

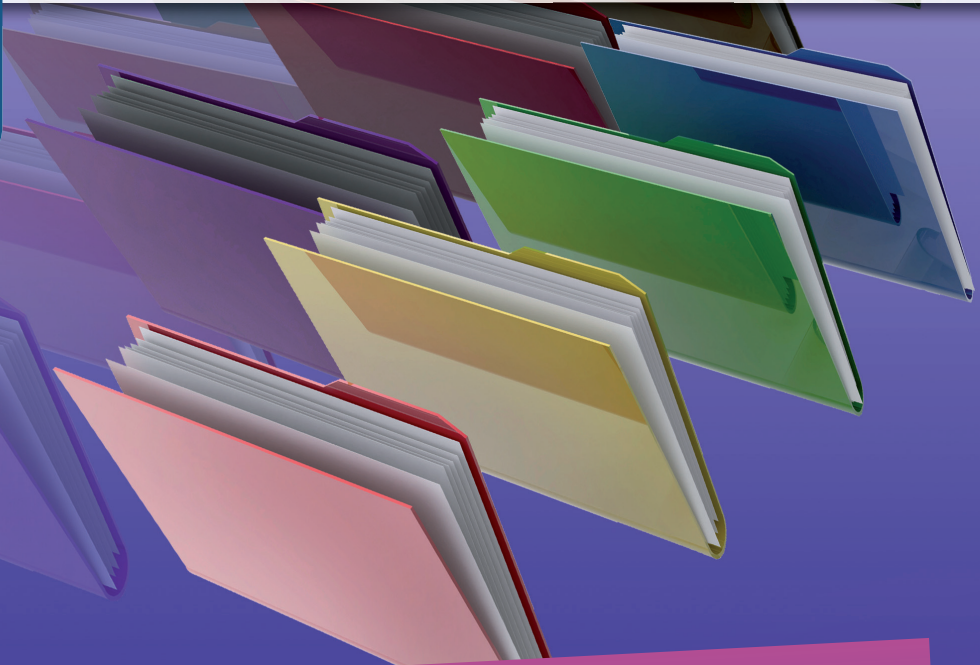
für iOS, OS X
und Windows



Horst-Dieter Radke

FileMaker Pro 12

Das Grundlagenbuch:
Datenbanken entwickeln und verwalten



Horst-Dieter Radke
**FileMaker Pro 13:
Das ist neu**



E-Book-Update zum Buch
kostenfrei downloaden

Filemaker Pro 13 Das ist neu!

Download und Infos:
smartbooks.de/fm13



SMARTBOOKS
Mac und wir.



FileMaker Pro 12

Horst-Dieter Radke

FileMaker Pro 12

**Das Grundlagenbuch: Datenbanken
entwickeln und verwalten**



SMARTBOOKS
Mac und wir.

FileMaker Pro 12

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.

Copyright © 2013 dpunkt.verlag GmbH, Ringstr. 19B, 69115 Heidelberg

ISBN:

Buch 978-3-944165-00-4

PDF 978-3-944165-84-4

ePub 978-3-944165-85-1

1. Auflage 2013

Projektleitung und Lektorat:

Gabriel Neumann

Korrektur:

Dr. Anja Stiller-Reimpell

Layout und Satz:

Peter Murr

Covergestaltung:

Johanna Voss, Florstadt

Coverfoto:

iStock_8570509 und iStock_11021295 (iStockphoto)

Kapitel-Trennseiten:

Seite 3: iStockphoto 11456255 diego_cervo

Seite 15: fotolia 55287 Stasys Eidiejus

Seite 49: iStockphoto 8446828 DNY59

Seite 137: iStockphoto 12138924 alexsl

Seite 215: iStockphoto 12116947 alexsl

Seite 231: fotolia 37475 Tom Mc Nemar

Seite 317: fotolia 58914 Andres Rodriguez

Seite 359: iStockphoto 9741751 alexsl

Seite 13, 33, 85, 109, 181, 279, 345, 377: Horst-Dieter Radke

Druck und Bindung:

M. P. Media-Print Informationstechnologie GmbH,
33100 Paderborn

Trotz sorgfältigem Lektorat schleichen sich manchmal Fehler ein. Autoren und Verlag sind Ihnen dankbar für Anregungen und Hinweise!

SmartBooks

Ein Imprint der dpunkt.verlag GmbH

Ringstr. 19B - 69115 Heidelberg - Deutschland

<http://www.smartbooks.de>

E-Mail: info@smartbooks.de

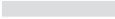
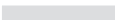
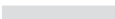
Alle Rechte vorbehalten. Die Verwendung der Texte und Bilder, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und strafbar. Das gilt insbesondere für die Vervielfältigung, Übersetzung, die Verwendung in Kursunterlagen oder elektronischen Systemen. Der Verlag übernimmt keine Haftung für Folgen, die auf unvollständige oder fehlerhafte Angaben in diesem Buch oder auf die Verwendung der mitgelieferten Software zurückzuführen sind. Nahezu alle in diesem Buch behandelten Hard- und Software-Bezeichnungen sind zugleich eingetragene Warenzeichen oder sollten als solche behandelt werden.

Besuchen Sie uns im Internet!

www.smartbooks.de

www.smartbooks.ch

Übersicht

	Vorwort	13
Kapitel 1	Datenbankgrundlagen	15
Kapitel 2	Mit FileMaker arbeiten ohne Vorkenntnisse	33
Kapitel 3	Datenbankentwicklung	49
Kapitel 4	Datenbanken nutzen	85
Kapitel 5	Suchen und Finden	109
Kapitel 6	Felder, Formeln und Funktionen	137
Kapitel 7	Layouten für die Datenbank	181
Kapitel 8	Berichte erstellen	215
Kapitel 9	Einführung in die Scriptprogrammierung	231
Kapitel 10	FileMaker Pro 12 kommunikativ und sicher	279
Kapitel 11	Tipps, Tricks & Troubleshooting	317
Kapitel 12	FileMaker mobil	345
	Glossar	359
	Index	377

Inhaltsverzeichnis

Vorwort

13

Kapitel 1 Datenbankgrundlagen

15

1.1	Datenbankgrundlagen	16
1.1.1	Was ist eine Datenbank?	16
1.1.2	Das relationale Modell	17
1.1.3	SQL – oder die Suche nach der letzten Information	18
1.1.4	Datenbanksysteme	19
1.1.5	Probleme der Informationsspeicherung	20
1.2	Datenbankkonzeption	20
1.2.1	Datenmodellierung	21
1.2.2	Eine Datenbank in fünf Schritten	21
1.2.3	Projektbeschreibung	21
1.2.4	Das Vermeiden von Mehrfachspeicherungen	22
	Ausgangssituation festlegen	22
	Die erste Normalform – Tabellen werden zerlegt	23
	Die zweite Normalform – doppelte Informationen entfernen	23
1.2.5	Schlüssel und Index	25
	Datensätze eindeutig identifizieren	25
	Auf den Index gesetzt	26
	Beziehungen pflegen	27
	Die monogame Beziehung	27
	Die offene Beziehung	27
	Jeder mit Jedem	28
	Beziehung zu sich selbst	28
1.3	FileMaker Pro als relationales Datenbank-Managementsystem	29
1.4	FileMaker Pro und andere Datenbank-Managementsysteme	30

Kapitel 2 Mit FileMaker arbeiten ohne Vorkenntnisse

33

2.1	FileMaker-Starter-Lösungen	34
2.1.1	FileMaker bedienen	34
	Installation	34
	Nach dem Start	35
	Werkzeuge	36
2.1.2	Die Starter-Lösungen	37
	Starter-Lösung einrichten	39
	Starter-Lösungen verstehen	40
2.2	Arbeiten mit einer FileMaker-Datenbank	42
2.2.1	Daten erfassen	42
2.2.2	Daten finden	44
2.2.3	Daten sichern und löschen	46

