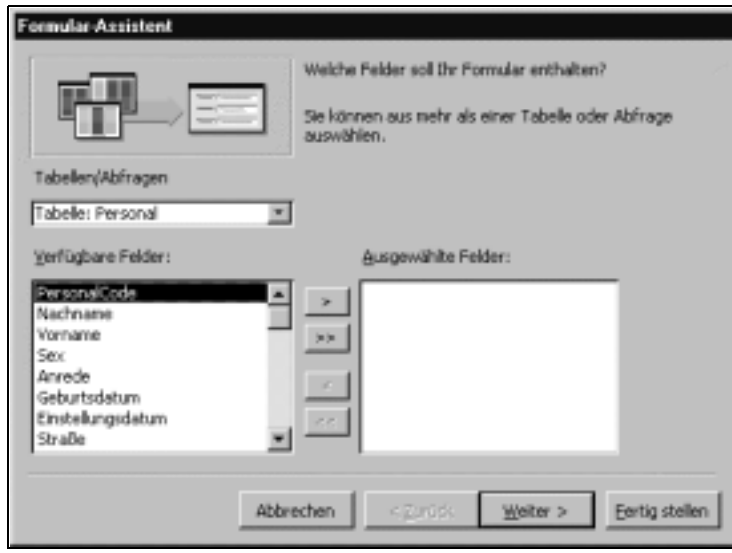


Bild 5.35: Erstes Dialogfeld des Formular-Assistenten zum Festlegen der im Formular zu berücksichtigenden Tabellen und Felder

- Belassen Sie es im ersten Dialogfeld des Formular-Assistenten unter *Tabellen/Abfragen* bei der Tabelle *Personal*, und nehmen Sie alle verfügbaren Felder in die Liste *Ausgewählte Felder* auf, indem Sie auf die Schaltfläche mit dem doppelten Pfeil klicken.
- Wählen Sie unter *Tabellen/Abfragen die Tabelle Projekte; dann erscheinen deren Feldnamen in der Liste Verfügbare Felder. Nehmen Sie auch alle diese Felder in die Liste Ausgewählte Felder* auf, indem Sie auf die Schaltfläche mit dem doppelten Pfeil klicken. Klicken Sie anschließend auf die Schaltfläche *Weiter* >, um das zweite Dialogfeld des Formular-Assistenten anzeigen zu lassen, vgl. Bild 5.36.
- Im zweiten Dialogfeld des Formular-Assistenten bestimmen Sie zunächst, welche der beiden Tabellen, aus denen Sie im vorigen Dialogfeld Felder für das Formular ausgewählt haben, als obere Gliederungsebene dienen soll. Dies soll die Tabelle *Personal* sein. Belassen Sie es daher bei dem Vorschlag des Formular-Assistenten, der die Frage *Wie sollen Ihre Daten angezeigt werden?* mit dem Markieren des Listeneintrags *nach Personal* beantwortet hat.

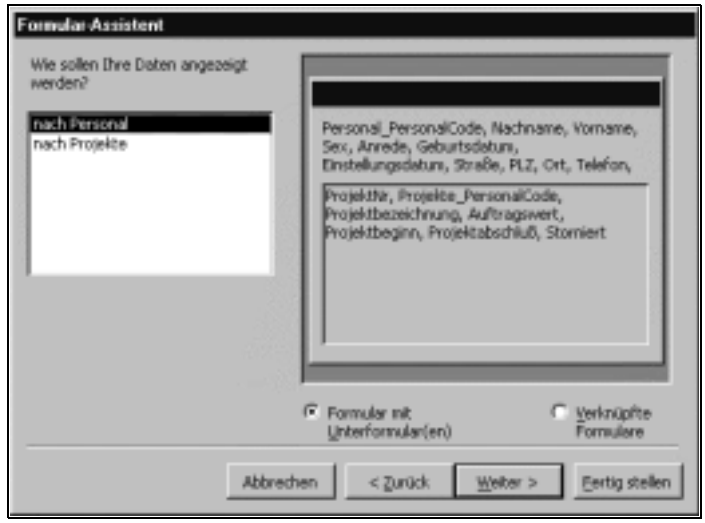


Bild 5.36: Zweites Dialogfeld des Formular-Assistenten zum Festlegen der Formularstruktur

- Zum anderen müssen Sie entscheiden, ob ein *Formular mit Unterformular(en)* oder *Verknüpfte Formulare* erstellt werden sollen. Wählen Sie bitte die erste dieser beiden Möglichkeiten, weil sie gerade für Einsteiger (aber nicht nur für diese) die klarere Formularstruktur bietet, und bestätigen Sie mit *Weiter* >. Dann wird das dritte Dialogfeld des Formular-Assistenten angezeigt, vgl. Bild 5.37.

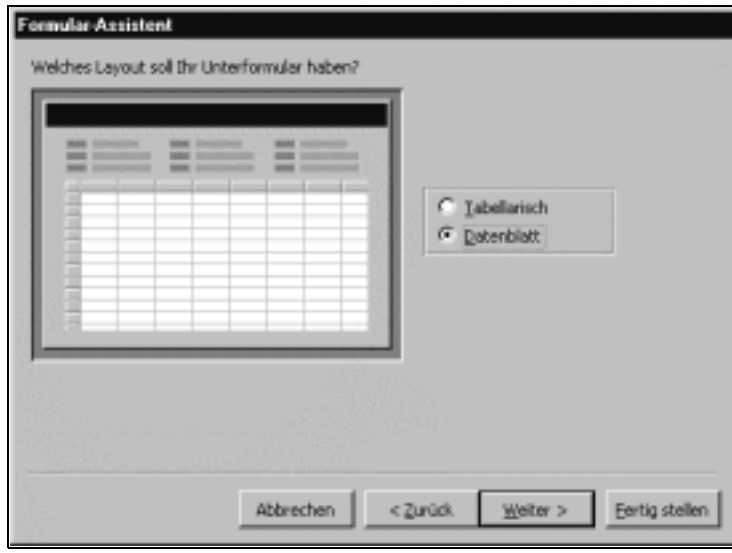


Bild 5.37: Drittes Dialogfeld des Formular-Assistenten zum Festlegen des Layout für das Unterformular

- Wählen Sie im dritten Dialogfeld des Formular-Assistenten für das Unterformular das Layout *Datenblatt*, und bestätigen Sie mit *Weiter* >. Dann wird das vierte Dialogfeld des Formular-Assistenten angezeigt, vgl. Bild 5.38.

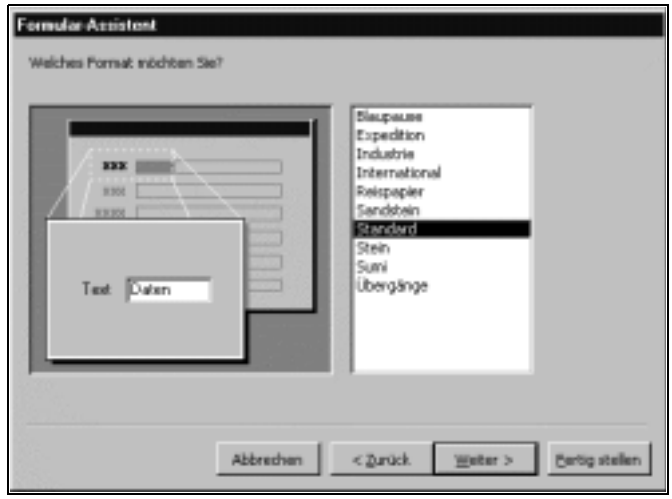


Bild 5.38: Viertes Dialogfeld des Formular-Assistenten zur Angabe der Formularstils

- Wählen Sie im vierten Dialogfeld des Formular-Assistenten den Stil *Standard*, dann erhalten Sie ein ziemlich neutrales Aussehen für Ihr Formular, und bestätigen Sie mit *Weiter* >. Dann wird das fünfte und letzte Dialogfeld des Formular-Assistenten angezeigt, vgl. Bild 5.39.
- Tragen Sie als Namen für das (Haupt-)Formular *Personal und Projekte* ein und für das Unterformular *Unterformular Projekte*. Daß Sie hier je einen Namen für das Haupt- und Unterformular angeben müssen, hat seinen Grund darin, daß Haupt- und Unterformular jeweils selbständige Formulare sind. Das Unterformular ist lediglich ein bestimmtes Element des Hauptformulars. Damit sich das Hauptformular auf das Unterformular beziehen kann, muß dieses einen Namen besitzen. Haupt- und Unterformular werden in der Liste des Registers *Formulare* im Datenbankfenster als jeweils eigenständige Formulare angezeigt, denen auf dieser Ebene dann auch nicht – außer über einen zweckmäßig festgelegten Namen – anzusehen ist, ob ein Formular als Unterformular fungiert. Belassen Sie es bei den anderen vom Formular-Assistenten voreingestellten Optionen und bestätigen Sie das Dialogfeld durch Klicken auf die Schaltfläche *Fertigstellen*. Dann zeigt der Formular-Assistent nach einer kurzen Bearbeitungszeit das fertige Formular *Personal und Projekte*, welches im unteren Teil das *Unterformular Projekte* enthält, vgl. Bild 5.40.

Bild 5.39: Fünftes Dialogfeld des Formular-Assistenten, in dem u.a. die Namen anzugeben sind, unter denen das Haupt- und das Unterformular gespeichert werden

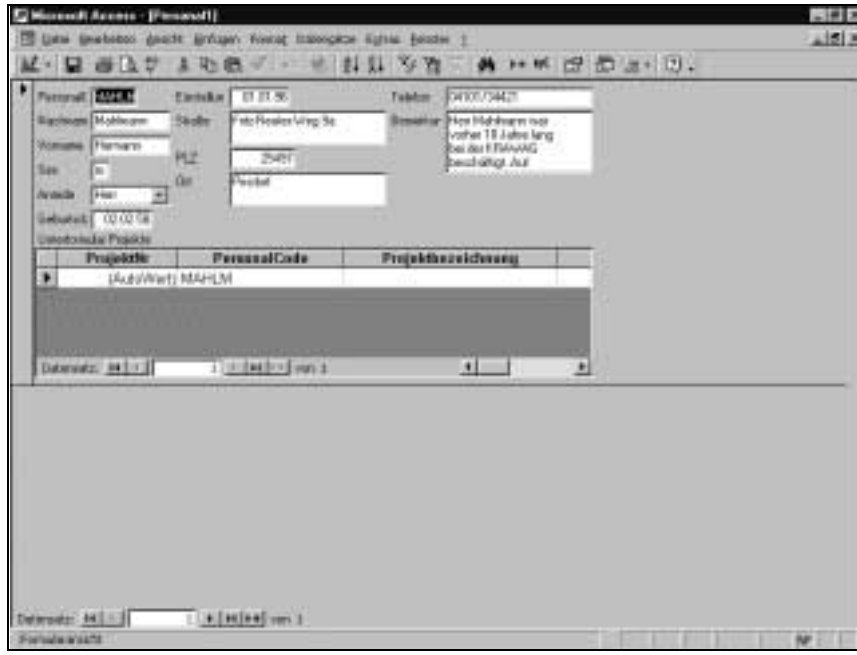


Bild 5.40: Fertiges Formular »Personal« und »Projekte« in Vollbilddarstellung mit »Unterformular Projekte« im unteren Teil als Ergebnis der Arbeit des Formular-Assistenten

Auf den ersten Blick scheint das Arbeitsergebnis des Formular-Assistenten perfekt zu sein: Für jedes Feld der Tabelle *Personal* findet sich im oberen Formularteil ein Textfeld mit jeweils zugeordnetem Bezeichnungsfeld, so daß auch für jedes Feld Daten eingegeben oder bearbeitet werden können. Das Unterformular ist im unteren Formularteil angeordnet und hat die Form einer Tabelle in Datenblattansicht. Zu einer solchen Darstellungsform gehört eine waagerechte Bildlaufleiste mit den vier Navigationsschaltflächen, um mit der Maus zum ersten, vorigen, nächsten oder letzten Datensatz zu blättern. Eine derartige Bildlaufleiste mit vier Navigationsschaltflächen weist auch das Hauptformular auf; diese befindet sich am unteren Rand des Formulars.

Bei genauerem Hinsehen erkennen Sie allerdings ein paar Unzulänglichkeiten im Arbeitsergebnis des Formular-Assistenten: Die Bezeichnungen mehrerer Textfelder werden unvollständig wiedergegeben, so z.B. die Bezeichnungen *PersonalCode* des ersten oder *Bemerkungen* des letzten Textfeldes. Darüber hinaus (und im Zusammenhang mit dem vorigen Punkt stehend) wurden die einzelnen Steuerelemente angesichts des auf dem Bildschirm verfügbaren Platzes unnötig eng angeordnet. Insbesondere wäre es vorteilhaft, wenn das Unterformular breiter und mit seinen Spalten besser aufgeteilt wäre, so daß die Spalten sämtlicher Projektfelder gleichzeitig sichtbar wären.

Sie sollten daher die Arbeit des Formular-Assistenten vervollständigen, um ein aus Ihrer Sicht angemessenes Formular zu erstellen. Dies können Sie erreichen, indem Sie einzelne Steuerelemente in der Entwurfsansicht des Formulars verschieben und ggf. in ihrer Größe verändern. Diese Aufgabe soll im folgenden Abschnitt gelöst werden. Wenn sie gelöst ist, wird das Formular einfacher weil übersichtlicher zu handhaben sein. Aus diesem Grunde sollten Sie auch erst danach Daten eingeben.

5.12 Position und Größe der Steuerelemente im Formular *Personal und Projekte ändern*

Aufgabenstellung

Ordnen Sie die Textfelder mit ihren Bezeichnungsfeldern im oberen Formularteil so an, daß Sie genügend Platz zum Vergrößern derjenigen Bezeichnungsfelder erhalten, deren Texte abgeschnitten erscheinen. Achten Sie auch auf eine ästhetisch ansprechende Anordnung, wozu auch gehören sollte, daß Textfelder, die ungefähr gleich lange Texte aufzunehmen haben, im allgemeinen auch dieselbe Höhe haben sollten.

Vergrößern Sie die betreffenden Bezeichnungsfelder so, daß die Texte vollständig lesbar sind.

Vergrößern Sie das Textfeld *Bemerkungen* entsprechend dem noch verfügbaren Platz so, daß möglichst viel Text darin lesbar ist, ohne mit der Bildlaufleiste rollen zu müssen.

Vergrößern Sie das Steuerelement mit der Bezeichnung *Unterformular Projekte* in der Waagerechten so, daß es die gesamte Formularbreite in Vollbilddarstellung einnimmt.

Verändern Sie die Spaltenbreiten im Unterformular so, daß die Spalten aller Felder gleichzeitig zu sehen sind, so daß das Datenblatt im Unterformular nicht mehr in der Waagerechten gerollt zu werden braucht; dann wird auch die waagerechte Bildlaufleiste verschwinden.

Fügen Sie in den Formularkopf ein Bezeichnungsfeld mit der Beschriftung *Personal und Projekte* ein, und formatieren Sie dieses mit einer größeren Schriftart.

Im Ergebnis soll das Formular ungefähr so aussehen, wie dies in Bild 5.41 zu sehen ist.

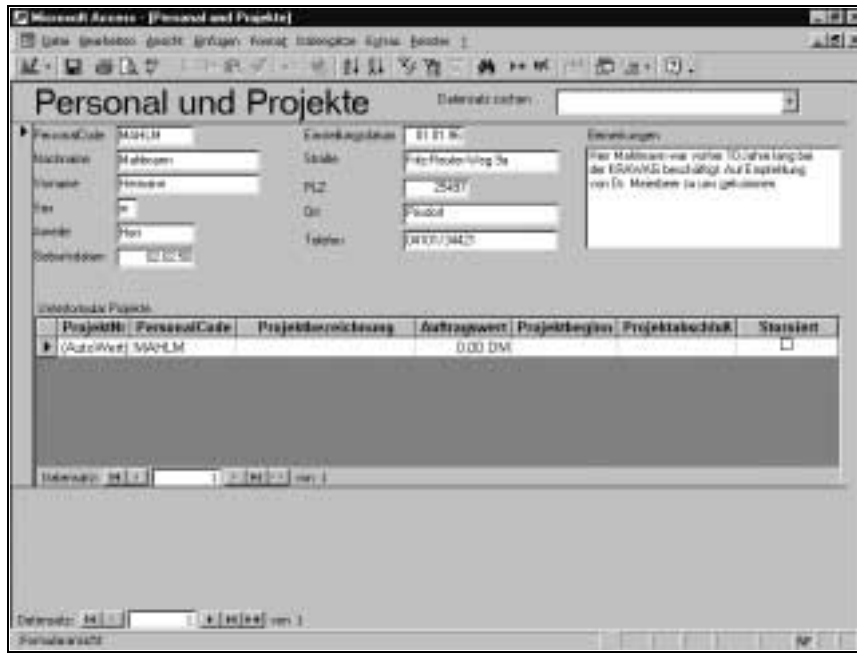


Bild 5.41: Nach dem Verschieben der Steuerelemente soll das Formular ungefähr so aussehen, wie hier dargestellt.

Hinweis: Die hier angenommenen und wiedergegebenen Formulardarstellungen basieren auf einer Bildschirmauflösung von 800 x 600 Punkten. Bei einer geringeren Auflösung – z.B. der VGA-Standardauflösung von 640 x

480 Punkten – oder der Verwendung einer größeren Systemschriftart steht auf dem Bildschirm weniger Platz zur Verfügung. Dann müssen Sie die hier gezeigten Beispiele entsprechend anpassen.

Steuerelemente praktisch bearbeiten

Um die formulierten Aufgaben zu lösen, gehen Sie folgendermaßen vor:

- Sorgen Sie ggf. dafür, daß das Formular *Personal und Projekte* aktiv ist; möglicherweise müssen Sie es erst aus dem Datenbankfenster heraus öffnen oder, wenn es bereits geöffnet ist, über das Menü *Fenster* auswählen.

- Wechseln Sie von der Formular- in die Entwurfsansicht, indem Sie auf die Symbol-Schaltfläche *Entwurfsansicht* klicken oder den gleichnamigen Befehl aus dem Menü *Ansicht* wählen. Dann zeigt sich das Formular in der Entwurfsansicht, wie dies in Bild 5.42 wiedergegeben ist.

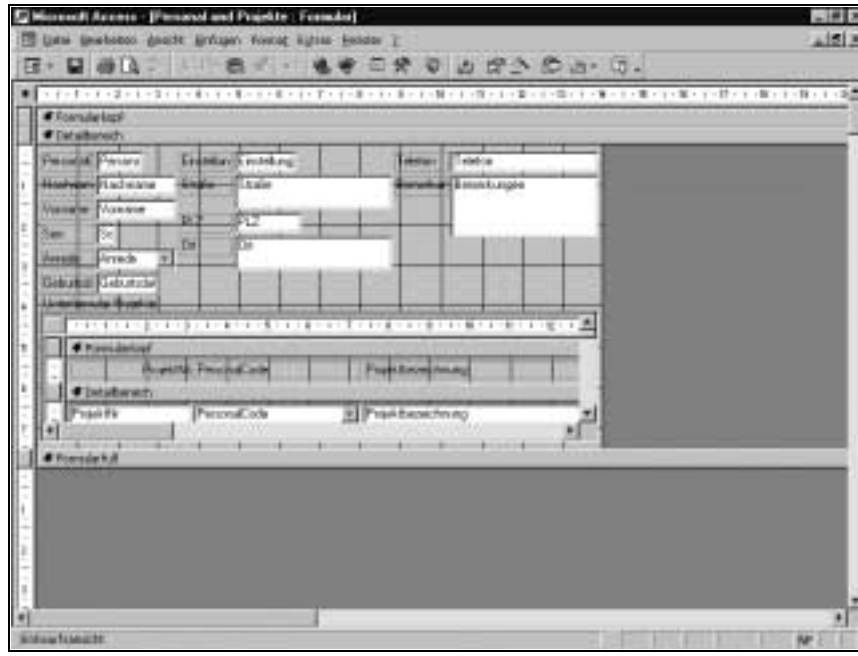


Bild 5.42: Formular »Personal« und »Projekte« im unbearbeiteten Zustand in der Entwurfsansicht

Wenn Ihr Bildschirm nach dem Wechseln in die Entwurfsansicht nicht so aussieht wie in Bild 5.42, sondern möglicherweise kleine Fenster für die Eigenschaften, die Feldliste oder die Toolbox zeigt, ändern Sie diese Darstellungsweise, indem Sie diese Fenster schließen. Vergewissern Sie sich auch, daß die Steuerelemente im Formular am Raster ausgerichtet werden. Aus dem Umstand, daß Rasterpunkte sichtbar oder unsichtbar sind, können Sie

dies nicht schließen. Vielmehr muß der Befehl *Am Raster ausrichten* aus dem Menü *Format* ein Häkchen tragen. Schlagen Sie daher dieses Menü auf und klicken Sie ggf. auf den Befehl.

Sie verschieben ein Steuerelement oder verändern seine Größe am einfachsten mit der Maus. Berücksichtigen Sie dazu die Symbole, die in Bild 5.43 gezeigt und benannt sind. Dort wird als Beispiel das Feld *Bemerkungen* aus dem in Arbeit befindlichen Formular verwendet. Bevor Sie ein Steuerelement mit der Maus bearbeiten, müssen Sie es zunächst markieren. Dazu klicken Sie einmal kurz auf einen beliebigen Punkt eines Randes vom Steuerelement. Um mehrere Steuerelemente zu markieren, klicken Sie mit gedrückter ⇧-Taste auf die weiteren Steuerelemente oder ziehen den Mauszeiger über mehrere Steuerelemente. Ein markiertes Textfeld mit Bezeichnungsfeld hat das in Bild 5.43 wiedergegebene Aussehen. Zum besseren Verständnis müssen Sie wissen, daß die üblichen Eingabefelder in Formularen, wie Sie sie bisher kennengelernt haben, bei Access als *Textfelder* bezeichnet werden. Davon zu unterscheiden ist das Feld, welches das Textfeld bezeichnet. Dieses Feld heißt *Bezeichnungsfeld*.

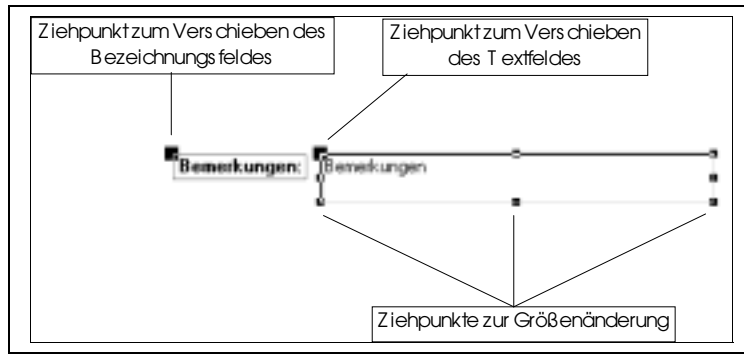


Bild 5.43: Ziehpunkte zum Verändern von Position und Größe. Textfeld und Bezeichnungsfeld werden zusammen verschoben, indem Sie den Mauszeiger auf einen beliebigen Punkt eines Textfeldrandes (jedoch nicht auf einen Ziehpunkt) fahren, so daß er wie eine geöffnete schwarze Hand aussieht.

Verschieben/Größenändern von Text- und/oder Bezeichnungsfeldern

Dabei sind drei verschiedene Möglichkeiten zu berücksichtigen:

Text- und Bezeichnungsfeld zusammen verschieben. Um das Textfeld zusammen mit seinem Bezeichnungsfeld zu verschieben, fahren Sie den Mauszeiger auf einen beliebigen Randpunkt des markierten Textfeldes (jedoch nicht auf einen Ziehpunkt), so daß er wie eine geöffnete schwarze Hand aussieht. Ziehen Sie dann (*Ziehen* bedeutet stets: mit gedrückter linker Maustaste) Text- mit Bezeichnungsfeld an die gewünschte Position. Wenn Sie mehrere Steuerelemente markiert haben, werden alle markierten verschoben, wenn Sie ein beliebiges davon verschieben.

Textfeld ohne sein Bezeichnungsfeld verschieben. Fahren Sie den Mauszeiger auf den Ziehpunkt zum Verschieben des Textfeldes. Er sieht dann aus wie eine schwarze Hand mit abgespreiztem Daumen und Zeigefinger. Ziehen Sie das Textfeld an die gewünschte Position.

Bezeichnungsfeld ohne sein Textfeld verschieben. Fahren Sie den Mauszeiger auf den Ziehpunkt zum Verschieben des Bezeichnungsfeldes. Er sieht dann aus wie eine schwarze Hand mit abgespreiztem Daumen und Zeigefinger. Ziehen Sie das Bezeichnungsfeld an die gewünschte Position.

Um die *Größe eines Steuerelements* mit der Maus zu verändern, ziehen Sie einen seiner Ziehpunkte zur Größenänderung. Der Mauszeiger hat dabei das Aussehen eines Doppelpfeils, wobei die Pfeile je nach gewähltem Ziehpunkt in verschiedene Himmelsrichtungen weisen.

Textfeld *Bemerkungen anpassen. Gehen Sie dazu folgendermaßen vor:*

- Markieren Sie zunächst das Textfeld *Bemerkungen*, indem Sie darauf klikken.
- Fahren Sie mit dem Mauszeiger auf einen Rand des markierten Feldes, so daß der Mauszeiger wie eine geöffnete schwarze Hand aussieht. Ziehen Sie jetzt an die gewünschte Position, d.h. bis an den rechten Bildschirmrand. Dabei werden Sie über den Rand des Formulars hinaus ziehen und feststellen, daß dieser Rand dadurch automatisch mit nach rechts verschoben wird.
- Verschieben Sie das Bezeichnungsfeld so, daß es nicht links neben, sondern oberhalb des Textfeldes positioniert ist. Dazu ziehen Sie den Ziehpunkt links oben am Bezeichnungsfeld.
- Vergrößern Sie das Textfeld *Bemerkungen* nach unten, indem Sie den mittleren Ziehpunkt seines unteren Randes nach unten ziehen.
- Verschieben Sie das Bezeichnungsfeld mit der Beschriftung *Bemerkungen*, die als Ergebnis des Formular-Assistenten nicht vollständig zu lesen ist, nach links, indem Sie dessen Ziehpunkt ziehen. Vergrößern Sie dann das Bezeichnungsfeld nach rechts, indem Sie den mittleren Ziehpunkt seines rechten Randes so weit nach rechts ziehen, bis die Beschriftung vollständig zu lesen ist.

Weitere Textfelder anpassen. Um die weiteren Textfelder zu verschieben und in ihrer Größe zu verändern, verfahren Sie entsprechend wie beim Textfeld *Bemerkungen*. Beachten Sie, daß Sie auch mehrere Steuerelemente gleichzeitig markieren können (weitere mit gedrückter ⇧-Taste markieren oder den Mauszeiger über mehrere Steuerelemente ziehen).

Unterformular anpassen. Schließlich muß noch das Unterformular angepaßt werden. Gehen Sie dazu folgendermaßen vor:

- **Vergrößern**. Markieren Sie das Unterformular. Dieses ist das große Element ganz unten im Formular mit der Beschriftung *Unterformular Pro-*

jekte. Vergrößern Sie dann das Unterformular nach rechts, indem Sie den mittleren Ziehpunkt seines rechten Randes bis an den Formularrand nach rechts ziehen.

- **Spaltenbreiten verändern**. Um die Spaltenbreiten der Felder, die im Unterformular dargestellt werden, zu verändern, müssen Sie von der Entwurfsansicht des Formulars in die Formularansicht wechseln. Klicken Sie dazu auf die Symbol-Schaltfläche *Formularansicht* oder wählen Sie den gleichlautenden Befehl aus dem Menü *Ansicht*. Ziehen Sie dann den rechten Rand des Kopfes einer Spalte in die gewünschte Richtung, um ihre Breite zu verändern. Die Breite der Spalte *ProjektNr* z.B. verkleinern Sie, indem Sie mit dem Mauszeiger den rechten Rand ihres Spaltenkopfes nach links ziehen. Wenn Sie auf den rechten Rand des Kopfes einer Spalte doppelklicken, wird deren Breite von Access optimiert: Sie wird so breit eingestellt, daß der längste Inhalt – die Spaltenbeschriftung eingeschlossen – gerade vollständig lesbar ist.

Formular speichern. Wenn alle Änderungen befriedigend ausgefallen sind, sollten Sie das Formular speichern (die Daten werden ja stets automatisch gespeichert, nicht jedoch Änderungen am Formularentwurf):

- Drücken Sie die Tastenkombination ⇧+F12 oder wählen Sie den Befehl *Speichern* aus dem Menü *Datei*.

5.13 Personal- und Projektdaten im Formular bearbeiten

Das Formular *Personal und Projekte* ermöglicht eine einfache Eingabe von Personal- und Projektdaten: Um die Daten eines neuen Mitarbeiters einzugeben, blättern Sie mit Hilfe der Navigationsschaltfläche *Neuer Datensatz*, die Sie sowohl in der Symbolleiste des Formulars wie auch unter den Navigationssymbolen des Hauptformulars am unteren Formularrand finden, zu einem neuen Datensatz. Wählen Sie den Unterbefehl *Neuer Datensatz* des Befehls *Gehe zu* aus dem Menü *Bearbeiten*. Die Daten für ein neues Projekt geben Sie einfach ein, indem Sie in die unterste leere Datensatzzeile des Unterformulars klicken und schreiben. Falls Sie für einen Mitarbeiter sehr viele Projekte eingegeben haben, blättern Sie mit den Navigationssymbolen in der waagerechten Bildlaufleiste des Unterformulars zum letzten Datensatz oder rollen mit der senkrechten Bildlaufleiste des Unterformulars.

Gegenüber der Eingabe von Projektdatensätzen in die *Tabelle Projekte* (vgl. oben Punkt 5.8, *Projektdaten in die Tabelle* eingeben) bietet das vorliegende Formular eine wesentliche Vereinfachung: Sie blättern zu dem Mitarbeiter, der das Projekt betreuen soll, und geben die Projektdaten in das Unterformular ein. Um die Verknüpfung zwischen Mitarbeiter- und Projektdatensatz, die ja über das jeweilige Feld *PersonalCode* hergestellt wird, brauchen Sie sich hier nicht zu kümmern, weil diese vom Formular automatisch vorgenommen und verwaltet wird. Bei direkter Eingabe in die Tabelle *Projekte* müßten Sie dagegen zunächst die *PersonalCode* des betreffenden Mitarbeiters ermitteln, bevor Sie die Daten des Projekts eingeben. Dies allerdings

könnten Sie dann bequem mit Hilfe des Nachschlagefeldes, das Ihnen für das Feld *PersonalCode* in Form einer Dropdown-Liste angeboten wird, bewerkstelligen.

Die Dateneingabe in dieses Formular ist nicht nur einfach, sondern auch sicher: Haupt- und Unterformular sind miteinander verknüpft. Dabei wird die 1:n-Beziehung mit referentieller Integrität, die zwischen den zugrundeliegenden Tabellen besteht, quasi implizit beachtet: Sie können einen Projektdatensatz stets nur zu einem bestimmten Mitarbeiter eingeben, dem er dann zugeordnet wird; Projektdatensätze ohne zugeordneten Mitarbeiter können nicht eingegeben werden.

Sie können jetzt sämtliche Mitarbeiterdaten aus der oben wiedergegebenen Tabelle 5.1, *Werte für die Datensätze der Tabelle Personal*, eingeben. Denken Sie sich dabei beliebige Projekte aus. Geben Sie (im Hinblick auf die im nächsten Abschnitt zu entwerfende Abfrage) einige Projekte mit einem Auftragswert von über, andere mit einem Auftragswert von unter 100.000 DM ein, von denen wiederum einige erledigt sind, andere nicht.

5.14 Abfrage: Unerledigte Projekte mit einem Auftragswert von mindestens 100.000 DM

Die Stärke eines DBMS zeigt sich darin, daß Sie Daten aus mehreren Tabellen nach praktisch allen denkbaren Kriterien zusammenstellen lassen können. Das Instrument zur Lösung einer derartigen Aufgabe ist eine *Abfrage*.

Aufgabenstellung

Stellen Sie in einer Abfragetabelle alle nicht stornierten Projekte mit einem Auftragswert von mindestens 100.000 DM zusammen. Die Abfrage soll

- die Projektbezeichnung
- den Auftragswert
- den Wert des Feldes *Storniert*
- den Vornamen des betreuenden Mitarbeiters
- den Namen des betreuenden Mitarbeiters

ausweisen.

Abfrage erstellen

Sie bearbeiten diese Aufgabenstellung auf die folgende Weise:

- Wechseln Sie ggf. zum Datenbankfenster (z.B. mit Taste F11)

- Aktivieren Sie im Datenbankfenster die Objektliste *Abfragen* und klicken Sie dann auf die Schaltfläche *Neu*. Alternativ: Schlagen Sie die Dropdown-Liste der Symbol-Schaltfläche *Neues Objekt* in der Symbolleiste auf, und

wählen Sie darin das Symbol für *Abfrage*. In diesem Falle müssen Sie nicht vorher zum Datenbankfenster wechseln. Welchen Weg Sie auch gehen: Als Ergebnis zeigt sich das Dialogfeld *Neue Abfrage*, vgl. Bild 5.44.

Bild 5.44: Dialogfeld »Neue Abfrage«

- Markieren Sie im Dialogfeld *Neue Abfrage* den Listeneintrag *Entwurfsansicht* (auf den Auswahlabfrage-Assistenten können wir verzichten, weil auch der direkte Weg sehr einfach und letztlich transparenter ist).
- Daraufhin öffnet Access das Fenster für eine neue Abfrage in der Entwurfsansicht und blendet gleichzeitig das Dialogfeld *Tabelle anzeigen* mit den verfügbaren Tabellen und Abfragen ein, vgl. Bild 5.45. Falls Sie die neue Abfrage angefordert haben, während eine Tabelle aktiv war oder während im Register *Tabellen* des Datenbankfensters ein Tabellensymbol markiert war, fügt Access dem Abfragefenster diese Tabelle gleich hinzu, ohne das Dialogfeld *Tabelle anzeigen* anzuzeigen. Dann blenden Sie dieses ein, indem Sie auf die Symbol-Schaltfläche *Tabelle anzeigen* klicken, vgl. links nebenstehend.

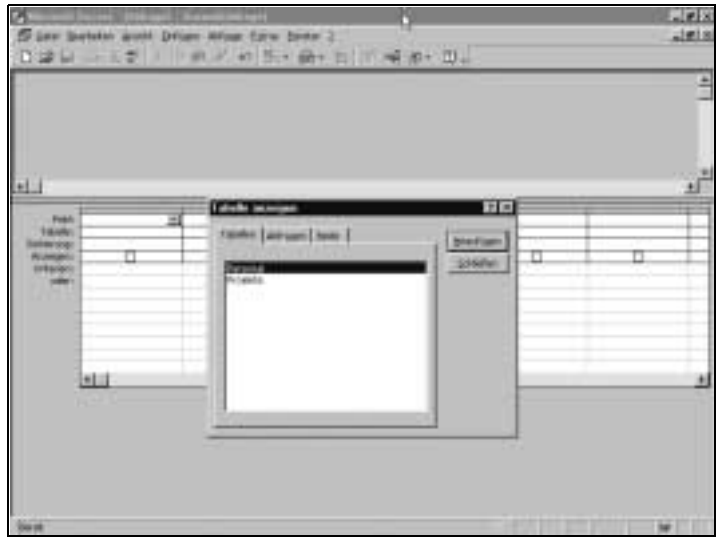


Bild 5.45: Abfragefenster in der Entwurfsansicht mit eingeblendetem Dialogfeld »Tabelle anzeigen«

- Fügen Sie der Abfrage die Tabellen *Personal* und *Projekte* hinzu, indem Sie im Dialogfeld *Tabelle anzeigen* die Tabellen doppelklicken. Jedesmal, wenn Sie der Abfrage eine Tabelle hinzugefügt haben, wird dem Abfrageentwurf in seinem oberen Fensterteil ein symbolisiertes Tabellenfenster mit den Feldnamen der Tabelle hinzugefügt, vgl. den oberen Teil von Bild 5.46. Dort sehen Sie auch, daß die beiden Felder *PersonalCode* der beiden Tabellen durch eine Linie miteinander verbunden sind. Access zeigt damit an, daß für die beiden Tabellen eine 1:n-Beziehung mit dem jeweiligen Schlüsselfeld *PersonalCode* definiert ist.
- Schließen Sie, wenn Sie beide Tabellen hinzugefügt haben, anschließend das Dialogfeld *Tabelle anzeigen*.

Nachdem Sie der Abfrage die beiden benötigten Tabellen hinzugefügt haben, müssen Sie ihr mitteilen, welche Felder im Abfrageergebnis berücksichtigt werden sollen. Gehen Sie dazu folgendermaßen vor:

- Ziehen Sie das Feld *Projektbezeichnung* aus der Tabelle *Projekte* in die erste Spalte der Zeile *Feld*. Unmittelbar nachdem Sie die linke Maustaste losgelassen haben, erscheint dort der Feldname *Projektbezeichnung*. Alternativ könnten Sie diesen Namen auch auf die folgende Weise in die erste Spalte der Zeile *Feld* eingeben: Klicken Sie in die erste Spalte der Zeile *Feld*, öffnen Sie das dann eingeblendete Dropdown-Listenfeld und wählen Sie aus der Liste der Feldnamen den gewünschten aus. Eine weitere Möglichkeit: Doppelklicken Sie auf das Feld *Projektbezeichnung* im symbolisierten Tabellenfenster *Projekte*. Dann erscheint dieses Feld in der nächsten freien Spalte (am Beginn der Arbeit also in der ersten Spalte) der Zeile *Feld*.
- Verfahren Sie entsprechend für die Felder *Auftragswert* und *Storniert* aus der Tabelle *Projekte* sowie für die Felder *Vorname* und *Nachname* aus der Tabelle *Personal*.

Bild 5.46: Fertig ausgefüllter Abfrageentwurf

Schließlich müssen Sie der Abfrage die Kriterien mitteilen, die bei der Datenzusammenstellung gelten sollen. Dies erreichen Sie folgendermaßen:

- Schreiben Sie in die Zeile *Kriterien:* für das Feld *Auftragswert*

  >=100000

- Schreiben Sie in die Zeile *Kriterien:* für das Feld *Storniert*

  Nein

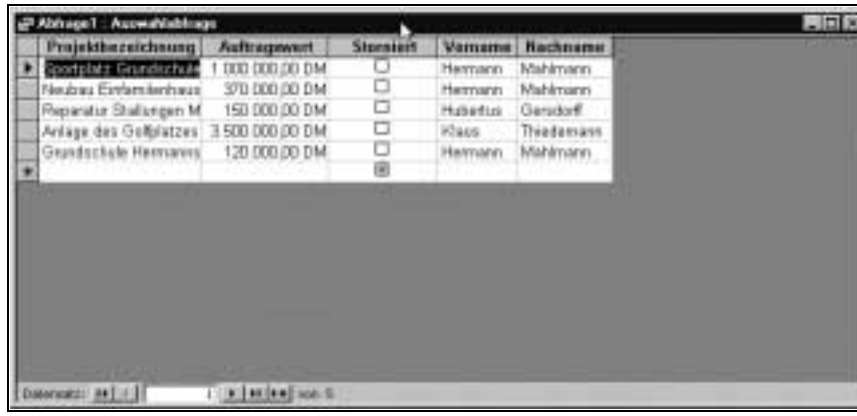
Den fertigen Abfrageentwurf sehen Sie in Bild 5.46.

Abfrageergebnis betrachten

Um das Ergebnis der Abfrage für die angegebenen Felder und Kriterien zu betrachten, wechseln Sie in die Datenblattansicht der Abfrage:

- Klicken Sie auf die Symbol-Schaltfläche *Datenblattansicht* in der Symbolleiste, oder wählen Sie den gleichlautenden Befehl aus dem Menü *Ansicht*.

Bild 5.47 zeigt das Abfrageergebnis für die in der Datenbank Projekte.mdb der Beispiel-CD-ROM enthaltenen Datensätze.



Projektbezeichnung	Auftragswert	Storniert	Vorname	Nachname
Sportplatz Grundschule	1.000.000,00 DM	☐	Hermann	Mahlmann
Neubau Einfamilienhaus	370.000,00 DM	☐	Hermann	Mahlmann
Reparatur Stallungen M	150.000,00 DM	☐	Hubertus	Gensdorff
Anlage des Golfplatzes	3.500.000,00 DM	☐	Klaus	Theodemann
Grundschule Hermanns	120.000,00 DM	☐	Hermann	Mahlmann

Bild 5.47: Abfrageergebnis für »Auftragswert ⇒ 100000« und »Storniert = Nein«

Wenn Sie die zu berücksichtigenden Felder oder die Kriterien ändern möchten, wechseln Sie wieder in die Entwurfsansicht der Abfrage, nehmen dort die gewünschten Änderungen vor und wechseln wiederum in die Datenblattansicht, um das aktualisierte Ergebnis zu betrachten.

Werte der Datensätze in der Datenblattansicht der Abfrage bearbeiten

Obwohl Sie mit einer Abfrage nicht direkt die zugrundeliegenden Tabellen geöffnet haben, können Sie die Werte der Datensätze in der Datenblattansicht einer Abfrage im allgemeinen genauso bearbeiten, als ob Sie die Tabellen direkt geöffnet hätten. (Zu Einschränkungen in dieser Hinsicht vgl. Kap.

11, *Auswahlabfragen*.) Wenn Sie beispielsweise für das in Bild 5.46 wiedergegebene Abfrageergebnis feststellen, daß der für den letzten Datensatz ausgewiesene Auftragswert von 120.000 DM tatsächlich 130.000 DM betragen müßte, so können Sie diesen Wert im Datenblatt der Abfrage direkt korrigieren. Die Aktualisierung wird dann an die zugrunde liegende Tabelle weitergegeben. Entsprechendes gilt für ein Formular, das auf einer Abfrage basiert.

Abfrage speichern

Sie können eine Abfrage auch speichern: Drücken Sie die Tastenkombination Strg+S oder wählen Sie den Befehl *Speichern* aus dem Menü *Datei* und geben Sie in das Dialogfeld *Speichern unter* einen Namen für die Abfrage an. Danach wird die Abfrage in die Liste der Abfragen des Datenbankfensters übernommen. Das Speichern einer Abfrage verbraucht übrigens kaum Platz auf der Festplatte, weil nur die im Abfrageentwurf gemachten Angaben, nicht aber das Abfrageergebnis gespeichert werden. Dies hat auch zur Konsequenz, daß jede Abfrage beim Öffnen neu berechnet wird, so daß Sie stets die aktuellsten Ergebnisse erhalten.

5.15 Datensätze filtern

Aufgabenstellung

Wenn Sie im Formular *Personal und Projekte* blättern, sollen nur die Datensätze der männlichen Mitarbeiter eingeblendet werden.

Sie lösen die Aufgabe, indem Sie für das Formular *Personal und Projekte* einen entsprechenden Filter definieren und dann anwenden. Gehen Sie folgendermaßen vor:

- Öffnen Sie ggf. das Formular *Personal und Projekte* aus dem Datenbankfenster heraus. Sollte das Formular bereits geöffnet, aber nicht aktiviert sein, wirkt das Klicken auf die Schaltfläche *Öffnen* im Register *Formular* des Datenbankfensters so, wie wenn Sie es im Menü *Fenster* auswählen.

- Blättern Sie im Formular zu einem Datensatz eines männlichen Mitarbeiters, und geben Sie dem Steuerelement *Sex* den Fokus.
- Klicken Sie auf die Symbol-Schaltfläche *Auswahlbasierter Filter* oder wählen Sie den gleichnamigen Unterbefehl des Befehls *Filter* aus dem Menü *Datensätze*.

Wenn Sie jetzt im Formular *Personal und Projekte* blättern, werden nur Datensätze männlicher Mitarbeiter angezeigt. Daß sich das Formular im gefilterten Zustand befindet, wird in der Symbolleiste angezeigt: Die links nebenstehende Symbol-Schaltfläche erscheint gedrückt und trägt in diesem Zustand den Namen *Filter entfernen*; im nicht gedrückten Zustand (also

ohne Filterwirkung) trägt sie den Namen *Filter/Sortierung anwenden*. Auf den eingeschalteten Filter weist außerdem der textliche Hinweis *(Gefiltert)* am rechten Rand der Navigationssymbole hin, vgl. den nachfolgenden Bildausschnitt.

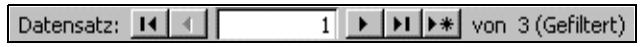


Bild 5.48: Hinweis auf den eingeschalteten Filter

Sie heben die Filterwirkung wieder auf, indem Sie auf die Symbol-Schaltfläche *Filter entfernen klicken oder den Befehl Filter/Sortierung entfernen aus dem Menü Daten wählen.*

5.16 Abfrage und Formular: Projekte und Personal

In den vorangehenden Punkten wurde u.a. ein Formular entwickelt, welches die gleichzeitige Bearbeitung von Personal- und Projektdaten auf bequeme und sichere Weise ermöglicht. Dieses Formular *Personal und Projekte* ist sinnvoll für die Aufgabenstellung, einen einzelnen Mitarbeiter (einschließlich neuer Datensatz) oder die Projekte eines bestimmten Mitarbeiters (ebenfalls einschließlich neuer Datensatz) zu bearbeiten.

Das Formular *Personal und Projekte* ist dagegen weniger geeignet, wenn Sie alle Projekte, unabhängig vom Mitarbeiter, durchblättern wollen, dabei aber gleichwohl den zugeordneten Mitarbeiter angezeigt haben möchten. Die Datenbank Projekte.mdb der Beispiel-CD-ROM enthält auch ein Formular, das für diese Fragestellung geeignet ist. Sie können es unter dem Formularnamen *Projekte und Personal 1* öffnen. Es besteht nur aus einem gewöhnlichen Formular (d.h. ohne Unterformular) und basiert auf der *Abfrage* gleichen Namens: Formulare können als Datenherkunft ebensogut eine Abfrage wie eine Tabelle besitzen. Wie Sie die Abfrage und das Formular in einzelnen Schritten erstellen, können Sie in Kap. 18, *Haupt- und Unterformular*, Punkt 18.6, *n:1-Formular erstellen*, erfahren. Dort wird auch ein entsprechendes Haupt- mit Unterformular mit dem Namen *Projekte und Personal 1* entwickelt, das Sie in der Datenbank *Projekte.mdb* finden können.

Kapitel 6

Daten bearbeiten

6.1 Objekte zur Bearbeitung von Datensätzen öffnen

Sie können Datensätze in den folgenden Objekten bearbeiten[1]:

- Tabelle in der Datenblattansicht
- Abfrage in der Datenblattansicht
- Formular in der Formular- oder Datenblattansicht

Jedes dieser drei Objekte können Sie entweder vom Datenbankfenster aus oder vom Desktop des PCs oder einem Ordner der Festplatte öffnen.

6.1.1 Öffnen aus dem Datenbankfenster heraus

Um eine Tabelle, Abfrage oder ein Formular vom Datenbankfenster aus zu öffnen, gehen Sie folgendermaßen vor:

- Aktivieren Sie das Datenbankfenster, z. B. durch Drücken von F11.
- Schlagen Sie die Objektliste für *Tabellen*, *Abfragen* oder *Formulare* auf.
- Markieren Sie das zu öffnende Objekt, und klicken Sie dann auf die Schaltfläche *Öffnen*.

  Oder:

- Doppelklicken Sie auf das zu öffnende Objekt.

  Oder:

- Klicken Sie mit der rechten Maustaste auf das zu öffnende Objekt, um das Kontextmenü aufzuschlagen, und wählen Sie darin den Befehl *Öffnen*.

---

1. Darüber hinaus lassen sich Datensätze auch in Datenzugriffsseiten bearbeiten. Diese werden jedoch nicht hier, sondern weiter unten in Kap. 34, *Access im Inter- und Intranet*, Punkt 34.8, *Datenzugriffsseiten*, behandelt.

Oder:

- Ziehen Sie das Objekt auf den Anwendungshintergrund von Access, das ist die Bildschirmfläche um das Datenbankfenster herum, soweit sie von keinem anderen geöffneten Objekt verdeckt wird.

6.1.2 Öffnen vom Desktop oder aus einem Ordner

Wenn Sie eine Verknüpfung zu einer Tabelle, Abfrage oder einem Formular erstellt und auf dem Desktop oder in einem Ordner der Festplatte abgelegt haben, öffnen Sie das betreffende Objekt durch Doppelklicken auf das Objektsymbol.

Um eine Verknüpfung zu einem Objekt zu erstellen, verfahren Sie wie folgt:

- Verkleinern und verschieben Sie die Anwendung Access so, daß der Desktop oder der Ordner im Explorer gleichzeitig zu sehen sind.
- Ziehen Sie aus dem Datenbankfenster das Objekt, zu dem Sie eine Verknüpfung erstellen wollen, auf den Desktop oder in den Ordner, wo Sie das Verknüpfungssymbol ablegen wollen.

  Oder:

- Klicken Sie mit der rechten Maustaste auf das gewünschte Objekt im Datenbankfenster und dann im Kontextmenü auf *Verknüpfung erstellen*. Dann öffnet sich das Dialogfeld *Verknüpfung erstellen*, in dem Sie entweder den (voreingestellten) Desktop oder einen Pfad zu einem Ordner wählen können.

6.2 Hinzufügen von Datensätzen

Jedes Datenblatt und jedes Formular stellt standardmäßig einen neuen leeren Datensatz zur Verfügung, in den Sie einfach neue Daten einzugeben brauchen. In der Datenblattansicht ist dies die letzte Zeile; sie ist mit einem Sternchen im Datensatzmarkierer gekennzeichnet:

Personal : Tabelle

PersonalCode	Nachname	Vorname	Sex	Anrede	Geburtsdatum	Einstellungsdatum	Straße
+ BIEDE	Biedermann	Sieglinde	w	Frau	22.06.63	01.07.83	Schillerstr. 7
+ GERSD	Gersdorff	Hubertus	m	Herr Dr.	04.06.63	01.08.85	Weidenstieg 89
+ LAUSE	Lauser	Hannelore	w	Frau Dr.	02.11.43	01.05.90	Fritzbalstr. 12
+ LEJEU	Lejeune	Micaela	w	Frau	24.12.58	01.11.92	
+ MAHLM	Mahlmann	Hermann	m	Herr	02.02.58	01.01.96	Fritz-Reuter-Weg
+ THIED	Thiedemann	Klaus	m	Herr	12.04.41	01.08.70	Tallinner Str. 1
						01.06.99	

Datensatz: 1 von 6

Bild 6.1: In die am linken Rand mit dem * gekennzeichnete Zeile können Sie einen neuen Datensatz eingeben.

Tastenkombination:
Strg + +

In der Formularansicht eines Formulars wie auch in der Datenblattansicht von Tabellen oder Abfragen bewegen Sie sich am besten mit der Symbol-Schaltfläche *Neuer Datensatz* (vgl. links nebenstehend) zum leeren Datensatz, in dessen Felder Sie dann Daten eingeben können. Darüber hinaus können Sie sich aber auch der Navigationsschaltflächen *Letzter Datensatz* und anschließend *Nächster Datensatz* bedienen, um zu einem neuen leeren Datensatz zu gelangen.

Falls Sie in einem Datenblatt oder Formular nur neue Datensätze anfügen wollen, können Sie mit dem Befehl *Daten eingeben* aus dem Menü Datensätze erreichen, daß Sie automatisch einen neuen Datensatz präsentiert bekommen, während die bestehenden Datensätze nicht angezeigt werden. Wenn das letzte Feld des neuen Datensatzes, in den Sie Werte eingegeben haben, den Fokus hat, drücken Sie die ⇆-Taste, um einen weiteren leeren Datensatz angezeigt zu bekommen, oder Sie blättern mit der Navigationsschaltfläche *Nächster Datensatz* zum nächsten leeren Datensatz. Um wieder alle Datensätze anzeigen zu lassen, wählen Sie den Befehl *Filter/Sortierung entfernen* aus dem Menü *Datensätze*.

Der Befehl *Daten eingeben* aus dem Menü *Datensätze* läßt sich auch gezielt nur für ein Unterformular anwenden: Geben Sie einem Steuerelement oder einer Datenzeile des Unterformulars den Fokus, und führen Sie den Befehl dann aus. Dann werden alle bis dahin ggf. im Unterformular angezeigten Datensätze ausgeblendet, und nur die Felder des neuen Datensatzes sind sichtbar. Sie heben auch diese Wirkung mit dem Befehl *Filter/Sortierung entfernen* aus dem Menü *Datensätze* wieder auf.

6.3 Bearbeiten von Datensätzen

6.3.1 Markieren von Daten

Sie bearbeiten die Daten eines Feldes, indem Sie diesem den Fokus geben und die gewünschten Änderungen vornehmen. Felder von Formularen und Datenblättern haben den Fokus, wenn sie markiert sind. Die allgemeinste Methode, einem Feld den Fokus zu geben, ist sicherlich, mit der Maus darauf zu klicken oder sich mit der ⇆-Taste von Feld zu Feld zu bewegen. Das Feld mit dem Fokus zeigt entweder einen blinkenden Cursor oder eine schwarze Markierung.

In einem Datenblatt markieren Sie ein ganzes Feld mit der Maus am schnellsten, wenn Sie auf den linken Feldrand von innen klicken; der Mauszeiger muß aussehen wie ein dickes weißes Balkenkreuz, vgl. nachfolgenden Bildausschnitt.

In einem Formular können Sie dieselbe Methode für Textfelder anwenden. Noch schneller markieren Sie hier jedoch, wenn Sie auf das zugehörige Bezeichnungsfeld klicken.

PersonalCode	Nachname	Vorname	Sex	Anrede	Geburtsdatum	Einstellungsdatum	Straße
+ BIEDE	Biedermann	Sieglinde	w	Frau	22.06.63	01.07.83	Schillerstr. 7
+ GERSD	Gersdorff	Hubertus	m	Herr Dr.	04.06.63	01.08.95	Weiderstieg 89
+ LAUSE	Lauser	Hannelore	w	Frau Dr.	02.11.43	01.06.90	Fröbelstr. 12
+ LEJEU	Lejeune	Micaela	w	Frau	24.12.58	01.11.92	
+ MAHLM	Mahlmann	Hermann	m	Herr	02.02.58	01.01.96	Fritz-Reuter-Weg
+ THIED	Thiedemann	Klaus	m	Herr	12.04.41	01.09.70	Tallinner Str. 1
						01.06.99	

Bild 6.2: In einem Datenblatt klicken Sie auf den linken Innenrand eines Feldes, um es zu markieren.

Bild 6.3: Bei einem Steuerelement (hier ein Textfeld) klicken Sie auf sein Bezeichnungsfeld (hier: Nachname), um es zu markieren.

Zum Bewegen im Formular und in der Datenblattansicht können Sie außerdem die folgenden Tastenkombinationen verwenden, weitere finden Sie in der Hilfe zum Stichwort *Tastenkombinationen*.

Navigationsmodus. Dieser Begriff spielt bei der folgenden Übersicht der *Tastenkombinationen* eine Rolle. Es ist der Zustand, in dem ein ganzes Feld markiert und der Cursor nicht sichtbar ist. Im Navigationsmodus können Sie sich mit Hilfe der Pfeiltasten zwischen den Feldern bewegen.

Tastenkombinationen zum Bewegen im Formular und in der Datenblattansicht

Gehen zu einem bestimmten Datensatz	Formular	Datenblatt
Anzeigen des Feldes *Datensatznummer*. Geben Sie dann die gewünschte Datensatznummer ein, und drücken Sie die ↵-Taste.	F5	F5

Bewegen zwischen Feldern und Datensätzen	Formular	Datenblatt
Bewegen zum nächsten Feld	[⇥]	[⇥], [↵] oder [→]
Bewegen zum vorherigen Feld	[⇧]+[⇥]	[⇧]+[⇥] oder [←]
Bewegen zum letzten Feld im aktuellen Datensatz, im Navigationsmodus	[Ende]	[Ende]
Bewegen zum letzten Feld im letzten Datensatz, im Navigationsmodus	[Strg]+[Ende]	[Strg]+[Ende]
Bewegen zum ersten Feld im aktuellen Datensatz, im Navigationsmodus	[Pos1]	[Pos1]
Bewegen zum ersten Feld im ersten Datensatz, im Navigationsmodus	[Strg]+[Pos1]	[Strg]+[Pos1]
Bewegen zum aktuellen Feld im nächsten Datensatz	[Strg]+[Bild↓]	[↓]
Bewegen zum aktuellen Feld im letzten Datensatz, im Navigationsmodus	[Strg]+[↓]	[Strg]+[↓]
Bewegen zum aktuellen Feld im vorherigen Datensatz	[Strg]+[Bild↑]	[↑]
Bewegen zum aktuellen Feld im ersten Datensatz, im Navigationsmodus	[Strg]+[↑]	[Strg]+[↑]

Die genaue Wirkung der Cursortasten hängt ferner von den Einstellungen ab, die Sie im Register *Tastatur* des Befehls *Optionen* aus dem Menü *Extras* vornehmen, vgl. Bild 6.4.

6.3.2 Datensatzindikatoren

Es gibt vier Indikatoren, die Ihnen den Bearbeitungszustand eines Datensatzes anzeigen. Die in der folgenden Übersicht wiedergegebenen Datensatzindikatoren werden in der Datenblattansicht am linken Rand des aktuellen Datensatzes angezeigt (vgl. z.B. oben Bild 6.2), in einem Formular auf dem Datensatzmarkierer, vgl. Bild 6.5.

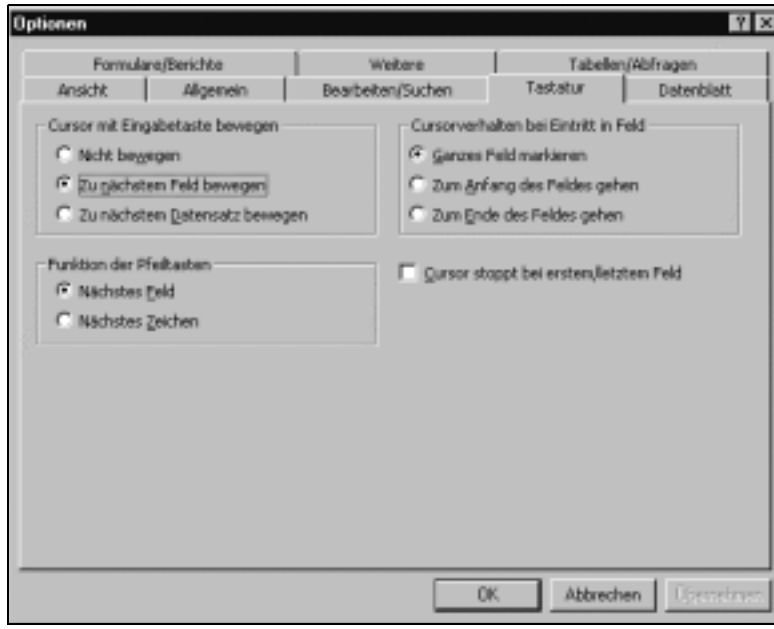


Bild 6.4: Mit der Registerkarte »Tastatur« aus dem Dialogfeld des Befehls »Extras Optionen«... können Sie die Wirkungsweise von Cursor und Pfeiltasten einstellen.

Bild 6.5: Am linken Rand des Formulars befindet sich der Datensatzmarkierer, auf dem einer von vier Datensatzindikatoren (hier das Bleistiftsymbol) angezeigt wird.

Datensatzindikatoren

▶	*Aktueller Datensatz*. Kennzeichnet den aktuellen Datensatz.
✎	*Bleistiftsymbol*. Der Datensatz wird gerade bearbeitet; die daran vorgenommenen Änderungen wurden noch nicht gespeichert. Um die Änderungen rückgängig zu machen, drücken Sie die Esc-Taste, ggf. mehrmals, bis das Bleistiftsymbol verschwindet und statt dessen das Symbol für den aktuellen Datensatz erscheint.

✳	*Neuer Datensatz*. Ein neuer, leerer, noch nicht gespeicherter Datensatz erscheint am unteren Rand des Datenblatts, damit Sie Daten eingeben können. Wenn dieser Datensatz den Fokus bekommt, weil Sie z.B. auf ein Feld klicken, wird das Symbol *Neuer Datensatz* durch das Symbol *Aktueller Datensatz* ersetzt.
⊘	*Gesperrter Datensatz*. Der Datensatz wird von einem anderen Benutzer gesperrt und kann daher nicht bearbeitet werden.

6.3.3 Eingeben von Daten

Sie geben Daten in der in Windows-Programmen üblichen Weise über die Tastatur ein. Dabei können Sie sich auch der in der folgenden Übersicht wiedergegebenen Tastenkombinationen bedienen.

Eingeben von Daten in der Datenblatt- oder Formularansicht

Aktion	*Tasten*
Kopieren der Markierung in die Zwischenablage	Strg + C
Einfügen der Markierung aus der Zwischenablage	Strg + V
Aktualisieren des Inhalts eines als Nachschlagefeld verwendeten Listenfeldes oder Kombinationsfelds	F9
Einfügen des aktuellen Datums	Strg + ;
Einfügen der aktuellen Uhrzeit	Strg + :
Einfügen des Standardwertes für ein Feld	Strg + Alt + ␣
Einfügen des Wertes aus demselben Feld des vorherigen Datensatzes	Strg + '
Hinzufügen eines neuen Datensatzes	Strg + +
Löschen des aktuellen Datensatzes	Strg + -
Speichern der Änderungen am aktuellen Datensatz	⇧ + ↵
Einfügen einer Zeilenschaltung in einem Text- oder Memo-Feld	Strg + ↵

Eingabe von Datumswerten in abgekürzter Form. Sie können Datumsangaben auch mit abgekürztem Jahresformat eingeben, dann wird die Eingabe im Hinblick auf die Jahrtausendwende seit der aktuellen Version Access 97 folgendermaßen interpretiert:

Eingabe	**Interpretation**
01.01.00 bis 31.12.29	01.01.2000 bis 31.12.2029
01.01.30 bis 31.12.99	01.01.1930 bis 31.12.1999

Aktualisieren von Daten in einem Nachschlagelistenfeld oder -kombinationsfeld. Zur Verbesserung der Leistung werden die Daten im Listen- oder Kombinationsfeld eines Nachschlagefelds, dessen Datenherkunft eine andere Tabelle ist, seit Version Access 97 nicht mehr automatisch aktualisiert, wenn die andere Tabelle verändert wird, während das Datenblatt oder das Formular geöffnet ist. (Beim Öffnen eines Datenblatts oder Formulars wird ein Nachschlagefeld dagegen stets automatisch aktualisiert.) Sie können die Daten im Nachschlagefeld durch Drücken der Taste F9 aktualisieren.

6.3.4 Rechtschreibprüfung

Manuelle Rechtschreibprüfung

Sie können die Texte einzelner Felder, Datensätze oder einer gesamten Tabelle oder Abfrage mit der *Rechtschreibprüfung* auf Fehler überprüfen lassen. Gehen Sie dazu folgendermaßen vor:

- Öffnen Sie ein Datenblatt oder Formular.
- Markieren Sie die Datensätze, Spalten, Felder oder den Text in einem Feld, deren/dessen Rechtschreibung Sie überprüfen möchten.
- Klicken Sie in der Symbolleiste auf die Symbol-Schaltfläche *Rechtschreibung*, vgl. links nebenstehend, oder drücken Sie F7, oder wählen Sie den Befehl *Rechtschreibung...* aus dem Menü *Extras*. Dann wird das Dialogfeld *Rechtschreibung* angezeigt, das Wörter, die weder im Office-Wörterbuch noch im Benutzer-Wörterbuch enthalten sind, anzeigt, die Sie dann u.a. korrigieren oder akzeptieren können.

AutoKorrektur: Automatisches Korrigieren von Text bei der Dateneingabe

Mit Hilfe der *AutoKorrektur* können Sie Texte, bei denen Ihnen häufig Schreibfehler passieren, automatisch korrigieren, oder Abkürzungen automatisch durch den ausgeschriebenen Text ersetzen lassen. Beispielsweise können Sie die Zeichenfolge »Beisiepl« automatisch beim Schreiben durch »Beispiel« ersetzen lassen oder »a2« durch »Access 2000«.

Sie steuern die AutoKorrektur mit Hilfe des gleichnamigen Dialogfeldes, vgl. Bild 6.6. Dieses rufen Sie auf mit dem Befehl *AutoKorrektur* aus dem Menü »*Extras*«.

AutoKorrektur korrigiert keinen Text, der bereits vor deren Aktivieren eingegeben wurde. Wenn im obigen Beispiel an einer bestimmten Stelle tatsächlich »a97« angezeigt werden soll, müssen Sie das Kontrollkästchen *Während der Eingabe ersetzen* vor dem Tippen der Zeichenfolge ausschalten und danach ggf. wieder einschalten.

Bild 6.6: Dialogfeld »AutoKorrektur«, in dem Sie die AutoKorrektur nach verschiedenen Kategorien ein- und ausschalten und spezielle Ersetzungen bearbeiten können.

6.3.5 Hyperlink-Adresse eingeben und bearbeiten

Eine Hyperlink-Adresse ist ein Pfad zu einem Objekt, einem Dokument, einer Web-Seite oder einem anderen Ziel. Eine Hyperlink-Adresse kann auch detailliertere Adreßinformationen (z.B. ein bestimmtes Datenbankobjekt, eine Word-Textmarke oder einen bestimmten Zellbereich oder Namen in einer Excel-Arbeitsmappe) enthalten. Wenn Sie auf einen zuvor eingegebenen und gespeicherten Hyperlink klicken, verwendet der Web-Browser (z.B. der Microsoft Internet Explorer oder der Browser von Netscape) oder Access die Hyperlink-Adresse, um zu dem angegebenen Ziel zu verzweigen.

An dieser Stelle wird nur dargestellt, wie Sie eine Hyperlink-Adresse eingeben und bearbeiten. Dies ist nur für Felder mit dem Datentyp *Hyperlink* möglich, die Sie zuvor in einer Tabelle oder einem Formular definiert haben müssen. Wie dies geschieht und alles weitere über den Umgang mit Hyperlinks erfahren Sie in Kap. 34, *Access im Inter- und Intranet*.

Hyperlink-Adresse neu einfügen

Um eine Hyperlink-Adresse in ein (bis dahin leeres) Feld neu einzufügen, gehen Sie folgendermaßen vor:

- Geben Sie dem Feld den Fokus, indem Sie beispielsweise darauf klicken.
- Geben Sie die Adresse über die Tastatur ein. In diesem Falle sind Sie für die korrekte Syntax verantwortlich. Sicherer und komfortabler arbeiten Sie daher, wenn Sie das Dialogfeld *Hyperlink* einfügen aufschlagen, um sich die Hyperlink-Adresse von Access einfügen zu lassen, wie im folgenden Aufzählungspunkt beschrieben.
- Klicken Sie auf die Symbol-Schaltfläche *Hyperlink einfügen*, oder wählen Sie den Befehl *Hyperlink...* aus dem Menü *Einfügen*, um das Dialogfeld *Hyperlink einfügen* anzuzeigen, vgl. Bild 6.7.

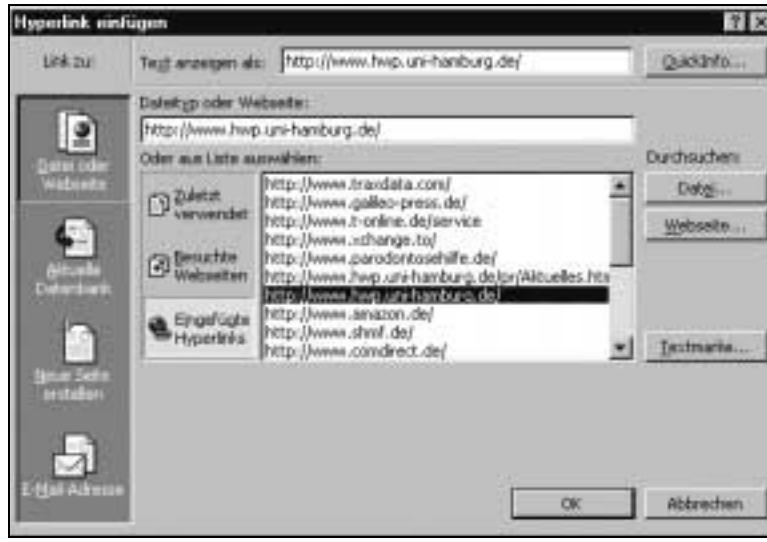


Bild 6.7: Dialogfeld »Hyperlink einfügen« mit aufgeschlagener Dropdown-Liste der zuletzt bearbeiteten Hyperlink-Adressen

- Wählen Sie eine Hyperlink-Adresse aus der Dropdown-Liste aus oder wählen Sie mit Hilfe einer der drei Schaltflächen *Datei*, *Webseite* oder *Textmarke* ein Dokument auf Ihrer Festplatte oder im Netzwerk. Access fügt dann den korrekten Pfad ein, vgl. Bild 6.8.

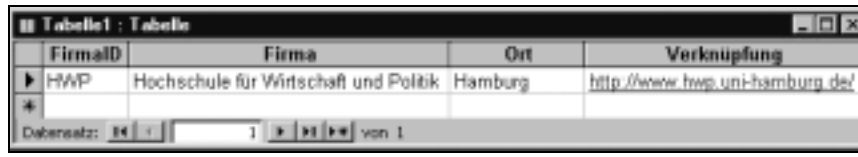


Bild 6.8: Die in Bild 6.7 ausgewählte Hyperlink-Adresse wurde eingefügt.

Hyperlink-Adresse bearbeiten

Um eine bereits eingegebene Hyperlink-Adresse zu bearbeiten, gehen Sie wie folgt vor:

- *Tastatur.* Geben Sie dem zu bearbeitenden Feld den Fokus, indem Sie sich z.B. mit der [⇆]-Taste dahin bewegen, und drücken Sie dann die Taste [F2], um das Feld in den Bearbeitungsmodus zu bringen. Wenn die Hyperlink-Adresse sich im Bearbeitungszustand befindet, kennzeichnet Access dies dadurch, daß der Adressentext von zwei Nummernzeichen # eingefaßt wird.
- *Maus*. Sie können zum Bearbeiten eines Feldes mit einem Hyperlink-Eintrag nicht mit der linken Maustaste auf das Feld klicken, weil dieses das Öffnen des mit der Adresse verbundenen Dokuments bedeuten würde. Daher müssen Sie mit der rechten Maustaste auf das Hyperlink-Feld klikken. Dann wird das Kontextmenü aufgeschlagen, und gleichzeitig erhält das Hyperlink-Feld den Fokus. Wählen Sie dann aus dem Kontextmenü den Befehl *Hyperlink* und aus dessen Unterbefehlen den geeigneten, vgl. Bild 6.9. Sie können auch das Kontextmenü durch Drücken der Taste [Esc] schließen und die Hyperlink-Adresse direkt im Feld bearbeiten, wobei Ihnen dann die Möglichkeiten des vorigen Aufzählungspunktes zur Verfügung stehen.

Bild 6.9: Unterbefehle des Befehls »Hyperlink« aus dem Kontextmenü

6.3.6 Änderungen rückgängig machen

Während sich ein Datensatz in der Bearbeitung befindet, ist sein Datensatzmarkierer mit dem Bleistiftsymbol gekennzeichnet. Solange dieses Symbol eingeblendet ist, sind die Änderungen noch nicht gespeichert worden. Daher können Sie versehentlich vorgenommene Bearbeitungen in dieser Situation noch durch Drücken der [Esc]-Taste, für Änderungen an mehreren Feldern ggf. durch mehrfaches Drücken, rückgängig machen.

Noch nicht gespeicherte Änderungen lassen sich auch durch Klicken auf die Symbol-Schaltfläche *Rückgängig: Eingabe* (vgl. links nebenstehend) oder den gleichnamigen Befehl aus dem Menü *Bearbeiten* rückgängig machen.

Wenn die Änderungen am Datensatz bereits gespeichert worden sind, was Sie daran erkennen, daß das Bleistiftsymbol nicht mehr angezeigt wird, können Sie versuchen, den ursprünglichen Datensatz wiederherzustellen, indem Sie auf die Symbol-Schaltfläche *Rückgängig: Gespeicherter Datensatz* (vgl. links nebenstehend) klicken oder den gleichnamigen Befehl aus dem Menü *Bearbeiten* wählen. Dieser Befehl steht allerdings nur unmittelbar nach dem Speichern eines Datensatzes zur Verfügung. Sobald Sie einen anderen Datensatz zu bearbeiten beginnen, und sei es nur durch Änderung eines Zeichens in einem Feld, kann das vorangegangene Speichern des Datensatzes nicht mehr aufgehoben werden. Das gleiche gilt auch, wenn Sie einen Filter anwenden oder zu einem anderen Fenster wechseln.

6.3.7 Prüfung von Datentyp und Gültigkeit

Sie können nur solche Datenbearbeitungen vornehmen, die dem Datentyp und/oder einer ggf. definierten Gültigkeitsregel – Eingabeformate und *Eingabe erforderlich* eingeschlossen – entsprechen.

Datentyp. Access prüft, ob ein eingegebener Wert zu dem in der zugrunde liegenden Tabelle für dieses Feld definierten Datentyp paßt, unabhängig davon, ob in der Tabelle oder im Formular eine Gültigkeitsregel festgelegt ist. Es ist unter keinen Umständen möglich, beispielsweise einem Feld mit dem Datentyp *Zahl* einen Text einzugeben. Die Prüfung erfolgt, wenn das Feld den Fokus verliert.

Gültigkeitsregeln. Diese können für Felder in Tabellen und Formularen sowie für den Datensatz einer Tabelle festgelegt werden. Wenn Sie für das Tabellenfeld und das entsprechende Steuerelement im Formular jeweils eine Gültigkeitsregel festgelegt haben, beide Regeln sich jedoch unterscheiden, muß der eingegebene Wert *jeder* der beiden Regeln genügen. Wenn beispielsweise für das Tabellenfeld die Gültigkeitsregel >100 festgelegt ist und für das entsprechende Steuerelement im Formular die Gültigkeitsregel <200, so können Sie in das Formular nur Werte eingeben, die zwischen 100 und 200 liegen. In die Datenblattansicht der Tabelle könnten Sie dagegen Werte über 100 eingeben.

Eingabe erforderlich. Wenn für ein Feld der Wert dieser Eigenschaft auf *Ja* festgelegt ist, muß das Feld oder entsprechende Steuerelement einen Wert enthalten, wenn der Datensatz gespeichert werden soll.

Eingabeformat. Für Felder können Eingabeformate festgelegt werden. Dann müssen die einzelnen Zeichen, die Sie eingeben, dem definierten Format entsprechen. Wenn Sie beispielsweise für ein Feld das Eingabeformat *00000* festgelegt haben, wie dies für Postleitzahlen sinnvoll sein kann, müssen Sie, wenn Sie einen Wert dafür eingeben, fünf Ziffern tippen. (Falls für dieses Feld *Eingabe erforderlich* mit *Nein* festgelegt wurde, könnten Sie es dagegen leer lassen.)

Gültigkeitsmakro. Wenn für die Eigenschaft *Vor Aktualisierung* eines Formulars ein Gültigkeitsmakro angegeben ist, kann es sein, daß die Speicherung des Datensatzes abgebrochen, d.h. nicht ausgeführt wird, weil eine der im Makro angegebenen Bedingungen verletzt ist. Zu Gültigkeitsmakros vgl. im einzelnen Kap. 28, *Ereignisabhängige Makros in Formularen und Steuerelementen*.

Falls Sie bei der Dateneingabe gegen eine dieser Regeln verstoßen haben, kommen Sie nur weiter, wenn Sie einen gültigen Wert eingeben oder die Bearbeitung mit der Esc-Taste abbrechen.

6.3.8 Wann können Felder nicht bearbeitet werden?

In den folgenden Fällen können Felder in Formularen oder Datenblättern nicht bearbeitet werden:

Datentyp AutoWert. Ein Feld mit dem Datentyp *AutoWert* wird von Access automatisch mit neuen Werten versorgt, und bestehende Werte dürfen nicht verändert werden, denn dies entspricht der Definition des Datentyps *AutoWert*.

Berechnetes Feld. Wenn ein Feld einen Ausdruck enthält, gibt dieser zwar im allgemeinen einen Wert wieder; dieser kann jedoch nicht bearbeitet werden. Derartige Felder sind nur in Formularen und Abfragen möglich, nicht jedoch in Tabellen.

Gesperrtes oder nicht aktiviertes Feld. Sie können für ein Feld im Formular die Eigenschaft *Gesperrt* auf Ja und/oder die Eigenschaft *Aktiviert* auf Nein einstellen; dann läßt sich dieses Feld nicht bearbeiten.

Gesperrter Datensatz. In einer Mehrbenutzerumgebung wird ein Datensatz, der von einem anderen Benutzer bearbeitet wird, gesperrt, wenn Sie zu diesem Datensatz wechseln. Dann läßt sich keines seiner Felder bearbeiten. Sie erkennen die Sperrung daran, daß der Datensatzmarkierer des Formulars oder eines Datensatzes im Datenblatt das Symbol *Gesperrter Datensatz* anzeigt, vgl. links nebenstehend.

Datenbasis Mehrtabellenabfrage. Bei bestimmten Mehrtabellenabfragen können einige Felder des Abfrageergebnisses und entsprechend der Formulare, die darauf basieren, nicht bearbeitet werden, vgl. genauer Kap. 11, *Auswahlabfragen*, Punkt 11.6, Datenaktualisierung bei Ein- und Mehrtabellenabfragen.

Felder in einem schreibgeschützten Formular oder in einer gesperrten Datenbank. Ein Formular besitzt die Eigenschaft *BearbeitungenZulassen*. Wenn diese Eigenschaft auf den Wert *Nein* eingestellt ist, können im Formular keine Daten bearbeitet werden. Wie diese Eigenschaft eingestellt ist, können Sie in der Entwurfsansicht des Formulars im Eigenschaftenfenster erkennen, wo Sie die Einstellung auch ändern können. Weiterhin kann auch die gesamte Datenbank schreibgeschützt geöffnet werden, vgl. Befehl *Datenbank öffnen* im Menü *Datei*, Dialogfeld *Öffnen*, Symbol-Schaltfläche

Befehle und Einstellungen, Kontextmenü-Befehl *Schreibgeschützt öffnen*. Schließlich ist es möglich, daß die Datenbank in einer Mehrbenutzerumgebung gesperrt ist. Auch in diesen Fällen ist eine Datenbearbeitung nicht möglich.

6.4 Speichern von Daten

Access speichert Datenänderungen stets automatisch, wenn eines der folgenden Ereignisse eintritt:

- Sie wechseln zu einem anderen Datensatz.
- Sie schließen das Formular oder Datenblatt.
- Sie ändern den Wert eines Primärschlüsselfeldes einer Mastertabelle. Dann wird der Wert des Fremdschlüsselfeldes in der Detailtabelle ebenfalls geändert, sofern Sie die Eigenschaft *Aktualisierungsweitergabe an Detailfeld* angekreuzt haben.
- Sie führen ausdrücklich den Befehl *Datensatz speichern* aus dem Menü *Datensätze* oder eine äquivalente Aktion aus.

Solange das Bleistiftsymbol noch angezeigt wird, hat keine Speicherung der aktuellen Änderung stattgefunden.

6.5 Datenblatt individuell gestalten

6.5.1 Optionen, die für alle Datenblätter gelten

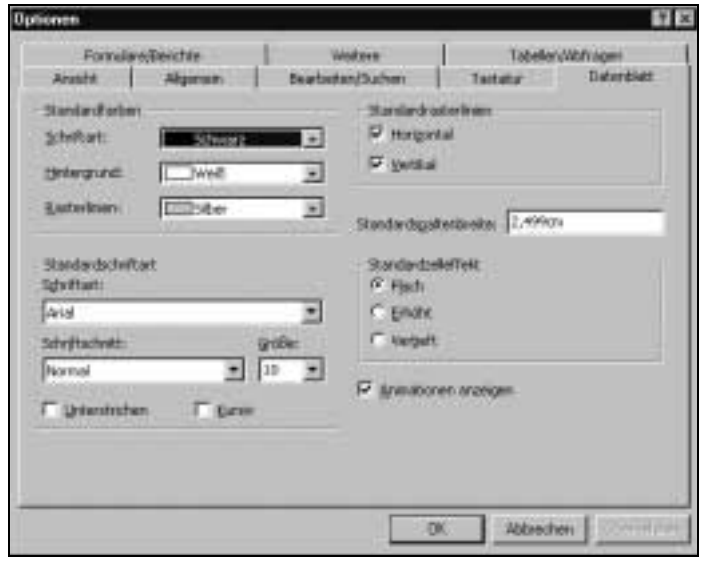


Bild 6.10: Dialogfeld des Befehls »Optionen« aus dem Menü »Extras, Register Datenblatt«

Im Dialogfeld des Befehls *Optionen* aus dem Menü *Extras*, Registerkarte *Datenblatt*, lassen sich verschiedene Optionen einstellen, die bestimmen, mit welchem Aussehen ein Datenblatt geöffnet wird, vgl. Bild 6.10.

6.5.2 Gestaltung einzelner Datenblätter

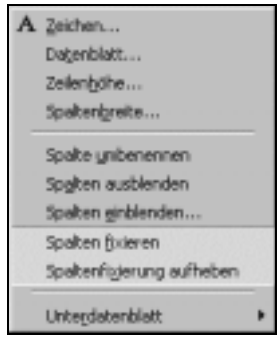


Bild 6.11: Befehle des Menüs »Format« bei aktivierter Datenblattansicht

Sie können ein Datenblatt in verschiedener Hinsicht individuell gestalten. Die im folgenden beschriebenen Änderungen können mit der Tabelle oder mit dem Formular gespeichert werden: Wählen Sie dazu Strg+S oder den Befehl *Speichern* aus dem Menü *Datei*.

Spaltenbreite ändern

Bild 6.12: Dialogfeld »Spaltenbreite«, das Sie mit dem Befehl »Spaltenbreite« aus dem Menü »Format« öffnen. Die Einstellung wirkt sich auf die markierte(n) Spalte(n) aus.

- **Menübefehl.** Markieren Sie die betreffende(n) Spalte(n), oder geben Sie einem Feld der betreffenden Spalte den Fokus. Schlagen Sie mit dem Befehl *Spaltenbreite...* aus dem Menü *Format* das Dialogfeld *Spaltenbreite* (vgl. Bild 6.12) auf, und geben Sie einen Wert für die Spaltenbreite ein. Wenn Sie zur Standardbreite zurückkehren wollen, klicken Sie auf das entsprechende Kontrollkästchen.

Oder:

- **Maus**. Ziehen Sie den rechten Rand des Spaltenkopfes nach links bzw. nach rechts, um die Spaltenbreite zu verkleinern bzw. zu vergrößern. Der Mauszeiger muß dabei aussehen wie ein Doppelpfeil in Ost-West-Richtung. Auch bei dieser Technik können Sie zuvor mehrere Spalten markieren, die dann alle gleich breit werden.

Spaltenbreite optimieren

Access kann die Spaltenbreite so anpassen, daß der längste in der Spalte vorkommende Eintrag gerade sichtbar ist. Gehen Sie dazu folgendermaßen vor:

- **Menübefehl**. Markieren Sie die betreffende(n) Spalte(n). Schlagen Sie mit dem Befehl *Spaltenbreite...* aus dem Menü *Format* das Dialogfeld *Spaltenbreite* (vgl. Bild 6.12) auf und wählen Sie die Schaltfläche *Anpassen*. Wenn Sie wieder zur Standardbreite zurückkehren wollen, klicken Sie auf das entsprechende Kontrollkästchen.

  Oder:

- **Maus**. Doppelklicken Sie auf den rechten Rand des Spaltenkopfes der betreffenden Spalte. Wenn Sie zuvor mehrere Spalten markiert haben, wird die Breite jeder einzelnen individuell optimiert.

Spalten verschieben

Sie können die Reihenfolge der einzelnen Spalten verändern, indem Sie eine oder mehrere Spalten verschieben. Gehen Sie folgendermaßen vor:

- Markieren Sie die Spalte, indem Sie auf ihren Kopf klicken. Mehrere Spalten markieren Sie durch Ziehen über die Spaltenköpfe oder mit gedrückter ⇧-Taste.
- Klicken Sie erneut auf die markierte(n) Spalte(n), und ziehen Sie in die gewünschte Richtung. Eine dunkel hervorgehobene senkrechte Linie zeigt an, an welcher Position die verschobene(n) Spalte(n) eingefügt wird (werden). In Bild 6.13 beispielsweise können Sie erkennen, daß die Spalte *Vorname* zwischen den Spalten *PersonalCode* und *Nachname* eingefügt werden wird, wenn in dieser Situation die linke Maustaste losgelassen wird.

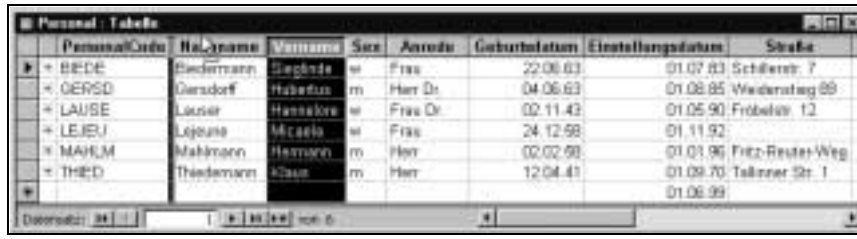


Bild 6.13: Die Spalte »Vorname« wird vor die Spalte »Nachname« verschoben.

Spalten aus- und einblenden

Spalten lassen sich aus- und einblenden. Verfahren Sie folgendermaßen:

Bild 6.14: Dialogfeld »Spalten einblenden«, mit dem Sie Spalten ein- und ausblenden können, hier mit ausgeblendeter Spalte »PersonalCode«.

- Verringern Sie die Spaltenbreite auf Null, um die markierte(n) Spalte(n) auszublenden.

  Oder:

- Wählen Sie den Befehl *Spalten Ausblenden* aus dem Menü *Format*. Dann wird/werden die markierte(n) Spalte(n) ausgeblendet.

  Oder:

- Wählen Sie den Befehl *Spalten einblenden* aus dem Menü *Format*. Aktivieren oder deaktivieren Sie im Dialogfeld *Spalten einblenden* (vgl. Bild 6.14) die Kontrollkästchen der aus- bzw. einzublendenden Spalten; Access blendet die Spalte(n) sofort bei noch geöffnetem Dialogfeld *Spalten einblenden* aus bzw. ein.

Spalten fixieren

Das Fixieren von Spalten kann sinnvoll sein, wenn Sie eine Datenblattansicht horizontal rollen: Fixierte Spalten erscheinen am linken Bildschirmrand und werden nicht mitgerollt. Auf diese Weise können Sie beispielsweise die zu einem Datensatz gehörigen Vor- und Nachnamen auch dann lesen, wenn Sie horizontal gerollt haben.

- Markieren Sie die zu fixierende(n) Spalte(n).
- Wählen Sie den Befehl *Spalten fixieren* aus dem Menü *Format*, um die Spalte(n) zu fixieren. Wählen Sie für die umgekehrte Wirkung den Befehl *Spaltenfixierung aufheben* aus dem Menü *Format*; in diesem Falle brauchen Sie zuvor keine Spalte zu markieren. Die Wirkung des Befehls *Spaltenfixierung aufheben* beschränkt sich darauf, die Fixierung aufzuheben,

die Stellung der zuvor fixierten Spalten wird nicht verändert, sie verbleiben also in der Position am linken Rand. Sie müßten diese Spalten ggf. verschieben, um ihnen die möglicherweise andere Position zuzuweisen, die sie vor dem Fixieren hatten.

In Bild 6.15 sehen Sie zwei Ansichten der Tabelle *Personal*, in der die beiden Spalten *Vorname* und *Nachname* fixiert wurden. Obwohl die Datensätze in der unteren Ansicht waagerecht gerollt wurden, so daß die weiter rechts liegenden Spalten zu sehen sind, verbleiben die beiden fixierten Spalten an ihrer Position am linken Fensterrand.

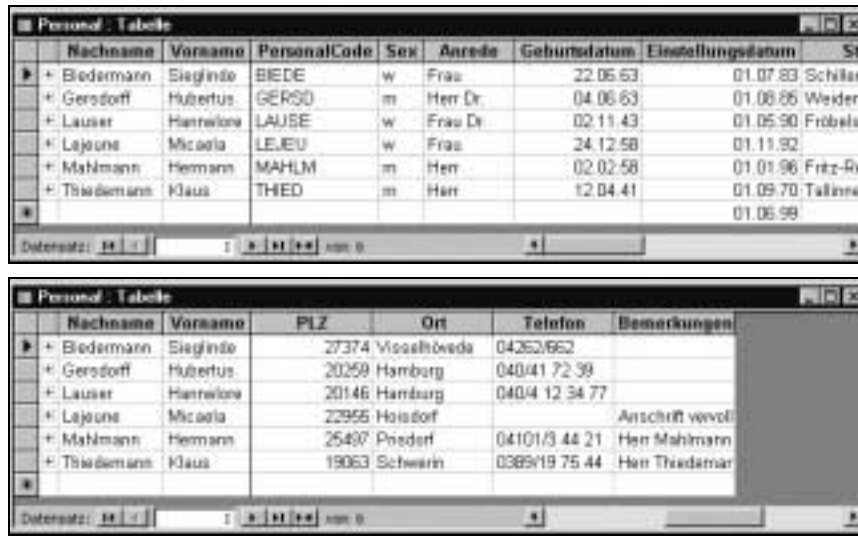


Bild 6.15: Die beiden Spalten Vorname und Nachname wurden fixiert, so daß diese auch nach dem waagerechten Rollen in der Position am linken Fensterrand verbleiben.

Wenn Sie ein Datenblatt drucken, in dem Spalten fixiert sind, wird die senkrechte Trennlinie zwischen den fixierten und den nicht fixierten Spalten hervorgehoben gedruckt. Um diesen Effekt zu vermeiden, müssen Sie die Spaltenfixierung vor dem Drucken aufheben.

Schrift ändern

Die Schrift kann nur für das Datenblatt insgesamt, nicht etwa nur für einzelne markierte Spalten oder Zeilen oder gar Zellen, geändert werden.

- Schlagen Sie mit dem Befehl *Zeichen...* aus dem Menü *Format* das Dialogfeld *Zeichen* auf, und wählen Sie die gewünschten Schriftmerkmale, vgl. Bild 6.16.

Wenn Sie einen größeren Schriftgrad wählen, vergrößert Access die Zeilenhöhe automatisch so, daß die Schrift in die Zeile hineinpaßt.

Bild 6.16: Dialogfeld »Zeichen«

Zeilenhöhe ändern

Die Zeilenhöhe kann nur für das Datenblatt insgesamt, nicht etwa gezielt für einzelne markierte Zeilen, geändert werden. In dieser Hinsicht unterscheiden sich Zeilenhöhe und Spaltenbreite. Gehen Sie folgendermaßen vor, um die Zeilenhöhe zu verändern:

Bild 6.17: Dialogfeld »Zeilenhöhe«

- Schlagen Sie mit dem Befehl *Zeilenhöhe* aus dem Menü *Format* das Dialogfeld *Zeilenhöhe* (vgl. Bild 6.17) auf, und geben Sie einen Wert für die Zeilenhöhe ein. Wenn Sie zur Standardhöhe zurückkehren wollen, kreuzen Sie das gleichnamige Kontrollkästchen an.

  Oder:

- Ziehen Sie eine beliebige Trennlinie zwischen den Zeilenköpfen.

Unterdatenblätter ein- und ausblenden

Für eine Mastertabelle lassen sich die entsprechenden Datensätze der Detailtabelle in einem Unterdatenblatt ein- und ausblenden. Dies geschieht gezielt für einzelne Datensätze der Mastertabelle mit Hilfe der am linken Rand des

Datenblatts angezeigten Erweiterungssymbole. In Bild 6.18 beispielsweise sind die zum zweiten Datensatz der Mastertabelle vorhandenen Datensätze der Detailtabelle in einem Unterdatenblatt eingeblendet, weil auf das entsprechende Erweiterungssymbol (Plus-Zeichen in der linken Randspalte) geklickt wurde.

Bild 6.18: Die zum zweiten Datensatz der Mastertabelle gehörenden Datensätze der Detailtabelle werden in einem Unterdatenblatt angezeigt.

Mit den Unterbefehlen *Alles einblenden*, *Alles ausblenden* und *Entfernen* steuern Sie nicht einzelne Unterdatenblätter einzelner Datensätze der Mastertabelle, sondern Sie bestimmen, ob alle möglichen Unterdatenblätter (*Alles einblenden*) oder kein Unterdatenblatt (*Alles ausblenden*) angezeigt wird oder ob die Erweiterungssymbole ein- oder ausgeblendet werden.

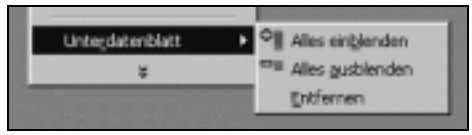


Bild 6.19: Unterbefehle des Befehls »Unterdatenblatt« aus dem Menü »Format«

Alle Unterdatenblätter einblenden

Mit dem Befehl *Format Unterdatenblatt Alles einblenden* veranlassen Sie, daß für jeden Datensatz der Mastertabelle die zugehörigen Datensätze der Detailtabelle jeweils in einem Unterdatenblatt angezeigt werden, vgl. Bild 6.20. Die Wirkung ist dieselbe, wie wenn für jeden Datensatz der Mastertabelle auf das Erweiterungssymbol geklickt worden wäre.

Alle Unterdatenblätter ausblenden

Mit dem Befehl *Format Unterdatenblatt Alles ausblenden* veranlassen Sie, daß kein Unterdatenblatt in der Mastertabelle angezeigt wird, gleichgültig, ob zuvor alle oder nur einzelne Unterdatenblätter angezeigt waren.

PersonalCode	Nachname	Vorname	Sex	Anrede	Geburtsdatum	Einstellungsdatum	Straße	PLZ
[illegible]	Biedermann	Sieglinde	w	Frau	22.06.63	01.07.89	Schillerstr. 7	2
GERSD	Oersdorff	Hubertus	m	Herr Dr.	04.06.63	01.08.85	Wiedenstieg 89	2
LAUSE	Lauser	Hannelore	w	Frau Dr.	02.11.43	01.05.90	Friesestr. 12	2
LEJEU	Lejeune	Micaela	w	Frau	24.12.68	01.11.92		2
MAHLM	Mahlmann	Hermann	m	Herr	02.02.58	01.01.96	Fritz-Reuter-Weg	2
THIED	Thiedemann	Klaus	m	Herr	12.04.41	01.09.70	Tallinner Str. 1	1

Projektnr	Projektbezeichnung	Auftragswert	Projektbeginn	Projektabschluß	Storniert
(AutoWert)		0,00 DM			☐
9	Dachausbau Hasenclev	70.000,00 DM	04.09.96		☐
10	Reparatur Stallungen M	150.000,00 DM	01.08.96		☐
(AutoWert)		0,00 DM			☐
7	Gartenanlage Meerburg	12.000,00 DM	02.07.96		☐
8	Stadtparkbepflanzung	5.000,00 DM	01.03.96		☑
(AutoWert)		0,00 DM			☐
(AutoWert)		0,00 DM			☐
2	Spielplatz Grundschule	1.000.000,00 DM	12.06.96		☐
4	Türdurchbruch Meier-Sc	3.600,00 DM	15.08.96		☑
5	Neubau Einfamilienhaus	370.000,00 DM	17.06.96		☐
6	Sielreparatur Erdmann	23.000,00 DM	05.05.96		☐
13	Grundschule Hermanns	120.000,00 DM	01.01.97		☐
(AutoWert)		0,00 DM			☐

Bild 6.20: Für jeden Datensatz der Mastertabelle wird ein Unterdatenblatt angezeigt.

Unterdatenblätter entfernen

Der Befehl *Format Unterdatenblatt Entfernen* bewirkt, daß in der betreffenden Mastertabelle keine Unterdatenblätter mehr angezeigt werden können: Nach dem Ausführen dieses Befehls werden in der Mastertabelle keine Erweiterungssymbole mehr angezeigt, und der Befehl *Unterdatenblatt* im Menü *Format* ist dann deaktiviert.

Unterdatenblätter (wieder) einfügen

Um für eine Mastertabelle die Möglichkeit zu schaffen, Unterdatenblätter einzublenden, gehen Sie wie folgt vor:

- Aktivieren sie die Datenblattansicht der Mastertabelle.
- Wählen Sie aus dem Menü *Einfügen* den Befehl *Unterdatenblatt...* Dann wird das Dialogfeld *Unterdatenblatt einfügen* angezeigt, vgl. Bild 6.21.
- Markieren Sie die Detailtabelle, deren Datensätze im Unterdatenblatt wiedergegeben werden sollen.
- WÄHLEN SIE ZU *VerknÜPFEN von* und *VerknÜPFEN NACH* den PRIMÄR- UND FREMDSCHLÜSSEL.
- BESTÄTIGEN SIE DAS DIALOGFELD *Unterdatenblatt einfügen* mit OK.

Mit den in den vorangehenden Aufzählungspunkten beschriebenen Schritten können Sie auch eine andere Detailtabelle für ein bestehendes Unterdatenblatt definieren. Dies kann sinnvoll sein für Mastertabellen, die mehr als eine Detailtabelle besitzen.

Bild 6.21: Dialogfeld »Unterdatenblatt« einfügen

6.6 Kopieren und Verschieben von Daten

Daten zwischen einzelnen Feldern kopieren und verschieben

Für das Kopieren und Verschieben zwischen beliebigen Eingabe- oder Bearbeitungsfeldern der gesamten Access-Ebene gelten die üblichen Windows-Techniken:

- Markieren Sie den zu kopierenden bzw. zu verschiebenden Teil.
- Drücken Sie die Tastenkombination [Strg]+[C] zum Kopieren bzw. Tastenkombination [Strg]+[X] zum Verschieben oder benutzen Sie die Befehle *Kopieren* bzw. *Ausschneiden* aus dem Menü *Bearbeiten*.
- Setzen Sie den Cursor an die Zielstelle, und drücken Sie die Tastenkombination [Strg]+[V] zum Einfügen, oder benutzen Sie den Befehl *Einfügen* aus dem Menü *Bearbeiten*.

Diese Technik gilt auch zwischen verschiedenen Windows-Anwendungsprogrammen. So können Sie beispielsweise den (ggf. teilweisen) Inhalt der Zelle eines Excel-Arbeitsblattes auf die angegebene Weise in ein Access-Tabellenfeld oder -Steuerelement kopieren oder verschieben.

Rechteckigen Bereich von Feldern kopieren oder verschieben

Sie können einen rechteckigen Bereich von Feldern innerhalb derselben Tabelle oder zwischen verschiedenen Tabellen (entsprechend für Abfragen) kopieren oder verschieben. Gehen Sie dazu folgendermaßen vor:

- Markieren Sie den Quellbereich. Sie markieren einen Bereich mit der Tastatur durch Betätigen einer der vier Pfeiltasten bei gedrückter [⇧]-Taste. Mit der Maus markieren Sie einen Bereich wie folgt: Klicken Sie

zunächst in die linke obere Ecke des zu markierenden Bereichs und dann, bei ebenfalls gedrückter ⇧-Taste, in die rechte untere Ecke.

- Drücken Sie die Tastenkombination Strg+C zum Kopieren bzw. Tastenkombination Strg+X zum Verschieben, oder benutzen Sie die Befehle *Kopieren* bzw. *Ausschneiden* aus dem Menü *Bearbeiten*.
- Markieren Sie den Zielbereich auf entsprechende Weise, wie Sie beim Quellbereich verfahren sind. Es reicht nicht aus, nur die linke obere Ecke des Zielbereichs zu markieren (genauere Hinweise dazu finden Sie im übernächsten Absatz).
- Drücken Sie die Tastenkombination Strg+V zum Einfügen, oder benutzen Sie den Befehl *Einfügen* aus dem Menü *Bearbeiten*.

Beim Kopieren werden die Zellen des Zielbereichs überschrieben. Wenn die Markierung von Quell- und Zielbereich unterschiedliche Größe hat, verfährt Access wie folgt:

Quellbereich umfaßt mehr Felder als Zielbereich. Es werden nur so viele Zeilen kopiert, wie im Zielbereich markiert sind. Wenn Sie beispielsweise Bild 6.22 betrachten, erkennen Sie, daß nur vier Zeilen in den Zielbereich der Tabelle *Lieferanten* kopiert werden sollen. Darauf weist auch die in Bild 6.22 wiedergegebene Meldung hin, die Access unmittelbar vor dem endgültigen Einfügen in die Tabelle *Lieferanten* einblendet: Obwohl sechs Zeilen ausgewählt und in die Zwischenablage kopiert wurden, werden nur vier Datensätze eingefügt (tatsächlich handelt es sich nicht um ganze Datensätze, sondern nur um Teile daraus).

Bild 6.22: Vor dem endgültigen Einfügen: Die Markierung im Quellbereich der Tabelle Kunden umfaßt sechs Zeilen und drei Spalten. Die Markierung im Zielbereich der Tabelle Lieferanten umfaßt vier Zeilen und zwei Spalten.

Quellbereich umfaßt weniger Felder als Zielbereich. In den Zielbereich werden nie mehr Felder eingefügt, als im Quellbereich markiert und kopiert wurden. Eine zu große Markierung im Zielbereich bleibt daher insoweit folgenlos.

Datensätze kopieren und verschieben

Es ist möglich, einzelne oder mehrere Datensätze zwischen Datenblättern und/oder Formularen zu kopieren und zu verschieben. Dabei reagiert Access unterschiedlich, je nachdem, ob Sie Datensätze in ein Datenblatt oder in ein Formular einfügen:

Datenblatt. In ein Datenblatt werden die Felder in der dortigen Reihenfolge der Spalten eingefügt, unabhängig von den Feldnamen. Daher müssen Sie sehr sorgfältig darauf achten, daß die Feldreihenfolge von Quelle und Ziel, deren Datentypen sowie relevante Feldeigenschaften wie *Eingabe erforderlich* übereinstimmen, damit kein Unsinn oder Fehler entsteht. Möglicherweise müssen Sie die Spalten im Quell- oder Zieldatenblatt vorher entsprechend verschieben und die Datentypen und Feldeigenschaften anpassen. Bei etwa auftretenden Fehlern wird der betreffende Datensatz nicht kopiert, vgl. die Meldung in Bild 6.23.

Bild 6.23: Meldung nach einem (teilweise) erfolglosen Einfügeversuch von Datensätzen

Formular. In ein Formular werden die Daten dagegen in Felder mit denselben Namen wie in der Quelle eingefügt, unabhängig von der Feldreihenfolge. Wenn Access in ein Formular einen Datensatz einfügen soll, der auch Felder enthält, deren Namen nicht mit denen im Zielformular übereinstimmen, wird die Meldung eingeblendet, die in Bild 6.24 wiedergegeben ist.

Um Datensätze zu kopieren oder zu verschieben, verfahren Sie ganz entsprechend wie im vorigen Punkt für Daten aus Feldern angegeben, mit dem Unterschied, daß als Quelle und Ziel jeweils ein ganzer Datensatz bzw. mehrere Datensätze zu markieren ist bzw. sind. In einem Datenblatt markieren Sie Datensätze, indem Sie auf den Datensatzmarkierer klicken oder, um mehrere zu markieren, über mehrere Datensatzmarkierer ziehen. Wenn Sie in der Zieltabelle weniger Zeilen markieren, als der Quellbereich umfaßt, werden entsprechend weniger Datensätze eingefügt.

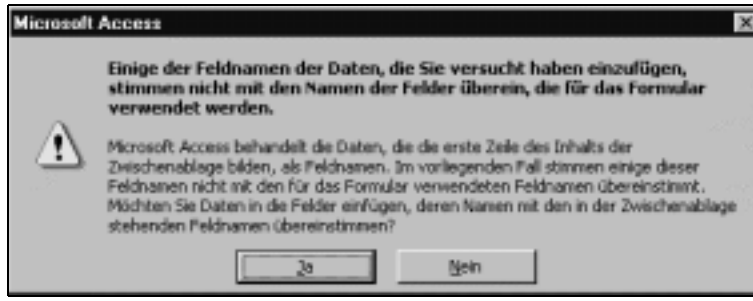


Bild 6.24: Es wurde versucht, in ein Formular einen Datensatz einzufügen, der Felder enthält, deren Namen im Zielformular keine Entsprechung haben. Wenn Sie mit OK bestätigen, werden die übereinstimmenden Felder eingefügt. In keinem Falle werden nicht übereinstimmende Felder eingefügt.

In einem Formular markieren Sie den aktuellen Datensatz, indem Sie auf den Datensatzmarkierer klicken. Auch in diesem Falle müssen Sie den Datensatz im Quell- und im Zielformular markieren.

Daten aus anderen Anwendungsprogrammen kopieren

Sie können Datensätze auch zwischen Access und anderen Anwendungsprogrammen kopieren und einfügen, und zwar in beide Richtungen. Verfahren Sie dabei analog wie in den vorangehenden Punkten beschrieben. Access versucht beim Einfügen in ein Formular auch in diesem Falle, nach übereinstimmenden Feldnamen einzufügen. Zumindest im folgenden Falle klappt dies auch sehr gut: Wenn Sie einen Bereich aus einem Excel-Arbeitsblatt, dessen erste Zeile Feldnamen enthält, in ein Formular kopieren, welches dieselben Feldnamen enthält, wird in der richtigen Weise eingefügt.

6.7 Löschen von Daten

Daten in Feldern löschen

- Markieren Sie den Feldinhalt oder einen Teil davon.
- Drücken Sie die [Entf]-Taste.

Datensätze löschen

- Markieren Sie den oder die Datensätze.
- Klicken Sie auf die Symbol-Schaltfläche *Datensatz löschen* (vgl. links nebenstehend), oder drücken Sie die [Entf]-Taste, und bestätigen Sie die dann eingeblendete Aufforderung oder brechen Sie ab.

Datensätze in Abfragen oder Formularen (z.B. Haupt- mit Unterformular für 1:n-Beziehung), denen eine definierte Tabellenbeziehung mit referentieller Integrität ohne die Option *Löschweitergabe an Detaildatensatz* zugrunde liegt, können nicht in einem Akt gelöscht werden. Wenn Sie dies versuchen, blendet Access die folgende Meldung ein (hier am Beispiel einer Detailtabelle mit dem Namen *Projekte*) (vgl. Bild 6.25):

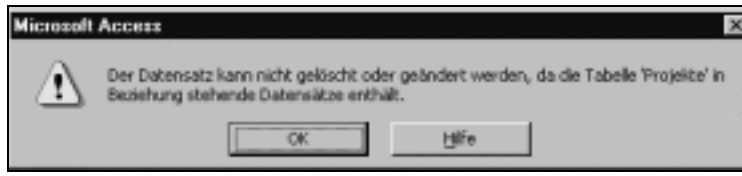


Bild 6.25: Es wurde versucht, einen Datensatz einer Mastertabelle zu löschen, für deren Beziehung zur Detailtabelle die Option »Löschweitergabe an Detaildatensatz« beim Festlegen der referentiellen Integrität nicht gewählt wurde.

In diesen und ähnlichen Fällen müssen Sie zunächst alle Datensätze der Detailtabelle, die mit dem zu löschenden Datensatz der Mastertabelle verknüpft sind, löschen, bevor Sie den Datensatz der Mastertabelle löschen können. Entsprechendes gilt im übrigen auch für Löschabfragen, vgl. Kap. 13, *Aktionsabfragen*, Punkt 13.2, *Löschabfragen*.

6.8 Suchen von Daten

Sie können *im aktuellen Feld* oder in allen Feldern des aktiven Formulars oder Datenblatts nach dem Vorkommen bestimmter Zeichenfolgen suchen lassen. Klicken Sie dazu, während das betreffende Feld den Fokus hat, auf die Schaltfläche *Suchen* (vgl. links nebenstehend) oder schlagen Sie mit dem Befehl *Suchen...* aus dem Menü *Bearbeiten* das in Bild 6.26 wiedergegebene Dialogfeld auf, geben Sie den Suchbegriff ein und wählen Sie die geeigneten Optionen. Beachten Sie, daß Sie in dem Dropdown-Listenfeld *Vergleichen* auch die Optionen *Anfang des Feldinhaltes* und *Teil des Feldinhaltes* wählen können.

Verwendung von Platzhalterzeichen

Im Suchbegriff dürfen Sie alle in Access gültigen Platzhalterzeichen verwenden. Dazu zählen nicht nur die Zeichen ?, * und #, die stellvertretend für ein Zeichen (?), eine Gruppe von Zeichen (*) und eine Ziffer (#) stehen, sondern eine ganze Reihe weiterer Möglichkeiten. Beispielsweise können Sie auch nach Zeichenfolgen suchen, die nicht vorkommen. Die Verwendung von Platzhalterzeichen wird vollständig in Kap. 39, *Ausdrücke, Funktionen und Operatoren*, erklärt. Detaillierte Erklärungen finden Sie auch in der Online-Hilfe unter dem Thema *Verwenden von Platzhalterzeichen in Zeichenfolgenvergleichen*.

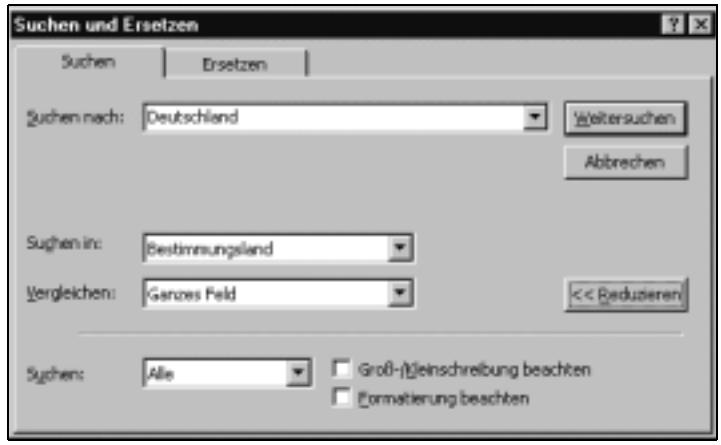


Bild 6.26: Dialogfeld »Suchen und Ersetzen«, Register »Suchen«

6.9 Suchen und Ersetzen von Daten

Sie können im aktuellen Feld oder in allen Feldern des Formulars oder Datenblatts mit dem Fokus nach dem Vorkommen bestimmter Zeichenfolgen suchen und diese durch eine andere Zeichenfolge ersetzen lassen. Verfahren Sie dazu, wie im vorangehenden Punkt 6.8, *Suchen von Daten*, für das Suchen beschrieben, wählen Sie indes im Dialogfeld *Suchen und Ersetzen* die Registerkarte *Ersetzen*. Sie können diese Registerkarte sofort aktivieren, wenn Sie das Dialogfeld *Suchen und Ersetzen* mit dem Befehl *Ersetzen...* aus dem Menü *Bearbeiten* aufschlagen.

Verwendung von Platzhalterzeichen

Im Suchbegriff des Dialogfeldes *Ersetzen* dürfen Sie, genau wie beim reinen Suchen, alle in Access gültigen Platzhalterzeichen verwenden. Die Verwendung von Platzhalterzeichen wird vollständig in Kap. 39, *Ausdrücke, Funktionen und Operatoren*, erklärt. Detaillierte Erklärungen finden Sie auch in der Online-Hilfe unter dem Thema *Verwenden von Platzhalterzeichen in Zeichenfolgenvergleichen*.

6.10 Filtern von Datensätzen

6.10.1 Überblick

FLTR
Gefiltert

Sie können die Datensätze der Datenblattansicht einer Tabelle oder Abfrage oder eines Formulars filtern. Auf diese Weise ist es beispielsweise möglich, die in einem Formular mit Adressen angezeigten Datensätze auf solche zu beschränken, deren Namen mit dem Buchstaben *A* beginnen oder, um ein anderes Beispiel zu nennen, auf Adressen aus den Orten München oder Kre-

feld, deren Namen mit »Müller-« beginnen. Ein solcher Filter kann jederzeit wieder entfernt werden, so daß dann wieder alle Datensätze angezeigt werden. Ferner können Sie jederzeit andere Kriterien festlegen. Wenn für ein Formular oder Datenblatt ein Filter wirksam ist, wird dies in der Statuszeile durch den Hinweis FLTR und in der Navigationsleiste des Datenblatts oder Formulars durch den Hinweis *Gefiltert* angezeigt.

Ein Filter wirkt stets nur zusammen mit dem Formular oder Datenblatt, für das er entworfen ist. Andere Formulare oder Datenblätter, die möglicherweise dieselbe Datenbasis verwenden, bleiben davon unberührt. (Ausnahme: Sie erstellen ein Formular auf einer gefilterten Datenblattansicht einer geöffneten Tabelle oder Abfrage. Dann werden in diesem Formular nur die gefilterten Datensätze wiedergegeben. Wenn Sie das Formular jedoch speichern und wieder öffnen, zeigt es alle Datensätze an.) Ein Filter läßt sich auch zusammen mit der Datenblattansicht einer Tabelle oder Abfrage oder dem Formular speichern. Beim nächsten Öffnen einer solchen Tabelle oder Abfrage in der Datenblattansicht oder eines solchen Formulars ist der Filter zunächst ausgeschaltet, und Sie können entscheiden, ob er wirksam werden oder ausgeschaltet bleiben soll.

Sie haben fünf verschiedene Möglichkeiten, Filter zu definieren:

- *Auswahlbasierter Filter.* Dies ist die schnellste und bequemste Art, einen Filter zu definieren: Sie markieren ein oder mehrere Feld(er) oder einen Teil eines Feldes, und der Inhalt der Markierung wird als Filterkriterium für das bzw. die Feld(er) verwendet.
- *Auswahlausschließender Filter.* Hierbei wird wie im vorangehenden Aufzählungspunkt verfahren, jedoch werden Datensätze angezeigt, die dem Inhalt der Markierung *nicht* entsprechen.
- *Filter nach*. Bei dieser Methode schlagen Sie das Kontextmenü für das Feld, nach dem Sie filtern wollen, auf, und geben in das Eingabefeld *Filter nach* im Kontextmenü das Kriterium ein.
- *Formularbasierter Filter.* Hierbei können Sie Kriterien in beliebiger Komplexität formulieren und diese bequem in ein Formular eingeben.
- *Spezialfilter/-sortierung*. Die Kriterienformulierung erfolgt hier in genau der gleichen Weise wie beim Entwerfen einer Abfrage, wobei Sie gleichzeitig verschiedene Sortierungen bestimmen können.

Verfügbare Befehle und Symbol-Schaltflächen

Zum Umgang mit Filtern stehen Ihnen die folgenden Unterbefehle des Befehls *Filter* aus dem Menü *Datensätze*, die Befehle des Kontextmenüs sowie vier Symbol-Schaltflächen zur Verfügung:

Unterbefehle des Befehls Filter

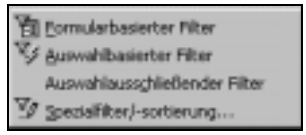


Bild 6.27: Unterbefehle des Befehls Filter aus dem Menü »Datensätze«

Befehle des Kontextmenüs

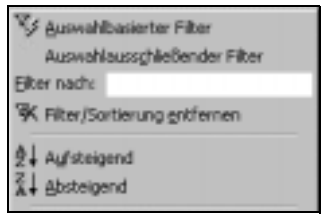


Bild 6.28: Befehle des Kontextmenüs, das angezeigt wird, wenn Sie mit der rechten Maustaste auf ein Feld klicken.

Symbol-Schaltflächen

	Auswahlbasierter Filter
	Formularbasierter Filter
	Filter/Sortierung anwenden
	Filter entfernen (die voranstehende Schaltfläche in gedrückter Form)
	Spezialfilter/Sortierung (standardmäßig nicht in der Symbolleiste enthalten)

Tabelle 6.1: Symbol-Schaltflächen zum Filtern

Filter für Datenblatt, Formular und Unterformular

Ein Filter läßt sich für die Datenblattansicht einer Tabelle oder Abfrage, ein Formular oder ein Unterformular in einem Formular definieren und anwenden.

Schritte zum Filtern

Um Datensätze zu filtern, gehen Sie im allgemeinen in der folgenden Schrittfolge vor:

- *Filterkriterien definieren*. Legen Sie einen auswahlbasierten, auswahlausschließenden, formularbasierten oder einen Spezialfilter fest. Im Falle des auswahlbasierten und auswahlausschließenden Filters wird dieser zugleich mit der Definition eingeschaltet, in den beiden anderen Fällen nicht.
- *Filter anwenden (einschalten)*. Um einen formularbasierten oder einen Spezialfilter wirksam werden zu lassen, müssen Sie diesen mit *Filter/Sortierung anwenden* einschalten.
- *Filter entfernen (ausschalten)*. Mit *Filter entfernen* setzen Sie einen wirksamen Filter zurück. Die Definitionsmerkmale des Filters werden dabei nicht gelöscht, solange das Datenblatt oder Formular geöffnet bleibt, so daß Sie diese jederzeit mit *Filter anwenden* wieder wirksam werden lassen können.
- *Filterkriterien ändern*. Sie können die Kriterien aller vier angeführten Filterarten jederzeit ändern. Danach müssen Sie den Filter ggf. wieder einschalten.

6.10.2 Auswahlbasierten Filter definieren

Um einen auswahlbasierten Filter festzulegen, gehen Sie folgendermaßen vor:

- Markieren Sie ein Feld, einen Teil der Zeichen in einem Feld oder in einer Datenblattansicht mehrere nebeneinander liegende Felder. Wenn Sie im aktuellen Feld gar nichts, nur das erste oder nur das letzte Zeichen markieren, wirkt dies für den Filter, wie wenn Sie das ganze Feld markieren.

- Klicken Sie auf die Symbol-Schaltfläche *Auswahlbasierter Filter* (vgl. links nebenstehend), oder wählen Sie den entsprechenden Menü- oder Kontextbefehl. Der Filter wird dann unmittelbar eingeschaltet, und es werden nur die Datensätze angezeigt, die in den betreffenden Feldern die Zeichenfolgen enthalten, welche beim Ausführen des Befehls markiert waren. Heben Sie die Filterwirkung ggf. wieder durch Klicken auf die Symbol-Schaltfläche *Filter entfernen* oder durch Wahl des entsprechenden Menübefehls auf.

Wenn Sie den Befehl *Auswahlbasierter Filter* noch einmal ausführen, während der vorige Filter eingeschaltet ist, wird das zuvor festgelegte Filterkriterium nicht entfernt, sondern es bleibt erhalten, und die beiden Filterkriterien sind dann durch das logische *Und* miteinander verbunden. Wenn Sie beispielsweise für die Tabelle *Bestellungen* der Datenbank Nordwind.mdb zunächst für das Feld *Bestimmungsland* einen Filter für *Deutschland* festlegen und anschließend für das Feld *Verkaufsberater* und die Markierung *Peacock, Margaret* erneut den Befehl ausführen, während der vorige Filter noch eingeschaltet ist, werden nur die Datensätze angezeigt, deren Bestimmungsland *Deutschland* und deren Verkaufsberaterin *Peacock, Margaret* ist. Entsprechendes gilt für weiteres mehrfaches Ausführen von *Auswahlbasierter Filter*.

Anders verhält sich Access, wenn Sie den Befehl *Auswahlbasierter Filter* noch einmal ausführen, während der vorige Filter mit *Filter entfernen* ausgeschaltet ist. Dann geht das zuvor definierte Filterkriterium verloren und es wird nur das neue wirksam.

6.10.3 Auswahlausschließenden Filter definieren

Zum Festlegen eines *auswahlausschließenden* Filters verfahren Sie analog, wie im vorigen Punkt für den auswahlbasierten Filter beschrieben. In diesem Falle steht Ihnen jedoch keine Symbol-Schaltfläche zur Verfügung. Wählen Sie daher nach dem Markieren von Zeichen in einem oder mehreren Feldern den Befehl *Auswahlausschließender Filter,* den Sie im Kontextmenü oder als Unterbefehl zum Befehl *Filter* im Menü *Datensätze* finden. Nachdem Sie diesen Filter festgelegt und damit implizit eingeschaltet haben, werden nur noch die Datensätze angezeigt, welche die markierten Zeichen in den betreffenden Feldern nicht enthalten. Auch diesen Filter setzen Sie mit der Symbol-Schaltfläche *Filter entfernen* oder dem entsprechenden Menübefehl zurück.

6.10.4 *Filter nach* definieren

Bei dieser Methode legen Sie ein Filterkriterium fest, indem Sie dieses in das Feld *Filter nach* des Kontextmenüs eingeben. Gehen Sie wie folgt vor:

- Klicken Sie mit der rechten Maustaste in der Datenblattansicht oder im Formular auf das Feld, für das Sie ein Kriterium angeben wollen. Dann wird das Kontextmenü aufgeschlagen, in dem das Eingabefeld *Filter nach* angeboten wird, vgl. oben Bild 6.28.
- Geben Sie in das Eingabefeld *Filter nach* das Kriterium ein, nach dem Sie filtern möchten.
- Drücken Sie die [↵]-Taste, um das Kontextmenü zu schließen und gleichzeitig den Filter wirksam werden zu lassen.

  Oder:

- Drücken Sie die [⇥]-Taste, um ein weiteres Kriterium für dasselbe Feld einzugeben. Wiederholen Sie diesen Schritt so oft, wie Sie weitere Kriterien eingeben möchten. Die so angegebenen Kriterien sind miteinander alle durch das logische *Und* verbunden. Daher macht diese Vorgehensweise im allgemeinen nur dann Sinn, wenn Sie statt einer Textkonstanten einen Ausdruck angeben, vgl. den folgenden Absatz

Kriterienausdrücke. Sie können in das Eingabefeld *Filter nach* des Kontextmenüs auch beliebige Ausdrücke eingeben, wie dies für Abfragen üblich ist (zu Abfragen vgl. ausführlich Kap. 11, *Auswahlabfragen*, Punkt 11.3*, Punkt Kriterien*). Um beispielsweise für die Datenblattansicht der Tabelle *Bestellungen* in der Datenbank *Nordwind.mdb* alle Bestellungen anzuzeigen, die nach Deutschland oder Frankreich gingen, schlagen Sie das Kontextmenü für das Feld *Bestimmungsland* auf, und geben Sie den Ausdruck

Deutschland oder Frankreich

ein. (Die eigentlich für die Textkonstanten *Deutschland* bzw. *Frankreich* erforderlichen Anführungszeichen dürfen Sie fortlassen, weil Access diese intern ergänzt.)

Auch diesen Filter setzen Sie mit der Symbol-Schaltfläche *Filter entfernen* oder dem entsprechenden Menübefehl zurück.

6.10.5 Formularbasierten Filter definieren

Im Vergleich zum auswahlbasierten Filter ist der formularbasierte etwas umständlicher zu handhaben, dafür bietet er jedoch wesentlich mehr Möglichkeiten, Filterkriterien zu formulieren. Insbesondere haben Sie bei einem formularbasierten Filter die Möglichkeit, Kriterien mit dem logischen *Oder* zu verbinden, was beim auswahlbasierten Filter nicht möglich ist. Der Umgang mit dem formularbasierten Filter soll am Beispiel der Tabelle *Projekte* aus der Datenbank *Projekte.mdb* erklärt werden. Der Filter soll so eingestellt werden, daß nur Projekte mit einem Auftragswert >100000, die von einem der beiden Mitarbeiter *Gersdorff* oder *Mahlmann* betreut werden, angezeigt werden. Gehen Sie folgendermaßen vor, um für diese Auswahl einen formularbasierten Filter einzustellen:

- Öffnen oder aktivieren Sie ggf. die Tabelle *Projekte* in der Datenbank *Projekte.mdb*.
- Klicken Sie auf die Symbol-Schaltfläche *Formularbasierter Filter* (vgl. links nebenstehend) oder wählen Sie den entsprechenden Menübefehl. Dann zeigt Access das Fenster *Projekte: Formularbasierter Filter* an, vgl. Bild 6.29. Es stellt ein Formular dar, mit dessen Hilfe Sie Filterkriterien definieren können.

Bild 6.29: Formular zur Eingabe von Filterkriterien für einen formularbasierten Filter

- Geben Sie in das Feld *Auftragswert* das Kriterium *>100000* ein. Klicken Sie in das Feld PersonalCode. Dann wird an dessen rechtem Rand das Symbol zum Aufschlagen einer Dropdown-Liste angezeigt. Schlagen Sie diese auf, und wählen Sie den Eintrag *mit dem Namen Gersdorff*, vgl. Bild 6.29.

- Klicken Sie am unteren Rand des Formularfensters auf die Registerkarte *Oder*. Dann wird für dieses Register eine leere Kriterienzeile angezeigt. Wählen Sie hier für das Feld *PersonalCode* den Eintrag für den Namen *Mahlmann*, und geben Sie in das Feld *Auftragswert* wiederum als Kriterium *>100000* ein.

Bild 6.30: Formular zur Eingabe von Filterkriterien, hier mit aktivierter Registerkarte für das erste »Oder«

- Klicken Sie auf die Symbol-Schaltfläche *Filter/Sortierung anwenden*, vgl. links nebenstehend. Dann wird wieder die Datenblattansicht der Tabelle *Projekte* angezeigt, in der vier Datensätze wiedergegeben werden, vgl. Bild 6.31.

Bild 6.31: Gefilterte Datensätze in der Datenblattansicht der Tabelle »Bestellungen«

Kriterienausdrücke. Im Formular zur Eingabe von Filterkriterien können für die einzelnen Felder beliebige Ausdrücke – ggf. auch sehr komplexe – eingegeben werden. Um beispielsweise nur Datensätze anzuzeigen, deren Projektbeginn zwischen dem 1.3.96 und dem 2.7.96 liegt, können Sie in das Feld *Bestelldatum* den Ausdruck

Zwischen #01.03.96# Und #01.07.96#

eingeben. Sie können hier dieselben Ausdrücke für Kriterien angeben, wie dies für Auswahlabfragen möglich ist. Die Formulierung von Ausdrücken für Kriterien wird ausführlich in Kap. 11, *Auswahlabfragen*, behandelt.

Kriterien bearbeiten

Die Kriterien eines formularbasierten Filters können jederzeit bearbeitet werden. Klicken Sie dazu erneut auf die Symbol-Schaltfläche *Formularbasierter Filter*, um das Kriterienformular anzuzeigen, in dem Sie dann die Bearbeitungen vornehmen können. Nach der Bearbeitung klicken Sie auf die Symbol-Schaltfläche *Filter/Sortierung anwenden*, um die den bearbeiteten Kriterien entsprechenden Datensätze anzuzeigen. Beim Bearbeiten von Kriterien im Kriterienformular kann es sinnvoll sein, sämtliche zuvor definierten Kriterien zu löschen. Dazu steht Ihnen bei aktiviertem Kriterienformular die Symbol-Schaltfläche *Alles löschen* zur Verfügung, vgl. links nebenstehend.

Wenn Sie vor dem Wählen des Befehls *Formularbasierter Filter* einen auswahlbasierten oder auswahlausschließenden Filter verwendet haben, erscheinen dessen Kriterien im Kriterienformular für den formularbasierten Filter. Sie müssen diese Einträge ggf. löschen oder bearbeiten, um den formularbasierten Filter entsprechend Ihren Vorstellungen richtig einzustellen. Beachten Sie vor allem, daß im Kriterienformular nicht alle Felder gleichzeitig angezeigt werden und daher möglicherweise ein durch einen auswahlbasierten Filter bereits vorhandenes Kriterium verdeckt ist.

Kriterienformular für ein Formular

Die bisherige Darstellung ging von einem formularbasierten Filter für die Datenblattansicht einer Tabelle oder Abfrage aus. Wenn Sie den Befehl *Formularbasierter Filter* für ein Formular wählen, hat das Kriterienformular ein anderes Aussehen, vgl. Bild 6.32, in dem das Kriterienformular für das Formular *Personal und Projekte* der Datenbank *Projekte.mdb* wiedergegeben wird. Die Felder sind darin so angeordnet wie im zugrunde liegenden Formular. Dies erleichtert den Überblick, weil sie, anders als im Falle der Datenblattansicht, die Kriterienfelder nicht in der Waagerechten zu rollen brauchen. Im übrigen aber gilt für den Umgang mit einem formularbasierten Filter für ein Formular das gleiche wie für die Datenblattansicht.

6.10.6 Spezialfilter/-sortierung definieren

Bei der Verwendung von *Spezialfilter/-sortierung* haben Sie die differenziertesten Möglichkeiten, Filterkriterien anzugeben. Dabei können Sie gleichzeitig Sortierreihenfolgen festlegen. Gehen Sie folgendermaßen vor, um einen Spezialfilter mit ggf. Sortierung für eine geöffnete Datenblattansicht einer Tabelle oder Abfrage oder für ein geöffnetes Formular festzulegen:

Bild 6.32: Kriterienformular für das Formular »Personal« und »Projekte« der Datenbank »Projekte.mdb«

- Wählen Sie den Unterbefehl *Spezialfilter/-sortierung* zum Befehl *Filter* aus dem Menü *Datensätze*. Die links nebenstehend wiedergegebene Symbol-Schaltfläche *Spezialfilter/-sortierung* steht in den Symbolleisten für Datenblattansicht und Formular standardmäßig nicht zur Verfügung. Sie können sich die Symbol-Schaltfläche jedoch in diese Symbolleisten einfügen, vgl. dazu im einzelnen Kap. 24, *Menüleisten, Symbolleisten und Kontextmenüs verwalten*. Nach der Wahl dieses Befehls wird der Entwurfsbereich für den Filter und die Sortierung angezeigt, der dem Entwurfsbereich einer Abfrage sehr ähnlich ist, vgl. Bild 6.33.
- Fügen Sie dem Entwurfsbereich das Feld oder die Felder hinzu, für das oder die Sie Kriterien angeben wollen. Zu diesem Zweck ziehen Sie jeweils ein Feld aus der Tabellenliste auf eine Spalte des Entwurfsbereichs oder doppelklicken auf das Feld in der Tabellenliste. Die Reihenfolge, in der Sie die Felder dem Entwurfsbereich hinzufügen, ist für die Kriterien gleichgültig, nicht jedoch für das Sortieren.
- Geben Sie in der Zelle *Kriterien* für die Felder, die Sie dem Entwurfsbereich hinzugefügt haben, jeweils den Wert oder Ausdruck ein, nach dem Sie filtern wollen.

- Um auch eine Sortierreihenfolge festzulegen, klicken Sie auf die Zelle *Sortierung* für das betreffende Feld, und wählen Sie in der Dropdown-Liste *Aufsteigend* oder *Absteigend*. Wenn Sie für mehrere Felder eine Sortierreihenfolge bestimmen, sortiert Access zuerst das äußere, linke Feld im Entwurfsbereich, dann das Feld rechts daneben usw.

- Klicken Sie auf die Symbol-Schaltfläche *Filter/Sortierung anwenden*, vgl. links nebenstehend. Dann wird wieder die Datenblattansicht oder das Formular angezeigt, nunmehr mit gefilterten Datensätzen.

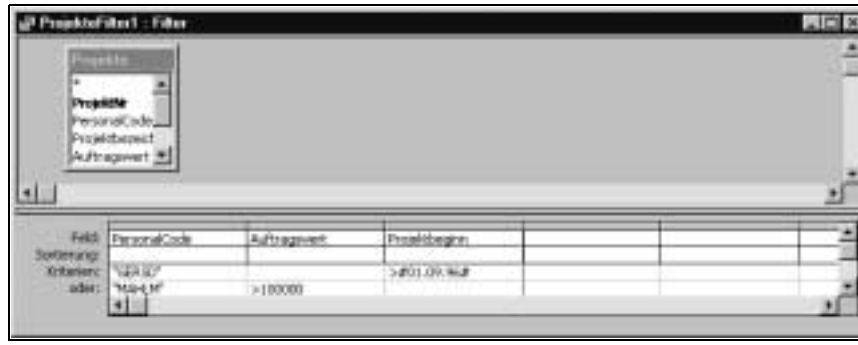


Bild 6.33: Spezialfilter definieren

Kriterien bearbeiten

Auch die Kriterien eines Spezialfilters können jederzeit bearbeitet werden. Wählen Sie dazu erneut den Befehl *Spezialfilter/-sortierung*, um den Entwurfsbereich anzuzeigen, in dem Sie dann die Bearbeitungen vornehmen können. Nach der Bearbeitung klicken Sie wieder auf die Symbol-Schaltfläche *Filter/Sortierung anwenden*, um die den bearbeiteten Kriterien entsprechenden Datensätze anzuzeigen. Beim Bearbeiten von Kriterien im Entwurfsbereich kann es sinnvoll sein, sämtliche zuvor definierten Kriterien zu löschen. Dazu steht Ihnen bei aktiviertem Entwurfsbereich auch hier die Symbol-Schaltfläche *Alles löschen* zur Verfügung, vgl. links nebenstehend.

Wenn Sie vor dem Wählen des Befehls *Spezialfilter/-sortierung* einen anderen Filter verwendet haben, erscheinen dessen Kriterien im Entwurfsbereich für den Spezialfilter. Sie müssen diese Einträge ggf. löschen oder bearbeiten, um den Spezialfilter entsprechend Ihren Vorstellungen richtig einzustellen.

6.10.7 Kumulierte Wirkung mehrerer Filter

Die vier in den vorangehenden Abschnitten beschriebenen Filterarten,

- *Auswahlbasierter Filter*
- *Auswahlausschließender Filter*
- *Filter nach*

- *Formularbasierter Filter*
- *Spezialfilter/-sortierung*

stellen nur unterschiedliche Wege dar, einen Filter für ein Datenblatt oder ein Formular zu erstellen. Jeder dieser Wege führt dazu, daß Access sich die festgelegten Kriterien merkt, solange die Datenblattansicht oder das Formular geöffnet bleibt. (Darüber hinaus können sie auch zusammen mit der Tabelle, der Abfrage oder dem Formular gespeichert werden, vgl. den folgenden Punkt *Filter speichern*.) Daher sind zuvor mit einer bestimmten Filterart festgelegte Kriterien wirksam, wenn Sie für dieselbe Datenblattansicht oder dasselbe Formular eine andere Filterart festlegen wollen. Im Falle eines formularbasierten oder eines Spezialfilters wird dies auch äußerlich sichtbar: Wenn Sie beispielsweise zuvor einen formularbasierten Filter festgelegt hatten und dann den Befehl *Spezialfilter/-sortierung* wählen, enthält der Entwurfsbereich für *Spezialfilter/-sortierung* bereits die anderen Kriterien. Es gilt allgemein:

Die Kriterien der mit den verschiedenen Methoden erzeugten Filter werden kumuliert. Daher ist es ohne weiteres möglich, beispielsweise zunächst einen auswahlbasierten Filter festzulegen, weil diese Methode besonders komfortabel ist, und diesen dann mit der Methode *Formularbasierter Filter* oder *Spezialfilter/-sortierung* zu bearbeiten. Die Aussage über die Kumulierungswirkung der verschiedenen Filterarten gilt uneingeschränkt, wenn der Filter eingeschaltet ist. Wenn Sie dagegen bei ausgeschaltetem Filter mit einer der Methoden *Auswahlbasierter Filter*, *Auswahlausschließender Filter* oder *Filter nach* einen Filter festlegen, schafft dies neue Anfangsbedingungen, und ggf. zuvor festgelegte Kriterien gehen verloren.

6.10.8 Filter speichern

Filter mit dem Datenblatt oder Formular speichern

Gleichgültig, auf welchem Wege Sie einen Filter definiert und angewandt haben und unabhängig davon, ob er gerade ein- oder ausgeschaltet ist: Sie können einen Filter zusammen mit der Tabelle, Abfrage oder dem Formular speichern. Access verhält sich dabei im Falle einer Tabelle oder Abfrage einerseits und eines Formulars andererseits allerdings etwas unterschiedlich:

- *Tabelle oder Abfrage*. Wenn Sie die Datenblattansicht einer Tabelle oder Abfrage, für die ein Filter definiert wurde, schließen, werden Sie von Access gefragt, ob Sie die am Entwurf der Tabelle oder Abfrage vorgenommenen Änderungen speichern möchten, vgl. Bild 6.34. Bestätigen Sie das Dialogfeld mit Ja, um den Filter mit dem Objekt zu speichern. Beachten Sie jedoch dabei, daß dann auch ggf. andere am Entwurf der Datenblattansicht vorgenommenen Änderungen (z.B. das Ausblenden einer Spalte oder eine Veränderung des Schriftgrades) gleichzeitig mit gespeichert werden, denn Access bietet leider keine gezielte Speicherungsmög-

lichkeit allein für die Filtermerkmale an. Verneinen Sie umgekehrt die Frage des Dialogfeldes, werden die Filtermerkmale nicht gespeichert, aber auch keine anderen ggf. vorgenommenen Änderungen am Entwurf der Datenblattansicht.

Bild 6.34: Speicherabfrage beim Schließen der Tabelle »Projekte«

- *Formular.* Der Filter für ein Formular wird automatisch, d.h. ohne Bestätigungsfrage an den Benutzer, gespeichert.

Wenn Sie eine Tabelle oder Abfrage in der Datenblattansicht oder ein Formular öffnen, für die/das ein Filter gespeichert ist, werden zunächst alle Datensätze ungefiltert angezeigt. Sie müssen den Filter ausdrücklich mit *Filter anwenden* einschalten, damit er wirksam wird.

Filter als Abfrage speichern

Da Access die von Ihnen festgelegten Filtermerkmale intern ohnehin als Auswahlabfrage verwaltet, ist es auch ohne weiteres möglich, einen Filter als eigenständige Abfrage zu speichern. Diese steht dann wie jede andere Abfrage unabhängig von dem Datenblatt oder Formular, für das der Filter ursprünglich definiert wurde, zur Verfügung. Gehen Sie folgendermaßen vor, um einen Filter als Abfrage zu speichern:

- Öffnen Sie ggf. die Tabelle, die Abfrage oder das Formular, und legen Sie die Filtermerkmale fest. Dabei ist es gleichgültig, welche der verschiedenen Methoden zum Festlegen eines Filters Sie verwenden.
- Klicken Sie auf eine der Symbol-Schaltflächen *Formularbasierter Filter* oder *Spezialfilter/-sortierung*, oder wählen Sie einen diesen entsprechenden Befehl.

- Klicken Sie im Kriterienformular des formularbasierten Filters oder im Entwurfsbereich des Spezialfilters auf die Symbol-Schaltfläche *Als Abfrage speichern*, vgl. links nebenstehend, oder wählen Sie den gleichlautenden Befehl aus dem Menü *Datei*.
- Geben Sie im Dialogfeld *Als Abfrage speichern* einen Namen für die Abfrage an, und bestätigen Sie mit OK.

Gespeicherte Abfrage als Filter verwenden

Sie können für die Datenblattansicht einer Tabelle oder Abfrage oder für ein Formular eine gespeicherte Abfrage als Filter verwenden. Dabei muß es sich nicht zwingend um eine Abfrage handeln, die als vormaliger Filter gespeichert wurde, sondern Sie können dafür im Prinzip jede passende Abfrage verwenden. Mit *passend* ist gemeint, daß die Abfrage den Feldern im zu filternden Objekt entsprechen muß. Sie können z.B. keine Abfrage als Filter für ein Formular heranziehen, der eine ganz andere Tabelle zugrunde liegt als dem Formular. Gehen Sie folgendermaßen vor, um eine Abfrage als Filter zu verwenden:

- Öffnen Sie ggf. die Tabelle, die Abfrage oder das Formular.
- Klicken Sie auf eine der Symbol-Schaltflächen *Formularbasierter Filter* oder *Spezialfilter/-sortierung*, oder wählen Sie einen diesen entsprechenden Befehl.
- Klicken Sie im Kriterienformular des formularbasierten Filters oder im Entwurfsbereich des Spezialfilters auf die Symbol-Schaltfläche *Von Abfrage laden*, vgl. links nebenstehend, oder wählen Sie den gleichlautenden Befehl aus dem Menü *Datei*.

6.10.9 Gespeicherten Filter löschen

Einen mit einem Objekt gespeicherten Filter können Sie auf die folgende Weise dauerhaft löschen:

- Öffnen Sie die Tabelle oder Abfrage oder das Formular in der Datenblattansicht bzw. der Formularansicht.
- Klicken Sie auf eine der Symbol-Schaltflächen *Formularbasierter Filter* oder *Spezialfilter/-sortierung*, oder wählen Sie einen diesen entsprechenden Befehl.
- Klicken Sie im Kriterienformular des formularbasierten Filters oder im Entwurfsbereich des Spezialfilters auf die Symbol-Schaltfläche *Alles löschen*, vgl. links nebenstehend.
- Klicken Sie im Kriterienformular des formularbasierten Filters oder im Entwurfsbereich des Spezialfilters auf die Symbol-Schaltfläche *Filter anwenden*, vgl. links nebenstehend. Dann wird die Datenblattansicht bzw. das Formular wieder angezeigt und darin alle Datensätze ungefiltert.
- Schließen Sie die Datenblattansicht oder das Formular, und verneinen Sie im Falle der Datenblattansicht die Frage, ob Änderungen am Entwurf gespeichert werden sollen.

6.11 Sortieren

Sie haben zwei Möglichkeiten, die Datensätze einer Datenblattansicht oder eines Formulars zu sortieren: Mit den beiden Symbol-Schaltflächen *Aufsteigend* und *Absteigend* bzw. ihren gleichlautenden (Unter-)Befehlen im Menü *Datensätze* und im Kontextmenü können Sie *jeweils nach einem Feld* sortieren lassen. Mit der Symbol-Schaltfläche *Spezialfilter/-sortierung* bzw. ihrem gleichlautenden (Unter-)Befehl im Menü *Datensätze* und im Kontextmenü können Sie auch nach mehreren Feldern sortieren lassen.

6.11.1 Sortierung erstellen

Symbol-Schaltflächen und Befehle *Aufsteigend und Absteigend*

Gehen Sie folgendermaßen vor, um auf diese Weise zu sortieren:

- *Aufsteigend*. Geben Sie einem Feld den Fokus, nach dessen Werten Sie aufsteigend sortieren wollen, und klicken Sie auf die Symbol-Schaltfläche *Aufsteigend* (vgl. links nebenstehend), oder wählen Sie den gleichlautenden (Unter-)Befehl aus dem Menü *Datensätze* oder dem Kontextmenü.

- *Absteigend*. Verfahren Sie entsprechend wie im voranstehenden Aufzählungspunkt.

Eine Sortierung, die Sie wie vorstehend beschrieben veranlaßt haben, wird von Access intern als ein (zusätzliches) Filtermerkmal verarbeitet. Daher erscheint jede irgendwie veranlaßte Sortierung auch im Entwurfsbereich von Spezialfilter/-sortierung, vgl. den folgenden Punkt.

Spezialfilter/-sortierung

Diese Methode wurde bereits oben in diesem Kapitel (vgl. Punkt 6.10.6, *Spezialfilter/-sortierung definieren*) als Filtermethode beschrieben. Dort wurde auch gezeigt, wie Sie für ein oder mehrere Felder Sortierungen festlegen. Wenn Sie für mehrere Felder eine Sortierreihenfolge bestimmen, sortiert Access zuerst das äußere, linke Feld im Entwurfsbereich, dann das Feld rechts daneben usw.

6.11.2 Sortierung entfernen (ausschalten)

Anders als ein Filter, dessen Wirkung Sie mit *Filter entfernen* (vorübergehend) ausschalten können, läßt sich eine Sortierung nicht durch einen speziell dafür vorgesehenen Befehl ausschalten. Sie müssen für diesen Zweck vielmehr den folgenden Weg gehen:

- Öffnen Sie mit *Spezialfilter/-sortierung* den zugehörigen Entwurfsbereich.
- Löschen Sie für die Felder, nach denen keine Sortierung erfolgen soll, im jeweiligen Feld *Sortierung* die dort stehenden Einträge *Aufsteigend* oder *Absteigend*.

- Klicken Sie auf die Symbol-Schaltfläche *Filter anwenden*, um diese Änderungen wirksam werden zu lassen und zur Datenblatt- oder Formularansicht zurückzukehren.

6.11.3 Sortierung speichern und gespeicherte Sortierung entfernen

Da, wie im vorangehenden Punkt beschrieben, Sortierungen von Access als Filter- bzw. Abfragemerkmale behandelt werden, gilt bezüglich ihrer Speicherung mit der Tabelle, der Abfrage oder dem Formular dasselbe, was oben für das Speichern von Filtern gesagt wurde, vgl. Punkt 6.10.8, *Filter speichern*.

6.12 Aktualisierung der Datenanzeige in Formularen und Datenblättern

Beim Öffnen von Tabellen, Abfragen oder Formularen zeigt Access stets die aktuellsten Daten an. Dies gilt auch in einer Mehrbenutzerumgebung. Es kann allerdings vorkommen, daß sich die Datenbasis eines Formulars oder Datenblatts ändert, *während es geöffnet ist*. Dafür kann es mehrere Gründe geben, vor allem die folgenden:

- Sie arbeiten in einer Mehrbenutzerumgebung, und ein anderer Benutzer hat Änderungen an einer Tabelle vorgenommen, die dem von Ihnen geöffneten Formular zugrunde liegt.
- Sie nehmen selbst in Ihrer Datenbank Änderungen an einer Tabelle vor, während ein darauf basierendes Formular geöffnet bleibt.
- Ein im Hintergrund laufendes Makro oder eine VBA-Prozedur veranlaßt Änderungen in der Datenbasis eines Formulars oder einer Abfrage.

Ob eine Datenänderung in einer derartigen Situation automatisch zur aktualisierten Anzeige in Formularen und Datenblattansichten führt, hängt von der Art der Datenänderung und vom anzeigenden Steuerelement ab. Es gelten die folgenden Regeln und Beschränkungen:

Datenänderungen an bestehenden Datensätzen

Änderungen an *bestehenden* Datensätzen werden in Formularen und Datenblattansichten automatisch aktualisiert (Ausnahme: Listen- und Kombinationsfelder, vgl. weiter unten in diesem Abschnitt). Dies erfolgt in gewissen Zeitintervallen, die vom Benutzer im Dialogfeld des Befehls *Optionen* aus dem Menü *Extras*, Register *Weitere*, eingestellt werden können, vgl. Bild 6.35. Entscheidend ist in diesem Zusammenhang das in Sekunden anzugebende *Intervall für Anzeigeaktualisierung*. Von Access wird es standardmäßig auf 60 Sekunden eingestellt. Sie können es auf einen Wert zwischen 1 und 32.766 einstellen. Zumindest in einer Einzelbenutzerumgebung erscheint ein deutlich kleinerer Wert als 60 Sekunden angemessen.

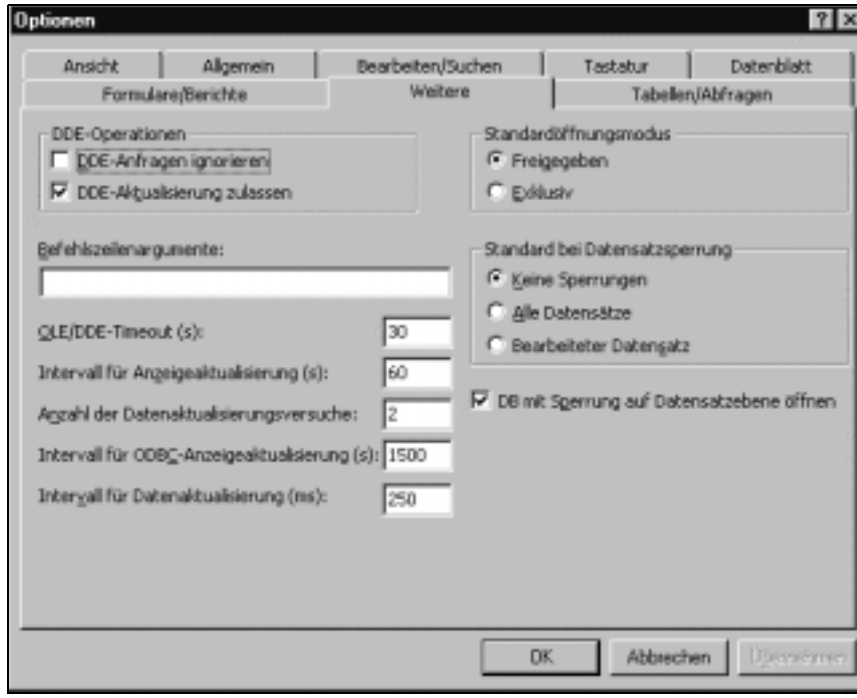


Bild 6.35: Einstellungsmöglichkeiten für Mehrbenutzerumgebung im Dialogfeld des Befehls »Optionen« aus dem Menü »Extras«. Wiedergegeben sind die von Access standardmäßig eingestellten Werte.

Sie können eine Aktualisierung der Anzeige bestehender Datensätze aber auch jederzeit manuell erzwingen. Dies mag sinnvoll sein, wenn die automatische Aktualisierung in einzelnen Fällen zu lange dauert, oder wenn Sie ganz sicher sein wollen, die aktuellsten Daten angezeigt zu bekommen. Wählen Sie dazu den Befehl *Anzeige aktualisieren* aus dem Menü *Datensätze* oder die Taste F9.

Mit diesem Befehl veranlassen Sie, daß im aktiven Formular oder in der aktiven Datenblattansicht alle zwischenzeitlich erfolgten Änderungen an bestehenden Datensätzen aktualisiert angezeigt werden. Der Befehl bezieht sich allerdings nicht auf neu hinzugefügte oder gelöschte Datensätze, vgl. dazu den folgenden Punkt.

Aktualisierung der Anzeige für neu hinzugefügte oder gelöschte Datensätze

Die Anzeigeaktualisierung bezüglich *neu hinzugefügter oder gelöschter* Datensätze erfolgt nicht automatisch. Dies muß vielmehr stets manuell (bzw. durch Makros und VBA-Prozeduren) veranlaßt werden. Drücken Sie zu diesem Zweck die Tastenkombination

⇧+F9

Damit erreichen Sie dieselbe Wirkung, wie wenn Sie das Formular oder die Datenblattansicht schließen und anschließend wieder öffnen: Access ermittelt alle Daten und Datensätze vollständig neu. Diese Aktion hat allerdings auch ihren Preis: Access setzt danach den Fokus auf den ersten Datensatz im Formular oder Datenblatt, so daß Sie ggf. wieder zum bis dahin aktuellen Datensatz blättern müssen.

Für Benutzer von Makros und VBA-Prozeduren: Der Tastenkombination ⇧+F9 entspricht das Makro *AktualisierenDaten* (VBA: Methode *Requery*).

Aktualisierung der Datenbasis von Listen- und Kombinationsfeldern

Listen- und Kombinationsfelder haben oft Tabellen oder Abfragen als Datenbasis. Deren Änderungen werden nicht automatisch im Listen- oder Kombinationsfeld wiedergegeben. Dies müssen Sie manuell mit der Taste F9 oder dem Befehl *Anzeige aktualisieren* aus dem Menü *Datensätze* bzw. durch ein entsprechendes Makro oder eine VBA-Prozedur veranlassen.

Aktualisierung der Datenbasis in einem Nachschlagelistenfeld oder -kombinationsfeld

Die Daten im Listen- oder Kombinationsfeld eines Nachschlagefeldes aus einer anderen Tabelle werden seit Access 97 nicht mehr automatisch aktualisiert, wenn die andere Tabelle verändert wird. Sie können die Daten im Nachschlagefeld durch Drücken der Taste F9 aktualisieren.

Kapitel 7

Eine Datenbank konzipieren

Vorbemerkung. Dieses Kapitel bietet eine geraffte Darstellung, wie Sie zweckmäßigerweise eine Datenbank konzipieren; Ziel der Darstellung ist es, einen Überblick über wichtige Probleme der Datenmodellierung zu geben. Dieser Überblick sollte ausreichen, um einfache Datenbanken so zu erstellen, daß sie fehlerfrei funktionieren. Wenn Sie sich jedoch genauer über Probleme der Datenmodellierung und deren Lösungen informieren wollen, sollten Sie Kap. 32, *Datenmodellierung für Fortgeschrittene*, lesen, wo die entsprechenden Zusammenhänge sehr viel ausführlicher behandelt werden.

7.1 Welche Aufgaben?

Zunächst sollten Sie sich und ggf. den anderen an den Ergebnissen der Datenbank Beteiligten (z.B. Mitarbeiter, die die Datenbank nutzen werden, oder der Verantwortliche, der den Datenbankentwurf in Auftrag gibt) darüber Klarheit verschaffen, welche Leistungen die Datenbank erbringen soll. Daraus wird am besten deutlich, zu welchen Themen Sie Daten zusammenstellen und halten müssen, und wie Sie die Daten auf die einzelnen Tabellen verteilen sollten.

- Stellen Sie eine Leistungsbeschreibung auf, die möglichst detailliert auflistet, welche einzelnen Aufgaben das Datenbanksystem lösen soll.
- Im allgemeinen entsteht eine neue Datenbank nicht im luftleeren Raum, sondern sie soll, zumindest teilweise, Aufgaben erledigen, die bisher in konventioneller Weise gelöst wurden. Bei herkömmlicher Datenbearbeitung werden häufig verschiedene Arbeitsmittel wie Karteikarten, Formulare (z.B. Auftragsformular, Lieferschein, Rechnungsformular, Mitteilungsformulare verschiedenster Art), Berichte, statistische Zusammenstellungen, Berechnungsschemata für komplizierte Abrechnungen, Grafiken, Serienbriefe, Adreßetiketten etc. verwendet. Tragen Sie diese Hilfsmittel zusammen, und überlegen Sie, welche Anforderungen an die Datenhaltung daraus resultieren

- Obwohl eine moderne Datenbank praktisch alle Vorfälle eines Betriebes oder jeder anderen Organisation berücksichtigen kann, ist es im allgemeinen nicht sinnvoll, wirklich jeden Vorgang aufzunehmen. Immerhin kostet es einige Mühe, ein Datenbanksystem so weit zu entwickeln, daß es praktisch brauchbar ist. Daher sollten Sie Aufgaben, die nur sehr selten anfallen, nicht unbedingt in die Datenbank einbeziehen. Fragen Sie verschiedene Mitarbeiter, welche einzelnen Aufgaben sie in der täglichen Arbeit am häufigsten beschäftigen (z.B.: Auskunft geben über eine Rechnungsposition) und welche seltener (z.B.: eine Rechnung mit US-amerikanischer VAT, der dortigen Mehrwertsteuer, erstellen). Als Erhebungsinstrumente kommen Interviews oder Fragebögen in Frage. Auf diese Weise bekommen Sie Anhaltspunkte dafür, welche Daten auf alle Fälle gründlich und detailliert berücksichtigt werden sollten, und für welche Daten eine Aufnahme in die Datenbank wahrscheinlich nicht lohnt.
- Erstellen Sie eine Liste mit allen Einzeldaten, die festgehalten werden sollen, und weisen Sie dabei den jeweiligen Zweck sowie den Namen des Mitarbeiters (oder der Abteilung o.ä.) aus, der die Daten verwendet bzw. für die Aufnahme in die Datenbank vorschlägt.
- Notieren Sie auch die übergreifenden Fragen, die die Datenbank beantworten soll. Zu solchen Fragen könnten z.B. die folgenden gehören: Welches waren im vergangenen Jahr die besten Kunden, gemessen an einem bestimmten Auftragsvolumen? Wie weit reicht der gegenwärtige Auftragsbestand? Welche Artikel wurden oft, welche selten nachgefragt? Wie sieht die regionale Verteilung von Kunden und Aufträgen aus? Hat sie sich in den letzten Jahren verändert?

7.2 Welche Tabellen?

Die Art und Weise, wie Sie die Daten auf verschiedene Tabellen verteilen, ist am kritischsten für die gesamte Datenbank. Gleichzeitig läßt sich diese Frage am schwierigsten in genereller Weise beantworten. Im allgemeinen empfiehlt es sich, thematisch gleichartige Daten in einer Tabelle zusammenzufassen. Solche Themen können beispielsweise Kundenadressen, Bestellungen, Angebote oder auch konstante Werte für das DBMS (z.B. Mehrwertsteuersatz, Rabattstaffeln, Sätze einer Gebührenordnung) sein. Oft stellt es sich allerdings als problematisch dar, herauszufinden, welche Daten zum selben Thema gehören. Nehmen Sie als Beispiel die Themen Kundenadressen und Bestellungen. Was spricht dafür, Adressen und Bestellungen in verschiedenen Tabellen zu halten, was dagegen?

Für getrennte Tabellen sprechen die folgenden Punkte:

- **Widersprüche bei redundanten Informationen**. Eine Information ist redundant, wenn sie überflüssigerweise an mehreren Stellen der Datenbank vorkommt. Wenn Sie Kundenadressen und Bestellungen in derselben

Tabelle halten, muß dieselbe Kundenadresse immer wieder mit jeder weiteren Bestellung desselben Kunden eingegeben werden. Im Falle einer Adressenänderung kann dies zu Widersprüchen führen, weil Sie vielleicht nicht alle Datensätze, die geändert werden müßten, identifizieren. Dies kann z.B. leicht passieren, weil der Kundenname infolge von Schreibfehlern verschieden geschrieben ist. Ähnliche Widersprüche können sich beim Einfügen oder Löschen eines neuen Datensatzes ergeben.

- **Inflexibilität bei sachlich zu breiten Tabellen**. Obwohl es im Einzelfall sinnvoll sein kann, eine Kundenadresse zu speichern, ohne daß eine Bestellung vorliegt, könnten Sie die Adresse stets nur zusammen mit den Feldern für Bestellinformationen halten. Abgesehen davon, daß dabei unnötiger Speicherplatz belegt werden würde, ergäben sich vor allem Probleme, wenn Sie nicht mehr benötigte Bestellungen löschen wollen: Falls Sie keine besonderen Prüfroutinen einbauen, könnten Kundenadressen dabei ganz verloren gehen.
- **Nachträgliche Strukturänderung der Datenbank**. Es gehört zum geschäftlichen Alltag, daß bisher sinnvolle Fragestellungen sich als überflüssig erweisen, gleichzeitig aber neue hinzukommen. Beispielsweise könnte es sich als notwendig erweisen, eine Zuordnung von Mitarbeitern zu den Bestellungen einzurichten, so daß jederzeit ersichtlich ist, welche Bestellung(en) von welchem Mitarbeiter betreut werden. Eine solche Zuordnung läßt sich nachträglich viel leichter einrichten, wenn die Bestelldaten unabhängig von den Kundenadressen gehalten werden.

Gegen getrennte Tabellen spricht der folgende Punkt:

- Die vorangehenden Überlegungen legen nahe, im allgemeinen mehrere einzelne Tabellen, die durch eine Beziehung miteinander verknüpft werden, vorzusehen, statt nur eine oder wenige große Tabellen zu konzipieren. Das wesentliche Argument ist dabei die Vermeidung von redundanten Informationen. Dabei kann sich allerdings als problematisch erweisen, wann eine Information als redundant anzusehen ist. Im obigen Beispiel scheint klar zu sein, daß Kundenadressen getrennt von den Bestellungen gehalten werden sollen und die Zuordnung von Kundenadresse zu Bestellung über eine 1:n-Beziehung hergestellt wird. Eine solche Konstruktion hat zur Folge, daß die Kundenadresse auch alter Bestellungen nachträglich geändert wird, wenn die Kundenadresse selbst sich ändert. Dieser Effekt ist im allgemeinen erwünscht, kann aber auch unerwünscht sein. Dies trifft z.B. zu, wenn eine Bestellung mit all ihren Bestellinformationen, einschließlich Kundenadresse, dokumentarischen Charakter haben soll: Wenn es notwendig ist festzuhalten, wie die Kundenadresse zum Zeitpunkt der Bestellung lautete, müssen Sie diese zum Bestandteil des Bestelldatensatzes und damit der Tabelle mit den Bestellungen machen oder eine weitere Tabelle einrichten, in der die Historie der Kundenadressen festgehalten wird.

7.2.1 Verknüpfte Tabellen verwenden?

Access bietet die Möglichkeit, in eine Datenbank Tabellen aus anderen Datenbankanwendungen, auch solche anderer Formate (z.B. dBase, Paradox, Btrieve), einzubinden, indem eine Verknüpfung zu diesen Tabellen hergestellt wird. Eine verknüpfte Tabelle bleibt Bestandteil der anderen Anwendung, wird aber von der Datenbank, in die sie durch Verknüpfung eingebunden ist, lesend und schreibend wie eine eigene Tabelle verwaltet. In folgenden Fällen empfiehlt es sich, mit verknüpften Tabellen zu arbeiten:

- Dieselben Daten werden auch von anderen Anwendungen in derselben Organisation benötigt. Die Verwendung mehrerer Tabellen für dieselben Daten würde Redundanz bedeuten und im allgemeinen zu Widersprüchen führen.
- Wenn Sie eine Access-Datenbankanwendung öfter für andere Benutzer überarbeiten müssen (z.B. Formulare ändern, Berichte ergänzen, Makros einfügen etc.), ist es im allgemeinen das Leichteste, den Benutzern eine Kopie der gesamten geänderten Datenbank zu geben. Falls die Datenbank allerdings Daten in Tabellen enthält, an denen die Benutzer in der Zwischenzeit selbst Bearbeitungen vorgenommen haben, gingen diese Datenänderungen durch das Kopieren verloren. Mit verknüpften Tabellen passiert dies nicht.

7.3 Welche Felder in den Tabellen?

Generell läßt sich sagen, daß jeder Einzelinformation, die zum selben Tabellenthema gehört, ein eigenes Feld in der Tabelle entspricht. Glücklicherweise wirkt es sich im allgemeinen für die Datenbank insgesamt nicht sehr kritisch aus, wenn Sie einzelne Felder nachträglich einfügen, die Sie zunächst vielleicht übersehen oder für unwichtig gehalten haben. Beispielsweise ist es mit wenig Aufwand und geringen Konsequenzen für die gesamte Datenbank verbunden, wenn Sie der Tabelle mit Kundenadressen nachträglich ein Feld für die Faxnummer hinzufügen. Gleichwohl sollten Sie bemüht sein, daß die Felder für jede einzelne Tabelle von Anfang an nach Möglichkeit den folgenden Punkten genügen:

- **Sachliche Homogenität**. Nehmen Sie nur solche Felder in dieselbe Tabelle auf, welche sich direkt auf das Tabellenthema beziehen. Dazu gehören im Falle einer Tabelle mit Kundenadressen sicherlich Name, Straße etc. Wie verhält es sich aber mit einer Information darüber, ob es sich um einen wichtigen oder weniger wichtigen Kunden handelt? Unterstellt, die Frage der Wichtigkeit werde am jährlichen Auftragsvolumen der Kunden gemessen, wäre es angemessen, dafür kein Feld in der Tabelle mit den Kundenadressen vorzusehen, sondern sie vielmehr aus der Tabelle mit den Bestellungen über eine Abfrage *ermitteln zu lassen*. Wenn die Frage andererseits

nur auf einer sehr subjektiven Basis durch einen langjährigen Mitarbeiter beantwortet werden kann, erschiene es angemessen, sie durch ein eigenes Feld in der Tabelle mit den Kundenadressen zu repräsentieren.

- **Vollständigkeit**. Bemühen Sie sich, alle benötigten Informationen, die zum Tabellenthema gehören, zu identifizieren und durch Tabellenfelder zu repräsentieren. Prüfen Sie die Frage der Vollständigkeit am besten, indem Sie verschiedene Ergebnisse der Datenbankanwendung simulieren. Beispielsweise könnten Sie ein bestimmtes Rechnungsformular entwerfen und sich fragen, wo die in der Rechnung auftauchenden Informationen in der Datenbank zu finden sind.
- **Abgeleitete und berechnete Informationen**. Für Informationen, die sich eindeutig aus Informationen herleiten lassen, die an anderer Stelle der Datenbank gehalten werden, sollten Sie keine eigenen Felder vorsehen. Dies trägt zur Widerspruchsfreiheit und Transparenz bei und verringert auch den benötigten Speicherplatz. Beispielsweise ist es überflüssig, den Bruttoumsatz in einem eigenen Feld zu berücksichtigen, wenn für den Nettoumsatz und den Mehrwertsteuerbetrag je ein eigenes Feld existiert (vorausgesetzt, daß der Bruttoumsatz sich ausschließlich aus diesen beiden Größen zusammensetzt). Letztlich läuft auch dies, wie oben bereits beim Problem der angemessenen Tabellenzusammenstellung besprochen, auf die Vermeidung von Redundanz hinaus. Daher gilt auch hier derselbe einschränkende Hinweis: Wenn die betreffende Information dokumentarischen Charakter hat und sich der abgeleitete oder berechnete Wert im Laufe der Zeit ändern kann, muß sie durch ein eigenes Feld repräsentiert werden. Ein praktisch bedeutsames Beispiel ist der Mehrwertsteuerbetrag in Angeboten, Lieferungen, Rechnungen u.ä.: Wenn Sie dafür kein Feld vorsehen, weil sich der Betrag ja zum gegebenen Zeitpunkt aus dem jeweils herrschenden Mehrwertsteuersatz und dem Nettorechnungsbetrag ergibt und so für die Rechnung jeweils durch das Programm ermittelt werden kann, könnten Sie nach einer generellen Veränderung des Mehrwertsteuersatzes für alte Rechnungen in der Datenbank den Mehrwertsteuerbetrag (und damit den Bruttobetrag) nicht mehr ermitteln.

7.4 Primärschlüsselfelder

Im allgemeinen sollte eine Tabelle ein Feld aufweisen, das als Primärschlüssel dient. In einem Primärschlüsselfeld ist jeder Wert einzigartig, d.h. zwei Datensätze haben niemals denselben Wert. Dafür sorgt Access, sobald ein Feld als Primärschlüsselfeld definiert ist. Sie können auch einen zusammengesetzten Primärschlüssel definieren, indem Sie mehrere Felder zum Primärschlüssel machen, vgl. weiter unten in diesem Punkt. Dies ist sinnvoll und notwendig, wenn ein Feld, wie z.B. das Feld *Nachname* in einer Mitarbeitertabelle, keine eindeutigen Werte zuläßt (es ist nicht auszuschließen, daß beispielsweise der Nachname *Schmidt* mehrfach vorkommt). Tabellen mit einem Primärschlüssel weisen die folgenden Vorteile auf:

- Eine Tabelle mit einem Primärschlüssel kann als Mastertabelle in einer Beziehung fungieren. Umgekehrt: Das Festlegen einer Beziehung setzt einen Primärschlüssel für die Mastertabelle voraus, im Falle einer 1:1-Beziehung auch für die Detailtabelle.
- Access kann auf Tabellen mit einem Primärschlüssel effizienter zugreifen, vor allem dann, wenn gleichzeitig Daten aus mehreren Tabellen abgefragt werden.
- Im Falle einer Abfrage kann es ohne Primärschlüssel zu Mehrdeutigkeiten kommen, wenn gleichzeitig Daten aus mehreren Tabellen abgefragt werden. In diesen Fällen ist es nicht möglich, Änderungen im Dynaset der Abfrage vorzunehmen, die bei definierten Primärschlüsseln möglich wären.

Beachten Sie bei der Frage, welches Feld als Primärschlüssel dienen soll, bitte die folgenden Gesichtspunkte:

- Wegen der Eigenschaft eines Primärschlüsselfeldes, nur einzigartige Werte zuzulassen, sollten Sie sicher sein, daß dies mit den übrigen Forderungen an das betreffende Feld vereinbar ist. Beispielsweise eignet sich ein Nachname in einer Adressentabelle im allgemeinen nicht als Primärschlüsselfeld. Geeignet sind dagegen Felder wie Auftragsnummer, individueller Kundencode etc.
- Die Feldgröße des Primärschlüsselfeldes beeinflußt die Geschwindigkeit verschiedener Datenbankoperationen: Je kleiner die Feldgröße, desto höher die Geschwindigkeit. Begrenzen Sie daher vor allem Felder vom Datentyp *Text*, die Sie als Primärschlüsselfeld verwenden wollen, auf eine möglichst kleine Feldgröße. Oftmals bietet sich für den Primärschlüssel ein Feld vom Datentyp *Zahl* an.
- Obwohl eine Tabelle nur einen Primärschlüssel enthalten kann, können doch mehrere Felder zusammen den Primärschlüssel bilden. Ein aus mehreren Feldern bestehender Primärschlüssel ist sinnvoll, wenn ein Feld allein keine eindeutigen Werte ermöglicht. Ein Beispiel für eine Tabelle mit einem aus zwei Feldern bestehenden Primärschlüssel bietet die Tabelle *Bestelldetails* in der Beispieldatenbank Nordwind.mdb. Die Tabelle *Bestelldetails* ist die dritte Tabelle für eine m:n-Beziehung zwischen den Tabellen *Artikel* einerseits und *Bestellungen* andererseits. In der Tabelle *Bestelldetails* ist der Primärschlüssel für die beiden Felder *Bestell-Nr* und *Artikel-Nr* gesetzt, so daß jede Wertekombination für diese Felder nur einmal vorkommen darf. Damit wird verhindert, daß dieselbe Bestellung mit zwei oder mehr identischen Artikeln aufgeführt wird. Auch umgekehrt gilt: Derselbe Artikel wird niemals öfter als einmal derselben Bestellung zugeordnet.

7.5 Welche Beziehungen zwischen den Tabellen?

Wenn Sie eine Beziehung zwischen zwei Tabellen herstellen, sorgen Sie dafür, daß Sie über einen Datensatz in der einen Tabelle einen oder mehrere Datensätze in der anderen Tabelle identifizieren können. Die Verknüpfung wird durch je ein Schlüsselfeld in den beiden Tabellen hergestellt. In der *Mastertabelle* (der linken Seite der Beziehung) ist dies der *Primärschlüssel*, in der *Detailtabelle* (der rechten Seite) ein Feld, das denselben oder dieselben Wert(e) aufweist, wie der Primärschlüssel in der Mastertabelle. Das Schlüsselfeld in der Detailtabelle wird *Fremdschlüssel* genannt. Beispielsweise können Sie eine Beziehung zwischen einer Tabelle *Kundenadressen* als Master- und einer Tabelle *Bestellungen* als Detailtabelle definieren. Ein Beispiel finden Sie u.a. in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, Punkt 5.7, *Beziehung zwischen den Tabellen definieren*.

7.5.1 1:n, 1:1 oder m:n

Formal sind zwischen zwei Tabellen drei Arten von Beziehungen möglich:

1:n-Beziehung
1:1-Beziehung
m:n-Beziehung

- **1:n-Beziehung**. Dies ist der für Beziehungen zwischen Tabellen allgemeine Fall, denn in der Praxis sind fast alle Beziehungen eine 1:n-Beziehung. Bei einer derartigen Beziehung entsprechen einem Datensatz in der Mastertabelle mehrere Datensätze in der Detailtabelle, wobei *mehrere* auch die Zahl 0 einschließt. *Ein Kunde – mehrere Bestellungen* stellt beispielsweise eine 1:n-Beziehung dar. Auch die Verknüpfung der Tabelle *Personal* mit der Tabelle *Projekte* im Einführungsbeispiel in Kap. 5 wurde als 1:n-Beziehung definiert.
- **1:1-Beziehung**. In diesem Fall entspricht einem Datensatz der Mastertabelle nicht mehr als ein Datensatz in der Detailtabelle, wobei wiederum auch möglich ist, daß in der Detailtabelle kein entsprechender Datensatz vorliegt. Sie mögen sich fragen, ob man nicht in den Fällen einer 1:1-Beziehung gleich alle Felder der Master- und Detailtabelle in eine Tabelle aufnehmen könnte. Vielfach mag das zutreffen, und in diesen Fällen erscheint eine derartige Verknüpfung in der Tat überflüssig. Manchmal aber erweist sich eine 1:1-Beziehung zwischen zwei Tabellen als sinnvoll und notwendig. Nehmen Sie das folgende Beispiel: Eine Datenbank enthält eine Tabelle für Aufträge. Jeder Auftrag soll eindeutig sein, daher dient die Auftragsnummer als Primärschlüssel. Für jeden Auftrag müssen auf der einen Seite dieselben Informationen verwaltet werden, beispielsweise der Auftraggeber und das Auftragsdatum. Die Aufträge unterscheiden sich jedoch in sachlicher Hinsicht, denn es gibt drei verschiedene Auftragsarten, für die quantitativ und qualitativ verschiedene Informationen verarbeitet und die auch auf unterschiedliche Weise abgerechnet werden.

So wird die eine Auftragsart nach Arbeitsstunden (Ingenieurstunden plus Hilfskraftstunden plus Verwaltungsstunden etc.) abgerechnet, die zweite nach der Anzahl erledigter Globalpositionen plus Hilfskraftstunden und die dritte nach dem Wert des verarbeiteten Materials plus einer Mischung aus 12 verschiedenen Einzelpositionen, die jeweils dem Wert einer Gebührentabelle entsprechen. In diesem Fall empfiehlt es sich, die für die Auftragsarten spezifischen Informationen in je einer eigenen Tabelle zu sammeln, also in drei eigenen Untertabellen. Die Auftragstabelle mit den für jeden Auftrag gleichen Informationen (Auftraggeber und Auftragsdatum) wird jeweils als Mastertabelle in einer 1:1-Beziehung zu jeder der drei Untertabellen definiert, die jeweils als Detailtabelle dienen. Auf diese Weise kann jeder Auftrag mit seinen verschiedenen Detailinformationen eindeutig identifiziert werden. Gleichzeitig können Sie die Detailtabellen so unterschiedlich gestalten, wie dies für die jeweilige Auftragsart erforderlich ist.

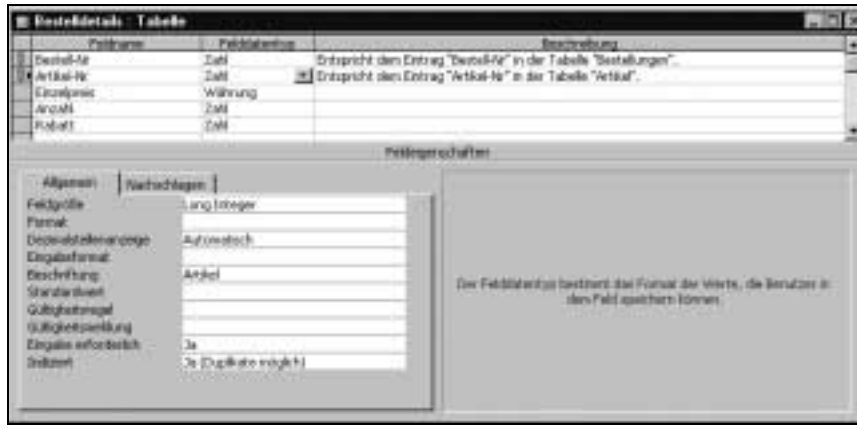


Bild 7.1: Tabelle »Bestelldetails« als dritte Tabelle, um die m: n-Beziehung zwischen den Tabellen Artikel einerseits und »Bestellungen« andererseits zu definieren (Entwurfsansicht)

- **m:n-Beziehung**. In einer m:n-Beziehung können einem Datensatz der Tabelle 1 mehrere Datensätze der Tabelle 2 entsprechen, aber auch umgekehrt können einem Datensatz der Tabelle 2 mehrere Datensätze der Tabelle 1 entsprechen. Ein solcher Fall liegt beispielsweise vor, wenn mehrere Bestellungen den gleichen Artikel enthalten, gleichzeitig aber mehrere Artikel in einer Bestellung enthalten sind. Eine m:n-Beziehung läßt sich nicht unmittelbar, sondern nur über den Umweg einer dritten Tabelle definieren. Dieses Beispiel können Sie in der Nordwind-Datenbank nachvollziehen. Dort wird die m:n-Beziehung zwischen der Tabelle *Bestellungen* einerseits und der Tabelle *Artikel* andererseits über die dritte Tabelle *Bestelldetails* hergestellt: Die Tabelle *Bestellungen* besitzt das Primärschlüsselfeld *Bestell-Nr*, die Tabelle *Artikel* das Primärschlüsselfeld *Arti-*

kel-Nr. Die dritte Tabelle *Bestelldetails* besitzt diese beiden Felder jeweils als Fremdschlüssel. Die m:n-Beziehung zwischen den Tabellen *Bestellungen* einerseits und *Artikel* andererseits ergibt sich durch zwei 1:n-Beziehungen: Eine 1:n-Beziehung ist zwischen *Bestellungen* und *Bestelldetails* definiert, die andere zwischen *Artikel* und *Bestelldetails*, vgl. Bild 7.1 sowie Bild 7.2. Im vorliegenden Beispiel sollte darüber hinaus für Eindeutigkeit bei den Artikel-Bestellung-Kombinationen gesorgt werden, so daß eine bestimmte Kombination stets nur einmal zulässig ist. Daher wurde in der Tabelle *Bestelldetails* für die beiden Felder *Artikel-Nr* und *Bestell-Nr* ein zusammengesetzter Primärschlüssel festgelegt.

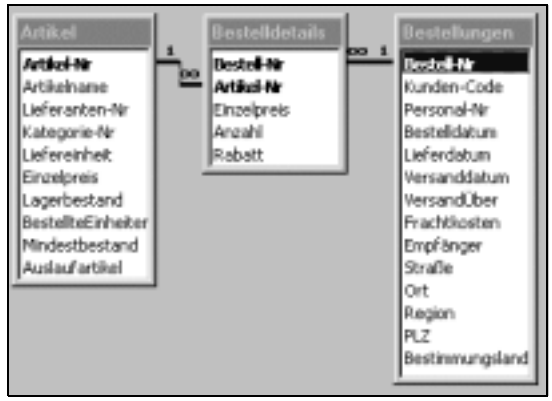


Bild 7.2: So stellt sich die m: n-Beziehung der Tabellen »Artikel« und »Lieferanten« im Fenster »Beziehungen« dar.

Kapitel 8

Tabellen: Grundlagen

8.1 Was ist eine Tabelle?

Eine Tabelle ist ein Access-Objekt, in dem Daten gespeichert werden. Tabellen sind in Form von *Zeilen* und *Spalten* organisiert. Jede Zeile enthält einen *Datensatz*, jede Spalte stellt ein *Feld* dar. Inhaltlich sollten alle Felder Eigenschaften desselben Sachverhalts oder Themas wiedergeben. In einer Tabelle beispielsweise, in der Informationen über Mitarbeiter gespeichert sind, enthält jeder Datensatz (Zeile) die Informationen zu einem bestimmten Mitarbeiter. Um Widersprüche in der Datenbankverwaltung zu vermeiden, sollten die Felder (Spalten) dabei ausschließlich Eigenschaften des Mitarbeiters (wie z.B. Name, Vorname, Wohnort etc.) wiedergeben, nicht jedoch auch gleichzeitig Eigenschaften anderer Themen wie etwa seines Arbeitsplatzes oder der von ihm bearbeiteten Projekte.

Die Gesamtheit der Informationen einer Datenbank wird im allgemeinen auf mehrere Tabellen verteilt. Welche Gesichtspunkte und Regeln dabei zu berücksichtigen sind, ist nicht Gegenstand dieses Kapitels, sondern wird im Überblick in Kap. 7, *Eine Datenbank konzipieren*, und detailliert in Kap. 32, *Datenmodellierung für Fortgeschrittene*, behandelt. In diesem Kapitel setze ich voraus, daß die inhaltliche Frage, welche Felder die Tabelle enthalten soll und welche Beziehung zu anderen Tabellen bestehen sollen, bereits beantwortet ist. Hier geht es zur Hauptsache um die technischen Aspekte der Angelegenheit.

Eine Tabelle kann nur Felder mit Wertkonstanten (z.B. Zahlen, Texte, Bilder) enthalten, nicht dagegen Felder mit Formelausdrücken. In einem derartigen Feld würde die Information auf Basis eines Ausdrucks (einer Formel) aus Werten anderer Felder ermittelt. Wenn Sie bisher wenig Erfahrung mit Datenbankprogrammen haben, sich aber mit einem Tabellenkalkulationsprogramm (beispielsweise *Excel* oder *Lotus*) auskennen, werden Sie vielleicht Felder mit Ausdrücken in Access-Tabellen vermissen, weil es in einem

Arbeitsblatt einer Tabellenkalkulation üblich ist, die Werte für neue Spalten aus anderen Spalten mittels Formeln zu ermitteln. Diese Aufgabe können Sie auch in Access lösen, jedoch nicht mit einer Tabelle, sondern mit einer Abfrage: Abfragen können Felder mit Ausdrücken enthalten, Tabellen nicht. Die Begründung für diese Organisation der Datenverwaltung liegt im wesentlichen darin, daß die Informationen in Datenbanken möglichst redundanzfrei (d.h. ohne überflüssige Informationen) gehalten werden sollen. Redundanzfreiheit spart nicht nur Speicherplatz, sondern hilft auch Widersprüche in der Datenstruktur zu vermeiden. Um die Informationen eines mit einem Ausdruck berechneten Feldes zu erhalten, bedarf es außer der Felder, aus denen sie ermittelt werden, lediglich des Ausdrucks (der Formel), welche die Berechnung leistet. Es wäre also völlig überflüssig (redundant), wenn der Berechnungsausdruck für jede Zeile des berechneten Feldes eigens gespeichert werden würde, wie dies in einer Tabelle ja sein muß. Statt dessen braucht der berechnende Ausdruck nur einmal gespeichert zu werden, damit die Informationen nicht verloren gehen. Genau dieses geschieht in einer Abfrage, die Sie speichern. Daß berechnete Felder in Tabellen nicht möglich sind, hat also seinen guten Grund im rationalen und effizienten DBMS, das eine Aufgabenteilung zwischen Tabellen, die (Ur-) Informationen speichern, und Abfragen, welche diese Informationen in beliebiger Weise zusammenstellen und modifizieren, vorsieht.

8.2 Tabellen entwerfen

Um eine neue Tabelle zu erstellen, gehen Sie grundsätzlich folgendermaßen vor:

- Aktivieren sie im Datenbankfenster (ggf. aktivierbar mit F11) die Objektliste *Tabellen*, und klicken Sie dann auf die Schaltfläche *Neu*.

  Oder:

- Schlagen Sie die Symbol-Schaltfläche *Neues Objekt* (vgl. links nebenstehend) auf, und wählen Sie darin das Symbol *Tabelle*.

In jedem der beiden Fälle wird das Dialogfeld *Neue Tabelle* eingeblendet, vgl. Bild 8.1. Hier können Sie aus fünf verschiedenen Wegen wählen, eine neue Tabelle zu erstellen:

- Datenblattansicht
- Entwurfsansicht
- Tabellen-Assistent
- Tabelle importieren
- Tabelle verknüpfen

Die letzten beiden Verfahren werden nicht in diesem Kapitel, sondern in Kap. 9, *Tabellen importieren, exportieren und verknüpfen*, behandelt.

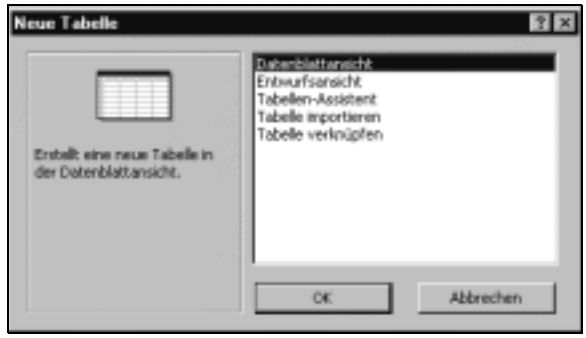


Bild 8.1: Dialogfeld »Neue Tabelle«

Der Tabellen-Assistent bietet Ihnen eine Reihe von Beispieltabellen (25 geschäftlich und 20 privat orientierte, also insgesamt 45) an, aus denen Sie die für Sie geeigneten Felder auswählen können. Der Umgang mit dem Tabellen-Assistenten ist praktisch selbsterklärend, so daß ich darauf nicht im einzelnen eingehe. Ich werde lediglich zeigen, wie Sie prinzipiell damit arbeiten können. Auch wenn Sie neue Tabellen stets oder fast immer mit Hilfe des Tabellen-Assistenten anlegen, werden Sie im allgemeinen zumindest einige Feldnamen, Eigenschaften, Eingabeformate oder Gültigkeitsregeln ändern müssen, damit die neue Tabelle Ihren Zwecken entspricht. Aus diesem Grunde konzentriert sich das vorliegende Kapitel auf die Erklärung derartiger Aufgaben, ohne deren Bewältigung Sie nicht auskommen werden.

8.2.1 Neue Tabelle mit dem Tabellen-Assistenten erstellen

Um eine neue Tabelle mit dem Tabellen-Assistenten zu erstellen, gehen Sie folgendermaßen vor:

- Öffnen Sie das Dialogfeld *Neue Tabelle*, indem Sie im Register *Tabellen* des Datenbankfensters (aktivierbar mit F11) auf die Schaltfläche *Neu* oder in der Symbolleiste auf die Symbol-Schaltfläche *Neues Objekt* (vgl. links nebenstehend) klicken.
- Wählen Sie im Dialogfeld *Neue Tabelle* (vgl. den vorangehenden Punkt und darin Bild 8.1) den Listeneintrag *Tabellen-Assistent*, und bestätigen Sie mit OK. Dann zeigt sich das Dialogfeld *Tabellen-Assistent*, vgl. Bild 8.2.
- Wählen Sie im Dialogfeld *Tabellen-Assistent* eine der Kategorien *Geschäftlich* oder *Privat*, und markieren Sie dann die gewünschte Beispieltabelle. Zu jeder jeweils markierten Beispieltabelle werden im Listenfeld *Beispielfelder* die Namen der jeweils zugehörigen Tabellenfelder angezeigt. In Bild 8.2 beispielsweise werden die Felder *Abteilungsname*, *PersonalNr* usw. angezeigt, die zur dort gerade markierten Beispieltabelle *Personal* gehören. Übernehmen Sie aus der Liste *Beispielfelder* die benö-

tigten Felder in die Liste *Felder der neuen Tabelle*, indem Sie auf eine der beiden Schaltflächen > für Übernahme des gerade markierten Feldes oder >> für Übernahme aller Felder klicken oder auf einzelne zu übernehmende Felder doppelklicken.

Bild 8.2: Dialogfeld »Tabellen-Assistent«

- Wenn die Liste *Felder der neuen Tabelle* alle von Ihnen als erforderlich betrachteten Felder enthält, kommen Sie mit der Schaltfläche *Weiter* im Prozeß der Tabellenerstellung voran. Sie werden in weiteren Dialogfeldern nach dem Namen, unter dem die Tabelle gespeichert werden soll, gefragt. Ferner müssen Sie sich bezüglich eines automatisch festgelegten oder selbstdefinierten Primärschlüssels entscheiden. Schließlich haben Sie die Möglichkeit, eine Beziehung zu einer anderen Tabelle – sofern in der Datenbank bereits vorhanden – festzulegen. Diese Schritte erklären sich hinsichtlich des formalen Vorgehens weitgehend selbst, so daß ich auf eine Darstellung und Erklärung der weiteren Dialogfelder des Tabellen-Assistenten verzichte. Inhaltlich müssen Sie natürlich über die Bedeutung von Primärschlüssel und Beziehungen zwischen Tabellen Bescheid wissen, wenn Sie die entsprechenden Fragen des Tabellen-Assistenten sinnvoll beantworten wollen. Dazu können Sie sich in diesem Buch an anderer Stelle informieren, vgl. dazu Kap. 7, *Eine Datenbank konzipieren*, und Kap. 32, *Datenmodellierung für Fortgeschrittene*, wo diese Fragen in den entsprechenden Punkten ausführlich behandelt werden.
- In aller Regel ist es erforderlich, eine vom Tabellen-Assistenten erstellte Tabellendefinition nachträglich zu überarbeiten. Dies betrifft vor allem die Datentypen, diverse Feldeigenschaften, z.B. Standardwerte, Gültigkeitsregeln oder Eingabepflicht sowie das Festlegen von Indizes. Zu diesem Zweck müssen Sie die Tabelle in der Entwurfsansicht betrachten und

darin die gewünschten Änderungen vornehmen. Inhaltlich wird auf alle Möglichkeiten, die Definition einer Tabelle zu bearbeiten, in den weiteren Punkten dieses Kapitels eingegangen.

8.2.2 Neue Tabelle in der Datenblattansicht erstellen

Sie können eine neue Tabelle auch in der Datenblattansicht erstellen und dabei gleich Daten in die einzelnen Felder eingeben, von denen Access Ihnen beliebig viele anbietet. Diese Vorgehensweise hat den Vorteil, daß Access die Datentypen entsprechend den von Ihnen eingegebenen Werten festlegt, so daß beispielsweise ein Feld, in das Sie einen Währungsbetrag eingegeben haben, den Datentyp *Währung* zugewiesen bekommt oder eines, das eine Zahl mit Dezimalstellen enthält, den Datentyp *Zahl* mit der Feldgröße *Double*. Die Feldnamen werden zunächst von Access als *Feld1*, *Feld2* usw. vergeben. Sie können diese Namen – wie auch jedes andere definitorische Tabellenmerkmal – selbstverständlich ändern. Gehen Sie folgendermaßen vor, um eine Tabelle in der Datenblattansicht neu zu erstellen:

- Öffnen Sie das Dialogfeld *Neue Tabelle*, indem Sie im Register *Tabellen* des Datenbankfensters (aktivierbar mit F11) auf die Schaltfläche *Neu* oder in der Symbolleiste auf die Symbol-Schaltfläche *Neues Objekt* (vgl. links nebenstehend) klicken.
- Wählen Sie im Dialogfeld *Neue Tabelle* (vgl. oben Bild 8.1) den Listeneintrag *Datenblattansicht*, und bestätigen Sie mit OK. Dann wird ein Fenster mit einer neuen und leeren Tabelle in der Datenblattansicht angezeigt, vgl. Bild 8.3.

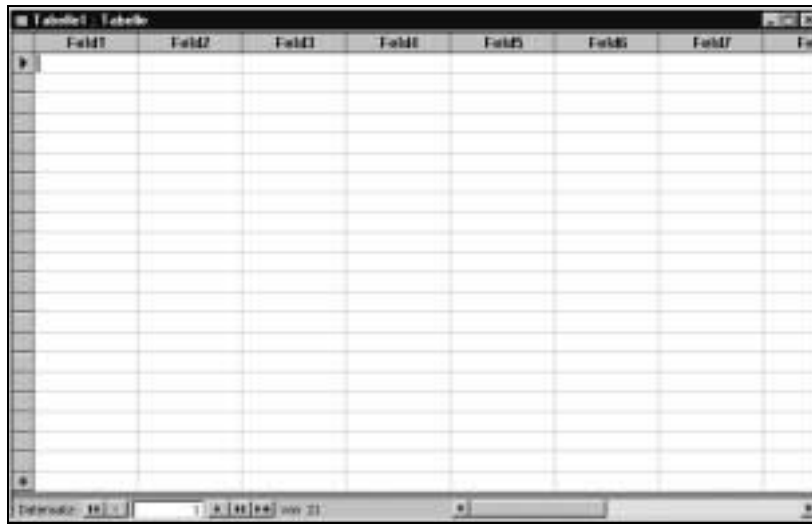


Bild 8.3: Leere Tabelle in der Datenblattansicht, wie diese von Access unmittelbar nach deren Anforderung angeboten wird

- *Felder umbenennen*. Obwohl formal nicht erforderlich, sollten Sie aus inhaltlichen Gründen die Felder bereits in der Datenblattansicht umbenennen. Verfahren Sie dazu wie folgt: Doppelklicken Sie auf den Spaltenkopf (in der Datenblattansicht auch Feldmarkierer genannt) des umzubenennenden Feldes, ändern Sie den Namen, und bestätigen Sie die Änderung durch Drücken der [↵]-Taste. Die Änderung von Feldnamen ist im übrigen auch jederzeit später möglich – sowohl in der Datenblattansicht wie in der Entwurfsansicht.

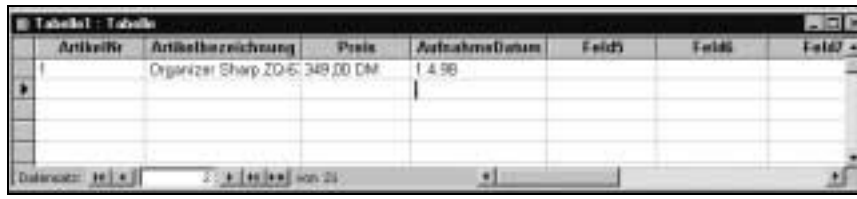


Bild 8.4: Neue Tabelle in der Datenblattansicht vor dem Speichern, jedoch mit umbenannten Feldern und eingegebenen Daten für einen Datensatz. Beachten Sie, daß in diesem Stadium sämtliche Werte als Zeichenketten linksbündig angeordnet sind.

- *Daten eingeben*. Geben Sie für jedes Feld Daten ein, zumindest einen Datensatz, damit Access vor dem Speichern den zu den eingegebenen Daten passenden Datentyp ermitteln kann. Geben Sie die Daten ggf. in formatierter Form ein, d.h. Währungsbeträge mit nachgestelltem Währungssymbol (in Deutschland DM) oder Datumswerte im entsprechenden Standardformat. Sämtliche Werte werden zunächst wie Textwerte dargestellt, also linksbündig.
- *Weitere Spalten einfügen*. Wenn Sie mehr als 20 Spalten benötigen, können Sie weitere Spalten folgendermaßen einfügen: Klicken Sie auf die Spalte, vor der eine neue Spalte eingefügt werden soll, und wählen Sie dann im Menü *Einfügen* den Befehl *Spalte*. Benennen Sie die neue Spalte ggf. wie oben beschrieben um.

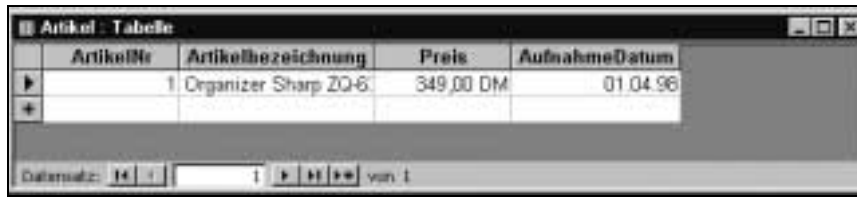


Bild 8.5: Tabelle in der Datenblattansicht nach dem Speichern: Sie trägt jetzt den Namen Artikel, die Zahlen-, Währungs- und Datumswerte werden rechtsbündig, der Textwert wird linksbündig angeordnet.

- *Tabelle speichern*. Sie können das Speichern der Tabelle explizit oder implizit veranlassen. Im ersteren Fall klicken Sie in der Symbolleiste auf die Symbol-Schaltfläche *Speichern* oder drücken die Tastenkombination

Strg + S. Sie veranlassen aber auch ein Speichern, wenn Sie von der Datenblattansicht der bisher nicht gespeicherten Tabelle in die Entwurfsansicht wechseln, denn Access läßt diesen Wechsel nur zu, wenn Sie die Tabelle zuvor gespeichert haben. Gleichgültig, welchen Weg Sie wählen, müssen Sie einen Namen für die Tabelle vergeben. Da in der Datenblattansicht kein Primärschlüssel festgelegt werden konnte, fragt Access beim Speichern, ob dieser nunmehr vergeben werden soll. Wenn Sie die Frage mit Ja beantworten, wird Access eine neues Feld mit dem Namen *ID* erzeugen, diesem den Datentyp *AutoWert* geben und das Feld zum Primärschlüsselfeld machen. Falls sich unter den von Ihnen in der Datenblattansicht definierten Feldern eines befindet, das diese Aufgabe übernehmen könnte, sollten Sie auf die Vergabe eines Primärschlüssels durch Access verzichten und nach dem Speichern in der Entwurfsansicht der Tabelle selbst den Primärschlüssel festlegen. Im obigen Beispiel aus Bild 8.4 würde sich dazu das Feld *ArtikelNr* anbieten, weil es eindeutige Werte enthalten wird. Genaueres über die Bedeutung des Primärschlüssels erfahren Sie in diesem Kapitel im Punkt 8.9, *Primärschlüssel setzen, ändern und löschen*.

Bild 8.6: Tabelle »Artikel«, die in der Datenblattansicht entworfen wurde, nach dem Speichern in der Entwurfsansicht. Die Datentypen wurden von Access, der Primärschlüssel wurde vom Benutzer festgelegt.

- *Definitionsmerkmale in der Entwurfsansicht ändern*. Die Leistung von Access, aus den von Ihnen eingegebenen Daten eine brauchbare Tabellendefinition zu erzeugen, ist beachtlich: In Bild 8.6 ist die gespeicherte Tabelle *Artikel* in der Entwurfsansicht wiedergegeben. Dort ist zu erkennen, daß Access korrekt die jeweils zu den eingegebenen Daten passenden Datentypen *Zahl*, *Text*, *Währung* und *Datum/Uhrzeit* ermittelt und zuge-

wiesen hat. Gleichwohl werden Sie im allgemeinen Änderungen und Ergänzungen an der automatisch erzeugten Tabellendefinition vornehmen müssen, um Ihre Vorstellungen von Datenmodellierung zu realisieren. Zu diesem Zweck wechseln Sie von der Datenblattansicht der Tabelle in deren Entwurfsansicht. Falls Sie die Änderungen am Tabellenentwurf erst später vornehmen wollen, können Sie die gespeicherte Tabelle auch zunächst schließen und später vom Datenbankfenster aus in der Entwurfsansicht öffnen. Die Bearbeitung der verschiedenen Definitionsmerkmale einer Tabelle wird ausführlich in den weiteren Punkten dieses Kapitels besprochen.

8.2.3 Neue Tabelle in der Entwurfsansicht erstellen

Um eine neue Tabelle in der Entwurfsansicht zu erstellen, gehen Sie folgendermaßen vor:

- Öffnen Sie das Dialogfeld *Neue Tabelle*, indem Sie in der Objektliste *Tabellen* des Datenbankfensters (aktivierbar mit F11) auf die Schaltfläche *Neu* klicken oder in der Symbolleiste in der Liste *Neues Objekt* das Objekt *Tabelle* wählen.
- Wählen Sie im Dialogfeld *Neue Tabelle* (vgl. oben Bild 8.1) den Listeneintrag *Entwurfsansicht*, und bestätigen Sie mit OK. Dann wird ein Fenster mit einer neuen und leeren Tabelle in der Entwurfsansicht angezeigt, vgl. Bild 8.7 (dort ist allerdings bereits ein Feld eingetragen worden).

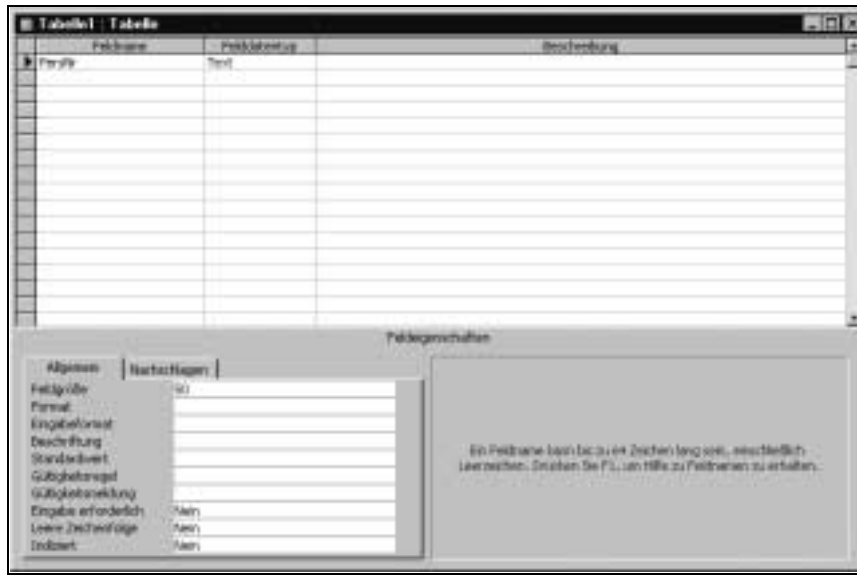


Bild 8.7: Neue (fast) leere Tabelle in der Entwurfsansicht, in die bisher lediglich der Name »PersNr« eingefügt wurde

Bei der Arbeit am Tabellenentwurf müssen Sie im allgemeinen die folgenden Punkte berücksichtigen:

- **Feldnamen**. Sie schreiben einen Feldnamen in die Spalte *Feldname* oder bearbeiten einen vorhandenen.
- **Felddatentypen**. Sie wählen einen Felddatentyp aus dem Dropdown-Listenfeld in der Spalte *Felddatentyp* aus.
- **Feldeigenschaften**. Sie legen eine von mehreren Feldeigenschaften fest, indem Sie einen entsprechenden Eintrag in eines der Eingabefelder für Feldeigenschaften vornehmen; einige dieser Eingabefelder bieten Ihnen, wenn sie den Fokus haben, ein Dropdown-Listenfeld zur Auswahl an. Das Festlegen der Feldeigenschaften erlaubt die differenzierteste Gestaltung eines Tabellenfeldes.
- **Primärschlüssel**. Sie setzen den Primärschlüssel für das Feld, das den Fokus hat (entsprechend für mehrere Felder).
- **Index**. Sie indizieren ein Feld über seine Feldeigenschaft *Indiziert*. Für mehrere Felder bilden Sie einen Index durch entsprechende Einträge im Fenster *Indizes*, das Sie in der Entwurfsansicht der Tabelle durch Klicken auf die Symbol-Schaltfläche *Indizes* (vgl. links nebenstehend) oder mit dem gleichnamigen Befehl aus dem Menü *Ansicht* öffnen.
- **Tabelleneigenschaften**. Im Eigenschaftenfenster für die Tabelle, das Sie mit der Symbol-Schaltfläche *Eigenschaften* öffnen (vgl. links nebenstehend), legen Sie Tabelleneigenschaften fest. Bedeutsam ist dies vor allem für die Gültigkeitsregel eines Datensatzes (statt eines Feldes).

Beziehungen zwischen Tabellen werden dagegen nicht in der Entwurfsansicht definiert; vielmehr müssen dazu die beteiligten beiden Tabellen geschlossen sein, vgl. unten, Punkt 8.11, *Beziehungen zwischen Tabellen festlegen*.

Wie Sie für die Bearbeitung eines Tabellenentwurfs im einzelnen vorgehen können und welche Überlegungen dabei zu berücksichtigen sind, erfahren Sie in den folgenden Punkten dieses Kapitels.

8.3 Felder hinzufügen, löschen, kopieren, verschieben

8.3.1 Felder hinzufügen oder Feldnamen ändern

Sie fügen einer Tabelle ein neues Feld hinzu, indem Sie einen gültigen Feldnamen in ein freies Feld der Spalte *Feldname* schreiben. Namen dürfen in Access bis zu 64 Zeichen lang sein und eine beliebige Kombination aus Buchstaben, Ziffern, Leerzeichen und Sonderzeichen (außer Punkt, Ausrufezeichen, Accent grave (') und eckige Klammern) darstellen. Sie dürfen ferner nicht mit einem Leerzeichen beginnen. Diese Regel gilt für alle Objektnamen (Namen für Tabellen, Formulare, Abfragen, Felder etc.). Inhaltlich sollten

Sie darauf achten, sprechende Namen zu bilden, d.h. solche, die auf den Inhalt verweisen. Mit bis zu 64 Zeichen sollte dies nicht schwer fallen. Gleichwohl spricht auch einiges für kurze Namen und die Vermeidung von Leer- und Sonderzeichen: Kurze Namen schreiben sich leichter. Wenn Sie sich in einem Ausdruck auf einen Namen beziehen, der Leer- oder Sonderzeichen enthält, müssen Sie den Namen zwischen eckige Klammern setzen. Andererseits sparen Sie sich das Schreiben der eckigen Klammern (in fast jedem Kontext), wenn Sie keine Leer- und Sonderzeichen in Namen verwenden.

Einen bestehenden Feldnamen können Sie jederzeit durch Überschreiben ändern. Für die Tabelle selbst ist dies ohne jede Komplikation möglich. Falls Sie allerdings bereits Formulare, Abfragen etc. erstellt haben, die sich auf das Feld unter seinem alten Namen beziehen, müssen Sie dort entsprechende Änderungen vornehmen, denn Access erledigt dies für Sie nicht automatisch, sondern meldet statt dessen einen Fehler (z.B. #Name?).

8.3.2 Felder mit dem Feld-Generator hinzufügen

Sie können einzelne Felder auch vom Feld-Generator einfügen lassen. Der Feld-Generator stellt die Beispieltabellen und deren Felder zur Verfügung, die auch der Tabellen-Assistent anbietet, vgl. oben Punkt 8.2.1, *Neue Tabelle mit dem Tabellen-Assistenten erstellen*. Gehen Sie zum Einfügen eines Feldes mit dem Feld-Generator folgendermaßen vor:

- Klicken Sie in die Spalte *Feldname* der Feldzeile, in die Sie das Feld einfügen möchten, so daß diese den Fokus bekommt.

Bild 8.8: Kontextmenü der Entwurfsansicht einer Tabelle

- Klicken Sie in der Symbolleiste auf die Symbol-Schaltfläche *Aufbauen*. *Alternativ*: Schlagen Sie das *Kontextmenü* durch Klicken mit der rechten Maustaste auf die Spalte *Feldname* der Feldzeile auf, in die Sie das Feld einfügen möchten. Wählen Sie darin den Befehl *Aufbauen ...*, vgl. Bild 8.8. Dann zeigt sich das Dialogfeld *Feld-Generator*, vgl. Bild 8.9.

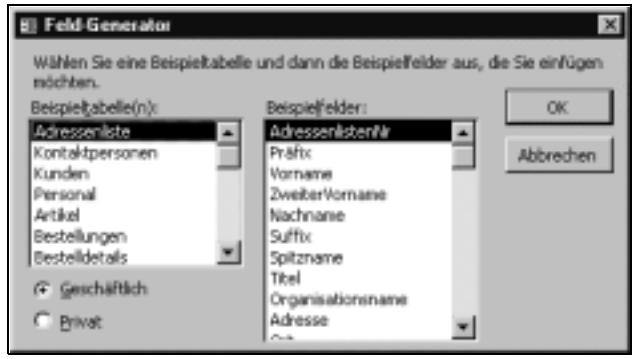


Bild 8.9: Dialogfeld »Feld-Generator«

- Wählen Sie im Dialogfeld *Feld-Generator* die passende Beispieltabelle und dann das passende Beispielfeld, und bestätigen Sie mit OK. Darauf wird das Feld mit seinen Definitionsmerkmalen eingefügt. Sämtliche Eigenschaften des Feldes lassen sich danach in der Entwurfsansicht der Tabelle bearbeiten.

8.3.3 Felder löschen

Sie entfernen ein bereits definiertes Feld aus der Tabelle auf die folgende Weise:

- Markieren Sie das Feld in der Entwurfsansicht der Tabelle, indem Sie auf den Feldmarkierer (das kleine Kästchen am linken Zeilenrand) klicken, vgl. Bild 8.10.

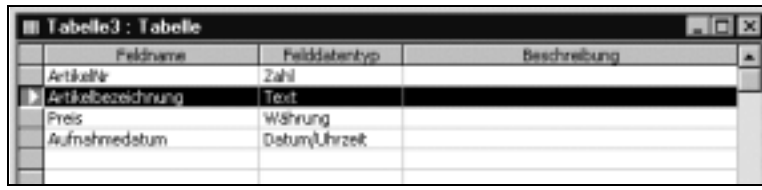


Bild 8.10: Die Feldzeile für »Artikelbezeichnung« wurde markiert. Wenn Sie in dieser Situation die Taste [Entf] drücken, werden Sie vor dem endgültigen Löschen vor möglichem Datenverlust gewarnt.

- Drücken Sie die [Entf]-Taste, oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten* oder aus dem *Kontextmenü*. Falls das Feld bereits Daten enthält, werden Sie durch eine entsprechende Meldung, die einen Abbruch ermöglicht, darauf hingewiesen.

Auch das Löschen eines Feldes wird von Access nicht automatisch an den Stellen, wo darauf Bezug genommen wird (Formulare, Berichte etc.), korrigiert. Dies müssen Sie manuell erledigen, damit Fehlermeldungen vermieden werden.

8.3.4 Felder kopieren

Sie können ein Feld mit all seinen Definitionsmerkmalen (Name, Datentyp, Eigenschaften) innerhalb derselben Tabelle, aber auch in eine andere Tabelle kopieren. Verfahren Sie dazu wie folgt:

- Markieren Sie die Zeile des Quellfeldes durch Klicken auf den Feldmarkierer.
- Drücken Sie die Tastenkombination [Strg]+[C], klicken Sie auf die Symbol-Schaltfläche *Kopieren* (vgl. links nebenstehend), oder wählen Sie den Befehl *Kopieren* aus dem Menü *Bearbeiten*.
- Setzen Sie den Cursor in eine beliebige Zelle der Zielzeile, und drücken Sie die Tastenkombination [Strg]+[V], klicken Sie auf die Symbol-Schaltfläche Einfügen (vgl. links nebenstehend), oder wählen Sie den Befehl *Einfügen* aus dem Menü *Bearbeiten*. Dann wird die Quellzeile eingefügt, so daß die darunterliegenden Felder nach unten verschoben werden. Wenn die ganze Zielzeile markiert ist, wird diese durch den Inhalt der Quellzeile überschrieben.

8.3.5 Felder verschieben

Die Reihenfolge der Felder in der Tabellendefinition hat weder für den logischen noch für den physikalischen Tabellenentwurf eine Bedeutung. Sie wirkt sich vermutlich jedoch für Sie als Entwickler der Datenbank auf die Übersichtlichkeit aus. Sie können die Feldreihenfolge jederzeit – während des ersten Entwurfs, aber auch nachträglich – in der Entwurfsansicht der Tabelle ändern, ohne daß die Funktionsweise von Objekten, die sich auf die Felder der betreffenden Tabelle unter Verwendung von Feldnamen beziehen, davon beeinträchtigt wird. (Wenn Sie sich mit einer VBA-Methode auf die *Position* eines Feldes in der Tabellendefinition beziehen, wirkt sich eine Änderung der Feldreihenfolge allerdings aus.) Gehen Sie folgendermaßen vor, um Felder in der Tabellendefinition zu verschieben:

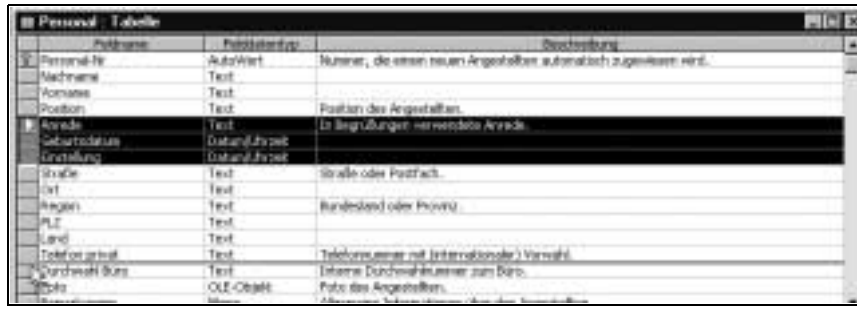


Bild 8.11: Die drei markierten Felder werden an die Position vor das Feld »Durchwahl Büro« verschoben.

- Markieren Sie, während sich die Tabelle in der Entwurfsansicht befindet, die zu verschiebende(n) Zeile(n) durch Klicken auf den Feldmarkierer bzw. durch Ziehen mit der Maus.
- Klicken Sie erneut auf den (die) markierten Feldmarkierer, und ziehen Sie die Zeile(n) an die gewünschte Stelle, vgl. Bild 8.11.

8.4 Felddatentypen festlegen

Für jedes Feld muß ein Datentyp festgelegt sein. Der Datentyp – zusammen mit den ihn weiter spezifizierenden Feldeigenschaften – wirkt sich auf den für ein Feld zulässigen *Wertebereich* (die sogen. Domäne), die möglichen *Rechenoperationen* sowie das *Sortieren* aus. Wenn Sie ein neues Feld durch Eintrag seines Feldnamens festgelegt haben, gibt Access ihm den Datentyp, der als *Standardfeldtyp* festgelegt ist. Per Voreinstellung ist dies der Datentyp *Text*. Sie weisen einem Feld einen anderen Felddatentyp zu, indem Sie auf die Spalte *Felddatentyp* der entsprechenden Feldzeile klicken, dort das Dropdown-Listenfeld aufschlagen und den geeigneten Datentyp auswählen.

Sie können den Standardfeldtyp ändern. Schlagen Sie dazu die Registerkarte *Tabellen/Abfragen* zum Befehl *Optionen* aus dem Menü *Extras* auf. Wählen Sie dort einen anderen Standardfeldtyp aus der Dropdown-Liste aus.

Bei der Wahl des geeigneten Datentyps beachten Sie die folgenden allgemeinen Überlegungen:

- *Wertebereich*. Der Datentyp legt fest, welche Art von Werten in ein Feld eingegeben werden kann. So lassen sich in ein Feld des Datentyps *Zahl* keine alphanumerischen Zeichen, sondern nur Zahlenwerte eingeben, und ein Feld mit dem Datentyp *Datum/Uhrzeit* nimmt nur Datums- oder Zeitangaben an.
- *Rechenoperationen*. Der Datentyp bestimmt auch, in welcher Weise die Werte eines Feldes von Access interpretiert und verarbeitet werden können. Ein Vergleich der beiden Datentypen *Text* und *Zahl* macht dies deutlich: Sie können dieselbe Zeichenfolge »123« sowohl in ein Feld mit dem Datentyp *Text* wie auch in ein solches mit dem Datentyp *Zahl* eingeben. Im ersten Fall wird die Zeichenfolge als Text interpretiert, so daß es nicht möglich ist, damit zu rechnen. Im zweiten Fall wird »123« als Zahlenwert 123 aufgefaßt, mit dem dann Rechenoperationen ausgeführt werden können.
- *Sortieren*. Felder des Datentyps *Memo*, *Hyperlink* und *OLE-Objekt* können nicht sortiert werden. In einem Feld des Datentyps *Text* werden Zahlen als Zeichenfolgen und nicht als numerische Werte sortiert, z.B. in der Reihenfolge 1, 10, 100, 2, 20, 200. Bei den Datentypen *Zahl* oder *Währung* wird dagegen nach den numerischen Werten sortiert, z.B. in der Reihenfolge 1, 2, 10, 20 100, 200. Datumswerte, die in ein Feld vom Daten-

typ *Text* eingegeben wurden, werden je nach Datumsformat unterschiedlich und im allgemeinen nicht richtig sortiert. Sie sollten daher stets den Datentyp *Datum/Uhrzeit* für Felder verwenden, die Datumswerte enthalten sollen.

Die Übersicht in Tabelle 8.1 gibt die in Access verfügbaren Datentypen wieder und führt die wichtigsten Merkmale an. Weitere Eigenschaften entnehmen Sie bitte der Online-Hilfe.

Datentypen und ihre wichtigsten Eigenschaften

Felddatentyp	Beschreibung	Größe
Text	Nimmt Zeichenfolgen (d.h. alphanumerische Zeichen) auf. Mit der Eigenschaft *Feldgröße* kann die zulässige Länge der eingegebenen Zeichenfolgen definiert werden. Dieser Datentyp ist auch für Ziffernfolgen, die nicht als numerische Werte interpretiert werden sollen, geeignet, beispielsweise für Telefonnummern oder Postleitzahlen. Mit einem Eingabeformat können Sie den Wertebereich auch bei diesem Datentyp weiter einschränken, z.B. für Postleitzahlen auf die Ziffern 0 bis 9.	Maximal 255 Zeichen (= Byte). In der Datenbank wird nur so viel Speicherplatz belegt, wie tatsächlich Zeichenfolgen eingegeben sind.
Memo	Text mit einer großen Länge, geeignet z.B. für Notizen oder längere Beschreibungen. Felder vom Datentyp *Memo* können nicht indiziert werden.	Maximal 64.000 Zeichen (= Byte). In der Datenbank wird nur so viel Speicherplatz belegt, wie tatsächlich Zeichenfolgen eingegeben sind.
Zahl	Numerischer Wert. Zu diesem Datentyp müssen Sie einen von sechs Untertypen angeben. Dies geschieht mit der Feldeigenschaft *Feldgröße*. Im Grunde genommen gibt es gar keinen Datentyp *Zahl*, denn der konkrete Datentyp ist erst durch Auswahl eines Untertyps bestimmt, und Typprüfungen erfolgen stets in Bezug auf diesen Untertyp. Die sechs speziellen Datentypen sind:	2, 4 oder 8 Byte

Tabelle 8.1: Felddatentypen

Felddatentyp	Beschreibung	Größe
	Byte. Ganzzahl mit Werten im Bereich von 0 bis 255	1 Byte
	Integer. Ganzzahl mit Werten im Bereich von -32.768 bis 32.767	2 Bytes
	Long Integer. Lange Ganzzahl mit Werten im Bereich von -2.147.483.648 bis 2.147.483.647	4 Bytes
	Single. Gleitkommazahl mit einfacher Genauigkeit mit Werten im Bereich von -3,402823E38 bis -1,401298E-45 für negative Werte und 1,401298E-45 bis 3,402823E38 für positive Werte.	4 Byte
	Double. Gleitkommazahl mit doppelter Genauigkeit mit Werten im Bereich von -1,79769313486232E308 bis -4,94065645841247E-324 für negative Werte und 4,94065645841247E-324 bis 1,79769313486232E308 für positive Werte.	8 Bytes
	Replikations-ID. Dient zum Definieren einer eindeutigen Replikationskennung und muß für Tabellen, die repliziert werden sollen, definiert sein. Wird auch als Globally Unique Identifier (GUID) bezeichnet.	16 Bytes
Datum/ Uhrzeit	Datum- und Zeitwerte für die Jahre 100 bis 9999. Bei der Eingabe werden die Jahresangaben 0 bis 29 als 2000 bis 2029 und 30 bis 99 als 1930 bis 1999 interpretiert.	8 Byte
Währung	Währungswerte mit einer Genauigkeit von bis zu 15 Stellen und 4 Dezimalstellen. Dieser Datentyp wird für Währungswerte verwendet, um bei Berechnungen ein Abrunden zu verhindern.	8 Byte

Tabelle 8.1: Felddatentypen

Felddatentyp	Beschreibung	Größe
AutoWert	Zahl, die von Access automatisch vergeben wird und die eindeutig ist (ohne Duplikat), wenn ein neuer Datensatz einer Tabelle hinzugefügt wird. Felder vom Datentyp *AutoWert* können nicht verändert werden. Zahlen, die bereits vergeben waren, stehen auch nach dem Löschen eines Datensatzes nicht mehr zur Verfügung. Sie können bestimmen, ob neue Zahlen jeweils um den Wert 1 erhöht oder als Zufallszahl eingefügt werden.	4 Byte
Ja/Nein	Ja- und Nein-Werte. Statt Ja dürfen Sie auch -1 oder Wahr oder Ein, statt Nein auch 0 oder Falsch oder Aus eingeben.	1 Bit (= 1/8 Byte)
OLE-Objekt	Objekte, wie z.B. Excel-Tabellen, Word-Dokumente, Bilder, Klänge oder andere mit binären Daten, die von einem OLE-Server erzeugt worden sind. Die Größe kann maximal 1 Gigabyte betragen. Felder vom Datentyp *OLE-Objekt* können nicht indiziert werden.	Maximal 1 GB
Hyperlink	Text oder Kombinationen aus Text und Zahlen, die als Text abgespeichert und als Hyperlink-Adresse verwendet werden. Eine Hyperlink-Adresse besteht aus bis zu drei Teilen: *Anzeigetext*: Der Text, der in einem Feld oder einem Steuerelement angezeigt wird. *Adresse*: Der Pfad auf eine lokale Datei (UNC-Pfad) oder eine Seite im Internet oder im Intranet (URL). *UnterAdresse*: Eine bestimmte Stelle innerhalb einer Datei oder Seite.	Jeder der drei nebenstehend angegebenen Teile kann bis zu 2048 Zeichen lang sein.

Tabelle 8.1: Felddatentypen

Auch Felddatentypen können nachträglich verändert werden. Allerdings können dabei, wenn das Feld bereits Daten enthält, Werte verloren gehen, je nach Umwandlungsart. Wenn Sie beispielsweise den Felddatentyp *Text* in den Typ *Zahl* umwandeln, gehen die bisherigen Textwerte verloren. Umgekehrt bleiben Zahlen erhalten, wenn Sie ein Feld vom Typ *Zahl* in einen solchen vom Typ *Text* umwandeln; die Zahlen werden dann allerdings als Text interpretiert.

Access meldet den möglichen Datenverlust durch Konvertieren eines Datentyps in einen anderen erst, wenn Sie den Tabellenentwurf zu speichern versuchen, vgl. die Meldung in Bild 8.12, die Ihnen die Möglichkeit zum Abbruch gibt. Wenn Access die Werte in den neuen Datentyp ohne Fehler konvertieren kann, unterbleibt diese Meldung, und es gibt keine Möglichkeit der Korrektur mehr. Beachten Sie, daß die aus Sicht von Access fehlerfreie Konvertierung möglicherweise aus Ihrer Sicht zu unsinnigen Ergebnissen führt. Bei wertvollen Daten sollten Sie daher zunächst einen Test mit einer Testtabelle ausführen, um die Wirkung der Konvertierung zu überprüfen.

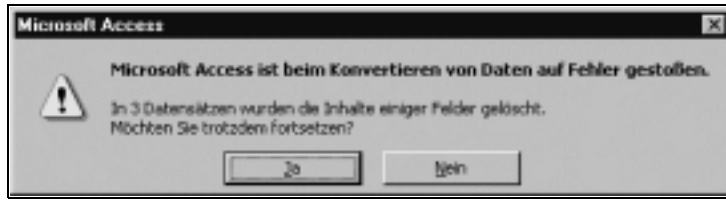


Bild 8.12: Meldung nach dem Ändern eines Datentyps beim Speichern des Tabellenentwurfs.

8.5 Feldeigenschaften festlegen

Mit den Feldeigenschaften können Sie erheblichen Einfluß auf das Aussehen, die Eingabesicherheit oder die Effizienz der Verwaltung Ihrer Daten nehmen. Beispielsweise können Sie ein Anzeigeformat definieren, so daß Zahlenwerte mit drei Dezimalstellen und der Bezeichnung *kg* ausgegeben werden. Ein anderes Beispiel: Sie geben als Gültigkeitsregel an, daß das Feld *Preis* bei der Dateneingabe keinen negativen Wert annehmen kann.

Die in einer Tabelle für die Felder festgelegten Eigenschaften haben unmittelbare Gültigkeit für die Dateneingabe in das Datenblatt der Tabelle. Auf Formulare wirken sie sich z.T. nur mittelbar aus: Wenn Sie ein Formular durch den Formular-Assistenten generieren lassen, werden die Feldeigenschaften als entsprechende Eigenschaften der jeweiligen Steuerelemente übernommen. Beispielsweise wird das Eingabeformat, das Sie für ein Tabellenfeld definiert haben, als Eingabeformat für das entsprechende Textfeld (oder sonstige Steuerelement) übernommen. Falls Sie jedoch nachträglich eine Feldeigenschaft der Tabelle ändern, wird dies nicht automatisch auf das entsprechende Steuerelement in einem Formular übertragen. Diese Anpassung müßten Sie dann ggf. manuell vornehmen. Aus diesem Grunde empfiehlt es sich, die Feldeigenschaften einer Tabelle sorgfältig zu definieren, so daß sie möglichst vollständig und richtig sind, bevor Sie darauf ein Formular aufbauen. Die Gültigkeitsregel eines Feldes wirkt allerdings auch dann, wenn diese im entsprechenden Steuerelement eines Formulars nicht angegeben ist. Umgekehrt gilt: Eine Gültigkeitsregel im Steuerelement eines Formu-

lars, die von der Gültigkeitsregel in dem Feld, an welches das Steuerelement gebunden ist, abweicht, wirkt auch für sich allein, so daß in das Steuerelement des Formulars nur ein Wert eingegeben werden kann, der beiden Regeln entspricht.

Sie legen Feldeigenschaften für ein bestimmtes Tabellenfeld folgendermaßen fest:

- Geben Sie dem zu bearbeitenden Feld den Fokus, indem Sie im oberen Teil des Tabellenfensters in der Entwurfsansicht auf eine beliebige Zelle seiner Feldzeile klicken.
- Dann erscheinen im unteren Teil des Fensters die Bearbeitungsfelder der Feldeigenschaften, die für das Feld, das den Fokus hat, eingestellt werden können. In Bild 8.13 beispielsweise hat das Feld *AuftragsNr* den Fokus, dem der Datentyp *Zahl* zugewiesen ist. Daher werden im unteren Fensterteil die Feldeigenschaften angezeigt, die zu diesem Datentyp verfügbar sind, das sind u.a. *Feldgröße* und *Dezimalstellen*. Für ein Feld mit dem Datentyp *Datum/Uhrzeit* wären diese beiden Eigenschaften dagegen nicht verfügbar. Die Bearbeitungsfelder einiger Eigenschaften bieten ein Dropdown-Listenfeld an, wenn sie den Fokus haben. In Bild 8.13 können Sie dies beispielsweise für die Eigenschaft *Feldgröße* erkennen. In diesem Fall können Sie die gewünschte Feldeigenschaft aus dem Dropdown-Listenfeld auswählen. Es ist aber auch möglich, sie einzutippen.

Bild 8.13: Tabelle in der Entwurfsansicht. Im unteren Fensterteil befinden sich die Bearbeitungsfelder für die Feldeigenschaften.

Wenn für eine Eigenschaft eine Liste mit Werten in Form einer Dropdown-Liste angeboten wird, wählen Sie den nächsten Wert der Liste aus, wenn Sie auf den Eintrag doppelklicken. Dies stellt eine Bearbeitungsalternative zum Aufschlagen der Dropdown-Liste dar. Allerdings kann diese Wirkung des

Doppelklickens auch leicht zu unbemerkten Fehlern führen: Da es in allen Bearbeitungsfeldern unter Windows möglich ist, ein einzelnes Wort durch Doppelklicken zu markieren, haben Sie sich diese Arbeitsweise vielleicht mehr oder minder unbewußt angewöhnt. Wenn Sie in einem Bearbeitungsfeld einer Eigenschaft, das eine Dropdown-Liste anbietet, auch nur auf ein Wort doppelklicken, wird der nächste Wert der Liste in das Feld eingetragen!

8.5.1 Feldgröße

Die Eigenschaft *Feldgröße* steht nur für Felder mit dem Datentyp *Text* oder *Zahl* zur Verfügung. Wenn Sie einen Datentyp *Text* oder *Zahl* für ein Feld festlegen, setzt Access dafür als Feldgröße einen Standardwert ein. Unmittelbar nach dem Installieren von Access 2000 ist dies 50 für den Datentyp *Text* und *Long Integer* für den Datentyp *Zahl*. Diese Standardfeldgrößen können Sie in der Registerkarte *Tabellen/Abfragen* des Befehls *Optionen* aus dem Menü *Extras* ändern.

Text. Bei Feldern dieses Datentyps geben Sie eine Zahl zwischen 1 und 255 für die maximale Anzahl von Textzeichen ein. Die Beschränkung der Feldgröße ist besonders geboten, wenn das Feld Primärschlüssel sein soll, weil dann die Effizienz besser wird. Der vom Feld benötigte Speicherplatz wird dagegen durch eine große Feldlänge nicht automatisch vergrößert, weil Access auch bei Feldern mit großer Feldgröße nur so viel Speicherplatz benötigt, wie er den tatsächlich eingegebenen Textlängen der einzelnen Datensätze entspricht.

Zahl. Für Felder vom Datentyp *Zahl* legen Sie mit der Feldgröße den speziellen Datentyp fest. Diesen wählen Sie aus dem Dropdown-Listenfeld aus, das für den Datentyp *Zahl* für die Eigenschaft *Feldgröße* angeboten wird. Orientieren Sie sich bei der Wahl der Feldgröße für ein Zahlenfeld an den diesbezüglichen Informationen zum Datentyp *Zahl*, die Sie oben im Punkt 8.4, *Felddatentypen festlegen*, Tabelle 8.1, finden.

8.5.2 Anzeigeformat

Das Anzeigeformat wird mit der Feldeigenschaft *Format* eingestellt. Wenn Sie kein besonderes Anzeigeformat festlegen, gilt das von Access für diesen Fall vorgesehene Standardformat, mit dem Sie stets vernünftige Anzeigeformen bekommen. So werden Werte von Feldern mit dem Datentyp *Datum/Uhrzeit* automatisch in einem im jeweiligen Land üblichen Datums- und ggf. Zeitformat dargestellt. Entsprechendes gilt auch für den Datentyp *Währung*. Neben den Standardformaten bietet Access für Felder der Datentypen *Zahl*, *AutoWert* und *Währung* darüber hinaus die folgenden vordefinierten Formate an, die Sie aus dem Dropdown-Listenfeld des Eingabefeldes *Format* wählen können:

Allgemeine Zahl	3456,789
Währung	3.456,79 DM
Euro	3.456,79 €
Festkommazahl	3456,79
Standardzahl	3.456,79
Prozentzahl	123,00%
Exponentialzahl	3,46E+03

Bild 8.14: Standard-Anzeigeformate.

Zu jedem der in Bild 8.14 angegebenen Formate läßt sich als Eigenschaft die Anzahl der Dezimalstellen angeben, indem Sie in das Bearbeitungsfeld der Eigenschaft *Dezimalstellen* eine ganze Zahl schreiben. Über diese Standardformate hinaus können Sie nahezu jede beliebige Anzeigeform erreichen, wenn Sie ein benutzerdefiniertes Format eingeben. Dazu können Sie bestimmte Symbole für Ziffern, Farben, Datumswerte etc. verwenden. Mit dem benutzerdefinierten Format

#.##0,00« Rubel«;-#.##0,00« Rubel"[Rot]

für ein Feld vom Typ *Zahl* beispielsweise erreichen Sie, daß die Zahlenwerte dieses Feldes mit Tausenderteilungspunkt, zwei Dezimalstellen und nachgestellter Angabe *Rubel* dargestellt werden, wobei positive Werte in der Standardtextfarbe, negative dagegen rot ausgegeben werden. Kurz: Die Zahl 1234,5 wird dargestellt als *1.234,50 Rubel*. In Punkt 8.7, *Benutzerdefinierte Anzeigeformate*, wird ausführlich beschrieben, wie Sie benutzerdefinierte Formate für die verschiedenen Datentypen angeben können.

8.5.3 Eingabeformat

Mit einem Eingabeformat legen Sie fest, in welcher Art und Weise die Daten in ein Tabellenfeld eingegeben werden. Für ein Feld, das Postleitzahlen speichert, können Sie beispielsweise das Eingabeformat so festlegen, daß fünf Ziffern eingegeben werden müssen, oder, als Alternative, daß den fünf Ziffern ein führender Länderbuchstabe und ein Bindestrich vorangestellt wird. Ein Eingabeformat wirkt wie eine Maske, die über das Tabellenfeld gelegt wird. Es werden nur Eingaben akzeptiert, die dem festgelegten Format entsprechen. Ein besonderes Format definieren Sie, wenn Sie das Eingabeformat auf den Wert *Kennwort* einstellen. Dann werden die in das Feld eingegebenen Zeichen als Sternchen (*) angezeigt, so daß diese vor dem Lesen durch Dritte (aber auch durch Sie!) geschützt sind.

Um ein Eingabeformat zu erstellen, geben Sie das Format in das Bearbeitungsfeld der Eigenschaft *Eingabeformat* ein. Sie können sich dabei auch des Eingabeformat-Assistenten bedienen, der weiter unten am Ende dieses Punktes kurz erläutert wird. Ein Eingabeformat besteht aus bis zu drei Abschnitten, die jeweils durch ein Semikolon gegeneinander getrennt sind:

- *Der erste Abschnitt* bestimmt das Eingabeformat selbst. Hier geben Sie z.B. ein Format in der Form *?- 0000!* an.
- *Der zweite Abschnitt* bestimmt, ob die im Eingabeformat ggf. definierten Literalzeichen in der Tabelle gespeichert werden. Wenn Sie hier *0* eingeben, werden alle Literalzeichen des Eingabeformats zusammen mit dem Wert gespeichert. Für eine Versicherungsnummer könnten Sie beispielsweise im ersten Formatabschnitt das Format *00"/"000"/"000000* angeben. Dann würden die Zeichen »/« zusammen mit den Ziffern in der Tabelle gespeichert werden, wenn Sie im zweiten Abschnitt den Wert 0 angeben. Geben Sie 1 ein oder lassen Sie den zweiten Abschnitt leer, werden nur die in das Feld direkt *eingegebenen* Zeichen gespeichert, nicht jedoch die durch Formatdefinition erzeugten.
- *Der dritte Abschnitt* legt das Zeichen fest, das Access für Leerzeichen im Eingabeformat anzeigen soll. Hier können Sie jedes Zeichen verwenden. Um ein Leerzeichen anzuzeigen, verwenden Sie ein in Anführungszeichen eingeschlossenes Leerzeichen (» »). Wenn Sie für diesen Abschnitt nichts angeben, werden Leerzeichen im Eingabeformat als Unterstreichungszeichen angezeigt.

Beispiel: Das Eingabeformat

00"/"000"/"000000;0;#

führt zu dieser Anzeige des aktivierten, aber noch leeren Feldes:

##/###/####

Im ersten Formatabschnitt können Sie die nachfolgenden Zeichen in der jeweils angegebenen Bedeutung verwenden:

Zeichen	Beschreibung
0	Ziffer (0-9, Eingabe erforderlich, Plus- [+] und Minuszeichen [-] nicht erlaubt).
9	Ziffer oder Leerzeichen (Eingabe optional, Plus- und Minuszeichen nicht erlaubt).
#	Ziffer oder Leerzeichen (Eingabe optional. Leerstellen werden zu Leerzeichen, Plus- und Minuszeichen erlaubt).
L	Buchstabe (A-Z, Eingabe erforderlich).
?	Buchstabe (A-Z, Eingabe optional).
A	Buchstabe oder Ziffer (Eingabe erforderlich).
a	Buchstabe oder Ziffer (Eingabe optional).
&	Beliebiges Zeichen oder Leerzeichen (Eingabe erforderlich).
C	Beliebiges Zeichen oder Leerzeichen (Eingabe optional).

Tabelle 8.2: Zeichen und ihre Bedeutung für den ersten Abschnitt des Eingabeformates

Zeichen	Beschreibung
. , : ; – /	Platzhalter für Dezimaltrennzeichen, Tausender-, Datums- und Zeit-Trennzeichen. (Das tatsächlich verwendete Zeichen ist abhängig von der Trennzeicheneinstellung in den Ländereinstellungen der Microsoft Windows-Systemsteuerung.)
<	Wandelt alle nachfolgenden Zeichen in Kleinbuchstaben um.
>	Wandelt alle nachfolgenden Zeichen in Großbuchstaben um.
!	Die Eingabe erfolgt von rechts nach links, anstatt von links nach rechts, wenn die Zeichen auf der linken Seite des Eingabeformats optional sind. Das Ausrufezeichen kann an einer beliebigen Stelle des Eingabeformats stehen.
\	Stellt das nachfolgende Zeichen als literales Zeichen dar (\A wird z. B. als A angezeigt).
""	Stellt die zwischen den Anführungszeichen stehende Zeichenfolge als Literalzeichen dar ("MWh" wird als MWh angezeigt).
Kennwort	Durch Einstellen der Eigenschaft *Eingabeformat* auf den Wert *Kennwort* verhält sich ein Feld wie bei der Kennworteingabe: Jedes der in das Feld eingegebenen Zeichen wird zwar als das entsprechende Zeichen gespeichert, aber als Sternchen (*) angezeigt.

Tabelle 8.2: Zeichen und ihre Bedeutung für den ersten Abschnitt des Eingabeformates

Wenn Sie für ein Tabellenfeld ein Eingabeformat festgelegt haben, wirkt sich dies für Abfragen und Textfelder in Formularen folgendermaßen aus:

- *Abfragen*. Das im Tabellenfeld definierte Eingabeformat ist in Abfragen, die dieses Feld anzeigen, voll wirksam.
- *Textfelder in Formularen*. Wenn Sie ein Formular vom Formular-Assistenten erstellen lassen oder ein Textfeld mit der Feldliste einfügen, wird das Eingabeformat des Tabellenfeldes als Einstellung für das Eingabeformat des eingefügten Textfeldes übernommen. Sie können dies aber nachträglich ändern oder löschen. Falls Sie ein Textfeld direkt mit der Toolbox erstellen und dann als dessen *Steuerelementinhalt* ein Tabellenfeld mit einem Eingabeformat angeben, wird dieses Eingabeformat nicht automatisch übernommen. Falls Sie im Textfeld ein Eingabeformat angeben, das von demjenigen im Tabellenfeld abweicht, wirkt im Formular nur das Eingabeformat des Textfeldes und das Eingabeformat des Tabellenfeldes ist insoweit außer Kraft gesetzt.

Der Eingabeformat-Assistent

Access bietet mehrere vordefinierte Eingabeformate an, allerdings nur für Felder der Datentypen *Text* und *Datum/Uhrzeit*. Diese können Sie sich vom Eingabeformat-Assistenten einfügen lassen. Gehen Sie dazu folgendermaßen vor:

- Setzen Sie den Cursor in das Bearbeitungsfeld des zu definierenden Eingabeformats.
- Rufen Sie den Eingabeformat-Assistenten auf, indem Sie auf die Symbol-Schaltfläche mit den drei Punkten am rechten Rand des Bearbeitungsfeldes klicken, vgl. das folgende Bild.

Bild 8.15: Aufrufen des Eingabeformatassistenten

Falls Sie seit der letzten Speicherung des Tabellenentwurfs Änderungen daran vorgenommen haben, werden Sie aufgefordert zu speichern.

- Markieren Sie im ersten Dialogfeld des Eingabeformat-Assistenten (vgl. Bild 8.16) in der Liste der angebotenen Formate ein passendes, und testen Sie ggf. im Eingabefeld *Test*, ob Sie mit der Wirkung des gewählten Formats zufrieden sind.

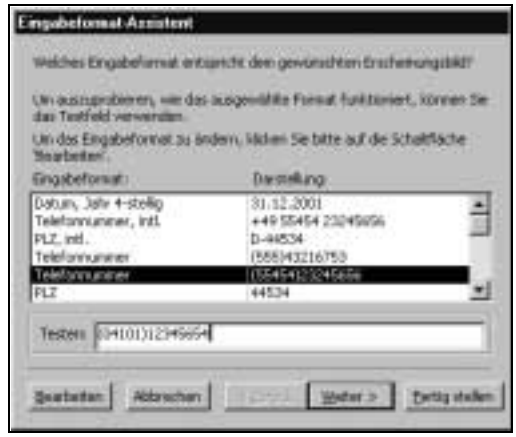


Bild 8.16: Erstes Dialogfeld des Eingabeformat-Assistenten.

- Wenn Sie das markierte Format übernehmen wollen, klicken Sie auf eine der Schaltflächen *Weiter* > oder *Fertigstellen*. Im letzteren Falle wird das Format unmittelbar eingefügt.
- Wenn Sie die Schaltfläche *Weiter* > gewählt haben, zeigt der Eingabeformat-Assistent sein zweites Dialogfeld (vgl. Bild 8.17). Im Bearbeitungsfeld *Eingabeformat* wird das im ersten Dialogfeld gewählte Eingabeformat in symbolischer Form wiedergegeben. Diese Form läßt sich bearbeiten. Falls Sie beispielsweise das Klammerpaar im Format von Bild 8.17 nicht wünschen, könnten Sie die beiden Klammer-Symbole löschen. Darüber hinaus bietet dieses Dialogfeld die Möglichkeit, aus der Dropdown-Liste *Platzhalterzeichen* ein solches zu wählen. Im Beispiel von Bild 8.17 wurde das Unterstreichungszeichen gewählt, das im Feld *Testen* entsprechend dargestellt wird.

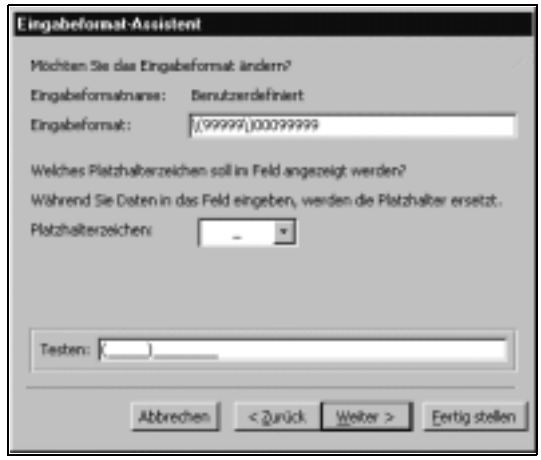


Bild 8.17: Zweites Dialogfeld des Eingabeformat-Assistenten

- Wenn Sie wiederum die Schaltfläche *Weiter >* wählen, zeigt der Eingabeformat-Assistent sein drittes Dialogfeld (hier nicht abgebildet), in dem gefragt wird, ob die Daten zusammen mit den Symbolen oder ohne diese in der Tabelle gespeichert werden sollen.

Die Liste der vom Eingabeformat-Assistenten angebotenen Formate ist nicht sehr umfassend. Daher werden Sie im allgemeinen doch Ihre eigenen Formate schreiben müssen. Allerdings kann es eine erfolgreiche Strategie sein, sich ein Grundformat vom Assistenten einfügen zu lassen und dieses dann anschließend zu überarbeiten.

8.5.4 Standardwert

Die Angabe eines Standardwertes empfiehlt sich, wenn ein Feld oft denselben Wert (oder einen Wert nach derselben Regel) besitzen soll. Wenn Sie beispielsweise in einer Tabelle mit Personendaten ein Feld für den Wohnort vorsehen und erwarten, daß 80% der Personen aus Salzburg kommen, sollten Sie als Standardwert für dieses Feld *Salzburg* angeben. Statt einer Textkonstante wie Salzburg oder einer Zahlenkonstante, z.B. 1, können Sie auch einen Ausdruck angeben, der den gewünschten Standardwert ergibt. So ergibt der Ausdruck

=Datum()

das aktuelle Datum, welches von der PC-Uhr ermittelt wird. Sie können diesen Ausdruck (mit führendem Gleichheitszeichen) als Standardwert angeben. Der als Standardwert verwendete Ausdruck kann auch wesentlich komplexer sein als der gerade angeführte. Falls Sie in Access Basic eine benutzerdefinierte Funktion geschrieben haben, können Sie auch diese in dem Ausdruck anführen. Zum Umgang mit Access-Ausdrücken vgl. genauer Kap. 39, *Ausdrücke, Funktionen und Operatoren*.

8.5.5 Dateneingabe erzwingen

Um eine Dateneingabe zu erzwingen, stellen Sie die Eigenschaft *Eingabe erforderlich* auf den Wert *Ja* ein. Dann speichert Access den Datensatz nur, wenn Sie einen Wert in das betreffende Feld eingegeben haben. Im anderen Falle erhält ein leeres Feld den Wert NULL, einen speziellen Wert für einen fehlenden Wert.

Die Eigenschaft *Eingabe erforderlich* ist in Steuerelementen von Formularen nicht verfügbar. Dies ist jedoch unerheblich, weil Access die entsprechende Prüfung stets vornimmt, gleichgültig, von welcher Stelle aus die Daten eingegeben werden, denn die Anweisung, keine Nullwerte zuzulassen, wird von der Datenbankmaschine *JetEngine* selbst verwaltet – wie es sich für ein ordentliches RDBMS gehört.

8.5.6 Leere Zeichenfolgen

Wenn Sie die Eigenschaft *Eingabe erforderlich* auf *Ja* einstellen, verhindern Sie, daß in dem Feld Nullwerte gespeichert werden. Von einem Nullwert zu unterscheiden ist eine leere Zeichenfolge, die als zwei Anführungszeichen ohne Zeichen dazwischen (»«) angegeben wird. Mit der für die Datentypen *Text* und *Memo* verfügbaren Eigenschaft *Leere Zeichenfolge* legen Sie fest, ob diese angenommen und gespeichert werden soll oder nicht. Obwohl ein Datensatzfeld mit einer leeren Zeichenfolge wie auch mit einem Nullwert leer ist (beide erscheinen auch in der Anzeige mit Standardformat leer), unterscheiden sich beide Werte dennoch: In Ausdrücken und Abfragen werden Nullwerte anders behandelt als leere Zeichenfolgen. Wenn Sie beispielsweise eine Abfrage mit verknüpften Tabellen erstellen, werden Datensätze mit Nullwerten in den Verknüpfungsfeldern nicht in das Abfrageergebnis aufgenommen, solche mit leeren Zeichenfolgen werden dagegen berücksichtigt. Auch wenn Sie für ein Feld einer Abfrage ein Kriterium angeben, können sich Nullwerte und leere Zeichenfolgen in den Datensätzen dieses Feldes unterschiedlich auswirken: Wenn Sie beispielsweise als Kriterium für ein Feld den Ausdruck

```
Wie "*"
```

angeben, werden die Datensätze mit Nullwerten in den Feldern im Abfrageergebnis nicht berücksichtigt, diejenigen mit leerer Zeichenfolge dagegen wohl.

Wenn Sie sowohl Nullwerte wie auch leere Zeichenfolgen zulassen (dann gilt: *Eingabe erforderlich* = *Nein*; *Leere Zeichenfolge* = *Ja*), können Sie zwischen zwei leeren Werten unterscheiden. Dies kann beispielsweise sinnvoll sein, um zwischen unbekannten und (noch) nicht verfügbaren Werten zu unterscheiden. Andererseits kann es auch zweckmäßig sein, die Eigenschaft *Eingabe erforderlich* auf *Ja* einzustellen, um den Bearbeiter zur bewußten Entscheidung zu zwingen. Dann werden keine Nullwerte (d.h. das bloße Drücken der Eingabetaste) zugelassen. Um dennoch eine angemessene Reak-

tion für den Fall nicht verfügbarer Information vorzusehen, könnten Sie dann gleichzeitig die Eigenschaft *Leere Zeichenfolge* auf den Wert *Ja* einstellen. Diese Kombination hätte auch den Vorteil, daß die entsprechenden Datensätze insoweit in Mehrtabellenabfragen berücksichtigt würden. Beachten Sie andererseits, daß Sie Nullwerte mit einem berechneten Feld einer Abfrage jederzeit in eine leere Zeichenfolge umwandeln können. Vgl. dazu u.a. Kap. 11, Auswahlabfragen, Punkt 11.3, *Kriterien*, sowie Kap. 12, *Komplexe Abfragen*, Punkt 12.2, *Abfragen mit Kriterien aus Steuerelementen in einem Formular (Query-by-Form)*.

Standardmäßig werden Nullwerte und leere Zeichenfolgen gleichermaßen als leere Felder angezeigt. Mit geeigneten Formatangaben für die Eigenschaft *Format* (vgl. oben, Punkt 8.5.2, *Anzeigeformat*) können Sie jedoch in der Anzeige zwischen beiden Werten unterscheiden. Mit dem Format

@;"Nicht verfügbar"[Rot];"Unbekannt«

beispielsweise erreichen Sie eine Anzeige, wie dies in Bild 8.18 dargestellt ist: Leere Zeichenfolgen werden in roter Schriftfarbe als »Nicht verfügbar« wiedergegeben, Nullwerte dagegen als »Unbekannt«.

Tabelle4 : Tabelle

Nachname	Vorname	Beruf
Meier	Hans	Maurer
Müller	Harry	Unbekannt
Schulze	Otto	Zimmermann
Grass	Paula	Nicht verfügbar
Eichbach	Emily	Malermeisterin
Schurping	Nicht verfügbar	Programmierer
Liberman	Nicht verfügbar	Verkäuferin
Gollach	Sigfried	Unbekannt
Gabriel	Ludwig	Nicht verfügbar
Petersen	Ulla	Unbekannt
Hansen	Silke	Unbekannt
	Unbekannt	Unbekannt

Datensatz: 1 von 11

Bild 8.18: Unterschiedliche Darstellung von leeren Zeichenfolgen und Nullwerten

8.5.7 Gültigkeitsregel

Mit einer Gültigkeitsregel stellen Sie sicher, daß in ein Feld nur Werte eingegeben werden können, die den Bedingungen der Gültigkeitsregel entsprechen. Eine Gültigkeitsregel wird immer als Ausdruck eingegeben. Der Ausdruck gibt den Wertebereich an, der für das Feld zulässig sein soll. Oft beginnt ein Ausdruck für die Gültigkeitsregel mit einem der Vergleichsoperatoren. Beispielsweise gibt der Ausdruck >=200 an, daß in das Feld nur Zahlen eingegeben werden können, die größer als oder gleich 200 sind. Die folgende Übersicht zeigt Ihnen ein paar Beispiele für typische Gültigkeitsausdrücke:

Gültigkeitsregel	Bedeutung
>=0	Wert muß größer als oder gleich Null sein.
>12 Und <44	Wert muß zwischen 12 und 44 liegen, die Grenzen ausgeschlossen.
>=12 Und <=44	Wert muß zwischen 12 und 44 liegen, die Grenzen eingeschlossen.
>Datum	Datum muß mindestens einen Tag hinter dem aktuellen Tagesdatum liegen.
>=Datum()-Tag(Datum())+1	Datum muß nach dem 1. des laufenden Monats liegen oder diesem gleich sein.
>=DatSeriell(Jahr(Datum());1;1) Und <DatSeriell(Jahr(Datum())+1;1;1)	Datum muß ein Wert des aktuellen Jahres sein.
Wie »M*"	Der Text muß mit einem M beginnen, Art und Menge der weiteren Zeichen sind beliebig.
Wie »M??"	Der Text muß mit einem M beginnen und zwei weitere beliebige Zeichen umfassen.

Tabelle 8.3: Beispiele für Gültigkeitsregeln eines Feldes

In einer Gültigkeitsregel für ein Tabellenfeld können Sie sich nicht auf ein anderes Feld beziehen. Wenn Sie die Gültigkeit eines Wertes in Abhängigkeit von einem oder mehreren anderen Feldern derselben Tabelle überprüfen wollen, müssen Sie dies in Form einer Gültigkeitsregel *für die Tabelle* formulieren, vgl. unten Punkt 8.8, *Tabelleneigenschaften festlegen*. Wenn Sie die Gültigkeit eines Wertes für ein Feld vom Wert eines Feldes einer anderen Tabelle (und damit eines anderen Datensatzes) abhängig überprüfen wollen, können Sie diese Aufgabe nur mit einem Makro oder einer VBA-Prozedur lösen.

Wirkung von Gültigkeitsregeln in Tabellen gegenüber Gültigkeitsregeln in Formularen

Der Formular-Assistent übernimmt Gültigkeitsregeln, soweit sie in der Tabelle formuliert sind, in das Formular. Entsprechendes gilt, wenn Sie ein Steuerelement mit Hilfe der Feldliste in ein Formular einfügen. Falls Sie aber eine dieser Regeln ändern oder ein Steuerelement auf andere Weise (z.B. mit der Toolbox) einfügen und keine oder eine andere Gültigkeitsregel formulieren, gilt bei Dateneingabe über das Formular stets der folgende Zusammenhang: Die Gültigkeitsregel eines Feldes wirkt auch dann, wenn diese im entsprechenden Steuerelement eines Formulars nicht angegeben ist. Umgekehrt gilt: Eine Gültigkeitsregel im Steuerelement eines Formulars, die von der

Gültigkeitsregel in dem Feld abweicht, an welches das Steuerelement gebunden ist, wirkt auch für sich allein, so daß nur ein Wert eingegeben werden kann, der jeder der beiden Regeln entspricht. Anders ausgedrückt: Die beiden Gültigkeitsregeln in der Tabelle und im Formular wirken wie durch das logische *Und* verbunden.

Verändern der Gültigkeitsregel

Eine Gültigkeitsregel ist nur bei der Eingabe eines neuen oder Bearbeitung eines bestehenden Wertes wirksam. Dies bedeutet umgekehrt, daß eine nachträglich formulierte oder geänderte Gültigkeitsregel bereits eingegebene Feldwerte, auch wenn sie gegen die neue Regel verstoßen, unbeanstandet läßt.

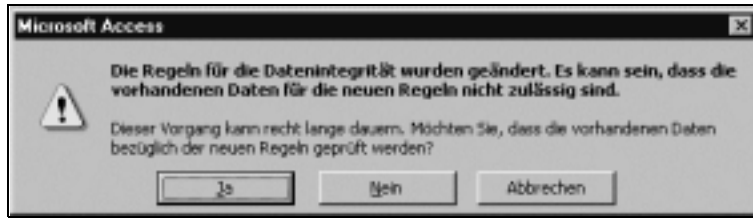


Bild 8.19: Diese Meldung erscheint, wenn Sie eine bestehende Gültigkeitsregel für ein Feld verändert (oder eine solche neu formuliert) haben und die Tabelle speichern wollen.

Allerdings bietet Access an, bei geänderter Gültigkeitsregel die bereits für das betreffende Feld vorhandenen Daten »bezüglich der neuen Regeln« zu prüfen, vgl. Bild 8.19. Falls Sie von diesem Angebot Gebrauch machen und Access Verstöße gegen die neue Gültigkeitsregel feststellt, wird dies in einem weiteren Dialogfeld mitgeteilt (vgl. Bild 8.20) und Sie können dann entscheiden, die neue Gültigkeitsregel anzunehmen oder zu verwerfen.

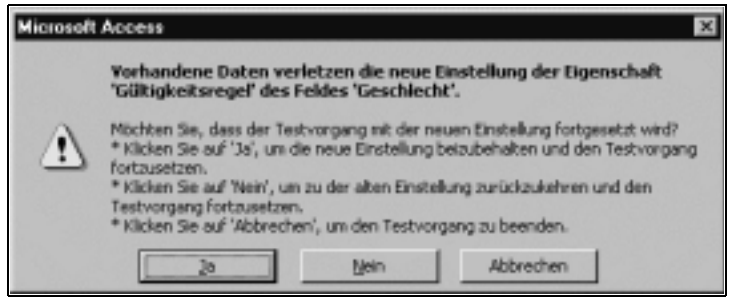


Bild 8.20: Nachdem Verstöße vorhandener Daten gegen eine veränderte Gültigkeitsregel entdeckt wurden, können Sie in der angegebenen Weise reagieren.

8.5.8 Gültigkeitsmeldung

Wenn bei der Dateneingabe in die Tabelle gegen eine festgelegte Gültigkeitsregel verstoßen wird, blendet Access einen Standardhinweis ein, vgl. Bild 8.21.

