NINTH EDITION

DATABASE CONCEPTS

David M. Kroenke
David J. Auer
Scott L. Vandenberg
Robert C. Yoder

DATABASE CONCEPTS

NINTH EDITION

David M. Kroenke

David J. Auer
Western Washington University

Scott L. Vandenberg
Siena College

Robert C. Yoder
Siena College

Pearson

Vice President of Courseware Portfolio Management: Andrew Gilfillan
Executive Portfolio Manager: Samantha Lewis
Team Lead, Content Production: Laura Burgess
Content Producer: Faraz Sharique Ali
Portfolio Management Assistant: Bridget Daly
Director of Product Marketing: Brad Parkins
Director of Field Marketing: Jonathan Cottrell
Product Marketing Manager: Heather Taylor
Field Marketing Manager: Bob Nisbet
Product Marketing Assistant: Liz Bennett
Field Marketing Assistant: Derrica Moser
Senior Operations Specialist: Diane Peirano
Senior Art Director: Mary Seiner
Interior and Cover Design: Pearson CSC
Cover Art: Artwork by Donna Auer
Senior Product Model Manager: Eric Hakanson
Manager, Digital Studio: Heather Darby
Course Producer, MyLab MIS: Jaimie Noy
Digital Studio Producer: Tanika Henderson
Full-Service Project Manager: Gowthaman Sadhanandham
Full Service Vendor: Integra Software Service Pvt. Ltd.
Manufacturing Buyer: LSC Communications, Maura Zaldivar-Garcia
Text Printer/Bindery: LSC Communications
Cover Printer: Phoenix Color

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

Names: Kroenke, David M., author. | Auer, David J., author. | Vandenberg, Scott L., author. | Yoder, Robert C., author.
Title: Database concepts/David M. Kroenke, David J. Auer, Western Washington University, Scott L. Vandenberg, Siena College, Robert C. Yoder, Siena College.
Description: Ninth edition. | New York, NY : Pearson, [2020] | Includes index.
Identifiers: LCCN 2018052988 | ISBN 9780135188149 | ISBN 0135188148
Subjects: LCSH: Database management. | Relational databases.
Classification: LCC QA76.9.D3 K736 2020 | DDC 005.74–dc23 LC record available at https://lccn.loc.gov/2018052988

1 19

Pearson

ISBN 10: 0-13-518814-8
ISBN 13: 978-0-13-518814-9

Brief Contents

Contents

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 multi-panel 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* (also called a *table*) 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:

- The structure of the book has been reorganized to provide more emphasis on the essential concepts of databases and their use, and on the newest developments in Business Intelligence (BI) Systems, Cloud Computing, Big Data, and nonrelational (NoSQL) databases. Much material previously found in online extensions has been reintegrated into the book itself or reorganized into one of three online Extensions.

- Microsoft Office 2019, and particularly Microsoft Access 2019, is now the basic software used in the book and is shown running on Microsoft Windows 10.[1]
- The sections in each chapter previously called "The Access Workbench" have been reorganized as "Working with Microsoft Access," and now integrate material on using Microsoft Visio 2019 for data modeling and database design.
- DBMS software coverage has been updated to include MySQL Community Server 8.0 and MySQL Workbench 8.0. MySQL is now the example DBMS used throughout the book (instead of Microsoft SQL Server), and all SQL output in the book is shown as displayed in MySQL Workbench.[2]
- Detailed coverage of MySQL is in the online Extension "Working with MySQL," which now includes coverage equivalent to "Working with Microsoft Access." There is a separate section of "Working with MySQL" for each chapter in the book, so that work on MySQL can be done in conjunction with the topics covered in the book.
- Support for Oracle and Microsoft SQL Server is continued in the text of Chapter 3 and online Extension B (SQL and Advanced SQL) by indicating the major syntactic differences and in the instructor solutions to those sections.
- The continuing discussion of SQL in DBC e08 Appendix E, "Advanced SQL," is retained as the online Extension "Advanced SQL." The extension contains a discussion of the SQL ALTER statement, SQL set operators (UNION, etc.), SQL correlated subqueries, SQL views, and SQL/Persistent Stored Modules (SQL/PSM).
- A short introduction to physical database design and related considerations has been added to Chapter 6 ("Database Administration"), along with an example of using MySQL database security using roles.
- The NoSQL introduction in Chapter 7 has been expanded to include coverage of the CAP theorem and a more detailed description of document databases, using examples from the ArangoDB NoSQL database management system. This description is continued and deepened in the Extension "Advanced Business Intelligence and Big Data."
- Updated material on cloud computing, virtualization using containers, and the new Hadoop processing model is now in Chapter 7.
- Updated material for the former Appendices J, "Business Intelligence Systems" and K, "Big Data," is now available in the online Extension "Advanced Business Intelligence and Big Data." This material includes enhanced coverage of NoSQL DBMSs, with additional examples of NoSQL graph databases from an actual system (ArangoDB) and an expansion of the NoSQL document database coverage from Chapter 7 (again using ArangoDB). Some of the database features of the Microsoft Azure cloud platform are also covered in this extension.
- The James River Jewelry project questions have been reintegrated into the book chapters to make it easier to use this material when assigning projects.
- Videos demonstrating many tasks described in the "Working with Microsoft Access" and "Working with MySQL" sections of the book will be available online to users of the e-text version of the book.

[1] Microsoft recommends installing and using the 32-bit version of Microsoft Office 2019, even on 64-bit versions of the Microsoft Windows operating system. We also recommend that you install and use the 32-bit version.

[2] While we still like and use Microsoft SQL Server, it has simply become too complex for use as our example DBMS in Database Concepts. MySQL is much easier to install and use when introducing students to relational databased DBMS products in a concepts book like this one. We still use Microsoft SQL Server as our example DBMS in our book Database Processing: Fundamentals, Design, and Implementation, 15th ed. (Upper Saddle River, NJ: Pearson, 2019) where it is discussed in depth.

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:

- Data warehouse structures
- Business intelligence (BI) systems
- Cloud computing and virtualization
- Nonrelational structured data storage (Big Data and NoSQL systems)

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, David J. Auer, Scott L. Vandenberg, and Robert C. Yoder's *Database Processing: Fundamentals, Design, and Implementation*.[3] 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 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 MySQL, 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.

[3]David M. Kroenke, David J. Auer, Scott L. Vandenberg, and Robert C. Yoder, *Database Processing: Fundamentals, Design, and Implementation,* 15th ed. (Upper Saddle River, NJ: Pearson, 2019).

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 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 and other DBMS products, and many of them can be packaged with this text.

WORKING WITH MICROSOFT ACCESS

This new edition of the text renames "The Access Workbench" to "Working with Microsoft Access," and continues to feature Microsoft Access as our introductory DBMS. 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 "Working with Microsoft Access," which illustrates the chapter's concepts and techniques using Microsoft Access. The "Working with…" topics start with creating a database and a single table in Chapter 1 and move through various topics, finishing with using Microsoft Access (together with Microsoft Excel) to produce PivotTable OLAP reports in Chapter 7. We have also integrated the Microsoft Visio material formerly found in Appendix G into the book itself to demonstrate the use of Visio for creating data models as described in Chapter 4. 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.

