

Andreas
Heuer

Gunter
Saake

Kai-Uwe
Sattler

Holger
Meyer

Hannes
Grunert

Datenbanken

Kompaktkurs





Andreas
Heuer

Gunter
Saake

Kai-Uwe
Sattler

Holger
Meyer

Hannes
Grunert

Datenbanken

Kompaktkurs



Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

Ihr mitp-Verlagsteam



Neuerscheinungen, Praxistipps, Gratiskapitel,
Einblicke in den Verlagsalltag –
gibt es alles bei uns auf Instagram und Facebook



[instagram.com/mitp_verlag](https://www.instagram.com/mitp_verlag)



[facebook.com/mitp.verlag](https://www.facebook.com/mitp.verlag)

**Andreas Heuer
Gunter Saake
Kai-Uwe Sattler
Holger Meyer
Hannes Grunert**

Datenbanken

Kompaktkurs



Bibliografische Information der Deutschen Nationalbibliothek

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

ISBN 978-3-95845-784-3

1. Auflage 2020

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953/7189-079

Telefax: +49 7953/7189-082

© 2020 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: Lisa Kresse

Sprachkorrektur: Petra Heubach-Erdmann

Covergestaltung: Christian Kalkert, www.kalkert.de

Bildnachweis Cover: iStock.com/ShawnHempel, fotolia.com/karpenko_ilia

Satz: Andreas Heuer, Rostock; Gunter Saake, Magdeburg; Kai-Uwe Sattler, Ilmenau; Holger Meyer, Rostock; Hannes Grunert, Rostock

Datenkonvertierung: CPI books GmbH, Leck

Inhaltsverzeichnis

Inhaltsverzeichnis

Vorwort

1 Was sind Datenbanken?

- 1.1 Warum Datenbanken?
- 1.2 Datenbanksysteme
- 1.3 Anforderungen: Die Codd'schen Regeln
- 1.4 DBMS-Architektur
- 1.5 Datenunabhängigkeit
- 1.6 Transaktionen
- 1.7 Konkrete Datenbankmanagementsysteme
- 1.8 Einsatzgebiete und Grenzen
- 1.9 Beispielanwendungen
- 1.10 Übersicht über die Kapitel des Buches
- Übungsaufgaben

1.11

Relationale Datenbanken - Daten als Tabellen

2 Relationen für tabellarische Daten

- 2.1 Integritätsbedingungen: Schlüssel und
- 2.2 Fremdschlüssel
- 2.3 Tabellendefinition in SQL
- 2.4 Anfrageoperationen auf Tabellen
 - 2.4.1 Selektion σ
 - 2.4.2 Projektion π
 - 2.4.3 Natürlicher Verbund \bowtie

- 2.4.4 Umbenennung β
- 2.4.5 Mengenoperationen \cup , \cap , $-$
- 2.5 Anfragen in SQL
- 2.6 Änderungsoperationen in SQL
- 2.7 Zusammenfassung
- 2.8 Übungsaufgaben

3 Das Entity-Relationship- Modell

- 3.1 Eigenschaften von Datenbankmodellen
- 3.2 Einführung in das Entity-Relationship-Modell
 - 3.2.1 Grundkonzepte
 - 3.2.2 Ein einfaches Beispiel für ein ER-Schema
- 3.3 ER-Modellierungskonzepte
 - 3.3.1 Wert und Attribut
 - 3.3.2 Entity und Entity-Typ
 - 3.3.3 Beziehung und Beziehungstyp
 - 3.3.4 Identifizierung durch Schlüssel
- 3.4 Kardinalitäten
 - 3.4.1 n:m-Beziehung
 - 3.4.2 1:n-Beziehung
 - 3.4.3 1:1-Beziehung
 - 3.4.4 Optionale und zwingende Beziehungen
 - 3.4.5 Abhängige Entity-Typen
 - 3.4.6 Die IST-Beziehung
 - 3.4.7 Alternative Notationen für Kardinalitäten
- 3.5 Zusammenfassung
- 3.6 Übungsaufgaben

