



Database Concepts

Seventh Edition

DAVID M. KROENKE

DAVID J. AUER

Database Concepts

This page intentionally left blank



Database Concepts

Seventh Edition

David M. Kroenke

David J. Auer

Western Washington University

PEARSON

Boston Columbus Indianapolis New York San Francisco Hoboken
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
Acquisitions Editor: Nicole Sam
Program Manager Team Lead: Ashley Santora
Program Manager: Denise Vaughn
Editorial Assistant: Kaylee Rotella
Executive Marketing Manager: Anne K. Fahlgren
Project Manager Team Lead: Judy Leale
Project Manager: Ilene Kahn
Procurement Specialist: Michelle Klein

Senior Art Director: Janet Slowik
Text Designer: Integra Software Services
Cover Designer: Integra
Cover Art: shibanuk/Fotolia
Full-Service Project Management: Integra
Composition: Integra
Printer/Binder: Courier/Kendallville
Cover Printer: Lehigh-Phoenix Color/Hagerstown
Text Font: 10/12 Simoncini Garamond Std.

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

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

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

Microsoft® Windows®, and Microsoft Office® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

MySQL®, the MySQL GUI Tools® (MySQL Query Browser® and MySQL Administrator®), the MySQL Command Line Client®, the MySQL Workbench®, and the MySQL Connector/ODBC® are registered trademarks of Sun Microsystems, Inc./Oracle Corporation. Screenshots and icons reprinted with permission of Oracle Corporation. This book is not sponsored or endorsed by or affiliated with Oracle Corporation.

Express Edition 11g Release 2 2014 by Oracle Corporation. Reprinted with permission.

PHP is copyright The PHP Group 1999–2012, and is used under the terms of the PHP Public License v3.01 available at http://www.php.net/license/3_01.txt. This book is not sponsored or endorsed by or affiliated with The PHP Group.

Copyright © 2015, 2013, 2011 by Pearson Education, Inc., 221 River Street, Hoboken, New Jersey 07030. All rights reserved. Manufactured in the United States of America. This publication is protected by Copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, 221 River Street, Hoboken, New Jersey 07030.

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

Library of Congress Cataloging-in-Publication Data

Kroenke, David M., 1948–
Database concepts / David M. Kroenke and David J. Auer.—7e.
pages cm
Includes index.
ISBN 978-0-13-354462-6 (student edition)—ISBN 978-0-13-354478-7 (instructor edition)
1. Database management. 2. Relational databases. I. Auer, David J. II. Title.
QA76.9.D3K736 2015
005.74—dc23

2014010915

10 9 8 7 6 5 4 3 2 1

PEARSON

ISBN 10: 0-13-354462-1
ISBN 13: 978-0-13-354462-6



Brief Contents

PART I DATABASE FUNDAMENTALS 1

- 1** Getting Started **3**
- 2** The Relational Model **62**
- 3** Structured Query Language **119**

PART II DATABASE DESIGN 243

- 4** Data Modeling and the Entity-Relationship Model **245**
- 5** Database Design **287**

PART III DATABASE MANAGEMENT 333

- 6** Database Administration **335**
- 7** Database Processing Applications **392**
- 8** Big Data, Data Warehouses, and Business Intelligence Systems **445**

Glossary 495

Index 503

ONLINE APPENDICES: SEE PAGE 493 FOR INSTRUCTIONS

- Appendix A:** Getting Started with Microsoft SQL Server 2014 Express Edition
- Appendix B:** Getting Started with Oracle Database Express Edition 11g Release 2
- Appendix C:** Getting Started with MySQL 5.6 Community Server
- Appendix D:** James River Jewelry Project Questions
- Appendix E:** SQL Views, SQL/PSM, and Importing Data
- Appendix F:** Getting Started in Systems Analysis and Design
- Appendix G:** Getting Started with Microsoft Visio 2013
- Appendix H:** The Access Workbench—Section H—Microsoft Access 2013 Switchboards
- Appendix I:** Getting Started with Web Servers, PHP, and the NetBeans IDE
- Appendix J:** Business Intelligence Systems
- Appendix K:** Big Data



Contents

PART I DATABASE FUNDAMENTALS 1

1	Getting Started 3	3
	WHY USE A DATABASE? 4	
	WHAT IS A DATABASE SYSTEM? 14	
	WHAT IS A NOSQL DATABASE? 24	
	THE ACCESS WORKBENCH SECTION 1—GETTING STARTED WITH MICROSOFT ACCESS 25	
	Summary 54 • Key Terms 55 • Review Questions 55 • Exercises 56 • Access Workbench Key Terms 57 • Access Workbench Exercises 57 • San Juan Sailboat Charters Case Questions 59 • Garden Glory Project Questions 60 • James River Jewelry Project Questions (See Online Appendix D) 61 • The Queen Anne Curiosity Shop Project Questions 61	
2	The Relational Model 62	62
	RELATIONS 63	
	TYPES OF KEYS 66	
	THE PROBLEM OF NULL VALUES 74	
	FUNCTIONAL DEPENDENCIES AND NORMALIZATION 76	
	NORMAL FORMS: ONE STEP AT A TIME 88	
	THE ACCESS WORKBENCH SECTION 2—WORKING WITH MULTIPLE TABLES IN MICROSOFT ACCESS 89	
	Summary 105 • Key Terms 106 • Review Questions 106 • Exercises 108 • Access Workbench Key Terms 110 • Access Workbench Exercises 110 • Regional Labs Case Questions 113 • Garden Glory Project Questions 114 • James River Jewelry Project Questions (See Online Appendix D) 116 • The Queen Anne Curiosity Shop Project Questions 116	
3	Structured Query Language 119	119
	AN EXAMPLE DATABASE 120	
	SQL FOR DATA DEFINITION (DDL)—CREATING TABLES AND RELATIONSHIPS 126	
	SQL FOR DATA MANIPULATION (DML)—INSERTING DATA 139	
	SQL FOR DATA MANIPULATION (DML)—SINGLE TABLE QUERIES 143	
	SQL FOR DATA MANIPULATION (DML)—MULTIPLE TABLE QUERIES 161	
	SQL FOR DATA MANIPULATION (DML)—DATA MODIFICATION AND DELETION 176	
	SQL FOR DATA DEFINITION (DDL)—TABLE AND CONSTRAINT MODIFICATION AND DELETION 179	
	SQL VIEWS 182	
	THE ACCESS WORKBENCH SECTION 3—WORKING WITH QUERIES IN MICROSOFT ACCESS 182	
	Summary 211 • Key Terms 212 • Review Questions 212 • Exercises 216 • Access Workbench Key Terms 218 • Access Workbench Exercises 218 • Heather Sweeney Designs Case Questions 222 • Garden Glory Project Questions 232 • James River Jewelry Project Questions (See Online Appendix D) 235 • The Queen Anne Curiosity Shop Project Questions 236	

PART II DATABASE DESIGN 243

- 4 Data Modeling and the Entity-Relationship Model 245**
REQUIREMENTS ANALYSIS 246
THE ENTITY-RELATIONSHIP DATA MODEL 247
ENTITY-RELATIONSHIP DIAGRAMS 252
DEVELOPING AN EXAMPLE E-R DIAGRAM 262
**THE ACCESS WORKBENCH SECTION 4—
 PROTOTYPING USING MICROSOFT
 ACCESS 270**
*Summary 279 • Key Terms 280 • Review
 Questions 280 • Exercises 281 • Access
 Workbench Key Terms 282 • Access Workbench
 Exercises 282 • Highline University Mentor
 Program Case Questions 282 • Washington
 State Patrol Case Questions 284 • Garden
 Glory Project Questions 285 • James
 River Jewelry Project Questions (See Online
 Appendix D) 286 • The Queen Anne Curiosity
 Shop Project Questions 286*
- 5 Database Design 287**
THE PURPOSE OF A DATABASE DESIGN 288
**TRANSFORMING A DATA MODEL INTO A DATABASE
 DESIGN 288**
**REPRESENTING ENTITIES WITH THE RELATIONAL
 MODEL 289**
REPRESENTING RELATIONSHIPS 297
**DATABASE DESIGN AT HEATHER SWEENEY
 DESIGNS 310**
**THE ACCESS WORKBENCH SECTION 5—
 RELATIONSHIPS IN MICROSOFT
 ACCESS 318**
*Summary 324 • Key Terms 325 • Review
 Questions 325 • Exercises 326 • Access
 Workbench Key Terms 327 • Access Workbench
 Exercises 327 • San Juan Sailboat Charters
 Case Questions 328 • Washington State Patrol
 Case Questions 330 • Garden Glory Project
 Questions 330 • James River Jewelry Project
 Questions (See Online Appendix D) 330 • The
 Queen Anne Curiosity Shop Project Questions 331*

