

**Database Management
Systems
Third Edition**

**数据库管理系统
原理与设计
(第3版)**

Raghu Ramakrishnan
Johannes Gehrke

清华大学出版社
北京

内 容 简 介

本书由于内容全面，实践性强，已经成为数据库课程的首选教材。全书分为数据库基础、应用程序开发、存储与索引、查询评估、事务管理、数据库设计与调整、高级主题等七大部分，对数据库的设计与使用、数据库管理系统基本原理与实现技术，以及数据库研究的新进展做了详细论述。本书以一个网上书店的完整设计为例，介绍了数据库系统的具体实现过程。第3版保留了前两版的传统优点，并且加入了许多关于数据库最新研究和发展的内容，每章后面精选的复习题和练习使本书更吸引人，可读性更强。

本书可作为高等院校计算机专业本科生、研究生和教师的教材和教学参考，对于从事数据库系统研究以及学习数据库系统技术的人也很有参考价值。

编
委
会

■ 名誉主任：陈火旺

■ 主 任：王志英

■ 副 主 任：钱德沛 周立柱

■ 编委委员：（按姓氏笔画为序）

马殿富 李晓明 李仲麟 吴朝晖

何炎祥 陈道蓄 周兴社 钱乐秋

蒋宗礼 焦金生 廖明宏

■ 责任编辑：马瑛珺

本书责任编辑：周立柱

序

未来的社会是信息化的社会,计算机科学与技术在其中占据了最重要的地位,这对高素质创新型计算机人才的培养提出了迫切的要求。计算机科学与技术已经成为一门基础技术学科,理论性和技术性都很强。与传统的数学、物理和化学等基础学科相比,该学科的教育工作者既要培养学科理论研究和基本系统的开发人才,还要培养应用系统开发人才,甚至是应用人才。从层次上来讲,则需要培养系统的设计、实现、使用与维护等各个层次的人才。这就要求我们的计算机教育按照定位的需要,从知识、能力、素质三个方面进行人才培养。

硕士研究生的教育需突出“研究”,要加强理论基础的教育和科研能力的训练,使学生能够站在一定的高度去分析研究问题、解决问题。硕士研究生要通过课程的学习,进一步提高理论水平,为今后的研究和发展打下坚实的基础;通过相应的研究及学位论文撰写工作来接受全面的科研训练,了解科学研究的艰辛和科研工作者的奉献精神,培养良好的科研作风,锻炼攻关能力,养成协作精神。

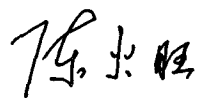
高素质创新型计算机人才应具有较强的实践能力,教学与科研相结合是培养实践能力的有效途径。高水平人才的培养是通过被培养者的高水平学术成果来反映的,而高水平的学术成果主要来源于大量高水平的科研。高水平的科研还为教学活动提供了最先进的高新技术平台和创造性的工作环境,使学生得以接触最先进的计算机理论、技术和环境。高水平的科研也为高水平人才的素质教育提供了良好的物质基础。

为提高高等院校的教学质量,教育部最近实施了精品课程建设工程。由于教材是提高教学质量的关键,必须加快教材建设的步伐。为适应学科的快速发展和培养方案的需要,要采取多种措施鼓励从事前沿研究的学者参与教材的编写和更新,在教材中反映学科前沿的研究成果与发展趋势,以高水平的科研促进教材建设。同时应适当引进国外先进的原版教材,确保所有教学环节充分反映计算机学科与产业的前沿研究水平,并与未来的发展趋势相协调。

中国计算机学会教育专业委员会在清华大学出版社的大力支持下,进行了计算机科学与技术学科硕士研究生培养的系统研究。在此基础上组织来自多所全国重点大学的计算机专家和教授们编写和出版了本系列教材。作者们以自己多年来丰富的教学和科研经验为基础,认真研究和结合我国计算机科学与技术学科硕士研究生教育的特点,力图使本系列教材对我国计算机科学与技术学科硕士研究生的教学方法和教学内容的改革起到引导作用。本系列教材的系统性和理论性强,学术水平高,反映科技新发展,具有合

适的深度和广度。同时本系列教材两种语种（中文、英文）并存，三种版权（本版、外版、合作出版）形式并存，这在系列教材的出版上走出了一条新路。

相信本系列教材的出版，能够对提高我国计算机硕士研究生教材的整体水平，进而对我国大学的计算机科学与技术硕士研究生教育以及培养高素质创新型计算机人才产生积极的促进作用。



2003年9月

陈火旺现任国防科学技术大学教授、中国工程院院士。

PREFACE

The advantage of doing one's praising for oneself is that one can lay it on so thick and exactly in the right places.

