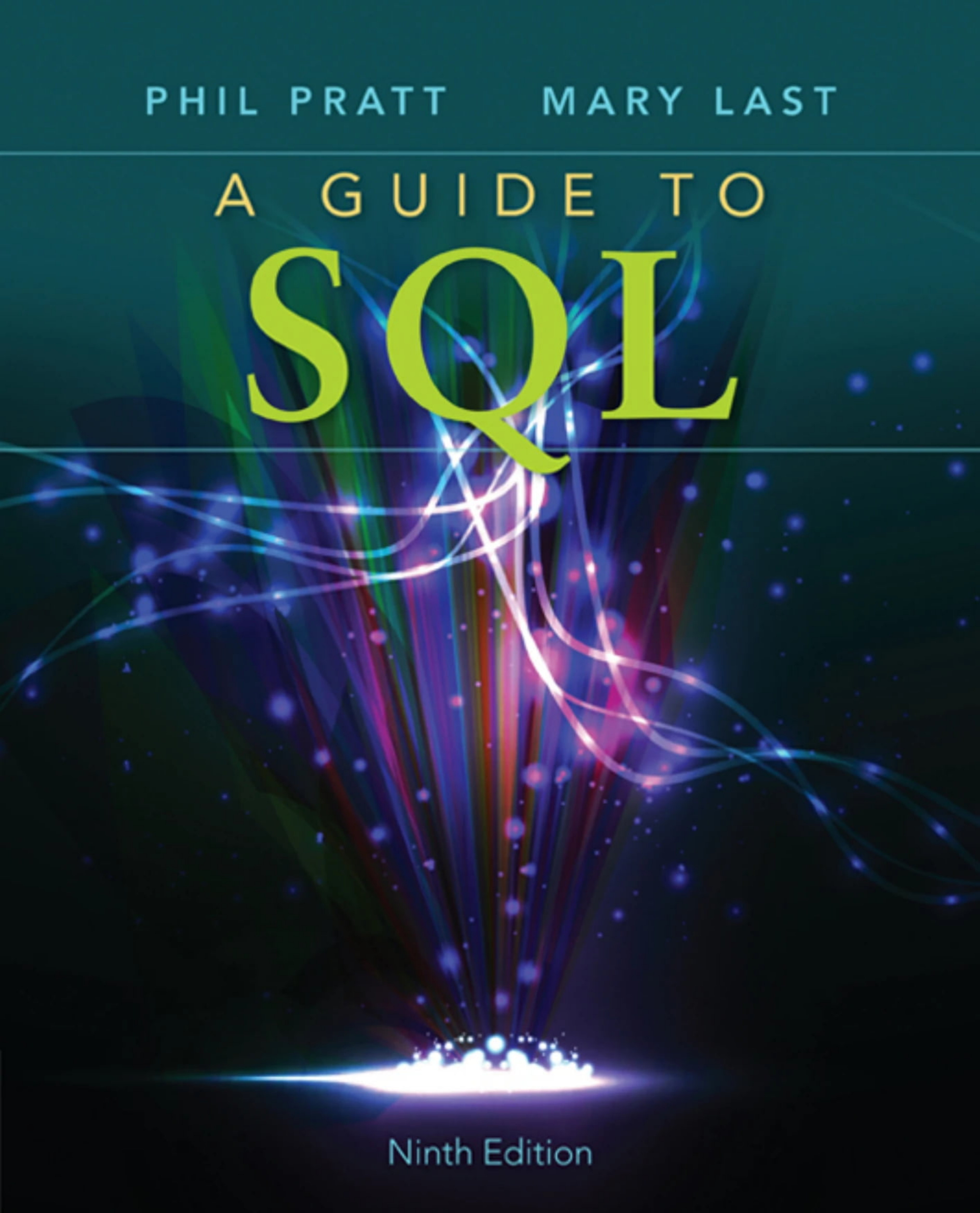


PHIL PRATT

MARY LAST

A GUIDE TO

SQL

The background of the cover is a dark, deep blue. It is filled with a complex network of glowing, multi-colored lines in shades of blue, green, and purple. These lines appear to be flowing and swirling, creating a sense of dynamic energy. Scattered throughout the scene are numerous small, bright particles or 'stars' in various colors, including blue, green, and white. At the bottom center, there is a bright, glowing white and blue light source that seems to be the origin of the lines and particles, creating a lens flare effect.

Ninth Edition

A GUIDE TO SQL

A GUIDE TO SQL

Ninth Edition

Philip J. Pratt

Grand Valley State University

Mary Z. Last



A Guide to SQL, Ninth Edition
Philip J. Pratt and Mary Z. Last

Vice President, General Manager, Social
Science & Qualitative Business: Erin Joyner

Product Director: Joe Sabatino

Product Manager: Jason Guyler

Content Developer: Wendy Langerud,
S4 Carlisle

Managing Developer: Greg Albert

Product Assistant: Brad Sullender

Marketing Director: Michèle McTighe

Senior Marketing Manager: Eric La Scola

Marketing Coordinator: William Guiliani

IP Analyst: Sara Crane

Senior IP Project Manager: Kathryn Kucharek

Manufacturing Planner: Ron Montgomery

Art and Design Direction, Production
Management, and Composition:

Lumina Datamatics, Inc.

Cover Image: © majcot/Shutterstock.com

© 2015 Cengage Learning®

WCN: 02-300

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored, or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706.

For permission to use material from this text or product,
submit all requests online at **www.cengage.com/permissions**.

Further permissions questions can be emailed to
permissionrequest@cengage.com.

Some of the product names and company names used in this book have been used for identification purposes only and may be trademarks or registered trademarks of their respective manufacturers and sellers.

Library of Congress Control Number: 2014951565

ISBN: 978-1-111-52727-3

Cengage Learning

20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at: **www.cengage.com/global**.

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

To learn more about Cengage Learning Solutions, visit **www.cengage.com**.

Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**.

Microsoft Access and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle is a registered trademark, and Oracle 11g is a trademark of Oracle Corporation.

The programs in this book are for instructional purposes only. They have been tested with care, but are not guaranteed for any particular intent beyond educational purposes. The author and the publisher do not offer any warranties or representations, nor do they accept any liabilities with respect to the programs.

Cengage Learning reserves the right to revise this publication and make changes from time to time in its content without notice.

TABLE OF CONTENTS

Preface	xi
Chapter 1 <i>Introduction to TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group</i>	1
What Is a Database?	1
The TAL Distributors Database	2
The Colonial Adventure Tours Database	8
The Solmaris Condominium Group Database	14
Chapter Summary	18
Key Term	18
Exercises	18
TAL Distributors	18
Colonial Adventure Tours	19
Solmaris Condominium Group	19
Chapter 2 <i>Database Design Fundamentals</i>	21
Database Concepts	22
Relational Databases	22
Entities, Attributes, and Relationships	23
Functional Dependence	27
Primary Keys	30
Database Design	32
Design Method	32
Database Design Requirements	34
Database Design Process Example	34
Normalization	40
First Normal Form	40
Second Normal Form	42
Third Normal Form	46
Diagrams for Database Design	51
Chapter Summary	54
Key Terms	55
Review Questions	55
Exercises	56
TAL Distributors	56
Colonial Adventure Tours	57
Solmaris Condominium Group	58

Chapter 3	<i>Creating Tables</i>	59
Creating and Running SQL Commands		59
Using the Oracle Database 11g Express Edition		60
Entering Commands		63
Creating a Table		65
Correcting Errors in SQL Commands		68
Dropping a Table		69
Using Data Types		70
Using Nulls		71
Adding Rows to a Table		72
The INSERT Command		72
Inserting a Row that Contains Nulls		74
Viewing Table Data		74
Correcting Errors in a Table		76
Saving SQL Commands		78
Creating the Remaining Database Tables		81
Describing a Table		86
Chapter Summary		88
Key Terms		88
Review Questions		88
Exercises		89
TAL Distributors		90
Colonial Adventure Tours		90
Solmaris Condominium Group		93
Chapter 4	<i>Single-Table Queries</i>	97
Constructing Simple Queries		97
Retrieving Certain Columns and All Rows		98
Retrieving All Columns and All Rows		99
Using a WHERE Clause		100
Using Compound Conditions		102
Using the BETWEEN Operator		105
Using Computed Columns		106
Using the LIKE Operator		108
Using the IN Operator		110
Sorting		110
Using the ORDER BY Clause		111
Additional Sorting Options		111
Using Functions		112
Using the COUNT Function		113
Using the SUM Function		113
Using the AVG, MAX, and MIN Functions		114
Using the DISTINCT Operator		115
Nesting Queries		117
Subqueries		118

Grouping	120
Using the GROUP BY Clause	121
Using a HAVING Clause	122
HAVING vs. WHERE	123
Nulls	125
Summary of SQL Clauses, Functions, and Operators	125
Chapter Summary	127
Key Terms	128
Review Questions	128
Exercises	129
TAL Distributors	129
Colonial Adventure Tours	130
Solmaris Condominium Group	131
Chapter 5 <i>Multiple-Table Queries</i>	133
Querying Multiple Tables	133
Joining Two Tables	134
Comparing Joins, IN, and EXISTS	137
Using the IN Operator	138
Using the EXISTS Operator	138
Using a Subquery Within a Subquery	140
A Comprehensive Example	141
Using an Alias	142
Joining a Table to Itself	143
Using a Self-Join on a Primary Key Column	145
Joining Several Tables	147
Set Operations	150
ALL and ANY	154
Special Operations	157
Inner Join	157
Outer Join	158
Product	159
Chapter Summary	161
Key Terms	162
Review Questions	162
Exercises	163
TAL Distributors	163
Colonial Adventure Tours	164
Solmaris Condominium Group	165
Chapter 6 <i>Updating Data</i>	167
Creating a New Table from an Existing Table	167
Changing Existing Data in a Table	169
Adding New Rows to an Existing Table	171
Autocommit, Commit, and Rollback	173