PART III DATABASE MANAGEMENT 333

- 6 Database Administration 335**
**THE HEATHER SWEENEY DESIGNS
 DATABASE 336**
**THE NEED FOR CONTROL, SECURITY, AND
 RELIABILITY 336**
CONCURRENCY CONTROL 338
**SQL TRANSACTION CONTROL LANGUAGE AND
 DECLARING LOCK CHARACTERISTICS 344**
CURSOR TYPES 348
DATABASE SECURITY 350
DATABASE BACKUP AND RECOVERY 357
ADDITIONAL DBA RESPONSIBILITIES 361
**THE ACCESS WORKBENCH SECTION 6—
 DATABASE ADMINISTRATION IN MICROSOFT
 ACCESS 361**
*Summary 381 • Key Terms 383 • Review
 Questions 383 • Exercises 385 • Access
 Workbench Key Terms 386 • Access Workbench
 Exercises 386 • Marcia's Dry Cleaning
 Case Questions 387 • Garden Glory Project
 Questions 388 • James River Jewelry Project
 Questions (See Online Appendix D) 389 • The
 Queen Anne Curiosity Shop Project Questions 390*
- 7 Database Processing
 Applications 392**
**THE DATABASE PROCESSING
 ENVIRONMENT 393**
WEB APPLICATION DATABASE PROCESSING 396
DATABASE PROCESSING AND XML 424
**THE ACCESS WORKBENCH SECTION 7—WEB
 DATABASE PROCESSING USING MICROSOFT
 ACCESS 426**
*Summary 435 • Key Terms 436 • Review
 Questions 436 • Exercises 438 • Access
 Workbench Exercises 440 • Marcia's Dry Cleaning
 Case Questions 440 • Garden Glory Project
 Questions 442 • James River Jewelry Project
 Questions (See Online Appendix D) 443 • The
 Queen Anne Curiosity Shop Project Questions 444*

8	Big Data, Data Warehouses, and Business Intelligence Systems	445
	BUSINESS INTELLIGENCE SYSTEMS	447
	THE RELATIONSHIP BETWEEN OPERATIONAL AND BI SYSTEMS	447
	REPORTING SYSTEMS AND DATA MINING APPLICATIONS	448
	DATA WAREHOUSES AND DATA MARTS	449
	OLAP	458
	DISTRIBUTED DATABASE PROCESSING	463
	OBJECT-RELATIONAL DATABASES	466
	BIG DATA AND THE NOT ONLY SQL MOVEMENT	466
	THE ACCESS WORKBENCH SECTION 8—BUSINESS INTELLIGENCE SYSTEMS USING MICROSOFT ACCESS	471
	Summary	484 • Key Terms 485 • Review Questions 485 • Exercises 487 • Access Workbench Exercises 488 • Marcia's Dry Cleaning Case Questions 489 • Garden Glory Project Questions 490 • James River Jewelry Project Questions (See Online Appendix D) 490 • The Queen Anne Curiosity Shop Project Questions 491

Glossary 495

Index 503

ONLINE APPENDICES: SEE PAGE 493 FOR INSTRUCTIONS

- Appendix A:** Getting Started with Microsoft SQL Server 2014 Express Edition
- Appendix B:** Getting Started with Oracle Database Express Edition 11g Release 2
- Appendix C:** Getting Started with MySQL 5.6 Community Server
- Appendix D:** James River Jewelry Project Questions
- Appendix E:** SQL Views, SQL/PSM, and Importing Data
- Appendix F:** Getting Started in Systems Analysis and Design
- Appendix G:** Getting Started with Microsoft Visio 2013
- Appendix H:** The Access Workbench—Section H—Microsoft Access 2013 Switchboards
- Appendix I:** Getting Started with Web Servers, PHP, and the NetBeans IDE
- Appendix J:** Business Intelligence Systems
- Appendix K:** Big Data



Preface

Colin Johnson is a production supervisor for a small manufacturer in Seattle. Several years ago, Colin wanted to build a database to keep track of components in product packages. At the time, he was using a spreadsheet to perform this task, but he could not get the reports he needed from the spreadsheet. Colin had heard about Microsoft Access, and he tried to use it to solve his problem. After several days of frustration, he bought several popular Microsoft Access books and attempted to learn from them. Ultimately, he gave up and hired a consultant who built an application that more or less met his needs. Over time, Colin wanted to change his application, but he did not dare try.

Colin was a successful businessperson who was highly motivated to achieve his goals. A seasoned Windows user, he had been able to teach himself how to use Microsoft Excel, Microsoft PowerPoint, and a number of production-oriented application packages. He was flummoxed at his inability to use Microsoft Access to solve his problem. “I’m sure I could do it, but I just don’t have any more time to invest,” he thought. This story is especially remarkable because it has occurred tens of thousands of times over the past decade to many other people.

Microsoft, Oracle, IBM, and other database management system (DBMS) vendors are aware of such scenarios and have invested millions of dollars in creating better graphical interfaces, hundreds of multipanel wizards, and many sample applications. Unfortunately, such efforts treat the symptoms and not the root of the problem. In fact, most users have no clear idea what the wizards are doing on their behalf. As soon as these users require changes to database structure or to components such as forms and queries, they drown in a sea of complexity for which they are unprepared. With little understanding of the underlying fundamentals, these users grab at any straw that appears to lead in the direction they want. The consequence is poorly designed databases and applications that fail to meet the users’ requirements.

Why can people like Colin learn to use a word processor or a spreadsheet product yet fail when trying to learn to use a DBMS product? First, the underlying database concepts are unnatural to most people. Whereas everyone knows what paragraphs and margins are, no one knows what a relation is. Second, it seems as though using a DBMS product ought to be easier than it is. “All I want to do is keep track of something. Why is it so hard?” people ask. Without knowledge of the relational model, breaking a sales invoice into five separate tables before storing the data is mystifying to business users.

This book is intended to help people like Colin understand, create, and use databases in a DBMS product, whether they are individuals who found this book in a bookstore or students using this book as their textbook in a class.

NEW TO THIS EDITION

Students and other readers of this book will benefit from new content and features in this edition. These include the following:

- Microsoft Office 2013, and particularly Microsoft Access 2013, is now the basic software used in the book and is shown running on Microsoft Windows 8.1.
- DBMS software coverage has been updated to include Microsoft SQL Server 2014 Express Edition and Oracle MySQL 5.6 Community Server.

- New material to illustrate the concepts of SQL joins has been added to Chapter 3 to make this material easier for students to understand.
- New material on SQL programming via SQL/Persistent Stored Modules (SQL/PSM) has been added to Appendix E to provide a better organized discussion and expanded discussion of this material, which had previously been spread among other parts of the book. This material also provides a discussion of importing Microsoft Excel worksheet data in to a database.
- Material on *Big Data* and the evolving *NoSQL movement* is now briefly introduced in Chapter 1, while the main, expanded discussion of this material is in Chapter 8 and the new Appendix K, “Big Data” Big Data remains the theme for Chapter 8, which includes the material on the development of nonrelational data stores (such as Cassandra and HBase) and the Hadoop Distributed File System (HDFS) first introduced in DBC e06. Appendix K, “Big Data”—has been added to provide additional, more detailed material on the NoSQL databases used in Big Data setups for those instructors who still want to cover this topic in depth.
- The Oracle NetBeans IDE is used in Chapter 7 in place of the Eclipse PDT IDE. Since an integrated downloadable installation file for both the Java JDK and NetBeans is available, this results in a much easier software installation of these products in Appendix I. The NetBeans IDE is also arguably easier to use than the Eclipse PDT, which will make it easier for students to create the HTML and PHP Web pages used in Chapter 7.

We kept all the main innovations included in DBC e06, including:

- The coverage of Web database applications in Chapter 7 now includes data input Web form pages. This allows Web database applications to be built with both data-input and data-reading Web pages.
- The coverage of Microsoft Access 2013 now includes Microsoft Access switchboard forms (covered in Appendix H, “The Access Workbench—Section H—Microsoft Access 2013 Switchboards”), which are used to build menus for database applications. Switchboard forms can be used to build database applications that have a user-friendly main menu that users can use to display forms, print reports, and run queries.
- 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.
- Support for Oracle Database Express Edition 11g Release 2 is maintained. Appendix B, “Getting Started with Oracle Database Express Edition 11g Release 2,” shows you how to use the product and the Oracle SQL Developer GUI utility. The appendix provides the basic knowledge, and Oracle SQL Developer screenshots in the text are used to illustrate specific concepts.

THE NEED FOR ESSENTIAL CONCEPTS

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 believe that the following database concepts are essential:

- Fundamentals of the relational model
- Structured Query Language (SQL)
- Data modeling
- Database design
- Database administration

And because of the increasing use of the Internet, the World Wide Web, commonly available analysis tools, and the emergence of the NoSQL movement, four more essential concepts need to be added to the list:

- Web database processing
- Data warehouse structures
- Business intelligence (BI) systems
- Nonrelational structured data storage

Users like Colin—and students who will perform jobs similar to his—need not learn these topics to the same depth as future information systems professionals. Consequently, this textbook presents only essential concepts—those that are necessary for users like Colin who want to create and use small databases. Many of the discussions in this book are rewritten and simplified explanations of topics that you will find fully discussed in David M. Kroenke and David J. Auer’s *Database Processing: Fundamentals, Design, and Implementation*.¹ However, in creating the material for this text, we have endeavored to ensure that the discussions remain accurate and do not mislead. Nothing here will need to be unlearned if students take more advanced database courses.