Kapitel 3 Datenbankentwicklung 49

3.1	Datenbankdesign mit FileMaker Pro	50
3.2	Entwicklungsmethode für überschaubare Projekte	50
3.2.1	Felder festlegen und Redundanzen erkennen	51
3.2.2	Tabellen und Felder anlegen	52
3.2.3	Eine weitere Tabelle anlegen	53
3.2.4	Felder anlegen	54
3.2.5	Datentyp festlegen und ändern	55
3.2.6	Werteliste erstellen	55
3.2.7	Eine Beziehung herstellen	57
3.2.8	Formulare erstellen	59
	Mit dem Layoutassistenten arbeiten	59
	Das Layout anpassen	62
	Datenausschnitt in einem Layout	65
3.3	Datenbankprogrammierung	69
3.3.1	Skriptprogrammierung für Einsteiger	69
3.3.2	Formeln für Einsteiger	71
	Eine Formel erstellen	72
3.3.3	Daten importieren	74
	Datenbasis besorgen	74
	Daten importieren	76
3.3.4	Der Web Viewer	79
	Das Objekt Web Viewer	79
	Scripteditor	80
3.4	Fazit	82

Kapitel 4 Datenbanken nutzen 85

4.1	FileMaker Pro: Grundlagen	86
4.2	FileMaker Pro bedienen	86
4.2.1	FileMaker Pro starten und bedienen	86
	FileMaker-Direktstart	86
	Menü- und Statusleisten	89
4.2.2	Bestehende Dateien konvertieren	92
	FileMaker-eigene Formate konvertieren	92
	Fremde Formate konvertieren	93
4.2.3	Starter-Lösungen verwenden und anpassen	96
	Datenbank aus Vorlage erstellen	97
	Datenbankvorlage anpassen	98
4.3	Mit Datenbanken arbeiten	103
	Datensatz anlegen und Daten eingeben	104

Kapitel 5 Suchen und Finden **109**

5.1	Daten importieren	110
5.1.1	Importieren statt Eingeben	110
5.1.2	Bankleitzahlen herunterladen	110
	Datenbank vorbereiten	112
5.1.3	Bankleitzahlendatenbank einrichten	112
5.1.4	Datenimport durchführen	116
5.1.5	Aktualisierungen durch neuen Import	119
5.2	Suchen in der Datenbank	121
5.2.1	Eine Suchabfrage definieren	122
5.2.2	Fast Match – die Suche per Mausklick	124
5.2.3	Mit Operatoren suchen	126
5.2.4	Ansichten der Ergebnismengen	129
5.2.5	Mehrere Kriterien bei der Suche einsetzen	131
	Zwei Abfragemethoden	132
	Ergebnismenge exportieren	133
	Ergebnismengen aus der Datenbank löschen	134

Kapitel 6 Felder, Formeln und Funktionen **137**

6.1	Felder definieren	138
6.1.1	Feldtypen	139
	Feldtyp Text	140
	Feldtyp Zahl	142
	Feldtyp Datum	143
	Feldtyp Zeit	145
	Feldtyp Zeitstempel	145
	Feldtyp Container	146
	Feldtyp Formel	149
	Feldtyp Statistik	149
6.1.2	Ergänzendes zu Felddefinitionen	152
	Besonderheiten bei Feldnamen	152
	Feldoptionen einstellen	153
	Indizierung	156
6.1.3	Formeln erstellen	160
	Arbeiten mit dem Formeleditor	160
	Mit Funktionen arbeiten	164
6.2	Formeln und Funktionen für Fortgeschrittene	169
6.2.1	Die Funktion Falls	169
6.2.2	Texte in Formeln	172
6.2.3	Informationen aus der Datenbank mit Statusinformationen	174
6.2.4	Containerfunktionen	176
6.2.5	Neue und geänderte Funktionen in FileMaker Pro 12	179

Kapitel 7	Layouten für die Datenbank	181
7.1	Der Layout-Assistent	182
7.2	Ein Layout zusammenstellen	182
	Layout-Assistent starten	183
	Layouttypen	184
	Felder auswählen	185
	Design wählen	186
7.3	Ein Standard-Layout anpassen	188
	Kopfbereich gestalten	189
	Fußbereich gestalten	194
	Objekte und Variablen einfügen	195
	Layouteinstellungen anpassen	198
	Tabulatorfolge festlegen	203
	Script-Trigger einstellen	204
	Grafische Gestaltungselemente	208
	Ausgabefelder formatieren	210
Kapitel 8	Berichte erstellen	215
8.1	Berichte	216
8.2	Mit dem Berichtsassistenten arbeiten	216
	Berichterstellung vorbereiten	216
	Die Berichteinstellungen festlegen	218
	Bericht ausgeben	222
	Berichte bearbeiten	223
8.3	Dynamische Berichte erstellen	225
8.4	Snapshot-Link	228
Kapitel 9	Einführung in die Scriptprogrammierung	231
9.1	Einführung in die Scriptprogrammierung	232
9.2	Ein erstes Script	233
9.2.1	Ein Script erstellen	233
	Einen Scriptbefehl verstehen	234
	Ein Script an ein Layout binden	236
	Scripteintrag aus dem Menü entfernen	237
9.2.2	Ein mehrzeiliges Script entwickeln	238
9.2.3	Ablaufsteuerung im Script	248
	StartScript erstellen	248
	Startscript zuweisen	251
9.2.4	Eine Schleife programmieren	252
9.2.5	Eine Meldung ausgeben	255
9.3	Einige wichtige Scriptschritte	256
	Aktuellen Datensatz prüfen	257

	Aktuelles Script verlassen	262
	Alle Datensätze löschen	263
	Alles auswählen	264
	Datei öffnen	264
	Datei schließen	265
	Datensätze zeigen als	265
	Einfügen	265
	Ergebnismenge suchen	266
	Ersetze alle Feldwerte	266
	Feldwert setzen	268
	Kopie speichern unter	269
	Nächste fortlaufende Nummer setzen	271
	Neues Fenster	273
	Programm beenden	273
9.4	FileMaker Pro Advanced	274
	Werkzeuge	274
	Runtime-Lösungen	274
	Weitere Advanced-Funktionen	277
9.5	Fazit	277

Kapitel 10 FileMaker Pro 12 kommunikativ und sicher 279

10.1	Sicherheit mit FileMaker	280
10.2	Datenbanken schützen	280
10.2.1	Konten einrichten	281
10.2.2	Eigene Berechtigungen definieren	283
10.2.3	Kontonamen und Passwörter	288
10.2.4	Duplizieren und Löschen von Konten	290
10.3	FileMaker Pro 12 im Netzwerk	291
10.3.1	Voraussetzungen für den Einsatz von FileMaker Pro im Netzwerk	291
	Windows- oder OS-X-Netzwerk	291
	Netzwerk Client und Host	292
10.3.2	FileMaker Pro 12 für das Netzwerk vorbereiten	294
10.3.3	Arbeiten mit einer FileMaker-Datenbank im Netzwerk	297
	Auf eine Netzwerkdatenbank vom Client aus zugreifen	298
	Gemeinsamer Zugriff auf einen Datensatz	300
	Meldungen an andere Datenbanknutzer verschicken	301
	Schließen einer Datenbank auf dem Host	302
	Probleme im Datenbanknetzwerk	302
10.4	FileMaker Pro 12 und das Internet	303
10.4.1	Ins Internet mit FileMaker Pro	303
10.4.2	Eine Datenbank für das Web-Publishing vorbereiten	304
10.4.3	Web-Datenbank austesten	306
10.4.4	Beenden einer Internet-Sitzung	310

10.5	Schnittstellen	312
10.5.1	Export und Importformate	312
	Importformate	312
	Datenexport	313
10.5.2	ODBC und JDBC	314
Kapitel 11	Tipps, Tricks & Troubleshooting	317
11.1	Beziehungen bearbeiten	318
11.1.1	Self-Join-Beziehung aufbauen und nutzen	318
11.1.2	Beziehungen bearbeiten	323
11.2	Diagramme mit FileMaker Pro erstellen	328
11.2.1	Eine schnelle Tabelle anlegen	328
11.2.2	Das Diagramm erstellen	329
11.2.3	Das Diagramm bearbeiten	330
11.3	Troubleshooting	332
11.3.1	Beschädigte Daten	332
11.3.2	Beschädigte Datei wiederherstellen	332
11.3.3	Einen beschädigten Datensatz retten	335
11.4	Tipps & Tricks	336
11.4.1	Dublettensuche	336
11.4.2	Erweiterte Menüs (OS X)	337
11.4.3	Feld duplizieren	337
11.4.4	Felder beim Duplizieren nicht füllen	337
11.4.5	Lästige Abfrage	338
11.4.6	Merkwürdige Zeilenumbrüche	339
11.4.7	Objekte und Felder kopieren	340
11.4.8	Tabulatorsprünge in einem Textfeld	340
11.4.9	Wertelisten mit zwei Feldern füllen	341
Kapitel 12	FileMaker mobil	345
12.1	Von FileMaker Mobile zu FileMaker Go	346
12.2	FileMaker Go benutzen	347
12.2.1	Installation und Einrichtung	347
12.2.2	Datenbanken übertragen und synchronisieren	349
12.2.3	FileMaker-Datenbanken entwickeln	350
12.2.4	Gemeinsam Datenbestände verwenden	354
Glossar		359
Index		377