Transactions	174
Changing and Deleting Existing Rows	174
Executing a Rollback	176
Changing a Value in a Column to Null	177
Changing a Table's Structure	179
Making Complex Changes	185
Dropping a Table	185
Chapter Summary	187
Key Terms	187
Review Questions	187
Exercises	188
TAL Distributors	188
Colonial Adventure Tours	189
Solmaris Condominium Group	190
Chapter 7 Database Administration	191
Creating and Using Views	191
Using a View to Update Data	198
Updating Row-and-Column Subset Views	199
Updating Views Involving Joins	200
Updating Views Involving Statistics	203
Dropping a View	203
Security	204
Indexes	207
Creating an Index	210
Dropping an Index	211
Creating Unique Indexes	212
System Catalog	212
Update of the System Catalog	214
Integrity Constraints in SQL	215
Chapter Summary	219
Key Terms	220
Review Questions	220
Exercises	221
TAL Distributors	221
Colonial Adventure Tours	223
Solmaris Condominium Group	225
Chapter 8 SQL Functions and Procedures	227
Using SQL in a Programming Environment	227
Using Functions	228
Character Functions	229
Number Functions	230
Working with Dates	231

Concatenating Columns	233
Stored Procedures	235
Retrieving a Single Row and Column	235
Error Handling	238
Using Update Procedures	239
Changing Data with a Procedure	239
Deleting Data with a Procedure	240
Selecting Multiple Rows with a Procedure	241
Using a Cursor	242
Opening a Cursor	243
Fetching Rows from a Cursor	243
Closing a Cursor	245
Writing a Complete Procedure Using a Cursor	245
Using More Complex Cursors	247
Advantages of Cursors	248
Using T-SQL in SQL Server	248
Retrieving a Single Row and Column	249
Changing Data with a Stored Procedure	249
Deleting Data with a Stored Procedure	250
Using a Cursor	250
Using More Complex Cursors	251
Using SQL in Microsoft Access	252
Deleting Data with Visual Basic	252
Running the Code	253
Updating Data with Visual Basic	254
Inserting Data with Visual Basic	255
Finding Multiple Rows with Visual Basic	255
Using a Trigger	257
Chapter Summary	260
Key Terms	261
Review Questions	261
Exercises	262
TAL Distributors	262
Colonial Adventure Tours	264
Solmaris Condominium Group	265
Appendix A SQL Reference	267
Aliases	267
ALTER TABLE	267
Column or Expression List (SELECT Clause)	268
Computed Columns	268
The DISTINCT Operator	268
Functions	269
COMMIT	269

Conditions	269
Simple Conditions	269
Compound Conditions	270
BETWEEN Conditions	270
LIKE Conditions	270
IN Conditions	271
EXISTS Conditions	271
ALL and ANY	271
CREATE INDEX	272
CREATE TABLE	272
CREATE VIEW	273
Data Types	274
DELETE Rows	275
DESCRIBE	275
DROP INDEX	275
DROP TABLE	276
DROP VIEW	276
GRANT	276
INSERT INTO (Query)	277
INSERT INTO (Values)	277
Integrity	278
REVOKE	279
ROLLBACK	279
SELECT	280
Subqueries	280
UNION, INTERSECT, and MINUS	281
UPDATE	282
Appendix B <i>How Do I Reference</i>	283
Appendix C <i>Answers to Odd-Numbered Review Questions</i>	287
Chapter 1—Introduction to TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group	287
Chapter 2—Database Design Fundamentals	287
Chapter 3—Creating Tables	289
Chapter 4—Single-Table Queries	289
Chapter 5—Multiple-Table Queries	290
Chapter 6—Updating Data	290
Chapter 7—Database Administration	291
Chapter 8—SQL Functions and Procedures	291
Index	293

PREFACE

Structured Query Language (or SQL, which is pronounced “se-quel,” or “ess-cue-ell”) is a popular computer language that is used by diverse groups such as home computer users, owners of small businesses, end users in large organizations, and programmers. Although this text uses the SQL implementation in the Oracle Database 11g Express Edition as a vehicle for teaching SQL, its chapter material, examples, and exercises can be completed using any SQL implementation.

A Guide to SQL, Ninth Edition is written for a wide range of teaching levels, from students taking introductory computer science classes to those students in advanced information systems classes. This text can be used for a standalone course on SQL or in conjunction with a database concepts text where students are required to learn SQL.

The chapters in this text should be covered in order. Students should complete the end-of-chapter exercises and the examples within the chapters for maximum learning. Because the content of Chapter 8 assumes that the reader has had instruction or experience in at least one programming language, the instructor should determine whether students will understand its concepts. Students without a programming background will have difficulty understanding the topic of embedded SQL. Instructors can easily omit Chapter 8 from the text in situations where students are not comfortable with programming examples.

The Ninth Edition builds on the success of previous editions by presenting basic SQL commands in the context of a business that uses SQL to manage orders, items, customers, and sales reps. Like in previous editions, this edition uses Oracle as the vehicle to present SQL commands. Like the last edition, this edition addresses SQL in Access™ by showing the Access versions of the same commands when they differ from the Oracle versions. This edition also shows SQL Server commands when they differ from the Oracle versions. Differences for Access and SQL Server users are featured in “User” notes, which make it easy for students to identify differences for the SQL implementation they are using. Students can download the Oracle 11g Database Express Edition from the Oracle Web site for free and use it to complete this text without having to purchase or install the full Oracle program.

The Ninth Edition includes an entire chapter on database design, showing students how to create an appropriate design that satisfies a given set of requirements, and includes coverage of the important topics of stored procedures and triggers. The text also contains updated case problems that feature a distributor of toys, an outdoor adventure company, and a condominium management company.

DISTINGUISHING FEATURES

Use of Examples

Each chapter contains multiple examples that use SQL to solve a problem. Following each example, students will read about the commands that are used to solve the stated problem, and then they will see the SQL commands used to arrive at the solution. For most students, learning through examples is the most effective way to master material. For this reason, instructors should encourage students to read the chapters at the computer and input the commands shown in the figures.

Case Studies

A running case study—TAL Distributors—is presented in all of the examples within the chapters and in the first set of exercises at the end of each chapter. Although the database is small in order to be manageable, the examples and exercises for the TAL Distributors database simulate what a real business can accomplish using SQL commands. Using the same case study as examples within the chapter and in the end-of-chapter exercises ensures a high level of continuity to reinforce learning.

A second case study—the Colonial Adventure Tours database—is used in a second set of exercises at the end of each chapter. A third case study—the Solmaris Condominium Group database—is used in a third set of exercises at the end of each chapter. The second and third case studies give students a chance to venture out “on their own” without the direct guidance of examples from the text.

Question and Answer Sections

A special type of exercise, called a Q&A, is used throughout the book. These exercises force students to consider special issues and understand important questions before continuing with their study. The answer to each Q&A appears after the question. Students are encouraged to formulate their own answers before reading the ones provided in the text to ensure that they understand new material before proceeding.

“User” Notes for Access™ and SQL Server Users

When an SQL command has a different use or format in Access or SQL Server, it appears in a User note. When you encounter a User note for the SQL implementation you are using, be sure to read its contents. You might also review the User notes for other SQL implementations so you are aware of the differences that occur from one implementation of SQL to another.

Review Material

A Summary and Key Terms list appear at the end of each chapter, followed by Review Questions that test students’ recall of the important points in the chapter and occasionally test their ability to apply what they have learned. The answers to the

odd-numbered Review Questions are provided in Appendix C. Each chapter also contains exercises related to the TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group databases. Critical-thinking questions that reinforce problem-solving and analytical skills are included for review questions and hands-on exercises.

Appendices

Three appendices appear at the end of this text. Appendix A is an SQL reference that describes the purpose and syntax for the major SQL commands featured in the text. Students can use Appendix A to identify how and when to use important commands quickly. The SQL reference appendix contains references to specific pages in the text where the command is discussed to make it easy for students to find additional information when they need to refer back to the section in the book where the topic is covered.

Appendix B includes a “How Do I” reference, which lets students cross-reference the appropriate section in Appendix A by searching for the answer to a question. Appendix C includes answers to the odd-numbered Review Questions.

Relationship to Concepts of Database Management, Eighth Edition

For database courses featuring SQL, this SQL text can be bundled with *Concepts of Database Management, Eighth Edition* by Pratt and Last (Cengage Learning). The data and pedagogy between the two texts is consistent, and the instructor’s manuals for both books include feedback and suggestions for using the texts together.

Instructor Support

The Ninth Edition includes a package of proven supplements for instructors and students. The Instructor’s Resources offer a detailed electronic Instructor’s Manual, figure files, Microsoft® PowerPoint® presentations, and the Cognero Test Bank. The Instructor’s Manual includes suggestions and strategies for using this text, as well as answers to Review Questions and solutions to the end-of-chapter exercises. Figure files allow instructors to create their own presentations using figures appearing in the text. Instructors can also take advantage of lecture presentations provided on PowerPoint slides; these presentations follow each chapter’s coverage precisely, include chapter figures, and can be customized.

The Instructor’s Resources include copies of the databases for the TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group cases in Microsoft Access 2010/2013 format and script files to create the tables and data in these databases in Oracle and SQL Server. These files are provided so instructors have the choice of assigning exercises in which students create the databases used in this text and load them with data, or they can provide the starting Access databases or Oracle or SQL Server script files to students to automate and simplify these tasks.

ORGANIZATION OF THE TEXT

The text contains eight chapters and three appendices, which are described in the following sections.

Chapter 1: Introduction to TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group

Chapter 1 introduces the three database cases that are used throughout the text: TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group. Many Q&A exercises are provided throughout the chapter to ensure that students understand how to manipulate the database on paper before they begin working in SQL.

Chapter 2: Database Design Fundamentals

Chapter 2 covers important concepts and terminology associated with relational databases, functional dependence, and primary keys, followed by a method for designing a database to satisfy a given set of requirements. It also illustrates the normalization process for finding and correcting a variety of potential problems in database designs. Finally, it shows how to represent database designs graphically using entity-relationship diagrams.

Chapter 3: Creating Tables

In Chapter 3, students begin using a DBMS by creating and running SQL commands to create tables, use data types, and add rows to tables. Chapter 3 also discusses the role of and use of nulls.

Chapter 4: Single-Table Queries

Chapter 4 is the first of two chapters on using SQL commands to query a database. The queries in Chapter 4 all involve single tables. Included in this chapter are discussions of simple and compound conditions; computed columns; the SQL BETWEEN, LIKE, and IN operators; using SQL aggregate functions; nesting queries; grouping data; and retrieving columns with null values.

