



mitp

Saake

Sattler

Heuer

6. Auflage

Datenbanken

Konzepte und Sprachen





Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. 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 fürervielfältigungen, Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

*Gunter Saake
Kai-Uwe Sattler
Andreas Heuer*

Datenbanken

Konzepte und Sprachen

Sechste 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 978-3-95845-777-5

6. Auflage 2018

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2018 mitp Verlags GmbH & Co. KG, Frechen

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.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Sabine Janatschek, Ernst-Heinrich Profener

Sprachkorrektur: Jürgen Dubau, Astrid Langen

Covergestaltung: Christian Kalkert, www.kalkert.de

Bildnachweis Cover: [iStock.com/Shawn Hempel](https://www.iStock.com/ShawnHempel) | [fotolia.com/karpenko_ilia](https://www.fotolia.com/karpenko_ilia)

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

Vorwort zur sechsten Auflage

Datenbanken sind nach Softwaretechnik das wichtigste Teilgebiet der Fachrichtung Informatik. In einer aktuellen Umfrage der Gesellschaft für Informatik (GI) bezeichnen sich 30,32 Prozent der Informatiker als Datenbänker¹, der stärkste Wert nach dem Gebiet Softwaretechnik mit 33,39 Prozent. Selbst dem Fachgebiet Wirtschaftsinformatik ordnen sich weniger GI-Mitglieder zu (24,54 Prozent). Und weitere Gebiete wie Technische Informatik (11,44 Prozent), Künstliche Intelligenz (8,68 Prozent) und Computergraphik (4,38 Prozent) sind weit abgehängt.

Datenbanken, genauer das Gebiet der Verwaltung großer Mengen strukturierter Daten, ist auch eines der *alten*, klassischen Gebiete der Informatik. Datenbanksysteme gibt es auf heutzutage veralteten Datenbankmodellen wie dem hierarchischen und Netzwerk-Modell seit den sechziger Jahren, das derzeit immer noch verbreitetste relationale Datenbankmodell ist bereits in den siebziger Jahren in der Forschung und dann in den achtziger Jahren in Form von kommerziellen Datenbank-Management-Systemen eingeführt worden. Die auch aus den siebziger Jahren stammende Datenbanksprache SQL ist immer noch *intergalactic dataspeak*, also die Standardsprache für die Verwaltung strukturierter Daten. Analysiert man große Online-Stellenportale wie das von Monster, so ist SQL nach Java und fast gleichauf mit C++ die dritthäufigste Sprache der Informatik — weit vor Python und anderen modernen Programmiersprachen und weit vor anderen Daten- und Dokumentbeschreibungssprachen wie XML und JSON.

Datenbanken hatte man aufgrund der langen Vorgeschichte im neuen Jahrtausend sowohl als Forschungsgebiet als auch als modernes Lehrgebiet in der Informatik den schleichenden Tod vorausgesagt: Relationale Datenbanksysteme waren nun einmal einfach da, veränderten sich kaum noch, gehörten

¹Mehrfachnennungen waren erlaubt.

zur Folklore für Informatiker. Das Kerngebiet der Informatik schien aus der Mode zu kommen.

Aber plötzlich erlebte das Forschungs- und Lehrgebiet Datenbanken einen riesigen Boom als Kernstück moderner Informationssystem-Infrastrukturen: Sowohl die Hardware entwickelte sich in verschiedene Richtungen weiter, als auch Anwendungen wie das Internet, Big Data, Industrie 4.0, das Internet der Dinge und Digitalisierung im allgemeinen Sinne sind abhängig von einer riesigen Menge von Daten, die effizient verwaltet und wiedergefunden werden muss. Und im Sinne von Big Data Analytics und Maschinellem Lernen müssen Daten heutzutage auch nicht nur wiedergefunden werden, es müssen auch komplexe Muster in riesigen Datenmengen gefunden werden und sie müssen statistisch analysiert werden.

All das führte dazu, dass sich in den 10er Jahren dieses Jahrhunderts das Gebiet Datenbanken dynamischer entwickelt als je zuvor. Und obwohl in diesem Lehrbuch für eine erste Grundvorlesung Datenbanken im Bachelor-Studium der Informatik, Wirtschaftsinformatik oder Technischen Informatik im Wesentlichen die klassischen Grundlagen wie das relationale Datenbankmodell und SQL vorgestellt werden müssen, müssen sich die Autoren auch auf die vielfältigen Neuentwicklungen einstellen. Die führen dazu, dass das Buch in dieser sechsten Auflage von 2018 wieder umstrukturiert wurde. Wir stellen im Folgenden kurz die Veränderungen von der ersten zur fünften Auflage vor und beschreiben dann, welche Neuerungen diese sechste Auflage bestimmen.

Von der ersten zur fünften Auflage



Abbildung 1: Die zweite und dritte Auflage des Biberbuches

Die erste Auflage des Biber-Buches erschien im September 1995, die zweite im Januar 2000, die dritte dann in 2008 (Abbildung 1 zeigt die Cover der Auflagen 2 und 3). In der Zeit bis zur dritten Auflage, immerhin über zehn Jah-

re, waren wichtige Themen wie SQL:2003, XML, objektrelationale Datenbanken, Data Warehouses und die Verwaltung multimedialer Daten dazugekommen. Weiterhin hatten sich auch im Aufbau, in laufenden Beispielen und in der Didaktik in den Vorlesungen zum Buch diverse Änderungen ergeben, die zur völligen Neuentwicklung schon bestehender Buchkapitel in der dritten Auflage führte.

Die Überarbeitung in der dritten Auflage wurde auch dazu genutzt, um den Stoff des Buches neu zu organisieren. Diese 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. Seit der dritten Auflage ist das Buch in drei Teile aufgeteilt:

Der erste Teil 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 die Themen des ersten 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 übernommen 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“ [SSH11] der Autoren behandelt und sind deshalb hier ausgeblendet.