—Samuel Butler

Database management systems are now an indispensable tool for managing information, and a course on the principles and practice of database systems is now an integral part of computer science curricula. This book covers the fundamentals of modern database management systems, in particular relational database systems.

We have attempted to present the material in a clear, simple style. A quantitative approach is used throughout with many detailed examples. An extensive set of exercises (for which solutions are available online to instructors) accompanies each chapter and reinforces students' ability to apply the concepts to real problems.

The book can be used with the accompanying software and programming assignments in two distinct kinds of introductory courses:

1. **Applications Emphasis:** A course that covers the principles of database systems, and emphasizes how they are used in developing data-intensive applications. Two new chapters on application development (one on database-backed applications, and one on Java and Internet application architectures) have been added to the third edition, and the entire book has been extensively revised and reorganized to support such a course. A running case-study and extensive online materials (e.g., code for SQL queries and Java applications, online databases and solutions) make it easy to teach a hands-on application-centric course.
2. **Systems Emphasis:** A course that has a strong systems emphasis and assumes that students have good programming skills in C and C++. In this case the accompanying Minibase software can be used as the basis for projects in which students are asked to implement various parts of a relational DBMS. Several central modules in the project software (e.g., heap files, buffer manager, B+ trees, hash indexes, various join methods)

are described in sufficient detail in the text to enable students to implement them, given the (C++) class interfaces.

Many instructors will no doubt teach a course that falls between these two extremes. The restructuring in the third edition offers a very modular organization that facilitates such hybrid courses. The also book contains enough material to support advanced courses in a two-course sequence.

Organization of the Third Edition

The book is organized into six main parts plus a collection of advanced topics, as shown in Figure 0.1. The Foundations chapters introduce database systems, the

(1) Foundations	Both
(2) Application Development	Applications emphasis
(3) Storage and Indexing	Systems emphasis
(4) Query Evaluation	Systems emphasis
(5) Transaction Management	Systems emphasis
(6) Database Design and Tuning	Applications emphasis
(7) Additional Topics	Both

Figure 0.1 Organization of Parts in the Third Edition

ER model and the relational model. They explain how databases are created and used, and cover the basics of database design and querying, including an in-depth treatment of SQL queries. While an instructor can omit some of this material at their discretion (e.g., relational calculus, some sections on the ER model or SQL queries), this material is relevant to every student of database systems, and we recommend that it be covered in as much detail as possible.

Each of the remaining five main parts has either an application or a systems emphasis. Each of the three Systems parts has an overview chapter, designed to provide a self-contained treatment, e.g., Chapter 8 is an overview of storage and indexing. The overview chapters can be used to provide stand-alone coverage of the topic, or as the first chapter in a more detailed treatment. Thus, in an application-oriented course, Chapter 8 might be the only material covered on file organizations and indexing, whereas in a systems-oriented course it would be supplemented by a selection from Chapters 9 through 11. The Database Design and Tuning part contains a discussion of performance tuning and designing for secure access. These application topics are best covered after giving students a good grasp of database system architecture, and are therefore placed later in the chapter sequence.

Suggested Course Outlines

The book can be used in two kinds of introductory database courses, one with an applications emphasis and one with a systems emphasis.

The *introductory applications-oriented course* could cover the Foundations chapters, then the Application Development chapters, followed by the overview systems chapters, and conclude with the Database Design and Tuning material. Chapter dependencies have been kept to a minimum, enabling instructors to easily fine tune what material to include. The Foundations material, Part I, should be covered first, and within Parts III, IV, and V, the overview chapters should be covered first. The only remaining dependencies between chapters in Parts I to VI are shown as arrows in Figure 0.2. The chapters in Part I should be covered in sequence. However, the coverage of algebra and calculus can be skipped in order to get to SQL queries sooner (although we believe this material is important and recommend that it should be covered before SQL).

The *introductory systems-oriented course* would cover the Foundations chapters and a selection of Applications and Systems chapters. An important point for systems-oriented courses is that the timing of programming projects (e.g., using Minibase) makes it desirable to cover some systems topics early. Chapter dependencies have been carefully limited to allow the Systems chapters to be covered as soon as Chapters 1 and 3 have been covered. The remaining Foundations chapters and Applications chapters can be covered subsequently.

