

Inhaltsverzeichnis

Vorwort	17
1 Einleitung und Übersicht	19
1.1 Motivation für den Einsatz eines DBMS	19
1.2 Datenabstraktion	21
1.3 Datenunabhängigkeit	22
1.4 Datenmodelle	23
1.5 Datenbankschema und Ausprägung	24
1.6 Einordnung der Datenmodelle	24
1.6.1 Modelle des konzeptuellen Entwurfs	24
1.6.2 Logische (Implementations-)Datenmodelle	25
1.7 Architekturübersicht eines DBMS	28
1.8 Übungen	30
1.9 Literatur	30
2 Datenbankentwurf	31
2.1 Abstraktionsebenen des Datenbankentwurfs	31
2.2 Allgemeine Entwurfsmethodik	32
2.3 Die Datenbankentwurfsschritte	33
2.4 Die Anforderungsanalyse	33
2.4.1 Informationsstrukturanforderungen	35
2.4.2 Datenverarbeitungsanforderungen	37
2.5 Grundlagen des Entity-Relationship-Modells	37
2.6 Schlüssel	39
2.7 Charakterisierung von Beziehungstypen	39
2.7.1 Funktionalitäten der Beziehungen	39
2.7.2 Funktionalitätsangaben bei n -stelligen Beziehungen	41
2.7.3 Die (min, max) -Notation	44
2.8 Existenzabhängige Entitytypen	48
2.9 Generalisierung	49
2.10 Aggregation	50
2.11 Kombination von Generalisierung und Aggregation	52
2.12 Konsolidierung, Sichtenintegration	53
2.13 Konzeptuelle Modellierung mit UML	59
2.13.1 UML-Klassen	59
2.13.2 Assoziationen zwischen Klassen	60
2.13.3 Aggregation in UML	61
2.13.4 Anwendungsbeispiel: Begrenzungsflächendarstellung von Poly- edern in UML	62
2.13.5 Generalisierung in UML-Notation	63
2.13.6 Die Modellierung der Universität in UML	63
2.13.7 Verhaltensmodellierung in UML	64
2.13.8 Anwendungsfall-Modellierung (use cases)	64

2.13.9	Interaktionsdiagramme	66
2.13.10	Interaktionsdiagramm zur Prüfungsdurchführung	66
2.14	Übungen	67
2.15	Literatur	69
3	Das relationale Modell	71
3.1	Definition des relationalen Modells	71
3.1.1	Mathematischer Formalismus	71
3.1.2	Schema-Definition	72
3.2	Umsetzung eines konzeptuellen Schemas in ein relationales Schema	73
3.2.1	Relationale Darstellung von Entitytypen	73
3.2.2	Relationale Darstellung von Beziehungen	73
3.3	Verfeinerung des relationalen Schemas	78
3.3.1	1:N-Beziehungen	78
3.3.2	1:1-Beziehungen	80
3.3.3	Vermeidung von Null-Werten	81
3.3.4	Relationale Modellierung der Generalisierung	82
3.3.5	Beispielausprägung der Universitäts-Datenbank	83
3.3.6	Relationale Modellierung schwacher Entitytypen	85
3.4	Die relationale Algebra	85
3.4.1	Selektion	86
3.4.2	Projektion	87
3.4.3	Vereinigung	87
3.4.4	Mengendifferenz	88
3.4.5	Kartesisches Produkt (Kreuzprodukt)	88
3.4.6	Umbenennung von Relationen und Attributen	89
3.4.7	Definition der relationalen Algebra	90
3.4.8	Der relationale Verbund (Join)	90
3.4.9	Mengendurchschnitt	95
3.4.10	Die relationale Division	96
3.4.11	Gruppierung und Aggregation	97
3.4.12	Operatorbaum-Darstellung	98
3.5	Der Relationenkalkül	98
3.5.1	Beispielanfrage im relationalen Tupelkalkül	99
3.5.2	Quantifizierung von Tupelvariablen	100
3.5.3	Formale Definition des Tupelkalküls	101
3.5.4	Sichere Ausdrücke des Tupelkalküls	102
3.5.5	Der relationale Domänenkalkül	102
3.5.6	Beispielanfragen im Domänenkalkül	103
3.5.7	Sichere Ausdrücke des Domänenkalküls	104
3.6	Ausdruckskraft der Anfragesprachen	105
3.7	Übungen	105
3.8	Literatur	108
4	Relationale Anfragesprachen	111
4.1	Geschichte	111
4.2	Datentypen	112
4.3	Schemadefinition	112
4.4	Schemaveränderung	113
4.5	Elementare Datenmanipulation: Einfügen von Tupeln	114