4 Datenbankentwurf

- 4.1 Der Datenbankentwurfsprozess
 - 4.1.1 Ziele des Datenbankentwurfs
 - 4.1.2 Das Phasenmodell

- 4.2 Formalisierung des Relationenmodells
- 4.3 Logischer Datenbankentwurf: Abbildung auf Relationen
 - 4.3.1 Übersicht über die Abbildungsregeln
 - 4.3.2 Abbildung von Entity-Typen
 - 4.3.3 Abbildung von Beziehungstypen
 - 4.3.4 Abbildung eines abhängigen Entity-Typs
 - 4.3.5 Abbildung der IST-Beziehung
 - 4.3.6 Komplexere Beispiele
- 4.4 Zusammenfassung
- 4.5 Übungsaufgaben

5 Normalisierung für eine redundanzfreie Datenbank

- 5.1 Funktionale Abhängigkeiten
- 5.2 Schema-Eigenschaften
 - 5.2.1 Update-Anomalien
 - 5.2.2 Erste Normalform
 - 5.2.3 Zweite Normalform
 - 5.2.4 Dritte Normalform
 - 5.2.5 Boyce-Codd-Normalform
 - 5.2.6 Minimalität
- 5.3 Transformationseigenschaften
 - 5.3.1 Abhängigkeitstreue
 - 5.3.2 Verbundtreue
- 5.4 Mehrwertige Abhängigkeiten
 - 5.4.1 Definition der mehrwertigen Abhängigkeit
 - 5.4.2 Vierte Normalform
- 5.5 Zusammenfassung
- 5.6 Übungsaufgaben

6 Datendefinition und Updates in SQL

- 6.1 Datendefinition in SQL

- 6.1.1 Die create table-Klausel
 - 6.1.2 Die create domain-Klausel
 - 6.1.3 Die Klauseln alter und drop
- 6.2 Änderungsoperationen in SQL
 - 6.2.1 Die insert-Klausel
 - 6.2.2 Die update-Klausel
 - 6.2.3 Die delete-Klausel
- 6.3 Zusammenfassung
- 6.4 Übungsaufgaben

7 Anfragen in SQL

- 7.1 Kriterien für Anfragesprachen
- 7.2 Der SFW-Block
- 7.3 Auswahl von Tabellen: Die from-Klausel
 - 7.3.1 Kartesisches Produkt
 - 7.3.2 Verbunde als explizite Operatoren
- 7.4 Projektionsattribute: Die select-Klausel
- 7.5 Selektionsbedingungen: Die where-Klausel
 - 7.5.1 Verbundbedingung
 - 7.5.2 Schachtelung von Anfragen
- 7.6 Mengenoperationen
- 7.7 Mächtigkeit des SQL-Kerns
- 7.8 Weitere Verbunde in SQL
 - 7.8.1 Äußere Verbunde
 - 7.8.2 Selbstverbunde
- 7.9 Weitere Selektionen in SQL
 - 7.9.1 Bereichsselektion
 - 7.9.2 Muster in Zeichenketten:
Ungewissheitsselektion
 - 7.9.3 Selektionen nach Nullwerten
 - 7.9.4 Bedingungen mit Quantoren
 - 7.9.5 exists-Prädikat

- 7.10 Skalare Ausdrücke: Funktionen zeilenweise
Aggregatfunktionen und Gruppierung: Funktionen
- 7.11 spaltenweise
Aggregatfunktionen
 - 7.11.1 Die group by- und having-Klausel
 - 7.11.2 Sortierung mit der order by-Klausel
- 7.12 SQL-Standards und ihr Funktionsumfang
Teilsprachen von SQL
- 7.13 Entwicklung der SQL-Standards
Zusammenfassung
 - 7.13.1 Übungsaufgaben
 - 7.13.2