The book also has ample material to support a multi-course sequence. Obviously, choosing an applications or systems emphasis in the introductory course results in dropping certain material from the course; the material in the book supports a comprehensive two-course sequence that covers both applications and systems aspects. The Additional Topics range over a broad set of issues, and can be used as the core material for an advanced course, supplemented with further readings.



Supplementary Material

This book comes with extensive online supplements:

- **Online Chapter:** To make space for new material such as application development, information retrieval, and XML, we've moved the coverage of QBE to an online chapter. Students can freely download the chapter from the book's web site, and solutions to exercises from this chapter are included in solutions manual.

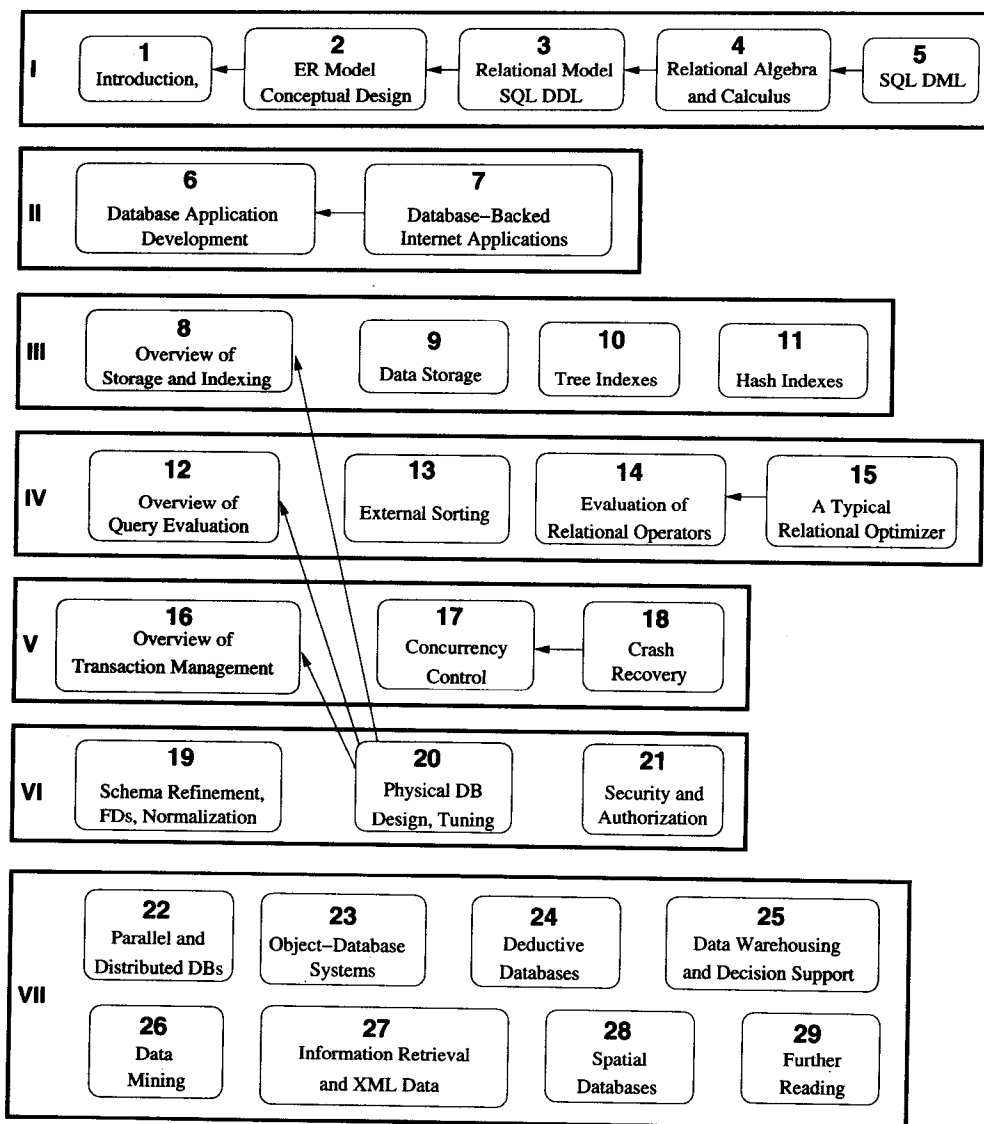


Figure 0.2 Chapter Organization and Dependencies

- **Lecture Slides:** Lecture slides are freely available for all chapters in Postscript, and PDF formats. Course instructors can also obtain these slides in Microsoft Powerpoint format, and can adapt them to their teaching needs. Instructors also have access to all figures used in the book (in xfig format), and can use them to modify the slides.

