FUNDAMENTALS OF

Database Systems

SIXTH 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

Addison-Wesley
Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editor in Chief: *Michael Hirsch*
Acquisitions Editor: *Matt Goldstein*
Editorial Assistant: *Chelsea Bell*
Managing Editor: *Jeffrey Holcomb*
Senior Production Project Manager: *Marilyn Lloyd*
Media Producer: *Katelyn Boller*
Director of Marketing: *Margaret Waples*
Marketing Coordinator: *Kathryn Ferranti*
Senior Manufacturing Buyer: *Alan Fischer*
Senior Media Buyer: *Ginny Michaud*
Text Designer: *Sandra Rigney and Gillian Hall*
Cover Designer: *Elena Sidorova*
Cover Image: *Lou Gibbs/Getty Images*
Full Service Vendor: *Gillian Hall, The Aardvark Group*
Copyeditor: *Rebecca Greenberg*
Proofreader: *Holly McLean-Aldis*
Indexer: *Jack Lewis*
Printer/Binder: *Courier, Westford*
Cover Printer: *Lehigh-Phoenix Color/Hagerstown*

Credits and acknowledgments borrowed from other sources and reproduced with permission in this textbook appear on appropriate page within text.

The interior of this book was set in Minion and Akzidenz Grotesk.

Library of Congress Cataloging-in-Publication Data

Elmasri, Ramez.
Fundamentals of database systems / Ramez Elmasri, Shamkant B. Navathe.—6th ed.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-136-08620-8
1. Database management. I. Navathe, Sham. II. Title.

QA76.9.D3E57 2010
005.74—dc22

Addison-Wesley is an imprint of

PEARSON

10 9 8 7 6 5 4 3 2 1—CW—14 13 12 11 10
ISBN 10: 0-136-08620-9
ISBN 13: 978-0-136-08620-8

To Katrina, Thomas, and Dora
(and also to Ficky)

R. E.

To my wife Aruna, mother Vijaya,
and to my entire family
for their love and support

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 sixth edition:

- A reorganization of the chapter ordering to allow instructors to start with projects and laboratory exercises very early in the course
- The material on SQL, the relational database standard, has been moved early in the book to Chapters 4 and 5 to allow instructors to focus on this important topic at the beginning of a course
- The material on object-relational and object-oriented databases has been updated to conform to the latest SQL and ODMG standards, and consolidated into a single chapter (Chapter 11)
- The presentation of XML has been expanded and updated, and moved earlier in the book to Chapter 12
- The chapters on normalization theory have been reorganized so that the first chapter (Chapter 15) focuses on intuitive normalization concepts, while the second chapter (Chapter 16) focuses on the formal theories and normalization algorithms
- The presentation of database security threats has been updated with a discussion on SQL injection attacks and prevention techniques in Chapter 24, and an overview of label-based security with examples

- Our presentation on spatial databases and multimedia databases has been expanded and updated in Chapter 26
- A new Chapter 27 on information retrieval techniques has been added, which discusses models and techniques for retrieval, querying, browsing, and indexing of information from Web documents; we present the typical processing steps in an information retrieval system, the evaluation metrics, and how information retrieval techniques are related to databases and to Web search

The following are key features of the book:

- A self-contained, flexible organization that can be tailored to individual needs
- A Companion Website (http://www.aw.com/elmasri) includes data to be loaded into various types of relational databases for more realistic student laboratory exercises
- A simple relational algebra and calculus interpreter
- 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 of the Sixth Edition

There are significant organizational changes in the sixth edition, as well as improvement to the individual chapters. The book is now divided into eleven parts as follows:

- Part 1 (Chapters 1 and 2) includes the introductory chapters
- The presentation on relational databases and SQL has been moved to Part 2 (Chapters 3 through 6) of the book; Chapter 3 presents the formal relational model and relational database constraints; the material on SQL (Chapters 4 and 5) is now presented before our presentation on relational algebra and calculus in Chapter 6 to allow instructors to start SQL projects early in a course if they wish (this reordering is also based on a study that suggests students master SQL better when it is taught before the formal relational languages)
- The presentation on entity-relationship modeling and database design is now in Part 3 (Chapters 7 through 10), but it can still be covered before Part 2 if the focus of a course is on database design
- Part 4 covers the updated material on object-relational and object-oriented databases (Chapter 11) and XML (Chapter 12)
- Part 5 includes the chapters on database programming techniques (Chapter 13) and Web database programming using PHP (Chapter 14, which was moved earlier in the book)
- Part 6 (Chapters 15 and 16) are the normalization and design theory chapters (we moved all the formal aspects of normalization algorithms to Chapter 16)

- Part 7 (Chapters 17 and 18) contains the chapters on file organizations, indexing, and hashing
- Part 8 includes the chapters on query processing and optimization techniques (Chapter 19) and database tuning (Chapter 20)
- Part 9 includes Chapter 21 on transaction processing concepts; Chapter 22 on concurrency control; and Chapter 23 on database recovery from failures
- Part 10 on additional database topics includes Chapter 24 on database security and Chapter 25 on distributed databases
- Part 11 on advanced database models and applications includes Chapter 26 on advanced data models (active, temporal, spatial, multimedia, and deductive databases); the new Chapter 27 on information retrieval and Web search; and the chapters on data mining (Chapter 28) and data warehousing (Chapter 29)

Contents of the Sixth Edition

Part 1 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.

Part 2 describes the relational data model, the SQL standard, and the formal relational languages. Chapter 3 describes the basic relational model, its integrity constraints, and update operations. Chapter 4 describes some of the basic parts of the SQL standard for relational databases, including data definition, data modification operations, and simple SQL queries. Chapter 5 presents more complex SQL queries, as well as the SQL concepts of triggers, assertions, views, and schema modification. Chapter 6 describes the operations of the relational algebra and introduces the relational calculus.

Part 3 covers several topics related to conceptual database modeling and database design. In Chapter 7, the concepts of the Entity-Relationship (ER) model and ER diagrams are presented and used to illustrate conceptual database design. Chapter 8 focuses on data abstraction and semantic data modeling concepts and shows how the ER model can be extended to incorporate these ideas, leading to the enhanced-ER (EER) data model and EER diagrams. The concepts presented in Chapter 8 include subclasses, specialization, generalization, and union types (categories). The notation for the class diagrams of UML is also introduced in Chapters 7 and 8. Chapter 9 discusses relational database design using ER- and EER-to-relational mapping. We end Part 3 with Chapter 10, which presents an overview of the different phases of the database design process in enterprises for medium-sized and large database applications.

Part 4 covers the object-oriented, object-relational, and XML data models, and their affiliated languages and standards. Chapter 11 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 12 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.

Part 5 is on database programming techniques. Chapter 13 covers SQL programming topics, such as embedded SQL, dynamic SQL, ODBC, SQLJ, JDBC, and SQL/CLI. Chapter 14 introduces Web database programming, using the PHP scripting language in our examples.

Part 6 covers normalization theory. Chapters 15 and 16 cover the formalisms, theories, and algorithms developed for relational database design by normalization. This material includes functional and other types of dependencies and normal forms of relations. Step-by-step intuitive normalization is presented in Chapter 15, which also defines multivalued and join dependencies. Relational design algorithms based on normalization, along with the theoretical materials that the algorithms are based on, are presented in Chapter 16.

Part 7 describes the physical file structures and access methods used in database systems. Chapter 17 describes primary methods of organizing files of records on disk, including static and dynamic hashing. Chapter 18 describes indexing techniques for files, including B-tree and B^+-tree data structures and grid files.

Part 8 focuses on query processing and database performance tuning. Chapter 19 introduces the basics of query processing and optimization, and Chapter 20 discusses physical database design and tuning.

Part 9 discusses transaction processing, concurrency control, and recovery techniques, including discussions of how these concepts are realized in SQL. Chapter 21 introduces the techniques needed for transaction processing systems, and defines the concepts of recoverability and serializability of schedules. Chapter 22 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. Finally, Chapter 23 focuses on database recovery protocols, and gives an overview of the concepts and techniques that are used in recovery.

