Database Processing

Fundamentals, Design, and Implementation

EDITION 13

David M. Kroenke • David J. Auer

Database Processing

Fundamentals, Design, and Implementation

EDITION 13

This page intentionally left blank

Database Processing

Fundamentals, Design, and Implementation

EDITION 13

David M. Kroenke

David J. Auer
Western Washington University

PEARSON

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

Editor in Chief: Stephanie Wall
Executive Editor: Bob Horan
Program Manager: Kelly Loftus
Editorial Assistant: Kaylee Rotella
Director of Marketing: Maggie Moylan
Executive Marketing Manager: Anne Fahlgren
Marketing Assistant: Gianna Sandri
Senior Managing Editor: Judy Leale
Production Project Manager: Jane Bonnell
Operations Specialist: Michelle Klein
Senior Art Director: Janet Slowik
Interior and Cover Designer: Karen Quigley
Cover Images: Shutterstock Inc.
Media Project Manager, Editorial: Denise Vaughn
Media Project Manager, Production: Lisa Rinaldi
Composition/Full-Service Project Management: Integra
Printer/Binder: R.R. Donnelley/Willard
Cover Printer: Lehigh-Phoenix Color/Hagerstown
Text Font: 10/12 Kepler Standard Light

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

Library of Congress Cataloging-in-Publication Data

Kroenke, David M.
Database processing : fundamentals, design, and implementation.—Edition 13/David M. Kroenke, David J. Auer, Western Washington University.
pages cm
ISBN 978-0-13-305835-2
1. Database management. I. Auer, David J. II. Title.
QA76.9.D3K76 2013
005.74—dc23

2013020780

10 9 8 7 6 5 4 3 2 1

PEARSON

ISBN 10: 0-13-305835-2
ISBN 13: 978-0-13-305835-2

Brief Contents

This page intentionally left blank

Contents

Chapter 5: Data Modeling with the Entity-Relationship Model 179

Chapter 6: Transforming Data Models into Database Designs 230

Part 3 • Database Implementation 277

Part 4 • Multiuser Database Processing 393

ONLINE CHAPTER: SEE PAGE 433 FOR INSTRUCTIONS

ONLINE CHAPTER: SEE PAGE 433 FOR INSTRUCTIONS

ONLINE CHAPTER: SEE PAGE 433 FOR INSTRUCTIONS

This page intentionally left blank

Preface

The 13th edition of *Database Processing: Fundamentals, Design, and Implementation* refines the organization and content of this classic textbook to reflect a new teaching and professional workplace environment. Students and other readers of this book will benefit from new content and features in this edition.

New to This Edition

Content and features new to the 13th edition of *Database Processing: Fundamentals, Design, and Implementation* include:

- Material on *Big Data* and the evolving *NoSQL movement* has been moved to Chapter 12 and expanded upon. Big Data is the theme for the chapter. New material on virtualization, cloud computing, and the development of non-relational unstructured data stores (such as Cassandra and HBase) and the Hadoop Distributed File System (HDFS) is also included in Chapter 12.
- Each chapter now features an independent Case Question set. The Case Question sets are problem sets that generally do not require the student to have completed work on the same case in a previous chapter (there is one intentional exception that ties data modeling and database design together). Although in some instances the same basic named case may be used in different chapters, each instance is still completely independent of any other instance.
- The SQL topics of JOIN…ON and OUTER JOIN previously in Chapter 7 have been moved to Chapter 2 so nearly all SQL query topics are covered in one chapter (the exception is correlated subqueries, which are still reserved for Chapter 8).
- The coverage of SQL Persistent Stored Modules (SQL/PSM) in Chapter 7, Chapter 10, Chapter 10A, Chapter 10B, and Chapter 10C now includes a discussion of user-defined functions.
- The use of Microsoft Access 2013 to demonstrate and reinforce basic principles of database creation and use. This book has been revised to update all references to Microsoft Access and other Microsoft Office products (e.g., Microsoft Excel) to the recently released Microsoft Office 2013 versions.
- The updating of the book to reflect the use of Microsoft SQL Server 2012, the current version of Microsoft SQL Server. Although most of the topics covered are backward compatible with Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2008 R2 Express edition, all material in the book now uses SQL Server 2012 in conjunction with Office 2013, exclusively.
- The updating of the book to use MySQL 5.6, which is the current generally available (GA) release of MySQL. Further, we also now use the MySQL Installer for Windows for installations on computers with the Windows operating system.
- The use of the Microsoft Windows Server 2012 as the server operating system and Windows 8 as the workstation operating system generally discussed and illustrated in the text. These are the current Microsoft server and workstation operating systems. We do keep some Windows 7 material where it seems appropriate and in some places present both the Windows 7 and Window 8 versions of operations and utilities.

- The addition of online Appendix J, "Business Intelligence Systems." This appendix contains material removed from Chapter 12 to make room for new material on Big Data and the Not Only SQL movement.
- We have updated online Appendix I, "Getting Started with Web Servers, PHP, and the Eclipse PDT." This new material provides a detailed introduction to the installation and use of the Microsoft IIS Web server, PHP, and the Eclipse IDE used for Web database application development as discussed in Chapter 11.

Fundamentals, Design, and Implementation

With today's technology, it is impossible to utilize a DBMS successfully without first learning fundamental concepts. After years of developing databases with business users, we have developed what we believe to be a set of essential database concepts. These are augmented by the concepts necessitated by the increasing use of the Internet, the World Wide Web, and commonly available analysis tools. Thus, the organization and topic selection of the 13th edition are designed to:

- Present an early introduction to SQL queries.
- Use a "spiral approach" to database design.
- Use a consistent, generic Information Engineering (IE) Crow's Foot E-R diagram notation for data modeling and database design.
- Provide a detailed discussion of specific normal forms within a discussion of normalization that focuses on pragmatic normalization techniques.
- Use current DBMS technology: Microsoft Access 2013, Microsoft SQL Server 2012, Oracle Database 11*g* Release 2, and MySQL 5.6.
- Create Web database applications based on widely used Web development technology.
- Provide an introduction to business intelligence (BI) systems.
- Discuss the dimensional database concepts used in database designs for data warehouses and OnLine Analytical Processing (OLAP).
- Discuss the emerging and important topics of server virtualization, cloud computing, Big Data, and the Not Only SQL movement.

These changes have been made because it has become obvious that the basic structure of the earlier editions (up to and including the 9th edition—the 10th edition introduced many of the changes we used in the 11th and 12th editions and retain in the 13th edition) was designed for a teaching environment that no longer exists. The structural changes to the book were made for several reasons:

- Unlike the early years of database processing, today's students have ready access to data modeling and DBMS products.
- Today's students are too impatient to start a class with lengthy conceptual discussions on data modeling and database design. They want to do something, see a result, and obtain feedback.
- In the current economy, students need to reassure themselves that they are learning marketable skills.

Early Introduction of SQL DML

Given these changes in the classroom environment, this book provides an early introduction to SQL data manipulation language (DML) SELECT statements. The discussion of SQL data definition language (DDL) and additional DML statements occurs in Chapters 7 and 8. By encountering SQL SELECT statements in Chapter 2, students learn early in the class how to query data and obtain results, seeing firsthand some of the ways that database technology will be useful to them.

The text assumes that students will work through the SQL statements and examples with a DBMS product. This is practical today because nearly every student has access to Microsoft

Access. Therefore, Chapters 1 and 2 and Appendix A, "Getting Started with Microsoft Access 2013," are written to support an early introduction of Microsoft Access 2013 and the use of Microsoft Access 2013 for SQL queries (Microsoft Access 2013 QBE query techniques are also covered).

If a non–Microsoft Access-based approach is desired, versions of SQL Server 2012, Oracle Database 11*g* Release 2, and MySQL 5.6 are readily available for use. Free versions of the three major DBMS products covered in this book (SQL Server 2012 Express, Oracle Database Express Edition 11*g* Release 2, and MySQL 5.6 Community Edition) are available for download. For a detailed discussion of the available DBMS products, see Chapter 10, pages 429-430. Thus, students can actively use a DBMS product by the end of the first week of class.