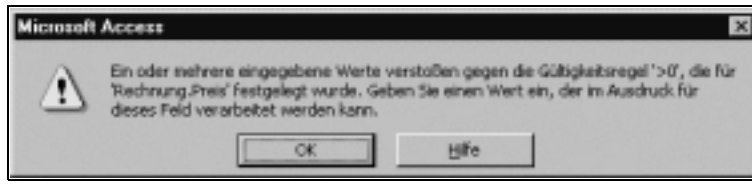


Bild 8.21: Standardmeldung bei Verstoß der Dateneingabe gegen eine Gültigkeitsregel

Der Text dieser Meldung wird durch den Text ersetzt, den Sie ggf. als Gültigkeitsmeldung angeben. Geben Sie Texte im Eigenschaftenfeld für die Gültigkeitsmeldung ohne Anführungszeichen ein, es sei denn, diese sollen in der Meldung ebenfalls erscheinen. Achten Sie inhaltlich darauf, daß dem Benutzer nicht nur mitgeteilt wird, gegen welche Regel er verstoßen hat, sondern auch, wie er sich für eine fehlerfreie Eingabe verhalten muß. Wenn Sie beispielsweise die Gültigkeitsregel >0 für ein Feld, in das ein Preis eingegeben wird, definiert haben, könnte die Gültigkeitsmeldung lauten:

> »Der von Ihnen eingegebene Wert verstößt gegen die Geschäftsregel: »Keine negativen Preise!«. Geben Sie bitte einen Preis ein, der größer als 0 ist. Falls Sie nicht weiter wissen, drücken Sie nach Bestätigung dieses Dialogfeldes die ESC-Taste.«

Dieser Text würde statt der in Bild 8.21 gezeigten Standard-Gültigkeitsmeldung angezeigt werden, wenn gegen die Gültigkeitsregel verstoßen wird, vgl. Bild 8.22.

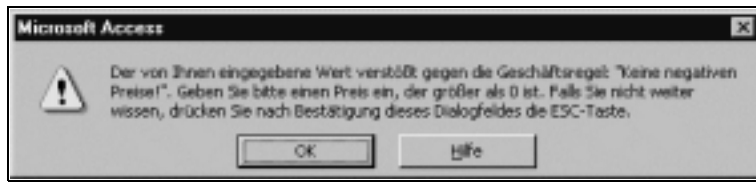


Bild 8.22: Benutzerdefinierte Gültigkeitsmeldung wird nach Verstoß gegen die entsprechende Gültigkeitsregel angezeigt.

8.5.9 Beschriftung

Falls Sie für die Feldeigenschaft *Beschriftung* keinen Text angeben, wird das Tabellenfeld in der Datenblattansicht mit seinem Feldnamen bezeichnet. Dieser Bezeichnungstext wird durch den Text ersetzt, den Sie für die Eigenschaft *Beschriftung* angeben. Er wird auch als Text für das Bezeichnungsfeld

eines Steuerelements übernommen, das Sie mittels des Werkzeugs *Feldliste* in ein Formular einfügen. Wenn Sie beispielsweise für ein Tabellenfeld mit dem Feldnamen *Preis* die Feldeigenschaft *Beschriftung* auf den Text *Einkaufspreis* einstellen, bekommt dieses Feld in der Datenblattansicht der Tabelle die Spaltenbezeichnung *Einkaufspreis*. Wenn Sie das Feld *Preis* mit der Feldliste in ein Formular einfügen, bekommt das Bezeichnungsfeld des so eingefügten Steuerelements die Beschriftung *Einkaufspreis*.

8.6 Nachschlagefeld erstellen

Für ein Feld mit dem Datentyp *Text* oder *Zahl* können Sie ein *Nachschlagefeld* erstellen. (Formal ist ein Nachschlagefeld auch für ein Feld des Datentyps *Ja/Nein* definierbar, jedoch inhaltlich nicht sinnvoll; für ein derartiges Feld sollten Sie statt dessen ein Kontrollkästchen vorsehen.) Ein Nachschlagefeld funktioniert folgendermaßen: Wenn Sie in der Datenblattansicht der Tabelle einen Wert in das Feld, für das ein Nachschlagefeld festgelegt wurde, eingeben wollen, können Sie das Nachschlagefeld als Dropdown-Liste aufschlagen, aus der Sie dann einen Wert auswählen können. Die Werte, die das Nachschlagefeld anbietet, stammen entweder aus einer zusammen mit der Definition des Nachschlagefeldes eingegebenen Werteliste, einer Tabelle oder einer Abfrage. Der wichtigste Nutzen eines Nachschlagefeldes dürfte darin bestehen, für das Fremdschlüsselfeld einer Detailtabelle eine Liste der möglichen Primärschlüsselwerte der zugehörigen Mastertabelle anzubieten.

Sie können ein Nachschlagefeld selbständig definieren, indem Sie die entsprechenden Eigenschaften einstellen. Access bietet jedoch auch einen Nachschlage-Assistenten an, mit dessen Hilfe Sie ein Nachschlagefeld erstellen können. Dieser Weg soll im folgenden zunächst beschrieben werden. Anschließend wird gezeigt, wie Sie dasselbe Nachschlagefeld, das zuvor vom Nachschlage-Assistenten erstellt wurde, selbständig aufbauen können.

8.6.1 Nachschlagefeld mit dem Nachschlage-Assistenten erstellen

Als Demonstrationsbeispiel dient die Tabelle *Projekte* aus der Beispieldatenbank Projekte.mdb, deren Einzelheiten in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, beschrieben sind und die Sie auf der Begleit-CD-ROM finden können. Die Tabelle *Projekte* ist die Detailtabelle der 1:n-Beziehung zur Mastertabelle *Personal*. Das Primärschlüsselfeld der Tabelle *Personal* hat den Feldnamen *PersonalCode*. Denselben Namen hat auch das Fremdschlüsselfeld in der Tabelle *Projekte*, das diesen Primärschlüssel referenziert. Mit Hilfe des Nachschlage-Assistenten soll für das Feld *PersonalCode* der Tabelle *Projekte* ein Nachschlagefeld erstellt werden, das die Vor- und Nachnamen der Mitarbeiter aus der Tabelle *Personal* zur Auswahl anbietet, in das Feld *PersonalCode* der Tabelle *Projekte* jedoch den zugehörigen Wert des Primärschlüsselfeldes der Tabelle *Personal* einträgt.

Gehen Sie folgendermaßen vor, um das beschriebene Nachschlagefeld mit dem Nachschlage-Assistenten zu erstellen:

Bild 8.23: Aufgeschlagene Liste der Spalte »Felddatentyp« zum Feld »PersonalCode«, deren letzter Eintrag den Nachschlage-Assistenten aufruft

- Öffnen Sie die Tabelle *Projekte* der Datenbank *Projekte.mdb* in der Entwurfsansicht.
- Geben Sie dem Feld *PersonalCode* den Fokus.
- Schlagen Sie die Dropdown-Liste in der Spalte *Felddatentyp* auf, und wählen Sie darin den Eintrag *Nachschlage-Assistent*, vgl. Bild 8.23. Dann wird das erste Dialogfeld des Nachschlage-Assistenten angezeigt, vgl. Bild 8.24.

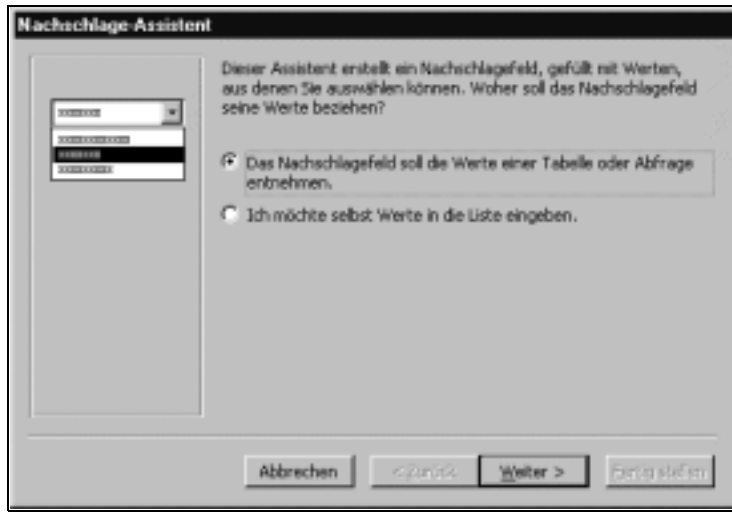


Bild 8.24: Erstes Dialogfeld des Nachschlage-Assistenten

- Da die Werte für das Nachschlagefeld aus der Tabelle *Personal* entnommen werden sollen, muß im ersten Dialogfeld des Nachschlage-Assistenten (vgl. Bild 8.24) die erste Option gewählt bleiben. Bei Wahl der Alternative würden die weiteren Dialogfelder des Nachschlage-Assistenten die Werte für eine starre Werteliste abfragen. Bestätigen Sie mit der Schaltfläche *Weiter* >. Dann wird das zweite Dialogfeld des Nachschlage-Assistenten angezeigt, vgl. Bild 8.25.

Bild 8.25: Zweites Dialogfeld des Nachschlage-Assistenten

- Das zweite Dialogfeld des Nachschlage-Assistenten bietet eine Liste aller in der Datenbank verfügbaren Tabellen und/oder Abfragen (außer der in Arbeit befindlichen Tabelle) an. Aus einer dieser Tabellen oder Abfragen muß das Nachschlagefeld seine Werte beziehen. Im dargestellten Beispiel ist nur die Tabelle *Personal* verfügbar, welche folglich zu markieren und mit der Schaltfläche *Weiter* > zu bestätigen ist. Dann zeigt sich das dritte Dialogfeld des Nachschlage-Assistenten, vgl. Bild 8.26.
- Kopieren Sie im dritten Dialogfeld des Nachschlage-Assistenten (vgl. Bild 8.26) mit Hilfe der Schaltflächen > oder >> aus der Liste der verfügbaren Felder diejenigen, die für das Nachschlagefeld benötigt werden. Dabei sind zwei Punkte zu beachten:

1. Die Werteliste des Nachschlagefeldes kann aus mehr als einer Spalte bestehen; daher kann es sinnvoll sein, mehrere Felder auszuwählen.
2. Obwohl im Nachschlagefeld mehr als ein Tabellenfeld angezeigt werden kann, kann dem Tabellenfeld, für welches das Nachschlagefeld definiert wird (hier: das Feld *PersonalCode* der Tabelle *Projekte*) nur der Wert eines Feldes (einer Spalte des Nachschlagefeldes) übergeben werden. Dieses Feld – hier das Feld *PersonalCode* der Tabelle *Personal* – ist in jedem Falle auszuwählen.

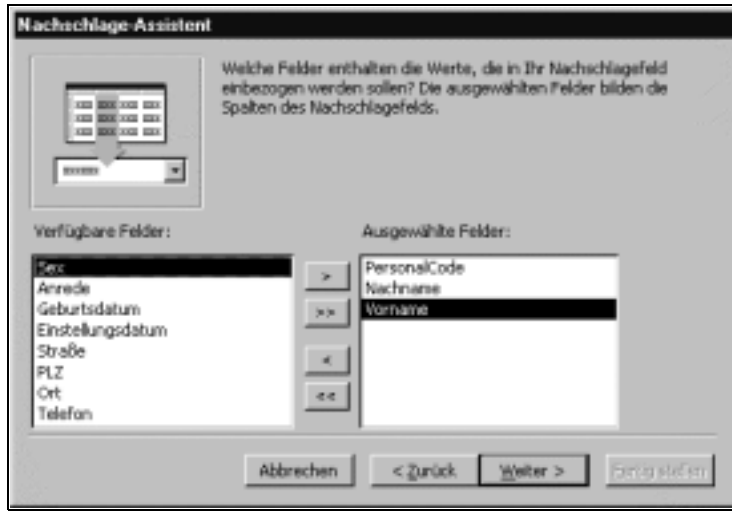


Bild 8.26: Drittes Dialogfeld des Nachschlage-Assistenten

Aus diesen Gründen sollen im Dialogfeld die Felder *PersonalCode*, *Nachname* und *Vorname* in die Liste der ausgewählten Felder übernommen werden. Die Darstellung in Bild 8.26 zeigt diesen Zustand an. Bestätigen Sie das dritte Dialogfeld schließlich mit der Schaltfläche *Weiter* >. Dann zeigt der Nachschlage-Assistent sein viertes Dialogfeld, vgl. Bild 8.27.

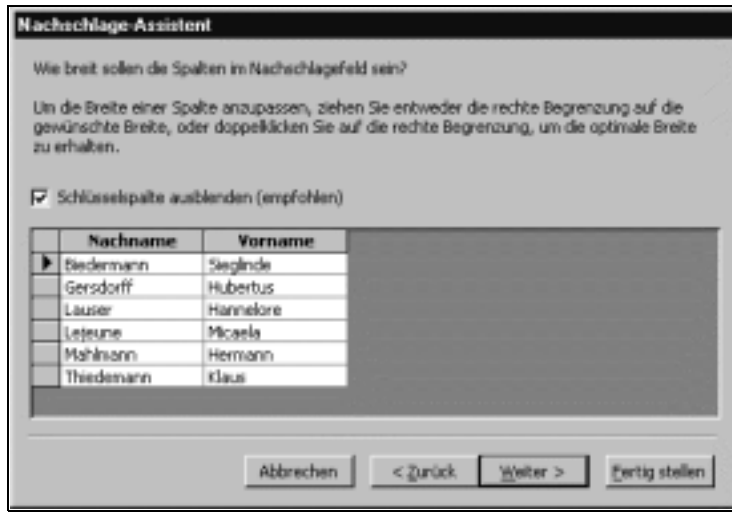


Bild 8.27: Viertes Dialogfeld des Nachschlage-Assistenten

- Im vierten Dialogfeld legen Sie das Layout des Nachschlagefeldes fest, indem Sie die Breiten der Spalten angeben. Wie dies geschieht, sagt der erklärende Text im Dialogfeld deutlich. Der Nachschlage-Assistent hat

erkannt, daß eines der ausgewählten Felder ein Schlüsselfeld ist (genauer: Primärschlüsselfeld). Er schlägt vor, dieses Feld im Nachschlagefeld nicht anzuzeigen. Dies ist sinnvoll, denn die Werte des Primärschlüsselfeldes sind im allgemeinen (wie auch hier) wenig aussagekräftig. Falls auch das Schlüsselfeld angezeigt werden soll, müßte das entsprechende Kontrollkästchen deaktiviert werden. Bestätigen Sie das Dialogfeld mit der Schaltfläche *Weiter >*. Dann wird das letzte Dialogfeld des Nachschlage-Assistenten (hier nicht wiedergegeben) angezeigt, in dem Sie einen Beschriftungstext für das Nachschlagefeld angeben können.

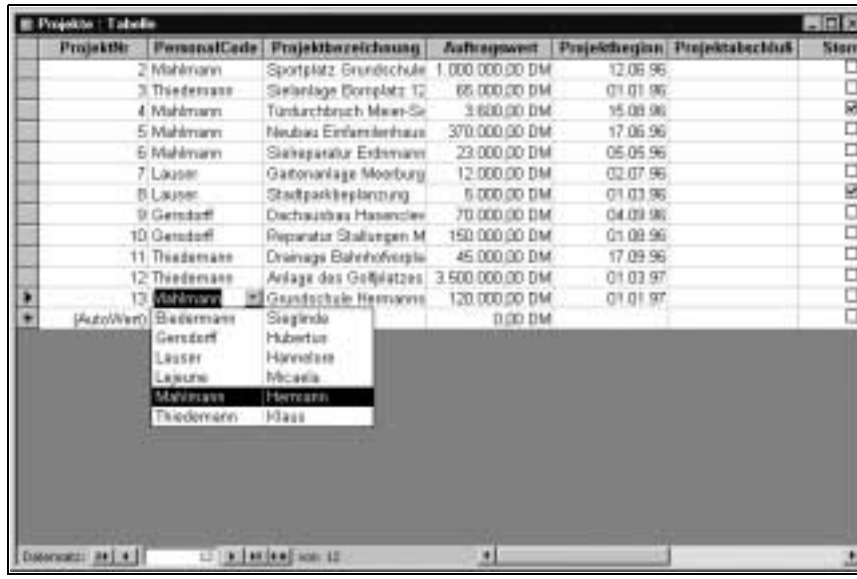


Bild 8.28: Aufgeschlagenes Nachschlagefeld zur Eingabe eines neuen Wertes in das Feld »PersonalCode« der Tabelle »Projekte«

- Bestätigen Sie das letzte Dialogfeld des Nachschlage-Assistenten schließlich mit der Schaltfläche *Fertigstellen*. Sie werden daraufhin (mit einem etwas unpassenden Aufforderungstext) zum Speichern der Tabelle aufgefordert. Wenn Sie dieses bestätigen, ist die Arbeit des Nachschlage-Assistenten beendet.

Wenn die Arbeit des Nachschlage-Assistenten erfolgreich war, steht in der Datenblattansicht der Tabelle *Projekte* für die Spalte *PersonalCode* ein Nachschlagefeld, das die definierte Werteliste anbietet, zur Verfügung (vgl. Bild 8.28). In der Entwurfsansicht der Tabelle *Projekte* können Sie die Eigenschaften-Einstellungen einsehen und ggf. bearbeiten, die der Nachschlage-Assistent für das Feld *PersonalCode* festgelegt hat. Der entsprechende Bildausschnitt ist in Bild 8.29 wiedergegeben.

Steuerelement anzeigen	Kombinationsfeld
Herkunftstyp	Tabelle/Abfrage
Datensatzherkunft	SELECT [Personal].[PersonalCode],
Gebundene Spalte	1
Spaltenanzahl	3
Spaltenüberschriften	Nein
Spaltenbreiten	0cm;2,54cm;2,54cm
Zeilenanzahl	8
Listenbreite	5,079cm
Nur Listeneinträge	Ja

Bild 8.29: Eigenschaften für das Nachschlagefeld, das für das Tabellenfeld »PersonalCode« der Tabelle »Projekte« mit dem Nachschlage-Assistenten definiert wurde.

Sie können die Einstellungen, die der Nachschlage-Assistent für das Nachschlagefeld auf Basis der Antworten auf die Dialogfeld-Fragen vorgenommen hat, ändern, indem Sie die Eigenschaften des Nachschlagefeldes bearbeiten. Diese werden im einzelnen im folgenden Punkt 8.6.2, *Nachschlagefeld selbständig erstellen*, erklärt. An dieser Stelle sei lediglich auf die Einstellung hingewiesen, die der Nachschlage-Assistent für die Eigenschaft *Datensatzherkunft* vorgenommen hat. Statt den Namen der Tabelle *Personal* anzugeben, was auch möglich gewesen wäre, wurde als Datensatzherkunft eine Abfrage in Form einer SQL-Anweisung angegeben. Ihr Text, der in Bild 8.29 nur unvollständig zu lesen ist, lautet vollständig:

```
SELECT [Personal].[PersonalCode], [Personal].[Nachname],
[Personal].[Vorname] FROM Personal;
```

Diese Form, die Datensatzherkunft anzugeben, hat gegenüber der Angabe eines Tabellennamens den Vorteil wesentlich größerer Flexibilität, weil genau die Felder in genau der Reihenfolge angegeben werden können, wie diese benötigt werden. Andererseits setzt die Bearbeitung einer SQL-Anweisung durch den Benutzer gewisse Minimalkenntnisse der Abfragesprache SQL voraus. In Kap. 14, *Einführung in SQL*, können Sie sich in diesem Buch grundlegend über den passiven und aktiven Gebrauch von SQL informieren.

8.6.2 Nachschlagefeld selbständig erstellen

Ein Nachschlagefeld ist ein Listen- oder Kombinationsfeld. Diese beiden Steuerelemente sind sich sehr ähnlich und werden in ihren Eigenschaften praktisch gleichartig festgelegt. Listen- und Kombinationsfelder als Steuerelemente werden an anderer Stelle ausführlich behandelt, vgl. Kap. 16, *Standard-Steuerelemente in Formularen*, Punkt *Listenfeld und Kombinationsfeld*. Daher sollen an dieser Stelle nur die Punkte angesprochen werden, die wesentlich für die Erstellung eines Nachschlagefeldes sind. Als Beispiel dient wiederum, wie bei der Arbeit mit dem Nachschlage-Assistenten, das Feld *PersonalCode* der Tabelle *Projekte* aus der Datenbank *Projekte.mdb*. Um

für dieses Feld ein Nachschlagefeld selbständig ohne den Nachschlage-Assistenten zu erstellen (bzw. ein vom Nachschlage-Assistenten erstelltes nachträglich zu bearbeiten), gehen Sie folgendermaßen vor:

- Öffnen Sie die Tabelle *Projekte* der Datenbank *Projekte.mdb* in der Entwurfsansicht.
- Geben Sie dem Feld *PersonalCode* den Fokus.
- Aktivieren Sie im Fensterteil *Feldeigenschaften* die Registerkarte *Nachschlagen*.
- Wählen Sie in der Dropdown-Liste zur Eigenschaft *Steuerelement anzeigen* den Eintrag *Kombinationsfeld* oder *Listenfeld*. Dann werden die Eigenschaften mit ihren Standardwerten angezeigt, wie diese in Bild 8.30wiedergegeben sind.

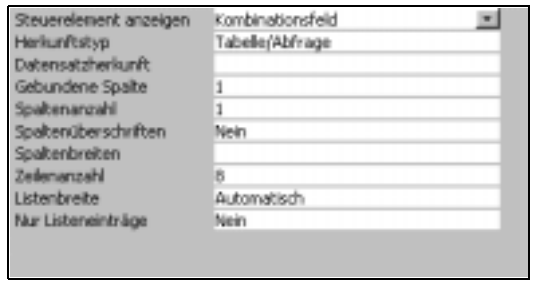


Bild 8.30: Eigenschaften für ein Nachschlagefeld, bevor diese bearbeitet wurden

Bearbeiten Sie die einzelnen Eigenschaften in der folgenden Weise:

Steuerelement anzeigen. Für ein Nachschlagefeld können Sie eines der beiden Steuerelemente *Kombinationsfeld* (Dropdown-Liste) oder *Listenfeld* wählen. Für die Datenblattansicht wirkt sich der Unterschied dieser beiden Steuerelemente nicht aus, weil das Nachschlagefeld dort stets nur als Dropdown-Liste angezeigt wird. Das Listenfeld verschafft sich Geltung, wenn Sie das Feld, für das Sie ein Nachschlagefeld als Listenfeld erstellt haben, mit dem Werkzeug *Feldliste* in die Entwurfsansicht eines Formulars einfügen.

Herkunftstyp. Sie können mittels einer Dropdown-Liste zwischen *Tabelle/Abfrage*, *Wertliste* und *Feldliste* wählen. Wenn das Nachschlagefeld dazu dienen soll, eine oder mehrere Feldwerte einer Tabelle oder Abfrage wiederzugeben, müssen Sie *Tabelle/Abfrage* wählen. Wenn Sie *Wertliste* wählen, müssen Sie unter *Datensatzherkunft* (vgl. die folgende Eigenschaft) eine Liste von Werten eingeben. *Feldliste* wählen Sie, wenn das Nachschlagefeld die Feldnamen einer Tabelle oder Abfrage anzeigen soll. Für unser Beispiel wählen Sie *Tabelle/Abfrage*.

Datensatzherkunft. Wenn die Eigenschaft *Herkunftstyp* auf *Wertliste* eingestellt ist, geben Sie hier die Wertliste an. Dabei werden die einzelnen Werte durch je ein Semikolon gegeneinander getrennt. Wenn das Nachschlagefeld

beispielsweise eine Liste mit den Einträgen *Sehr geehrter Herr*, *Sehr geehrte Frau* und *Sehr geehrte Damen und Herren* anzeigen soll, müssen Sie die Wertliste in der Form

```
Sehr geehrter Herr;Sehr geehrte Frau;Sehr geehrte Damen und Herren
```

angeben.

Wenn die Eigenschaft *Herkunftstyp* auf *Tabelle/Abfrage* oder *Feldliste* eingestellt ist, geben Sie hier den Namen der Tabelle oder Abfrage an, welche die Felder, deren Werte oder Namen im Nachschlagefeld angezeigt werden sollen, enthält. Sie können die Datenquelle aber auch als SQL-Anweisung, beginnend mit dem Schlüsselwort SELECT, angeben. Ein Beispiel für diese Art, die Datensatzherkunft anzugeben, finden Sie oben am Ende des vorangehenden Punktes 8.6.1, *Nachschlagefeld mit dem Nachschlage-Assistenten* erstellen. Für unser Beispiel wählen Sie aus der Dropdown-Liste die Tabelle *Personal*.

Gebundene Spalte. Das Nachschlagefeld kann, wie jedes Listen- oder Kombinationsfeld, mehrere Spalten als Liste anbieten, jedoch nur den Wert *einer* Spalte an das Tabellenfeld weitergeben. Diese Spalte wird als *Gebundene Spalte* bezeichnet. Sie wird mittels der Position in der Spaltenreihenfolge identifiziert. Die Reihenfolge der Spalten bzw. Felder ist durch die Angabe zu *Datensatzherkunft* bestimmt: Wenn Sie dort den Namen einer Tabelle oder Abfrage angegeben haben, wird die Reihenfolge der Spalten durch die Reihenfolge der Felder in der Tabelle/Abfrage bestimmt. Wenn eine SQL-SELECT-Anweisung angegeben ist, bestimmt deren Feldreihenfolge die Spaltenreihenfolge des Nachschlagefeldes. Wenn Sie beispielsweise für *Gebundene Spalte* die Zahl 3 angeben, wird der Wert der dritten Spalte des Nachschlagefeldes weitergegeben. Geben Sie für unser Beispiel die Zahl 1 an, weil *PersonalCode* die erste Spalte ist und deren Wert weitergegeben werden soll.

Spaltenanzahl. Geben Sie an, wie viele Spalten der Datenquelle im Nachschlagefeld berücksichtigt werden sollen. Wenn Sie beispielsweise die Zahl 4 angeben, werden im Nachschlagefeld die ersten vier Felder der Datenquelle als vier Spalten angezeigt. Da Sie mit *Spaltenanzahl* stets die *ersten* n Felder der Datenquelle bestimmen und daher nicht selektiv einzelne als Mehrfachauswahl identifizieren können, scheint sich ein Problem zu ergeben, wenn Sie tatsächlich einzelne, nicht zusammen liegende Felder darstellen wollen. Sie lösen dies Problem, indem Sie die Breiten der Spalten, die Sie nicht im Nachschlagefeld anzeigen lassen wollen, auf den Wert 0 setzen, vgl. dazu etwas weiter unten die Eigenschaft *Spaltenbreiten*. Geben Sie für unser Beispiel für *Spaltenanzahl* die Zahl 3 an, damit die Felder *PersonalCode*, *Nachname* und *Vorname* als Spalten im Nachschlagefeld berücksichtigt werden.

Spaltenüberschriften. Wählen Sie den Wert *Ja*, wenn die Feldnamen als Spaltenüberschriften im Nachschlagefeld angezeigt werden sollen, andernfalls *Nein*.

Spaltenbreiten. Sie können diese Eigenschaft unausgefüllt lassen. Dann werden die Breiten der einzelnen Spalten automatisch von Access eingestellt. Andererseits lassen sich die Spaltenbreiten über diese Eigenschaft sehr gezielt steuern, weil Sie beispielsweise für einzelne Spalten auch die Breite von 0 cm angeben können, wodurch deren Anzeige unterdrückt wird. Dies empfiehlt sich in unserem Beispiel für die erste Spalte *PersonalCode*. Geben Sie z.B.

0cm;3cm;3cm

an (zulässig ist auch der Eintrag *0;3;3*, den Access dann in *0cm;3cm;3cm* umwandelt). Dann wird die erste Spalte, die für das Feld *PersonalCode* steht, im Nachschlagefeld nicht angezeigt, und die beiden anderen Spalten *Nachname* und *Vorname* werden jeweils mit einer Breite von 3 cm angezeigt. Beachten Sie, daß der Wert der Spalte *PersonalCode* weitergegeben wird, weil diese als gebundene Spalte angegeben ist; der Umstand, daß sie nicht angezeigt wird, ändert nichts daran. Diese Funktionsweise ist gerade im Zusammenhang mit einem Primärschlüsselfeld, das vom Fremdschlüsselfeld der Detailtabelle referenziert wird, besonders vorteilhaft, weil dadurch im Nachschlagefeld die Spalten angezeigt werden können, deren Inhalte vom Benutzer sinnvoll gedeutet werden können, während der Inhalt des Primärschlüsselfeldes, der oft eine sprachlich kaum deutbare Zahl oder ein Code ist, zwar in der Anzeige unterdrückt, aber an das Fremdschlüsselfeld der Detailtabelle weitergegeben wird.

Zeilenanzahl. Mit dieser Eigenschaft geben Sie an, wie viele Zeilen die Dropdown-Liste des Nachschlagefeldes anzeigen soll. Ändern Sie den Standardwert von 8 ggf. auf eine Ihren Vorstellungen entsprechende Zahl.

Listenbreite. Der Wert dieser Eigenschaft bestimmt, wie breit der Listenteil der Dropdown-Liste ist. Er kann breiter, aber nicht schmaler als die Dropdown-Liste selbst sein. In der Datenblattansicht ist die Breite der Dropdown-Liste identisch mit der Breite der Tabellenspalte. Daher kann der Listenteil der (aufgeschlagenen) Dropdown-Liste zwar breiter sein als die Tabellenspalte, jedoch nicht schmaler. Wenn Sie eine Liste mit mehreren Spalten anzeigen möchten, geben Sie einen entsprechend großen Wert an, damit alle Spalten im Listenfeld angezeigt werden können. Für unser Beispiel empfiehlt es sich, die Einstellung von *Listenbreite* auf *Automatisch* zu belassen.

Nur Listeneinträge. Diese Eigenschaft hat eine beachtliche inhaltliche Bedeutung: Wenn sie auf *Ja* eingestellt ist, können für das Tabellenfeld, für welches das Nachschlagefeld definiert ist, nur Werte aus der Liste eingegeben werden, andernfalls erscheint die Meldung, die in Bild 8.31 zu sehen ist.

Für unser Beispiel erscheint es sinnvoll, die Eigenschaft *Nur Listeneinträge* auf Ja einzustellen, weil andere als Primärschlüsselwerte der Tabelle *Personal* für den Fremdschlüssel der Tabelle *Projekte* nicht zulässig sind. (Dar-

über hinaus werden die Werte für den Fremdschlüssel auch von der JetEngine überprüft, weil für die 1:n-Beziehung der Tabellen *Personal* und *Projekte* referentielle Integrität vereinbart ist.)

Bild 8.31: Wenn die Eigenschaft »Nur Listeneinträge« für ein Nachschlagefeld auf »Ja« eingestellt ist, erscheint diese Meldung, falls in das Tabellenfeld ein Wert eingegeben wird, der keinem Listeneintrag entspricht.

8.7 Benutzerdefinierte Anzeigeformate

Tabellenfelder und das Steuerelement *Textfeld* eines Formulars weisen die Eigenschaft *Format* auf, mit denen Sie die Anzeige und das Druckbild der Werte bestimmen können. Die Eigenschaft legt die Darstellung von Datums- und Zeitangaben, Ja/Nein-Werten, Texten sowie Zahlen- und Währungswerten fest.

Die möglichen Formateinstellungen für die verschiedenen Datentypen sind unterschiedlich. Daher werden sie im folgenden im einzelnen unter getrennten Überschriften *Datum/Uhrzeit*, *Ja/Nein*, *Text und Memo* sowie *Zahl und Währung* behandelt.

8.7.1 Allgemeine Hinweise

Standardformate

Für die Felddatentypen *Datum/Uhrzeit*, *Ja/Nein* sowie *Zahl* und *Währung* benutzt Access Standardformate, wenn Sie dafür kein benutzerdefiniertes Format bestimmen. Die Wirkung verschiedener Standardformate hängt davon ab, welche Einstellungen Sie in der Windows-Systemsteuerung vorgenommen haben. Wenn dort beispielsweise ein vorangestelltes DM-Zeichen für die Standardwährung angegeben ist, erscheint auch im Access-Standard-Währungsformat ein vorangestelltes DM-Zeichen. Ähnlich ist dort auch die Stellung des Minuszeichens etc. geregelt. Nehmen Sie ggf. Änderungen in der Windows-Systemsteuerung, Ländereinstellungen, vor.

Benutzerdefinierte Formate

Wenn Sie ein benutzerdefiniertes Format erstellen, sollten Sie die folgenden Zusammenhänge beachten:

Zahlenformate. In Formaten für Zahlen können Sie bis zu vier Formatabschnitte vorsehen, die die Formatierung der folgenden Zahlenbereiche regelt:

Formatabschnitt	Zahlenbereich
Erster	Positive Zahlen
Zweiter	Negative Zahlen
Dritter	Zahl 0
Vierter	"NULL" oder "Leer"

Tabelle 8.4: Formatabschnitte für Zahlenformate

Sie brauchen nur den ersten Formatabschnitt anzugeben. Dann werden alle Zahlen so, wie darin bestimmt, formatiert. Wenn Sie jedoch mehrere Formatabschnitte angeben, müssen Sie diese gegeneinander durch je ein Semikolon trennen. Zahlenbereiche, für die kein Formatabschnitt vorgesehen ist, werden mit den Angaben des ersten Formatabschnitts formatiert. Das folgende benutzerdefinierte Format

```
#.##0,00" DM";-#.##0,00" DM"[Rot]
```

bestimmt z.B., daß positive Zahlen mit einem Tausenderteilungspunkt, zwei Dezimalstellen und nachgestelltem DM-Zeichen ausgewiesen werden. Negative Zahlen werden genauso wiedergegeben, jedoch zusätzlich mit roter Schrift und vorangestelltem Minuszeichen. Die Zahl 0 wird, da ein dritter Formatabschnitt nicht angegeben wurde, entsprechend dem ersten Formatabschnitt, also in schwarzer Schrift als 0,00 DM, wiedergegeben.

Textformate. In Formaten für Text- und Memofelder können Sie bis zu drei Formatabschnitte vorsehen, die die Formatierung in folgender Weise regeln:

Formatabschnitt	Feldinhalt
Erster	Felder mit Text
Zweiter	Felder mit einer leeren Zeichenfolge (nur "" ohne ein Zeichen dazwischen)
Dritter	Felder mit dem Wert Null für einen fehlenden Wert

Tabelle 8.5: Formatabschnitte für Textformate

Das folgende benutzerdefinierte Textformat

```
@;"Keine Bemerkung angegeben"
```

wirkt z.B. folgendermaßen: Wenn das Feld keinen Text enthält, erscheint der Text *Keine Bemerkung angegeben*, ansonsten erscheint der vorhandene Text selbst, wofür das @-Zeichen (Platzhalter für Text) sorgt.

8.7.2 Symbole, die für jeden Datentyp verwendbar sind

Die nachfolgenden Symbole können Sie in Ihren benutzerdefinierten Formaten für jeden Felddatentyp verwenden. Die für die einzelnen Felddatentypen spezifischen Symbole finden Sie auf den nächsten Seiten unter den entsprechenden Punkten.

Symbol	Bedeutung
Leerzeichen	Zeigt ein Leerzeichen an.
"Zeichen ..."	Eine beliebige Zeichenfolge, die zwischen Anführungszeichen steht, wird unverändert wiedergegeben. Umgekehrt: Wenn Sie ein oder mehrere Zeichen ausgeben wollen, denen kein von Access vordefiniertes Symbol entspricht, müssen Sie diese zwischen Anführungszeichen (oder, bei einem Zeichen, mit vorangestelltem Backslash, s.u.) angeben.
!	Linksbündige Ausrichtung (anstelle der voreingestellten rechtsbündigen).
*	Füllt verfügbaren Leerraum mit dem auf das Sternchen nachfolgenden Zeichen auf.
\	Das diesem Zeichen (Backslash) folgende eine Zeichen wird unverändert angezeigt. Wenn Sie eine Zeichenfolge anführen wollen, geben Sie diese zwischen Anführungszeichen an, s.o.
" "	Die zwischen Anführungszeichen stehende Zeichenfolge wird unverändert angezeigt.
[Farbe]	Geben Sie zwischen eckigen Klammern eine der folgenden Farben in genau dieser Schreibweise an: Schwarz, Blau, Grün, Cyan, Rot, Magenta-Rot, Gelb, Weiß.

Tabelle 8.6: Formatsymbole für jeden Datentyp

8.7.3 Datum/Uhrzeit-Formate

Die folgende Übersicht gibt die vordefinierten Standardformate für den Datentyp *Datum/Uhrzeit* wieder.

Einstellung	Beschreibung
Standarddatum	Standardeinstellung. Das Datum wird ggf. mit Uhrzeit angezeigt: 02.02.1999 14:23. Ist nur ein Datumswert eingegeben, wird nur dieser angezeigt: 02.02.1999. Ist nur eine Uhrzeit eingegeben, wird nur diese angezeigt: 14:23.
Datum, lang	Gibt das Datum so aus, wie Sie dies in der Windows-Systemsteuerung, Ländereinstellungen, für *Langes Datumsformat* angegeben haben.

Tabelle 8.7: Standardformate für den Datentyp »Datum/Uhrzeit«

Einstellung	Beschreibung
Datum, mittel	Beispiel: 02. Feb.1999
Datum, kurz	Gibt das Datum so aus, wie Sie dies in der Windows-Systemsteuerung, Ländereinstellungen, für *Kurzes Datumsformat* angegeben haben.
Zeit, lang	Gibt die Zeit so aus, wie Sie dies in der Windows-Systemsteuerung, Ländereinstellungen, für Zeitformat angegeben haben.
Zeit, 12Std	Beispiel: 6:30
Zeit, 24Std	Beispiel: 18:30

Tabelle 8.7: Standardformate für den Datentyp »Datum/Uhrzeit«

Für benutzerdefinierte Datums- und Zeitformate stehen Ihnen die nachfolgenden Symbole zur Verfügung:

Symbol *Access*	*VBA*	Bedeutung
:	:	Zeit-Trennzeichen
.	/	Datum-Trennzeichen
g	c	Entspricht dem Format Standarddatum, s.o.
t	d	Der Tag wird als Zahl ohne führende Null dargestellt (1-31).
tt	dd	Der Tag wird als Zahl mit führender Null dargestellt (01-31).
ttt	ddd	Der Tag wird als abgekürzter Wochentag dargestellt (Son-Sam).
tttt	dddd	Der Tag wird als voller Wochentag dargestellt (Sonntag-Samstag).
ttttt	ddddd	Entspricht dem Format Datum, lang, s.o.
w	w	Tag der Woche (1-7).
ww	ww	Woche im Jahr (1-54). Beachten Sie, daß diese Woche oft gegenüber der in Kalendern ausgewiesenen sogen. »Kalenderwoche« um die Zahl 1 größer ist, weil Access die tatsächlich erste angebrochene Woche eines Jahres als Woche 1 annimmt.
m	m	Der Monat wird als Zahl ohne führende Null dargestellt (1-12).
mm	mm	Der Monat wird als Zahl mit führender Null dargestellt (01-12).

Tabelle 8.8: Formatsymbole für Datums- und Zeitformate

Symbol *Access*	*VBA*	Bedeutung
mmm	mmm	Der Monat wird mit seinem abgekürzten Namen dargestellt (Jan-Dez).
mmmm	mmmm	Der Monat wird mit seinem vollen Namen dargestellt (Januar-Dezember).
q	q	Quartal (1-4)
j	y	Kalendertag (1-366)
jj	yy	Letzte beide Ziffern der Jahreszahl (01-99)
jjjj	yyyy	Das Jahr wird als vierziffrige Zahl dargestellt (1900-1999).
h	h	Die Stunde wird ohne führende Null dargestellt (0-23).
hh	hh	Die Stunde wird mit führender Null dargestellt (00-23).
n	n	Die Minute wird ohne führende Null dargestellt (0-59).
nn	nn	Die Minute wird mit führender Null dargestellt (00-59).
s	s	Die Sekunde wird ohne führende Null dargestellt (0-59).
ss	ss	Die Sekunde wird mit führender Null dargestellt (00-59).
zzzzz	ttttt	Entspricht dem Format Zeit, lang, s.o.

Tabelle 8.8: Formatsymbole für Datums- und Zeitformate

Beispiele. Die folgenden Format-Beispiele geben die Angabe 02.02.1999 09:30 auf die jeweils angegebene Weise aus:

Format	Ausgabe
tt. mmmm jjjj	02. Februar 1999
tttt«, den »tt. mmmm jjjj	Sonntag, den 02. Februar 1999
ww	6

Tabelle 8.9: Beispiele für Datums- und Zeitformate

8.7.4 Ja/Nein-Formate

Die folgende Übersicht gibt die vordefinierten Standardformate für den Datentyp *Ja/Nein* wieder. Intern werden die Werte stets als 0 und -1 gespeichert.

Einstellung	Beschreibung
Ja/Nein	Standardeinstellung. Nein = 0, Ja = -1
Wahr/Falsch	Falsch = 0, Wahr = -1
An/Aus	Aus = 0; An = -1

Tabelle 8.10: Tabelle 8.11: Standardformate für den Datentyp »Ja/Nein«

Für benutzerdefinierte Formate stehen drei Formatabschnitte zur Verfügung, von denen aber nur die beiden letzten wirksam sind. Sie müssen daher für den ersten Abschnitt stets ein leeres Semikolon vorsehen. Der zweite Formatabschnitt regelt die Darstellung von -1 (Ja, Wahr, An), der dritte die Anzeige von 0 (Nein, Falsch, Aus).

Beispiel. Das Format

;"zutreffend"[Grün];"unzutreffend"[Rot]

gibt den Wert -1 (Ja, Wahr, An) in grüner Schrift als *zutreffend* aus, den Wert 0 (Nein, Falsch, Aus) in roter Schrift als *unzutreffend*.

8.7.5 Text- und Memo-Formate

Für benutzerdefinierte Formate stehen Ihnen die nachfolgenden Symbole zur Verfügung:

Einstellung	Beschreibung
@	Platzhalter für Textzeichen: Wenn Sie ein @-Zeichen angeben, wird der gesamte Text, den das Feld enthält, angezeigt. Wenn Sie mehr als ein @-Zeichen angeben, steht jedes rechts stehende für ein Zeichen des Textes, den Text von rechts einlesend; das im Format am weitesten links stehende @-Zeichen steht für den restlichen Text, der durch die anderen @-Zeichen nicht angesprochen ist. Beispiel: Das Format @/@-@@-..@ gibt den Text »ABCDEFGHIJK« in der Form »ABCDEFG/H-IJ-..K« aus.
&	Wirkt praktisch gleich wie das Zeichen @, vgl. die vorstehende Erläuterung.
<	Alle Zeichen werden als Kleinbuchstaben dargestellt.
>	Alle Zeichen werden als Großbuchstaben dargestellt.

Tabelle 8.11: Formatsymbole für Textformate

8.7.6 Zahlen- und Währungs-Formate

Die folgende Übersicht gibt die vordefinierten Standardformate für die Datentypen *Zahl* und *Währung* wieder.

Einstellung	Beschreibung
Allgemeine Zahl	Standardeinstellung. Zahlen werden angezeigt, wie sie eingegeben werden.
Währung	Zeigt Zahlen mit Währungssymbol, Tausender-Gruppierungspunkt und zwei Dezimalstellen an.
Festkommazahl	Zeigt mindestens eine Ziffer an; zwei Dezimalstellen.

Tabelle 8.12: Standardformate für Zahlen und Währung

Einstellung	Beschreibung
Standardzahl	Zeigt Tausender-Gruppierungspunkt und zwei Dezimalstellen an.
Prozentzahl	Zeigt eine Zahl mit nachgestelltem Prozentzeichen, zwei Dezimalstellen und mit 100 multipliziert an. Die Multiplikation mit 100 ist (wie die anderen Formatierungen) nur oberflächlich; gerechnet wird mit dem Originalwert.
Exponentialzahl	Zeigt Zahlen im wissenschaftlichen Standardformat an.

Tabelle 8.12: Standardformate für Zahlen und Währung

Für benutzerdefinierte Zahlen- und Währungsformate stehen Ihnen die nachfolgenden Symbole zur Verfügung:

Symbol	Bedeutung
,	Dezimalzeichen.
.	Tausender-Gruppierungspunkt
0	Die Ziffer Null ist ein Platzhalter für alle Ziffern einer Zahl, nicht nur für die Nullen. Wenn im Zahlenformat links oder rechts vom Komma mehr Nullen angegeben sind, als die darzustellende Zahl an diesen Stellen tatsächlich an Ziffern aufweist, wird mit Nullen aufgefüllt. So wird z.B. die Zahl 23,6 im Format 000,00 als 023,60 wiedergegeben. Wenn die darzustellende Zahl rechts vom Komma mehr Ziffern enthält, als im Format Nullen dafür angegeben sind, wird in der Darstellung auf die Anzahl von rechts vom Komma definierten Platzhalternullen aufgerundet (nicht nur abgeschnitten). Wenn andererseits links vom Komma mehr Ziffern als durch Nullen vorgesehen vorhanden sind, werden diese auch dargestellt. So wird die Zahl 123,987 im Format 00,00 als 123,99 wiedergegeben.
#	Dies Ziffernzeichen (im EDV-Jargon manchmal auch als Schweinegitter bezeichnet) hat mit Ausnahme des folgenden Unterschieds dieselbe Bedeutung und Wirkung wie das Formatsymbol 0: Wenn die Zahl rechts oder links vom Komma weniger Ziffern aufweist als dort durch die Anzahl von #-Symbolen vorgesehen sind, wird nicht durch Nullen aufgefüllt. Beispiel: Die Zahl 23,6 im Format ###,## wird als 23,6 dargestellt. Das Nichtauffüllen mit Nullen hat auch zur Folge, daß z.B. die Zahl 0,23 im Format ###,## in der Form ,23 erscheint, also ohne die bei uns übliche Null vor dem Komma. Um zu erreichen, daß Werte kleiner als 1 mit einer Null vor dem Komma ausgegeben werden, sollten Sie als Platzhalter für die Ziffer unmittelbar vor dem Komma stets das Symbol 0 angeben. So wird die Zahl 0,23 im Format ##0,## als 0,23 dargestellt. Entsprechendes gilt für die Ziffer unmittelbar rechts vom Komma: Die Zahl 23 wird mit ##0,## als 23, und mit ##0,0# als 23,0 angezeigt.

Symbol	Bedeutung
$	Zeigt das Zeichen $ an.
%	Die Angabe des Prozentzeichens in einem Format bewirkt, daß die Zahl in der Darstellung mit 100 multipliziert und mit einem Prozentzeichen versehen erscheint; am Wert der Zahl wird indes, wie stets bei Formaten, nichts geändert.
E- oder e- E+ oder e+	Die Angabe eines dieser vier Symbole hinter einem durch die Ziffernsymbole usw. definierten Formats bewirkt die Ausgabe in der Gleitkomma- bzw. Exponentialdarstellung; rechts neben einem dieser vier Symbole ist ferner mindestens ein Ziffernsymbol (0, # oder ?) anzuführen; die Anzahl und Art der Ziffernsymbole legt fest, mit welcher Anzahl von Ziffern und in welcher Art der Exponent ausgegeben wird. So wird die Zahl 12345 im Format #,##E+0 als 1,23E+4 ausgegeben; die Zahl 0,12345 erscheint mit demselben Format als 1,23E-1. Wenn rechts von E+, e+ usw. z.B. mehr als ein Symbol 0 angeführt wird, wirkt dies auch an dieser Stelle wie oben für das Symbol 0 beschrieben, d.h., es werden ggf. führende Nullen beim Exponenten ergänzt. So bewirkt das Format #,##e+000 für die Zahl 0,123 die Ausgabeform 1,23e-001. Wenn statt E das kleine e angegeben wird, wird dies auch wiedergegeben. E+ bzw. e+ bewirkt, daß negative Exponenten mit dem Minuszeichen und positive mit dem Pluszeichen ausgegeben werden. Wird mit E- bzw. e- formatiert, so erscheint nur das Minuszeichen im Falle negativer Exponenten, positive Exponenten werden dagegen ohne Vorzeichen dargestellt.

Beispiele. Die folgende Übersicht gibt ein paar benutzerdefinierte Zahlenformate und ihre Wirkung wieder.

Format	Wert	Darstellung
#,##	1234,567	1234,57
	1234	1234,
	0,567	,57
#0,0#	1234,567	1234,57
	1234	1234,0
	0,567	0,57
"Y=" #0,0##	1234,567	Y= 1234,567
# »Mio« DM	1234,567	1 Mio DM

Tabelle 8.13: Beispiele für Zahlen- und Währungsformate

8.8 Tabelleneigenschaften festlegen

Eine Tabelle besitzt die fünf Eigenschaften *Beschreibung*, *Gültigkeitsregel*, *Gültigkeitsmeldung*, *Filter* und *Sortiert nach* sowie fünf weitere, die sich auf ein mögliches Unterdatenblatt für die Tabelle beziehen. Sie stellen diese Eigenschaften im Eigenschaftenfenster (vgl. Bild 8.32) ein, welches Sie ggf. durch Klicken auf die Symbol-Schaltfläche *Eigenschaften* oder mit dem Befehl *Eigenschaften* aus dem Menü *Ansicht* einblenden müssen.

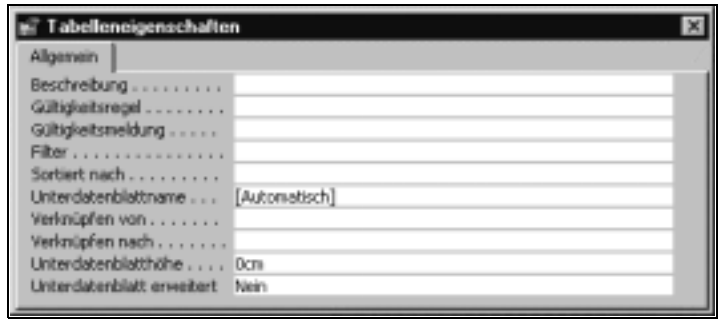


Bild 8.32: Eigenschaftenfenster für eine Tabelle in der Entwurfsansicht

8.8.1 Gültigkeitsregel

Die Gültigkeitsregel für die Tabelle wird überprüft, wenn der Datensatz gespeichert wird oder den Fokus verliert. Dagegen wird die Gültigkeitsregel für ein Tabellenfeld überprüft, wenn das bearbeitete Feld den Fokus verliert, ohne daß dabei notwendigerweise der Datensatz den Fokus verliert. Die Gültigkeitsregel für die Tabelle dient daher vor allem dem Zweck, die Gültigkeit des Datensatzes zu überprüfen. Dabei ist insbesondere ein Vergleich zweier oder mehrerer Felder miteinander möglich. Der folgende Ausdruck – als Gültigkeitsregel für die Tabelle eingegeben – beispielsweise prüft vor dem Speichern des aktuellen Datensatzes, ob das Lieferdatum nicht vor dem Bestelldatum liegt:

[Lieferdatum] >= [Bestelldatum]

Sie können für eine Tabelle nur *einen* Ausdruck als Tabellen-Gültigkeitsregel angeben. Gleichwohl ist es de facto möglich, mehrere Gültigkeitsregeln zu formulieren, denn der Gültigkeitsausdruck darf auch den logischen Operator *Und* enthalten. Wenn Sie beispielsweise außer der gerade angeführten Regel auch noch sicherstellen wollen, daß das Lieferdatum nicht vor dem aktuellen Datum liegen darf, formulieren Sie den Gültigkeitsausdruck in der Form

[Lieferdatum]>=[Bestelldatum] Und [Lieferdatum]>=Datum()

Wenn Sie sich in einem Ausdruck auf ein Feld beziehen, brauchen Sie den Feldnamen im allgemeinen nur dann zwischen eckigen Klammern anzugeben, wenn er Leerzeichen oder Sonderzeichen enthält. Im Ausdruck für die Tabelleneigenschaft *Gültigkeitsregel* müssen Sie die eckigen Klammern jedoch stets anführen.

Ausdrücke für die Tabellen-Gültigkeitsregel, die eine der drei Datumsfunktionen *DatAdd*, *DatDiff* oder *DatTeil* enthielten, führte in Access 97 und vorangehenden Versionen zu einem Fehler. Dieser Bug ist mit Access 2000 endlich fixiert worden!

8.8.2 Gültigkeitsmeldung

Um die Standardmeldung von Access, die beim Verstoß gegen die Tabellen-Gültigkeitsregel eingeblendet wird, durch einen benutzerdefinierten Text zu ersetzen, geben Sie diesen Text als Einstellung der Tabelleneigenschaft *Gültigkeitsmeldung* an.

8.8.3 Beschreibung

Sie können die Eigenschaft *Beschreibung* dazu verwenden, einen beliebigen Text mit der Tabelle zu speichern, der diese aus Ihrer Sicht beschreibt. Im Falle einer verknüpften Tabelle gibt Access mit dieser Eigenschaft die Herkunftsbezeichnung (Name und Pfad der Datenbank, Name der Tabelle) der verknüpften Tabelle an.

8.8.4 Filter

Mit Hilfe der Eigenschaft *Filter* können Sie einen Filter speichern und zu einem späteren Zeitpunkt anwenden. Sie geben den Filter wie die WHERE-Klausel in einer SQL-SELECT-Anweisung an, jedoch ohne das Schlüsselwort WHERE. Beispielsweise können Sie für die Eigenschaft *Filter* die folgende Anweisung angeben (vorausgesetzt, die betreffende Tabelle enthält ein Feld mit dem Namen *Umsatz*):

```
Umsatz >= 2000
```

Dann werden, wenn Sie in der Tabelle von der Entwurfsansicht in die Datenblattansicht wechseln, nur Datensätze mit einem Umsatz von wenigstens 2000 angezeigt.

Ein Filter wird mit der Tabelle gespeichert, ist jedoch nicht automatisch angewendet, wenn die Tabelle erneut geöffnet wird. Wenn Sie einen gespeicherten Filter auf eine Tabelle anwenden möchten, können Sie in der Symbolleiste auf die Symbol-Schaltfläche *Filter/Sortierung anwenden* klicken oder im Menü *Datensätze* den gleichnamigen Befehl wählen.

Wie für Abfragen und Formulare können Sie auch für Tabellen einen Filter mit Hilfe verschiedener dafür vorgesehener Befehle erstellen, vgl. dazu im einzelnen Kap. 6, *Daten bearbeiten*, Punkt 6.10, *Filtern von Datensätzen*. Ein mit den dort beschriebenen Befehlen erstellter Filter für eine Tabelle wird als Einstellung zur Tabelleneigenschaft *Filter* gespeichert.

8.8.5 Sortiert nach

Mit der Eigenschaft *Sortiert nach* können Sie angeben, wie Datensätze in der Tabelle sortiert werden sollen. Sie geben einen Zeichenfolgenausdruck an, der den Namen des Feldes oder der Felder angibt, nach dem oder denen die Datensätze sortiert werden sollen. Wenn Sie mehrere Feldnamen angeben, müssen die Namen durch Semikolons (;) voneinander getrennt werden. Standardmäßig werden die Datensätze in aufsteigender Folge sortiert. Falls Sie Datensätze in absteigender Reihenfolge sortieren möchten, geben Sie am Ende des Zeichenfolgenausdrucks das Schlüsselwort DESC an.

Um beispielsweise die Datensätze einer Tabelle, die die Felder *Ort*, *Nachname* und *Vorname* enthält, in *aufsteigender* Folge nach diesen Feldern zu sortieren, geben Sie für die Eigenschaft *Sortiert nach* den Ausdruck

```
Ort;Nachname;Vorname
```

an. Um eine entsprechende Sortierung in *absteigender* Folge zu erhalten, geben Sie den Ausdruck

```
Ort;Nachname;Vorname DESC
```

an.

Die Einstellung der Eigenschaft *Sortiert nach* wird zusammen mit der Tabelle gespeichert. Sie ist – im Unterschied zur Einstellung der Eigenschaft *Filter*, der erst durch expliziten Befehl angewendet werden muß (vgl. den vorangehenden Punkt) – unmittelbar nach dem Öffnen der gespeicherten Tabelle in der Datenblattansicht wirksam.

8.8.6 Unterdatenblatt-Eigenschaften

Sie können mit fünf Tabelleneigenschaften bestimmen, ob und wie ein Unterdatenblatt für eine Mastertabelle angezeigt wird. Diesem Zweck dienen die Eigenschaften *Unterdatenblattname*, *Verknüpfen von*, *Verknüpfen nach*, *Unterdatenblatthöhe* und *Unterdatenblatt erweitert*. Die Werte dieser Eigenschaften werden von Access eingestellt, wenn Sie ein Unterdatenblatt mit Hilfe von Menübefehlen ein- oder ausblenden, entfernen oder einfügen, vgl im einzelnen Kap. 6, *Daten bearbeiten*, Punkt 6.5.2 *Gestaltung einzelner Datenblätter.* Die Eigenschaften haben im einzelnen die nachfolgende Bedeutung:

Unterdatenblattname

Mit Hilfe dieser Eigenschaft geben Sie die Tabelle oder Abfrage an, die an das Unterdatenblatt gebunden ist. Wenn Sie beispielsweise für die Tabelle *Personal* ein Unterdatenblatt anzeigen lassen wollen, dessen Datensätze aus der Tabelle *Projekte* stammen, geben Sie den Namen *Projekte* an. Wenn Sie keinen Tabellen- oder Abfragenamen für die Eigenschaft *Unterdatenblattname* angeben oder einen vorher vorhandenen löschen, wird kein Unterdatenblatt angezeigt.

Verknüpfen von, Verknüpfen nach

Die Eigenschaft *Verknüpfen von* wird auf den Primärschlüssel der Mastertabelle, *Verknüpfen nach* auf den Fremdschlüssel der Tabelle oder Abfrage eingestellt, die zur Eigenschaft *Unterdatenblattname* angegeben ist. Im allgemeinen füllt Access die Werte dieser beiden Eigenschaften automatisch aus, nachdem ein Name für die Eigenschaft *Unterdatenblattname* angegeben wurde.

Unterdatenblatthöhe

Mit Hilfe der Eigenschaft *Unterdatenblatthöhe* können Sie die Anzeigehöhe beim Einblenden eines Unterdatenblattes angeben. Der Wert für diese Eigenschaft hat begrenzende Wirkung: Enthält ein Unterdatenblatt mehr Datensätze, als in der spezifizierten *Unterdatenblatthöhe* angezeigt werden können, werden nur die ersten Datensätze, die in das Unterdatenblatt passen, angezeigt. Das Unterdatenblatt läßt sich allerdings mit Hilfe einer Bildlaufleiste scrollen, wenn es aktiviert ist, so daß auch seine unteren Datensätze angezeigt werden können. Enthält ein Unterdatenblatt dagegen weniger Datensätze, als mit der angegebenen *Unterdatenblatthöhe* angezeigt werden könnten, so wird das Unterdatenblatt nur so hoch angezeigt, wie für die vorhandenen Datensätze notwendig.

Unterdatenblatt erweitert

Diese Eigenschaft läßt sich auf einen der Werte *Ja* oder *Nein* einstellen. Die Einstellung *Ja* entspricht der Ausführung des Menübefehls *Format Unterdatenblatt Alles einblenden*, *Nein* entspricht *Format Unterdatenblatt Alles ausblenden*. Im ersten Falle werden beim Anzeigen der Mastertabelle alle Unterdatenblätter angezeigt, d.h., es wird für jeden Datensatz der Mastertabelle ein Unterdatenblatt eingeblendet. Im letzten Falle wird kein Unterdatenblatt angezeigt, sondern nur die Erweiterungssymbole (+-Zeichen) in der linken Randspalte.

8.9 Primärschlüssel setzen, ändern und löschen

Die Bedeutung von Primärschlüsselfeldern wurde bereits in Kap. 7, *Eine Datenbank konzipieren*, Punkt 7.4, *Primärschlüsselfelder*, erklärt. Noch detailliertere Informationen zur Bedeutung und Verwendung des Primärschlüssels finden Sie in Kap. 32, *Datenmodellierung für Fortgeschrittene*, Punkt 32.4, *Primärschlüssel und Indizes bestimmen*. In diesem Abschnitt wird der praktische Umgang mit dem Primärschlüssel erklärt.

8.9.1 Primärschlüssel setzen

- Geben Sie dem Feld, für das Sie den Primärschlüssel setzen möchten, den Fokus. Falls Sie den Primärschlüssel für mehr als ein Feld festlegen wollen: Markieren Sie alle entsprechenden Feldzeilen, indem Sie zunächst auf den Feldmarkierer der ersten Zeile, dann mit jeweils gedrückter [Strg]-Taste auf den jeweiligen Feldmarkierer der weiteren Zeilen klicken.

- Klicken Sie auf die Symbol-Schaltfläche *Primärschlüssel* in der Symbolleiste oder wählen Sie den Befehl *Primärschlüssel* aus dem Menü *Bearbeiten* oder dem Kontextmenü. Im Ergebnis erscheint das Feld (bzw. die Felder), das (die) jetzt den Primärschlüssel bildet(n), mit einem Schlüsselsymbol versehen, vgl. Bild 8.33.

Bild 8.33: Im oben dargestellten Tabellenausschnitt ist ein Primärschlüssel für das Feld »ArtikelNr« festgelegt. Im unten wiedergegebenen Tabellenausschnitt ist der Primärschlüssel für die beiden Felder »ArtikelNr« und »LieferantenNr« festgelegt.

8.9.2 Primärschlüssel ändern

Um den Primärschlüssel einem anderen Feld oder mehreren anderen Feldern zuzuweisen, verfahren Sie wie gerade unter 8.9.1, *Primärschlüssel setzen*, beschrieben. Sie heben die alte Zuweisung auf, indem Sie einen neuen Primärschlüssel setzen.

Wenn die betreffende Tabelle die Mastertabelle in einer Beziehung ist, können Sie den Primärschlüssel erst ändern, nachdem Sie die Beziehung zur Detailtabelle im Fenster *Beziehungen* gelöscht haben.

8.9.3 Primärschlüssel löschen

Wenn für eine Tabelle ein Primärschlüssel festgelegt ist, erscheint die Symbol-Schaltfläche *Primärschlüssel* in der Symbolleiste gedrückt, wenn das bzw. ein Feld mit dem Primärschlüssel den Fokus hat. Klicken Sie in dieser Situation auf die Symbol-Schaltfläche *Primärschlüssel*, um die Festlegung aufzuheben. Dies ist, entsprechend wie bei einer Änderung, nur möglich, wenn die betreffende Tabelle nicht Mastertabelle in einer festgelegten Beziehung ist. Diese müßte ggf. vor dem Löschen des Primärschlüssels im Fenster *Beziehungen* gelöscht werden.

8.10 Index erstellen

Ein Index dient dazu, Such- und Sortiervorgänge in einer Datenbank zu beschleunigen. Mit dem Primärschlüssel wird stets implizit auch ein Index für das betreffende Feld gebildet. Darüber hinaus können Sie aber für weitere Felder der Tabelle Indizes festlegen. Allerdings sollten Sie beachten, daß ein Index nicht nur Geschwindigkeitsvorteile, sondern auch -nachteile hat: Während Such- und Sortiervorgänge für ein indiziertes Feld deutlich schneller vonstatten gehen, verlangsamt sich die Daten*eingabe*, denn der Index muß nach jeder Eingabe eines neuen oder Bearbeitung eines bestehenden Datensatzes aktualisiert werden, was mit Zeitaufwand verbunden ist. Sie sollten ein Feld daher im allgemeinen nur indizieren, wenn die *beiden* folgenden Bedingungen zutreffen:

- Es ist zu erwarten, daß nach Werten dieses Feldes oft gesucht oder sortiert wird. Dabei zählen nicht zuletzt auch Such- und Sortiervorgänge mit Abfragen.
- Es ist zu erwarten, daß das Feld viele unterschiedliche Werte aufweisen wird. Für ein Feld, das sehr viele gleiche Werte enthält (z.B. das Feld *Sex* mit den beiden möglichen Werten *m* und *n*), trägt ein Index praktisch nicht zur Beschleunigung bei.

Weitere Einzelheiten zum Gebrauch von Indizes finden Sie in Kap. 32, *Datenmodellierung für Fortgeschrittene* Punkt 32.4.4, *Vor- und Nachteile weiterer Indizes*.

Ein Index kann sowohl für ein Feld wie auch für mehrere Felder gleichzeitig definiert werden. Eine Tabelle kann, auch wenn dies nicht stets empfehlenswert ist, viele Indizes gleichzeitig haben. Mehrfelder-Indizes bieten sich an, wenn in Abfragen oft nach derselben Feldkombination gesucht oder sortiert wird. Beispielsweise könnte es sein, daß in einer Adressenabfrage stets nach den Feldern *Nachname*, *Vorname* und *Ort* sortiert werden soll. Dann bietet sich für die Tabelle *Adressen*, welche der Abfrage zugrunde liegt, ein Mehrfelder-Index für diese Felder an.

8.10.1 Einfeld-Indizes

Um einen Index für *ein* Feld zu definieren, gehen Sie wie folgt vor:

- Geben Sie dem Feld im oberen Teil des Tabellenentwurfsfensters den Fokus, indem Sie auf eine beliebige Zelle der Feldzeile klicken
- Klicken Sie auf das Eingabefeld *Indiziert* im unteren Fensterteil *Feldeigenschaften*.
- Schlagen Sie das Dropdown-Listenfeld auf, und wählen Sie die gewünschte Indexart. *Ja (Duplikate möglich)* bedeutet, daß auf doppelte Werte in diesem Feld nicht geachtet wird. Mit *Ja (Ohne Duplikate)* werden für das Feld nur eindeutige Werte zugelassen. Diese Wirkung stimmt insoweit mit der des Primärschlüssels überein.

Sie heben einen Einfeld-Index wieder auf, wenn Sie für die Feldeigenschaft *Indiziert* die Einstellung *Nein* wählen.

8.10.2 Mehrfelder-Indizes

Mehrfelder-Index erstellen

Ein Index für mehr als ein Feld wird im Indexfenster der Tabelle definiert. Gehen Sie im einzelnen folgendermaßen vor:

- Blenden Sie ggf. das Indexfenster ein. Klicken Sie dazu auf die Symbol-Schaltfläche *Indizes* oder wählen Sie den gleichnamigen Befehl aus dem Menü *Ansicht*. Das Indexfenster ist in Bild 8.34 zu sehen.

Bild 8.34: Das Indexfenster zeigt an, daß der Primärschlüssel für das Feld »KundenNr« festgelegt ist. Darüber hinaus ist unter dem Namen »Namen« ein Index festgelegt, der die Felder Nachname, Vorname und »Ort« umfaßt, nach denen jeweils in aufsteigender Richtung sortiert wird.

- Tragen Sie in die erste freie Zeile des Indexfensters in die Spalte *Indexname* einen frei gewählten Namen ein. Im Beispiel von Bild 8.34 ist dies der Name *Namen*.
- Klicken Sie in die Spalte *Feldname* derselben Zeile, schlagen Sie darin das Dropdown-Listenfeld auf, und wählen Sie den gewünschten Feldnamen aus. Im Beispiel von Bild 8.34 ist dies der Feldname *Nachname*.
- Klicken Sie in die Spalte *Feldname* der nächsten Zeile, und wählen Sie dort den zweiten Feldnamen, der zum Mehrfelder-Index gehören soll. Verfahren Sie entsprechend für ggf. weitere Feldnamen für denselben Index. Es werden alle Feldnamen, die in Zeilen unmittelbar unterhalb der Zeile mit dem Indexnamen stehen, in diesen Index einbezogen. Im Beispiel von Bild 8.34 bilden die Felder *Nachname*, *Vorname* und *Ort* einen Index.

Per Voreinstellung wird ein Index *aufsteigend* sortiert. Sie können die Sortierrichtung im Indexfenster für jedes Feld getrennt auch als *absteigend* angeben, vgl. die Spalte *Sortierreihenfolge* im Indexfenster von Bild 8.34. Bei einem Mehrfelder-Index wird zunächst nach dem ersten Feld, dann – bei Gleichheit mehrerer Werte im ersten Feld – nach dem zweiten Feld usw. sortiert.

Mehrfelder-Index ändern oder löschen

Sie bearbeiten oder löschen einen derartigen Index, indem Sie die entsprechenden Beschreibungszeilen im Indexfenster bearbeiten oder löschen.

8.10.3 AutoIndex

Sie können Text, der üblicherweise am Anfang oder Ende eines Feldnamens verwendet wird (z.B. die Zeichenfolgen *ID*, *Code* oder *Nr*), mit der Option *AutoIndex bei Importieren/Erstellen* im Register *Tabellen/Abfragen* des Befehls *Optionen* aus dem Menü *Extras* angeben. Wenn Sie dann eine Datendatei (z.B.eine Excel-Tabelle) importieren, die diese Zeichenfolgen in ihren Feldnamen enthält, erstellt Access automatisch einen Index für diese Felder.

8.11 Beziehungen zwischen Tabellen festlegen

Die Bedeutung der verschiedenen zwischen je zwei Tabellen möglichen Beziehungen wird in Kap. 7, *Eine Datenbank konzipieren*, Punkt 7.5, *Welche Beziehungen zwischen den Tabellen?*, sowie in Kap. 32, *Datenmodellierung für Fortgeschrittene*, Punkt 32.6, *Beziehungen bestimmen*, behandelt. An dieser Stelle soll gezeigt werden, wie Sie eine Beziehung praktisch festlegen, bearbeiten und löschen.

8.11.1 Beziehung festlegen

- Vergewissern Sie sich zunächst, daß in der als Mastertabelle vorgesehenen Tabelle ein Primärschlüssel definiert ist, weil dies eine Voraussetzung für das Festlegen einer Beziehung ist. Prüfen Sie ferner, ob in der Detailtabelle ein Feld vorhanden ist, das Sie als Fremdschlüssel angeben können. Primärschlüssel der Mastertabelle und Fremdschlüssel der Detailtabelle müssen sich nicht nur inhaltlich, sondern auch im Datentyp entsprechen. Wenn beispielsweise der Primärschlüssel den Datentyp *Text* hat, muß auch der Fremdschlüssel den Datentyp *Text* besitzen. Beim Datentyp *Zahl* gilt darüber hinaus, daß auch die Feldgröße beider Schlüssel übereinstimmen muß (z.B. beide Felder *Long Integer* oder beide *Double*). Eine Besonderheit gilt, wenn für den Primärschlüssel der Datentyp *AutoWert* festgelegt ist; dann muß der Fremdschlüssel der Detailtabelle den Datentyp *Zahl* mit der Feldgröße *Long Integer* haben.
- Vergewissern Sie sich, daß keine der Tabellen, zwischen denen Sie eine Beziehung festlegen wollen, geöffnet ist. Eine Tabelle ist in dem hier relevanten Sinne auch dann geöffnet, wenn ein Formular, ein Bericht oder eine Abfrage mit Anbindung an die Tabelle geöffnet ist. Andernfalls wird das Herstellen einer Beziehung mit der in Bild 8.35 wiedergegebenen Meldung verweigert. Diese Meldung erhalten Sie übrigens erst, wenn Sie das Dialogfeld *Beziehungen* mit OK bestätigen.

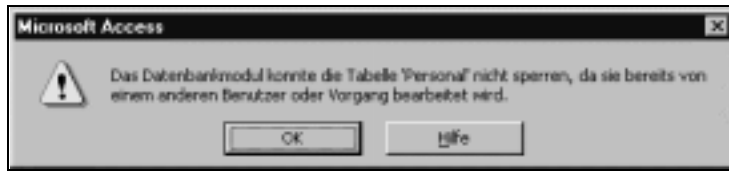


Bild 8.35: Zum Festlegen einer neuen oder Bearbeiten einer bestehenden Beziehung müssen beide Tabellen geschlossen sein. Diese Meldung ist die Reaktion auf den Versuch, eine Beziehung zwischen der Tabelle »Persona«l und einer anderen Tabelle festzulegen, während die Tabelle »Personal« geöffnet war.

- Wechseln Sie ggf. zum Datenbankfenster (z.B. mit F11).
- Klicken Sie auf die Symbol-Schaltfläche *Beziehungen* in der Symbolleiste oder wählen Sie den gleichnamigen Befehl aus dem Menü *Extras*. Die Symbol-Schaltfläche steht nur zur Verfügung, wenn das Datenbankfenster aktiv ist. Als Ergebnis wird das Fenster *Beziehungen* eingeblendet. Sofern Sie dem Fenster *Beziehungen* in der aktiven Datenbank bis dahin noch keine Tabelle oder Abfrage hinzugefügt haben, wird zugleich das Dialogfeld *Tabelle anzeigen* eingeblendet. Diese Situation ist in Bild 8.36 wiedergegeben.

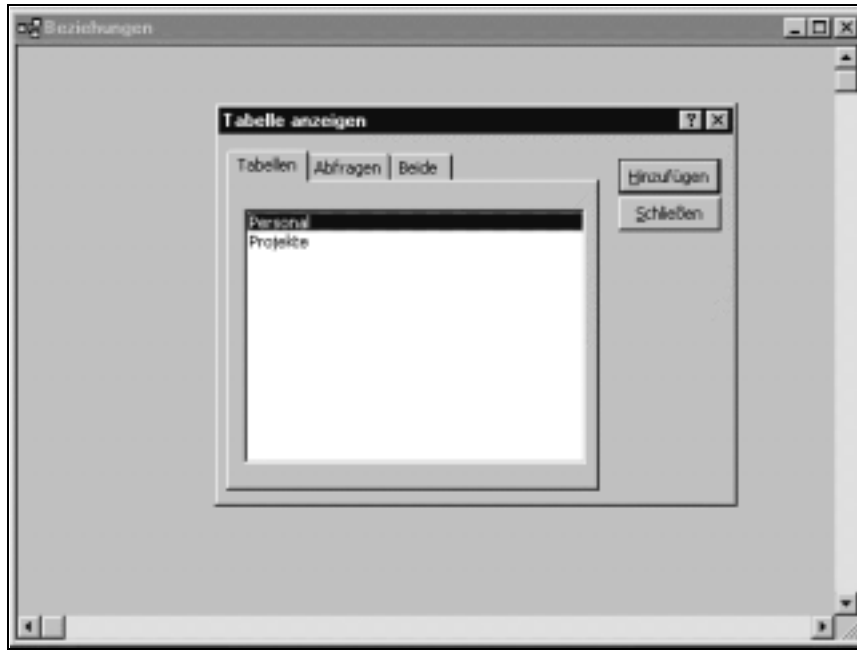


Bild 8.36: In dieser Datenbank wurde dem Fenster »Beziehungen« bisher noch keine Tabelle hinzugefügt. Daher blendet Access das Dialogfeld »Tabelle anzeigen« automatisch mit dem Aufrufen des Fensters »Beziehungen« ein.

- Falls das Dialogfeld *Tabelle anzeigen* nicht bereits automatisch eingeblendet wurde: Sie öffnen es, indem Sie auf die Symbol-Schaltfläche *Tabelle anzeigen* (vgl. links nebenstehend) klicken oder den gleichnamigen Befehl aus dem Menü *Beziehungen* wählen. Wenn sich die beiden Tabellen, zwischen denen Sie eine Beziehung definieren wollen, aufgrund früherer Bearbeitungen bereits im Fenster *Beziehungen* befinden, brauchen Sie diesen Schritt natürlich nicht auszuführen.
- Fügen Sie die benötigten Tabellen mit Hilfe des Dialogfeldes *Tabelle anzeigen* dem Fenster *Beziehungen* hinzu, und schließen Sie dann das Dialogfeld *Tabelle anzeigen*. Das Fenster *Beziehungen* sollte jetzt die beiden Tabellen (ggf. neben anderen) enthalten, zwischen denen Sie eine Beziehung festlegen wollen. In Bild 8.37 sehen Sie ein Fenster *Beziehungen*, dem die beiden Tabellen *Personal* und *Projekte* hinzugefügt wurden.
- Sie leiten das Erstellen einer Beziehung zwischen zwei Tabellen ein, indem Sie mit der Maus das Primärschlüsselfeld der Mastertabelle auf das Fremdschlüsselfeld der Detailtabelle ziehen. Im Beispiel von Bild 8.37 ziehen Sie das Feld *PersonalCode* aus der Tabelle *Personal* auf das Feld *PersonalCode* der Tabelle *Projekte*. Daraufhin zeigt sich das Dialogfeld *Beziehungen*, vgl. Bild 8.38.

Bild 8.37: Fenster »Beziehungen«, nachdem diesem die beiden Tabellen »Personal« und »Projekte« hinzugefügt wurden

Primär- und Fremdschlüsselfeld brauchen übrigens nicht dieselben Namen zu besitzen, obwohl dies beim Design der Tabellen im allgemeinen zweckmäßig ist. Damit eine Beziehung formal korrekt festgelegt werden kann, kommt es nur darauf an, daß die Datentypen der beiden Schlüssel übereinstimmen; vgl. dazu genauer oben den Beginn dieses Punktes 8.11.1, *Beziehung festlegen*.

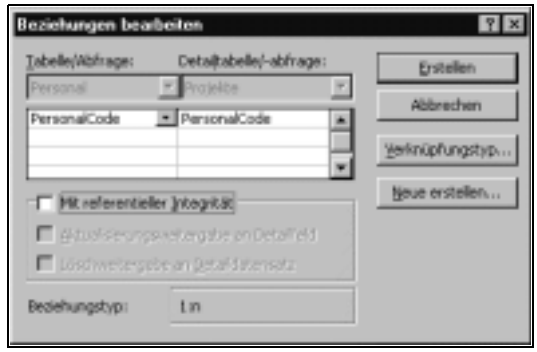


Bild 8.38: Dialogfeld »Beziehungen«

- Sofern Sie beim Ziehen des Primärschlüsselfeldes der Mastertabelle auf das Fremdschlüsselfeld der Detailtabelle keinen Fehler gemacht haben, werden diese beiden Felder im Dialogfeld *Beziehungen* korrekt unter *Tabelle/Abfrage* bzw. *Detailtabelle/-abfrage* angezeigt. Sofern dies nicht zutrifft, können Sie mit Hilfe der Dropdown-Listenfelder die richtigen Feldnamen bestimmen.
- Wählen Sie die gewünschten Optionen: Wenn die Beziehung *Mit referentieller Integrität* sein soll (vgl. dazu genauer Kap. 7, *Eine Datenbank konzipieren*, Punkt 7.5, *Welche Beziehungen zwischen den Tabellen?*, sowie

Kap. 32, *Datenmodellierung für Fortgeschrittene*, Punkt 32.6, *Beziehungen bestimmen*), kreuzen Sie das entsprechende Kontrollkästchen an. Dann werden auch die Kontrollkästchen *Aktualisierungsweitergabe an Detailfeld* sowie *Löschweitergabe an Detaildatensatz* aktiviert.

Falls Sie eine Beziehung ohne referentielle Integrität festlegen, dient diese im wesentlichen dazu, daß Access beim Erstellen von Abfragen und beim Einfügen eines Unterformulars in ein Hauptformular für diese beiden Tabellen die angegebene Beziehung automatisch als Verknüpfung anbietet. Eine Garantie formal stimmiger Verknüpfungen haben Sie damit jedoch nicht. Dazu müssen Sie referentielle Integrität durchsetzen lassen.

In früheren Versionen von Access konnten Sie explizit bestimmen, ob die Beziehung vom Typ 1:1- oder 1:n sein sollte. Dies ist seit Version 7 (Access 95) nicht mehr der Fall. Vielmehr legt Access den Beziehungstyp nunmehr implizit fest auf Basis des folgenden Zusammenhangs: Wenn der Fremdschlüssel der Detailtabelle gleichzeitig Primärschlüssel der Detailtabelle ist, wird eine 1:1-Beziehung festgelegt, in den anderen Fällen eine 1:n-Beziehung. Der von Access auf diese Weise ermittelte Beziehungstyp wird unten im Dialogfeld *Beziehungen* angegeben, vgl. Bild 8.38.

- Bestätigen Sie das Dialogfeld *Beziehungen* mit der Schaltfläche *Erstellen*. Dann wird die neu erstellte Beziehung in Form einer Linie zwischen den Schlüsselfeldern der Master- und der Detailtabelle angezeigt. Die Art der Beziehung (1:1- oder 1:n-Beziehung) wird durch entsprechende Beschriftung der Linie wiedergegeben.

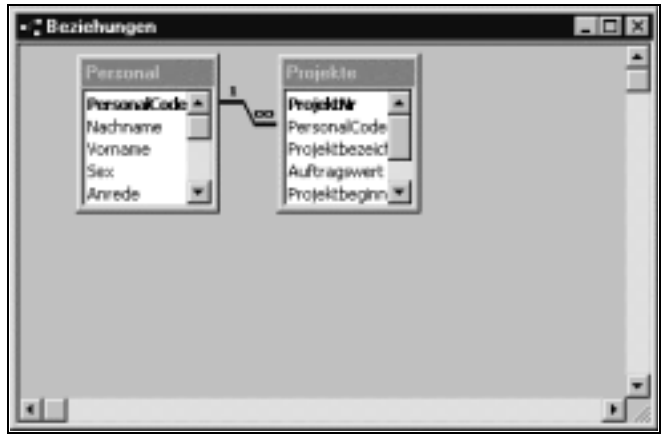


Bild 8.39: Die Linie zwischen den beiden Tabellen zeigt an, daß zwischen den Tabellen »Personal« als Master- und »Projekte« als Detailtabelle eine 1: n-Beziehung festgelegt ist. »PersonalCode« in »Personal« ist der Primärschlüssel der Beziehung, »PersonalCode« in »Projekte« der Fremdschlüssel.

- Schließen Sie das Fenster *Beziehungen*. Dabei werden Sie gefragt, ob Layout-Änderungen an den Beziehungen gespeichert werden sollen, vgl. Bild 8.40. Im allgemeinen sollten Sie diese Frage bejahen, damit Sie die vorgenommene Beziehungsdefinition später auch optisch nachprüfen können. Im hier unterstellten Beispiel betreffen die Layout-Änderungen das Einfügen der beiden Tabellen und das Anzeigen der Beziehungslinie zwischen den Schlüsselfeldern. Falls Sie die Frage nach der Speicherung der Layout-Änderungen verneinen, bleibt die definierte Beziehung gleichwohl gespeichert; sie würde dann beim nächsten Aufschlagen des Fensters *Beziehungen* lediglich nicht angezeigt werden. Aber auch diesen Mangel können Sie, wenn er sich denn später als ein solcher erweisen sollte, leicht wieder beheben: Wählen Sie bei geöffnetem Fenster *Beziehungen* den Befehl *Alle anzeigen* aus dem Menü *Beziehungen* oder klicken Sie auf die Symbol-Schaltfläche *Alle Beziehungen anzeigen*, vgl. links nebenstehend.

Bild 8.40: Beim Schließen des Fensters »Beziehungen« wird Ihnen diese Frage gestellt. Auch wenn Sie die Frage mit »Nein« beantworten, bleibt die Definition der Beziehung davon unberührt.

8.11.2 Beziehung bearbeiten oder löschen

Um eine früher festgelegte Beziehung zu bearbeiten oder wieder aufzuheben, gehen Sie wie folgt vor:

- Vergewissern Sie sich, daß keine der Tabellen, deren Beziehung Sie bearbeiten oder löschen wollen, geöffnet ist (auch nicht indirekt durch Anbindung an ein geöffnetes Formular usw.).
- Wechseln Sie ggf. zum Datenbankfenster (z.B. mit F11).

- Klicken Sie auf die Symbol-Schaltfläche *Beziehungen* in der Symbolleiste (vgl. links nebenstehend), oder wählen Sie den Befehl *Beziehungen* aus dem Menü *Extras*; die Symbol-Schaltfläche steht nur zur Verfügung, wenn das Datenbankfenster aktiv ist. Als Ergebnis wird das Fenster *Beziehungen* eingeblendet.

- Falls Sie die zu bearbeitende oder zu löschende Beziehung nicht finden: Klicken Sie bei aktivem Fenster *Beziehungen* auf die Symbol-Schaltfläche *Alle Beziehungen anzeigen* (vgl. links nebenstehend) oder wählen Sie den Befehl *Alle anzeigen* aus dem Menü *Beziehungen*.
- **Bearbeiten**. Doppelklicken Sie auf die betreffende Beziehungslinie. Dann öffnet sich das Dialogfeld *Beziehungen*, vgl. oben Bild 8.38. Nehmen Sie die gewünschten Änderungen vor, und bestätigen Sie diese mit (in diesem Falle) OK.

Löschen. Klicken Sie einmal auf die betreffende Beziehungslinie, um diese zu markieren. Drücken Sie dann die [Entf]-Taste oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*.

8.11.3 Tabellenentwurf vom Beziehungsfenster aus bearbeiten

Es kann vorkommen, daß sich eine Beziehung nicht erstellen läßt, weil beispielsweise die Datentypen von Primärschlüssel und Fremdschlüssel nicht zusammenpassen. In derartigen Fällen müssen Sie Änderungen am Tabellenentwurf vornehmen. Dies können Sie unmittelbar vom Fenster *Beziehungen* aus bewerkstelligen. Um beispielsweise die Entwurfsansicht der Tabelle *Projekte* unmittelbar aus dem Beziehungsfenster heraus zu öffnen, gehen Sie folgendermaßen vor (vgl. das Beziehungsfenster oben in Bild 8.39):

- Klicken Sie im Fenster *Beziehungen* mit der rechten Maustaste auf die Tabelle *Projekte*, um das Kontextmenü aufzuschlagen.
- Wählen Sie den Kontextmenü-Befehl *Tabellenentwurf*. Dann wird die Tabelle *Projekte* in der Entwurfsansicht geöffnet.
- Nehmen Sie die erforderlichen Bearbeitungen am Tabellenentwurf vor, und speichern und schließen Sie die Tabelle. Dann gelangen Sie wieder zum Fenster *Beziehungen*, in dem Ihnen nunmehr die geänderte Tabellendefinition zur Verfügung steht.

8.11.4 Bestehende Beziehungen anzeigen

Wenn Sie sich darüber informieren wollen, welche Tabellen-Beziehungen insgesamt in der aktuellen Datenbank definiert sind, gehen Sie folgendermaßen vor:

- Klicken Sie bei aktivem Fenster *Beziehungen* auf die Symbol-Schaltfläche *Alle Beziehungen anzeigen* (vgl. links nebenstehend) oder wählen Sie den Befehl *Alle anzeigen* aus dem Menü *Beziehungen*.

8.11.5 Tabellen aus dem Fenster *Beziehungen entfernen*

Um eine Tabelle aus dem Fenster *Beziehungen* zu entfernen, gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. zum Datenbankfenster (z.B. mit [F11]).

- Klicken Sie auf die Symbol-Schaltfläche *Beziehungen* in der Symbolleiste (vgl. links nebenstehend) oder wählen Sie den gleichnamigen Befehl aus dem Menü *Extras*.
- Klicken Sie auf einen Punkt der zu entfernenden Tabelle, um diese zu markieren.
- Drücken Sie die [Entf]-Taste oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*.

8.11.6 Beziehungen zwischen verknüpften Tabellen festlegen und bearbeiten

Wenn Sie Tabellen einer Access-Datenbank in die aktuelle Datenbank durch Verknüpfung einbinden (vgl. Kap. 9, *Tabellen importieren, exportieren und verknüpfen*, Punkt 9.3, *Verknüpfen von Tabellen*), werden die ggf. in der anderen Datenbank definierten Beziehungen in die aktuelle übernommen. Sie erkennen dies im Dialogfeld *Beziehungen* daran, daß der Beziehungstyp als *Extern* gekennzeichnet wird, vgl. das Dialogfeld *Beziehungen* in Bild 8.41. Die Beziehung zwischen zwei verknüpften Tabellen kann nicht in der Datenbank, zu der die Tabellen verknüpft (in die sie eingebunden) sind, definiert oder gelöscht werden, sondern nur in der Datenbank, zu der die Tabellen unmittelbar gehören, aus der sie also verknüpft sind. Die Bearbeitung einer derartigen Beziehung ist eingeschränkt möglich; beispielsweise können Sie die Eigenschaft *Mit referentieller Integrität* auch für die Beziehung zweier verknüpfter Tabellen ändern. Andererseits sind einige Dialogfeldelemente, die sonst zum Bearbeiten einer Beziehung zur Verfügung stehen, im Dialogfeld in Bild 8.41 deaktiviert. Um eine solche Beziehung zu definieren, zu löschen oder uneingeschränkt zu bearbeiten, müssen Sie die Datenbank, aus der die Tabellen stammen, öffnen. Dort kann die Beziehung im Fenster *Beziehungen* bearbeitet werden.

Bild 8.41: Das Dialogfeld »Beziehungen« ist hier für eine Beziehung zwischen zwei aus einer anderen Datenbank verknüpften Tabellen geöffnet. Der Beziehungstyp wird daher als »Extern« gekennzeichnet.

8.11.7 Fenster *Beziehungen drucken*

Mit Access 2000 ist es erstmalig möglich, den Inhalt des Fensters *Beziehungen* in annähernd der visualisierten Form auszudrucken, wie er auf dem Bildschirm angezeigt wird. Gehen Sie dazu folgendermaßen vor:

- Öffnen Sie das Fenster *Beziehungen*.
- Wählen Sie aus dem Menü *Datei* den Befehl *Beziehungen drucken*. Dann erstellt Access einen Bericht, der den Inhalt des Fensters *Beziehungen* in visualisierter Form wiedergibt.

Sie können den Bericht dann mit dem Befehl *Drucken* aus dem Menü *Datei* ausdrucken. Da der Bericht standardmäßig im Hochformat erstellt wird, empfiehlt es sich im allgemeinen, im Dialogfeld *Drucken* durch Klicken auf die Schaltfläche *Eigenschaften* das entsprechende Dialogfeld *Eigenschaften* für den aktuellen Drucker zu öffnen und darin *Querformat* zu wählen.

8.12 Tabelle drucken

Daten drucken

Mit dem Befehl *Drucken* aus dem Menü *Datei* können Sie die Daten einer Tabelle auf dem Drucker ausgeben lassen. Dieser Befehl zeigt das Dialogfeld *Drucken* an, vgl. Bild 8.42.

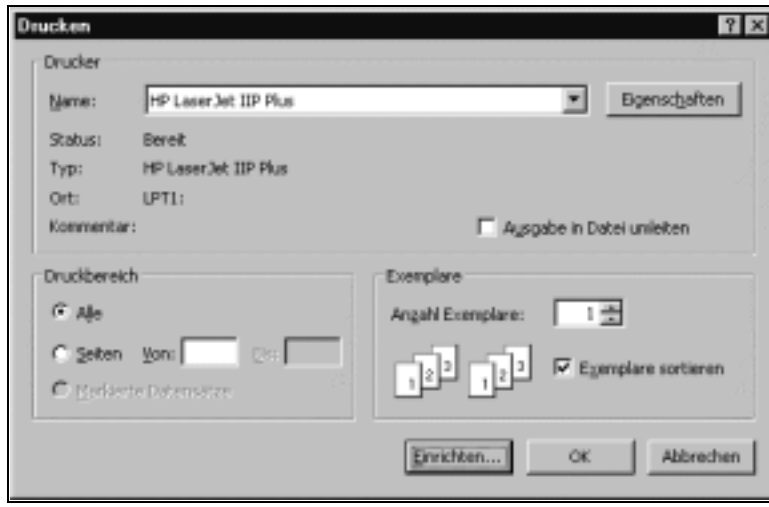


Bild 8.42: Dialogfeld »Drucken«

Vielleicht sollten Sie vor dem Drucken die Seitenränder, das Ausgabeformat (Hoch- oder Querformat) und weitere Optionen festlegen. Dazu öffnen Sie zwei Dialogfelder mit den Schaltflächen *Einrichten* und *Eigenschaften*, die das Dialogfeld *Drucken* anbietet, vgl. Bild 8.42. In jedem Falle sollten Sie sich die Tabelle, bevor Sie den Drucker aufrufen, in der Seitenansicht betrachten, um zu überprüfen, ob z.B. alle Beschriftungen wie gewünscht wiedergegeben werden und alle Spalten wie gewünscht auf dieselbe Seite passen.

Tabellendefinitionen drucken

Bild 8.43: Registerkarte »Tabellen« des Dokumentierers.

Sie können auch die Definitionen einer oder mehrerer Tabellen drucken lassen. Rufen Sie dazu den *Dokumentierer* auf:

- Wählen Sie im Menü *Extras* den Befehl *Analyse* und dessen Unterbefehl *Dokumentierer.* Wählen Sie dann die Registerkarte *Tabellen*, und markieren Sie darin die Tabellen, deren Definitionsmerkmale Sie drucken wollen, vgl. Bild 8.43.
- Öffnen Sie mit der Schaltfläche *Optionen* in der Registerkarte *Tabellen* das Dialogfeld *Tabellendefinition drucken*, vgl. Bild 8.44. Legen Sie dort fest, welche Definitionsmerkmale im Druck ausgegeben werden sollen. Beachten Sie dabei, daß dabei leicht ein viele Seiten umfassendes Druckergebnis zustande kommt. Insbesondere die Option *Namen, Datentypen, Größen und Eigenschaften* führt zu einer großen Menge von Informationen.
- Bestätigen Sie das Dialogfeld *Tabellendefinition drucken* mit OK. Dann kehren Sie zum Dialogfeld *Dokumentierer*, Registerkarte *Tabellen*, zurück. Schließen Sie dieses Dialogfeld mit OK. Dann erstellt der Dokumentierer einen Bericht, den Sie sich zunächst auf dem Bildschirm anschauen sollten, bevor Sie diesen dann möglicherweise auf dem Drucker ausgeben lassen.

In Bild 8.45 ist ein Ausschnitt eines Berichts mit Definitionsmerkmalen der Tabelle *Personal* aus der Beispieldatenbank *Projekte.mdb* wiedergegeben.

Bild 8.44: Dialogfeld »Tabellendefinition drucken«

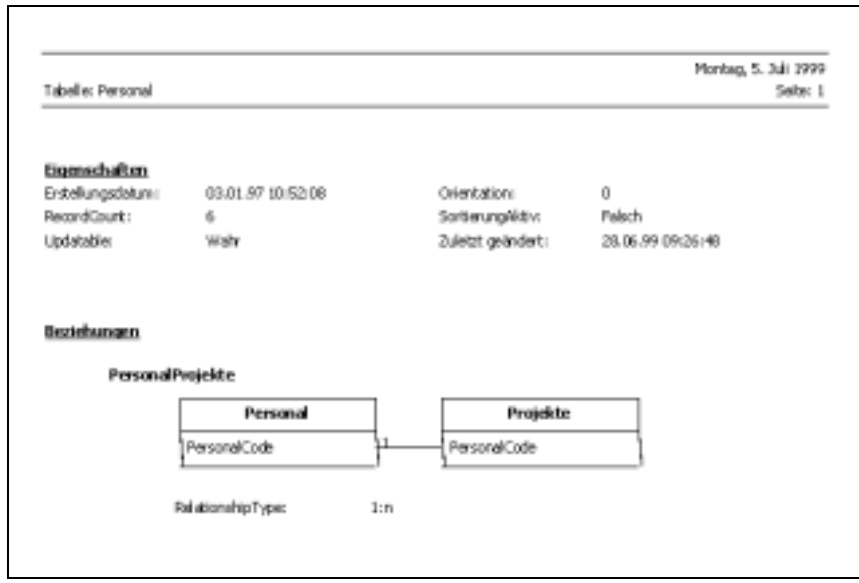


Bild 8.45: Definitionen der Tabelle »Personal« mit den reinen Tabellendefinitionen ohne Definitionen für Felder und Indizes, wie dies den gewählten Optionen in Bild 8.44 entspricht.

Kapitel 9

Tabellen importieren, exportieren und verknüpfen

Wenn Access eine Tabelle aus einer anderen Anwendung importiert, stellt es davon eine (in Access-Format konvertierte) Kopie her, beläßt somit die Quelltabelle unverändert. Tabellenexport bedeutet entsprechend genau das Umgekehrte. Import und Export sind streng vom *Verknüpfen* einer Tabelle zu unterscheiden, worauf im Punkt 9.3, *Verknüpfen von Tabellen*, eingegangen wird.

Sie können Tabellen in diversen Formaten im- und exportieren. Die wichtigsten sind die folgenden:

- *Microsoft Access* (andere Datenbanken als die geöffnete Datenbank), alle Versionen
- *Text mit Trennzeichen* (Werte, die durch Kommas, Tabulatorzeichen oder andere Zeichen voneinander getrennt sind)
- *Text mit festgelegtem Format* (Werte, die so angeordnet sind, daß jedes Feld eine definierte Breite hat)
- *Microsoft Excel*, alle Versionen
- Lotus 1-2-3
- *Paradox*
- *FoxPro*
- *dBase*
- *Btrieve*
- HTML-Dateien
- Dateien aus einem MAPI-Ordner oder Outlook-Adressbücher
- *ODBC-Datenbanken*, z.B. *SQL-Datenbanken*, die einen ODBC-Treiber verwenden.

Für die unterschiedlichen Formate müssen Sie ggf. verschiedene Einzelheiten beachten. So können Sie beispielsweise beim Import einer Excel-Tabelle den zu importierenden Zellbereich angeben und mitteilen, ob die Spaltenüberschriften als Feldnamen verwendet werden sollen. In diesem Buch wird aus Platzgründen auf die vielen in diesem Zusammenhang möglichen Einzelheiten nicht eingegangen. Sie erklären sich im allgemeinen aus den Dialogfeldern von selbst. Sollten Sie Fragen haben, werden Sie auch sehr ausführlich von der Online-Hilfe informiert. Suchen Sie dort nach den Stichwörtern *Importieren*, *Exportieren* oder einem der Formate, z.B. Paradox, dBase oder Excel.

Hier seien nur kurz die prinzipiellen Vorgehensweisen für das Importieren bzw. Exportieren von Tabellen angegeben. Als Demonstrationsbeispiel wird wegen der besonderen praktischen Relevanz der Fall einer importierten Excel-Kalkulationstabelle vergleichsweise ausführlich dargestellt.

9.1 Importieren von Tabellen

9.1.1 Importieren von Tabellen aus einer Access-Datenbank

Am leichtesten und sichersten ist das Importieren von Tabellen aus einer anderen Access-Datenbank. Dabei darf es sich auch um eine vorangehende Version von Access handeln. Wenn Sie im Dialogfeld *Importieren* (vgl. weiter unten) eine Access-Datenbank auswählen, erkennt der Import-Assistent dieses, und die weiteren Dialogfelder bieten dann nicht nur die Möglichkeit, Tabellen zum Importieren auszuwählen, sondern Sie können auch alle anderen Objekte, die in der betreffenden Datenbank vorhanden sind, auswählen und importieren, z.B. Formulare, Berichte oder Makros. Gehen Sie folgendermaßen vor, um eine oder mehrere Access-Tabellen (und/oder andere Objekte) zu importieren:

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11.
- Klicken Sie auf die Symbol-Schaltfläche *Importieren* in der Symbolleiste (vgl. links nebenstehend, diese wird standardmäßig nicht angezeigt, sondern muß benutzerseitig in eine Symbolleiste eingefügt werden), oder wählen Sie im Menü *Datei* den Befehl *Externe Daten*, Unterbefehl *Importieren. Dann wird das Dialogfeld Importieren* angezeigt.
- Wählen Sie im Dialogfeld *Importieren* aus dem Listenfeld *Dateityp* das (voreingestellte) Format *Microsoft Access*. Dann werden im oberen Teil des Dialogfeldes nur Access-Datenbanken angezeigt. Wechseln Sie ggf. zu einem anderen Laufwerk und/oder Ordner, und markieren Sie die Datenbank, aus der Sie eine oder mehrere Tabellen (ggf. auch andere Objekte, vgl. weiter unten) importieren wollen. Bestätigen Sie das Dialogfeld *Importieren* mit der Schaltfläche *Importieren*. Dann wird das Dialogfeld *Objekte importieren* angezeigt, vgl. Bild 9.1.

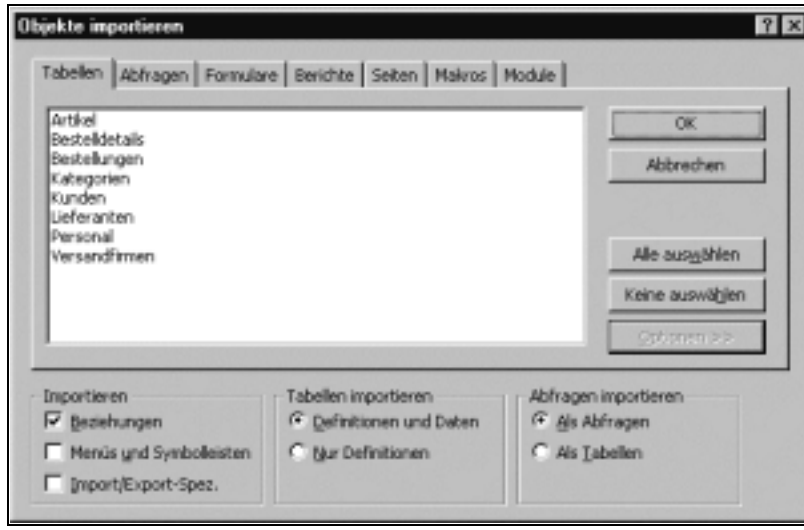


Bild 9.1: Dialogfeld »Objekte importieren« für eine Access-Datenbank. Die im unteren Teil angezeigten Optionen werden eingeblendet, wenn Sie auf die Schaltfläche »Optionen>>« klicken.

- Aktivieren Sie die Registerkarte *Tabellen*, und wählen Sie im Listenfeld eine oder mehrere Tabellen zum Importieren. Wenn Sie auf die Schaltfläche *Optionen>>* klicken, erweitert sich das Dialogfeld *Objekte importieren* um den unteren Teil, der in Bild 9.1 wiedergegeben ist. Mit diesen Optionen können Sie den Umfang der zu importierenden Tabellen und weiterer Objekte bestimmen. Beispielsweise können Sie bewirken, daß auch die Beziehungsdefinition zweier Tabellen zusammen mit diesen importiert wird, wenn Sie das Kontrollkästchen *Beziehungen* ankreuzen. Mit den weiteren fünf Registerkarten können Sie im übrigen auch die anderen Objekte der gewählten Datenbank zum Importieren auswählen. Bestätigen Sie das Dialogfeld schließlich mit OK. Dann werden die ausgewählten Objekte ohne weitere Dialogfeld-Fragen importiert.

Wenn Sie alle im Dialogfeld *Objekte importieren* angebotenen Objekte markiert haben, importieren Sie praktisch die gesamte Datenbank.

9.1.2 Importieren von Tabellen fremder Formate

Hinweis: Im folgenden demonstriere ich den Import einer Excel-Tabelle als Beispiel für das Importieren von Tabellen fremder Formate im allgemeinen. Dabei konzentriere ich mich zunächst auf die rein technischen Aspekte des Importvorgangs. Das von mir verwendete konkrete Beispiel der Excel-Tabelle *Alte Länder*, die Ergebnisse einer Bevölkerungsprognose für die alten Bundesländer der BRD enthält, wird formal korrekt, jedoch inhaltlich fehlerhaft sein. Anschließend zeige ich, wie ein inhaltlich befriedigendes Ergeb-

nis erzielt werden kann. Ich habe bewußt einen Weg mit Fehlern demonstriert, weil auf diese Weise deutlicher wird, welche Vorarbeiten Sie an der zu importierenden Tabelle erledigen müssen, um zu korrekten Ergebnissen zu kommen.

Um eine Tabelle zu importieren, gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11.
- Klicken Sie auf die Symbol-Schaltfläche *Importieren* in der Symbolleiste (vgl. links nebenstehend, diese wird standardmäßig nicht angezeigt, sondern muß benutzerseitig einer Symbolleiste eingefügt werden), oder wählen Sie im Menü *Datei* den Befehl *Externe Daten*, Unterbefehl *Importieren*. *Dann wird das Dialogfeld Importieren* angezeigt, vgl. Bild 9.2.

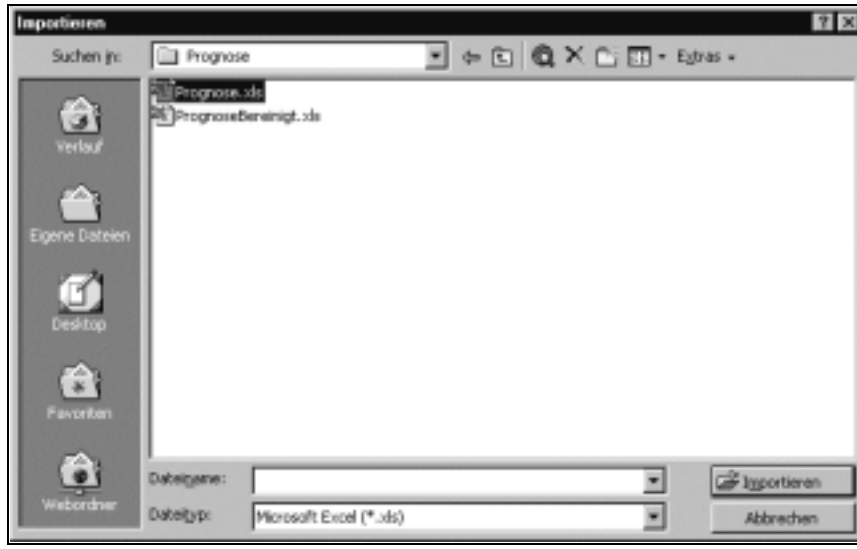


Bild 9.2: Dialogfeld »Importieren«

- Wählen Sie im Dialogfeld *Importieren* aus dem Listenfeld *Dateityp* das Format der zu importierenden Tabelle aus. Wählen Sie hier z.B. *Microsoft Excel (*.xls)*, um eine Excel-Tabelle zu importieren. Dann werden im oberen Teil des Dialogfeldes nur Dateien mit der Erweiterung *.xls* angezeigt, vgl. Bild 9.2.
- Wechseln Sie ggf. das Laufwerk und/oder den Ordner, und markieren Sie die zu importierende Datei bzw. geben Sie ihren Namen mit Pfad an. (Sie können im Dialogfeld *Importieren* auch einen Suchlauf nach Dateien starten, für deren Auswahl Sie Kriterien angeben können, vgl. die Symbolschaltfläche *Extras* mit dem Befehl *Suchen*.) Bestätigen Sie die gewählte Datei mit *Importieren*.

Bild 9.3: Erstes Dialogfeld des Import-Assistenten für Kalkulationstabellen, das die Auswahl eines Tabellenblattes in der Arbeitsmappe oder eines benannten Bereiches ermöglicht.

- Anzahl und Inhalt der folgenden Dialogfelder hängt vom gewählten Dateiformat ab, weil die weiteren Fragen des Import-Assistenten sich vor allem auf die Auswahl und Zuordnung der Datenelemente in der Quelldatei (wie z.B. Textdateien mit Trennzeichen oder fester Zeichenlänge, Auswahl aller Daten oder nur einer Teilmenge usw.) beziehen. Hier wird beispielhaft der Import einer Excel-Tabelle dargestellt, weil der Zugriff auf Daten einer Kalkulationstabelle vermutlich ein praktisch besonders relevanter Fall ist.
- Wenn die im Dialogfeld *Importieren* (vgl. Bild 9.2) markierte Datei Prognose.xls mit der Schaltfläche *Importieren* bestätigt wird, zeigt der *Import-Assistent für Kalkulationstabellen* sein erstes Dialogfeld, vgl. Bild 9.3. Es dient dazu, ein Tabellenblatt oder einen benannten Bereich für den Datenimport auszuwählen. Wenn die Option *Tabellenblätter anzeigen* aktiviert ist, werden im rechts daneben stehenden Listenfeld die Namen der in der Arbeitsmappe vorhandenen Tabellenblätter angezeigt. Wenn Sie eines dieser Tabellenblätter in der Liste markieren, zeigt das Fenster im unteren Teil des Dialogfeldes Beispieldaten aus dem markierten Tabellenblatt an. In der in Bild 9.3 wiedergegebenen Situation werden Beispieldaten für das Tabellenblatt *Alte Länder* angezeigt, weil dieses markiert wurde. Mit der waagerechten und vertikalen Bildlaufleiste läßt sich der Inhalt des Fensters *Beispieldaten* ... rollen, wovon in der in Bild 9.3 dargestellten Situation Gebrauch gemacht wurde. Wenn Sie auf die Option *Benannte Bereiche anzeigen* klicken, zeigt das obere rechte Listenfeld die

Namen der in der gesamten Arbeitsmappe vorhandenen benannten Bereiche an. Wenn Sie den gewünschten Datenbereich (Tabellenblatt oder benannter Bereich) markiert haben, bestätigen Sie dies mit der Schaltfläche *Weiter* >. Dann wird das zweite Dialogfeld des Import-Assistenten für Kalkulationstabellen angezeigt, vgl. Bild 9.4.

Bild 9.4: Zweites Dialogfeld des Import-Assistenten für Kalkulationstabellen, in dem Sie angeben, ob ggf. vorhandene Spaltenüberschriften als Feldnamen übernommen werden sollen.

- Geben Sie im zweiten Dialogfeld des Import-Assistenten (vgl. Bild 9.4) an, ob die erste Zeile Spaltenüberschriften enthält. Falls Sie das entsprechende Kontrollkästchen ankreuzen, werden die Spaltenüberschriften als Feldnamen für die neue Tabelle in Access übernommen. Im dargestellten Beispiel sind keine Spaltenüberschriften vorhanden, so daß das Kontrollkästchen nicht angekreuzt wird. Bestätigen Sie das Dialogfeld mit der Schaltfläche *Weiter* >, dann wird das nächste Dialogfeld angezeigt.
- Bestimmen Sie im dritten Dialogfeld des Import-Assistenten für Kalkulationstabellen (vgl. Bild 9.5), ob die Daten in eine neue Tabelle eingefügt oder an eine bestehende angefügt werden sollen. Wenn Sie die letztere Option wählen, müssen Sie in der rechts neben der Option angezeigten Dropdown-Liste den Namen einer in der Datenbank vorhandenen Tabelle auswählen. Diese Option ist natürlich nur sinnvoll, wenn die bestehende Tabelle, an die Sie Datensätze anfügen wollen, dieselbe Datenstruktur (Reihenfolge oder Namen der Spalten, Datentyp) hat wie der ausgewählte Bereich der importierten Tabelle. Im vorliegenden Beispiel soll in eine neue Tabelle eingefügt werden. Bestätigen Sie dies mit der Schaltfläche *Weiter* >.

Bild 9.5: Drittes Dialogfeld des Import-Assistenten für Kalkulationstabellen, in dem bestimmt wird, ob die Daten in eine neue Tabelle eingefügt oder an eine bestehende angefügt werden sollen.

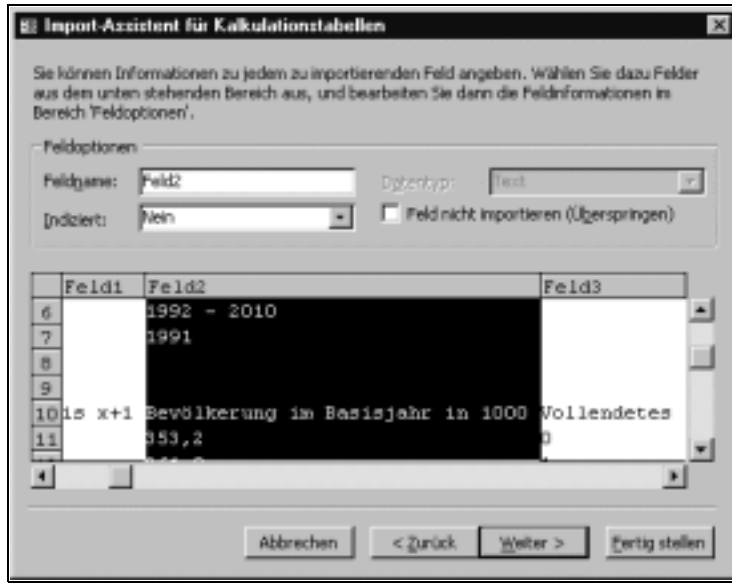


Bild 9.6: Viertes Dialogfeld des Import-Assistenten für Kalkulationstabellen. Hier können Sie Feldnamen vergeben, Spalten überspringen und Indizes festlegen.

- Im vierten Dialogfeld des Import-Assistenten für Kalkulationstabellen (vgl. Bild 9.6) können Sie Anweisungen zu jedem einzelnen Feld (zu jeder einzelnen Spalte) des Inportbereiches angeben: Markieren Sie im unteren Fenster des Dialogfeldes eine Spalte, vergeben Sie dann einen Feldnamen, und entscheiden Sie, ob das Feld indiziert werden soll. Sie können auch bestimmen, daß die markierte Spalte (das Feld) nicht importiert werden soll. Wenn Sie überhaupt keine Veränderung in diesem Dialogfeld vornehmen, also die Voreinstellungen übernehmen, bekommen die Felder die Namen *Feld1*, *Feld2* ..., keines wird indiziert und alle werden importiert. (Falls Sie zuvor die Übernahme der Spaltenüberschriften als Feldnamen gewählt hatten, werden natürlich diese Feldnamen wirksam.) Bestätigen Sie das Dialogfeld mit *Weiter* >.

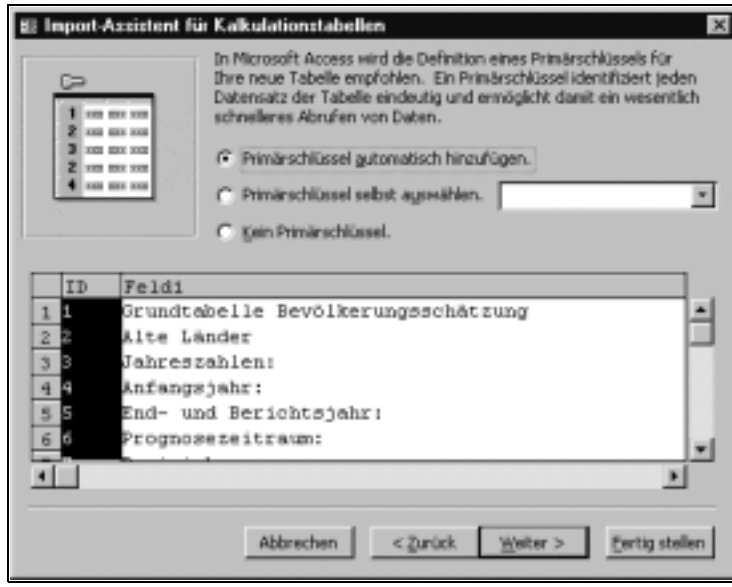


Bild 9.7: Fünftes Dialogfeld des Import-Assistenten für Kalkulationstabellen, in dem Sie über die Vergabe eines Primärschlüssels entscheiden können.

- Im fünften Dialogfeld des Import-Assistenten für Kalkulationstabellen legen Sie fest, ob für die neue Tabelle ein Primärschlüssel festgelegt werden soll, vgl. Bild 9.7. Falls Sie dies wünschen, können Sie vom Import-Assistenten einen Primärschlüssel erstellen lassen; dann wird ein zusätzliches Feld mit den Werten 1, 2, 3 ... erzeugt, vgl. das erste Feld *ID* im unteren Fensterteil des Dialogfeldes von Bild 9.7. Wenn Sie die Option für einen selbst auszuwählenden Primärschlüssel anklicken, müssen Sie das betreffende Feld im unteren Fensterteil des Dialogfeldes markieren. Bestätigen Sie schließlich mit der Schaltfläche *Weiter* >.

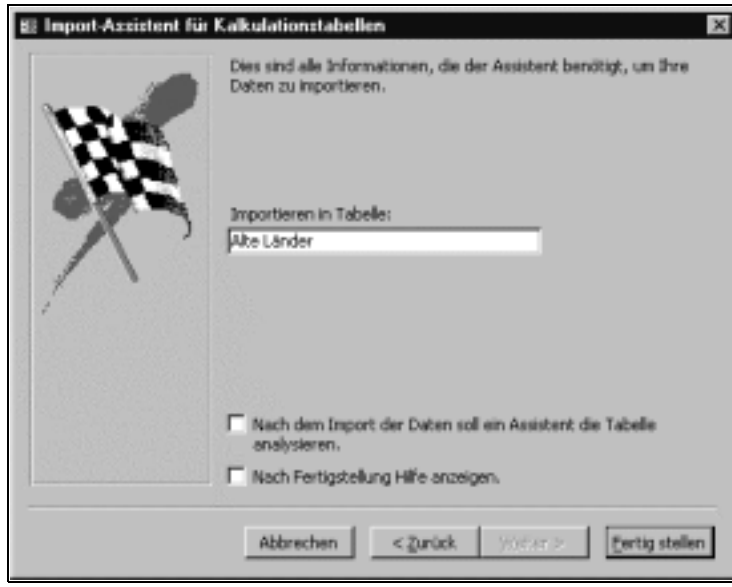
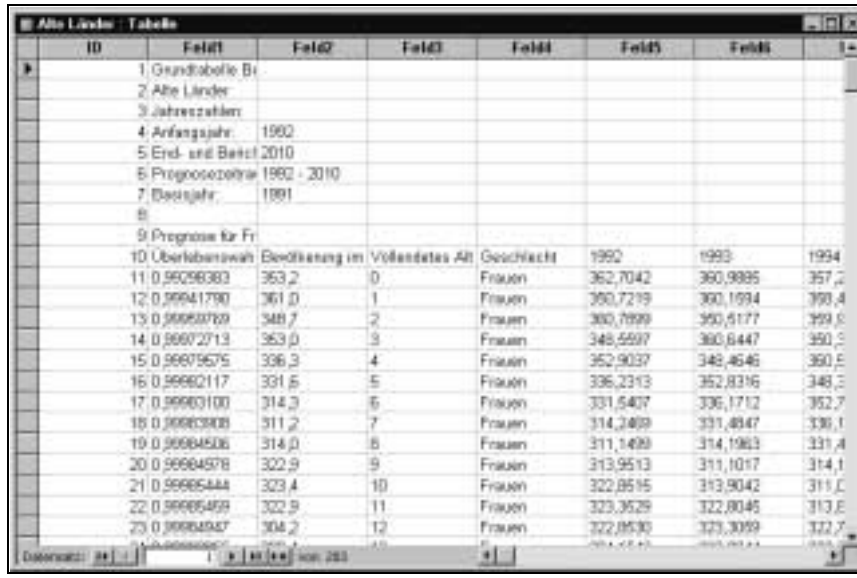


Bild 9.8: Letztes Dialogfeld des Import-Assistenten für Kalkulationstabellen, in dem Sie u.a. einen Namen für die neue Tabelle angeben können.

- Im letzten Dialogfeld des Import-Assistenten für Kalkulationstabellen geben Sie (für unser Beispiel, in dem die Daten in eine neue Tabelle eingefügt werden) einen Namen für die neue Access-Tabelle an, oder Sie übernehmen den vom Import-Assistenten vorgeschlagenen. Ferner können Sie entscheiden, ob die Daten nach dem Import einer Konsistenzanalyse unterzogen und/oder ob zusätzliche Hilfe angezeigt werden soll. Nach meiner persönlichen Erfahrung und Einschätzung ist der Assistent zur Datenanalyse nur von recht begrenztem Wert. Zur Gewährleistung einer für professionelle Zwecke brauchbaren und daher widerspruchsfreien Datenstruktur müssen Sie die Daten selbst analysieren und dürfen dies keinesfalls allein dem Assistenten überlassen. Im allgemeinen sollte die Datenanalyse in der Quelltabelle bereits *vor dem Importieren* erfolgt sein. Bestätigen Sie das letzte Dialogfeld schließlich mit der Schaltfläche *Fertigstellen*.

Das Ergebnis des Tabellenimports ist in Bild 9.9 wiedergegeben. Die Datenblattansicht der Tabelle *Alte Länder*, die vom Import-Assistenten neu erzeugt wurde und die die importierten Daten enthält, sieht der Excel-Quelltabelle recht ähnlich: Sie vermittelt den Eindruck einer typischen Kalkulationstabelle, in der Text und Daten auf dem Arbeitsblatt mehr oder minder wild neben- und untereinander stehen. Dies ist allerdings mit der für Tabellen in Datenbanken erforderlichen widerspruchsfreien Datenstruktur, nach der insbesondere alle Daten derselben Spalte (desselben Feldes) denselben Datentyp besitzen müssen, nicht vereinbar. Entsprechend schlecht ist das

Ergebnis des Tabellenimports geworden: Mit Ausnahme des vom Import-Assistenten neu erzeugten Primärschlüsselfeldes, das den Datentyp *Auto-Wert* bekommen hat und widerspruchsfreie Werte enthält, haben alle anderen Felder den (sozusagen schwächsten) Datentyp *Text* erhalten. Dies liegt daran, daß die Spalten des Importbereichs sowohl Texte wie auch Zahlen aufweisen. Außerdem ist es natürlich gänzlich unbefriedigend, daß die Spalten, deren Zahlen den Jahren 1992, 1993 ... zuzuordnen sind, die Feldnamen *Feld5*, *Feld6* ... bekommen haben. Diese Tabelle muß offensichtlich einer erheblichen Bearbeitung unterzogen werden, bevor sie für die weitere Verwendung in einer Datenbank freigegeben werden kann.



ID	Feld1	Feld2	Feld3	Feld4	Feld5	Feld6	
1	Grundtabelle B						
2	Alte Länder						
3	Jahreszahlen:						
4	Anfangsjahr:	1992					
5	End- und Basis	2010					
6	Prognosezeitra	1992 - 2010					
7	Basisjahr:	1991					
8							
9	Prognose für Fr						
10	Überlebenswah	Bevölkerung im	Vollendetes Alt	Geschlecht	1992	1993	1994
11	0,99298383	363,2	0	Frauen	362,7042	360,9885	357,2
12	0,99941790	361,0	1	Frauen	360,7219	360,1594	358,4
13	0,99993769	348,7	2	Frauen	360,7899	360,6177	359,9
14	0,99972713	353,0	3	Frauen	348,5597	360,6447	360,3
15	0,99979575	336,3	4	Frauen	352,9037	348,4646	360,5
16	0,99962117	331,6	5	Frauen	336,2313	352,8316	348,3
17	0,99983100	314,3	6	Frauen	331,5407	336,1712	352,7
18	0,99983908	311,2	7	Frauen	314,2409	331,4847	336,1
19	0,99984506	314,0	8	Frauen	311,1499	314,1963	331,4
20	0,99984978	322,9	9	Frauen	313,9513	311,1017	314,1
21	0,99985444	323,4	10	Frauen	322,8515	313,9042	311,0
22	0,99985459	322,9	11	Frauen	323,3529	322,8045	313,8
23	0,99984947	304,2	12	Frauen	322,8530	323,3059	322,7

Bild 9.9: Datenblattansicht der Tabelle »Alte Länder«, welche die importierten (unbereinigten) Daten enthält

Es ist allerdings zu entscheiden, ob die Bearbeitung nach dem Importieren oder vorher erfolgen soll. Wenn sich nach dem Import kleinere Fehler ergeben – beispielsweise mag ein einzelnes unter vielen Feldern den falschen Datentyp bekommen haben –, kann es sinnvoll sein, die Korrektur an der neuen Access-Tabelle vorzunehmen. Wenn die Fehler allerdings derartig umfangreich und verschiedenartig sind wie in unserem Beispiel, erscheint es sehr viel sinnvoller, vor dem Importieren die Quelltabelle so zu bereinigen, daß sich automatisch ein korrektes Importergebnis einstellt. Diese Vorgehensweise sei abschließend für unser Beispiel demonstriert.

Vor dem Datenimport muß natürlich in erster Linie klar sein, welche Daten überhaupt verfügbar sein sollen. Für unser Beispiel von Daten einer Bevölkerungsprognose sei angenommen, daß die Bevölkerungsdaten für die Jahre

des Prognosezeitraums (hier 1992 ff.) benötigt werden, unterschieden nach Lebensalter und Geschlecht. Die Quelltabelle muß daher so bereinigt werden, daß nur Spalten mit diesen Informationen darin vorkommen. Alternativ: Sie könnten auch den Bereich benennen (in der Excel-Arbeitsmappe einen Namen für einen Bereich vergeben), der die so aufbereiteten Daten enthält. Denn der Import-Assistent für Kalkulationstabellen erlaubt, wie oben demonstriert, auch die Datenauswahl auf Basis benannter Bereiche. Ferner sollten Sie dafür sorgen, daß im Datenbereich keine Leerzeilen vorkommen. Bild 9.10 zeigt im oberen Teil die Excel-Tabelle *Alte Länder* in unbereinigter Form, wie diese im ersten, fehlerhaften Beispiel importiert wurde. Im unteren Teil von Bild 9.10 wird dagegen eine zweite Excel-Tabelle mit dem Namen *Alte Länder bereinigt* wiedergegeben, in der die Daten bereinigt wurden:

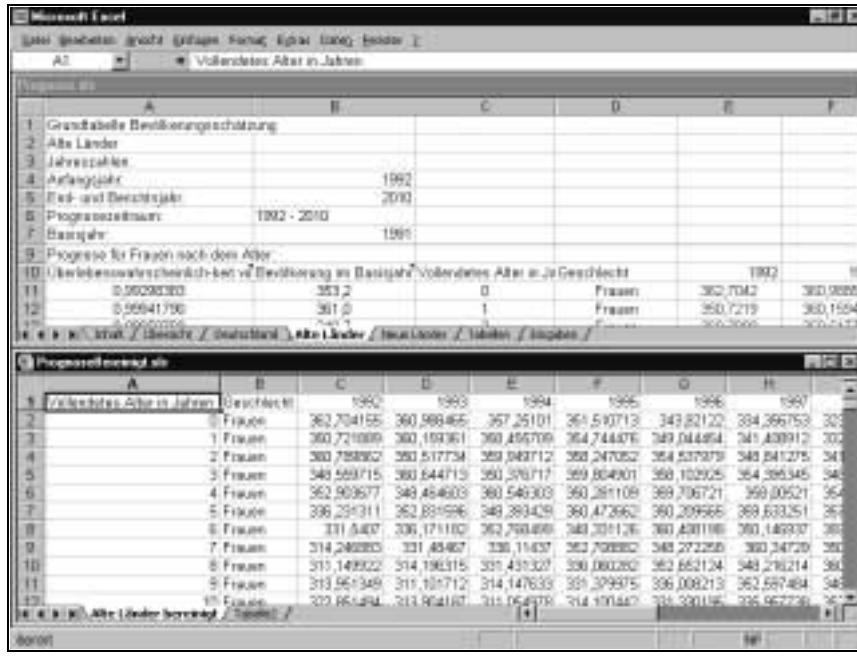


Bild 9.10: Die obere Excel-Tabelle enthält die unbereinigten Daten, die untere die bereinigten.

- Sämtliche Spalten tragen eine Überschrift, die beim Importieren als Feldnamen verwendet werden kann.
- Es existieren im zu importierenden Datenblock keine Leerzeilen.
- Jede Spalte enthält (außer der Überschrift) nur Daten desselben Datentyps, hier: Zahlen.
- Alle überflüssigen Spalten wurden entfernt.

Alte Länder bereinigt : Tabelle

ID	Vollendetes Alter in Jahren	Geschlecht	1992	1993	1994	1995
1	0	Frauen	362,70415511	360,98846537	357,25100995	351,51071296
2	1	Frauen	360,72188876	360,15936110	358,45570892	354,74447613
3	2	Frauen	360,7898619	360,51773354	359,94871234	358,24705186
4	3	Frauen	348,55971450	360,64471253	350,37671676	359,80490097
5	4	Frauen	352,90367689	348,46460301	360,54630341	350,28110946
6	5	Frauen	336,23131073	352,83159631	348,39342912	360,47266183
7	6	Frauen	331,54069997	336,17118248	352,76849944	348,33112592
8	7	Frauen	314,2468833	331,45466959	336,11436955	352,70888156
9	8	Frauen	311,1499217	314,19631489	331,43132708	336,06028203
10	9	Frauen	313,95134884	311,10171213	314,14763311	331,37997511
11	10	Frauen	322,85149396	313,90416707	311,05497843	314,10044186
12	11	Frauen	323,3529259	322,8044997	313,85849517	311,00970127
13	12	Frauen	322,85304711	323,30590715	322,7575607	313,81285701

Datensatz: ver. 102

Bild 9.11: Tabelle »Alte Länder bereinigt« in der Datenblattansicht: Die Feldnamen haben eine vernünftige Bedeutung.

Wenn die Excel-Tabelle *Alte Länder bereinigt* nach Access importiert wird, können die Spaltenüberschriften als Feldnamen übernommen werden. Wenn dann vom Import-Assistenten für Kalkulationstabellen ein Primärschlüssel hinzugefügt wird, sieht das Importergebnis in der neuen Access-Tabelle aus, wie in Bild 9.11 wiedergegeben. Daß auch die Datentypen korrekt festgelegt wurden, zeigt Bild 9.12, in dem dieselbe Tabelle in der Entwurfsansicht dargestellt ist.

Alte Länder bereinigt : Tabelle

Feldname	Felddatentyp
ID	AutoWert
Vollendetes Alter in Jahren	Zahl
Geschlecht	Text
1992	Zahl
1993	Zahl
1994	Zahl
1995	Zahl
1996	Zahl
1997	Zahl
1998	Zahl
1999	Zahl
2000	Zahl
2001	Zahl

Bild 9.12: Tabelle »Alte Länder bereinigt« in der Entwurfsansicht: Sämtliche Datentypen wurden vom Import-Assistenten für Kalkulationstabellen korrekt definiert.

9.2 Exportieren von Tabellen

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11.
- Aktivieren Sie die Objektliste *Tabellen*, und markieren Sie darin die zu exportierende Tabelle.
- Wählen Sie aus dem Menü *Datei* den Befehl *Exportieren*.
- Wählen Sie im Dialogfeld *Exportieren von Tabelle ... nach ...* die Option *In eine externe Datei oder Datenbank*, und bestätigen Sie mit OK.

- Wählen Sie im Dialogfeld *Exportieren von Tabelle ... nach ...* aus dem Listenfeld *Dateityp* das Format der Anwendung aus, in die Sie die Tabelle exportieren wollen. Wechseln Sie ggf. zu dem Laufwerk und/oder Ordner, in dem die Datei gespeichert werden soll, in die Sie die Tabelle exportieren. Geben Sie im Eingabefeld *Dateiname* den Namen der Datei an, in die die Tabelle exportiert werden soll, oder markieren Sie einen Dateinamen im Listenfeld, das die gefundenen Dateien des aktuellen Ordners anzeigt. Wenn Sie den Namen einer bestehenden Datei wählen, versucht der Export-Assistent, die exportierten Daten der Datei anzufügen. Im Falle einer Excel-Arbeitsmappe beispielsweise wird der Arbeitsmappe ein neues Tabellenblatt eingefügt, das den Namen der exportierten Access-Tabelle bekommt und die exportierten Daten enthält.

9.3 Verknüpfen von Tabellen

In Kap. 7, *Eine Datenbank konzipieren*, Punkt 7.2.1, *Verknüpfte Tabellen verwenden?*, wird besprochen, was unter einer verknüpften Tabelle zu verstehen ist und unter welchen Bedingungen es sinnvoll ist, mit verknüpften Tabellen zu arbeiten. An dieser Stelle wird die praktische Handhabung verknüpfter Tabellen erklärt. Als Beispiel dienen die beiden Datenbanken *ProjekteFrontend.mdb* und *ProjekteBackend.mdb*, die Sie auf der Begleit-CD-ROM finden. Die Datenbank *ProjekteBackend.mdb* enthält die beiden Tabellen *Personal* und *Projekte*, die aus der in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, entwickelten Datenbank stammen, vgl. dort. Für die beiden Tabellen ist eine 1:n-Beziehung definiert, mit der Tabelle *Personal* als Master- und *Projekte* als Detailtabelle. Die Datenbank ProjekteFrontend.mdb enthält, neben weiteren Datenbankobjekten, Formulare, die sich auf die beiden Tabellen *Personal* und *Projekte* als Datenherkunft beziehen, ohne daß diese beiden Tabellen physikalisch in der Datenbank *ProjekteFrontend.mdb* vorhanden sind. Daher müssen die Tabellen *Personal* und *Projekte* mit der Datenbank ProjekteFrontend.mdb verknüpft werden. Wie Sie dies bewerkstelligen können, wird im folgenden erklärt. Falls Sie das Beispiel nacharbeiten wollen, müssen Sie zunächst die in der Datenbank ProjekteFrontend.mdb bereits vorhandene Verknüpfung löschen, vgl. dazu ggf. weiter unten in diesem Kapitel, Punkt 9.3.4, *Verknüpfung einer Tabelle aufheben*.

9.3.1 Tabelle verknüpfen

Um die Tabellen *Personal* und *Projekte* aus der Datenbank ProjekteBakkend.mdb mit der Datenbank *ProjekteFrontend.mdb* zu verknüpfen, gehen Sie folgendermaßen vor:

- Öffnen Sie die Datenbank *ProjekteFrontend.mdb*.
- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11.

- Wählen Sie im Menü *Datei* den Befehl *Externe Daten* und dessen Unterbefehl *Tabellen verknüpfen...* .aus dem Menü *Datei,* oder klicken Sie auf die gleichnamige Symbol-Schaltfläche, vgl. links nebenstehend. (Diese Symbol-Schaltfläche ist standardmäßig nicht in der Datenbank-Symbolleiste enthalten, Sie müssen sie ggf. einfügen.)

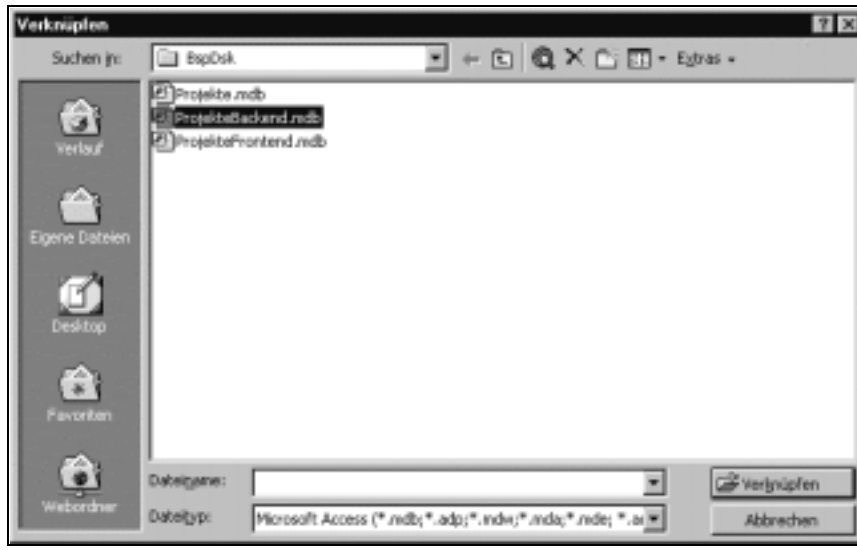


Bild 9.13: Dialogfeld »Verknüpfen«, mit dem Sie die Datei auswählen, welche die zu verknüpfende(n) Tabelle(n) enthält.

- Wählen Sie im Dialogfeld *Verknüpfen* (vgl. Bild 9.13) aus der Liste *Dateityp* das Format der einzubindenden Tabelle bzw. Datenbank, hier also Microsoft Access. Aus dem Angebot unterschiedlicher Formate können Sie erkennen, daß auch Tabellen anderer Formate verknüpft werden können. Dies gilt sogar für Textdateien, wenngleich für dieses Format inhaltlich einige Einschränkungen in Kauf zu nehmen sind, vgl. dazu genauer die Online-Hilfe, die bezüglich unterschiedlicher Formate verknüpfter Tabellen sehr ausführlich informiert. Wechseln Sie im Dialogfeld *Verknüpfen* ggf. zu dem Laufwerk und/oder Ordner, in dem sich die Datei mit den zu verknüpfenden Tabellen, in unserem Beispiel also die Datenbank *ProjekteBackend.mdb*, befindet. Markieren Sie dann die Datei, und bestätigen Sie mit der Schaltfläche *Verknüpfen*.
- Markieren Sie im folgenden Dialogfeld *Tabellen verknüpfen* die zu verknüpfende(n) Tabelle(n), im vorliegenden Beispiel also die beiden Tabellen *Personal* und *Projekte*, und bestätigen Sie mit OK.

Bild 9.14: Dialogfeld »Tabellen verknüpfen« zur Auswahl der zu verknüpfenden Tabellen

Bild 9.15: Die verknüpften Tabellen »Personal« und »Projekte« werden durch einen kleinen Pfeil als verknüpfte gekennzeichnet.

Wenn die Tabellen erfolgreich verknüpft wurden, werden sie von diesem Zeitpunkt an in der Objektliste *Tabellen* des Datenbankfensters zusammen mit den anderen Tabellen aufgeführt. Dabei werden verknüpfte Tabellen gegenüber nicht verknüpften durch einen kleinen Pfeil gekennzeichnet, vgl. Bild 9.14. Das Tabellensymbol verrät außerdem das Format der Quelle: Access-Tabellen werden, wie auch sonst, durch eine symbolisierte Tabelle angezeigt, dBase-Tabellen durch die Buchstaben dB, Paradox-Tabellen durch die Buchstaben Px usw., vgl. Bild 9.16.

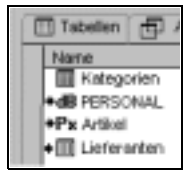


Bild 9.16: Verknüpfte Tabellen anderer Formate werden als solche gekennzeichnet. Diese Darstellung zeigt: Die erste Tabelle ist eine nicht verknüpfte Access-Tabelle, die zweite bis vierte sind verknüpft und haben die Formate dBase, Paradox und Access.

9.3.2 Eigenschaften verknüpfter Tabellen ändern

Prinzipiell läßt sich die Struktur einer verknüpften Tabelle nur in der Anwendung bzw. der Access-Datenbank ändern, der die Tabelle zugeordnet ist. Die meisten Feldeigenschaften, insbesondere Formate, Gültigkeitsregeln und Standardwerte, lassen sich jedoch auch in der Datenbank ändern, in die die Tabelle verknüpft ist. Solche Änderungen gelten dann nur für die Gastdatenbank, also diejenige, in die die Tabelle verknüpft ist, und wirken nicht auf die Ursprungstabelle zurück. Für eine Beziehung zweier verknüpfter Tabellen gilt Ähnliches wie für die Eigenschaften: Sie läßt sich vollständig nur in der Quelldatenbank, der die verknüpften Tabellen entstammen, bearbeiten. Jedoch können Sie die Eigenschaft *Mit referentieller Integrität* einer Beziehung auch im Fenster *Beziehungen* der Gastdatenbank bearbeiten, mit der Folge, daß die Änderung dann auch für die Quelldatenbank wirksam wird. Um andererseits beispielsweise eine Beziehung zweier verknüpfter Tabellen zu löschen, müßten Sie die Datenbank, in der sich diese Tabellen physikalisch befinden, öffnen und dann im Fenster *Beziehungen* das Löschen ausführen.

Um Eigenschaften einer verknüpften Tabelle in der Gastdatenbank zu ändern, verfahren Sie prinzipiell wie bei einer nicht verknüpften Tabelle. Um beispielsweise eine Eigenschaft der im obigen Beispiel verknüpften Tabelle *Personal* in der Datenbank *ProjekteFrontend.mdb* zu ändern, verfahren Sie wie folgt:

- Öffnen Sie ggf. die Datenbank ProjekteFrontend.mdb.

Bild 9.17: Meldung beim Öffnen einer verknüpften Tabelle in der Entwurfsansicht

- Öffnen Sie die Tabelle *Personal* in der Entwurfsansicht. Access macht Sie in einer Meldung darauf aufmerksam, daß bei einer verknüpften Tabelle Änderungen von Eigenschaften nur in beschränktem Umfang möglich sind, vgl. Bild 9.17. Bestätigen Sie die Meldung mit OK.
- Nehmen Sie die gewünschten Änderungen vor. Wenn Sie im Fenster der Entwurfsansicht dem Bearbeitungsfeld einer bestimmten Feldeigenschaft den Fokus geben, meldet Access – hervorgehoben durch rote Schriftart –, wenn diese Eigenschaft für eine verknüpfte Tabelle nicht verändert werden kann. Dies trifft beispielsweise für die Feldgröße zu, vgl. Bild 9.18.
- Speichern, und schließen Sie die Tabelle.

Bild 9.18: Access meldet, daß die Feldeigenschaft »Feldgröße«, die hier den Fokus hat, in einer verknüpften Tabelle nicht geändert werden kann.

9.3.3 Verknüpfte Tabelle umbenennen

Sie können einer verknüpften Tabelle einen anderen als ihren Ursprungsnamen geben. Verfahren Sie dabei in der üblichen Weise, Objekte in Access 2000 umzubenennen: Klicken Sie auf den Namen des Objekts, um diesen dann direkt zu bearbeiten. Das Umbenennen hat nur für die aktuelle Datenbank Bedeutung und läßt den Ursprungsnamen der Tabelle in der Quelle unverändert.

9.3.4 Verknüpfung einer Tabelle aufheben

Auf die folgende Weise entfernen Sie eine verknüpfte Tabelle aus der Datenbank:

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11.
- Aktivieren Sie die Objektliste *Tabellen*.
- Markieren Sie die Tabelle, für die Sie die Verknüpfung aufheben wollen.
- Drücken Sie die Taste Entf oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*. Keine Angst: Die Tabelle wird nicht physikalisch in der anderen Datenbank gelöscht, sondern es wird nur ihre Verknüpfung zur aktuellen Datenbank aufgehoben. Daß Access dieses erkennt, zeigt auch der Text des entsprechenden Meldungsdialogfeldes, vgl. Bild 9.18.

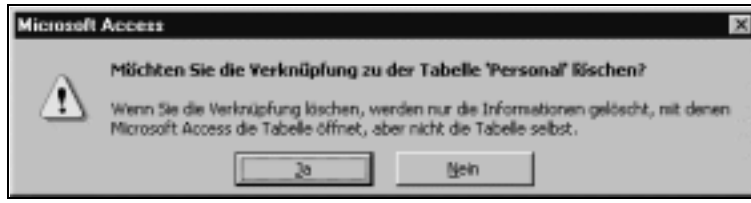


Bild 9.19: Um die Verknüpfung der Tabelle »Personal« aufzuheben, muß diese Meldung mit Ja bestätigt werden.

- Bestätigen Sie das Meldungsdialogfeld mit *Ja*.

9.3.5 Pfad einer verknüpften Tabelle

Verknüpfungsinformation einsehen

Access merkt sich den vollen Pfad einer verknüpften Tabelle, den Laufwerksbuchstaben eingeschlossen. Sie können diesen Pfadnamen einsehen, wenn Sie die verknüpfte Tabelle in der Entwurfsansicht öffnen und dort das Eigenschaftenfenster öffnen. In der Zeile *Beschreibung* des Eigenschaftenfensters wird der Pfad der Datenbank und der Quellname der verknüpften Tabelle wiedergegeben. Sie können diesen Eintrag nicht direkt ändern.

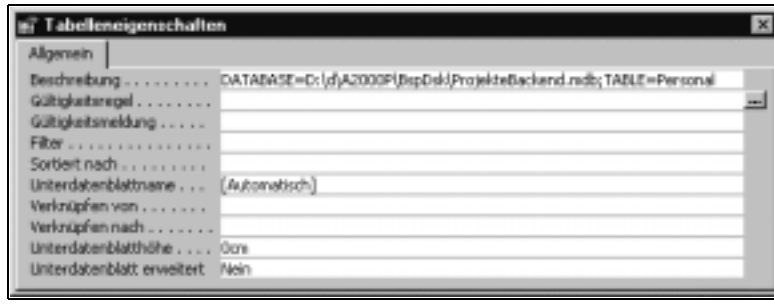


Bild 9.20: Die Tabelleneigenschaft »Beschreibung« gibt die Verknüpfungsinformation für die verknüpfte Tabelle an.

Der Umstand, daß sich die Gastdatenbank den Pfad einer verknüpften Tabelle einschließlich Laufwerksbuchstaben merkt, kann Probleme bereiten, wenn Sie die Datenbank in einer Netzwerkumgebung oder auf einem anderen PC öffnen, weil dann möglicherweise trotz gleicher Verzeichnisstruktur andere Laufwerksbuchstaben gelten. Auf Access-Ebene läßt sich dieses Problem nicht umgehen. Eine Lösung besteht darin, daß Sie im Netzwerk eine entsprechende Zuordnung des Verzeichnisnamens mit der verknüpften Tabelle zu dem Laufwerksbuchstaben vornehmen, unter dem die Datenbank sich den Pfad gemerkt hat. Im Netzwerk Novell beispielsweise würden Sie dies durch eine ROOT MAP-Anweisung erreichen. Auf einem PC können Sie sich des DOS-Befehls SUBST bedienen.

Verknüpfungsinformation manuell ändern

Wenn Sie den Namen, das Verzeichnis oder das Laufwerk der Quelldatenbank einer verknüpften Tabelle verändert haben, wird diese von der Gastdatenbank nicht automatisch angepaßt. Vielmehr würde Access bei dem Versuch, auf die verknüpfte Tabelle zuzugreifen, den Fehler melden, daß die Datei nicht gefunden werden könne. Daher müssen Sie in diesen oder ähnlichen Fällen der Gastdatenbank den neuen Namen bzw. Pfad mitteilen. Diesem Zweck dient der *Tabellenverknüpfungs-Manager.* Gehen Sie folgendermaßen vor, um für eine verknüpfte Tabelle eine geänderte Verknüpfungsinformation anzugeben:

- Wählen Sie aus dem Menü *Extras* den Befehl *Datenbank-Dienstprogramme*, Unterbefehl *Tabellenverknüpfungs-Manager*.

Bild 9.21: Dialogfeld »Tabellenverknüpfungs-Manager«, in dem Sie die verknüpfte(n) Tabelle(n) ankreuzen, für die Sie den Verknüpfungspfad ändern möchten.

- Das Dialogfeld *Tabellenverknüpfungs-Manager* (vgl. Bild 9.20) bietet Ihnen eine Liste aller verknüpften Tabellen an. Kreuzen Sie darin die Tabelle(n) an, deren Verknüpfungsinformation(en) geändert werden soll(en), und bestätigen Sie mit OK.
- Der *Tabellenverknüpfungs-Manager* versucht, den neuen Pfad selbst zu finden. Wenn ihm dieses gelingt, wird darüber eine entsprechende Meldung eingeblendet. Falls der neue Pfad nicht automatisch gefunden werden kann, müssen Sie in einem weiteren Dialogfeld zum richtigen Laufwerk und/oder Ordner wechseln und die Quelldatei markieren.

9.3.6 Verknüpfungspfad automatisch mit VBA-Prozedur ändern

Das im vorangehenden Punkt beschriebene Verfahren, den Verknüpfungspfad verknüpfter Tabellen mit Hilfe des Verknüpfungs-Managers zu ändern, ist einem durchschnittlichen Benutzer, für den Sie eine Datenbank entwikkeln, nicht zuzumuten. Statt dessen sollten Sie ein Verfahren vorsehen, das den Verknüpfungspfad automatisch ändert, wenn sich das Laufwerk und/oder der Ordner ändert, in dem die Datenbank, aus der Tabellen verknüpft sind, liegt. Die Beispieldatenbank *ProjekteFrontend.mdb* enthält eine VBA-Prozedur, welche dies leistet: Für den Fall, daß sich Gast- und Quelldatenbank (Front- und Back-End) vor und nach der Änderung des Pfades im selben Ordner befinden – dieser Fall dürfte die größte praktische Relevanz haben und könnte dem Benutzer vorgeschrieben werden – ändert die VBA-Funktion *AutoReattachTables*, die sich im Modul *Verknüpfungspfad* befindet, den alten in den neuen Verknüpfungspfad um. Dies ist möglich, weil

mit VBA ermittelt werden kann, aus welchem Ordner die aktuelle Datenbank, die ja die Gastdatenbank ist, stammt. Wenn der Aufruf der Funktion *AutoReattachTables* noch automatisch beim Öffnen der Datenbank durch das Makro *AutoExec* erfolgt – wie dies für die Datenbank *ProjekteFrontend.mdb* zutrifft –, ist der gesamte Prozeß der Pfadanpassung automatisiert.

9.3.7 Bearbeitungsgeschwindigkeit für verknüpfte Tabellen erhöhen

Verknüpfte Tabellen können von Access im allgemeinen nicht ganz so schnell verwaltet werden wie eigene Tabellen der Datenbank. Dies gilt insbesondere für Tabellen eines Netzwerks oder einer SQL-Datenbank. Der Grund liegt darin, daß die Bearbeitung verknüpfter Tabellen häufigere Dateizugriffe erforderlich macht, weil die verknüpfte Tabelle eine fremde Datei oder Teil einer solchen ist. Zur Minimierung der Bearbeitungszeiten beachten Sie die folgenden Hinweise:

- Blättern Sie nicht unnötig im Datenblatt oder Formular zwischen den Datensätzen einer verknüpften Tabelle. Vermeiden Sie vor allem, wenn dies möglich ist, große Sprünge (z.B. zum ersten oder zum letzten Datensatz), weil diese besonders viel Lesearbeit in der fremden Datei erforderlich machen.
- Wenn Sie effektiv formulierte Abfragen einsetzen statt der gesamten originären verknüpften Tabelle, können Sie oft die Anzahl der angezeigten Datensätze verringern. Dies reduziert die Bearbeitungszeit.
- In Abfragen, die verknüpfte Tabellen einbeziehen, sollten Sie keine Funktionen für die Abfragekriterien verwenden. Zeitraubend wirken sich vor allem Aggregatfunktionen aus (z.B. DomSumme, DomMax), weil diese voraussetzen, daß alle Datensätze der Tabelle verarbeitet werden.
- Falls Sie einer verknüpften Tabelle oftmals Datensätze hinzufügen wollen, ohne andere Datensätze dieser Tabelle bearbeiten zu müssen, können Sie ein Formular für die Dateneingabe erstellen. Stellen Sie die Eigenschaft *Standardbearbeitung* dieses Formulars auf *Nur Daten eingeben* ein. Dann werden beim Öffnen des Formulars keine Datensätze angezeigt, so daß auch keine gelesen werden müssen.

Kapitel 10

Abfragen: Grundlagen

10.1 Was ist eine Abfrage?

Eine *Abfrage* (engl. *Query*) stellt im allgemeinen Datensätze aus den Tabellen der Datenbank unter bestimmten Gesichtspunkten zusammen. Beispielsweise können Sie mit einer Abfrage alle Datensätze aus einer Tabelle oder mehreren Tabellen, die einem oder mehreren Kriterien entsprechen (z.B. alle Datensätze mit einem Auftragsvolumen >= 100.000 DM), zusammenstellen. Dieser Art von Datensammlung verdankt die Abfrage ihren Namen, weil Sie an die Tabellen eine bestimmte Frage stellen. Tatsächlich können Sie mit Abfragen noch mehr Aufgaben erledigen. Insgesamt können Sie in Access die folgenden Abfragearten ausführen:

- **Auswahlabfragen**. Bei dem im letzten Absatz skizzierten Beispiel handelt es sich um eine Auswahlabfrage. Bei dieser Abfrageart, die im allgemeinen am häufigsten verwendet wird, werden Daten aus bestehenden Tabellen oder anderen Abfragen gesammelt und als Abfrageergebnis angezeigt oder in anderer Form zur Verfügung gestellt. Das Abfrageergebnis, das Sie sich in der Datenblattansicht einer Abfrage betrachten können, sieht aus wie eine Tabelle, und Sie können es in vielerlei Hinsicht auch so benutzen. Der entscheidende Unterschied zwischen einer Tabelle und der Zusammenstellung von Datensätzen in einem Abfrageergebnis liegt darin, daß die Daten einer Tabelle gespeichert sind, während die Daten eines Abfrageergebnisses stets nur temporär gehalten werden. Das Abfrageergebnis verschwindet, wenn Sie die Abfrage schließen – spätestens jedoch, wenn Sie die aktuelle Datenbank schließen. Wenn Sie eine Abfrage speichern, wird stets nur ihre Definition, nicht jedoch ihr Abfrageergebnis gespeichert. Daher wird das Ergebnis einer Abfrage jedesmal neu ermittelt, wenn Sie die Abfrage öffnen oder ein Formular oder einen Bericht, das oder der darauf basiert, öffnen oder eine entsprechende Aktion ausführen. Auf diese Weise wird sicher gestellt, daß Abfrageergebnisse stets den aktuellsten Datenstand widerspiegeln.

Das Ergebnis einer Auswahlabfrage stellt Ihnen die Daten nicht nur passiv zum Lesen bereit, sondern Sie können darin auch Daten ändern und Datensätze ergänzen mit der Wirkung, daß die Änderungen an die zugrunde liegende(n) Tabelle(n) weitergegeben werden. (Die Aktualisierbarkeit der Datensätze im Abfrageergebnis von Auswahlabfragen unterliegt in bestimmten Zusammenstellungen Beschränkungen, vgl. dazu genauer Kap. 11, *Auswahlabfragen*, Punkt 11.6, *Datenaktualisierung bei Ein- und Mehrtabellenabfragen*.)

- **Kreuztabellenabfragen**. Eine Kreuztabelle stellt Daten in tabellarischer Form zweidimensional zusammen, so daß nicht nur die Spalten, sondern auch die Zeilen im Abfrageergebnis sachlich interpretierbare Überschriften haben. Beispiel: Sie haben eine Tabelle, welche u.a. die Felder *Höchster Schulabschluß* und *Geschlecht* enthält. Mit einer Kreuztabellenabfrage können Sie eine tabellarische Darstellung ausgeben lassen, die anzeigt, wie viele Frauen Hauptschulabschluß, wie viele Frauen Realschulabschluß ..., wie viele Männer Hauptschulabschluß, wie viele Männer Realschulabschluß ... haben. Außer Auszählungen können Sie auch andere Formen der Berechnung und Aggregation verwenden, beispielsweise Summen oder Mittelwerte bilden.
- **Aktionsabfragen**. Derartige Abfragen dienen dazu, Daten in Tabellen zu verändern, das Löschen ganzer Datensätze eingeschlossen. Dies erfolgt meistens unter Verwendung bestimmter Kriterien.
- **Union-Abfragen**. Union-Abfragen dienen dazu, Datensätze zweier oder mehrerer Tabellen senkrecht (statt waagerecht, wie bei einer »normalen« Join-Abfrage) in einem Abfrageergebnis zu vereinigen.
- **SQL Pass-Through-Abfragen**. Access kann mit bestimmten SQL-Servern zusammenarbeiten. Eine SQL Pass-Through-Abfrage wird an einen SQL-Server weitergeleitet, der die Abfrage dann auswertet.
- **Datendefinitionsabfragen**. Mit dieser Abfrageart können Sie mittels SQL-Anweisungen Tabellen erstellen, ihre Definitionen bearbeiten oder löschen. Anders ausgedrückt: Sie können die Arbeiten, die Sie sonst in der Entwurfsansicht einer Tabelle erledigen, auch mit einer SQL-Anweisung ausführen. Falls Sie beispielsweise über den SQL-Code für bestimmte Tabellendefinitionen verfügen (sei es aus einer SQL-Datenbank, sei es von einem Programm zum Datenbankdesign, sei es aus eigener Formulierung), können Sie diesen in Access zum Erstellen einer neuen Tabelle verwenden.

10.2 Wozu werden Abfragen verwendet?

Eine Abfrage dient dazu, Daten der Datenbank auf eine ganz bestimmte Weise, z.B. nach bestimmten Kriterien, zusammenzustellen. Am häufigsten werden Sie eine Abfrage für einen der folgenden Zwecke verwenden:

Felder beschränken

Wenn Sie sich eine Tabelle in der Datenblattansicht betrachten, werden stets alle Felder angezeigt. Für bestimmte Zwecke mögen viele davon überflüssig und störend sein. Mit einer Abfrage können Sie die Wiedergabe der Daten auf die benötigten Felder beschränken. Außerdem können Sie deren Anordnung bestimmen.

Felder berechnen

In einer Tabelle können Sie keinerlei Berechnungen ausführen. Vielmehr können in Tabellen nur konstante Werte gespeichert werden. Mit einer Abfrage lassen sich dagegen neue Felder definieren, die Ergebnisse von Berechnungen ausweisen. Beispielsweise können Sie ein Abfragefeld definieren, das die Summe bestimmter Einzelpositionen für jeden Datensatz ausweist. Berechnungen in diesem Sinne umfassen auch Operationen mit Texten. So könnten Sie beispielsweise in einem berechneten Feld einer Abfrage aus den Einzelfeldern *Vorname*, *Nachname*, *Straße*, *Postleitzahl* und *Ort* in einem berechneten Feld die Postanschrift komponieren, die den Vornamen und Nachnamen (getrennt durch ein Leerzeichen) in der ersten Zeile, dann einen Zeilenumbruch, dann die Straße in der zweiten Zeile, dann wiederum einen Zeilenumbruch und schließlich Postleitzahl, Leerzeichen und Ort in der dritten Zeile wiedergibt. Ebenso gut können Sie jedoch auch mathematische Ausdrücke zum Berechnen von Feldern verwenden, beispielsweise die Differenz zwischen dem Feld *Umsatz* und dem Feld *Kosten*.

Datensätze beschränken

Sie können Kriterien angeben, nach denen die Datensätze der Abfrage zusammengestellt werden. So läßt sich das Abfrageergebnis beispielsweise auf alle Adressen aus einem bestimmten Postleitzahlgebiet beschränken.

Daten aus mehreren Tabellen zusammenstellen

In einer relationalen Datenbank besteht eine der wesentlichen Aufgaben von Abfragen darin, Datensätze aus mehreren Tabellen zusammenzustellen. Wenn Sie beispielsweise die Datensätze für Kundenadressen, Aufträge und Rechnungen auf drei Tabellen verteilt haben, so können Sie eine bestimmte Zusammenstellung davon, z.B. alle Aufträge und unbezahlten Rechnungen eines bestimmten Kunden, mit einer Abfrage ausgeben lassen.

Datensätze sortiert ausgeben

Mit einer Abfrage lassen sich Datensätze leicht in sortierter Form (in auf- oder absteigender Reihenfolge, ggf. nach mehreren Feldern sortiert) ausgeben.

Aggregierte Zahlen für Gruppen von Datensätzen ermitteln

Beispiel: Aus einer Tabelle, in der die Tagesumsätze unter dem jeweiligen Datum festgehalten werden, lassen sich die jeweiligen Monatsumsätze ermitteln. Wenn diese Tabelle mit einer anderen Tabelle verknüpft ist, welche die Filialen der Firma enthält, können Sie auch aggregierte Ergebnisse je Filiale und Monat ermitteln lassen.

Datensätze anfügen

Sie können einer bestehenden Tabelle Datensätze anfügen, deren Inhalt Sie von einer Abfrage zusammenstellen (und dabei ggf. auch berechnen) lassen. Dabei ist es auch möglich, Datensätze einer Tabelle in einer anderen als der aktuellen Access-Datenbank anzufügen, also einer nicht geöffneten.

Datensätze aktualisieren

Wenn Sie die Werte eines oder mehrerer Felder einer Tabelle nach gewissen Regeln und/oder Kriterien verändern möchten, können Sie dies mittels einer Aktualisierungsabfrage erledigen. Beispielsweise kann es vorkommen, daß Sie den Lieferanten einer bestimmten Gruppe von Artikeln durch einen anderen Lieferanten ersetzen möchten. Diese Aufgabe lösen Sie mit einer Aktualisierungsabfrage.

Datensätze löschen

Mit einer Löschabfrage können Sie Datensätze nach bestimmten Kriterien löschen. So lassen sich mit einer Löschabfrage z.B. alle stornierten Aufträge mit einem Auftragsdatum aus dem vorangehenden Jahr löschen.

Neue Tabellen erstellen

Sie können das Ergebnis einer Abfrage als neue Tabelle speichern lassen. Diese Lösung ist beispielsweise sinnvoll, wenn Sie Abfrageergebnisse in andere Anwendungen exportieren wollen.

Datengrundlage für Formulare, Berichte und Steuerelemente

Vielfach ist es sinnvoll, ein Formular, einen Bericht oder ein Steuerelement – beispielsweise ein Listenfeld – auf einer Abfrage statt auf einer Tabelle zu basieren. Auf diese Weise können Sie die in einem Formular, Bericht oder Steuerelement verfügbaren Datensätze nach gewissen Kriterien einschränken oder sortieren, die Felder beschränken oder anordnen usw. Darüber hinaus lassen sich mit Abfragen verknüpfte Daten aus mehreren Tabellen zusammenstellen, an die ein Formular, ein Bericht oder ein Steuerelement sonst nicht gleichzeitig gebunden werden könnte.

Datengrundlage für andere Abfragen

Die Datenbasis einer Abfrage muß keine Tabelle, sondern kann auch eine bestehende andere Abfrage sein. Auf diese Weise lassen sich z.B. für komplexe Abfragen schnell weitere Einschränkungen formulieren, ohne die Struktur der bestehenden Abfrage verändern zu müssen.

10.3 Wie Sie Abfragen erstellen

Access stellt vier Assistenten zur Verfügung, mit denen Sie Abfragen erstellen können:

- Auswahlabfrage-Assistent
- Kreuztabellenabfrage-Assistent
- Abfrage-Assistent zur Duplikatsuche
- Abfrage-Assistent zur Inkonsistenzsuche

Die drei letztgenannten Abfrage-Assistenten leisten eine wirksame Unterstützung zur Formulierung der entsprechenden Abfragen, die bei freiem und selbständigem Entwurf nicht einfach zu bewältigen wären. Der Auswahlabfrage-Assistent unterstützt den Benutzer dagegen nicht mehr, als wenn er die Auswahlabfrage von vornherein in der Entwurfsansicht beginnt. Der letztere Weg unterstützt dagegen viel stärker konzeptionelles Denken für den Entwurf von Auswahlabfragen. Für auch nur etwas anspruchsvolleres Arbeiten ist es ohnehin erforderlich, daß Sie einen Abfrageentwurf in der Entwurfsansicht erstellen und bearbeiten können, sind doch Abfragen nach Tabellen das wichtigste Werkzeug einer Datenbank. Aus diesen Gründen erkläre ich den Weg zum Erstellen einer Abfrage nicht mit dem Auswahlabfrage-Assistenten, sondern wähle sofort den Weg über die Entwurfsansicht. Dabei beschränke ich mich in diesem Kapitel auf das Erstellen von Auswahlabfragen: Einerseits handelt es sich dabei um den am häufigsten verwendeten Abfragetyp, andererseits können Sie die wesentlichen Punkte, die für alle Arten von Abfragen gelten, an folgendem Beispiel (Bild 10.1) kennenlernen.

Ein praktisches Einführungsbeispiel zum Erstellen einer Abfrage finden Sie in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, Punkt 5.14, *Abfrage: Unerledigte Projekte mit einem Auftragswert von mindestens 100.000 DM*. An dieser Stelle soll nur kurz die prinzipielle Vorgehensweise zum Erstellen einer Abfrage aufgelistet werden.

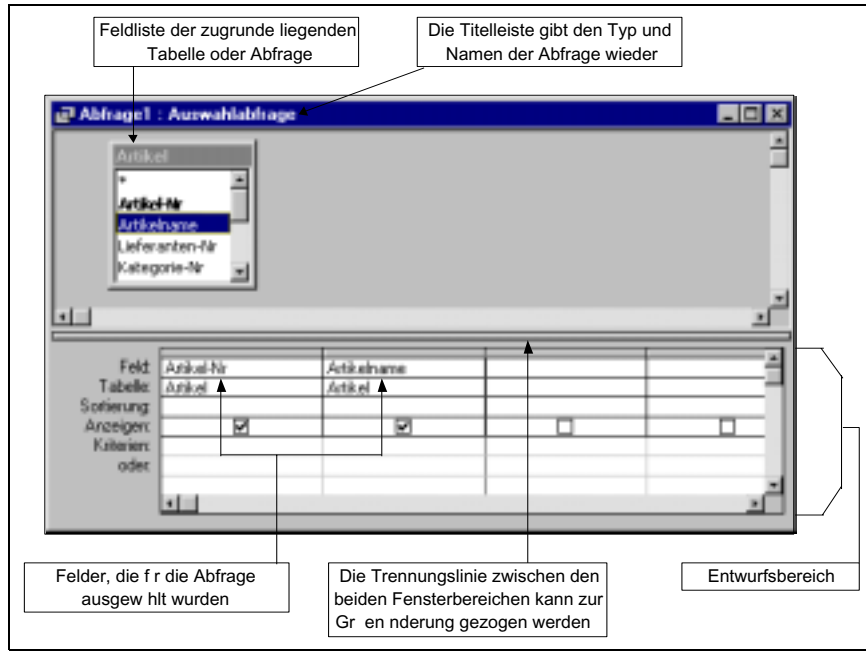


Bild 10.1: Abfragefenster in der Entwurfsansicht

Um eine Auswahlabfrage zu erstellen, gehen Sie folgendermaßen vor:

- Schlagen Sie in der Standard-Symbolleiste die Dropdown-Liste der Symbol-Schaltfläche *Neues Objekt* auf, und wählen Sie darin das Symbol *Abfrage*, vgl. links nebenstehend.

  Oder:

  Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11, aktivieren Sie die Objektliste *Abfragen*, und klicken Sie auf die Schaltfläche *Neu*.
- Wählen Sie im Dialogfeld *Neue Abfrage* die Option *Entwurfsansicht*.
- Access zeigt das Abfragefenster in der Entwurfsansicht an und gleichzeitig das Dialogfeld *Tabelle anzeigen* zur Auswahl der benötigten Tabellen bzw. Abfragen. Wählen Sie die Tabelle(n) und/oder Abfrage(n) aus, auf denen die neue Abfrage basieren soll, und schließen Sie das Dialogfeld *Tabelle anzeigen* anschließend.
- Wählen Sie die gewünschten Felder aus den Tabellen, und bestimmen Sie ggf. Kriterien, Funktionen, Sortierungen etc. Wie Sie einer Abfrage Felder, Kriterien, Funktionen, Sortierreihenfolgen etc. hinzufügen oder diese daraus entfernen, können Sie ausführlich in Kap. 11, *Auswahlabfragen*, nachlesen.

- Um die Abfrageergebnisse zu betrachten, wechseln Sie in die Datenblattansicht der Abfrage. Wählen Sie dazu in der Dropdown-Liste der Symbol-Schaltfläche *Ansicht* in der Standard-Symbolleiste das Symbol *Datenblattansicht* (vgl. links nebenstehend), oder wählen Sie den gleichnamigen Befehl aus dem Menü *Ansicht*.

- Wenn Sie die Definition der Abfrage speichern wollen, klicken Sie z.B. auf die Symbol-Schaltfläche *Speichern* (vgl. links nebenstehend) in der Standard-Symbolleiste, und geben Sie im Dialogfeld *Speichern unter* einen Namen für die Abfrage an. Dabei ist es gleichgültig, ob Sie das Speichern in der Entwurfs- oder der Datenblattansicht veranlassen, denn Access speichert ohnehin nur die Abfrage*definition* und niemals die Ergebnisse.
- Nach dem Speichern können Sie die Abfrage jederzeit wieder öffnen. Wenn Sie diese in der Datenblattansicht öffnen, sehen Sie die Abfrageergebnisse. In der Entwurfsansicht können Sie die Abfrage selbst verändern. Bei jedem neuen Öffnen einer Abfrage in der Datenblattansicht wird das Abfrageergebnis neu ermittelt, so daß es dann stets dem aktuellsten Datenstand entspricht. Entsprechendes gilt für das Öffnen von Formularen, Berichten und Steuerelementen, deren Datenherkunft eine Abfrage ist: Das Ergebnis der Abfrage wird jedesmal beim Öffnen eines derartigen Formulars oder Berichts neu ermittelt.

Wenn Sie eine Abfrage auf die hier angegebene Weise erstellen, also durch Einträge in die Zeilen und Spalten des Abfragefensters in der Entwurfsansicht, verfahren Sie nach der *QBE-Methode*. QBE ist die Abkürzung für *Query by Example* (Abfrage durch Beispiel).

10.4 Abfrageeigenschaften einstellen

Bild 10.2: Eigenschaftenfenster für eine Abfrage

Einige Merkmale einer Abfrage können Sie als deren Eigenschaften angeben, die Sie mit dem Eigenschaftenfenster (vgl. Bild 10.2) einstellen. Dieses öffnen Sie ggf. durch Klicken auf die Symbol-Schaltfläche *Eigenschaften* (vgl. links nebenstehend) oder durch Wahl des gleichnamigen Befehls aus dem Menü *Ansicht*. Das Eigenschaftenfenster kann sich auf eine Feldliste oder die gesamte Abfrage beziehen. Damit das Letztere zutrifft, klicken Sie auf einen beliebigen Punkt des Abfragefensters, jedoch nicht auf eine Feldliste. Die verschiedenen Abfrageeigenschaften haben die nachfolgenden Bedeutungen:

Beschreibung

Hier können Sie einen beliebigen Text zur Kennzeichnung der Abfrage eingeben.

Alle Felder ausgeben

Diese Eigenschaft können Sie auf *Nein* oder *Ja* einstellen. Im letzteren Falle werden in der Datenblattansicht alle Felder wiedergegeben, unabhängig davon, ob Sie dem Entwurfsbereich einzelne Felder hinzugefügt haben oder ob dieser leer ist.

Spitzenwerte

Mit dieser Eigenschaft legen Sie fest, ob alle Datensätze des Abfrageergebnisses oder nur ein Teil davon ausgegeben werden. Sie können das Schlüsselwort *Alle* angeben oder eine absolute oder eine Prozentzahl. Mit der Angabe einer Zahl beschränken Sie die Ausgabe auf die ersten n oder n% der insgesamt im Abfrageergebnis vorhandenen Datensätze. Bequemlichkeitshalber können Sie auch die Dropdown-Liste dieser Eigenschaft aufschlagen und einen Eintrag daraus wählen.

Keine Duplikate

Sie können diese Eigenschaft auf *Ja* einstellen, wenn Sie Datensätze ausschließen möchten, die mehrfach vorkommende Daten in den in der Datenblattansicht angezeigten Feldern enthalten. Enthält z.B. die Ausgabe einer Abfrage mehrere Felder, so *muß die Kombination der Werte aller Felder für einen gegebenen Datensatz eindeutig sein*, damit der Datensatz in die Ergebnismenge aufgenommen wird. Mit der Einstellung der Eigenschaft auf *Nein* – der Voreinstellung – werden alle Datensätze angezeigt.

Wenn Sie die Eigenschaft *Keine Duplikate* auf *Ja* eingestellt haben, dann können die Ergebnisse der Abfrage nicht aktualisiert werden. Zwischen den Eigenschaften *Keine Duplikate* und *Eindeutige Datensätze* (vgl. den folgenden Punkt) besteht ein Zusammenhang: Es kann immer nur eine der beiden

Eigenschaften auf *Ja* eingestellt werden. Stellen Sie z.B. die Eigenschaft *Keine Duplikate* auf *Ja* ein, so stellt Access die Eigenschaft *Eindeutige Datensätze* automatisch auf *Nein* ein. Sie können jedoch beide Eigenschaften auf *Nein* einstellen. Sind beide Eigenschaften auf Nein eingestellt, werden alle Datensätze zurückgegeben.

Eindeutige Datensätze

Mit der Eigenschaft *Eindeutige Datensätze* können Sie angeben, ob nur eindeutige Datensätze zurückgegeben werden sollen, die auf allen Feldern der zugrundeliegenden *Datenquelle* basieren, und nicht nur auf den in der Abfrage vorhandenen Feldern, wie dies mit der Eigenschaft *Keine Duplikate* bewirkt wird, vgl. den vorangehenden Punkt. Die Eigenschaft *Eindeutige Datensätze* ist nur wirksam, wenn Sie mehrere Tabellen in der Abfrage verwenden und Felder aus den in der Abfrage verwendeten Tabellen auswählen. Ihre Einstellung wird ignoriert, wenn die Abfrage nur eine Tabelle umfaßt.

Ausführungsberechtigungen

Sie können diese Eigenschaft sinnvollerweise in einer Mehrbenutzerumgebung mit einer geschützten Arbeitsgruppe verwenden, um die bestehenden Benutzerberechtigungen außer Kraft zu setzen. Dadurch können Sie eine Abfrage für Benutzer verfügbar machen, die andernfalls – d.h. aufgrund der ansonsten eingestellten Zugriffsrechte – keine Berechtigung dazu hätten. Die Eigenschaft kann auf *Eigentümer* oder *Benutzer* (voreingestellt) gesetzt werden. Es ergibt sich dann jeweils die folgende Wirkung:

Einstellung	Beschreibung
Besitzer	Alle Benutzer haben die Berechtigungen des Besitzers, um eine Abfrage anzuzeigen oder auszuführen.
Benutzer	Die Benutzer verfügen nur über ihre eigenen Berechtigungen, um eine Abfrage anzuzeigen oder auszuführen.

Quelldatenbank

Normalerweise müssen Tabellen oder Abfragen, die als Datenquelle für eine Abfrage dienen sollen, in der aktuellen Datenbank vorhanden oder mit dieser verknüpft sein. Mit der Eigenschaft *Quelldatenbank* können Sie diese Beschränkung umgehen, indem Sie den Pfad zu einer externen Datenbank angeben, deren Tabellen und Abfragen dann in der aktuellen Abfrage als Datenquelle zur Verfügung stehen. Wenn Sie beispielsweise den Pfad

D:\MSOffice2000\Office\Beispiel\Nordwind.mdb

angeben, können Sie in der aktuellen Abfrage als Datenquelle über die Tabellen und Abfragen der angegebenen Datenbank verfügen. Das Dialogfeld *Tabelle anzeigen*, mit dem Sie ja einer Abfrage weitere Tabellen oder Abfragen hinzufügen können, gibt dann ebenfalls die Tabellen und Abfragen der externen Datenbank wieder.

Wenn die externe Datenbank kein Microsoft Access-Format hat, müssen Sie auch die Eigenschaft *Quellverbindung* (vgl. den folgenden Punkt) einstellen, für eine Access-Datenbank reicht dagegen die Angabe zur Eigenschaft *Quelldatenbank* aus.

Quellverbindung

Für diese Eigenschaft müssen Sie eine Einstellung angeben, wenn Sie zur Eigenschaft *Quelldatenbank* eine Datenbank angegeben haben, die kein Microsoft Access-Format hat, z.B. dBase IV oder Paradox 3.0. Suchen Sie in der Online-Hilfe zum Stichwort *SourceConnectStr* (das ist der VBA-Begriff für den deutschen Begriff *Quellverbindung*), oder drücken Sie F1, während das Bearbeitungsfeld der Eigenschaft *Quellverbindung* den Fokus hat, um die für die verschiedenen Formate notwendigen Angaben zu erfahren.

Datensätze sperren

Diese Eigenschaft ist relevant für das Arbeiten in einer Mehrbenutzerumgebung. Sie beeinflußt die Art, in der Datensätze gesperrt werden, wenn mehrere Benutzer gleichzeitig auf die Abfrage oder ihr zugrunde liegende Tabellen zugreifen. Sie können zwischen den drei Einstellungen

- Keine Sperrungen
- Alle Datensätze
- Bearbeiteter Datensatz

wählen. Zur genauen Wirkung dieser drei Einstellungen vgl. die Online-Hilfe zum Stichwort *RecordLocks*.

Recordsettyp

Möglich sind die Einstellungen

- Dynaset
- Dynaset (Inkonsistente Aktualisierungen)
- Snapshot

Voreingestellt ist *Dynaset*. Mit *Snapshot* stellt das Abfrageergebnis lediglich eine passive Kopie der Datensätze dar, die nicht aktualisiert werden können. Mit den beiden *Dynaset*-Einstellungen sind Aktualisierungen möglich. Zur genaueren Wirkung vgl. die Online-Hilfe zum Stichwort *RecordsetType*.

ODBC-Wartezeit

Mit dieser Eigenschaft können Sie angeben, wie viele Sekunden Access wartet, bevor ein Zeitüberschreitungsfehler auftritt, wenn eine Abfrage einer ODBC-Datenbank ausgeführt wird. Als Einstellung geben Sie die Anzahl der zu wartenden Sekunden an. Die Voreinstellung beträgt 60 Sekunden. Ist die Eigenschaft auf 0 eingestellt, so tritt keine Zeitüberschreitung auf.

Filter

Mit der Eigenschaft *Filter* können Sie Kriterien zum Filtern der Datensätze angeben. Sie geben einen Zeichenfolgenausdruck wie eine WHERE-Klausel einer SQL-Anweisung ohne das Schlüsselwort WHERE an. Geben Sie beispielsweise für eine Abfrage, die das Feld *Artikel-Nr* enthält, den Filterausdruck

```
[Artikel-Nr]<=10
```

an. Dann werden nur Datensätze mit einer *Artikel-Nr* von <= 10 angezeigt.

Diese Eigenschaft wird auch von den verschiedenen Filter-Methoden eingestellt, die Sie in der Datenblattansicht der Abfrage, z.B. mit der Symbol-Schaltfläche *Auswahlbasierter Filter*, wählen können.

Sortiert nach

Geben Sie z.B. einen Feldnamen der Abfrage in eckigen Klammern an, um nach diesem Feld in aufsteigender Folge zu sortieren. Um absteigend zu sortieren, hängen Sie das reservierte Wort DESC an.

Diese Eigenschaft wird auch von Sortiermethoden eingestellt, die Sie in der Datenblattansicht der Abfrage, z.B. mit den Symbol-Schaltflächen *Aufsteigend* oder *Absteigend*, wählen können.

Max Datensätze

Mit der Eigenschaft *Max Datensätze* können Sie für eine Abfrage, die Daten aus einer ODBC-Datenbank abruft, ermitteln oder angeben, wie viele Datensätze sie maximal zurückgibt. Geben Sie zum Einstellen eine Ganzzahl an.

Mehrere der im Eigenschaftenfenster einstellbaren Abfrageeigenschaften schlagen sich auch in der SQL-Formulierung der Abfrage nieder und umgekehrt. Schlagen Sie die SQL-Ansicht des Abfragefensters auf (vgl. dazu in diesem Kapitel Punkt 10.6, *SQL*), um die jeweilige Auswirkung einer Einstellungsänderung zu überprüfen.

Eigenschaften zum Unterdatenblatt

Die letzten fünf Eigenschaften beziehen sich sämtlich auf ein mögliches Unterdatenblatt für die Datenblattansicht der Abfrage. Sie bedeuten im Einzelnen:

Unterdatenblattname	Mit einem Zeichenfolgenausdruck geben Sie den Namen der an das Unterdatenblatt gebundenen Tabelle oder Abfrage an.
Verknüpfen von	Geben Sie den Namen des Feldes aus der Tabelle/Abfrage des Unterdatenblatts an, von dem verknüpft werden soll.
Verknüpfen nach	Geben Sie den Namen des Feldes aus der Abfrage an, zu dem verknüpft werden soll. Der betreffende Feldname muß im Abfrageergebnis angezeigt werden!
Unterdatenblatthöhe	Mit dieser Eigenschaft wird die Anzeigehöhe des Unterdatenblattes angegeben.
Unterdatenblatt erweitert	Wenn diese Eigenschaft auf den Wert *Ja* eingestellt ist, werden die Unterdatenblätter zu allen Datensätzen der Mastertabelle eingeblendet, mit *Nein* werden nur die Erweiterungssymbole angezeigt.

10.5 Voreinstellungen für neuen Abfrageentwurf festlegen

Im unteren Teil der Registerkarte *Tabellen/Abfragen* des Dialogfeldes *Optionen* (gleichnamiger Befehl aus dem Menü *Extras*) finden Sie mehrere Optionen, deren Einstellungen beim Erstellen neuer Abfragen wirksam sind, vgl. Bild 10.3.

Die auf der Registerkarte *Tabellen/Abfragen* eingestellten Optionen zum Abfrageentwurf sind nur wirksam, wenn Sie eine neue Abfrage erstellen; auf bereits erzeugte Abfragen wirken sie sich nicht aus. Die vier Optionen bedeuten:

Tabellennamen anzeigen. Die Abfrage-Entwurfsansicht jeder neu erstellten Abfrage zeigt im Entwurfsbereich die Zeile *Tabelle*, in der die Namen der zu den jeweiligen Feldern gehörigen Tabellen/Abfragen ausgewiesen werden (sinnvoll bei Mehrtabellenabfragen, um gleichnamige Felder identifizieren zu können). Unabhängig von der Anfangseinstellung können Sie die Zeile *Tabelle* jederzeit mit dem Befehl *Tabellennamen* aus dem Menü *Ansicht* aus- oder einblenden.

Alle Felder ausgeben. Mit dieser Option wird für neue Abfragen voreingestellt, welche Einstellung die Abfrageeigenschaft *Alle Felder ausgeben* haben soll. Zur Bedeutung dieser Eigenschaft vgl. den vorangehenden Punkt 10.4, *Abfrageeigenschaften einstellen*, Unterpunkt *Alle Felder ausgeben*.

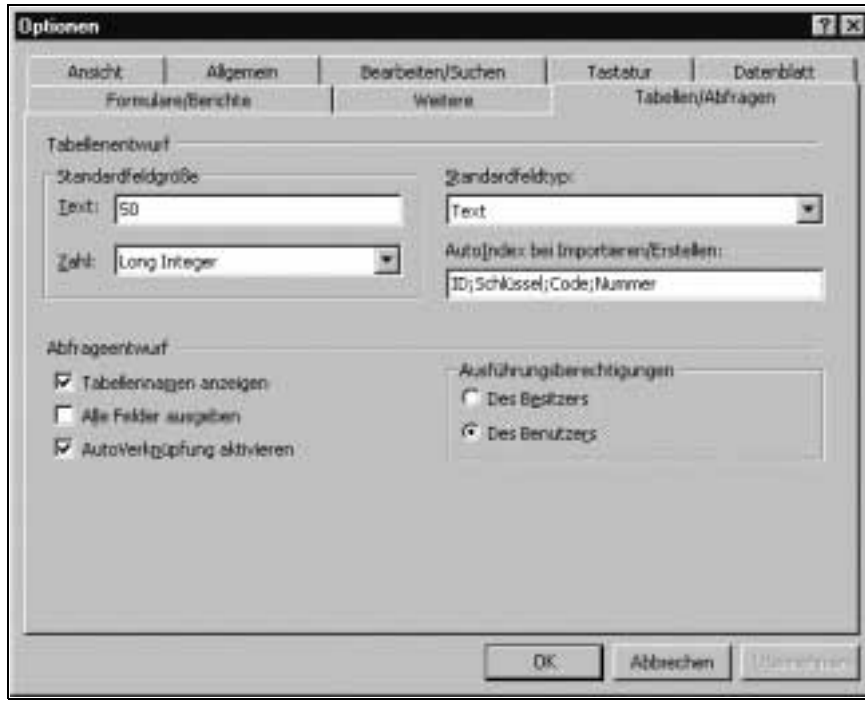


Bild 10.3: Registerkarte »Tabellen/Abfragen« mit Optionen für das Erstellen von Tabellen und Abfragen

AutoVerknüpfung aktivieren. Diese Eigenschaft ist nur für Abfragen wirksam, die auf mehr als einer Tabelle oder Abfrage basieren. Wenn sie angekreuzt ist, erstellt Access eine Verknüpfung zwischen den Feldern zweier Tabellen, die Sie der aktuellen Abfrage als Feldlisten hinzufügen. Eine derartige (Exklusions-)Verknüpfung kann allerdings nur erstellt werden, wenn in den beiden Tabellen Felder gleichen Namens mit kompatiblem Datentyp vorhanden sind. Sofern es sich um zwei Tabellen handelt, zwischen denen eine 1:1- oder eine 1:n-Beziehung definiert ist, verknüpft Access diese beiden Tabellen in der Abfrage unabhängig von der Einstellung der hier besprochenen Option *AutoVerknüpfung aktivieren*. Falls Sie die letztere Verknüpfung nicht wünschen, können Sie diese wie jede Verknüpfung im Abfragefenster löschen (markieren und Taste [Entf] drücken), ohne die zugrunde liegende Beziehungsdefinition zu beeinträchtigen.

Ausführungsberechtigungen. Mit dieser Option wird für neue Abfragen voreingestellt, welche Einstellung die Abfrageeigenschaft *Ausführungsberechtigungen* haben soll. Zur Bedeutung dieser Eigenschaft vgl. den vorangehenden Punkt 10.4, *Abfrageeigenschaften einstellen*, Unterpunkt *Ausführungsberechtigungen*.

10.6 SQL

SQL ist eine Abkürzung für *Structured Query Language* (= Strukturierte Abfragesprache). SQL stellt die in der Datenbankwelt übliche und normierte Abfragesprache dar. Jeder Abfrageformulierung, die Sie mit dem im letzten Punkt angegebenen Entwurfsverfahren vornehmen, entspricht eine ganz bestimmte SQL-Formulierung. Und auch das Umgekehrte gilt: Jeder SQL-Formulierung entspricht eine nach dem Beispiel-Entwurfsverfahren mögliche Abfrageformulierung. (Die letzte Aussage gilt allerdings nur mit gewissen Einschränkungen. Beispielsweise können Sie eine Datendefinitionsabfrage ausschließlich in SQL formulieren, sie läßt sich nicht im Abfrage-Entwurfsbereich nachvollziehen.) Aus diesem Grunde können beide Formulierungsarten jederzeit ineinander überführt werden, vgl. Bild 10.4 und Bild 10.5.

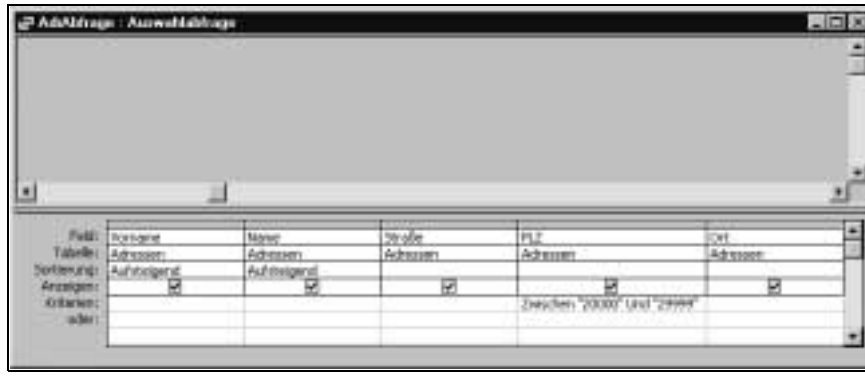


Bild 10.4: Abfrageentwurf im QBE-Bereich

Diese Möglichkeit sieht auch Access vor. Sie können die einem Abfrageentwurf entsprechende SQL-Formulierung für jeden beliebigen Formulierungsstand erfahren, indem Sie in der Dropdown-Liste der Symbol-Schaltfläche *Ansicht* in der Standard-Symbolleiste das Symbol *SQL* wählen oder den Befehl *SQL* aus dem Menü *Ansicht* wählen. Dann wird die SQL-Formulierung des momentanen Abfrageentwurfs als Text in einem Dialogfeld eingeblendet, vgl. Bild 10.5

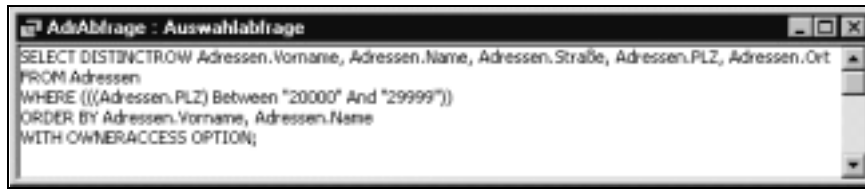


Bild 10.5: SQL-Ansicht der Abfrage aus Bild 10.4

Sie können diese SQL-Anweisung mit den üblichen Windows-Techniken (Markieren, Kopieren, Einfügen) an jede beliebige Stelle kopieren. Diese Möglichkeit erweist sich vor allem als nützlich, wenn Sie mit VBA programmieren, denn dort lassen sich verschiedene Aktionen am besten mit SQL-Anweisungen steuern. Aber auch für Formulare, Berichte und verschiedene Steuerelemente (z.B. bei einem Listen- oder Kombinationsfeld) können Sie als Datenquelle statt einer Tabelle oder Abfrage den SQL-Code einer Auswahlabfrage angeben. Auch das Umgekehrte gilt: Sie können eine SQL-Anweisung in die SQL-Ansicht einer Abfrage (vgl. Bild 10.5) schreiben oder kopieren. Wenn Sie dann wieder in die Entwurfsansicht der Abfrage wechseln, erscheint die SQL-Formulierung als entsprechender Abfrageentwurf.

In Kap. 14, *Einführung in SQL*, können Sie sich ausführlich über die verschiedenen Möglichkeiten, SQL-Abfragen zu formulieren, informieren.

10.7 Abfrage drucken

Abfrageergebnis drucken

Mit dem Befehl *Drucken* aus dem Menü *Datei* können Sie die Daten einer Abfrage auf dem Drucker ausgeben lassen. Vielleicht sollten Sie vorher jedoch die Seitenränder, das Ausgabeformat (Hoch- oder Querformat) etc. festlegen: Klicken Sie dazu auf die beiden Schaltflächen *Einrichten* bzw. *Eigenschaften* des Dialogfeldes *Drucken*, um die entsprechenden Dialogfelder zu öffnen und Einstellungen vorzunehmen. In jedem Falle sollten Sie sich die Daten, bevor Sie den Drucker aufrufen, in der Seitenansicht betrachten, um zu überprüfen, ob z.B. alle Beschriftungen wie gewünscht wiedergegeben werden und alle Spalten wie gewünscht auf dieselbe Seite passen.

Definition drucken

Sie können die Definition einer Abfrage auch drucken lassen. Rufen Sie dazu den Dokumentierer auf, indem Sie im Menü *Extras* den Befehl *Analyse*, Unterbefehl *Dokumentierer*, wählen. Mit der Schaltfläche *Optionen* im Dialogfeld *Dokumentierer* können Sie das Dialogfeld *Abfragedefinition drucken* aufrufen und bestimmen, welche Definitionsmerkmale ausgegeben werden sollen, vgl. Bild 10.6. Die Ausgabe erfolgt als Bericht zunächst auf dem Bildschirm. Wie üblich können Sie den Bericht jedoch auch drucken und/oder als Word für Windows-Datei ausgeben lassen.

Bild 10.6: Dialogfeld »Abfragedefinition drucken«

Kapitel 11

Auswahlabfragen

Wie Sie prinzipiell vorgehen, um eine neue Abfrage zu erstellen, wird in Kap. 10, *Abfragen: Grundlagen*, Punkt 10.3, *Wie Sie Abfragen erstellen*, beschrieben. In diesem Kapitel werden im Detail sämtliche Punkte behandelt, die das Erstellen und Bearbeiten von Auswahlabfragen betreffen.

11.1 Felder hinzufügen und bearbeiten

11.1.1 Felder hinzufügen

Sie können nur Felder solcher Tabellen oder Abfragen in den Abfrage-Entwurfsbereich aufnehmen, deren Feldlisten Sie zuvor in den oberen Teil des Abfragefensters aufgenommen haben. In das Abfragefenster von Bild 11.2 (vgl. weiter unten) beispielsweise wurden die Feldlisten der Tabellen *Personal* und *Projekte* aufgenommen. Sofern in das Abfragefenster keine weiteren Feldlisten aufgenommen werden, können Sie in der Abfrage nur über die Felder der Tabellen *Personal* und *Projekte* verfügen. Allerdings können Sie einer Abfrage jederzeit weitere Feldlisten hinzufügen.

Feldliste einer Tabelle oder Abfrage hinzufügen

- Öffnen Sie ggf. die zu bearbeitende Abfrage in der Entwurfsansicht.
- Klicken Sie auf die Symbol-Schaltfläche *Tabelle anzeigen* (vgl. links nebenstehend), oder wählen Sie den gleichnamigen Befehl aus dem Menü *Abfrage*. Dann zeigt sich das Dialogfeld *Tabelle anzeigen*, vgl. Bild 11.1.
- Im Dialogfeld *Tabelle anzeigen* können Sie gezielt nur Tabellen, nur Abfragen oder beide Arten auflisten lassen, vgl. die drei Registerkarten. Das Listenfeld zeigt, je nach aktivierter Registerkarte, die Tabellen und/oder Abfragen an, die in der aktuellen Datenbank bisher definiert sind. Markieren Sie in der Liste die Tabelle oder Abfrage, auf der die in

Bearbeitung befindliche Abfrage basieren soll, und klicken Sie auf die Schaltfläche *Hinzufügen*. Alternativ: Doppelklicken Sie auf die Tabelle oder Abfrage im Listenfeld.

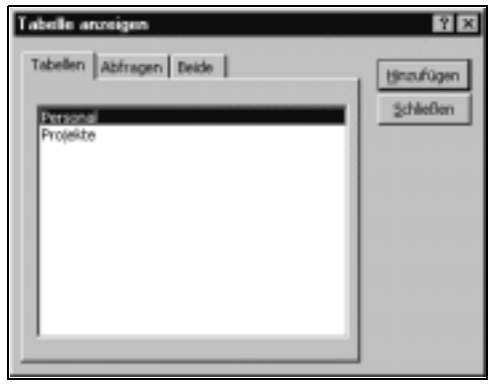


Bild 11.1: Dialogfeld »Tabelle anzeigen«

- Wiederholen Sie diesen Schritt für jede hinzuzufügende Tabelle oder Abfrage. Wenn Sie dem Abfrage-Entwurf mehr als eine Tabelle oder Abfrage hinzufügen wollen, können Sie im Listenfeld des Dialogfeldes *Tabelle anzeigen* auch mehrere Einträge gleichzeitig markieren: Wie unter Windows üblich markieren Sie zusammenliegende Einträge mit gedrückter ⇧-Taste und auseinanderliegende mit jeweils gedrückter Strg-Taste und Klicken mit der Maus.
- Schließen Sie das Dialogfeld *Tabelle anzeigen*.

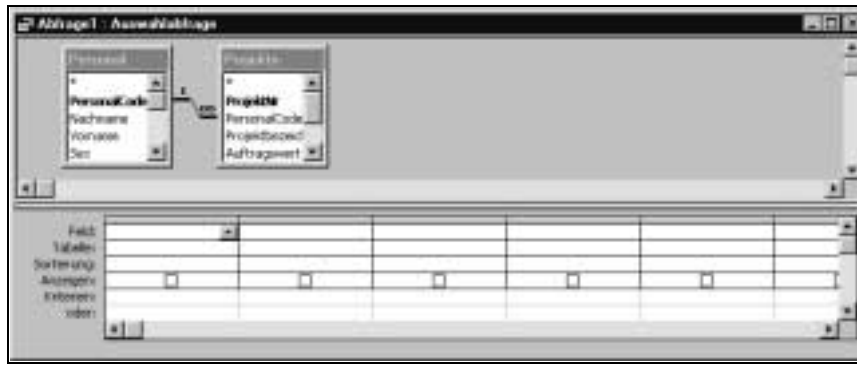


Bild 11.2: Abfrageentwurf, dem die beiden Tabellen »Personal« und »Projekte«, für den jedoch noch keine Felder hinzugefügt wurden.

Wenn zwischen den hinzugefügten Tabellen eine Beziehung definiert ist, wird diese Beziehung im Abfragefenster durch eine Linie zwischen den verknüpften Feldern dargestellt, was bedeutet, daß diese Verknüpfung auch

inhaltlich in der Abfrage wirksam ist. Eine Verknüpfung zwischen zwei Tabellen erstellt Access auch dann, wenn keine Beziehung zwischen diesen definiert ist, jedoch die folgenden Bedingungen zutreffen: In der Registerkarte *Tabellen/Abfragen* des Befehls *Optionen* aus dem Menü *Extras* ist die Option *AutoVerknüpfung aktivieren* angekreuzt, und die beiden Tabellen haben Felder mit gleichen Namen und kompatiblem Datentyp.

Sie können die Verknüpfung zwischen den Tabellen einer Abfrage löschen; eine etwa definierte Beziehung zwischen den Tabellen bleibt davon unberührt, sie bleibt weiter bestehen. Um die Verknüpfung zu löschen, markieren Sie die Verknüpfungslinie, indem Sie darauf klicken. Drücken Sie dann die [Entf]-Taste.

Einzelnes Feld oder mehrere Felder hinzufügen

Um einer Abfrage ein einzelnes Feld oder mehrere Felder hinzuzufügen, gehen Sie folgendermaßen vor:

- Ziehen Sie in der Entwurfsansicht der Abfrage das hinzuzufügende Feld aus der betreffenden Feldliste in eine Zelle der Zeile *Feld* im Entwurfsbereich. Um mehrere Felder gleichzeitig hinzuzufügen, markieren Sie diese zuvor in der Feldliste mit Maus und gedrückter [Strg]- oder [⇧]-Taste, und ziehen dann. Wenn Sie auf ein Feld der Feldliste doppelklicken, wird es ebenfalls dem Entwurfsbereich hinzugefügt.

  Oder:

  Klicken Sie in eine Zelle der Zeile *Feld* im Entwurfsbereich, schlagen Sie das dann angebotene Dropdown-Listenfeld mit den Feldnamen auf, und wählen Sie darin das gewünschte Feld, vgl. Bild 11.3. Sie dürfen den Feldnamen auch schreiben, statt ihn aus der Liste auszuwählen.

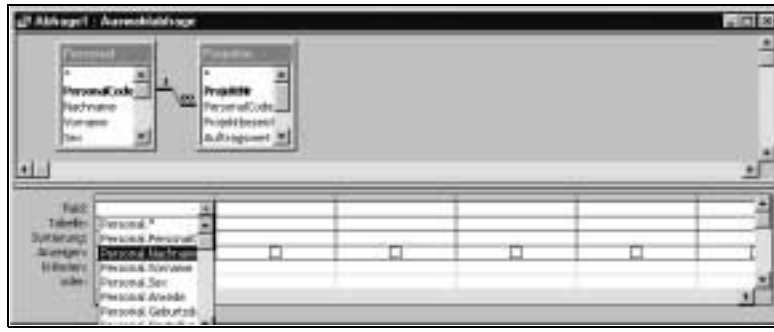


Bild 11.3: Aufgeschlagenes Dropdown-Listenfeld mit den Feldnamen der im Abfrageentwurf vorhandenen Feldlisten

Wenn Sie einen Feldnamen durch Tippen seiner Zeichenfolge eingeben und sich dabei verschreiben, meldet Access dies nicht als einen Fehler. Sie werden auf den Irrtum erst aufmerksam gemacht, wenn Sie das Abfrageergebnis

anfordern, z.B. durch Wechsel in die Datenblattansicht. Dann zeigt Access das Dialogfeld *Parameterwerte eingeben* an, mit dem Sie zur Eingabe eines Parameterwertes aufgefordert werden. Dies hat seine Bewandtnis darin, daß Access im Abfrage-Entwurfsbereich jede Zeichenfolge, die dem System weder als reserviertes Wort noch als Feldname bekannt ist, als benutzerdefinierten Parameter interpretiert. (Zur genaueren Bedeutung von Parametern in Abfragen vgl. Kap. 12, *Komplexe Abfragen*.)

Falls ein Feldname mit einem Doppelpunkt endet, umschließen Sie ihn nach der Aufnahme in die Zeile *Feld* mit eckigen Klammern, weil er von der Abfrage sonst nicht als Feldname, sondern als Benennung eines solchen interpretiert würde.

Alle Felder hinzufügen

Access bietet drei Verfahren an, einer Abfrage alle Felder einer Feldliste hinzuzufügen. Beim ersten Verfahren wird im Ergebnis jedes einzelne Feld namentlich hinzugefügt. Das zweite Verfahren verwendet das Platzhalterzeichen *.

Alle Felder namentlich hinzufügen:

- Doppelklicken Sie auf die Titelleiste der Feldliste. Daraufhin ist die gesamte Feldliste markiert.
- Ziehen Sie die markierte Feldliste in den Entwurfsbereich.

Alle Felder mittels Platzhaltersymbol Sternchen (*) hinzufügen:

- Ziehen Sie das Sternchen aus der Feldliste in den Entwurfsbereich. Dann erscheint in der Zeile *Feld* das Sternchen mit vorangestelltem Tabellennamen, getrennt durch einen Punkt, vgl. Bild 11.4.

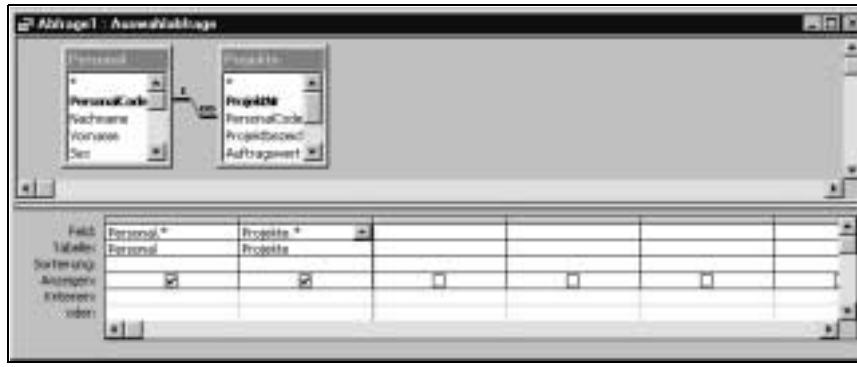


Bild 11.4: Wenn Sie bei dieser Abfrage in die Datenblattansicht wechseln, werden alle Felder der Tabellen »Personal« und »Projekte« angezeigt, denn das Platzhaltersymbol Sternchen (*) steht jeweils für alle Felder der Tabelle.

Abfrageeigenschaft »Alle Felder ausgeben« auf Ja einstellen:

- Stellen Sie diese Abfrageeigenschaft im Eigenschaftenfenster des Abfrageentwurfs auf *Ja* ein. Dann werden alle Felder im Abfrageergebnis angezeigt. (Zur Bedeutung und Einstellung von Abfrageeigenschaften vgl. genauer Kap 10, *Abfragen: Grundlagen*, Punkt 10.4, *Abfrageeigenschaften einstellen*.)

Die drei Verfahren haben im Ergebnis eine etwas unterschiedliche Wirkung: Der Vorteil des Sternchens und der Einstellung der Abfrageeigenschaft *Alle Felder ausgeben* auf den Wert *Ja* besteht darin, daß dies späteren Veränderungen der Feldliste (z.B. Namensänderungen, Hinzufügen oder Löschen von Feldern in der zugrundeliegenden Tabelle/Abfrage) automatisch Rechnung trägt: Wenn Sie beispielsweise einer Tabelle nachträglich Felder hinzufügen oder Felder daraus entfernen, berücksichtigt eine Abfrage, der die Felder dieser Tabelle unter Verwendung des Sternchens oder mittels der Abfrageeigenschaft *Alle Felder ausgeben* hinzugefügt wurden, die neue Feldzusammenstellung automatisch. Diese Wirkung spricht für die Verwendung dieser beiden Verfahren statt des namentlichen Anführens der Feldnamen. Allerdings können Sie Felder, die lediglich durch das Sternchen oder mittels der Abfrageeigenschaft *Alle Felder ausgeben* repräsentiert sind, nicht als Kriterien oder zum Sortieren verwenden, weil die Felder zu diesem Zweck im Entwurfsbereich explizit angeführt sein müssen. Diese kleine Klippe läßt sich jedoch auf einfache Weise umgehen, indem Sie einzelne namentliche Felder zusätzlich hinzufügen und diese dann von der Anzeige ausschließen; vgl. im einzelnen weiter unten in diesem Abschnitt, Punkt 11.1.5, *Felder ausblenden*. Beachten Sie darüber hinaus, daß sich aggregierte Berechnungen mit den Aggregatfunktionen nur ausführen lassen, wenn der Entwurfsbereich kein Sternchen enthält.

11.1.2 Felder verschieben, einfügen und löschen

Felder verschieben

Sie können die Feldreihenfolge, die auch die Reihenfolge im Abfrageergebnis bestimmt, nachträglich verändern, indem Sie einzelne oder mehrere Felder im Entwurfsbereich verschieben. Verfahren Sie dazu folgendermaßen:

- Markieren Sie ein Feld oder mehrere nebeneinanderliegende, falls Sie mehrere verschieben wollen. Zum Markieren klicken Sie auf den Spaltenkopf und, für mehrere Felder, ziehen Sie dann.
- Klicken Sie erneut auf den Spaltenkopf des markierten Feldes bzw. der markierten Felder, und ziehen Sie an die gewünschte Position, vgl. Bild 11.5.

Bild 11.5: Das Feld »Vorname« wird vor das Feld »Nachname« geschoben, wenn die linke Maustaste in dieser Situation losgelassen wird.

Felder einfügen

Zum Einfügen eines Feldes oder mehrerer Felder verfahren Sie wie beim Hinzufügen von Feldern: Ziehen Sie in diesem Falle jedoch auf die Spalte eines bestehenden Feldes. Dann wird das neue Feld an der Stelle des alten eingefügt, und die bereits vorhandenen Felder werden von dieser Stelle an nach rechts verschoben.

Felder löschen

Einzelne Felder löschen:

- Markieren Sie ein Feld oder mehrere nebeneinander liegende, falls Sie mehrere löschen wollen. Zum Markieren klicken Sie auf den Spaltenkopf und, für mehrere Felder, ziehen Sie dann.
- Drücken Sie die [Entf]-Taste, oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*.

Alle Felder einer Feldliste löschen:

- Falls Sie alle Felder einer Tabelle/Abfrage aus der in Bearbeitung befindlichen Abfrage entfernen wollen, können Sie auch einfach die Feldliste im oberen Teil des Abfragefensters entfernen, denn damit werden automatisch alle zu ihr gehörenden Felder aus der Abfrage entfernt. Sie entfernen eine Feldliste, indem Sie diese durch Klicken auf einen beliebigen Punkt markieren und dann die [Entf]-Taste drücken. Anschließend können Sie die entfernte Feldliste der Abfrage ggf. wieder hinzufügen, vgl. oben, Punkt 11.1.1, *Felder hinzufügen*.

Alle Felder der Abfrage löschen:

- Zum Löschen aller Felder der Abfrage wählen Sie den Befehl *Alles löschen* aus dem Menü *Bearbeiten*. Damit werden zwar alle Felder aus dem Entwurfsbereich entfernt, die Feldlisten verbleiben jedoch im oberen Teil des Abfragefensters.

11.1.3 Felder benennen

Wenn Sie nichts weiteres veranlassen, werden die Feldnamen der Tabelle(n) oder Abfrage(n), auf der oder denen die Abfrage basiert, als Feldnamen der neuen Abfrage und als deren Spaltenüberschriften für die Datenblattansicht übernommen. Sie können aber jedem Feld für das Abfrageergebnis einen anderen Namen geben. Dann erscheint dieser andere Name beispielsweise im Datenblatt der Abfrage, aber z.B. auch in der Feldliste eines Formulars, das auf dieser Abfrage basiert. Der Formular-Assistent verwendet ihn ebenfalls. Wenn Sie sich dann in einem Ausdruck auf das umbenannte Feld der Abfrage beziehen, müssen Sie den neuen Feldnamen anführen. Um ein Feld zu benennen, gehen Sie folgendermaßen vor:

- Schreiben Sie den neuen Namen vor den Feldnamen in die entsprechende Zelle der Zeile *Feld*, und schließen Sie den neuen Namen mit einem Doppelpunkt ab. Wenn Sie beispielsweise den Feldern *Nachname* und *Sex* die neuen Namen *Familienname* und *Geschlecht* geben wollen, müssen die Zellen dieser Felder in der Zeile *Feld* des Entwurfsbereichs so beschriftet sein, wie dies in Bild 11.6 wiedergegeben ist.

Bild 11.6: Die Felder »Nachname« und »Sex« wurden mit »Familienname« und »Geschlecht« benannt.

11.1.4 Felder mit Tabellennamen anzeigen

Falls eine Abfrage auf mehr als einer Tabelle/Abfrage basiert und in den beteiligten Tabellen/Abfragen namensgleiche Felder vorhanden sind, könnten Sie diese nicht mehr unterscheiden, wenn sie mit ihren puren Namen in den Entwurfsbereich aufgenommen würden. In solchen Fällen verfährt Access folgendermaßen: Dem Feldnamen wird der Name der Tabelle/Abfrage und ein Trennungspunkt vorangestellt. Das Feld *PersonalCode* aus der Tabelle *Personal* beispielsweise würde dann als *Personal.PersonalCode* angegeben werden.

Sie können aber auch unabhängig von solchen Benennungskonflikten im Entwurfsbereich für jedes Feld seine Herkunft anzeigen lassen:

- Wählen Sie den Befehl *Tabellennamen* aus dem Menü *Ansicht*. Dann erscheint unterhalb der Feldzeile im Entwurfsbereich zusätzlich die Zeile *Tabelle*, in deren Zellen der jeweilige Name der Tabelle/Abfrage, aus der das Feld stammt, angegeben wird.

11.1.5 Felder ausblenden

Es ist möglich, Felder in die Abfrage aufzunehmen, ohne ihre Daten in das Abfrageergebnis zu übernehmen. Dies ergibt z.B. einen Sinn, wenn Sie nach einem Feld sortieren lassen oder es als Kriterium verwenden wollen, ohne seine Daten anzeigen zu lassen. Sie schließen ein Feld von der Aufnahme in das Abfrageergebnis folgendermaßen aus oder nehmen es wieder mit herein:

- Schalten Sie das Kontrollkästchen in der Zeile *Anzeigen* durch Klicken aus oder ein. Im Beispiel von Bild 11.7 wird das Feld *Projektbeginn* nicht in das Abfrageergebnis aufgenommen, obgleich nach seinem Inhalt sortiert wird. Alle anderen Felder sind im Abfrageergebnis vorhanden und werden somit auch in der Datenblattansicht der Abfrage angezeigt.

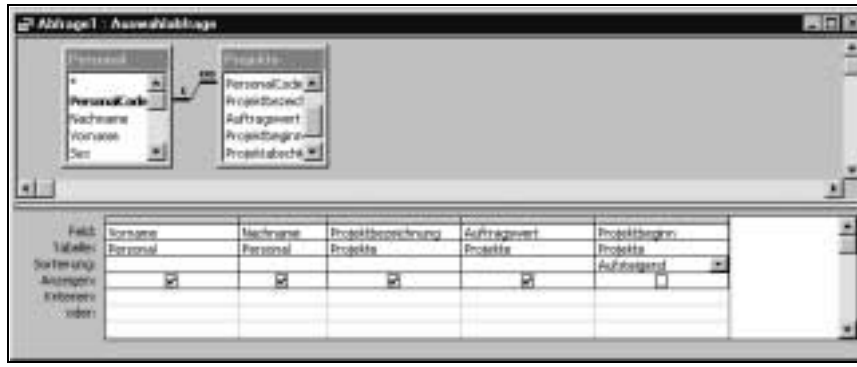


Bild 11.7: Das Feld »Projektbeginn« wird nicht im Abfrageergebnis angezeigt, weil sein Kontrollkästchen »Anzeigen« ausgeschaltet ist. Gleichwohl wird nach diesem Feld sortiert.

Besonders sinnvoll ist das Ausschließen einzelner Felder von der Anzeige im Datenblatt, wenn Sie der Abfrage alle Felder einer Feldliste (z.B. mittels des Sternchens *) hinzugefügt haben und trotzdem nach bestimmten Feldern dieser Feldliste sortieren und/oder Kriterien dafür angeben möchten. Für diese Zwecke müssen die betreffenden Felder im Entwurfsbereich explizit angeführt sein. Dies ist auch möglich, denn Sie können einer Abfrage einzelne Felder zusätzlich zum Sternchen hinzufügen und dafür dann eine Sortierung und/oder Kriterien angeben. Wenn Sie für diese Felder jedoch das Kontrollkästchen *Anzeigen* eingeschaltet lassen würden, würden sie doppelt angezeigt werden. Aus diesem Grunde sollten Sie das Kontrollkästchen *Anzeigen* in derartigen Fällen ausschalten.

11.1.6 Berechnete Felder erstellen

Anders als Tabellen können Abfragen auch Felder enthalten, in denen Berechnungen ausgeführt werden. Die Berechnung basiert auf einem Ausdruck, den Sie in die Zeile *Feld* eingeben. Beispielsweise könnten Sie den Mehrwertsteuerbetrag auf diese Weise in einem Feld ermitteln lassen. Wie jedes Feld muß auch ein berechnetes Feld einen Namen haben, damit man sich darauf beziehen kann. Wenn Sie keinen Namen angeben, vergibt Access automatisch die Namen *Ausdr1*, *Ausdr2* etc. Sie vergeben selbst einen Namen in der Weise, wie dies oben im Punkt 11.1.3, *Felder benennen*, beschrieben wurde: Führen Sie vor dem Ausdruck einen Namen an, den Sie mit einem Doppelpunkt (:) enden lassen.

Beispielhafte Vorgehensweise

Ermitteln Sie die Mehrwertsteuer, die sich für den *Auftragswert* (dieser ist netto zu verstehen) ergibt. Dabei wird im ersten Schritt kein Name vergeben. Dann wird gezeigt, wie Sie den von Access vergebenen Namen in einen von Ihnen definierten ändern. Gehen Sie im einzelnen folgendermaßen vor:

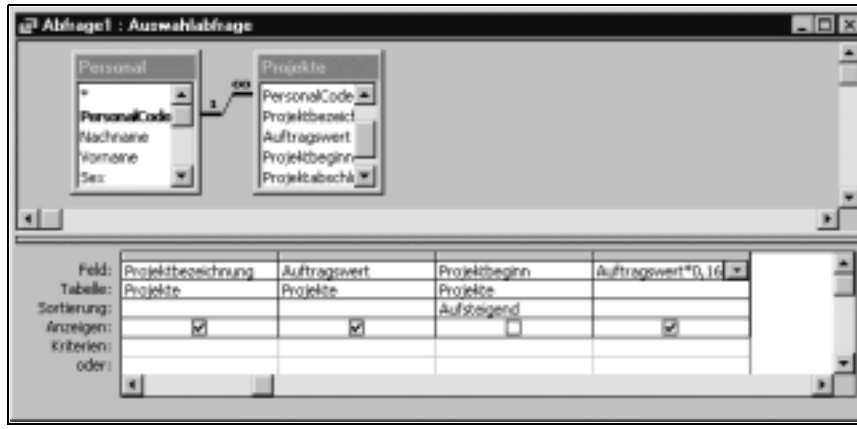


Bild 11.8: In die ganz rechts sichtbare Spalte wurde der Ausdruck »Auftragswert*0, 16« eingegeben, das Feld hat jedoch noch den Fokus, so daß Access die Eingabe noch nicht angenommen hat. Die Umwandlung dieser Eingabe nach deren Akzeptanz sehen Sie in Bild 11.9.

- Geben Sie in eine Zelle der Zeile *Feld* den Ausdruck ein, mit dem die Berechnung durchgeführt werden soll, im genannten Beispiel also den Ausdruck

  Auftragswert*0,16

  Diese Situation ist in Bild 11.8 dargestellt.

- Klicken Sie auf eine andere Zelle, um prüfen zu lassen, ob Ihr Ausdruck Syntaxfehler enthält. Falls dies zutrifft, korrigieren Sie den Ausdruck, bis Access ihn akzeptiert, oder brechen Sie mit Esc ab. Wenn der Ausdruck von Access angenommen wurde, wird er in der Form dargestellt, wie dies in Bild 11.9 wiedergegeben ist: Dem Ausdruck *Auftragswert*0, 16* wurde die Zeichenfolge *Ausdr1:* vorangestellt. Damit hat Access dem Feld den Namen *Ausdr1* gegeben. (Zum Benennen von Feldern in Abfragen vgl. oben Punkt 11.1.3, *Felder benennen*.) Zugleich hat Access den Feldnamen *Auftragswert* zwischen eckige Klammern gesetzt, wie dies bei Ausdrücken üblicherweise geschieht.

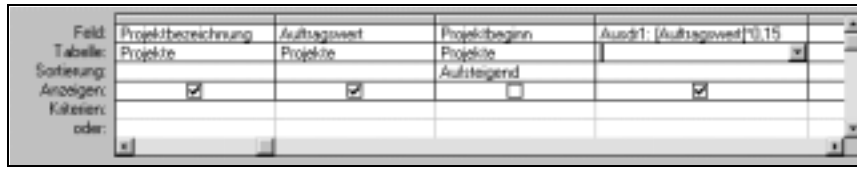


Bild 11.9: Das Feld, in das der Ausdruck »Auftragswert*0, 16« eingegeben wurde, bekam von Access den Namen »Ausdr1«.

- Um den von Access vergebenen Namen *Ausdr1* (bzw. *Ausdr2* etc.) durch einen eigenen Namen zu ersetzen, überschreiben Sie diesen einfach. Den Doppelpunkt dürfen Sie jedoch nicht löschen.
- Selbstverständlich hätten Sie einen von Ihnen definierten Namen auch gleich in einem Schritt zusammen mit dem Ausdruck angeben können. Um beispielsweise zugleich den Namen *Mehrwertsteuer* zu vergeben, hätten Sie schreiben müssen:

  Mehrwertsteuer: Auftragswert*0,16

Weitere Beispiele

Anzahl der Datensätze einer Tabelle/Abfrage bestimmen. Um die Anzahl der Datensätze einer Tabelle oder Abfrage zu ermitteln, nehmen Sie in die Abfrage nur die betreffende Tabelle oder Abfrage auf und definieren ein berechnetes Feld auf die in Bild 11.10 wiedergegebene Weise.

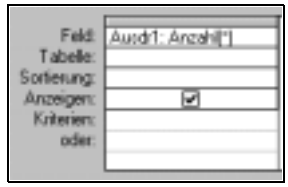


Bild 11.10: Berechnetes Feld zum Ermitteln der Anzahl von Datensätzen einer Abfrage

Anschriftstring. In der Praxis kommt es oft vor, daß Sie sich die Postanschrift für Briefe aus einzelnen Feldern wie *Vorname*, *Name* ... *Ort* zusammensetzen müssen. Dabei soll gelten, daß Firmenname, Vor- und Nachname, Straße sowie Postleitzahl und Ort in je einer Anschriftenzeile stehen. Sie müssen also für Zeilenumbrüche innerhalb des Anschriftstring sorgen. Dies können Sie mit einem berechneten Feld einer Abfrage erledigen, welches Sie dann beispielsweise zur Grundlage eines Listenfeldes in einem Formular machen könnten. Mit dem folgenden Ausdruck komponieren Sie einen Anschriftstring, der in einem berechneten Feld mit dem Namen *Anschrift* ermittelt wird:

Anschrift: [Firma] & Zchn$(13) & Zchn$(10) & [Vorname] & Wenn(IstNull([Vorname]);"";« ») & [Name] & Zchn$(13) & Zchn$(10) & [Straße] & Zchn$(13) & Zchn$(10) & [PLZ] & » » & [Ort]

Mit dem Verkettungsoperator *&* verschmelzen Sie die einzelnen Felder zu einer Zeichenkette. Die *Wenn*-Funktion sorgt dafür, daß vor dem Nachnamen nur dann ein Leerzeichen eingefügt wird, wenn ein Vorname vorhanden ist. Einen Zeilenvorschub auf dem Bildschirm wie auch auf dem Drucker bewirken Sie mit den Zeichen ASCII-Code 13 bzw. 10, welche von den Funktionsausdrücken *Zchn$(13)* bzw. *Zchn$(10)* erzeugt werden.

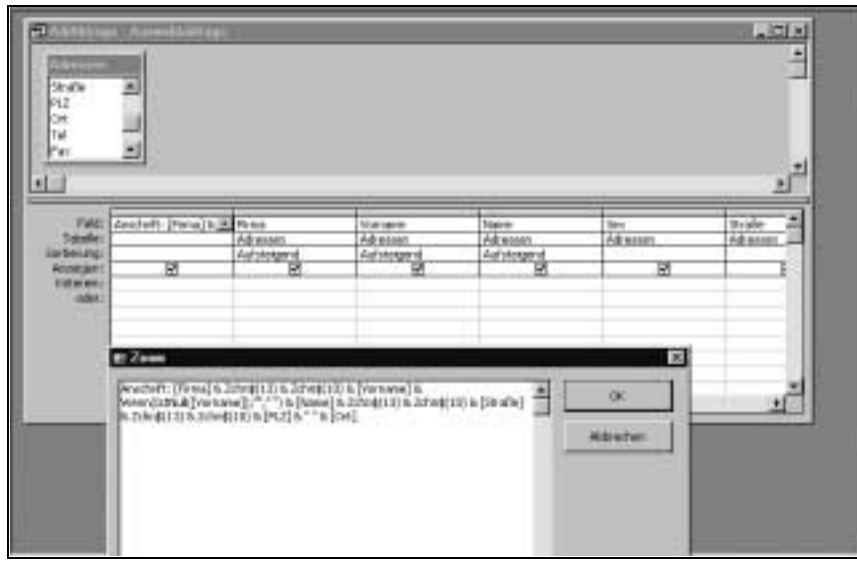


Bild 11.11: In die erste Spalte des Entwurfsbereichs wird der im Dialogfeld »Zoom« (und oben im Text) wiedergegebene Ausdruck zum Ermitteln eines Anschriftstrings eingegeben.

Der in Bild 11.11 wiedergegebene Abfrageentwurf zeigt, wie Sie diesen Ausdruck in eine Abfrage eingeben. Da der Ausdruck sehr lang ist, öffnen Sie für die Eingabe und ggf. erforderliche Bearbeitung am besten das Dialogfeld *Zoom* (⇧+F2). Bild 11.12 zeigt das Ergebnis dieses Abfrageentwurfs.

Anschrift	Firma	Vorname	Name	Sex	Straße	PLZ
Ulrike Grob Fliederweg 45 2000 Hamburg 60		Ulrike	Grob	w	Fliederweg 45	2000
Baumann AG Hans Peter Arzt Ochsenwerder Norderdeich 178 2050 Hamburg 80	Baumann AG	Hans Peter	Arzt	m	Ochsenwerder Norderdeich 178	2050
Drüper-Werke Irmgard Dunen Am Wiedbachle 18 7000 Stuttgart 61	Drüper-Werke	Irmgard	Dunen	w	Am Wiedbachle 18	7000
Im- und Export Handelsgesellschaft Elisabeth	Im- und Export Handelsgesellschaft	Elisabeth	Kirkhammer	w	Bliser Straße 43	2000

Bild 11.12: Datenblattansicht des Abfrageentwurfs aus Bild 11.11. Die Zeilenhöhe wurde manuell so vergrößert, daß die auf mehrere Zeilen umbrochene Anschrift sichtbar wird.

Beim Definieren berechneter Felder erweist sich die Spaltenbreite des betreffenden Feldes fast regelmäßig als zu schmal. Für relativ kurze Ausdrücke könnten Sie die Spalte durch Ziehen ihres rechten Randes im Spaltenkopf verbreitern. Für längere Ausdrücke sei auf die hier wie für andere Eingabefelder existierende Möglichkeit hingewiesen, für die Eingabe das Dialogfeld *Zoom* mit ⇧+F2 zu öffnen.

Beispiel. Ein weiteres, relativ komplexes Beispiel mit berechneten Feldern einer Abfrage finden Sie in Kap. 12, *Komplexe Abfragen*, Punkt 12.2, *Abfragen mit Kriterien aus Steuerelementen in einem Formular (Query-by-Form)*. Dort werden berechnete Felder verwendet, um auch Datensätze mit Nullwerten in den Kriterienfeldern in das Abfrageergebnis aufzunehmen.

Um einen komplexeren Ausdruck für ein berechnetes Feld zu formulieren, empfiehlt es sich möglicherweise, mit dem *Ausdrucks-Generator* zu arbeiten. Klicken Sie auf die Symbol-Schaltfläche *Aufbauen* (vgl. links nebenstehend), um den Ausdrucks-Generator aufzurufen. Das Arbeiten mit Ausdrücken wird genauer in Kap. 39, *Ausdrücke, Funktionen und Operatoren*, behandelt. Dort können Sie sich auch im einzelnen über den Umgang mit dem Ausdrucks-Generator informieren, vgl. den Punkt 39.5, *Arbeiten mit dem Ausdrucks-Generator.*

11.2 Sortieren

Um die Datensätze für das Ergebnis einer Abfrage nach einem oder mehreren Feldern sortieren zu lassen, gehen Sie folgendermaßen vor:

- Klicken Sie für das entsprechende Feld in die Zelle der Zeile *Sortierung*.
- Schlagen Sie das Dropdown-Listenfeld auf, und wählen Sie *Aufsteigend* oder *Absteigend*.
- Gehen Sie für jedes Feld, nach dem sortiert werden soll, entsprechend vor.

Beim Sortieren nach mehr als einem Feld hat der Sortierschlüssel des am weitesten links stehenden Feldes höchste Priorität, der dann folgende zweithöchste usw. Sie können daher durch geeignetes Verschieben der Reihenfolge jede erwünschte Sortierreihenfolge erreichen.

Wenn Sie alle Felder einer Tabelle/Abfrage mit dem Sternchen in die Abfrage aufgenommen haben, können Sie nicht unmittelbar nach einem dieser Felder sortieren lassen. Dazu müssen Sie die Felder, nach denen sortiert werden soll, zusätzlich namentlich in die Abfrage aufnehmen und dann die Sortierfolge dafür angeben. Um zu verhindern, daß dieselben Felder im Abfrageergebnis doppelt wiedergegeben werden, müssen Sie die zusätzlich namentlich aufgenommenen Felder von der Aufnahme in das Abfrageergebnis ausschließen, vgl. oben Punkt 11.1.5, *Felder ausblenden*, vgl. das Beispiel in Bild 11.13.

Bild 11.13: Um nach dem Feld »PLZ« sortieren zu lassen, müssen Sie dieses explizit in die Zeile »Feld« des Entwurfsbereichs aufnehmen und die Sortierfolge einstellen. Da dieses Feld jedoch bereits durch die Angabe »Adressen.*« berücksichtigt ist und aus diesem Grunde angezeigt wird, müssen Sie gleichzeitig das Kontrollkästchen »Anzeigen« ausschalten.

11.3 Kriterien

Häufig werden Sie mit einer Abfrage den Zweck verfolgen, Datensätze auf der Grundlage gewisser Kriterien zusammenzustellen. Access ermöglicht praktisch jede gewünschte Kombination von Kriterien. Wie Sie den folgenden Punkten entnehmen können, ist es nicht schwierig, ein Kriterium im Entwurfsbereich an geeigneter Stelle anzugeben, vorausgesetzt, Sie kennen den zutreffenden Ausdruck dafür. Etwas schwieriger mag es manchmal sein, diesen richtigen Ausdruck zu formulieren. Aber auch dafür bietet Access eine Fülle von Möglichkeiten, indem es leistungsfähige Operatoren (beispielsweise zum Rechnen, Vergleichen, Ausschließen etc.) und Funktionen bereitstellt. Den Umgang mit Ausdrücken können Sie ausführlich in Kap. 39, *Ausdrücke, Funktionen und Operatoren*, nachlesen. Am Ende des vorliegenden Abschnitts finden Sie mehrere Beispiele für Ausdrücke, die speziell bei der Formulierung von Abfragekriterien praktische Bedeutung haben.

Beispiele. Viele Beispiele dieses Punktes *Kriterien* verwenden Tabellen der Datenbank *Nordwind.mdb*, so daß Sie diese Beispiele nacharbeiten können, wenn Sie diese Datenbank öffnen. Sie finden *Nordwind.mdb* im Ordner \Office\Samples, einem Unterordner zum Ordner, in den das Microsoft Office-Paket oder Microsoft Access installiert wurde. Möglicherweise müssen Sie die Beispieldatenbanken nachinstallieren, vgl. dazu Kap. 3, *Access installieren*.

11.3.1 Ein Kriterium für ein Feld

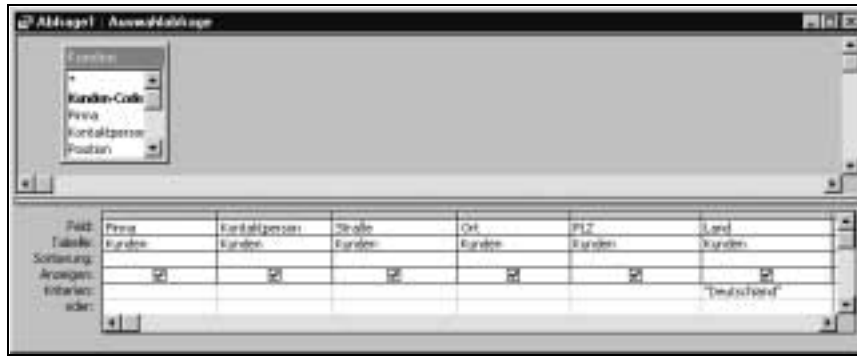


Bild 11.14: Als Kriterium wurde für das Feld »Land« die Textkonstante »Deutschland« angegeben.

Sie geben ein Kriterium für ein Feld an, indem Sie einen Ausdruck in die Zelle der Zeile *Kriterien* schreiben. Der denkbar einfachste Ausdruck ist die Angabe einer Text- oder Zahlenkonstante. Der in Bild 11.14 wiedergegebene Abfrageentwurf enthält die Feldliste *Kunden* der Datenbank *Nordwind.mdb*, von der die Felder *Firma*, *Kontaktperson*, *Straße*, *Ort* und *Land* in den Entwurfsbereich aufgenommen wurden. Um in das Abfrageergebnis

nur Datensätze aufzunehmen, die im Feld *Land* den Eintrag *Deutschland* haben, geben Sie für dieses Feld in die Zeile *Kriterien* den Text *Deutschland* ein, vgl. Bild 11.14. Dabei ist es an dieser Stelle gleichgültig, ob Sie die Textkonstante mit oder ohne Anführungszeichen eingeben, denn Access fügt die im Prinzip für Textkonstanten erforderlichen Anführungszeichen hinzu, wenn der eingegebene Text nicht als reserviertes Wort (z.B. ein Funktionsname oder ein Operator, z.B. der Operator *Wie*) erkannt wird. Das Ergebnis dieser Abfrage sehen Sie in Bild 11.15.

Firma	Kontaktperson	Straße	Ort	PLZ	Land
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Deutschland
Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Deutschland
Drachenblut Delikatessen	Sven Ottlieb	Walserweg 21	Aachen	52066	Deutschland
Frankenversand	Peter Franken	Berliner Platz 43	München	80805	Deutschland
Königlich Essen	Philip Cramer	Maubelstr. 90	Brandenburg	14776	Deutschland
Lehmanns Marktstand	Renate Messner	Magazinweg 7	Frankfurt a.M.	60528	Deutschland
Morgenstern Gesundkost	Alexander Feuer	Heerstr. 22	Leipzig	04179	Deutschland
Ottilies Käseladen	Henriette Pfalzheim	Mühlheimerstr. 369	Köln	50739	Deutschland
QUICK-Stop	Horst Kloss	Taucherstraße 10	Cunewalde	01307	Deutschland
Terra Spezialitäten	Karin Josephs	Luisenstr. 48	Münster	44087	Deutschland
Die Wandernde Kuh	Rita Müller	Adenauerallee 900	Stuttgart	70563	Deutschland

Bild 11.15: Datenblattansicht der Abfrage aus Bild 11.14. Es werden nur Datensätze mit dem Eintrag »Deutschland« im Feld »Land« angezeigt.

11.3.2 Mehrere Kriterien für ein Feld

Für ein Feld können Sie zwei oder mehr Kriterien angeben. Diese müssen dann durch die logischen Operatoren *Und* oder *Oder* miteinander verbunden werden. Das Abfrageergebnis des folgenden Beispiels enthält alle Kunden (und nur diese) aus Deutschland und USA. (Sprachlich scheint die Formulierung *Kunden aus Deutschland und USA* verwirrend, wo doch im Kriterienausdruck der logische Operator *Oder* angegeben ist. Sie können jedoch auch formulieren: Die Kunden stammen aus Deutschland oder USA. Die Umgangssprache ist in diesem Punkte etwas nachlässiger als die Sprache der reinen Logik.)

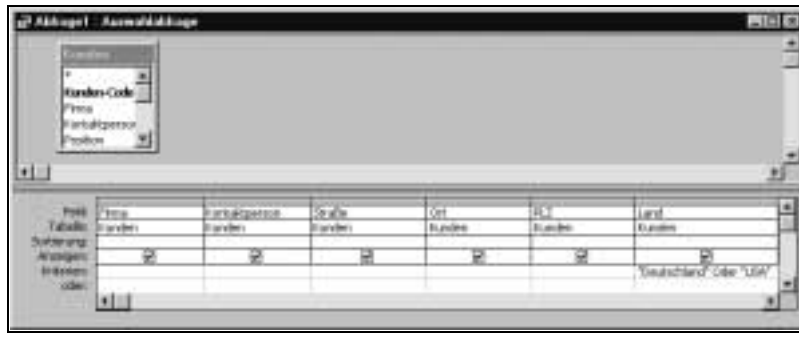


Bild 11.16: Zwei Kriterien, die mit dem logischen Operator »Oder« verbunden wurden.

Dieselbe Wirkung der Oder-Verknüpfung können Sie auch erreichen, wenn Sie die Einträge *Deutschland* sowie *USA* in zwei verschiedene Zeilen *derselben* Feldspalte schreiben, vgl. dazu genauer unten, Punkt 11.3.4, *Kriterien für mehrere Felder.*

11.3.3 Wirkung von Nullwerten in Kriterienfeldern

Wenn Sie eine Abfrage erstellen, ohne ein Kriterium anzugeben, werden stets alle Datensätze in das Abfrageergebnis aufgenommen, gleichgültig, ob Felder in einzelnen Datensätzen Nullwerte (gemeint ist der fehlende Wert NULL, nicht die Zahl 0) enthalten oder nicht. (Im Falle verknüpfter Mehrtabellenabfragen werden Datensätze mit Nullwerten in den Verknüpfungsfeldern nicht angezeigt, vgl. dazu 11.5.1, *Verknüpfung erstellen und aufheben.*) Wenn Sie jedoch für ein Feld ein Kriterium angeben, werden im Abfrageergebnis nur Datensätze berücksichtigt, die in diesem Feld keinen Nullwert haben. Dieser Fall ist dann relevant, wenn das Kriterium in Form eines Ausdrucks angegeben ist und dieser Ausdruck einen leeren Wert ergibt. In diesem Falle ist auch das Abfrageergebnis völlig leer, weil es keinen Datensatz enthält. Beachtenswert und störend ist dieser Effekt besonders, wenn Sie die Kriterien für eine Abfrage aus den Steuerelementen eines Formulars beziehen, deren Inhalt das Abfrageergebnis insoweit steuern soll. Beispielsweise könnten Sie für das Feld *Firma* einer Abfrage den Ausdruck

Formulare!AdressenSuchen!Firma

angeben, um im Ergebnis dieser Abfrage nur Datensätze zu berücksichtigen, die zu der im Textfeld *Firma* des Formulars *AdressenSuchen* angegebenen Firma passen. Falls dieses Textfeld überhaupt keinen Eintrag enthält, bleibt das Abfrageergebnis leer. Im allgemeinen würde der Benutzer jedoch die Nichtangabe eines Wertes so interpretieren, daß er kein Suchkriterium angegeben hat, womit kein Datensatz ausgeschlossen werden dürfte. Eine Lösung für dieses Problem finden Sie in Kap. 12, *Komplexe Abfragen*, Punkt 12.2, *Abfragen mit Kriterien aus Steuerelementen in einem Formular (Query-by-Form)*.

11.3.4 Kriterien für mehrere Felder

Sie können prinzipiell für jedes Feld, das Sie in die Abfrage aufgenommen haben, Kriterien angeben. Diese können dann mit dem logischen *Und* oder dem logischen *Oder* miteinander verbunden sein. Die Art der logischen Verbindung wird in diesem Falle nicht durch Angabe eines der beiden logischen Operatoren definiert. Vielmehr ergibt sie sich daraus, in welchen Kriterienzeilen die Kriterien für verschiedene Felder stehen; denn außer der Zeile *Kriterien* bietet der Entwurfsbereich weitere Kriterienzeilen an, von denen die erste die Bezeichnung *oder* trägt und die weiteren, die Sie ggf. durch senkrechtes Rollen des Entwurfsbereich-Fensters einsehen können, keine Beschriftung haben.

Dieselbe Zeile

Kriterien, die in derselben Zeile stehen, sind durch das logische *Und* miteinander verbunden.

Verschiedene Zeilen

Kriterien, die in verschiedenen Zeilen stehen, sind durch das logische *Oder* miteinander verbunden.

Das Abfrageergebnis des Beispiels in Bild 11.17 enthält alle Datensätze, deren Versanddatum größer als der 1.12.1997 ist *oder* deren Frachtkosten zwischen 800 und 900 liegen:

Bild 11.17: Die beiden Kriterien sind durch das logische »Oder« verbunden, weil sie in verschiedenen Zeilen stehen.

Das Abfrageergebnis des Beispiels in Bild 11.18 enthält alle Datensätze, deren Versanddatum größer als der 1.12.1997 ist *und* deren Frachtkosten zwischen 800 und 900 liegen:

Bild 11.18: Die beiden Kriterien sind durch das logische »Und« verbunden, weil sie in derselben Zeile stehen.

Sie können die logischen Verknüpfungsarten *Und* und *Oder* zwischen den Kriterien verschiedener Felder in beliebiger Weise auch miteinander kombinieren. Das Abfrageergebnis des Abfrageentwurfs aus Bild 11.19 enthält Datensätze, die den folgenden Bedingungen genügen:

Das Versanddatum liegt nach dem 1.12.1997 *und* die Frachtkosten sind zwischen 800 und 900

Oder:

die Postleitzahl lautet 60528

Oder:

der Lieferort ist Aachen oder Köln

Oder:

der Lieferort ist München.

Feld:	Versanddatum	Frachtkosten	PLZ	Ort
Tabelle:	Bestellungen	Bestellungen	Bestellungen	Bestellungen
Sortierung:				
Anzeigen:	☑	☑	☑	☑
Kriterien:	>=#01.12.97#	Zwischen 800 Und 900		
oder:			"60528"	
				"Aachen" Oder "Köln"
				"München"

Bild 11.19: Kombination von »Und«- und »Oder«-Verknüpfungen durch Angabe der Kriterien in derselben bzw. verschiedenen Zeile(n).

Die Oder-Verknüpfung von *Aachen*, *Köln* und *München* hätte auch auf andere Weise realisiert werden können, z.B. dadurch, daß die drei Ortsangaben in derselben Zeile angeführt und mit *Oder* verknüpft worden wären.

Sie können dem Entwurfsbereich weitere Kriterienzeilen hinzufügen, falls die vorhandenen nicht ausreichen: Aktivieren Sie ggf. ein beliebiges Feld im Entwurfsbereich, und wählen Sie dann den Befehl *Zeilen* aus dem Menü *Einfügen*. Überflüssige Zeilen löschen Sie mit dem Befehl *Zeilen löschen* aus dem Menü *Bearbeiten*.

11.3.5 Angabe von Kriterien, wenn das Sternchen für die Feldliste verwendet wird

Oft erleichtert es Ihnen die Arbeit am Abfrageentwurf, wenn Sie alle Felder einer Feldliste durch Angabe des Sternchens (*) in die Abfrage aufnehmen. Um andererseits Kriterien angeben zu können, müssen die betreffenden Felder ausdrücklich (und nicht nur implizit durch das Sternchen) im Abfrageentwurf erscheinen. Sie lösen dieses Problem dadurch, daß Sie die Felder, für die Sie Kriterien angeben wollen, zusätzlich zum Sternchen in den Entwurfsbereich aufnehmen, dafür Kriterien angeben und für diese Felder das Kontrollkästchen *Anzeigen* ausschalten, um eine Doppelanzeige von Feldern in der Datenblattansicht zu verhindern, vgl. das Beispiel in Bild 11.20.

Feld:	Bestellungen.*	Versanddatum	Bestimmungsland
Tabelle:	Bestellungen	Bestellungen	Bestellungen
Sortierung:			
Anzeigen:	☑	☐	☐
Kriterien:		Zwischen #01.12.97# Und #31.12.99#	"Italien"
oder:			

Bild 11.20: Die Angabe »Bestellungen.*« in der ersten Spalte bewirkt, daß alle Felder der Feldliste »Bestellungen« in das Abfrageergebnis aufgenommen werden. Um für die beiden Felder »Versanddatum« und »Bestimmungsland« Kriterien angeben zu können, sind auch diese in den Abfrageentwurf aufgenommen worden. Gleichzeitig ist für diese beiden Felder jedoch das Kontrollkästchen »Anzeigen« ausgeschaltet worden, um eine Doppelanzeige zu verhindern.

11.3.6 Kriterien, die Berechnungen enthalten

Da ein Kriterium als ein Ausdruck angegeben wird, kann es auch Berechnungen oder ähnliche Operationen enthalten. Insbesondere dürfen Sie in Kriterienausdrücken auch Funktionen (einschließlich benutzerdefinierte VBA-Funktionen) anführen. Das Beispiel in Bild 11.21 ermittelt Datensätze, deren Versanddatum mehr als zwei Jahre vor dem aktuellen Datum liegt.

Feld:	Bestellungen.*	Versanddatum
Tabelle:	Bestellungen	Bestellungen
Sortierung:		
Anzeigen:	☑	☐
Kriterien:		<DatAdd("jjjj";-2;Datum())
oder:		

Bild 11.21: Kriterium, das auf einer Berechnung basiert

Sie können sich in einem Kriterienausdruck auch auf andere Access-Objekte, die einen Wert ergeben, beziehen. Insbesondere können Sie sich auf andere Felder in der Abfrage beziehen.

Wenn Sie sich in Kriterienausdrücken auf andere Felder beziehen, müssen die Feldnamen stets zwischen eckigen Klammern angegeben werden.

Das Beispiel in Bild 11.22 verwendet die Tabelle *Bestellungen* der Datenbank *Nordwind.mdb*. Die Abfrage enthält zum Feld *Lieferdatum* einen Kriterienausdruck, der sich auf ein anderes Feld derselben Abfrage – *Bestelldatum* – bezieht. Das Ergebnis dieser Abfrage enthält die Bestellungen, die (erst) mehr als 30 Tage nach dem Bestelldatum ausgeliefert wurden. Dabei wird nur die zugehörige *Personal-Nr* angezeigt.

Bild 11.22: Der Kriterienausdruck bezieht sich auf das Feld »Bestelldatum«, das zwischen eckigen Klammern angegeben werden muß.

Oft kommt es vor, daß ein Kriterium für Werte angegeben werden soll, die implizit in einem Feld enthalten sind. Nehmen wir beispielsweise an, aus der Tabelle *Bestellungen* der Datenbank *Nordwind.mdb* sollen die Bestellungen ausgewählt werden, die an einem Montag aufgegeben wurden. Die Information, an welchem Wochentag eine Bestellung aufgegeben wurde, ist offensichtlich im Feld *Bestelldatum* enthalten, jedoch nicht explizit, sondern nur implizit. Um den zu einer Datumsangabe gehörigen Wochentag zu ermitteln, stellt Access die Funktion *Wochentag()* zur Verfügung. Diese gibt den Wert 1 aus, wenn das Datum einem Sonntag entspricht, 2, wenn es einem Montag entspricht usw. Die Lösung der genannten Aufgabe läuft also darauf hinaus, alle Datensätze auszuwählen, für die gilt:

Wochentag([Bestelldatum]) = 2

Sie können diesen Ausdruck, obwohl er den Vergleichsoperator = nicht links, sondern mitten im Ausdruck enthält, dennoch als Kriterium angeben, wie im Beispiel von Bild 11.23 zu sehen. Das Ergebnis der dort wiedergegebenen Abfrage zeigt die Personalnummer und das Bestelldatum für alle Montagsbestellungen an.

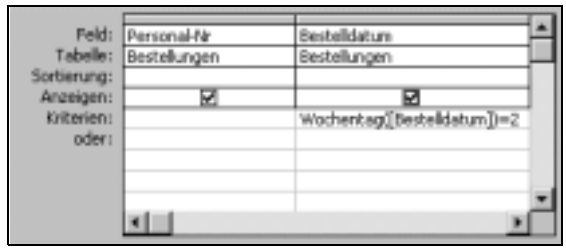


Bild 11.23: Der Kriterienausdruck enthält auf der linken Seite des Vergleichsoperators eine Berechnung.

In derartigen Fällen, bei denen der Kriterienausdruck links vom Vergleichsoperator selbst einen Ausdruck enthält, ist es übrigens gleichgültig, zu welchem Feld Sie diesen Kriterienausdruck angeben, denn dabei wird nicht mit

dem Wert des Feldes verglichen, sondern es wird die linke mit der rechten Seite verglichen. Die Abfrageformulierungen in Bild 11.23 und Bild 11.24 sind daher gleichwertig.

Feld:	Personal-Nr	Bestelldatum
Tabelle:	Bestellungen	Bestellungen
Sortierung:		
Anzeigen:	☑	☑
Kriterien:	Wochentag([Bestelldatum])=2	
oder:		

Bild 11.24: Der Kriterienausdruck aus Bild 11.23 kann in eine beliebige andere Feldspalte geschrieben werden, weil nicht mit den Feldwerten verglichen wird, sondern die linke mit der rechten Seite des Ausdrucks.

11.3.7 Kriterien, die sich auf Formularfelder beziehen

In Kriterienausdrücken können Sie sich auch auf Steuerelemente eines Formulars beziehen. Dabei müssen Sie allerdings die besondere Syntax für die Bezugnahme auf Steuerelemente eines Formulars beachten, wenn dieser Bezug von außerhalb des Formulars erfolgt. Angenommen, Sie wollen als Kriterium den Inhalt des Textfeldes *Auftragsnummer* im Formular mit dem Namen *Hauptaufträge* angeben. Diese Angabe erreichen Sie auf die folgende Weise:

Feld:	Auftragsnummer	Teilauftragsart
Sortierung:		
Anzeigen:	☑	☑
Kriterien:	[Formulare]![Hauptaufträge]![Auftragsnummer]	
oder:		

Bild 11.25: Angabe des Steuerelementes »Auftragsnummer« aus dem Formular »Hauptaufträge« als Kriterium für das Feld »Auftragsnummer« in der bearbeiteten Abfrage

Das Ergebnis der Abfrage in Bild 11.25 würde alle Teilauftragsarten enthalten, die der Auftragsnummer, die im Zeitpunkt der Abfrageaktualisierung im (geöffneten) Formular Hauptaufträge angezeigt wird, zugeordnet sind. Zur Syntax derartiger Ausdrücke vgl. genauer Kap. 39, *Ausdrücke, Funktionen und Operatoren*.

Um komplexere Ausdrücke wie den in Bild 11.25 wiedergegebenen Kriterienausdruck zu formulieren, empfiehlt es sich möglicherweise, mit dem *Ausdrucks-Generator* zu arbeiten. Dieser erscheint besonders nützlich zum Einfügen von Steuerelementen aus Formularen in einen Ausdruck, weil die Syntax derartiger Angaben etwas kompliziert erscheint, der Ausdrucks-

Generator ein Steuerelement jedoch syntaktisch korrekt einfügt. Andererseits versagt der Ausdrucks-Generator gerade beim Einfügen eines Steuerelementes aus einem in ein Hauptformular eingebetteten Unterformular in einen Ausdruck: In diesem Falle ist die Syntax besonders kompliziert (vgl. dazu im einzelnen Kap. 39, *Ausdrücke, Funktionen und Operatoren*).

Um einen Kriterienausdruck mit Hilfe des Ausdrucks-Generators zu erstellen, verfahren Sie prinzipiell wie folgt:

- Setzen Sie den Cursor in das Feld, in das Sie einen Kriterienausdruck eingeben wollen.
- Klicken Sie auf die Symbol-Schaltfläche *Aufbauen*, vgl. links nebenstehend. Dann zeigt sich das Dialogfeld *Ausdrucks-Generator*, mit dem Sie einen Ausdruck erstellen können.

11.3.8 Beispiele

Textkonstante

Das Beispiel in Bild 11.26 gibt alle Nachnamen mit Vornamen aus dem Postleitzahlbezirk 24100 wieder, ohne die Postleitzahl selbst anzuzeigen. Das Feld *Postleitzahl* ist vom Datentyp *Text*.

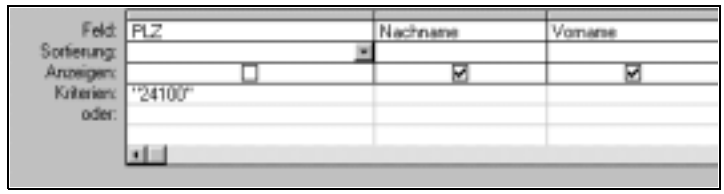


Bild 11.26: Textkonstante als Kriterium

Zahlenbereich

Das Beispiel in Bild 11.27 gibt alle Bestimmungsländer wieder, zu denen Frachtkosten zwischen 30 und 40 realisiert wurden, die Grenzen eingeschlossen. Dem angegebenen Kriterienausdruck ist der folgende gleichwertig: >=30 Und <=40.

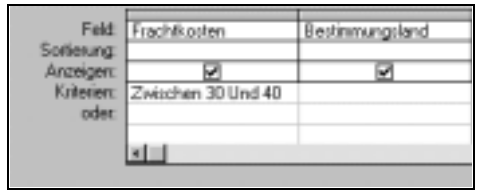


Bild 11.27: Zahlenbereich als Kriterium

Datumsbereich

Das Beispiel in Bild 11.28 gibt alle Nachnamen und Vornamen wieder, deren Geburtsdatum zwischen dem 1.1.1950 und dem 31.12.1989 liegt, die Grenzen eingeschlossen.

Feld:	Geburtsdatum	Nachname	Vorname
Sortierung:	Aufsteigend		
Anzeigen:	☑	☑	☑
Kriterien:	Zwischen #01.01.50# Und #31.12.89#		
oder:			

Bild 11.28: Datumsbereich als Kriterium

Wer hat Heiligabend Geburtstag?

Das Beispiel in Bild 11.29 stellt alle Datensätze mit Geburtstagen am 24.12, gleichgültig in welchem Jahr, zusammen.

Feld:	Geburtsdatum	Nachname	Vorname
Sortierung:	Aufsteigend		
Anzeigen:	☑	☑	☑
Kriterien:	Wie "24.12.*"		
oder:			

Bild 11.29: Dieser Kriterienausdruck ist, obgleich als Textausdruck formuliert, auch für ein Datumsfeld möglich.

Datensätze ohne Nullwert

Das Beispiel in Bild 11.30 ermittelt alle Datensätze, die im Feld *Region* keinen Nullwert haben, d.h., daß sie einen Eintrag haben, eine leere Zeichenfolge (»«) eingeschlossen.

Feld:	Empfänger	Region
Sortierung:		
Anzeigen:	☑	☑
Kriterien:		Ist Nicht Null
oder:		

Bild 11.30: Kriterium für Feld »Region« ohne Nullwert

Datensätze mit Nullwert

Das Beispiel in Bild 11.31 ermittelt alle Datensätze, die im Feld *Region* einen Nullwert haben.

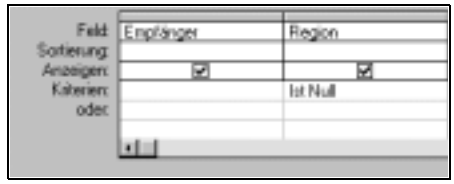


Bild 11.31: Kriterium für Feld »Region« mit Nullwert

Datensätze mit leerer Zeichenfolge

Das Beispiel in Bild 11.32 ermittelt alle Datensätze, die im Feld *Region* eine leere Zeichenfolge (`""`) haben.

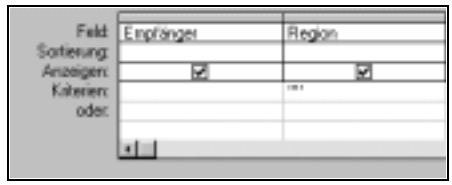


Bild 11.32: Kriterium für Feld »Region« mit leerer Zeichenfolge

Datensätze mit einem Eintrag, leere Zeichenfolge ausgeschlossen

Das Beispiel in Bild 11.33 ermittelt alle Datensätze, die im Feld *Region* einen Eintrag haben, wobei die leere Zeichenfolge (`""`) ausgeschlossen ist. Datensätze mit einem Nullwert im Feld *Region* werden automatisch ausgeschlossen, sobald ein Kriterium für dieses Feld angegeben ist.

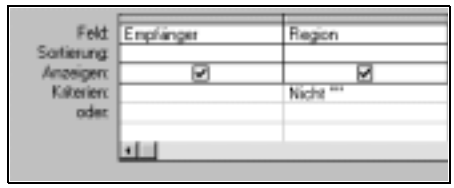


Bild 11.33: Kriterium für nicht leeres Feld »Region«, die leere Zeichenfolge ausgeschlossen

Datensätze, die einem Kriterium nicht entsprechen

Das Beispiel in Bild 11.34 stellt alle Datensätze zusammen, deren Region weder *Québec* noch *Essex* ist. Eine deutlich andere Wirkung hätte der folgende Kriterienausdruck:

Nicht "Québec" Oder "Essex"

Das Fortlassen des Klammerpaares würde bewirken, daß in das Abfrageergebnis Datensätze aufgenommen würden, deren Region nicht *Québec* ist oder deren Region *Essex* ist. Damit wären alle Datensätze mit einer anderen Region als *Québec* berücksichtigt.

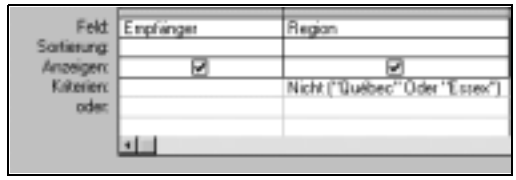


Bild 11.34: Kriterium für Feld »Region: Weder »Québec« noch »Essex«

Doppelte Verneinungen fallen dem Rechner sehr leicht, den Menschen schon schwerer. Welche Datensätze filtert der Ausdruck

Nicht (Nicht ("Québec" Oder "Essex"))

heraus? Antwort: Alle Datensätze mit der Region *Québec* oder *Essex*.

Platzhalterzeichen

Das Beispiel in Bild 11.35 gibt alle Datensätze wieder, deren Nachnamen mit *Brand* und deren Vornamen mit einem *S* beginnen, als zweiten und vierten Buchstaben einen beliebigen, als dritten ein *b* aufweisen und mit der Zeichenfolge *lle* enden. Beispielsweise würden die Namen *Sibylle Brandt* oder *Sybille Brandauer* den Kriterien genügen. Welche weiteren Platzhalterzeichen in Access zur Verfügung stehen und wie diese eingesetzt werden, erfahren Sie in der Online-Hilfe unter dem Stichwort *Platzhalterzeichen*, Thema *Verwenden von Platzhalterzeichen in Zeichenfolgenvergleichen*.

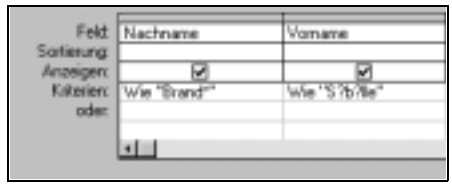


Bild 11.35: Kriterien mit Platzhalterzeichen

Angaben, die einer Werteliste entsprechen

Das Beispiel in Bild 11.36 berücksichtigt Datensätze, deren Lieferorte Berlin, Dresden, London oder Prag sind. Dieselbe Wirkung würde erreicht, wenn Sie statt dessen als Kriterium angäben:

"Berlin" Oder "Dresden" Oder "London" Oder "Prag".

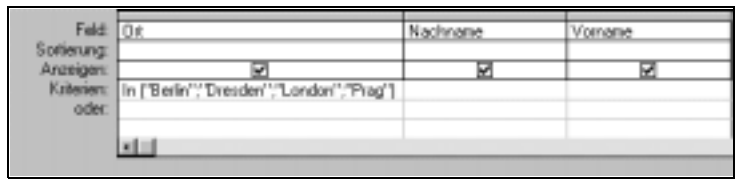


Bild 11.36: Kriterium mit einer Werteliste

Zeichenfolgen, die in alphabetischer Reihenfolge nach einer bestimmten Zeichenfolge kommen

Das Beispiel in Bild 11.37 stellt alle Datensätze zusammen, deren Namen im Alphabet hinter *Mahlmann* kommen. Wie Sie erkennen, lassen sich Vergleichsoperatoren (hier der Operator >) auch zum Vergleich mit Zeichenfolgen angeben. Dabei gilt, daß beispielsweise *A* kleiner ist als *B* etc.; vgl. genauer die Online-Hilfe unter dem Stichwort *Vergleichsoperatoren*.