Chapter 5: Multiple-Table Queries

Chapter 5 completes the discussion of querying a database by demonstrating queries that join more than one table. Included in this chapter are discussions of the SQL IN and EXISTS operators, nested subqueries, using aliases, joining a table to itself, SQL set operations, and the use of the ALL and ANY operators. The chapter also includes coverage of various types of joins.

Chapter 6: Updating Data

In Chapter 6, students learn how to use the SQL COMMIT, ROLLBACK, UPDATE, INSERT, and DELETE commands to update table data. Students also learn how to

create a new table from an existing table and how to change the structure of a table. The chapter also includes coverage of transactions, including both their purpose and implementation.

Chapter 7: Database Administration

Chapter 7 covers the database administration features of SQL, including the use of views; granting and revoking database privileges to users; creating, dropping, and using an index; using and obtaining information from the system catalog; and using integrity constraints to control data entry.

Chapter 8: SQL Functions and Procedures

Chapter 8 begins with a discussion of some important SQL functions that act on single rows. Students will also learn how to use PL/SQL and T-SQL to cover the process of embedding SQL commands in another language. Included in this chapter are discussions of using embedded SQL to insert new rows and change and delete existing rows. Also included is a discussion of how to retrieve single rows using embedded SQL commands and how to use cursors to retrieve multiple rows. Chapter 8 also includes a section showing some techniques for using SQL in Visual Basic (Access). The chapter concludes with a discussion of triggers.

Appendix A: SQL Reference

Appendix A includes a command reference for all the major SQL clauses and operators that are featured in the chapters. Students can use Appendix A as a quick resource when constructing commands. Each command includes a short description, a table that shows the required and optional clauses and operators, and an example and its results. It also contains a reference to the pages in the text where the command is covered.

Appendix B: How Do I Reference

Appendix B provides students with an opportunity to ask a question, such as “How do I delete rows?” and to identify the appropriate section in Appendix A to use to find the answer. Appendix B is extremely valuable when students know what task they want to accomplish but can’t remember the exact SQL command they need.

Appendix C: Answers to Odd-Numbered Review Questions

Answers to the odd-numbered Review Questions in each chapter appear in this appendix so students can make sure that they are completing the Review Questions correctly.

GENERAL NOTES TO THE STUDENT

1. You can download the databases used in this text from www.cengage.com. Use your browser to go to www.CengageBrain.com.

2. The first time you go to the site, you will need to register. It's free. Click on "Sign Up" in the top right corner of the page and fill out the registration information. (After you have signed in once, whenever you return to CengageBrain, you will enter the user name and password you have chosen and you will be taken directly to the companion site for your book.)
3. Once you have registered and logged in for the first time, go to the "Search for Books or Materials" bar and enter the author or ISBN for your textbook. When the title of your text appears, click on it and you will be taken to the companion site. There you can choose among the various folders provided on the Student side of the site. NOTE: If you are currently using more than one Cengage textbook, the same user name and password will give you access to all the companion sites for your Cengage titles. After you have entered the information for each title, all the titles you are using will appear listed in the pull-down menu in the "Search for Books or Materials" bar. Whenever you return to CengageBrain, you can click on the title of the site you wish to visit and go directly there.

The script files saved in the Oracle folder have the following functions:

OracleSolmaris.sql: Creates all the tables in the Solmaris Condominium Group database and adds all the data. Run this script file to create the Solmaris Condominium Group database. (Note: This script file assumes you have not previously created any of the tables in the database. If you have created any of the tables, you should run the OracleDropSolmaris.sql script prior to running the OracleSolmaris.sql script.)

OracleColonial.sql: Creates all the tables in the Colonial Adventure Tours database and adds all the data. Run this script file to create the Colonial Adventure Tours database. (Note: This script file assumes you have not previously created any of the tables in the database. If you have created any of the tables, you should run the OracleDropColonial.sql script prior to running the OracleColonial.sql script.)

OracleTAL.sql: Creates all the tables in the TAL Distributors database and adds all the data. Run this script file to create the TAL Distributors database. (Note: This script file assumes you have not previously created any of the tables in the database. If you have created any of the tables, you should run the OracleDropTAL.sql script prior to running the OracleTAL.sql script.)

OracleDropSolmaris.sql: Drops (deletes) all the tables and data in the Solmaris Condominium Group database.

OracleDropColonial.sql: Drops (deletes) all the tables and data in the Colonial Adventure Tours database.

OracleDropTAL.sql: Drops (deletes) all the tables and data in the TAL Distributors database.

The script files saved in the SQL Server folder have the following functions:

SQLServerSolmaris.sql: Creates all the tables in the Solmaris Condominium Group database and adds all the data. Run this script file to create the Solmaris Condominium Group database. (*Note:* This script file assumes you have not previously created any of the tables in the database. If you have created any of the tables, you should run the SQLServerDropSolmaris.sql script prior to running the SQLServerSolmaris.sql script.)

SQLServerColonial.sql: Creates all the tables in the Colonial Adventure Tours database and adds all the data. Run this script file to create the Colonial Adventure Tours database. (*Note:* This script file assumes you have not previously created any of the tables in the database. If you have created any of the tables, you should run the SQLServerDropColonial.sql script prior to running the SQLServerColonial.sql script.)

SQLServerTAL.sql: Creates all the tables in the TAL Distributors database and adds all the data. Run this script file to create the TAL Distributors database. (*Note:* This script file assumes you have not previously created any of the tables in the database. If you have created any of the tables, you should run the SQLServerDropTAL.sql script prior to running the SQLServerTAL.sql script.)

SQLServerDropSolmaris.sql: Drops (deletes) all the tables and data in the Solmaris Condominium Group database.

SQLServerDropColonial.sql: Drops (deletes) all the tables and data in the Colonial Adventure Tours database.

SQLServerDropTAL.sql: Drops (deletes) all the tables and data in the TAL Distributors database.

For details on running script files in Oracle or SQL Server, check with your instructor. You can also refer to Chapter 3 in the text for information about creating and using scripts.

For information about downloading the Oracle Database 11g Express Edition software, please visit the Oracle Web site. For information about SQL Server Express, please visit the Microsoft Web site. Information about Microsoft Access is also available at the Microsoft Web site.

Embedded Questions

In many places, you'll find Q&A sections to ensure that you understand some crucial material before you proceed. In some cases, the questions are designed to give you the chance to consider some special concept in advance of its actual presentation. In all cases, the answer to each question appears immediately after the question. You can simply read the question and its answer, but you will benefit from taking time to determine the answer to the question before checking your answer against the one given in the text.

End-of-Chapter Material

The end-of-chapter material consists of a Summary, a Key Terms list, Review Questions, and exercises for the TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group databases. The Summary briefly describes the material covered in the chapter. The Review Questions require you to recall and apply the important material in the chapter. The answers to the odd-numbered Review Questions appear in Appendix C so you can check your progress. The TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group exercises test your knowledge of the chapter material; your instructor will assign one or more of these exercises for you to complete. Critical-thinking questions that reinforce problem-solving and analytical skills are included for review questions and hands-on exercises.

ACKNOWLEDGMENTS

We would like to acknowledge several individuals for their contributions in the preparation of this text. The efforts of the following members of the staff at Cengage Learning have been invaluable and have made this text possible: Joe Sabatino, Product Director; Jason Guyler Product Manager; Wendy Langerued, Content Developer; Arul Joseph Raj, Project Manager; and GreenPen Quality Assurance testers.

We have once again had the great pleasure to work with an absolutely amazing Developmental Editor, Jessica Evans, on several books. Thanks for all your efforts, Jess. You're the best! We've said it before, but it is just as true as ever!

CHAPTER 1

INTRODUCTION TO TAL DISTRIBUTORS, COLONIAL ADVENTURE TOURS, AND SOLMARIS CONDOMINIUM GROUP

OBJECTIVES

- Introduce TAL Distributors, a company whose database is used as the basis for many of the examples throughout the text
- Introduce Colonial Adventure Tours, a company whose database is used as a case that runs throughout the text
- Introduce Solmaris Condominium Group, a company whose database is used as an additional case that runs throughout the text

INTRODUCTION

In this chapter, you will examine the database requirements of TAL Distributors, a company that will be used in the examples throughout the text. Then you will examine the database requirements for Colonial Adventure Tours and Solmaris Condominium Group, whose databases are featured in the exercises that appear at the end of each chapter.

WHAT IS A DATABASE?

Throughout this text, you will work with databases for three organizations: TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group. A **database** is a structure that contains different categories of information and the relationships between these categories. The TAL Distributors database, for example, contains information about categories such as

sales representatives (sales reps), customers, orders, and items. The Colonial Adventure Tours database contains information about categories such as trips, guides, customers, and reservations. The Solmaris Condominium Group database contains information about categories such as condominiums, condo units, condo owners, service categories, and service requests.

Each database also contains relationships between categories. For example, the TAL Distributors database contains information that relates sales reps to the customers they represent and customers to the orders they have placed. The Colonial Adventure Tours database contains information that relates customers to the reservations they make and guides to the trips they lead. The Solmaris Condominium Group database contains information that relates the condo units in each condo location to their owners.

As you work through the chapters in this text, you will learn more about these databases and how to view and update the information they contain. As you read each chapter, you will see examples from the TAL Distributors database. At the end of each chapter, your instructor might assign the exercises for the TAL Distributors, Colonial Adventure Tours, or Solmaris Condominium Group databases.

THE TAL DISTRIBUTORS DATABASE

The management of TAL Distributors, a wholesaler of finely crafted wooden toys, games, and puzzles, has determined that the company's recent growth no longer makes it feasible to maintain customer, order, and inventory data using its manual systems. With the data stored in a database, management will be able to ensure that the data is up to date and more accurate than in the current manual systems. In addition, managers will be able to obtain answers to their questions concerning the data in the database easily and quickly, with the option of producing a variety of useful reports.

Management has determined that TAL Distributors must maintain the following information about its sales reps, customers, and inventory in the new database:

- The number, last name, first name, address, total commission, and commission rate for each sales rep
- The customer number, name, address, current balance, and credit limit for each customer, as well as the number of the sales rep who represents the customer
- The item number, description, number of units on hand, category, number of the storehouse where the item is stored, and unit price for each item in inventory

TAL Distributors also must store information about orders. Figure 1-1 shows a sample order.