BY THE WAY

The presentation and discussion of SQL are spread over four chapters so students can learn about this important topic in small bites. SQL SELECT statements are taught in Chapter 2. SQL data definition language (DDL) and SQL data manipulation language (DML) statements are presented in Chapter 7. Correlated subqueries and EXISTS/NOT EXISTS statements are described in Chapter 8, while SQL transaction control language (TCL) and SQL data control language (DCL) are discussed in Chapter 9. Each topic appears in the context of accomplishing practical tasks. Correlated subqueries, for example, are used to verify functional dependency assumptions, a necessary task for database redesign.

This box illustrates another feature used in this book: BTW boxes are used to separate comments from the text discussion. Sometimes they present ancillary material; other times they reinforce important concepts.

A Spiral Approach to the Database Design Process

Today, databases arise from three sources: (1) from the need to integrate existing data from spreadsheets, data files, and database extracts; (2) from the need to develop new information systems to support business processes; and (3) from the need to redesign an existing database to adapt to changing requirements. We believe that the fact that these three sources exist present instructors with a significant pedagogical opportunity. Rather than teach database design just once from data models, why not teach database design three times, once for each of these sources? In practice, this idea has turned out to be even more successful than expected.

Database Design Iteration 1: Databases from Existing Data

Considering the design of databases from existing data, if someone were to e-mail us a set of tables and say, "Create a database from them," how would we proceed? We would examine the tables in light of normalization criteria and then determine whether the new database was for query only or whether it was for query and update. Depending on the answer, we would denormalize the data, joining them together, or we would normalize the data, pulling them apart. All of this is important for students to know and understand.

Therefore, the first iteration of database design gives instructors a rich opportunity to teach normalization, not as a set of theoretical concepts, but rather as a useful toolkit for making design decisions for databases created from existing data. Additionally, the construction of databases from existing data is an increasingly common task that is often assigned to junior staff members. Learning how to apply normalization to the design of databases from existing data not only provides an interesting way of teaching normalization, it is also common and useful!

We prefer to teach and use a pragmatic approach to normalization and present this approach in Chapter 3. However, we are aware that many instructors like to teach normalization in the context of a step-by-step normal form presentation (1NF, 2NF, 3NF, then BCNF), and Chapter 3 now includes additional material to provide more support for this approach as well.

In today's workplace, large organizations are increasingly licensing standardized software from vendors such as SAP, Oracle, and Siebel. Such software already has a database design. But with every organization running the same software, many are learning that they can only gain a competitive advantage if they make better use of the data in those predesigned databases. Hence, students who know how to extract data and create read-only databases for reporting and data mining have obtained marketable skills in the world of ERP and other packaged software solutions.

Database Design Iteration 2: Data Modeling and Database Design

The second source of databases is from new systems development. Although not as common as in the past, many databases are still created from scratch. Thus, students still need to learn data modeling, and they still need to learn how to transform data models into database designs which are then implemented in a DBMS product.

The IE Crow's Foot Model as a Design Standard

This edition uses a generic, standard IE Crow's Foot notation. Your students should have no trouble understanding the symbols and using the data modeling or database design tool of your choice.

IDEF1X (which was used as the preferred E-R diagram notation in the 9th edition of this text) is explained in Appendix C, "E-R Diagrams and the IDEF1X Standard," in case your students will graduate into an environment where it is used or if you prefer to use it in your classes. UML is explained in Appendix D, "E-R Diagrams and the UML Standard," in case you prefer to use UML in your classes.

BY THE WAY

The choice of a data modeling tool is somewhat problematic. Of the two most readily available tools, Microsoft Visio 2013 has been rewritten as a very rudimentary database design tool, while Sun Microsystems MySQL Workbench, is a database design tool, not a data modeling tool. MySQL Workbench cannot produce an N:M relationship as such (as a data model requires), but has to immediately break it into two 1:N relationships (as database design does). Therefore, the intersection table must be constructed and modeled. This confounds data modeling with database design in just the way that we are attempting to teach students to avoid.

To be fair to Microsoft Visio 2013, it is true that data models with N:M relationships can be drawn using the standard Microsoft Visio 2013 drawing tools. Unfortunately, Microsoft has chosen to remove many of the best database design tools that were in Microsoft Visio 2010, and Microsoft Visio 2013 lacks the tools that made it a favorite of Microsoft Access and Microsoft SQL Server users. For a full discussion of these tools, see Appendix E, "Getting Started with the MySQL Workbench Data Modeling Tools," and Appendix F, "Getting Started with the Microsoft Visio 2013."

Good data modeling tools are available, but they tend to be more complex and expensive. Two examples are Visible Systems' Visible Analyst and CA Technologies' CA ERwin Data Modeler. Visible Analyst is available in a student edition (at a modest price), and a one-year time-limited CA ERwin Data Modeler Community Edition suitable for class use can be downloaded from *http://erwin.com/products/data-modeler/community-edition* has limited the number of objects that can be created by this edition to 25 entities per model and disabled some other features (see *http://erwin.com/content/products/CA-ERwin-r9-Community-Edition-Matrix-na.pdf*), but there is still enough functionality to make this product a possible choice for class use.

Database Design from E-R Data Models

As we discuss in Chapter 6, designing a database from data models consists of three tasks: representing entities and attributes with tables and columns; representing maximum cardinality by creating and placing foreign keys; and representing minimum cardinality via constraints, triggers, and application logic.

The first two tasks are straightforward. However, designs for minimum cardinality are more difficult. Required parents are easily enforced using NOT NULL foreign keys and referential integrity constraints. Required children are more problematic. In this book, however, we simplify the discussion of this topic by limiting the use of referential integrity actions and by supplementing those actions with design documentation. See the discussion around Figure 6-28.

Although the design for required children is complicated, it is important for students to learn. It also provides a reason for students to learn about triggers as well. In any case, the discussion of these topics is much simpler than it was in prior editions because of the use of the IE Crow's Foot model and ancillary design documentation.

Database Implementation from Database Designs

Of course, to complete the process, a database design must be implemented in a DBMS product. This is discussed in Chapter 7, where we introduce SQL DDL for creating tables and SQL DML for populating the tables with data.

BY THE WAY David Kroenke is the creator of the semantic object model (SOM). The SOM is presented in Appendix H, "The Semantic Object Model." The E-R data model is used everywhere else in the text.

Database Design Iteration 3: Database Redesign

Database redesign, the third iteration of database design, is both common and difficult. As stated in Chapter 8, information systems cause organizational change. New information systems give users new behaviors, and as users behave in new ways, they require changes in their information systems.

Database redesign is by nature complex. Depending on your students, you may wish to skip it, and you can do so without loss of continuity. Database redesign is presented after the discussion of SQL DDL and DML in Chapter 7 because it requires the use of advanced SQL. It also provides a practical reason to teach correlated subqueries and EXISTS/NOT EXISTS statements.

Active Use of a DBMS Product

We assume that students will actively use a DBMS product. The only real question becomes "which one?" Realistically, most of us have four alternatives to consider: Microsoft Access, Microsoft SQL Server, Oracle Database, or MySQL. You can use any of those products with this text, and tutorials for each of them are presented for Microsoft Access 2013 (Appendix A), SQL Server 2012 (Chapter 10A), Oracle Database 11*g* Release 2 (Chapter 10B), and MySQL 5.6 (Chapter 10C). Given the limitations of class time, it is probably necessary to pick and use just one of these products. You can often devote a portion of a lecture to discussing the characteristics of each, but it is usually best to limit student work to one of them. The possible exception to this is starting the course with Microsoft Access and then switching to a more robust DBMS product later in the course.

