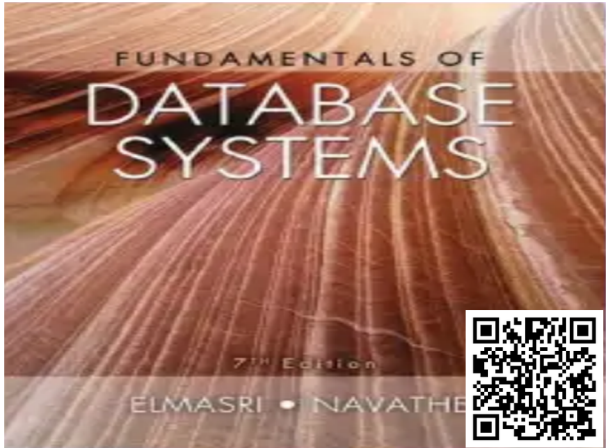


Fundamentals of Database Systems 7th Edition PDF

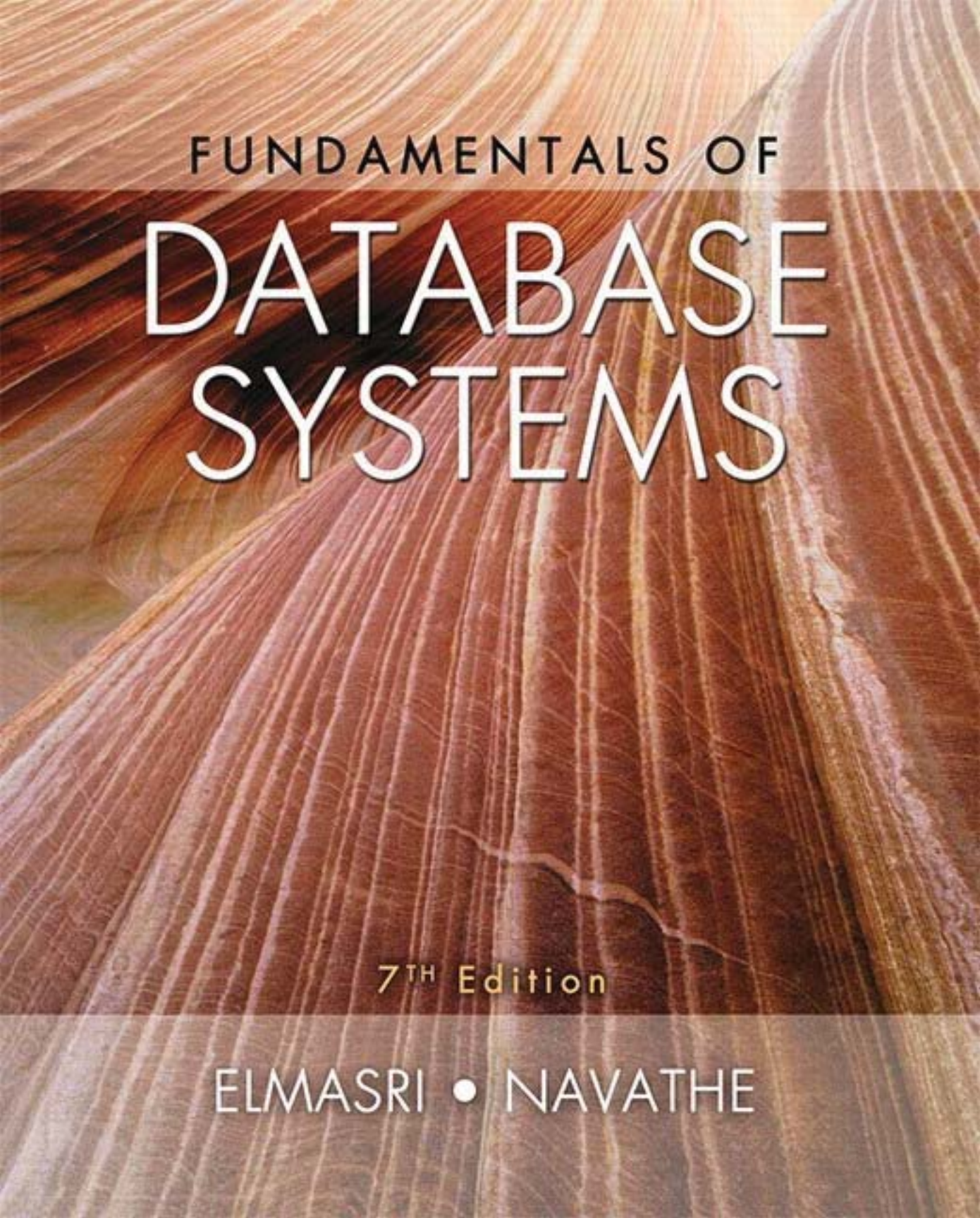
Visit the link below to download the full version of the ebook

[DOWNLOAD NOW](#)



Scan to Download
or Type the Link

ebook.ac/fundamentals7e

The background of the cover is a close-up photograph of a wood grain, showing various shades of brown and tan with prominent vertical and diagonal lines. The lighting creates a sense of depth and texture.

FUNDAMENTALS OF
DATABASE
SYSTEMS

7TH Edition

ELMASRI • NAVATHE

FUNDAMENTALS OF

Database Systems

SEVENTH EDITION

This page intentionally left blank

FUNDAMENTALS OF

Database Systems

SEVENTH EDITION

Ramez Elmasri

*Department of Computer Science and Engineering
The University of Texas at Arlington*

Shamkant B. Navathe

*College of Computing
Georgia Institute of Technology*

PEARSON

Boston Columbus Indianapolis New York San Francisco Hoboken
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Vice President and Editorial Director, ECS:
Marcia J. Horton
Acquisitions Editor: *Matt Goldstein*
Editorial Assistant: *Kelsey Loanes*
Marketing Managers: *Bram Van Kempen, Demetrius Hall*
Marketing Assistant: *Jon Bryant*
Senior Managing Editor: *Scott Disanno*
Production Project Manager: *Rose Kernan*
Program Manager: *Carole Snyder*
Global HE Director of Vendor Sourcing
and Procurement: *Diane Hynes*
Director of Operations: *Nick Sklitsis*

Operations Specialist: *Maura Zaldivar-Garcia*
Cover Designer: *Black Horse Designs*
Manager, Rights and Permissions: *Rachel Youdelman*
Associate Project Manager, Rights and Permissions:
Timothy Nicholls
Full-Service Project Management: *Rashmi Tickyani,*
iEnergizer Aptara®, Ltd.
Composition: *iEnergizer Aptara®, Ltd.*
Printer/Binder: *Edwards Brothers Malloy*
Cover Printer: *Phoenix Color/Hagerstown*
Cover Image: *Micha Pawlitzki/Terra/Corbis*
Typeface: *10.5/12 Minion Pro*

Copyright © 2016, 2011, 2007 by Ramez Elmasri and Shamkant B. Navathe. All rights reserved. Manufactured in the United States of America. This publication is protected by Copyright and permissions should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission(s) to use materials from this work, please submit a written request to Pearson Higher Education, Permissions Department, 221 River Street, Hoboken, NJ 07030.

Many of the designations by manufacturers and seller to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages with, or arising out of, the furnishing, performance, or use of these programs.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability. Whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract. Negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Library of Congress Cataloging-in-Publication Data on File

10 9 8 7 6 5 4 3 2 1

PEARSON

ISBN-10: 0-13-397077-9

ISBN-13: 978-0-13-397077-7

*To Amalia
and
to Ramy, Riyad, Katrina, and Thomas
R. E.*

*To my wife Aruna for her love, support, and understanding
and
to Rohan, Maya, and Ayush for bringing so much joy into our lives
S.B.N.*

This page intentionally left blank

Preface

This book introduces the fundamental concepts necessary for designing, using, and implementing database systems and database applications. Our presentation stresses the fundamentals of database modeling and design, the languages and models provided by the database management systems, and database system implementation techniques. The book is meant to be used as a textbook for a one- or two-semester course in database systems at the junior, senior, or graduate level, and as a reference book. Our goal is to provide an in-depth and up-to-date presentation of the most important aspects of database systems and applications, and related technologies. We assume that readers are familiar with elementary programming and data-structuring concepts and that they have had some exposure to the basics of computer organization.

New to This Edition

The following key features have been added in the seventh edition:

- A reorganization of the chapter ordering (this was based on a survey of the instructors who use the textbook); however, the book is still organized so that the individual instructor can choose to follow the new chapter ordering or *choose a different ordering of chapters* (for example, follow the chapter order from the sixth edition) when presenting the materials.
- There are two new chapters on recent advances in database systems and big data processing; one new chapter (Chapter 24) covers an introduction to the newer class of database systems known as **NOSQL databases**, and the other new chapter (Chapter 25) covers technologies for processing **big data**, including **MapReduce** and **Hadoop**.
- The chapter on query processing and optimization has been expanded and reorganized into two chapters; Chapter 18 focuses on strategies and algorithms for query processing whereas Chapter 19 focuses on query optimization techniques.
- A second UNIVERSITY database example has been added to the early chapters (Chapters 3 through 8) in addition to our COMPANY database example from the previous editions.
- Many of the individual chapters have been updated to varying degrees to include newer techniques and methods; rather than discuss these enhancements here,

we will describe them later in the preface when we discuss the organization of the seventh edition.

The following are key features of the book:

- A self-contained, flexible organization that can be tailored to individual needs; in particular, *the chapters can be used in different orders* depending on the instructor's preference.
- A companion website (<http://www.pearsonhighered.com/cs-resources>) includes data to be loaded into various types of relational databases for more realistic student laboratory exercises.
- A dependency chart (shown later in this preface) to show which chapters depend on other earlier chapters; this can guide the instructor who wants to tailor the *order of presentation of the chapters*.
- A collection of supplements, including a robust set of materials for instructors and students such as PowerPoint slides, figures from the text, and an instructor's guide with solutions.

Organization and Contents of the Seventh Edition

There are some organizational changes in the seventh edition as well as improvement to the individual chapters. The book is now divided into 12 parts as follows:

- Part 1 (Chapters 1 and 2) describes the basic introductory concepts necessary for a good understanding of database models, systems, and languages. Chapters 1 and 2 introduce databases, typical users, and DBMS concepts, terminology, and architecture, as well as a discussion of the progression of database technologies over time and a brief history of data models. These chapters have been updated to introduce some of the newer technologies such as NOSQL systems.
- Part 2 (Chapters 3 and 4) includes the presentation on entity-relationship modeling and database design; however, it is *important to note* that instructors can cover the relational model chapters (Chapters 5 through 8) *before Chapters 3 and 4* if that is their preferred order of presenting the course materials. In Chapter 3, the concepts of the Entity-Relationship (ER) model and ER diagrams are presented and used to illustrate conceptual database design. Chapter 4 shows how the basic ER model can be extended to incorporate additional modeling concepts such as subclasses, specialization, generalization, union types (categories) and inheritance, leading to the enhanced-ER (EER) data model and EER diagrams. The notation for the class diagrams of UML are also introduced in Chapters 7 and 8 as an alternative model and diagrammatic notation for ER/EER diagrams.
- Part 3 (Chapters 5 through 8) includes a detailed presentation on relational databases and SQL with some additional new material in the SQL chapters to cover a few SQL constructs that were not in the previous edition. Chapter 5

describes the basic relational model, its integrity constraints, and update operations. Chapter 6 describes some of the basic parts of the SQL standard for relational databases, including data definition, data modification operations, and simple SQL queries. Chapter 7 presents more complex SQL queries, as well as the SQL concepts of triggers, assertions, views, and schema modification. Chapter 8 describes the formal operations of the relational algebra and introduces the relational calculus. The material on SQL (Chapters 6 and 7) is presented before our presentation on relational algebra and calculus in Chapter 8 to allow instructors to start SQL projects early in a course if they wish (it is possible to cover Chapter 8 before Chapters 6 and 7 if the instructor desires this order). The final chapter in Part 2, Chapter 9, covers ER- and EER-to-relational mapping, which are algorithms that can be used for designing a relational database schema from a conceptual ER/EER schema design.

- Part 4 (Chapters 10 and 11) are the chapters on database programming techniques; these chapters can be assigned as reading materials and augmented with materials on the particular language used in the course for programming projects (much of this documentation is readily available on the Web). Chapter 10 covers traditional SQL programming topics, such as embedded SQL, dynamic SQL, ODBC, SQLJ, JDBC, and SQL/CLI. Chapter 11 introduces Web database programming, using the PHP scripting language in our examples, and includes new material that discusses Java technologies for Web database programming.
- Part 5 (Chapters 12 and 13) covers the updated material on object-relational and object-oriented databases (Chapter 12) and XML (Chapter 13); both of these chapters now include a presentation of how the SQL standard incorporates object concepts and XML concepts into more recent versions of the SQL standard. Chapter 12 first introduces the concepts for object databases, and then shows how they have been incorporated into the SQL standard in order to add object capabilities to relational database systems. It then covers the ODMG object model standard, and its object definition and query languages. Chapter 13 covers the XML (eXtensible Markup Language) model and languages, and discusses how XML is related to database systems. It presents XML concepts and languages, and compares the XML model to traditional database models. We also show how data can be converted between the XML and relational representations, and the SQL commands for extracting XML documents from relational tables.
- Part 6 (Chapters 14 and 15) are the normalization and relational design theory chapters (we moved all the formal aspects of normalization algorithms to Chapter 15). Chapter 14 defines functional dependencies, and the normal forms that are based on functional dependencies. Chapter 14 also develops a step-by-step intuitive normalization approach, and includes the definitions of multivalued dependencies and join dependencies. Chapter 15 covers normalization theory, and the formalisms, theories,

and algorithms developed for relational database design by normalization, including the relational decomposition algorithms and the relational synthesis algorithms.

- Part 7 (Chapters 16 and 17) contains the chapters on file organizations on disk (Chapter 16) and indexing of database files (Chapter 17). Chapter 16 describes primary methods of organizing files of records on disk, including ordered (sorted), unordered (heap), and hashed files; both static and dynamic hashing techniques for disk files are covered. Chapter 16 has been updated to include materials on buffer management strategies for DBMSs as well as an overview of new storage devices and standards for files and modern storage architectures. Chapter 17 describes indexing techniques for files, including B-tree and B⁺-tree data structures and grid files, and has been updated with new examples and an enhanced discussion on indexing, including how to choose appropriate indexes and index creation during physical design.
- Part 8 (Chapters 18 and 19) includes the chapters on query processing algorithms (Chapter 18) and optimization techniques (Chapter 19); these two chapters have been updated and reorganized from the single chapter that covered both topics in the previous editions and include some of the newer techniques that are used in commercial DBMSs. Chapter 18 presents algorithms for searching for records on disk files, and for joining records from two files (tables), as well as for other relational operations. Chapter 18 contains new material, including a discussion of the semi-join and anti-join operations with examples of how they are used in query processing, as well as a discussion of techniques for selectivity estimation. Chapter 19 covers techniques for query optimization using cost estimation and heuristic rules; it includes new material on nested subquery optimization, use of histograms, physical optimization, and join ordering methods and optimization of typical queries in data warehouses.
- Part 9 (Chapters 20, 21, and 22) covers transaction processing concepts; concurrency control; and database recovery from failures. These chapters have been updated to include some of the newer techniques that are used in some commercial and open source DBMSs. Chapter 20 introduces the techniques needed for transaction processing systems, and defines the concepts of recoverability and serializability of schedules; it has a new section on buffer replacement policies for DBMSs and a new discussion on the concept of snapshot isolation. Chapter 21 gives an overview of the various types of concurrency control protocols, with a focus on two-phase locking. We also discuss timestamp ordering and optimistic concurrency control techniques, as well as multiple-granularity locking. Chapter 21 includes a new presentation of concurrency control methods that are based on the snapshot isolation concept. Finally, Chapter 23 focuses on database recovery protocols, and gives an overview of the concepts and techniques that are used in recovery.

- Part 10 (Chapters 23, 24, and 25) includes the chapter on distributed databases (Chapter 23), plus the two new chapters on NOSQL storage systems for big data (Chapter 24) and big data technologies based on Hadoop and MapReduce (Chapter 25). Chapter 23 introduces distributed database concepts, including availability and scalability, replication and fragmentation of data, maintaining data consistency among replicas, and many other concepts and techniques. In Chapter 24, NOSQL systems are categorized into four general categories with an example system in each category used for our examples, and the data models, operations, as well as the replication/distribution/scalability strategies of each type of NOSQL system are discussed and compared. In Chapter 25, the MapReduce programming model for distributed processing of big data is introduced, and then we have presentations of the Hadoop system and HDFS (Hadoop Distributed File System), as well as the Pig and Hive high-level interfaces, and the YARN architecture.
- Part 11 (Chapters 26 through 29) is entitled Advanced Database Models, Systems, and Applications and includes the following materials: Chapter 26 introduces several advanced data models including active databases/triggers (Section 26.1), temporal databases (Section 26.2), spatial databases (Section 26.3), multimedia databases (Section 26.4), and deductive databases (Section 26.5). Chapter 27 discusses information retrieval (IR) and Web search, and includes topics such as IR and keyword-based search, comparing DB with IR, retrieval models, search evaluation, and ranking algorithms. Chapter 28 is an introduction to data mining including overviews of various data mining methods such as associate rule mining, clustering, classification, and sequential pattern discovery. Chapter 29 is an overview of data warehousing including topics such as data warehousing models and operations, and the process of building a data warehouse.
- Part 12 (Chapter 30) includes one chapter on database security, which includes a discussion of SQL commands for discretionary access control (GRANT, REVOKE), as well as mandatory security levels and models for including mandatory access control in relational databases, and a discussion of threats such as SQL injection attacks, as well as other techniques and methods related to data security and privacy.