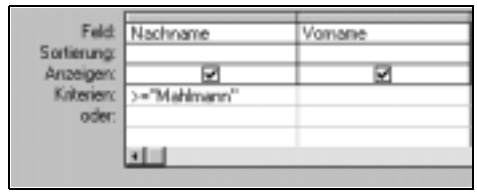


Bild 11.37: Ermittelt werden Datensätze mit Namen, die alphabetisch nach »Mahlmann« kommen.

11.4 Duplikate unterdrücken

In Abfragen kann es leicht vorkommen, daß zwei oder mehr Datensätze im Abfrageergebnis in allen Feldern dieselben Werte besitzen. Man spricht in diesen Fällen von *Duplikaten*. Sie können das Anzeigen von Duplikaten auf die folgende Weise unterdrücken:

- Blenden Sie das Eigenschaftenfenster ein, indem Sie auf die Symbol-Schaltfläche *Eigenschaften* klicken oder den gleichnamigen Befehl aus dem Menü *Ansicht* wählen.
- Das Eigenschaftenfenster muß in seiner Titelleiste die Beschriftung *Abfrageeigenschaften* aufweisen. Falls dort eine andere Beschriftung steht (z.B. *Feldeigenschaften*), müssen Sie im Abfragefenster auf einen Punkt außerhalb einer Feldliste klicken.
- Stellen Sie die Eigenschaft *Keine Duplikate* auf den Wert *Ja* ein.

Normalerweise können Sie die Felder einer Abfrage bearbeiten, so daß auch die Daten in den zugrundeliegenden Tabellen aktualisiert werden; vgl. dazu im einzelnen weiter unten in diesem Kapitel, Abschnitt *Datenaktualisierung bei Ein- und Mehrtabellenabfragen*. Wenn Sie jedoch die Anzeige von Duplikaten unterdrückt haben, ist keine Datenaktualisierung möglich.

11.5 Verknüpfungen bei Mehrtabellenabfragen

Wenn Sie in eine Abfrage mehr als eine Tabelle aufnehmen, sollten die aufgenommenen Tabellen für die Abfrage miteinander verknüpft sein, weil unverknüpfte Tabellen im allgemeinen undurchschaubare und sinnlose Datenzusammenstellungen ergeben.

Eine Verknüpfung wird zwischen zwei Feldern zweier Tabellen/Abfragen definiert. Das Abfrageergebnis enthält dann nur die Datensätze, die in den verknüpften Feldern gleiche Werte haben. Eine Verknüpfung mit dieser Wirkung wird *Gleichheitsverknüpfung* genannt; daneben sind auch andere Verknüpfungsarten mit anderen Wirkungen möglich, vgl. im einzelnen weiter unten in diesem Abschnitt, Punkte *Inklusionsverknüpfung* und *Reflexivverknüpfung*.

Bei Tabellen, zwischen denen von Ihnen bereits eine Beziehung definiert wurde (vgl. Kap. 7, *Eine Datenbank konzipieren*, und Kap. 8, *Tabellen: Grundlagen*, Punkt 8.11, *Beziehungen zwischen Tabellen festlegen*), übernimmt Access diese Beziehung automatisch als Verknüpfung in die Abfrage, wenn die beteiligten Tabellen darin aufgenommen werden. In Bezug auf Tabellen, für die keine Beziehung definiert ist, können (und sollten) Sie eine Verknüpfung für die Abfrage herstellen. Auch in diesen Fällen erstellt Access automatisch eine Verknüpfung, wenn die Option *AutoVerknüpfung aktivieren* auf der Registerkarte *Tabellen/Abfragen* des Befehls *Optionen* aus dem Menü *Extras* angekreuzt ist und kompatible Felder zum Verknüpfen gefunden werden.

Tabellenverknüpfungen in einer Abfrage sind prinzipiell unabhängig von Beziehungen im gerade angesprochenen Sinne: Eine Verknüpfung in einer Abfrage definiert keine dauerhafte Tabellenbeziehung, sondern gilt nur für die Abfrage. Und auch das Umgekehrte gilt: Eine Beziehung zwischen Tabellen wird zwar von Access beim Hinzufügen der Tabellen in die Abfrage als Verknüpfung übernommen, Sie können diese Verknüpfung jedoch für die Abfrage aufheben, ohne die Tabellenbeziehung zu beeinflussen.

11.5.1 Verknüpfung erstellen und aufheben

Verknüpfung erstellen

Um zwei Tabellen bzw. Abfragen im Abfragefenster zu verknüpfen, gehen Sie folgendermaßen vor:

- Klicken Sie auf ein Feld in der Feldliste der einen Tabelle/Abfrage, und ziehen Sie es auf das entsprechende Feld in der Feldliste der anderen Tabelle/Abfrage. Wenn Sie die Maustaste loslassen, fügt Access eine Verbindungslinie zwischen den verknüpften Feldern ein. In Bild 11.38 wurde eine Verknüpfung zwischen den beiden Tabellen *Lieferanten* und *Artikel* hergestellt mit dem jeweiligen Feld *Lieferanten-Nr* als Verknüpfungsfeld.

Anmerkung: Wenn Sie diese Aufgabe für die gleichnamigen Tabellen der Datenbank *Nordwind.mdb* nacharbeiten, fügt Access automatisch eine Verknüpfungslinie ein, sobald Sie die zweite der beiden Tabellen dem Abfrageentwurf hinzugefügt haben, weil in *Nordwind.mdb* zwischen den beiden Tabellen eine 1:n-Beziehung definiert ist.

Zwischen den verknüpften Feldern braucht übrigens keine Namensgleichheit zu existieren. Sie müssen jedoch miteinander verträgliche Datentypen haben. Beispielsweise können Sie zwar ein Feld des Datentyps *Double* mit einem solchen des Datentyps *Integer* verknüpfen, weil beide Zahlenfelder sind, jedoch kein Feld des Typs *Text* mit einem Zahlenfeld. Die Felder sollten sich darüber hinaus inhaltlich entsprechen, damit das Abfrageergebnis keine sinnlosen Datenzusammenstellungen enthält. Während der erste dieser beiden Punkte von Access überprüft wird, ist für den zweiten Punkt allein der Benutzer verantwortlich.

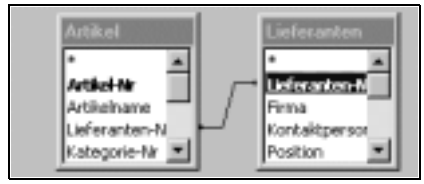


Bild 11.38: Verknüpfung der beiden Tabellen »Artikel« und »Lieferanten«. Verknüpfungsfeld ist »Lieferanten-Nr« in jeder der beiden Tabellen.

Verknüpfung aufheben

Sie heben eine Verknüpfung zwischen zwei Tabellen bzw. Abfragen wieder auf, indem Sie wie folgt verfahren:

- Klicken Sie auf die Verbindungslinie zwischen den beiden Tabellen bzw. Abfragen, um diese zu markieren; sie erscheint dann fett hervorgehoben.
- Drücken Sie die Taste [Entf] oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*.

Wenn Sie eine Verknüpfung zweier Tabellen in einer Abfrage löschen, bleibt eine eventuell zwischen diesen Tabellen definierte Beziehung davon unberührt, d.h. diese bleibt weiterhin bestehen.

Beispiel

Artikel der »Firma Norske Meierier« auflisten. Das folgende Beispiel verwendet zwei Tabellen der Datenbank *Nordwind.mdb*. Das Abfrageergebnis soll alle Artikel wiedergeben, die von der Firma *Norske Meierier* geliefert werden können. Gehen Sie dazu folgendermaßen vor:

- Erstellen Sie eine neue Abfrage, indem Sie im Datenbankfenster die Registerkarte *Abfragen* aktivieren und danach auf die Schaltfläche *Neu* klikken.
- Fügen Sie der Abfrage die Tabellen *Lieferanten* sowie *Artikel* hinzu (Auswahl aus dem Dialogfeld *Tabelle anzeigen*). Da in der Datenbank *Nordwind.mdb* zwischen den Tabellen *Lieferanten* und *Artikel* eine Beziehung definiert ist, fügt Access automatisch eine Verknüpfungslinie zwischen den Verknüpfungsfeldern mit dem jeweiligen Namen *Lieferanten-Nr* ein.

- Ziehen Sie die Felder *Firma* und *Artikelname* in den Entwurfsbereich.
- Geben Sie als Kriterium zum Feld *Firma* die Zeichenfolge »Norske Meierier« an, die Anführungszeichen eingeschlossen. (Falls Sie die Anführungszeichen nicht angeben, fügt Access diese ein.)

Ihr Abfrageentwurf sollte nunmehr so aussehen wie in Bild 11.39.

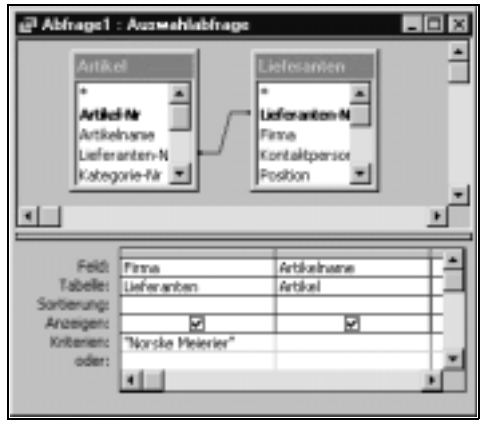


Bild 11.39: Abfrageentwurf für zwei verknüpfte Tabellen mit Kriteriumangabe für ein Feld

Um das Ergebnis dieser Abfrage anzuschauen, wechseln Sie zur Datenblattansicht. Im Ergebnis sollten Sie die folgenden drei Datensätze sehen:

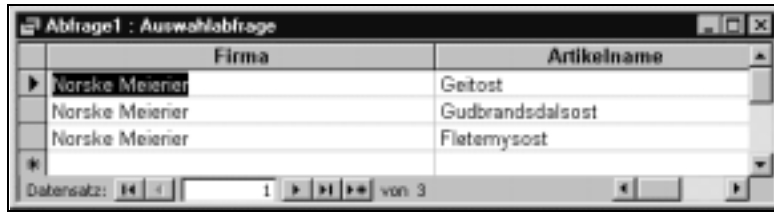


Bild 11.40: Datenblattansicht der Abfrage aus Bild 11.39

Inhalt des Abfrageergebnisses bei unverknüpfter Mehrtabellenabfrage

Beispiel. Sie können die Wirkung der Verknüpfung sehr deutlich erkennen, wenn Sie diese einmal aufheben und sich das dann resultierende Abfrageergebnis anschauen. Nehmen Sie dabei den Endstand des Abfrageentwurfs aus dem vorangehenden Beispiel (vgl. Bild 11.39) als Ausgangspunkt und verändern Sie diesen wie folgt:

- Wechseln Sie ggf. in die Entwurfsansicht.
- Markieren Sie die Verknüpfungslinie zwischen den beiden Tabellen, indem Sie darauf klicken.

- Drücken Sie die Taste [Entf], oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*.
- Löschen Sie das Kriterium für das Feld *Firma*.
- Wechseln Sie in die Datenblattansicht.

Bild 11.41: Mehrtabellenabfrage ohne Verknüpfung

Bild 11.42: Das Abfrageergebnis der unverknüpften Zweitabellen-Abfrage enthält für jede Datensatz-Kombination aus den beiden Tabellen einen Datensatz; das sind im Ergebnis 2233 Datensätze.

In diesem Falle sollten Sie im Abfrageergebnis ein ziemlich sinnloses Ergebnis sehen: Es werden 2233 Datensätze angezeigt, vgl. Bild 11.42. Jeder Datensatz weist eine andere Kombination Firma/Artikel aus. Da die Tabelle *Lieferanten* 29 Datensätze enthält und die Tabelle *Artikel* 77 Datensätze, ergeben sich 29 * 77 = 2233 Kombinationen.

11.5.2 Datensätze mit Nullwerten in verknüpften Feldern bleiben unberücksichtigt

Wenn Sie eine Verknüpfung zwischen zwei Tabellen/Abfragen erstellt haben, enthält das Ergebnis der Abfrage nur Datensätze mit übereinstimmenden Werten in den Verknüpfungsfeldern. Dabei gelten Nullwerte nicht als Übereinstimmung. Die Konsequenz ist daher, daß alle Datensätze mit Nullwerten in den Verknüpfungsfeldern im Abfrageergebnis unberücksichtigt bleiben. Das folgende Beispiel demonstriert diesen Effekt für die beiden Tabellen *Lieferanten* und *Kunden* aus der Datenbank *Nordwind.mdb*. Jede der beiden Tabellen enthält ein Feld *Region*, über das eine Verknüpfung hergestellt werden kann. In jeder der beiden Tabellen existieren zahlreiche Datensätze, die für dieses Feld keinen Eintrag haben, also jeweils einen Nullwert besitzen.

Feld:	Firma	Region
Tabelle:	Lieferanten	Lieferanten
Sortierung:		Absteigend
Anzeigen:	☑	☑
Kriterien:		

Bild 11.43: Eintabellenabfrage für die Tabelle »Lieferanten«

Die Abfrage für die Verteilung der Werte für *Firma* und *Region* in der Tabelle *Lieferanten* allein wird in Bild 11.43 und Bild 11.44 wiedergegeben. Die Datenblattansicht zeigt zahlreiche Nullwerte für das Feld *Region* und weist insgesamt 29 Datensätze aus.

Abfrage1 : Auswahlabfrage

Firma	Region
Pavlova, Ltd.	Victoria
Ma Maison	Québec
Fromages d'érables	Québec
Bigfoot Breweries	OR
Gitay, Mate	NSW
Grandma Kelly's Homestead	MI
New England Seafood Cannery	MA
New Orleans Cajun Delights	LA
Cooperativa de Quesos 'Las Cabras'	Asturias
Heli Süßwaren GmbH & Co. KG	
Refrescos Americanas LTDA	
PB Knäckebröd AB	
Plutzer Lebensmittelgroßmärkte AG	
Mayumi's	
Tokyo Traders	
Specialty Biscuits, Ltd.	

Bild 11.44: Ergebnis der Abfrage aus Bild 11.43, d.h. für die Tabelle »Lieferanten« allein

Die Abfrage für die Verteilung der Werte für *Firma* und *Region* in der Tabelle *Kunden* allein wird in Bild 11.45 und Bild 11.46 wiedergegeben. Die Datenblattansicht zeigt auch hier zahlreiche Nullwerte für das Feld *Region* und weist insgesamt 91 Datensätze aus.

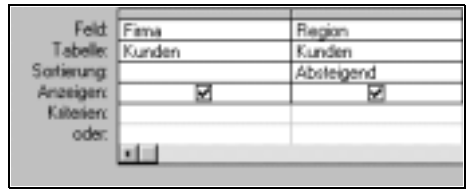


Bild 11.45: Eintabellenabfrage für die Tabelle »Kunden«

Bild 11.46: Ergebnis der Abfrage aus Bild 11.45, d.h. für die Tabelle »Kunden« allein

Die Abfrage für die Verteilung der Werte für *Firma* und *Region* in den beiden über das Feld *Region* verknüpften Tabellen *Lieferanten* und *Kunden* wird in Bild 11.47 und Bild 11.48 wiedergegeben. Die Datenblattansicht zeigt hier nur noch sechs Datensätze, davon keiner mit einem Nullwert für das Verknüpfungsfeld *Region*.

Falls Sie Datensätze mit leeren Einträgen in einer Mehrtabellenabfrage im Abfrageergebnis berücksichtigt haben wollen, müssen Sie statt mit Nullwerten mit leeren Zeichenfolgen (»«) arbeiten, denn Verknüpfungsfelder mit leeren Zeichenfolgen werden in das Abfrageergebnis aufgenommen, weil

diese als übereinstimmend betrachtet werden. Zur Behandlung von Nullwerten und leeren Zeichenfolgen bei der Tabellendefinition vgl. Kap. 8, *Tabellen: Grundlagen*, Punkte 8.5.5, *Dateneingabe erzwingen* und 8.5.6, *Leere Zeichenfolgen*.

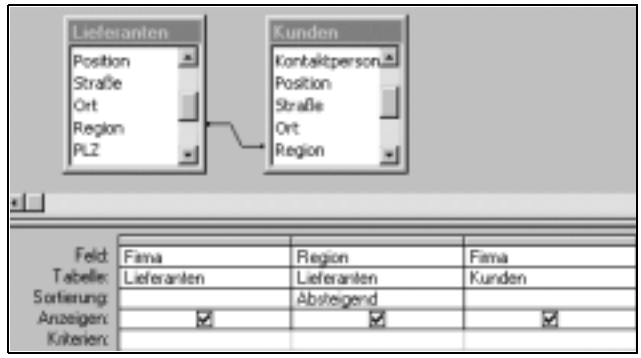


Bild 11.47: Mehrtabellenabfrage für die Tabellen »Lieferanten« und »Kunden« mit einer Verknüpfung über das Feld »Region«

Bild 11.48: Ergebnis der Abfrage aus Bild 11.47, d.h. für die Tabellen »Lieferanten« und »Kunden«, die über das Feld »Region« verknüpft sind

11.5.3 Verknüpfte Tabellen aufnehmen, ohne ihre Felder anzuzeigen

Ähnlich wie Sie Kriterien für Felder angeben können, die selbst gar nicht angezeigt werden, können bzw. müssen Sie verknüpfte Tabellen in die Abfrage einbeziehen, ohne auch nur eines ihrer Felder anzeigen zu lassen.

Abfrage für drei Tabellen, ohne Felder der mittleren Tabelle anzuzeigen

Die Datenbank *Nordwind.mdb* enthält die Tabellen *Kategorien* und *Bestelldetails*. Eine Abfrage soll für jedes Bestelldetail den Umsatz und den zugehörigen Kategorienamen ausweisen. Den Umsatz erhalten Sie als mathematisches Produkt der Felder *Einzelpreis* und *Anzahl* minus gewährten *Rabatt* aus *Bestelldetails*, *Kategoriename* liefert die Tabelle *Kategorien*. Zwischen

diesen beiden Tabellen kann allerdings keine direkte Verknüpfung hergestellt werden, vielmehr bedarf es der dritten Tabelle *Artikel*, die jeweils mit den beiden Tabellen *Bestelldetails* und *Kategorien* verknüpfbar ist. Verfahren Sie folgendermaßen, um die Aufgabe zu lösen:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*.
- Erstellen Sie eine neue Abfrage.
- Nehmen Sie in die Abfrage die Tabellen *Kategorien*, *Artikel* und *Bestelldetails* auf, indem Sie diese aus dem Dialogfeld *Tabelle anzeigen* auswählen. Die Beziehungen, die für die Tabellen in der Datenbank definiert sind, werden von Access automatisch als Abfrageverknüpfungen eingefügt.
- Ziehen Sie das Feld *Kategoriename* aus der Feldliste der Tabelle *Kategorien* in den Entwurfsbereich.
- Schreiben Sie zur Berechnung des Umsatzes in die nächste freie Spalte des Entwurfsbereichs den Ausdruck

  Umsatz: [Bestelldetails].[Einzelpreis]*[Anzahl]*(1-[Rabatt])
- Lassen Sie die Datensätze nach dem Umsatz absteigend sortieren: Klicken Sie in die Zeile *Sortierung* für dieses Feld, und wählen Sie aus dem Dropdown-Listenfeld den Eintrag *Absteigend*.

Bild 11.49: Abfrage für drei Tabellen. Aus der dritten Tabelle, hier der Tabelle »Artikel«, wurde kein Feld in die Abfrage aufgenommen.

Nach diesen Schritten sollte Ihr Abfrageentwurf aussehen, wie in Bild 11.49 wiedergegeben. Wenn Sie nun zur Datenblattansicht wechseln, sehen Sie die Gegenüberstellungen von Kategorien und Umsätzen,. sortiert nach der Umsatzgröße der Bestelldetails.

Abfrage1 : Auswahlabfrage

Kategoriename	Umsatz
Getränke	15810
Getränke	15019,4999882206
Getränke	10540
Getränke	10540
Fleischprodukte	9903,2
Getränke	8431,99996858835
Getränke	8263,39996921659
Getränke	7906
Getränke	7906
Getränke	7509,74999411032
Getränke	6687,5
Getränke	6324
Getränke	6324
Milchprodukte	6050
Naturprodukte	6041,99999526143
Fleischprodukte	5570,55

Bild 11.50: Ergebnis der Abfrage aus Bild 11.49

11.5.4 Inklusionsverknüpfung

Verknüpfungen in Abfragen dienen dazu, aus (je) zwei Tabellen/Abfragen diejenigen Datensätze zusammenzustellen, die in den verknüpften Feldern jeweils denselben Wert haben. Je nachdem, ob im Abfrageergebnis auch Datensätze aus einer Seite der Verknüpfung berücksichtigt werden, denen kein Datensatz auf der anderen Verknüpfungsseite entspricht, lassen sich drei Arten von Verknüpfung unterscheiden:

Gleichheitsverknüpfung (equi-join). Das Abfrageergebnis enthält nur Datensätze, die auf beiden Seiten der Verknüpfung jeweils denselben Wert in den verknüpften Feldern besitzen. Alle bisher behandelten Verknüpfungen waren Gleichheitsverknüpfungen, denn Access nimmt diese Verknüpfungsart automatisch an, wenn der Benutzer nicht ausdrücklich eine der beiden folgenden anderen Verknüpfungsarten bestimmt.

Inklusionsverknüpfung, left join. Wenn das Abfrageergebnis alle Datensätze der linken Verknüpfungsseite enthalten soll, von der rechten Verknüpfungsseite dagegen nur diejenigen, bei denen die Inhalte der verknüpften Felder beider Seiten gleich sind, müssen Sie eine *Inklusionsverknüpfung* (Inklusion = Einschluß) vom Typ *left join* wählen.

Inklusionsverknüpfung, right join. Wenn das Abfrageergebnis alle Datensätze der rechten Verknüpfungsseite enthalten soll, von der linken Verknüpfungsseite dagegen nur diejenigen, bei denen die Inhalte der verknüpften Felder beider Seiten gleich sind, müssen Sie eine *Inklusionsverknüpfung* vom Typ *right join* wählen.

Der Unterschied zwischen einer Inklusions- und einer Gleichheitsverknüpfung wird in den folgenden Ausführungen an einem Beispiel mit zwei Tabellen der Datenbank Projekte.mdb genauer erklärt. Dabei wird auch gezeigt,

wie Sie eine Inklusionsverknüpfung definieren. Die Datenbank Projekte.mdb wurde in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, entwickelt. Sie finden sie auch auf der Begleit-CD-ROM.

Gleichheitsverknüpfung erstellen

Aufgabe

Stellen Sie für die Tabellen *Personal* und *Projekte* aus der Datenbank *Projekte.mdb* zunächst eine Gleichheitsverknüpfung her. Das Abfrageergebnis soll die Felder *PersonalCode* und *Nachname* aus der Tabelle *Personal* sowie die Felder *PersonalCode* und *Projektbezeichnung* aus der Tabelle *Projekte* enthalten. Diese Aufgabe lösen Sie wie folgt:

- Erstellen Sie, nachdem Sie die Datenbank *Projekte.mdb* geöffnet haben, aus dem Datenbankfenster heraus eine neue Abfrage.
- Fügen Sie der Abfrage die beiden Tabellen *Personal* und *Projekte* hinzu. Da für diese beiden Tabellen eine 1:n-Beziehung definiert ist, fügt Access automatisch eine Verknüpfung zwischen den Feldern mit dem jeweils selben Namen *PersonalCode* ein, was an der Verknüpfungslinie zu erkennen ist, vgl. Bild 11.51.
- Ziehen Sie aus der Feldliste der Tabelle *Personal* die Felder *PersonalCode* und *Nachname*, aus der Feldliste *Projekte* die Felder *PersonalCode* und *Projektbezeichnung* in den Entwurfsbereich.
- Um besser erkennen zu können, aus welcher Tabelle die Felder im Entwurfsbereich jeweils stammen, blenden Sie ggf. die Zeile *Tabelle* ein: Wählen Sie den Befehl *Tabellennamen* aus dem Menü *Ansicht*.

Ihr Abfrageentwurf sollte jetzt aussehen wie derjenige in Bild 11.51.

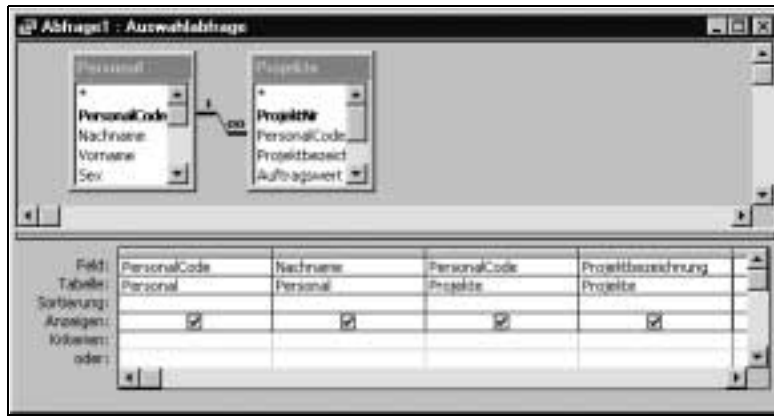


Bild 11.51: Abfrageentwurf für eine Gleichheitsverknüpfung der beiden Tabellen »Personal« und »Projekte«

Um sich das Abfrageergebnis anzuschauen, wechseln Sie zur Datenblattansicht. Sie erhalten dann die Zusammenstellung von Datensätzen, wie sie in Bild 11.52 zu sehen ist: Es werden nur Datensätze mit demselben *PersonalCode* in jeder der beiden Tabellen wiedergegeben.

Abfrage1 : Auswahlabfrage

Personal.Perso	Nachname	Projekte.Perso	Projektbezeichnung
GERSD	Gersdorff	GERSD	Dachausbau Hasenclever
GERSD	Gersdorff	GERSD	Reparatur Stallungen Meierhof Ebbingen
LAUSE	Lauser	LAUSE	Gartenanlage Meerburg
LAUSE	Lauser	LAUSE	Stadtparkbeplanzung
MAHLM	Mahlmann	MAHLM	Sportplatz Grundschule Heisterbach
MAHLM	Mahlmann	MAHLM	Türdurchbruch Meier-Seelmann
MAHLM	Mahlmann	MAHLM	Neubau Einfamilenhaus Ottingen
MAHLM	Mahlmann	MAHLM	Sielreparatur Erdnmannsweg
MAHLM	Mahlmann	MAHLM	Grundschule Hermannsdorf: Toilettenneubau
THIED	Thiedemann	THIED	Sielanlage Bornplatz 12
THIED	Thiedemann	THIED	Drainage Bahnhofvorplatz Klein Haren
THIED	Thiedemann	THIED	Anlage des Golfplatzes Peiner Hag

Datensatz: 1 von 12

Bild 11.52: Ergebnis der Abfrage mit Gleichheitsverknüpfung aus Bild 11.51. Es werden nur Datensätze mit demselben »PersonalCode« in jeder der beiden Tabellen wiedergegeben.

Inklusionsverknüpfung, left join, erstellen

Aufgabe. Wandeln Sie die Gleichheitsverknüpfung aus dem vorangehenden Beispiel (vgl. Bild 11.51) in eine *Inklusionsverknüpfung, left join*, um, und betrachten Sie anschließend das Abfrageergebnis in der Datenblattansicht. Diese Aufgabe lösen Sie wie folgt:

- Wechseln Sie zunächst ggf. wieder in die Entwurfsansicht der Abfrage.
- Doppelklicken Sie auf die Verknüpfungslinie zwischen den beiden Tabellen im Abfragefenster. Dann zeigt sich das Dialogfeld *Verknüpfungseigenschaften*, das in Bild 11.53 wiedergegeben ist.
- Wählen Sie die Option 2, und bestätigen Sie mit OK. Daraufhin erscheint die Verknüpfungslinie mit einem Pfeil an ihrer rechten Seite, vgl. Bild 11.54.
- Schauen Sie sich das Ergebnis der *Inklusionsverknüpfung, left join*, an, indem Sie in die Datenblattansicht wechseln. Sie sollten eine Zusammenstellung der Datensätze sehen, wie sie in Bild 11.55 wiedergegeben ist. Beachten Sie die beiden gegenüber dem Abfrageergebnis der Gleichheitsverknüpfung (vgl. Bild 11.52) neu hinzugekommenen Datensätze mit den Namen *Biedermann* und *Lejeune*: Diese beiden Datensätze stammen aus der Tabelle *Personal* (also der linken Verknüpfungsseite) und haben kein zugeordnetes Projekt.

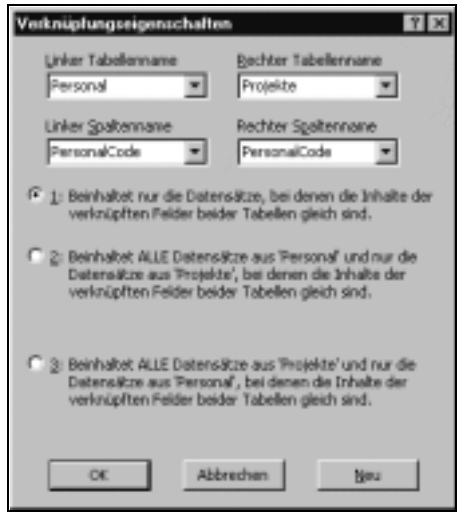


Bild 11.53: Dialogfeld »Verknüpfungseigenschaften« zum Festlegen der Verknüpfungsart

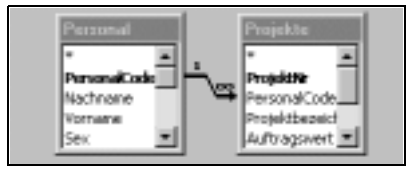


Bild 11.54: Die Verknüpfungslinie weist einen Pfeil auf um anzuzeigen, daß eine »Inklusionsverknüpfung, left join«, definiert ist. Bei einer »Inklusionsverknüpfung, right join«, wiese der Pfeil auf die linke Seite.

Personal.Perso	Nachname	Projekte.Perso	Projektbezeichnung
BIEDE	Biedermann		
GERSD	Gersdorff	GERSD	Dachausbau Hasenclever
GERSD	Gersdorff	GERSD	Reparatur Stallungen Meierhof Ebingen
LAUSE	Lauser	LAUSE	Gartenanlage Meerburg
LAUSE	Lauser	LAUSE	Stadtparkbepflanzung
LEJEU	Lejeune		
MAHLM	Mahlmann	MAHLM	Sportplatz Grundschule Heisterbach
MAHLM	Mahlmann	MAHLM	Türdurchbruch Meier-Seelmann
MAHLM	Mahlmann	MAHLM	Neubau Einfamilienhaus Ottingen
MAHLM	Mahlmann	MAHLM	Sielreparatur Erdmannsweg
MAHLM	Mahlmann	MAHLM	Grundschule Hermannsdorf: Toilettenneubau
THIED	Thiedemann	THIED	Sielanlage Bornplatz 12
THIED	Thiedemann	THIED	Drainage Bahnhofvorplatz Klein Haren
THIED	Thiedemann	THIED	Anlage des Golfplatzes Peiner Hag

Bild 11.55: Datenblattansicht der Abfrage aus Bild 11.54, d.h. »Inklusionsverknüpfung, left join«. Das Ergebnis zeigt alle Datensätze der linken Seite (Tabelle »Personal«), unabhängig davon, ob auf der rechten Seite (Tabelle »Projekte«) ein zugeordneter Datensatz vorhanden ist, vgl. die Namen »Biedermann« und »Lejeune«.

Eine *Inklusionsverknüpfung, right join*, können Sie auf entsprechende Weise wie die vorangehende erstellen. Im vorliegenden Beispiel würde sich ihr Ergebnis nicht von der Gleichheitsverknüpfung unterscheiden, weil zwischen den beiden Tabellen *Personal* und *Projekte* eine 1:n-Beziehung *mit referentieller Integrität* besteht: In einem derartigen Fall darf es auf der rechten Seite (der n-Seite) keinen Datensatz geben, dem kein Datensatz auf der linken Seite (der 1-Seite) entspricht. Eine Ausnahme würden allerdings Datensätze in der Detailtabelle mit Nullwerten im Fremdschlüsselfeld bilden: Diese sind nach den Regeln der referentiellen Integrität zugelassen und würden in einer *Inklusionsverknüpfung, right join* auch ausgegeben werden. Im vorliegenden Beispiel sind Nullwerte im Feld *PersonalCode* – dem Fremdschlüssel in der Detailtabelle – allerdings dadurch ausgeschlossen, daß für dieses Feld Eingabepflicht definiert ist, weil die Eigenschaft *Eingabe erforderlich* auf *Ja* eingestellt wurde.

11.5.5 Reflexivverknüpfung

In einer Abfrage kann eine Tabelle auch mit sich selbst verknüpft werden. Man spricht dann von einer *Reflexivverknüpfung*. Bei einer Reflexivverknüpfung wird ein Feld der Tabelle mit einem anderen Feld *in derselben Tabelle* verknüpft. Dies ist natürlich nur in bestimmten Fällen sinnvoll, denn die Tabelle muß zwei Felder aufweisen, die das Ergebnis einer Abfrage mit Reflexivverknüpfung auch interpretierbar machen. Die Datenbank *Nordwind.mdb* bietet mit ihrer Tabelle *Personal* ein gutes Beispiel für die Möglichkeit einer Reflexivverknüpfung. Diese Tabelle enthält u.a. die beiden Felder *Vorgesetzte(r)* und *Personal-Nr.* Das Feld *Personal-Nr* enthält eine eindeutige Personalnummer für jeden Mitarbeiter und ist gleichzeitig Primärschlüssel. Das Feld *Vorgesetzte(r)* enthält die Personalnummer des Vorgesetzten, der, da er ja auch Mitarbeiter ist, selbst durch einen Datensatz in der Tabelle *Personal* repräsentiert wird. Mit einer Reflexivverknüpfung können Sie ein Abfrageergebnis erzielen, das für jeden Mitarbeiter u.a. den Namen seines Vorgesetzten ausweist – wenn er denn einen hat. Um dies zu erreichen, gehen Sie folgendermaßen vor:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*.
- Erstellen Sie eine neue Abfrage.
- Fügen Sie der Abfrage die Tabelle *Personal* hinzu.
- Fügen Sie der Abfrage die Tabelle *Personal ein zweites Mal* hinzu, und schließen Sie dann das Dialogfeld *Tabelle anzeigen*. Access löst den Namenskonflikt, der durch das zweimalige Hinzufügen derselben Tabelle entsteht, dadurch, daß es dem Namen der weiteren Tabelle die Zeichen *_1* anfügt. Im Abfragefenster sollten Sie daher die beiden Feldlisten *Personal* und *Personal_1* sehen, vgl. Bild 11.56.

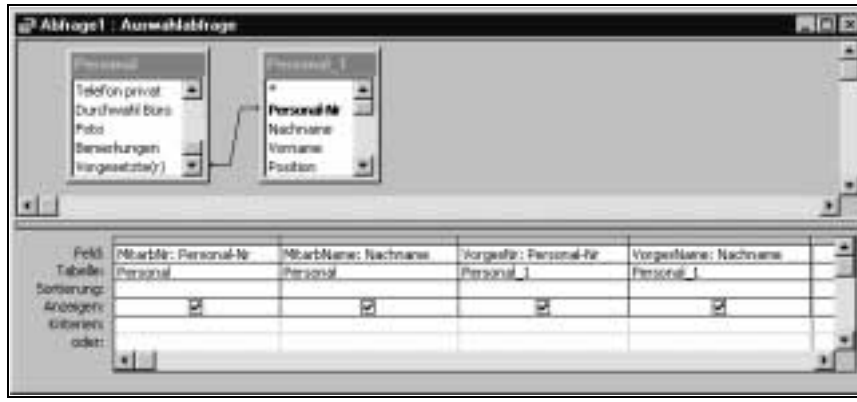


Bild 11.56: Die Tabelle »Personal« wurde dem Abfragefenster zweifach hinzugefügt. Daher erhielt die als zweite hinzugefügte Feldliste den Namen »Personal_1«.

- Fügen Sie eine Verknüpfungslinie vom Feld *Vorgesetzte(r)* der Feldliste *Personal* zum Feld *Personal-Nr* der Feldliste *Personal_1* ein, indem Sie vom erstgenannten Feld zum zweitgenannten ziehen.
- Ziehen Sie die Felder *Personal-Nr* und *Nachname* aus der Feldliste *Personal* in den Entwurfsbereich.
- Benennen Sie das Feld *Personal-Nr* mit *MitarbNr*, indem Sie in die entsprechende Zelle im Entwurfsbereich *MitarbNr:* vor den Feldnamen *Personal-Nr* schreiben. Verfahren Sie für das Feld *Nachname* entsprechend, indem Sie dieses mit dem Namen *MitarbName* benennen. (Sie können sich über das Benennen von Feldern in Abfragen in diesem Kapitel oben unter Punkt 11.1.3, *Felder benennen*, informieren.)
- Ziehen Sie die Felder *Personal-Nr* und *Nachname* aus der Feldliste *Personal_1* in den Entwurfsbereich.
- Benennen Sie das zweite Feld *Personal-Nr* mit *VorgesNr* und das zweite Feld *Nachname* mit *VorgesName*.

Nach diesen Schritten sollte Ihr Abfrageentwurf aussehen, wie in Bild 11.56 dargestellt. Wenn Sie jetzt in die Datenblattansicht wechseln, sollten Sie die Zusammenstellung von Datensätzen sehen, die in Bild 11.57 wiedergegeben ist. Es werden drei Mitarbeiter ausgewiesen, jeder mit seinem Vorgesetzten. Wenn Sie schon öfter mit der Tabelle *Personal* aus der Datenbank *Nordwind.mdb* gearbeitet haben, wird es Sie vermutlich stutzig machen, daß das Abfrageergebnis nur drei Datensätze ausweist, obwohl die Tabelle *Personal* Datensätze für neun Mitarbeiter enthält. Dies hat seinen Grund darin, daß die Abfrage mit einer *Gleichheitsverknüpfung*, die ja voreingestellt ist, ausgeführt wurde: Die meisten Mitarbeiter haben keinen Vorgesetzten, so daß in ihren Datensätzen das Feld *Vorgesetzte(r)* einen Nullwert hat. (Die Nordwind GmbH scheint eine wahrhaft demokratische weil hierarchiefreie Firma zu sein, haben doch nur drei ihrer neun Mitarbeiter einen Vorgesetzten!)

Abfrage1 : Auswahlabfrage

MitarbNr	MitarbName	VorgesNr	VorgesName
1	Davolio	2	Fuller
2	Fuller	8	Callahan
3	Leverling	8	Callahan
(AutoWert)		(AutoWert)	

Datensatz: 1 von 3

Bild 11.57: Ergebnis der Abfrage mit Reflexivverknüpfung. Verknüpfungsart: Gleichheitsverknüpfung

Wenn auch die Datensätze der Mitarbeiter ohne Vorgesetzten im Abfrageergebnis erscheinen sollen, muß die Abfrage mit einer *Inklusionsverknüpfung, left join*, ausgeführt werden, damit *alle* Datensätze der linken Verknüpfungsseite angezeigt werden. Zu diesem Zweck gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. in die Entwurfsansicht der Abfrage zurück.
- Doppelklicken Sie auf die Verknüpfungslinie zwischen den beiden Feldlisten *Personal* und *Personal_1*.
- Wählen Sie im Dialogfeld *Verknüpfungseigenschaften* die Option 2 für *Inklusionsverknüpfung, left join*, und bestätigen Sie mit OK.

Als Ergebnis dieser Schritte sollte die Verknüpfungslinie auf ihrer rechten Seite einen Pfeil aufweisen, vgl. Bild 11.58.

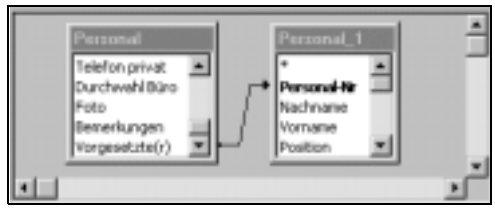


Bild 11.58: Reflexivverknüpfung als» Inklusionsverknüpfung, left join«

Abfrage1 : Auswahlabfrage

MitarbNr	MitarbName	VorgesNr	VorgesName
1	Davolio	2	Fuller
2	Fuller	8	Callahan
3	Leverling	8	Callahan
4	Peacock		
5	Buchanan		
6	Suyama		
7	King		
8	Callahan		
9	Dodsworth		
(AutoWert)		(AutoWert)	

Datensatz: 1 von 9

Bild 11.59: Ergebnis der Abfrage mit Reflexivverknüpfung als »Inklusionsverknüpfung, left join«. Auch für die Mitarbeiter ohne Vorgesetzten werden Datensätze wiedergegeben.

Im Ergebnis der Abfrage mit Reflexivverknüpfung als *Inklusionsverknüpfung, left join*, werden nunmehr auch die Mitarbeiter ohne Vorgesetzten berücksichtigt, vgl. Bild 11.59. Korrekterweise bleiben in diesen Datensätzen die Felder *VorgesNr* und *VorgesName* leer.

11.6 Datenaktualisierung bei Ein- und Mehrtabellenabfragen

Das Abfrageergebnis dient nicht nur dazu, auf seine Datensätze lesend zuzugreifen. Prinzipiell können die Datensätze eines Abfrageergebnisses auch bearbeitet werden, so daß die Änderungen an den Daten in der zugrundeliegenden Tabelle gespeichert werden. Dies schließt das Hinzufügen neuer Datensätze und das Löschen bestehender ein. Allerdings lassen sich nicht immer alle Felder in einem Abfrageergebnis aktualisieren. Einschränkungen ergeben sich, wenn das Abfrageergebnis auf mehr als einer Tabelle basiert und eine 1:n-Beziehung zwischen den Tabellen existiert. Im wesentlichen lassen sich dabei die folgenden Situationen unterscheiden:

Abfrage basiert auf einer Tabelle

Die Felder des Ergebnisses einer Eintabellen-Abfrage lassen sich aktualisieren. Einschränkungen ergeben sich bei Ausblenden von Duplikaten, Kreuztabellenabfragen und aggregierten Berechnungen.

Abfrage basiert auf mehr als einer Tabelle

Bei einer 1:1-Beziehung sind die Felder beider Seiten aktualisierbar. Im Falle einer 1:n-Beziehung der beiden Tabellen lassen sich alle Felder mit Ausnahme des Verknüpfungsfeldes der 1-Seite aktualisieren. Wenn dabei jedoch im Abfrageergebnis für den betreffenden Datensatz kein Feld der n-Seite enthalten ist, läßt sich auch das Verknüpfungsfeld der 1-Seite aktualisieren. Ferner können Sie das Verknüpfungsfeld der 1-Seite aktualisieren, wenn es Primärschlüssel einer 1:n-Beziehung mit referentieller Integrität und eingeschalteter Option *Aktualisierungsweitergabe an Detaildatensatz* ist.

Wenn Sie drei oder mehr Tabellen haben, von denen die erste zur zweiten eine 1:n-Beziehung und die zweite zur dritten ebenfalls eine 1:n-Beziehung hat, gilt Entsprechendes, wie im letzten Absatz für zwei Tabellen ausgeführt.

Unter gewissen Bedingungen können weitere Einschränkungen bezüglich der Aktualisierbarkeit von Abfrageergebnissen bestehen. Die Online-Hilfe liefert dazu recht ausführliche Informationen: Suchen Sie unter dem Stichwort *Aktualisieren von Daten, Abfrageergebnisse*, Thema *Wann können Daten aus einer Abfrage aktualisiert werden?*.

Kapitel 12

Komplexe Abfragen

Dieses Kapitel setzt voraus, daß Sie mit den Grundlagen von Abfragen vertraut sind. Falls dies nicht zutrifft, sollten Sie vollständig Kap. 10, *Abfragen: Grundlagen*, und die wesentlichen Teile von Kap. 11, *Auswahlabfragen*, lesen.

12.1 Parameterabfragen

Anders als der Name suggeriert, handelt es sich bei Parameterabfragen eigentlich nicht um einen eigenen Abfragetyp, sondern Sie können Parameter in jeder Abfrageart verwenden. Die Verwendung von Parametern ist sinnvoll, wenn Sie bestimmte Angaben, die in einer Abfrage berücksichtigt werden sollen (beispielsweise Werte für Kriterien oder Berechnungen), nicht als Konstanten angeben, sondern diese Werte vor jeder Berechnung des Abfrageergebnisses von Access neu abfragen lassen wollen. Auf diese Weise können Sie Abfragen in allgemeiner Weise formulieren, ohne Makros oder VBA-Prozeduren verwenden zu müssen.

12.1.1 Parameter angeben

Beispiel

Aufgabe. Erstellen Sie eine Abfrage mit den Feldern *Firma*, *Kontaktperson* und *Telefon* der Tabelle *Kunden* in der Datenbank *Nordwind.mdb*. Als Kriterium soll für das Feld *Firma* der Firmenname dienen, wobei beliebig viele Anfangszeichen reichen sollen. Die konkrete Zeichenfolge soll vor jeder Ermittlung des Abfrageergebnisses von Access erfragt werden. Gehen Sie zur Lösung dieser Aufgabe wie folgt vor:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*.
- Erstellen Sie eine neue Abfrage, und fügen Sie dieser die Tabelle *Kunden* hinzu.

- Ziehen Sie aus der Feldliste die Felder *Firma*, *Kontaktperson* und *Telefon* in den Entwurfsbereich.
- Geben Sie als Kriterium für das Feld Firma den folgenden Ausdruck an:

  Wie [Anfang des Firmennamens:] & »*«
- Ihr Abfrageentwurf sollte jetzt so aussehen wie in Bild 12.1 wiedergegeben.

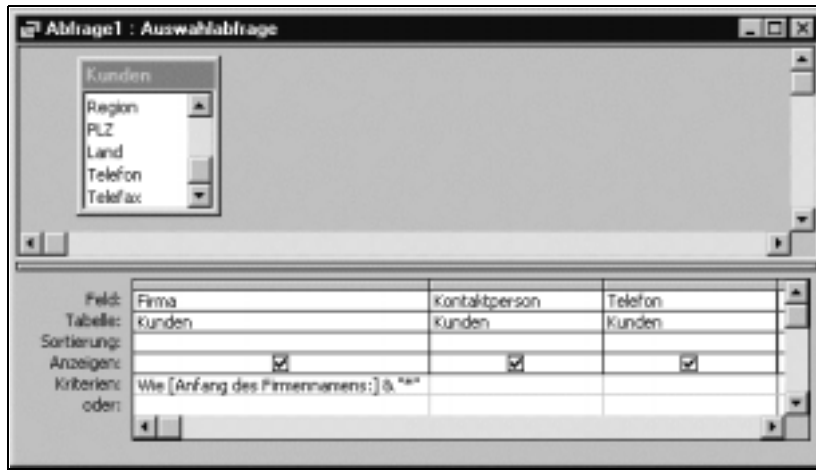


Bild 12.1: Der Kriterienausdruck enthält den Parameter »Anfang des Firmennamens: «.

Der als Kriterium angegebene Ausdruck ist folgendermaßen zu verstehen: Wenn Sie starr nach Datensätzen suchen würden, deren Firmenname beispielsweise mit der Zeichenfolge *bo* beginnt (Groß- und Kleinschreibung soll nicht unterschieden werden), würden Sie als Kriterium den Ausdruck

Wie "bo*"

angeben. Die Zeichenfolge *bo* soll nun durch einen Parameter ersetzt werden.

Ein Parameter in Abfragen ist eine beliebige Zeichenfolge, die zwischen eckigen Klammern steht; die Zeichenfolge darf allerdings nicht einem in der Abfrage vorkommenden Feldnamen oder einem reservierten Wort gleichen.

Daher fungiert der Teil *[Anfang des Firmennamens:]* im Kriterienausdruck des obigen Beispiels als Parameter. Access führt den Parameternamen im abfragenden Dialogfeld an, daher wurde der Name so gewählt, daß er im Dialogfeld gleichzeitig eine auffordernde Bedeutung hat (vgl. das Dialogfeld *Parameterwerte eingeben* in Bild 12.2). Der Verkettungsoperator *&* verkettet den Parameter mit dem Zeichen * zu der für den *Wie*-Operator anzugebenden Zeichenkette. Wenn Sie dann beispielsweise als Parameter die Zeichenfolge *bo* angeben, resultiert daraus die Zeichenfolge *bo**, so daß der gesamte Kriterienausdruck in diesem konkreten Fall lautet: *Wie »bo*«*.

- Wechseln Sie nunmehr in die Datenblattansicht. Aus diesem Anlaß will Access das Abfrageergebnis erstellen, benötigt dafür jedoch den Wert des Parameters, welcher mit dem Dialogfeld *Parameterwerte eingeben* erfragt wird, vgl. Bild 12.2.

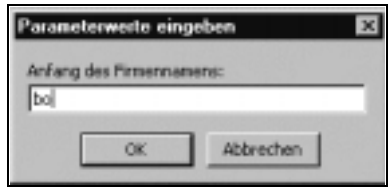


Bild 12.2: Dialogfeld »Parameterwerte eingeben«

In Bild 12.2 wurde die Zeichenfolge *bo* als Parameter in das Dialogfeld *Parameterwerte eingeben* eingegeben. Das daraufhin ermittelte Abfrageergebnis zeigt in der Datenblattansicht die in Bild 12.3 wiedergegebenen Daten.

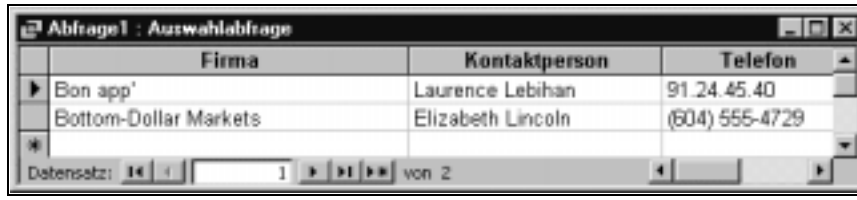


Bild 12.3: Datenblattansicht der Abfrage aus Bild 12.1 für den Parameterwert »bo«

Statt mit nur einem Parameter können Sie mit vielen in einem oder mehreren Kriterien arbeiten. Access erfragt sie alle vor dem jeweiligen Neuberechnen eines Abfrageergebnisses. Dabei können Sie die Reihenfolge, in der die Parameter abgefragt werden, bestimmen, vgl. weiter unten in diesem Abschnitt, Punkt 12.1.2, *Reihenfolge und Datentypprüfung bei der Parametereingabe festlegen*.

Parameter in anderen als Kriterienausdrücken

Obwohl Ausdrücke, die Sie zur Bestimmung von Kriterien angeben, der häufigste Anwendungsfall für Parameter sein dürften, ist deren Einsatz keinesfalls auf Kriterien beschränkt. Im Prinzip können Sie Parameter in einer Abfrage in jedem Ausdruck, in dem Sie sonst einen Wert auf andere Weise spezifizieren (sei es als Konstante, sei es als Feldname), anführen. Das folgende Beispiel zeigt, wie ein Parameter in einem berechneten Feld verwendet werden kann:

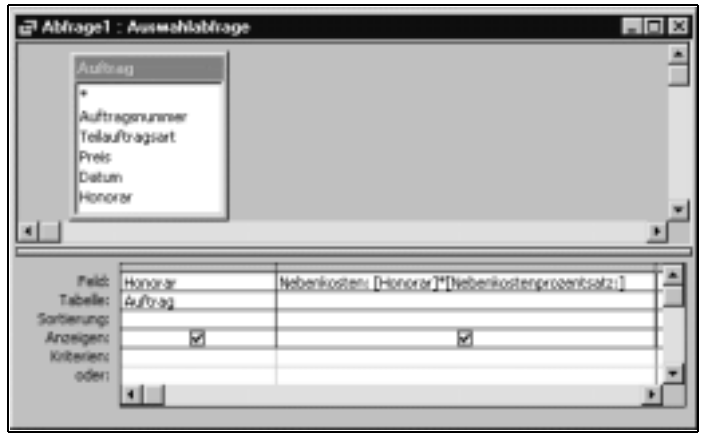


Bild 12.4: Im berechneten Feld der zweiten Spalte ist »Nebenkostenprozentsatz: « ein Parameter, weil diese Zeichenfolge nicht als Feldname in einer Feldliste der Abfrage vorkommt.

Hier wird das Feld *Nebenkosten* als berechnetes Feld definiert und durch den Ausdruck

[Honorar]*[Nebenkostenprozentsatz:]

berechnet. Dabei ist *Honorar* ein Feld der der Abfrage zugrundeliegenden Tabelle. Da andererseits ein Feld mit dem Namen *Nebenkostenprozentsatz:* in der Abfrage nicht vorkommt, interpretiert Access diese Zeichenfolge als Parameter und fragt dessen Wert vor dem Ausführen der Abfrage ab.

Parameterabfrage als Reaktion auf unbekannte Namen

Access interpretiert, wie bereits gesagt, jede zwischen eckigen Klammern stehende Zeichenfolge, die nicht als Feldname erkannt wird, als Parameter. Daher werden Sie bei einer Abfrageaktualisierung auch nach einem Parameter gefragt, wenn vielleicht nur ein Tippfehler bezüglich eines Feldnamens vorliegt.

Ein anderer Fall, bei dem Sie möglicherweise überraschend nach einem Parameter gefragt werden, kann der folgende sein: Sie haben als Kriterium zu einem Feld *AuftragsNr* in der Abfrage beispielsweise den Ausdruck

Formulare!Hauptaufträge!AuftragsNr

angegeben, um die Datensätze im Abfrageergebnis stets auf die jeweils im Formular *Hauptaufträge* aktuelle AuftragsNr zu begrenzen. Solange das Formular *Hauptaufträge* geöffnet ist, wird die Abfrage ohne weiteres ausgeführt. Wenn Sie es jedoch geschlossen haben und die Abfrage dann ausführen, fragt Access Sie nach dem Parameter *Formulare!Hauptaufträge!AuftragsNr.*

12.1.2 Reihenfolge und Datentypprüfung bei der Parametereingabe festlegen

Die Reihenfolge, in der die Parameter (falls mehrere angegeben sind) abgefragt werden, wird von Access bestimmt, es sei denn, Sie legen diese ausdrücklich fest. Dies ermöglicht Ihnen das Dialogfeld *Abfrageparameter*, das Sie mit dem Befehl *Parameter* aus dem Menü *Abfrage* aufrufen können. Es ist in Bild 12.5 wiedergegeben.

Bild 12.5: Dialogfeld »Abfrageparameter«

In die einzelnen Zellen der Spalte *Parameter* geben Sie die in der Abfrage festgelegten (oder festzulegenden) Parameter an, hier jedoch ohne die im Entwurfsbereich erforderlichen eckigen Klammern, ansonsten aber buchstäblich übereinstimmend. Für jeden Parameter wird außerdem ein Datentyp festgelegt. Dies hat den Vorteil, daß beim Erfragen der Parameterwerte jeweils geprüft wird, ob der eingegebene Wert dem angegebenen Datentyp entspricht. Trifft dies nicht zu, wird die Eingabe nicht angenommen, worauf eine entsprechende Meldung hinweist. Dieses Verfahren erhöht die Sicherheit der Abfrageergebnisse.

12.2 Abfragen mit Kriterien aus Steuerelementen in einem Formular (Query-by-Form)

Die Angabe von Kriterien im Entwurfsbereich einer Abfrage ist für den im Umgang mit Access Erfahrenen zwar nicht schwierig vorzunehmen, dennoch kann man sich für Standard-Suchroutinen mit wechselnden Kriterieninhalten bequemere Formen vorstellen. Erst recht gilt dieses für den mit Access wenig Erfahrenen, für dessen Suchroutinen praktisch nur die Angabe von Suchkriterien in Steuerelementen von Formularen in Betracht kommt. In diesem Abschnitt möchte ich Ihnen zeigen, wie Sie auf relativ einfache Weise Suchvorgänge sehr leistungsfähig mit Formularen organisieren kön-

nen. Dabei werden Grundkenntnisse über den Umgang mit Formularen vorausgesetzt, die Sie sich ggf. in Kap. 15, *Formulare: Grundlagen*, und Kap. 16, *Standard-Steuerelemente in Formularen*, aneignen können. Das ausgearbeitete Beispiel, das Sie in der Datenbank *Adressen.mdb* der Begleit-CD-ROM finden, verwendet auch kleine Makros, so daß insoweit (sehr geringe) Kenntnisse über Makros vorausgesetzt werden.

12.2.1 Abfragekriterien aus Steuerelementen eines Formulars übernehmen

Als Kriterium für eine Abfrage können Sie jeden in Access möglichen Ausdruck angeben, der einen interpretierbaren Wert für das Kriterium ergibt. Da Sie sich in einem Access-Ausdruck auch auf Steuerelemente eines Formulars beziehen können, liegt es nahe, Suchkriterien aus Steuerelementen von Formularen zu übernehmen. Diese Vorgehensweise hat gegenüber der direkten Angabe von Kriterien im Entwurfsbereich einer Abfrage nicht nur den Vorteil größerer Bequemlichkeit und Sicherheit, sondern sie erlaubt auch eine flexiblere Angabe von Kriterien.

Beispiel

Aufgabe. Für die Datenbank *Adressen.mdb* der Begleit-CD-ROM soll eine Abfrage erstellt werden, welche eine Adressenliste auf Basis von je einem Suchkriterium für *Firma*, *Vorname* und *Name* erstellt. Die genannten Suchkriterien sollen in Textfelder eines Formulars eingegeben und von der Abfrage als Kriterien übernommen werden. Gehen Sie folgendermaßen vor, um diese Aufgabe zu lösen:

Bild 12.6: Dialogfeld »Neues Formular«

- Öffnen Sie ggf. die Datenbank *Adressen.mdb*.
- Erstellen Sie ein neues Formular. Geben Sie für dieses Formular keine Datenherkunft an: Wählen Sie im Dialogfeld *Neues Formular* (vgl. Bild 12.6) im Listenfeld den Eintrag *Entwurfsansicht*, und bestätigen Sie mit OK. Dann erhalten Sie ein leeres Formular in der Entwurfsansicht.

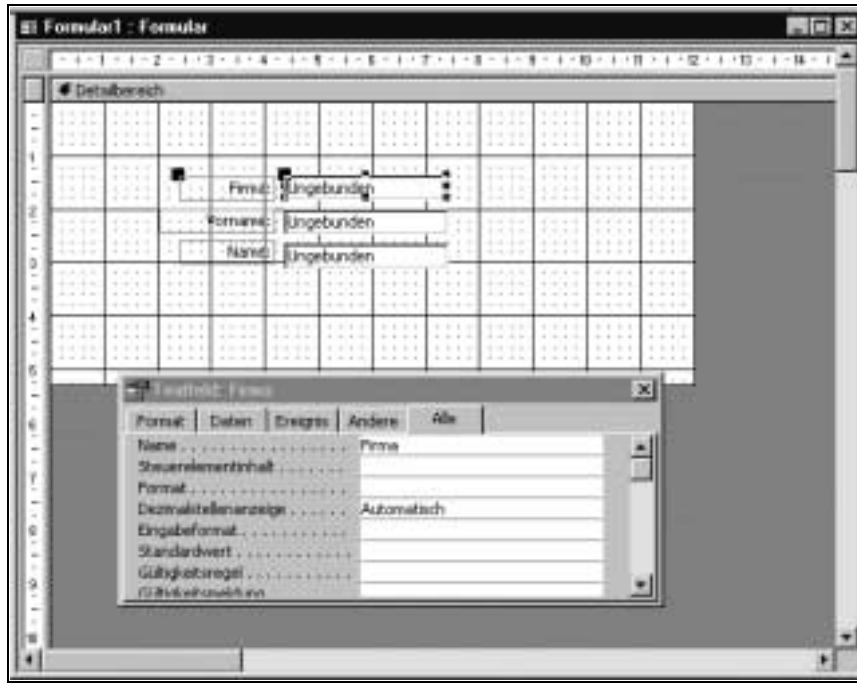


Bild 12.7: So könnte das fertige Formular in der Entwurfsansicht aussehen. In der wiedergegebenen Situation ist gerade das oberste Textfeld markiert. Das Eigenschaftenfenster zeigt daher für dieses Textfeld an, daß ihm der Name »Firma« zugewiesen wurde.

- Fügen Sie dem Formularentwurf drei Textfelder ein, denen Sie die Steuerelementnamen *Firma*, *Vorname* und *Name* geben. Geben Sie für diese Textfelder keinen Steuerelementinhalt an, so daß sie ungebundene Steuerelemente sind. In Bild 12.7 ist zu erkennen, daß das oberste Steuerelement *Textfeld* markiert ist. Daher gibt das Eigenschaftenfenster, das Sie ggf. durch Klicken auf die Symbol-Schaltfläche *Eigenschaften* einblenden müssen, die Einstellungen der Eigenschaften für dieses Textfeld wieder. Sie können erkennen, daß es vom Benutzer den Steuerelementnamen *Firma* bekommen hat und daß zur Eigenschaft *Steuerelementinhalt* kein Eintrag vorgenommen wurde.
- Speichern Sie das Formular unter dem Namen *Frm3Suchfelder* (oder einem anderen Namen).

- Wechseln Sie zum Datenbankfenster, aktivieren Sie die Registerkarte *Abfragen*, erstellen Sie eine neue Abfrage, und fügen Sie dieser die Tabelle *Adressen* hinzu.
- Fügen Sie im Entwurfsbereich alle Felder der Feldliste ein, mit Ausnahme des Feldes *AdrNr.*
- Geben Sie als Kriterien für die Felder *Firma*, *Vorname* und *Name* den jeweils zutreffenden der folgenden drei Ausdrücke ein:

  Wie [Formulare]![Frm3Suchfelder]![Firma] & "*"
  Wie [Formulare]![Frm3Suchfelder]![Vorname] & "*"
  Wie [Formulare]![Frm3Suchfelder]![Name] & "*"

  Jeder dieser Ausdrücke leistet für das betreffende Feld das folgende: Die im Textfeld des Formulars angegebene Zeichenfolge wird mit dem Zeichen * verkettet, und dieser Zeichenkette ist der Vergleichsoperator *Wie* vorangestellt. Wenn beispielsweise für *Firma* die Zeichenkette *zoo* angegeben ist, erzeugt der erste der drei angeführten Ausdrücke das Kriterium

  Wie "zoo*"

  Bild 12.8 zeigt, wie der Abfrageentwurf für die Felder *Firma*, *Vorname* und *Name* jetzt aussehen sollte.

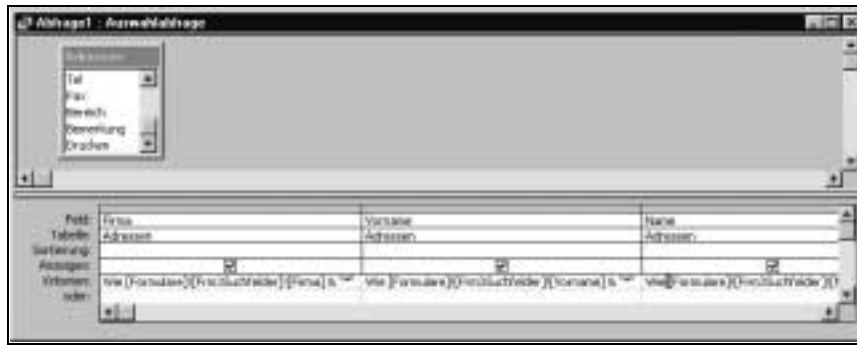


Bild 12.8: Abfrageentwurf mit Kriterien, die sich auf Felder des Formulars »Frm3Suchen« beziehen

- Aktivieren Sie das Formular *Frm3Suchfelder*, und wechseln Sie von der Entwurfsansicht in die Formularansicht. Falls Sie das Formular inzwischen geschlossen hatten: Öffnen Sie es in der Formularansicht. Geben Sie in die Felder *Firma*, *Vorname* bzw. *Name* die Texte *zoo*, *elke* bzw. *mü* ein, vgl. Bild 12.9. Klicken Sie nach der letzten Eingabe noch einmal auf ein anderes Feld, damit der Text, der zuletzt eingegeben wurde, vom betreffenden Textfeld angenommen wird; andernfalls befände sich das Textfeld noch im Bearbeitungsmodus.

Bild 12.9: Formular »Frm3Suchfelder« in der Formularansicht mit eingegebenen Suchkriterien für die drei Suchfelder

- Aktivieren Sie die erstellte Abfrage, und wechseln Sie in die Datenblattansicht. Diese sollte so aussehen, wie in Bild 12.10 wiedergegeben: Es wurde ein Datensatz gefunden, der den im Formular angegebenen Suchkriterien entspricht.

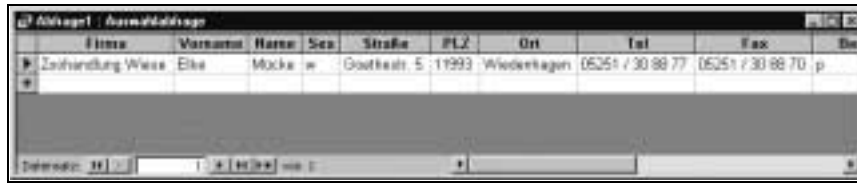


Bild 12.10: Abfrageergebnis für die drei im Formular (vgl. Bild 12.9) angegebenen Suchkriterien

Problem bei Nullwerten in den Datensätzen

Im vorangehenden Beispiel wurde für jedes Suchfeld ein Kriterium angegeben. Nicht immer jedoch möchten oder können Sie für jedes der im Formular (hier: das Formular *Frm3Suchfelder*) angebotenen Suchfelder ein Kriterium angeben. Beispielsweise könnte es sein, daß Sie zwar den Anfang des Vornamens und des Namens der gesuchten Adresse wissen, jedoch nichts über den Firmennamen. Dann möchten Sie vermutlich das Suchfeld *Firma* leer lassen. Prüfen Sie die Wirkung einer derartigen Eingabe für das Abfrageergebnis auf die folgende Weise:

- Aktivieren Sie das Formular mit den drei Suchfeldern, löschen Sie in der Formularansicht den Eintrag im Suchfeld *Firma*, und geben Sie für *Vorname* bzw. *Name* die Zeichenfolgen *Franz* bzw. *Barsch* ein (vgl. Bild 12.11). Klicken Sie auf ein anderes Steuerelement im Formular, um die letzte Eingabe wirksam zu machen.
- Wechseln Sie zum Abfragefenster der neu erstellten Abfrage, und wechseln Sie darin zunächst in die Entwurfsansicht und danach wieder in die Datenblattansicht, damit die Abfrage aktualisiert wird. Als Ergebnis erhalten Sie ein leeres Abfrageergebnis, vgl. Bild 12.12.

Bild 12.11: Formular »Frm3Suchfelder« in der Formularansicht mit Suchkriterien für nur zwei der drei Suchfelder

Bild 12.12: Leeres Abfrageergebnis auf Basis der Angaben in Bild 12.11

Das Abfrageergebnis auf Basis der beiden für *Vorname* und *Name* angegebenen Suchkriterien bleibt leer, obwohl es in der Tabelle *Adressen*, wie Sie leicht nachprüfen können, einen Datensatz mit einem *Franz Barsch* gibt. Diese vielleicht überraschende, jedenfalls hier unerwünschte Wirkung erklärt sich folgendermaßen: Für die Adresse *Franz Barsch* ist keine Firma angegeben, das entsprechende Feld in der Tabelle *Adressen* hat daher einen Nullwert. Wenn Sie in einer Abfrage für ein Feld ein Kriterium angeben, dann gibt die Abfrage nur die Datensätze wieder, für die dieses Feld keinen Nullwert enthält. Obwohl im Formular für das Suchfeld *Firma* kein Wert eingetragen wurde, ist für die Abfrage gleichwohl für das Feld *Firma* ein Kriterienausdruck – wenn auch in allgemeiner Form, was jedoch insoweit unbedeutend ist – angegeben. Daher werden im Ergebnis der bisher erstellten Abfrage überhaupt nur solche Datensätze wiedergegeben, die in keinem der drei Felder *Firma*, *Vorname* und *Name*, für die im Entwurfsbereich Kriterien angegeben sind, einen Nullwert haben.

Lösung des Problems bei Nullwerten in den Datensätzen

Eine Lösung des im letzten Punkt geschilderten Problems könnte darin bestehen, von vornherein keine Nullwerte in den Feldern, für die jemals Suchkriterien relevant sein könnten, zuzulassen. Dann jedoch müssen Sie in diese Felder, wenn Sie keine anderen Angaben dafür kennen, jeweils die leere Zeichenfolge »« eingeben. Mit leeren Zeichenfolgen entsteht das Problem der Unterdrückung von Datensätzen im Abfrageergebnis nicht. Dieser Weg allerdings würde den Eingabeaufwand deutlich erhöhen, weil Sie bei der

Dateneingabe jedes Feld ansprechen müßten, statt Felder, für die Sie keinen Wert kennen oder vergeben wollen, einfach frei zu lassen. Dieser Preis erscheint zu hoch.

Einen Ausweg bieten berechnete Felder: Sie können den Inhalt der Felder, für die Sie Kriterien angeben wollen, in der Abfrage in berechneten Feldern ausgeben lassen, wobei Sie Nullwerte in leere Zeichenfolgen umwandeln. Verändern Sie den Abfrageentwurf folgendermaßen:

Geben Sie anstelle der Feldnamen *Firma*, *Vorname* und *Name* jeweils den zutreffenden der folgenden Ausdrücke ein:

Firma: Wenn(IstNull([Adressen].[Firma]);" ";[Adressen].[Firma])
Vorname: Wenn(IstNull([Adressen].[Vorname]);" ";[Adressen].
[Vorname])
Name: Wenn(IstNull([Adressen].[Name]);" ";[Adressen].[Name])

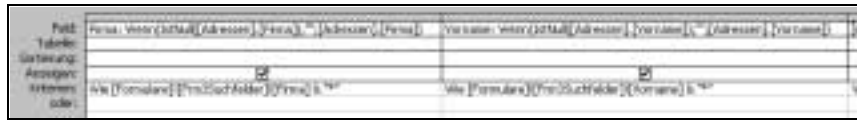


Bild 12.13: Veränderter Abfrageentwurf mit nunmehr berechneten Feldern, damit die Datensätze leere Zeichenfolgen statt Nullwerte enthalten.

Bild 12.13 zeigt die veränderte Abfrage in der Entwurfsansicht. Die drei Ausdrücke erzeugen nicht nur ein berechnetes Feld, sondern sie benennen die berechneten Felder auch gleichzeitig mit den Namen *Firma*, *Vorname* und *Name*. Aus diesem Grunde muß der Bezug auf die entsprechenden Felder im Ausdruck auch etwas umständlich in der Form *[Adressen].[Firma]* usw. angegeben werden. Die berechneten Felder enthalten die textlichen Angaben aus den entsprechenden Tabellenfeldern und anstelle von Nullwerten leere Zeichenfolgen. Dies wird mit der *Wenn*-Funktion erreicht. Da die berechneten Felder die Namen *Firma*, *Vorname* und *Name* bekommen haben, sind die Spaltenüberschriften im Abfrageergebnis auch nach dieser Änderung richtig.

Um die Wirkung des veränderten Abfrageentwurfs zu prüfen, gehen Sie folgendermaßen vor:

- Vergewissern Sie sich, daß im Formular *Frm3Suchfelder* für *Vorname* bzw. *Name* die Zeichenfolgen *Franz* bzw. *Barsch* angegeben sind, vgl. oben Bild 12.11.
- Aktivieren Sie ggf. die veränderte Abfrage, und wechseln Sie von der Entwurfsansicht in die Datenblattansicht. Nunmehr sollte das Abfrageergebnis den Datensatz wiedergeben, der in Bild 12.14 zu sehen ist.

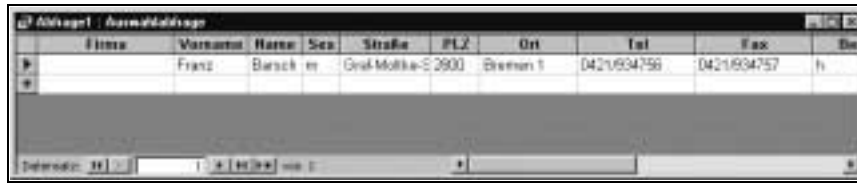


Bild 12.14: Das Abfrageergebnis zeigt einen Datensatz mit einem leeren Eintrag im Feld »Firma«, für das ein Kriterienausdruck angegeben ist.

12.2.2 Beispiel einer kompletten Formularsteuerung der Suchvorgänge (Query-by-Form)

Die im vorangehenden Punkt dargestellte Arbeitsweise ermöglicht es, Suchkriterien für eine Abfrage in ein Formular einzugeben. Damit ist gegenüber der Eingabe von Kriterienausdrücken in den Entwurfsbereich einer Abfrage bereits ein großes Stück an Komfort und Sicherheit gewonnen. Die bisher skizzierte Vorgehensweise hat jedoch noch den Nachteil, daß Sie nach der Eingabe der Suchkriterien die Abfrage in der Datenblattansicht über einen geeigneten Menübefehl oder mit der entsprechenden Symbol-Schaltfläche öffnen müssen. Nach jeder Änderung der Suchkriterien im Formular müssen Sie die Abfrage wieder schließen oder in deren Entwurfsansicht wechseln, damit sie anschließend wieder in der Datenblattansicht geöffnet werden kann. Für Routinevorgänge erscheint diese Arbeitsweise zu kompliziert und unbequem. Die Suchvorgänge lassen sich jedoch erheblich leichter ausführen, wenn Sie zum Darstellen des Suchergebnisses ebenfalls ein Formular verwenden. Darüber hinaus wird der Umgang mit den Formularen erleichtert, wenn Sie zum Wechseln zwischen Formularen Befehlsschaltflächen in den Formularen verwenden.

Ein derartiges Modell ist in der Datenbank *Adressen.mdb* von der Begleit-CD-ROM realisiert. Der entscheidende Unterschied des dortigen Suchmodells gegenüber dem im vorangehenden Punkt 12.2.1, *Abfragekriterien aus Steuerelementen eines Formulars übernehmen*, entwickelten Modell liegt darin, daß das Suchergebnis nicht in der Datenblattansicht der Abfrage, sondern in einem eigenen Formular dargestellt wird. Diesem Formular liegt als Datenbasis die Suchabfrage zugrunde, welche selbst zu keinem Zeitpunkt geöffnet werden muß.

Die strukturellen Zusammenhänge der diesbezüglichen Objekte in der Datenbank Adressen.mdb sind in Bild 12.15 skizzenhaft dargestellt. Der Zusammenhang und die einzelnen Objekte stellen sich in Bild 12.16 dar.

Formular »AdressenSuchen«. Dieses Formular enthält Textfelder zur Angabe der Suchkriterien. Darüber hinaus weist es die beiden Befehlsschaltflächen *Suchen* und *Schließen* auf: Mit *Suchen* wird das Formular *Adressen-Suchergebnis*, welches die Suchergebnisse wiedergibt, aufgerufen; mit *Schließen* schließt das Formular sich selbst. Das Formular *AdressenSuchen* wird vom Datenbankfenster aus geöffnet.

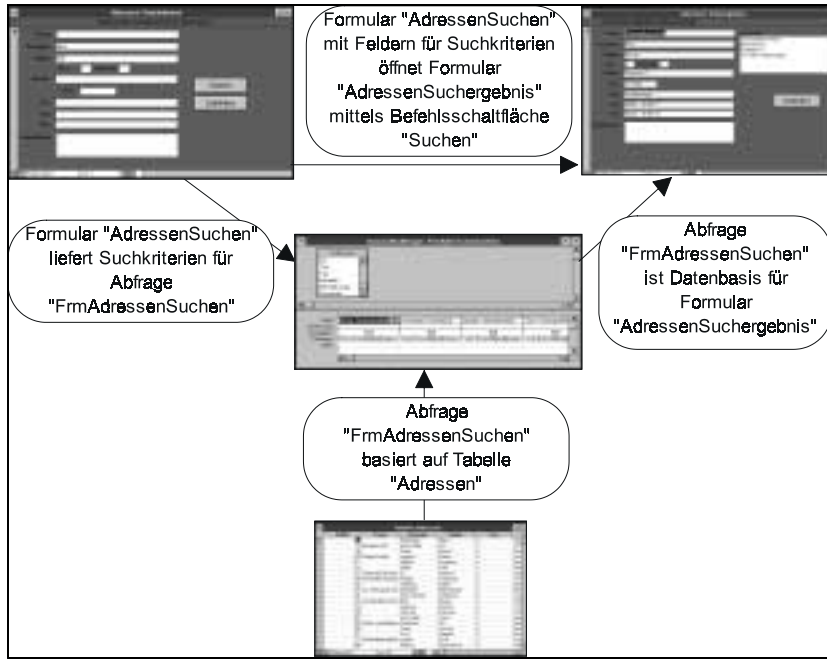


Bild 12.15: Objektstruktur der »Query-by-Form« in der Datenbank Adressen.mdb

Bild 12.16: Formular »AdressenSuchen« mit Textfeldern zur Eingabe von Suchkriterien und zwei Befehlsschaltflächen: Die Schaltfläche »Suchen« öffnet das Formular »AdressenSuchergebnis« als Dialogfeld, und die Schaltfläche »Schließen« schließt das Formular selbst. Im gezeigten Zustand wurden für Vorname bzw. Name als Suchkriterien die Zeichenfolgen »Elke« bzw. »mü« eingegeben.

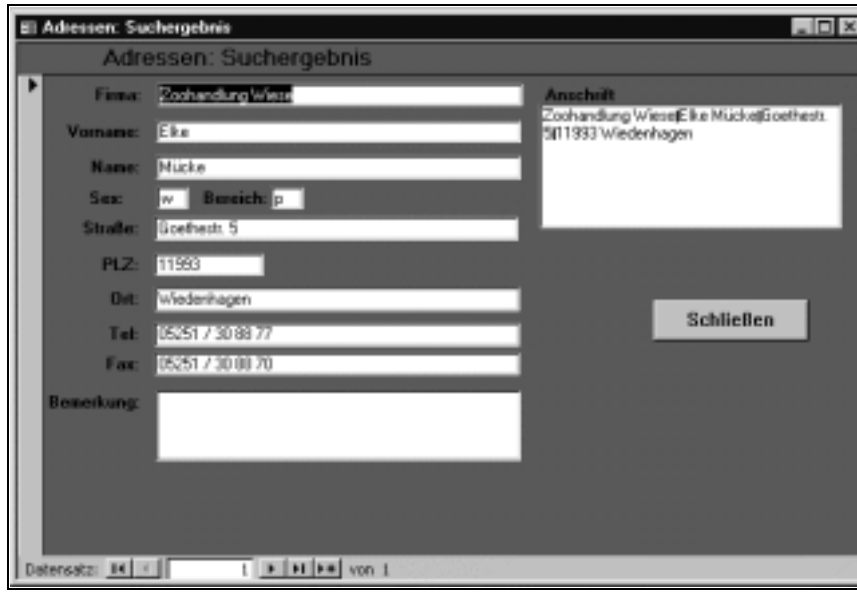


Bild 12.17: Formular »AdressenSuchergebnis« mit dem Suchergebnis für die in das Formular »AdressenSuchen« eingegebenen Suchkriterien, vgl. Bild 12.16.

Formular »AdressenSuchergebnis«. Dieses Formular hat als Datenbasis (Formulareigenschaft *Datenherkunft*) die Auswahlabfrage *FrmAdressenSuchen*, deren Abfrageergebnis die Suchergebnisse enthält. Jedesmal, wenn das Formular *AdressenSuchergebnis* geöffnet wird, wird das Ergebnis der Abfrage *FrmAdressenSuchen* aktualisiert. Das Formular *AdressenSuchergebnis* wird vom Formular *AdressenSuchen* aus über die Befehlsschaltfläche *Suchen* geöffnet. Mit dieser Befehlsschaltfläche ist ein entsprechendes Ereignismakro verbunden. Die Formulareigenschaften *PopUp* und *Gebunden* sind jeweils auf den Wert *Ja* eingestellt, wodurch das Formular zu einem Dialogfeld-Formular wird. Dies bedeutet, daß Sie so lange keine anderen Aufgaben in Access bearbeiten können, wie das Dialogfeld-Formular geöffnet ist. Sie müssen es daher schließen, um zum Ausgangsformular *AdressenSuchen* mit den Feldern für Suchkriterien zurückzukehren. Das Formular *AdressenSuchergebnis* enthält außer den Textfeldern zum Anzeigen der Felder des gefundenen Datensatzes das Textfeld *Anschrift*, in dem die Postanschrift als eine Zeichenkette mit Zeilenumbrüchen wiedergegeben wird. Diese Anschrift-Zeichenkette wird durch den folgenden Ausdruck ermittelt, der als Steuerelementinhalt für das Feld *Anschrift* angegeben ist:

```
=[Firma] & Zchn$(13) & [Vorname] & Wenn(IstNull([Vorname]);"";" ") &
[Formular]![Name] & Zchn$(13) & [Straße] & Zchn$(13) & [PLZ] & " " &
[Ort]
```

Sie können die Anschrift-Zeichenkette kopieren und in das Adreßfeld eines Briefes, beispielsweise im Textverarbeitungsprogramm Word für Windows, einfügen.

Wenn auf Basis der Suchkriterien mehr als ein Datensatz gefunden wird, kann mit Hilfe der Navigationsschaltflächen am unteren Fensterrand zu weiteren Datensätzen geblättert werden. Eine Bearbeitung der Textfelder im Formular ist nicht möglich, weil die entsprechenden Felder der Abfrage berechnete Felder sind.

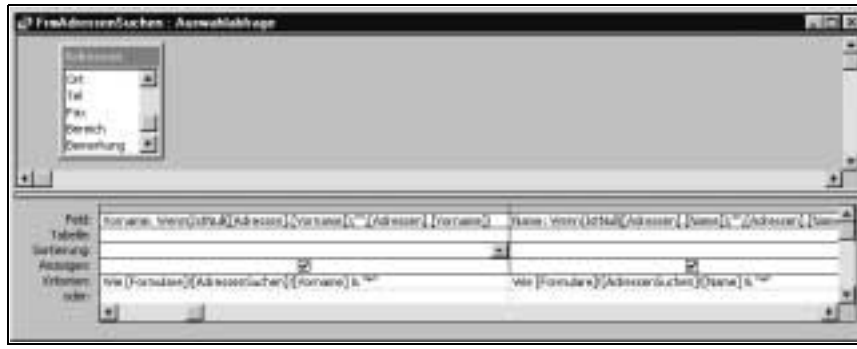


Bild 12.18: Auswahlabfrage »FrmAdressenSuchen« in der Entwurfsansicht. Die Felder für die Adreßinformationen sind als berechnete Felder definiert, in denen Nullwerte in leere Zeichenketten umgewandelt werden. Die Kriterienausdrücke nehmen Bezug auf die Textfelder des Formulars »AdressenSuchen« aus Bild 12.16.

Auswahlabfrage »FrmAdressenSuchen«. Diese Abfrage, die auf der Tabelle *Adressen* basiert, ist hinsichtlich der Kriterienausdrücke und der berechneten Felder ganz entsprechend konstruiert wie die Abfrage am Ende des vorangehenden Punktes 12.2.1, *Abfragekriterien aus Steuerelementen eines Formulars übernehmen*, vgl. oben Bild 12.13 und den erläuternden Text dazu: Adreßinformationen werden in berechneten Feldern ermittelt, um Nullwerte in leere Zeichenfolgen umzuwandeln. Die Kriterienausdrücke nehmen Bezug auf die Textfelder des Formulars *AdressenSuchen*. Beispielsweise lautet der Kriterienausdruck für das Feld *Firma*:

Wie [Formulare]![AdressenSuchen]![Firma] & "*"

Diese Abfrage braucht nie geöffnet zu werden, um wirksam zu sein. Es reicht, wenn das Formular *AdressenSuchergebnis*, dem diese Abfrage als Datenherkunft zugrunde liegt, geöffnet wird. Dann berechnet Access das Abfrageergebnis neu und stellt es dem Formular zur Verfügung, ohne die Abfrage zu öffnen.

12.3 Aggregierte Berechnungen

In Kap. 11, *Auswahlabfragen*, Punkt 11.1.6, *Berechnete Felder erstellen*, wurde gezeigt, wie Sie für eine Abfrage ein Feld hinzufügen, welches für jeden Datensatz des Abfrageergebnisses ein berechnetes Ergebnis ausweist. Auch im vorangehenden Abschnitt 12.2, *Abfragen mit Kriterien aus Steuerelementen in einem Formular (Query-by-Form)*, wurde intensiv Gebrauch von dieser Möglichkeit gemacht. Mit einem berechneten Feld können Sie beispielsweise die Mehrwertsteuer für jeden Datensatz des Abfrageergebnisses ausweisen. Bei einem berechneten Feld wird die Berechnung sozusagen *waagerecht*, d.h. jeweils *pro Datensatz*, ausgeführt. In diesem Abschnitt geht es um Berechnungen, deren Ergebnis eine Zusammenfassung in der *Senkrechten*, d.h. *pro Feld* für alle Datensätze (bzw. Gruppen daraus) darstellt. Die Summe aller Auftragswerte einer Abfrage stellt ein typisches Beispiel für eine aggregierte (= zusammenfassende) Berechnung dar. Die zusammenfassenden Berechnungen werden mit Hilfe der im Abfrageentwurf zur Verfügung stehenden *Aggregatfunktionen* durchgeführt.

Berechnung für alle Datensätze

Beispiel. Ermitteln Sie für die Abfrage *Rechnungen* der Datenbank *Nordwind.mdb* die gesamte Endpreissumme aller Bestellungen sowie den mittleren Endpreis je Bestellung. Sie lösen die Aufgabe folgendermaßen:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*.
- Erstellen Sie eine neue Abfrage aus dem Datenbankfenster heraus.
- Fügen Sie der Abfrage die Abfrage *Rechnungen* hinzu, indem Sie im Dialogfeld *Tabelle anzeigen* die Registerkarte *Abfragen* wählen und die genannte Abfrage aus der Liste hinzufügen, vgl. Bild 12.19.

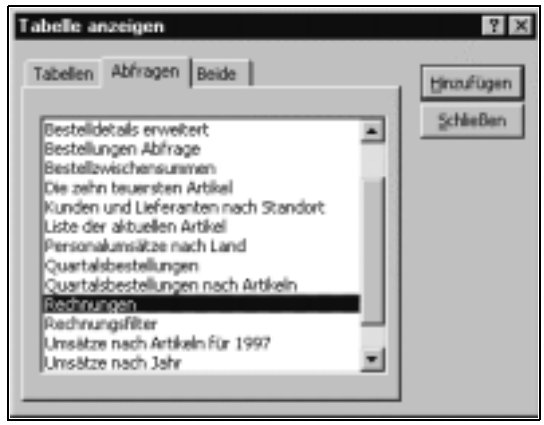


Bild 12.19: Im Dialogfeld »Tabelle anzeigen« werden alle Abfragen der Datenbank Nordwind.mdb angezeigt, weil die Registerkarte »Abfragen« gewählt ist.

- Ziehen Sie das Feld *Endpreis* zweimal aus der Feldliste in den Entwurfsbereich, so daß es in der Zeile *Feld* auch zweimal erscheint, vgl. Bild 12.20.
- Veranlassen Sie, daß eine aggregierte Berechnung durchgeführt werden kann: Klicken Sie auf die Symbol-Schaltfläche *Funktionen* (vgl. links nebenstehend), oder wählen Sie den gleichnamigen Befehl aus dem Menü *Ansicht*. Daraufhin erscheint im Entwurfsbereich die zusätzliche Zeile *Funktion*. Deren Zellen sind zunächst für jedes im Entwurf berücksichtigte Feld mit dem Eintrag *Gruppierung* beschriftet.
- Klicken Sie in die Zelle der Zeile *Funktion*, die sich auf das erste Feld mit dem Namen *Endpreis* bezieht, schlagen Sie das Dropdown-Listenfeld auf, und wählen Sie daraus die Funktion *Summe*.
- Klicken Sie in die Zelle der Zeile *Funktion*, die sich auf das zweite Feld mit dem Namen *Endpreis* bezieht, schlagen Sie das Dropdown-Listenfeld auf, und wählen Sie daraus die Funktion *Mittelwert*.

Ihr Abfrageentwurf sollte jetzt so aussehen wie der in Bild 12.20.

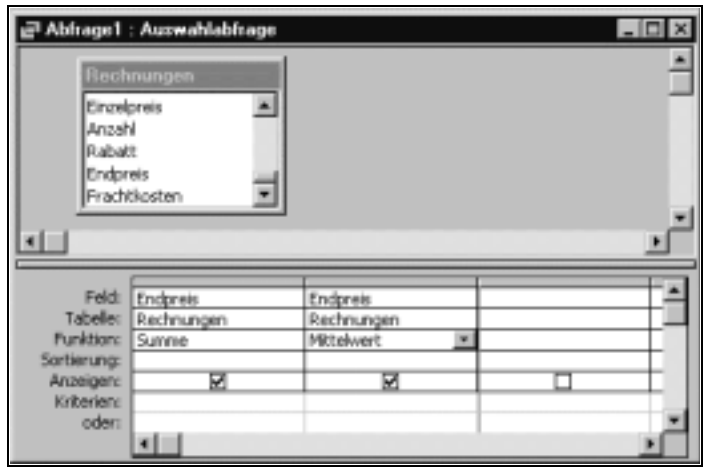


Bild 12.20: Abfrageentwurf zur Berechnung der Endpreissumme und des mittleren Endpreises

Das Ergebnis der zusammenfassenden Berechnung können Sie sehen, wenn Sie zur Datenblattansicht der Abfrage wechseln, vgl. Bild 12.21.

Berechnung für nach einem Feld gruppierte Datensätze

Im vorangehenden Beispiel wurden die Preissumme sowie der mittlere Preis für alle Datensätze der Abfrage zusammen berechnet. Daher wurde im Abfrageergebnis auch nur eine einzige Ergebniszeile wiedergegeben, vgl. Bild 12.21. Sie können aggregierte Berechnungen jedoch auch für Gruppen von Datensätzen ausführen. Dabei werden die Gruppen nach dem Inhalt eines

für die Gruppierung bestimmten Feldes gebildet (oder mehrerer Felder, vgl. den folgenden Punkt *Berechnung für nach mehreren Feldern gruppierte Datensätze*).

Bild 12.21: Datenblattansicht der aggregierenden Abfrage aus Bild 12.20. Die Spaltenüberschriften »Summe von Endpreis« und »Mittelwert von Endpreis« wurden von Access automatisch erzeugt.

Beispiel. Ermitteln Sie für die Abfrage *Rechnungen* der Datenbank *Nordwind.mdb* die Preissumme der Bestellungen sowie deren Durchschnitt, gruppiert nach den zugehörigen Mitarbeitern. Verfahren Sie zur Lösung der Aufgabe folgendermaßen:

- Nehmen Sie den Abfrageentwurf des vorangehenden Beispiels als Ausgangspunkt, vgl. Bild 12.20.
- Fügen Sie der Abfrage zusätzlich die beiden Tabellen *Bestellungen* und *Personal* hinzu. Weil zwischen den Tabellen Beziehungen definiert sind, fügt Access automatisch Verknüpfungslinien ein, vgl. Bild 12.22.

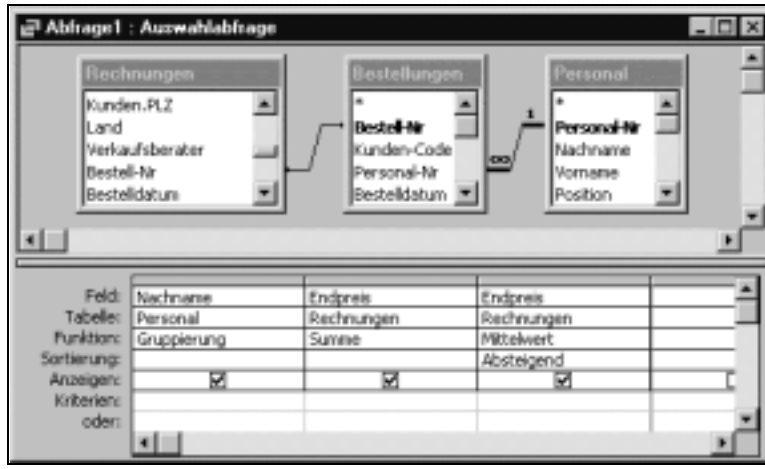


Bild 12.22: Abfrageentwurf zur Berechnung der Preissumme und des mittleren Preises, gruppiert nach Nachnamen der Mitarbeiter

- Ziehen Sie aus der Feldliste *Personal* das Feld *Nachname* in den Entwurfsbereich, am besten so, daß es vor den beiden Preis-Feldern angeordnet wird.

- In der Zeile *Funktion* steht dann für dieses Feld der Eintrag *Gruppierung*. Belassen Sie es dabei, um eine Gruppierung der Berechnungen nach dem Inhalt des Feldes *Nachname* zu bewirken.
- Lassen Sie die Abfrageergebnisse nach dem Durchschnittspreis in absteigender Reihenfolge sortieren: Klicken Sie in die Zelle der Zeile *Sortierung*, die sich auf das Feld *Endpreis* mit der Mittelwertberechnung bezieht, schlagen Sie das Dropdown-Listenfeld auf, und wählen Sie den Eintrag *Absteigend*.

Ihr Entwurfsbereich sollte jetzt aussehen, wie in Bild 12.22 wiedergegeben. Die Datenblattansicht dieser Abfrage zeigt die in Bild 12.23 dargestellten Ergebnisse.

Abfrage1 : Auswahlabfrage

Nachname	Summe von Endpreis	Mittelwert von Endpreis
Dodsworth	77.308,04 DM	722,51 DM
King	124.568,22 DM	707,77 DM
Fuller	166.537,75 DM	691,03 DM
Leverling	202.812,82 DM	631,82 DM
Buchanan	68.739,75 DM	587,52 DM
Davolio	192.107,57 DM	556,83 DM
Peacock	232.890,83 DM	554,50 DM
Callahan	126.862,27 DM	487,93 DM
Suyama	73.913,13 DM	439,96 DM

Datensatz: 1 von 9

Bild 12.23: Datenblattansicht des Abfrageentwurfs zur Berechnung der Preissumme und des mittleren Preises, gruppiert nach Nachnamen der Mitarbeiter, aus Bild 12.22.

Berechnung für nach mehreren Feldern gruppierte Datensätze

Das Gruppieren ist nicht nur auf ein Feld beschränkt. Vielmehr können Sie mehrere Felder zum Gruppieren angeben. Dann wird, ähnlich wie beim Sortieren nach mehreren Feldern, zunächst nach den Werten des am weitesten links stehenden Gruppierfeldes gruppiert, innerhalb der einzelnen Kategorien dieser Gruppierung dann nach den Werten des zweiten Gruppierfeldes etc.

Beispiel. Weisen Sie für jeden Mitarbeiter die gesamte und die mittlere Bestellsumme aus, die auf ihn für jedes Jahr entfällt. Entwickeln Sie dieses Beispiel der Einfachheit halber durch entsprechende Bearbeitung des vorangehenden Beispiels (vgl. den Abfrageentwurf oben in Bild 12.22).

Um nach Jahren gruppieren zu können, müssen Sie ein berechnetes Feld erstellen, das für die einzelnen Datensätze aus dem jeweiligen Bestelldatum die Jahreszahl ermittelt. Gehen Sie im einzelnen wie folgt vor:

- Löschen Sie den Eintrag *Absteigend* in der Spalte, in der der Mittelwert für den Endpreis ermittelt wird, denn sonst würde weiterhin in der Reihenfolge der Mittelwerte sortiert statt nach Mitarbeiternamen und Jahreszahlen.
- Belassen Sie es für das Feld *Nachname* bei dem Eintrag *Gruppierung* in der Zeile *Funktion*.
- Fügen Sie links neben dem Feld *Endpreis* mit der Funktion *Summe* eine neue Spalte ein: Markieren Sie diese Feldspalte *Endpreis*, und drücken Sie die [Einfg]-Taste.
- Schreiben Sie in die Zeile *Feld* der eingefügten, bisher noch leeren Spalte den folgenden Ausdruck:

  Bestelljahr: Jahr([Rechnungen].[Bestelldatum])
- Damit erstellen Sie ein berechnetes Feld, dem Sie den Namen *Bestelljahr* geben. Der Ausdruck *Jahr([Rechnungen].[Bestelldatum])* ermittelt die Jahreszahl aus dem Bestelldatum. Der Tabellenname *[Rechnungen]* muß dem Feldnamen *[Bestelldatum]* vorangestellt werden, weil letzterer in den drei Tabellen der Abfrage mehrfach vorkommt und andernfalls keine Eindeutigkeit vorläge.

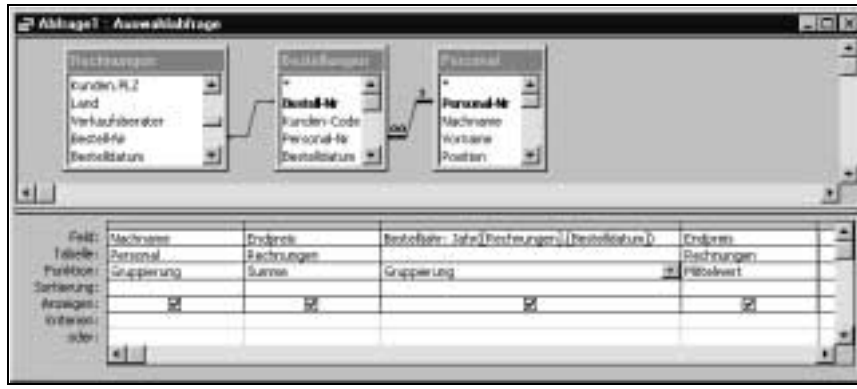


Bild 12.24: Abfrageentwurf zur Berechnung der Preissumme und des mittleren Preises, gruppiert nach Nachnamen der Mitarbeiter und dem Bestelljahr

Ihr Abfrageentwurf sollte jetzt aussehen wie der in Bild 12.24 wiedergegebene. Wechseln Sie zur Datenblattansicht, um das Abfrageergebnis zu betrachten. Bild 12.25 zeigt die ersten Datensätze. Sie erkennen, daß zunächst nach Mitarbeitern und innerhalb eines jeden Mitarbeiters nach Bestelljahren gruppiert wurde. Die gesamte und mittlere Bestellsumme (*Endpreis*) wird für jede Teilgruppe ausgewiesen.

Abfrage1 : Auswahlabfrage

Nachname	Summe von Endpreis	Bestelljahr	Mittelwert von Endpreis
Buchanan	18.331,42 DM	1996	678,94 DM
Buchanan	30.716,44 DM	1997	579,56 DM
Buchanan	19.691,89 DM	1998	532,21 DM
Callahan	22.240,12 DM	1996	453,88 DM
Callahan	56.032,60 DM	1997	451,88 DM
Callahan	48.589,55 DM	1998	558,50 DM
Davolio	35.764,51 DM	1996	510,92 DM
Davolio	93.148,04 DM	1997	597,10 DM
Davolio	63.195,02 DM	1998	531,05 DM
Dodsworth	9.894,51 DM	1996	618,41 DM
Dodsworth	26.310,39 DM	1997	684,68 DM
Dodsworth	41.103,14 DM	1998	893,55 DM
Fuller	21.757,06 DM	1996	543,93 DM
Fuller	70.444,14 DM	1997	690,63 DM
Fuller	74.336,55 DM	1998	750,87 DM

Datensatz: 1 von 27

Bild 12.25: Datenblattansicht der Abfrage zur Berechnung der Preissumme und des mittleren Preises, gruppiert nach Nachnamen der Mitarbeiter und dem Bestelljahr

Kriterien festlegen

Auch für Abfragen mit aggregierten Berechnungen können Kriterien angegeben werden, und zwar sowohl für Gruppierungsfelder wie auch für Felder mit aggregierten Berechnungen.

Kriterien für Gruppierungsfelder: Wenn Sie ein Kriterium für ein Feld mit der Angabe *Gruppierung* in der Zeile *Funktion* angeben, wirkt dies im Prinzip genauso wie bei anderen Abfragen. Im Abfrageentwurf, den Sie in Bild 12.26 sehen, ist dem Abfrageentwurf aus Bild 12.24 für das Bestelljahr ein Kriterium hinzugefügt worden, welches die Datensätze auf die Bestelljahre 1997 und 1998 beschränkt.

Feld:	Nachname	Endpreis	Bestelljahr: Jahr([Rechnungen].[BestellMatum])	Endpreis
Tabelle:	Personal	Rechnungen		Rechnungen
Funktion:	Gruppierung	Summe	Gruppierung	Mittelwert
Sortierung:				
Anzeigen:	☑	☑	☑	☑
Kriterien:			1997 Oder 1998	
oder:				

Bild 12.26: Angabe eines Kriteriums für ein Gruppierungsfeld

Das Abfrageergebnis zum Abfrageentwurf aus Bild 12.26 ist in Bild 12.27 wiedergegeben. Es weist bezüglich der Bestelljahre nur die Untergruppen 1997 und 1998 aus.

Kriterien für ein Feld mit aggregierter Berechnung: Erst berechnen, dann Datensätze einschränken. Im Abfrageentwurf in Bild 12.28 ist für das Feld *Endpreis*, für das die Summe ermittelt wird, als Kriterium >100000 angegeben. Das Ergebnis dieser Beschränkung sehen Sie in Bild 12.29, welches das vollständige Abfrageergebnis wiedergibt.

Abfrage1 : Auswahlabfrage

Nachname	Summe von Endpreis	Bestelljahr	Mittelwert von Endpreis
Buchanan	30.716,44 DM	1997	579,56 DM
Buchanan	19.691,89 DM	1998	532,21 DM
Callahan	56.032,60 DM	1997	451,88 DM
Callahan	48.589,55 DM	1998	558,50 DM
Davolio	93.148,04 DM	1997	597,10 DM
Davolio	63.195,02 DM	1998	531,05 DM
Dodsworth	26.310,39 DM	1997	584,68 DM
Dodsworth	41.103,14 DM	1998	893,55 DM
Fuller	70.444,14 DM	1997	690,63 DM
Fuller	74.336,55 DM	1998	750,87 DM
King	60.471,19 DM	1997	664,52 DM
King	48.864,87 DM	1998	814,41 DM
Leverling	108.026,13 DM	1997	587,10 DM
Leverling	76.562,73 DM	1998	814,50 DM
Peacock	128.809,78 DM	1997	590,87 DM
[illegible]	[illegible]	[illegible]	[illegible]

Datensatz: 1 von 18

Bild 12.27: Datenblattansicht des Abfrageentwurfs aus Bild 12.26. Es werden nur (aggregierte) Daten für die Bestelljahre 1997 und 1998 ausgewiesen.

Feld:	Nachname	Endpreis
Tabelle:	Personal	Rechnungen
Funktion:	Gruppierung	Summe
Sortierung:		
Anzeigen:	☑	☑
Kriterien:		>100000
oder:		

Bild 12.28: Für das Feld mit der Aggregatfunktion »Summe« ist ein Kriterium angegeben. Daher wird erst berechnet, und danach werden die Gruppensummen entsprechend dem Kriterium ausgewählt.

Abfrage1 : Auswahlabfrage

Nachname	Summe von Endpreis
Callahan	126.862,27 DM
Davolio	192.107,57 DM
Fuller	166.537,75 DM
King	124.568,22 DM
Leverling	202.812,82 DM
Peacock	232.890,83 DM

Datensatz: 1 von 6

Bild 12.29: Datenblattansicht der Abfrage aus Bild 12.28

Das Abfrageergebnis enthält nur Preissummen von über 100.000 DM. Diese Summen selbst sind identisch mit den Preissummen, die oben im Abfrageergebnis in Bild 12.23 zu sehen sind. Dort jedoch sind weitere Preissummen mit jeweils weniger als 100.000 DM ausgewiesen. Daran können Sie erken-

nen, daß in diesem Falle zunächst die Preissummen auf Basis aller Datensätze berechnet wurden, und erst danach wurden die Gruppen aus dem Abfrageergebnis ausgeschlossen, deren Preissumme 100.000 DM nicht übersteigt.

Kriterien für ein Feld mit aggregierter Berechnung: Erst Datensätze einschränken, dann berechnen.

Ein gänzlich anderer Sachverhalt als im letzten Beispiel liegt vor, wenn Sie die Datensätze *vor* der Berechnung mit einem Kriterium einschränken und danach die aggregierten Ergebnisse ausweisen. Im folgenden Beispiel soll wiederum die Preissumme, gruppiert nach dem Nachnamen der Mitarbeiter, berechnet werden. Dabei sollen jedoch nur solche Datensätze mit Einzelbestellungen über 1.000 DM berücksichtigt werden. (Beachte: 1.000, nicht 100.000, weil keine Einzelbestellung einen Preis von über 100.000 aufweist.) Dies erreichen Sie dadurch, daß Sie das Feld *Endpreis* ein zweites Mal in den Entwurfsbereich aufnehmen, in der Zeile *Funktion* den Eintrag *Bedingung* vornehmen und als Kriterium >1000 angeben, vgl. Bild 12.30.

Feld:	Nachname	Endpreis	Endpreis
Tabelle:	Personal	Rechnungen	Rechnungen
Funktion:	Gruppierung	Summe	Bedingung
Sortierung:			
Anzeigen:	☑	☑	☐
Kriterien:			>1000
oder:			

Bild 12.30: Für das Feld »Endpreis« mit der Funktion »Bedingung« ist ein Kriterium angegeben. Daher werden Datensätze vor der Aggregierung entsprechend dem angegebenen Kriterium (hier: ⇒ 1000) ausgewählt, und danach wird aggregiert.

Wenn zu einem Feld mit der Funktion *Bedingung* ein Kriterium angegeben wird, wirkt dieses Kriterium beschränkend auf die Datensätze, *bevor* die Berechnung durchgeführt wird. Das Ergebnis dieser Abfrage ist in Bild 12.31 wiedergegeben. Es weist nicht nur teilweise andere Mitarbeiternamen aus als das Abfrageergebnis aus dem letzten Beispiel, sondern auch geringere Preissummen. Dies hat seinen Grund darin, daß alle Bestellungen mit einem Preis kleiner/gleich 1.000 DM aus der Summenberechnung ausgeschlossen wurden, so daß nur geringere Summen resultieren.

Eigene Ausdrücke für aggregierte Berechnungen verwenden

In der Art, wie Sie für andere Abfragen berechnete Felder erstellen, lassen sich auch für Abfragen mit aggregierten Berechnungen eigene Ausdrücke angeben. Solche selbst formulierten Ausdrücke unterliegen allerdings der Beschränkung, daß sie eine oder mehrere Aggregatfunktionen enthalten müssen. Die verfügbaren Aggregatfunktionen können Sie in dem Dropdown-Listenfeld der Zeile *Funktion* im Entwurfsbereich nachschlagen.

Abfrage1 : Auswahlabfrage

Nachname	Summe von Endpreis
Buchanan	31.528,90 DM
Callahan	53.180,50 DM
Davolio	100.558,24 DM
Dodsworth	47.030,16 DM
Fuller	97.154,02 DM
King	77.587,51 DM
Leverling	114.214,66 DM
Peacock	110.556,15 DM
Suyama	28.014,85 DM

Datensatz: 1 von 9

Bild 12.31: Datenblattansicht der Abfrage aus Bild 12.30

Beispiel. Weisen Sie für jeden Mitarbeiter den *Variationskoeffizienten* der Endpreise für alle ihm zuzuordnenden Bestellungen aus. Der Variationskoeffizient ist eine statistische Maßzahl für die relative Streuung von Werten um ihren Mittelwert, die sich als Quotient aus der sogen. Standardabweichung und dem Mittelwert ergibt. Der Variationskoeffizient ermöglicht in diesem Falle einen unmittelbaren Vergleich der Bestellpreise zwischen den Mitarbeitern. Access bietet die beiden Aggregatfunktionen *StdAbw* und *Mittelwert* an, aus denen der Variationskoeffizient ermittelt werden kann. Um die Aufgabe zu lösen, gehen Sie wie folgt vor:

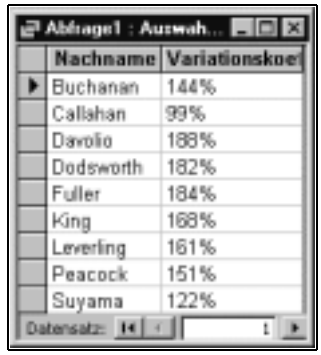
- Erstellen Sie eine Abfrage, welcher die Abfrage *Rechnungen* und die beiden Tabellen *Bestellungen* und *Personal* aus der Datenbank *Nordwind.mdb* hinzugefügt sind (vgl. die vorangehenden Beispiele).
- Nehmen Sie in den Entwurfsbereich das Feld *Nachname* aus der Tabelle *Personal* als Gruppierungsfeld auf.
- Wählen Sie aus dem Dropdown-Listenfeld der Zeile *Funktion* in der nächsten *freien* Spalte den Eintrag *Ausdruck* aus.
- Schreiben Sie in die Zeile *Feld* dieser Spalte die folgende Zeichenfolge:

  Variationskoeffizient: Format(StAbw([Endpreis])/Mittelwert([Endpreis]);"0%")

Der Teilausdruck *StAbw([Endpreis])/Mittelwert([Endpreis])* ermittelt den Variationskoeffizienten, und die Funktion *Format*() sorgt dafür, daß das Ergebnis als Prozentzahl ohne Dezimalstellen ausgegeben wird.

Der Abfrageentwurf sollte nach diesen Schritten so aussehen wie der aus Bild 12.32. Sein Abfrageergebnis gibt Bild 12.33 wieder.

Bild 12.32: Abfrageentwurf für eine aggregierende Berechnung auf Basis eines benutzerdefinierten Ausdrucks, hier zur Ermittlung des Variationskoeffizienten für den Endpreis



Nachname	Variationskoef
Buchanan	144%
Callahan	99%
Davolio	188%
Dodsworth	182%
Fuller	184%
King	168%
Leverling	161%
Peacock	151%
Suyama	122%

Bild 12.33: Datenblattansicht der Abfrage aus Bild 12.32

12.4 Kreuztabellenabfragen

In Bild 12.34 sehen Sie eine Kreuztabelle. Sie gibt die Anzahl der Bestellungen für jede Kombination aus *Bestimmungsland* und *Bestelljahr* wieder. In das Bestimmungsland Finnland beispielsweise gingen 1995 zwölf Bestellungen.

Die Kreuztabelle aus Bild 12.34 ist das Abfrageergebnis des Abfrageentwurfs, den Sie in Bild 12.35 betrachten können. Dieser Abfrageentwurf enthält drei Felder im Entwurfsbereich. Die Zeile *Kreuztabelle* gibt dort für jedes der drei Felder die Rolle an, die es in der Kreuztabelle spielt: Die Werte des Feldes *Bestimmungsland* geben die Zeilenüberschriften ab, die Werte des berechneten Feldes *Bestelljahr* die Spaltenüberschriften. Das Feld *Bestell-Nr* dient als Tabelleneintrag: Dieses Feld wird von der Funktion *Anzahl* ausgezählt, und die Ergebnisse erscheinen als Zellinhalte der Kreuztabelle.

Abfrage1 : Kreuztabellenabfrage

Bestimmungsland	1996	1997	1998
Argentinien		6	10
Belgien	2	7	10
Brasilien	13	42	28
Dänemark	3	11	4
Deutschland	24	64	33
Finnland	4	13	5
Frankreich	15	39	23
Großbritannien	10	30	16
Irland	5	10	4
Italien	3	15	10
Kanada	4	17	10
Mexiko	9	12	7
Norwegen	1	2	3
Österreich	8	21	11
Polen	1	2	4
Portugal	4	7	2
Schweden	6	17	14
Schweiz	3	8	7
Spanien	6	5	12
USA	23	60	39
Venezuela	8	20	18

Datensatz: 1 von 21

Bild 12.34: Kreuztabelle mit dem Feld »Bestimmungsland« als Zeilenüberschrift, dem berechneten Feld »Bestelljahr« als Spaltenüberschrift und der Anzahl von Bestellungen als Wert

Hinweis: Daß es sich bei dem Feld *Bestelljahr* um ein berechnetes handelt, ist für Kreuztabellen nicht notwendig, sondern dies erwies sich für das vorliegende Beispiel als zweckmäßig.

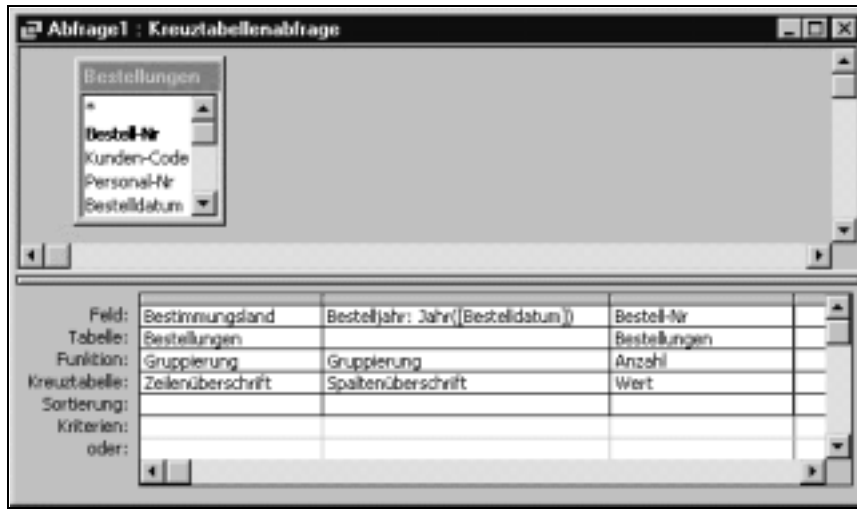


Bild 12.35: Abfrageentwurf für die Kreuztabelle in Bild 12.34

Eine Kreuztabellenabfrage stellt im Grunde genommen eine Fortentwicklung einer Abfrage mit aggregierten Berechnungen dar, denn dort wie hier wird nach Feldinhalten gruppiert, und es werden aggregierte Feldergebnisse ausgegeben. Eine Kreuztabellenabfrage muß die folgenden Merkmale aufweisen:

- Eine Kreuztabellenabfrage enthält mindestens zwei Gruppierungsfelder, von denen das eine die Zeilenüberschriften, das andere die Spaltenüberschriften bestimmt. Darüber hinaus darf die Kreuztabellenabfrage weitere Gruppierungsfelder für Zeilenüberschriften enthalten. Für die Spaltenüberschriften ist dagegen *nur ein* Gruppierungsfeld erlaubt.
- Eine Kreuztabellenabfrage enthält ein Feld, dessen Inhalte mit einer Funktion oder einem benutzerdefinierten Ausdruck aggregiert und als Werte in den Zellen der Kreuztabelle ausgegeben werden.

Im folgenden zeige ich, wie Sie eine Kreuztabellenabfrage von Grund auf selbständig erstellen. Ein alternativer Weg ist die Benutzung des Kreuztabellen-Assistenten. Diesen Weg können Sie selbst ohne weitere Anleitung gehen, weil Sie anschaulich durch mehrere erklärende Dialogfelder geleitet werden. Ich meine, daß Sie in jedem Falle Aufbau und Funktionsweise einer Kreuztabellenabfrage verstehen sollten, damit Sie komplexere Abfragen, die der Kreuztabellen-Assistent Ihnen ohnehin nie erstellen kann, selbständig aufbauen können.

Um den Abfrageentwurf aus Bild 12.35 zu erstellen, gehen Sie wie folgt vor:

- Öffnen Sie ggf. die Datenbank Nordwind.mdb, und erstellen Sie eine neue Abfrage.
- Fügen Sie der Abfrage die Tabelle *Bestellungen* hinzu, und schließen Sie das Dialogfeld *Tabelle anzeigen*.

- Schlagen Sie die Dropdown-Liste der Symbol-Schaltfläche *Abfragetyp* auf, und wählen Sie darin *Kreuztabellenabfrage* (vgl. das voranstehende Bild), oder wählen Sie den gleichnamigen Befehl aus dem Menü *Abfrage*. Daraufhin erscheinen im Entwurfsbereich auch die beiden Zeilen *Funktion* und *Kreuztabelle*.
- Ziehen Sie das Feld *Bestimmungsland* aus der Feldliste in den Entwurfsbereich, belassen Sie es bei der Funktion *Gruppierung*, und wählen Sie in der Zeile *Kreuztabelle* aus dem Dropdown-Listenfeld den Eintrag *Zeilenüberschrift*.
- Da die Spalten der Kreuztabelle von den Jahreszahlen der Bestelldatum-Werte gebildet werden sollen, brauchen wir als Gruppierungsfeld für die Spaltenüberschrift ein berechnetes Feld, denn das Bestelljahr wird in der Tabelle nicht unmittelbar ausgewiesen. Schreiben Sie in die Zeile *Feld* der ersten freien Spalte den Ausdruck

  Bestelljahr: Jahr([Bestelldatum])

Belassen Sie es auch hier bei der Funktion *Gruppierung*, und wählen Sie in der Zeile *Kreuztabelle* aus dem Dropdown-Listenfeld den Eintrag *Spaltenüberschrift*.

- Ziehen Sie das Feld *Bestell-Nr* aus der Feldliste in die dritte Spalte des Entwurfsbereichs, wählen Sie dafür aus dem Dropdown-Listenfeld der Zeile *Funktion* den Eintrag *Anzahl* und aus dem Dropdown-Listenfeld der Zeile *Kreuztabelle* den Eintrag *Wert*.

Nach diesen Schritten sollte Ihr Abfrageentwurf so aussehen, wie in Bild 12.35 dargestellt. Wenn Sie in die Datenblattansicht wechseln, müßte das Abfrageergebnis die Daten anzeigen, die Sie in Bild 12.34 sehen können.

Spaltenüberschriften fixieren

Anzahl und Reihenfolge der Spaltenüberschriften hängen vom jeweiligen Ergebnis einer Kreuztabellenabfrage ab, weil sie von der Anzahl verschiedener Feldinhalte und deren Art abhängen. Wenn Sie dieselbe Abfrage zu zwei verschiedenen Zeitpunkten ausführen, können sich daher die beiden resultierenden Kreuztabellen in der Spaltenanzahl und den Überschrifttexten unterscheiden. Manchmal mag es aber geboten sein, stets dieselbe Anzahl von Spalten mit den stets gleichen Überschriften anzufordern, um beispielsweise mehrere Kreuztabellen zum selben Sachverhalt im Laufe der Zeit besser vergleichen zu können.

Beispiel. Für das letzte Beispiel (vgl. Bild 12.34 und Bild 12.35) sollen folgende vier Spaltenüberschriften vorgesehen sein: 1995, 1996 und 1997. Um dies zu erreichen, gehen Sie folgendermaßen vor:

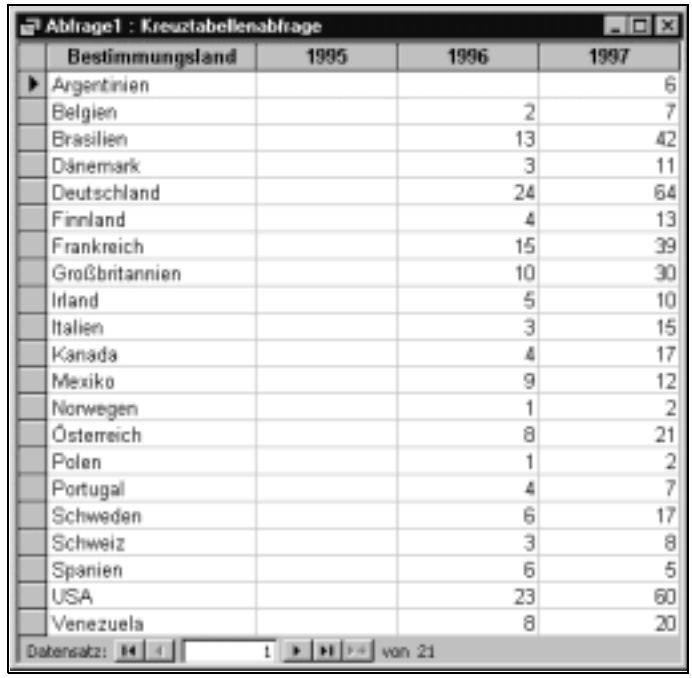
- Wechseln Sie ggf. in die Entwurfsansicht zurück.
- Klicken Sie auf die Symbol-Schaltfläche *Eigenschaften* in der Symbolleiste (vgl. links nebenstehend), oder wählen Sie den gleichnamigen Befehl aus dem Menü *Ansicht*. Klicken Sie ggf. auf einen Punkt, der nicht auf der Feldliste liegt, damit das Eigenschaftenfenster die Überschrift *Abfrageeigenschaften* trägt und sich somit auf die gesamte Abfrage bezieht. Das Eigenschaftenfenster einer Kreuztabellenabfrage bietet andere Eigenschaften an als dasjenige einer normalen Auswahlabfrage, vgl. Bild 12.36.
- Tragen Sie in das Eingabefeld *Fixierte Spaltenüberschriften* die Zeichenkette

  1995;1996;1997

  ein, vgl. Bild 12.36.

Wenn Sie nun in die Datenblattansicht wechseln, sehen Sie das Abfrageergebnis, das Bild 12.37 wiedergibt: Obwohl die zugrundeliegende Tabelle auch Bestellungen aus dem Jahr 1998 enthält, wird die diesem Jahr entsprechende Spalte nicht ausgewiesen, weil sie nicht in die Fixierung aufgenommen wurde. Umgekehrt wird eine Spalte mit der Überschrift 1995 angeführt, obgleich ihre Zellen keine Werte enthalten.

Bild 12.36: Als fixierte Spaltenüberschriften wurden die Werte 1995, 1996, und 1997 angegeben.



Abfrage1 : Kreuztabellenabfrage

Bestimmungsland	1995	1996	1997
Argentinien			6
Belgien		2	7
Brasilien		13	42
Dänemark		3	11
Deutschland		24	64
Finnland		4	13
Frankreich		15	39
Großbritannien		10	30
Irland		5	10
Italien		3	15
Kanada		4	17
Mexiko		9	12
Norwegen		1	2
Österreich		8	21
Polen		1	2
Portugal		4	7
Schweden		6	17
Schweiz		3	8
Spanien		6	5
USA		23	60
Venezuela		8	20

Datensatz: 1 von 21

Bild 12.37: Kreuztabellenabfrage mit fixierten Spaltenüberschriften

Bei der Angabe fixierter Spaltenüberschriften können Sie auch die Reihenfolge der Spaltenüberschriften beliebig wählen. Obwohl Sie formal jede Zeichenfolge als fixierte Spaltenüberschrift angeben können, sind im allgemeinen nur solche Texte sinnvoll, die prinzipiell Inhalt des als Spaltenüberschrift fungierenden Feldes sein *können*, weil Spalten, deren Überschrift nie als Wert im Feld vorkommen kann, stets leer bleiben müssen.

12.5 Abfragen mit SQL formulieren oder modifizieren

In diesem Punkt gehe ich nur sehr knapp auf die Bedeutung von SQL-Abfragen in Access ein, um das Wichtigste im Überblick mitzuteilen. Eine ausführliche Einführung in SQL-Abfragen bietet Kap. 14, *Einführung in SQL*.

SQL ist das Kürzel für Structured Query Language, der Standard-Abfragesprache in der Datenbankwelt. Access unterstützt diese Sprache in nahezu jeder Einzelheit: Sie können jede Abfrage, die Sie im Entwurfsbereich formuliert haben, als SQL-Anweisung nicht nur lesen, sondern auch bearbeiten. Eine bearbeitete SQL-Anweisung verändert umgekehrt die Formulierung im Entwurfsbereich. Beide Versionen – SQL und Entwurfsbereich – entsprechen sich daher stets und sind simultan verfügbar.

Es gibt einige Abfragearten, die sich in Access nicht im Entwurfsbereich erstellen lassen, sondern nur als SQL-Anweisungen. Es handelt sich um die folgenden Abfragearten:

- Union-Abfragen
- SQL-Pass-Through-Abfragen
- Unterabfragen
- Datendefinitionsabfragen

Das Erstellen einer Union-Abfrage wird im folgenden dargestellt, die anderen drei Abfragearten werden dagegen an anderen Stellen dieses Buches behandelt. Im nächsten Punkt wird jedoch zunächst kurz erklärt, wie der Entwurfsbereich und das SQL-Fenster interagieren und wie Sie prinzipiell eine Abfrage in der SQL-Ansicht bearbeiten oder erstellen können.

12.5.1 Entwurfsbereich und SQL

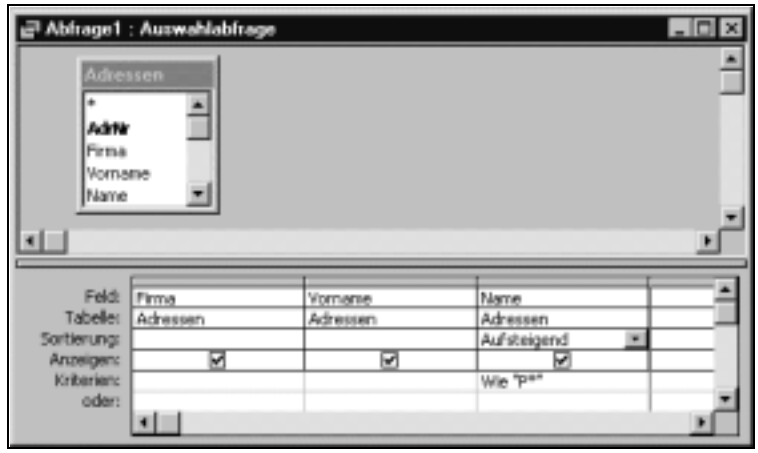


Bild 12.38: Auswahlabfrage in der Entwurfsansicht

In Bild 12.38 ist eine einfache Auswahlabfrage in der Entwurfsansicht wiedergegeben. Die Abfrage wählt aus der Tabelle *Adressen* die Felder *Firma*, *Vorname* und *Name* aus. Das Abfrageergebnis wird nach den Werten des Feldes *Name* in aufsteigender Richtung sortiert. Für dieses Feld ist als Kriterium der Ausdruck *Wie »P*«* angegeben.

Bild 12.39: SQL-Ansicht der Entwurfsansicht aus Bild 12.38

Der Formulierung in Bild 12.38 entspricht eine bestimmte SQL-Anweisung. Diese sehen Sie, wenn Sie in die SQL-Ansicht wechseln, vgl. Bild 12.39. Mit gewissen Kenntnissen der englischen Sprache finden Sie die einzelnen Elemente der Abfrageformulierung in der Entwurfsansicht in der SQL-Anweisung wieder:

- SELECT DISTINCTROW Adressen.Firma, Adressen.Vorname, Adressen.Name – entspricht dem Hinzufügen der Feldnamen aus der Feldliste.
- FROM Adressen – entspricht der Feldliste.
- WHERE ((Adressen.Name Like »P*«)) – entspricht dem Kriterienausdruck für das Feld *Name*.
- ORDER BY Adressen.Name – entspricht der Sortierung nach dem Feld *Name*.

SQL-Anweisung bearbeiten

Beispiel. Verändern Sie die in den letzten Absätzen besprochene SQL-Anweisung (vgl. Bild 12.39) so, daß statt nach dem Feld *Name* nach dem Feld *Firma* sortiert wird. Schauen Sie sich die Änderung danach in der Entwurfsansicht an.

Gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. in die SQL-Ansicht.
- Ersetzen Sie in der Zeichenkette *ORDER BY Adressen.Name* die Zeichenkette *Name* durch die Zeichenkette *Firma*.
- Wechseln Sie in die Entwurfsansicht und beobachten Sie, daß nunmehr die Sortierung *Aufsteigend* zum Feld *Firma* statt, wie vorher, zum Feld *Name* erscheint.

SQL-Anweisung kopieren

Sie können SQL-Anweisungen direkt für die Eigenschaften *Datenherkunft* bzw. *Datensatzherkunft* in Formularen und Berichten bzw. Listenfeldern darin angeben. Wenn Sie diese oder eine entsprechende Aufgabe lösen wol-

len, können Sie den Code für die SQL-Anweisung, der stets eine Zeichenkette ist, auf die folgende Weise erzeugen:

- Entwerfen Sie die Abfrage als Auswahlabfrage in der Entwurfsansicht.
- Wechseln Sie in die SQL-Ansicht.
- Kopieren Sie den SQL-Code in die Zwischenablage. Beachten Sie, daß Sie das Semikolon am Ende der Anweisung mit einbeziehen, weil es syntaktisch notwendig ist.
- Wechseln Sie zur Einfügestelle, und fügen Sie den SQL-Code aus der Zwischenablage ein.

12.5.2 SQL-SELECT-Anweisung in Formularen, Berichten und Steuerelementen

Für die Eigenschaft *Datenherkunft* eines Formulars oder Berichts und für die Eigenschaft *Datensatzherkunft* eines Listen- oder Kombinationsfeldes können Sie eine SQL-SELECT-Anweisung angeben. Den erforderlichen Code können Sie auf drei verschiedene Weisen eingeben:

- *Tippen*. Tippen Sie den erforderlichen Code als Text ein.
- *Kopieren*. Kopieren Sie die SQL-Anweisung aus dem Fenster der SQL-Ansicht einer bestehenden Abfrage, vgl. dazu den vorangehenden Unterpunkt *SQL-Anweisung kopieren*.

- *Abfrage-Generator.* Klicken Sie im Eigenschaftenfenster in das Bearbeitungsfeld der Eigenschaft *Datenherkunft* bzw. *Datensatzherkunft*, und rufen Sie den *Abfrage-Generator* auf, indem Sie auf die am rechten Rand des Bearbeitungsfeldes eingeblendete Symbol-Schaltfläche mit den drei Pünktchen (vgl. links nebenstehend) klicken. Der *Abfrage-Generator* läßt Sie eine Abfrage in der Entwurfsansicht erstellen. Wenn Sie den Generator beenden, wird der Abfrageentwurf als SQL-Code in das Bearbeitungsfeld der Eigenschaft eingefügt. Auf entsprechende Weise können Sie auch eine bestehende SQL-Anweisung für eine der beiden genannten Eigenschaften bearbeiten.

12.5.3 Union-Abfragen

Wenn Sie in einer Abfrage zwei Tabellen verknüpfen, enthält das Abfrageergebnis so viele Datensätze, wie übereinstimmende Fälle in den Verknüpfungsfeldern existieren. Bei einer verknüpften Abfrage findet eine Vereinigung von Feldern aus den beiden Tabellen sozusagen in der Waagerechten statt: Das Abfrageergebnis enthält je Datensatz Felder aus beiden Tabellen.

Bei einer Union-Abfrage werden Datensätze dagegen in der Senkrechten zusammengeführt.

Beispiel. Bild 12.40 und Bild 12.41 geben für die Tabelle *Kunden* bzw. *Lieferanten* aus der Datenbank Nordwind.mdb jeweils die Firmen (Kunden bzw. Lieferanten) wieder, die aus Italien stammen. Das erste Abfrageergebnis enthält drei, das zweite zwei Datensätze. Insgesamt stammen also fünf Firmen aus Italien. In einer Union-Abfrage können Sie diese fünf Firmen in einem Abfrageergebnis ausgeben lassen.

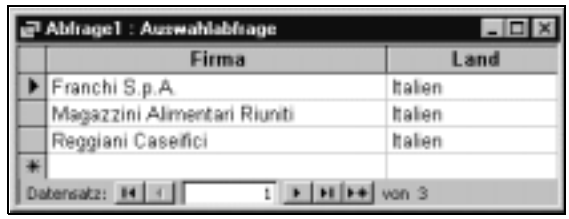


Bild 12.40: Diese Datensätze stammen allein aus der Tabelle »Kunden« und wurden durch die folgende SQL-Anweisung ausgewählt: SELECT Firma, Land FROM Kunden WHERE Land="Italien«;.

Bild 12.41: Diese Datensätze stammen allein aus der Tabelle Lieferanten und wurden durch die folgende SQL-Anweisung ausgewählt: SELECT Firma, Land FROM Lieferanten WHERE Land="Italien«;.

Um die Kunden- und Lieferantenfirmen von einer Union-Abfrage insgesamt auswählen zu lassen, müssen Sie eine SQL-Anweisung in das Fenster der SQL-Ansicht einer Abfrage eingeben. Verfahren Sie im einzelnen wie folgt:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*.
- Erstellen Sie eine neue Abfrage, ohne dieser eine Tabelle oder Abfrage hinzuzufügen.
- Wählen Sie aus dem Menü *Abfrage* den Befehl *SQL-spezifisch* und dazu den Unterbefehl *Union*. Dann zeigt sich ein Abfragefenster mit der Überschrift *Union-Abfrage: Abfrage 1* (oder 2 etc.), in das Sie SQL-Anweisungen für eine Union-Abfrage eingeben können.
- Geben Sie die folgende Anweisung ein, vgl. auch Bild 12.42.

```
SELECT Firma, Land
FROM Kunden
WHERE Land="Italien"
```

```
UNION SELECT Firma, Land
FROM Lieferanten
WHERE Land="Italien";
```

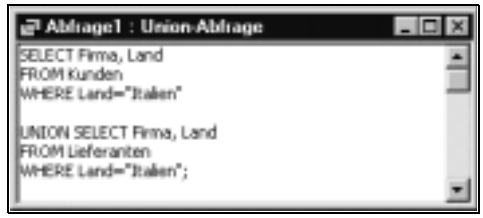


Bild 12.42: SQL-Anweisung für die Union-Abfrage, um die Firmen aus Italien aus den beiden Tabellen »Kunden« und »Lieferanten« in einem Abfrageergebnis senkrecht zusammenzuführen.

Wechseln Sie in die Datenblattansicht. *Hinweis*: Die Entwurfsansicht steht in dieser Situation nicht zur Verfügung, denn Access erkennt, daß diese SQL-Anweisung nicht im Entwurfsbereich widergespiegelt werden kann. Die Datenblattansicht der Union-Abfrage ist in Bild 12.43 zu sehen. Das Abfrageergebnis enthält alle fünf Datensätze, die dem Kriterium *Italien* insgesamt entsprechen: Drei Firmen stammen aus der Tabelle *Kunden*, zwei aus der Tabelle *Lieferanten*.

Bild 12.43: Abfrageergebnis der Union-Abfrage, deren SQL-Anweisung u.a. in Bild 12.42 zu sehen ist

Beachten Sie allgemein:

Eine Union-Abfrage muß als SQL-Anweisung formuliert werden. Sie enthält zwei oder mehr SELECT-Anweisungen. Jede SELECT-Anweisung muß dieselbe Anzahl von Feldern in derselben Reihenfolge liefern. Die korrespondierenden Felder müssen kompatible Datentypen haben. Allerdings darf, als Ausnahme, ein Feld vom Datentyp *Zahl* mit einem vom Datentyp *Text* korrespondieren. Im vorangehenden Beispiel sind die beiden Felder *Firma* bzw. *Land* jeweils korrespondierende Felder.

Weitere Informationen zu Union-Abfragen finden Sie in der Online-Hilfe zum gleichnamigen Stichwort.

Kapitel 13

Aktionsabfragen

Anders als Auswahlabfragen, die ein Abfrageergebnis aus bestehenden Tabellen erzeugen, an den zugrundeliegenden Daten jedoch nichts verändern (es sei denn, Sie bearbeiten ein Abfrageergebnis manuell), ist es wesentlich für Aktionsabfragen, daß sie Daten in Tabellen in der einen oder anderen Weise verändern. In Access können Sie über die folgenden Arten von Aktionsabfragen verfügen:

- **Tabellenerstellungsabfrage**: Kopiert die Daten des Abfrageergebnisses in eine neue Tabelle
- **Löschabfrage**: Löscht Datensätze aus Tabellen
- **Anfügeabfrage**: Fügt einer bestehenden Tabelle die Datensätze des Abfrageergebnisses hinzu
- **Aktualisierungsabfrage**: Ändert Daten in bestehenden Datensätzen von Tabellen

Diese Abfragearten werden in den folgenden Punkten dieses Kapitels im einzelnen behandelt. Wenn Sie eine Aktionsabfrage gespeichert haben, wird ihrem Symbol im Datenbankfenster ein Ausrufezeichen beigefügt.

13.1 Abfragen, die neue Tabellen erzeugen

Beispiel. Nehmen Sie als Ausgangspunkt die Tabellen *Personal* und *Projekte* aus der Datenbank *Projekte.mdb* von der Begleit-CD-ROM (oder die entsprechende von Ihnen nach den Ausführungen von Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, geschaffene). Erstellen Sie mittels einer Tabellenerstellungsabfrage eine neue Tabelle mit dem Namen *Test*, welche alle Datensätze mit allen relevanten Informationen für nicht stornierte Projekte enthält. Sie erledigen diese Aufgabe am besten wie folgt:

- Öffnen Sie die Datenbank *Projekte.mdb*.
- Erstellen Sie eine neue Abfrage, und fügen Sie dieser die Tabellen *Personal* sowie *Projekte* hinzu.
- Ziehen Sie das Sternchen-Feld der Feldliste *Personal* in den Entwurfsbereich.

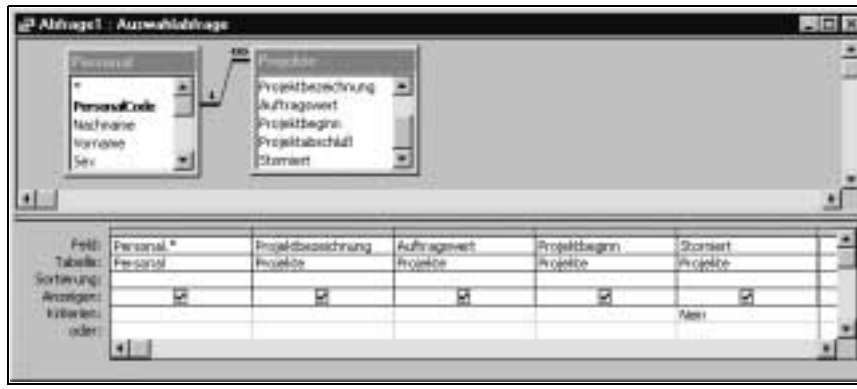


Bild 13.1: Entwurf der Aktionsabfrage zum Erstellen einer neuen Tabelle mit Datensätzen für nicht stornierte Projekte

- Ziehen Sie die Felder *Projektbezeichnung*, *Auftragswert*, *Projektbeginn* sowie *Storniert* aus der Feldliste *Projekte* in den Entwurfsbereich.
- Geben Sie für das Feld *Storniert* als Kriterium *Nein* an. Ihr Abfrageentwurf sollte jetzt so aussehen wie der in Bild 13.1 wiedergegebene.

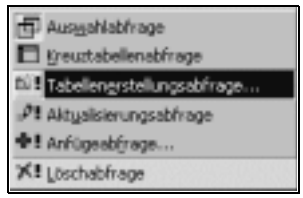


Bild 13.2: Aufgeschlagene Dropdown-Liste der Symbol-Schaltfläche »Abfragetyp«

- Schlagen Sie die Dropdown-Liste der Symbol-Schaltfläche *Abfragetyp* in der Symbolleiste auf, und wählen Sie in der Liste (vgl. Bild 13.2) *Tabellenerstellungsabfrage*, oder wählen Sie den gleichnamigen Befehl aus dem Menü *Abfrage*. Access blendet das Dialogfeld *Neue Tabelle erstellen* ein, vgl. Bild 13.3.
- Tragen Sie den Namen *Test* (oder einen anderen) in das Eingabefeld *Tabellenname* des Dialogfeldes ein, und bestätigen Sie mit OK. Die Titelleiste des Abfragefensters trägt jetzt die Beschriftung *Abfrage 1* (oder 2 etc.): *Tabellenerstellungsabfrage*.

Bild 13.3: Dialogfeld »Neue Tabelle erstellen« zur Angabe des Namens für die durch die Aktionsabfrage neu zu erzeugende Tabelle

- Klicken Sie auf die Symbol-Schaltfläche *Ausführen* (vgl. links nebenstehend), oder wählen Sie den gleichnamigen Befehl aus dem Menü *Abfrage*. Access zeigt in einem Meldungsfeld an, wie viele Datensätze in die neue Tabelle kopiert werden, vgl. Bild 13.4. Bestätigen Sie mit *Ja* (oder brechen Sie die Aktion mit *Nein* ab).

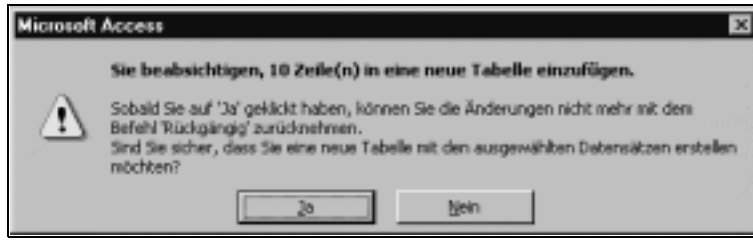


Bild 13.4: Meldung beim Ausführen der Tabellenerstellungsabfrage. Mit »Ja« wird die neue Tabelle erstellt, und 10 Datensätze werden hinein kopiert. Mit »Nein« wird die Aktion abgebrochen.

Eine Aktionsabfrage zum Erstellen einer neuen Tabelle läßt sich jederzeit ohne Informationsverlust wieder in eine normale Auswahlabfrage zurückverwandeln. Dazu müßten Sie aus dem Menü *Abfrage* den Befehl *Auswahlabfrage* oder aus der Dropdown-Liste der Symbol-Schaltfläche *Abfragetyp* den Eintrag *Auswahlabfrage* wählen, während die betreffende Abfrage in der Entwurfsansicht geöffnet ist.

Wenn Sie eine Tabellenerstellungsabfrage speichern, wird ihr Symbol in der Objektliste *Abfragen* des Datenbankfensters durch ein nachgestelltes Ausrufezeichen gekennzeichnet, vgl. Bild 13.5, in dem das Symbol für die Tabellenerstellungsabfrage mit dem Namen *AbfrTest* zu sehen ist.

Bild 13.5: Objektliste »Abfragen« des Datenbankfensters: Das Symbol für die Tabellenerstellungsabfrage »AbfrTest« unterscheidet sich deutlich vom Symbol der normalen Auswahlabfrage »Personal und Projekte« und »Projekte und Personal«.

13.2 Löschabfragen

Beispiel. Die im vorangehenden Punkt 13.1, *Abfragen, die neue Tabellen erzeugen*, erstellte Tabelle *Test* enthält nur Datensätze mit dem Eintrag *Nein* (bzw. 0) im Feld *Storniert*. Ändern Sie zunächst für einige dieser Datensätze den Eintrag zu *Storniert* in *Ja* (bzw. -1) um. Löschen Sie diese Datensätze dann mit einer entsprechenden Löschabfrage. Gehen Sie im einzelnen folgendermaßen vor:

- Nehmen Sie die genannten Änderungen an der Tabelle *Test* vor.
- Erstellen Sie eine neue Abfrage, und fügen Sie dieser die Tabelle *Test* hinzu.
- Ziehen Sie das Feld *Storniert* aus der Feldliste *Test* in den Entwurfsbereich, und geben Sie dafür als Kriterium *Ja* an.
- Ziehen Sie das Sternchen * aus der Feldliste *Test* in den Entwurfsbereich.
- Wählen Sie aus der Dropdown-Liste der Symbol-Schaltfläche *Abfragetyp* den Eintrag *Löschabfrage*, oder wählen Sie den gleichnamigen Befehl aus dem Menü *Abfrage*. Dann sollte der Abfrageentwurf aussehen wie in Bild 13.6 wiedergegeben: Der Entwurfsbereich weist jetzt auch die Zeile *Löschen* auf. In den Zellen dieser Zeile müssen die beiden Einträge *Bedingung* und *Von* stehen: *Von* gilt für die Tabelle, in der die entsprechenden Datensätze gelöscht werden sollen, und *Bedingung* bezieht sich auf das Feld, welches das Kriterium angibt.

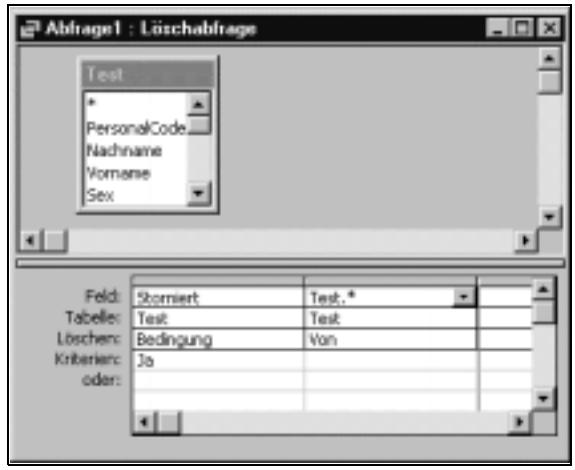


Bild 13.6: Abfrageentwurf nach Umwandlung in eine Löschabfrage. Beachten Sie die Beschriftung der Titelleiste des Abfragefensters und die Zeile »Löschen« im Entwurfsbereich.

- Klicken Sie auf die Symbol-Schaltfläche *Ausführen* (vgl. links nebenstehend), oder wählen Sie den Befehl *Ausführen* aus dem Menü *Abfrage*. Access meldet, wie viele Datensätze in der Tabelle gelöscht werden, vgl. Bild 13.7. Bestätigen Sie mit *Ja* (oder brechen Sie die Aktion mit *Nein* ab).

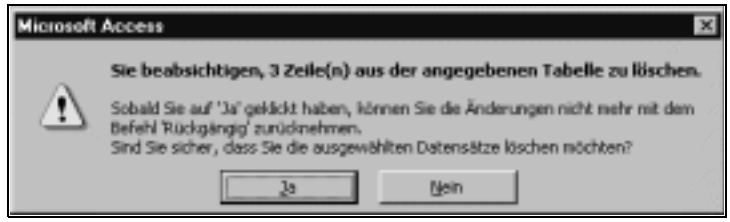


Bild 13.7: Meldung beim Ausführen der Löschabfrage. Mit »Ja« werden die fünf Datensätze gelöscht. Mit »Nein« wird die Aktion abgebrochen.

Eine Löschabfrage kann auch für mehr als eine Tabelle konzipiert werden, ganz so, wie eine Auswahlabfrage. Sie müssen dann jede Tabelle, in der Datensätze gelöscht werden sollen, dadurch angeben, daß Sie ihr Sternchen * in den Entwurfsbereich ziehen. Access erkennt dies und nimmt automatisch den Eintrag *Von* für die Zeile *Löschen* vor.

Wenn Sie Datensätze aus Tabellen löschen wollen, zwischen denen eine 1:n-Beziehung mit referentieller Integrität besteht, können Sie Datensätze der 1-Seite nur löschen, wenn alle mit ihnen verknüpften Datensätze auf der n-Seite gelöscht wurden oder wenn Sie die Eigenschaft *Löschweitergabe an Detaildatensatz* angekreuzt haben.

Da Sie mit einer Löschabfrage möglicherweise weitreichende Änderungen an den Datensätzen Ihrer Tabelle(n) bewirken, die sofort gespeichert werden, empfiehlt es sich im allgemeinen, vor dem Ausführen der Löschabfrage eine Auswahlabfrage mit denselben Feldern und denselben Kriterien zu erstellen. Wenn Sie sich das Ergebnis einer solchen Abfrage anschauen, können Sie überprüfen, ob die Datensätze selektiert wurden, die Sie entsprechend den angegebenen Kriterien erwarten. Erst nach erfolgreicher Prüfung sollten Sie die Auswahlabfrage in die entsprechende Löschabfrage umwandeln und diese ausführen. Beachten Sie dann jedoch, daß die für eine Auswahlabfrage ggf. gewählte Option *Keine Duplikate* nach dem Umwandeln einer Auswahl- in eine Löschabfrage ihre Wirkung verliert!

13.3 Anfügeabfragen

Beispiel. Der Tabelle *Test* in der Datenbank *Projekte.mdb*, die oben als Demonstrationsbeispiel (vgl. Punkt 13.1, *Abfragen, die neue Tabellen erzeugen*) erzeugt wurde, sollen mit einer Anfügeabfrage Datensätze hinzugefügt werden. Die Abfrage soll alle Felder der Tabelle *Personal* sowie die Felder *Projektbezeichnung*, *Auftragswert*, *Projektbeginn* und *Storniert* der Tabelle *Projekte* wiedergeben. Gehen Sie im einzelnen wie folgt vor:

- Öffnen Sie ggf. die Datenbank *Projekte.mdb*, und erstellen Sie ggf. die Tabelle *Test*, wie dies oben im Punkt 13.1, *Abfragen, die neue Tabellen erzeugen*, beschrieben ist.
- Erstellen Sie eine neue Abfrage, und fügen Sie dieser die Tabellen *Personal* sowie *Projekte* hinzu.
- Ziehen Sie das Sternchen-Feld der Feldliste *Personal* in den Entwurfsbereich.
- Ziehen Sie die Felder *Projektbezeichnung*, *Auftragswert*, *Projektbeginn* sowie *Storniert* in den Entwurfsbereich.
- Wählen Sie aus der Dropdown-Liste der Symbol-Schaltfläche *Abfragetyp* den Eintrag *Anfügeabfrage*, oder wählen Sie den gleichnamigen Befehl aus dem Menü *Abfrage*. Access blendet das Dialogfeld *Anfügen* ein, vgl. Bild 13.8.

Bild 13.8: Dialogfeld »Anfügen«

- Geben Sie den Namen *Test* in das Eingabefeld *Tabellenname* des Dialogfeldes ein, und bestätigen Sie mit OK. Als Ergebnis sollte Ihr Abfrageentwurf wie in Bild 13.9 wiedergegeben aussehen.

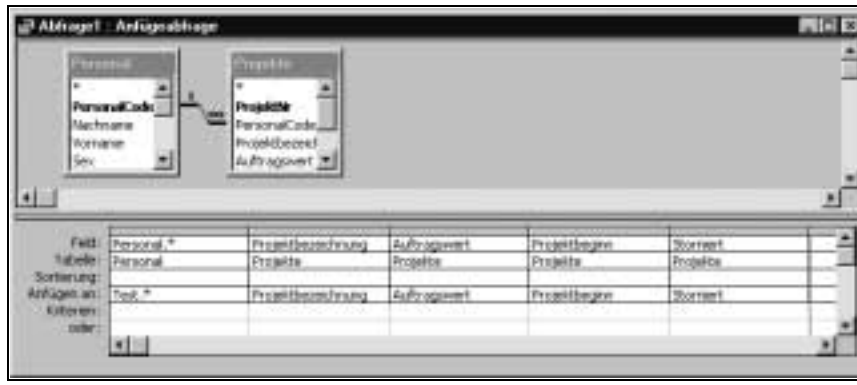


Bild 13.9: Abfrageentwurf nach Umwandlung in eine Anfügeabfrage. Beachten Sie die Zeile »Anfügen an«, die von Access zusammen mit den Einträgen nach Umwandlung in die Anfügeabfrage eingefügt wurde.

- Klicken Sie auf die Symbol-Schaltfläche *Ausführen* (vgl. links nebenstehend), oder wählen Sie den Befehl *Ausführen* aus dem Menü *Abfrage*. Access meldet, wie viele Datensätze an die Tabelle angefügt werden, vgl. Bild 13.10. Bestätigen Sie mit *Ja*, oder brechen Sie mit *Nein* ab.

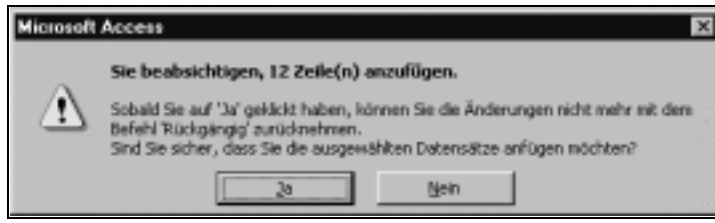


Bild 13.10: Meldung beim Ausführen der Anfügeabfrage. Mit »Ja« werden die 12 Datensätze angefügt. Mit »Nein« wird die Aktion abgebrochen.

Nach der Umwandlung in eine Anfügeabfrage erscheint im Entwurfsbereich die Zeile *Anfügen an*. Im vorliegenden Beispiel hat Access automatisch dafür gesorgt, daß ihre Zellen die richtigen Einträge aufweisen, weil die Zieltabelle *Test*, der die Datensätze angefügt werden sollen, namensgleiche Felder enthält wie die in der Zeile *Feld* im Entwurfsbereich angeführten Quellfelder. Die Übereinstimmung der Feldnamen in der Abfrage einerseits und in der Zieltabelle andererseits ist jedoch keine Voraussetzung für das Ausführen einer Anfügeabfrage, denn Sie können jedem Feld der Zeile *Feld* im Entwurfsbereich gezielt ein Feld der Zieltabelle zuordnen, indem Sie in

der entsprechenden Zelle der Zeile *Anfügen an* das Dropdown-Listenfeld aufschlagen, welches die Namen der Zieltabelle auflistet, und aus diesem das gewünschte Feld wählen. Wenn in der Zeile *Feld* das Sternchen * angegeben ist, muß in der entsprechenden Zelle der Zeile *Anfügen an* ebenfalls das entsprechende Sternchen für die Zieltabelle stehen, wie dies für den Abfrageentwurf in Bild 13.9 ja auch zutrifft.

13.4 Aktualisierungsabfragen

Mit einer Aktualisierungsabfrage können Sie Felder neu berechnen und in den zugrundeliegenden Tabellen entsprechend aktualisieren lassen.

Beispiel. In der Datenbank *Nordwind.mdb* existieren die beiden Tabellen *Lieferanten* sowie *Artikel*, die durch eine 1:n-Beziehung miteinander verknüpft sind. Der Lieferant *Plutzer Lebensmittelgroßmärkte AG* hat seine Preise um 10% erhöht. Ändern Sie die Preise aller Artikel dieses Lieferanten entsprechend mit Hilfe einer Aktualisierungsabfrage. Gehen Sie dazu folgendermaßen vor:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*.
- Erstellen Sie eine neue Abfrage, und fügen Sie die Tabellen *Lieferanten* sowie *Artikel* hinzu.
- Ziehen Sie das Feld *Firma* aus der Feldliste *Lieferanten* und das Feld *Einzelpreis* aus der Feldliste *Artikel* in den Entwurfsbereich.
- Wählen Sie aus der Dropdown-Liste der Symbol-Schaltfläche *Abfragetyp* den Eintrag *Aktualisierungsabfrage*, oder wählen Sie den gleichnamigen Befehl aus dem Menü *Abfrage*. Daraufhin fügt Access dem Entwurfsbereich die Zeile *Aktualisieren* hinzu, vgl. Bild 13.11.
- Geben Sie in die Zeile *Aktualisieren* für das Feld *Einzelpreis* den Ausdruck

  [Einzelpreis]*1,1

  und in die Zeile *Kriterien* für das Feld *Firma* die Zeichenkette

  " Plutzer Lebensmittelgroßmärkte AG "

  ein. Ihr Abfrageentwurf sollte jetzt aussehen wie derjenige in Bild 13.11.

- Klicken Sie auf die Symbol-Schaltfläche *Ausführen* (vgl. links nebenstehend), oder wählen Sie den Befehl *Ausführen* aus dem Menü *Abfrage*. Access meldet, wie viele Datensätze aktualisiert werden, vgl. Bild 13.12. Bestätigen Sie mit Ja, oder brechen Sie mit Nein ab.

Hinweis. Falls Sie die Daten der Datenbank *Nordwind.mdb* nach der letzten Aktualisierungsabfrage wieder in den ursprünglichen Zustand bringen wollen, können Sie die letzte Aktualisierungsabfrage erneut durchführen. Vorher müssen Sie jedoch den bisherigen Ausdruck

[Einzelpreis]*1,1

durch den Ausdruck

[Einzelpreis]/1,1

ersetzen.

Da Sie mit einer Aktualisierungsabfrage möglicherweise weitreichende Änderungen an den Daten Ihrer Tabelle(n) bewirken, die sofort gespeichert werden, empfiehlt es sich im allgemeinen, vor dem Ausführen der Aktualisierungsabfrage eine Auswahlabfrage mit denselben Feldern und denselben Kriterien und zusätzlich einem oder mehreren berechneten Feldern zu erstellen. Die berechneten Felder sollten dabei genau die Ausdrücke enthalten, mit denen Sie die Aktualisierung an den Feldern durchführen wollen. Wenn Sie sich das Ergebnis einer solchen Abfrage anschauen, können Sie überprüfen, ob die Datensätze selektiert wurden, die Sie entsprechend den angegebenen Kriterien erwarten, und ebenfalls, ob die Berechnungsergebnisse Ihren Vorstellungen entsprechen. Erst wenn dies zutrifft, sollten Sie die Auswahlabfrage in eine Aktualisierungsabfrage umwandeln und die geprüften Ausdrücke der berechneten Felder in die Zeile *Aktualisieren* der entsprechenden zu aktualisierenden Felder übertragen.

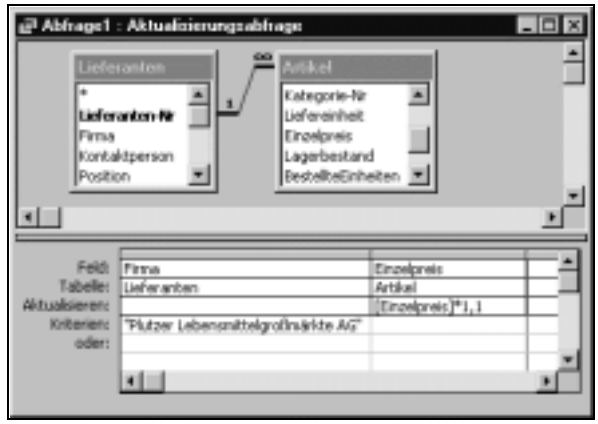


Bild 13.11: Abfrageentwurf nach Umwandlung in eine Aktualisierungsabfrage

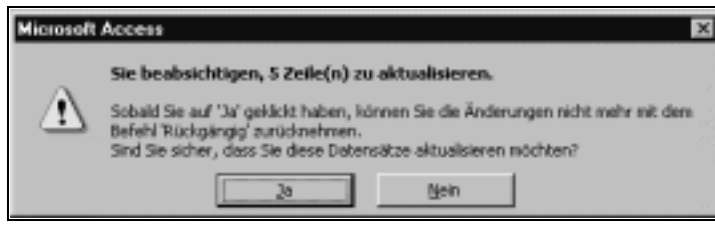


Bild 13.12: Meldung beim Ausführen der Aktualisierungsabfrage. Mit »Ja« werden die fünf Datensätze aktualisiert. Mit »Nein« wird die Aktion abgebrochen.

Kapitel 14

Einführung in SQL

14.1 Was ist SQL?

SQL ist die Abkürzung für *Structured Query Language*. Diese zunächst Anfang der siebziger Jahre von der Firma IBM im Rahmen eines Forschungsprojekts entwickelte Abfragesprache für relationale Datenbanksysteme hatte ursprünglich den Namen *Structured English Query Language* und wurde entsprechend als SEQUEL abgekürzt. Später hat sich dann daraus das Kürzel SQL ergeben. Im englischen und z.T. auch im deutschen Sprachraum ist davon noch übrig geblieben, daß SQL häufig nicht als *EsKuEl*, sondern als *Siequel* ausgesprochen wird. Inhaltlich bedeutsamer ist die Tatsache, daß sich SQL von seiner proprietären IBM-Herkunft zu einem internationalen Standard entwickelt hat: Seit 1982 arbeitet das American National Standards Institute – im allgemeinen besser als ANSI bekannt und ungefähr dem deutschen DIN vergleichbar – an einer Standardisierung von SQL. Der gegenwärtige ANSI-SQL-Standard wird als SQL2 oder SQL-92 bezeichnet.

In der Regel realisieren kommerzielle RDBMS-Produkte nicht alle Leistungsspezifikationen des ANSI-Standards. Den Kernumfang haben jedoch alle bedeutsamen Anbieter relationaler Datenbanksysteme in den SQL-Sprachumfang ihrer Software aufgenommen. Dies gilt auch für Microsoft Access: Alle auch nur einigermaßen wichtigen Leistungsmerkmale – repräsentiert durch die entsprechenden Sprachelemente – sind in Access verfügbar. Darüber hinaus existieren in Access einige SQL-Sprachelemente, die im ANSI-Standard nicht definiert sind. Wenn Sie sich über Einzelheiten des Vergleichs von Access- und ANSI-SQL informieren wollen, schauen Sie in der Online-Hilfe nach unter dem Thema *Vergleich zwischen der SQL-Sprache von Microsoft Jet und ANSI SQL*. Was für Microsoft Access gilt, bieten auch alle verbreiteten anderen RDBMS: Der praktisch bedeutsame Kern des ANSI-SQL ist realisiert. Dies hat die wichtige Konsequenz, daß Datenbanksysteme, zu denen eine ODBC-Verbindung eingerichtet ist, dieselbe Sprache sprechen wie Access, woraus sich beispielsweise die Möglichkeit ergibt,

einem (in der Regel leistungsfähigeren) SQL-Server eine SQL-Abfrage zu übergeben, diese dort auswerten zu lassen und das Abfrageergebnis zurückzubekommen. Diese Form der Abfragen wird in Access als Pass-Through-Abfragen bezeichnet. Darüber hinaus ist es seit Access 2000 möglich, in einer Access-Anwendung – die dann als Access-Projekt bezeichnet wird – statt der *JetEngine* den *Microsoft SQL Server* als Datenbank-Maschine einzusetzen, vgl. dazu im einzelnen Kap. 37, *Access-Projekte als Client/Server-Datenbank: Grundlagen*.

SQL kann als eine Sprache sehr hoher Ordnung bezeichnet werden. Dies bedeutet, daß Sie durch Angabe weniger Schlüsselwörter und Tabellennamen oder anderer benutzerseitiger Informationen eine differenzierte Abfrage formulieren können. Zumindest einfache Abfragen, die nur auf einer Tabelle basieren, jedoch bereits Kriterien und eine Selektion einzelner Tabellenfelder umfassen, sind auch von bisher mit SQL wenig vertrauten Benutzern leicht zu erstellen – auch ohne die Entwurfsansicht einer Abfrage zu Hilfe zu nehmen. Wenn Sie jedoch auch diese Hilfe in Anspruch nehmen, können Sie leicht auch komplexere Abfragen formulieren. An genau dieser Arbeitsweise habe ich mich beim Schreiben dieses Kapitels orientiert: Ich empfehle Ihnen, eine Abfrage zunächst so weit wie möglich in der Entwurfsansicht einer Abfrage zu entwerfen und sich dann den SQL-Code in der SQL-Ansicht zu betrachten. Diesen können Sie dann bearbeiten. Dazu sind allerdings gewisse minimale SQL-Kenntnisse erforderlich, die in diesem Kapitel vermittelt werden sollen. Ich gehe also nicht davon aus, daß Sie – zumindest komplexere – Abfragen Wort für Wort selbständig formulieren, sondern in der Regel bestehenden SQL-Code bearbeiten. Dies gilt insbesondere für Mehrtabellenabfragen mit einer oder mehreren Verknüpfungen (*Joins*).

Da SQL-Anweisungen stets als Zeichenketten (als Text) formuliert und übergeben werden, können Sie diese mit jedem Text-Generator bearbeiten. Darüber hinaus sind auch alle Formen von Verkettungen von Zeichenfolgen möglich. Dies erlaubt vor allem die Manipulation einer SQL-Anweisung in VBA-Prozeduren.

Konventionen zur Syntaxdarstellung

Die Syntaxdarstellung in diesem Kapitel folgt den üblichen Konventionen für die SQL-Darstellung, d.h. der *Erweiterten Backus Naur Form* (*EBNF*). Diese verwendet die folgenden Symbole in der angegebenen Bedeutung:

Notation	Beispiel
Reservierte Wörter werden in Großbuchstaben angegeben	SELECT, FROM, WHERE
Optionale Angaben stehen zwischen eckigen Klammern: []. Im Beispiel kann das reservierte Wort WHERE fortgelassen werden.	SELECT FROM ... [WHERE ...]

Tabelle 14.1: Syntaxnotation für SQL

Notation	Beispiel
Alternative Angaben werden durch einen senkrechten Strich getrennt: |. Im Beispiel darf vor der Zahl ein Plus- oder ein Minuszeichen angegeben werden, jedoch nicht beide.	[+ | -] Zahl
Wiederholungen gleichartiger Angaben werden zwischen geschweiften Klammern angegeben: {}. Wenn dabei ein Komma spezifiziert ist, muß dieses mit angegeben werden. Im Beispiel muß zum reservierten Wort FROM mindestens eine Tabelle, es kann auch eine zweite angegeben werden, dann jedoch durch ein Komma getrennt.	FROM Tabelle {, Tabelle}
Auslassungen werden durch Punkte gekennzeichnet: ... Im Beispiel erfordert die SQL-Anweisung die Angabe einer Tabelle und weitere, nicht näher spezifizierte Informationen.	SELECT Tabelle ...

Tabelle 14.1: Syntaxnotation für SQL

14.2 Vorteile der Verwendung von SQL in Access

Access stellt mit der Entwurfsansicht von Abfragen einen *Abfrage-Generator* zur Verfügung, der es erlaubt, auf einfache Weise auch sehr umfangreiche Abfragen zu formulieren. Im Vergleich dazu sind SQL-Anweisungen – trotzt der relativ einfachen Sprache – kompliziert. Daher bedarf es schon besonderer Begründung für die Verwendung von SQL-Anweisungen in Access. Die Vorteile der Verwendung von SQL sind die folgenden:

- *SQL als Datenquelle in Formularen, Berichten und Steuerelementen*. Formulare, Berichte, Listen- bzw. Kombinationsfelder (einschließlich Nachschlagefelder in Tabellen) sowie Tabellenfelder akzeptieren SQL-Anweisungen als Angaben für die Eigenschaft *Datenherkunft* bzw. *Datensatzherkunft*. Durch Angabe einer Abfrage statt einer Tabelle als Datenquelle verfügen Sie über viel mehr Möglichkeiten, geeignete Datensätze zusammenzustellen. Insbesondere in Listen- bzw. Kombinationsfeldern können Sie mit Abfragen genau die Felder und Reihenfolge der Felder bestimmen, die Sie für die angezeigte Liste benötigen. Darüber hinaus erlauben Abfragen für alle Datenquellen die Selektion von Datensätzen auf Basis von Kriterien. Die Abfragen könnten zwar als feste Abfrage-Objekte mit Namen gespeichert werden, jedoch erscheint dies nicht immer sehr praktisch: Bedenken Sie, daß Sie dann vermutlich sehr viele Abfragen gespeichert haben müßten, während der entsprechende SQL-Code genau an der Stelle zusammen mit dem Objekt (Formular, Listenfeld usw.) gespeichert und entsprechend zu finden ist, an der er gebraucht wird.

- *SQL-spezifische Abfragen*. Einige Abfragetypen lassen sich nur mit SQL-Code formulieren und nicht in der Abfrage-Entwurfsansicht. Dies betrifft *SQL Pass-Through-Abfragen*, *Datendefinitionsabfragen*, *Union-Abfragen* sowie *Unterabfragen*.
- *Abfragen in VBA-Prozeduren*. Beim Arbeiten mit Datenzugriffsobjekten (z.B. ein *Recordset*-Objekt) kann es notwendig sein, eine SQL-Anweisung über Code zu erstellen. Häufig basiert eine Abfrage dann auf Werten, die der Benutzer während der Laufzeit der Prozedur verändert. Dann müssen Werte aus Variablen oder Steuerelementen in die Abfrage einbezogen werden. Die JetEngine oder der SQL Server verarbeiten allerdings Variablen oder Steuerelemente nicht direkt. Daher muß in der Prozedur eine SQL-Anweisung als verkettete Folge von Zeichenketten und Informationen aus Variablen und Steuerelementen erstellt und an die Datenbank-Maschine übergeben werden. Diese Arbeitsweise wird von den verschiedenen DAO- oder ADO-Methoden unterstützt. So können Sie zur Methode *OpenRecordset* als Argument statt eines Tabellen- oder Abfragenamens auch eine SQL-Anweisung als Zeichenkette angeben.
- *Migration auf andere Datenbanken*. Wenn Sie Abfragen nur als Access-Objekte erstellen und speichern, sind Sie insoweit auf die Access-Welt angewiesen. Soweit Sie jedoch ihre Abfragen als SQL-Anweisungen formulieren, können Sie leichter in andere Datenbankprogramme migrieren. Letzteres kann beispielsweise erforderlich werden, weil die Anzahl der Datensätze in der Access-Datenbank für deren vergleichsweise beschränkte Leistungsfähigkeit zu groß geworden ist. Die Umwandlung einer Access-Datenbank (.mdb-Datei) in ein Access-Projekt (.adp-Datei) stellt eine derartige Migration dar, die Sie durch den Upsizing-Assistenten unterstützen lassen können.

Zusammengefaßt liegen die Vorteile beim Arbeiten mit SQL-formulierten Abfragen vor allem in größerer Flexibilität und Leistung: Sie können derartige Abfragen leichter an geänderte Bedingungen anpassen und unter gewissen Bedingungen Ergebnisse erzielen, die Sie mit Abfragen, die nur in der Abfrage-Entwurfsansicht erstellt wurden, nicht erreichen können.

14.3 Überblick über die möglichen SQL-Anweisungen

14.3.1 Datenmanipulations- und -definitionssprache (DML und DDL)

Der SQL-Sprachvorrat bezieht sich auf die beiden Gebiete der Datenbearbeitung bzw. -manipulation und der Datendefinition. Wegen der englischen Bezeichnungen *Data Manipulation Language* und *Data Definition Language* wird dies meistens kurz als *DML* und *DDL* bezeichnet. Die Anweisungen der DDL dienen vor allem dazu, neue Tabellen zu definieren oder Definitionsmerkmale bestehender Tabellen zu verändern und Berechtigungen festzulegen. Dieser Teil des SQL-Sprachumfangs wird hier nicht behan-

delt. Mit den DML-Anweisungen werden Daten vorhandener Tabellen bearbeitet. Im Vordergrund steht hierbei der lesende Zugriff in Form von Auswahlabfragen, die mit der SELECT-Anweisung formuliert werden.

14.3.2 Die DML-Anweisungen

Jede datenmanipulierende SQL-Aktion (= Abfrage) wird als Anweisung formuliert, die mit einem spezifischen Schlüsselwort beginnt. Dem Anweisungsschlüsselwort folgen dann weitere Angaben in Form reservierter Wörter wie FROM oder WHERE und benutzerspezifischer Angaben wie Namen von Tabellen und Feldern. Die Syntax jeder einzelnen Anweisung wird im nächsten Punkt 14.4, *Anweisungen, Operationen und Deklarationen zur Datenmanipulation*, behandelt. Hier wird zunächst ein grober Überblick gegeben. Die folgenden DML-Anweisungen stehen zur Verfügung:

SELECT

Mit dieser wohl wichtigsten Anweisung werden lesende Abfragen formuliert, sogen. *Auswahlabfragen*. Als Ergebnis wird eine Menge von Datensätzen zurückgegeben, die der speziellen Formulierung der SELECT-Anweisung entspricht und auch leer sein kann.

SELECT...INTO

Diese Anweisung definiert eine *Tabellenerstellungsabfrage*. Dabei werden die Datensätze einer Abfrage, die gleichzeitig mit der Anweisung angegeben werden muß, in eine neue Tabelle kopiert. So betrachtet stellt diese Anweisung implizit auch eine DDL-Anweisung dar.

INSERT INTO

Mit dieser Anweisung wird eine *Anfügeabfrage* erzeugt: Die Datensätze einer Abfrage, die gleichzeitig mit der Anweisung angegeben werden muß, werden einer bestehenden Tabelle hinzugefügt.

UPDATE

Mit der UPDATE-Anweisung wird eine *Aktualisierungsabfrage* erzeugt, die Werte in Feldern einer angegebenen Tabelle aufgrund angegebener Kriterien und Ausdrücke ändert.

DELETE

Mit dieser Anweisung wird eine *Löschabfrage* erstellt, die Datensätze aus einer oder mehreren aufgeführten Tabellen löscht, sofern diese die angegebenen Bedingungen erfüllen.

TRANSFORM

Diese Anweisung, die eine Erweiterung von Access gegenüber dem ANSI-Standard darstellt, erstellt eine *Kreuztabellenabfrage*.

14.4 Anweisungen, Operationen und Deklarationen zur Datenmanipulation

14.4.1 SELECT-Anweisung

Beschreibung

Diese Anweisung dient zur Formulierung einer Auswahlabfrage, mit der Datensätze aus der Datenbank zum lesenden Zugriff zurückgegeben werden.

Syntax

```
SELECT [Prädikat] { * | Tabelle.* | [Tabelle.]Feld1 [AS Alias1] [,
[Tabelle.]Feld2 [AS Alias2] [, ...]]}
FROM Tabellenausdruck [, ...] [IN ExterneDatenbank]
[WHERE... ]
[GROUP BY... ]
[HAVING... ]
[ORDER BY... ]
[WITH OWNERACCESS OPTION]
```

Bemerkungen

Die minimal erforderliche Syntax für eine SELECT-Anweisung ist:

```
SELECT Felder FROM Tabelle
```

Sie können ein Sternchen (*) verwenden, um alle Felder in einer Tabelle auszuwählen. Im folgenden Beispiel werden alle Felder in der Tabelle *Kunden* ausgewählt:

```
SELECT * FROM Kunden;
```

Wenn derselbe Feldname in mehreren Tabellen des FROM-Abschnitts vorkommt, müssen Sie den Tabellennamen und den Punktoperator (.) vor dem Feldnamen angeben. Im folgenden Beispiel enthält sowohl die Tabelle *Artikel* als auch die Tabelle *Lieferanten* das Feld *FirmenCode*. Die SQL-Anweisung wählt das Feld *ArtikelNr* und das Feld *Firma* aus einer der beiden Tabellen aus. Diese beiden Felder dürfen jeweils nicht in beiden Tabellen vorkommen.

```
SELECT ArtikelNr, Firma FROM Artikel INNER JOIN Lieferanten
WHERE Artikel.FirmenCode = Lieferanten.FirmenCode;
```

Beispiel

Das folgende Beispiel ergibt Werte für die Felder *Artikel-Nr* und *Artikelname* für alle Datensätze der Tabelle *Artikel*, die keinen Auslaufartikel beschreiben. Die Ausgabe erfolgt sortiert nach den Werten des Feldes *Artikelname* in aufsteigender Richtung.

```
SELECT [Artikel-Nr], Artikelname FROM Artikel
WHERE (((Auslaufartikel)=No))
ORDER BY Artikelname;
```

14.4.2 SELECT...INTO-Anweisung

Beschreibung

Mit dieser Anweisung wird eine Tabellenerstellungsabfrage formuliert. Dabei werden die Datensätze mit den spezifizierten Feldern aus der zu FROM angegebenen Quelle in eine neue Tabelle kopiert. Die Felder der neuen Tabelle bekommen die Datentypen der Felder aus der Quelle.

Syntax

```
SELECT Feld1[, Feld2[, ...]] INTO NeueTabelle [IN ExterneDatenbank]
FROM Quelle
```

Bemerkungen

Wenn Sie die SQL-Anweisung von einer VBA-Prozedur aus aufrufen und der mit *NeueTabelle* angegebene Name mit dem einer bereits bestehenden Tabelle übereinstimmt, tritt ein auffangbarer Fehler auf. Auf der allgemeinen Access-Ebene erfolgt eine entsprechende Fehlermeldung.

Beispiele

Das folgende Beispiel kopiert sämtliche Datensätze der Tabelle *Artikel* mit allen Feldern in die neue Tabelle *ArtikelTmp*.

```
SELECT * INTO [ArtikelTmp] FROM Artikel;
```

Das folgende Beispiel berechnet den Ausdruck [Einzelpreis]*[Anzahl] unter dem Namen *Endpreis*, summiert dieses Ergebnis für jeden Mitarbeiter (definiert durch seinen Vor- und Nachnamen) auf und kopiert die resultierenden Datensätze – sortiert nach dem Inhalt des berechneten Endpreises – in die neue Tabelle *Bestellsummen*.

```
SELECT Nachname, Vorname, Sum([Einzelpreis]*[Anzahl]) AS Endpreis
INTO Bestellsummen
FROM (Personal INNER JOIN Bestellungen ON Personal.[Personal-Nr] =
Bestellungen.[Personal-Nr]) INNER JOIN Bestelldetails ON
Bestellungen.[Bestell-Nr] = Bestelldetails.[Bestell-Nr]
GROUP BY Nachname, Vorname
ORDER BY Sum([Einzelpreis]*[Anzahl]);
```

14.4.3 INSERT INTO-Anweisung

Beschreibung

Die Anweisung INSERT INTO fügt einer Tabelle einen oder mehrere Datensätze hinzu. Dies wird als *Anfügeabfrage* bezeichnet.

Syntax

Abfrage zum Anfügen einer Gruppe von Datensätzen:

```
INSERT INTO Ziel [IN ExterneDatenbank] [(Feld1[, Feld2[, ...]])]
SELECT [Quelle.]Feld1[, Feld2[, ...]
FROM Tabellenausdruck
```

Abfrage zum Anfügen eines einzelnen Datensatzes:

```
INSERT INTO Ziel [(Feld1[, Feld2[, ...]])]
VALUES (Wert1[, Wert2[, ...])
```

Dabei bedeuten:

Ziel: Der Name der Tabelle, an die Datensätze angefügt werden sollen.

Quelle: Der Name der Tabelle, aus der Datensätze kopiert werden sollen.

Tabellenausdruck: Der Name der Tabelle(n), aus der oder denen Datensätze eingefügt werden. Bei diesem Argument kann es sich um den Namen einer einzelnen Tabelle oder eine zusammengesetzte Datenquelle (als Ergebnis einer JOIN-Operation) oder um eine gespeicherte Abfrage handeln.

Wert1, Wert2: Die Werte für die Felder des neuen Datensatzes. Jeder Wert wird in das Feld eingefügt, das der Position des Wertes in der Liste entspricht: Wert1 in Feld1, Wert2 in Feld2 usw. Alle Werte müssen in Anführungszeichen (» «) eingeschlossen und die Werte durch Kommas voneinander getrennt werden.

Beispiele

Das folgende Beispiel fügt alle Datensätze der Tabelle oder Abfrage *KundenTmp* der Tabelle *Kunden* hinzu, wobei *Kunden* bereits existieren muß.

```
INSERT INTO Kunden SELECT * FROM KundenTmp;
```

Das folgende Beispiel berechnet den Ausdruck [Einzelpreis]*[Anzahl] unter dem Namen *Endpreis*, summiert dieses Ergebnis für jeden Mitarbeiter (definiert durch seinen Vor- und Nachnamen) auf und fügt die resultierenden Datensätze – sortiert nach dem Inhalt des berechneten Endpreises – der Tabelle *Bestellsummen* an. Dieses Beispiel ist identisch mit dem Beispiel zur SELECT...INTO-Anweisung mit dem Unterschied, daß dort die Tabelle *Bestellsummen* neu kreiert wird und dann die Datensätze eingefügt werden, während hier die Existenz dieser Tabelle vorausgesetzt wird.

```
INSERT INTO Bestellsummen ( Nachname, Vorname, Endpreis )
SELECT Nachname,Vorname, Sum([Einzelpreis]*[Anzahl]) AS Endpreis
FROM Personal INNER JOIN (Bestellungen INNER JOIN Bestelldetails ON
Bestellungen.[Bestell-Nr] = Bestelldetails.[Bestell-Nr]) ON
Personal.[Personal-Nr] = Bestellungen.[Personal-Nr]
GROUP BY Personal.Nachname, Personal.Vorname
ORDER BY Sum([Einzelpreis]*[Anzahl]);
```

14.4.4 UPDATE-Anweisung

Beschreibung

Mit der UPDATE-Anweisung wird eine *Aktualisierungsabfrage* erzeugt, die Werte in Feldern einer angegebenen Tabelle aufgrund angegebener Kriterien und Ausdrücke ändert.

Syntax

```
UPDATE Tabelle
SET NeuerWert
WHERE Kriterien;
```

Dabei bedeutet:

NeuerWert: Ein Ausdruck, der den Wert angibt, der in einem bestimmten Feld in den aktualisierten Datensätzen eingefügt werden soll.

Beispiel

Das folgende Beispiel erhöht alle Einzelpreise der Artikel des Lieferanten mit der Nummer 12 um 10%.

```
UPDATE Artikel SET Einzelpreis = Einzelpreis*1.1
WHERE ([Lieferanten-Nr]=12);
```

Das folgende Beispiel verringert die Einzelpreise aller Artikel der Firma *Chippys Basar* auf 79% des bisherigen Preises.

```
UPDATE Lieferanten INNER JOIN Artikel
ON Lieferanten.[Lieferanten-Nr] = Artikel.[Lieferanten-Nr] SET
Einzelpreis = Einzelpreis * .79
WHERE Firma = 'Chippys Basar' AND Auslaufartikel = No;
```

14.4.5 DELETE-Anweisung

Beschreibung

Diese Anweisung erstellt eine Löschabfrage, die Datensätze in der(n) im FROM-Abschnitt aufgeführten Tabelle(n) löscht, sofern diese die im WHERE-Abschnitt angegebene Bedingung erfüllen.

Syntax

```
DELETE [Tabelle.*]
FROM Tabelle
WHERE Kriterien
```

Bemerkungen

Mit DELETE können Sie auch Datensätze aus Tabellen entfernen, die in einer 1:n-Beziehung zu anderen Tabellen stehen. Wenn Löschweitergabe vereinbart ist, werden auch die Datensätze auf der n-Seite gelöscht, die mit dem Datensatz auf der 1-Seite, der durch die Abfrage gelöscht wird, in Beziehung stehen.

Eine Löschabfrage löscht stets ganze Datensätze, nicht nur Daten in bestimmten Feldern. Wenn Sie letzteres erreichen möchten, müssen Sie eine Aktualisierungsabfrage erstellen, die die Werte auf Null setzt.

Beispiele

Das folgende Beispiel löscht alle Datensätze der Tabelle *Projekte*, für die das Feld *Storniert* den Wert *Ja* hat.

```
DELETE * FROM Projekte WHERE Storniert = Yes;
```

14.4.6 TRANSFORM-Anweisung

Beschreibung

Erstellt eine Kreuztabelle.

Syntax

```
TRANSFORM AggFunktion
Auswahlanweisung
PIVOT Pivot-Feld [IN (Wert1[, Wert2[, ...]])]
```

Dabei bedeuten:

AggFunktion: Eine SQL-Aggregatfunktion (wie *Count* oder *Sum*), die Operationen auf den ausgewählten Daten ausführt.

Auswahlanweisung: Eine SELECT-Anweisung.

Pivot-Feld: Das Feld oder der Ausdruck, mit dem Spaltenüberschriften in der Ergebnismenge der Abfrage erstellt werden sollen.

Wert1, Wert2: Feststehende Werte, die als Spaltenüberschriften verwendet werden. Mit dieser optionalen Angabe kann eine Spaltenfixierung erreicht werden.

Bemerkungen

Diese Anweisung ist nicht im ANSI-Standard enthalten.

Beispiel

Das folgende Beispiel erstellt für die Tabelle *Bestellungen* eine Kreuztabelle: Zeilenfeld ist *Bestimmungsland*, die Kategorien des Spaltenfeldes werden von den Jahreszahlen gebildet, die die Funktion *Year* für das Feld *Bestelldatum* ergibt, und als Werte werden die Anzahl (Funktion *Count*) der Bestellungen in den Kreuzungspunkten ausgewiesen. Die Kategorien der Spalten sind fest auf die Werte 1994, 1995 und 1996 eingestellt mit der Folge, daß weitere Spalten für ggf. in den Daten vorhandene Jahreszahlen wie 1993 oder 1997 nicht ausgewiesen werden.

```
TRANSFORM Count([Bestell-Nr])
SELECT Bestimmungsland
FROM Bestellungen
GROUP BY Bestimmungsland
PIVOT Year([Bestelldatum]) In (1994,1995,1996);
```

14.4.7 JOIN-Operationen

Beschreibung

Kombiniert Datensätze aus zwei Tabellen, wenn ein gemeinsames Feld dieselben (=) oder für einen anderen Vergleichsoperator sich entsprechende Werte enthält.

Syntax

```
FROM Tabelle1 INNER JOIN Tabelle2 ON Tabelle1.Feld1 VerglOp
Tabelle2.Feld2
FROM Tabelle1 [INNER | LEFT | RIGHT ] JOIN Tabelle2 ON
Tabelle1.Feld1 VerglOp Tabelle2.Feld2
```

Dabei bedeuten:

Tabelle1, Tabelle2: Die Namen der Tabellen, aus denen Datensätze kombiniert werden.

Feld1, Feld2: Die Namen der Felder, die verknüpft werden. Nichtnumerische Felder müssen im Datentyp und in der Art der enthaltenen Daten übereinstimmen, können aber unterschiedliche Feldnamen haben.

VerglOp: Ein beliebiger relationaler Vergleichsoperator: =, <, >, <=, >= oder <>.

Mit der folgenden Syntax können Sie in einer JOIN-Anweisung auch mehrere ON-Abschnitte verknüpfen:

```
SELECT Felder
FROM Tabelle1 INNER JOIN Tabelle2
ON Tabelle1.Feld1 VerglOp Tabelle2.Feld1 AND
ON Tabelle1.Feld2 VerglOp Tabelle2.Feld2) OR
ON Tabelle1.Feld3 VerglOp Tabelle2.Feld3)];
```

Mit der folgenden Syntax können Sie JOIN-Anweisungen verschachteln:

```
SELECT Felder
FROM Tabelle1 INNER JOIN
(Tabelle2 INNER JOIN [( ]Tabelle3
[INNER JOIN [( ]Tabellex [INNER JOIN ...)]
ON Tabelle3.Feld3 VerglOp Tabellex.Feldx)]
ON Tabelle2.Feld2 VerglOp Tabelle3.Feld3)
ON Tabelle1.Feld1 VerglOp Tabelle2.Feld2;
```

Bemerkungen

In der Entwurfsansicht des Abfragefensters können Sie (implizit) nur über den Vergleichsoperator = verfügen. In einer SQL-Formulierung stehen Ihnen dagegen auch die anderen Vergleichsoperatoren zur Verfügung. Wenn Sie eine derartige Abfrage in der SQL-Ansicht des Abfragefensters entwerfen, können Sie diese allerdings nicht mehr in der Entwurfsansicht darstellen. Das Abfrageergebnis läßt sich gleichwohl in der Datenblattansicht wiedergeben.

Beispiel

Das folgende Beispiel gibt Datensätze zurück, für die das den Tabellen *Personal* und *Projekte* gemeinsame Feld *PersonalCode* gleiche Werte hat. (Den Tabellen liegt eine 1:n-Beziehung zugrunde.) Es werden die Felder *Nachname*, *Vorname*, *Projektbezeichnung* und *Auftragswert* ausgegeben.

```
SELECT Nachname, Vorname, Projektbezeichnung, Auftragswert
FROM Personal INNER JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode;
```

Das folgende Beispiel eines LEFT JOIN unterscheidet sich vom vorangehenden nur dadurch, daß von der linken Seite (das ist hier die Tabelle *Personal*) auch diejenigen Datensätze wiedergegeben werden, denen kein Datensatz auf der rechten Seite entspricht.

```
SELECT Nachname, Vorname, Projektbezeichnung, Auftragswert
FROM Personal LEFT JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode;
```

14.4.8 UNION-Operation

Beschreibung

Erstellt eine Union-Abfrage, in der die Ergebnisse zweier oder mehrerer unabhängiger Abfragen oder Tabellen in der Senkrechten kombiniert werden.

Syntax

```
[TABLE] Abfrage1 UNION [ALL] [TABLE] Abfrage2 [UNION [ALL] [TABLE]
AbfrageN [ ... ]]
```

Bemerkungen

Sie können zwei oder mehr Abfragen, Tabellen und SELECT-Anweisungen in jeder beliebigen Kombination in einer einzelnen UNION-Operation zusammenstellen. Alle Abfragen in einer UNION-Operation müssen gleich viele Felder abrufen. Diese Felder müssen aber weder gleich groß noch vom selben Datentyp sein.

Beispiel

Im folgenden Beispiel werden alle Datensätze der Tabelle *Kunden* und die Datensätze der Tabelle *Lieferanten*, die Postleitzahlen zwischen 20000 und 29999 haben, zusammengeführt.

```
TABLE Kunden UNION ALL
SELECT *
FROM Lieferanten
WHERE (PLZ >= 20000 AND PLZ <= 29999);
```

14.4.9 PARAMETERS-Deklaration

Beschreibung

Deklariert die Namen und Datentypen der Parameter in einer Abfrage.

Syntax

```
PARAMETERS Name Datentyp [, Name Datentyp [, ...]]
```

Bemerkungen

Die PARAMETERS-Deklaration ist optional, muß jedoch jeder anderen Anweisung (einschließlich SELECT) vorangestellt werden, wenn sie verwendet wird. Das Deklarieren von Parametern hat im wesentlichen den Sinn, dafür Datentypen festzulegen. Sie können den Parameternamen in einem WHERE- oder HAVING-Abschnitt verwenden.

Beispiele

Das folgende Beispiel deklariert die beiden Parameter *Aufnahmegebühr* und *Beginn* und weist ihnen die Datentypen *Currency* (*Währung*) und *DateTime* (*Datum/Zeit*) zu.

PARAMETERS [Aufnahmegebühr:] Currency, Beginn DateTime;

Im folgenden Beispiel werden die beiden Parameter *Aufnahmegebühr:* und *Eintritt:* mit jeweils einem Datentyp deklariert. Die anschließend formulierte Auswahlabfrage erwartet dann die Übergabe der Werte für diese beiden Parameter, die in der WHERE-Klausel angeführt werden.

```
PARAMETERS [Aufnahmegebühr:] Currency, [Eintritt:] DateTime;
SELECT MitgliedsNr, Mitgliedsbeitrag
FROM Mitglieder
WHERE Aufnahmegebühr = [Aufnahmegebühr:]
AND Eintrittsdatum = [Eintritt:];
```

14.5 Klauseln

14.5.1 FROM

Beschreibung

Die FROM-Klausel gibt die Tabellen oder Abfragen an, in denen die in der SELECT-Anweisung aufgeführten Felder enthalten sind.

Syntax

```
SELECT Feldliste
FROM Tabellenausdruck [IN ExterneDatenbank]
```

Dabei bedeutet:

Tabellenausdruck: Ein Ausdruck, der eine oder mehrere Tabellen angibt, aus denen Daten abgerufen werden. Der Ausdruck kann ein einzelner Tabellenname, ein Name einer gespeicherten Abfrage oder eine Verknüpfung (INNER JOIN, LEFT JOIN oder RIGHT JOIN) sein.

Bemerkungen

Die Angabe von FROM muß auf alle SELECT-Anweisungen folgen. Die Reihenfolge der Tabellennamen in *Tabellenausdruck* ist ohne Belang.

Beispiele

Das folgende Beispiel gibt alle Datensätze mit allen Feldern der Tabelle *Kunden* zurück:

SELECT * FROM Kunden;

Das folgende Beispiel gibt die Werte des Feldes *Nachname* für alle Datensätze wieder, in denen die Werte der beiden Felder mit dem jeweiligen Namen *PersonalCode* in den Tabellen *Personal* und *Projekte* gleich sind.

```
SELECT Nachname
FROM Personal INNER JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode;
```

14.5.2 GROUP BY

Beschreibung

Mit dieser Klausel werden alle Datensätze, die in der angegebenen Gruppenfeldliste dieselben Werte haben, zu einem einzelnen Datensatz zusammengefaßt. Für jeden zusammengefaßten Datensatz wird für Felder, für die eine

SQL-Aggregatfunktion wie *Sum* oder *Count* in der SELECT-Anweisung angeben ist, ein Wert nach dieser Funktion berechnet und für das jeweilige Feld ausgegeben.

Syntax

```
SELECT Feldliste
FROM Tabelle
WHERE Kriterien
[GROUP BY Gruppenfeldliste]
```

Bemerkungen

Nullwerte in GROUP BY-Feldern werden berücksichtigt und gruppiert. Nullwerte werden jedoch in keiner SQL-Aggregatfunktion ausgewertet. Verwenden Sie den WHERE-Abschnitt, um Zeilen auszuschließen, die nicht gruppiert werden sollen, und verwenden Sie den HAVING-Abschnitt, um Datensätze nach dem Gruppieren zu selektieren.

Beispiel

Das folgende Beispiel faßt alle Datensätze, die zu demselben Nachnamen gehören, jeweils zu einem Datensatz zusammen und gibt je zusammengefaßten Datensatz die Summe der Auftragswerte je Mitarbeiter aus.

```
SELECT Nachname, Sum(Auftragswert)
FROM Personal INNER JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode
GROUP BY Nachname;
```

14.5.3 HAVING

Beschreibung

Mit dieser Klausel wird bestimmt, welche der gruppierten Datensätze in einer SELECT-Anweisung mit einer GROUP BY-Klausel ausgegeben werden sollen.

Syntax

```
SELECT Feldliste
FROM Tabelle
WHERE Auswahlkriterien
GROUP BY Gruppenfeldliste
[HAVING Gruppenkriterien]
```

Bemerkungen

Die HAVING-Klausel hat für die gruppierten Datensätze dieselbe Funktion wie die WHERE-Klausel für die noch ungruppierten: HAVING bestimmt, welche Datensätze zurückgegeben werden, *nachdem* Datensätze mit GROUP BY gruppiert und Felder entsprechend der angegebenen Aggregatfunktionen berechnet wurden.

Beispiele

Das folgende Beispiel gibt von den nach dem Nachnamen gruppierten Datensätzen nur diejenigen aus, deren aggregierter Auftragswert mehr als 200000 beträgt.

```
SELECT Nachname, Sum(Auftragswert)
FROM Personal INNER JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode
GROUP BY Nachname
HAVING Sum(Auftragswert) > 200000;
```

Das folgende Beispiel gibt nach dem Nachnamen gruppierte Datensätze aus. Für die Gruppierung und Aggregierung werden nur solche Datensätze berücksichtigt, deren Auftragswert im ungruppierten Datensatz größer als 100000 ist. Die gruppierten Datensätze selbst werden nicht gefiltert.

```
SELECT Nachname, Sum(Auftragswert)
FROM Personal INNER JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode
WHERE Auftragswert>100000
GROUP BY Nachname;
```

Das folgende Beispiel stellt eine Kombination der beiden vorangehenden Beispiele dar: Für die Gruppierung und Aggregierung werden nur solche Datensätze berücksichtigt, deren Auftragswert im ungruppierten Datensatz größer als 100000 ist. Sodann werden nur diejenigen gruppierten Datensätze zurückgegeben, deren aggregierter Auftragswert mehr als 200000 beträgt.

```
SELECT Nachname, Sum(Auftragswert)
FROM Personal INNER JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode
WHERE Auftragswert>100000
GROUP BY Nachname
HAVING Sum(Auftragswert)>200000;
```

14.5.4 IN

Beschreibung

Mit dem reservierten Wort IN geben Sie eine externe Datenbank an, in der Tabellen, auf die zugegriffen werden soll, liegen.

Syntax

So geben Sie eine Zieltabelle an:

```
[SELECT | INSERT] INTO Zieltabelle IN
{Pfad | ["Pfad" "Typ"] | ["" [Typ; DATABASE = Pfad]]}
```

So geben Sie eine Quelltabelle an:

```
FROM Tabellenausdruck IN
{Pfad | ["Pfad" "Typ"] | ["" [Typ; DATABASE = Pfad]]}
```

Dabei bedeuten:

Tabellenausdruck: Der Name einer oder mehrerer Tabellen, aus denen Daten gelesen werden sollen. Dieses Argument kann ein einzelner Tabellen-/Abfragename sein oder eine zusammengesetzte Quelle, die sich aus einer Verknüpfung mit INNER JOIN, LEFT JOIN oder RIGHT JOIN ergibt.

Pfad: Der vollständige Pfad für die Datei, welche die Tabelle enthält.

Typ: Die Bezeichnung für den Typ der externen Datenbank, wenn es sich nicht um eine Microsoft Jet-Datenbank handelt (z.B. dBASE III, dBASE IV, Paradox 3.x oder Paradox 4.x).

Bemerkungen

In einigen Fällen verweist das Argument *Pfad* auf den Ordner, der die Datenbankdateien enthält. Beim Arbeiten mit dBASE-, FoxPro- oder Paradox-Datenbanktabellen gibt das Pfadargument z.B. den Ordner an, in dem die .DBF- oder .DB-Dateien enthalten sind. Der Dateiname für die Tabelle wird aus dem Argument *Zieltabelle* oder *Tabellenausdruck* abgeleitet. Sie geben eine Datenbank an, die keine Microsoft Jet-Datenbank ist, indem Sie ein Semikolon (;) an den Namen anhängen und ihn in Apostrophe (' ') oder Anführungszeichen (» «) einschließen. Beispielsweise können Sie sowohl 'dBASE IV;' als auch »dBASE IV;« angeben. Sie können auch das reservierte Wort DATABASE verwenden, um die externe Datenbank anzugeben. Die folgenden Zeilen sind gleichwertig:

```
FROM Tabelle IN "" [dBASE IV; DATABASE=C:\GER\HH\ABSATZ;];
FROM Tabelle IN " C:\GER\HH\ABSATZ " "dBASE IV;"
```

Die Verwendung von IN anstelle verknüpfter Tabellen führt zu schnelleren Datenzugriffen.

Beispiele

Das folgende Beispiel gibt Datensätze mit den Feldern *Firma*, *Name* und *Vorname* aus der Tabelle *Adressen* der Datenbank mit dem Pfad *c:\Privdat\adrlist.mdb* aus, wobei nur Adressen berücksichtigt werden, die im Feld *Firma* einen Eintrag haben. Die Datenbank adrlist.mdb hat Access-Format.

```
SELECT Firma, Name, Vorname
FROM Adressen
IN "c:\Privdat\adrlist.mdb"
WHERE Not IsNull(Firma) ;
```

Vgl. die verschiedenen Beispiele der Online-Hilfe zum Stichwort IN-Abschnitt, um sich über die Angaben für externe Datenbanken im anderen als Access-Format zu informieren.

14.5.5 ORDER BY

Beschreibung

Sortiert die Datensätze einer Abfrage nach einem oder mehreren angegebenen Feldern in aufsteigender oder absteigender Reihenfolge.

Syntax

```
SELECT Feldliste
FROM Tabelle
WHERE Auswahlkriterien
[ORDER BY Feld1 [ASC | DESC ][, Feld2 ... ] ][, ... ]]]
```

Mit den reservierten Wörtern ASC bzw. DESC geben Sie an, ob in auf- bzw. absteigender Folge sortiert werden soll. Bei Fortlassen dieser Angabe wird aufsteigend sortiert.

Beispiele

Das folgende Beispiel gibt alle Datensätze der Tabelle *Rechnungen* mit den Feldern *KundenName* und *Gesamtpreis* aus und sortiert in aufsteigender Folge nach dem Inhalt des Feldes *KundenName*, bei gleichem Kundennamen nach dem Gesamtpreis in absteigender Folge.

```
SELECT KundenName, Gesamtpreis
FROM Rechnungen
ORDER BY KundenName, Gesamtpreis DESC;
```

14.5.6 WHERE

Beschreibung

Mit dieser Klausel geben Sie an, welche Datensätze der im FROM-Abschnitt aufgeführten Tabellen von einer SELECT-, UPDATE- oder DELETE-Anweisung selektiert werden.

Syntax

```
SELECT Feldliste
FROM Tabellenausdruck
WHERE Kriterien
```

Dabei bedeutet:

Kriterien: Ein Ausdruck mit Kriterien, welche die Datensätze der Abfrage erfüllen müssen, damit sie in das Abfrageergebnis aufgenommen werden.

Bemerkungen

Die WHERE-Klausel filtert Datensätze, bevor diese ggf. gruppiert werden. Wenn Sie mit der GROUP BY-Klausel Datensätze zu Gruppen zusammengefaßt haben und diese auf Basis von Kriterien filtern wollen, müssen Sie die HAVING-Klausel verwenden.

Beispiele

Die folgende Abfrage gibt die Datensätze wieder, in denen das Gehalt größer als oder gleich 7000 ist.

```
SELECT Nachname, Gehalt
FROM Personal
WHERE Gehalt >= 7000;
```

Die folgende Abfrage enthält eine Kombination der WHERE- mit der HAVING-Klausel: Für die Gruppierung und Aggregierung werden nur solche Datensätze berücksichtigt, deren Auftragswert im ungruppierten Datensatz größer als 100000 ist. Sodann werden nur diejenigen gruppierten Datensätze zurückgegeben, deren aggregierter Auftragswert mehr als 200000 beträgt.

```
SELECT Nachname, Sum(Auftragswert)
FROM Personal INNER JOIN Projekte ON Personal.PersonalCode =
Projekte.PersonalCode
WHERE Auftragswert>100000
GROUP BY Nachname
HAVING Sum(Auftragswert)>200000;
```

14.6 Unterabfragen

Eine Unterabfrage ist eine SELECT-Anweisung, die entweder innerhalb einer SELECT-, SELECT...INTO-, INSERT...INTO-, DELETE- oder UPDATE-Anweisung oder innerhalb einer anderen Unterabfrage verschachtelt ist. Sie kann in drei Syntaxvarianten formuliert werden:

Syntax

```
Vergleich [ANY | ALL | SOME] (Unterabfrage als SQL-Anweisung)
Ausdruck [NOT] IN (Unterabfrage als SQL-Anweisung)
[NOT] EXISTS (Unterabfrage als SQL-Anweisung)
```

Dabei bedeuten:

Vergleich: Ein Ausdruck und ein Vergleichsoperator, der den Ausdruck mit den Ergebnissen der Unterabfrage vergleicht.

Ausdruck: Ein Ausdruck, nach dem im Abfrage-Ergebnis der Unterabfrage gesucht wird.

Unterabfrage als SQL-Anweisung: Eine SELECT-Anweisung, die dasselbe Format hat und nach denselben Regeln aufgebaut ist wie jede andere SELECT-Anweisung. Diese Anweisung muß in runde Klammern eingeschlossen sein.

Beispiele

Die folgende Abfrage gibt alle Firmen der Tabelle *Kunden* wieder, die nicht gleichzeitig auch in der Tabelle *Lieferanten* vorhanden sind. (Das gleiche Ergebnis könnten Sie auch mit einem Join erzielen, wenn Sie diesen mit dem Vergleichsoperator <> durchführen.)

```
SELECT Kunden.Firma
FROM Kunden, Lieferanten
GROUP BY Kunden.Firma
HAVING (((Kunden.Firma) Not In (SELECT Lieferanten.Firma from
Lieferanten)))
ORDER BY Kunden.Firma;
```

Im folgenden Beispiel werden die Datensätze aller Artikel gefiltert, deren Einzelpreis größer ist als der Einzelpreis irgendeines Artikels mit einem Rabatt von 25 Prozent oder mehr.

```
SELECT * FROM Artikel
WHERE Einzelpreis > ANY
(SELECT Einzelpreis FROM Bestelldetails
WHERE Rabatt >= .25);
```

Das folgende Beispiel gibt die Bestellungen der Tabelle *Bestellungen* aus, die Duplikate im Feld *VersandÜber* besitzen. Das heißt, es werden nur Datensätze mit Versendern (Feld *VersandÜber*), die mehr als eine Bestellung abgewickelt haben, wiedergegeben.

```
SELECT DISTINCTROW Bestimmungsland, [Kunden-Code], Empfänger,
VersandÜber
FROM Bestellungen
WHERE (((VersandÜber) In (SELECT [VersandÜber] FROM [Bestellungen]
As Tmp GROUP BY [VersandÜber] HAVING Count(*)>1 )))
ORDER BY VersandÜber;
```

14.7 Datumsangaben in Access-SQL-Anweisungen

Wenn Sie in der WHERE-Klausel das Argument *Kriterien* angeben, müssen Datumsangaben als Zeichenketten und im amerikanischen Datumsformat erfolgen, auch wenn Sie nicht die amerikanische Version des Microsoft Jet-Datenbankmoduls verwenden. Der 12. Juli 1999 z.B. wird in Deutschland in der Form 12.7.99 und in den USA in der Form 7/12/99 geschrieben. Datumsangaben müssen stets in #-Zeichen eingeschlossen werden.

Zur Suche nach Datensätzen vom 12. Juli 1999 müssen Sie die folgende SQL-Anweisung verwenden:

```
SELECT *
FROM Lieferungen
WHERE [Lieferdatum] = #7/12/99#;
```

Sie können auch die *DateValue*-Funktion verwenden, die die unter Microsoft Windows eingestellten Ländereinstellungen erkennt. Verwenden Sie z.B. die folgenden Codezeilen für Deutschland:

```
SELECT *
FROM Bestellungen
WHERE [Versanddatum] = DateValue('10.5.96');
```

Der Einsatz der Funktion *DateValue* empfiehlt sich vor allem, wenn Sie SQL-Code in VBA-Prozeduren verwenden. Dabei kommt es oft vor, daß Sie einen Datumswert aus einem Steuerelement oder einer Variablen beziehen und diesen im Kriterienausdruck einer WHERE-Klausel verwenden müssen. Das folgende Beispiel zeigt, wie Sie dabei vorgehen können: Das Formular *Projektauswahl* enthält das Textfeld *Projektbeginn*, in das ein Datumswert eingegeben wird. Es sollen alle Datensätze ausgegeben werden, deren Projektbeginn vor dem oder am ins Textfeld *Projektbeginn* eingegebenen Datum liegt. In der Prozedur werden die Werte der Felder *Projektbezeichnung* und *Projektbeginn* ins Testfenster geschrieben.

```
Sub ProjekteAusgeben()
Beginn = Forms!Projektauswahl!Projektbeginn
SQL = "SELECT Projektbezeichnung, Projektbeginn FROM "
SQL = SQL & "Projekte WHERE (((Projektbeginn) "
SQL = SQL & "<=DateValue('" & Beginn & "')));"
Set DB = CurrentDb()
Set RS1 = DB.OpenRecordset(SQL)
Do While Not RS1.EOF
Debug.Print RS1.Projektbezeichnung, RS1.Projektbeginn
RS1.MoveNext
Loop
End Sub
```

Kapitel 15

Formulare: Grundlagen

15.1 Was ist ein Formular?

Rein theoretisch könnten Sie Ihre Datenbank allein mit Tabellen und Abfragen verwalten, denn alle Daten (mit Ausnahme von Bildern) lassen sich in der Datenblattansicht von Tabellen und Abfragen eingeben und bearbeiten. Formulare dienen dazu, die Datenbearbeitung komfortabler und sicherer zu gestalten. Dies gilt ganz allgemein auch für andere Datenbankprogramme, so daß der Einsatz von Formularen sich generell empfiehlt. Speziell beim DBMS Access kommt jedoch hinzu, daß die hier angebotenen Formulare sehr weitgehend die Möglichkeiten der grafischen Benutzeroberfläche ausnutzen und eine objektorientierte Handhabung der Formularelemente ermöglichen. Dies hat vor allem folgende bedeutsame Konsequenzen:

- Leistungsfähige Formulare können mit wenig Aufwand erstellt werden. Insbesondere die Möglichkeit, mehrere Typen von AutoFormularen zu wählen oder die Unterstützung des Formular-Assistenten in Anspruch zu nehmen, ermöglicht es, praktisch ohne Vorkenntnisse funktionsfähige Formulare, die bereits durchschnittlichen Ansprüchen genügen, zu erzeugen.

Bild 15.1: Wenn Sie ein neues Formular erstellen, können Sie verschiedene AutoFormulare wählen oder sich vom Formular-Assistenten unterstützen lassen.

- Der Umgang mit fertigen Access-Formularen ist im allgemeinen so einfach, daß diese von Dritten nach sehr kurzer Einweisung bearbeitet werden können. Vor allem die reichhaltigen Möglichkeiten optischer Unterstützung bestimmter Formularelemente (z.B. Bilder statt erklärende Texte, Symbole zum Klicken mit kurzen Hilfetexten statt Tastenkombinationen usw.) und der Einsatz von Befehlsschaltflächen machen den Umgang mit Access-Formularen sehr leicht.

Bild 15.2: Dieses Formular, das auch ein Unterformular enthält, wurde mit dem Formular-Assistenten erstellt und nur ganz geringfügig bearbeitet. Es kann in der vorliegenden Form leicht und bequem für die Datenbearbeitung benutzt werden. Dieses Formular wurde im Einführungskapitel (Kap. 5) erstellt, vgl. Punkt 5.11, Formular »Personal« und »Projekte« erstellen.

- Formulare können auf unterschiedliche Weise gestaltet werden. Dies wird zum einen dadurch ermöglicht, daß Access für Formulare viele verschiedene sogen. Steuerelemente (z.B. Textfelder zum Eingeben von Text und Zahlen, Optionsfelder, Kontrollkästchen, Kombinationsfelder mit aufschlagbaren Dropdown-Listen usw.) zur Verfügung stellt. Zum anderen resultiert die Vielfalt daraus, daß jedes der Steuerelemente wiederum sehr unterschiedlich ausgestaltet werden kann. Die reichhaltigen Gestaltungsmöglichkeiten würden eher abschrecken, wenn es kompliziert wäre, sie zu nutzen. Tatsächlich ist es in den meisten Fällen jedoch sehr einfach, ein Textfeld, Listenfeld, Kontrollkästchen usw. so auszugestalten, daß es den persönlichen Wünschen (oder den für Drittbenutzer unterstellten) gerecht wird. Dies liegt vor allem am Konzept der Objektorientierung, zu dem u.a. gehört, daß Objekte (z.B. ein Formular, ein Steuerelement des Formulars usw.) *Eigenschaften* besitzen, die durch einfaches Eintragen in kontextabhängig angebotene Listen festgelegt – im Sprachgebrauch von

Access *eingestellt* – werden können. Darüber hinaus können Sie sich beim Hinzufügen oder Bearbeiten eines Steuerelements von verschiedenen Assistenten unterstützen lassen.

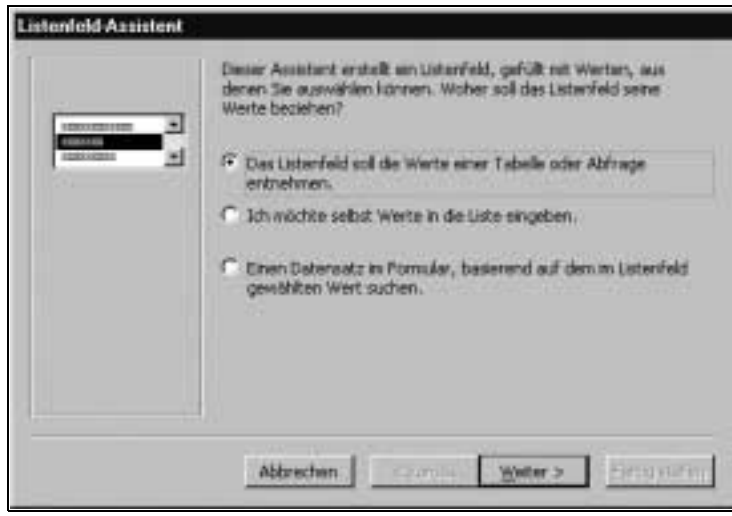


Bild 15.3: Steuerelement-Assistenten unterstützen Sie, wenn Sie dieses wünschen, beim Einfügen eines neuen Steuerelementes, hier eines Listenfeldes.

- Zu den Steuerelement- und Formulareigenschaften, die Sie nach Ihren Wünschen einstellen können, gehört auch eine Gruppe, die als *Ereignis* bezeichnet wird. So gibt es beispielsweise für Textfelder die Ereignis-Eigenschaft *Vor Aktualisierung*. Sie können dieser Eigenschaft z.B. ein Makro zuweisen, welches prüft, ob der in das Feld eingegebene Wert in diesem bestimmten Kontext zulässig ist, und damit eine Gültigkeitsprüfung vornehmen, die weit über die Möglichkeiten hinaus geht, die Sie sonst zur Eigenschaft *Gültigkeitsregel* angeben können. Entsprechend können Sie z.B. *Beim Schließen* eines Formulars veranlassen, daß gleichzeitig ein bestimmtes anderes Formular geöffnet oder, ein anderes Beispiel, die gesamte Anwendung beendet wird.

In Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, wurde bereits *zusammenhängend* gezeigt, wie Sie mit dem Formular-Assistenten Formulare erstellen und diese anschließend bearbeiten können. In diesem Kapitel geht es darum, die *einzelnen* Formularelemente systematisch zu besprechen.

Bild 15.4: Das Eigenschaftenfenster für das Formular zeigt hier einen Teil der Ereigniseigenschaften eines Formulars an. Für das Ereignis »Vor Aktualisierung« ist hier das Makro »FrmHauptauftrag.Plausibilität« eingetragen, das ausgeführt wird, wenn die Daten dieses Formulars aktualisiert werden sollen.

15.2 Wie Sie Formulare erstellen

In diesem Abschnitt werden die prinzipiellen Möglichkeiten und Wege aufgezeigt, Formulare zu erstellen. Dabei wird nicht auf Einzelheiten eingegangen. Diese werden vielmehr in den weiteren Abschnitten des vorliegenden Kapitels sowie in den Kapiteln 16 bis 20 behandelt.

Datenherkunft des Formulars

Normalerweise dient ein Formular dazu, Daten einzugeben, zu lesen oder zu bearbeiten. Zu diesem Zweck muß das Formular mit einer Datenquelle verbunden sein, denn es kann selbst keine Daten speichern, sondern nur vermitteln. Als Datenquelle kann eine Tabelle oder Abfrage dienen, letztere auch in Form einer SQL-SELECT-Anweisung. Wenn Sie ein Formular vom Formular-Assistenten erstellen lassen, werden Sie nach der Tabelle/Abfrage gefragt, die dem Formular zugrunde gelegt werden soll. Wenn Sie ein Formular ohne den Formular-Assistenten erstellen, müssen Sie die Tabelle oder Abfrage (oder eine entsprechende SQL-SELECT-Anweisung) als Eintrag zur Formulareigenschaft *Datenherkunft* selbst angeben. Es ist jedoch auch möglich, ein Formular ohne Anbindung an eine Tabelle oder Abfrage zu erstellen. Solch ein Formular kann z.B. sinnvoll als Dialogfeld oder Auswahltafel mit verschiedenen Befehlsschaltflächen dienen.

Formular mit Formular-Assistent erstellen

Am einfachsten ist es, ein Formular mit dem Formular-Assistenten herzustellen, vgl. im einzelnen Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, Punkte 5.9, *Formular Personal erstellen*, 5.11, *Formu-*

lar Personal und Projekte erstellen und 5.12, *Position und Größe der Steuerelemente im Formular Personal und Projekte ändern*. Der Formular-Assistent erstellt zwar immer nur Formulare mit gewissen Standardanordnungen der Steuerelemente oder in bestimmten Standardansichten, jedoch sind die Ergebnisse in vielen Fällen ausreichend. Dies gilt vor allem für Formulare mit relativ wenigen Feldern. Aber selbst dann, wenn Sie mit dem Ergebnis des Formular-Assistenten nicht zufrieden sind, lohnt es sich meistens, ein vom Formular-Assistenten erstelltes Formular oder ein AutoFormular als Ausgangspunkt für ein Formular zu nehmen, das Sie dann Ihren Wünschen entsprechend ändern. Dies gilt vor allem, wenn Sie in der zugrundeliegenden Tabelle sorgfältig Feldeigenschaften wie Standardwerte, Gültigkeitsregeln usw. festgelegt haben, denn der Formular-Assistent überträgt diese Eigenschaften auf die entsprechenden Steuerelemente im Formular. Allerdings können Sie diesen Effekt weitgehend auch erreichen, wenn Sie Steuerelemente in ein leeres Formular durch Ziehen der Felder aus der Feldliste in der Entwurfsansicht des Formulars einfügen, vgl. genauer in diesem Kapitel, Punkt 15.5.2, *Steuerelemente einfügen*.

Formular ohne Formular-Assistent erstellen

Sie fordern ein neues Formular grundsätzlich an, indem Sie aus dem Datenbankfenster heraus, während die Objektliste *Formulare* aktiviert ist, auf die Schaltfläche *Neu* klicken. Weiterhin können Sie im Datenbankfenster auf einen der Einträge *Erstellt ein Formular in der Entwurfsansicht* (erstellt unmittelbar ein leeres Formular in der Entwurfsansicht) oder *Erstellt ein Formular unter Verwendung des Assistenten* (ruft den Formular-Assistenten auf) klicken, vgl. Bild 15.5. Alternativ wählen Sie in der Dropdown-Liste der Symbol-Schaltfläche *Neues Objekt* den Eintrag *Formular.* (Die Symbol-Schaltfläche *Neues Objekt* steht standardmäßig zur Verfügung, wenn ein Formular aktiv ist.)

Wenn Sie den Weg über die Schaltfläche *Neu* wählen, müssen Sie sich in einem Dialogfeld entscheiden, ob Sie eines der drei *AutoFormulare* erstellen, den Schritten des Formular-Assistenten folgen oder ein leeres Formular erstellen wollen. Wenn Sie sich für ein leeres Formular entscheiden, erstellt Access ein solches und zeigt es in der Entwurfsansicht des Formularfensters an, damit Sie das Formular entsprechend einrichten können. Sie müßten dann vor allem die *Datenherkunft* (Eigenschaft des Formulars) angeben, falls das Formular der Datenbearbeitung dienen soll. Wie Sie Formulareigenschaften einstellen, erfahren Sie weiter unten in diesem Kapitel, vgl. Punkt 15.7, *Eigenschaften von Steuerelementen, Formular und Bereichen einstellen*. Darüber hinaus müssen Sie in diesem Falle alle Steuerelemente (Textfelder, Kontrollkästchen usw.) selbst einfügen, sei es mit der Toolbox, sei es durch Ziehen aus der Feldliste.

Formular kopieren oder importieren

Vielleicht verfügen Sie bereits über ein Formular, das dem neu zu erstellenden ähnlich ist. Für diesen Fall bietet Access Ihnen zwei Möglichkeiten.

Das Formular befindet sich in der aktuellen Datenbank. Dann können Sie es kopieren. Gehen Sie dazu folgendermaßen vor:

- Markieren Sie im Datenbankfenster den Namen des zu kopierenden Formulars, und kopieren Sie es mit dem Befehl *Kopieren* aus dem Menü *Bearbeiten* oder der Tastenkombination Strg+C in die Zwischenablage, vgl. Bild 15.5.

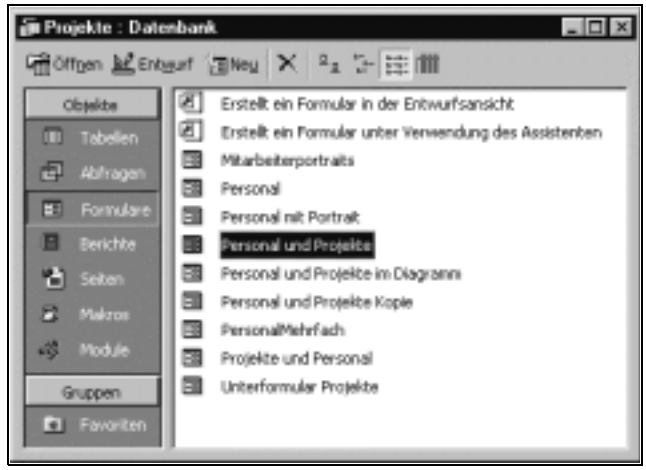


Bild 15.5: Um beispielsweise das Formular »Personal« und »Projekte« zu kopieren, markieren Sie es im Datenbankfenster und kopieren es in die Zwischenablage.

- Fügen Sie das Formular aus der Zwischenablage mit dem Befehl *Einfügen* aus dem Menü *Bearbeiten* oder der Tastenkombination Strg+V ein. Access fragt Sie dann nach einem Namen für das einzufügende Formular, vgl. Bild 15.6.

Bild 15.6: Geben Sie dem Formular, das Sie aus der Zwischenablage einfügen wollen, einen Namen.

Das Formular befindet sich in einer nicht geöffneten anderen Access-Datenbank. Dann können Sie es importieren. Gehen Sie dazu folgendermaßen vor:

- Wählen Sie aus dem Datenbankfenster heraus im Menü *Datei* den Befehl *Externe Daten*, Unterbefehl *Importieren...*
- Access läßt Sie mit dem Dialogfeld *Importieren* aus einem Ordner Ihrer Festplatte die Datenbank wählen, aus der Sie ein Formular importieren wollen. Wenn Sie die entsprechende Datenbank markiert und mit der Schaltfläche *Importieren* bestätigt haben, wird das Dialogfeld *Objekte importieren* angezeigt, vgl. Bild 15.7. Aktivieren Sie in diesem Dialogfeld ggf. die Registerkarte *Formulare*, wählen Sie das gewünschte Formular, und bestätigen Sie mit OK.

Bild 15.7: Dialogfeld »Objekte importieren« mit aktivierter Registerkarte »Formulare«

Nachdem Sie das Formular kopiert oder importiert haben, wird es zusammen mit den anderen Formularen in der Objektliste *Formulare* aufgelistet. Sie müssen es dann im allgemeinen mehr oder minder stark überarbeiten. Sie sollten vor allem prüfen, ob es dieselbe Datenherkunft behalten kann wie das Ausgangsformular. Andernfalls müssen Sie die Formulareigenschaft *Datenherkunft* entsprechend anders einstellen.

15.3 Formularansichten

Wechsel zwischen den Ansichten

Ein Formular kann in vier verschiedenen Ansichten betrachtet und in dreien davon bearbeitet werden:

Entwurfsansicht

Sie wechseln in die Entwurfsansicht, indem Sie die Dropdown-Liste der Symbol-Schaltfläche *Ansicht* aufschlagen und darin das Symbol *Entwurfsansicht* wählen, vgl. Bild 15.8. Sie können den gleichnamigen Befehl auch aus dem Menü *Ansicht* wählen. In der Entwurfsansicht erstellen und bear-

beiten Sie die verschiedenen Formularelemente, das Formular selbst eingeschlossen. Sämtliche Formulardefinitionen sind standardmäßig nur in der Entwurfsansicht bearbeitbar. Wenn Sie allerdings die Formular-Eigenschaft *Entwurfsänderungen zulassen* auf den Wert *Alle Ansichten* einstellen, können Sie das Eigenschaftenfenster auch in den anderen Ansichten einblenden und Eigenschaften von Formular und Steuerelementen einstellen.

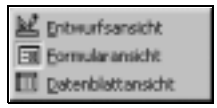


Bild 15.8: Aufgeschlagene Dropdown-Liste der Symbol-Schaltfläche »Ansicht«, wenn ein Formular aktiv ist. Im dargestellten Fall befindet sich das Formular in der Formularansicht, denn dieses Symbol erscheint gedrückt.

Formularansicht

Sie wechseln in die Formularansicht, indem Sie die Dropdown-Liste der Symbol-Schaltfläche *Ansicht* aufschlagen und darin das Symbol *Formularansicht* wählen, vgl. Bild 15.8. Sie können den gleichnamigen Befehl auch aus dem Menü *Ansicht* wählen. In der Formularansicht können Sie z.B. zu einzelnen Datensätzen blättern und deren Felder bearbeiten bzw. auf Befehlsschaltflächen klicken.

Datenblattansicht

Sie wechseln in die Datenblattansicht, indem Sie die Dropdown-Liste der Symbol-Schaltfläche *Ansicht* aufschlagen und darin das Symbol *Datenblattansicht* wählen, vgl. Bild 15.8. Sie können den gleichnamigen Befehl auch aus dem Menü *Ansicht* wählen. In der Datenblattansicht können Sie mehrere Datensätze in einer tabellarischen Darstellung gleichzeitig sehen und deren einzelne Felder bearbeiten. Dem Vorteil der gleichzeitigen Darstellung mehrerer Datensätze steht im allgemeinen – wenn das Formular für viele Tabellenfelder Steuerelemente enthält – der Nachteil gegenüber, daß Sie nicht alle Felder eines Datensatzes gleichzeitig einsehen können, so daß Sie in der Waagerechten rollen müssen, um an jedes Feld heranzukommen.

Seitenansicht

Sie wechseln in die Seitenansicht durch Klicken auf die Symbol-Schaltfläche *Seitenansicht* (vgl. links nebenstehend) oder durch Wahl des Befehls *Seitenansicht* aus dem Menü *Datei*. In der Seitenansicht prüfen Sie, wie das Formular aussehen wird, wenn es gedruckt wird. Durch Klicken auf die einzelnen Navigationssymbole können Sie zwischen den Seiten wechseln und entsprechend feststellen, wie viele Seiten der Ausdruck umfassen würde, wenn Sie alle Seiten drucken ließen. Ferner können Sie durch Klicken auf einen

Punkt der Seitenansicht zwischen gezoomter und nicht gezoomter Ansicht wechseln.

Standardansichten

Mit der Formulareigenschaft *Standardansicht* können Sie angeben, in welcher Ansicht ein Formular zunächst erscheint, wenn Sie es aus dem Datenbankfenster mit der Schaltfläche *Öffnen* (oder einem entsprechenden Makro bzw. einer VBA-Prozedur) öffnen. Dafür können Sie angeben, daß es als *Endlosformular*, *Einzelnes Formular* oder als *Datenblatt* geöffnet wird. Zum Verfahren, Formulareigenschaften einzustellen, vgl. weiter unten in diesem Kapitel, Punkt 15.7, *Eigenschaften von Steuerelementen, Formular und Bereichen einstellen*.

15.4 Das Formularfenster in der Entwurfsansicht

15.4.1 Arbeitsweise mit dem Entwurfsfenster

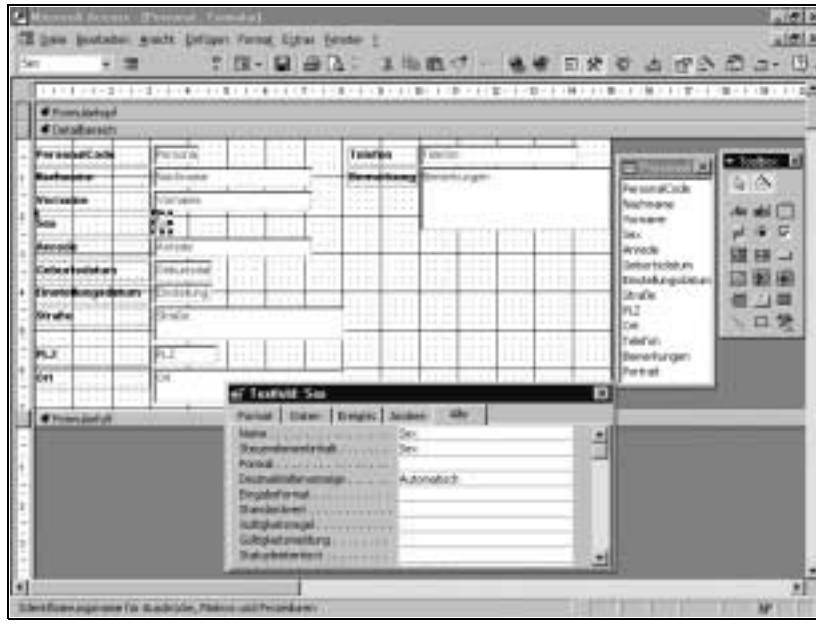


Bild 15.9: Das Formularfenster in der Entwurfsansicht mit verschiedenen Text- und Bezeichnungsfeldern im Detailbereich, von denen das Feld »Sex« gerade markiert ist.

Formularentwürfe werden in der Entwurfsansicht bearbeitet. Das Formularfenster in der Entwurfsansicht bietet zahlreiche Werkzeuge zum Bearbeiten der verschiedenen Elemente im Formular an. In Bild 15.9 sehen Sie die Wiedergabe eines Formularfensters in der Entwurfsansicht, bei dem die drei

Fenster *Eigenschaftenfenster*, *Feldliste* und *Toolbox*, die Sie ein- und ausblenden können, angezeigt werden. Im allgemeinen werden Sie nicht alle diese drei Hilfsmittel gleichzeitig benötigen. So unerläßlich jedes einzelne bei der Entwurfsarbeit ist, wirken sie doch oftmals störend, weil sie stets bestimmte Flächen des Formulars überdecken. Ständiges Hin und Her zwischen Ein- und Ausblenden dieser Werkzeuge gehört daher zur Arbeit am Formularentwurf.

Im Formularfenster in Bild 15.9 werden folgende vier Hilfsmittel angezeigt, die Sie normalerweise nicht alle gleichzeitig benötigen:

- *Eigenschaftenfenster* (trägt gerade die Überschrift *Textfeld: Sex*, weil das Steuerelement *Sex* markiert ist)
- *Feldliste* (trägt gerade die Überschrift *Personal*)
- *Toolbox*
- *Lineal* (linker und oberer Formularrand)

Jedes der Werkzeuge *Eigenschaftenfenster, Feldliste, Toolbox, Lineal* kann mit den entsprechenden Befehlen aus dem Menü *Ansicht* ein- und ausgeblendet werden. Bequemer noch lassen sich die folgenden drei dieser Werkzeuge durch Klicken auf die entsprechenden Symbol-Schaltflächen ein- oder ausblenden:

Symbol-Schaltflächen

Eigenschaften

Feldliste

Toolbox

Farb- und Formatpaletten dauerhaft einblenden

Die Symbolleiste *Formatierung (Formular/Bericht)* enthält verschiedene Symbole, mit denen Sie eine Palette aufschlagen können, aus deren einzelnen Symbolen (z.B. Farben oder Linienstärken) Sie dann eines zum Formatieren auswählen können. Diese Paletten können auch dauerhaft angezeigt werden, vgl. beispielsweise Bild 15.10, in dem die dauerhaft eingeblendete Palette für *Linien/Rahmenbreite* wiedergegeben ist. Gehen Sie folgendermaßen vor:

- Klicken Sie in der Formularentwurfsansicht je nach der gewünschten Palette auf den Pfeil neben der entsprechenden Schaltfläche in der Symbolleiste *Formatierung*, um die Palette anzuzeigen.
- Klicken Sie auf die Leiste am oberen Rand der Palette, und ziehen Sie die Palette von der Schaltfläche herunter.

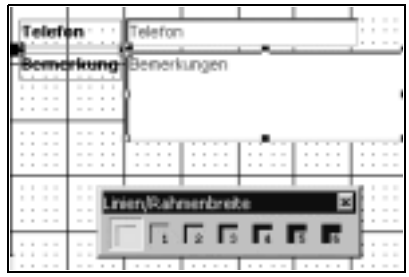


Bild 15.10: Dauerhaft eingeblendete Palette für »Linien/Rahmenbreite«

15.4.2 Eigenschaftenfenster

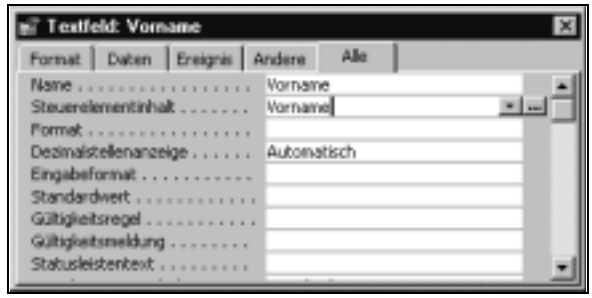


Bild 15.11: Eigenschaftenfenster. Aktiv ist gerade das Bearbeitungsfeld für die Eigenschaft »Steuerelementinhalt«. Beachten Sie die beiden Symbol-Schaltflächen am rechten Rand des Bearbeitungsfeldes zum Aufschlagen eines Dropdown-Listenfeldes (Symbol mit Pfeil) und zum Aufrufen eines Generators (Symbol mit drei Punkten).

Einblenden

Wählen Sie eine der folgenden Möglichkeiten:

- Klicken Sie auf die Symbol-Schaltfläche *Eigenschaften* in der Symbolleiste.
- Doppelklicken Sie auf eine beliebige Stelle des Formularentwurfs (z.B. Textfeld, Bereich usw.). Dadurch wird dieses Element markiert und gleichzeitig das Eigenschaftenfenster ggf. eingeblendet.
- Wählen Sie den Befehl *Eigenschaften* aus dem Menü *Ansicht*, der dann beim nächsten Aufschlagen des Menüs eingeschaltet erscheint.

Ausblenden

Wählen Sie eine der folgenden Möglichkeiten:

- Klicken Sie auf die Symbol-Schaltfläche *Schließen* am rechten Rand der Titelleiste des Eigenschaftenfensters.

- Klicken Sie auf die (gedrückte) Symbol-Schaltfläche *Eigenschaften*.
- Wählen Sie den (eingeschalteten) Befehl *Eigenschaften* aus dem Menü *Ansicht*.

Einsatzzweck des Eigenschaftenfensters

Mit Hilfe des Eigenschaftenfensters können Sie sich über die Eigenschaften eines Formularelements (Steuerelement, Formularbereich, das Formular selbst) informieren oder diese einstellen, d.h. entsprechend Ihren Wünschen verändern. Im linken, grauen Teil des Eigenschaftenfensters steht die Liste der Eigenschaften, die für das jeweils markierte Formularelement verfügbar sind: Je nach markiertem Element ist diese Liste mehr oder weniger lang, meistens mehr als weniger, so daß Sie diese mit der Bildlaufleiste am rechten Rand des Eigenschaftenfensters ggf. rollen müssen. Sie können allerdings durch Aktivieren einer der Registerkarten *Format*, *Daten*, *Ereignis* oder *Andere* die Liste der angezeigten Eigenschaften beschränken, so daß Sie sich besser orientieren können. Im rechten, hellen Teil des Eigenschaftenfensters befinden sich Bearbeitungsfelder. Diese dienen einerseits dazu, Ihnen die jeweiligen Einstellungen der einzelnen Eigenschaften anzuzeigen. Sie ermöglichen andererseits, durch Überschreiben oder Auswählen aus Dropdown-Listenfeldern die Einstellungen der einzelnen Eigenschaften zu verändern.

☞ Das Eigenschaftenfenster wird standardmäßig für ein Formular nur angezeigt, wenn sich dieses in der Entwurfsansicht befindet. Seit Access 2000 können Sie es jedoch auch in den anderen Ansichten (Formularansicht und Datenblattansicht) anzeigen lassen. Dann können Eigenschaftenwerte auch aus diesen Ansichten heraus geändert werden. Dies kann vorteilhaft sein, weil Sie die entsprechenden Effekte dann sofort in der Ansicht erkennen können, in der auch der Benutzer sie sehen wird. Damit das Eigenschaftenfenster für ein Formular auch in den anderen Ansichten angezeigt wird, gehen Sie folgendermaßen vor:

- Stellen Sie die Formulareigenschaft *Entwurfsänderungen zulassen* auf den Wert *Alle Ansichten* ein. Standardmäßig ist diese Eigenschaft auf den Wert *Nur Entwurfsansicht* eingestellt.

Fenstergröße und -position verändern

Wie bei allen Fenstern mit einer Titelleiste können Sie auch für das Eigenschaftenfenster die Größe und Position verändern: Ziehen Sie einen Rand oder Eckpunkt mit der Maus in die gewünschte Richtung, um die Größe zu ändern. Ziehen Sie die Titelleiste, um das Fenster zu verschieben.

Für ein Bearbeitungsfeld das Dialogfeld *Zoom öffnen*

Mit der Tastenkombination ⇧+F2 läßt sich für viele Bearbeitungsfelder des Eigenschaftenfensters das Dialogfeld *Zoom* einblenden, mit dem Sie längere Zeichenfolgen (z.B. eine längere Gültigkeitsmeldung für ein Textfeld,

eine SQL-SELECT-Anweisung für die Datenherkunft oder einen umfangreichen Ausdruck als Steuerelementinhalt) bequemer eingeben oder einsehen können, vgl. das geöffnete Dialogfeld *Zoom* in Bild 15.12.

Bild 15.12: Dialogfeld »Zoom«, hier mit einer Zeichenfolge einer SQL-Select-Anweisung

Neu ist in Access 2000, daß Sie im Dialogfeld *Zoom* die Schriftart einstellen können. Die voreingestellte Schriftgröße 8 beispielsweise mag vielen Bearbeitern zu klein erscheinen, so daß sie auf einen größeren Schriftgrad eingestellt werden kann.

Für ein Bearbeitungsfeld den Generator aufrufen

Für viele Bearbeitungsfelder läßt sich ein *Generator* aufrufen. Wenn ein Generator zur Verfügung steht, wird die entsprechende Symbol-Schaltfläche mit den drei Punkten (vgl. Bild 15.13) für die aktuelle Eigenschaft eingeblendet. Je nach Art der aktiven Eigenschaft meldet sich nach dem Klicken auf die Symbol-Schaltfläche *Generator* entweder sofort der *Ausdrucks-Generator* (zum Erstellen eines Ausdrucks), oder, wenn im Prinzip mehrere Generatoren in Frage kommen (z.B. bei Ereigniseigenschaften), ein Dialogfeld, in dem Sie den *Ausdrucks-*, *Makro-* oder *Code-Generator* wählen können.

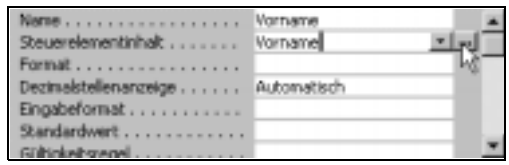


Bild 15.13: Für die Eigenschaft »Steuerelementinhalt« wird ein Generator angeboten.

15.4.3 Feldliste

Bild 15.14: Feldliste

Einblenden

Die Feldliste läßt sich nur anzeigen, wenn für das Formular eine Datenherkunft (Tabelle, Abfrage oder SQL-SELECT-Anweisung) festgelegt ist, weil sonst keine Felder zur Verfügung stehen können. Wählen Sie eine der folgenden Möglichkeiten:

- Klicken auf die Symbol-Schaltfläche *Feldliste*.
- Wählen Sie den Befehl *Feldliste* aus dem Menü *Ansicht*.

Ausblenden

Wählen Sie eine der folgenden Möglichkeiten:

- Klicken Sie auf die Symbol-Schaltfläche *Schließen* am rechten Rand der Titelleiste des Feldlisten-Fensters.
- Klicken auf die (gedrückte) Symbol-Schaltfläche *Feldliste*.
- Wählen Sie den (eingeschalteten) Befehl *Feldliste* aus dem Menü *Ansicht*.

Einsatzzweck der Feldliste

Mittels der Feldliste können Sie, ähnlich wie in der Entwurfsansicht von Abfragen, dem Formular Steuerelemente für die entsprechenden Felder der dem Formular zugrundeliegenden Tabelle/Abfrage hinzufügen: Ziehen Sie ein Feld oder mehrere Felder aus der Feldliste an die Stelle im Formular, an der das entsprechende Steuerelement stehen soll. Mehrere bzw. alle Felder markieren Sie mit gedrückter Strg-Taste (nicht angrenzende Felder) bzw. ⇧-Taste (angrenzende Felder) bzw. durch Doppelklicken auf die Titelleiste der Feldliste (alle Felder).

Steuerelemente, die mittels der Feldliste eingefügt wurden, haben den Vorteil, die Einstellungen der Feldeigenschaften aus der zugrundeliegenden Tabelle automatisch zu übernehmen. Wenn Sie ein Textfeld dagegen mit der Toolbox hinzufügen, bekommt es die gerade gültigen Standardeigenschaften zugewiesen.

15.4.4 Toolbox

Bild 15.15: Toolbox

Einblenden

Wählen Sie eine der folgenden Möglichkeiten:

- Klicken Sie auf die Symbol-Schaltfläche *Toolbox*.
- Wählen Sie den Befehl *Toolbox* aus dem Menü *Ansicht*.

Am Fensterrand verankern

Die Toolbox läßt sich wie eine Symbolleiste an jedem der vier Fensterränder verankern: Ziehen Sie die Titelleiste der Toolbox auf den oberen, unteren, linken oder rechten Rand des Access-Anwendungsfensters, um sie dort zu verankern. Entsprechend läßt sich die zur Symbolleiste gewordene Toolbox durch Ziehen in das Anwendungsfenster wieder zu einem Fenster machen.

Form des Toolbox-Fensters verändern

In Bild 15.15 sehen Sie die Toolbox als Fenster, in dem drei Steuerelement-Symbole in eine Reihe passen. Sie können die Form des Toolbox-Fensters durch Ziehen eines seiner Ränder verändern und es dadurch schmaler und länger oder breiter und kürzer machen.

Ausblenden

Wählen Sie eine der folgenden Möglichkeiten:

- Klicken Sie auf die Symbol-Schaltfläche *Schließen* am rechten Rand der Titelleiste der Toolbox.
- Klicken Sie auf die (gedrückte) Symbol-Schaltfläche *Toolbox*.
- Wählen Sie den (eingeschalteten) Befehl *Toolbox* aus dem Menü *Ansicht*.

Einsatzzweck der Toolbox

Die Toolbox ist ein Werkzeug, mit dem Sie dem Formular neue Steuerelemente hinzufügen: Klicken Sie auf eines der Steuerelement-Symbole in der Toolbox, und ziehen Sie dann mit dem Mauszeiger im Formular ein Rechteck an der gewünschten Position und in gewünschter Größe, um ein bestimmtes Steuerelement einzufügen. Das Steuerelement bekommt dann für seine Eigenschaften die Einstellungen der Standardeigenschaften zugewiesen.

Ein zweiter Einsatzzweck der Toolbox liegt darin, die Änderung der Standardeigenschaften für die verschiedenen Steuerelemente und das Formular zu ermöglichen, vgl. dazu im einzelnen weiter unten in diesem Kapitel, Punkt 15.7.2, *Standardeigenschaften von Steuerelementen* .

15.4.5 Lineal

Bild 15.16: Waagerechtes Lineal

Einblenden

- Wählen Sie den Befehl *Lineal* aus dem Menü *Ansicht*.

Ausblenden

- Wählen Sie den (eingeschalteten) Befehl *Lineal* aus dem Menü *Ansicht*.

Einsatzzweck des Lineals

Bild 15.17: Die Lineale zeigen an, auf welche Größe das Steuerelement »Sex« vergrößert wird, wenn die Maustaste in dieser Situation losgelassen wird.

Wenn die beiden Linealleisten eingeblendet sind, können Sie Formularelemente millimetergenau positionieren: Während Sie in der Entwurfsansicht mit der Maus ein Steuerelement verschieben oder vergrößern oder eine Markierung

vornehmen, wird die waagerechte und senkrechte Position des gezogenen Rechtecks durch gestrichelte Linien auf dem Formular und durch abgedunkelte Bereiche in den beiden Linealleisten angezeigt, vgl. Bild 15.17. Diese Linien und Linealbereiche werden mitgeführt, wenn Sie die Maus ziehen.

15.4.6 Raster

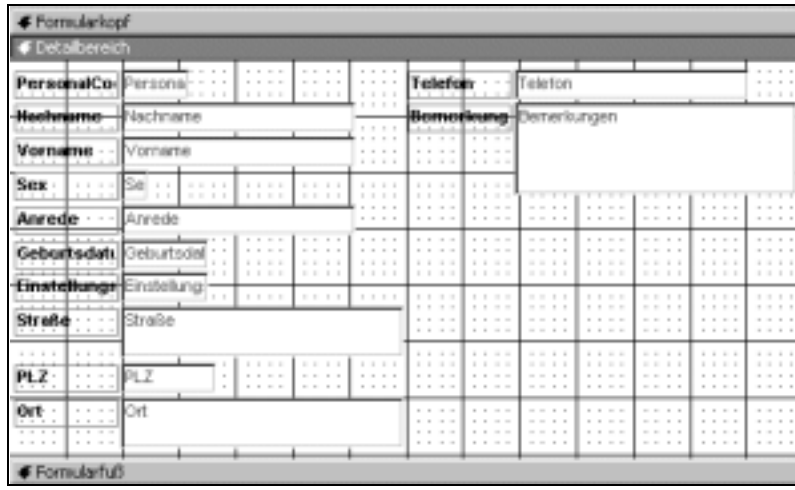


Bild 15.18: Rasterpunkte werden angezeigt.

Einblenden

- Wählen Sie den Befehl *Raster* aus dem Menü *Ansicht*. Rasterpunkte werden allerdings nur angezeigt, wenn Sie nicht mehr als sechs Rasterpunkte je cm in der Waagerechten und Senkrechten festgelegt haben. Den Abstand von Rasterpunkten stellen Sie mit den Formulareigenschaften *Raster X* und *Raster Y* im Eigenschaftenfenster ein.

Ausblenden

- Wählen Sie den (eingeschalteten) Befehl *Raster* aus dem Menü *Ansicht*.

Einsatzzweck des Rasters

Das Einblenden des Rasters ermöglicht eine optische Orientierung beim Positionieren von Steuerelementen im Formular. Ob das Raster ein- oder ausgeblendet ist, hat keinerlei Wirkung auf die verschiedenen *Raster*-Befehle des Menüs *Format*; vgl. dazu genauer den Punkt 15.6.8, *Steuerelemente am Raster ausrichten*. Mit anderen Worten: Wenn Sie Rasterpunkte sehen, muß dies nicht bedeuten, daß Steuerelemente auch automatisch am Raster ausgerichtet werden. Für diesen Zweck muß vielmehr der Befehl *Am Raster ausrichten* aus dem Menü *Format* eingeschaltet sein.

15.5 Steuerelemente: Einführung und Überblick

Als Steuerelemente (englisch: *Control*) werden die verschiedenen grafischen Objekte bezeichnet, die Sie in ein Formular einfügen können, um Daten zu bearbeiten (z.B. *Textfeld*, *Kontrollkästchen*, *Listenfeld*), Aktionen auszulösen (z.B. *Befehlsschaltfläche*) oder auch nur visuell zu gestalten (z.B. *Rechteck*). Sie können über 17 Standard-Steuerelemente und über zahlreiche Zusatz-Steuerelemente verfügen. Die Zusatz-Steuerelemente werden als *ActiveX*-Steuerelemente bezeichnet. In diesem Kapitel wird nur auf die Standard-Steuerelemente eingegangen. Der Umgang mit den ActiveX-Steuerelementen wird in Kap. 17, *ActiveX-Steuerelemente in Formularen*, erklärt.

15.5.1 Gebundene, ungebundene und berechnete Steuerelemente

Gebundene Steuerelemente

Ein Steuerelement wird als gebunden bezeichnet, wenn es an ein Feld einer Tabelle oder Abfrage gebunden ist. In diesem Falle gibt es den Feldinhalt des jeweiligen Datensatzes wieder und ermöglicht im allgemeinen Bearbeitungen daran, die an die Tabelle/Abfrage weitergegeben werden. Bei einem gebundenen Steuerelement ist die Eigenschaft *Steuerelementinhalt* auf den Feldnamen einer Tabelle/Abfrage eingestellt. Ein gebundenes Textfeld (mit Bezeichnungsfeld) beispielsweise erscheint in der Entwurfsansicht des Formulars, wie in Bild 15.19 dargestellt: Der Name des Feldes, an das das Steuerelement gebunden ist, wird im Steuerelement angezeigt.

Bild 15.19: Gebundene Steuerelemente

Ungebundene Steuerelemente

Ein ungebundenes Steuerelement ist weder an ein Feld einer Tabelle/Abfrage gebunden, noch enthält es einen berechnenden Ausdruck. Seine Eigenschaft *Steuerelementinhalt* (sofern sie überhaupt verfügbar ist) enthält keinen Eintrag. Einige Steuerelemente (z.B. Bezeichnungsfeld, Rechteck, Linie) können nur als ungebundene fungieren; dann ist die Eigenschaft *Steuerelementinhalt* nicht verfügbar. Ungebundene Steuerelemente, für die die Eigenschaft *Steuerelementinhalt* verfügbar ist, werden oft dazu verwendet, Werteingaben zu ermöglichen, die mit Hilfe eines Ausdrucks, eines Makros oder einer VBA-Prozedur weiterverarbeitet werden. Ein ungebundenes Textfeld (mit Bezeichnungsfeld) erscheint in der Entwurfsansicht des Formulars, wie in Bild 15.20 zu sehen.

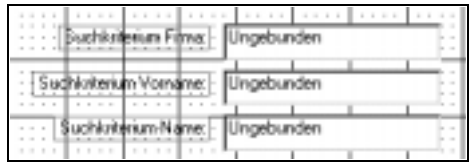


Bild 15.20: Ungebundene Steuerelemente

Berechnete Steuerelemente

Wenn die Eigenschaft *Steuerelementinhalt* einen Ausdruck enthält, der mit einem Gleichheitszeichen (=) beginnt und einen Wert ergibt, liegt ein berechnetes Steuerelement vor. Beispielsweise stellen die Textfelder in Bild 15.21, die Datumswerte wiedergeben, berechnete Steuerelemente dar.

Bild 15.21: Berechnete Steuerelemente

15.5.2 Steuerelemente einfügen

Entsprechend der Bedeutung, die das Einfügen von Steuerelementen in ein Formular für die praktische Arbeit besitzt, stellt Access dafür mehrere Möglichkeiten zur Verfügung, die im folgenden kurz mit ihren jeweiligen Vor- und Nachteilen beschrieben werden.

Bezeichnungsfeld automatisch oder nachträglich einfügen

Nach der Voreinstellung erhalten die meisten Steuerelemente, wenn sie einem Formular hinzugefügt werden, ein zugeordnetes Bezeichnungsfeld. Dies liegt daran, daß die Standardeigenschaft *Mit Bezeichnungsfeld* dieser Steuerelemente auf den Wert *Ja* eingestellt ist. Sie können diese Standardeigenschaft für jedes dafür in Frage kommende Steuerelement auf *Nein* einstellen; vgl. zum diesbezüglichen Verfahren in diesem Kapitel Punkt 15.7.2, *Standardeigenschaften von Steuerelementen* .

Steuerelemente, die mit AutoFormular oder durch den Formular-Assistenten eingefügt wurden

Wenn Sie ein AutoFormular oder ein Formular mit dem Formular-Assistenten erstellen lassen, werden für die Felder, die Sie zur Aufnahme in das Formular angegeben haben, automatisch geeignete Steuerelemente erstellt.

Vorteile

Die Art des eingefügten Steuerelements entspricht dem Datentyp des zugrunde liegenden Feldes der Tabelle/Abfrage. So erstellt der Formular-Assistent beispielsweise automatisch ein Kontrollkästchen für ein Feld mit dem Datentyp *Ja/Nein* oder ein Textfeld für die Datentypen *Text*, *Zahl*, *Datum/Zeit*, *Währung* und *Zähler*. Die Feldeigenschaften einer Tabelle werden auf die entsprechenden Eigenschaften der Formular-Steuerelemente übertragen.

Nachteile

In einem AutoFormular oder einem vom Formular-Assistenten erzeugten Formular werden die Steuerelemente selten so angeordnet, daß sie den Benutzerbedürfnissen entsprechen. Daher müssen sie im allgemeinen nachträglich neu positioniert werden. Ungebundene und berechnete Steuerelemente können überhaupt nicht automatisch erstellt werden, auch nicht vom Formular-Assistenten.

Steuerelemente durch Ziehen aus der Feldliste einfügen

Um Steuerelemente durch Ziehen aus der Feldliste einzufügen, gehen Sie folgendermaßen vor:

- Blenden Sie die Feldliste ggf. ein, indem Sie z.B. auf die Symbol-Schaltfläche *Feldliste* klicken.
- Ziehen Sie ein Feld oder mehrere Felder aus der Feldliste an die Stelle im Formular, an der das entsprechende Steuerelement stehen soll. Mehrere bzw. alle Felder markieren Sie mit gedrückter [Strg]-Taste bzw. [⇧]-Taste bzw. durch Doppelklicken auf die Titelleiste der Feldliste.

Vorteile

Die Einstellungen der Feldeigenschaften aus der zugrundeliegenden Tabelle werden automatisch übernommen. Gegenüber dem Formular-Assistenten haben Sie hier die Möglichkeit, das Steuerelement zu positionieren, indem Sie die Maustaste an geeigneter Stelle loslassen.

Nachteile

Mittels der Feldliste können Sie nur gebundene Steuerelemente erstellen.

Beim Ziehen von Feldern aus der Feldliste erstellt Access standardmäßig Textfelder. Sie können jedoch einen anderen Steuerelementtyp bestimmen, wenn Sie unmittelbar vor dem Ziehen einen entsprechenden anderen Steuerelementtyp in der Toolbox auswählen.

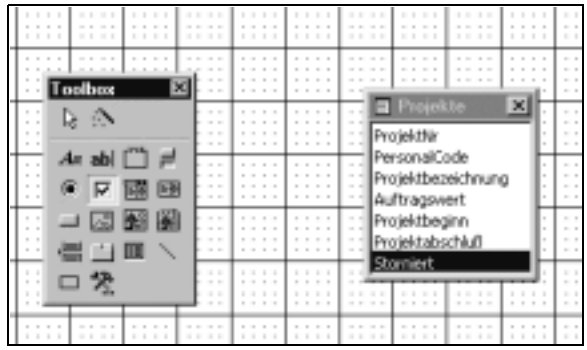


Bild 15.22: Wenn Sie in dieser Situation ein Feld aus der Feldliste in das Formular ziehen, wird es als Kontrollkästchen eingefügt, weil in der Toolbox die Symbol-Schaltfläche »Kontrollkästchen« gewählt ist (sie erscheint daher gedrückt).

Steuerelemente mit der Toolbox einfügen

Um ein Steuerelement mit der Toolbox einzufügen, gehen Sie folgendermaßen vor:

- Blenden Sie die Toolbox ggf. ein, indem Sie auf die Symbol-Schaltfläche *Toolbox* klicken oder den Befehl *Toolbox* aus dem Menü *Ansicht* wählen.
- Klicken Sie auf die Symbol-Schaltfläche in der Toolbox, die dem zu erstellenden Steuerelementtyp entspricht.
- Ziehen Sie mit dem Mauszeiger, der jetzt das Aussehen eines Fadenkreuzes mit zugeordnetem Steuerelementsymbol besitzt, ein Rechteck an der Stelle, an der das einzufügende Steuerelement positioniert werden soll. Die Größe des gezogenen Rechtecks bestimmt die Größe des Steuerelements.
- *Alternativ*
  Um ein gebundenes Steuerelement zu erstellen, ziehen Sie ein Feld oder mehrere Felder aus der Feldliste.

Mehrere Steuerelemente gleichen Typs nacheinander erstellen

Standardmäßig nimmt der Mauszeiger nach dem Einfügen eines Steuerelements mit der Toolbox wieder das Aussehen und die Funktion des reinen Zeigers (Pfeil in Nord-West-Richtung) an, mit dem Sie beispielsweise markieren können. Um mehrere Steuerelemente gleichen Typs nacheinander einzufügen, gehen Sie folgendermaßen vor:

- Doppelklicken Sie auf das Tool, das Sie aktiv halten möchten.
- Erstellen Sie Steuerelemente durch Ziehen aus der Feldliste oder durch Ziehen eines Rechtecks im Formular.
- Drücken Sie [Esc], um die Daueraktivierung des betreffenden Tools aufzuheben.

Vorteile

Bei Verwendung der Toolbox können Sie den Typ, die Position und Größe von Steuerelementen unmittelbar beim Erstellen bestimmen.