The photograph shows a wall densely packed with various items pinned to it. The items include:

- Handwritten notes:**
 - Top left: "Saverio della Maria e della (St. Maria della...)"
 - Top right: "DO PEOPLE OF ALL RACES AND ALL PARTS OF THE WORLD... THE QUESTION AND THE ANSWER... NOW THERE CAN... AND THE QUESTION... BEING..."
 - Center: "K.K. Kelly beautiful lady. You're the only girl that I adore. When she sees Simon Wilson over the one about I'll be making at the kitchen door."
 - Below center: "AIR KINGSCHEN ZAHLREICHE GLÜCKSTERNE"
 - Bottom left: "ho il mono dei tanti paesi dentro il cuore."
 - Bottom center: "di eros sono felice e amato"
 - Right side: "20.000... 100.000... 200.000... 300.000... 400.000... 500.000... 600.000... 700.000... 800.000... 900.000... 1.000.000... 1.100.000... 1.200.000... 1.300.000... 1.400.000... 1.500.000... 1.600.000... 1.700.000... 1.800.000... 1.900.000... 2.000.000... 2.100.000... 2.200.000... 2.300.000... 2.400.000... 2.500.000... 2.600.000... 2.700.000... 2.800.000... 2.900.000... 3.000.000... 3.100.000... 3.200.000... 3.300.000... 3.400.000... 3.500.000... 3.600.000... 3.700.000... 3.800.000... 3.900.000... 4.000.000... 4.100.000... 4.200.000... 4.300.000... 4.400.000... 4.500.000... 4.600.000... 4.700.000... 4.800.000... 4.900.000... 5.000.000... 5.100.000... 5.200.000... 5.300.000... 5.400.000... 5.500.000... 5.600.000... 5.700.000... 5.800.000... 5.900.000... 6.000.000... 6.100.000... 6.200.000... 6.300.000... 6.400.000... 6.500.000... 6.600.000... 6.700.000... 6.800.000... 6.900.000... 7.000.000... 7.100.000... 7.200.000... 7.300.000... 7.400.000... 7.500.000... 7.600.000... 7.700.000... 7.800.000... 7.900.000... 8.000.000... 8.100.000... 8.200.000... 8.300.000... 8.400.000... 8.500.000... 8.600.000... 8.700.000... 8.800.000... 8.900.000... 9.000.000... 9.100.000... 9.200.000... 9.300.000... 9.400.000... 9.500.000... 9.600.000... 9.700.000... 9.800.000... 9.900.000... 10.000.000... 10.100.000... 10.200.000... 10.300.000... 10.400.000... 10.500.000... 10.600.000... 10.700.000... 10.800.000... 10.900.000... 11.000.000... 11.100.000... 11.200.000... 11.300.000... 11.400.000... 11.500.000... 11.600.000... 11.700.000... 11.800.000... 11.900.000... 12.000.000... 12.100.000... 12.200.000... 12.300.000... 12.400.000... 12.500.000... 12.600.000... 12.700.000... 12.800.000... 12.900.000... 13.000.000... 13.100.000... 13.200.000... 13.300.000... 13.400.000... 13.500.000... 13.600.000... 13.700.000... 13.800.000... 13.900.000... 14.000.000... 14.100.000... 14.200.000... 14.300.000... 14.400.000... 14.500.000... 14.600.000... 14.700.000... 14.800.000... 14.900.000... 15.000.000... 15.100.000... 15.200.000... 15.300.000... 15.400.000... 15.500.000... 15.600.000... 15.700.000... 15.800.000... 15.900.000... 16.000.000... 16.100.000... 16.200.000... 16.300.000... 16.400.000... 16.500.000... 16.600.000... 16.700.000... 16.800.000... 16.900.000... 17.000.000... 17.100.000... 17.200.000... 17.300.000... 17.400.000... 17.500.000... 17.600.000... 17.700.000... 17.800.000... 17.900.000... 18.000.000... 18.100.000... 18.200.000... 18.300.000... 18.400.000... 18.500.000... 18.600.000... 18.700.000... 18.800.000... 18.900.000... 19.000.000... 19.100.000... 19.200.000... 19.300.000... 19.400.000... 19.500.000... 19.600.000... 19.700.000... 19.800.000... 19.900.000... 20.000.000... 20.100.000... 20.200.000... 20.300.000... 20.400.000... 20.500.000... 20.600.000... 20.700.000... 20.800.000... 20.900.000... 21.000.000... 21.100.000... 21.200.000... 21.300.000... 21.400.000... 21.500.000... 21.600.000... 21.700.000... 21.800.000... 21.900.000... 22.000.000... 22.100.000... 22.200.000... 22.300.000... 22.400.000... 22.500.000... 22.600.000... 22.700.000... 22.800.000... 22.900.000... 23.000.000... 23.100.000... 23.200.000... 23.300.000... 23.400.000... 23.500.000... 23.600.000... 23.700.000... 23.800.000... 23.900.000... 24.000.000... 24.100.000... 24.200.000... 24.300.000... 24.400.000... 24.500.000... 24.600.000... 24.700.000... 24.800.000... 24.900.000... 25.000.000... 25.100.000... 25.200.000... 25.300.000... 25.400.000... 25.500.000... 25.600.000... 25.700.000... 25.800.000... 25.900.000... 26.000.000... 26.100.000... 26.200.000... 26.300.000... 26.400.000... 26.500.000... 26.600.000... 26.700.000... 26.800.000... 26.900.000... 27.000.000... 27.100.000... 27.200.000... 27.300.000... 27.400.000... 27.500.000... 27.600.000... 27.700.000... 27.800.000... 27.900.000... 28.000.000... 28.100.000... 28.200.000... 28.300.000... 28.400.000... 28.500.000... 28.600.000... 28.700.000... 28.800.000... 28.900.000... 29.000.000... 29.100.000... 29.200.000... 29.300.000... 29.400.000... 29.500.000... 29.600.000... 29.700.000... 29.800.000... 29.900.000... 30.000.000... 30.100.000... 30.200.000... 30.300.000... 30.400.000... 30.500.000... 30.600.000... 30.700.000... 30.800.000... 30.900.000... 31.000.000... 31.100.000... 31.200.000... 31.300.000... 31.400.000... 31.500.000... 31.600.000... 31.700.000... 31.800.000... 31.900.000... 32.000.000... 32.100.000... 32.200.000... 32.300.000... 32.400.000... 32.500.000... 32.600.000... 32.700.0

Vorwort

Das Vorwort für ein Buch über ein Datenbankmanagementsystem zu schreiben ist eine undankbare Sache. Erkläre ich nun, was eine Datenbank ist? Oder gehört das nicht schon in den Hauptteil des Buches? Ich kann ja nicht davon ausgehen, dass das Wissen über Datenbanken schon so bekannt ist wie beispielsweise die Funktion einer Textverarbeitung. Andererseits irritiere ich dann diejenigen, die mit voller Kenntnis der Thematik zu diesem Buch greifen, weil sie von einer anderen Datenbank kommen und nun einen Einstieg in FileMaker Pro suchen. Sicher ist es da besser, ich halte das Vorwort kurz und lasse die Themen dort, wo sie hingehören: in den Hauptteil des Buches. Der geneigte Leser wird sie dort schon finden.

Dieses Buch hat einige Vorgänger. Seit der Version FileMaker Pro 7 sind vier Bücher erschienen, die diese Datenbanksoftware bis zur Version 11 begleitet haben. Jedes Buch hat davon profitiert, dass Leser dem Autor Rückmeldungen gegeben haben. Das Konzept ist im Großen und Ganzen gleich geblieben – die Vorgehensweise so zu erklären, dass sie praktisch nachvollzogen werden kann. Im Detail hat sich von Buch zu Buch manches geändert. Außerdem ist es mir wichtig, die theoretischen und grundlegenden Zusammenhänge zu Datenbanken nicht nur mit dem Blick auf FileMaker zu betrachten, sondern auf allgemeiner Ebene zu behandeln, ohne dabei zu theorielastig zu werden.