Appendix A gives a number of alternative diagrammatic notations for displaying a conceptual ER or EER schema. These may be substituted for the notation we use, if the instructor prefers. Appendix B gives some important physical parameters of disks. Appendix C gives an overview of the QBE graphical query language, and Appendixes D and E (available on the book's Companion Website located at <http://www.pearsonhighered.com/elmasri>) cover legacy database systems, based on the hierarchical and network database models. They have been used for more than thirty years as a basis for many commercial database applications and transaction-processing systems.

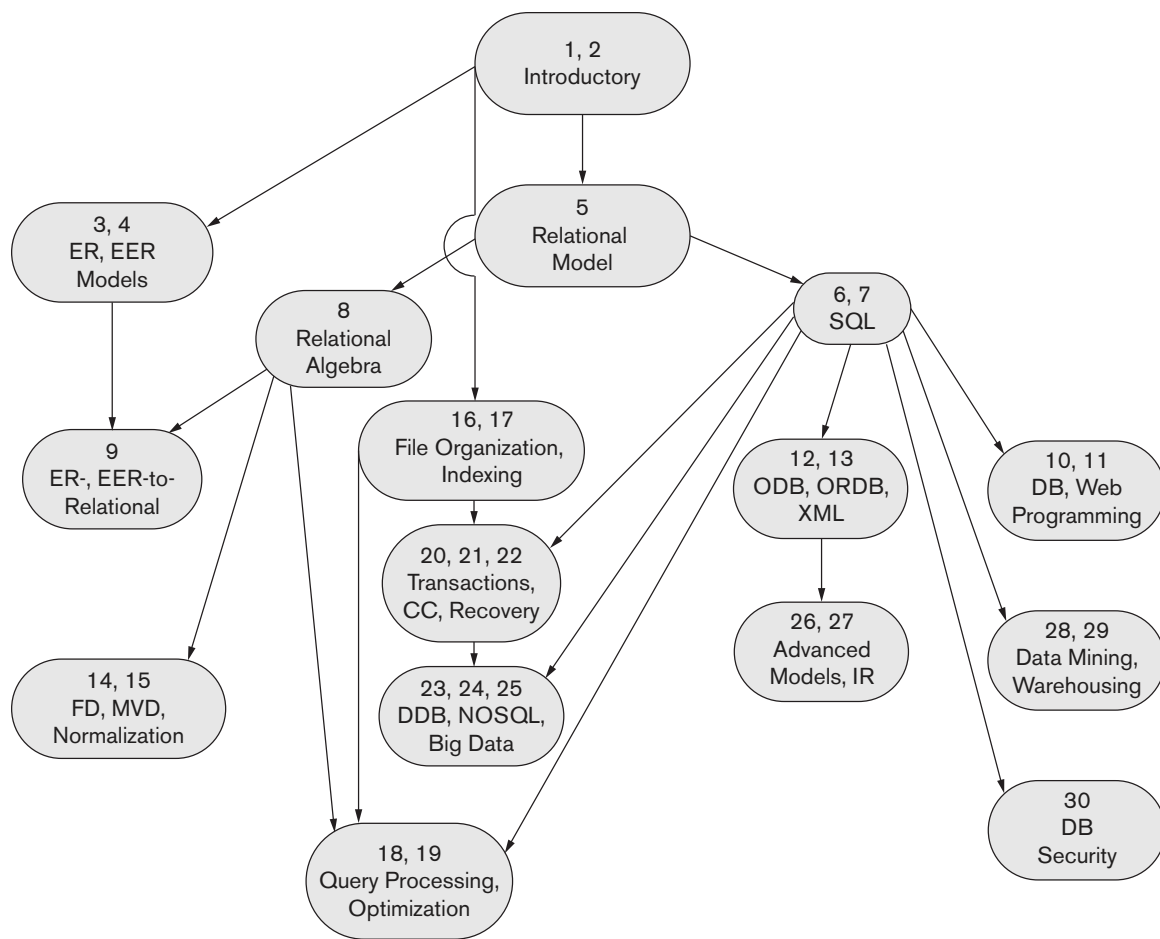
Guidelines for Using This Book

There are many different ways to teach a database course. The chapters in Parts 1 through 7 can be used in an introductory course on database systems in the order that they are given or in the preferred order of individual instructors. Selected chapters and sections may be left out and the instructor can add other chapters from the rest of the book, depending on the emphasis of the course. At the end of the opening section of some of the book's chapters, we list sections that are candidates for being left out whenever a less-detailed discussion of the topic is desired. We suggest covering up to Chapter 15 in an introductory database course and including selected parts of other chapters, depending on the background of the students and the desired coverage. For an emphasis on system implementation techniques, chapters from Parts 7, 8, and 9 should replace some of the earlier chapters.

Chapters 3 and 4, which cover conceptual modeling using the ER and EER models, are important for a good conceptual understanding of databases. However, they may be partially covered, covered later in a course, or even left out if the emphasis is on DBMS implementation. Chapters 16 and 17 on file organizations and indexing may also be covered early, later, or even left out if the emphasis is on database models and languages. For students who have completed a course on file organization, parts of these chapters can be assigned as reading material or some exercises can be assigned as a review for these concepts.

If the emphasis of a course is on database design, then the instructor should cover Chapters 3 and 4 early on, followed by the presentation of relational databases. A total life-cycle database design and implementation project would cover conceptual design (Chapters 3 and 4), relational databases (Chapters 5, 6, and 7), data model mapping (Chapter 9), normalization (Chapter 14), and application programs implementation with SQL (Chapter 10). Chapter 11 also should be covered if the emphasis is on Web database programming and applications. Additional documentation on the specific programming languages and RDBMS used would be required. The book is written so that it is possible to cover topics in various sequences. The following chapter dependency chart shows the major dependencies among chapters. As the diagram illustrates, it is possible to start with several different topics following the first two introductory chapters. Although the chart may seem complex, it is important to note that if the chapters are covered in order, the dependencies are not lost. The chart can be consulted by instructors wishing to use an alternative order of presentation.

For a one-semester course based on this book, selected chapters can be assigned as reading material. The book also can be used for a two-semester course sequence. The first course, *Introduction to Database Design and Database Systems*, at the sophomore, junior, or senior level, can cover most of Chapters 1 through 15. The second course, *Database Models and Implementation Techniques*, at the senior or first-year graduate level, can cover most of Chapters 16 through 30. The two-semester sequence can also be designed in various other ways, depending on the preferences of the instructors.



Supplemental Materials

Support material is available to qualified instructors at Pearson's instructor resource center (<http://www.pearsonhighered.com/irc>). For access, contact your local Pearson representative.

- PowerPoint lecture notes and figures.
- A solutions manual.

Acknowledgments

It is a great pleasure to acknowledge the assistance and contributions of many individuals to this effort. First, we would like to thank our editor, Matt Goldstein, for his guidance, encouragement, and support. We would like to acknowledge the excellent work of Rose Kernan for production management, Patricia Daly for a

thorough copy editing of the book, Martha McMaster for her diligence in proofing the pages, and Scott Disanno, Managing Editor of the production team. We also wish to thank Kelsey Loanes from Pearson for her continued help with the project, and reviewers Michael Doherty, Deborah Dunn, Imad Rahal, Karen Davis, Gilliean Lee, Leo Mark, Monisha Pulimood, Hassan Reza, Susan Vrbsky, Li Da Xu, Weining Zhang and Vincent Oria.

Ramez Elmasri would like to thank Kulsawasd Jitkajornwanich, Vivek Sharma, and Surya Swaminathan for their help with preparing some of the material in Chapter 24. Sham Navathe would like to acknowledge the following individuals who helped in critically reviewing and revising various topics. Dan Forsythe and Satish Damle for discussion of storage systems; Rafi Ahmed for detailed re-organization of the material on query processing and optimization; Harish Butani, Balaji Palanisamy, and Prajakta Kalmegh for their help with the Hadoop and MapReduce technology material; Vic Ghorpadey and Nenad Jukic for revision of the Data Warehousing material; and finally, Frank Rietta for newer techniques in database security, Kunal Malhotra for various discussions, and Saurav Sahay for advances in information retrieval systems.

We would like to repeat our thanks to those who have reviewed and contributed to previous editions of *Fundamentals of Database Systems*.

- **First edition.** Alan Apt (editor), Don Batory, Scott Downing, Dennis Heimbinger, Julia Hodges, Yannis Ioannidis, Jim Larson, Per-Ake Larson, Dennis McLeod, Rahul Patel, Nicholas Roussopoulos, David Stemple, Michael Stonebraker, Frank Tompa, and Kyu-Young Whang.
- **Second edition.** Dan Joraanstad (editor), Rafi Ahmed, Antonio Albano, David Beech, Jose Blakeley, Panos Chrysanthis, Suzanne Dietrich, Vic Ghorpadey, Goetz Graefe, Eric Hanson, Junguk L. Kim, Roger King, Vram Kouramajian, Vijay Kumar, John Lowther, Sanjay Manchanda, Toshimi Minoura, Inderpal Mumick, Ed Omiecinski, Girish Pathak, Raghu Ramakrishnan, Ed Robertson, Eugene Sheng, David Stotts, Marianne Winslett, and Stan Zdonick.
- **Third edition.** Maite Suarez-Rivas and Katherine Harutunian (editors); Suzanne Dietrich, Ed Omiecinski, Rafi Ahmed, Francois Bancilhon, Jose Blakeley, Rick Cattell, Ann Chervenak, David W. Embley, Henry A. Etlinger, Leonidas Fegaras, Dan Forsyth, Farshad Fotouhi, Michael Franklin, Sreejith Gopinath, Goetz Craefe, Richard Hull, Sushil Jajodia, Ramesh K. Karne, Harish Kotbagi, Vijay Kumar, Tarcisio Lima, Ramon A. Mata-Toledo, Jack McCaw, Dennis McLeod, Rokia Missaoui, Magdi Morsi, M. Narayanaswamy, Carlos Ordonez, Joan Peckham, Betty Salzberg, Ming-Chien Shan, Junping Sun, Rajshekhar Sunderraman, Aravindan Veerasamy, and Emilia E. Villareal.
- **Fourth edition.** Maite Suarez-Rivas, Katherine Harutunian, Daniel Rausch, and Juliet Silveri (editors); Phil Bernhard, Zhengxin Chen, Jan Chomicki, Hakan Ferhatosmanoglu, Len Fisk, William Hankley, Ali R. Hurson, Vijay Kumar, Peretz Shoval, Jason T. L. Wang (reviewers); Ed Omiecinski (who contributed to Chapter 27). Contributors from the University of Texas at

Arlington are Jack Fu, Hyoil Han, Babak Hojabri, Charley Li, Ande Swathi, and Steven Wu; Contributors from Georgia Tech are Weimin Feng, Dan Forsythe, Angshuman Guin, Abrar Ul-Haque, Bin Liu, Ying Liu, Wanxia Xie, and Waigen Yee.

- **Fifth edition.** Matt Goldstein and Katherine Harutunian (editors); Michelle Brown, Gillian Hall, Patty Mahtani, Maite Suarez-Rivas, Bethany Tidd, and Joyce Cosentino Wells (from Addison-Wesley); Hani Abu-Salem, Jamal R. Alsabbagh, Ramzi Bualuan, Soon Chung, Sumali Conlon, Hasan Davulcu, James Geller, Le Gruenwald, Latifur Khan, Herman Lam, Byung S. Lee, Donald Sanderson, Jamil Saquer, Costas Tsatsoulis, and Jack C. Wileden (reviewers); Raj Sunderraman (who contributed the laboratory projects); Salman Azar (who contributed some new exercises); Gaurav Bhatia, Fariborz Farahmand, Ying Liu, Ed Omiecinski, Nalini Polavarapu, Liora Sahar, Saurav Sahay, and Wanxia Xie (from Georgia Tech).
- **Sixth edition.** Matt Goldstein (editor); Gillian Hall (production management); Rebecca Greenberg (copy editing); Jeff Holcomb, Marilyn Lloyd, Margaret Waples, and Chelsea Bell (from Pearson); Rafi Ahmed, Venu Dasigi, Neha Deodhar, Fariborz Farahmand, Hariprasad Kumar, Leo Mark, Ed Omiecinski, Balaji Palanisamy, Nalini Polavarapu, Parimala R. Pranesh, Bharath Rengarajan, Liora Sahar, Saurav Sahay, Narsi Srinivasan, and Wanxia Xie.

Last, but not least, we gratefully acknowledge the support, encouragement, and patience of our families.

R. E.

S.B.N.

This page intentionally left blank

Contents

Preface	vii
About the Authors	xxx

■ part 1

Introduction to Databases ■

chapter 1 Databases and Database Users 3

1.1 Introduction	4
1.2 An Example	6
1.3 Characteristics of the Database Approach	10
1.4 Actors on the Scene	15
1.5 Workers behind the Scene	17
1.6 Advantages of Using the DBMS Approach	17
1.7 A Brief History of Database Applications	23
1.8 When Not to Use a DBMS	27
1.9 Summary	27
Review Questions	28
Exercises	28
Selected Bibliography	29

chapter 2 Database System Concepts and Architecture 31

2.1 Data Models, Schemas, and Instances	32
2.2 Three-Schema Architecture and Data Independence	36
2.3 Database Languages and Interfaces	38
2.4 The Database System Environment	42
2.5 Centralized and Client/Server Architectures for DBMSs	46
2.6 Classification of Database Management Systems	51
2.7 Summary	54
Review Questions	55
Exercises	55
Selected Bibliography	56

■ part 2

Conceptual Data Modeling and Database Design ■

chapter 3 Data Modeling Using the Entity–Relationship (ER) Model 59

- 3.1 Using High-Level Conceptual Data Models for Database Design 60
- 3.2 A Sample Database Application 62
- 3.3 Entity Types, Entity Sets, Attributes, and Keys 63
- 3.4 Relationship Types, Relationship Sets, Roles, and Structural Constraints 72
- 3.5 Weak Entity Types 79
- 3.6 Refining the ER Design for the COMPANY Database 80
- 3.7 ER Diagrams, Naming Conventions, and Design Issues 81
- 3.8 Example of Other Notation: UML Class Diagrams 85
- 3.9 Relationship Types of Degree Higher than Two 88
- 3.10 Another Example: A UNIVERSITY Database 92
- 3.11 Summary 94
- Review Questions 96
- Exercises 96
- Laboratory Exercises 103
- Selected Bibliography 104

chapter 4 The Enhanced Entity–Relationship (EER) Model 107

- 4.1 Subclasses, Superclasses, and Inheritance 108
- 4.2 Specialization and Generalization 110
- 4.3 Constraints and Characteristics of Specialization and Generalization Hierarchies 113
- 4.4 Modeling of UNION Types Using Categories 120
- 4.5 A Sample UNIVERSITY EER Schema, Design Choices, and Formal Definitions 122
- 4.6 Example of Other Notation: Representing Specialization and Generalization in UML Class Diagrams 127
- 4.7 Data Abstraction, Knowledge Representation, and Ontology Concepts 128
- 4.8 Summary 135
- Review Questions 135
- Exercises 136
- Laboratory Exercises 143
- Selected Bibliography 146

■ part **3**

The Relational Data Model and SQL ■

chapter 5	The Relational Data Model and Relational Database Constraints	149
5.1	Relational Model Concepts	150
5.2	Relational Model Constraints and Relational Database Schemas	157
5.3	Update Operations, Transactions, and Dealing with Constraint Violations	165
5.4	Summary	169
	Review Questions	170
	Exercises	170
	Selected Bibliography	175
chapter 6	Basic SQL	177
6.1	SQL Data Definition and Data Types	179
6.2	Specifying Constraints in SQL	184
6.3	Basic Retrieval Queries in SQL	187
6.4	INSERT, DELETE, and UPDATE Statements in SQL	198
6.5	Additional Features of SQL	201
6.6	Summary	202
	Review Questions	203
	Exercises	203
	Selected Bibliography	205
chapter 7	More SQL: Complex Queries, Triggers, Views, and Schema Modification	207
7.1	More Complex SQL Retrieval Queries	207
7.2	Specifying Constraints as Assertions and Actions as Triggers	225
7.3	Views (Virtual Tables) in SQL	228
7.4	Schema Change Statements in SQL	232
7.5	Summary	234
	Review Questions	236
	Exercises	236
	Selected Bibliography	238
chapter 8	The Relational Algebra and Relational Calculus	239
8.1	Unary Relational Operations: SELECT and PROJECT	241
8.2	Relational Algebra Operations from Set Theory	246