WORKING WITH MYSQL

This new edition adds support for MySQL Community Server 8.0 at a level equivalent to "Working with Microsoft Access." This allows us to fully integrate an Enterprise class DBMS into the text. This material is found in online Extension A, "Working with MySQL." Topics include some previously found in Appendix C, in particular using the MySQL Workbench to create database designs as discussed in Chapter 5. Because MySQL is an Enterprise class DBMS widely used in introductory database classes, we feel it is important to include specific information on using MySQL. There is a section in "Working with MySQL" that aligns with each chapter in the book and with online Extension B, "Advanced SQL." While, like "Working with Microsoft Access," this material is not intended to provide comprehensive coverage of MySQL, all of the necessary basic MySQL topics are covered.[4]

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 "Working with Microsoft Access"), 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.

[4]Note that while we have increased our support of MySQL in this edition, we have dropped the material on Microsoft SQL Server and Oracle XE formerly found in Appendices A and B, respectively. We still include some basic discussion of Microsoft SQL Server and Oracle Database XE in the text, but if you need more complete coverage of either Microsoft SQL Server or Oracle Database, please see David M. Kroenke, David J. Auer, Scott L. Vandenberg, and Robert C. Yoder, *Database Processing: Fundamentals, Design, and Implementation,* 15th ed. (Upper Saddle River, NJ: Pearson, 2019).

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. 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 Developer edition and Express edition, Oracle Corporation's Oracle Database Express Edition (Oracle Database XE), and MySQL Community Server). In this text, we will be using Microsoft Access 2019, Microsoft Excel 2019, and MySQL Community Edition 8.0.

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 EIGHTH EDITION

This edition represents a restructuring of Database Concepts to reemphasize the basic concepts of database systems and database development in today's mobile environment. We have streamlined the content and presentation to make it easier for instructors to cover the essentials of database systems. We frame database topic discussions within today's Internet- and mobile applications–based networked environment and economy. Today, databases are no longer isolated entities found somewhere in obscure server rooms, but rather are ubiquitous parts of websites and tablet and smartphone apps. We are literally dependent upon databases in our lives, whether exchanging email messages, posting to our Facebook pages, or shopping online. To this end, we have reintegrated material formerly in several appendices back into the book itself, and reduced online material to three extensions:

- Online extension A—Working with MySQL. This extension provides detailed coverage of MySQL Community Server 8.0 and the MySQL Workbench similar to the coverage of Microsoft Access in the "Working with Microsoft Access" sections.
- Online extension B—Advanced SQL. This extension provides the same material to supplement the SQL material in Chapter 3 that was previously found in Appendix E. This material should meet the needs of those classes that want to go beyond basic SQL without the need for an additional textbook.
- Online extension C—Advanced Business Intelligence and Big Data. This extension provides coverage of how to use BI, Big Data, Azure Cloud, and nonrelational database tools so that you can have your students experience these tools in action.

BOOK OVERVIEW

This textbook consists of seven chapters and three online extensions (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 describes 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 Extension B, "Advanced SQL," adds coverage of advanced SQL topics, such as the SQL ALTER TABLE statement, SQL set operators (e.g., UNION), SQL views, and SQL/Persistent Stored Modules (SQL/PSM).

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 revisits normalization as a step in database design. 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 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. For instructors who want a bit more detail on normal forms, short definitions of most normal forms are included in Chapter 2.

The last two 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 discusses the emerging world of Big Data and the NoSQL movement, including under this umbrella business intelligence (BI) systems and the data warehouse architectures that support them, which often involve Big Data and NoSQL concepts. Chapter 7 also provides a discussion of distributed databases, object-relational databases, virtualization, and cloud computing as they relate to the continuing evolution of NoSQL systems and Big Data. Many details of BI and NoSQL systems can be found in online Extension C, "Advanced Business Intelligence and Big Data." More specifically, Chapter 7 discusses dimensional databases as an example of a data warehouse architecture, walking through how to build a dimensional database for Heather Sweeney Designs and then using it to produce a PivotTable online analytical processing (OLAP) report as an example of BI reporting. This chapter also includes a brief introduction to NoSQL databases in general and document databases in particular, using the ArangoDB NoSQL DBMS.

Online Extension A introduces MySQL Community Server 8.0 and the MySQL Workbench. There is a section to match each chapter in the book and online Extension B, "Advanced SQL." This allows students to gradually learn the capabilities of an Enterprise class DBMS.

Online Extension B provides additional coverage of SQL topics to allow instructors to teach advanced SQL topics without the need of an additional textbook. This includes material on SQL programming via SQL/Persistent Stored Modules (SQL/PSM).

Online Extension C describes some data analysis techniques, including report systems, data mining, market basket analysis, and decision trees. An introduction to XML and JSON provides context for more details on document databases and other NoSQL database systems, using ArangoDB to illustrate both graph databases and more advanced features of document databases. We present a case study of using Microsoft SQL Server Management Studio to create an account and connect to Microsoft's Azure cloud to migrate a SQL database from a personal computer to Azure. Lastly, we show how to create switchboards in our "Working with Microsoft Access" section.

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:

Arthur Lee, Lord Fairfax Community College
Behrooz Saghafi, Ph.D., Chicago State University
Betsy Page Sigman, Georgetown University
Bijoy Bordoloi, Southern Illinois University, Edwardsville
Carolyn Carvalho, Kent State University at Ashtabula
David Chou, Eastern Michigan University
David L. Olson, University of Nebraska
Fen Wang, Central Washington University
Gabriel Peterson, North Carolina Central University
Jeffrey Burton, Daytona State College
Jim Pierson, Forsyth Technical Community College
Jing Wang, University of New Hampshire
Jose Nieves, Lord Fairfax Community College
Joshua S. White, PhD, State University of New York Polytechnic Institute
Julie Lewis, Baker College
June Lane, Bucks County Community College
Kui Du, University of Massachusetts Boston
Manuel Rossetti, University of Arkansas
Matt Hightower, Cerro Coso Community College
Maya Tolappa, Waubonsee Community College
Meg Murray, Kennesaw State University
Norman Hahn, Thomas Nelson Community College
Patrick Appiah-Kubi, Indiana State University
Paul Pennington, University of Houston
Paul Tallon, Loyola University Maryland
Richard Grant, Seminole State College of Florida
Richard T. Evans, South Suburban College
Robert Demers, University of Massachusetts - Lowell
Stephen Larson, Slippery Rock University

We would like to thank Donna Auer for letting us use one of her cold wax paintings as the cover art for this book. This artwork was also the basis for design elements within the book.

We would like to thank Samantha Lewis, our editor; Faraz Sharique Ali, our content producer, and Gowthaman Sadhanandham, our project manager, for their professionalism, insight, support, and assistance in the development of this project. We would also like to thank William Morris for his comments on the text and Harold Wise for his work on the supplements. Finally, David Kroenke would like to thank his wife, Lynda; David Auer would like to thank his wife, Donna; Scott Vandenberg would like to thank his wife, Kristin; and Robert Yoder would like to thank Diane, Rachael, and Harrison Yoder for their love, encouragement, and patience while this project was being completed.