Das Buch ist als Lern- und Arbeitsbuch aufgebaut, das heißt, die Informationen sind so aufbereitet, dass ein systematisches Lernen, aber auch ein gezieltes Nachschlagen möglich ist. Falls Ihnen etwas nicht oder im Gegenteil besonders gut gefällt, falls Sie meinen, dass etwas überflüssig ist oder fehlt, dann schreiben Sie mir eine Mail (radke@smartbooks.de). Ich bemühe mich, auf jede Anfrage zu reagieren, zumindest ist in der Vergangenheit kaum eine Anfrage unbeantwortet geblieben. Es gibt auch Leser, die sich zu jeder neuen Ausgabe melden, was mich freut.

Horst-Dieter Radke

Lauda, im April 2013

Datenbankgrundlagen



Dass eine Datenbank nichts ist, auf das man sich setzen kann, weiß jeder, der dieses Buch zur Hand nimmt. Damit ist aber bei vielen schon fast das Ende der Informationen darüber, worum es sich bei diesem Begriff handelt, erreicht. Allerdings sind zwei Gruppen zu unterscheiden: die eine, die es mit griechischen Philosophen Sokrates hält und »weiß, dass sie nichts weiß«. Und die andere, die meint, alles über das Thema Datenbanken zu wissen. Die zweite Gruppe hätte aber niemals dieses Buch zur Hand genommen und wenn doch, dann das erste Kapitel gleich überschlagen. Sie, der Sie es lesen, gehören also zur ersten Gruppe und Sie tun gut daran, bevor Sie mit der Arbeit mit FileMaker beginnen, sich über Grundlegendes bei Datenbanken, auch über FileMaker hinaus, zu informieren. Das Kapitel dient diesem Zweck.

1.1 Datenbankgrundlagen

Wenn Ihnen heute noch jemand den Vergleich mit dem Karteikasten liefert, um eine Datenbank zu beschreiben, dann sollten Sie skeptisch und vorsichtig sein. Ein Karteikasten ist ein lineares System, das nur die Anordnung der Datensätze (Karteikarten) hintereinander zulässt. Der Zugriff von einer Karteikarte auf eine andere Karteikarte ist nur umständlich möglich und auch nur dann, wenn direkte Verweise auf der Karte angebracht sind, die mit dem Ordnungssystem des Kastens übereinstimmen. Je umfangreicher solch eine Karteikartensammlung wird, um so schwieriger wird das Pflegen dieses Verweissystems und der Zugriff auf bestimmte Datensätze, wenn die Informationen nicht direkt im Ordnungsbegriff verankert sind.

In einer Datenbank ist es aber möglich, direkt von einem Datensatz auf einen weiteren zu verweisen. Auch der Zugriff auf bestimmte Datensätze ist auf vielfache Weise möglich. Deshalb schließt sich der Verweis auf den Karteikasten eigentlich aus, wenn man von (elektronischen) Datenbanken spricht.

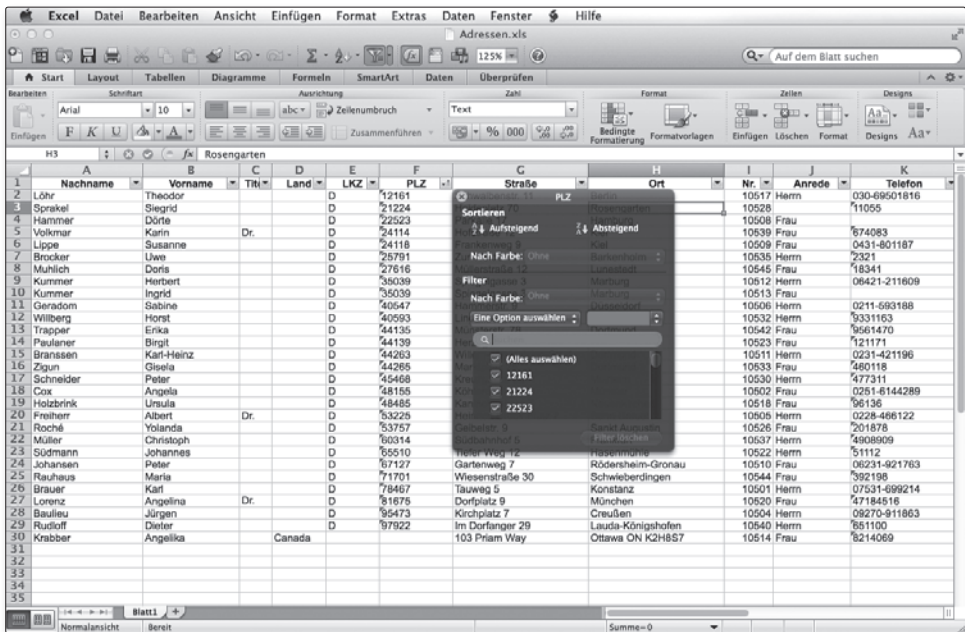
1.1.1 Was ist eine Datenbank?

Als »Datenbank« bezeichnet man ein System, mit dem umfangreiche Informationen gesammelt und verwaltet werden können. Besteht diese Datenbank aus mehreren Tabellen, zwischen denen Verknüpfungen (Beziehungen) hergestellt werden können, so spricht man von einer »relationalen Datenbank«. Nicht selten wird schon eine einfache Tabelle, wie man sie zum Beispiel aus einer Tabellenkalkulation kennt, als Datenbank bezeichnet. Das ist allerdings nicht korrekt, auch wenn man in einer einzelnen Tabelle Datensätze sinnvoll anordnen kann. Zwar entspricht die Grundstruktur – Zeilen und Spalten nehmen an den Schnittstellen (den Feldern) die Informationen auf – durchaus einer relationalen Datenbank. Da aber keine echten Beziehungen zwischen mehreren Tabellen hergestellt werden können, stimmt der Vergleich nicht.

Eine etwas vereinfachte Definition für eine Datenbank ist folgende:

GRUNDLAGEN

Eine Datenbank – oder genauer ein Datenbanksystem – speichert große Datenmengen dauerhaft und stellt sie Benutzern und Anwendungsprogrammen zur weiteren Verarbeitung zur Verfügung.



Eine Tabellenkalkulation wie Excel kann auch Daten verwalten, ist aber trotzdem keine vollwertige Datenbank.

1.1.2 Das relationale Modell

Das wohl bekannteste Datenbankmodell ist das relationale. Es wurde in den sechziger und siebziger Jahren des vergangenen Jahrhunderts bei IBM entwickelt und zwar von Edgar F. Codd (1923 – 2003).

GRUNDLAGEN

Ein relationales Datenbanksystem dient zur Darstellung der vielfältigen Beziehungen zwischen den Daten und nicht nur einfach zur Aufbewahrung einzelner Informationen. Die Tabelle ist die Grundlage einer relationalen Datenbank. In ihr werden die Informationen (die Daten) abgelegt und aufbewahrt. Eine relationale Datenbank enthält in der Regel nicht nur eine, sondern mehrere Tabellen, zwischen denen Beziehungen bestehen. Ziel ist es, Daten mit möglichst geringer oder ganz ohne Redundanz zu speichern. Mit Redundanz ist gemeint, dass möglichst wenige Daten mehrfach abgespeichert werden sollen. Deshalb wird beim relationalen Modell eine Methode benutzt, die helfen soll, Mehrfachspeicherung zu vermeiden. Tabellen werden dabei in mehreren Schritten zerlegt. Diese Methode nennt man »Normalisierung«. Ich komme im Laufe dieses Kapitels noch darauf zurück.

Dr. Codd erhielt 1981 für seine Arbeiten auf dem Gebiet der Datenbank den Turing-Award, eine der höchsten Auszeichnungen in der Informatik.

1.1.3 SQL – oder die Suche nach der letzten Information

Es genügt nicht, dass Informationen in einer Datenbank gesammelt sind, man muss sie auch auswerten können. Je größer die Informationssammlung ist, um so schwieriger wird das Heraussuchen gewünschter Informationen. Deshalb wurden schon früh Such- und Abfragemethoden für Datenbanksysteme entwickelt.

