

Das
Standard-
werk

6. Auflage

Keine Angst vor Microsoft Access!

Datenbanken verstehen,
entwerfen und entwickeln
Für Access 2007 bis 2019



Andreas Stern



O'REILLY®

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren O'Reilly-Büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei oreilly.plus⁺:

www.oreilly.plus

Andreas Stern

Keine Angst vor Microsoft Access!

Datenbanken verstehen, entwerfen und entwickeln
– Für Access 2007 bis 2019

6. Auflage

O'REILLY®

Andreas Stern

Lektorat: Ariane Hesse

Korrektorat: Sibylle Feldman, www.richtiger-text.de

Satz: Haselier IT Services Übach-Palenberg

Herstellung: Stefanie Weidner

Umschlaggestaltung: Michael Oréal, www.oreal.de unter Verwendung eines Fotos von © iStock by Getty Images, jacoblund

Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-116-5

PDF 978-3-96010-333-2

epub 978-3-96010-334-9

mobi 978-3-96010-335-6

6., aktualisierte Auflage

Copyright © 2019 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: komentar@oreilly.de



Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Inhaltsverzeichnis

Ein Leserbrief	13
Einleitung	15
Warum ist Access anders?	15
Was habe ich mit Ihnen vor?	15
Wer sollte dieses Buch kaufen?	17
1 Ein Blick hinter die Kulissen	21
In diesem Kapitel	22
Technische Voraussetzungen	22
Ist Access eine Datenbank?	23
Access ohne Access?	26
Die Access-Anwendung	27
Formulare benutzen	29
Das Formular frmMitglieder	30
Das Formular frmMannschaften.....	36
Das Formular frmTraining	38
Das Formular frmTypen	39
Hilfe!	40
Benutzerfreundlichkeit	41
Tabellen, Felder und Primärschlüssel	42
Beziehungen und Fremdschlüssel	50
Datenmodellierung muss sein!	53
Formulare entwerfen	54
Eigenschaften von Formularelementen	55
Datensatzquelle	56
Unterformulare	58
Datensatzherkunft und Steuerelementinhalt	61
VBA – ganz kurz	65
VBA-Prozeduren	65
Schaltflächen starten VBA-Prozeduren	67
Was ist wichtig?	68

2 Datenmodellierung	69
In diesem Kapitel	70
Ein Unternehmensdatenmodell	70
... und seine Erstellung	74
Überall Bemerkungen	76
Preise – wofür?	76
Kunden ohne Aufträge?	77
Kein Material und keine Leute?	77
Keine Fremdschlüssel?	78
Fragen, fragen, fragen!	79
Anschritt1, Anschrift2	80
Typ	81
Status	82
Material vs. Materialart	82
Mengen und Zeiten in Zwischentabellen	83
Die strukturierte Szenario-Beschreibung	86
Die Überführung in das physische Modell	87
Das Vereinsmodell	89
Das logische Datenmodell	90
Das physische Datenmodell	91
Das Privatmodell	92
Das logische Datenmodell	93
Das physische Datenmodell	95
Hierarchische Ordnungen	97
Modellbesonderheiten und -erweiterungen	99
1:1-Beziehungen	99
Dreifachbeziehungen	102
Rollen in Beziehungen	104
Typ, Art, Status	105
Funktionshinterlegung	106
Listen	107
Benutzerverwaltung	108
Flexible Objekteigenschaften	110
Wie alles zusammenhängt	111
Modellierungswerkzeuge	111
PowerPoint	112
Visio	117
»Richtige« Tools	118
Interviews sind schwierig	119
ADaMo	120
Was ist wichtig?	120
3 Erste Formulare	123
In diesem Kapitel	124
Tabellen anlegen	124
Beziehungen definieren	128
Referentielle Integrität	128
Beziehung einer Tabelle zu sich selbst	129
Kardinalitäten	130
Löschweitergabe	130

Versionen	131
Testdaten eingeben	132
Auf die Reihenfolge achten	132
Aus Fehlern lernen	133
Nachschlagfelder	134
Ein einfaches Stammdatenformular	134
Schritt 1: Mit dem Formular-Assistenten Formulare erzeugen	135
Schritt 2: Text- und Bezeichnungsfelder anordnen	136
Schritt 3: Listenfeld hinzufügen	138
Schritt 4: Schaltflächen hinzufügen	140
Schritt 5: Standardbedienelemente entfernen	142
Schritt 6: VBA-Code ergänzen	143
Schritt 7: Layout gestalten	145
Ein Formular mit Unterformular	146
Schritt 1: Mit dem Formular-Assistenten Formulare erzeugen	146
Schritt 2: Text- und Bezeichnungsfelder anordnen	148
Schritt 3: Listenfeld hinzufügen	149
Schritt 4: Schaltflächen hinzufügen	150
Schritt 5: Standardbedienelemente entfernen	151
Schritt 6: VBA-Code ergänzen	151
Schritt 7: Layout gestalten	151
Einzelzuordnung	152
Schritte 1 bis 3	153
Schritt 4: Aktive Bedienelemente hinzufügen	154
Schritte 5 bis 7	155
Die beiden Grundbausteine	157
Das Startformular	159
Was ist wichtig?	160
4 Daten für die Datenbank	163
In diesem Kapitel	164
Die Ausgangssituation	164
Datenarten	166
Zahlen	166
Datumsangaben	167
Texte	167
Excel-Funktionen zur Textbearbeitung	169
Generierung von Testdaten	171
Erzeugung der Daten in Excel	171
Import der Daten in Access	173
Übernahme von Echtdaten	176
Datenorganisation: Datenmenge	177
Formeln eingeben	178
Datenqualität	179
Noch einmal: Datenorganisation	184
Was ist wichtig?	190

5 VBA – Grundlagen	191
In diesem Kapitel	192
Programmieren – muss das sein?	192
Ein Beispiel	192
»Hackermentalität«	193
Fehler finden und korrigieren	194
Das VBA-Fenster	194
Syntaxfehler	196
Laufzeitfehler	198
Logische Fehler	201
Die Entwicklungsumgebung	202
Der Editor	203
Objekte und Ereignisse	205
Der Debugger	207
Symbolleiste anpassen	208
Programmierbefehle	209
Das Drumherum	209
Hallo Welt!	212
If-Then-Else	212
For-Next	216
Do-While-Loop	217
Select-Case	219
MsgBox	220
Laufzeitfehler verhindern	224
Benutzereingaben prüfen	224
Bedienreihenfolge erzwingen	225
Fehlfunktionen vorhersehen	226
OnError	227
Was ist wichtig?	229
6 Steuerelemente	231
In diesem Kapitel	232
Eigenschaften von Steuerelementen	232
Bezeichnungsfeld	233
Textfeld	234
Schaltfläche	239
Der Befehlsschaltflächen-Assistent	239
Programmierung der Schaltfläche	241
Nachträgliche Namensänderung	241
Listenfeld und Kombinationsfeld	242
Unsichtbare Spalten	243
Gebundene Spalte	243
Spaltennummerierung	244
Verwenden der Assistenten	245
Optionsgruppe	247
Registerkarten	249
Enabled, Visible und Locked	252
Exakte Platzierung	253

Das eigene Menüband	255
XML in Ten Minutes	256
Der XML-Code für unser Menüband	257
1. Schritt: Definition der Struktur	259
2. Schritt: Bekanntmachen der Definition	261
3. Schritt: Anzeige des Menübands	262
4. Schritt: Programmierung der Klicks	262
Ein letztes Sahnehäubchen	263
Was ist wichtig?	264
7 SQL	265
In diesem Kapitel	266
Wofür SQL?	266
Die Übungsdatenbank	267
CREATE, INSERT, UPDATE, DELETE	268
CREATE TABLE	268
INSERT INTO	269
UPDATE	270
DELETE FROM	270
Ausprobieren?	270
SELECT	272
Daten aus einer Tabelle abrufen	272
Daten aus mehreren Tabellen abrufen	272
Ausprobieren!	274
Inner Join	275
Outer Join	278
Schnell soll es gehen!	284
Was fehlt noch?	288
Abfragen!	289
Beziehungen in Abfragen	290
Weil es so wichtig ist	291
VBA, SQL – und was noch?	292
Was ist wichtig?	292
8 VBA – Anwendungen	295
In diesem Kapitel	296
Die Datenbank	296
Gebundene und ungebundene Formulare	297
»Handgeschnittze« Kommunikation	300
Recordsets	300
Direktzugriff mit SQL	302
Eingebettetes Makro vs. Ereignisprozedur	303
Speichern der Formularinhalte	304
Mit einem Recordset	304
Mit SQL	305
Implizites Speichern	305
Muss-Felder überprüfen	306
»Geisterdaten«	307
Datenauswahl mit Listenfeldern	307

Exkurs: Das Objektmodell	309
Lesen von Werten aus Tabellen	311
Mit einem Recordset	311
Mit SQL	312
Neue Datensätze in Tabellen anlegen	313
Mit einem Recordset	313
Mit SQL	314
Löschen von Werten in Tabellen	314
Mit einem Recordset	315
Mit SQL	316
Löschen rückgängig machen?	317
Standardlösungen	317
VBA-Code in Formularentwürfen ergänzen (»Schritt 6«)	318
Kombinationsfelder absichern (»Einzelzuordnung, Schritt 4«)	320
Erste Zeile eines Listenfelds anzeigen	321
Sub-Prozeduren	321
Text in ein Bezeichnungsfeld schreiben	323
Prüfen, ob ein bestimmter Datensatz existiert	323
Disponieren	324
Bearbeiteten Datensatz nach Requery wieder anzeigen	326
Formular von einem anderen Formular aus öffnen	327
Dialogfeld für die Datensuche öffnen	328
Alle Datensätze eines Recordsets bearbeiten	329
Alle Textfelder leeren (For Each ...)	330
Auf das übergeordnete Formular zugreifen (Parent)	330
Abhängige Listen- und Kombinationsfelder (RowSource)	331
Zwischen verschiedenen Anzeigemodi umschalten (Umschaltfläche)	332
Mehrere Werte aus einer Liste zuordnen	334
Mehrere Spalten in einem Kombinationsfeld anzeigen	335
Was ist wichtig?	336
9 Formulargrundtypen	337
In diesem Kapitel	338
Eine Tabelle	338
Eine Tabelle mit einer 1:n-Beziehung	339
Eine Tabelle mit einer n:1-Beziehung	342
Eine Tabelle mit einer 1:n:1- und einer n:1-Beziehung	344
Nur Fremdschlüssel in der Zwischentabelle	344
Daten in der Zwischentabelle	346
Eine Tabelle mit zwei n:1-Beziehungen	348
Eine Tabelle mit einer 1:m:n- und einer 1:n:1-Beziehung	350
Eine Tabelle mit einer 1:n- und einer n:1-Beziehung	351
Eine Tabelle mit zwei 1:n:1-Beziehungen	353
Eine Tabelle mit zwei 1:n- und zwei n:1-Beziehungen	354
Was ist wichtig?	356

10 Datenbankanwendungen	357
In diesem Kapitel	358
Vorgehensweise	358
Schrittfolge	358
Versionen	359
Entwicklung für andere	360
Aufteilung in Frontend und Backend	361
Das Problem	361
Die Lösung	362
Das verbleibende Problem	365
Welche Version ist es denn?	367
Frontend-gesteuertes Backend-Update	369
Schutz der Datenbank	371
Schutz vor unbefugten Personen	371
Der Rundumschutz	372
Schutz vor befugten Personen	373
... durch ein VBA-Kennwort	373
... durch die Erzeugung einer .accde-Datei	373
Datenbankaufteilung plus .accde-Datei	374
Access ohne Access?	374
Wie geht's weiter?	377
Abfragen	377
Berichte	377
Makros	378
Dateien lesen und schreiben	380
Benutzerberechtigungen	381
Mehrbenutzerzugriff	381
DAO und ADO	382
Was ist wichtig?	383
A Wichtige Standardaktionen durchführen	385
So geht es mit Access 2016/2019	386
Datenbank erstellen und öffnen	386
Tabellen und Beziehungen	386
Abfragen	387
Formulare	388
Makros	389
VBA	389
So geht es mit Access 2013	389
Datenbank erstellen und öffnen	389
Tabellen und Beziehungen	390
Abfragen	390
Formulare	391
Makros	392
VBA	392
So geht es mit Access 2010	393
Datenbank erstellen und öffnen	393
Tabellen und Beziehungen	393
Abfragen	394
Formulare	395

Inhaltsverzeichnis

Makros	396
VBA	396
So geht es mit Access 2007	396
Datenbank erstellen und öffnen	396
Tabellen und Beziehungen	397
Abfragen	397
Formulare	398
Makros	399
VBA	399
B Namenskonventionen	401
Namen für Access-Objekte	401
Namen für Steuerelemente auf dem Formular	401
Namen für Variablen im VBA-Code	402
Namen für Tabellen und Tabellenspalten	402
Stichwortverzeichnis	403

Ein Leserbrief

Hallo Herr Stern,

meine Buchverwaltung mit Mahnung läuft jetzt, wie ich es wollte. Mein Fehler war, ich habe zu viel im Internet gesucht, anstatt in Ihrem Buch nachzuschlagen.