Nachteile

Die Feldeigenschaften aus der Tabelle werden nur dann auf das Steuerelement übertragen, wenn Sie aus der Feldliste ziehen. Steuerelemente, die durch Ziehen eines Rechtecks im Formular erstellt werden, sind zunächst immer ungebunden, so daß auf sie keine Eigenschaften von Tabellenfeldern übertragen werden können.

Unterstützung vom Steuerelement-Assistenten anfordern

Sie können sich beim Einfügen einiger bestimmter Steuerelementarten vom Steuerelement-Assistenten unterstützen lassen. Dazu muß die Symbol-Schaltfläche *Steuerelement-Assistenten* (vgl. links nebenstehend) in der Toolbox gedrückt sein.

Wenn die Symbol-Schaltfläche *Steuerelement-Assistenten* in der Toolbox gedrückt ist, wird der Steuerelement-Assistent automatisch aufgerufen, nachdem Sie eines der folgenden Steuerelemente eingefügt haben:

- Optionsgruppe
- Kombinationsfeld
- Listenfeld
- Diagramm
- Objektfeld
- Befehlsschaltfläche

Der Steuerelement-Assistent stellt Ihnen dann mit verschiedenen, dem jeweiligen Steuerelementtyp angepaßten Dialogfeldern Fragen, deren Antworten bestimmte wichtige Eigenschaften des Steuerelements betreffen.

Besonders nützlich ist der Steuerelement-Assistent beispielsweise beim Einfügen eines Listenfeldes, weil dieses nur dann zufriedenstellend funktioniert, wenn mehrere seiner Eigenschaften (z.B. Datensatzherkunft, gebundene Spalte usw.) vernünftig eingestellt sind. In Bild 15.23 beispielsweise wird das erste Dialogfeld des Listenfeld-Assistenten gezeigt. Wenn Sie sich dafür entscheiden, daß das »Listenfeld die Werte einer Tabelle oder Abfrage entnehmen« soll, werden Sie mit weiteren Dialogfeldern nach dem Namen der Tabelle/Abfrage, den daraus für das Listenfeld auszuwählenden Feldern, den gewünschten Spaltenbreiten (vgl. Bild 15.24) usw. gefragt. Gegenüber der Alternative, die Eigenschaften für ein Listenfeld direkt im Eigenschaftenfenster einzustellen, ist die Unterstützung durch den Steuerelement-Assisten-

ten wesentlich komfortabler und vor allem sicherer bezüglich der Wirkung. Sehr empfehlenswert ist dessen Einsatz auch für Kombinationsfelder, Optionsgruppen, Diagramme und manchmal auch Befehlsschaltflächen.

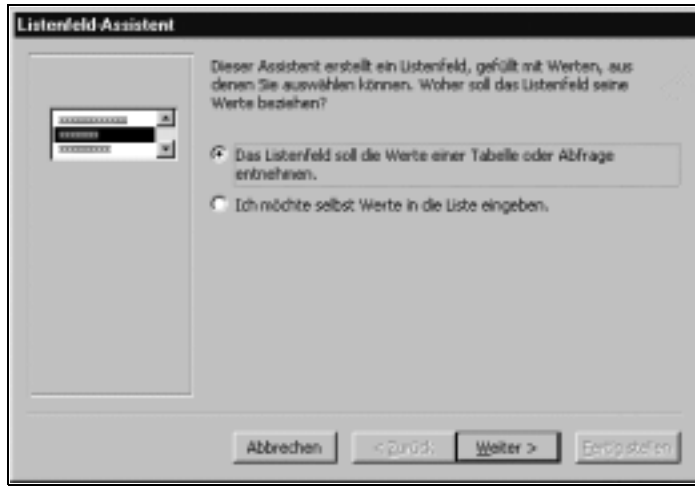


Bild 15.23: Erstes Dialogfeld des Steuerelement-Assistenten zum Einfügen eines Listenfeldes

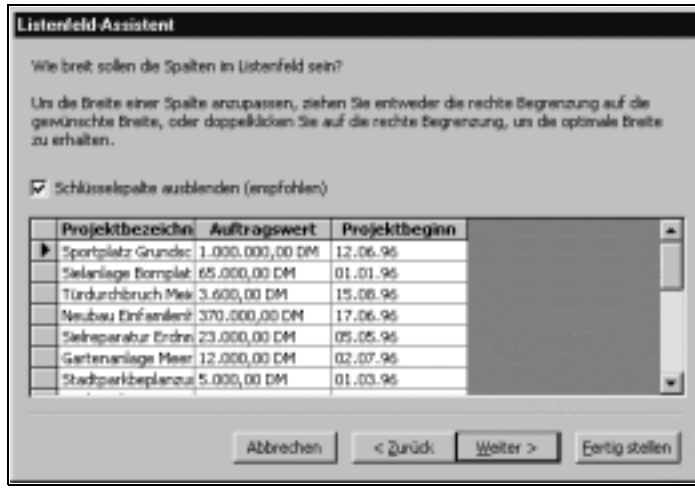


Bild 15.24: Viertes Dialogfeld des Steuerelement-Assistenten zum Einfügen eines Listenfeldes

Sie schalten den Steuerelement-Assistenten aus, indem Sie erneut auf die (gedrückte) Symbol-Schaltfläche *Steuerelement-Assistenten* in der Toolbox klicken, so daß das Symbol nicht mehr gedrückt erscheint.

15.5.3 Typ des Steuerelements ändern

Sie können ein Steuerelement eines bestimmten Typs in einen anderen Typ ändern. Beispielsweise können Sie ein Textfeld in ein Bezeichnung-, Listen- oder Kombinationsfeld ändern, und auch das Umgekehrte ist möglich. Um beispielsweise ein Textfeld in ein Kombinationsfeld zu ändern, verfahren Sie wie folgt:

- Markieren Sie das Textfeld.
- Wählen Sie im Menü *Format* den Befehl *Ändern zu*. Dann wird eine Liste mit Steuerelementen angezeigt. Wählen Sie daraus das Steuerelement *Kombinationsfeld*.
- Bearbeiten Sie die Eigenschaften-Einstellungen des neuen Steuerelementtyps. Beispielsweise müssen Sie für ein Kombinationsfeld die Datensatzherkunft oder eine Werteliste angeben, damit seine Werteliste nicht leer ist. Der Steuerelementinhalt bleibt im Falle gebundener Steuerelemente dagegen bei der Umwandlung erhalten.

Es ist allerdings keinesfalls möglich, jeden Steuerelementtyp in jeden anderen umzuwandeln. Vielmehr gibt es dafür nur beschränkte Möglichkeiten. Die verfügbaren Möglichkeiten für das jeweils markierte Steuerelement ergeben sich aus der Liste des Befehls *Ändern zu*, in der die nicht verfügbaren Steuerelemente deaktiviert sind.

15.5.4 Steuerelemente kopieren

Allgemeine Vorgehensweise

Steuerelemente lassen sich kopieren – innerhalb desselben Formulars oder zwischen verschiedenen Formularen. Die Kopie eines Steuerelements stimmt in allen Merkmalen (Typ, Eigenschaften) mit dem Original überein, soweit dies nicht aus sachlichen Gründen ausgeschlossen sein muß; beispielsweise bekommt die Kopie einen anderen Namen als das Original, sofern Sie diese im selben Formular einfügen, aus dem kopiert wurde, weil Namen von Steuerelementen im selben Formular singulär sein müssen. Um eine Kopie von einem oder mehreren Steuerelementen zu erstellen, gehen Sie folgendermaßen vor:

- Markieren Sie das oder die zu kopierende(n) Steuerelement(e).
- Drücken Sie die Tastenkombination [Strg]+[C], oder wählen Sie den Befehl *Kopieren* aus dem Menü *Bearbeiten*.
- Markieren Sie ggf. die Stelle, an der die Kopie eingefügt werden soll. Wenn Sie ein Steuerelement markieren, wird die Kopie etwas unterhalb davon eingefügt. Wenn Sie einen Formularbereich markieren, indem Sie auf einen freien Punkt in dem Bereich klicken, wird die Kopie am oberen Rand des Bereichs eingefügt.

- Drücken Sie zum Einfügen der Kopie die Tastenkombination Strg + V, oder wählen Sie den Befehl *Einfügen* aus dem Menü *Bearbeiten*.

Befehl *Duplizieren*

Innerhalb desselben Formulars können Sie ein oder mehrere Steuerelemente mit Hilfe des Befehls *Duplizieren* auf die schnellste und bequemste Weise kopieren. Gehen Sie folgendermaßen vor:

- Markieren Sie das oder die zu duplizierenden Steuerelement(e).
- Wählen Sie im Menü *Bearbeiten* den Befehl *Duplizieren*. Dann erzeugt Access unterhalb des markierten Steuerelements ein Duplikat davon, vgl. Bild 15.25. Im Falle mehrerer Steuerelemente werden auch entsprechend viele Duplikate erzeugt.

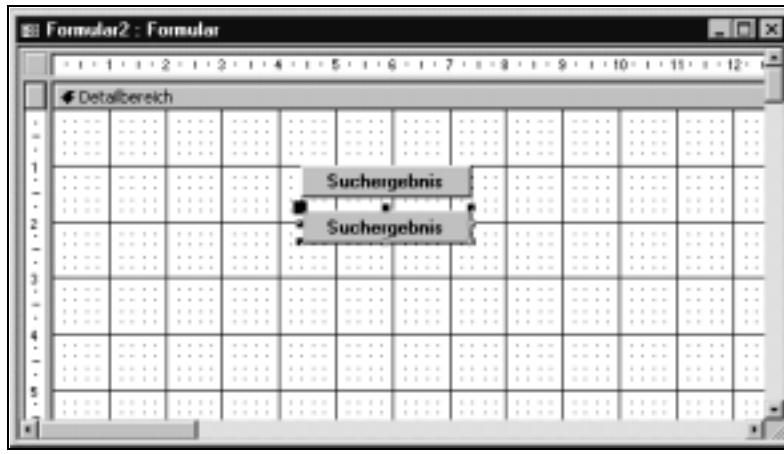


Bild 15.25: Die untere Befehlsschaltfläche wurde mit dem Befehl »Duplizieren« des Menüs »Bearbeiten« aus der oberen erzeugt.

15.6 Steuerelemente markieren und bearbeiten

In diesem Punkt wird gezeigt, wie Sie Steuerelemente *mit der Maus* markieren und bearbeiten. Jede Bearbeitung (Ändern der Position, Größe, Farbe usw.) können Sie statt mit der Maus auch durch Einstellen der entsprechenden Eigenschaften im Eigenschaftenfenster vornehmen. Die Bearbeitung mit Hilfe des Eigenschaftenfensters ist zwar im allgemeinen umständlicher als die mit der Maus, kann aber in einzelnen Fällen zu präziseren Ergebnissen führen.

15.6.1 Markieren von Steuerelementen

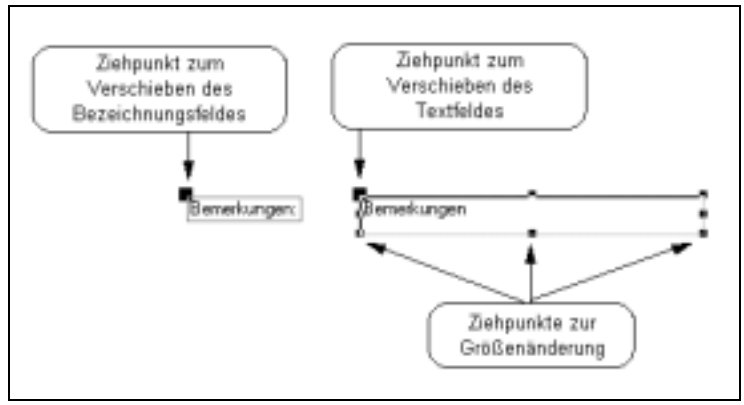


Bild 15.26: Ziehpunkte zum Verändern von Position und Größe eines zusammengesetzten Steuerelements, dargestellt am Beispiel eines Textfeldes mit Bezeichnungsfeld. Textfeld und Bezeichnungsfeld werden zusammen verschoben, indem Sie den Mauszeiger auf einen beliebigen Punkt eines Textfeldrandes (jedoch nicht auf einen Ziehpunkt) fahren, so daß er wie eine schwarze Hand aussieht.

Einzelnes Steuerelement markieren

- Klicken Sie auf einen beliebigen Punkt des Steuerelements. Bei einem zusammengesetzten Steuerelement, wie z.B. einem Textfeld mit zugeordnetem Bezeichnungsfeld, klicken Sie auf einen Punkt des eigentlichen Steuerelements (nicht auf einen Punkt des Bezeichnungsfeldes). Das markierte Steuerelement sieht dann ähnlich aus wie das in Bild 15.26 wiedergegebene.

Alternativ

- Schlagen Sie die Dropdown-Liste *Objekt* auf, und wählen Sie darin das zu markierende Objekt, also beispielsweise ein Steuerelement. Die Dropdown-Liste *Objekt* befindet sich standardmäßig am linken Rand der Symbolleiste *Formatierung (Formular/Bericht)*.

Markierung eines oder mehrerer Steuerelemente aufheben

Um die Markierung eines oder mehrerer Steuerelemente aufzufheben, klikken Sie auf einen beliebigen Punkt im Formular.

Zusammenliegende Steuerelemente markieren

- Ziehen Sie ein Rechteck um alle Steuerelemente, die Sie markieren möchten. Beginnen Sie dabei an einem beliebigen Punkt außerhalb eines der zu markierenden Steuerelemente.

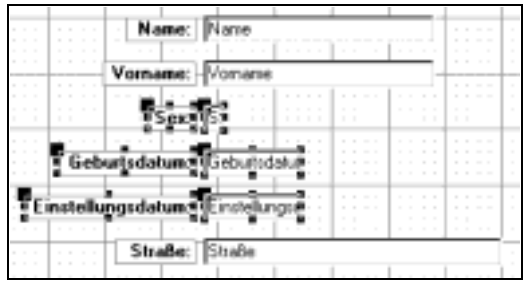


Bild 15.27: Zusammenliegende Steuerelemente wurden durch Ziehen eines Rechtecks mit dem Mauszeiger markiert.

Mehrere nicht zusammenliegende Steuerelemente markieren

- Klicken Sie auf jedes Steuerelement, während Sie die ⇧-Taste gedrückt halten.

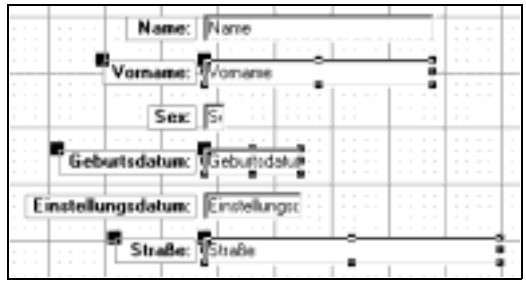


Bild 15.28: Nicht zusammenliegende Steuerelemente wurden mit gedrückter ⇧-Taste markiert.

Markierung einzelner Steuerelemente aus einer Gruppe markierter Steuerelemente aufheben

Sie können die Markierung einzelner Steuerelemente aus einer Gruppe markierter Steuerelemente auf die folgende Weise aufheben:

- Klicken Sie auf das Steuerelement, dessen Markierung Sie zurücksetzen wollen, während Sie die ⇧-Taste gedrückt halten.

Alle Steuerelemente markieren

- Wählen Sie im Menü *Bearbeiten* den Befehl *Alles markieren*, den Sie auch mit Strg+A aufrufen können.

Wenn Sie viele nicht zusammenliegende Steuerelemente markieren wollen, kann es effektiv sein, zunächst durch Ziehen eines Rechtecks eine Gruppe zusammenliegender Steuerelemente zu markieren und dann die Markierung für einzelne daraus aufzuheben.

15.6.2 Gruppieren von Steuerelementen

Sie können mehrere Steuerelemente zu einer Gruppe zusammenfassen. Dann lassen sich diese gruppierten Steuerelemente gemeinsam bearbeiten, beispielsweise verschieben oder in einem Eigenschaftswert einstellen. Solange mehrere Steuerelemente gruppiert sind, können Sie sie nicht einzeln markieren und bearbeiten. Dazu müßten Sie die Gruppierung – ggf. nur vorübergehend – wieder aufheben.

Steuerelemente gruppieren

Gehen Sie folgendermaßen vor, um mehrere Steuerelemente (mindestens zwei) zu gruppieren:

- Markieren Sie die zu gruppierenden Steuerelemente.
- Wählen Sie im Menü *Format* den Befehl *Gruppierung*.

Um eine Gruppe von Steuerelementen zu markieren, klicken Sie auf ein beliebiges Steuerelement darin. Dann wird die Gruppierung durch einen Rahmen kenntlich gemacht, vgl. Bild 15.29.

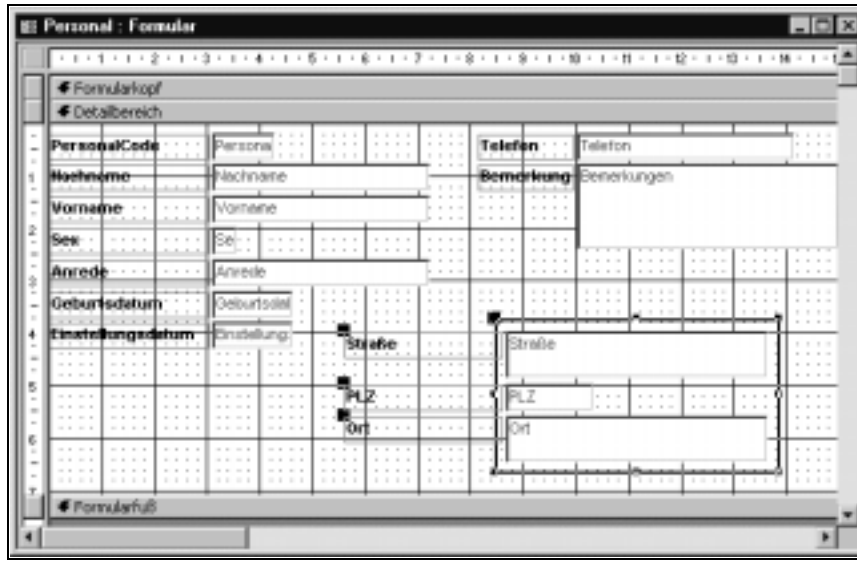


Bild 15.29: Die Gruppierung mit den drei Steuerelementen »Straße«, »PLZ« und »Ort« wurde markiert.

Gruppierung aufheben

Um eine Gruppierung von Steuerelementen aufzuheben, verfahren Sie wie folgt:

- Markieren Sie die Elemente der Gruppe, indem Sie auf eines ihrer Elemente klicken.
- Wählen Sie aus dem Menü *Format* den Befehl *Gruppierung aufheben*.

15.6.3 Löschen von Steuerelementen

Um ein oder mehrere Steuerelemente zu löschen, verfahren Sie wie folgt:

- Markieren Sie das oder die Steuerelemente.
- Drücken Sie die [Entf]-Taste, oder wählen Sie den Befehl *Löschen* aus dem Menü *Bearbeiten*.

15.6.4 Verschieben von Steuerelementen

Einzelnes Steuerelement verschieben

- Markieren Sie das Steuerelement.
- Fahren Sie den Mauszeiger auf einen beliebigen Punkt eines Steuerelementrandes (jedoch nicht auf einen Ziehpunkt), so daß er wie eine offene schwarze Hand aussieht. Ziehen Sie dann das Steuerelement an die gewünschte Position.

Steuerelement ohne sein zugeordnetes Bezeichnungsfeld verschieben.

- Markieren Sie das Steuerelement.
- Fahren Sie den Mauszeiger auf den Ziehpunkt zum Verschieben des eigentlichen Steuerelements, vgl. Bild 15.26. Er sieht dann aus wie eine schwarze Hand mit abgespreiztem Daumen und Zeigefinger. Ziehen Sie an die gewünschte Position.

Bezeichnungsfeld ohne sein zugeordnetes Steuerelement verschieben.

- Markieren Sie das Steuerelement.
- Fahren Sie den Mauszeiger auf den Ziehpunkt zum Verschieben des Bezeichnungsfeldes, vgl. Bild 15.26. Er sieht dann aus wie eine schwarze Hand mit abgespreiztem Daumen und Zeigefinger. Ziehen Sie an die gewünschte Position.

Mehrere Steuerelemente verschieben

- Markieren Sie die Steuerelemente (zusammen- oder auseinanderliegende).
- Bewegen Sie den Mauszeiger so über die markierten Steuerelemente, daß er die Form einer offenen Hand annimmt. Ziehen Sie dann die ganze Gruppe von Steuerelementen an die gewünschte Position.

Steuerelemente nur horizontal oder nur vertikal verschieben

Wenn Sie die [Umschalt]-Taste gedrückt halten, während Sie ein Steuerelement verschieben, können Sie dessen horizontale oder vertikale Ausrichtung beibehalten: Das Steuerelement läßt sich nur horizontal oder nur vertikal verschieben, je nachdem, in welche Richtung Sie es zuerst verschieben. Entsprechendes gilt für mehrere Steuerelemente, die Sie gleichzeitig markiert haben und verschieben.

15.6.5 Größe von Steuerelementen ändern

Manuell

- Um die Größe eines Steuerelements zu ändern, ziehen Sie einen seiner Ziehpunkte zur Größenänderung (vgl. oben Bild 15.26).

  Oder:

- Markieren Sie das Steuerelement. Drücken Sie eine der vier Pfeiltasten bei gedrückter [⇧]-Taste. Beispielsweise vergrößern Sie das Steuerelement nach rechts, wenn Sie bei gedrückter [⇧]-Taste die [→] drücken. Diese Methode kann dem Ziehen mit der Maus vorzuziehen sein, wenn es um sehr geringe Größenänderungen geht.

Automatische Größenanpassung

Sie können die Größe eines Steuerelements automatisch a) an die Größe seines Inhalts bzw. seiner Schriftgröße oder b) seine Ränder an die nächsten Rasterpunkte anpassen lassen. Dies empfiehlt sich besonders a) bei Bezeichnungsfeldern, deren Textlänge Sie verändert haben oder b) wenn Sie nachträglich die Rastergenauigkeit geändert haben. Gehen Sie folgendermaßen vor:

- Markieren Sie das oder die Steuerelement(e).
- Wählen Sie den Befehl *Grösse* aus dem Menü *Format* und einen der Unterbefehle *An Textgrösse* bzw. *Am Raster.*

15.6.6 Schriftbild von Steuerelementen ändern

Bild 15.30: Symbol-Schaltflächen zum Formatieren in der Symbolleiste

Das Schriftbild von Steuerelementen verändern Sie auf die folgende Weise:

- Markieren Sie das oder die Steuerelement(e).
- Wählen Sie mit den oben dargestellten Symbol-Schaltflächen das oder die Schriftbildmerkmale, das/die Sie zuweisen wollen. Die Symbole bedeuten in der Reihenfolge von links nach rechts:

- Schriftart
- Schriftgrad
- Fett
- Kursiv
- Unterstrichen
- Linksbündige Ausrichtung
- Zentrierte Ausrichtung
- Rechtsbündige Ausrichtung

15.6.7 Farben für Steuerelemente ändern

Die Farben von Steuerelementen verändern Sie auf die folgende Weise:

Bild 15.31: Symbol-Schaltflächen der Format-Symbolleiste zum Einstellen von Farben. Die Symbol-Schaltflächen bedeuten von links nach rechts: »Füll-/Hintergrundfarbe«, »Schrift-/Vordergrundfarbe, »Linien-/Rahmenfarbe«.

- Markieren Sie das oder die Steuerelement(e).
- Schlagen Sie in der Format-Symbolleiste die Dropdown-Liste einer der drei Symbol-Schaltflächen *Füll-/Hintergrundfarbe*, *Schrift-/Vordergrundfarbe* oder *Linien-/Rahmenfarbe* auf, und wählen Sie die gewünschte Farbe.

15.6.8 Steuerelemente am Raster ausrichten

Den Abstand der Rasterpunkte legen Sie fest, indem Sie die Formulareigenschaften *Raster X* und *Raster Y* einstellen; zum Einstellen von Formulareigenschaften vgl. weiter unten in diesem Kapitel, Punkt 15.7, *Eigenschaften von Steuerelementen, Formular und Bereichen einstellen*.

Manuell am Raster ausrichten

Um zu erreichen, daß eine Positions- oder Größenänderung oder das Einfügen eines Steuerelements am Raster ausgerichtet wird, müssen Sie ggf. den Befehl *Am Raster ausrichten* aus dem Menü *Format* wählen, so daß er eingeschaltet ist. Nochmalige Wahl dieses Befehls schaltet die automatische Ausrichtung am Raster aus.

Nachträglich an das Raster anpassen lassen

Sie haben zwei Möglichkeiten, Steuerelemente nachträglich an das Raster anpassen zu lassen:

Alle Seiten am Raster ausrichten lassen

- Markieren Sie das oder die Steuerelement(e).
- Wählen Sie den Befehl *Grösse anpassen* aus dem Menü *Format*.
- Wählen Sie den Unterbefehl *am Raster*.

Linke obere Ecke am Raster ausrichten lassen

- Markieren Sie das oder die Steuerelement(e).
- Wählen Sie aus dem Menü *Format* den Befehl *Ausrichten* und dessen Unterbefehl *am Raster*.

15.6.9 Steuerelemente horizontal oder vertikal ausrichten

Sehr oft ergibt sich die Notwendigkeit, verschiedene Steuerelemente nachträglich so auszurichten, daß sie alle beispielsweise rechts an derselben senkrechten (gedachten) Linie oder alle an derselben waagerechten Linie ausgerichtet werden. Gehen Sie dazu folgendermaßen vor:

- Markieren Sie die Steuerelemente.
- Wählen Sie aus dem Menü *Format* den Befehl *Ausrichten* und dazu einen der Unterbefehle *Linksbündig*, *Rechtsbündig*, *Oben* oder *Unten*. Die Unterbefehle haben die nachfolgende Bedeutung:

Linksbündig

Richtet den linken Rand der markierten Steuerelemente am äußerst linken markierten Steuerelement aus.

Rechtsbündig

Richtet den rechten Rand der markierten Steuerelemente am äußerst rechten markierten Steuerelement aus.

Oben

Richtet den oberen Rand der markierten Steuerelemente am obersten markierten Steuerelement aus.

Unten

Richtet den unteren Rand der markierten Steuerelemente am untersten markierten Steuerelement aus.

15.6.10 Horizontalen oder vertikalen Abstand von Steuerelementen gleichmäßig ausrichten

Sie können den vertikalen oder horizontalen Abstand von Steuerelementen am Raster ausrichten lassen, vergrößern oder verkleinern. Gehen Sie folgendermaßen vor:

- Markieren Sie die Steuerelemente.
- Wählen Sie aus dem Menü *Format* den Befehl *Vertikaler Abstand* bzw. *Horizontaler Abstand* und dann einen der Unterbefehle *Angleichen*, *Vergrößern* oder *Vermindern*. Die Unterbefehle haben die nachfolgende Bedeutung:

Angleichen

Gleicht den vertikalen bzw. horizontalen Abstand zwischen drei oder mehr markierten Steuerelementen an.

Vergrößern

Vergrößert den vertikalen bzw. horizontalen Abstand zwischen markierten Steuerelementen um einen Rasterpunkt. Wenn drei oder mehr Steuerelemente markiert sind, wird zunächst deren Abstand angeglichen und anschließend vergrößert.

Vermindern

Verkleinert den vertikalen bzw. horizontalen Abstand zwischen markierten Steuerelementen um einen Rasterpunkt. Wenn drei oder mehr Steuerelemente markiert sind, wird zunächst deren Abstand angeglichen und anschließend verkleinert.

15.6.11 Bedingte Formatierung von Steuerelementen

Bedingte Formatierung hinzufügen

Sie können das Erscheinungsbild von Steuerelementen in Abhängigkeit von seinen Wertinhalten gestalten. Gehen Sie folgendermaßen vor:

- Markieren Sie das Steuerelement.
- Wählen Sie aus dem Menü *Format* den Befehl *Bedingte Formatierung...*
- Stellen Sie im Dialogfeld *Bedingte Formatierung* (vgl. Bild 15.32) die Bedingung und die zugehörigen Formatierungsmerkmale ein.
- Klicken Sie auf die Schaltfläche *Hinzufügen* >>, um anschließend eine weitere Bedingung anzugeben, andernfalls auf OK. Sie können insgesamt bis zu drei bedingte Formatierungen für ein Steuerelement angeben.

Bild 15.32: Dialogfeld »Bedingte Formatierung«

Bedingte Formatierung löschen

Sie löschen einzelne bedingte Formatierungen auf die folgende Weise:

- Markieren Sie das Steuerelement.
- Wählen Sie aus dem Menü *Format* den Befehl *Bedingte Formatierung...*
- Klicken Sie im Dialogfeld *Bedingte Formatierung* auf die Schaltfläche Löschen...

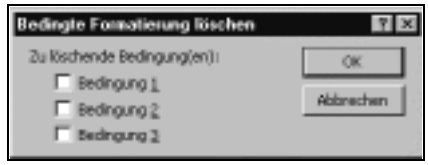


Bild 15.33: Dialogfeld »Bedingte Formatierung löschen«

- Aktivieren Sie im Dialogfeld *Bedingte Formatierung löschen* die zu löschende(n) Bedingung(en), und klicken Sie auf OK.
- Schließen Sie anschließend das Dialogfeld *Bedingte Formatierung* mit OK.

Bild 15.34: Das Steuerelement »Auftragswert« wurde mit den Merkmalen aus Bild 15.32 formatiert.

15.7 Eigenschaften von Steuerelementen, Formular und Bereichen einstellen

15.7.1 Eigenschaften mit dem Eigenschaftenfenster einstellen

Jedes Steuerelement, das Formular selbst und jeder Formular-Bereich besitzt Eigenschaften, für die verschiedene Werte eingestellt werden können. Dies geschieht mit dem Eigenschaftenfenster. Das Eigenschaftenfenster arbeitet kontextabhängig und bietet jeweils die Eigenschaften des markierten Formularelements zur Bearbeitung an. Sie können umgekehrt erfahren, welche Eigenschaften für ein bestimmtes Formularelement verfügbar sind, wenn Sie, während dieses Element markiert ist, in der Liste der Eigenschaften im Eigenschaftenfenster blättern.

Um eine Eigenschaft eines Formularelements (Formular, Bereich, Steuerelement) einzustellen, gehen Sie folgendermaßen vor:

- Markieren Sie das betreffende Element: Ein Steuerelement markieren Sie durch Klicken darauf. Einen Bereich markieren Sie durch Klicken auf einen beliebigen freien Punkt darin. Sie können auch mehrere Steuerelemente markieren. Dann bietet das Eigenschaftenfenster nur die Eigenschaften zum Bearbeiten an, die jedes der markierten Steuerelemente besitzt.

Bild 15.35: Der Mauszeiger weist auf das kleine Quadrat, den Formular-Markierer.

- *Formular markieren*. Das Formular markieren Sie mit dem Befehl *Formular auswählen* aus dem Menü *Bearbeiten*. Mit der Maus klicken Sie auf den Formular-Markierer, das kleine Quadrat, das sich in der linken oberen Ecke des Formulars befindet (bei eingeblendeten Linealen dort, wo die beiden Lineale zusammenstoßen). Wenn das Formular markiert ist, wird im Formular-Markierer ein kleines schwarzes Quadrat angezeigt, vgl. Bild 15.35.
- Blenden Sie ggf. das Eigenschaftenfenster ein, beispielsweise durch Klikken auf die entsprechende Symbol-Schaltfläche. Das eingeblendete Eigenschaftenfenster zeigt in seiner Titelleiste an, auf welches Objekt es sich

momentan bezieht. Wenn z.B. ein Textfeld markiert ist, enthält die Titelleiste des Eigenschaftenfensters den Eintrag *Textfeld*. Wenn mehrere Steuerelemente markiert sind, lautet der Titel *Mehrfachauswahl*.

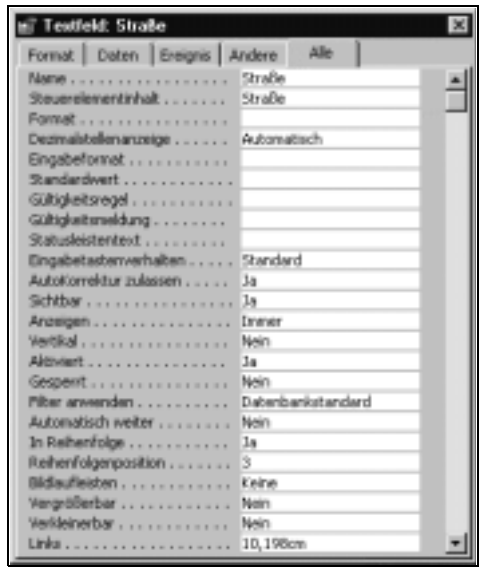


Bild 15.36: Eigenschaftenfenster mit verschiedenen Registerkarten zum Beschränken der angezeigten Eigenschaften auf eine bestimmte Kategorie.

- Die meisten Steuerelemente besitzen sehr viele Eigenschaften. Beispielsweise hat ein Textfeld 65 Eigenschaften. Daher ist es oft schwierig, in der langen Liste von Eigenschaften die gewünschte zu suchen und zu aktivieren. Sie können jedoch Art und damit Anzahl der im Eigenschaftenfenster angebotenen Eigenschaften beschränken, indem Sie nicht die Registerkarte *Alle*, sondern eine der anderen Registerkarten aktivieren, vgl. Bild 15.36. Wenn Sie beispielsweise die Registerkarte *Daten* wählen, werden für ein Textfeld nur noch acht Eigenschaften angeboten, vgl. Bild 15.37.
- Blättern Sie ggf. durch die Eigenschaftenliste, und klicken Sie in das Bearbeitungsfeld der gewünschten Eigenschaft. Viele Bearbeitungsfelder bieten, wenn sie den Fokus haben, ein aufschlagbares Dropdown-Listenfeld an, aus dem Sie den Eintrag wählen können. In die anderen Felder schreiben Sie den Wert, auf den die Eigenschaft eingestellt werden soll, hinein. In einigen Fällen kann es auch sinnvoll sein, einen Text oder einen Ausdruck in ein Eingabefeld zu schreiben, wenn ein Dropdown-Listenfeld angeboten wird. Dies trifft beispielsweise für die Eigenschaft *Steuerelementinhalt* eines Textfeldes zu, wenn Sie es als berechnetes Steuerelement behandeln wollen. Für lange Texte oder Ausdrücke empfiehlt es sich, das Dialogfeld *Zoom* aufzuschlagen (Tastenkombination ⇧+F2 oder Befehl *Zoom* aus dem Kontextmenü).

Bild 15.37: Die vom Eigenschaftenfenster angebotenen Eigenschaften sind auf die Daten-Eigenschaften beschränkt, so daß beispielsweise für ein Textfeld nur acht Eigenschaften angeboten werden.

Einige Eigenschaften bieten darüber hinaus die Möglichkeit, durch Klikken auf die Symbol-Schaltfläche *Generator* (dieses Symbol trägt drei Pünktchen, vgl. in Bild 15.37 das Symbol am rechten Rand der Eigenschaft *Steuerelementinhalt*) den Ausdrucks-, Makro- oder Code-Generator aufzurufen.

Wenn Sie auf das Bearbeitungsfeld einer Eigenschaft, das eine Dropdown-Liste anbietet, doppelklicken, wird der nächste Eintrag der Liste angezeigt. Diese Wirkung tritt (leider) auch dann ein, wenn Sie lediglich ein Wort in dem Bearbeitungsfeld durch Doppelklicken markieren wollen. Daher kann aus Doppelklicken leicht eine unbeabsichtigte und unbemerkte Änderung der Eigenschafteneinstellung resultieren.

Falls Sie Hilfe für eine Eigenschaft brauchen: In der Statusleiste wird jeweils für die Eigenschaft mit dem Fokus eine Kurzerklärung eingeblendet. Für längere Erklärungen zu einer bestimmten Eigenschaft drücken Sie die Taste F1, während die Eigenschaft im Eigenschaftenfenster den Fokus hat.

15.7.2 Standardeigenschaften von Steuerelementen festlegen

Mit dem Eigenschaftenfenster

Wenn Sie einem Formular ein neues Steuerelement hinzufügen, bekommt es bestimmte Einstellungen für seine Eigenschaften zugewiesen. Diese werden *Standardeigenschaften* genannt. Die Standardeigenschaften können für jedes Steuerelement vom Benutzer festgelegt werden. Beispielsweise können Sie für ein Textfeld als Standardeigenschaft den Schriftgrad 12 festlegen. Daraufhin bekommen alle diesem Formular neu hinzugefügten Textfelder den Schriftgrad 12, bis Sie ausdrücklich einen neuen Schriftgrad als Standardeigenschaft festlegen. Standardeigenschaften gelten formularspezifisch und werden zusammen mit dem Formular gespeichert. Formularübergreifende Standardeigenschaften können Sie mit einer Formularvorlage definieren, vgl. Punkt 15.7.3, *Formularvorlage definieren*.

Sie legen Standardeigenschaften von Steuerelementen auf die folgende Weise fest:

- Blenden Sie ggf. die Toolbox und das Eigenschaftenfenster ein.
- Klicken Sie in der Toolbox auf das Symbol für das Steuerelement, dessen Standardeigenschaften Sie ändern wollen. In der Titelleiste des Eigenschaftenfensters erscheint dann der Text *Standard: Steuerelementname*. In Bild 15.38 beispielsweise können im Eigenschaftenfenster die Format-Eigenschaften für ein Textfeld eingestellt werden. In der dort wiedergegebenen Situation wurde u.a. die Eigenschaft *Mit Bezeichnungsfeld* auf den Wert *Nein* eingestellt mit der Folge, daß künftig neue Textfelder ohne Bezeichnungsfeld eingefügt werden.
- Ändern Sie im Eigenschaftenfenster die gewünschten Eigenschaften. Sie dürfen zwischen dem vorigen und diesem Schritt nicht auf einen Punkt des Formulars klicken, weil Sie sonst eine Elementeigenschaft, nicht jedoch eine Standardeigenschaft ändern würden.

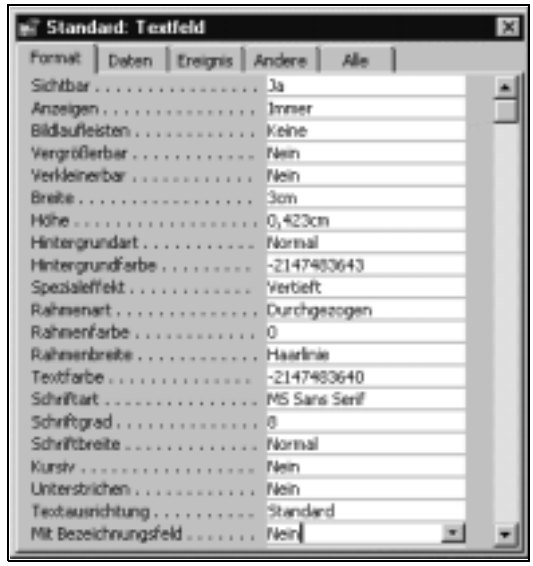


Bild 15.38: In diesem Eigenschaftenfenster können die Standard-Format-Eigenschaften für ein Textfeld eingestellt werden: In der Toolbox ist die Symbol-Schaltfläche »Textfeld« gedrückt, und im Eigenschaftenfenster ist die Registerkarte »Format« gewählt.

Auf Basis eines Beispiel-Steuerelements

Sie können die Standardeigenschaften für ein Steuerelement auch auf der Grundlage eines vorhandenen Steuerelements festlegen: Die Eigenschaften-einstellungen dieses Beispiels werden dann für neu in das betreffende Formular eingefügte Steuerelemente allgemeingültig. Gehen Sie folgendermaßen vor:

- Markieren Sie das Steuerelement, dessen Eigenschafteneinstellungen Standard werden sollen.
- Wählen sie aus dem Menü *Format* den Befehl *Steuerelementvorgaben festlegen*.

Der Vorteil dieser Methode zeigt sich beispielsweise, wenn Sie die Eigenschaften eines bestimmten Steuerelements sorgfältig eingestellt haben und die weiteren in das Formular einzufügenden Steuerelemente gleichen Typs genauso wie dieses Vorbild haben möchten. Die vorangehend geschilderte Methode, Standardeigenschaften mit Hilfe des Eigenschaftenfensters einzustellen, bietet sich dagegen an, wenn Sie gezielt einzelne Standardeigenschaften einstellen wollen.

15.7.3 Formularvorlage definieren

Wenn Sie ein neues leeres Formular erstellen, stellt Access dessen Eigenschaften auf bestimmte Werte ein, und auch die Steuerelemente, die Sie in ein neues Formular einfügen, bekommen bestimmte Standardeigenschaften zugewiesen. Sie können Access veranlassen, die Standardeigenschaften für neue Formulare und deren Steuerelemente einer von Ihnen definierten Formatvorlage zu entnehmen. Auf diese Weise können Sie beispielsweise erreichen, daß alle neuen leeren Formulare einen grauen Hintergrund bekommen oder, um ein anderes Beispiel zu nennen, neue Textfelder ohne Bezeichnungsfeld erstellt werden. Um eine Formularvorlage festzulegen, gehen Sie folgendermaßen vor:

- Wählen Sie den Befehl *Optionen* aus dem Menü *Extras*, und aktivieren Sie in dessen Dialogfeld die Registerkarte *Formulare/Berichte*.
- Geben Sie in das Eingabefeld *Formularvorlage* den Namen eines Formulars ein, das als Formularvorlage dienen soll. (Standardmäßig enthält dieses Eingabefeld den Eintrag *Normal*.) Bestätigen Sie mit OK.

Wenn Sie auf diese Weise ein Formular als Vorlage angegeben haben, verwendet Access dessen Eigenschaften als Standardeigenschaften. Wenn das Formular beispielsweise einen Seitenkopf enthält, bekommt ein neues leeres Formular ebenfalls einen Seitenkopf. Es ist unerheblich, ob das als Vorlage angegebene Formular Steuerelemente enthält oder nicht: Wenn Sie ein neues leeres Formular erstellen, wird dies stets leer, d.h. ohne Steuerelemente, erzeugt. Daher können Sie ohne weiteres ein Formular als Formularvorlage angeben, das praktisch in anderen Zusammenhängen eingesetzt wird.

Access speichert den Namen des Formulars, das Sie im Dialogfeld des Befehls *Optionen* als Formularvorlage angegeben haben, nicht zusammen mit der Datenbank, sondern in der Windows-Registrierung, und zwar unter dem Schlüssel

\HKEY_CURRENT_USER\SOFTWARE\MICROSOFT\OFFICE\9.0
\ACCESS\SETTINGS, Wert zum Element *Form Template*.

Daher erscheint der als Formularvorlage angegebene Name auch dann im Dialogfeld des Befehls *Optionen*, wenn Sie eine andere Datenbank geöffnet haben, in der ein Formular mit diesem Namen nicht existiert. Dann allerdings *verwendet* Access die Formularvorlage *Normal*. Wenn Sie wollen, daß die Eigenschaften des als Vorlage angegebenen Formulars auch in einer anderen Datenbank zur Wirkung kommen, müssen Sie dieses Formular in die andere Datenbank kopieren bzw. importieren. Dies ist leider nicht sehr glücklich geregelt!

15.8 Ereignisprozeduren und -makros in Formularen

Das Formular selbst und die meisten Steuerelemente besitzen Ereigniseigenschaften. Ein Ereignis ist ein von einem Objekt (hier: Formular oder Steuerelement) erkannter Vorgang, z.B. das Klicken mit der Maus oder das Zuweisen des Fokus an das Objekt, für den Sie eine Reaktion festlegen können. Dies geschieht dadurch, daß Sie als Einstellung der betreffenden Ereigniseigenschaft den Namen einer VBA-Prozedur bzw. eines Makros angeben. Die Prozedur bzw. das Makro wird dann ausgeführt, wenn das Ereignis eintritt. Beispielsweise könnten Sie für die Ereigniseigenschaft *Nach Aktualisierung* eines Textfeldes eine VBA-Prozedur festlegen, die ein anderes Textfeld in Abhängigkeit vom neuen Wert des ersten Textfeldes deaktiviert.

Wie Sie derartige Makros oder VBA-Prozeduren erstellen können, ist nicht Gegenstand dieses Abschnitts, sondern wird ausführlich in späteren Teilen dieses Buches behandelt. Hier geht es um die Organisation der Ereignisprozeduren und -makros aus der Sicht eines Formulars.

Organisation von Ereignismakros

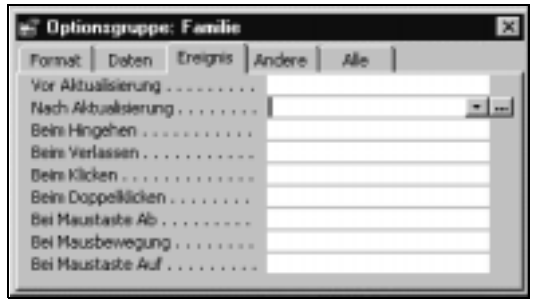


Bild 15.39: Für die Ereigniseigenschaft »Nach Aktualisierung«, die im Eigenschaftenfenster den Fokus hat, ist die Symbol-Schaltfläche »Generator« (mit den drei Pünktchen) verfügbar.

Makros sind prinzipiell unabhängig von Formularen organisiert. Sie werden als eigenständige Makro-Objekte erstellt, die völlig unabhängig von einem Formular existieren können. Ein Makro wird nur dadurch zu einem Ereignismakro, daß sein Name als Einstellung einer Ereigniseigenschaft

angeführt ist. Umgekehrt gilt, daß ein Makro auch unabhängig von einem Ereignis ausgeführt werden kann, wenngleich dies in den meisten Fällen inhaltlich wenig Sinn ergeben mag. Zum Bearbeiten eines bestehenden oder Erstellen eines neuen Ereignismakros können Sie, während die betreffende Ereigniseigenschaft im Eigenschaftenfenster den Fokus hat, den Makro-Generator durch Klicken auf die Symbol-Schaltfläche *Generator* aufrufen, vgl. Bild 15.39.

Organisation von Ereignisprozeduren

Ereignisprozeduren sind in *Visual Basic for Applications (VBA)* geschriebene Prozeduren. Wie alle VBA-Prozeduren werden sie in Modulen erstellt. Um eine Ereignisprozedur zu erstellen, rufen Sie, während die betreffende Eigenschaft im Eigenschaftenfenster den Fokus hat, den Code-Generator auf, indem Sie auf die Symbol-Schaltfläche *Generator* klicken, vgl. Bild 15.39, und darauf im Dialogfeld *Generator auswählen* den *Code- Generator* auswählen. (Sofern die Eigenschaft bereits den Eintrag *[Ereignisprozedur]* aufweist, wird der Code-Generator unmittelbar nach dem Klicken auf die Symbol-Schaltfläche *Generator* geöffnet.) Der Code-Generator öffnet dann ein Modul, das speziell dem aktiven Formular zugeordnet ist. In dieses Modul werden alle Ereignisprozeduren für dieses Formular geschrieben. Das Modul bekommt den Namen *Form_Formularname*. Wenn das betreffende Formular beispielsweise den Namen *Sozialangaben* hat, bekommt das Modul den Namen *Form_ Sozialangaben*, vgl. Bild 15.40.

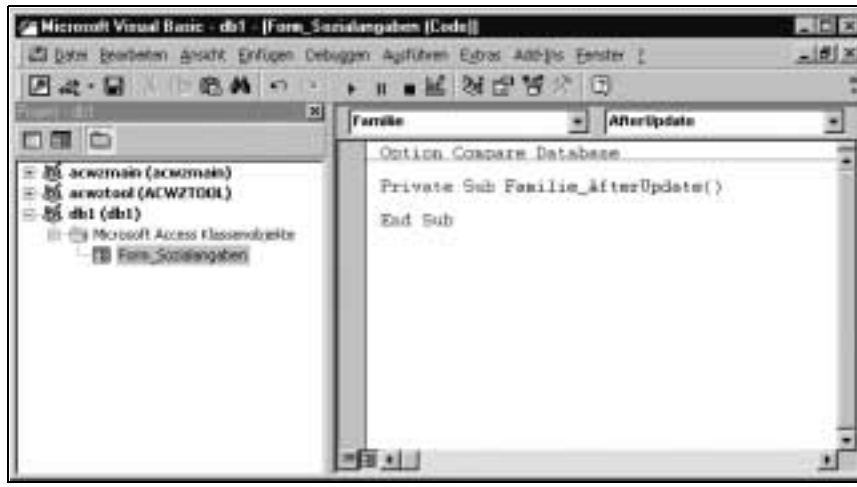


Bild 15.40: Modul für Ereignisprozeduren des Formulars »Sozialangaben«. Die Sub-Prozedur »Familie_AfterUpdate«, von der hier nur die erste und letzte Prozedurzeile existieren, ist dem Ereignis »Nach Aktualisierung« des Steuerelements »Familie« zugeordnet. Die Nomenklatur für Modul und Prozedur wird von Access festgelegt.

Access vergibt nicht nur für das Ereignismodul Namen, sondern auch für dessen Prozeduren. Der in Bild 15.40 wiedergegebene Name *Familie_AfterUpdate* beispielsweise erklärt sich folgendermaßen: Das Ereignismodul wurde mit dem Code-Editor geöffnet, während die Eigenschaft *Nach Aktualisierung* des Steuerelements *Familie* im Eigenschaftenfenster den Fokus hatte. Daher bietet Access im Ereignismodul die zugehörige Ereignisprozedur zum Bearbeiten an. Der Prozedurname setzt sich stets aus dem Namen des Steuerelements und dem (amerikanischen) Namen der betreffenden Eigenschaft zusammen.

Ereignismodule sind formulargebunden

Ein Ereignismodul wird zusammen mit dem Formular gespeichert. Dies bedeutet umgekehrt, daß die darin enthaltenen Prozeduren verloren gehen, wenn das Formular gelöscht wird. Anders als gewöhnliche VBA-Module werden Ereignismodule nicht in der Objektliste *Module* des Datenbankfensters angezeigt.

Um ein Ereignismodul zu öffnen, muß daher das zugehörige Formular in der Entwurfsansicht geöffnet und aktiv sein. Dann können Sie entweder eine Ereignisprozedur in der oben beschriebenen Weise vom Eigenschaftenfenster aus mit der Symbol-Schaltfläche *Generator* bearbeiten, oder Sie klicken auf die Symbol-Schaltfläche *Code* in der Symbolleiste, vgl. links nebenstehend. Alternativ öffnen Sie den VBA-Editor – z.B. mit der Tastenkombination Alt+F11 – und dort im Projektexplorer das Klassenmodul, das den Namen *Form_Formularname* besitzt.

Ereignismodule sind sogen. Klassenmodule. Die Prozeduren eines Klassenmoduls stehen nur für das zugehörige Formular zur Verfügung. Das Modul ist *privat*, so daß seine Prozeduren nicht für Prozeduren anderer Module zur Verfügung stehen. Umgekehrt sind Prozeduren allgemeiner Module in einem Klassenmodul durchaus verfügbar.

Gewöhnliche Prozeduren als Ereignisprozeduren einstellen

Sie können jede gewöhnliche VBA-Function-Prozedur als Einstellung für eine Ereigniseigenschaft angeben. In diesem Falle müssen Sie dem Funktionsnamen ein Gleichheitszeichen (=) voranstellen. Um beispielsweise für die Eigenschaft *Nach Aktualisierung* die VBA-Funktion *EinkommenNeu()* festzulegen, müssen Sie dies in der in Bild 15.41 dargestellten Weise angeben. (VBA-Prozeduren werden ausführlich in Kap. 29, *VBA (Visual Basic for Applications): Grundlagen*, behandelt.)

Gewöhnliche VBA-Sub-Prozeduren können – anders als die formularspezifischen Sub-Prozeduren – für eine Ereigniseigenschaft nicht direkt angegeben werden. Um eine Sub-Prozedur für eine Ereigniseigenschaft zu verwenden, müssen Sie mit dem folgenden Trick arbeiten: Erstellen Sie eine VBA-Funktion, welche die Sub-Prozedur aufruft, und geben Sie die Funktion für die Ereigniseigenschaft an.

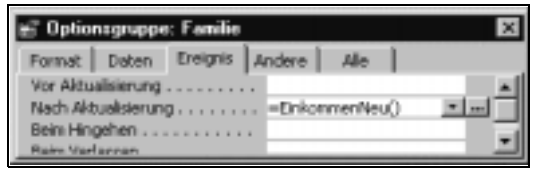


Bild 15.41: Der Ereigniseigenschaft »Nach Aktualisierung« ist die VBA-Funktion »EinkommenNeu()« zugeordnet.

15.9 Bereiche einfügen und entfernen

Ein Formular kann die folgenden Bereiche aufweisen:

- *Detailbereich*
- *Seitenkopf und -fuß*
- *Formularkopf und -fuß.*

Detailbereich

Dieser Bereich ist immer vorhanden und wird daher beim Erstellen eines neuen Formulars automatisch angezeigt. Er stellt den Hauptbereich eines Formulars dar, in den die Steuerelemente aufgenommen werden, die den Inhalt der Datensätze wiedergeben. Wenn Sie ein Formular in der Datenblattansicht darstellen, kann dieser Bereich gerollt werden.

Seitenkopf und -fuß

Sie fügen diesen Bereich mit dem Befehl *Seitenkopf/-fuß* aus dem Menü *Ansicht* hinzu und entfernen ihn ggf. mit demselben Befehl. Ein Seitenkopf und -fuß wird beim Drucken eines Formulars, nicht dagegen in der Formularansicht auf dem Bildschirm wiedergegeben, so daß auch der Inhalt von Steuerelementen, die in diesem Bereich plaziert sind, nur dann erscheint. Da Sie die Größe von Seitenkopf und -fuß getrennt einstellen können, ist es beispielsweise möglich, den Seitenkopf zu unterdrücken, indem Sie seine Höhe auf Null einstellen, während der Seitenfuß angezeigt bleibt.

Formularkopf und -fuß

Sie fügen diesen Bereich mit dem Befehl *Formularkopf/-fuß* aus dem Menü *Ansicht* hinzu und entfernen ihn ggf. mit demselben Befehl. Ein Formularkopf und -fuß wird nicht nur beim Drucken, sondern auch in der Formulardarstellung auf dem Bildschirm wiedergegeben. Der Inhalt eines Formularkopfes erscheint am Beginn eines Formulars, der eines Fußes am Ende. Die Größe von Formularkopf und -fuß kann, wie beim Seitenkopf und -fuß, getrennt für Kopf und Fuß eingestellt werden, so daß Sie auch hier den einen Teil unterdrücken können, während der andere angezeigt wird.

15.10 Formular für den Druck einrichten

In den meisten Fällen mögen Sie ein Formular ausschließlich für den Bildschirm konzipieren. Es ist jedoch auch möglich und vielfach sehr sinnvoll, ein Formular auszudrucken. Wegen der reicheren Gestaltungsmöglichkeiten von Formularen gegenüber Berichten, die ja fast ausschließlich für die Druckausgabe konzipiert werden, ist ein gedrucktes Formular manchmal einem Bericht vorzuziehen.

Seitenränder, Papierformat und Spalten

Bei Formularen, die Sie zum Ausdrucken entwerfen, müssen Sie vor allem darauf achten, daß die Breite der Formularbereiche, die Sie z.B. am waagerechten Lineal ablesen können, mit den für das Papier angegebenen Seitenrändern korrespondiert. Mit dem folgenden Verfahren bestimmen Sie das Papierformat, die Seitenränder und ggf. Spalten für ein bestimmtes Formular:

Bild 15.42: Dialogfeld »Seite einrichten«

- Sorgen Sie dafür, daß das betreffende Formular den Fokus hat, gleichgültig, ob in Formular-, Datenblatt oder Entwurfsansicht. Es reicht für diesen Zweck auch, wenn es im Datenbankfenster markiert ist.
- Wählen Sie den Befehl *Seite einrichten...* aus dem Menü *Datei*.

- Um die Seitenränder festzulegen, aktivieren Sie die Registerkarte *Ränder* des Dialogfeldes *Seite einrichten*, vgl. Bild 15.42.
- Um das Papierformat – u.a. Hoch- oder Querformat – festzulegen, aktivieren Sie die Registerkarte *Seite*.
- Um Spalten und deren Breiten und Abstände festzulegen, aktivieren Sie die Registerkarte *Spalte*.

Die Angaben für den Druck sind formularspezifisch und werden zusammen mit dem Formular gespeichert.

Seitenansicht

Um zu prüfen, mit welchem Aussehen ein Formular auf dem Drucker ausgegeben wird, können Sie es in der Seitenansicht betrachten.

- Sorgen Sie ggf. dafür, daß das betreffende Formular den Fokus hat, sei es in Formular-, Datenblatt oder Entwurfsansicht. Es reicht für diesen Zweck auch, wenn es im Datenbankfenster markiert ist.
- Sie rufen die Seitenansicht durch Klicken auf die Symbol-Schaltfläche *Seitenansicht* auf. Alternativ wählen Sie den Befehl *Seitenansicht* aus dem Menü *Datei*.
- Lassen Sie das Formular aus der Seitenansicht heraus drucken oder kehren Sie durch Klicken auf die Symbol-Schaltfläche *Schließen* zur Ausgangsansicht zurück. Zum vergrößerten Betrachten klicken Sie mit dem Lupen-Mauszeiger auf einen Punkt der Seitenansicht. Mit nochmaligem Klicken schalten Sie wieder in die verkleinerte Darstellung zurück.

15.11 Formulardefinition drucken

Sie können auch die Definitionen eines oder mehrerer Formulare drucken lassen:

- Rufen Sie den *Dokumentierer* auf, indem Sie im Menü *Extras* den Befehl *Analyse*, Unterbefehl *Dokumentierer*, wählen.
- Aktivieren Sie im Dialogfeld *Dokumentierer* die Registerkarte *Formulare*, und kreuzen Sie die Formulare an, deren Definitionsmerkmale ausgegeben werden sollen.
- Klicken Sie im Dialogfeld *Dokumentierer* auf die Schaltfläche *Optionen...*, um das Dialogfeld *Formulardefinition drucken* zu öffnen, vgl. Bild 15.43, in dem die beiden Dialogfelder *Dokumentierer* und *Formulardefinitionen drucken* angezeigt werden.
- Schließen Sie die beiden Dialogfelder jeweils mit OK. Dann wird ein Bericht mit den gewählten Merkmalen der Formulardefinitionen angezeigt, in dem Sie blättern und den Sie drucken lassen können.

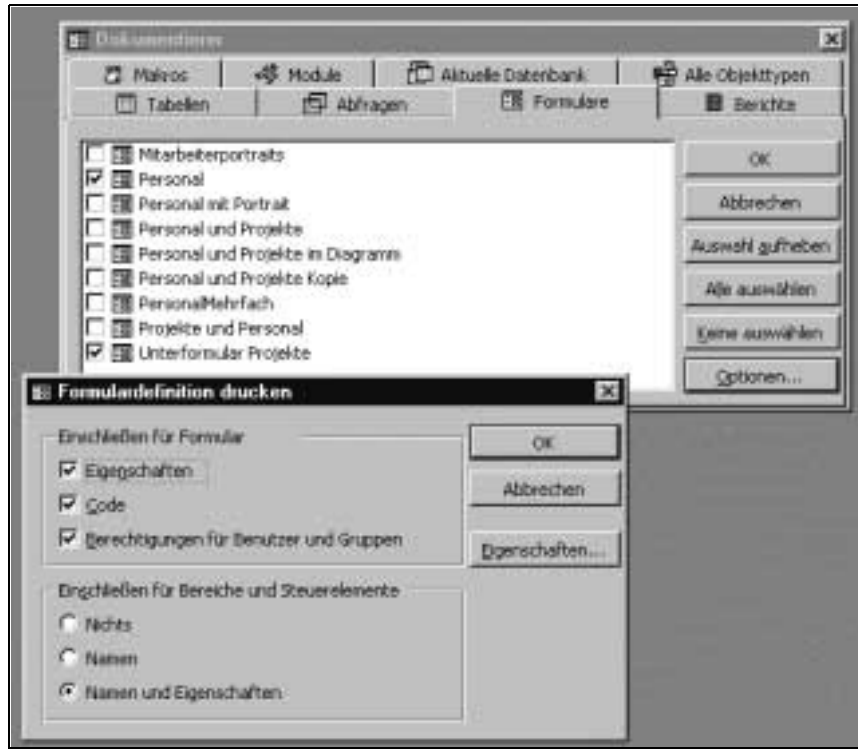


Bild 15.43: Dialogfelder »Dokumentierer« und »Formulardefinitionen drucken«. Das letztere wurde aus dem ersteren heraus mit der Schaltfläche »Optionen...« aufgerufen.

Kapitel 16

Standard-Steuerelemente in Formularen

Dieses Kapitel behandelt die einzelnen Steuerelemente, die standardmäßig in der Toolbox angeboten werden und die Sie in ein Formular und (mit Einschränkungen) in einen Bericht einfügen können, im Detail. Hier wird vorausgesetzt, daß Ihnen die grundsätzlichen Funktionen zum Einfügen und Bearbeiten von Steuerelementen bekannt sind. Diese werden im vorangehenden Kap. 15, *Formulare: Grundlagen*, beschrieben und erklärt. Zusätzlich zu den Standard-Steuerelementen können Sie zahlreiche weitere Steuerelemente in Formulare und Berichte einfügen, die als ActiveX-Steuerelemente oder OLE-Zusatzsteuerelemente bezeichnet werden. Darauf wird im nächsten Kapitel eingegangen, vgl. Kap. 17, *ActiveX-Steuerelemente in Formularen*.

16.1 Textfeld

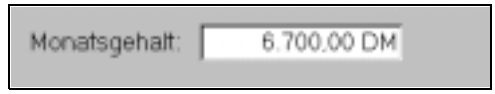


Bild 16.1: Textfeld mit Bezeichnungsfeld in der Formularansicht. Die Eigenschaft »Format« des Textfeldes ist auf den Wert »Währung« eingestellt.

Textfelder leisten mehr, als ihr Name vermuten läßt: Sie geben nicht nur Texte wieder und ermöglichen deren Bearbeitung, sondern auch Zahlenwerte. Aus diesem Grunde sind Textfelder vermutlich die am meisten in Formularen verwendeten Steuerelemente.

Textfeld für längere Texte

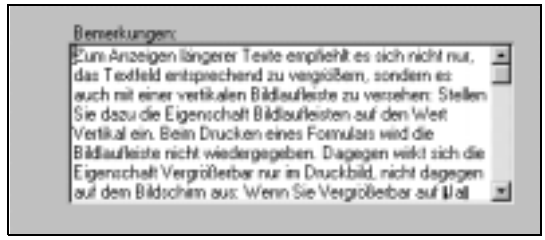


Bild 16.2: Textfeld mit Bezeichnungsfeld, das Textfeld mit vertikaler Bildlaufleiste zum Rollen längerer Texte.

Zum Anzeigen längerer Texte empfiehlt es sich nicht nur, das Textfeld entsprechend zu vergrößern, sondern es auch mit einer vertikalen Bildlaufleiste zu versehen: Stellen Sie dazu die Eigenschaft *Bildlaufleisten* auf den Wert *Vertikal* ein. Beim Drucken eines Formulars wird die Bildlaufleiste nicht wiedergegeben. Dagegen wirkt sich die Eigenschaft *Vergrößerbar* nur im Druckbild, nicht dagegen auf dem Bildschirm aus: Wenn Sie *Vergrößerbar* auf *Ja* eingestellt haben, wird im Druck das Textfeld so weit vergrößert, daß der gesamte Text für dieses Feld wiedergegeben wird; die darunterliegenden Steuerelemente werden entsprechend nach unten verschoben. Entsprechend umgekehrt wirkt die Eigenschaft *Verkleinerbar.*

Textfeld zur Wiedergabe von Zahlen oder Datumswerten

Wenn ein Textfeld Zahlen oder Datumswerte anzeigen soll, ist es im allgemeinen empfehlenswert, die Eigenschaft *Format* entsprechend einzustellen. Sie können dabei aus einer Liste vordefinierter Formate wählen, aber auch ein eigenes benutzerdefiniertes Format angeben. Wie Sie eigene Formate schreiben können, erfahren Sie in Kap. 8, *Tabellen: Grundlagen*, Punkt 8.7, *Benutzerdefinierte Anzeigeformate*.

Berechnetes Textfeld

Sie können für die Eigenschaft *Steuerelementinhalt* einen Ausdruck angeben, der einen Wert ermittelt. Beispielsweise können Sie den Ausdruck

=[Nettobetrag]*0,16

als *Steuerelementinhalt* angeben, um den Mehrwertsteuerbetrag, der sich auf Basis des Feldes *Nettobetrag* ergibt, anzeigen zu lassen. Es ist nicht möglich, den Wert, den der Ausdruck ergibt, unmittelbar in der dem Formular zugrundeliegenden Tabelle zu speichern. Um dies zu erreichen, müssen Sie ein Makro mit der Aktion *SetzenWert* oder eine entsprechende Ereignisprozedur schreiben und dieses Makro bzw. die Ereignisprozedur einem Formular- oder Steuerelementereignis (z.B. *Nach Aktualisierung*) zuordnen.

16.2 Bezeichnungsfeld

Bild 16.3: Zwei Bezeichnungsfelder in der Formularansicht.

Bezeichnungsfelder dienen zur allgemeinen Beschriftung oder zur speziellen Beschriftung von Steuerelementen. Im letzteren Falle empfiehlt es sich im allgemeinen, das Bezeichnungsfeld an das Steuerelement, z.B. an ein Textfeld oder Kontrollkästchen, zu binden. Standardmäßig fügt Access einem neu erstellten Steuerelement ein Bezeichnungsfeld hinzu und bindet dieses an das Steuerelement, so daß beide beispielsweise zusammen verschoben werden können. Sie können diese Voreinstellung für jedes einzelne Steuerelement aufheben, indem Sie seine Standardeigenschaft *Mit Bezeichnungsfeld* auf den Wert *Nein* einstellen. Wie Sie die Standardeigenschaft eines Steuerelements ändern, erfahren Sie in Kap. 15, *Formulare: Grundlagen*, Punkt 15.7.2, *Standardeigenschaften von Steuerelementen festlegen*.

Bezeichnungsfeld einem Steuerelement nachträglich zuordnen

Es ist auch möglich, einem Steuerelement nachträglich ein Bezeichnungsfeld zuzuordnen. Gehen Sie dazu folgendermaßen vor:

- Markieren Sie das zuzuordnende Bezeichnungsfeld. (Sie müßten es zuvor ggf. mit der Toolbox neu erstellen.)
- Drücken Sie die Tastenkombination Strg+X, oder wählen Sie den Befehl *Ausschneiden* aus dem Menü *Bearbeiten*.
- Markieren Sie das Steuerelement, dem das Bezeichnungsfeld zugeordnet werden soll.
- Drücken Sie die Tastenkombination Strg+V, oder wählen Sie den Befehl *Einfügen* aus dem Menü *Bearbeiten*.

16.3 Kontrollkästchen, Optionsfeld und Umschaltfläche zur Darstellung von Ja/Nein-Werten

	Ja	Nein
Kontrollkästchen	Erledigt	Erledigt
Optionsfeld	Erledigt	Erledigt
Umschaltfläche	Erledigt	Erledigt

Bild 16.4: Steuerelemente »Kontrollkästchen«, »Optionsfeld« und »Umschaltfläche« im Zustand »Ja« und »Nein«.

Jeder der drei folgenden Typen von *Ja/Nein*-Steuerelementen kann eigenständig oder als Teil einer Gruppe gleichartiger Steuerelemente in einer Optionsgruppe verwendet werden. Die Funktionsweise ist für die beiden Situationen unterschiedlich. In diesem Punkt wird beschrieben, wie Sie mit eigenständigen Kontrollkästchen, Optionsfeldern und Umschaltflächen umgehen können. Die Funktionsweise dieser Steuerelemente als Teil einer Optionsgruppe wird im nächsten Abschnitt, *Optionsgruppe und Optionsfelder*, erklärt.

Funktionsweise der *Ja/Nein-Steuerelemente*

Die Steuerelemente *Kontrollkästchen*, *Optionsfeld* und *Umschaltfläche* sind funktionell völlig gleichwertig. Sie ermöglichen die Anzeige und Bearbeitung von *Ja/Nein*-Werten. Der Unterschied zwischen den drei Steuerelementen ist lediglich visuell. Sie geben den Wert *Ja* (bzw. *-1* oder *Wahr*) aus, wenn sie eingeschaltet bzw. gedrückt sind, und *Nein* (bzw. *0* oder *Falsch*), wenn sie nicht eingeschaltet bzw. nicht gedrückt sind. Einem Kontrollkästchen und einem Optionsfeld kann ein Bezeichnungsfeld zugeordnet werden, einer Umschaltfläche dagegen nicht. Statt dessen kann die Umschaltfläche selbst einen Bezeichnungstext tragen, vgl. Bild 16.4.

Ja/Nein-Steuerelement erstellen

Um eines der drei genannten Steuerelemente zu erstellen, verfahren Sie wie folgt:

- Klicken Sie in der Toolbox auf die entsprechende Symbol-Schaltfläche für *Umschaltfläche*, *Optionsfeld* oder *Kontrollkästchen* (links nebenstehend in dieser Reihenfolge von oben nach unten wiedergegeben).
- Ziehen Sie dann aus der Feldliste ein Feld mit dem Datentyp *Ja/Nein* in das Formular, um ein *gebundenes* Kontrollkästchen, Optionsfeld oder eine gebundene Umschaltfläche zu erstellen; es wird die Art von Steuerelement erstellt, die in der Toolbox gedrückt ist, während Sie das Feld aus der Feldliste ziehen.

  Oder:

- Ziehen Sie ein Rechteck im Formular, um ein entsprechendes *ungebundenes* Steuerelement zu erstellen. Selbstverständlich können Sie ein zunächst als ungebunden erstelltes *Ja/Nein*-Steuerelement nachträglich an ein Feld einer Tabelle/Abfrage binden, indem Sie dies für die Eigenschaft *Steuerelementinhalt* angeben.

Umschaltfläche mit einem Bild versehen

Sie können eine Umschaltfläche mit einem Bild versehen. Das Verfahren dazu ist dasselbe wie bei einer Registerkarte, vgl. daher im einzelnen weiter unten in diesem Kapitel, Punkt 16.8, *Register,* und dort den Punkt *Beschriftung einer Registerkarte mit einem Bild versehen*.

Wenn Sie eine Leiste mit lauter Umschaltflächen, jede mit einem Bild versehen, erstellen, sieht diese aus wie eine Symbolleiste. Wenn Sie das Gedrücktsein bzw. Nicht-Gedrücktsein mit entsprechenden Makros oder VBA-Prozeduren verbinden, haben Sie auch die Funktion einer Symbolleiste. Ein Beispiel einer derartigen Zusammenstellung von Umschaltflächen finden Sie in den Formularen *Adressen (VBA-Steuerung)* sowie *Adressen (Makro-Steuerung)* in der Datenbank *Adressen.mdb* von der Begleit-CD-ROM, vgl. Bild 16.5.

Bild 16.5: Formular »Adressen (VBA-Steuerung)« der Datenbank »Adressen.mdb« von der Begleit-CD-ROM: Mit den Umschaltflächen mit den Buchstaben A – Z kann ein Filter für Firmen bzw. Namen eingeschaltet werden.

16.4 Optionsgruppe und Optionsfelder

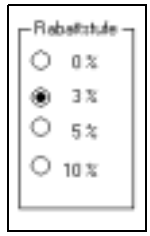


Bild 16.6: Optionsgruppe mit runden Optionsfeldern (auch »Radio-Buttons« genannt)

Eine Optionsgruppe eignet sich gut, um eine Auswahl aus einer beschränkten Zahl von alternativen Wahlmöglichkeiten anzuzeigen oder vorzunehmen. Bild 16.6 zeigt z.B. eine Optionsgruppe mit vier Wahlmöglichkeiten für Rabattstufen.

Funktionsweise einer Optionsgruppe

Eine Optionsgruppe besteht aus einem *Optionsgruppenfeld* und zwei oder mehr *Optionsfeldern*. Jedes der Optionsfelder besitzt einen Optionswert, der über die Eigenschaft *Optionswert* eingestellt werden kann. Im allgemeinen wird für das erste Optionsfeld einer Optionsgruppe der Wert 1 genommen, für das zweite der Wert 2 etc. Das ausgewählte Optionsfeld übergibt seinen Optionswert an das Optionsgruppenfeld. Für das Optionsgruppenfeld im obigen Bild 16.6 gilt beispielsweise: Sofern das Optionsfeld mit der Bezeichnung 3 % den Optionswert 2 hat, besitzt das Steuerelement *Optionsgruppe* im abgebildeten Zustand den Wert 2. Die Bezeichnungstexte *0 %*, *3 %* etc. haben formal nichts mit dem Optionswert des zugehörigen Optionsfeldes zu tun. Wenn eine Optionsgruppe an ein Tabellenfeld gebunden ist, übergibt sie ihren Wert, also den Optionswert des gewählten Optionsfeldes, an das Tabellenfeld. Anders ausgedrückt: In der Tabelle werden keinesfalls Texte, mit denen die Optionsfelder beschriftet sind, gespeichert, sondern stets nur deren Optionswerte.

Sie mögen es als unbefriedigend empfinden, in einer Tabelle nur codierte Werte statt ihrer zugehörigen Klartexte (im obigen Beispiel: Zahlen 1 bis 4 statt der Texte *0 %*, *3 %* etc.) zu speichern, weil diese stets nur in Verbindung mit einer Codeliste interpretierbar sind. Beachten Sie dabei bitte die beiden folgenden Gesichtspunkte:

- Das Speichern von Zahlenwerten ist wesentlich rationeller als dasjenige langer Texte. Sofern, wie im obigen Beispiel der Rabattstaffel zutreffend, die zu speichernde Information auch als Zahlenwert interpretierbar ist, empfiehlt es sich, die Optionswerte der einzelnen Optionsfelder entsprechend einzustellen. So könnten Sie die Optionswerte für die vier Rabattstufen auf die Größen 0, 3, 5 und 10 einstellen. (Als Optionswerte können nur ganze Zahlen angegeben werden, so daß die Prozentwerte nicht, was eigentlich sinnvoller wäre, als Dezimalbrüche angegeben werden können.)
- Wenn Sie dennoch Klartexte statt der Optionswerte speichern wollen, sollten Sie ein Listen- oder Kombinationsfeld verwenden, weil diese insoweit viel leichter einzurichten sind. Um nämlich die Optionswerte einer Optionsgruppe in Klartext umzuformen und in der Tabelle speichern zu lassen, müßten Sie Makros oder VBA-Prozeduren verwenden.

Optionsgruppe mit dem Steuerelement-Assistenten erstellen

Eine Optionsgruppe »freihändig« zu erstellen, ist zumindest für wenig Erfahrene etwas kompliziert. Daher empfiehlt sich der Einsatz des Steuerelement-Assistenten.

Beispiel

Die oben in Bild 16.6 dargestellte Optionsgruppe soll mit Hilfe des Steuerelement-Assistenten erstellt werden. Die Optionsgruppe soll an das Feld *Rabattstufe* gebunden werden. Als Werte für die vier Optionsfelder sollen die ganzen Zahlen 0, 3, 5 und 10 eingestellt werden.

Gehen Sie folgendermaßen vor, um die Optionsgruppe mit dem Steuerelement-Assistenten zu erstellen:

- Blenden Sie ggf. die Toolbox ein, und klicken Sie ggf. darin auf die Symbol-Schaltfläche *Steuerelement-Assistenten*, so daß diese gedrückt erscheint.

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Optionsgruppe*, und ziehen Sie dann mit dem Mauszeiger ein Rechteck an der gewünschten Position im Formular.

Bild 16.7: Erstes Dialogfeld des Optionsgruppen-Assistenten

Tragen Sie die vier Bezeichnungen ein, mit denen die vier Optionsfelder versehen werden sollen, vgl. Bild 16.7. Der Optionsgruppen-Assistent wird Sie in einem späteren Dialogfeld fragen, welche Art von Optionsfeldern Sie wünschen. Für runde Optionsfelder und Kontrollkästchen werden die hier angegebenen vier Bezeichnungen neben den Optionen, für Umschaltflächen als Beschriftungen der Flächen selbst erscheinen. Klicken Sie nach der Eingabearbeit auf die Schaltfläche *Weiter* >.

- Wählen Sie im zweiten Dialogfeld die Einstellung, die in Bild 16.8 wiedergegeben ist. Damit wird der Standardwert für das Optionsgruppenfeld festgelegt, so daß für jeden neuen Datensatz standardmäßig die Option *0 %* gewählt erscheinen wird. Klicken Sie nach der Eingabearbeit auf die Schaltfläche *Weiter* >.

Bild 16.8: Zweites Dialogfeld des Optionsgruppen-Assistenten

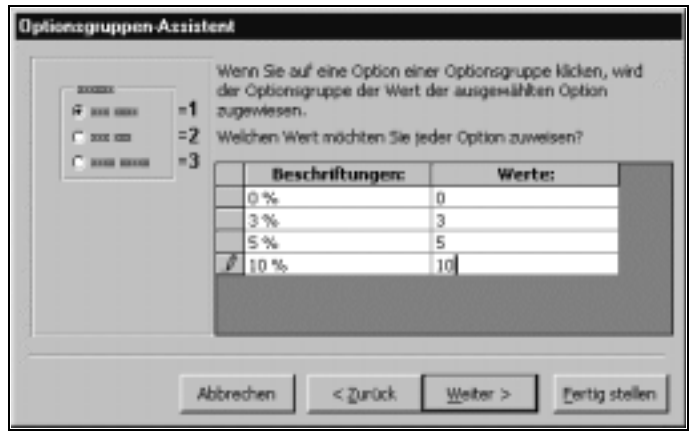


Bild 16.9: Drittes Dialogfeld des Optionsgruppen-Assistenten

- Tragen Sie in der Spalte *Werte* des dritten Dialogfeldes die Zahlen 0, 3, 5 und 10 ein, wie in Bild 16.9 dargestellt. Damit weisen Sie der Eigenschaft *Optionswert* der vier Optionsfelder jeweils die angegebene Zahl als Wert zu. Klicken Sie nach der Eingabearbeit auf die Schaltfläche *Weiter* >.
- Wählen Sie im vierten Dialogfeld die Option *Den Wert hier speichern*, schlagen Sie das Dropdown-Listenfeld mit den Feldnamen der dem Formular zugrunde liegenden Tabelle/Abfrage auf, und wählen Sie darin das Feld *Rabattstufe*. Damit binden Sie die Optionsgruppe an dieses Feld, so daß die jeweilige Optionswahl als Optionswert in diesem Feld gespeichert wird. Klicken Sie auf die Schaltfläche *Weiter* >.
- Wählen Sie im fünften Dialogfeld (vgl. Bild 16.11) die Art der in die Optionsgruppe aufzunehmenden Optionsschaltflächen und den Stil. Klicken Sie nach der Eingabearbeit auf die Schaltfläche *Weiter* >.

Bild 16.10: Viertes Dialogfeld des Optionsgruppen-Assistenten

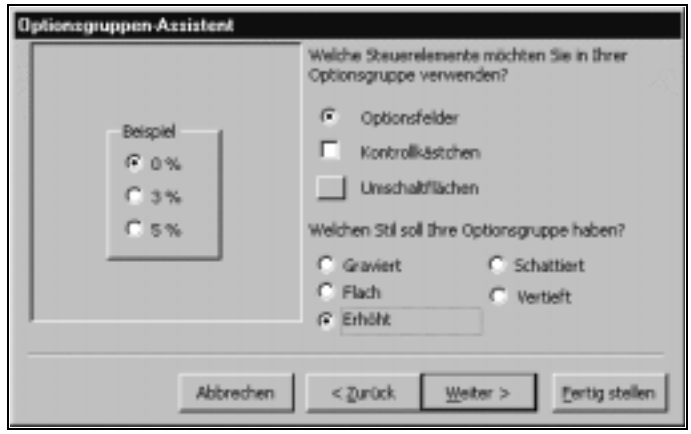


Bild 16.11: Fünftes Dialogfeld des Optionsgruppen-Assistenten

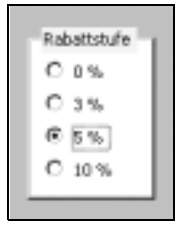


Bild 16.12: Fertige Optionsgruppe mit runden Optionsfeldern in der Formulansicht

- Geben Sie im sechsten Dialogfeld (hier nicht als Bild wiedergegeben) an, welche Bezeichnung das Optionsgruppenfeld bekommen soll: Geben Sie die Bezeichnung *Rabattstufe* an. Klicken Sie schließlich auf die Schaltfläche *Fertigstellen*. Dann fügt der Optionsgruppen-Assistent die fertige

Optionsgruppe in die Entwurfsansicht des Formulars ein, wie dies in Bild 16.12 zu sehen ist. Dort ist die entsprechende Darstellung in der Formularansicht wiedergegeben.

Optionsgruppe ohne den Steuerelement-Assistenten erstellen

Wenn Sie die Optionsgruppe aus dem letzten Beispiel freihändig, d.h. ohne den Steuerelement-Assistenten, erstellen wollen, gehen Sie wie folgt vor:

- Blenden Sie ggf. die Toolbox ein, und klicken Sie ggf. darin auf die Symbol-Schaltfläche *Steuerelement-Assistenten*, so daß diese nicht gedrückt erscheint.

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Optionsgruppe*.
- Ziehen Sie aus der Feldliste das Feld *Rabattstufe*, an das die Optionsgruppe gebunden werden soll, an die Stelle, an der die Optionsgruppe im Formular angezeigt werden soll. (Sie können die Feldliste auch in dieser Situation ggf. einblenden.) Wenn Sie die Maustaste loslassen, wird die Optionsgruppe als Rahmen, beschriftet mit dem zugehörigen Feldnamen, erstellt.
- Doppelklicken Sie in der Toolbox auf eine der Symbol-Schaltflächen *Kontrollkästchen*, *Optionsfeld* oder *Umschaltfläche*. Mit dem Doppelklicken statt des einfachen Klickens erreichen Sie, daß Sie mehrere Optionsfelder (oder worauf immer Sie doppelgeklickt haben) nacheinander einfügen können, ohne jedesmal erneut auf die entsprechende Symbol-Schaltfläche in der Toolbox klicken zu müssen.
- Klicken Sie im Optionsgruppenrahmen auf die Stelle, an der Sie das erste Optionsfeld einfügen wollen. Dann wird das erste Optionsfeld eingefügt.
- Wiederholen Sie den vorangehenden Schritt für jedes weitere benötigte Optionsfeld.
- Jedem Optionsfeld ist ein Bezeichnungsfeld mit einer Standardbeschriftung zugeordnet. Überschreiben Sie die Texte dieser Felder entsprechend Ihren Bedürfnissen, für das obige Beispiel also mit *0 %*, *3 %* etc. für die Optionsfelder.
- Richten Sie die einzelnen Optionsfelder aus.

Eine ungebundene Optionsgruppe erzeugen Sie ganz entsprechend, wie in den vorigen Schritten beschrieben, jedoch ziehen Sie nach dem Klicken auf die Symbol-Schaltfläche *Optionsgruppe* in der Toolbox kein Feld aus der Feldliste, sondern Sie ziehen direkt im Formular ein Rechteck.

Für eine bestehende Optionsgruppe können Sie die Eigenschaften des Steuerelements *Optionsgruppe* und der einzelnen Optionsfelder getrennt einstellen. Insbesondere sollten Sie überlegen, ob Sie die Optionswerte der Optionsfelder, die Access entsprechend der Reihenfolge, in der diese erstellt wurden, auf die Werte 1, 2, 3 ... festsetzt, beibehalten oder auf andere ganzzahlige Werte einstellen.

Optionsgruppe zur Steuerung von Aktionsfolgen

Eine Optionsgruppe – gebunden oder ungebunden – eignet sich gut zur Steuerung von Aktionen, die als Folge von Ereignissen eintreten sollen. Solche Aktionen können Sie mit Makros oder mit VBA-Prozeduren kontrollieren. Nehmen Sie beispielsweise an, ein Formular enthält eine Befehlsschaltfläche, nach deren Wahl kontextabhängig eines von drei weiteren möglichen Formularen geöffnet werden soll. In einem Makro können Sie dann mit jedem der drei möglichen Optionswerte einer Optionsgruppe eine Bedingung verbinden und, abhängig von deren Zutreffen, das entsprechende Formular mit der Aktion *ÖffnenFormular* öffnen. In einer Ereignisprozedur würden Sie die Formulare entsprechend mit einer *Select Case*-Gruppe öffnen.

16.5 Listenfeld und Kombinationsfeld

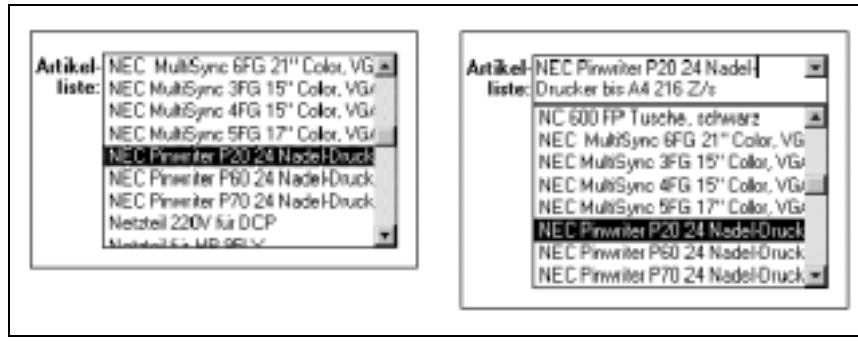


Bild 16.13: Listenfeld (links) und aufgeschlagenes Kombinationsfeld (rechts), jeweils mit derselben Datensatzherkunft

Listen- oder Kombinationsfelder ermöglichen eine Auswahl aus einer Liste von Werten. Die Liste eines Listenfeldes wird immer aufgeschlagen angezeigt, diejenige eines Kombinationsfeldes muß als Dropdown-Listenfeld aufgeschlagen werden. Die beiden Steuerelemente sind in ihrer Art und Wirkung ansonsten weitgehend gleich, so daß sie hier zusammen besprochen werden. Auf die Unterschiede wird weiter unten in diesem Abschnitt eingegangen, vgl. Punkt *Unterschiede von Listen- und Kombinationsfeldern*.

Funktionsweise von Listen- und Kombinationsfeldern

Listen- und Kombinationsfelder bieten eine prinzipiell beliebig lange Liste von Zeilen mit Einträgen an, aus der Sie einen Eintrag auswählen können. Wenn ein Listen- oder Kombinationsfeld an ein Tabellenfeld gebunden ist, gibt es den gewählten Eintrag an das Feld weiter, so daß dieser in der Tabelle gespeichert wird. Ein ungebundenes Listen- oder Kombinationsfeld

besitzt den gewählten Eintrag als Wert, so daß man diesen mit einem Ausdruck, einem Makro oder einer VBA-Prozedur auslesen und ggf. weiterverarbeiten kann.

Bild 16.14: Eigenschaftenfenster mit den für ein Listen- oder Kombinationsfeld wichtigsten Eigenschaften

Der Listeninhalt von Listen- und Kombinationsfeldern wird durch die beiden Eigenschaften *Herkunftstyp* und *Datensatzherkunft* bestimmt, die miteinander korrespondieren:

Herkunftstyp

Diese Eigenschaft gibt an, von welcher Art die Datenherkunft ist. Sie können in dem Dropdown-Listenfeld aus den Einträgen *Tabelle/Abfrage*, *Wertliste* sowie *Feldliste* wählen.

Datensatzherkunft

Die mögliche und sinnvolle Einstellung dieser Eigenschaft hängt davon ab, welcher Herkunftstyp gewählt ist. Es gelten die folgenden Zusammenhänge:

Gewählter Herkunftstyp	Angabe für die Datensatzherkunft
Tabelle/Abfrage	Name einer Tabelle oder Abfrage oder ein SQL-Ausdruck, der Datensätze zurückgibt. Die Liste wird dann aus dem ersten Feld der Tabelle bzw. des Abfrageergebnisses oder, bei mehrspaltigen Listen- und Kombinationsfeldern, aus den ersten Feldern gebildet.
Wertliste	Liste von beliebigen Werten (z.B. Texte, Zahlen), wobei die Werte jeweils durch ein Semikolon gegeneinander abgetrennt sind.
Feldliste	Name einer Tabelle oder Abfrage. Die Liste wird dann aus den Feldnamen der Tabelle/Abfrage gebildet.

Mehrspaltige Listen- und Kombinationsfelder

Wenn Sie die Eigenschaft *Spaltenanzahl* eines Listen- oder Kombinationsfeldes auf einen Wert einstellen, der größer als 1 ist, erhalten Sie eine mehrspaltige Liste. Bei einer Tabelle/Abfrage als Herkunftstyp geben dann die ersten n Felder die Spalteninhalte für ein n-spaltiges Listen- oder Kombinationsfeld an. Wenn Sie beispielsweise eine Abfrage mit den drei im Abfrageergebnis am weitesten links stehenden Feldern *Artikelname*, *Artikelnummer* und *Einzelpreis* als Datensatzherkunft angegeben und die Spaltenzahl auf den Wert 3 eingestellt haben, gibt jede Zeile des entsprechenden Listen- oder Kombinationsfeldes von links nach rechts den Artikelnamen, die Artikelnummer und den Einzelpreis wieder.

Obwohl ein Listen- oder Kombinationsfeld mehrere Spalten darstellen kann, *gibt es stets nur einen Wert aus*. Daher müssen Sie angeben, aus welcher Spalte dieser *gebundene Wert* stammen soll. Dies geschieht mit der Eigenschaft *Gebundene Spalte*, für die eine ganze Zahl zwischen 1 und der Spaltenanzahl angegeben wird. Wenn Sie im Beispiel des letzten Absatzes die Eigenschaft *Gebundene Spalte* auf den Wert 2 einstellen, gibt das Listen- oder Kombinationsfeld den Wert der Artikelnummer aus.

Unterschiede von Listen- und Kombinationsfeldern

Ein Listen- und ein Kombinationsfeld unterscheiden sich in zwei Punkten, von denen der erste nur die Anzeigeform im Formular, der zweite jedoch die Funktionsweise betrifft:

- Die Liste eines Kombinationsfeldes wird erst angezeigt, wenn Sie sein Dropdown-Listenfeld aufschlagen. Die Liste eines Listenfeldes wird dagegen immer mit der Anzahl von Zeilen, auf die die Eigenschaft *Zeilenanzahl* eingestellt ist, angezeigt.
- Ein Kombinationsfeld enthält einerseits eine Wertliste, andererseits ein Textfeld, das den Kopf des Kombinationsfeldes bildet. Letzteres trifft für ein Listenfeld nicht zu. Wenn Sie das Dropdown-Listenfeld eines Kombinationsfeldes aufschlagen und eine Zeile daraus wählen, wird der Inhalt der gewählten Zeile in das Textfeld im Kopf übertragen. Sofern Sie die Eigenschaft *Nur Listeneinträge* auf *Nein* eingestellt haben, können Sie in das Textfeld im Kopf des Kombinationsfeldes eine Zeichenfolge eingeben, der kein Listeneintrag entspricht; andernfalls werden nur Listenwerte akzeptiert. Wenn die Eigenschaft *Nur Listeneinträge* auf *Ja* eingestellt ist, besteht kein funktioneller Unterschied zwischen einem Listenfeld und einem Kombinationsfeld. Das Letztere hat dann den Oberflächenvorteil, im Formular weniger Platz zu beanspruchen. Dieser Vorteil wird mit der kleinen Zusatzmühe erkauft, vor jeder Verwendung das Dropdown-Listenfeld aufschlagen zu müssen.

Spaltenbreiten der Liste einstellen und Spalten ausblenden

Die Breite jeder einzelnen Spalte wird mit der Eigenschaft *Spaltenbreiten* eingestellt: Geben Sie als Einstellung eine Liste mit Zahlenwerten, jeweils durch ein Semikolon voneinander getrennt, für die Spaltenbreiten ein. Der erste Wert bezieht sich dann auf die erste Spalte etc. Um beispielsweise in einem Listen- oder Kombinationsfeld mit einer dreispaltigen Liste zu bewirken, daß die erste Spalte 2 cm, die zweite 1 cm und die dritte 0,5 cm breit wird, geben Sie für die Eigenschaft *Spaltenbreiten* die Zeichenfolge *2;1;0,5* ein; Access gibt diese Angabe in der Form *2cm;1cm;0,5cm* wieder. Achten Sie im übrigen darauf, daß die Summe der Spaltenbreiten auf die Breite des gesamten Listen- oder Kombinationsfeldes abgestimmt ist, damit alle Spalten auch angezeigt werden können bzw. kein überschüssiger Platz dargestellt wird.

Wenn Sie für einzelne Spalten den Wert 0 angeben, werden diese nicht angezeigt. Sie können sich dennoch ohne Einschränkung auf ausgeblendete Spalten beziehen. Nehmen Sie beispielsweise an, ein Listen- oder Kombinationsfeld solle dem Benutzer eine Auswahl aus einer Liste ermöglichen, die ihm den Artikelnamen sowie den zugehörigen Einzelpreis anzeigt. Das Listen- oder Kombinationsfeld soll jedoch die zugehörige Artikelnummer und nicht den Artikelnamen oder den Einzelpreis ausgeben (z.B. an ein Tabellenfeld, an das es gebunden ist). In diesem Falle müßten Sie das Listen- oder Kombinationsfeld mit einer dreispaltigen Liste versehen, deren drei Spalten den Artikelnamen, die Artikelnummer und den Einzelpreis wiedergeben. Die Spaltenbreite für die Artikelnummer müßten Sie auf 0 einstellen, und diese Spalte müßte gleichzeitig die *Gebundene Spalte* sein.

Welcher Wert wird im Textfeld eines Kombinationsfeldes angezeigt?

Im Textfeld eines Kombinationsfeldes wird der zuletzt ausgewählte Wert angezeigt. In einem mehrspaltigen Kombinationsfeld stammt dieser Wert aus der am weitesten links stehenden, nicht ausgeblendeten Spalte. Wenn Sie das Dropdown-Listenfeld eines Kombinationsfeldes aufschlagen, werden stets alle nicht ausgeblendeten Spalten, soweit sie in die insgesamt verfügbare Listenbreite passen, wiedergegeben.

Wie Sie sich auf die nicht gebundenen Spalten der Liste beziehen

Sie können den Wert der gebundenen Spalte ermitteln, indem Sie sich auf das Listen- oder Kombinationsfeld beziehen. Wenn beispielsweise die zweite Spalte gebunden ist und diese Spalte die Artikelnummer anzeigt, erhalten Sie die Artikelnummer der aktuellen Wahl in einem Listenfeld mit dem Steuerelementnamen *Artikelliste* durch Bezugnahme auf dieses Feld. So würde ein berechnetes Textfeld mit dem Ausdruck *=[Artikelliste]* als Steuerelementinhalt die jeweils aktuelle Artikelnummer wiedergeben.

Auf jede Spalte, also auch auf eine nicht gebundene, können Sie sich über die Listen- bzw. Kombinationsfeld-Eigenschaft *Column* beziehen; diese Eigenschaft ist nur in Ausdrücken ansprechbar und wird nicht im Eigenschaftenfenster angezeigt. Sie nehmen dabei Bezug auf eine bestimmte Spalte, indem Sie deren Positionsnummer in der Liste angeben. Aus der Sicht der Eigenschaft *Column* sind die Spalten mit den laufenden Nummern 0, 1, 2 ... numeriert. (*Achtung*: Aus der Sicht des Eigenschaftenfensters beginnt die Spaltennumerierung bei 1 und nicht bei 0!) Nehmen Sie an, es solle der Wert der dritten Spalte des Listenfeldes mit dem Steuerelementnamen *Artikelliste* mit einem Ausdruck ermittelt werden. Dann müßte dieser Ausdruck folgendermaßen lauten:

```
=[Artikelliste].Column(2)
```

Listen- oder Kombinationsfeld mit dem Steuerelement-Assistenten erstellen

Zumindest für wenig Erfahrene empfiehlt sich zum Erstellen von Listen- und Kombinationsfeldern der Einsatz des Steuerelement-Assistenten. Dessen Arbeitsweise wurde detailliert für das Erstellen einer Optionsgruppe demonstriert, vgl. oben Punkt 16.4, *Optionsgruppe und Optionsfelder*, Unterpunkt *Optionsgruppe mit dem Steuerelement-Assistenten erstellen*. Aus diesem Grunde beschränke ich mich im folgenden auf eine skizzenhafte Darstellung. Der Vorteil des Steuerelement-Assistenten für Listen- und Kombinationsfelder zeigt sich vor allem bei der Festlegung der Spaltenbreiten, die Sie interaktiv grafisch vornehmen können, und darin, daß eine SQL-SELECT-Anweisung für die Eigenschaft *Datensatzherkunft* erzeugt wird, die für eine Auswahl von in der Ursprungstabelle nicht zusammenliegenden Feldern unerläßlich ist, es sei denn, Sie haben für das Listen- oder Kombinationsfeld bereits eine entsprechende Abfrage erstellt und unter einem Namen gespeichert.

Gehen Sie folgendermaßen vor, um ein Listen- oder Kombinationsfeld mit dem Steuerelement-Assistenten zu erstellen:

- Blenden Sie ggf. die Toolbox ein, und klicken Sie ggf. darin auf die Symbol-Schaltfläche *Steuerelement-Assistenten*, so daß diese gedrückt erscheint.

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Kombinationsfeld* bzw. *Listenfeld*, und ziehen Sie dann mit dem Mauszeiger ein Rechteck an der gewünschten Position im Formular.
- Folgen Sie den weiteren Schritten des Steuerelement-Assistenten.

Listen- oder Kombinationsfeld ohne den Steuerelement-Assistenten erstellen

Um ein Listen- oder Kombinationsfeld ohne den Steuerelement-Assistenten zu erstellen, gehen Sie folgendermaßen vor:

- Blenden Sie ggf. die Toolbox ein, und klicken Sie ggf. darin auf die Symbol-Schaltfläche *Steuerelement-Assistenten*, so daß diese nicht gedrückt erscheint.
- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Listenfeld* bzw. *Kombinationsfeld*.
- Ziehen Sie das Feld, an welches das Listen- oder Kombinationsfeld gebunden werden soll, aus der Feldliste in das Formular. Für ein ungebundenes Listen- oder Kombinationsfeld ziehen Sie im Formular an geeigneter Stelle ein Rechteck.
- Stellen Sie für das Listen- oder Kombinationsfeld die Eigenschaft *Herkunftstyp* auf den zutreffenden Typ ein, z.B. *Tabelle/Abfrage* oder *Wertliste*.
- Geben Sie für die Eigenschaft *Datensatzherkunft* den Namen einer Tabelle oder Abfrage oder eine SQL-SELECT-Anweisung an, wenn Sie als Herkunftstyp *Tabelle/Abfrage* gewählt haben; im Falle einer Wertliste als Herkunftstyp geben Sie die Wertliste ein, die Werte gegeneinander durch je ein Semikolon getrennt.
- Bestimmen Sie mit der Eigenschaft *Spaltenanzahl* durch Angabe einer ganzen Zahl, wie viele Spalten die Liste anzeigen soll. Beachten Sie, daß diese Angabe mit der Datensatzherkunft korrespondieren muß.
- Falls die Liste Spaltenüberschriften tragen soll, stellen Sie die Eigenschaft *Spaltenüberschriften* auf *Ja* ein. Wenn die Liste auf einer Tabelle/Abfrage beruht, werden die Feldnamen als Überschriften genommen. Bei einer Wertliste werden die ersten Werte der Liste als Überschriften verwendet; beispielsweise würde eine zweispaltige Liste dann die ersten beiden Einträge der Wertliste als Überschriften verwenden.
- Geben Sie eine ganze Zahl für die *Gebundene Spalte* an; Access trägt hier standardmäßig den Wert 1 ein.
- Stellen Sie die Breiten der Spalten mit der Eigenschaft *Spaltenbreiten* ein.
- Geben Sie bei einem Kombinationsfeld für die Eigenschaft *Nur Listeneinträge Ja* oder *Nein* an.
- Stellen Sie ggf. weitere Eigenschaften des Listen- oder Kombinationsfeldes ein. In Betracht kommen vor allem die Eigenschaften *Listenbreite* (nur Kombinationsfeld) sowie *Breite* und *Höhe* (letztere natürlich auch durch Ziehen der Feldränder einstellbar).

Beispiele

Einspaltiges Listenfeld mit Wertliste: Erstellen Sie ein einspaltiges Listenfeld mit einer Wertliste, welche die Namen der fünf norddeutschen Bundesländer wiedergibt. Binden Sie das Listenfeld an das Tabellenfeld *Bundesland*. Gehen Sie wie folgt vor, und orientieren Sie sich für das Ergebnis an Bild 16.15:

Bild 16.15: Einspaltiges Listenfeld

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Listenfeld*.
- Öffnen Sie ggf. die Feldliste und ziehen Sie das Feld *Bundesland* in das Formular. (Alternativ für ein ungebundenes Listenfeld: Ziehen Sie ein Rechteck im Formular.)
- Stellen Sie im Eigenschaftenfenster die Eigenschaft *Herkunftstyp* auf *Wertliste* ein.
- Geben Sie im Eigenschaftenfenster für die Eigenschaft *Datensatzherkunft* die Wertliste in folgender Weise ein:

  Schleswig-Holstein;Mecklenburg-Vorpommern;Hamburg;Niedersachsen;Bremen
- Beschriften Sie das dem Listenfeld zugeordnete Bezeichnungsfeld mit *Bundesland*.
- Justieren Sie ggf. die Größe und Position für Listenfeld und Bezeichnungsfeld.

Dreispaltiges ungebundenes Kombinationsfeld mit einer Abfrage als Datensatzherkunft

Die Datenbank *Auftrag.mdb*, die Sie auf der Begleit-CD-ROM zu diesem Buch finden, enthält das Formular *UFAngebote*, welches als Unterformular im Hauptformular *Angebote bearbeiten* fungiert. Das Formular *UFAngebote* enthält ein dreispaltiges ungebundenes Kombinationsfeld mit dem Steuerelementnamen *Artikelliste*. Die Datensatzherkunft der Liste ist die Abfrage *Artikelliste*, deren Abfrageergebnis die drei Felder *Artikelname*, *ArtikelNr* sowie *Einzelpreis* (in dieser Reihenfolge) enthält. Angezeigt wird nur die erste Spalte, und zwar mit einer Spaltenbreite von 4,82 cm. Das Eigenschaftenfenster gibt die wesentlichen Eigenschaften folgendermaßen wieder, vgl. Bild 16.16.

Das aufgeschlagene Kombinationsfeld ist zu Beginn dieses Abschnitts wiedergegeben, vgl. Bild 16.13. Um das angegebene Kombinationsfeld zu erstellen, müßten Sie folgendermaßen vorgehen:

- Öffnen Sie ggf. die Datenbank *Auftrag.mdb*.
- Öffnen Sie das Formular *UFAngebote* in der Entwurfsansicht, und löschen Sie das vorhandene Kombinationsfeld *Artikelliste*.

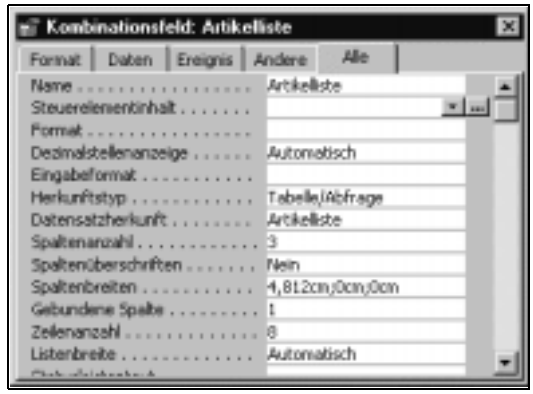


Bild 16.16: Eigenschaften für ein Kombinationsfeld

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Kombinationsfeld*, und ziehen Sie ein Rechteck an der Stelle im Formular, an der das Kombinationsfeld positioniert werden soll (bzw. war).
- Beschriften Sie das Bezeichnungsfeld mit der Bezeichnung *Artikelliste*.
- Belassen Sie die Einstellung der Eigenschaft *Herkunftstyp* bei *Tabelle/Abfrage*.
- Klicken Sie im Eigenschaftenfenster in das Eingabefeld zur Eigenschaft *Datensatzherkunft*, schlagen Sie dort das Dropdown-Listenfeld auf, und wählen Sie den Eintrag *Artikelliste*.
- Stellen Sie die Eigenschaft *Spaltenanzahl* auf den Wert 3 ein.
- Geben Sie für die *Spaltenbreiten* die folgende Zahlenliste an: *4,82;0;0*.
- Wechseln Sie in die Formularansicht, und schlagen Sie das Dropdown-Listenfeld des neu erstellten Kombinationsfeldes auf. Es sollte etwa so aussehen, wie am Beginn dieses Abschnitts in Bild 16.13 wiedergegeben.

Anmerkung. Das Kombinationsfeld *Artikelliste* ist ungebunden. Damit die Auswahl in seiner Liste Konsequenzen hat, muß in diesem Falle durch Makros bzw. VBA-Prozeduren für entsprechende Aktionen und Operationen gesorgt werden. Im (Unter-)Formular *UFAngebote* der Datenbank *Auftrag.mdb* geschieht dies durch die benutzerdefinierte VBA-Funktion *Artikel-Übernehmen*, vgl. dort im Modul1. Diese Funktion ist an das Ereignis *Nach Aktualisierung* des Kombinationsfeldes *Artikelliste* im (Unter-)Formular *UFAngebote* gebunden. Die Funktion überträgt u.a. die drei aktuellen Spaltenwerte in die Felder *ArtikelBezeichnung*, *ArtikelNr* sowie *Einzelpreis* des Formulars *UFArtikelPositionen*, das selbst wiederum ein Unterformular zum Unterformular *UFAngebote* ist. VBA Funktionen werden ausführlich in Kap. 29, *VBA (Visual Basic for Applications): Grundlagen*, behandelt. Hier sei nur nachrichtlich der diesbezügliche Teil der angesprochenen Funktion ohne weitere Erklärung wiedergegeben:

```
Function ArtikelÜbernehmen ()
Dim FF1 As Form, FF2 As Form
Set FF1 = Forms![Angebote bearbeiten]![UFAngebote].Form
Set FF2 = FF1!UFArtikelPositionen.Form
FF2!ArtikelBezeichnung = FF1!Artikelliste.Column(0)
FF2!ArtikelNr = FF1!Artikelliste.Column(1)
FF2!Einzelpreis = FF1!Artikelliste.Column(2)
'...
End Function
```

Sofern Sie für ein Listen- oder Kombinationsfeld als Datensatzherkunft eine Abfrage angegeben haben, können Sie unangenehme Überraschungen erleben, wenn Sie den Abfrageentwurf nachträglich ändern: Access stellt in nicht immer vorhersagbarer Weise die Feldreihenfolge im Abfrageergebnis um, wenn Sie beispielsweise nachträglich nach einem Feld sortieren lassen und/oder zusätzliche Kriterien angeben. Dies kann für das Listen- oder Kombinationsfeld fatal sein, weil darin von bestimmten Feldpositionen im Abfrageergebnis ausgegangen wird. In der Access-Dokumentation wird nur darauf verwiesen, daß die Reihenfolge der Felder im Abfrageentwurf geändert wird, wenn Sie Felder zum Sortieren angeben. Dies betreffe aber nicht das Abfrageergebnis. Diese Annahme trifft jedoch nach meinen Erfahrungen und denen anderer Benutzer nicht immer zu. Sie sind daher gut beraten, wenn Sie nach einer Änderung am Abfrageentwurf das Abfrageergebnis bezüglich der Feldreihenfolge überprüfen und diese mit der im betreffenden Listen- oder Kombinationsfeld angenommenen vergleichen. Besser noch: Geben Sie statt einer gespeicherten Abfrage eine SQL-SELECT-Anweisung für die Eigenschaft *Datensatzherkunft* an.

Die Datenbasis eines Listen- oder Kombinationsfeldes, dessen Liste als Datensatzherkunft eine Tabelle/Abfrage oder einen entsprechenden SQL-Ausdruck besitzt, wird jeweils unmittelbar nach dem Öffnen des Formulars, in das es eingebunden ist, ermittelt. Erfolgen Datenänderungen an der dem Listen- oder Kombinationsfeld zugrunde liegenden Tabelle/Abfrage, während das Formular geöffnet ist, werden diese nicht automatisch auf das Listen- oder Kombinationsfeld übertragen. Sie können jedoch eine Aktualisierung der Datenbasis für das gesamte geöffnete Formular veranlassen; dann wird auch die Datenbasis des Listen- oder Kombinationsfeldes aktualisiert. Manuell erreichen Sie dies mit dem Befehl *Anzeige aktualisieren* aus dem Menü *Datensätze*. Noch sinnvoller erscheint ein Makro oder eine entsprechende VBA-Prozedur mit der Aktion *AktualisierenDaten* (*Requery*), auf die Sie eine Ereigniseigenschaft (z.B. *Beim Doppelklicken* oder *Beim Hingehen*) des Listen- oder Kombinationsfeldes einstellen.

Listen- oder Kombinationsfeld in einen anderen Steuerelementtyp ändern

Sie können ein vorhandenes Listenfeld in ein Kombinationsfeld ändern und umgekehrt. Jeder dieser beiden Steuerelementtypen läßt sich auch in ein Textfeld ändern. Bei der Umwandlung eines Listenfeldes in ein Kombinati-

onsfeld oder umgekehrt bleiben die Einstellungen für die Eigenschaften, welche die Daten betreffen (z.B. *Datensatzherkunft* oder *Gebundene Spalte*) erhalten, so daß Sie ggf. auch wieder eine Rückwandlung vornehmen können.

Gehen Sie folgendermaßen vor, um ein Listen- oder Kombinationsfeld in einen anderen Typ umzuwandeln:

- Markieren Sie das Listen- oder Kombinationsfeld in der Entwurfsansicht des Formulars.
- Wählen Sie den Befehl *Ändern zu* aus dem Menü *Format* und dann einen der angebotenen, kontextabhängigen Unterbefehle. Im Falle eines markierten Listenfeldes werden beispielsweise die beiden Unterbefehle *Textfeld* und *Kombinationsfeld* angeboten.
- Nehmen Sie ggf. Anpassungen an dem geänderten Steuerelement vor. Im allgemeinen werden diese nach einer Umwandlung eines Listenfeldes in ein Kombinationsfeld oder umgekehrt nur in sehr geringem Maße erforderlich sein.

16.6 Mit einem Listen- oder Kombinationsfeld einen Datensatz im Formular suchen

Sie können ein Listen- oder Kombinationsfeld dazu verwenden, in einem Formular nach einem Datensatz zu suchen und diesen zu aktivieren. Beispielsweise können Sie in der Liste eines Kombinationsfeldes Firma, Nachname, Vorname usw. einer Kundenliste anbieten und nach der Auswahl eines bestimmten Kunden im Kombinationsfeld dessen Datensatz im Formular aktivieren. Diese Aufgabe ist natürlich nur ein spezieller Einsatzzweck von Listen- oder Kombinationsfeldern, jedoch kommt diese Aufgabe in der Praxis so häufig vor, daß sie wiederum allgemeine Bedeutung hat. Aus diesem Grunde wird diesem Zweck ein eigener Punkt gewidmet.

Um in einem Formular mit einem Listen- oder Kombinationsfeld nach einem Datensatz zu suchen, muß u.a. ein Ereignismakro oder eine Ereignisprozedur erstellt werden. Dies wäre die Aufgabe späterer Kapitel dieses Buches. Jedoch können Sie sich die erforderliche Ereignisprozedur vollständig und zuverlässig vom Steuerelement-Assistenten erstellen lassen, so daß Sie die Aufgabe ohne Programmierkenntnisse bewältigen können. Diese sind – in geringem Maße – nur erforderlich, wenn Sie zusätzlich gewisse Anpassungen vornehmen wollen.

Aufgabe und Lösung an einem Beispiel

Die Lösung der Aufgabe soll an einem Beispiel gezeigt werden: In das Formular *Personal und Projekte* der Datenbank *Projekte.mdb* von der Begleit-CD-ROM soll ein Kombinationsfeld eingefügt werden, mit dessen Hilfe Sie

gezielt zu einem Datensatz blättern können, dessen Nachnamen und Vornamen Sie im Kombinationsfeld gewählt haben. Gehen Sie folgendermaßen vor:

- Öffnen Sie die Datenbank *Projekte.mdb*.
- Öffnen Sie das Formular *Personal und Projekte* in der Entwurfsansicht. In Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, Punkt 5.11, *Formular Personal und Projekte erstellen*, wurde gezeigt, wie dieses Formular erstellt wird. Benutzen Sie am besten ggf. Ihr eigenes Formular, weil dasjenige der Datenbank *Projekte.mdb* von der Begleit-CD-ROM bereits das einzufügende Kombinationsfeld enthält. Wenn Sie dennoch mit dem Formular der Datenbank von der Begleit-CD-ROM arbeiten, sollten Sie das Kombinationsfeld mit der Bezeichnung *Datensatz suchen* in der Entwurfsansicht des Formulars löschen, bevor Sie das folgende Beispiel nacharbeiten.

- Blenden Sie ggf. die Toolbox ein, und klicken Sie darin auf die Symbol-Schaltfläche *Steuerelement-Assistenten* (vgl. links nebenstehend), falls diese nicht bereits gedrückt erscheint, um für die folgende Aufgabe den Kombinationsfeld-Assistenten zu aktivieren.

Bild 16.17: Das Kombinationsfeld wurde im rechten Teil des Formularkopfes eingefügt.

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Kombinationsfeld*, vgl. links nebenstehend, und ziehen Sie im rechten Teil des Formularkopfes ein Rechteck von ca. 6 cm Breite und ca. 0,5 cm Höhe (nachträgliche Änderungen sind ohne weiteres möglich), vgl. Bild 16.17. (Sie können das eingefügte Kombinationsfeld erst nach beendeter Arbeit des Kombinationsfeld-Assistenten sehen, hier gebe ich es wieder, um Ihnen die Positionierung zu zeigen.) Der Kombinationsfeld-Assistent zeigt sein erstes Dialogfeld, vgl. Bild 16.18.
- Wählen Sie im ersten Dialogfeld des Kombinationsfeld-Assistenten die Option *Einen Datensatz im Formular, basierend auf dem im Kombinationsfeld gewählten Wert suchen*, und bestätigen Sie mit Weiter >. Dann wird das zweite Dialogfeld des Kombinationsfeld-Assistenten angezeigt.
- Doppelklicken Sie in der Liste *Verfügbare Felder* auf die beiden Felder *Nachname* und *Vorname*, um diese in die Liste *Ausgewählte Felder* zu verschieben, vgl. Bild 16.19, wo bereits die Situation nach dem Verschieben der Felder dargestellt ist. Bestätigen Sie das Dialogfeld mit *Weiter* >.

Bild 16.18: Erstes Dialogfeld des Kombinationsfeld-Assistenten

Bild 16.19: Zweites Dialogfeld des Kombinationsfeld-Assistenten

Bild 16.20: Drittes Dialogfeld des Kombinationsfeld-Assistenten

- Im dritten Dialogfeld des Kombinationsfeld-Assistenten hätten Sie die Möglichkeit, die Breiten der Spalten zu verändern. Gleichfalls könnten Sie das Kontrollkästchen *Schlüsselspalte ausblenden* ausschalten, dann würde das Primärschlüsselfeld *PersonalCode*, das der Kombinationsfeld-Assistent automatisch im Hintergrund in das Kombinationsfeld aufgenommen hat, ebenfalls angezeigt werden. Sie sollten weder die Spaltenbreite noch diese Option verändern. Darüber hinaus könnten Sie auch – worüber das Dialogfeld nicht informiert! – die Reihenfolge der Spalten ändern: Markieren Sie zu diesem Zweck die zu verschiebende Spalte durch Klicken auf den Spaltenkopf, und ziehen Sie die Spalte dann an die gewünschte Position. Bestätigen Sie schließlich mit *Weiter* >. Dann wird das vierte Dialogfeld angezeigt.

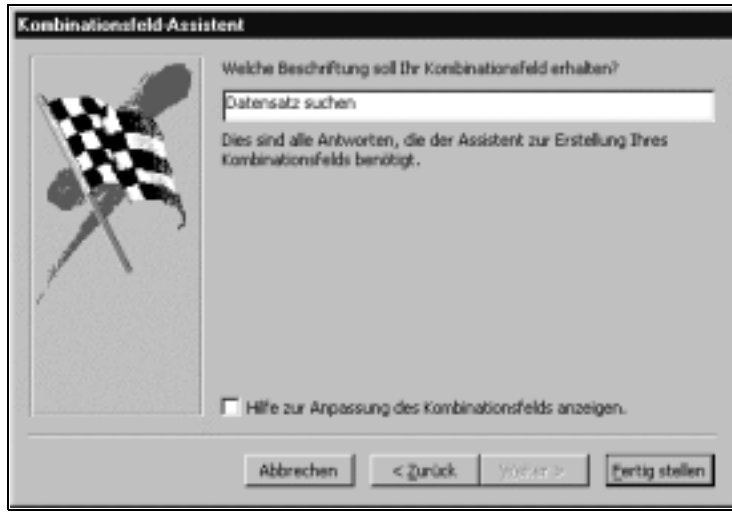


Bild 16.21: Viertes und letztes Dialogfeld des Kombinationsfeld-Assistenten

- Geben Sie im vierten und letzten Dialogfeld des Kombinationsfeld-Assistenten einen Namen ein, z.B. *Datensatz suchen*. Dieser wird als Text in das Bezeichnungsfeld eingetragen, das dem Kombinationsfeld beigefügt wird. Leider wird dieser Name nicht gleichzeitig als Name für das Kombinationsfeld verwendet, vgl. dazu weiter unten in diesem Punkt. Bestätigen Sie das Dialogfeld mit *Fertigstellen*. Dann beendet der Kombinationsfeld-Assistent seine Arbeit.
- Wechseln Sie in die Formularansicht, schlagen Sie die Dropdown-Liste des Kombinationsfeldes auf, und wählen Sie einen Eintrag darin, vgl. Bild 16.22. Dann wird im Formular der Datensatz, dessen Namen und Vornamen Sie ausgewählt haben, aktiviert.

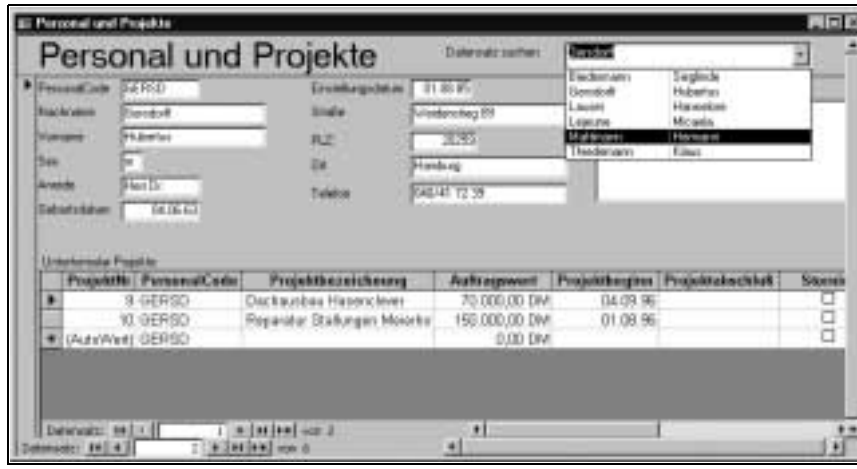


Bild 16.22: Eingefügtes und aufgeschlagenes Kombinationsfeld in der Formularansicht

Namen des Kombinationsfeldes und Code der Ereignisprozedur ändern

Dieser Punkt ist nur nachvollziehbar, wenn Sie minimale Kenntnisse über Ereignisprozeduren besitzen. Lesen Sie dazu ggf. an folgenden Stellen dieses Buches nach: Kap. 27, *Ereignisse*, und Kap. 29, *VBA (Visual Basic for Applications): Grundlagen*.

Sie können mit dem eingefügten Kombinationsfeld zur Datensatzsuche ohne weitere Änderungen bequem arbeiten. In manchen Fällen mag es Sie jedoch stören, daß das Kombinationsfeld vom Assistenten den Standardnamen *Kombinationsfeldn*, z.B. *Kombinationsfeld27* (wie im folgenden Beispiel) bekommen hat. Falls Sie sich auf das eingefügte Kombinationsfeld in Ausdrücken, Makros oder VBA-Prozeduren beziehen wollen, wäre dieser Name natürlich ziemlich ungeeignet, so daß Sie ihn ändern sollten. Das jedoch hat Konsequenzen für die Ereignisprozedur, die der Kombinationsfeld-Assistent zusammen mit dem Kombinationsfeld erstellt hat. Die Ereignisprozedur können Sie auf die folgende Weise betrachten und bearbeiten:

- Wechseln Sie mit dem Formular in die Entwurfsansicht, und doppelklikken Sie auf das eingefügte Kombinationsfeld, um es zu markieren und sein Eigenschaftenfenster anzuzeigen.
- Klicken Sie im Eigenschaftenfenster auf das Einstellungsfeld der Eigenschaft *Nach Aktualisierung*, und klicken Sie dann auf die Schaltfläche *Generator* (mit drei Punkten) am rechten Rand des Einstellungsfeldes. Dann wird der VBA-Editor geöffnet und darin das sogen. Klassenmodul *Form_Personal und Projekte,* in dem dann die Ereignisprozedur angezeigt wird, vgl. Bild 16.23.

Bild 16.23: Formularmodul mit der Ereignisprozedur für das vom Kombinationsfeld-Assistenten eingefügte Kombinationsfeld, das in diesem Falle den Namen »Kombinationsfeld27« bekommen hat.

Für den Fall, daß das Kombinationsfeld den Namen *Kombinationsfeld27* bekommen hat, wie dies im dargestellten Beispiel zutraf, hat die Ereignisprozedur den Prozedurnamen *Kombinationsfeld27_AfterUpdate* bekommen. Dieser Name korrespondiert mit dem Namen des Kombinationsfeldes und dem Ereignis *Nach Aktualisierung* (*AfterUpdate*). Außerdem bezieht sich die Ereignisprozedur in der dritten Codezeile mit dem Ausdruck

Me![Kombinationsfeld27]

auf den Namen des Kombinationsfeldes.

Wenn Sie den Namen des Kombinationsfeldes ändern, müssen Sie daher auch den Namen der Ereignisprozedur und den angegebenen Ausdruck anpassen. Wenn Sie dem Kombinationsfeld beispielsweise den Namen *DatensatzSuchen* geben, müßte die Ereignisprozedur so geändert werden, daß sie wie folgt lautet (Änderungen fett hervorgehoben):

```
Private Sub DatensatzSuchen_AfterUpdate()
    Dim rs As Object
    Set rs = Me.Recordset.Clone
    rs.FindFirst "[PersonalCode] = '" _
    & Me![DatensatzSuchen] & "'"
    Me.Bookmark = rs.Bookmark
End Sub
```

16.7 Unterformular

Der Umgang mit Haupt- und Unterformularen wird ausführlich in Kap. 18, *Haupt- und Unterformulare*, behandelt. Dort wird auch ein Verfahren zum Einfügen eines Unterformulars in ein Hauptformular beschrieben, das in vielen Fällen dem in diesem Abschnitt geschilderten Verfahren vorzuziehen ist. An dieser Stelle geht es, nicht zuletzt aus systematischen Gründen, um ein Unterformular als einzelnes Steuerelement. Sie benötigen die Kenntnis

der hier beschriebenen Zusammenhänge auch, wenn Sie nachträglich strukturelle Änderungen an einem in ein Hauptformular eingefügten Unterformular vornehmen wollen.

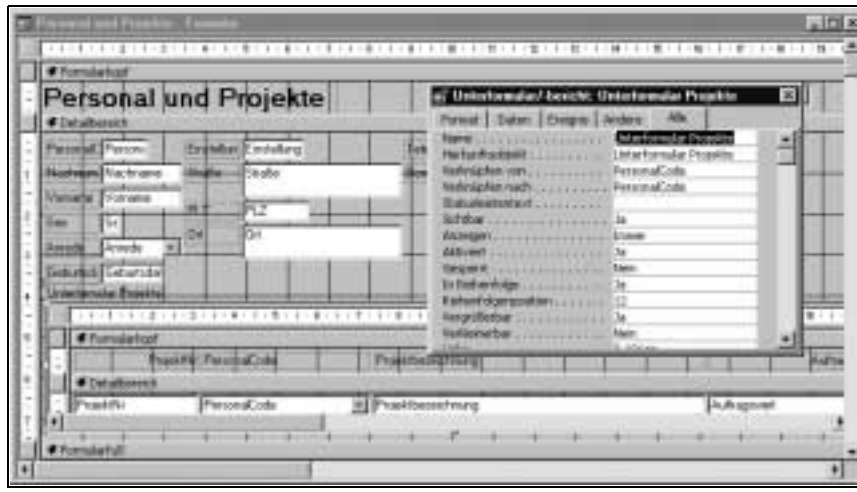


Bild 16.24: Im unteren Teil ist ein Ausschnitt aus dem Steuerelement »Unterformular« mit dem Namen »Unterformular Projekte« zu sehen. Das Eigenschaftenfenster gibt die für ein Unterformular spezifischen Eigenschaften wieder.

Funktionsweise eines Unterformulars

Die Art und Weise, wie ein Hauptformular ein anderes Formular aufnehmen und verwalten kann, läßt sich folgendermaßen beschreiben: Dem Formular, welches als Hauptformular fungieren soll, muß das Steuerelement *Unterformular* hinzugefügt sein. Die wichtigste Eigenschaft dieses Unterformulars lautet *Herkunftsobjekt* (vgl. das Eigenschaftenfenster in Bild 16.24). Die Eigenschaft *Herkunftsobjekt* muß auf den Namen eines bereits gespeicherten, im Prinzip völlig selbständigen Formulars eingestellt werden. Dann wird dieses letztere Formular zu einem Unterformular. Es wird dann in der Entwurfs- und Formularansicht des Hauptformulars als ein Teilformular angezeigt werden.

Ein Unterformular kann mit dem Hauptformular verknüpft sein. Eine Verknüpfung ist aber nicht notwendig. Für eine sinnvolle Verknüpfung müssen, – ganz analog zum Verknüpfen zweier Tabellen in einer Abfrage – in den Tabellen bzw. Abfragen, die dem Haupt- und Unterformular zugrunde liegen, zwei Felder vorhanden sein, die sich inhaltlich und im Datentyp entsprechen. Das verknüpfende Feld des Unterformulars wird mit der Eigenschaft *Verknüpfen von* angegeben, dasjenige aus dem Hauptformular mit der Eigenschaft *Verknüpfen nach*. Bei einer 1:n-Beziehung beispielsweise wären dies das Primärschlüsselfeld auf der 1-Seite (angegeben mit der Eigen-

schaft *Verknüpfen nach*) und der Fremdschlüssel auf der n-Seite (angegeben mit der Eigenschaft *Verknüpfen von*). Es lassen sich jedoch auch Verknüpfungen zwischen Haupt- und Unterformular herstellen, ohne daß den beiden Formularen Tabellen mit einer definierten Beziehung zugrunde liegen. Beispiele finden Sie in Kap. 18, *Haupt- und Unterformular.* Wenn Sie Haupt- und Unterformular nicht miteinander verknüpfen, verhalten sich die beiden Formulare wie zwei selbständige Formulare.

Ein Hauptformular kann mehrere Unterformulare enthalten. Dabei ist es auch möglich, daß ein Unterformular selbst ein anderes Unterformular enthält. Diese Art von Verschachtelung ist allerdings auf drei Stufen beschränkt: Hauptformular, Unterformular und Unterformular im Unterformular. Die drei verschachtelten Formulare können jeweils miteinander verknüpft sein; dann verwaltet das Hauptformular indirekt auch das unterste Unterformular.

Ein Unterformular mit der Toolbox erstellen

Um mittels der Toolbox einem Formular ein anderes Formular als Unterformular einzufügen, gehen Sie folgendermaßen vor:

- Erstellen Sie ggf. zunächst das Formular, welches als Unterformular eingefügt werden soll. (Dieser Schritt kann im Prinzip auch später nachgeholt werden.)
- Öffnen Sie das Formular, welches als Hauptformular dienen soll, in der Entwurfsansicht.
- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Unterformular/-bericht*, und ziehen Sie im Formular ein Rechteck. Es sollte im Ergebnis ungefähr so groß sein, wie das Unterformular Platz braucht; andernfalls müßte das Unterformular in der Formularansicht des Hauptformulars gerollt werden.
- Klicken Sie im Eigenschaftenfenster (das Steuerelement *Unterformular* muß markiert sein) in das Bearbeitungsfeld der Eigenschaft *Herkunftsobjekt,* und wählen Sie aus dem Dropdown-Listenfeld den Namen des einzufügenden Formulars aus; spätestens an dieser Stelle muß das Formular, das als Unterformular dienen soll, gespeichert mit einem Namen existieren.
- Sobald Sie das Formular in der Dropdown-Liste der Eigenschaft *Herkunftsobjekt* ausgewählt haben (vgl. den vorangehenden Aufzählungspunkt), sucht Access im Haupt- und Unterformular nach Feldern gleichen Namens und Datentyps. Wenn diese existieren, werden sie in die Eigenschaften *Verknüpfen von* und *Verknüpfen nach* eingetragen, so daß Haupt- und Unterformular miteinander verknüpft sind. Falls Access keine verknüpfenden Felder findet, geben Sie diese zu den Eigenschaften *Verknüpfen von* (für das Verknüpfungsfeld im Unterformular) und *Verknüp-*

fen nach (für das Verknüpfungsfeld im Hauptformular) an, falls Sie dieses wünschen. Wenn Sie manuell verknüpfen, können Sie auch Felder ungleichen Namens als Verknüpfungsfelder wählen. Achten Sie jedoch auf eine inhaltliche Entsprechung: Im allgemeinen werden Sie nur mit Feldern verknüpfen, die in der Datenbasis des Hauptformulars als Primärschlüssel und in der Datenbasis des Unterformulars als Fremdschlüssel definiert sind.

Wenn die Verknüpfung über mehr als ein Feld erfolgen soll – beispielsweise könnten Primärschlüssel und Fremdschlüssel einer 1:n-Beziehung aus mehr als einem Feld zusammengesetzt sein – , werden alle Verknüpfungsfelder angegeben, jeweils getrennt durch ein Semikolon. Wenn Sie beispielsweise für die Eigenschaften *Verknüpfen von* und *Verknüpfen nach* die beiden Felder *MaßnahmeNr* und *Amt* angeben wollen schreiben Sie

MaßnahmeNr; Amt

Die verknüpfenden Felder müssen nicht in dem Hauptobjekt oder dem untergeordneten Objekt enthalten sein. Solange diese Felder in den den Objekten zugrundeliegenden Tabellen oder Abfragen enthalten sind, können Sie diese Felder zum Verknüpfen der Objekte verwenden.

Änderungen am Unterformular vornehmen

Seit Access 2000 erscheint das Unterformular, das dem Steuerelement *Unterformular/-bericht* als *Herkunftsobjekt* zugeordnet ist, in diesem Steuerelement in der Entwurfsansicht, wenn sich das Hauptformular in derselben Ansicht befindet. Es kann daher direkt aus der Entwurfsansicht des Hauptformulars heraus bearbeitet werden. Sie können das Unterformular jedoch ebensogut selbständig in der Entwurfsansicht öffnen und dann bearbeiten.

16.8 Register

Sie kennen Registerkarten aus vielen Access-Dialogfeldern, beispielsweise dem Dialogfeld *Optionen* zum gleichnamigen Befehl aus dem Menü *Extras*. Register mit ihren aufschlagbaren Karten ermöglichen es, viele Steuerelemente in einem Formular unterzubringen, ohne daß Sie in der Senkrechten oder Waagerechten mit einer Bildlaufleiste blättern müßten. Die Alternative zur Verwendung von Registerkarten sind Seitenumbrüche im Formular, die Sie ebenfalls definieren können, vgl. weiter unten in diesem Kapitel, Punkt 16.14, Seitenwechsel.

In Bild 16.25 ist das Formular *Personal* aus der Datenbank *Nordwind.mdb* wiedergegeben, das ein Steuerelement Register mit zwei Registerkarten enthält. Um ein ähnliches Register in einem Formular zu erstellen, gehen Sie folgendermaßen vor:

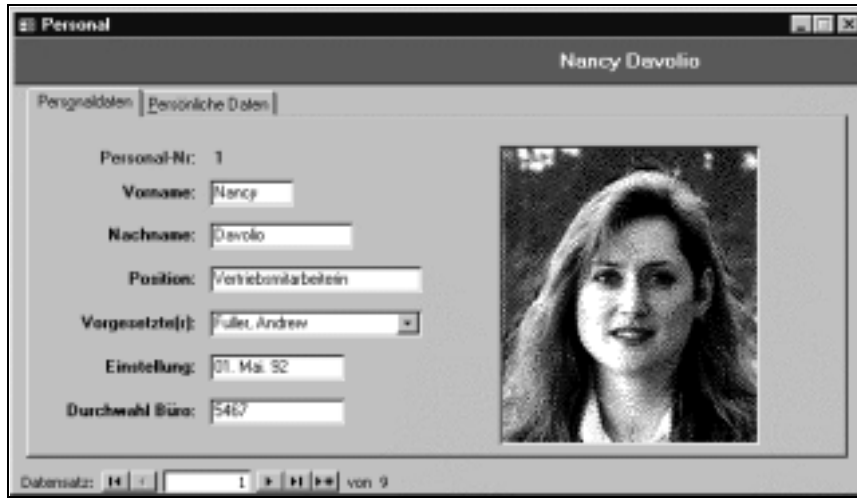


Bild 16.25: Dieses Formular »Personal« in der Datenbank »Nordwind.mdb« enthält das Steuerelement »Register« mit den beiden Registerkarten »Personaldaten« und »Persönliche Daten«.

- Öffnen Sie die Datenbank *Nordwind.mdb*.
- Erstellen Sie ein neues Formular, und weisen Sie diesem als *Datenherkunft* die Tabelle *Personal* zu.
- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Register-Steuerelement*, vgl. links nebenstehend.

Bild 16.26: Steuerelement »Register« unmittelbar nach dem Einfügen

- Ziehen Sie mit der Maus ein Rechteck der gewünschten Größe im Detailbereich des Formulars. Dann sollte der Formularentwurf etwa so aussehen, wie in Bild 16.26 dargestellt.

Eigenschaften des Registers einstellen

Ein Register hat, wie jedes andere Steuerelement, Eigenschaften, die Sie mit dem Eigenschaftenfenster einstellen können. Um das Steuerelement *Register* zu markieren, klicken Sie auf einen Rand des Registers. Im geöffneten Eigenschaftenfenster können Sie dann seine Eigenschaften bearbeiten. Wenn Sie auf den Rand des Registers doppelklicken, wird es markiert, und gleichzeitig wird ggf. das Eigenschaftenfenster angezeigt.

Eigenschaften der Registerkarten einstellen

Jede Registerkarte ist ein eigenständiges Element des Registers mit entsprechend eigenen Eigenschaften. Um eine Registerkarte zu markieren, klicken Sie auf deren Namen – Doppelklicken öffnet ggf. gleichzeitig ihr Eigenschaftenfenster.

Namen einer Registerkarte ändern

Um den Namen einer Registerkarte zu ändern, doppelklicken Sie auf die entsprechende Seite, um diese zu markieren und ggf. ihr Eigenschaftenfenster zu öffnen. Geben Sie dann im Bearbeitungsfeld der Eigenschaft *Beschriftung* den neuen Namen ein. Wenn für die Eigenschaft *Beschriftung* nichts angegeben ist, verwendet Access die Einstellung der Eigenschaft *Name*.

Registerkarte hinzufügen, löschen oder ihre Reihenfolge ändern

Klicken Sie mit der rechten Maustaste auf den Rahmen des Register-Steuerelements, um dessen Kontextmenü aufzuschlagen, vgl. Bild 16.27. Wählen Sie dann einen der geeigneten Befehle.

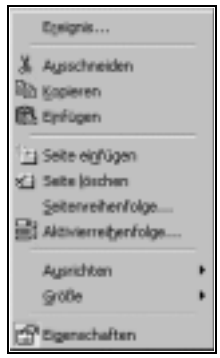


Bild 16.27: Kontextmenü des Steuerelements »Register«

Aktivierreihenfolge der Steuerelemente einer Seite ändern

Um die Reihenfolge der Steuerelemente einer Seite zu ändern, klicken Sie mit der rechten Maustaste auf die entsprechende Seite, um das Kontextmenü zu öffnen, vgl. Bild 16.27. Wählen Sie dann den Befehl *Aktivierreihenfolge...*, und nehmen Sie im Dialogfeld *Reihenfolge* die gewünschten Änderungen vor.

Steuerelemente in die Registerkarten einfügen

Das Einfügen eines Registers in das Formular bliebe im allgemeinen sinnlos, wenn seinen Registerkarten keine Steuerelemente hinzugefügt würden. Um dies zu bewerkstelligen, gehen Sie vor wie folgt:

- Klicken Sie auf die Registerkarte der Seite, der Sie Steuerelemente hinzufügen möchten.
- Fügen Sie der markierten Registerkarte mit einer der üblichen Methoden ein Steuerelement hinzu: Verwenden Sie die Toolbox, ziehen Sie aus der Feldliste, oder kopieren Sie ein Steuerelement von einer anderen Stelle auf die markierte Registerkarte. Zum Letzteren beachten Sie die Beschränkung gegenüber sonstigen Kopierverfahren, daß sich ein Steuerelement nicht auf ein Register ziehen läßt.

Beschriftung einer Registerkarte mit einem Bild versehen

Sie können die Beschriftung einer Registerkarte mit einem Bild versehen. Gehen Sie dazu folgendermaßen vor:

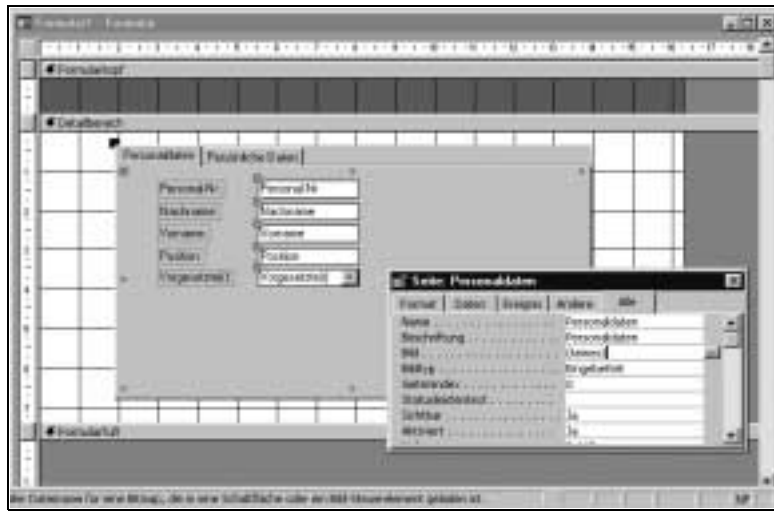


Bild 16.28: Den beiden Registerkarten wurden benutzerdefinierte Namen zugewiesen, und in die Registerkarte »Personaldaten« sind Steuerelemente eingefügt worden. Das Bearbeitungsfeld der Eigenschaft »Bild« im Eigenschaftenfenster hat den Fokus, so daß die Schaltfläche »Generator« an dessen rechtem Rand angezeigt wird.

- Doppelklicken Sie auf den Namen der Registerkarte, um diese zu markieren und gleichzeitig ggf. das Eigenschaftenfenster zu öffnen.
- Klicken Sie auf das Einstellungsfeld der Eigenschaft *Bild*, so daß dieses den Fokus bekommt. Dann wird am rechten Rand dieses Einstellungsfeldes die Schaltfläche *Generator* angezeigt, vgl. Bild 16.28.

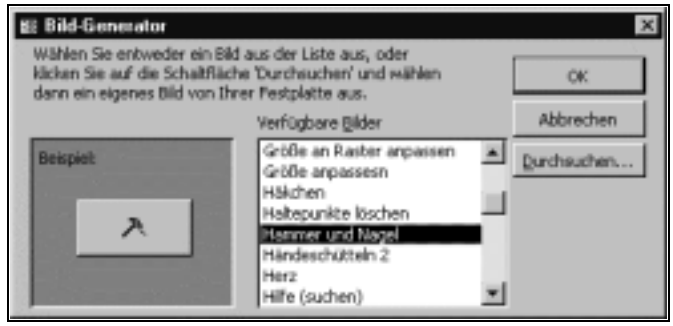


Bild 16.29: Dialogfeld »Bild-Generator«

- Klicken Sie auf die Schaltfläche *Generator*, um das Dialogfeld *Bild-Generator* aufzurufen, vgl. Bild 16.29. Wählen Sie in der Liste *Verfügbare Symbole* ein Symbol aus, oder durchsuchen Sie Ihre Festplatte nach einer geeigneten Bilddatei, indem Sie auf die Schaltfläche *Durchsuchen...* klikken. Bestätigen Sie schließlich mit OK. Das Ergebnis für die Registerkarten *Personaldaten* und *Persönliche Daten* können Sie in Bild 16.30 betrachten.

Bild 16.30: Register mit aufgeschlagener Registerkarte »Personaldaten« in der Formularansicht: Die Namen der Registerkarten wurden mit je einem Bild versehen.

Steuerelemente, die sich auf Registerkarten befinden, in Ausdrücken identifizieren

Für die Identifikation von Steuerelementen eines Formulars ist es gleichgültig, ob diese sich auf der Karte eines Registers befinden oder an einer anderen Stelle des Formulars. Beispielsweise wird das Steuerelement mit dem Namen *Nachname*, das sich auf der Registerkarte *Personaldaten* des Formulars *Formular1* befindet (vgl. Bild 16.30), mit einem Ausdruck auf die folgende Weise identifiziert:

=[Formulare]![Formular1]![Nachname]

16.9 Diagramm

Diagramme lassen sich auch einfügen, wenn Sie ein neues Formular erstellen: Wählen Sie im Dialogfeld *Neues Formular* den Diagramm-Assistenten, und folgen Sie dessen Aufforderungen. Bei diesem Weg müssen Sie allerdings gleichzeitig ein neues Formular erstellen. In diesem Punkt zeige ich dagegen, wie Sie ein Diagramm mit der Toolbox in ein bestehendes Formular einfügen, eine Aufgabe, die ebenfalls einen häufigen Anwendungsfall darstellt.

Arbeitsweise des Diagramm-Assistenten

Wenn Sie in ein Formular mit Hilfe des Diagramm-Assistenten ein Diagramm einfügen, leitet dieser Sie mit Hilfe mehrerer Dialogfelder bis hin zum fertigen Diagramm. Dieses ist dann in ein Steuerelement vom Typ *Diagramm* eingebunden.

Das Diagramm selbst wird mit Hilfe des Zusatzprogramms *Microsoft Graph* erstellt. Der Diagramm-Assistent ruft dieses Programm automatisch auf, während das Diagramm zum Einfügen erstellt wird, obgleich der Benutzer dies beim Vorgang der Diagramm-Erzeugung nicht merkt. Das fertige Diagramm ist ein OLE-Objekt (OLE = Object Linking and Embedding, d.h. Objekt Verknüpfen und Einbetten) und mit der Anwendung Microsoft Graph verbunden. OLE-Objekte zeichnen sich allgemein dadurch aus, daß sie an eine Windows-Anwendung gebunden sind, welche als sogen. *Server* dient. Für ein Access-Diagramm dient Microsoft Graph als Server. Praktisch bedeutet dies, daß Sie ein fertiges Diagramm jederzeit nachträglich von Access aus als Diagramm bearbeiten können (z.B. ein Säulen- in ein Liniendiagramm umwandeln oder die Achseneinteilung ändern). Zu diesem Zweck ruft Access dann Microsoft Graph auf, welches in den Vordergrund kommt. Nach der Diagrammbearbeitung schließen Sie Microsoft Graph, und das Diagramm nimmt in Access das Aussehen an, welches Sie ihm mit den Bearbeitungsmöglichkeiten von Microsoft Graph gegeben haben.

So betrachtet stellt ein in ein Formular eingefügtes Diagramm nur einen Spezialfall eines allgemeinen OLE-Objekts dar, welche generell in Formulare

auch mit Hilfe von Objektfeldern eingebettet werden können. Auf Objektfelder wird im einzelnen weiter unten in diesem Kapitel eingegangen, vgl. Punkt 16.10, *Gebundenes und ungebundenes Objektfeld*.

Welche Informationen benötigt ein Diagramm?

Ein Diagramm kann nur erstellt werden, wenn dem Diagramm-Assistenten bestimmte Minimalinformationen mitgeteilt werden. Dabei handelt es sich um die folgenden Punkte:

Minimale Angaben

- *Datenherkunft*. Ganz analog zur Datenherkunft eines Formulars können Sie den Namen einer Tabelle oder Abfrage angeben. Im Eigenschaftenfenster zum fertigen Diagramm können Sie nachträglich auch eine Wertliste oder eine Feldliste angeben.
- *Felder für Daten und Beschriftungen*. Da eine Datenreihe in einem Diagramm (das ist z.B. eine Reihe von Balken, Säulen oder Linienpunkten) nur auf Zahlenwerten aufgebaut werden kann, müssen Sie mindestens ein Feld angeben, welches numerische Werte enthält. Ferner sollten Sie ein Feld bestimmen, das die Rubriken für die einzelnen Datenpunkte beschriftet.

Wahlfreie Angaben

- *Verknüpfungsfelder.* So, wie Sie ein Unterformular in ein Hauptformular verknüpft oder unverknüpft einfügen können, ist dies auch für ein Diagramm möglich: Ein unverknüpftes Diagramm bleibt unverändert, wenn Sie in den Datensätzen des Formulars blättern. Ein verknüpftes Diagramm zeigt dagegen für jeden Datensatz des Formulars ein anderes Aussehen, weil es die Werte aus verknüpften Datensätzen, welche in Inhalt und Anzahl im allgemeinen variieren, darstellt.
- *Überschrift*. Sie können eine frei gewählte Überschrift angeben.

Über die voranstehenden Informationen hinaus kommen für ein neu erstelltes Diagramm zahlreiche Voreinstellungswerte zum Tragen. Sie können diese nachträglich aus Access heraus mit Hilfe von Microsoft Graph bearbeiten. Auch eine Verknüpfung kann nachträglich durch entsprechende Angaben im Eigenschaftenfenster für das Diagramm-Objektfeld hergestellt werden.

Ein Diagramm einfügen

Im folgenden wird gezeigt, wie Sie ein Diagramm praktisch erstellen und bearbeiten. Dabei wird zunächst ein unverknüpftes Diagramm erzeugt, welches anschließend durch entsprechende Bearbeitung mit dem Formular, in das es eingebettet ist, verknüpft wird. Darüber hinaus lernen Sie exemplarisch kennen, wie Sie Bearbeitungen eines Diagramms mit Microsoft Graph vornehmen können.

Aufgabe

Für die Datenbank *Projekte.mdb* von der Begleit-CD-ROM soll ein neues Formular erstellt werden, dem ein Diagramm eingefügt wird. Das Formular soll als Datenherkunft die Tabelle *Personal* haben und deren Felder *Vorname* und *Nachname* darstellen. Die Datenbasis des Diagramms soll die Tabelle *Projekte* sein. Das Diagramm soll die Auftragswerte aller Projekte als Säulendiagramm darstellen und die Säulen mit den Projektbezeichnungen beschriften.

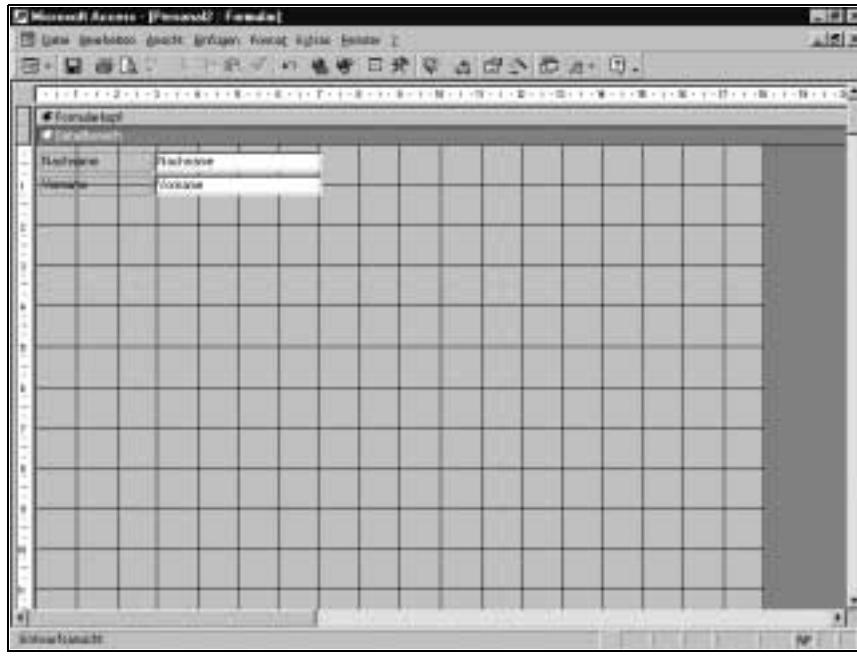


Bild 16.31: Entwurfsansicht des Formulars »Personal und Projekte im Diagramm« vor dem Einfügen des Diagramms.

Um diese Aufgabe zu lösen, gehen Sie wie folgt vor:

- Öffnen Sie ggf. die Datenbank *Projekte.mdb* von der Begleit-CD-ROM.
- Erstellen Sie ein neues Formular, am besten mit dem Formular-Assistenten.
- Geben Sie dafür als Datenherkunft die Tabelle *Personal* an.
- Nehmen Sie die Felder *Vorname* und *Nachname* in das Formular auf.
- Lassen Sie das Formular in der Entwurfsansicht anzeigen, und positionieren Sie die beiden Textfelder für *Vorname* und *Nachname* ungefähr so, wie dies in Bild 16.31 zu sehen ist. Vergrößern Sie den Detailbereich so, daß er etwa bis 18 cm in der Waagerechten und 12 cm in der Senkrechten reicht.

Bild 16.32: Erstes Dialogfeld des Diagramm-Assistenten

- Wählen Sie aus dem Menü *Einfügen* den Befehl *Diagramm*, und ziehen Sie dann mit dem Diagramm-Mauszeiger im Detailbereich des Formulars ein Rechteck von ca. 9 cm Höhe und 16 cm Breite. Sobald Sie das Rechteck gezogen und die Maustaste losgelassen haben, beginnt der Diagramm-Assistent seine Arbeit und präsentiert Ihnen sein erstes Dialogfeld, vgl. Bild 16.32.
- Wählen Sie als Datenherkunft die Tabelle *Projekte*. Klicken Sie auf die Schaltfläche *Weiter* >.

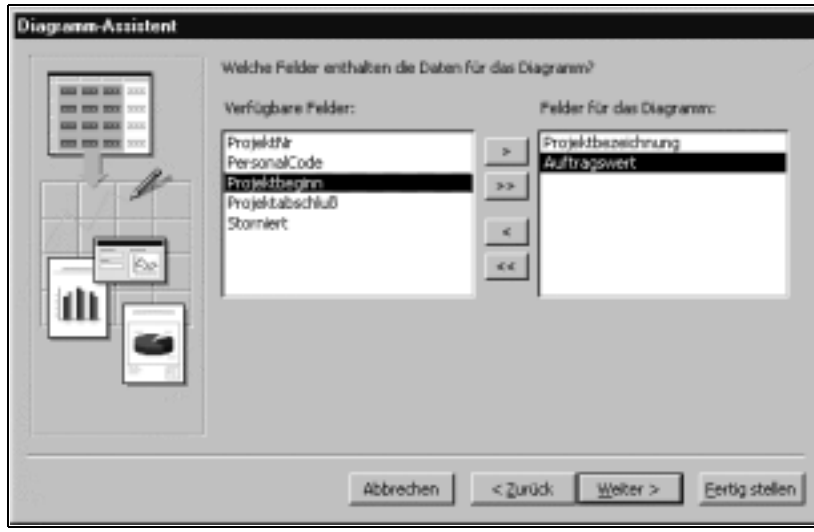


Bild 16.33: Zweites Dialogfeld des Diagramm-Assistenten

- Doppelklicken Sie im zweiten Dialogfeld des Diagramm-Assistenten (vgl. Bild 16.33) in der Liste *Verfügbare Felder* nacheinander auf die Felder *Projektbezeichnung* und *Auftragswert*, um diese in die Liste *Felder für das Diagramm* zu verschieben. Klicken Sie auf die Schaltfläche *Weiter* >.

Bild 16.34: Drittes Dialogfeld des Diagramm-Assistenten

- Wählen Sie im dritten Dialogfeld (vgl. Bild 16.34) das erste Säulendiagramm, und bestätigen Sie mit der Schaltfläche *Weiter* >.

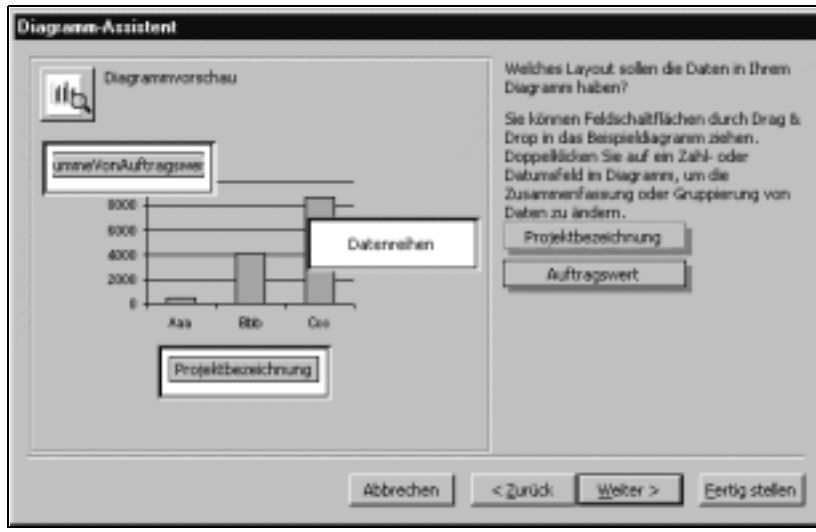


Bild 16.35: Viertes Dialogfeld des Diagramm-Assistenten

- Das vierte Dialogfeld des Diagramm-Assistenten ermöglicht es, auf die Anordnung und auf die Art der Berechnung der Datenreihen Einfluß zu nehmen. Der Diagramm-Assistent hat die beiden im zweiten Dialogfeld für das Diagramm ausgewählten Felder *Projektbezeichnung* und *Auftragswert* den Diagrammachsen richtig zugeordnet: Der Auftragswert wird als Datenreihe dargestellt, die Projektbezeichnungen sind für die waagerechte Rubrikenachse vorgesehen. Der Diagramm-Assistent konnte die Zuordnung offenbar richtig vornehmen, weil das Feld *Projektbezeichnung* Textwerte enthält, das Feld *Auftragswert* dagegen Zahlenwerte. Wenn beide Felder Zahlenwerte enthalten hätten, könnte es an dieser Stelle Probleme geben, so daß Sie ggf. durch Ziehen der Felder im Dialogfeld eine andersartige Zuordnung zu den Achsen vornehmen müßten.

  Allerdings soll die Art der vom Diagramm-Assistenten vorgesehenen Aggregierung der Auftragswerte nicht übernommen, sondern geändert werden, denn es soll keine Zusammenfassung der Auftragswerte erfolgen. Doppelklicken Sie zu dem Zweck auf das Feld mit der Beschriftung *Summe von Auftragswert* (nur etwas verstümmelt zu lesen). Dann wird das Dialogfeld *Zusammenfassen* angezeigt, vgl. Bild 16.36.

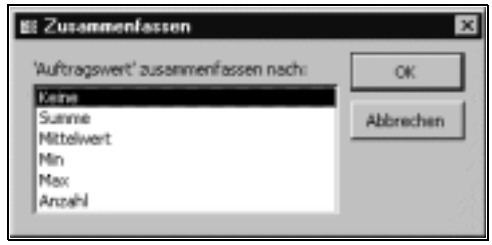


Bild 16.36: Dialogfeld »Zusammenfassen«, in dem Sie die Art der Aggregierung der Datenreihe ändern können.

- Wählen Sie im Dialogfeld *Zusammenfassen* die Kategorie *Keine*, und bestätigen Sie mit OK. Daraufhin sollte das Feld für den Auftragswert mit *Auftragswert* beschriftet sein. Bestätigen Sie das vierte Dialogfeld des Diagramm-Assistenten mit *Weiter* >.
- Das fünfte Dialogfeld des Diagramm-Assistenten (vgl. Bild 16.37) dient dazu, eine Verknüpfung der Datenbasis des Diagramms mit derjenigen des Formulars, in das es eingebettet ist, herzustellen oder abzulehnen. Der Diagramm-Assistent schlägt vor, daß das Diagramm mit dem Formular über das Feld *PersonalCode* verknüpft werden soll. Dies ist sinnvoll, weil für die beiden entsprechenden Tabellen *Personal* für das Formular und *Projekte* für das Diagramm eine 1:n-Beziehung definiert ist. Daher soll die vom Diagramm-Assistenten vorgeschlagene Verknüpfung übernommen werden. Falls Sie an dieser Stelle selbst eine Verknüpfung herstellen oder eine vorgeschlagene ablehnen wollten, könnten Sie die beiden Dropdown-

Listen *Formular Felder* und *Diagrammfelder* aufschlagen und die richtigen Felder oder den Eintrag *Kein Feld* für beide Listen wählen. Bestätigen Sie das fünfte Dialogfeld schließlich mit *Weiter* >. Dann wird das sechste und letzte Dialogfeld des Diagramm-Assistenten angezeigt.

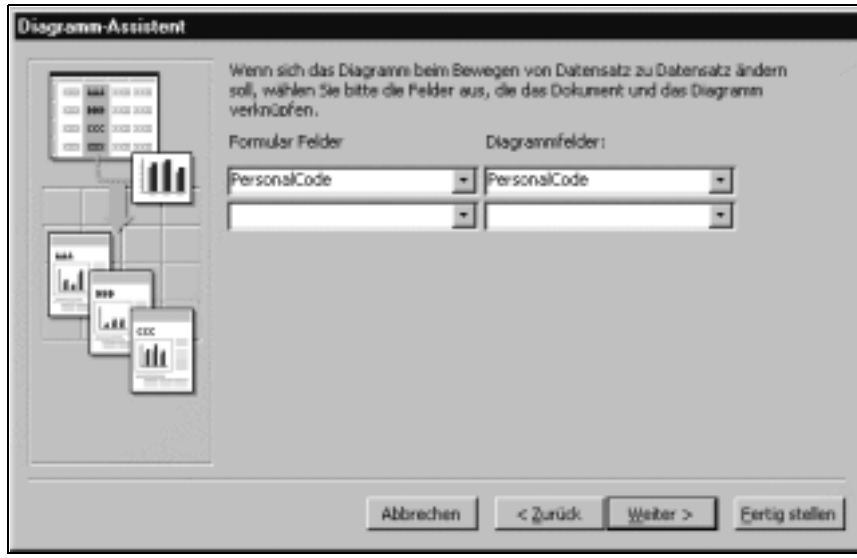


Bild 16.37: Fünftes Dialogfeld des Diagramm-Assistenten

Bild 16.38: Sechstes Dialogfeld des Diagramm-Assistenten

- Vergeben Sie im sechsten Dialogfeld des Diagramm-Assistenten (vgl. Bild 16.38) für das Diagramm den Namen *Auftragswerte der Projekte* (oder einen anderen), und wählen Sie die Option *Keine Legende anzeigen*, weil dieses bei nur einer Datenreihe überflüssig ist. Klicken Sie schließlich auf die Schaltfläche *Fertigstellen*. Dann erstellt der Diagramm-Assistent das Diagramm und zeigt es als Skizze – ohne echte Daten – in der Entwurfsansicht des Formulars, in der wir uns ja die ganze Zeit befanden.

Bild 16.39: Vom Diagramm-Assistenten erstelltes Diagramm in der Formularansicht.

- Wechseln Sie in die Formularansicht, um das Diagramm in der richtigen Darstellungsweise zu betrachten, vgl. Bild 16.39. Blättern Sie mit Hilfe der Navigationsschaltflächen durch die einzelnen Datensätze des Formulars. Dann erkennen Sie, daß für einige Mitarbeiter gar keine Säulen im Diagramm angezeigt werden. Dies liegt daran, daß diese keine Projekte bearbeitet haben.
- Das Diagramm erscheint nicht gerade optimal, weil beispielsweise die Beschriftungen der Rubrikenachse unzureichend sind und der rechte Diagrammteil stets leer ist. Daher soll es bearbeitet werden. Wechseln Sie zu diesem Zweck wiederum in die Entwurfsansicht des Formulars, und doppelklicken Sie auf das Diagramm. Dann wird über die OLE-Verbindung Microsoft Graph aufgerufen, vgl. Bild 16.40
- Nehmen Sie am Diagramm die von Ihnen für erforderlich gehaltenen Bearbeitungen vor. Gehen Sie dabei folgendermaßen vor: Doppelklicken Sie auf das zu bearbeitende Element. Damit wird dieses markiert und zugleich das zugehörige Dialogfeld zum Bearbeiten dieses Diagrammele-

ments angezeigt. Sie sollten insbesondere die Zeichnungsfläche des Diagramms durch Ziehen vergrößern und den Schriftgrad für die Rubrikenbeschriftungen verkleinern. Schließen Sie Microsoft Graph, wenn Sie mit Ihren Bearbeitungen fertig sind. Dann kommen Sie wieder zur Entwurfsansicht des Formulars. Vergrößern Sie ggf. auch das Diagramm-Steuerelement in der Entwurfsansicht des Formulars, damit die Beschriftung der Rubrikenachse besser ausgegeben wird. In Bild 16.41 ist das Diagramm in der Formularansicht wiedergegeben, wie es sich nach Bearbeitung mit Microsoft Graph in der Formularansicht des Formulars zeigt.

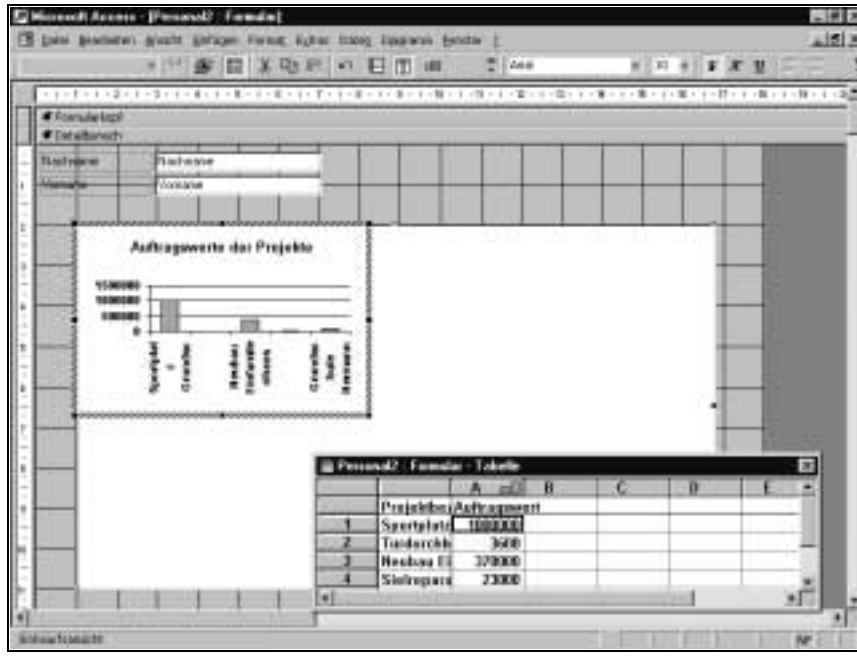


Bild 16.40: Anwendungsfenster von Microsoft Graph zum Bearbeiten des Diagramms

- Falls Sie, wie in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, empfohlen, selbst eine Datenbank *Projekte.mdb* erstellt, darin eigene Projekte eingegeben haben und im vorliegenden Beispiel mit dieser Datenbank arbeiten, dürfte Ihr Diagramm inhaltlich anders aussehen als das in Bild 16.41 wiedergegebene, wenngleich es ihm strukturell gleichen sollte.
- Speichern Sie das Formular unter dem Namen *Personal und Projekte im Diagramm*. Falls Sie mit der Datenbank *Projekte.mdb* von der Begleit-CD-ROM und nicht mit Ihrer eigenen entsprechenden Datenbank (vgl. Kap. 5) arbeiten, sollten Sie einen anderen Namen wählen, weil das in den vorangehenden Schritten erstellte Formular dort bereits namensgleich existiert.

Bild 16.41: Formular, wie es sich nach Bearbeitung mit Microsoft Graph in der Formularansicht zeigt

Das Diagramm ist vom Diagramm-Assistenten in das Steuerelement *Diagramm* eingefügt worden. Dieses Steuerelement hatten wir in das Formular eingefügt, indem wir nach Wahl des Befehls *Diagramm* aus dem Menü *Einfügen* in der Entwurfsansicht des Formulars ein Rechteck gezogen hatten. Dieses Steuerelement *Diagramm* hat – wie jedes Steuerelement – verschiedene Eigenschaften, die im Eigenschaftenfenster eingesehen und eingestellt werden können, vgl. das in Bild 16.42 wiedergegebene Eigenschaftenfenster.

Bild 16.42: Eigenschaftenfenster für das Steuerelement »Diagramm«

Der Diagramm-Assistent hatte eine Verknüpfung von Formular und Diagramm über das Feld *PersonalCode vorgeschlagen*, und wir hatten diesen Vorschlag bestätigt, vgl. oben die Ausführungen zum fünften Dialogfeld des Diagramm-Assistenten, Bild 16.37. Diese Verknüpfung hat sich in den Einstellungen zu den Eigenschaften *Verknüpfen von* und *Verknüpfen nach* des Steuerelements *Diagramm* im Formular niedergeschlagen, vgl. das Eigenschaftenfenster in Bild 16.42. Dort sehen Sie auch, daß es sich bei dem Diagramm um eine OLE-Verknüpfung zu Microsoft Graph 2000 handelt. Für die *Datensatzherkunft* ist die SQL-SELECT-Anweisung

```
SELECT [Projektbezeichnung],[Auftragswert] FROM [Projekte];
```

eingetragen worden. Hier zeigt sich der Vorteil einer derartigen Angabe gegenüber der Alternative, ein Abfrage-Objekt zu erzeugen, unter einem Namen zu speichern und diesen als Datensatzherkunft anzugeben. Die kurze SQL-SELECT-Anweisung ist sehr viel einfacher zu handhaben.

16.10 Gebundenes und ungebundenes Objektfeld

Funktionsweise eines Objektfeldes

Ein Objektfeld dient dazu, ein grafisches Objekt in einem Formular unterbringen zu können. Dabei kann das grafische Objekt ein OLE-Objekt sein; dann ist es mit einer anderen Windows-Anwendung verknüpft und kann bearbeitet und automatisch aktualisiert werden. Wenn Sie auf ein OLE-Objekt doppelklicken, wird die mit ihm verbundene Anwendung gestartet, so daß Sie das Objekt bearbeiten können. Wenn das grafische Objekt nicht mit einer anderen Anwendung verknüpft ist, liegt kein OLE-Objekt vor; ein solches Objekt kann auch nicht von Access aus bearbeitet werden.

Gebundenes Objektfeld. Access unterscheidet zwischen gebundenen und ungebundenen Objektfeldern. Ein gebundenes Objektfeld ist, ganz ähnlich wie z.B. ein Textfeld, an ein Tabellenfeld gebunden. Dabei enthält das Tabellenfeld, welches den Datentyp *OLE-Objekt* besitzen muß, die verschiedenen Objekte, und zwar je Datensatz eines. Wenn Sie einem Formular ein gebundenes Objektfeld eingefügt haben, wird beim Blättern im Formular für jeden Datensatz ein neues Objekt angezeigt.

Ungebundenes Objektfeld. Bei einem ungebundenen Objektfeld wird das Objekt mit dem Objektfeld selbst, und nicht in einer Tabelle, gespeichert. Daher kann ein ungebundenes Objektfeld auch nur *ein* Objekt wiedergeben. Es ist in dieser Hinsicht einem Bezeichnungsfeld vergleichbar, bei dem der Bezeichnungstext ebenfalls zusammen mit dem Steuerelement *Bezeichnungsfeld* gespeichert wird, weshalb er zwischen den Datensätzen nicht variiert.

Jeder der beiden Objektfeld-Typen kann sowohl ein OLE-Objekt wie auch ein unverknüpftes Objekt enthalten.

Ungebundenes Objektfeld erstellen

Um ein ungebundenes Objektfeld zu erstellen, gehen Sie wie folgt vor:

- Öffnen Sie ein Formular in der Entwurfsansicht.
- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Objektfeld* (vgl. links nebenstehend), und ziehen Sie an der Stelle, wo es eingefügt werden soll, ein Rechteck in der gewünschten Größe. Sobald Sie loslassen, fordert Access Sie mit dem Dialogfeld *Objekt einfügen* auf, einen Objekttyp auszuwählen, vgl. Bild 16.43.

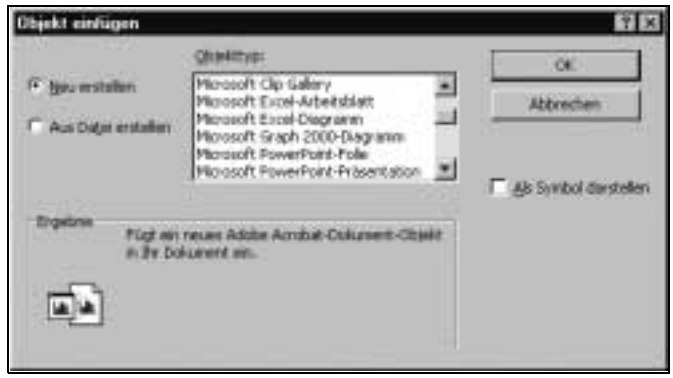


Bild 16.43: Dialogfeld zum Einfügen eines ungebundenen Objektfeldes

- Sie können eine vorhandene Datei als Objekt einfügen (Option *Aus Datei erstellen*, dann wählen Sie in einem weiteren Dialogfeld einen Dateinamen von der Festplatte) oder ein Objekt *Neu erstellen*. In diesem letzteren Fall wählen Sie aus der Liste *Objekttyp* die Anwendung aus, mit der Sie ein Objekt erzeugen möchten, beispielsweise *Microsoft Excel-Diagramm* oder *Paintbrush-Bild*, die aufgeschlagene Dropdown-Liste Objekttyp enthält – zumindest, wenn Sie das Office 2000-Paket installiert haben – eine sehr lange Liste mit Anwendungen, die ein Objekt erzeugen können.

Gebundenes Objektfeld erstellen

Ein gebundenes Objektfeld in ein Formular einzufügen setzt sinnvollerweise voraus, daß Sie über eine Tabelle verfügen, welche ein Feld vom Datentyp *OLE-Objekt* enthält und entsprechende Objekte gespeichert hat. Diese Zusammenhänge werden an einem Beispiel erklärt.

Beispiel

Aufgabe. Erstellen Sie für die Datenbank *Projekte.mdb* von der Begleit-CD-ROM ein Formular, welches für jeden Datensatz Namen und Vornamen des Mitarbeiters sowie dessen Portrait wiedergibt. Zu diesem Zweck enthält die

Datenbank *Projekte.mdb* das Formular *Mitarbeiterportraits* mit sechs Portraits fiktiver Mitarbeiter; diese habe ich aus der Datenbank *Nordwind.mdb* kopiert.

Im folgenden wird gezeigt, wie Sie in der Datenbank Projekte.mdb der Tabelle *Personal* ein neues Feld mit dem Datentyp *OLE-Objekt* hinzufügen und es mit Bildern versorgen. Ferner wird ein neues Formular erstellt. Die Datenbank *Projekte.mdb* von der Begleit-CD-ROM enthält diese Änderungen bereits. Sie sollten daher mit der in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, von Ihnen selbst erstellten Datenbank arbeiten. Falls dies nicht möglich ist, können Sie in der Tabelle *Personal* das Feld *Portrait* in der Entwurfsansicht löschen. Ferner sollten Sie das Formular *Personal mit Portrait* löschen, obwohl dieser Schritt nicht zwingend erforderlich ist, wenn Sie das Formular des folgenden Beispiels unter einem anderen Namen speichern.

Gehen Sie zur Lösung der Aufgabe wie folgt vor:

- Öffnen Sie die Datenbank *Projekte.mdb*.
- Öffnen Sie die Tabelle *Personal* in der Entwurfsansicht, und fügen Sie ihr das neue Feld *Portrait* hinzu, welchem Sie den Datentyp *OLE-Objekt* zuweisen. Speichern und schließen Sie die Tabelle *Personal*.
- Erstellen Sie ein neues Formular mit der Datenherkunft *Personal*, dem Sie im oberen Bereich je ein Textfeld für Vorname und Nachname einfügen. Es ist dabei gleichgültig, ob Sie sich dafür des Formular-Assistenten bedienen oder ein leeres Formular neu aufbauen.
- Fügen Sie dem neuen Formular ein gebundenes Objektfeld auf die folgende Weise ein:
- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Gebundenes Objektfeld* (vgl. links nebenstehend), und ziehen Sie ungefähr in der Mitte des Formulars ein Rechteck von 5 cm Breite und 5,8 cm Höhe; da Sie diese Größe nicht gleich treffen werden, korrigieren Sie am besten anschließend im Eigenschaftenfenster für das gebundene Objektfeld die Eigenschaften *Breite* und *Höhe* entsprechend.
- Stellen Sie die Eigenschaft *Steuerelementinhalt* für das *Gebundene Objektfeld* auf den Feldnamen *Portrait* ein, indem Sie im entsprechenden Eingabefeld das Dropdown-Listenfeld aufschlagen und das Feld auswählen.
- Speichern Sie das Formular unter dem Namen *Personal mit Portrait*.
- Wechseln Sie in die Formularansicht. Sie werden die Textfelder für den Vornamen und Namen mit Inhalt gefüllt sehen, während das neu eingefügte Objektfeld leer ist. Dies hat seinen Grund darin, daß das Feld *Portrait* in der Tabelle *Personal* zwar definiert, aber noch nicht mit Bildern versorgt ist. Um für jeden Mitarbeiter ein Portraitbild aus dem für diesen Zweck vorbereiteten Formular *Mitarbeiterportraits* zu kopieren und zu speichern, gehen Sie folgendermaßen vor:

- Öffnen Sie das Formular *Mitarbeiterportraits* in der Entwurfsansicht, ohne das neu erstellte Formular *Personal mit Portrait* zu schließen. Ordnen Sie am besten beide Formulare auf dem Bildschirm so an, daß beide gleichzeitig sichtbar sind. Achten Sie darauf, daß sich das Formular *Mitarbeiterportraits* in der Entwurfsansicht befindet, während für das andere die Formularansicht eingestellt ist.
- Kopieren Sie nacheinander die sechs Bilder aus dem Formular *Mitarbeiterportraits* in das Objektfeld des Formulars *Personal mit Portrait*, so daß schließlich jedem Datensatz ein Bild zugeordnet ist: Markieren Sie das erste Bild im Formular *Mitarbeiterportraits*, kopieren Sie es mit der Tastenkombination Strg+C in die Zwischenablage, klicken Sie in das Objektfeld im Formular *Personal mit Portrait*, und fügen Sie das Bild aus der Zwischenablage mit der Tastenkombination Strg+V ein. Wiederholen Sie den Kopier- und Einfügevorgang für die anderen fünf Datensätze, wobei Sie jeweils vorher zum nächsten Datensatz blättern. Das Ergebnis für den ersten Datensatz sollte ungefähr so aussehen, wie in Bild 16.44 wiedergegeben.

Bild 16.44: Erster Datensatz im Formular »Personal mit Portrait«

Verknüpfung eines OLE-Objekts aufheben, um das Objekt statisch zu machen

Manchmal möchten Sie vielleicht verhindern, daß ein OLE-Objekt bearbeitet werden kann. Dann müssen Sie die Verbindung zur entsprechenden Anwendung lösen, wonach das Objekt dann kein OLE-Objekt mehr sein wird. Gehen Sie dazu folgendermaßen vor:

- Markieren Sie das Objekt. Bei einem ungebundenen Objektfeld ist dies nur in der Entwurfsansicht des Formulars möglich, bei einem gebundenen auch in der Formularansicht.
- Wählen Sie aus dem Menü *Bearbeiten* den Befehl, der mit *-Objekt* beginnt, angeführt vom Namen der zugehörigen Anwendung. Für ein Paintbrush-Objekt beispielsweise lautet der Befehl*: Paintbrush-Objekt*. Wählen Sie den Befehl und dann den Unterbefehl *Konvertieren*. Dann öffnet sich das Dialogfeld *Konvertieren*, mit dem Sie beispielsweise zu einer Bitmap-Grafik konvertieren können.

Nach dieser Operation ist das Bild statisch in dem Sinne, daß es von Access aus nicht mehr durch Doppelklicken darauf bearbeitet werden kann.

Unverknüpftes Objekt durch Kopieren einfügen

Sie können in ein Formular Bildobjekte durch Kopieren über die Zwischenablage einfügen. Wenn Sie es in ein Formular in der Entwurfsansicht einfügen, erzeugt Access automatisch ein Steuerelement *Objektfeld*, dem das kopierte Bild hinzugefügt ist. Um beispielsweise aus der Anwendung Excel heraus ein unverknüpftes Diagramm in ein Access-Formular einzufügen, gehen Sie wie folgt vor:

- Öffnen Sie Excel und darin die Arbeitsmappe mit dem Diagramm. Markieren Sie das Diagramm und kopieren Sie es in die Zwischenablage, z.B. mit der Tastenkombination Strg+C.
- Wechseln Sie zu Access, und öffnen Sie das Formular in der Entwurfsansicht. Fügen Sie das Excel-Diagramm aus der Zwischenablage ein, z.B. mit der Tastenkombination Strg+V. Verschieben Sie das Objektfeld an die gewünschte Position und bearbeiten Sie es ggf. weiter.

16.11 Bild

Das Steuerelement *Bild* kann nur statische Bilder aufnehmen, d.h. diese können keine OLE-Objekte sein. Der Vorteil dieses Steuerelements liegt darin, daß seine Bilder wesentlich schneller angezeigt werden können als OLE-Bilder. Ein Steuerelement vom Typ *Bild* besitzt keine Eigenschaft *Steuerelementinhalt* und ist daher stets ein ungebundenes Steuerelement. Sie können in ein Steuerelement dieses Typs ein Bild nur als Bilddatei einfügen, z.B. Dateien mit der Namenserweiterung WMF oder BMP.

Um ein Steuerelement *Bild* in ein Formular einzufügen, verfahren Sie wie folgt:

- Öffnen Sie das Formular in der Entwurfsansicht.
- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Bild*, vgl. links nebenstehend, und klicken Sie dann auf die Stelle im Formular, an der das Bild eingefügt werden soll. Dann wird das Dialogfeld *Grafik einfügen* angezeigt.

- Wählen Sie im Dialogfeld *Grafik einfügen* die Bilddatei, die Sie in das Steuerelement einfügen wollen, und bestätigen Sie mit OK. Dann wird das Bild eingefügt.

16.12 Linie

Sie können einem Formular Linien hinzufügen, um beispielsweise einen Bereich von einem anderen optisch zu trennen oder, um ein anderes Beispiel zu nennen, ein Steuerelement durch Unterstreichen hervorzuheben. Gehen Sie zum Einfügen einer Linie folgendermaßen vor:

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Linie,* und ziehen Sie im Formular an geeigneter Stelle eine Linie. Um eine genau senkrechte oder genau waagerechte Linie zu ziehen, drücken Sie die ⇧-Taste, und halten sie während des Ziehens gedrückt. Je nachdem, in welche Richtung Sie zuerst ziehen, wird die Linie waagerecht oder senkrecht.
- Bearbeiten Sie die Linie ggf. anschließend mit den Möglichkeiten der Format-Symbolleiste, mit deren Schaltflächen Sie Farbe und Linienstärke einstellen können.

16.13 Rechteck

Sie können in ein Formular Rechtecke einfügen, um beispielsweise eine Gruppe von zusammengehörigen Steuerelementen einzurahmen. Gehen Sie zum Einfügen eines Rechtecks folgendermaßen vor:

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Rechteck,* und ziehen Sie im Formular an geeigneter Stelle ein Rechteck.
- Bearbeiten Sie die Linie ggf. anschließend mit den Möglichkeiten der Format-Symbolleiste, mit deren Schaltflächen Sie Farbe und Linienstärke einstellen und vor allem den Hintergrund transparent (erforderlich für Umrahmungen) machen können.

Wenn Sie ein nicht-transparentes Rechteck als Hintergrund für ein anderes Steuerelement – z.B. eine Befehlsschaltfläche – verwenden wollen, kann es zunächst passieren, daß das Rechteck die Befehlsschaltfläche überdeckt. Um das Rechteck in den Hintergrund und die Befehlsschaltfläche in den Vordergrund zu bringen, markieren Sie das Rechteck, und wählen Sie dann den Befehl *In den Hintergrund* aus dem Menü *Format*.

16.14 Seitenwechsel

Bei längeren Formularen umbricht Access die Bildschirm- oder Druckseiten automatisch. Sie können statt dessen selbst Seitenwechsel (Seitenumbrüche) setzen. Gehen Sie für diesen Zweck folgendermaßen vor:

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Seitenwechsel*.
- Klicken Sie im Formular auf einen beliebigen Punkt in der (gedachten) waagerechten Linie, an der die Seite umbrochen werden soll. Das Steuerelement *Seitenwechsel* wird am äußerst linken Formularrand plaziert. Es wird durch eine kurze gestrichelte Linie folgendermaßen dargestellt:

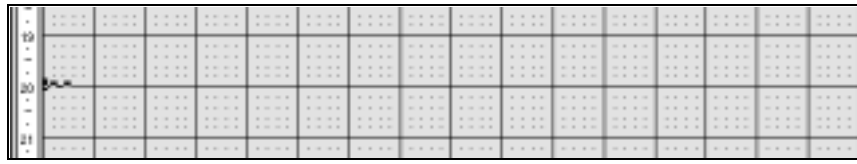


Bild 16.45: Hier erfolgt ein Seitenumbruch bei 20 cm.

Da ein Seitenwechsel ein Steuerelement ist, können Sie es markieren und seine Position auch nachträglich durch Ziehen mit der Maus oder durch Einstellen seiner Eigenschaft *Oben* im Eigenschaftenfenster verändern.

16.15 Befehlsschaltfläche

Zweck einer Befehlsschaltfläche ist es, eine Aktion zu starten, wenn Sie darauf klicken, die [Esc]-Taste oder die [↵]-Taste drücken oder ein entsprechendes Ereignis auslösen. Daher ist eine Befehlsschaltfläche im allgemeinen nur sinnvoll, wenn einer ihrer Ereigniseigenschaften (z.B. *Beim Klicken*, *Bei Eingabetaste* etc.) ein Ereignismakro oder eine Ereignisprozedur zugeordnet ist. Sie erstellen eine Befehlsschaltfläche auf die folgende Weise:

- Klicken Sie in der Toolbox auf die Symbol-Schaltfläche *Befehlsschaltfläche*, und ziehen Sie ein Rechteck an geeigneter Stelle.
- Access vergibt für eine Befehlsschaltfläche unmittelbar nach dem Einfügen die Beschriftung *Befehl n*, wobei n für eine Zahl steht. Ändern Sie die Beschriftung, und weisen Sie einer der ereignisorientierten Eigenschaften ein Makro oder eine Prozedur zu.

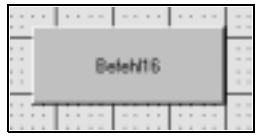


Bild 16.46: Access hat für diese Befehlsschaltfläche unmittelbar nach dem Einfügen diese Beschriftung vergeben, die Sie ändern sollten.

Wenn Sie selbst über kein Makro oder keine VBA-Prozedur verfügen, die Sie einer Befehlsschaltfläche als Ereignismakro oder -prozedur zuordnen könnten, kann sich der Befehlsschaltflächen-Assistent als nützlich erweisen, denn

dieser erzeugt selbst für bestimmte Standardsituationen die geeigneten Ereignisprozeduren. Verfahren Sie wie folgt, um den Befehlsschaltflächen-Assistenten aufzurufen:

- Klicken Sie ggf. in der Toolbox auf die Symbol-Schaltfläche *Steuerelement-Assistenten*, damit diese gedrückt erscheint.
- Klicken Sie dann auf die Symbol-Schaltfläche *Befehlsschaltfläche*, und ziehen Sie ein Rechteck im Formular. Dann zeigt der Befehlsschaltflächen-Assistent sein erstes Dialogfeld, vgl. Bild 16.47.

Bild 16.47: Erstes Dialogfeld des Befehlsschaltflächen-Assistenten

- Wenn Sie eine bestimmte Kategorie markieren, werden im Listenfeld *Aktionen* die für diese Kategorie verfügbaren Aktionen angezeigt. Wählen Sie die gewünschte Kombination von Kategorie und Aktion, und bestätigen Sie mit *Weiter* >.
- In zwei weiteren Dialogfeldern haben Sie dann noch die Möglichkeit zu entscheiden, ob die Befehlsschaltfläche mit Text oder einem Symbol, das Sie ggf. auch von der Festplatte als Bilddatei laden können, versehen werden soll. Schließlich können Sie der Befehlsschaltfläche auch einen neuen Namen geben.

Die wesentliche Leistung des Befehlsschaltflächen-Assistenten liegt darin, daß er im Klassenmodul des Formulars eine VBA-Prozedur erzeugt und diese der Ereigniseigenschaft *Beim Klicken* zuweist. Wie jede VBA-Prozedur können Sie auch diese anschließend ggf. bearbeiten.

Befehlsschaltfläche mit einem Bild versehen

Um die Befehlsschaltfläche mit einem Bild zu versehen, verfahren Sie folgendermaßen:

- Doppelklicken Sie auf die Befehlsschaltfläche, um sie zu markieren und gleichzeitig ggf. das Eigenschaftenfenster einzublenden.
- Klicken Sie im Eigenschaftenfenster in das Bearbeitungsfeld der Eigenschaft *Bild*.
- Klicken Sie auf die Symbol-Schaltfläche *Generator* am rechten Rand des Einstellungsfeldes (sie trägt drei Pünktchen).
- Wählen Sie im Dialogfeld *Bild-Generator* das passende Bild, oder durchsuchen Sie die Festplatte nach einer Bilddatei. Bestätigen Sie schließlich mit OK.

Bild von der Schaltfläche entfernen

Um ein Bild ggf. wieder von der Befehlsschaltfläche zu entfernen, löschen Sie den Eintrag *(Bitmap)* für die Eigenschaft *Bild*, den der Bild-Generator nach dem Einfügen eines Bildes dort vorgenommen hat. Wenn Sie die Warnmeldung mit *Ja* bestätigen, verschwindet das Bild von der Befehlsschaltfläche, und die Eigenschaft *Bild* erhält wieder die ursprüngliche Einstellung *(keines)*.

16.16 Reihenfolge der Steuerelemente ändern

Die Reihenfolge der Steuerelemente ist beispielsweise bedeutsam, wenn Sie sich in der Formularansicht zwischen den Steuerelementen mit der [⇆]-Taste bewegen. Access legt die Reihenfolge historisch fest, d.h. sie bestimmt sich nach der zeitlichen Reihenfolge, in der Sie die einzelnen Steuerelemente erstellt haben. Da diese im Vergleich zum Formularaufbau oft eher zufällig ist, müssen Sie die Reihenfolge häufig ändern. Das geschieht folgendermaßen:

- Öffnen Sie das betreffende Formular in der Entwurfsansicht.
- Wählen Sie den Befehl *Aktivierreihenfolge* aus dem Menü *Ansicht*. Access blendet das Dialogfeld *Reihenfolge* ein, vgl. Bild 16.48.
- Wählen Sie, ob Sie die Reihenfolge von Steuerelementen aus dem Formularkopf, dem Detailbereich oder dem Formularfuß ändern wollen.
- Markieren Sie das zu verschiebende Steuerelement, indem Sie auf seinen Markierer (das kleine graue Quadrat am linken Rand der Zeile mit dem Steuerelementnamen) klicken. Um mehrere Felder zu markieren, klicken Sie, und ziehen Sie anschließend.
- Klicken Sie erneut auf den Markierer des(r) markierten Steuerelements(e), ziehen Sie an die gewünschte Position, und lassen Sie die Maustaste los. Bestätigen Sie anschließend mit OK.

Bild 16.48: Dialogfeld »Reihenfolge«. In der dargestellten Situation wird das Feld »Vorname« vor das Feld »Nachname« verschoben.

Kapitel 17

ActiveX-Steuerelemente in Formularen

17.1 Überblick

Ein ActiveX-Steuerelement unterscheidet sich von einem Standard-Steuerelement zunächst dadurch, daß es kein integraler Bestandteil des Programms Microsoft Access ist, sondern zusätzlich installiert sein muß. Dieser Umstand bleibt Ihnen auf den ersten Blick verborgen, weil Sie unmittelbar nach der Installation von Access 2000 im allgemeinen über mehrere ActiveX-Steuerelemente verfügen können, obwohl Sie selbst keinerlei zusätzliche Installation vorgenommen haben. Dies liegt daran, daß mit dem Office 2000-Paket diverse ActiveX-Steuerelemente installiert werden, die prinzipiell allen Office 2000-Anwendungen – also z.B. auch Excel 2000 oder Word 2000 – zur Verfügung stehen.

Die Möglichkeit, daß dasselbe ActiveX-Steuerelement, obgleich nur einmal installiert, verschiedenen Anwendungen zur Verfügung steht, ist darauf zurückzuführen, daß ActiveX-Steuerelemente nach dem OLE- und COM-Konzept (COM = Component Object Model) erstellt sind, einem Windows-Standard, der Programme als Objekte modelliert, auf die von anderen Anwendungen aus zugegriffen werden kann. An dieser Stelle kann auf dieses Konzept nicht näher eingegangen werden, vgl. dazu genauer in diesem Buch Kap. 33, *Integration von Access 2000 in andere Office 2000-Anwendungen*, wo auch das OLE/COM-Konzept näher erklärt wird.

Wegen der OLE/COM-Basis von ActiveX-Steuerelementen ist es auch möglich, derartige Objekte von Drittanbietern zur Verfügung zu stellen und in Access zu integrieren. Allerdings verweist dies zugleich auf eine Schwäche für den normalen Benutzer: Fast alle in Access zur Verfügung stehenden ActiveX-Steuerelemente sind nicht speziell für den Gebrauch in Access erstellt, sondern ganz allgemein für OLE/COM-fähige Anwendungen. Daher

existieren im allgemeinen auch nicht die komfortablen Schnittstellen zu diesen Komponenten, wie man sie von den Standard-Steuerelementen gewohnt ist. Während Sie beispielsweise ein Access-Kombinationsfeld, dessen Leistung doch immerhin ziemlich komplex ist, mit nur wenig Aufwand funktionsfähig in ein Formular einfügen können, gilt dies für nur wenige der zur Verfügung stehenden ActiveX-Steuerelemente: Obwohl Sie fast jedes in Access zur Verfügung stehende ActiveX-Steuerelement leicht durch Ziehen mit der Maus in ein Formular oder einen Bericht einfügen können, bedarf es eines relativ großen Programmieraufwands mit VBA, dieses auch funktionsfähig zu machen. Die meisten ActiveX-Steuerelemente sind tot, solange Sie diese nicht mit Programmcode versorgt haben. Dürfte bereits dieser Umstand für viele Benutzer prohibitiv sein, so wirkt noch weiter abschreckend, daß die notwendigen Informationen über die Eigenschaften und Methoden, die ein bestimmtes ActiveX-Steuerelement besitzt, nicht von Access zur Verfügung gestellt werden. Dies wiederum ist Access nicht vorzuwerfen, liegt es doch am Konzept, daß derartige Steuerelemente eben von dritter Seite stammen und kein Bestandteil von Access sind. In der Access Online-Hilfe wird darüber hinaus betont, daß auch die Funktionsfähigkeit der ActiveX-Steuerelemente für Access nicht in allen Fällen gewährleistet werden kann. Die Hilfe sagt dazu:

> »Wenn Sie ActiveX-Steuerelemente mit anderen Anwendungen, wie z.B. Microsoft Excel, installiert haben, werden diese angezeigt, sobald Sie in der Toolbox auf *Weitere Steuerelemente* oder im Menü *Einfügen* auf den Befehl *ActiveX-Steuerelement* klicken. Einige ActiveX-Steuerelemente funktionieren jedoch nicht einwandfrei in Microsoft Access. Eine Liste der Steuerelemente, die für die Verwendung in Microsoft Access getestet und für gut befunden wurden, finden Sie in der INFO-Datei oder erhalten Sie beim Händler des Steuerelements.«

Aus diesen Gründen ist es nicht angezeigt, in diesem Kapitel ausführlich auf die einzelnen ActiveX-Steuerelemente einzugehen. Ich beschränke mich vielmehr darauf zu zeigen, wie Sie prinzipiell derartige Steuerelemente einfügen und ohne Programmieraufwand funktionsfähig einbinden können. Darüber hinaus wird kurz gezeigt, wie Sie ein neues ActiveX-Steuerelement installieren können.

17.2 ActiveX-Steuerelement einfügen

Unter den vielen Zusatzsteuerelementen findet sich eines, das sehr gut in Access integriert und schon fast ein fester Bestandteil davon geworden ist – das *Kalender-Steuerelement*. Im folgenden wird am Beispiel gezeigt, wie dieses Steuerelement in ein Formular eingefügt und mit sinnvollen Eigenschaften-Einstellungen versorgt werden kann.

Aufgabe und Lösung an einem Beispiel

Die Lösung der Aufgabe soll an einem Beispiel gezeigt werden: In das Formular *Personal* der Datenbank *Projekte.mdb* von der Begleit-CD-ROM sollen zwei Steuerelemente vom Typ *Kalender* eingefügt werden, mit deren Hilfe die beiden Felder *Geburtsdatum* und *Einstellungsdatum* mit Werten versorgt werden sollen. Gehen Sie folgendermaßen vor:

- Öffnen Sie die Datenbank *Projekte.mdb*.
- Öffnen Sie das Formular *Personal* in der Entwurfsansicht. In Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, Punkt 5.9, *Formular Personal erstellen*, wurde gezeigt, wie dieses Formular erstellt wird.

Bild 17.1: Formular »Personal« in der Entwurfsansicht. Der Detailbereich wurde vergrößert, um Platz zum Einfügen der Kalender-Steuerelemente zu schaffen.

- Vergrößern Sie den Detailbereich des Formulars ungefähr auf die Größe, wie dies in Bild 17.1 wiedergegeben ist, damit genügend Platz für das Einfügen zweier Kalender-Steuerelemente vorhanden ist.

- Klicken Sie in der Toolbox auf das Symbol *Weitere Steuerelemente*, vgl. links nebenstehend, oder wählen Sie im Menü *Einfügen* den Befehl *ActiveX-Steuerelement*. Im letzteren Fall erscheint das Dialogfeld *ActiveX-Steuerelement einfügen*, aus dem Sie das Steuerelement auswählen können, im ersteren Fall wird eine Liste der ActiveX-Steuerelemente mit ihren Symbolen angeboten, vgl. Bild 17.2.

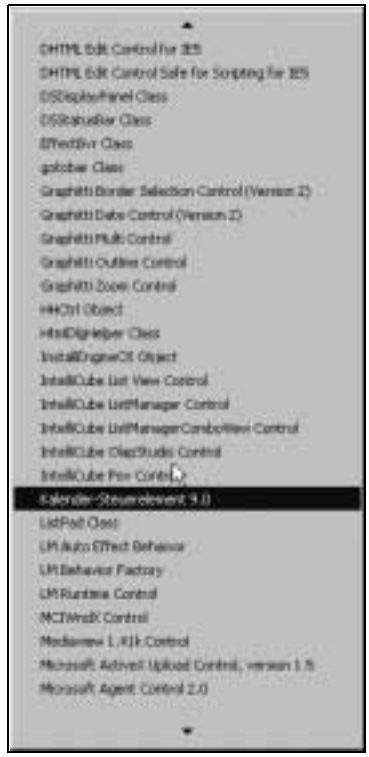


Bild 17.2: Liste der ActiveX-Steuerelemente, die nach dem Klicken auf die Symbol-Schaltfläche »Weitere Steuerelemente« in der Toolbox eingeblendet wird. Beachten Sie den Pfeil am unteren Rand, über den diese Liste gerollt werden kann.

- Klicken Sie in der Liste auf das Kalender-Steuerelement, und ziehen Sie dann ein Rechteck von ungefähr 4 x 5 cm Größe, mit der linken oberen Ecke bei 3 cm Höhe und 9 cm Breite.
- Wiederholen Sie den letzten Schritt, um das zweite Kalender-Steuerelement einzufügen, nur diesmal mit der linken oberen Ecke 6,5 cm Höhe und 10 cm Breite.
- Fügen Sie für die beiden Kalender-Steuerelemente je ein Bezeichnungsfeld hinzu, um sie mit den Beschriftungen *Geburtsdatum* und *Einstellungsdatum* zu kennzeichnen. Der Formularentwurf sollte dann aussehen, wie in Bild 17.3 dargestellt.
- Binden Sie das obere Kalender-Steuerelement an das Tabellenfeld *Geburtsdatum*: Blenden Sie ggf. das Eigenschaftenfenster ein, und markieren Sie dann das obere Kalender-Steuerelement. Klicken Sie im Eigenschaftenfenster in das Einstellungsfeld der Eigenschaft *Steuerelementinhalt*, und wählen Sie dort mit der Dropdown-Liste das Feld *Geburtsdatum*.

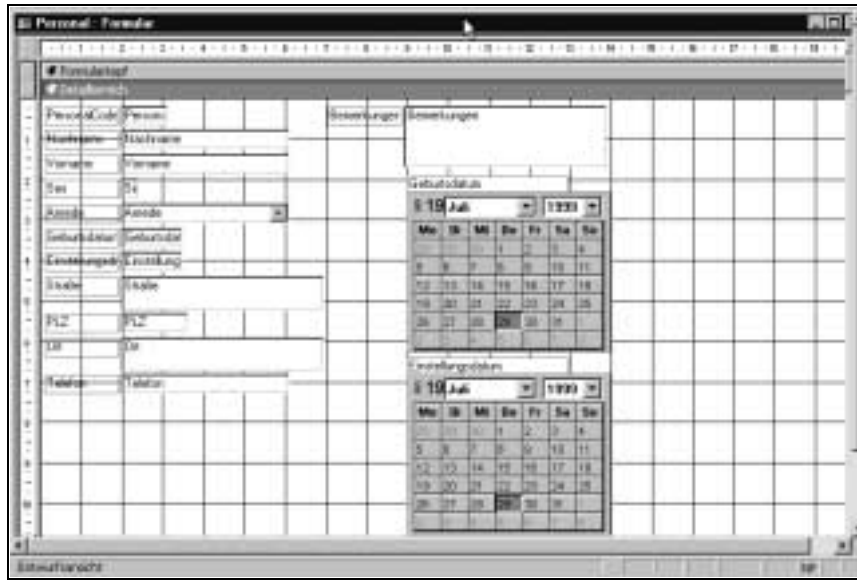


Bild 17.3: Formular »Personal« mit eingefügten, aber unbearbeiteten Kalender-Steuerelementen und Bezeichnungsfeldern

- Binden Sie das untere Kalender-Steuerelement an das Tabellenfeld *Einstellungsdatum*: Verfahren Sie entsprechend wie im letzten Schritt.

Bild 17.4: Formular »Personal« mit eingefügten Kalender-Steuerelementen in der Formularansicht

- Wechseln Sie nunmehr in die Formularansicht, und probieren Sie die Funktionsweise der neuen Steuerelemente (vgl. Bild 17.4). Ändern Sie beispielsweise das Einstellungsdatum durch Klicken auf die entsprechenden Elemente des unteren Kalenders. Wenn Sie danach auf ein anderes Steuerelement klicken, um diesem den Fokus zu geben, sollte das neue Einstellungsdatum auch im entsprechenden Textfeld angezeigt werden. Dieser Effekt liegt daran, daß das Textfeld *Einstellungsdatum* bereits vorher an das gleichnamige Tabellenfeld gebunden war und damit Änderungen des Tabellenwertes, der vom Kalender-Steuerelement bewirkt wurde, widerspiegelt. (In einem realistischen Fall würde man das Textfeld sicherlich für die Eingabe sperren, wenn es denn überhaupt im Formular verbliebe.)

Die beiden Kalender-Steuerelemente weisen noch Schönheitsfehler auf: Sie zeigen am linken Rand ihrer Titelleiste einen verstümmelten Eintrag. Die Verstümmelung ist darauf zurückzuführen, daß die Beschriftung innerhalb der bestehenden Größe der Kalender nicht wiedergegeben werden kann. Entweder müßte die Schriftgröße insoweit verkleinert oder der Rahmen vergrößert oder die Titelbeschriftung unterdrückt werden. Hier soll das Letztere geschehen. Um eine derartige Änderung herbeizuführen, muß sicherlich eine entsprechende Eigenschaft der beiden Kalender-Steuerelemente in ihrer Einstellung geändert werden. Gehen Sie daher wie folgt vor:

- Wechseln Sie in die Entwurfsansicht des Formulars, und markieren Sie das Kalender-Steuerelement zum Geburtsdatum. Blenden Sie ggf. das Eigenschaftenfenster ein, und suchen Sie nach einer Eigenschaft, die die Titelbezeichnung betrifft. Unter den Standard-Eigenschaften der Steuerelemente, die im normalen Eigenschaftenfenster angeboten werden, findet sich keine diesbezügliche. Die meisten ActiveX-Steuerelemente besitzen jedoch spezifische Eigenschaften, die in einem besonderen Eigenschaftenfenster eingestellt werden können. Dieses öffnen Sie durch Doppelklicken auf das betreffende ActiveX-Steuerelement. Wenn Sie dieses für das obere Kalender-Steuerelement ausführen, wird das Eigenschaftenfenster eingeblendet, das in Bild 17.5 wiedergegeben ist.
- Schalten Sie im Eigenschaftenfenster des Kalender-Steuerelements das Kontrollkästchen *Monat/Jahr-Titel* aus. Ändern Sie ggf. weitere, Ihren Wünschen entsprechende Eigenschaften in ihren Einstellungen, und klikken Sie dann auf die Schaltfläche *Übernehmen*, bevor Sie das Dialogfeld mit OK schließen.
- Verfahren Sie für das zweite Kalender-Steuerelement entsprechend wie für das erste.

Nach diesen Änderungen sollten die beiden Kalender ordentlich aussehen. Entscheiden Sie, ob Sie das Formular *Personal* mit den eingefügten Kalender-Steuerelementen speichern wollen. In der Datenbank *Projekte.mdb* ist es ohne die beiden Kalender gespeichert, weil diese hier nur aus Übungsgründen eingefügt und bearbeitet wurden.

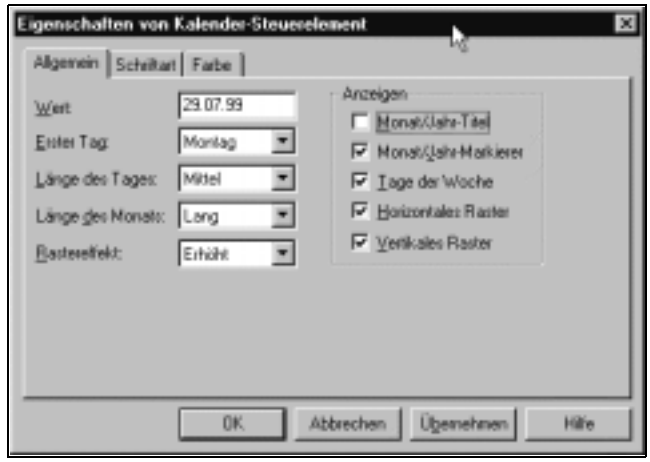


Bild 17.5: Eigenschaftenfenster des Kalender-Steuerelements

Nicht alle ActiveX-Steuerelemente besitzen ein zusätzliches Eigenschaftenfenster. Um herauszufinden, ob ein Steuerelement ein zusätzliches Eigenschaftenfenster bereitstellt, suchen Sie im Access-Eigenschaftenfenster für dieses Steuerelement nach der Eigenschaft *Benutzerdefiniert*. Wenn diese Eigenschaft in der Liste der Eigenschaften enthalten ist, besitzt das Steuerelement das zusätzliche Eigenschaftenfenster. Dieses können Sie übrigens auch öffnen, indem Sie auf die Generator-Schaltfläche mit den drei Punkten rechts neben dem Einstellungsfeld zur Eigenschaft *Benutzerdefiniert* klicken.

17.3 Registrieren eines ActiveX-Steuerelements

Wenn Sie ein neues ActiveX-Steuerelement erhalten, das noch nicht auf Ihrem PC installiert ist, sollten Sie dieses auf die folgende Weise installieren und registrieren:

- Kopieren Sie die Dateien für das ActiveX-Steuerelement in einen Ordner Ihrer Festplatte. Ob Sie dabei bestimmte Laufwerke und/oder Ordner beachten müssen, wird sich in der Dokumentation des ActiveX-Steuerelements finden.
- Starten Sie ggf. Access, und wählen Sie im Menü *Extras* den Befehl *ActiveX-Steuerelemente*. Dann wird das Dialogfeld *ActiveX-Steuerelemente* angezeigt, vgl. Bild 17.6.
- Klicken Sie im Dialogfeld *ActiveX-Steuerelemente* auf die Schaltfläche *Registrieren*, wählen Sie im dann eingeblendeten Dialogfeld *ActiveX-Steuerelement hinzufügen* die Datei, die das ActiveX-Steuerelement darstellt, und klicken Sie anschließend auf die Schaltfläche *OK*.
- Klicken Sie im Dialogfeld *ActiveX-Steuerelemente* auf die Schaltfläche *Schließen*.

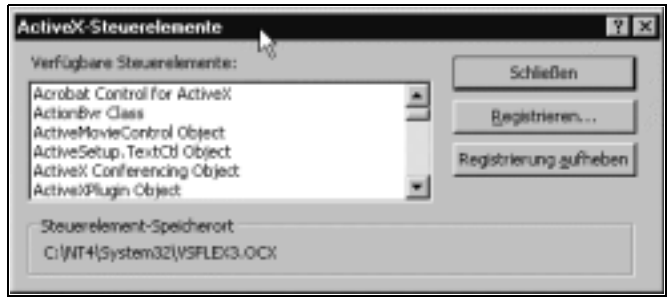


Bild 17.6: Dialogfeld »ActiveX-Steuerelemente«

Mit dieser Prozedur haben Sie dafür gesorgt, daß das neue ActiveX-Steuerelement in der Windows-Registrierung eingetragen wurde. Nach dem nächsten Start von Access wird es dann – wie auch in den anderen Office 2000-Anwendungen – zur Verfügung stehen.

Kapitel 18

Haupt- und Unterformular

In diesem Abschnitt wird gezeigt, wie Sie Formulare auf mehr als einer Tabelle basieren können. Es gibt zwei Möglichkeiten, dies zu erreichen:

- Sie erstellen eine Mehrtabellenabfrage und geben diese als Datenherkunft für das Formular an. Dies ist häufig die effektivste Art, die Daten mehrerer Tabellen in einem Formular zu verwalten, weil der Formularaufbau dabei verhältnismäßig einfach ist.
- Die andere Möglichkeit besteht darin, in ein (Haupt-)Formular ein oder mehrere Unterformulare einzubetten. Dieses Verfahren bietet den Vorteil, daß jedes Unterformular gegenüber dem Hauptformular eine gewisse Eigenständigkeit und damit eigene Gestaltbarkeit besitzt. So können Sie gezielt für ein Unterformular zwischen der Datenblattansicht und der Formularansicht wechseln.

In diesem Kapitel stehen die verschiedenen Verfahren, mit Haupt- und Unterformularen umzugehen, sowie verschiedene Beispiele dazu im Vordergrund. Am Ende des Kapitels wird jedoch auch gezeigt, daß und wie Sie ein n:1-Formular erstellen, dem eine Abfrage zugrunde liegt. Das grundsätzliche Konzept, nach dem Access Formulare als Unterformulare organisiert und verwaltet, wurde bereits in Kap. 16, *Standard-Steuerelemente in Formularen*, Punkt 16.7, *Unterformular*, erklärt. Zum tieferen Verständnis der Zusammenhänge sowie zur Bearbeitung von Unterformularen sollten Sie die dortigen Ausführungen nachlesen.

18.1 Verfahren zum Erstellen von Haupt- und Unterformularen

Sie können ein Hauptformular mit einem eingebetteten Unterformular auf dreierlei Arten erstellen:

Formular-Assistent

Wenn sowohl das Haupt- wie auch das Unterformular gänzlich neu erstellt werden sollen, nutzen Sie am besten die Bequemlichkeit des Formular-Assistenten. Ein Beispiel für das Erstellen eines neuen Haupt- und Unterformulars finden Sie in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, Punkt 5.11, *Formular Personal und Projekte erstellen*.

Toolbox

Mit der Toolbox können Sie in ein Hauptformular ein Unterformular einfügen. Dieses Verfahren, bei dem Sie am wenigsten von Access »geführt« werden und das daher das größte Verständnis der Zusammenhänge voraussetzt, wird in Kap. 16, *Standard-Steuerelemente in Formularen*, Punkt 16.7, *Unterformular*, geschildert.

Ziehen aus dem Datenbankfenster

Wenn sowohl das Formular, welches als Hauptformular fungieren soll, wie auch dasjenige, welches als Unterformular eingefügt werden soll, bereits als einzelne Formulare, die gespeichert wurden, existieren, ist es am bequemsten, das Unterformular durch Ziehen aus dem Datenbankfenster dem Hauptformular hinzuzufügen. Dieses Verfahren wird im folgenden Punkt beschrieben.

18.2 Ein Unterformular durch Ziehen aus dem Datenbankfenster einfügen

Um einem Formular ein anderes durch Ziehen aus dem Datenbankfenster hinzuzufügen, gehen Sie wie folgt vor:

- Vergewissern Sie sich, daß die beiden Formulare (künftiges Hauptformular und künftiges Unterformular) bereits als gespeicherte Formulare existieren, so daß sie in der Formularliste des Datenbankfensters aufgeführt werden. Außerdem müssen die Formulare sich natürlich auch inhaltlich entsprechen, so daß die Datensätze des Unterformulars mit denen des Hauptformulars verknüpft werden können. Dies setzt voraus, daß die Datenbasis des Hauptformulars einen Primärschlüssel hat, der in der Datenbasis des Unterformulars als Fremdschlüssel auftaucht.
- Öffnen Sie das als Hauptformular vorgesehene Formular in der Entwurfsansicht.
- Ordnen Sie das in der Entwurfsansicht geöffnete Formular so auf dem Bildschirm an, daß einerseits die Objektliste *Formulare* des Datenbankfensters, andererseits die Einfügestelle im Unterformular sichtbar sind. In Bild 18.1 finden Sie eine solche Fensteranordnung.

- Ziehen Sie das Symbol des Formulars, welches als Unterformular eingefügt werden soll, aus der Formularliste des Datenbankfensters an die Stelle im Hauptformular, an der das Unterformular eingefügt werden soll. Access fügt ein Steuerelement *Unterformular* ein, das Sie als mehr oder minder große Fläche erkennen können. Access fügt das Steuerelement *Unterformular* in einer Größe ein, die der Größe des eingefügten Formulars entspricht.
- Wechseln Sie in die Formularansicht, um sich das Ergebnis zu betrachten. Wechseln Sie ggf. anschließend wieder in die Entwurfsansicht, um Änderungen am Unterformular vorzunehmen.

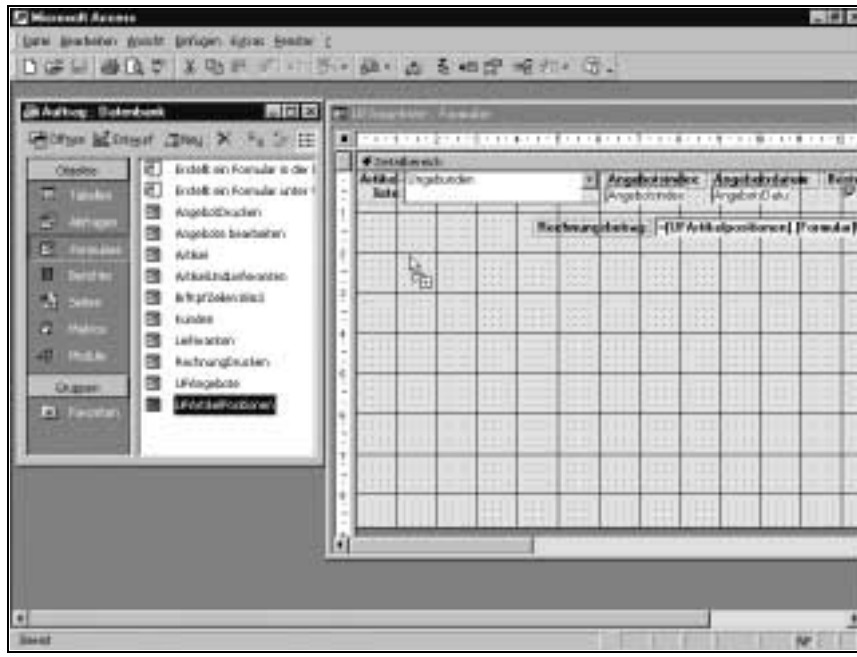


Bild 18.1: Das Formular mit dem Namen »UFArtikelPositionen« wird aus dem Datenbankfenster in das Formular »UFAngebote« gezogen.

18.3 Automatisches und manuelles Verknüpfen von Haupt- und Unterformular

Access verknüpft (vgl. die Eigenschaften *Verknüpfen von* und *Verknüpfen nach* des Steuerelements *Unterformular*) Haupt- und Unterformular unter gewissen Bedingungen automatisch, wenn dem Hauptformular ein Unterformular durch Ziehen hinzugefügt wird oder Haupt- und Unterformular mit dem Formular-Assistenten erstellt werden. Eine automatische Verknüpfung wird in den folgenden Konstellationen durchgeführt:

- Haupt- und Unterformular beruhen auf Tabellen, zwischen denen eine Beziehung (1:n oder 1:1) definiert ist. Dann werden der Primärschlüssel der Mastertabelle und der Fremdschlüssel der Detailtabelle als Verknüpfungsfelder verwendet.
- Haupt- und Unterformular beruhen auf Tabellen oder Abfragen, die Felder mit identischen Namen und Datentypen enthalten, wobei das Feld in der Datenbasis des Hauptformulars Primärschlüssel ist. Dann werden diese beiden Felder zum Verknüpfen eingesetzt.

Sie sollten in jedem Falle nach dem Einfügen eines Unterformulars durch Ziehen aus dem Datenbankfenster oder nach dem Erstellen eines Haupt- und Unterformulars mit dem Formular-Assistenten prüfen, ob Access die von Ihnen beabsichtigte Verknüpfung vorgenommen hat: Öffnen Sie dazu das Hauptformular in der Entwurfsansicht, markieren Sie das Steuerelement *Unterformular,* und prüfen Sie, wie die Eigenschaften *Verknüpfen von* und *Verknüpfen nach* eingestellt sind. Bearbeiten Sie diese dann, falls erforderlich.

18.4 Ansichten des Unterformulars

Ein Unterformular kann im Hauptformular in den beiden Ansichten erscheinen, die auch für ein singuläres Formular möglich sind: Datenblatt- und Formularansicht, vgl. Bild 18.2 und Bild 18.3, die das Formular *Angebote bearbeiten* aus der Datenbank *Auftrag.mdb* von der Begleit-CD-ROM mit den beiden Ansichten für das Unterformular im unteren Teil des Hauptformulars wiedergeben.

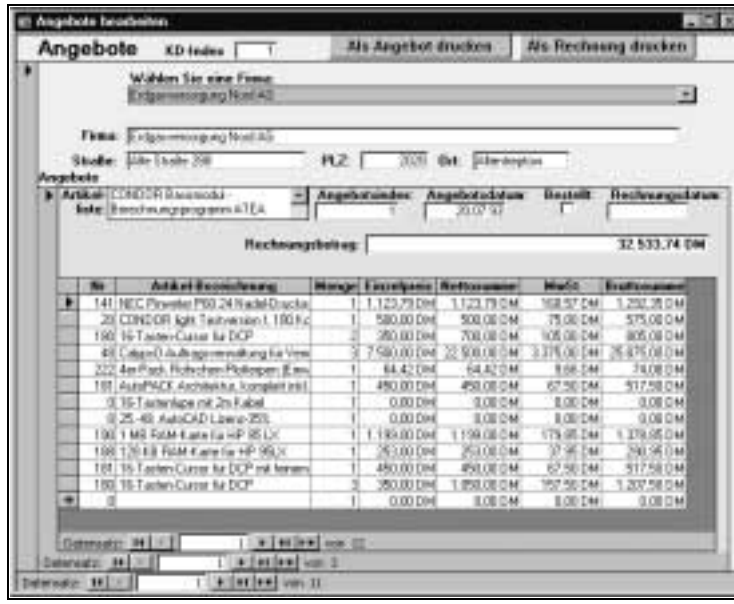


Bild 18.2: Hauptformular Angebote bearbeiten aus der Datenbank Auftrag.mdb: Unterformular »UFArtikelPositionen« in Datenblattansicht.

Umschaltmöglichkeit und Standardansichten

Zwischen der Datenblatt- und der Formularansicht schalten Sie folgendermaßen um:

- Sorgen Sie dafür, daß ein Feld im Unterformular den Fokus hat.
- Wählen Sie den Befehl *Unterformulardatenblatt* aus dem Menü *Ansicht*. Wenn dieser Befehl bisher kein Häkchen trägt, wird in die Datenblattansicht umgeschaltet, andernfalls in die Formularansicht.

Diese Wahlmöglichkeit wird allerdings nur angeboten, wenn für das betreffende Unterformular die Formulareigenschaft *Zugelassene Ansicht(en)* auf den Wert *Beide* eingestellt ist. Wenn Sie andererseits die Umschaltmöglichkeit verhindern wollen, müssen Sie diese Eigenschaft auf einen der beiden Werte *Formular* oder *Datenblatt* einstellen.

In welcher Ansicht das Unterformular unmittelbar nach dem Öffnen des Hauptformulars erscheint, bestimmen Sie mit der Eigenschaft *Standardansicht* des Unterformulars: Setzen Sie diese auf einen der Werte *Datenblatt*, *Einzelnes Formular* oder *Endlosformular.*

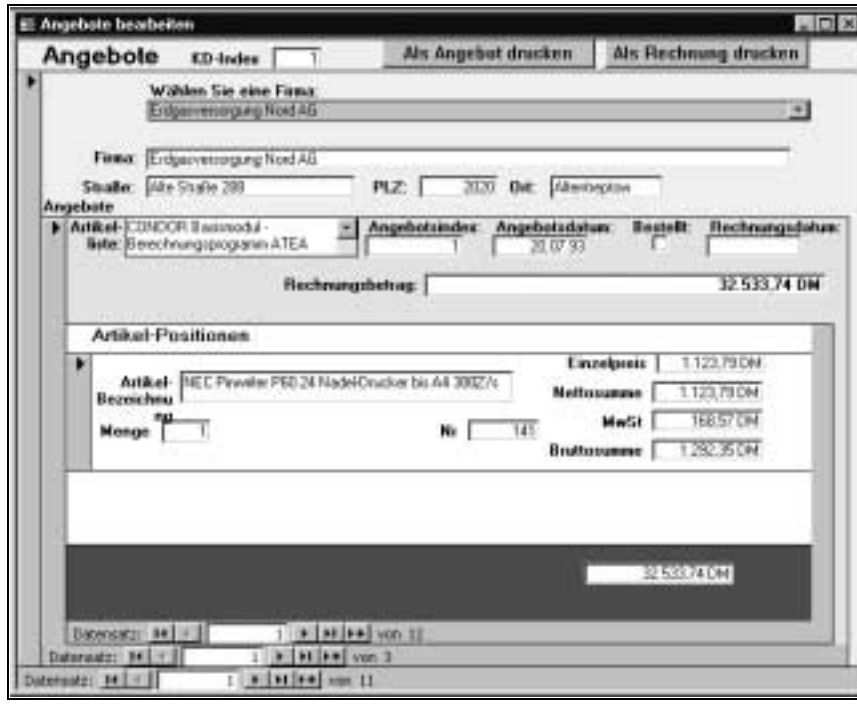


Bild 18.3: Hauptformular Angebote bearbeiten aus der Datenbank Auftrag.mdb: Unterformular »UFArtikelPositionen« in Formularansicht.

18.5 Hauptformular mit einem verknüpften Unterformular im verknüpften Unterformular

Einem Formular können viele andere Formulare hinzugefügt werden. Dabei ist es auch möglich, daß in ein Formular ein verknüpftes Unterformular eingefügt wird, welches selbst ein Unterformular enthält, mit dem es verknüpft ist. Diese Art von Verschachtelung ist allerdings auf drei Stufen (Hauptformular → Unterformular → darin eingebettetes Unterformular) beschränkt.

Beispiel

Die Datenbank *Auftrag.mdb* von der Begleit-CD-ROM enthält ein solches Formular. Sie finden es unter dem Formularnamen *Angebote bearbeiten*, vgl. auch oben Bild 18.2 und Bild 18.3. Das Hauptformular *Angebote bearbeiten* hat als Datenherkunft die Tabelle *Kunden*. Das Unterformular *UFAngebote*, welches unmittelbar in das Hauptformular eingebettet ist, basiert auf der Tabelle *Angebote*. Für das Unterformular *UFArtikelPositionen*, welches in das Formular *UFAngebote* eingebettet ist, ist die Datenherkunft die Tabelle *ArtikelPositionen*. Die für die drei Tabellen bestehenden 1:n-Beziehungen erkennen Sie in Bild 18.4.

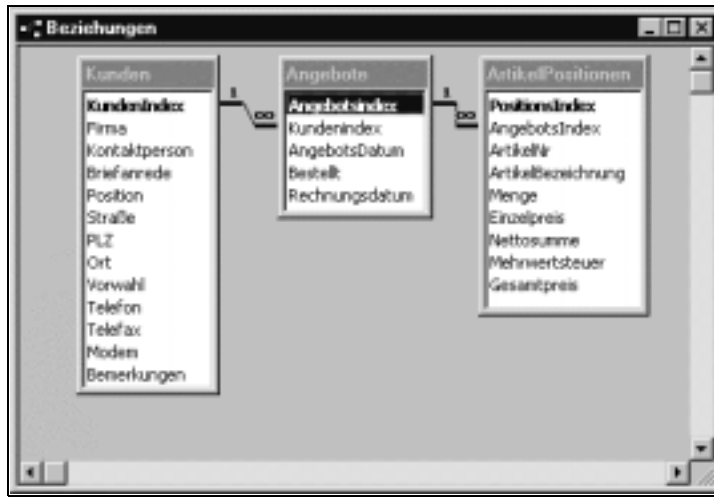


Bild 18.4: Beziehungen zwischen den drei Tabellen, die den drei Teilformularen des Formulars Angebote bearbeiten zugrunde liegen. Die Beziehungen sind jeweils vom Typ 1: n mit referentieller Integrität.

Den Tabellenbeziehungen entsprechen die folgenden Formularverknüpfungen:

	Hauptformular	Unterformular 1	Unterformular 2
Formularname:	Angebote bearbeiten	UFAngebote	UFArtikelPositionen
Tabelle:	Kunden	Angebote	ArtikelPositionen
Verknüpft nach:		Kundenindex	Angebotsindex
Verknüpft von:		Kundenindex	Angebotsindex

Wie das dreifach geschachtelte Formular funktioniert

Das Formular *Angebote bearbeiten* aus der Datenbank *Auftrag.mdb* funktioniert folgendermaßen: Für jedes der drei Formulare ist eine waagerechte Bildlaufleiste mit den Navigationssymbolen verfügbar: Die unterste bezieht sich auf das Hauptformular, die mittlere auf das erste Unterformular (hier: *UFAngebote*) und die innerste auf das zweite Unterformular, das in das erste eingebettet ist (hier: *UFArtikelPositionen*). In den Unterformularen sind stets nur die Datensätze verfügbar, die zu dem Datensatz im jeweils (hierarchisch) darüber liegenden Formular, der den Fokus hat, passen. Wenn Sie beispielsweise im Hauptformular zu einem neuen Kunden blättern, zeigen auch die beiden Unterformulare die dem Kunden entsprechenden Datensätze neu an. Wenn Sie im ersten Unterformular zu einem neuen Angebot blättern, werden im darunterliegenden Unterformular die dem Angebot entsprechenden ArtikelPositionen eingeblendet, während das Hauptformular den bestehenden Kunden weiter anzeigt. Blättern im untersten Unterformular mit den Artikelpositionen bleibt wirkungslos für die beiden anderen Teilformulare.

Hauptformular mit verknüpftem Unterformular im verknüpften Unterformular erstellen

Gehen Sie wie folgt vor:

- *Unteres Unterformular.* Richten Sie zunächst das Formular ein (neu erstellen und/oder bearbeiten), welches als (hierarchisch) unteres Unterformular dienen soll. Im obigen Beispiel ist dies das Formular mit dem Namen *UFArtikelPositionen*. Bestimmen Sie ggf. mit den Eigenschaften *Zugelassene Ansicht(en)* und *Standardansicht*, in welcher Ansicht das Unterformular im Hauptformular erscheinen soll. Speichern Sie schließlich.
- *Oberes Unterformular.* Richten Sie das Formular ein (neu erstellen und/oder bearbeiten), welches als (hierarchisch) oberes Unterformular dienen und das unterste aufnehmen soll. Im obigen Beispiel ist dies das Formular mit dem Namen *UFAngebote*. Lassen Sie genügend Platz zur Aufnahme des untersten Unterformulars. Ziehen Sie das unterste Unterformular (im obigen Beispiel: *UFArtikelPositionen*) aus dem Datenbankfenster an die Stelle, an der es erscheinen soll. Überprüfen Sie, ob Access die von Ihnen

gewünschte Verknüpfung vorgenommen hat. Markieren Sie zu diesem Zweck das gerade implizit mit dem Ziehen aus dem Datenbankfenster eingefügte Steuerelement *Unterformular*, und schauen Sie im Eigenschaftenfenster nach den Eigenschaften *Verknüpfen von* und *Verknüpfen nach*. Tragen Sie ggf. die Namen der Verknüpfungsfelder ein. Im obigen Beispiel ist dies für beide Verknüpfungsrichtungen der Name *Angebotsindex*. Bestimmen Sie ggf. mit den Eigenschaften *Zugelassene Ansicht(en)* und *Standardansicht*, in welcher Ansicht dieses Unterformular im Hauptformular erscheinen soll. Speichern Sie schließlich.

- *Hauptformular.* Richten Sie das Formular ein (neu erstellen und/oder bearbeiten), welches als Hauptformular dienen und das obere Unterformular aufnehmen soll. Im obigen Beispiel ist dies das Formular mit dem Namen *Angebote bearbeiten*. Ziehen Sie das obere Unterformular (im obigen Beispiel: *UFAngebote*) aus dem Datenbankfenster an die Stelle, an der es im Hauptformular erscheinen soll. Überprüfen Sie, ob Access die von Ihnen gewünschte Verknüpfung vorgenommen hat. Markieren Sie zu diesem Zweck das gerade eingefügte Steuerelement *Unterformular* und schauen Sie im Eigenschaftenfenster nach den Eigenschaften *Verknüpfen von* und *Verknüpfen nach*. Tragen Sie ggf. die Namen der Verknüpfungsfelder ein. Im obigen Beispiel ist dies für beide Verknüpfungsrichtungen der Name *Kundenindex*. Speichern Sie schließlich.

In einem Formular, das ausschließlich in der Datenblattansicht gezeigt werden soll, brauchen Sie sich beim Positionieren der Steuerelemente (Textfelder, Kontrollkästchen etc.) keine allzu große Mühe zu geben, weil die Felder für die Datenblattansicht von Access ohnehin automatisch angeordnet werden. Die Feldreihenfolge im Sinne des Befehls *Aktivierreihenfolge* aus dem Menü *Ansicht* ist dagegen bedeutsam.

18.6 n:1-Formular erstellen

Wenn die Datensätze zweier Tabellen, zwischen denen eine 1:n-Beziehung besteht, in einem Haupt- mit Unterformular dargestellt werden sollen, stellt sich prinzipiell die Frage, ob die 1-Seite im Haupt- und die n-Seite im Unterformular repräsentiert sein soll oder umgekehrt. Die bisherigen Beispiele gingen von Ersterem aus. Auch die Verknüpfungsroutinen, mit denen Access beim Ziehen aus dem Datenbankfenster oder beim Gebrauch des Formular-Assistenten automatische Verknüpfungen zwischen Haupt- und Unterformular einrichtet, unterstellen diesen Fall: Der Formular-Assistent beispielsweise läßt es gar nicht zu, daß Sie die n-Seite zweier Tabellen als Basis für ein Hauptformular angeben; er erstellt in diesem Falle ein Formular ohne Unterformular, dessen Datenbasis eine Mehrtabellenabfrage ist. Entsprechend reagiert Access beim Ziehen solcher Formulare aus dem Datenbankfenster: Es wird keine Verknüpfung zwischen Haupt- und Unterformular hergestellt.

Gleichwohl ist es möglich, ein Formular zu erstellen, das sich auf die Datensätze der n-Seite einer Tabelle konzentriert und gleichzeitig den zugehörigen der 1-Seite wiedergibt. Sie können dies auf zwei Weisen, die allerdings nicht gleichwertig sind, realisieren:

- *Haupt- und Unterformular.* Der Vorteil dieser Lösung besteht darin, daß Sie das Unterformular – wie üblich – gegenüber dem Hauptformular eigenständig gestalten können. Dafür müssen Sie allerdings selbst die Verknüpfung im Steuerelement *Unterformular* des Hauptformulars vornehmen, weil dies durch Access nicht automatisch erfolgt.
- *Formular ohne Unterformular, basierend auf einer Abfrage*. Diese Lösung ist im allgemeinen die einfachere und effektivere.

Im folgenden wird nur auf die Abfrage-Lösung eingegangen, die an einem Beispiel demonstriert wird. Das Beispiel ist auch in der Datenbank Projekte.mdb der Begleit-CD-ROM realisiert.

Beispiel eines n:1-Formulars

Wenn Sie das folgende Beispiel in der Datenbank *Projekte.mdb* praktisch nacharbeiten wollen, verwenden Sie bitte andere Namen für die beiden Formulare und die Abfrage, als hier angegeben, weil diese dort bereits existieren. Das zu erstellende Formular soll den Namen *Projekte und Personal* bekommen und in der Formularansicht so aussehen, wie in Bild 18.5 wiedergegeben. Es zeigt Felder der Datensätze aus den Tabellen *Projekte* (dies ist die n-Seite) sowie *Personal* (dies ist die 1-Seite) an. Seine Datenbasis soll die entsprechend konzipierte Abfrage mit dem Namen *Projekte und Personal* sein.

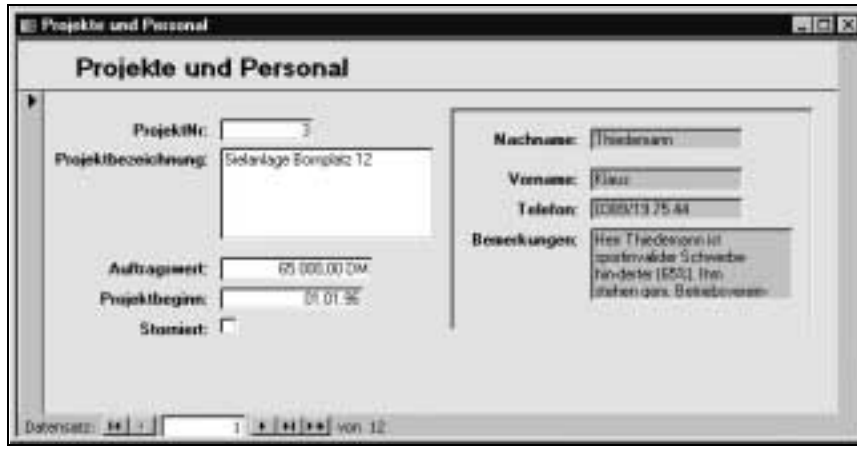


Bild 18.5: Fertiges Formular »Projekte und Personal«

Gehen Sie zur Lösung der Aufgabe wie folgt vor:

- Öffnen Sie ggf. die Datenbank *Projekte.mdb*.
- Erstellen Sie eine neue Abfrage, indem Sie im Datenbankfenster die Registerkarte *Abfragen* aktivieren und dann die Schaltfläche *Neu* wählen.
- Fügen Sie die Tabellen *Personal* sowie *Projekte* der Abfrage hinzu.
- Ziehen Sie die Felder *ProjektNr*, *Projektbezeichnung*, *Auftragswert*, *Projektbeginn* und *Storniert* aus der Feldliste *Projekte* in den Entwurfsbereich.
- Ziehen Sie die Felder *Nachname*, *Vorname*, *Telefon* und *Bemerkungen* aus der Feldliste *Personal* in den Entwurfsbereich.

Bild 18.6: Abfrageentwurf »Projekte und Personal«

- Geben Sie für das Feld *Projektbeginn* eine Sortierung in aufsteigender Richtung an. Ihr Abfrageentwurf sollte jetzt so aussehen wie in Bild 18.6.
- Betrachten Sie die Abfrage in der Datenblattansicht. Wenn die Ergebnisse prinzipiell so aussehen wie in Bild 18.7, speichern Sie die Abfrage unter dem Namen *Projekte und Personal*.

Bild 18.7: Datenblattansicht der Abfrage »Projekte und Personal«

- Erstellen Sie ein neues Formular. Geben Sie dafür als Datenherkunft die gerade erstellte Abfrage *Projekte und Personal* an. Nehmen Sie alle Felder der Abfrage in das Formular auf.
- Bearbeiten Sie die einzelnen Steuerelemente so, daß das Formular ungefähr so aussieht wie oben in Bild 18.5. Um die Felder der 1-Seite (Tabelle *Personal*) von denen der n-Seite (Tabelle *Projekte*) visuell abzugrenzen, können Sie die ersteren auf einem Rechteck-Hintergrund plazieren: Ziehen Sie um die Felder ein Rechteck, geben Sie diesem einen grauen Hintergrund, und setzen Sie das Rechteck mit dem Befehl *In den Hintergrund* aus dem Menü *Format* in den Hintergrund.
- Speichern Sie das fertige Formular unter dem Namen *Projekte und Personal*.

Kapitel 19

PopUp- und Dialogfeldformulare

Wenn ein Formular in der Formularansicht geöffnet ist, können Sie normalerweise alle Bearbeitungen darin vornehmen. Ferner ist es möglich, anderen Objekten den Fokus zu geben (beispielsweise zu einem anderen Formular zu wechseln) und an ihnen Bearbeitungen ausführen, ohne das erste Formular vorher schließen zu müssen. Mit Hilfe der Formulareigenschaften *PopUp* und *Gebunden*, die standardmäßig auf *Nein* eingestellt sind, können diese Freiheiten eingeschränkt werden. Auf diese Weise können Sie beispielsweise ein Formular zu einem Dialogfeld machen, das erst geschlossen werden muß, bevor Sie eine andere Aufgabe beginnen oder weiterführen können.

Wenn Sie nur eine kurze Meldung, z.B. eine Warnung oder einen Hinweis, anzeigen möchten, können Sie dafür ein vordefiniertes Dialogfeld, das sogen. *Meldungsfeld*, verwenden. Erstellen Sie zu diesem Zweck ein Makro mit der Aktion *Meldung* oder eine VBA-Prozedur mit der Aktion oder Funktion *MsgBox*, und rufen Sie das Makro oder die Prozedur an geeigneter Stelle auf.

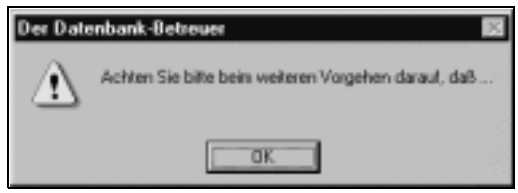


Bild 19.1: Einfaches Meldungsfeld

Die Formulareigenschaften *PopUp* und *Gebunden* können unabhängig voneinander eingestellt werden:

PopUp: Ja

Wenn Sie die Eigenschaft *PopUp* auf *Ja* einstellen, bedeutet dies, daß das betreffende Formular im Vordergrund bleibt, solange es in der Formularansicht geöffnet ist. Anders ausgedrückt: Es verdeckt alle anderen Fenster, das Datenbankfenster eingeschlossen, soweit seine Größe reicht. Allerdings können Sie in diesem Zustand Menübefehle (und von Symbol-Schaltflächen ausgelöste Aktionen) ausführen, d.h., Menü- und Symbolleisten sind aktivierbar, und es können auch andere geöffnete Formulare bearbeitet werden.

Gebunden: Ja

Wenn Sie die Eigenschaft *Gebunden* auf *Ja* einstellen, bedeutet dies, daß nur Menü- oder Symbolbefehle ausgeführt werden können, die dieses Formular betreffen und darüber hinaus allgemeine Befehle. Es stehen jedoch keine Befehle zur Verfügung, die sich auf andere Formulare beziehen. Insbesondere können Sie kein anderes Formular öffnen oder einem geöffneten den Fokus geben, solange das Formular mit der Eigenschaftseinstellung *Gebunden = Ja* geöffnet ist.

Beide Eigenschaften: Ja

In diesem Falle ergibt sich die Wirkung eines echten Dialogfeldes: Sie können keine Aktion an einem anderen Objekt als dem geöffneten Dialogfeld-Formular vornehmen. Wenn Sie auf irgendeinen Punkt außerhalb des Dialogfeld-Formulars klicken, bleibt dies wirkungslos. Sie müssen das Dialogfeld-Formular schließen, um andere Aktionen ausführen zu können. Da auch der ansonsten für ein geöffnetes Formular verfügbare Befehl *Schließen* aus dem Menü *Datei* – wie alle anderen Befehle auch – nicht verfügbar ist, müssen Sie das Formular über sein Systemmenü schließen oder eine Befehlsschaltfläche mit einem Makro, das die Aktion *Schließen* ausführt (oder eine entsprechende Ereignisprozedur) in diesem Formular vorsehen.

Die folgende Tabelle gibt Ihnen einen Überblick über die jeweiligen Wirkungen der beiden Eigenschaften, wenn Sie diese auf *Ja* einstellen.

Wirkungen bei PopUp- und/oder gebundenen Formularen

	PopUp = Ja	**Gebunden = Ja**
Formular ist verschiebbar	Ja	Ja
Formulargröße ist änderbar	Nein	Ja
Formular verbleibt oberhalb anderer Objekte, bis Sie dieses schließen oder verbergen	Ja	Ja

	PopUp = Ja	**Gebunden = Ja**
Menübefehle können gewählt werden, während das Formular geöffnet ist	Für das PopUp-Formular: Nein; für andere Objekte: Ja	Für das gebundene Formular: Ja; für andere Objekte: Nein

Ein als Dialogfeld geöffnetes Formular ist im allgemeinen nur sinnvoll im Zusammenhang mit Makros oder VBA-Prozeduren, die Aktionsabläufe steuern und dabei bestimmte Benutzerangaben mit Hilfe eines Dialogfeldes abfragen. Dabei ergibt sich die folgende Schwierigkeit: Sie kommen, nachdem ein Formular als Dialogfeld geöffnet wurde, im Programmablauf erst weiter, wenn es wieder geschlossen oder ausgeblendet ist. Andererseits wollen Sie die vom Benutzer in das Dialogfeld eingegebenen Werte verarbeiten, von einem geschlossenen Formular lassen sich aber keine Werte mehr auslesen. Daher müssen Sie eine der beiden folgenden Möglichkeiten realisieren:

- Sie binden die Steuerelemente des Dialogfeld-Formulars an eine Tabelle, in der sie automatisch beim Schließen des Formulars gespeichert werden, und lesen später die Tabellenwerte aus.
- Sie blenden das Dialogfeld-Formular aus, indem Sie seine Eigenschaft *Sichtbar* (*Visible*) auf *Nein* (*False*) setzen. Dies können Sie z.B. erreichen, indem Sie eine Befehlsschaltfläche zum Schließen vorsehen, die *Beim Klikken* ein entsprechendes Ereignismakro oder eine entsprechende Ereignisprozedur aufruft. Auf die Steuerelemente eines ausgeblendeten Formulars können Sie lesend zugreifen, weil es nach wie vor geöffnet, nur nicht sichtbar, ist.

Mit einem Makro oder einer VBA-Prozedur können Sie ein Formular auch vorübergehend, d.h. ohne die beiden Eigenschaften *PopUp* und *Gebunden* auf *Ja* zu setzen, als Dialogfeld öffnen: Verwenden Sie in einem Makro die Aktion *ÖffnenFormular*, und wählen Sie für das Argument *Fenstermodus* die Einstellung *Dialog*.

Wenn Sie die Formulareigenschaften *PopUp* und/oder *Gebunden* auf *Ja* eingestellt haben und dann von der Entwurfs- in die Formularansicht wechseln, werden diese Eigenschaftsänderungen sofort wirksam, auch dann, wenn Sie die Änderungen noch nicht gespeichert haben. Dies bedeutet, daß Sie nicht wieder zurück in die Entwurfsansicht wechseln können, weil weder die Symbol-Schaltfläche *Ansicht* noch der entsprechende Befehl – wie alle anderen auch – zur Verfügung stehen. In dieser Situation müssen Sie das Formular mit seiner System-Schaltfläche *Schließen* (dem Kreuz am rechten Rand der Titelleiste) schließen. Access fragt Sie dann, ob Änderungen gespeichert werden sollen.

Wenn Sie für ein Dialogfeld-Formular die Eigenschaft *Rahmenart* auf den Wert *Keine* einstellen (die Voreinstellung ist *Veränderbar*) und/oder die Eigenschaft *Schließen Schaltfläche* auf *Nein*, sollten Sie unbedingt eine Befehlsschaltfläche in diesem Formular vorsehen, mit der es geschlossen werden kann: Für ein Dialogfeld-Formular ist der Befehl *Schließen* aus dem Menü *Datei* nicht verfügbar. Gleichzeitig unterdrückt die Eigenschaftseinstellung *Rahmenart* = *Keine* bzw. *Schließen Schaltfläche* =*Nein* mit dem Rahmen auch die Titelleiste bzw. die Schaltfläche *Schließen* und damit die Möglichkeit zum Schließen des Formularfensters. Wenn Sie keine Befehlsschaltfläche zum Schließen vorsehen, können Sie ein derartiges Formular nicht schließen. Sie müßten dann die gesamte Datenbank schließen, um weiter zu kommen. Da hierfür in dieser Situation kein Access-Befehl verfügbar ist, müßten Sie unter Windows die Tastenkombination Strg+Alt+Entf drücken und Access mit dem Task-Manager beenden. Dabei gingen ggf. ungesicherte Änderungen an Access-Objekten verloren.

Eigenschaften mit besonderer Bedeutung für PopUp- und Dialogfeldformulare

Die folgende Übersicht führt Eigenschaften an, die Sie bei PopUp- und Dialogfeldformularen besonders beachten sollten.

Eigenschaft	Zweck
PopUp	Legt fest, ob ein Formular als PopUp-Formular geöffnet wird. Ein PopUp-Formular wird stets im Vordergrund angezeigt.
Gebunden	Legt fest, ob ein Formular als nicht gebundenes Formular (das Wechseln zu anderen Fenstern ist möglich) oder als gebundenes Formular (das Formular behält den Fokus bis zum Schließen) geöffnet wird.
Rahmenart	Legt die für das Formular zu verwendende Rahmenart fest. Wenn Sie diese Eigenschaft auf »Keine« einstellen, weist das Formular auch keine Titelleiste auf, womit auch deren Schaltflächen *Schließen*, *Systemmenü*, *Maximieren* und *Minimieren* verschwinden.
Mit Systemmenüfeld	Legt fest, ob für ein Formular in der Formularansicht das Systemmenü aufgeschlagen werden kann.
MinMaxSchaltflächen	Legt fest, ob ein Formular in der Formularansicht die Schaltflächen *Maximieren* und *Minimieren* anzeigt. Wenn Sie die Eigenschaft *Rahmenart* auf »Dialog« einstellen, blendet Access diese Schaltflächen automatisch aus.

Eigenschaft	Zweck
Schließen Schaltfläche	Legt fest, ob die Schaltfläche *Schließen* aktiviert ist.
Bildlaufleisten	Legt fest, ob ein Formular Bildlaufleisten aufweist.
Navigationsschaltflächen	Legt fest, ob ein Formular Navigationsschaltflächen anbietet.
Datensatzmarkierer	Legt fest, ob ein Formular Datensatzmarkierer enthält.
Kontextmenü	Legt fest, ob ein Kontextmenü angezeigt wird, wenn Sie mit der rechten Maustaste auf ein Formular klicken.
Automatisch zentrieren	Legt fest, ob ein Formular beim Öffnen automatisch im Anwendungsfenster zentriert wird.
Bearbeitungen zulassen	Legt fest, ob Datensätze in dem Formular bearbeitet werden können.
Löschen zulassen	Legt fest, ob Datensätze in dem Formular gelöscht werden können.
Anfügen zulassen	Legt fest, ob Datensätze in dem Formular hinzugefügt werden können.
Daten eingeben	Legt fest, ob ein Formular beim Öffnen alle vorhandenen Datensätze oder einen leeren (neuen) Datensatz anzeigt.
Standardansicht	Legt fest, ob ein Formular unmittelbar nach dem Öffnen in der Datenblattansicht, als Einzelformular oder als Endlosformular angezeigt wird. Nach dem Öffnen kann jedoch in eine andere Ansichtsart gewechselt werden, sofern die Einstellung zur Eigenschaft *Zugelassene Ansicht(en)* (vgl. die folgende Tabellenzeile) dem nicht entgegen steht.
Zugelassene Ansicht(en)	Legt fest, ob Sie zwischen der Formularansicht und der Datenblattansicht wechseln können.

Kapitel 20

Ausdrücke in Formularen

20.1 Ausdruck in Steuerelement eingeben

Mit einem Ausdruck ermitteln Sie einen Wert (Zahlen-, Wahrheits- oder Textwert). In einem Formular können Sie einen Ausdruck z.B. als Einstellung der Eigenschaft *Steuerelementinhalt* eines Steuerelements eingeben. Im allgemeinen ist dies ein Textfeld, jedoch kann die Eigenschaft *Steuerelementinhalt* für jedes Steuerelement, das diese Eigenschaft besitzt, auf einen Ausdruck eingestellt werden. Steuerelemente mit einem Ausdruck als Inhalt der Eigenschaft *Steuerelementinhalt* werden *berechnete Steuerelemente* genannt. Sie können den Wert, den der Ausdruck ergibt, aus dem berechneten Steuerelement auslesen, indem Sie sich in einem anderen Ausdruck auf den Namen des Steuerelements beziehen. Daneben können auch andere Eigenschaften von Steuerelementen auf einen Ausdruck eingestellt werden, beispielsweise die Eigenschaft *Standardwert* oder *Gültigkeitsregel*.

Welche Regeln Sie allgemein bei dem Erstellen von Ausdrücken beachten müssen, und welche allgemeinen Möglichkeiten dafür existieren, erfahren Sie ausführlich in Kap. 39, *Ausdrücke, Funktionen und Operatoren*. In diesem Kapitel konzentriere ich mich auf die Besonderheiten, die Sie beim Verwenden von Ausdrücken für die Eigenschaft *Steuerelementinhalt* in Steuerelementen von Formularen beachten sollten.

20.1.1 Arbeiten mit dem Ausdrucks-Generator

Für komplexere Ausdrücke empfiehlt es sich vielfach, den Ausdrucks-Generator aufzurufen. Gehen Sie dazu folgendermaßen vor:

- Klicken Sie in das Bearbeitungsfeld der Eigenschaft *Steuerelementinhalt* des betreffenden Steuerelements.
- Schlagen Sie den Ausdrucks-Generatorauf, indem Sie auf die Symbol-Schaltfläche *Generator* (mit drei Punkten) am rechten Rand des Bearbeitungsfeldes klicken, vgl. Bild 20.1.

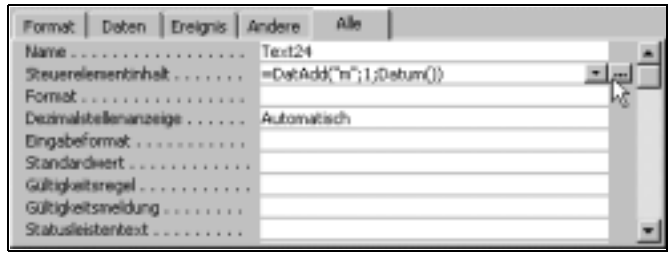


Bild 20.1: Wenn die Eigenschaft »Steuerelementinhalt« den Fokus hat, steht am rechten Rand des Bearbeitungsfeldes die Symbol-Schaltfläche »Generator« zur Verfügung.

- Wählen Sie im Dialogfeld *Ausdrucks-Generator* die geeigneten Operatoren, Funktionen, Objektnamen etc., und bestätigen Sie mit OK, wenn der Ausdruck fertig ist. Sie können auch einen Teil des Ausdrucks vom Generator erstellen lassen und anschließend den Rest über die Tastatur eingeben.

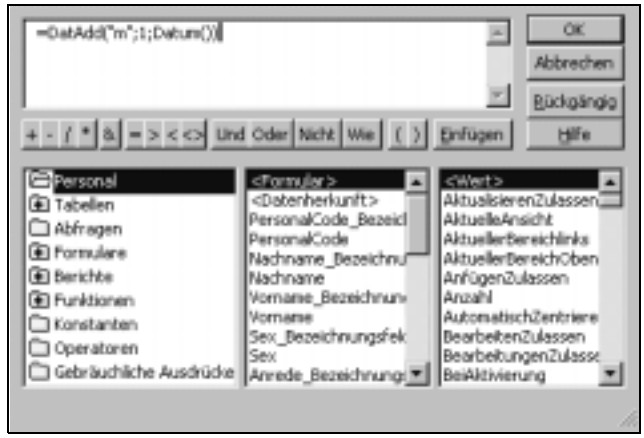


Bild 20.2: Dialogfeld »Ausdrucks-Generator«

Der Umgang mit dem Ausdrucks-Generator wird im einzelnen in Kap. 39, *Ausdrücke, Funktionen und Operatoren*, erklärt.

Der Ausdrucks-Generator kann vor allem bei komplizierter Syntax eine große Hilfe sein, weil er Elemente in einen Ausdruck syntaktisch richtig einfügt. Sein Vorteil zeigt sich in ganz besonderem Maße, wenn Sie sich in einem Ausdruck auf ein Steuerelement eines Unterformulars beziehen wollen – oder gar auf ein Steuerelement eines Unterformulars, das in ein Unterformular eingebettet ist; vgl. dazu den folgenden Punkt.

20.1.2 Syntax bei Steuerelementen aus demselben Formular, fremdem Formular sowie Unterformular

Sie können sich in einem berechneten Steuerelement auf jedes Access-Objekt beziehen, sofern dies sachlich sinnvoll ist. Dabei gilt die allgemeine Regel, daß der angeführte Bezeichner eindeutig sein muß. Dies bedeutet, daß Sie dem Namen eines Steuerelements, das nicht im selben Formular, sondern in einem anderen liegt, den Formularnamen und das Auflistungsobjekt *Formulare* voranstellen müssen. Wenn Sie sich beispielsweise auf das Steuerelement *Gesamtbetrag* im Formular *Ergebnisse* beziehen, müssen Sie schreiben:

Formulare![Ergebnisse]![Gesamtbetrag]

In einem Formular können Sie sich auch unmittelbar auf die Felder der Tabelle/Abfrage beziehen, die dem Formular als Datenherkunft zugrunde liegt. Wenn Sie beispielsweise in einem Textfeld 10% des Wertes vom Tabellenfeld *Honorar* ausgeben lassen wollen, schreiben Sie den Ausdruck

=[Honorar]*0,1

Falls allerdings – was zulässig und üblich ist – ein Steuerelement im selben Formular denselben Namen hat wie das Tabellenfeld, wertet der Ausdruck das Steuerelement und nicht das Tabellenfeld aus. Wenn Sie sich in Formularausdrücken auf Felder der dem Formular zugrunde liegenden Tabelle/Abfrage beziehen wollen, müssen Sie daher derartige Namenskonflikte vermeiden. Es gibt keine Möglichkeit, sich auf andere Weise direkt auf die Tabellen/Abfrage-Felder zu beziehen. (In gewisser Weise eine Ausnahme zu dieser Regel bilden die Domänenfunktionen wie z.B. *DomWert*, vgl. weiter unten Punkt 20.2.6, *Werte aus einer anderen Tabelle oder Abfrage ermitteln*.)

Dasselbe Formular

Wenn Sie sich in einem Ausdruck auf ein Steuerelement beziehen, das im selben Formular vorkommt wie das Steuerelement, in das Sie den Ausdruck eingeben, brauchen Sie nur den Namen des Steuerelements anzuführen. Wenn dieser Name Sonderzeichen oder Leerzeichen enthält, müssen Sie ihn zwischen eckigen Klammern angeben. Einen Namen ohne Sonder- oder Leerzeichen dürfen Sie ohne eckige Klammern eingeben. Access fügt diesem Namen dann allerdings die eckigen Klammern hinzu, wenn der Ausdruck angenommen wird.

Fremdes Formular

Um ein Steuerelement eines fremden Formulars in einem Ausdruck anzuführen, stellen Sie dem Steuerelementnamen den Namen des Formulars und das reservierte Wort *Formulare* voran, jeweils durch ein Ausrufezeichen getrennt.

Beispiel

Ermitteln Sie in einem Ausdruck eines Formulars das Zweifache des Wertes, den das Steuerelement *Gewicht bei Aufnahme* aus dem Formular *Patienten-Basiswerte* hat. Der Ausdruck soll in ein Steuerelement eines anderen Formulars als Einstellung der Eigenschaft *Steuerelementinhalt* eingegeben werden. Dann lautet der Ausdruck:

=[Formulare]![Patienten-Basiswerte]![Gewicht bei Aufnahme]*2

Unterformular

Aus der Sicht eines Hauptformulars ist ein Unterformular ein Steuerelement. Wenn Sie sich auf ein Steuerelement des Unterformulars beziehen wollen, heißt dies aus der Sicht eines Hauptformulars, daß Sie sich auf ein Steuerelement eines Steuerelements beziehen wollen. Dies funktioniert im allgemeinen nicht, ist allerdings speziell für ein Unterformular möglich, weil ein Unterformular die spezielle Eigenschaft *Formular* besitzt: Diese ergibt sozusagen das Formular, welches dem Steuerelement *Unterformular* als Herkunftsobjekt zugrunde liegt, und auf dessen Steuerelemente können Sie sich beziehen.