- **Solutions to Chapter Exercises:** The book has an unusually extensive set of in-depth exercises. Students can obtain solutions to odd-numbered chapter exercises and a set of lecture slides for each chapter through the Web in Postscript and Adobe PDF formats. Course instructors can obtain solutions to all exercises.
- **Software:** The book comes with two kinds of software. First, we have Minibase, a small relational DBMS intended for use in systems-oriented courses. Minibase comes with sample assignments and solutions, as described in Appendix 30. Access is restricted to course instructors. Second, we offer code for all SQL and Java application development exercises in the book, together with scripts to create sample databases, and scripts for setting up several commercial DBMSs. Students can only access solution code for odd-numbered exercises, whereas instructors have access to all solutions.
- **Instructor's Manual:** The book comes with an online manual that offers instructors comments on the material in each chapter. It provides a summary of each chapter and identifies choices for material to emphasize or omit. The manual also discusses the on-line supporting material for that chapter and offers numerous suggestions for hands-on exercises and projects. Finally, it includes samples of examination papers from courses taught by the authors using the book. It is restricted to course instructors.

For More Information

The home page for this book is at URL:

<http://www.cs.wisc.edu/~dbbook>

It contains a list of the changes between the 2nd and 3rd editions, and a frequently updated *link to all known errors in the book and its accompanying supplements*. Instructors should visit this site periodically or register at this site to be notified of important changes by email.

Acknowledgments

This book grew out of lecture notes for CS564, the introductory (senior/graduate level) database course at UW-Madison. David DeWitt developed this course and the Minirel project, in which students wrote several well-chosen parts of a relational DBMS. My thinking about this material was shaped by teaching CS564, and Minirel was the inspiration for Minibase, which is more comprehensive (e.g., it has a query optimizer and includes visualization software) but

tries to retain the spirit of Minirel. Mike Carey and I jointly designed much of Minibase. My lecture notes (and in turn this book) were influenced by Mike's lecture notes and by Yannis Ioannidis's lecture slides.

Joe Hellerstein used the beta edition of the book at Berkeley and provided invaluable feedback, assistance on slides, and hilarious quotes. Writing the chapter on object-database systems with Joe was a lot of fun.

C. Mohan provided invaluable assistance, patiently answering a number of questions about implementation techniques used in various commercial systems, in particular indexing, concurrency control, and recovery algorithms. Moshe Zloof answered numerous questions about QBE semantics and commercial systems based on QBE. Ron Fagin, Krishna Kulkarni, Len Shapiro, Jim Melton, Dennis Shasha, and Dirk Van Gucht reviewed the book and provided detailed feedback, greatly improving the content and presentation. Michael Goldweber at Beloit College, Matthew Haines at Wyoming, Michael Kifer at SUNY Stony Brook, Jeff Naughton at Wisconsin, Praveen Seshadri at Cornell, and Stan Zdonik at Brown also used the beta edition in their database courses and offered feedback and bug reports. In particular, Michael Kifer pointed out an error in the (old) algorithm for computing a minimal cover and suggested covering some SQL features in Chapter 2 to improve modularity. Gio Wiederhold's bibliography, converted to Latex format by S. Sudarshan, and Michael Ley's online bibliography on databases and logic programming were a great help while compiling the chapter bibliographies. Shaun Flisakowski and Uri Shaft helped me frequently in my never-ending battles with Latex.

I owe a special thanks to the many, many students who have contributed to the Minibase software. Emmanuel Ackaouy, Jim Pruyne, Lee Schumacher, and Michael Lee worked with me when I developed the first version of Minibase (much of which was subsequently discarded, but which influenced the next version). Emmanuel Ackaouy and Bryan So were my TAs when I taught CS564 using this version and went well beyond the limits of a TAship in their efforts to refine the project. Paul Aoki struggled with a version of Minibase and offered lots of useful comments as a TA at Berkeley. An entire class of CS764 students (our graduate database course) developed much of the current version of Minibase in a large class project that was led and coordinated by Mike Carey and me. Amit Shukla and Michael Lee were my TAs when I first taught CS564 using this version of Minibase and developed the software further.