Nachdem ich mich mehr mit Ihrem Buch beschäftigt habe, konnte ich meine Fehler schnell beseitigen. Ich habe vor dem Kauf des Buches lange in einer hiesigen Buchhandlung in diversen Access-Büchern gelesen. Bei keinem hatte ich den Eindruck, dass es mir richtig weiterhilft. Ganz zum Schluss habe ich dann noch Ihr Buch entdeckt und spontan gekauft.

Jetzt, nachdem ich das Buch schon genauer kenne, muss ich sagen: »Volltreffer.« Vielen Dank noch mal.

Helmut Ebeling

Einleitung

Warum ist Access anders?	15
Was habe ich mit Ihnen vor?	16
Wer sollte dieses Buch kaufen?	17

Warum ist Access anders?

Haben Sie Access auf Ihrem Rechner? Bestimmt!

Haben Sie Access schon einmal gestartet? Bestimmt!

Konnten Sie etwas damit anfangen? Höchstwahrscheinlich nicht!

Sehen Sie – und genau darum habe ich dieses Buch geschrieben: weil ich es furchtbar schade finde, dass diese wunderbare Software ungenutzt auf so vielen Rechnern herumliegt.

Warum ist das so? Wenn Sie Word, PowerPoint oder Excel starten, können Sie sofort auch ohne Übung etwas damit anfangen – nämlich einen Text schreiben, eine Grafik erstellen bzw. etwas berechnen. Wenn Sie dagegen Access starten, können Sie damit erst einmal gar nichts anfangen. Schreiben, zeichnen und rechnen kann jeder.

Bei Access geht es aber um mehr. Access ist eine Datenbank, die Tabellen verwendet, um Daten zu verwalten. Da stellt sich als Erstes die Frage: Welche Tabellen brauche ich denn? Beim Nachdenken darüber müssen Sie erst mal einen Schritt zurückgehen und überlegen: Welche Daten will ich denn überhaupt verwalten? Und beim Nachdenken darüber wiederum müssen Sie noch einmal einen Schritt zurückgehen und überlegen: Welche Problematik will ich denn eigentlich mit Access bearbeiten?

Damit sind Sie mittendrin in einer sogenannten »Systemanalyse« – einer Tätigkeit, die bestimmte Fertigkeiten voraussetzt, bei der Sie bestimmte Methoden brauchen und eventuell Softwaretools benutzen. Sie können also im Gegensatz zu Word, PowerPoint und Excel nicht sofort loslegen. Und das ist genau die große Hemmschwelle beim Einsatz von Access.

Was habe ich mit Ihnen vor?

Mit diesem Buch möchte Ihnen helfen, diese Hemmschwelle zu überwinden.

Dazu habe ich einen relativ bequemen Weg zum Ziel vorbereitet, auf dem ich Sie mitnehmen möchte. Es wird manchmal so aussehen, als wäre eine Abkürzung schneller, und Sie werden geneigt sein, einige Seiten oder Kapitel zu überblättern. Bitte tun Sie das nicht! Bei dem, was ich Ihnen hier zeigen will, kommt es auf große Genauigkeit im Detail an. Wenn Sie also von meinem vorgezeichneten Weg abweichen, kann es passieren, dass Sie ganz woanders im Nirwana landen.

Das Gebiet, das ich Ihnen hier nahebringen möchte, ist sehr umfangreich und zum Teil ziemlich schwierig zu verstehen. Klassischerweise müsste ich Baustein für Baustein die Grundlagen erklären und das Ganze dann am Ende zu einer Datenbankanwendung zusammenbauen. Nach meiner Erfahrung verliere ich auf diesem Weg aber den größten Teil der Zuhörer bzw. Leser, weil das Ziel nicht bekannt ist und aus den einzelnen Bausteinen auch nicht erahnt werden kann. Das ist so, als würden Sie ein Haus bauen wollen und sich daher lange Vorträge über die verschiedenen Arten von Mauersteinen und Dachziegeln anhören müssen. Nein – Sie wollen erst einmal ein Musterhaus besichtigen!

So will ich es auch in diesem Buch machen.

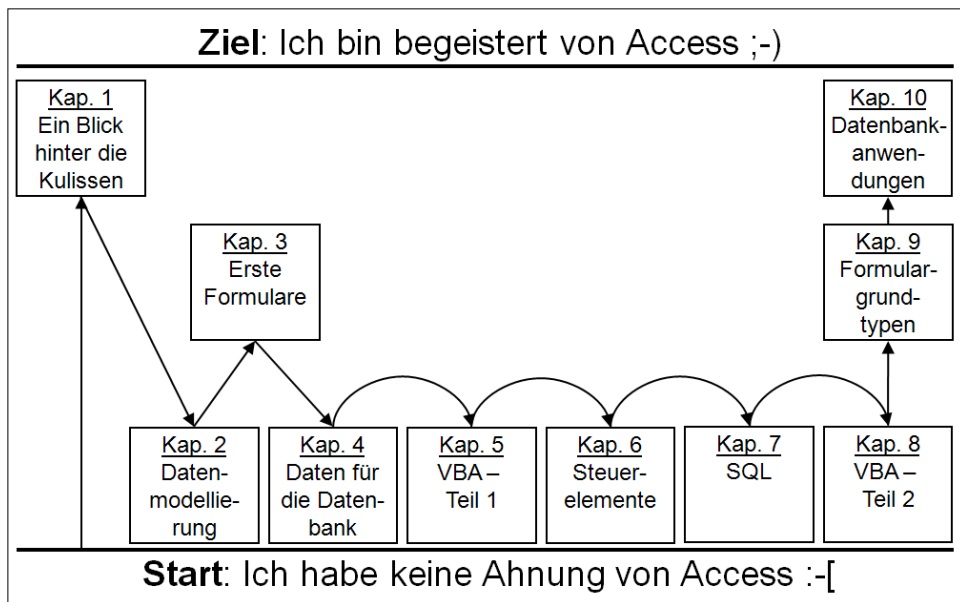


Abbildung 1: Das ist der Weg, den ich gemeinsam mit Ihnen gehen will!

In Kapitel 1 fliegen wir über alle Probleme hinweg gleich vom Start bis ans Ziel und sehen uns gemeinsam eine fertige Datenbankanwendung an. Ich hoffe, dass Sie ihre Nützlichkeit erkennen und danach motiviert sind, einige Anstrengungen auf sich zu nehmen, um selbst solche Anwendungen entwickeln zu können.

In Kapitel 2 gehen wir noch einmal ganz zurück an den Start und legen erste, einfache Grundlagen dafür, den Weg zum Ziel selbst gehen zu können.

In Kapitel 3 schaffen Sie den Weg zum Ziel schon fast aus eigener Kraft. Sie lernen, einfache Access-Formulare selbst zu entwickeln. Mit diesem Kapitel bekommen Sie auch eine Ahnung davon, was man alles an theoretischen Grundlagen und praktischen Fähigkeiten braucht, um Datenbankanwendungen zu entwickeln.

Darum begeben wir uns mit Kapitel 4 erneut zurück an den Start und legen zusammen mit den Kapiteln 5, 6, 7 und 8 erst einmal eine breite Basis für den erneuten Start zum Ziel: die Entwicklung von Datenbankanwendungen. Da Sie in den Kapiteln 1 und 3 schon einmal am Ziel gewesen sind, wissen Sie jetzt, warum die hier erläuterten Themen so wichtig sind.

In den Kapiteln 9 und 10 möchte ich Sie dann befähigen, das Ziel aus eigener Kraft zu erreichen. Sie sind anschließend in der Lage, selbst Datenbankanwendungen zu entwickeln.

Wer sollte dieses Buch kaufen?

Ganz vorn auf dem Buchtitel steht: »Datenbanken verstehen, entwerfen und entwickeln« – also **nicht** nur »Datenbanken benutzen«! Als Zielgruppe für dieses Buch sehe ich daher

- Schüler und Studenten, die etwas über die Entwicklung von Datenbankanwendungen lernen wollen,
- kleine Unternehmen, Handwerksbetriebe und Vereine, die ihre Daten bisher mit Bleistift und Papier oder aber maximal mit Excel verwalten,
- Mitarbeiter größerer Unternehmen, die mit der Funktionalität der »großen Unternehmenssoftware« nicht ganz zufrieden sind und sich eine eigene Datenbank anlegen wollen, und
- Softwareentwickler, die sich bisher nicht an Access-Anwendungen herangetraut haben.

Für die meisten Zwecke gibt es fix und fertige (Branchen-)Software – zum Teil sogar sehr preisgünstig auf dem Sharewaremarkt. Eine »selbst gestrickte« Anwendung hat aber mindestens zwei schwergewichtige Vorteile:

- Sie können sie 100%ig auf Ihre eigenen Bedürfnisse zuschneiden. Branchensoftware hat immer das Problem, dass sie für den »allgemeinen Friseursalon«, für den »allgemeinen Tennisklub« usw. entwickelt wurde. Infolgedessen werden Ihnen einerseits Funktionen fehlen, die Sie benötigen – andererseits gibt es eine Reihe von überflüssigen Dingen, die Sie nicht brauchen, die Sie aber in der Arbeit mit der Software eher behindern.
- Sie sind bezüglich Updates, Fehlerbehebung und Funktionserweiterung nicht von einem Hersteller abhängig, sondern können das alles selbst erledigen.

Die Entwicklung einer Datenbankanwendung erfordert nämlich viel detailliertes Know-how über die Abläufe im Betrieb oder im Verein, d. h., Sie müssen wissen,

- welche Objekte von Bedeutung sind (Kunden, Aufträge, Mitarbeiter, Fahrzeuge usw. oder Mitglieder, Trainer, Wettkämpfe ...),
- wie diese Objekte zusammenhängen (»Kunden erteilen Aufträge«, »Autoren schreiben Bücher«, »Mitglieder nehmen an Wettkämpfen teil« ...) und
- welche Eigenschaften diese Objekte haben (Name, Datum, Preis, Menge ...).

Dieses Wissen lässt sich nicht so schnell erwerben – es wird meist in jahrelanger Tätigkeit in der entsprechenden Umgebung gesammelt. Im Gegensatz dazu lässt sich das EDV-Wissen, das für die Entwicklung einer Datenbankanwendung erforderlich ist, relativ schnell erlernen. Ich will Ihnen dazu eine kurze Geschichte erzählen:

Einleitung

Ich lasse meine Studenten Datenbankanwendungen entwickeln für Gebiete, auf denen sie schon selbst Erfahrungen gesammelt haben. Dabei war einmal eine Studentin, deren Vater einen Bauernhof mit Viehzucht hatte. Sie entwickelte in der Informatikvorlesung eine Datenbankanwendung, in der es darum ging festzuhalten, welche Kuh von wann bis wann in welchem Stall war und wann welches Futter bekommen hatte. Das war gerade zu der Zeit der großen BSE-Krise. Nun hatte diese Studentin zufällig auch noch einen Bekannten, der in der Softwareentwicklung arbeitete – der aber natürlich keine Ahnung davon hatte, wie es auf einem Bauernhof zugeht. Er erkannte jedoch sehr schnell das große Potenzial dieser Datenbankanwendung und seine Firma kaufte sie meiner Studentin ab, um sie professionell weiterzuentwickeln. Der Erlös reichte meiner Studentin für die Anschaffung eines gebrauchten Kleinwagens.