7.14 **Sichten und Datenschutz**

Sichten

- 7.15 Definition von Sichten
Definition von Sichten in SQL

8

- Vorteile von Sichten
- 8.1 Probleme mit Sichten
 - 8.1.1 Effektkonformität bei Updates auf Sichten
 - 8.1.2 Behandlung von Sichten in SQL
 - 8.1.3 Rechtevergabe und Zugriffskontrolle
 - 8.1.4 Datensicherheit, Datenschutz, Integrität
 - 8.1.5 Rechtevergabe in Datenbanksystemen
 - 8.1.6 Rechtevergabe in SQL
- 8.2 Zurücknahme von Rechten
 - 8.2.1 Authentifikation und Autorisierung
 - 8.2.2 Datenschutz
 - 8.2.3 Privatsphäre und Datenbanken
 - 8.2.4 Statistische Auswertungen
 - 8.2.5 Anonymisierung und Pseudonymisierung
- 8.3 Quasi-Identifizierer
 - 8.3.1 Datensparsame Anfrageverarbeitung

- 8.3.2 Zusammenfassung
 - 8.3.3 Übungsaufgaben
 - 8.3.4
 - 8.3.5 **Integrität und Trigger**
- 8.4 Integritätsbedingungen
- 8.5 Integritätsbedingungen in SQL
- 9** Die check-Klausel
- 9.1 Die assertion-Klausel
 - 9.1.1 Trigger
 - 9.1.2 Grundprinzipien von Triggern
 - 9.1.3 Trigger in SQL
- 9.2 Integritätssicherung durch Trigger
 - 9.2.1 Zusammenfassung
 - 9.2.2 Übungsaufgaben
 - 9.2.3
- 9.3 **Arbeitsweise eines DBMS und Tuning**
- 9.4 Schichten-Architektur für Datenbanken
- 10** Interne Datenstrukturen und Zugriffspfade
 - Organisation von Dateien
 - 10.1 Dateiorganisationsformen
 - Indexdateien als Zugriffspfade
 - 10.2 Weitere Unterstützungen des Zugriffs
 - Indexdateien in SQL: create index
 - 10.2.1 Pufferverwaltung
 - 10.2.2 Optimierung von Anfragen
 - 10.2.3 Motivation der Optimierung
 - 10.2.4 Phasen der Anfrageverarbeitung
 - 10.2.5 Logische Optimierung
 - 10.2.6 Interne Optimierung: Effiziente
 - 10.3 Basialgorithmen
 - Kostenbasierte Auswahl
 - 10.3.1 Transaktionen und Mehrbenutzerbetrieb
 - 10.3.2

- 10.3.3 Motivation für Transaktionen
- 10.3.4 Eigenschaften von Transaktionen: Das ACID-
- 10.3.5 Prinzip
- 10.4 Isolation im Mehrbenutzerbetrieb
 - Verfahren zur Synchronisation
 - 10.4.1 Transaktionen in SQL
 - 10.4.2 Datensicherheit: Recovery und Logbuch
 - 10.4.3 Fehlerklassifikation
 - 10.4.4 Aufbau des Logbuchs
 - 10.4.5 Recovery: UNDO und REDO
- 10.5 Tuning von Datenbanksystemen
 - Tuning von Schemata
 - 10.5.1 Tuning der internen Strukturen
 - 10.5.2 Tuning von Anfragen und Transaktionen
 - 10.5.3 Tuning-Werkzeuge in SQL-Systemen
- 10.6 Zusammenfassung
 - 10.6.1 Übungsaufgaben
 - 10.6.2 **OLTP- und OLAP-Systeme**
 - 10.6.3 OLTP-Systeme
 - 10.6.4 Anfragen in OLTP-Anwendungen
- 10.7 Transaktionen und Updates in OLTP-Anwendungen
- 10.8 OLAP-Systeme
 - Datenanalysen in OLAP-Anwendungen
- 11** Vergleich OLTP und OLAP
 - 11.1 Data Warehouses und OLAP
 - Konzepte und Modelle für OLAP und Data Warehouses
 - 11.1.1 Prinzipien von Data Warehouses und OLAP-Systemen
 - 11.1.2