David Kroenke
Langley, Washington

David Auer
Clinton, Washington

Scott Vandenberg
Loudonville, New York

Robert Yoder
Loudonville, New York

About the Authors

David M. Kroenke entered the computing profession as a summer intern at the RAND Corporation in 1967. Now retired, his career 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 14th edition (coauthored with David Auer for the 11th, 12th, 13th, and 14th editions, and with David Auer, Scott Vandenberg, and Robert Yoder for the 15th [40th anniversary] edition). He introduced *Database Concepts* (now in the 9th edition that you are reading) in 2003 (coauthored with David Auer for the 3rd, 4th, 5th, 6th, and 7th editions, and coauthored with David Auer, Scott Vandenberg, and Robert Yoder for the 8th and 9th editions). Kroenke has published many other textbooks, including the classic *Business Computer Systems* (1981). Recently, he has authored *Using MIS* (8th edition), *Experiencing MIS* (6th 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 on Whidbey Island in Washington state. He has two children and three grandchildren.

David J. Auer is a Senior Instructor Emeritus at the College of Business (CBE) of Western Washington University in Bellingham, WA. He served as the director of Information Systems and Technology Services at CBE from 1994 to 2014 and taught in CBE's Department of Decision Sciences from 1981 to 2015. 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, including *Database Processing: Fundamentals, Design, and Implementation*, first published in 1977, and now in its 15th edition (coauthored with David Kroenke for the 11th, 12th, 13th, and 14th editions, and with David Kroenke, Scott Vandenberg, and Robert Yoder for the 15th [40th anniversary] edition), and *Database Concepts*, now in the 9th edition that you are reading (coauthored with David Kroenke for the 3rd, 4th, 5th, 6th, and 7th editions, and coauthored with David Kroenke, Scott Vandenberg, and Robert Yoder for the 8th and 9th editions).

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 on Whidbey Island in Washington state. He has two children and four grandchildren.

Scott L. Vandenberg has been on the Computer Science faculty at Siena College since 1993, where he regularly teaches three different database courses at several levels to both computer science and business majors. Prior to arriving at Siena, he taught undergraduate and graduate courses in database systems at the University of Massachusetts–Amherst. Since arriving at Siena, he has also taught graduate and undergraduate database courses at the University of Washington–Seattle. He has developed five different database courses over this time. His other teaching experience includes introductory computer science, introductory programming, data structures, management information systems, and three years teaching Siena's required interdisciplinary freshman writing course.

Vandenberg's recent research publications are mainly in the areas of computer science education and data science applications, with earlier work on query optimization and algebraic query languages. He holds a bachelor's degree in mathematics and computer science from Cornell University and master's and PhD degrees in computer science from the University of Wisconsin–Madison. Medieval history and playing hockey are two things that can tear him away from a database. Vandenberg lives in Averill Park, NY, with his wife, Kristin, and has two children.

Robert C. Yoder began his professional career at the University at Albany as a systems programmer managing mainframes and Unix servers. He became the Assistant Director of Systems Programming, gaining over 25 years' experience as a programmer and technical manager. Yoder has research experience working on 3-D geographic information systems. Yoder holds BS and MS degrees in computer science and a PhD in information science, all from the University at Albany.

Yoder joined the Computer Science department at Siena College in 2001 and teaches business database, management information systems, geographic information systems, data structures, networks, and operating systems courses. Yoder lives in Niskayuna, NY, with his wife, Diane, and has two children. He enjoys traveling, hiking, and walking his dog.

This page intentionally left blank

PART 1

Database Fundamentals

Part 1 introduces fundamental concepts and techniques of 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 (SQL), 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 2. Finally, we will discuss database management, data warehouses, business intelligence (BI) systems, cloud computing, virtualization, Big Data, and nonrelational "NoSQL" database management systems in Part 3.

This page intentionally left blank

CHAPTER 1

Getting Started

CHAPTER OBJECTIVES

- Understand the importance of databases in Internet Web applications and mobile apps
- Understand the nature and characteristics of databases
- Understand 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 Web database applications
- Introduce data warehouses and business intelligence (BI) systems
- Introduce Big Data and cloud computing

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.

This chapter discusses the importance of databases in the Internet world and then introduces database processing concepts. We will 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 a database management system (DBMS), and the functions of a database application. Finally, we introduce nonrelational databases.

THE IMPORTANCE OF DATABASES IN THE INTERNET AND MOBILE APP WORLD

Let's stop for a moment and consider the incredible information technology available for our use today.

The **personal computer (PC)** became widely available with the introduction of the **Apple II** in 1977 and the **IBM Personal Computer (IBM PC)** in 1981. PCs were networked into **Local Area Networks (LANs)** using the **Ethernet networking technology**, which was developed at the **Xerox Palo Alto Research Center (Xerox PARC)**[1] in the early 1970s and adopted as a national standard in 1983.

The **Internet**—the global computer network of networks—was created as the Department of Defense **Advanced Research Projects Agency Network (ARPANET)** in 1969 and then grew and was used to connect all the LANs (and other types of networks). The Internet became widely known and used when the **World Wide Web**[2] (or **the Web,** or **WWW**) became easily accessible in 1993. Everyone got a computer software application called a **Web browser** and starting *browsing* **Web sites**. Online retail Web sites such as **Amazon.com** (online since 1995) and "brick-and-mortar" stores with an online presence such as Best Buy appeared, and people started extensively *shopping online*.

In the early 2000s, **Web 2.0**[3] Web sites started to appear—allowing users to add content to Web sites that had previously held static content. Web applications such as Facebook, Wikipedia, and Twitter appeared and flourished.

In a parallel development, the **mobile phone** or **cell phone** was demonstrated and developed for commercial use in the 1970s. After decades of mobile phone and cell phone network infrastructure development, the **smartphone** appeared. Apple brought out the **iPhone** in 2007. Google created the **Android operating system**, and the first Android-based smartphone entered the market in 2008. Ten years later, in 2018 (as this is being written), smartphones and **tablet computers (tablets)** are widely used, and thousands of application programs known as **apps** are widely available and in daily use. Most Web applications now have corresponding smartphone and tablet apps (you can "tweet" from either your computer or your smartphone)!

The latest development is the **Internet of Things (IoT)**,[4] where devices such as smart speakers, smart home devices (smoke detectors and thermostats), and even appliances such as refrigerators connect to the Internet and are network accessible. In particular, **smart speakers** such as the Amazon Echo series, Apple HomePod, the Google Home series, and Harmon Kardon INVOKE enable users to interact by voice with network-accessible apps via virtual assistants such as Amazon Alexa, Apple's Siri, Google Assistant, and Microsoft Cortana.

What many people do not understand is that in today's Web application, smartphone app, and IoT app environment, most of what they do depends upon databases.

We can define **data** as recorded facts and numbers. We can initially define a *database* (we will give a better definition later in this chapter) as the structure used to hold or store

---

[1]The mouse and the multi-window graphical user interface commonly used in computer operating systems today were also developed at Xerox PARC. From there, they were adapted and popularized by Apple and Microsoft. For more information, see the Wikipedia article **PARC (company)** (accessed June 2018) at https://en.wikipedia.org/wiki/PARC_(company).

[2]The World Wide Web and the first Web browser were created by Tim Berners-Lee in 1989 and 1990, respectively. For more information, see the Wikipedia article **World Wide Web** (accessed June 2018) at https://en.wikipedia.org/wiki/World_Wide_Web and the World Wide Web Consortium (W3C) Web site (accessed June 2018) at https://www.w3.org/Consortium/.

[3]Web 2.0 was originated by Darcy DiNucci in 1999 and introduced to the world at large in 2004 by publisher Tim O'Reilly. See the Wikipedia article **Web 2.0** (accessed June 2018) at https://en.wikipedia.org/wiki/Web_2.0.