Several students worked with me on independent projects, over a long period of time, to develop Minibase components. These include visualization packages for the buffer manager and B+ trees (Huseyin Bektas, Harry Stavropoulos, and Weiqing Huang); a query optimizer and visualizer (Stephen Harris, Michael Lee, and Donko Donjerkovic); an ER diagram tool based on the Opossum schema

editor (Eben Haber); and a GUI-based tool for normalization (Andrew Prock and Andy Therber). In addition, Bill Kimmel worked to integrate and fix a large body of code (storage manager, buffer manager, files and access methods, relational operators, and the query plan executor) produced by the CS764 class project. Ranjani Ramamurty considerably extended Bill's work on cleaning up and integrating the various modules. Luke Blanshard, Uri Shaft, and Shaun Flisakowski worked on putting together the release version of the code and developed test suites and exercises based on the Minibase software. Krishna Kunchithapadam tested the optimizer and developed part of the Minibase GUI.

Clearly, the Minibase software would not exist without the contributions of a great many talented people. With this software available freely in the public domain, I hope that more instructors will be able to teach a systems-oriented database course with a blend of implementation and experimentation to complement the lecture material.

I'd like to thank the many students who helped in developing and checking the solutions to the exercises and provided useful feedback on draft versions of the book. In alphabetical order: X. Bao, S. Biao, M. Chakrabarti, C. Chan, W. Chen, N. Cheung, D. Colwell, C. Fritz, V. Ganti, J. Gehrke, G. Glass, V. Gopalakrishnan, M. Higgins, T. Jasmin, M. Krishnaprasad, Y. Lin, C. Liu, M. Lusignan, H. Modi, S. Narayanan, D. Randolph, A. Ranganathan, J. Reminga, A. Therber, M. Thomas, Q. Wang, R. Wang, Z. Wang, and J. Yuan. Arcady Grenader, James Harrington, and Martin Reames at Wisconsin and Nina Tang at Berkeley provided especially detailed feedback.

Charlie Fischer, Avi Silberschatz, and Jeff Ullman gave me invaluable advice on working with a publisher. My editors at McGraw-Hill, Betsy Jones and Eric Munson, obtained extensive reviews and guided this book in its early stages. Emily Gray and Brad Kosirog were there whenever problems cropped up. At Wisconsin, Ginny Werner really helped me to stay on top of things.

Finally, this book was a thief of time, and in many ways it was harder on my family than on me. My sons expressed themselves forthrightly. From my (then) five-year-old, Ketan: "Dad, stop working on that silly book. You don't have any time for *me*." Two-year-old Vivek: "You working *boook*? No no no come play basketball me!" All the seasons of their discontent were visited upon my wife, and Apu nonetheless cheerfully kept the family going in its usual chaotic, happy way all the many evenings and weekends I was wrapped up in this book. (Not to mention the days when I was wrapped up in being a faculty member!) As in all things, I can trace my parents' hand in much of this; my father, with his love of learning, and my mother, with her love of us, shaped me. My brother Kartik's contributions to this book consisted chiefly of phone calls in which he kept me from working, but if I don't acknowledge him, he's liable to

be annoyed. I'd like to thank my family for being there and giving meaning to everything I do. (There! I knew I'd find a legitimate reason to thank Kartik.)

Acknowledgments for the Second Edition

Emily Gray and Betsy Jones at McGraw-Hill obtained extensive reviews and provided guidance and support as we prepared the second edition. Jonathan Goldstein helped with the bibliography for spatial databases. The following reviewers provided valuable feedback on content and organization: Liming Cai at Ohio University, Costas Tsatsoulis at University of Kansas, Kwok-Bun Yue at University of Houston, Clear Lake, William Grosky at Wayne State University, Sang H. Son at University of Virginia, James M. Slack at Minnesota State University, Mankato, Herman Balsters at University of Twente, Netherlands, Karen C. Davis at University of Cincinnati, Joachim Hammer at University of Florida, Fred Petry at Tulane University, Gregory Speegle at Baylor University, Salih Yurttas at Texas A&M University, and David Chao at San Francisco State University.