- 11.2 Data Warehouses und klassische Datenbanken
 - 11.2.1 SQL-Unterstützung für Data Warehouses
 - 11.2.2 Data Mining in Datenbanken
- 11.3 Data Mining im Vergleich zu OLAP
 - Mining-Probleme und -Verfahren
 - 11.3.1 Data Mining in SQL
 - 11.3.2 Zusammenfassung
 - 11.3.3 Übungsaufgaben
 - 11.3.4
- 11.4 **Row und Column Stores, NoSQL und NewSQL**
 - Einführung in Big Data
 - 11.4.1 Speicherung und Verarbeitung von Tabellen
 - 11.4.2 Zeilenorientierte Verarbeitung von Tabellen
 - 11.4.3 Spaltenorientierte Analyse von Tabellen
- 11.5 Kompression von Spalten
 - Semistrukturierte Datenbankmodelle
- 11.6 NoSQL-Datenbankmodelle
 - Schlüssel-Wert-Paare
 - Wide Column
 - Dokumenten-Modell und JSON
- 12**
 - 12.1 NewSQL: SQL und JSON
 - Zusammenfassung
 - 12.2 Übungsaufgaben
 - 12.2.1 **Ausblick**
 - 12.2.2 Objektorientiertes Datenbankmodell
 - 12.2.3 Objektrelationales Datenbankmodell
 - 12.3 Typkonstrukturen: row, array und multiset
 - Untertabellen und Untertypen
 - 12.4 Multimediale Daten
 - Multimedia in Datenbanken

	12.4.1	Der SQL/MM-Standard
	12.4.2	Texte in Datenbanken: SQL/MM Full Text
	12.4.3	Bilder in Datenbanken: SQL/MM Still Image
	12.5	Verarbeitung von Big Data
		Verteilte Datenbanken
	12.6	Parallele Datenbanken
		Big-Data-Frameworks
	12.7	
		Hotel-Beispiel
13		ER-Modellierung
		Relationale Repräsentation
	13.1	
		Universitäts-Beispiel
	13.2	ER-Modellierung
		Relationale Repräsentation
		13.2.1 Beispiel-Datenbank
		13.2.2
	13.3	Literaturverzeichnis
		Sachindex
		13.3.1
		13.3.2 Schlüsselwortindex
		13.3.3
		13.3.4
	13.4	
		13.4.1
		13.4.2
		13.4.3
A		
	A.1	
	A.2	
B		

Vorwort

Das Gebiet der *Datenbanksysteme* gehört zu den klassischen Ausbildungsgebieten der Informatik-Studiengänge. Datenbanksysteme kommen immer dann zum Einsatz, wenn an die Datenhaltung besondere Anforderungen betreffend der Zuverlässigkeit, des zu speichernden Volumens, der Ausfallsicherheit, des Mehrbenutzerzugriffs, der Komplexität der Datenbeschreibung oder der Datenqualität gestellt werden. Zu Beginn des Informationszeitalters ist es daher nicht verwunderlich, dass der Umgang mit Datenbanksystemen für viele Absolventinnen und Absolventen der Informatikstudiengänge zum Berufsalltag gehört.

Zielgruppe dieses Buches

Durch die zunehmende Verbreitung des Informationsaustausches über das Internet werden Datenbanksysteme aber nicht nur von Informatikern benutzt und weiterentwickelt. Die Bereitstellung von Informationen über das Internet und die Verwaltung dieser Informationen mit Datenbanksystemen werden auch für Schüler, Anwender und andere an dieser Technologie interessierte Personen ein ernst zu nehmendes Thema. Allerdings sind bei der Benutzung von Datenbanksystemen

Grundkenntnisse über den Datenbankentwurf und die Datenbanksprachen unerlässlich.

