



Gunter
Saake

Kai-Uwe
Sattler

Andreas
Heuer

4. Auflage

A close-up photograph of a beaver's head, wet with water droplets, looking directly at the camera. The beaver's fur is brown and matted with water. Its mouth is slightly open, showing its yellow teeth. The background is a blurred, greyish water surface.

Datenbanken

Konzepte und Sprachen

*Gunter Saake
Kai-Uwe Sattler
Andreas Heuer*

Datenbanken

Konzepte und Sprachen

Vierte Auflage



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.

Bei der Herstellung des Werkes haben wir uns zukunftsbewusst für umweltverträgliche und wiederverwertbare Materialien entschieden. Der Inhalt ist auf elementar chlorfreiem Papier gedruckt.

ISBN 978-3-8266-9057-0

4. Auflage 2010

E-Mail: kundenbetreuung@hjr-verlag.de

Telefon: +49 89/2183-7928

Telefax: +49 89/2183-7620

© 2010 mitp, eine Marke der Verlagsgruppe Hüthig Jehle Rehm GmbH
Heidelberg, München, Landsberg, Frechen, Hamburg

www.mitp.de

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Lektorat: Ernst-Heinrich Profener

Korrektorat: Astrid Sander

Satz: Gunter Saake, Magdeburg; Kai-Uwe Sattler, Ilmenau; Andreas Heuer, Rostock

Druck: Beltz Druckpartner, Hemsbach

Vorwort zur vierten Auflage

Die erste Auflage des Biber-Buches erschien im September 1995, die zweite im Januar 2000. In der seitdem verflossenen Zeit, immerhin zehn Jahre, ist die Entwicklung im Datenbankbereich nicht stehengeblieben – wichtige Themen wie SQL:2003, XML, objektrelationale Datenbanken, Data Warehouses und die Verwaltung multimedialer Daten spielten damals noch keine wichtige Rolle oder waren nicht in präsentierbarer Form. Weiterhin haben sich auch im Aufbau, in laufenden Beispielen und in der Didaktik in den letzten Jahren in den Vorlesungen zum Buch diverse Änderungen ergeben, die nun endlich auch zur völligen Neuentwicklung schon bestehender Buchkapitel führen musste. Daher war es an der Zeit, mit der dritten Auflage eine umfassende Neubearbeitung vorzunehmen. Einige kleinere Fehler, die in der dritten Auflage übersehen oder unglücklicherweise neu eingebaut wurden, haben wir in der hiermit vorliegenden vierten Auflage korrigiert.

Die Überarbeitung in der dritten Auflage wurde genutzt, um den Stoff des Buches neu zu organisieren. Dieses Umstrukturierung wurde auch dadurch motiviert, dass in einer nur zweistündigen Vorlesung die ersten Kapitel in den vorherigen Auflagen zu gehaltvoll waren, um direkt chronologisch als Stoff für eine Vorlesung herzuhalten. Nun ist das Buch in drei Teile aufgeteilt: Der erste Teil bis zum Kapitel 7 bearbeitet umfassend die Kernkonzepte der relationalen Datenbanken, ohne auf Spezialitäten und andere Datenmodelle einzugehen. Er bildet den Kern einer Vorlesung „Grundlagen von Datenbanken“ auch mit geringerem Stundenumfang.

Der zweite Teil vertieft nun die Themen des erste Teils, und kann insbesondere für eine 3- oder 4-stündige Vorlesung herangezogen werden. Zusammen geben die ersten beiden Teile eine umfassende Behandlung von Theorie, Entwurfsmethoden, und Sprachkonzepten für relationale Datenbanken inklusive der ausführlichen Behandlung von SQL.

Der dritte Teil behandelt Alternativen und Erweiterungen bei Datenmodellen. Hiervon können ausgewählte Teile in eine Grundvorlesung übernom-

men werden, oder die Basis für Spezialvorlesungen bilden. In diesen Kapiteln wird die Sprach- bzw. Anwendungssicht in den Vordergrund gestellt; Implementierungsaspekte werden im Buch „Datenbanken: Implementierungstechniken“ der Autoren behandelt und sind deshalb hier ausgeblendet.