Using Microsoft Access 2013

The primary advantage of Microsoft Access is accessibility. Most students already have a copy, and, if not, copies are easily obtained. Many students will have used Microsoft Access in their introductory or other classes. Appendix A, "Getting Started with Microsoft Access 2013," is a tutorial on Microsoft Access 2013 for students who have not used it but who wish to use it with this book.

However, Microsoft Access has several disadvantages. First, as explained in Chapter 1, Microsoft Access is a combination application generator and DBMS. Microsoft Access confuses students because it confounds database processing with application development. Also, Microsoft Access 2013 hides SQL behind its query processor and makes SQL appear as an afterthought rather than a foundation. Furthermore, as discussed in Chapter 2, Microsoft Access 2013 does not correctly process some of the basic SQL-92 standard statements in its default setup. Finally, Microsoft Access 2013 does not support triggers. You can simulate triggers by trapping Windows events, but that technique is nonstandard and does not effectively communicate the nature of trigger processing.

Using SQL Server 2012, Oracle Database 11*g* Release 2, or MySQL 5.6

Choosing which of these products to use depends on your local situation. Oracle Database 11*g* Release 2, a superb enterprise-class DBMS product, is difficult to install and administer.

However, if you have local staff to support your students, it can be an excellent choice. As shown in Chapter 10B, Oracle's SQL Developer GUI tool (or SQL*Plus if you are dedicated to this beloved command-line tool) is a handy tool for learning SQL, triggers, and stored procedures. In our experience, students require considerable support to install Oracle on their own computers, and you may be better off to use Oracle from a central server.

SQL Server 2012, although probably not as robust as Oracle Database 11*g* Release 2, is easy to install on Windows machines, and it provides the capabilities of an enterprise-class DBMS product. The standard database administrator tool is the Microsoft SQL Server Management Studio GUI tool. As shown in Chapter 10A, SQL Server 2012 can be used to learn SQL, triggers, and stored procedures.

MySQL 5.6, discussed in Chapter 10C, is an open-source DBMS product that is receiving increased attention and market share. The capabilities of MySQL are continually being upgraded, and MySQL 5.6 supports stored procedures and triggers. MySQL also has excellent GUI tools in the MySQL Workbench and an excellent command-line tool (the MySQL Command Line Client). It is the easiest of the three products for students to install on their own computers. It also works with the Linux operating system and is popular as part of the AMP (Apache–MySQL–PHP) package (known as WAMP on Windows and LAMP on Linux).

BY THE WAY If the DBMS you use is not driven by local circumstances and you do have a choice, we recommend using SQL Server 2012. It has all of the features of an enterprise-class DBMS product, and it is easy to install and use. Another option is to start with Microsoft Access 2013 if it is available and switch to SQL Server 2012 at Chapter 7. Chapters 1 and 2 and Appendix A are written specifically to support this approach. A variant is to use Microsoft Access 2013 as the development tool for forms and reports running against an SQL Server 2012 database.

If you prefer a different DBMS product, you can still start with Microsoft Access 2013 and switch later in the course. See the detailed discussion of the available DBMS products on pages 429–430 in Chapter 10 for a good review of your options.

Focus on Database Application Processing

In this edition, we clearly draw the line between *application development* per se and *database application processing*. Specifically, we have:

- Focused on specific database dependent applications:
  - Web-based, database-driven applications
  - XML-based data processing
  - Business intelligence (BI) systems applications
- Emphasized the use of commonly available, multiple–OS-compatible application development languages.
- Limited the use of specialized vendor-specific tools and programming languages as much as possible.

There is simply not enough room in this book to provide even a basic introduction to programming languages used for application development such as the Microsoft .NET languages and Java. Therefore, rather than attempting to introduce these languages, we leave them for other classes where they can be covered at an appropriate depth. Instead, we focus on basic tools that are relatively straightforward to learn and immediately applicable to database-driven applications. We use PHP as our Web development language, and we use the readily available Eclipse integrate development environment (IDE) as our development tool. The result is a very focused final section of the book, where we deal specifically with the interface between databases and the applications that use them.

BY THE WAY Although we try to use widely available software as much as possible, there are, of course, exceptions where we must use vendor-specific tools. For BI applications, for example, we draw on Microsoft Excel's PivotTable capabilities and the Microsoft PowerPivot for Microsoft Excel 2013 add-in, and on the Microsoft SQL Server 2012 SP1 Data Mining Add-ins for Microsoft Office. However, either alternatives to these tools are available (*OpenOffice.org* DataPilot capabilities, the Palo OLAP Server) or the tools are generally available for download.

Business Intelligence Systems and Dimensional Databases

This edition maintains coverage of business intelligence (BI) systems (Chapter 12 and Appendix J). The chapter includes a discussion of dimensional databases, which are the underlying structure for data warehouses, data marts, and OLAP servers. It still covers data management for data warehouses and data marts and also describes reporting and data mining applications, including OLAP.

Appendix J includes in-depth coverage of two applications that should be particularly interesting to students. The first is RFM analysis, a reporting application frequently used by mail order and e-commerce companies. The complete RFM analysis is accomplished in Appendix J through the use of standard SQL statements. Additionally, this chapter includes a market basket analysis that is also performed using SQL correlated subqueries. This chapter can be assigned at any point after Chapter 8 and could be used as a motivator to illustrate the practical applications of SQL midcourse.

Overview of the Chapters in the 13th Edition

Chapter 1 sets the stage by introducing database processing, describing basic components of database systems and summarizing the history of database processing. If students are using Microsoft Access 2013 for the first time (or need a good review), they will also need to study Appendix A, "Getting Started with Microsoft Access 2013," at this point. Chapter 2 presents SQL SELECT statements. It also includes sections on how to submit SQL statements to Microsoft Access 2013, SQL Server 2012, Oracle Database 11*g* Release 2, and MySQL 5.6.

The next four chapters, Chapters 3 through 6, present the first two iterations of database design. Chapter 3 presents the principles of normalization to Boyce-Codd normal form (BNCF). It describes the problems of multivalued dependencies and explains how to eliminate them. This foundation in normalization is applied in Chapter 4 to the design of databases from existing data.

Chapters 5 and 6 describe the design of new databases. Chapter 5 presents the E-R data model. Traditional E-R symbols are explained, but the majority of the chapter uses IE Crow's Foot notation. Chapter 5 provides a taxonomy of entity types, including strong, ID-dependent, weak but not ID-dependent, supertype/subtype, and recursive. The chapter concludes with a simple modeling example for a university database.

Chapter 6 describes the transformation of data models into database designs by converting entities and attributes to tables and columns, by representing maximum cardinality by creating and placing foreign keys, and by representing minimum cardinality via carefully designed DBMS constraints, triggers, and application code. The primary section of this chapter parallels the entity taxonomy in Chapter 5.

Chapter 7 presents SQL DDL, DML, and SQL/Persistent Stored Modules (SQL/PSM). SQL DDL is used to implement the design of an example introduced in Chapter 6. INSERT, UPDATE, MERGE, and DELETE statements are discussed, as are SQL views. Additionally, the principles of embedding SQL in program code are presented, SQL/PSM is discussed, and triggers and stored procedures are explained.

Database redesign, the third iteration of database design, is described in Chapter 8. This chapter presents SQL correlated subqueries and EXISTS/NOT EXISTS statements and uses those statements in the redesign process. Reverse engineering is described, and basic redesign patterns are illustrated and discussed.

Chapters 9, 10, 10A, 10B, and 10C consider the management of multiuser organizational databases. Chapter 9 describes database administration tasks, including concurrency, security, and backup and recovery. Chapter 10 is a general introduction to the online Chapters 10A, 10B, and 10C, which describe SQL Server 2012, Oracle Database 11*g* Release 2, and MySQL 5.6, respectively. These chapters show how to use these products to create database structures and process SQL statements. They also explain concurrency, security, and backup and recovery with each product. The discussion in Chapters 10A, 10B, and 10C parallels the order of discussion in Chapter 9 as much as possible, though rearrangements of some topics are made, as needed, to support the discussion of a specific DBMS product.