In den siebziger Jahren entwickelte IBM eine Abfragesprache für Datenbanken unter der Bezeichnung SEQUEL (Structured English Query Language). SEQUEL war der Vorläufer von SQL (Structured Query Language). Diese Sprache wurde schnell vom Markt aufgegriffen, führte aber auch zu individuellen »Anpassungen«, sodass bereits 1982 ein Komitee eine ANSI-Norm für SQL zu erarbeiten begann. 1986 wurde die erste offizielle Norm verabschiedet und ein Jahr später die Übernahme des Standards als ISO-Norm festgelegt. Eine zweite Normierung erfolgte 1992 (SQL 2 oder SQL 92). Aktueller Stand der Sprachnorm ist SQL 99 bzw. SQL3.

Die meisten Standardanwendungen sind heute mit SQL-Schnittstellen oder Modulen versehen. Selbst eine Tabellenkalkulation wie Excel kann damit heute aufwarten.

GRUNDLAGEN

FileMaker Pro kann inzwischen Echtzeitverbindungen zu SQL-Datenbanken (Microsoft SQL Server, Oracle- oder MySQL-Datenquellen) herstellen. FileMaker ist deshalb nicht mehr nur eine Insellösung, sondern kann in vorhandene Datenbankstrukturen so eingebunden werden, dass Daten genutzt und verändert werden können.

Relationale Datenbanken arbeiten mit Tabellen. In diesen Tabellen werden die klassischen relationalen Operationen der Relationen-Algebra – also Projektion, natürlicher Verbund und Selektion – eingesetzt. Trotzdem ist das Ganze keine Sache ausschließlich für Mathematiker. Wenn Sie noch nie etwas von dieser Art Algebra gehört haben und auch mit den genannten Begriffen (Projektion, Verbund, Selektion) nicht vertraut sind, verstehen Sie die Abfragesprache trotzdem leicht. Sie besteht aus Symbolwörtern wie z. B. SELECT, UPDATE, DELETE, INSERT, die meistens selbsterklärend sind. Man kann den Umgang und die Ausnutzung dieses Systems somit schnell erlernen. FileMaker-Pro-Anwender werden auf diese Sprache zunächst gar nicht zurückgreifen müssen, da die eigenen Such- und Selektionsoptionen so leistungsfähig und dabei noch so bequem anzuwenden sind, dass eine Abfragesprache in der Art von SQL völlig unangemessen erscheint. Interessant wird es dann, wenn mit FileMaker fremde SQL-Datenbanken ausgewertet werden sollen. Aber das ist dann eher ein Thema für den fortgeschrittenen FileMaker-Anwender und -Entwickler.

Wir werden uns in diesem Grundlagenbuch nicht weiter mit dem Thema SQL beschäftigen. Dafür werden die Such- und Filterfähigkeiten von FileMaker Pro 12 genauer ausgeleuchtet.

1.1.4 Datenbanksysteme

Trotz der einfachen Struktur der relationalen Datenbanken, die aus zwei oder mehreren Datentabellen bestehen, zeigt sich bei genauerer Betrachtung, welche Leistungsfähigkeit darin steckt: Die tabellarisch aufgebauten Daten kann man als festes Set in einem Schritt bearbeiten. Der Anwender braucht sich um die physikalischen Details, also darum, an welchen Speicherstellen die Daten abgelegt sind, nicht mehr zu kümmern. Es muss selbst für komplexe Informationsabfragen kein langer und fehlerträchtiger Programmcode geschrieben werden.

GRUNDLAGEN

Beziehungen zwischen Datensätzen werden durch eine Konstruktion bestimmt, die man »Datensets« (oder einfach »Set«) nennt. Im einfachsten Fall besteht jedes Set aus zwei verschiedenen Satzarten.

Obwohl die relationalen Datenbanksysteme dominieren, sind die hierarchischen Datenbanken, mit denen alles angefangen hat, längst nicht verschwunden. Ein hierarchisches Datenbankmodell bildet die reale Welt durch eine Baumstruktur ab. Jeder Satz (Record) hat also genau einen Vorgänger, mit Ausnahme genau eines Satzes, nämlich der Wurzel der so entstehenden Baumstruktur. Diese alten Datenbanksysteme haben einen Vorteil: Aufgrund ausgefeilter Suchsysteme sind Suchvorgänge sehr schnell. Deshalb werden diese hierarchischen Datenbankstrukturen gern bei der Indexsuche oder in Mailprogrammen (z. B. bei der Namensverwaltung) eingesetzt. Da aber Redundanzen nur schwer zu vermeiden sind und jede Änderung neu programmiert werden muss, ist die Einsatzmöglichkeit dieser hierarchischen Datenbanksysteme inzwischen nur noch selten sinnvoll.

Eine neue Entwicklung sind die objektorientierten Datenbankenmodelle. Dabei gibt es zwei Entwicklungslinien: Die eine versucht die relationalen Datenbanken in Richtung »Objektorientierung« zu erweitern (objektrelationale Datenbanken), die andere geht von den objektorientierten Programmiersprachen aus und bemüht sich um eine Neuentwicklung.

GRUNDLAGEN

Auch bei FileMaker ist von Objekten die Rede. So sind zum Beispiel Felder Objekte und Text- und Grafikobjekte können in ein Layout eingefügt werden. Dies entspricht allerdings nicht einer objektorientierten Datenbank, da sich diese »Objekte« in FileMaker nicht auf die Daten (Informationen die gespeichert werden), sondern lediglich auf das Layout (die Ansicht der Daten) beziehen.

Datenbanken lassen sich aber auch in lokale und Client/Server-Datenbanken unterteilen. Eine lokale Datenbank ist nur auf einem Arbeitsplatz und in der Regel auch nur für einen

Benutzer vorhanden. Eine lokale Datenbank kann aber auch auf einem Server liegen und den Zugriff verschiedener Benutzer gleichzeitig zulassen. Dabei werden allerdings immer komplette Tabellen im Netz übertragen. Effektiver ist eine Client/Server-Datenbank. Bei dieser wird ebenfalls die Abfrage verschiedener Benutzer zugelassen – übertragen wird aber immer nur das Abfrageergebnis an den Client. Die Netzbelastung ist erheblich geringer.

Die Standardversion FileMaker Pro 12 ist bereits in Netzwerken einsetzbar. Bis zu neun Anwender können auf eine Serverdatenbank zugreifen. Sollen Datenbanken für mehr Arbeitsplatzrechner zur Verfügung gestellt werden, ist allerdings die FileMaker-Server-Version nötig, die in zwei Ausführungen zur Verfügung steht. FileMaker Server 12 ermöglicht den Zugriff von 250 FileMaker-Pro-Clients. Bei FileMaker Server 12 Advanced ist die Menge der zugreifenden Clients nicht limitiert.

Die Server-Version ist nicht Thema dieses Buches. Weitere Informationen zum Einsatz von FileMaker Pro 12 in Netzwerken finden Sie im Kapitel »FileMaker kommunikativ«.

1.1.5 Probleme der Informationsspeicherung

Eine Datenbank ist mithilfe moderner Datenbank-Managementsysteme (DBMS) wie FileMaker Pro 12 schnell eingerichtet. Eine oder mehrere Tabellen sind in kürzester Zeit konzipiert, und Daten können erfasst werden. Je mehr Daten allerdings in solch einer Datenbank enthalten sind, desto länger müssen Sie suchen, um bestimmte Informationen zu finden. Darüber hinaus kann es vorkommen, dass Informationen doppelt oder mehrfach vorliegen. Die Aktualisierung solcher (redundanten) Daten ist nicht unproblematisch.

Verwenden Sie beispielsweise eine Adressdatenbank, so können in verschiedenen Datensätzen gleiche Orte vorkommen. Kritisch kann es werden, wenn wesentliche Details verändert werden.

Diese Überlegungen machen deutlich, dass ein Minimum an Planung nötig ist, wenn eine neue Datenbank angelegt werden soll.

1.2 Datenbankkonzeption

Anstoß zur Entwicklung einer Datenbank gibt in der Regel ein Problem, das einer Lösung bedarf. Informationen, die bereitgehalten, ausgewertet, ergänzt und verändert werden sollen, stehen dabei meist im Mittelpunkt. Die Problemlösung selbst geschieht nicht mit dem Datenbank-Managementsystem – etwa FileMaker Pro 12 – sondern auf dem Papier. Ein Stapel Konzeptpapier und Schreibgeräte in greifbarer Nähe sind deshalb für Datenbankentwickler kein veraltetes Requisit. Selbstverständlich ist auch die elektronische Erfassung am PC in einem Editor oder für den, der es grafisch mag, in einem Zeichenprogramm möglich; keinesfalls aber in der Datenbankanwendung direkt.