Beispiel

Sie wollen die Mehrwertsteuer (Satz: 16%) für den Betrag ermitteln, der vom Textfeld *Nettobetrag* im Unterformular *UFArtikel* ausgewiesen wird. Geben Sie dafür den folgenden Ausdruck an:

= UFArtikel.Formular!Nettobetrag*0,16

Beispiel

Ermitteln Sie für ein Textfeld des Hauptformulars *AngeboteBearbeiten* den Wert des Steuerelements *Gesamtpreis* im Unterformular *UFArtikelpositionen*, das selbst in das Unterformular *UFAngebote* eingebettet ist. Benutzen Sie dafür den Ausdrucks-Generator.

Gehen Sie folgendermaßen vor:

- Öffnen Sie das Formular *Angebote bearbeiten* in der Entwurfsansicht.
- Doppelklicken Sie auf das Textfeld, für das Sie den Ausdruck eingeben wollen, um es zu markieren und gleichzeitig ggf. das Eigenschaftenfenster anzuzeigen.
- Klicken Sie in das Einstellungsfeld der Eigenschaft *Steuerelementinhalt*.
- Klicken Sie auf die Symbol-Schaltfläche *Generator* (diese weist drei Punkte auf). Dann zeigt sich das Dialogfeld *Ausdrucks-Generator* so, wie es in Bild 20.2 wiedergegeben ist.

- Schreiben Sie das Gleichheitszeichen = in das Editierfeld im oberen Teil des Dialogfeldes.
- Doppelklicken Sie auf das Symbol *Angebote bearbeiten* im linken Teil des Dialogfeldes, um die untergeordneten Objekte dieses Formulars einzublenden.
- Doppelklicken Sie auf das Symbol des untergeordneten Objekts *UFAngebote*, um dessen untergeordnetes Objekt *UFArtikelpositionen* einzublenden.
- Klicken Sie auf das Symbol des untergeordneten Objekts *UFArtikelpositionen*, um dessen Steuerelemente in der mittleren Liste einzublenden.
- Markieren Sie das Steuerelement *Gesamtpreis*. Klicken Sie auf die Schaltfläche *Einfügen*. Dann wird der Ausdruck in das Editierfeld eingefügt. Diese Situation ist in Bild 20.3 wiedergegeben.
- Schließen Sie den Ausdrucks-Generator mit OK.

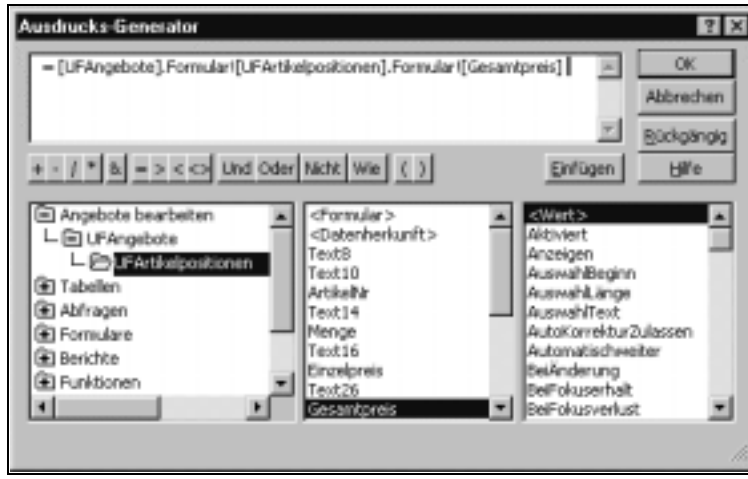


Bild 20.3: Mit dem Ausdrucks-Generator wurde hier ein Ausdruck erstellt, der sich auf das Steuerelement »Gesamtpreis« des Unterformulars »UFArtikelpositionen« bezieht, welches selbst wiederum in das Unterformular »UFAngebote« eingebettet ist. Das Hauptformular, für dessen Steuerelement der Ausdruck erstellt wurde, hat den Namen »Angebote bearbeiten«.

Der Ausdrucks-Generator sollte den folgenden Ausdruck erstellt haben:

```
= [UFAngebote].Formular![UFArtikelpositionen].Formular!
[Gesamtpreis]
```

20.1.3 Berechnete Werte in Tabelle speichern

Der Wert eines berechneten Steuerelements läßt sich nicht unmittelbar in der Tabelle speichern, weil der berechnende Ausdruck als Einstellung zur Eigenschaft *Steuerelementinhalt* angegeben ist und daher für dieses Steuerelement kein Feldname mehr angegeben werden kann. Wenn Sie den Wert eines berechneten Steuerelements in einem Feld der dem Formular zugrundeliegenden Tabelle speichern wollen, müssen Sie dies über einen Umweg mit Hilfe eines *SetzenWert*-Makros (oder einer entsprechenden VBA-Prozedur) bewerkstelligen.

Beispiel

Betrachten Sie Bild 20.4. Der Grundpreis und der Rabattsatz wird vom Benutzer frei eingetragen bzw. ausgewählt. Der resultierende Rabattbetrag soll von Access berechnet und im Tabellenfeld *Rabattbetrag* gespeichert werden.

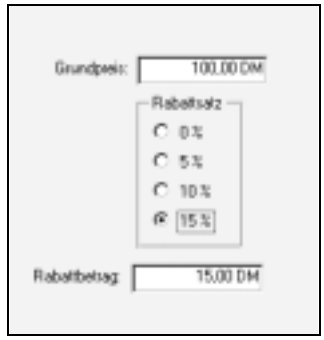


Bild 20.4: Steuerelemente in der Formularansicht. Der Rabattbetrag soll im Tabellenfeld »Rabattbetrag« gespeichert werden.

Lösen Sie diese Aufgabe wie folgt:

- Binden Sie die Textfelder *Grundpreis* und *Rabattbetrag* an die entsprechenden Tabellenfelder, vgl. Bild 20.5.
- Erstellen Sie ein Makro mit der Aktion *SetzenWert*. Geben Sie als Aktionsargument *Feld* den Feldnamen *Rabattbetrag* an, damit das Makro einen Wert in dieses Feld schreibt, wenn es ausgeführt wird. Geben Sie für das Makroargument *Ausdruck* den folgenden Ausdruck an:

  ```
  =[Rabattsatz]/100*[Grundpreis]
  ```

  Dieser Ausdruck setzt voraus, daß die Optionswerte der vier Schaltflächen in der Optionsgruppe *Rabattsatz* auf die Zahlen 0, 5, 10 und 15 eingestellt sind. Geben Sie dem Makro einen Namen, z.B. *RabattErmitteln*, vgl. Bild 20.6.

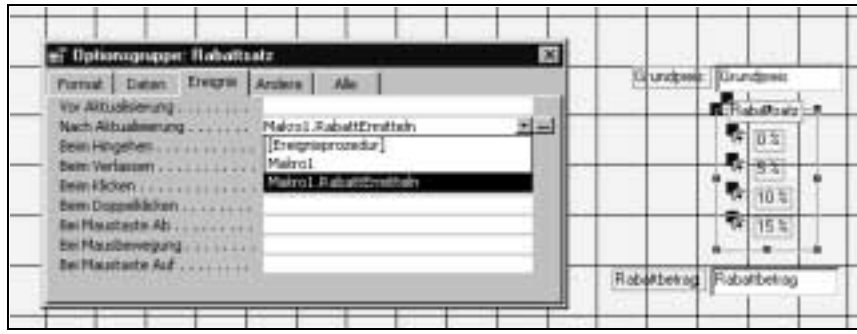


Bild 20.5: Steuerelemente in der Entwurfsansicht. Der Rabattbetrag wird im Tabellenfeld »Rabattbetrag« gespeichert. Die Ereigniseigenschaft »Nach Aktualisierung« der Optionsgruppe »Rabattsatz« ist auf das Ereignismakro »Makro1.RabattErmitteln« (vgl. Bild 20.6) eingestellt.

- Stellen Sie die Ereigniseigenschaft *Nach Aktualisierung* der Optionsgruppe *Rabattsatz* auf den Namen des Ereignismakros *Makro1.RabattErmitteln* ein, vgl. Bild 20.5.

Bild 20.6: Das Makro »RabattErmitteln« setzt den Wert des Feldes des aktiven Objekts auf den Wert, den der Ausdruck ergibt.

20.2 Beispiele

20.2.1 Mathematische Berechnungen

Mit Ausdrücken in Steuerelementen können Sie mathematische Berechnungen jeder Art ausführen. Im allgemeinen kommen für diesen Zweck Textfelder in Frage. Der folgende Ausdruck beispielsweise ermittelt Nebenkosten

inhaltlich als Summe von *Vermarkungs- und Zeichenmaterial*, *Reisen und Aufwand* sowie einem auf das *Honorar* angewandten *Aufwand-Prozentsatz*, was formal der Addition bzw. Multiplikation der entsprechenden Steuerelementnamen im Formular entspricht. Bild 20.7 zeigt, wie Sie einen derartigen Ausdruck mit Hilfe des Dialogfeldes *Zoom* (Sie öffnen dieses mit Umschalt+F2) als Eigenschaft *Steuerelementinhalt* einstellen oder bearbeiten können.

=[VermarkZeichenMat]+[ReisenUndAufwand]+[AufwandProzentsatz]*[Honorar]/100

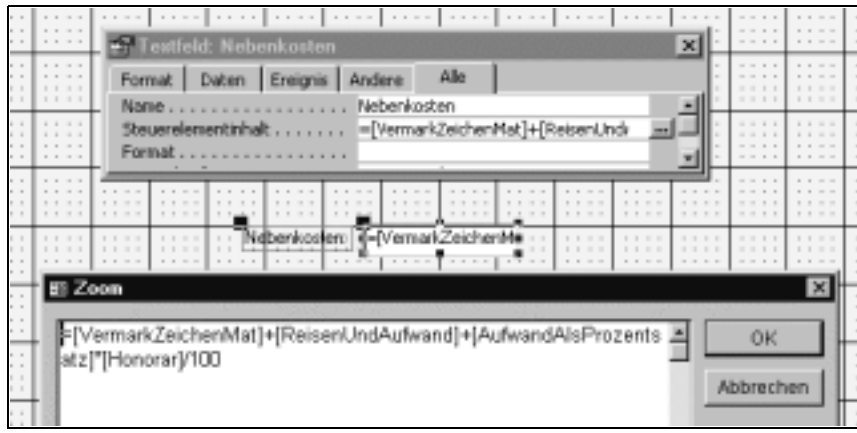


Bild 20.7: Das Dialogfeld »Zoom« ermöglicht eine bequemere Bearbeitung längerer Ausdrücke: Drücken Sie die Tastenkombination Umschalt+F2, während die Eigenschaft »Steuerelementinhalt« den Fokus hat, um das Dialogfeld »Zoom« zu öffnen.

Ausdrücke in Steuerelementen dürfen außer den integrierten Access-Funktionen auch benutzerdefinierte Funktionen, die in VBA erstellt werden können, enthalten. Die folgende VBA-Funktion beispielsweise ermittelt den Mehrwertsteuersatz aus dem Feld *MwStSatz* der Tabelle *Konstanten*:

```
Function MwStSatz ()
'Gibt den in der Tabelle Konstanten festgelegten
'Mehrwertsteuersatz aus
Dim DB As Database, TT1 As Table
Set DB = CurrentDB()
Set TT1 = DB.OpenRecordset("Konstanten")
MwStSatz = TT1!MwStSatz
End Function
```

Sie können diese Funktion in einem Ausdruck statt einer Konstanten (aktuell: 0,16) angeben. Diese Arbeitsweise hat den großen Vorteil, daß Sie bei einer Änderung des gesetzlichen Mehrwertsteuersatzes diesen nur an einer

Stelle, nämlich in der Tabelle *Konstanten*, zu ändern brauchen statt in allen möglichen Ausdrücken aller möglichen Objekte. Ein derartiger Ausdruck könnte beispielsweise heißen:

= Format([Nettobetrag]*MwStSatz();"0,00")

Dieser Ausdruck ermittelt den Mehrwertsteuerbetrag als Multiplikation des Steuerelements oder Feldes *Nettobetrag* und der benutzerdefinierten Funktion *MwStSatz* und gibt diesen auf zwei Dezimalstellen gerundet aus.

20.2.2 Datumswerte ermitteln und anzeigen

Sehr praxisrelevant ist der Umgang mit Datumswerten. Access stellt dafür mehrere Funktionen zur Verfügung, die Sie alle in berechneten Steuerelementen von Formularen verwenden können. (Zu Datums- und Zeitfunktionen vgl. im einzelnen Kap. 39, *Ausdrücke, Funktionen und Operatoren*, wo Sie verschiedene weitere Beispiele finden; vgl. auch die Online-Hilfe.)

Beispiel

Das Tagesdatum geben Sie auf die folgende Weise in einem Textfeld aus:

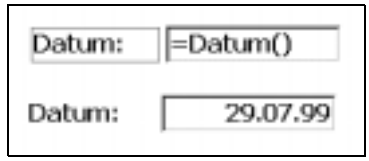


Bild 20.8: Ermittlung des aktuellen Datums in einem Textfeld, oben in der Entwurfsansicht, unten in der Formularansicht.

Beispiel

In der kaufmännischen Praxis wird für die Zahlungsbedingungen vielfach die folgende Alternative vorgegeben: Wenn mit Skonto gezahlt wird, muß die Zahlung sofort erfolgen; bei Zahlung ohne Skonto liegt das Zahlungsziel (beispielsweise) vier Wochen nach dem aktuellen Rechnungsdatum.

In diesem Beispiel wird mit dem Kontrollkästchen *MitSkonto* angegeben, ob Skonto in Anspruch genommen wird. In Abhängigkeit davon zeigt das Textfeld *Zahlungsziel* den Text *Zahlung sofort* oder das Zahlungsziel an, welches vier Wochen hinter dem aktuellen Datum liegt. Entscheidend ist der Ausdruck

=Wenn([MitSkonto];"Zahlung sofort";DatAdd("ww";4;Datum()))

im Textfeld *Zahlungsziel*, der u.a. mit der Funktion *DatAdd* – eingebettet in eine Wenn-Funktion – das vier Wochen nach dem aktuellen Datum liegende Datum ermittelt, vgl. Bild 20.9 für die Entwurfsansicht und Bild 20.10 für das Ergebnis.

Bild 20.9: Steuerelemente »MitSkonto« und »Zahlungsziel« in der Entwurfsansicht

Bild 20.10: Steuerelemente »Datum«, »MitSkonto« und »Zahlungsziel« in der Formularansicht: Oben mit nicht gewähltem Skonto, unten mit gewähltem Skonto.

20.2.3 Texte verketten

Mit dem Verkettungsoperator & lassen sich Texte und Zahlenwerte zu einer neuen Zeichenkette zusammensetzen. Angenommen, Vor- und Nachname einer Person seien in getrennten Feldern abgelegt. Dann können Sie beide als Gesamtnamen mit dem folgenden Ausdruck ausgeben lassen:

=[Vorname] & Wenn(IstNull([Vorname]);"";" ") & [Nachname]

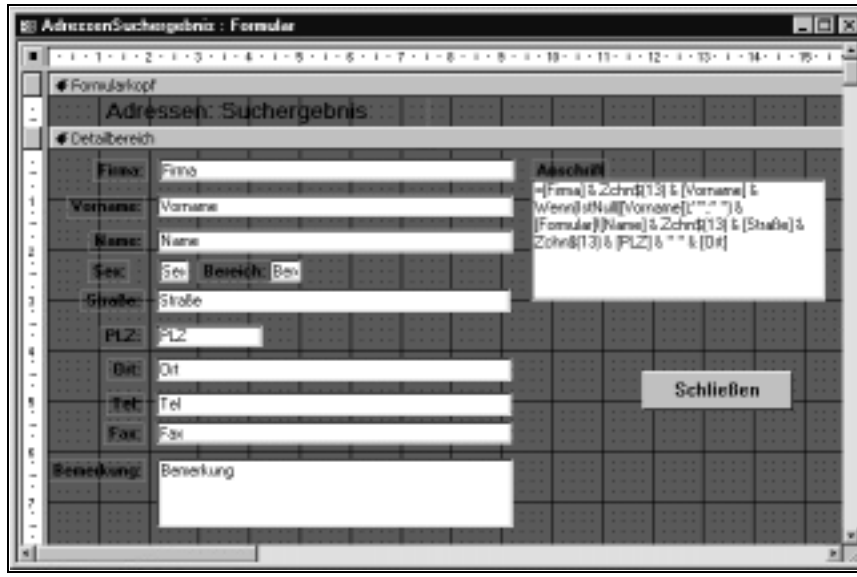


Bild 20.11: Entwurfsansicht des Formulars mit einem Textfeld, das die Briefanschrift ermittelt

Beispiel

Wenn Sie einen Brief schreiben wollen und die Anschrift einem Access-Formular entnehmen wollen, können Sie diese aus den Elementen der Adresse zusammen mit den für das Anschriftenfeld im Brief erforderlichen Zeilenumbrüchen in einem verkettenden Ausdruck generieren. Das Textfeld *Anschrift* im Formular von Bild 20.11 und Bild 20.12 enthält den folgenden Ausdruck:

=[Firma] & Zchn$(13) & [Vorname] & Wenn(IstNull([Vorname]);"";" ") & [Formular]![Name] & Zchn$(13) & [Straße] & Zchn$(13) & [PLZ] & " " & [Ort]

Dieser Ausdruck ergibt die umbrochene Anschrift, die Sie in Bild 20.12 erkennen können. Dabei werden die Zeilenumbrüche mit dem ASCII-Zeichen 13 erzeugt. Sie können den Inhalt des Textfeldes beispielsweise über die Zwischenablage in das Anschriftenfeld eines Briefes kopieren.

Das in Bild 20.12 dargestellte Formular gibt das Ergebnis einer Adressensuche wieder. Es ist Teil eines komplexeren Beispiels für eine Abfrage, die ihre Kriterien einem Formular entnimmt. Das Beispiel wird ausführlich erklärt in Kap. 12, *Komplexe Abfragen*, Punkt 12.2.2, *Beispiel einer kompletten Formularsteuerung der Suchvorgänge (Query-by-Form)*.

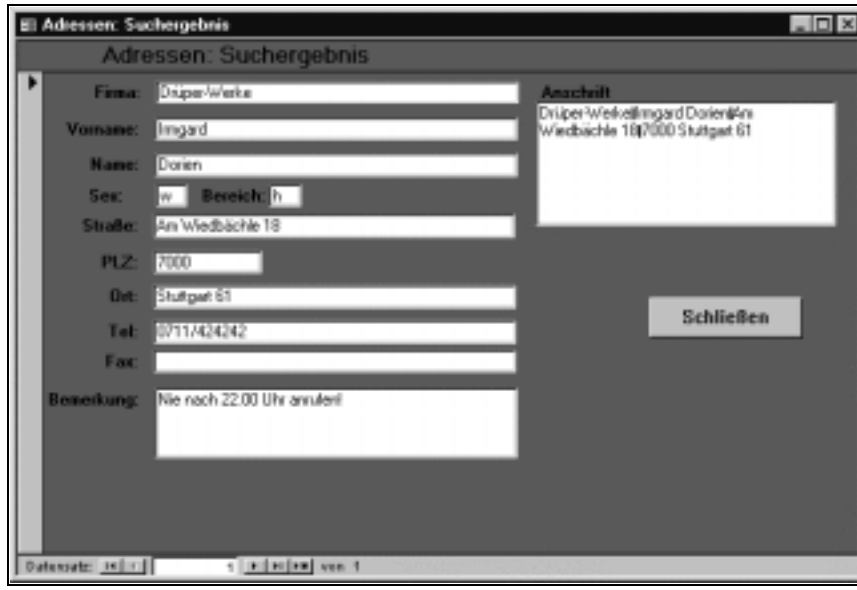


Bild 20.12: Formularansicht des Formulars mit einem Textfeld, das die Briefanschrift ermittelt.

20.2.4 Seitenzahl beim Drucken ausgeben

Für den Fall, daß Sie ein längeres Formular, das sich über mehrere Druckseiten erstreckt, drucken lassen, kann es sinnvoll sein, die Seitenzahl mit ausdrucken zu lassen. Dafür steht die Formulareigenschaft *Seite* zur Verfügung. Sie beziehen sich auf diese Eigenschaft praktisch wie auf eine integrierte Funktion, wenngleich in diesem Fall kein Klammerpaar angegeben wird. Die Eigenschaft *Seite* wird nur beim Drucken und in der Seitenansicht des Formulars ausgewertet, nicht jedoch in der Formularansicht. Aus diesem Grunde sollten Sie einen Ausdruck, der *Seite* verwendet, nur im Seitenkopf oder -fuß anführen: Ein Seitenkopf oder -fuß wird in der Formularansicht nicht dargestellt, sondern nur beim Drucken und in der Seitenansicht ausgegeben. Wenn Sie *Seite* in einem Ausdruck anführen, der im Detailbereich verwendet wird, führt dies zwar beim Drucken und in der Seitenansicht zu korrekten Ergebnissen, jedoch wird in der Formularansicht ein Fehlerwert ausgegeben.

Neben der Eigenschaft *Seite* verfügen Sie auch über die Eigenschaft *Seiten*, welche die Gesamtzahl der Seiten eines Formulars (oder Berichts) angibt. Sie können auch beide Eigenschaften in einem Ausdruck durch Verkettung zusammenführen. Sinnvoll könnte beispielsweise der folgende Ausdruck sein, den Sie in einem Textfeld, das z.B. im Seitenfuß positioniert ist, anführen könnten:

```
=="Seite " & Seite & " von " & Seiten
```

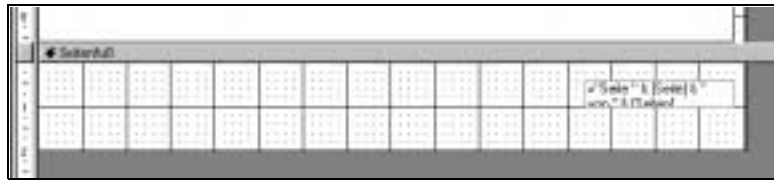


Bild 20.13: Ein Textfeld im Seitenfuß enthält den Ausdruck: ="Seite » & Seite & » von » & Seiten.

Um einem Formular ein Textfeld mit einem Ausdruck, wie gerade angeführt, hinzuzufügen, verfahren Sie wie folgt:

- Blenden Sie ggf. den Seitenkopf/-fuß ein, indem Sie den gleichnamigen Befehl aus dem Menü *Ansicht* wählen.
- Fügen Sie in den Seitenkopf oder -fuß ein Textfeld ein. Ziehen Sie den ggf. nicht benötigten Seitenkopf oder -fuß auf die Länge 0.
- Geben Sie für den Steuerelementinhalt des eingefügten Textfeldes den o.a. Ausdruck ein (vgl. Bild 20.13). Wenn Sie die Wirkung dieses Ausdrucks vor dem Drucken überprüfen wollen, wechseln Sie in die Seitenansicht (vgl. Bild 20.14).

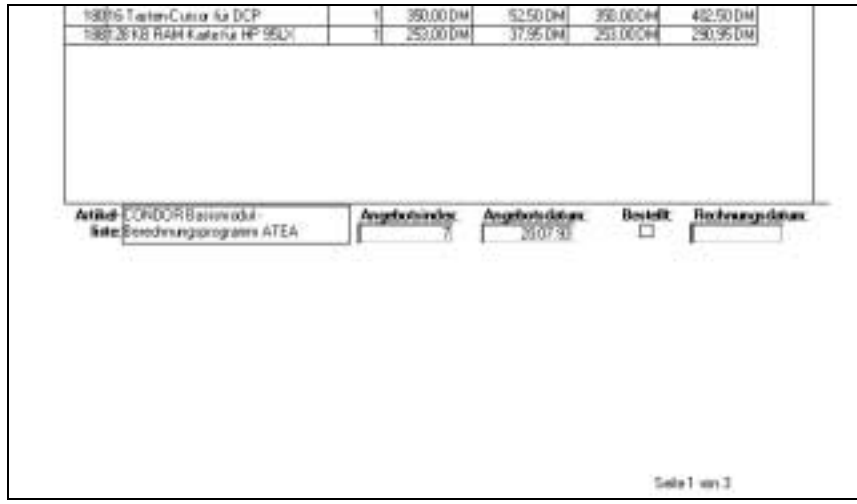


Bild 20.14: In der Seitenansicht (hier) sowie beim Drucken wird der Ausdruck, der im Seitenfuß im Textfeld angegeben ist, ausgewertet und wiedergegeben.

20.2.5 Summe für eine Gruppe von Datensätzen im Haupt- oder Unterformular

Sie können die Summe aller Datensätze eines Feldes ermitteln. Dazu steht die Funktion *Summe* zur Verfügung. Die Syntax dieser Funktion lautet:

```
Summe(Ausdruck)
```

Dabei darf (und muß) das Argument *Ausdruck* Feldnamen der Tabelle/Abfrage enthalten, jedoch keinen Steuerelementnamen, denn diese Funktion kann sich nur direkt auf Felder der Datenbasis beziehen, die dem Formular zugrunde liegt, in dem sich das Textfeld mit dem Ausdruck befindet. (Zur Bezugnahme auf Felder anderer Tabellen vgl. den folgenden Punkt 20.2.6, *Werte aus einer anderen Tabelle oder Abfrage ermitteln*.)

Summe aus demselben Formular ermitteln

Bild 20.15: Das Feld »Auftrags-Gesamtwert« gibt die Summe aller Auftragswerte wieder.

Beispiel. Dem in Bild 20.15 wiedergegebenen Formular *Teilauftrag Bearbeiten* liegt eine Tabelle zugrunde, die u.a. das Feld *Auftragswert* enthält. Der jeweilige Inhalt dieses Feldes wird im gleichnamigen Formularfeld wiedergegeben. Für den in Bild 20.15 aktuellen Datensatz beträgt er 4.000,00 DM. Der für dieses Feld für die gesamte Tabelle aufsummierte Gesamtwert wird im Feld *Auftrags-Gesamtwert* ausgewiesen. Er beträgt in Bild 20.15 146.450,00 DM. Das berechnete Textfeld, das diesen Betrag ermittelt, ist in Bild 20.16 in der Entwurfsansicht wiedergegeben: Es enthält den Ausdruck

```
=Summe([Auftragswert])
```

Bild 20.16: Die Funktion »Summe« bezieht sich mit »Auftragswert« auf ein Feld der Tabelle, die dem Formular zugrunde liegt.

Summe aus Unterformular ermitteln

Wenn ein Hauptformular ein Unterformular enthält, weist das letztere sehr oft in der Datenblattansicht mehrere Datensätze aus. Dabei erscheint es häufig angezeigt, die Summe für ein Feld dieser Datensätze im Hauptformular auszuweisen. Diese Aufgabe kann jedoch nicht in derselben einfachen Weise gelöst werden, wie dies im vorangehenden Punkt für ein und dasselbe

Formular gezeigt wurde. Eine Komplikation ergibt sich, weil Sie sich in der Funktion *Summe* nicht auf ein Steuerelement beziehen dürfen. Ein Unterformular jedoch ist aus der Sicht eines Hauptformulars ein Steuerelement, so daß Sie sich auf dessen Textfelder nur beziehen können, wenn Sie gleichzeitig den Steuerelementnamen des Unterformulars anführen. Als Ausdruck für die Funktion *Summe* dürfen Sie daher beispielsweise den folgenden Ausdruck *nicht* angeben:

```
=Summe(UFArtikelpositionen.Formular!Preis)
```

Sie müssen die Aufgabe deshalb über einen Umweg lösen: Sie ermitteln die Summe zunächst in einem Textfeld des Unterformulars und beziehen sich dann in einem Textfeld des Hauptformulars auf das summierende Textfeld des Unterformulars. Damit letzteres nicht angezeigt wird, positionieren Sie es im Formularfuß des Unterformulars, weil dieser in der Datenblattansicht nicht wiedergegeben wird. Alternativ stellen Sie die Eigenschaft *Sichtbar* des Textfeldes auf *Nein* ein.

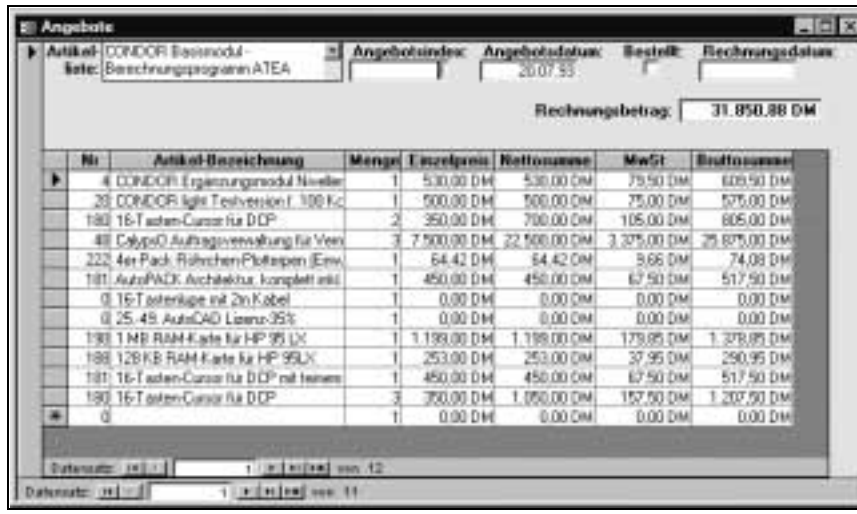


Bild 20.17: Formular »Angebote« in der Formularansicht: Das mit »Rechnungsbetrag« bezeichnete Textfeld gibt die Summe des Feldes »Bruttosumme« für alle Datensätze des Unterformulars wieder.

Beispiel

Bild 20.17 zeigt das Formular *Angebote*. Es enthält ein Unterformular in der Datenblattansicht. Die Spaltenüberschrift *Bruttosumme* bezeichnet ein Feld, das in der Tabelle den Namen *Gesamtpreis* hat. Die Summe dieses Gesamtpreises für alle Artikelpositionen des Unterformulars wird im Hauptformular im Feld mit der Bezeichnung *Rechnungsbetrag:* ausgewiesen. In Bild 20.17 beträgt diese Summe 31.850,88 DM.

Bild 20.18: Formular »Angebote« in der Entwurfsansicht: Das mit »Rechnungsbetrag« bezeichnete Textfeld bezieht sich auf das Steuerelement »Rechnungsbetrag« im Unterformular »UFArtikelpositionen«. Das Textfeld wurde für diese Darstellung in der Länge vergrößert, um seinen Ausdruck ganz darzustellen.

Bild 20.18 zeigt, daß das Textfeld mit der Bezeichnung *Rechnungsbetrag* sich direkt auf das Steuerelement *Rechnungsbetrag* im Unterformular *UFArtikelpositionen* bezieht. Dieses wiederum – vgl. Bild 20.19 – ermittelt die Summe mit dem Ausdruck

```
=Summe([Gesamtpreis])
```

Da dieses Textfeld im Formularfuß des Formulars *UFArtikelpositionen* positioniert ist, wird es im Hauptformular nicht angezeigt, denn das Unterformular befindet sich dort in der Datenblattansicht.

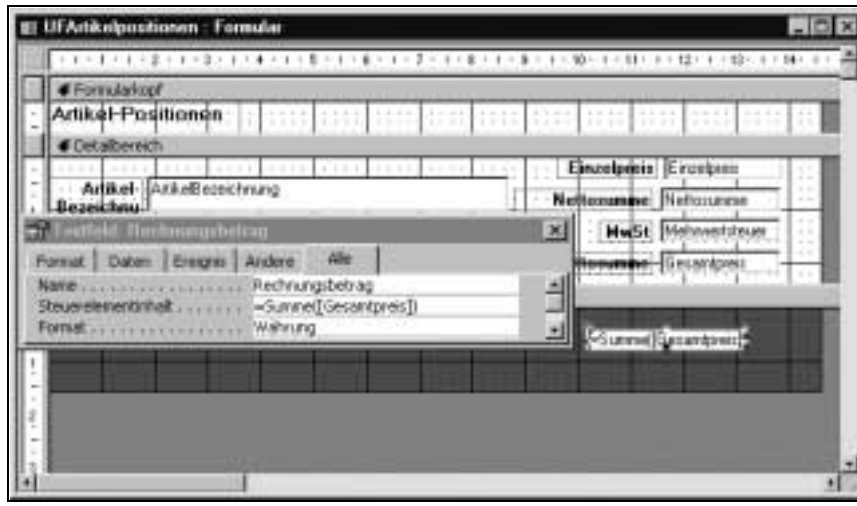


Bild 20.19: Formular »UFArtikelpositionen« in der Entwurfsansicht: Das Textfeld »Rechnungsbetrag«, das die Summe für das Tabellenfeld »Gesamtpreis« ermittelt, ist im Formularfuß positioniert.

20.2.6 Werte aus einer anderen Tabelle oder Abfrage ermitteln

Auf Steuerelemente anderer Formulare können Sie sich direkt beziehen, vgl. oben Punkt 20.1.2, *Syntax bei Steuerelementen aus demselben Formular, fremdem Formular sowie Unterformular.* Auf ein Feld eines bestimmten Datensatzes einer Tabelle/Abfrage können Sie sich dagegen von einem Formular aus nicht direkt beziehen. Mit Hilfe der Domänenfunktionen (z.B. *DomSumme*, *DomAnzahl*, *DomWert* etc.) ist dies jedoch, wenngleich über einen Umweg, ebenfalls möglich. Die Domänenfunktionen dienen in erster Linie dazu, auf Basis eines Kriterienausdrucks aggregierte Werte aus einem Tabellenfeld zu ermitteln. Die allgemeine Syntax der *Dom*-Funktionen – hier am Beispiel von *DomSumme* dargestellt – lautet:

```
DomSumme(Ausdruck;Domäne;[,Kriterien])
```

Das letzte Argument *Kriterien* ist wahlfrei. Wenn Sie es fortlassen, wird die Aggregation für alle Datensätze der Tabelle ausgeführt. Das Argument *Ausdruck* gibt als Zeichenkette – also in Anführungszeichen – einen Ausdruck an, der ein Feld der Tabelle/Abfrage enthält. Das Argument *Domäne* gibt den Namen der Tabelle/Abfrage an, ebenfalls als Zeichenkette. Wenn Sie mit der Funktion *DomWert* den Feldwert eines bestimmten Datensatzes einer Tabelle/Abfrage ermitteln wollen, müssen Sie das Kriterium kennen und als drittes Argument anführen, das diesen einzelnen Datensatz eindeutig identifiziert. Vgl. genauer zu den Domänenfunktionen Kap. 39, *Ausdrücke, Funktionen und Operatoren*, sowie die Online-Hilfe.

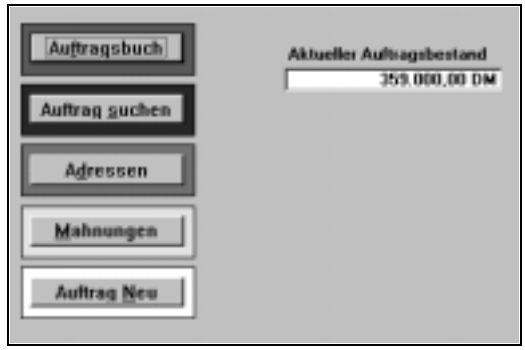


Bild 20.20: Formular mit Befehlsschaltflächen und einem Textfeld, welches mit Hilfe der Funktion »DomSumme« den Auftragsbestand aus der Tabelle »TAF« ermittelt.

Beispiel. Bild 20.20 gibt ein Formular wieder, für das selbst gar keine Datenherkunft angegeben ist, weil es als reines Schaltflächen-Formular zur Steuerung einer Anwendung dient. Jedoch soll in diesem Hauptformular der Anwendung der aktuelle Bestand aller noch nicht abgeschlossenen Aufträge ausgewiesen werden. Dies geschieht im Textfeld mit der Bezeichnung *Aktueller Auftragsbestand*.

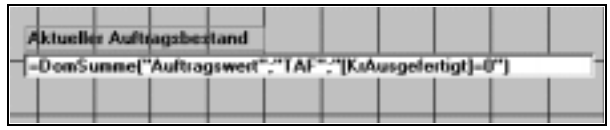


Bild 20.21: Textfeld aus Bild 20.20 in der Entwurfsansicht: Es wurde in der Länge vergrößert, um seinen gesamten Steuerelementinhalt darstellen zu können.

Bild 20.21 zeigt dieses Textfeld mit seinem Steuerelementinhalt in der Entwurfsansicht des Formulars. Im Ausdruck

=DomSumme(»Auftragswert«;"TAF«;"[KrAusgefertigt]=0«)

gibt das Argument *Auftragswert* das Feld an, dessen Inhalt aggregiert – hier: summiert – werden soll. Das Argument *TAF* identifiziert die Tabelle, in der das Aggregationsfeld liegt. Der Kriterienausdruck *[KrAusgefertigt]=0* gibt an, daß nur Datensätze berücksichtigt werden sollen, für die das Feld *KrAusgefertigt* den Wert 0 hat: *KrAusgefertigt* steht für *Kostenrechnung ausgefertigt* und ist ein *Ja/Nein*-Feld, für das der Wert 0 dem Wahrheitswert *Falsch* entspricht.

Die Domänenfunktionen akzeptieren nicht den Wahrheitswert *Falsch*, sondern nur die beiden Werte 0 und *False*, entsprechend -1 und *True*. Hier hat das Eindeutschen offenbar immer noch nicht (es handelt sich um die fünfte Access-Version!) geklappt, obgleich die Semikola akzeptiert werden!

Kapitel 21

Berichte: Grundlagen

21.1 Einführung und Überblick

☞ Die Bearbeitung von Berichten deckt sich weitgehend mit derjenigen von Formularen. Um überflüssige Wiederholungen zu vermeiden, konzentrieren sich die Ausführungen der Kap. 21 bis 23, die sich mit Berichten befassen, auf die Besonderheiten, die Sie beim Umgang mit Berichten kennen und berücksichtigen sollten. Die Kenntnis der Kap. 15 und 16, die grundlegend für Formulare sind, wird daher für das vorliegende Kapitel vorausgesetzt. Dies gilt in ganz besonderem Maße für das Erstellen und Bearbeiten der Steuerelemente, die in Kap. 16 ausführlich behandelt sind.

21.1.1 Was ist ein Bericht?

Berichte sind in vielerlei Hinsicht Formularen vergleichbar. Der bedeutsamste Unterschied liegt darin, daß Berichte zum Drucken bestimmt sind. Sie können zwar auch auf dem Bildschirm betrachtet werden, jedoch nur in der Seitenansicht, die dazu dient, das Druckbild zu kontrollieren. Eine der Formularansicht von Formularen vergleichbare Berichtsansicht gibt es nicht. Daraus ergibt sich auch, daß Sie in einen Bericht keine Daten eingeben können. Gleichwohl stehen die Steuerelemente, die für Formulare verwendbar sind, auch für Berichte zur Verfügung. Jedoch dienen sie hier nur zur Wiedergabe von Daten, nicht zu deren Bearbeitung.

21.1.2 Besondere Leistungsmerkmale von Berichten

Obwohl Sie auch Formulare ausdrucken können und damit in vielen Situationen gute Ergebnisse erzielen, eignen sich Berichte für die Druckausgabe oft besser als Formulare. Dies ist vor allem in den folgenden drei Leistungsmerkmalen begründet, die nur Berichte aufweisen:

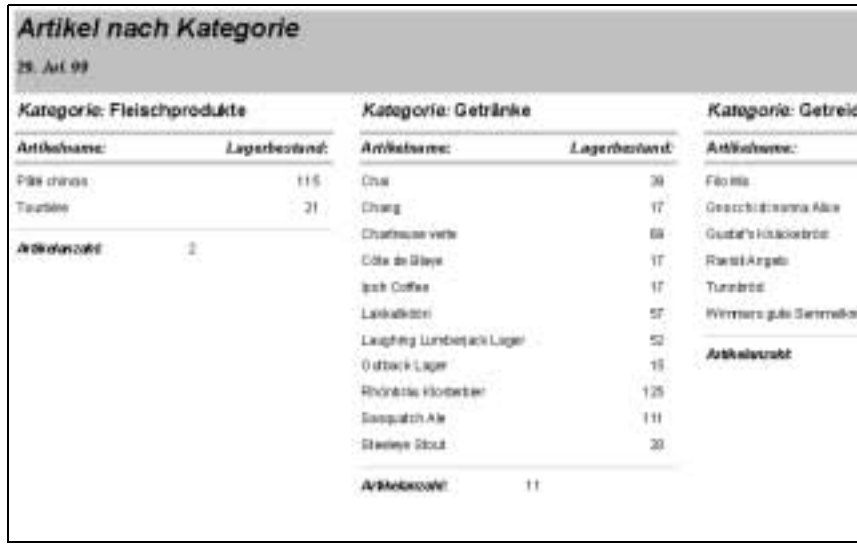


Bild 21.1: Gruppierungsbericht »Artikel nach Kategorie« aus der Datenbank »Nordwind.mdb«, Ausschnitt. Der Bericht wird mit drei Spalten im Querformat ausgegeben.

- *Gruppieren*. Sie können Datensätze zu Gruppen zusammenfassen und entsprechend tabellarisch anordnen. Für jede Gruppe lassen sich zusammenfassende Berechnungen ausgeben, beispielsweise Summen oder Anzahlen. Der in Bild 21.1 wiedergegebene Bericht *Artikel nach Kategorie* aus der Datenbank Nordwind.mdb beispielsweise listet den Artikelnamen und Lagerbestand für alle Artikel derselben Kategorie jeweils als Gruppe auf und weist unterhalb jeder Gruppe die Artikelanzahl aus. Dieser Bericht ist nach dem Feld *Kategoriename* gruppiert. Access-Berichte ermöglichen es auch, innerhalb einer Gruppe eine weitere Gruppierung zu bilden. Beispielsweise könnten Sie einen Bericht mit Angaben über Bestellungen nach den zugehörigen Mitarbeitern und innerhalb jedes Mitarbeiters nach Artikelkategorien gruppieren lassen. Insgesamt dürfen Sie bis zu zehn Gruppen verschachteln.
- *Sortieren*. Wenn Sie die Datensätze einer Tabelle in einem Formular oder Datenblatt in besonderer Weise (d.h. anders als nach dem Primärschlüssel) sortiert ausgeben wollen, müssen Sie zuvor eine Abfrage bilden, welche die Sortierung leistet, weil ein Formular oder Datenblatt von sich aus nicht sortieren kann. Dies ist jedoch in Berichten möglich: Sie können Datensätze nach mehreren Feldern sortiert (z.B. nach dem Namen, Vornamen und Ort) ausgeben lassen, ohne die Datenbasis vorher entsprechend umgeformt zu haben. Daher können auch Tabellendaten unmittelbar von einem Bericht in sortierter Form ausgegeben werden.
- *Mehrere Spalten*. Moderne Textverarbeitungsprogramme ermöglichen die Textwiedergabe in mehreren Spalten, wobei zwischen den Spalten ein automatischer Umbruch erfolgt. Diese Leistung können Sie auch von

einem Bericht fordern. Die Darstellung in mehreren Spalten eignet sich besonders gut für den Druck von Adreßetiketten. Der in Bild 21.1 wiedergegebene Bericht wird ebenfalls in drei Spalten ausgegeben.

21.2 Berichte und Formulare: Gemeinsamkeiten und Unterschiede

Gemeinsamkeiten

- Berichte werden wie Formulare in der Entwurfsansicht bearbeitet.
- In der Entwurfsansicht stehen dieselben Werkzeuge, z.B. Eigenschaftenfenster, Toolbox, Feldliste und Formatierungssymbole, zur Verfügung.
- Die Art und Weise, wie Sie einzelne Steuerelemente bearbeiten (z.B. Markieren, Einstellen von Eigenschaften, Verschieben, Verändern der Größe etc.), unterscheidet sich in Formularen und Berichten nicht.
- Auch ein Objekt *Bericht* wird insgesamt so verwaltet, wie ein Objekt *Formular*: Beide können Sie speichern, kopieren, importieren, löschen, umbenennen etc.
- Sie erstellen einen neuen Bericht auf die prinzipiell gleiche Weise wie ein Formular: Sie fordern einen neuen Bericht an (z.B. mit der Schaltfläche *Neu* im Datenbankfenster, während die Objektliste *Berichte* aktiv ist). Dann bedienen Sie sich entweder des Berichts-Assistenten, oder Sie fordern einen leeren Bericht an, den Sie dann in der Entwurfsansicht mit Steuerelementen, Bereichen, Eigenschaften etc. versehen.

Unterschiede

Die wesentlichen Unterschiede zwischen Formularen und Berichten ergeben sich aus ihren unterschiedlichen Leistungsmerkmalen:

- Da Berichte für die Druckausgabe von Daten bestimmt sind und nicht für deren Darstellung auf dem Bildschirm, ist keine der Formularansicht vergleichbare Berichtsansicht verfügbar.
- Für Berichte sind, ganz analog zu Formularen, die Bereiche *Berichtskopf/-fuß*, *Seitenkopf/-fuß* und *Detailbereich* verfügbar. Darüber hinaus gibt es in Berichten einen *Gruppenkopf* und *-fuß* für jede definierte Gruppierung.
- Der Besonderheit des Gruppierens und Sortierens entspricht ein besonderes Werkzeug in der Entwurfsansicht eines Berichts: Mit dem Symbol *Sortieren und Gruppieren* (vgl. links nebenstehend) öffnen bzw. schließen Sie das Fenster *Sortieren und Gruppieren*, vgl. Bild 21.2.

Bild 21.2: Dialogfeld »Sortieren und Gruppieren«

21.3 Neuen Bericht erstellen

Wenn Sie einen neuen Bericht erstellen wollen, leiten Sie dies mit einer der beiden folgenden Aktionen ein:

- Wechseln Sie zum Datenbankfenster, aktivieren Sie die Objektliste *Berichte*, und klicken Sie dann auf die Schaltfläche *Neu*.

  Oder:

- Schlagen Sie die Dropdown-Liste der Symbol-Schaltfläche *Neues Objekt* auf, und wählen Sie darin *Bericht*.

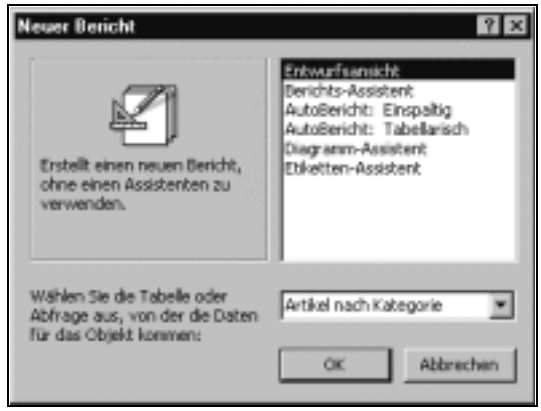


Bild 21.3: Dialogfeld »Neuer Bericht«

Als Folge wird in beiden Fällen das Dialogfeld *Neuer Bericht* eingeblendet, in dem Sie u.a. entscheiden müssen, ob Sie den neuen Bericht als leeren oder mit Hilfe des Berichts-Assistenten erstellen wollen. Sie können aber auch zwei Varianten eines *AutoBerichts*, einen Bericht mit einem Diagramm sowie einen Bericht, der Etiketten beschreibt, erstellen lassen. Im folgenden zeige ich exemplarisch, wie Sie einen Bericht mit Hilfe des Berichts-Assisten-

ten erstellen und diesen anschließend bearbeiten. Sie sollten jedoch – und sei es nur experimentell – auch einmal die verschiedenen anderen Wege, einen Bericht zu erstellen, erkunden. Insbesondere ein mit dem Etiketten-Assistenten erstellter Bericht kann sehr nützlich sein, weil er viele Standard-Formate handelsüblicher Etiketten anbietet und zuverlässig beschreibt, so daß Ihnen das in derartigen Fällen oft mühsame Verfahren von Versuch und Irrtum erspart bleibt. Bild 21.4 gibt einen kleinen Einblick in die Vielfalt der angebotenen Etikettentypen, aus denen Sie wählen können.

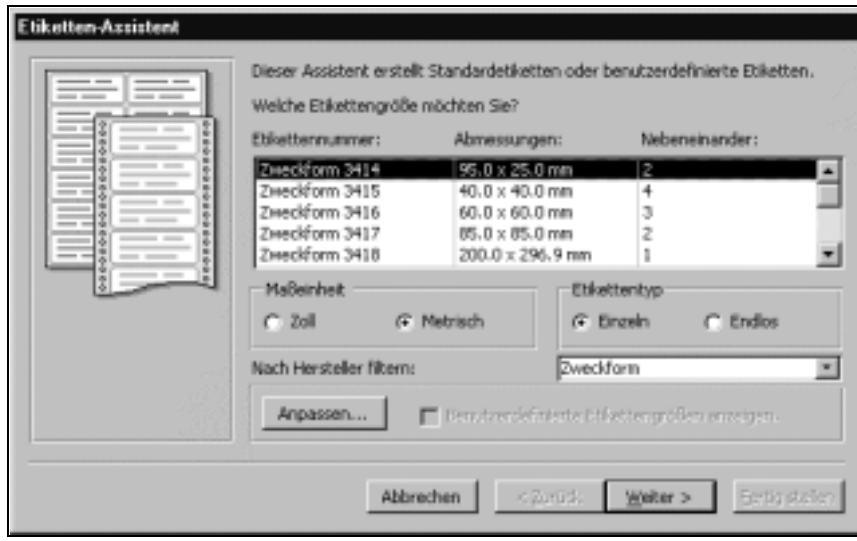


Bild 21.4: Dialogfeld des Etiketten-Assistenten zur Auswahl eines Etikettentyps

21.3.1 Bericht mit dem Berichts-Assistenten erstellen

Im folgenden wird am Beispiel eines gruppierten Berichts gezeigt, wie Sie einen Bericht mit dem Berichts-Assistenten erstellen.

Aufgabe

Erstellen Sie mit dem Berichts-Assistenten einen gruppierten Bericht. Ausgangspunkt soll die Datenbank *Projekte.mdb* von der Begleit-CD-ROM sein. Sie enthält die beiden Tabellen *Personal* einerseits und *Projekte* andererseits. Die beiden Tabellen weisen eine 1:n-Beziehung auf. Der Bericht soll den Namen und das Portrait jedes Mitarbeiters wiedergeben. Neben diesen Angaben soll für alle vom jeweiligen Mitarbeiter betreuten Projekte die Projektbezeichnung sowie der Auftragswert als Gruppe wiedergegeben werden. Am Ende jeder Gruppe soll die Auftragssumme der Gruppe ausgewiesen werden. Der fertige Bericht soll in der Seitenansicht so aussehen, wie in Bild 21.5 und Bild 21.6 wiedergegeben.

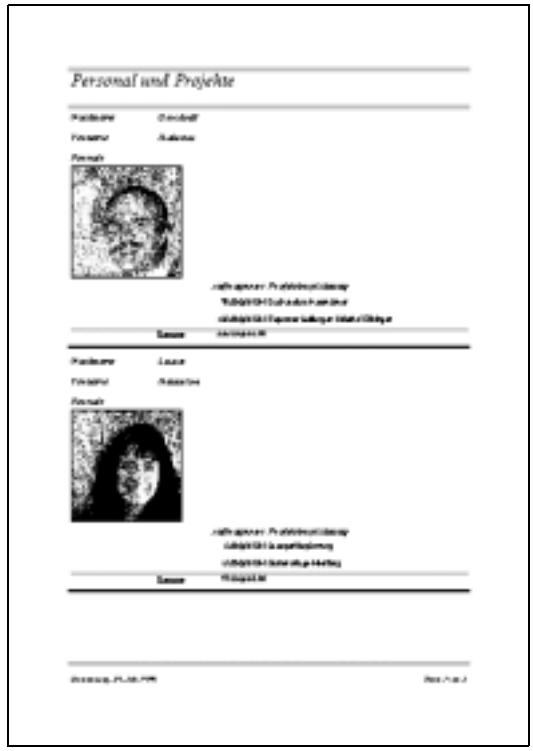


Bild 21.5: Bericht »Personal und Projekte« aus der Datenbank »Projekte.mdb«, nicht gezoomt

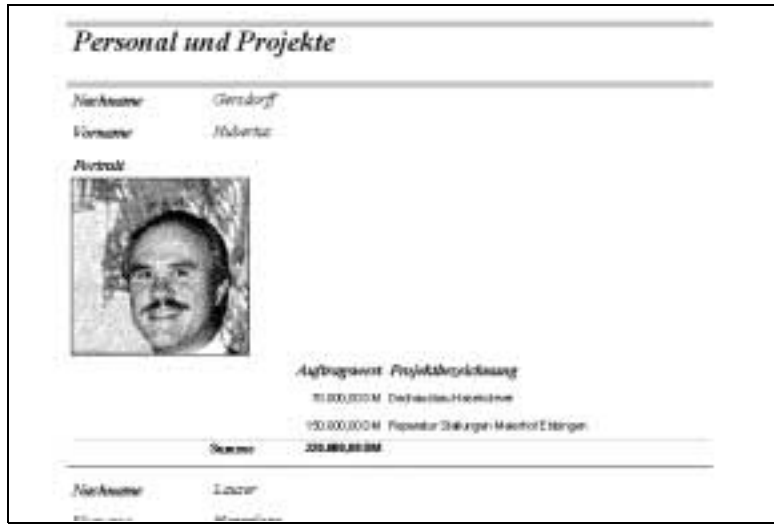


Bild 21.6: Bericht »Personal und Projekte« aus der Datenbank »Projekte.mdb«, Ausschnitt aus gezoomter Darstellung

Gehen Sie folgendermaßen vor, um den Bericht zu erstellen:

- Öffnen Sie die Datenbank Projekte.mdb von der Begleit-CD-ROM.
- Fordern Sie einen neuen Bericht an, indem Sie im Datenbankfenster die Objektliste *Berichte* aktivieren und dann auf die Schaltfläche *Neu* klicken. Alternativ klicken Sie auf die Symbol-Schaltfläche *Neues Objekt* und wählen darin *Bericht*. Access blendet das Dialogfeld *Neuer Bericht* ein, vgl. oben Bild 21.3.
- Wählen Sie im Dialogfeld *Neuer Bericht* den Berichts-Assistenten, und bestätigen Sie mit OK. Dann wird das erste Dialogfeld des Berichts-Assistenten angezeigt, vgl. Bild 21.7.

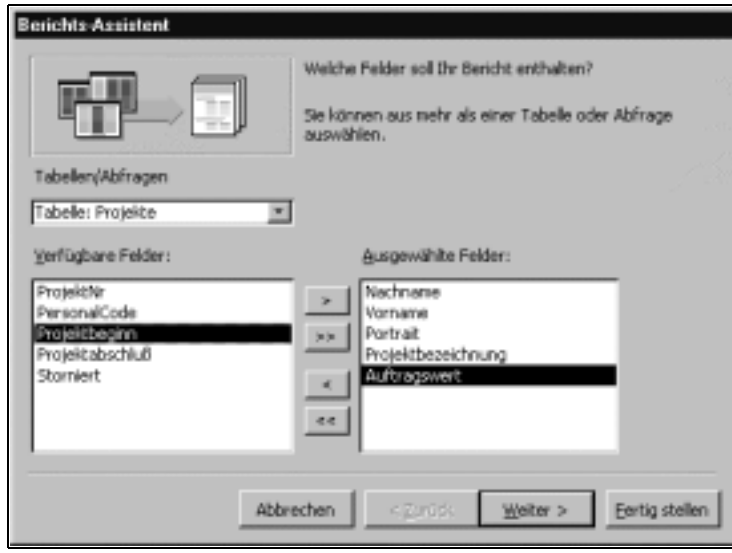


Bild 21.7: Erstes Dialogfeld des Berichts-Assistenten

- Wählen Sie im ersten Dialogfeld des Berichts-Assistenten aus der Liste *Tabellen/Abfragen* die Tabelle *Personal*. Dann werden in der Liste *Verfügbare Felder* die Felder dieser Tabelle angezeigt.
- Verschieben Sie aus der Liste *Verfügbare Felder* die Felder *Nachname*, *Vorname* und *Portrait* mit Hilfe der Schaltfläche > in die Liste *Ausgewählte Felder.*
- Wählen Sie aus der Liste *Tabellen/Abfragen* die Tabelle *Projekte*. Dann werden in der Liste *Verfügbare Felder* die Felder dieser Tabelle angezeigt.
- Verschieben Sie aus der Liste *Verfügbare Felder* die Felder *Projektbezeichnung* und *Auftragswert* mit Hilfe der Schaltfläche > in die Liste *Ausgewählte Felder.* Dann sollte das erste Dialogfeld des Berichts-Assistenten so aussehen, wie in Bild 21.7 wiedergegeben.

- Bestätigen Sie das Dialogfeld mit *Weiter>*. Dann zeigt sich das zweite Dialogfeld des Berichts-Assistenten, vgl. Bild 21.8.

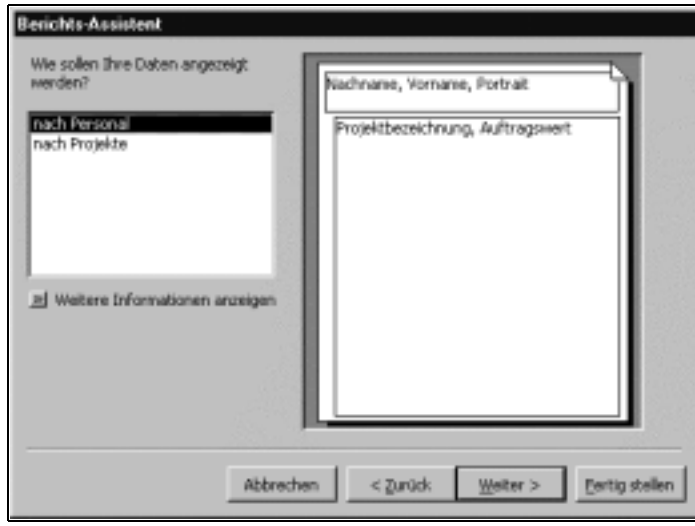


Bild 21.8: Zweites Dialogfeld des Berichts-Assistenten zur Wahl Darstellungshierarchie

- Belassen Sie es im zweiten Dialogfeld bei dem Vorschlag des Berichts-Assistenten, denn die beiden Tabellen *Personal* und *Projekte* stehen in einer 1:n-Beziehung, so daß die Tabelle *Personal* die oberste Darstellungsebene sein sollte. Bestätigen Sie mit *Weiter>*. Dann wird das dritte Dialogfeld des Berichts-Assistenten angezeigt, vgl. Bild 21.9.

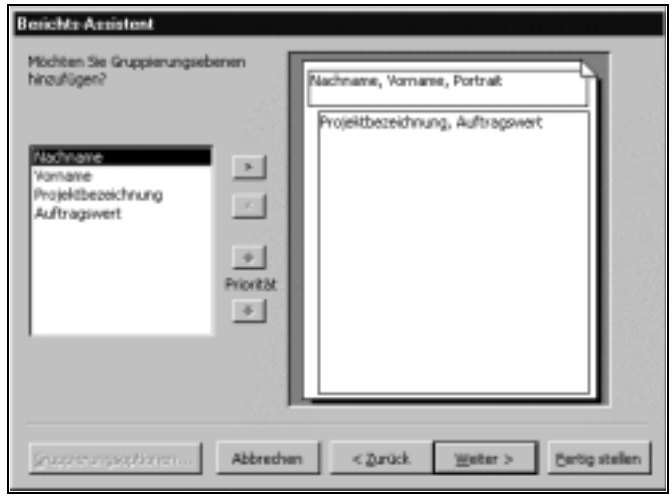


Bild 21.9: Drittes Dialogfeld des Berichts-Assistenten, in dem weitere Gruppierungsebenen hinzugefügt werden könnten

- Belassen Sie es auch im dritten Dialogfeld bei dem Vorschlag des Berichts-Assistenten, der mit den Feldern *Nachname*, *Vorname* und *Portrait* bereits eine – die richtige! – Gruppierungsebene vorgesehen hat, und weitere sind hier nicht sinnvoll. Bestätigen Sie das Dialogfeld mit *Weiter* >. Dann wird das vierte Dialogfeld des Berichts-Assistenten eingeblendet, vgl. Bild 21.10.

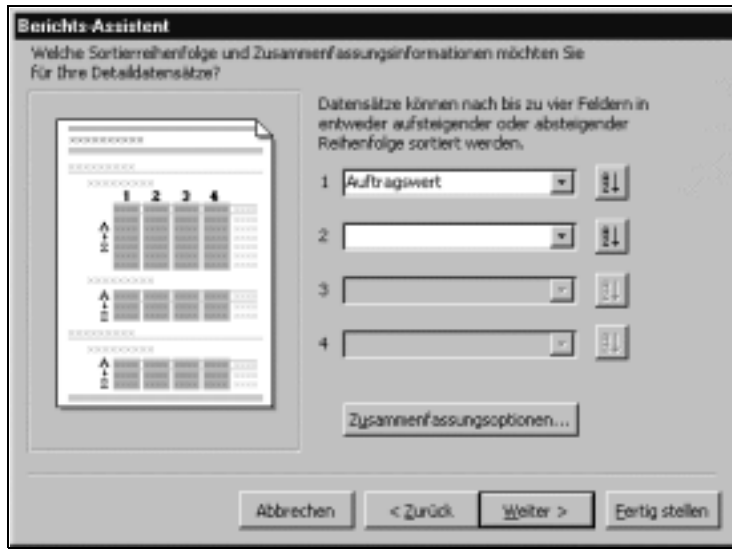


Bild 21.10: Viertes Dialogfeld des Berichts-Assistenten zur Bestimmung der Felder, nach denen sortiert werden soll. Hier wurde von mir bereits das Feld »Auftragswert« als Gruppierungsfeld gewählt.

- Schlagen Sie die oberste Dropdown-Liste auf, und wählen Sie das Feld *Auftragswert*, weil nach diesem Feld sortiert werden soll.
- Der Bericht soll auch für jede Gruppe die Summe der Auftragswerte ausweisen. Klicken Sie daher auf die Schaltfläche *Zusammenfassungsoptionen*. Dann wird das zusätzliche Dialogfeld *Zusammenfassungsoptionen* eingeblendet, vgl. Bild 21.11.
- Kreuzen Sie im Dialogfeld *Zusammenfassungsoptionen* das Kontrollkästchen *Summe* an, und bestätigen Sie mit OK. Bestätigen Sie dann das Dialogfeld des Berichts-Assistenten mit *Weiter* >. Dann wird dessen fünftes Dialogfeld angezeigt, vgl. Bild 21.12.
- Wählen Sie im fünften Dialogfeld die Optionen, wie sie in Bild 21.12 wiedergegeben sind, und bestätigen Sie mit *Weiter* >. Dann zeigt sich das sechste Dialogfeld des Berichts-Assistenten, vgl. Bild 21.13.

Bild 21.11: Dialogfeld »Zusammenfassungsoptionen«

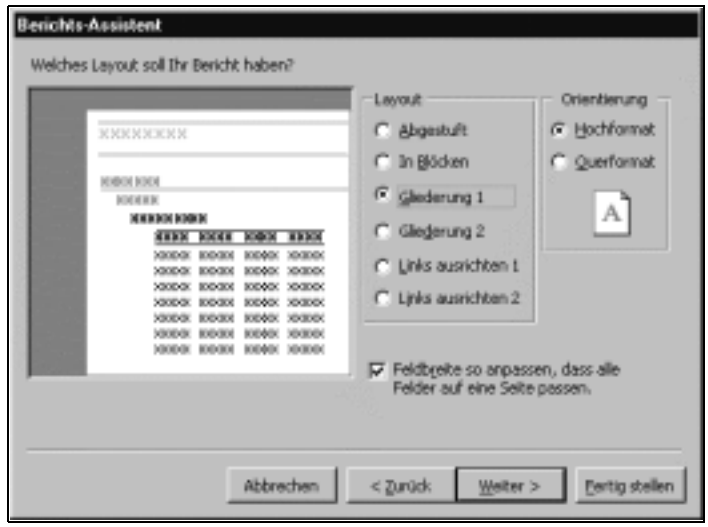


Bild 21.12: Fünftes Dialogfeld des Berichts-Assistenten zur Festlegung des Layouts

- Wählen Sie den Stil *Geschäftlich*. Bestätigen Sie mit der Schaltfläche *Weiter >*. Dann zeigt sich das siebte und letzte Dialogfeld des Berichts-Assistenten, vgl. Bild 21.14.
- Tragen Sie als Titel für den Bericht *Personal und Projekte* ein. Dann bekommt der Bericht diese Überschrift und wird unter diesem Namen gespeichert. Wählen Sie die Schaltfläche *Fertigstellen*. Dann wird der Bericht in der Seitenansicht gezeigt, vgl. Bild 21.15.

Bild 21.13: Sechstes Dialogfeld des Berichts-Assistenten zur Bestimmung des Berichtsstils

Bild 21.14: Letztes Dialogfeld des Berichts-Assistenten zum Festlegen eines Berichtstitels und weiterer Optionen

Der Bericht sollte nun so aussehen, wie in Bild 21.15 wiedergegeben. Aus ästhetischen, nicht zuletzt aber auch aus Übungsgründen soll er jedoch bearbeitet werden, und zwar in folgender Hinsicht:

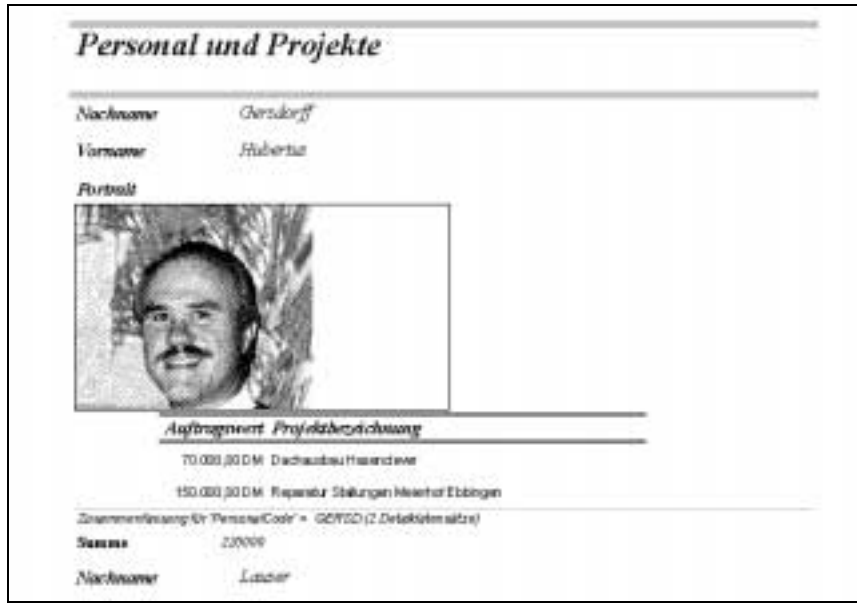


Bild 21.15: Gruppierter Bericht als unbearbeitetes Ergebnis des Berichts-Assistenten

- Der Rahmen um das Portrait herum ist zu groß: Er reicht zu weit nach rechts.
- Die Doppellinien, die sich oberhalb und unterhalb der Bezeichnungen *Auftragswert* und *Projektbezeichnung* befinden, sollen verschwinden.
- Die Felder *Auftragswert* und *Projektbezeichnung* samt ihren Bezeichnungsfeldern sollen so weit wie möglich nach rechts verschoben werden, damit sich die Detailebene von der darüber liegenden gruppierenden Ebene besser abhebt.
- Die Zeile *Zusammenfassung für ...* soll gelöscht werden, weil sie überflüssig erscheint.
- Das Feld, in dem die Summe der Auftragswerte der Gruppe ausgegeben wird (in Bild 21.15 weist es die Zahl 220000 aus), soll mit dem Format *Währung* und fett formatiert und so weit nach rechts verschoben werden, daß es unter den Auftragswerten des Detailbereichs steht.
- Das Bezeichnungsfeld mit der Beschriftung *Summe* soll unmittelbar links von dem Feld mit der Summe der Auftragswerte stehen.

Im Ergebnis soll der bearbeitete Bericht ungefähr so aussehen, wie in Bild 21.16 wiedergegeben.

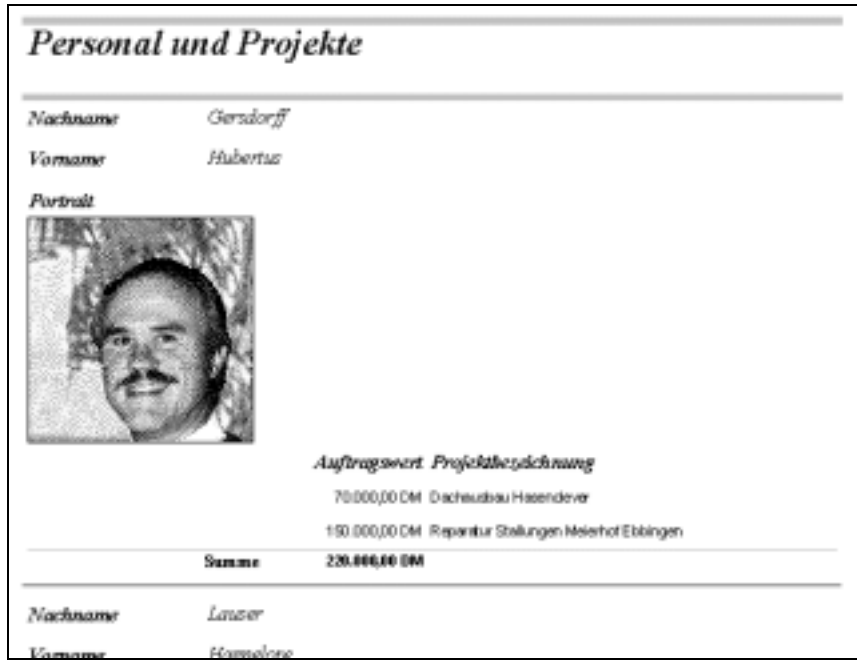


Bild 21.16: Bearbeiteter Bericht

Gehen Sie folgendermaßen vor, um die angeführten Änderungen auszuführen:

- Wechseln Sie in die Entwurfsansicht, indem Sie aus der Seitenansicht heraus auf die Esc-Taste drücken. Falls Sie den Bericht nicht gerade neu erstellt, sondern direkt vom Datenbankfenster aus in der Seitenansicht geöffnet hatten, kämen Sie mit Esc ins Datenbankfenster zurück; in diesem Falle müßten Sie den Bericht von dort als Entwurf öffnen. Sie sollten dann den Bericht in der Entwurfsansicht sehen wie in Bild 21.17 dargestellt.
- Markieren Sie das Objektfeld *Portrait*, und ziehen Sie dann den rechten Rahmen so, daß das Feld die Form eines Quadrats annimmt.
- Markieren Sie die Linien ober- und unterhalb der Bezeichnungsfelder *Auftragswert* und *Projektbezeichnung*, und löschen Sie diese, indem Sie die Entf-Taste drücken.
- Markieren Sie die Felder *Auftragswert* und *Projektbezeichnung* mit ihren Bezeichnungsfeldern, und verschieben Sie diese nach rechts.
- Markieren Sie das berechnete Textfeld mit der Formel

  ```
  ="Zusammenfassung für ..."
  ```

  im Personal-Code-Fußbereich, und löschen Sie dieses Textfeld.

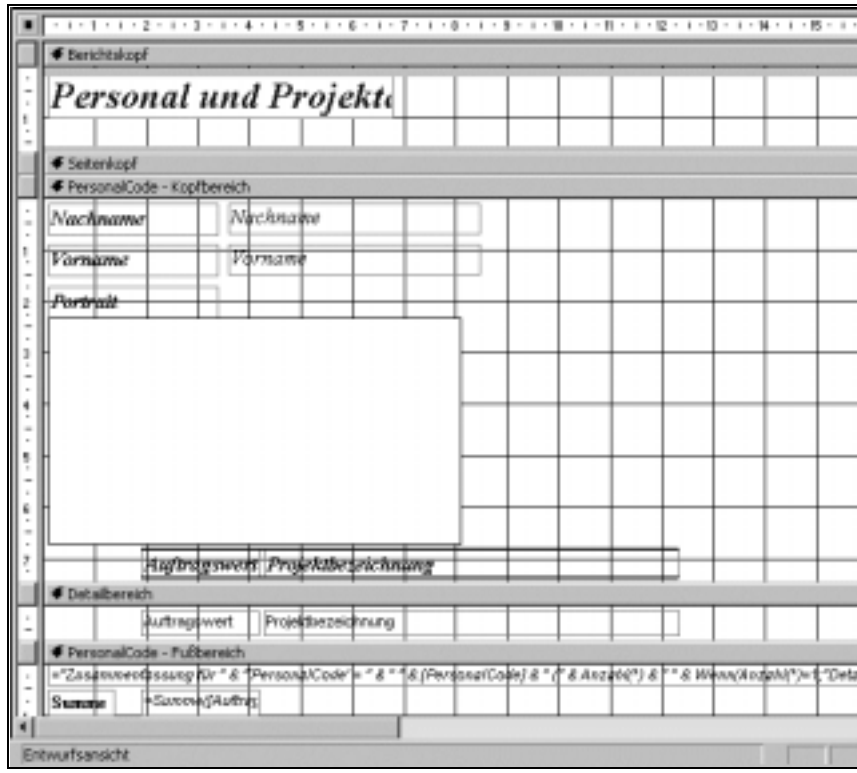


Bild 21.17: Der vom Berichts-Assistenten erstellte Gruppierungsbericht in der Entwurfsansicht, unbearbeitet

- Markieren Sie das Textfeld mit der Formel

  ```
  =Summe(Auftragswert)
  ```

  und nehmen Sie daran folgende Bearbeitungen vor:

  - Verschieben Sie es so, daß es unterhalb des Feldes *Auftragswert* im Detailbereich steht.
  - Stellen Sie die Eigenschaft *Format* auf *Währung* ein. Nehmen Sie diese Bearbeitung im Eigenschaftenfenster vor.
  - Klicken Sie auf die Schaltfläche *Fett* in der Format-Symbolleiste, während das Textfeld markiert ist.
  - Verschieben Sie das Bezeichnungsfeld *Summe* so, daß es links vom Feld mit der Zusammenfassung der Auftragswerte steht.
  - Wechseln Sie in die Seitenansicht um zu prüfen, ob die Änderungen so ausgefallen sind, wie dies gewünscht wurde.

Besonderheit im Bericht: Gruppenkopf- und -fuß

Bild 21.17 läßt erkennen, daß die Entwurfsansicht eines Berichts derjenigen eines Formulars auf den ersten Blick ähnlich ist. Sie können jedoch zwei Bereichsbezeichnungen sehen, wie sie in der Entwurfsansicht eines Formulars nie vorkommen: Die eine lautet *PersonalCode - Kopfbereich*, die andere *PersonalCode - Fußbereich*. Diese Überschriften geben den Beginn eines *Gruppenkopf*- bzw. *Gruppenfuß*-Bereiches an.

Der Berichts-Assistent hat dem Gruppenfuß mit der Bezeichnung *PersonalCode - Fußbereich* ein berechnetes Textfeld hinzugefügt. Dieses Feld enthält als Steuerelementinhalt den Ausdruck =*Summe([Auftragswert])*. Damit werden alle Auftragswerte, die zu einer Gruppe gehören, aufsummiert. Beispielsweise wird oben in Bild 21.17 für den Mitarbeiter Gersdorff die Auftragssumme in fetter Schrift mit 220.000 DM ausgewiesen. Diese Summe setzt sich aus den Auftragswerten der beiden von ihm betreuten Projekte von 70.000 DM und 150.000 DM zusammen. Entsprechend werden die gesamten Auftragssummen für jeden anderen Mitarbeiter ausgewiesen. Eine derartige Summenbildung, die sich nicht auf den ganzen Bericht, sondern auf Gruppen von Datensätzen darin bezieht, ist nur in Berichten möglich; in einem Formular ergibt die Funktion *Summe* stets den Gesamtwert aller in der Datenbasis des Formulars (ggf. auch des Unterformulars) vorhandenen Datensätze, gleichgültig, ob sie in einem Ausdruck im Detailbereich, Formularkopf/-fuß oder Seitenkopf/-fuß angeführt ist.

21.4 Leeren Bericht erstellen und bearbeiten

Statt einen Bericht mit dem Berichts-Assistenten erstellen zu lassen und dann ggf. zu bearbeiten, können Sie ihn von Anfang an selbst aufbauen. Gehen Sie dazu folgendermaßen vor:

- Fordern Sie einen neuen Bericht an, indem Sie im Datenbankfenster die Objektliste *Berichte* aktivieren und dann auf die Schaltfläche *Neu* klicken. Alternativ schlagen Sie die Dropdown-Liste der Symbol-Schaltfläche *Neues Objekt* auf und wählen darin *Bericht*. Access blendet das Dialogfeld *Neuer Bericht* ein. Als dritte Möglichkeit doppelklicken Sie in der Objektliste *Berichte* auf den Eintrag *Erstellt einen Bericht in der Entwurfsansicht*.
- Wählen Sie im Dialogfeld *Neuer Bericht* eine Tabelle oder Abfrage, oder lassen Sie das entsprechende Feld frei, weil Sie die Datenherkunft des Berichts auch später im Eigenschaftenfenster des Berichts angeben können. Markieren Sie im Listenfeld die Kategorie *Entwurfsansicht*, und bestätigen Sie mit OK. Sie erhalten einen leeren Bericht in der Entwurfsansicht.

- Bearbeiten Sie den neuen Bericht, indem Sie ihm im Eigenschaftenfenster für den Bericht ggf. eine Tabelle/Abfrage oder eine SQL-SELECT-Anweisung als Datenherkunft zuweisen, Berichts- und Seitenkopf/-fuß sowie Gruppenbereiche (vgl. dazu genauer Kap. 22, *Komplexe Berichte*, Punkt 22.3.1, *Gruppierungsebenen hinzufügen*) und Steuerelemente verschiedenster Art hinzufügen. Betrachten Sie den Bericht in der Seitenansicht, und speichern oder verwerfen Sie ihn schließlich.

21.5 Bericht drucken

Bericht drucken

Um einen Bericht zu drucken, verfahren Sie ganz analog zum Drucken eines Formulars: Wählen Sie, während der Bericht den Fokus hat (ggf. auch im Datenbankfenster!) den Befehl *Drucken* aus dem Menü *Datei*, oder klicken Sie, wenn Sie sich in der Seitenansicht befinden, auf die Symbol-Schaltfläche *Drucken*. Bevor Sie einen Bericht ausdrucken, sollten Sie jedoch gewisse Vorbereitungen für den Druck treffen, z.B. Seitenränder einstellen, Hoch- oder Querformat wählen etc. Die Hinweise in Kap. 15, *Formulare: Grundlagen*, Punkt 15.10, *Formular für den Druck einrichten*, gelten ganz entsprechend auch für Berichte.

Berichtsdefinition drucken

Beachten Sie, daß Sie auch die Definitionen des Berichts drucken lassen können. Wählen Sie dazu den Befehl *Analyse*, Unterbefehl *Dokumentierer*, aus dem Menü *Extras*, und stellen Sie die verschiedenen Optionen ein, bevor Sie die Ausgabe der Definitionen als Bericht veranlassen.

Kapitel 22

Komplexe Berichte

22.1 Mehrspaltenbericht

Sie können einen Bericht mehrspaltig ausgeben lassen. Im Prinzip brauchen Sie dazu nur eine kleine Anweisung im Dialogfeld des Befehls *Seite einrichten* zu geben. Jedoch müssen Sie möglicherweise auch Anpassungen in der Entwurfsansicht vornehmen, damit Sie zufriedenstellende Ergebnisse erzielen. Gehen Sie im einzelnen folgendermaßen vor:

- Wählen Sie, während der betreffende Bericht den Fokus hat (auch Markierung im Datenbankfenster reicht für diesen Zweck), den Befehl *Seite einrichten* im Menü *Datei*.
- Aktivieren Sie im Dialogfeld *Seite einrichten* die Registerkarte *Spalten*. Geben Sie dort die gewünschte Spaltenzahl und den Spaltenabstand an, und bestätigen Sie mit OK.
- Prüfen Sie in der Seitenansicht, ob alle Daten vollständig wiedergegeben werden. Wenn dies nicht zutrifft, sollten Sie in der Entwurfsansicht prüfen, ob die dort definierte Breite der Bereiche, soweit sie in der Horizontalen Steuerelemente enthalten, nicht größer ist als eine Spaltenbreite. Wenn Sie beispielsweise nach Abzug der Seitenränder 19 cm Platz in der Horizontalen haben und für den Bericht zwei Spalten mit einem Spaltenabstand von 1 cm festgelegt haben, ist jede Spalte 9 cm breit. Mithin darf der rechte Rand des am weitesten rechts liegenden Steuerelements im Bericht nicht weiter als 9 cm vom linken Berichtsrand entfernt sein.

Falls Sie einen mehrspaltigen Bericht zum Ausdrucken von Adreßetiketten erstellen wollen, sollten Sie den Etiketten-Assistenten benutzen. Er erfüllt in dieser Hinsicht praktisch jeden Wunsch, der sich in Bezug auf gängige Etikettenvorlagen stellt.

22.2 Sortieren

Sortierung erstellen

Um die Datensätze eines Formulars nach den Inhalten bestimmter Felder sortieren zu lassen, müssen Sie, abgesehen vom Primärschlüssel, eine Abfrage mit sortierten Datensätzen erstellen und diese als Datenherkunft für das Formular angeben. In einem Bericht können Sie dagegen unmittelbar für die Ausgabe sortieren lassen, so daß Sie ggf. auch eine unsortierte Tabelle als Datenherkunft heranziehen können. Gehen Sie folgendermaßen vor, um einen Bericht nach einem oder mehreren Feldern sortieren zu lassen:

- Öffnen Sie den Bericht in der Entwurfsansicht.
- Blenden Sie das Fenster *Sortieren und Gruppieren* ein (vgl. Bild 22.1): Klicken Sie auf die gleichnamige Symbol-Schaltfläche (vgl. links nebenstehend), oder wählen Sie den Befehl *Sortieren und Gruppieren* aus dem Menü *Ansicht*.

Bild 22.1: Fenster »Sortieren und Gruppieren«

- Klicken Sie im Fenster *Sortieren und Gruppieren* in eine Zeile der Spalte *Feld/Ausdruck*, schlagen Sie das Dropdown-Listenfeld auf, und wählen Sie das Feld, nach dem sortiert werden soll. Statt ein Feld auszuwählen, können Sie auch einen Ausdruck, der mit einem Gleichheitszeichen beginnt, eintragen: Nehmen Sie beispielsweise an, in einer Anschriftendatei seien Postleitzahl und Ort stets zusammen im selben Feld *Ort* angegeben, so daß der Ortsname erst mit dem siebten Zeichen beginnt (fünf Ziffern und ein Leerzeichen für die Postleitzahl). Wenn Sie gleichwohl nach dem Ortsnamen sortieren lassen wollen, können Sie den folgenden Ausdruck eingeben:

```
=TeilStr([Ort];7)
```

- Wählen Sie in der Spalte Sortierreihenfolge *Aufsteigend* oder *Absteigend*.
- Wiederholen Sie die letzten beiden Schritte für jedes weitere Feld, nach dem sortiert werden soll.

Sortierung aufheben

Sie heben eine Sortierung auf, indem Sie im Fenster *Sortieren und Gruppieren* die betreffende Zeile durch Klicken auf den Feldmarkierer am linken Rand markieren und sie dann mit der [Entf]-Taste löschen.

22.3 Gruppieren

22.3.1 Gruppierungsebenen hinzufügen

Sie fügen einem Bericht eine (erste oder weitere) Gruppierungsebene hinzu, indem Sie für ein Feld oder einen Ausdruck einen (ersten oder weiteren) Gruppenkopf oder -fuß festlegen. Dies geschieht mit Hilfe des Fensters *Sortieren und Gruppieren*. Im Bericht *Personal und Projekte*, der in Kap. 22, *Berichte: Grundlagen*, Punkt 22.3.1, *Bericht mit dem Berichts-Assistenten erstellen*, mit Hilfe des Berichts-Assistenten erstellt wurde, wird nach dem Feld *PersonalCode*, das die Nachnamen der Mitarbeiter enthält, gruppiert. Wenn Sie diese Gruppierung statt mit dem Berichts-Assistenten selbständig definieren wollen, müssen Sie folgendermaßen vorgehen:

- Öffnen Sie den Bericht in der Entwurfsansicht, und blenden Sie ggf. das Fenster *Sortieren und Gruppieren* ein: Klicken Sie auf die entsprechende Symbol-Schaltfläche (vgl. links nebenstehend), oder wählen Sie den Befehl *Sortieren und Gruppieren* aus dem Menü *Ansicht*.
- Schreiben Sie den Feldnamen *PersonalCode* in eine Zeile der Spalte *Feld/Ausdruck*, oder wählen Sie ihn mittels des Dropdown-Listenfeldes aus.
- Bestimmen Sie die Sortierreihenfolge.
- Wählen Sie im unteren Teil *Gruppeneigenschaften* für die Eigenschaften *Gruppenkopf* und/oder *Gruppenfuß* die Einstellung *Ja*; zum Definieren einer Gruppierungsebene reicht es, wenn Sie entweder einen Gruppenkopf oder einen Gruppenfuß festlegen.
- Falls Sie nicht nach dem gesamten Feldinhalt, sondern nach einer bestimmten Anzahl von Anfangszeichen (bei Feldern mit dem Datentyp *Zahl* z.B. auch nach Intervallen) gruppieren wollen, müßten Sie noch Angaben zu *Gruppieren nach* sowie zu *Intervall* vornehmen, vgl. dazu weiter unten in diesem Abschnitt den Punkt 22.3.2, *Gruppierungseinheit und -intervall festlegen*. Ferner können Sie – ähnlich wie bei einem Textverarbeitungsprogramm für aufeinanderfolgende Absätze – festlegen, daß Ergebnisse einer Gruppe zusammengehalten werden, d.h. nicht auf die nächste Seite umbrochen werden. In diesem Falle stellen Sie die Eigenschaft *Zusammenhalten* auf *Ja* ein.

Wenn Sie die beschriebenen Schritte vollzogen haben, sollte das Fenster *Sortieren und Gruppieren* aussehen, wie in Bild 22.2 wiedergegeben. Beachten Sie vor allem das Gruppierungssymbol, welches am linken Rand der Zeile

mit dem Eintrag *PersonalCode* eingeblendet ist; Access zeigt dieses Symbol in diesem Fenster für jedes Feld an, für das ein Gruppenkopf und/oder -fuß festgelegt ist.

Bild 22.2: Fenster »Sortieren und Gruppieren« mit Angaben zum Gruppieren nach dem Feld »*PersonalCode*«. Beachten Sie das Gruppierungssymbol am linken Rand der Zeile mit dem Feld »*PersonalCode*«, welches anzeigt, daß nach diesem Feld gruppiert wird.

Wenn Sie auf die beschriebene Weise eine Gruppierungsebene definiert haben, richtet Access automatisch in der Entwurfsansicht des Berichts die zugehörigen Bereiche ein: Einen *Gruppenkopf* und/oder *-fuß* und dazwischen (bzw. darunter oder darüber) einen zugehörigen *Detailbereich*. Sie müssen diese Bereiche dann bearbeiten, denn sie sind zunächst völlig leer. Vor allem müssen Sie diese neu definierten Bereiche mit einem oder mehreren Steuerelementen wie Textfeldern, Kontrollkästchen usw. versorgen, damit überhaupt Daten dargestellt werden können. Die Bereichshöhe können Sie wie üblich, z.B. durch Ziehen mit der Maus, verändern, im Einzelfalle auch auf die Größe Null bringen.

Auf dieselbe Weise, wie vorangehend beschrieben, können Sie weitere Gruppierungsebenen festlegen. Dabei dürfen Sie, genau wie beim Sortieren (vgl. den obigen Punkt 22.2, *Sortieren*), statt eines Feldnamens auch einen Ausdruck, der Gruppierungswerte ergibt, angeben. Beispielsweise könnten Sie mit dem Ausdruck

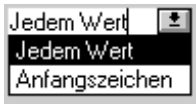
```
=Monat([Bestelldatum])
```

nach Bestellmonaten gruppieren lassen. Für jede weitere Gruppierungsebene, die Sie definieren, fügt Access in der Entwurfsansicht des Berichts den zugehörigen Gruppenkopf/-fuß ein. Wenn mehr als eine Gruppierungsebene definiert ist, wird in folgender Weise gruppiert: Zuerst wird nach dem Feld bzw. Ausdruck gruppiert, das/der in der Liste des Fensters *Sortieren und Gruppieren* als erster steht, dann, innerhalb der einzelnen Kategorien dieser Gruppierungsebene, nach dem Feld bzw. Ausdruck, welches(r) in der Feldliste an nächster Stelle steht usw. In entsprechender Reihenfolge ordnet Access die zugehörigen Gruppenbereiche in der Entwurfsansicht des Berichts an.

22.3.2 Gruppierungseinheit und -intervall festlegen

Standardmäßig stellt Access die Gruppierungseigenschaften so ein, daß nach jedem Wert des Gruppierungsfeldes bzw. -ausdrucks gruppiert wird. Sie können statt dessen jedoch Bereiche und Intervalle zum Gruppieren festlegen. Dies geschieht ebenfalls im Fenster *Sortieren und Gruppieren*. Stellen Sie dort die Eigenschaften *Gruppieren nach* und *Intervall* ein: Mit *Gruppieren nach* bestimmen Sie die Einheit, in der das *Intervall* gemessen wird. Je nach Datentyp bietet Access unterschiedliche Gruppierungseinheiten an:

Textfelder



Hierfür können Sie nur nach den beiden Einheiten *Jedem Wert* und *Anfangszeichen* gruppieren; zu *Jedem Wert* paßt dann auch nur das Intervall 1. Zu *Anfangszeichen* können Sie dagegen eine Zahl n angeben, so daß dann nicht nach der vollen Textlänge, sondern nach den n ersten Zeichen gruppiert wird.

Datum/Uhrzeit

- Für Felder dieses Datentyps stehen diverse Zeiteinheiten wie *Jahr*, *Quartal* ... *Minute* zur Verfügung, zu denen Sie jeweils einen Intervallwert angeben können. Mit der Einheit *Monat* und dem Intervall 3 würden Sie dann beispielsweise dieselbe Wirkung erzielen wie mit der Einheit *Quartal* und dem Intervall 1.

AutoWert, Währung und Zahl

- Hierfür stehen die Einheiten *Jedem Wert* und *Intervall* zur Verfügung. Bei Wahl von *Intervall* geben Sie eine Zahl an, die größer als 1 ist. Dann werden Gruppierungintervalle entsprechend dieser Größe gebildet, die bei dem Wert 0 beginnen. Wenn Sie beispielsweise als Intervall den Wert 100 angeben, bildet Access die folgenden Bereiche zum Gruppieren: 0-99, 100-199, 200-299 etc.

22.3.3 Gruppierungs- und Sortierreihenfolge ändern

Sie können jederzeit die Reihenfolge des Gruppierens und Sortierens ändern. Dies gilt auch dann, wenn Sie bereits Steuerelemente in Gruppenbereiche eingefügt haben, denn Access paßt die Gruppenbereiche im Berichtsentwurf automatisch an.

Um die Gruppierungs- oder Sortierreihenfolge zu ändern, gehen Sie wie folgt vor:

- Blenden Sie ggf. das Fenster *Sortieren und Gruppieren* ein.
- Markieren Sie die Zeile des zu verschiebenden Feldes, indem Sie auf den Feldmarkierer am linken Zeilenrand klicken; bei einem Gruppierungsfeld trägt der Feldmarkierer das Symbol für *Gruppieren*, vgl. z.B. Bild 22.3.

- Klicken Sie erneut auf den Feldmarkierer der markierten Zeile, und ziehen Sie nach oben oder unten an die gewünschte Stelle; eine mitlaufende waagerechte Linie zeigt an, wohin das Feld verschoben wird, wenn Sie die Maustaste loslassen, vgl. Bild 22.3.

Bild 22.3: Das Gruppierungsfeld »Einzelpreis« wird zwischen die Felder »Empfänger« und »Bestelldatum« geschoben.

Unmittelbar nach dem Verschieben eines Gruppierungsfeldes paßt Access die Gruppenbereiche im Berichtsentwurf an. Wenn Sie beispielsweise ein gruppierendes Feld, das bisher an erster Stelle stand, so verschieben, daß es an dritter Stelle der Gruppierungsebenen steht, wird sein Gruppenbereich ebenfalls an dritter Stelle stehen.

22.4 Haupt- und Unterbericht

Ganz analog zu Formularen können Sie auch in einen Bericht einen oder mehrere Unterberichte einfügen. Die dafür geltenden Zusammenhänge, insbesondere die Verfahren zum Herstellen von Haupt- und Unterberichten, können Sie dem Kap. 18, *Haupt- und Unterformular*, entnehmen. Auch die Ausführungen zum Verknüpfen von Haupt- und Unterformular gelten ganz entsprechend für Haupt- und Unterberichte. Die einzige Besonderheit für Berichte ergibt sich aus deren besonderer Möglichkeit, gruppierte Berichte zu erstellen: Sie können einen Unterbericht auch in einen Gruppenbereich einfügen, vgl. den folgenden Punkt 22.4.1, *Verknüpfter Unterbericht in einem gruppierten Hauptbericht*.

22.4.1 Verknüpfter Unterbericht in einem gruppierten Hauptbericht

Beispiel

Ein gutes Beispiel für einen gruppierten Bericht, in den ein verknüpfter Unterbericht eingefügt ist, bietet der Bericht *Umsätze nach Jahr* der Datenbank *Nordwind.mdb*. Der Bericht listet die Versand-Datumswerte, Bestellnummern und Umsätze auf, wobei das Versanddatum als Gruppierungsfeld definiert ist.

Bild 22.4: Entwurfsansicht des Hauptberichts »Umsätze nach Jahr«, in den der verknüpfte Unterbericht »UmsätzeNachJahrUnterbericht« eingebettet ist.

In den *Versanddatum-Kopfbereich* dieses Berichts ist der Bericht *UmsätzeNachJahrUnterbericht* als Unterbericht eingebettet. Der Unterbericht stellt die Anzahl von Bestellungen sowie die Umsätze für die Quartale des im Hauptbericht gerade dargestellten Versandjahres dar. Dies wird dadurch erreicht, daß der Unterbericht mit dem Hauptbericht verknüpft ist. Dabei wird vom Feld *Jahr* des Unterberichts zum Feld *Jahr* des Hauptberichts verknüpft, vgl. Bild 22.5.

Bild 22.5: Wesentliche Eigenschaften des verknüpften Unterberichts »UmsätzeNachJahrUnterbericht«. Beachten Sie insbesondere die Eigenschaften »Verknüpfen von« und »Verknüpfen nach« sowie die Eigenschaft »Vergrößerbar«.

Das Ergebnis dieses Haupt- mit Unterberichtes ist in Bild 22.6 und Bild 22.7 zu sehen. Dargestellt werden die Zahlen für den Zeitraum 01.01.1996 – 31.12.1997 (beim Öffnen des Berichts in der Seitenansicht müssen Sie Datumsangaben treffen). Ich habe hier die *beiden* Darstellungen wiedergegeben, um zu zeigen, daß Haupt- und verknüpfter Unterbericht im Prinzip entsprechend funktionieren wie ein Haupt- mit verknüpftem Unterformular:

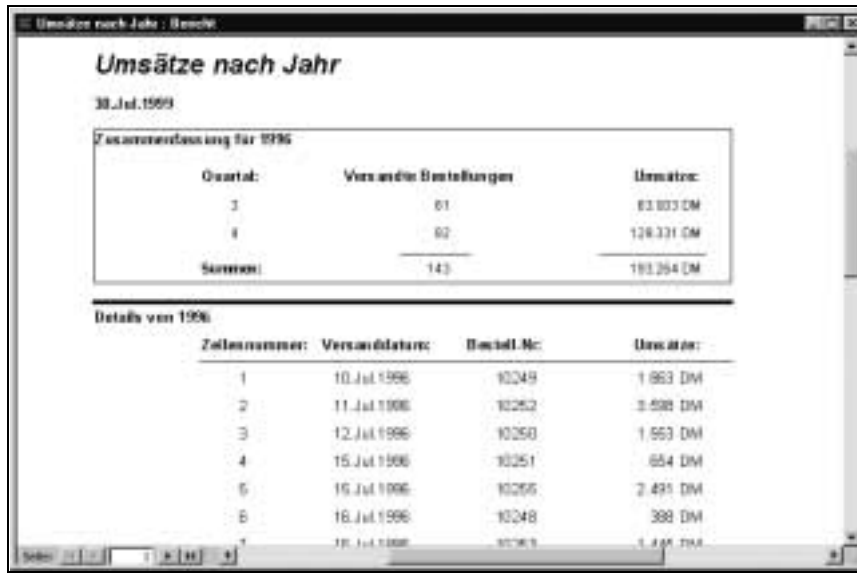
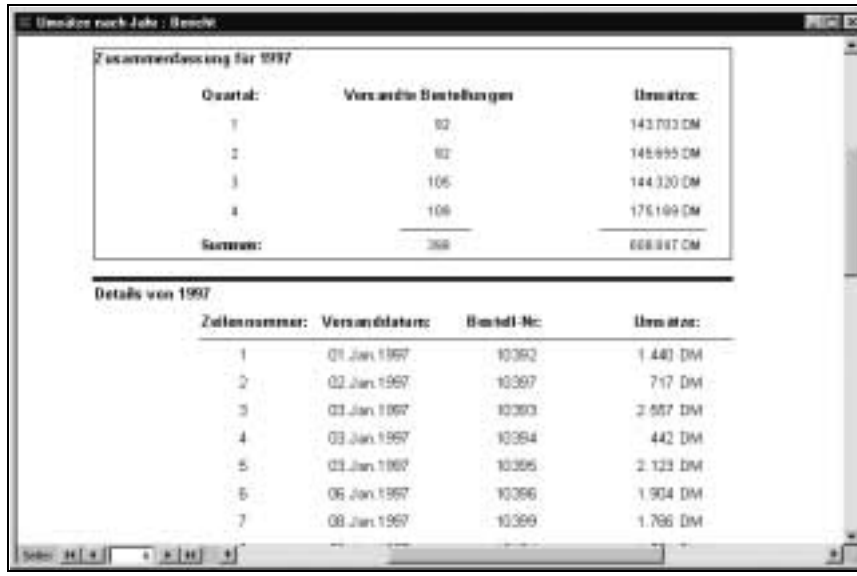


Bild 22.6: Seitenansicht des Berichts »Verkaufszahlen nach Jahren« für den Zeitraum 01.01.1996 – 31.12.1997, Seite 1.

Seite 1 der Seitenansicht (und damit des Drucks) gibt die erste Gruppe der Zusammenfassung im Gruppenkopf als *Zusammenfassung für 1996* wieder. (*Anmerkung*: Die Datenbasis enthält nur Datensätze mit Werten für das Versanddatum >= 10.07.96, so daß die Angaben für 1996 erst mit dem dritten Quartal beginnen.) Unter den zusammengefaßten Zahlen erscheinen im Detailbereich alle Einzeldaten für 1996, die sich über vier Seiten erstrecken. Auf Seite 6 beginnen dann die Zahlen für 1997, die wiederum mit einer Zusammenfassung für 1997 beginnen und mit den Detailangaben für dieses Jahr fortgesetzt werden.



Zusammenfassung für 1997

Quartal:	Versandte Bestellungen	Umsatz:
1	92	143.703 DM
2	92	146.995 DM
3	106	144.320 DM
4	109	175.199 DM
Summen:	399	608.917 DM

Details von 1997

Zeilennummer:	Versanddatum:	Bestell-Nr:	Umsatz:
1	01.Jan.1997	10392	1.440 DM
2	02.Jan.1997	10397	717 DM
3	03.Jan.1997	10393	2.557 DM
4	03.Jan.1997	10394	442 DM
5	03.Jan.1997	10395	2.123 DM
6	06.Jan.1997	10396	1.904 DM
7	08.Jan.1997	10399	1.786 DM

Bild 22.7: Seitenansicht des Berichts »Verkaufszahlen nach Jahren« für den Zeitraum 01.01.1996 – 31.12.1997, Seite 6.

Kapitel 23

Ausdrücke in Berichten

23.1 Zusammenfassende Funktionen

Sie können einen Ausdruck mit einer zusammenfassenden Funktion wie z.B. *Summe*, *Mittelwert* oder *Anzahl* im Berichtskopf oder -fuß, im Detailbereich sowie im Gruppenkopf oder -fuß anführen. Im Berichtskopf/-fuß sowie im Detailbereich werden alle Datensätze der dem Bericht zugrunde liegenden Datenbasis (Tabelle, Abfrage) in die Zusammenfassung einbezogen. Im Gruppenkopf/fuß werden dagegen nur die Datensätze, die zur jeweiligen Gruppe gehören, berücksichtigt.

Beispiel

Die Datenbank *Projekte.mdb* von der Begleit-CD-ROM enthält einen Bericht mit dem Namen *Personal und Projekte*, der in Kap. 21, *Berichte: Grundlagen*, Punkt 21.3.1, *Bericht mit dem Berichts-Assistenten erstellen*, erstellt und bearbeitet wurde. Es handelt sich um einen Gruppierungsbericht mit den Tabellen *Personal* und *Projekte* als Datenherkunft. Gruppierungsfeld ist *PersonalCode*. Die Entwurfsansicht dieses Berichts ist in Bild 23.1 wiedergegeben. Beachten Sie, daß die Funktion *Summe()* an zwei Stellen des Berichts verwendet wird: Einmal wird sie im *PersonalCode-Fußbereich* angeführt, ein weiteres Mal im Berichtsfuß.

Die unterschiedliche Wirkung der Summenfunktion in Abhängigkeit von der Position im Berichtsentwurf gibt Bild 23.2, das die Seitenansicht des Berichts zeigt, wieder: Die Gesamtsumme der Aufträge beläuft sich auf einen Betrag von 5.363.600 DM. Die Gruppensumme, die ebenfalls mit der Funktion *Summe*, jedoch im Gruppenfuß plaziert, ermittelt wird, beläuft sich dagegen für die zuletzt im Bericht ausgewiesene Gruppe auf 3.610.000 DM.

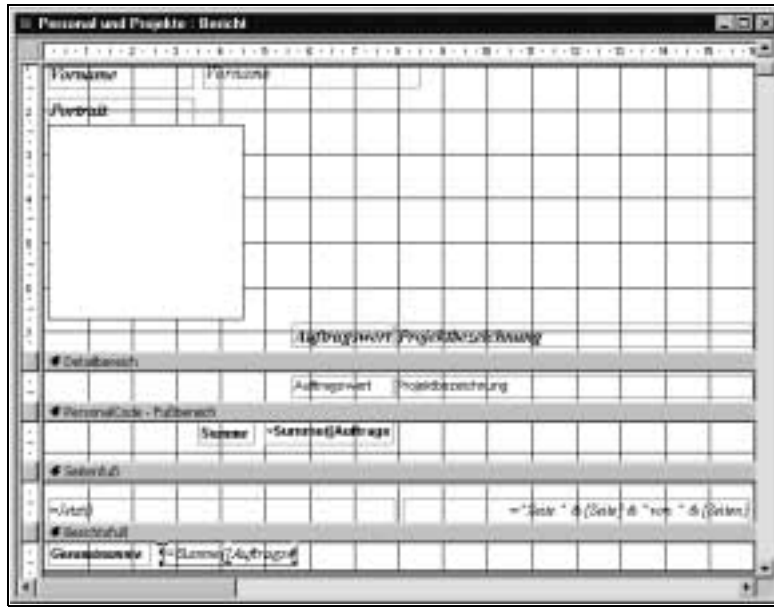


Bild 23.1: Bericht »Personal und Projekte« aus der Datenbank »Projekte.mdb« von der Begleit-CD-ROM

Bild 23.2: Seitenansicht des Berichtsentwurfs aus Bild 23.1

Mit den Ergebnissen von Textfeldern, die zusammenfassende Funktionen enthalten, kann selbst wiederum gerechnet werden. Um beispielsweise die jeweilige Gruppensumme als Prozentsatz der Gesamtsumme auszuweisen, könnten Sie den folgenden Ausdruck in ein Textfeld schreiben, welches im Gruppenfuß positioniert ist:

```
=[Gruppensumme]/[Gesamtsumme]
```

Dabei ist vorausgesetzt, daß die Textfelder zur Ermittlung der Gruppensumme bzw. der Gesamtsumme die Namen *Gruppensumme* bzw. *Gesamtsumme* besitzen. Zweckmäßigerweise sollten Sie dem Textfeld, welches die Prozentberechnung ausgibt, das Format *Prozentzahl* geben (Eigenschaft *Format* im Eigenschaftenfenster).

23.2 Textfeldeigenschaft *Laufende Summe*

Vorname Hermann

Portrait

Auftragswert	Auftragswert kumuliert	Projektbezeichnung
3.600,00 DM	3.600,00 DM	Türdurchbruch Meier-Seelmann
23.000,00 DM	26.600,00 DM	Sielreparatur Erdmannsweg
120.000,00 DM	146.600,00 DM	Grundschule Hermannsdorf Toilettenneubau
370.000,00 DM	516.600,00 DM	Neubau Einfamilienhaus Ottingen
1.000.000,00 DM	1.516.600,00 DM	Sportplatz Grundschule Heisterbach
Summe	1.516.600,00 DM	

Bild 23.3: Bericht mit »Laufende Summe« für die Spalte »Auftragswert kumuliert«

Im Bericht in Bild 23.3 wird der Auftragswert auch in kumulierter Form ausgewiesen. Dem liegt eine Bearbeitung des o.a. Berichts *Personal und Projekte* zugrunde. Access nennt einen derartigen Sachverhalt *Laufende Summe*. Eine laufende Summe ermittelt die Summe aller vorangehenden Datensätze für das Feld, das aktuelle eingeschlossen. Diese Leistung wird nicht mit Hilfe einer Funktion erreicht, wie man vermuten könnte, sondern durch entsprechende Einstellung einer Eigenschaft: In Berichten hat das Steuerelement *Textfeld* die Eigenschaft *Laufende Summe*, die Sie auf *Nein*,

Über Gruppe oder *Über Alles* einstellen können. Der Bericht in Bild 23.3 beispielsweise enthält zwei Textfelder, jeweils mit dem Feldnamen *Auftragswert* als Steuerelementinhalt. Die beiden Textfelder geben dennoch unterschiedliche Ergebnisse aus, weil die Eigenschaft *Laufende Summe* beim linken auf *Nein*, beim rechten dagegen auf *Über Gruppe* eingestellt ist. Beachten Sie im übrigen, daß die letzte *Laufende Summe* einer Gruppe mit dem Ergebnis der Funktion *Summe* für dieselbe Gruppe beim selben Feld übereinstimmen muß, vgl. Bild 23.3, andernfalls läge ein Fehler vor.

23.3 Doppelte Feldinhalte ausblenden

Normalerweise können Sie Duplikate nur als ganze Datensätze ausblenden, indem Sie eine Abfrage konzipieren, bei der die Eigenschaft *Keine Duplikate* (vgl. Fenster *Abfrageeigenschaften*) auf Ja gesetzt ist. In Berichten ist es dagegen möglich, die Anzeige von mehrfach gleichen Feldinhalten zu unterdrücken, ohne die anderen Felder desselben Datensatzes ebenfalls zu unterdrücken. Dieses geschieht mit Hilfe der berichtsspezifischen Steuerelementeigenschaft *Duplikate ausblenden*, die auf *Ja* oder *Nein* eingestellt werden kann.

Die Verwendung der Eigenschaft *Duplikate ausblenden* kann eine Alternative zu einem Bericht mit einem Gruppenkopf darstellen: Wenn Sie als Datenbasis beispielsweise eine Abfrage verwenden, die auf zwei Tabellen mit einer 1:n-Beziehung basiert und Felder aus beiden Tabellen wiedergibt, so würde die Aufnahme dieser Felder in den Detailbereich eines ungruppierten Berichts im Prinzip dazu führen, daß die Felder, die aus der Mastertabelle stammen, wiederholt ausgegeben würden, was störend wirken kann. Wenn Sie für diese Felder die Eigenschaft *Duplikate ausblenden* auf Ja einstellen, werden sie nur einmal für jede Gruppe von Werten der Detailtabelle, die auf sie referenziert, ausgegeben. Damit wird die Wirkung eines gruppierten Berichts erzielt.

Kapitel 24

Menüleisten, Symbolleisten und Kontextmenüs verwalten

24.1 Das Konzept

Gegenüber früheren Versionen ist das Konzept der Verwaltung von Menüleisten, Symbolleisten und Kontextmenüs mit Access 97 grundlegend verändert worden: Es wurde wesentlich vereinfacht und ist zugleich leistungsfähiger geworden.

Die früheren Symbolleisten konnten ausschließlich Schaltflächen enthalten. Seit Access 97 können Symbolleisten, Menüleisten und Kontextmenüs sowohl Schaltflächen wie auch Menüs enthalten. Im Prinzip sind diese daher in der Funktion austauschbar, und Sie können alle drei auf dieselbe Weise anpassen. Obwohl z.B. die eingebaute Menüleiste nach wie vor standardmäßig am oberen Rand des Bildschirms angezeigt wird und Standardmenüs wie *Suchen*, *Bearbeiten* und *Ansicht* enthält, können Sie diese durch Hinzufügen oder Entfernen von Schaltflächen und Menüs oder durch Verschieben der Leiste an eine andere Position anpassen. Darüber hinaus können Sie eigene Symbolleisten, Menüleisten und Kontextmenüs erstellen.

Trotz des geänderten Konzepts für Menüleisten sind auch die Menüleisten, die in früheren Versionen erstellt wurden, weiterhin funktionsfähig. Wenn Sie daher eine Datenbank mit derartigen Menüleisten nach Access 2000 konvertieren, werden auch deren benutzerdefinierte Menüleisten funktionieren.

Bis auf eine Ausnahme (die Ausnahme: Zuordnung zu einem Formular, Bericht oder Steuerelement) erfolgt die gesamte Verwaltung von Symbolleisten, Menüleisten und Kontextmenüs mit dem Dialogfeld *Anpassen*. Dieses öffnen Sie auf eine der beiden folgenden Weisen:

- Wählen Sie aus dem Menü *Ansicht* den Befehl *Symbolleisten* und dann dessen Unterbefehl *Anpassen*.

  Oder:

- Klicken Sie mit der rechten Maustaste auf einen Punkt einer Symbol- oder Menüleiste, um das Kontextmenü zu öffnen, und wählen Sie darin den Befehl *Anpassen*.

Im folgenden fasse ich mich sehr kurz und beschreibe nur die strukturell wichtigen Punkte zur Verwaltung von Menüleisten, Symbolleisten und Kontextmenüs, weil Sie die vielen Einzelheiten entweder leicht mit Hilfe der Schaltflächen im Dialogfeld *Anpassen* erkunden und probieren können, und weil andererseits die Online-Hilfe eine sehr ausführliche Erklärung bietet: Schlagen Sie ggf. das Thema *Arbeiten mit Befehls- und Menüleisten* und seine diversen Unterthemen auf.

24.2 Menüleisten, Symbolleisten und Kontextmenüs erstellen

24.2.1 Menü- oder Symbolleiste erstellen

Um eine neue Menü- oder Symbolleiste zu erstellen, verfahren Sie wie folgt:

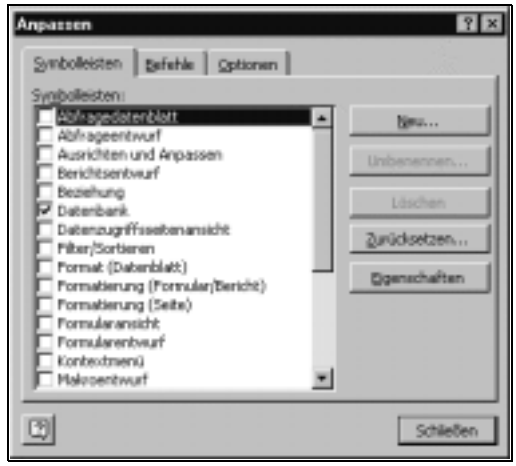


Bild 24.1: Dialogfeld »Anpassen«, Registerkarte »Symbolleisten«

- Öffnen Sie das Dialogfeld *Anpassen*, und aktivieren Sie darin ggf. die Registerkarte *Symbolleisten*, vgl. Bild 24.1.
- Klicken Sie auf die Schaltfläche *Neu*, und geben Sie in das Dialogfeld *Neue Symbolleiste* einen Namen für die neue Symbolleiste ein. Dann erscheint eine neue Symbolleiste als kleines Fenster, das Sie verschieben können. Die neue Symbolleiste ist zunächst leer, weil Sie ihr noch keine Schaltfläche und keinen Befehl eingefügt haben.

- Fügen Sie der Symbolleiste Befehle und Schaltflächen hinzu, indem Sie im Dialogfeld *Anpassen* die Registerkarte *Befehle* aktivieren, eine Kategorie markieren und den gewünschten Befehl in die Symbolleiste ziehen, vgl. Bild 24.3.

Bild 24.2: Dialogfeld »Symbolleisteneigenschaften«

- Um festzulegen, ob die neue Leiste das Aussehen einer Symbol- oder einer Menüleiste hat (funktionell besteht zwischen beiden kein Unterschied), klicken Sie im Dialogfeld *Anpassen* auf die Schaltfläche *Eigenschaften*. Dann zeigt sich das Dialogfeld *Symbolleisteneigenschaften*, vgl. Bild 24.2.
- Wählen Sie die neu eingefügte Leiste in der Liste *Ausgewählte Symbolleiste* aus, und legen Sie mit der Dropdown-Liste *Typ* fest, ob die Leiste eine Menü- oder Symbolleiste sein soll. Sie können den Typ später jederzeit wieder ändern.

24.2.2 Kontextmenü erstellen

- Erstellen Sie eine Symbolleiste, wie im vorangehenden Punkt beschrieben. Legen Sie dann den Typ dieser Symbolleiste in der Dropdown-Liste *Typ* des Dialogfeldes *Symbolleisten-Eigenschaften* (vgl. Bild 24.2) als *Popup* fest. Dann wird ein Meldungsdialogfeld eingeblendet, das Sie darauf hinweist, daß die in den Typ *Popup* umgewandelte Symbolleiste nunmehr ein Kontextmenü ist, das daher nicht ohne weiteres am Bildschirm angezeigt wird. Das Kontextmenü ist jetzt vielmehr Bestandteil einer (internen) Symbolleiste mit dem Namen *Kontextmenü*, die in der Liste *Symbolleisten* der Registerkarte *Symbolleisten* im Dialogfeld *Anpassen* angezeigt wird.
- Um dem Kontextmenü neue Schaltflächen oder Befehle hinzuzufügen, aktivieren Sie das Kontrollkästchen *Kontextmenü* in der Liste *Symbolleisten* der Registerkarte *Symbolleisten* im Dialogfeld *Anpassen*. Dann wird

eine Symbolleiste mit den Namen aller Kontextmenüs eingeblendet. Die letzte Position in dieser Leiste trägt den Namen *Anpassen*. Klicken Sie darauf, um eine Dropdown-Liste zu öffnen, die alle benutzerdefinierten Kontextmenüs enthält. Wählen Sie das zu bearbeitende, und fügen Sie diesem dann neue Schaltflächen oder Befehle hinzu.

24.3 Schaltflächen, Menüs und Befehle hinzufügen und löschen

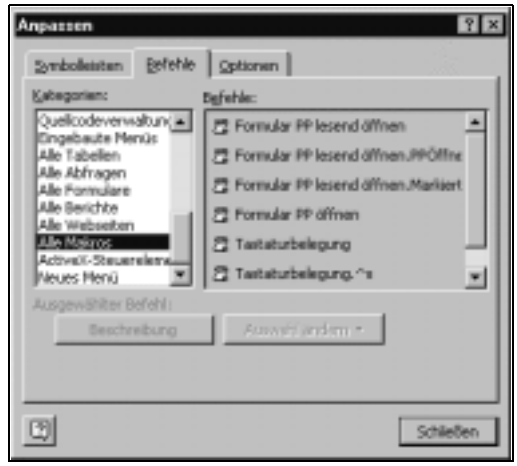


Bild 24.3: Dialogfeld »Anpassen«, Registerkarte »Befehle«

Sie können bestehenden Menü- und Symbolleisten und Kontextmenüs (integrierten wie benutzerdefinierten) neue Schaltflächen, Menüs und Befehle hinzufügen oder diese daraus entfernen. Um einen Befehl oder eine Schaltfläche hinzuzufügen, verfahren Sie wie folgt:

- Schlagen Sie das Dialogfeld *Anpassen* auf (vgl. oben, Punkt 24.1, *Das Konzept*).
- Aktivieren Sie in der Registerkarte *Symbolleisten* ggf. die Leiste, die Sie bearbeiten möchten, damit diese auf dem Bildschirm angezeigt wird. Im Falle eines Kontextmenüs aktivieren Sie das Kontrollkästchen *Kontextmenü*.
- Aktivieren Sie die Registerkarte *Befehle*, markieren Sie darin die zutreffende Kategorie, und ziehen Sie den gewünschten Befehl auf die Leiste oder in das Kontextmenü. Für benutzerdefinierte Kontextmenüs müssen Sie auf das Menü *Anpassen* in der Kontextmenüleiste ziehen und darin auf das betreffende Kontextmenü. Beachten Sie, daß Sie auch Makros als Befehle bzw. Schaltflächen einfügen können, wenn Sie die Kategorie *Alle Makros* markieren.

- Um einen Befehl oder eine Schaltfläche aus einer Leiste oder einem Kontextmenü zu entfernen, ziehen Sie den Befehl oder die Schaltfläche von der Leiste oder aus dem Kontextmenü auf eine Stelle außerhalb der Leiste oder des Kontextmenüs.

Sie können Symbol-Schaltflächen in einer Menü- oder Symbolleiste auch verschieben, ohne das Dialogfeld *Anpassen* zu öffnen: Halten Sie die Alt-Taste gedrückt, während Sie ein Symbol an eine andere Stelle ziehen. Auf diese Weise können Sie auch ein Symbol entfernen, indem Sie es von der Menü- oder Symbolleiste herunterziehen.

24.4 Eigenschaften von Menüleisten, Symbolleisten und Kontextmenüs einstellen

Für Menü- und Symbolleisten sowie Kontextmenüs können Sie verschiedene Eigenschaften einstellen. Beispielsweise können Sie festlegen, ob eine Leiste an einem Rand des Anwendungsfensters angedockt oder ob sie überhaupt verschoben werden kann. Dies geschieht in dem Dialogfeld *Symbolleisteneigenschaften*, vgl. oben Bild 24.2. Gehen Sie wie folgt vor, um eine Eigenschaft einzustellen:

- Aktivieren Sie die Registerkarte *Symbolleisten* des Dialogfeldes *Anpassen*, und klicken Sie auf die Schaltfläche *Eigenschaften*. Dann wird das Dialogfeld *Symbolleisteneigenschaften* angezeigt, vgl. oben Bild 24.2.
- Wählen Sie ggf. mit der Dropdown-Liste *Ausgewählte Symbolleiste* die Symbolleiste, deren Eigenschaften Sie einstellen wollen, und stellen Sie die Eigenschaften dann wie gewünscht ein.

24.5 Einbinden von Menüleisten, Symbolleisten und Kontextmenüs in Formulare, Berichte oder Steuerelemente

Sie können eine Menü- oder Symbolleiste in ein Formular oder einen Bericht einbinden. Sobald eine benutzerdefinierte Symbolleiste eingebunden ist, ersetzt diese die eingebaute Symbolleiste für das Formular oder den Bericht. Verfahren Sie dazu wie folgt:

- Öffnen Sie das Formular oder den Bericht in der Entwurfsansicht, und geben Sie diesem den Fokus.
- Zeigen Sie ggf. das Eigenschaftenfenster an, und stellen Sie die Eigenschaft *Menüleiste* bzw. *Symbolleiste* auf den Namen der betreffenden Leiste ein.
- Verfahren Sie für ein Kontextmenü entsprechend, indem Sie die Eigenschaft *Kontextmenüleiste* auf den Namen eines Kontextmenüs einstellen. Diese Eigenschaft besitzen auch Steuerelemente.

24.6 Bild oder Text von Schaltflächen und Menübefehlen bearbeiten

Sie können das Aussehen von Befehlen und Schaltflächen in Menü- und Symbolleisten sowie Kontextmenüs auf einfache Art ändern, indem Sie andere Bilder darauf kopieren, das bestehende Bild bearbeiten, Texte ändern usw. Verfahren Sie prinzipiell folgendermaßen:

- Sorgen Sie dafür, daß das oder die zu bearbeitende(n) Element(e) auf dem Bildschirm sichtbar sind.
- Zeigen Sie das Dialogfeld *Anpassen* an (vgl. oben Punkt 24.1, *Das Konzept*).
- Klicken Sie mit der rechten Maustaste auf den Befehl oder die Schaltfläche, die Sie bearbeiten wollen, um das Kontextmenü zu öffnen, vgl. Bild 24.4.
- Wählen Sie den geeigneten Befehl aus dem Kontextmenü.

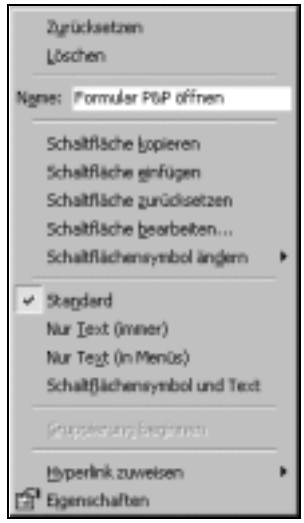


Bild 24.4: Kontextmenü eines Schaltflächensymbols bzw. Befehls

Der Befehl *Schaltfläche einfügen* arbeitet nur dann sinnvoll, wenn sich ein einfügbares Bild in der Zwischenablage befindet. Um diesen Befehl anzuwenden, müßten Sie ggf. zuvor den Befehl *Schaltfläche kopieren* oder eine andere Aktion, die ein Bild in die Zwischenablage kopiert, ausführen. Mit dem Befehl *Schaltfläche bearbeiten* öffnen Sie den *Schaltflächen-Editor*, vgl. Bild 24.5, mit dem Sie jedes Schaltflächen-Bild Punkt für Punkt bearbeiten können.

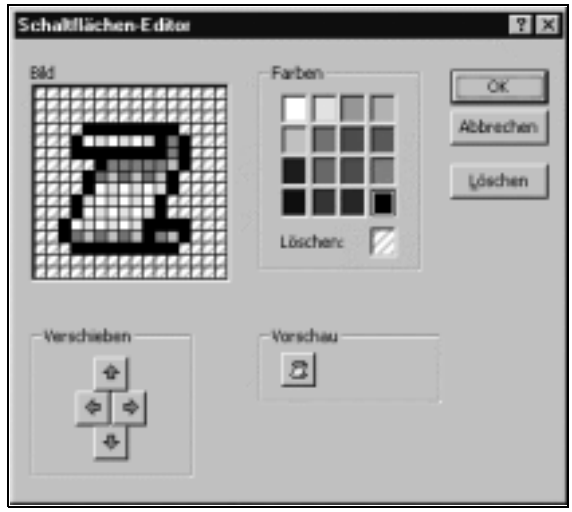


Bild 24.5: Dialogfeld »Schaltflächen-Editor«

Kapitel 25

Access benutzerdefiniert anpassen

In diesem Kapitel geht es darum, wie Sie gewisse Voreinstellungen von Access für eine bestimmte Datenbank oder generell für das installierte Access-System ändern können. Dabei stehen Ihnen prinzipiell vier Möglichkeiten zur Verfügung:

- *Makro AutoExec*. Ein Makro, das diesen besonderen Namen trägt, wird beim Öffnen der Datenbank, zu der es gehört, ausgeführt. Auf diese Weise können Sie im Grunde jede gewünschte Anfangseinstellung realisieren, weil das Makro nicht nur alle Access-Aktionen ausführen, sondern mit der speziellen Makroaktion *AusführenCode* auch VBA-Prozeduren aufrufen kann, so daß Sie über diese Brücke beliebigen VBA-Code beim Start einer Datenbank zur Ausführung bringen können. Auf die Möglichkeiten von *AutoExec*-Makros soll an dieser Stelle nicht näher eingegangen werden, weil diese im einzelnen in Kap. 26, *Makros: Grundlagen*, Punkt 26.6, *Autoexec-Makro*, behandelt werden.
- *Startoptionen für eine bestimmte Datenbank im Dialogfeld* Start. In diesem Dialogfeld, das Sie mit dem gleichnamigen Befehl aus dem Menü *Extras* öffnen, können Sie 12 verschiedene Optionen einstellen, die unmittelbar nach dem Öffnen der Datenbank, für die diese Optionen eingestellt wurden, wirksam werden. Diese Optionen sind Datenbank-spezifisch, d.h., sie wirken sich nur auf die eine Datenbank aus. Wenn Sie eine andere Datenbank öffnen, kommen wieder die Voreinstellungen oder die für diese andere Datenbank festgelegten Startoptionen zur Wirkung. Die Einstellungen der Optionen aus dem Dialogfeld *Start* werden zusammen mit der Datenbank (und nicht für das Access-System) gespeichert, so daß diese auch dann wirksam werden, wenn Sie die Datenbank unter einem anderen installierten Access-System (z.B. auf einem anderen PC) öffnen.

- *Optionen für das installierte Access-System im Dialogfeld* Optionen. In den acht Registerkarten des Dialogfeldes *Optionen*, das Sie mit dem gleichnamigen Befehl aus dem Menü *Extras* öffnen, können Sie insgesamt zahlreiche Einstellungen vornehmen, die nicht nur für die aktuelle Datenbank, sondern für jede im weiteren geöffnete wirken, weil sie für das installierte Access-System gelten. Diese Einstellungen werden nicht in einer Datenbank, sondern in der Windows-Registrierung gespeichert.
- *Starten von Access mit Befehlszeilenoptionen*. Sie können Access starten und dabei bestimmte Parameter, z.B. den Namen einer zu öffnenden Datenbank oder den Namen einer auszuführenden VBA-Prozedur, übergeben.

25.1 Startoptionen für eine bestimmte Datenbank

Die im folgenden wiedergegebenen Optionen des Dialogfeldes *Start* können im Konflikt mit Aktionen stehen, die durch ein *AutoExec*-Makro ausgeführt werden. Beachten Sie, daß zunächst die Optionen des Dialogfeldes *Start* und erst danach die Aktionen des *AutoExec*-Makro ausgeführt werden.

25.1.1 Optionen einstellen

Sie legen die Startoptionen im Dialogfeld *Start* des Befehls *Start* aus dem Menü *Extras fest. Die Optionen bedeuten im einzelnen:*

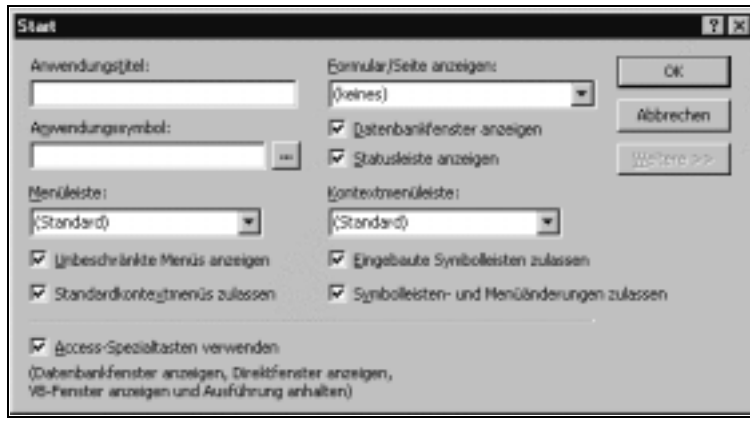


Bild 25.1: Dialogfeld »Start« des gleichnamigen Befehls aus dem Menü »Extras«. Die wiedergegebenen Einstellungen entsprechen den Voreinstellungen, wie diese für eine neu erstellte Datenbank wirksam sind.

Anwendungstitel

Geben Sie einen Text ein, der dann statt des voreingestellten Textes Microsoft Access in der Titelleiste der Anwendung erscheint.

Anwendungssymbol

Geben Sie Namen und Pfad einer *.ico*- oder *.bmp*-Datei ein, die Sie verwenden möchten. Dann erscheint das entsprechende Symbol am linken Rand der Titelleiste der Anwendung. Wenn Sie den Namen der Datei nicht kennen, klicken Sie neben dem Feld auf die Schaltfläche mit den drei Punkten und verwenden dann den Datei-Browser, um die Datei zu suchen.

Formular/Seite anzeigen

Wählen Sie in der Dropdown-Liste *Formular/Seite anzeigen* ein Formular aus der aktuellen Datenbank. Dann wird dieses Formular künftig nach dem Öffnen der Datenbank geöffnet.

Datenbankfenster anzeigen

Wenn dieses Kontrollkästchen deaktiviert ist, wird das Datenbankfenster nach dem Öffnen der Datenbank ausgeblendet. Es kann allerdings mit dem Befehl *Einblenden* aus dem Menü *Fenster* wieder eingeblendet werden – es sei denn, Sie verhindern das Anzeigen dieses Befehls, vgl. unten.

Statusleiste anzeigen

Deaktivieren Sie dieses Kontrollkästchen, um zu verhindern, daß für diese Datenbank die Statusleiste angezeigt wird.

Menüleiste

Wählen Sie in der Dropdown-Liste *Menüleiste* den Namen einer benutzerdefinierten Menüleiste aus der aktuellen Datenbank. Dann wird diese Menüleiste die globale Menüleiste der Datenbank, die die eingebaute Menüleiste für alle Access-Fenster ersetzt. Ausgenommen sind jedoch die Fälle, in denen Sie für ein Formular oder einen Bericht eine benutzerdefinierte Menüleiste erstellt haben, die Vorrang gegenüber dieser Option hat.

Kontextmenüleiste

Verfahren Sie entsprechend wie im vorangehenden Punkt für die Menüleiste beschrieben.

Unbeschränkte Menüs anzeigen

Wenn Sie dieses Kontrollkästchen deaktivieren, werden nur diejenigen eingebauten Menüs und darin diejenigen Befehle angezeigt, die unmittelbar zum Bearbeiten des gerade aktiven Objekts erforderlich sind. Ausgeblendet ist dann beispielsweise das Menü *Ansicht*, so daß nicht in die Entwurfsansicht eines Objekts gewechselt werden kann.

Standardkontextmenüs zulassen

Das Deaktivieren dieses Kontrollkästchens wirkt entsprechend, wie im Punkt *Unbeschränkte Menüs anzeigen* beschrieben.

Eingebaute Symbolleisten zulassen

Das Deaktivieren dieses Kontrollkästchens wirkt entsprechend, wie im Punkt *Unbeschränkte Menüs anzeigen* beschrieben.

Symbolleisten- und Menüänderungen zulassen

Sie können festlegen, ob der Benutzer in der Lage sein soll, eingebaute oder benutzerdefinierte Symbol- und Menüleisten zu ändern. Durch Deaktivieren des Kontrollkästchens *Symbolleisten- und Menüänderungen zulassen* nehmen Sie dem Benutzer diese Möglichkeit.

Wenn Sie diese Option deaktivieren, können Sie Symbolleisten und Menüs trotzdem noch verschieben, vergrößern oder verkleinern und andocken, vorausgesetzt, Sie haben nicht zuvor im Dialogfeld *Anpassen* für eine bestimmte Symbol- oder Menüleiste etwas anderes festgelegt. Wenn diese Option deaktiviert ist, steht das Dialogfeld *Anpassen* nicht mehr zur Verfügung. Der Benutzer ist nicht mehr in der Lage, Symbol- oder Menüleisten anzupassen, auch wenn für eine bestimmte Symbol- oder Menüleiste vorher im Dialogfeld *Symbolleisten-Eigenschaften* das Kontrollkästchen *Anpassen zulassen* aktiviert worden war.

Access-Spezialtasten verwenden

Mit dieser Option aktivieren oder deaktivieren Sie Tasten, die das Datenbank-, Direkt- oder das VB-Fenster anzeigen oder die Ausführung von Prozeduren anhalten.

Betroffen sind die folgenden Tasten (-kombinationen):

Tasten	Ergebnis
F11 oder Alt+F1	Holt das Datenbankfenster in den Vordergrund.
Strg+G	Zeigt das Direktfenster an.
Strg+F11	Wechselt zwischen der benutzerdefinierten und der eingebauten Menüleiste hin und her.
Strg+Pause	Beendet die Codeausführung, und zeigt das aktuelle Modul im Modulfenster an.
Alt+F11	Öffnet das VB-Fenster.

25.1.2 Wirksamkeit der Optionen umgehen

Um die im Dialogfeld *Start* für eine Datenbank festgelegten Optionen zu umgehen, müssen Sie die Datenbank öffnen und dabei die ⇧-Taste gedrückt halten. Damit verhindern Sie, daß die Optionen wirksam werden.

25.2 Optionen für Access

Optionen, die sich für alle Sitzungen des installierten Access-Systems auswirken, werden im Dialogfeld *Optionen* des gleichnamigen Befehls aus dem Menü *Extras* eingestellt, vgl. Bild 25.7. Die Registerkarten dieses Dialogfeldes bieten insgesamt so zahlreiche Einstellmöglichkeiten, daß diese hier aus räumlichen Gründen nicht im einzelnen behandelt werden können. In mehreren vorangehenden Kapiteln ist im jeweiligen Zusammenhang bereits auf verschiedene Optionen dieses Dialogfeldes eingegangen worden. Machen Sie im Zweifel von der Möglichkeit Gebrauch, Kontexthilfe anzufordern: Klikken Sie bei geöffnetem Dialogfeld auf das Fragezeichen in der Titelleiste des Dialogfelds, und klicken Sie dann auf die Option, zu der Sie Hilfe benötigen.

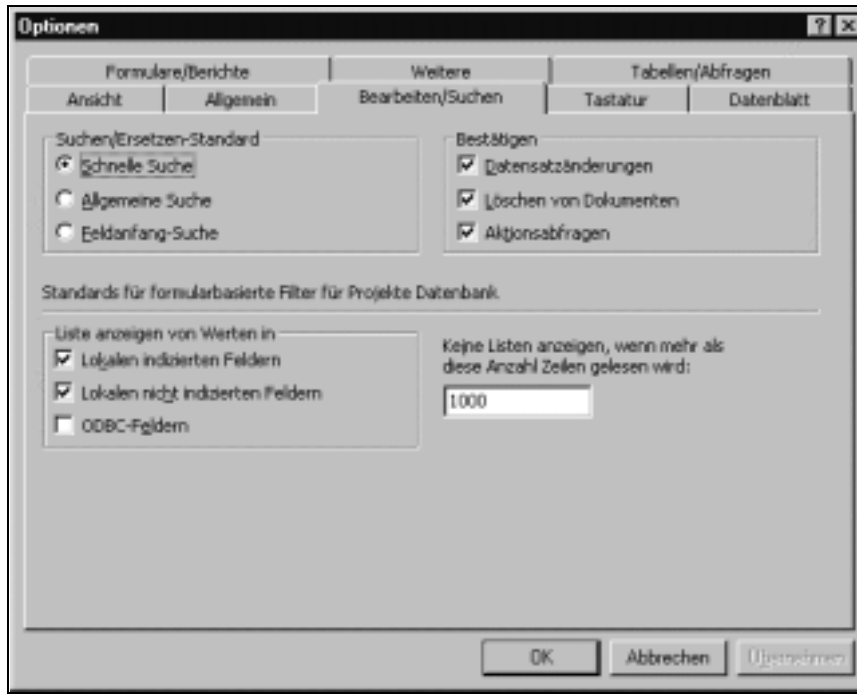


Bild 25.2: Dialogfeld »Optionen«

25.3 Starten von Access mit Befehlszeilenoptionen

Sie können mit dem Start von Access beispielsweise automatisch eine Datenbank öffnen, ein Makro ausführen, den Namen eines Benutzerkontos oder ein Kennwort eingeben und andere Optionen ausführen lassen, wenn Sie beim Starten von Access in der Befehlszeile entsprechende Optionen eingeben. Diese Optionen werden *Befehlszeilenoptionen* genannt. Sie können Befehlszeilenoptionen für Access entweder im Windows-Menü *Start*, Befehl *Ausführen*, oder für eine Verknüpfung angeben.

In der folgenden Tabelle sind die verfügbaren Befehlszeilenoptionen aufgelistet.

Option	Funktion
Datenbankname, ggf. mit Pfadangabe	Öffnet die angegebene Datenbank.
/excl	Öffnet die angegebene Datenbank für exklusiven Zugriff.
/ro	Öffnet die angegebene Datenbank für schreibgeschützten Zugriff.
/user Benutzername	Startet Access unter Verwendung des angegebenen Benutzernamens.
/pwd Kennwort	Startet Access unter Verwendung des angegebenen Kennworts.
/profile Benutzerprofil	Startet Access unter Verwendung der Optionen im angegebenen Benutzerprofil anstelle der Standardregistrierungseinstellungen, die beim Installieren von Access festgelegt wurden.
/compact Zieldatenbank	Komprimiert die vor der Option */compact* angegebene Datenbank und schließt Access. Wenn Sie nach der Option */compact* keine Zieldatenbank eingeben, wird die Originaldatenbank mit der komprimierten Datenbank überschrieben. Um der komprimierten Datenbank einen anderen Namen zu geben, geben Sie eine Zieldatenbank an. Wenn Sie in Zieldatenbank keinen Pfad angeben, wird die Datenbank standardmäßig in Ihrem Ordner Eigene Dateien erstellt.
/repair	Repariert die vor der Option */repair* angegebene Datenbank und schließt Access.
/convert Zieldatenbank	Konvertiert eine Datenbank der Version 1.x, 2.0 oder 7.0 in eine Access 97-Datenbank mit einem neuen Namen und schließt Access. Geben Sie vor der Option */convert* die Quelldatenbank an.

Option	Funktion
/x Makro	Startet Access und führt das angegebene Makro aus. Sie können ein Makro beim Öffnen einer Datenbank auch unter Verwendung eines *AutoExec*-Makros ausführen.
/cmd	Gibt an, daß der in der Befehlszeile folgende Wert der Wert ist, der von der Funktion *Befehl* (*Command*) zurückgegeben wird. Diese Option muß die letzte Option in der Befehlszeile sein. Sie können anstelle von */cmd* auch ein Semikolon (;) verwenden.
/nostartup	Startet Access, ohne das Start-Dialogfeld anzuzeigen (das zweite Dialogfeld, das beim Starten von Access angezeigt wird).
/wrkgrp Datei	Startet Access unter Verwendung der angegebenen Arbeitsgruppen-Informationsdatei.

Anmerkung. Um in der Befehlszeile einen Schrägstrich (/) oder ein Semikolon (;) anzugeben, geben Sie das Zeichen doppelt ein. Geben Sie zum Beispiel für das Kennwort *;ich/du* nach der Befehlszeilenoption */pwd* folgendes ein: *;;ich//du*.

Beispiel

Wenn Sie Access mit der Befehlszeile

```
msaccess.exe \O9\Office\Samples\Nordwind.mdb /nostartup/excl
```

starten, wird die Datenbank *Nordwind.mdb* aus dem angegebenen Ordner exklusiv geöffnet, ohne daß das Start-Dialogfeld angezeigt wird.

Kapitel 26

Makros: Grundlagen

26.1 Einführung und Überblick

26.1.1 Was sind Makros?

Makros ermöglichen Ihnen, eine oder mehrere Aktionen, die Sie vorher in tabellarischer Form zusammengestellt haben, automatisch ausführen zu lassen. Beispielsweise können Sie in ein Makro zwei Aktionen schreiben, von denen die erste den aktuellen Datensatz markiert und die zweite den markierten Datensatz ausdruckt (vgl. Bild 26.1, das ein derartiges Makro wiedergibt). Wenn dann das Makro ausgeführt wird – z. B. durch Drücken einer Taste, Wahl eines benutzerdefinierten Menübefehls oder Klicken auf eine Schaltfläche im Formular –, wird der aktuelle Datensatz zunächst markiert und danach gedruckt. Statt zwei Aktionen könnte ein Makro auch nur eine Aktion oder, umgekehrt, praktisch unbegrenzt viele Aktionen enthalten, die dann jeweils beim Aufrufen des Makros nacheinander ausgeführt werden.

Bild 26.1: Dieses Makro besteht aus zwei Aktionen: Die erste Aktion »AusführenBefehl« markiert den aktuellen Datensatz, die zweite Aktion »Drucken« druckt den markierten Datensatz.

Aktionen

Alle in Makros ausführbaren Aufgaben werden *Aktionen* genannt. Die meisten Aktionen entsprechen bestimmten Menü- oder Symbolbefehlen. So entspricht die Aktion *ÖffnenFormular* dem Vorgang, den Sie vom Datenbankfenster aus durch Klicken auf die Schaltfläche *Öffnen* bzw. *Entwurf* bei markiertem Formularnamen auslösen. Darüber hinaus verfügen Sie in Makros über eine Reihe von Aktionen, die manuell nicht ausführbar sind. Dazu zählt beispielsweise die Aktion *SetzenWert*, mit der Sie beispielsweise in Formularen Steuerelementen im Hintergrund automatisch Werte zuweisen können, oder die Aktion *Meldung*, die ein kleines Dialogfeld mit einer benutzerdefinierten Meldung auf dem Bildschirm anzeigt und den Fluß der Makroaktionen so lange unterbricht, bis die Meldung bestätigt wird.

Argumente

Oft können Sie allerdings mit einer Makroaktion differenziertere Abläufe auslösen, als dies mit den entsprechenden manuellen Befehlen möglich wäre. Dies liegt vor allem daran, daß Sie für die meisten Makroaktionen Argumente angeben können bzw. müssen, welche die Aktion im Detail steuern. Für die gerade angeführte Aktion *ÖffnenFormular* beispielsweise ist es möglich, das Argument *Fenstermodus* auf den Wert *Dialog* einzustellen; dann wird das Formular als Dialogfeld geöffnet mit der Folge, daß u.a. der Makrofluß so lange unterbrochen wird, bis das Formular geschlossen oder ausgeblendet wird.

Bedingungen

Ein weiterer Unterschied von Makroaktionen gegenüber normalen Befehlen liegt darin, daß Sie die Ausführung einer Makroaktion vom Zutreffen einer Bedingung abhängig machen können. Beispielsweise können Sie das Abblenden oder Deaktivieren eines bestimmten Steuerelements vom Wert eines anderen Steuerelements abhängig machen oder, um ein anderes Beispiel zu nennen, die Ausführung einer ganzen Gruppe von Makroaktionen von einer Bedingung abhängig machen.

Objekt *Makro und Makronamen*

Sie können ein fertiges Makro nur ausführen, nachdem Sie es zuvor gespeichert haben. Dies gilt auch für jede Änderung an einem bestehenden Makro. Sie speichern ein Makro genauso wie eine Tabelle, ein Formular etc., also z.B. mit der Tastenkombination Strg+S oder mit dem Befehl *Speichern* aus dem Menü *Datei* oder nach Aufforderung durch Access (dies geschieht beispielsweise beim Versuch, ein noch nicht gespeichertes Makro ausführen zu lassen). Gespeicherte Makros werden im Datenbankfenster in der Objektliste *Makros* mit ihren Namen angezeigt, auch dies ganz analog zu Tabellen, Abfragen etc., vgl. Bild 26.2.

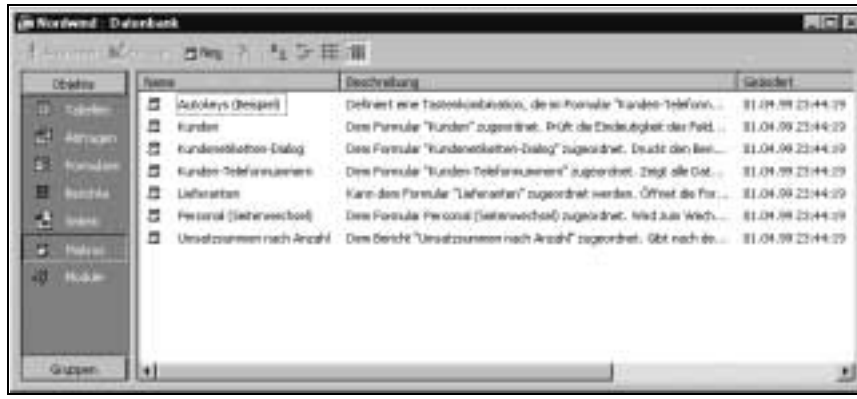


Bild 26.2: Makros im Datenbankfenster der Datenbank Nordwind.mdb, hier in der Fensteransicht »Details«, die Sie durch Klicken auf das gleichnamige Symbol einstellen können.

Sie führen ein gespeichertes Makro aus, indem Sie seinen Namen an geeigneter Stelle anführen (z.B. in einer ereignisorientierten Eigenschaft eines Formulars, beispielsweise *Beim Öffnen*). Eine andere Möglichkeit, ein Makro aufzurufen, haben Sie im Datenbankfenster: Wenn Sie dort auf die Schaltfläche *Ausführen* klicken, wird das Makro, welches gerade markiert ist, gestartet. Dann werden alle Aktionen, die das Makro enthält, in der angeführten Reihenfolge ausgeführt. Wenn Sie viele unabhängige Makros benötigen, kann diese Organisation lästig sein, weil Sie dann für jede auch noch so kleine Aktion ein eigenes Makro erstellen müssen. Aus diesem Grunde sieht Access auch die Möglichkeit vor, Gruppen von Aktionen in einem Makro einen Namen zuzuweisen, damit diese Gruppe dann als eigenständiges Makro ausgeführt werden kann, ohne daß die anderen Aktionen desselben Makrofensters ausgeführt werden.

26.1.2 Für welche Zwecke lassen sich Makros einsetzen?

Die Einsatzzwecke von Makros sind nahezu unbegrenzt. Dennoch lassen sich einige Gebiete nennen, für die sie besonders nützlich sind:

- *Befehlsschaltflächen und ähnliche Auslöser in Formularen*. Sie können in einem Formular Befehlsschaltflächen vorsehen, die, wenn darauf geklickt wird, ein Makro mit mehr oder minder vielen Aktionen starten. Beispielsweise könnten Sie eine Befehlsschaltfläche dazu verwenden, einen ganz bestimmten Filter für das Formular zu setzen. Für viele Steuerelemente besteht daneben die Möglichkeit, durch Doppelklicken darauf ein Makro ausführen zu lassen.
- *Ereignisabhängiges Ausführen eines Makros*. Die Steuerelemente in Formularen sowie das Formular selbst besitzen Eigenschaften wie z.B. *Vor Aktualisierung* (Steuerelement) oder *Beim Öffnen* (Formular), denen Sie ein Makro zuordnen können. Wenn das Ereignis eintritt, wird das Makro

gestartet. So können Sie beispielsweise der Textfeldeigenschaft *Vor Aktualisierung* ein Makro zuordnen, das die Gültigkeit eines eingegebenen Wertes prüft und die Aktualisierung abbricht, wenn die Gültigkeit verletzt ist. Mit Makros können Sie leistungsfähigere Gültigkeitsregeln aufstellen als mit der Steuerelementeigenschaft *Gültigkeitsregel*. Insbesondere können Sie sich dabei auf andere Steuerelemente oder Felder beziehen und so die Gültigkeit des eingegebenen Wertes vom aktuellen Wert eines anderen Feldes oder Steuerelementes abhängig machen. Das Ereignismodell von Access wird ausführlich in Kap. 27, *Ereignisse*, behandelt.

- *Steuerelemente in Formularen mit Werten versorgen*. Tabellen kennen keine berechneten Felder. Andererseits kommt es oft vor, daß berechnete Werte in der Tabelle gespeichert werden sollen. Ein klassisches Problem ist die Mehrwertsteuer in Rechnungen: Sie können diese ohne weiteres im Formular in einem berechneten Steuerelement ausweisen (z.B. durch den Ausdruck =*Nettobetrag*0,16*). Allerdings kann ein berechnetes Steuerelement nicht gleichzeitig gebunden sein, so daß sein (berechneter) Wert nicht automatisch in einem Tabellenfeld gespeichert werden kann. Zu diesem Zweck benutzen Sie die Aktion *SetzenWert* in einem Makro, um einem an ein Tabellenfeld (z.B. das Feld *Mehrwertsteuerbetrag*) gebundenen Steuerelement den Wert zuzuweisen, der vom berechneten Steuerelement (z.B. dem Steuerelement *Mehrwertsteuer*) ermittelt wird.
- *Differenzierte Gültigkeitsregeln*. Nehmen Sie an, die Gebühr, die Sie für eine erbrachte Leistung in Rechnung stellen und dafür in das Formular eingeben wollen, sei in der Höhe von einer Gebührenstaffel abhängig, deren Ergebnisse wiederum von gewissen individuellen Merkmalen des Auftrags abhängig sind. Um die zulässige Gebühr zu ermitteln, ist die Eigenschaft *Gültigkeitsregel* von Steuerelementen im allgemeinen überfordert, weil dafür eine mehrstufige Berechnung durchgeführt werden muß. Mit einem Makro, das Sie an die Eigenschaft *Vor Aktualisierung* binden, können Sie sehr komplexe Gültigkeitsregeln aufstellen und prüfen.
- *Autoexec-Makro*. Wenn Sie einem Makro den speziellen Namen *Autoexec* geben, wird es beim Öffnen der Datenbank automatisch ausgeführt. Auf diese Weise können Sie bestimmte Standardeinstellungen und eine bestimmte Benutzeroberfläche (z.B. Öffnen eines bestimmten Formulars) automatisch erzeugen. (Hinweis: Verschiedene Startoptionen können Sie auch einstellen, ohne ein Makro verwenden zu müssen, vgl. den Befehl *Start* aus dem Menü *Extras*.)
- *Benutzerdefinierte Tastenbelegungen*. Jedem Makro kann eine Taste oder Tastenkombination zugeordnet werden. Dann wird das Makro ausgeführt, wenn die betreffende Taste oder Tastenkombination gedrückt wird. Da Sie mit Makros praktisch jeden standardmäßigen Menübefehl ausführen lassen können, ist es möglich, für jeden Menübefehl eine eigene Tastenkombination zu vereinbaren, benutzerdefinierte eingeschlossen.

26.1.3 Das Makrofenster

Makros werden, ganz analog zu Tabellen, Abfragen, Formularen und Berichten, in einem eigenen Fenster erstellt und bearbeitet. Sie öffnen das Makrofenster, wenn Sie ein neues Makro erstellen oder ein bestehendes bearbeiten, vgl. den folgenden Abschnitt.

Bild 26.3: Makrofenster unmittelbar nach dem Anfordern eines neuen Makros

Makrofenster in Standardansicht

Bild 26.3 zeigt ein Makrofenster, wie es unmittelbar nach dem Anfordern eines neuen Makros erscheint. Das Makrofenster besteht aus zwei Teilen: In den hellen oberen, tabellarisch aufgebauten Teil werden die Aktionen, ggf. mit Kommentaren, weiteren Namen und Bedingungen, geschrieben. Im unteren Teil mit der Überschrift *Aktionsargumente* werden die Argumente für die einzelnen Aktionen angegeben. Dieser Teil arbeitet kontextabhängig: Je nach der Aktion, die gerade den Fokus hat, werden unterschiedliche Argumente und, rechts daneben, erklärende Hilfetexte angeboten. In Bild 26.3 beispielsweise wird gar kein Argument angeboten, weil noch keine Aktion eingegeben bzw. ausgewählt wurde.

Im Dialogfeld *Optionen* des gleichnamigen Befehls aus dem Menü *Extras* können Sie in der Registerkarte *Ansicht* angeben, ob für ein neues Makrofenster die Namens- und die Bedingungsspalte angezeigt werden soll. Für das in Bild 26.3 dargestellte Fenster sind diese beiden Spalten ausgeblendet, weil die genannten Startoptionen nicht gewählt waren. Sie können indes jederzeit manuell aus- und eingeblendet werden, vgl. die folgenden Punkte.

Symbolleiste des Makrofensters

Die Symbolleiste des Makrofensters weist die folgenden sechs makrospezifischen Symbol-Schaltflächen auf:

Symbol-Schaltfläche *Makronamen*. Blendet die Spalte für Makronamen ein oder aus.

Symbol-Schaltfläche *Bedingungen*. Blendet die Spalte für Bedingungen ein oder aus.

Symbol-Schaltfläche *Ausführen*. Führt das Makro im Ganzen aus.

Symbol-Schaltfläche *Einzelschritt*. Führt das Makro schrittweise aus.

Symbol-Schaltfläche *Zeilen einfügen*. Fügt vor der(n) markierten Zeile(n) eine oder mehrere neue Zeilen ein.

Symbol-Schaltfläche *Zeilen löschen*. Löscht die markierte(n) Zeile(n).

Zusätzliche Spalten im Makrofenster

Standardmäßig zeigt das Makrofenster nur die beiden Spalten *Aktion* und *Kommentar*, vgl. oben Bild 26.3 – es sei denn, Sie haben in der Registerkarte *Ansicht* des Dialogfeldes *Optionen* (gleichnamiger Befehl im Menü *Extras*) etwas anderes veranlaßt. Darüber hinaus können Sie mit Hilfe der entsprechenden Symbol-Schaltflächen die beiden weiteren Spalten *Makroname* und *Bedingung* einblenden (und ausblenden) lassen, vgl. Bild 26.4. Die Spalte *Makroname* benötigen Sie, wenn Sie einer Gruppe von Aktionen einen Namen zuweisen wollen, um diese Gruppe als eigenständiges Makro identifizieren und ausführen zu können. Die Spalte *Bedingung* brauchen Sie, um für eine oder mehrere Aktionen Bedingungen anzugeben, von deren Zutreffen das Ausführen der Aktion abhängig gemacht wird.

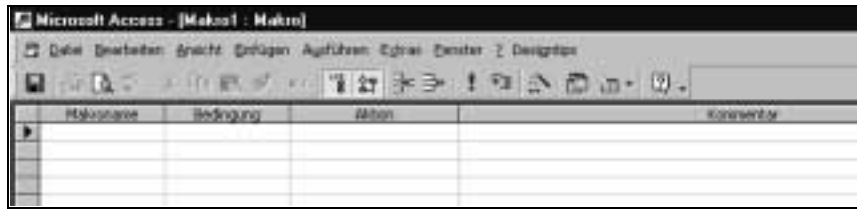


Bild 26.4: Makrofenster mit allen vier Spalten eingeblendet. Es ist zu erkennen, daß in der Symbolleiste die beiden Symbol-Schaltflächen »Makronamen« und »Bedingungen« gedrückt sind.

26.2 Makro erstellen

26.2.1 Einfaches Makro erstellen

Um ein neues Makro zu erstellen, haben Sie zwei Möglichkeiten. Die eine geht vom Datenbankfenster aus, die andere vom Eigenschaftenfenster eines Formulars oder Berichts. Der erste der beiden Wege ist der allgemeinste. Er empfiehlt sich, wenn Sie ein Makro unabhängig von einem Ereignis erstellen wollen, was nicht ausschließt, daß Sie das Makro später einem Ereignis zuordnen. Der zweite Weg bietet sich an, wenn Sie ein Makro gezielt für ein bestimmtes Ereignis eines bestimmten Steuerelements erstellen wollen. Im folgenden werden beide Möglichkeiten kurz skizziert.

Vom Datenbankfenster aus ein neues Makro erstellen

Gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit der Taste F11.
- Aktivieren Sie im Datenbankfenster die Objektliste *Makros*.
- Klicken Sie auf die Schaltfläche *Neu*. Access öffnet ein Makrofenster, wie Sie es oben in Bild 26.3 bzw. – je nach Voreinstellung – in Bild 26.4 sehen können.
- Geben Sie Aktionen und ggf. Aktionsargumente ein.
- Speichern Sie das Makro: Wählen Sie den Befehl *Speichern* aus dem Menü *Datei*, klicken Sie auf die Symbol-Schaltfläche *Speichern*, oder drücken Sie Strg+S.

Vom Eigenschaftenfenster aus ein neues Makro erstellen

Im folgenden wird unterstellt, daß Sie für das Ereignis *Vor Aktualisierung* des Textfeldes *Einstellungsdatum* im Formular *Formular1* ein Makro erstellen wollen. Dieses Makro wird dann als Ereignismakro fungieren. Die in den folgenden Aufzählungspunkten geschilderte Vorgehensweise gilt völlig analog für jede andere Ereigniseigenschaft, sei es für ein Steuerelement, das Formular oder den Bericht.

Verfahren Sie wie folgt:

- Markieren Sie das Steuerelement, hier also das Textfeld *Einstellungsdatum*.
- Blenden Sie ggf. das Eigenschaftenfenster ein, und klicken Sie auf das Einstellungsfeld der Eigenschaft *Vor Aktualisierung*, damit diese den Fokus bekommt.
- Klicken Sie auf die Symbol-Schaltfläche *Generator* am rechten Rand der Eigenschaft mit dem Fokus, vgl. Bild 26.5. Dann öffnet sich das Dialogfeld *Generator auswählen*, vgl. Bild 26.6.

Bild 26.5: Das Textfeld »Einstellungsdatum« ist markiert, und seine Ereigniseigenschaft »Vor Aktualisierung« hat den Fokus. Daher ist an deren rechtem Rand die Symbol-Schaltfläche »Generator« eingeblendet.

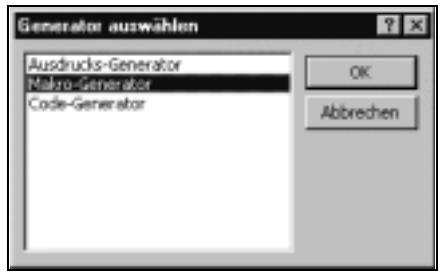


Bild 26.6: Dialogfeld »Generator auswählen«

- Markieren Sie im Dialogfeld *Generator auswählen* den Eintrag *Makro-Generator*, und bestätigen Sie mit OK.

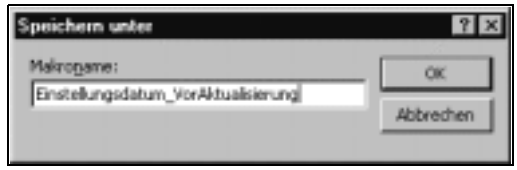


Bild 26.7: Dialogfeld »Speichern unter« des Makro-Generators

- Der Generator fordert Sie mit dem Dialogfeld *Speichern unter* auf, das neue Makro gleich unter einem Namen zu speichern, vgl. Bild 26.7. Angeboten wird der Makroname *Makro1* (bzw. *Makro2* etc.). Im allgemeinen

sollten Sie einen sprechenden Namen wählen, der auf den Inhalt und Zweck des Makros schließen läßt. Im vorliegenden Beispiel könnte dies der Name *Einstellungsdatum_VorAktualisierung* sein. Bestätigen Sie mit OK.

Falls Sie die Schaltfläche *Abbrechen* wählen, wird nur das Speichern abgebrochen. Sie können dann trotzdem in ein leeres Makrofenster die gewünschten Makroaktionen eingeben. Spätestens allerdings, wenn das neue Makro ausgeführt werden soll, müssen Sie es speichern.

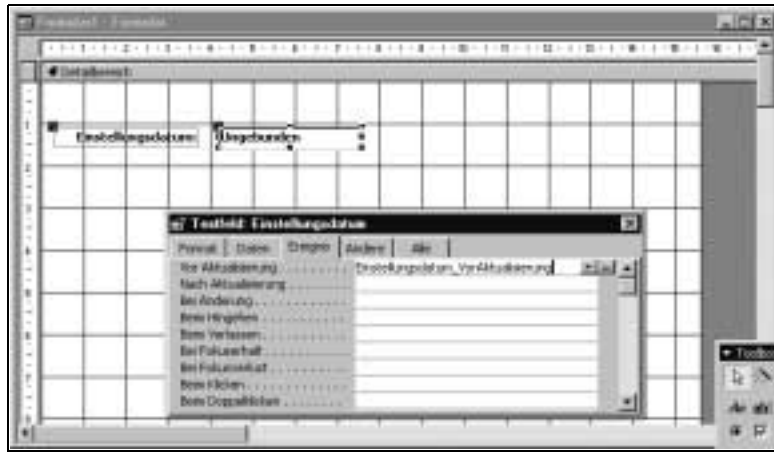


Bild 26.8: Die Eigenschaft »Vor Aktualisierung« ist automatisch auf den Namen eingestellt worden, unter dem das Makro gespeichert wurde.

- Wenn Sie im vorigen Punkt das neue Makro unter dem Namen *Einstellungsdatum_VorAktualisierung* gespeichert haben, ist die Eigenschaft *Vor Aktualisierung* danach auf diesen Namen eingestellt, vgl. Bild 26.8.

Beispiel

Im folgenden Beispiel soll ein kleines Makro erstellt werden, das (zunächst) nicht einem bestimmten Ereignis zugeordnet werden soll. Dies könnte jedoch zu jedem späteren Zeitpunkt geschehen. Beispielsweise könnte das Makro der Eigenschaft *Beim Klicken* einer Befehlsschaltfläche eines Formulars zugeordnet werden.

Aufgabe

Erstellen Sie ein Makro, welches das Formular *Personal und Projekte* aus der Datenbank *Projekte.mdb* von der Begleit-CD-ROM in Vollbilddarstellung so öffnet, daß seine Datensätze nur gelesen, nicht aber bearbeitet werden können. Gehen Sie folgendermaßen vor, um das Makro zu erstellen:

- Öffnen Sie ggf. die Datenbank *Projekte.mdb*.
- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit F11, und aktivieren Sie dort die Objektliste *Makros*.
- Klicken Sie auf die Schaltfläche *Neu*. Sie erhalten ein neues leeres Makrofenster.
- Setzen Sie den Cursor ggf. in die erste Zeile der Spalte *Aktion*.

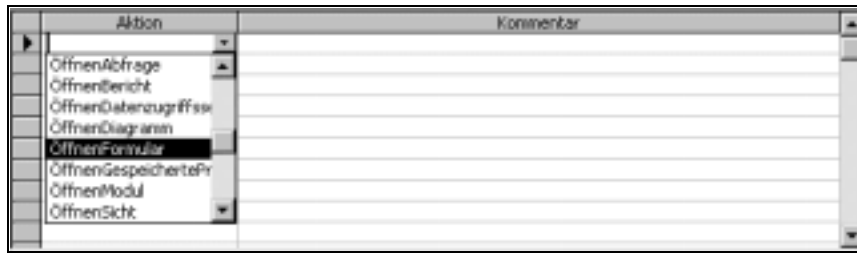


Bild 26.9: Das Dropdown-Listenfeld für die Makroaktionen ist aufgeschlagen und zur gewünschten Aktion »ÖffnenFormular« gerollt worden.

- Schlagen Sie das Dropdown-Listenfeld für die Aktionen auf, und wählen Sie daraus die Aktion *ÖffnenFormular*, vgl. Bild 26.9. Sie dürfen diesen Aktionsnamen auch schreiben, statt ihn aus dem Dropdown-Listenfeld auszuwählen. Im unteren Fensterteil sollten jetzt die verfügbaren Aktionsargumente gezeigt werden.

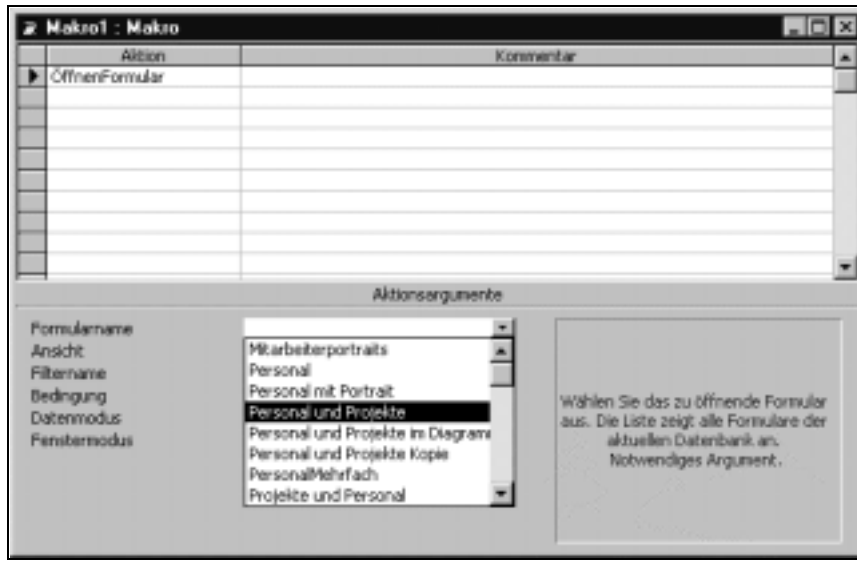


Bild 26.10: Das Dropdown-Listenfeld »Formularname« ist aufgeschlagen und zum gewünschten Formularnamen »Personal und Projekte« gerollt worden.

- Klicken Sie im Teil *Aktionsargumente* in die Zeile *Formularname*, und wählen Sie aus dem Dropdown-Listenfeld den Namen *Personal und Projekte* aus, vgl. Bild 26.10.
- Klicken Sie im Teil *Aktionsargumente* in die Zeile *Datenmodus*, und wählen Sie den Argumentwert *Nur lesen* aus.
- Klicken Sie in der Spalte *Aktion* in die nächste freie Zeile und wählen Sie die Aktion *Maximieren* aus. Diese Aktion erweitert das aktuelle Fenster zum Vollbild. Wenn Sie in dieser Situation den Teil *Aktionsargumente* des Makrofensters betrachten, erkennen Sie, daß diese Makroaktion zu den wenigen gehört, für die kein Argument angegeben werden kann.
- Speichern Sie das Makro unter dem Namen *Formular P&P lesend öffnen* (oder einem beliebigen anderen Namen). Drücken Sie dazu z.B. die Tastenkombination [Strg]+[S], und geben Sie in das Dialogfeld den Namen ein. Das fertige Makro ist in Bild 26.11 wiedergegeben.
- Starten Sie das Makro, falls das Makrofenster noch nicht geschlossen ist, durch Klicken auf die Symbol-Schaltfläche *Ausführen*. Wenn das Fenster für dieses Makro bereits geschlossen ist, rufen Sie das Makro auf, indem Sie im Datenbankfenster bei aktivierter Registerkarte *Makros* dieses Makro markieren und dann auf die Schaltfläche *Ausführen* klicken. Doppelklicken auf den Makronamen im Datenbankfenster startet das Makro ebenfalls.

Bild 26.11: Das fertige Makro wurde unter dem Namen »Formular P&P lesend öffnen« gespeichert.

26.2.2 Vorhandenes Makro bearbeiten

Analog zum Erstellen eines neuen Makros gibt es auch für das Bearbeiten zwei Wege: Zum einen können Sie ein bestehendes Makro vom Datenbankfenster aus zum Bearbeiten öffnen, zum anderen über das Eigenschaftenfenster mit Hilfe des Makro-Generators. Beide Wege seien kurz skizziert:

Makro zum Bearbeiten über das Datenbankfenster öffnen

Gehen Sie folgendermaßen vor:

- Wechseln Sie ggf. zum Datenbankfenster, z.B. mit F11, und aktivieren Sie darin die Objektliste *Makros*.
- Markieren Sie in der Liste der Makros das zu bearbeitende.
- Klicken Sie auf die Symbol-Schaltfläche *Entwurf*. Dann öffnet sich das Fenster des zu bearbeitenden Makros.
- Führen Sie die Bearbeitungen aus, testen Sie ggf. einzelne Aktionen im Einzelschritt-Modus unter Verwendung der Symbol-Schaltfläche *Einzelschritt*, und speichern Sie das geänderte Makro.

Makro zum Bearbeiten über das Eigenschaftenfenster im Formular öffnen

Gehen Sie folgendermaßen vor:

- Markieren Sie das Steuerelement, dessen Ereigniseigenschaft das Makro zugeordnet ist.
- Klicken Sie im Eigenschaftenfenster auf die Eigenschaft, die auf den Namen des zu bearbeitenden Makros eingestellt ist.
- Klicken Sie auf die Symbol-Schaltfläche *Generator* (mit den drei Punkten) am rechten Rand der Eigenschaft. Dann öffnet sich unmittelbar das Fenster des betreffenden Makros. Oben in Bild 26.8 beispielsweise ist die Eigenschaft *Vor Aktualisierung* des Textfeldes *Einstellungsdatum* auf das Makro *Einstellungsdatum_VorAktualisierung* eingestellt. Wenn Sie in diesem Falle auf die Symbol-Schaltfläche *Generator* am rechten Rand der Eigenschaft klicken, öffnet Access das Makrofenster *Einstellungsdatum_VorAktualisierung*.
- Führen Sie die Bearbeitungen aus, testen Sie ggf. einzelne Aktionen im Einzelschritt-Modus unter Verwendung der Symbol-Schaltfläche *Einzelschritt*, und speichern Sie das geänderte Makro. Prüfen Sie anschließend in der Formularansicht des Formulars, ob das Makro beim Eintreten des Ereignisses wie gewünscht ausgeführt wird.

26.2.3 Bedingte Ausführung von Aktionen

Sie können das Ausführen einer oder mehrerer Aktionen in einem Makro davon abhängig machen, ob eine bestimmte Bedingung zutreffend (d.h. *Wahr*) ist. Zu diesem Zweck geben Sie links neben der betreffenden Aktion in der Spalte *Bedingung*, die Sie zuvor ggf. mit der Symbol-Schaltfläche *Bedingung* (vgl. links nebenstehend) einblenden müssen, einen logischen Ausdruck an, der *Wahr* oder *Falsch* sein kann. Wenn die Bedingung wahr ist, wird die Aktion ausgeführt, andernfalls nicht.

Es ist auch möglich, das Ausführen von mehr als einer Aktion von derselben Bedingung abhängig zu machen. Zu diesem Zweck geben Sie in der Spalte *Bedingung* für jede weitere Aktion, deren Ausführung an die Bedingung geknüpft sein soll, drei Punkte (...) ein. Wenn die Bedingung zutrifft, werden die Aktion, zu der die Bedingung unmittelbar angegeben ist, und alle mit den drei Punkten versehenen Aktionen ausgeführt. Umgekehrt wird die Ausführung dieser Gruppe von Aktionen unterdrückt, wenn die Bedingung unzutreffend (*Falsch*) ist. Die Aktionen, die an dieselbe Bedingung geknüpft sind, müssen allerdings unmittelbar aufeinander folgen, denn Access ordnet Bedingungszeilen mit drei Punkten nur so lange der letzten darüber stehenden Bedingung zu, bis eine Aktion mit einer leeren Bedingungszeile oder eine neue Bedingung vorkommt.

Standardmäßig ist die Spalte *Bedingung* nicht eingeblendet, wenn Sie ein neues Makro erstellen. Diese Voreinstellung können Sie jedoch auf die folgende Weise ändern:

- Wählen Sie den Befehl *Optionen* aus dem Menü *Extras*.
- Wählen Sie im Dialogfeld *Optionen* die Registerkarte Ansicht, und aktivieren Sie dort *Bedingungsspalte*.

Beispiel

Aufgabe. Bearbeiten Sie das Makro *Formular P&P lesend öffnen* aus dem letzten Punkt (vgl. oben, Punkt 26.2.1, *Einfaches Makro erstellen*) so, daß das Öffnen des Formulars von der Beantwortung der Frage *Formular öffnen?*, die mit einem Meldungsdialogfeld eingeblendet wird, abhängig gemacht wird. Zu diesem Zweck verwenden Sie die Funktion[1] *Meldung*, über die Sie Genaueres in Kap. 39, *Ausdrücke, Funktionen und Operatoren*, nachlesen können. Hier sei nur so viel angemerkt: Wenn Sie als zweites Argument für die Funktion *Meldung* die Zahl 4 angeben, erscheint das Meldungsdialogfeld mit den beiden Schaltflächen *Ja* und *Nein*. Beim Klicken auf *Ja* gibt die Funktion den Wert 6 aus, beim Klicken auf *Nein* den Wert 7.

Gehen Sie folgendermaßen vor, um die Aufgabe zu lösen:

- Öffnen Sie das Makro *Formular P&P lesend öffnen*, indem Sie im Datenbankfenster die Registerkarte *Makros* aktivieren, das Makro in der Makroliste markieren und dann auf die Schaltfläche *Entwurf* klicken.
- Blenden Sie ggf. die Spalte *Bedingung* durch Klicken auf die gleichnamige Symbol-Schaltfläche ein.

1. Beachten Sie, daß hier nicht die Makro*aktion Meldung*, die auch existiert, verwendet wird, sondern die gleichlautende Funktion: In der Bedingungsspalte können nur Ausdrücke, die einen Wert ausgeben, spezifiziert werden, jedoch keine Aktionen.

- Schreiben Sie neben die Aktion *ÖffnenFormular* in die Spalte *Bedingung* den folgenden Ausdruck:

```
Meldung("Formular öffnen?";4)=6
```

- Geben Sie neben die Aktion *Maximieren* in die Spalte *Bedingung* drei Punkte ein. Ihr Makro sollte jetzt so aussehen wie in Bild 26.12.

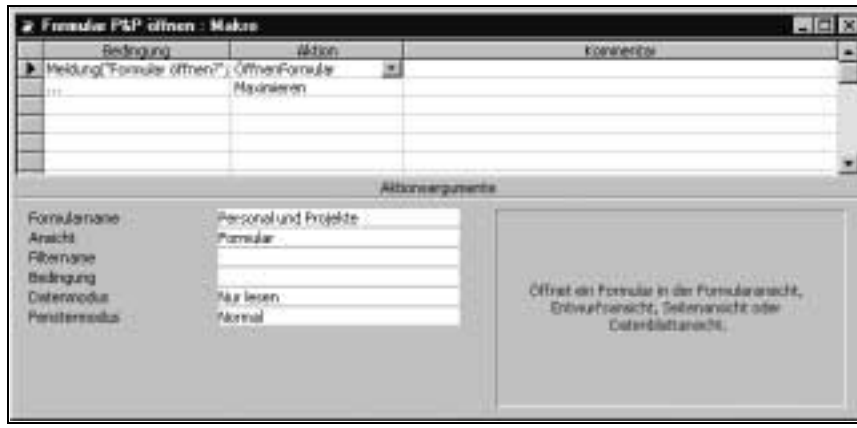


Bild 26.12: Die Bedingung »Meldung(»Formular öffnen?«;4)=6« gilt nicht nur für die erste, sondern auch für die zweite Makroaktion, weil zu dieser drei Punkte angegeben sind.

- Speichern Sie das Makro, und starten Sie es dann durch Klicken auf die Symbol-Schaltfläche *Ausführen* (vgl. links nebenstehend). Es sollte die folgende Meldungsbox eingeblendet werden:

Bild 26.13: Meldungsdialogfeld, das von der Funktion »Meldung«, wie diese in der Makrobedingung angegeben ist, eingeblendet wird.

Wenn Sie die Frage mit *Ja* beantworten, wird das Formular schreibgeschützt geöffnet. Wenn Sie die Frage mit Nein beantworten, passiert gar nichts, weil die beiden Aktionen *ÖffnenFormular* und *Maximieren* nicht ausgeführt werden.

Manchmal ist es sinnvoller, nicht nur einige Aktionen mit einer Bedingung zu verknüpfen, sondern den ganzen Rest des Makros. Dies trifft ja auch für das vorige Beispiel zu. In solchen Fällen empfiehlt es sich oft, das Makro auf die folgende Weise zu schreiben:

Bedingung	Aktion	Kommentar
Meldung("Formular öffnen?";4)=7	StoppMakro	
	ÖffnenFormular	
	Maximieren	

Bild 26.14: Hier wird die (beachten Sie: etwas umformulierte) Bedingung mit der Aktion »StopMakro« verbunden.

In diesem Falle wird die Aktion *StopMakro* verwendet und an die Bedingung geknüpft, daß die Funktion *Meldung* den Wert 7 ausgibt, was dem Klicken auf die Schaltfläche *Nein* entspricht. Eine Vorgehensweise wie in diesem Beispiel ist bei längeren Makros vorzuziehen, weil Sie sich die vielen Dreifachpunkte in der Bedingungsspalte sparen. Vor allem gewinnt das Makro an Klarheit und Übersichtlichkeit.

Die Online-Hilfe bietet mehrere Beispiele zur Verwendung von Bedingungen in Makros. Schlagen Sie das Thema *Festlegen von Bedingungen für das Ausführen von Makros* auf und klicken Sie auf die entsprechende Schaltfläche am Ende des Textes.

26.2.4 Bezugnahme auf Steuerelemente und Felder in Ausdrücken

Zum Ausfüllen von Argumenten und zur Angabe von Bedingungen müssen Sie oftmals auf Steuerelemente oder Felder von Tabellen/Abfragen Bezug nehmen. Dabei gilt prinzipiell die folgende Syntax (vgl. auch Kap. 39, *Ausdrücke, Funktionen und Operatoren*):

```
Formulare![Formularname]![Steuerelementname]
```

bzw.

```
Berichte![Berichtname]![Steuerelementname]
```

Wenn sich das Steuerelement in dem Formular bzw. Bericht befindet, von dem aus das Makro aufgerufen wird, reicht es aus, einfach den Steuerelementnamen anzugeben:

```
Steuerelementname
```

Es ist auch möglich, sich auf das Feld einer Tabelle oder Abfrage zu beziehen, die einem geöffneten Formular zugrunde liegt. Dabei muß sich aber wenigstens ein Steuerelement des Formulars auf ein Feld der Tabelle beziehen, sei es mit einem Ausdruck, sei es dadurch, daß es an ein Tabellenfeld gebunden ist.

Die Zusammenhänge seien beispielhaft an Bedingungsausdrücken demonstriert. Als Bedingung können Sie jeden logischen Ausdruck angeben, der den Wert *Wahr* oder *Falsch* ergeben kann. Die folgenden Beispiele geben eine kleine Auswahl möglicher Bedingungen im Zusammenhang mit Formularen wieder:

Bedingung	Bedeutung
[Bruttoeinkommen]>=5000	Das Steuerelement *Bruttoeinkommen* kommt im Formular vor, von dem aus das Makro aufgerufen wird.
Formulare!Mitarbeiter![Bruttoeinkommen] >=5000	Das Feld *Bruttoeinkommen* kommt im geöffneten Formular mit dem Namen *Mitarbeiter* vor. Das Makro kann von beliebiger Stelle aus aufgerufen werden.
Formulare!Aufträge![Auftragsdatum] =Formulare!Lieferungen![Lieferdatum]	Keines der beiden Formulare muß das Makro starten, aber beide müssen geöffnet sein.

26.3 Makro ausführen

Sie haben mehrere Möglichkeiten, ein Makro zu starten. In jedem Falle muß die letzte Änderung des Makros gespeichert worden sein, bevor es ausgeführt werden kann. Falls Sie ein Makro starten wollen, dessen letzte Änderung noch nicht gespeichert wurde, blendet Access eine entsprechende Meldung ein und fordert Sie auf, das Speichern zu bestätigen. Falls Sie dabei abbrechen, wird das Makro nicht ausgeführt. Zum Ausführen eines Makros wählen Sie eine der folgenden Möglichkeiten:

- *Datenbankfenster.* Wechseln Sie zum Datenbankfenster, aktivieren Sie dort die Objektliste *Makros*, markieren Sie das auszuführende Makro, und wählen Sie die Schaltfläche *Ausführen*. Doppelklicken auf den Makronamen hat dieselbe Wirkung.
- *Menübefehl Makro ausführen.* Wählen Sie den Befehl *Makro ausführen* aus dem Menü *Extras*, und wählen Sie in dessen Dialogfeld (vgl. Bild 26.15) aus dem Dropdown-Listenfeld das auszuführende Makro aus.

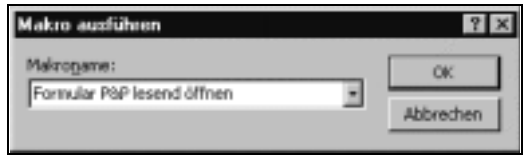


Bild 26.15: Dialogfeld »Makro ausführen« des Befehls Makro ausführen aus dem Menü »Extras«

- *Makro-Entwurfsfenster.* Klicken Sie auf die Symbol-Schaltfläche *Ausführen*.

- *Einzelschritt im Makro-Entwurfsfenster.* Klicken Sie auf die Symbol-Schaltfläche *Einzelschritt*, so daß diese gedrückt ist, und dann auf die Symbol-Schaltfläche *Ausführen*. Dann wird das Makro Schritt für Schritt ausgeführt, und Sie können vor jeder Aktion entscheiden, ob der Schritt ausgeführt werden, das Makro angehalten oder alle weiteren Aktionen ohne Anhalten ausgeführt werden sollen. Zu diesem Zweck blendet Access das Dialogfeld *Einzelschritt* ein, vgl. Bild 26.16.

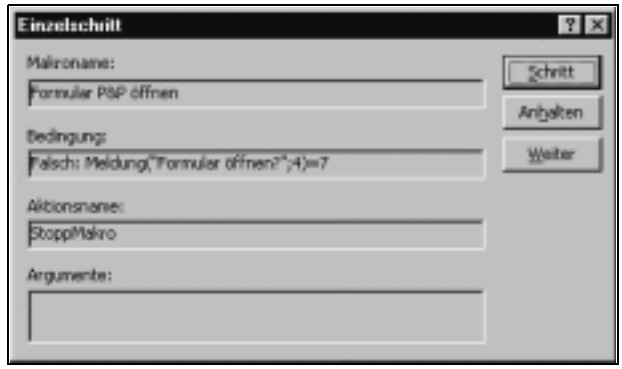


Bild 26.16: Dialogfeld »Einzelschritt«, das Access einblendet, wenn in der Symbolleiste des Makrofensters die Symbol-Schaltfläche »Einzelschritt« gedrückt ist.

- *Ereigniseigenschaft*. Stellen Sie eine Ereigniseigenschaft wie z.B. *Vor Aktualisierung* auf den Namen eines Makros ein, indem Sie dieses im Eigenschaftenfenster zum betreffenden Steuerelement, Formular oder Bericht an die entsprechende Stelle schreiben. Bestehende Makros können Sie aus einem Dropdown-Listenfeld wählen. Diese Vorgehensweise ist oft unerläßlich, weil ein Makro oft einen bestimmten Kontext voraussetzt, der beispielsweise nur gegeben ist, wenn das auslösende Ereignis stattgefunden hat.
- *Anderes Makro*. Ein Makro kann von einem anderen Makro aus aufgerufen werden. Dazu dient die spezielle Aktion *AusführenMakro*, zu der Sie auch angeben können, wie oft das Makro ggf. wiederholt ausgeführt werden und bei welcher Bedingung die Wiederholung beendet werden soll. Das aufgerufene Makro verhält sich zum aufrufenden wie ein Unterprogramm zum Hauptprogramm: Das Hauptmakro verzweigt an der Stelle der Aktion *AusführenMakro* zum Untermakro und führt die darauf folgenden Aktionen aus, wenn das Untermakro beendet ist.
- *VBA*. Mit der Methode *RunMacro* des Objekts *DoCmd* können Sie ein Makro von einer VBA-Prozedur aus aufrufen. Für den Programmfluß gilt das Entsprechende, wie zum letzten Aufzählungspunkt beschrieben.

26.4 Makrogruppen bilden

26.4.1 Makrogruppe erstellen

Bisher wurde davon ausgegangen, daß alle Aktionen eines Makrofensters zu einem Makro gehören, so daß sie auch zusammen ausgeführt werden, wenn das Makro aufgerufen wird. Umgekehrt setzen die bisherigen Ausführungen voraus, daß Sie für jedes Makro ein eigenes Makroobjekt (Fenster) erstellen. Tatsächlich können Sie jedoch in einem Makrofenster mehrere untereinander liegende Aktionen jeweils als ein eigenständiges Makro definieren. Solche Makrofenster werden von Access als *Makrogruppe* bezeichnet. Ein Makro in einer Makrogruppe kann aus beliebig vielen Aktionen bestehen, und eine Makrogruppe kann beliebig viele Makros enthalten. Sie bilden ein eigenes Makro in einem Makrofenster, indem Sie für seine erste Aktion in die Spalte *Makroname* einen Namen eingeben. Sie müssen die Spalte *Makroname* möglicherweise zunächst einblenden. Zum Makro gehören dann alle Aktionen, bis in der Spalte *Makroname* erneut ein Name auftaucht, der den Beginn eines weiteren Makros festlegt. Bild 26.17 zeigt eine Makrogruppe mit zwei Makros. Das erste Makro hat den Namen *P&PÖffnen*, das zweite den Namen *MarkiertenDatensatzDrucken*.

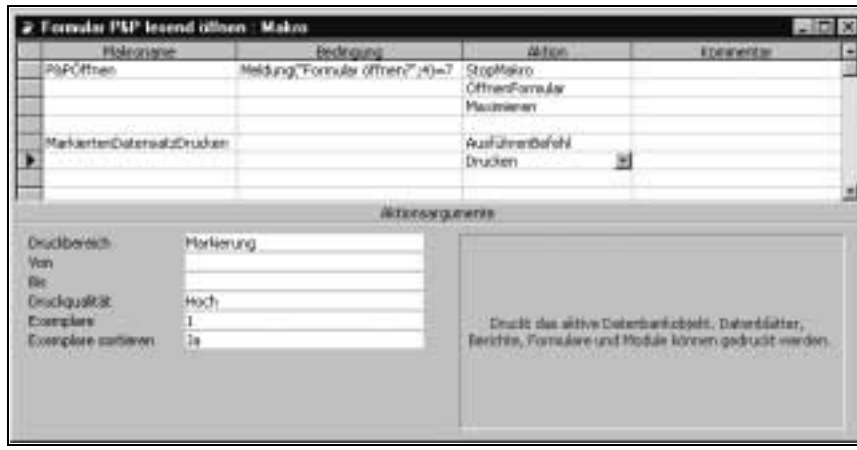


Bild 26.17: Dieses Makrofenster enthält die beiden Makros »P&PÖffnen« und »MarkiertenDatensatzDrucken«.

Standardmäßig ist die Spalte *Makroname* nicht eingeblendet, wenn Sie ein neues Makro erstellen. Diese Voreinstellung können Sie jedoch auf die folgende Weise ändern:

- Wählen Sie den Befehl *Optionen* aus dem Menü *Extras*.
- Aktivieren Sie im Dialogfeld *Optionen* die Registerkarte *Ansicht*, und kreuzen Sie die Option *Namensspalte* an.

26.4.2 Makro aus Makrogruppe ausführen

Sie können ein Makro aus einer Makrogruppe überall dort ausführen lassen, wo Sie einen Makronamen zum Ausführen angeben können. Dies trifft insbesondere für die ereignisorientierten Eigenschaften von Formularen, Berichten und Steuerelementen zu, aber auch für die Aktion *AusführenMakro*, mit der Sie von einem Makro aus ein anderes Makro starten können. Sie geben das Makro aus einer Makrogruppe an, indem Sie den Namen der Makrogruppe schreiben, gefolgt von einem Punkt und dem Namen des Makros. Beispielsweise geben Sie das erste Makro aus dem vorangehenden Punkt (vgl. Bild 26.17) in der Form

```
Formular P&P lesend öffnen.P&PÖffnen
```

und das zweite in der Form

```
Formular P&P lesend öffnen.MarkiertenDatensatzDrucken
```

an.

Wenn Sie ein Makro einer Makrogruppe als Ereignismakro verwenden wollen, schlagen Sie am besten das Dropdown-Listenfeld des betreffenden Ereignisses auf, denn dieses bietet Ihnen alle Makros – diejenigen aus Makrogruppen eingeschlossen – zur Auswahl an, vgl. Bild 26.18.

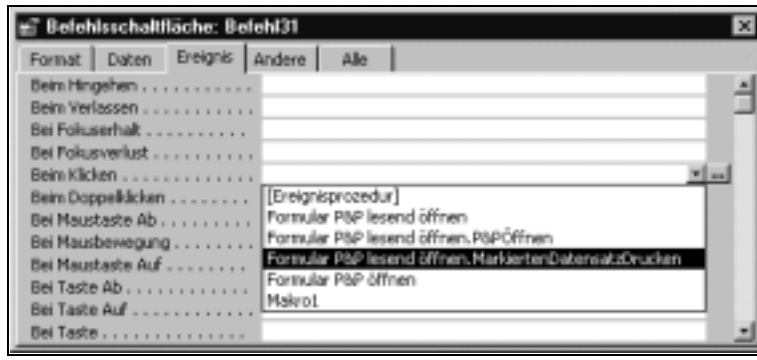


Bild 26.18: Der Eigenschaft »Beim Klicken« einer Befehlsschaltfläche wird mittels des Dropdown-Listenfeldes das Makro »Formular P&P lesend öffnen.MarkiertenDatensatzDrucken« zugeordnet.

Im allgemeinen empfiehlt es sich, häufigen Gebrauch von Makros in Makrogruppen zu machen, weil diese Arbeitsweise zu größerer Übersicht und Klarheit führt. Sinnvoll erscheint z.B. die folgende Organisationsstrategie für Makros: Bilden Sie je eine Makrogruppe für eine größere sachliche Aufgabe (z.B. für die Verwaltung eines bestimmten Formulars), die selbst wiederum aus Teilaufgaben besteht, und fassen Sie die Aktionen, die als

selbständige Einheit angesprochen werden sollen, jeweils als Makro zusammen. Angenommen, Sie benötigen für das Formular *Hauptaufträge* mehrere selbständig aufrufbare Makros. Dann empfiehlt es sich, eine Makrogruppe mit z.B. dem Namen *FrmHauptaufträge* zu erstellen, in die Sie dann die verschiedenen zusammengehörigen und selbständig ausführbaren Aktionsgruppen als Makros schreiben.

26.5 Verfügbare Makroaktionen: Übersicht

Die folgende Tabelle listet alle verfügbaren Makroaktionen auf und gibt jeweils eine Kurzerklärung dazu. Um detaillierte Auskunft zu einer Aktion zu erhalten, schlagen Sie unter dem jeweiligen Stichwort in der Online-Hilfe nach.

Sämtliche Makroaktionen

AbbrechenEreignis	Bricht das Ereignis ab, welches das Aufrufen des Makros veranlaßte, in dem diese Aktion angeführt ist.
AktualisierenDaten	Aktualisiert die Datenbasis des angegebenen Steuerelements im Objekt mit dem Fokus oder, falls kein Steuerelement angegeben wird, für das gesamte Objekt, das den Fokus hat.
AktualisierenObjekt	Aktualisiert die Bildschirmanzeige (nicht die ggf. zugrundeliegende Datensatzbasis) des angegebenen Objekts.
AnwendenFilter	Wendet einen Filter auf ein Formular, einen Bericht, eine Abfrage oder eine SQL-WHERE-Klausel an. Für Berichte ist diese Aktion nur wirksam, wenn sie als Makro der Eigenschaft *Beim Öffnen* (*OnOpen*) zugeordnet ist.
AnzeigenAlleDatensätze	Entfernt alle ggf. gesetzten Filter im aktiven Formular, so daß alle Datensätze angezeigt werden.
AusführenAnwendung	Startet eine Anwendung (Windows oder DOS).
AusführenBefehl	Führt einen Access-Befehl aus, der in einem Menü oder einer Symbolleiste abgelegt sein kann.
AusführenCode	Führt eine Funktion von Access Basic aus.
AusführenMakro	Ruft ein Makro auf.
AusführenSQL	Führt eine Aktionsabfrage unter Benutzung von SQL-Anweisungen aus.

AusgabeIn	Gibt die Daten des angegebenen Datenbankobjekts (Datenblatt, Formular, Bericht oder Modul) in verschiedenen Formaten aus.
AuswählenObjekt	Wählt das angegebene Datenbankobjekt aus.
Drucken	Druckt das aktive Objekt. Dies kann ein Datenblatt, Bericht oder Formular sein.
Echo	Bestimmt, ob die Bildschirmanzeige während eines Makros oder nach Ausführen einer VBA-Prozedur aktualisiert wird.
EinblendenSymbolleiste	Blendet eingebaute oder benutzerdefinierte Symbolleisten ein bzw. aus.
GeheZuDatensatz	Richtet den Fokus auf den angegebenen Datensatz.
GeheZuSeite	Richtet den Fokus im aktiven Formular auf die angegebene Seite.
GeheZuSteuerelement	Richtet den Fokus auf das angegebene Steuerelement. Anwendbar auf aktive(s) Formular, Datenblatt eines Formulars, Tabelle oder Dynaset.
HinzufügenMenü	Fügt der benutzerdefinierten Menüleiste ein Menü hinzu.
KopierenObjekt	Kopiert ein ausgewähltes Datenbankobjekt in eine andere oder die aktuelle Datenbank.
LöschenObjekt	Löscht das angegebene Datenbankobjekt.
Maximieren	Vergrößert das aktive Fenster zum Vollbild.
Meldung	Gibt ein Meldungsdialogfeld aus.
Minimieren	Verkleinert das aktive Fenster zum Symbol.
ÖffnenAbfrage	Öffnet eine Auswahl- oder Kreuztabellenabfrage in Datenblatt-, Entwurfs- oder Seitenansicht und führt eine Aktionsabfrage aus.
ÖffnenBericht	Öffnet einen Bericht in Entwurfs- oder Seitenansicht oder druckt ihn. Die Anzahl der zu druckenden Datensätze kann beschränkt werden.
ÖffnenDatenzugriffseite	Öffnet eine Datenzugriffsseite in Browse- oder Entwurfsansicht.
ÖffnenDiagramm	Öffnet ein Schema in der Datenblatt-, Entwurfs- oder Seitenansicht.
ÖffnenFormular	Öffnet ein Formular in Formular-, Entwurfs-, Seiten- oder Datenblattansicht. Die Anzahl der anzeigbaren Datensätze kann beschränkt werden, darüber hinaus kann auch der Dateneingabe- und Fenstermodus festgelegt werden.

ÖffnenGespeicherte-Prozedur	Öffnet eine Gespeicherte Prozedur in der Datenblatt-, Entwurfs- oder Seitenansicht.
ÖffnenModul	Öffnet das angegebene Modul an einer bestimmten Prozedur. Dabei kann es sich um eine Sub-Prozedur, eine Funktion oder eine Ereignisprozedur handeln.
ÖffnenSicht	Öffnet eine Sicht in der Datenblatt-, Entwurfs- oder Seitenansicht.
ÖffnenTabelle	Öffnet eine Tabelle in Datenblatt-, Entwurfs- oder Seitenansicht. Dabei kann der Dateneingabemodus festgelegt werden.
Positionieren	Verschiebt das aktive Fenster und/oder ändert seine Größe.
Sanduhr	Läßt den Mauszeiger zur Sanduhr werden, wenn und solange das Makro ausgeführt wird, in dem diese Aktion angeführt ist. Nach Beendigung des Makros hat der Mauszeiger wieder sein normales Aussehen.
Schließen	Schließt ein angegebenes oder das aktive Fenster.
SendenObjekt	Nimmt das angegebene Microsoft Access-Datenblatt, -Formular, -Modul oder den Bericht in eine E-Mail-Nachricht auf; diese Nachricht kann angezeigt und versendet werden.
SetzenMenüelement	Legt für das aktive Fenster den Status der Menüelemente (*Aktiviert*, *Deaktiviert*, *Mit Häkchen*, *Ohne Häkchen*) auf der benutzerdefinierten Menüleiste oder der globalen Menüleiste fest.
SetzenWert	Legt einen Wert fest für ein Feld, Steuerelement oder Eigenschaft in einem Formular, einem Datenblatt eines Formulars oder einem Bericht.
Signalton	Erzeugt einen Signalton. Die Aktion hat kein Argument.
Speichern	Speichert das angegebene oder, wenn keines angegeben ist, das aktive Objekt.
StopAlleMakros	Beendet alle laufenden Makros.
StopMakro	Beendet das laufende Makro.

SuchenDatensatz	Sucht nach dem ersten Datensatz, der die durch die Argumente festgelegten Kriterien erfüllt. Die Aktion entspricht dem Befehl *Suchen* aus dem Menü *Bearbeiten*, die Argumente korrespondieren mit den Optionen des zugehörigen Dialogfeldes.
SuchenWeiter	Sucht nach dem nächsten Datensatz, der den zuletzt zur Aktion *SuchenDatensatz* bzw. im Dialogfeld des Befehls *Suchen* aus dem Menü *Bearbeiten* angegebenen Argumenten bzw. Optionen genügt. Entspricht der Schaltfläche *Weitersuchen* im angegebenen Dialogfeld.
Tastaturbefehle	Sendet Tastenanschläge an Access oder eine andere aktive Windows-Anwendung.
TransferArbeitsblatt	Im- oder exportiert eine Tabelle von einer Tabellenkalkulation nach Access bzw. umgekehrt. Entspricht dem Befehl *Importieren* bzw. *Exportieren* bzw. *Tabelle einbinden* aus dem Menü *Datei*.
TransferDatenbank	Im- oder exportiert Daten von einer anderen Datenbank in die aktuelle Datenbank bzw. umgekehrt.
TransferText	Im- oder exportiert Text von einer Textdatei nach Access bzw. umgekehrt.
UmbenennenObjekt	Benennt ein im Datenbankfenster ausgewähltes (markiertes) Objekt um.
Verlassen	Beendet Access mit oder ohne Speichern, je nach angegebener Option.
Warnmeldungen	Legt fest, ob Systemmeldungen angezeigt werden sollen oder nicht.
Wiederherstellen	Stellt die ursprüngliche Größe eines zum Symbol verkleinerten oder zum Vollbild vergrößerten Fensters wieder her.

26.6 Autoexec-Makro

Wie Sie auf der Ebene des Betriebssystems mit der Datei AUTOEXEC.BAT eine beim PC-Start automatisch ausgeführte Batch-Datei einrichten können, haben Sie die entsprechende Möglichkeit für das Öffnen einer Access-Datenbank: Wenn Sie ein Makro erstellen und diesem den Namen *Autoexec* geben, wird es beim Öffnen der Datenbank automatisch ausgeführt. Auf diese Weise können Sie z.B. gewisse Anfangseinstellungen veranlassen.

Beispiel

Die Datenbank *Adressen.mdb* von der Begleit-CD-ROM enthält ein Makro *Autoexec*. Darin steht nur die Aktion *ÖffnenFormular*, mit der das Adreßverwaltungsformular *Adressen (Basic-Steuerung)* geöffnet wird. Daher erscheint dieses Formular unmittelbar nach dem Öffnen der Datenbank automatisch auf dem Bildschirm.

Wer die Geister ruft, muß sie auch wieder los werden können: Das automatische Ausführen des Autoexec-Makros kann in gewissen Situationen störend sein. Manchmal hat es sogar fatale Folgen: Wie können Sie an Ihre Datenbank herankommen, wenn das Autoexec-Makro z.B. die Aktion *Verlassen* enthält, mit der Access beendet wird? Sie verhindern das automatische Ausführen des Autoexec-Makros, wenn Sie beim Öffnen der Datenbank die ⇧-Taste gedrückt halten.

Wenn Sie eine Datenbank öffnen, die unter Verwendung einer älteren Version von Access erstellt wurde und noch nicht konvertiert ist, halten Sie die ⇧-Taste auch beim Bestätigen der Meldung gedrückt, die Sie darauf hinweist, daß die Datenbank in einer früheren Version von Access erstellt wurde.

Wenn Sie eine Datenbank beim Start von Access öffnen, indem Sie diese in der Befehlszeile aufführen, können Sie auch ein Makro aus dieser Datenbank angeben, das bei ihrem Öffnen ausgeführt wird. Verwenden Sie zu diesem Zweck die Option /x in der Befehlszeile. Um beispielsweise die Datenbank *Kunden.mdb* zu öffnen und dabei deren Makro *Intro* auszuführen, müßte die Befehlszeile folgendermaßen lauten (es wird unterstellt, daß der Access-Programmordner im Suchweg angegeben ist):

```
msaccess.exe kunden.mdb /x Intro
```

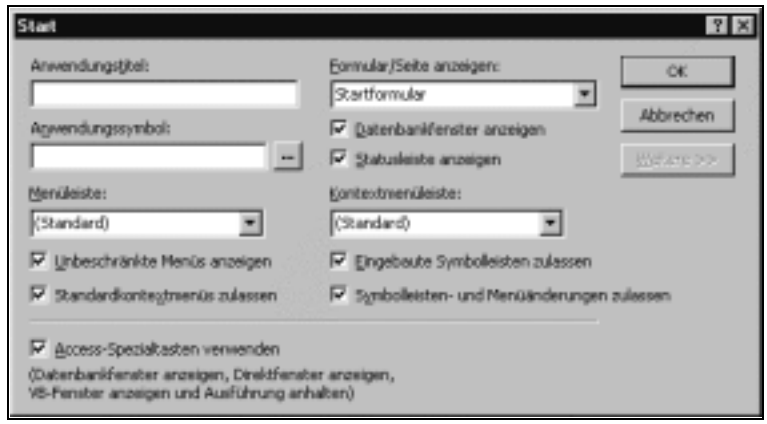


Bild 26.19: Dialogfeld »Start« des gleichnamigen Befehls aus dem Menü »Extras«

Das zur Option /x angegebene Makro darf auch einen Namen haben, der Leerzeichen enthält; Sie geben diesen ohne weitere Zeichen (z.B. keine eckigen Klammern) an.

Beachten Sie auch die Startoptionen, die Sie im Dialogfeld *Start* des gleichnamigen Befehls aus dem Menü *Extras* festlegen können. Dort können Sie u.a. ein Formular festlegen, das nach dem Start von Access automatisch geöffnet wird. In Bild 26.19 beispielsweise ist für diese Option das Formular *Startformular* angegeben.

26.7 Makrodefinitionen drucken

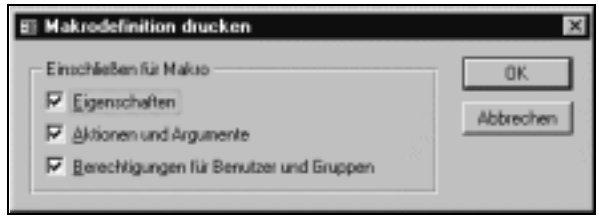


Bild 26.20: Dialogfeld »Makrodefinition drucken«

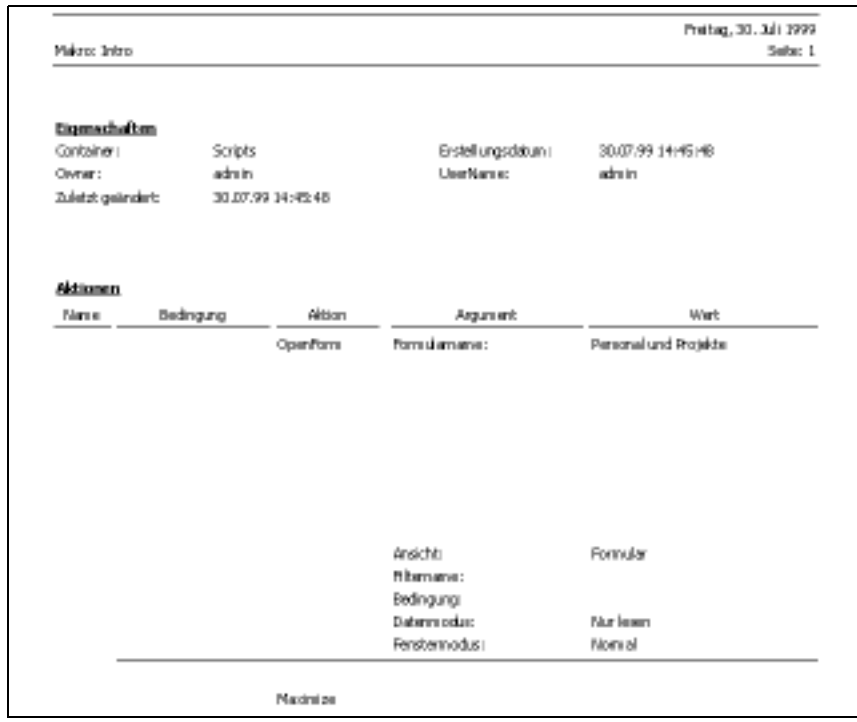


Bild 26.21: Seitenansicht der gedruckten Definitionen für das Makro »Tastaturbelegung«

Um die Makrodefinitionen – das sind die Makronamen, Bedingungen, Aktionen und deren Argumente – zu drucken, können Sie den Dokumentierer aufrufen (Menü *Extras*, Befehl *Analyse*, Unterbefehl *Dokumentierer*) oder den Befehl *Drucken* aus dem Menü *Datei* wählen, während das betreffende Makrofenster aktiv ist. Sie können dann im Dialogfeld *Makrodefinition drucken* (vgl. Bild 26.20) bestimmen, in welchem Umfang Informationen ausgegeben werden sollen.

Bild 26.21 zeigt einen Ausschnitt des Druckbildes (in der Seitenansicht) für das Makro *Intro*. Dieser Druckoutput gibt Ihnen einen besseren Überblick über die Aktionen und deren Argumente, als Sie diesen am Bildschirm im Makrofenster erhalten können, denn im letzteren Fall werden stets nur die Argumente für die Aktion eingeblendet, die gerade den Fokus hat.

26.8 Makros kopieren

Makros lassen sich wie Tabellen, Abfragen etc. kopieren, exportieren und importieren. Zum Kopieren verwenden Sie die üblichen Windows-Techniken: Markieren Sie das Makro im Datenbankfenster, kopieren Sie, und fügen Sie ein. Das Exportieren bzw. Importieren ist ganz entsprechend geregelt wie bei Tabellen, vgl. Kap. 9, Tabellen importieren, exportieren und verknüpfen.

Kapitel 27

Ereignisse

27.1 Was sind Ereignisse?

Bei der Verwendung von Formularen und Berichten werden bestimmte Vorgänge von Access als *Ereignisse* interpretiert. Beispielsweise ist das Öffnen eines Formulars oder der Vorgang, daß ein Textfeld den Fokus bekommt, jeweils ein Ereignis. Aus diesem Konzept ergeben sich große Gestaltungsmöglichkeiten, weil Sie diesen Ereignissen Makros oder VBA-Prozeduren zuordnen können, die jeweils ausgeführt werden, wenn das Ereignis eintritt.

☞ Obwohl Ereignisse für VBA-Prozeduren dieselbe Bedeutung haben wie für Makros und daher die Ausführungen dieses Kapitels für Makros und VBA-Prozeduren gleichermaßen gelten, werde ich im weiteren Verlauf dieses Kapitels aus Gründen einer einfacheren Sprache nur von Makros sprechen. Nur wenn Unterschiede existieren, wird ausdrücklich auch von VBA-Prozeduren gesprochen.

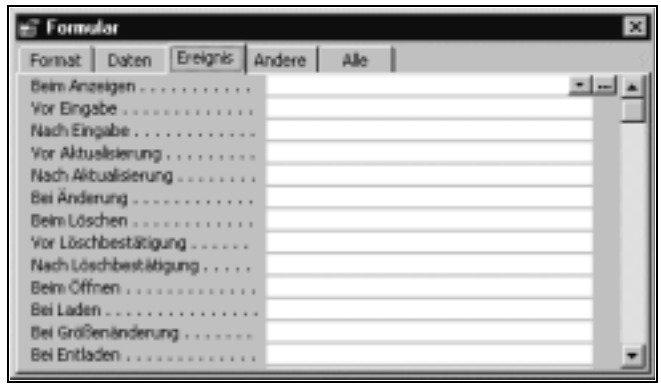


Bild 27.1: Eigenschaftenfenster für das Objekt »Formular«, hier mit aktivierter Registerkarte »Ereignis«

Ein Ereignis ist erst dann ein ganz bestimmtes Ereignis, wenn angegeben ist, für welches Objekt (Formular, Bericht oder Steuerelement) es gilt. Die Zuordnung von Objekt und Ereignis geschieht über die Eigenschaften der Objekte. Daher ordnen Sie ein bestimmtes Makro einem bestimmten Ereignis dadurch zu, daß Sie eine bestimmte Ereigniseigenschaft auf den Makronamen einstellen. Dazu verwenden Sie das Eigenschaftenfenster. Da die meisten Objekte eine große Vielfalt von Ereigniseigenschaften besitzen, empfiehlt es sich in diesen Fällen oft, im Eigenschaftenfenster die Registerkarte *Ereignis* zu aktivieren, vgl. Bild 27.1.

27.1.1 Ereignismakros

Weder der Inhalt noch der Name eines Makros macht dieses formal zu einem Ereignismakro. Dies geschieht vielmehr allein dadurch, daß Sie eine bestimmte Ereigniseigenschaft auf den Namen eines Makros einstellen. Dazu stehen Ihnen prinzipiell zwei Wege zur Verfügung:

Makro existiert bereits. Dann können Sie den Namen dieses Makros als Einstellung für die betreffende Eigenschaft angeben, indem Sie es aus dem Dropdown-Listenfeld auswählen, vgl. Bild 27.2. Ein auf diese Weise zugeordnetes Makro können Sie übrigens am schnellsten bearbeiten, wenn Sie, während die betreffende Ereigniseigenschaft im Eigenschaftenfenster den Fokus hat, auf die Symbol-Schaltfläche *Generator* (Symbol am rechten Rand der Eigenschaft mit drei Punkten) klicken. Dann öffnet sich das entsprechende Makrofenster unmittelbar zum Bearbeiten.

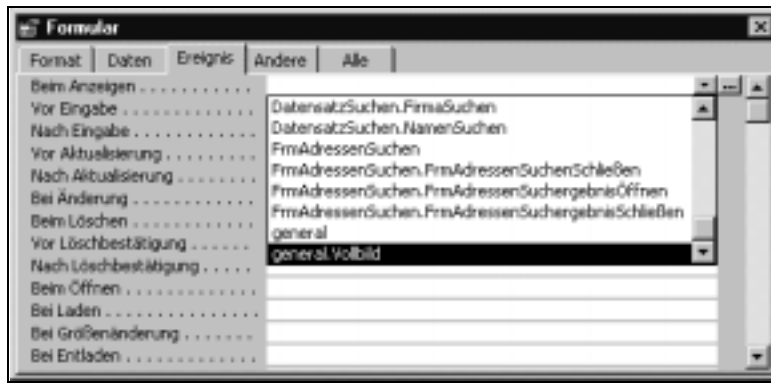


Bild 27.2: Aufgeschlagenes Dropdown-Listenfeld, das die in der Datenbank verfügbaren Makros zur Auswahl anzeigt. Die Liste umfaßt auch die Makros aus Makrogruppen.

Makro soll für die Ereigniseigenschaft neu erstellt werden. Wie Sie für diesen Zweck im einzelnen verfahren, ist ausführlich in Kap. 26, *Makros: Grundlagen*, Punkt 26.2.1, *Einfaches Makro erstellen*, beschrieben. Prinzipiell gehen Sie folgendermaßen vor:

- Klicken Sie im Eigenschaftenfenster auf das Einstellungsfeld der betreffenden Eigenschaft, so daß diese den Fokus hat.
- Klicken Sie auf die Symbol-Schaltfläche *Generator* (mit drei Punkten) am rechten Rand der Ereigniseigenschaft, und wählen Sie im sich dann öffnenden Dialogfeld den Makro-Generator. Nach dem Speichern des Makros wird dessen Name als Einstellung der Ereigniseigenschaft von Access automatisch eingetragen.

27.1.2 Ereignisprozeduren

Statt eines Makros können Sie einer Ereigniseigenschaft auch eine VBA-Prozedur zuordnen. Zu diesem Zweck können Sie eine speziell und nur für diese Ereigniseigenschaft geltende Ereignisprozedur erstellen, oder Sie verwenden eine bereits vorhandene VBA-Funktion. Auf die inhaltliche Seite von VBA-Prozeduren und -Funktionen wird ausführlich in Kap. 30, *Programmieren mit VBA*, eingegangen. Wie Ereignisprozeduren in Formularen und für deren Steuerelemente organisiert sind und wie Sie allgemein VBA-Prozeduren und -Funktionen einer Ereigniseigenschaft zuordnen, können Sie in Kap. 15, *Formulare: Grundlagen*, Punkt 15.8, *Ereignisprozeduren und -makros in Formularen*, erfahren.

27.2 Liste sämtlicher Ereignisse nach Kategorien

27.2.1 Fensterereignisse

Fensterereignisse treten ein, wenn Sie ein Formular oder einen Bericht öffnen, schließen oder seine Größe ändern.

Ereigniseigenschaft	Situation	Abbruch möglich
BeimSchließen	Das Ereignis *Schließen* tritt ein, wenn ein Formular oder Bericht geschlossen und nicht mehr angezeigt wird.	Nein
BeimÖffnen	Bei Formularen tritt das Ereignis *Öffnen* beim Öffnen des Formulars, aber vor dem Anzeigen des ersten Datensatzes ein. Bei Berichten tritt dieses Ereignis ein, bevor ein Bericht in der Seitenansicht angesehen oder gedruckt wird.	Ja
BeiLaden	Das Ereignis *Laden* tritt ein, wenn ein Formular geöffnet wird und Datensätze angezeigt werden.	Nein

Ereigniseigenschaft	Situation	Abbruch möglich
BeiEntladen	Das Ereignis *Entladen* tritt ein, wenn ein Formular geschlossen, aber noch auf dem Bildschirm angezeigt wird.	Ja
BeiGrößenänderung	Das Ereignis *Größenänderung* tritt ein, wenn ein Formular geladen wird. Außerdem tritt es immer dann ein, wenn sich die Größe eines Formulars ändert.	Nein

27.2.2 Fokusereignisse

Fokusereignisse treten ein, wenn ein Formular oder ein Bericht den Fokus erhält oder verliert, oder wenn das Formular oder der Bericht aktiviert oder deaktiviert wird.

Ereigniseigenschaft	Situation	Abbruch möglich
BeiAktivierung	Das Ereignis *Aktivierung* tritt ein, wenn ein Formular oder ein Bericht den Fokus erhält und zum aktiven Fenster wird.	Nein
BeiDeaktivierung	Das Ereignis *Deaktivierung* tritt ein, wenn ein Formular den Fokus an ein anderes Formular oder an das Tabellen-, Abfrage-, Berichts-, Makro-, Modul- oder Datenbankfenster abgibt.	Nein
BeimHingehen	Das Ereignis *Hingehen* tritt ein, bevor ein Steuerelement tatsächlich den Fokus von einem anderen Steuerelement im selben Formular erhält.	Nein
BeimVerlassen	Das Ereignis *Verlassen* tritt unmittelbar vor der Abgabe des Fokus von einem Steuerelement an ein anderes Steuerelement im selben Formular ein.	Ja
BeiFokuserhalt	Das Ereignis *Fokuserhalt* tritt ein, wenn ein Formular oder ein Steuerelement den Fokus erhält.	Nein
BeiFokusverlust	Das Ereignis *Fokusverlust* tritt ein, wenn ein Formular oder ein Steuerelement den Fokus abgibt.	Nein

27.2.3 Datenereignisse

Datenereignisse treten ein, wenn Daten in ein Formular oder Steuerelement eingegeben, gelöscht oder verändert werden, oder wenn der Fokus von einem Datensatz zu einem anderen wechselt.

Ereigniseigenschaft	Situation	Abbruch möglich
BeimLöschen	Das Ereignis *Löschen* tritt ein, nachdem Sie, z.B. durch Drücken der Entf-Taste, einen Vorgang ausgelöst haben, der einen Datensatz löscht, jedoch bevor der Datensatz tatsächlich gelöscht wird.	Ja
VorLöschbestätigung	Das Ereignis *VorLöschbestätigung* tritt ein, nachdem Sie einen oder mehrere Datensätze gelöscht haben, jedoch bevor Access ein Dialogfeld einblendet, das Sie dazu auffordert, den Löschvorgang zu bestätigen.	Ja
NachLöschbestätigung	Das Ereignis *NachLöschbestätigung* tritt ein, wenn Sie die Löschvorgänge bestätigt haben und die Datensätze tatsächlich gelöscht sind oder wenn der Löschvorgang abgebrochen wurde.	Nein
VorEingabe	Das Ereignis *VorEingabe* tritt ein, nachdem Sie das erste Zeichen eines neuen Datensatzes eingegeben haben, jedoch bevor der zugehörige Datensatz tatsächlich angelegt wird.	Ja
NachEingabe	Das Ereignis *NachEingabe* tritt ein, nachdem ein neuer Datensatz hinzugefügt wurde.	Nein
VorAktualisierung	Das Ereignis *VorAktualisierung* tritt ein, bevor geänderte Daten in einem Steuerelement oder einem Datensatz aktualisiert werden.	Ja
NachAktualisierung	Das Ereignis *NachAktualisierung* tritt ein, nachdem geänderte Daten in einem Steuerelement oder einem Datensatz aktualisiert worden sind.	Nein

Ereigniseigenschaft	Situation	Abbruch möglich
BeiÄnderung	Das Ereignis *Änderung* tritt ein, wenn sich der Inhalt eines Textfeldes oder des Textfeldteils eines Kombinationsfeldes ändert. Beispiele: Ein Zeichen wird unmittelbar in das jeweilige Text- oder Kombinationsfeld eingegeben; die Einstellung der Eigenschaft *Text* des Steuerelements wird mit einem Makro oder mit VBA geändert.	Nein
BeimAnzeigen	Das Ereignis *Anzeigen* tritt ein, wenn ein Datensatz den Fokus erhält und so zum aktuellen Datensatz wird oder wenn das Formular erneut abgefragt wird.	Nein
BeiNichtInListe	Das Ereignis *NichtInListe* tritt ein, wenn Sie in das Textfeld eines Kombinationsfeldes einen Wert eingeben, der nicht in der Liste des Kombinationsfeldes enthalten ist.	Nein
BeiOLE-Aktualisierung	Das Ereignis *OLEAktualisierung* tritt ein, wenn Daten eines OLE-Objekts geändert wurden.	Nein

27.2.4 Mausereignisse

Ein Mausereignis tritt ein, wenn Sie in einem Formular oder Steuerelement eine Mausaktion ausführen.

Ereigniseigenschaft	Situation	Abbruch möglich
BeimKlicken	Das Ereignis *Klicken* tritt ein, wenn Sie eine Maustaste drücken und wieder freigeben, während sich der Mauszeiger über einem Objekt befindet. Das Ereignis »Klicken« kann auch eintreten, wenn sich die Einstellung der Eigenschaft *Wert* eines Steuerelements ändert.	Nein

Ereigniseigenschaft	Situation	Abbruch möglich
BeimDoppelklicken	Das Ereignis *Doppelklicken* tritt ein, wenn Sie zweimal die linke Maustaste drücken und wieder freigeben, während sich der Mauszeiger über einem Objekt befindet. In einem Formular tritt das Ereignis *Doppelklicken* ein, wenn Sie auf eine leere Fläche oder ein deaktiviertes Steuerelement des Formulars doppelklicken. Bei einem Steuerelement tritt dieses Ereignis ein, wenn Sie auf ein Steuerelement oder das zugehörige Bezeichnungsfeld doppelklicken. Das Ereignis *Doppelklicken* tritt ein, wenn Sie auf das Formular oder das Steuerelement doppelklicken, jedoch bevor das Ergebnis des Doppelklickens angezeigt wird.	Ja
BeiMaustasteAb	Das Ereignis *MaustasteAb* tritt ein, wenn Sie eine Maustaste drücken.	Ja (nur mit Makro)
BeiMaustasteAuf	Das Ereignis *MaustasteAuf* tritt ein, wenn Sie eine Maustaste loslassen.	Nein
BeiMausbewegung	Das Ereignis *Mausbewegung* tritt ein, wenn Sie die Maus bewegen.	Nein

27.2.5 Tastaturereignisse

Ein Tastaturereignis tritt ein, wenn Sie eine Eingabe über die Tastatur vornehmen oder wenn Tastenanschläge mit der Makroaktion *Tastaturbefehle* (SendKeys) an die Anwendung gesendet werden.

Ereigniseigenschaft	Situation	Abbruch möglich
BeiTaste	Das Ereignis *Taste* tritt ein, wenn Sie eine Taste oder Tastenkombination drücken und loslassen, die einem ANSI-Code entspricht.	Ja (nur mit Makro)
BeiTasteAb	Das Ereignis *TasteAb* tritt ein, wenn Sie eine Taste drücken, während das Formular oder das Steuerelement den Fokus hat.	Nein
BeiTasteAuf	Das Ereignis *TasteAuf* tritt ein, wenn Sie eine Taste loslassen, während das Formular oder das Steuerelement den Fokus hat.	Nein

27.2.6 Druckereignisse

Ein Druckereignis tritt ein, wenn ein Bericht gedruckt oder zum Drucken formatiert wird.

Ereigniseigenschaft	Situation	Abbruch möglich
BeimFormatieren	Das Ereignis *Formatieren* tritt ein, wenn Access die Daten für einen Bereich eines Berichts bestimmt, aber bevor der Bereich für die Seitenansicht oder den Ausdruck formatiert wird.	Ja
BeimDrucken	Das Ereignis *Drucken* tritt ein, wenn Daten in einem Bereich eines Berichts zum Drucken formatiert werden, aber bevor der Bereich gedruckt wird.	Ja
BeiRücknahme	Das Ereignis *Rücknahme* tritt ein, wenn Access während der Formatierung eines Berichts zu einem vorherigen Berichtsbereich zurückkehrt.	Nein
BeiSeite	Das Ereignis *Seite* tritt ein, wenn Access eine Seite zum Drucken formatiert hat, jedoch bevor die Seite gedruckt wird.	Ja
BeiOhneDaten	Das Ereignis *Ohne Daten* tritt ein, wenn Access einen Bericht, der keine Daten enthält, zum Drucken formatiert hat (der Bericht ist an eine leere Datensatzgruppe gebunden), jedoch bevor der Bericht gedruckt wird. Sie können dieses Ereignis verwenden, um das Drucken eines leeren Berichts abzubrechen.	Ja

27.2.7 Fehlerereignisse und zeitabhängige Ereignisse

Die beiden folgenden Ereignisse beziehen sich auf das Auftreten von Fehlern bzw. die Synchronisation von Daten.

Ereigniseigenschaft	Situation	Abbruch möglich
BeiFehler	Das Ereignis *Fehler* tritt ein, wenn in Access ein Laufzeitfehler erzeugt wird. Dazu gehören Fehler der Jet Datenbank-Engine, nicht jedoch Laufzeitfehler in VBA.	Ja
BeiZeitgeber	Das Ereignis *Zeitgeber* tritt in gleichbleibenden Intervallen ein, die von der Eigenschaft *Zeitgeberintervall* des Formulars festgelegt sind. Diese Eigenschaft ist für das Arbeiten in einer Mehrbenutzerumgebung relevant.	Nein

27.2.8 Filterereignisse

Ereigniseigenschaft	Situation
BeiFilter	Das Ereignis *BeiFilter* tritt ein, wenn Sie auf *Formularbasierter Filter* klicken, nachdem Sie im Menü *Datensätze* auf *Filter* gezeigt haben, oder in der Befehlsleiste auf die Schaltfläche *Formularbasierter Filter* klicken, oder wenn Sie auf *Spezialfilter/-sortierung* klicken, nachdem Sie im Menü *Datensätze* auf *Filter* gezeigt haben.
BeiAngewendetemFilter	Das Ereignis *BeiAngewendetemFilter* tritt ein, wenn Sie im Menü *Datensätze* auf *Filter/Sortierung anwenden* oder in der Befehlsleiste auf die Schaltfläche *Filter/Sortierung anwenden* klicken; wenn Sie auf *Auswahlbasierter Filter* klicken, nachdem Sie im Menü *Datensätze* auf *Filter* gezeigt haben, oder in der Befehlsleiste auf die Schaltfläche *Auswahlbasierter Filter* klicken; wenn Sie im Menü *Datensätze* auf *Filter/Sortierung entfernen* oder in der Symbolleiste auf die Schaltfläche *Filter entfernen* klicken; wenn Sie das Fenster *Filter* oder *Formularbasierter Filter* schließen.

Kapitel 28

Ereignisabhängige Makros in Formularen und Steuerelementen

Wenn Sie viel mit Formularen arbeiten und diese benutzerfreundlich gestalten wollen, werden Sie dazu sehr oft mehr oder minder umfangreiche Makros verwenden. Wie Sie prinzipiell Makros in ein Formular und dessen Steuerelemente einbinden, wird in Kap. 27, *Ereignisse*, Punkt 27.1.1, *Ereignismakros*, gezeigt. In diesem Abschnitt sollen einige für die praktische Anwendung repräsentative Beispiele zeigen, wie Sie Makros in Formularen vorteilhaft verwenden.

28.1 Befehlsschaltfläche

Befehlsschaltflächen eignen sich hervorragend, dem Benutzer die Möglichkeit zu geben, bestimmte Aktionen zu veranlassen. Beispielsweise können Sie von einem Formular aus ein anderes öffnen lassen und dafür eine Befehlsschaltfläche vorsehen. Sie binden das entsprechende Makro an die Eigenschaft *Beim Klicken*.

Beispiel

In Bild 28.1 ist das Formular *Auswahltafel* in der Entwurfsansicht, die jedoch kaum von der Formularansicht abweicht, wiedergegeben. Sie erkennen, daß es die vier Befehlsschaltflächen *Auftrag*, *Adressen*, *Mahnungen* und *Beenden* enthält. Für jede dieser Befehlsschaltflächen ist die Eigenschaft *Beim Klicken* auf ein Makro eingestellt. Diese vier Makros befinden sich in der Makrogruppe *Auswahltafel*, vgl. Bild 28.2. Die ersten drei Makros

bestehen aus der einzigen Aktion *ÖffnenFormular* mit jeweils anderem Formularnamen als Aktionsargument, das Makro *Beenden* enthält die Aktion *Verlassen*, mit der Access beendet wird.

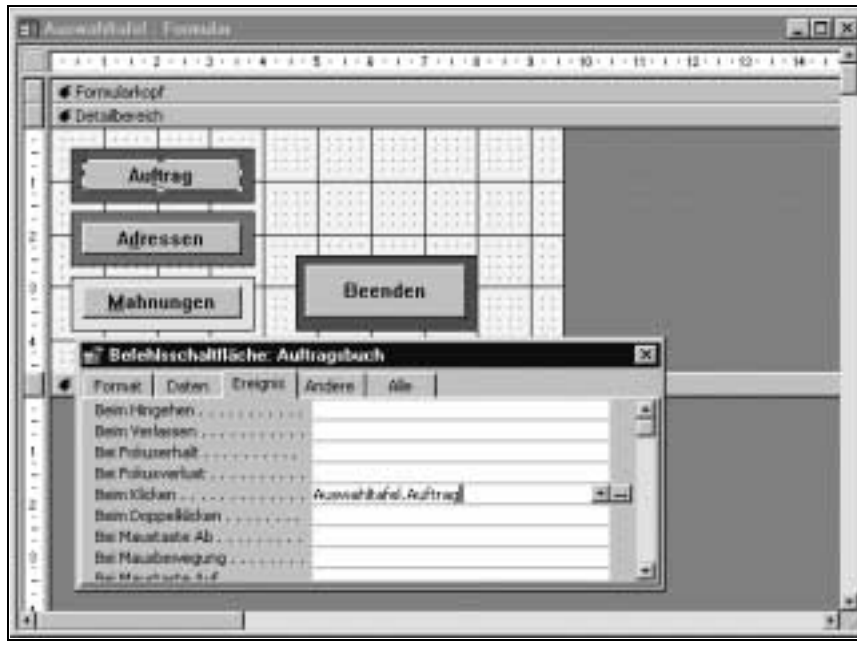


Bild 28.1: Formular »Auswahltafel« in der Entwurfsansicht. Das Eigenschaftenfenster zeigt die Eigenschaften der Befehlsschaltfläche »Auftragsbuch« an und läßt erkennen, daß deren Eigenschaft »Beim Klicken« auf das Makro »Auswahltafel.Auftrag« eingestellt ist.

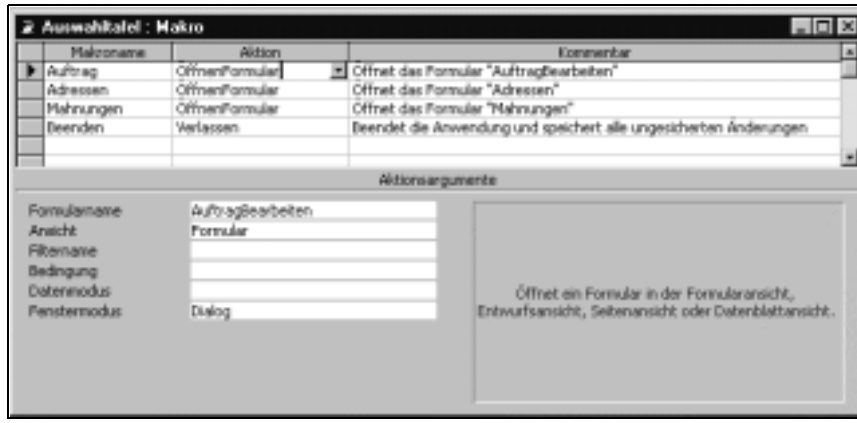


Bild 28.2: Makrogruppe »Auswahltafel« mit den vier Makros für die Befehlsschaltflächen des Formulars »Auswahltafel«

Isoliertes Schließen eines Formulars verhindern

Das Formular *Auswahltafel* hat den Charakter einer reinen Auswahltafel. Um zu verhindern, daß ein derartiges Formular geschlossen werden kann und dennoch die Anwendung weiter läuft, können Sie der Formulareigenschaft *Beim Schließen* ein Makro mit der Makroaktion *Verlassen*, die die Anwendung Access beendet, zuordnen. Um dem Benutzer jedoch Gelegenheit zu geben, auf möglicherweise versehentliches Schließen noch reagieren zu können, weisen Sie der Formulareigenschaft *Bei Entladen* ein Makro zu, das Access nach einer Warnmeldung, auf die der Benutzer vor dem Schließen reagieren kann, beendet. Bild 28.3 gibt ein derartiges Makro wieder.

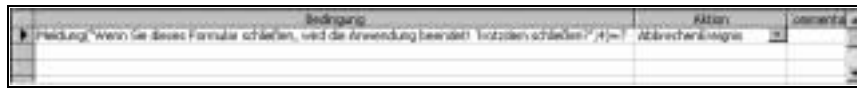


Bild 28.3: Die Bedingung zur Makroaktion »Beenden« lautet: »Meldung(»Wenn Sie dieses Formular schließen, wird die Anwendung beendet! Trotzdem schließen?«;4)=7«.

Wenn das Formular, dessen Eigenschaft *Bei Entladen* das Makro aus Bild 28.3 zugeordnet ist, geschlossen werden soll, erscheint die in Bild 28.4 wiedergegebene Meldung. Die als Bedingung angegebene Funktion gibt den Wert 7 aus, wenn auf die Meldung mit *Nein* reagiert wird. Dann wird das Formular nicht geschlossen und Access nicht beendet, weil das Ereignis des Schließens mit der Makroaktion *AbbrechenEreignis* abgebrochen wird.

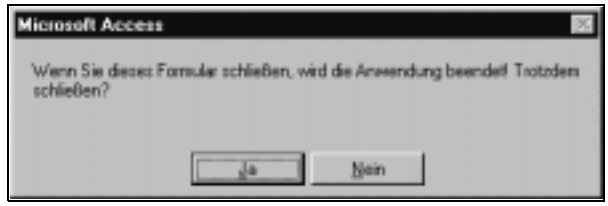


Bild 28.4: Meldung des Makros aus Bild 28.3

Sie können im vorstehenden Beispiel nicht die Eigenschaft *Beim Schließen* verwenden, weil sich das Ereignis *Schließen* nicht mit der Makroaktion *AbbrechenEreignis* abbrechen läßt.

Befehlsschaltflächen eignen sich, wie gezeigt, besonders gut dazu, ein anderes Formular vom aktuellen Formular aus zu öffnen. Beachten Sie dabei, daß die Makro-Aktion *ÖffnenFormular* auch das Aktionsargument *Fenstermodus* besitzt. Dieses können Sie auf *Dialog* einstellen, wenn Sie das Formular als Dialogfeld öffnen wollen. Dann ist das Formular, von dem aus Sie das Dialogfeld-Formular geöffnet haben, nach dem Schließen des letzteren automatisch wieder im Vordergrund, weil Sie kein anderes Formular öffnen, schließen oder ausblenden konnten, solange das Dialogfeld-Formular noch geöffnet war.

28.2 Gültigkeitsprüfung (Beispiel)

Beispiel

Aufgabe. Das Formular in Bild 28.5 sieht u.a. die Eingabemöglichkeit für die drei angeführten Felder vor. Jedesmal, wenn ein Wert in einem der drei Felder geändert wird, soll geprüft werden, ob die Summe der beiden Felder *Mit Erschwerniszulage* und *Mit Zeitzulage* größer ist als die *Leistungseinheiten insgesamt*. Wenn dies zutrifft, soll die Werteingabe bzw. -änderung verhindert werden, begleitet von der in Bild 28.5 wiedergegebenen Meldung.

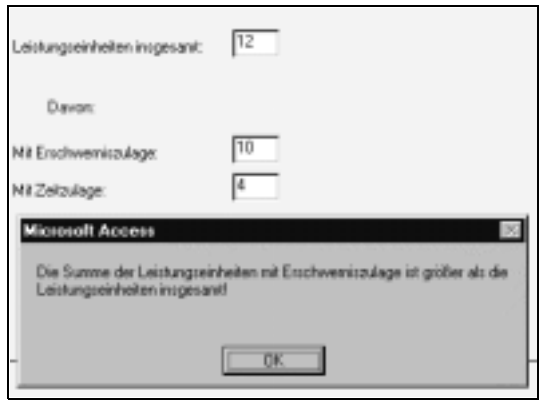


Bild 28.5: Formular mit Meldung über verletzte Gültigkeit

Um die Aufgabe zu lösen, gehen Sie folgendermaßen vor:

- Erstellen Sie ein Makro mit den beiden Aktionen *Meldung* und *AbbrechenEreignis*.
- Geben Sie für das Aktionsargument *Meldung* den in Bild 28.5 zu lesenden Text an.
- Geben Sie für die Aktion *Meldung* die folgende Bedingung an (die zwischen eckigen Klammern stehenden Namen sind die Namen der drei Steuerelemente):

```
[Leistungseinheiten insgesamt]-[Mit Erschwerniszulage]-[Mit
Zeitzulage]<0
```

- Geben Sie für die Aktion *AbbrechenEreignis* in die Spalte *Bedingung* drei Punkte ein, damit die Bedingung für die vorangehende Aktion auch für diese gilt.
- Speichern Sie das Makro unter einem Namen, z.B. *LeistungseinheitenPrüfen*.

Bild 28.6: Gültigkeitsmakro für die drei Textfelder im Formular aus Bild 28.5

- Öffnen Sie das Formular in der Entwurfsansicht, und stellen Sie für jedes der drei Steuerelemente die Eigenschaft *Vor Aktualisierung* auf das Makro *LeistungseinheitenPrüfen* (oder einen anderen von Ihnen gewählten Namen) ein.
- Wechseln Sie in die Formularansicht, und prüfen Sie, ob alles klappt.

Sie hätten diese Gültigkeitsprüfung auch als Gültigkeitsregel der dem Formular zugrunde liegenden Tabelle formulieren können. Dann allerdings würde die Regel erst überprüft, wenn der bearbeitete Datensatz gespeichert werden soll. Im hier dargestellten Beispiel werden die Werte dagegen gleich bei der Eingabe in die Felder überprüft – bevor der Datensatz gespeichert wird.

28.3 Steuerelementen Werte zuweisen (Beispiel)

Im folgenden Formularausschnitt werden die *GesamtkostenNetto* eingegeben. Die *Mehrwertsteuer* und die *GesamtkostenBrutto* sollen berechnet und in der Tabelle gespeichert werden. Da Sie in der Tabelle keine Berechnungen vornehmen können, muß dies im Formular erledigt werden. Andererseits können Sie berechnete Steuerelemente nicht in Tabellenfeldern speichern, weil bei solchen Steuerelementen eine Formel als Steuerelementinhalt angegeben ist, so daß kein Tabellenfeld mehr angegeben werden kann. Die Lösung besteht darin, daß alle drei Felder an die Tabelle gebunden werden. Jedoch wird nur in das Textfeld *GesamtkostenNetto* ein Wert vom Benutzer eingegeben. Die anderen beiden Felder werden mit der Aktion *SetzenWert* von einem Makro mit den entsprechenden Werten versorgt.

Bild 28.7: Rechnungsfelder

Gehen Sie im einzelnen folgendermaßen vor:

- Erstellen Sie ein Makro.
- Fügen Sie die Aktion *SetzenWert* hinzu. Geben Sie für das Argument *Feld* den Steuerelementnamen [Mehrwertsteuer] an und für das Argument *Ausdruck* den folgenden Ausdruck (vgl. auch Bild 28.8):

```
[GesamtkostenNetto]*0,16
```

- Fügen Sie eine weitere Aktion *SetzenWert* hinzu. Geben Sie für das Argument *Feld* den Steuerelementnamen [GesamtkostenBrutto] an und für das Argument *Ausdruck* den folgenden Ausdruck:

```
[GesamtkostenNetto]+[Mehrwertsteuer]
```

- Speichern Sie das Makro unter dem Namen *Rechnung* (jeder andere Name tut es natürlich auch).

Bild 28.8: Makro »Rechnung« mit den beiden »SetzenWert«-Aktionen zum Festlegen der Werte in den Textfeldern »Mehrwertsteuer« und »GesamtkostenBrutto«.

- Öffnen Sie das Formular in der Entwurfsansicht, und geben Sie zur Eigenschaft *Nach Aktualisierung* des Steuerelements *GesamtkostenNetto* den Makronamen *Rechnung* an (oder den anderen von Ihnen beim Speichern gewählten).
- Um Widersprüche zu vermeiden ist es erforderlich, die Steuerelemente *Mehrwertsteuer* und *GesamtkostenBrutto* zu sperren, weil Benutzer sonst versehentlich falsche Werte eingeben könnten. Stellen Sie zu diesem Zweck deren Eigenschaft *Gesperrt* auf *Ja* ein.

28.4 Datensätze filtern (Beispiel)

Beispiel

Der Begleit-CD-ROM ist die Datenbank *Adressen.mdb* beigefügt. Sie enthält u.a. das Formular *Adressen (Makro-Steuerung)* zur Adreßverwaltung. Der Formularname erklärt sich daraus, daß in der Datenbank ein zweites Formular mit dem Namen *Adressen (VBA-Steuerung)* existiert. Beide Formulare leisten ungefähr das Gleiche, wenngleich das letztere auf leistungsfähigere Art Datensätze suchen kann. Ich habe die Datenbank *Adressen.mdb* mit den beiden angegebenen Formularen eigens für den Zweck konzipiert, einen direkten Vergleich von Makros und VBA-Prozeduren zu ermöglichen. An dieser Stelle geht es um die Steuerung mit Makros.

Bild 28.9 gibt das Formular *Adressen (Makro-Steuerung)* aus der im letzten Abschnitt genauer bezeichneten Datenbank *Adressen.mdb* wieder. Im vorliegenden Beispiel geht es um die beiden Reihen mit den Schaltflächen *A*, *B* ...*Z*, *Alle* für *Firma* bzw. *Name*. Sie dienen dazu, Filter für das Formular zu setzen bzw. zurückzunehmen: Wenn Sie beispielsweise auf die Schaltfläche *A* in der Reihe *Firma* klicken, wird ein Filter für das Formular gesetzt, so daß nur Datensätze angezeigt werden, deren Firma mit dem Buchstaben A beginnt. Entsprechend wirken die anderen Buchstaben-Schaltflächen, in der einen Reihe für das Feld *Firma*, in der anderen für das Feld *Name*. Mit der Schaltfläche *Alle* werden wieder alle Datensätze angezeigt.

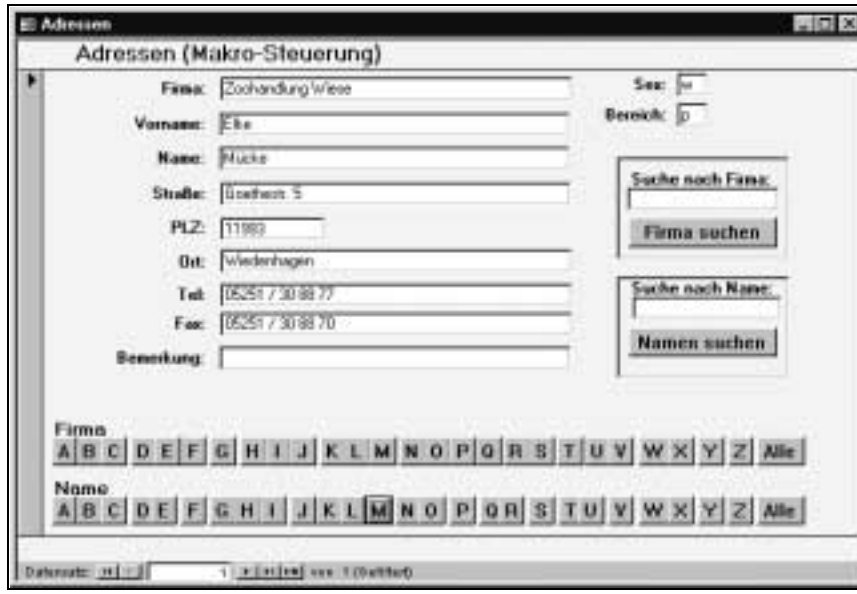


Bild 28.9: Formular »Adressen (Makro-Steuerung)« aus der Datenbank Adressen.mdb in der Formularansicht

Im folgenden skizziere ich, wie die beschriebenen Wirkungen erreicht werden. Da Sie die Datenbank *Adressen.mdb* von der Begleit-CD-ROM, das Formular und die Makros öffnen und betrachten können, ist es möglich, die Zusammenhänge am Bildschirm anzuschauen und auszuprobieren. Daher verzichte ich an dieser Stelle darauf, einzelne Schritte zum Nacharbeiten aufzulisten.

Basis für das Setzen der Filter sind die beiden Makrogruppen mit den Namen *A-Z Firmenfilter* und *A-Z Namenfilter.* Ich konzentriere mich auf die erstere der beiden Makrogruppen. Bild 28.10 zeigt einen Ausschnitt daraus. Die Makrogruppe enthält die 26 Makros *A*, *B* ... *Z* für die 26 Buchstaben das Alphabets sowie das Makro *Alle* zum Aufheben des Filters. Jedes Makro besteht aus nur der einzigen Aktion *AnwendenFilter* (das letzte Makro *Alle* besteht aus der Aktion *AnzeigenAlleDatensätze*). Allerdings ist das Argument *Bedingung* für jeden Buchstaben anders angegeben. In Bild 28.10 hat die Zeile für den Buchstaben *B* gerade den Fokus. Daher sehen Sie, daß in diesem Falle das Aktionsargument *Bedingung* mit

```
[Firma] Wie "B*"
```

angegeben ist. Entsprechend lautet die Argumentspezifikation für die anderen Buchstaben. Lediglich für die Vokale und einige andere Buchstaben wurde noch dem Umstand Rechnung getragen, daß auch Umlaute und/oder weitere Sonderzeichen berücksichtigt werden sollen. Für den Buchstaben O beispielsweise ist

```
[Firma] Wie "[OÒÓÔÕÖ]*"
```

angegeben worden.

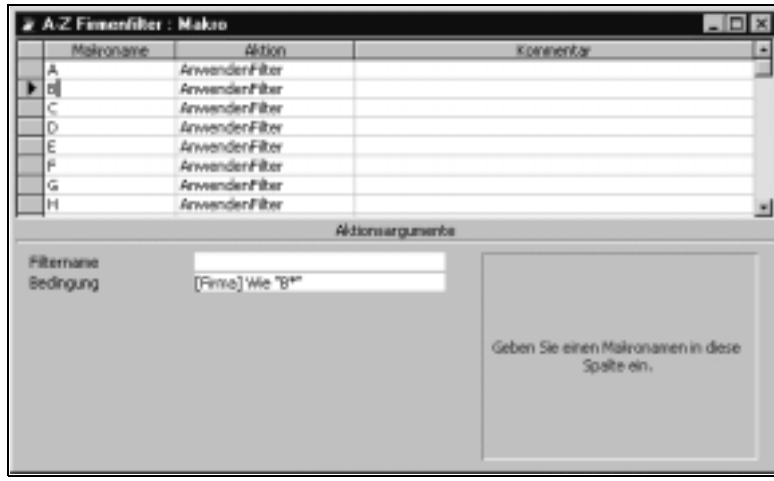


Bild 28.10: Makrogruppe »A-Z Firmenfilter«, Ausschnitt

Im Formular *Adressen (Makro-Steuerung)* ist die Eigenschaft *Beim Klicken* für jede der Firmenfilter-Schaltflächen *A*, *B*, ... *Z*, *Alle* auf das zugehörige Makro eingestellt. Für die Schaltfläche *A* beispielsweise lautet die Eigenschafteneinstellung *A-Z Firmenfilter.A*, für die Schaltfläche *B* lautet sie *A-Z Firmenfilter.B* etc.

Die Leiste mit den Namenfilter-Schaltflächen ist auf ganz entsprechende Weise versorgt.

Am Formular *Adressen (VBA-Steuerung)* können Sie studieren, wie das Setzen von Filtern mit den Buchstaben-Schaltflächen mit VBA-Prozeduren bewerkstelligt werden kann, vgl. auch Kap. 31, *Programmieren mit Access-Basic*.

28.5 Datensatz suchen (Beispiel)

Hinweis. Formular und Makro dieses Beispiels sind in der Datenbank Adressen.mdb auf der Begleit-CD-ROM enthalten; vgl. dazu genauer den ersten Abschnitt im vorangehenden Punkt 28.4, *Datensätze filtern (Beispiel)*.

Das Formular *Adressen (Makro-Steuerung)* sieht auch eine Möglichkeit vor, nach einem Datensatz mit einer bestimmten Firma bzw. einem bestimmten Namen zu suchen, vgl. oben im vorangehenden Punkt Bild 28.9. Dabei reicht es für die Suche aus, wenn nur die ersten Buchstaben als Suchbegriff angegeben werden. Ferner sind Stellvertreterzeichen erlaubt. Eine solche Suche könnten Sie mit dem Menübefehl *Suchen* aus dem Menü *Bearbeiten* durchführen. Das setzt aber nicht nur voraus, daß zuvor das richtige Feld im Formular den Fokus hat, sondern Sie müssen auch weitere Angaben im Dialogfeld des Befehls vornehmen. Für den durchschnittlichen Benutzer ist es vermutlich einfacher, sicherer und bequemer, die Suche durch direkte Eingabe des Firmennamens (bzw. Nachnamens) in das Formular und Klicken auf die entsprechende Schaltfläche auszuführen.

Wie diese Suche nach einem Datensatz organisiert ist, erkläre ich für die Firma; für den Namen ist dies völlig analog dazu geregelt.

Vorkehrung im Formular

Das Formular enthält in seinem rechten oberen Teil das Textfeld *SuchbegriffFirma* und darunter die Schaltfläche *Firma suchen* und die entsprechenden Steuerelemente für die Suche nach Namen, vgl. Bild 28.11.

Die Eigenschaft *Beim Klicken* der Schaltfläche *Firma suchen* wurde auf das Makro *DatensatzSuchen.FirmaSuchen* eingestellt. Der Inhalt dieses Makros ist weitgehend aus Bild 28.12 zu erkennen:

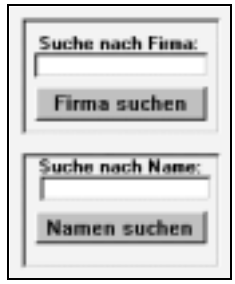


Bild 28.11: Ausschnitt aus dem Formular »Adressen (Makro-Steuerung)« der Datenbank Adressen.mdb

Bild 28.12: Makrogruppe »DatensatzSuchen« mit den beiden Makros »FirmaSuchen« und »NamenSuchen«

Die erste Aktion des Makros *FirmaSuchen* lautet *GeheZuSteuerelement*. Als Aktionsargument *Steuerelementname* ist dazu der Name *Firma* angegeben. Daher richtet das Makro *FirmaSuchen* zunächst den Fokus auf das Steuerelement *Firma*. Dies ist für das richtige Funktionieren der nächsten Aktion *SuchenDatensatz*, welche dem Befehl *Suchen* aus dem Menü *Bearbeiten* entspricht, vorausgesetzt. Die Argumentspezifikationen für die Aktion *SuchenDatensatz* können Sie in Bild 28.12 erkennen. Die Argumente entsprechen im wesentlichen den Optionen im Dialogfeld des Menübefehls *Suchen*. Hier gilt allerdings die Besonderheit, daß als Suchbegriff ein Ausdruck mit dem Steuerelement *SuchbegriffFirma* aus dem Formular *Adressen (Makro-Steuerung)* angegeben ist. Auf diese Weise wird erreicht, daß die Aktion *SuchenDatensatz* einen Datensatz sucht, dessen Feld *Firma* einen Text aufweist, der so anfängt wie der in das Formular eingegebene *SuchbegriffFirma*.

Die Suche nach Namen ist ganz analog zur Suche nach Firmen organisiert.

Am Formular *Adressen (VBA-Steuerung)* können Sie nachvollziehen, wie die Suche nach Datensätzen mit VBA-Prozeduren bewerkstelligt werden kann. Mit Makros steht Ihnen für die Datensatzsuche nur die Aktion *SuchenDatensatz* zur Verfügung. Mit dieser Aktion können Sie nur nach dem Inhalt *eines* Feldes suchen. Die in VBA verfügbare Methode *FindFirst* erlaubt es dagegen, nach mehr als einem Suchbegriff bzw. Feldinhalt suchen zu lassen. Im Formular *Adressen (VBA-Steuerung)* beispielsweise können Sie gleichzeitig nach Firma, Vorname und Name suchen lassen und auf diese Weise die Suche spezifischer durchführen.

28.6 Formulare synchronisieren

In einem Hauptformular, das ein verknüpftes Unterformular enthält, werden Haupt- und Unterformular automatisch von Access synchronisiert: Das Unterformular zeigt immer die Datensätze an, die dem aktuellen Datensatz des Hauptformulars entsprechen, wobei die Entsprechung über die beiden Verknüpfungsfelder hergestellt wird.

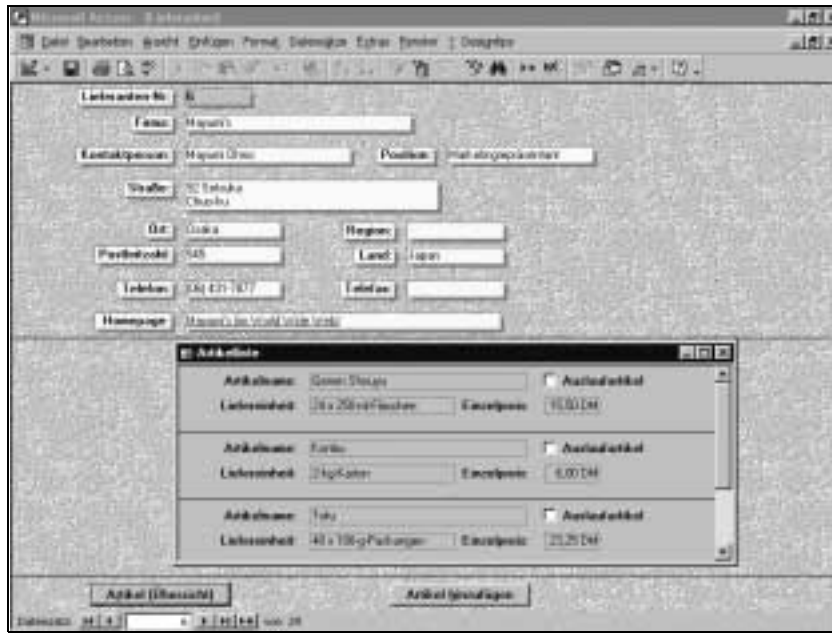


Bild 28.13: Formular »Lieferanten« der Datenbank »Nordwind.mdb« mit geöffnetem Formular »Artikelliste«, dessen Datensätze mit dem Formular »Lieferanten« synchronisiert sind.

Mit Hilfe von Makros können Sie auch zwei selbständige Formulare, die nicht miteinander verknüpft sind, synchronisieren. Die Datenbank *Nordwind.mdb* enthält ein interessantes Beispiel dafür. Bild 28.13 zeigt das Formular *Lieferanten* dieser Datenbank. Nach dem Klicken auf die Schaltfläche

Artikel (Übersicht) wurde das Formular *Artikelliste* als Endlosformular eingeblendet, vgl. ebenfalls Bild 28.13. Es ist mit dem Formular *Lieferanten* synchronisiert, denn es zeigt nur und gerade die Artikel an, die zu dem aktuellen Lieferanten des Formulars *Lieferanten* gehören. Die Synchronisation erfolgt beim erstmaligen Öffnen und beim Blättern zwischen Datensätzen im Formular *Lieferanten*.

Die Steuerung des Formulars *Lieferanten* erfolgt tatsächlich mittels VBA-Ereignisprozeduren. Der Datenbank Nordwind.mdb ist jedoch die Makrogruppe *Lieferanten* beigefügt, die Makros enthält, welche die gleiche Steuerung des Formulars Lieferanten leisten könnten, wie dieses die Ereignisprozeduren tun. So heißt es in der ersten Kommentarzeile dieser Makrogruppe: »Die Makros in dieser Gruppe können dem Formular »Lieferanten«, statt dem Code im Modul des Formulars, zugeordnet werden.« Aus sprachlichen Gründen (sonst müßte alles weitere im Konjunktiv formuliert werden) unterstelle ich für die folgenden Ausführungen, daß das Formular durch die Makros der Makrogruppe *Lieferanten* gesteuert wird.

Die Synchronisation zwischen den beiden Formularen erfolgt über zwei Makros: Das Makro *Artikelübersicht* (vgl. Bild 28.14) öffnet das Formular *Artikelliste* und synchronisiert seine Datensätze mit dem aktuellen Datensatz im Formular *Lieferanten*. Dieses Makro ist der Eigenschaft *Beim Klikken* der Befehlsschaltfläche *Artikel (Übersicht)* zugeordnet. Das Makro *Zugehörige Artikel anzeigen* bringt das geöffnete Formular *Artikelliste* in den Vordergrund und stellt ebenfalls die Synchronisation her. Dieses Makro ist der Eigenschaft *Beim Anzeigen* des Formulars *Lieferanten* zugeordnet und wird daher ausgeführt, wenn in diesem Formular ein neuer Datensatz angezeigt wird, also nach dem Blättern zu einem anderen Datensatz.

Die Synchronisation der beiden Formulare erfolgt in beiden Makros durch die Bedingung

```
[Lieferanten-Nr]=[Formulare]![Lieferanten]![Lieferanten-Nr]
```

die jeweils als Aktionsargument *Bedingung* für die Aktion *ÖffnenFormular* angegeben ist, vgl. Bild 28.14, wo diese Bedingung unvollständig zu lesen ist. Die linke Seite des Bedingungsausdrucks bezieht sich auf das Feld *Lieferanten-Nr* des zu öffnenden Formulars *Artikelliste*, die rechte auf die entsprechende im Formular *Lieferanten*. Sie können diese Bedingung als eine Anweisung interpretieren, im zu öffnenden Formular *Artikelliste* unmittelbar nach dem Öffnen dem Datensatz mit der angegebenen *Lieferanten-Nr* den Fokus zu geben.