Dies ist ein Buch zum Studium und Selbststudium für Schüler und Studenten insbesondere außerhalb des Fachbereichs Informatik. Natürlich kann es auch von Informatikern als einfache Einführung genutzt werden. In kompakter Form und anhand von zwei durchgängigen Beispielen vermittelt es Grundkenntnisse in den folgenden Fragestellungen:

- Wie entwerfe ich eine Datenbankstruktur für meine Anwendung?
- Wie definiere ich die Datenbankstruktur im Datenbanksystem?
- Wie baue ich den Datenbestand auf? Wie ändere ich den Datenbestand?
- Wie selektiere ich aus dem Datenbestand mithilfe einer Anfragesprache gesuchte Informationen?
- Wie benutze ich die Grundkonstrukte der relationalen Datenbanksprache SQL?
- Wie kann ich die Korrektheit (Integrität) der Daten sichern und die Daten vor unerlaubten Zugriffen schützen?
- Wie kann ich die Effizienz des Systems bei steigenden Datenmengen verbessern?
- Wie kann ich die Daten nicht nur auffinden und selektieren, sondern auch mit statistischen Methoden analysieren?

- Was unterscheidet die klassischen relationalen Datenbanksysteme von den aktuellen NoSQL-Systemen?

Für Details zu jedem dieser Themen sei auf den ersten Band des Biberbuchs [[SSH18](#)] von Saake, Sattler und Heuer im gleichen Verlag verwiesen, für Details zu Kapitel [10](#) auf den zweiten Band des Biberbuchs von Heuer, Saake und Sattler [[HSS19](#)].

Was dieses Buch NICHT bietet

Dieses Buch führt in die Prinzipien der Nutzung von Datenbanksystemen ein, bietet aber keinen Ersatz für Systemhandbücher. Das Buch bietet also kein Praxiswissen über die Installation von Datenbanksystemen, die konkrete Nutzung eines Systems, die Abweichungen der Datenbanksysteme vom Sprachstandard SQL, dem Aufruf der Systeme aus Webpräsentationen heraus etc. Für die Nutzung eines Datenbanksystems x, y und z muss man sich neben den Systemhandbüchern auf Spezialbücher verlassen, die es aber weitaus häufiger auf dem Markt gibt als dieses hier vorliegende: die anwendungsgerechte Erläuterung der Prinzipien (insbesondere relationaler) Datenbanksysteme.

Einsatz in Kursen

Das Buch kann auch als Grundlage für einen 20-bis 30-stündigen Kurs dienen. Hat man mehr Zeit als eine Doppelstunde pro Kapitel, also insgesamt mehr Zeit als 30 Stunden, so können Kapitel [4](#) über den Datenbankentwurf, Kapitel [7](#) über Anfragen in SQL und das Kapitel [10](#) über die

Arbeitsweise eines Datenbankmanagementsystems auf je zwei Doppelstunden ausgedehnt werden. Alternativ kann man auch bei einem 30-Stunden-Kurs die Kapitel 4, 7 und 10 jeweils in zwei Doppelstunden behandeln und dafür die Kapitel 11 bis 13 weglassen.

An der Universität Rostock haben zwei der Autoren (Holger Meyer und Hannes Grunert) bereits seit vielen Jahren regelmäßig eine Vorlesung *Datenbanken für Anwender* gehalten, die von Naturwissenschaftlern, Ingenieurwissenschaftlern, Wirtschaftswissenschaftlern und Geisteswissenschaftlern gehört wird, stetig wachsende Hörerzahlen aufweist und die mittlerweile auch in einigen Studiengängen wie Biologie und Umweltingenieurwesen zum Standard-Kanon gehört. Die Vorlesung läuft in 14 Wochen mit 2 Semesterwochenstunden (SWS) Vorlesung (sie ist also auf 28 Stunden ausgelegt). Dazu gibt es insgesamt 14 Stunden Übungen und ein 14-stündiges Datenbankpraktikum. Im Datenbankpraktikum entwickeln kleine Arbeitsgruppen mit bis zu fünf Studierenden jeweils eine Datenbankanwendung, inklusive Datenbankentwurf, Datendefinition in SQL, Füllen der Datenbank mit Daten und das Stellen von Anfragen bzw. das Durchführen von Auswertungen auf der Datenbank. Viele Studierenden-Teams bringen dazu auch Problemstellungen aus ihrem eigenen Studienfach mit: So wurden schon biologische Artendatenbanken, Gendatenbanken, Datenbanken für Messreihen in Laborversuchen oder von Messfahrten in der Ostsee entworfen, in einem Datenbanksystem umgesetzt und für Auswertungen genutzt.