Danksagungen

Zu danken haben wir bei dieser Neuauflage insbesondere für Korrekturen und Hinweise für Aktualisierungen und Erweiterungen unseren (jetzigen und ehemaligen) Mitarbeitern und Studenten Ingolf Geist, Eike Schallehn, Andreas Lübcke, Stephan Vornholt, Christine Krause, Rita Schindler, Meike Klettke, Holger Meyer, Temenushka Ignatova, Andre Peters, Martin Garbe, Alf-Christian Schering, Dagmar Köhn und Nils Weber sowie allen Lesern, die uns Korrekturhinweise geschickt haben. Weiterhin danken wir Knut Stolze für die Unterstützung bei praktischen Tests.

Ein Dankeschön geht auch an den zuständigen Lektor des MITP-Verlages Ernst-Heinrich Profener, der sehr viel Geduld aufgebracht hat, sowie an Astrid Sander für das sorgfältige Korrekturlesen.

Gunter Saake bedankt sich bei Birgit, Torben und Annkristin für den liebevollen und familiären Rückhalt, der ein sich lange hinziehendes Buchprojekt erst erträglich machen kann. Bedanken möchte er sich ebenfalls für die (zum Teil neuen) Erfahrungen in verteiltem kooperativen Arbeiten, die er mit seinen Mitautoren in den letzten Jahren sammeln konnte.

Kai-Uwe Sattler bedankt sich bei Britta, Arved und Bennett, ohne deren Liebe und Rückhalt sowie das Verständnis für die langen „durchgeschriebenen“ Nächte ein solches Buchprojekt wohl nicht möglich wäre. Dank gilt auch seinen Mitautoren dafür, dass das Buch letztendlich doch noch fertig wurde.

Andreas Heuer möchte sich schließlich bei Renate und insbesondere bei seinen beiden Mitautoren bedanken, die die drastischen Aufwandsunterschätzungen und das absolut perfekte Zeitmanagement des Drittautors immer wieder ausgleichen mussten („Sagte ich drei Tage für die Aktualisierung des Kapitels? Ich hatte vor drei Monaten wohl mindestens drei Wochen gemeint.“).

Ergänzende Informationen zu Buch, wie Verweise zu begleitenden Vorlesungsmaterialien und gegebenenfalls erforderliche Fehlerkorrekturen sind im Web unter folgender Adresse zu finden:

<http://www.biberbuch.de>

Magdeburg, Ilmenau und Rostock, im November 2007 und August 2010

Gunter Saake, Kai-Uwe Sattler und Andreas Heuer

Inhaltsverzeichnis

Vorwort zur vierten Auflage	v
Inhaltsverzeichnis	vii
1 Grundlegende Konzepte	1
1.1 Motivation und Historie	1
1.2 Komponenten und Funktionen	7
1.2.1 Prinzipien und Aufgaben	7
1.2.2 Aufbau und Funktionen eines Datenbanksystems	9
1.2.3 Einsatzgebiete, Grenzen und Entwicklungstendenzen	19
1.2.4 Wann kommt was?	22
1.3 Beispielanwendung	23
1.4 Vertiefende Literatur	25
1.5 Übungsaufgaben	25
I Kernkonzepte relationaler Datenbanken	27
2 Architekturen von Datenbanksystemen	29
2.1 Schemaarchitektur und Datenunabhängigkeit	30
2.2 Systemarchitekturen	35
2.2.1 ANSI-SPARC-Architektur	35
2.2.2 Fünf-Schichten-Architektur	37
2.2.3 Konkrete Systemarchitekturen	40
2.3 Anwendungsarchitekturen	43
2.4 Zusammenfassung	47
2.5 Vertiefende Literatur	47
2.6 Übungsaufgaben	49