Parts 10 and 11 cover a number of advanced topics. Chapter 24 gives an overview of database security including the discretionary access control model with SQL commands to GRANT and REVOKE privileges, the mandatory access control model with user categories and polyinstantiation, a discussion of data privacy and its relationship to security, and an overview of SQL injection attacks. Chapter 25 gives an introduction to distributed databases and discusses the three-tier client/server architecture. Chapter 26 introduces several enhanced database models for advanced applications. These include active databases and triggers, as well as temporal, spatial, multimedia, and deductive databases. Chapter 27 is a new chapter on information retrieval techniques, and how they are related to database systems and to Web search methods. Chapter 28 on data mining gives an overview of the process of data mining and knowledge discovery, discusses algorithms for association rule mining, classification, and clustering, and briefly covers other approaches and commercial tools. Chapter 29 introduces data warehousing and OLAP concepts.

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. Appendixes D and E (available on the book's Companion Website located at http://www.aw.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. We consider it important to expose database management students to these legacy approaches so they can gain a better insight of how database technology has progressed.

Guidelines for Using This Book

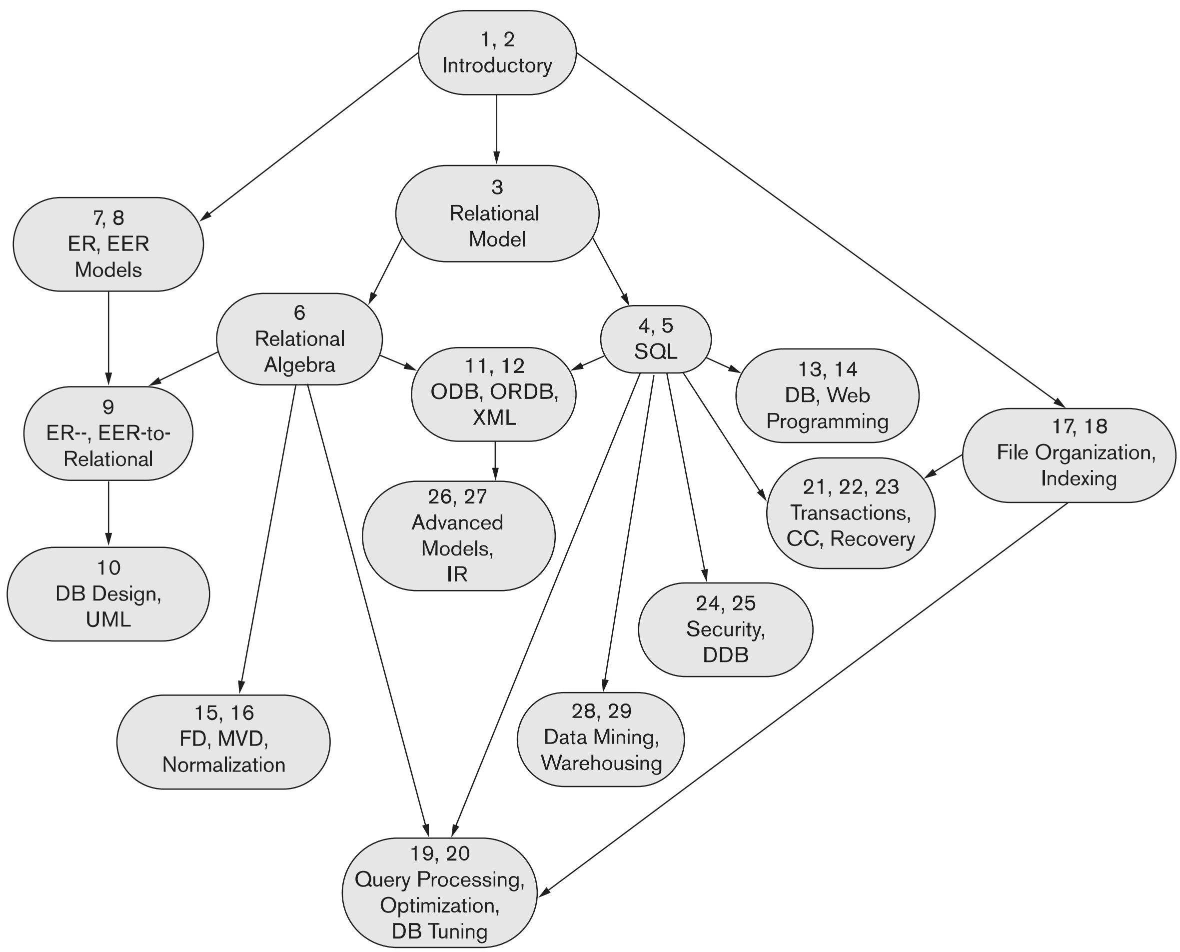
There are many different ways to teach a database course. The chapters in Parts 1 through 7 can be used in an introductory course on database systems in the order that they are given or in the preferred order of individual instructors. Selected chapters and sections may be left out, and the instructor 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 many 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 7 and 8, 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 17 and 18 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 7 and 8 early on, followed by the presentation of relational databases. A total life-cycle database design and implementation project would cover conceptual design (Chapters 7 and 8), relational databases (Chapters 3, 4, and 5), data model mapping (Chapter 9), normalization (Chapter 15), and application programs implementation with SQL (Chapter 13). Chapter 14 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 chapter dependency chart below 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 29. The two-semester sequence can also been designed in various other ways, depending on the preferences of the instructors.