8.3 Binary Relational Operations: JOIN and DIVISION	251
8.4 Additional Relational Operations	259
8.5 Examples of Queries in Relational Algebra	265
8.6 The Tuple Relational Calculus	268
8.7 The Domain Relational Calculus	277
8.8 Summary	279
Review Questions	280
Exercises	281
Laboratory Exercises	286
Selected Bibliography	288

chapter **9** Relational Database Design by ER- and EER-to-Relational Mapping 289

9.1 Relational Database Design Using ER-to-Relational Mapping	290
9.2 Mapping EER Model Constructs to Relations	298
9.3 Summary	303
Review Questions	303
Exercises	303
Laboratory Exercises	305
Selected Bibliography	306

■ part 4

Database Programming Techniques ■

chapter **10** Introduction to SQL Programming Techniques 309

10.1 Overview of Database Programming Techniques and Issues	310
10.2 Embedded SQL, Dynamic SQL, and SQLJ	314
10.3 Database Programming with Function Calls and Class Libraries: SQL/CLI and JDBC	326
10.4 Database Stored Procedures and SQL/PSM	335
10.5 Comparing the Three Approaches	338
10.6 Summary	339
Review Questions	340
Exercises	340
Selected Bibliography	341

chapter **11** Web Database Programming Using PHP 343

11.1 A Simple PHP Example	344
11.2 Overview of Basic Features of PHP	346

11.3 Overview of PHP Database Programming	353
11.4 Brief Overview of Java Technologies for Database Web Programming	358
11.5 Summary	358
Review Questions	359
Exercises	359
Selected Bibliography	359

■ part **5**

Object, Object-Relational, and XML: Concepts, Models, Languages, and Standards ■

chapter **12** **Object and Object-Relational Databases** 363

12.1 Overview of Object Database Concepts	365
12.2 Object Database Extensions to SQL	379
12.3 The ODMG Object Model and the Object Definition Language ODL	386
12.4 Object Database Conceptual Design	405
12.5 The Object Query Language OQL	408
12.6 Overview of the C++ Language Binding in the ODMG Standard	417
12.7 Summary	418
Review Questions	420
Exercises	421
Selected Bibliography	422

chapter **13** **XML: Extensible Markup Language** 425

13.1 Structured, Semistructured, and Unstructured Data	426
13.2 XML Hierarchical (Tree) Data Model	430
13.3 XML Documents, DTD, and XML Schema	433
13.4 Storing and Extracting XML Documents from Databases	442
13.5 XML Languages	443
13.6 Extracting XML Documents from Relational Databases	447
13.7 XML/SQL: SQL Functions for Creating XML Data	453
13.8 Summary	455
Review Questions	456
Exercises	456
Selected Bibliography	456

■ part 6

Database Design Theory and Normalization ■

chapter **14** Basics of Functional Dependencies and Normalization for Relational Databases 459

- 14.1 Informal Design Guidelines for Relation Schemas 461
- 14.2 Functional Dependencies 471
- 14.3 Normal Forms Based on Primary Keys 474
- 14.4 General Definitions of Second and Third Normal Forms 483
- 14.5 Boyce-Codd Normal Form 487
- 14.6 Multivalued Dependency and Fourth Normal Form 491
- 14.7 Join Dependencies and Fifth Normal Form 494
- 14.8 Summary 495
- Review Questions 496
- Exercises 497
- Laboratory Exercises 501
- Selected Bibliography 502

chapter **15** Relational Database Design Algorithms and Further Dependencies 503

- 15.1 Further Topics in Functional Dependencies: Inference Rules, Equivalence, and Minimal Cover 505
- 15.2 Properties of Relational Decompositions 513
- 15.3 Algorithms for Relational Database Schema Design 519
- 15.4 About Nulls, Dangling Tuples, and Alternative Relational Designs 523
- 15.5 Further Discussion of Multivalued Dependencies and 4NF 527
- 15.6 Other Dependencies and Normal Forms 530
- 15.7 Summary 533
- Review Questions 534
- Exercises 535
- Laboratory Exercises 536
- Selected Bibliography 537

■ part 7

File Structures, Hashing, Indexing, and Physical Database Design ■

chapter 16 Disk Storage, Basic File Structures, Hashing, and Modern Storage Architectures 541

- 16.1 Introduction 542
- 16.2 Secondary Storage Devices 547
- 16.3 Buffering of Blocks 556
- 16.4 Placing File Records on Disk 560
- 16.5 Operations on Files 564
- 16.6 Files of Unordered Records (Heap Files) 567
- 16.7 Files of Ordered Records (Sorted Files) 568
- 16.8 Hashing Techniques 572
- 16.9 Other Primary File Organizations 582
- 16.10 Parallelizing Disk Access Using RAID Technology 584
- 16.11 Modern Storage Architectures 588
- 16.12 Summary 592
- Review Questions 593
- Exercises 595
- Selected Bibliography 598

chapter 17 Indexing Structures for Files and Physical Database Design 601

- 17.1 Types of Single-Level Ordered Indexes 602
- 17.2 Multilevel Indexes 613
- 17.3 Dynamic Multilevel Indexes Using B-Trees and B⁺-Trees 617
- 17.4 Indexes on Multiple Keys 631
- 17.5 Other Types of Indexes 633
- 17.6 Some General Issues Concerning Indexing 638
- 17.7 Physical Database Design in Relational Databases 643
- 17.8 Summary 646
- Review Questions 647
- Exercises 648
- Selected Bibliography 650

■ part 8

Query Processing and Optimization ■

chapter **18** Strategies for Query Processing 655

- 18.1 Translating SQL Queries into Relational Algebra and Other Operators 657
- 18.2 Algorithms for External Sorting 660
- 18.3 Algorithms for SELECT Operation 663
- 18.4 Implementing the JOIN Operation 668
- 18.5 Algorithms for PROJECT and Set Operations 676
- 18.6 Implementing Aggregate Operations and Different Types of JOINS 678
- 18.7 Combining Operations Using Pipelining 681
- 18.8 Parallel Algorithms for Query Processing 683
- 18.9 Summary 688
- Review Questions 688
- Exercises 689
- Selected Bibliography 689

chapter **19** Query Optimization 691

- 19.1 Query Trees and Heuristics for Query Optimization 692
- 19.2 Choice of Query Execution Plans 701
- 19.3 Use of Selectivities in Cost-Based Optimization 710
- 19.4 Cost Functions for SELECT Operation 714
- 19.5 Cost Functions for the JOIN Operation 717
- 19.6 Example to Illustrate Cost-Based Query Optimization 726
- 19.7 Additional Issues Related to Query Optimization 728
- 19.8 An Example of Query Optimization in Data Warehouses 731
- 19.9 Overview of Query Optimization in Oracle 733
- 19.10 Semantic Query Optimization 737
- 19.11 Summary 738
- Review Questions 739
- Exercises 740
- Selected Bibliography 740

■ part 9

Transaction Processing, Concurrency Control, and Recovery ■

chapter **20** Introduction to Transaction Processing Concepts and Theory 745

- 20.1 Introduction to Transaction Processing 746
- 20.2 Transaction and System Concepts 753
- 20.3 Desirable Properties of Transactions 757
- 20.4 Characterizing Schedules Based on Recoverability 759
- 20.5 Characterizing Schedules Based on Serializability 763
- 20.6 Transaction Support in SQL 773
- 20.7 Summary 776
- Review Questions 777
- Exercises 777
- Selected Bibliography 779

chapter **21** Concurrency Control Techniques 781

- 21.1 Two-Phase Locking Techniques for Concurrency
Control 782
- 21.2 Concurrency Control Based on Timestamp Ordering 792
- 21.3 Multiversion Concurrency Control Techniques 795
- 21.4 Validation (Optimistic) Techniques and Snapshot Isolation
Concurrency Control 798
- 21.5 Granularity of Data Items and Multiple Granularity
Locking 800
- 21.6 Using Locks for Concurrency Control in Indexes 805
- 21.7 Other Concurrency Control Issues 806
- 21.8 Summary 807
- Review Questions 808
- Exercises 809
- Selected Bibliography 810

chapter **22** Database Recovery Techniques 813

- 22.1 Recovery Concepts 814
- 22.2 NO-UNDO/REDO Recovery Based on Deferred
Update 821
- 22.3 Recovery Techniques Based on Immediate Update 823

22.4 Shadow Paging	826
22.5 The ARIES Recovery Algorithm	827
22.6 Recovery in Multidatabase Systems	831
22.7 Database Backup and Recovery from Catastrophic Failures	832
22.8 Summary	833
Review Questions	834
Exercises	835
Selected Bibliography	838

■ part 10

Distributed Databases, NOSQL Systems, and Big Data ■

chapter **23** Distributed Database Concepts 841

23.1 Distributed Database Concepts	842
23.2 Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design	847
23.3 Overview of Concurrency Control and Recovery in Distributed Databases	854
23.4 Overview of Transaction Management in Distributed Databases	857
23.5 Query Processing and Optimization in Distributed Databases	859
23.6 Types of Distributed Database Systems	865
23.7 Distributed Database Architectures	868
23.8 Distributed Catalog Management	875
23.9 Summary	876
Review Questions	877
Exercises	878
Selected Bibliography	880

chapter **24** NOSQL Databases and Big Data Storage Systems 883

24.1 Introduction to NOSQL Systems	884
24.2 The CAP Theorem	888
24.3 Document-Based NOSQL Systems and MongoDB	890
24.4 NOSQL Key-Value Stores	895
24.5 Column-Based or Wide Column NOSQL Systems	900
24.6 NOSQL Graph Databases and Neo4j	903
24.7 Summary	909
Review Questions	909
Selected Bibliography	910

chapter **25** Big Data Technologies Based on MapReduce and Hadoop 911

- 25.1 What Is Big Data? 914
- 25.2 Introduction to MapReduce and Hadoop 916
- 25.3 Hadoop Distributed File System (HDFS) 921
- 25.4 MapReduce: Additional Details 926
- 25.5 Hadoop v2 alias YARN 936
- 25.6 General Discussion 944
- 25.7 Summary 953
- Review Questions 954
- Selected Bibliography 956

■ part **1**

Advanced Database Models, Systems, and Applications ■

chapter **26** Enhanced Data Models: Introduction to Active, Temporal, Spatial, Multimedia, and Deductive Databases 961

- 26.1 Active Database Concepts and Triggers 963
- 26.2 Temporal Database Concepts 974
- 26.3 Spatial Database Concepts 987
- 26.4 Multimedia Database Concepts 994
- 26.5 Introduction to Deductive Databases 999
- 26.6 Summary 1012
- Review Questions 1014
- Exercises 1015
- Selected Bibliography 1018

chapter **27** Introduction to Information Retrieval and Web Search 1021

- 27.1 Information Retrieval (IR) Concepts 1022
- 27.2 Retrieval Models 1029
- 27.3 Types of Queries in IR Systems 1035
- 27.4 Text Preprocessing 1037
- 27.5 Inverted Indexing 1040
- 27.6 Evaluation Measures of Search Relevance 1044
- 27.7 Web Search and Analysis 1047

27.8 Trends in Information Retrieval	1057
27.9 Summary	1063
Review Questions	1064
Selected Bibliography	1066

chapter **28** **Data Mining Concepts** 1069

28.1 Overview of Data Mining Technology	1070
28.2 Association Rules	1073
28.3 Classification	1085
28.4 Clustering	1088
28.5 Approaches to Other Data Mining Problems	1091
28.6 Applications of Data Mining	1094
28.7 Commercial Data Mining Tools	1094
28.8 Summary	1097
Review Questions	1097
Exercises	1098
Selected Bibliography	1099

chapter **29** **Overview of Data Warehousing and OLAP** 1101

29.1 Introduction, Definitions, and Terminology	1102
29.2 Characteristics of Data Warehouses	1103
29.3 Data Modeling for Data Warehouses	1105
29.4 Building a Data Warehouse	1111
29.5 Typical Functionality of a Data Warehouse	1114
29.6 Data Warehouse versus Views	1115
29.7 Difficulties of Implementing Data Warehouses	1116
29.8 Summary	1117
Review Questions	1117
Selected Bibliography	1118

■ **part 12**

Additional Database Topics: Security ■

chapter **30** **Database Security** 1121

30.1 Introduction to Database Security Issues	1122
30.2 Discretionary Access Control Based on Granting and Revoking Privileges	1129
30.3 Mandatory Access Control and Role-Based Access Control for Multilevel Security	1134

30.4 SQL Injection	1143
30.5 Introduction to Statistical Database Security	1146
30.6 Introduction to Flow Control	1147
30.7 Encryption and Public Key Infrastructures	1149
30.8 Privacy Issues and Preservation	1153
30.9 Challenges to Maintaining Database Security	1154
30.10 Oracle Label-Based Security	1155
30.11 Summary	1158
Review Questions	1159
Exercises	1160
Selected Bibliography	1161

appendix **A** **Alternative Diagrammatic Notations for ER Models** 1163

appendix **B** **Parameters of Disks** 1167

appendix **C** **Overview of the QBE Language** 1171

C.1 Basic Retrievals in QBE	1171
C.2 Grouping, Aggregation, and Database Modification in QBE	1175

appendix **D** **Overview of the Hierarchical Data Model**
(located on the Companion Website at
<http://www.pearsonhighered.com/elmasri>)

appendix **E** **Overview of the Network Data Model**
(located on the Companion Website at
<http://www.pearsonhighered.com/elmasri>)

Selected Bibliography 1179

Index 1215

About the Authors

Ramez Elmasri is a professor and the associate chairperson of the Department of Computer Science and Engineering at the University of Texas at Arlington. He has over 140 refereed research publications, and has supervised 16 PhD students and over 100 MS students. His research has covered many areas of database management and big data, including conceptual modeling and data integration, query languages and indexing techniques, temporal and spatio-temporal databases, bio-informatics databases, data collection from sensor networks, and mining/analysis of spatial and spatio-temporal data. He has worked as a consultant to various companies, including Digital, Honeywell, Hewlett Packard, and Action Technologies, as well as consulting with law firms on patents. He was the Program Chair of the 1993 International Conference on Conceptual Modeling (ER conference) and program vice-chair of the 1994 IEEE International Conference on Data Engineering. He has served on the ER conference steering committee and has been on the program committees of many conferences. He has given several tutorials at the VLDB, ICDE, and ER conferences. He also co-authored the book “Operating Systems: A Spiral Approach” (McGraw-Hill, 2009) with Gil Carrick and David Levine. Elmasri is a recipient of the UTA College of Engineering Outstanding Teaching Award in 1999. He holds a BS degree in Engineering from Alexandria University, and MS and PhD degrees in Computer Science from Stanford University.

Shamkant B. Navathe is a professor and the founder of the database research group at the College of Computing, Georgia Institute of Technology, Atlanta. He has worked with IBM and Siemens in their research divisions and has been a consultant to various companies including Digital, Computer Corporation of America, Hewlett Packard, Equifax, and Persistent Systems. He was the General Co-chairman of the 1996 International VLDB (Very Large Data Base) conference in Bombay, India. He was also program co-chair of ACM SIGMOD 1985 International Conference and General Co-chair of the IFIP WG 2.6 Data Semantics Workshop in 1995. He has served on the VLDB foundation and has been on the steering committees of several conferences. He has been an associate editor of a number of journals including *ACM Computing Surveys*, and *IEEE Transactions on Knowledge and Data Engineering*. He also co-authored the book “Conceptual Design: An Entity Relationship Approach” (Addison Wesley, 1992) with Carlo Batini and Stefano Ceri. Navathe is a fellow of the Association for Computing Machinery (ACM) and recipient of the IEEE TCDE Computer Science, Engineering and Education Impact award in 2015. Navathe holds a PhD from the University of Michigan and has over 150 refereed publications in journals and conferences.

part **1**

**Introduction
to Databases**

This page intentionally left blank

Databases and Database Users

Databases and database systems are an essential component of life in modern society: most of us encounter several activities every day that involve some interaction with a database. For example, if we go to the bank to deposit or withdraw funds, if we make a hotel or airline reservation, if we access a computerized library catalog to search for a bibliographic item, or if we purchase something online—such as a book, toy, or computer—chances are that our activities will involve someone or some computer program accessing a database. Even purchasing items at a supermarket often automatically updates the database that holds the inventory of grocery items.

These interactions are examples of what we may call **traditional database applications**, in which most of the information that is stored and accessed is either textual or numeric. In the past few years, advances in technology have led to exciting new applications of database systems. The proliferation of social media Web sites, such as Facebook, Twitter, and Flickr, among many others, has required the creation of huge databases that store nontraditional data, such as posts, tweets, images, and video clips. New types of database systems, often referred to as **big data** storage systems, or **NOSQL systems**, have been created to manage data for social media applications. These types of systems are also used by companies such as Google, Amazon, and Yahoo, to manage the data required in their Web search engines, as well as to provide **cloud storage**, whereby users are provided with storage capabilities on the Web for managing all types of data including documents, programs, images, videos and emails. We will give an overview of these new types of database systems in Chapter 24.