3	Das Entity-Relationship-Modell	51
3.1	Datenbankmodelle	51
3.2	Semantikfestlegung für Datenbankmodelle	56
3.3	Grundlagen des Entity-Relationship-Modells	59
3.3.1	Grundkonzepte des klassischen ER-Modells	59
3.3.2	Ein einfaches Beispiel für ein ER-Schema	66
3.3.3	Semantik eines ER-Schemas	67
3.4	Eigenschaften von Beziehungen	68
3.4.1	Stelligkeit	68
3.4.2	Kardinalitäten und funktionale Beziehungen	71
3.5	Weitere Konzepte im Entity-Relationship-Modell	78
3.5.1	Abhängige Entity-Typen	78
3.5.2	Die IST-Beziehung	79
3.5.3	Optionalität von Attributen	81
3.6	Zusammenfassung	82
3.7	Vertiefende Literatur	82
3.8	Übungsaufgaben	83
4	Relationenmodell und Anfragemodelle	85
4.1	Relationenmodell: Strukturteil	85
4.1.1	Schemata und Instanzen	85
4.1.2	Integritätsbedingungen	90
4.2	Relationale Anfragemodelle: Operationenteil	94
4.2.1	Kriterien für Anfragesprachen	94
4.2.2	Relationenalgebra	95
4.2.3	Anfragekalkül	106
4.2.4	Tupelkalkül	110
4.3	Änderungsoperationen	112
4.3.1	Allgemeine Grundprinzipien	112
4.3.2	Relationale Änderungsoperationen	113
4.4	Zusammenfassung	114
4.5	Vertiefende Literatur	114
4.6	Übungsaufgaben	114
5	Phasen des Datenbankentwurfs	119
5.1	Entwurfsaufgabe	119
5.2	Phasenmodell	121
5.2.1	Anforderungsanalyse	123
5.2.2	Konzeptioneller Entwurf	125
5.2.3	Verteilungsentwurf	127
5.2.4	Logischer Entwurf	128
5.2.5	Datendefinition	129
5.2.6	Physischer Entwurf	130

5.2.7	Implementierung und Wartung	130
5.2.8	Objektorientierte Entwurfsmethoden	130
5.2.9	Phasenbegleitende Methoden	131
5.3	Konzeptioneller Entwurf	132
5.3.1	Konzeptionelles Schema	132
5.3.2	Objektschicht	133
5.3.3	Datenschicht	134
5.3.4	Entwicklungsschicht	137
5.3.5	Aktionsschicht	138
5.3.6	Modellierung von Anwendungsprozessen	139
5.4	Aspekte der Datenintegration	140
5.4.1	Heterogenität der Datenmodelle	141
5.4.2	Heterogene Datenbankschemata	142
5.4.3	Heterogenität auf der Datenebene	142
5.4.4	Schemakonflikte bei der Integration	143
5.5	Entity-Relationship-Abbildung auf das Relationenmodell	144
5.5.1	Informationskapazität	145
5.5.2	Beispiel für eine Abbildung auf das Relationenmodell	147
5.5.3	Abbildungsregeln für das relationale Modell	148
5.6	Zusammenfassung	156
5.7	Vertiefende Literatur	156
5.8	Übungsaufgaben	158
6	Relationaler Datenbankentwurf	159
6.1	Funktionale Abhängigkeiten	161
6.2	Schemaeigenschaften	173
6.2.1	Änderungsanomalien	174
6.2.2	Normalformen	175
6.2.3	Minimalität	182
6.3	Transformationseigenschaften	183
6.3.1	Abhängigkeitstreue	183
6.3.2	Verbundtreue	185
6.4	Entwurfsverfahren	188
6.4.1	Ziele	188
6.4.2	Dekompositionsverfahren	189
6.4.3	Syntheseverfahren	192
6.4.4	Verfeinerung des Entity-Relationship-Datenbankentwurfs	196
6.5	Mehrwertige Abhängigkeiten	197
6.5.1	Grundlagen	197
6.5.2	Schemaeigenschaften	200
6.5.3	Transformationseigenschaften	202
6.6	Weitere Abhängigkeiten und Verfahren	202
6.7	Weitere relationale Entwurfsverfahren	204