Der Unterschied zwischen den beiden Makros ist gering:

Makro Artikelübersicht, das der Eigenschaft *Beim Klicken* der Befehlsschaltfläche *Artikel (Übersicht)* zugeordnet ist, öffnet das Formular *Artikelliste* – vorausgesetzt, das Feld *Lieferanten-Nr* enthält keinen Nullwert. In diesem Falle erfolgt eine Meldung, und das Makro wird abgebrochen.

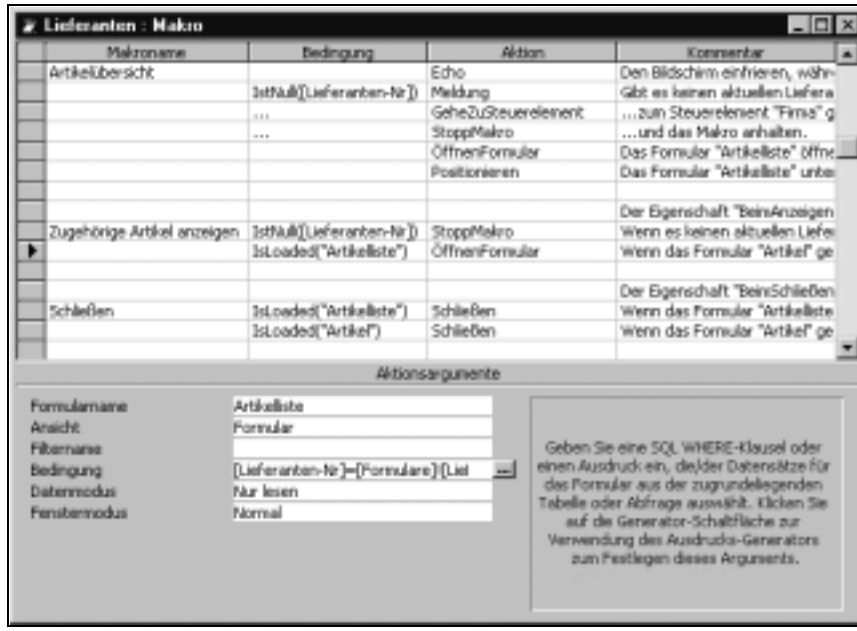


Bild 28.14: Makro »Artikelübersicht«. Dieses Makro ist der Eigenschaft »Beim Klicken« der Befehlsschaltfläche »Artikel (Übersicht)« zugeordnet.

Makro Zugehörige Artikel anzeigen, das der Eigenschaft *Beim Anzeigen* des Formulars *Lieferanten* zugeordnet ist, bringt das geöffnete Formular *Artikelliste* in den Vordergrund und synchronisiert dabei über den Bedingungsausdruck für die Aktion *ÖffnenFormular* die beiden Formulare. Dafür müssen Sie wissen, wie die Aktion *ÖffnenFormular* wirkt, wenn sie an einem bereits geöffneten Formular ausgeführt wird: Dieses Formular erhält dann den Fokus und kommt damit in den Vordergrund. Diese Aktion wird allerdings nur ausgeführt, wenn das Formular *Artikelliste* bereits geöffnet ist. Dies wird mit der Bedingung

```
IsLoaded("Artikelliste")
```

geprüft. Die Funktion *IsLoaded* ist eine benutzerdefinierte, die sich im Modul *Dienstprogramme* der Datenbank *Nordwind.mdb* befindet. Sie gibt den Wert -1 (d.h. *Wahr*) aus, wenn das Formular – hier: *Artikelliste* – bereits geöffnet ist. In diesem Falle wird die Aktion *ÖffnenFormular* ausgeführt. Wenn das Formular nicht geöffnet ist, gibt die Funktion den Wert 0 (oder Falsch) aus, so daß die Aktion *ÖffnenFormular* nicht ausgeführt wird.

Kapitel 29

VBA (Visual Basic for Applications): Grundlagen

29.1 Was ist VBA?

Gleichgültig, ob Sie es *Visual Basic for Applications* oder *Visual Basic für Anwendungen* nennen, in jedem Falle gilt dieselbe Abkürzung: VBA. Gemeint ist damit die Programmiersprache von Microsoft, die nicht nur in Access 2000, sondern in allen Office 2000-Anwendungen verfügbar ist. Diese für das Office-Paket einheitliche Programmiersprache soll es den Benutzern erleichtern, ihre Aufgaben zu automatisieren. Bei VBA in der vorliegenden Version handelt es sich um eine sehr leistungsfähige objektorientierte Sprache, die nicht nur über in modernen Programmiersprachen übliche Strukturen, z.B. lokale und globale Variable oder Bedingungs-, Schleifen- und Programmkontrollstrukturen, verfügt, sondern die den Programmierer auch in die Lage versetzt, über Objekte und deren Eigenschaften und Methoden der eigenen, aber auch anderer Anwendungen zu verfügen. So ist es möglich, von Access 2000 aus über die Funktionalitäten von Excel 2000 oder Word 2000 zu verfügen. Man spricht dann davon, daß diese Anwendungen als *Automationsserver* dienen, die vom *Client* Access aufgerufen werden. Auch umgekehrt können von diesen Anwendungen aus die Dienste von Access 2000 in Anspruch genommen werden. Für alle Office 2000-Anwendungen gemeinsam verfügbar sind auch die Objekte DAO (Data Access Object) und ADO (ActiveX Data Objects), mit deren Methoden und Eigenschaften über Datenbanken und deren Objekte verfügt werden kann. Mit DAO wird den anderen Office 2000-Anwendungen die *JetEngine*, die den Kern des RDBMS Access bildet, zur Verfügung gestellt. Wenn Sie ausschließlich in Access programmieren, nehmen Sie möglicherweise gar nicht wahr, daß die Methoden zur Datenmanipulation dem besonderen Objekt DAO zuzurechnen sind, weil dieses ohne weitere Vorkehrun-

gen im VBA von Access zur Verfügung steht. Wegen der gleichzeitigen Verfügbarkeit der DAO-Methoden in den anderen Office 2000-Anwendungen habe ich die Ausführungen zum Datenzugriff mit DAO jedoch in ein eigenes Kapitel gelegt, vgl. Kap. 31, *Datenzugriff mit DAO (Data Access Objects) und ADO (ActiveX Data Objects)*.

Anders als beispielsweise in Excel 2000 und Word 2000 steht in Access 2000 kein Makro-Recorder, der Aktionen von Access aufzeichnen und damit das Entwerfen von Programmcode erleichtern könnte, zur Verfügung. Dafür bietet Access aber sehr leistungsfähige Makros, mit denen Sie jede Access-Aktion automatisieren können, ohne programmieren zu müssen (vgl. dazu in diesem Buch Kap. 26-28). Da ein Makro-Recorder letztlich auch nichts anderes leistet, als Anwendungsaktionen in Form von Programmcode aufzuzeichnen, ist der Umstand, daß in Access kein Makro-Recorder zur Verfügung steht, kein größerer Mangel. Typische Programmstrukturen wie bedingte Schleifen oder bedingte Anweisungen kann ein Makro-Recorder ohnehin nicht erstellen.

29.2 Prozeduren, Funktionen und Sub-Prozeduren: Sprachvereinbarung zur Klarheit

Programme in VBA werden entweder als Funktionen oder als Sub-Prozeduren geschrieben. Auf die inhaltliche Bedeutung dieser beiden Programmtypen wird weiter unten in diesem Kapitel eingegangen, vgl. Punkt 29.7, Funktionen und Sub-Prozeduren. An dieser Stelle geht es nur darum, sprachliche Klarheit zu schaffen: Das Wort *Prozedur* gilt im allgemeinen als Oberbegriff für alle möglichen selbständigen Programmteile in Programmcodes. In diesem Sinne ist sowohl eine VBA-Funktion wie auch eine VBA-Sub-Prozedur eine Prozedur. Um die beiden Bedeutungen von Prozedur auseinanderhalten zu können, gilt in diesem Buch (Access hält etwas ähnliches leider nicht immer konsequent ein) die folgende Konvention:

- **VBA-Funktion**. Damit wird eine vom Benutzer in VBA geschriebene Funktion bezeichnet. Gelegentlich bezeichne ich eine solche Funktion auch als benutzerdefinierte Funktion, um sie von den in Access standardmäßig vordefinierten Funktionen wie z.B. *Datum*, *Minute* etc. zu unterscheiden.
- **VBA-Sub-Prozedur**. Damit wird eine vom Benutzer in VBA geschriebene Prozedur, im Unterschied zu einer Funktion, bezeichnet. (Der Wortteil *Sub* erklärt sich daraus, daß eine solche Prozedur mit dem Schlüsselwort *Sub* definiert wird.)
- **Prozedur**. Das Wort Prozedur ohne den Zusatz *Sub* wird in der Bedeutung eines Oberbegriffs für Funktionen und Sub-Prozeduren verwendet.

29.3 Für welche Zwecke lassen sich VBA-Prozeduren einsetzen?

29.3.1 Vorteile von VBA-Prozeduren

In VBA erstellte Funktionen und Sub-Prozeduren lassen sich für vielfältige Zwecke verwenden. Die wichtigsten Einsatzgebiete sind die folgenden:

- **Benutzerdefinierte Funktionen** – also solche, die in VBA vom Benutzer erstellt und nicht von vornherein in Access eingebaut sind – können in jedem Ausdruck in Access verwendet werden. Sie können sie einsetzen wie eine der Access-Standardfunktionen. Auf diese Weise können Sie beispielsweise sehr komplexe Berechnungen für ein berechnetes Steuerelement in einem Formular ausführen lassen, indem Sie die entsprechende benutzerdefinierte Funktion in einem Ausdruck, den Sie als *Steuerelementinhalt* angeben, anführen.
- **Datenmanipulation**. Sie können Manipulationen von Daten in Tabellen, Formularen, Abfragen etc. durch VBA-Funktionen oder Sub-Prozeduren automatisch ausführen lassen, wenn bestimmte Ereignisse in Formularen oder Berichten eintreten, indem Sie die Prozeduren an entsprechende ereignisorientierte Eigenschaften binden. Darüber hinaus können Sie in VBA eine Datensatzgruppe Datensatz für Datensatz durchlaufen und ggf. für jeden Datensatz Operationen ausführen. Auf diese Weise lassen sich mit einer einzigen Prozedur oft große Datenbestände automatisch bearbeiten.
- **Leichte Wartbarkeit der Anwendung**. Gegenüber Makros, mit denen Sie Access-Vorgänge ebenfalls weitestgehend automatisieren können, bieten VBA-Prozeduren den Vorteil besserer Wartbarkeit der Anwendung: Makros sind im allgemeinen auf viele verschiedene Makroobjekte verteilt, VBA-Code können Sie dagegen in wenigen Modulen halten. Darüber hinaus werden die Ereignisprozeduren zusammen mit den Formularen und Berichten, für deren Ereigniseigenschaften sie gelten, in speziellen Ereignismodulen gehalten, die stets zusammen mit dem Formular oder Bericht gespeichert werden, so daß die Prozeduren ggf. auch zusammen mit einem Formular oder Bericht in eine andere Datenbank exportiert oder von dort importiert werden.
- **Anzeigen benutzerdefinierter Fehlermeldungen**. Wenn Sie eine Anwendung nur mit Makros, jedenfalls ohne VBA erstellen, sind Sie auf die von Access vordefinierten Fehlermeldungen angewiesen. Mit VBA können Sie diese jedoch im allgemeinen vor ihrem Auftreten abfangen und durch selbstformulierte Fehlermeldungen, die dem Sachzusammenhang der Anwendung besser entsprechen, ersetzen.

- **OLE-Automatisierung**. Mit Hilfe der OLE-Automatisierung können Sie eine andere Anwendung nicht nur starten (dies ist auch mit der Makro-Aktion *AusführenAnwendung* möglich), sondern diese auch steuern, indem Sie beispielsweise Aktionen der anderen Anwendung ausführen lassen. Diese Möglichkeiten stehen nur in VBA zur Verfügung.
- **Komplexe Datenbankanwendungen**. Mit VBA – oft in Verbindung mit Makros – ist es möglich, benutzerfreundliche und sehr komplexe Datenbankanwendungen zu erstellen, die weit über die Möglichkeiten der normalen Access-Benutzeroberfläche hinausgehen.

29.3.2 Vorteile von Makros

- **Einfach, sicher und bequem**. Mit VBA-Prozeduren besitzen Sie gegenüber Makros den Vorteil der größeren Flexibilität und Leistungsfähigkeit. Dafür zahlen Sie den Preis eines erheblichen Lernaufwandes, bis Sie in der Lage sind, komplexere VBA-Prozeduren fehlerfrei und absturzsicher zu programmieren. Makros können Sie dagegen fast kinderleicht erstellen, sie laufen darüber hinaus nahezu immer fehlerfrei ab.
- **Prototypentwicklung**. Da es bei Prototypen im allgemeinen nicht auf ausgefeilte Einzelheiten, sondern auf die großen strukturellen Zusammenhänge ankommt und natürlich darauf, daß der Prototyp schnell verfügbar ist, eignen sich Makros besonders gut für diese Aufgabe.
- **Autoexec**. Das spezielle Makro mit dem Namen *Autoexec* wird automatisch mit dem Öffnen der Datenbank gestartet. Eine analoge VBA-Prozedur – wie die Prozedur *Auto_Open* im Excel-VBA – existiert in Access nicht.
- **Benutzerdefinierte Symbol-Schaltflächen**. Um einer Schaltfläche in eine Menü- oder Symbolleiste benutzerdefinierte Aktionsfolgen zuzuweisen, müssen Sie ein Makro verwenden. Nur indirekt können Sie dafür eine VBA-Prozedur heranziehen, indem Sie ein Makro mit der Aktion *AusführenCode* angeben, welche die VBA-Prozedur aufruft.

29.4 Visual Basic-Entwicklungsumgebung

29.4.1 Projekte, Formulare und Module

Die Organisierung von VBA ist mit Access 2000 so geändert worden, daß sie derjenigen in Visual Basic als Einzelversion und den anderen Office 2000-Anwendungen gleicht: VBA ist in Projekten organisiert. Jedes Projekt enthält die Module und Klassenobjekte (Code-Module von Formularen und Berichten) der geöffneten Datenbank.

Projekte

Praktisch ist ein VBA-Projekt in Access im allgemeinen eine Datenbank, wie diese sich aus der Sicht von VBA darstellt. In Bild 29.1 sehen Sie beispielsweise das Projekt *Nordwind*, wie es im Fenster des Projekt-Explorers dargestellt wird. (Der Projekt-Explorer wird im einzelnen weiter unten in diesem Kapitel behandelt, vgl. Punkt 29.4.2, *Visual Basic-Editor*, Unterpunkt *Projekt-Explorer*.) Wenn Sie die Datenbank *Nordwind.mdb* öffnen, laden Sie gleichzeitig das Projekt *Nordwind*. Sie erkennen, daß das Projekt *Nordwind* zunächst zwei weitere Objekte enthält: *Microsoft Access Klassenobjekte* und *Module*. Jedes dieser Objekte enthält weitere Unterobjekte: 13 Klassenobjekte und drei Module. Bei den Klassenobjekten wie *Form_Artikel*, *Report_Umsätze nach Jahr* etc. handelt es sich jeweils um ein Modul, das dem entsprechenden Formular/Bericht zugeordnet ist und den VBA-Code der Ereignisprozeduren dieses Formulars enthält. Die allgemeinen Module wie *Dienstprogramme* oder *Start* enthalten dagegen VBA-Code für Prozeduren, die der gesamten Datenbank zur Verfügung stehen.

Bild 29.1: Projekt, dargestellt im Projekt-Explorer

VBA-Code in Modulen

VBA-Prozeduren werden als Texte geschrieben und in Modulen gespeichert und verwaltet. Es gibt drei Arten von Modulen:

- **Formular- und Berichtsmodule**. Diese erscheinen im Visual Basic-Editor als Klassenobjekte, vgl. den vorangehenden Abschnitt Projekte. Alle Ereignisprozeduren eines bestimmten Formulars oder Berichts werden in einem Modul gehalten, das in das Formular bzw. den Bericht integriert ist.

Formular- und Berichtsmodule werden zusammen mit dem Formular bzw. Bericht gespeichert. Prozeduren dieser Module sind für das bestimmte Formular bzw. den Bericht spezifisch, d.h., sie können nicht von anderen Access-Objekten aus aufgerufen werden, insbesondere auch nicht von Prozeduren anderer Module. Formular- und Berichtsmodule können auch mit dem *Code-Generator* des Formulars oder Berichts bzw. mit dem Befehl *Ereignis* des Kontextmenüs geöffnet werden. Weitere Details über die Organisierung von Ereignisprozeduren in Formularen finden Sie in Kap. 15, *Formulare: Grundlagen*, Punkt 15.8, *Ereignisprozeduren und -makros in Formularen*.

- **Globale Standardmodule**. Globale Standardmodule werden außer im Projekt-Explorer des Visual Basic-Editors auch im Datenbankfenster in der Objektliste *Module* angezeigt und lassen sich auch von dort aus öffnen. Obwohl es möglich ist, alle Funktionen und Sub-Prozeduren in ein und dasselbe Modul zu schreiben (abgesehen von Klassenmodulen, vgl. den folgenden Aufzählungspunkt), können Sie diese andererseits auf mehr als ein Modul verteilen. Die letztere Vorgehensweise ermöglicht Ihnen ggf. eine bessere Ordnung, indem Sie z.B. funktionelle Gruppen jeweils im selben Modul zusammenfassen. Globale Module werden zusammen mit der Datenbank gespeichert. Die entsprechenden Funktionen und Sub-Prozeduren können dennoch Datenbank-übergreifend zur Verfügung stehen, wenn Sie in einer anderen Datenbank einen Verweis auf die Datenbank einrichten, über deren Prozeduren Sie verfügen wollen. Durch entsprechende Verweise in anderen Office 2000-Anwendungen, wie Excel 2000 oder Word 2000, auf eine Access-Datenbank können Sie dort auch über die Prozeduren der Access-Datenbank verfügen. Ein globales Standardmodul wird im Datenbankfenster durch das links nebenstehende Symbol gekennzeichnet.

- **Globale Klassenmodule**. Klassenmodule ermöglichen es, benutzerdefinierte Objekte zu erstellen. Die Prozeduren des Klassenmoduls werden zu Methoden und Eigenschaften des Objekts. Alle Formular- und Berichtsmodule sind Klassenmodule, jedoch stehen deren Prozeduren nicht global, sondern lediglich formular- und berichtsspezifisch zur Verfügung. Auf die Prozeduren eines globalen Klassenmoduls kann dagegen von jeder anderen Prozedur aus zugegriffen werden. Globale Klassenmodule wurden mit der Version Access 97 neu eingeführt. Sie werden zusammen mit den globalen Standardmodulen auf der Registerkarte *Module* des Datenbankfensters angezeigt und können wie diese geöffnet werden. Ein globales Klassenmodul wird im Datenbankfenster durch das links nebenstehende Symbol gekennzeichnet.

Ein Modul enthält außer den Prozeduren auch einen Deklarationsbereich. Dieser weist standardmäßig für ein neu erstelltes Modul die Anweisung

```
Option Compare Database
```

auf. Der Deklarationsbereich wird, wenn Sie ein Modul öffnen, zunächst im Modulfenster angezeigt. Um die inhaltliche Bedeutung dieses Bereichs brauchen Sie sich für den Anfang nicht besonders zu kümmern, weil alle Prozeduren prinzipiell laufen, ohne daß Sie Angaben in diesem Bereich gemacht haben. Für fortgeschrittene Programmiertechniken sollten Sie aber die Möglichkeiten der Deklarationen im Deklarationsbereich beachten, vgl. vor allem Kap. 30, *Programmieren mit VBA*, Punkt 30.2, *Variablen und Konstanten*.

29.4.2 Visual Basic-Editor

Visual Basic-Editor aufrufen und zu Excel zurückkehren

Sämtliche Arbeiten im Zusammenhang mit VBA – das sind vor allem Erstellen und Bearbeiten von VBA-Code und Formularen, Testen derselben, Aufrufen der VBA-Hilfe, allgemeine Projektverwaltung, Nutzen des Objektkatalogs – finden im Visual Basic-Editor statt, den Sie von Access aus explizit aufrufen müssen.

Visual Basic-Editor aufrufen

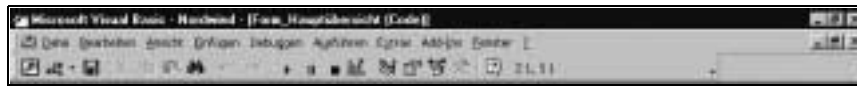


Bild 29.2: Menü- und Symbolleiste des Visual Basic-Editors

Sie rufen den Visual Basic-Editor auf mehrere verschiedene Weisen auf:

- Wählen Sie im Menü *Extras* den Befehl *Makro*, Unterbefehl *Visual Basic-Editor*.

  Oder:
- Drücken Sie die Tastenkombination [Alt]+[F11].

  Oder:
- Klicken Sie in der Task-Leiste von Windows auf das Anwendungssymbol des Visual Basic-Editor, wenn dieser bereits geöffnet wurde, Sie aber inzwischen eine andere Anwendung (z.B. Access) aktiviert hatten.

  Oder:
- Aktivieren Sie im Datenbankfenster die Objektliste *Module* und öffnen Sie ein Modul, indem Sie auf eines der Symbole *Entwurf* oder *Neu* klicken.

  Oder:

- Klicken Sie in der Entwurfsansicht eines Formulars oder Berichts auf das Symbol *Code* (vgl. links nebenstehend oben) oder öffnen Sie den Code-Generator mittels des Symbols Generator (vgl. links nebenstehend unten)

Wenn der Visual Basic-Editor aktiv ist, stehen Ihnen nur dessen Leistungen zur Verfügung, während die allgemeinen Access-Leistungen nicht verfügbar sind. Die Menü- und Symbolleiste des Visual Basic-Editors ist in Bild 29.2 wiedergegeben.

Zur Bedienung des Visual Basic-Editors stehen Ihnen, wie üblich, diverse Befehle, Tastenkombinationen, Menüs usw. zur Verfügung. Aus räumlichen Gründen können diese hier nicht in allen Einzelheiten behandelt werden. Einen vollständigen und systematischen Überblick mit jeweiligen Erklärungen erhalten sie jedoch in der VBA-Hilfe, vgl. Bild 29.3. Verfahren Sie wie folgt, um das Inhaltsverzeichnis dieser Hilfe zu aktivieren:

Bild 29.3: Inhaltsverzeichnis der VBA-Hilfe

- Rufen Sie ggf. den Visual Basic-Editor auf, z.B. mit [Alt]+[F11].
- Wählen Sie im Hilfe-Menü den Befehl *Microsoft Visual Basic-Hilfe*.
- Schlagen Sie das Thema *Hilfe zur Visual Basic-Benutzeroberfläche* auf, vgl. Bild 29.3, und informieren Sie sich in einem der Unterthemen, die z.T. weitere Unterthemen aufweisen.

Vom Visual Basic-Editor zu Access zurückkehren: Um zum Access-Fenster zurückzukommen, haben Sie vier Möglichkeiten:

- Drücken Sie [Alt]+[F11].

  Oder:

- Klicken sie auf das Symbol *Ansicht Microsoft Access* (vgl. links nebenstehend).

  Oder:

- Klicken Sie in der Task-Leiste von Windows auf das Symbol der Access-Anwendung.

Oder:

- Wählen Sie im Menü *Datei* den Befehl *Schließen und zurück zu Microsoft Access*.

  Oder:

- Klicken Sie mit der rechten Maustaste im Projekt-Explorer auf ein Klassenobjekt, um das Kontextmenü zu öffnen, und wählen Sie den Befehl *Objekt anzeigen*. Dann wird zu Access gewechselt und das/der entsprechende Formular/Bericht in der Entwurfsansicht angezeigt.

Die ersten drei Möglichkeiten lassen den Visual Basic-Editor weiter geöffnet, die letzte schließt ihn. Bei keinem der Wechsel wird gespeichert, es geht aber auch keine mit dem Visual Basic-Editor vorgenommene Änderung verloren, denn das gesamte VBA ist Bestandteil der Arbeitsmappendatei, die ja weiterhin geöffnet ist.

Projekt-Explorer

Mit dem Projekt-Explorer, der hierarchisch organisiert ist wie der Windows-Explorer, navigieren Sie zwischen den verschiedenen Objekten eines Projekts und den unterschiedlichen geöffneten Projekten. Sie öffnen das Fenster des Projekt-Explorers, indem Sie auf das links nebenstehende Symbol *Projekt-Explorer* klicken oder im Menü *Ansicht* den gleichlautenden Befehl wählen. Das Fenster des Projekt-Explorers ist oben in Bild 29.1 wiedergegeben. Sie schließen den Projekt-Explorer durch Klicken auf das Symbol *Schließen* am rechten Rand seiner Titelleiste oder mit dem Befehl *Ausblenden*, den Sie in seinem Kontextmenü finden.

Verankern des Projekt-Explorers. Sie können das Fenster des Projekt-Explorers wie eine Symbolleiste an einem der vier Ränder des Fensters vom Visual Basic-Editor verankern, indem Sie die Titelleiste über den entsprechenden Fensterrand hinauszuziehen versuchen. Außerdem gilt: Wenn Sie auf die Titelleiste des Projekt-Explorers doppelklicken, wird er am Rand des Anwendungsfensters verankert, sofern er zuvor in einem nicht verankerten Fenster dargestellt wurde. Umgekehrt: Wenn er zuvor verankert war, nimmt er durch Doppelklicken auf die Titelleiste wieder die alte Fensterposition ein.

Code-Fenster

Code-Fenster für ein Modul öffnen: Im Code-Fenster bearbeiten Sie den VBA-Code für bestimmte Module. Um den Code eines bestimmten Moduls im Code-Fenster anzuzeigen, gehen Sie folgendermaßen vor:

- Stellen Sie sicher, daß das Objekt, dessen Code Sie im Code-Fenster wiedergeben wollen, im Projekt-Explorer angezeigt wird.

- Markieren Sie das betreffende Objekt und klicken Sie dann auf das Symbol *Code anzeigen* (vgl. links nebenstehend) in der Symbolleiste des Projekt-Explorers.

Bild 29.4: Code-Fenster für das Klassenobjekt »Form_Artikel« aus dem Projekt »Nordwind«

Oder:

- Doppelklicken Sie im Projekt-Explorer auf das betreffende Modul.

Oder:

- Sie können auch mit der rechten Maustaste auf das Objekt klicken und im Kontextmenü den Befehl *Code anzeigen* wählen.

Geöffnetes Code-Fenster anzeigen

Wenn Sie mehrere Code-Fenster geöffnet haben, wird im allgemeinen mindestens eines davon durch das oder die andere(n) verdeckt sein. Wenn Sie das Menü *Fenster* aufschlagen, werden alle geöffneten Code-Fenster angezeigt. Wählen Sie eines davon, um es im Vordergrund anzuzeigen. Alternativ: Doppelklicken Sie im Projekt-Explorer auf das betreffende Modul.

Code-Fenster für ein Modul schließen

Klicken Sie auf die Schaltfläche *Schließen* des betreffenden Code-Fensters.

Eigenschaftenfenster

Das Eigenschaftenfenster listet für das jeweils markierte Objekt die Eigenschaften und deren aktuelle Einstellungen auf. Es funktioniert kontextabhängig in dem Sinne, daß es stets nur die Eigenschaften anbietet, die für das markierte Objekt verfügbar sind. Sie können die Einstellungen dieser Eigenschaften ändern.

Sie arbeiten folgendermaßen mit dem Eigenschaftenfenster:

- Klicken Sie in der Symbolleiste des Visual Basic-Editors auf das Symbol *Eigenschaftenfenster*, vgl. links nebenstehend, oder drücken Sie die Taste F4, um das Eigenschaftenfenster anzuzeigen. Durch Klicken auf die Schaltfläche *Schließen* in seiner Titelleiste schließen Sie es ggf. wieder.
- Wählen Sie das Objekt aus, dessen Eigenschaften angezeigt werden sollen.
- Klicken Sie auf die Registerkarte *Alphabetisch*, um Eigenschaften in alphabetischer Reihenfolge anzuzeigen, oder klicken Sie auf die Registerkarte *Nach Kategorien*, um Objekteigenschaften nach Kategorien geordnet anzuzeigen.

Verankern des Eigenschaftenfensters. Sie können das Eigenschaftenfenster wie eine Symbolleiste an einem der vier Ränder des Fensters vom Visual Basic-Editor verankern, indem Sie die Titelleiste über den entsprechenden Fensterrand hinauszuziehen versuchen. Außerdem gilt: Wenn Sie auf die Titelleiste des Eigenschaftenfensters doppelklicken, wird es am Rand des Anwendungsfensters verankert, sofern es zuvor in einem nicht verankerten Fenster dargestellt wurde. Umgekehrt: Wenn das Eigenschaftenfenster zuvor verankert war, nimmt es durch Doppelklicken auf die Titelleiste wieder die alte Fensterposition ein.

Objektkatalog

Der Vollständigkeit halber sei an dieser Stelle auch auf den Objektkatalog hingewiesen, ohne ihn hier inhaltlich zu beschreiben. Dies geschieht im einzelnen in Kap. 30, *Programmieren mit VBA*, Punkt 30.1.6, *Sich über Objekte, Methoden und Eigenschaften informieren*. Einen Anwendungsfall für den zweckmäßigen Gebrauch des Objektkatalogs können Sie auch weiter unten in diesem Kapitel kennenlernen, vgl. Punkt 29.4.3, *Code im Code-Fenster bearbeiten*, Unterpunkt *Prozeduren gezielt auffinden*.

Sie öffnen und schließen den Objektkatalog auf die folgende Weise:

- Klicken Sie in der Symbolleiste des Visual Basic-Editors auf das Symbol *Objektkatalog*, vgl. links nebenstehend, oder drücken Sie die Taste F2, um den Objektkatalog zu öffnen.
- Sie schließen sein Fenster wieder durch Klicken auf das Symbol *Schließen*.

29.4.3 Code im Code-Fenster bearbeiten

Bearbeitungsmöglichkeiten

Sie bearbeiten Code im Fenster eines Modulblatts prinzipiell wie Text in einem Textverarbeitungsprogramm. Außer den Grundfunktionen wie Löschen, Einfügen, Kopieren oder Verschieben von Codeteilen stehen Ihnen vor allem die folgenden weiteren Bearbeitungsmöglichkeiten zur Verfügung:

Suchen und Ersetzen von Code

Mit den gleichnamigen Befehlen aus dem Menü *Bearbeiten* können Sie Codeteile suchen und ersetzen, wobei Sie über die entsprechenden Dialogfelder den Bearbeitungsbereich gezielt steuern können.

Code als Text aus der Zwischenablage einfügen

Sie können Texte, die Sie (teilweise) als Code verwenden wollen, aus beliebigen Quellen verwenden, indem Sie diese in die Zwischenablage kopieren und im Modulfenster von dort einfügen. Beachten Sie dabei insbesondere die zahlreichen Beispiele der Online-Hilfe, die Sie in die Zwischenablage kopieren und von dort abrufen können.

Code-Zeile auf mehrere Zeilen verteilen

Um eine bessere Lesbarkeit und Bearbeitungsmöglichkeit zu erhalten, können Sie eine Code-Zeile, die ansonsten zu lang würde, auf mehrere Zeilen umbrechen. Der Umbruch muß am Ende einer syntaktischen Einheit, wie z.B. einem Schlüsselwort, einem Objektnamen etc. erfolgen. Um eine Zeile zu umbrechen, fügen Sie an der Umbruchstelle ein Leerzeichen und ein Unterstreichungszeichen _ an und setzen die Code-Zeile in einer neuen Textzeile fort. Das folgende Beispiel verteilt eine Code-Zeile auf zwei Textzeilen:

```
DoCmd.OutputTo acOutputReport, _
  "Alphabetische Artikelliste", acFormatHTML, _
  "Artikel.htm", True, "Nwindtem.htm"
```

Prozeduren gezielt auffinden

Um die Einfügemarke gezielt zu einer Prozedurdefinition zu bewegen, so daß Sie diese betrachten und ggf. bearbeiten können, stehen Ihnen zwei Wege zur Verfügung:

Sprung zur Prozedurdefinition. Wenn ein Prozedurname in einer beliebigen Code-Zeile angeführt wird – im allgemeinen, um diese aufzurufen –, können Sie auf die folgende Weise zur Definition dieser Prozedur springen:

- Setzen Sie die Einfügemarke in den Prozedurnamen oder markieren Sie den ganzen Prozedurnamen.
- Wählen Sie den Befehl *Definition* aus dem Menü *Ansicht*, oder drücken Sie die Tastenkombination ⇧+F2.

Prozedurdefinition im Objektkatalog wählen. Der Objektkatalog enthält alle Prozeduren des geöffneten Projekts. Um darin eine bestimmte Prozedur zu finden und zu deren Definition zu blättern, verfahren Sie wie folgt:

- Klicken Sie in der Symbolleiste des Visual Basic-Editors auf das Symbol *Objektkatalog*, vgl. links nebenstehend, oder drücken Sie die Taste F2, um den Objektkatalog zu öffnen, vgl. Bild 29.5.
- Wählen Sie aus der Dropdown-Liste links oben die geeignete Bibliothek aus. In Bild 29.5 wurde das Projekt *Nordwind* gewählt.
- Markieren Sie in der Liste *Klassen* das Modul, welches die gesuchte Prozedurdefinition enthält. In Bild 29.5 wurde das Modul *Start* gewählt.
- Doppelklicken Sie in der Liste *Elemente von (Klassenname)* auf die gesuchte Prozedur. Dann wird das Code-Fenster mit dem gewählten Modul angezeigt, und der Cursor befindet sich an der Stelle der doppelgeklickten Prozedur, vgl. das Code-Fenster in Bild 29.5.

Bild 29.5: Fenster »Objektkatalog« und Code-Fenster mit dem Modul »Start«, in dessen Prozedur »HideStartupForm« sich die Einfügemarke befindet.

Kommentare einfügen

Sie können an beliebiger Stelle Kommentarzeilen einfügen. Kommentare können auch am Ende einer Codezeile, die selbst keinen Kommentarcharakter hat, angefügt werden. Ein Kommentar beginnt mit dem Zeichen ' (Hochkomma). Eine Kommentarzeile können Sie auch mit der Anweisung *Rem* (für *Remark*) beginnen. Von den folgenden drei Code-Zeilen sind die beiden ersten vollständige Kommentarzeilen, in der dritten beginnt der Kommentar an der Stelle des Hochkommas.

```
'Dies ist die erste Kommentarzeile
Rem Dies ist die zweite Kommentarzeile
Sub GanzzahlTest() 'Prüfung auf Ganzzahligkeit
```

Standardmäßig gibt der Visual Basic-Editor Kommentare in grüner Farbe wieder. Im Unterpunkt *Optionen für den Visual Basic-Editor einstellen* (vgl. weiter unten in diesem Punkt 29.4.3) erfahren Sie, wie Sie dafür eine andere Farbe festlegen können.

Hilfe im Code-Fenster nutzen

Allgemeine VBA-Hilfe. Sie haben folgende Möglichkeiten, um Hilfe zu VBA anzufordern:

- Drücken Sie, während ein Modulfenster aktiv ist, die Taste F1. Dann wird das Hilfe-Fenster zu VBA angezeigt. Sie können direkt auf die Hilfe zu einer Methode, Eigenschaft, Funktion, Anweisung oder einem Objekt zugreifen, indem Sie auf das Schlüsselwort in einer Codezeile klicken und dann F1 drücken.
- Mit Hilfe des Objektkatalogs können Sie Informationen zu Objekten, Eigenschaften, Methoden und Konstanten anzeigen. Drücken Sie F1 im Objektkatalog. Dann wird Hilfe zu dem gerade markierten Begriff angezeigt.
- Verwenden Sie die Registerkarten *Inhalt*, *Antwort-Assistent* und *Index* der allgemeinen VBA-Hilfe.

Automatische QuickInfo. Damit die *Automatische QuickInfo* aktiv ist, muß das gleichnamige Kontrollkästchen auf der Registerkarte *Editor* im Dialogfeld *Optionen* des gleichnamigen Befehls aus dem Menü *Extras* aktiviert sein. Dann funktioniert diese Hilfe folgendermaßen: Wenn Sie im Modulfenster Code schreiben, wird automatisch eine Hilfe zu dem gerade eingegebenen Schlüsselwort angezeigt, sobald das Schlüsselwort durch eine öffnende Klammer, ein Leerzeichen, einen Punkt oder ein entsprechendes Zeichen abgeschlossen und damit für den Editor erkennbar geworden ist.

Bild 29.6: »Automatische QuickInfo«, hier zum Schlüsselwort »MsgBox«

In Bild 29.6 ist zu sehen, daß gerade die Zeichenfolge *msgbox(* eingegeben wurde. Daraufhin hat der Editor automatisch die dargestellte Hilfe angezeigt. Das nächste anzugebende Argument ist *Prompt*, was an der fetten Schrift, mit der dieses Argument wiedergegeben ist, zu erkennen ist. Wenn dieses Argument eingegeben ist, wird das nächste in der Argumentliste fett hervorgehoben, vgl. Bild 29.7.

Bild 29.7: »Automatische QuickInfo«: Das erste Argument zum Schlüsselwort »MsgBox« ist eingegeben und durch ein Komma begrenzt. Daraufhin wird das nächste anzugebende Argument fett hervorgehoben.

Elemente automatisch auflisten. Damit diese Hilfe funktioniert, muß das Kontrollkästchen *Elemente automatisch auflisten* auf der Registerkarte *Editor* im Dialogfeld *Optionen* des gleichnamigen Befehls aus dem Menü *Extras* aktiviert sein. Dann arbeitet diese Hilfe folgendermaßen: Wenn Sie ein Schlüsselwort geschrieben haben (z.B. eine Methode oder eine Anweisung) und in der Liste der Argumente ein Argument angeben können, das einer Liste vordefinierter Elemente entnommen werden kann, wird diese Liste als Listenfeld, aus dem Sie das geeignete Element wählen können, eingeblendet. In Bild 29.7 beispielsweise wurde das erste Argument für die *MsgBox*-Anweisung bereits eingegeben und durch ein Komma abgeschlossen. Das nächste anzugebende Argument *Buttons* verlangt die Angabe einer Zahl oder einer vordefinierten Konstanten in Textform, z.B. *vbOKCancel* oder *vbYesNo*. Die automatische Direkthilfe hat die Liste der möglichen Konstanten eingeblendet, sobald das vorangehende Argument mit einem Komma abgeschlossen war. In der eingeblendeten Elementliste, die auch gerollt werden kann, braucht nur das gewünschte Element markiert zu werden. Dann wird es in die Codezeile an der richtigen Stelle eingefügt.

Optionen für den Visual Basic-Editor einstellen

Wenn Sie das Dialogfeld *Optionen* des gleichnamigen Befehls aus dem Menü *Extras* aufschlagen, finden Sie darin vier Registerkarten, mit denen Sie diverse Optionen für das Arbeiten im Visual Basic-Editor einstellen können. In Bild 29.9 ist die Registerkarte *Editorformat* aktiviert, mit der Sie Schriftmerkmale für den Code im Code-Fenster festlegen können. Die ande-

ren Registerkarten enthalten außer weiteren Optionen, die sich auf das Code-Fenster beziehen, auch solche, die andere Teile des Visual Basic-Editors betreffen.

Bild 29.8: Automatisch eingeblendete Liste mit möglichen Konstanten für das Argument »Buttons« zur Funktion »MsgBox«

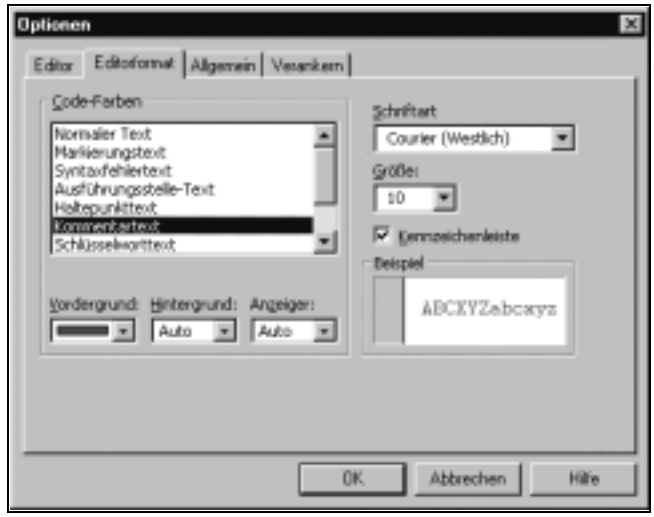


Bild 29.9: Registerkarte »Editorformat« im Dialogfeld »Optionen« des gleichnamigen Befehls aus dem Menü »Extras«

Module als Textdateien speichern oder laden

Der VBA-Code wird als ASCII-Code gespeichert. Daher ist es möglich, den Inhalt eines Moduls als Textdatei zu speichern oder zu laden. Dies geschieht bei geöffnetem und aktiviertem Modul mit den Befehlen *Datei exportieren*

bzw. Datei importieren aus dem Menü *Datei* bzw. *Kontextmenü*. Wenn Sie ein Modul als Textdatei speichern, haben Sie gleichzeitig die Möglichkeit, diese mit jedem Texteditor zu bearbeiten. Ferner bietet dies eine gute Möglichkeit, Ihre Funktionen und Sub-Prozeduren schriftlich zu dokumentieren und die Dokumentation ggf. in andere Dokumente einzubeziehen.

29.5 VBA-Prozedur erstellen und bearbeiten

29.5.1 Prozedur in globalem Modul erstellen

Um eine VBA-Prozedur (Funktion oder Sub-Prozedur) in einem globalen Modul zu erstellen, gehen Sie folgendermaßen vor (zum inhaltlichen Aufbau von Funktionen und Sub-Prozeduren vgl. Punkt 29.7, *Funktionen und Sub-Prozeduren*):

- Öffnen Sie ggf. aus dem Datenbankfenster oder dem Visual Basic-Editor heraus ein bestehendes Modul oder erstellen Sie ein neues.
- Klicken Sie auf die Symbol-Schaltfläche *Prozedur einfügen* (diese befindet sich in einer Dropdown-Liste, die auch die Schaltflächen *Modul einfügen* und *Klassenmodul einfügen* enthält, so daß es sein kann, daß Sie zunächst die Dropdown-Liste aufschlagen müssen), oder wählen Sie den Befehl *Prozedur* aus dem Menü *Einfügen*. Dann wird das Dialogfeld *Prozedur hinzufügen* angezeigt, vgl. Bild 29.10.

Bild 29.10: Dialogfeld »Prozedur hinzufügen«

- Wählen Sie im Dialogfeld *Prozedur hinzufügen* den Typ *Function* oder *Sub*, und geben Sie einen Namen für die Funktion bzw. Sub-Prozedur an, vgl. Bild 29.10, wo als Funktionsname *Fälligkeit* eingegeben ist. Bestätigen Sie das Dialogfeld mit OK.

- Nach Bestätigung des Dialogfeldes erzeugt Access die Anfangs- und Endzeile einer Prozedur sowie eine leere dazwischenliegende Zeile, in die Sie Ihre erste inhaltliche Programmzeile schreiben können.
- Weitere Programmzeilen erzeugen Sie durch Drücken der [↵]-Taste. Im allgemeinen werden Sie für jede weitere VBA-Anweisung eine neue Zeile verwenden, weil dies zur Übersichtlichkeit Ihres Codes beiträgt. Jedoch ist es auch möglich, mehrere Anweisungen in derselben Zeile aufzuschreiben, indem Sie zwischen den einzelnen Anweisungen einen Doppelpunkt eingeben. Dies kann sich beispielsweise für kurze *For-Next*-Schleifen empfehlen.
- Nachdem Sie eine Sub-Prozedur oder Funktion in ein globales Modul geschrieben haben, können Sie sofort darüber verfügen, im Modul, im Testfenster sowie auf allen anderen Access-Ebenen, wo dies der Sache nach möglich ist. Zwar müssen neue oder geänderte Prozeduren kompiliert werden, bevor sie ausgeführt werden können, jedoch besorgt Access dies ggf. automatisch. Wenn Sie allerdings sicher sein wollen, daß Sie mögliche Kompilierfehler schon vor dem Ausführen einer Prozedur entdecken und beheben wollen, was empfehlenswert ist, führen Sie den Befehl *Kompilieren von (Modulname)* aus dem Menü *Debuggen* aus. Falls Kompilierfehler auftreten, meldet Access dies und markiert die entsprechende Codestelle im Modulfenster.
- Um die neue oder geänderte Prozedur dauerhaft zu sichern, müssen Sie das Modul, in dem sie steht, speichern, z.B. mit [Strg]+[S] oder dem Befehl *Speichern* aus dem Menü *Datei*. Sie werden aber auch beim Schließen des Modulfensters zum Speichern aufgefordert, falls noch ungesicherte Änderungen vorliegen.

Statt eine neue Prozedur mit dem Befehl *Prozedur* aus dem Menü *Einfügen* oder mit der entsprechenden Symbol-Schaltfläche anzufordern, können Sie diese auch durch direkte Eingabe in das Modulfenster erstellen: Schreiben Sie in eine neue Zeile außerhalb einer bestehenden Prozedur

```
Function Funktionsname()
```

für eine neue Funktion und

```
Sub Prozedurname()
```

für eine neue Sub-Prozedur. Sobald Sie die Zeile mit dem Cursor verlassen oder, während der Cursor am Zeilenende steht, sobald Sie die [↵]-Taste drücken, ergänzt Access die zur Funktion gehörende Endzeile in der Form

```
End Function
```

für eine Funktion und

```
End Sub
```

für eine Sub-Prozedur.

Beispiel

Aufgabe. Erstellen Sie eine Funktion, die das Datum ausgibt, welches 14 Tage nach dem Tagesdatum liegt. Gehen Sie dazu folgendermaßen vor:

- Klicken Sie auf die Symbol-Schaltfläche *Prozedur einfügen*, oder wählen Sie den Befehl *Prozedur* aus dem Menü *Einfügen*.
- Wählen Sie im Dialogfeld *Prozedur hinzufügen* den Typ *Function*, und geben Sie den Namen *Fälligkeit* für die Funktion an, vgl. oben Bild 29.10. Lassen Sie für den *Gültigkeitsbereich* die voreingestellte Einstellung auf *Public* stehen, damit auf die Funktion von allen anderen Modulen aus zugegriffen werden kann. Bestätigen Sie das Dialogfeld *Prozedur einfügen*. Access erstellt die Anfangs- und Endzeile für die neue Funktion und blendet sie auf dem Bildschirm zum weiteren Bearbeiten ein, vgl. Bild 29.11.

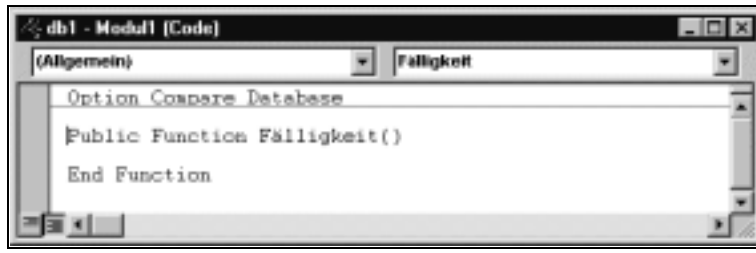


Bild 29.11: Erste und letzte Zeile der Funktion »Fälligkeit«, die mit dem Befehl Prozedur aus dem Menü Einfügen bzw. der entsprechenden Symbol-Schaltfläche in das Modulfenster eingefügt wurde.

- Geben Sie zwischen der ersten und der letzten Zeile der Funktion die folgende Codezeile ein:

```
Fälligkeit = Format(DateAdd("d", 14, Now), "dd.mm.yyyy")
```

- Testen Sie die Funktion *Fälligkeit* im Direktbereich des Testfensters (vgl. in diesem Kapitel Punkt 29.6, *Testen des Codes in Testfenstern*), und speichern Sie ggf. das Modul.

29.5.2 Ereignisprozedur erstellen

Im folgenden soll an einem sehr einfachen Beispiel gezeigt werden, wie Sie eine Ereignisprozedur erstellen.

Aufgabe. Erstellen Sie für die Eigenschaft *Nach Aktualisierung* des Textfeldes *GesamtkostenNetto* im Formular *Rechnung* eine Ereignisprozedur, die dafür sorgt, daß den Textfeldern *Mehrwertsteuer* und *GesamtkostenBrutto* nach der Aktualisierung des Textfeldes *GesamtkostenNetto* die richtigen Werte zugewiesen werden; vgl. die Textfelder im Formular von Bild 29.12.

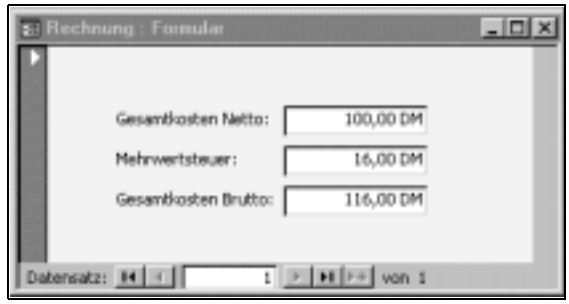


Bild 29.12: Formular »Rechnung« mit drei Textfeldern: Nachdem das Feld »Gesamtkosten-Netto« aktualisiert wurde, sollen die beiden anderen Felder automatisch neu berechnet werden.

Gehen Sie folgendermaßen vor:

- Öffnen Sie das Formular *Rechnung* in der Entwurfsansicht, vgl. Bild 29.13.
- Markieren Sie das Textfeld *GesamtkostenNetto*.
- Blenden Sie ggf. das Eigenschaftenfenster ein, und geben Sie darin der Eigenschaft *Nach Aktualisierung* den Fokus.

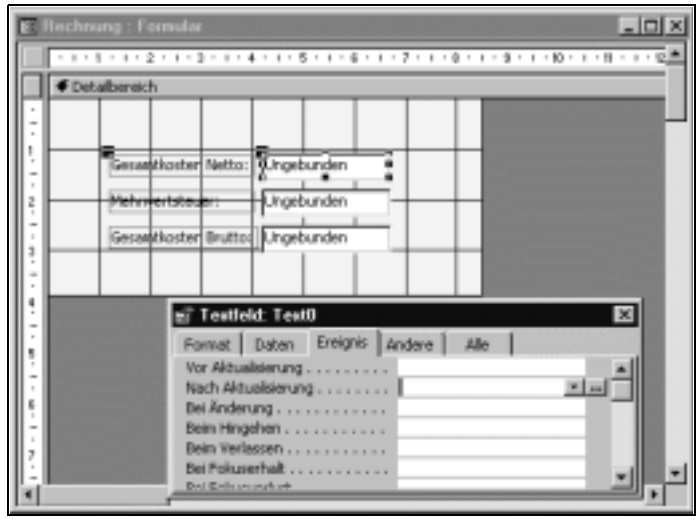


Bild 29.13: Formular »Rechnung« in der Entwurfsansicht. Das Textfeld »Gesamtkosten-Netto« ist markiert, und im Eigenschaftenfenster hat die Eigenschaft »Nach Aktualisierung« den Fokus.

- Klicken Sie auf die Symbol-Schaltfläche *Generator* am rechten Rand der Eigenschaft. Dann wird das Dialogfeld *Generator auswählen* eingeblendet.

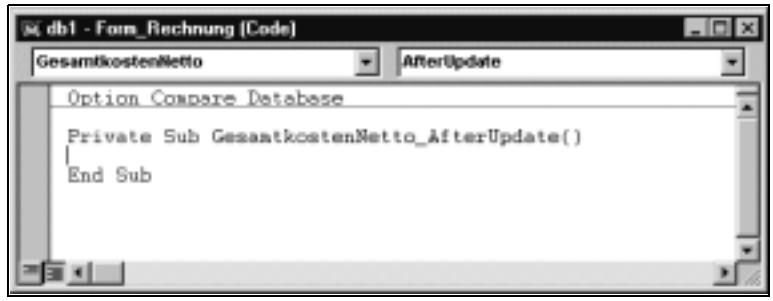


Bild 29.14: Modulfenster unmittelbar nach dem Öffnen des Formularmoduls mit dem Code-Generator für die Eigenschaft »Nach Aktualisierung« des Textfeldes »GesamtkostenNetto«.

- Wählen Sie im Dialogfeld *Generator* den *Code-Generator*. Dann wird der Visual Basic-Editor geöffnet und darin das Formularmodul (Klassenobjekt) mit dem Namen *Form_Rechnung* geöffnet. Access hat darin automatisch die erste und letzte Zeile einer VBA-Prozedur erstellt (vgl. Bild 29.14) und der Sub-Prozedur mit der ersten Zeile den Namen *GesamtkostenNetto_AfterUpdate* gegeben. Die letzte Zeile gibt das Prozedurende an. Zwischen die erste und letzte Zeile schreiben Sie Ihre Anweisungen.
- Es reichen zwei Anweisungen aus, um die Wertzuweisung an die beiden Textfelder *Mehrwertsteuer* und *GesamtkostenBrutto* vorzunehmen. Sie lauten:

```
[Mehrwertsteuer] = 0.16 * [GesamtkostenNetto]
[GesamtkostenBrutto] = [GesamtkostenNetto] _
   + [Mehrwertsteuer]
```

- Geben Sie diese beiden Codezeilen auf die folgende Weise ein: Schreiben Sie zunächst die erste Zeile zwischen die Anfangs- und Endzeile der Prozedur. Erzeugen Sie dann (oder bereits vorher, das ist gleichgültig) durch Drücken der [↵]-Taste wie in einem Textverarbeitungsprogramm eine leere Zeile, und schreiben Sie in diese die zweite Codezeile. Das Ergebnis sollte so aussehen, wie in Bild 29.15 zu sehen.

In Bild 29.15 ist die zweite neue Codezeile zusammenhängend wiedergegeben, während diese oben als Teil dieses Textes umbrochen dargestellt wurde. Ein derartiger Zeilenumbruch muß in diesem Buch des öfteren aus Platzgründen vorgenommen werden, während er im Code vielfach nicht erforderlich ist, weil das Code-Fenster mehr Platz bietet.

- Wechseln Sie zum Formularfenster und dort in die Formularansicht. Verändern Sie einen Wert im Feld *GesamtkostenNetto*, und geben Sie danach einem anderen Feld den Fokus. Dann sollten die beiden anderen Felder richtig berechnet werden.

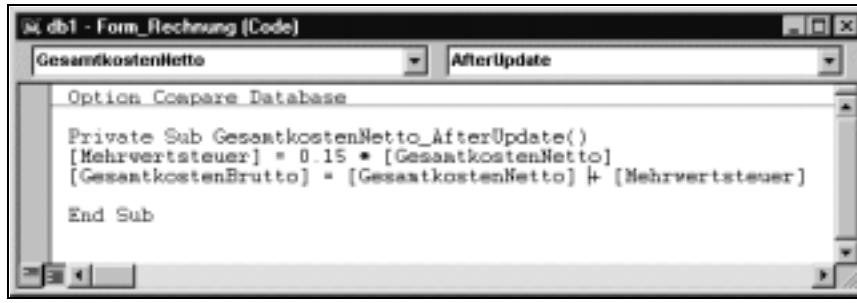


Bild 29.15: Fertige Ereignisprozedur »GesamtkostenNetto_AfterUpdate«

- Wechseln Sie wiederum in die Entwurfsansicht des Formulars. Im Eigenschaftenfenster sehen Sie jetzt, sofern das Textfeld *GesamtkostenNetto* noch markiert ist, für die Eigenschaft *Nach Aktualisierung* den Eintrag *[Ereignisprozedur]*. Access braucht hier – anders als bei einem Ereignismakro – keinen Namen anzugeben, weil dieser durch entsprechende Nomenklatur für jede Ereignisprozedur eindeutig festgelegt ist: Er setzt sich aus dem Namen des Steuerelements und dem (englischen) Namen der Eigenschaft zusammen: *GesamtkostenNetto_AfterUpdate*.

Bild 29.16: Access hat für die Eigenschaft »Nach Aktualisierung« des Textfeldes »GesamtkostenNetto« den Eintrag [Ereignisprozedur] vorgenommen. Wenn Sie in dieser Situation erneut auf die Symbol-Schaltfläche »Generator« klicken, wird das Formularmodul geöffnet und sofort die diesem Ereignis zugeordnete Prozedur zum Bearbeiten angezeigt.

29.5.3 Prozedur bearbeiten

Im Prinzip bearbeiten Sie den Code im Modulfenster wie einen Text in einem Textprogramm. Der Editor des Modulfensters bietet jedoch einige wichtige Werkzeuge, die auf die Besonderheiten von Codetext abgestellt sind:

Prozedur anzeigen lassen

Sie können den Text innerhalb einer Prozedur und zwischen Prozeduren rollen, letzteres jedoch nur, wenn *Vollständige Modulansicht* gewählt ist (vgl. die gleichnamige Symbol-Schaltfläche an linken Rand der waagerechten Bildlaufleiste). Sie können aber auch gezielt zu einer bestimmten Prozedur springen. Dazu stehen mehrere Möglichkeiten zur Verfügung:

Bild 29.17: Aufgeschlagene Prozedurliste im Modulfenster

- **Prozedurliste**. Am oberen Rand des Modulfensters befindet sich das Dropdown-Listenfeld *Prozedur*, das alle Prozeduren des Moduls in alphabetischer Reihenfolge auflistet, vgl. Bild 29.17. Wenn Sie aus der Liste eine Prozedur wählen, wird diese im Fenster zum Bearbeiten angezeigt.
- **Objektkatalog**. Das Menü ANSICHT enthält den Befehl OBJEKTKATALOG, der ein Dialogfeld zum Auswählen einer Prozedur, ggf. auch aus einem anderen als dem aktiven Modul, anzeigt. Mit diesem Dialogfeld läßt sich gezielter vorgehen als mit der im vorangehenden Aufzählungspunkt genannten Prozedurliste. Tastaturschlüssel für diesen Befehl: F2.

Bild 29.18: Dialogfeld »Objektkatalog« des gleichnamigen Befehls aus dem Menü »Ansicht«

- **Tasten für *Vorige Prozedur und Nächste Prozedur*.** Um zur vorigen bzw. nächsten Prozedur zu blättern, drücken Sie die Tastenkombination [Strg]+[↓] bzw. [Strg]+[↑].
- **Sprung zu einer im Codetext markierten Prozedur.** Wenn der Cursor im Namen einer Prozedur im Codetext steht, bewirkt das Drücken von [⇧]+[F2], daß diese Prozedur angezeigt wird, vgl. Bild 29.19.

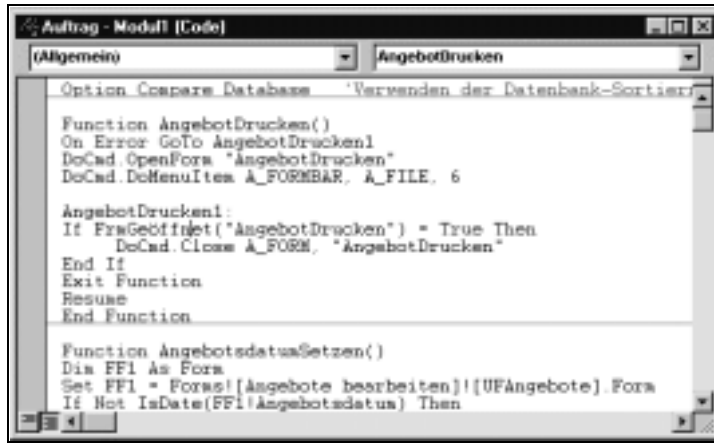


Bild 29.19: Der Cursor steht im Namen der in der sechsten Codezeile angesprochenen benutzerdefinierten Funktion »FrmGeöffnet«. Wenn Sie in dieser Situation die Taste [⇧]+[F2] drücken, wird im Modulfenster die Prozedur »FrmGeöffnet« angezeigt.

- **Prozedur suchen.** Mit dem Befehl *Suchen* aus dem Menü *Bearbeiten* können Sie nach beliebigen Textstellen im Code des aktuellen Moduls wie auch anderer Module suchen lassen, also auch nach einem Prozedurnamen.

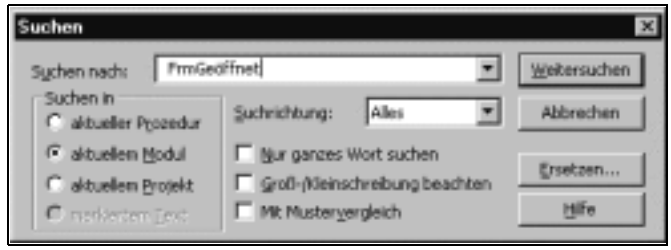


Bild 29.20: Dialogfeld »Suchen« des gleichnamigen Befehls aus dem Menü »Bearbeiten«. Hier würde nach der Zeichenkette »FrmGeöffnet« im aktuellen Modul gesucht werden.

29.5.4 Prozedur im Modulfenster ausführen

Es existieren verschiedene Möglichkeiten, eine Prozedur vom Modulfenster aus zu starten:

- Mit der Symbol-Schaltfläche *Sub/UserForm ausführen* oder dem gleichnamigen Befehl aus dem Menü *Ausführen* (auch Taste F5) starten Sie die Prozedur, in der aktuell der Cursor steht. Diese Art des Aufrufs ist allerdings auf Prozeduren ohne Argumente beschränkt, weil letztere dabei nicht übergeben werden können. Ferner gilt sie nicht für Ereignisprozeduren aus Formular/Berichtmodulen.
- Mit den vier Symbol-Schaltflächen *Einzelschritt*, *Prozedurschritt*, *Prozedur abschließen* und *Ausführen bis Cursor-Position*, die sich auch als gleichnamige Befehle im Menü *Debuggen* wählen lassen, können Sie Prozeduren schrittweise für Testzwecke laufen lassen.
- Prozeduren lassen sich auch aus dem Direktfenster heraus aufrufen. Dies stellt die umfassendste Möglichkeit dar, weil Sie dabei auch Argumente übergeben können, vgl. dazu im einzelnen weiter unten Punkt 29.6.1, *Direktfenster.*
- Prozeduren lassen sich von anderen Prozeduren aus aufrufen.

29.6 Testen des Codes in Testfenstern

Zum Testen des Codes stehen Ihnen drei besondere Fenster mit jeweils eigenen Leistungen zur Verfügung:

- Direktfenster
- Lokal-Fenster
- Überwachungsfenster

29.6.1 Direktfenster

Sie können das Direktfenster nutzen, um die Ergebnisse von VBA-Code zu überprüfen. Mit dem Direktfenster können Sie aber auch den Wert eines Steuerelements, Feldes oder einer Eigenschaft überprüfen, das Ergebnis eines Ausdrucks anzeigen oder einer Variablen, einem Feld oder einer Eigenschaft einen neuen Wert zuweisen. Das Direktfenster ist eine Art von aktivem Notizblockfenster, in dem Anweisungen, Methoden, Funktions- und Sub-Prozeduren sofort ausgeführt und ausgewertet werden.

Funktionen testen

Um beispielsweise das Ergebnis der benutzerdefinierten Funktion EinkSt1999(), zu der als obligatorisches Argument das Jahreseinkommen angegeben werden muß, im Direktfenster ausgeben zu lassen, verfahren Sie wie folgt (das Jahreseinkommen soll 80.000 DM betragen):

Bild 29.21: Das Direktfenster hat die benutzerdefinierte Funktion »EinkSt1999« ausgeführt und das Ergebnis ausgegeben.

- Blenden Sie ggf. das Testfenster ein, beispielsweise mit der Tastenkombination [Strg]+[G].
- Geben Sie in das Direktfenster ein Fragezeichen *?* (oder das Wort *print* mit nachfolgendem Leerzeichen) und dann den Namen der Funktion mit Klammern und Argumentwerten ein, hier also in der Form (vgl. Bild 29.21)

```
?EinkSt95(80000)
```

- Drücken Sie die [↵]-Taste, während der Cursor an einer beliebigen Stelle in der auszuwertenden Zeile des Testfensters blinkt. Dann wird die Funktion ausgewertet, und das Ergebnis wird im Direktbereich des Testfensters ausgegeben, vgl. Bild 29.21.

Sub-Prozeduren testen

Eine Sub-Prozedur können Sie auf die gleiche Weise wie eine Funktion vom Direktfenster aus aufrufen. Allerdings dürfen Sie dabei nicht das Wort *print* bzw. das Fragezeichen *?* voranstellen, weil Sub-Prozeduren keine Ergebnisse ausgeben. Ferner geben Sie, wie auch sonst beim Aufrufen von Sub-Prozeduren, kein Klammerpaar am Ende an. Prozedurargumente werden ebenfalls

in der Weise übergeben, wie auch sonst üblich: Führen Sie diese hinter dem Prozedurnamen ohne Klammerpaar an, bei mehr als einem Argument durch Kommas getrennt.

Mit der folgenden Zeile beispielsweise testen Sie die Sub-Prozedur *FrmÖffnen* mit dem Prozedurargument *Rechnung schreiben*:

```
FrmÖffnen "Rechnung schreiben"
```

Um eine in einem Klassenmodul definierte Prozedur im Direktbereich auszuführen, stellen Sie dem Prozedurnamen den Namen des Klassenmoduls und einen Punkt voran, es sei denn, das Modul wird gerade im Unterbrechungsmodus ausgeführt. Im Unterbrechungsmodus ist es nicht erforderlich, die Prozedur zu kennzeichnen, da das Modul dann im Gültigkeitsbereich ist. Beachten Sie, daß alle Formular- und Berichtsmodule Klassenmodule in diesem Sinne sind. Um beispielsweise die Prozedur *Form_Load* (Ereignisprozedur des Formularereignisses *Beim Laden*) des Formularmoduls *Form_Artikel* (das ist das Formularmodul des Formulars *Artikel*) vom Direktbereich des Testfensters aus zu starten, schreiben Sie in den Direktbereich die Zeile

```
Form_Artikel.Form_Load
```

und drücken die [↵]-Taste, während der Cursor in dieser Zeile blinkt.

Ausdrücke oder Anweisungen testen

Im Direktfenster können Sie auch Ausdrücke einschließlich eingebaute VBA-Funktionen oder -Anweisungen auswerten lassen. Beispielsweise können Sie die Zeile

```
? Date() +12
```

eingeben und mit der [↵]-Taste ausführen, um das Datum ausgeben zu lassen, das zwölf Tage nach dem aktuellen liegt, vgl. Bild 29.22.

Bild 29.22: Das aktuelle Datum in dieser Situation war der 31.07.1999

Selbst Schleifen können Sie auf diese Weise direkt testen, solange Sie dafür sorgen, daß all ihre Elemente in eine Zeile passen. Dafür müssen Sie den Doppelpunkt verwenden, um mehrere Anweisungen in eine Codezeile zu schreiben, vgl. Bild 29.23.

```
for i = 1 to 10 : print i*100 : next
```

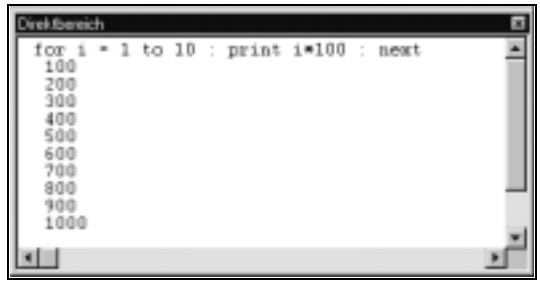


Bild 29.23: Wenn Sie, wie hier, mehrere Anweisungen auf einmal testen wollen, müssen Sie die einzelnen Anweisungen in eine Zeile schreiben und sie durch je einen Doppelpunkt trennen.

Ebenso gut können Sie beispielsweise auch einem Steuerelement eines (in der Formularansicht geöffneten) Formulars einen Wert zuweisen, wie in dem folgenden Beispiel:

```
Forms!Artikel!Einzelpreis = 5000
```

Wenn Sie diese Zeile im Direktbereich des Testfensters starten, wird dem Steuerelement *Einzelpreis* des Formulars *Artikel* der Wert 5000 zugewiesen, sofern das Formular in der Formularansicht geöffnet ist; andernfalls erfolgt eine Fehlermeldung. Beachten Sie, daß die Zeile ohne führendes Fragezeichen geschrieben ist, weil es sich um eine Anweisung handelt.

Im Direktbereich können Sie jeden Ausdruck auswerten lassen, der an beliebigen anderen Stellen von Access, auch außerhalb von VBA, ausgewertet werden könnte. Wenn Sie beispielsweise die folgende Zeile vom Direktbereich aus starten, erhalten Sie 16% des Wertes im Steuerelement *Einzelpreis* für den aktuellen Datensatz im (in der Formularansicht geöffneten) Formular *Artikel*:

```
?Forms!Artikel!Einzelpreis*0.16
```

Werte von Prozeduren aus in das Direktfenster ausgeben lassen

Das Direktfenster kann auch dazu verwendet werden, Werte von Prozeduren aus zum Testen direkt ausgeben zu lassen. Dazu dient die Methode *Print* des Objekts *Debug*.

Sie können in Ihren Code eine Anweisung mit der Syntax

Debug.Print [*Elemente*] [;]

einbauen. Dann wird der Wert des mit *Elemente* angegebenen Ausdrucks in das Direktfenster geschrieben.

Der folgende Programmausschnitt beispielsweise schreibt einen der beiden Sätze in den Direktbereich:

```
If A = C Eqv B = C Then
   Debug.Print "Die zwei Größen A und B sind gleich."
Else
   Debug.Print "Die zwei Größen A und B sind verschieden."
End If
```

29.6.2 Lokal-Fenster

Im Lokal-Fenster können Sie die Werte, die Variablen einer Prozedur während ihrer Ausführung in den verschiedenen Schritten der Prozedurausführung annehmen, simultan beobachten. Im Lokal-Fenster wird eine Variablenliste mit drei Spalten angezeigt: *Ausdruck*, *Wert* und *Typ*. Einige Variablen, z.B. benutzerdefinierte Typen, Datenfelder und Objekte, enthalten häufig hierarchische Informationen. Diese Variablen besitzen dann links neben dem Variablennamen eine Schaltfläche zum Reduzieren/Erweitern, mit der Sie die Anzeige der hierarchischen Informationen steuern können. Wenn das Lokal-Fenster angezeigt wird, wird es bei jedem Wechsel vom Ausführungsmodus in den Unterbrechungsmodus automatisch aktualisiert. Ein solcher Moduswechsel findet statt, wenn ein Haltepunkt erreicht wird oder Sie beginnen, den Code im Einzelschrittmodus auszuführen.

Die erste Variable in der Liste ist eine spezielle Modulvariable. Für ein Klassenmodul ist es die vom System definierte Variable *Me*. Sie stellt einen Objektverweis auf die aktuelle Instanz der Klasse dar, die vom aktuellen Modul definiert ist. Da die Variable ein Objektverweis ist, kann sie erweitert werden und zeigt dann alle Eigenschaften und Datenmitglieder der aktuellen Klasseninstanz an. Für Standardmodule stellt die erste Variable den Namen des aktuellen Moduls dar; sie kann ebenfalls erweitert werden, um alle Variablen auf Modulebene im aktuellen Modul anzuzeigen. Sie können den Wert einer Variablen im Lokal-Fenster ändern, jedoch nicht ihre Ausdruck- und Typ-Einstellungen.

Beispiel

In Bild 29.24 wird der Code einer Funktion mit dem Namen *EinkSt1999* wiedergegeben. Die Variablenwerte dieser Funktion sollen während ihrer Ausführung im Lokalbereich des Testfensters angezeigt werden. Als Argumentwert für das Einkommen soll 80000 angegeben werden. Verfahren Sie wie folgt, um diese Aufgabe zu lösen:

- Setzen Sie für eine Codezeile der Funktion einen Haltepunkt, indem Sie den Cursor in diese Zeile bringen und auf die Symbol-Schaltfläche *Haltepunkt ein/aus* (befindet sich standardmäßig in der Symbolleiste *Debuggen*) klicken, vgl. links nebenstehend. Eine Codezeile, in der sich ein Haltepunkt befindet, wird farblich hervorgehoben und in der linken Rand-

spalte (auch *Kennzeichenleiste* genannt) durch einen dicken Punkt gekennzeichnet, vgl. die zweite Codezeile der Funktion in Bild 29.24. An dieser Stelle wird die Ausführung der Funktion dann unterbrochen werden, und Sie können von dort an die Entwicklung der Variablen dieser Prozedur im Lokal-Fenster verfolgen.

- Blenden Sie das Lokal-Fenster und das Direktfenster ein.
- Geben Sie in das Direktfenster die Zeile

```
? EinkSt1999(80000)
```

ein, und drücken Sie die [↵]-Taste. Dann wird die Funktion gestartet, und die Ausführung wird an der Stelle des Haltepunktes unterbrochen, so daß sich das System im *Unterbrechungsmodus* befindet. Im Lokalbereich des Testfensters werden dann die Variablen der Funktions-Prozedur und ihre aktuellen Werte angezeigt. Im vorliegenden Beispiel, in der die Ausführung der Prozedur sofort in der zweiten Zeile angehalten wird, sind noch sämtliche Variablen leer.

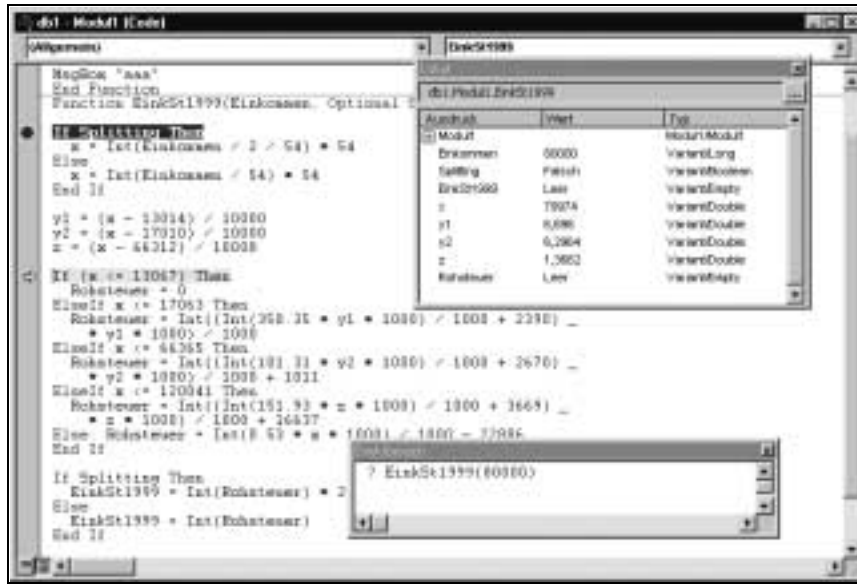


Bild 29.24: Code der Funktion »EinkSt1999« mit Haltepunkt in der zweiten Prozedurzeile. Die Funktion wird im Einzelschrittmodus ausgeführt.

- Führen Sie die Prozedur schrittweise aus, indem Sie Schritt für Schritt auf die Symbol-Schaltfläche *Einzelschritt* (Symbolleiste *Debuggen*) klicken, vgl. links nebenstehend. Dann werden im Lokal-Fenster die Variablenwerte für jeden weiteren Schritt aktualisiert. In Bild 29.24 beispielsweise ist eine Situation wiedergegeben, in der die Prozedur bis zu der Codezeile mit dem Code

```
If (x <= 13067) Then
```

ausgeführt wurde, diese Zeile ausgeschlossen. Daher gibt das Lokal-Fenster die Variablenwerte wieder, wie sie nach dem Ausführen der vorangehenden Codezeile

```
z = (x - 66312) / 10000
```

aktuell sind.

Wenn die Prozedur bis zum Ende ausgeführt oder die Ausführung abgebrochen worden ist, wird das Lokal-Fenster wieder gelöscht.

29.6.3 Überwachungsfenster

Im Überwachungsfenster können Sie den Wert eines Ausdrucks oder einer Variablen anzeigen, während Code ausgeführt wird:

- Um einen Überwachungsausdruck zu erstellen, wählen Sie im Menü *Debuggen* den Befehl *Überwachung hinzufügen*. Dann wird das Dialogfeld *Überwachung hinzufügen* angezeigt, vgl. Bild 29.25.

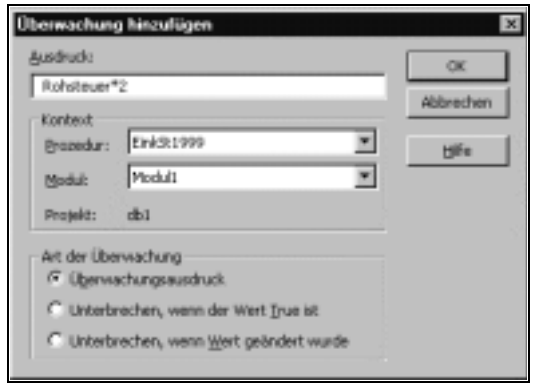


Bild 29.25: Dialogfeld »Überwachung hinzufügen«

- Geben Sie in das Eingabefeld *Ausdruck* einen auszuwertenden Ausdruck ein, und bestimmen Sie durch entsprechende Auswahl in den Dropdown-Listen *Prozedur* und *Modul* den Kontext, in dem der Ausdruck ausgewertet werden soll. Wählen Sie außerdem die gewünschte *Überwachungsart*. Beachten Sie, daß die Option *Unterbrechen, wenn der Wert True ist* nur sinnvoll ist, wenn Sie einen Ausdruck mit einem Vergleichsoperator angegeben haben. Bestätigen Sie das Dialogfeld mit OK.
- Wiederholen Sie die letzten beiden Schritte für jeden weiteren Ausdruck, der im Überwachungsfenster ausgewertet werden soll, denn dort können durchaus die Ergebnisse mehrerer Ausdrücke wiedergegeben werden.

- Setzen Sie im Code der zu testenden Prozedur an geeigneter(n) Stelle(n) einen oder mehrere Haltepunkte.
- Blenden Sie ggf. das Überwachungsfenster ein.
- Starten Sie die Prozedur aus dem Direktbereich des Testfensters heraus, und beobachten Sie die Ergebnisse im Überwachungsfenster, während Sie die Prozedur ggf. schrittweise ausführen lassen.

Das Dialogfeld *Aktuellen Wert anzeigen* zeigt den Wert einer markierten Variablen oder eines markierten Ausdrucks direkt an, während sich das System im Unterbrechungsmodus befindet: Markieren Sie eine Variable oder einen Ausdruck im Code, und drücken Sie die Tastenkombination `Umschalt`+`F9`, oder klicken Sie im Menü *Debuggen* auf *Aktuellen Wert anzeigen*, oder klicken Sie auf die gleichnamige Symbol-Schaltfläche, vgl. links nebenstehend.

Noch einfacher gestaltet sich die Überwachung eines Ausdrucks auf die folgende Weise: Zeigen Sie, während sich das System im Unterbrechungsmodus befindet, mit dem Mauszeiger auf eine Variable oder einen Ausdruck. Dann wird deren aktueller Wert direkt durch eine QuickInfo unterhalb des Mauszeigers angezeigt, vgl. Bild 29.26.

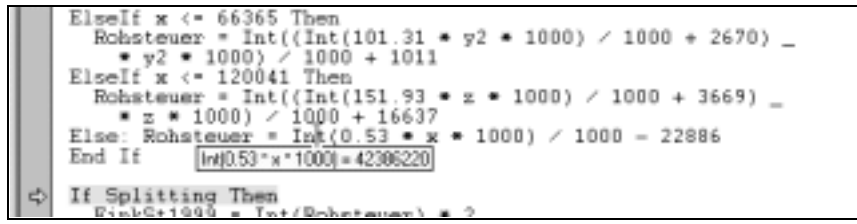


Bild 29.26: QuickInfo mit dem Wert des Ausdrucks »Int(0.53*x*1000)«, auf den der Mauszeiger zeigt

29.7 Funktionen und Sub-Prozeduren

29.7.1 Vergleich von Funktionen und Sub-Prozeduren

Funktionen und Sub-Prozeduren sind für viele Anwendungssituationen gleichwertig. Sie unterscheiden sich jedoch in den folgenden Punkten:

- Funktionen können einen Wert ausgeben, Sub-Prozeduren dagegen nicht.
- Funktionen können auf allen Access-Ebenen, z.B. in Formularen oder Berichten, aufgerufen werden, Sub-Prozeduren dagegen können nur von VBA-Prozeduren aufgerufen werden. Insbesondere gilt auch, daß Sie eine VBA-Funktion als Einstellung einer Ereigniseigenschaft angeben können. Auf diese Weise können Sie de facto Ereignisprozeduren erstellen, die Formular- oder Berichtübergreifend einsetzbar sind, weil sie in Standardmo-

dulen formuliert sind. Ereignisprozeduren in Formular- und Berichtsmodulen sind dagegen stets auf das eine Formular oder den einen Bericht beschränkt, in dessen Modul sie definiert sind.

Um eine Sub-Prozedur von einem Formular oder Bericht zu starten, können Sie allerdings eine Funktion schreiben, welche die Sub-Prozedur aufruft, und dann diese Funktion im Formular anführen.

Wenn Sie für eine Ereigniseigenschaft eine VBA-Funktion angeben wollen, müssen Sie dies in der in Bild 29.27 beispielhaft wiedergegebenen Weise vornehmen: Geben Sie den Funktionsnamen an, angeführt von einem Gleichheitszeichen und abgeschlossen mit einem Klammerpaar. Im Formular, das Bild 29.27 zugrunde liegt, wird die Funktion *FrmRechnungAktualisierungsPrüfung* aufgerufen werden, wenn das Formularereignis *Vor Aktualisierung* eintritt.

Bild 29.27: Eine VBA-Funktion als Einstellung einer Ereigniseigenschaft geben Sie mit führendem Gleichheitszeichen und endendem Klammerpaar an.

29.7.2 Funktionen: Grundlagen

Zulässige Funktionsnamen

Bei der Vergabe von Funktionsnamen sind Sie im Prinzip frei. Beachten Sie allerdings die Beschränkungen, daß ein Funktionsname

- mit einem Buchstaben beginnen muß,
- kein Leerzeichen, keinen Punkt (.), Ausrufezeichen (!) oder die Zeichen @, &, $, # enthalten darf,
- nicht länger als 255 Zeichen lang sein darf,
- einmalig sein muß, d.h., daß er kein reserviertes Wort sein und auch nicht zweimal vergeben werden darf. Die Einmaligkeit gilt auch gegenüber Namen von Sub-Prozeduren sowie allen Prozedurnamen in anderen Modulen derselben Datenbank. Namen von *privaten* Prozeduren müssen dagegen nur im selben Modul einmalig sein.

Diese Namensregeln gelten entsprechend auch für Namen von Sub-Prozeduren, Argumenten, Variablen und Konstanten.

VBA berücksichtigt die Groß-/Kleinschreibung nicht, behält jedoch die Schreibweise der Anweisung bei, mit der der Name deklariert wurde. Dies bedeutet, daß Sie einen Namen unter Mißachtung der Groß-/Kleinschreibung im Code schreiben dürfen, der Editor und andere VBA-Instanzen diesen jedoch unter Beachtung der bei der ursprünglichen Definition verwendeten Groß-/Kleinschreibung wiedergibt.

Funktionen, die einen Wert ausgeben

Funktionen können so geschrieben werden, daß sie einen Wert an die Stelle, die sie aufrufen, ausgeben. Da Funktionen andererseits jede Form von Anweisung ausführen können, so daß ihr Zweck nicht allein in der Ausgabe eines Wertes liegt, ist es vielfach sinnvoll, von Funktionen einfach gewisse Aktionen oder Berechnungen ausführen zu lassen, ohne daß es sinnvoll wäre, einen Wert auszugeben.

Um eine Funktion einen Wert ausgeben zu lassen, müssen Sie diesen Wert der Funktion selbst zuweisen. Die bereits oben als Beispiel verwendete Funktion *Fälligkeit* gibt das um 14 Tage nach dem aktuellen Datum liegende Datum an die Stelle aus, von der sie aufgerufen wird, weil der entsprechende Wert in der Funktion selbst zugewiesen wird:

```
Function Fälligkeit ()
Fälligkeit = Format(DateAdd("d", 14, Now), "dd.mm.yyyy")
End Function
```

Funktionen mit Argumenten

Sie können Funktionen schreiben, denen beim Aufrufen Werte als Argumente übergeben werden können, genau so, wie dies für die in Access eingebauten Funktionen gilt. Sie legen die Argumente fest, indem Sie diese zwischen dem Klammerpaar am Ende des Funktionsnamens als Argumentnamen angeben. Innerhalb der Funktion beziehen Sie sich dann auf die Argumentnamen.

Die Beispielfunktion *Fälligkeit* aus dem letzten Punkt nimmt als Ausgangspunkt zur Berechnung des um 14 Tage erhöhten Datumswertes das aktuelle Datum, das von der Funktion *Now* als zweiter Argumentwert an die Funktion *DateAdd* übergeben wird. Die Funktion soll so geändert werden, daß der Benutzer das Ausgangsdatum zusammen mit der Funktion angeben kann. Die neue Funktion *FälligkeitVariabel* sieht daher das Argument *Datumswert* vor, das in der Funktion *DateAdd* verarbeitet wird:

```
Function FälligkeitVariabel (Datumswert)
FälligkeitVariabel = Format(DateAdd("d", 14, Datumswert), _
  "dd.mm.yyyy")
End Function
```

Sie rufen diese Funktion beispielsweise im Direktfenster auf die folgende Weise auf:

Bild 29.28: Funktion »FälligkeitVariabel« wird mit übergebenem Argumentwert im Direktfenster getestet. Beachten Sie das US-amerikanische Datumsformat für die Eingabe und das deutsche für die Ausgabe!

Statt nur ein Argument können Sie mehrere vorsehen. Diese werden dann beim Definieren zwischen dem Klammerpaar durch je ein Komma voneinander getrennt.

Sie können für jedes Argument einen Datentyp angeben. Dies geschieht mit dem reservierten Wort *As* und darauffolgendem Datentyp. Die folgende Zeile definiert die Funktion *Gebühr1* mit den beiden angegebenen Argumenten. Das Argument *AnzahlPositionen* ist vom Typ *Integer*, das Argument *Gesamtfläche* vom Typ *Variant*, dem voreingestellten standardmäßigen Datentyp. Die Funktion akzeptiert für das Argument *AnzahlPositionen* nur Zahlen vom Typ *Integer*, andernfalls erfolgt eine Fehlermeldung.

```
Function Gebühr1 (AnzahlPositionen As Integer, Gesamtfläche)
```

Parameter-Datenfelder

Ein Parameter-Datenfeld kann dazu verwendet werden, ein Datenfeld mit Argumenten an eine Prozedur zu übergeben. Sie müssen die Anzahl der Elemente im Datenfeld bei der Definition der Prozedur nicht kennen.

Sie verwenden das Schlüsselwort *ParamArray* zur Kennzeichnung eines Parameter-Datenfeldes. Das Datenfeld muß als ein Datenfeld vom Typ *Variant* deklariert und das letzte Argument in der Prozedurdefinition sein.

Das folgende Beispiel, das (etwas umgeformt) der Online-Hilfe entnommen ist, zeigt, wie Sie eine Prozedur mit einem Parameter-Datenfeld definieren können:

```
Function BeliebigeAnzArg (strName As String, ParamArray intPunkte()
As Variant)
   Dim intI As Integer
   Debug.Print strName; "    Punkte"
   ' Verwenden Sie zum Bestimmen der oberen Grenze
   ' des Datenfeldes die Funktion Ubound.
   For intI = 0 To UBound(intPunkte())
```

```
            Debug.Print "          "; intPunkte(intI)
      Next intI
   End Function
```

Das folgende Beispiel zeigt, wie Sie diese Funktion aus dem Direktfenster aufrufen können:

```
?BeliebigeAnzArg("Mona", 10, 26, 32, 15, 22, 24, 16)
```

Oder:

```
?BeliebigeAnzArg("Lisa", "Höchste", "Niedrigste", "Mittlere")
```

Aufrufen benutzerdefinierter Funktionen

Benutzerdefinierte Funktionen werden genau so verwendet wie solche, die von Access vordefiniert sind: Das Klammerpaar muß hinter dem Funktionsnamen stets mit angeführt werden, auch dann, wenn die Funktion keine Argumente besitzt. Sofern Argumente definiert sind, müssen dafür Werte angegeben werden. Diese werden zusammen mit dem Funktionsnamen zwischen den Klammern angeführt, bei mehr als einem Argument durch je ein Trennungszeichen voneinander getrennt.

Wenn Sie eine benutzerdefinierte Funktion von einer VBA-Prozedur oder aus dem Direktfenster aufrufen, verwenden Sie als Trennzeichen das Komma, auf der allgemeinen Access-Ebene dagegen das Semikolon. Entsprechendes gilt für das Dezimaltrennzeichen: Verwenden Sie den Punkt, um einen numerischen Argumentwert innerhalb von VBA zu übergeben. Verwenden Sie dagegen das Komma als Dezimaltrennzeichen, wenn Sie die Funktion außerhalb von VBA – beispielsweise in einem Formular – verwenden.

Benannte Argumente

Beim Aufruf von Prozeduren können Sie Argumente außer in der Reihenfolge, in der sie in der Prozedurdefinition auftreten, auch über den Namen und unabhängig von der Position übergeben.

Die folgende Funktions-Prozedur benötigt z.B. drei Argumente:

```
Function Person(Nachname, Wohnort, Geburtsdatum)
   Debug.Print Nachname, Wohnort, Geburtsdatum
End Function
```

Sie können diese Prozedur aufrufen, indem Sie ihre Argumente mit der korrekten Position, jeweils getrennt durch ein Komma, angeben, z.B. in der Form

```
Dummy = Person("Hamann", "Hamburg", #11/2/63#)
```

oder in der Form

```
Person "Hamann", "Hamburg", #11/2/63#
```

Sie können diese Prozedur auch über die Angabe von benannten Argumenten, jeweils getrennt durch ein Komma, aufrufen, z.B. in der Form

```
Person Geburtsdatum := #11/2/63#,Nachname := _
  "Hamann", Wohnort := "Hamburg"
```

Ein benanntes Argument enthält einen Argumentnamen, gefolgt von einem Doppelpunkt, einem Gleichheitszeichen (:=) und dem Argumentwert.

Benannte Argumente sind besonders dann nützlich, wenn Sie eine Prozedur aufrufen, die über optionale Argumente verfügt. Wenn Sie benannte Argumente verwenden, müssen Sie keine Kommas für fehlende Argumente angeben. Mit benannten Argumenten können Sie auch leichter nachvollziehen, welche Argumente Sie übergeben und welche Sie nicht angegeben haben.

Optionale Argumente

Optionalen Argumenten geht das Schlüsselwort *Optional* in der Prozedurdefinition voraus. Sie können auch einen Standardwert für das optionale Argument in der Prozedurdefinition angeben. Beispiel:

```
Function EinkSt96ff(Einkommen, Optional Splitting = False)
...
...
End Function
```

Wenn Sie eine Prozedur mit einem optionalen Argument aufrufen, brauchen Sie das optionale Argument nicht anzugeben. Geben Sie das optionale Argument nicht an, so wird der Standardwert, falls definiert, verwendet.

29.7.3 Sub-Prozeduren: Grundlagen

Für Sub-Prozeduren gelten die gleichen Konventionen wie für Funktionen. Der wesentliche Unterschied zu Funktionen besteht darin, daß Sub-Prozeduren keinen Wert ausgeben und nur von VBA-Prozeduren aus aufgerufen werden können. (Ereignisprozeduren in Formular- und Berichtsmodulen werden allerdings direkt von Formularen und Berichten aus aufgerufen.)

Eine Sub-Prozedur wird durch die Prozedur-Anfangs- und Endzeile

```
Sub Sub-Prozedurname (Argumentliste)
...
...
End Sub
```

definiert.

Die folgende Sub-Prozedur ändert die Größe des aktiven Fensters auf die mit den beiden Argumenten angegebene Breite und Höhe.

```
Sub Größe (DokBreite, DokHöhe)
    DoCmd.MoveSize , , DokBreite, DokHöhe
End Sub
```

Sie rufen diese Prozedur aus einer anderen VBA-Prozedur oder im Direktfenster mit der folgenden Codezeile auf (dabei wird die Fensterbreite auf 6000 und die -höhe auf 5000 Twips (1 Twip = 1/20 Punkt) eingestellt):

```
Größe DokBreite:=6000, DokHöhe:=5000
```

29.7.4 Private Prozeduren

Standardmäßig gilt das folgende: Jede Prozedur kann jede andere Prozedur der Datenbank aufrufen – auch solche der anderen globalen Module. Sie können den Geltungsbereich einer Prozedur jedoch auf das Modul, in dem sie definiert ist, beschränken. Diese Beschränkung schließt auch den Aufruf der Prozedur von der allgemeinen Access-Ebene aus. Um eine Prozedur als Privat-Prozedur zu definieren, stellen Sie bei der Prozedurdefinition dem Wort *Function* bzw. *Sub* das reservierte Wort *Private* voran:

```
Private Function Funktionsname()
...
End Function

bzw.
Private Sub Sub-Prozedurname()
...
End Sub
```

29.7.5 Vorzeitiges Verlassen einer Prozedur

Normalerweise wird eine Prozedur erst beendet, wenn alle Codezeilen abgearbeitet sind, die vor der Endzeile `End Function` bzw. `End Sub` stehen. Sie können eine Prozedur jedoch vorzeitig verlassen. Dazu dienen die Anweisungen

Exit Function

für eine Funktion und

Exit Sub

für eine Sub-Prozedur. In der Regel werden Sie eine dieser Anweisungen in eine *If ... Then*-Figur einbauen, um das vorzeitige Beenden von einer Bedingung abhängig zu machen. Beim Testen von Prozeduren können die Anweisungen jedoch auch ohne Bedingung nützlich sein, um einen Prozedurteil

(vorübergehend) abzutrennen. Ein weiterer häufiger Anwendungsfall sind Fehlerbehandlungsroutinen, vgl. dazu besonders Kap. 30, *Programmieren mit VBA*, Punkt 30.6, *Fehlerbehandlung*.

29.7.6 Verweisen auf ein Visual Basic-Projekt in einer anderen Microsoft Access-Datenbank

Jede Access-Datenbank enthält ein VBA-Projekt. Es besteht aus allen Modulen des Projekts – den Standardmodulen, den Klassenmodulen und den Klassenobjekten (Formular/Berichtsmodule). Datenbankname und Projektname können sich unterscheiden. Der Name der Datenbank entspricht dem Namen der .MDB-Datei, während der Name des Projekts durch die Einstellung der Projekteigenschaft *Name* im Eigenschaftenfenster festgelegt wird. Wenn Sie eine Datenbank neu erstellen, sind Datenbankname und Projektname standardmäßig identisch. Wenn Sie die Datenbank später umbenennen, ändert sich der Projektname jedoch nicht automatisch. Umgekehrt hat auch das Ändern des Projektnamens keine Auswirkung auf den Datenbanknamen.

Sie können von einem VBA-Projekt in einer Access-Datenbank einen Verweis auf ein Projekt in einer anderen Access-Datenbank, auf eine Bibliotheksdatenbank oder auf ein in einer .MDE-Datei enthaltenes Add-In einrichten. Anschließend können Sie über die VBA-Prozeduren des Projekts, auf das Sie einen Verweis eingerichtet haben, verfügen.

Um beispielsweise einen Verweis auf das Projekt in der Beispieldatenbank *Nordwind.mdb* festzulegen, gehen Sie folgendermaßen vor:

- Aktivieren Sie den Visual Basic-Editor.
- Wählen Sie im Menü *Extras* den Befehl *Verweise*, und klicken Sie dann im Dialogfeld *Verweise* auf die Schaltfläche *Durchsuchen*.
- Suchen und markieren Sie die Datei *Nordwind.mdb*, und bestätigen Sie das Dialogfeld mit OK.

Danach sollte die Datenbank *Nordwind.mdb* in die Liste der verfügbaren Verweise im Dialogfeld *Verweise* sowie in den Projekt-Explorer aufgenommen worden sein mit der Wirkung, daß in der Datenbank, in der der Verweis eingerichtet wurde, über die VBA-Prozeduren von *Nordwind.mdb* so verfügt werden kann, als ob sie Bestandteil der eigenen Datenbank wären.

Ein Verweis kann nur auf ein Projekt in einer anderen Access 2000-Datenbank eingerichtet werden, nicht auf eine Access-Datenbank einer früheren Version.

29.8 Entscheidungs- und Schleifenstrukturen

VBA bietet die in modernen Programmiersprachen üblichen Möglichkeiten, um Entscheidungen mit Hilfe von Bedingungen zu treffen sowie bedingte und unbedingte Schleifen auszuführen. Im folgenden werden die wichtigsten davon behandelt.

29.8.1 Entscheidungsstrukturen

Um Wertzuweisungen oder den Programmfluß von Bedingungen abhängig zu machen, stehen Ihnen zwei Möglichkeiten zur Verfügung:

If ... Then

und

Select Case ... End Select.

Beide Konstruktionen können mit oder ohne *Else* oder *ElseIf* bzw. *Case Else* angegeben werden. Die Figur *If ... Then ... Else* ist letztlich leistungsfähiger als die Figur *Select Case,* weil Sie mit ihr differenziertere Entscheidungsstrukturen aufbauen können als mit letzterer. Dafür bietet *Select Case ... End Case* in Situationen, wo Sie fallweise und listenartige Auswertungen vorzunehmen haben, größere Bequemlichkeit und Übersicht.

If ... Then ... Else

Diese Anweisung führt andere Anweisungen in Abhängigkeit von Bedingungen aus. Es stehen zwei Syntaxformen zur Verfügung:

Syntax 1

If Bedingung **Then** Anweisungen-1 [**Else** Anweisungen-2]

Syntax 2

If Bedingung1 **Then**
  [Anweisungsblock-1]
[**ElseIf** Bedingung2 **Then**
  Anweisungsblock-2]]
[Else
  Anweisungsblock-n]]
End If

Bemerkungen zu Syntax 1

Mit Syntax 1 wird die einzeilige Form der *If*-Anweisungen beschrieben. Die *Anweisungen-1* werden ausgeführt, wenn *Bedingung* Wahr ist. Wenn *Bedingung* Falsch ist, werden *Anweisungen-2* ausgeführt; falls der *Else*-Teil nicht angegeben ist, passiert in diesem Falle gar nichts. Wenn *Anweisungen-1* oder *Anweisungen-2* aus mehr als einer Anweisung besteht, müssen die einzelnen Anweisungen durch je einen Doppelpunkt gegeneinander getrennt werden.

Als Anweisung darf auch eine Zeilennummer angegeben werden; dann wird zu der entsprechenden Programmzeile verzweigt. Das folgende Beispiel stellt eine gültige einzeilige *If*-Anweisung dar:

```
If A > 2 Then B = A: C = A ^ 2:  Else B = A / 2: C = A / 4
```

Sie sollten die einzeilige Form nur für sehr kurze *If*-Anweisungen verwenden, weil sie bei längeren Anweisungen nicht sehr gut lesbar ist. Außerdem erlaubt die Blockform der Syntax 2 flexiblere Kontrollstrukturen.

Bemerkungen zu Syntax 2

Gegenüber der einzeiligen Form von Syntax 1 liegen hier folgende Unterschiede vor:

- Jede Anweisung wird in eine eigene Zeile geschrieben. Das führt zu größerer Übersichtlichkeit.
- Mit den optionalen *ElseIf...Then*-Anweisungen, von denen Sie beliebig viele anführen dürfen, können Sie in einer *If*-Figur mehr als eine Bedingung auswerten lassen. Die *ElseIf...Then*-Anweisungen kommen nur zum Tragen, wenn die *If*-Bedingung *Falsch* ist. Es werden so lange weitere *ElseIf*-Bedingungen ausgewertet, wie keine der in der *If*-Figur vorangehenden Bedingungen *Wahr* ergab. Das folgende Beispiel stellt eine gültige If-Figur mit mehreren *ElseIf...Then*-Anweisungen dar.

```
If Einkommen <= 1000 Then
    Abgabe = Einkommen * 0.1
ElseIf Einkommen <= 5000 Then
    Abgabe = Einkommen * 0.2
ElseIf Einkommen <= 10000 Then
    Abgabe = Einkommen * 0.3
Else
    Abgabe = Einkommen * 0.4
End If
```

Bedingungen

Diese kann aus zwei verschiedenen Arten von Ausdrücken bestehen:

- **Bedingungsform 1**. Geben Sie einen numerischen Ausdruck oder einen Zeichenfolgenausdruck an, der den Wert *True* (*Wahr*, *-1*) oder *False* (*Falsch*, *0*) ergibt. Ein solcher Ausdruck kann auch Objektelemente wie z.B. ein Tabellenfeld oder ein Steuerelement enthalten, z.B. in der Form

```
Forms!Hauptaufträge!AuftragsNr >= 19920001
```

- **Bedingungsform 2**. Sie können prüfen, ob ein Objekt von einem bestimmten Typ ist. Dies geschieht in der Form

  If TypeOf Objekt *Is* Objekttyp *Then*

Objekt ist eine Steuerelementvariable. Für *Objekttyp* geben Sie ein reserviertes Wort an, welches den zu prüfenden Objekttyp bezeichnet, z.B. *CommandButton* für eine Befehlsschaltfläche oder *TextBox* für ein Textfeld.

Bei *Objekttyp* kann es sich z.B. um einen der folgenden Typen handeln (da es sich um VBA handelt, sind nur die englischen Namen anzugeben):

BoundObjectFrame (Gebundenes Objektfeld), *CheckBox* (Kontrollkästchen), *ComboBox* (Kombinationsfeld), *CommandButton* (Befehlsschaltfläche), *Label* (Bezeichnungsfeld), *Line* (Linie), *ListBox* (Listenfeld), *ObjectFrame* (Objektfeld für eingebettete und verknüpfte Objektfelder und Diagramme), *OptionButton* (Optionsfeld), *OptionGroup* (Optionsgruppe), *PageBreak* (Seitenumbruch), *Rectangle* (Rechteck), *SubForm* (Unterformular), *SubReport* (Unterbericht), *TextBox* (Textfeld) oder *ToggleButton* (Umschaltfläche).

Beispiel für Bedingungsform 2:

```
Dim SteuerElement As Control
Set SteuerElement = Forms!Hauptaufträge!AuftragsNr
If V1 >= 1000 And V1 <= 2000 Then
'...
ElseIf TypeOf SteuerElement Is TextBox Then
...
End If
```

Select Case ... End Select

Diese Anweisung führt, ähnlich wie die *If ... Then ... End If*-Anweisung, andere Anweisungen in Abhängigkeit von einer Bedingung aus. *Select Case*-Anweisungen können geschachtelt werden; zu jeder *Select Case*-Anweisung muß dann eine korrespondierende *End Select*-Anweisung angeführt werden. *Select Case* führt den Anweisungsblock aus, für dessen *Ausdrucksliste* der *Testausdruck* zutrifft.

Syntax

Select Case Testausdruck
  [**Case** Ausdrucksliste1
    [Anweisungsblock-1]]
  [**Case** Ausdrucksliste2
    [Anweisungsblock-2]]
  [**Case Else**
    [Anweisungsblock-n]]
End Select

Argument	Beschreibung
Testausdruck	Beliebiger numerischer oder Zeichenfolgenausdruck.
Anweisungsblock	Jeder Anweisungsblock besteht aus beliebig vielen Anweisungen.
Ausdrucksliste	Die Ausdrucksliste kann eine der drei folgenden Formen haben: Ausdruck [, Ausdruck] ... Ausdruck To Ausdruck Is Vergleichsoperator Ausdruck

Wenn Sie das Schlüsselwort *To* zum Definieren eines Wertebereichs verwenden, muß der kleinere Wert links stehen. Vergleichsoperatoren können nur in Verbindung mit dem Schlüsselwort *Is* verwendet werden. Wenn Sie eine *Case Else*-Anweisung verwenden und keine der Ausdrucklisten-Bedingungen mit *Testausdruck* übereinstimmt, wird der zu *Case Else* gehörige Anweisungsblock ausgeführt. Wenn Sie keine *Case Else*-Anweisung verwenden und keine der Ausdrucklisten-Bedingungen mit *Testausdruck* übereinstimmt, bleibt die *Select Case*-Anweisung wirkungslos.

In jeder *Case*-Klausel dürfen Sie auch mehr als einen Ausdruck bzw. Bereich anführen. Beispielsweise ist die folgende Klausel möglich:

```
Case 4, 5, 7, 10 To 20, Is > 100
```

Für Zeichenfolgenausdrücke sind ebenfalls Bereiche möglich. Im folgenden Beispiel wird der auf die *Case*-Klausel folgende Anweisungsblock ausgeführt, wenn *Testausdruck* genau *Bass* ist, ein Wort zwischen *Cello* und *Flöte* (in alphabetischer Reihenfolge) oder genau *Violine* enthält.

```
Case "Bass" "Cello" To "Flöte" "Violine"
```

Beispiel

Die folgende Funktion *Rabatt* gibt einen nach Umsatzgröße gestaffelten Rabatt aus.

```
Function Rabatt (X)
Select Case X
    Case Is < 1000
        Rabatt = 0
    Case Is < 10000
        Rabatt = .02 * X
    Case Is < 50000
        Rabatt = .04 * X
```

```
        Case Else
            Rabatt = .05 * X
    End Select
    End Function
```

29.8.2 Schleifen

Unbedingte Schleifen

Eine unbedingte Schleife führt eine Anweisung oder einen Block von Anweisungen mit einer vorher bestimmten Anzahl von Wiederholungen aus, die zusammen mit der Schleifenfigur anzugeben ist. Unbedingte Schleifen werden mit einer der Anweisungen ***For ... Next*** oder *For Each ... Next* realisiert.

For ... Next

Syntax

For Zähler = Startwert **To** Endwert [**Step** Schrittgröße]
    [Anweisungsblock]
[**Exit For**]
    [Anweisungsblock]
Next [Zähler [, Zähler]]

Argument	Beschreibung
Zähler	Numerische Variable. Falls diese nicht deklariert wurde, geschieht dies hier implizit.
Startwert	Anfangswert des Zählers; darf positiv und negativ sein.
Endwert	Endwert des Zählers; darf positiv und negativ sein.
Schrittgröße	Wert, um den Zähler bei jedem Schleifendurchlauf verändert wird; darf positiv und negativ sein. Wenn Schrittgröße ausgelassen wird, erhält sie den Wert 1.
Anweisungsblock	Beliebige Anzahl von Programmanweisungen oder Methoden. Darunter darf auch *Exit For* sein (im allgemeinen in Verbindung mit *If ... Then*); dann wird die Ausführung der weiteren Anweisungen innerhalb der Schleife an der Stelle abgebrochen, und die Schleife wird nicht weiter wiederholt, auch, wenn Zähler noch nicht den Endwert erreicht hat.

VBA führt eine *For ... Next*-Schleife in der folgenden Weise aus: Zunächst wird *Zähler* auf den Wert von *Startwert* gesetzt. Auf die Variable *Zähler* kann sowohl innerhalb wie außerhalb der Schleife zugegriffen werden. Wenn der Wert von *Zähler* größer als der von *Endwert* ist, wird gar nicht erst in die Schleife eingesprungen, sondern der Befehlsfluß wird gleich hinter

Next fortgesetzt. Sofern dies nicht zutrifft, werden die auf *For* folgenden Befehle zunächst bis *Next* ausgeführt. Dann wird *Schrittgröße* zu *Zähler* addiert und geprüft, ob *Zähler* größer als *Endwert* ist. Trifft dies zu, wird die Schleife verlassen, sonst wird sie erneut hinter *For* begonnen.

For ... Next-Schleifen können auch ineinander verschachtelt werden. Dann sollten Sie aus Gründen der Übersichtlichkeit hinter jedem *Next* den an dieser Stelle sonst wahlfreien Variablennamen von *Zähler* anführen. Das folgende Beispiel, das den 1728 (=12*12*12) Elementen einer Datenfeldvariablen (Array) Werte zuweist, erscheint übersichtlich:

```
Static DatFeld(12, 12, 12)
For i = 1 To 12
    For j = 1 To 12
        For k = 1 To 12
            DatFeld(i, j, k) = k
        Next k
    Next j
Next i
```

Die drei Next-Anweisungen hätten auch in der Form

```
        ...
        Next
    Next
Next
```

geschrieben werden dürfen: Ein *Next* ohne nachfolgenden *Zähler* korrespondiert mit dem nach oben nächsten *For* und dessen *Zähler*.

Obwohl es möglich ist, den Wert von *Zähler* durch Anweisungen innerhalb der Schleife zu verändern, sollten Sie, gutem Programmierstil folgend, davon absehen, weil dadurch die klare Figur der *For ... Next*-Schleife ihre Eindeutigkeit verliert. Dadurch wird die Lesbarkeit von Code und die Fehlersuche in Programmen wesentlich beeinträchtigt.

Wenn Sie beachten, daß Sie eine *For ... Next*-Schleife mit der Anweisung *Exit For* in Verbindung mit einer Bedingung vorzeitig beenden können, empfiehlt sich vielfach die Verwendung einer *For ... Next*-Schleife auch dort, wo eine bedingte Schleife angebracht zu sein scheint.

For Each ... Next

Diese Figur wiederholt eine Gruppe von Anweisungen für jedes Element in einem *Datenfeld* oder einer *Auflistung*.

Syntax

For Each Element **In** Gruppe
    [Anweisungsblock]
[**Exit For**]
    [Anweisungsblock]
Next [Element]

Argument	Beschreibung
Element	Variable zum Durchlauf durch die Elemente der Auflistung oder des Datenfeldes. Bei Auflistungen ist für *Element* nur eine Variable vom Typ *Variant*, eine allgemeine Objektvariable oder eine beliebige spezielle Objektvariable zulässig. Bei Datenfeldern ist für Element nur eine Variable vom Typ *Variant* zulässig.
Gruppe	Name einer Objektauflistung oder eines Datenfeldes (außer Datenfeldern mit benutzerdefinierten Typen).
Anweisungen	Optional. Eine oder mehrere Anweisungen, die für jedes Element in Gruppe ausgeführt werden.

Das Innere des *For…Each*-Blocks wird ausgeführt, wenn sich mindestens ein Element in *Gruppe* befindet. Nachdem mit der Ausführung der Schleife begonnen wurde, werden alle Anweisungen in der Schleife für das erste Element in *Gruppe* ausgeführt. Enthält Gruppe mehrere Elemente, so werden die Anweisungen in der Schleife für jedes Element wiederholt.

Sie können *For…Each...Next*-Schleifen verschachteln, indem Sie eine *For…Each...Next*-Schleife innerhalb einer anderen verwenden.

Das folgende Beispiel gibt die Namen aller Tabellen der aktuellen Datenbank in das Direktfenster aus:

```
Sub FeldnamenAusgeben()
Dim DB
Set DB = CurrentDb()
For Each Tabelle In DB.TableDefs
  Debug.Print Tabelle.Name
Next Tabelle
End Sub
```

Bedingte Schleifen

Eine bedingte Schleife wiederholt eine Anweisung oder einen Block von Anweisungen so oft, bis eine Bedingung erfüllt oder nicht mehr erfüllt ist. Für bedingte Schleifen stehen die Figuren

Do ... Loop

und

While ... Wend

zur Verfügung. Eine *Do ... Loop*-Schleife wird zusammen mit einem der beiden reservierten Wörter *While* oder *Until* angegeben.

Do ... Loop

Wiederholt einen Block von Anweisungen, solange eine Bedingung *Wahr* ist (*While*) oder bis eine Bedingung *Wahr* wird (*Until*).

Syntax 1

```
Do [{While | Until}] Bedingung]
    [Anweisungsblock]
    [Exit Do]
    [Anweisungsblock]
Loop
```

Syntax 2

```
Do
    [Anweisungsblock]
    [Exit Do]
    [Anweisungsblock]
Loop [{While | Until}] Bedingung]
```

Bedingung ist ein Ausdruck, der *True* (*Wahr*, *-1*) oder *False* (*Falsch*, *0*) werden kann. Die Programmzeilen zwischen *Do* und *Loop* werden so oft wiederholt, solange (*While*) oder bis (*Until*) *Bedingung True* (*Wahr*, *-1*) ist.

Exit Do ist nur innerhalb einer *Do ... Loop*-Schleife zulässig. Wenn der Programmfluß auf *Exit Do* trifft, wird die *Do ... Loop*-Schleife an dieser Stelle beendet, und der Programmfluß wird unmittelbar hinter dem zu dieser *Do ... Loop*-Schleife gehörenden *Loop* fortgesetzt. *Exit Do* ergibt im allgemeinen nur einen Sinn, wenn es als mögliche Folge einer bedingten Anweisung (z.B. *If ... Then*) angeführt wird.

Sie dürfen mehrere *Do ... Loop*-Schleifen ineinander verschachteln.

Die zur *Do ... Loop*-Schleife angegebene Bedingung vergleicht häufig Zeichenfolgen miteinander. Wenn Sie in dieser Hinsicht überraschende Ergebnisse registrieren, prüfen Sie die *Option Compare*-Anweisung im Deklarationsteil des Moduls, die Einfluß auf die Art des Zeichenfolgenvergleichs hat.

Beispiel. Das folgende Beispiel gibt die Zahlenfolge 2, 4, 8, 16, 32, 64 in das Direktfenster aus.

```
Zahl = 2
Do Until Zahl > 32
    Zahl = Zahl * 2
    Debug.Print Zahl
Loop
```

Beispiel. Das folgende Beispiel gibt die Zahlenfolge 2, 4, 8, 16, 32 in das Direktfenster aus.

```
Zahl = 2
Do While Zahl < 32
    Zahl = Zahl * 2
    Debug.Print Zahl
Loop
```

Beispiel. Das folgende Beispiel gibt die Zahlenfolge 2, 4, 8, 16 in das Direktfenster aus.

```
Zahl = 2
Do While Zahl < 32
    Zahl = Zahl * 2
    Debug.Print Zahl
    If Zahl >= 16 Then Exit Do
Loop
```

While ... Wend

Führt den *Anweisungsblock* in einer Schleife aus, solange *Bedingung Wahr* ist.

Syntax

While Bedingung
  [Anweisungsblock]
Wend

Um unendliche Schleifen zu vermeiden, müssen Sie dafür sorgen, daß die Bedingung während der Schleifenwiederholungen Falsch wird. Im übrigen empfiehlt es sich im allgemeinen, statt mit *While ... Wend* mit *Do ... Loop* zu arbeiten, weil diese Figur strukturiertere und weitergehende Möglichkeiten bietet.

Beispiel. Das folgende Beispiel gibt die Zahlenfolge 2, 4, 8 ... 1024 in das Direktfenster aus.

```
A = 2
While A <= 1024
Debug.Print A
A = A * 2
Wend
```

29.9 Deutsche und englische Sprache

Die Access-Sprachelemente kommen sowohl in deutscher wie in englischer Sprache vor. Dieser Umstand stiftet zunächst Verwirrung, zumal zwischen beiden Sprachebenen keine vollständige Parallelität besteht: Zwar existiert für jeden deutschen Ausdruck ein entsprechender englischer, jedoch gilt das Umgekehrte nicht, denn es gibt eine Reihe von Elementen, die nur in englischer Sprache zur Verfügung stehen. Darüber hinaus können Sie an einigen Stellen beide Sprachen benutzen, an anderen dagegen nur die englische Sprachversion eines Sprachelements. Klarheit stellt sich ein, wenn Sie die folgenden Hinweise beachten:

- Deutsche Sprachelemente können Sie nur auf der allgemeinen Access-Ebene verwenden, nicht jedoch in VBA.
- In VBA können Sie nur englische Elemente angeben.

In Ausdrücken auf der allgemeinen Access-Ebene können Sie in der Regel auch die englischen Sprachelemente angeben; diese werden dann aber von Access in das entsprechende deutsche Wort umbenannt, nachdem sie von Access akzeptiert wurden. Beispielsweise können Sie für die Eigenschaft *Steuerelementinhalt* eines Textfeldes den Ausdruck =Now() eingeben. Wenn Sie den Fokus wechseln, akzeptiert Access diesen Ausdruck und wandelt ihn in die Schreibweise =Jetzt() um

Einige Sprachelemente existieren nur in englischer Version. Dies gilt für alle *Methoden* und *Anweisungen*. Solche Elemente sind dann in der Regel auch nur in VBA verwendbar.

Kapitel 30

Programmieren mit VBA

30.1 Das Access 2000 Objektmodell

30.1.1 Objekte, Methoden, Eigenschaften

Objekte. Mit VBA können Sie fast alle Vorgänge steuern, die überhaupt in Access möglich sind. Dabei müssen Sie zahlreiche unterschiedliche Elemente von Access wie z. B. Formulare, Berichte oder Steuerelemente ansprechen können, um damit die unterschiedlichsten Aktionen vorzunehmen. Die Vielzahl der unterschiedlichen Access-Elemente ist in Form von Objekten organisiert, die Sie ansprechen können. Ein Beispiel für ein Objekt ist *Form* (Formular).

Eigenschaften. Objekte haben Eigenschaften und Methoden, über die sie gesteuert werden. Beispielsweise hat ein Objekt *Form* die Eigenschaft *Visible*, die verschiedene Einstellungen besitzen kann. Wenn Sie beispielsweise die Eigenschaft *Visible* eines bestimmten Formulars auf den Wert *False* einstellen, wird das Formular ausgeblendet, mit der Eigenschaftseinstellung *True* wird es angezeigt. Die meisten Eigenschaften von Objekten lassen sich von VBA aus verändern. Bestimmte Eigenschaften lassen sich dagegen nur lesen und nicht verändern.

Methoden. Objekte besitzen außer Eigenschaften auch Methoden. Die Methode eines Objekts führt im allgemeinen eine Aktion aus. Beispielsweise besitzt das Objekt *Form* die Methode *Requery*. Diese Methode aktualisiert die zugrundeliegenden Daten des angegebenen Formulars durch erneutes Abfragen der Datenherkunft des Formulars.

Wenn Sie die Eigenschaft oder die Methode zu einem Objekt angeben, werden beide durch einen Punkt voneinander getrennt. Beispielsweise wird die Eigenschaft *ActiveForm* des Objekts *Screen* (Bildschirm) durch die Schreibweise

```
Screen.ActiveForm
```

angegeben.

30.1.2 Vordefinierte und benutzerdefinierte Objekte

Access stellt zahlreiche vordefinierte Objekte zur Verfügung. Vordefinierte Objekte sind beispielsweise das Objekt *Forms*, das alle Formulare (ggf. auch kein einziges, wenn keines definiert ist) enthält, oder das Objekt *Screen*, das den Bildschirm bezeichnet. Ein benutzerdefiniertes Objekt, z.B. ein bestimmtes Formular und ein bestimmtes Steuerelement, werden im allgemeinen dadurch identifiziert, daß man Bezug auf ein vordefiniertes Objekt nimmt und eine bestimmte Konkretisierung davon angibt. Dazu dienen vor allem die Container-Objekte und Auflistung-Objekte.

30.1.3 Container-Objekte

Ein Objekt kann in einem übergeordneten Objekt enthalten sein. Beispielsweise sind ein bestimmtes Formular und ein darin enthaltenes bestimmtes Steuerelement jeweils ein Objekt. Das Objekt *Control* (Steuerelement) ist im Objekt *Form* (Formular) enthalten. Daher wird das Objekt *Form* als Container-Objekt bezeichnet. Dadurch existiert im allgemeinen eine Hierarchie von Objekten. Wenn Sie ein bestimmtes konkretes Objekt im VBA-Code identifizieren wollen, müssen Sie prinzipiell zusätzlich zum Namen des konkreten Objekts alle übergeordneten Objekte mit angeben. Das Objekt auf der höchsten Hierarchie-Ebene hat den Namen *Application*. Um beispielsweise ein Steuerelement mit dem Namen *Preisliste* im Formular *Angebote* auszublenden, müßten Sie prinzipiell die folgende Anweisung schreiben:

```
Application.Forms!Angebote!Preisliste.Visible=False
```

Das Erfordernis, zur Identifizierung untergeordneter Objekte deren übergeordnete Container-Objekte mit anzugeben, führt oft zu sehr langem Code. VBA sieht daher für bestimmte Fälle Möglichkeiten vor, die übergeordneten Objekte fortzulassen oder durch äquivalente kürzere Angaben zu ersetzen. Von wenigen Ausnahmen abgesehen dürfen Sie die Angabe des Objekts *Application* fortlassen. Dann würde die gerade angeführte Codezeile sich verkürzen auf

```
Forms!Angebote!Preisliste.Visible=False
```

Wenn die Anweisung in einem Kontext ausgeführt würde, in dem das Formular *Angebote* das aktive Fenster ist, würde es sogar ausreichen zu schreiben

```
Preisliste.Visible=False
```

Eine vollständige Darstellung aller Access 2000-Objekte und deren Hierarchie finden Sie in der Access 2000-Hilfe (nicht in der VBA-Hilfe). Schlagen Sie im Inhaltsverzeichnis das Thema *Informationen zur Programmierung*, Unterthema *Microsoft Access Visual Basic-Sprachverzeichnis*, Unterthema *Microsoft Access-Objekte* auf.

Bild 30.1: Darstellung der Objekthierarchie in der VBA-Online-Hilfe

30.1.4 Objekte, die eine Auflistung von Objekten enthalten

Das Objektmodell faßt gleichartige Objekte jeweils in einem übergeordneten Objekt, das als *Auflistung-Objekt* bezeichnet wird, zusammen. Beispielsweise werden alle Formulare derselben Arbeitsmappe in dem Objekt mit dem vordefinierten Namen *Forms* gesammelt. Entsprechend werden alle geöffneten Berichte im Objekt *Reports* gesammelt usw. Derartige Auflistung-Objekte sind notwendig, damit einerseits die im letzten Punkt besprochene Objekt-Hierarchie durchgehalten werden kann, andererseits in diese Objekt-Hierarchie vom Benutzer neu geschaffene Objekte aufgenommen und eingeordnet werden können.

Um ein konkretes Objekt in einem Auflistung-Objekt zu identifizieren, müssen Sie dieses prinzipiell hinter dem Namen des Auflistung-Objekts angeben. Dabei können Sie den Namen des konkreten Objekts als Text angeben oder dieses über seine fortlaufende Nummer im Auflistung-Objekt bezeichnen. Dabei ist zu beachten, daß die fortlaufende Numerierung der Objekte in einem Auflistung-Objekt mit 0 beginnt (und nicht, wie beispielsweise in MS Excel 2000, mit 1). Wenn beispielsweise das Formular mit dem Namen *Verkäufe* das zweite in der Objektliste ist, sind die beiden folgenden Code-Zeilen gleichwertig:

```
Forms!Verkäufe.Visible = True
Forms(1).Visible = True
```

Wenn Sie alle Objekte eines Auflistung-Objekts ansprechen wollen, verwenden Sie am besten eine *For Each ... Next*-Schleife. Die folgenden Codezeile beispielsweise geben die Namen und Höheneinstellungen sämtlicher Steuerelemente des Formulars *Hauptmenü* in das Direktfenster aus:

```
For Each Element In Forms!Hauptmenü.Controls
   Debug.Print Element.Name, Element.Height
Next
```

30.1.5 Objekte mit kürzerem Code identifizieren

Die vorangehenden Überlegungen dieses Abschnitts haben gezeigt, daß Code zum Identifizieren von Objekten zuweilen sehr lang werden kann. Dies ist nicht nur unbequem, sondern führt auch oft zu Schreibfehlern, die Programmfehler zur Folge haben. Sie haben jedoch mehrere Möglichkeiten, mit kürzerem Code auszukommen.

- **Optionale Objektangaben fortlassen**. Das Objekt *Application* darf bei der Identifizierung von Objekten im allgemeinen fortgelassen werden.
- **Mehrere Aktionen auf dasselbe Objekt ausführen**. Wenn Sie mehrere Aktionen auf dasselbe Objekt ausführen wollen, können Sie die **With**-Anweisung verwenden.
- **Die Auflistung *Controls fortlassen***. Zur Identifizierung von Steuerelementen in Formularen und Berichten brauchen Sie das Auflistungsobjekt *Controls* nicht anzuführen, so daß Sie beispielsweise

  ```
  Forms!Artikel!Preis
  ```

  schreiben dürfen, um das Steuerelement *Preis* im Formular *Artikel* zu identifizieren.
- **Schlüsselwort *Me in Formular- und Berichtsmodulen***. In Formular- und Berichtsmodulen können Sie das Schlüsselwort *Me* verwenden, um auf das Formular oder den Bericht zu verweisen, zu dem dieses Klassenmodul gehört. Die letzte Codezeile dürfte dann beispielsweise in der Form

  ```
  Me!Preis
  ```

  geschrieben werden.
- **Objekt-Variablen verwenden**. Sie können jedes Objekt als Variable definieren und dann entsprechend identifizieren, vgl. dazu im einzelnen Punkt 30.3, Objektvariablen, in diesem Kapitel. Selbst deutlich »sprechende« Variablennamen sind im allgemeinen viel kürzer als eine lange Liste von Objekten, Eigenschaften oder Methoden. Objekt-Variablen sind ein geradezu ideales Mittel, um zu kürzerem VBA-Code zu kommen. Darüber hinaus bieten sie noch weitere Vorteile. So erlauben sie beispielsweise viel allgemeinere Formulierungen von Prozeduren, als dies sonst möglich wäre.

30.1.6 Sich über Objekte, Methoden und Eigenschaften informieren

In den vorangehenden Ausführungen war implizit lediglich von den Objekten der Anwendung Access 2000 die Rede. Darüber hinaus stehen Ihnen in VBA zahlreiche weitere Objekte mit ihren Methoden und Eigenschaften zur Verfügung. Hier sind vor allem die Objekte DAO mit ihren weiteren Objekten, Methoden und Eigenschaften hervorzuheben, mit denen Sie Datenzu-

griffe programmieren können. Diese Objekte, die auch anderen Office 2000-Anwendungen wie Excel 2000 oder Word 2000 zur Verfügung stehen, werden eigens in Kap. 31, *Datenzugriff mit DAO (Data Access Objects) und ADO (ActiveX Data Objects)*, behandelt. Weiterhin können Sie sich durch Einrichten von Verweisen auf andere Bibliotheken und Access-Datenbanken (vgl. den Befehl *Verweise* im Menü *Extras*) andere Objekte mit ihren Methoden und Eigenschaften verfügbar machen.

Zur Information über die in einer Datenbank aktuell verfügbaren Objekte dient der Objektkatalog, den Sie mit der Symbol-Schaltfläche *Objektkatalog* (vgl. links nebenstehend) oder dem gleichlautenden Befehl aus dem Menü *Ansicht* aufschlagen können.

Bild 30.2: Objektkatalog

Im Objektkatalog können Sie nicht nur die verfügbaren Objekte identifizieren, sondern sich auch über die für ein bestimmtes Objekt vorhandenen Methoden, Eigenschaften und Ereignisse informieren. In Bild 30.2 beispielsweise werden in der Liste *Klassen*, die die Objekte wiedergibt, gerade die Objekte der Anwendung Access angezeigt. In der Liste *Klassen* ist das Objekt *Form* markiert, so daß in der rechten Liste des Fensters die *Elemente von Form* wiedergegeben werden. Jedes Element ist durch ein Symbol

gekennzeichnet, an dem Sie erkennen können, ob es sich um eine Eigenschaft, Methode oder ein Ereignis handelt. In der in Bild 30.2 wiedergegebenen Situation ist gerade die Methode *Requery* markiert. Sie könnten sich sofort über diese Methode in der Hilfe informieren, indem Sie auf die Symbol-Schaltfläche mit dem Fragezeichen klicken. Dann wird der entsprechende Hilfetext aufgeschlagen. Allerdings ist nicht für jedes Objekt oder Element Hilfe verfügbar, denn es werden ja ggf. auch Objekte angezeigt, auf die benutzerdefiniert ein Verweis eingerichtet wurde, und zumindest für diese Objekte ist nicht gesichert, daß Hilfe angeboten wird.

Wenn Sie sich erst einmal in der VBA-Online-Hilfe befinden, können Sie ebenfalls von der Systematik des Objektmodells profitieren: Zum Hilfetext eines bestimmten Objekts wird stets je ein Verweisbegriff für die zugehörigen Eigenschaften und Methoden angeboten, so daß Sie schnell einen Überblick über die für ein bestimmtes Objekt verfügbaren Möglichkeiten gewinnen können.

30.2 Variablen und Konstanten

30.2.1 Variablen deklarieren

Variablen können implizit oder explizit deklariert werden. Bei expliziter Deklaration stehen Ihnen differenzierte Möglichkeiten bezüglich Datentyp, Gültigkeitsbereich und Lebensdauer der Variablen zur Verfügung. Ferner führt explizite Deklaration zu größerer Klarheit und Sicherheit Ihres Programmcodes. Bei impliziter Deklaration trifft dies nicht zu, dafür ist diese Arbeitsweise bequemer.

Variablennamen

Die Regeln für die Vergabe von Variablennamen sind dieselben wie für Funktionsnamen, vgl. Kap. 29, *VBA (Visual Basic for Applications): Grundlagen*, Punkt 29.7.2, *Funktionen: Grundlagen*.

Implizite Deklaration

Sie können eine Variable einfach im Code anführen, ohne sie vorher explizit deklariert zu haben. Sie wird dann von VBA implizit deklariert. Die Programmzeile

```
Mehrwertsteuer = Nettobetrag*0.16
```

beispielsweise deklariert die Variable *Mehrwertsteuer* implizit, wenn diese nicht an anderer Stelle bereits (implizit oder explizit) deklariert wurde. Implizit deklarierte Variablen bekommen den Datentyp *Variant* zugewiesen (abgesehen von Objektvariablen, für die dies anders geregelt ist), der eine sehr flexible Behandlung verschiedener Datentypen ermöglicht. Der Gültig-

keitsbereich implizit deklarierter Variablen ist stets auf die Prozedur beschränkt, in der sie implizit deklariert werden. Die Lebensdauer implizit deklarierter Variablen ist stets auf die Dauer ihrer Prozedur begrenzt.

Wenn Sie Variablen implizit deklarieren wollen, darf im Modul nicht die Anweisung

```
Option Explicit
```

stehen. Achten Sie für neue Module ggf. darauf, daß das Kontrollkästchen *Variablen-Deklaration erforderlich* auf der Registerkarte *Editor* des Dialogfeldes *Optionen* (gleichlautender Befehl im Menü *Extras*) deaktiviert ist, und/oder entfernen Sie die angegebene Codezeile im Deklarationsteil des Moduls.

Explizite Deklaration

Sie deklarieren Variablen explizit mit einer der Anweisungen

Dim bzw. **ReDim** bzw. **Public** bzw. **Private** bzw. **Static**.

Die Anweisungen leisten im Prinzip das Gleiche, sie unterscheiden sich jedoch hinsichtlich der Wirkung auf den Gültigkeitsbereich bzw. die Lebensdauer von Variablen, vgl. dazu in diesem Kapitel die Punkte 30.2.2, *Gültigkeitsbereich* und 30.2.3, *Lebensdauer.*

Im folgenden wird die Syntax der *Dim*-Anweisung – beschränkt auf die wichtigsten Elemente – wiedergegeben. Für die Anweisungen *Public*, *Private* und *Static* gilt eine ganz entsprechende Syntax, vgl. im übrigen zu detaillierten Informationen die Online-Hilfe.

Syntax

Dim Variable [([Indizes])][**As** Datentyp][, Variable [([Indizes])] [**As** Datentyp]] ...

Dim deklariert Variablen auf Modul- oder Prozedurebene. Auf Modulebene deklarierte Variable sind für alle Prozeduren dieses Moduls verfügbar, auf Prozedurebene deklarierte nur für diese Prozedur.

Argument	Beschreibung
Variable	Name, den die Variable bekommt.
Indizes	Dimension(en) einer Datenfeldvariablen (Array). Es dürfen mehrere Dimensionen deklariert werden, vgl. unten.
As Datentyp	Reserviertes Wort mit nachfolgender Angabe des Datentyps. Datentyp kann u.a. sein: Integer, Long, Single, Double, Currency, String (für variable Länge), String * Länge (für feste Länge), Variant, benutzerdefinierte Datentypen, Objektdatentypen (Dynaset, Database etc.). Für jede Variable wird eine eigene As-Klausel angegeben.

Das Argument *Indizes* bezieht sich auf Datenfeldvariable, die genauer im Punkt 30.2.4, *Datenfelder (Arrays)*, behandelt werden. Es hat die folgende Syntax:

[Untergrenze **To**]Obergrenze[,[Untergrenze **To**]Obergrenze]

Das folgende Beispiel eines Ausschnitts aus einer Funktion zeigt, wie die Variable *Mehrwertsteuer* explizit mit einer *Dim*-Anweisung deklariert wird:

```
Function Endsumme (UmsatzMit, UmsatzOhne)
Dim Mehrwertsteuer
Mehrwertsteuer = 0.16 * UmsatzMit
...
...
Endsumme = UmsatzMit + UmsatzOhne + Merwertsteuer
End Function
```

Bei expliziter Variablendeklaration haben Sie auch den Vorteil, daß für jede Variable ein bestimmter Datentyp festgelegt werden kann. Wenn Sie keinen besonderen Datentyp angeben, legt VBA den Datentyp *Variant* fest, vgl. dazu detaillierte Hinweise in der Online-Hilfe. Wenn Sie die Variable *Mehrwertsteuer* im vorangehenden Beispiel als Datentyp *Currency* (Währung) deklarieren wollen, wäre die Deklarationszeile wie folgt zu schreiben:

```
Dim Mehrwertsteuer As Currency
```

Sie können auch mehrere Variablen, auch unterschiedlicher Datentypen, in einer *Dim*-Anweisung deklarieren. Verfahren Sie dabei nach dem folgenden Beispiel:

```
Dim MwSt, Total As Currency, Position As Integer, Var1, Var2
```

Die voranstehende Codezeile deklariert die Variable *Total* mit dem Datentyp *Currency* und *Position* als *Integer*. *MwSt*, *Var1* und *Var2* werden als Variable vom Typ *Variant* deklariert, weil das Fortlassen eines Datentyps implizit das Zuweisen dieses Datentyps bewirkt.

Verhindern impliziter Deklaration mit *Option Explicit*

Das vorangehende Beispiel mit der Funktion-Prozedur *Endsumme* enthält einen Fehler, der bei flüchtigem Hinsehen nicht auffällt: In der vorletzten Zeile steht das Wort *Merwertsteuer*, das einen Schreibfehler enthält, weil der Buchstabe *h* fehlt. Die Funktion würde gleichwohl insoweit ohne (syntaktischen) Fehler ausgeführt werden, denn es würde eine neue Variable mit dem Namen *Merwertsteuer* implizit deklariert werden. Da diese aber den fehlenden Wert Null hätte, würde sich eine fehlerhafte Endsumme ergeben. Solche sachlichen Fehler werden im allgemeinen schwer bemerkt, und auch dann,

wenn sie bemerkt werden, ist es oft besonders schwierig, den Grund für den sachlichen Fehler im Code zu entdecken. Das Beispiel zeigt, daß implizite Variablendeklaration, vor allem wegen möglicher Schreibfehler oder ähnlicher Irrtümer, Unsicherheiten schafft. Sie können VBA anweisen, implizite Variablendeklaration nicht zuzulassen. Um dies zu erreichen, schreiben Sie die Anweisung

```
Option Explicit
```

in den Deklarationsteil des betreffenden Moduls. Diese Anweisung gilt Modul-spezifisch. Sie müssen die Anweisung ggf. im Deklarationsteil jedes Moduls, für das eine implizite Variablendeklaration verhindert werden soll, anführen.

Wenn Sie das Kontrollkästchen *Variablen-Deklaration erforderlich* auf der Registerkarte *Editor* des Dialogfeldes *Optionen* (gleichlautender Befehl im Menü *Extras*) aktiviert haben, schreibt Access automatisch in den Deklarationsteil jedes neuen Moduls die Anweisung *Option Explicit*.

30.2.2 Gültigkeitsbereich

Variablen können drei verschiedene Gültigkeitsbereiche haben:

- **Prozedurebene**. Alle implizit und alle explizit in einer Prozedur deklarierten Variablen gelten nur für diese Prozedur. Von anderen Prozeduren kann nicht auf solche lokalen Variablen zugegriffen werden. Der Vorteil lokaler Variablen besteht u.a. darin, daß Sie denselben Variablennamen zwischen verschiedenen Prozeduren mehrfach verwenden können, ohne daß es zu formalen Namenskonflikten kommt.
- **Modulebene**. Um zu erreichen, daß eine Variable für alle Prozeduren desselben Moduls (nicht jedoch für andere Module) zur Verfügung steht, müssen Sie diese im Deklarationsteil des Moduls mit der Anweisung *Dim* deklarieren.
- **Öffentliche Ebene**. Variablen stehen für alle Prozeduren und alle Module zur Verfügung, wenn sie im Deklarationsteil eines Moduls mit der Anweisung *Public* deklariert werden.

Wenn namensgleiche Variablen mit unterschiedlichem Gültigkeitsbereich (nur dann ist dies möglich!) existieren, wird der mögliche Zugriffskonflikt folgendermaßen geregelt: VBA verwendet dann immer die Variable mit dem Gültigkeitsbereich der geringsten Reichweite. Lokale Variablen rangieren daher stets vor solchen auf Modulebene und öffentlicher Ebene, Modulvariablen haben Priorität vor öffentlichen.

30.2.3 Lebensdauer

Vom Gültigkeitsbereich von Variablen ist deren Lebensdauer zu unterscheiden. Der Gültigkeitsbereich hat allerdings Einfluß auf die Lebensdauer. Lokale Variablen gehen unter, wenn die Prozedur, in der sie vorkommen, beendet ist. Variablen mit einem modularen oder öffentlichen Gültigkeitsbereich bleiben dagegen erhalten, bis die Datenbank geschlossen wird oder ein Laufzeitfehler auftritt.

Sie können auch lokalen Variablen eine Lebensdauer geben, die für die Dauer der geöffneten Datenbank gilt. Zu diesem Zweck müssen Sie diese explizit mit der Anweisung

Static

deklarieren. Diese Anweisung wird in einer Prozedur genau so gebraucht wie eine *Dim*-Anweisung, die sie insoweit ersetzt. Die folgende Programmzeile deklariert die Variable *LetzteAuftragsNr* als statische, so daß sie nach dem Ende der Prozedur *bei erneutem Aufruf dieser Prozedur* (nicht jedoch für andere Prozeduren, weil die Variable nach wie vor lokal bleibt) weiterhin zur Verfügung steht:

```
Static LetzteAuftragsNr
```

Wenn Sie erreichen wollen, daß alle Variablen einer Prozedur statisch sind, können Sie einen abkürzenden Weg wählen: Stellen Sie dem Wort *Function* bzw. *Sub* im Prozedurkopf das reservierte Wort *Static* voran, wie in der folgenden Beispielzeile, mit der alle Variablen der Funktion *Immerwährend* zu statischen gemacht werden:

```
Static Function Immerwährend ()
```

Wenn der VBA-Code neu initialisiert wird (Befehl *Zurücksetzen* aus dem Menü *Ausführen bzw. das entsprechende Symbol in der Standard-Symbolleiste*), was zumindest in der Testphase häufig infolge von Laufzeitfehlern notwendig ist, verlieren alle Variablen ihren Wert, gleichgültig, welche Lebensdauer ihnen zugewiesen ist.

30.2.4 Datenfelder (Arrays)

Variablen können als Datenfelder, häufig auch als Arrays bezeichnet, deklariert werden. Dies geschieht durch entsprechende Definition mit einer der Anweisungen *Dim*, *Public* oder *Static*.

Die Syntax zur Deklaration einer Datenfeldvariablen lautet prinzipiell:

Dim Variable [Untergrenze **To**]Obergrenze[,[Untergrenze **To**]Obergrenze]

Sofern Sie keine Anweisung *Option Base* (vgl. hierzu genauer in der Online-Hilfe), die die implizit angenommene Untergrenze von Datenfeld-Dimensio-

nen betrifft, verwendet haben, sind die folgenden vier Beispiele wirkungsgleich: Jedes Beispiel definiert *Z* als Datenfeldvariable mit jeweils zwei Dimensionen, wobei die erste Dimension jeweils die 21 Elemente mit den Indexwerten 0, 1, ... 20 und die zweite jeweils die sechs Elemente mit den Indexwerten 0, 1, ... 5 besitzt.

```
Dim Z(20,5)
Dim Z(0 To 20, 0 To 5)
Dim Z(20, 0 To 5)
Dim Z(0 To 20, 5)
```

Sie dürfen auch negative ganzzahlige Werte als Indexangaben verwenden, z. B.

```
Dim Z(-44 To 12)
```

Bis zu 60 Dimensionen können Sie in einer Datenfeldvariablen (Array) verwenden. Das folgende Beispiel definiert sieben Dimensionen:

```
Dim Z(20, 20, 20, 20, 30, 10, -5 To 100)
```

Wenn Sie eine Variable mit leeren Klammern deklarieren, ist ihr Datenfeld dynamisch, d. h., die Anzahl ihrer Datenfeldelemente ist variabel.

```
Dim Z()
```

Sie können aus einer dynamischen Datenfeldvariablen in einer Prozedur mit *ReDim* nachträglich eine mit festem Index machen.

Beispiel. Das folgende Beispiel deklariert die lokale Variable *Zahlenfeld* als zweidimensionales Datenfeld, weist ihren insgesamt 30 Elementen Werte zu und gibt diese Werte anschließend mittels der Anweisung *Debug.Print* in das Direktfenster aus.

```
Function Array10x3 ()
Dim Zahlenfeld(10, 3)
For i = 1 To 10
    For j = 1 To 3
        Zahlenfeld(i, j) = i * j
    Next
Next
For i = 1 To 10
    For j = 1 To 3
        Debug.Print Zahlenfeld(i, j)
    Next
Next
End Function
```

Dynamische Datenfelder

Manchmal ist nicht unbedingt im voraus bekannt, wie groß die Dimensionen einer Datenfeldvariablen sein müssen. Um Speicherplatz zu sparen, könnte es sinnvoll sein, die Größe nachträglich zu ändern. Solche Variablen mit dynamischen Datenfeldern können Sie mit der Anweisung *ReDim* definieren, vgl. genauer die Online-Hilfe.

30.2.5 Symbolische Konstanten

Für den Fall, daß Sie einen festen Wert an verschiedenen Stellen einer Prozedur, eines Moduls oder öffentlich immer wieder benötigen, kann es sinnvoll sein, ihn mit der Anweisung *Const* als sogen. symbolische Konstante zu deklarieren. Die folgende Zeile deklariert die Konstante *PI*:

```
Const PI = 3.14159265
```

Anders als bei der Deklaration von Variablen wird einer symbolischen Konstanten beim Deklarieren ein Wert zugewiesen. Dies ist der wesentliche Vorteil, denn Sie können auf diese Weise z.B. alle symbolischen Konstanten im Deklarationsteil eines Moduls definieren. So haben Sie eine bestimmte Stelle, an der Sie alle Festlegungen von Konstanten einsehen können, statt in vielen Prozeduren suchen zu müssen.

Symbolische Konstanten werden in Prozeduren wie Variablen verwendet, allerdings mit dem wesentlichen Unterschied, daß Sie ihren Wert nur lesen, ihn jedoch nicht verändern können.

Wie Variablen haben symbolische Konstanten einen lokalen, modularen oder öffentlichen Gültigkeitsbereich. Bezüglich der Deklaration besteht in dieser Hinsicht nur der folgende Unterschied: Um eine symbolische Konstante öffentlich verfügbar zu haben, müssen Sie beim Deklarieren im Deklarationsteil eines Moduls dem Wort *Const* das Wort *Public* voranstellen. Die folgende Zeile beispielsweise definiert die beiden öffentlichen symbolischen Konstanten *PI* und *MwStSatz*:

```
Public Const PI = 3.14159265, MwStSatz = 0.16
```

30.3 Objektvariablen

30.3.1 Grundlagen

VBA ermöglicht es, Datenbankobjekte als Variablen zu behandeln. Diese Arbeitsweise erleichtert und erweitert die Manipulationsmöglichkeiten an Tabellen, Formularen, Steuerelementen etc. ganz erheblich. Obwohl Sie auf die meisten Datenbankobjekte wie z.B. Formulare, Steuerelemente etc. Bezug nehmen und diese auch manipulieren können, ohne Objektvariablen zu verwenden, können Sie über bestimmte, sehr leistungsfähige Methoden,

vor allem die Methoden des DAO-Objekts für die Datenmanipulation, nur verfügen, wenn Sie Objekte als Objektvariable behandeln. Dies gilt beispielsweise für die *Find*-Methoden, die *Move*-Methoden, das Editieren von Datensätzen mit der Methode *Edit* usw., auf die Sie kaum verzichten können, wenn Sie anspruchsvollere Datenbankanwendungen erstellen wollen.

Objektvariable werden wie andere Variable deklariert. Für den Gültigkeitsbereich und die Lebensdauer von Objektvariablen gilt ebenfalls dasselbe wie für Variablen im allgemeinen. Um einen bestimmten Typ von Objekt wie z.B. eine Tabelle oder ein Steuerelement als Objektvariable zu deklarieren, können Sie der Variablen z.B. einen der folgenden Objekttypen zuweisen:

Objekttyp	Gilt für
Database	Datenbank
Form	Formular
Report	Bericht
Control	Steuerelement in Formular oder Bericht
Container	Objekt, das Informationen zu anderen Objekten enthält
Document	Informationen über Objekte, die die JetEngine verwaltet
Field	Feld einer Tabelle, einer Abfrage, einer Datensatzgruppe, eines Index oder einer Beziehung
Group	Gruppenkonto in der aktuellen Arbeitsgruppe
Index	Tabellenindex
Parameter	Abfrageparameter
Property	Eigenschaft eines Objekts
QueryDef	Abfragedefinition
Recordset	Datensatzgruppe, die von einer Tabelle oder Abfrage definiert wurde
Relation	Beziehung zwischen zwei Tabellen oder Abfragefeldern
TableDef	Gespeicherte Tabelle
User	Benutzerkonto in der aktuellen Arbeitsgruppe
Workspace	Aktive Sitzung der Jet Datenbank-Engine

Anmerkung: Für Makros können keine Objektvariablen deklariert werden.

Um mit einer Objektvariablen arbeiten zu können, müssen Sie stets zwei Schritte ausführen:

- Dieser Schritt ist optional: Sie deklarieren eine Objektvariable für einen Objekttyp, der dem im nächsten Schritt zuzuweisenden Objekt entspricht. Wenn es sich z.B. um ein Formular handelt, müssen Sie eine Objektvariable vom Typ *Form* deklarieren, für ein Steuerelement eine Objektvariable vom Typ *Control* etc. Sie ordnen einer Variablen einen Objekttyp so zu, wie Sie einer üblichen Variablen einen Datentyp zuordnen, also mit dem Schlüsselwort *As*.

- Sie weisen der deklarierten Objektvariablen ein bestimmtes Objekt zu, z.B. ein bestimmtes Formular oder ein bestimmtes Steuerelement. Dies geschieht stets mit der Anweisung *Set* und nicht, wie bei der Zuweisung von Werten an einfache Variable, allein mit dem Gleichheitszeichen.

Nach diesen beiden Schritten können Sie das Objekt, welches Sie der Objektvariablen zugewiesen haben, als Variable ansprechen.

Das folgende Beispiel deklariert die Variable *FF1* als Objektvariable mit dem Objekttyp *Form*, weist ihr das konkrete Formular *Adressen* (dieses muß geöffnet sein) als Objekt zu und weist der Eigenschaft *Visible* des Formulars den Wert False zu, um es auszublenden.

```
Dim FF1 As Form          'Deklaration als Objektvariable
Set FF1 = Forms!Adressen 'Zuweisung des Formulars an FF1
FF1.Visible = False      'Zuweisung des Wertes False an die
                         'Eigenschaft Visible
```

30.3.2 Objekttyp *Database zuweisen*

Für das Zuweisen von Datenbanken zu einer Objektvariablen vom Typ *Database* stehen Ihnen die beiden Funktionen *CurrentDB* und *OpenDatabase* zur Verfügung.

Mit *CurrentDB* weisen Sie der Objektvariablen die aktuell geöffnete Datenbank zu, mit *OpenDatabase* eine nicht geöffnete. Anders als auf der allgemeinen Access-Ebene können in einer VBA-Prozedur mehrere Datenbanken gleichzeitig als Objektvariable geöffnet sein. Dies erweitert den Spielraum ganz erheblich.

Das folgende Beispiel deklariert die Objektvariablen DB1 und DB2 mit dem Objekttyp *Database* und weist DB1 die aktuelle Datenbank, DB2 die Datenbank *Auftrag.mdb* aus dem Verzeichnis DB des Laufwerks C: zu. Damit der Objekttyp *Database* explizit verfügbar ist, muß ein Verweis auf die Bibliothek *DAO* erstellt sein.

```
Dim DB1 As Database, DB2 As Database
Set DB1 = CurrentDB()
Set DB2 = OpenDatabase("c:\db\auftrag.mdb")
```

30.3.3 Objekttypen *Form und Report zuweisen*

Ein Formular oder einen Bericht können Sie ohne weiteres, d.h. ohne Verwendung einer Funktion oder Methode, zuweisen. Sie müssen zur Bezeichnung des Formulars bzw. Berichts allerdings dem Namen das Auflistungsobjekt *Forms* bzw. *Reports*, gefolgt von einem Ausrufezeichen (!), voranstellen. Das folgende Beispiel weist den Objektvariablen FF1 bzw. BB1, die als Variable des Typs *Form* bzw. *Report* deklariert werden, das Formular *Formular1* bzw. den Bericht *Bericht1* zu:

```
Dim FF1 As Form, BB1 As Report
DoCmd.OpenForm ("Formular1")
DoCmd.OpenReport ("Bericht1")
Set FF1 = Forms!Formular1
Set BB1 = Reports!Bericht1
```

Sie können das Formular oder den Bericht, das bzw. der den Fokus hat, auch über die Eigenschaft *ActiveForm* bzw. *ActiveReport* des vordefinierten Objekts *Screen* identifizieren und zuweisen. Das folgende Beispiel definiert die Objektvariable *FF1* vom Typ *Form,* weist ihr das aktuelle Formular zu und blendet dieses dann aus. Auf diese Weise können Sie sehr allgemeine Prozeduren für das jeweils aktuelle Formular bzw. den aktuellen Bericht schreiben.

```
Dim FF1 As Form
DoCmd.SelectObject A_Form, "Artikel"
Set FF1 = Screen.ActiveForm
FF1.Visible = False
```

Beachten Sie, daß das Formular bzw. der Bericht geöffnet sein muß, bevor es/er einer Objektvariablen zugewiesen werden kann. Sie müßten dies ggf. mit einer Anweisung *DoCmd.OpenForm* bzw. *DoCmd.OpenReport* in der Prozedur veranlassen. Um zu prüfen, ob ein bestimmtes Formular (bzw. Bericht) geöffnet ist, müssen Sie selbst eine kleine Funktion erstellen, die dies ermittelt, weil VBA für diesen Zweck keine Funktion zur Verfügung stellt. Im Modul *Dienstprogramme* der Datenbank Nordwind.mdb ist die Funktion *IsLoaded* zu finden, die Sie dafür verwenden können.

30.3.4 Objekttyp *Control zuweisen*

Ein Steuerelement weisen Sie direkt zu, d.h. ohne eine Funktion oder Methode verwenden zu müssen. Allerdings müssen Sie dem Steuerelementnamen den Namen des Formulars bzw. Berichts, in dem es enthalten ist, voranstellen. Das Auflistungsobjekt *Controls*, zu dessen Sammlung jedes Steuerelement gehört, können Sie dagegen fortlassen. Das folgende Beispiel deklariert die Objektvariable ST1 mit dem Objekttyp *Control,* weist ihr das Steuerelement *Firmenbezeichnung* aus dem Formular *Kunden* zu, das bei der Zuweisung *in der Formularansicht geöffnet* sein muß, und weist dem Steuerelement *Firmenbezeichnung* die Zeichenfolge *Hamburger Cigarrenfabrik* zu.

```
Dim ST1 As Control
Set ST1 = Forms!Kunden!Firmenbezeichnung
ST1 = "Hamburger Cigarrenfabrik"
```

Wenn das Formular *Kunden* bereits als Objektvariable existiert, können Sie bei der Zuweisung des Steuerelements darauf Bezug nehmen, wie in dem folgenden Beispiel gezeigt:

```
Dim FF1 As Form, ST1 As Control
Set FF1 = Forms!Kunden
Set ST1 = FF1!Firmenbezeichnung
ST1 = "Hamburger Cigarrenfabrik"
```

Sie können ein Steuerelement, das den Fokus hat, auch über die Eigenschaft *ActiveControl* des vordefinierten Objekts *Screen* identifizieren und zuweisen. Probieren Sie dazu die folgende Testfunktion (setzen Sie ggf. einen anderen Formularnamen ein):

```
Function AktSte
Dim AktSt As String, ST1 As Control
DoCmd.SelectObject A_Form, "Kunden"
Set ST1 = Screen.ActiveControl
AktSt = ST1.ControlName
MsgBox ("Aktuelles Steuerelement ist: " & AktSt)
End Function
```

30.3.5 Datenzugriffsobjekte zuweisen

Speziell für die Manipulation von Datensätzen in Tabellen, Abfragen und Formularen stehen die Objekte der Bibliothek *DAO (Data Access Objects)* zur Verfügung. Diesen kommt naturgemäß in einem RDBMS eine zentrale Bedeutung zu. Diese Objekte mit ihren Methoden und Eigenschaften werden in Kap. 31, *Datenzugriff mit DAO (Data Access Objects) und ADO (ActiveX Data Objects)*, behandelt.

30.4 Programmieren mit Klassenmodulen

In Access 2000 gibt es zwei Arten von Modulen: Standardmodule und Klassenmodule. Bereits in Access 95 gab es Klassenmodule, jedoch nur als Formular- oder Berichtsmodule. Dieses gilt nach wie vor: Jedes Formular- oder Berichtsmodul ist ein Klassenmodul. Seit Access 97 können Klassenmodule auch unabhängig von einem Formular oder Bericht erstellt werden. Sie werden dann in der Registerkarte *Module* des Datenbankfensters zusammen mit den Standardmodulen angezeigt. Klassen- und Standardmodule werden dabei durch verschiedene Symbole gekennzeichnet.

30.4.1 Unabhängiges Klassenmodul

Mit einem Klassenmodul können Sie eine Definition für ein benutzerdefiniertes Objekt erstellen. Der Name, unter dem Sie das Klassenmodul speichern, wird der Name des benutzerdefinierten Objekts. Öffentliche *Sub*- und *Function*-Prozeduren, die Sie in einem Klassenmodul definieren, werden zu benutzerdefinierten Methoden des Objekts. Öffentliche *Property Let*-, *Property Get*- und *Property Set*-Prozeduren werden zu Eigenschaften des Objekts.

Für ein Klassenmodul können Sie neue (benutzerdefinierte) Objekte erstellen, indem Sie eine neue Instanz der Klasse erstellen. Um eine neue Instanz einer Klasse zu erstellen, deklarieren Sie eine Variable des durch die Klasse definierten Typs. Wenn der Name eines Klassenmoduls z.B. *GB_Spezial* lautet, erstellen Sie eine neue Instanz dieser Klasse auf die folgende Weise:

```
Dim GB As New GB_Spezial
```

Sie können dann auf die öffentlichen Prozeduren des Klassenmoduls als Methoden bzw. Eigenschaften des Objekts, das in diesem Beispiel als Objektvariable *GB* verfügbar ist, zugreifen.

Beispiel

Aufgabe. Erstellen Sie unter dem Namen *GB_Spezial* ein Klassenmodul. Definieren Sie darin eine Function-Prozedur, die den Pfad der aktuellen Datenbank ausgibt. Definieren Sie in einem anderen Modul ein benutzerdefiniertes Objekt, das diese Funktion als Methode des benutzerdefinierten Objekts einsetzt.

Gehen Sie folgendermaßen vor, um diese Aufgabe zu lösen:

- Wählen Sie den Befehl *Klassenmodul* aus dem Menü *Einfügen*; dieser Befehl steht nur zur Verfügung, wenn das Datenbankfenster oder ein Modulfenster aktiv ist.
- Erstellen Sie die folgende Function-Prozedur:

```
Function DB_Pfad()
Dim DB
Set DB = CurrentDb()
DB_Pfad = DB.Name
End Function
```

- Speichern Sie das Klassenmodul unter dem Namen *GB_Spezial*.
- Öffnen Sie ein Standardmodul.
- Erstellen Sie in diesem Standardmodul die folgende Prozedur:

```
Sub Objekttest()
Dim GB As New GB_Spezial
MsgBox GB.DB_Pfad
End Sub
```

Der Prozedurname *Objekttest* und der Name *GB* für die Objektvariable sind beliebig, nicht jedoch die Namen *GB_Spezial* für die Klasse und *DB_Pfad* für die Methode, da beide mit dem bzw. im Klassenmodul vergeben sind und an dieser Stelle darauf referenziert wird. Wenn Sie die vorangehende Prozedur aufrufen, wird der Pfad der aktuellen Datenbank in einem Meldungsfeld ausgegeben.

30.4.2 Klassenmodul eines Formulars oder Berichts

Wenn Sie ein Formular in der Formularansicht öffnen (von der Benutzeroberfläche oder von VBA aus), erstellen Sie eine Instanz des Klassenmoduls dieses Formulars. Damit reservieren Sie Platz im Arbeitsspeicher, in dem das Objekt existiert, solange es geöffnet ist. Sie können dann mit VBA-Prozeduren die Methoden des Formulars aufrufen und seine Eigenschaften im Code einstellen oder zurückgeben wie für jedes eingebaute Objekt. Entsprechendes gilt für einen Bericht, wenn Sie diesen in der Seitenansicht öffnen. Sie arbeiten in diesen Fällen mit der Standardinstanz der Klasse des Formulars/Berichts. (Im weiteren wird aus Gründen sprachlicher Vereinfachung nur von Formularen gesprochen.) Sie haben jedoch auch die Möglichkeit, weitere Instanzen derselben Klasse eines Formulars zu erstellen und damit das Formular ggf. mehrfach zu öffnen. In diesen Fällen erstellen Sie Nicht-Standardinstanzen.

Sie können mehrere Nicht-Standardinstanzen einer Klasse eines Formulars erstellen, um beispielsweise mehrere Instanzen des Formulars gleichzeitig anzuzeigen. Für ein Formular mit Datensätzen von Mitarbeitern kann dies z.B. nützlich sein, um die Angaben von zwei Mitarbeitern gleichzeitig betrachten zu können. Sie können dazu eine Instanz des Formulars zur Anzeige des Datensatzes von Mitarbeiter A erstellen und eine Instanz zur Anzeige des Datensatzes von Mitarbeiter B.

Um Nicht-Standardinstanzen einer Klasse eines Formulars mit VBA-Code zu erstellen, deklarieren Sie eine Variable, deren Typ der Klassenname des Formularmoduls ist. Für ein Formular mit dem Namen *Personal* beispielsweise hat das Klassenmodul den Namen *Form_Personal*. Verwenden Sie in der Variablendeklaration das Schlüsselwort *New*. Die folgende Codezeile beispielsweise erstellt eine neue Instanz des Formulars *Personal* und weist sie der Variablen *frmPM* mit dem Datentyp *Form* zu.

```
Dim frmPM As New Form_Personal
```

Beispiel

In der Datenbank *Projekte.mdb* von der Begleit-CD-ROM finden Sie ein Formular mit dem Namen *PersonalMehrfach*. Dieses Formular ist eine Kopie des ursprünglichen Formulars *Personal*, das in Kap. 5, *Einführungsbeispiel: Eine einfache relationale Datenbank erstellen*, erzeugt wurde. Es enthält allerdings zusätzlich die Schaltfläche *Weiteres Formular*. Wenn Sie das Formular *PersonalMehrfach* öffnen und dann auf die Schaltfläche *Weiteres Formular* klicken, wird das Formular *PersonalMehrfach* ein weiteres Mal angezeigt, vgl. Bild 30.3. Sie können dann jedes der beiden Formulare unabhängig voneinander bearbeiten. Das heißt insbesondere auch, daß Sie in jedem Formular zu einem anderen Datensatz blättern können.

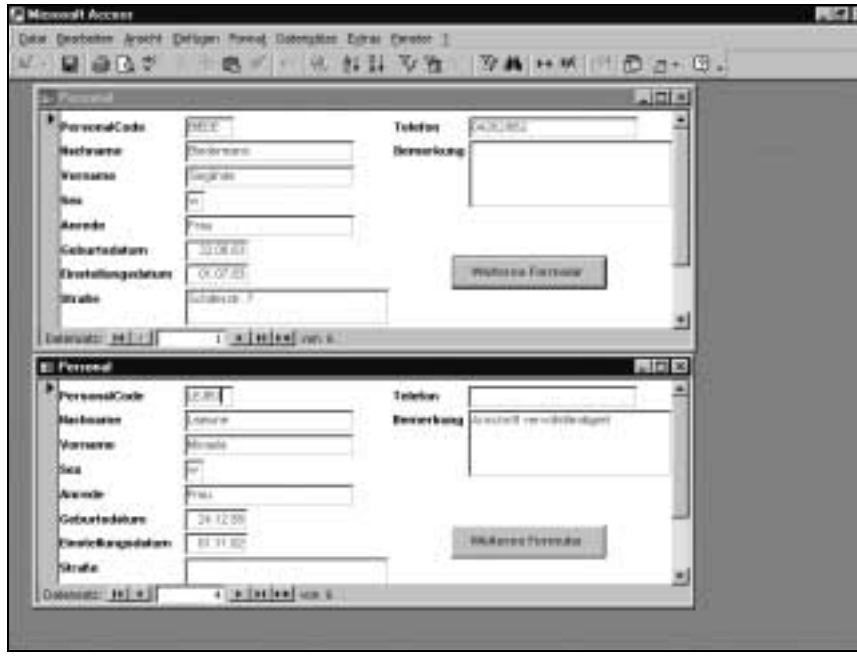


Bild 30.3: Zwei Instanzen desselben Formulars »PersonalMehrfach«

Um diese Funktionalität zu erreichen, gehen Sie folgendermaßen vor:

- Erstellen Sie in einem Standardmodul die folgende Prozedur. Beachten Sie dabei, daß die Deklaration der Variablen *frmPM* außerhalb der Prozedur erfolgt (vgl. den Kommentar innerhalb der Prozedur *FrmPersonalMehrfachAlsNeueInstanzÖffnen*).

```
Dim frmPM As Form
Sub FrmPersonalMehrfachAlsNeueInstanzÖffnen()
'Das Objekt frm1 ist im Deklarationsteil des Moduls
'deklariert worden, damit das Formular über die
'Lebensdauer der Prozedur Bestand hat. Bei Deklaration
'auf Prozedurebene würde es mit Beendigung der Prozedur
'wieder geschlossen werden.
Set frmPM = New Form_PersonalMehrfach
frmPM.SetFocus
End Sub
```

- Weisen Sie dem Ereignis *Beim Klicken* der Schaltfläche *Weiteres Formular* des Formulars *PersonalMehrfach* eine Ereignisprozedur zu, welche die voranstehende Prozedur *FrmPersonalMehrfachAlsNeueInstanzÖffnen* aufruft.

Alle Eigenschaften, die Sie einstellen, beeinflussen diese Instanz der Klasse des Formulars, werden jedoch nicht mit dem Formular gespeichert. Auch kann keine neue Instanz der Klasse des Formulars erstellt werden, wenn das Formular in der Entwurfsansicht geöffnet ist.

Auf eine Nicht-Standardinstanz eines Formulars kann nicht über den Namen in der Auflistung *Forms* verwiesen werden. Es sind nur Verweise über die Indexnummer möglich. Da Sie mehrere Nicht-Standardinstanzen eines Formulars erstellen können und jede Instanz den gleichen Namen hat, kann die Auflistung *Forms* mehrere gleichnamige Formulare enthalten, die sich lediglich durch ihre Indexnummer unterscheiden.

30.5 Fehlersuche

Die Ausführungen dieses Punktes beziehen sich nur auf Fehler, die in VBA-Prozeduren auftreten. Im Punkt 30.6, *Fehlerbehandlung*, wird dagegen auch auf die Behandlung von Fehlern eingegangen, die unabhängig von VBA auf der allgemeinen Access-Ebene auftreten können.

Unterbrechungsmodus

Wenn in einer Prozedur ein Laufzeitfehler auftritt, wird die Programmausführung unterbrochen, und VBA schaltet in den Unterbrechungsmodus um, es sei denn, Sie haben für diesen Fall in der Prozedur eine entsprechende Fehlerbehandlungsroutine vorgesehen, vgl. den folgenden Punkt 30.6, *Fehlerbehandlung*.

Für das Debugging stellt VBA mehrere Werkzeuge zur Verfügung, die Sie anwenden können, wenn sich das Programm im Unterbrechungsmodus befindet. VBA schaltet unter jeder der folgenden Bedingungen in den Unterbrechungsmodus:

- Es trat ein Laufzeitfehler auf.
- In einer Zeile befindet sich ein von Ihnen gesetzter Haltepunkt. Haltepunkte werden zum Zwecke des Unterbrechens gesetzt. In einer Prozedur können viele Zeilen einen Haltepunkt enthalten. Das Programm hält beim jeweils nächsten an. Wie Sie Haltepunkte setzen, wird etwas weiter unten in diesem Abschnitt erklärt.
- Eine Zeile enthält die Anweisung *Stop*. Hierfür gilt Entsprechendes wie für Haltepunkte.
- Sie drücken während der Programmausführung die Tastenkombination Strg + Pause.

Wenn VBA in den Unterbrechungsmodus wechselt, hält das Programm an der Zeile mit dem Laufzeitfehler, dem Haltepunkt, *Stop*-Anweisung oder der

mit Strg+Pause »erwischten« an und hebt diese Zeile im Modulfenster durch farbliche Markierung (Voreinstellung: gelb) hervor. Sie können in dieser Situation eine der folgenden Möglichkeiten wählen:

- Lassen Sie das Programm vom Unterbrechungspunkt an in Einzelschritten weiter ausführen. Dafür stehen Ihnen die beiden Befehle *Einzelschritt* und *Prozedurschritt* aus dem Menü *Debuggen* zur Verfügung, die Sie auch mit den gleichnamigen Symbol-Schaltflächen wählen können. Bei der Einzelschrittmethode wird nach jedem Wählen des Befehls *Einzelschritt* die nächste und nur diese Zeile der Prozedur ausgeführt. Auf diese Weise können Sie sich schrittweise bis zum nächsten Fehler vortasten. Falls die Prozedur eine andere Prozedur aufruft, würden deren Programmzeilen bei der Einzelschrittmethode ebenfalls Schritt für Schritt abgearbeitet werden. Wenn Sie Prozeduren in der betreffenden Prozedur dagegen wie eine Programmzeile behandelt wissen wollen, wählen Sie den Befehl *Prozedurschritt*.
- Starten Sie das Programm mit dem Befehl *Weiter* aus dem Menü *Ausführen*, für den Sie auch auf die gleichnamige Symbol-Schaltfläche klicken können. Dann wird der gesamte noch nicht ausgeführte Programmrest auf einmal ausgeführt, es sei denn, es kommen weitere Fehler, Haltepunkte oder *Stop*-Anweisungen vor.
- Initialisieren Sie den gesamten Code neu mit dem Befehl *Zurücksetzen* aus dem Menü *Ausführen*, für den Sie auch auf die gleichnamige Symbol-Schaltfläche klicken können. Dieser Weg ist manchmal erforderlich, um überhaupt wieder eine Prozedur starten zu können. Die Wirkung ist entsprechend weitreichend: U.a. werden alle Variablen neu initialisiert, so daß z.B. auch diejenigen mit modularem oder öffentlichem Geltungsbereich ihre bisherigen Werte verlieren.

Haltepunkte setzen und löschen

Sie setzen einen Haltepunkt, indem Sie den Cursor in die betreffende Zeile setzen und den Befehl *Haltepunkt ein/aus* aus dem Menü *Debuggen* (auch Taste F9) wählen oder auf die entsprechende Symbol-Schaltfläche klicken. Alternativ: Klicken Sie an der betreffenden Stelle auf die graue Kennzeichenleiste am linken Rand des Code-Fensters. Der Befehl wirkt als Ein/Ausschalter, so daß Sie auf entsprechende Weise einen Haltepunkt wieder aufheben können. Zeilen mit einem Haltepunkt werden im Modulfenster farblich hervorgehoben (Voreinstellung: rotbraun). Mit dem Befehl *Alle Haltepunkte löschen* aus dem Menü *Debuggen* können Sie alle Haltepunkte mit einem Mal zurücksetzen.

Nächste auszuführende Anweisung festlegen

Wenn Sie die Programmausführung an einer anderen Stelle als der unterbrochenen fortsetzen möchten, können Sie dies mit dem Befehl *Nächste Anweisung festlegen* aus dem Menü *Debuggen* erreichen: Setzen Sie, während sich die Prozedur im Unterbrechungsmodus befindet, den Cursor in die gewünschte Programmzeile, und führen Sie den Befehl *Nächste Anweisung festlegen* aus. Dann wird das Programm von dieser Zeile aus fortgesetzt werden, wenn Sie dies mit einer der oben angegebenen Methoden veranlassen.

Eine andere Methode, bestimmte Programmteile bei der Programmausführung zu umschiffen, besteht darin, eine Anweisung *Goto* mit entsprechender Zeilenmarke in das Programm zu schreiben. Eine Zeilenmarke ist eine Codezeile mit einem beliebigen, jedoch im Modul eindeutigen Namen, der mit einem Doppelpunkt endet. Für das Aussparen kurzer Programmteile können Sie einzelne Zeilen auch (vorübergehend) zum Kommentar machen, indem Sie an ihren Anfang ein Apostroph (') schreiben.

30.6 Fehlerbehandlung

Trotz sorgfältiger Konzeption und Ausführung einer Anwendung müssen Sie sich darauf einstellen, daß unter gewissen Bedingungen Fehler auftreten werden. Ihre Anwendung wird um so sicherer und benutzerfreundlicher, je mehr Sie dem Benutzer ermöglichen, auf Fehler kontextbezogen zu reagieren. Aus diesem Grunde kann es sinnvoll sein, Routinen vorzusehen, die auftretende Fehler auffangen, statt beispielsweise lediglich den schlichten Abbruch einer Aufgabe oder gar den Absturz der gesamten Anwendung als Konsequenz in Kauf zu nehmen. Derartige Vorsorge wird Fehlerbehandlung genannt.

Bei der Fehlerbehandlung müssen Sie zwei grundsätzlich verschiedene Arten von Fehlern unterscheiden:

- **Allgemeine Access-Ebene.** Hiermit sind Fehler gemeint, die auch auftreten können, wenn in der Anwendung überhaupt kein VBA-Code existiert. Wenn Sie beispielsweise in ein Formular für ein Primärschlüsselfeld einen Wert eingeben, der bereits existiert, wird dies von Access als Fehler betrachtet, auf den das System mit einer Unterbrechung der Aufgabe und einer Meldung reagiert. Fehler, die in der allgemeinen Access-Ebene bei der Verwendung von Formularen und Berichten auftreten, können Sie mit Ereignisprozeduren abfangen.
- **VBA-Laufzeitfehler.** Wenn bei der Ausführung einer VBA-Prozedur (d.h. während der Laufzeit, daher der Ausdruck *Laufzeitfehler*) ein Fehler auftritt, wird die Prozedur abgebrochen. Dabei ist es gleichgültig, ob die Fehlerursache direkt im Programmcode liegt, oder ob es sich um einen allgemeinen Access-Fehler handelt, wie er im vorangehenden Aufzählungspunkt beschrieben wurde. Sie können den Abbruch der Prozedur allerdings verhindern, wenn Sie in der Prozedur eine Fehlerbehandlungs-

routine vorsehen. Mit einer Ereignisprozedur können Sie VBA-Laufzeitfehler nicht auffangen.

Im weiteren wird die Behandlung von Fehlern auf der allgemeinen Access-Ebene mit Ereignisprozeduren einerseits und die Behandlung von VBA-Laufzeitfehlern andererseits in getrennten Punkten besprochen.

30.6.1 Fehlercodes

Wenn Sie gezielt auf bestimmte Fehler reagieren wollen, müssen Sie diese identifizieren können. Access ordnet jeder Fehlerart eine Zahl zu, den *Fehlercode*. Dies ist ein Integerwert. Wenn Sie eine Routine erstellen wollen, die die Folgen dieses bestimmten Fehlers behandeln soll, müssen Sie sich auf diesen Fehlercode beziehen. Dazu müssen Sie über zwei Informationen verfügen:

- Um in einer Fehlerbehandlungsroutine auf einen oder ein paar bestimmte(n) Fehler reagieren zu können, müssen Sie – unabhängig von einer konkreten Fehlersituation – die möglichen Fehler und ihre Fehlercodes kennen. Mit der Funktion *Error* ist es möglich, den Meldungstext, der einem bestimmten Fehlercode zugeordnet ist, ausgeben zu lassen.
- Wenn ein konkreter Fehler aufgetreten ist, müssen Sie seinen Fehlercode ermitteln, um davon die konkrete Fehlerbehandlung abhängig zu machen. Der Fehlercode des aufgetretenen Fehlers wird in einer Ereignisprozedur anders ermittelt als in einer allgemeinen VBA-Prozedur. In einer Ereignisprozedur übergibt Access den Fehlercode des aufgetretenen allgemeinen Access-Fehlers an die speziell für diesen Zweck in der Ereignisprozedur definierte Variable *DataErr*. Wenn dagegen ein Laufzeitfehler in einer VBA-Prozedur auftritt, wird dieser mit der Funktion *Err* ermittelt. Die Funktion *Err* gibt den Fehlercode 0 aus, wenn während der Laufzeit der Prozedur zwar ein allgemeiner Access-Fehler, jedoch kein VBA-Fehler aufgetreten ist.

30.6.2 Fehlerbehandlung auf der allgemeinen Access-Ebene

Das Auftreten eines Fehlers ist für Access ein Ereignis. Daher besitzen Formulare und Berichte die Eigenschaft *Bei Fehler*. Sie können also im Prinzip mit einem Makro oder einer Ereignisprozedur auf einen Fehler reagieren. Allerdings können Sie mit einem Makro den Fehler nicht auffangen in dem Sinne, daß Sie auf den spezifischen Fehler reagieren, weil Sie mit einem Makro weder die Art des Fehlers (den Fehlercode) ermitteln, noch eine flexible Reaktion darauf vorsehen können. Andererseits unterstützt Access Fehler-Ereignisprozeduren, mit denen Sie Fehler der allgemeinen Access-Ebene auffangen können.

Wenn Sie für ein Ereignis eines Formulars, Berichts oder Steuerelements eine Ereignisprozedur erstellen wollen, bietet Access Ihnen dafür stets die erste und letzte Zeile der Prozedur vordefiniert an. Bei einer Ereignisprozedur für das Ereignis *Bei Fehler* (gilt für Formulare und Berichte) sehen die erste und letzte Zeile der Prozedur folgendermaßen aus:

```
Private Sub Form_Error(DataErr As Integer, _
  Response As Integer)

End Sub
```

Die Besonderheit besteht darin, daß in diesem Falle die beiden Prozedur-Argumente *DataErr* und *Response* definiert werden, jedes vom Datentyp *Integer*. Diese beiden Argumente haben die folgende Aufgabe:

- **DataErr**. Access übergibt an das Argument *DataErr* den Fehlercode des aufgetretenen Fehlers, so daß Sie diesen innerhalb der Ereignisprozedur unter Bezugnahme auf das Argument *DataErr* identifizieren können.
- **Response**. Das Argument wird nach fehlerfreier Beendigung der Ereignisprozedur von Access ausgelesen. Wenn Sie diesem Argument während der Ereignisprozedur die vordefinierte Konstante *acDataErrContinue* zuweisen, wird die Standardfehlermeldung des aufgetretenen Fehlers unterdrückt. Für diesen Fall sollten Sie eine Meldung einblenden, die den Benutzer zu einer angemessenen Reaktion auf den Fehler auffordert.

Beispiel

Die Beispieldatenbank *Nordwind.mdb* enthält u.a. das Formular *Bestellungen* mit dem Unterformular *Bestellungen Unterformular*. Für die Ereigniseigenschaft *Bei Fehler* dieses Unterformulars ist eine Ereignisprozedur definiert, die alle Fehler mit einem Fehlercode >0 – das sind alle möglichen Fehler – auffängt: Wenn der Benutzer einen Datensatz in das Unterformular eingibt, ohne daß im Hauptformular ein Kunde bestimmt ist, tritt ein Fehler auf, der durch die folgende Ereignisprozedur des Unterformulars *Bestellungen Unterformular* aufgefangen wird[1]:

```
Private Sub Form_Error(DataErr As Integer, _
  Response As Integer)
 'Hier können Fehler auftreten, wenn der Benutzer
 'nicht bereits einen Kunde für diese Bestellung im
 'Hauptformular ausgewählt hat. Wenn ein Fehler auftritt
 'und "Bestellungen!Rechnung an" Null ist,
 'den Benutzer informieren.

If DataErr > 0 Then
    If IsNull(Me.Parent![Kunden-Code]) Then
     MsgBox "Wählen Sie einen Kunde aus, an den die " & _
     "Rechnung geschickt werden soll, bevor Sie " & _
     "Bestelldetails eingeben."
     RunCommand acCmdUndo
```

1. Die Zeilenumbrüche habe ich vorgenommen, der kleine sprachliche Fehler (»einen Kunde«) indessen ist original.

```
        Me.Parent![Kunden-Code].SetFocus
        Response = acDataErrContinue
       Else
        Response = acDataErrDisplay
       End If
   End If
   End Sub
```

Funktionsweise der Ereignisprozedur *Form_Error des Unterformulars Bestellungen Unterformular.* Die Prozedur wird aufgerufen, wenn bei der Bearbeitung des Unterformulars *Bestellungen Unterformular* ein Fehler auftritt. Der Fehlercode des aufgetretenen Fehlers wird von Access automatisch an das Argument *DataErr* übergeben. Mit der Anweisung

```
If DataErr > 0 Then
```

wird zunächst geprüft, ob überhaupt ein Fehler, auf den reagiert werden soll, aufgetreten ist. Mit der Figur

```
       If IsNull(Me.Parent![Kunden-Code]) Then
        MsgBox "Wählen Sie einen Kunde aus, an den die " & _
        "Rechnung geschickt werden soll, bevor Sie " & _
        "Bestelldetails eingeben."
        RunCommand acCmdUndo
        Me.Parent![Kunden-Code].SetFocus
        Response = acDataErrContinue
       Else
        Response = acDataErrDisplay
       End If
```

wird zunächst geprüft, ob das Steuerelement *Kunden-Code* den Wert NULL enthält. Falls dies zutrifft, wird eine Meldung ausgegeben und die zuletzt ausgeführte Aktion wird rückgängig gemacht mit der Codezeile

```
        RunCommand acCmdUndo
```

Außerdem wird dem Steuerelement *Kunden-Code* der Fokus gegeben und mit der Zeile

```
        Response = acDataErrContinue
```

dafür gesorgt, daß die Standardfehlermeldung des aufgetretenen Fehlers unterdrückt wird.

Falls ein anderer Fehler als Nullwert im Steuerelement *Kunden-Code* aufgetreten ist, wird mit der Zeile

```
        Response = acDataErrDisplay
```

dafür gesorgt, daß die entsprechende Fehlermeldung angezeigt wird.

30.6.3 Fehlerbehandlung von VBA-Laufzeitfehlern

Wenn ein VBA-Laufzeitfehler auftritt, hält das VBA-Programm an, so daß die Prozedur nicht fortgesetzt wird. Da Sie nicht alle möglichen auftretenden Fehler voraussehen können, weil einige beispielsweise nur in bestimmten Kontexten auftreten, die Sie noch nie getestet haben, kann es passieren, daß ein Programmfehler bei der Benutzung Ihrer Datenbankanwendung durch Dritte auftritt. Unter gewissen Bedingungen kann dies zur Folge haben, daß sich kein einziger Access-Befehl mehr ausführen läßt. Daher sollten Sie in Prozeduren, die nicht offensichtlich in jeder Hinsicht ausgetestet sind, eine Routine vorsehen, die vordefinierte Folgen für den Fall eines Laufzeitfehlers festlegt. Es gibt in VBA keine Möglichkeit, solche Fehlerbehandlungsroutinen für den gesamten Programmcode festzulegen. Dies muß vielmehr für jede einzelne Prozedur geschehen.

Fehlerbehandlungsroutinen für VBA-Laufzeitfehler werden mit der Anweisung *On Error* geschrieben. Sie haben die folgende Syntax:

On Error {**GoTo** Programmzeile | **Resume Next** | **GoTo 0**}

Ohne die Anweisung *On Error* ist jeder auftretende Laufzeitfehler ein fataler Fehler, in dessen Folge die Prozedurausführung mit einer System-Fehlermeldung beendet wird. Durch die Anweisung *On Error* wird die Prozedur beim Auftreten eines Fehlers nicht abgebrochen, und es erscheint auch keine System-Fehlermeldung, sondern es wird eine benutzerdefinierte Fehlerbehandlungsroutine in Gang gesetzt, mit der ggf. eine Korrektur der Fehlerursache (z.B. Einlegen einer Diskette in das Floppy-Laufwerk) ermöglicht oder die Prozedur auf eine benutzerdefinierte Weise fortgesetzt oder beendet wird.

Die folgenden drei Argumente sind alternativ; eines von ihnen muß jedoch angegeben werden.

Argument	Beschreibung
GoTo Programmzeile	Aktiviert die Fehlerbehandlungsroutine, die an der Stelle *Programmzeile* beginnt. *Programmzeile* ist eine Zeilennummer oder eine Zeilenmarke, die sich in derselben Prozedur befindet, in der auch *On Error* steht.
Resume Next	Bestimmt, daß die Programmausführung mit der Anweisung fortgesetzt wird, die der Anweisung, welche den Fehler verursachte, unmittelbar folgt. Damit ergibt sich eine Wirkung, als ob gar keine Fehlerprüfung erfolgte, da das Programm trotz eines oder mehrerer auftretender Fehler(s) fortgesetzt wird.
GoTo 0	Deaktiviert jeden in der aktuellen Prozedur aktivierten Fehlerbehandler.

Sprachlich ist zwischen einer aktivierten und einer aktiven Fehlerbehandlungsroutine zu unterscheiden: Eine Fehlerbehandlungsroutine wird aktiviert, wenn *On Error GoTo Programmzeile* ausgeführt wird; sie ist dann jedoch noch nicht aktiv. Sie wird aktiv, wenn ein Fehler auftritt. Solange eine einmal aktivierte Fehlerbehandlungsroutine nicht ausdrücklich deaktiviert wird, wird nach jedem weiteren Auftreten eines Fehlers zu ihr verzweigt.

Eine Fehlerbehandlungsroutine ist ein Programmteil in der aktuellen Prozedur, der mit *Programmzeile* beginnt und mit *Resume* oder *Resume Next* endet.

Die Anweisung *On Error* darf in derselben Prozedur mehrfach angeführt werden; dann wird jedesmal eine andere Fehlerbehandlungsroutine aktiviert bzw., im Falle von *On Error GoTo 0*, die bis dahin aktivierte deaktiviert.

Um zu verhindern, daß ein Fehlerbehandlungscode ausgeführt wird, wenn kein Fehler aufgetreten ist, führen Sie unmittelbar vor der Fehlerbehandlungsroutine eine der Anweisungen *Exit Sub* oder *Exit Function* an, wie es das folgende Beispiel zeigt:

```
Sub RechnungAktualisieren(Var1, Var2)
   On Error GoTo FBehandler
   ...
   Exit Sub
FBehandler:
   ...
   Resume Next
End Sub
```

Beispiel

Im folgenden Beispiel wird die Fehlerbehandlungsroutine, die bei der Zeilenmarke (eine Zeilenmarke wird durch endenden Doppelpunkt definiert) *Fehlerbehandler1:* beginnt, ausgeführt, wenn der Benutzer das erste Mal einen Nennerwert von Null eingibt. Diese erste Fehlerbehandlungsroutine fordert den Benutzer erneut zur Eingabe eines Nennerwertes auf und aktiviert eine zweite Fehlerbehandlungsroutine, nämlich *Fehlerbehandler2:*. Zu dieser wird beim Auftreten des nächsten Fehlers verzweigt. Wenn dieser Fall eintritt, wird die Meldung »Sie haben Ihre zwei Chancen verspielt!« ausgegeben und die Funktion beendet.

```
On Error GoTo Fehlerbehandler1
   Nenner = InputBox("Geben Sie den Nenner ein:")
   a = 2 / Nenner
   Exit Function
```

```
Fehlerbehandler1:
   MsgBox "Sie haben einen Nenner von Null eingegeben!"
   Nenner = InputBox("Geben Sie den Nenner erneut ein:")
   On Error GoTo Fehlerbehandler2
Resume
Fehlerbehandler2:
   MsgBox "Sie haben Ihre zwei Chancen verspielt!"
Exit Function
```

Wenn Sie gezielt auf einen bestimmten Fehler reagieren wollen, können Sie den Fehlercode des auslösenden Fehlers mit der Funktion *Err* abfragen. Das folgende Beispiel unterscheidet zwischen den Fehlerarten mit den Fehlercodes 11 bzw. 13, denen die Standard-Meldungstexte »Division durch Null« bzw. »Typen unverträglich« zugeordnet sind.

```
On Error GoTo Fehlerbehandler1
    Nenner = InputBox("Geben Sie den Nenner ein:")
    a = 2 / Nenner
    Exit Function
Fehlerbehandler1:
Select Case Err
    Case 11
        MsgBox "Sie haben einen Nenner von Null eingegeben!"
        Nenner = InputBox("Geben Sie den Nenner erneut ein:")
    Case 13
        MsgBox "Sie haben keinen Zahlenwert eingegeben!"
        Nenner = InputBox("Geben Sie den Nenner erneut ein:")
    Case Else
        MsgBox "Allgemeiner Fehler !"
End Select
Resume
```

Kapitel 31

Datenzugriff mit DAO (Data Access Objects) und ADO (ActiveX Data Objects)

31.1 Strategiewechsel im Datenzugriffskonzept

Mit Office 2000 und damit auch in Access 2000 realisiert Microsoft eine neue Datenzugriff-Strategie. Standen für den Datenzugriff mittels Programmierung bisher die Konzepte *DAO*, *ODBCDirect* und *ODBC* zur Verfügung und wurden auch von Microsoft favorisiert, so setzt die Firma nunmehr für die Zukunft auf die Konzepte OLE DB und dessen Derivate für VB- und VBA-Programmierer ADO, ADOX (ADO-Erweiterungen) und ADOMD (für multidimensionalen Datenzugriff). Gegenüber den bisherigen Konzepten werden mit dem neuen Datenzugriffskonzept laut Microsoft verschiedene Verbesserungen erreicht, insbesondere

- breiterer Zugriff auf Datenquellen (auch wenig verbreitete werden einbezogen),
- Verminderung des benötigten Arbeitsspeichers und der DLLs,
- leichterer Umgang mit den Objekten, Methoden und Eigenschaften der neuen Datenzugriffsobjekte.

Diese Vorteile sind unbestritten. Gleichwohl konnte (und durfte) Microsoft nicht abrupt ausschließlich und ersatzlos auf das neue Datenzugriffskonzept umstellen. Dies hat nicht nur Gründe der Kompatibilität mit Access-Datenbanken älterer Versionen, sondern erklärt sich auch aus gewissen Beschränkungen, die ADO gegenüber DAO (noch) besitzt. Beispielsweise ist ein *Recordset*-Objekt für eine Jet-Datenbank, das über die (in Access 2000 neue) Formular-Eigenschaft *Recordset* erzeugt wurde, unter DAO aktualisierbar, unter ADO dagegen nur lesbar. Aber auch die Kompatibilität mit

Access-Datenbanken älterer Versionen stellt zumindest zum gegenwärtigen Zeitpunkt ein starkes Argument dar, in erster Linie oder gar vollständig von DAO auf ADO umzusteigen, denn es gelten die folgenden Konsequenzen für die beiden Datenzugriffsobjekte bei der Konversion und Neubildung von Access-Datenbanken:

- Wenn eine Datenbank einer älteren Version nach Access 2000 konvertiert wird, verwendet die konvertierte Datenbank DAO 3.6 (das ist die zur aktuellen JetEngine 4.0 gehörige DAO-Bibliothek), die abwärtskompatibel zu den Access-Versionen 2.0 bis 97 ist.
- Konvertierte Client/Server-Datenbanken, die eine ODBC-Verbindung nutzen, verwenden weiterhin ODBC.
- Alle neu unter Access 2000 erzeugte Datenbanken und Projekte verwenden ADO 2.1, ADOX 2.1 und OLE DB 2.0.
- Da neu unter Access 2000 erzeugte Datenbanken die JetEngine 4.0 verwenden, kann auch in diesen Fällen DAO statt ADO eingesetzt werden. Lediglich für die Verbindung eines Access-Projekts zum SQL-Server oder zur MSDE (Microsoft Data Engine) wird ADO und OLE DB benötigt.
- Für Verweise auf die entsprechenden Objektbibliotheken ergeben sich damit diese Konsequenzen: Für neu in Access 2000 erzeugte Datenbanken und Projekte (diese können ohnehin nur in Access 2000 gebildet worden sein) wird automatisch ein Verweis auf die Microsoft ActiveX Data Objects 2.1 Library eingerichtet, ein Verweis auf eine DAO-Bibliothek fehlt dagegen. Für nach Access 2000 konvertierte Datenbanken früherer Versionen wird dagegen ein Verweis auf die *Microsoft DAO 3.6 Object Library* eingerichtet, während ein Verweis auf eine ADO-Bibliothek fehlt. Wenn Sie über die jeweils nicht automatisch referenzierten Bibliotheken verfügen wollen, können Sie einen entsprechenden Verweis jedoch manuell einrichten. Der Dateiname für die aktuelle DAO-Bibliothek lautet *Dao360.dll*, für die aktuelle ADO-Bibliothek *msado15.dll*.

Aus diesen Gründen konzentriere ich mich im folgenden weiterhin auf das Datenzugriffsobjekt DAO und gehe am Ende dieses Kapitels bezüglich ADO nur auf die obersten Objektebenen ein, so daß Sie in der Lage sein werden, über ein *Recordset*-Objekt auch unter ADO zu verfügen. Die Methoden und Eigenschaften des DAO-Recordset-Objekts, die ausführlich auch an Beispielen erklärt werden, können im übrigen weitgehend auf ein ADO-Recordset-Objekt übertragen werden.

31.2 Das DAO-Objektmodell

Für den Datenzugriff auf den Kern einer Jet-Datenbank, im wesentlichen also auf Tabellen, Beziehungen, Abfragen und Zugriffsrechte, stellt Access eine eigene Objektbibliothek zur Verfügung – *Data Access Objects*, kurz DAO genannt. Die in dieser Bibliothek vereinten Objekte mit ihren Metho-

den und Eigenschaften bieten, zusammen mit der JetEngine, umfassende Möglichkeiten, auf Daten lesend, schreibend und definierend zuzugreifen. Die Reichweite von DAO geht im übrigen weit über die Anwendung von Access hinaus, denn Microsoft stellt diese Bibliothek allen Office 2000-Anwendungen zur Verfügung, so daß Sie beispielsweise von Excel, Powerpoint oder Word aus über dieselben Möglichkeiten des Datenzugriffs verfügen wie von Access aus. Dies bedeutet u.a., daß Sie sich den Umgang mit DAO nur einmal anzueignen brauchen, wobei es gleichgültig ist, ob Sie dies im Zusammenhang mit VBA-Projekten von Excel, Access oder einer anderen Office 2000-Komponente tun.

Die DAO-Objekte sind hierarchisch aufgebaut, vgl. Bild 31.1. In der Hierarchie an oberster Stelle steht das Objekt *DBEngine*, das ist die Datenbankmaschine selbst. Darin enthalten sind in der nächsten Hierarchieebene die Objekte *Errors* und *Workspaces*. Diese sind zugleich Auflistungsobjekte. Als Auflistung wird ein Objekt dadurch identifiziert, daß ihm das Zeichen *s* für den Plural angefügt wird, beispielsweise *Workspaces*, *Databases*, *TableDefs* usw.

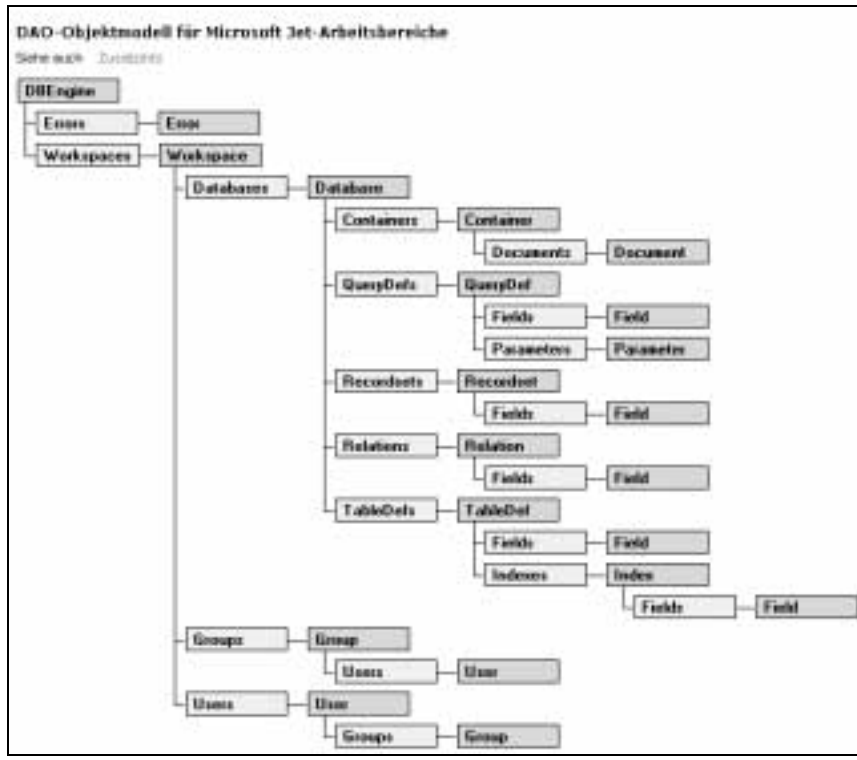


Bild 31.1: Hierarchie der DAO-Objekte

Wir können uns im Rahmen des verfügbaren Platzes nicht mit allen DAO-Objekten und ihren Methoden und Eigenschaften im einzelnen beschäftigen. Die wichtigsten Objekte sind für den praktischen Gebrauch *Database*, *TableDef*, *QueryDef* und *Recordset*. Deren Bedeutung sei zunächst kurz erklärt:

Database. Wie die Übersicht in Bild 31.1 zeigt, ist dieses Objekt Container u.a. für die Objekte *TableDef*, *QueryDef* und *Recordset*. Sie können daher über eines der drei letztgenannten Objekte nur verfügen, wenn Sie das zugehörige *Database*-Objekt – das ist praktisch die zu bearbeitende Datenbank – identifiziert haben. Dieselbe Argumentation gilt zwar im Prinzip auch im Hinblick auf das Objekt *Workspace*, jedoch können Sie im allgemeinen praktisch von diesem Objekt abstrahieren, weil die Voreinstellungen erlauben, für die meisten Zusammenhänge über *Database*-Objekte zu verfügen, ohne sich explizit auf das Objekt *Workspace* zu beziehen.

TableDef. Dieses Objekt stellt die gespeicherte Definition einer Basistabelle oder einer verknüpften Tabelle dar. Es enthält die Definitionsmerkmale der Tabellenfelder und -indizes sowie der Tabelle als Objekt.

QueryDef. Dieses Objekt stellt die gespeicherte Definition einer Abfrage dar.

Recordset. Die beiden vorangehenden Objekte *TableDef* und *QueryDef* geben jeweils lediglich die Definitionsmerkmale von Tabellen bzw. Abfragen wieder, nicht jedoch deren Daten. Wenn Sie auf die Daten einer Tabelle oder Abfrage zugreifen wollen, müssen sie das Objekt *Recordset* verwenden. Dem Objekt *Recordset* kommt unter den DAO-Objekten vermutlich die größte praktische Bedeutung zu. Aus diesem Grunde bilden seine Methoden auch den Schwerpunkt der Ausführungen dieses Kapitels.

31.3 DAO-Objekte Objektvariablen zuweisen

Über die DAO-Objekte können Sie in VBA-Code nur verfügen, wenn Sie diese zuvor in einer Prozedur Objektvariablen zugewiesen haben. Dabei hat die Objektvariable jeweils als Datentyp den Typ des Objekts, das ihr zugewiesen wird. Eine Objektvariable beispielsweise, der eine Datenbank zugewiesen wird, hat den Datentyp *Database*, eine Objektvariable für ein *Recordset*-Objekt den Datentyp *Recordset* usw. Dies ist zu beachten, wenn diese Objektvariablen explizit mit Datentypzuweisung deklariert werden.

Im folgenden zeige ich zunächst für die o.a. wichtigsten DAO-Objekte, wie Sie auf diese Objekte zugreifen können.

31.3.1 Auf Objekttyp *Database* zugreifen

Für das Zuweisen von Datenbanken zu einer Objektvariablen vom Typ *Database* stehen Ihnen die beiden Funktionen *CurrentDB* und *OpenDatabase* zur Verfügung.

Mit *CurrentDB* weisen Sie der Objektvariablen die aktuell geöffnete Datenbank zu, mit *OpenDatabase* eine nicht geöffnete. Anders als auf der allgemeinen Access-Ebene können in einer VBA-Prozedur mehrere Datenbanken gleichzeitig als Objektvariable geöffnet sein. Dies erweitert den Spielraum ganz erheblich.

Das folgende Beispiel deklariert die Objektvariablen DB1 und DB2 mit dem Objekttyp *Database* und weist DB1 die aktuelle Datenbank, DB2 die Datenbank *Auftrag.mdb* mit dem Pfad d:\d\a2000p\bspdsk zu und gibt die Namen der beiden Datenbanken in das Direktfenster aus.

```
Dim DB1 As Database, DB2 As Database
Set DB1 = CurrentDb()
Set DB2 = OpenDatabase("d:\d\a2000p\bspdsk\auftrag.mdb")
Debug.Print DB1.Name, DB2.Name
```

31.3.2 Auf Objekttyp *TableDef* zugreifen

Jedes *TableDef*-Objekt ist Teil des Auflistungsobjekts *TableDefs*. Sie können sich auf eine bestimmte Tabelle beziehen, indem Sie zum *TableDefs*-Objekt die Position in der Auflistung oder den Tabellennamen in Klammern angeben. Die Positionsindizes in DAO beginnen mit dem Wert 0 (nicht mit 1 !). Sofern die erste Tabelle in der Auflistung der Datenbank Nordwind.mdb den Namen *Artikel* besitzt, sind die beiden letzten der folgenden Codezeilen inhaltlich gleichwertig: Sie geben in einer Meldungsbox den Wert *Wahr* oder *Falsch* aus, je nachdem, ob die Tabelle *Artikel* aktualisierbar ist oder nicht.

```
strDBName = "d:\Programme\Microsoft Office” & _
  "\Office\Samples\nordwind.mdb"
Set DB = OpenDatabase(strDBName)
MsgBox DB.TableDefs(0).Updatable
MsgBox DB.TableDefs("Artikel").Updatable
```

Die folgende Prozedur *DBTabnamenAuflisten* identifiziert zunächst die erste Tabelle im Auflistungsobjekt *TableDefs*, indem dazu als Positionswert 0 angegeben wird, und weist diese der Objektvariblen TD1 vom Typ *TableDef* zu. Zum anderen macht sie in ihrem zweiten Teil von der Möglichkeit Gebrauch, das Auflistungsobjekt *TableDefs* in eine *For Each ... Next*-Schleife einzubinden.

```
Sub DBTabnamenAuflisten()
Dim DB As Database, TD1 As TableDef
strDBName = "d:\Programme\Microsoft Office" & _
  "\Office\Samples\nordwind.mdb"
Set DB = OpenDatabase(strDBName)
Set TD1 = DB.TableDefs(0)
```

```
Prüf = MsgBox("Die erste Tabelle hat den Namen: " _
  & Chr(13) & Chr(13) & TD1.Name _
  & Chr(13) & Chr(13) _
  & "Wollen Sie alle Tabellennamen sehen?" _
  & Chr(13), vbYesNo)
If Prüf = vbNo Then
  Exit Sub
End If
TabNamen = ""
For Each tblname In DB.TableDefs
  TabNamen = TabNamen & tblname.Name & Chr(13)
Next
MsgBox TabNamen
End Sub
```

Diese Prozedur zeigt zunächst eine Meldungsbox an, vgl. Bild 31.2.

Bild 31.2: Meldungsbox, die von der Prozedur »DBTabnamenAuflisten« angezeigt wird

Wenn die in Bild 31.2 wiedergegebene Meldungsbox mit *Ja* bestätigt wird, werden in einer weiteren Meldungsbox die Namen aller in der Datenbank Nordwind.mdb enthaltenen Tabellen aufgelistet, die Systemtabellen eingeschlossen.

31.3.3 Auf Objekttyp *QueryDef* zugreifen

Ein *QueryDef*-Objekt ist eine gespeicherte Abfragedefinition in einer Microsoft Jet-Datenbank, wozu auch eine Access-Datenbank zählt. Hierzu zählen daher alle Abfragen, die in einer Access-Datenbank gespeichert sind und auf der Registerkarte *Abfragen* im Datenbankfenster angezeigt werden. Mit den Eigenschaften und Methoden dieses Objekts können Sie eine gespeicherte Abfrage ausführen und ihre Ergebnisse verfügbar machen, Abfrageparameter oder auch die gesamte Abfrageformulierung in Form des entsprechenden SQL-Statements ändern. Das *QueryDefs*-Objekt ist das zugehörige Auflistungsobjekt, auf dessen einzelne Abfragedefinitionen Sie analog zum *TableDefs*-Objekt zugreifen. Mit den folgenden beiden Codezeilen können Sie auf die Abfrage *Rechnungen* der Datenbank *Nordwind.mdb* zugreifen (auf die darin verwendete Methode *OpenRecordset* des Datenbankobjekts wird genauer im folgenden Punkt eingegangen):

```
strDBName = "d:\Programme\Microsoft Office" & _
  "\Office\Samples\nordwind.mdb"
Set DB = OpenDatabase(strDBName)
Set RS1 = DB.OpenRecordset("Rechnungen")
```

Weiter unten in diesem Kapitel wird gezeigt, daß Sie eine Abfrage auch mit einem als Text formulierten SQL-Statement ausführen können. Die Verwendung von gespeicherten Datenbank-Abfragen ist allerdings effizienter als die Verwendung eines SQL-Statement, weil die in kompilierter Form gespeicherte Abfrage von der JetEngine unmittelbar ausgeführt werden kann. SQL-Abfragen erlauben dagegen im allgemeinen eine größere Flexibilität.

31.3.4 Auf Objekttyp *Recordset zugreifen*

Objektvariable vom Typ *Recordset* dienen dazu, auf Datensatzgruppen wie z.B. die Daten einer Tabelle, einer Abfrage oder der einem Formular zugrunde liegenden Datenherkunft (Tabelle/Abfrage) in VBA zugreifen zu können.

Sie deklarieren eine Objektvariable vom Typ *Recordset* in der üblichen Weise, z.B. in der Form

```
Dim RS1 As Recordset
```

Methode *OpenRecordset*

Sie weisen einer derartigen Variablen die Datensatzgruppe einer Tabelle, Abfrage etc. im allgemeinen mit der Methode *OpenRecordset* zu. Sie hat die folgende allgemeine

Syntax 1

Set Variable = Datenbank.**OpenRecordset**(Herkunft[, Typ[, Optionen]] [, Sperren])

Syntax 2

Set Variable = Objekt.**OpenRecordset**([Typ[, Optionen]] [, Sperren])

Der Unterschied zwischen den beiden Syntax-Formen ist der folgende:

- Syntax 1 gilt, wenn das Objekt, aus dem ein Recordset erzeugt werden soll, eine *Datenbank* ist. In diesem Falle muß die *Herkunft* – das ist z.B. der Name einer Tabelle oder Abfrage – angegeben werden.
- Syntax 2 gilt für alle hier möglichen Objekte, die keine Datenbank sind, das sind in diesem Zusammenhang *QueryDef*-, *TableDef*- und *Recordset*-Objekte. Aus dem letzten Typ ergibt sich, daß ein neues Recordset aus einem anderen erzeugt werden kann.

Im einzelnen haben die Argumente die nachfolgend angegebene Bedeutung:

Argument	Bedeutung
Variable	Eine Variable, die als Typ *Recordset* deklariert wurde.
Datenbank	Der Name eines vorhandenen *Database*-Objekts, das zum Erstellen des neuen *Recordset*-Objekts verwendet werden soll.
Objekt	Der Name eines vorhandenen *QueryDef*-, *Recordset*- oder *TableDef*-Objekts, auf dessen Basis das neue *Recordset*-Objekt erstellt werden soll.
Herkunft	Eine Zeichenkette, die die Herkunft der Datensätze für das neue *Recordset*-Objekt angibt. *Herkunft* kann der Name einer Tabelle oder Abfrage sein oder eine SQL-Anweisung, die Datensätze zurückgibt. Zum Gebrauch des Arguments *Herkunft* in Abhängigkeit vom Typ des Arguments *Datenbank* bzw. *Objekt* vgl. die beiden Absätze oberhalb dieser Übersicht.
Typ	Eine Typ-Konstante, die den Datentyp des neuen *Recordset*-Objekts definiert. Folgende Typen sind möglich: *dbOpenTable* *dbOpenDynaset* *dbOpenSnapshot* *dbOpenForwardOnly* *dbOpenDynamic* Zur Bedeutung des Typs eines *Recordset* vgl die Ausführungen unterhalb dieser Übersicht. Dort wird auch angegeben, welcher Typ resultiert, wenn dieses wahlfreie Argument fortgelassen wird.
Optionen	Eine Konstante, die Merkmale des neuen *Recordset*-Objekts festlegt, z. B. Einschränkungen für andere Benutzer beim Bearbeiten und Ansehen des *Recordset*-Objekts. Das Argument ist wahlfrei. Vgl. dazu im einzelnen die Online-Hilfe zur Methode *OpenRecordset*.
Sperren	Eine Konstante, die Sperrungen für das neue *Recordset*-Objekt bestimmt, z. B. nur lesender Zugriff mit dem Schlüsselwort *dbReadOnly*. Das Argument ist wahlfrei. Vgl. dazu im einzelnen die Online-Hilfe zur Methode *OpenRecordset*.

Wenn Sie die wahlfreie Typ-Konstante nicht angeben, setzt Access dafür einen Standardwert ein, der sich nach folgenden Regeln ergibt:

Ausgangsobjekt vom Typ	Standard-Typ-Konstante
TableDef für lokale Tabelle	dbOpenTable
Database	dbOpenDynaset
QueryDef	
TableDef für eingebundene Tabelle	
Recordset	Typ-Konstante, die dem Ausgangs-Recordset entspricht.

Die praktisch wichtigsten drei Recordset-Typen *Tabelle*, *Dynaset* und *Snapshot* haben die nachfolgenden Bedeutungen:

- **Tabelle**. Dieser Typ verweist auf eine lokale Tabelle. Er kann nicht für eine eingebundene Tabelle verwendet werden (hierfür muß der Typ *Dynaset* genommen werden). Einige Operationen – z.B. die Methode *Seek* oder das Indizieren- sind nur für diesen Typ ausführbar.
- **Dynaset**. Dieser Typ verweist auf eine lokale oder eingebundene Tabelle oder auf das Datenergebnis einer Abfrage. Mit dem Typ *Dynaset* können Sie Datensätze aus mehreren Tabellen – darunter auch eingebundenen – extrahieren und aktualisieren.
- **Snapshot**. Dieser Typ ist dem Typ *Dynaset* gleich mit den folgenden wichtigen Ausnahmen: Ein *Dynaset* enthält eine Gruppe von indirekten Referenzen auf die eigentlichen Datensätze in den Tabellen, während das *Snapshot* eine statische Kopie der Datensätze ist. Das *Snapshot* kann daher nicht aktualisiert werden. Gegenüber dem Typ *Dynaset* weist es beim passiven Zugriff auf Datensätze, auf den es ja beschränkt ist, einen Geschwindigkeitsvorteil auf.

Beispiele zur Methode *OpenRecordset*

Ausgangsobjekt lokale Tabelle. Das folgende Beispiel definiert mit der ersten Zeile die Variable *DB* als Objekttyp *Database* und *RS1* als Objekttyp *Recordset*. Die zweite Zeile weist *DB* die aktuelle Datenbank zu, die dritte weist der Variablen *RS1* mit der Methode *OpenRecordset*, die hier für das Datenbankobjekt *DB* angewandt wird, die Tabelle *Kunden* zu. Die beiden letzten Zeilen haben Testcharakter: Sie sorgen dafür, daß der Wert des Feldes *Firma* des ersten Datensatzes in das Direktfenster ausgegeben wird.

```
Dim DB As Database, RS1 As Recordset
Set DB = CurrentDB()
Set RS1 = DB.OpenRecordset("Kunden")
RS1.MoveFirst
Debug.Print RS1("Firma")
```

Ausgangsobjekt Abfrage. Wenn es sich bei der Abfrage, die als Ausgangsobjekt eines neuen *Recordset*-Objekts dienen soll, um eine in der Datenbank unter einem Namen gespeicherte Abfrage handelt, verfahren Sie syntaktisch wie im vorangehenden Beispiel einer lokalen Tabelle. Wenn *KundenAdressen* der Name einer bestehenden Abfrage in der aktuellen Datenbank ist, können Sie der Objektvariablen *RS1* das Datenergebnis dieser Abfrage folgendermaßen zuweisen und den Inhalt des Feldes *KundenCode* im ersten Datensatz mit dem folgenden Code in das Direktfenster ausgeben lassen:

```
Dim DB As Database, RS1 As Recordset
Set DB = CurrentDB()
Set RS1 = DB.OpenRecordset("KundenAdressen")
RS1.MoveFirst
Debug.Print RS1("KundenCode")
```

Die Methode *OpenRecordset* akzeptiert aber auch eine als Zeichenkette angegebene SQL-SELECT-Anweisung. Die folgenden Codezeilen weisen der Objektvariablen *RS1* vom Typ *Recordset* das Abfrageergebnis der als SQL-Anweisung formulierten Abfrage mit der Tabellenbasis *Kunden* zu, geben dem ersten Datensatz den Fokus und schreiben den Inhalt des Feldes *Firma* in das Direktfenster.

```
Dim DB As Database, RS1 As Recordset, RS2 As Recordset
Set DB = CurrentDB()
Set RS1 = DB.OpenRecordset("SELECT * FROM Kunden;")
RS1.MoveFirst
Debug.Print RS1("Firma")
```

Ausgangsobjekt Recordset. Wenn das Objekt, auf das die Methode *OpenRecordset* angewandt wird, kein Datenbankobjekt ist, sondern z.B. eine andere Objektvariable vom Typ *Recordset*, ist die Herkunft durch die Objektangabe eindeutig. In diesem Falle bleibt das Klammerpaar am Ende leer, wie in der letzten Zeile des folgenden Beispiels, in dem eine zweite Datensatzgruppe von der mit der Methode *Sort* sortierten ersten erstellt und der Objektvariablen *RS2* zugewiesen wird.

```
Dim DB As Database, RS1 As Recordset, RS2 As Recordset
Set DB = CurrentDB()
Set RS1 = DB.OpenRecordset("Kundenliste")
RS1.Sort = "[Firma]"
Set RS2 = RS1.OpenRecordset()
```

Um ein *Recordset*-Objekt zu kopieren, können Sie auch die *Recordset*-Methode *Clone* verwenden. Für diesen Fall würde die letzte Zeile des vorangehenden Beispiels lauten:

```
Set RS2 = RS1.Clone
```

Ausgangsobjekt Formular. Oft ist es wünschenswert, die Datenbasis eines Formulars als Objektvariable vom Typ *Recordset* zu handhaben. Dann lassen sich beispielsweise die leistungsfähigen *Find*- und *Move*-Methoden anwenden, so daß Sie im Formular mit komplexeren Kriterien nach bestimmten Datensätzen suchen und zwischen Datensätzen blättern können. (Um diese beiden Möglichkeiten zu realisieren, müssen Sie allerdings noch weiteren Aufwand treiben, vgl. dazu weiter unten Punkt 31.4, *Datenmanipulation*.) Um eine Datensatzgruppe zu erstellen, die auf einem Formular basiert, verwenden Sie nicht die Methode *OpenRecordset*, sondern die Formular-Eigenschaft *Recordset*. (In früheren Versionen stand nur die Eigenschaft *RecordsetClone*, die etwas weniger leistungsfähig ist als Recor*d*set, zur Verfügung. Sie wird aber aus Kompatibilitätsgründen weiterhin unterstützt.)

Die folgenden Beispielzeilen weisen der Objektvariablen *RS1* vom Typ *Recordset* das Dynaset zu, welches dem (geöffneten) Formular *Adressen* als Datenbasis zugrunde liegt:

```
Dim RS1 As Recordset
Set RS1 = Forms!Adressen.Recordset
```

Statt auf einen bestimmten Formularnamen können Sie sich auch auf das aktuelle Formular beziehen, ohne dessen Namen anzugeben. In diesem Falle müssen Sie natürlich durch geeignete Anweisungen vorher dafür sorgen, daß das richtige Formular nicht nur geöffnet ist, sondern auch den Fokus hat. Die folgenden Beispielzeilen weisen der Objektvariablen *RS1* vom Typ *Recordset* das Dynaset zu, welches dem Formular mit dem Fokus als Datenbasis zugrunde liegt:

```
Dim RS1 As Recordset
Set RS1 = Screen.ActiveForm.Recordset
```

Ein praktisch relevantes Anwendungsbeispiel für diese Duplizierung der Datenbasis eines Formulars enthält die Datenbank *Adressen.mdb* von der Begleit-CD-ROM in der Prozedur *AdressenSuchen*: Diese Prozedur sucht einen Datensatz des Formulars auf Basis von bis zu drei Suchkriterien und gibt dem ersten gefundenen Datensatz im Formular den Fokus.

31.4 Datenmanipulation mit DAO-Recordset-Objekten

Die Online-Hilfe zu Access enthält unter dem Thema *Microsoft Data Access Objects 3.60 (DAO)* eine vollständige Darstellung der Objekte, Methoden, Eigenschaften und weiterer Sprachelemente, die mit DAO überhaupt für die Manipulation von Daten zur Verfügung stehen, und in der Regel finden Sie dort auch Anwendungsbeispiele für die einzelnen Sprachelemente. Die dortige alphabetische Anordnung erlaubt es aber kaum, nach Sachthemen begründete Strukturen zu entdecken. Dieser Abschnitt soll Ihnen helfen, die

richtigen Programmierwerkzeuge für bestimmte, immer wiederkehrende Aufgaben im Zusammenhang mit der Bearbeitung von Datensätzen im weitesten Sinne zu finden. Die Ausführungen sind aus räumlichen Gründen und aus Gründen der Übersicht jeweils sehr knapp gehalten. Lesen Sie zu den einzelnen Sprachelementen ausführlicher unter den entsprechenden Stichwörtern in der Referenz der Online-Hilfe zu den Objekten, Methoden und Eigenschaften von DAO nach.

31.4.1 In Datensätzen blättern

Verwenden Sie die Methoden **Movefirst**, **MoveLast**, **MoveNext**, **MovePrevious**.

Diese Methoden wechseln den Fokus auf den ersten, letzten, nächsten bzw. vorangehenden Datensatz. Sie haben die folgende Syntax:

Syntax

Recordset-Objekt**.MoveFirst**
Recordset-Objekt**.MoveLast**
Recordset-Objekt**.MoveNext**
Recordset-Objekt**.MovePrevious**

Beispiel

Das folgende Beispiel erzeugt ein *Recordset*-Objekt mit allen Datensätzen der Tabelle *Kunden*, die im Feld *Firma* nicht den Wert Null haben, und gibt den Feldinhalt von *Firma* für alle passenden Datensätze in das Direktfenster aus.

```
Dim DB As Database, RS As Recordset
Set DB = CurrentDb()
strSQL = "SELECT * FROM Kunden WHERE Firma Is Not Null;"
Set RS = DB.OpenRecordset(strSQL)
Do Until RS.EOF
    Debug.Print RS!Firma
    RS.MoveNext
Loop
```

Beachten Sie die Verwendung der *Recordset*-Eigenschaft *EOF* (= End of File) um zu prüfen, ob der Datensatz mit dem Fokus der letzte in der Objektvariablen ist. Zum Test auf den ersten Datensatz verwenden Sie entsprechend die *Recordset*-Eigenschaft *BOF* (= Begin of File).

31.4.2 Datensätze neu sortieren

Für Recordset-Variablen des Typs *Tabelle* verwenden Sie deren Eigenschaft *Index*, um eine neue Sortierordnung festzulegen. Für Recordset-Variablen der Typen *Dynaset* und *Snapshot* setzen Sie deren Eigenschaft *Sort* auf einen

geeigneten Wert. Wenn Sie die Datensätze eines *Recordset*-Objekts, das eine Tabelle als Datenquelle hat, sortieren wollen, ohne die Eigenschaft *Index* der Tabelle neu einzustellen, öffnen Sie das *Recordset*-Objekt als Typ *Dynaset*, indem Sie zur Methode *OpenRecordset* das Argument *dbOpenDynaset* anführen. Dann lassen sich auch diese Datensätze sortieren, wie im folgenden beispielhaft gezeigt.

Beispiele

Im folgenden Beispiel ist das Dynaset DS2 nach dem Auftragswert sortiert, nicht jedoch das Dynaset DS1.

```
Dim DB As Database, DS1 As Recordset, DS2 As Recordset
Set DB = CurrentDB()
strSQL = "SELECT * FROM Artikel;"
Set DS1 = DB.OpenRecordset(strSQL)
DS1.Sort = "Gesamtpreis ASC"
Set DS2 = DS1.OpenRecordset()
```

Das Ergebnis des vorangehenden Beispiels ist übrigens effizienter zu erzielen, wenn Sie die Sortierung gleich in der SQL-Anweisung berücksichtigen; das folgende Beispiel ist dem vorangehenden im Ergebnis gleichwertig.

```
Dim DB As Database, DS2 As Dynaset
Set DB = CurrentDB()
strSQL = "SELECT * FROM Artikel ORDER BY Gesamtpreis;"
Set DS2 = DB.OpenRecordset(strSQL)
```

31.4.3 Filter ändern

Für Objektvariablen der Typen *Dynaset* und *Snapshot* geben Sie für deren Eigenschaft *Filter* einen geeigneten Ausdruck an.

Beispiel

Das folgende Beispiel weist zunächst der Objektvariablen *DS1* vom Typ *Recordset* eine Datensatzgruppe vom Typ *Dynaset* auf Basis der Tabelle *Kunden* zu. Die explizite Typ-Angabe zur Methode *OpenRecordset* ist notwendig, weil einerseits das Ausgangsobjekt vom Typ *Table* standardmäßig ebenfalls einen Datensatzgruppentyp *Table* erzeugt, andererseits die Eigenschaft *Filter* nur für Dynasets und Snapshots verfügbar ist. Dem Dynaset *DS1* wird die *Filter*-Eigenschaft

```
"Firma Like 'A*'"
```

zugewiesen. Danach ist dieses Dynaset aber weiterhin insoweit ungefiltert, als das Filter erst beim Erzeugen eines Dynaset wirksam werden kann. Daher wird das zweite Dynaset *DS2* erzeugt, das nur Datensätze enthält,

deren *Firma* mit *A* anfängt. Möglicherweise wäre es effizienter, gleich beim Erzeugen des ersten Dynaset eine SQL-Anweisung mit einer entsprechenden WHERE-Klausel als Argument beizufügen.

```
Dim DB As Database, DS1 As Recordset, DS2 As Recordset
Set DB = CurrentDB()
Set DS1 = DB.OpenRecordset("Kunden", dbOpenDynaset)
DS1.Filter = "Firma Like 'A*'"
Set DS2 = DS1.OpenRecordset()
Do Until DS2.EOF
    Debug.Print DS2!Firma
    DS2.MoveNext
Loop
```

31.4.4 Datensätze suchen

Für *Recordset*-Variablen der Typen *Table* einerseits sowie *Dynaset* und *Snapshot* andererseits müssen Sie unterschiedliche Methoden anwenden: Auf Objektvariablen des Typs *Table* können Sie nur die Methode *Seek* anwenden. Umgekehrt lassen sich die Methoden *FindFirst*, *FindLast*, *FindNext*, *FindPrevious* nur für Dynasets und Snapshots verwenden. Die *Find*-Methoden erlauben differenziertere Suchkriterien als die *Seek*-Methode.

Die *Find*-Methoden einerseits und die *Seek*-Methode andererseits besitzen im gegenseitigen Vergleich jeweils Vor- und Nachteile. Der Vorteil der *Find*-Methoden liegt darin, daß diese sehr differenzierte Kriterienausdrücke erlauben, wogegen mit der *Seek*-Methode nur nach Werten in indizierten Feldern gesucht werden kann. Dafür ist die *Seek*-Methode ungleich schneller als die *Find*-Methoden, denn letztere gehen die Datensatzgruppe sequentiell durch, um einen bestimmten Datensatz zu suchen, während bei der *Seek*-Methode nach einem intelligenteren Algorithmus unter Verwendung des Index gezielt gesucht wird.

Find-Methoden

Die Methoden suchen nach dem ersten (letzten, nächsten, vorherigen) Datensatz, der den *Kriterien* entspricht, und machen ihn, sofern ein solcher gefunden wird, zum aktuellen Datensatz.

Syntax

Recordset-Objekt.**FindFirst** Kriterien
Recordset-Objekt.**FindLast** Kriterien
Recordset-Objekt.**FindNext** Kriterien
Recordset-Objekt.**FindPrevious** Kriterien

Die Argumente haben die nachfolgende Bedeutung:

Argument	Beschreibung
Recordset-Objekt	Name eines Recordset vom Typ *Dynaset* oder *Snapshot.*
Kriterien	Suchbedingung als Zeichenkette (wie eine SQL-WHERE-Klausel ohne das Wort WHERE).

Im Unterschied zu den *Move*-Methoden, die sich auf alle Datensätze beziehen, berücksichtigen die *Find*-Methoden nur solche, die den Kriterien entsprechen.

Vergewissern Sie sich prinzipiell, ob die Suche mit einer der *Find*-Methoden erfolgreich war. Trifft dies nicht zu, so hat die Eigenschaft *NoMatch* des betreffenden Objekts den Wert *True* (*Wahr*, *-1*), im anderen Falle ist sie *False* (*Falsch*, *0*).

Beispiel

Das folgende Beispiel sucht den ersten Datensatz, der dem eingegebenen Firmennamen entspricht, und weist dem Feld *Auftrag-abgeschlossen* den Wert 0 zu, wenn der Datensatz gefunden wurde. Im anderen Falle wird eine entsprechende Meldung ausgegeben.

```
Dim DB As Database, DS As Recordset
Dim FName As String, SuchKrit As String
Set DB = CurrentDB()
Set DS = DB.OpenRecordset("Hauptaufträge", dbOpenDynaset)
FName = InputBox("Firmenname:") & "*"
SuchKrit = "Firma Like '" & FName & "'"
DS.FindFirst SuchKrit
If DS.NoMatch Then
    MsgBox "Firma nicht gefunden!"
Else
    DS.Edit
    DS![Auftrag-abgeschlossen] = 0
    DS.Update
End If
```

31.4.5 Lesezeichen verwenden

Die Eigenschaft *Bookmark* erlaubt es, Lesezeichen zu lesen und zu setzen. Auf diese Weise können Sie u.a. Formulare mittels ihrer Recordsets synchronisieren oder nach einer Suche mit einer der *Find*-Methoden im Recordset eines Formulars den gefundenen Datensatz im Formular anzeigen lassen. Beispiele dafür finden Sie u.a. in der Datenbank *Adressen.mdb* von der

Begleit-CD-ROM, vgl. dort z.B. die Prozeduren *NamenSuchen* und *Adresse-Suchen*. Die erstere dieser beiden Prozeduren wird im folgenden Beispiel wiedergegeben. Entscheidend für die Fokussierung im Formular auf den Datensatz, der mit der Methode *FindFirst* gefunden wurde, ist die drittletzte Codezeile, die die Eigenschaft *Bookmark* anwendet.

```
Function NamenSuchen()
'Diese Funktion sucht nach einem Datensatz auf
'Basis eines mit der Eingabebox eingegebenen
'Namens. Die Funktion wird in der Datenbank nicht mehr
'verwendet. Sie verblieb aus demonstrativen
'Gründen in der Datenbank
Dim DForm As Recordset, Such, SuchKrit
DoCmd.OpenForm ("Adressen (Makro-Steuerung)")
Set DForm = Screen.ActiveForm.RecordsetClone
Such = InputBox("Name:")
SuchKrit = "Name Like """ & Such & "*"""
DForm.FindFirst SuchKrit
If DForm.NoMatch Then
    MsgBox "Der Name " & Such & " ist nicht vorhanden."
Else
    Screen.ActiveForm.Bookmark = DForm.Bookmark
End If
End Function
```

31.4.6 Inhalte von Datensätzen ändern

Verwenden Sie die Methoden *Edit* und *Update* im Zusammenwirken, um Datensätze zu aktualisieren. Die Methode *Edit* kopiert den aktuellen Datensatz des angegebenen Objekts zur Bearbeitung in den Access-internen Zwischenspeicher (nicht zu verwechseln mit der Zwischenablage von Windows). Wenn Sie den aktuellen Datensatz mit der Methode *Edit* in den Zwischenspeicher kopiert und ihn dort nach Ihren Wünschen bearbeitet haben, ist er noch nicht automatisch gespeichert. Um ihn zu speichern, müssen Sie noch ausdrücklich die Methode *Update* anwenden, bevor Sie den Fokus auf einen anderen Datensatz richten. Solange Sie nicht ausdrücklich zu einem anderen Datensatz wechseln, bleibt der mit *Edit* bearbeitete auch der aktuelle. Beachten Sie auch, daß die Bearbeitungsmöglichkeit des Datensatzes auch von der Einstellung der Eigenschaft *LockEdits* für das Objekt abhängt.

Syntax

Recordset-Objekt.Edit
Recordset-Objekt.Update

Argument	Beschreibung
Recordset-Objekt	Name eines geöffneten Recordset vom Typ *Tabelle* oder *Dynaset*.

Beispiel

Das folgende Beispiel weist dem Feld *Auftrag-abgeschlossen* für jeden Datensatz aus der Tabelle *Hauptaufträge*, dessen Auftragsnummer zwischen 19920001 und 19929999 liegt, den Wert *True* zu (das ist in der Tabelle der Wert *Ja* für ein Feld vom Felddatentyp *Ja/Nein*).

```
Dim DB As Database, DS As Recordset, SuchKrit As String
Set DB = CurrentDB()
'Die Typ-Angabe dbOpenDynaset ist notwendig, weil
'andernfalls der Typ Tabelle resultieren würde,
'für den die Find-Methoden nicht verfügbar sind.
Set DS = DB.OpenRecordset("Hauptaufträge", dbOpenDynaset)
SuchKrit = "AuftragsNr >= 19920001 And AuftragsNr "
SuchKrit = SuchKrit & "<= 19929999"
DS.FindFirst SuchKrit
Do While Not DS.NoMatch
    DS.Edit
    DS![Auftrag-abgeschlossen] = True
    DS.Update
    DS.FindNext SuchKrit
Loop
```

31.4.7 Neue Datensätze hinzufügen

Verwenden Sie die Methoden *AddNew* und *Update*, um einen neuen Datensatz in einem Recordset vom Typ *Tabelle* oder *Dynaset* anzufügen.

Syntax

Recordset-Objekt.**AddNew**

Argument	Beschreibung
Recordset-Objekt	Name eines Recordset vom Typ *Table* oder *Dynaset*.

Es wird am Ende des Recordset ein neuer Datensatz angefügt, dessen Felder auf den Wert Null gesetzt werden. Die ggf. als Feldeigenschaften definierten Standardwerte werden nicht gesetzt. Nachdem der neue Datensatz bearbeitet wurde, werden die Änderungen mit *Update* gespeichert (und nicht mit *Edit* bearbeitet). Falls Sie zu einem anderen Datensatz wechseln, ohne zuvor *Update* ausgeführt zu haben, wird kein Datensatz an die Tabelle angefügt, weil sämtliche Änderungen vor *Update* nur im Zwischenspeicher aufbewahrt werden.

Beispiel

Die folgende Prozedur *DatensatzNeu* erzeugt einen neuen Datensatz für die Tabelle *Hauptaufträge* der aktuellen Datenbank. Sie ermittelt eine neue Auftragsnummer (achtziffrig, erste vier Ziffern Jahreszahl, letzte vier Ziffern laufende Nummer, jeweils um 1 höher als die größte für das aktuelle Jahr vorhandene) und weist diese dem Feld *AuftragsNr* zu. Der Firmenname wird erfragt und ebenfalls dem entsprechenden Feld zugewiesen. Nach der Speicherung des neuen Satzes mit den eingegebenen Werten mit `Tabelle1.Update` wird der neue Datensatz zum aktuellen gemacht. Mittels einer Abfrage kann er (aus Testgründen) gleich wieder gelöscht werden.

```
Sub DatensatzNeu ()
Dim Db1 As Database, Tabelle1 As Recordset
Dim NeueAuftragsNr As Double, Firma, Löschen
Set Db1 = CurrentDB()
'Tabelle öffnen
Set Tabelle1 = Db1.OpenRecordset("Hauptaufträge")
'Index festlegen, sonst Suche mit .Seek nicht möglich
'Der Primärschlüssel ist auf das Feld AuftragsNr festgelegt.
Tabelle1.index = "PrimaryKey"
'Leeren Datensatz erzeugen
Tabelle1.AddNew
'Neue Auftragsnummer ermitteln
NeueAuftragsNr = DMax("[AuftragsNr]", "Hauptaufträge", _
  "Int([Auftragsnr]/10000)=Year(Date())") + 1
'Neue Auftragsnummer an Tabellenfeld
Tabelle1("AuftragsNr") = NeueAuftragsNr
'Firmennamen angeben
Firma = InputBox$("Geben Sie den Firmennamen an:", "Firmenname")
'Firmennamen an Tabellenfeld
Tabelle1("Firma") = Firma
'Änderungen am neuen Datensatz speichern
Tabelle1.Update
'Neuen Datensatz zum aktuellen machen
Tabelle1.Seek "=", NeueAuftragsNr
'Datensatz löschen?
Löschen = MsgBox("Soll der Datensatz gelöscht werden?", 36, _
  Str(NeueAuftragsNr))
If Löschen = 6 Then
  'Akt. Datensatz löschen bei Ja-Button
  Tabelle1.Delete
End If
'Tabelle schließen
Tabelle1.Close
End Sub
```

31.4.8 Datensätze löschen

Verwenden Sie die Methode *Delete*, um den aktuellen Datensatz im Recordset zu löschen.

Syntax

Recordset-Objekt.**Delete**

Argument	Beschreibung
Recordset-Objekt	Name eines Recordset vom Typ *Tabelle* oder *Dynaset*.

In einer Tabelle löscht *Delete* den aktuellen Datensatz, in einem Dynaset gibt sie ihm den Wert Null und löscht ihn in den zugrundeliegenden Tabellen. Obwohl Sie einen so gelöschten Datensatz nicht mehr bearbeiten können, bleibt er der aktuelle, bis Sie explizit einen anderen dazu machen. Der bis zum Löschen vorherige bzw. nächste Datensatz bleibt zunächst der vorherige bzw. nächste, so daß Sie z.B. mit *MoveNext* zum nächsten wechseln können. Sobald der Fokus auf einen anderen Datensatz gerichtet wird, zählt der gelöschte in keiner Hinsicht mehr.

Wenn Sie das Löschen von Datensätzen in eine Transaktion einbinden, können Sie vor dem endgültigen Löschen, ggf. auch mehrerer Datensätze, nach einer Bestätigung fragen und das Löschen ggf. rückgängig machen. Vgl. die Methoden *BeginTrans*, *CommitTrans* und *Rollback*, die im folgenden Punkt 31.4.9, *Transaktionen*, behandelt werden.

Beispiel. Ein Beispiel zur Methode *Delete* finden Sie im ausführlichen Beispiel zum vorangehenden Punkt 31.4.7, *Neue Datensätze hinzufügen*.

31.4.9 Transaktionen

Um ganze Gruppen von Operationen, die eine sachliche Einheit bilden, ggf. rückgängig machen zu können, stehen die Methoden *BeginTrans*, *Commitxrans* und *Rollback* zur Verfügung.

Syntax

Arbeitsbereich.**BeginTrans**
Arbeitsbereich.**CommitTrans**
Arbeitsbereich.**Rollback**

Argument	Beschreibung
Arbeitsbereich	Das Argument *Arbeitsbereich* ist eine Objektvariable des Typs *Workspace*.

Eine Transaktion besteht aus einer Reihe von Operationen, die an einer Datensatzgruppe (Recordset vom Typ *Table* oder *Dynaset*) vorgenommen werden. Der Beginn einer Transaktion wird durch die Methode *BeginTrans*

markiert, das Ende durch eine der Methoden *Rollback* oder *CommitTrans*. Wird die Transaktion mit *Rollback* abgeschlossen, so werden die zur Transaktion gehörenden Operationen nicht wirksam. Wird die Transaktion mit *CommitTrans* beendet, so werden ihre Operationen wirksam. Transaktionen können bis zu fünffach ineinander geschachtelt werden.

Transaktionen dienen nicht nur dazu, Gruppen von Operationen an Datengruppen ggf. unwirksam machen zu können. Viele Operationen (vor allem schreibende Zugriffe) werden deutlich effizienter ausgeführt, wenn sie Teil einer Transaktion sind, vgl. auch den Hinweis weiter unten.

Einige fremde Datenbanken unterstützen möglicherweise keine Transaktionen. Daher sind Transaktionen für entsprechende Recordsets unsichere Vorgänge. Prüfen Sie im Zweifel mit der Eigenschaft *Transactions* des Recordset, ob alle seine Felder zurückgesetzt werden können (True) oder ob dies nicht zutrifft (False).

Beispiel

Das folgende Beispiel löscht die ersten 980 Datensätze der Tabelle *Tabelle1* in einer Transaktion. Wenn die Frage »Datensätze löschen?« mit »Ja« beantwortet wird, schließt die Transaktion mit *CommitTrans*, so daß die Datensätze tatsächlich gelöscht werden. Im anderen Falle schließt die Transaktion mit *Rollback*, und die Datensätze werden nicht gelöscht.

```
Const Ja = 6, Ja_Nein = 4, Fragezeichen = 32
Dim AB1 As WorkSpace, DB As Database
Dim T1 As Recordset
Set AB1 = DBEngine.Workspaces(0)' Standardarbeitsbereich abrufen.
Set DB = CurrentDB()
Set T1 = DB.OpenRecordset("Tabelle1")
AB1.BeginTrans
For i = 1 To 980
T1.Delete
T1.MoveNext
Next
If MsgBox("Die ersten " & i - 1 & "Datensätze löschen?", _
  Fragezeichen + Ja_Nein) = Ja Then
  AB1.CommitTrans
Else
    AB1.Rollback
End If
T1.Close
```

Häufig kann die Performance verbessert werden, indem Operationen, die auf den Datenträger zugreifen, in Transaktionsblöcke aufgeteilt werden. Dadurch werden die Operationen im Arbeitsspeicher gepuffert, und die Anzahl der Zugriffe auf den Datenträger kann deutlich herabgesetzt werden.

In einem Jet-Arbeitsbereich werden Transaktionen in einer Datei protokolliert, die in dem Ordner gespeichert wird, der durch die Umgebungsvariable TEMP auf der Workstation angegeben ist. Wenn für die Transaktions-Protokolldatei nicht mehr genügend Speicher auf dem durch TEMP bezeichneten Laufwerk vorhanden ist, löst das Datenbankmodul einen Laufzeitfehler aus. Wenn Sie zu diesem Zeitpunkt *CommitTrans* verwenden, wird eine unbestimmte Anzahl an Operationen abgeschlossen, aber die verbleibenden, noch nicht abgeschlossenen Operationen sind verloren, und die Operation muß erneut gestartet werden. Durch das Aufrufen einer *Rollback*-Methode werden das Transaktionsprotokoll freigegeben und alle Operationen in der Transaktion zurückgesetzt.

31.5 Datenzugriff mit ADO

31.5.1 Minimalangaben für den Datenzugriff

Für den Datenzugriff mit ADO müssen Sie prinzipiell wenigstens die beiden folgenden Schritte gehen:

- *Verbindung zur Datenquelle herstellen*. Bevor in irgendeiner Weise (definierend, lesend, einfügend, aktualisierend) auf Daten einer Datenquelle zugegriffen werden kann, muß erst einmal eine Verbindung zu dieser aufgebaut werden. Dies geschieht mit Hilfe des Objekts *Connection*.
- *Recordset erstellen*. Auf Basis einer bestehenden aktiven *Connection* wird ein Recordset geöffnet, und diesem wird eine Datenmenge zugewiesen, beispielsweise mittels einer SQL-Anweisung.

Einführungsbeispiel

Das folgende Code-Beispiel greift auf die Datenbank *Nordwind.mdb* zu und gibt mit Hilfe eines Recordsets Datensätze aus der Tabelle *Kunden* aus.

```
Sub ADO_FremdDB()
'Variable für Connection-Objekt deklarieren
Dim cnnNordwind As New ADODB.Connection
'Variable für Recordset-Objekt deklarieren
Dim rstTab As New ADODB.Recordset
'OLE DB-Provider angeben
cnnNordwind.Provider = "Microsoft.Jet.OLEDB.4.0"
'Die Datenbank Nordwind darf für die folgende Art des
'Connection-Aufbaus nicht die aktuelle sein.
cnnNordwind.ConnectionString = "Data Source=D:\Programme" & _
  "\Microsoft Office\Office\Samples" & _
  "\Nordwind.mdb;User ID=Admin"
cnnNordwind.Open
```

```
'Recordset auf die aktive Connection einstellen
Set rstTab.ActiveConnection = cnnNordwind
'Dynamischen CursorType einstellen
rstTab.CursorType = adOpenKeyset
'Datenquelle für Recordset einstellen
rstTab.Source = "Select * From Kunden"
'Recordset öffnen
rstTab.Open

'Ersten Datensatz ins Direktfenster ausgeben
For Each Feld In rstTab.Fields
  Debug.Print Feld.Name, rstTab(Feld.Name)
Next

'Cursor zum letzten Datensatz bewegen
rstTab.MoveLast
'Leerzeile im Direktfenster erzeugen
Debug.Print

'Letzten Datensatz ins Direktfenster ausgeben
For Each Feld In rstTab.Fields
  Debug.Print Feld.Name, rstTab(Feld.Name)
Next

End Sub
```

Erklärung des Beispiels

- Für die beiden entscheidenden Objekte *Connection* und *Recordset* werden zunächst mit *Dim*-Anweisungen die Objektvariablen *cnnNordwind* und *rstTab* deklariert.
- Dem *Connection*-Objekt muß der OLE DB-Provider mitgeteilt werden. Dies geschieht hier dadurch, daß die *Connection*-Eigenschaft *Provider* auf den Wert *Microsoft.Jet.OLEDB.4.0*, der für eine Access-Jet-Datenbank gilt, eingestellt wird.
- Der *Connection*-Eigenschaft *ConnectionString* wird die Verbindungszeichenfolge übergeben, die hier den Pfad zur Datenbank *Nordwind.mdb* und die User ID *Admin* umfaßt.
- Die Verbindung wird mit der *Connection*-Methode *Open* aktiviert.
- Nachdem die Verbindung zur Datenquelle mit der *Connection*-Methode *Open* aktiviert wurde, wird die Recordset-Eigenschaft *ActiveConnection* auf die aktive Verbindung eingestellt, indem ihr der Name der *Connection*-Objektvariablen zugewiesen wird.

- Die Einstellung des Recordset auf einen dynamischen *CursorType* ist notwendig, damit später die Recordset-Methode *MoveLast* fehlerfrei angewandt werden kann.
- Die *Recordset*-Eigenschaft *Source* wird auf die SQL-Select-Anweisung eingestellt, mit deren Datenmenge das Recordset dann beim Öffnen gefüllt wird.
- Das Recordset wird schließlich mit der Recordset-Methode *Open* geöffnet.
- Die weiteren Anweisungen geben Daten ins Direktfenster aus. Sie unterscheiden sich nicht von den bisher in anderen Beispielen angewandten Anweisungen und sind nicht ADO-spezifisch.

31.5.2 Das Connection-Objekt

Das *Connection*-Objekt ist das wichtigste Objekt auf der obersten Ebene des ADO-Objektmodells: Sie können erst dann über Recordset-Objekte verfügen, nachdem Sie eine Verbindung erfolgreich geöffnet haben.

Eigenschaften

Unter den elf Eigenschaften des *Connection*-Objekts, die hier keinesfalls alle besprochen werden sollen, sind zwei besonders hervorzuheben:

ConnectionString. Dieser Eigenschaft wird eine Zeichenkette übergeben, die Verbindungsinformationen für die zu öffnende Datenquelle enthält. Der *ConnectionString* kann verschiedene Informationen enthalten, die nicht nur von der Art der Datenquelle abhängig sind, sondern auch davon, ob die Eigenschaften *DefaultDatabase* und *Provider* des *Connection*-Objekts mit entsprechenden Informationen versorgt wurde. Die Eigenschaft *ConnectionString* muß nicht versorgt werden, statt dessen kann der *ConnectionString* auch als Argument zur *Open*-Methode des *Connection*-Objekts angeführt werden. In den Beispielen dieses Kapitels verfahre ich jedoch stets so wie im letzten Code-Beispiel: Der *ConnectionString* wird der gleichlautenden Eigenschaft übergeben.

Außer den beiden Argumenten *Data Source* und *User ID*, die im obigen Beispiel im *ConnectionString* verwendet wurden, können im Ganzen sieben verschiedene Argumente angeführt werden. Die Access 2000-Hilfe gibt allerdings nur vier an. Genauer können Sie sich jedoch darüber wie über das gesamte ADO-Modell an der folgenden Stelle informieren:

```
www.microsoft.com/data/ado/adords15
```

Dort finden sie unter anderem auch ein Tutorium zu ADO.

Provider. Diese *Connection*-Eigenschaft wird auf eine Zeichenfolge eingestellt, die den OLE DB-Provider spezifiziert. Im obigen Code-Beispiel lautete diese Zeichenfolge *Microsoft.Jet.OLEDB.4.0*, womit der OLE DB-Provider

für eine Access-Jet-Datenbank identifiziert wird. Sie finden die anderen derzeit verfügbaren Provider und die diese identifizierenden Schlüssel-Zeichenfolgen in der Access 2000-Hilfe zu ADO, vgl. das Thema Verwenden von Providern mit ADO. Mit der Provider-Angabe verhält es sich wie mit dem *ConnectionString*: Diese Angabe muß nicht mittels der Eigenschaft *Provider* erfolgen, sie kann vielmehr auch mit dem Wert zum Argument *Provider=* im *ConnectionString* gemacht werden. Beispielsweise hätte der *ConnectionString* im obigen Code-Beispiel auch lauten können:

```
"\Microsoft Office\Office\Samples" & _
"\Nordwind.mdb;User ID=Admin;Provider=" & _
"Microsoft.Jet.OLEDB.4.0"
```

In diesem Falle wäre die Connection-Eigenschaft *Provider* nicht auf den Provider eingestellt worden.

Die folgende Tabelle gibt alle sieben für den *ConnectionString* verfügbaren Argumente und ihre Beschreibungen wieder.

Argument	Beschreibung
Provider	Gibt den Namen des Providers für die Verbindung an.
Data Source	Gibt den Namen einer Datenquelle für die Verbindung an. Dies kann beispielsweise ein SQL-Server sein, der als ODBC-Datenquelle registriert ist, aber auch der Pfad auf eine Access-Datenbank.
User ID	Gibt den Namen des Benutzers an, der beim Öffnen der Verbindung verwendet werden soll.
Password	Gibt das Kennwort an, das für die Verbindung verwendet werden soll.
File Name	Gibt den Namen einer Provider-spezifischen Datei an (beispielsweise ein dauerhaftes Datenquellenobjekt), die vordefinierte Verbindungsinformationen enthält.
Remote Provider	Gibt den Namen eines Providers an, der verwendet werden soll, wenn eine Client-seitige Verbindung geöffnet wird. (Nur für Remote Data Service).
Remote Server	Gibt den Pfadnamen eines Servers an, der verwendet werden soll, wenn eine Client-seitige Verbindung geöffnet wird. (Nur für Remote Data Service).

Über weitere Eigenschaften des *Connection*-Objekts informieren Sie sich bitte in der Access-Hilfe oder, besser, im Web an der o.a. Stelle.

Methoden

Für ein *Connection*-Objekt sind die folgenden Methoden definiert:

Methode	Beschreibung
BeginTrans, CommitTrans, RollbackTrans	Mit diesen Methoden kann eine Transaktionsgruppe gebildet werden, die dann bestätigt (*CommitTrans*) oder verworfen (*RollbackTrans*) werden kann.
Close	Beendet die Verbindung. Diese Methode erscheint weniger wichtig bei Deklaration der *Connection*-Variablen auf Prozedurebene als auf Modulebene, weil die Verbindung im letzteren Falle auch nach Beendigung der Prozedur fortbesteht, wenn sie nicht mit *Close* beendet wird.
Execute	Ergibt ein *Forward-Only-Recordset*-Objekt aus einem SQL-Select-Statement.
Open	Öffnet eine Verbindung. Der *ConnectionString* wird entweder als Argument zu dieser Methode angegeben, oder er wurde mit einer vorangehenden Anweisung der *Connection*-Eigenschaft *ConnectionString* zugewiesen.
Openschema	Ergibt ein Recordset, das Informationen über die Struktur der Datenquelle (Metadaten) enthält.

Aufbau einer *Connection* zur aktuellen Datenbank

Wenn eine Verbindung zur aktuellen Datenbank – also der Datenbank, in der die betreffende Prozedur sich befindet und die geöffnet ist – hergestellt werden soll, darf dies nicht in der o.a. Weise über die Methode *Open* und den *ConnectionString* erfolgen, sondern muß unter Verwendung der speziellen Access-Eigenschaft *CurrentProject* geschehen. Das folgende Code-Beispiel demonstriert dies: In der Prozedur *ADO_AktDB* wird die Connection durch die Code-Zeile

Set cnnAktDB = CurrentProject.Connection

aktiviert statt durch die *Connection*-Methode *Open*. Da mit dieser Anweisung eine Verbindung zur aktuellen Datenbank hergestellt wird, entfallen auch weitere Verbindungsangaben: Nicht nur *Data Source*, *User ID* oder *Password*, sondern auch auf die *Provider*-Angabe kann verzichtet werden.

```
Sub ADO_AktDB()
Dim cnnAktDB As New ADODB.Connection
Dim rstTab As New ADODB.Recordset
Set cnnAktDB = CurrentProject.Connection
Set rstTab.ActiveConnection = cnnAktDB
rstTab.CursorType = adOpenKeyset
```

```
rstTab.Source = "Select * From Projekte"
rstTab.Open
For Each Feld In rstTab.Fields
  Debug.Print Feld.Name, rstTab(Feld.Name)
Next

End Sub
```

31.5.3 Das Recordset-Objekt

Wenn Sie ADO verwenden, ändern Sie Daten fast ausschließlich mit Hilfe der Recordset-Objekte. Das Recordset-Objekt bezieht sich stets nur auf einen einzelnen Datensatz innerhalb der Gruppe, der als aktueller Datensatz bezeichnet wird.

Eigenschaften

Das *Recordset*-Objekt besitzt 26 Eigenschaften. Diese können hier nicht alle aufgeführt werden, sondern müssen der Access-Hilfe entnommen werden. Neben solchen Eigenschaften wie *BOF* (Begin Of File), *EOF* (End Of File), *Bookmark* oder *RecordCount*, die ganz ähnliches leisten wie die entsprechenden Eigenschaften beim DAO-*Recordset*-Objekt, existieren hier aber auch ADO-spezifische wie *CursorType* oder *ActiveConnection*. Letztere ist besonders wichtig für den Verbindungsaufbau:

ActiveConnection. Diese Eigenschaft wird auf ein zuvor geöffnetes *Connection*-Objekt eingestellt, mit der das *Recordset*-Objekt verbunden werden soll. Es kann auch ein voll qualifizierter *ConnectionString* angegeben werden.

Unter den Eigenschaften des *Recordset*-Objekts findet sich keine, die der DAO-*Recordset*-Eigenschaft *NoMatch* gleich oder ähnlich ist. Mit dieser Eigenschaft wird bei Anwendung der DAO-Find-Methoden geprüft, ob ein Datensatz, der den angegebenen Suchkriterien entspricht, gefunden wurde. Diese Prüfung erfolgt bei ADO auf andere Weise, vgl. weiter unten zur *Find*-Methode.

Methoden

Die Methoden des ADO-Recordset ähneln denen des DAO-Recordset sehr, obwohl sie in ihren Einzelheiten vielfach Abweichungen aufweisen. Da es sich um 24 Methoden handelt, können auch diese hier nicht im Einzelnen wiedergegeben werden. Es seien lediglich zwei Besonderheiten hervorgehoben:

Open. Mit dieser ADO-spezifischen *Recordset*-Methode öffnen Sie ein Recordset auf Basis einer aktiven *Connection*. Als Argumente zu dieser Methode können Sie Verbindungsargumente anführen. Sie müssen dies tun, wenn dies nicht mit den entsprechenden Recordset-Eigenschaften wie *Source* oder *CursorType* geschehen ist.

Close. Wie bei der *Connection*-Variablen erscheint diese Methode weniger wichtig bei Deklaration auf Prozedurebene als auf Modulebene, weil die Verbindung im letzteren Falle auch nach Beendigung der Prozedur fortbesteht, wenn sie nicht mit *Close* beendet wird.

Find. Mit dieser Methode suchen Sie in einem Recordset auf Basis bestimmter Suchkriterien nach einem Datensatz. Die Suchkriterien werden durch eine SQL *Where*-Klausel angegeben, jedoch ohne das Wort *Where*. Darüber hinaus sind weitere Argumente zu dieser Methode möglich, mit denen Sie die Suchrichtung oder den Beginn der Suche ab einem bestimmten Datensatz spezifizieren können. 1) Die *Find*-Methode des ADO-*Recordset*-Objekts unterscheidet sich von der entsprechenden Regelung bei DAO vor allem in zwei Punkten: Stehen dort die vier Methoden *FindFirst*, *FindLast*, *FindNext* und *FindPrevious* zur Verfügung, so gibt es hier nur die eine Methode *Find*, zu der Sie allerdings die Suchrichtung angeben können. Außerdem können Sie die Suche bei einer bestimmten *Bookmark* beginnen lassen, so daß im Ganzen dieselben Leistungen wie mit den vier DAO-*Find*-Methoden erreicht werden können, wenngleich auf unbequemere Weise. 2) Anders als bei DAO ist für ein ADO-*Recordset*-Objekt keine Eigenschaft *NoMatch* definiert. Sie können den Sucherfolg jedoch mit Hilfe der beiden Eigenschaften *BOF* und *EOF* kontrollieren: Wenn kein Datensatz gefunden wurde, steht der Cursor je nach Suchrichtung am Ende oder am Anfabg des Recordsets, so daß entweder *BOF* oder *EOF* den Wert *True* besitzt. Umgekehrt: Wenn keine dieser beiden Eigenschaften *True* ist, war die Suche erfolgreich.

31.5.4 Weitere Features im ADO-Modell

Wie oben bereits erwähnt, setzt Microsoft für die Zukunft ganz auf ADO (bzw. OLE DB für C-Programmierer). DAO wird demgegenüber als auslaufendes Modell betrachtet. Dies begründet sich vor allem mit den größeren Möglichkeiten, die das ADO-Konzept gegenüber dem DAO-Konzept besitzt. Die größeren Möglichkeiten liegen zum einen in vielen Details, die hier natürlich nicht angeführt werden können. Sie ergeben sich aber auch aus der viel größeren Perspektive und inzwischen gesammelter Erfahrung, mit der ADO im Vergleich zu DAO entwickelt wurde. Zwei Features seien hervorgehoben, die in ADO realisiert, in DAO jedoch nicht verfügbar sind.

Ereignismodell. Obwohl die Anwendung Access zahlreiche Ereignisse kennt, sind den verschiedensten Objekten von DAO keine Ereignisse zugeordnet. Im ADO Modell ist dies anders: Objekte wie *Connection* oder *Recordset* besitzen Ereignisse, auf die Sie sich im Code beziehen und nach denen Sie daher Programmabläufe steuern können. Beispielsweise besitzt ein *Recordset*-Objekt das Ereignis *EndOfRecordset*, das eintritt, wenn der Datensatz-Cursor sich hinter den letzten Datensatz zu bewegen versucht. Damit können Sie u.a. verhindern, daß der Cursor auf diese Position kommt, was in vielen Kontexten fehlerhaft wäre.

ADOMD. In den letzten Jahren hat die Anwendung von OLAP-Technologien stark zugenommen. Damit haben Sie es mit multidimensionalen (statt, wie bei SQL-strukturierten Daten, mit transaktionalen) Daten zu tun. Weder DAO noch ADO können mit multidimensional strukturierten Daten umgehen. Seit kurzem ist aber die ADO-Erweiterung ADOMD verfügbar, die speziell auf den Zugriff auf multidimensionale Daten ausgelegt ist. Es überstiege den Rahmen dieses Buches, hierauf näher einzugehen, vgl. aber meine ausführliche Behandlung des ADOMD-Objekts an anderer Stelle[1].

1. [Brosius], Kap. 14 bis 16.

Kapitel 32

Datenmodellierung für Fortgeschrittene

32.1 Grundlegende Überlegungen zur Datenmodellierung

Die Entwicklung einer Datenbank ist kein Selbstzweck, sondern stellt im allgemeinen den Versuch dar, ein Teilsystem der realen Welt – z.B. eine Auftragsverwaltung, ein System zum optimalen Outsourcing oder ein Controllingsystem – auf dem Computer nachzubilden. Um die sehr unterschiedlichen Gegenstände, Sachverhalte, Ereignisse, Zusammenhänge usw. der realen Welt in dem relationalen Datenbank-Managemensystem (RDBMS) MS Access abzubilden, stehen Ihnen Tabellen, Abfragen, Formulare, Berichte, Makros, VBA-Module, Steuerelemente und verschiedene weitere Objekte der grafischen Benutzeroberfläche zur Verfügung. Die vorangehenden Kapitel dieses Buches haben ausführlich erklärt, wie Sie diese Objekte im einzelnen einsetzen. Dabei wurde im allgemeinen von dem Problem abstrahiert, wie Teilsysteme der realen Welt angemessen in eine relationale Datenbank umgesetzt werden. Darum soll es hier gehen. Allerdings beschäftige ich mich in diesem Kapitel nicht mit allen Aspekten der Entwicklung eines Datenbank-Managementsystems, sondern ich konzentriere mich ausschließlich auf den Aspekt der *Datenmodellierung*. Die folgenden Überlegungen dieses Kapitels werden zeigen, warum dies gerechtfertigt ist. Um es in wenigen Sätzen vorweg zu nehmen: Der Datenmodellierung kommt im Prozeß der allgemeinen Anwendungsentwicklung eine überragende und grundlegende Bedeutung zu, denn ohne eine konsistente Datenstruktur wird eine Datenbank auf Dauer nicht widerspruchsfrei funktionieren. Außerdem treten beim Prozeß der Datenmodellierung verschiedene Probleme auf, deren Lösung nicht immer leicht fällt, die aber vor allem deshalb zu einem schlechten Ergebnis führen können, weil sie oft schlicht und einfach übersehen werden. Aus diesen Gründen erscheint es gerechtfertigt, der Datenmodellierung besondere Aufmerksamkeit zu widmen.