4.6	Einfache SQL-Anfragen	114
4.7	Anfragen über mehrere Relationen	115
4.8	Aggregatfunktionen und Gruppierung	118
4.9	Geschachtelte Anfragen	119
4.10	Quantifizierte Anfragen in SQL	124
4.11	Nullwerte	126
4.12	Spezielle Sprachkonstrukte	127
4.13	Joins in SQL-92	129
4.14	Rekursion	129
4.15	Veränderungen am Datenbestand	134
4.16	Sichten	136
4.17	Sichten zur Modellierung von Generalisierungen	137
4.18	Charakterisierung update-fähiger Sichten	139
4.19	Einbettung von SQL in Wirtssprachen	140
4.20	Anfragen in Anwendungsprogrammen	141
4.21	JDBC: Java Database Connectivity	144
4.21.1	Verbindungsaufbau zu einer Datenbank	145
4.21.2	Resultset-Programmbeispiel	147
4.21.3	Vorübersetzung von SQL-Ausdrücken	149
4.22	SQLJ: Eine Einbettung von SQL in Java	149
4.23	Query by Example	152
4.24	Übungen	154
4.25	Literatur	158
5	Datenintegrität	159
5.1	Referentielle Integrität	160
5.2	Gewährleistung referentieller Integrität	160
5.3	Referentielle Integrität in SQL	161
5.4	Überprüfung statischer Integritätsbedingungen	162
5.5	Das Universitätsschema mit Integritätsbedingungen	164
5.6	Komplexere Integritätsbedingungen	166
5.7	Trigger	167
5.8	Übungen	169
5.9	Literatur	170
6	Relationale Entwurfstheorie	171
6.1	Funktionale Abhängigkeiten	171
6.1.1	Konventionen zur Notation	172
6.1.2	Einhaltung einer funktionalen Abhängigkeit	172
6.2	Schlüssel	173
6.3	Bestimmung funktionaler Abhängigkeiten	174
6.3.1	Kanonische Überdeckung	177
6.4	„Schlechte“ Relationenschemata	178
6.4.1	Die Updateanomalien	178
6.4.2	Einfügeanomalien	179
6.4.3	Löschanomalien	179
6.5	Zerlegung (Dekomposition) von Relationen	179
6.5.1	Verlustlosigkeit	180
6.5.2	Kriterien für die Verlustlosigkeit einer Zerlegung	182
6.5.3	Abhängigkeitsbewahrung	183

6.6	Erste Normalform	185
6.7	Zweite Normalform	186
6.8	Dritte Normalform	188
6.9	Boyce-Codd Normalform	190
6.10	Mehrwertige Abhängigkeiten	193
6.11	Vierte Normalform	195
6.12	Zusammenfassung	197
6.13	Übungen	198
6.14	Literatur	202
7	Physische Datenorganisation	203
7.1	Speichermedien	203
7.2	Speicherhierarchie	204
7.3	Speicherarrays: RAID	206
7.4	Der Datenbankpuffer	210
7.5	Abbildung von Relationen auf den Sekundärspeicher	211
7.6	Indexstrukturen	213
7.7	ISAM	214
7.8	B-Bäume	216
7.9	B ⁺ -Bäume	220
7.10	Präfix-B ⁺ -Bäume	222
7.11	Hashing	222
7.12	Erweiterbares Hashing	224
7.13	Mehrdimensionale Indexstrukturen	226
7.14	Ballung logisch verwandter Datensätze	231
7.15	Unterstützung eines Anwendungsverhaltens	232
7.16	Physische Datenorganisation in SQL	235
7.17	Übungen	235
7.18	Literatur	237
8	Anfragebearbeitung	241
8.1	Logische Optimierung	242
8.1.1	Äquivalenzen in der relationalen Algebra	244
8.1.2	Anwendung der Transformationsregeln	246
8.2	Physische Optimierung	250
8.2.1	Implementierung der Selektion	252
8.2.2	Implementierung von binären Zuordnungsoperatoren	252
8.2.3	Gruppierung und Duplikateliminiierung	259
8.2.4	Projektion und Vereinigung	260
8.2.5	Zwischenspeicherung	260
8.2.6	Sortierung von Zwischenergebnissen	261
8.2.7	Übersetzung der logischen Algebra	265
8.3	Kostenmodelle	268
8.3.1	Selektivitäten	268
8.3.2	Kostenabschätzung für die Selektion	271
8.3.3	Kostenabschätzung für den Join	271
8.3.4	Kostenabschätzung für die Sortierung	272
8.4	„Tuning“ von Datenbankabfragen	272
8.5	Kostenbasierte Optimierer	274
8.5.1	Suchraum für die Join-Optimierung	274