TEACHING CONCEPTS INDEPENDENT OF DBMS PRODUCTS

This book does not assume that students will use any particular DBMS product. The book does illustrate database concepts with Microsoft Access, SQL Server Express edition, Oracle Database Express Edition, and MySQL Community Server so that students can use these products as tools and actually try out the material, but all the concepts are presented in a DBMS-agnostic manner. When students learn the material this way, they come to understand that the fundamentals pertain to any database, from the smallest Microsoft Access database to the largest Microsoft SQL Server or Oracle Database database. Moreover, this approach avoids a common pitfall. When concepts and products are taught at the same time, students frequently confound concepts with product features and functions. For example, consider referential integrity constraints. When they are taught from a conceptual standpoint, students learn that there are times when the values of a column in one table must always be present as values of a column in a second table. Students also learn how this constraint arises in the context of relationship definition and how either the DBMS or the application must enforce this constraint. If taught in the context of a DBMS—say in the context of Microsoft Access—students will only learn that in some cases you check a check box and in other cases you do not. The danger is that the underlying concept will be lost in the product feature.

All this is not to say that a DBMS should not be used in this class. On the contrary, students can best master these concepts by applying them using a commercial DBMS product. This edition of the book was written to include enough basic information about Microsoft Access, SQL Server Express edition, Oracle Database Express Edition, and MySQL so that you can use these products in your class without the need for a second book or other materials. Microsoft Access is covered in some depth because of its popularity as a personal database and its inclusion in the Microsoft Office Professional suite of applications. However, if you want to cover a particular DBMS in depth or use a DBMS product not discussed in the book, you need to supplement this book with another text or additional materials. Pearson provides a number of books for Microsoft Access 2013 and other DBMS products, and many of them can be packaged with this text.

¹David M. Kroenke and David J. Auer, *Database Processing: Fundamentals, Design, and Implementation*, 13th edition (Upper Saddle River, NJ: Pearson/Prentice Hall, 2014).

THE ACCESS WORKBENCH

This new edition of the text continues using “The Access Workbench,” a feature first introduced in the third edition. Because Microsoft Access is widely used in introductory database classes, we feel it is important to include specific information on using Microsoft Access. Each chapter has an accompanying section of “The Access Workbench,” which illustrates the chapter’s concepts and techniques using Microsoft Access. “The Access Workbench” topics start with creating a database and a single table in Chapter 1 and move through various topics, finishing with Web database processing against a Microsoft Access database in Chapter 7 and using Microsoft Access (together with Microsoft Excel) to produce PivotTable OLAP reports in Chapter 8. This material is not intended to provide comprehensive coverage of Microsoft Access, but all the necessary basic Microsoft Access topics are covered so that your students can learn to effectively build and use Microsoft Access databases.

KEY TERMS, REVIEW QUESTIONS, EXERCISES, CASES, AND PROJECTS

Because it is important for students to apply the concepts they learn, each chapter concludes with sets of key terms, review questions, exercises (including exercises tied to “The Access Workbench”), Case Question sets, and three projects that run throughout the book. Students should know the meaning of each of the key terms and be able to answer the review questions if they have read and understood the chapter material. Each of the exercises requires students to apply the chapter concepts to a small problem or task.

The first of the projects, Garden Glory, concerns the development and use of a database for a partnership that provides gardening and yard maintenance services to individuals and organizations. The second project, James River Jewelry, addresses the need for a database to support a frequent-buyer program for a retail store. The third project, The Queen Anne Curiosity Shop, concerns the sales and inventory needs of a retail business. These three projects appear in all of the book’s chapters (although the actual text of the James River Jewelry project is found in online Appendix D). In each instance, students are asked to apply the project concepts from the chapter. Instructors will find more information on the use of these projects in the instructor’s manual and can obtain databases and data from the password-protected instructor’s portion of this book’s Web site (www.pearsonhighered.com/kroenke).

SOFTWARE USED IN THE BOOK

Just as we have treated our discussions in a DBMS-agnostic way, whenever possible, we have selected software to be as operating system independent as possible. It is amazing how much excellent software is available online. Many major DBMS vendors provide free versions of their premier products (for example, Microsoft’s SQL Server Express edition, Oracle Corporation’s Oracle Database Express Edition, and MySQL Community Server). Web editors and integrated development environments (IDEs) are also available (for example, Eclipse, NetBeans, and Visual Studio Express edition). PHP, considered the fourth most commonly used programming language, is downloadable for use with many operating systems and Web servers.

So although the examples in this book were created using a Microsoft operating system, SQL Server 2014 Express edition, Microsoft Access 2013, Microsoft Excel 2013, and the IIS Web Server, most of them could just as easily be accomplished using Linux, MySQL Server Community edition, Apache OpenOffice Base, Apache OpenOffice Calc, and the Apache Web server. Some software products used in the book, such as PHP and NetBeans, are available for multiple operating systems.

Over the past 30-plus years, we have found the development of databases and database applications to be an enjoyable and rewarding activity. We believe that the number, size, and importance of databases will increase in the future and that the field will achieve even greater prominence. It is our hope that the concepts, knowledge, and techniques presented in this book will help students to participate successfully in database projects now and for many years to come.

CHANGES FROM THE SIXTH EDITION

The most significant change in this edition is the coverage of the rapidly evolving use of *Big Data* and the associated *NoSQL movement*. The need to be able to store and process extremely large datasets is transforming the database world. Although these developments leave the database fundamentals covered in this book unchanged, they do require us to put the relational databases that are the core of this text into the context of the overall database picture and to provide the reader with an understanding of the nonrelational structured storage used in the Big Data environment. Therefore, Chapter 8 is now organized around the topic of Big Data, and the topics of data warehouses, clustered database servers, distributed databases, and an introduction to business intelligence (BI) systems find a natural home in that chapter. To provide additional coverage of Big Data, Appendix K has been added to allow a discussion in more depth than the page limitations of the book allow. For those wanting the same coverage of BI found in the previous edition of *Database Concepts*, we have moved BI material that no longer fit in Chapter 8 to Appendix J.

Finally, we have maintained the chapter-independent Case Question sets we added in the sixth edition. Although the chapter projects tie the topics in each chapter together, the case questions do not require the student to have completed work on the same case in a previous chapter or chapters. There is one intentional exception that spans Chapters 4 and 5 that ties data modeling and database design together, but each of these chapters also includes a standalone case. 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, and we provide needed Microsoft Access 2013 database and SQL scripts at the text Web site at www.pearsonhighered.com/kroenke.

We have, of course, also updated information on all the other products in the book. In particular, we cover the newly released Microsoft SQL Server 2014 and MySQL 5.6 Community Server.

We have kept and improved upon several features introduced in earlier editions of the book:

- The use of “The Access Workbench” sections in each chapter to provide coverage of Microsoft Access fundamentals now includes Microsoft Access switchboards (Appendix H, “The Access Workbench—Section H—Microsoft Access 2013 Switchboards,” available online).
- Introductions to the use of Microsoft SQL Server 2014 Express Edition (Appendix A, “Getting Started with Microsoft SQL Server 2014 Express Edition,” available online), Oracle Database Express Edition 11g Release 2 (Appendix B, “Getting Started with Oracle Database Express Edition 11g Release 2,” available online) and Oracle MySQL 5.6 Community Server (Appendix C, “Getting Started with MySQL 5.6 Community Server,” available online).
- The use of fully developed datasets for the three example databases that run throughout various portions of the book—Wedgewood Pacific Corporation, Heather Sweeney Designs, and Wallingford Motors.
- The use of the PHP scripting language, now used in the NetBeans IDE, in the Web database processing topics now includes code for Web page input forms.
- Coverage of the dimensional database model is maintained in the restructured Chapter 8, together with coverage of OLAP.

In order to make room for this new material, we have had to move some valuable material previously found in the book itself to online appendices. This includes the James River Jewelry set of project questions, which is now in online Appendix D, “James River Jewelry Project Questions,” the material on SQL Views is now in online Appendix E, “SQL Views, SQL/PSM, and Importing Data” with additional material on SQL Persistent Stored Modules (SQL/PSM) and how to import Microsoft Excel data. The business intelligence systems material on reporting systems and data mining is now in online Appendix J “Business Intelligence Systems.”

BOOK OVERVIEW

This textbook consists of 8 chapters and 11 appendices (all of which are readily available online at www.pearsonhighered.com/kroenke). Chapter 1 explains why databases are used, what their components are, and how they are developed. Students will learn the purpose of databases and their applications, as well as how databases differ from and improve on lists in spreadsheets. Chapter 2 introduces the relational model and defines basic relational terminology. It also introduces the fundamental ideas that underlie normalization and describe the normalization process.

Chapter 3 presents fundamental SQL statements. Basic SQL statements for data definition are described, as are SQL SELECT and data modification statements. No attempt is made to present advanced SQL statements; only the essential statements are described. Online Appendix E adds coverage of SQL views.

The next two chapters consider database design. Chapter 4 addresses data modeling, using the entity-relationship (E-R) model. This chapter describes the need for data modeling, introduces basic E-R terms and concepts, and presents a short case application (Heather Sweeney Designs) of E-R modeling. Chapter 5 describes database design and explains the essentials of normalization. The data model from the case example in Chapter 4 is transformed into a relational design in Chapter 5.