Sie finden auch in der Online-Hilfe von Access eine Anleitung zum Datenbankentwurf, die im Vergleich zu den übrigen Hilfe-Texten sogar relativ ausführlich ist (Online-Hilfe Thema *Entwerfen einer Datenbank*). Access rät Ihnen zu dieser Schrittfolge:

1. »Legen Sie den Verwendungszweck Ihrer Datenbank fest.
2. Bestimmen Sie die Tabellen, die Sie in der Datenbank benötigen.
3. Bestimmen Sie die Felder, die Sie in den Tabellen benötigen.
4. Bestimmen Sie Felder mit eindeutigen Werten in jedem Datensatz.
5. Legen Sie die Beziehungen zwischen den Tabellen fest.
6. Überarbeiten Sie Ihren Entwurf.
7. Fügen Sie Daten hinzu, und erstellen Sie andere Datenbankobjekte.
8. Verwenden Sie die Analyse-Tools von Microsoft Access.«

Zu jedem dieser Schritte wird ein, z.T. nach weiteren Begriffen in der Tiefe gestaffelter, Text angeboten. Es ist zu empfehlen, diese Texte nachzulesen, weil Sie darin nützliche Hinweise zur prinzipiellen Vorgehensweise finden. Wenn Sie jedoch etwas tiefergehende Probleme lösen wollen, werden Sie dort keinen Rat finden. So wird zwar auf das (überragende) Problem der Redundanz in Tabellen hingewiesen, jedoch hört der Text der Online-Hilfe dort auf, wo die Probleme anfangen:

> »Entwerfen Sie Tabellen ohne redundante Daten. Ein durchdachter Datenbankentwurf ist Voraussetzung für schnelle Datenabrufe und –aktualisierungen. Wenn bestehende Tabellen redundante Daten enthalten, können Sie den Tabellenanalyse-Assistenten verwenden, um Ihre Tabellen in verknüpfte Tabellen zu teilen und damit Ihre Daten effizienter zu speichern.«

Der Hinweis auf den Tabellenanalyse-Assistenten ist leider nur eine scheinbare Hilfe, denn er ist, anders als der Hilfe-Text suggeriert, tatsächlich nicht in der Lage, eine zufriedenstellende Aufteilung einer Tabelle mit redundanten Daten auf mehrere Tabellen zu bewerkstelligen. Umgekehrt mag es Ihnen ergehen, wenn Sie einen Blick in eines der üblichen Lehrbücher zur Datenbank-Modellierung werfen, an denen in der Informatik-Literatur kein Mangel herrscht. Nach meiner Erfahrung aus Datenbank-Seminaren erscheinen diese Bücher vielen zu theoretisch, jedenfalls im Sprachgebrauch allzu weit von der Sprache des Datenbank-Praktikers entfernt.

Es ist nicht nur für den Datenbank-Entwickler nützlich, sich gründlich mit der Datenmodellierung zu beschäftigen, bevor er daran geht, ausgefeilte Formulare zu erstellen oder VBA-Code zu schreiben. In die Datenmodellierung sollte vielmehr auch der Auftraggeber, der hier stellvertretend für die diversen künftigen Benutzer des Datenbanksystems gemeint ist, einbezogen werden, denn es ist vielfach bestätigte Erfahrung, daß das Einbeziehen des

Auftraggebers in die Datenmodellierung auch bei diesem das Bewußtsein für manches Problem erst entstehen läßt, das ohne seine Mitwirkung nicht sichtbar geworden wäre.

32.1.1 Realwelt und Datenbank-Managementsystem (DBMS)

Bevor der Entwurfsprozeß für eine Datenbank überhaupt eingeleitet werden kann, muß zunächst einmal eine ausführliche Anforderungsanalyse erstellt werden. Dabei handelt es sich um eine möglichst umfassende und genaue Beschreibung der Sachverhalte und Aufgaben, die in der realen Welt mit Hilfe des Datenbanksystems abgebildet und bewältigt werden sollen. Diese Zweckbestimmung, die u.a. durch Auswertung vorhandener Formulare, Rechnungen, Interviews mit Sachbearbeitern, Fragebögen, Sichtung vorhandener Datenbestände usw. unterstützt werden kann, muß zwar am Anfang jeder weiteren Überlegung stehen, jedoch wird sie sich auch im Laufe der weiteren Entwicklung der Datenbank noch weiter präzisieren. Bild 32.1 gibt schematisch wieder, in welchen Schritten Sie von der Beschreibung der *Sachverhalte der Realwelt* zum gesamten Informationssystem kommen.

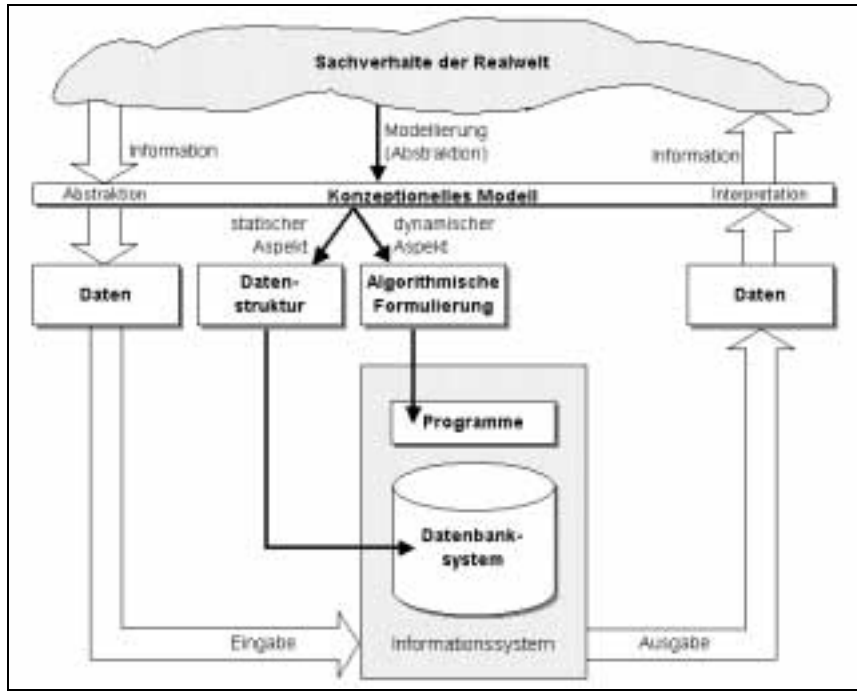


Bild 32.1: Modellierung eines Informationssystems, nach Behme/Ohlendorf, S. 118.

Nachdem mit der Anforderungsanalyse festgestellt wurde, welche Sachverhalte der realen Welt berücksichtigt und welche Aufgaben gelöst werden sollen, müssen diese in verbalen Beschreibungen formalisiert werden. Bei die-

sem Schritt geht es darum, durch Abstraktion und Konzentration auf das Wesentliche die genauen strukturellen Zusammenhänge zwischen den Sachverhalten der Realwelt zu erkennen, Unvollständigkeiten festzustellen und Widersprüche zu entdecken. Das Ergebnis dieser Phase ist ein *konzeptionelles Modell*. Um dieses auszudrücken, bedarf es einer geeigneten Entwurfssprache, welche die konzeptionellen Zusammenhänge formal präzise beschreibt, dies jedoch noch unabhängig von der Umsetzung in ein konkretes EDV-Informationssystem (RDBMS + Programme) leistet. Diesen Zweck erfüllen sogen. *semantische Datenmodelle*, die von der Informatik in verschiedenen Ausprägungen angeboten werden. Vorherrschend ist, insbesondere für relationale Datenbanken, die sogen. Entity-Relationship-Modellierung, auf die ich weiter unten etwas genauer eingehe, vgl. in diesem Kapitel Punkt 32.1.4, *Entity-Relationship-Modell* (ERM) und Relationenmodell. Von den verschiedenen Schritten der Umsetzung eines Teilsystems der realen Welt in ein konkretes Datenbanksystem (einschließlich Programmierung) beschäftigt sich das vorliegende Kapitel im wesentlichen mit dem konzeptionellen Modell, weil dies das Kernstück jeder Datenbankmodellierung ist.

32.1.2 Externe, konzeptionelle und interne Sicht

Eine wichtige Forderung bei der Datenbankmodellierung ist die nach *Unabhängigkeit der logischen von der physischen Datenstruktur (oder Datenorganisation)*. Hiermit ist Folgendes gemeint: Damit der aus der realen Welt gesetzte Zweck des Datenbanksystems insgesamt erfüllt werden kann, bedarf es einerseits der angemessenen Datenmodellierung, d.h. eines Konzepts der Umsetzung der verschiedensten Sachverhalte der realen Welt in eine adäquate Datenstruktur. Andererseits können nicht alle Anforderungen der realen Welt allein durch eine geeignete Datenmodellierung erfüllt werden. Beispielsweise können die Anforderungen eines Benutzers an möglichst komfortable Eingabe der Daten oder gar deren Analyse, z.B. mittels einer Grafik, nicht durch Datenmodellierung allein erfüllt werden. Dazu müssen möglicherweise VBA-Prozeduren geschrieben oder Formulare erstellt werden, die mit der Datenmodellierung direkt nichts zu tun haben. Damit ein konkretes DBMS effizient und zuverlässig funktioniert, müssen die Daten darüber hinaus auch physisch auf konkreten Maschinen (z.B. PC mit Festplatte, oder verteilt im Netz, oder bereit für den Zugriff anderer Anwendungen) verarbeitet werden, und dies ist eine ganz andere Ebene als die der Datenmodellierung.

Mit *logischer und physischer Datenunabhängigkeit* ist daher gemeint, daß die logische (konzeptionelle) Organisation der Daten unabhängig von der physischen Organisation erfolgt. Wenn Sie diese Forderung bei der Entwicklung eines Datenbanksystems berücksichtigen, haben Sie den Vorteil, daß eine Änderung auf der physischen Ebene (z.B. ein neues Betriebssystem oder gar ein anderes, effizienteres Datenbanksystem) Ihre Formulierung der logischen Datenstruktur nicht beeinträchtigt. Sie können sich daher hinsichtlich der Datenstrukturierung allein auf die logische Seite der Angelegenheit kon-

zentrieren und Überlegungen zur physischen Seite entweder gänzlich dem von Ihnen verwendeten DBMS (hier: MS Access) überlassen oder eigene Überlegungen dazu – z.B. zu einem möglichen Übergang von der JetEngine von Access zum Microsoft SQL-Server – anstellen, ohne daß dieses Rückwirkungen auf das Konzept der Datenstruktur hat.

Um die Datenstrukturen logisch unabhängig zu konzipieren, erweist es sich als nützlich, die drei Ebenen der *externen*, *konzeptionellen* und der *internen* Sicht zu unterscheiden und zu beachten, wie dies im sogen. Drei-Sichten-Modell der ANSI/SPARC Study Group on Database Management24.10.2001 vorgeschlagen ist, vgl. Bild 32.2.

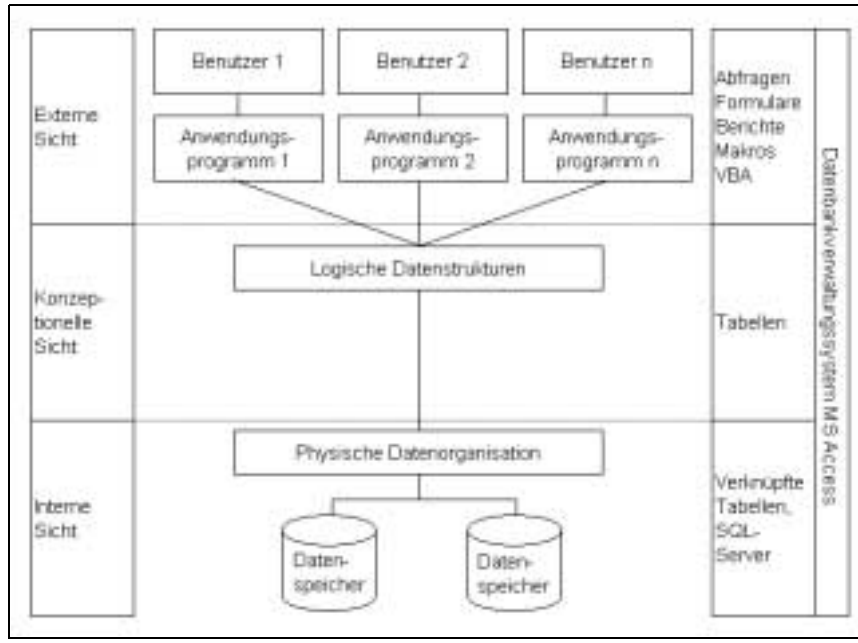


Bild 32.2: Drei-Sichten-Modell von ANSI/SPARC, dargestellt für das DBMS MS Access (rechter Teil), hier nach Stahlknecht, S. 219.

Dieses Schema ist folgendermaßen zu verstehen:

Interne Sicht. Bei dieser Sicht geht es um die *physische* Organisation der Daten. Hierbei spielen z.B. Fragen der Organisierung oder Strukturierung der Daten im Speicher eine Rolle. Bei einem modernen RDBMS wie MS Access braucht der Entwickler sich um diese Fragen nicht zu kümmern, weil sie von diesem System gelöst werden. Zur physischen Datenorganisation gehört auch die Frage, ob Sie alle Tabellen und die anderen Access-Objekte in ein und derselben Datenbank halten wollen, oder ob Sie die Tabellen in einer Back-End-Datenbank ablegen und sie dann mit der Front-End-Datenbank, die alle anderen Objekte enthält, verknüpfen.

Externe Sicht. Hiermit ist die Sichtweise der Benutzer bzw. Anwender gemeint. Dabei gibt es im allgemeinen viele Benutzer einer Datenbank, die oft sehr unterschiedliche Anforderungen an diese stellen. Entsprechend existieren im allgemeinen viele verschiedene Sichtweisen der Benutzer, die häufig jeweils nur einen Ausschnitt des gesamten Datenmodells betreffen Darüber hinaus verlangen die Anforderungen der verschiedenen Benutzer oft auch, daß bestimmte Anwendungsprogramme für deren Befriedigung erstellt werden.

Konzeptionelle Sicht. Diese Sicht betrifft ausschließlich die logische Datenstrukturierung, weshalb sie oft auch als die *logische Sicht* bezeichnet wird. Entscheidend ist, daß die konzeptionelle Sicht unabhängig von der externen und der internen Sicht ist: Verfolgt die externe Sicht die Anforderungen der Benutzer und führt dabei zu bestimmten Vorschlägen z.B. hinsichtlich Formulargestaltung oder VBA-Prozeduren und versucht die interne Sicht, durch physische Datenorganisation vor allem Gesichtspunkten der Performance Rechnung zu tragen, so steht bei der konzeptionellen Sicht die widerspruchsfreie, konsistente und redundanzfreie (vgl. hierzu ausführlich weiter unten in diesem Kapitel) Datenstrukturierung im Mittelpunkt.

Die wichtigste Erkenntnis, die der Entwickler einer Datenbank für deren Entwurf aus dem Drei-Sichten-Modell (auch Drei-Schichten-Modell genannt) ziehen kann, ist die der *Unabhängigkeit der konzeptionellen Ebene von den anderen beiden Ebenen*. Diese Unabhängigkeit betrifft mehrere Aspekte und hat besondere praktische Bedeutung im Vergleich der externen mit der konzeptionellen Ebene. Eine Nichtbeachtung dieser Unabhängigkeit könnte z.B. dazu führen, daß bestimmte spezielle Anforderungen eines Benutzers statt auf der externen auf der konzeptionellen Ebene und damit innerhalb der logischen Datenstrukturierung zu realisieren versucht würden. Dies aber könnte zu Widersprüchen innerhalb der Datenstruktur und/oder dazu führen, daß den Anforderungen anderer Benutzer nur schlecht oder gar nicht Rechnung getragen würde. Die Anforderung eines bestimmten Benutzers könnte es beispielsweise sein, stets gleichzeitig über die Adresse eines Kunden und seine Bestellungen zu verfügen, so daß er aus seiner Sicht Bestellungen bequem bearbeiten kann. Wenn diese Sichtweise in die Datenstrukturierung hineinregieren würde (die konzeptionelle also nicht gegen die externe Sicht abgeschirmt wäre), könnte dies dazu führen, daß die Kunden- und Bestelldaten zusammen in eine Tabelle gebracht würden. Abgesehen davon, daß dies vermutlich zu Redundanz und damit zur Möglichkeit von Widersprüchen in den Daten führen würde, könnten damit gleichzeitig die Anforderungen anderer Benutzer, für die gerade der getrennte Zugriff auf Kunden- und Bestelldaten wichtig sein mag, beeinträchtigt werden. Das Datenmodell sollte daher nicht auf bestimmte Benutzersichtweisen ausgerichtet sein, sondern so, daß es allen Anforderungen aller Benutzer gerecht wird, was nur dadurch zu realisieren ist, daß es umfassend und allgemein formuliert wird. Die jeweils speziellen Anforderungen und Sichtweisen ein-

zelner Benutzer oder Benutzergruppen müssen und können dann durch Werkzeuge und Methoden, die nichts mit der Datenmodellierung zu tun haben, erfüllt werden, beispielsweise durch Abfragen und Formulare.

Für das RDBMS Access bedeutet dies konkret, daß sich die praktische Realisierung eines logischen Datenmodells ausschließlich in der Aufteilung der Daten auf Tabellen niederschlägt. Die externe Sicht dagegen findet ihren Niederschlag in der konkreten Ausgestaltung von Abfragen, Formularen, Berichten, Makros und VBA-Prozeduren.

32.1.3 Entity-Relationship-Modell (ERM) und Relationenmodell

Bei der Datenmodellierung kommt es darauf an, Sachverhalte von Ausschnitten der realen Welt – in diesem Zusammenhang oft auch als *Miniwelt* bezeichnet – in einem logisch konsistenten Datenmodell abzubilden. Der dazu erforderliche intellektuelle Prozeß muß vom Datenbankdesigner geleistet werden, und in jedem Falle ist für die Abbildung der Miniwelt als Datenmodell Intuition, Phantasie und nicht zuletzt Erfahrung notwendig. Andererseits hat sich in der Informatik in den letzten zwei Jahrzehnten für den Zweck der Datenmodellierung auf der konzeptionellen Ebene eine Beschreibungssprache als vorherrschend herausgebildet, die als *Entity-Relationship-Modell* (kurz: *ERM*) bekannt ist und auf einen Aufsatz von Chen[1] zurückzuführen ist. In der Online-Hilfe von Access werden Sie keinen der zentralen ERM-Begriffe finden, weil diese offenbar bewußt darauf verzichtet, auf diese mehr theoretische Welt Bezug zu nehmen. Auch ich habe lange gezweifelt, ob es sinnvoll ist, für die Entwicklung von Access-Datenbanken, die ja im Vergleich zu SQL-Datenbanken oder solchen auf Großrechnern im allgemeinen weniger komplex ausfallen dürften, in diesem Buch den Umweg über das Entity-Relationship-Modell zu gehen und Sie mit neuen Begriffen wie *Entity* und *Attribut* zu belasten, statt gleich und ausschließlich mit den diesen entsprechenden Begriffen *Tabelle* und *Feld* zu arbeiten. Für diesen (produktiven) Umweg sprechen aber die folgenden Überlegungen:

- Wenn Sie sich in der allgemeinen Datenbank-Literatur gründlicher über Datenbankdesign unterrichten wollen, werden Sie in aller Regel auf die ER-Begrifflichkeit stoßen. Sie ist mehr oder minder Standard als Beschreibungssprache zur Datenmodellierung.
- Die Entity-Relationship-Modellierung ist von vornherein auf Visualisierung der Sachverhalte bereits beim Entwurf abgestellt. Access bietet dagegen für normale Datenbanken im Fenster *Beziehungen* nur eine nachträgliche Visualisierung der definierten Tabellen, Felder und Beziehungen, die in diesem Fenster für Designzwecke nicht geändert werden können. (Für Access Projekte verhält es sich allerdings in dieser Hinsicht anders, vgl den nächsten Aufzählungspunkt.)

1. [Chen]

- Für den Zweck der Datenmodellierung nach dem Entity-Relationship-Modell existieren verschiedene CASE-Tools (Computer Aided Software Engineering), mit deren Hilfe das Entwerfen eines Datenmodells logisch und praktisch erleichtert wird. Als ein besonders gut gelungenes Produkt ist hier das CASE-Tool ERwin der Firma Logic Works hervorzuheben. Mit diesem Werkzeug lassen sich Datenmodelle grafisch entwickeln, die dann mittels SQL-Anweisungen in die entsprechende Access-Tabellenstruktur (sowie in andere Datenbankformate wie Oracle) umgesetzt werden können. Aber auch das Entwurfswerkzeug *Datenbankdiagramm*, das für Access *Projekte*, die ja mit einer SQL-Datenbank arbeiten, verfügbar ist, lassen sich SQL-Datenbanken unmittelbar in ERM-Manier mit grafischen Tools entwerfen und definieren.

Zusammengefaßt: Auch Access-Entwickler können auf Dauer bei anspruchsvoller Arbeitsweise keinen Bogen um die Entity-Relationship-Modellierung machen, jedenfalls sollte ihnen die entsprechende Terminologie vertraut sein.

Grundlegende ER-Begriffe

Entität. Eine Entität (wörtlich: Seiendes) repräsentiert Objekte der realen Welt, wie diese sich in unserer Anschauung darstellen. Entitäten können gegenständlich sein (wie Kunde, Personal) oder auch abstrakt (wie Kategorien, Bestellungen). In der natürlichen Sprache treten Entitäten meistens als Substantive auf. Im Rahmen der ER-Modellierung kommt es darauf an, Sachverhalte oder Gegenstände der realen Welt als in sich möglichst homogene Entitäten zu erkennen, die gegen andere Entitäten möglichst klar abgrenzbar sein sollen. In einem ERM werden Entitäten meistens als grafische Rechtecke dargestellt. In einer relationalen Datenbank, also auch bei Access, werden Entitäten als Tabellen modelliert: Jede Entität wird durch eine Tabelle wiedergegeben.

Attribut. Entitäten besitzen *Attribute* (Eigenschaften). Beispielsweise kann die Entität *Kunde* die Eigenschaft *Nachname* oder *Firmenname* besitzen. »Die Attribute nehmen den eigentlichen Informationsgehalt des Datenmodells auf, das durch die Entitäten lediglich strukturiert wird.«[2] Attribute werden in realen Tabellen als *Felder* wiedergegeben, der theoretische Begriff *Attribut* entspricht daher dem praktischen Objekt *Feld*.

Beziehung. Eine Beziehung drückt den Zusammenhang zwischen zwei Entitäten aus. In einem ERM werden Beziehungen als gerade Linien zwischen zwei Entitätssymbolen dargestellt. Die genauere Art der Beziehung wird häufig als Beschriftung dieser Linie angegeben. In Beziehungen schlagen sich oft auch Geschäftsregeln des Teilsystems der realen Welt nieder. In Sätzen der natürlichen Sprache erscheinen Beziehungen meistens als Prädikate, wes-

2. [Sommer], S. 127

halb man diese in ER-Modellen oft auch bewußt so anführt. Beispielsweise entspricht dem natürlich-sprachlichen Satz *Lieferant liefert Artikel* eine Beziehung zwischen der Tabelle *Lieferant* und der Tabelle *Artikel*.

32.2 Tabellen bestimmen

32.2.1 Aufteilung der unterscheidbaren Themen auf einzelne Tabellen

Der erste Schritt bei der Datenmodellierung besteht im allgemeinen darin, daß Sie versuchen, die Informationen bzw. Daten, die in der Datenbank verarbeitet werden sollen, auf mehrere Tabellen zu verteilen. Bei dieser Aufteilung der Daten gilt es vor allem, die beiden folgenden Gesichtspunkte zu berücksichtigen:

- Das Tabellenthema soll einen homogenen Gegenstand oder Sachverhalt der Realwelt betreffen, und die einzelnen Tabellenthemen sollen sich klar voneinander unterscheiden. Wenn jede Tabelle nur Daten zu einem einzigen Thema enthält, können Sie die Informationen zu jedem Thema unabhängig von anderen Themen verwalten. Wenn Sie beispielsweise *Bestellungen* und *Artikel* in zwei verschiedenen Tabellen wiedergeben, können Sie einzelne Bestellungen löschen, ohne die Informationen über die bestellten Artikel, die Sie möglicherweise weiterhin benötigen, gleichzeitig zu verlieren. Darüber hinaus läßt sich eine später erforderliche Änderung der Datenstruktur im allgemeinen leichter durchführen, wenn Sie Tabellen zu homogenen Themen gebildet haben, weil dies ja gleichzeitig eine modulare Arbeitsweise darstellt. Auch unter dem Gesichtspunkt von Zugriffsrechten ist eine klar unterscheidbare Aufteilung der Themen auf verschiedene Tabellen geboten, weil Sie Zugriffsrechte – zumindest in Access – zwar für ganze Tabellen, nicht aber für einzelne Felder vergeben können.
- Jede einzelne Tabelle und die Tabellen in ihrer Gesamtheit sollen frei von Redundanz sein, um Widersprüche in der Datenpflege zu vermeiden.

In diesem Punkt werden die Fragen behandelt, die mit dem ersten Aufzählungspunkt zusammenhängen. Probleme, die aus der Forderung nach Redundanzfreiheit resultieren, werden weiter unten in diesem Kapitel besprochen, vgl. Punkt 32.5, *Das Redundanzproblem*. Obwohl sich diese beiden Gesichtspunkte in der Logik klar voneinander unterscheiden lassen, erscheinen sie im praktischen Datenbankdesign oft nicht ganz unabhängig voneinander: Die Aufteilung der Daten auf gegeneinander abgrenzbare Themen führt zwar meistens schon zu einer Verringerung von Redundanz, und außerdem versucht man im allgemeinen auch intuitiv erfolgreich, Redundanz in Tabellen zu vermeiden. Dies ist aber keine Garantie für völlige Redundanzfreiheit, so daß Sie die (vorläufig) gefundenen thematisch homogenen Tabellen im allgemeinen noch daraufhin überprüfen müssen, ob sie Redundanz enthalten. Die mit dieser Prüfung verbundenen Überlegungen führen

dann aber oft auch dazu, daß Sie eine andere Auffassung von der Realwelt selbst bekommen, weil Sie Zusammenhänge entdecken, die Ihnen vorher so nicht klar waren.

Sie sollten Ihren Auftraggeber (der hier stellvertretend auch für seine Mitarbeiter steht) unbedingt an verschiedenen Etappen des Prozesses der Datenaufteilung auf die Tabellen beteiligen, indem Sie Zwischenergebnisse mit ihm besprechen, denn dabei werden beiden Seiten oft erst Widersprüche klar, die bis dahin verborgen waren, und es werden Unklarheiten und Unschärfen als solche erkannt. Dabei sollten Sie die vorgeschlagene *Tabellenstruktur in visualisierter Form präsentieren* und besprechen, z.B. als Wandzeitung oder mit einem Overhead-Projektor. Gelegentlich habe ich erlebt, daß das Zwischenergebnis einer Datenbankentwicklung mit Hilfe mehr oder minder ausgearbeiteter Access-Formulare präsentiert wurde. Dies vermag zwar oberflächlich zu beeindrucken, weil Sie mit Formularen gewisse Funktionalitäten eindrucksvoll demonstrieren können. Datenstrukturen lassen sich damit jedoch nur unzureichend wiedergeben. Wenn Sie daran interessiert sind, zusammen mit Ihrem Auftraggeber eine seinem Informationsbedürfnis entsprechende Datenstruktur zu konzipieren, sollten Sie ihn auf alle Fälle in die Diskussion der Tabellenstruktur einbeziehen. Nach meiner Erfahrung bringen auch Personen, die wenig Erfahrung mit Datenbankprojekten haben, relativ schnell Verständnis für eine aus Tabellen, Feldern und Beziehungen bestehende Datenstruktur auf und können die relevanten Zusammenhänge mit diskutieren.

32.2.2 Grundlegende Beziehungen zwischen den Tabellen bestimmen

Die Beziehungen zwischen den Tabellen lassen sich formal korrekt erst bestimmen, wenn Sie auch Einzelheiten zu den Feldern der Tabellen, vor allem den Primärschlüssel, festgelegt haben. Andererseits lassen die Tabellen erst dann eine sinnvolle Datenstruktur erkennen, wenn die Beziehungen zwischen ihnen sichtbar werden. Daher erscheint es schon in einem frühen Entwurfsstadium geboten, sich Gedanken über die zwischen den Tabellen festzulegenden Beziehungen zu machen und diese auch grafisch in Form von geraden Linien darzustellen. Dabei sollten Sie sich auch über das Mengenverhältnis der Beziehung zwischen den beteiligten Tabellen (1:1, 1:n, m:n) klar werden und dieses ebenfalls visuell ausweisen, damit es mit anderen am Entwurfsprozeß Beteiligten diskutierbar wird. Über weitere Eigenschaften der Beziehungen, wie das mögliche Festlegen referentieller Integrität und damit zusammenhängender weiterer Fragen, können Sie auch in einem späteren Entwurfsstadium, insbesondere nach dem Bestimmen der Primärschlüsselfelder, befinden. Entsprechend werden diese Fragen auch in diesem Kapitel weiter unten behandelt, vgl. Punkt 32.6, *Beziehungen bestimmen*.

32.3 Felder bestimmen

32.3.1 Benötigte Felder identifizieren

Durch Identifizierung der Tabellen, die Sie durch die abgrenzbaren und homogenen Themen des Informationssystems bestimmt haben, haben Sie Ihr Datenmodell auf einer relativ hohen Abstraktionsebene strukturiert. Solange Sie für eine Tabelle jedoch kein Feld festgelegt haben, kann sie keinerlei Information aufnehmen, denn die verschiedenen Informationen werden in den Tabellenfeldern als Feldwerte gespeichert.

Das Bestimmen der für eine Tabelle erforderlichen Felder ist im allgemeinen nicht so schwierig wie das Auffinden der bedeutungs- und sinnvollen Themen und damit Tabellen für das gesamte Datenmodell. Gleichwohl müssen auch hierbei bestimmte Gesichtspunkte beachtet werden, auf die im folgenden genauer eingegangen wird.

Sämtliche relevanten Informationen des Themas berücksichtigen. Sie müßten diesen Hinweis nicht bereits in der Entwurfsphase einer Datenbank beachten, wenn es ohne weiteres möglich wäre, zu einem späteren Zeitpunkt Informationen in eine mit ihren Feldern definierte Tabelle einzugeben, an die beim Entwerfen der Datenbank noch nicht gedacht wurde. Dies ist jedoch nicht zutreffend, weil dabei die Forderung verletzt würde, in Tabellenfelder nur *atomare Werte* einzugeben. Allerdings ist es für den Entwickler – anders als für den Benutzer – verhältnismäßig unproblematisch, einer Tabelle auch später Felder hinzuzufügen.

Nur atomare Werte. Bestimmen Sie die Felder so, daß nur atomare Werte eingegeben werden. Nehmen wir beispielsweise eine Tabelle *Artikel* an, für die die Felder *Artikelnummer*, *Artikelname* und *Einzelpreis* festgelegt sind. Zu einem späteren Zeitpunkt werde dann die Notwendigkeit erkannt, als Bestandteil eines Artikels auch die *Kategorie* zu speichern, der dieser zuzuordnen ist. Man könnte auf die Idee kommen, die Kategorie jeweils zusammen mit dem Artikelnamen in dasselbe Feld einzugeben und zu speichern. Diese Vorgehensweise würde allerdings schnell zu Disfunktionalitäten und möglicherweise sogar zu Widersprüchen in der Datenbank führen. Die eingeschränkte Funktionalität würde sich z.B. zeigen, wenn Sie mit einer Gruppierungsabfrage den jeweils mittleren Preis aller zu einer Kategorie gehörenden Artikel auflisten möchten, denn dies wäre, zumindest ohne aufwendige Umwege, nicht möglich. Ein Widerspruch bei der Dateneingabe könnte sich z.B. ergeben, wenn der eine neue Artikel mit vorangestelltem Kategorienamen, der nächste mit nachgestelltem, der dritte mit einem trennenden Semikolon, der vierte ohne Semikolon usw. eingegeben würde. Die Chancen, die beiden objektiv trennbaren Informationen *Artikelname* und *Kategorie* nachträglich z.B. in Abfragen sauber getrennt auszuwerten, würden dabei annähernd auf Null sinken.

Wann sind Werte atomar? Diese Frage läßt sich nicht in allgemeiner Weise und schon gar nicht formal beantworten, denn es hängt davon ab, in welchem Zusammenhang und für welchen Zweck die Information später ausgewertet werden soll. Ein typischer Fall sind Straßenname und Hausnummer in einer Adresse. Wenn es sicher ist, daß diese beiden Angaben stets nur zusammen als Bestandteil der Adreßangabe benötigt werden, spricht nichts dagegen, sie in einem Feld abzulegen. Wenn sich andererseits ergeben sollte, daß die Adressen einer bestimmten Straße nach Hausnummern ausgewertet und/oder sortiert werden müssen, wären die beiden Informationen in zwei Feldern abzulegen.

Auf die Bedeutung atomarer Werte wird noch einmal weiter unten in diesem Kapitel im Zusammenhang mit dem Redundanzproblem eingegangen, vgl. Punkt 32.5.3, *Redundanz beseitigen*.

Keine berechneten Felder. Im allgemeinen sollten Sie es vermeiden, Felder für Werte vorzusehen, die sich vollständig aus einer Berechnung aus Werten anderer Felder ergeben. Wenn Sie beispielsweise in einer Tabelle *Bestelldetails* ein Feld für den (Netto-)Preis und ein anderes für den Mehrwertsteuersatz (der bekanntlich für verschiedene Positionen unterschiedlich sein kann) vorsehen, sollten Sie kein Feld für den Mehrwertsteuerbetrag vorsehen, weil sich dieser vollständig aus dem Preis und dem Mehrwertsteuersatz berechnen läßt. Wenn Sie den Mehrwertsteuerbetrag später benötigen (z.B. beim Erstellen der Rechnung), können Sie diesen in einer Abfrage als berechnetes Feld ausgeben lassen. Die Nichtberücksichtigung von Feldern für berechnete Werte resultiert unmittelbar aus der Forderung nach Redundanzfreiheit. Wie diese ist sie auch gewissen Gegenargumenten ausgesetzt, vgl. dazu genauer Punkt 32.5.3, *Redundanz beseitigen*, Unterpunkt *Berechnete Felder.*

Reihenfolge der Felder ist beliebig. Eine Entwurfsregel für relationale Datenbanken kommt dem Entwickler zum Glück entgegen, wenn er sich genötigt sieht, einer Tabellendefinition nachträglich Felder hinzuzufügen: Für das Datenmodell ist es unerheblich, in welcher Reihenfolge die Felder in einer Tabelle definiert sind. Daher können Sie insoweit einer Tabelle auch nachträglich ohne weiteres Felder hinzufügen, wobei Sie diese in der Tabellendefinition sowohl hinter den anderen Feldern anfügen wie auch zwischen bestehenden einfügen können. Diese Möglichkeit sollte Sie allerdings nicht dazu verleiten, nicht bereits beim Entwurf einer Datenbank in der Bestimmung der Felder sorgfältig auf Vollständigkeit zu achten, denn das nachträgliche Einfügen von Feldern bedeutet immerhin ein Redesign der Datenbank und kann daher nicht vom Benutzer vorgenommen werden. Es hat darüber hinaus im allgemeinen weitere Anpassungen zur Folge, die oft mit zusätzlichem Aufwand verbunden sind.

32.3.2 Domänen

Unter der Domäne eines Tabellenfeldes ist der Bereich aller möglichen Werte, die das Feld annehmen kann, zu verstehen. Dieser Bereich kann durch verschiedene Festlegungen auf Feld- und Tabellenebene eingeengt werden. Die Festlegungen betreffen

- Felddatentyp und -größe
- Zulässigkeit von Nullwerten und Leerzeichen
- Gültigkeitsregeln
- Eingabeformate
- Nachschlagefelder

Hierauf wird im weiteren genauer eingegangen.

Felddatentyp und -größe

Für jedes Feld muß ein Datentyp festgelegt werden. Der Datentyp – zusammen mit den ihn weiter spezifizierenden Feldeigenschaften – wirkt sich auf den für ein Feld zulässigen *Wertebereich* (seine Domäne), die für die Werte des Feldes zulässigen *Rechenoperationen* sowie das *Sortieren* aus:

- *Wertebereich*. Der Datentyp legt fest, welche Art von Werten in ein Feld eingegeben werden kann. So lassen sich in ein Feld des Datentyps *Zahl* keine alphanumerischen Zeichen, sondern nur Zahlenwerte eingeben, und ein Feld mit dem Datentyp *Datum/Uhrzeit* nimmt nur Datums- oder Zeitangaben an.
- *Rechenoperationen*. Der Datentyp bestimmt auch, in welcher Weise die Werte eines Feldes von Access interpretiert und verarbeitet werden können. Ein Vergleich der beiden Datentypen *Text* und *Zahl* macht dies deutlich: Sie können dieselbe Zeichenfolge *123* sowohl in ein Feld mit dem Datentyp *Text* wie auch in ein solches mit dem Datentyp *Zahl* eingeben. Im ersten Fall wird die Zeichenfolge als Text interpretiert und kann auch nur als solcher verarbeitet werden, so daß es nicht möglich ist, damit zu rechnen. Im zweiten Fall wird *123* als Zahlenwert 123 aufgefaßt, mit dem dann Rechenoperationen ausgeführt werden können. Die Verarbeitungslogik betrifft vor allem auch den Unterschied zwischen den Datentypen *Zahl* und *Währung*. Für beide Datentypen ist die Eingabe von Zahlenwerten zulässig, jedoch wird bei Rechenoperationen mit Zahlenwerten des Datentyps *Währung* das Abrunden verhindert.
- *Sortieren*. Felder des Datentyps *Memo*, *Hyperlink* und *OLE-Objekt* können nicht sortiert werden. (Ebensowenig lassen sie sich indizieren.) In einem Feld des Datentyps *Text* werden Zahlen als Zeichenfolgen und nicht als numerische Werte sortiert, z.B. in der Reihenfolge 1, 10, 100, 2, 20, 200. Bei den Datentypen *Zahl* oder *Währung* wird dagegen nach den numerischen Werten sortiert, z.B. in der Reihenfolge 1, 2, 10, 20 100,

200. Datumswerte, die in ein Feld vom Datentyp *Text* eingegeben wurden, werden je nach Datumsformat unterschiedlich und im allgemeinen nicht richtig sortiert, vgl. die folgende Gegenüberstellung. Sie sollten daher stets den Datentyp *Datum/Uhrzeit* für Felder verwenden, die Datumswerte aufnehmen sollen.

Datentyp *Text*	**Datentyp *Datum/Uhrzeit***
21. Jan. 99	21. Jan. 99
24. Feb. 99	24. Jan. 99
24. Jan. 99	24. Feb. 99

Datentypen von Primärschlüssel und Fremdschlüssel müssen übereinstimmen. Achten Sie beim Festlegen eines Datentyps für ein Feld auch darauf, daß dieser ggf. zu dem Datentyp eines anderen Feldes passen muß. Dies gilt z.B. beim Definieren einer Beziehung mit referentieller Integrität, bei der die Datentypen des Primärschlüsselfeldes der Mastertabelle und des Fremdschlüsselfeldes der Detailtabelle übereinstimmen müssen. Dabei gibt es allerdings zwei Ausnahmen: 1) Ein Feld des Datentyps *AutoWert* kann mit einem Feld des Datentyps *Zahl* verknüpft werden, dessen Eigenschaft *Feldgröße* auf *Long Integer* eingestellt ist. 2) Ein Feld des Datentyps *AutoWert* kann mit einem Feld des Typs *Zahl* verknüpft werden, wenn die Eigenschaft *Feldgröße* beider Felder auf *Replikations-ID* eingestellt ist.

Die folgende Übersicht gibt die in Access verfügbaren Datentypen wieder und hebt die wichtigsten Charakteristika hervor. Weitere Eigenschaften entnehmen Sie bitte der Online-Hilfe.

Datentypen und ihre wichtigsten Eigenschaften

Felddatentyp	**Beschreibung**	**Größe**
Text	Nimmt Zeichenfolgen (d.h. alphanumerische Zeichen) auf. Mit der Eigenschaft *Feldgröße* kann die zulässige Länge der eingegebenen Zeichenfolgen definiert werden. Dieser Datentyp ist auch für Ziffernfolgen, die nicht als numerische Werte interpretiert werden sollen, geeignet, beispielsweise für Telefonnummern oder Postleitzahlen. Mit einem Eingabeformat können Sie den Wertebereich auch bei diesem Datentyp weiter einschränken, z.B. für Postleitzahlen auf die Ziffern 0 bis 9.	Maximal 255 Zeichen (= Byte). In der Datenbank wird nur so viel Speicherplatz belegt, wie tatsächlich Zeichenfolgen eingegeben sind.

Felddatentyp	Beschreibung	Größe
Memo	Text mit einer großen Länge, geeignet z.B. für Notizen oder längere Beschreibungen. Felder vom Datentyp *Memo* können nicht indiziert werden.	Maximal 64.000 Zeichen (= Byte). In der Datenbank wird nur so viel Speicherplatz belegt, wie tatsächlich Zeichenfolgen eingegeben sind.
Zahl	Numerischer Wert. Zu diesem Datentyp müssen Sie einen von sechs Untertypen angeben. Dies geschieht mit der Feldeigenschaft *Feldgröße*. Im Grunde genommen gibt es gar keinen Datentyp Zahl, denn der konkrete Datentyp ist erst durch Auswahl eines Untertyps bestimmt, und Typprüfungen erfolgen stets in Bezug auf diesen Untertyp. Die sechs speziellen Datentypen sind:	2, 4 oder 8 Byte
	Byte. Ganzzahl mit Werten im Bereich von 0 bis 255	1 Byte
	Integer. Ganzzahl mit Werten im Bereich von -32.768 bis 32.767	2 Bytes
	Long Integer. Lange Ganzzahl mit Werten im Bereich von -2.147.483.648 bis 2.147.483.647	4 Bytes
	Single. Gleitkommazahl mit einfacher Genauigkeit mit Werten im Bereich von -3,402823E38 bis -1,401298E-45 für negative Werte und 1,401298E-45 bis 3,402823E38 für positive Werte.	4 Byte
	Double. Gleitkommazahl mit doppelter Genauigkeit mit Werten im Bereich von -1,79769313486232E308 bis -4,94065645841247E-324 für negative Werte und 4,94065645841247E-324 bis 1,79769313486232E308 für positive Werte.	8 Bytes
	Replikations-ID. Dient zum Definieren einer eindeutigen Replikationskennung und muß für Tabellen, die repliziert werden sollen, definiert sein. Wird auch als Globally Unique Identifier (GUID) bezeichnet.	16 Bytes

Felddatentyp	Beschreibung	Größe
Datum/ Uhrzeit	Datum- und Zeitwerte für die Jahre 100 bis 9999. Bei der Eingabe werden die Jahresangaben 0 bis 29 als 2000 bis 2029 und 30 bis 99 als 1930 bis 1999 interpretiert.	8 Byte
Währung	Währungswerte mit einer Genauigkeit von bis zu 15 Stellen und 4 Dezimalstellen. Dieser Datentyp wird für Währungswerte verwendet, um bei Berechnungen ein Abrunden zu verhindern.	8 Byte
AutoWert	Zahl, die von Access automatisch vergeben wird, wenn ein neuer Datensatz einer Tabelle hinzugefügt wird. Die Werte eines Feldes vom Datentyp *AutoWert* sind eindeutig (ohne Duplikat) und können nicht verändert werden. Zahlen, die bereits vergeben waren, stehen auch nach dem Löschen eines Datensatzes nicht mehr zur Verfügung. Sie können bestimmen, ob neue Zahlen jeweils um den Wert 1 erhöht oder als Zufallszahl eingefügt werden.	4 Byte
Ja/Nein	Ja- und Nein-Werte. Statt Ja dürfen Sie auch -1 oder Wahr oder Ein, statt Nein auch 0 oder Falsch oder Aus eingeben.	1 Bit (= 1/8 Byte)
OLE-Objekt	Objekte, wie Excel-Tabellen, Word-Dokumente, Bilder, Klänge oder andere Objekte mit binären Daten, die von einem OLE-Server erzeugt worden sind. Die Größe kann maximal 1 Gigabyte betragen. Felder vom Datentyp *OLE-Objekt* können nicht indiziert werden.	Maximal 1 GB
Hyperlink	Text oder Kombinationen aus Text und Zahlen, die als Text abgespeichert und als Hyperlink-Adresse verwendet werden. Eine Hyperlink-Adresse besteht aus bis zu drei Teilen: *Anzeigetext*: Der Text, der in einem Feld oder einem Steuerelement angezeigt wird. *Adresse*: Der Pfad auf eine lokale Datei (UNC-Pfad) oder eine Seite im Internet oder im Intranet (URL). *UnterAdresse*: Eine bestimmte Stelle innerhalb einer Datei oder Seite.	Jeder der drei nebenstehend angegebenen Teile kann bis zu 2048 Zeichen lang sein.

Zulässigkeit von Nullwerten oder leerer Zeichenfolge

Der Wert NULL (nicht zu verwechseln mit der Zahl 0) steht für einen fehlenden Wert. Felder, die allein oder zusammen mit anderen Feldern den Primärschlüssel bilden, dürfen keinen Nullwert enthalten. Bei einem aus mehr als einem Feld bestehenden eindeutigen Index dürfen zwar einzelne Felder, nicht aber alle gleichzeitig einen Nullwert haben. Andere Felder dürfen durchaus Nullwerte enthalten. Dies gilt sogar für den Fremdschlüssel in einer Beziehung mit referentieller Integrität. Der Wert NULL wird in relationalen Datenbankprogrammen üblicherweise zum Darstellen eines fehlenden Wertes verwendet. Bei Access können Sie zum expliziten Angeben eines fehlenden Wertes für Felder vom Datentyp *Text* und *Memo* zusätzlich die leere Zeichenfolge »« eingeben. Äußerlich lassen sich die Werte leere Zeichenfolge und Nullwert nicht unterscheiden. Access empfiehlt daher in der Online-Hilfe, für Felder, für die eine leere Zeichenfolge zulässig ist, mit Hilfe eines entsprechenden Anzeigeformates diese explizit kenntlich zu machen. Das folgende Format

@;"unbekannt«;"nicht verfügbar«

sorgt z.B. dafür, daß ein leerer Wert als *unbekannt* und ein Nullwert als *nicht verfügbar* dargestellt wird, vgl. die Wiedergabe im Feld *e-Mail* in Bild 32.3.

	Nachnan	Vorname	e-Mail
	Meier	Horst	MeierHorst@uni-hamburg
	Müller	Emil	unbekannt
	Schultze	Gerda	nicht verfügbar
▶			nicht verfügbar

Bild 32.3: Das Feld »e-Mail« vom Datentyp »Text« hat das Anzeigeformat '@;"unbekannt«;"nicht verfügbar"'.

Sie lassen Nullwerte für ein Feld zu, wenn Sie dessen Eigenschaft *Eingabe erforderlich* auf *Nein* einstellen. Leere Zeichenfolgen lassen Sie dadurch zu, daß Sie die Eigenschaft *Leere Zeichenfolge* auf *Ja* einstellen. Da beide Eigenschaften unabhängig voneinander festgelegt werden können, sind vier Einstellungskombinationen möglich, die Sie je nach Sachzusammenhang verwenden können. Beachten Sie jedoch die folgenden Hinweise, die für oder gegen die Verwendung von Nullwerten und leeren Zeichenfolgen sprechen können:

Eingabekomfort. Wenn Sie ein Feld bei der Eingabe einfach frei lassen, wird diesem bei der Speicherung des Datensatzes ein Nullwert zugewiesen. Wenn Sie eine leere Zeichenfolge eingeben wollen, müssen Sie sich dagegen stets explizit zu dem betreffenden Feld bewegen und die leere Zeichenfolge als zwei doppelte Anführungszeichen eingeben. (Bei der Kombination *Eingabe erforderlich* = Ja und *Leere Zeichenfolge* = Ja können Sie auch die [ ]-

Taste oder die [Entf]-Taste zum Eingeben einer leeren Zeichenfolge drükken.) Wenn es oft vorkommt, daß mehrere Felder eines Datensatzes einen fehlenden Wert besitzen und nicht zwischen verschiedenen fehlenden Werten unterschieden werden muß, spricht der Eingabekomfort für die generelle Verwendung von Nullwerten statt leerer Zeichenfolgen.

Steuerelementwerte als Kriterien in Abfragen. In Access können Sie Abfrageergebnisse mit unterschiedlichen Kriterienwerten sehr komfortabel und benutzerfreundlich anbieten, indem Sie als Abfragekriterien Steuerelementwerte eines für diesen Zweck entworfenen Kriterienformulars verwenden, vgl. dazu im einzelnen Kap. 12, *Komplexe Abfragen*. Wenn in einem solchen Falle ein Steuerelement des Kriterienformulars leer bleibt, führt dies im allgemeinen dazu, daß das Abfrageergebnis leer ist, also keinen einzigen Datensatz enthält. Dies hat den folgenden Grund: Wenn Sie in einer Abfrage für ein Feld ein Kriterium angeben, dann gibt die Abfrage unabhängig vom konkreten Wert des Kriteriums nur solche Datensätze wieder, für die dieses Feld keinen Nullwert enthält, es sei denn, der Kriterienausdruck enthält die Komponente *Oder Ist Null*, was aber die Allgemeinheit des Ausdrucks beeinträchtigen würde. Das Problem taucht nicht auf, wenn Sie Nullwerte für das betreffende Feld nicht zulassen und statt dessen leere Zeichenfolgen verwenden.

In jedem Falle sollten Sie, falls Sie sich für das Zulassen leerer Zeichenfolgen entscheiden, eine entsprechende Anzeigeformatierung vorsehen, die bei der Wiedergabe zwischen Nullwert und leerer Zeichenfolge unterscheidet, vgl. oben in diesem Punkt. Darüber hinaus bleibt es Ihnen natürlich unbenommen, andere Codierungen für fehlende Werte vorzunehmen, etwa *-99* oder *nv*. Diese Vorgehensweise ist oft in Statistikprogrammen anzutreffen.

Gültigkeitsregeln

Gültigkeitsregel für ein Feld. Der Datentyp eines Feldes stellt bereits eine implizite weitreichende Gültigkeitsregel dar. Wenn Sie beispielsweise für ein Feld den Datentyp *Zahl* mit dem Untertyp *Integer* festgelegt haben, so können in dieses Feld nur Ganzzahlen zwischen -32.768 und 32.767 eingegeben werden, alphanumerische Zeichen z.B. sind dagegen ausgeschlossen. Zusätzlich haben Sie aber die Möglichkeit, für jedes Feld eine den Wertebereich weiter einengende Gültigkeitsregel zu definieren. Wenn Sie beispielsweise für ein Feld vom Datentyp *Währung* negative Beträge und solche über 3000.- DM ausschließen wollen, können Sie für die Eigenschaft *Gültigkeitsregel* dieses Feldes den Ausdruck

Zwischen 0 und 3000

eingeben. Die als Gültigkeitsregel angebbaren Ausdrücke können sehr vielfältig und ggf. sehr komplex sein, worauf an dieser Stelle nicht näher eingegangen wird, vgl. dazu Kap. 39, *Ausdrücke, Funktionen und Operatoren*.

Gültigkeitsregel für einen Datensatz. Der Ausdruck einer Gültigkeitsregel für ein Feld darf sich nicht auf ein anderes Feld beziehen. Es ist daher beispielsweise nicht zulässig, für ein Feld *Input1* einen Gültigkeitsausdruck anzugeben, der den Wert dieses Feldes auf das Zweifache des Wertes von Feld *Input2* begrenzt. Diese Wirkung können Sie jedoch erreichen, wenn Sie eine *Gültigkeitsregel für den Datensatz* formulieren. Sie geben diese als Tabellen- und nicht als Feldeigenschaft an, wozu Sie das Eigenschaftenfenster in der Entwurfsansicht der Tabelle anzeigen lassen müssen.

Anders als für Steuerelemente, für die die Eigenschaft *Gültigkeitsregel* auf jeden gültigen Ausdruck eingestellt werden kann, darf der Ausdruck bei Feld- und Datensatz-Gültigkeitsregeln keine benutzerdefinierten Funktionen, Domänenaggregat- oder Aggregatfunktionen, die Funktionen *AktuellerBenutzer* oder *Auswerten* sowie Verweise auf Formulare, Abfragen oder Tabellen enthalten.

Feld- und Datensatz-Gültigkeitsregeln werden, weil sie Definitionsbestandteile der Tabelle sind, stets überprüft, wenn Daten hinzugefügt oder bearbeitet werden. Dies gilt unabhängig davon, ob das Hinzufügen oder Bearbeiten in der Datenblattansicht der Tabelle, in einem Steuerelement eines Formulars, das an das Feld gebunden ist, in einer Anfügeabfrage, in einer Aktualisierungsabfrage, in Form von VBA-Code – auch aus anderen Applikationen – oder durch Importieren von Daten aus einer anderen Tabelle erfolgt. Auch wenn beispielsweise mit einer DAO-Methode, z.B. *Edit* und *Update*, von Excel aus auf eine Access-Tabelle zugegriffen wird, werden deren Gültigkeitsregeln überprüft. Wenn Sie eine neue Gültigkeitsregel eingegeben oder eine bestehende bearbeitet haben, empfiehlt es sich zu testen, ob die bereits in der Tabelle befindlichen Daten dieser Regel entsprechen. Dazu können Sie in der Titelleiste der Entwurfsansicht der Tabelle mit der rechten Maustaste das Kontextmenü aufschlagen und dann auf *Gültigkeitsregeln testen* klikken.

Gültigkeitsmeldungen. Die von Access bei Verletzung einer Gültigkeitsregel standardmäßig ausgegebene Fehlermeldung ist naturgemäß für den durchschnittlichen Benutzer oft wenig brauchbar, weil sie zwar auf die Ursache des Fehlers hinweist, nicht aber darauf, wie der Benutzer sich zum Vermeiden desselben verhalten sollte. Daher erscheint es stets geboten, zu einer Gültigkeitsregel auch eine entsprechende *Gültigkeitsmeldung* zu formulieren. Da die Gültigkeitsmeldung wie die Gültigkeitsregel zusammen mit der Tabellendefinition gespeichert und von der JetEngine ausgewertet wird, wird sie im Falle einer Verletzung der Gültigkeitsregel auch dann wiedergegeben, wenn die Verletzung durch einen Zugriff über VBA mit einer DAO-Methode von einer anderen Applikation aus erfolgt.

Eingabeformate

Mit einem Eingabeformat können Sie die Eingabe von Werten einschränken, bevor diese auf Datentyp und Gültigkeitsregel überprüft werden. So würde die Eingabe beispielsweise mit dem Format

00000

auf genau fünf Ziffern beschränkt und erzwungen werden. Eingabeformate, die auf Tabellenebene definiert wurden, wirken allerdings nur bei der Eingabe in der Datenblattansicht von Tabellen und Abfragen, nicht jedoch im Formular, wenn dort das Eingabeformat nicht ebenfalls definiert ist. Da die JetEngine das Eingabeformat nicht überprüft, wird dieses auch nicht überprüft, wenn Sie über VBA mit einer DAO- oder ADO-Methode von einer anderen Applikation aus Werte verändern. Aus Gründen der Datenintegrität sollten Sie daher, wo immer dieses möglich ist, statt Eingabeformate oder zusätzlich zu diesen Gültigkeitsregeln formulieren.

Nachschlagefelder

Für Felder der Datentypen *Text* oder *Zahl* können Sie Nachschlagefelder einrichten, mit denen die Eingabe von Werten mittels eines Listen- oder Kombinationsfeldes (Dropdown-Liste) vorgenommen werden kann. Der Wertevorrat, den ein solches Nachschlagefeld in seiner Liste anbietet, kann dabei aus einer Tabelle, einer Abfrage oder einer zusammen mit dem Nachschlagefeld definierten Werteliste stammen. Für ein Kombinationsfeld können Sie dabei auch bestimmen, daß in dem entsprechenden Feld ausschließlich Listeneinträge zulässig sind. In diesem Falle schränkt das Nachschlagefeld den für das Feld zulässigen Wertebereich ein. Wenn Sie dagegen die Eigenschaft *Nur Listeneinträge* für ein Kombinationsfeld auf Nein eingestellt oder ein Listenfeld, das diese Eigenschaft gar nicht anbietet, als Nachschlagefeld gewählt haben, wirken die Listeneinträge nur wie eine Liste von Standardwerten, die überschrieben werden können und daher die Domäne des Feldes nicht einschränken.

Hinsichtlich der Reichweite sind Nachschlagefelder mit Eingabeformaten (vgl. den vorigen Punkt) vergleichbar: Obwohl sie automatisch auch in Abfragen oder neu erzeugten Formularen, die auf der Tabelle basieren, angeboten werden, werden die mit ihnen festgelegten Beschränkungen nicht von der JetEngine überprüft. Daher ist es beispielsweise möglich, in einem Formular für das betreffende Feld eine andere Eingabemöglichkeit (z.B. ein Textfeld ohne Gültigkeitsregel) vorzusehen und andere Eingaben vorzunehmen, als im Nachschlagefeld festgelegt sind. Erst recht gilt daher, daß die Festlegungen eines Nachschlagefeldes nicht überprüft werden, wenn die Datenmanipulation mit VBA-Code vorgenommen wird. Nachschlagefelder stellen daher keine zuverlässige Möglichkeit dar, die Domäne eines Tabellenfeldes zu beschränken. Der mit ihnen verbundene Eingabekomfort, insbesondere bei der Eingabe von Fremdschlüsselwerten, ist zwar unverkennbar und in vielen Fällen praktisch unverzichtbar, jedoch müssen für die sichere

Verwaltung der Domäne eines Feldes zusätzliche Vorkehrungen wie das Festlegen entsprechender Datentypen, das Vereinbaren von Gültigkeitsregeln oder die Definition einer referentiellen Integrität getroffen werden.

32.4 Primärschlüssel und Indizes bestimmen

Zu den grundlegenden Prinzipien einer relationalen Datenbank gehört, daß es gleichgültig ist, in welcher Reihenfolge die Datensätze in einer Tabelle angeordnet sind. Daraus folgt, daß die einzelnen Datensätze einer Tabelle keine automatische Ordnungsnummer besitzen. Anders als in anderen nicht konsequent relationalen Datenbankprogrammen existiert daher in Access für Datenzugriffsobjekte keine Eigenschaft RecordNumber o.ä.

Betrifft VBA: Für Datenzugriffsobjekte vom Typ *Recordset* gibt es die Eigenschaft *Bookmark* (Lesezeichen), mit der für jeden Datensatz des Recordset eine von der JetEngine automatisch für das Recordset bei dessen Öffnung generierte eindeutige Kennung abgerufen werden kann, die einer RecordNumber vergleichbar ist. Diese existiert jedoch, wie das Recordset-Objekt selbst, nur temporär und ist kein Bestandteil einer Tabelle. Ein Bookmark dient u.a. dazu, den aktuellen Datensatz des Recordset im selben Recordset wieder zu identifizieren, wenn er nicht mehr der aktuelle ist.

Andererseits ist es notwendig, jeden einzelnen Datensatz eindeutig identifizieren zu können. Dies geschieht in einer relationalen Datenbank mit Hilfe von Schlüsseln.

> Unter einem Schlüssel ist ein Feld oder eine Kombination von Feldern einer Tabelle zu verstehen, dessen (deren) Wert(e) einen Datensatz eindeutig identifizieren.

In einer Tabelle *Personal* könnte sich beispielsweise neben anderen Feldern wie Nachname, Vorname etc. ein Feld mit einer jeden Mitarbeiter eindeutig identifizierenden *Personalnummer* befinden. Dann könnte dieses Feld als Schlüssel zum Identifizieren eines jeden Datensatzes dienen. Eine Tabelle kann natürlich mehr als ein Feld besitzen, das als Schlüssel dienen kann. Wenn es sich z.B. bei den Mitarbeitern, die in der Personaltabelle erfaßt werden, ausschließlich um abhängig Beschäftigte handelt, wird jeder Mitarbeiter eine eindeutige Versicherungsnummer besitzen, die dann neben seiner Personalnummer als Schlüssel dienen kann. Umgekehrt kann es sein, daß eine Tabelle kein Feld besitzt, das allein einen Datensatz identifizieren kann. Dann muß eine Kombination aus mehreren Feldern den Schlüssel bilden. Ein solcher Schlüssel wird *zusammengesetzter Schlüssel* genannt. In einer Personaltabelle könnte dies beispielsweise die Kombination der Felder

> Nachname + Vorname + Geburtsdatum

sein. Die Kombination dieser Felder hat allerdings nur dann Schlüsseleigenschaft, wenn sichergestellt ist, daß im Kontext der betreffenden Tabelle niemals derselbe Nachname mit demselben Vornamen mit demselben Geburts-

datum vorkommen kann. Für eine kleinere Firma wird man dies im allgemeinen annehmen können, für einen Konzern mit 300.000 Mitarbeitern dagegen nicht. Dann müßten ggf. weitere Felder zum Schlüssel hinzugenommen werden. Offensichtlich nimmt die Chance, daß die Feldkombination Schlüsselcharakter hat, zu, je mehr Felder dabei berücksichtigt sind. Andererseits werden Schlüssel um so unhandlicher, je mehr Felder darin berücksichtigt sind. Oft empfiehlt sich daher sogar die Definition eines künstlichen Ersatzschlüssels, vgl. weiter unten Punkt 32.4.2, *Zusammengesetzter Primärschlüssel oder künstlicher Ersatzschlüssel*?.

Die Felder und Kombinationen von Feldern, die Schlüsselcharakter haben, werden oft auch als *Schlüsselkandidaten* bezeichnet. Diese Bezeichnung ist sprachlich allerdings ungenau und etwas verwirrend, weil damit gemeint ist, daß sie Kandidaten für den sogen. *Primärschlüssel* (vgl. den unmittelbar folgenden Punkt) sind. Tatsächlich gibt es aber neben dem Primärschlüssel weitere mögliche Schlüssel, die dann als *Sekundär- oder Alternativschlüssel* bezeichnet werden. In Access würden diese als weitere eindeutige Indizes mit Eingabepflicht in den Feldern definiert werden, vgl. weiter unten.

32.4.1 Bedeutung des Primärschlüssels

Obwohl technisch nicht zwingend erforderlich, sollte für jede Tabelle ein Primärschlüssel definiert werden, weil dadurch die Verwaltung der Datenbank effizienter wird. Dies gilt in besonderem Maße für Tabellen, zwischen denen eine Beziehung definiert werden soll. Andererseits wirkt sich das Nichtvorhandensein eines Primärschlüssels bei kleineren Tabellen ohne Beziehung zu einer anderen Tabelle offenbar nicht merklich effizienzmindernd aus. Der Primärschlüssel ist derjenige Schlüssel, der unter den Schlüsselkandidaten für diesen Zweck ausgewählt wird. Er identifiziert einen Datensatz in einer Tabelle stets eindeutig. Von dieser Regel ist keinerlei Ausnahme zugelassen. Dies hat zur Konsequenz, daß keines der den Primärschlüssel bildenden Felder einen Wert NULL enthalten darf. Die Einhaltung dieser Regel wird von der JetEngine kontrolliert, sobald Sie ein Feld oder eine Feldkombination als Primärschlüssel definiert haben und Sie dann neue Daten in die Schlüsselfelder eingeben oder bestehende bearbeiten. Die Nichtzulässigkeit von Nullwerten in Feldern des Primärschlüssels gilt auch dann, wenn Sie die Feldeigenschaft *Eingabe erforderlich* für die Schlüsselfelder auf Nein eingestellt haben, weil die Erfordernisse des Primärschlüssels Vorrang haben.

32.4.2 Zusammengesetzter Primärschlüssel oder künstlicher Ersatzschlüssel?

Als Primärschlüssel können Sie ein einziges Feld oder eine Kombination von Feldern verwenden. Wenn die Tabelle ein Feld mit eindeutigen Werten wie Personalnummer oder Artikelnummer besitzt, bietet sich dieses Feld für den Primärschlüssel an. Wenn ein Feld mit eindeutigen Werten nicht vorhanden

ist, muß überlegt werden, ob eine Kombination von Feldern, die eindeutig ist (*zusammengesetzter Schlüssel*), oder ein zusätzliches künstliches Feld mit dem Datentyp *AutoWert* (*Ersatzschlüssel*) als Primärschlüssel verwendet wird. Für oder gegen die Verwendung eines künstlichen Ersatzschlüssels – auch Surrogatschlüssel genannt – sprechen die folgenden Überlegungen:

Der Primärschlüssel einer Mastertabelle wird regelmäßig zum *Fremdschlüssel* in der Detailtabelle, wenn zwischen beiden eine Beziehung definiert wird. In Abfragen muß dann sehr häufig ein Join zwischen den beiden Tabellen formuliert werden. Ein derartiger Join wird um so komplizierter, je mehr Felder in die Join-Bedingung aufgenommen werden. Die Größe des zusammengesetzten Schlüssels erhöht somit die Anzahl der Join-Bedingungen. Dies sei an dem folgenden Beispiel demonstriert:

Bild 32.4: Beziehungen mit zusammengesetztem Schlüssel (links) und mit Ersatzschlüssel (rechts)

In Bild 32.4 wird je eine Beziehung der Tabellen *Kunde* und *Auftrag* bzw. *Kunde_1* und *Auftrag_1* wiedergegeben. Die beiden Tabellenpaare haben im Prinzip denselben Informationsgehalt. Die erste Beziehung erfolgt über den zusammengesetzten Primärschlüssel mit den Feldern *Firma*, *Straße*, *PLZ* und *Ort*, der in der Detailtabelle *Auftrag* als Fremdschlüssel fungiert. Die zweite Beziehung erfolgt mit Hilfe des künstlichen Ersatzschlüssels *KundeID*, der in der Tabelle *Kunde_1* Primärschlüssel und in der Tabelle *Auftrag_1* Fremdschlüssel ist. Wenn Kunden- und Auftragsinformationen mit Hilfe einer Abfrage wiedergegeben werden sollen, lautet die SQL-Formulierung der jeweiligen Abfrage folgendermaßen:

Beziehung 1

```
SELECT Auftrag.*, Kunde.* FROM Kunde INNER JOIN Auftrag ON
(Kunde.Ort = Auftrag.Ort) AND (Kunde.PLZ = Auftrag.PLZ) AND
(Kunde.Straße = Auftrag.Straße) AND (Kunde.Firma = Auftrag.Firma);
```

Beziehung 2

```
SELECT Auftrag_1.*, Kunde_1.* FROM Kunde_1 INNER JOIN Auftrag_1 ON
Kunde_1.KundeID = Auftrag_1.KundeID;
```

Die erste Abfrage benötigt – entsprechend den vier Feldern des zusammengesetzten Primärschlüssels – vier Join-Bedingungen, die zweite dagegen nur eine. In tiefer gestaffelten Abfragen kann sich dieses Problem noch sehr viel weiter verschärfen: Der Fremdschlüssel wird oft auch Bestandteil des Primärschlüssels der Detailtabelle, wenn in der Detailtabelle kein einzelnes Feld als Primärschlüssel dienen kann, so daß der Primärschlüssel bei gestaffelten Beziehungen immer größer wird und entsprechend die Abfragen immer komplizierter.

Dieser Sachverhalt scheint für Beziehungen unbedingt die Verwendung von künstlichen Ersatzschlüsseln nahezulegen. Allerdings ist zu bedenken, daß dies nur gilt, wenn und soweit der Benutzer Abfragen in Form von SQL-Statements handhaben und insbesondere manipulieren muß: Wenn Sie, wie es auf vielen Hosts heute noch üblich ist, SQL-Abfragen autonom formulieren müssen, kann die Kompliziertheit der Join-Bedingungen ein starkes Argument für die Verwendung von künstlichen Ersatzschlüsseln statt zusammengesetzter Schlüssel sein. In Access steht Ihnen andererseits für die Formulierung von Abfragen der Abfrage-Entwurfsbereich zur Verfügung. Wenn und solange Sie damit arbeiten, kann Ihnen die Anzahl der Join-Bedingungen gleichgültig sein, weil Sie den entsprechenden SQL-Code gar nicht wahrnehmen. Allerdings kommt es bei professionellem Arbeiten häufig vor, daß man eine Abfrage zwar mit Hilfe des Abfrage-Entwurfsbereichs formuliert, das entsprechende SQL-Statement jedoch übernimmt und beispielsweise in VBA-Code weiterverarbeitet und ggf. manipuliert. Bei solcher Arbeitsweise bleibt ein Teil des genannten Argumentes erhalten, weil es zwar nicht kompliziert ist, die SQL-Anweisung zu formulieren, jedoch ist sie unübersichtlicher als im Falle der Verwendung künstlicher Ersatzschlüssel, so daß sie tendenziell auch schwieriger zu manipulieren sein dürfte.

Dieser Folgerung, die gegen die Verwendung eines zusammengesetzten Schlüssels in Beziehungen spricht, steht allerdings die folgende Überlegung gegenüber: Wenn der zusammengesetzte Schlüssel Fremdschlüssel der Detailtabelle ist, enthält diese mehr Informationen. Dies kann bei gewissen Fragestellungen dazu führen, daß eine Abfrage ohne einen Join auskommt, den diese bei der alternativen Verwendung eines Ersatzschlüssels erfordern würde. Nehmen Sie beispielsweise die Beziehung der beiden Tabellen *Kunde* und *Auftrag* aus Bild 32.4 oben. Wenn für diese Beziehung nur die Werte für die Felder *Gegenstand*, *Firma*, *Straße*, *PLZ* und *Ort* ermittelt werden müßten, so könnte dies mit einer Abfrage ohne Join geschehen, weil sämtliche Informationen in der Detailtabelle *Auftrag* enthalten sind. Wenn daher derartige Fragestellungen oft zu erwarten sind, empfiehlt sich auch in Beziehungen eher ein zusammengesetzter Schlüssel gegenüber einem Ersatzschlüssel.

Ein künstlicher Ersatzschlüssel hat gegenüber einem zusammengesetzten Schlüssel stets den Nachteil, daß er sprachlich gehaltlos ist, so daß Sie in seinem Wert im allgemeinen keine Bedeutung erkennen können. Daher sollten Sie einen Ersatzschlüssel nur dann wählen, wenn sich aus der Verwendung eines zusammengesetzten Schlüssels als Primärschlüssel Nachteile ergeben.

Nachteile zeigen sich jedoch nur dann, wenn der zusammengesetzte Schlüssel zum Fremdschlüssel in der Detailtabelle einer Beziehung wird. Wenn es sich daher um eine Tabelle handelt, die nicht als Mastertabelle in einer Beziehung stehen soll, wenn also keine andere Tabelle davon abhängt, empfiehlt sich ein (sprachlich gehaltvoller) zusammengesetzter Primärschlüssel gegenüber einem künstlichen Ersatzschlüssel.

32.4.3 Alternativschlüssel

Einzelne Felder oder Kombinationen von Feldern, die eindeutige und damit einen Datensatz identifizierende Werte enthalten, so daß sie als Primärschlüssel dienen *könnten*, werden als *Alternativschlüssel* (engl. alternate key) oder als *Sekundärschlüssel* bezeichnet. Praktisch machen Sie ein Feld oder eine Kombinationen von Feldern zum Sekundärschlüssel, indem Sie einen eindeutigen Index mit Eingabepflicht für seine Felder erstellen.

Vom datentheoretischen Standpunkt aus besteht eigentlich überhaupt kein Unterschied zwischen einem Primärschlüssel und einem Sekundärschlüssel. Oft wird gesagt, der Unterschied zwischen beiden Schlüsselarten bestehe darin, daß »Sekundärschlüssel nicht von Fremdschlüsseln anderer Tabellen referenziert werden« können[3], während dies für Primärschlüssel zutreffe, so daß letztere in einer Tabelle zwingend erforderlich seien, um diese zur Mastertabelle in einer Beziehung zur Detailtabelle machen zu können. Auch Access selbst teilt diese scheinbare Regel mit, wenn Sie eine Tabelle erstmalig speichern, ohne einen Primärschlüssel definiert zu haben, vgl. Bild 32.5.

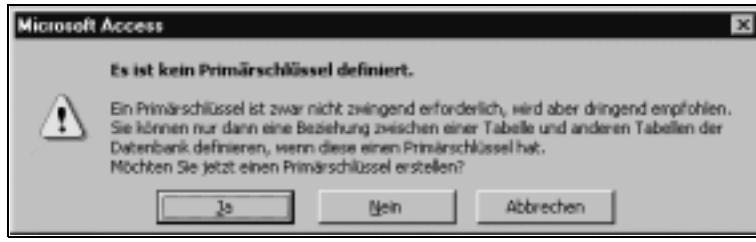


Bild 32.5: Hinweis von Access, wenn Sie eine Tabelle erstmalig speichern, ohne einen Primärschlüssel definiert zu haben.

Tatsächlich können Sie einen Sekundärschlüssel einer Mastertabelle von einem Fremdschlüssel einer Detailtabelle durchaus referenzieren lassen, indem Sie eine Beziehung mit referentieller Integrität zwischen beiden Tabellen erstellen und sich dabei auf den Sekundärschlüssel der Mastertabelle beziehen. Die JetEngine von Access verwaltet diese Beziehung mit referentieller Integrität genauso zuverlässig, als wenn sie statt mit dem Sekundärschlüssel mit dem Primärschlüssel definiert worden wäre. Es ist auch mög-

3. [Sommer], S. 54

lich, eine Tabelle mehrmals als Mastertabelle in verschiedenen Beziehungen zu definieren, wobei in der einen Beziehung der Primärschlüssel und in den anderen die Sekundärschlüssel der Mastertabelle vom jeweiligen Fremdschlüssel der Detailtabelle referenziert wird. Diese Leistung ist im übrigen nicht auf die JetEngine beschränkt, sondern beispielsweise auch der SQL-Server von Microsoft erlaubt ebenfalls, den Sekundärschlüssel einer Mastertabelle vom Fremdschlüssel einer Detailtabelle zu referenzieren.

Vom logischen Standpunkt gibt es kein Kriterium, das dem Primärschlüssel gegenüber einem Sekundärschlüssel den Vorzug in einer Beziehung mit referentieller Integrität gäbe, sie sind insoweit austauschbar. Inwieweit die Verwendung des einen gegenüber dem anderen einen Vorteil in der Performance bietet, bliebe zu testen.

32.4.4 Vor- und Nachteile weiterer Indizes

Vorteile. Der grundlegende Zweck eines Index liegt darin, Such- und Sortiervorgänge für die Felder des Index zu beschleunigen und/oder Eindeutigkeit für diese Felder zu gewährleisten. Für den Primärschlüssel einer Tabelle erzeugt Access stets automatisch einen eindeutigen Index, für andere Felder können oder müssen Sie dies mit Hilfe des Fensters *Indizes*, das Sie in der Entwurfsansicht einer Tabelle einblenden können, selbst erledigen. Die Beschleunigung von Such- und Sortiervorgängen ist gravierend. Sie wirkt sich um so stärker aus, je mehr Datensätze eine Tabelle oder Abfrage enthält. Umgekehrt ist die Notwendigkeit, Indizes zu definieren, für Tabellen mit nur wenigen Datensätzen gering. Beschleunigt werden auch Abfragen mit Joins, wenn für den Fremdschlüssel der Detailtabelle ein Index definiert ist.

Nachteile. Indizes wirken sich nachteilig auf die Performance aus, wenn Datensätze gelöscht, angefügt oder in Feldern des Index aktualisiert werden. In diesen Fällen müssen die betroffenen Indizes vom RDBMS stets sofort aktualisiert werden, was zeitaufwendig sein kann. Dies kann sich besonders nachteilig in einer Mehrbenutzer-Umgebung auswirken: Access sperrt die zu einem Datensatz gehörige ganze Page, während der Datensatz bearbeitet wird. Entsprechend werden die zum Index gehörigen Pages gesperrt, während dieser aktualisiert wird, so daß andere Benutzer in der Zeit der Aktualisierung ggf. mehrere Datensätze nicht bearbeiten können.

Wegen der Gleichzeitigkeit von Vor-und Nachteilen stellt die Aufgabe, die richtigen Indizes auszuwählen, ein Optimierungsproblem dar. Lassen Sie sich bei der Auswahl der Indizes von folgenden Regeln leiten[4]:

- Minimieren Sie die Anzahl von Indizes für Tabellen, mit denen vorwiegend Transaktionen (Löschen, Einfügen, Aktualisieren) vorgenommen werden. Beachten Sie dabei, daß hierunter auch Tabellen fallen, deren

4. [Jennings], S. 844

Datensätze mit Lösch- oder Anfügeabfragen verändert werden. Bei massenhaftem Löschen oder Anfügen von Datensätzen fällt die für die Aktualisierung der Indizes erforderliche Zeit besonders ins Gewicht.

- Fügen Sie Indizes in Tabellen ein, die viele Datensätze enthalten und oft in Suchvorgängen verwendet werden, beispielsweise in häufigen Ad hoc-Abfragen zur Unterstützung von geschäftlichen Entscheidungen. Bedenken Sie dabei besonders die Felder, für die (mutmaßlich) oft Kriterien zum Suchen oder Selektieren angegeben werden.
- Fügen Sie Indizes für die Fremdschlüssel von Detailtabellen ein, wenn diese in Joins mit den Mastertabellen verwendet werden. Falls Sie dabei für die Schlüsselfelder ein Kriterium angeben müssen, sollten Sie dieses allerdings für die Felder des Primärschlüssels der Mastertabelle tun, weil dies den Suchvorgang effektiver macht.

32.5 Das Redundanzproblem

In den vorangehenden Ausführungen dieses Kapitels spielte die Frage der Vermeidung von Redundanz bereits öfter eine Rolle, ohne daß wir bisher systematisch darauf eingegangen wären. Das soll nunmehr geschehen. Die Vermeidung von Redundanz hat für manche Autoren, die sich theoretisch mit Datenbanken beschäftigen, fast schon dogmatischen Charakter, und manchmal bekommt man den Eindruck, daß Redundanzvermeidung ein Axiom sei, von dem aus sich jedes Detail des Datenbankdesign deduzieren lasse. Dies ist sicherlich nicht zutreffend, und dogmatischer Umgang mit Leitsätzen sollte stets zur Vorsicht gemahnen. Andererseits kommt dem Redundanzproblem sicher eine zentrale Bedeutung für das Datenbankdesign zu, was in den folgenden Überlegungen deutlich werden soll.

32.5.1 Was bedeutet Redundanz?

Wörtlich bedeutet Redundanz *Überfluß*: Informationen sind dann redundant, wenn sie überflüssig sind, d.h., wenn sie ohne Informationsverlust für das System fortgelassen werden können. In diesem Sinne erscheint es fast natürlich, eine Datenbank ohne Redundanz aufzubauen, weil es sicherlich eine allgemeine Lebensregel ist, nach Möglichkeit überflüssige Dinge zu vermeiden. Zwar binden auch überflüssige Dinge im allgemeinen Ressourcen – redundante Informationen binden und kosten damit Speicherplatz. Andererseits scheint es ohne genaueres Hinsehen aber auch nicht weiter schlimm zu sein, wenn man redundant arbeitet, letztlich könnte es die Sache jedes einzelnen sein, ob er sich sozusagen den Luxus des Überflusses leistet oder nicht. Gerade das Speicherplatzargument hat heute keine große Bedeutung mehr, weil die entsprechenden Ressourcen sehr billig geworden sind. Tatsächlich sind mit dem Verarbeiten redundanter Informationen in einer Datenbank gravierende Nachteile verbunden, die es geboten erscheinen lassen, Redundanz zu vermeiden – ob dogmatisch oder mit Augenmaß, darauf wird weiter unten eingegangen, nachdem die einzelnen Nachteile dargestellt wurden.

32.5.2 Nachteile von Redundanz

Außer zum Verbrauch von unnötig viel Speicherplatz kann Redundanz vor allem zu Widersprüchen in den Daten führen, die auf Dauer den Gebrauch der Datenbank einschränken oder gar unmöglich machen. Diese Widersprüche werden in der Literatur im allgemeinen als *Anomalien* bezeichnet. Zu unterscheiden sind *Änderungsanomalien*, *Einfügeanomalien* und *Löschanomalien*.

Änderungsanomalien

Änderungsanomalien treten auf, wenn die Änderung eines Datensatzes unnötig kompliziert wird, so daß z.B. die Änderung des Wertes eines oder mehrerer Felder (z.B. Adressenänderung) nicht nur an einem Datensatz, sondern an vielen Datensätzen vorgenommen werden muß, obwohl es sich eigentlich nur um die Änderung einer singulären Information handelt. Dabei können Widersprüche auftreten, die oft nur schwer zu entdecken oder nachträglich zu beseitigen sind. Dies sei an folgendem Beispiel demonstriert:

Nachname	Straße	Ort	Ferienhaus	Wohnfläche
Webern	Goethestr. 1	Amberg	Buntspecht	49
Behlmer	Werkstr. 7	Schwerin	Buntspecht	49
Webern	Goethestr. 1	Amberg	Amsel	43
Webern	Goethestr. 1	Amberg	Rotkehlchen	50
Behlmer	Werkstr. 7	Schwerin	Stieglitz	40

Tabelle 32.1: Vor der Adressenänderung

Nachname	Straße	Ort	Ferienhaus	Wohnfläche
Webern	Lyoner Str. 32	Frankfurt	Buntspecht	49
Behlmer	Werkstr. 7	Schwerin	Buntspecht	49
Webern	Goethestr. 1	Amberg	Amsel	43
Webern	Goethestr. 1	Amberg	Rotkehlchen	50
Behlmer	Werkstr. 7	Schwerin	Stieglitz	40

Tabelle 32.2: Nach der Adressenänderung

Tabelle 32.1 gibt einen Ausschnitt aus den redundant gehaltenen Daten einer Ferienhausvermietung wieder. Redundant sind die Daten u.a. deshalb, weil die Angaben zu den Mietern, die hier jeweils mehrere Ferienhäuser gemietet haben, sich überflüssigerweise wiederholen. Es sei unterstellt, daß sich die Adresse des Mieters *Webern* geändert habe, seine neue Anschrift sei *Lyoner Str. 32*, *Frankfurt*. Um diese Adressenänderung zu veranlassen, müßten die drei Datensätze, in denen der Mieter *Webern* vorkommt, geändert werden. Wenn dies korrekt geschieht, besteht der einzige Nachteil der red-

undanten Datenhaltung darin, daß die Änderung auf unnötig komplizierte Weise (nämlich an vielen Datensätzen statt nur an einem) vorzunehmen ist. Wenn jedoch, wie dies in Tabelle 32.2 wiedergegeben ist, die Änderung nicht an allen erforderlichen Datensätzen vorgenommen wird, sondern nur an einem oder einigen, entsteht ein Widerspruch in den Daten: Der Mieter *Webern* wird dann mit zwei Anschriften geführt. Man könnte einwenden, daß ein derartiger Widerspruch wie in diesem Beispiel angenommen, unwahrscheinlich sei, weil – zumindest in Access – das Instrument der Änderungsabfrage zur Verfügung steht, mit dem Änderungen sicher und vollständig an allen Datensätzen der Abfrage vorgenommen werden können. Für das sehr kleine vorliegende Beispiel mag dieser Einwand gelten. Er setzt jedoch im allgemeinen voraus, daß Sie alle Stellen (praktisch: alle Felder aller Tabellen), an denen die Anschrift des Mieters *Weber* in der Datenbank vorkommt, sicher kennen. Dies ist jedoch bei redundanter Datenhaltung kaum sicherzustellen, weil Sie im allgemeinen keinen systematischen Überblick über die verschiedenen Stellen, an denen dieselbe Information wiederholt gespeichert ist, haben. Dieser Einwand gilt verstärkt, wenn Widersprüche durch Änderungen in der Vergangenheit bereits vorhanden sind. Nehmen wir für das vorige Beispiel an, daß die Adressenänderung des Mieters *Webern* zwar an allen drei erforderlichen Datensätzen vorgenommen worden ist, daß dabei aber *Lyoner Str. 32* einmal in der Form *Lyoner Str 32* vorgenommen worden sei. Da es nicht unwahrscheinlich ist, daß der Name *Webern* in der Datenbank mehreren verschiedenen Mietern zugehörig ist, müssen Sie im allgemeinen mit einer Abfrage nach einer Kombination der Felder *Nachname*, *Straße* und *Ort* für die bestehende Adresse suchen lassen, um alle Datensätze dieses bestimmten *Webern* zu erhalten. Wenn Sie dabei als Kriterium für *Straße* den String *Lyoner Str. 32* angeben, wird der Datensatz mit dem Eintrag *Lyoner Str 32* für *Straße* nicht ausgewählt, und etwaige Änderungen, die Sie auf Basis der ausgewählten Datensätze für den Mieter *Webern* vornehmen wollen, erfolgen dann nicht an allen Datensätzen, so daß die Widersprüche sich insoweit noch verstärken.

Einfügeanomalien

Einfügeanomalien können bei redundanter Datenhaltung entstehen, wenn Sie einen neuen Datensatz in eine Tabelle einfügen. Zum einen können dieselben Anomalien auftreten wie beim Ändern eines Datensatzes (*Änderungsanomalien*), vgl. den vorangehenden Punkt. Wenn Sie beispielsweise in die obige Tabelle 32.1 einen neuen Mietvertrag, d.h. einen neuen Datensatz, für einen bereits in früheren Verträgen enthaltenen Mieter aufnehmen wollen, müssen Sie denselben Namen mit derselben Anschrift, die bereits in anderen Datensätzen vorhanden sind, erneut eingeben. Dabei können dieselben Widersprüche entstehen, wie dies bereits oben für das Ändern von Daten gezeigt wurde. Diese Art von Anomalie ist daher eigentlich keine besondere Einfügeanomalie, obwohl in der Literatur oft als solche ausgewiesen.

Eine spezifische Einfügeanomalie zeigt dagegen der folgende Fall: Es sei angenommen, für das obige Beispiel der Ferienhausvermietung sei ein weiteres Haus in der Siedlung entstanden, das nunmehr mit seinen Informationen (hier: Name des Ferienhauses und Wohnfläche) in die Datenbank aufgenommen werden soll. Im Beispiel der obigen Tabelle 32.1 könnte ein neues Haus eigentlich nur zusammen mit einem neuen Mietvertrag aufgenommen werden, d.h. zusammen mit den Angaben zum Mieter. Wenn gleichwohl das neue Ferienhaus in die Datenbank eingegeben werden soll, ohne daß gleichzeitig ein entsprechender neuer Mietvertrag für dieses Ferienhaus vorliegt, müßte die *Krücke* gewählt werden, einen neuen Datensatz einzufügen, dessen Felder *Ferienhaus* und *Wohnfläche* mit den Angaben des neuen Ferienhauses versorgt würden, dessen Felder *Nachname*, *Straße* und *Ort* dagegen frei blieben (d.h. jeweils den Wert NULL enthielten). Dieser Weg wäre allerdings gänzlich ausgeschlossen, wenn die Felder *Nachname*, *Straße* und *Ort* Teil des Primärschlüssels (oder eines anderen eindeutigen Schlüssels) wären, weil sie dann keinen Nullwert enthalten dürften.

Löschanomalien

Löschanomalien können bei redundanter Datenhaltung auftreten, wenn Sie aus einer Tabelle einen oder mehrere Datensätze löschen. Nehmen wir beispielsweise für die obige Tabelle 32.1 an, die Datensätze

Behlmer	Werkstr. 7	Schwerin	Buntspecht	49
Behlmer	Werkstr. 7	Schwerin	Stieglitz	40

für den Mieter *Behlmer* sollen aus irgendeinem Grunde gelöscht werden. Möglicherweise war das Haus *Stieglitz* gerade zusammen mit diesem Mieter neu in die Datenbank aufgenommen worden und noch in keinem weiteren Datensatz abgelegt. Dann würde die zum Ferienhaus gehörige Information der Wohnfläche (und im realistischen Falle sicherlich weitere zum Ferienhaus gehörende Informationen) zusammen mit den Mieterdaten gelöscht werden und wären aus der Datenbank verschwunden.

Im Ergebnis zeigt sich, was bereits als Einfügeanomalie deutlich wurde, daß eine derartig redundante Form der Datenhaltung wie im angeführten Beispiel von Tabelle 32.1 eine *unabhängige* Verwaltung der Mieterinformationen einerseits und der Ferienhausinformationen andererseits nicht zuläßt. Das Problem ist dadurch zu lösen, daß die Informationen für die Mieter und die Ferienhäuser in zwei verschiedenen Tabellen abgelegt und diese dann ggf. durch eine Beziehung miteinander verbunden werden. Damit und generell mit dem Problem der Beseitigung von Redundanz beschäftigt sich der folgende Punkt.

32.5.3 Redundanz beseitigen

Die vorangehenden Punkte haben gezeigt, daß redundante Datenhaltung zu Widersprüchen in den Daten führen kann. Daher ist es geboten, die Tabellen einer Datenbank so zu konzipieren, daß sie keine Redundanz enthalten. Dieser Punkt beschäftigt sich mit der Frage, auf welche Weise dieses Ziel zu erreichen ist. In der Datenbankliteratur wird – unter Berufung auf *Codd* als den Schöpfer des Relationenmodells – als das alles überragende Verfahren der Weg der *Normalisierung* beschrieben und empfohlen. Als Ziel wird dabei im allgemeinen akzeptiert, wenn sich Tabellen in der sogen. *3. Normalform* befinden. Tabellen, die nicht mindestens dieser Form gerecht werden, werden als unnormalisiert betrachtet und damit als für eine relationale Datenbank ungeeignet abgelehnt. Wegen der Prädominanz der Normalformenlehre in der Literatur scheint es für fortgeschrittene Datenbankdesigner unerläßlich zu erfahren, was unter der *Normalisierung* von Tabellen zu verstehen ist, weil sonst das Verständnis anderer Literatur schwierig werden könnte. Daher werden die drei Normalformen im folgenden ebenfalls vorgestellt und erklärt. Von vornherein sei jedoch auf ein mögliches Mißverständnis hingewiesen: Die Normalformenlehre stellt eine Systematik dar, mit deren Hilfe Sie im Zweifelsfall prüfen können, ob Tabellen einer Datenbank Redundanz enthalten, und welcher Art diese Redundanz ggf. ist. Sie gibt nicht unbedingt den praktisch empfehlenswerten Weg an, nach dem Sie jede einzelne Tabelle einer Datenbank konzipieren sollten. Im allgemeinen Fall wird es in der praktischen Arbeit ausreichen, sich des Redundanzproblems bewußt zu sein und die Tabellen mehr oder minder intuitiv ohne Redundanz zu entwerfen. Schon gar nicht garantiert ein der Normalformenlehre genügendes System von Tabellen, daß die entsprechende Datenbank die Sachverhalte der realen Welt, die sie abbilden soll, angemessen wiedergibt. Dieses Problem ist ungleich schwieriger zu lösen als das der Redundanzvermeidung, weil es durch keinerlei formale Vorschrift anzugehen ist.

Zur Demonstration der Normalformen dient als Beispiel eine fiktive Ferienhausvermietung: In einer Datenbank sollen die Mietverträge von Mietern verwaltet werden. Ein Mieter kann im Laufe der Zeit durchaus mehrere Mietverträge haben, von denen jeder mit seinen Daten in der Datenbank festgehalten wird. Die in der Feriensiedlung zur Vermietung verfügbaren Häuser haben individuelle und eindeutige Namen, z.B. *Buntspecht* oder *Rotkehlchen*. Jedes Haus ist einer Kategorie zugeordnet, welche den Tages- und Wochenpreis bestimmt. Für die folgenden Beispieldaten werden vom Mieter nur der *Nachname* und eine ihn eindeutig identifizierende *MieterNr* angeführt. Weitere Angaben wie der Vorname und die Anschrift bleiben aus Platzgründen unberücksichtigt, so daß die Tabellen im Druck nicht umbrochen werden müssen und damit übersichtlich bleiben.

Wiederholungsgruppen beseitigen (1. Normalform)

Mieter-Nr	Nach-name	Ferienhaus	Miet-beginn	Miet-ende	Anzahl-Zimmer	Wohn-fläche	Kate-gorie	Tages-preis	Wochen-preis
(1)	Webern	Buntspecht	04.05.96	24.05.96	(3)	49	Standard	130	700
		Amsel	03.05.96	23.05.96	(2)	43	Standard	130	700
		Rotkehlchen	21.12.96	03.01.97	(3)	50	Standard	130	700
(3)	Note-boom	Pirol	03.08.97	30.08.97	(3)	67	Luxus	200	1.000
		Sperling	16.08.97	29.08.97	(2)	35	Einfach	90	500
(4)	Behlmer	Buntspecht	15.06.96	05.07.96	(3)	49	Standard	130	700
		Stieglitz	07.06.97	27.06.97	(2)	40	Einfach	90	500
(5)	Preisel-beer	Nachtigall	20.07.96	02.08.96	(4)	75	Luxus	200	1.000
		Pirol	21.12.96	03.01.97	(3)	67	Luxus	200	1.000
(6)	Hase	Lerche	21.12.96	03.01.97	(4)	75	Luxus	200	1.000
		Lerche	25.05.96	14.06.96	(4)	75	Luxus	200	1.000

Tabelle 32.3: Unnormalisierte Daten

Tabelle 32.3 enthält für alle Felder bis auf *MieterNr* und *Nachname* sogen. *Wiederholungsgruppen* (ein für mein Sprachverständnis ungünstiger weil mißverständlicher Terminus, denn gemeint sind *Mehrfach*einträge, die allenfalls in Ausnahmefällen Wiederholungen darstellen): Für den Mieter *Webern* beispielsweise enthält das *Ferienhaus* die drei Einträge *Buntspecht*, *Amsel* und *Rotkehlchen*. Wenn sich in Feldern derartige Wiederholungsgruppen von Werten befinden, liegt ein Verstoß gegen die Forderung vor, daß Werte stets nur in *atomarer* Form in Tabellen festgehalten werden sollen. Wenn diese Forderung erfüllt ist, befindet sich eine Tabelle in der sogen. *1. Normalform*. Umgekehrt: Sämtliche Felder einer Tabelle, die sich in der 1. Normalform befindet, enthalten nur atomare Werte.

MieterNr	Nach-name	Ferienhaus	Miet-beginn	Miet-ende	Anzahl-Zimmer	Wohn-fläche	Kategorie	Tages-preis	Wochen-preis
(1)	Webern	Buntspecht	04.05.96	24.05.96	(3)	49	Standard	130	700
(1)	Webern	Amsel	03.05.96	23.05.96	(2)	43	Standard	130	700
(1)	Webern	Rotkehlchen	21.12.96	03.01.97	(3)	50	Standard	130	700
(3)	Note-boom	Pirol	03.08.97	30.08.97	(3)	67	Luxus	200	1.000
(3)	Note-boom	Sperling	16.08.97	29.08.97	(2)	35	Einfach	90	500
(4)	Behlmer	Buntspecht	15.06.96	05.07.96	(3)	49	Standard	130	700
(4)	Behlmer	Stieglitz	07.06.97	27.06.97	(2)	40	Einfach	90	500

Tabelle 32.4: Sämtliche Daten in einer Tabelle in der 1. Normalform

MieterNr	Nachname	Ferienhaus	Mietbeginn	Mietende	Anzahl-Zimmer	Wohnfläche	Kategorie	Tagespreis	Wochenpreis
(5)	Preiselbeer	Nachtigall	20.07.96	02.08.96	(4)	75	Luxus	200	1.000
(5)	Preiselbeer	Pirol	21.12.96	03.01.97	(3)	67	Luxus	200	1.000
(6)	Hase	Lerche	21.12.96	03.01.97	(4)	75	Luxus	200	1.000
(6)	Hase	Lerche	25.05.96	14.06.96	(4)	75	Luxus	200	1.000

Tabelle 32.4: Sämtliche Daten in einer Tabelle in der 1. Normalform

Formal beseitigen Sie die Wiederholungsgruppen der einzelnen Zeilen (= Datensätze), indem Sie für jeden Wert jeder Wiederholungsgruppe eine eigene Zeile vorsehen. Dies ist in Tabelle 32.4 geschehen. Tabelle 32.4 befindet sich in der 1. Normalform, weil sich in allen Feldern aller Zeilen (aller Datensätze) nur atomare Werte befinden.

Für sich genommen hat der Übergang vom völlig unnormalisierten Zustand der Tabelle 32.3 zum Zustand der 1. Normalform der Tabelle 32.4 statt zu weniger zu mehr Redundanz geführt: Die Angaben zu *MieterNr* und *Nachname* enthalten in der 1. Normalform mehr Redundanz als vorher, weil sie jetzt mehrfach vorkommen, vorher dagegen für jeden Mieter nur einmal. Schon aus diesem Grunde kann man es im Prozeß der Normalisierung nicht bei der 1. Normalform belassen. Ihr Vorteil gegenüber völlig unnormalisierten Daten ist für sich genommen zweifelhaft. Der Vorteil der 1. Normalform liegt vielmehr im wesentlichen darin, daß sie den Übergang zur 2. Normalform erlaubt. Stahlknecht äußert sich zur 1. Normalform folgendermaßen: »Nach der strengen Normalformenlehre ist der Übergang von der unnormalisierten Form zur 1. Normalform ... immer so vorzunehmen, daß formal alle Zeilen mit mehrfach besetzten Spalten in mehrere Zeilen aufgelöst werden. Damit werden jedoch Redundanzen erzeugt, die im Grunde dem Prinzip des Normalisierungsprozesses, Redundanzen zu verringern, widersprechen. Eine mehr pragmatische Vorgehensweise besteht darin, unter weitgehender Vermeidung zusätzlicher Redundanzen bereits beim Übergang zur 1. Normalform eine Aufteilung der unnormalisierten Relation in mehrere, von vornherein erkennbare Relationen vorzunehmen."[5] Dieser m.E. zutreffende Hinweis wird durch den folgenden noch verstärkt: Die Auflösung der Zeilen mit Wiederholungsgruppen in jeweils mehrere neue Zeilen ist nur möglich für Tabellen, die bereits Daten enthalten. Anders als im obigen fiktiven Lehrbuchbeispiel der Tabelle 32.3 bzw. Tabelle 32.4 haben Sie es beim Konzipieren einer Datenbank jedoch im allgemeinen zunächst lediglich mit Tabellen im Entwurfsstadium zu tun. Diese Tabellen, die noch keine Daten enthalten, müssen so konzipiert werden, daß sie mindestens der 1. Normal-

5. [Stahlknecht], S. 208

form genügen, d.h. *später* nur atomare Werte enthalten *werden*. Bei der Prüfung dieser Regel hilft es für Tabellen ohne Daten überhaupt nicht weiter, gedachte Zeilen mit gedachten Wiederholungsgruppen in gedachte neue Zeilen aufzulösen. Entsprechend der gerade wiedergegebenen Anregung von Stahlknecht erscheint es dagegen viel praktischer zu prüfen, für welche Felder Mehrfachangaben (um das Wort Wiederholungsgruppen zu vermeiden, tatsächlich sind weitere Angaben in der Regel keine Wiederholungen) erforderlich werden, ohne redundant zu sein, und für welche Felder eine Angabe nur einmal erforderlich ist. Dann sollten die verschiedenartigen Felder verschiedenen Tabellen zugeordnet werden.

MieterNr	Nachname
(1)	Webern
(3)	Noteboom
(4)	Behlmer
(5)	Preiselbeer
(6)	Hase

Tabelle 32.5: Tabelle in 1. Normalform

Mieter-Nr	Ferienhaus	Mietbeginn	Mietende	Anzahl-Zimmer	Wohn-fläche	Kategorie	Tages-preis	Wochen-preis
(1)	Buntspecht	04.05.96	24.05.96	(3)	49	Standard	130	700
(1)	Amsel	03.05.96	23.05.96	(2)	43	Standard	130	700
(1)	Rotkehlchen	21.12.96	03.01.97	(3)	50	Standard	130	700
(3)	Pirol	03.08.97	30.08.97	(3)	67	Luxus	200	1.000
(3)	Sperling	16.08.97	29.08.97	(2)	35	Einfach	90	500
(4)	Buntspecht	15.06.96	05.07.96	(3)	49	Standard	130	700
(4)	Stieglitz	07.06.97	27.06.97	(2)	40	Einfach	90	500
(5)	Nachtigall	20.07.96	02.08.96	(4)	75	Luxus	200	1.000
(5)	Pirol	21.12.96	03.01.97	(3)	67	Luxus	200	1.000
(6)	Lerche	21.12.96	03.01.97	(4)	75	Luxus	200	1.000
(6)	Lerche	25.05.96	14.06.96	(4)	75	Luxus	200	1.000

Tabelle 32.6: Tabelle in 1. Normalform

Dies ist in Tabelle 32.5 und Tabelle 32.6 geschehen: Die unnormalisierten Daten aus obiger Tabelle 32.3 wurden so auf zwei Tabellen verteilt, daß alle Angaben in atomarer Form vorliegen und daher keine Mehrfachangaben für dasselbe Feld eines Datensatzes vorkommen. Um die Zusammengehörigkeit der Daten weiterhin zu gewährleisten, war es erforderlich, für Tabelle 32.5 einen Primärschlüssel zu definieren, der in Tabelle 32.6 als Fremdschlüssel erscheint: *MieterNr* ist in Tabelle 32.5 Primärschlüssel und taucht in Tabelle 32.6 als Fremdschlüssel wieder auf.

Nicht-Primärschlüsselfelder müssen vom Primärschlüssel voll funktional abhängig sein (2. Normalform)

Für die 2. und 3. Normalform wird vorausgesetzt, daß für jede Tabelle der Datenbank ein Primärschlüssel existiert. Dabei kann es sich um einen einfachen oder zusammengesetzten Primärschlüssel handeln. In den folgenden Beispieltabellen zur Normalisierung werden Primärschlüsselfelder unterstrichen gekennzeichnet. Das Problem, welches mit der 2. Normalform gelöst werden soll, kann nur für Tabellen mit einem zusammengesetzten Primärschlüssel existieren. Daher gilt: Tabellen, die sich in der 1. Normalform befinden und jeweils einen einfachen Primärschlüssel besitzen, befinden sich zugleich auch in der 2. Normalform. Für Tabellen mit einem zusammengesetzten Primärschlüssel lautet die Forderung der 2. Normalform: Jedes Feld, das nicht zum Primärschlüssel gehört, muß vom gesamten Primärschlüssel – und nicht nur von einzelnen seiner Felder – abhängig sein. Umgekehrt: Ein Nicht-Primärschlüsselfeld darf sich nicht schon durch einen Teil des Primärschlüssels kennzeichnen lassen. Es darf somit nicht sein, daß sich jeder Eintrag eines Nicht-Primärschlüsselfeldes bereits aus einem Teil des Primärschlüssels ableiten läßt. Dies wird auch als Forderung nach *voller funktionaler Abhängigkeit* bezeichnet.

Von den beiden obigen Tabellen der 1. Normalform befindet sich Tabelle 32.5 auch in der 2. Normalform, weil diese Tabelle einen einfachen Primärschlüssel besitzt, das Problem voller funktionaler Abhängigkeit von jedem Primärschlüsselfeld also definitionsgemäß gar nicht existiert. Tabelle 32.6 dagegen verstößt mit mehreren Feldern gegen die Forderung der 2. Normalform: Der Primärschlüssel dieser Tabelle besteht aus den Feldern *MieterNr*, *Ferienhaus* und *Mietbeginn*, die Felder *AnzahlZimmer*, *Wohnfläche* und *Kategorie* jedoch ergeben sich bereits eindeutig aus dem Feld *Ferienhaus*. Diese Felder sind also bereits von *einem* Feld des Primärschlüssels hinreichend determiniert, die anderen beiden Primärschlüsselfelder sind in Bezug auf diese drei Felder überflüssig.

MieterNr	Nachname
(1)	Webern
(3)	Noteboom
(4)	Behlmer
(5)	Preiselbeer
(6)	Hase

Tabelle 32.7: Tabelle in 2. Normalform

MieterNr	Ferienhaus	Mietbeginn	Mietende
(1)	Buntspecht	04.05.96	24.05.96
(1)	Amsel	03.05.96	23.05.96
(1)	Rotkehlchen	21.12.96	03.01.97
(3)	Pirol	03.08.97	30.08.97
(3)	Sperling	16.08.97	29.08.97
(4)	Buntspecht	15.06.96	05.07.96
(4)	Stieglitz	07.06.97	27.06.97
(5)	Nachtigall	20.07.96	02.08.96
(5)	Pirol	21.12.96	03.01.97
(6)	Lerche	21.12.96	03.01.97
(6)	Lerche	25.05.96	14.06.96

Tabelle 32.8: Tabelle in 2. Normalform

Ferienhaus	Anzahl-Zimmer	Wohn-fläche	Kategorie	Tages-preis	Wochen-preis
Buntspecht	(3)	49	Standard	130	700
Amsel	(2)	43	Standard	130	700
Rotkehlchen	(3)	50	Standard	130	700
Pirol	(3)	67	Luxus	200	1.000
Sperling	(2)	35	Einfach	90	500
Stieglitz	(2)	40	Einfach	90	500
Nachtigall	(4)	75	Luxus	200	1.000
Lerche	(4)	75	Luxus	200	1.000

Tabelle 32.9: Tabelle in 2. Normalform

Der Verstoß von Tabelle 32.6 gegen die Forderung der 2. Normalform nach voller funktionaler Abhängigkeit der Nicht-Primärschlüsselfelder vom gesamten Primärschlüssel wird dadurch beseitigt, daß Tabelle 32.6 in so viele einzelne Tabellen aufgelöst wird, daß keine mehr gegen das Gebot der 2. Normalform verstößt. Dies trifft für Tabelle 32.7, Tabelle 32.8 und Tabelle 32.9 zu.

Nicht-Primärschlüsselfelder dürfen von keinem anderen Nicht-Primärschlüsselfeld funktional abhängig sein (3. Normalform)

Die 3. Normalform fordert, daß jedes Nicht-Primärschlüsselfeld direkt und ausschließlich vom Primärschlüssel abhängt und nicht bereits durch ein anderes Nicht-Primärschlüsselfeld determiniert ist. Dies wird auch als Verbot *indirekter* oder *transitiver* Abhängigkeit bezeichnet. Unter den obigen Tabellen der 2. Normalform verstößt Tabelle 32.9 gegen diese Regel: Die

Felder *Tagespreis* und *Wochenpreis* sind bereits durch das Feld *Kategorie* bestimmt und hängen somit nur indirekt vom in diesem Falle einfachen Primärschlüssel *Ferienhaus* ab.

MieterNr	Nachname
(1)	Webern
(3)	Noteboom
(4)	Behlmer
(5)	Preiselbeer
(6)	Hase

Tabelle 32.10: Tabelle in 3. Normalform

MieterNr	Ferienhaus	Mietbeginn	Mietende
(1)	Buntspecht	04.05.96	24.05.96
(1)	Amsel	03.05.96	23.05.96
(1)	Rotkehlchen	21.12.96	03.01.97
(3)	Pirol	03.08.97	30.08.97
(3)	Sperling	16.08.97	29.08.97
(4)	Buntspecht	15.06.96	05.07.96
(4)	Stieglitz	07.06.97	27.06.97
(5)	Nachtigall	20.07.96	02.08.96
(5)	Pirol	21.12.96	03.01.97
(6)	Lerche	21.12.96	03.01.97
(6)	Lerche	25.05.96	14.06.96

Tabelle 32.11: Tabelle in 3. Normalform

Ferienhaus	Anzahl-Zimmer	Wohnfläche	Kategorie
Buntspecht	(3)	49	Standard
Amsel	(2)	43	Standard
Rotkehlchen	(3)	50	Standard
Pirol	(3)	67	Luxus
Sperling	(2)	35	Einfach
Stieglitz	(2)	40	Einfach
Nachtigall	(4)	75	Luxus
Lerche	(4)	75	Luxus

Tabelle 32.12: Tabelle in 3. Normalform

Kategorie	Tagespreis	Wochenpreis
Einfach	90	500
Standard	130	700
Luxus	200	1.000

Tabelle 32.13: Tabelle in 3. Normalform

Der Verstoß der Tabelle 32.9 gegen die Forderung der 3. Normalform wird dadurch beseitigt, daß sie in zwei Tabellen aufgelöst wird. Dies ist mit Tabelle 32.12 und Tabelle 32.13 geschehen. (Der Wochenpreis in Tabelle 32.13 ergibt sich nicht als Funktion des Tagespreises, sondern ist eigenständig gegeben.) Die Zusammenstellung von Tabelle 32.10, Tabelle 32.11, Tabelle 32.12 und Tabelle 32.13 genügt nunmehr insgesamt der Forderung nach redundanzfreien Tabellen einer Datenbank, da sich jede dieser Tabellen in der 3. Normalform befindet.

Die vorangehenden Ausführungen zum sogen. Normalisierungsprozeß mögen im Verhältnis zum Ergebnis, das damit erreicht wurde, übermäßig aufwendig erscheinen. In dieser Hinsicht ist m.E. Stahlknecht zuzustimmen, der zu einem ganz ähnlichen Beispiel wie dem hier verwendeten anmerkt: »Allerdings ist festzustellen, daß in der Fachliteratur die Originalarbeiten von Codd nicht nur unterschiedlich interpretiert, sondern z.T. auch über Gebühr formalisiert und verkompliziert werden. Zu den Relationen der 3. Normalform hätte man ... sicher auch mit Erfahrung und einer Portion gesundem Menschenverstand kommen können.«[6] Daher sei die eingangs dieses Punktes formulierte Bemerkung wiederholt: Die Normalformenlehre gibt nicht unbedingt den praktisch empfehlenswerten Weg an, nach dem Sie jede einzelne Tabelle einer Datenbank konzipieren sollten. Sie stellt vielmehr eine Systematik dar, mit deren Hilfe Sie *im Zweifelsfall* prüfen können, ob Tabellen einer Datenbank Redundanz enthalten und welcher Art diese Redundanz ggf. ist. Der praktische Prozeß des Datenbank- und Tabellenentwurfs dürfte dagegen eher intuitiv auf Erfahrung basierend ablaufen, wobei Sie sich stets des Redundanzproblems bewußt sein sollten.

Berechnete Felder

Ein Feld, dessen Werte sich aus den Werten anderer Felder durch einen Formelausdruck berechnen lassen, wird als *berechnetes Feld* (auch *abgeleitetes* Feld) bezeichnet. Dabei können die Felder, auf deren Werten die Berechnung basiert, derselben Tabelle, aber durchaus auch anderen Tabellen entstammen. Typische Beispiele für berechnete Felder sind der Gesamtpreis einer Artikelbestellung, wenn er sich aus der Multiplikation von bestellter Menge und Artikelpreis ergibt, oder der Mehrwertsteuerbetrag, der sich als mathematisches Produkt von Mehrwertsteuersatz und Nettoumsatz berechnen läßt.

6. [Stahlknecht], S. 210

In der Datenbankliteratur wird grundsätzlich gefordert, daß berechnete Felder nicht in Tabellen aufgenommen werden sollen, weil sie redundant seien. So fordert auch die Online-Hilfe von Access unter dem Thema *Entwerfen einer Datenbank* zur Bestimmung der für Tabellen benötigten Felder: »Fügen Sie keine abgeleiteten oder berechneten Daten hinzu (Daten, die das Ergebnis eines Ausdrucks sind).« Auf den ersten Blick überzeugt der Hinweis auf die Redundanz berechneter Werte, weil sie sich mit Hilfe eines Ausdrucks aus anderen Werten rekonstruieren lassen. Betrachtet man allerdings nur das reine Datenmodell einer Datenbank, wie es sich als Gesamtheit der Tabellendefinitionen ergibt, so sind berechnete Werte keinesfalls völlig redundant, weil sie sich nur rekonstruieren lassen, sofern auch der berechnende Ausdruck bekannt ist. Nun mag es sein, daß viele Ausdrücke sich quasi natürlich und unveränderlich im Zeitablauf aus dem Sachzusammenhang ergeben, wie dies z.B. für den Gesamtpreis einer Artikelbestellung, der sich aus der Multiplikation von bestellter Menge und Artikelpreis ergibt, zutrifft. Bei anderen berechneten Werten wird dies jedoch nicht ohne weiteres zutreffen. Als Beispiel sei der monatliche Lohnsteuerbetrag angeführt, der sich aus gespeicherten persönlichen Daten (wie Merkmalen der Lohnsteuerkarte und Monatseinkommen) und einem Formelausdruck zur Berechnung der Lohnsteuer, der Lohnsteuerfunktion, ergibt. Letztere wird sich im allgemeinen im Zeitablauf ändern, weil sich die Steuergesetzgebung ändert. Daher läßt sich der Lohnsteuerbetrag beliebiger Monate nicht ohne weiteres aus den persönlichen Monatsdaten eines Arbeitnehmers rekonstruieren.

Tatsächlich gilt daher für berechnete Werte, daß diese nicht völlig redundant sind, weil der berechnende Ausdruck eine notwendige Information darstellt, um den Wert zu berechnen. Um den berechneten Wert jederzeit und unter allen Umständen ermitteln bzw. rekonstruieren zu können, muß daher entweder der berechnete Wert selbst oder der diesen berechnende Ausdruck in der Datenbank gespeichert sein. Bei im Zeitablauf sich ändernden Ausdrücken müssen alle Ausdrücke gespeichert werden und zeitlich eindeutig zuzuordnen sein. Anders als in verschiedenen SQL-Servern, bei denen berechnende Ausdrücke als sogen. *stored procedures* mit den Tabellen gespeichert werden können, müssen diese in Access jedoch in Form von VBA-Prozeduren oder als Makros – also nicht als Teil des Datenmodells – abgelegt und, was in diesem Zusammenhang wichtiger ist, durch Programmierung statt durch Automatik der Datenbankmaschine ausgelöst werden. Die Datenintegrität berechneter Werte hängt daher in Access auch vom fehlerfreien Programmieren ab, wenn berechnete Werte statt in berechneten Feldern gespeichert zu werden stets mit Hilfe von ggf. sich ändernden Ausdrücken berechnet werden müssen. Daher kann sich in diesen Fällen ein Zielkonflikt zwischen der Forderung nach Redundanzvermeidung einerseits und der möglichst weitgehenden Trennung von logischen Datenstrukturen und Anwendungsprogrammierung (vgl. hierzu oben in diesem Kapitel, Punkt 32.1.3, *Externe, konzeptionelle und interne Sicht*) andererseits ergeben: Die Vermeidung von Redundanz, die nicht zuletzt Widersprüche in den

Daten verhindern soll, ist nur um den Preis zu realisieren, daß ein Teil der logischen Datenstruktur durch Programmierung realisiert wird, was bei fehlerhafter Programmierung zu Widersprüchen in den Daten führen kann. Aus diesem Grunde kann es durchaus geboten sein, berechnete Felder in Tabellen vorzusehen. Dies gilt in besonderem Maße bei komplexen, im Zeitablauf wechselnden Berechnungsprozeduren.

Zusammenfassend zu berechneten Werten läßt sich sagen: Als oberster Grundsatz sollte gelten, daß berechnete Werte jederzeit korrekt wiedergegeben werden können. Dies kann entweder dazu führen, daß die berechneten Werte selbst gespeichert werden (dies ist sozusagen das Sicherste), oder daß die berechnenden Ausdrücke mit zeitlichem Gültigkeitsbereich gespeichert werden. Dies ist bei Access allerdings nur prozedural möglich (VBA oder Makro).

32.5.4 Gründe für das Zulassen von Redundanz

Die bisherigen Überlegungen dieses Punktes haben gezeigt, daß ein redundant konzipiertes Datenmodell zu Widersprüchen (Anomalien) in den Daten führen kann. Daher stellt die Forderung nach Redundanzvermeidung einen zentralen Grundsatz der logischen Datenmodellierung dar. Dieses allein auf abstrakt-logischer Beweisführung gewonnene Prinzip kann jedoch in der praktischen Datenbankadministration auch zu Nachteilen führen, so daß Zielkonflikte entstehen können. Die Nachteile liegen außer in manchmal eingeschränkter Flexibilität vor allem in geringerer Performance von (lesenden) Abfragen: Der lesende Zugriff einer Abfrage, welche die Daten mit einem Join aus mehreren Tabellen zusammenfügen muß, ist langsamer als der gleichwertige lesende Zugriff auf eine einzige Tabelle, in der diese Daten – redundant – gespeichert sind. Dieser Performance-Nachteil redundanzfrei konstruierter relationaler Datenmodelle kann im Einzelfall recht gravierend sein. Er ist um so bedeutsamer, je mehr Datensätze gelesen und je mehr Tabellen in der Abfrage verknüpft werden müssen. In derartigen Fällen erscheint es durchaus gerechtfertigt, Redundanz im Datenmodell in Kauf zu nehmen, um Performance in Abfragen zu gewinnen. Dann allerdings ist ggf. durch entsprechende Gestaltung von Eingabeformularen oder durch Programmierung sicherzustellen, daß Widersprüche in den Daten nicht auftreten können.

Eine besonders häufige Form geplanter (und kontrollierter) Redundanz sind Abfrageergebnisse, die *temporär* als Tabellen gespeichert werden. Dieser Weg wird z.B. bei der Konstruktion betrieblicher Informationssysteme, die auf Prozeßdaten basieren, beschritten: Häufige Abfragen auf die (ohne Redundanz modellierten) Prozeßdaten sind unter Umständen nicht nur selbst zu langsam, sondern können umgekehrt auch die Administration der Prozeßdaten, die Vorrang hat, erheblich beeinträchtigen, weil schreibende Zugriffe möglicherweise wegen laufender Abfragen immer wieder gesperrt werden. Die gewissermaßen Spiegelung der Prozeßdaten in Form mehr oder minder

gebrauchsfertiger Abfrageergebnisse – temporär als Tabellen gespeichert – erlaubt dagegen eine physikalische Trennung der Datenbasis des Informationssystems und der Prozeßdaten. Der wesentliche Nachteil eines derartigen *Data Warehouse* besteht außer in zusätzlich erforderlichen Ressourcen darin, daß die Aktualität gegenüber den Prozeßdaten stets hinterher hinkt. Je nach Aktualisierungsintervall der temporären Tabellen ist dieser Nachteil mehr oder minder groß und für viele praktische Informationsbedürfnisse vernachlässigbar.

Im übrigen ist auch an dieser Stelle darauf hinzuweisen, was bereits oben im Zusammenhang mit berechneten Feldern angeführt wurde (vgl. gegen Ende von Punkt 32.5.3, *Redundanz beseitigen*, den Unterpunkt *Berechnete Felder*): Die Existenz von Redundanz kann auch eine temporäre Erscheinung sein. So kann der doppelte Eintrag des Preises eines Artikels einerseits in einer Tabelle *Artikel* und andererseits in einer Tabelle *Rechnungen* zunächst redundant sein. Wird hier jedoch die redundante Datenhaltung vermieden, kann dies zu Fehlern führen, wenn sich der Preis des Artikels ändert und dies in der Artikel-Tabelle aktualisiert wird, während in den zeitlich zurückliegenden Datensätzen der anderen Tabelle weiterhin der alte Preis gelten muß.

32.6 Beziehungen bestimmen

In den vorangehenden Punkten wurde deutlich, daß die in der Datenbank zu berücksichtigenden Informationen in aller Regel auf mehrere verschiedene Tabellen aufgeteilt werden, wobei für jede Tabelle mindestens ein Primärschlüssel (und ggf. weitere Indizes) festgelegt werden sollte. Der wesentliche Grund für die Aufteilung der Daten auf getrennte Tabellen ist die Vermeidung von Redundanz. Andererseits besteht im allgemeinen durchaus ein innerer Zusammenhang zwischen den nunmehr auf mehrere Tabellen verteilten Daten. Dieser Zusammenhang wird in einer relationalen Datenbank dadurch wiederhergestellt, daß *Beziehungen* zwischen den Tabellen definiert werden. Dabei ist zwischen den folgenden Beziehungstypen zu unterscheiden:

1:n-Beziehung

m:n-Beziehung

1:1-Beziehung

Diese Beziehungstypen und die mit ihnen möglicherweise weiter verbundenen Charakteristika seien im weiteren beschrieben und erklärt.

32.6.1 Beziehungstypen

Die im folgenden beschriebenen Beziehungen gelten jeweils zwischen zwei Tabellen.

1:n-Beziehung

Zwischen zwei Tabellen herrscht dann eine 1:n-Beziehung, wenn einem Datensatz der Tabelle A mehrere Datensätze der Tabelle B entsprechen können. 1:n-Beziehungen sind die am häufigsten vorkommenden Beziehungstypen. Typische Beispiele für derartige Beziehungen sind *Kunde – Auftrag*, *Lieferanten – Artikel* (sofern derselbe Artikel nicht auch von anderen Lieferanten geliefert werden kann) oder, wie in Bild 32.6 wiedergegeben, *Mieter – Mietvertrag*; dieses letztere Beispiel wurde oben bei der Erklärung des Normalisierungsprozesses verwendet. Es ist üblich, die Tabelle der 1-Seite als *Mastertabelle*, die der n-Seite als *Detailtabelle* zu bezeichnen. Eine 1:n-Beziehung wird in Access im Beziehungsfenster festgelegt; das Verfahren ist im einzelnen in Kap. 7, *Eine Datenbank konzipieren*, beschrieben worden, vgl. genauer dort. Eine 1:n-Beziehung wird im Access-Beziehungsfenster durch eine gerade Linie dargestellt, die masterseitig mit der Ziffer 1 und detailseitig mit einer liegenden 8 gekennzeichnet ist, vgl. Bild 32.6. Die Linie verbindet das Primärschlüsselfeld der Mastertabelle mit dem Fremdschlüsselfeld der Detailtabelle. Wenn Primärschlüssel und Fremdschlüssel aus mehr als einem Feld bestehen, wird die Beziehung durch entsprechend viele Linien dargestellt, vgl. z.B. oben Bild 32.4.

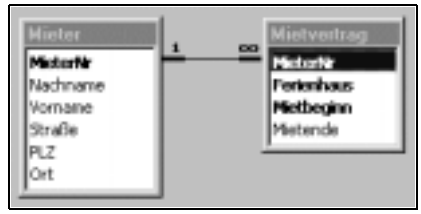


Bild 32.6: 1: n-Beziehung zwischen der Mastertabelle »Mieter« und der Detailtabelle »Mietvertrag«

Eine 1:n-Beziehung kann mit oder ohne referentielle Integrität festgelegt werden, auf die Konsequenz dieses wichtigen Sachverhalts wird im nächsten Punkt genauer eingegangen, vgl. 32.6.2, *Referentielle Integrität*. Mit einer 1:n-Beziehung ist es prinzipiell vereinbar, daß die Mastertabelle keinen oder einen Datensatz enthält, während die Detailtabelle keinen, einen oder mehrere Datensätze aufweist. Die verschiedenen möglichen Konstellationen dieser Mengenverhältnisse werden ebenfalls weiter unten in diesem Kapitel genauer behandelt, vgl. 32.6.3, *Mengenverhältnisse von Master- und Detailtabelle* (Minimal- und Maximalkardinalitäten).

m:n-Beziehung

Eine m:n-Beziehung liegt dann vor, wenn einem Datensatz der Tabelle A mehrere Datensätze der Tabelle B und gleichzeitig einem Datensatz der Tabelle B mehrere Datensätze der Tabelle A entsprechen können. Eine m:n-

Beziehung läßt sich nicht direkt zwischen zwei Tabellen definieren, vielmehr bedarf es dazu in relationalen Datenbanken stets einer dritten Tabelle, die die m:n-Beziehung sozusagen vermittelt. In dem oben bei der Darstellung des Normalisierungsprozesses verwendeten Beispiel einer Ferienhausvermietung herrscht zwischen den Tabellen *Mieter* und *Ferienhaus* eine m:n-Beziehung, weil ein Mieter im Laufe der Zeit durchaus mehrere verschiedene Ferienhäuser mieten kann; umgekehrt kann dasselbe Ferienhaus im Laufe der Zeit an mehrere verschiedene Mieter vermietet werden. Diese m:n-Beziehung kann nicht direkt zwischen den beiden Tabellen *Mieter* und *Ferienhaus* definiert werden, sondern muß über die vermittelnde Tabelle *Mietvertrag* festgelegt werden: Zwischen den Tabellen *Mieter* und *Mietvertrag* einerseits und *Ferienhaus* und *Mietvertrag* andererseits muß jeweils eine 1:n-Beziehung festgelegt werden, um eine m:n-Beziehung zwischen *Mieter* und *Ferienhaus* zu erstellen, wobei die vermittelnde Tabelle *Mietvertrag* in beiden Fällen die Detailtabelle und die Tabellen *Mieter* und *Ferienhaus* jeweils die Mastertabelle bilden. Dabei mußten die Primärschlüssel *MieterNr* aus der Tabelle *Mieter* und *Ferienhaus* aus der Tabelle *Ferienhaus* jeweils als Fremdschlüssel in die Tabelle *Mietvertrag* aufgenommen werden. Gleichzeitig bilden diese beiden Fremdschlüssel einen Teil des zusammengesetzten Primärschlüssels der Tabelle *Mietvertrag*, der außerdem noch das Feld *Mietbeginn* umfaßt, um die zeitliche Dimension der m:n-Beziehung zu berücksichtigen. Ohne diesen zeitlichen Aspekt würde es ausreichen, wenn der Primärschlüssel der Tabelle *Mietvertrag* allein aus den beiden Fremdschlüsseln gebildet würde – eine Konstellation, die für m:n-Beziehungen nicht untypisch ist.

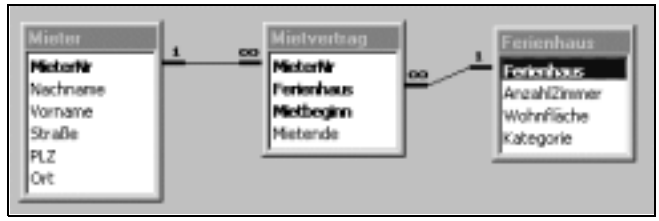


Bild 32.7: m: n-Beziehung zwischen den Tabellen »Mieter« und »Ferienhaus«, vermittelt durch die Tabelle »Mietvertrag«

1:1-Beziehung

Zwischen zwei Tabellen herrscht eine 1:1-Beziehung, wenn einem Datensatz der Tabelle A höchstens ein Datensatz der Tabelle B entsprechen kann. 1:1-Beziehungen sind praktisch seltener als 1:n-Beziehungen, weil die Daten dann in der Regel in einer Tabelle dargestellt werden können, ohne daß Redundanz auftreten muß. Gleichwohl gibt es Fälle, in denen eine 1:1-Beziehung sinnvoll sein kann. Beispielsweise könnte es sein, daß Sie einen Teil der Felder einer Tabelle mit anderen Zugriffsrechten ausstatten wollen als einen anderen Teil. Da sich Zugriffsrechte in Access nur jeweils für die gesamte

Tabelle (und nicht differenziert nach einzelnen Feldern einer Tabelle) vergeben lassen, läßt sich dieses Problem durch Aufteilung der Felder auf zwei Tabellen mit einer 1:1-Beziehung lösen. Die Aufspaltung einer Tabelle in zwei Tabellen mit einer 1:1-Beziehung kann auch sinnvoll sein, wenn die Tabelle einige Felder enthält, für die nur für wenige Datensätze Werteinträge zu erwarten sind, für die meisten Datensätze dagegen Nullwerte. Beispielsweise könnte eine Personaltabelle Felder enthalten, in die Angaben zur Ausübung des Segelsports eingetragen werden. Da zu erwarten ist, daß nur wenige Mitarbeiter derartige Auskünfte geben können, weil die meisten keinen Segelsport ausüben, wäre es sinnvoll, diese Felder in eine eigene Tabelle auszulagern und mit der Ausgangstabelle eine 1:1-Beziehung herzustellen.

In Access wird eine 1:1-Beziehung – wie auch eine 1:n-Beziehung – im Beziehungsfenster festgelegt. Sie können eine 1:1-Beziehung allerdings nicht explizit anfordern, sondern Sie legen diese implizit fest, indem Sie das Primärschlüsselfeld der Mastertabelle auf das Primärschlüsselfeld der Detailtabelle ziehen (bei zusammengesetzten Primärschlüsseln ziehen Sie entsprechend mehrere Felder). Access legt diese Beziehung dann automatisch als 1:1-Beziehung fest. Eine 1:n-Beziehung wäre in dieser Situation, in der der Fremdschlüssel der Detailtabelle zugleich deren Primärschlüssel ist, ja auch nicht möglich, weil Primärschlüsselwerte eindeutig sein müssen und daher nur einmal vorkommen dürfen.

Auf die Bedeutung der referentiellen Integrität für 1:1-Beziehungen wird im folgenden Punkt im einzelnen eingegangen.

32.6.2 Referentielle Integrität

1:n- oder 1:1-Beziehungen können mit oder ohne referentielle(r) Integrität definiert werden. Referentielle Integrität bezieht sich auf den Wert des Fremdschlüssels der Detailtabelle im Verhältnis zum Wert des Primärschlüssels der Mastertabelle und bedeutet im einzelnen das folgende:

> *Jeder Wert des Fremdschlüssels der Detailtabelle muß mit einem Wert des Primärschlüssels der Mastertabelle übereinstimmen oder NULL sein.*

Wenn für eine Beziehung in Access referentielle Integrität festgelegt ist, hat die Überwachung dieser Regel zur Konsequenz, daß das Einfügen, Löschen oder Ändern von Datensätzen, sofern es einen Verstoß gegen die Regel zur Folge hätte, abgelehnt wird. Damit wird sichergestellt, daß eine Detailtabelle niemals *verwaiste Datensätze* enthält, deren Fremdschlüsselwert keinem Primärschlüsselwert der Mastertabelle entspräche – es sei denn, es handelt sich um einen Nullwert. Die Beachtung der referentiellen Integrität wird, sofern für eine Beziehung vereinbart, von der JetEngine überwacht.

Die Regel der referentiellen Integrität stellt ausschließlich auf die Übereinstimmung der Fremdschlüsselwerte der Detailtabelle mit den Primärschlüsselwerten der Mastertabelle ab, wobei Nullwerte zugelassen sind. Nicht

weniger, aber auch nicht mehr bedeutet referentielle Integrität. Weil dies nach meiner Erfahrung nicht immer klar ist, erscheint es nützlich, sich ebenfalls zu verdeutlichen, was referentielle Integrität *nicht* bedeutet: Referentielle Integrität sagt nichts aus über das Mengenverhältnis der Datensätze in der Mastertabelle und der Detailtabelle. So ist es mit der Regel der referentiellen Integrität gleichermaßen vereinbar, daß einem Datensatz der Mastertabelle in einer 1:n-Beziehung kein Datensatz, ein Datensatz oder mehrere Datensätze in der Detailtabelle entsprechen. Wenn ein oder mehrere Datensätze der Detailtabelle einen Nullwert im Fremdschlüssel besitzen, kann diesen kein einziger Datensatz in der Mastertabelle entsprechen, weil Nullwerte im Primärschlüssel unzulässig sind. Daraus folgt, daß besondere Restriktionen bezüglich der Mengenverhältnisse von Master- und Detailtabelle nicht mit der Regel der referentiellen Integrität durchgesetzt werden können, sondern auf andere Weise erzwungen werden müssen, vgl. dazu im einzelnen weiter unten Punkt 32.6.3, *Mengenverhältnisse von Master- und Detailtabelle* (Minimal- und Maximalkardinalitäten).

Standardmäßig führt die Beachtung der referentiellen Integrität dazu, daß der Primärschlüsselwert eines Datensatzes der Mastertabelle nicht geändert werden kann, wenn dazu in der Detailtabelle bereits ein Datensatz existiert, der diesen Primärschlüsselwert als Fremdschlüsselwert besitzt. Gleichfalls gilt unter diesen Umständen, daß ein derartiger Datensatz nicht aus der Mastertabelle gelöscht werden kann. Ändern und Löschen wird verhindert, um keine verwaisten Datensätze in der Detailtabelle entstehen zu lassen.

Diese sehr restriktive Konsequenz referentieller Integrität kann für das Aktualisieren und Löschen von Datensätzen der Mastertabelle gemildert werden, ohne daß die Regeln der referentiellen Integrität verletzt werden, wenn Sie *Aktualisierungsweitergabe an Detailfeld* und/oder *Löschweitergabe an Detaildatensatz* vereinbaren, vgl. Bild 32.8.

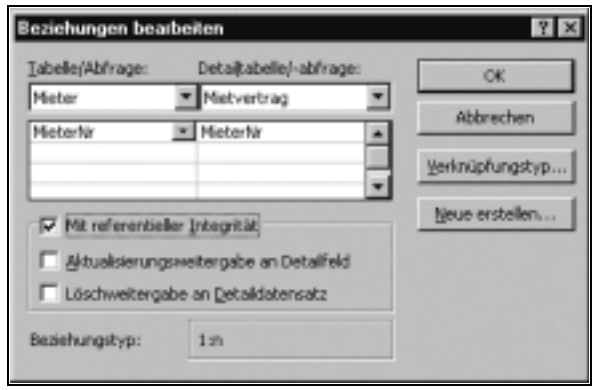


Bild 32.8: Dialogfeld »Beziehungen«, in dem Sie in Access auch referentielle Integrität mit Aktualisierungs- und Löschweitergabe festlegen können.

Mit Aktualisierungsweitergabe

Die Vereinbarung von *Aktualisierungsweitergabe an Detailfeld* für eine Beziehung mit referentieller Integrität ermöglicht es, den Primärschlüsselwert eines Datensatzes der Mastertabelle auch dann zu ändern, wenn diesem ein oder mehrere Datensätze in der Detailtabelle entsprechen, weil deren Fremdschlüsselwerte mit denen des zu aktualisierenden Primärschlüsselwertes übereinstimmen: Access ändert in diesem Falle auch die Fremdschlüsselwerte der korrespondierenden Detaildatensätze, so daß im Ergebnis nach wie vor der Änderung nicht gegen die Regel der referentiellen Integrität verstoßen wird.

Bei der Vereinbarung referentieller Integrität mit *Aktualisierungsweitergabe*, die auch als *kaskadierende Aktualisierung* bezeichnet wird, müssen Sie beachten, daß diese jeweils nur unmittelbar für die eine Beziehung, für die sie definiert wurde, gilt. Ein Problem kann sich ergeben, wenn drei (oder mehr) Tabellen in einer Kette jeweils eine Beziehung mit referentieller Integrität aufweisen, wobei der Primärschlüssel der ersten Tabelle als Fremdschlüssel und zugleich als Bestandteil des (zusammengesetzten) Primärschlüssels in der zweiten Tabelle und zugleich als Fremdschlüssel in der dritten Tabelle erscheint. Um auch in diesem Fall eine kaskadierende Aktualisierung bei gleichzeitiger Gewährleistung referentieller Integrität zu erreichen, müssen beide Beziehungen mit *Aktualisierungsweitergabe an Detailfeld* vereinbart sein.

Mit Löschweitergabe

Standardmäßig kann ein Datensatz der Mastertabelle bei vereinbarter referentieller Integrität nicht gelöscht werden, wenn mindestens ein Datensatz der Detailtabelle von ihm abhängt. Dies ist jedoch möglich, wenn für die Beziehung der beiden Tabellen referentielle Integrität mit *Löschweitergabe an Detaildatensatz* vereinbart wurde. Dann werden die vom Datensatz der Mastertabelle abhängigen Datensätze der Detailtabelle – nach Bestätigung einer entsprechenden Warnung – zusammen mit dem Datensatz der Mastertabelle gelöscht. Auch hier gilt Entsprechendes wie für die Aktualisierungsweitergabe (vgl. den vorangehenden Absatz), wenn drei oder mehr Tabellen eine Kette von Beziehungen mit referentieller Integrität bilden: Damit *kaskadierendes Löschen* auch in diesem Falle möglich ist, muß für jede der die Kette bildenden Beziehungen referentielle Integrität mit Löschweitergabe vereinbart werden.

Löschen mit generierten Nullwerten

Es ist darauf hinzuweisen, daß die Wirkung der Löschweitergabe bei referentieller Integrität sehr weitreichend ist. Für manche Situation mag dies zu weitreichend erscheinen. Als Beispiel sei eine 1:n-Beziehung mit referentiel-

ler Integrität zwischen den beiden Tabellen *Lieferanten* und *Artikel* betrachtet. Wenn hier ein Lieferant gelöscht werden soll und Löschweitergabe vereinbart ist, gehen auch die Informationen über die von diesem Lieferanten bisher gelieferten Artikel verloren. Um dies zu vermeiden, müßte entweder vor dem Löschen des Lieferanten die referentielle Integrität aufgehoben werden – ein sicher im allgemeinen zu weitgehender Eingriff in das Datenmodell –, oder die Fremdschlüsselwerte der betreffenden Artikel-Datensätze müßten vor dem Löschen des Lieferanten-Datensatzes auf NULL gesetzt oder einem anderen, möglicherweise fiktiven Lieferanten zugeordnet werden. Insbesondere das Löschen des Datensatzes der Mastertabelle in Verbindung mit dem Umwandeln der Fremdschlüsselwerte der betroffenen Datensätze in der Detailtabelle dürfte von praktischem Interesse sein. Access stellt für diese Variante allerdings keine Option zur Verfügung. Daher müßte sie durch Anwendungsprogrammierung realisiert werden, was jedoch relativ leicht durch eine vom Ereignis *Beim Löschen* auszulösende Prozedur, die eine entsprechende Aktualisierungsabfrage ausführt, erreicht werden kann.

32.6.3 Mengenverhältnisse von Master- und Detailtabelle (Minimal- und Maximalkardinalitäten)

Das Festlegen einer 1:1- oder einer 1:n-Beziehung mit referentieller Integrität determiniert die möglichen Mengenverhältnisse der Datensätze in der Master- und Detailtabelle nur in sehr weiten Grenzen. Im weiteren beschränke ich mich auf 1:n-Beziehungen mit referentieller Integrität, weil diese die größte praktische Bedeutung haben. Die analoge Übertragung der für diesen Beziehungstyp gewonnenen Ergebnisse auf 1:1-Beziehungen dürfte dann nicht schwer sein.

Fall	Anzahl Datensätze Mastertabelle	Anzahl Datensätze Detailtabelle
1	0	1
2	0	n
3	1	0
4	1	1
5	1	n

Tabelle 32.14: Mögliche Mengenverhältnisse der Datensätze bei einer 1: n-Beziehung mit referentieller Integrität

Die bei einer 1:n-Beziehung mit referentieller Integrität möglichen Mengenkonstellationen der Datensätze sind in Tabelle 32.14 angegeben. Es mag auf den ersten Blick überraschen, daß die Detailtabelle einer 1:n-Beziehung mit referentieller Integrität einen oder mehrere Datensätze enthalten kann, dem oder denen kein einziger Datensatz in der Mastertabelle entspricht, vgl. Fälle 1 und 2 in Tabelle 32.14. Dies ist jedoch möglich, weil Nullwerte im

Fremdschlüssel der Detailtabelle mit referentieller Integrität vereinbar sind. Die Konstellation von Fall 3 trifft zu, wenn in die Mastertabelle ein Datensatz eingegeben wurde, dessen Primärschlüsselwert sich im Fremdschlüssel keines Datensatzes der Detailtabelle wiederfindet. Fälle 4 und 5 beschreiben den vermutlichen Standardfall einer 1:n-Beziehung mit referentieller Integrität, in dem einem Datensatz der Mastertabelle ein oder mehrere Datensätze in der Detailtabelle entsprechen.

In den weiteren Ausführungen wird gezeigt, wie einzelne der in Tabelle 32.14 angeführten Konstellationen ausgeschlossen werden können. Dazu ist es nützlich, den Terminus *Kardinalität* einzuführen, mit dem die Anzahl der Datensätze einer Tabelle oder Beziehung bezeichnet wird. Die *Maximalkardinalität* gibt die Höchstzahl, die *Minimalkardinalität* entsprechend die Mindestzahl von Datensätzen an[7]. Kardinalitäten können einzelne Tabellen, aber auch Tabellenbeziehungen betreffen. So gibt die *Maximalkardinalität* der Detailtabelle einer 1:n-Beziehung an, wie viele Datensätze einem Datensatz der Mastertabelle höchstens entsprechen können. Analog gibt die *Minimalkardinalität* der Detailtabelle einer 1:n-Beziehung an, wie viele Datensätze einem Datensatz der Mastertabelle mindestens entsprechen müssen.

Die pure Definition einer 1:n-Beziehung mit referentieller Integrität vermag keinen der in Tabelle 32.14 angeführten Fälle auszuschließen. Unter Verwendung des Begriffs Kardinalität lassen sich die Zusammenhänge für eine 1:n-Beziehung mit referentieller Integrität folgendermaßen formulieren:

Die *Maximalkardinalität der Mastertabelle* ist durch die Definition einer 1:n-Beziehung mit referentieller Integrität auf 1 festgelegt, weil der Primärschlüsselwert, der zu einem oder mehreren Datensätzen der Detailtabelle paßt, eindeutig sein muß.

Die *Minimalkardinalität der Mastertabelle* ist standardmäßig 0, weil der Fremdschlüssel der Detailtabelle standardmäßig Nullwerte enthalten darf. Um für die Mastertabelle eine Minimalkardinalität von 1 zu erreichen, muß die Eigenschaft *Eingabe erforderlich* des Fremdschlüssels der Detailtabelle auf Ja eingestellt werden, weil dadurch Nullwerte für den Fremdschlüssel ausgeschlossen werden. Für eine 1:n-Beziehung von *Lieferant* und *Artikel* würde dies zur Folge haben, daß kein Artikel eingegeben werden kann, ohne daß gleichzeitig sein Lieferant angegeben wird.

Die *Maximalkardinalität der Detailtabelle* ist in einer 1:n-Beziehung definitionsgemäß n und damit theoretisch unbegrenzt. Es würde dem Sinn dieses Beziehungstyps widersprechen, wenn sie auf 1 beschränkt würde, was innerhalb einer 1:n-Beziehung – außer durch Programmierung – auch gar nicht zu erreichen ist. Wenn die Detailtabelle einer Beziehung eine Maximalkardinalität von 1 besitzen soll, wäre von vornherein eine 1:1-Beziehung zu definieren, welche genau dies leistet.

7. vgl. besonders [Sommer], S. 227 f.

Die *Minimalkardinalität der Detailtabelle* ist standardmäßig 0. Obwohl man sich einen Sachverhalt vorstellen kann, bei dem eine Minimalkardinalität der Detailtabelle von 1 (oder >1) sinnvoll wäre (Beispiel: Ein Kunde soll nur aufgenommen werden oder weiter existieren, wenn von ihm wenigstens eine Bestellung vorliegt), ist dies nicht einfach durch eine Änderung eines Tabellenmerkmals zu erreichen. Vielmehr müßte durch wohl verhältnismäßig aufwendige Anwendungsprogrammierung sichergestellt werden, daß weder durch Einfügen oder Ändern eines Masterdatensatzes noch durch Löschen oder Ändern eines Detaildatensatzes ein verwaister Masterdatensatz entstehen kann.

32.7 Datenbankkonzept dokumentieren

Das fertige Datenmodell, das die Definition der Tabellen einschließlich aller Felder mit ihren diversen Eigenschaften und der Indizes sowie die Beziehungen zwischen den Tabellen umfaßt, sollte unbedingt in schriftlicher Form dokumentiert werden. Zwar stecken sämtliche Festlegungen implizit in den verschiedenen Datenbankobjekten, so daß sie prinzipiell jederzeit aus den Objekten ausgelesen werden können. Jedoch ist dies erstens bei laufendem Datenbankbetrieb häufig nicht oder nur schwer möglich, und zweitens verlieren Sie bei punktuellem Nachsehen leicht den Blick für strukturelle Zusammenhänge. Zu professionellem Arbeiten gehört, daß das Datenmodell systematisch in schriftlicher Form dokumentiert und so aufbewahrt wird, daß es auch von Dritten nachvollzogen werden kann.

Access unterstützt dieses Erfordernis, indem es den *Dokumentierer* zur Verfügung stellt. Mit diesem Werkzeug können Sie nicht nur das reine Datenmodell, wie es sich in Tabellen- und Beziehungsdefinitionen niederschlägt, auf ggf. sehr differenzierte Weise schriftlich dokumentieren, sondern auch definitorische Informationen der anderen Objekte wie Abfragen oder Formulare ausgeben lassen. Die Informationen des Dokumentierers werden standardmäßig als Access-Bericht auf dem Bildschirm ausgegeben, den Sie dann drucken lassen können. Sie können jedoch auch, wie von jedem Access-Objekt, eine Ausgabe des Berichts als Winword-Datei (allerdings im RTF-Format) veranlassen, um dann ggf. Ihre eigenen Bearbeitungen daran vorzunehmen. (Die ebenfalls mögliche Ausgabe als Excel-Arbeitsmappe führt zu völlig unbefriedigenden Ergebnissen und erscheint daher nicht empfehlenswert.)

Sie rufen den *Dokumentierer* auf mit dem gleichnamigen Unterbefehl zum Befehl *Analyse* im Menü *Extras*. Er zeigt das Dialogfeld *Dokumentierer*, vgl. Bild 32.9, in dem Sie die Objekte auswählen, deren definitorische Merkmale Sie dokumentiert haben möchten. Für die Dokumentation des Datenmodells wären in erster Linie die Tabellen auszuwählen. Bevor Sie mit der Schaltfläche OK das Erstellen des Berichts mit den Tabellendefinitionen veranlassen, sollten Sie auf jeden Fall mit der Schaltfläche *Optionen...* das Dialogfeld

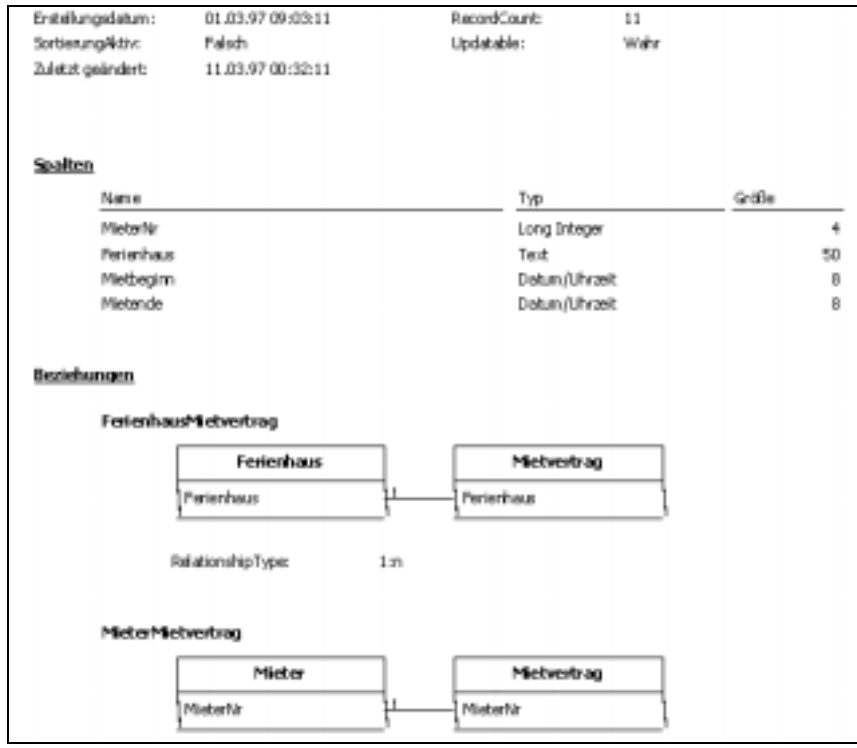
Tabellendefinition drucken (es heißt für andere Objekte entsprechend anders) öffnen, um zu bestimmen, wie differenziert die Informationen ausgegeben werden, vgl. Bild 32.10. Wenn Sie für die Tabellen, Felder und Indizes jeweils den größtmöglichen Detaillierungsgrad wählen, erhalten Sie je Tabelle leicht einen 10-seitigen Bericht. Da Sie die vom Dokumentierer erzeugten Berichte auf dem Bildschirm betrachten können, bevor Sie diese drucken oder als Textdatei ausgeben lassen, ist es nicht weiter schlimm, wenn Sie zunächst einen zu knappen oder zu detaillierten Bericht erhalten.

Bild 32.9: Eröffnungsdialogfeld des Dokumentierers

Bild 32.10: In diesem Dialogfeld können Sie angeben, wie detailliert die ausgegebenen Informationen sein sollen.

Einen Ausschnitt aus dem Bericht des Dokumentierers für die Tabelle *Mietvertrag* mit dem Differenziertheitsgrad für die Definitionsinformationen, wie er in den Optionen in Bild 32.10 gewählt ist, gibt Bild 32.11 wieder. Man

erkennt u.a., daß außer gewissen Tabelleneigenschaften und Feldmerkmalen auch die beiden 1:n-Beziehungen zu den Tabellen *Ferienhaus* und *Mieter* festgehalten sind.



Erstellungsdatum:	01.03.97 09:03:11	RecordCount:	11
SortierungAktiv:	Falsch	Updatable:	Wahr
Zuletzt geändert:	11.03.97 00:32:11		

Spalten

Name	Typ	Größe
MieterNr	Long Integer	4
Ferienhaus	Text	50
Mietbegim	Datum/Uhrzeit	8
Mietende	Datum/Uhrzeit	8

Beziehungen

FerienhausMietvertrag

RelationshipType: 1:n

MieterMietvertrag

Bild 32.11: Ausschnitt aus dem Bericht des Dokumentierers, hier zur Tabelle »Mietvertrag«. Die Informationsdetails entsprechen der Optionsauswahl, wie sie in Bild 32.10 wiedergegeben ist.

Kapitel 33

Integration von Access 2000 in andere Office 2000-Anwendungen

33.1 Überblick

Alle Anwendungen, die unter Windows laufen, sind zumindest in der Weise integriert, als sie die Zwischenablage zum Austausch von Daten unterschiedlichsten Formats (Zahlen, Text, Tabellen, Grafiken usw.) nutzen können. Über diese minimale Integrationsform hinaus existieren, je nach Anwendung, weitergehende Integrationsmöglichkeiten. Beispielsweise können sie in den meisten Anwendungen Dokumente anderer Formate importieren, wobei dann in das Format der importierenden Anwendung konvertiert wird. Entsprechendes gilt für den Datenexport. Speziell für Access 2000 gilt, daß diese Anwendung zahlreiche Möglichkeiten des Im- und Exports von Tabellen, Abfragen usw. bietet, vgl. z.B. für Tabellen genauer Kap. 9, *Tabellen importieren, exportieren und verknüpfen*. Seit Office 97 hat Microsoft die wechselseitige Integration der Office-Anwendungskomponenten wie Access, Excel , Word oder Powerpoint ein großes Stück vorangetrieben. Seitdem kann jede dieser Anwendungen sowohl als OLE-Client wie auch als OLE-Server fungieren. (Diese Begriffe wie überhaupt der Sprachgebrauch für diesen Zusammenhang werden gleich näher erläutert). In diesem Buch kann aus räumlichen Gründen das Integrationskonzept nicht umfassend für alle Office-Komponenten dargestellt werden[1].

1. Zu einer umfassenden Einführung, Darstellung und Erklärung anhand zahlreicher Beispiele vgl. die Veröffentlichung in diesem Verlag: Brosius/Benkwitz/Böker/Tegel, Integrierte Anwendungslösungen mit MS Office 2000. ActiveX, ODBC, DAO, Bonn 1997.

In dem vorliegenden Kapitel möchte ich zeigen, wie sich die Integration der Office-Komponenten aus der Sicht von Access 2000 darstellt. Dabei konzentriere ich mich auf zwei Betrachtungsweisen, denen auch zwei Funktionsarten entsprechen:

- **Access als Client**. Die Begriffe *Client* und *Server* sind allgemein und werden nicht speziell im Zusammenhang mit Microsoft-Produkten verwendet. Eine Anwendung tritt dann in der Funktion eines Clients auf, wenn sie die Dienste einer anderen Anwendung in Anspruch nimmt. Diese andere Anwendung fungiert dann als Server. Beispiel: Von der Anwendung Access aus soll, basierend auf den Daten einer Tabelle oder Abfrage, eine Pivot-Tabelle mit Hilfe von Excel 2000 erstellt werden. In diesem Falle würde Access als Client und Excel als Server fungieren.
- **Access als Server**. Wenn umgekehrt eine andere Anwendung eine Aufgabe von Access erledigen lassen möchte, steht diese andere Anwendung in der Funktion eines Clients und Access in der eines Servers. Beispiel: Powerpoint 2000 fordert ein bestimmtes Access-Formular an, das VBA-gesteuert in eine Folie eingebaut werden und das Bearbeiten von Daten zulassen soll. In diesem Beispiel fungiert Access als Server und Powerpoint als Client.

Das Client/Server-Konzept ist sehr allgemein und kann von unterschiedlichen Hard- und Softwaretechnologien unterstützt und getragen werden. Im Kontext von Office-Komponenten spielen vor allem die Konzepte *ActiveX (COM/OLE) DAO* und ADO eine besondere Rolle, daneben als allgemeine Grundlage für einen Datenaustausch auch *ODBC*. Das besondere Kennzeichen des Office 2000-Pakets liegt in der nunmehr für alle Office-Komponenten realisierten Möglichkeit, basierend auf dem ActiveX-Konzept sowohl als Client wie auch als (dann sogenannter) Automationsserver zu fungieren. Aus diesem Grunde konzentriere ich die folgenden Ausführungen auf diesen (umfassenden) Integrationsaspekt.

33.1.1 Automatisierung mit ActiveX (COM/OLE)

Die Technologie, auf der die Möglichkeit des Client/Server-Konzepts unter Windows beruht, wird seit einiger Zeit als ActiveX-Technologie bezeichnet. Vielleicht waren auch Sie zunächst erstaunt, mit welch revolutionär neuem Konzept Microsoft da aufwartet, als dieser Begriff propagiert wurde. Tatsächlich handelt es sich um alten Wein in neuen Schläuchen (wobei der Wein nicht schlecht ist!): Was jetzt mit ActiveX bezeichnet wird, war bis dahin unter dem Namen OLE (Object Linking and Embedding) bekannt. In grundlegender Literatur, die sich mit der Frage verteilter Objekte beschäftigt, wurde und wird statt dessen auch von COM (Component Object Model), manchmal auch von COM/OLE gesprochen. Welchen Begriff auch immer sie finden, es ist stets dasselbe gemeint, nämlich das, was jetzt (und wohl auch in Zukunft) mit ActiveX bezeichnet wird. Die Online-Hilfe definiert eine ActiveX-Komponente folgendermaßen:

Eine Anwendung, die von einer anderen Anwendung bereitgestellte Objekte verwenden kann, oder die ihre eigenen Objekte zur Verwendung durch eine andere Anwendung exponiert. ActiveX-Komponenten wurden früher als *OLE-Automatisierungs-Server* und *OLE-Automatisierungs-Controller* bezeichnet.

Entscheidend an dieser Definition ist der Umstand, daß Objekte fremder Anwendungen verwendet werden können, was natürlich nur geht, wenn die fremde Komponente ihre Objekte zur Verfügung stellt (exponiert). Die Reichweite dieses Sachverhaltes wird erst richtig klar, wenn Sie sich verdeutlichen, daß die Verwendung fremder Komponenten im objektorientierten Modell stattfindet: Zu den Objekten gehören in der Regel auch Methoden und Eigenschaften. Wenn eine Anwendung einer anderen ihre Objekte zum Gebrauch exponiert, bedeutet dies gleichzeitig, daß damit auch die mit den Objekten verbundenen Methoden und Eigenschaften und damit implizit die gesamte Funktionalität der Anwendung angeboten werden. Auf diese Weise ist es z.B. möglich, von Access aus über die Funktionalität von Excel oder Word zu verfügen, deren Verwendung von Access gesteuert wird.

Beispiel. Da sowohl Excel 2000 wie auch Access 2000 ActiveX-Komponenten sind, bieten sie sich wechselseitig ihre Objekte zur Verwendung in der jeweils anderen Anwendung an. In Excel 2000 gibt es beispielsweise das Objekt *Application*, das für die Anwendung Excel 2000 selbst steht und in der Objekthierarchie des Excel 2000-Objektmodells die oberste Stelle einnimmt. Indem Excel 2000 u.a. dieses Objekt zur Verfügung stellt, stellt es die Anwendung selbst zur Verfügung. Ein Unterobjekt des Excel 2000-Objekts *Application* ist das Auflistungsobjekt *Workbooks*, das alle (tatsächlichen und potentiellen) Arbeitsmappen enthält. Da *Workbooks* ein Unterobjekt von *Applications* ist, wird es von Excel 2000 ebenfalls exponiert, damit aber auch dessen Methode, z.B. die Methode *Add*. Aus diesem Grunde verfügen Sie in Access 2000, dem die Excel 2000-Objekte ja offenstehen, über das Objekt *Workbooks* und dessen Methode *Add*, also auch über die Anweisung

```
Workbooks.Add
```

In dieser Form würden sie mit Excel 2000-VBA-Code der laufenden Sitzung eine neue Arbeitsmappe hinzufügen. Mit einer kleinen Ergänzung können Sie diese Codezeile auch in Access 2000-VBA verwenden, um einer Excel 2000-Sitzung eine neue Arbeitsmappe hinzuzufügen. Die kleine Ergänzung betrifft den Umstand, daß Sie in Access 2000-VBA Access mitteilen müssen, daß diese Anweisung nicht Access 2000, sondern den ActiveX-Server Excel 2000 betrifft. Dies geschieht dadurch, daß der Anweisung der Name der Objektvariablen, unter der das Objekt *Application* von Excel 2000 in der aktuellen Prozedur identifiziert wird, vorangestellt wird. Dies könnte z.B. der Variablenname *xlAnw* sein. Die folgenden Codezeilen in einer Access 2000-VBA-Prozedur reichen tatsächlich aus, um einer Excel 2000-Sitzung eine neue Arbeitsmappe hinzuzufügen:

```
Dim xlAnw As Object
Set xlAnw = CreateObject("Excel.Application")
xlAnw.Workbooks.Add
```

Anmelden in der Windows-Registrierung

Von speziellen Ausnahmen abgesehen kann eine ActiveX-Komponente ihre Objekte anderen Anwendungen nur dann wirksam exponieren, wenn sie in der Windows-Registrierung angemeldet ist. Daher ist es – insbesondere auch für das Installieren neuer ActiveX-Komponenten – oberstes Gebot, diese mit einem Setup-Programm in der Registrierung anzumelden. Wenn Sie die Anwendungen des Office 2000-Pakets mit dessen Setup-Programm ordnungsgemäß installiert haben, ist diese Anmeldung erfolgt.

Einstellen von Verweisen auf Klassenbibliotheken

Nach korrekter Anmeldung in der Windows-Registrierung stehen zwar viele Objekte, Methoden und Eigenschaften einer ActiveX-Komponente, auf die von VBA-Code aus zugegriffen wird, zur Verfügung, jedoch gilt dieses nicht für alle Schlüsselwörter der Anwendung. Gerade die verschiedenen Anwendungskomponenten des Office 2000-Pakets arbeiten mit zahlreichen Objekten und Konstanten, die nicht ohne weiteres erkannt werden. Excel 2000 beispielsweise verwendet Konstanten wie *xlDataBase* oder *xlOpaque*. Wenn Sie sicher sein wollen, daß wirklich alle Objekte, Methoden, Eigenschaften und weiteren Elemente der verschiedenen Objekte (=Klassen) einer ActiveX-Komponente in den VBA-Prozeduren einer Access-Datenbank erkannt werden, müssen Sie auf diese einen Verweis einstellen. Dies geschieht mit dem Befehl *Verweise* aus dem Menü *Extras*, der zur Verfügung steht, wenn ein Modulfenster aktiv ist. Sie haben dann auch den Vorteil, daß Sie sämtliche Objekte und weiteren Elemente der entsprechenden Klassenbibliothek im Objektkatalog einsehen und möglicherweise auch Hilfe dazu anfordern können. In Bild 33.1 ist der aufgeschlagene Objektkatalog zu sehen. Dort können Sie erkennen, daß u.a. die Klassenbibliothek *Excel* verfügbar ist. Dies resultiert daraus, daß ein Verweis darauf eingerichtet wurde. Unter den Klassen (=Objekten) von Excel befindet sich auch die Klasse *XlPivotTableSourceType*, die vier Elemente enthält, das sind in diesem Falle Konstanten, darunter auch die Konstante *xlDatabase*. Wenn Sie auf Excel als Klassenbibliothek keinen Verweis einstellen, tritt ein Laufzeitfehler auf, wenn Sie beispielsweise die Konstante xlDatabase im VBA-Code verwenden. Ist umgekehrt ein solcher Verweis eingerichtet, tritt der Laufzeitfehler nicht auf, weil sie dann über alle Elemente verfügen können.

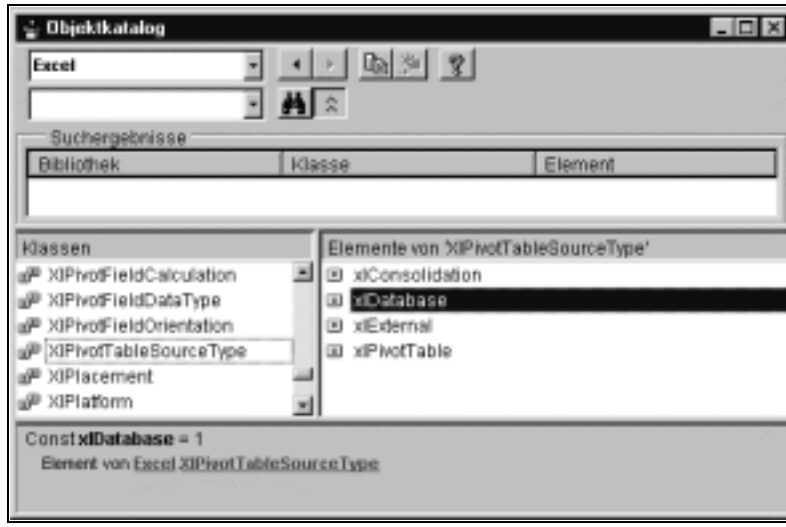


Bild 33.1: Dialogfeld »Objektkatalog«

Weitere Vorteile von Verweisen

Die Prozeduren, die mit Objekten anderer Anwendungen arbeiten, werden schneller ausgeführt, wenn Sie einen Verweis auf die Klassenbibliothek einer anderen Anwendung eingerichtet haben.

Haben Sie einen Verweis eingerichtet, so können Sie eine Objektvariable auch mit dem Schlüsselwort *New* deklarieren, beispielsweise in der Form

```
Dim xlAnw As New Excel.Application
```

Wenn Sie keinen Verweis auf die Microsoft Excel-Klassenbibliothek eingerichtet haben, müssen Sie die Variable als sogen. generische Variable vom Typ *Object* deklarieren und eine *Set*-Zuweisung mit einer der beiden Funktionen *CreateObject* oder *GetObject* vornehmen. Der folgende Code wird langsamer ausgeführt als der zuletzt angeführte:

```
Dim xlAnw As Object
Set xlAnw = CreateObject("Excel.Application")
```

Was bisher aus der Sicht von Access beschrieben wurde, gilt auch umgekehrt, wenn Sie von einer anderen Anwendung aus über die Objekte von Access verfügen wollen: Da Access 2000 eine ActiveX-Komponente ist, die die Automatisierung unterstützt, können Sie auch aus einer anderen Anwendung heraus einen Verweis auf die Klassenbibliothek von Access 2000 einrichten und dann mit Access-Objekten aus dieser Anwendung heraus effektiver arbeiten.

33.1.2 Im VBA-Code auf Objekte anderer Anwendungen verweisen

Allgemein gehen Sie folgendermaßen vor, um über die Objekte, Methoden und Eigenschaften anderer ActiveX-Komponenten mit VBA-Prozeduren zu verfügen:

- Deklarieren Sie eine Variable vom Typ *Object*. Dieser Schritt ist nicht notwendig, jedoch empfehlenswert – nicht zuletzt aus Gründen der besseren Performance.
- Weisen Sie einer Objektvariablen ein ActiveX-Objekt zu. Hierzu stehen Ihnen die Funktionen *CreateObject* und *GetObject* zur Verfügung. Alternativ können Sie in einer *Dim*- oder einer *Set*- Anweisung das Schlüsselwort *New* verwenden.
- Um über die Objekte, Methoden und Eigenschaften der ActiveX-Komponente zu verfügen, schreiben Sie Codezeilen, wie Sie diese im VBA-Code der anderen Anwendung schreiben würden – mit dem einzigen Unterschied, daß Sie hier jeweils den Namen der Objektvariablen mit einem Punkt voranstellen müssen.

Die drei Möglichkeiten der Funktionen *CreateObject* und *GetObject* und des Schlüsselwortes *New* seien im folgenden kurz erklärt und gegeneinander abgewogen:

CreateObject

Diese Funktion wird verwendet, um in einer VBA-Prozedur auf ein ActiveX-Objekt zuzugreifen. Der Zugriff ist auch dann möglich, wenn die betreffende Anwendung bis dahin nicht aktiv war, denn sie wird ggf. durch die Funktion gestartet. Die Syntax der Funktion lautet:

CreateObject(*Klasse*)

Das Argument *Klasse* wird als Zeichenfolge angegeben. Es hat die Syntax *AnwName.Objekttyp* und besteht aus den folgenden Elementen:

Element	Beschreibung
AnwName	Erforderlich. Der Name der Anwendung, die das Objekt bereitstellt.
Objekttyp	Erforderlich. Der Typ oder die Klasse des zu erstellenden Objekts.

Sie erstellen ein ActiveX-Objekt, indem Sie das von *CreateObject* zurückgegebene Objekt einer Objektvariablen mit der Anweisung *Set* zuweisen. Das folgende Beispiel deklariert zunächst die Variable »xlBlatt« vom Typ *Object* und weist dieser dann mit der *Set*-Anweisung unter Verwendung der Funktion *CreateObject* ein Excel Arbeitsblatt zu.

```
Dim xlBlatt As Object
Set xlBlatt = CreateObject("Excel.Sheet")
```

Diese beiden Codezeilen veranlassen die Anwendung, das Objekt (hier: ein Excel-Arbeitsblatt in einer Arbeitsmappe) zu erstellen. Sie können dann im weiteren Verlauf der Prozedur unter Bezugnahme auf die Variable »xlBlatt« über das Objekt verfügen. Dabei müssen Sie die Methoden und Objekte verwenden, die in der anderen Anwendung für das neu erzeugte Objekt verfügbar sind. Im folgenden Beispiel greifen Sie auf Eigenschaften und Methoden des neuen Objekts mit der Objektvariablen »xlBlatt« und verschiedenen Excel-Methoden, -Objekten und -Eigenschaften zu:

```
Sub ExcelAlsServer1()
'In dieser Prozedur erfüllt Excel eine Aufgabe als
'ActiveX-Server, ohne dabei sichtbar zu sein. Wenn Sie das
'Excel-Fenster sehen wollen, müssen Sie nach dem Erzeugen des
'Objekts die folgende Zeile anführen
'xlBlatt.Application.Visible = True
'Objektvariable deklarieren
Dim xlBlatt As Object
'Objekt der Variablen zuweisen
Set xlBlatt = CreateObject("Excel.sheet")
' Wert in Zelle A1 einfügen.
xlBlatt.ActiveSheet.Cells(1, 1) = 23
' Wert in Zelle A2 einfügen.
xlBlatt.ActiveSheet.Cells(2, 1).Value = 25
' Formel in Zelle A3 einfügen.
xlBlatt.ActiveSheet.Cells(3, 1).Value = "=A1+A2"
'Wert aus Zelle A3 ausgeben lassen
MsgBox xlBlatt.ActiveSheet.Cells(3, 1).Value
'Excel beenden
xlBlatt.Application.Quit
'Objektvariable freigeben.
Set xlBlatt = Nothing
End Sub
```

Das Deklarieren einer Objektvariablen mit *As Object* erstellt eine Variable, die einen Verweis auf *jeden* Objekttyp enthalten kann. Das konkrete Binden der Variable an das Objekt erfolgt jedoch erst zur Laufzeit, d.h., der Verweis wird erst beim Ausführen des Programms aufgelöst. Dieses wird auch als *late binding* bezeichnet. Sie können dagegen auch eine Objektvariable erstellen, für die der Verweis zur Kompilierungszeit aufgelöst wird (*early binding*), indem Sie die Objektvariable mit einer bestimmten *Klassen-ID* deklarieren. In Bezug auf Excel 2000 können Sie beispielsweise die folgenden Verweise deklarieren:

```
Dim xlAnw As Excel.Application
Dim xlMappe As Excel.Workbook
Dim xlBlatt As Excel.Worksheet
```

```
Set xlAnw = CreateObject("Excel.Application")
Set xlMappe = xlAnw.Workbooks.Add
Set xlBlatt = xlMappe.Worksheets(1)
```

Der Verweis mit Hilfe einer Variablen, der zur Kompilierungszeit aufgelöst wird, kann das Leistungsverhalten verbessern. Andererseits kann die Variable nur einen Verweis auf die in der Deklaration angegebene bestimmte Klasse enthalten und daher nicht so allgemein modelliert werden.

GetObject

Die Leistung von *GetObject* unterscheidet sich von derjenigen von *CreateObject* nur darin, daß Sie bei *GetObject* ein konkretes Objekt angeben können, das als Datei gespeichert ist und durch die Funktion geladen wird. Die Syntax lautet:

```
GetObject([Pfad] [,Klasse])
```

Die benannten Argumente haben diese Bedeutung:

Element	Beschreibung
Pfad	Optional. Der vollständige Pfad und Name der Datei, die das abzurufende Objekt enthält. Wird *Pfad* nicht angegeben, so ist *Klasse* erforderlich.
Klasse	Optional. Eine Zeichenfolge, die die Klasse des Objekts angibt.

Das Argument *Klasse* wird als Zeichenfolge angegeben. Es hat die Syntax *AnwName.Objekttyp* und besteht aus den folgenden Elementen:

Element	Beschreibung
AnwName	Erforderlich. Der Name der Anwendung, die das Objekt bereitstellt.
Objekttyp	Erforderlich. Der Typ oder die Klasse des zu erstellenden Objekts.

Das folgende Beispiel weist der Objektvariablen xlKMM die konkrete Arbeitsmappe *kmm.xls* als Objekt zu. Die Meldungsbox gibt den Inhalt von Zelle A1 des ersten Arbeitsblatts der Arbeitsmappe aus.

```
Dim xlKMM As Object
Set xlKMM = GetObject("d:\d\a2000p\kmm.xls")
MsgBox xlKMM.Application.Workbooks(1) _
  .Worksheets(1).Range("A1")
```

New

Die Verwendung dieses Schlüsselwortes setzt voraus, daß auf die Klassenbibliothek, in der sich das zu referenzierende Objekt befindet, ein Verweis eingestellt ist. Der Vorteil von *New* liegt in größerer Performance, weil die Auflösung des Verweises auf das Objekt bereits mit dem Kompilieren (*early binding*) erfolgt statt – wie bei den Funktionen *CreateObject* und *GetObject* – erst während der Laufzeit (*late binding*).

Sie können das Schlüsselwort *New* in einer *Dim*- oder *Static*-Deklarationsanweisung oder in einer *Set*-Anweisung anführen. Die folgenden beiden Codezeilen weisen der Objektvariablen *xlAnw* die Anwendung Excel bereits mit der Deklaration als Objekt zu und bringen Excel dann auf dem Bildschirm in den Vordergrund.

```
Static xlAnw As New Excel.Application
xlAnw.Application.Visible = True
```

Wenn Sie innerhalb einer Prozedur – also auf lokaler Ebene – statt *Static* das Schlüsselwort *Dim* verwenden, verschwindet Excel sofort wieder, weil die Variable untergeht. Sie können aber *Dim* im Deklarationsteil des Moduls verwenden, dann ist die Lebensdauer der Variablen nicht auf die Prozedur begrenzt.

Wie Sie *New* in einer *Set*-Anweisung verwenden, zeigen die beiden folgenden Zeilen, die ebenfalls die Anwendung Excel auf dem Bildschirm in den Vordergrund bringen.:

```
Set xlAnw = New Excel.Application
xlAnw.Application.Visible = True
```

33.2 Access 2000 als Client: Beispiel

Im folgenden wird der Code eines Beispiels wiedergegeben, in dem Access als Client und Excel als Server fungiert. Sie können das Beispiel auch direkt am Rechner nachvollziehen und laufen lassen, denn es ist auf der Beispiel CD-ROM in der Datenbank *AccessAlsClient.mdb* im Modul *Pivot* zu finden. Das Beispiel arbeitet folgendermaßen: Aus den Daten der Access-Abfrage mit dem Namen *UmsätzeNachPersonalLandJahr* wird eine Pivot-Tabelle in Excel erzeugt. Dies geschieht mit der Prozedur *PivotTabelleErzeugen*. Wie diese Prozedur im einzelnen funktioniert, entnehmen Sie bitte dem ausführlichen Kommentar darin.

Dieses Beispiel erzeugt die Pivot-Tabelle auf Basis von Daten, die zuvor über die Zwischenablage von der Abfrage in eine Excel-Tabelle kopiert wurden. Dieses Verfahren ist nicht besonders elegant, jedenfalls gäbe es eine elegantere Alternative: Die Pivot-Tabelle könnte in Excel auf Basis einer externen Datenquelle erstellt werden, nämlich unter direkter Bezugnahme auf die

Abfrage *AbfrageUmsätzeNachPersonalLandJahr* in der Datenbank *AccessAlsClient.mdb*. Dafür wäre es aber erforderlich, daß die Datenbank *AccessAlsClient.mdb* auf dem Rechner, auf dem die Prozedur *PivotTabelleErzeugen* laufen soll, als ODBC-Datenquelle eingerichtet und im Code angeführt würde. Dies erscheint jedoch für den Zweck dieses Beispiels und auch für andere ad hoc-Lösungen zu kompliziert, so daß sich der scheibar primitivere Weg, eine Pivot-Tabelle zu erstellen, empiehlt. Dieser hat im übrigen den Vorteil, daß Excel dabei in höherem Maße als Server für Access in Erscheinung treten muß, so daß Sie insoweit besseres Anschauungsmaterial für den aktuellen Zusammenhang haben.

Die Pivot-Tabelle, die Sie mit dem folgenden Code von der Datenbank *AccessAlsClient.mdb* aus erzeugen können, sollte so aussehen, wie in Bild 33.2 wiedergegeben.

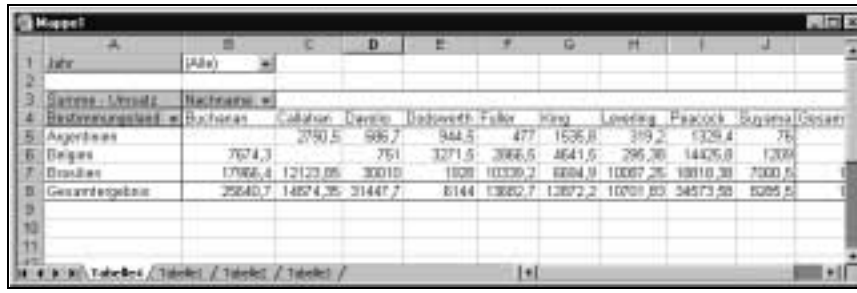


Bild 33.2: Pivot-Tabelle in Excel, die von der Prozedur »PivotTabelleErzeugen« erstellt wurde

Beispiel

```
'Objektvariable auf Modulebene deklarieren,
'damit ihre Lebensdauer größer ist als die der Prozedur
Dim xlAnw As Object

Sub PivotTabelleErzeugen()
'Diese Prozedur erzeugt für die Daten der Abfrage
'UmsätzeNachPersonalLandJahr in dieser Datenbank
'eine Pivot-Tabelle in Excel. Dazu wird die
'Abfrage in Access geöffnet, und es werden ihre
'Datensätze in die Zwischenablage kopiert.
'Dann wird eine Excel-Sitzung unter der Regie
'von Access gestartet, darin eine neue Arbeitsmappe
'erzeugt und in deren erste Tabelle der Inhalt
'der Zwischenablage kopiert. Auf dieser Datenbasis
'wird dann eine Pivot-Tabelle erstellt.

'Die folgenden drei Konstanten müssen nicht
'versorgt werden, wenn ein Verweis auf
```

```
'die Excel-Klassenbibliothek besteht
Const xlDataField = 4
Const xlDataOnly = 4
Const xlDatabase = 1

'Schaltet die Aktualisierung des Bildschirms aus
DoCmd.Echo False
'Schaltet Access-Meldungen aus. Ohne diese Anweisung
'würde eine unerwünschte Meldung beim Schließen
'der Datenblattansicht der Abfrage angezeigt werden.
DoCmd.SetWarnings False
DoCmd.OpenQuery "UmsätzeNachPersonalLandJahr"
'Markiert alle Datensätze
DoCmd.RunCommand acCmdSelectAllRecords
'Kopiert die markierten Datensätze in die Zwischenablage
DoCmd.RunCommand acCmdCopy

'Weist der Objektvariablen xlAnw die Anwendung Excel
'als Objekt zu und erzeugt somit eine Excel-Sitzung
Set xlAnw = CreateObject("Excel.Application")
'Fügt der Excel-Sitzung eine Arbeitsmappe hinzu
xlAnw.Workbooks.Add
'Bringt das Excel-Fenster in den Bildschirmvordergrund
xlAnw.Visible = True
'Fügt den Inhalt der Zwischenablage in das aktive
'Arbeitsblatt ein
xlAnw.ActiveSheet.Paste
'Schließt das Fenster mit der Abfrage in Access,
'das noch aktiv war.
DoCmd.RunCommand acCmdClose
DoCmd.Echo True
'Die folgenden drei Anweisungen erstellen eine neue
'Pivot-Tabelle mit dem aktuellen Datenblock der aktiven
'Tabelle als Datenquelle und den Feldern wie ausgewiesen
'als Zeilen-, Spalten-, Seiten- und Datenfeld
xlAnw.ActiveSheet.PivotTableWizard SourceType:=xlDatabase, _
  SourceData:=ActiveCell.CurrentRegion.Address, _
  TableDestination:="", TableName:="Pivot-Tabelle1"
xlAnw.ActiveSheet.PivotTables("Pivot-Tabelle1"). _
  AddFields RowFields:="Bestimmungsland", _
  ColumnFields:="Nachname", PageFields:="Jahr"
xlAnw.ActiveSheet.PivotTables("Pivot-Tabelle1"). _
PivotFields("Umsatz").Orientation = xlDataField
Set xlAnw = Nothing
End Sub
```

33.3 Access 2000 als Server: Beispiel

Im folgenden Beispiel zeige ich eine Anwendung, in der Word 2000 die Rolle eines ActiveX-Clients und Access 2000 die Rolle eines ActiveX-Servers spielt. Sie können das Beispiel auf dem Rechner nachvollziehen, denn die beiden dafür erforderlichen Dateien *Adressen.mdb* und *WordAlsClient.dot* befinden sich auf der Beispiel CD-ROM.

Das Beispiel löst die folgende Aufgabe: Neue Briefe werden unter Verwendung der Dokumentvorlage *WordAlsClient.dot* erstellt. Diese Vorlage bietet ein Briefblatt mit Adreßbereich, Datum, Kopf- und Fußzeile und ein paar weiteren Einzelheiten an, die Sie nutzen, wenn sie in Word ein neues Dokument unter Verwendung dieser Dokumentvorlage erstellen. Wenn sie dieses veranlaßt haben, stehen Ihnen die beiden folgenden kurzen Prozeduren als Makros zur Verfügung. (Im realistischen Fall würden sie diese Makros vermutlich jeweils einer Symbol-Schaltfläche oder einer Tastenkombination zuweisen.)

```
Dim AC1 As Object ' Variable für Verweis auf Access-Adressen

Sub DBAdressenAufrufen()
Const Path = "D:\D\A2000p\BspDsk\Adressen.mdb"
' Objektvariable so festlegen, daß sie
'auf die gewünschte Datei verweist.
Set AC1 = GetObject(Path)
AC1.Application.Visible = True
End Sub

Sub AdresseSchreiben()
Selection.GoTo What:=wdGoToBookmark, _
  Name:="Adressenanfang"
Selection.TypeText Text:= _
  AC1.Forms![Adressen (VBA-Steuerung)]!Anschrift
End Sub
```

Die Prozedur *DBAdressenAufrufen* öffnet die Datenbank *Adressen.mdb* und bringt sie auf dem Bildschirm in den Vordergrund. Wenn Sie das Beispiel auf Ihrem Rechner nachvollziehen wollen, müßten sie die Pfadangabe in der Prozedur so ändern, daß diese auf den Ordner verweist, in den Sie die Datei *Adressen.mdb* kopiert haben. Sie können dann im Formular, das durch ein AutoExec-Makro der Datenbank automatisch geöffnet wird, mit den dort zur Verfügung stehenden Werkzeugen wie Filter, Kriteriensuche oder Navigationsschaltflächen eine Adresse suchen. Wenn Sie die richtige gefunden haben, können Sie mit der Prozedur bzw. dem Makro *AdresseSchreiben* die gefundene Adresse in das Anschriftenfeld des Briefes kopieren. Dort befindet sich eine Textmarke mit dem Namen *Adressenanfang*. Die

Prozedur bringt den Cursor zunächst an diese Stelle, liest dann den Inhalt des Steuerelements *Anschrift*, das die Anschrift in der erforderlichen Form enthält, aus und schreibt diesen Inhalt an die Stelle der Textmarke. Die Situation unmittelbar nach dem Einfügen der Adresse in das Anschriftenfeld ist in Bild 33.3 wiedergegeben.

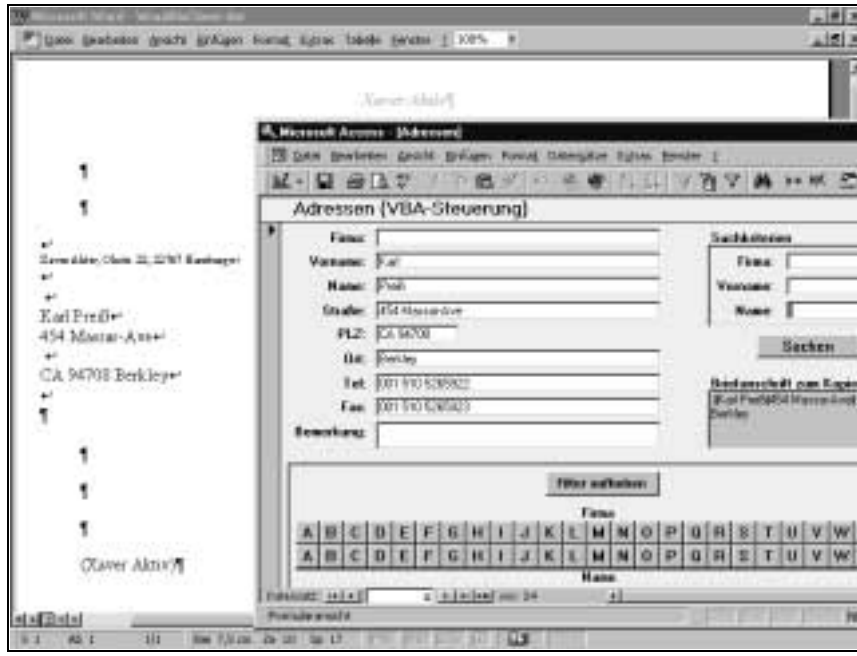


Bild 33.3: Neues Dokument auf Basis der Vorlage WordAlsClient.dot. Das Adreßformular der Datenbank »Adressen.mdb« ist eingeblendet. Seine aktuelle Anschrift wurde in den Anschriftenbereich des Briefes geschrieben.

Kapitel 34

Access im Inter- und Intranet

34.1 Überblick

Die Web-Fähigkeit aller Office-Anwendungen ist spätestens seit 1996 ein großes Ziel bei Microsofts Softwareentwicklern. Die Fähigkeit, mit Hyperlinks und dem Internet umgehen zu können, war eine der grundlegend neuen Funktionen, mit denen Access 97 und die anderen Anwendungen des Office 97-Pakets ausgestattet wurden. Diese Möglichkeiten wurden in Access 2000 im wesentlichen übernommen und im Detail erweitert. Hyperlinks können genauso wie Telefonnummern oder Adressen in Feldern von Tabellen gespeichert werden. Daneben ist es auch möglich, einen Hyperlink in einem Bezeichnungsfeld, einem Bild oder einer Befehlsschaltfläche in einem Formular oder Bericht (in diesem Falle keine Befehlsschaltfläche möglich) abzulegen. Mit Hilfe von Hyperlinks in Formularen und Datenblättern können Sie zu Objekten in derselben oder einer anderen Access-Datenbank, zu Dokumenten, die mit Word, Excel und PowerPoint erstellt wurden, und zu Dokumenten im Internet oder in einem lokalen Intranet verzweigen. Hyperlinks können auch in Berichte eingefügt werden. Wenn ein Bericht in Access angezeigt wird, funktionieren die Hyperlinks darin zwar nicht. Sie werden aber aktiv, wenn Sie den Bericht nach Word, Excel oder im HTML-Format exportieren.

Auch in der vorliegenden Version 2000 sind wiederum neue Web-Fähigkeiten hinzugekommen. Für Access 2000 besteht die Neuerung im wesentlichen in der Bereitstellung von *Datenzugriffsseiten*. Dabei handelt es sich um einen Dokumenttyp, der im Inter- und Intranet publiziert werden kann. Gegenüber den beiden verwandten Dokumenttypen *Webseite im statischen* und *Webseite im dynamischen HTML-Format* (letztere auch als *Active Server Page* bezeichnet), die bereits in der Vorgängerversion Access 97 existierten, stellen die neuen Datenzugriffsseiten folgende Veränderung dar: Eine Datenzugriffsseite enthält im allgemeinen Steuerelemente wie beispielsweise Textfelder oder Symbole zum Erweitern oder Einschränken der Darstellung von Detailfeldern oder PivotTable-Elemente, mit denen die Daten aktiv ana-

lysiert werden können. Eine Datenzugriffsseite ist direkt an die Datenquelle – d.h. eine Tabelle oder Abfrage einer Access-Datenbank oder eines Projekts – gebunden. Daher ist es unter gewissen Bedingungen auch möglich, daß die Daten dieser Datenbasis mit der Datenzugriffsseite aus dem Web heraus direkt bearbeitet werden.

Die meisten Features der Datenzugriffsseiten sind nur verfügbar, wenn auch auf der Client-Seite – also auf dem PC, auf dem die Datenzugriffsseite geladen und bearbeitet wird – die Office Web Komponenten installiert sind, was eine Office Lizenz voraussetzt. Obwohl mit großer Leistungsfähigkeit ausgestattet, sind Datenzugriffsseiten daher nicht allgemein Web-fähig. Ihr genereller Einsatz dürfte daher weitgehend auf Intranets beschränkt sein – ein Bereich, der allerdings auch nicht unterschätzt werden sollte. Trotz dieser Einschränkung erscheint es lohnend, sich mit den Datenzugriffsseiten vertraut zu machen.

34.1.1 Was ist ein Hyperlink?

Hyperlink

Ein Hyperlink ist ein farbig und unterstrichen formatierter Text oder eine Grafik, auf den oder die Sie klicken können, um zu einer Stelle in einer Datei Ihres lokalen PCs, des lokalen Intranet oder des Internet zu verzweigen. Die Stelle, zu der verzweigt werden kann, wird durch die Hyperlink-Adresse angegeben.

Hyperlink-Adresse

Eine Hyperlink-Adresse gibt den Pfad zu einem Objekt, einem Dokument, einer Web-Seite oder einem anderen Ziel an. Dies kann in der Form einer URL-Adresse (Internet-Format, z.B. als *http://www.microsoft.com/*) oder eines UNC-Netzwerkpfades (z.B. als *\\DatServ\Kursdaten\TK1\Übung3.xls*) geschehen. Eine Hyperlink-Adresse kann auch speziellere Adreßinformationen enthalten, beispielsweise die Angabe eines bestimmten Zellbereichs in einem Excel-Tabellenblatt, eine Textmarke in Word oder ein bestimmtes Objekt in Access.

Die drei Steuerelement-Typen *Befehlsschaltfläche*, *Bezeichnungsfeld* und *Bild*, die Hyperlinks enthalten können, besitzen eine Eigenschaft *Hyperlink*. Diese gibt ein Hyperlink-Objekt zurück, das den im Steuerelement enthaltenen Hyperlink darstellt. Diese Steuerelemente besitzen außerdem die Eigenschaften *HyperlinkAddress* und *HyperlinkSubAddress*, mit denen die Adresse und Unteradresse für den Hyperlink eingestellt oder zurückgegeben werden kann.

Wenn Sie auf ein Steuerelement klicken, das einen Hyperlink enthält, wird dem Hyperlink gefolgt, und das Ereignis *Click* des Steuerelements wird ausgelöst. Beim Folgen des Hyperlinks wird das in den Eigenschaften *Hyperlin-*

kAddress und *HyperlinkSubAddress* bezeichnete Dokument oder Objekt geöffnet. Bezieht sich die Eigenschaft *HyperlinkAddress* beispielsweise auf ein Formular in Ihrer Access-Datenbank, so wird dieses Formular geöffnet. Bezieht sie sich z.B. auf ein Word-Dokument auf einem anderen Computer im Netzwerk, so wird Word mit dem betreffenden Dokument geöffnet. Bezieht sie sich auf eine Adresse im World Wide Web, so wird Ihr Web-Browser (z.B. der Microsoft Internet Explorer) geöffnet und ggf. die Internet-Verbindung hergestellt. Dann wird die Web Site angezeigt.

In einer VBA-Prozedur können Sie das Klicken auf einen Hyperlink modellieren, indem Sie die Methode *Follow* des Objekts *Hyperlink* oder die Methode *FollowHyperlink* des Objekts *Application* verwenden. Beispielsweise können Sie im Ereignis *Load* eines Formulars Codezeilen anführen, die dafür sorgen, daß einem angegebenen Hyperlink gefolgt wird, wenn das Formular geöffnet wird.

34.1.2 Darstellung von Hyperlinks einstellen

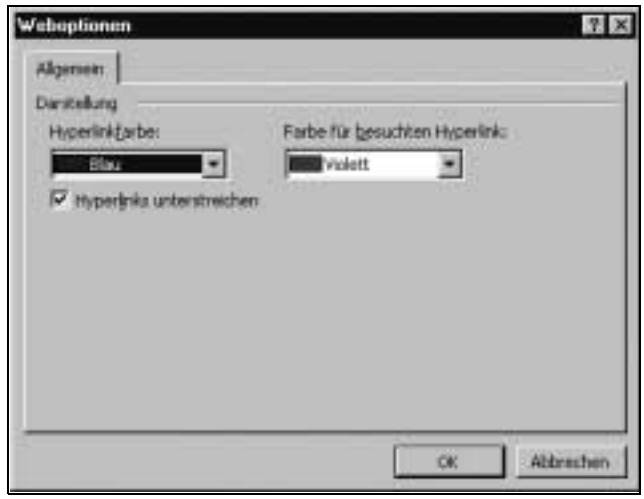


Bild 34.1: Registerkarte »Hyperlinks/HTML« des Dialogfeldes »Optionen«

Das Aussehen der Hyperlink-Wiedergabe läßt sich allgemein für die Datenbank im Dialogfeld *Weboptionen* (vgl. Bild 34.1) bestimmen:

- Wählen Sie im Menü *Extras* den Befehl *Optionen* und aktivieren Sie die Registerkarte *Allgemein*.
- Klicken Sie auf die Schaltfläche *Weboptionen*.

34.2 Hyperlinks

34.2.1 Ausführen eines Hyperlinks in einem Datenblatt oder Formular

Um einen bestehenden Hyperlink auszuführen, d.h., zu dem mit ihm verbundenen Objekt zu verzweigen, gehen Sie folgendermaßen vor:

- Öffnen Sie die Tabelle, die Abfrage oder das Formular in der Datenblattansicht, das Formular in der Formularansicht oder die Seite in der Seitenansicht. Der Hyperlink kann sich in einem Feld eines Datenblatts oder in einem Bezeichnungsfeld, Textfeld, Kombinationsfeld, auf einer Befehlsschaltfläche oder in einem Bild-Steuerelement eines Formulars befinden.
- Fahren Sie den Mauszeiger auf das Objekt mit dem Hyperlink. Wenn sich der Zeiger in eine Hand verwandelt, klicken Sie auf den Hyperlink.

Bevor Sie auf einen Hyperlink klicken, ist dieser standardmäßig blau und unterstrichen formatiert. Nach dem Klicken ist er standardmäßig lila und unterstrichen formatiert. Diese Formatierung können Sie ändern, vgl. den voranstehenden Punkt.

34.2.2 Erstellen eines Tabellenfeldes zum Speichern von Hyperlinks

Sie erstellen ein derartiges Feld im Prinzip genauso, wie Sie jedes andere Tabellenfeld erstellen, nur daß Sie in diesem Falle den Felddatentyp *Hyperlink* vergeben, vgl. dazu im einzelnen Kap. 8, *Tabellen: Grundlagen*, Punkte 8.3, *Felder hinzufügen, löschen, kopieren, verschieben*, und 8.4, *Felddatentypen festlegen*. Verfahren Sie prinzipiell folgendermaßen:

- Öffnen Sie eine vorhandene Tabelle in der Entwurfsansicht, oder erstellen Sie eine neue Tabelle.
- Legen Sie einen Feldnamen für das Feld, das Hyperlinks speichern soll, fest.
- Legen Sie den Felddatentyp *Hyperlink* für dieses Feld fest.
- Speichern Sie die Tabelle.

34.2.3 Steuerelement mit Hyperlink in Formular oder Bericht erstellen

Gebundenes Steuerelement

Wenn Sie ein Steuerelement in ein Formular oder einen Bericht einfügen wollen, das an ein Tabellenfeld mit dem Datentyp *Hyperlink* gebunden ist, wählen Sie eines der üblichen Verfahren zum Einfügen gebundener Steuerelemente. Sie können z.B. den Formular-Assistenten ein Formular erstellen lassen, in ein bestehendes Formular das Steuerelement durch Ziehen aus der *Feldliste* einfügen oder die *Toolbox* einsetzen und die Eigenschaft *Steuerelementinhalt* auf das Feld mit dem Hyperlink einstellen. In jedem Falle wird das betreffende Steuerelement in der Formularansicht ausgelöst oder bearbeitet werden können.

Ungebundenes Steuerelement

Sie können die drei Steuerelemente *Bezeichnungsfeld*, *Bild* und *Befehlsschaltfläche* als ungebundene Steuerelemente in ein Formular einfügen und sie mit einem Hyperlink versehen. Für alle drei Steuerelemente können sie folgendermaßen vorgehen:

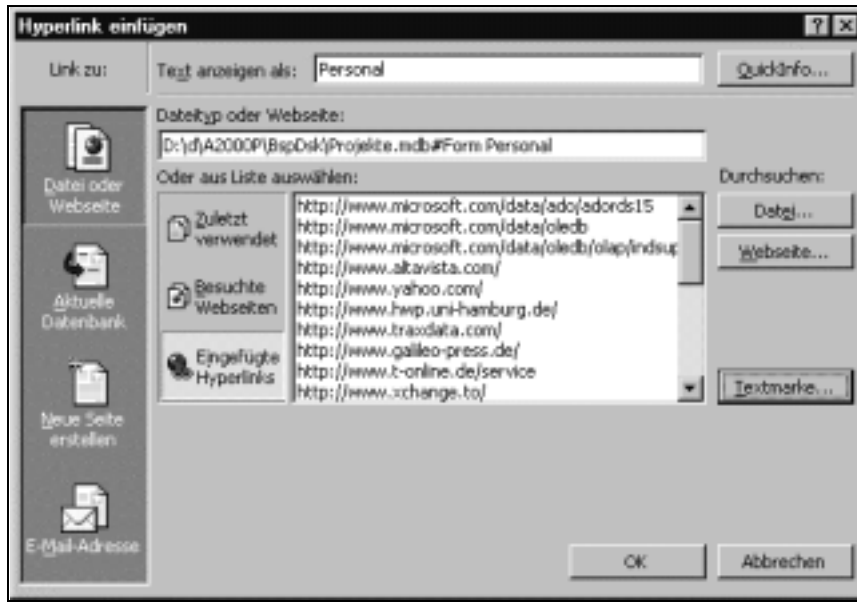


Bild 34.2: Dialogfeld »Hyperlink einfügen«

- Fügen Sie das Steuerelement auf eine beliebige Weise ein, z.B. mit der Toolbox.
- Stellen sie die Eigenschaft *Hyperlink-Adresse* auf die gewünschte Adresse im URL- oder UNC-Format ein. Eine URL-Adresse könnte z.B. lauten: *http://www.hwp.uni-hamburg.de*. Sie können auch den *Editor* für diese Eigenschaft öffnen, indem Sie auf die Schaltfläche mit den drei Punkten klicken. Dann wird das Dialogfeld *Hyperlink einfügen* angezeigt, vgl. Bild 34.2. Mit diesem Dialogfeld können Sie mit Hilfe der Schaltflächen *Durchsuchen* und der Dropdown-Listen Hyperlink-Adressen auf bequeme und sichere Weise einfügen.
- Stellen Sie ggf. auch eine Adresse für die Eigenschaft *Hyperlink-Unteradresse* ein. Wenn Sie mit dem Dialogfeld *Hyperlink einfügen* arbeiten, können Sie dort durch Klicken auf die Schaltfläche *Textmarke* die entsprechende Unteradresse einfügen lassen. Im Dialogfeld von Bild 34.2 beispielsweise wurde zunächst ein Link zur Datenbank *Projekte.mdb* erstellt. Wenn Sie danach auf die Schaltfläche *Textmarke* klicken, werden in einem weiteren Dialogfeld die Objekte der Datenbank *Projekte.mdb*, zu denen

ein Link möglich ist, angeboten. Wenn Sie darunter eines auswählen, wird dies als Unteradresse in die entsprechende Eigenschaft eingetragen, vgl. Bild 34.3.

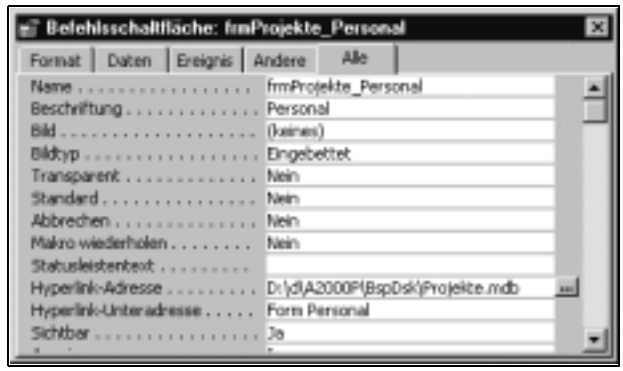


Bild 34.3: Eigenschaftenfenster mit Einträgen zu den beiden Hyperlink-Eigenschaften, wie sie vom Dialogfeld »Hyperlink einfügen« in Bild 34.2 eingetragen wurden.

Wenn Sie das Formular in der Formularansicht öffnen, wird das Steuerelement als Hyperlink aktiv sein, d.h., wenn Sie darauf klicken, wird der Hyperlink ausgeführt, indem zu der angegebenen Adresse verzweigt wird.

Bezeichnungsfeld besonders komfortabel einfügen

Wenn Sie ein Bezeichnungsfeld mit einem Hyperlink einfügen wollen, können Sie von vornherein das Dialogfeld *Hyperlink einfügen* öffnen, vgl. Bild 34.2, und gleichzeitig ein Bezeichnungsfeld einfügen. Gehen Sie folgendermaßen vor:

- Klicken Sie auf die Symbol-Schaltfläche *Hyperlink einfügen* (vgl. links nebenstehend), oder wählen Sie den Befehl *Hyperlink* aus dem Menü *Einfügen*. Dann wird das Dialogfeld *Hyperlink einfügen* angezeigt, vgl. Bild 34.2.
- Wählen Sie eine Adresse aus und bestätigen Sie mit OK. Dann wird ein Bezeichnungsfeld in das Formular eingefügt, dessen Eigenschaft *Hyperlink-Adresse* (und ggf. *Hyperlink-Unteradresse*) auf die von Ihnen gewählte Adresse eingestellt ist.
- Verschieben Sie das Bezeichnungsfeld an eine geeignete Position, denn es wird bei diesem Verfahren stets in der linken oberen Ecke des Detailbereichs eingefügt.

Sie können das eingefügte Bezeichnungsfeld anschließend in ein Bild oder eine Befehlsschaltfläche umwandeln: Markieren Sie es, und wählen Sie dann aus dem Menü *Format* den Befehl *Ändern zu*.

34.3 Eingeben einer Hyperlink-Adresse in der Formular- oder Datenblattansicht

Um für einen neuen Datensatz eine Hyperlink-Adresse einzugeben, können Sie diese direkt in das Feld oder Steuerelement der Tabelle oder des Formulars in der Datenblattansicht oder in Formularansicht eingeben. Sie können sich dazu aber auch des Dialogfeldes *Hyperlink einfügen* bedienen, vgl. Bild 34.2. Zu diesem Zweck klicken Sie, während das betreffende Feld den Fokus hat, auf die Symbol-Schaltfläche *Hyperlink einfügen*, vgl. den vorigen Punkt. Dann verfahren sie entsprechend, wie im vorigen Punkt beschrieben.

34.4 Bearbeiten eines Hyperlinks

Ein Steuerelement oder Feld mit einem Hyperlink können Sie nicht in der üblichen Art, die für andere Felder oder Steuerelemente gilt, bearbeiten, nicht zuletzt deshalb, weil Sie Felder und Steuerelemente mit Hyperlinks nicht mit der Maus markieren können: Sobald Sie auf ein derartiges Element klicken, wird der Hyperlink ausgeführt. Verfahren Sie daher zum Bearbeiten eines Hyperlinks wie folgt:

- Für ein gebundenes Feld oder Steuerelement blättern sie in der Datenblatt- oder Formularansicht zum gewünschten Datensatz. Für ein ungebundenes Steuerelement wechseln sie in die Entwurfsansicht des Formulars oder Berichts.
- Klicken Sie mit der rechten Maustaste auf das Feld oder Steuerelement, um sein Kontextmenü zu öffnen.
- Wählen sie den Befehl *Hyperlink* und dessen Unterbefehl *Hyperlink bearbeiten*. Dann wird das Dialogfeld *Hyperlink bearbeiten angezeigt, das mit dem Dialogfeld Hyperlink einfügen (vgl. oben* Bild 34.2) fast identisch ist. Nehmen Sie die erforderlichen Bearbeitungen, die auch das Entfernen der Verknüpfung einschließen können, vor, und bestätigen Sie mit OK.

34.5 Hyperlink-Basis für eine Datenbank festlegen

Standardmäßig werden Hyperlink-Adressen als absolute Pfade angegeben, als URL-Pfad z.B. in der Form *http://www.hwp.uni-hamburg.de* oder als UNC-Pfad in der Form *C:\Publikum\Geschäftsbericht1996.doc*. Access erlaubt es, statt dessen einen relativen Pfad als Hyperlink-Adresse anzugeben und ggf. zu speichern. Ein relativer Pfad setzt voraus, daß ein Basis-Pfad vereinbart ist, der von Access als Hyperlink-Basis bezeichnet wird. Die Adresse *C:\Publikum\Geschäftsbericht1996.doc* beispielsweise können Sie sich als aus einem absoluten und einem relativen Teil zusammengesetzt vorstellen: *C:\Publikum* könnte der absolute und *Geschäftsbericht1996.doc* der relative Teil sein. Wenn Sie in der Datenbank *C:\Publikum* als Hyperlink-Basis festlegen, können Sie die Hyperlink-Adresse, die auf die Datei

Geschäftsbericht1996.doc verweisen soll, als relative Adresse angeben. Dann brauchen Sie als Hyperlink-Adresse nur die Zeichenfolge *Geschäftsbericht1996.doc* zu speichern, und Access ergänzt automatisch beim Verfolgen des Hyperlinks um den als Hyperlink-Basis festgelegten Adreßteil *C:\Publikum*. Diese Verfahrensweise kann vorteilhaft sein, wenn Sie in einer Datenbank mehrere Hyperlink-Adressen angeben, die einen gemeinsamen Basispfad haben. Sie sparen sich dann nicht nur Arbeit bei der Dateneingabe, sondern Sie können auch leichter den Fall regeln, daß die relativ angegebenen Adressen nachträglich einem anderen Basispfad zugeordnet werden, weil beispielsweise Dateien in einen anderen Ordner oder auf ein anderes Laufwerk verschoben wurden. In diesem Falle brauchen sie lediglich den neuen Ordner als neue Hyperlink-Basis anzugeben, statt sämtliche einzelnen Adressen zu bearbeiten.

Um eine Hyperlink-Basis festzulegen, gehen Sie folgendermaßen vor:

- Wählen sie im Menü *Datei* den Befehl *Datenbankeigenschaften*. Dann wird das Dialogfeld *(Datenbankname) Eigenschaften* angezeigt.
- Aktivieren sie die Registerkarte *Zusammenfassung*, und geben Sie in das Einstellungsfeld *Hyperlink-Basis* den linken Teil der Hyperlink-Adressen an, den viele Adressen gemeinsam haben. Für das im letzten Absatz angeführte Beispiel würden sie *C:\Publikum* angeben. Schließen sie das Dialogfeld mit OK.

Von jetzt an werden Hyperlink-Adressen, wenn möglich, relativ eingefügt.

34.6 Drei Arten von Webseiten in Access 2000 – Überblick

In Access 2000 können Sie drei verschiedene Arten von Webseiten – das sind Dokumente im HTML-Format – erstellen:

- Datenzugriffsseiten
- Active Server Pages
- Statischer HTML-Code

Datenzugriffsseiten. Sie können eine Datenzugriffsseite als ein Datenbankobjekt erstellen, das eine Verknüpfung mit dem Pfad der entsprechenden HTML-Datei der Seite enthält. Datenzugriffsseiten können mit dem Internet Explorer 5.x aktuelle Daten aus einer Microsoft Access-Datenbank oder einem SQL-Server-basierten Projekt betrachtet und bearbeitet werden. Eine Seite kann außerdem zusätzliche Steuerelemente enthalten, unter anderem Tabellen, PivotTable-Listen oder Diagrammme. Der Internet Explorer muß die Seite nur einmal von der Quelle oder vom Webserver übertragen, damit der Benutzer sie anzeigen und mit den Daten der Seite arbeiten kann. Da eine Datenzugriffsseite Dynamic HTML (XML) verwendet, ist in einer Client-/Server-Umgebung der Zugriff auf die Datenbank in der Regel sehr effizient.

Active Server Pages. Sie können serverseitig generierte HTML-Dateien (entweder ASP- oder IDC/HTX-Dateien) aus Tabellen, Abfragen und Formularen erstellen. Diese Seiten werden auch als *Active Server Pages* (ASP) bezeichnet. Serverseitig generierte HTML-Dateien werden in einem Webbrowser in Tabellenform dargestellt. Verwenden Sie serverseitig generierte HTML-Dateien, wenn der Benutzer unabhängig von proprietärer Microsoft-Technologie arbeitet und beliebige Webbrowser einsetzen möchte und sich Ihre Daten häufig ändern. Jedesmal, wenn ein Benutzer eine ASP- oder HTX-Datei mit einem Webbrowser öffnet bzw. aktualisiert, erstellt der Webserver dynamisch eine HTML-Datei und sendet dann diese HTML-Datei an den Webbrowser. Dadurch sind die Daten stets in der aktuellsten Form. Der Einsatz dieser Seiten setzt jedoch den Microsoft Internet Information Server voraus.

Statischer HTML-Code. Sie können statische HTML-Dateien aus Tabellen, Abfragen, Formularen und Berichten erstellen. Webbrowser stellen Berichte in einer speziellen Berichtsform dar, während Tabellen, Abfragen sowie Formulare in Form eines Datenblatts angezeigt werden. Verwenden Sie statische HTML-Dateien, wenn Sie beliebige Webbrowser mit Unterstützung für HTML Version 3.2 oder später verwenden möchten und wenn Ihre Daten nur selten geändert werden. Beim Zugriff auf die Daten über einen Webbrowser muß der Browser die statische HTML-Datei nur einmal vom Webserver übertragen, damit der Benutzer sie anzeigen kann. Allerdings zeigen diese HTML-Dateien die Daten nur in dem Zustand, in dem sie sich zum Zeitpunkt ihrer Generierung befanden. Es gibt also, anders als bei Datenzugriffsseiten und Active Server Pages, keine mit der statischen HTML-Datei verknüpfte Datenquelle. Wenn sich die Daten ändern, müssen Sie daher Ihre Dateien erneut exportieren, damit Benutzer in der Lage sind, die neuen Daten in einem Browser anzuzeigen.

34.7 Ex- und Importieren eines Access-Objekts im statischen HTML-Format oder als Active Server Page

34.7.1 Exportieren im statischen HTML-Format

Gehen Sie folgendermaßen vor:

- Aktivieren Sie das zu exportierende Objekt (Tabelle, Abfrage, Formular oder Bericht). Es reicht für diesen Zweck, wenn Sie das Objekt im Datenbankfenster markieren.
- Wählen Sie aus dem Menü *Datei* den Befehl *Exportieren*.
- Wählen Sie im Dialogfeld *Exportieren von (Objektname) nach* in der Liste *Dateityp* den Eintrag *HTML-Dokumente (*.html, *.htm)*, geben Sie den Dateinamen an und bestimmen Sie den Ordner, unter dem das Dokument gespeichert werden soll. Bestätigen Sie schließlich mit *Speichern*.

34.7.2 Exportieren als Active Server Page

Eine Active Server Page wird an eine ODBC-Datenquelle gebunden. Wenn Sie beispielsweise eine Active Server Page erstellen wollen, die aus einem Objekt der Datenbank *Nordwind.mdb* erzeugt wird, müssen Sie zuvor eine ODBC-Datenquelle für die Datenbank *Nordwind.mdb* eingerichtet haben. Um eine ODBC-Datenquelle einzurichten, verfahren Sie wie folgt:

- Klicken Sie in der Task-Leiste auf das Menü *Start*, wählen Sie den Befehl *Einstellungen* und in dessen Untermenü Systemsteuerung.
- Doppelklicken Sie im Fenster *Systemsteuerung* auf das Symbol ODBC-Datenquellen, um das entsprechende Dienstprogramm zu starten.
- Richten Sie die neue Datenquelle ein. Der Name für die Datenquelle, den Sie hier vergeben, wird beim Exportieren der Active Server Page benötigt.

Nachdem eine ODBC-Datenquelle für die Datenbank eingerichtet wurde, verfahren sie ähnlich wie im letzten Punkt für den Export im statischen HTML-Format beschrieben:

- Aktivieren Sie das zu exportierende Objekt (Tabelle, Abfrage, Formular oder Bericht). Es reicht für diesen Zweck, wenn Sie das Objekt im Datenbankfenster markieren.
- Wählen Sie aus dem Menü *Datei* den Befehl *Exportieren*.
- Wählen Sie im Dialogfeld *Exportieren von (Objektname) nach* in der Liste *Dateityp* den Eintrag *Microsoft Active Server Pages (*.asp)*, geben Sie den Dateinamen an und bestimmen Sie den Ordner, unter dem das Dokument gespeichert werden soll. Bestätigen Sie mit *Speichern*. Dann öffnet sich das Dialogfeld *Microsoft Active Server Pages-Ausgabeoptionen*, vgl. Bild 34.4.
- Geben Sie mindestens den Namen der Datenquelle an, in Bild 34.4 ist dies der Name *Nordwind*. Die weiteren Optionen sind wahlfrei. Bestätigen Sie das Dialogfeld mit OK. Daraufhin wird die .asp-Datei erzeugt und gespeichert.

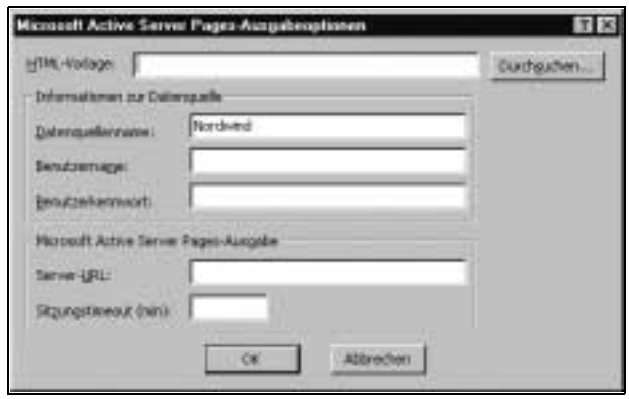


Bild 34.4: Dialogfeld »Microsoft Active Server Pages-Ausgabeoptionen«

34.7.3 Importieren oder Verknüpfen von HTML-Tabellen und -Listen

Sie können HTML-Tabellen und -Listen importieren oder verknüpfen. Auf eine verknüpfte HTML-Tabelle kann allerdings nur lesend zugegriffen werden. Zum Importieren wählen Sie im Menü Datei den Befehl *Externe Daten*, Unterbefehl *Importieren*, zum Verknüpfen den Unterbefehl *Tabellen verknüpfen*. Wählen Sie im folgenden Dialogfeld den Dateityp *HTML-Dokumente (*.html;*.htm), wählen sie die zu importierende oder zu verknüpfende Datei, und bestätigen Sie das Dialogfeld mit der Schaltfläche Importieren bzw. Verknüpfen. Das Importieren und Verknüpfen von Dateien ist im übrigen ausführlich in Kap. 9, Tabellen importieren, exportieren und verknüpfen*, erklärt.

34.8 Datenzugriffsseiten

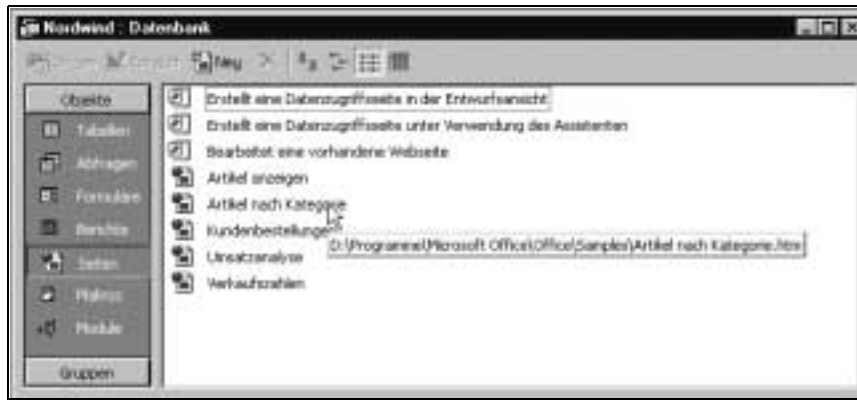


Bild 34.5: Objektliste »Seiten« in der Nordwind-Datenbank

Gegenüber den in den vorangehenden Punkten besprochenen beiden Arten von HTML-Dokumenten stellen Datenzugriffsseiten (amerikanisch *Data Access Pages* (DAP)) ein völlig neues Konzept dar: Obwohl Datenzugriffsseiten ebenfalls dynamisch sind – es besteht eine Verbindung zur Datenquelle, so daß die Daten aktualisiert und in gewissem Kontext sogar bearbeitet werden können –, kommen sie ohne den Microsoft Internet Information Server aus. Allerdings bedarf es des Microsoft Internet Explorer 5.x und der Office Web Komponenten auf der Clientseite. Datenzugriffsseiten können aber nicht nur mit dem Internet Explorer gehandhabt werden, sondern auch in Access 2000 selbst: Sie werden als Objekte der Datenbank ausgewiesen, in der sie erzeugt wurden, obgleich sie als eigenständige .htm-Dateien gespeichert sind. Im Datenbankfenster erscheinen sie in der Objektliste *Seiten*, vgl. Bild 34.5.

Wenn Sie mit dem Mauszeiger auf ein Seitenobjekt in der Objektliste *Seiten* zeigen, wird eine Quickinfo eingeblendet, die den Namen und Pfad der zugehörigen .htm-Datei wiedergibt, vgl. Bild 34.5, wo der Mauszeiger auf die Seite *Artikel nach Kategorie* zeigt, deren Pfad in der QuickInfo angezeigt wird.

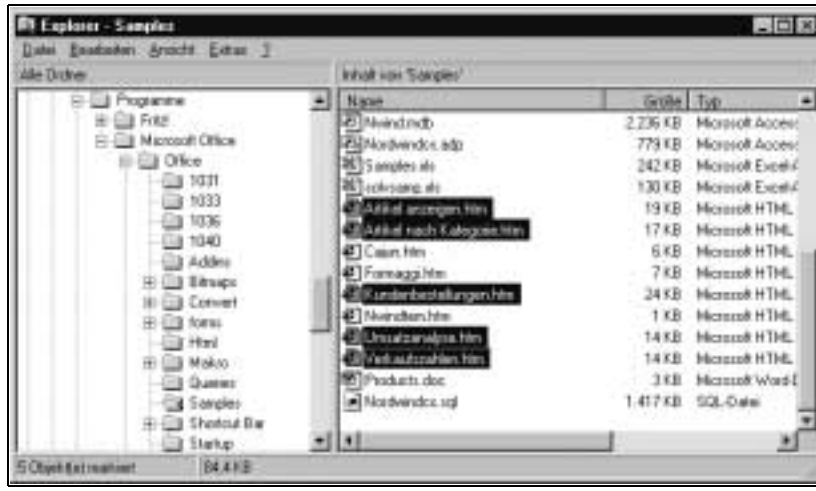


Bild 34.6: Der Explorer weist die Datenzugriffsseiten der Objektliste aus Bild 34.5 im Samples-Ordner als Dateien im .htm-Format aus.

34.8.1 Beispielseiten der Nordwind-Datenbank

Die Leistungsfähigkeit von Datenzugriffsseiten kann am besten an fertigen Beispielen demonstriert werden. Die Datenbank *Nordwind.mdb* enthält fünf verschiedene Datenzugriffsseiten, vgl. die obige Objektliste im Datenbankfenster aus Bild 34.5.

Die Datenzugriffsseiten der Datenbank *Nordwind.mdb* können, wie alle anderen Datenzugriffsseiten, auch mit dem Microsoft Internet Explorer 5.x geöffnet und dort analysiert und ggf. datenmäßig bearbeitet werden. Im folgenden zeige ich jedoch die Darstellung in Access, weil dies hier die relevante Betrachtungsebene ist: Außer der Darstellung fertiger Datenzugriffsseiten geht es auch um deren Generierung und Bearbeitung in der Entwurfsansicht, was nur in Access und nicht mit dem Internet Explorer möglich ist.

Artikel nach Kategorie. Öffnen Sie in der Datenbank *Nordwind.mdb* die Datenzugriffsseite *Artikel nach Kategorie*, und analysieren Sie deren Daten. Verfahren Sie wie folgt:

- Aktivieren Sie im Datenbankfenster die Objektliste *Seiten*.
- Doppelklicken Sie auf das Objekt *Artikel nach Kategorie*. Dann wird die betreffende Datenzugriffsseite angezeigt, wie in Bild 34.7 wiedergegeben.

Bild 34.7: Geöffnete Datenzugriffsseite »Artikel nach Kategorie« der Datenbank »Nordwind.mdb«.

Die Seite *Artikel nach Kategorie* gibt unmittelbar nach der Öffnung die ersten drei von insgesamt acht Artikelkategorien wieder. Um zu anderen Kategorien zu blättern, können Sie mit den Navigationssymbolen auf der unten positionierten Navigationsleiste arbeiten. Beachten Sie auch die dort ebenfalls vorhandenen Symbole zum Sortieren und Filtern.

Bild 34.8: Die zur Kategorie »Getränke« gehörenden Artikel werden angezeigt, nachdem auf das zugehörige Erweiterungssymbol geklickt wurde.

Links neben jeder Kategorie befindet sich jeweils ein Erweiterungssymbol. Wenn Sie darauf klicken, werden die zur betreffenden Kategorie gehörigen Artikel wiedergegeben, vgl. Bild 34.8, wo dies für die Kategorie *Getränke* geschehen ist: Für die Kategorie *Getränke* wird eine Art Unterdatenblatt angezeigt, dem selbst wiederum eine Navigationsleiste zugeordnet ist, aus der sich erkennen läßt, daß dieser Kategorie zwölf Artikel zugehören, zu denen auch geblättert werden kann.

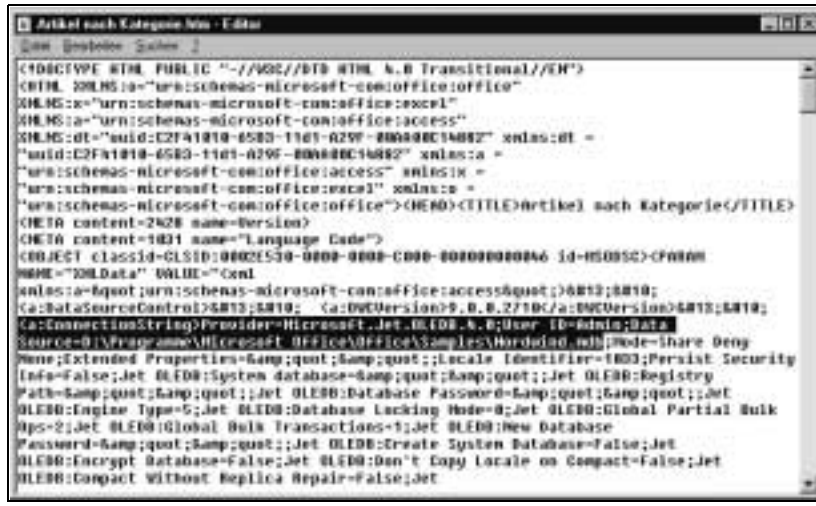


Bild 34.9: HTML-Quellcode der Datenzugriffsseite Artikel nach Kategorie in »Notepad«. Markiert ist der Beginn des »Connection String«.

Interessant erscheint es auch, diese Datenzugriffsseite einmal im HTML-Quellcode zu betrachten. In Bild 34.9 ist der HTML-Code in der Darstellung von *Notepad* wiedergegeben (die Betrachtung in Access 2000 mit Menü *Ansicht*, Befehl *HTML-Quelle* wäre ebenfalls möglich, erlaubt aber keinen Zeilenumbruch). Es ist klar, daß dieser Code nur in Ausnahmefällen direkt bearbeitet werden sollte – weiter unten wird gezeigt, wie Sie Datenzugriffsseiten in Access 2000 in der Entwurfsansicht quasi wie ein Formular bearbeiten können. Hier geht es mir um einen anderen Punkt: In Bild 34.9 ist der Teil des Codes, der den Beginn des *ConnectionString* angibt, markiert. Sie finden dort dieselben Angaben, die auch in ADO zum Öffnen eines *Connection*-Objekts erforderlich sind[1]. Als OLE DB-Provider ist *Microsoft.Jet.OLEDB.4.0* angegeben, und die *Data Source* ist *D:\Programme\Microsoft Office\Office\Samples\Nordwind.mdb*. Diese letztere Information zeigt deutlich, daß das Dokument selbst mit der Daten-

1. Das ADO-Konzept wird weiter oben in Kap. 31, *Datenzugriff mit DAOL (Data Access Objects) und ADO (Active X Data Objects)*, Punkt 31.5, Datenzugriff mit DAO, behandelt.

quelle verbunden ist, weshalb auch kein Umweg über den Internet Information Server erforderlich ist. Die Verbindung zur Datenquelle wird vielmehr über ein ADO-Recordset hergestellt, wofür u.a. die angeführte Datenquelle eine notwendige Information darstellt. Falls Sie einmal die Datenbank in einen anderen Ordner verlegt haben, müssen Sie diese Information ändern. Dies können Sie zwar auch mit einem bequemeren Hilfsmittel innerhalb von Access 2000 erledigen (vgl. weiter unten Punkt 34.8.6, Name oder Pfad der Datenbank wurden geändert). Wenn Ihnen dieses jedoch aus irgendeinem Grunde nicht zur Verfügung steht, können Sie den HTML-Code direkt an dieser Stelle mit *Notepad* ändern.

Kundenbestellungen. Öffnen Sie in der Datenbank *Nordwind.mdb* die Datenzugriffsseite *Kundenbestellungen*, und analysieren Sie deren Daten. Verfahren Sie dazu entsprechend wie im vorangehenden Beispiel der Seite *Artikel nach Kategorie*.

Bild 34.10: Datenzugriffsseite »Kundenbestellungen« unmittelbar nach dem Öffnen, jedoch mit aufgeschlagener Dropdown-Liste zum Auswählen einer Firma

Die Seite *Kundenbestellungen* weist unmittelbar nach dem Öffnen lediglich eine Dropdown-Liste zum Auswählen einer Firma auf. Wenn Sie darin eine Firma wählen, werden die Bestellungen der Firma in Fünfer-Gruppen angezeigt, wobei jede Bestellung ein Erweiterungssymbol besitzt, dessen Klicken die zugehörigen Bestelldetails wiedergibt, vgl. Bild 34.11, in dem die Bestelldetails für die erste Bestellung der Firma *Around the Horn* angezeigt werden.

Interessant erscheint die Betrachtung der Seite *Kundenbestellungen* in der Entwurfsansicht. Um eine Datenzugriffsseite in der Entwurfsansicht darzustellen und dort ggf. zu bearbeiten, gehen Sie ganz entsprechend wie bei einem Formular oder Bericht vor, also folgendermaßen:

Bild 34.11: Hier wurde die Firma »Around the Horn« ausgewählt, und die Bestelldetails der ersten Bestellung werden angezeigt.

- Bei in der Seitenansicht geöffneter Datenzugriffsseite klicken Sie auf das Symbol *Entwurf*.
- Bei ungeöffneter Datenzugriffsseite markieren Sie diese in der Objektliste im Datenbankfenster und klicken dann auf das Symbol *Entwurf*.

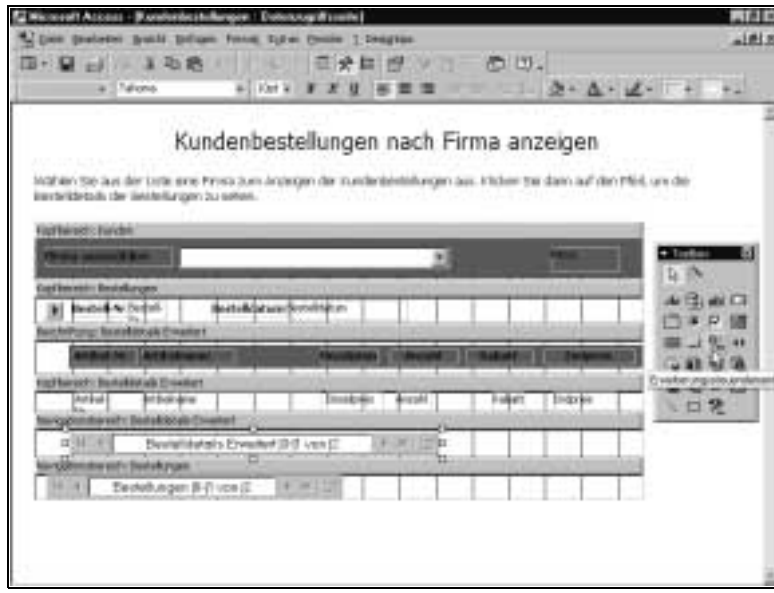


Bild 34.12: Datenzugriffsseite »Kundenbestellungen« in der Entwurfsansicht

Die Seitenansicht in Bild 34.12 ähnelt sehr derjenigen eines Berichts: Es sind drei Kopfbereiche zu erkennen, ein Beschriftungsbereich und zwei Navigationsbereiche. Die Kopfbereiche haben offensichtlich die Funktion, die Daten der zugrundeliegenden Datenquelle zu gruppieren. Dies zeigt sich auch, wenn – wiederum entsprechend wie bei einem Bericht – das Dialogfeld *Sortieren und gruppieren* angezeigt wird, vgl. Bild 34.13.

Bild 34.13: Dialogfeld »Sortieren und gruppieren« für die Datenzugriffsseite »Kundenbestellungen«

Das Dialogfeld *Sortieren und gruppieren* läßt erkennen, daß die Datenzugriffsseite *Kundenbestellungen* drei Gruppierungsebenen aufweist, deren Namen genau den drei Kopfbereichen entsprechen, vgl. die Entwurfsansicht in Bild 34.12. In der Entwurfsansicht in Bild 34.12 ist auch die Toolbox eingeblendet. Diese enthält ähnliche Steuerelemente wie für ein Formular und einen Bericht, jedoch zusätzlich weitere, die nur in Datenzugriffsseiten verfügbar sind, beispielsweise das *Erweiterungssteuerelement*, auf das in Bild 34.12 der Mauszeiger gerade zeigt, so daß die zugehörige QuickInfo angezeigt wird.

Verkaufszahlen. Als letzte Leistungsdemonstration der Datenzugriffsseiten soll die Seite Verkaufszahlen geöffnet und betrachtet werden. Verfahren Sie dazu entsprechend wie in den vorangehenden beiden Beispielen. Bild 34.14 gibt diese Seite wieder.

Die Seite *Verkaufszahlen* unterscheidet sich von den beiden vorangehenden Beispielen vor allem dadurch, daß sie eine PivotTable und eine Grafik enthält. Letztere ist auf der PivotTable aufgebaut und gibt deren Zahlen dynamisch wieder: Die PivotTable ist interaktiv, so daß Sie beispielsweise eine andere Artikelkategorie im sogen. Seitenfeld links oben in der PivotTable wählen können, mit der Folge, daß dann auf eine andere Kategorie gefiltert und sowohl die PivotTable wie auch die Grafik neu aufgebaut wird.

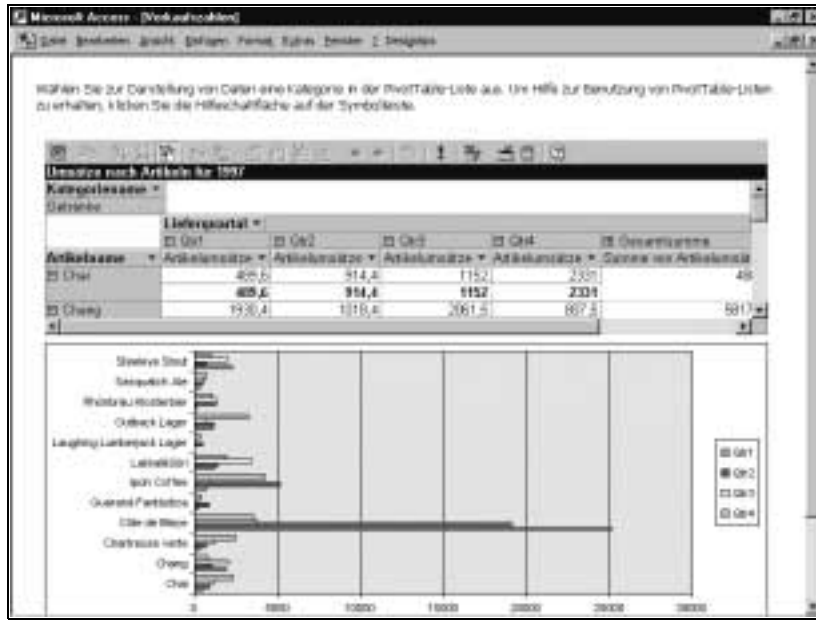


Bild 34.14: Datenzugriffsseite »Verkaufszahlen«, unverändert nach dem Öffnen

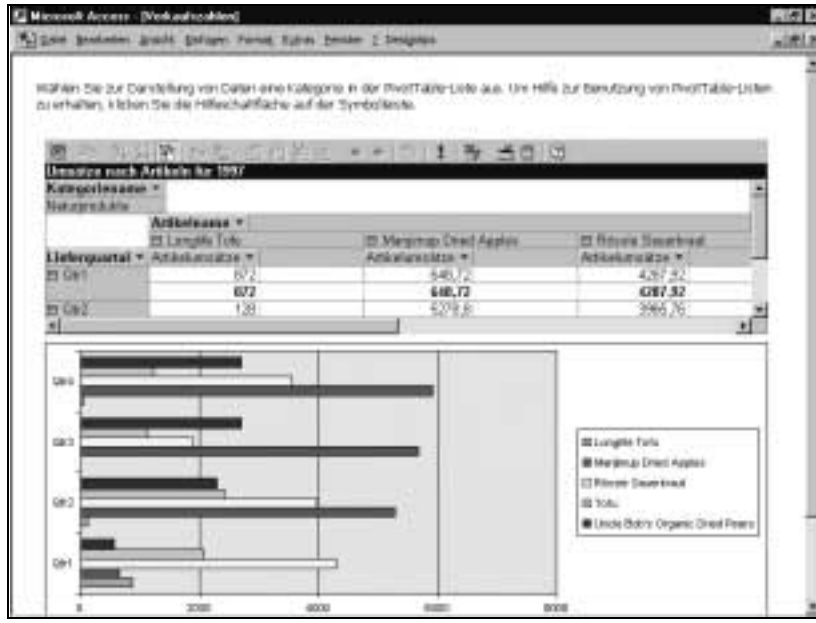


Bild 34.15: Datenzugriffsseite »Verkaufszahlen« mit geänderter PivotTable: Spalten- und Zeilenfeld wurden vertauscht, und es wird nunmehr auf die Kategorie »Naturprodukte« gefiltert.

Bild 34.15 zeigt die Seite *Verkaufszahlen* mit einer veränderten PivotTable: Durch Ziehen mit der Maus (Drag & Drop) wurden Spalten- und Zeilenfeld (d.h. die Dimensionen *Lieferquartal* und *Artikelname*) vertauscht, und außerdem wurde im Seitenfeld *Kategoriename* die Kategorie *Naturprodukte* ausgewählt, so daß jetzt hierauf gefiltert wird. Die Grafik ist entsprechend angepaßt worden.

34.8.2 Einfache Datenzugriffsseite mit dem Assistenten erstellen und nachfolgend bearbeiten

Datenzugriffsseite mit dem Assistenten erzeugen

In diesem Punkt soll eine Datenzugriffsseite mit dem Assistenten erzeugt und anschließend in der Entwurfsansicht bearbeitet werden. Als Beispiel dient die Datenbank *Nordwind.mdb*. Die Seite soll Details aus den Tabellen *Artikel* und *Lieferanten*, gruppiert nach Kategorien, darstellen. Gehen Sie dazu folgendermaßen vor:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*.
- Aktivieren Sie im Datenbankfenster die Objektliste *Seiten*, und doppelklicken Sie auf *Erstellt eine Datenzugriffsseite unter Verwendung des Assistenten*.

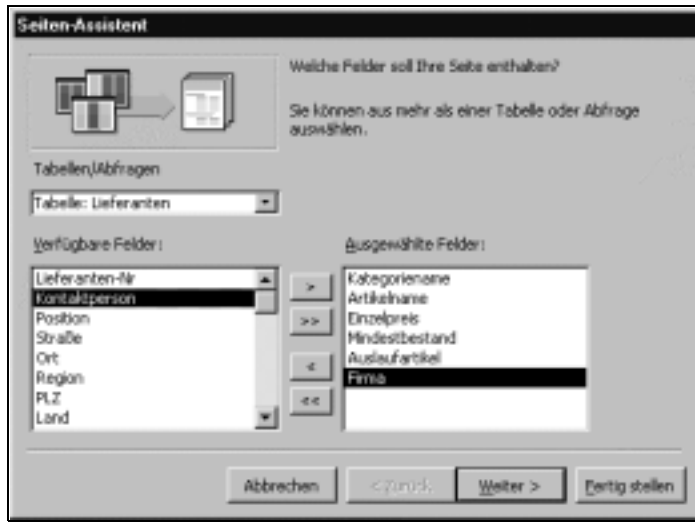


Bild 34.16: Im ersten Dialogfeld des Seiten-Assistenten werden die Felder ausgewählt.

- Wählen Sie im ersten Dialogfeld des Seiten-Assistenten (vgl. Bild 34.16.) die Tabelle *Kategorien* aus, markieren Sie das Feld Kategoriename, und klicken Sie dann auf die Schaltfläche >, um das Feld in die Liste *Ausgewählte Felder* aufzunehmen.

- Wählen Sie in der gleichen Weise wie im vorangehenden Aufzählungspunkt aus der Tabelle *Artikel* die Felder *Artikelname*, *Einzelpreis*, *Mindestbestand* und *Auslaufartikel* sowie aus der Tabelle *Lieferanten* das Feld *Firma*. Das Dialogfeld sollte danach so aussehen wie das in Bild 34.16 wiedergegebene. Klicken Sie schließlich auf die Schaltfläche *Weiter* >.

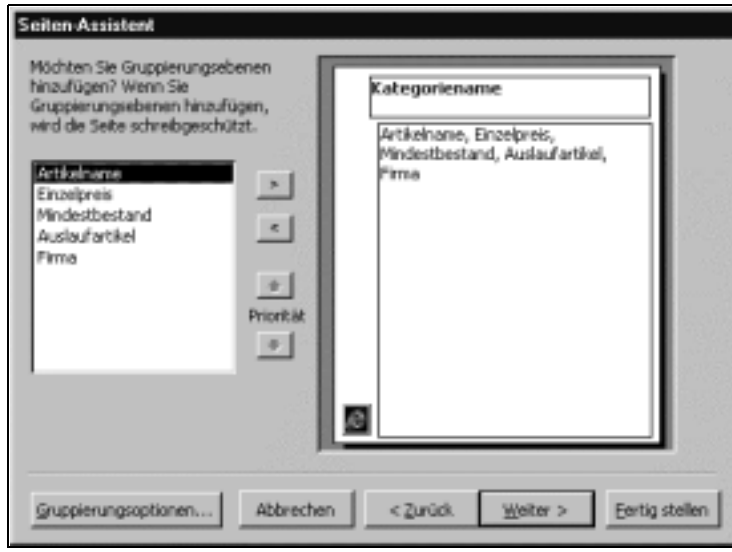


Bild 34.17: Das zweite Dialogfeld des Seiten-Assistenten dient zum Gruppieren. Hier wird nach dem Feld »Kategoriename« gruppiert.

- Im zweiten Dialogfeld des Seiten-Assistenten können Sie Felder für Gruppierungsebenen bestimmen. In unserem Beispiel sollen die Details der Tabellen *Artikel* und *Lieferanten* nach den übergeordneten Kategorien gruppiert werden. Markieren Sie deshalb in der linken Liste das Feld *Kategoriename*, und klikken Sie dann auf die Schaltfläche >. Danach sollte das Dialogfeld aussehen wie das in Bild 34.17 dargestellte. Klicken Sie auf die Schaltfläche *Weiter* >.
- Im dritten Dialogfeld des Seiten-Assistenten (vgl. Bild 34.18) können Sie Sortierfolgen für die nicht in die Gruppierung aufgenommenen Felder festlegen. Schlagen sie die oberste Dropdown-Liste auf, und wählen Sie das Feld Artikelname aus, um die Datensätze im Detailbereich nach diesem Feld zu sortieren. Klicken Sie auf die Schaltfläche *Weiter* >.
- Geben Sie im vierten Dialogfeld des Seiten-Assistenten einen Namen für die Seite an, beispielsweise *Artikel und Lieferanten nach Kategorien*, aktivieren Sie die Option *Seitenentwurf ändern* sowie das Kontrollkästchen *Möchten Sie Ihrer Seite ein Design zuweisen?*, vgl. Bild 34.19. Klicken Sie auf die Schaltfläche *Weiter* >.

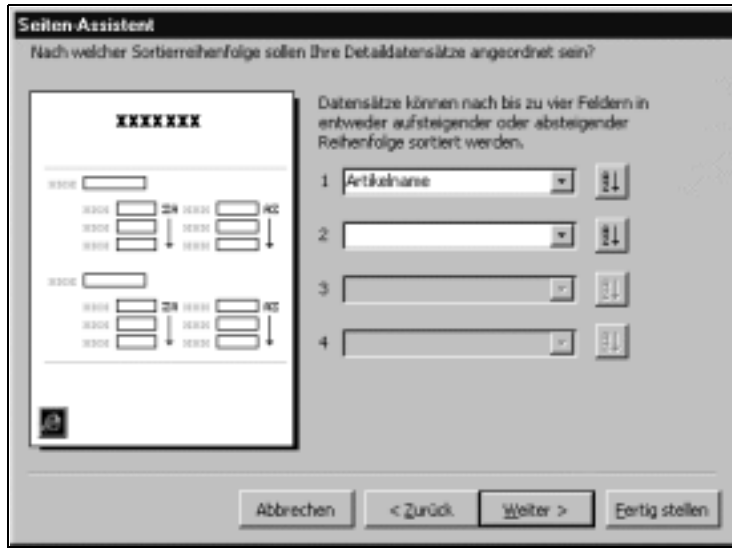


Bild 34.18: Im dritten Dialogfeld des Seiten-Assistenten werden Sortierfolgen festgelegt.

Bild 34.19: Viertes Dialogfeld des Seiten-Assistenten

- Das Dialogfeld *Design* bietet eine große Fülle von Stilvorlagen an, nach denen die Datenzugriffsseite gestaltet werden kann. Die meisten dieser Vorlagen müssen nachträglich von der Installations-CD-ROM installiert werden. Wenn Sie *Geradlinig* wählen, brauchen Sie nicht nachzuinstallieren, weil dieses Design standardmäßig installiert wurde. Klicken Sie auf OK.

Bild 34.20: In diesem Dialogfeld »Design« können Sie eine Mustervorlage für das Layout der Datenzugriffsseite auswählen.

Bild 34.21: Die neu erstellte Datenzugriffsseite unbearbeitet in der Entwurfsansicht

- Die neu erstellte Seite wird in der Entwurfsansicht angezeigt, vgl. Bild 34.21. Diese ähnelt sehr der Entwurfsansicht eines Berichts. Bei genauerem Hinsehen ist bereits hier zu erkennen, daß der oberste Kopfbereich *Lieferanten-Kategoriename* eine sehr geringe Höhe aufweist, denn das Bezeichnungsfeld grenzt hart an seinen unteren Rand. Wechseln Sie mit Hilfe des aufschlagbaren Symbols *Ansicht* in die Datenzugriffsseitenansicht.

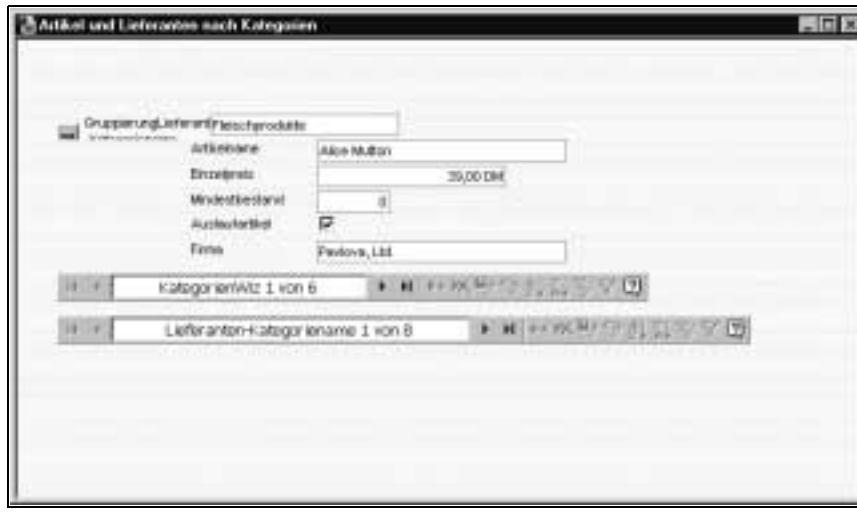


Bild 34.22: Datenzugriffsseitenansicht der neu erstellten Seite, unbearbeitet, jedoch mit erweiterter Detaildarstellung

- Die Datenzugriffsseitenansicht ist in Bild 34.22 wiedergegeben. Sie erkennen, daß der Inhalt des Bezeichnungsfeldes links oben verkrüppelt wiedergegeben wird. Die Seite ist gleichwohl funktionsfähig: Klicken Sie beispielsweise auf das Erweiterungssymbol. Dann wird der erste der Detaildatensätze zur aktuellen Kategorie angezeigt. Mit den Navigationssymbolen können Sie zu weiteren Detaildatensätzen blättern und/oder zu weiteren Kategorien. Beim Blättern der Kategorien werden die Erweiterungen jedesmal zurückgesetzt.

Datenzugriffsseite in der Entwurfsansicht bearbeiten

Aufgabe. Bearbeiten Sie die vom Assistenten erstellte Datenzugriffsseite in mehrfacher Hinsicht gemäß den folgenden Punkten:

- Wechseln Sie wieder mit Hilfe des Symbols *Ansicht* in die Entwurfsansicht.
- Vergrößern Sie die Höhe des obersten Kopfbereichs, damit das Bezeichnungsfeld unverstümmelt dargestellt werden kann: Markieren Sie den Kopfbereich, indem Sie auf seine Kopfleiste klicken, und ziehen Sie dann eines der unteren Ziehkästchen nach unten.

- Der Inhalt des obersten Bezeichnungsfeldes erscheint auch nicht gerade glücklich gewählt: Löschen Sie die ersten Textteile, so daß nur noch der Teil *Kategoriename* übrig bleibt. Arbeiten Sie dabei so, wie Sie es von Bezeichnungsfeldern aus Formularen und Berichten gewohnt sind.
- In Bild 34.22 ist zu erkennen, daß nur jeweils eine Kategorie auf einer Seite dargestellt wird. Bearbeiten Sie die Seite so, daß jeweils drei Kategorien auf einer Seite wiedergegeben werden. Verfahren Sie dabei gemäß den folgenden Aufzählungspunkten.

- Klicken Sie in der Symbolleiste der Entwurfsansicht der Datenzugriffsseite auf das Symbol *Sortieren und gruppieren*, vgl. links nebenstehend. Dann wird das gleichnamige Dialogfeld eingeblendet, vgl. Bild 34.23.

Bild 34.23: Im Dialogfeld »Sortieren und gruppieren« wurde die Datenseitengröße für die erste Gruppierungsebene auf den Wert 3 eingestellt.

- Klicken Sie im Teil *Gruppendatensatzquelle* in die Zeile *Lieferanten-Kategoriename*, um diese Gruppierungsebene im Dialogfeld zu aktivieren.
- Überschreiben sie im Teil *Gruppeneigenschaften* für die Eigenschaft *Datenseitengröße* den Wert 1 mit dem neuen Wert 3.
- Schließen Sie das Dialogfeld *Sortieren und gruppieren*, und geben Sie der Datenzugriffsseite eine Überschrift: Klicken Sie im obersten Teil der Seite auf die Stelle *Titeltext durch Klicken hinzufügen* (vgl. oben Bild 34.21), und geben Sie einen passenden Text ein, z.B. *Artikel und Lieferanten nach Kategorien*.
- Nunmehr sollte die Datenzugriffsseite zufriedenstellend aussehen. Wechseln Sie daher in die Datenzugriffsseitenansicht, um dies zu überprüfen. In Bild 34.24 ist die bearbeitete Datenzugriffsseite *Artikel und Lieferanten nach Kategorien* zu sehen: Keine Bezeichnung erscheint verkrüppelt, die

Seite weist einen Titel auf, und es werden drei Kategorien pro Seite angezeigt, jede mit einem Erweiterungssymbol versehen, von denen in Bild 34.24 auf das zweite geklickt wurde, so daß dort das Verminderungssymbol angezeigt wird.

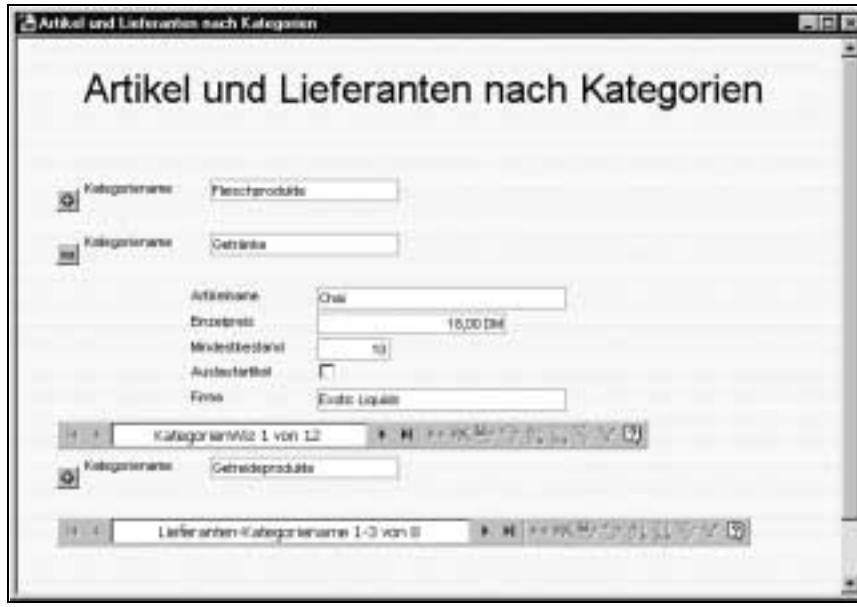


Bild 34.24: Bearbeitete Seite in der Datenzugriffsseitenansicht

- Die Datenzugriffsseite wurde bisher noch nicht gespeichert. Holen Sie dies auf die folgende Weise nach: Klicken Sie, während die Datenzugriffsseite aktiviert ist, auf das Symbol Speichern, oder drücken Sie Strg+S. Wählen Sie im Dialogfeld *Als Datenzugriffsseite speichern* den gewünschten Ordner, und geben Sie einen passenden Dateinamen an, z.B. *Artikel und Lieferanten nach Kategorien.htm*. Klicken Sie schließlich auf die Schaltfläche *Speichern*.

34.8.3 Datenzugriffsseite mit Gruppierungsebene in der Entwurfsansicht erstellen

In diesem Punkt soll gezeigt werden, wie Sie prinzipiell eine neue Datenzugriffsseite ohne den Seiten-Assistenten erstellen. Aus räumlichen Gründen kann es nicht darum gehen, Ihnen jedes Steuerelement der Datenzugriffsseite zu erklären. Diese sind vielmehr in der Hilfe beschrieben, und das meiste funktioniert auch mit Versuch und Irrtum. Gegenüber einem Formular oder Bericht ist die Vorgehensweise, eine Datenzugriffsseite in der Entwurfsansicht von Anfang an neu zu erstellen, jedoch an einigen Stellen prinzipiell anders (und besser!) geregelt. Dies soll hier vor allem aufgezeigt werden.

Aufgabe. Erstellen Sie eine Datenzugriffsseite, die der im vorangehenden Punkt mit dem Seiten-Assistenten erstellten und anschließend bearbeiteten funktional und im Aussehen gleichwertig ist. Gehen Sie zur Lösung dieser Aufgabe folgendermaßen vor:

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*, und aktivieren Sie im Datenbankfenster die Objektliste *Seiten*.
- Doppelklicken sie in der Objektliste *Seiten* auf *Erstellt eine Datenzugriffsseite in der Entwurfsansicht*.

Bild 34.25: Entwurfsansicht unmittelbar nach dem Öffnen einer neuen Datenzugriffsseite

- Klicken Sie in der Symbolleiste auf das Symbol *Feldliste*, um diese anzuzeigen.
- Erweitern Sie in der Registerkarte *Datenbank* den Knoten *Tabellen*, so daß alle Tabellen in der Feldliste angezeigt werden.
- Erweitern Sie die Tabelle *Kategorien*, so daß ihre Felder angezeigt werden.
- Ziehen Sie das Feld Kategoriename in den *Bereich: Ungebunden*, vgl. Bild 34.26.
- Klicken Sie, während das neu eingefügte Textfeld *Kategoriename* markiert ist, in der Symbolleiste auf das Symbol *Höher stufen*, um eine Gruppierungsebene einzufügen, damit nach diesem Feld gruppiert werden kann. Das Ergebnis dieser Operation ist in Bild 34.27 wiedergegeben.
- Erweitern Sie in der Feldliste die Tabelle *Artikel*, und ziehen Sie daraus das Feld *Artikelname* in den *Kopfbereich: Kategorien*. Dann wird das Dialogfeld *Layout-Assistent* angezeigt, vgl. Bild 34.28. Aktivieren Sie darin die Option *Einzelne Steuerelemente*, und bestätigen Sie mit OK. Der *Kopfbereich: Kategorien* wird dann umbenannt in *Kopfbereich: Artikel*.

Bild 34.26: Das Feld »Kategoriename« wird gerade aus der Feldliste in den »Bereich: Ungebunden« gezogen.

Bild 34.27: Entwurfsansicht der neuen Datenzugriffsseite unmittelbar nach dem Höherstufen des Feldes »Kategoriename«

Bild 34.28: Dialogfeld »Layout-Assistent«

- Fügen Sie auf die gleiche Weise aus der Tabelle *Artikel* auch die Felder *Einzelpreis*, *Mindestbestand* und *Auslaufartikel* und aus der Tabelle *Lieferanten* das Feld *Firma* in den *Kopfbereich: Artikel* ein. Das Dialogfeld *Layout-Assistent* wird beim Einfügen dieser weiteren Felder nicht mehr angezeigt, weil die Entscheidung zwischen einer PivotTable-Liste und einzelnen Steuerelementen nur beim Einfügen des ersten Feldes in einen Kopfbereich ansteht.
- Ordnen Sie die Steuerelemente im *Kopfbereich: Artikel* am besten so an, daß sie gegenüber dem Steuerelement *Kategoriename* im Kopfbe*reich: Kategorien-Kategoriename* nach rechts versetzt sind, um ihre Unterordnung auch äußerlich deutlich zu machen.
- Geben Sie im obersten Bereich der Seite einen Titel ein, beispielsweise *Artikel und Lieferanten nach Kategorien*.

Bild 34.29: Entwurfsansicht der neuen Datenzugriffsseite nach dem Einfügen aller Felder und eines Titels. Für jeden Kopfbereich wurde automatisch eine Navigationsleiste eingefügt.

- Nach dem Einfügen aller Felder und des Titels sollte Ihr Entwurf ungefähr so aussehen wie in Bild 34.29. Wenn Sie aus dieser Situation heraus in die Datenzugriffsseitenansicht wechseln, werden Sie erkennen, daß für jeden Bereich jeweils fünf Datenseiten angezeigt werden. Dies erscheint zu unübersichtlich. Vielmehr sollen im Datenbereich des obersten Bereichs *Kategoriename* drei Datenseiten, im unteren dagegen nur eine angezeigt

werden. Um dieses Ergebnis zu erzielen, lassen Sie das Dialogfeld *Sortieren und gruppieren* anzeigen, indem Sie auf das gleichnamige Symbol in der Symbolleiste klicken. Dieses Dialogfeld wurde bereits oben in Bild 34.23 wiedergegeben, weil dort eine gleichartige Aufgabe wie hier zu lösen war. Geben Sie für die oberste Gruppierungsebene für die Eigenschaft *Datenseitengröße* den Wert 3 und für die untere Ebene den Wert 1 ein.