Insbesondere werden im Praktikum Hinweise gegeben, wie freie Datenbanksysteme wie MySQL oder PostgreSQL oder

(für Zwecke der Lehre) freie Versionen kommerzieller Datenbanksysteme wie IBM Db2 und Oracle bezogen, installiert und benutzt werden können. Drei dieser Systeme (außer Oracle) können die Studierenden dann auch mit den vorgefertigten Beispiel-Datenbanken dieses Buches zum Nachvollziehen der Vorlesungs- und Übungs-Beispiele über ein Web-Portal nutzen.

Zum Cover

Dieses Buch ist eine auf Nicht-Informatiker zugeschnittene, extrem kompakte Version der für Bachelor-Studenten von Informatik-Studiengängen geeigneten Datenbanklehrbücher *Datenbanken - Konzepte und Sprachen* [SSH18] und *Datenbanken - Implementierungstechniken* [HSS19] von Gunter Saake, Kai-Uwe Sattler und Andreas Heuer. Das Symbolbild auf dieser Buchreihe ist schon seit 25 Jahren der Biber. Die Bücher werden deshalb auch kurz als Biber-1- und Biber-2-Buch bezeichnet. Da dieses Buch *Datenbanken - Kompaktkurs* auch vom Umfang und Detaillierungsgrad her als Auftakt dieser Reihe angesehen werden kann, bezeichnen die Autoren es auch als das Biber-0-Buch. Auf dem Cover ist *Biber 0* dadurch gekennzeichnet, dass zwei Biber in unterschiedliche Richtungen schauen – und sich somit wie bei +1 und -1 gegenseitig aufheben. Nun ist das Buch natürlich keine Null, sondern in der Informatikerzählweise das Bit 0 und damit der Anfang von allem (Wissen über Datenbanken).

Der Vorgänger

Dieses Buch *Datenbanken – Kompaktkurs* liegt nun in der ersten Auflage vor. Es gab aber bereits einen – mittlerweile lange vergriffenen – Vorgänger, der in den Jahren 2001 bis 2003 in zwei Auflagen erschienen ist: *Datenbanken kompakt* (Abbildung 1), ebenfalls vom selben Verlag. Da dieses Buch aber komplett aktualisiert, vom Inhalt her um mehrere veraltete Kapitel erleichtert, gleichzeitig aber um mehrere aktuelle Kapitel ergänzt wurde, haben sich die Autoren entschlossen, dieses Buch nicht als 3. Auflage des alten Buches anzusehen. Auch die Zielgruppe war beim Biber-Kompakt-Buch noch anders: Es war vor allem auf Bachelor-Studiengänge der Informatik an Fachhochschulen ausgerichtet.



Abbildung 1: Der knapp 20 Jahre alte Vorgänger dieses Buches

Schreibweisen und Umgebungen

In diesem Buch werden wir folgende Schreibweisen verwenden:

- Wortsymbole aus Programmier-oder Datenbanksprachen werden wie bei `select` oder `where` geschrieben.

- Namen von Elementen eines Datenbankschemas, wie Attribut-oder Relationennamen, werden wie bei *Person* oder *Nachname* geschrieben.
- Platzhalter für konkrete Namen wie Attribut-oder Relationennamen, die oft in Syntaxdefinitionen auftauchen, werden wie in *Attributname* oder *Bedingungsname* notiert.
- Einträge in einer Datenbank oder Attributwerte in Programmen oder Datenbankabfragen werden wie in *4711* oder *Meyer* notiert.
- Begriffe, die an der betreffenden Stelle im Buch gerade definiert werden, werden wie in *Relationenschema* oder *natürlicher Verbund* hervorgehoben. Diese Kursivschrift wird teilweise auch für andere Hervorhebungen wie *Betonungen* oder englische (nicht übersetzte) *Fachbegriffe* verwendet.