In this edition, we continue to use the more effective discussion of normalization added in an earlier edition. We have presented a prescriptive procedure for normalizing relations through the use of a four-step process. This approach not only makes the normalization task easier, it also makes normalization principles easier to understand. Therefore, this approach has been retained in this edition. For instructors who want a bit more detail on normal forms, short definitions of most normal forms are included in Chapter 5.

The last three chapters consider database management and the uses of databases in applications. Chapter 6 provides an overview of database administration. The case example database is built as a functioning database, and it serves as the example for a discussion of the need for database administration. The chapter surveys concurrency control, security, and backup and recovery techniques. Database administration is an important topic because it applies to all databases, even personal, single-user databases. In fact, in some ways this topic is more important for those smaller databases because no professional database administrator is present to ensure that critical tasks are performed.

Chapter 7 introduces the use of Web-based database processing, including a discussion of Open Database Connectivity (ODBC) and the use of the PHP scripting language. It also discusses the emergence and basic concepts of Extensible Markup Language (XML).

Chapter 8 discusses the emerging world of Big Data and the NoSQL movement. Business intelligence (BI) systems and the data warehouse architectures that support them are discussed, but many details of BI systems have been moved to online Appendix J. Chapter 8 also discusses dimensional databases. The chapter also walks through how to build a dimensional database for Heather Sweeney Designs and then use it to produce a PivotTable Online Analytical Processing (OLAP) report. Chapter 8 also provides a discussion of distributed databases and object-relational databases.

Appendix A provides a short introduction to Microsoft SQL Server 2014 Express Edition, Appendix B provides an introduction for Oracle Database Express Edition 11g

Release 2, and Appendix C provides a similar introduction for MySQL 5.6 Community Server. Microsoft Access is covered in “The Access Workbench” sections included in each chapter. Appendix D now contains the James River Jewelry project questions, and the material on SQL views is located in Appendix E. Appendix F provides an introduction to systems analysis and design and can be used to provide context for Chapter 4 (data modeling) and Chapter 5 (database design)—although in this book we focus on databases, databases are used in applications. Appendix F describes the application development process in more detail. Appendix G is a short introduction to Microsoft Visio 2013, which can be used as a tool for data modeling (Chapter 4) and database design (Chapter 5). Another useful database design tool is the MySQL Workbench, and this use of the MySQL Workbench is discussed in Appendix C. Appendix H extends Chapter 5’s section of “The Access Workbench” by providing coverage of Microsoft Access 2013 switchboards. Appendix I provides detailed support for Chapter 7 by giving detailed instructions on getting the Microsoft IIS Web server, PHP, and the NetBeans IDE up and running. Appendix J provides additional material on business intelligence (BI) systems to supplement and support Chapter 8 by giving details on report systems and data mining. Finally, Appendix K provides additional material on Big Data and NoSQL databases to supplement and support Chapter 8.

KEEPING CURRENT IN A RAPIDLY CHANGING WORLD

In order to keep *Database Concepts* up to date between editions, we post updates on the book’s Web site at www.pearsonhighered.com/kroenke as needed. Instructor resources and student materials are also available on the site, so be sure to check it from time to time.

ACKNOWLEDGMENTS

We would like to thank the following reviewers for their insightful and helpful comments:

Namjoo Choi, University of Kentucky
 David Chou, Eastern Michigan University
 Geoffrey Decker, Northern Illinois University
 Deena Engel, New York University
 Marni Ferner, University of North Carolina, Wilmington
 Jean Hendrix, University of Arkansas at Monticello
 Malini Krishnamurthi, California State University, Fullerton
 Rashmi Malhotra, Saint Joseph’s University
 Gabriel M. Petersen, North Carolina Central University
 Eliot Rich, University at Albany, State University of New York
 Liz Thiry, Pennsylvania State University
 Bond Wetherbe, Texas Tech University
 Diana Wolfe, Oklahoma State University–Oklahoma City

We would like to thank Nicole Sam, our editor; Denise Vaughn, our program manager; and Ilene Kahn, our project manager, for their professionalism, insight, support, and assistance in the development of this project. We would also like to thank Robert Mills, Robert Yoder, and Scott Vandenberg for their 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 entered the computing profession as a summer intern at the RAND Corporation in 1967. Since then, his career has spanned education, industry, consulting, and publishing.

He has taught at the University of Washington, Colorado State University, and Seattle University. Over the years, he has led dozens of teaching seminars for college professors. In 1991 the International Association of Information Systems named him Computer Educator of the Year.

In industry, Kroenke has worked for the U.S. Air Force and Boeing Computer Services, and he was a principal in the startup of three companies. He was also vice president of product marketing and development for the Microrim Corporation and was chief technologist for the database division of Wall Data, Inc. He is the father of the semantic object data model. Kroenke's consulting clients include IBM Corporation, Microsoft, Computer Sciences Corporation, and numerous other companies and organizations.

His text *Database Processing: Fundamentals, Design, and Implementation*, first published in 1977, is now in its 13th Edition (coauthored with David Auer for the 11th, 12th, and 13th editions). He introduced *Database Concepts* (now in the seventh edition that you are reading) in 2003. Kroenke has published many other textbooks, including the classic *Business Computer Systems* (1981). Recently, he has authored *Using MIS* (7th Edition), *Experiencing MIS* (5th Edition), *MIS Essentials* (4th Edition), *Processes, Systems and Information: An Introduction to MIS* (2nd Edition) (coauthored with Earl McKinney), and *Essentials of Processes, Systems and Information* (coauthored with Earl McKinney).

An avid sailor, Kroenke also wrote *Know Your Boat: The Guide to Everything That Makes Your Boat Work*. Kroenke lives in Seattle, Washington. He is married and has two children and three grandchildren.

Since 1994, **David J. Auer** has been the director of Information Systems and Technology Services at Western Washington University's College of Business and Economics (CBE) and a lecturer in CBE's Department of Decision Sciences. Since 1981, he has taught CBE courses in quantitative methods, production and operations management, statistics, finance, and management information systems. Besides managing CBE's computer, network, and other technology resources, he also teaches management information systems courses. He has taught the Principles of Management Information Systems and Business Database Development courses, and he was responsible for developing CBE's network infrastructure courses, including Computer Hardware and Operating Systems, Telecommunications, and Network Administration. He has coauthored several MIS-related textbooks.

Auer holds a bachelor's degree in English literature from the University of Washington, a bachelor's degree in mathematics and economics from Western Washington University, a master's degree in economics from Western Washington University, and a master's degree in counseling psychology from Western Washington University. He served as a commissioned officer in the U.S. Air Force, and he has also worked as an organizational development specialist and therapist for an employee assistance program (EAP).

Auer and his wife, Donna, live in Bellingham, Washington. He has two children and four grandchildren.



PART 1

Database Fundamentals

Part I introduces fundamental concepts and techniques of relational database management. Chapter 1 explains database technology, discusses why databases are used, and describes the components of a database system. Chapter 2 introduces the relational model and defines key relational database terms. It also presents basic principles of relational database design. Chapter 3 presents Structured Query Language, an international standard for creating and processing relational databases.

After you have learned these fundamental database concepts, we will focus on database modeling, design, and implementation in Part II. Finally, we will discuss database management, Web database applications, Big Data, and business intelligence (BI) systems in Part III.

This page intentionally left blank

CHAPTER 1

Getting Started

CHAPTER OBJECTIVES

- Identify the purpose and scope of this book
- Know the potential problems with lists
- Understand the reasons for using a database
- Understand how using related tables helps you avoid the problems of using lists
- Know the components of a database system
- Learn the elements of a database
- Learn the purpose of a database management system (DBMS)
- Understand the functions of a database application
- Introduce nonrelational databases

Knowledge of database technology increases in importance every day. Databases are used everywhere: They are key components of e-commerce and other Web-based applications. They lay at the heart of organization-wide operational and decision support applications. Databases are also used by thousands of work groups and millions of individuals. It is estimated that there are more than 10 million active databases in the world today.

The purpose of this book is to teach you the essential relational database concepts, technology, and techniques that you need to begin a career as a database developer. This book does not teach everything of importance in relational database technology, but it will give you sufficient background to be able to create your own personal databases and to participate as a member of a team in the development of larger, more complicated databases. You will also be able to ask the right questions to learn more on your own.

In this first chapter, we investigate the reasons for using a relational database. We begin by describing some of the problems that can occur when using lists. Using a series of examples, we illustrate how using sets of related tables helps you to avoid those problems. Next, we describe the components of a database system and explain the elements of a database, the purpose of the database management system (DBMS), and the functions of a database application. Finally, we introduce nonrelational databases.

WHY USE A DATABASE?

A database is used to help people keep track of things. You might wonder why we need a special term (and course) for such technology when a simple **list** could serve the same purpose. Many people do keep track of things by using lists, and sometimes such lists are valuable. In other cases, however, simple lists lead to data inconsistencies and other problems.

In this section, we examine several different lists and show some of these problems. As you will see, we can solve the problems by splitting lists into tables of data. Such tables are the key components of a database. A majority of this text concerns the design of such tables and techniques for manipulating the data they contain.