[4]For more information, see the Wikipedia article **Internet of Things** (accessed June 2018) at https://en.wikipedia.org/wiki/Internet_of_things. The article states that there were 8.4 billion network-capable IoT devices in 2017, with an expected 30 billion IoT devices by 2020!

that data. We process that data to provide *information* (which we also define in more detail later in this chapter) for use in Web applications and smartphone apps.

Do you have a Facebook account? If so, all your posts, your comments, your "likes," and other data you provide to Facebook (such as photos) are stored in a *database*. When your friend posts an item, it is initially stored in the *database* and then displayed to you.

Do you have a Twitter account? If so, all your tweets are stored in a *database*. When your friend tweets something, it is initially stored in the *database* and then displayed to you.

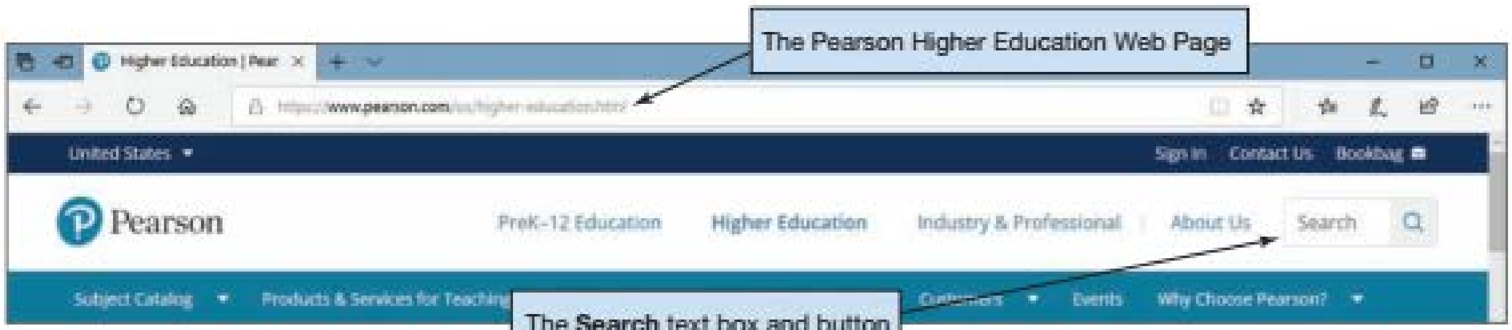
Do you shop at **Amazon.com**? If so, how do you find what you are looking for? You enter some words in a search text window on the Amazon home Web page (if you are using a Web browser) and click the Go button. Amazon's computers then search Amazon's *databases* and return a formatted on-screen report of items that match what you searched for.

The search process is illustrated in Figure 1-1, where we search the Pearson Higher Education Web site for books authored by *David Kroenke*. Figure 1-1(a) shows the upper portion of the Pearson Higher Education Web site home page. While many Web sites (including Amazon.com, REI, and Best Buy) have a text box for entering search keywords on the home page itself for immediate use, at the Pearson site we have to click on a Search catalog button to access the search function on the *Advanced Catalog Search* page shown in Figure 1-1(b). On this page, we enter the author name *Kroenke* in the Author text box, and then click the Search button. The Pearson catalog database is searched, and the Web application returns a *Search Results* page containing a listing of books authored by David Kroenke, as shown in Figure 1-1(c).

The use of databases by Web applications and smartphone apps is illustrated in Figure 1-2. In this figure, people have computers (desktop or notebook), smartphones, and smart speakers, which are examples of **devices** used by people, who are referred to as **users**.