BY THE WAY We have maintained or extended our coverage of Microsoft Access, Microsoft SQL Server, Oracle Database, and MySQL (introduced in *Database Processing: Fundamentals, Design, and Implementation,* 11th edition) in this book. In order to keep the bound book to a reasonable length and to keep the cost of the book down, we have chosen to provide some material by download from our Web site at *www.pearsonhighered.com/kroenke*. There you will find:

- Chapter 10A—Managing Databases with SQL Server 2012
- Chapter 10B—Managing Databases with Oracle Database 11*g* Release 2
- Chapter 10C—Managing Databases with MySQL 5.6
- Appendix A—Getting Started with Microsoft Access 2013
- Appendix B—Getting Started with Systems Analysis and Design
- Appendix C—E-R Diagrams and the IDEF1X Standard
- Appendix D—E-R Diagrams and the UML Standard
- Appendix E—Getting Started with MySQL Workbench Data Modeling Tools
- Appendix F—Getting Started with Microsoft Visio 2013
- Appendix G—Data Structures for Database Processing
- Appendix H—The Semantic Object Model
- Appendix I—Getting Started with Web Servers, PHP, and the Eclipse PDT
- Appendix J—Business Intelligence Systems

Chapters 11 and 12 address standards for accessing databases. Chapter 11 presents ODBC, OLE DB, ADO.NET, ASP.NET, JDBC, and JavaServer Pages (JSP). It then introduces PHP (and the Eclipse IDE) and illustrates the use of PHP for the publication of databases via Web pages. This is followed by a description of the integration of XML and database technology. The chapter begins with a primer on XML and then shows how to use the FOR XML SQL statement in SQL Server.

Chapter 12 concludes the text with a discussion of BI systems, dimensional data models, data warehouses, data marts, server virtualization, cloud computing, Big Data, structured storage, and the Not Only SQL movement.

Supplements

This text is accompanied by a wide variety of supplements. Please visit the text's Web site at *www.pearsonhighered.com/kroenke* to access the instructor and student supplements described below. Please contact your Pearson sales representative for more details. All supplements were written by David Auer and Robert Crossler.

For Students

- Many of the sample databases used in this text are available online in Microsoft Access, SQL Server 2012, Oracle Database 11*g* Release 2, and MySQL 5.6 format.

For Instructors

- The *Instructor's Resource Manual* provides sample course syllabi, teaching suggestions, and answers to end-of-chapter review, project, and case questions.
- The *Test Item File* and *TestGen* include an extensive set of test questions in multiple-choice, true/false, fill-in-the-blank, short-answer, and essay format. The difficulty level and where the topic is covered in the text are noted for each question. The Test Item File is available in Microsoft Word and in TestGen. The TestGen software is PC/MAC compatible and preloaded with all of the Test Item File questions. You can manually or randomly view test questions and drag and drop to create a test. You can add or modify test-bank questions as needed. Our TestGens are converted for use in BlackBoard, WebCT, Angel, D2L, and Moodle. All conversions are available on the IRC.
- *PowerPoint Presentation Slides* feature lecture notes that highlight key terms and concepts. Instructors can customize the presentation by adding their own slides or editing the existing ones.
- The *Image Library* is a collection of the text art organized by chapter. This includes all figures, tables, and screenshots (as permission allows) to enhance class lectures and PowerPoint presentations.

Acknowledgments

We are grateful for the support of many people in the development of this 13th edition and previous editions. Thanks to Rick Mathieu at James Madison University for interesting and insightful discussions on the database course. Professor Doug MacLachlan from the Marketing Department at the University of Washington was most helpful in understanding the goals, objectives, and technology of data mining, particularly as it pertains to marketing. Don Nilson, formerly of the Microsoft Corporation, helped us understand the importance of XML to database processing. Kraig Pencil and Jon Junell of the College of Business and Economics at Western Washington University helped us refine the use of the book in the classroom.

In addition, we wish to thank the reviewers of this and previous editions:

Ann Aksut, *Central Piedmont Community College*
Allen Badgett, *Oklahoma City University*
Rich Beck, *Washington University*
Jeffrey J. Blessing, *Milwaukee School of Engineering*
Alan Brandyberry, *Kent State University*
Larry Booth, *Clayton State University*
Jason Deane, *Virginia Polytechnic Institute and State University*
Barry Flaschbart, *Missouri University of Science and Technology*
Andy Green, *Kennesaw State University*
Dianne Hall, *Auburn University*
Jeff Hassett, *University of Utah*
Barbara Hewitt, *Texas A&M, Kingsville*
William Hochstettler, *Franklin University*
Margaret Hvatum, *St. Louis Community College*
Nitin Kale, *University of Southern California, Los Angeles*
Darrel Karbginsky, *Chemeketa Community College*
Johnny Li, *South University*
Lin Lin, *New Jersey Institute of Technology*
Mike Morris, *Southeastern Oklahoma State University*
Jane Perschbach, *Texas A&M University–Central Texas*
Catherine Ricardo, *Iona College*
Kevin Roberts, *DeVry University*

Ioulia Rytikova, *George Mason University*
Christelle Scharff, *Pace University*
Julian M. Scher, *New Jersey Institute of Technology*
Namchul Shin, *Pace University*
K. David Smith, *Cameron University*
M. Jane Stafford, *Columbia College–Jefferson City*
Marcia Williams, *Bellevue Community College*
Timothy Woodcock, *Texas A&M University–Central Texas*

Finally, we would like to thank Bob Horan, our editor; Kelly Loftus, our editorial project manager during this project; Jane Bonnell, our production project manager; and Angel Chavez, our project manager; for their professionalism, insight, support, and assistance in the development of this project. We would also like to thank Robert Crossler for his detailed comments on the final manuscript. Finally, David Kroenke would like to thank his wife, Lynda, and David Auer would like to thank his wife, Donna, for their love, encouragement, and patience while this project was being completed.

David Kroenke
Seattle, Washington

David Auer
Bellingham, Washington

About the Authors

David M. Kroenke

Work Experience

David M. Kroenke has more than 35 years' experience in the computer industry. He began as a computer programmer for the U.S. Air Force, working both in Los Angeles and at the Pentagon, where he developed one of the world's first DBMS products while part of a team that created a computer simulation of World War III. That simulation served a key role for strategic weapons studies during a 10-year period of the Cold War.

From 1973 to 1978, Kroenke taught in the College of Business at Colorado State University. In 1977, he published the first edition of *Database Processing,* a significant and successful textbook that, more than 30 years later, you now are reading in its 13th edition. In 1978, he left Colorado State and joined Boeing Computer Services, where he managed the team that designed database management components of the IPAD project. After that, he joined with Steve Mitchell to form Mitchell Publishing and worked as an editor and author, developing texts, videos, and other educational products and seminars. Mitchell Publishing was acquired by Random House in 1986. During those years, he also worked as an independent consultant, primarily as a database disaster repairman helping companies recover from failed database projects.

In 1982, Kroenke was one of the founding directors of the Microrim Corporation. From 1984 to 1987, he served as the Vice President of Product Marketing and Development and managed the team that created and marketed the DBMS product R:base 5000 as well as other related products.

For the next five years, Kroenke worked independently while he developed a new data modeling language called the *semantic object model.* He licensed this technology to the Wall Data Corporation in 1992 and then served as the Chief Technologist for Wall Data's SALSA line of products. He was awarded three software patents on this technology.

Since 1998, Kroenke has continued consulting and writing. His current interests concern the practical applications of data mining techniques on large organizational databases. An avid sailor, he wrote *Know Your Boat: The Guide to Everything That Makes Your Boat Work,* which was published by McGraw-Hill in 2002.