8.5.2	Dynamische Programmierung	276
8.6	Übungen	280
8.7	Literatur	282
9	Transaktionsverwaltung	285
9.1	Begriffsbildung	285
9.2	Anforderungen an die Transaktionsverwaltung	286
9.3	Operationen auf Transaktions-Ebene	286
9.4	Abschluss einer Transaktion	287
9.5	Eigenschaften von Transaktionen	289
9.6	Transaktionsverwaltung in SQL	290
9.7	Zustandsübergänge einer Transaktion	291
9.8	Literatur	292
10	Fehlerbehandlung	293
10.1	Fehlerklassifikation	293
10.1.1	Lokaler Fehler einer Transaktion	293
10.1.2	Fehler mit Hauptspeicherverlust	294
10.1.3	Fehler mit Hintergrundspeicherverlust	295
10.2	Die Speicherhierarchie	295
10.2.1	Ersetzung von Puffer-Seiten	295
10.2.2	Einbringen von Änderungen einer Transaktion	296
10.2.3	Einbringstrategie	297
10.2.4	Hier zugrunde gelegte Systemkonfiguration	298
10.3	Protokollierung von Änderungsoperationen	298
10.3.1	Struktur der Log-Einträge	299
10.3.2	Beispiel einer Log-Datei	299
10.3.3	Logische oder physische Protokollierung	299
10.3.4	Schreiben der Log-Information	300
10.3.5	Das WAL-Prinzip	302
10.4	Wiederanlauf nach einem Fehler	302
10.4.1	Analyse des Logs	303
10.4.2	Redo-Phase	304
10.4.3	Undo-Phase	304
10.5	Fehlertoleranz des Wiederanlaufs	304
10.6	Lokales Zurücksetzen einer Transaktion	306
10.7	Partielles Zurücksetzen einer Transaktion	307
10.8	Sicherungspunkte	308
10.8.1	Transaktionskonsistente Sicherungspunkte	308
10.8.2	Aktionskonsistente Sicherungspunkte	309
10.8.3	Unschärfe (fuzzy) Sicherungspunkte	311
10.9	Recovery nach einem Verlust der materialisierten Datenbasis	312
10.10	Übungen	313
10.11	Literatur	314
11	Mehrbenutzersynchronisation	315
11.1	Fehler bei unkontrolliertem Mehrbenutzerbetrieb	316
11.1.1	Verlorengegangene Änderungen (<i>lost update</i>)	316
11.1.2	Abhängigkeit von nicht freigegebenen Änderungen	316
11.1.3	Phantomproblem	317