We now mention some other applications of databases. The wide availability of photo and video technology on cellphones and other devices has made it possible to

store images, audio clips, and video streams digitally. These types of files are becoming an important component of **multimedia databases**. **Geographic information systems (GISs)** can store and analyze maps, weather data, and satellite images. **Data warehouses** and **online analytical processing (OLAP)** systems are used in many companies to extract and analyze useful business information from very large databases to support decision making. **Real-time** and **active database technology** is used to control industrial and manufacturing processes. And database **search techniques** are being applied to the World Wide Web to improve the search for information that is needed by users browsing the Internet.

To understand the fundamentals of database technology, however, we must start from the basics of traditional database applications. In Section 1.1 we start by defining a database, and then we explain other basic terms. In Section 1.2, we provide a simple UNIVERSITY database example to illustrate our discussion. Section 1.3 describes some of the main characteristics of database systems, and Sections 1.4 and 1.5 categorize the types of personnel whose jobs involve using and interacting with database systems. Sections 1.6, 1.7, and 1.8 offer a more thorough discussion of the various capabilities provided by database systems and discuss some typical database applications. Section 1.9 summarizes the chapter.

The reader who desires a quick introduction to database systems can study Sections 1.1 through 1.5, then skip or browse through Sections 1.6 through 1.8 and go on to Chapter 2.

1.1 Introduction

Databases and database technology have had a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, social media, engineering, medicine, genetics, law, education, and library science. The word *database* is so commonly used that we must begin by defining what a database is. Our initial definition is quite general.

A **database** is a collection of related data.¹ By **data**, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. Nowadays, this data is typically stored in mobile phones, which have their own simple database software. This data can also be recorded in an indexed address book or stored on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database.

The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to

¹We will use the word *data* as both singular and plural, as is common in database literature; the context will determine whether it is singular or plural. In standard English, *data* is used for plural and *datum* for singular.

- INTERSECT operation, SQL sets, 194–195
- INTERSECTION operation, 247–249
- INTERVAL data type, 184
- INTO clause, 317
- Intraquery parallelism, 687
- inverse references, 366, 370, 396–397
- Inverse relationships, ODMG objects, 396–397
- Inverted files, 641
- Inverted indexing
 - construction of, 1041–1042
 - defined, 1041
 - information retrieval (IR), 1040–1044
 - Lucern indexing/search engine for, 1043–1044
 - process of, 1042
- IS-A relationship, 109, 126
- IS/IS NOT comparison operators, 209
- Isolation. *See also* Snapshot isolation
 - levels of in transactions, 758
 - property, transactions, 14, 158
- Iterator object, ODMG models, 393
- Iterator variables
 - query results and, 312
 - OQL, 409–410
- Iterators
 - defined, 682
 - pipelining implementation using, 682–683
 - SQLJ query result processing with, 323–325
- Java
 - embedding SQL commands (SQLJ), 321–325
 - exceptions for error handling, 322–323
 - Web programming technologies, 358
- Java server pages (JSP), 358
- Java servlets, 358
- JavaScript, 358
- JavaScript Object Notation (JSON), 358
- JDBC (Java Database Connectivity)
 - class library imported from, 331, 332
 - drivers, 331–332
 - programming steps, 332–335
 - SQL class library, 326, 331–335
 - two-tier client/server architecture and, 49
- Join attribute, 253
- Join condition, 189, 191, 252, 278
- Join dependency (JD), 5NF, 494–495
- JOIN operations
 - aggregate operation implementation and, 678–679
 - anti-join (AJ) operator, 658–660, 677–678, 681, 719–720
 - attributes, 668
 - bucket join, 931
 - buffer space and, 672–673
 - cardinality, 719–720
 - cost functions for, 717–726
 - distributed query processing, 862–863
 - dynamic programming approach to ordering, 725–726
 - EQUIJOIN (=) comparison operator, 253
 - hybrid-hash join, 675–676
 - index-based nested-loop join, 559, 718–719
 - inner/outer, 254, 263–264
 - join selectivity (js) operator, 717–718
 - MapReduce (MR), 930–932
 - map-side hash join, 930
 - multiway joins, 668
 - N-way joins, 931–932
 - NATURAL JOIN (*) comparison operator, 253, 262–263
 - nested-loop join, 558–559, 672–673, 718
 - non-equi-join, 681
 - optimization based on cost formulas, 720–721
 - ordering choices in multirelational queries, 721–724
 - OUTER JOIN operations, 262–264, 679–681
 - parallel algorithms, 685–686
 - partition-hash join, 559, 674–675, 719, 930–931
 - performance of, 673–674
 - physical optimization, 724
 - query processing implementation, 668–676, 679–681
 - recursive closure operations, 262
 - relational algebra and, 251–255, 262–264
 - semi-join (SJ) operator, 658–660, 681, 719–720, 862–863
 - SQL query retrieval, 215–216
 - SQL relations, 215–216
 - sort-merge join, 559, 719, 930
 - two-way join, 668
- k-means algorithm, 1088–1089
- Key constraints
 - attributes, 68–69, 302
 - database integrity and, 21
 - integrity constraints and, 163–165
 - referential integrity constraints and, 163–165
 - relational modeling and, 158–160, 163–165
 - relational schema and, 157–165
 - surrogate, 302
 - uniqueness property, 68–69, 159
- Key field, records, 568
- Key-value storage (data models), 34, 51, 53
- Key-value stores, NOSQL, 888, 895–900
- Keys
 - attributes, 477
 - candidate key, 159–160, 477
 - composite keys, 631
 - defined, 476
 - foreign keys, 163–165, 186–187
 - indexes with, 631–633
 - multiple keys, 631–633
 - normal forms and, 476–477
 - ODMG object model, 398
 - primary key, 159, 186–187, 441, 477
 - SQL, 186–187
 - superkey, 158–159, 476–477
 - unique keys, 160
 - XML schema specification, 441
- Keyword-based data search, 41
- Keyword queries, 1035
- Knowledge discovery in databases (KDD), 1070–1073
- Knowledge representation (KR)
 - abstraction concepts, 129–133
 - domain of knowledge for, 129
 - EER modeling and, 128–129
 - ontology and, 129
 - reasoning mechanisms, 129
- Label-based security policy
 - architecture, 1156–1157
 - multilevel security, 1139–1140
- Oracle, 1155–1158
- Virtual private database (VPD)
 - technology, 1156
- Language design for database programming, 312, 339
- Latches, concurrency control and, 807
- Late (dynamic) binding, 377
- Lattices
 - EER models, 116–119
 - generalization, 119
 - inheritance and, 117–118
 - specialization, 116–119
- Lazy updates, SQL views, 230
- Leaf class, 127
- Leaf nodes, tree structures, 257, 617, 623
- Least recently used (LRU) strategy, buffering, 559
- Legacy data models, 33, 51, 53
- Legal relation states (extensions), 472
- Level trigger, 967
- Library of functions or classes
 - application programming interface (API), 312, 326
 - database programming approach, 311, 338–339
 - JDBC: SQL class library, 326, 331–335
 - SQL/CLI (SQI call level interface), 326–331
- Lifetime of an object, 388
- LIKE comparison operator, SQL, 195–196
- Linear hashing, 580–582
- Linear regression, data mining, 1092
- Linear scale-up, 684
- Linear search, files, 564, 567–568
- Linear speed-up, 684
- Link structure analysis, Web search and, 1050–1051
- Linked allocation, file blocks, 564

- Links, UML class diagrams, 87
- List constructor, 369
- Literal declaration, 392
- Literals
 - atomic (single-valued) types, 368, 388
 - collection, 392
 - constructors for, 368–370
 - deductive databases, 1002–1003
 - objects compared to, 368
 - ODBs, 368–370, 388, 392
 - ODMG models, 388, 392
 - structured, 388
 - type generators, 368–369
 - type structures for, 368–370
- Loading utility, 45
- Local area network, 842
- Local depth, hashing, 578
- Local query optimization, 860
- Localization, DDB query processing, 859
- Location analysis, 988
- Location transparency, DDBs, 843
- Locking data items, 781
- Locks
 - binary locks, 782–784
 - certify locks, 796–797
 - concurrency control and, 782–786, 796–797, 805–806
 - conversion of, 786
 - downgrading, 786
 - index concurrency control using, 805–806
 - shared/exclusive (read/write) locks, 784–786
 - upgrading, 786, 797
- Log buffers, 755, 756
- Log sequence number (LSN), 828
- Logic databases, 962
- Logical (conceptual) level, RDB design, 459–460
- Logical comparison operators, SQL, 188–190
- Logical data independence, 37–38
- Logical database design, *see* Data model mapping
- Logical design, 62
- Logical index, 638–639
- Logical theory, ontology as, 134
- Loss of confidentiality, database threat of, 1122
- Loss of integrity, database threat of, 1122
- Lossy design, 515
- Lost update problem, transaction processing, 750
- Low-level (physical) data models, 33–34
- Low-level (procedural) DML, 40
- Lucern indexing/search engine, 1043–1044
- Magnetic tape
 - backing up databases using, 555–556
 - memory hierarchy and, 544–545
 - storage devices, 555–556
 - tape reel, 555
- Main (master) file, 571
- Main memory, 543
- Maintenance, databases, 6
- Maintenance personnel, 17
- Mandatory access control (MAC), 1121, 1134–1137
- Mandatory security mechanisms, 1123
- Map data, 989
- Mappings
 - data model, 62
 - database schema views, 37
 - distributed query processing, 859
 - EER model constructs to relations, 298–303
 - EER schema to ODB schema, 407–408
 - ER-to-relational, 290–298
 - ODB conceptual design, 407–408
 - tuples for relations, 154
- MapReduce (MR)
 - advantages of technology, 936
 - Big data technology for, 917–921, 926–936
 - historical background of, 917–918
 - joins in, 930–932
 - parallel RDBMS compared to, 944–946
 - programming model, 918–921
 - runtime, 927–930
- Map-side hash join, MapReduce (MR), 930
- Mark up, XML documents for HTML, 428–429
- Market-basket data model, 1073–1075
- Mass storage, 543
- Master data management (MDM), 1110
- Master-master replication, NOSQL, 886
- Master-slave replication, NOSQL, 886
- Materialized evaluation, 681, 702–702
- Materialized views, query execution, 707–710
- Mathematical relation, domains, 152
- MAX function, SQL, 217
- MAXIMUM function, grouping, 260
- Measurement operations, 988
- Mechanical arm, disk devices, 551
- Memory
 - cache, 543
 - dynamic random-access (DRAM), 543
 - flash memory, 543–544
 - hierarchies, 543–545
 - magnetic tape, 544–545
 - main, 543
 - optical drives, 544
 - random-access (RAM), 543
 - storage capacity and, 543
 - storage devices for, 543–545
- Menu-based interfaces, 40
- Merging phase, external algorithms, 661
- Meta-data
 - database catalog and, 10–11
 - defined, 6
 - schema storage, 35
- Methods
 - database operations, 12
 - object data models, 53
 - operation implementation and, 366, 371
- Middle-tier Web server, PHP as, 344
- Middleware layer, *n*-tier architecture, 50–51
- MIN function, SQL, 217
- Minimal sets of functional dependency, 510–512
- MINIMUM function, grouping, 260
- Miniworld, 5
- MINUS operation, 247–249
- Mirroring, (shadowing), RAID, 585
- Mixed (hybrid) fragmentation, DDB data, 847–848
- Mixed records, files for, 582–583
- Mobile applications, access control of, 1141–1142
- Mobile device apps
 - ER modeling and, 59
 - interfacing, 40–41
 - user transactions by, 16
- Model-theoretic interpretation of rules, 1005
- Models, *see* Data models; EER (Enhanced Entity-Relationship) model; ER (Entity-Relationship) model; Object data models
- Modification anomalies, RDB design and, 467
- Modifier, object operations, 371
- Modules
 - buffering (caching), 20, 42
 - client module, 31
 - compilers, 42–45
 - database queries and, 20, 43–44
 - database systems, 31, 42–45
 - DBMS components, 42–45
 - interactive query interface, 43–44
 - server module, 31
 - stored data manager, 42
- MOLAP (multidimensional OLAP) function, 1114
- MongoDB data model
 - CRUD operations, 893
 - documents, 890–893
 - NOSQL, 890–895
 - replication in, 894
 - sharding in, 894–895
- Moveable head disks, 551
- Multidatabase system recovery, 831–834
- Multidimensional models, 1108
- Multilevel indexes
 - dynamic, 616, 617–630
 - fan-out, 613, 622
 - levels, 613–616
 - physical database design and, 613–617

- Multimedia databases
 - audio data source analysis, 999
 - concepts, 994–996
 - enhanced data models, 962, 994–999
 - image automatic analysis, 996–997
 - object recognition, 997–998
 - semantic tagging of images, 998–999
 - types of, 3–4
- Multiple granularity locking
 - concurrency control and, 801–804
 - granularity levels for, 801
 - granularity of data items, 800–801
 - protocol, 802–804
- Multiple hashing, collision resolution, 575
- Multiple inheritance, 118, 301, 377–378, 393
- Multiple keys
 - grid files and, 632–633
 - indexes on, 613–633
 - multiple attributes and, 631–632
 - ordered index on, 631–632
 - partitioned hashing with, 632
 - physical database design and, 613–633
- Multiple-relation options, EER-to-relation mapping, 299–300
- Multiple user interfaces, 20–21
- Multiplicities, UML class diagrams, 87
- Multiprogramming
 - concept of, 746–747
 - operating systems, 747
- Multirelational queries, JOIN ordering
 - choices and, 721–724
- Multiset (tuple) operations
 - comparisons for query retrieval, 209–211
 - SQL tables, 193–195
- Multisuser DBMS systems, 51
- Multisuser transaction processing, 13–14
- Multivalued attributes, 66, 295–296, 481
- Multivalued dependency, *see* MVD (multivalued dependency)
- Multiversion concurrency control, 781, 795–797
 - certify locks for, 796–797
 - timestamp ordering (TO), 796
 - two-phase locking (2PL), 796–797
- Multisway joins
 - implementing, 668
 - SQL table (relations), 216
- Mutator function, SQL encapsulation, 384
- MVD (multivalued dependency)
 - all-key relation of, 491, 493
 - definition of, 491–492
 - fourth normal form (4NF) and, 491–494, 527–530
 - inference rules for, 527–528
 - normalizing relations, 493–494
 - trivial/nontrivial, 493
- n*-ary relationship types, mapping of, 296
- n*-degree relationships, 88–92
- n*-tier architecture for Web applications, 49–51
- N*-way joins, MapReduce (MR), 931–932
- Named iterator, SQLJ, 323
- Namespace, XML, 440
- Naming mechanisms
 - constraints, SQL, 187
 - database entrypoints, 373
 - object persistence and, 373–374
 - operations for renaming attributes, 245–246
 - query retrieval and, 192, 214–215
 - renaming attributes, 192, 214–215, 245–246
 - schema constructs, 82
- Naming transparency, DDBs, 843
- NATURAL JOIN (*) comparison operator, 253, 262–263
- NATURAL JOIN operation, SQL tables, 215
- Natural language interfaces, 41
- Natural language queries, 1037
- Neo4j system
 - cypher query language of, 905–908
 - distributed system concepts for, 908–909
 - nodes, 904–905
 - NOSQL, 903–909
 - relationships, 904–905
- Nested-loop join, 558–559, 672–673, 718
- Nested queries
 - comparison operators for, 210–211
 - correlated, 211–212
 - innermost query of, 211
 - outer query of, 209
 - query optimization and, 702–704
 - subqueries, 702–704
 - tuple values in, 209–211
 - unnesting (decorrelation), 704
- Nested relations, 1NF in, 479–480
- Network-attached storage (NAS), 589–590
- Network data models, 33, 51, 53
- Network systems using databases, 23–24
- Network topologies, 843
- Neural networks, data mining, 1092
- No waiting algorithm, deadlock prevention, 791
- NodeManager, YARN, 942
- Nodes
 - constant, query graphs, 273
 - leaf, query trees, 257
 - relation, query graphs, 273
 - tree structures, 617
- Non-equi join implementation, 681
- Nonadditive (lossless) join property algorithms, 519–523
 - Boyce-Codd normal form (BCNF) schemas using, 522–523
 - dependency preservation and, 519–522
 - 4NF schema using, 530
 - normalization process, 476
 - RDB decomposition, 515–518, 519–522
 - successive decompositions, 517–518
 - testing binary decompositions for, 517
 - 3NF schema using, 519–522
- Nonadditive join test for binary decomposition (NJB), 490
- Noninstantiable object behavior, interface and, 392
- Nonprocedural language, 268
- Nonrecursive query evaluation, 1010–1012
- Nonserial schedules, 763, 764–765
- Normal form test, 475
- Normal forms
 - Boyce-Codd normal form (BCNF), 487–491
 - defined, 475
 - denormalization, 476
 - domain-key (DKNF), 532–533
 - fifth normal form (5NF), 494–495
 - first normal form (1NF), 477–481
 - fourth normal form (4NF), 491–494
 - insufficiency of for relational decomposition, 513–514
 - join dependency (JD) and, 494–495
 - keys, attributes and definitions for, 476–477
 - multivalued dependency (MVD) and, 491–494
 - normalization of relations, 474–476, 482, 485, 486–487, 493–494
 - practical use of, 476
 - primary keys for, 483–495
 - RDB design and, 474–495, 513–514, 528–533
 - second normal form (2NF), 481–482, 484–486
 - third normal form (3NF), 483–484, 486–487
- Normalization process
 - algorithms, 519–527
 - data normalization, 475–476
 - dependency preservation property, 476
 - multivalued dependency (MVD), 493–494
 - nonadditive (lossless) join property, 476
 - normal form test for, 475
 - relations, 474–476
- NOSQL database system
 - availability, 885–886
 - big data storage uses, 3, 26
 - CAP theorem, 888–890
 - categories of, 887–888
 - column-based, 888, 900–903
 - CRUD (create, read, update, and delete) operations, 887, 893, 903
 - data models, 34, 51
 - DDB similar characteristics, 885–887
 - distributed storage using, 883
 - document-based, 888, 890–895
 - emergence of, 884–885
 - eventual consistency, 885–886
 - graph-based, 888, 903–909
 - Hbase data model, 900–903