Consulting

Kroenke has consulted with numerous organizations during his career. In 1978, he worked for Fred Brooks, consulting with IBM on a project that became the DBMS product DB2. In 1989, he consulted for the Microsoft Corporation on a project that became Microsoft Access. In the 1990s, he worked with Computer Sciences Corporation and with General Research Corporation for the development of technology and products that were used to model all of the U.S. Army's logistical data as part of the CALS project. Additionally, he has consulted for Boeing Computer Services, the U.S. Air Force Academy, Logicon Corporation, and other smaller organizations.

Publications

- *Database Processing,* Pearson Prentice Hall, 13 editions, 1977–present (coauthor with David Auer, 11th, 12th, and 13th editions)
- *Database Concepts*, Pearson Prentice Hall, six editions, 2004–present (coauthor with David Auer, 3rd, 4th, 5th, and 6th editions)

- *Using MIS*, Pearson Prentice Hall, six editions, 2006–present
- *Experiencing MIS*, Pearson Prentice Hall, four editions, 2007–present
- *MIS Essentials*, Pearson Prentice Hall, three editions, 2009–present
- *Processes, Systems, and Information: An Introduction to MIS*, Pearson Prentice Hall, 2013 (coauthor with Earl McKinney)
- *Know Your Boat: The Guide to Everything That Makes Your Boat Work*, McGraw-Hill, 2002
- *Management Information Systems*, Mitchell Publishing/Random House, three editions, 1987–1992
- *Business Computer Systems*, Mitchell Publishing/Random House, five editions, 1981–1990
- *Managing Information for Microcomputers*, Microrim Corporation, 1984 (coauthor with Donald Nilson)
- *Database Processing for Microcomputers*, Science Research Associates, 1985 (coauthor with Donald Nilson)
- *Database: A Professional's Primer*, Science Research Associates, 1978

Teaching

Kroenke taught in the College of Business at Colorado State University from 1973 to 1978. He also has taught part time in the Software Engineering program at Seattle University. From 1990 to 1991, he served as the Hanson Professor of Management Science at the University of Washington. Most recently, he taught at the University of Washington from 2002 to 2008. During his career, he has been a frequent speaker at conferences and seminars for computer educators. In 1991, the International Association of Information Systems named him Computer Educator of the Year.

Education

B.S., Economics, U.S. Air Force Academy, 1968
M.S., Quantitative Business Analysis, University of Southern California, 1971
Ph.D., Engineering, Colorado State University, 1977

Personal

Kroenke is married, lives in Seattle, and has two grown children and three grandchildren. He enjoys skiing, sailing, and building small boats. His wife tells him he enjoys gardening as well.

David J. Auer

Work Experience

David J. Auer has more than 30 years' experience teaching college-level business and information systems courses and for the past 20 years has worked professionally in the field of information technology. He served as a commissioned officer in the U.S. Air Force, with assignments to NORAD and the Alaskan Air Command in air defense operations. He later taught both business administration and music classes at Whatcom Community College and business courses for the Chapman College Residence Education Center at Whidbey Island Naval Air Station. He was a founder of the Puget Sound Guitar Workshop (now in its 39th year of operations). He worked as a psychotherapist and organizational development consultant for the Whatcom Counseling and Psychiatric Clinic's Employee Assistance Program and provided training for the Washington State Department of Social and Health Services. He has taught for Western Washington University's College of Business and Economics since 1981 and has been the college's Director of Information Systems and Technology Services since 1994.

Publications

- *Database Processing*, Pearson Prentice Hall, three editions, 2009–present (coauthor with David Kroenke)
- *Database Concepts*, Pearson Prentice Hall, four editions, 2007–present (coauthor with David Kroenke)
- *Network Administrator: NetWare 4.1*, Course Technology, 1997 (coauthor with Ted Simpson and Mark Ciampa)
- *New Perspectives on Corel Quattro Pro 7.0 for Windows 95,* Course Technology, 1997 (coauthor with June Jamrich Parsons, Dan Oja, and John Leschke)
- *New Perspectives on Microsoft Excel 7 for Windows 95—Comprehensive*, Course Technology, 1996 (coauthor with June Jamrich Parsons and Dan Oja)
- *New Perspectives on Microsoft Office Professional for Windows 95—Intermediate*, Course Technology, 1996 (coauthor with June Jamrich Parsons, Dan Oja, Beverly Zimmerman, Scott Zimmerman, and Joseph Adamski)
- *The Student's Companion for Use with Practical Business Statistics*, Irwin, two editions 1991 and 1993
- *Microsoft Excel 5 for Windows—New Perspectives Comprehensive,* Course Technology, 1995 (coauthor with June Jamrich Parsons and Dan Oja)
- *Introductory Quattro Pro 6.0 for Windows*, Course Technology, 1995 (coauthor with June Jamrich Parsons and Dan Oja)
- *Introductory Quattro Pro 5.0 for Windows*, Course Technology, 1994 (coauthor with June Jamrich Parsons and Dan Oja)

Teaching

Auer has taught in the College of Business and Economics at Western Washington University from 1981 to the present. From 1975 to 1981, he taught part time for community colleges, and from 1981 to 1984, he taught part time for the Chapman College Residence Education Center System. During his career, he has taught a wide range of courses in Quantitative Methods, Production and Operations Management, Statistics, Finance, and Management Information Systems. In MIS, he has taught Principles of Management Information Systems, Business Database Development, Computer Hardware and Operating Systems, and Telecommunications and Network Administration.

Education

B.A., English Literature, University of Washington, 1969
B.S., Mathematics and Economics, Western Washington University, 1978
M.A., Economics, Western Washington University, 1980
M.S., Counseling Psychology, Western Washington University, 1991

Personal

Auer is married, lives in Bellingham, Washington, and has two grown children and five grandchildren. He is active in his community, where he has been president of his neighborhood association and served on the City of Bellingham Planning and Development Commission. He enjoys music, playing acoustic and electric guitar, five-string banjo, and a bit of mandolin.

This page intentionally left blank

Getting Started

The two chapters in Part 1 provide an introduction to database processing. In Chapter 1, we consider the characteristics of databases and describe important database applications. Chapter 1 discusses the various database components, provides a survey of the knowledge you need to learn from this text, and also summarizes the history of database processing.

You will start working with a database in Chapter 2 and use that database to learn how to use Structured Query Language (SQL), a database-processing language, to query database data. You will learn how to query both single and multiple tables, and you will use SQL to investigate a practical example—looking for patterns in stock market data. Together, these two chapters will give you a sense of what databases are and how they are processed.

PART 1

Introduction

1

Chapter Objectives

- To understand the nature and characteristics of databases
- To survey some important and interesting database applications
- To gain a general understanding of tables and relationships
- To describe the components of a Microsoft Access database system and explain the functions they perform
- To describe the components of an enterprise-class database system and explain the functions they perform
- To define the term *database management system* (DBMS) and describe the functions of a DBMS
- To define the term *database* and describe what is contained within the database
- To define the term *metadata* and provide examples of metadata
- To define and understand database design from existing data
- To define and understand database design as new systems development
- To define and understand database design in database redesign
- To understand the history and development of database processing

This chapter introduces database processing. We will first consider the nature and characteristics of databases and then survey a number of important and interesting database applications. Next, we will describe the components of a database system and then, in general terms, describe how databases are designed. After that, we will survey the knowledge that you need to work with databases as an application developer or as a database administrator. Finally, we conclude this introduction with a brief history of database processing.

This chapter assumes a minimal knowledge of database use. It assumes that you have used a product such as Microsoft Access to enter data into a form, to produce a report, and possibly to execute a query. If you have not done these things, you should obtain a copy of Microsoft Access 2013 and work through the tutorial in Appendix A.

The Characteristics of Databases

The purpose of a database is to help people keep track of things, and the most commonly used type of database is the **relational database**. We will discuss the relational database model in depth in Chapter 3, so for now we just need to understand a few basic facts about how a relational database helps people track things of interest to them.

A relational database stores data in tables. **Data** are recorded facts and numbers. A **table** has rows and columns, like those in a spreadsheet. A database usually has multiple tables, and each table contains data about a different type of thing. For example, Figure 1-1 shows a database with two tables: the STUDENT table holds data about students, and the CLASS table holds data about classes.