6.8	Zusammenfassung	205
6.9	Vertiefende Literatur	205
6.10	Übungsaufgaben	207
7	Die relationale Datenbanksprache SQL	209
7.1	SQL als Datendefinitionssprache	209
7.1.1	Erzeugen von Tabellen	210
7.1.2	Tabellen mit Integritätsbedingungen	214
7.1.3	Löschen und Ändern von Tabellendefinitionen	215
7.1.4	Erzeugen und Löschen von Indexen	217
7.2	SQL als relationale Anfragesprache	218
7.2.1	Überblick	219
7.2.2	Die from -Klausel	220
7.2.3	Die select -Klausel	223
7.2.4	Die where -Klausel	226
7.2.5	Mengenoperationen	230
7.2.6	Schachtelung von Anfragen	232
7.2.7	Mächtigkeit des SQL-Kerns	238
7.3	Änderungsoperationen in SQL	238
7.3.1	Übersicht über Änderungen in SQL	239
7.3.2	Die update -Anweisung	239
7.3.3	Die delete -Anweisung	241
7.3.4	Die insert -Anweisung	242
7.3.5	Probleme bei SQL-Änderungen	243
7.4	Zusammenfassung	244
7.5	Vertiefende Literatur	245
7.6	Übungsaufgaben	245
II	Erweiterte Konzepte für relationale Datenbanken	247
8	Erweiterte Entwurfsmodelle	249
8.1	Erweiterungen des ER-Modells	249
8.1.1	Erweiterungen bei Attributen	250
8.1.2	Spezialisierung und Generalisierung	251
8.1.3	Komplexe Objekte	252
8.1.4	Beziehungen höheren Typs	253
8.2	Das EER-Modell – ein erweitertes ER-Modell	254
8.2.1	Übernommene Grundkonzepte aus dem klassischen ER-Modell	254
8.2.2	Erweiterung bei Wertebereichen	255
8.2.3	Mengenwertige und strukturierte Attribute	256
8.2.4	Der Typkonstruktor: Spezialisierung, Generalisierung, Partitionierung	256

8.2.5	Aggregation und Sammlung mittels objektwertiger Attribute	266
8.2.6	Erweitertes Schlüsselkonzept	268
8.2.7	Abgeleitete Konzepte	269
8.2.8	Vergleich zu anderen erweiterten ER-Modellen	270
8.3	UML für den Datenbankentwurf	271
8.3.1	Das Objektmodell von UML	272
8.3.2	Darstellung von Klassen in UML	273
8.3.3	Beziehungen in UML	274
8.3.4	Aggregation in UML	276
8.3.5	Spezialisierung in UML	277
8.4	Zusammenfassung	278
8.5	Vertiefende Literatur	278
8.6	Übungsaufgaben	280
9	Grundlagen von Modellen und Anfragen	281
9.1	Erweiterungen der Relationenalgebra	281
9.2	Erweiterte Modelle und Anfragealgebren	285
9.2.1	Geschachtelte Relationen: Das NF ² -Modell	285
9.2.2	PNF-Relationen	286
9.2.3	Verallgemeinerte geschachtelte Relationen	286
9.2.4	Erweiterte Anfragealgebren	288
9.3	Anfragekalküle	292
9.3.1	Bereichskalkül	292
9.3.2	Sichere Anfragen	294
9.3.3	Beispiele für Anfragen im Bereichskalkül	296
9.3.4	Eigenschaften des Bereichskalküls	297
9.3.5	Kalküle für andere Datenmodelle	298
9.4	Zusammenfassung	302
9.5	Vertiefende Literatur	303
9.6	Übungsaufgaben	303
10	Erweiterte Konzepte von SQL	305
10.1	Weitere Operationen und Prädikate	305
10.1.1	Skalare Ausdrücke	305
10.1.2	Prädikate	311
10.1.3	Quantoren und Mengenvergleiche	311
10.1.4	Behandlung von Nullwerten	314
10.2	Aggregation und Gruppierung	316
10.2.1	Aggregatfunktionen	316
10.2.2	Gruppierung	319
10.3	Sortierung	322
10.4	Äußere Verbunde	323

10.5	Künstliche Schlüssel und Sequenzgeneratoren	325
10.6	Benannte Anfragen und Rekursion	327
10.6.1	Benannte Anfragen	327
10.6.2	Rekursive Anfragen	328
10.7	SQL-Versionen	336
10.7.1	SEQUEL2	336
10.7.2	SQL-89	338
10.7.3	SQL-92	339
10.7.4	SQL:1999 und SQL:2003	340
10.8	Zusammenfassung	342
10.9	Vertiefende Literatur	342
10.10	Übungsaufgaben	343
11	Weitere relationale Datenbanksprachen	345
11.1	QUEL	346
11.1.1	Anfragen in QUEL	346
11.1.2	Änderungsoperationen in QUEL	349
11.2	Query by Example	349
11.2.1	Anfragen in QBE	350
11.2.2	Funktionen, Sortierung und Aggregation in QBE	354
11.2.3	Formale Semantik von QBE	355
11.2.4	Ausdrucksfähigkeit von QBE	356
11.2.5	Änderungen in QBE	357
11.2.6	Anfragen in MS Access	359
11.2.7	Andere graphische Anfragesprachen	362
11.3	Datalog	363
11.3.1	Grundbegriffe	364
11.3.2	Semantik rekursiver Regeln	366
11.3.3	Semantik und Auswertung von Datalog	367
11.4	Tutorial D	368
11.4.1	Datentypen	368
11.4.2	Anfrageoperatoren	370
11.4.3	Änderungsoperationen	373
11.4.4	Constraints	374
11.5	Zusammenfassung	375
11.6	Vertiefende Literatur	375
11.7	Übungsaufgaben	376
12	Transaktionen, Integrität & Trigger	377
12.1	Grundlagen von Transaktionen	378
12.1.1	ACID-Prinzip	378
12.1.2	Probleme im Mehrbenutzerbetrieb	380
12.1.3	Transaktionssteuerung in SQL	385