Erkennen Sie, was ich meine? Wenn Sie gut Bescheid wissen in einer Firma, einem Verein oder irgendeiner anderen Organisation, in der Menschen zusammenarbeiten, und bisher noch nie auf den Gedanken gekommen sind, dieses Wissen in eine Datenbankanwendung umzusetzen, dann sollten Sie es jetzt vielleicht tun! Es gibt sehr viele potenzielle Kunden, die nur darauf warten!

Die Chance, damit erfolgreich zu sein, ist seit Access 2007 noch einmal gestiegen: Microsoft verteilt nämlich seitdem die sogenannte Access Runtime kostenlos. Das ist eine Software, die es ermöglicht, dass auch Menschen, die gar kein Access auf ihrem Rechner installiert haben, trotzdem Access-Anwendungen benutzen können.



Ich sehe hier für viele Leute eine Chance, Geld zu verdienen!

Gehen Sie dazu einmal auf die Webseite www.microsoft.com/downloads und geben Sie dort den Suchbegriff »Access Runtime« ein. Sie erhalten mehrere Treffer für die verschiedenen Access-Versionen.

Ich wünsche Ihnen daher nicht nur viel Lesevergnügen mit diesem Buch, sondern anschließend auch wirtschaftlichen Erfolg, der sich für Sie in Euro und Cent auszahlt!

Meine Devise beim Schreiben dieses Buchs war:

Alles Notwendige so	kurz	wie möglich darstellen!
Schwierige Dinge	verständlich	erläutern!
Die Auswahl der Inhalte	konzentriert	auf das Wesentliche beschränken!

»So eine Arbeit wird eigentlich nie fertig, man muss sie für fertig erklären, wenn man nach Zeit und Umständen das Möglichste getan hat.«

(J. W. v. Goethe, Italienische Reise, Caserta, 16.3.1787)

Andreas Stern

Rastede, im März 2019



Beispieldatenbanken

Für dieses Buch gibt es einige Beispieldatenbanken und weitere Dateien. Diese finden Sie zum Download im Internet auf oreilly.de auf der Webseite zum Buch.

Die Namen der Beispieldatenbanken sind folgendermaßen aufgebaut:

Name-v6xx.accdb bzw. Name-v6xx.mdb

Dabei bezeichnet

- *Name* den Namen der Datenbank,
- *v* die Version,
- *6* die 6. Auflage und
- *xx* die Versionsnummer der Datenbank.

Verein-v601.accdb ist dann also z.B. die Version 1 der Vereinsdatenbank aus der sechsten Auflage dieses Buchs. Ich werde im Buch dann nur auf *Verein* verweisen und die Versionsbezeichnungen weglassen.

Die Beispieldatenbanken sind mit den Access-Versionen 2007, 2010, 2013, 2016 und 2019 benutzbar – aber nicht mit der Version 2003.



Schauen Sie ab und zu einmal auf der Webseite www.buch.andreasstern.de vorbei! Dort werde ich neue Versionen der Beispieldatenbanken und weiteres Material zum Download anbieten.



Access 2003 Auf die Access-Version 2003 wird in dieser Auflage nicht mehr eingegangen. Lesern, die noch mit dieser Version arbeiten wollen, empfehle ich, sich ein Exemplar dieses Buchs aus der dritten Auflage zu besorgen.

Kapitel 1

Ein Blick hinter die Kulissen

In diesem Kapitel	22
Technische Voraussetzungen	22
Ist Access eine Datenbank?	23
Formulare benutzen	29
Tabellen, Felder und Primärschlüssel	42
Beziehungen und Fremdschlüssel	50
Formulare entwerfen	54
VBA – ganz kurz	65
Was ist wichtig?	68

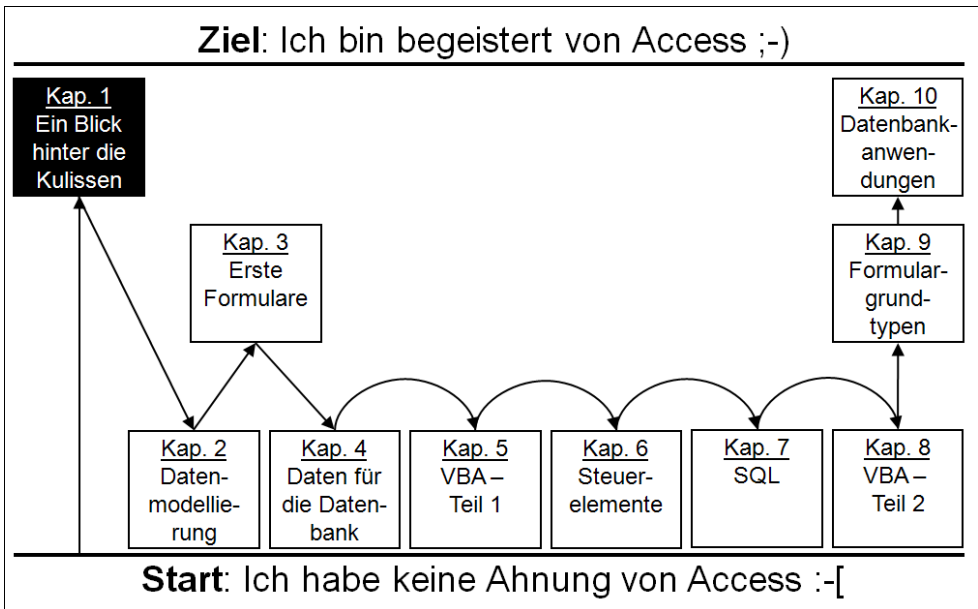


Abbildung 1.1: Das Kapitel 1, »Ein Blick hinter die Kulissen«.

In diesem Kapitel

... beginnen wir mit dem Ende: Ich stelle Ihnen eine fertige Datenbankanwendung vor, und wir schauen uns einmal an, wie das Ganze funktioniert. Das hat einen großen Vorteil: Sie wissen dann, wo wir hinwollen, und gewinnen hoffentlich die Überzeugung, dass es sich lohnt, in den folgenden Kapiteln für dieses Ziel einige Mühen auf sich zu nehmen.

Technische Voraussetzungen

Damit Sie die in diesem Kapitel beschriebenen Aktionen auf Ihrem Computer nachvollziehen können, sind einige technische Voraussetzungen erforderlich.

Zunächst einmal brauchen Sie Access auf Ihrem Computer. Welche Version? Nun – ich habe die Beispiele mit der Version 2010 erstellt, sie lassen sich aber auch mit den Access-Versionen 2007, 2013, 2016 und 2019 öffnen. Daher sollte alles funktionieren, wenn Sie mit einer dieser fünf Versionen arbeiten.

Jetzt kommt aber etwas sehr Wichtiges:



Damit Access-Datenbanken funktionieren, werden sogenannte Bibliotheken benötigt – das sind Sammlungen von vielen einzelnen kleinen Programmen, die für diverse Aktionen benötigt werden.

Diese Bibliotheken befinden sich schon auf Ihrem Computer – Access hat aber unter Umständen noch nicht den Befehl bekommen, sie auch zu benutzen.

Um diesen Befehl zu erteilen, gehen Sie bitte folgendermaßen vor:

1. Öffnen Sie die Datenbank, mit der Sie arbeiten wollen – z.B. *Verein* –, entweder durch Doppelklick auf die Datei im Explorer oder durch Starten von Access und Aufrufen des Befehls *Öffnen* auf der Registerkarte *Datei* (bzw. im Menü der Schaltfläche *Office* von Access 2007).



Die Beispieldatenbanken finden Sie im Internet unter der in der Einleitung angegebenen Adresse.

2. Öffnen Sie den Visual Basic-Editor mit der Tastenkombination **Alt + F11**.
3. Rufen Sie im Visual Basic-Editor den Menübefehl *Extras/Verweise* auf.

Das Dialogfeld, das sich jetzt öffnet, muss genau so aussehen, wie in Abbildung 1.2 dargestellt. Das bedeutet, es müssen genau die gleichen Einträge in der Liste der Verweise aktiviert (d.h. mit einem Häkchen versehen) sein, und zwar auch in der gleichen **Reihenfolge** wie in Abbildung 1.2. Eventuell ist einer der erforderlichen Einträge bei Ihnen nicht aktiviert. Dann suchen Sie diesen Eintrag bitte in der Liste, aktivieren ihn durch einen Mausklick in das Kontrollkästchen links daneben und klicken auf die Schaltfläche *OK*. Dadurch wird der neu aktivierte Eintrag in den oberen Teil der Liste verschoben – aber noch nicht an die richtige Stelle. Dazu öffnen Sie bitte das Dialogfeld *Verweise* erneut, klicken einmal auf den neu aktivierten Eintrag und benutzen dann die Pfeile rechts neben der Liste, um den

Eintrag so zu verschieben, wie in Abbildung 1.2 dargestellt. So muss das Dialogfeld *Verweise* bei Ihnen aussehen, wenn Sie Access 2016/19 benutzen (in Access 2007, 2010 und 2013 stehen dort statt der Versionsnummer 16.0 bei drei Einträgen die Versionsnummern 12.0, 14.0 bzw. 15.0).

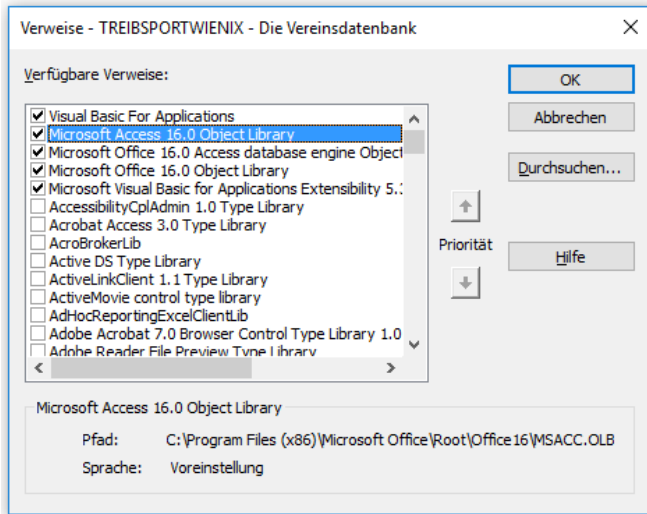


Abbildung 1.2: Das Dialogfeld *Verweise* in Access 2016/19



Bitte achten Sie unbedingt darauf, dass Ihr Dialogfeld *Verweise* sowohl hinsichtlich der aktivierten Einträge als auch hinsichtlich deren **Reihenfolge** genauso aussieht, wie in Abbildung 1.2 dargestellt.

Ist Access eine Datenbank?

Bevor wir uns mit der Beispieldatenbank beschäftigen, wollen wir uns die zentrale Frage stellen: Eine Datenbank – was ist das überhaupt? Leider ist es so, dass der Begriff »Datenbank« von verschiedenen Leuten in ganz unterschiedlichem Sinne benutzt wird. Um hier Klarheit zu schaffen, schauen Sie sich bitte Abbildung 1.3 an. Sie sehen dort schematisch dargestellt den Aufbau einer Datenbankanwendung. Die Daten selbst befinden sich in Tabellen – das werden wir uns weiter unten in diesem Kapitel noch genauer anschauen.

Der Inhalt dieser Tabellen muss verwaltet werden: Es müssen neue Daten an den richtigen Stellen eingefügt werden, vorhandene Daten müssen gegebenenfalls geändert werden, und ab und zu müssen auch überflüssig gewordene Daten gelöscht werden. All das wird von einer speziellen Software erledigt: dem sogenannten Datenbankmanagementsystem (DBMS, engl. Data Base Management System). Das DBMS arbeitet jedoch für den Benutzer unsichtbar. Er (oder sie ;-)) benötigt eine grafische Benutzeroberfläche (engl. Graphical User Interface, GUI).