11.2	Serialisierbarkeit	317
11.2.1	Beispiele serialisierbarer Ausführungen (Historien)	318
11.2.2	Nicht serialisierbare Historie	318
11.3	Theorie der Serialisierbarkeit	321
11.3.1	Definition einer Transaktion	321
11.3.2	Historie (Schedule)	322
11.3.3	Äquivalenz zweier Historien	323
11.3.4	Serialisierbare Historien	324
11.3.5	Kriterien für Serialisierbarkeit	324
11.4	Eigenschaften von Historien bezüglich der Recovery	326
11.4.1	Rücksetzbare Historien	326
11.4.2	Historien ohne kaskadierendes Rücksetzen	326
11.4.3	Strikte Historien	327
11.4.4	Beziehungen zwischen den Klassen von Historien	327
11.5	Der Datenbank-Scheduler	328
11.6	Sperrbasierte Synchronisation	329
11.6.1	Zwei Sperrmodi	329
11.6.2	Zwei-Phasen-Sperrprotokoll	330
11.6.3	Kaskadierendes Rücksetzen (Schneeballeffekt)	332
11.7	Verklemmungen (Deadlocks)	332
11.7.1	Erkennung von Verklemmungen	333
11.7.2	Preclaiming zur Vermeidung von Verklemmungen	334
11.7.3	Verklemmungsvermeidung durch Zeitstempel	335
11.8	Hierarchische Sperrgranulate	336
11.9	Einfüge- und Löschooperationen, Phantome	340
11.10	Zeitstempel-basierende Synchronisation	341
11.11	Optimistische Synchronisation	343
11.12	Klassifizierung der Verfahren	344
11.13	Synchronisation von Indexstrukturen	344
11.14	Mehrbenutzersynchronisation in SQL-92	348
11.15	Übungen	350
11.16	Literatur	353
12	Sicherheitsaspekte	355
12.1	Discretionary Access Control	357
12.2	Zugriffskontrolle in SQL	357
12.2.1	Identifikation und Authentisierung	358
12.2.2	Autorisierung und Zugriffskontrolle	358
12.2.3	Sichten	359
12.2.4	Individuelle Sicht für eine Benutzergruppe	360
12.2.5	k-Anonymität	361
12.2.6	Auditing	361
12.3	Verfeinerung des Autorisierungsmodells	362
12.3.1	Rollenbasierte Autorisierung: Implizite Autorisierung von Subjekten	363
12.3.2	Implizite Autorisierung von Operationen	364
12.3.3	Implizite Autorisierung von Objekten	364
12.3.4	Implizite Autorisierung entlang einer Typhierarchie	365
12.4	Mandatory Access Control	367

12.5	Multilevel-Datenbanken	367
12.6	SQL-Injection	370
12.6.1	Attacken	371
12.6.2	Schutz vor SQL-Injection-Attacken	372
12.7	Kryptographie	374
12.7.1	Der Data Encryption Standard	374
12.7.2	Public-Key Kryptographie	376
12.7.3	Public Key Infrastruktur (PKI)	377
12.8	Zusammenfassung	378
12.9	Übungen	378
12.10	Literatur	380
13	Objektorientierte Datenbanken	381
13.1	Bestandsaufnahme relationaler Datenbanksysteme	381
13.2	Vorteile der objektorientierten Datenmodellierung	385
13.3	Der ODMG-Standard	386
13.4	Eigenschaften von Objekten	387
13.4.1	Objektidentität	388
13.4.2	Typ eines Objekts	389
13.4.3	Wert eines Objekts	389
13.5	Definition von Objekttypen	390
13.5.1	Attribute	390
13.5.2	Beziehungen	390
13.5.3	Typeigenschaften: Extensionen und Schlüssel	397
13.6	Modellierung des Verhaltens: Operationen	397
13.7	Vererbung und Subtypisierung	400
13.7.1	Terminologie	400
13.7.2	Einfache und Mehrfachvererbung	401
13.8	Beispiel einer Typhierarchie	402
13.9	Verfeinerung (Spezialisierung) und spätes Binden von Operationen	405
13.10	Mehrfachvererbung	408
13.11	Die Anfragesprache OQL	409
13.11.1	Einfache Anfragen	409
13.11.2	Geschachtelte Anfragen und Partitionierung	410
13.11.3	Pfadausdrücke	411
13.11.4	Erzeugung von Objekten	412
13.11.5	Operationsaufruf	412
13.12	C++-Einbettung	412
13.12.1	Objektidentität	414
13.12.2	Objekterzeugung und Ballung	415
13.12.3	Einbettung von Anfragen	415
13.13	Übungen	416
13.14	Literatur	417
14	Erweiterbare und objekt-relationale Datenbanken	419
14.1	Übersicht über die objekt-relationalen Konzepte	419
14.2	Large Objects (LOBs)	420
14.3	Distinct Types: Einfache benutzerdefinierte Datentypen	422
14.4	Table Functions	426
14.4.1	Nutzung einer <i>Table Function</i> in Anfragen	427