12.1.4	Transaktionen und Integritätssicherung	388
12.2	Architekturen zur Integritätssicherung	389
12.2.1	Integritätssicherung durch Anwendung	390
12.2.2	Integritätsmonitor als Komponente des DBMS	390
12.2.3	Integritätssicherung durch Einkapselung	391
12.3	Integritätsbedingungen in SQL	392
12.3.1	Inhärente Integritätsbedingungen im Relationenmodell	392
12.3.2	Weitere Bedingungen in der SQL-DDL	393
12.3.3	Die assertion -Klausel	394
12.3.4	Verwaltung und Überprüfung von Bedingungen	394
12.4	Klassifikation von Integritätsbedingungen	395
12.5	Trigger und aktive Datenbanken	398
12.5.1	Grundprinzipien von Triggern	399
12.5.2	Aktive Datenbanken und ECA-Regeln	401
12.6	Methoden der Integritätssicherung	405
12.6.1	Integritätssicherung durch Trigger	405
12.6.2	Integritätssicherung durch Anfragemodifikation	407
12.7	Zusammenfassung	409
12.8	Vertiefende Literatur	410
12.9	Übungsaufgaben	410
13	Datenbankanwendungsentwicklung	413
13.1	Grundprinzipien	414
13.2	Programmiersprachenanbindung: Call-Level-Schnittstellen	416
13.2.1	SQL/CLI: Der Standard	417
13.2.2	ODBC	420
13.2.3	JDBC	421
13.2.4	Weitere Call-Level-Schnittstellen	426
13.3	Eingebettetes SQL	427
13.3.1	Statische Einbettung: Embedded SQL	427
13.3.2	Dynamische Einbettung: Dynamic SQL	435
13.3.3	SQLJ: Embedded SQL für Java	436
13.4	High-Level-Schnittstellen	438
13.4.1	Grundlagen der Abbildung	439
13.4.2	Hibernate	442
13.4.3	Weitere Technologien	450
13.5	Prozedurale SQL-Erweiterungen und Datenbanksprachen	451
13.5.1	Vorteile von gespeicherten Prozeduren	452
13.5.2	SQL/PSM: Der Standard	453
13.5.3	PL/SQL von Oracle	460
13.5.4	Gespeicherte Prozeduren in Java	462
13.6	Zusammenfassung	465
13.7	Vertiefende Literatur	467

13.8	Übungsaufgaben	467
14	Sichten	469
14.1	Motivation und Begriffsbildung	470
14.1.1	Sichten und externe Schemata	471
14.1.2	Definition von Sichten	471
14.1.3	Definition von Sichten in SQL	472
14.1.4	Vorteile von Sichten	473
14.2	Probleme mit Sichten	474
14.2.1	Kriterien für Änderungen auf Sichten	475
14.2.2	Projektionssichten	476
14.2.3	Selektionssichten	478
14.2.4	Verbundsichten	479
14.2.5	Aggregationssichten	481
14.2.6	Klassifikation der Problembereiche	482
14.3	Behandlung von Sichten in SQL	483
14.3.1	Auswertung von Anfragen an Sichten in SQL	484
14.3.2	Sichtänderungen in SQL-92	486
14.3.3	Sichtänderungen in SQL:2003	486
14.4	Theorie änderbarer Sichten	487
14.5	Instead-of-Trigger für Sichtänderungen	489
14.6	Zusammenfassung	492
14.7	Vertiefende Literatur	494
14.8	Übungsaufgaben	494
15	Zugriffskontrolle & Privacy	495
15.1	Sicherheitsmodelle	497
15.1.1	Diskrete Sicherheitsmodelle	497
15.1.2	Verbindliche Sicherheitsmodelle	497
15.2	Rechtevergabe in SQL	498
15.2.1	Benutzer und Schemata	499
15.2.2	Rechtevergabe in SQL	499
15.2.3	Zurücknahme von Rechten	501
15.2.4	Rollenmodell in SQL:2003	502
15.2.5	Auditing	502
15.2.6	Authentifikation und Autorisierung	504
15.3	Privacy-Aspekte in Datenbanken	504
15.3.1	Statistische Datenbanken	505
15.3.2	k-Anonymität	507
15.4	Zusammenfassung	508
15.5	Vertiefende Literatur	509
15.6	Übungsaufgaben	509