Abbildung 2: Die vierte und fünfte Auflage des Biberbuches

In der fünften Auflage [SSH13] (siehe Abbildung 2) wurden erstmals Teile wieder aus dem Buch entfernt: Der Stoff zu Datenbankkonzepten für Data-Warehouse-Anwendungen wurde derart umfangreich, dass wir ihn aus dem

Buch herausnahmen und stattdessen auf ein eigenes Spezialbuch zu dieser Thematik verweisen mussten [KSS12]. Des Weiteren wurden aufgrund von Anregungen von Lesern die Abschnitte zur relationalen Entwurfstheorie und zu SQL an einigen Stellen überarbeitet. Neu hinzugekommen waren Abschnitte zu Datenbankzugriffen in der Cloud, zu RDF-Daten und zu temporalen Daten.

Die hier vorliegende sechste Auflage

Die sechste Auflage bietet nun sowohl einige Umstrukturierungen, neue Kapitel zu aktuellen Themen als auch (leider) wieder Streichungen bestimmter Kapitel. Einige kleinere Fehler, die in den bisherigen Auflagen übersehen oder unglücklicherweise neu eingebaut wurden, haben wir in dieser sechsten Auflage auch korrigiert — und wahrscheinlich unvermeidlicherweise auch neue Fehler eingebaut. Im Wesentlichen haben wir aber den Stand der Technik auf das Jahr 2018 gebracht und beispielsweise den Stand der SQL-Standardisierung auf die derzeit aktuelle Version SQL:2016. Im Folgenden geben wir einen Überblick über die Änderungen. Eine Änderung auf dem Cover möchten wir noch erwähnen: Der Biber als Logo für die gesamte Buchreihe findet sich auch weiterhin (stilisiert und klein) auf dem Cover, hat aber dort seine seitenfüllende Präsenz verloren. Als Ausgleich haben wir in Abbildung 3 noch einmal den Biber in voller Schönheit präsentiert (gezeichnet von Arved Sattler).



Abbildung 3: Der Biber als Logo für die gesamte Buchreihe (Zeichnung von Arved Sattler)

Umstrukturierungen. Während die oben für die dritte Auflage beschriebene Aufteilung des Buches in Teile eins bis drei stabil bleibt, wurden bestehende Kapitel aufgetrennt und verschoben bzw. Teile des Stoffes neu zugeordnet.

Kapitel 1 haben wir stark gekürzt, da die Einführung in das klassische relationale Datenbankmodell nun in einem eigenen Kapitel direkt anschließend erfolgt.

Kapitel 2 ist neu und führt nun das relationale Datenbankmodell zunächst informal, aber bereits sehr frühzeitig im Buch ein.

In **Kapitel 5** wird das relationale Datenbankmodell und die Relationenalgebra als erstes Anfragemodell formal eingeführt. Die alternativen Anfragekalküle werden aber erst im zweiten Teil des Buches in **Kapitel 10** eingeführt.

Ebenso werden im **Kapitel 7** über den Relationalen Datenbankentwurf nur noch die Kriterien für einen guten Datenbankentwurf und das einfachste Entwurfsverfahren, die Dekomposition, vorgestellt. Erweiterte Kriterien und Verfahren werden nun auch erst in einem eigenen **Kapitel 9** im zweiten Teil des Buches aufgegriffen.

Der **dritte Teil** des Buches ab **Kapitel 17** ist völlig neu strukturiert und sortiert worden. Hier beschreiben wir die diversen Erweiterungen des Relationenmodells und in den letzten Jahrzehnten entstandenen alternativen Datenbankmodelle, die in vielen Fällen auch als Erweiterungen in den SQL-Standard mit eingeflossen sind.

Neue Kapitel zu neuen Themen. In den letzten Jahren sind diverse Datenbanktechniken neu entwickelt worden, in eigenen Linien von Datenbanksystemen auch bereits umgesetzt und breit verwendet worden, teilweise aber auch bereits als Erweiterungen in den aktuellen SQL-Standard aufgenommen worden. Diese neuen Techniken werden in eigenen Abschnitten oder Kapiteln eingeführt, etwa die Erkennung von Tupelmustern in relationalen Datenbanken als neue Analyseform in SQL:2016 (**Abschnitt 11.6**) und erweiterte Programmierschnittstellen als Verbindung von Java mit relationalen Datenbanken (**Abschnitt 14.4** zu JPA: Java Persistence API). Stark erweitert wurden im Bereich des Datenschutzes Anonymitätsmaße und die Prinzipien der datensparsamen Anfrageverarbeitung (**Abschnitt 16.3**) sowie im Bereich des Text Retrieval das Ranking im Vektorraummodell (**Abschnitt 17.2**). In zwei eigenen Kapiteln am Schluss des dritten Teiles gehen wir auf die Entwicklungen zu NoSQL- und NewSQL-Datenbanksystemen (**Kapitel 21**) sowie Graph-Datenbanken (**Kapitel 22**) näher ein.

Streichungen. Aufgrund des Umfangs des Buches und der produktionstechnisch vorgegebenen Seitengrenze mussten wir diverse Teile der fünften Auflage aus dem Buch streichen. Im digitalen Zeitalter nutzen wir aber die Möglichkeit, diese eliminierten Teile als *virtuelle Kapitel* über das Internet als pdf-Dateien zur Verfügung stellen zu können. Alle virtuellen Kapitel führen wir in **Anhang B** kurz ein, dort befindet sich auch der Link auf die elektronischen

schen Dokumente. Die eliminierten Abschnitte oder Kapitel betreffen die historischen Modelle (hierarchisches Modell, Netzwerkmodell) aus der vorrelationalen Zeit, die Erweiterungen des Entity-Relationship-Modells als Entwurfsmodell, objektorientierte Datenbankmodelle inklusive des ODMG-Standards, einige Erweiterungen des Relationenmodells und ihrer Anfragesprachen um strukturierte Tabellen nicht in erster Normalform, eine Umsetzung der Relationenalgebra in eine Lehrsprache (Tutorial D) sowie ECA-Regeln und aktive Datenbanken.