TAL Distributors
 555 Main Street
 San Rita, TX 78364
 Phone (512) 555-0190 Fax (512) 555-0191

INVOICE

ORDER: 51610
 DATE: October 12, 2015

TO:
 CUSTOMER: 334
 The Everything Shop
 342 Magee St.
 Congaree, CA 90097

SALES REP: 45
 Hui Tian

ITEM NUMBER	DESCRIPTION	NUMBER ORDERED	PRICE	TOTAL
KL78	Pick Up Sticks	25	10.95	273.75
TR40	Tic Tac Toe	10	13.99	139.90
TOTAL				413.65

Make all checks payable to TAL Distributors
 Total due in 15 days. Overdue accounts subject to a service charge of 1% per month.

Thank you for your business!

FIGURE 1-1 Sample order

The sample order shown in Figure 1-1 has three sections:

- The heading (top) of the order contains the company name; the order number and date; the customer's number, name, and address; and the sales rep's number and name.
- The body of the order contains one or more order lines, sometimes called line items. Each order line contains an item number, an item description, the number of units of the item ordered, and the quoted price for the item. Each order line also contains a total, usually called an extension, which is the result of multiplying the number ordered by the quoted price.
- Finally, the footing (bottom) of the order contains the order total.

TAL Distributors also must store the following items in the database for each customer's order:

- For each order, the database must store the order number, the date the order was placed, and the number of the customer that placed the order. The customer's name and address and the number of the sales rep who represents the customer are stored with the customer information. The name of the sales rep is stored with the sales rep information.
- For each order, the database must store the order number, the item number, the number of units ordered, and the quoted price for each order line. The item description is stored with the information about items. The result of multiplying the number of units ordered by the quoted price is not stored because the database can calculate it when needed.

The overall order total is not stored. Instead, the database calculates the total whenever an order is printed or displayed on the screen.

Figure 1-2 shows sample data for TAL Distributors.

REP

REP_NUM	LAST_NAME	FIRST_NAME	STREET	CITY	STATE	POSTAL_CODE	COMMISSION	RATE
15	Campos	Rafael	724 Vinca Dr.	Grove	CA	90092	\$23,457.50	0.06
30	Gradey	Megan	632 Liatris St.	Fullton	CA	90085	\$41,317.00	0.08
45	Tian	Hui	1785 Tyler Ave.	Northfield	CA	90098	\$27,789.25	0.06
60	Sefton	Janet	267 Oakley St.	Congaree	CA	90097	\$0.00	0.06

CUSTOMER

CUSTOMER_NUM	CUSTOMER_NAME	STREET	CITY	STATE	POSTAL_CODE	BALANCE	CREDIT_LIMIT	REP_NUM
126	Toys Galore	28 Laketon St.	Fullton	CA	90085	\$1,210.25	\$7,500.00	15
260	Brookings Direct	452 Columbus Dr.	Grove	CA	90092	\$575.00	\$10,000.00	30
334	The Everything Shop	342 Magee St.	Congaree	CA	90097	\$2,345.75	\$7,500.00	45
386	Johnson's Department Store	124 Main St.	Northfield	CA	90098	\$879.25	\$7,500.00	30
440	Grove Historical Museum Store	3456 Central Ave.	Fullton	CA	90085	\$345.00	\$5,000.00	45
502	Cards and More	167 Hale St.	Mesa	CA	90104	\$5,025.75	\$5,000.00	15
586	Almondton General Store	3345 Devon Ave.	Almondton	CA	90125	\$3,456.75	\$15,000.00	45
665	Cricket Gift Shop	372 Oxford St.	Grove	CA	90092	\$678.90	\$7,500.00	30
713	Cress Store	12 Rising Sun Ave.	Congaree	CA	90097	\$4,234.60	\$10,000.00	15
796	Unique Gifts	786 Passmore St.	Northfield	CA	90098	\$124.75	\$7,500.00	45
824	Kline's	945 Gilham St.	Mesa	CA	90104	\$2,475.99	\$15,000.00	30
893	All Season Gifts	382 Wildwood Ave.	Fullton	CA	90085	\$935.75	\$7,500.00	15

FIGURE 1-2 Sample data for TAL Distributors

ORDERS

ORDER_NUM	ORDER_DATE	CUSTOMER_NUM
51608	10/12/2015	126
51610	10/12/2015	334
51613	10/13/2015	386
51614	10/13/2015	260
51617	10/15/2015	586
51619	10/15/2015	126
51623	10/15/2015	586
51625	10/16/2015	796

ORDER_LINE

ORDER_NUM	ITEM_NUM	NUM_ORDERED	QUOTED_PRICE
51608	CD33	5	\$86.99
51610	KL78	25	\$10.95
51610	TR40	10	\$13.99
51613	DL51	5	\$104.95
51614	FD11	1	\$124.95
51617	NL89	4	\$115.99
51617	TW35	3	\$116.95
51619	FD11	2	\$121.95
51623	DR67	5	\$29.95
51623	FH24	12	\$36.95
51623	KD34	10	\$13.10
51625	MT03	8	\$45.79

ITEM

ITEM_NUM	DESCRIPTION	ON_HAND	CATEGORY	STOREHOUSE	PRICE
AH74	Patience	9	GME	3	\$22.99
BR23	Skittles	21	GME	2	\$29.99
CD33	Wood Block Set (48 piece)	36	TOY	1	\$89.49
DL51	Classic Railway Set	12	TOY	3	\$107.95
DR67	Giant Star Brain Teaser	24	PZL	2	\$31.95
DW23	Mancala	40	GME	3	\$50.00
FD11	Rocking Horse	8	TOY	3	\$124.95
FH24	Puzzle Gift Set	65	PZL	1	\$38.95
KA12	Cribbage Set	56	GME	3	\$75.00
KD34	Pentominoes Brain Teaser	60	PZL	2	\$14.95
KL78	Pick Up Sticks	110	GME	1	\$10.95
MT03	Zauberkasten Brain Teaser	45	PZL	1	\$45.79
NL89	Wood Block Set (62 piece)	32	TOY	3	\$119.75
TR40	Tic Tac Toe	75	GME	2	\$13.99
TW35	Fire Engine	30	TOY	2	\$118.95

FIGURE 1-2 Sample data for TAL Distributors (*continued*)

In the REP table, you see that there are four reps, whose numbers are 15, 30, 45, and 60. The name of sales rep 15 is Rafael Campos. His street address is 724 Vinca Dr. He lives in Grove, CA, and his postal code is 90092. His total commission is \$23,457.50, and his commission rate is 0.06 (six percent).

In the CUSTOMER table, 12 TAL Distributors customers are identified with the numbers 126, 260, 334, 386, 440, 502, 586, 665, 713, 796, 824, and 893. The name of customer number 126 is Toys Galore. This customer's address is 28 Laketon St. in Fullton, CA, with a postal code of 90085. The customer's current balance is \$1,210.25, and its credit limit is \$7,500.00. The number 15 in the REP_NUM column indicates that Toys Galore is represented by sales rep 15 (Rafael Campos).

Skipping to the table named ITEM, you see that there are 15 items, whose item numbers are AH74, BR23, CD33, DL51, DR67, DW23, FD11, FH24, KA12, KD34, KL78, MT03, NL89, TR40, and TW35. Item AH74 is Patience, and TAL Distributors has nine units of this item on hand. The Patience item is in the GME (games) category, and is located in storehouse 3. The price of the Patience game is \$22.99. Other categories are PZL (puzzles) and TOY (toys).

Moving back to the table named ORDERS, you see that there are eight orders, which are identified with the numbers 51608, 51610, 51613, 51614, 51617, 51619, 51623, and 51625. Order number 51608 was placed on October 12, 2015, by customer 126 (Toys Galore).

NOTE

In some database systems, the word *order* has a special purpose. Having a table named ORDER could cause problems in such systems. For this reason, TAL Distributors uses the table name ORDERS instead of ORDER.

The table named ORDER_LINE might seem strange at first glance. Why do you need a separate table for the order lines? Could they be included in the ORDERS table? The answer is technically yes. You could structure the table named ORDERS as shown in Figure 1-3. Notice that this table contains the same orders as shown in Figure 1-2, with the same dates and customer numbers. In addition, each table row in Figure 1-3 contains all the order lines for a given order. Examining the second row, for example, you see that order 51610 has two order lines. One of the order lines is for 25 units of item KL78 at \$10.95 each, and the other order line is for 10 units of item TR40 at \$13.99 each.

ORDERS

ORDER_NUM	ORDER_DATE	CUSTOMER_NUM	ITEM_NUM	NUM_ORDERED	QUOTED_PRICE
51608	10/12/2015	126	CD33	5	\$86.99
51610	10/12/2015	334	KL78	25	\$10.95
			TR40	10	\$13.99
51613	10/13/2015	386	DL51	5	\$104.95
51614	10/13/2015	260	FD11	1	\$124.95
51617	10/15/2015	586	NL89	4	\$115.99
			TW35	3	\$116.95
51619	10/15/2015	126	FD11	2	\$121.95
51623	10/15/2015	586	DR67	5	\$29.95
			FH24	12	\$36.95
			KD34	10	\$13.10
51625	10/16/2015	796	MT03	8	\$45.79

FIGURE 1-3 Alternative ORDERS table structure

Q & A

Question: How is the information from Figure 1-2 represented in Figure 1-3?

Answer: Examine the ORDER_LINE table shown in Figure 1-2 and note the second and third rows. The second row indicates that there is an order line on order 51610 for 25 units of item KL78 at \$10.95 each. The third row indicates that there is an order line on order 51610 for 10 units of item TR40 at \$13.99 each. Thus, the information that you find in Figure 1-3 is represented in Figure 1-2 in two separate rows rather than in one row.