Supplemental Materials

Support material is available to all users of this book and additional material is available to qualified instructors.

- PowerPoint lecture notes and figures are available at the Computer Science support Website at http://www.aw.com/cssupport.
- A lab manual for the sixth edition is available through the Companion Website (http://www.aw.com/elmasri). The lab manual contains coverage of popular data modeling tools, a relational algebra and calculus interpreter, and examples from the book implemented using two widely available database management systems. Select end-of-chapter laboratory problems in the book are correlated to the lab manual.
- A solutions manual is available to qualified instructors. Visit Addison-Wesley's instructor resource center (http://www.aw.com/irc), contact your local Addison-Wesley sales representative, or e-mail computing@aw.com for information about how to access the solutions.

Additional Support Material

Gradiance, an online homework and tutorial system that provides additional practice and tests comprehension of important concepts, is available to U.S. adopters of this book. For more information, please e-mail computing@aw.com or contact your local Pearson representative.

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 Gillian Hall for production management and Rebecca Greenberg for a thorough copy editing of the book. We thank the following persons from Pearson who have contributed to the sixth edition: Jeff Holcomb, Marilyn Lloyd, Margaret Waples, and Chelsea Bell.

Sham Navathe would like to acknowledge the significant contribution of Saurav Sahay to Chapter 27. Several current and former students also contributed to various chapters in this edition: Rafi Ahmed, Liora Sahar, Fariborz Farahmand, Nalini Polavarapu, and Wanxia Xie (former students); and Bharath Rengarajan, Narsi Srinivasan, Parimala R. Pranesh, Neha Deodhar, Balaji Palanisamy and Hariprasad Kumar (current students). Discussions with his colleagues Ed Omiecinski and Leo Mark at Georgia Tech and Venu Dasigi at SPSU, Atlanta have also contributed to the revision of the material.

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).

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

R. E.

S.B.N.

Contents

part 2
The Relational Data Model and SQL

chapter 6 The Relational Algebra and Relational Calculus 145

part 3
Conceptual Modeling and Database Design

chapter 7 Data Modeling Using the Entity-Relationship (ER) Model 199

part 5
Database Programming Techniques

part 6
Database Design Theory and Normalization

This page intentionally left blank

part 1

Introduction to Databases

This page intentionally left blank

chapter 1

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. New media technology 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 (GIS)** 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 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, 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. You may have recorded this data in an indexed address book or you may have stored it 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 constitute a database. However, the common use of the term *database* is usually more restricted. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the **miniworld** or the **universe of discourse (UoD)**. Changes to the miniworld are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively inter-

[1]We 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.

ested in its contents. The end users of a database may perform business transactions (for example, a customer buys a camera) or events may happen (for example, an employee has a baby) that cause the information in the database to change. In order for a database to be accurate and reliable at all times, it must be a true reflection of the miniworld that it represents; therefore, changes must be reflected in the database as soon as possible.