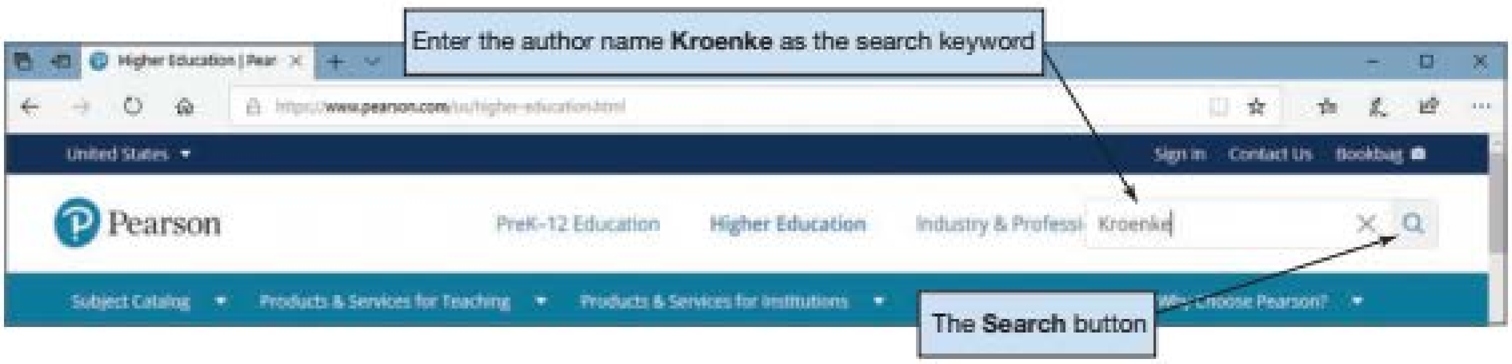
FIGURE 1-1

Searching a Database in a Web Browser



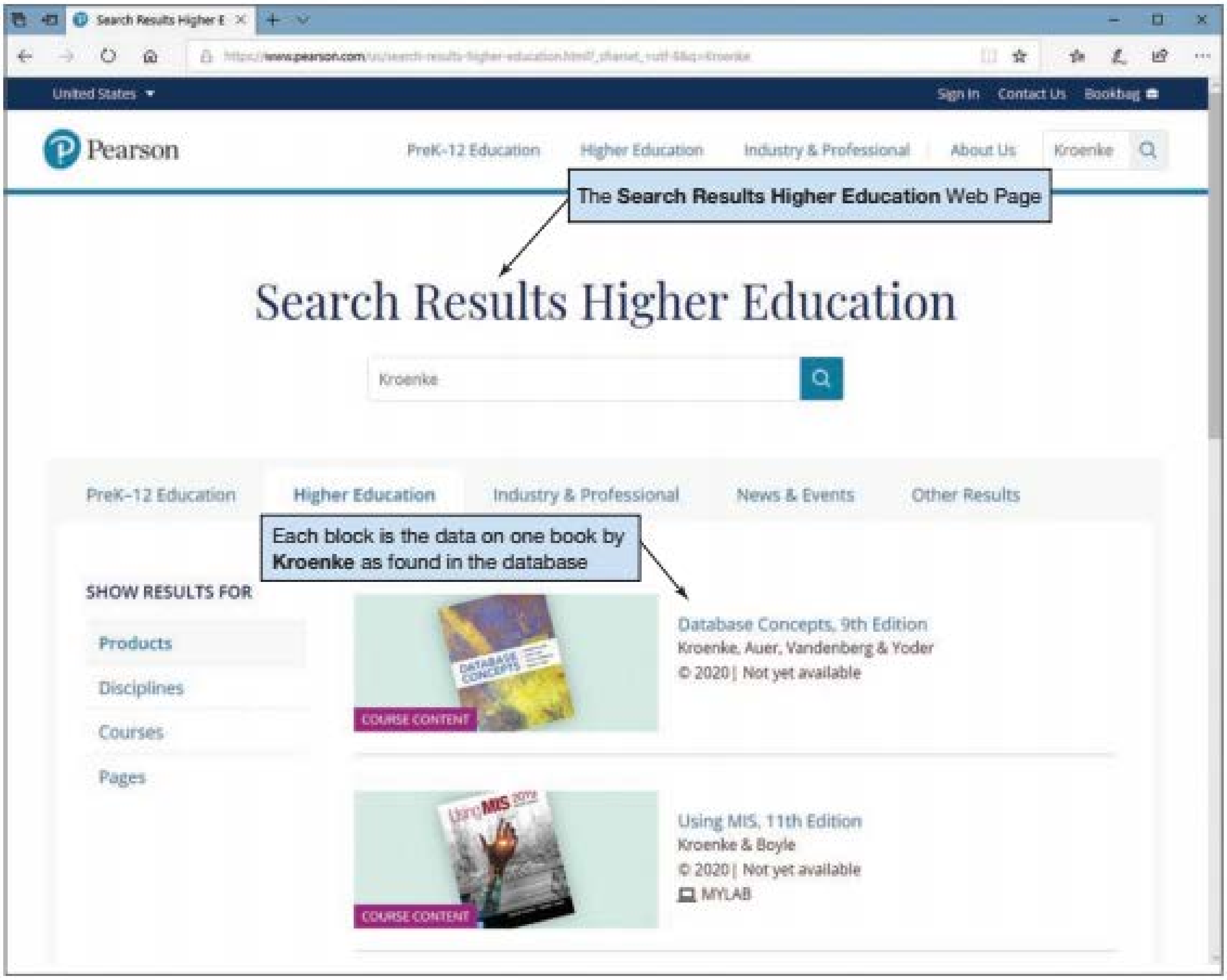
(a) The Pearson Higher Education Web Site Home Page

Courtesy of Pearson Education.



(b) Entering Author Name *Kroenke* as the Search Keyword

Courtesy of Pearson Education.



(c) Books by Author Kroenke Found in the Database

BTW

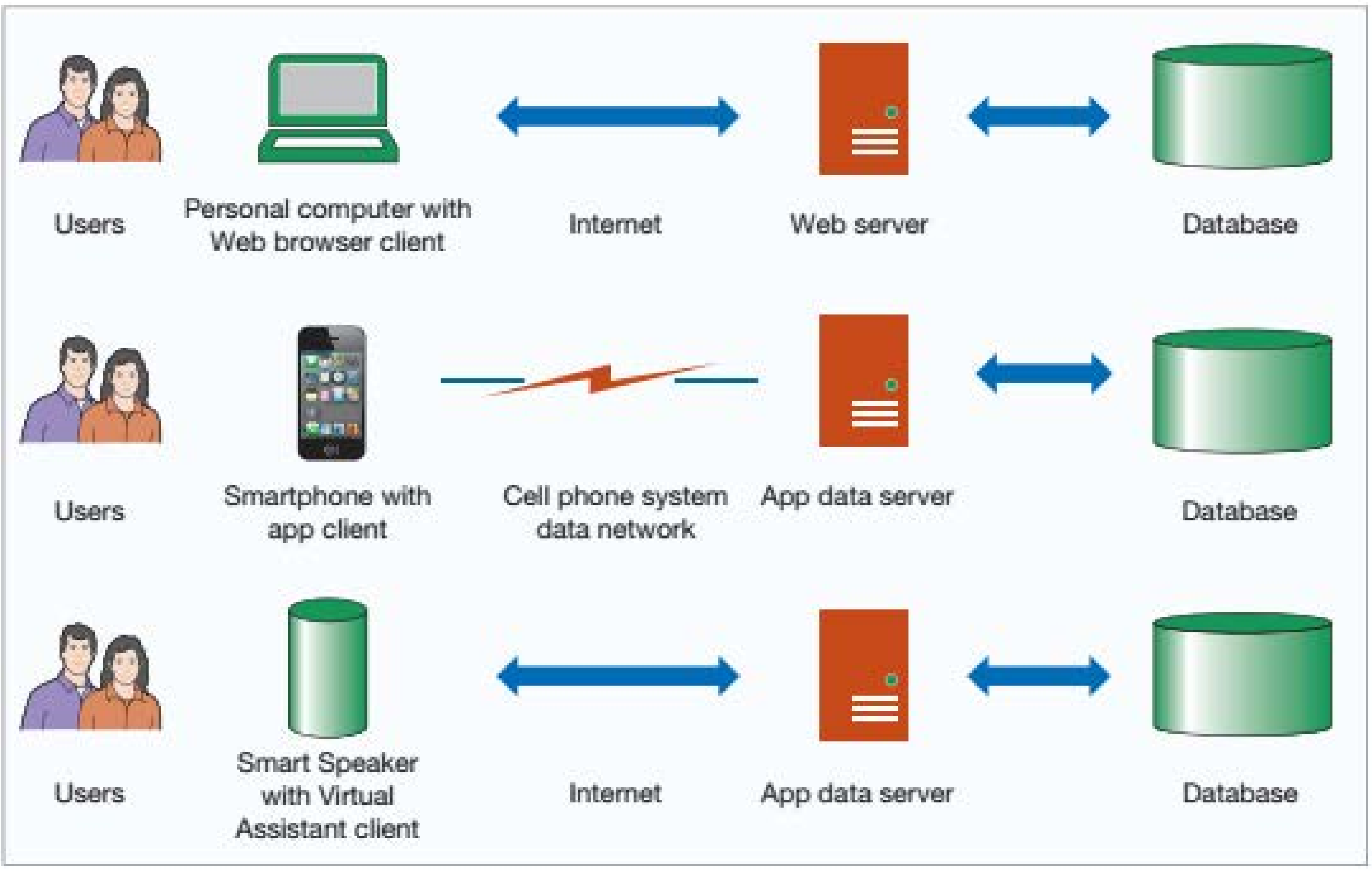
Seeing this process is much more effective than just reading about it. Take a minute, open a Web browser, and go to Amazon.com (or any other online retailer, such as Best Buy, L.L.Bean, or REI). Search for something you are interested in, and watch the database search results be displayed for you. You just used a *database*.

BTW

Even if you are simply shopping in a local grocery store (or a coffee shop or pizzeria), you are interacting with databases. Businesses use **Point of Sale (POS) systems** to record every purchase in a database, to monitor inventory, and, if you have a sales promotion card from the store (the one you use to get those special prices for "cardholders only"), to keep track of everything you buy for marketing purposes. All the data POS systems gather is stored in, of course, a *database*.

FIGURE 1-2

The Internet and Mobile Device World



On these devices are **client** applications (Web browsers, apps, virtual assistants) used by people to obtain **services** such as searching, browsing, online purchasing, and tweeting over the Internet or cell phone networks. These services are provided by **server** computers, and these computers hold the databases containing the data needed by the client applications.

This structure is known as **client-server architecture**, and it supports most of the Web applications in use today. The simple fact is that without databases, we could not have the ubiquitous Web applications and apps that are currently used by so many people.

WHY USE A DATABASE?