Kapitel 1: Ein Blick hinter die Kulissen

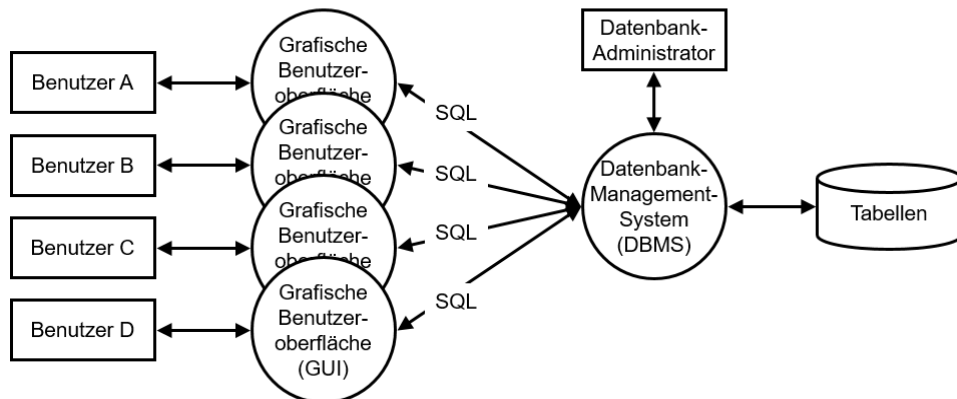


Abbildung 1.3: Die Komponenten einer Datenbankanwendung.

Damit sind die uns allen wohlbekannten Fenster mit Eingabefeldern, Schaltflächen (engl. Buttons), Symbolleisten usw. gemeint.

In einer »richtigen, großen« Unternehmensanwendung ist das so: Das DBMS läuft auf einem eigenen Gerät – dem Datenbankserver – und wird dort vom Datenbank-Administrator verwaltet. Er führt die eventuell notwendigen Korrekturen an der Tabellenstruktur der Datenbank aus, startet das System nach einem Absturz neu, installiert Updates, macht Sicherungen (engl. Backups) der Daten usw.

Über das Netzwerk (engl. Local Area Network, LAN) des Unternehmens sind mehr oder weniger viele Computer (engl. Clients) der Mitarbeiter angeschlossen. Auf ihnen läuft eine weitere Software, die den Benutzern das bereits erwähnte GUI zur Verfügung stellt. Damit können die Benutzer die Daten bearbeiten.

Was bedeutet der letzte Satz: »Die Benutzer können die Daten bearbeiten.« eigentlich genau?

Nehmen wir einmal an, es gibt eine Tabelle mit Kundendaten, und nun will ein Benutzer eine Kundenadresse ändern. Er öffnet ein bestimmtes Fenster seiner Clientanwendung (= GUI) und sucht sich den zu bearbeitenden Kunden heraus. Dazu fordert die Clientanwendung die entsprechenden Daten beim DBMS an. Das kann bedeuten, dass einfach pauschal alle Kundendaten aus der Kundentabelle herausgelesen und an die Clientanwendung übermittelt werden. Das kann aber auch bedeuten, dass der Benutzer in seiner Clientanwendung bestimmte Auswahlkriterien eingibt (z.B. »alle Hamburger Kunden« oder »alle Kunden, die noch offene Rechnungen haben« oder »Hartmut Meier aus 28195 Bremen, Martinistr. 111«) und dass nur solche Kundendaten übermittelt werden, die diese Kriterien erfüllen.

Bei diesem ganzen Prozess kommunizieren also das DBMS und die Clientanwendung über das LAN miteinander. Reden sie dabei eine bestimmte Sprache? Ja, so seltsam es klingt, das ist tatsächlich der Fall. Diese Sprache heißt SQL (engl. Structured Query Language), und wir werden uns damit in einem späteren Kapitel noch näher beschäftigen.

Hier nur so viel: SQL besteht aus einzelnen Befehlen, die das DBMS anweisen, bestimmte Aktionen auszuführen – z.B.:

```
SELECT kun_name FROM tblKunde WHERE kun_anschrift LIKE "Bremen"
```


Wie Sie sehen, ist das einfaches Englisch, das man auch ohne Kenntnisse der SQL-Sprache verstehen kann: Der obige Befehl sucht die Namen aller Bremer Kunden aus der Tabelle *tblKunde* heraus.

Hat der Benutzer die Daten dann mit seiner Clientanwendung bearbeitet (weil sich z.B. die Telefonnummer des Kunden geändert hat), geht etwa ein solcher SQL-Befehl an das DBMS:

```
UPDATE tblKunde SET kun_telefon="(0421) 11 22 33" WHERE kun_id=1423
```

Also: Eine Datenbankanwendung besteht aus

- den eigentlichen Daten, die in Tabellen abgelegt werden,
- dem DBMS, das diese Daten verwaltet, und
- einer grafischen Benutzeroberfläche (GUI), mit deren Hilfe die Benutzer die Daten bearbeiten.

Und jetzt ist es wie gesagt leider so, dass wahlweise eine dieser drei Komponenten oder Kombinationen daraus als »Datenbank« bezeichnet werden. Eine Datenbank kann also sein:

- der reine Datenbestand (»Wir haben jetzt eine europaweite Verbrecherdatenbank.«),
- das DBMS (»Wir haben eine Datenbank von Microsoft gekauft.«),
- der Datenbestand und das DBMS zusammen (»Herr Meier administriert unsere Unternehmensdatenbank.«) oder
- der Datenbestand, das DBMS und das GUI zusammen (»Die Kollegen sind sehr zufrieden mit der neuen Gefahrgutdatenbank.«).

Eine Festlegung darüber, was denn nun »richtig« ist, gibt es nicht. Sie sollten daher beim Thema »Datenbanken« immer hinterfragen, was im konkreten Einzelfall damit gemeint ist. Ich persönlich bevorzuge den Begriff *Datenbank* als Zusammenfassung für den Datenbestand und das DBMS, das diesen verwaltet. Kommt eine Clientanwendung hinzu, so nenne ich das Ganze *Datenbankanwendung*.

Und was ist dann Access? Eine Datenbank?

Access ist ein Softwarewerkzeug (engl. Tool) zur Erstellung von Datenbankanwendungen im oben genannten Sinne. Access enthält zum einen das DBMS (*Microsoft Jet Engine*), das die in separaten Dateien mit der Erweiterung *.accdb* hinterlegten Daten verwaltet. Es enthält jedoch auch Werkzeuge zum Erstellen der grafischen Benutzeroberfläche mit entsprechenden Fenstern zum Bearbeiten der Daten. Der Programmcode für diese grafische Benutzeroberfläche ist ebenfalls in den *.accdb*-Dateien enthalten.

Damit ist Access also wesentlich komplizierter als Word, Excel und PowerPoint. Diese drei Programme ermöglichen es Ihnen, Texte, Tabellen und Präsentationen zu bearbeiten, die in *.docx*-, *.xlsx*- bzw. *.pptx*-Dateien abgelegt sind. Mit einem Doppelklick auf eine solche Datei im Explorer weisen Sie Windows an, die damit verknüpfte Anwendung zu starten, den Inhalt der Datei zu laden und ihn zur Bearbeitung auf dem Bildschirm darzustellen.

Bei einem Doppelklick auf eine Access-Datei mit der Endung *.accdb* passiert dagegen mehr:

1. Das in Access enthaltene DBMS (*MS Jet Engine*) wird gestartet und wartet anschließend auf Befehle zur Bearbeitung der in der Access-Datei enthaltenen Daten.

Kapitel 1: Ein Blick hinter die Kulissen

2. Der in der Access-Datei enthaltene Programmcode der grafischen Benutzeroberfläche wird geladen und ausgeführt, d.h., entsprechende Fenster zur Bearbeitung von Daten erscheinen auf dem Bildschirm.

Darüber hinaus enthält Access Softwarekomponenten, die es Ihnen ermöglichen,

- Programmcode für grafische Benutzeroberflächen zu entwickeln, d.h. Fenster zur Datenbearbeitung (*Formulare* genannt) zu entwerfen, sowie
- das DBMS zu administrieren, d.h. Tabellen anzulegen, zu ändern, zu löschen und miteinander zu verknüpfen (*Beziehungen*).

Das wird alles zusammen auch *Entwicklungsumgebung* genannt. Damit funktioniert Access so, wie in Abbildung 1.4 dargestellt:

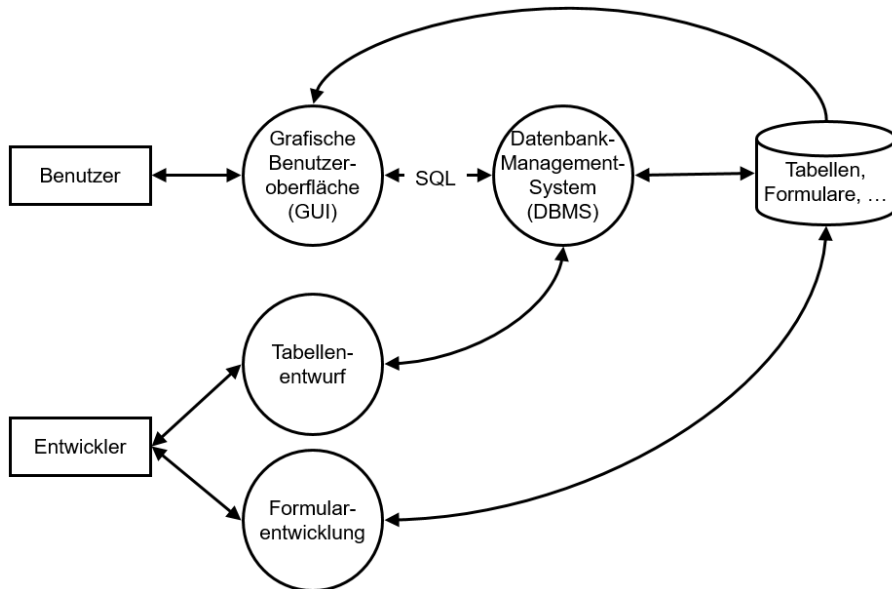


Abbildung 1.4: Access ist ein Werkzeug zur Erstellung von Datenbankanwendungen.

Wenn Sie noch keine Erfahrungen mit Access haben, wird Ihnen das eben Gesagte sicher noch etwas unverständlich erscheinen. Aber dafür ist dieses Kapitel ja auch gedacht: Wir wollen uns die ganze Theorie nun mal praktisch anschauen!

Access ohne Access?

Vorher kommt aber noch eine kurze Erläuterung zu einer besonderen »Variante« von Access. Mit der Version 2007 hat Microsoft nämlich die sogenannte Runtime-Version freigegeben, d.h., diese kann nun kostenlos heruntergeladen und benutzt werden. Da sich die Adressen von Webseiten viel zu schnell ändern, ergibt es keinen Sinn, hier die konkrete Downloadadresse anzugeben. Tippen Sie einfach in einer Internetsuchmaschine »download access runtime« ein, und Sie werden ruck, zuck an der richtigen Stelle sein.

Das können Sie sich so vorstellen wie bei den bereits seit Langem bekannten Viewern für Word, Excel und PowerPoint: Damit können Benutzer sich Word-, Excel- bzw. PowerPoint-

Dateien anschauen, ohne die entsprechende Software auf ihrem Computer installiert zu haben. Die Dateien können mit den Viewern nicht bearbeitet werden.

Bitte verwechseln Sie das aber nicht mit einem Schutz Ihrer Dateien vor Bearbeitung oder Verfälschung. Wer Word, Excel oder PowerPoint auf seinem Computer hat, kann Ihre Datei natürlich öffnen und bearbeiten. Nur wer diese Software nicht hat, sondern lediglich den Viewer, kann sich Ihre Datei nur anschauen und sie nicht bearbeiten.

Die Viewer befinden sich auf Ihrer Microsoft Office-Installations-CD. Sie können sie legal und kostenfrei weitergeben, damit andere, die kein Microsoft Office auf ihrem Computer haben, sich Ihre Word-, Excel- bzw. PowerPoint-Dateien anschauen können.

Genauso funktioniert auch die Access Runtime: Wenn diese Software auf einem Computer installiert ist, können Datenbank Anwendungen benutzt werden, ohne dass hierzu Access selbst installiert sein muss.