Die Biber-Reihe

Das Gebiet Datenbanken wird in mehreren Büchern behandelt, an denen Autoren dieses Buches auch beteiligt sind. Obwohl nicht auf allen Covern der Bücher ein Bezug zum Biber hergestellt wird, nennen wir diese Bücher Biber 1 bis Biber 4, wobei Biber 1 das hier vorliegende Buch „Datenbanken: Konzepte und Sprachen“ darstellt.

Biber 2. Implementierungsaspekte von Datenbanksystemen wie Dateiorganisationsformen und Zugriffstrukturen (etwa Indexdateien), Anfrageoptimierung und Transaktionskonzepte (Concurrency Control) werden im Buch „Datenbanken: Implementierungstechniken“ [SSH11] behandelt (siehe Abbildung 4). Aufgrund der dramatischen Entwicklung im Bereich der Prozessoren (Single Instruction Multiple Data, Mehrkern) und der Speichertechnologie (Cache-Levels, Non-Volatile RAM, Storage Class Memory) müssen wir die hierfür vollständig neu entwickelten Implementierungstechniken in den nächsten Jahren in einem getrennten Band anbieten.



Abbildung 4: Der zweite Band der Biberbuch-Reihe: Biber 2 oder Datenbanken: Implementierungstechniken

Biber 3. Bereits in Auflage drei dieses Buches wurden die Datenbankkonzepte für Data-Warehouse-Anwendungen derart umfangreich, dass wir sie aus dem Buch herausnahmen und stattdessen in ein eigenes Spezialbuch zu dieser Thematik aufgenommen haben [KSS12].

Biber 4. Bei einem anderen Verlag ist zusätzlich ein Buch erschienen, das die Reihe durch Techniken für extrem große, auf diversen Rechnern im Netz verteilten Datenbeständen ergänzt sowie die parallele Verarbeitung von Datenbankoperationen auf Rechnerclustern thematisiert [RSS15].

Danksagungen

Zu danken haben wir bei den Neuauflagen dieses Biber-1-Buches insbesondere für Korrekturen und Hinweise zu nötigen Aktualisierungen und Erweiterungen unseren (jetzigen und ehemaligen) Mitarbeitern und Studenten Ingolf Geist, Eike Schallehn, Andreas Lübcke, Stephan Vornholt, Christine Krause, Rita Schindler, Constantin Pohl, Meike Klettke, Holger Meyer, Temenushka Ignatova, Andre Peters, Martin Garbe, Alf-Christian Schering, Dagmar Waltemath, Nils Weber, Sebastian Schick, Thomas Nösinger sowie den Lesern Lorenz Froihofer und Andreas Hilmer. Weiterhin danken wir Knut Stolze für die Unterstützung bei praktischen Tests.

Speziell bei der sechsten Auflage bedanken wir uns bei Tanja Auge, Henrik Hertel, Mark Lukas Möller, Johannes Goltz, Michael Poppe, Daniel Dietrich, Ben Hellmanzik, Enrico Gruner, Frank Röger, Hannes Grunert, Ilvio Bruder und Frank Meyer.

Mit praktischen Tests haben bei der sechsten Auflage mitgeholfen: Ilvio Bruder, Daniel Dietrich, Johannes Goltz, Hannes Grunert, Martin Jurklies und Rita Schindler.

Ein Dankeschön geht auch an die zuständige Lektorin des MITP-Verlages Sabine Janatschek, die viel Geduld aufgebracht hat, sowie an Jürgen Dubau und Astrid Langen für das sorgfältige Korrekturlesen.

Gunter Saake bedankt sich bei Birgit, Torben und Annkristin für den liebevollen und familiären Rückhalt, der sich lange hinziehende Buchprojekte erst erträglich machen kann.

Kai-Uwe Sattler bedankt sich bei Britta, Arved und Bennett, ohne deren Liebe, Rückhalt und Verständnis ein solches Buchprojekt wohl nicht möglich wäre.

Andreas Heuer möchte sich schließlich bei Renate für die über 30-jährige Unterstützung und Geduld über alle sechs Auflagen hinweg bedanken.

Ergänzende Informationen zum Buch, wie Verweise zu begleitenden Vorlesungsmaterialien, gegebenenfalls erforderliche Fehlerkorrekturen und alle virtuellen Kapitel, die aus Platzgründen nicht mehr in dieses Buch passten, sind im Web unter folgender Adresse zu finden:

<http://www.biberbuch.de>

Informationen und Downloads zur Buchreihe gibt es auch auf der Verlagsseite:

<http://www.mitp.de/776>

Magdeburg, Ilmenau und Rostock, im Februar 2018

Gunter Saake, Kai-Uwe Sattler und Andreas Heuer

Inhaltsverzeichnis

Vorwort zur sechsten Auflage	v
Inhaltsverzeichnis	xiii
1 Grundlegende Konzepte	1
1.1 Motivation und Historie	1
1.2 Komponenten und Funktionen	7
1.2.1 Prinzipien und Aufgaben	8
1.2.2 Einsatzgebiete, Grenzen und Entwicklungstendenzen	10
1.2.3 Wann kommt was?	13
1.3 Beispielanwendung	14
1.4 Vertiefende Literatur	16
1.5 Übungsaufgaben	16
2 Relationale Datenbanken – Daten in Tabellen	17
2.1 Relationen für tabellarische Daten	17
2.1.1 Begriffe im Relationenmodell	18
2.1.2 Integritätsbedingungen: Schlüssel	19
2.1.3 Integritätsbedingungen: Fremdschlüssel	20
2.2 Datendefinition in SQL	21
2.2.1 Mögliche Wertebereiche in SQL	22
2.2.2 Beispiele für die Datendeklaration	22
2.2.3 Nullwerte	23
2.3 Grundoperationen: Die Relationenalgebra	23
2.3.1 Selektion σ	24
2.3.2 Projektion π	25
2.3.3 Natürlicher Verbund \bowtie	25
2.3.4 Umbenennung β	27
2.3.5 Mengenoperationen	27
2.4 Qualität entworfener Tabellen	28
2.5 SQL als Anfragesprache	31