A database is used to help people keep track of things, and the most commonly used type of database is the *relational database*, although *nonrelational databases* (which are used extensively in online applications such as Facebook and Twitter) are now also widely used. We will discuss the relational database model in depth in Chapter 2, and nonrelational databases are discussed in Chapter 7. For now, you just need to understand a few basic facts about how a relational database helps people track things of interest to them.

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 relational database. A majority of this text concerns the design of such tables and techniques for manipulating the data they contain.

WHAT ARE THE PROBLEMS WITH USING LISTS?

Figure 1-3 shows a simple list of student data, named the Student List,[5] 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 1-3, 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 1-4. 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 1-5, if you delete the eighth row (the row numbered 8—this is actually the seventh row of data because of the column headers, but it is easier to refer to the row number shown in the figure) 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 fifth row, you will have inconsistent data. After the change, the fourth row indicates one email address for Professor Taing, and the fifth 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.

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 1-5, 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 Chapter 2, where we will show that null values are always problematic and that we want to avoid them whenever possible.

FIGURE 1-3

The Student List in a Spreadsheet

	A	B	C	D
1	SID	StudentLastName	StudentFirstName	StudentEmail
2	S0023	Andrews	Matthew	Matthew.Andrews@ourcampus.edu
3	S0065	Fischer	Douglas	Douglas.Fisher@ourcampus.edu
4	S0083	Hwang	Terry	Terry.Hwang@ourcampus.edu
5	S0132	Thompson	James	James.Thompson@ourcampus.edu
6	S0154	Brisbon	Lisa	Lis.Brisbon@ourcampus.edu
7	S0167	Lai	Tzu	Tzu.Lai@ourcampus.edu
8	S0212	Marino	Chip	Chip.Marino@ourcampus.edu

Excel 2019, Windows 10, Microsoft Corporation.

FIGURE 1-4

The Student with Adviser List

	A	B	C	D	E	F
1	SID	StudentLastName	StudentFirstName	StudentEmail	AdviserLastName	AdviserEmail
2	S0023	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
3	S0065	Fischer	Douglas	Douglas.Fisher@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
4	S0083	Hwang	Terry	Terry.Hwang@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu
5	S0132	Thompson	James	James.Thompson@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu
6	S0154	Brisbon	Lisa	Lis.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu
7	S0167	Lai	Tzu	Tzu.Lai@ourcampus.edu	Valdez	Bill.Yeats@ourcampus.edu
8	S0212	Marino	Chip	Chip.Marino@ourcampus.edu	Tran	Ken.Tran@ourcampus.edu

Excel 2019, Windows 10, Microsoft Corporation.

[5] In 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.

Now, what exactly happened in these two examples? We had a simple list with four columns, added two more columns to it, and thereby created several problems. The problem is not just that the list has six columns instead of four. Consider a different list that has six columns: the Student with Residence List shown in Figure 1-6. This list has five columns, yet it suffers from none of the problems of the Student with Adviser List in Figure 1-5.

In the Student with Residence List in Figure 1-6, 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 1-5 and the Student with Residence List in Figure 1-6. Looking at those two figures, can you determine the difference? The essential difference is that the Student with Residence List in Figure 1-6 is all about a *single thing*: All the data in that list concern *students*. In contrast, the Student with Adviser List in Figure 1-5 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 1-7. 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 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 has thousands of rows—and if several people process it, the list will be a mess in a very short time.

FIGURE 1-5

Modification Problems in the Student with Adviser List

	A	B	C	D	E	F
1	SID	StudentLastName	StudentFirstName	StudentEmail	AdvisorLastName	AdviserEmail
2	S0023	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
3	S0065	Fischer	Douglas	Douglas.Fisher@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu
4	S0083	Hwang	Terry	Terry.Hwang@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu
5	S0132	Thompson	James	James.Thompson@ourcampus.edu	Taing	Sue.Taing@ourcampus.edu
6	S0154	Brisbon	Lisa	Lis.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu
7	S0167	Lai	Tzu	Tzu.Lai@ourcampus.edu	Valdez	Bill.Yeats@ourcampus.edu
8	S0212	Marino	Chip	Chip.Marino@ourcampus.edu	Tran	Ken.Tran@ourcampus.edu
9	???	???	???	???	Green	George.Green@ourcampus.edu

Excel 2019, Windows 10, Microsoft Corporation.