14.4.2	Implementierung einer <i>Table Function</i>	427
14.5	Benutzerdefinierte strukturierte Objekttypen	429
14.6	Geschachtelte Objekt-Relationen	433
14.7	Vererbung von SQL-Objekttypen	437
14.8	Komplexe Attribut-Typen	440
14.9	Übungen	441
14.10	Literatur	442
15	Deduktive Datenbanken	443
15.1	Terminologie	443
15.2	Datalog	443
15.3	Eigenschaften von Datalog-Programmen	447
15.3.1	Rekursivität	447
15.3.2	Sicherheit von Datalog-Regeln	447
15.4	Auswertung von nicht-rekursiven Datalog-Programmen	448
15.4.1	Auswertung eines Beispielprogramms	448
15.4.2	Auswertungs-Algorithmus	451
15.5	Auswertung rekursiver Regeln	453
15.6	Inkrementelle (semi-naive) Auswertung rekursiver Regeln	455
15.7	Bottom-Up oder Top-Down Auswertung	459
15.8	Negation im Regelrumpf	461
15.8.1	Stratifizierte Datalog-Programme	461
15.8.2	Auswertung von Regeln mit Negation	462
15.8.3	Ein etwas komplexeres Beispiel	463
15.9	Ausdruckskraft von Datalog	463
15.10	Übungen	465
15.11	Literatur	469
16	Verteilte Datenbanken	471
16.1	Terminologie und Abgrenzung	471
16.2	Entwurf verteilter Datenbanken	473
16.3	Horizontale und vertikale Fragmentierung	475
16.3.1	Horizontale Fragmentierung	476
16.3.2	Abgeleitete horizontale Fragmentierung	478
16.3.3	Vertikale Fragmentierung	479
16.3.4	Kombinierte Fragmentierung	481
16.3.5	Allokation für unser Beispiel	482
16.4	Transparenz in verteilten Datenbanken	483
16.4.1	Fragmentierungstransparenz	483
16.4.2	Allokationstransparenz	484
16.4.3	Lokale Schema-Transparenz	484
16.5	Anfrageübersetzung und -optimierung in VDBMS	485
16.5.1	Anfragebearbeitung bei horizontaler Fragmentierung	485
16.5.2	Anfragebearbeitung bei vertikaler Fragmentierung	487
16.6	Join-Auswertung in VDBMS	489
16.6.1	Join-Auswertung ohne Filterung	489
16.6.2	Join-Auswertung mit Semijoin-Filterung	490
16.6.3	Join-Auswertung mit Bitmap-Filterung	492
16.7	Transaktionskontrolle in VDBMS	494
16.8	Mehrbenutzersynchronisation in VDBMS	499

16.8.1	Serialisierbarkeit	499
16.8.2	Das Zwei-Phasen-Sperrprotokoll in VDBMS	499
16.9	Deadlocks in VDBMS	500
16.9.1	Erkennung von Deadlocks	500
16.9.2	Deadlock-Vermeidung	503
16.10	Synchronisation bei replizierten Daten	504
16.11	Peer-to-Peer-Informationssysteme	507
16.11.1	P2P-Systeme für den Datenaustausch (File-Sharing)	507
16.11.2	Verteilte Hashtabellen (Distributed Hash Tables DHTs)	509
16.11.3	Mehrdimensionaler P2P-Datenraum	513
16.12	No-SQL- und Key/Value-Datenbanksysteme	515
16.13	Übungen	516
16.14	Literatur	520
17	Betriebliche Anwendungen: OLTP, Data Warehouse, Data Mining	523
17.1	SAP R/3: Ein betriebswirtschaftliches Datenbankanwendungssystem	523
17.1.1	Architektur von SAP R/3	523
17.1.2	Datenmodell und Schema von SAP R/3	524
17.1.3	ABAP/4	525
17.1.4	Transaktionen in SAP R/3	528
17.2	Data Warehouse, Decision-Support, OLAP	529
17.2.1	Datenbankentwurf für das Data Warehouse	530
17.2.2	Anfragen im Sternschema: Star Join	533
17.2.3	Roll-Up/Drill-Down-Anfragen	534
17.2.4	Flexible Auswertungsmethoden	536
17.2.5	Materialisierung von Aggregaten	536
17.2.6	Der cube -Operator	538
17.2.7	Wiederverwendung materialisierter Aggregate	538
17.2.8	Bitmap-Indices für OLAP-Anfragen	541
17.2.9	Auswertungsalgorithmen für komplexe OLAP-Anfragen	542
17.3	Bewertung (Ranking) von Objekten	544
17.3.1	Top-k-Anfragen	545
17.3.2	Skyline-Anfragen	549
17.3.3	Data Warehouse-Architekturen	552
17.3.4	Attribut-basierte Speicherung (Column-Stores)	553
17.4	Data Mining	555
17.4.1	Klassifikation von Objekten	555
17.4.2	Assoziationsregeln	556
17.4.3	Der À Priori-Algorithmus	557
17.4.4	Bestimmung der Assoziationsregeln	559
17.4.5	Cluster-Bestimmung	560
17.5	Übungen	561
17.6	Literatur	562
18	Internet-Datenbankanbindungen	565
18.1	HTML- und HTTP-Grundlagen	565
18.1.1	HTML: Die Hypertext-Sprache des World Wide Web	565
18.1.2	Adressierung von Web-Dokumenten	566
18.1.3	Client/Server-Architektur des World Wide Web	568
18.1.4	HTTP: Das HyperText Transfer Protokoll	568