Each **row** of a table has data about a particular occurrence or **instance** of the thing of interest. For example, each row of the STUDENT table has data about one of four students: Cooke, Lau, Harris, and Greene. Similarly, each row of the CLASS table has data about a particular class. Because each row *records* the data for a specific instance, rows are also known as **records**. Each **column** of a table stores a characteristic common to all rows. For example, the first column of STUDENT stores StudentNumber, the second column stores LastName, and so forth. Columns are also known as **fields**.

BY THE WAY

A table and a *spreadsheet* (also known as a *worksheet*) are very similar in that you can think of both as having rows, columns, and cells. The details that define a table as something different from a spreadsheet are discussed in Chapter 3. For now, the main differences you will see are that tables have column names instead of identifying letters (for example, *Name* instead of *A*) and that the rows are not necessarily numbered.

Although, in theory, you could switch the rows and columns by putting instances in the columns and characteristics in the rows, this is never done. Every database in this book—and 99.999999 percent of all databases throughout the world—stores instances in rows and characteristics in columns.

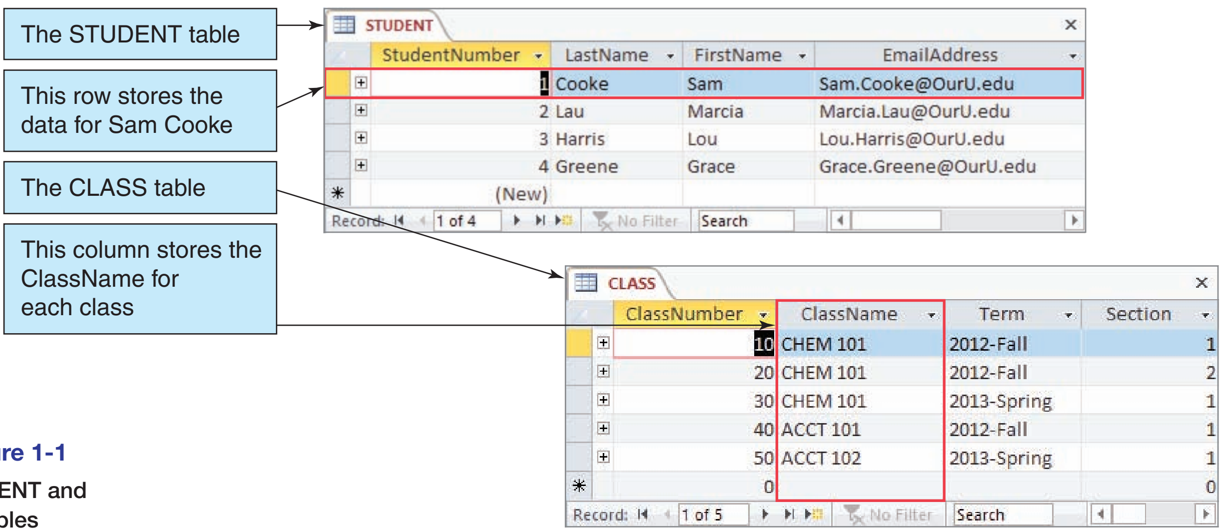


Figure 1-1
The STUDENT and CLASS Tables

A Note on Naming Conventions

In this text, table names appear in capital letters. This convention will help you to distinguish table names in explanations. However, you are not required to set table names in capital letters. Microsoft Access and similar programs will allow you to write a table name as STUDENT, student, Student, or stuDent, or in some other way.

Additionally, in this text, column names begin with a capital letter. Again, this is just a convention. You could write the column name Term as term, teRm, or TERM, or in any other way. To ease readability, we will sometimes create compound column names in which the first letter of each element of the compound word is capitalized. Thus, in Figure 1-1, the STUDENT table has columns StudentNumber, LastName, FirstName, and EmailAddress. Again, this capitalization is just a convenient convention. However, following these or other consistent conventions will make interpretation of database structures easier. For example, you will always know that STUDENT is the name of a table and that Student is the name of a column of a table.

A Database Has Data and Relationships

Figure 1-1 illustrates how database tables are structured to store data, but a database is not complete unless it also shows the relationships among the rows of data. To see why this is important, examine Figure 1-2. In this figure, the database contains all of the basic data shown in Figure 1-1 together with a GRADE table. Unfortunately, the relationships among the data are missing. In this format, the GRADE data are useless. It is like the joke about the sports commentator who announced: "Now for tonight's baseball scores: 2–3, 7–2, 1–0, and 4–5." The scores are useless without knowing the teams that earned them. Thus, a database contains both data and the relationships among the data.

Figure 1-3 shows the complete database that contains not only the data about students, classes, and grades, but also the relationships among the rows in those tables. For example, StudentNumber 100, who is Sam Cooke, earned a Grade of 3.7 in ClassNumber 10, which is Chem101. He also earned a Grade of 3.5 in ClassNumber 40, which is Acct101.

Figure 1-3 illustrates an important characteristic of database processing. Each row in a table is uniquely identified by a **primary key**, and the values of these keys are used to create