It might seem inefficient to use two rows to store information that could be represented in one row. There is a problem, however, with the arrangement shown in Figure 1-3—the table is more complicated. In Figure 1-2, there is a single entry at each location in the table. In Figure 1-3, some of the individual positions within the table contain multiple entries, making it difficult to track the information between columns. In the row for order number 51610, for example, it is crucial to know that the KL78 corresponds to the 25 in the NUM_ORDERED column (not to the 10) and that it corresponds to the \$10.95 in the QUOTED_PRICE column (not to the \$13.99). In addition, a more complex table raises practical issues, such as:

- How much room do you allow for these multiple entries?
- What happens when an order has more order lines than you have allowed room for?
- For a given item, how do you determine which orders contain order lines for that item?

Although none of these problems is unsolvable, they do add a level of complexity that is not present in the arrangement shown in Figure 1-2. In Figure 1-2, there are no multiple entries to worry about, it does not matter how many order lines exist for any order, and finding every order that contains an order line for a given item is easy (just look for all order lines with the given item number in the ITEM_NUM column). In general, this simpler structure is preferable, and that is why order lines appear in a separate table.

To test your understanding of the TAL Distributors data, use Figure 1-2 to answer the following questions.

Q & A

Question: What are the numbers of the customers represented by Rafael Campos?

Answer: 126, 502, 713, and 893. (Look up the REP_NUM value of Rafael Campos in the REP table and obtain the number 15. Then find all customers in the CUSTOMER table that have the number 15 in the REP_NUM column.)

Q & A

Question: What is the name of the customer that placed order 51613, and what is the name of the rep who represents this customer?

Answer: Johnson's Department Store is the customer, and Megan Gradey is the rep. (Look up the CUSTOMER_NUM value in the ORDERS table for order number 51613 and obtain the number 386. Then find the customer in the CUSTOMER table with the CUSTOMER_NUM value of 386. Using the REP_NUM value, which is 30, find the name of the rep in the REP table.)

Q & A

Question: List all items that appear in order 51617. For each item, give the description, number ordered, and quoted price.

Answer: Item number: NL89; description: Wood Block Set (62 piece), number ordered: 4; and quoted price: \$115.99. Also, item number: TW35; description: Fire Engine; number ordered: 3; and quoted price: \$116.95. (Look up each ORDER_LINE table row on which the order number is 51617. Each of these rows contains an item number, the number ordered, and the quoted price. Use the item number to look up the corresponding item description in the ITEM table.)

Q & A

Question: Why is the QUOTED_PRICE column in the ORDER_LINE table? Can't you just use the item number and look up the price in the ITEM table?

Answer: If the QUOTED_PRICE column did not appear in the ORDER_LINE table, you would need to obtain the price for an item on an order line by looking up the price in the ITEM table. Although this approach is reasonable, it prevents TAL Distributors from charging different prices to different customers for the same item. Because TAL Distributors wants the flexibility to quote and charge different prices to different customers, the QUOTED_PRICE column is included in the ORDER_LINE table. If you examine the ORDER_LINE table, you will see cases in which the quoted price matches the actual price in the ITEM table and cases in which it differs. For example, in order number 51608, Toys Galore bought five Wood Block Sets (48 piece), and TAL Distributors charged only \$86.99 per set, and not the regular price of \$89.49.

THE COLONIAL ADVENTURE TOURS DATABASE

Colonial Adventure Tours is a small business that organizes day-long, guided trips to New England. Like the management of TAL Distributors, Colonial Adventure Tours has decided to store its data in a database. The company wants to achieve the same benefits; that is, it wants to ensure that its data is current and accurate. In addition, the management of

Colonial Adventure Tours wants to be able to ask questions concerning the data and to obtain answers to these questions easily and quickly.

In running the guided tours, management gathers and organizes information about guides, trips, customers, and reservations. Figure 1-4 shows sample guide data for Colonial Adventure Tours. Each guide has a number that uniquely identifies the guide. In addition, Colonial Adventure Tours records the guide's last name, first name, address, city, state, postal code, telephone number, and hire date.

GUIDE

GUIDE_NUM	LAST_NAME	FIRST_NAME	ADDRESS	CITY	STATE	POSTAL_CODE	PHONE_NUM	HIRE_DATE
AM01	Abrams	Miles	54 Quest Ave.	Williamsburg	MA	01096	617-555-6032	6/3/2012
BR01	Boyers	Rita	140 Oakton Rd.	Jaffrey	NH	03452	603-555-2134	3/4/2012
DH01	Devon	Harley	25 Old Ranch Rd.	Sunderland	MA	01375	781-555-7767	1/8/2012
GZ01	Gregory	Zach	7 Moose Head Rd.	Dummer	NH	03588	603-555-8765	11/4/2012
KS01	Kiley	Susan	943 Oakton Rd.	Jaffrey	NH	03452	603-555-1230	4/8/2013
KS02	Kelly	Sam	9 Congaree Ave.	Fraconia	NH	03580	603-555-0003	6/10/2013
MR01	Marston	Ray	24 Shenandoah Rd.	Springfield	MA	01101	781-555-2323	9/14/2015
RH01	Rowan	Hal	12 Heather Rd.	Mount Desert	ME	04660	207-555-9009	6/2/2014
SL01	Stevens	Lori	15 Riverton Rd.	Coventry	VT	05825	802-555-3339	9/5/2014
UG01	Unser	Glory	342 Pineview St.	Danbury	CT	06810	203-555-8534	2/2/2015

FIGURE 1-4 Sample guide data for Colonial Adventure Tours

Figure 1-5 shows sample trip data for Colonial Adventure Tours. Each trip has a number that uniquely identifies the trip. In addition, management tracks the trip name, the trip's starting location, the state in which the trip originates, the trip's total distance (in miles), the trip's maximum group size, the trip's type, and the trip's season.

TRIP

TRIP_ID	TRIP_NAME	START_LOCATION	STATE	DISTANCE	MAX_GRP_SIZE	TYPE	SEASON
1	Arethusa Falls	Harts Location	NH	5	10	Hiking	Summer
2	Mt Ascutney - North Peak	Weathersfield	VT	5	6	Hiking	Late Spring
3	Mt Ascutney - West Peak	Weathersfield	VT	6	10	Hiking	Early Fall
4	Bradbury Mountain Ride	Lewiston-Auburn	ME	25	8	Biking	Early Fall
5	Baldpate Mountain	North Newry	ME	6	10	Hiking	Late Spring
6	Blueberry Mountain	Batchelders Grant	ME	8	8	Hiking	Early Fall
7	Bloomfield - Maidstone	Bloomfield	CT	10	6	Paddling	Late Spring
8	Black Pond	Lincoln	NH	8	12	Hiking	Summer
9	Big Rock Cave	Tamworth	NH	6	10	Hiking	Summer
10	Mt. Cardigan - Firescrew	Orange	NH	7	8	Hiking	Summer
11	Chocorua Lake Tour	Tamworth	NH	12	15	Paddling	Summer
12	Cadillac Mountain Ride	Bar Harbor	ME	8	16	Biking	Early Fall
13	Cadillac Mountain	Bar Harbor	ME	7	8	Hiking	Late Spring
14	Cannon Mtn	Franconia	NH	6	6	Hiking	Early Fall
15	Crawford Path Presidentials Hike	Crawford Notch	NH	16	4	Hiking	Summer
16	Cherry Pond	Whitefield	NH	6	16	Hiking	Spring
17	Huguenot Head Hike	Bar Harbor	ME	5	10	Hiking	Early Fall
18	Low Bald Spot Hike	Pinkam Notch	NH	8	6	Hiking	Early Fall
19	Mason's Farm	North Stratford	CT	12	7	Paddling	Late Spring
20	Lake Mephremagog Tour	Newport	VT	8	15	Paddling	Late Spring
21	Long Pond	Rutland	MA	8	12	Hiking	Summer
22	Long Pond Tour	Greenville	ME	12	10	Paddling	Summer
23	Lower Pond Tour	Poland	ME	8	15	Paddling	Late Spring
24	Mt Adams	Randolph	NH	9	6	Hiking	Summer
25	Mount Battie Ride	Camden	ME	20	8	Biking	Early Fall
26	Mount Cardigan Hike	Cardigan	NH	4	16	Hiking	Late Fall
27	Mt. Chocorua	Albany	NH	6	10	Hiking	Spring
28	Mount Garfield Hike	Woodstock	NH	5	10	Hiking	Early Fall
29	Metacomet-Monadnock Trail Hike	Pelham	MA	10	12	Hiking	Late Spring
30	McLennan Reservation Hike	Tyringham	MA	6	16	Hiking	Summer
31	Missisquoi River - VT	Lowell	VT	12	10	Paddling	Summer
32	Northern Forest Canoe Trail	Stark	NH	15	10	Paddling	Summer
33	Park Loop Ride	Mount Desert Island	ME	27	8	Biking	Late Spring
34	Pontook Reservoir Tour	Dummer	NH	15	14	Paddling	Late Spring
35	Pisgah State Park Ride	Northborough	NH	12	10	Biking	Summer
36	Pondicherry Trail Ride	White Mountains	NH	15	16	Biking	Late Spring
37	Seal Beach Harbor	Bar Harbor	ME	5	16	Hiking	Early Spring
38	Sawyer River Ride	Mount Carrigain	NH	10	18	Biking	Early Fall
39	Welch and Dickey Mountains Hike	Thorton	NH	5	10	Hiking	Summer
40	Wachusett Mountain	Princeton	MA	8	8	Hiking	Early Spring
41	Westfield River Loop	Fort Fairfield	ME	20	10	Biking	Late Spring

FIGURE 1-5 Sample trip data for Colonial Adventure Tours

Figure 1-6 shows sample customer data for Colonial Adventure Tours. Each customer is identified by a unique customer number. In addition, management stores each customer's last name, first name, address, city, state, postal code, and telephone number.