- Die unterste Navigationsleiste, die sich auf *Kategoriename* bezieht, ist nicht breit genug, um in der Datenzugriffsseitenansicht den ganzen Text wiederzugeben. Tragen Sie daher für ihre Eigenschaft *Width* den Wert 500 ein: Markieren Sie die Navigationsleiste, lassen Sie das Eigenschaftenfenster anzeigen, rollen Sie darin zur Eigenschaft *Width*, und geben Sie dort den Wert 500 ein.
- Jetzt sollte Einfügen und Bearbeiten der neuen Datenzugriffsseite fertig sein. Wechseln Sie daher in die Datenzugriffsseitenansicht. Das Ergebnis sollte ungefähr so aussehen wie in Bild 34.30. Es dürfte dem bearbeiteten Ergebnis des Seiten-Assistenten im vorigen Punkt (vgl. Bild 34.24) sehr nahe kommen, funktionell ist es jedenfalls gleichwertig.

Bild 34.30: Neue und fertig bearbeitete Datenzugriffsseite in der Datenzugriffsseitenansicht

34.8.4 Datenzugriffsseite mit PivotTable und Diagramm erstellen

Mit diesem letzten Beispiel soll gezeigt werden, wie Sie eine Datenzugriffsseite mit einer PivotTable und einem Diagramm erstellen können. Diese Aufgabe läßt sich nicht – auch nicht teilweise – mit dem Seiten-Assistenten lösen, sondern Sie müssen mit einer leeren Datenzugriffsseite in der Entwurfsansicht beginnen. Sie werden jedoch sehen, daß die Möglichkeiten der

Office Web Komponenten, die Sie dabei implizit unterstützen, so benutzerfreundlich sind, daß die Datenzugriffsseite recht einfach erstellt werden kann.

Aufgabe. Erstellen Sie für die Nordwind-Datenbank eine Datenzugriffsseite mit einer PivotTable und einem darauf aufbauenden Diagramm mit der folgenden, zunächst zu erstellenden Abfrage als Datenbasis: Die Abfrage soll aus der Tabelle *Personal das Feld Nachname, aus der Tabelle Bestellungen die Felder Bestimmungsland, Versanddatum und Bestell-Nr sowie aus der Abfrage Bestellzwischensummen das Feld Zwischensumme unter der Bezeichnung Umsatz enthalten. Gehen Sie folgendermaßen vor, um die Aufgabe zu lösen:*

Abfrage erstellen

- Öffnen Sie ggf. die Datenbank *Nordwind.mdb*, und erstellen Sie eine neue Abfrage in der folgenden Weise.
- Doppelklicken Sie in der Objektliste *Abfragen* auf *Erstellt eine neue Abfrage in der Entwurfsansicht*.
- Nehmen Sie die Tabellen *Personal und Bestellungen sowie die Abfrage Bestellzwischensummen in die Abfrage auf*.
- FÜGEN SIE DER ABFRAGE AUS DER TABELLE *Personal* das Feld *Nachname*, aus der Tabelle *Bestellungen* die Felder *Bestimmungsland*, *Versanddatum* und *Bestell-Nr* sowie aus der Abfrage *Bestellzwischensummen* das Feld *Zwischensumme* hinzu. Geben Sie dem Feld *Zwischensumme* die Bezeichnung *Umsatz*, indem Sie dieses Wort, GEFOLGT VON EINEM DOPPELPUNKT UND EINEM LEERZEICHEN, IN DER SPALTE *Feld* vor den Namen *Zwischensumme* SCHREIBEN. IHRE ABFRAGE SOLLTE SCHLIEßLICH SO AUSSEHEN, WIE IN Bild 34.31 wiedergegeben. Speichern Sie diese Abfrage schließlich unter dem Namen *Personalumsätze nach Bestimmungsland (oder einem vergleichbaren Namen), und schließen Sie sie.*

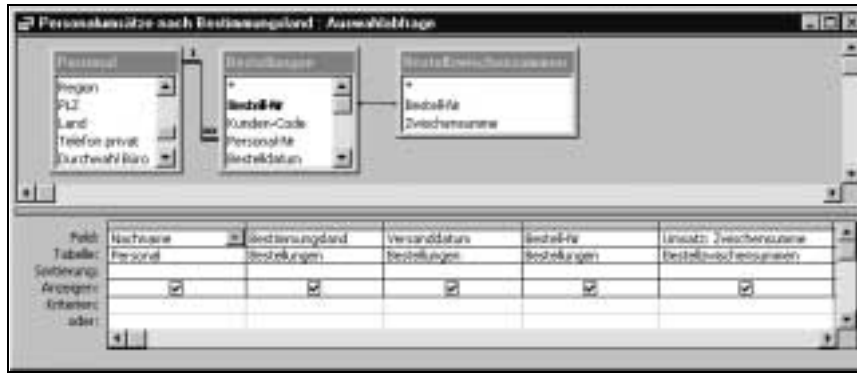


Bild 34.31: Abfrage »Personalumsätze nach Bestimmungsland« in der Entwurfsansicht

Steuerelement *PivotTable* einfügen und bearbeiten

- Erstellen Sie eine neue Datenzugriffsseite in der Entwurfsansicht: Aktivieren Sie die Objektliste *Seite*, und doppelklicken Sie auf *Erstellt eine Datenzugriffsseite in der Entwurfsansicht*.
- Fügen Sie einen Titel ein, beispielsweise *Umsätze nach Personal und Bestimmungsland*.
- Blenden Sie die *Feldliste* ein, indem Sie auf das gleichlautende Symbol klicken.
- Erweitern Sie in der Registerkarte *Datenbank* den Knoten *Abfragen*, und ziehen Sie die in den vorangehenden Aufzählungspunkten erstellte Abfrage *Personalumsätze nach Bestimmungsland* in den *Bereich: Ungebunden*. Auch wenn es ungewöhnlich erscheint: Ziehen Sie tatsächlich die gesamte Abfrage und nicht einzelne Felder daraus in den *Bereich: Ungebunden*! Dieser Bereich erhält vorübergehend einen blauen Rahmen, während Sie das Abfrage-Objekt in den Bereich gezogen, jedoch die Maustaste noch nicht losgelassen haben. Nachdem Sie die Maustaste losgelassen haben, wird das Dialogfeld *Layout-Assistent* angezeigt, vgl. oben Bild 34.28.
- Wählen Sie im Dialogfeld *Layout-Assistent* die Option *PivotTable-Liste*, und bestätigen Sie mit OK. Daraufhin wird ein PivotTable-Steuerelement in den *Bereich: Ungebunden* eingefügt, vgl. Bild 34.32.

Bild 34.32: Entwurfsansicht der Datenzugriffsseite unmittelbar nach dem Einfügen des PivotTable-Steuerelements

- Vergrößern Sie die Fensterbreite der Datenzugriffsseiten-Entwurfsansicht auf annähernd die gesamte Bildschirmbreite, vergrößern Sie die Breite des

Bereich: Ungebunden auf die Breite des Fensters, und vergrößern Sie die Breite des PivotTable-Steuerelements so, daß es praktisch die gesamte Bereichbreite einnimmt.

- Markieren Sie ggf. das PivotTable-Steuerelement, und klicken Sie im markierten Zustand noch einmal darauf, um die PivotTable bearbeitbar zu machen. Sie erkennen den Bearbeitungszustand an einem schraffierten Rahmen um das PivotTable-Steuerelement.
- Sorgen Sie dafür, daß die PivotTable eine Symbolleiste erhält: Klicken sie, während sich die PivotTable im Bearbeitungsmodus befindet, mit der rechten Maustaste auf ein Feld der PivotTable, und klicken Sie im Kontextmenü auf *Eigenschaftentoolbox*. Dann wird die (kontextabhängig arbeitende) Eigenschaftentoolbox angezeigt. Erweitern Sie die Eigenschaftentoolbox durch Klicken auf *Ein-/Ausblenden*, und klicken Sie danach auf das Symbol *Symbolleiste*. Daraufhin wird der PivotTable eine Symbolleiste hinzugefügt.
- Das PivotTable-Feld *Nachname* soll zum PivotTable-Spaltenfeld werden: Markieren Sie die Spalte *Nachname*, und klicken Sie dann in der PivotTable-Symbolleiste auf das Symbol *In Spaltenbereich verschieben*.
- Das PivotTable-Feld *Bestimmungsland* soll zum PivotTable-Zeilenfeld werden: Markieren Sie die Spalte *Bestimmungsland*, und klicken Sie dann in der PivotTable-Symbolleiste auf das Symbol *In Zeilenbereich verschieben*. Nach dieser Bearbeitung sollte der Datenzugriffsseiten-Entwurf etwa so aussehen, wie in Bild 34.33 wiedergegeben.

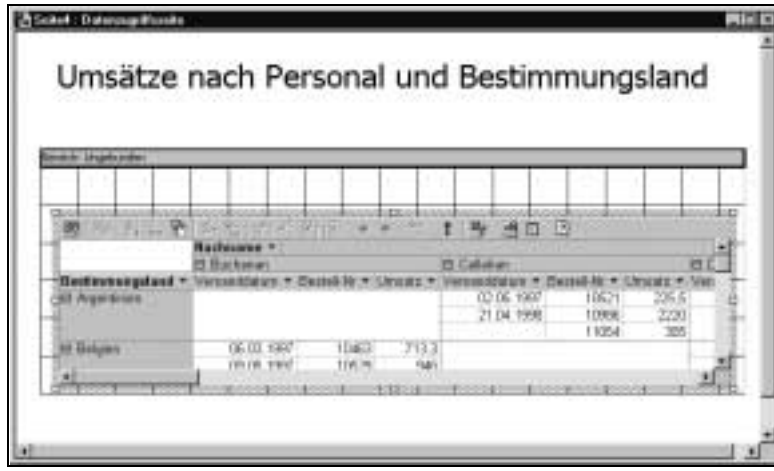


Bild 34.33: Bearbeitungszustand der Datenzugriffsseite in der Entwurfsansicht nach dem Zuweisen von Spalten- und Zeilenfeld.

- Die Umsatzwerte sollen mit dem Währungsformat formatiert werden: Markieren Sie eine Umsatzspalte, klicken Sie mit der rechten Maustaste

auf die markierte Spalte, und wählen Sie im Kontextmenü *Eigenschaftentoolbox*. Wählen Sie dann in der Eigenschaftentoolbox das Zahlenformat *Währung*, vgl. Bild 34.34.

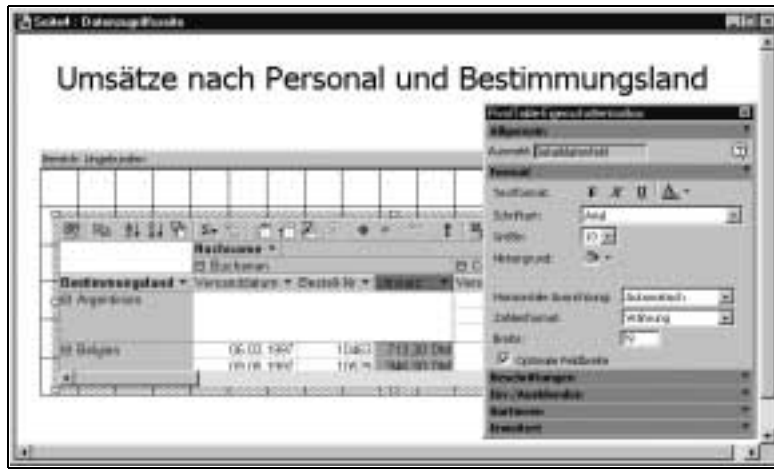


Bild 34.34: Situation unmittelbar nach dem Formatieren der Umsatz-Felder mit dem Währungsformat, mit angezeigter Eigenschaftentoolbox.

- Die PivotTable soll Teil- und Gesamtsummen ausweisen: Markieren Sie eine Umsatzspalte (bzw. lassen Sie diese aus dem letzten Schritt markiert), klicken Sie auf das Symbol *AutoBerechnung* (es trägt das Summenzeichen), und wählen Sie in der dann angezeigten Liste die Funktion *Summe*. In der Seitenansicht sollte die Datenzugriffsseite nunmehr so ähnlich aussehen, wie in Bild 34.35 dargestellt.

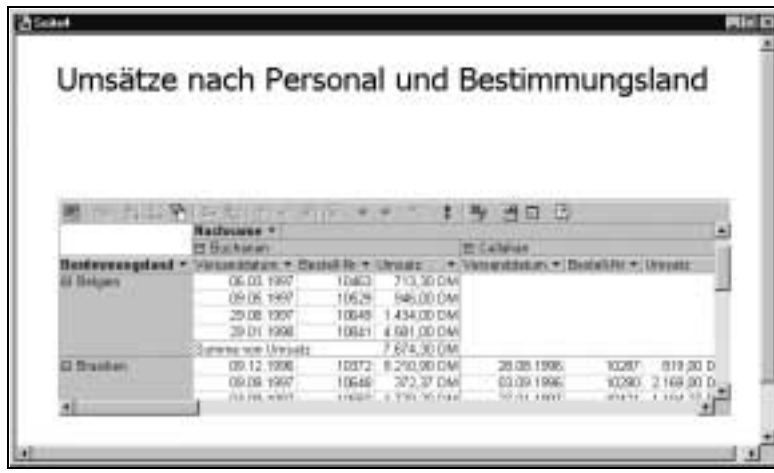


Bild 34.35: Datenzugriffsseite in der Seitenansicht nach endgültiger Bearbeitung der PivotTable

Diagramm einfügen

- Um der Datenzugriffsseite ein Diagramm hinzuzufügen, wechseln Sie ggf. wieder in die Entwurfsansicht
- Vergrößern Sie das Fenster so, daß es annähernd den gesamten Bildschirm einnimmt.
- Vergrößern Sie den *Bereich: Ungebunden* so, daß er bis zum unteren Fensterrand reicht.
- Verändern Sie die Größe des Steuerelements *PivotTable* so, daß es ungefähr die untere Hälfte des *Bereich: Ungebunden* einnimmt, so daß im oberen Teil Platz für das einzufügende Diagramm bleibt.
- Klicken Sie auf das Symbol *Toolbox*, um diese einzublenden.
- Klicken Sie in der Toolbox auf das Symbol *Office Chart*, und ziehen Sie dann über den gesamten freien Platz im *Bereich: Ungebunden* ein Rechteck.

Bild 34.36: Erstes Dialogfeld des Office Chart-Assistenten zum Bestimmen des Diagrammtyps

- Markieren Sie den Diagrammtyp *Balken* und dazu den ersten Untertyp, wie in Bild 34.36 dargestellt. Klicken Sie dann auf *Weiter* >.
- Wählen Sie im zweiten Dialogfeld des Office Chart-Assistenten als Datenquelle *Microsoft Office PivotTable*, und klicken Sie dann auf *Weiter* >.
- Belassen Sie es im dritten Dialogfeld des Office Chart-Assistenten bei der Option *Reihe in Zeilen*, und klicken Sie auf *Fertig stellen*.

Bild 34.37: Zweites Dialogfeld des Office Chart-Assistenten zum Bestimmen der Datenquelle

- Die gesamte Datenzugriffsseite ist nunmehr fertig. Wechseln Sie daher in die Seitenansicht, um sich das Ergebnis anzuschauen, das so ähnlich aussehen sollte, wie in Bild 34.38 wiedergegeben.

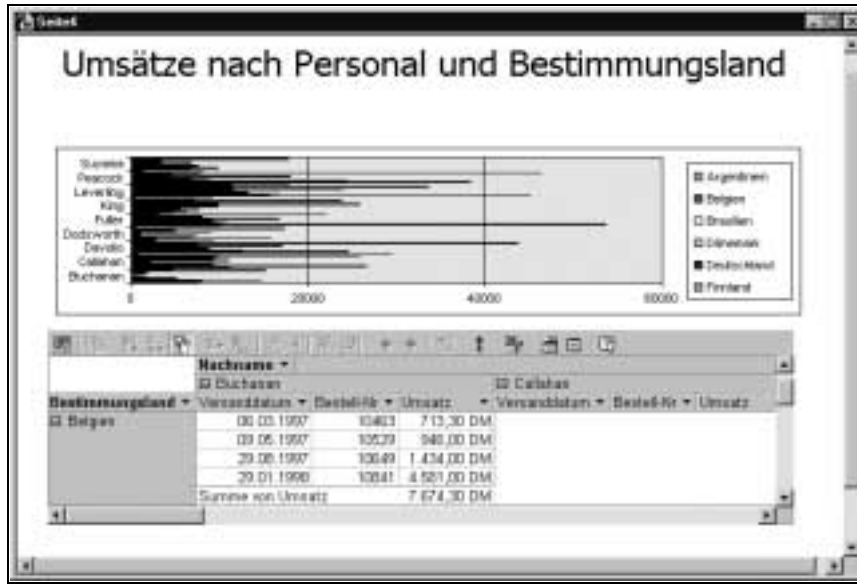


Bild 34.38: Fertig bearbeitete Datenzugriffsseite mit eingefügtem Diagramm

Die Darstellung in Bild 34.38 ist im Prinzip richtig, jedoch zeigt sie auch, daß das Diagramm mit der Wiedergabe so vieler Datenpunkte, wie sie momentan in der PivotTable ausgewiesen werden, überfordert ist: Bildlich gesprochen ist der Wald vor lauter Bäumen nicht zu erkennen. Schränken Sie daher die in der PivotTable ausgewiesenen Länder und Personen so ein, daß nur noch die Länder *Deutschland*, *Frankreich* und *Großbritannien* und nur noch die Personen *Buchanan*, *Davolio* und *Fuller* dargestellt werden. Gehen Sie dazu folgendermaßen vor:

- Verbleiben Sie in der Seitenansicht. Klicken sie in der PivotTable auf den Pfeil neben dem Feld *Bestimmungsland*, um das Dialogfeld für die Elementauswahl aufzuschlagen (vgl. Bild 34.39), und deaktivieren Sie darin alle Länder außer *Deutschland*, *Frankreich* und *Großbritannien*. Verfahren Sie dabei zweckmäßigerweise wie folgt: Deaktivieren sie zunächst alle Länder, indem sie das Kontrollkästchen *(Alle anzeigen)* deaktivieren. Aktivieren Sie daraufhin die Kontrollkästchen der drei genannten Länder. Bestätigen Sie schließlich mit OK.

Bild 34.39: Dialogfeld für die Elementauswahl im Feld »Bestimmungsland«

- Verfahren Sie entsprechend für die Auswahl der drei Personen *Buchanan*, *Davolio* und *Fuller*. Nach dieser Änderung ist das Diagramm mit der Wiedergabe der Daten aus der PivotTable nicht mehr überfordert, sondern liefert sinnvoll interpretierbare Informationen, vgl. Bild 34.40.

Speichern Sie die Datenzugriffsseite schließlich unter einem bestimmten Namen und öffnen Sie die Seite anschließend im Internet Explorer. Es zeigt sich, daß Sie die PivotTable dort genauso bearbeiten können wie in Access 2000. Beispielsweise können Sie die einzelnen Felder der PivotTable mit Hilfe der Symbole auf der Symbolleiste in die Positionen des Zeilen-, Spalten- oder Filterbereichs bringen, indem Sie die Felder markieren und anschließend die entsprechenden Symbole klicken. Die Darstellung in Bild 34.41 beispielsweise wurde im Internet Explorer erzielt: Gegenüber der Situation unmittelbar nach dem Öffnen im Internet Explorer wurden die Länder und Personen auf jeweils drei Elemente eingeschränkt, die Felder *Versanddatum* und *Bestell-Nr* wurden in den Filter-, das Feld *Bestimmungsland* in den Spalten- und das Feld *Nachname* in den Zeilenbereich verschoben.

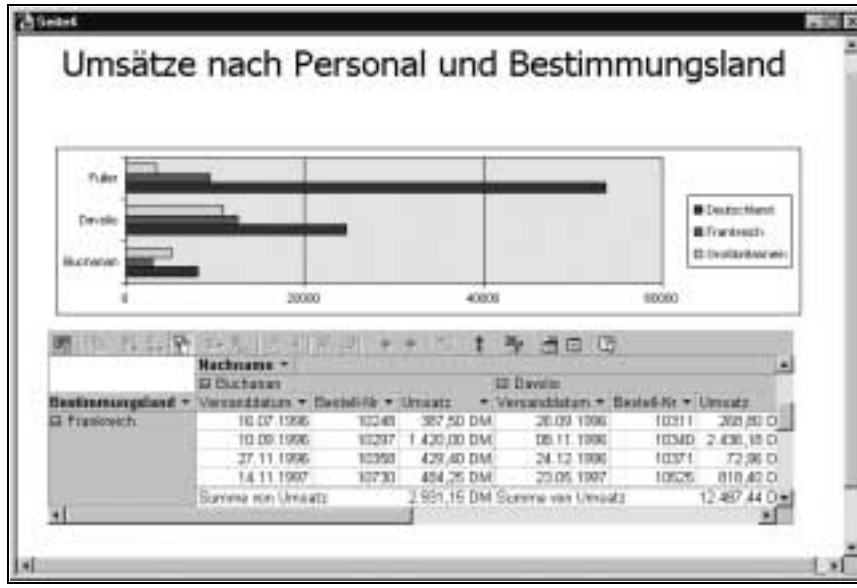


Bild 34.40: Nach Einschränkung der PivotTable auf wenige Länder und Personen stellt das Diagramm die Daten übersichtlich dar.

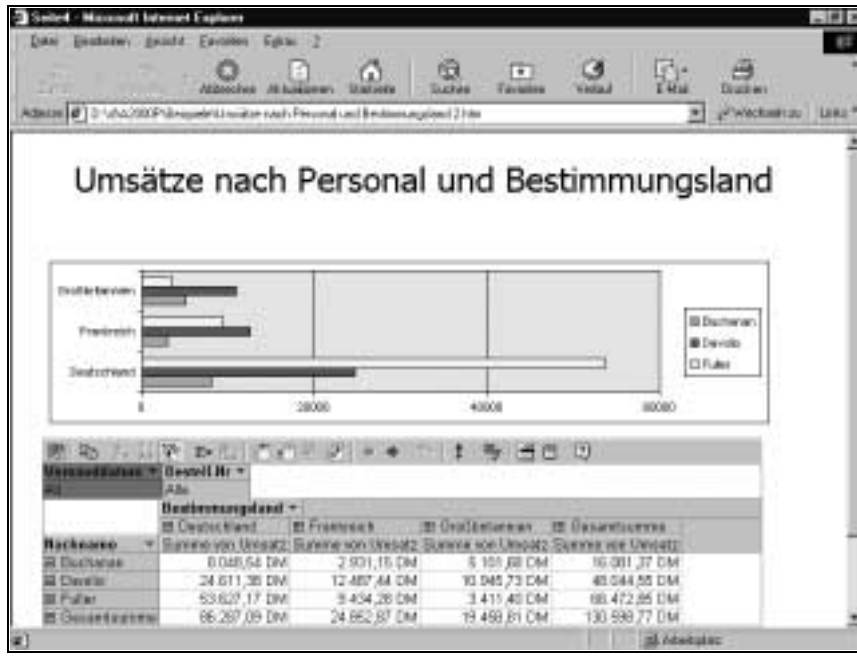


Bild 34.41: Die in diesem Punkt erstellte Datenzugriffsseite wurde im Internet Explorer geöffnet und darin bearbeitet.

34.8.5 Wann und wie sind Daten aktualisierbar?

Mit Datenzugriffsseiten können die Daten der zugrundeliegenden Datenquelle unter gewissen Bedingungen aktualisiert werden. Diese Bedingungen sind zunächst einmal dieselben, wie sie für Mehrtabellenabfragen gelten: Sind diese aktualisierbar, so können auch die Daten der Datenzugriffsseite prinzipiell aktualisiert werden. (Zur Aktualisierbarkeit von Abfragen vgl. Kap. 11, *Auswahlabfragen*, Punkt 11.6, *Datenaktualisierung bei Ein- und Mehrtabellenabfragen*.) Darüber hinaus muß auch die Eigenschaft *RecordsetType* des Objekts *Seite* (das ist die Datenzugriffsseite im ganzen) der Datenzugriffsseite auf den Wert *dscUpdatableSnapshot* eingestellt sein. In manchen Fällen sorgt der Seiten-Assistent für diese Einstellung, in anderen müssen Sie dies manuell nachholen.

In diesem Zusammenhang ist auch die Eigenschaft *DataEntry* des Objekts *Seite* bedeutsam. Wenn sie auf den Wert *Wahr* eingestellt wird (die Voreinstellung dieser Eigenschaft ist *Falsch*), können in die Datenzugriffsseite nur neue Daten eingegeben werden; die Datenzugriffsseite erscheint dann beim Öffnen mit leeren Eingabefeldern, sie dient daher als reine Eingabemaske.

☞ Wenn Sie in der Entwurfsansicht einer Datenzugriffsseite das Objekt *Seite* markieren wollen (analog zum Formular oder Bericht), so werden Sie vermutlich dieselbe Erfahrung machen wie ich: Dies scheint weder durch Klikken mit der Maus noch mit dem in der Hilfe für diesen Zweck genannten Befehl *Seite markieren* aus dem Menü *Bearbeiten* (dies wäre auch analog zu Formularen und Berichten) möglich: Dieser Befehl wird gar nicht angeboten, und ein Symbol entsprechend demjenigen in Formularen (kleines schwarzes Quadrat in der linken oberen Ecke des Formulars) existiert hier nicht. Mit folgendem Trick können Sie dennoch das Objekt *Seite* in der Entwurfsansicht markieren: Verlassen Sie ggf. die Vollbildansicht, so daß die Entwurfsansicht der Datenzugriffsseite in einem eigenen Fenster dargestellt wird, und klicken Sie auf die Titelleiste dieses Fensters. Dann sollte in der Titelleiste des Eigenschaftenfensters *Seite* angezeigt werden, so daß deren Eigenschaften bearbeitbar sind.

Bei der Bearbeitung und Neueingabe von Datensätzen sind verschiedene Symbole auf der Navigationsleiste besonders bedeutsam: Nach der Bearbeitung oder Neueingabe eines Datensatzes müssen Sie diesen explizit speichern, damit die Änderungen nicht verlorengehen! Mit den entsprechenden Symbolen können Sie in einer Datenzugriffsseite, deren Datensätze Sie bearbeiten können, auch Datensätze löschen oder neue einfügen.

34.8.6 Name oder Pfad der Datenbank wurden geändert

Wenn die Datenbank, die Datenquelle einer Datenzugriffsseite ist, in einen anderen Ordner verlegt wurde oder wenn sich ihr Name geändert hat, kann die Datenzugriffsseite keine Verbindung zu den Daten mehr herstellen. Der Internet Explorer gibt dann beispielsweise die in Bild 34.42 wiedergegebene

Meldung aus. Sie müssen, um derartige Fehler zu vermeiden, die betreffende Datenzugriffsseite in Access 2000 bearbeiten und ihr den neuen Pfad oder Namen mitteilen. Gehen Sie dazu folgendermaßen vor:

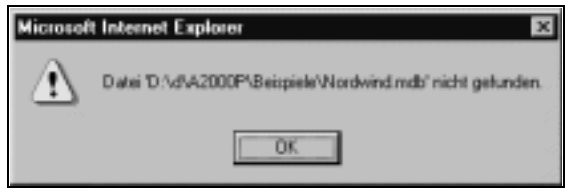


Bild 34.42: Meldung des Internet Explorer beim Versuch, eine Datenzugriffsseite mit nicht mehr zutreffender Datenquelle zu öffnen

- Öffnen Sie die Datenbank, in der die betreffende Datenzugriffsseite als Objekt *Seite* gespeichert ist.
- Öffnen Sie die Seite in der Entwurfsansicht.
- Zeigen sie die Feldliste an, aktivieren Sie darin die Registerkarte *Datenbank*, klicken Sie mit der rechten Maustaste auf den obersten Knoten, der den Namen der bisherigen Datenquellen-Datenbank trägt, und wählen Sie im Kontextmenü den Befehl *Verbindung*, vgl. Bild 34.43. Dann wird das Dialogfeld *Datenlinkeigenschaften* angezeigt, vgl. Bild 34.44.

Bild 34.43: Kontextmenü im Dialogfeld »Feldliste«

- Geben Sie im Dialogfeld *Datenlinkeigenschaften* den neuen voll mit Pfad qualifizierten Namen der Datenbank ein, oder durchsuchen Sie die Festplatte oder das Netz, indem Sie auf die Schaltfläche mit den drei Punkten

klicken. Bevor Sie das Dialogfeld mit OK bestätigen, können Sie mittels der Schaltfläche *Verbindung testen* prüfen, ob die neue Verbindung (und natürlich in anderen Kontexten jederzeit auch die alte) funktioniert.

Bild 34.44: Dialogfeld »Datenlinkeigenschaften«, in dem der neue Pfad oder Name angegeben werden kann.

Kapitel 35

Arbeiten in einer Mehrbenutzerumgebung

35.0.1 Überblick

Wenn Sie in einem Netzwerk arbeiten, können Sie Daten einer Datenbank gemeinsam mit anderen Mitarbeitern nutzen. Sie arbeiten dann in einer *Mehrbenutzerumgebung*. Es gibt mehrere Möglichkeiten, Daten in einer Mehrbenutzerumgebung gemeinsam zu nutzen:

Gemeinsame Nutzung der gesamten Datenbank

Sie können die gesamte Datenbank in einem Ordner eines Netzwerk-Servers ablegen, auf den alle Benutzer der Datenbank Zugriff haben. Dann kann dieselbe Datenbank von mehreren Benutzern gleichzeitig geöffnet werden. Diese Vorgehensweise läßt sich am einfachsten realisieren. Allerdings nutzen die Benutzer dann nicht nur die Daten gemeinsam, sondern sie verwenden auch dieselben Formulare, Berichte, Abfragen, Makros und Module. Dieses Verfahren kann mit Leistungseinschränkungen verbunden sein, weil nicht nur die Daten, sondern auch die anderen Datenbankobjekte über das Netz transportiert werden müssen.

Gemeinsame Nutzung der Datenbanktabellen

Eine bessere Performance erzielen Sie, wenn Sie eine Datenbank in zwei Datenbanken aufteilen: Die eine Datenbank – das Backend – enthält nur die Tabellen mit den Daten; diese Datenbank wird auf dem Netzwerk-Server abgelegt und gemeinsam genutzt. Die andere Datenbank – das Frontend – enthält alle übrigen Objekte der Ursprungsdatenbank, und sie enthält verknüpfte Tabellen mit einer Verknüpfung zur Backend-Datenbank. Die Frontend-Datenbank wird dann bei jedem Benutzer lokal installiert. Die Leistung

dieser Vorgehensweise ist im Vergleich zu der im vorangehenden Punkt beschriebenen besser, da nur Daten über das Netzwerk übertragen werden, während die anderen Objekte wie Formulare oder Berichte, die oft große Informationsmengen umfassen, lokal geladen werden. Die Benutzer können außerdem ihre Formulare, Berichte und anderen Datenbankobjekte ihren individuellen Bedürfnissen und Arbeitsweisen anpassen, ohne daß dies Auswirkungen auf die anderen Benutzer hat. Das Arbeiten mit verknüpften Tabellen ist ausführlich an anderer Stelle dieses Buches behandelt worden, vgl. Kap. 9, *Tabellen importieren, exportieren und verknüpfen*, Punkt 9.3, *Verknüpfen von Tabellen*.

Erstellen einer Client/Server-Anwendung

Sie können auch eine Client/Server-Anwendung erstellen, vgl. dazu im einzelnen Kap. 37, *Access-Projekte als Client/Server-Datenbank: Grundlagen* und Kap. 38, *Access-Projekt erstellen* und bearbeiten. Diese Vorgehensweise bietet die größten Leistungsvorteile, und Sie können die zusätzlichen Möglichkeiten der Datensicherheit nutzen, die eine SQL-Datenbank im allgemeinen bietet. Auch in diesem Falle liegt die SQL-Datenbank als Backend auf einem Netzwerk-Server, während die Datenbank, die die Access-Objekte enthält, entweder ebenfalls auf einem Netzwerk-Server oder jeweils lokal auf dem PC der einzelnen Benutzer installiert sein kann.

Replikation der Datenbank

Wenn Sie mit zwei Rechnern arbeiten, z.B. einem PC im Büro und einem Notebook für unterwegs, können Sie den *Aktenkoffer* verwenden, um von der Access-Datenbank *Replikate* anzulegen. Bei dieser Vorgehensweise können mehrere Benutzer an verschiedenen Orten unverbunden mit ihren eigenen Datenbank-Kopien, die auch die Daten umfassen, arbeiten und diese anschließend über das Netzwerk synchronisieren. Die inhaltliche Seite der Synchronisation wird dabei von der Replikation überwacht. Beispielsweise muß beim Arbeiten in einer derartigen Umgebung überwacht werden, daß keine Widersprüche in den Datensätzen auftreten. In diesem Buch wird der Umgang mit Replikaten nicht behandelt, vgl. dazu im einzelnen die Online-Hilfe, die diese Arbeitsweise ausführlich dokumentiert.

35.0.2 Erstellen von und Anschließen an Arbeitsgruppen

Wenn mehrere Benutzer gleichzeitig mit derselben Datenbank arbeiten wollen, müssen sie sich derselben Arbeitsgruppe angeschlossen haben. Eine Arbeitsgruppe wird mit Hilfe einer sogen. Arbeitsgruppen-Informationsdatei definiert: Alle Benutzer, die sich derselben Arbeitsgruppen-Informationsdatei angeschlossen haben, bilden eine Arbeitsgruppe. Wenn Sie Access installieren, erstellt das Setup-Programm automatisch eine Arbeitsgruppen-Informationsdatei. Diese bekommt den Namen *System.mdw* und wird im Ordner

\Programmdateien\Microsoft Office\Office abgelegt. Wenn Sie nach der Installation von Access nichts weiter veranlassen, ist jeder Benutzer automatisch dieser Arbeitsgruppen-Informationsdatei angeschlossen. Daher gibt es in dieser Situation auch nur eine Arbeitsgruppe.

Im allgemeinen sollten Sie eine (oder mehrere) eigene, neue Arbeitsgruppen-Informationsdateien erstellen und sich nicht auf die mit dem Setup-Programm installierte beziehen, wenn Sie Zugriffsrechte auf Datenbanken (vgl. das folgende Kap. 36, *Zugriffsrechte für Datenbank oder einzelne Objekte festlegen*) zuverlässig verwalten wollen, weil nur dies volle Sicherheit gewährleistet. Die Online-Hilfe weist ausdrücklich darauf hin, daß Ihre Sicherheitsvorkehrungen bezüglich der Verwaltung von Zugriffsrechten unterlaufen werden können, wenn Sie mit der vom Setup-Programm installierten Arbeitsgruppen-Informationsdatei *\Programmdateien\Microsoft Office\Office\System.mdw* arbeiten, weil deren Kennung nur auf Basis öffentlich bekannter und daher leicht reproduzierbarer Namens- und Firmeninformationen gebildet wird.

Erstellen einer neuen Microsoft Access-Arbeitsgruppen-Informationsdatei

Erstellen Sie eine neue Arbeitsgruppen-Informationsdatei und legen eine Arbeitsgruppen-Identifikationsnummer (WID) fest. Nur wer diese Identifikationsnummer kennt, wird in der Lage sein, eine Kopie dieser Arbeitsgruppen-Informationsdatei zu erstellen. Gehen Sie folgendermaßen vor:

- Beenden Sie ggf. jede laufende Access-Sitzung.
- Starten Sie den Arbeitsgruppen-Administrator. Verfahren Sie dazu am besten folgendermaßen: Suchen Sie mit dem Explorer im Ordner *\Programmdateien\Microsoft Office\Office* die Verknüpfung *MS Access Arbeitsgruppen-Administrator*, und doppelklicken Sie darauf. (Alternativ: Suchen Sie die Datei *Wrkgadm.exe*, und doppelklicken Sie darauf.) Dann wird das Dialogfeld *Arbeitsgruppen-Administrator* angezeigt, vgl. Bild 35.1.

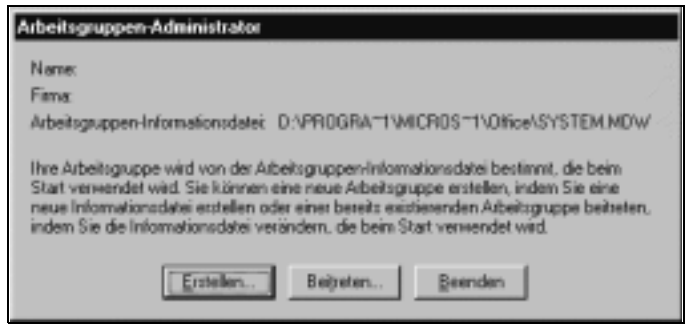


Bild 35.1: Dialogfeld »Arbeitsgruppen-Administrator«

- Klicken Sie im Dialogfeld *Arbeitsgruppen-Administrator* auf die Schaltfläche *Erstellen*, und geben Sie dann Ihren Namen und den Ihrer Firma ein.
- Geben Sie im Dialogfeld *Informationen zum Arbeitsgruppeneigentümer* (vgl. Bild 35.2) für die *Arbeitsgruppen-ID* eine beliebige Kombination aus bis zu 20 Ziffern und Buchstaben ein, und klicken Sie dann auf OK.

Bild 35.2: Dialogfeld »Informationen zum Arbeitsgruppeneigentümer«

- Geben Sie im Dialogfeld *Arbeitsgruppen-Informationsdatei* einen neuen Namen für die neue Arbeitsgruppen-Informationsdatei ein, und klicken Sie dann auf OK. Die Arbeitsgruppen-Informationsdatei wird standardmäßig im Access-Ordner installiert. Wenn Sie die Datei an einer anderen Stelle speichern möchten, geben Sie einen Pfadnamen ein oder klicken auf *Durchsuchen*, um den neuen Pfad anzugeben.

Nach dem Erstellen einer neuen Arbeitsgruppen-Informationsdatei sind Sie automatisch an diese angeschlossen, worauf das abschließende Dialogfeld Sie auch hinweist.

Notieren Sie sich die genaue Schreibweise des Namens, der Firma und der Arbeitsgruppen-Identifikationsnummer, einschließlich der Groß-/Kleinschreibung für alle drei Einträge. Wenn Sie die Arbeitsgruppen-Informationsdatei neu erstellen müssen, weil sie beispielsweise beschädigt wurde, müssen Sie diese Informationen genau wie beim ersten Erstellen wieder eingeben. Ohne diese Einträge wäre eine Arbeitsgruppen-Informationsdatei nicht so reproduzierbar, daß sie die Funktion der alten erfüllt.

Die neue Arbeitsgruppen-Informationsdatei wird beim nächsten Starten von Access verwendet. Alle Benutzerkonten, Gruppenkonten oder Kennwörter, die Sie erstellen, werden in der neuen Arbeitsgruppen-Informationsdatei gespeichert. Wenn weitere Benutzer an die in der neuen Arbeitsgruppen-Informationsdatei definierte Arbeitsgruppe angeschlossen werden sollen, kopieren Sie die Datei in einen gemeinsam genutzten Ordner und lassen dann jeden dieser Benutzer zum Anschließen an die Arbeitsgruppen-Informationsdatei das Dienstprogramm *Arbeitsgruppen-Administrator* – wie oben beschrieben – ausführen.

Sich einer Arbeitsgruppe anschließen

Um sich einer anderen Arbeitsgruppe anzuschließen, gehen Sie folgendermaßen vor:

- Beenden Sie ggf. alle laufenden Access-Sitzungen auf Ihrem Rechner.
- Starten Sie das Dienstprogramm *Arbeitsgruppen-Administrator* wie oben beschrieben, vgl. den vorangehenden Punkt. Dann zeigt sich das Dialogfeld *Arbeitsgruppen- Administrator*, vgl. Bild 35.1.
- Klicken Sie auf die Schaltfläche *Beitreten*, und wählen Sie im Dialogfeld *Arbeitsgruppen-Informationsdatei* die Datei aus, der Sie sich anschließen möchten.

Beim nächsten Start von Access werden Sie als Mitglied der entsprechenden Arbeitsgruppe behandelt.

35.0.3 Mehrbenutzeroptionen

Für das Arbeiten in einer Mehrbenutzerumgebung lassen sich verschiedene Optionen festlegen. Dies geschieht mit der Registerkarte *Weitere* im Dialogfeld *Optionen* des gleichlautenden Befehls aus dem Menü *Extras*, vgl. Bild 35.3.

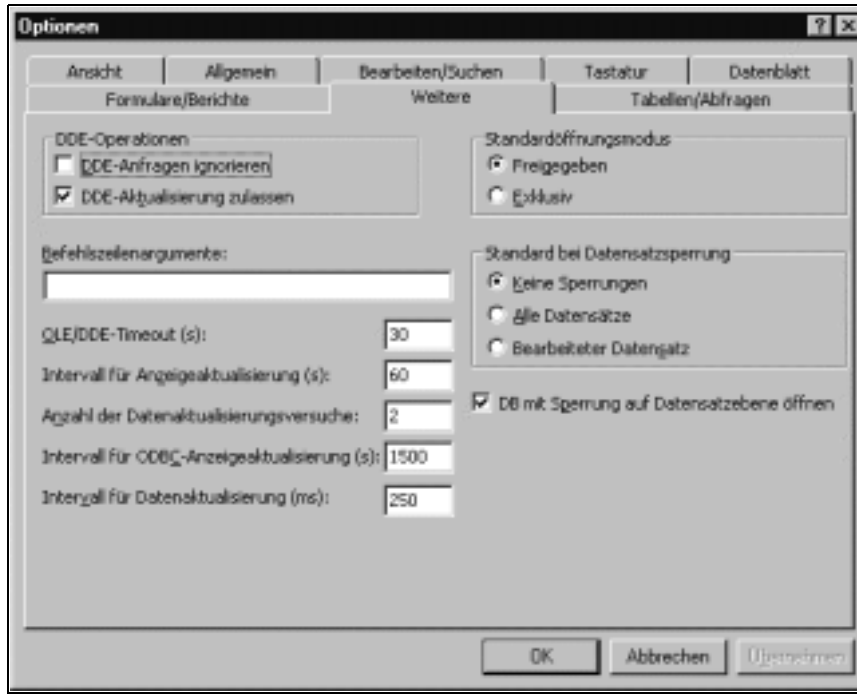


Bild 35.3: Registerkarte »Weitere« im Dialogfeld »Optionen« des gleichnamigen Befehls aus dem Menü Extras mit mehreren Optionen für das Arbeiten in einer Mehrbenutzerumgebung

Die Optionen, die das Arbeiten in einer Mehrbenutzerumgebung betreffen, haben die nachfolgende Bedeutung:

Standard bei Datensatzsperrung

Keine Sperrungen. Access sperrt den Datensatz, den Sie gerade bearbeiten, nicht. Wenn Sie versuchen, Änderungen an einem Datensatz, den ein anderer Benutzer ebenfalls geändert hat, zu speichern, zeigt Access eine Meldung mit den folgenden Optionen an: *Überschreiben der Änderungen des anderen Benutzers*, *Kopieren Ihrer Version des Datensatzes in die Zwischenablage* oder *Verwerfen Ihrer Änderungen*. Wenn Sie diese Option wählen, stellen Sie zwar sicher, daß Datensätze stets bearbeitet werden können, jedoch kann dies zu Konflikten bei der Bearbeitung führen.

Bearbeiteter Datensatz. Access sperrt den Datensatz, den Sie gerade bearbeiten, so daß kein anderer Benutzer ihn ändern kann. Falls ein anderer Benutzer versucht, einen von Ihnen gesperrten Datensatz zu bearbeiten, zeigt Access im Datenblatt des anderen Benutzers den Indikator *Gesperrter Datensatz* an. Diese Strategie stellt sicher, daß Sie begonnene Änderungen immer zu Ende führen können. Sie empfiehlt sich, wenn es bei der Bearbeitung von Datensätzen nur selten zu Kollisionen wegen gleichzeitigen Datenzugriffs kommt.

Alle Datensätze. Access sperrt alle Datensätze im Formular oder Datenblatt und in den zugrundeliegenden Tabellen, die Sie bis zum Schließen des Formulars oder Datenblattes bearbeiten, so daß sie von niemand anderem bearbeitet oder gesperrt werden können. Diese Strategie ist sehr weitreichend und einschränkend. Sie empfiehlt sich nur dann, wenn Sie Ihrer Bearbeitung absolute Priorität vor allen anderen Benutzern beimessen.

Wenn Sie Daten in einer verknüpften SQL-Datenbanktabelle unter Verwendung von ODBC bearbeiten, werden die Datensätze nicht von Access, sondern vom SQL-Server nach dessen Regeln gesperrt. In diesem Fall verhält sich Access so, als ob die Einstellung *Keine Sperrungen* gewählt wurde, unabhängig davon, welche Einstellung Sie tatsächlich gewählt haben.

Standard-Öffnungsmodus

Wenn Sie zulassen möchten, daß andere Benutzer die Datenbank auch dann öffnen können, wenn Sie diese geöffnet haben, aktivieren Sie das Kontrollkästchen *Freigegeben*. Wenn Sie andererseits alleinigen Zugriff auf eine Datenbank, die Sie geöffnet haben, haben möchten, aktivieren Sie *Exklusiv*.

Die Option legt nur den Standard-Öffnungsmodus fest. Unabhängig davon können Sie eine Datenbank im exklusiven und/oder schreibgeschützten Modus öffnen, auch wenn Sie als Standardmodus die gemeinsame Nutzung angeben, und auch das Umgekehrte gilt: Wählen Sie den Befehl *Öffnen* aus

dem Menü *Datei*, und bestätigen Sie das Dialogfeld *Öffnen* mit dem Befehl *Exklusiv öffnen*, den Sie wählen können, wenn Sie die Dropdown-Liste der Schaltfläche *Öffnen* aufschlagen, vgl. Bild 35.4.

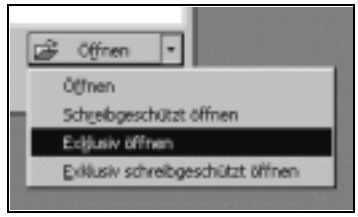


Bild 35.4: Aufgeschlagene Dropdown-Liste der Schaltfläche »Öffnen« im gleichnamigen Dialogfeld

Minimieren von Konflikten bei der Datensperrung

Durch das Festlegen von fünf Optionen können Sie Konflikte mit anderen Benutzern bei der Datensperrung wenn nicht vermeiden so doch minimieren.

OLE/DDE-Timeout. Diese Option legt fest, nach wieviel Sekunden Access versuchen soll, eine gescheiterte OLE- oder DDE-Operation erneut auszuführen. Möglich sind Werte zwischen 0 und 300.

Intervall für Anzeigeaktualisierung. Diese Option legt die Anzahl der Sekunden fest, nach deren Ablauf Access automatisch Datensätze in der Datenblattansicht oder Formularansicht aktualisiert. Sie können Werte zwischen 0 und 32.766 eingeben. Wenn Sie das aktuelle Datenblatt oder Formular aktualisieren, werden die Datensätze nicht neu sortiert, keine neuen Datensätze hinzugefügt oder gelöschte Datensätze entfernt. Um derartige Änderungen zu aktualisieren, müssen Sie die Abfrage für die Datensätze, die dem Datenblatt oder Formular zugrunde liegen, erneut ausführen.

Anzahl der Datenaktualisierungsversuche. Diese Option legt fest, wie oft Access versuchen soll, einen geänderten Datensatz, der von einem anderen Benutzer gesperrt ist, zu speichern. Sie können Werte zwischen 0 und 10 eingeben.

Intervall für ODBC-Anzeigeaktualisierung. Diese Option legt das Intervall fest, nach deren Ablauf Access automatisch die Datensätze aktualisiert, auf die Sie mit ODBC zugreifen. Gültige Werte sind 1 bis 32.766 Sekunden.

Intervall für Datenaktualisierung. Diese Option legt die Anzahl der Millisekunden fest, nach deren Ablauf Access automatisch versucht, einen geänderten Datensatz, der von einem anderen Benutzer gesperrt ist, zu speichern. Gültige Werte sind 0 bis 1.000.

Kapitel 36

Zugriffsrechte für Datenbank oder einzelne Objekte festlegen

36.1 Das Sicherungskonzept

Das Sicherungskonzept von Access ermöglicht es, die Objekte einer Datenbank vor dem unberechtigten Zugriff von Benutzern zu schützen. Dabei können objektweise differenzierte Zugriffsrechte vergeben werden – beispielsweise für eine Tabelle nur *Daten lesen* oder für ein Formular zwar den *Entwurf lesen*, jedoch nicht *Öffnen/Ausführen*, um die Daten zu lesen.

36.1.1 Sicherungskomponenten

Eine Anwendung wird durch das Zusammenwirken der folgenden Komponenten gesichert:

- **Kennwortschutz für Datenbank**. Sie können die Datenbank durch Festlegen eines Datenbankkennworts sichern. Wenn Sie allein diese Strategie wählen und keine weitere Zugriffsverwaltung mit Hilfe der in den folgenden Punkten geschilderten Zugriffsrechte für Benutzer in Arbeitsgruppen pflegen, können Sie nur die Datenbank insgesamt schützen, jedoch nicht differenziert einzelne ihrer Objekte. Jedem Benutzer, der Kenntnis des Datenbankkennworts hat, stehen alle Möglichkeiten des Datenbankzugriffs offen – einschließlich der Möglichkeit, das Kennwort selbst zu ändern und damit ggf. Dritten den Zugriff auf die Datenbank zu verwehren. Aus diesem Grunde empfiehlt sich das alleinige Vertrauen auf die Sicherung durch ein Datenbankkennwort, wenn Sie mit einer überschaubaren Anzahl von Mitbenutzern in einer vertrauensvollen Umgebung arbeiten. Diese Art der Sicherung hat den Vorteil, daß sie sehr einfach einzurichten und zu verwalten ist.

- **Gesicherte Arbeitsgruppen**. Auf die Bedeutung von Arbeitsgruppen für den gemeinsamen Datenzugriff auf dieselbe Datenbank in einer Mehrbenutzerumgebung wurde prinzipiell bereits in Kap. 35, *Arbeiten in einer Mehrbenutzerumgebung*, eingegangen. Unter dem Gesichtspunkt der Sicherung von Anwendungen spielt die Arbeitsgruppe eine zentrale Rolle. Sie kann diese Rolle allerdings erst spielen, wenn für den Benutzer *Administrator* der Arbeitsgruppe ein Kennwort angegeben ist, vgl. dazu in diesem Kapitel Punkt 36.5, *Kennwort ändern und löschen*. Ein Mitglied der Gruppe *Administratoren* kann stets alle Berechtigungen für alle Datenbanken und Objekte der Arbeitsgruppe erlangen.
- **Konten**. Um einzelnen Personen Berechtigungen für bestimmte Objekte einer Datenbank zuweisen zu können, muß für diese zunächst ein Benutzerkonto in der Arbeitsgruppe eingerichtet werden. Der Begriff *Konto* ist sprachlich vielleicht etwas unglücklich gewählt. Gemeint ist damit jedenfalls eine Registrierung von Benutzern in der Arbeitsgruppe, zu der die Anwendung gehört.
- **Besitzer**. Jede Datenbank und jedes Objekt darin hat einen Besitzer. Dieser hat stets alle Rechte an dem betreffenden Objekt bzw. der Datenbank. Besitzer einer Datenbank ist zunächst immer der Benutzer, der die Datenbank neu erstellt hat. Der Besitzer der Datenbank und/oder eines einzelnen Objekts aus der Datenbank kann nachträglich leicht geändert werden.
- **Berechtigungszuweisungen**. Berechtigungszuweisungen werden einzelnen Benutzern (oder Benutzergruppen) für einzelne Objekte sowie die Datenbank als Objekt zugewiesen.
- **Kennwörter zuweisen**. Access weist dem Standardbenutzerkonto *Administrator* und jedem neuen Benutzerkonto, das Sie in Ihrer Arbeitsgruppe erstellen, ein leeres Kennwort zu. Sie können eine Datenbank nur dann richtig schützen, wenn Sie für die folgenden Konten ein Kennwort festlegen:
  - für das Benutzerkonto *Administrator*, zum Einschalten des Anmeldeprozesses,
  - für das Benutzerkonto, das Besitzer der Datenbank und ihrer Objekte ist,
  - für alle Benutzerkonten, die Sie zur Gruppe *Administratoren* hinzufügen.

  Darüber hinaus können Sie Kennwörter für Konten festlegen, die Sie für Benutzer erstellen, oder Benutzer anweisen, selbst Kennwörter festzulegen.

  Jeder Benutzer kann sein eigenes Benutzerkontenkennwort erstellen oder ändern. Wenn ein Benutzer jedoch sein Kennwort vergißt, kann dieses nur von einem Konto, das der Gruppe *Administratoren* zugeordnet ist, gelöscht werden.

Bedeutsam ist das *Zusammenwirken* dieser Komponenten des Sicherungskonzepts, insbesondere das Zusammenwirken von Datenbank und Arbeitsgruppe: Wenn eine Datenbank bei eingeschalteter Sicherung der Arbeitsgruppe erstellt wird, haben Benutzer einer anderen Arbeitsgruppe keinen Zugriff auf ihre Objekte. Daher müssen Sie bei der Installation einer gesicherten Datenbank auf einem anderen Rechner stets auch die zugehörige Arbeitsgruppen-Informationsdatei (vgl. dazu Kap. 35, *Arbeiten in einer Mehrbenutzerumgebung*) mit installieren, deren Benutzerkonten Sie dann freilich nachträglich verändern können.

Welche Sicherungsinformationen werden wo gespeichert?

- Die Informationen über Benutzer und Benutzergruppen – das sind vor allem die Kontonamen, Kennwörter und Zuordnungen zu Benutzergruppen – werden in der Arbeitsgruppen-Informationsdatei der Arbeitsgruppe gespeichert. Die Informationen über Besitzer und Berechtigungen sowie das Datenbankkennwort werden in der jeweiligen Datenbank gespeichert. Da eine Berechtigung stets an einen bestimmten Benutzer der bestimmten Arbeitsgruppe gekoppelt ist, speichert die Datenbank also auch die Namen der Benutzer. Wenn die Arbeitsgruppen-Informationsdatei der bestimmten Arbeitsgruppe verlorengeht, können Sie auf gesicherte Objekte der mit dieser Arbeitsgruppe verbundenen Datenbank im allgemeinen nicht mehr zugreifen, es sei denn, Sie rekonstruieren die Arbeitsgruppen-Informationsdatei, vgl. dazu die Online-Hilfe.

36.1.2 Datenbankkennwort festlegen, löschen und ändern

Festlegen

Um für eine Datenbank ein Kennwort festzulegen, gehen Sie folgendermaßen vor:

- Sorgen Sie zunächst ggf. dafür, daß kein anderer Benutzer die Datenbank geöffnet hat, und öffnen Sie die Datenbank dann exklusiv. Dies geschieht beispielsweise mit dem Befehl *Öffnen* aus dem Menü *Datei*, dessen Dialogfeld Sie mit der Schaltfläche *Exklusiv öffnen* bestätigen können (die aufschlagbare Schaltfläche *Öffnen* enthält diesen Befehl).
- Wählen Sie aus dem Menü *Extras* den Befehl *Sicherheit* und dessen Unterbefehl *Datenbankkennwort zuweisen*. Dann wird das gleichnamige Dialogfeld angezeigt, vgl. Bild 36.1.
- Geben Sie das Kennwort ein, und wiederholen Sie die Eingabe zum Bestätigen. Beachten Sie die Groß-/Kleinschreibung, denn diese ist relevant.
- Bestätigen Sie das Dialogfeld mit OK. Beim nächsten Öffnen der aktuellen Datenbank wird jeder Benutzer nach dem Datenbankkennwort gefragt.

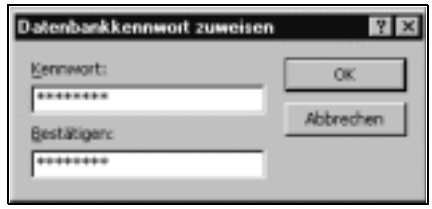


Bild 36.1: Dialogfeld »Datenbankkennwort zuweisen«

Beachten Sie, daß jeder Benutzer, der das Datenbankkennwort kennt, dieses löschen und nachfolgend ein anderes festlegen kann. Damit könnte ein Benutzer möglicherweise gewollt oder unbewußt andere Benutzer daran hindern, die Datenbank zu öffnen. Wenn Sie verhindern wollen, daß andere Benutzer das Datenbankkennwort ändern können, müssen Sie Benutzerkonten führen und allen Benutzern, für die Sie diese Möglichkeit ausschließen wollen, die Berechtigung zum *Verwalten* der Datenbank entziehen.

Löschen

Für eine Datenbank, für die ein Datenbankkennwort festgelegt ist, wird der Unterbefehl *Datenbankkennwort zuweisen* auf den Unterbefehl *Datenbankkennwort löschen* umgeschaltet. Rufen Sie diesen Unterbefehl zum Befehl *Sicherheit* aus dem Menü *Extras* auf, und geben Sie in das Dialogfeld *Datenbankkennwort löschen* das bestehende Kennwort ein.

Datenbankkennwort ändern

Sie können das Datenbankkennwort nicht direkt in einem einzigen Dialogfeld ändern. Gehen Sie vielmehr folgendermaßen vor:

- Löschen Sie das bestehende Datenbankkennwort.
- Legen Sie ein neues Datenbankkennwort fest.

36.2 Berechtigungsprüfung der Arbeitsgruppe ein- und abschalten

Für jede Arbeitsgruppe wird von Access automatisch standardmäßig das Benutzerkonto mit dem speziellen Namen *Administrator* eingerichtet. Dies ist, solange Sie nicht ausdrücklich weitere Benutzerkonten einrichten, auch der einzige Benutzer einer Arbeitsgruppe. Nach der Installation von Access ist für das Benutzerkonto *Administrator* zunächst kein Kennwort festgelegt. Dies bedeutet, daß Access beim Start des Programms nach keiner Berechtigung fragt, so daß jeder Benutzer – gleichgültig, ob für ihn bei der Arbeitsgruppe ein Benutzerkonto eingerichtet ist oder nicht – Access für diese Arbeitsgruppe starten kann. Ohne es zu merken, wird dann jeder Benutzer

als *Administrator* angemeldet. Jeder neu erstellten Datenbank wird in dieser Situation der Besitzer *Administrator* zugeordnet. Die Berechtigungsprüfung, das System Access als berechtigter Benutzer der Arbeitsgruppe aufrufen zu dürfen, erfolgt erst, wenn für den Benutzer *Administrator* ein Kennwort festgelegt ist. Solange dies nicht geschehen ist, ist jede Datenbank ungeschützt – es sei denn, Sie haben ein Datenbankkennwort festgelegt.

Berechtigungsprüfung einschalten

Um die Berechtigungsprüfung für die Arbeitsgruppe einzuschalten, gehen Sie daher folgendermaßen vor:

- Starten Sie Access für die Arbeitsgruppe, für deren Benutzer *Administrator* Sie ein Kennwort eingeben wollen. Dies bedeutet praktisch: Starten Sie Access von einem PC aus, der der entsprechenden Arbeitsgruppe angeschlossen ist. Sie werden dabei implizit als Benutzer *Administrator* angemeldet.
- Wählen Sie aus dem Menü *Extras* den Befehl *Sicherheit* und dessen Unterbefehl *Benutzer- und Gruppenkonten*.
- Aktivieren Sie im Dialogfeld *Benutzer- und Gruppenkonten die Registerkarte Anmeldungskennwort ändern (vgl.* Bild 36.2), und geben Sie das Kennwort als *Neues Kennwort und zum Bestätigen ein. In der hier angenommenen Situation bleibt das Einstellungsfeld Altes Kennwort leer, weil bisher keines vergeben war.*

Bild 36.2: Dialogfeld »Kennwort ändern«, hier für den Benutzer »Administrator«

Beim nächsten Öffnen einer Datenbank für diese Arbeitsgruppe werden Sie mit dem Dialogfeld *Anmelden* (vgl. Bild 36.3) zur Angabe eines Benutzernamens und dessen Kennwort aufgefordert. Sie können für diese Arbeitsgruppe keine Datenbank mehr öffnen, ohne einen angemeldeten Benutzer (ein eingerichtetes Benutzerkonto) und sein Kennwort – sofern dieses definiert ist – anzugeben. Dies kann allerdings ein anderer Benutzer als der *Administrator* sein.

Bild 36.3: Dialogfeld »Anmelden«

Berechtigungsprüfung abschalten

Sie heben die Berechtigungsprüfung für den Zugang zur Arbeitsgruppe wieder auf, wenn Sie das Kennwort für den Benutzer *Administrator* löschen. Dies geschieht ebenfalls mit dem Dialogfeld *Kennwort ändern*, vgl. oben Bild 36.2. Geben Sie in diesem Falle das alte Kennwort an, und lassen Sie die beiden anderen Eingabefelder leer. Bei den folgenden Öffnungsversuchen einer Datenbank für diese Arbeitsgruppe wird das Dialogfeld *Anmelden* nicht mehr eingeblendet, bis Sie wieder ein Kennwort für den Benutzer *Administrator* angeben.

36.3 Administrator-, Benutzer- und Gruppenkonten erstellen und verwalten

36.3.1 Konten erstellen

Benutzerkonto erstellen

Gehen Sie folgendermaßen vor:

- Starten Sie Access für die Arbeitsgruppe, für die Sie ein neues Konto erstellen wollen. Wenn für die Arbeitsgruppe die Berechtigungsprüfung eingeschaltet ist, melden Sie sich als ein Benutzer an, der der Gruppe *Administratoren* zugeordnet ist, denn nur Mitglieder dieser Gruppe sind berechtigt, Benutzerkonten zu verwalten. Der Benutzer *Administrator* ist standardmäßig dieser Gruppe zugeordnet. Es kann jedoch sein, daß ihm die Zuordnung zur Gruppe *Administratoren* entzogen worden ist. Bei einer Arbeitsgruppe ohne eingeschaltete Berechtigungsprüfung werden Sie automatisch als *Administrator* angemeldet.

- Wählen Sie aus dem Menü *Extras* den Befehl *Sicherheit* und dessen Unterbefehl *Benutzer- und Gruppenkonten*.
- Aktivieren Sie im Dialogfeld *Benutzer- und Gruppenkonten die Registerkarte Benutzer.*
- KLICKEN SIE AUF DIE SCHALTFLÄCHE *Neu. Dann zeigt sich das Dialogfeld Neue(r) Benutzer/Gruppe*, vgl. Bild 36.4.

Bild 36.4: Dialogfeld »Neue(r) Benutzer/Gruppe«

- Geben Sie einen Benutzernamen und eine *Persönliche ID* für diesen Benutzer ein. Diese muß mindestens vier und darf höchsten 20 Zeichen lang sein. Sie ist nicht identisch mit dem Kennwort für den Benutzer. Access verwendet die *Persönliche ID* zur Identifizierung des Benutzers. Für den Benutzernamen und die *Persönliche ID* sind Groß- und Kleinbuchstaben relevant. Bestätigen Sie mit OK.

Sie sollten sich die genaue Schreibweise des Benutzernamens und der *Persönlichen ID* aufschreiben. Wenn Sie ein gelöschtes Benutzerkonto oder einen Benutzer, der in einer anderen Arbeitsgruppe existiert, wiederherstellen bzw. erneut erstellen möchten, muß der Benutzername und die *Persönliche ID* unter Berücksichtigung der Groß-/Kleinschreibung eingegeben werden und exakt dem ursprünglichen Namen entsprechen. Wenn Sie sich bei Access anmelden, braucht beim Eingeben des Benutzernamens im Dialogfeld *Anmelden* dagegen die Groß-/Kleinschreibung nicht beachtet zu werden. Es kann durchaus sinnvoll sein, die identisch gleiche Benutzerkennung in zwei verschiedenen Arbeitsgruppen-Informationsdateien gespeichert zu haben: Nehmen Sie beispielsweise an, derselbe Mitarbeiter solle, weil er viel auf Reisen ist, über ein Netzwerk von verschiedenen Orten aus auf dieselbe geschützte Datenbank, die an nur einem bestimmten Ort liegt, mit stets denselben Rechten zugreifen können. Dann ist es sinnvoll, diesen Benutzer in verschiedenen Arbeitsgruppen-Informationsdateien mit derselben Kennung zu führen.

Ein neuer Benutzer ist zunächst ohne Kennwort. Der neue Benutzer muß sich daher im Dialogfeld *Anmelden* beim erstenmal ohne Kennwort anmelden. Dann sollte er während der Sitzung ein Kennwort festlegen. Der Administrator, der das neue Konto eingerichtet hat, kann für einen anderen Benutzer kein Kennwort festlegen – es sei denn, er meldet sich als neuer Benutzer an. Es gibt auch keine Möglichkeit, die Angabe eines Kennwortes

zu erzwingen. Dies kann gravierende Folgen haben, wenn Sie einen Benutzer der Gruppe *Administratoren* zuordnen, denn dann hat er alle Rechte eine Administrators.

Gruppenkonto erstellen

Um ein neues Gruppenkonto zu erstellen, verfahren Sie analog zum Erstellen eines neuen Benutzerkontos, aktivieren Sie jedoch im Dialogfeld *Benutzer- und Gruppenkonten* die Registerkarte *Gruppen*, und klicken Sie dann auf die Schaltfläche *Neu*.

Administratorkonto erstellen

Es gibt kein spezifisches Administratorkonto. De facto können Sie jedoch mehr als ein Administratorkonto einrichten. Erstellen Sie zu diesem Zweck ein beliebiges neues Benutzerkonto, und ordnen Sie dieses der Gruppe *Administratoren* zu, vgl. den folgenden Punkt 36.3.2, *Konten verwalten*.

36.3.2 Konten verwalten

Konten können nur von einem Benutzer, der der Gruppe *Administratoren* zugeordnet ist, verwaltet werden. Um Kontomerkmale zu ändern, gehen Sie folgendermaßen vor:

Bild 36.5: Dialogfeld »Benutzer- und Gruppenkonten« mit aktivierter Registerkarte »Benutzer«

- Wählen Sie aus dem Menü *Extras* den Befehl *Sicherheit* und dessen Unterbefehl *Benutzer- und Gruppenkonten*.
- Aktivieren Sie im Dialogfeld *Benutzer- und Gruppenkonten die Registerkarte Benutzer, vgl.* Bild 36.5.
- **Benutzer löschen**. Markieren Sie den betreffenden Benutzer im Dropdown-Listenfeld *Name*, und klicken Sie auf die Schaltfläche *Löschen*.
- **Benutzer einer Gruppe zuordnen**. Wenn Sie mehrere Benutzer derselben Gruppe zuordnen, erleichtern Sie sich die Arbeit als Administrator, denn Sie können Berechtigungen für Gruppen vergeben, die dann automatisch den Benutzern zustehen, die der Gruppe zugeordnet sind. Markieren Sie den betreffenden Benutzer im Dropdown-Listenfeld *Name*. Markieren Sie dann die zuzuordnende Gruppe in der Liste *Verfügbare Gruppen*, und klicken Sie auf die Schaltfläche *Hinzufügen*>>. Zum Entfernen einer Gruppe von einem Benutzer verfahren Sie entsprechend umgekehrt.
- **Kennwort löschen**. Markieren Sie den betreffenden Benutzer im Dropdown-Listenfeld *Benutzer*, und klicken Sie auf die Schaltfläche *Kennwort löschen*. Diese Aktion ist im allgemeinen geboten, wenn der Benutzer sein Kennwort vergessen hat. Bedenken Sie jedoch, daß, solange kein neues Kennwort für den Benutzer festgelegt ist, jedermann durch Angabe des im allgemeinen nicht geheimen Benutzernamens sich unter diesem Benutzer anmelden kann. Sie können sich jedoch selbst unter diesem Benutzerkonto anmelden, ein Kennwort angeben und dieses dem Benutzer dann mitteilen.

36.4 Berechtigungen zuweisen und entfernen

Berechtigungen zuweisen

Berechtigungen werden für bestimmte Benutzer oder Gruppen auf bestimmte Objekte einer Datenbank vergeben. Verfahren Sie wie folgt:

- Wählen Sie aus dem Menü *Extras* den Befehl *Sicherheit* und dessen Unterbefehl *Benutzer- und Gruppenberechtigungen*.
- Aktivieren Sie im Dialogfeld *Benutzer- und Gruppenberechtigungen die Registerkarte Berechtigungen, vgl.* Bild 36.6.
- Sie vergeben eine Berechtigung für ein oder mehrere Objekte eines Objekttyps. Daher müssen Sie zunächst den Objekttyp wie z.B. Formular oder Tabelle wählen. Daraufhin markieren Sie in der Liste der angezeigten Objektnamen das oder die Objekte, für die Sie eine Berechtigung zuweisen wollen. Eine Mehrfachauswahl ist mit den Tasten [Strg] und [⇧] möglich. Wählen Sie die Option *Benutzer* oder *Gruppen*, je nachdem, wem Sie eine Berechtigung zuweisen wollen, und markieren Sie dann den Benutzer bzw. die Gruppe in der Liste *Benutzer/Gruppenname*. Aktivieren Sie

schließlich die Kontrollkästchen der gewünschten Berechtigungen, und bestätigen Sie dann mit *Übernehmen*. Bei geöffnetem Dialogfeld können Sie auf diese Weise mehrere verschiedene Zuweisungen nacheinander vornehmen.

Bild 36.6: Dialogfeld »Benutzer- und Gruppenberechtigungen«, Registerkarte »Berechtigungen«

Berechtigungen entfernen

Sie entfernen eine Berechtigung, indem Sie eine neue Berechtigung mit keinerlei (oder geänderten) Rechten vergeben, wie im letzten Punkt beschrieben.

Wenn Sie einem Benutzer Berechtigungen entziehen, kann die Änderung wirkungslos sein, weil der Benutzer einer Gruppe zugeordnet ist, die diese Berechtigung besitzt. In diesem Falle müssen Sie entweder die Gruppenberechtigung ebenfalls ändern, oder Sie heben die Zuordnung des Benutzers zu der Gruppe auf.

36.5 Kennwort ändern und löschen

Der Benutzer, der der Gruppe *Administratoren* zugeordnet ist, kann das Kennwort jedes anderen Benutzers löschen, jedoch nicht ändern, vgl. dazu oben, Punkt 36.3.2, *Konten verwalten*, Aufzählungspunkt *Kennwort löschen*.

Ein Benutzer kann sein eigenes Kennwort mit dem Befehl *Sicherheit*, Unterbefehl *Benutzer- und Gruppenkonten*, aus dem Menü *Extras* ändern. Um ein Kennwort zu löschen, bleiben die beiden Eingabefelder *Neues Kennwort* und *Bestätigen* im Dialogfeld *Kennwort ändern* leer.

36.6 Besitzer ändern

Der Besitzer eines Objekts oder einer Datenbank hat stets alle Rechte daran. Jedem Objekt einer Datenbank kann einer der Benutzer oder eine der Gruppen der Arbeitsgruppe als Besitzer zugeordnet werden. Dies geschieht mit dem Befehl *Sicherheit*, Unterbefehl *Benutzer- und Gruppenberechtigungen*, aus dem Menü *Extras*, Registerkarte *Besitzer ändern*. Das entsprechende Dialogfeld ist in Bild 36.7 wiedergegeben. Wählen Sie darin die gewünschte Zuordnung aus. Mit den Tasten [Strg] und [⇧] können Sie eine Mehrfachmarkierung der Objekte vornehmen.

Bild 36.7: Dialogfeld »Benutzer- und Gruppenberechtigungen«, Registerkarte »Besitzer ändern«

Wenn für eine Arbeitsgruppe die Berechtigungsprüfung nicht eingeschaltet ist, weil für den Benutzer *Administrator* kein Kennwort angegeben ist, meldet das System automatisch *Administrator* als Benutzer an, ohne daß der tatsächliche Benutzer dies merkt. Neu erstellten Datenbanken wird daher automatisch der Benutzer *Administrator* als Besitzer zugeordnet.

36.7 Datenbank verschlüsseln

Auf die Objekte einer geschützten Datenbank kann nur von den berechtigten Benutzern zugegriffen werden. Diese Sicherung besteht allerdings nur, wenn Sie mit dem Programm Access arbeiten. Sofern Sie eine Access-Datenbank mit einem anderen Programm – beispielsweise einem Hilfsprogramm oder einem Texteditor – öffnen, ist das in den vorangehenden Punkten beschriebene Sicherungskonzept nicht wirksam. Um zu verhindern, daß jemand mit fremden Programmen Teile einer Datenbank lesen kann, bietet Access die Möglichkeit, die Datenbank zu verschlüsseln. Eine verschlüsselte Datenbank kann weiterhin mit Access vollständig bearbeitet werden. Allerdings wird die Leistung der Anwendung dann eingeschränkt sein.

Verschlüsseln

Um eine Datenbank zu verschlüsseln, gehen Sie folgendermaßen vor:

- Schließen Sie ggf. die zu verschlüsselnde Datenbank. Die Datenbank darf in einer Mehrbenutzerumgebung auch von keinem anderen Benutzer geöffnet sein.
- Wählen Sie den Befehl *Sicherheit*, Unterbefehl *Datenbank ver-/entschlüsseln*, aus dem Menü *Extras*.
- Wählen Sie im Dialogfeld *Datenbank ver-/entschlüsseln* die zu verschlüsselnde Datenbankdatei aus, und bestätigen Sie mit OK.
- Geben Sie im Dialogfeld *Datenbank verschlüsseln als* einen neuen Namen an, wenn Sie die unverschlüsselte Version behalten wollen, oder geben Sie den Namen der zu verschlüsselnden Datenbankdatei an. (Falls der Verschlüsselungsvorgang fehlerhaft ist, wird die vorhandene Datenbankdatei nicht überschrieben.) Bestätigen Sie mit OK.

Entschlüsseln

Um eine verschlüsselte Datenbankdatei wieder zu entschlüsseln, gehen Sie denselben Weg, wie zum Verschlüsseln beschrieben. Access merkt nach der Auswahl der Datei im Dialogfeld *Datenbank ver-/entschlüsseln*, wenn die markierte und mit OK bestätigte Datenbank verschlüsselt ist, so daß sie im weiteren Fortgang automatisch entschlüsselt wird. Das zweite Dialogfeld trägt dann auch den Namen *Datenbank entschlüsseln als*.

Kapitel 37

Access-Projekte als Client/ Server-Datenbank: Grundlagen

37.1 Bedeutung und Funktionsweise von Microsoft Access-Projekten

Access-Projekte stellen die hervorragende Neuerung in Access 2000 dar. Ein äußerlich erkennbarer Unterschied zu den bisherigen Access-Datenbanken, deren Datei die Endung *.mdb* besitzt, besteht darin, daß Access-Projekte in Dateien mit der Endung *.adp* (steht für Access Data Project) gespeichert werden. Wesentlicher ist jedoch ein anderer, konzeptioneller Unterschied: In einer Access-Datenbank werden die Daten von der *JetEngine*, der in Access integrierten Datenbankmaschine, verwaltet. In einem Access-Projekt wird die JetEngine statt dessen durch einen SQL-Server ersetzt, wobei dies Microsoft SQL Server der Version 6.5 oder 7.0 oder MSDE (Microsoft Data Engine) sein kann. MSDE ist im Lieferumfang von Access 2000 enthalten, muß jedoch eigenständig installiert werden, vgl. dazu genauer unten Punkt 37.2, *MSDE installieren, starten und nutzen*. SQL Server haben die Eigenschaft, Daten nicht nur zu verwalten, sondern auch zu halten. Daher ergibt sich als Konsequenz, daß die in Access-Datenbanken vorhandenen Objekte *Tabellen* und *Abfragen* nicht physikalischer Bestandteil von Access-Projekten sind. Vielmehr liegt der gesamte Datenteil in Form der Objekte *Tabellen, Sichten, Gespeicherten Prozeduren* und *Datenbankdiagrammen* beim SQL Server, vgl. die Darstellung in Bild 37.1. Allerdings können diese Objekte von einem Access-Projekt aus mit Access-Werkzeugen komfortabel bearbeitet werden.

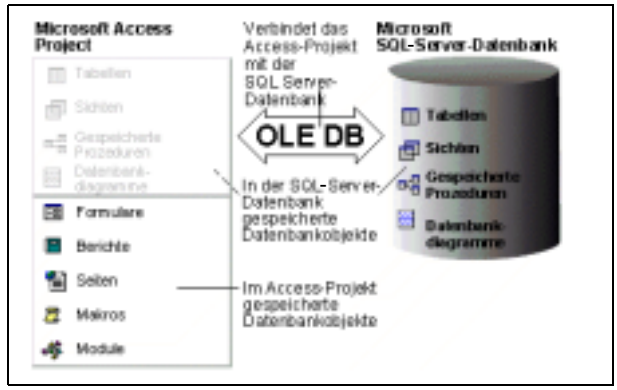


Bild 37.1: Zusammenwirken von Access-Projekt und SQL-Server-Datenbank in der Darstellung der Online-Hilfe. (Die Symbole für »Gespeicherte Prozeduren« und »Datenbankdiagramme« sind offensichtlich vertauscht.).

Obgleich sich Access-Datenbanken und Access-Projekte, wie gezeigt, konzeptionell wesentlich unterscheiden, ist ihre Bearbeitung weitgehend gleichartig. Insbesondere werden Formulare, Berichte, Datenzugriffsseiten, Makros und Module auf praktisch identische Weise bearbeitet. Sobald eine Verbindung mit der SQL-Server-Datenbank besteht, lassen sich auch die Tabellen, Sichten, Gespeicherten Prozeduren und Datenbankdiagramme anzeigen, erstellen, ändern und löschen. Dazu stehen dieselben Microsoft SQL Server-Entwurfswerkzeuge zur Verfügung, die auch in Verbindung mit *Microsoft Visual Interdev* und *Microsoft Visual Basic* verwendet werden. Die Benutzeroberfläche dieser Werkzeuge unterscheidet sich zwar von der der entsprechenden Werkzeuge in einer Access-Datenbank, sie ist jedoch ebenfalls sehr einfach zu bedienen. Allerdings unterscheidet sich die Datenverwaltungsphilosophie von SQL Servern z.T. deutlich von derjenigen in Access-Datenbanken. Dies zeigt sich insbesondere bei Abfragen, die bei SQL Servern wohl etwas komplizierter zu handhaben sind als bei Access-Datenbanken: Einer Auswahlabfrage einer Access-Datenbank entspricht im wesentlichen eine *Sicht* (View) des SQL Servers, wenngleich in eingeschränkter Weise. Dagegen finden die verschiedenen Arten von Aktionsabfragen einer Access-Datenbank ihre Entsprechung in *Gespeicherten Prozeduren*. Diese stellen andererseits nicht nur den Ersatz für die Aktionsabfragen dar, sondern ermöglichen komplexere und effizientere Abfragen und SQL-Aktionen, als sie in Access-Datenbanken realisierbar sind. Der Entwickler und/oder Administrator eines Access-Projekts ist daher über kurz oder lang genötigt, sich intensiver mit der Datenverwaltungsphilosophie des verwendeten SQL Servers zu beschäftigen.

Unterschiede der verwendbaren SQL Server Versionen

Access-Projekte können mit dem Microsoft SQL Server der Versionen 6.5 und 7.0 verbunden werden. SQL Server 7.0 gibt es in vier verschiedenen Versionen: *Desktop*, *Standard*, *Enterprise* und *MSDE*. Desktop und MSDE sind dazu ausgelegt, unter Windows 9x, Windows NT 4.0 Workstation oder Windows 2000 Workstation zu laufen, können allerdings auch unter einem Server betrieben werden, während Standard und Enterprise nur auf einem Windows NT 4.0 oder Windows 2000 Server laufen. Der wesentliche Unterschied zwischen MSDE und Desktop liegt darin, daß bei MSDE die Microsoft Management Console sowie weitere grafische Werkzeuge zur Datenverwaltung nicht zur Verfügung stehen. Statt dessen werden die Werkzeuge zur Datenverwaltung für MSDE von Access 2000 zur Verfügung gestellt. Diese weisen allerdings im Vergleich zu den entsprechenden Tools beim SQL Server nur eine eingeschränkte Funktionalität auf.

37.2 MSDE installieren, starten und nutzen

37.2.1 Merkmale von MSDE

MSDE ist ein SQL Server, der für die Verwendung auf kleinen Rechnersystemen und für kleinere Anwendungen konzipiert wurde, wobei das Wort *kleinere* im Vergleich zur großen Schwester Microsoft SQL Server 7.0 zu verstehen ist, denn im Vergleich zur JetEngine der Access-Datenbanken ist MSDE durchaus leistungsfähiger. MSDE basiert auf derselben Datenbankengine wie Microsoft SQL Server 7.0. Daher »... lassen sich die meisten Microsoft Access-Projekte oder Client/Server-Anwendungen ohne Änderungen unter beiden Versionen ausführen.«[1] Gegenüber dem SQL Server 7.0 weist MSDE folgende Beschränkungen auf:

- Die Größe einer Datenbank ist auf 2 GB beschränkt.
- Das System ist auf eine gleichzeitige Nutzung durch fünf oder sechs Benutzer optimiert, wenngleich durchaus mehr Benutzer unterstützt werden mit immer noch besserer Performance als bei der JetEngine.
- MSDE bietet keine Unterstützung für Parallelverarbeitung durch Symmetrical Multiprocessing (SMP) unter Windows 9x, unter Windows NT 4.0 oder 2000 werden maximal zwei parallele Prozessoren unterstützt.
- Bei Verwendung von Transaktionsreplikation kann MSDE nicht als Replikationsverleger agieren (aber durchaus als Replikationsabonnent auftreten).

1. Die Online-Hilfe, aus der dieses Zitat stammt, sagt leider nichts darüber aus, um welche ggf. notwendigen Änderungen es sich handeln könnte.

37.2.2 MSDE installieren

Obwohl MSDE im Lieferumfang von Access 2000 enthalten ist, wird es nicht zugleich mit Access 2000 installiert. Dies muß vielmehr als eigenständiger Vorgang erfolgen. Bevor Sie die Installation starten, sollten Sie sich über die folgenden Informationen im klaren sein:

- Da MSDE ein Microsoft SQL Server ist, kann er nicht gleichzeitig mit einer anderen Version von Microsoft SQL Server 7.0 auf demselben PC installiert sein.
- Die Installation von MSDE erfordert ca. 60 MB freien Speicherplatz auf der Platte und zusätzlich Platz für die später anzulegenden Datenbanken.
- Der Arbeitsspeicher unter Windows 9.x sollte mindestens 32 MB, besser 64 MB umfassen, um gleichzeitig MSDE und Office 2000 laufen zu lassen. Unter Windows NT 4.0 oder 2000 sind mindestens 64 MB erforderlich, um häufiges paging auf der Platte zu vermeiden.

☞ Wenn Sie MSDE unter Windows NT 4.0 installieren wollen, muß das System vorher mit *Service Pack 4* (*SP4*) aktualisiert worden sein. Eine gleichwertige Alternative für diesen Fall bietet auch *SP3* plus Ausführen des Programms *Hotfix.exe*, das sich auf der Office 2000 CD-ROM befindet. Unter Windows 9x oder 2000 sind keine derartigen Vorbereitungen erforderlich

Gehen Sie folgendermaßen vor, um MSDE zu installieren:

- Legen Sie die Office 2000 CD-ROM ins Laufwerk und starten Sie im Ordner *\Sql\x86\Setup* das Programm *Setupsql.exe*, indem Sie beispielsweise darauf doppelklicken.
- Unter Windows NT können Sie im ersten Dialogfeld wählen, ob Sie MSDE auf dem lokalen oder auf einem Remote-Computer installieren wollen. Ich gehe im folgenden von einem lokalen Computer aus. Bestätigen Sie ggf. mit *Weiter* >.
- Nach einem Willkommen-Dialogfeld, das Sie mit *Weiter* > bestätigen, können Sie im Dialogfeld *Benutzerinformationen* die voreingestellten Angaben für Ihren Namen und die Firma ändern oder bestätigen. Klicken Sie auf die Schaltfläche *Weiter* >.
- Das Dialogfeld *Setup-Typ* (vgl. Bild 37.2) informiert Sie über den erforderlichen und verfügbaren Speicherplatz. Mit Hilfe der beiden *Durchsuchen*-Schaltflächen können Sie die Zielordner, in die die MSDE-Programme und die Daten installiert werden, ändern. Am besten belassen Sie es bei dem jeweils voreingestellten Ordner C:\MSSQL7. Klicken sie auf die Schaltfläche *Weiter* >.
- Im Dialogfeld *Zeichensatz/Sortierreihenfolge/Unicode-Sortierreihenfolge* können Sie diverse Optionen für Zeichensätze und Sortieren einstellen. Die voreingestellte Auswahl, die in Bild 37.3 wiedergegeben ist, sollten sie im Zweifel mit *Weiter* > bestätigen.

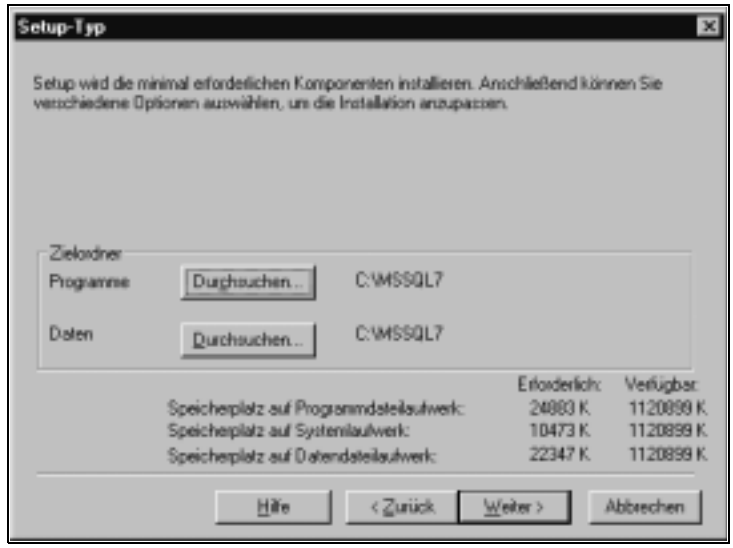


Bild 37.2: Dialogfeld »Setup-Typ«

Bild 37.3: Dialogfeld »Zeichensatz/Sortierreihenfolge/Unicode-Sortierreihenfolge«

- Das in Bild 37.4 wiedergegebene Dialogfeld *Netzwerkbibliotheken* gibt ebenfalls die voreingestellte Optionsauswahl wieder. Wie erkennbar könnten Sie hier weitere Netzwerkbibliotheken angeben. Die Anschlussnummer des TCP/IP-Sockets sollten Sie bei 1433 lassen, denn dies ist die Internet Assigned Number Authority (IANA) Portadresse für SQL Server. Klikken Sie auf die Schaltfläche *Weiter >*.

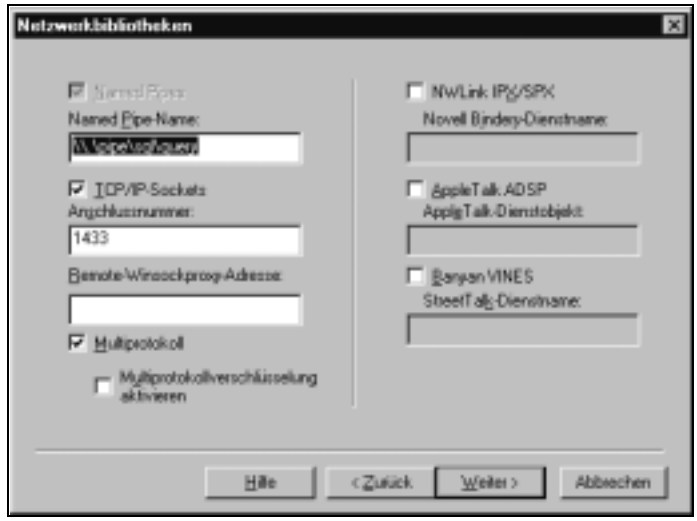


Bild 37.4: Dialogfeld »Netzwerkbibliotheken«

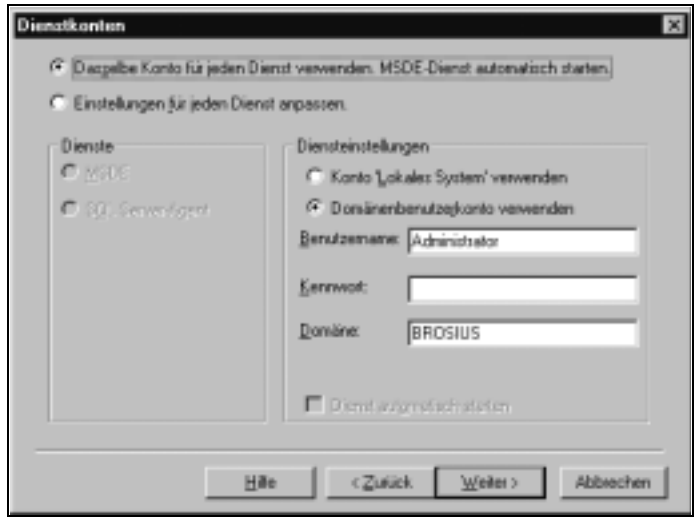


Bild 37.5: Dialogfeld »Dienstkonten«, das nur unter Windows NT erscheint.

- Wenn die Installation unter Windows NT erfolgt, wird das Dialogfeld *Dienstkonten* angezeigt, vgl. Bild 37.5. Am besten dürften Sie im allgemeinen fahren, wenn Sie die voreingestellten Optionen akzeptieren, wie sie in Bild 37.5 wiedergegeben sind. Achten Sie jedoch darauf, daß das Domänenbenutzerkonto Administratorenrechte besitzt, und geben Sie ggf. das Kennwort an. Bestätigen Sie mit der Schaltfläche *Weiter* >. Dann beginnt ein länger (ggf. mehrere Minuten) währender Vorgang, der die benötigten Dateien auf die Festplatte kopiert, Dienste einrichtet und Registrierungseinträge vornimmt. Nach erfolgreicher Installation erscheint ein Dialog-

feld mit einer entsprechenden Meldung, wobei Sie unter Windows 9x aufgefordert werden, das System neu zu booten, was unter Windows NT nicht erforderlich ist.

Das Installationsprogramm fügt dem System die neue Programmgruppe MSDE hinzu, mit der Sie die fünf Programme aufrufen können, die in Bild 37.6 wiedergegeben sind. Diese Programme bieten die nachfolgenden Funktionen:

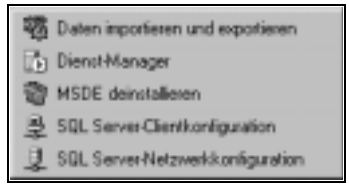


Bild 37.6: Programme der Programmgruppe MSDE

- *Daten importieren und exportieren*. Mit diesem Programm wird der DTS-Assistent aufgerufen. DTS steht für *Data Transformation Service*. Dieser Dienst erweist sich als äußerst nützlich, wenn es darum geht, heterogene Daten verschiedener Datenquellen und/oder Formate zu einer Datenquelle zusammenzuführen, wobei gleichzeitig eine auch Code-gesteuerte Datenbereinigung möglich ist. Es ist an dieser Stelle aus räumlichen Gründen nicht möglich, die umfangreichen Leistungen dieses Programms zu erläutern.

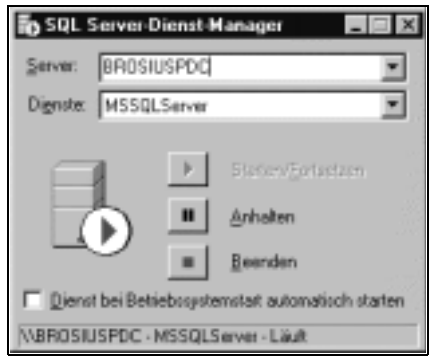


Bild 37.7: Dialogfeld »SQL Server-Dienst-Manager«.

- *Dienst-Manager.* MSDE ist als SQL Server unter dem Namen *MSSQLServer* installiert und angemeldet worden. Dieser läuft unter Windows NT als *Dienst*, daher der Name *Dienst-Manager*. Außer *MSSQLServer* wurden auch die weiteren Dienste *MSDTC* und *SQLServerAgent* installiert. Mit dem *Dienst-Manager* können Sie diese Dienste ggf. starten, pausieren lassen oder beenden, vgl. das Dialogfeld in Bild 37.7. Unter Windows NT

könnten Sie dieselben Wirkungen auch mit dem Programm *Dienste* aus der Programmgruppe *Systemsteuerung* erzielen. Per Voreinstellung wird der Dienst *MSSQLServer* – also der SQL Server selbst – automatisch mit dem Betriebssystem gestartet, die anderen beiden Dienste dagegen nicht. Wenn Sie allein und nur selten den SQL Server nutzen, sollten sie ihn auf manuellen Start einstellen, weil er natürlich Platz im Arbeitsspeicher verbraucht.

- *MSDE deinstallieren*. Mit diesem Programm deinstallieren Sie MSDE. Beachten Sie, daß dies nur möglich ist, wenn keiner der drei im vorangehenden Aufzählungspunkt angeführten Dienste läuft. yeachten Sie ferner, daß die Deinstallation zwar MSDE, nicht jedoch die Dateien der ggf. angelegten SQL-Datenbanken beseitigt. Dies sollten Sie ggf. mit dem Explorer selbst erledigen, weil es andernfalls bei erneuter Installation von MSDE möglicherweise Probleme geben könnte. Wenn Sie andererseits die Dateien der Datenbanken nach einer Deinstallation von MSDE nicht löschen, werden diese nach erneuter Installation von MSDE keinesfalls als Datenbanken reaktiviert, auch wenn die beteiligten Ordner dieselben sind wie vorher.
- *SQL-Server-Clientkonfiguration*. Mit diesem Programm können sie für den Client weitere Serveraliasnamen und Netzwerkbibliotheken hinzufügen und kynfigurieren.
- *SQL-Server-Netzwerkkonfiguration*. Mit diesem Programm können sie für den Server weitere Netzwerkbibliotheken hinzufügen und konfigurieren und damit ggf. die Einstellungen ändern, die bei der Installation von MSDE im Dialogfeld *Netzwerkbibliotheken* vorgenommen wurden, vgl. oben Bild 37.4.

37.3 Vergleich von Access-Datenbanken und Access-Projekten

Obwohl es dem Access-Entwicklerteam gelungen ist, die Übereinstimmungen zwischen Access-Datenbanken und Access-Projekten sehr weitgehend zu machen, so daß Benutzer, die Erfahrungen mit Access-Datenbanken haben, die dort erworbenen Kenntnisse in hohem Maße auf Access-Projekte übertragen können, ergeben sich doch einige Unterschiede aus der Sache, die nicht beseitigt werden können. Insbesondere gelten für SQL Server und zugehörige Sprachelemente z.T. andere Konventionen als für Access. Die Access-Hilfe listet die wichtigsten Unterschiede systematisch zu den folgenden Themen auf:

- Vergleich der Datenbankobjekte (z.B. *Abfragen* vs *Sichten* und *Gespeicherte Prozeduren*).
- Vergleich der Datentypen (z.B. *Datum/Uhrzeit* vs *datetime* und *smalldatetime*).

- Vergleich der Trennzeichen, Operatoren, Konstanten und Platzhalterzeichen (z.B. *Sternchen (*)* vs *Prozentzeichen (%)*).
- Vergleich der SQL-Syntax (z.B. *Beliebiges Zeichen am Anfang von Bezeichner erlaubt* vs *Bezeichner muss mit Buchstaben beginnen*).
- Vergleich der Funktionen von Visual Basic für Applikationen und Microsoft SQL Server Transact-SQL (z.B. *int(x)* vs *floor(x)*).

Dies sind Unterthemen zum größeren Hilfe-Thema *Vergleich von Access-Projekten mit Access-Datenbanken*, das wiederum Unterthema zum Thema *Arbeiten mit Microsoft Access-Projekten* ist. Detailliertes Arbeiten mit Access-Projekten erfordert häufigeres Nachschlagen unter diesen Themen.

37.4 Kennenlernen eines Access-Projekts: Beispielprojekt *NordwindCS.adp*

Entsprechend wie für Access-Datenbanken die Beispieldatenbank *Nordwind.mdb* bereitgestellt wird, bietet Access 2000 für Access-Projekte das Beispielprojekt *NordwindCS.adp* an (der Zusatz *CS* steht für Client/Server). Dabei handelt es sich inhaltlich in beiden Fällen um dieselbe Datenbank, mit den gleichen Formularen, Berichten etc. Im Datenteil – also bei Tabellen und Abfragen – zeigt sich das Projekt *NordwindCS.adp* aber mit anderen Objekten, nämlich Tabellen des SQL Servers, Sichten und Gespeicherten Prozeduren.

37.4.1 NordwindCS.adp erzeugen

Um das Access-Projekt *NordwindCS.adp* zu erzeugen, müssen zwei Voraussetzungen erfüllt sein:

- Beim Installieren von Office 2000 muß für Access 2000 unter dem Feature *Beispieldatenbanken* die *Nordwind SQL-Projektdatei* ausgewählt gewesen sein, so daß sich nach der Installation im Ordner *\Office\Samples* die Datei *NordwindCS.adp* mit einer Größe von etwas über 700 KB befindet. Sofern dies nicht zutrifft, können Sie das Beispielprojekt jederzeit nachinstallieren.
- Ein Microsoft SQL Server, z.B. MSDE, muß installiert sein.

Wenn diese Voraussetzungen vorliegen, erzeugen Sie das Access-Projekt *NordwindCS.adp* auf die folgende Weise:

- Öffnen Sie die Datei *NordwindCS.adp* im Ordner *\Office\Samples*, indem Sie z.B. darauf doppelklicken.

Bild 37.8: Dieses Dialogfeld erscheint nach dem erstmaligen Öffnen der Datei »NordwindCS.adp«.

Daraufhin wird Access gestartet und das Projekt geladen. Die datenbezogenen Objekte existieren anfangs noch nicht, sondern werden mittels VBA- und Transact-SQL-Code erzeugt. Dazu wird eine Verbindung zum SQL Server hergestellt, der, falls er nicht bereits gestartet war, automatisch mit VBA gestartet wird. Es erscheint dann das Dialogfeld *Datenbank installieren*, das in Bild 37.8 wiedergegeben ist. Nachdem Sie das Dialogfeld *Datenbank installieren* mit *Ja* bestätigt haben, wird nach einigen Sekunden *Datenbank auf SQL Server erstellt* gemeldet.

37.4.2 Gemeinsamkeiten und Unterschiede zwischen *Nordwind.mdb* und *NordwindCS.adp*

Das Datenbankfenster

Das Datenbankfenster für das Access-Projekt *NordwindCS.adp* (vgl. Bild 37.9) zeigt teilweise dieselben, teilweise aber auch andere Objekte an als für die Access-Datenbank *Nordwind.mdb*: Die obersten vier Objektlisten *Tabellen*, *Sichten*, *Datenbankdiagramme* und *Gespeicherte Prozeduren* unterscheiden sich deutlich vom Datenbankfenster einer Access-Datenbank.

Tabellen

Für die Objektliste *Tabellen* mag diese Aussage auf den ersten Blick nicht gerechtfertigt erscheinen, da sie denselben Namen hat wie bei einer Access-Datenbank. Jedoch werden die Tabellen im rechten Fensterteil mit anderen Symbolen dargestellt, als dies nach den Erfahrungen mit dem Datenbankfenster in Access-Datenbanken zu erwarten gewesen wäre: Die Tabellen sind ja nicht physikalischer Bestandteil der Access-Projektdatei, sondern sind physikalisch beim SQL Server gespeichert. Daher müßten sie eigentlich mit dem Symbol verknüpfter Tabellen erscheinen. Tatsächlich werden sie mit dem Symbol gewöhnlicher lokaler Tabellen dargestellt. Dies ist konsequent: Die Verbindung zum SQL Server erfolgt nach dem OLE DB- und nicht nach dem ODBC-Konzept, eine Verknüpfung im Sinne der Access-Datenbanken findet daher auch nicht statt. Es gibt nur einen Typ von Tabellen, so daß eine Unterscheidung von lokalen und verknüpften auch gar nicht notwendig ist.

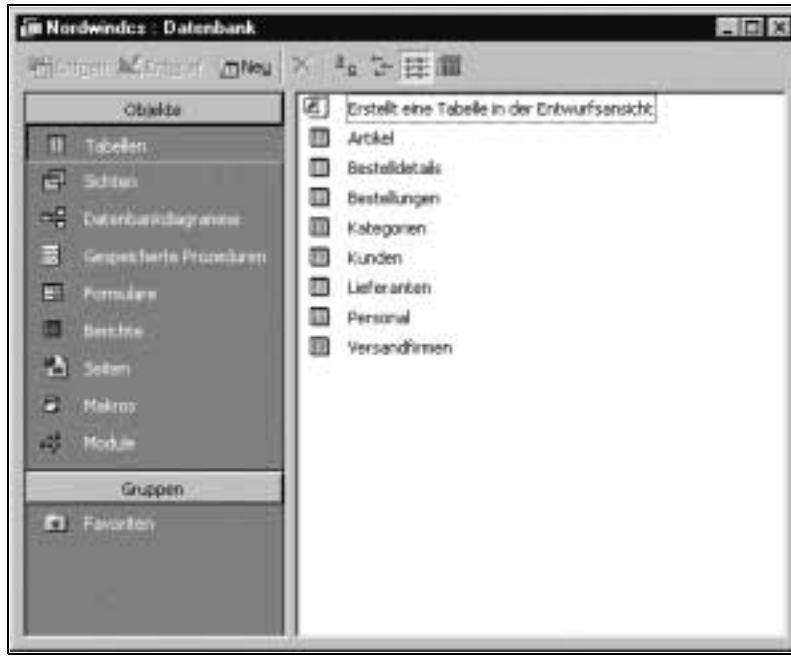


Bild 37.9: Datenbankfenster des Access-Projekts »NordwindCS.adp«

Wenn Sie eine Tabelle in der Datenblattansicht öffnen, so scheint auf den ersten Blick gar kein Unterschied zur Datenblattansicht einer Access-Datenbank-Tabelle zu bestehen. Wenn die Sache dagegen weiter geprüft wird, so zeigt sich, daß Access-Projekt-Tabellen weder Nachschlagefelder noch die in Access 2000 für Access-Datenbank-Tabellen neuen Unterdatenblätter darstellen können. Dafür läßt sich die Anzahl der dargestellten Datensätze mit Hilfe des Symbols *Max. Datensätze* beschränken, dem am weitesten rechts angeordneten Navigationssymbol, vgl. Bild 37.10, in dem das Dialogfeld *Max. Datensätze* vorübergehend durch Klicken auf das gleichnamige Symbol geöffnet wurde.

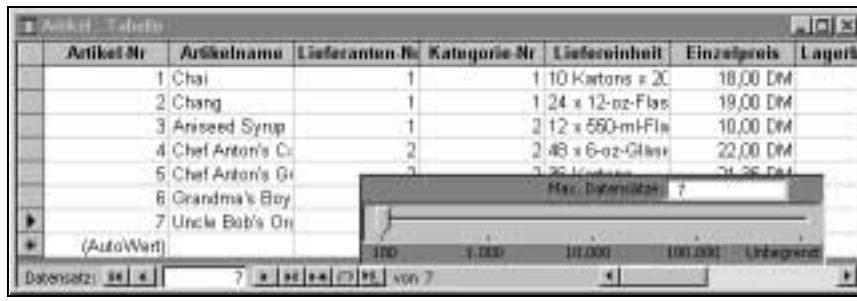


Bild 37.10: Die Zahl der maximal dargestellten Datensätze wurde auf 7 beschränkt.

Vollends anders zeigt sich eine Access-Projekt-Tabelle in der Entwurfsansicht, vgl. Bild 37.11, wo die Tabelle *Personal* in der Entwurfsansicht dargestellt ist. Die Feldeigenschaften und ihre möglichen Einstellungen haben hier nicht nur andere Namen, sondern sind auch anders angeordnet: Statt die Eigenschaften nur für das jeweils aktive Feld anzuzeigen, werden sie hier in tabellarischer Form gleichzeitig für alle Felder wiedergegeben, was die Übersichtlichkeit deutlich erhöht.

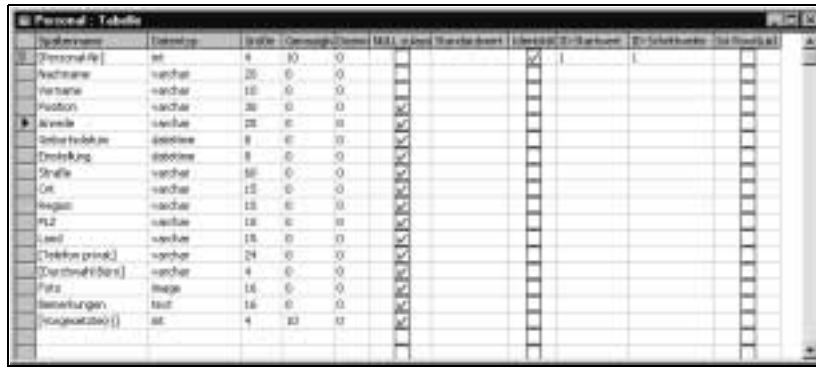


Bild 37.11: Tabelle »Personal« in der Entwurfsansicht

Sichten

Die Objektliste *Sichten* wird mit dem Symbol dargestellt, das in Access-Datenbanken für Auswahlabfragen verwendet wird. Dies ist gerechtfertigt, weil Sichten bis auf ein paar Einschränkungen den Auswahlabfragen der Access-Datenbanken entsprechen.

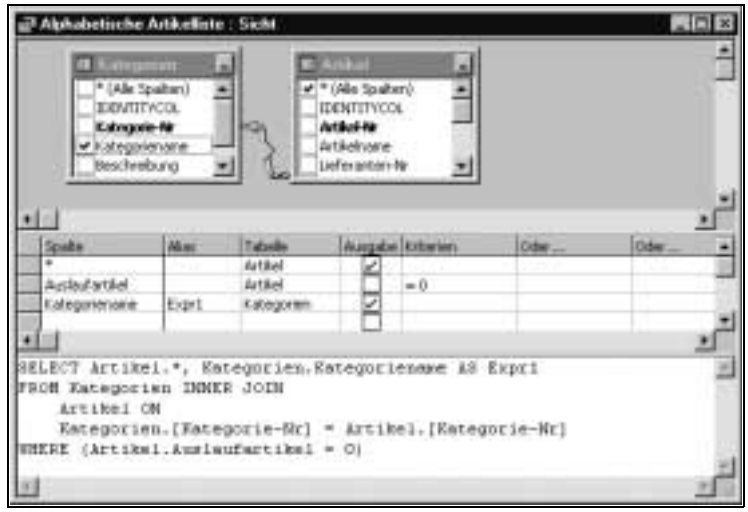


Bild 37.12: Sicht »Alphabetische Artikelliste« in der Entwurfsansicht. Alle drei Bereiche (Diagramm, Raster und SQL) werden angezeigt.

Für die Datenblattansicht einer Sicht gilt Entsprechendes wie für Tabellen: Es können weder Nachschlagefelder noch Unterdatenblätter dargestellt werden. Dafür läßt sich die Zahl der maximal angezeigten Datensätze beschränken.

In der Entwurfsansicht unterscheiden sich eine Auswahlabfrage und eine Sicht deutlich, vgl. Bild 37.12, in dem die Entwurfsansicht der Sicht *Alphabetische Artikelliste* wiedergegeben ist. Die Entwurfsansicht einer Sicht besitzt die drei Bereiche *Diagramm* (oben), *Raster* (Mitte) und *SQL* (unten), die in Bild 37.12 alle eingeblendet sind. Jeder dieser Bereiche kann mittels der entsprechenden Symbole in der Symbolleiste ein- und ausgeblendet werden. Die gleichzeitige Darstellung von *Raster* und *SQL* ist eine Verbesserung gegenüber der Entwurfsansicht von Auswahlabfragen in Access-Datenbanken, wo diese beiden Darstellungen zeitlich alternativ sind, lassen sich auf diese Weise doch die Konsequenzen von Änderungen im Rasterbereich unmittelbar im SQL-Bereich erkennen und umgekehrt. Auch die Tabellendiagramme sind im Vergleich zu den entsprechenden Darstellungen bei Auswahlabfragen leistungsfähiger: Der Zustand des Kontrollkästchens am linken Rand eines Feldes wirkt sich unmittelbar auf die SQL-Formulierung und das Raster der Sicht aus, was den Komfort beim Entwurf deutlich erhöht.

Die bedeutsamste Einschränkung einer Sicht gegenüber einer Auswahlabfrage liegt in dem Umstand, daß keine gezielte Sortierung nach einzelnen Feldern angegeben werden kann. Dies muß mit Gespeicherten Prozeduren bewerkstelligt werden (vgl. dort), was deutlich umständlicher als die direkte Angabe einer Sortieranweisung in der Abfrage ist.

Gespeicherte Prozeduren

Das Projekt *NordwindCS.adp* enthält insgesamt drei Gespeicherte Prozeduren. Jede von ihnen erzeugt eine Select-Abfrage, deren Ergebnis in der Datenblattansicht betrachtet werden kann: Doppelklicken Sie beispielsweise auf *Die zehn teuersten Artikel*, um die Datenblattansicht anzuzeigen. Diese unterscheidet sich in keiner Weise von der Datenblattansicht einer Sicht oder Tabelle.

Bild 37.13: Entwurfsansicht der Gespeicherten Prozedur »Die zehn teuersten Artikel« nach dem Einfügen von Zeilenumbrüchen.

Die Entwurfsansicht einer Gespeicherten Prozedur gibt den Text der SQL-Anweisungen wieder, die sie enthält. Bild 37.13 zeigt den SQL-Code der Gespeicherten Prozedur *Die zehn teuersten Artikel*, allerdings in einer etwas bearbeiteten Form: Unmittelbar nach dem Erzeugen einer Gespeicherten Prozedur (vgl. dazu Kap. 38, Access-Projekt erstellen und bearbeiten, Punkt 38.5, *Gespeicherte Prozeduren*) existiert der SQL-Code ohne jeden Zeilenumbruch. Dies gilt auch für die Gespeicherte Prozedur *Die zehn teuersten Artikel*. Um den Code besser lesen zu können, sollten sie an geeigneten Stellen Zeilenumbrüche (Drücken der Eingabetaste) erzeugen, die dann auch gespeichert werden können. Dies ist in Bild 37.13 geschehen.

Die Gespeicherte Prozedur *Die zehn teuersten Artikel* war erforderlich, um im Access-Projekt *NordwindCS.adp* das Äquivalent zur gleichnamigen Auswahlabfrage der Nordwind-Datenbank zu erzeugen. Letztere nämlich sortiert die Datensätze in absteigender Folge nach dem Feld *Einzelpreis*. Eine Sortierung (das Anführen der *Order By*-Klausel) ist in einer Sicht dagegen nicht möglich, während die *Order By*-Klausel in einer Gespeicherten Prozedur angeführt werden darf. Die anderen beiden Gespeicherten Prozeduren *Personalumsätze nach Land* und *Umsätze nach Jahr* waren in der Datenbank *Nordwind.mdb* Parameterabfragen. In einer Sicht lassen sich dagegen keine Parameter spezifizieren, was wiederum in Gespeicherten Prozeduren möglich ist.

Datenbankdiagramme

Datenbankdiagramme sind im Prinzip mit dem Fenster *Beziehungen* in einer Access-Datenbank vergleichbar. So stellen sie u.a. die Beziehungen zwischen den Tabellen in der SQL-Datenbank dar, die auch mit dem Datenbankdiagrammdesigner erstellt und bearbeitet werden können. In Bild 37.14 ist das in dem Access-Projekt *NordwindCS.adp* definierte Datenbankdiagramm mit dem Namen *Beziehungen* wiedergegeben. Dies ist das einzige dort vorhandene Datenbankdiagramm, es könnten jedoch prinzipiell mehr als eines vorhanden sein.

Die Bearbeitungsmöglichkeiten eines Datenbankdiagramms gehen jedoch weit über diejenigen im Fenster *Beziehungen* hinaus: Zunächst unterscheidet sich ein Datenbankdiagramm positiv vom Fenster *Beziehungen*, weil es so auf dem Drucker ausgegeben werden kann, wie es auf dem Bildschirm wiedergegeben wird.[2] Darüber hinaus können Sie in einem Datenbankdiagramm auch neue Tabellen erstellen, Felder und deren Eigenschaften definieren oder Indizes erstellen. Die Access 2000-Hilfe unterstellt, daß die Pflege der Tabellendefinitionen mit dem Datenbankdiagrammdesigner – also in

2. Mit Access 2000 sind allerdings auch die Beziehungen des Fensters *Beziehungen* einer Access-Datenbank ausdruckbar, jedoch in einer etwas eingeschränkten Form, nämlich als Bericht.

einem Datenbankdiagramm – erfolgt, obgleich Tabellendefinitionen auch in der Entwurfsansicht von Tabellen bearbeitet werden können. Der Datenbankdiagrammdesigner wird als das allgemeine Tool betrachtet. Daher reichen Datenbankdiagramme auch konzeptionell weit über das Fenster *Beziehungen* in Access-Datenbanken hinaus. Nicht zuletzt lassen sich damit Datenbanken auch experimentell grafisch entwerfen. Die vorgenommenen Änderungen werden erst im angebundenen SQL Server gespeichert, wenn auch das entsprechende Datenbankdiagramm gespeichert wird.

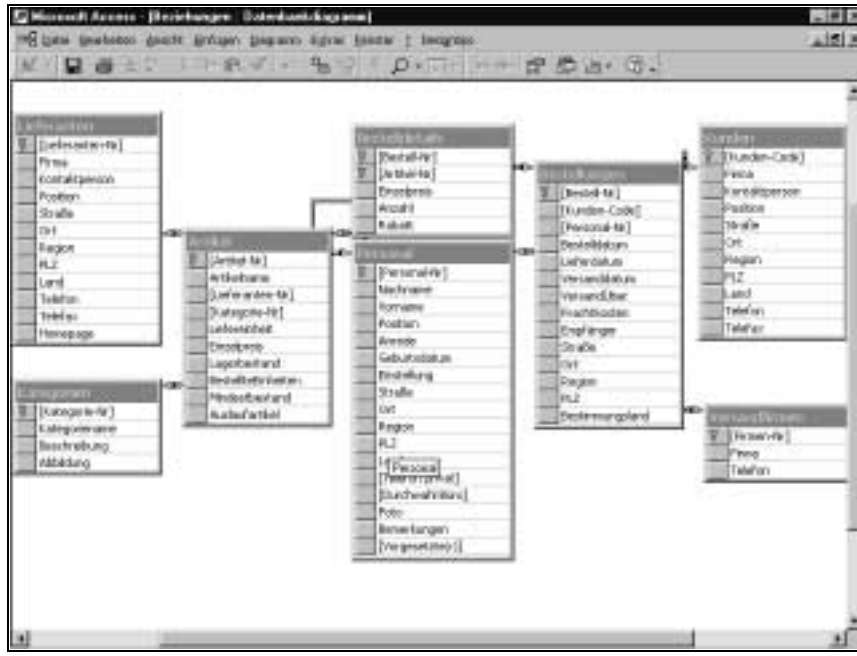


Bild 37.14: Datenbankdiagramm »Beziehungen« aus dem Access-Projekt »NordwindCS.adp«

Weitere Objekte

Die anderen im Datenbankfenster angezeigten Objekte wie Formulare, Berichte oder Module weisen nur geringe Änderungen gegenüber ihren Entsprechungen in einer Access-Datenbank auf. Wo solche Unterschiede relevant sind, wird im nächsten Kapitel im Punkt 38.6, *Datenquellen und Abfrageparameter in Formularen und Berichten*, detaillierter darauf eingegangen.

Kapitel 38

Access-Projekt erstellen und bearbeiten

Es gibt mehrere Möglichkeiten, ein Access-Projekt zu erstellen:

- *Upsizing-Assistent*. Sie können eine vorhandene Access-Datenbank mit Hilfe des *Upsizing-Assistenten* in ein Access-Projekt konvertieren. Dabei bleibt die vorhandene Datenbank erhalten. Dieser Weg wird im folgenden Punkt 38.1, *Access-Datenbank mit dem Upsizing-Assistenten in Access-Projekt umwandeln*, beschrieben. Wie sich zeigen wird, führt die Arbeit des Upsizing-Assistenten allerdings nicht immer zu zufriedenstellenden Ergebnissen. Schon aus diesem Grunde sollten Sie auch die im folgenden Aufzählungspunkt beschriebene Möglichkeit kennen.
- *Gänzlich neues Projekt*. Mit dem Befehl *Neu* aus dem Menü *Datei* können Sie, analog zum Erstellen einer neuen Access Datenbank, ein neues Access-Projekt erstellen. Dabei können Sie zwischen den Möglichkeiten wählen, gleichzeitig eine neue Datenbank auf dem SQL Server anzulegen, oder das neue Projekt mit einer vorhandenen Datenbank (und deren Tabellen, Sichten und Gespeicherten Prozeduren) zu verbinden. Den letzten Schritt können Sie auch abbrechen, um das Projekt ggf. später mit einer Server-Datenbank zu verbinden. Diese Möglichkeiten werden im Punkt 38.2, *Access-Projekt neu erstellen*, beschrieben.

38.1 Access-Datenbank mit dem Upsizing-Assistenten in Access-Projekt umwandeln

Die Verwendung des Upsizing-Assistenten soll am Beispiel der Access-Datenbank *Nordwind.mdb*, aus der ein Access-Projekt erzeugt werden soll, demonstriert werden. Neben anderen Gründen spricht für die Verwendung gerade dieses Beispiels, daß das Ergebnis der Arbeit des Upsizing-Assistenten dann mit dem Beispielprojekt *NordwindCS.adp* aus dem Ordner

\Office\Samples (vgl. oben Kap.37, *Access-Projekte als Client/Server-Datenbank: Grundlagen*, Punkt 37.4, *Kennenlernen eines Access-Projekts: Beispielprojekt NordwindCS.adp*) verglichen werden kann, denn dieses Projekt wurde nicht mit dem Upsizing-Assistenten, sondern direkt mit Transfer-SQL-Skript erzeugt.

Gehen Sie folgendermaßen vor, um mit dem Upsizing-Assistenten aus der Access-Datenbank *Nordwind.mdb* das entsprechende Access-Projekt zu erzeugen:

- Starten Sie den SQL Server, falls dieser nicht bereits läuft: Öffnen Sie zunächst in der Programmgruppe *MSDE* den *Dienst-Manager*, und starten Sie in dessen Fenster den SQL Server.
- Öffnen Sie die Datenbank *Nordwind.mdb*.
- Wählen Sie aus dem Menü *Extras* den Befehl *Datenbank-Dienstprogramme* und dazu den Unterbefehl *Upsizing-Assistent*.

Bild 38.1: Erstes Dialogfeld des Upsizing-Assistenten

- Belassen Sie es im ersten Dialogfeld des Upsizing-Assistenten (vgl. Bild 38.1) bei der voreingestellten Option *Neue Datenbank erstellen*, bei der aus den Tabellen und Abfragen der Ausgangsdatenbank im SQL Server eine neue Datenbank erstellt und mit Tabellen, Sichten und Gespeicherten Prozeduren gefüllt wird. Statt dessen hätten sie auch die vorhandene Datenbank *NordwindCS*, die beim Erzeugen der Beispieldatenbank *NordwindCS.adp* gebildet wurde, verwenden können. Der erstere Weg ist jedoch der allgemeinere und ermöglicht darüber hinaus einen besseren Vergleich der Ergebnisse des Upsizing-Assistenten mit dem Beispielprojekt *NordwindCS.adp*. Klicken Sie auf die Schaltfläche *Weiter >*.

Bild 38.2: Hier werden Servername und Name der SQL-Datenbank sowie Anmeldeinformationen angegeben.

- Wählen Sie im zweiten Dialogfeld des Upsizing-Assistenten (vgl. Bild 38.2) ggf. den zu verwendenden SQL Server, und geben Sie einen Namen für die darin zu bildende Datenbank an. Das Dialogfeld in Bild 38.2 zeigt die vorbesetzten Werte für diese beiden Felder an. Geben Sie für die *Anmeldungs-ID* im Falle von MSDE als SQL Server *sa* (steht für *system administrator*) an, weil dies dabei das voreingestellte Konto ist. Wenn Sie einen anderen SQL Server verwenden, müßten Sie ggf. ein anderes Konto mit Kennwort eingeben. Klicken Sie auf die Schaltfläche *Weiter* >.

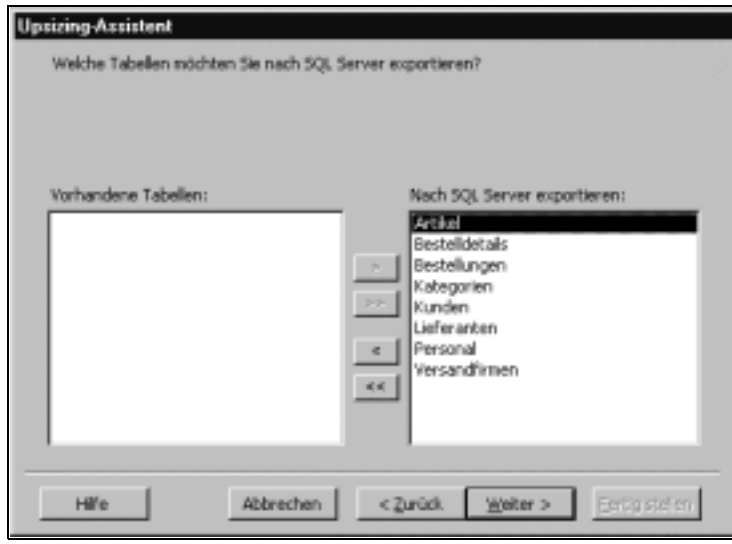


Bild 38.3: Übernahme der Tabellen aus der Quelldatenbank in die SQL-Datenbank

- Übernehmen Sie im folgenden Dialogfeld des Upsizing-Assistenten (vgl. Bild 38.3) sämtliche Tabellen der Ausgangsdatenbank in den SQL-Server, indem Sie auf die Schaltfläche >> klicken. Klicken Sie auf die Schaltfläche *Weiter* >.

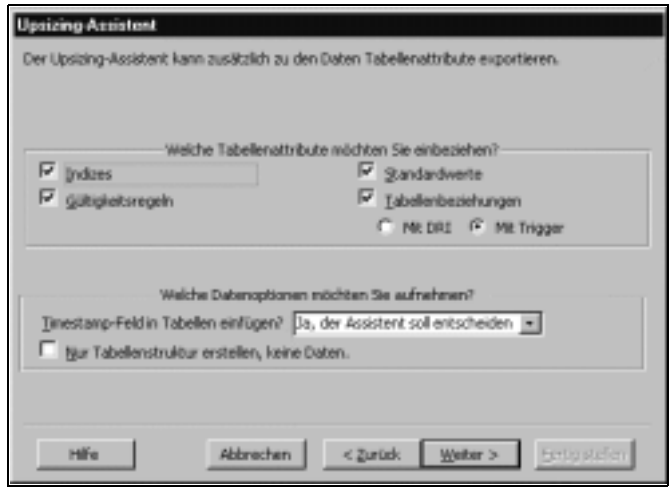


Bild 38.4: Dieses Dialogfeld gibt die voreingestellten Werte für die Tabellenattribute und Datenoptionen wieder.

- Belassen Sie es im nächsten Dialogfeld (vgl. Bild 38.4) bei den voreingestellten Werten für die Tabellenattribute und Datenoptionen, um ein möglichst umfassendes Upsizing zu erzielen. Klicken Sie auf die Schaltfläche *Weiter* >.

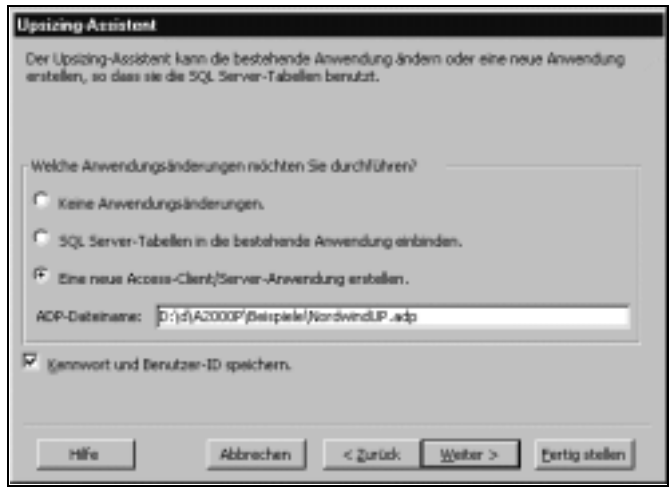


Bild 38.5: Lassen Sie eine neue Anwendung unter einem spezifizierten Projektnamen erstellen.

- Um den Upsizing-Assistenten möglichst umfassend zu testen, belassen sie es im folgenden Dialogfeld (vgl. Bild 38.5) bei der voreingestellten Option *Eine neue Access-Client/Server-Anwendung erstellen*, und geben Sie für das Access-Projekt Pfad und Dateinamen an. Im dargestellten Beispiel wurde statt des vom Upsizing-Assistenten vorgeschlagenen Dateinamens *NordwindCS.adp* der Dateiname *NordwindUP.adp* gewählt, um einer Verwechslung mit dem gleichnamigen Access 2000 Beispielprojekt vorzubeugen. Klicken Sie auf die Schaltfläche *Weiter >*.

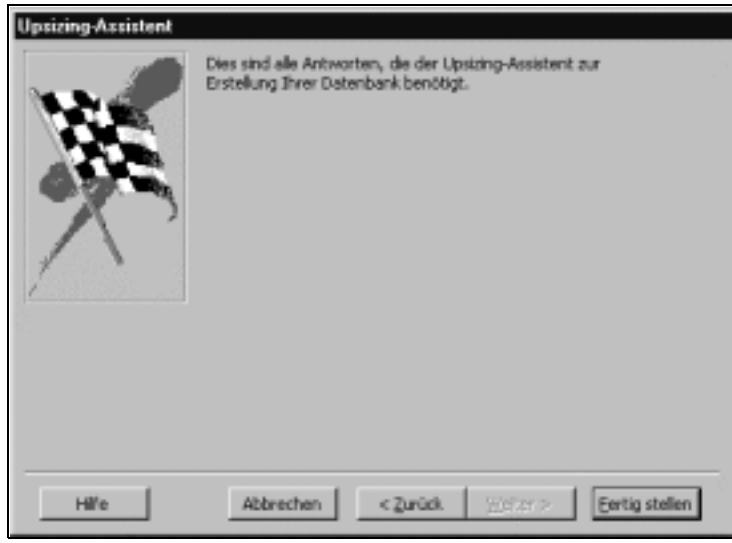


Bild 38.6: Alle erforderlichen Informationen liegen vor.

- Wenn bis hier alles gut ging, sollte der Upsizing-Assistent in seinem nächsten Dialogfeld (vgl. Bild 38.6) mitteilen, daß er alle erforderlichen Informationen erhalten hat. Klicken Sie dann auf die Schaltfläche *Fertig stellen*.

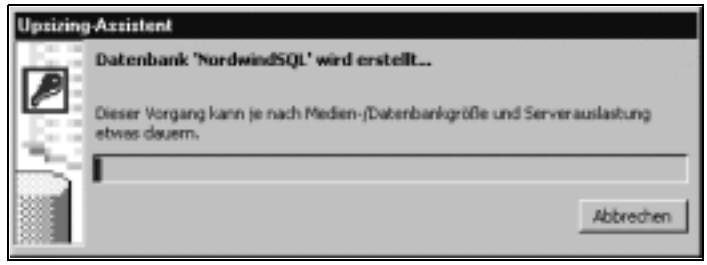


Bild 38.7: In diesem Dialogfeld werden die verschiedenen Etappen bei der Bildung des neuen Access-Projekts angezeigt.

- Im folgenden Dialogfeld (vgl. Bild 38.7) werden die verschiedenen Etappen bei der Bildung des neuen Access-Projekts angezeigt. Dieser Vorgang kann je nach Größe der Ausgangsdatenbank zwischen einigen Sekunden und vielen Minuten dauern. Im Falle von *Nordwind.mdb* dauert er wenige Minuten.

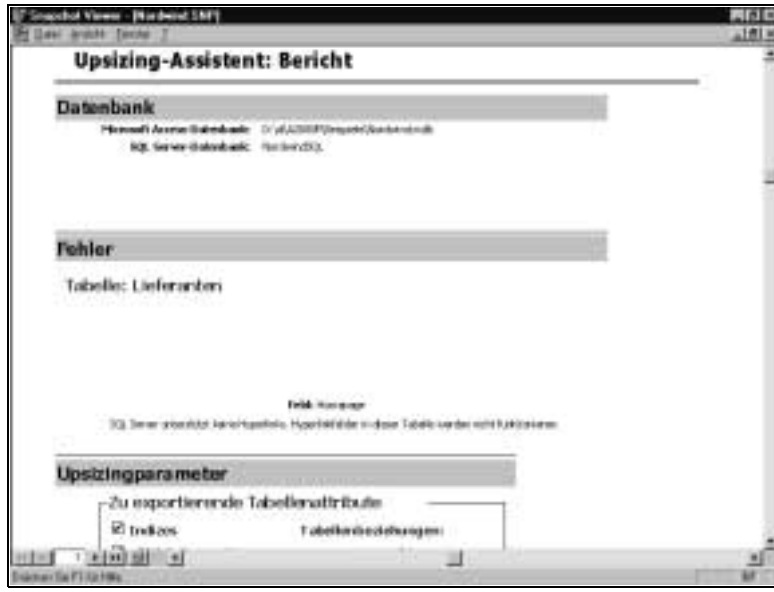


Bild 38.8: Bericht des Upsizing-Assistenten

Nach getaner Arbeit erstellt der Upsizing-Assistent einen Bericht über den Vorgang, der den Charakter eines Protokolls hat, das beispielsweise aufgetretene Fehler, Schwierigkeiten oder Unmöglichkeiten bei der Konversion und Ähnliches auflistet, vgl. Bild 38.8. Diesen Bericht, der als Snapshot erstellt und im Falle der Nordwind-Datenbank unter dem Namen *Nordwind.SNP* gespeichert wird, sollten Sie im Detail zur Kenntnis nehmen, denn er liefert wertvolle Anhaltspunkte für die Änderungen, die nach der Arbeit des Upsizing-Assistenten noch von Ihnen zu erledigen sind, wenn das neue Access-Projekt der alten Access-Datenbank gleichwertig sein soll.

Abfragename: Umsätze nach Kategorie für 1997

Upsizing konnte nicht durchgeführt werden. Es wurde versucht mit SQL:

```
CREATE VIEW "Umsätze nach Kategorie für 1997"
AS
SELECT "Umsätze nach Artikeln für 1997".Kategoriename, sum("Umsätze nach Artikeln für 1997".) AS
"Kategorieumsätze"
FROM "Umsätze nach Artikeln für 1997"
GROUP BY "Umsätze nach Artikeln für 1997".Kategoriename
```

Bild 38.9: Ausschnitt aus dem Bericht des Upsizing-Assistenten

Beispielsweise enthält der Upsizing-Bericht über die Umwandlung der Abfrage *Umsätze nach Kategorie für 1997* den in Bild 38.9 wiedergegebenen Text. Daraus können Sie nicht nur erkennen, daß die Umwandlung dieser Abfrage in eine Sicht mißlungen ist, sondern Sie können den mitgeteilten SQL-Text auch als Ausgangspunkt für Ihre eigenen Anpassungsversuche heranziehen.

Wenn man das Ergebnis des Upsizing von *Nordwind.mdb* mit dem von Access 2000 mitgelieferten Access-Projekt *NordwindCS.adp* vergleicht, kann man verstehen, warum dieses letztere Projekt durch ein dezidiert für dieses Projekt geschriebenes Transfer-SQL-Skript erstellt wurde statt durch den Upsizing-Assistenten: Die Lösungen in dem originalen Beispielprojekt sind nicht nur eleganter weil einfacher als im Ergebnis des Upsizing-Assistenten, sondern sie sind im Vergleich auch vollständig: Der Upsizing-Assistent konnte zwar die meisten Objekte umwandeln, mehrere Abfragen jedoch nicht. Dadurch können auch einzelne Formulare und Berichte nicht geöffnet werden, weil ihnen die Datenbasis fehlt.

Zusammenfassend kann die Nützlichkeit des Upsizing-Assistenten daher folgendermaßen beurteilt werden: Er leistet eine nützliche Arbeit, weil er insbesondere die Tabellen der Ausgangsdatenbank in eine SQL-Server-Datenbank transformiert. Darüber hinaus werden die meisten Objekte der Ausgangsdatenbank zufriedenstellend umgewandelt. Insbesondere Abfragen und einzelne Tabellenfelder können jedoch nicht hinreichend angepaßt werden. Dies liegt nicht zuletzt an den folgenden systematischen Beschränkungen des Upsizing-Assistenten:

- Union-, Kreuztabellen- und Tabellenerstellungsabfragen werden nicht umgewandelt. Sie müßten sich die benötigten Union-Abfragen ggf. selbst mit Transact-SQL erstellen.
- Nicht alle Gültigkeitsregeln können umgewandelt werden.
- VBA-Ausdrücke in Access-SQL wie beispielsweise *Iif* werden nicht umgewandelt. Sie müßten dies ggf. durch selbst erstellte Sichten oder Gespeicherte Prozeduren nachholen.

38.2 Access-Projekt neu erstellen

Ein neues Access-Projekt erstellen Sie mit dem Befehl *Neu* aus dem Menü *Datei*. Dabei lassen sich die folgenden vier Arten, ein Projekt zu erstellen, unterscheiden:

- Access-Projekt und SQL-Server-Datenbank erstellen und miteinander verbinden.
- Access-Projekt erstellen und mit einer vorhandenen SQL-Server-Datenbank verbinden.

- Access-Projekt erstellen, ohne es mit einer SQL-Server-Datenbank zu verbinden.
- Vorhandenes Access-Projekt mit einer (anderen) vorhandenen Microsoft SQL-Server-Datenbank verbinden.

Diese vier Arbeitsweisen sollen in den folgenden Punkten jeweils kurz beschrieben und erklärt werden.

38.2.1 Access-Projekt und SQL-Server-Datenbank erstellen und miteinander verbinden

Gehen Sie folgendermaßen vor, um ein neues Access-Projekt und gleichzeitig eine neue SQL-Server-Datenbank zu erstellen:

- Starten Sie den SQL Server, falls dieser nicht bereits läuft, beispielsweise mit dem *Dienst-Manager* aus der Programmgruppe *MSDE*.
- Wählen Sie aus dem Menü *Datei* den Befehl *Neu*.
- Doppelklicken Sie im Dialogfeld *Neu* in der Registerkarte *Allgemein* auf das Symbol *Projekt (Neue Datenbank)*.
- Bestimmen Sie im Dialogfeld *Neue Datenbankdatei* (es müßte besser lauten: Neue Projektdatei!) den Speicherordner, geben Sie einen Namen für die zu bildende *.adp*-Datei an, und bestätigen Sie mit der Schaltfläche *Erstellen*.

Bild 38.10: Geben Sie Angaben zur neuen SQL-Server-Datenbank an

- Geben Sie im Dialogfeld *Microsoft SQLServer Datenbank-Assistent* (vgl. Bild 38.10) Angaben zur neuen SQ-Server-Datenbank an: Wählen Sie den geeigneten SQL Server (falls mehr als einer verfügbar ist), vergeben Sie einen Namen für die neue SQL-Server-Datenbank, und geben Sie ggf. eine

Anmeldungs-ID und ein Kennwort für den SQL Server an. Im Falle des lokalen MSDE SQL Servers können Sie, wenn Sie an den Voreinstellungen nichts geändert haben, als Anmeldungs-ID *sa* ohne Kennwort angeben oder auch die Anmeldungs-ID leer lassen. Klicken Sie auf die Schaltfläche *Weiter >*.

- Klicken Sie im folgenden Dialogfeld des SQL Server Datenbank-Assistenten, in dem das Vorhandensein aller benötigten Informationen bestätigt wird, auf die Schaltfläche *Fertig stellen*. Dann wird das neue Access-Projekt erstellt und anschließend geöffnet.

Das neue Access-Projekt enthält natürlich keinerlei Objekte, weder solche, die mit dem Projekt gespeichert werden – beispielsweise Formulare oder Berichte – noch solche, die auf dem SQL Server gehalten werden – Tabellen, Sichten und Gespeicherte Prozeduren. Die weitere Arbeit mit dem neu erstellten Access-Projekt sollte deshalb darin bestehen, der SQL-Server-Datenbank Tabellen sowie ggf. Sichten und Gespeicherte Prozeduren und dem Projekt weitere Objekte wie Formulare, Berichte oder Makros hinzuzufügen. Das Erstellen und Bearbeiten von Tabellen, Sichten und Gespeicherten Prozeduren wird weiter unten in diesem Kapitel gezeigt.

Obwohl die zusammen mit dem Access-Projekt neu erstellte SQL-Server-Datenbank bisher keinerlei Objekte enthält, existiert sie dennoch auf dem SQL Server. Darüber hinaus müßte das neue Access-Projekt mit dieser Datenbank verbunden worden sein. Um dies zu überprüfen, gehen Sie folgendermaßen vor:

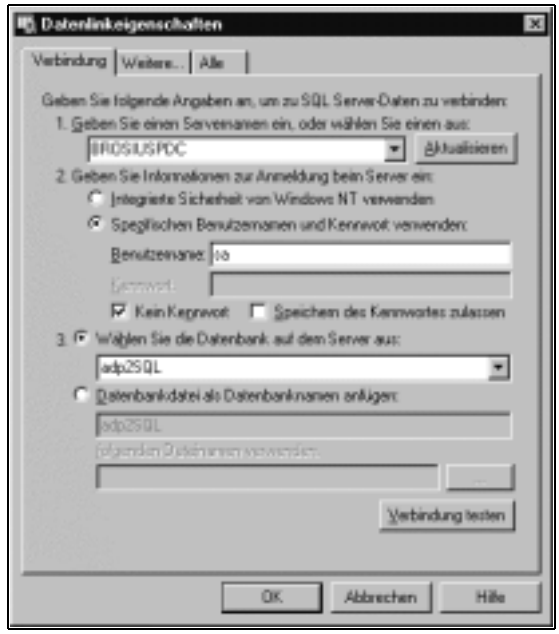


Bild 38.11: Dialogfeld »Datenlinkeigenschaften«

- Wählen Sie, während das neu erstellte Projekt geöffnet ist, im Menü *Datei* den Befehl *Verbindung*. Dann wird das Dialogfeld *Datenlinkeigenschaften* angezeigt, vgl. Bild 38.11. Auf der Registerkarte *Verbindung* werden die Verbindungseigenschaften des geöffneten Projekts zum SQL Server angezeigt (und könnten hier auch geändert werden). In Bild 38.11 besteht eine Verbindung zur SQL-Server-Datenbank *adp2SQL*. Dieser Name wurde oben beim Erstellen der Datenbank angegeben, vgl. das entsprechende Dialogfeld in Bild 38.10.
- Durch Klicken auf die Schaltfläche *Verbindung testen* können Sie – auch für eine leere SQL-Server-Datenbank – prüfen, ob eine Verbindung vom Access-Projekt zur SQL-Server-Datenbank hergestellt werden kann. Dies wird ggf. mit der in Bild 38.12 wiedergegebenen Meldung bestätigt oder mit einer entsprechenden Meldung verneint.

Bild 38.12: Bestätigung des positiven Verbindungstests

38.2.2 Access-Projekt erstellen und mit einer vorhandenen SQL-Server-Datenbank verbinden.

Gehen Sie folgendermaßen vor, um ein neues Access-Projekt zu erstellen, das mit einer vorhandenen SQL-Server-Datenbank verbunden werden soll:

- Starten Sie den SQL Server, falls dieser nicht bereits läuft, beispielsweise mit dem *Dienst-Manager* aus der Programmgruppe *MSDE*.
- Wählen Sie aus dem Menü *Datei* den Befehl *Neu*.
- Doppelklicken Sie im Dialogfeld *Neu* in der Registerkarte *Allgemein* auf das Symbol *Projekt (Bestehende Datenbank)*.
- Bestimmen Sie im Dialogfeld *Neue Datenbankdatei* den Speicherordner, geben Sie einen Namen für die zu bildende *.adp*-Datei an, und bestätigen Sie mit der Schaltfläche *Erstellen*.
- Wählen Sie im Dialogfeld *Datenlinkeigenschaften* (vgl. Bild 38.13) den Servernamen aus. Entscheiden Sie sich für eine der beiden Anmeldeoptionen: Wenn Sie *Integrierte Sicherheit von Windows NT verwenden* aktivieren, können Sie keine serverspezifischen Anmeldeangaben vornehmen. Wenn Sie *Spezifischen Benutzernamen und Kennwort verwenden* aktivieren, geben Sie im Falle des MSDE SQL Servers als Benutzernamen *sa* ohne Kennwort an, sofern Sie nicht die Voreinstellungen in dieser Hinsicht

geändert haben. Schlagen Sie schließlich die Dropdown-Liste mit den Server-Datenbanken auf, wählen Sie eine geeignete, und bestätigen Sie das Dialogfeld mit OK.

Bild 38.13: Dialogfeld »Datenlinkeigenschaften«, hier mit aufgeschlagener Dropdown-Liste zum Auswählen einer anzubindenden SQL-Server-Datenbank.

Das neu erstellte Access-Projekt ist an die im letzten Schritt ausgewählte SQL-Server-Datenbank angebunden. Dies zeigt sich darin, daß im Datenbankfenster sämtliche Objekte der Datenbank angezeigt werden, womit sie in dem Projekt natürlich auch verfügbar sind. Im Datenbankfenster des Access-Projekts *adp3.adp*, das mit der Access 2000 Beispiel-SQL-Server-Datenbank *NordwindCS* verbunden ist, werden daher die Tabellen der Nordwind-Datenbank angezeigt, vgl. Bild 38.14. Entsprechend werden in diesem Datenbankfenster auch die Sichten, Gespeicherten Prozeduren und das Datenbankdiagramm des Nordwind-Projekts angezeigt.

☞ Möglicherweise wollen Sie in dem Access-Projekt, das Sie mit einer bestimmten SQL-Server-Datenbank verbunden haben, nur einen Teil von deren Objekten nutzen. Auch in diesem Falle können Sie jedoch nicht verhindern, daß alle Objekte der SQL-Server-Datenbank, also auch die nicht benötigten, angezeigt werden. Anders als bei verknüpften Tabellen einer Access-Datenbank hätte das Löschen einer Tabelle der SQL-Server-Daten-

bank im Access-Projekt zur Folge, daß die Tabelle auch in der SQL-Server-Datenbank physikalisch gelöscht würde (bei verknüpften Tabellen wird dagegen nur die Verbindung aufgehoben). Andererseits dürfte es aber auch kaum stören, nicht benötigte Objekte der SQL-Server-Datenbank im Access-Projekt anzeigen zu lassen, denn diese verbrauchen keinen Speicherplatz.

Bild 38.14: Datenbankfenster des an die Beispiel-SQL-Server-Datenbank »NordwindCS« angebundenen Access-Projekts"adp3«, hier mit der Objektliste »Tabellen«.

38.2.3 Access-Projekt erstellen, ohne es mit einer SQL-Server-Datenbank zu verbinden.

Es kann sinnvoll sein, zunächst ein Access-Projekt zu erstellen und dieses erst später mit einer geeigneten Datenbank, die zum Zeitpunkt des Erstellens des Access-Projekts möglicherweise noch gar nicht zur Verfügung steht, zu verbinden. Allerdings können Sie solange keine datenbezogenen Objekte (Tabellen etc.) im Access-Projekt erstellen, wie dieses nicht mit einer SQL-Server-Datenbank verbunden ist.

Gehen Sie folgendermaßen vor, um ein Access-Projekt zu erstellen, ohne dieses mit einer SQL-Server-Datenbank zu verbinden:

- Wählen Sie aus dem Menü *Datei* den Befehl *Neu*.
- Doppelklicken Sie im Dialogfeld *Neu* in der Registerkarte *Allgemein* auf das Symbol *Projekt (Bestehende Datenbank)*.

- Bestimmen Sie im Dialogfeld *Neue Datenbankdatei* den Speicherordner, geben Sie einen Namen für die zu bildende *.adp*-Datei an, und bestätigen Sie mit der Schaltfläche *Erstellen*.
- Klicken Sie im Dialogfeld *Datenlinkeigenschaften* (vgl. z.B. oben Bild 38.13) auf die Schaltfläche *Abbrechen*.

Sie können das dann gebildete unverbundene Access-Projekt jederzeit später mit einer bestehenden SQL-Server-Datenbank verbinden, vgl. dazu den folgenden Punkt 38.2.4, *Vorhandenes Access-Projekt mit einer (anderen) vorhandenen Microsoft SQL-Server-Datenbank verbinden*.

38.2.4 Vorhandenes Access-Projekt mit einer (anderen) vorhandenen Microsoft SQL-Server-Datenbank verbinden

Sie können ein bestehendes Access-Projekt jederzeit mit einer neuen oder, falls bereits eine Verbindung zu einer Datenbank besteht, anderen SQL-Server-Datenbank verbinden. Gehen Sie dazu folgendermaßen vor:

- Starten Sie den SQL Server, falls dieser nicht bereits läuft, beispielsweise mit dem *Dienst-Manager* aus der Programmgruppe *MSDE*.
- Öffnen Sie das betreffende Projekt.
- Wählen Sie im Menü *Datei* den Befehl *Verbindung*.
- Wählen Sie im Dialogfeld *Datenlinkeigenschaften* (vgl. z.B. oben Bild 38.13) die gewünschte Datenbank, nachdem Sie im Dialogfeld die geeigneten Angaben zum Server und zur Anmeldung angegeben haben. Testen Sie ggf. die neue Verbindung, und bestätigen Sie mit OK.

Nach dem Herstellen der neuen Verbindung werden im Datenbankfenster des geöffneten Access-Projekts die Objekte der SQL-Server-Datenbank, zu der die Verbindung hergestellt wurde, angezeigt.

☞ Mit einem kleinen Trick können Sie auch eine neue SQL-Server-Datenbank erstellen, ohne diese in einem Access-Projekt zu verwenden: Erstellen Sie zunächst ein neues Access-Projekt mit neuer Datenbank, wie oben in Punkt 38.2.1, *Access-Projekt und SQL-Server-Datenbank erstellen* und miteinander verbinden, dargestellt wurde, und löschen Sie anschließend die Access-Projektdatei mit dem Explorer. Auf die neue (leere) SQL-Server-Datenbank können Sie dann jederzeit von einem beliebigen Access-Projekt aus zugreifen, indem Sie diese damit verbinden.

38.3 Tabellen und Datenbankdiagramme

Tabellen können, analog zum Arbeiten in einer Access-Datenbank, in der Entwurfsansicht entworfen und bearbeitet werden. Die Entwurfsansicht von Tabellen sieht zwar in einem Access-Projekt anders aus als in einer Access-Datenbank, jedoch unterscheiden sich die Arbeitsweisen prinzipiell nicht.

Daneben gibt es aber in Access-Projekten die Möglichkeit, Tabellen mit dem Datenbankdiagrammdesigner zu definieren und zu bearbeiten. Dieser Weg wird von der Access 2000 Hilfe empfohlen und dort auch ausführlich dokumentiert. Auch wir folgen hier dieser Arbeitsweise. Daher wird zunächst gezeigt, wie Sie ein Datenbankdiagramm erstellen und bearbeiten, um anschließend die Definition und Bearbeitung von Tabellen zu behandeln.

38.3.1 Datenbankdiagramm erstellen und bearbeiten

Wenn Sie eine SQL-Server-Datenbank neu erstellt haben, ist dort zunächst kein Datenbankdiagramm vorhanden, denn Datenbankdiagramme werden mit dem SQL Server und nicht mit dem Access-Projekt physikalisch gespeichert. Umgekehrt gilt: Wenn Sie ein Access-Projekt mit einer SQL-Server-Datenbank verbinden, für die ein oder mehrere Datenbankdiagramme definiert sind, werden diese im Datenbankfenster des Projekts angezeigt und stehen zur Bearbeitung und Einsichtnahme zur Verfügung.

Datenbankdiagramm erstellen

Sie können für eine SQL-Server-Datenbank jederzeit ein neues Datenbankdiagramm erstellen, gleichgültig, ob bereits ein anderes vorhanden ist oder nicht, denn für eine SQL-Server-Datenbank kann es mehrere Datenbankdiagramme geben. Gehen Sie folgendermaßen vor, um für eine SQL-Server-Datenbank ein neues Datenbankdiagramm zu erstellen:

- Öffnen Sie ein Access-Projekt, das mit der zu bearbeitenden SQL-Server-Datenbank verbunden ist. Dabei muß der entsprechende SQL Server laufen.
- Aktivieren Sie im Datenbankfenster die Objektliste *Datenbankdiagramme*, und klicken Sie auf das Symbol *Neu*. Dann wird ein neues leeres Datenbankdiagrammfenster angezeigt.
- Fügen Sie dem Datenbankdiagramm ggf. Tabellen hinzu, indem Sie das Dialogfeld *Tabellen anzeigen* durch Klicken auf das gleichnamige Symbol öffnen und Tabellen aus dem Dialogfeld in das Datenbankdiagrammfenster ziehen.
- Speichern Sie das neue Datenbankdiagramm unter einem frei festzulegenden Namen.

☞ Sie können in ein Datenbankdiagramm eine beliebige Zusammenstellung der in der SQL-Server-Datenbank definierten Tabellen aufnehmen. Dabei erfolgt das Einfügen vorhandener Tabellen mit dem Dialogfeld *Tabellen anzeigen*, vgl. den vorletzten Aufzählungspunkt. Wollen Sie eine Tabelle aus dem Datenbankdiagramm entfernen, so markieren Sie diese und wählen aus dem Kontextmenü den Befehl *Tabelle ausblenden*. Wenn Sie die Tabelle dagegen markieren und mit dem Kontextmenübefehl *Tabelle aus Datenbank löschen* entfernen wollten, so würden sie die Tabelle physikalisch aus der

SQL-Server-Datenbank löschen! Dagegen können Sie ein komplettes Datenbankdiagramm ohne Konsequenzen für die Tabellen in der SQL-Server-Datenbank löschen, indem Sie es im Datenbankfenster markieren und anschließend die [Entf]-Taste drücken.

Datenbankdiagramm formatieren

Der Datenbankdiagrammdesigner bietet zahlreiche Möglichkeiten, das Erscheinungsbild der in das Datenbankdiagramm aufgenommenen Tabellen zu gestalten. Die wichtigsten Möglichkeiten beziehen sich darauf, einzelne oder alle Tabellen in unterschiedlicher Weise anzuzeigen.

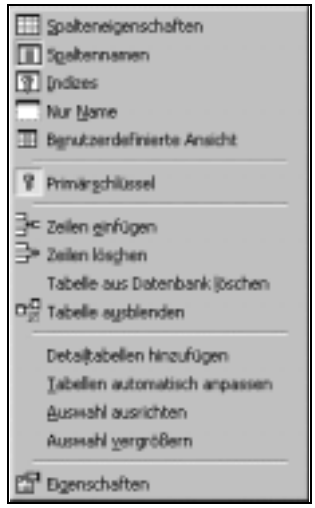


Bild 38.15: Kontextmenü für eine Tabelle im Datenbankdiagrammfenster

In Bild 38.15 ist das Kontextmenü für eine Tabelle im Datenbankdiagramm wiedergegeben. Mit seinen ersten fünf Befehlen kann die Darstellungsart der markierten Tabelle(n) bestimmt werden. Bild 38.16 gibt eine Darstellung fünf verschiedener Tabellen im Datenbankdiagramm wieder. Dabei gelten für die einzelnen Tabellen die folgenden Darstellungsarten:

Tabelle	Darstellungsart
Kategorien	Spalteneigenschaften
Lieferanten	Spaltennamen
Artikel	Indizes
Bestellungen	Nur Name
Bestelldetails	Benutzerdefinierte Ansicht

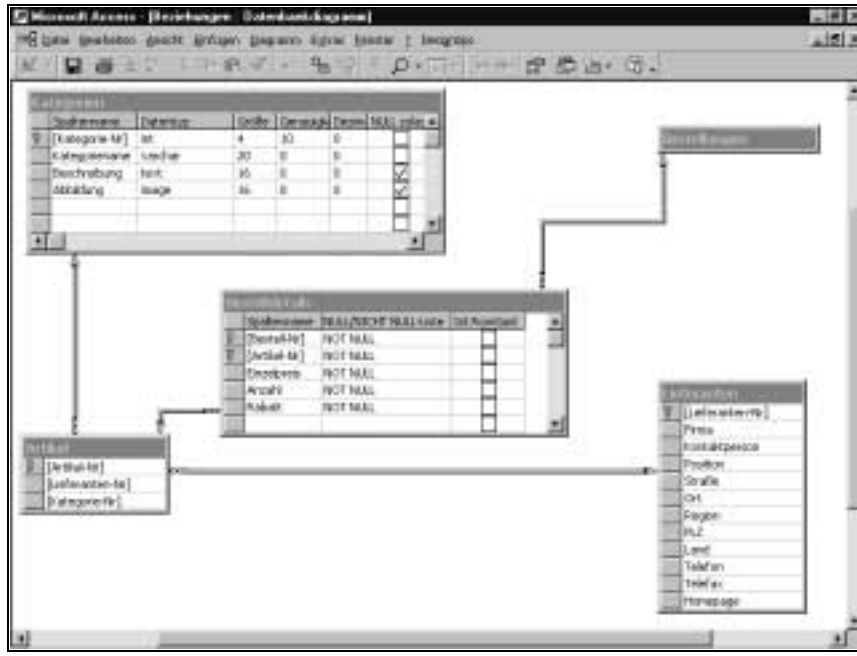


Bild 38.16: Jede Tabelle dieses Datenbankdiagramms wird in einer anderen Weise dargestellt, vgl. die o.a. tabellarische Auflistung der entsprechenden Kontextmenübefehle.

Neben diesen hier kurz dargestellten Formatierungsmöglichkeiten von Datenbankdiagrammen bietet der Designer zahlreiche weitere, die Sie bitte der Access 2000 Hilfe entnehmen (Thema: *Formatieren und Drucken von Diagrammen*) oder durch Versuch und Irrtum mit den Befehlen des Menüs *Ansicht* oder des Kontextmenüs ausprobieren.

38.3.2 Tabellen erstellen und bearbeiten

In diesem Punkt wird die prinzipielle Vorgehensweise zum Erstellen unf Bearbeiten von Tabellen eines SQL Servers mit Hilfe des Designers im Datenbankdiagramm dargestellt. Der praktische Umgang mit Tabellen und ihren Feldern erfordert, viele Details zu berücksichtigen. Darauf kann im Rahmen dieses Buches nicht eingegangen werden. Insbesondere sei darauf hingewiesen, daß hier weder auf Gültigkeitsregeln noch auf Indizes eingegangen wird. Entnehmen Sie die dafür erforderlichen Einzelheiten bitte der Access 2000 Hilfe: Über Gültigkeitsregeln informieren Sie sich am besten unter dem Thema *Tabellen*, Unterthema *Erstellen von Einschränkungen*, über Indizes unter dem Unterthema *Arbeiten mit Indizes*.

Tabelle erstellen

Um eine neue Tabelle für eine SQL-Server-Datenbank zu erstellen, gehen Sie folgendermaßen vor:

- Öffnen Sie ein Access-Projekt, das mit der zu bearbeitenden SQL-Server-Datenbank verbunden ist. Dabei muß der entsprechende SQL Server laufen.
- Aktivieren Sie im Datenbankfenster die Objektliste *Datenbankdiagramme*, und doppelklicken Sie auf ein Datenbankdiagramm. Dann wird dieses Datenbankdiagrammfenster angezeigt.
- Klicken Sie mit der rechten Maustaste auf einen freien Punkt im Datenbankdiagrammfenster, wählen Sie im Kontextmenü den Befehl *Neue Tabelle*, geben Sie im Dialogfeld *Namen wählen* einen Namen für die neue Tabelle ein, und bestätigen Sie das Dialogfeld mit OK. Dann wird die neue Tabelle in der Darstellungsart *Spalteneigenschaften* angezeigt (vgl. Bild 38.17).

Bild 38.17: Neu eingefügte Tabelle im Datenbankdiagramm.

- Fügen Sie der neuen Tabelle Felder hinzu, indem Sie Spaltennamen vergeben und die Feldattribute einstellen.
- Speichern Sie das Datenbankdiagramm. Wenn Sie diesen Schritt nicht ausführen, werden die Änderungen an den Tabellen im Datenbankdiagramm – hier also die Definition einer neuen Tabelle – nicht für die SQL-Server-Datenbank wirksam.

Tabelle bearbeiten

Die Bearbeitung einer Tabelle funktioniert analog zum Definieren einer neuen Tabelle:

- Öffnen Sie ein Datenbankdiagramm und nehmen Sie die zu bearbeitende Tabelle ggf. darin auf (Symbol und Dialogfeld *Tabelle anzeigen*).
- Führen Sie die Bearbeitungen an der Tabelle aus.
- Speichern Sie das Datenbankdiagramm. Wenn Sie diesen Schritt auslassen, werden die Bearbeitungen nicht in der SQL-Server-Datenbank wirksam.

Tabelle löschen

Verfahren Sie analog zum Bearbeiten einer Tabelle (vgl. vorangehenden Punkt), markieren Sie die Tabelle, und wählen Sie im Kontextmenü den Befehl *Tabelle aus Datenbank löschen*. Bestätigen Sie die sichernde Frage mit *Ja*.

38.3.3 Beziehung erstellen und bearbeiten

Beziehung erstellen

Eine Beziehung zwischen zwei Tabellen wird erstellt, wenn Sie im Datenbankdiagramm das Primärschlüsselfeld der (künftigen) Mastertabelle markieren und es dann auf das Fremdschlüsselfeld der (künftigen) Detailtabelle ziehen – ganz ähnlich wie im Beziehungsfenster einer Access-Datenbank. Dann wird das Dialogfeld *Beziehung erstellen* angezeigt, vgl. Bild 38.18, in dem Sie einen Namen für die Beziehung angeben und Attribute festlegen können. Allerdings weichen die Inhalte der Attribute von denen in einer Access-Datenbank ab: Es läßt sich weder eine *Aktualisierungsweitergabe an Detailfeld* noch eine *Löschweitergabe an Detaildatensatz* (sogenanntes kaskadierendes Löschen) definieren. Dies müßte jeweils mit einem Trigger realisiert werden, deren Behandlung jedoch über den Rahmen dieses Kapitels hinausgeht, vgl. jedoch dazu die Access 2000 Hilfe unter dem Thema *Erstellen und Ändern von Triggern*.

Beziehung bearbeiten

Um eine Beziehung zu bearbeiten, gehen Sie folgendermaßen vor:

- Öffnen Sie ein Datenbankdiagramm, in dem die Beziehung angezeigt wird.
- Lassen Sie das Eigenschaftenfenster anzeigen, und klicken Sie auf die Beziehungslinie, um diese zu markieren.
- Aktivieren Sie die Registerkarte *Beziehungen*, und bearbeiten Sie die Beziehung im Dialogfeld dieser Registerkarte.

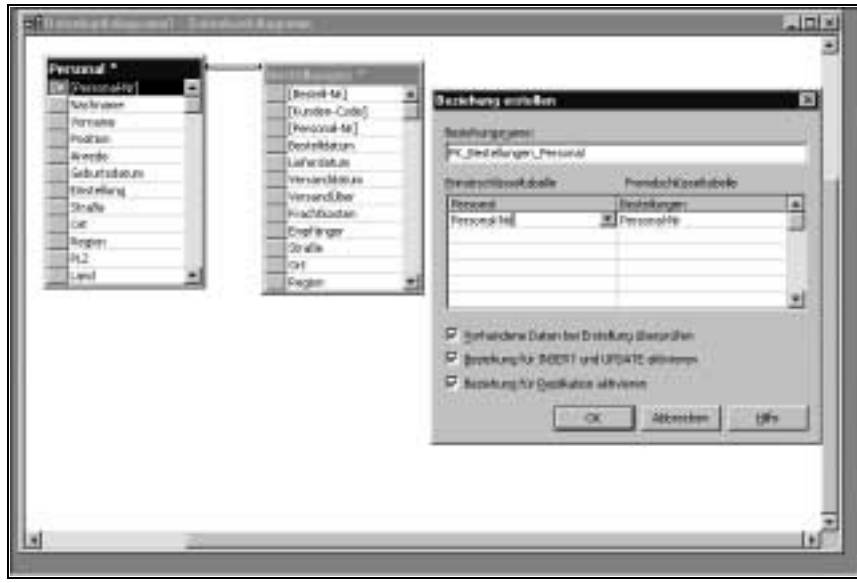


Bild 38.18: Das Dialogfeld »Beziehung erstellen« wird angezeigt, nachdem ein markiertes Primärschlüsselfeld auf das (künftige) Fremdschlüsselfeld gezogen worden ist.

- Speichern Sie das Datenbankdiagramm. Ohne diesen Schritt wird die Bearbeitung der Beziehung nicht in der SQL-Server-Datenbank wirksam.

38.4 Sichten (Views) erstellen und bearbeiten

Sichten sind den Abfragen einer Access-Datenbank sehr ähnlich und werden auch auf entsprechende Weise erstellt und bearbeitet. Gegenüber den Abfragen in Access-Datenbanken können in den Sichten der Access-Projekte jedoch nur eingeschränkte Abfragetypen realisiert werden: Es sind nur Auswahlabfragen mit der Select-Anweisung möglich, und es werden weder die *Order By*-Klausel[1] noch Parameter unterstützt. Diese Leistungen können jedoch über Gespeicherte Prozeduren realisiert werden, vgl. den folgenden Punkt 38.5, *Gespeicherte Prozeduren*.

Sicht erstellen

Gehen Sie folgendermaßen vor, um eine neue Sicht zu erstellen:

- Aktivieren Sie im Datenbankfenster die Objektliste *Sichten*, und klicken Sie auf das Symbol *Neu*. Dann wird ein leeres Fenster des Abfragedesigners geöffnet.

1. Diese kann in einer Sicht allerdings in Verbindung mit dem Schlüsselwort *Top* angeführt werden.

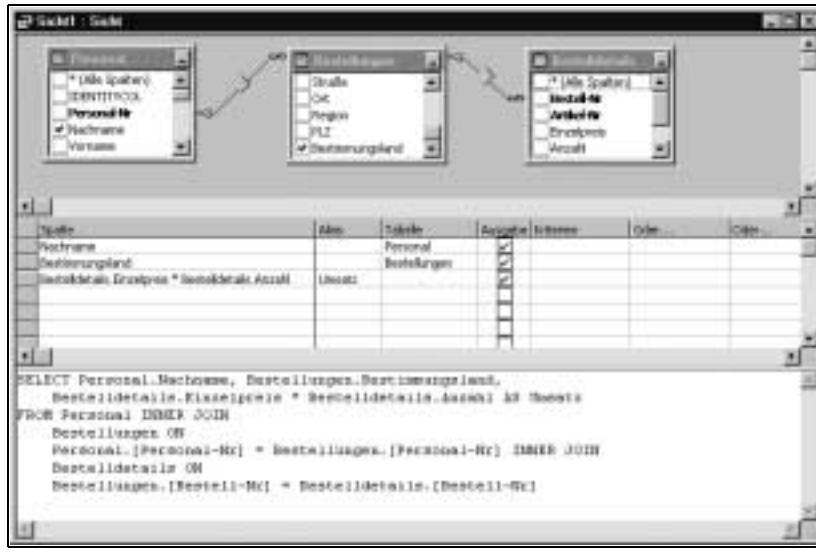


Bild 38.19: Sicht in der Ansicht des Abfragedesigners

- Klicken Sie auf das Symbol *Tabelle anzeigen*, um das gleichnamige Dialogfeld zu öffnen.
- Ziehen Sie die Tabelle(n) oder Sicht(en), die in der neuen Sicht berücksichtigt werden soll(en), aus dem Dialogfeld *Tabelle anzeigen* in den oberen Bereich (Diagrammbereich) des Abfragedesignerfensters.
- Aktivieren Sie in den Tabellen die Kontrollkästchen der Felder, die in der Sicht berücksichtigt werden sollen. Dies wird dann simultan im Rasterbereich entsprechend dargestellt und in die Formulierung der SQL Select-Anweisung aufgenommen.
- Um ein berechnetes Feld einzufügen, geben Sie den berechnenden Ausdruck im Rasterbereich in die Spalte *Spalte* und einen Namen für das berechnete Feld in die Spalte *Alias* ein. Der Sicht-Entwurf in Bild 38.19 enthält ein berechnetes Feld mit dem Aliasnamen *Umsatz* und dem berechnenden Ausdruck `Bestelldetails.Einzelpreis * Bestelldetails.Anzahl`.
- Wechseln Sie in die Datenblattansicht, um das Abfrageergebnis zu überprüfen. Zu diesem Zweck müssen Sie den Sicht-Entwurf zuvor speichern.

Zu den Einschränkungen einer Sicht im Vergleich zu den in Access-Datenbanken möglichen Abfragetypen gehört nicht die Gruppierung. Gruppierungsabfragen sind uneingeschränkt in derselben Weise in Sichten möglich. Sie wandeln eine gewöhnliche Auswahlabfrage in eine gruppierte Abfrage um, indem Sie auf das Symbol *Gruppieren* klicken. Dann wird im Rasterbereich die zusätzliche Spalte *Gruppieren nach* angezeigt, in der Sie für die einzelnen Felder *Gruppieren nach* oder eine Aggregatfunktion angeben können.

Sicht1 : Sicht

Nachname	Bestimmungsl	Umsatz
Buchanan	Frankreich	168,00 DM
Buchanan	Frankreich	98,00 DM
Buchanan	Frankreich	121,50 DM
Suyama	Deutschland	167,40 DM
Suyama	Deutschland	1.696,00 DM
Peacock	Brasilien	77,00 DM
Peacock	Brasilien	1.484,00 DM
Peacock	Brasilien	252,00 DM
Leverling	Frankreich	100,80 DM
Leverling	Frankreich	234,00 DM
Leverling	Frankreich	336,00 DM
Peacock	Belgien	2.592,00 DM

Datensatz: 1 von 2155

Bild 38.20: Sicht aus Bild 38.19 in der Datenblattansicht

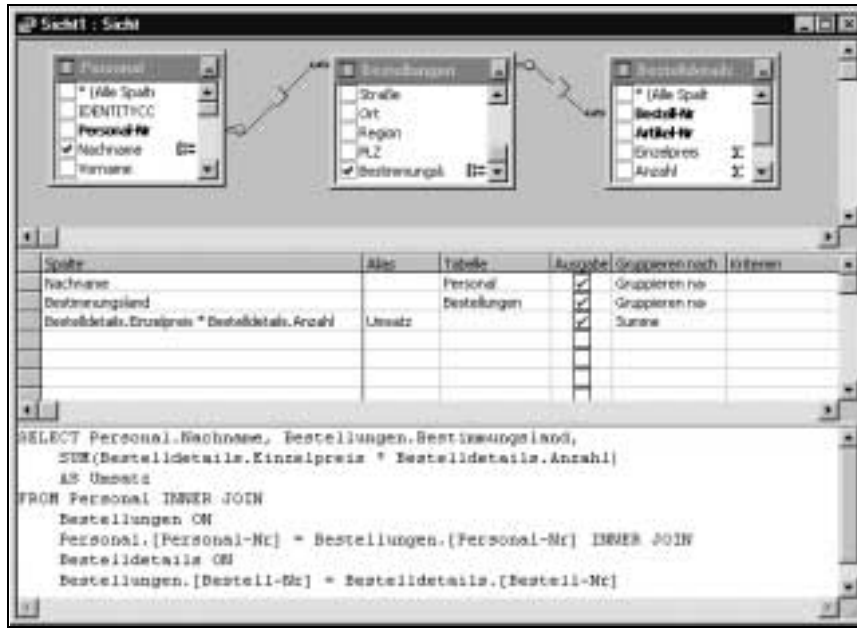


Bild 38.21: Sicht aus Bild 38.19 als gruppierte Abfrage

Sicht bearbeiten

Sie bearbeiten eine Sicht, indem Sie diese aus dem Datenbankfenster in der Entwurfsansicht öffnen und dann im Fenster des Abfragedesigners die Änderungen vornehmen.

38.5 Gespeicherte Prozeduren erstellen und bearbeiten

Eine Gespeicherte Prozedur ist eine kompilierte Sammlung von SQL-Anweisungen und optionalen Control-Of-Flow-Anweisungen, die unter einem Namen gespeichert und als eine Einheit verarbeitet werden. Gespeicherte Prozeduren werden in einer SQL-Server-Datenbank gespeichert und können aus einer Anwendung heraus ausgeführt werden. Mit Gespeicherten Prozeduren können benutzerdeklarierte Variablen, eine bedingte Ausführung und weitere Programmierfunktionen eingesetzt werden. Gespeicherte Prozeduren können auch Parameter akzeptieren und Parameter ausgeben. Die Gespeicherte Prozedur *Umsätze nach Jahr* des Beispielprojekts *NordwindCS.adp*, deren SQL-Code in Bild 38.22 wiedergegeben ist, definiert beispielsweise die beiden Parameter *Anfangsdatum* und *Enddatum* und gibt die Felder *Versanddatum*, *Bestell-Nr*, *Zwischensumme* und *Jahr* aus, wobei die beiden Parameter als Kriterien in der *Where*-Klausel verwendet werden. Eine derartige Abfrage wäre nicht als Sicht formulierbar, weil Sichten keine Parameter akzeptieren.

Bild 38.22: Gespeicherte Prozedur »Umsätze nach Jahr« in der Entwurfsansicht

Gespeicherte Prozedur erstellen

Sie erstellen eine Gespeicherte Prozedur auf die folgende Weise:

- Aktivieren Sie im Datenbankfenster die Objektliste *Gespeicherte Prozeduren*, und klicken sie auf das Symbol *Neu*. Dann wird im Designfenster für Gespeicherte Prozeduren das Skelett einer Prozedur zum Erzeugen einer Gespeicherten Prozedur angezeigt, vgl. Bild 38.23.

Bild 38.23: : Skelett der Prozedur zum Erzeugen einer Gespeicherten Prozedur, unmittelbar nach dem Klicken auf das Symbol »Neu«

- Ersetzen Sie den vorläufigen Namen – im Beispiel *GespeichertеProzedur1* – durch den Namen, unter dem die Gespeicherte Prozedur gespeichert werden soll. Der Text zwischen einem /* ... */ Paar stellt Kommentar dar, den Sie ggf. ändern und zum SQL-Text machen können.

Geben Sie weiteren Text in SQL ein. Insgesamt könnte der Entwurf beispielsweise so aussehen, wie in Bild 38.24 wiedergegeben: Es wird eine Auswahlabfrage für die Felder *Artikel-Nr* und *Artikelname* der Tabelle *Artikel* gebildet, die nur Datensätze mit dem Wert 0 für *Auslaufartikel* wiedergibt und diese nach dem Feld *Artikelname* sortiert.

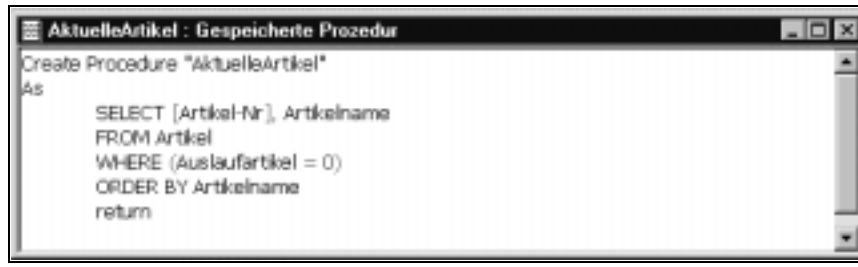


Bild 38.24: Geänderte Prozedur vor dem Speichern

- Speichern Sie den Prozedurentwurf. Dabei wird die *Create Procedure*-Anweisung ausgeführt, die die Prozedur kompiliert und unter dem angegebenen Namen speichert.
- Öffnen Sie anschließend die Gespeicherte Prozedur in der Datenblattansicht (sofern dies einen Sinn ergibt, denn nicht alle Gespeicherten Prozeduren geben Datensätze als Ergebnis aus).

Beim erneuten Öffnen der Gespeicherten Prozedur in der Entwurfsansicht erscheint der Anfangsausdruck *Create Procedure* ersetzt durch den Ausdruck *Alter Procedure*. Dies ist das Zeichen dafür, daß die Prozedur als Gespeicherte Prozedur erzeugt und kompiliert wurde.

Gespeicherte Prozedur bearbeiten

Um eine Gespeicherte Prozedur zu bearbeiten, öffnen Sie diese in der Entwurfsansicht, bearbeiten sie den SQL-Text und speichern abschließend.

38.5.1 Trigger erstellen und bearbeiten

Ein *Trigger* ist eine besondere Form einer Gespeicherten Prozedur. Ein Trigger ist stets einem der drei Ereignisse *Update*, *Insert* oder *Delete* einer bestimmten Tabelle zugeordnet und wird automatisch gestartet, wenn dieses Ereignis eintritt. In der Sprache einer Access-Datenbank könnte ein Trigger auch als Ereignisprozedur bezeichnet werden. In Bild 38.25 ist ein Trigger der Tabelle *Kunden* wiedergegeben. Er wurde vom Upsizing-Assistenten

beim Umwandeln der Datenbank *Nordwind.mdb* in ein Access-Projekt erzeugt (vgl. dazu oben Punkt 38.1, *Access-Datenbank mit dem Upsizing-Assistenten in Access-Projekt umwandeln*, wo dieses Beispiel verwendet wird). Der Trigger wird ausgelöst, wenn das Ereignis *Update* für die Tabelle eintritt. Die SQL-Anweisungen zwischen BEGIN und END sorgen dafür, daß eine Änderung im Primärschlüsselfeld *Kunden-Code* der Mastertabelle *Kunden* an das gleichnamige Fremdschlüsselfeld in der Detailtabelle weitergegeben wird. Trigger lassen sich auch sehr gut für die Formulierung von Gültigkeitsregeln formulieren, auch solchen, die auf Felder anderer Tabellen Bezug nehmen.

Bild 38.25: Update-Trigger der Tabelle »Kunden«

Trigger erstellen

Da Trigger nicht unabhängig von einer bestimmten Tabelle existieren können, werden sie konsequenterweise nicht als eigenständige Objekte im Datenbankfenster aufgelistet. Das Erstellen und Bearbeiten eines Triggers erfolgt daher stets unter Bezugnahme auf die zugeordnete Tabelle. Gehen Sie folgendermaßen vor, um einen Trigger zu erstellen:

- Klicken Sie im Datenbankfenster mit der rechten Maustaste auf die Tabelle, für die Sie einen Trigger erstellen wollen.
- Wählen Sie im Kontextmenü den Befehl *Trigger.*

Bild 38.26: Dialogfeld Trigger für Tabellen: Artikel

- Klicken Sie im Dialogfeld *Trigger für Tabellen: Tabellenname* (vgl. Bild 38.26) auf die Schaltfläche *Neu*. Dann wird in einem Fenster des Trigger-Designers das Skelett einer Triggerprozedur angezeigt, vgl. Bild 38.27.

Bild 38.27: Skelett einer Triggerprozedur unmittelbar nach dem Klicken auf die Schaltfläche »Neu«

- Überschreiben Sie den hinter *Create Trigger* angeführten vorläufigen Namen mit einem geeigneten anderen Namen für die Triggerprozedur, und geben Sie den weiteren benötigten SQL-Code ein.
- Speichern Sie die Prozedur.

Nach dem letzten Schritt ist die Triggerprozedur ein Objekt der Tabelle, für die sie definiert wurde, geworden und wird bei dem betreffenden Update-, Insert- oder Delete-Ereignis ausgeführt.

Trigger bearbeiten

Um einen vorhandenen Trigger zu bearbeiten, gehen Sie folgendermaßen vor:

- Klicken Sie im Datenbankfenster mit der rechten Maustaste auf die Tabelle, deren Trigger Sie bearbeiten wollen.
- Wählen Sie im Kontextmenü den Befehl *Trigger*.
- Wählen Sie im Dialogfeld *Trigger für Tabellen: Tabellenname* (vgl. oben Bild 38.26) in der Dropdown-Liste *Triggername* den zu bearbeitenden Trigger aus, und klicken Sie auf die Schaltfläche *Bearbeiten*. Dann wird die Triggerprozedur angezeigt, wie oben in Bild 38.25 für den Update-Trigger der Tabelle *Kunden* dargestellt.
- Nehmen Sie die erforderliche Bearbeitung vor, und speichern Sie schließlich die Triggerprozedur.

38.6 Datenquellen und Abfrageparameter in Formularen und Berichten

SQL-Select-Anweisungen

Die Anbindung von Datenquellen an ganze Formulare und Berichte oder einzelne Steuerelemente ist in Access-Projekten prinzipiell genauso geregelt wie in Access-Datenbanken. Das bedeutet insbesondere, daß Sie für die Formular- oder Berichteigenschaft *Datenherkunft* den Namen einer Tabelle oder Sicht, aber auch eine als Text formulierte SQL-Select-Anweisung anführen dürfen. SQL-Select-Anweisungen können auch als Datenherkunft für einzelne Steuerelemente wie Kombinations- oder Listenfeld angegeben werden. Dabei gelten nicht die Einschränkungen, die für die Formulierung einer Sicht bestehen: Sie dürfen bei einer SQL-Select-Anweisung, die Sie als Datenquelle in einem Formular/Bericht oder einem Steuerelement anführen, auch die *Order By*-Klausel spezifizieren.

Abfrageparameter

Formulare und Berichte in Access-Projekten besitzen im Vergleich zu Access-Datenbanken die zusätzliche Eigenschaft *Eingabeparameter*. Damit können auf bequeme Weise Werte für Parameter einer Gespeicherten Prozedur, die dem Formular/Bericht als Datenquelle dient, versorgt werden. Die nützliche Verwendung dieser Eigenschaft kann am besten am Bericht *Umsätze nach Jahr* aus dem Beispiel Access-Projekt *NordwindCS.adp* demonstriert werden. Wenn Sie diesen Bericht in der Seitenansicht öffnen, wird zunächst das Dialogfeld *Umsätze nach Jahr-Dialog* mittels einer dem Ereignis *Beim Öffnen* zugeordneten Ereignisprozedur angezeigt, vgl. Bild 38.28. Wenn auf die Schaltfläche OK geklickt wird, wird das Formular nicht geschlossen, sondern nur ausgeblendet, so daß andere Objekte darauf zugreifen können.

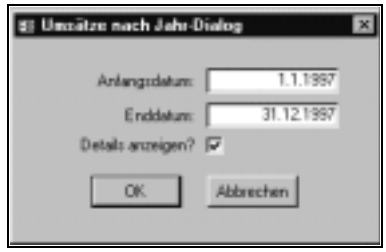


Bild 38.28: Dialogfeld »Umsätze nach Jahr-Dialog«, das beim Öffnen des Berichts »Umsätze nach Jahr« angezeigt wird

Die Eigenschaft *Eingabeparameter* des Berichts *Umsätze nach Jahr* ist auf den Text eingestellt, dessen Anfang in Bild 38.29 zu erkennen ist und der in ganzer Länge folgendermaßen lautet:

```
@Anfangsdatum datetime = Forms![Umsätze nach Jahr-
Dialog]![Anfangsdatum], @Enddatum datetime = forms![Umsätze nach
Jahr-Dialog]![Enddatum]
```

Bild 38.29: Eigenschaftenfenster für den Bericht »Umsätze nach Jahr«

@Anfangsdatum und *@Enddatum* sind als Parameter in der Gespeicherten Prozedur *Umsätze nach Jahr* definiert, die dem Bericht als Datenquelle zugrunde liegt. Mit der Angabe zur Eigenschaft *Eingabeparameter* werden den beiden Parametern *@Anfangsdatum* und *@Enddatum* die Werte der entsprechenden Steuerelemente aus dem Formular *Umsätze nach Jahr-Dialog* zugewiesen und automatisch an die Gespeicherte Prozedur übergeben. Diese Konstruktion erscheint deutlich eleganter und bequemer als eine entsprechende bei einer Access-Datenbank, in der für die Parameterübergabe eine eigene VBA-Prozedur erstellt werden müßte.

38.7 Sicherungskopien erstellen und wiederherstellen

Mit dem SQL Server 7.0 und damit mit MSDE können auf sehr einfache Weise Sicherungskopien einzelner Datenbanken des SQL Servers erstellt werden, die auf ebenso bequeme Weise als Datenbank wiederhergestellt werden können. Damit wird nicht nur die Sicherung von Daten erleichtert, sondern nicht zuletzt auch das Portieren auf einen anderen Rechner, vgl. den folgenden Punkt 38.8, *Access-Projekt portieren*.

Sicherungskopie von SQL-Server-Datenbank erstellen

Gehen Sie folgendermaßen vor, um eine Sicherungskopie von einer SQL-Server-Datenbank zu erstellen:

- Öffnen Sie ein Projekt, das mit der zu sichernden Datenbank verbunden ist.
- Wählen Sie im Menü *Extras* zum Befehl *Datenbank-Dienstprogramme* den Unterbefehl *Sicherungskopie*.
- Bestimmen Sie im Dialogfeld *Sicherungskopie* den Speicherordner, und geben Sie einen Namen für die Sicherungsdatei an, der die Erweiterung *.dat* aufweisen sollte.

Nach erfolgreicher Sicherung wird dies in einer Meldung mitgeteilt, die auch darauf hinweist, daß nur die serverbezogenen Objekte gesichert wurden, nicht jedoch das Access-Projekt mit seinen Formularen, Berichten etc. Dies müßte in einem eigenständigen Akt als gewöhnliche Kopie der Access-Projektdatei erfolgen.

SQL-Server-Datenbank aus Sicherungskopie wiederherstellen

Gehen Sie folgendermaßen vor, um von einer Sicherungskopie die SQL-Server-Datenbank wiederherzustellen:

- Öffnen Sie ein Access-Projekt, das mit dem SQL Server verbunden ist, für den Sie die Datenbank wiederherstellen wollen.
- Wählen Sie im Menü *Extras* zum Befehl *Datenbank-Dienstprogramme* den Unterbefehl *Wiederherstellen*.
- Wählen Sie im Dialogfeld *Wiederherstellen* die Sicherungsdatei, die die Informationen der wiederherzustellenden Datenbank enthält, und klicken Sie auf OK. Bei erfolgreicher Wiederherstellung teilt Access dies und den Namen mit, unter dem die Datenbank wiederhergestellt wurde.

38.8 Access-Projekt portieren

Um ein Access-Projekt auf einen anderen PC zu portieren, reicht es nicht aus, die Access-Projektdatei dorthin zu kopieren, weil sich die Datenbank ja auf dem SQL Server befindet. Der SQL Server selbst läßt sich jedoch nicht in Form einer einfachen Kopie portieren. Vielmehr muß er auf dem anderen PC zunächst installiert werden (sofern dies nicht bereits geschehen ist), und anschließend muß die Datenbank, die mit dem Access-Projekt portiert werden soll, dort erstellt werden.

Gehen Sie im einzelnen folgendermaßen vor, um ein Access-Projekt auf einen anderen Rechner zu portieren:

- Erstellen Sie eine Sicherungskopie von der SQL-Server-Datenbank, mit der das Access-Projekt verbunden ist (vgl. dazu den vorangehenden Punkt 38.7, *Sicherungskopien erstellen und wiederherstellen*).
- Kopieren Sie die Access-Projektdatei und die Sicherungskopiedatei auf den Zielrechner.
- Installieren Sie auf dem Zielrechner, falls erforderlich, den SQL Server.
- Wiederherstellen Sie auf dem Zielrechner aus der Sicherungskopiedatei die SQL-Server-Datenbank.
- Öffnen Sie auf dem Zielrechner das Access-Projekt, und verbinden Sie es mit der wiederhergestellten SQL-Server-Datenbank.

38.9 SQL-Datenbank löschen

Um eine SQL-Datenbank physikalisch zu löschen, gehen Sie folgendermaßen vor:

- Öffnen Sie ein Access-Projekt, das mit der zu löschenden Datenbank verbunden ist.
- Wählen Sie im Menü *Extras* zum Befehl *Datenbank-Dienstprogramme* den Unterbefehl *SQL-Datenbank löschen*.
- Bestätigen Sie das in Bild 38.30 wiedergegebene Dialogfeld mit *Ja*. Dann werden die möglicherweise geöffneten Objekte der Datenbank geschlossen, die Verbindung des Access-Projekts zur Datenbank gelöst und die Datenbank gelöscht.

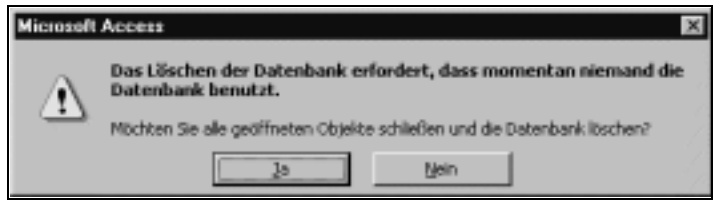


Bild 38.30: Dieses Dialogfeld wird vor dem endgültigen Löschen der Datenbank angezeigt.

Da eine gelöschte Datenbank nicht wiederherstellbar ist, sollten Sie möglicherweise zuvor aus Sicherheitsgründen eine Sicherungskopie erstellen, aus der Sie dann die Datenbank ggf. wiederherstellen können.

☞ Nach meiner Erfahrung klappt das Löschen einer SQL-Server-Datenbank nicht, wenn zuvor in Verbindung mit einem Designtool des SQL Servers eine Störung aufgetreten war, die der Benutzer manchmal gar nicht merkt. Vergewissern Sie sich auf zweifache Weise, ob die Datenbank tatsächlich gelöscht wurde: 1) Wählen Sie aus dem Menü *Datei* den Befehl *Verbindung*, und prüfen Sie im Dialogfeld *Datenlinkeigenschaften*, ob die Datenbank noch auf dem SQL Server vorhanden ist. 2) Prüfen Sie mit dem Explorer, ob die beiden Dateien mit der Namenserweiterung *.mdf* bzw. *.ldf* mit dem Namen der zu löschenden Datenbank noch im Ordner *\Data* (Unterordner zum Programmordner des SQL Servers) vorhanden sind. Wenn Sie feststellen, daß die Datenbank nur als Eintrag im SQL Server, nicht jedoch mit ihren Dateien gelöscht wurde, so löschen Sie die entsprechende *.mdf*- und *.ldf*-Datei. Wenn die Datenbank noch auf dem SQL Server registriert ist, verfahren Sie am besten wie folgt: Fahren Sie das Betriebssystem herunter und anschließend wieder herauf, und wiederholen Sie den Löschvorgang für die Datenbank von Anfang an erneut.

Kapitel 39

Ausdrücke, Funktionen und Operatoren

39.1 Komponenten eines Ausdrucks

Ein Ausdruck ist eine Kombination von Operatoren, Konstanten, Funktionen, Objektnamen usw., die einen Wert ergibt. Ein Ausdruck kann eine bis viele verschiedene Komponenten enthalten. Die in einem Ausdruck möglichen Komponenten sind in der folgenden Übersicht zusammengestellt:

Komponente	Beispiel	Beschreibung
Operator	+, =, >, Oder, Wie	Führt an einer oder mehreren anderen Komponenten eine Operation aus.
Bezeichner	[Postleitzahl], Formulare!Aufträge! AuftragsNr	Gibt ein Feld, ein Steuerelement, ein Formular, eine Eigenschaft usw. an.
Funktion	Datum, Summe, DomAnzahl	Gibt einen Funktionswert aus. Zulässig sind auch in Access-Basic geschriebene benutzerdefinierte Funktionen.
Zeichenkette	"Hamburg", #1.1.93#	Zahl, Datumsangabe oder Text. Eine Zeichenkette wird ohne Interpretation zeichentreu übernommen.
Konstante	Wahr, Falsch, Ja, Nein, Null	Fester Wert, der eine Zahl oder ein Text sein kann.

39.2 Wie Sie einen Ausdruck eingeben

Manche Ausdrücke sind im Verhältnis zu den Eingabefeldern sehr lang. In diesem Falle empfiehlt es sich, mit der Tastenkombination ⇧+F2 oder dem Befehl *Zoom* aus dem Kontextmenü das Dialogfeld *Zoom* zu öffnen und den Ausdruck darin einzugeben.

Im allgemeinen schreiben Sie einen Ausdruck selbst. Access bietet jedoch an vielen Stellen ein Dropdown-Listenfeld an, das Sie aufschlagen und aus dem Sie eine Zeichenkette oder eine Konstante wählen können.

Wenn Sie einen Ausdruck in Eingabefelder, z.B. im Eigenschaftenfenster oder in Kriterienfeldern einer Abfrage, eingeben, ergänzt Access die Schreibweise des von Ihnen eingegebenen Ausdrucks in den folgenden Fällen:

- Um Namen von Objekten werden eckige Klammern [ ] gesetzt, auch wenn diese, wie bei Namen ohne Leerzeichen, syntaktisch nicht erforderlich sind.
- Datumsangaben werden von je einem Zeichen # eingerahmt.
- Texte werden zwischen Anführungszeichen gesetzt.

Diese Routine wird allerdings von Access nicht auf allen Ebenen konsequent eingehalten. Daher sollten Sie sich angewöhnen, eckige Klammern für Namen, Anführungszeichen für Texte und das Zeichen # für Datumsangaben auf alle Fälle zu schreiben.

39.2.1 Namen und Bezeichner

In diesem Kapitel wird im Punkt 39.5, *Arbeiten mit dem Ausdrucks-Generator*, der Umgang mit dem *Ausdrucks-Generator* erklärt. Dieser erweist sich als besonders nützlich beim Erstellen eines Ausdrucks mit einem komplizierten Bezeichner. Daher ist der Einsatz des Ausdrucks-Generators gerade für die Fälle, auf die sich der vorliegende Punkt bezieht, zu empfehlen. Die Lektüre dieses Punktes ist dennoch nicht überflüssig, weil Sie sich in jedem Falle mit den Grundzügen von Bezeichnern in Ausdrücken auskennen sollten.

Namen

Namen von Objekten, wie Feldnamen, Steuerelementnamen, Formularnamen usw., die nicht von Access, sondern vom Benutzer vergeben wurden, müssen prinzipiell zwischen eckigen Klammern angegeben werden. Diese Regel brauchen Sie nur dann nicht zu befolgen, wenn der Name keine Leer- oder Sonderzeichen enthält. Aber auch diese Ausnahme ist nicht ohne Ausnahme: Die Dom-Funktionen (z.B. DomAnzahl) beispielsweise akzeptieren Namen unter allen Umständen nur, wenn sie zwischen eckigen Klammern stehen.

Bezeichner

Unter einem Bezeichner wird von Access eine Angabe verstanden, die einen benutzerdefinierten Namen, z.B. einen Formularnamen oder einen Steuerelementnamen, eindeutig identifiziert. Zur eindeutigen Identifizierung reicht es manchmal aus, allein den Namen anzugeben. Im allgemeinen ist es jedoch erforderlich, die Namen der Objekte, in denen das zu bezeichnende Objekt enthalten ist, dem Namen vorangestellt ebenfalls anzugeben. Dies trifft stets dann zu, wenn Sie sich von einem anderen Objekt aus als dem, welches das zu bezeichnende Objekt enthält, darauf beziehen wollen. Wenn Sie beispielsweise in einem Formular den Namen eines in diesem Formular enthaltenen Steuerelements angeben wollen, so reicht es aus, allein diesen Namen anzugeben. Wenn Sie sich jedoch von einem anderen Formular aus darauf beziehen, müssen Sie eine erweiterte Angabe entsprechend den im folgenden beschriebenen Regeln machen.

Ein Bezeichner kann die folgenden drei Komponenten aufweisen:

- Benutzerdefinierte Namen von Steuerelementen, Formularen usw.
- Die Bezeichner-Operatoren ! und . (Punkt). Sie dienen dazu, die Namen in einer Kette von Namensangaben gegeneinander zu trennen. Dabei steht das Ausrufezeichen ! stets vor einem benutzerdefinierten Namen, während der Punkt vor einem von Access definierten Namen steht.
- Die Namen vordefinierter Systemobjekte, z.B. *Formulare*, *Berichte* und *Bildschirm* (englisch *Forms*, *Reports* und *Screen*). Wenn Sie ein Formular oder einen Bericht oder ein Steuerelement daraus bezeichnen wollen, müssen Sie einen dieser Namen stets als erste Komponente des Bezeichners angeben. Dabei verwenden Sie den Namen *Formulare* bzw. *Berichte*, wenn Sie danach einen konkreten Formular- bzw. Berichtsnamen anführen, wie in dem Ausdruck `Formulare![Hauptaufträge]!AuftragsNr`. Den Namen *Bildschirm* können Sie in Verbindung mit einer seiner beiden Eigenschaften *AktivesFormular* bzw. *AktiverBericht* (*ActiveForm* bzw. *ActiveReport*) verwenden, um sich auf das gerade aktive Formular bzw. den gerade aktiven Bericht zu beziehen. Auf diese Weise können Sie einen Ausdruck schreiben, der sich auf ein Formular oder einen Bericht bezieht, dessen Name wechseln kann. Der folgende Ausdruck bezeichnet das Steuerelement *Nummer* im aktiven Formular, welches immer dies sei:

```
Bildschirm.AktivesFormular![Nummer]
```

Bezug auf Steuerelemente in Unterformularen/berichten. Bezeichner für Steuerelemente in Unterformularen/berichten sehen auf den ersten Blick sehr verwirrend aus. Der folgende Ausdruck beispielsweise bezeichnet das Steuerelement *Firma* im Unterformular *Lieferanten*, welches in das Hauptformular *AngeboteBearbeiten* eingebettet ist:

```
Formulare![AngeboteBearbeiten]![Lieferanten].Formular!Firma
```

Diese notwendige Syntax im Zusammenhang mit Unterformularen/berichten erklärt sich folgendermaßen: Ein Unterformular/bericht ist für das Hauptformular ein Steuerelement und wird daher insoweit wie ein solches bezeichnet; daraus erklärt sich der erste Teil

```
Formulare![AngeboteBearbeiten]![Lieferanten]
```

im Bezeichner. Das Steuerelement *Lieferanten* ist jedoch vom Typ Unterformular/bericht und besitzt daher die Eigenschaft *Formular*. Daher wird mit dem Teilausdruck

```
Formulare![AngeboteBearbeiten]![Lieferanten].Formular
```

das Steuerelement *Lieferanten* in seiner Eigenschaft als Formular bezeichnet. Da das Steuerelement *Firma* in diesem (Unter-) Formular liegt, wird es danach, getrennt durch ein Ausrufezeichen, angegeben.

Noch komplizierter, jedoch nach den vorangehenden Ausführungen konsequent, gestaltet sich die Bezeichnung eines Steuerelements in einem Unterformular/bericht, das/der selbst wiederum in einen anderen Unterformular/bericht geschachtelt ist. Nehmen Sie beispielsweise den folgenden Ausdruck:

```
Formulare![Auftrag]![Teilauftrag].Formular![Artikel].Formular![Artik
elname]
```

Dieser Ausdruck bezeichnet das Steuerelement *Artikelname* im Unterformular *Artikel*, das in das Unterformular *Teilauftrag* eingebettet ist, welches selbst wiederum Unterformular im Hauptformular *Auftrag* ist.

Bezugnahme auf Steuerelemente im übergeordneten Hauptformular/bericht vom Unterformular/bericht aus. Um sich von einem Unterformular/bericht aus auf ein Steuerelement im übergeordneten Hauptformular zu beziehen, können Sie statt des Hauptformularnamens die Eigenschaft *Hauptobjekt* (*Parent*) verwenden. Angenommen, Sie wollen in einem berechneten Textfeld eines Unterformulars den Wert aus dem Steuerelement *Gewicht* im Hauptformular ermitteln, so können Sie dies mit dem folgenden Ausdruck erreichen:

```
= Hauptobjekt!Gewicht
```

39.2.2 Datums- und Zeitwerte

Datums- und Zeitwerte werden in einem der landesüblichen Formate angegeben. Einer Datums-/Zeitangabe muß das Zeichen # voran- und nachgestellt werden, wie im folgenden Ausdruck:

```
#01.07.99#
```

Sie könnten den vorstehenden Ausdruck beispielsweise auch in der Form #1.7.99# oder #1.7.1999# angeben, und auch viele weitere Formen werden von Access erkannt.

39.2.3 Texte

Texte müssen im allgemeinen zwischen Anführungszeichen stehen. Dabei gelten als Text nur solche Zeichenketten, die vom Benutzer stammen. Für vordefinierte Werte, wie Werte für Eigenschaften im Eigenschaftenfenster eines Formulars, gilt dies nicht; diese können ohne Anführungszeichen eingegeben werden.

39.2.4 Wann muß das Gleichheitszeichen vorangestellt werden?

Gehen Sie davon aus, daß Sie einem Ausdruck in der Regel ein Gleichheitszeichen voranstellen müssen, wenn Sie nicht eine einfache Konstante oder einen Ausdruck mit einem Vergleichsoperator eingeben. Manchmal ist das Gleichheitszeichen allerdings nicht erforderlich, wo man dies erwarten würde. Gelegentlich weist Access dann darauf hin. Leider läßt sich für diesen Zusammenhang keine allgemeinverbindliche Regel formulieren, so daß Sie gelegentlich nach der Methode Versuch und Irrtum verfahren müssen.

39.3 Operatoren

39.3.1 Arithmetische Operatoren

Die folgenden arithmetischen Operatoren stehen zur Verfügung:

Operator	Bedeutung
*	Multiplikation
+	Addition
-	Subtraktion oder Vorzeichenumkehr
/	Division von Gleitkommazahlen
\	Division von Integerzahlen
^	Potenzieren
Mod	Division zweier Zahlen und Ausgabe des Rests

39.3.2 Vergleichsoperatoren

Die folgenden Vergleichsoperatoren stehen zur Verfügung:

Operator	Bedeutung
<	Kleiner als
<=	Kleiner als oder gleich

Operator	Bedeutung
>	Größer als
>=	Größer als oder gleich
=	Gleich
<>	Ungleich

39.3.3 Verkettungsoperatoren

Der folgende Verkettungsoperator steht zur Verfügung:

Operator	Bedeutung
&	Verkettet zwei Operanden

39.3.4 Logische Operatoren

Die folgenden logischen Operatoren stehen zur Verfügung:

Operator	Bedeutung
Und	Logisches Und
Äqu	Logische Äquivalenz
Imp	Logische Implikation
Nicht	Logisches Nicht
Oder	Inklusives Oder
ExOder	Exklusives Oder

39.3.5 Weitere Operatoren

Operator	Bedeutung
Zwischen ... Und	Prüft, ob ein Wert zwischen zwei Grenzen liegt.
In	Prüft, ob ein Wert in einer Wertliste vorkommt.
Ist Null Oder: Ist Nicht Null	Prüft, ob ein Wert Null ist oder nicht Null ist.
Wie	Prüft, ob ein Zeichenfolgenausdruck einem als Suchbegriff angegebenen Zeichenfolgenausdruck gleicht; dabei dürfen Stellvertreterzeichen angegeben werden.

39.4 Stellvertreterzeichen

Das Sternchen (*), das Fragezeichen (?), das Zeichen #, das Ausrufezeichen (!), der Bindestrich (-), und die eckigen Klammern ([ ]) sind Stellvertreterzeichen. Sie können diese Zeichen in Abfragen, Befehlen und Ausdrücken ver-

wenden, um alle Datensätze, Dateinamen oder andere Elemente ein- oder auszuschließen, die mit bestimmten Zeichen beginnen oder einem gewissen Muster entsprechen. Die folgende Zusammenstellung bietet einen Überblick über Verwendungen der verschiedenen Stellvertreterzeichen .

Zeichen	Beispiel	Bedeutung
*	ma* findet man, Mann und Mauer; *er findet Heiner, einer und Schornsteinfegermeister	Das Sternchen wirkt wie das DOS-Stellvertreterzeichen *, d.h., es kann einer beliebigen Anzahl Zeichen entsprechen. Im Unterschied zu DOS kann es in Access jedoch auch für das erste oder letzte Zeichen stehen.
?	R?gen findet Regen und Rügen	Steht für ein beliebiges einzelnes Zeichen; wirkt wie bei DOS.
#	1#3 findet 103, 113, 123	Steht für eine beliebige einzelne Ziffer.
[ ]	R[eü]gen findet Regen und Rügen, nicht jedoch Rogen	Steht für eines der innerhalb der eckigen Klammern angegebenen Zeichen.
!	!eü]gen findet Rogen, nicht jedoch Regen und Rügen	Steht für eines der innerhalb der eckigen Klammern nicht angegebenen Zeichen.
-	b[a-c]d findet bad, bbd und bcd	Steht für ein einzelnes, beliebiges Zeichen innerhalb des angegebenen Bereichs.

39.5 Arbeiten mit dem Ausdrucks-Generator

Mit dem Ausdrucks-Generator bietet Access Ihnen ein Werkzeug an, das Sie an praktisch allen Stellen, an denen Sie in Access einen Ausdruck eingeben können, aufrufen können, um sich den Ausdruck im Dialog vom Assistenten erstellen zu lassen. (In VBA steht der Ausdrucks-Generator allerdings nicht zur Verfügung.) Kurze und einfache Ausdrücke wie z.B. =*Datum()* oder =*Nettobetrag*0,16* werden Sie wahrscheinlich direkt eingeben, weil dies wesentlich schneller zu erledigen ist. Der Ausdrucks-Generator erweist sich aber als besonders nützlich, wenn Sie Ausdrücke mit relativ komplizierten Bezeichner-Angaben erstellen müssen. An einem derartigen Beispiel soll im folgenden der Umgang mit dem Ausdrucks-Generator demonstriert werden.

Beispiel

Aufgabe. Geben Sie in die Abfrage *FrmAdressenSuchen* der Datenbank Adressen.mdb von der Beispiel CD-ROM für das berechnete Feld *Firma* den Kriterienausdruck

```
Wie [Formulare]![AdressenSuchen]![Firma] & "*"
```

ein, indem Sie sich diesen vom Ausdrucks-Generator erstellen lassen. Wenn Sie in der Praxis diesen Ausdruck bereits sicher kennen, werden Sie ihn natürlich direkt eintippen. Hier geht es jedoch um eine Demonstration des Ausdrucks-Generators, dessen Arbeitsweise besser nachvollzogen werden kann, wenn der zu erstellende Ausdruck vorher bekannt ist.

Gehen Sie folgendermaßen vor:

- Öffnen Sie ggf. die Datenbank *Adressen.mdb* von der Beispiel CD-ROM, und öffnen Sie darin die Abfrage *FrmAdressenSuchen* in der Entwurfsansicht.
- Löschen Sie in der Zeile *Kriterien* für das Feld *Firma* den Ausdruck, der dort bereits eingegeben wurde, und belassen Sie diesem Feld den Fokus.

Klicken Sie auf die Symbol-Schaltfläche *Aufbauen*, um den Ausdrucks-Generator aufzurufen. Dann wird das Dialogfeld *Ausdrucks-Generator* angezeigt, vgl. Bild 39.1.

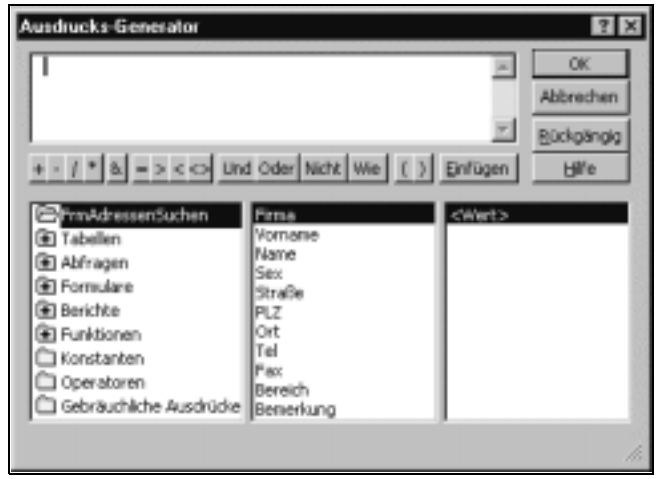


Bild 39.1: Dialogfeld »Ausdrucks-Generator« unmittelbar nach dem Aufrufen aus der Abfrage »FrmAdressenSuchen« der Datenbank Adressen.mdb

- Der Ausdruck soll mit dem Operator *Wie* beginnen. Klicken Sie daher auf die entsprechende Schaltfläche im Dialogfeld *Ausdrucks-Generator.* Daraufhin wird dieser Operator links oben in das Bearbeitungsfeld des Dialogfeldes eingefügt.
- Feld- und Objektnamen werden im mittleren Listenfeld des Dialogfeldes angeboten. Allerdings werden dort stets nur und genau die Namen angeboten, die zu dem im linken Listenfeld markierten Objekt (Formular, Bericht, Tabelle/Abfrage) gehören. Unmittelbar nach dem Aufrufen des Ausdrucks-Generators ist dort das Objekt markiert, von dem aus der Ausdrucks-Generator aufgerufen wurde. In Bild 39.1 ist dies die Abfrage *Frm-*

AdressenSuchen. Um ein Steuerelement aus dem Formular *AdressenSuchen* einzufügen, verfahren Sie, wie im nächsten Aufzählungspunkt beschrieben.

- Doppelklicken Sie im linken Listenfeld des Dialogfeldes *Ausdrucks-Generator* auf das Objekt *Formulare*. Daraufhin verzweigt dieses Objekt in die Objekte *Geladene Formulare* und *Alle Formulare*.
- Doppelklicken Sie im linken Listenfeld des Dialogfeldes auf den Objekteintrag *Alle Formulare*. Dann werden in diesem Listenfeld in der Baumstruktur alle Formulare der aktuellen Datenbank angezeigt.
- Klicken Sie in der Liste aller Formulare auf den Namen des Formulars *AdressenSuchen*. Dann ist dieses Formular markiert, und im mittleren Listenfeld werden alle Steuerelemente dieses Formulars angezeigt. Markieren Sie darin den Namen *Firma*.
- Klicken Sie auf die Schaltfläche *Einfügen*, um den Bezeichner für *Firma* vom Ausdrucks-Generator in den Ausdruck einfügen zu lassen. Dieses Zwischenergebnis ist in Bild 39.2 wiedergegeben. Man erkennt, daß der Ausdrucks-Generator den Teilausdruck `Formulare![AdressenSuchen]![Firma]` hinzugefügt hat.

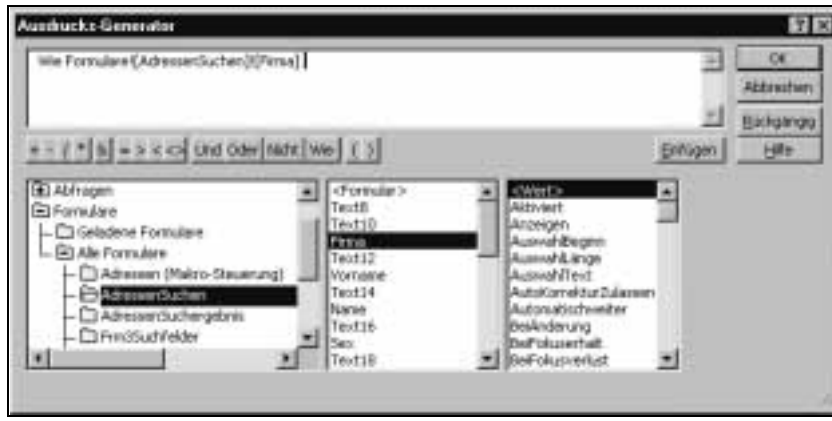


Bild 39.2: Der Ausdrucks-Generator hat den Teilausdruck »Formulare![AdressenSuchen]![Firma]« hinzugefügt.

- Klicken Sie auf die Operator-Schaltfläche mit dem Zeichen *&*, um dem Ausdruck den Verkettungsoperator hinzuzufügen. Alternativ könnten Sie diesen – wie im übrigen jeden anderen Teil des Ausdrucks – auch direkt in das Bearbeitungsfeld des Editors eingeben.
- Nach dem Hinzufügen des letzten Teilausdrucks blinkt der Cursor im Bearbeitungsfeld des Dialogfeldes rechts neben dem bisher erzeugten Ausdruck. Geben Sie dort über die Tastatur die Zeichenkette »*« ein. Das Bearbeitungsfeld im Dialogfeld *Ausdrucks-Generator* sollte jetzt aussehen, wie in Bild 39.3 wiedergegeben.

Wie Formulare![AdressenSuchen]![Firma] & "*"

Bild 39.3: Fertig erstellter Ausdruck im Bearbeitungsfeld des Ausdrucks-Generators.

- Bestätigen Sie die Arbeit des Ausdrucks-Generators mit OK. Daraufhin wird das Dialogfeld *Ausdrucks-Generator* geschlossen, und der erstellte Ausdruck wird an die aufrufende Stelle – hier das Kriterienfeld in der Abfrage – übergeben.

Wie Sie in Bild 39.1 erkennen können, enthält das linke Listenfeld des Dialogfeldes *Ausdrucks-Generator* nicht nur Objekte als Einträge, sondern auch die Kategorien *Funktionen*, *Konstanten*, *Operatoren* und *Gebräuchliche Ausdrücke*. Durch Markieren dieser Kategorien können Sie die entsprechenden Elemente in einen Ausdruck einfügen. Wenn Sie beispielsweise die Kategorie *Funktionen* durch Doppelklicken öffnen, können Sie zwischen eingebauten Funktionen und – falls vorhanden – benutzerdefinierten Funktionen wählen, vgl. Bild 39.4. Dort werden benutzerdefinierte Funktionen unter der Kategorie Adressen angeboten, dem Namen der aktuellen Datenbank. Im mittleren Listenfeld werden Kategorien von Funktionen angeboten, und im rechten die zugehörigen Funktionen. Wenn Sie einem Ausdruck eine Funktion auf diese Weise hinzufügen, werden deren Argumentnamen als Platzhalter für die Argumentwerte zusammen mit der Funktion eingefügt. Sie müssen dann die Argumentnamen durch Werte oder weitere Teilausdrücke (ein Argument kann beispielsweise selbst durch einen Ausdruck angegeben werden) ersetzen.

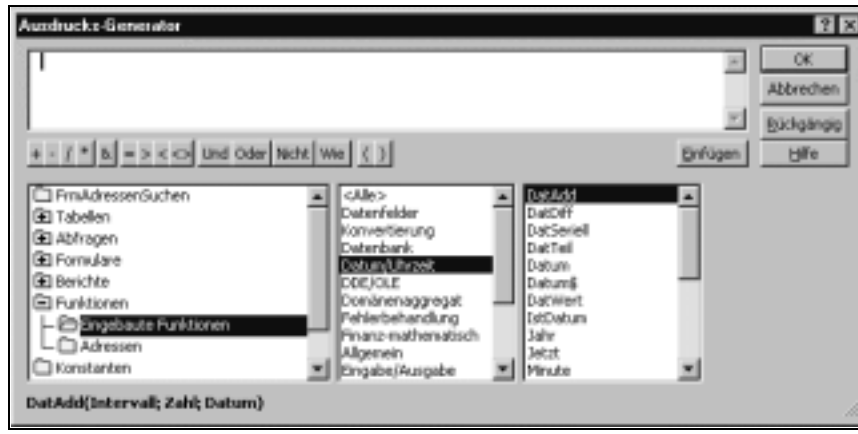


Bild 39.4: Dialogfeld »Ausdrucks-Generator«. In dieser Situation könnte die Funktion »DatAdd« hinzugefügt werden.

39.6 Funktionen und Operatoren

In einer früheren Auflage dieses Buches (MS-Access 2.0 Professionell) wurden an dieser Stelle sämtliche Funktionen und Operatoren, die in Access 2.0 verfügbar waren, wiedergegeben und mit zahlreichen Beispielen erklärt. Wenn dieser Text, der mehr als 100 Buchseiten umfaßte, für die vorliegende Ausgabe aktualisiert worden wäre, hätte sein Umfang noch einmal um ca. 50% zugenommen, weil sehr viele Funktionen hinzugekommen sind. Dies war aus räumlichen Gründen nicht mehr vertretbar. Eine Möglichkeit, einen Teil der Erklärungen in dieses Buch hinüber zu »retten«, hätte darin bestanden, nur eine Auswahl der wichtigsten Funktionen und Operatoren wiederzugeben. Ein derartiges Vorgehen erschien mir jedoch unbefriedigend, weil jede Auswahl sehr subjektiv sein muß. Ich habe daher einen Weg gewählt, Ihnen die verfügbaren Informationen ungekürzt – aber auch nicht aktualisiert – zur Verfügung zu stellen, indem ich sie Ihnen auf der Begleit-CD-ROM anbiete. Dort finden Sie unter dem Dateinamen

FunktionenUndOperatoren.doc

sämtliche Funktionen und Operatoren, wie sie in der angegebenen Auflage des Buches wiedergegeben wurden. Da ich die Zusammenstellung nicht aktualisiert habe, fehlen die seitdem hinzugekommenen Funktionen. An einigen wenigen Stellen mögen sich auch Ergänzungen für bestehende Funktionen ergeben haben. Dieser Mangel wiegt m.E. jedoch nicht sehr stark im Vergleich zu dem Vorteil, den Sie mit den zahlreichen Beispielen und Erklärungen haben.

Kapitel 40

Literaturverzeichnis

Awad, E.M., / Gotterer, M.H., Database Management. Danvers (Mass.) 1992.

Brosius, G., Microsoft OLAP Services. Multidimensionale Datenverwaltung im Microsoft SQL Server 7. Addison-Wesley-Longman, Bonn 1999.

Behme, W. / Ohlendorf, Th., Datenbanksysteme, -modelle und Entwurfsmethoden als Grundlage von Controlling-Informationssystemen. In: Biethan, J. / Huch, B., Informationssysteme für das Controlling, Berlin 1994, S. 117-174.

Chen, P.P.S., The Entity-Relationship Model: Towards a Unified View of Data. In: ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, p. 9-36.

Haught, D. / Ferguson, J., Microsoft Jet Datenbank-Engine Programmierleitfaden. Unterschleißheim 1996.

Jennings, R., Using Access 2000. Ohne Ort 1999.

Sauer, H., Relationale Datenbanken. Theorie und Praxis inklusive SQL-2. Bonn 1992.

Scheer, A.-W., Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse. 6. Aufl. Berlin 1995.

Solomon, Chr., Developing Applications with Microsoft Office95. Second Edition Redmond 1996.

Sommer, M., Datenbankdesign für Access 7.0/2.0. Bonn 1996.

Stahlknecht, P., Einführung in die Wirtschaftsinformatik. 7. Aufl. Berlin 1995.

Wells, E., Lösungen entwickeln mit Excel 95. Leistungsfähige Datenanalyse und Informationssysteme mit Visual Basic für Applikationen und Excel-Objekten. Unterschleißheim 1996.

Index

!
#Name? 186
1:1-Beziehung 173
1:n-Beziehung 66, 173

A
Abfrage 30
Abfragefenster in der Entwurfsansicht 266
Aggregierte Berechnungen 334
aggregierte Berechnungen mit eigenen Ausdrücken 341
aggregierte Berechnungen mit Kriterien 339
Aktions- 262, 353
Aktualisierungs- 353, 360
alle Felder einer Feldliste hinzufügen 280
Anfüge- 353, 358
Anwendungszwecke 262
Anzahl der Datensätze bestimmen 286
-Assistent 40
Ausführungsberechtigungen 269
Auswahlabfrage,Definition 261
Beispiele für Kriterien 298
berechnetes Feld erstellen 285
Berechnung für nach einem Feld gruppierte Datensätze 335
Berechnung für nach mehreren Feldern gruppierte Datensätze 337
Datenaktualisierung bei Ein- und Mehrtabellenabfragen 318
Datenbasis in Mehrtabellen- 135
Datendefinitions- 262
Datensätze bearbeiten 119
Datensätze mit Nullwerten in verknüpften Feldern 307
Datensätze sperren 270
Datensatzgruppentyp 270
-definition 687
Definition drucken 275
Duplikate unterdrücken 302
durch Beispiel 267
Eigenschaften einstellen 267
eindeutige Datensätze 269
Einführungsbeispiel 116
-entwurf 57
-entwurf, Optionen 272
Entwurfsansicht 56
Entwurfsbereich und SQL 349
equi-join 311
-ergebnis betrachten 119
Ergebnis drucken 275
erstellen 116, 266
Feld hinzufügen 279
Felder ausblenden 284
Felder benennen 283
Felder einfügen 282
Felder Hinzufügen 277
Felder löschen 282
Felder mit Tabellennamen anzeigen 283
Felder verschieben 281
Feldliste einer Tabelle oder Abfrage hinzufügen 277
Filter 271
Funktionen 334
Gleichheitsverknüpfung 303, 311
Gleichheitsverknüpfung erstellen 312
in VBA-Prozedur 366
Inklusionsverknüpfung 311
Inklusionsverknüpfung, left join 311
Inklusionsverknüpfung, left join, erstellen 313
Inklusionsverknüpfung, right join 311

Keine Duplikate 268
Kreuztabellen- 343
Kreuztabellen-, Definition 262
Kreuztabellen-, Spaltenüberschriften fixieren 346
Kriterien 58, 290
Kriterien aus Steuerelementen in einem Formular 323
Kriterien für mehrere Felder 292
Kriterien, die Berechnungen enthalten 295
Kriterien, die sich auf Formularfelder beziehen 297
Lösch- 353, 356
Max Datensätze 271
Migration auf andere Datenbanken 366
neue Tabelle erzeugen 353
Nullwerte in den Datensätzen 327
Nullwerte in Kriterienfeldern 292
Nullwerte in verknüpften Feldern 307
ODBC-Wartezeit 271
öffnen 55
Parameter 319, 320, 687
Parameter angeben 319
Parameter in anderen als Kriterienausdrücken 321
Parameterabfrage als Reaktion auf unbekannte Namen 322
Parametereingabe, Reihenfolge festlegen 323
Quelldatenbank 269
Quellverbindung 270
Query by Form 323
Reflexivverknüpfung 315
Sortieren 288
Sortiert nach 271
speichern 120
Spitzenwerte 268
SQL 274
SQL Pass-Through-, Definition 262
SQL-Abfrage 348
SQL-Anweisung bearbeiten 349
SQL-Anweisung kopieren 349
SQL-spezifische 366
Tabellenerstellungs- 353
Union- 350
Union-, Definition 262
verknüpfte Tabellen aufnehmen, ohne ihre Felder anzuzeigen 309
Verknüpfung erstellen und aufheben 303
Verknüpfungen bei Mehrtabellenabfragen 302
Access
beenden 61
starten 53
Access als Client 784
Access als Server 784
Access Objektmodell 675
Access_als Client
Beispiel 791
Access_als Server
Beispiel 794
AccessAlsClient.mdb 791
Access-Projekt
Dialogfeld Datenlinkeigenschaften 882
Eingabeparameter in Formular oder Bericht 898
Grundlagen 857
mehrere Möglichkeiten zum Erstellen 18, 20, 838, 870, 873
mit (anderer) SQL Server-Datenbank verbinden 885
neu erstellen 879
Parameterabfrage als Gespeicherte Prozedur 894
portieren 900
SQL-Select-Anweisungen in Formular oder Bericht 898
vs Access-Datenbank 864
Access-Spezialtasten verwenden 570
ActiveX 784
Administrator 848
Administratorkonto
erstellen 852
ADO
Connection 724, 727
Connection-Objekt 725, 727
ConnectionString 724, 725, 726
Dokumentation 725
Einführungsbeispiel 723
Minimalangaben für den Datenzugriff 723
Provider 725
Recordset 724
Recordset-Objekt 728
ADOMD 730
Aktion 576
Aktualisierung
Datenanzeige 163
der Datenbasis Listen- und Kombinationsfelder 165
Aktualisierungsweitergabe an Detailfeld 102
Kontrollkästchen 100
Alternativschlüssel 755
Anmelden, Dialogfeld 850
Anschriftstring erstellen 287
Anweisung
bedingte 664, 666
nächste auszuführende festlegen 696
testen im Testfenster 651

Anwendungssymbol 569
Anwendungstitel 568
Anzeigeaktualisierung
gelöschter Datensätze 164
neuer Datensätze 164
Anzeigeformat 195, 215
Arbeitsgruppe
anschließen an 838
Berechtigungsprüfung ein- und abschalten 848
gesicherte 846
-Informationsdatei erstellen 839
sich anschließen 841
Argument
Makro 576
Übergabe an Funktion 658
Array-Variable 684
Assistenten
zur Aufgabenvereinfachung 39
Auflistung-Objekt 677
Ausdruck
arithmetische Operatoren 907
Bericht, Systemobjekt 905
Bezeichner 903
Bezeichner in - 905
Bildschirm, Systemobjekt 905
Datums- und Zeitwerte in - 906
Datumsangaben 904
eckige Klammern 904
eingeben 904
Forms, Systemobjekt 905
Formular, Systemobjekt 905
Funktion in Ausdruck 903
-Generator 909
Gleichheitszeichen, wann voranstellen 907
Hauptformular/bericht, Bezug auf
Steuerelemente 906
Komponenten 903
Konstante 903
Literal 903
logische Operatoren 908
Namen in - 904
Operator 903
Operatoren, weitere 908
Platzhalterzeichen 908
Reports, Systemobjekt 905
Screen, Systemobjekt 905
Stellvertreterzeichen 908
testen im Testfenster 652
Texte in - 907
Unterformular/bericht, Bezug auf
Steuerelemente 905
Vergleichsoperatoren 907
Verkettungsoperatoren 908
auswahlausschließender Filter 153
auswahlbasierter Filter 152
AutoBericht 40
AutoExec-Makro 568
Autoexec-Makro 578, 597
AutoFormular 39
AutoKorrektur 130
Automatische QuickInfo 638
Automatisierung mit ActiveX 784
AutoReattachTables 259
AutoWert
Datentyp 90, 135
Felddatentyp 192, 746
Awad, E.M. 915

B

Bearbeitungsfeld, Generator aufrufen 399
bedingte Anweisung 664, 666
bedingte Schleife 670, 672
Befehlsschaltfläche 481
Ereignismakro 611
Behme, W. 915
Beispieldatenbanken 47
Benutzer
einer Gruppe zuordnen 853
löschen 853
benutzerdefinierte Funktion 627, 660
benutzerdefinierte Tastenbelegungen 578
Benutzerkonto 687, 846
erstellen 850
berechnetes Steuerelement 405
Berechtigung
zuweisen und entfernen 853
Berechtigungszuweisungen 846
Bericht 33, 687
Adreßetiketten 545
besondere Leistungsmerkmale 529
Definition drucken 544
doppelte Feldinhalte ausblenden 558
drucken 544
erstellen 532
Fenster Sortieren und Gruppieren 546
Gemeinsamkeiten und Unterschiede mit
Formular 531
Gruppenkopf- und -fuß 543
Gruppenkopf/fuß 548
Gruppieren 530
gruppieren 547
Gruppierungs- mit Berichts-Assistenten
erstellen 533

Gruppierungs- und Sortierreihenfolge
ändern 549
Gruppierungsebenen hinzufügen 547
Gruppierungseinheit und -intervall festlegen 549
Haupt- und Unter- 550
Laufende Summe, Textfeldeigenschaft 557
leeren erstellen und bearbeiten 543
mehrspaltig 545
Sortieren 530
Sortieren und Gruppieren, Fenster 546
Sortierung 546
Spalten, mehrere 530
Unterbericht in gruppiertem Hauptbericht 550
zusammenfassende Funktionen 555
Berichts-Asistent 533
Berichts-Assistent 40
Beschriftung, Feldeigenschaft 205
Bezeichnungsfeld 113, 435
automatisch oder nachträglich einfügen 405
nachträglich einem Steuerelement zuordnen 435
ohne sein Steuerelement verschieben 415
ohne Textfeld verschieben 114
Beziehung
1:1 173
1:n 66
1:n- 173
Access-Projekt, erstellen und bearbeiten 890
definieren 98
m:n- 174
Objekttyp Relation 687
Brosius, G. 915

C

Chen, P.P.S. 915
Client 784
Client/Server-Anwendung 838
Client/Server-Datenbank 857
Code
als Text aus der Zwischenablage einfügen 636
Suchen und Ersetzen 636
-Zeile auf mehrere Zeilen verteilen 636
Code-Fenster 633
Code-Generator 645
COM 485
Const 686
CreateObject 788

D

DAO
AddNew 719
aktuellen Datensatz im Objekt löschen 721
BeginTrans 721
BOF 714
Bookmark 717
CommitTrans 721
Data Access Objects 704
Database 706
Datenmanipulation 713
Datensätze ändern 718
Datensätze blättern 714
Datensätze hinzufügen 719
Datensätze löschen 721
Datensätze neu sortieren 714
Datensätze suchen 716
Datensatz editieren 718
Datensatz, Focus wechseln 714
DB_OPEN_DYNASET 716, 719
dbOpenDynamic 710
dbOpenDynaset 710, 711
dbOpenForwardOnly 710
dbOpenSnapshot 710
dbOpenTable 710, 711
Delete 721
Dynaset, Objekttyp 711
Edit 718
EOF 714
ersten Datensatz suchen 716
Filter ändern 715
FindFirst 716
FindLast 716
Find-Methoden 716
FindNext 716
FindPrevious 716
Lesezeichen verwenden 717
letzten Datensatz suchen 716
Methode Recordset 709
Movefirst 714
MoveLast 714
Move-Methoden 714
MoveNext 714
MovePrevious 714
nächsten Datensatz suchen 716
neuen Datensatz im Objekt hinzufügen 719
-Objektmodell 704
Objekttyp Database zuweisen 706
OpenRecordset 709, 715
QueryDef 706
QueryDef-Objekt 708
Recordset 706, 709
Recordset-Typen 711
Recordset-Variablen 714
Rollback 721
Snapshot, Objekttyp 711
TableDef 706

TableDef-Objekt 707
Transaktion 721
Update 718, 719
vorherigen Datensatz suchen 716
DAP 807
Data Access Page 807
Database Management 915
DateAdd 643
Daten
Anzahl der Datenaktualisierungsversuche 843
Anzeige aktualisieren 163
automatisch speichern 38
Eingabe erzwingen 201
ersetzen 149
in Feldern kopieren und verschieben 144
in Feldern löschen 147
Intervall für Anzeigeaktualisierung 843
Intervall für Datenaktualisierung 843
Intervall für ODBC-Anzeigeaktualisierung 843
rechteckigen Feldbereich kopieren oder verschieben 144
suchen 148
Datenbank 687
-design 65, 167
-design für Access 915
entwerfen 732
-entwurf 65
Kennwort 847
relationale 29, 66
verschlüsseln 856
Datenbankdiagramm 870
erstellen 886
formatieren 887
Datenbankfenster 48
Datenbanksysteme, -modelle und Entwurfsmethoden 915
Datenblatt
individuell gestalten 137
Optionen 137
Datenblattansicht 55, 83
Datendefinitionssprache 366
Datenereignisse 605
Datenfeld 684
dynamisches 686
Datenfeldvariable
dynamisch 686
Datenmanipulationssprache 366
Datenmodell
Anforderungsanalyse 733
ANSI/SPARC-Modell 735
atomare Werte 741
Attribut 738
berechnetes Feld 742, 768
Beziehung 738
Datenbankkonzept dokumentieren 779
Dokumentierer 779
Drei-Schichten-Modell 736
Entität 738
Entity-Relationship-Modell 734, 737
ERM 737
externe Sicht 734, 736
interne Sicht 734, 735
konzeptionelle Sicht 734, 736
konzeptionelles Modell 734
Relationenmodell 737
semantisches 734
Tabellen bestimmen 739
Tabellenstruktur in visualisierter Form präsentieren 740
Unabhängigkeit der logischen von der physischen Datenstruktur 734
Datenmodellierung 731
Datensatz
Änderungen rückgängig machen 133
automatisch speichern lassen 86
blättern 54
erster 55
filtern 120, 149
gesperrter 135
-gruppe 687
hinzufügen 124
jede Zeile 30
kopieren und verschieben 146
letzter 54
löschen 86, 147
markieren 86
-markierer 105
sortieren 149
speichern 136
Sperrung 842
suchen mit Makro 619
verwaist 101
vorheriger 55
Datentyp
konvertieren in einen anderen 193
prüfen 134
Tabellenfeld 70
Datenzugriffsseite 807
ADO-ConnectionString 810
ADO-Recordset 811
Aktualisierbarkeit von Daten 834
Beispiele in Nordwind-Datenbank 808

Diagramm einfügen 830
erstellen mit PivotTable und Diagramm 826
Gruppierungsebene erstellen 821
HTML-Code 810
in Entwurfsansicht bearbeiten 819
mit Assistent erzeugen 815
PivotTable 813
Sortieren und Gruppieren 820
speichern 821
Steuerelement PivotTable einfügen und bearbeiten 827
Stilvorlagen 817
Datum
Beispiel 79
Funktion 79
Konzept 79
Datum/Uhrzeit
Datentyp 70
Felddatentyp 191, 746
Datumsangaben in Access-SQL-Anweisungen 384
DBEngine 705
DBMS 29
DDL 366
Debug.Print 652
DELETE-Anweisung 367, 372
Detailbereich 429
Detailtabelle 99, 101
Dezimalstellen
Eigenschaft 196
Diagramm 465
mit der Toolbox erstellen 466
Tool 465
Dialogfeld Zoom 51, 398
Dialogfeldformular 505
Dim 681
DML 366
DM-Zeichen 215
Do ... Loop 671
Domäne eines Tabellenfeldes 743
Druckereignisse 608
Duplikate
ausblenden in Bericht 558

E

early binding 791
eckige Klammern 904
Eigenschaft
Hilfe 423
VBA 675
Eigenschaftenfenster 634
Eigentümer 846
ändern 855
Eingabe
Daten in Formular 104
Daten in Tabelle 102
lange Zeichenfolgen 51
rückgängig machen 86
unerwartete Schwierigkeiten 88
Eingabe erforderlich 71, 76, 77, 78, 134, 201
Verstoß dagegen 88
Eingabeformat 72, 134, 196, 750
-Assistent 196, 198
Kennwort 196
Verstoß dagegen 87
Eingebaute Symbolleisten zulassen 570
End Function 642
Entity-Relationship Model 915
Entscheidungsstrukturen 664
Entwerfen einer Datenbank 732
Entwurfsansicht
Abfrage 56
Formular 112
Ereignis 601
BeiÄnderung 606
BeiAktivierung 604
BeiDeaktivierung 604
BeiEntladen 604, 613
BeiFehler 609
BeiFokuserhalt 604
BeiFokusverlust 604
BeiGrößenänderung 604
BeiLaden 603
BeimAnzeigen 606, 623
BeiMausbewegung 607
BeiMaustasteAb 607
BeiMaustasteAuf 607
BeimDoppelklicken 607
BeimDrucken 608
BeimFormatieren 608
BeimHingehen 604
BeimKlicken 606, 619
BeimLöschen 605
BeimÖffnen 603
BeimSchließen 603
BeimVerlassen 604
BeiNichtInListe 606
BeiOhneDaten 608
BeiOLEAktualisierung 606
BeiRücknahme 608
BeiSeite 608
BeiTaste 607
BeiTasteAb 607
BeiTasteAuf 607
BeiZeitgeber 609
Ereignisse nach Kategorien 603
Fenster- 603
NachAktualisierung 605, 616
NachEingabe 605

NachLöschbestätigung 605
ÖffnenFormular 613
VorAktualisierung 605, 615
VorEingabe 605
VorLöschbestätigung 605
Ereignismakro 593, 602
Befehlsschaltfläche 611
in Formularen und Steuerelementen 611
Organisation 426
Ereignismodul 428
ereignisorientierte Eigenschaften 389
Ereignisprozedur 603
erstellen 643
Organisation 427
Vergleich zu gewöhnlicher Prozedur 428
Ersatzschlüssel 752
Exit Do 671
Exit For 668
Exit Function 662
Exit Sub 662
Exportieren von Tabellen 252

F

Fehler- und Zeitgeberereignisse 609
Fehlerbehandlung
Access Basic-Laufzeitfehlern 700
allgemeine Access-Ebene 697
Begriff 696
BeiFehler 697
DataErr 698
Fehlercodes 697
GoTo 700
Laufzeitfehler 700
On Error 700
Response 698
Resume Next 700
-sroutine 697, 700, 701
Fehlersuche 694
Feld 687
berechnetes 135
Datentyp festlegen 185, 189
Datentypen 70
Datentypen festlegen 74
Domäne 743
Eigenschaft Format 195
Eigenschaften 70
Eigenschaften festlegen 75, 185, 193
Focusbesitz 75
-Generator 186
gesperrtes oder nicht aktiviertes 135
Größe 71, 76, 77, 78
-größe 195
-größe für ein Zahlenfeld 195
Gültigkeitsregel 203
hinzufügen 185
in schreibgeschütztem Formular 135
kopieren 188
löschen 187
Name 69
Namen ändern 185
Namen festlegen 73, 185
nicht bearbeitbar 135
verschieben 189
Fenstermodus 613
Ferguson, J. 915
Filter
aufheben 121
auswahlausschließender 153
auswahlbasierter 152
Formularbasierter 154
Kriterienausdrücke 153
löschen 161
speichern 159
Spezialfilter/-sortierung 156
Filter/Sortierung
anwenden 158
Fokusereignisse 604
For ... Next 668
For Each ... Next 669
Format 20
-abschnitt 216
Anzeigeformate 215
benutzerdefiniert 215
Datum lang, mittel, kurz 217
Datum/Uhrzeit 217
Dezimalstellen 216
Feldeigenschaft 195
Ja/Nein 219
Standarddatum 217
Standardformate für Datum/Uhrzeit 217
Standardformate für Ja/Nein 219
Standardformate für Zahlen und Währung 220
Symbole 217
Symbole für Datum/Uhrzeit 218
Symbole für Text 220
Tausenderteilungspunkt 216
Text- 216
Text- und Memo 220
Zahlen- und Währung 220
Zahlen, Datums- und Zeitangaben, Texte 215
Zeit lang, mittel, kurz 218
Formular 687
-ansicht 394
-ansichten 393

Ausdrücke in 511
-basierter Filter 154
berechnete Werte in Tabelle speichern 516
Bereich einfügen und entfernen 429
Daten bearbeiten 115
Datenblattansicht 394
Datenherkunft, Formulareigenschaft 390
Definition drucken 431
Detailbereich 429
Dialog, Fenstermodus 507
Eigenschaftenfenster 396, 397
Entwurfsansicht 393
Ereignisprozedur 629
Ereignisprozedur und -makro 426
erstellen 103, 390
Feldliste 396, 400
für Druck einrichten 430
Gebunden, Eigenschaft 505
Gruppensumme im Haupt- oder Unterformular 523
importieren 392
-kopf und -fuß 429
kopieren 392
Lineal 402
mehrere synchronisieren 621
mit Formular-Assistent erstellen 390
n:1- erstellen 500
ohne Formular-Assistenten erstellen 391
Popup- 505
PopUp, Eigenschaft 505
Raster 403
Schließen verhindern 613
Seitenansicht 394
Seitenkopf und -fuß 429
Seitenränder und Ausgabeformat 430
Seitenzahl beim Drucken ausgeben 522
Sichtbar, Eigenschaft 507
Standardansicht 395
Toolbox 396, 401
Unterformular erstellen 105
Unterformular positionieren 114
Visible, Eigenschaft 507
-vorlage definieren 425
Formular-Assistent 39, 103
Fremdschlüssel 99, 173, 753
und Primärschlüssel, Datentypen 744
FROM-Klausel 377
Function 642
Funktion
benutzerdefiniert 627, 660
die einen Wert ausgibt 658
Grundlagen 657
mit Argumenten 658
Sprachvereinbarung 626
testen im Testfenster 651
zulässige Namen 657
FunktionenUndOperatoren.doc 913

G

gebundenes Steuerelement 404
Generator für Bearbeitungsfeld 399
Gespeicherte Prozedur 869
erstellen und bearbeiten 894
GetObject 790
Gotterer, M.H. 915
Graph 465
GROUP BY-Klausel 377
Gruppenkonto 687
erstellen 852
Gültigkeitsmakro 135
Gültigkeitsmeldung 71, 78, 205
Gültigkeitsprüfung
mit Makro 614
Gültigkeitsregel 71, 77
ändern 204
Datensatz 749
Feld 748
Formular 105
für Tabelle 223
für Tabelle »Projekte« 95
prüfen 134
Tabelle vs Formular 203
Verstoß dagegen 87

H

Haltepunkt
Modulfenster 694
setzen und löschen 695
Haught, D. 915
HAVING-Klausel 378
Hyperlink
-Adresse 131, 798
Adresse bearbeiten 133
Adresse einfügen 131
-Adresse eingeben 803
ausführen 800
-Basis für eine Datenbank festlegen 803
bearbeiten 803
Bedeutung 798
HyperlinkAddress 798
HyperlinkSubAddress 798
Steuerelement in Formular oder Bericht erstellen 800

I
If_..._Then_..._Else 664
Importieren von Tabellen 244
IN 380
In
  Beispiel 78
Index
  erstellen 228
Initialisieren VBA-Ausführung 695
INSERT INTO-Anweisung 367, 370
Integrität
  referentielle 101
IsLoaded 623

J
Ja/Nein
  Felddatentyp 192, 746
  Steuerelement 436
Jennings, R. 915
JetEngine 687
JOIN-Operationen 373

K
Kennwort
  ändern 850
  ändern und löschen 854
  Eingabeformat 196
  löschen 853
Klasse
  Nicht-Standardinstanzen 692
Klassenbibliothek
  Verweis auf 786
Klassenmodul 690
  Beispiel 691
  eines Formulars oder Berichts 692
Kombinationsfeld 443
  dreispaltiges ungebundenes, Beispiel 449
Kommentar in Visual Basic-Code einfügen 637
Konstante
  symbolische 686
Konten
  verwalten 852
Kontextmenü
  erstellen 561
  verwalten 559
Konto 846
Kontrollkästchen 436
Kriterien
  Abfrage 58

L
late binding 791
Laufzeitfehler 694
Leere Zeichenfolge
  Eigenschaft 201
leere Zeichenfolge
  Vergleich mit Nullwert 201
Linie 480
Listen- und Kombinationsfeld
  Aktualisierung der Datenbasis 165
  Datenbasis aktualisieren 451
  Datensatzherkunft, Eigenschaft 444
  Herkunftstyp, Eigenschaft 444
  mehrspaltig 445
  mit dem Steuerelement-Assistenten erstellen 447
  ohne den Steuerelement-Assistenten erstellen 447
  Spalten ausblenden 446
  Spaltenbreiten einstellen 446
  unangenehme Überraschungen 451
  Unterschiede 445
Listenfeld 443
  mit Wertliste, Beispiel 448
Löschweitergabe an Detaildatensatz 101
  Kontrollkästchen 100
Loop 671

M
m:n-Beziehung 174
Makro 34
  AbbrechenEreignis 614
  AbbrechenEreignis, Aktion 594
  -aktion 576
  Aktionsargumente 585
  AktualisierenDaten, Aktion 594
  AktualisierenObjekt, Aktion 594
  AnwendenFilter 618
  AnwendenFilter, Aktion 594
  Anwendungszwecke 577
  AnzeigenAlleDatensätze 618
  AnzeigenAlleDatensätze, Aktion 594
  -argument 576
  aus -gruppe ausführen 593
  ausführen 590
  AusführenAnwendung, Aktion 594
  AusführenCode, Aktion 594
  AusführenMakro, Aktion 594
  AusführenMenübefehl, Aktion 594
  AusführenSQL, Aktion 594
  AusgabeIn, Aktion 595

AuswählenObjekt, Aktion 595
Autoexec 578, 597
bearbeiten 585
bedingte Ausführung von Aktionen 586
Bedingung 580, 586
Bezugnahme auf Steuerelemente und Felder 589
Datensätze filtern mit - 617
Dialogfeld Einzelschritt 591
Drucken, Aktion 595
Echo, Aktion 595
EinblendenSymbolleiste, Aktion 595
Einzelschritt im Entwurfsfenster 591
Ereignis- 593
erstellen 581
-fenster 579
-fenster, Spalte Bedingung 580
-fenster, zusätzliche Spalten 580
GeheZuDatensatz, Aktion 595
GeheZuSeite, Aktion 595
GeheZuSteuerelement 620
GeheZuSteuerelement, Aktion 595
-gruppe erstellen 592
HinzufügenMenü, Aktion 595
kopieren 600
KopierenObjekt, Aktion 595
LöschenObjekt 595
Maximieren, Aktion 595
Meldung, Aktion 595
Minimieren, Aktion 595
-name 576
-name und -gruppe 592
-name, Spalte in Makrofenster 580
ÖffnenAbfrage, Aktion 595
ÖffnenBericht, Aktion 595
ÖffnenFormular, Aktion 595
ÖffnenModul, Aktion 596
ÖffnenTabelle, Aktion 596
Positionieren, Aktion 596
sämtliche Makroaktionen 594
Sanduhr, Aktion 596
Schließen, Aktion 596
SendenObjekt, Aktion 596
SetzenMenüelement 596
SetzenWert 615
SetzenWert, Aktion 596
Signalton, Aktion 596
Speichern 596
StopAlleMakros, Aktion 596
StopMakro, Aktion 596
SuchenDatensatz 620
SuchenDatensatz, Aktion 597
SuchenWeiter, Aktion 597
Symbolleiste des -fensters 580
Tastaturbefehle, Aktion 597
TransferArbeitsblatt, Aktion 597
TransferDatenbank, Aktion 597
TransferText, Aktion 597
UmbenennenObjekt, Aktion 597
und Steuerelement 577
Verlassen 597
Vorteil gegenüber VBA-Prozedur 628
Warnmeldungen, Aktion 597
Wiederherstellen, Aktion 597
Mastertabelle 99, 101
Mausereignisse 606
Me
Schlüsselwort in Formular- und Berichtsmodulen 678
Mehrbenutzeroptionen 841
Meldung 614
Memo
Datentyp 70
Felddatentyp 190, 745
Menü- oder Symbolleiste erstellen 560
Menüleiste
verwalten 559
Methode
VBA 675
Microsoft Graph 465
Microsoft Jet Datenbank-Engine 915
Mit referentieller Integrität
Kontrollkästchen 99
Modul 35
als Textdatei speichern oder laden 640
Deklarationsbereich 631
Formular- und Berichts- 629
globales 630
Klassen- 630
Optionen festlegen 639
MSDE
grundlegende Merkmale 859
installieren 860
Programmordner 860
Unterschied zu SQL Server 7.0 859
multidimensional strukturierte Daten 730

N

Nachschlagefeld 750
-Assistent 206
Begriff 206
erstellen 206
selbständig erstellen 211
Navigationsschaltflächen 50
Neu initialisieren VBA-Ausführung 695

New 791
Nicht-Standardinstanzen einer Klasse 692
NordwindCS.adp
  Beispielprojekt 865
  erzeugen 865
  vs Nordwind.mdb 866
Normalform 761
  dritte 766
  erste 762
  zweite 765
Null
  Wert 88
NULL, Zulässigkeit in Feld 747
Nullwert
  Vergleich mit leerer Zeichenfolge 201
Nullwerte in den Datensätzen bei Abfrage 327

O

Objekt
  Auflistung 677
  Begriff 37
  Container- 676
  -katalog 679
  kopieren 37
  mit Informationen zu anderen Objekten 687
  mit kürzerem Code identifizieren 678
  -Variablen verwenden 678
  VBA 675
  vordefiniert und benutzerdefiniert 676
Objektfeld 475
  erstellen 476
  gebundenes 475
  gebundenes erstellen 476
  ungebundenes 475
Objektkatalog 635
Objekttyp
  Container 687
  Control 687, 689
  Database 687, 688
  Document 687
  Field 687
  Form 687
  Form und Report 688
  Group 687
  Index 687
  Parameter 687
  Property 687
  QueryDef 687
  Recordset 687
  Relation 687
  Report 687
  TableDef 687
  User 687
  Workspace 687
Objektvariable 687
  Typ Control 689
  Typ Database 688
  Typ Form 688
  Typ Report 688
Ohlendorf, Th. 915
OLAP 730, 915
OLE 485
OLE DB-Provider 725
OLE/DDE-Timeout 843
OLE-Client 783
OLE-Objekt 465
  Felddatentyp 192
  Verknüpfung aufheben 478
OLE-Objekt, Felddatentyp 746
OLE-Server 783
Online-Hilfe 47
Optionen für Access 571
Optionsfeld 436, 438
Optionsgruppe 437
  mit dem Steuerelement-Assistenten erstellen 438
  ohne den Steuerelement-Assistenten erstellen 442
  Steuerung von Aktionsfolgen 443
Optionsgruppenfeld 438
Optionswert 438
ORDER BY-Klausel 381

P

PARAMETERS-Deklaration 376
Platzhalterzeichen 148
Primärschlüssel 99
  ändern 227
  Bedeutung 752
  bestimmen 751
  -feld 172
  löschen 229
  Mehrdeutigkeiten ohne 172
  setzen 81, 227
  Symbol-Schaltfläche 81
  und Fremdschlüssel, Datentypen 744
  zusammengesetzter oder künstlicher
    Ersatzschlüssel 752
Projekt 629
Projekte.mdb 63
Projekt-Explorer 633
Property Get-Prozedur 690
Property Let-Prozedur 690
Property Set-Prozedur 690
Prozedur 626
  anzeigen lassen 647

bearbeiten 647
-definition im Objektkatalog wählen 636
Einzelschritt 695
erstellen 641
gezielt auffinden 636
nächste 648
Privat 662
Prozedurschritt 695
Sprachvereinbarung 626
Sprung zur -definition, 636
Sub- testen 650
Sub- vs Function- 656
Sub-, Grundlagen 661
suchen 649
unterbrechen 694
vorherige 648
Vorteile 627
vorzeitiges Beenden 662

Q
Query by Example 267
QuickInfo 638

R
Rechteck 480
Rechtschreibprüfung 130
Redundanz 65, 757
Änderungsanomalien 758
Bedeutung 757
beseitigen 761
Einfügeanomalien 759
frei von 29
-freiheit 178
Gründe für Zulassen 770
Löschanomalien 760
Nachteile 758
Widersprüche 168
referentielle Integrität 101
Verstoß dagegen 102
Register
Registerkarte hinzufügen 462
Reihenfolge der Registerkarten ändern 462
Steuerelement 460
Reihenfolge der Steuerelemente ändern 483
Replikation der Datenbank 838

S
Sauer, H. 915
Schaltflächen-Editor 564
Scheer, A.-W. 915
Schleife
bedingt 670, 672
unbedingt 668
Schließen eines Formulars verhindern 613
Schriftart
ändern 140
Seitenkopf und -fuß 429
Seitenumbruch 480
Sekundärschlüssel 755
SELECT 367
Select Case ... End Select 666
SELECT...INTO 367
SELECT...INTO-Anweisung 369
SELECT-Anweisung 368
Server 784
Sicherungsinformationen, Speicherung in Datenbank oder SYSTEM.MDA 847
Sicherungskonzept von Access 845
Sicht 868
Entwurfsansicht 869
erstellen und bearbeiten 891
Solomon, Chr. 915
Sommer, M. 915
Sortieren 162
Spalte
aus- und einblenden 139
Breite optimieren 138
fixieren 139
verschieben 138
Spezialfilter/-sortierung 156
SQL
Begriff 274
Datumsangaben 384
Unterabfragen 383
SQL als Datenquelle in Formularen, Berichten und Steuerelementen 365
SQL Server
Clientkonfiguration 864
Data Transformation Service 863
Datenbank erstellen 880, 885
Datenbank löschen 901
Dienst-Manager 863
DTS 863
Netzwerkkonfiguration 864
Sicherungskopie von Datenbank 899
SQL-SELECT-Anweisung in Formularen und Berichten 350
Stahlknecht, P. 915
Standardformat 215
Standard-Kontextmenüs zugelassen 570
Standardwährung 215
Standardwert 72, 79
Eigenschaft 200
Starten von Access mit Befehlszeilenoptionen 568, 572
Startoptionen 567

Startoptionen für eine bestimmte Datenbank 568
Static 684
Statusleiste anzeigen 569
Stellvertreterzeichen 148
Steuerelement
Assistent 408
Ausdrucks-Generator 511
Ausdrucks-Syntax 513
Ausdrücke, Datumswerte ermitteln und anzeigen 519
Ausdrücke, mathematische Berechnungen 517
Ausdrücke, Texte verketten 520
Ausdrücke, Werte aus einer anderen Tabelle oder Abfrage ermitteln 527
berechnetes 405
durch Formular-Assistent eingefügt 405
durch Ziehen aus der Feldliste einfügen 406
Eigenschaften einstellen 421
einfügen 405
Einführung und Überblick 404
Farbe ändern 417
gebundenes 404
Größe ändern 416
horizontalen oder vertikalen Abstand gleichmäßig ausrichten 419
in Formular oder Bericht 687
-inhalt, Eigenschaft 404
Ja/Nein 436
kopieren 410
löschen 415
manuell am Raster ausrichten 417
markieren und bearbeiten 411
mehrere gleichen Typs erstellen 407
mehrere untereinander ausrichten 418
mit Toolbox einfügen 407
nachträglich ans Raster anpassen lassen 417
nur horizontal oder nur vertikal verschieben 416
ohne sein Bezeichnungsfeld verschieben 415
Reihenfolge ändern 483
Schriftbild ändern 416
Standardeigenschaften mit dem Eigenschaftenfenster einstellen 423
ungebunden 404
verschieben 415
Wert zuweisen mit Makro 615
Steuerelement-Assistent 42
Structured Query Language 274
Sub-Prozedur
Sprachvereinbarung 626
Suchen mit Formular 330
Suchen und Ersetzen von Code 636
Symbolische Konstanten 686
Symbolleisten
verwalten 559
Symbolleisten- und Menüänderungen zulassen 570
Synchronisieren von Formularen 621

T

Tabelle 30
1-Beziehung 773
n-Beziehung 772
Access-Projekt 866
Access-Projekt, erstellen und bearbeiten 888
Aktualisierungsweitergabe an Detailfeld 234, 776
Anzahl der Datensätze bestimmen 286
benötigte Felder identifizieren 741
Beziehung 173
Beziehung bearbeiten 235
Beziehung definieren 98
Beziehung festlegen 230
Beziehung löschen 235
Beziehungen anzeigen im Fenster 235
Beziehungen bestimmen 771
Beziehungen zwischen den Tabellen 740
Daten drucken 238
Definitionen drucken 239
Design 68
Detail- 99, 101
Eigenschaft Beschreibung 224
Eigenschaften 92
Eigenschaften festlegen 223
Einfeld-Index 229
exportieren 241
Fenster Beziehungen 235
Gültigkeitsregel 91
importieren 241
Index 687
Index erstellen 228
Indizes bestimmen 751
Indiziert, Eigenschaft 229
Kardinalität 778
Löschen mit generierten Nullwerten 776
Löschweitergabe an Detaildatensatz 234, 776
m:n-Beziehung 772
Master- 99, 101
Maximalkardinalität 778
Maximalkardinalität der Detailtabelle 778
Maximalkardinalität der Mastertabelle 778
Mehrfelder-Index 229
Mengenverhältnisse von Master- und Detailtabelle 772, 775, 777
Minimalkardinalität 778
Minimalkardinalität der Detailtabelle 778

Minimalkardinalität der Mastertabelle 778
öffnen 59
referentielle Integrität 774
sachlich zu breit 169
speichern 82
SQL Server 866
Tabellenverknüpfungs-Manager 259
verknüpfen 253
verknüpfte 170
verknüpfte umbenennen 257
verknüpfte, Bearbeitungsgeschwindigkeit erhöhen 260
verknüpfte, Eigenschaften ändern 256
verknüpfte, Pfad 258
Verknüpfung aufheben 257
Verknüpfungsinformation ändern 258
Verknüpfungspfad automatisch mit VBA-Prozedur ändern 259
Vor- und Nachteile weiterer Indizes 756
TABELLE.Beziehungen anzeigen 236
Tabelle.Beziehungen zwischen verknüpften Tabellen festlegen und bearbeiten 237
Tabellen-Assistent 39, 179
Tag
Beispiel 79
Tastaturereignisse 607
Tastenkombination, benutzerdefiniert 578
Text
Datentyp 70
Felddatentyp 190
Text, Felddatentyp 744
Texte verketten 520
Textfeld 113, 433
berechnetes 434
für längere Texte 434
mit Bezeichnungsfeld verschieben 113
ohne Bezeichnungsfeld verschieben 114
Vergrößerbar, Eigenschaft 434
Verkleinerbar, Eigenschaft 434
Wiedergabe von Zahlen oder Datumswerten 434
TRANSFORM 368
TRANSFORM-Anweisung 372
Trigger
erstellen und bearbeiten 895
TypeOf 665

U

Umschaltfläche 436
mit Bild versehen 436
unbedingte Schleife 668
Unbeschränkte Menüs anzeigen 569
ungebundenes Steuerelement 404
UNION-Operation 375
Unterabfragen, SQL 383
Unterbericht
verknüpfter in gruppiertem Hauptbericht 550
Unterbrechen einer Prozedur 694
Unterbrechungsmodus 694
Unterformular 457, 493
Ansichten 496
automatisch und manuell verknüpfen 495
durch Ziehen aus dem Datenbankfenster einfügen 494
Funktionsweise 458
Hauptformular mit zwei verschachtelten 498
mit der Toolbox erstellen 459
Verfahren zum Erstellen 493
Verknüpfen nach, Eigenschaft 458
Verknüpfen von, Eigenschaft 458
Verknüpfung mit Hauptformular 458
Until 671
UPDATE 367
UPDATE-Anweisung 371
Upsizing-Assistent 873
Vorteile vs Nachteile 879

V

Variable
Array 684
Datenfeld- 684
deklarieren 680
explizit deklarieren 681
explizite Deklaration 681
Gültigkeitsbereich 683
implizite Deklaration 680
implizite Deklaration verhindern 682
Lebensdauer 684
Namen 680
VBA 625
VBA-Funktion 626
VBA-Projekt 629
VBA-Sprachverzeichnis 47
VBA-Sub-Prozedur 626
Verknüpfen von Tabellen 253
verknüpfte Tabellen 170
Verknüpfungen bei Mehrtabellenabfragen 302
Verschlüsseln der Datenbank 856
verwaister Datensatz 101
Verweis auf Klassenbibliothek 786
View 891
Visual Basic-Editor 631
Optionen 639

W

Währung
  Felddatentyp 191
Währung, Felddatentyp 746
Währungsformate 220
Wells, E. 915
WHERE-Klausel 382
While 671
While ... Wend 671
While_..._Wend 672
Windows-Registrierung 786
Wirtschaftsinformatik 915
WordAlsClient.dot 794

Z

Zahl
  Felddatentyp 190, 745
Zahlenformat 216
Zahlenformate 220
Zeile
  Höhe ändern 141
Zeilenschaltung
  einfügen 97
Zoom
  Dialogfeld 51
Zoom, Dialogfeld 398