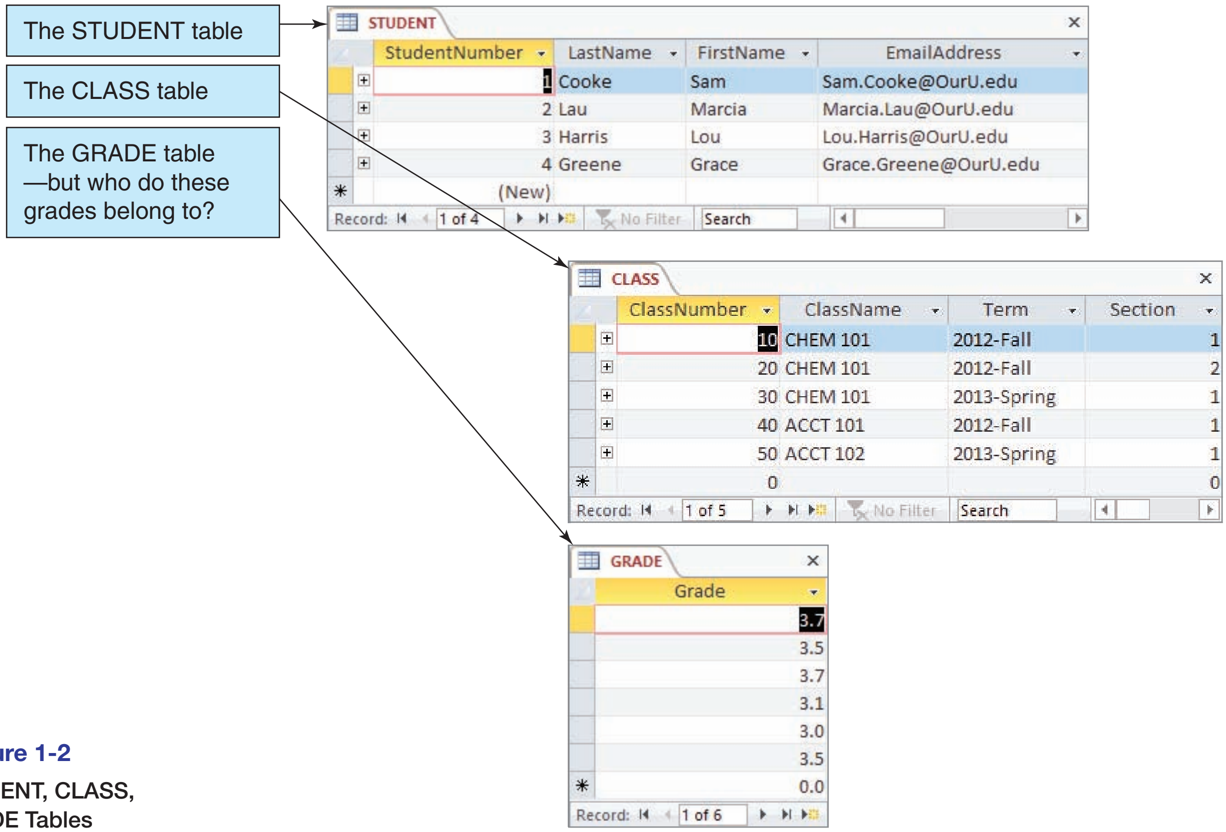


Figure 1-2
The STUDENT, CLASS, and GRADE Tables

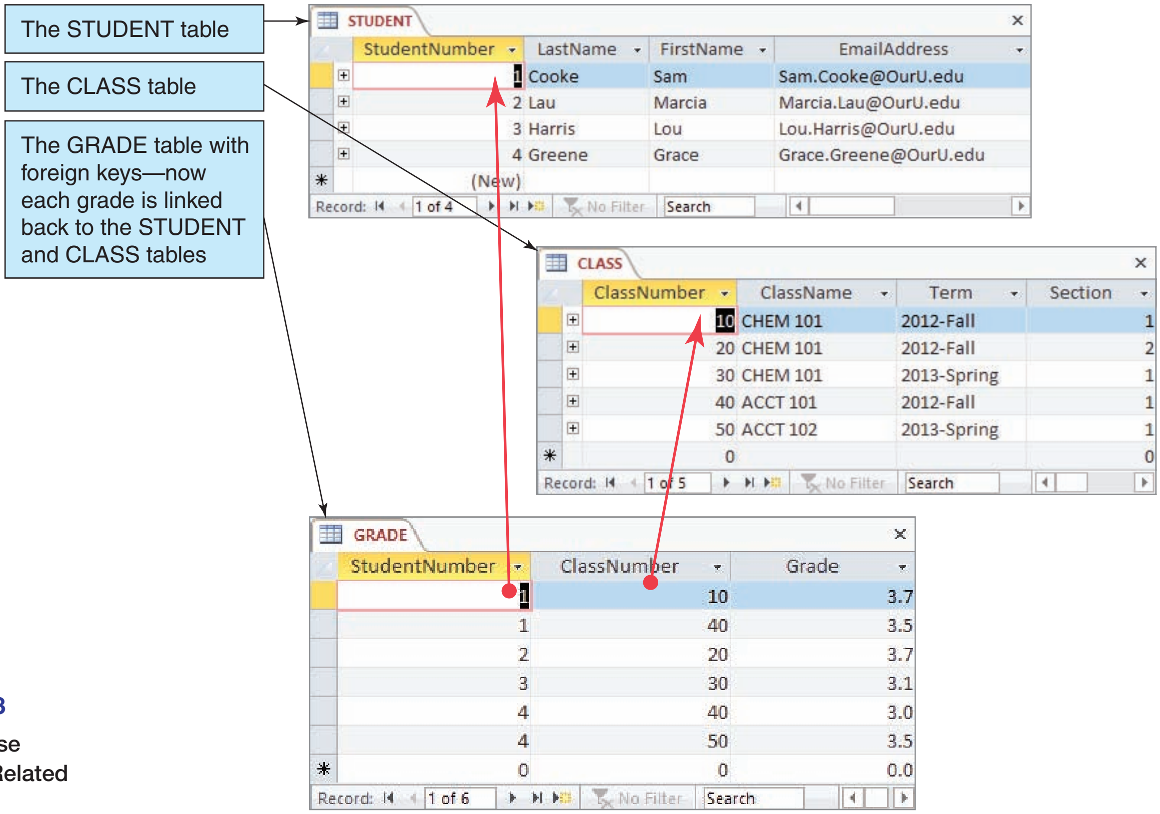


Figure 1-3
The Key Database Characteristic: Related Tables

the relationships between the tables. For example, in the STUDENT table, StudentNumber serves as the primary key. Each value of StudentNumber is unique and identifies a particular student. Thus, StudentNumber 1 identifies Sam Cooke. Similarly, ClassNumber in the CLASS table identifies each class. If the numbers used in primary key columns such as StudentNumber and ClassNumber are automatically generated and assigned in the database itself, then the key is also called a **surrogate key**.

By comparing Figures 1-2 and 1-3, we can see how the primary keys of STUDENT and CLASS were added to the GRADE table to provide GRADE with a primary key of (StudentNumber, ClassNumber) to uniquely identify each row. More important, in GRADE, StudentNumber and ClassNumber each now serve as a **foreign key**. A foreign key provides the link between two tables. By adding a foreign key, we create a **relationship** between the two tables.

Figure 1-4 shows a Microsoft Access 2013 view of the tables and relationships shown in Figure 1-3. In Figure 1-4, primary keys in each table are marked with key symbols, and connecting lines representing the relationships are drawn from the foreign keys (in GRADE) to the corresponding primary keys (in STUDENT and CLASS). The symbols on the relationship line (the number 1 and the infinity symbol) mean that, for example, one student in STUDENT can be linked to many grades in GRADE.

Databases Create Information

In order to make decisions, we need information upon which to base those decisions. Because we have already defined *data* as recorded facts and numbers, we can now define[1] **information** as:

- Knowledge derived from data
- Data presented in a meaningful context
- Data processed by summing, ordering, averaging, grouping, comparing, or other similar operations

[1]These definitions are from David M. Kroenke's books *Using MIS*, 6th ed. (Upper Saddle River, NJ: Prentice-Hall, 2014) and *Experiencing MIS*, 4th ed. (Upper Saddle River, NJ: Prentice-Hall, 2014). See these books for a full discussion of these definitions, as well as a discussion of a fourth definition, "a difference that makes a difference."

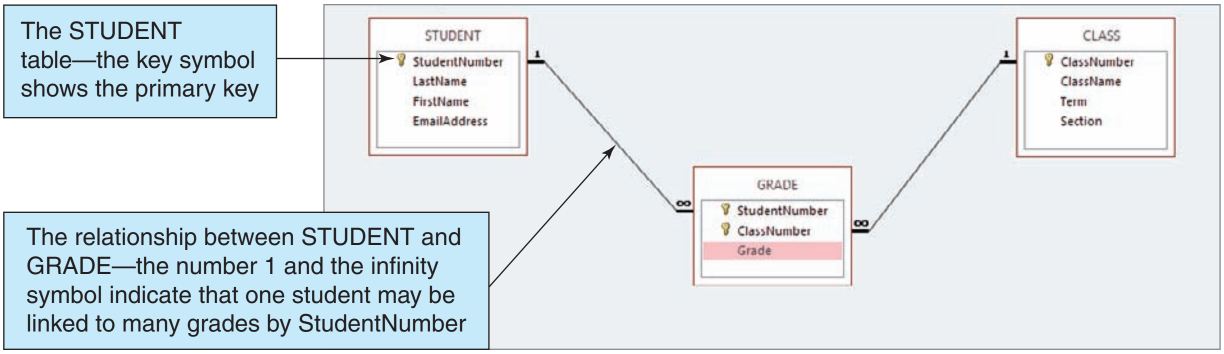


Figure 1-4
Microsoft Access 2013 View of Tables and Relationships

Databases record facts and figures, so they record data. They do so, however, in a way that enables them to produce information. The data in Figure 1-3 can be manipulated to produce a student's GPA, the average GPA for a class, the average number of students in a class, and so forth. In Chapter 2, you will be introduced to a language called Structured Query Language (SQL) that you can use to produce information from database data.

To summarize, relational databases store data in tables, and they represent the relationships among the rows of those tables. They do so in a way that facilitates the production of information. We will discuss the relational database model in depth in Part 2 of this book.

Database Examples

Today, database technology is part of almost every information system. This fact is not surprising when we consider that every information system needs to store data and the relationships among those data. Still, the vast array of applications that use this technology is staggering. Consider, for example, the applications listed in Figure 1-5.