Problems with Lists

Figure 1-1 shows a simple list of student data, named the Student List,¹ stored in a spreadsheet. The Student List is a very simple list, and for such a list a spreadsheet works quite well. Even if the list is long, you can sort it alphabetically by last name, first name, or email address to find any entry you want. You can change the data values, add data for a new student, or delete student data. With a list like the Student List in Figure 1-1, none of these actions is problematic, and a database is unnecessary. Keeping this list in a spreadsheet is just fine.

Suppose, however, we change the Student List by adding adviser data, as shown in Figure 1-2. You can still sort the new Student with Adviser List in a number of ways to find an entry, but making changes to this list causes **modification problems**. Suppose, for example, that you want to delete the data for the student Chip Marino. As shown in Figure 1-3, if you delete the seventh row you not only remove Chip Marino's data, you also remove the fact that there is an adviser named Tran and that Professor Tran's email address is Ken.Tran@ourcampus.edu.

Similarly, updating a value in this list can have unintended consequences. If, for example, you change AdviserEmail in the eighth row, you will have inconsistent data. After the change, the fifth row indicates one email address for Professor Taing, and the eighth row indicates a different email address for the same professor. Or is it the same professor? From this list, we cannot tell if there is one Professor Taing with two inconsistent email addresses or whether there are two professors named Taing with different email addresses. By making this update, we add confusion and uncertainty to the list.

FIGURE 1-1

The Student List in a Spreadsheet

	A	B	C
1	LastName	FirstName	Email
2	Andrews	Matthew	Matthew.Andrews@ourcampus.edu
3	Brisbon	Lisa	Lisa.Brisbon@ourcampus.edu
4	Fischer	Douglas	Douglas.Fischer@ourcampus.edu
5	Hwang	Terry	Terry.Hwang@ourcampus.edu
6	Lai	Tzu	Tzu.Lai@ourcampus.edu
7	Marino	Chip	Chip.Marino@ourcampus.edu
8	Thompson	James	James.Thompson@ourcampus.edu

FIGURE 1-2

The Student with Adviser List

	A	B	C	D	E
1	LastName	FirstName	Email	Adviser.LastName	AdviserEmail
2	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
3	Brisbon	Lisa	Lisa.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu
4	Fischer	Douglas	Douglas.Fischer@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
5	Hwang	Terry	Terry.Hwang@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu
6	Lai	Tzu	Tzu.Lai@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu
7	Marino	Chip	Chip.Marino@ourcampus.edu	Tran	Ken.Tran@ourcampus.edu
8	Thompson	James	James.Thompson@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu

¹In order to easily identify and reference the lists being discussed, we capitalize the first letter of each word in the list names in this chapter. Similarly, we capitalize the names of the database tables associated with the lists.

FIGURE 1-3

Modification Problems in the Student with Adviser List

	A	B	C	D	E
1	LastName	FirstName	Email	AdviserLastName	AdviserEmail
2	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
3	Brisbon	Lisa	Lisa.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu
4	Fischer	Douglas	Douglas.Fischer@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
5	Hwang	Terry	Terry.Hwang@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu
6	Lai	Tzu	Tzu.Lai@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu
7	Marino	Chip	Chip.Marino@ourcampus.edu	Tran	Ken.Tran@ourcampus.edu
8	Thompson	James	James.Thompson@ourcampus.edu	Taing	Sue.Taing@ourcampus.edu
9	???	???	???	Green	George.Green@ourcampus.edu

Deleted row—too much data lost

Changed row—inconsistent data

Inserted row—data missing

Finally, what do we do if we want to add data for a professor who has no advisees? For example, Professor George Green has no advisees, but we still want to record his email address. As shown in Figure 1-3, we must insert a row with incomplete values, called **null values**, in the database field. In this case, the term *null value* means a missing value, but there are other meanings of the term *null value* that are used when working with databases. We will discuss the problems of null values in detail in the next chapter, where we will show that null values are always problematic and that we want to avoid them whenever possible.

Now, what exactly happened in these two examples? We had a simple list with three columns, added two more columns to it, and thereby created several problems. The problem is not just that the list has five columns instead of three. Consider a different list that has five columns: the Student with Residence List shown in Figure 1-4. This list has five columns, yet it suffers from none of the problems of the Student with Adviser List in Figure 1-3.

In the Student with Residence List in Figure 1-4, we can delete the data for student Chip Marino and lose only data for that student. No unintended consequences occur. Similarly, we can change the value of Residence for student Tzu Lai without introducing any inconsistency. Finally, we can add data for student Garret Ingram and not have any null values.

An essential difference exists between the Student with Adviser List in Figure 1-3 and the Student with Residence List in Figure 1-4. Looking at those two figures, can you determine the difference? The essential difference is that the Student with Residence List in Figure 1-4 is all about a *single thing*: All the data in that list concern *students*. In contrast, the Student with Adviser List in Figure 1-3 is about *two things*: Some of the data concern *students* and some of the data concern *advisers*. In general, whenever a list has data about two or more different things modification problems will result.

To reinforce this idea, examine the Student with Adviser and Department List in Figure 1-5. This list has data about three different things: *students*, *advisers*, and *departments*. As you can see in the figure, the problems with inserting, updating, and deleting data just get worse. A change in the value of AdviserLastName, for example, might

FIGURE 1-4

The Student with Residence List

	A	B	C	D	E
1	LastName	FirstName	Email	Phone	Residence
2	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	301-555-2225	123 15th St Apt 21
3	Brisbon	Lisa	Lisa.Brisbon@ourcampus.edu	301-555-2241	Dorsett Room 201
4	Fischer	Douglas	Douglas.Fischer@ourcampus.edu	301-555-2257	McKinley Room 109
5	Hwang	Terry	Terry.Hwang@ourcampus.edu	301-555-2229	McKinley Room 208
6	Ingram	Garrett	Garett.Ingram@ourcampus.edu	301-555-2223	Dorsett Room 218
7	Lai	Tzu	Tzu.Lai@ourcampus.edu	301-555-2231	McKinley Room 115
8	Marino	Chip	Chip.Marino@ourcampus.edu	301-555-2243	234 16th St Apt 32
9	Thompson	James	James.Thompson@ourcampus.edu	301-555-2245	345 17th St Apt 43

Inserted row—data OK

Changed row—no inconsistent data

Deleted row—no data loss

FIGURE 1-5

The Student with Adviser and Department List

If Adviser **Baker** is changed to **Taing**, we need to change *AdviserEmail* as well. If changed to **Valdez**, we need to change *AdviserEmail*, *Department*, and *AdminLastName*.

	A	B	C	D	E	F	G
1	LastName	FirstName	Email	AdviserLastName	AdviserEmail	Department	AdminLastName
2	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
3	Brisbon	Lisa	Lisa.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu	Chemistry	Chaplin
4	Fischer	Douglas	Douglas.Fischer@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
5	Hwang	Terry	Terry.Hwang@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu	Accounting	Smith
6	Lai	Tzu	Tzu.Lai@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu	Chemistry	Chaplin
7	Marino	Chip	Chip.Marino@ourcampus.edu	Tran	Ken.Tran@ourcampus.edu	InfoSystems	Rogers
8	Thompson	James	James.Thompson@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu	Accounting	Smith
9	???	???	???	???	???	Biology	Kelly

Deleted row—Student, Adviser, and Department data lost

Inserted row—both Student and Adviser data missing

necessitate a change in only *AdviserEmail*, or it might require a change in *AdviserEmail*, *Department*, and *AdminLastName*. As you can imagine, if this list is long—for example, if the list thousands of rows—and if several people process it, the list will be a mess in a very short time.

Using Relational Database Tables

The problems of using lists were first identified in the 1960s, and a number of different techniques were developed to solve them. Over time, a methodology called the **relational model** emerged as the leading solution, and today almost every commercial database is based on the relational model. We will examine the relational model in detail in Chapter 2. Here, however, we introduce the basic ideas of the relational model by showing how it solves the modification problems of lists.

Remember your eighth-grade English teacher? He or she said that a paragraph should have a single theme. If you have a paragraph with more than one theme, you need to break it up into two or more paragraphs, each with a *single theme*. That idea is the foundation of the design of relational databases. A **relational database** contains a collection of separate tables. A **table** holds data about one and only one theme in most circumstances. If a table has two or more themes, we break it up into two or more tables.

BTW

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 2. For now, the main differences you 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.

A Relational Design for the Student with Adviser List The Student with Adviser List in Figure 1-2 has two themes: *students* and *advisers*. If we put this data into a relational database, we place the student data in one table named STUDENT and the adviser data in a second table named ADVISER.

BTW

In this book, table names appear in all capital, or uppercase, letters (STUDENT, ADVISER). Column names have initial capitals (Phone, Address), and where column names consist of more than one word the initial letter of each word is capitalized (LastName, AdviserEmail).

We still want to show which students have which advisers, however, so we leave AdviserLastName in the ADVISER table. As shown in Figure 1-6, the values of AdviserLastName now let us link rows in the two tables to each other.

Now consider possible modifications to these tables. As you saw in the last section, three basic **modification actions** are possible: **insert**, **update**, and **delete**. To evaluate a design, we need to consider each of these three actions. As shown in Figure 1-7, we can insert, update, and delete in these tables with no modification problems.