2.6	Änderungsoperationen in SQL	33
2.6.1	Die update -Anweisung	33
2.6.2	Die delete -Anweisung	35
2.6.3	Die insert -Anweisung	36
2.7	Sichten in SQL	37
2.8	Wie geht es weiter?	37
2.9	Übungsaufgaben	38

I Kernkonzepte relationaler Datenbanken 39

3 Architekturen von Datenbanksystemen 41

3.1	Schemaarchitektur und Datenunabhängigkeit	42
3.2	Systemarchitekturen	47
3.2.1	ANSI-SPARC-Architektur	47
3.2.2	Der Weg einer Anfrage	50
3.2.3	Fünf-Schichten-Architektur	52
3.2.4	Konkrete Systemarchitekturen	55
3.3	Anwendungsarchitekturen	59
3.4	Zusammenfassung	63
3.5	Vertiefende Literatur	63
3.6	Übungsaufgaben	64

4 Das Entity-Relationship-Modell 67

4.1	Datenbankmodelle	67
4.2	Grundlagen des Entity-Relationship-Modells	74
4.2.1	Grundkonzepte des klassischen ER-Modells	74
4.2.2	Ein einfaches Beispiel für ein ER-Schema	81
4.2.3	Semantik eines ER-Schemas	83
4.3	Eigenschaften von Beziehungen	83
4.3.1	Stelligkeit	84
4.3.2	Kardinalitäten und funktionale Beziehungen	87
4.3.3	Kardinalitäten in der klassischen Chen-Notation	90
4.3.4	Kardinalitäten in funktionaler Notation	91
4.3.5	Kardinalitäten in Intervallnotation	92
4.4	Weitere Konzepte im Entity-Relationship-Modell	97
4.4.1	Abhängige Entity-Typen	98
4.4.2	Die IST-Beziehung	99
4.4.3	Optionalität von Attributen	101
4.5	Zusammenfassung	102
4.6	Vertiefende Literatur	102
4.7	Übungsaufgaben	103

5	Relationenmodell und Relationenalgebra	105
5.1	Relationenmodell: Strukturteil	105
5.1.1	Schemata und Instanzen	106
5.1.2	Integritätsbedingungen	110
5.2	Relationenalgebra: Operationenteil	114
5.2.1	Kriterien für Anfragesprachen	115
5.2.2	Relationenalgebra	116
5.3	Änderungsoperationen	127
5.3.1	Allgemeine Grundprinzipien	128
5.3.2	Relationale Änderungsoperationen	129
5.4	Zusammenfassung	130
5.5	Vertiefende Literatur	130
5.6	Übungsaufgaben	130
6	Phasen des Datenbankentwurfs	135
6.1	Entwurfsaufgabe	135
6.2	Phasenmodell	138
6.2.1	Anforderungsanalyse	140
6.2.2	Konzeptioneller Entwurf	141
6.2.3	Verteilungsentwurf	143
6.2.4	Logischer Entwurf	144
6.2.5	Datendefinition	146
6.2.6	Physischer Entwurf	146
6.2.7	Implementierung und Wartung	147
6.2.8	Objektorientierte Entwurfsmethoden	147
6.2.9	Phasenbegleitende Methoden	148
6.3	Aspekte der Datenintegration	148
6.3.1	Heterogenität der Datenmodelle	149
6.3.2	Heterogene Datenbankschemata	149
6.3.3	Heterogenität auf der Datenebene	150
6.3.4	Schemakonflikte bei der Integration	151
6.4	Entity-Relationship-Abbildung auf das Relationenmodell	152
6.4.1	Informationskapazität	153
6.4.2	Beispiel für eine Abbildung auf das Relationenmodell	155
6.4.3	Abbildungsregeln für das relationale Modell	156
6.5	Zusammenfassung	165
6.6	Vertiefende Literatur	165
6.7	Übungsaufgaben	166
7	Relationaler Datenbankentwurf	167
7.1	Funktionale Abhängigkeiten	169
7.1.1	Definition funktionaler Abhängigkeiten	169
7.1.2	Ableitung von funktionalen Abhängigkeiten	171

7.2	Schemaeigenschaften	177
7.2.1	Änderungsanomalien	177
7.2.2	Normalformen	179
7.2.3	Minimalität	186
7.3	Transformationseigenschaften	187
7.3.1	Abhängigkeitstreue	187
7.3.2	Verbundtreue	189
7.4	Entwurfsverfahren	192
7.4.1	Ziele	192
7.4.2	Dekompositionsverfahren	193
7.4.3	Ausblick Syntheseverfahren	197
7.5	Zusammenfassung	198
7.6	Vertiefende Literatur	198
7.7	Übungsaufgaben	198
8	Die relationale Datenbanksprache SQL	201
8.1	SQL als Datendefinitionssprache	202
8.1.1	Erzeugen von Tabellen	203
8.1.2	Tabellen mit Integritätsbedingungen	207
8.1.3	Löschen und Ändern von Tabellendefinitionen	208
8.1.4	Erzeugen und Löschen von Indexen	210
8.2	SQL als relationale Anfragesprache	211
8.2.1	Überblick	212
8.2.2	Die from -Klausel	213
8.2.3	Die select -Klausel	216
8.2.4	Die where -Klausel	218
8.2.5	Mengenoperationen	222
8.2.6	Schachtelung von Anfragen	223
8.2.7	Mächtigkeit des SQL-Kerns	229
8.3	Änderungsoperationen in SQL	230
8.3.1	Übersicht über Änderungen in SQL	231
8.3.2	Die update -Anweisung	231
8.3.3	Die delete -Anweisung	233
8.3.4	Die insert -Anweisung	234
8.3.5	Die merge -Anweisung	235
8.3.6	Probleme bei SQL-Änderungen	236
8.4	Zusammenfassung	237
8.5	Vertiefende Literatur	237
8.6	Übungsaufgaben	238