FIGURE 1-6

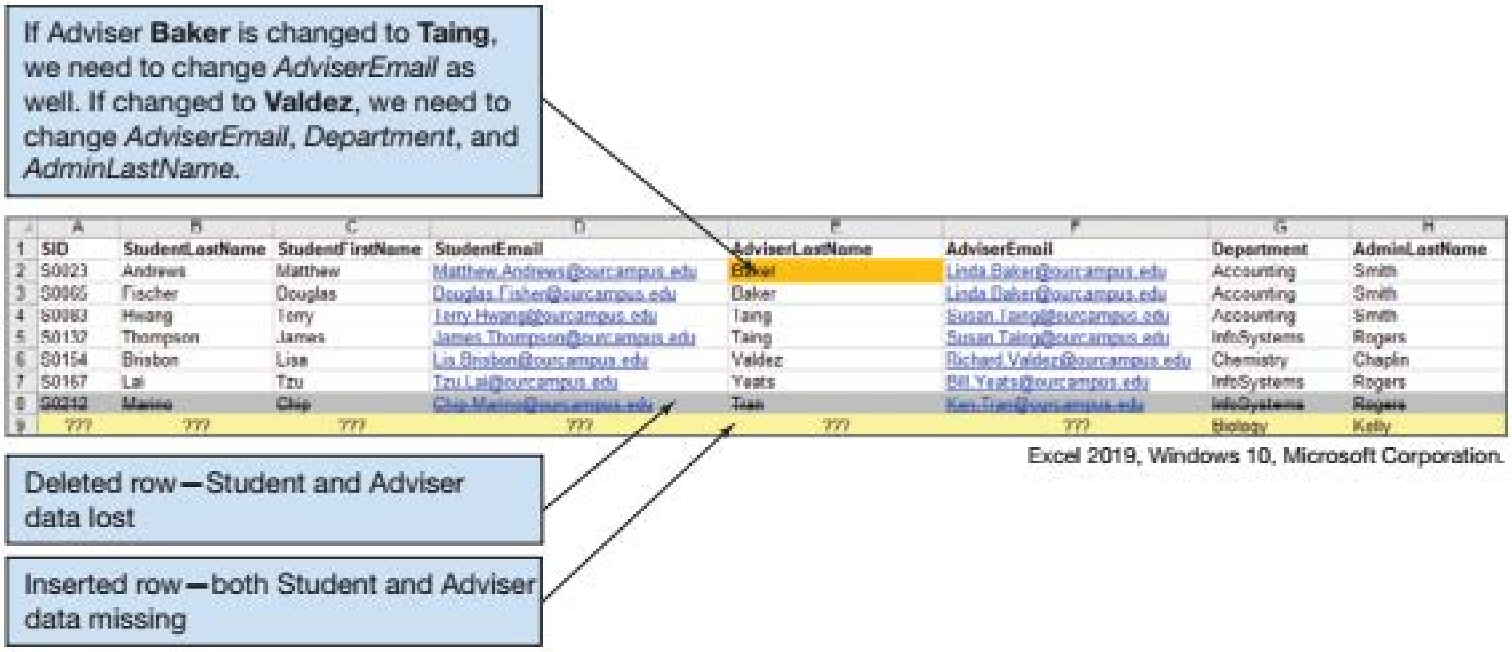
The Student with Residence List

	A	B	C	D	E	F
1	SID	StudentLastName	StudentFirstName	StudentEmail	Phone	Residence
2	S0023	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	301-555-2225	123 15th St Apt 21
3	S0065	Fischer	Douglas	Douglas.Fisher@ourcampus.edu	301-555-2257	McKinley Room 109
4	S0083	Hwang	Terry	Terry.Hwang@ourcampus.edu	301-555-2229	McKinley Room 208
5	S0132	Thompson	James	James.Thompson@ourcampus.edu	301-555-2245	345 17th St Apt 43
6	S0154	Brisbon	Lisa	Lis.Brisbon@ourcampus.edu	301-555-2241	Dorsett Room 201
7	S0167	Lai	Tzu	Tzu.Lai@ourcampus.edu	301-555-2231	McKinley Room 115
8	~~S0212~~	~~Marino~~	~~Chip~~	~~Chip.Marino@ourcampus.edu~~	~~301-555-2243~~	~~234 16th St Apt 32~~
9	S0213	Ingram	Garrett	Garret.Ingram@ourcampus.edu	301-555-2223	Dorsett Room 218

Excel 2019, Windows 10, Microsoft Corporation.

FIGURE 1-7

The Student with Adviser and Department List



	A	B	C	D	E	F	G	H
1	SID	StudentLastName	StudentFirstName	StudentEmail	AdviserLastName	AdviserEmail	Department	AdminLastName
2	S0023	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
3	S0065	Fischer	Douglas	Douglas.Fisher@ourcampus.edu	Baker	Linda.Baker@ourcampus.edu	Accounting	Smith
4	S0083	Hwang	Terry	Terry.Hwang@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu	Accounting	Smith
5	S0132	Thompson	James	James.Thompson@ourcampus.edu	Taing	Susan.Taing@ourcampus.edu	InfoSystems	Rogers
6	S0154	Brisbon	Lisa	Lis.Brisbon@ourcampus.edu	Valdez	Richard.Valdez@ourcampus.edu	Chemistry	Chaplin
7	S0167	Lai	Tzu	Tzu.Lai@ourcampus.edu	Yeats	Bill.Yeats@ourcampus.edu	InfoSystems	Rogers
8	~~S0212~~	~~Marino~~	~~Chip~~	~~Chip.Marino@ourcampus.edu~~	~~Tran~~	~~Ken.Tran@ourcampus.edu~~	~~InfoSystems~~	~~Rogers~~
9	???	???	???	???	???	???	Biology	Kelly

Excel 2019, Windows 10, Microsoft Corporation.

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 most commercial databases are still based on the relational model. We 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.

A Relational Design for the Student with Adviser List

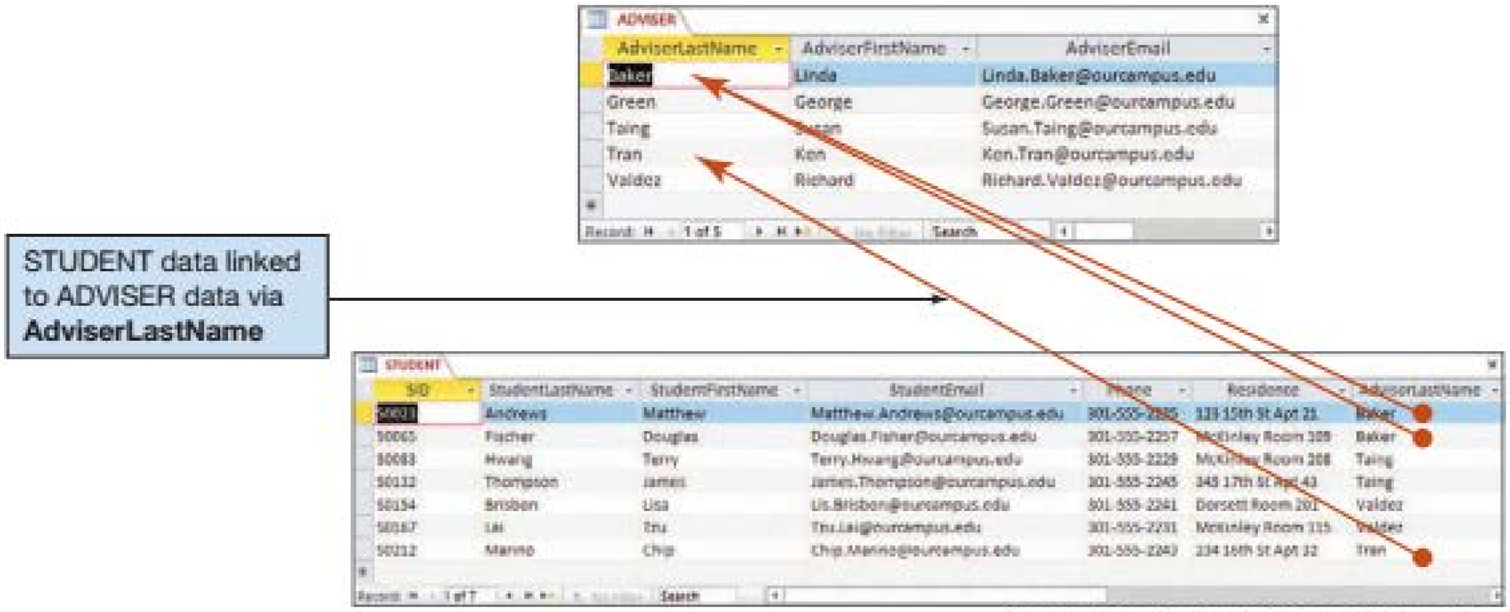
The Student with Adviser List in Figures 1-4 and 1-5 has two themes: *students* and *advisers*. If we put this data into a relational database, we place the student data in one table named STUDENT and the adviser data in a second table named ADVISER.

A database usually has multiple tables, and each table contains data about a different type of thing. For example, Figure 1-8 shows a database with two tables: The STUDENT table holds data about students, and the ADVISER table holds data about advisers.

A table has *rows* and *columns*, like those in a spreadsheet. 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 seven students: Andrews, Brisbon, Fischer, Hwang, Lai, Marino, and Thompson. Similarly, each row of the ADVISER table has data about a particular adviser. Because each row *records* the data for a specific instance, each row is also known as a **record**. 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 StudentLastName, and so forth. Columns are also known as **fields**.

FIGURE 1-8

The Adviser and Student Tables



Access 2019, Windows 10, Microsoft Corporation.

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.

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 text and 99.999999 percent of all databases throughout the world store instances in rows and characteristics in columns.