Single-User Database Applications

In Figure 1-5, the first application is used by a single salesperson to keep track of the customers she has called and the contacts that she's had with them. Most salespeople do not build their own contact manager applications; instead, they license products such as GoldMine (see *www.goldmine.com*) or ACT! (see *http://na.sage.com/sage-act*).

Multiuser Database Applications

The next applications in Figure 1-5 are those that involve more than one user. The patient-scheduling application, for example, may have 15 to 50 users. These users will be appointment clerks, office administrators, nurses, dentists, doctors, and so forth. A database like this one may have as many as 100,000 rows of data in perhaps 5 or 10 different tables.

When more than one user employs a database application, there is always the chance that one user's work may interfere with another's. Two appointment clerks, for example, might assign the same appointment to two different patients. Special concurrency-control mechanisms are used to coordinate activity against the database to prevent such conflict. You will learn about these mechanisms in Chapter 9.

The third row of Figure 1-5 shows an even larger database application. A customer relationship management (CRM) system is an information system that manages customer contacts from initial solicitation through acceptance, purchase, continuing purchase, support, and so forth. CRM systems are used by salespeople, sales managers, customer service and support staff, and other personnel. A CRM database in a larger company might have 500 users and 10 million or more rows in perhaps 50 or more tables. According to Microsoft, in 2004, Verizon had an SQL Server customer database that contained more than 15 terabytes of data. If that data were published in books, a bookshelf 450 miles long would be required to hold them.

Enterprise resource planning (ERP) is an information system that touches every department in a manufacturing company. It includes sales, inventory, production planning,

Application	Example Users	Number of Users	Typical Size	Remarks
Sales contact manager	Salesperson	1	2,000 rows	Products such as GoldMine and Act! are database centric.
Patient appointment (doctor, dentist)	Medical office	15 to 50	100,000 rows	Vertical market software vendors incorporate databases into their software products.
Customer relationship management (CRM)	Sales, marketing, or customer service departments	500	10 million rows	Major vendors such as Microsoft and Oracle PeopleSoft Enterprise build applications around the database.
Enterprise resource planning (ERP)	An entire organization	5,000	10 million+ rows	SAP uses a database as a central repository for ERP data.
E-commerce site	Internet users	Possibly millions	1 billion+ rows	Drugstore.com has a database that grows at the rate of 20 million rows per day!
Digital dashboard	Senior managers	500	100,000 rows	Extractions, summaries, and consolidations of operational databases.
Data mining	Business analysts	25	100,000 to millions+	Data are extracted, reformatted, cleaned, and filtered for use by statistical data mining tools.

Figure 1-5
Example Database Applications

purchasing, and other business functions. SAP is the leading vendor of ERP applications, and a key element of its product is a database that integrates data from these various business functions. An ERP system may have 5,000 or more users and perhaps 100 million rows in several hundred tables.

E-Commerce Database Applications

E-commerce is another important database application. Databases are a key component of e-commerce order entry, billing, shipping, and customer support. Surprisingly, however, the largest databases at an e-commerce site are not order-processing databases. The largest databases are those that track customer browser behavior. Most of the prominent e-commerce companies, such as Amazon.com (*www.amazon.com*) and Drugstore.com (*www.drugstore.com*) keep track of the Web pages and the Web page components that they send to their customers. They also track customer clicks, additions to shopping carts, order purchases, abandoned shopping carts, and so forth.

E-commerce companies use Web activity databases to determine which items on a Web page are popular and successful and which are not. They also can conduct experiments to determine if a purple background generates more orders than a blue one and so forth. Such Web usage databases are huge. For example, Drugstore.com adds 20 million rows to its Web log database each day!

Reporting and Data Mining Database Applications

Two other example applications in Figure 1-5 are digital dashboards and data mining applications. These applications use the data generated by order processing and other operational systems to produce information to help manage the enterprise. Such applications do not generate new data, but instead summarize existing data to provide insights to management. Digital dashboards and other reporting systems assess past and current performance. Data mining applications predict future performance. We will consider such applications in Chapter 12. The bottom line is that database technology is used in almost every information system and involves databases ranging in size from a few thousand rows to many millions of rows.

BY THE WAY Do not assume that just because a database is small that its structure is simple. For example, consider parts distribution for a company that sells $1 million in parts per year and parts distribution for a company that sells $100 million in parts per year. Despite the difference in sales, the companies have similar databases. Both have the same kinds of data, about the same number of tables of data, and the same level of complexity in data relationships. Only the amount of data varies from one to the other. Thus, although a database for a small business may be small, it is not necessarily simple.

The Components of a Database System

As shown in Figure 1-6, a **database system** is typically defined to consist of four components: users, the database application, the database management system (DBMS), and the database. However, given the importance of **Structured Query Language (SQL)**, an internationally recognized standard language that is understood by all commercial DBMS products, in database processing and the fact that database applications typically send SQL statements to the DBMS for processing, we can refine our illustration of a database system to appear as shown in Figure 1-7.

Starting from the right of Figure 1-7, the **database** is a collection of related tables and other structures. The **database management system (DBMS)** is a computer program used to create, process, and administer the database. The DBMS receives requests encoded in SQL and translates those requests into actions on the database. The DBMS is a large, complicated program that is licensed from a software vendor; companies almost never write their own DBMS programs.

Figure 1-6
The Components of a Database System

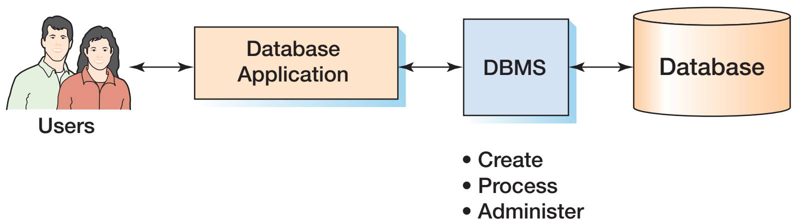
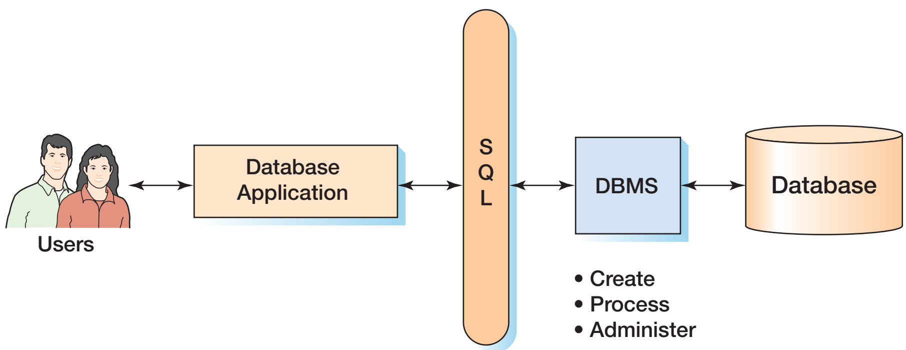


Figure 1-7
The Components of a Database System with SQL



A **database application** is a set of one or more computer programs that serves as an intermediary between the user and the DBMS. Application programs read or modify database data by sending SQL statements to the DBMS. Application programs also present data to users in the format of forms and reports. Application programs can be acquired from software vendors, and they are also frequently written in house. The knowledge you gain from this text will help you write database applications.

Users, the final component of a database system, employ a database application to keep track of things. They use forms to read, enter, and query data, and they produce reports to convey information.

Database Applications and SQL

Figure 1-7 shows that users interact directly with database applications. Figure 1-8 lists the basic functions of database applications.

First, an application program creates and processes forms. Figure 1-9 shows a typical form for entering and processing student enrollment data for the Student-Class-Grade database shown in Figures 1-3 and 1-4. Notice that this form hides the structure of the underlying tables

Figure 1-8
Basic Functions of Application Programs

Basic Functions of Application Programs
Create and process forms
Process user queries
Create and process reports
Execute application logic
Control the application itself

Figure 1-9
An Example Data Entry Form