CUSTOMER

CUSTOMER_NUM	LAST_NAME	FIRST_NAME	ADDRESS	CITY	STATE	POSTAL_CODE	PHONE
101	Northfold	Liam	9 Old Mill Rd.	Londonderry	NH	03053	603-555-7563
102	Ocean	Arnold	2332 South St. Apt 3	Springfield	MA	01101	413-555-3212
103	Kasuma	Sujata	132 Main St. #1	East Hartford	CT	06108	860-555-0703
104	Goff	Ryan	164A South Bend Rd.	Lowell	MA	01854	781-555-8423
105	McLean	Kyle	345 Lower Ave.	Wolcott	NY	14590	585-555-5321
106	Morontoia	Joseph	156 Scholar St.	Johnston	RI	02919	401-555-4848
107	Marchand	Quinn	76 Cross Rd.	Bath	NH	03740	603-555-0456
108	Rulf	Uschi	32 Sheep Stop St.	Edinboro	PA	16412	814-555-5521
109	Caron	Jean Luc	10 Greenfield St.	Rome	ME	04963	207-555-9643
110	Bers	Martha	65 Granite St.	York	NY	14592	585-555-0111
112	Jones	Laura	373 Highland Ave.	Somerville	MA	02143	857-555-6258
115	Vaccari	Adam	1282 Ocean Walk	Ocean City	NJ	08226	609-555-5231
116	Murakami	Iris	7 Cherry Blossom St.	Weymouth	MA	02188	617-555-6665
119	Chau	Clement	18 Ark Ledge Ln.	Londonderry	VT	05148	802-555-3096
120	Gernowski	Sadie	24 Stump Rd.	Athens	ME	04912	207-555-4507
121	Bretton-Borak	Siam	10 Old Main St.	Cambridge	VT	05444	802-555-3443
122	Hefferson	Orlauh	132 South St. Apt 27	Manchester	NH	03101	603-555-3476
123	Barnett	Larry	25 Stag Rd.	Fairfield	CT	06824	860-555-9876
124	Busa	Karen	12 Foster St.	South Windsor	CT	06074	857-555-5532
125	Peterson	Becca	51 Fredrick St.	Albion	NY	14411	585-555-0900
126	Brown	Brianne	154 Central St.	Vernon	CT	06066	860-555-3234

FIGURE 1-6 Sample customer data for Colonial Adventure Tours

Figure 1-7 shows sample reservations data for Colonial Adventure Tours. Each reservation is identified by a unique reservation number that uses the last two digits of the current year followed by a five-digit number that is incremented sequentially as each reservation is received. The table also stores the trip number, the trip date, the number of persons, the trip price per person, any additional fees for transportation and equipment rentals, and the customer number.

RESERVATION

RESERVATION_ID	TRIP_ID	TRIP_DATE	NUM_PERSONS	TRIP_PRICE	OTHER_FEES	CUSTOMER_NUM
1600001	40	3/26/2016	2	\$55.00	\$0.00	101
1600002	21	6/8/2016	2	\$95.00	\$0.00	101
1600003	28	9/12/2016	1	\$35.00	\$0.00	103
1600004	26	10/16/2016	4	\$45.00	\$15.00	104
1600005	39	6/25/2016	5	\$55.00	\$0.00	105
1600006	32	6/18/2016	1	\$80.00	\$20.00	106
1600007	22	7/9/2016	8	\$75.00	\$10.00	107
1600008	28	9/12/2016	2	\$35.00	\$0.00	108
1600009	38	9/11/2016	2	\$90.00	\$40.00	109
1600010	2	5/14/2016	3	\$25.00	\$0.00	102
1600011	3	9/15/2016	3	\$25.00	\$0.00	102
1600012	1	6/12/2016	4	\$15.00	\$0.00	115
1600013	8	7/9/2016	1	\$20.00	\$5.00	116
1600014	12	10/1/2016	2	\$40.00	\$5.00	119
1600015	10	7/23/2016	1	\$20.00	\$0.00	120
1600016	11	7/23/2016	6	\$75.00	\$15.00	121
1600017	39	6/18/2016	3	\$20.00	\$5.00	122
1600018	38	9/18/2016	4	\$85.00	\$15.00	126
1600019	25	8/29/2016	2	\$110.00	\$25.00	124
1600020	28	8/27/2016	2	\$35.00	\$10.00	124
1600021	32	6/11/2016	3	\$90.00	\$20.00	112
1600022	21	6/8/2016	1	\$95.00	\$25.00	119
1600024	38	9/11/2016	1	\$70.00	\$30.00	121
1600025	38	9/11/2016	2	\$70.00	\$45.00	125
1600026	12	10/1/2016	2	\$40.00	\$0.00	126
1600029	4	9/19/2016	4	\$105.00	\$25.00	120
1600030	15	7/25/2016	6	\$60.00	\$15.00	104

FIGURE 1-7 Sample reservations data for Colonial Adventure Tours

To check your understanding of the relationship between customers and reservations, answer the following questions.

Q & A

Question: Which customer made reservation 1600010? For which trip date(s) does Karen Busa have reservations?

Answer: Arnold Ocean made reservation 1600010. Find the row in the RESERVATION table with the reservation ID 1600010 (see Figure 1-7), and then find the customer number 102. Next, review the CUSTOMER table (see Figure 1-6), and determine that the customer name with the customer number 102 is Arnold Ocean. Karen Busa has reservations for trips on August 27, 2016 and August 29, 2016. To find the trip dates for Karen Busa, find her customer number (124) in the CUSTOMER table. Next, find all rows in the RESERVATION table that contain the customer number 124.

The table named TRIP_GUIDES shown in Figure 1-8 is used to relate trips and guides. It includes the trip number and the guide number. The trip number in the TRIP_GUIDES table matches a trip number in the TRIP table, and the guide number in the TRIP_GUIDES table matches a guide number in the GUIDE table.

TRIP_GUIDES		TRIP_GUIDES (CONTINUED)	
TRIP_ID	GUIDE_NUM	TRIP_ID	GUIDE_NUM
1	GZ01	19	DH01
1	RH01	20	SL01
2	AM01	21	AM01
2	SL01	22	UG01
3	SL01	23	DH01
4	BR01	23	SL01
4	GZ01	24	BR01
5	KS01	25	BR01
5	UG01	26	GZ01
6	RH01	27	GZ01
7	SL01	28	BR01
8	BR01	29	DH01
9	BR01	30	AM01
10	GZ01	31	SL01
11	DH01	32	KS01
11	KS01	33	UG01
11	UG01	34	KS01
12	BR01	35	GZ01
13	RH01	36	KS02
14	KS02	37	RH01
15	GZ01	38	KS02
16	KS02	39	BR01
17	RH01	40	DH01
18	KS02	41	BR01

FIGURE 1-8 Table used to relate trips and guides

To check your understanding of the relationship between trips and guides, answer the following questions.

Q & A

Question: Which trips are led by Glory Unser? Which guides lead the Lower Pond Tour trip?

Answer: Glory Unser leads the Baldpate Mountain, Chocorua Lake Tour, Long Pond Tour, and Park Loop Ride trips. To determine which trips Glory Unser leads, first examine the GUIDE table (see Figure 1-4) to find her guide number (UG01). Next, look for all rows in the TRIP_GUIDES table that contain her guide number (UG01), and find that these rows contain the trip numbers 5, 11, 22, and 33. Then examine the TRIP table to

determine the trip names for the trips with the trip numbers 5, 11, 22, and 33 to learn the corresponding trip names: Baldpate Mountain, Chocorua Lake Tour, Long Pond Tour, and Park Loop Ride.

Harley Devon and Lori Stevens lead the Lower Pond Tour trips. To find the guides who lead the Long Pond Tour trip, use the TRIP table to identify the trip number for this tour and determine that it has the trip number 23, and then look for all rows in the TRIP_GUIDES table that contain the trip number 23. There are two such rows, which contain the guide numbers DH01 and SL01. Finally, find the rows with these guide numbers in the GUIDE table, and then determine that Harley Devon and Lori Stevens lead the Lower Pond Tour trips.

Q & A

Question: One of Jean Luc Caron's friends called the tour office to reserve the same trip as Jean Luc, but he can't remember the trip name. Which trip did Jean Luc Caron reserve?

Answer: Jean Luc has a reservation for the Sawyer River Ride trip. First, find the customer number for Jean Luc Caron in the CUSTOMER table to determine that his customer number is 109. Then review the RESERVATION table and find all rows with the customer number 109, and determine the trip number for that trip, which is trip number 38. Finally, review the TRIP table to identify that trip number 38 is the Sawyer River Ride trip.

Q & A

Question: Which guides lead paddling trips in the summer season?

Answer: Harley Devon, Susan Kiley, Lori Stevens, and Glory Unser are leading paddling trips in the summer. To identify paddling trips that are offered in the summer, look for rows in the TRIP table that have Paddling in the Type column and Summer in the Season column. There are four such rows with the trip numbers 11, 22, 31, and 32. Locate these trip numbers in the TRIP_GUIDES table, and then determine that the guide numbers DH01, KS01, SL01, and UG01 are associated with these trip numbers. Finally, find the guide numbers DH01, KS01, SL01, and UG01 in the GUIDE table, and then identify the corresponding guides as Harley Devon, Susan Kiley, Lori Stevens, and Glory Unser.

THE SOLMARIS CONDOMINIUM GROUP DATABASE

Solmaris Condominium Group manages two condominium complexes, Solmaris Ocean and Solmaris Bayside, which are located in Florida. In addition to maintaining the common areas, Solmaris also provides a variety of maintenance services for a fee. Solmaris stores

the data it needs to manage its operations in a database containing the tables described in the following section.

In the LOCATION table shown in Figure 1-9, Solmaris stores information about its two locations. Each location is identified by a unique number. The table also includes the location name, address, city, state, and postal code.

LOCATION

LOCATION_NUM	LOCATION_NAME	ADDRESS	CITY	STATE	POSTAL_CODE
1	Solmaris Ocean	100 Ocean Ave.	Bowton	FL	31313
2	Solmaris Bayside	405 Bayside Blvd.	Glander Bay	FL	31044

FIGURE 1-9 Sample location data for Solmaris Condominium Group

Solmaris stores information about the owners of each condo unit in the OWNER table shown in Figure 1-10. Each owner is identified by a unique owner number that consists of two uppercase letters followed by a three-digit number. For each owner, the table also includes the last name, first name, address, city, state, and postal code. You'll notice that only some of the owners actually live in the Solmaris condos that they own.