- high-performance data access, 886–887
- key-value stores, 888, 895–900
- MongoDB data model for, 890–895
- Neo4j system, 903–909
- query language similar characteristics, 887
- replication models for, 886
- replication, 885–886, 894
- scalability, 885
- sharding, 886, 894–895
- versioning, 887, 899, 900–902
- NOT FINAL, UDT inheritance specification, 385
- NOT operator, *see* AND/OR/NOT operators
- NO-UNDO/REDO algorithm, 815, 821–823
- NULL values
 - aggregate functions and, 218
 - attribute not applicable, 208
 - complex query retrieval and, 208–209
 - constraints on attributes, 160, 184–186
 - discarded values, 218
 - entity attributes, 66
 - grouping attributes with, 219
 - IS/IS NOT comparison operators for, 209
 - query retrieval in SQL, 208–209, 218, 219
 - RDB design problems, 523–524
 - referential integrity and, 163–164
 - relational modeling and, 155–156, 160
 - relation schema for RDB design and, 467–468
 - grouping attributes, 219
 - SQL attribute constraints, 184–186
 - three-valued logic for comparisons, 208–209
 - tuples for relations, 155–156, 163, 467–468
 - unavailable (or withheld) value, 208
 - unknown value, 208
- Numeric arrays, PHP, 349
- Numeric data types, 182, 348
- Object-based storage, 591–592
- Object Data Management Group, *see* ODMG (Object Data Management Group)
- Object data models
 - classes, 52
 - data model type, 33
 - DBMS classification from, 51, 52–53
 - hierarchies (acyclic graphs), 52
 - methods, 53
 - ODMG, 387–400
- Object databases, *see* ODBs (object databases)
- Object definition language, *see* ODL (object definition language)
- Object identifier, *see* OID (object identifier)
- Object identity
 - literal values for, 368
 - ODBs, 367–368, 378
 - OID implementation of, 367
 - SQL, 379
- Object-oriented systems, persistent storage, 19–20
- Object query language, *see* OQL (object query language)
- Object recognition, multimedia databases, 997–998
- Object-relational systems
 - extended-relational systems, 53
 - SQL, 202
- Objects
 - arrow (→) notation for, 392
 - atomic (single-valued) types, 368, 388, 396–398
 - attributes, 396
 - behavior of based on operations, 371
 - collections, 373, 376
 - constructors for, 368–370
 - dot notation for, 372, 392
 - encapsulation of, 366, 371
 - exceptions, 397–398
 - hidden attributes, 371
 - instance variables, 365–366
 - interfaces, noninstantiable behavior and, 392
 - lifetime, 388
 - literals compared to, 368
 - naming, 373–374, 387
 - ODBs, 365–371, 387–388, 395–400
 - ODMG models, 387–388, 392, 395–400
 - operations for, 370–372
 - persistent, 365, 373–374, 376
 - reachability, 373–374
 - relationships, 396–397
 - signatures, 366, 397
 - state of, 387
 - structure of, 388
 - transient, 365, 373, 376
 - type generators, 368–369
 - type structures for, 368–370
 - unique identity, 367–368
 - visible/hidden attributes, 371, 375
- Observer function, SQL encapsulation, 384
- ODBC (Open Database Connectivity)
 - data mining, 1094–1095
 - standard, 49, 326
- ODBs (object databases)
 - C++ language binding, 417–418
 - conceptual design, 405–408
 - development of, 363–365
 - encapsulation of operations, 366, 370–374, 384–385
 - inheritance and, 366, 374–377, 378, 385, 393
 - instance variables, 365–366
 - inverse references, 366, 370, 396–397
 - literals in, 368–370, 388–392
 - Object Data Management Group (ODGM) model, 386–405, 417–418
 - object definition language (ODL) and, 386, 400–405
 - object identifier (OID), 367–368
 - object query language (OQL), 408–416
 - object-oriented (OO) concepts, 365–366
 - objects in, 365–371, 387–388, 395–400
 - polymorphism (operator overloading), 366, 377
 - RDB compared to, 405–406
 - SQL extended from, 379–386
 - type (class) hierarchy, 366, 374–377
- ODL (object definition language)
 - classes, 400, 404–405
 - class-schema interface inheritance, 401–404
 - Object Data Management Group (ODGM) model and, 386, 400–405
 - object databases (ODBs) and, 386–387, 400–405
 - schemas, 400–403
 - type constructors in, 369
- ODMG (Object Data Management Group)
 - atomic (user-defined) objects, 395–398
 - bindings, 386, 417–418
 - built-in interfaces and classes, 393–396
 - C++ language binding, 386, 417–418
 - database standard, 33, 364–365
 - extents, 373, 376–377, 398
 - factory objects, 398–400
 - inheritance in object models, 393
 - interface definitions for object models, 389–392
 - keys, 398
 - literals in object models, 388, 392
 - object databases (ODBs), 386–405, 417–418
 - object definition language (ODL) and, 386, 400–405
 - object model of, 387–400
 - object query language (OQL) and, 386, 408
 - objects, 387–388, 392, 395–400
 - standards, 386, 417–417
- OID (object identifier)
 - immutable property of, 367
 - ODB unique object identity and, 367–368
 - ODMG models, 387
 - reference types used for in SQL, 383
- OLAP (Online analytical processing)
 - data warehousing and, 1102
 - data warehousing characteristics and, 1104
 - HOLAP (hybrid OLAP) option, 1114
 - MOLAP (multidimensional OLAP) function, 1114
 - ROLAP (relational OLAP) function, 1114
 - use of, 4

- OLTP (online transaction processing)
 - data warehousing and, 1102
 - multiuser transaction processing, 14
 - relational data modeling, 169
 - special-purpose DBMS use, 52
- Online analytical processing, *see* OLAP (Online analytical processing)
- Online transaction processing, *see* OLTP (online transaction processing)
- Ontology
 - conceptualization and, 134
 - defined, 134
 - knowledge representation (KR) and, 129
 - semantic Web data models, 133–134
 - specification and, 134
 - types of, 134
- Ontology-based information integration, 1052–1053
- OO (object-oriented) concepts, 365–366
- OODB (object-oriented database)
 - attribute versioning, 982–984
 - database complexity and, 24–25
 - development of, 363
 - temporal databases incorporating time in, 982–984
- OQL (object query language)
 - aggregate functions, 413–414
 - Boolean (true/false) results, 414
 - collection operators, 413–416
 - element operator, 413
 - exists quantifier, 415
 - grouping operator, 415–416
 - indexed (ordered) collection expressions, 415
 - iterator variables for, 409–410
 - named query specification, 412–413
 - ODBs, 408–416
 - ODGM model queries and, 408–416
 - ODMG standard and, 386
 - path expressions, 410–412
 - query results, 410–412
 - select...from...where structure of, 409
- OOPL (object-oriented programming language), class library for, 312
- op comparison operator, 270
- Open addressing, hashing collision resolution, 574
- OPEN CURSOR command, SQL, 317
- OpenPGP (Pretty Good Privacy) protocol, XML, 1140–1141
- Operating system (OS), 42
- Operational data store (ODS), 583, 1105
- Operations. *See also* Query processing strategies
 - aggregate, 678–679
 - assignment (←) for, 245
 - binary, 240, 251–259, 262–264
 - defined, 12
 - delete, 166, 167–168
 - dot notation for objects, 372
 - encapsulation, 366, 370–374, 384–385
 - files, 564–567
 - generalized projection, 259–260
 - insert, 166–167
 - JOIN, 251–255, 262–264, 668–676
 - method (body) of, 366, 371
 - ODBs, 366, 370–374, 384–385
 - pipelining for combinations of, 681–683
 - program variables for, 565–566
 - record-at-a-time, 566
 - recursive closure, 262
 - relational algebra, 240–259, 262–265
 - relational data modeling, 165–168
 - renaming attributes, 245–246
 - retrievals, 165–166, 564–565
 - schedules, 759–760, 773
 - selection conditions for, 564–565
 - sequence of, 245–246
 - set-at-a-time, 566
 - set theory and, 246–251, 264–265
 - signature (interface) of, 366, 371
 - SQL query recovery and, 194–197
 - SQL sets, 194–195
 - unary, 240, 241–246
 - UNION, 194–195, 264–265
 - update (modify), 166, 168–169, 564–565
 - user-defined functional requirements, 61
- Operator-level parallelism, 684–686
- Operators
 - aggregate functions, 216–219, 260–261
 - arithmetic, SQL, 196–197
 - collections, 413–416
 - comparison, 209–211
 - nested queries, 209–211
 - defined, 17
 - grouping, 415–416
 - logical comparison, SQL, 188–190
 - OQL collections, 413–416
 - spacial, 990–991
 - SQL query recovery, 188–190, 196–197, 209–211
 - SQL query translation into, 657–660
- Optical drives, 544
- Optimistic protocols, 781
- Optional field, records, 561–562
- OR logical connective, SQL, 209–210
- OR operator, *see* AND/OR/NOT operators
- Oracle
 - adaptive optimization, 735
 - array processing, 735–736
 - global query optimizer, 734–735
 - hints, 736
 - key-value store, 899
 - label-based security policy, 1155–1158
 - outlines, 736
 - physical optimizer, 733–734
 - query optimization in, 733–737
 - SQL plan management, 736–737
 - virtual private database (VPD) technology, 1156
- ORDBMS (object-relational database management system), 364
- ORDER BY clause
 - SQL, 197–198
 - XQuery, 446
- Order preserving, hashing, 577
- Ordered (sorted) records, 568–572
- Ordering field, records, 568
- OUTER JOIN operations, 216, 262–264
- Outer query, 209
- OUTER UNION operation, 264–265
- Outlines, Oracle, 736
- Overflow (transaction) file, 571
- Overlapping entities, 115, 126
- PageRank ranking algorithm, 1051
- Parallel algorithms
 - aggregate operations for, 686
 - architectures for, 683–684
 - interquery parallelism, 687
 - intraquery parallelism, 687
 - join techniques, 685
 - operator-level parallelism, 684–686
 - partitioning strategies, 684
 - projection and duplicate elimination, 685
 - query processing using, 683–687
 - selection conditions, 685
 - set operations for, 686
 - sorting, 684
- Parallel database architecture, 683
- Parallel processing, 747
- Parameters
 - binding, 329, 333
 - disks, 1167–1169
 - JDBC statement parameters, 333
 - SQL/CLI statement parameters, 329
 - stored procedure type and mode, 336–337
- Parametric (naïve) end users, 16
- Parametric user interfaces, 42
- Parent nodes, tree structures, 617
- Parser, query processing, 655
- Partial categories, 122
- Partial key, 79, 479
- Partial specialization, 115, 126
- Participation constraints, 77–78
- Partition algorithm, 1081
- Partition-hash join, 559, 674–675, 719, 930–931
- Partition tolerance, DDBs, 845
- Partitioned hashing, 632
- Partitioning strategies
 - NOSQL, 886
 - parallel algorithms, 684
- Partitions
 - key-value 416
 - grouping and, 219, 415–416
 - SQL query retrieval and, 219
- Path expressions
 - OQL, 410–412
 - SQL, 386
 - XPath for, 443–445
- Path separators (/ and //), XML, 443
- Patterns, substring matching in SQL, 195–197

- PEAR (PHP Extension and Application Repository), 353–354
- Performance, Big data technology and, 945
- Performance monitoring, 45
- Periodic updates, SQL views, 230
- Persistent data, storage of, 545
- Persistent objects, 365, 373–374
- Persistent storage, 19–20
- Persistent storage modules, 336
- Phantom records, concurrency control and, 806–807
- PHP (Hypertext processor)
 - arrays, 345–346, 348–350
 - built-in variables, 352–353
 - comments in, 345
 - connecting to a database, 353–355
 - data collection and records, 355–356
 - error checking, 355
 - Extension and Application Repository (PEAR), 353–354
 - functions, 350–352
 - here documents, 347–348
 - HTML and, 343–346
 - middle-tier Web server as, 344
 - numeric data types for, 348
 - placeholders, 356
 - predefined variables, 345–346
 - query retrieval, 356–357
 - query submission, 355
 - text strings in, 346, 347–348
 - use of, 343–345
 - variable names for, 346, 347
 - Web programming using, 343–359
- Phrase queries, 1036
- Physical clustering, mixed records, 583
- Physical data independence, 38
- Physical data models, 33–34
- Physical database design
 - data storage and, 546
 - indexing design decisions, 645–646
 - indexing structures, 601–652
 - job mix factors for, 643–645
 - multilevel indexes, 613–617
 - relational databases (RDBs) with, 643–646
 - single-level ordered indexes, 602–613
- Physical database file structures, 641. *See also* Indexes
- Physical design, data modeling, 62
- Physical index, 638–639
- Physical optimization, queries, 724
- Physical optimizer, Oracle, 733–734
- Pin count, buffer management, 558
- Pin-unpin bit, database recovery cache, 816
- Pipelined parallelism, 687
- Pipelining
 - combining operations using, 681–683
 - iterators for implementation of, 682–683
 - materialized evaluation and, 681
 - pipelined evaluation, 682
 - processing information, 1028–1029
 - query processing using, 681–683
- Placeholders, PHP, 356
- Plan caching, query optimization, 730
- Pointers
 - B-trees, 620, 623–624
 - file records, 563, 575–576
- Polymorphism (operator overloading)
 - binding and, 377
 - ODBs, 366, 377
- Populating (loading) databases, 35
- Positional iterator, SQL, 323
- Practical relational model, 177–206.
 - See also* SQL (Structured Query Language) system
- Precompiler
 - DML command extraction, 44
 - embedded SQL and, 311, 314
- Predefined variables, PHP, 345–346
- Predicate, relation schema and, 156
- Predicate-defined subclasses, 113, 126
- Prefix compression, string indexing, 640
- PreparedStatement objects, JDBC, 333
- Preprocessor, embedded SQL and, 311, 314
- Primary file organization, 546
- Primary indexes, 602, 603–606
- Primary keys
 - arbitrary designation of, 477
 - normal form based on, 483–495
 - relational data modeling, 159
 - SQL constraints, 186–187
 - XML specification, 441
- Primary storage, 542, 543
- Prime/nonprime attributes, 477
- Printer servers, client/server architecture, 47
- Privacy issues and preservation, 1153–1154
- Privileged software use, 19
- Privileges, granting and revoking in SQL, 202
- Probabilistic model, IR, 1033–1034
- Probabilistic topic modeling, IR, 1059–1061
- Program variables
 - embedded SQL, 314–315
 - file operations, 565–566
- Program-data independence, 12
- Programming, *see* Database programming; SQL programming
- Programming languages
 - DBMS, 38–40
 - declarative, 40
 - design for database programming, 312–313, 339
 - impedance mismatch, 312–313
 - Java, 321–325, 358
 - PHP (Hypertext processor), 343–359
 - QBE (Query-by-Example), 1171–1178
 - XML, 434, 436–447
- Programming model, MapReduce (MR), 918–921
- Program-operation independence, 12
- Project attributes, 189
- PROJECT operation
 - degree of relations, 244
 - duplicate elimination and, 244–245
 - query processing, algorithms for, 676–678
 - relational algebra using, 243–245
- Prolog language, deductive databases, 1000–1003
- Proof by contradiction, 507
- Proof-theoretic interpretation of rules, 1005
- Properties of decomposition
 - attribute preservation condition, 513
 - dependency preservation, 514–515
 - insufficiency of normal forms, 513–514
 - nonadditive (lossless) join, 515–517, 519–523
 - RDB design and, 504, 513–518
 - universal relations and, 513
- Protection, databases, 6
- Proximity queries, 1036
- Public key encryption, 1151–1152
- Pure distributed database architecture, 869–871
- QBE (Query-by-Example) language
 - aggregate functions in, 1175–1177
 - grouping, 1175–1177
 - modifying the database, 1177–1178
 - retrievals in, 1171–1175
- Qualified association, UML class diagrams, 88
- Qualifier conditions, XML, 443
- Quantifiers
 - domain relational calculus, 279
 - existential, 271, 274
 - queries using, 274–276
 - transformation of, 274
 - tuple relational calculus, 271, 274–276
 - universal, 271, 274–276
- Queries
 - buffering (caching) modules for, 20, 42
 - compiler, 43–44
 - complex retrieval, 207–225
 - constant nodes, 273
 - Datalog language, 1004, 1010–1012
 - defined, 6
 - indexes for, 20
 - indexing hints in, 641–642
 - information retrieval (IR) systems, 1035–1037
 - interactive interface, 43–44
 - join condition, 189, 191
 - keyword-based, 41
 - named specification, OQL, 412–413
 - nested, 209–212
 - nonrecursive evaluation, 1010–1012
 - object query language (OQL), 408–416
 - ODMG model for, 408–416
 - optimizer, 44
 - outer, 209
 - processing in databases, 20