A database can be of any size and complexity. For example, the list of names and addresses referred to earlier may consist of only a few hundred records, each with a simple structure. On the other hand, the computerized catalog of a large library may contain half a million entries organized under different categories—by primary author's last name, by subject, by book title—with each category organized alphabetically. A database of even greater size and complexity is maintained by the Internal Revenue Service (IRS) to monitor tax forms filed by U.S. taxpayers. If we assume that there are 100 million taxpayers and each taxpayer files an average of five forms with approximately 400 characters of information per form, we would have a database of $100 \times 10^6 \times 400 \times 5$ characters (bytes) of information. If the IRS keeps the past three returns of each taxpayer in addition to the current return, we would have a database of 8×10^{11} bytes (800 gigabytes). This huge amount of information must be organized and managed so that users can search for, retrieve, and update the data as needed.

An example of a large commercial database is Amazon.com. It contains data for over 20 million books, CDs, videos, DVDs, games, electronics, apparel, and other items. The database occupies over 2 terabytes (a terabyte is 10^{12} bytes worth of storage) and is stored on 200 different computers (called servers). About 15 million visitors access Amazon.com each day and use the database to make purchases. The database is continually updated as new books and other items are added to the inventory and stock quantities are updated as purchases are transacted. About 100 people are responsible for keeping the Amazon database up-to-date.

A database may be generated and maintained manually or it may be computerized. For example, a library card catalog is a database that may be created and maintained manually. A computerized database may be created and maintained either by a group of application programs written specifically for that task or by a database management system. We are only concerned with computerized databases in this book.

A **database management system (DBMS)** is a collection of programs that enables users to create and maintain a database. The DBMS is a *general-purpose software system* that facilitates the processes of *defining, constructing, manipulating,* and *sharing* databases among various users and applications. **Defining** a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called **meta-data**. **Constructing** the database is the process of storing the data on some storage medium that is controlled by the DBMS. **Manipulating** a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data. **Sharing** a database allows multiple users and programs to access the database simultaneously.

An **application program** accesses the database by sending queries or requests for data to the DBMS. A **query**[2] typically causes some data to be retrieved; a **transaction** may cause some data to be read and some data to be written into the database.

Other important functions provided by the DBMS include *protecting* the database and *maintaining* it over a long period of time. **Protection** includes *system protection* against hardware or software malfunction (or crashes) and *security protection* against unauthorized or malicious access. A typical large database may have a life cycle of many years, so the DBMS must be able to **maintain** the database system by allowing the system to evolve as requirements change over time.

It is not absolutely necessary to use general-purpose DBMS software to implement a computerized database. We could write our own set of programs to create and maintain the database, in effect creating our own *special-purpose* DBMS software. In either case—whether we use a general-purpose DBMS or not—we usually have to deploy a considerable amount of complex software. In fact, most DBMSs are very complex software systems.

To complete our initial definitions, we will call the database and DBMS software together a **database system**. Figure 1.1 illustrates some of the concepts we have discussed so far.

1.2 An Example

Let us consider a simple example that most readers may be familiar with: a UNIVERSITY database for maintaining information concerning students, courses, and grades in a university environment. Figure 1.2 shows the database structure and a few sample data for such a database. The database is organized as five files, each of which stores **data records** of the same type.[3] The STUDENT file stores data on each student, the COURSE file stores data on each course, the SECTION file stores data on each section of a course, the GRADE_REPORT file stores the grades that students receive in the various sections they have completed, and the PREREQUISITE file stores the prerequisites of each course.

To *define* this database, we must specify the structure of the records of each file by specifying the different types of **data elements** to be stored in each record. In Figure 1.2, each STUDENT record includes data to represent the student's Name, Student_number, Class (such as freshman or '1', sophomore or '2', and so forth), and

---

[2]The term *query*, originally meaning a question or an inquiry, is loosely used for all types of interactions with databases, including modifying the data.

[3]We use the term *file* informally here. At a conceptual level, a *file* is a *collection* of records that may or may not be ordered.