18.1.5	HTTPS	569
18.2	Web-Datenbank-Anbindung via Servlets	570
18.2.1	Beispiel-Servlet	570
18.3	Java Server Pages / Active Server Pages	576
18.3.1	JSP/HTML-Seite mit Java-Code	577
18.3.2	HTML-Seite mit Java-Bean-Aufruf	579
18.3.3	Die Java-Bean Komponente <i>VorlesungenBean</i>	580
18.3.4	Sokrates' Homepage	582
18.4	Datenbankanbindung via Java-Applets	582
18.5	Übungen	583
18.6	Literatur	584
19	XML-Datenmodellierung und Web-Services	587
19.1	XML-Datenmodellierung	587
19.1.1	Schema oder kein Schema	588
19.1.2	Rekursive Schemata	590
19.1.3	Universitätsinformation in XML-Format	590
19.1.4	XML-Namensräume	592
19.1.5	XML Schema: Eine Schemadefinitionssprache	594
19.1.6	Verweise (Referenzen) in XML-Daten	596
19.2	XQuery: Eine XML-Anfragesprache	597
19.2.1	Pfadausdrücke	597
19.2.2	Verkürzte XPath-Syntax	602
19.2.3	Beispiel-Pfadausdrücke in verkürzter Syntax	603
19.2.4	Anfragesyntax von XQuery	604
19.2.5	Geschachtelte Anfragen	606
19.2.6	Joins in XQuery	606
19.2.7	Join-Prädikat im Pfadausdruck	607
19.2.8	Das let-Konstrukt	608
19.2.9	Dereferenzierung in FLWOR-Ausdrücken	609
19.2.10	Das if-then-else-Konstrukt	611
19.2.11	Rekursive Anfragen	612
19.3	Zusammenspiel von relationalen Datenbanken und XML	614
19.3.1	XML-Repräsentation gemäß Pre- und Postorder-Rängen	620
19.3.2	Der neue Datentyp xml	624
19.3.3	Änderungen der XML-Dokumente	628
19.3.4	Publikation relationaler Daten als XML-Dokumente	629
19.3.5	Fallstudie: XML-Unterstützung in IBM DB2 V9	633
19.4	Web-Services	638
19.4.1	Erstellen und Nutzen eines Web-Services im Überblick	640
19.4.2	Das Auffinden von Diensten	642
19.4.3	Ein Beispiel-Web-Service	644
19.4.4	Definition der Web-Service-Schnittstellen	644
19.4.5	Nachrichtenformat für die Interaktion mit Web-Services	647
19.4.6	Implementierung des Web-Services	649
19.4.7	Aufruf des Web-Services	650
19.5	Übungen	652
19.6	Literatur	655