Wir werden viele Konzepte anhand von Beispielen erklären. Die Beispiele beziehen sich zum größten Teil auf zwei Datenbanken, die im Anhang noch einmal unter „Laufende Beispiele“ (Anhänge **A** und **B**) aufgeführt sind. Diese Beispiele beschreiben kleine Anwendungen in einer Universität und für den Betrieb eines Hotels. Wenn wir im laufenden Text die allgemeine Erläuterung von Konzepten deutlicher von veranschaulichenden Beispielen abheben wollen, so verwenden wir in einigen Kapiteln des Buches eine eigene Beispielumgebung.

◀ **Beispiel 0-1** ▶ Diese Beispielumgebung ist dann pro Kapitel durchnummeriert, sodass wir auf Beispiele

verweisen können. Das Beispiel endet schließlich mit einem kleinen Kästchen am rechten Spaltenrand. □

Durch dieses Kästchen kann man das Ende des Beispiels, in diesem Fall Beispiel [0-1](#), und damit die Fortsetzung des erklärenden Textes leichter finden.

Die Strukturierung des Buches erfolgt in nummerierte Kapitel (etwa 2), Abschnitte (etwa 3.4) und Unterabschnitte (etwa 4.1.2). Paragraphen und Absätze haben eine Zwischenüberschrift, tragen aber keine Nummer.

Danksagungen

Zu danken haben wir einer großen Anzahl von Personen, die direkt oder indirekt zum Entstehen dieses Buches beigetragen haben. Insbesondere aber geht unser Dank an die jeweiligen Mitarbeiter der Lehrstühle für Datenbank- und Informationssysteme an den Universitäten Ilmenau, Magdeburg und Rostock sowie an die Studierenden der Vorlesung *Datenbanken für Anwender* an der Universität Rostock. Weiterhin danken wir Lisa Kresse vom mitp-Verlag für das Lektorat und insbesondere ihre Geduld: Das Erscheinen des Buches war eigentlich bereits vor Beginn der Corona-Krise geplant und wurde durch Letztere dann über Monate verzögert. Bedanken möchten wir uns auch bei Petra Heubach-Erdmann (und ihrem Familien-Team inklusive mitkorrigierender Katze) für das sehr hilfreiche Sprachkorrektorat.

Weitere Informationen

Ergänzende Informationen zum Buch, wie Verweise zu begleitenden Vorlesungsmaterialien, gegebenenfalls

erforderliche Fehlerkorrekturen und alle virtuellen Teile, die aus Platzgründen nicht mehr in dieses Buch passten (Praktikumsaufgaben, Abbildungs-und Tabellenverzeichnis, Glossar), sind im Web unter folgender Adresse zu finden:

<http://www.biberbuch.de>

Weiterhin stehen dort auch für alle am Ende der einzelnen Kapitel aufgeführten Übungsaufgaben Musterlösungen zum Download bereit. Sollte die eigene Lösung von der bereitgestellten Musterlösung abweichen, sind die Autoren gern bereit, hierzu Fragen zu beantworten. Kontaktadressen der Autoren finden sich auch unter der obigen Web-Adresse.

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

<http://www.mitp.de/782>

Ilmenau, Magdeburg und Rostock, im August 2020

Andreas Heuer, Gunter Saake, Kai-Uwe Sattler, Holger Meyer und Hannes Grunert

Was sind Datenbanken?

Der Leser dieses Buches hat sicher bereits eine Vorstellung davon, was man unter Datenbanken zu verstehen hat. Allerdings wird dieser Begriff im Alltagsdeutsch sehr großzügig benutzt, sodass wir doch mit einer etwas genaueren Charakterisierung dieses Begriffes beginnen wollen.