OWNER

OWNER_NUM	LAST_NAME	FIRST_NAME	ADDRESS	CITY	STATE	POSTAL_CODE
AD057	Adney	Bruce and Jean	100 Ocean Ave.	Bowton	FL	31313
AN175	Anderson	Bill	18 Wilcox St.	Brunswick	GA	31522
BL720	Blake	Jack	2672 Condor St.	Mills	SC	29707
EL025	Elend	Bill and Sandy	100 Ocean Ave.	Bowton	FL	31313
FE182	Feenstra	Daniel	7822 Coventry Dr.	Rivard	FL	31062
JU092	Juarez	Maria	892 Oak St.	Kaleva	FL	31521
KE122	Kelly	Alyssa	527 Waters St.	Norton	MI	49441
NO225	Norton	Peter and Caitlin	281 Lakewood Ave.	Lawndale	PA	19111
RO123	Robinson	Mike and Jane	900 Spring Lake Dr.	Springs	MI	49456
SM072	Smeltz	Jim and Cathy	922 Garland Dr.	Lewiston	FL	32765
TR222	Trent	Michael	405 Bayside Blvd.	Glander Bay	FL	31044
WS032	Wilson	Henry and Karen	25 Nichols St.	Lewiston	FL	32765

FIGURE 1-10 Sample owner data for Solmaris Condominium Group

Each unit at each location is identified by a unit number. Location 1 (Solmaris Ocean) uses a three-digit unit number to identify each condo unit. The first digit identifies the floor and the last two digits indicate the unit on the floor. Location 2 (Solmaris Bayside) uses the letters A, B, and C to identify the three levels of the complex, followed by the unit number on the level.

Information about the condo units at each location is stored in the CONDO_UNIT table shown in Figure 1-11. The table contains the condo ID number, location number, unit number, square footage of the unit, number of bedrooms, number of bathrooms, monthly condo association fee, and the owner number.

CONDO_UNIT

CONDO_ID	LOCATION_NUM	UNIT_NUM	SQR_FT	BDRMS	BATHS	CONDO_FEE	OWNER_NUM
1	1	102	675	1	1	\$475	AD057
2	1	201	1,030	2	1	\$550	EL025
3	1	306	1,575	3	2	\$625	AN175
4	1	204	1,164	2	2	\$575	BL720
5	1	405	1,575	3	2	\$625	FE182
6	1	401	1,030	2	2	\$550	KE122
7	1	502	745	1	1	\$490	JU092
8	1	503	1,680	3	3	\$670	RO123
9	2	A03	725	1	1	\$190	TR222
10	2	A01	1,084	2	1	\$235	NO225
11	2	B01	1,084	2	2	\$250	SM072
12	2	C01	750	1	1	\$190	AN175
13	2	C02	1,245	2	2	\$250	WS032
14	2	C06	1,540	3	2	\$300	RO123

FIGURE 1-11 Sample data about condo units for Solmaris Condominium Group

Solmaris provides maintenance services at its two locations. The types of service provided are stored in the SERVICE_CATEGORY table shown in Figure 1-12. A category number uniquely identifies each service that Solmaris provides. The table also contains a description of the category.

SERVICE_CATEGORY

CATEGORY_NUM	CATEGORY_DESCRIPTION
1	Plumbing
2	Heating/Air Conditioning
3	Painting
4	Electrical Systems
5	Carpentry
6	Janitorial

FIGURE 1-12 Sample data about service categories at Solmaris Condominium Group

Information about the services requested by owners is stored in the SERVICE_REQUEST table shown in Figure 1-13. Each row in the table contains a service ID number that uniquely identifies each service request. The condo ID number identifies which condo requires service. For example, the condo with the ID number 5 on the second row represents the condo in location 1 and unit number 405 (as found by examining row 5 in the CONDO_UNIT table). The SERVICE_REQUEST table also stores the category number of the service to be performed, a description of the specific service to be performed, the current status of the service, and the estimated number of hours required to complete the service. For completed jobs, the table contains the actual number of hours required to complete the service. If another appointment is required to complete additional service, the appointment date appears in the NEXT_SERVICE_DATE column.

SERVICE_REQUEST

SERVICE_ID	CONDO_ID	CATEGORY_NUM	DESCRIPTION	STATUS	EST_HOURS	SPENT_HOURS	NEXT_SERVICE_DATE
1	2	1	Back wall in pantry has mold indicating water seepage. Diagnose and repair.	Service rep has verified the problem. Plumbing contractor has been called.	4	2	10/12/2015
2	5	2	Air conditioning doesn't cool.	Service rep has verified problem. Air conditioning contractor has been called.	3	1	10/12/2015
3	4	6	Hardwood floors must be refinished.	Service call has been scheduled.	8	0	10/16/2015
4	1	4	Switches in kitchen and adjoining dining room are reversed.	Open	1	0	10/13/2015
5	2	5	Molding in pantry must be replaced.	Cannot schedule until water leak is corrected.	2	0	
6	14	3	Unit needs to be repainted due to previous tenant damage.	Scheduled	7	0	10/19/2015
7	11	4	Tenant complained that using microwave caused short circuits on two occasions.	Service rep unable to duplicate problem. Tenant to notify condo management if problem recurs.	1	1	
8	9	3	Kitchen must be repainted. Walls discolored due to kitchen fire.	Scheduled	5	0	10/16/2015
9	7	6	Shampoo all carpets.	Open	5	0	10/19/2015
10	9	5	Repair window sills.	Scheduled	4	0	10/20/2015

FIGURE 1-13 Sample data about service requests for Solmaris Condominium Group

The Solmaris Condominium Group exercises at the end of this chapter will give you a chance to check your understanding of the data in this database.

Chapter Summary

- TAL Distributors is an organization whose information requirements include sales reps, customers, items, orders, and order lines.
- Colonial Adventure Tours is a company whose requirements include information about the following entities: guides, trips, customers, reservations, and trip guides.
- Solmaris Condominium Group is a company whose requirements include information about the following entities: condo locations, owners, condo units, service categories, and service requests.

Key Term

database

Exercises

TAL Distributors

Answer each of the following questions using the TAL Distributors data shown in Figure 1-2. No computer work is required.

1. List the names of all customers that have a credit limit of \$7,500 or less.
2. List the order numbers for orders placed by customer number 586 on 10/15/2015.
3. List the item number, item description, and on-hand value for each item in category TOY. (*Hint: On-hand value is the result of multiplying the number of units on hand by the price.*)
4. List the item number and item description of all items that are in category PZL.
5. How many customers have a balance that exceeds their credit limit?
6. What is the item number, description, and price of the least expensive item in the database?
7. For each order, list the order number, order date, customer number, and customer name.
8. For each order placed on October 13, 2015, list the order number, customer number, and customer name.
9. List the sales rep number and name for every sales rep who represents at least one customer with a credit limit of \$10,000.
10. For each order placed on October 12, 2015, list the order number, item number, item description, and category for each item ordered.
11. TAL Distributors needs to be able to contact customers when problems arise concerning an order. What other types of data could TAL include in the CUSTOMER table to assist in contacting customers?



Colonial Adventure Tours

Answer each of the following questions using the Colonial Adventure Tours data shown in Figures 1-4 through 1-8. No computer work is required.

1. List the first and last name of each guide that lives in Maine (ME).
2. List the trip name of each trip that has a maximum group size of greater than 16.
3. List the trip ID and trip name of each trip that has the type Paddling.
4. List the trip ID and trip name of each trip that has the type Biking and that has the season Early Fall.
5. List the trip ID and trip name of each trip that is in the state of Vermont (VT) or that has the type Paddling.
6. How many trips have the type Hiking?
7. List the trip name of each trip that has Hal Rowan as a guide.
8. Find the name of each trip containing the word *Loop*.
9. List the first and last name of each guide who lives in Maine (ME), Vermont (VT), or Connecticut (CT).
10. How many reservations are for trips that occur during July 2016?
11. For each customer that has more than one reservation, list the customer's first and last names, the trip name, and the trip type.
12. List the last name of each customer that has a reservation for a trip in Massachusetts (MA).
13. How many reservations have a trip price that is greater than \$15.00 and less than \$40.00?
14. Currently, trip cost is determined by the number of people taking the trip and the type of trip. If trip cost were based only on the type of trip, in which table would you place the trip cost field? Why?



Solmaris Condominium Group

Answer each of the following questions using the Solmaris Condominium Group data shown in Figures 1-9 through 1-13. No computer work is required.

1. List the owner number, last name, and first name of every condominium owner.
2. List the last name and first name of every owner located in Bowton.
3. List the condo ID number for each condo that is smaller than 1,084 square feet.
4. List the last name, first name, and city of every owner who owns more than one condo.
5. List the last name, first name, and city of every owner with a condo that has a condo fee of less than \$250 per month.
6. List the unit number of each condo at the Solmaris Ocean location.
7. How many condos have two bedrooms and two bathrooms?
8. How many owners live in Florida (FL), Georgia (GA), or South Carolina (SC)?

9. List the owner's last and first names and the unit number for each condo that has a scheduled or open service request.
10. List the condo ID number and square footage for each condo that has a janitorial service request.
11. List the condo ID number and location name for all service requests for which the estimated number of hours is greater than 5.
12. What is the average condo fee for all three-bedroom condos?
13. The Solmaris database does not include a column for service fees. In which table would you place the information for service fees? Why?



CHAPTER 2

DATABASE DESIGN FUNDAMENTALS

OBJECTIVES

- Understand the terms *entity*, *attribute*, and *relationship*
- Understand the terms *relation* and *relational database*
- Understand functional dependence and identify when one column is functionally dependent on another
- Understand the term *primary key* and identify primary keys in tables
- Design a database to satisfy a set of requirements
- Convert an unnormalized relation to first normal form
- Convert tables from first normal form to second normal form
- Convert tables from second normal form to third normal form
- Create an entity-relationship diagram to represent the design of a database

INTRODUCTION

In Chapter 1, you reviewed the tables and columns in the TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group databases that you will use to complete the rest of this text. The process of determining the particular tables and columns that will comprise a database is known as **database design**. In this chapter, you will learn a method for designing a database to satisfy a set of requirements. In the process, you will learn how to identify the tables and columns in the database. You also will learn how to identify the relationships between the tables.

This chapter begins by examining some important concepts related to databases. It also presents the design method using the set of requirements that TAL Distributors identified to produce the appropriate database design. The chapter then examines the process of normalization, in which you identify and fix potential problems in database designs. Finally, you will learn a way of visually representing the design of a database.