III	Weitere Datenbankmodelle	511
16	Historische Modelle	513
16.1	Das Netzwerkmodell	513
16.1.1	Netzwerkschema	514
16.1.2	Simulation einer allgemeinen Beziehung	516
16.2	Das hierarchische Modell	517
16.3	ER-Abbildung auf das Netzwerkmodell	518
16.4	ER-Abbildung auf das hierarchische Modell	519
16.5	Anwendungsprogrammierung in den historischen Modellen	521
16.5.1	Datenmanipulation im Netzwerkmodell	521
16.5.2	Datenmanipulation im hierarchischen Modell	525
16.6	Zusammenfassung	527
16.7	Vertiefende Literatur	527
16.8	Übungsaufgaben	527
17	Objektrelationale Modelle & SQL:2003	529
17.1	Exkurs: Objektorientierte Datenbankmodelle	529
17.1.1	Objektorientierte Datenbanken	530
17.1.2	Der ODMG-Standard	531
17.1.3	OQL	534
17.2	Abbildung von Objekten auf Relationen	536
17.2.1	Typkonstruktoren	536
17.2.2	Abbildung der Spezialisierungshierarchie	537
17.3	Objektrelationale Erweiterungen	540
17.3.1	Large Objects: BLOB und CLOB	541
17.3.2	Typkonstruktoren	541
17.3.3	Identitäten, Referenzen und Pfadausdrücke	545
17.3.4	Hierarchien und Vererbung	546
17.3.5	Methoden	547
17.4	Objektrelationale Konzepte in SQL:2003	548
17.4.1	Typsystem und DDL	548
17.4.2	Anfragen	556
17.4.3	Methoden in SQL:2003	561
17.5	Objektrelationale Konzepte in kommerziellen DBMS	562
17.6	Zusammenfassung	564
17.7	Vertiefende Literatur	565
17.8	Übungsaufgaben	566
18	XML	567
18.1	Semistrukturierte Datenmodelle	567
18.1.1	Merkmale semistrukturierter Datenmodelle	568
18.1.2	Datenmodelle für semistrukturierte Dokumente	569
18.1.3	XML	572

18.2	Datendefinition in XML	578
18.2.1	Dokumenttypdefinition	578
18.2.2	XML Schema	582
18.2.3	XML-Abbildung auf relationale Schemata	587
18.3	Navigation in XML-Dokumenten: XPath	588
18.3.1	Pfadausdrücke und Lokalisierungsschritte	589
18.3.2	Selektionsprädikate und Funktionen	593
18.4	Die Anfragesprache XQuery	595
18.4.1	FLWOR-Ausdrücke	596
18.4.2	Elementkonstruktoren	599
18.4.3	Verbunde und Gruppierungen	601
18.4.4	Ausdrücke und Vergleiche	605
18.4.5	Funktionen	608
18.5	SQL/XML: XML-Erweiterungen für SQL	609
18.5.1	XML-Datentypen	610
18.5.2	XML-Konstruktion mit SQL	611
18.6	Zusammenfassung	616
18.7	Vertiefende Literatur	616
18.8	Übungsaufgaben	618