II Erweiterte Konzepte für relationale Datenbanken 241

9 Erweiterter relationaler Datenbankentwurf	243
9.1 Überdeckungen von funktionalen Abhängigkeiten	244
9.1.1 Nicht-redundante Überdeckung	245
9.1.2 Reduzierte Überdeckung	247
9.1.3 Bildung von Äquivalenzklassen	249
9.1.4 Minimale Überdeckung	250
9.1.5 Ringförmige Überdeckung	251
9.2 Syntheseverfahren	252
9.2.1 Ablauf der Synthese	253
9.2.2 Erreichung der Verbundtreue	255
9.3 Verfeinerung des Entity-Relationship-Datenbankentwurfs . . .	256
9.4 Mehrwertige Abhängigkeiten	257
9.4.1 Grundlagen	257
9.4.2 Schemaeigenschaften	260
9.4.3 Transformationseigenschaften	262
9.5 Weitere Abhängigkeiten und Normalformen	262
9.5.1 Verbundabhängigkeiten	262
9.5.2 Inklusionsabhängigkeiten	263
9.5.3 Weitere relationale Entwurfsverfahren	264
9.5.4 Weitere Anwendungen der relationalen Theorie	265
9.6 Zusammenfassung	266
9.7 Vertiefende Literatur	267
9.8 Übungsaufgaben	268
10 Grundlagen von relationalen Anfragen	269
10.1 Erweiterungen der Relationenalgebra	269
10.2 Anfragekalküle	273
10.2.1 Ein allgemeiner Kalkül	273
10.2.2 Ergebnisbestimmung einer Anfrage	275
10.3 Tupelkalkül	276
10.3.1 Definition des Tupelkalküls	276
10.3.2 Beispielanfragen im Tupelkalkül	277
10.3.3 Bezug zu SQL	278
10.4 Bereichskalkül	280
10.4.1 Sichere Anfragen	282
10.4.2 Beispielanfragen im Bereichskalkül	283
10.4.3 Eigenschaften des Bereichskalküls	285
10.4.4 Relationenalgebraoperationen im Bereichskalkül	285
10.5 Zusammenfassung	286
10.6 Vertiefende Literatur	286
10.7 Übungsaufgaben	288

11	Erweiterte Konzepte von SQL	291
11.1	Weitere Operationen und Prädikate	291
11.1.1	Skalare Ausdrücke	291
11.1.2	Prädikate	297
11.1.3	Quantoren und Mengenvergleiche	297
11.1.4	Behandlung von Nullwerten	300
11.2	Aggregation, Gruppierung und Sortierung	302
11.2.1	Aggregatfunktionen	302
11.2.2	Gruppierung	305
11.2.3	Sortierung	308
11.2.4	Erweiterte Aggregatfunktionen in SQL:2003	309
11.2.5	Top-k-Anfragen	314
11.2.6	Skyline-Anfragen	317
11.3	Äußere Verbunde	318
11.4	Künstliche Schlüssel und Sequenzgeneratoren	320
11.5	Benannte Anfragen und Rekursion	322
11.5.1	Benannte Anfragen	322
11.5.2	Rekursive Anfragen	323
11.6	Erkennung von Tupelmustern	331
11.7	SQL-Versionen	335
11.7.1	SEQUEL2	335
11.7.2	SQL-89	337
11.7.3	SQL-92	337
11.7.4	SQL:1999 und SQL:2003	339
11.7.5	SQL:2006 und SQL:2008	341
11.7.6	SQL:2011 und SQL:2016	341
11.8	Zusammenfassung	344
11.9	Vertiefende Literatur	344
11.10	Übungsaufgaben	344
12	Weitere relationale Datenbanksprachen	349
12.1	QUEL	350
12.1.1	Anfragen in QUEL	350
12.1.2	Änderungsoperationen in QUEL	353
12.2	Query by Example	353
12.2.1	Anfragen in QBE	354
12.2.2	Funktionen, Sortierung und Aggregierung in QBE	358
12.2.3	Formale Semantik von QBE	359
12.2.4	Ausdrucksfähigkeit von QBE	360
12.2.5	Änderungen in QBE	361
12.2.6	Anfragen in MS Access	363
12.3	Datalog	367
12.3.1	Grundbegriffe	367

12.3.2	Semantik rekursiver Regeln	369
12.3.3	Semantik und Auswertung von Datalog	370
12.4	Zusammenfassung	371
12.5	Vertiefende Literatur	372
12.6	Übungsaufgaben	372
13	Transaktionen, Integrität & Trigger	373
13.1	Grundlagen von Transaktionen	374
13.1.1	ACID-Prinzip	374
13.1.2	Probleme im Mehrbenutzerbetrieb	376
13.1.3	Transaktionssteuerung in SQL	381
13.1.4	Transaktionen und Integritätssicherung	384
13.2	Architekturen zur Integritätssicherung	385
13.2.1	Integritätssicherung durch Anwendung	386
13.2.2	Integritätsmonitor als Komponente des DBMS	386
13.2.3	Integritätssicherung durch Einkapselung	387
13.3	Integritätsbedingungen in SQL	388
13.3.1	Inhärente Integritätsbedingungen im Relationenmodell	389
13.3.2	Weitere Bedingungen in der SQL-DDL	389
13.3.3	Die assertion -Klausel	390
13.3.4	Verwaltung und Überprüfung von Bedingungen	391
13.4	Klassifikation von Integritätsbedingungen	392
13.5	Trigger	394
13.6	Methoden der Integritätssicherung	397
13.6.1	Integritätssicherung durch Trigger	398
13.6.2	Integritätssicherung durch Anfragemodifikation	400
13.7	Zusammenfassung	401
13.8	Vertiefende Literatur	402
13.9	Übungsaufgaben	403
14	Datenbankanwendungsentwicklung	405
14.1	Grundprinzipien	406
14.2	Programmiersprachenanbindung: Call-Level-Schnittstellen	408
14.2.1	SQL/CLI: Der Standard	409
14.2.2	ODBC	412
14.2.3	JDBC	413
14.2.4	Weitere Call-Level-Schnittstellen	418
14.3	Eingebettetes SQL	419
14.3.1	Statische Einbettung: Embedded SQL	420
14.3.2	Dynamische Einbettung: Dynamic SQL	427
14.3.3	SQLJ: Embedded SQL für Java	428
14.4	High-Level-Schnittstellen	430
14.4.1	Persistenz von Objekten	431