1.1 Warum Datenbanken?

Daten ist die Pluralform von *Datum*, lateinisch für das Geschriebene, das Gegebene, Wert, Tatsache, Information. Eine Datenbank ist nun eine Bank für Daten, also im übertragenen Sinne ein sicherer Aufbewahrungsort für (wertvolle) Informationen. Diese übertragene Bedeutung einer Art Geldinstitut für Daten gibt uns bereits einige Charakteristika von Datenbanken:

- Eine Datenbank hat die (langfristige) *Aufbewahrung* von Daten als Aufgabe.
- Die *Sicherheit vor Verlusten* ist eine Hauptmotivation, etwas „auf die Bank zu bringen“.

- Eine Bank bietet *Dienstleistungen für mehrere Kunden* an, um effizient arbeiten zu können.

Eine Bank kostet Gebühren, auch eine Datenbank verursacht Kosten, also bringt man natürlich nur Daten auf die Bank, für die sich das tatsächlich lohnt. In einer Datenbank bewahren wir daher operative, sozusagen für einen Betrieb „lebenswichtige“ Daten, sehr große Datenbestände, langfristig vorzuhaltende und von vielen gleichzeitig zu nutzende Daten auf.

Betrachten wir als typisches Beispiel die Daten einer Hotelkette. In einer Datenbank aus obigen Gründen zu speichernde Informationen sind hier Reservierungen von Kunden und Preise sowie Verfügbarkeiten von Zimmern, sowie alle Angaben für die Rechnungslegung. Hingegen würden die Geburtstage und Hochzeitstage der Geschäftsführung nicht unbedingt in einer Datenbank aufzubewahren sein.

Für ein Hotel ist es wichtig, die Rechnungsdaten längere Zeit korrekt zu speichern (Regressansprüche von Gästen, Jahresumsatz), sowie verschiedene Arten von Software auf diese Daten zugreifen zu lassen (Rechnungserstellung, Umsatzberechnung, Vorratshaltung der Hotelküche, Datenaustausch zu Buchungsportalen, Erstellung von Gästeprofilen etc.).

Eine Datenbank hat nun die Aufgabe, die Daten in strukturierter Form zur Verwendung durch mehr als ein Software-System zu speichern. Die obigen Beispiele zeigen unterschiedlichen Zugriffsbedarf: In einem Fall werden die Daten eines Gastes analysiert, im Fall der Vorratshaltung die Nachbestellung von Lebensmitteln ausgelöst. Es ist daher sinnvoll, die Daten in einem Format zu speichern, das für

viele Zugriffsarten offen ist. Bei Datenbanken ist insbesondere die Speicherung in *Tabellenform in SQL-Datenbanken* verbreitet.

Die Abbildung von Datenbeständen in eine Tabellenform wird uns in diesem Buch intensiv beschäftigen. Die oben genannten Daten liegen durchaus teilweise bereits in Tabellenform vor. Auf einer Rechnung werden aber auch verschiedene Informationen gemischt, die man in einer Datenbank trennen sollte. Auch macht es Sinn, bestimmte Daten (etwa die Adresse des Gastes oder die Informationen zum Zimmertyp) nur einmal abzuspeichern, um Tippfehler bei der erneuten Eingabe von vornherein zu verhindern. Für unser Beispiel einer Hotelrechnung bedeutet dies, dass wir mehrere Tabellen zur Speicherung nutzen sollten:

- Eine Gästetabelle speichert relevante Informationen des Gastes (Namen, Adresse etc.).
- Analog gibt es Tabellen mit Informationen zu Zimmertypen, Zimmern und – im Falle einer Hotelkette – zu den einzelnen Hotels.
- Weitere Tabellen können Angaben zu reservierten sowie belegten (und damit in Rechnung zu stellenden) Zimmern verwalten.
- Eine weitere Tabelle enthält die Daten einzelner Rechnungen.

Abbildung [1.1](#) zeigt die Daten des Rechnungsausschnitts in einer derartigen Tabellenform.

Die Einträge in den verschiedenen Tabellen werden über die identifizierenden Nummern miteinander verknüpft. Die Speicherung in Tabellen erscheint bei den ersten Einträgen