Zitat von Microsofts Downloadseite (am 12.2.2019; Hervorhebungen von mir):

»Microsoft Access 2016 bietet eine umfassende Plattform zur Entwicklung von Datenbankverwaltungslösungen mit benutzerfreundlichen Anpassungstools. Wenn keine Endbenutzeranpassung erforderlich ist ..., können Sie diese Access 2016-Lösungen verteilen, damit sie **ohne eine vollständige Installation von Access 2016** ausgeführt werden können. Dazu müssen Sie die Anwendung mit der Access 2016-Runtime verteilen ... Sie müssen kein spezielles Produkt erwerben, um die Access 2016-Runtime verteilen zu können. **Sie können sie kostenlos verteilen** oder auf diesen Download verweisen.«



Der kostenlose Zugang zur Access Runtime ist eine große Chance für Softwareentwickler! Sie können damit Datenbank Anwendungen entwickeln, die beim Benutzer ohne eine Access-Installation laufen

Wie die Access Runtime zu benutzen ist, erläutere ich im letzten Kapitel.

Die Access-Anwendung

Jetzt wollen wir uns aber – wie versprochen – eine fertige Access-Anwendung ansehen. Ich habe für dieses Buch drei komplette Beispielanwendungen entwickelt:

- eine Firmenanwendung, mit der die Abarbeitung von Kundenaufträgen in einem mittelständischen Unternehmen bzw. einem Handwerksbetrieb verwaltet werden kann,
- eine Vereisanwendung, mit der Mitglieder, Trainingszeiten und Wettkämpfe eines Sportvereins verwaltet werden können, und
- eine Privatanwendung, mit der eine Sammlung (z.B. Bücher) und deren Ausleihe an Bekannte verwaltet werden kann.

Für die Erläuterungen in diesem Kapitel verwende ich vorrangig die Vereisanwendung. Beim Namen dieser Datei (z.B. *Verein-v601.accdb*) wird Ihnen auffallen, dass sich darin eine Versionsnummer befindet (z.B. »v601«). Das ist ein Grundprinzip bei der Softwareentwicklung, das ich Ihnen gleich hier wärmstens ans Herz legen möchte: Erreichte Zwischenstände bei der Entwicklung der Anwendung in separaten Dateien mit Versionsnummern sichern!

Kapitel 1: Ein Blick hinter die Kulissen

Wie Sie sicherlich aus leidvoller Erfahrung wissen, geht am Computer immer irgendetwas schief, und das meiste läuft erst einmal nicht so, wie Sie es sich vorgestellt haben. Das gilt erst recht bei der Softwareentwicklung, denn dabei versuchen Sie ja besonders intensiv, den Computer zu bewegen, das zu tun, was Sie wollen. Jeder Programmierer – auch der erfahrenste – macht aber Fehler,

- die nur mit viel Aufwand gefunden und korrigiert werden können,
- die im schlimmsten Fall zum Absturz des Computers und zum Verlust der entwickelten Anwendung führen.

Dann ist derjenige gut dran, der sagen kann: »Macht nichts! Ich habe ja noch eine vorherige Version, die ich gespeichert habe, bevor ich den Fehler gemacht habe.«



Daher sollten Sie immer, wenn Sie einen bestimmten Stand der Entwicklung erreicht haben (das kann im Minutentakt sein oder aber am Ende des Arbeitstags), die entsprechenden Dateien unter einer Versionsnummer speichern, eine Kopie davon anfertigen, diese mit der nächsthöheren Versionsnummer versehen und dann damit weiterarbeiten.

Bitte öffnen Sie jetzt die Anwendung *Verein*.



Die Beispieldatenbanken finden Sie im Internet unter der in der Einleitung angegebenen Adresse.

Das kann durch einen Doppelklick mit der linken Maustaste auf den Dateinamen im Explorer geschehen. Sollte das aber aus irgendeinem Grund nicht funktionieren, starten Sie zunächst Access und klicken dann auf *Datei/Öffnen* (bzw. in Access 2007 auf die Office-Schaltfläche und dann auf *Öffnen*). Die weitere Vorgehensweise ist Ihnen sicherlich aus anderen Office-Produkten bekannt.

Jetzt kann schon das erste kleine Problem auftauchen: Im Allgemeinen erhalten Sie eine Sicherheitswarnung, und wenn Sie darauf nicht reagieren, wird die Datenbankanwendung nicht funktionieren.


 **SICHERHEITSWARNUNG** Einige aktive Inhalte wurden deaktiviert. Klicken Sie hier, um weitere Details anzuzeigen.

Abbildung 1.5: Wahrscheinlich bekommen Sie gleich zu Beginn der Arbeit mit der Beispielanwendung diese Sicherheitswarnung von Access zu Gesicht.

In diesem Fall folgen Sie einfach den Anweisungen, die in dem Fenster *Achtung! Was ist beim ersten Start dieser Anwendung zu tun?* erscheinen.

Innerhalb des Access-Fensters sehen Sie jetzt ein Fenster mit dem Titel *Startformular*. Am oberen Rand befindet sich das Menüband (Access 2007: die Multifunktionsleiste), und links sehen Sie den Navigationsbereich. Letzterer ist das »Steuerpult« von Access, von dem aus Sie alle weiteren Fenster erreichen, die Sie für die Anwendungsentwicklung benötigen. Wir wollen diese Bereiche zunächst nicht weiter beachten und uns nur mit dem Fenster *Startformular* beschäftigen.

Formulare benutzen

Das Fenster *Startformular* ist Bestandteil der weiter oben in diesem Kapitel beschriebenen grafischen Benutzeroberfläche (GUI). Diese Fenster werden in Access *Formulare* genannt (von engl. »form«). Das ist für den Access-Neuling immer etwas verwirrend, denn im deutschen Sprachgebrauch ist ein Formular ein Blatt Papier. Ausdrucke der Daten auf Papier heißen aber in Access *Berichte* (von engl. »report«).

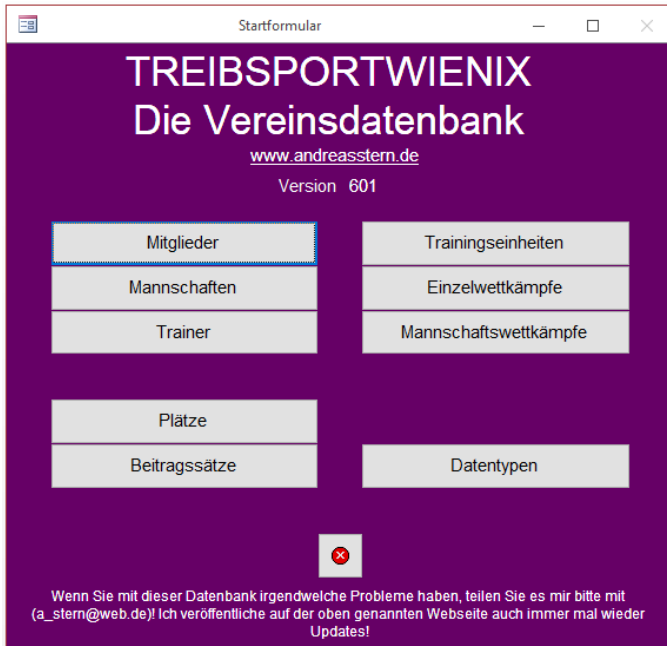


Abbildung 1.6: Das Startformular der Vereinsanwendung.

Das Startformular bietet noch keine Möglichkeiten, irgendwelche Daten zu bearbeiten. Es enthält einige Schaltflächen (engl. »Buttons«), auf die Sie klicken und damit weitere Formulare öffnen können. Tun Sie das ruhig einmal und beachten Sie dabei, dass Sie die sich öffnenden Formulare durch einen Klick auf die Schaltfläche rechts oben wieder schließen können.



Die Schaltfläche mit dem Schließen-Symbol ganz unten in der Mitte des Startformulars schließt nicht nur die Vereinsanwendung, sondern beendet auch Access!

Wie Sie aus der Beschriftung der Schaltflächen unschwer erkennen können, geht es in der Beispielanwendung *Vereinsdatenbank* um Folgendes:

- Es gibt Mitglieder im Verein, die Beiträge bezahlen.
- Die Höhe des Beitrags richtet sich nach dem Mitgliedstyp (»Erwachsener«, »ermäßig« ...).
- Die Mitglieder können in einer Mannschaft sein.

Kapitel 1: Ein Blick hinter die Kulissen

- Es gibt außerdem Trainer, die sowohl Einzelmitglieder als auch Mannschaften trainieren können.
- Mitglieder können Trainingszeiten auf Plätzen reservieren.
- Es finden Wettkämpfe statt, an denen Einzelmitglieder oder Mannschaften teilnehmen können.

Bevor Sie dieses Kapitel weiter durcharbeiten, möchte ich Sie ausdrücklich ermuntern, einmal etwas mit der Datenbankanwendung herumzuspielen. Tragen Sie einige Personen als Mitglieder ein, teilen Sie alle neu eingetragenen Mitglieder Mannschaften zu, reservieren Sie Trainingszeiten und nominieren Sie die Personen für Wettkämpfe. Dabei verändern Sie natürlich die Daten in der Datenbank. Im weiteren Verlauf dieses Kapitels nehme ich aber immer mal wieder Bezug auf die Version, die Sie sich heruntergeladen haben. Daher empfehle ich Ihnen dringend, sich eine »Spielkopie« der Originaldatei anzulegen. Wenn Sie dann die von mir beschriebenen Aktionen nachvollziehen möchten, können Sie immer wieder auf diese Originaldatei zurückgreifen. Ansonsten haben Sie ja immer noch die Möglichkeit, diese noch einmal herunterzuladen.

Dies ist ein Buch für Anfänger. Daher sollte immer ganz genau beschrieben werden, was wie getan werden muss. Das würde aber zu endlosen Wiederholungen führen und das Buch um etliche Seiten dicker machen. Dann müsste es mehr kosten, und Sie hätten es nicht gekauft. ;-)

Aber mal im Ernst: Die immer wiederkehrenden Standardaktionen erläutere ich nicht immer wieder. Diese sind in Anhang A, »Wichtige Standardaktionen durchführen«, beschrieben.



Außerdem finden Sie das Dokument *WichtigeStandardaktionen.pdf* zum Ausdrucken im Internet (Adresse in der Einleitung – dort im Ordner \KapA)!

Das Formular frmMitglieder

So, jetzt wollen wir uns einmal das Formular *frmMitglieder* etwas genauer ansehen. Es enthält viele Elemente, die wir später selbst erstellen wollen und die uns zeigen, wie praktisch die Arbeit mit einer Datenbankanwendung sein kann.

Abbildung 1.7: Das Formular frmMitglieder enthält viele typische Elemente einer grafischen Benutzeroberfläche.

Sie sehen links ein Listenfeld mit einer Liste der Mitglieder. Ein Mausklick in diese Liste führt dazu, dass die Daten des ausgewählten Mitglieds im rechten Teil des Formulars angezeigt werden. Wenn Sie einmal in die Liste geklickt haben, können Sie sich anschließend auch mit den Tasten und in der Liste auf und ab bewegen.

In der Mitte unten und rechts sehen Sie zwei sogenannte Unterformulare – zu erkennen daran, dass deren Inhalt gegenüber dem übergeordneten Formular etwas »eingedrückt« zu sein scheint. Das Unterformular in der Mitte zeigt, welche Art von Mitgliedschaft von wann bis wann bestand. Die Datenfelder mit der kleinen Schaltfläche an ihrem rechten Rand heißen Kombinationsfelder. Nach dem Klick auf diese Schaltfläche öffnet sich eine Liste, aus der Sie per Mausklick einen Eintrag auswählen können.

Versuchen Sie nun einmal, Herrn Hansen eine weitere Mitgliedschaft zuzuweisen. Vielleicht ist er seit Juni 2018 beitragsfrei. Sobald Sie anfangen, neue Daten einzugeben, erscheint links neben der mit Lö. (= Löschen) beschrifteten Schaltfläche das Symbol eines Sternchens. Das besagt, dass diese Daten bisher noch nicht in der Datenbank gespeichert wurden.



Denken Sie bitte an dieser Stelle an meine Erläuterungen im Abschnitt »Ist Access eine Datenbank?«: Die grafische Benutzeroberfläche (d.h. das Formular) hat also den SQL-Befehl zum Speichern der eingegebenen Daten noch nicht an das DBMS geschickt.