1.2.1 Datenmodellierung

Datenmodellierung ist ein Stichwort, das in diesem Zusammenhang immer wieder genannt wird. Gemeint sind damit Verfahren zur Abbildung der relevanten Objekte, ihrer Attribute und Beziehungen. Bis es zu einem brauchbaren Datenmodell kommt, werden in der Regel mehrere Modellierungsstufen durchlaufen.

Es gibt verschiedene Verfahren zur Datenmodellierung. Eines der bekanntesten ist die Entity-Relationship-Methode, die insbesondere bei relationalen Datenbanken häufig angewendet wird. Wichtige Begriffe sind:

- Entitätstyp (für die Kopfzeile in einer relationalen Datenbank)
- Attribut (für die Spaltenüberschrift)
- Entitätsmenge oder –set (die gesamte Tabelle)
- funktionale Beziehungen (Relationship, auch Fremdschlüssel)
- Entität (Datensatz, Zeile der Tabelle)
- Attributwert (Inhalt einer Datenzeile)

Für dieses Buch reicht ein verkürztes Modell, mit dem sich funktionierende Datenbanken selbst mit komplexen Strukturen erstellen lassen, um eine Einführung zu geben.

1.2.2 Eine Datenbank in fünf Schritten

Der Weg zur fertigen Datenbank kann als ein Weg in fünf Schritten beschrieben werden. Um dies besser nachvollziehen zu können, wird dieser Weg exemplarisch an einem Beispiel gezeigt. Sie können diese Schritte allerdings auf jedes Problem, für das Sie eine Datenbank entwickeln wollen, übertragen.

Als Beispiel soll eine Datenbank zur Speicherung von Musik-CDs erstellt werden. Dabei soll es möglich sein, nicht nur nach kompletten CDs zu suchen, sondern auch nach Interpreten und nach Titeln zu strukturieren und zu suchen.

1.2.3 Projektbeschreibung

1. **Informationen sortieren:** Im ersten Schritt werden nur die Objekte betrachtet, die für die Datenbank in Frage kommen. In diesem Beispiel sind das die CDs.
2. **Daten klassifizieren:** Im zweiten Schritt werden die Objekte näher untersucht. Dabei ist schnell festgestellt, dass die Titel auf der CD fehlen. Sie kommen als Objekt hinzu.
3. **Objekteigenschaften erfassen:** Im dritten Schritt werden die Objekteigenschaften beschrieben. Diese Eigenschaften dienen der näheren Beschreibung der Objekte. Den CDs kann ein Veröffentlichungsjahr zugeordnet werden, den Titeln Künstler, vielleicht

Komponisten usw. Aus der Vielzahl von Eigenschaften werden diejenigen herausgesucht, die für den speziellen Verwendungszweck der Datenbank benötigt werden.

4. **Datenbank einrichten:** In der Implementierungsphase wird die Datenbank eingerichtet. Tabellen, Layouts (Formulare, Berichte) und Programme (Skripte) werden erstellt und Testdaten eingegeben.
5. **Daten erfassen:** In der Nutzungsphase werden die echten Informationen (Daten) erfasst und zur Anwendung freigegeben.

Schritt 1 und 2 stellen die **Projektdefinition** dar, Schritt 3 die **Entwurfsphase**. Es folgt dann mit Schritt 4 die **Implementierungsphase** und mit Schritt 5 die anschließende **Nutzung**. Mit den ersten drei Schritten beschäftigt sich dieses Kapitel, mit den Schritten 4 und 5 die folgenden.

1.2.4 Das Vermeiden von Mehrfachspeicherungen

Auf das Problem der Mehrfachspeicherung (Redundanz) von Daten, die durch das Sammeln in einer einzigen Tabelle oder durch das Speichern in mehrere, voneinander unabhängige Tabellen entstehen können, habe ich bereits hingewiesen. Es wird dadurch nicht nur mehr Festplattenspeicher benötigt, sondern auch die Performance beeinträchtigt. Hinzu kommt der höhere Verwaltungsaufwand, insbesondere bei der Aktualisierung der Daten. Außerdem ist die Gefahr des Auftretens von Fehlern höher.

Der Prozess der Normalisierung wird an diesem Beispiel in einfacher Form und ohne theoretischen Ballast dargestellt.

Ausgangssituation festlegen

Zunächst muss festgelegt werden, welche Informationen benötigt werden. Die erste Konzeption der Tabellen kann dabei frei und ohne Einschränkung erfolgen. Bezogen auf unser Beispiel könnte die Tabelle *CD* folgendermaßen aussehen:

CD_Nr.	Album	Jahr	Interpret	Label	Tracks
1	Who's next	1971	The Who	Polydor	1. Baba O'Riley 2. Bargain 3. Love ain't for keeping 4. My Wife
2	Brainwashed	2002	George Harrison	Parlophone	1. Any road 2. P2 Vatican Blues 3. Pisces Fish

Diese Tabelle zeigt deutlich einige Probleme:

- Die Informationen im Feld *Tracks* sind unübersichtlich. Ein gezielter Zugriff auf einen bestimmten Track ist schwierig zu realisieren.
- Das Eintragen jeweils nur eines Tracks in einer Zeile führt zu erheblichen Redundanzen in der Tabelle.
- Informationen zum Feld *Interpret* sind dürftig. Das Ergänzen um weitere Spalten führt ebenfalls zu Redundanzen in dieser Tabelle.

Eine brauchbare Datenbank enthält keinerlei redundante Informationen. Das erreicht man durch den Prozess der Normalisierung. Und diese führt man durch, indem man die vorhandenen Tabellen zerlegt, und zwar so lange, bis alle Redundanzen verschwunden sind.

Die erste Normalform – Tabellen werden zerlegt

In der ersten Stufe werden in der Tabelle alle Felder so weit zerlegt, bis jedes Feld nur noch eine Information enthält. Die Tabelle CD sieht in der ersten Normalform folgendermaßen aus:

CD_Nr.	Album	Jahr	Interpret	Label	Tracknr.	Track
1	Who's next	1971	The Who	Polydor	1.	Baba O'Riley
2	Who's next	1971	The Who	Polydor	2.	Bargain
3.	Who's next	1971	The Who	Polydor	3.	Love Ain't For Keeping
4.	Who's next	1971	The Who	Polydor	4.	My Wife
5.	Brainwashed	2002	George Harrison	Parlophone	1.	Any Road
6.	Brainwashed	2002	George Harrison	Parlophone	2.	P2 Vatican Blues

Das Feld *Tracks* wurde in zwei Felder (*Tracknr*, *Track*) aufgeteilt. Nun ist der Zugriff auf jeden einzelnen Song möglich. Das Problem der Redundanz – wiederholtes Speichern der Informationen – tritt aber überdeutlich zu Tage.

Die zweite Normalform – doppelte Informationen entfernen

Da es das Ziel der Normalisierung ist, redundante Datenspeicherung zu vermeiden, müssen diese doppelten Einträge entfernt werden. Trotzdem sollen die Informationen mit den anderen Objektdaten in einen Zusammenhang gebracht werden können. Erreicht wird dies durch Aufteilung der Tabelle und Festlegung von Primärschlüsseln.

GRUNDLAGEN

Ein Primärschlüssel ist ein bestimmtes Feld oder eine Kombination von Feldern, die einen Datensatz eindeutig identifizieren. Verbindet man Tabellen über solche Schlüssel, entstehen Beziehungen zwischen diesen Tabellen. Nach diesem Schritt spricht man davon, dass die Tabellen in der »zweiten Normalform« vorliegen.

Die Tabellen CD und Tracks sehen in der zweiten Normalform folgendermaßen aus:

CD_Nr.	Album	Jahr	Interpret
1	Who's next	1971	The Who
2	Brainwashed	2002	George Harrison

CD_Nr.	Tracknr	Track
1	1	Baba O'Riley
1	2	Bargain
1	3	Love Ain't For Keeping
1	9	Won't Get Fooled Again
2	1	Any Road
2	2	P2 Vatican Blues
2	3	Pisces Fish
2	10	Between the Devil and the Deep Blue Sea

Die komplette Trackliste ist nun in eine eigene Tabelle ausgelagert worden. Über das Feld *CD-Nr.* kann aber auf die Daten der Tabelle CD zugegriffen werden.