A number of people reported bugs in the first edition. In particular, we wish to thank the following: Joseph Albert at Portland State University, Han-yin Chen at University of Wisconsin, Lois Delcambre at Oregon Graduate Institute, Maggie Eich at Southern Methodist University, Raj Gopalan at Curtin University of Technology, Davood Rafiei at University of Toronto, Michael Schrefl at University of South Australia, Alex Thomasian at University of Connecticut, and Scott Vandenberg at Siena College.

A special thanks to the many people who answered a detailed survey about how commercial systems support various features: At IBM, Mike Carey, Bruce Lindsay, C. Mohan, and James Teng; at Informix, M. Muralikrishna and Michael Ubell; at Microsoft, David Campbell, Goetz Graefe, and Peter Spiro; at Oracle, Hakan Jacobsson, Jonathan D. Klein, Muralidhar Krishnaprasad, and M. Ziauddin; and at Sybase, Marc Chanliau, Lucien Dimino, Sangeeta Doraiswamy, Hanuma Kodavalla, Roger MacNicol, and Tirumanjanam Rengarajan.

After reading about himself in the acknowledgment to the first edition, Ketan (now 8) had a simple question: "How come you didn't dedicate the book to us? Why mom?" Ketan, I took care of this inexplicable oversight. Vivek (now 5) was more concerned about the extent of his fame: "Daddy, is my name in *every* copy of your book? Do they have it in *every* computer science department in the world?" Vivek, I hope so. Finally, this revision would not have made it without Apu's and Keiko's support.

Acknowledgments for the Third Edition

We thank Raghav Kaushik for his contribution to the discussion of XML, and Alex Thomasian for his contribution to the coverage of concurrency control. A special thanks to Jim Melton for giving us a pre-publication copy of his book on object-oriented extensions in the SQL:1999 standard, and catching several bugs in a draft of this edition. Marti Hearst at Berkeley generously permitted us to adapt some of her slides on Information Retrieval, and Alon Levy and Dan Suciu were kind enough to let us adapt some of their lectures on XML. Mike Carey offered input on Web services.

Emily Lupash at McGraw-Hill has been a source of constant support and encouragement. She coordinated extensive reviews from Ming Wang at Embry-Riddle Aeronautical University, Cheng Hsu at RPI, Paul Bergstein at Univ. of Massachusetts, Archana Sathaye at SJSU, Bharat Bhargava at Purdue, John Fendrich at Bradley, Ahmet Ugur at Central Michigan, Richard Osborne at Univ. of Colorado, Akira Kawaguchi at CCNY, Mark Last at Ben Gurion, Vassilis Tsotras at Univ. of California, and Ronald Eaglin at Univ. of Central Florida. It is a pleasure to acknowledge the thoughtful input we received from the reviewers, which greatly improved the design and content of this edition. Gloria Schiesl and Jade Moran dealt cheerfully and efficiently with last-minute snafus, and, with Sherry Kane, made a very tight schedule possible. Michelle Whitaker iterated many times on the cover and end-sheet design.

On a personal note for Raghu, Ketan, following the canny example of the camel that shared a tent, observed that “it is only fair” that Raghu dedicate this edition solely to him and Vivek, since “mommy already had it dedicated only to her.” Despite this blatant attempt to hog the limelight, enthusiastically supported by Vivek and viewed with the indulgent affection of a doting father, this book is also dedicated to Apu, for being there through it all.

For Johannes, this revision would not have made it without Keiko’s support and inspiration and the motivation from looking at Elisa’s peacefully sleeping face.

CONTENTS

PREFACE	xxiv
Part I FOUNDATIONS	1
1 OVERVIEW OF DATABASE SYSTEMS	3
1.1 Managing Data	4
1.2 A Historical Perspective	6
1.3 File Systems versus a DBMS	8
1.4 Advantages of a DBMS	9
1.5 Describing and Storing Data in a DBMS	10
1.5.1 The Relational Model	11
1.5.2 Levels of Abstraction in a DBMS	12
1.5.3 Data Independence	15
1.6 Queries in a DBMS	16
1.7 Transaction Management	17
1.7.1 Concurrent Execution of Transactions	17
1.7.2 Incomplete Transactions and System Crashes	18
1.7.3 Points to Note	19
1.8 Structure of a DBMS	19
1.9 People Who Work with Databases	21
1.10 Review Questions	22
2 INTRODUCTION TO DATABASE DESIGN	25
2.1 Database Design and ER Diagrams	26
2.1.1 Beyond ER Design	27
2.2 Entities, Attributes, and Entity Sets	28
2.3 Relationships and Relationship Sets	29
2.4 Additional Features of the ER Model	32
2.4.1 Key Constraints	32
2.4.2 Participation Constraints	34
2.4.3 Weak Entities	35
2.4.4 Class Hierarchies	37
2.4.5 Aggregation	39