For example, we can insert the data for Professor Bill Yeats by just adding his data to the ADVISER table. No student references Professor Yeats, but this is not a problem. Perhaps a student will have Professor Yeats as an adviser in the future. We can also update data values without unintended consequences. The email address for Professor Susan Taing can be changed to Sue.Taing@ourcampus.edu, and no inconsistent data will result because Professor Taing's email address is stored just once in the ADVISER table. Finally, we can delete data without unintended consequences. For example, if we delete the data for student Chip Marino from the STUDENT table, we lose no adviser data.

A Relational Design for the Student with Adviser and Department List We can use a similar strategy to develop a relational database for the Student with Adviser and

FIGURE 1-6

The Adviser and Student Tables

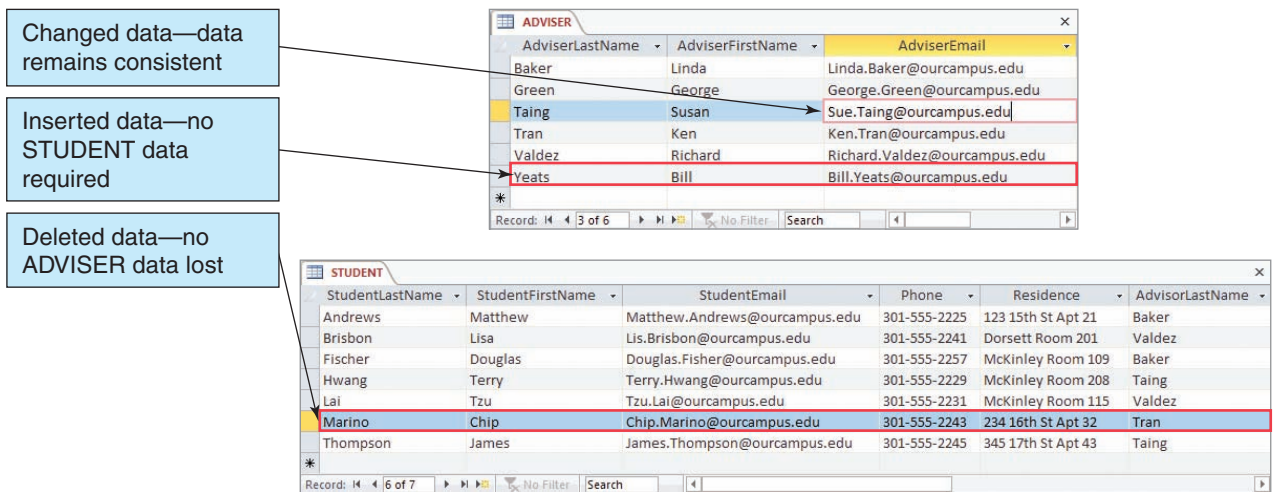
STUDENT data linked to ADVISER data via AdviserLastName

AdviserLastName	AdviserFirstName	AdviserEmail
Baker	Linda	Linda.Baker@ourcampus.edu
Green	George	George.Green@ourcampus.edu
Taing	Susan	Susan.Taing@ourcampus.edu
Tran	Ken	Ken.Tran@ourcampus.edu
Valdez	Richard	Richard.Valdez@ourcampus.edu

StudentLastName	StudentFirstName	StudentEmail	Phone	Residence	AdviserLastName
Andrews	Matthew	Matthew.Andrews@ourcampus.edu	301-555-2225	123 15th St Apt 21	Baker
Brisbon	Lisa	Lis.Brisbon@ourcampus.edu	301-555-2241	Dorsett Room 201	Valdez
Fischer	Douglas	Douglas.Fisher@ourcampus.edu	301-555-2257	McKinley Room 109	Baker
Hwang	Terry	Terry.Hwang@ourcampus.edu	301-555-2229	McKinley Room 208	Taing
Lai	Tzu	Tzu.Lai@ourcampus.edu	301-555-2231	McKinley Room 115	Valdez
Marino	Chip	Chip.Marino@ourcampus.edu	301-555-2243	234 16th St Apt 32	Tran
Thompson	James	James.Thompson@ourcampus.edu	301-555-2245	345 17th St Apt 43	Taing

FIGURE 1-7

Modifying the Adviser and Student Tables

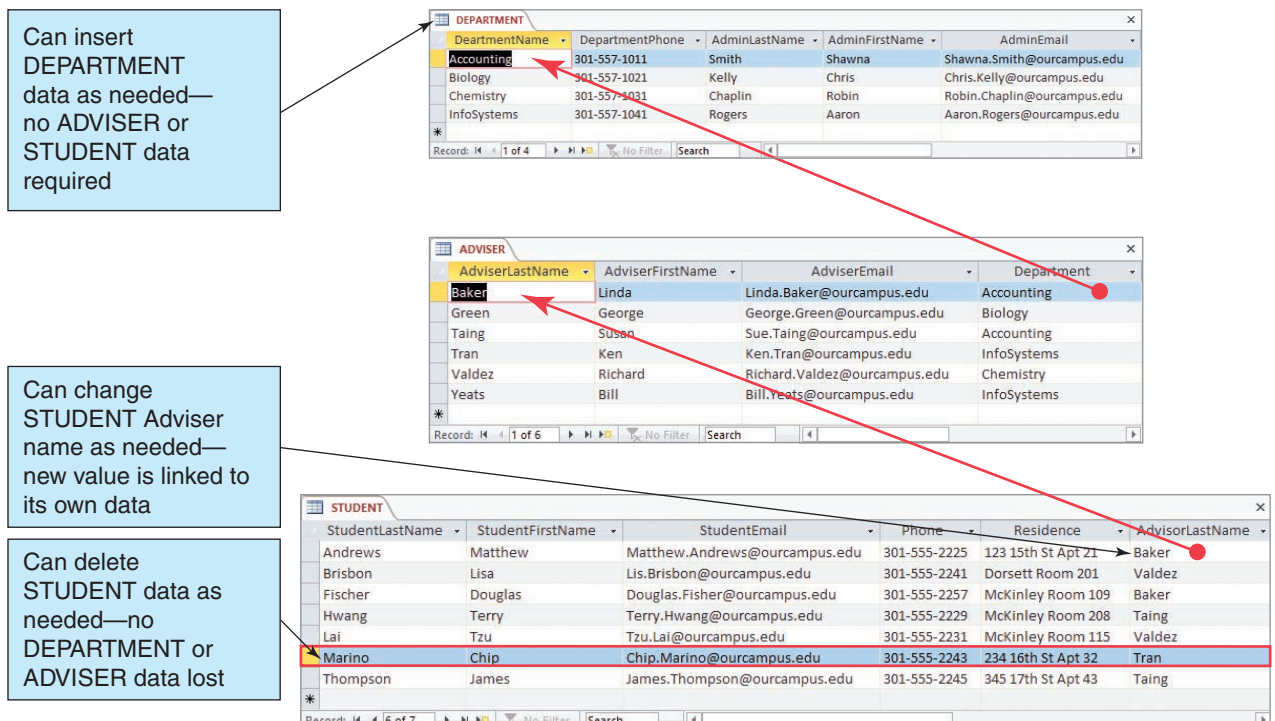


Department List shown in Figure 1-5. This list has three themes: *students*, *advisers*, and *departments*. Accordingly, we create three tables, one for each of these three themes, as shown in Figure 1-8.

As illustrated in Figure 1-8, we can use *AdviserLastName* and *Department* to link the tables. Also, as shown in this figure, this set of tables does not have any modification problems. We can insert new data without creating null values, we can modify data without

FIGURE 1-8

The Department, Adviser, and Student Tables



creating inconsistencies, and we can delete data without unintended consequences. Notice in particular that when we add a new row to DEPARTMENT we can add rows in ADVISER, if we want, and we can add rows in STUDENT for each of the new rows in ADVISER, if we want. However, all these actions are independent. None of them leaves the tables in an inconsistent state.

Similarly, when we modify an AdviserLastName in a row in STUDENT, we automatically pick up the adviser's correct first name, email address, and department. If we change AdviserLastName in the first row of STUDENT to Taing, it will be connected to the row in ADVISER that has the correct AdviserFirstName, AdviserEmail, and Department values. If we want, we can also use the value of Department in ADVISER to obtain the correct DEPARTMENT data. Finally, notice that we can delete the row for student Marino without a problem.

As an aside, the design in Figure 1-8 has removed the problems that occur when modifying a list, but it has also introduced a new problem. Specifically, what would happen if we deleted the first row in ADVISER? Students Andrews and Fischer would have an invalid value of AdviserLastName because Professor Baker would no longer exist in the ADVISER table. To prevent this problem, we can design the database so that a deletion of a row is not allowed if other rows depend on it, or we can design it so that the dependent rows are deleted as well. We are skipping way ahead here; however, we will discuss such issues in later chapters.

A Relational Design for Art Course Enrollments To fix in your mind the ideas we have been examining, consider the Art Course List in Figure 1-9, which is used by an art school that offers art courses to the public. This list has modification problems. For example, suppose we change the value of CourseDate in the first row. This change might mean that the date for the course is changing, in which case the CourseDate values should be changed in other rows as well. Alternatively, this change could mean that a new Advanced Pastels (Adv Pastels) course is being offered. Either is a possibility.