GRUNDLAGEN

In der Tabelle CD ist das Feld *CD_Nr.* ein Primärschlüssel. In der Tabelle Tracks spricht man beim Feld *CD_Nr.* von einem Fremdschlüssel, weil damit auf einen Datensatz in einer fremden Tabelle verwiesen wird.

Bei umfangreicheren Projekten muss man den Prozess der Normalisierung noch weiter treiben. Manchmal geht es bis zur 5. Normalform. In diesem Fall ist die zweite Normalform ausreichend. Auch das Erweitern der Informationen (Interpret) schenken wir uns an dieser Stelle, da das Ziel, die Normalisierung im Ansatz zu erklären, hier schon erreicht ist.

1.2.5 Schlüssel und Index

Zwei Begriffe schauen wir uns noch an, um das Thema Datenbankgrundlagen abzurunden: Primärschlüssel und Index.

Datensätze eindeutig identifizieren

Der Primärschlüssel wurde in den vorangegangenen Beschreibungen bereits erwähnt. Er ist in relationalen Datenbanktabellen eine wichtige Information, denn er dient zur eindeutigen Identifikation eines Datensatzes in der Tabelle. Kein anderer Datensatz der Tabelle darf den gleichen Primärschlüssel besitzen. Über diesen Schlüssel ist also ein Datensatz eindeutig identifizierbar.

Tatsächlich kann eine Tabelle auch ohne Primärschlüsselfeld erstellt werden. Allerdings kann diese dann niemals als Primärtabelle eingesetzt werden und damit an einer gültigen Beziehung teilhaben. Das bedeutet, dass andere Tabellen nicht auf Datensätze dieser Tabelle verweisen können.

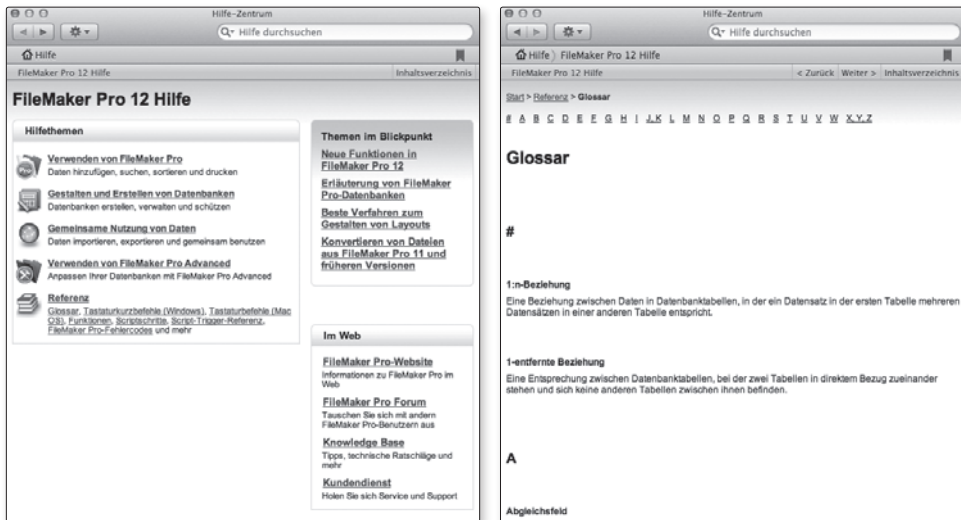
Ein Primärschlüssel in einer Datenbanktabelle hat vielfältige Aufgaben. Dazu gehören:

- eindeutige Identifizierung eines jeden Datensatzes
- Indexerstellung für die schnelle Suche nach Datensätzen
- Anzeige der Daten in Primärschlüssel-Reihenfolge
- Verhindern eines weiteren Datensatzes mit identischem Schlüssel.

TIPP

Bauen Sie immer zunächst die Tabellen ohne Primärschlüssel auf, bis Sie die Tabellen durch den Prozess der Normalisierung endgültig festgelegt haben. Prüfen Sie vor der Vergabe von Schlüsseln alle Felder, die als Schlüsselfelder definiert werden sollen und benennen Sie diese anschließend um. Entscheiden Sie sich dann, welches Feld ein Primärschlüsselfeld und welche Felder Fremdschlüsselfelder werden.

FileMaker war schon immer die »etwas andere Datenbank«. Bei FileMaker Pro sind es die Abgleichfelder, die eine entsprechende Schlüsselfunktion haben. Ein Abgleichfeld ist ein Feld in der aktuellen Tabelle und ein Feld in einer Bezugstabelle, deren Werte für den Zugriff auf übereinstimmende Datensätze verwendet werden. Im Glossar von FileMaker Pro (finden Sie über *Hilfe | FileMaker Pro Hilfe*; das Glossar steht als letzter Eintrag unter *Referenz*) findet man dann unter *Schlüssel* noch die Definition: *eine oder mehrere Spalten, durch die eine bestimmte Zeile eindeutig wird (entspricht einem Abgleichfeld)*.



Das FileMaker-Glossar wird über die Hilfe aufgerufen.

Auf den Index gesetzt

In Datenbanken werden Indizes verwendet, um Datenbestände zu sortieren und Datenbankoperationen zu beschleunigen. Wird eine Tabelle ohne Index angelegt, werden die Daten in den Tabellen durch das Datenbankmanagementsystem in der Reihenfolge der Eingabe abgespeichert und auch in dieser Reihenfolge angezeigt. Ist ein Primärschlüssel festgelegt, wird zwar immer noch in der Reihenfolge der Eingabe gespeichert, die Anzeige der Daten erfolgt jetzt aber in der Sortierung nach dem Primärschlüssel.

Das funktioniert, weil das Datenbankmanagementsystem automatisch eine weitere Tabelle anlegt, in der die sortierten Primärschlüssel mit einem Hinweis auf den zugehörigen Datensatz hinterlegt werden. Auf diese Indextabelle wird zuerst zugegriffen, sodass die eigentliche Tabelle immer in der vorgegebenen Sortierung erscheint.

Haben Sie zum Beispiel eine umfangreiche Adressdatenbank, in der Sie oft auch Adressen anhand des Namens suchen, erhöht die Indizierung dieses Feldes die Suchgeschwindigkeit. Nicht die gesamte Tabelle wird dadurch in den Speicher geladen und durchsucht, sondern nur der Index. Wird ein passender Name gefunden, wird die zugehörige Adresse geladen. Dies geschieht vollautomatisch im Hintergrund. Sie müssen in der Tabelle nur die entsprechenden Felder als Index auswählen.

Die Nachteile einer Indizierung sind die, dass zusätzlicher Platz auf dem Datenträger benötigt wird und dass Aktualisierungsvorgänge sich verlangsamen. Schließlich muss jede Änderung auch in den Indextabellen berücksichtigt werden. Es ist also immer abzuwägen, inwieweit eine Tabelle beziehungsweise Datenbank durch zusätzliche Indizierung noch

vergrößert werden soll. Eine richtig eingesetzte Indizierung sorgt allerdings für eine funktionale und gut zu handhabende Datenbank.

FileMaker Pro 12 stellt Optionen zur Indizierung von Feldern bereit und erlaubt auch das Speichern von berechneten Ergebnissen. Dazu mehr in den folgenden Kapiteln.

Beziehungen pflegen

Eine Tabelle macht noch keine Datenbank; zumindest keine relationale Datenbank. Erst wenn mehrere Tabellen miteinander verknüpft werden, kann man von einer echten, relationalen Datenbank sprechen. Diese Verknüpfungen nennt man auch »Beziehungen«.

Im Beispiel der CD-Datenbank wurde der Prozess der Normalisierung durchgeführt, damit redundante Informationen in der Datenbank nicht mehr vorkommen. Erreicht wurde das durch Aufteilung des Datenbestands in zwei Tabellen. Da die Informationen in diesen Tabellen aber irgendwie doch zusammengehören, muss eine Möglichkeit gefunden werden, Verbindungen zwischen diesen Tabellen herzustellen. Diese Verbindungen sind die Beziehungen.

Es gibt vier verschiedene Arten von Beziehungen zwischen Tabellen:

- 1:1-Beziehungen
- 1:n-Beziehungen
- n:1-Beziehungen
- n:m-Beziehungen

Die monogame Beziehung

Wird ein Datensatz einer Tabelle mit nur genau einem Datensatz einer anderen Tabelle in Beziehung gesetzt, spricht man von einer 1:1-Beziehung (Eins-zu-Eins-Beziehung). Das kommt zum Beispiel dann vor, wenn bestimmte Datensätze aufgespalten werden, etwa um differenzierte Zugriffsrechte zu gewährleisten.