2.5	Conceptual Design With the ER Model	40
2.5.1	Entity versus Attribute	41
2.5.2	Entity versus Relationship	42
2.5.3	Binary versus Ternary Relationships	43
2.5.4	Aggregation versus Ternary Relationships	45
2.6	Conceptual Design for Large Enterprises	46
2.7	The Unified Modeling Language	47
2.8	Case Study: The Internet Shop	49
2.8.1	Requirements Analysis	49
2.8.2	Conceptual Design	50
2.9	Review Questions	51
3	THE RELATIONAL MODEL	57
3.1	Introduction to the Relational Model	59
3.1.1	Creating and Modifying Relations Using SQL	62
3.2	Integrity Constraints over Relations	63
3.2.1	Key Constraints	64
3.2.2	Foreign Key Constraints	66
3.2.3	General Constraints	68
3.3	Enforcing Integrity Constraints	69
3.3.1	Transactions and Constraints	72
3.4	Querying Relational Data	73
3.5	Logical Database Design: ER to Relational	74
3.5.1	Entity Sets to Tables	75
3.5.2	Relationship Sets (without Constraints) to Tables	76
3.5.3	Translating Relationship Sets with Key Constraints	78
3.5.4	Translating Relationship Sets with Participation Constraints	79
3.5.5	Translating Weak Entity Sets	82
3.5.6	Translating Class Hierarchies	83
3.5.7	Translating ER Diagrams with Aggregation	84
3.5.8	ER to Relational: Additional Examples	85
3.6	Introduction to Views	86
3.6.1	Views, Data Independence, Security	87
3.6.2	Updates on Views	88
3.7	Destroying/Altering Tables and Views	91
3.8	Case Study: The Internet Store	92
3.9	Review Questions	94
4	RELATIONAL ALGEBRA AND CALCULUS	100
4.1	Preliminaries	101
4.2	Relational Algebra	102
4.2.1	Selection and Projection	103
4.2.2	Set Operations	104

4.2.3	Renaming	106
4.2.4	Joins	107
4.2.5	Division	109
4.2.6	More Examples of Algebra Queries	110
4.3	Relational Calculus	116
4.3.1	Tuple Relational Calculus	117
4.3.2	Domain Relational Calculus	122
4.4	Expressive Power of Algebra and Calculus	124
4.5	Review Questions	126
5	SQL: QUERIES, CONSTRAINTS, TRIGGERS	130
5.1	Overview	131
5.1.1	Chapter Organization	132
5.2	The Form of a Basic SQL Query	133
5.2.1	Examples of Basic SQL Queries	138
5.2.2	Expressions and Strings in the SELECT Command	139
5.3	UNION, INTERSECT, and EXCEPT	141
5.4	Nested Queries	144
5.4.1	Introduction to Nested Queries	145
5.4.2	Correlated Nested Queries	147
5.4.3	Set-Comparison Operators	148
5.4.4	More Examples of Nested Queries	149
5.5	Aggregate Operators	151
5.5.1	The GROUP BY and HAVING Clauses	154
5.5.2	More Examples of Aggregate Queries	158
5.6	Null Values	162
5.6.1	Comparisons Using Null Values	163
5.6.2	Logical Connectives AND, OR, and NOT	163
5.6.3	Impact on SQL Constructs	163
5.6.4	Outer Joins	164
5.6.5	Disallowing Null Values	165
5.7	Complex Integrity Constraints in SQL	165
5.7.1	Constraints over a Single Table	165
5.7.2	Domain Constraints and Distinct Types	166
5.7.3	Assertions: ICs over Several Tables	167
5.8	Triggers and Active Databases	168
5.8.1	Examples of Triggers in SQL	169
5.9	Designing Active Databases	171
5.9.1	Why Triggers Can Be Hard to Understand	171
5.9.2	Constraints versus Triggers	172
5.9.3	Other Uses of Triggers	172
5.10	Review Questions	173