Noch können Sie Ihre Eingaben jederzeit durch Drücken der Taste rückgängig machen. Probieren Sie bitte auch das einmal aus.

Kapitel 1: Ein Blick hinter die Kulissen

Das Speichern der eingegebenen Daten in der Datenbank erfolgt

- entweder durch Mausklick auf das Sternchensymbol am linken Rand der Zeile, in der Sie die Daten eingegeben haben, oder
- indem Sie die Einfügemarke per Maus oder Tastatur in eine andere Zeile bewegen.

Letzteres ist besonders praktisch, wenn Sie viele neue Datenzeilen nacheinander einzugeben haben: Einfach in der nächsten Zeile weitertippen, und die Daten aus der Zeile, die Sie gerade verlassen haben, sind gespeichert.

Für das Löschen von Daten aus der Datenbank gibt es am linken Rand der Liste die bereits erwähnte Schaltfläche mit der Beschriftung *Lö.* Nach dem Klick auf diese Schaltfläche erscheint zunächst eine Warnung: »Wollen Sie die Mitgliedschaft wirklich löschen?«. Bestätigen Sie das, so verschwinden die gelöschten Daten aus der Liste.

Das Erscheinen dieser Warnung ist ein Merkmal guter Software. Der Benutzer sollte niemals in eine Falle gelockt werden (»Ätsch – weg ist weg!«). Das bedeutet, dass Sie als Entwickler immer bedenken müssen, dass der Benutzer Ihre Software unbeabsichtigt falsch bedienen könnte. Damit trotzdem nichts passiert, müssen Sie als Entwickler Vorsorge treffen – z. B. durch solche Warnungen. Wir werden noch öfter auf diesen Punkt zurückkommen, und Sie werden sehen, dass ein ganz erheblicher Teil des Aufwands bei der Softwareentwicklung in solche Vorsorgemaßnahmen fließt!

Bitte klicken Sie noch einmal auf die Schaltfläche *Lö.* und beachten Sie in dem sich öffnenden Warnfenster ein kleines, aber wichtiges Detail: Die Schaltfläche *Nein* ist etwas stärker umrandet als die Schaltfläche *Ja*. Sie ist die sogenannte Standardschaltfläche, die benutzt wird, wenn Sie statt der Maus die Tastatur verwenden und die Taste drücken. Wenn also der Benutzer auf *Lö.* klickt und dann in Unkenntnis dessen, was er tut, die -Taste drückt, ist immer noch nichts Schlimmes passiert. Damit haben wir einen weiteren Grundsatz fürs Programmieren: Die Standardaktion in einem Dialogfeld sollte immer die ungefährlichere sein.

Auf diese Weise muss sich der Benutzer also zweimal explizit fürs Löschen entscheiden: beim Klick auf die Schaltfläche und dann noch einmal durch die Auswahl von *Ja*. Wer hier beide Male falsch handelt, dem ist wohl wirklich nicht mehr zu helfen ...

Dieselben Erläuterungen wie für das mittlere Unterformular gelten auch für das rechte. Hier können Sie ebenfalls neue Daten eingeben und vorhandene ändern oder löschen. Einen kleinen Unterschied gibt es bei der Überschrift des rechten Unterformulars: Diese ist in Form einer Schaltfläche angelegt. Klicken Sie darauf, öffnet sich das Formular *frmBeitrags-saetze*. Das hat wieder etwas mit Benutzerfreundlichkeit zu tun: Wenn Sie Beiträge in die Liste eintragen, möchten Sie vielleicht kontrollieren, ob auch der richtige Betrag gezahlt wurde. Damit Sie dafür nicht ins Startformular zurück müssen, gibt es diese Schaltfläche hier. Das bedeutet für Sie als zukünftigen Entwickler von Datenbankanwendungen: Sie müssen sich in die Lage des Benutzers versetzen, sich fragen, was er (oder sie ;-)) in dieser Situation wohl machen möchte, und dann entsprechende Möglichkeiten für ihn schaffen. Auch das ist ein wichtiger Grundsatz der Softwareentwicklung, auf den wir noch öfter zurückkommen werden.

Was fällt Ihnen noch an den beiden Unterformularen auf? Beide enthalten Angaben zu aufeinanderfolgenden Zeiträumen. Diese Zeiträume sollten normalerweise nahtlos aneinanderpassen. Es sollte also keine Zeiträume undefinierter Mitgliedschaft oder ohne Beitragszahlung geben. Das könnte man natürlich softwaretechnisch durch eine entsprechende Programmierung lösen. Sowie ein Zeitraum ohne Mitgliedschaft oder ohne Beitrag entsteht,

könnte der Benutzer darüber informiert werden. Das würde aber erheblichen Programmieraufwand erfordern, und ich habe das für diese Beispielanwendung mal gelassen.

Nun zur weiteren Diskussion des Formulars *frmMitglieder*. Schauen wir einmal in die Mitte. Dort befinden sich – ähnlich wie bereits in den Unterformularen – zwei Kombinationsfelder, mit denen dem Mitglied ein Trainer bzw. eine Mannschaft zugewiesen wird. Die Beschriftung dieser beiden Kombinationsfelder ist wiederum als Schaltfläche gestaltet. Was passiert, wenn Sie darauf klicken? Das können Sie aus der Beschriftung der Schaltfläche nicht ableiten – aber der Spieltrieb siegt, und der Benutzer klickt drauf. Ups – das ist ja eine Löschfunktion!

Man kann diese Löschfunktion auch so programmieren, dass sie durch Drücken der Taste `[Entf]` ausgelöst wird. Ich habe das sogar getan – probieren Sie es ruhig aus! (Die Einfügemarke muss dazu in dem Feld mit dem Nachnamen des Trainers stehen.) Aber: Diese Funktionalität ist auf dem Bildschirm nicht zu erkennen. Der Benutzer kann davon nur durch eine Schulung, durch das Handbuchstudium oder durch erfahrungsgesteuertes Ausprobieren (»Learning by Doing«) erfahren. Dagegen verleitet die Schaltfläche einfach zum Draufklicken – und, schwupp, Sie wissen, wie's geht. Richtig gute Software sollte – wie in der vorliegenden Anwendung geschehen – beide Möglichkeiten bieten, um unterschiedliche Benutzertypen zufriedenzustellen.

Dass die Zuordnung eines Trainers durch Drücken der Taste `[Entf]` aufgehoben wird, bezeichnet man in der Softwaretechnik übrigens als »erwartungskonform«. Gute Software sollte sich erwartungskonform verhalten, d.h., wenn ich `[Entf]` drücke, kann ich erwarten, dass etwas entfernt wird, und wenn ich `[Einfüg]` drücke, kann ich erwarten, dass etwas eingefügt wird. Damit haben wir wieder einen Grundsatz, den Sie sich als zukünftiger Anwendungsentwickler merken sollten!

Und weil wir gerade dabei sind – hier noch ein Grundsatz: Die Bedienung von Software sollte gleichförmig sein. Damit ist allerdings nicht »langweilig« gemeint, sondern dass sich die gleichen Bedienelemente immer am gleichen Ort befinden und gleich aussehen. Umgekehrt sollen gleich aussehende Elemente auch die gleiche Funktion haben.


Was damit gemeint ist?

Öffnen Sie dazu bitte neben dem Formular *frmMitglieder* gleichzeitig das Formular *frmMannschaften*. Es sieht auf den ersten Blick ganz anders aus. Aber auf den zweiten Blick fällt Ihnen bestimmt Folgendes auf:

- Beide Formulare haben die gleiche Farbgestaltung: dunkle Hintergrundfarbe mit weißer Schrift oben und unten, heller Hintergrund in der Mitte.
- In beiden Formularen befinden sich die gleichen Schaltflächen fürs Einfügen, Speichern und Löschen von Datensätzen sowie für das Schließen des Formulars an den gleichen Stellen rechts unten bzw. oben.
- In beiden Formularen befindet sich links eine Liste der Objekte (einmal Mitglieder, einmal Mannschaften), und ein Klick in diese Liste führt in beiden Formularen zur Anzeige der Daten rechts daneben.

Kommen wir zu einem weiteren unsichtbaren Merkmal guter Software. Setzen Sie im Formular *frmMitglieder* bitte einmal die Einfügemarke in das Eingabefeld mit dem Nachnamen des Mitglieds, indem Sie mit der Maus dorthin klicken. Jetzt lassen Sie die Maus los und drücken die `[↵]`-Taste auf Ihrer Tastatur (ganz links). Die Einfügemarke wandert in das Textfeld mit dem Vornamen. Beim nächsten Drücken der `[↵]`-Taste ist die Einfügemarke im Textfeld *Geb.datum* usw.

Kapitel 1: Ein Blick hinter die Kulissen

Das sieht ziemlich banal aus, nicht wahr? Das ist aber ganz wichtig! Benutzer, die täglich stundenlang am Computer sitzen und deren Haupttätigkeit in der Datenerfassung besteht, benutzen die Maus häufig eher ungern. Das wird nämlich auf die Dauer ziemlich umständlich: Griff zur Maus, Klick in ein Textfeld, Hand zurück zur Tastatur, Daten eintippen, Hand wieder zur Maus, Klick ins nächste Textfeld usw. usw. Da ist es doch viel einfacher, die Finger auf der Tastatur zu lassen und sich mit der -Taste von Textfeld zu Textfeld zu bewegen! Dass dabei aber nacheinander die richtigen Textfelder angesprungen werden – dafür müssen Sie als Anwendungsentwickler sorgen.

Jetzt kommt wieder ein sichtbares Merkmal guter Software: das Layout des Formulars. Damit ist zum einen die Farbgestaltung gemeint, zum anderen Größe und Platzierung der Textfelder, Listen und Schaltflächen.

- Zur Farbgestaltung ist nicht viel zu sagen. Sie sollte so dezent wie möglich sein – also keine grellen Farben und nicht mehrere verschiedene Farben im selben Formular, sondern besser helle und dunkle Varianten derselben Farbe. Berücksichtigen Sie in diesem Zusammenhang die sogenannte Corporate Identity, d. h., verwenden Sie die für die Zielgruppe (Unternehmen, Verein) typischen Farben!
- Die Textfelder sollten so groß sein, dass der Benutzer alle darin möglicherweise erscheinenden Texte ohne zu scrollen lesen kann.
- Listen sollten so lang wie nötig sein – also weder eine zu lange und damit überwiegend leere Liste für drei Platztypen noch eine zu kurze Liste für die vielen Mitgliedernamen eines großen Vereins.
- Schaltflächen sollten im Allgemeinen ziemlich klein sein – gerade groß genug, dass der Text oder das Bild darauf noch erkennbar ist. Riesige Schaltflächen wirken plump.
- Übereinanderliegende Elemente sollten gleich breit und eventuell auch gleich hoch sein sowie links und rechts bündig angeordnet werden.
- Die Anordnung der Elemente sollte so erfolgen, dass diese von links nach rechts und von oben nach unten abgearbeitet werden können. Das wird natürlich nicht immer gelingen – Sie sollten es aber für den Normalfall anstreben. Auf jeden Fall sollte der Benutzer beim Bearbeiten der Daten nicht mehrfach im Formular hin und her springen müssen.

Alle diese Vorgaben dienen der »Augenfreundlichkeit« Ihrer Anwendung. Wer möglicherweise mehrere Stunden am Tag damit arbeiten muss, wird es Ihnen danken.

Beim Layout des Formulars *frmMitglieder* habe ich mir z. B. Folgendes überlegt:

- Das generelle Schema »links die Liste – rechts die Details« soll eingehalten werden.
- Die Mitgliederliste soll möglichst lang sein, um so viele Mitglieder wie möglich gleichzeitig im Blick zu haben, ohne scrollen zu müssen.
- Die Liste der Mitgliedschaften kann kurz sein, denn der Mitgliedstyp wird nicht so häufig gewechselt.
- Die Liste der gezahlten Beiträge wiederum soll möglichst lang sein, um viele Beiträge gleichzeitig auf dem Bildschirm zu haben.
- Beide Listen (Mitgliedschaften und Beiträge) brauchen nicht in der vollen Breite sichtbar zu sein, denn die Bemerkungsfelder werden häufig leer sein. Hauptsache, die wichtigen Eigenschaften sind auf einen Blick sichtbar.
- Das Bezeichnungsfeld *Mitgliedschaften seit* kann in die Kopfzeile der Liste der Mitgliedschaften, weil sich das vom Sachzusammenhang her anbietet.