Ein Beispiel: Eine umfangreiche Kundentabelle wird aufgeteilt in eine Tabelle, die Informationen allgemeiner Art (etwa die Anschrift) und eine weitere Tabelle, die Informationen spezieller Art (zum Beispiel Umsatz, Bestellhäufigkeit, Ansprechpartner) enthält. Hier kann nur ein Datensatz eines Kunden einem Datensatz in der anderen Tabelle gegenüberstehen.

In unserem CD-Beispiel kommt diese Beziehung nicht vor. Würde aber je CD ein Platz im Regal eingetragen und dieser Platz in einer eigenen Tabelle verwaltet, dann läge solch eine Beziehung vor. Es kann ja immer nur einen Platz für eine CD im Regal geben.

Die offene Beziehung

Am häufigsten kommen die 1:n-Beziehung (Eins-zu-Viele-Beziehung) und die n:1-Beziehung (Viele-zu-Eins-Beziehung) in relationalen Datenbanken vor. Ein Datensatz der einen Tabelle ist mit vielen Datensätzen der anderen Tabelle verbunden (1:n). In der zweiten

Tabelle gibt es aber viele Datensätze, die mit einem Datensatz der ersten Tabelle in Beziehung stehen (n:1).

Ein Artikeldatensatz kann zum Beispiel in mehreren Datensätzen der Auftragsdatentabelle vorkommen (1:n). Mehrere Datensätze der Auftragsdatentabelle können aber einem Datensatz der Artikeltabelle gegenüberstehen (n:1). Das Schlüsselfeld in der Tabelle mit der 1:n-Beziehung muss eindeutig sein und kann keine Duplikate zulassen.

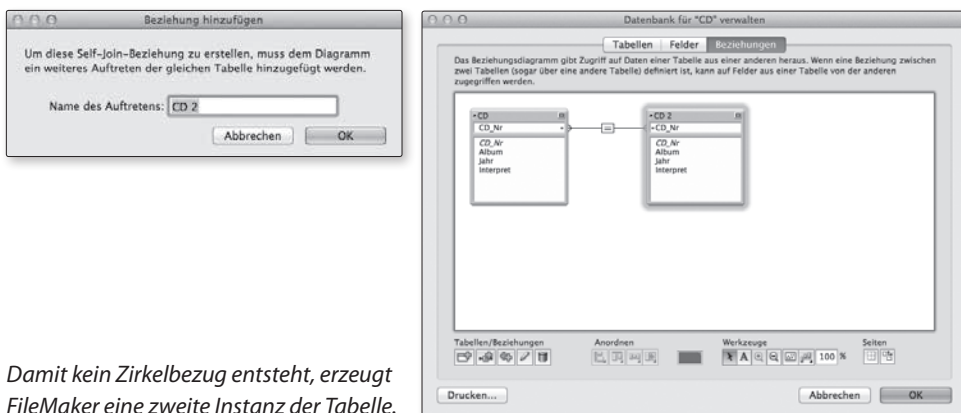
In unserem Beispiel besteht zwischen der CD-Tabelle und der Track-Tabelle eine n:1-Beziehung. Ein Track kann immer nur auf eine CD verweisen (n:1). Eine CD kann aber mehrere Tracks in der Tabelle haben (1:n). Das Schlüsselfeld in der Tabelle mit der n:1-Beziehung darf mehrdeutig sein und kann Duplikate zulassen.

Jeder mit Jedem

Die n:m- oder Viele-zu-Viele-Beziehungen sind nicht so unwahrscheinlich, wie man zunächst denken mag. Werden Schüler und Unterrichtsfächer in Verbindung gebracht, so wird dies sofort deutlich. Ein Schüler hat verschiedene Unterrichtsfächer. In einem Unterrichtsfach ist aber nicht nur ein Schüler, sondern es sind viele Schüler vertreten. Eine Datenbank, die Schüler und Unterrichtsfächer verwaltet, zum Beispiel um Stundenpläne zu erstellen, muss dies berücksichtigen.

Beziehung zu sich selbst

Es klingt zunächst paradox: eine Beziehung zu sich selbst – wie soll das möglich sein? Und welchen Nutzen hat das? Diese so genannte Self-Join-Beziehung ist mit FileMaker Pro durchaus zu realisieren. Dabei wird eine Information in einem Datensatz mit allen anderen Datensätzen verglichen. Sie können über solch eine Beziehung das Anlegen von Duplikaten bereits bei der Datenerfassung verhindern. Oder es können über bestimmte Informationen vorhandene weitere Informationen übernommen werden (zum Beispiel über die Bankleitzahl das Bankinstitut). Damit solch eine Beziehung realisiert werden kann, erzeugt FileMaker Pro eine zweite Instanz der Tabelle.

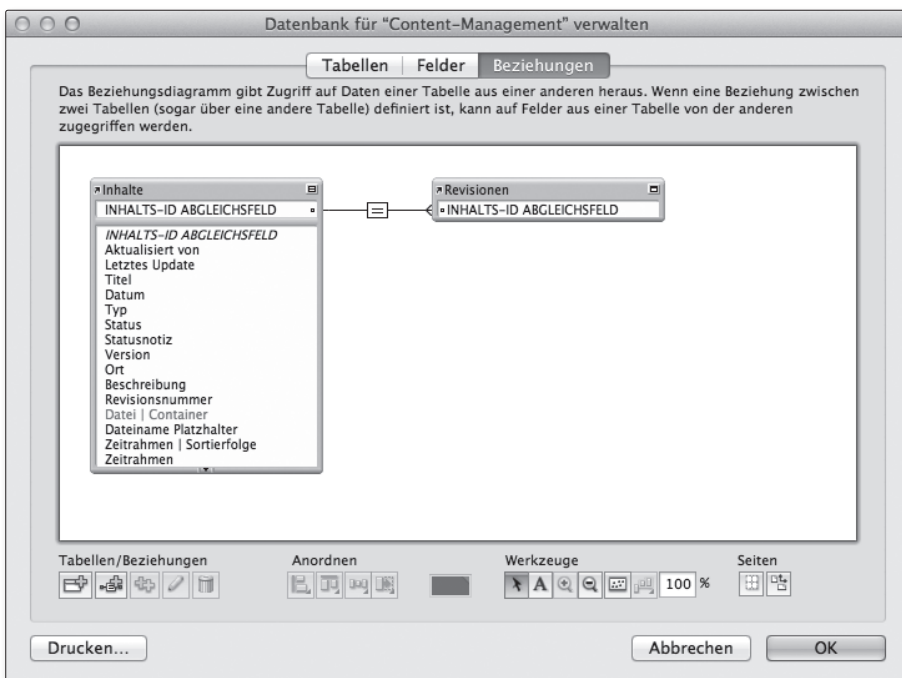


Damit kein Zirkelbezug entsteht, erzeugt FileMaker eine zweite Instanz der Tabelle.

Datenbank-Entwickler haben sich also intensiv mit den Beziehungen in der Datenbank auseinanderzusetzen. Gut organisierte Beziehungen zwischen den Tabellen sorgen für ein effektives und weitgehend problemfreies Arbeiten mit den Datenbanktabellen.

1.3 FileMaker Pro als relationales Datenbank-Managementsystem

FileMaker Pro ist seit der Version 3 ein relationales Datenbanksystem. Daten wurden in Tabellen verwaltet, und zwischen den Tabellen können über Abgleichfelder Beziehungen hergestellt werden. Die Kritik, dass dieses relationale System nur angeflickt sei, ist nur teilweise berechtigt und seit der Version FileMaker Pro 7 gar nicht mehr. Die komplette Datenbank-Engine – also der Teil des Programms, der sich mit der Datenbankverwaltung beschäftigt – wurde neu entwickelt. Sämtliche Tabellen können nun in einer Datei verwaltet werden, und Beziehungen lassen sich grafisch erzeugen. Was FileMaker Pro nach wie vor von anderen Datenbank-Managementsystemen unterscheidet, ist die einfache Handhabung, die schnelle Entwicklung selbst komplexer Datenbanken ermöglicht und leider auch das Fehlen einer objektorientierten Programmiersprache.



Beziehungen erzeugen Sie per Mausklick.