Queries (*continued*)

- quantifiers for, 274–276
 - recursive, 223
 - relation nodes, 273
 - relational algebra for, 265–268
 - select-from-where structure, 188–190
 - selection condition, 189
 - select-project-join, 189–190, 273
 - spatial, 991
 - SQL retrieval, 187–198, 207–225
 - temporal constructs, 984–986
 - TSQL2 language for, 984–986
 - tuple relational calculus for, 272–276
 - XML languages for, 443–447
- Query block, 657–658
- Query decomposition, DDBMS, 863–865
- Query execution
- aggregate functions for, 709
 - cost components for, 711–712
 - GROUP-BY view merging, 705–706
 - incremental view maintenance, 707–710
 - materialized views for, 707–710
 - nested subqueries, 702–704
 - query evaluation for, 701–702
 - subquery (view) merging
 - transformation for, 704–706
- Query graphs
- internal query representation by, 655
 - notation, 692–694
 - query optimization, 692–697
 - tuple relational calculus, 273–274
- Query modification, SQL views, 229–230
- Query optimization
- cost estimation for, 657, 710–713, 716–717
 - cost functions for, 714–715, 717–726
 - cost-based optimization, 710–712, 716, 726–728
 - data warehouses, 731–733
 - distributed databases (DDBs), 859–863
 - dynamic programming, 716, 725–726
 - execution plan, display of, 729
 - heuristic rules for, 657, 692, 697–701
 - histograms for, 713
 - JOIN operation for, 717–726
 - multirelation queries, 721–724
 - operation size estimation, 729–730
 - Oracle, 733–737
 - physical optimization, 724
 - plan caching, 730
 - query execution and, 701–712
 - query processing compared to, 655–657
 - query trees and graphs for, 692–697
 - SELECT operation for, 714
 - semantic query optimization, 737–738
 - star-transformation optimization, 731–733
 - top-*k* results, 730
 - transformation rules for relational algebra operations, 697–699
- Query optimizer, 655

Query processing strategies

- aggregate operation implementation, 678–679
 - anti-join (AJ) operator for, 658–660
 - distributed databases (DDBs), 859–863
 - external sorting algorithms, 660–663
 - importance of, 656–657
 - JOIN operation implementation, 668–676, 679–681
 - parallel algorithms for, 683–687
 - pipelining to combine operations, 681–683
 - PROJECT operation algorithm, 676–678
 - query block for, 657–658
 - query optimization compared to, 655–657
 - SELECT operation algorithms, 663–668
 - semi-join (SJ) operator for, 658–660
 - set operation algorithm, 676–678
 - SQL query translation, 657–660
 - steps for, 655–656
- Query results
- bound columns approach, 329
 - cursor (iterator variable) for, 312, 317–320
 - embedded SQL, 312, 317–320
 - impedance mismatch and, 312
 - iterators for, 323–325
 - OQL, 410–412
 - path expressions, 386, 410–412
 - PHP, 356–357
 - SQL/CLI processing, 329
 - SQLJ processing of, 323–325
- Query retrieval
- aggregate functions in, 216–219
 - alias for, 192
 - arithmetic operators for, 196–197
 - asterisk (*) uses, 193, 218
 - attribute name qualification, 191
 - Boolean (TRUE/FALSE) statements for, 212–214
 - CASE clause for, 222–223
 - clauses used in, 198–199
 - comparison operators, 188–191, 195–197
 - complex queries, 207–225
 - EXISTS function for, 212–214
 - explicit sets of values, 214–215
 - FROM clause for, 188–189, 197, 232
 - grouping, 216–222
 - joined tables (relations), 215–216
 - LIKE comparison operator, 195–196
 - logical comparison operators for, 188–190
 - multiset of tuples, 188, 193–195
 - nested queries, 209–212
 - NULL values and, 208–209
 - ORDER BY clause for, 197–198
 - ordering results, 197
 - PHP, 356–357
 - QBE (Query-by-Example) language, 1171–1175
 - recursive queries, 223

- renaming attributes, 192, 214–215
 - SELECT statement (clause) for, 187–188, 194–195, 197
 - select-from-where block, 188–191
 - set operations for, 194–195
 - set/multiset comparisons, 209–211
 - SQL, 187–198, 207–225, 230–231
 - substrng pattern matching, 195–197
 - table set relations, 193–195
 - three-valued logic for comparisons, 208–209
 - tuple variables for, 192, 209–211
 - UNIQUE function for, 212–214
 - views (virtual tables) for, 230–231
 - WHERE clause for, 188, 192–193, 197
 - WITH clause for, 222–223
- Query server, two-tier client/server architecture, 49
- Query submission, PHP, 355
- Query tree
- defined, 257
 - heuristic optimization of, 694–694
 - internal query representation by, 655
 - notation, 257–259, 692–694
 - query optimization, 692–697
 - RDBMS use of, 257–259
 - semantic equivalence of, 694–695
 - query validation, 655
- Question answering (QA) systems, 1061–1063
- RAID (redundant arrays of inexpensive disks) technology
- bit-level striping, 584, 586
 - block-level striping, 584–585, 586
 - data striping, 584–585
 - levels, 586–588
 - mirroring, (shadowing), 585
 - parallelizing disk access using, 542, 584–588
 - performance, improvement with, 586
 - reliability, improvement with, 585–586
- Random-access memory (RAM), 543
- Random access storage devices, 554
- Range partitioning, 684, 886
- Range relations, tuple variables and, 269–270
- RDBMS (Relational database management system)
- query tree notation, 257–259
 - two-tier client/server architecture and, 49
- RDBs (relational databases)
- application flexibility with, 24
 - data abstraction in, 24
 - indexing for, 643–646
 - integrity constraints and, 160–163
 - physical database design in, 643–646
 - relation schema sets as, 160
 - schemas, 160–163
 - temporal databases incorporating time in, 977–982

- tuple versioning, 977–982
- valid and invalid relational states, 160–161
- Reachability, object persistence and, 373–374
- Read/write head, disk devices, 551
- Read/write transactions, 748
- Real-time database technology, 4
- Reasoning mechanisms, 129
- Recall and precision metrics, IR, 1044–1046
- Record type (format), 560
- Record-at-a-time, file operations, 566
- Record-at-time DML, 40
- Record-based data models, 33
- Records
 - blocking, 563–564
 - data types, 560–561
 - data values, 560
 - fields, 560, 561–563, 568–569, 582–583
 - file storage, 560–564, 567–572, 582–583
 - fixed-length, 561–563
 - mixed, 582–583
 - ordered (sorted), 568–572
 - spanned versus unspanned, 563–564
 - unordered (heaps), 567–568
 - variable-length, 561–563
- Recoverability basis of schedules, 761–762
- Recoverable/nonrecoverable schedule, 761
- Recursive closure operations, 262
- Recursive queries, 223
- Recursive (self-referencing) relationships, 75
- Redis key-value cache, 900
- Redundancy control, 18–19
- REF keyword, 383, 386
- Reference types, OIDs created using, 383
- References
 - dot notation for path expressions, 386
 - inverse, 366, 370, 396–397
 - object identity from, 370
 - object type relationships, 369–370
 - relationships specified by, 386
 - SQL, 370, 386
- Referential integrity
 - constraints, 21, 163–165, 186–187
 - NULL values and, 163–164
 - relational data modeling, 163–165
 - SQL constraints, 186–187
- Referential triggered action clause, SQL, 186
- Reflexive association, UML class diagrams, 87
- Regression, data mining, 1091–1092
- Regression function, data mining, 1092
- Relation extension/intension, 152
- Relation nodes, query graphs, 273
- Relation schema
 - anomalies and, 465–467
 - assertion, 156
 - attribute clarity and, 464
 - degree (arity) of attributes, 152
 - facts, 156
 - functional dependency of, 471–474
 - goodness of, 459
 - interpretation of, 156
 - key of, 159
 - nested relations, 479–480
 - normalization of relations, 474–476
 - NULL value in tuples, 467–468
 - predicate, 156
 - redundant information in tuples, 465–467
 - relational database (RDB) design guidelines, 461–471
 - relational model constraints and, 157–165
 - relational model domains and, 152
 - semantics of, 461–465
 - spurious tuple generation, 468–471
 - superkey of, 158–159
 - universal, 471–474
- Relation state
 - current, 153
 - relational model domains and, 152–153
 - relational database, 160–161
 - tuple values in, 152–156
 - valid and not valid, 160–161
- Relational algebra
 - aggregate functions, 240, 260–261
 - binary operations, 240, 251–259, 262–264
 - expressions for, 239, 241–242, 245
 - formal relational modeling and, 239–240
 - generalized projection operation, 259–260
 - groupings, 260–261
 - operations, purpose and notation of, 258
 - procedural order of, 268
 - queries in, 265–268
 - query optimization and, 697–699
 - recursive closure operations, 262
 - set theory and, 246–251, 264–265
 - SQL query translation into, 657–660
 - transformation rules for operations, 697–699
 - unary operations, 240, 241–246
- Relational calculus
 - declarative expressions for, 268
 - domains and, 268, 277–279
 - formal relational modeling and, 240–241
 - nonprocedural language of, 268
 - query graphs, 273–274
 - relationally complete language of, 268
 - tuples and, 268–277
- Relational data models
 - attributes, 152–153
 - breaking cycle for tree-structure model conversion, 452–453
 - concepts, 150–157
 - constraints, 157–167
 - DBMS criteria and, 51–52
 - delete operation, 166, 167–168
 - domains, 151–152
 - entity integrity, 163–165
 - extraction of XML documents using, 447–449
 - flat files, 150
 - formal languages for, *see* Relational algebra; Relational calculus
 - insert operation, 166–167
 - key constraints, 21, 158–160, 163–165
 - mathematical relation of, 149
 - notation for, 156–157
 - operations, 165–168
 - referential integrity, 163–165
 - practical language for, *see* SQL (Structured Query Language)
 - relations, 152–156
 - representational model type, 33
 - retrievals (operations), 165–166
 - schemas, 152–165
 - table of values, 150–151
 - transactions, 169
 - tuples, 152–156
 - update (modify) operation, 166, 168–169
- Relational database (RDB) design
 - algorithms for schema design, 519–523, 524–527
 - bottom-up method, 460, 504
 - by analysis, 503
 - by synthesis, 504, 503
 - dangling tuple problems, 523–524
 - data model mapping for, 289
 - designer intention for, 459–460
 - EER-to-relational mapping, 298–303
 - ER-to-relational mapping, 290–298
 - functional dependency and, 471–474, 505–512, 527–528, 532
 - implementation (physical storage) level, 459–460
 - inclusion dependency and, 531–532
 - inference rules for, 505–509, 527–528
 - join dependency (JD) and, 494–495, 530–531
 - keys for, 474–483
 - logical (conceptual) level, 459–460
 - multivalued dependency (MVD) and, 491–494, 527–530
 - normal forms, 474–495, 513–514, 528–533
 - normalization algorithm problems, 524–527
 - normalization of relations, 474–476, 482, 485, 486–487, 493–494
 - NULL value problems, 523–524
 - ODBs compared to, 405–406
 - properties of decomposition, 504, 513–518
 - relation schema, guidelines for, 461–471
 - top-down method, 460
 - universal relations, 471–474, 504

- Relational database management system, *see* RDBMS (Relational database management system)
 - Relational database state, 160–161
 - Relational databases, *see* RDBs (relational databases)
 - Relational operators for deductive databases, 1010
 - Relationally complete language of, 268
 - Relationships
 - aggregation, 87–88
 - associations, 87–88
 - attributes of, 78
 - attributes, as, 74
 - binary types, 76–78, 293–295
 - cardinality ratios for, 76–77
 - comparison of ternary and binary, 88–91
 - conceptual data models, 33
 - constraints on, 76–78, 91–92
 - degree of types, 71–74, 88
 - entity participation in, 72–73
 - ER models and, 72–78, 88–92
 - ER-to-relational mapping, 293–296
 - existence dependency, 77–78
 - identifying, 79
 - instances, 72
 - inverse, 396–397
 - multivalued attributes, 295–296
 - n*-degree, 88–92, 296
 - ODMG model objects, 396–397
 - order of instances in, 87
 - participation constraints of, 77–78
 - recursive (self-referencing), 75
 - role names and, 75
 - sets, 72
 - structural constraints of, 78
 - subtype/supertype, 375–376
 - ternary, 88–92
 - type, 72–78, 126
 - type hierarchies, 375–376
 - UML class diagrams, 87–88
 - Reliability, DDBs, 844–845
 - RENAME operator (*p*), 245–246
 - Renaming attributes in SQL, 192, 214–215
 - Repeating field, records, 561–563
 - Replication models, 886
 - Replication transparency
 - DDBs, 843
 - NOSQL, 885–886, 894
 - Representational (implementation) data models, 33
 - Resource Description Framework (RDF), 447
 - ResourceManager (RM), YARN, 941–942
 - RESTRICT option, SQL, 233, 234
 - Result equivalence, schedules, 765
 - ResultSet object JDBC, 334–335
 - Retrieval operations
 - files, 564–565
 - object information, 371
 - relational data models, 165–166
 - selection conditions, 564–565
 - Retrieval, 1027
 - RETURN clause, XQuery, 446
 - ROLAP (relational OLAP) function, 1114
 - Role-based access control (RBAC), 1121, 1137–1139
 - Role names, 75
 - Roles of domain attributes, 152
 - Root, tree structures, 617
 - Root element, XML, 440
 - Root tag, XML documents, 434
 - Rotational delay (latency), disk devices, 552
 - Round-robin partitioning, 684
 - Row, SQL, 179
 - Row-based constraints, SQL, 187
 - Row-level access control, 1139–1140
 - ROW TYPE command, 380
 - RSA public key encryption algorithm, 1152
 - Rules
 - active databases systems, 22
 - active rules, 962–964, 970–973
 - association rules, 1073–1084
 - axioms, 1005
 - deductive database systems, 22
 - deductive databases, 1000, 1005–1007
 - defined, 1000
 - force/no-force rules, 817–818
 - 4NF schema, 527–528
 - functional dependencies, 505–509, 527–528
 - inference rules, 505–509, 527–528
 - inferencing information using, 22
 - interpretation of, 1005–1007
 - models for, 1005–1006
 - model-theoretic interpretation of, 1005
 - proof-theoretic interpretation of, 1005
 - stored procedure for, 22
 - theorem proving, 1005
 - triggers as, 22
 - Runtime, MapReduce (MR), 927–930
 - Runtime database processor, 44, 655
 - Safe expressions, 276–277
 - Sampling algorithm, 1076–1077
 - Scalability
 - DDBs, 845
 - NOSQL, 885
 - Scale-invariant feature transform (SIFT), 998
 - Scanner, query processing, 655
 - Schedules (histories)
 - cascading rollback phenomenon, 762
 - committed projection of, 760
 - complete schedule conditions, 760
 - concurrency control and serializability, 770–771
 - conflict equivalence of, 765–766
 - conflicting operations in, 759–760
 - debt-credit transactions, 773
 - nonserial schedules, 763, 764–765
 - operation semantics for, 773
 - recoverability basis of, 761–762
 - recoverable/nonrecoverable schedule, 761
 - result equivalence of, 765
 - serial schedules, 763–764
 - serializability basis of, 763–766
 - serializable schedules, 763, 765–766
 - strict schedule, 762
 - testing for serializability, 767–770
 - transaction processing, 759–773
 - transactions for, 759–760
 - view equivalence, 771–772
 - view serializability, 771–772
- Schema-based (explicit) constraints, 157
 - Schema change statements
 - ALTER command, 233–234
 - DROP command, 233
 - schema evolution command use, 232–233
 - Schema diagram, 34–35
 - Schema matching, 1052
 - Schemaless documents, XML, 432–433
 - Schemas
 - authorization identifier, 179
 - bottom-up conceptual synthesis, 119
 - catalog collection of, 35, 38, 180
 - conceptual level, 37, 61–62
 - constraints and, 157–165
 - constructs, 35
 - data independence and, 37–38
 - database descriptions, 34
 - database state (snapshot) and, 35
 - database requirements, 122–124
 - descriptors, 179
 - design creation (conceptual) of, 61–62
 - EER modeling and, 119–120, 122–124
 - EER schema to ODB schema, 407–408
 - ER diagram notation for, 81, 83–85
 - ER modeling and, 61–62
 - evolution, 35
 - external level (views), 37
 - intention, 35
 - interface inheritance, ODL, 404–405
 - internal level, 36
 - mappings, 37, 407–408
 - meta-data storage of, 35
 - naming constructs, 82
 - ODB conceptual design and, 407–408
 - ODL, 400–403
 - refinement using generalization and specialization, 119–120
 - relation, 157–160, 163–165
 - relational database, 160–163
 - SQL concepts, 179–180
 - three-schema architecture, 36–38
 - top-down conceptual refinement, 119
 - XML language, 434, 436–441
 - Script functions, HTML, 428
 - Search, B-trees, 625–626
 - Search engines
 - desktop, 1025
 - Lucern, 1043–1044
 - Web search, 1047