BTW

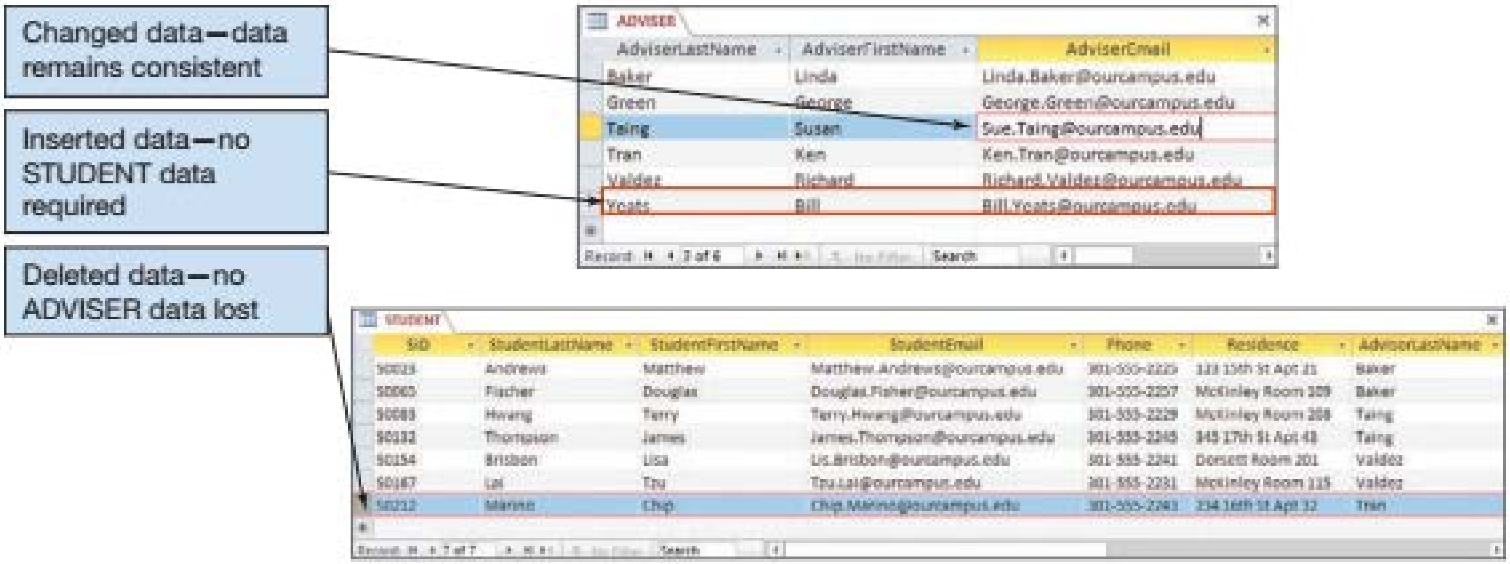
In this book, table names appear in all capital, or uppercase, letters (STUDENT, ADVISER). Column names have initial capitals (Phone, Address), and where column names consist of more than one word, the initial letter of each word is capitalized (LastName, AdviserEmail).

We still want to show which students have which advisers, however, so we leave AdviserLastName in the STUDENT table. As shown in Figure 1-8, 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 1-9, we can insert, update, and delete in these tables with no modification problems.

FIGURE 1-9

Modifying the Adviser and Student Tables



Access 2019, Windows 10, Microsoft Corporation.

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 Department List shown in Figure 1-7. 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 1-10.

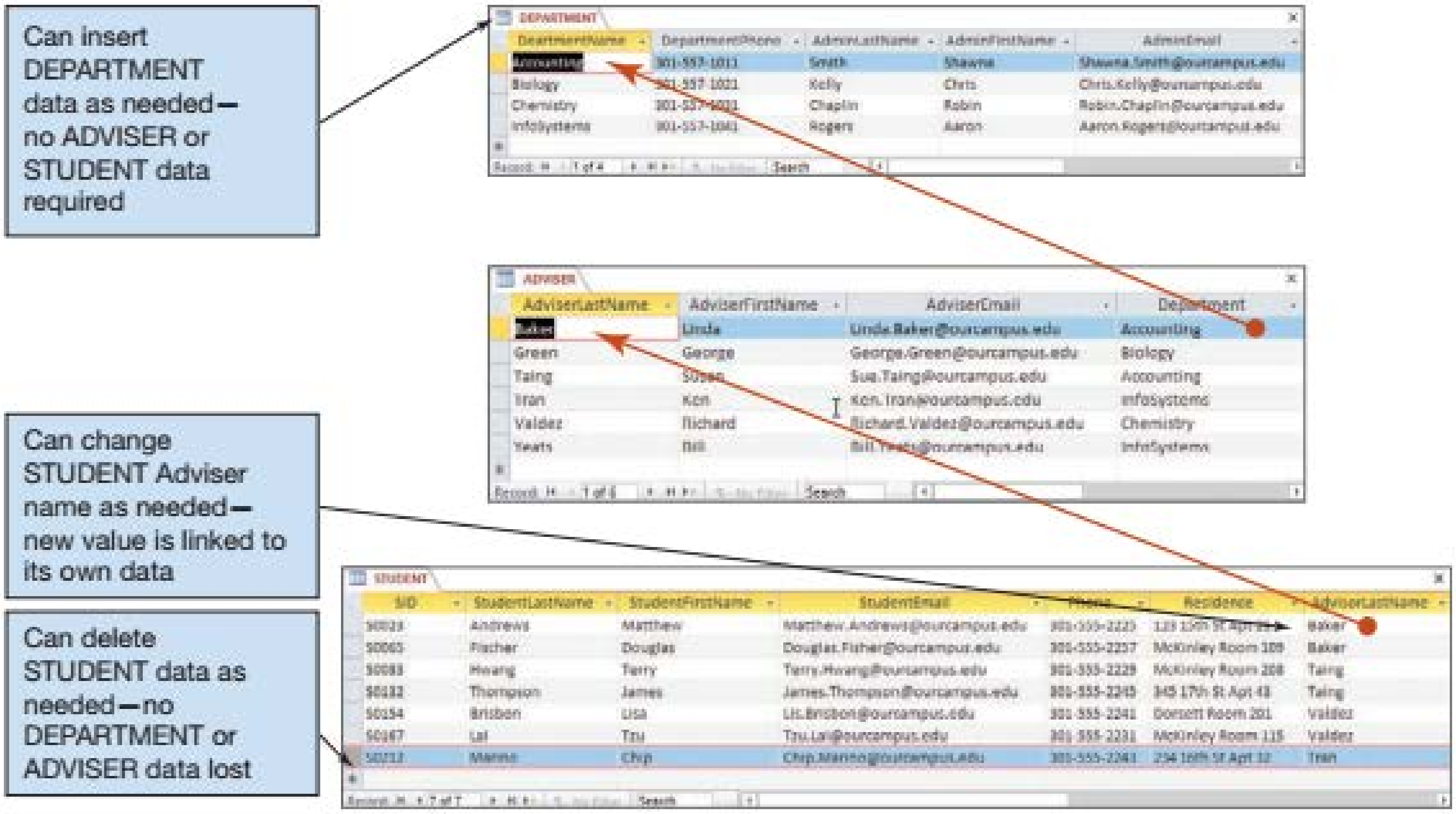
As illustrated in Figure 1-10, we can use AdviserLastName and Department to link the tables. Also, as shown in this figure, this set of tables does not have any modification problems. We can insert new data without creating null values, we can modify data without creating inconsistencies, and we can 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 1-10 has removed the problems that occur when modifying a list, but it has also introduced a new problem, this time in the ADVISER table.

FIGURE 1-10

The Department, Adviser, and Student Tables



Access 2019, Windows 10, Microsoft Corporation.

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. Also, with the current table design we can't have two or more advisers with the same last name! We are skipping way ahead here, however, and 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 1-11, 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.