Solche Überlegungen sollten Sie auch anstellen, wenn Sie später mal ein eigenes Formular entwerfen. Überlassen Sie dabei nichts dem Zufall. Überlegen Sie immer,

- wie lang und breit ein Element (Textfeld, Kombinationsfeld, Liste) sein muss,
- wo es am besten auf dem Formular platziert werden sollte,
- welche Elemente thematisch zusammenpassen,
- in welcher Reihenfolge ein Benutzer die Elemente wohl zur Datenbearbeitung benutzen wird und
- wie Sie den zur Verfügung stehenden Platz am besten ausnutzen, ohne dass die Elemente unübersichtlich zusammengedrängt werden.

Damit haben wir das Formular *frmMitglieder* fast erledigt. Zu den Schaltflächen rechts unten gibt es nicht mehr viel zu sagen – sie verhalten sich erwartungskonform. Die Schaltfläche *neu* gibt Ihnen die Möglichkeit, die Daten eines neuen Mitglieds einzugeben. Anschließend müssen Sie die Daten mithilfe der Schaltfläche *speichern* speichern. Dabei gibt es wieder einen Grundsatz zur Benutzerfreundlichkeit: Klicken Sie einmal auf die Schaltfläche *speichern* unmittelbar nach dem Klicken auf die Schaltfläche *neu*. Es erscheint die Warnung: »Bitte geben Sie einen Namen ein!« Die Ursache hierfür ist, dass ein Mitglied einen Namen haben muss, denn ein Mitglied ohne Namen ergibt keinen Sinn, und solche Daten sollen gar nicht erst in der Datenbank landen. Also weigert sich Access, zu speichern.

Das kennen Sie sicherlich aus dem Internet. Wenn Sie dort etwas bestellen und die Absenderangaben unvollständig angeben, erscheint das Eingabeformular noch einmal im Browser und fordert Sie auf, die Daten vollständig einzugeben.

Dass bestimmte Daten »Muss-Daten« sind, wird schon bei der Anlage der Tabellen in der Datenbank festgelegt – wir kommen später darauf zurück.

Bitte beachten Sie auch, dass nach dem Klick auf die Schaltfläche *neu* die Schaltflächen *neu* und *löschen* sowie die Mitgliederliste deaktiviert werden. Auch das trägt wieder zur Benutzerfreundlichkeit der Anwendung bei, indem es Fehlbedienungen vorbeugend verhindert. Der Benutzer **muss** jetzt neue Daten eingeben und kann nicht auf *löschen* klicken oder einfach einen anderen Datensatz auswählen.

Mit der dritten Schaltfläche wird ein Datensatz gelöscht. Erstaunlicherweise ist das datenbanktechnisch eine ganz, ganz schwierige Sache. Bei naivem Herangehen sollte man denken: »Wieso denn das? Weg damit und Ruhe ist!« Aber: In einer Datenbank gibt es nicht nur die Tabellen, sondern auch vielfältige Abhängigkeiten zwischen den Tabellen. In der Tabelle *tblBeitrag* stehen z. B. nicht nur die Beiträge, sondern auch Vermerke darüber, welches Mitglied diesen Beitrag gezahlt hat. Datenbanktechnisch heißt das: In der entsprechenden Zeile der Tabelle *tblBeitrag* steht, in welcher Zeile der Tabelle *tblMitglied* das dazugehörige Mitglied steht. Lösche ich nun dieses Mitglied, zeigt der Verweis in der Tabelle *tblBeitrag* auf ein nicht mehr existierendes Mitglied. »Okay«, werden Sie sagen. »Dann muss ich diesen Verweis eben auch löschen.« Und genau das ist die Problematik: Man kann eine Löschaktion nicht isoliert für sich ausführen, sondern muss die Auswirkungen auf andere Tabellen berücksichtigen. Mehr dazu im Abschnitt »Beziehungen und Fremdschlüssel« weiter unten in diesem Kapitel.

Abschließend folgt rechts oben noch eine Schaltfläche, mit der Sie das Startformular öffnen können. Bitte beachten Sie Folgendes: Wenn Sie jetzt im Startformular erneut auf *Mitglieder* klicken, wird dieses Formular nicht ein zweites Mal geöffnet – stattdessen wird das bereits geöffnete Formular *Mitglieder* wieder sichtbar.

Kapitel 1: Ein Blick hinter die Kulissen

Auch hier hat die Anordnung dieser Schaltflächen etwas mit Benutzerfreundlichkeit zu tun – und zwar entspricht die Reihenfolge der Schaltflächen der voraussichtlichen Reihenfolge der Benutzung: Zuerst wird ein Datensatz neu angelegt, dann wird er gespeichert und anschließend gegebenenfalls wieder gelöscht.

Warum befinden sich die Schaltflächen *Hilfe*, *Startformular öffnen* und *Formular schließen* oben statt unten? Nun, ich denke, das sind wir als Windows-Benutzer so gewohnt. Jedes Fenster hat rechts oben eine Schaltfläche zum Schließen des Fensters. Daher erwarten wir diese Schaltfläche auch rechts oben. Zur Hilfe gelangen wir in den allermeisten Softwareprodukten über einen Menübefehl mit dem Fragezeichen – dieser befindet sich meist rechts oben in der Menüleiste.

Ich möchte Ihre Geduld aber jetzt nicht überstrapazieren, indem ich jedes einzelne Formular mit der gleichen Gründlichkeit diskutiere wie das Formular *frmMitglieder*. Bei den weiteren Formularen werde ich immer nur auf bestimmte Besonderheiten eingehen. Einige Formulare werde ich hier gar nicht besprechen – bitte erkunden Sie deren Funktionen selbst!

Das Formular frmMannschaften

Mannschaftsmitglieder (DK)		Clubmitglieder ohne Mannschaft (DK)	
Schumacher	Britta	Hübner	Klaus
Krause	Carmen	Mayer	Egon
Dreyer	Martina	Meier	Karl
Wegener	Paula		

Abbildung 1.8: Mitglieder können zwischen zwei Listen hin- und hergeschoben werden.

Das Besondere an diesem Formular sind die beiden Listen in der Mitte und rechts unten. Dahinter verbirgt sich folgende Problematik: Der Benutzer möchte am Bildschirm genauso wie im echten Leben sagen können: »Frau Schumacher und Frau Krause sind in der Mannschaft Damen A.« Dazu braucht er eine Liste aller Clubmitglieder, aus der er mittels Mauseklick einzelne auswählen kann.

Diese sollen dann in die Mannschaft übernommen werden. Außerdem sollen natürlich Mitglieder aus Mannschaften austreten können. Genau das leisten die beiden Schaltflächen zwischen den beiden Listen. Wenn Sie ein Mitglied in der rechten Liste anklicken, wird die Schaltfläche mit dem Pfeil nach links aktiv. Durch Anklicken dieser Schaltfläche können Sie das rechts ausgewählte Mitglied in die Mannschaft übernehmen. Analog können Sie links ein Mannschaftsmitglied auswählen und durch Anklicken der Schaltfläche mit dem nach rechts gerichteten Pfeil aus der Mannschaft entfernen.

Bitte beachten Sie, dass in der rechten Liste immer nur Clubmitglieder auftauchen, die noch in keiner Mannschaft sind. Wenn also ein Mitglied die Mannschaft wechseln soll, müssen Sie es zunächst aus der einen Mannschaft entfernen und dann in die andere Mannschaft aufnehmen. Sollten solche Wechsel häufig vorkommen, kann man natürlich auch ein Formular entwickeln, mit dem direkte Wechsel von Mannschaft zu Mannschaft möglich sind.

Auf eine weitere Besonderheit dieses Formulars weist die Abkürzung *DK* (für »Doppelklick«) in der Überschrift der linken Liste hin. Was das bedeutet? Machen Sie doch mal einen Doppelklick auf einen Nachnamen in der Liste! Und schwupp – öffnet sich das bereits gut bekannte *Mitglieder*-Formular.

Zunächst ein paar Worte zu der Frage: »Warum diese Funktion?« Das ist ähnlich wie mit der kleinen Schaltfläche *Beiträge* im *Mitglieder*-Formular: Es ist eine Frage der Benutzerfreundlichkeit. Vielleicht möchte der Benutzer ja beim Zusammenstellen der Mannschaften noch einmal einen Blick auf die Mitgliederdaten werfen. Der Doppelklick erspart ihm dann den Wechsel zum Startformular. Auch hier ist es wieder die Aufgabe des Anwendungsentwicklers, sich in die Denk- und Arbeitsweise des Benutzers hineinzusetzen und ihm durch geschickte Programmierung das Arbeiten mit der Anwendung so bequem wie möglich zu machen!

Der Doppelklick zum Öffnen eines anderen Formulars funktioniert übrigens auch im *Mitglieder*-Formular: Ein Doppelklick auf den Namen des Trainers oder auf den Mannschaftsnamen öffnet das *Trainer*- bzw. das *Mannschaften*-Formular. Hier ist noch eine weitere Besonderheit zu beachten: Sind die entsprechenden Felder leer – hat das Mitglied also keinen Trainer bzw. ist es in keiner Mannschaft –, darf sich natürlich kein anderes Formular öffnen.

Das Formular frmTraining

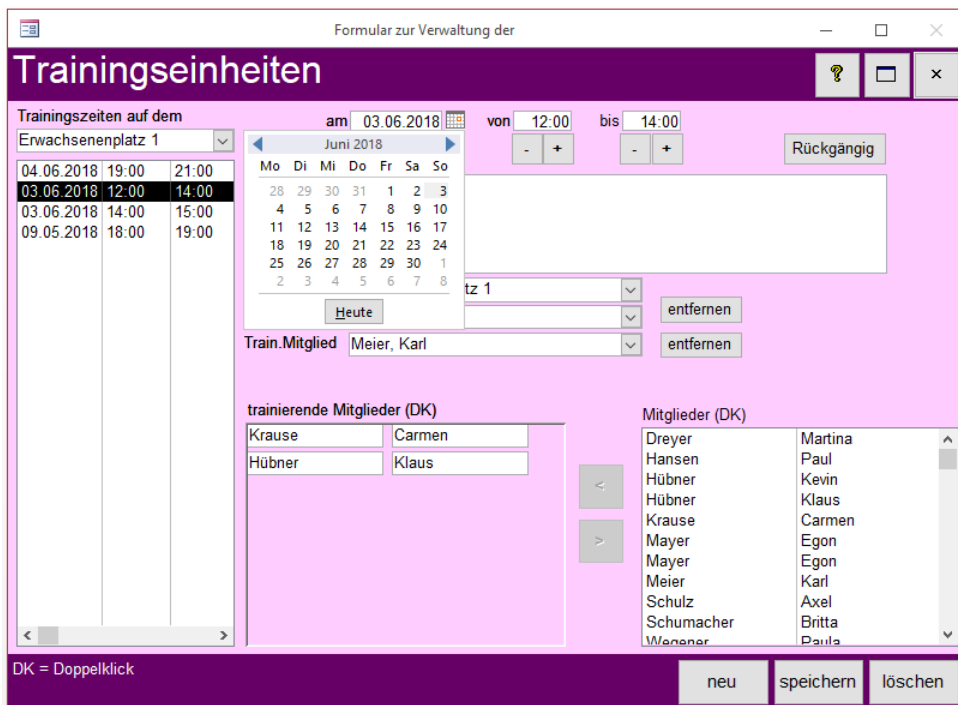


Abbildung 1.9: Zeiten können durch Mausklicks schrittweise geändert werden.

Mit diesem Formular werden Trainingszeiten vergeben. Der Benutzer wählt Datum, Startzeit, Endezeit und Platz sowie gegebenenfalls noch einen Trainer aus und kann diese Kombination dann an mehrere Mitglieder vergeben – mithilfe der gleichen Zwei-Listen-Technik wie im *Mannschaften*-Formular.