- Search relevance, IR, 1044–1047
- Search techniques
 - conjunctive selection, 665–666
 - disjunctive selection, 666–667
 - keyword-based, 41
 - query processing, 663–667
 - SELECT operation algorithms, 663–667
 - simple selection, 663–665
 - Web database applications, 4
- Search trees, dynamic multilevel indexes, 618–619
- Second normal form (2NF)
 - definition of, 481
 - full functional dependency and, 481–482
 - general definition of, 484–486
 - normalizing relations, 482, 484–486
 - primary key and, 483–484
- Secondary access path, indexing, 601
- Secondary indexes, 603, 609–612
- Secondary storage
 - capacity of, 534
 - devices for, 547–556
 - random access devices, 554
 - sequential access devices, 554–555
 - solid-state drive (SSD), 542
- Security, *see* Data security; Database security
- Security and authorization subsystems, 19
- Seek time, disk devices, 552
- SELECT clause statement
 - ALL option with, 194–195
 - AS option with, 196
 - DISTINCT option with, 188, 194
 - mandatory use of, 197
 - multiset tables and, 194–195
 - SQL query retrieval and, 187–188, 194–197
- SELECT operation
 - Boolean expressions (clauses), 241–242
 - cascade (sequence) with, 243
 - conjunctive selection, 665–666
 - cost functions for, 714
 - degree of relations, 243
 - disjunctive selection, 666–667
 - estimating selectivity of conditions, 667–668
 - implementation options for, 663
 - query processing algorithms, 663–668
 - relational algebra using, 241–243
 - search methods for, 663–667
 - selectivity of a condition, 243, 667–668
 - simple selection, 663–665
- SELECT operator (σ), 241
- Select...from...where structure, OQL, 409
- Select-from-where block, SQL, 188–191
- Select-project-join query, 189–190, 273
- Selection conditions
 - domain variables, 278
 - file operations, 564–565
 - parallel algorithms, 685
 - WHERE clause queries, 189
- Selective inheritance, 377
- Selectivity
 - join operations, 254, 719–720
 - of a condition, 243, 667–668
- Self-describing data, 10, 427
- Self-describing data models, 34
- Self-describing documents, 425. *See also* JSON; XML (EXtended Markup Language)
- Semantic approach, IR, 1028
- Semantic data models
 - abstraction concepts, 129–133
 - EER modeling, 107–108
 - ontology for, 132–134
- Semantic equivalence, query trees, 694–695
- Semantic heterogeneity, 857–858
- Semantic model, IR, 1034–1035
- Semantic query optimization, 737–738
- Semantic tagging, images, 998–999
- Semantics
 - attribute clarity, 461–465
 - data constraints, 21
 - functional dependency of, 472–473
 - relation schema, 461–465
 - RDB design, 461–465, 472–473
 - schedule operations, 773
- Semi-join (SJ) operator, 658–660, 681, 719–720, 862–863
- Semistructured data, XML, 426–428
- Separator characters, records, 561
- Sequence of interaction, database programming and, 313–314
- Sequence of operations, relational algebra, 245–246
- Sequential access storage devices, 554–555
- Sequential pattern discovery, data mining, 1091
- Serial ATA (SATA), 551
- Serial schedules, 763–764
- Serializability
 - basis of schedules, 763–766
 - concurrency control and, 770–771
 - testing for, 767–770
- Serializable schedules, 763, 765–766
- Server, defined, 48
- Servers
 - application, 44
 - database, 44
 - DBMS module, 31
- SET clause, SQL, 201
- SET CONNECTION command, SQL, 316
- Set constructor, 369
- SET DIFFERENCE operation, 247–249
- Set operations
 - anti-join (AJ) operator for set difference, 677–678
 - parallel algorithms, 686
 - query processing, algorithms for, 676–678
 - SQL, 194–195
- Set theory
 - CARTESIAN PRODUCT operation, 249–251
 - INTERSECTION operation, 247–249
 - MINUS operation, 247–249
 - OUTER UNION operation, 264–265
 - relational algebra operations from, 246–251, 264–265
 - SET DIFFERENCE operation, 247–249
 - type compatibility, 247
 - UNION operation, 246–249
- Set type, legacy data modeling with, 53
- Set-at-a-time, file operations, 566
- Set-at-time DML, 40
- Sets
 - explicit set of values, 214
 - multiset comparisons, SQL query retrieval, 209–211
 - parentheses for, 214
 - SQL table relations, 188, 193–195
- Shadow directory, 826
- Shadow paging, database recovery, 826–827
- Shadowing, 816
- Sharding
 - DDBs, 847–848
 - NOSQL, 886, 894–895
- Shared-disk architecture, 683
- Shared/exclusive (read/write) locks, 784–786
- Shared-memory architecture, 683
- Shared-nothing architecture, 684
- Shared subclasses, 118, 301
- Shared variables in embedded SQL, 314
- Signature of operations, 366, 397. *See also* Interfaces
- Simple (atomic) attributes, 65–66
- Simple elements, XML, 431
- Simple Object Access Protocol (SOAP), 447
- Simple selection, search methods for, 663–665
- Single character replacement symbol ($_$), 195–196
- Single inheritance, 118–119
- Single-level ordered indexes
 - clustering indexes, 602, 606–608
 - concept of, 602–603
 - physical database design and, 602–613
 - primary indexes, 602, 603–606
 - secondary indexes, 603, 609–612
- Single-relation options, EER-to-relational mapping, 299–300
- Single-sided disks, 547
- Single time point, 976
- Single-user DBMS systems, 51
- Single-valued attribute, ER modeling, 66

- Small computer system interface (SCSI), 551
- Snapshot isolation
 - concurrency control and, 758, 781, 799–800
 - defined, 775
 - SQL transaction support and, 775–776
- Snapshot (database) state, 35
- Snowflake schema, 1108–1109
- Social search, IR, 1058–1059
- Software engineers, 16
- Solid-state device (SSD) storage, 553–555
- Solid-state drive (SSD), secondary storage of, 542
- Sophisticated end users, 16
- Sorting phase, external algorithms, 661
- Sort-merge join, 559, 719, 930
- Spanned versus unspanned records, 563–564
- Spatial analysis operations, 988
- Spatial colocation rules, 993–994
- Spatial databases
 - analytical operations, 988
 - applications of spatial data, 994
 - data mining, 993–994
 - data types, 989–990
 - enhanced data models, 962, 987–994
 - indexing, 991–993
 - models of information, 990
 - object storage by, 987–988
 - operators, 990–991
 - queries, 991
- Specialization
 - attribute-defined, 114
 - conceptual schema refinement, 119–120
 - constraints on, 113–116
 - defined, 110
 - design choices for, 124–128
 - disjointness (d notation), 114–115
 - EER diagram notation for, 109, 110
 - EER modeling concept, 108, 110–120, 124–128
 - EER-to-relational mapping options, 298–301
 - hierarchies, 116–119
 - instances of, 111–112
 - lattices, 116–119
 - partial, 115
 - semantic modeling process, 131
 - total, 115
 - UML notation for, 127–128
- Specialized servers, client/server architecture, 47
- Specification, ontology and, 134
- Speech input and output, 41
- Spurious tuple generation, RDB design and, 468–471
- SQL (Structured Query Language) system
 - active database techniques, 202
 - arithmetic operators, 196–197
 - assertions, 158, 156, 165, 225–226
 - attribute data types in, 182–184
 - catalog concepts, 179–180
 - CHECK clause, 187
 - comparison operators, 188–191, 195–197
 - complex queries, 207–225
 - constraints, 165, 184–187, 225–227
 - core specifications, 178
 - CREATE ASSERTION statement, 225–226
 - CREATE TABLE command, 180–182
 - CREATE TRIGGER statement, 225, 226–227
 - data definition, 179
 - DBMS use of, 177–178
 - DELETE command, 200
 - domains, 184
 - encapsulation of operations, 384–385
 - extensions, 178
 - function overloading, 385
 - granting and revoking privileges, 202
 - history of, 178
 - index creation, 201–202
 - inheritance, type specification of, 385
 - INSERT command, 198–200
 - logical comparison operators, 188–190
 - NOSQL database system and, 26
 - object identifiers, 383
 - object-relational systems, 202
 - ODB extensions to, 379–386
 - operators, query translation into, 657–660
 - practical relational model, 177–206
 - query processing, translation for, 657–660
 - query retrieval, 187–198, 207–225
 - reference types, 383
 - relational algebra, query translation into, 657–660
 - relational data models and, 51, 165
 - schema change statements, 232–234
 - schema concepts, 179–180
 - syntax of, 235
 - table creation, 383–384
 - transaction support, 773–776
 - triggers, 158, 165, 226–227
 - UPDATE command, 200–201
 - user-defined types (UDTs), 380–384
 - views (virtual tables), 228–232
 - XML data creation functions (XML/SQL), 453–455
- SQL injection
 - bind variables, 1145–1146
 - code injection, 1144
 - database security, 1143–1146
 - filtering input, 1146
 - function call injection, 1144–1145
 - function security for, 1146
 - manipulation, 1143–1144
 - protection against attacks, 1145–1146
 - risks associated with, 1145
- SQL plan management, Oracle, 736–737
- SQL programming
 - comparison of approaches, 338–339
 - database programming language approaches, 309–314, 339
 - database stored procedures, 335–338
 - dynamic SQL, 320–321
 - embedded SQL, 311, 314–320, 338–339
 - JDBC: SQL class library, 331–335
 - library of functions or classes for, 311–312, 326–335, 339
 - query specification and, 320–321
 - SQL/CLI (SQI call level interface), 326–331
 - SQLJ: Java commands, 321–325
- SQL server, two-tier client/server architecture, 49
- SQL/CLI (SQI call level interface)
 - connection record, 327–328
 - database programming with, 326–331
 - description record, 327–328
 - environment record, 327–328
 - handles for records, 328
 - statement record, 327–328
 - steps for programming, 328–331
- SQL/PSM (SQL/persistent stored modules), 337–338
- SQLCODE variable, 316
- SQLJ
 - embedding SQL commands in Java, 321–325
 - exceptions for error handling, 322–323
 - iterators for, 323–325
 - query result processing, 323–325
- SQLSTATE variable, 316
- Standalone users, 16
- Standards, enforcement of, 22
- Star schema, 1108
- STARBURST, statement-level rules in, 970–972
- Star-transformation optimization, 731–733
- Starvation, 792
- State constraints, 165
- State of an object or literal, 387
- Statement object JDBC, 335
- Statement parameter
 - binding, 329, 333
 - JDBC, 333
 - SQL/CLI, 329
- Statement record, SQL/CLI, 327–329
- Statement string, SQL/CLI, 329
- Statement-level rules, STARBURST, 970–972
- Statement-level trigger, 967
- Static files, 566
- Static hashing, 577
- Statistical approach, IR, 1028
- Statistical database security, 1146–1147
- Steal/no-steal rules, 817–818
- Stemming, IR text processing, 1038

- Stopword removal, IR text processing, 1037–1038
- Storage
 - architectures for, 588–592
 - automated storage tiering (AST), 591
 - big data, 3
 - buffering blocks, 541, 556–560
 - capacity, 543
 - cloud, 3
 - column-based, indexing for, 642
 - database catalog for, 10–11
 - database organization of, 545–546
 - database reorganization, 45
 - devices for, 543–545, 547–556
 - Fibre Channel over Ethernet (FCoE), 590–591
 - Fibre Channel over IP (FCIP), 590
 - file records, 560–564, 567–572, 582–583
 - files, 10–11, 560–572, 582–583
 - hashing techniques, 572–582
 - Internet SCSI (iSCSI), 590
 - memory hierarchies, 543–545
 - meta-data, 6, 10
 - network-attached storage (NAS), 589–590
 - object-based, 591–592
 - objects, 987–988
 - persistent, 19–20, 545
 - primary, 542, 543
 - program objects, 19–20
 - RAID technology, 542, 584–588
 - secondary, 542, 543, 547–556
 - spatial databases for, 987–988
 - storage area networks (SANs), 588–589
 - tertiary, 542, 543
 - XML documents, 442–443
- Storage area networks (SANs), 588–589
- Storage definition language (SDL), 39
- Storage devices
 - databases, organization and, 545–546
 - disks, 547–553
 - flash memory, 543–544
 - magnetic tape, 544–545, 555–556
 - memory, 543–545, 547–556
 - optical drives, 544
 - secondary, 547–556
 - solid-state device (SSD), 553–555
- Stored attribute, 66
- Stored data manager, 42, 44
- Stored procedures
 - CALL statement, 337
 - database programming and, 335–338
 - parameter type and mode, 336–337
 - persistent storage modules, 336
 - rule enforcement using, 22
 - SQL/PSM (SQL/persistent stored modules), 337–338
- Stream-based processing, 682. *See also* Pipelining
- Strict schedule, 762
- Strings. *See also* Text strings
 - character data types, 182–183
 - double quotations (“ ”) for, 196, 347
 - indexing, 640
 - prefix compression, 640
 - single quotations (‘ ’) for, 182, 196, 347
 - SQL use of, 182–183, 195–197
 - substring pattern matching, 195–197
- Strong entity types, 79
- Struct (tuple) constructor, 368, 369
- Structural constraints, 78
- Structured data, XML, 426
- Structured data extraction, WEB, 1052
- Structured objects and literals, 388, 396
- Structured Query Language, *see* SQL (Structured Query Language)
- Subclasses
 - class relationships, 108–110
 - defined, 126
 - defining predicate of, 113–114
 - EER diagram notation for, 109
 - EER modeling concept, 108–110, 126
 - EER-to-relational mapping, 301
 - entity type as, 110
 - inheritance, 110, 117–119, 301
 - IS-A relationship, 109, 126
 - leaf class (UML node), 127
 - local attributes of, 110–111
 - overlapping entities, 115
 - predicate-defined, 113–114
 - shared, 118, 301
 - specialization of set of, 110–112
 - specific relationship types, 110–111
 - union type, 108, 120–122
 - user-defined, 114
- Subqueries
 - nested, 702–704
 - query optimization and, 702–706
 - unnesting (decorrelation), 704
 - view merging transformation, 704–706
- Substring pattern matching, SQL, 195–197
- Subtrees, 617
- Subtypes, 375–376
- SUM function
 - grouping, 260
 - SQL, 217
- Superclasses
 - base class (UML root), 127
 - categories of, 120–122
 - class relationships, 109
 - EER modeling concept, 109, 110, 126
 - entity type as, 110
 - inheritance, 110, 117–118
 - subclass relationships, 110, 117–118
- Superkey, 158–159, 476–477
- Supertypes, 375–376
- Surrogate key, 302
- Symmetric key algorithms, 1150–1151
- Synthesis, RDB design by, 503, 504
- System analysts, 16
- System designers and implementers, 17
- System log
 - database recovery, 814, 817, 818–819
 - modifications for database security, 1125
 - transaction processing, 755–756
- Table inheritance, SQL, 385
- Table of values, 150–151
- Table-based constraints, SQL, 184–187
- Tables
 - ALTER TABLE command, 180
 - base relations, 180, 182
 - CREATE TABLE command, 180–182
 - data definition statements, 180–182
 - database recovery, 828–831
 - inner join, 215–216
 - joined relations, 215–216
 - multiset operations, 193–195
 - multiway join, 216
 - NATURAL JOIN operation, 215
 - OUTER JOIN operations, 216
 - query retrieval and, 193–195
 - query retrieval and, 193–195, 215–216
 - sets of relations in, 188, 193–195
 - transaction, 828–831
 - trigger activation from, 22
 - UDT creation of for SQL, 383–384
 - views (virtual tables), 228–232
 - virtual relations, 82
- Tags
 - attributes, 430
 - document body specification, 429
 - document header specifications, 428
 - end/start tag (</...>), 428
 - HTML tag (<...>), 428
 - mark up of documents using, 428–429
 - notation and use, HTML, 428–430
 - semantic tagging of images, 998–999
 - XML unstructured data and, 428–430
- Tape jukeboxes, 544
- Taxonomy, ontology as, 134
- Temporal databases
 - applications of, 974
 - calendar, 975
 - enhanced data models, 962, 974–987
 - implementation considerations, 982
 - incorporating time, 977–984
 - object-oriented databases for, 982–984
 - relational databases for, 977–982
 - time representation, 975–977
 - versioning, 977–984
- Temporal querying constructs, 984–986
- Temporary update problem, transaction processing, 750
- Ternary relationships
 - binary relationships compared to, 88–89
 - degree of, 73–74
 - ER diagrams, 88–92
 - notation for diagrams, 88–89
- Tertiary storage, 542, 543