As with the previous examples, we can remove the problems and ambiguities by creating a separate table for each theme. However, in this case the themes are more difficult to determine. Clearly, one of the themes is *customer* and another one is *art course*. However, a third theme exists that is more difficult to bring to light. The customer has paid a certain amount toward a course. The amount paid is not a property of the customer because it varies depending on which course the customer is taking. For example, customer Ariel Johnson paid \$250 for the Advanced Pastels (Adv Pastels) course and \$350 for the Intermediate Pastels (Int Pastels) course. Similarly, the amount paid is not a property of the course because it varies with which customer has taken the course. Therefore, the third theme of this list must concern the *enrollment* of a particular student in a particular class. Figure 1-10 shows a design using three tables that correspond to these three themes—we name this set of three tables the *Art Course Database*.

Notice that the *Art Course Database* design assigns an **ID column** named CustomerNumber that assigns a unique identifying number to each row of CUSTOMER;

FIGURE 1-9

The Art Course List with Modification Problems

	A	B	C	D	E	F	G
1	CustomerLastName	CustomerFirstName	Phone	CourseDate	AmountPaid	Course	Fee
2	Johnson	Ariel	206-567-1234	10/1/2015	\$250.00	Adv Pastels	\$500.00
3	Green	Robin	425-678-8765	9/15/2015	\$350.00	Beg Oils	\$350.00
4	Jackson	Charles	360-789-3456	10/1/2015	\$500.00	Adv Pastels	\$500.00
5	Johnson	Ariel	206-567-1234	3/15/2015	\$350.00	Int Pastels	\$350.00
6	Pearson	Jeffery	206-567-2345	10/1/2015	\$500.00	Adv Pastels	\$500.00
7	Sears	Miguel	360-789-4567	9/15/2015	\$350.00	Beg Oils	\$350.00
8	Kyle	Leah	425-678-7654	11/15/2015	\$250.00	Adv Pastels	\$500.00
9	Myers	Lynda	360-789-5678	10/15/2015	\$0.00	Beg Oils	\$350.00

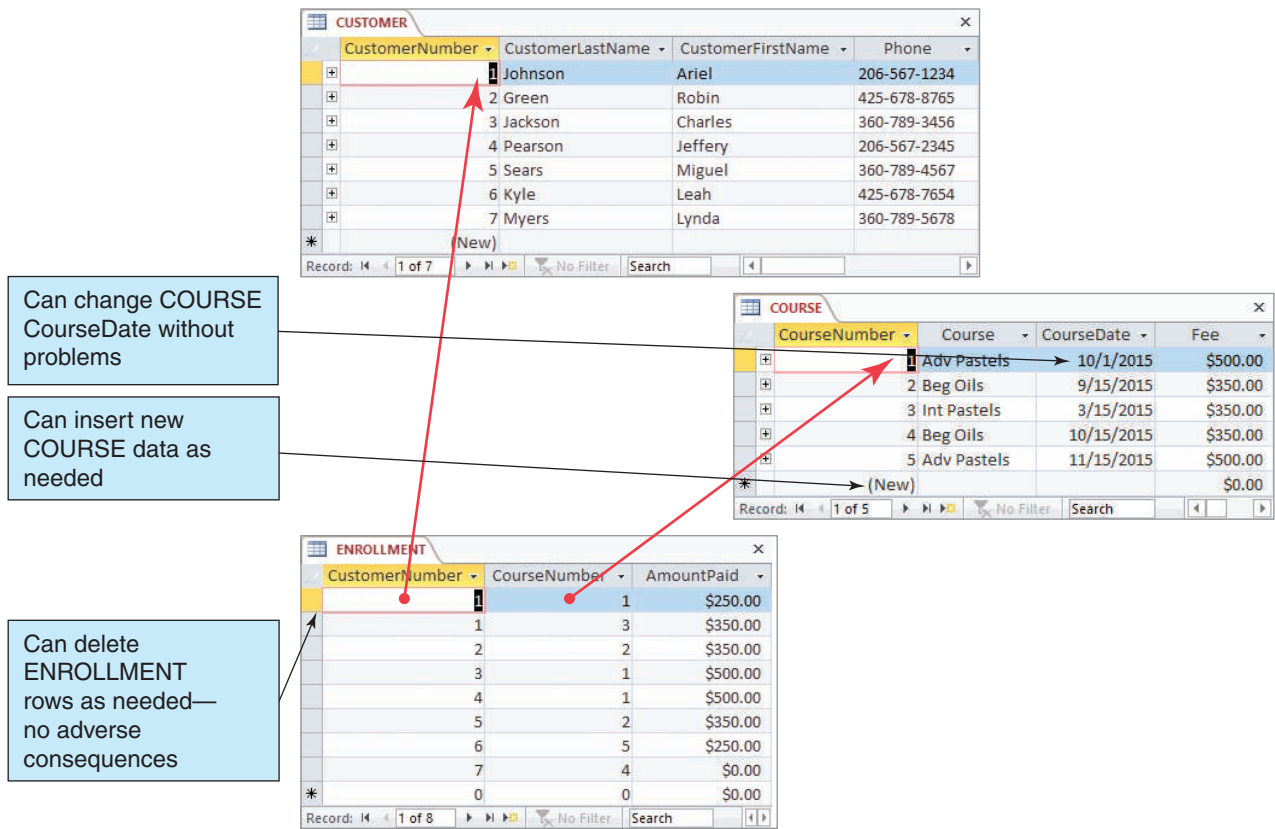
How to enter the fee for a new course?

Consequences of changing this date?

Consequences of deleting this row?

FIGURE 1-10

The Art Course Database Tables



this is necessary because some customers might have the same name. Another ID column, named CourseNumber, has also been added to COURSE. This is necessary because some courses have the same name. Finally, notice that the rows of the ENROLLMENT table show the amount paid by a particular customer for a particular course and that the ID columns CustomerNumber and CourseNumber are used as linking columns to the other tables.

A Relational Design for Parts and Prices Now let's consider a more complicated example. Figure 1-11 shows a spreadsheet that holds the Project Equipment List used by a housing contractor named Carbon River Construction to keep track of the parts that it buys for various construction projects.

The first problem with this list concerns modifications to the existing data. Suppose your job is to maintain the Project Equipment List, and your boss tells you that customer Elizabeth Barnaby changed her phone number. How many changes would you need to make to this spreadsheet? For the data in Figure 1-11, you would need to make this change 10 times. Now suppose the spreadsheet has 5,000 rows. How many changes might you need to make? The answer could be dozens, and you need to worry not only about the time this will take but also about the possibility of errors—you might miss her name in a row or two and fail to properly update her phone number in these rows.

Consider a second problem with this list. In this business, each supplier agrees to a particular discount for all parts it supplies. For example, in Figure 1-11 the supplier NW Electric has agreed to a 25 percent discount. With this list, every time you enter a new part quotation, you must enter the supplier of that part, along with the correct discount. If

FIGURE 1-11

The Project Equipment List as a Spreadsheet

ProjectName	OwnerContact	Phone	Category	Quantity	ItemDescription	UnitPrice	ExtendedPrice	Supplier	Discount
Highland House	Elizabeth Barnaby	555-444-8899	Electrical	1	200 Amp Panel	\$170.00	\$170.00	NW Electric	25.00%
Highland House	Elizabeth Barnaby	555-444-8899	Electrical	3	50 Watt Breaker	\$60.00	\$180.00	NW Electric	25.00%
Highland House	Elizabeth Barnaby	555-444-8899	Electrical	7	20 Watt Breaker	\$35.00	\$245.00	NW Electric	25.00%
Highland House	Elizabeth Barnaby	555-444-8899	Electrical	15	15 Watt Breaker	\$35.00	\$525.00	NW Electric	25.00%
Highland House	Elizabeth Barnaby	555-444-8899	Electrical	200	12 ga, 3 Wire, per foot	\$1.50	\$300.00	EB Supplies	15.00%
Highland House	Elizabeth Barnaby	555-444-8899	Electrical	300	12 ga, 3 Wire, per foot	\$1.25	\$375.00	EB Supplies	15.00%
Baker Remodel	John Stanley	555-787-8392	Exterior	35	Siding, 4x8 feet	\$22.50	\$787.50	Contractor, Inc.	35.00%
Highland House	Elizabeth Barnaby	555-444-8899	Electrical	10	15 Watt Breaker	\$35.00	\$350.00	EB Supplies	15.00%
Baker Remodel	John Stanley	555-787-8392	Exterior	28	1x4 - 8 feet	\$4.75	\$133.00	Contractor, Inc.	35.00%
Baker Remodel	John Stanley	555-787-8392	Exterior	100	Cedar Shingles, bundle	\$65.00	\$6,500.00	Contractor, Inc.	35.00%
Highland House	Elizabeth Barnaby	555-444-8899	Interior	15	Door	\$275.00	\$4,125.00	Interior, Inc.	15.00%
Highland House	Elizabeth Barnaby	555-444-8899	Interior	15	Door Hinge Set	\$29.95	\$449.25	Interior, Inc.	15.00%
Highland House	Elizabeth Barnaby	555-444-8899	Interior	15	Door Handle Set	\$52.50	\$787.50	Interior, Inc.	15.00%
Hew Remodel	Ralph Hew	555-298-4244	Electrical	1	Panel, 200 Amp	\$170.00	\$170.00	NW Electric	25.00%
Hew Remodel	Ralph Hew	555-298-4244	Electrical	2	50 Watt Breaker	\$60.00	\$120.00	NW Electric	25.00%
Hew Remodel	Ralph Hew	555-298-4244	Electrical	5	20 Watt Breaker	\$35.00	\$175.00	NW Electric	25.00%
Hew Remodel	Ralph Hew	555-298-4244	Electrical	20	15 Watt Breaker	\$35.00	\$700.00	NW Electric	25.00%
Hew Remodel	Ralph Hew	555-298-4244	Electrical	150	12 ga, 3 Wire, per foot	\$1.50	\$225.00	NW Electric	25.00%
Hew Remodel	Ralph Hew	555-298-4244	Electrical	300	14 ga, 3 Wire, per foot	\$1.25	\$375.00	NW Electric	25.00%

dozens or hundreds of suppliers are used, there is a chance that you will sometimes enter the wrong discount. If you do, the list will have more than one discount for one supplier—a situation that is incorrect and confusing.