20 Neue Datenbank-Entwicklungen	659
20.1 Datenbanken für das Semantic Web	659
20.1.1 RDF: Resource Description Framework	659
20.1.2 SPARQL: Die RDF Anfragesprache	661
20.1.3 Implementierung einer RDF-Datenbank	664
20.2 Hauptspeicher-Datenbanken für OLTP&OLAP	668
20.2.1 Update Staging	671
20.2.2 Heterogene Workload-Verwaltung	672
20.2.3 Kontinuierliche Datawarehouse-Auffrischung	673
20.2.4 Versionierung der transaktionalen Daten	673
20.2.5 Batch-Verarbeitung	673
20.2.6 Das Schattenspeicher-Konzept	674
20.2.7 Berechnete Snapshots	675
20.2.8 Reduzierte Isolationsstufen	676
20.2.9 Snapshots des virtuellen Speichers	676
20.2.10 Transaktionsverwaltung im Hauptspeicher-DBMS	679
20.3 Datenströme	681
20.4 Information Retrieval und Suchmaschinen	688
20.4.1 TF-IDF: Dokument-Ranking basierend auf Begriffs-Häufigkeit	689
20.4.2 Invertierte Indexierung	690
20.4.3 Page Rank	691
20.4.4 Der HITS Algorithmus	694
20.5 MapReduce: Massiv parallele Datenverarbeitung	697
20.6 Multi-Tenancy, Cloud Computing und Software as a Service	702
20.7 Literatur	708
21 Leistungsbewertung	711
21.1 Überblick über Datenbanksystem-Benchmarks	711
21.2 Der TPC-C Benchmark	711
21.3 Die TPC-H und TPC-R (früher TPC-D) Benchmarks	714
21.4 Der OO7 Benchmark für oo-Datenbanken	720
21.5 Hybrider OLTP&OLAP-Benchmark: TPC-CH	721
21.6 Der TPC-W Benchmark	723
21.7 Neue TPC-Benchmarks	726
21.7.1 TPC-E: Der neue OLTP-Benchmark	726
21.7.2 TPC-App: der neue Webservice-Benchmark	727
21.7.3 TPC-DS: der neue Decision Support Benchmark	729
21.8 Übungen	729
21.9 Literatur	730
Literaturverzeichnis	733
Index	775

Vorwort

Wir drohen derzeit von einer wahren Informationsflut „überrollt“ zu werden und sind auf dem besten Weg in die Informationsgesellschaft. Datenbanksysteme spielen eine immer größere Rolle in Unternehmen, Behörden und anderen Organisationen. Ihre Bedeutung wird durch die zunehmende weltweite Vernetzung – Internet und World Wide Web – noch stärker wachsen. Gleichzeitig wird der systematische Einsatz von Datenbanksystemen wegen der zunehmenden Informationsmenge, der Verteilung der Information auf ein Netz von Datenbankservern, der steigenden Komplexität der Anwendungen und der erhöhten Leistungsanforderungen immer schwieriger – auch wenn sich die Datenbanksystemprodukte weiterentwickeln.

In diesem Buch zur Einführung in Datenbanksysteme haben wir die Lehrinhalte zusammengestellt, die nach unserer Meinung für alle Informatik-nahen Studiengänge an Universitäten oder Fachhochschulen – wie z.B. Informatik, Software-Engineering, Wirtschafts-Informatik, Bio-Informatik etc. – essenziell sind.

Im Vergleich zu anderen Datenbank-Lehrbüchern setzten wir folgende Akzente:

- Es wurde ein durchgängiges Beispiel aus dem Hochschulbereich gewählt, das den Datenbankeinsatz gut illustriert. Dieses Beispiel haben wir bewusst einfach gehalten, damit man es sich gut einprägen kann.
- Das Buch eignet sich auch zum Selbststudium, da wir uns bemüht haben, alle Konzepte an gut verständlichen Beispielen zu veranschaulichen. Eine ideale Ergänzung bietet darüber hinaus das neue *Übungsbuch Datenbanksysteme* von Kemper und Wimmer (2009), das Lösungsvorschläge für die Übungsaufgaben und weitergehende (teilweise multimediale) Lernhilfen enthält.
- Das Buch behandelt nur „moderne“ Datenbanksysteme. Sehr ausführlich gehen wir auf das relationale Modell ein, da es derzeit die marktbeherrschende Rolle spielt. Es werden aber auch neuere Entwicklungen, wie XML und Web-Services behandelt. Ältere Datenmodelle (die sogenannten satzorientierten Modelle, zu denen das Netzwerkmodell und das hierarchische Modell zählen) haben wir ausgeklammert, da diese Datenbanksysteme in absehbarer Zeit wohl nur noch „historische“ Bedeutung haben werden.
- Das Buch behandelt auch Implementierungsaspekte – wie z.B. physische Strukturen für die Datenverwaltung, Realisierungskonzepte für die Mehrbenutzersynchronisation und die Recovery, Optimierungsmethoden zur Abfrageauswertung etc. Auch wenn die wenigsten Informatiker später ein Datenbanksystem „bauen“ werden, so meinen wir doch, dass ein tiefgehendes Wissen unabdingbar ist, um ein Datenbanksystem in der „harten“ industriellen Praxis systematisch einsetzen und optimieren zu können.
- Das Buch betont die praktischen Aspekte des Datenbankbereichs – ohne jedoch die theoretischen Grundlagen zu vernachlässigen. Die zugrundeliegende Theorie wird eingeführt, auf Beweise haben wir aber bewusst verzichtet.