14.4.2	Grundlagen der Abbildung	432
14.4.3	JPA und Hibernate	435
14.4.4	Weitere Technologien	443
14.5	Prozedurale SQL-Erweiterungen und Datenbanksprachen	444
14.5.1	Vorteile von gespeicherten Prozeduren	445
14.5.2	SQL/PSM: Der Standard	446
14.5.3	PL/SQL von Oracle	453
14.5.4	Gespeicherte Prozeduren in Java	455
14.6	Anwendungsentwicklung in der Cloud	458
14.6.1	Database-as-a-Service und Cloud-Datenbanken	459
14.6.2	Klassische DBMS in der Cloud	460
14.6.3	NoSQL-Systeme in der Cloud	461
14.7	Zusammenfassung	461
14.8	Vertiefende Literatur	463
14.9	Übungsaufgaben	463
15	Sichten	465
15.1	Motivation und Begriffsbildung	466
15.1.1	Sichten und externe Schemata	467
15.1.2	Definition von Sichten	467
15.1.3	Definition von Sichten in SQL	468
15.1.4	Vorteile von Sichten	469
15.2	Probleme mit Sichten	470
15.2.1	Kriterien für Änderungen auf Sichten	471
15.2.2	Projektionssichten	472
15.2.3	Selektionssichten	474
15.2.4	Verbundsichten	475
15.2.5	Aggregationssichten	477
15.2.6	Klassifikation der Problembereiche	478
15.3	Behandlung von Sichten in SQL	479
15.3.1	Auswertung von Anfragen an Sichten in SQL	480
15.3.2	Sichtänderungen in SQL-92	481
15.3.3	Sichtänderungen ab SQL:2003	482
15.4	Theorie änderbarer Sichten	484
15.5	Instead-of-Trigger für Sichtänderungen	486
15.6	Zusammenfassung	489
15.7	Vertiefende Literatur	489
15.8	Übungsaufgaben	491
16	Zugriffskontrolle & Privacy	493
16.1	Sicherheitsmodelle	495
16.1.1	Diskrete Sicherheitsmodelle	495
16.1.2	Verbindliche Sicherheitsmodelle	495

16.2	Rechtevergabe in SQL	496
16.2.1	Benutzer und Schemata	497
16.2.2	Rechtevergabe in SQL	497
16.2.3	Zurücknahme von Rechten	499
16.2.4	Rollenmodell in SQL:2003	500
16.2.5	Auditing	500
16.2.6	Authentifikation und Autorisierung	502
16.3	Privacy-Aspekte in Datenbanken	502
16.3.1	Statistische Datenbanken	503
16.3.2	Quasi-Identifikator	505
16.3.3	k-Anonymität	506
16.3.4	l-Diversität, t-Closeness, Differential Privacy	507
16.3.5	Datensparsame Anfrageverarbeitung	508
16.4	Zusammenfassung	511
16.5	Vertiefende Literatur	512
16.6	Übungsaufgaben	512

III Erweiterte Datenbankmodelle und -techniken 515

17	Multimediale Daten	517
17.1	Multimedia-Datenbanken	518
17.1.1	Grundbegriffe	518
17.1.2	Grundlagen des Multimedia Retrieval	522
17.2	Text Retrieval	529
17.2.1	Information Retrieval auf Texten	529
17.2.2	Grundtechniken des Text Retrieval	529
17.2.3	Deskribierung	531
17.2.4	Recherche	534
17.2.5	Ranking	539
17.2.6	Information-Retrieval-Systeme	542
17.3	SQL/MM	543
17.3.1	SQL/MM Full Text	544
17.3.2	SQL/MM Still Image	546
17.3.3	Der Datentyp Video	547
17.3.4	SQL/MM Spatial	548
17.4	Zusammenfassung	548
17.5	Vertiefende Literatur	548
17.6	Übungsaufgaben	550

18 Räumliche und temporale Daten	551
18.1 Verwaltung raumbezogener Daten	551
18.1.1 Grundbegriffe	552
18.1.2 Modellierung raumbezogener Daten	554
18.1.3 Prädikate und Anfragen auf raumbezogenen Daten	559
18.1.4 Oracle Spatial	566
18.1.5 Weitere Systeme	569
18.2 Temporale Daten	570
18.2.1 Grundbegriffe	571
18.2.2 Umsetzung in SQL	572
18.2.3 Temporale Schlüssel, Fremdschlüssel und Anfragen	574
18.2.4 Weitere Entwicklung und Einordnung	576
18.3 Zusammenfassung	577
18.4 Vertiefende Literatur	578
18.5 Übungsaufgaben	579
19 Objektorientierte und objektrelationale Modelle	581
19.1 Exkurs: Objektorientierte Datenbankmodelle	581
19.2 Abbildung von Objekten auf Relationen	583
19.2.1 Typkonstruktoren	584
19.2.2 Abbildung der Spezialisierungshierarchie	585
19.3 Objektrelationale Erweiterungen	587
19.3.1 Large Objects: BLOB und CLOB	588
19.3.2 Typkonstruktoren	588
19.3.3 Identitäten, Referenzen und Pfadausdrücke	593
19.3.4 Hierarchien und Vererbung	593
19.3.5 Methoden	595
19.4 Objektrelationale Konzepte in SQL:2003	595
19.4.1 Typsystem und DDL	596
19.4.2 Anfragen	603
19.4.3 Methoden in SQL:2003	606
19.5 Zusammenfassung	607
19.6 Vertiefende Literatur	607
19.7 Übungsaufgaben	608
20 XML, XQuery und SQL/XML	609
20.1 Semistrukturierte Datenmodelle	609
20.1.1 Merkmale semistrukturierter Datenmodelle	610
20.1.2 Datenmodelle für semistrukturierte Dokumente	611
20.2 XML	614
20.2.1 Bausteine von XML	614
20.2.2 Verarbeitung von XML	617
20.3 Datendefinition in XML	620