- Testing for serializability, 767–770
- Text/document source, multimedia databases, 996
- Text preprocessing
 - information extraction (IE), 1040
 - information retrieval (IR), 1037–1040
 - stemming, 1038
 - stopword removal, 1037–1038
 - thesaurus use, 1038–1039
- Text strings
 - double-quoted, 347–348
 - interpolating variables within, 347
 - length of, 346
 - PHP programming, 346, 347–348
 - single-quoted, 347–348
- Thematic search, 989
- Theorem proving, 1005
- Thesaurus
 - IR text processing, 1038–1039
 - ontology as, 134
- THETA JOIN condition, 252
- Third normal form (3NF)
 - algorithm for RDB schema design, 519–522
 - definition of, 483
 - dependency preservation and, 519–522
 - general definition of, 486–487
 - nonadditive (lossless) join decomposition and, 519–522
 - normalizing relations, 485, 486–487
 - primary key and, 483–484
 - transitive dependency and, 483
- Thomas's write rule, 795
- Three-schema architecture, 36–38
- Three-tier/client-server architecture
 - discrete databases (DDBs), 872–875
 - Web applications, 49–51
- Three-valued logic for SQL NULL
 - comparisons, 208–209
- Thrown exceptions, SQLJ, 322–323
- TIME data type, 183
- Time period, 976
- Time reduction, development of, 22–23
- Time representation, temporal databases, 975–977
- Time series data, 986–987
- Time series management systems, 987
- Timeouts, deadlock prevention, 792
- TIMESTAMP data type, 183–184
- Timestamp ordering (TO)
 - algorithm, 793
 - basic, 794
 - concurrency control based on, 792–795
 - multiversion technique based on, 796
 - strict, 794–795
 - Thomas's write rule for, 795
- Timestamps
 - concurrency control and, 781, 790–791, 793
 - deadlock prevention using, 790–791, 793
 - generation of, 793
 - transaction timestamps, 790–791
- Tool developers, 17
- Tools, DBMS, 45–46
- Top-down conceptual refinement, 119
- Top-down method, RDB design, 460
- Top-*k* results, query optimization, 730
- Topological relationships, 989
- Total categories, 122
- Total specialization, 115, 126
- Transaction management, DDBs, 857–859
- Transaction processing
 - commit point, 756
 - concurrency control, 749–752
 - concurrency of, 746–747
 - data buffers, 748–749
 - database items, 748
 - DBMS-specific buffer replacement policies, 756–757
 - read/write transactions, 748
 - recovery for, 752–753
 - schedules (histories), 759–773
 - single-user versus multiuser systems, 746–747
 - SQL transaction support, 773–776
 - system log, 755–756
 - systems, 745
 - transaction failures, 752–753
 - transaction states, 753–754
 - transactions for, 747–749, 757–758
- Transaction rollback, database recovery, 819
- Transaction server, two-tier client/server architecture, 49
- Transaction tables, database recovery, 828–831
- Transaction time dimensions, 976–977
- Transaction time relations, 979–980
- Transaction timestamps, deadlock prevention, 790–791
- Transaction-id, 755
- Transactions
 - atomicity property, 14, 757
 - certification of, 781
 - concurrency control and, 781, 798–799, 807
 - consistency preservation, 757
 - database recovery, 821
 - debt-credit, 773
 - defined, 6, 169
 - desirable properties of, 757–758
 - durability (permanency) property, 758
 - interactive, 807
 - isolation property, 14, 758
 - multiuser processing, 13–14
 - not affecting database, 821
 - OTLP systems, 14, 52, 169
 - relational data modeling, 169
 - user-defined functional requirements, 61
 - validation (optimistic) of, 781, 798–799
- Transient data, storage of, 545
- Transient objects, 365, 373
- Transition constraints, 165
- Transitive dependency, 3NF, 483
- Transparency, DDBs, 843–844
- Tree search data structures, *see* B-trees; B⁺-trees
- Tree-structured data models
 - attributes, 433
 - breaking graph cycles for conversion to, 452–453
 - data-centric documents, 431
 - data mining, 1077–1080, 1085–1086
 - decision trees, 1085–1086
 - document-centric documents, 431
 - document extraction using, 447–453
 - elements, 430–431
 - frequent-pattern (FP) tree, 1077–1080
 - graph conversion into, 452–453
 - hierarchies for, 116, 452–453
 - hybrid documents, 431
 - schemaless documents, 432–433
 - XML, 51, 430–433, 447–453
- Triggers
 - active databases, 963–967, 973–974
 - database tables and, 22
 - CREATE TRIGGER statement, 225, 226–227
 - database monitoring, 226–227
 - event-condition-action (ECA) components, 227, 963–964
 - Oracle notation for, 965–967
 - SQL, 158, 165, 226–227
 - SQL-99 standards for, 973–974
- Trivial/nontrivial MVD, 493
- Truth value of atoms, 270, 277
- TSQL2 language, 984–986
- Tuning indexes, 640–641
- Tuple relational calculus
 - expressions, 270–271, 276–277
 - formulas (conditions), 270–271
 - nonprocedural language of, 268
 - quantifiers, 271, 274–276
 - queries using, 272–276
 - query graphs, 273–274
 - range relations, 269–270
 - requested attributes, 269
 - safe expressions, 276–277
 - selected combinations for, 269
 - variables, 269–270
- Tuple variables
 - alias of attributes, 192
 - bound, 271
 - free, 271
 - iterators, 189
 - range relations and, 269–270
- Tuples
 - alternative definition of a relation and, 154–155
 - anomalies and, 465–467
 - asterisk (*) for rows in query results, 218

- atomic value of, 155
- attribute ambiguity and, 191–192
- CHECK clause for, 187
- CROSS PRODUCT operation for
 - combinations, 192–193
- dangling tuple problems, 523–524
- delete operation for, 166, 167–168
- embedded SQL retrieval of, 311, 314–317
- grouping and, 219
- mapping relations with, 154
- matching, 264–265
- multisets of, 193–195
- nested query values, 209–211
- n*-tuple for relations, 152
- NULL value of, 155–156, 163, 467–468
- ordering of, 154–155
- OUTER UNION operation and, 264–265
- parentheses for comparisons, 210
- partially compatible relations, 264
- partitioning relations into, 219
- precompiler or preprocessor for
 - retrieval of, 311, 314
- query retrieval and, 191–195, 209–211
- RDB design problems, 523–524
- redundant information in, 465–467
- referential integrity of, 163
- relation schema for RDB design, 465–471
- relation state values, 152–156
- row-based constraints, 187
- separate groups for NULL grouping
 - attributes, 219
- set of, 154–155
- spurious tuple generation, 468–471
- SQL tables and, 187, 191–195
- type (union) compatibility, 247
- update (modify) operation for, 166, 168–169
- versioning, 977–982
- Two-phase locking (2PL)
 - basic 2PL, 788
 - concurrency control, 782–792, 796–797
 - conservative 2RL, 788
 - deadlock, 789–792
 - expanding (first) phase, 786
 - locks for, 782–786
 - multiversion concurrency control and, 796–797
 - protocol, 786–788
 - rigorous 2PL, 789
 - shrinking (second) phase, 786
 - starvation, 792
 - strict 2PL, 788–789
 - subsystem for, 789
- Two-tier client/server architecture, 49
- Two-way join, 668
- Type (class) hierarchies
 - constraints on extents corresponding to, 376–377
 - functions in, 374–375
 - inheritance, 385
 - ODBs, 366, 374–377
 - subtype/supertype relationships, 375–376
 - visible/hidden attributes, 371, 375
- Type (union) compatibility, 247
- Type constructors
 - array, 369
 - atom, 368, 369
 - bag, 369
 - collection (multivalued), 369
 - dictionary, 369
 - list, 369
 - object definition language (ODL)
 - and, 369
 - object operation, 371
 - ODB objects and literals, 368–370
 - references to object type relationships, 369–370
 - set, 369
 - SQL, 379
 - struct (tuple), 368, 369
 - type structures and, 368–370
- Type generators
 - ODB objects and literals, 368–369
 - ODMG models, 394–395
- Type inheritance, 385
- Type structures, 368–370. *See also* Type constructors
- UDTs (User-defined types)
 - arrays, 383
 - built-in functions for, 384
 - CARDINALITY function, 383
 - CREATE TYPE command, 380–383
 - dot notation for, 383
 - encapsulation of operations, 384–385
 - inheritance specification (NOT FINAL), 385
 - SQL, 380–385
 - table creation based on, 383–384
- UML (Unified Modeling Language)
 - aggregation, 87–88
 - associations, 87–88
 - base class, 127
 - bidirectional associations, 87
 - class diagrams, 85–88, 127–128
 - EER models and, 127–128
 - ER models and, 60, 85–88
 - leaf class, 127
 - links, 87
 - qualified association, 88
 - reflexive association, 87
 - unidirectional association, 87
- Unary operations
 - assignment operations (\leftarrow) for, 245
 - Boolean expressions (clauses), 241–242
 - cascade (sequence) with, 243
 - defined, 243
 - degree of relations, 243, 244
 - duplicate elimination and, 244–245
 - PROJECT operation, 243–245
 - relational algebra and, 240, 241–246
 - renaming attributes, 245–246
 - SELECT operation, 241–243
 - selectivity of condition, 243
 - sequence of operations for, 245–246
- Unauthorized access restriction, 19
- UNDO/REDO algorithm, 815, 818
- Unidirectional association, UML class diagrams, 87
- Unified Modeling Language, *see* UML (Unified Modeling Language)
- UNION operations
 - matching tuples, 264–265
 - OUTER UNION operation, 264–265
 - partially compatible relations, 264
 - relational algebra, 264–265
 - SQL sets, 194–195
- Union types
 - categories of, 120–122, 302–303
 - EER diagram notation for, 120
 - EER modeling concept, 108, 120–122
 - EER-to-relational mapping, 302–303
 - set union operation (\cup), 120
 - surrogate key for, 302
- UNIQUE function, SQL query retrieval, 212–214
- Unique keys, 160
- Uniqueness constraints
 - ER model entity types, 68–68
 - key attributes as, 68–69
 - key constraints with, 158–160
 - relation schema and, 158–160
- Universal quantifiers, 271, 274–276
- Universal relation assumption, 513
- Universal schema relations, 471–474, 504, 513
- Universe of Discourse (UoD), 5
- Unnest relation, 1NF, 479–480
- Unordered file records (heaps), 567–568
- Unrepeatable read problem, transaction processing, 752
- Unstructured data, XML, 428–430
- Unstructured information, 1022
- Unstructured/semistructured data
 - handling, Big data technology and, 945
- Update (modify) operations
 - relational data models, 166, 168–169
 - files, 564–565
 - relational data models, 166, 168–169
 - selection conditions for, 564–565
 - tuple modification using, 166, 168–169
- Update anomalies, RDB design and, 465–467
- UPDATE command, SQL, 200–201
- Update decomposition, DDBMS, 863–865
- Update strategies for SQL views, 230–232
- Upgrading locks, 786

- User views, 37
- User-defined subclass, 114, 126
- User-defined types, *see* UDTs (User-defined types)
- Utilities, DBMS functions, 45
- Valid documents, XML, 434
- Valid state, databases, 35, 160–161
- Valid time, temporal databases, 976
- Valid time relations, temporal databases, 977–979
- Validation (optimistic) concurrency control, 781, 798–799
- Value (state) of an object or literal, 387
- Value sets (domains) of attributes, 69–70
- Variable-length records, 561–563
- Variables
 - built-in, 352–353
 - communication, 316
 - domain, 277
 - embedded SQL, 314–316
 - iterator, OQL, 409–410
 - interpolating within text strings, 347
 - names for, 346, 347
 - PHP, 345–347, 352–353
 - predefined, 345–346
 - program, 314–315
 - shared, 314
 - tuple, 189, 192, 169–170
- Vector space model, IR, 1031–1033
- Versioning
 - attribute approach, 982–984
 - NOSQL, 887, 899, 900–902
 - object-oriented databases
 - incorporating time, 982–984
 - relational databases incorporating time, 977–982
 - tuple approach, 977–982
- Vertical fragmentation, DDBs, 844, 848–849
- Video source, multimedia databases, 996
- View definition language, 39
- View merging transformation, subqueries, 704–706
- Views
 - database designer development of, 15
 - equivalence, schedules, 771–772
 - serializability, schedules, 771–772
 - support of multiple data, 13
- Views (virtual tables)
 - authorization using, 232
 - base tables compared to, 228
 - CREATE VIEW statement, 228–229
 - data warehouses compared to, 1115
 - defining tables of, 228
 - hierarchical, 447–452
 - in-line, 232
 - DROP VIEW command, 229
 - materialization, 230
 - query modification for, 229–230
 - query retrieval using, 230–231
 - SQL virtual tables, 228–232
 - update strategies for, 230–232
 - virtual data in, 13
 - WITH CHECK option for, 232
 - XML document extraction and, 447–452
- Virtual data, 13
- Virtual private database (VPD)
 - technology, 1156
- Virtual relations (tables), 82
- Virtual storage access method (VSAM), 541
- Virtual tables, 228–232. *See also* Views (virtual tables)
- Visible attributes, objects, 371, 375
- Volatile/nonvolatile storage, 545
- Voldemort key-value data store, 897–899
- Weak entity types, 79, 292–293
- Web analytics, 1057
- Web-based user interfaces, 40
- Web crawlers, 1057
- Web database programming
 - HTML and, 343–346
 - Java technologies for, 358
 - PHP for, 343–359
- Web database systems
 - access control policies, 1141–1142
 - data interchanging using XML, 25
 - HTML and, 25
 - menu-based interfaces, 40
 - n*-tier architecture for, 49–51
 - security, 1141–1142
 - three-tier architecture for, 49–51
- Web information integration, 1052
- Web pages
 - hypertext documents for, 425
 - segmentation and noise reduction, 1053
 - XML and formatting of, 425–426
- Web search
 - defined, 1028
 - digital libraries for, 1047–1048
 - HITS ranking algorithm, 1051
 - link structure analysis, 1050–1051
 - PageRank ranking algorithm, 1051
 - search engines for, 1047
 - Web analysis and, 1048–1049
 - Web context analysis, 1051–1054
 - Web structure analysis, 1049–1050
 - Web usage analysis, 1054–1057
- Web servers
 - client/server architecture, 47
 - three-tier architecture, 50
- Web Services Description Language (WSDL), 447
- Web spamming, 1057
- Well-formed documents, XML, 433–424
- WHERE clause
 - asterisk (*) for all attributes, 193
 - explicit set of values in, 214–215
 - grouping and, 221–222
 - SQL query retrieval and, 188–189, 192–193, 197, 214–215
 - selection (Boolean) condition of, 189
 - unspecified, 192–193
 - XQuery, 446
- WHERE CURRENT OF clause, SQL, 318
- Wide area network, 842
- Wildcard (*) queries, 1036–1037
- WITH CHECK option, SQL views, 232
- WITH clause, SQL, 222–223
- Wrapper, 1025
- Write-ahead logging (WAL), database recovery, 816–818
- XML (EXtended Markup Language)
 - access control, 1140–1141
 - data models, 34, 51, 53
 - database extraction of documents, 442–443, 447–453
 - document type definition (DTD), 434–436
 - documents, 433–436, 442–443, 447–453
 - hierarchical (tree) data models, 51, 430–433, 447–453
 - hypertext documents and, 425
 - OpenPGP (Pretty Good Privacy) protocol, 1140–1141
 - protocols for, 446–447
 - query languages, 443–447
 - relational data model for document extraction, 447–449
 - schema language, 434, 436–441
 - semistructured data, 426–428
 - SQL functions for creation of data, 453–455
 - structured data, 426
 - tag notation and use, HTML, 428–430
 - unstructured data, 428–430
 - Web data interchanging using, 25
 - Web page formatting by, 425–426
 - XPath for path expressions, 443–445
 - XQuery, 445–446
 - XPath, XML path expressions, 443–445
 - XQuery, XML query specifications, 445–446
- YARN (Hadoop v2)
 - architecture, 940–942
 - Big data technology for, 936–944, 949–953
 - frameworks on, 943–944
 - rational behind development of, 937–939