- UML wird als objekt-orientierte Datenmodellierungs-Alternative zum ER-Modell eingeführt und die objekt-orientierten und objekt-relationalen Datenbankkonzepte werden detailliert diskutiert.
- Der Einsatz von Datenbanken als Data Warehouse für Decision Support-Anfragen sowie für das Data Mining wird beschrieben.
- Ein umfangreiches Kapitel behandelt die Realisierung von Internet-Datenbanken mit Hilfe der Java-Anbindungen (JDBC, SQLJ, Servlets und JSP).
- Die XML-Datenbanktechnologien werden ausführlich behandelt: XML-Datenmodell, XPath und XQuery als Anfragesprachen sowie XML-basierte Web Services. Weiterhin wird die XML-Unterstützung der kommerziellen relationalen Datenbanksysteme ausführlich diskutiert.
- In dieser achten Auflage wurden die Ausführungen aktualisiert und neuere Entwicklungen aufgegriffen. Techniken für die Beherrschbarkeit der Informationsflut des Webs, wie NoSQL Key-Value-Speicher, RDF/SPARQL als Grundlage des *Semantic Web*, Information Retrieval und Suchmaschinen-Grundlagen (u.a. PageRank), hochgradig verteilte Datenverarbeitung (MapReduce), Datenströme, Hauptspeicher-Datenbanken und Cloud/Multi-Tenancy Datenbanken sind einige der neuen Themen.
- Zusätzliche Unterlagen zu diesem Buch findet man über unseren Webserver (<http://www-db.in.tum.de>).

Wir haben uns bemüht, die inhaltlichen Abhängigkeiten zwischen den Kapiteln gering zu halten. Deshalb ist es problemlos möglich (und wird von uns an der TU München auch praktiziert), eine schon im Grundstudium enthaltene Einführung in Datenbanksysteme – in der beispielsweise die Grundlagen der konzeptuellen Datenmodellierung, der physischen Datenorganisation, des relationalen Datenmodells und der Anfragesprache SQL, der Datenbanktheorie und der Transaktionsverwaltung vermittelt werden – aus diesem Buch zu „extrahieren“, um das Themengebiet dann im Hauptstudium mit den übrigen Kapiteln zu vervollständigen. Es ist auch möglich, einige der weiterführenden Kapitel in einer Vorlesung zur Datenbankimplementierung oder in einer projektorientierten Datenbankeinsatz-Vorlesung zu „verwerten“.

Danksagung Dr. Reinhard Braumandl, Dr. Christian Wiesner, Dr. Jens Claußen, Dr. Carsten Gerlhof, Prof. Donald Kossmann, Dr. Natalija Krivokapić, Dr. Klaus Peithner und Dr. Michael Steinbrunn danken wir für ihre Hilfe bei früheren Auflagen. Dr. Stefan Seltzsaam, Dr. Richard Kuntschke und Dr. Martin Wimmer haben wesentlich bei der Ausarbeitung zu den Web-Datenbankschnittstellen geholfen. Frau Martina-Cezara Albutiu und Herr Stefan Kinauer haben bei der Korrektur geholfen. Wir haben von etlichen „externen“ Lesern Anregungen bekommen. Besonders hilfreich waren die Hinweise von Prof. Stefan Brass, Dr. Sven Helmer, Prof. Volker Linnemann, Prof. Guido Moerkotte, Prof. Reinhard Pichler und Frau Katrin Seyr. Dr. Michael Ley danken wir weiterhin für seinen phantastischen Bibliographie-Server <http://dblp.uni-trier.de/>.

München, im März 2011

Alfons Kemper