Das Besondere an diesem Formular sind zunächst einmal ganz offensichtlich die Plus- und Minusschaltflächen unter den Textfeldern für das Datum und die Zeiten. Sie werden es kaum glauben – aber das ist schon wieder mal eine Sache der Benutzerfreundlichkeit. Datums- und Zeitangaben einzutippen, ist nämlich ziemlich umständlich und vor allen Dingen auch fehleranfällig. Da wollen wir es dem Anwender doch wieder mal etwas bequemer machen. Er kann die genannten Schaltflächen benutzen, um das Datum und die Zeiten, die bereits in den Textfeldern stehen, einfach mit einem Mausklick zu verändern – das Datum um einen Tag, die Uhrzeitangaben um eine Viertelstunde.

Außerdem fällt in Abbildung 1.9 noch ein Detail auf, das Sie zunächst gar nicht sehen – es wird erst sichtbar, wenn Sie in das Textfeld mit dem Trainingsdatum hineinklicken. Dann erscheint nämlich rechts daneben ein kleines Kalendersymbol, und wenn Sie darauf klicken, öffnet sich tatsächlich ein Kalenderblatt (Abbildung 1.9). Ein Klick in dieses Kalenderblatt übernimmt das angeklickte Datum in das Textfeld des Formulars. Dadurch werden Datumseingaben sehr viel bequemer! Sie brauchen dieses Verhalten nicht einmal zu programmieren – sowie Sie ein Textfeld eines Formulars mit dem Format *Datum* versehen, bietet Access automatisch diese Eingabehilfe an.

Vielleicht fragen Sie sich noch, was das Textfeld *Train.Mitglied* bedeutet? Damit können Sie den trainierenden Mitgliedern statt eines Trainers ein weiteres Mitglied zuweisen, damit die Mitglieder sich gegebenenfalls gegenseitig trainieren können. Natürlich können Sie einer Trainingszeit auch einen Trainer und ein Trainermitglied zuweisen.

Mithilfe des Kombinationsfelds *Platz* können Sie eine Trainingszeit auf einen anderen Platz verschieben.

Bitte beachten Sie auch hier wieder, dass die beiden Schaltflächen < und > abwechselnd aktiv werden – je nachdem, in welche der beiden Listen Sie vorher geklickt haben.

Das Formular frmTypen

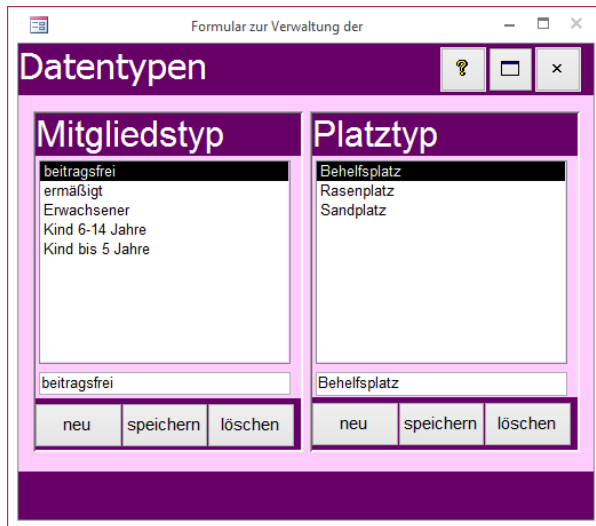


Abbildung 1.10: Datentypen werden in den anderen Formularen benötigt.

Dieses Formular bietet von der Funktionalität her nichts Besonderes – nur die Art der hier verwalteten Daten erscheint mir erwähnenswert. Es handelt sich um Daten, die die Art oder den Typ anderer Daten charakterisieren – im vorliegenden Fall den Mitgliedstyp und den Platztyp. Das sind Daten, die in anderen Formularen in Kombinationsfeldern zur Auswahl bei der Dateneingabe angeboten werden. Der Benutzer kann damit in begrenztem Maße seine Benutzeroberfläche (GUI) selbst gestalten, indem er die Kombinationsfelder mit selbst definierten Daten füllt!

Der Mitgliedstyp ist Ihnen schon im Formular *frmMitglieder* in der Liste der Mitgliedschaften begegnet (bitte noch einmal nachschauen!). Der Platztyp findet im Formular *frmPlaetze* Verwendung. Dort kann der Benutzer neue Plätze anlegen und ihnen einen selbst definierten Platztyp zuweisen. Die im Formular *frmPlaetze* angelegten Plätze finden wiederum im Formular *frmTraining* Verwendung, denn sie füllen dort das Kombinationsfeld *Platz* (bitte nachschauen!).

An dieser Stelle ist der Hinweis auf eine besondere Problematik bei Access angebracht. Dazu machen Sie einmal folgendes Experiment: Öffnen Sie die beiden Formulare *frmTypen* und *frmPlaetze* und ordnen Sie sie nebeneinander auf Ihrem Bildschirm an. Erzeugen Sie mit dem Formular *frmTypen* einen neuen Platztyp (z. B. »Übungsplatz«) und speichern Sie ihn

ab, sodass er in der Liste der verfügbaren Platztypen erscheint. Nun wechseln Sie zum Formular *frmPlaetze* und klappen das Kombinationsfeld *Platztyp* auf. Was sehen Sie? Der neue, gerade von Ihnen kreierte Platztyp taucht in dem Kombinationsfeld auf.

»Klar!«, werden Sie sagen. »Ich habe einen neuen Platztyp angelegt, dann muss er auch da sein!« Aber so selbstverständlich ist das leider gar nicht. Wieso?

Dazu rekapitulieren Sie (Nicht kapitulieren! Oder ist es schon so weit ;-)) bitte noch einmal das, was ich im Abschnitt »Ist Access eine Datenbank?« über die Komponenten einer Datenbankanwendung gesagt habe. Ihr Formular *frmTypen* als Bestandteil der Benutzeroberfläche (GUI) hat jetzt das DBMS angewiesen, den neuen Platztyp in der entsprechenden Tabelle zu speichern. Das andere Formular *frmPlaetze* hat aber schon **vorher** bei seinem Erscheinen auf dem Bildschirm das DBMS angewiesen, alle erforderlichen Daten herauszusuchen, damit sie im Formular dargestellt werden können. Die Änderung der auf dem Bildschirm dargestellten Daten (Inhalt des Kombinationsfelds *Platztyp*) erfolgte also »hinter seinem Rücken!« Jetzt stehen somit in der Tabelle *tblPlatztyp* vier Platztypen – im Kombinationsfeld *Platztyp* des Formulars *frmPlaetze* aber nur drei. Das liegt daran, dass das Formular die darzustellenden Daten nur einmal bei seinem Erscheinen auf dem Bildschirm beim DBMS anfordert – danach nicht wieder; jedenfalls nicht automatisch.

Trotzdem sehen Sie aber im Kombinationsfeld *Platztyp* des Formulars *frmPlaetze* **vier** Platztypen. Damit das funktioniert, müssen Sie als Entwickler einen entsprechenden VBA-Befehl an der richtigen Stelle platzieren. Ich komme später in Kapitel 8, »VBA – Teil 2«, im Abschnitt »Standardlösungen« darauf zurück.

Hilfe!

Auf allen Formularen befindet sich rechts oben eine Schaltfläche zum Öffnen eines Fensters mit Hilfeinformationen. Bestimmt haben Sie das schon ausprobiert!

Das Fenster, das sich dann öffnet, sieht nicht ganz so aus wie die Hilfefenster, die Sie von vielen anderen Softwareprodukten kennen. Deren Fenster können nicht nur Text, sondern auch Abbildungen und die aus dem Internet bekannten Hyperlinks enthalten, d.h., Sie können auf ein markiertes Wort klicken, und eine neue Seite mit weiteren Hilfeinformationen wird angezeigt. Solche komfortablen Hilfefenster sind natürlich sehr schön – erfordern aber zu ihrer Erstellung eine spezielle Software, deren Bedienung Sie erst einmal erlernen müssten.

Ich möchte Ihnen hier etwas Einfacheres anbieten, das Sie nicht erst lernen müssen, sondern sofort und ohne spezielle Software einsetzen können. Die Hilfefenster in meinen Beispielanwendungen zeigen Texte an, die sich in separaten Dateien befinden. Wenn Sie also z. B. im Formular *frmMitglieder* auf *Hilfe* klicken, öffnet sich ein Fenster, das den Inhalt der Datei *hilfe-mitglieder.rtf* anzeigt. Das Format dieser Dateien ist RTF (Rich Text Format). Es erlaubt Formatierungen – wie fett, kursiv, unterstrichen, farbige Texte und Aufzählungen. Solche Dateien können Sie mit Word erstellen, indem Sie beim Befehl *Speichern unter* das RTF-Format auswählen.

Es geht aber auch noch einfacher: Auf jedem Windows-Computer finden Sie als Zubehör das kleine Textbearbeitungsprogramm WordPad. Auch damit können Sie auf sehr einfache Weise *.rtf*-Dateien erstellen.



Die Speicherung der Hilfeinformationen in Textdateien hat einen weiteren Vorteil: Der Benutzer der Datenbankanwendung kann die Hilfedateien selbst bearbeiten und für sich oder andere Hilfeinformationen dort hineinschreiben.

Oft ist es ja so, dass man selbst bestimmte Erfahrungen beim Umgang mit einer Software gesammelt hat. Man hat Tricks und Kniffe herausgefunden, sie sich auf einen Zettel geschrieben und findet im entscheidenden Augenblick den Zettel nicht wieder. Hier können Sie dieses gesammelte Know-how jetzt in die Hilfedateien hineinschreiben und für sich und andere bewahren!

Für diejenigen unter Ihnen, die sich mit der Erstellung von Webseiten (.html-Dateien) auskennen, habe ich noch eine Idee. Öffnen Sie doch einmal das Formular *frmPlaetze*. Dort sehen Sie rechts oben drei statt zwei Schaltflächen. Die ganz linke zeigt Ihnen eine weitere Möglichkeit, dem Benutzer Hilfeinformationen zu präsentieren: über eine Webseite. Ein Klick auf diese Schaltfläche öffnet Ihren Standardbrowser und zeigt die Hilfeseite an.

Das funktioniert aber nur, wenn Ihr Standardbrowser mit Dateien mit der Endung *.html* verknüpft ist. Um diese Verknüpfung gegebenenfalls herzustellen,

1. öffnen Sie den Explorer (nicht den Internet Explorer!),
2. suchen sich irgendeine Datei mit der Endung *.html* (z.B. die Hilfedatei *hilfe-plaetze.html* im Ordner mit den Beispielen für dieses Kapitel),
3. klicken mit der rechten Maustaste auf den Dateinamen und
4. wählen den Menübefehl *Öffnen mit/Standardprogramm auswählen*.
5. Wählen Sie nun den Browser in der Liste der Programme aus. Wenn er dort nicht auftaucht, klicken Sie auf die Schaltfläche *Durchsuchen*.
6. Nachdem Sie den Browser gefunden haben, aktivieren Sie das Kontrollkästchen *Dateityp immer mit dem ausgewählten Programm öffnen*.

Anschließend sollte die Schaltfläche *HTML-Hilfe anzeigen* im Formular *frmPlaetze* funktionieren.

Eine weitere Hilfe sind die Steuerelement-Tipps, die auftauchen, wenn Sie den Mauszeiger einige Sekunden lang über ein Steuerelement halten. Betrachten wir z. B. das Formular *frmMannschaften*: Über der Schaltfläche *Trainer* erscheint der Tipp *Trainerzuweisung entfernen* und über dem Nachnamen in der Liste *Mannschaftsmitglieder* der Tipp *Doppelklick öffnet das Mitgliederformular*. Diese Tipps müssen Sie als Anwendungsentwickler selbst einrichten. Die Tipps *Datensatz hinzufügen*, *Datensatz speichern*, *Datensatz löschen* und *Formular schließen* über den entsprechenden Schaltflächen werden hingegen standardmäßig von Access eingerichtet.

Benutzerfreundlichkeit

In den vorigen Abschnitten wurden immer mal wieder Grundsätze für eine gute Anwendungsentwicklung genannt. Diese sind in der Tabelle auf der folgenden Seite kurz zusammengefasst.