20.3.1	Dokumenttypdefinition	620
20.3.2	XML Schema	624
20.3.3	XML-Abbildung auf relationale Schemata	629
20.4	Navigation in XML-Dokumenten: XPath	630
20.4.1	Pfadausdrücke und Lokalisierungsschritte	631
20.4.2	Selektionsprädikate und Funktionen	635
20.5	Die Anfragesprache XQuery	637
20.5.1	FLWOR-Ausdrücke	638
20.5.2	Elementkonstruktoren	641
20.5.3	Verbunde und Gruppierungen	643
20.5.4	Ausdrücke und Vergleiche	647
20.5.5	Funktionen	650
20.6	SQL/XML: XML-Erweiterungen für SQL	652
20.6.1	XML-Datentypen	652
20.6.2	XML-Konstruktion mit SQL	654
20.7	Zusammenfassung	658
20.8	Vertiefende Literatur	659
20.9	Übungsaufgaben	660
21	NoSQL-Datenbanken	661
21.1	Exkurs: Big Data	662
21.2	Motivation für NoSQL	665
21.3	KV-Stores und das Wide-Column-Datenmodell	667
21.3.1	Datenmodell: Key-Value-Stores	667
21.3.2	Datenmodell: Wide Column	668
21.4	Document Stores	671
21.4.1	Das JSON-Format	671
21.4.2	Anfragen bei dokumentenorientierter Speicherung	672
21.4.3	Datenrepräsentation und Anfragen in MongoDB	673
21.5	NewSQL – relationale Datenbanken schlagen zurück	681
21.6	Zusammenfassung	685
21.7	Vertiefende Literatur	686
21.8	Übungsaufgaben	687
22	Graph-Datenbanken	689
22.1	Graph-Datenmodelle: Grundlagen	689
22.1.1	Repräsentation von Graphstrukturen	690
22.1.2	Operationen und Anfragen auf Graphen	691
22.2	Das Resource Description Framework	692
22.2.1	Das RDF-Modell	694
22.2.2	RDF-Repräsentationen	697
22.2.3	RDF Schema und Vokabulare	701
22.3	Die RDF-Anfragesprache SPARQL	705

22.3.1	Grundlagen	706
22.3.2	SPARQL-Elemente	707
22.3.3	Aggregation und Gruppierung	714
22.3.4	Weitere Anfragetypen	714
22.3.5	Updates	715
22.4	Property-Graph-Modelle	717
22.4.1	Anfragen in Cypher	720
22.4.2	Anfragen in Gremlin	726
22.5	Zusammenfassung	732
22.6	Vertiefende Literatur	733
22.7	Übungsaufgaben	734
A	Laufendes Beispiel	735
A.1	ER-Schema der Weindatenbank	735
A.2	Relationale Repräsentation	736
A.3	Vereinfachtes Schema und Beispieldaten	737
B	Zusätzliche Kapitel	739
B.1	Historische Modelle	739
B.2	Erweiterte Entwurfsmodelle	740
B.3	Erweiterte Modelle und Anfragealgebren	740
B.4	Objektorientierte und objektrelationale Modelle inklusive SQL:2003	740
B.5	Tutorial D	741
B.6	ECA-Regeln und aktive Datenbanken	741
B.7	Grundlegende Datenbanktechniken	741
B.8	SQL/JSON	742
	Literaturverzeichnis	743
	Sachindex	763

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.

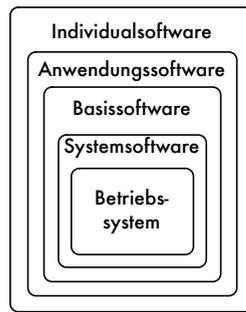


Abbildung 1.1: Aufteilung in Softwareschichten

- 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, Tabellenkalkulationen, 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 3.1 intensiver diskutiert. Auch ist die Sicherstellung der *Zugriffskontrolle* und der *Datensicherheit* ohne zentrale Datenhaltung nicht gewährleistet.

Datenintegration

Die obigen Probleme können mithilfe des Einsatzes von Datenbanktechnologien gelöst werden. Wir sprechen dann im Gegensatz zur Datenredundanz von einer *Datenintegration*. Das Prinzip der Datenintegration basiert auf folgenden Überlegungen:

Die gesamte Basis- und Anwendungssoftware arbeitet mit denselben Daten, die in einer zentralen Datenhaltungskomponente verwaltet werden. Der Gesamtbestand der Daten wird nun als *Datenbank* bezeichnet. Diese Architekturvorstellung wird in Abbildung 1.4 auf Seite 6 im Rahmen der historischen Entwicklung von Datenhaltungskomponenten graphisch verdeutlicht. Eine derartige Datenbank muss natürlich äußerst sorgfältig entworfen und in einer geeigneten Datendefinitionssprache beschrieben werden.

In unserem Beispielszenario bedeutet Datenintegration, dass zum Beispiel Adressen und Artikel nur einmal gespeichert werden, also nicht mehr redundant vorliegen.

Auch andere Probleme im Umgang mit großen Datenbeständen, etwa Fragestellungen der Effizienz, Parallelität, Zugriffskontrolle und Datensicherheit können mit heutigen kommerziellen Datenbankmanagementsystemen zufriedenstellend gelöst werden. Diese Systeme zeichnen sich durch folgende Eigenschaften aus:

- Datenbanksysteme können große Datenmengen effizient verwalten. Sie bieten benutzergerechte *Anfragesprachen* an, die eine komfortable Anfrageformulierung ohne Rücksichtnahme auf die interne Realisierung der Datenspeicherung ermöglichen. Eine interne *Optimierung* ermöglicht trotzdem einen effizienten Zugriff auf die Datenbestände.
- Viele Benutzer können parallel auf Datenbanken arbeiten. Das *Transaktionskonzept* verhindert hier unerwünschte Nebeneffekte beim Zugriff auf gemeinsam genutzte Daten.
- Die *Datenunabhängigkeit* wird durch ein Drei-Ebenen-Konzept gewährleistet, das eine externe Ebene der Anwendungssicht, eine konzeptuelle Ebene der logischen Gesamtsicht auf den Datenbestand und eine interne Ebene der implementierten Datenstrukturen unterscheidet.
- Zugriffskontrolle (kein unbefugter Zugriff) und Datensicherheit (kein ungewollter Datenverlust) werden vom System gewährleistet.