19 Data Warehousing und OLAP 619

19.1	Grundkonzepte	619
19.1.1	Motivation und Anwendungen	620
19.1.2	Architektur	622
19.2	Multidimensionales Datenmodell	625
19.2.1	Multidimensionale Daten: Der Datenwürfel	625
19.2.2	Hierarchische Dimensionen	626
19.2.3	Formalisierung von Dimensionen und Datenwürfel	627
19.2.4	Summierbarkeit	629
19.3	MOLAP und ROLAP	630
19.3.1	MOLAP-Operationen	631
19.3.2	OLAP-Anfragesprachen: MDX	634
19.3.3	Snowflake- und Star-Schema	638
19.4	OLAP-Operationen und SQL	640
19.4.1	Relationale Umsetzung multidimensionaler Anfragen: Star Join	640
19.4.2	cube und rollup	643
19.4.3	OLAP-Funktionen in SQL:2003	647
19.5	Materialisierte Sichten	653
19.5.1	Anfragebeantwortung mit materialisierter Sichten	654
19.5.2	Auswahl materialisierter Sichten	659
19.5.3	Aktualisierung materialisierter Sichten	661
19.5.4	Materialisierte Sichten in DBMS	666

19.6	Zusammenfassung	669
19.7	Vertiefende Literatur	669
19.8	Übungsaufgaben	670
20	Multimediale und raumbezogene Daten	673
20.1	Multimedia-Datenbanken	674
20.1.1	Grundbegriffe	674
20.1.2	Grundlagen des Multimedia Retrieval	678
20.2	Text Retrieval	685
20.2.1	Information Retrieval auf Texten	685
20.2.2	Grundtechniken des Text Retrieval	685
20.2.3	Deskribierung	687
20.2.4	Recherche	690
20.2.5	Information Retrieval-Systeme	698
20.3	SQL/MM	699
20.3.1	SQL/MM Full Text	699
20.3.2	SQL/MM Still Image	702
20.3.3	Der Datentyp Video	702
20.3.4	SQL/MM Spatial	703
20.4	Verwaltung raumbezogener Daten	703
20.4.1	Grundbegriffe	704
20.4.2	Modellierung raumbezogener Daten	705
20.4.3	Prädikate und Anfragen auf raumbezogenen Daten	711
20.4.4	Oracle Spatial	718
20.4.5	Weitere Systeme	722
20.5	Zusammenfassung	722
20.6	Vertiefende Literatur	722
20.7	Übungsaufgaben	724
A	Laufendes Beispiel	725
A.1	ER-Schema der Weindatenbank	725
A.2	Relationale Repräsentation	726
A.3	Vereinfachtes Schema und Beispieldaten	727
	Literaturverzeichnis	729
	Abbildungsverzeichnis	751
	Tabellenverzeichnis	758
	Sachindex	761
	Schlüsselwortindex	777

Grundlegende Konzepte

Dieses erste Kapitel ist den grundlegenden Konzepten der Datenbankterminologie und -technik gewidmet. Wir werden uns die historische Entwicklung von Datenbanksystemen ansehen, Gründe für den Einsatz von derartigen Systemen diskutieren sowie Funktionen und Architektur von Datenbanksystemen betrachten. Ferner stellen wir als eine Beispielanwendung eine Weinkellerverwaltung vor, die wir über das ganze Buch hinweg verwenden werden.

1.1 Motivation und Historie

Wie ordnen sich Datenbanksysteme in die Vielfalt von Softwarepaketen ein, die heutzutage eingesetzt werden? Zur Beantwortung dieser Frage diskutieren wir zuerst eine verbreitete Klassifikation von Softwaresystemen.

Softwareschichten

Üblicherweise teilt man die Software eines Computersystems in mehrere Schichten ein, etwa der Aufteilung in Abbildung 1.1 folgend. In der Praxis können natürlich einige *Softwarepakete* mehrere Schichten umfassen.

Jede Schicht baut auf den weiter innen liegenden Schichten auf. Beispielsweise bietet das Betriebssystem Dateien und Operationen auf Dateien, Möglichkeiten zum Drucken etc. an. Anwendungssoftware wie Textverarbeitungssoftware nutzt diese Möglichkeiten als Dienste der niedrigeren Schicht. Als Beispiele für typische Softwareprodukte auf den einzelnen Schichten mag die folgende Auswahl dienen:

- Typische *Betriebssysteme* sind etwa Windows, Linux, MacOS X oder z/OS.