A third problem occurs when you enter data correctly but inconsistently. The first row has a part named 200 Amp panel, whereas the 15th row has a part named Panel, 200 Amp. Are these two parts the same item, or are they different? It turns out that they are the same item, but they were named differently.

A fourth problem concerns partial data. Suppose you know that a supplier offers a 20 percent discount, but Carbon River has not yet ordered from the supplier. Where do you record the 20 percent discount?

Just as we did for the previous examples, we can fix the Project Equipment List by breaking it up into separate tables. Because this list is more complicated, we need to use more tables. When we analyze the Project Equipment List, we find data about four themes: *projects*, *items*, *price quotations*, and *suppliers*. Accordingly, we create a database with four tables and relate those four tables using linking values, as before. Figure 1-12 shows our four tables and their relationships—we will name this set of tables the *Project Equipment Database*.

In Figure 1-12, note that the QUOTE table holds a unique quote identifier (QuoteID), a quantity, a unit price, an extended price (which is equal to [quantity * unit price]), and three ID columns as linking values: ProjectID for PROJECT, ItemNumber for ITEM, and SupplierID for SUPPLIER.

Now if Elizabeth Barnaby changes her phone number we need to make that change only once—in the PROJECT table. Similarly, we need to record a supplier discount only once—in the SUPPLIER table.

Processing Relational Tables

By now, you may have a burning question: It may be fine to tear the lists up into pieces in order to eliminate processing problems, but what if the users want to view their data in the format of the original list? With the data separated into different tables, the users will have to jump from one table to another to find the information they want, and this jumping around will become tedious.

FIGURE 1-12

The Project Equipment Database Tables

The figure displays three database tables with their respective fields and data. Red arrows indicate relationships between fields in different tables.

PROJECT Table:

ProjectID	ProjectName	OwnerContact	Phone
1	Highland House	Elizabeth Barnaby	555-444-8899
2	Baker Remodel	John Stanley	555-787-8392
3	Hew Remodel	Ralph Hew	555-298-4244

ITEM Table:

ItemNumber	ItemDescription	Category
1100	200 Amp Panel	Electrical
1200	50 Watt Breaker	Electrical
1300	20 Watt Breaker	Electrical
1400	15 Watt Breaker	Electrical
1500	12 ga, 3 Wire, per foot	Electrical
1550	14 ga, 3 Wire, per foot	Electrical
1600	Siding, 4x8 feet	Exterior
1700	1x4 - 8 feet	Exterior
1800	Cedar Shingles, bundle	Exterior
2000	Door	Interior
2100	Door Hinge Set	Interior
2200	Door Handle Set	Interior

QUOTE Table:

QuoteID	ProjectID	ItemNumber	SupplierID	Quantity	UnitPrice	ExtendedPrice
1	1	1100	1	1	\$170.00	\$170.00
2	1	1200	1	3	\$60.00	\$180.00
3	1	1300	1	7	\$35.00	\$245.00
4	1	1400	1	15	\$35.00	\$525.00
5	1	1500	2	200	\$1.50	\$300.00
6	1	1500	2	300	\$1.25	\$375.00
7	2	1600	3	35	\$22.50	\$787.50
8	1	1400	2	10	\$35.00	\$350.00
9	2	1700	3	28	\$4.75	\$133.00
10	2	1800	3	100	\$65.00	\$6,500.00
11	1	2000	4	15	\$275.00	\$4,125.00
12	1	2100	4	15	\$29.95	\$449.25
13	1	2200	4	15	\$52.50	\$787.50
14	3	1100	1	1	\$170.00	\$170.00
15	3	1200	1	2	\$60.00	\$120.00
16	3	1300	1	5	\$35.00	\$175.00
17	3	1400	1	20	\$35.00	\$700.00
18	3	1500	1	150	\$1.50	\$225.00
19	3	1550	1	300	\$1.25	\$375.00

SUPPLIER Table:

SupplierID	Supplier	Discount
1	NW Electric	25.00%
2	EB Supplies	15.00%
3	Contractor, Inc.	35.00%
4	Interior, Inc.	15.00%

This is an important question and one that many people addressed in the 1970s and 1980s. Several approaches were invented for combining, querying, and processing sets of tables. Over time, one of those approaches, a language called **Structured Query Language (SQL)**, emerged as the leading technique for data definition and manipulation. Today, SQL is an international standard. Using SQL, you can reconstruct lists from their underlying tables; you can query for specific data conditions; you can perform computations on data in tables; and you can insert, update, and delete data.

Processing Tables by Using SQL You will learn how to code SQL statements in Chapter 3. However, to give you an idea of the structure of such statements, let's look at an SQL statement that joins the three tables in Figure 1-10 to produce the original Art Course List. Do not worry about understanding the syntax of this statement, just realize that it produces the result shown in Figure 1-13, which contains all the Art Course List data (although in a slightly different row order²).

```
SELECT CUSTOMER.CustomerLastName,
        CUSTOMER.CustomerFirstName, CUSTOMER.Phone,
        COURSE.CourseDate, ENROLLMENT.AmountPaid,
        COURSE.Course, COURSE.Fee
FROM    CUSTOMER, ENROLLMENT, COURSE
WHERE   CUSTOMER.CustomerNumber = ENROLLMENT.CustomerNumber
        AND COURSE.CourseNumber = ENROLLMENT.CourseNumber;
```

²We will discuss how to sort data to control the row order in Chapter 3.

FIGURE 1-13

Results of the SQL Query to Recreate the Art Course List

CustomerLastName	CustomerFirstName	Phone	CourseDate	AmountPaid	Course	Fee
Johnson	Ariel	206-567-1234	10/1/2015	\$250.00	Adv Pastels	\$500.00
Johnson	Ariel	206-567-1234	3/15/2015	\$350.00	Int Pastels	\$350.00
Green	Robin	425-678-8765	9/15/2015	\$350.00	Beg Oils	\$350.00
Jackson	Charles	360-789-3456	10/1/2015	\$500.00	Adv Pastels	\$500.00
Pearson	Jeffery	206-567-2345	10/1/2015	\$500.00	Adv Pastels	\$500.00
Sears	Miguel	360-789-4567	9/15/2015	\$350.00	Beg Oils	\$350.00
Kyle	Leah	425-678-7654	11/15/2015	\$250.00	Adv Pastels	\$500.00
Myers	Lynda	360-789-5678	10/15/2015	\$0.00	Beg Oils	\$350.00

As you will learn in Chapter 3, it is also possible to select rows, to order them, and to make calculations on row data values. Figure 1-14 shows the result of the SQL statement:

```
SELECT    CUSTOMER.CustomerLastName,
          CUSTOMER.CustomerFirstName, CUSTOMER.Phone,
          COURSE.Course, COURSE.CourseDate, COURSE.Fee,
          ENROLLMENT.AmountPaid,
          (COURSE.Fee-ENROLLMENT.AmountPaid) AS AmountDue
FROM      CUSTOMER, ENROLLMENT, CUSTOMER
WHERE     CUSTOMER.CustomerNumber = ENROLLMENT.CustomerNumber
          AND COURSE.CourseNumber = ENROLLMENT.CourseNumber
          AND (COURSE.Fee - ENROLLMENT.AmountPaid) > 0
ORDER BY  CUSTOMER.CustomerLastName;
```

This SQL statement joins the Art Course Database tables together, computes the difference between the course Fee and the AmountPaid, and stores this result in a new column named AmountDue. The SQL statement then selects only rows for which AmountDue is greater than zero and presents the results sorted by CustomerLastName. Compare the data in Figure 1-13 with the results in Figure 1-14 to ensure that the results are correct.

FIGURE 1-14

Results of the SQL Query to Compute Amount Due

CustomerLastName	CustomerFirstName	Phone	Course	CourseDate	Fee	AmountPaid	AmountDue
Johnson	Ariel	206-567-1234	Adv Pastels	10/1/2015	\$500.00	\$250.00	\$250.00
Kyle	Leah	425-678-7654	Adv Pastels	11/15/2015	\$500.00	\$250.00	\$250.00
Myers	Lynda	360-789-5678	Beg Oils	10/15/2015	\$350.00	\$0.00	\$350.00
*							