Historische Entwicklung

Die historische Entwicklung hin zu Datenbankmanagementsystemen kann in drei Stufen skizziert werden:

Die *erste Stufe* ist zu Beginn der 60er Jahre anzusiedeln, also zu einem Zeitpunkt, als die ersten Anwendungen der Massendatenverarbeitung auf Rechnern realisiert wurden. Die Daten wurden in elementaren Dateien abgelegt, und es erfolgte eine anwendungsspezifische Datenorganisation. Die Datenorganisation war geräteabhängig, zwangsweise redundant und führte leicht zu inkonsistenten Datenbeständen. Die Situation ist in Abbildung 1.2 verdeutlicht.

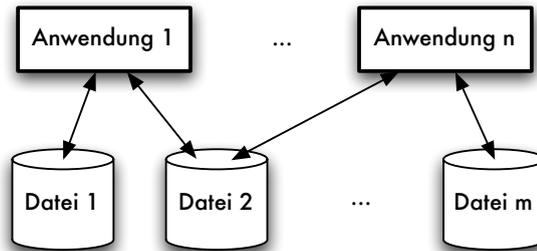


Abbildung 1.2: Historische Entwicklung 1: Zugriff auf Dateien ohne spezielle Verwaltung

Die *zweite Stufe* kennzeichnet die Situation Ende der 60er Jahre. Sie ist durch die Verwendung sogenannter *Dateiverwaltungssysteme* gekennzeichnet (bekannte Methoden sind etwa die Systeme SAM und ISAM für den sequentiellen und indexsequentiellen Dateizugriff, die auch in der Datenbankimplementierung eine große Rolle spielen [HR01, SSH11]). Dateiverwaltungssysteme konnten um zusätzliche Dienstprogramme ergänzt werden, etwa zum Sortieren von Datenbeständen. Die Situation der zweiten Stufe ist in Abbildung 1.3 dargestellt. Als wesentlicher Fortschritt wurde die *Geräteunabhängigkeit* der Datenhaltung erreicht, die Probleme der redundanten und eventuell inkonsistenten Datenhaltung blieben aber bestehen.

Diese Probleme konnten ab den 70er Jahren mit dem Einsatz von *Datenbanksystemen* gelöst werden. Sie garantieren Geräte- und Datenunabhängigkeit und ermöglichen eine redundanzfreie und konsistente Datenhaltung. Das Prinzip der Datenbanksysteme ist in Abbildung 1.4 skizziert: Der Datenbestand ist in *einer* Datenbank integriert, und jeder Zugriff erfolgt ausschließlich durch den „Filter“ des DBMS.

Ein wesentlicher Erfolgsfaktor war das von Codd vorgeschlagene *Relationenmodell* und dessen Umsetzung in relationalen Datenbanksystemen. Das erste System wurde – noch als Forschungsprototyp – von IBM unter dem Namen *System R* entwickelt und 1977 erstmals in einer größeren Installation eingesetzt. Später wurde es in das kommerzielle Produkt unter dem Namen DB2 überführt. Fast zeitgleich wurde an der University of California in Berkeley (UCB) unter Leitung von Mike Stonebraker das System *Ingres* entwickelt,

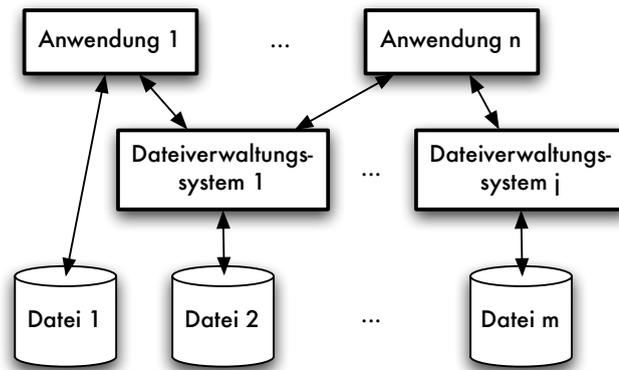


Abbildung 1.3: Historische Entwicklung 2: Dateiverwaltungssoftware

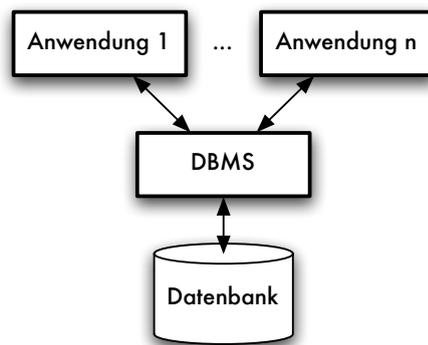


Abbildung 1.4: Historische Entwicklung 3: Datenbankmanagementsysteme

das Vorläufer für Systeme wie Postgres, Sybase und der aktuellen Version von Ingres war. 1979 wurde auch *Oracle* erstmals veröffentlicht – interessanterweise gleich als Version 2, weil die Firma (wohl berechtigterweise) davon ausging, dass Kunden der zweiten Version eines Produktes mehr Vertrauen schenken würden. Darüber hinaus hat Oracle von Beginn an die Bedeutung einer Standardisierung erkannt und dies in Form einer Kompatibilität zum IBM-Konkurrenzprodukt umgesetzt. Dagegen war das DBMS von Microsoft – der *SQL Server* – zunächst keine Eigenentwicklung: Als Microsoft Ende der 80er Jahre den Bedarf eines DBMS für die eigene Serverplattform Windows NT erkannte, wurde kurzerhand der Quellcode von Sybase gekauft und als eigenes