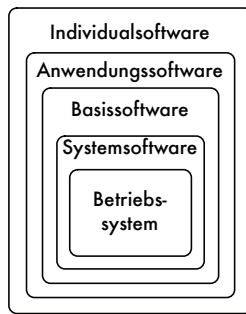


Abbildung 1.1: Aufteilung in Softwareschichten

- Zur *Systemsoftware*, die direkt auf diesen Betriebssystemen aufbaut, zählen Datenbanksysteme und Benutzerschnittstellen (wie das Windows-GUI oder X11-Produkte unter Unix).
- Zur *Basissoftware*, die wiederum auf der Systemsoftware aufbaut, gehören etwa Graphiksysteme wie OpenGL.
- *Anwendungs- und Individualsoftware* ist auf bestimmte Anwendungsklassen hin zugeschnitten: CAD-Systeme für Konstruktionsanwendungen, Desktop-Publishing-Systeme für Publikationsanwendungen sowie Buchhaltungssysteme, Lagerverwaltungssysteme oder allgemeiner ERP-Systeme (Enterprise Resource Planning) zur Unterstützung aller Geschäftsprozesse in Unternehmen.

Die Rolle der Datenbanksysteme ist also eine sehr elementare. Idealerweise sollten selbst Textverarbeitungssysteme ihre Texte und Informationen über Texte in einem Datenbanksystem verwalten und nicht einfach in einem Dateisystem. Genauso sollten CAD-Systeme sich allgemeinerer Graphiksysteme bedienen und diese wiederum zur Speicherung von Graphiken auf Datenbanksysteme zurückgreifen. Die Welt der kommerziellen Software ist von dieser Idealvorstellung jedoch leider noch etwas entfernt.

Das Problem der Datenredundanz

Ohne den Einsatz von Datenbanksystemen tritt das Problem der *Datenredundanz* auf. Die Basis- oder Anwendungssoftware verwaltet in diesem Szenario jeweils ihre eigenen Daten in ihren eigenen Dateien, und zwar jeweils in eigenen speziellen Formaten. Ein typisches Szenario gibt die folgende Auflistung wieder:

- Ein Textverarbeitungssystem verwaltet Texte, Artikel und Adressen.

- Die Buchhaltung speichert ebenso Artikel- und Adressinformationen.
- In der Lagerverwaltung werden Artikel und Aufträge benötigt und verwendet.
- Die Auftragsverwaltung manipuliert Aufträge, Artikel und Kundenadressen.
- Das CAD-System verwaltet Artikeldaten, technische Daten und technische Bausteine.
- Die Bereiche Produktion, Bestelleingang und Kalkulation benötigen teilweise ebenfalls diese Daten.

In diesem Szenario sind die Daten *redundant*, also mehrfach gespeichert. So werden Artikel und Adressen von mehreren Anwendungen verwaltet. Die entstehenden Probleme sind Verschwendung von Speicherplatz und „Vergessen“ von lokalen Änderungen, die typisch für das Fehlen einer zentralen, genormten Datenhaltung sind. Ein Ziel der Entwicklung von Datenbanksystemen ist die Beseitigung der Datenredundanz.

Weitere Problemfelder

Die meisten anderen Softwaresysteme (auch Programmiersprachen, Tabellenkalkulation, Dateiverwaltungssysteme ...) können große Mengen von Daten nicht *effizient* verarbeiten, so dass fehlender Einsatz von Datenbankmanagementsystemen (DBMS) zu erheblichen Effizienzeinbußen führen kann. Auch ermöglichen es viele Systeme nicht, dass mehrere Benutzer oder Anwendungen *parallel* mit den gleichen Daten arbeiten können, ohne einander zu stören. Weiterhin können gar Datenverluste durch unkontrolliertes Überschreiben entstehen. Diese Kontrolle ist eine Basisfunktion moderner DBMS.

Auch in der Anwendungserstellung führt der fehlende Einsatz einer zentralen Datenhaltungskomponente zu erheblichen Defiziten. Die Anwendungsprogrammierer oder auch Endanwender können Anwendungen nicht programmieren bzw. benutzen, ohne

- die interne Darstellung der Daten sowie
- Speichermedien oder Rechner (bei verteilten Systemen)

zu kennen. Dieses Problem wird als fehlende *Datenunabhängigkeit* bezeichnet und in Abschnitt 2.1 intensiver diskutiert. Auch ist die Sicherstellung der *Zugriffskontrolle* und der *Datensicherheit* ohne zentrale Datenhaltung nicht gewährleistet.