DATABASE CONCEPTS

Before learning how to design a database, you need to be familiar with some important database concepts related to relational databases, which are the types of databases you examined in Chapter 1 and that you will use throughout the rest of this text. The terms entity, attribute, and relationship are important to understand when designing a database; the concepts of functional dependence and primary keys are critical when learning about the database design process.

Relational Databases

A **relational database** is a collection of tables like the ones you examined for TAL Distributors in Chapter 1 and that appear in Figure 2-1. Formally, these tables are called relations, and this is how this type of database gets its name.

REP								
REP_NUM	LAST_NAME	FIRST_NAME	STREET	CITY	STATE	POSTAL_CODE	COMMISSION	RATE
15	Campos	Rafael	724 Vinca Dr.	Grove	CA	90092	\$23,457.50	0.06
30	Gradey	Megan	632 Liatris St.	Fullton	CA	90085	\$41,317.00	0.08
45	Tian	Hui	1785 Tyler Ave.	Northfield	CA	90098	\$27,789.25	0.06
60	Sefton	Janet	267 Oakley St.	Congaree	CA	90097	\$0.00	0.06

CUSTOMER								
CUSTOMER_NUM	CUSTOMER_NAME	STREET	CITY	STATE	POSTAL_CODE	BALANCE	CREDIT_LIMIT	REP_NUM
126	Toys Galore	28 Laketon St.	Fullton	CA	90085	\$1,210.25	\$7,500.00	15
260	Brookings Direct	452 Columbus Dr.	Grove	CA	90092	\$575.00	\$10,000.00	30
334	The Everything Shop	342 Magee St.	Congaree	CA	90097	\$2,345.75	\$7,500.00	45
386	Johnson's Department Store	124 Main St.	Northfield	CA	90098	\$879.25	\$7,500.00	30
440	Grove Historical Museum Store	3456 Central Ave.	Fullton	CA	90085	\$345.00	\$5,000.00	45
502	Cards and More	167 Hale St.	Mesa	CA	90104	\$5,025.75	\$5,000.00	15
586	Almondton General Store	3345 Devon Ave.	Almondton	CA	90125	\$3,456.75	\$15,000.00	45
665	Cricket Gift Shop	372 Oxford St.	Grove	CA	90092	\$678.90	\$7,500.00	30
713	Cress Store	12 Rising Sun Ave.	Congaree	CA	90097	\$4,234.60	\$10,000.00	15
796	Unique Gifts	786 Passmore St.	Northfield	CA	90098	\$124.75	\$7,500.00	45
824	Kline's	945 Gilham St.	Mesa	CA	90104	\$2,475.99	\$15,000.00	30
893	All Season Gifts	382 Wildwood Ave.	Fullton	CA	90085	\$935.75	\$7,500.00	15

FIGURE 2-1 Sample data for TAL Distributors

ORDERS

ORDER_NUM	ORDER_DATE	CUSTOMER_NUM
51608	10/12/2015	126
51610	10/12/2015	334
51613	10/13/2015	386
51614	10/13/2015	260
51617	10/15/2015	586
51619	10/15/2015	126
51623	10/15/2015	586
51625	10/16/2015	796

ORDER_LINE

ORDER_NUM	ITEM_NUM	NUM_ORDERED	QUOTED_PRICE
51608	CD33	5	\$86.99
51610	KL78	25	\$10.95
51610	TR40	10	\$13.99
51613	DL51	5	\$104.95
51614	FD11	1	\$124.95
51617	NL89	4	\$115.99
51617	TW35	3	\$116.95
51619	FD11	2	\$121.95
51623	DR67	5	\$29.95
51623	FH24	12	\$36.95
51623	KD34	10	\$13.10
51625	MT03	8	\$45.79

ITEM

ITEM_NUM	DESCRIPTION	ON_HAND	CATEGORY	STOREHOUSE	PRICE
AH74	Patience	9	GME	3	\$22.99
BR23	Skittles	21	GME	2	\$29.99
CD33	Wood Block Set (48 piece)	36	TOY	1	\$89.49
DL51	Classic Railway Set	12	TOY	3	\$107.95
DR67	Giant Star Brain Teaser	24	PZL	2	\$31.95
DW23	Mancala	40	GME	3	\$50.00
FD11	Rocking Horse	8	TOY	3	\$124.95
FH24	Puzzle Gift Set	65	PZL	1	\$38.95
KA12	Cribbage Set	56	GME	3	\$75.00
KD34	Pentominoes Brain Teaser	60	PZL	2	\$14.95
KL78	Pick Up Sticks	110	GME	1	\$10.95
MT03	Zauberkasten Brain Teaser	45	PZL	1	\$45.79
NL89	Wood Block Set (62 piece)	32	TOY	3	\$119.75
TR40	Tic Tac Toe	75	GME	2	\$13.99
TW35	Fire Engine	30	TOY	2	\$118.95

FIGURE 2-1 Sample data for TAL Distributors (*continued*)**NOTE**

The names of columns and tables in this text follow a common naming convention in which column names use uppercase letters and replace spaces between words with underscores (_). For example, TAL Distributors uses the column named LAST_NAME to store last names and the column named CREDIT_LIMIT to store credit limits.

Entities, Attributes, and Relationships

There are some terms and concepts that are very important for you to know when working in the database environment. The terms *entity*, *attribute*, and *relationship* are fundamental when discussing databases. An **entity** is like a noun; it is a person, place, thing, or event. The entities of interest to TAL Distributors, for example, are such things as customers, orders, and sales reps. The entities that are of interest to a school include students, faculty, and classes; a real estate agency is interested in clients, houses, and agents; and a used car dealer is interested in vehicles, customers, and manufacturers.

An **attribute** is a property of an entity. The term is used here exactly as it is used in everyday English. For the entity *person*, for example, the list of attributes might include such things as eye color and height. For TAL Distributors, the attributes of interest for the entity *customer* are such things as name, address, city, and so on. For the entity *faculty* at a school, the attributes would be such things as faculty number, name, office number, phone, and so on. For the entity *vehicle* at a car dealership, the attributes are such things as the vehicle identification number, model, color, year, and so on.

A **relationship** is the association between entities. There is an association between customers and sales reps, for example, at TAL Distributors. A sales rep is associated with all of his or her customers, and a customer is associated with his or her sales rep. Technically, you say that a sales rep is *related* to all of his or her customers, and a customer is *related* to his or her sales rep.

The relationship between sales reps and customers is an example of a **one-to-many relationship** because one sales rep is associated with many customers, but each customer is associated with only one sales rep. (In this type of relationship, the word *many* is used in a way that is different from everyday English; it might not always mean a large number. In this context, for example, the term *many* means that a sales rep might be associated with *any* number of customers. That is, one sales rep can be associated with zero, one, or more customers.)

How does a relational database handle entities, attributes of entities, and relationships between entities? Entities and attributes are fairly simple. Each entity has its own table. In the TAL Distributors database, there is one table for sales reps, one table for customers, and so on. The attributes of an entity become the columns in the table. In the table for sales reps, for example, there is a column for the sales rep number, a column for the sales rep's first name, and so on.

What about relationships? At TAL Distributors, there is a one-to-many relationship between sales reps and customers (each sales rep is related to the *many* customers that he or she represents, and each customer is related to the *one* sales rep who represents the customer). How is this relationship implemented in a relational database?

Consider Figure 2-1 again. If you want to determine the name of the sales rep who represents Brookings Direct (customer number 260), you would locate the row for Brookings Direct in the CUSTOMER table and determine that the value for REP_NUM is 30. Then you would look for the row in the REP table on which the REP_NUM is 30. The *one* rep with REP_NUM 30 is Megan Gradey, who represents Brookings Direct.

On the other hand, if you want to determine the names of all the customers of the rep named Rafael Campos, you would locate the row for Rafael Campos in the REP table and determine that the value in the REP_NUM column is 15. Then you would look for all the rows in the CUSTOMER table on which the REP_NUM is 15. After identifying Rafael Campos's rep number, you find that the *many* customers he represents are

numbered 126 (Toys Galore), 502 (Cards and More), 713 (Cress Store), and 893 (All Season Gifts).

You implement these relationships by having common columns in two or more tables. The REP_NUM column in the REP table and the REP_NUM column in the CUSTOMER table are used to implement the relationship between sales reps and customers. Given a sales rep, you can use these columns to determine all the customers that he or she represents; given a customer, you can use these columns to find the sales rep who represents the customer.

In this context, a relation is essentially a two-dimensional table. If you consider the tables shown in Figure 2-1, however, you can see that certain restrictions are placed on relations. Each column has a unique name, and entries within each column should *match* this column name. For example, if the column name is CREDIT_LIMIT, all entries in that column must be credit limits. In addition, each row should be unique—when two rows are identical, the second row does not provide any new information. For maximum flexibility, the order of the columns and rows should be immaterial. Finally, the table's design should be as simple as possible by restricting each position to a single entry and by preventing multiple entries (also called **repeating groups**) in an individual location in the table. Figure 2-2 shows a table design that includes repeating groups.

ORDERS

ORDER_NUM	ORDER_DATE	CUSTOMER_NUM	ITEM_NUM	NUM_ORDERED	QUOTED_PRICE
51608	10/12/2015	126	CD33	5	\$86.99
51610	10/12/2015	334	KL78	25	\$10.95
			TR40	10	\$13.99
51613	10/13/2015	386	DL51	5	\$104.95
51614	10/13/2015	260	FD11	1	\$124.95
51617	10/15/2015	586	NL89	4	\$115.99
			TW35	3	\$116.95
51619	10/15/2015	126	FD11	2	\$121.95
51623	10/15/2015	586	DR67	5	\$29.95
			FH24	12	\$36.95
			KD34	10	\$13.10
51625	10/16/2015	796	MT03	8	\$45.79

FIGURE 2-2 Table with repeating groups

Figure 2-3 shows a better way to represent the same information shown in Figure 2-2. In Figure 2-3, every position in the table contains a single value.

ORDERS

ORDER_NUM	ORDER_DATE	CUSTOMER_NUM	ITEM_NUM	NUM_ORDERED	QUOTED_PRICE
51608	10/12/2015	126	CD33	5	\$86.99
51610	10/12/2015	334	KL78	25	\$10.95
51610	10/12/2015	334	TR40	10	\$13.99
51613	10/13/2015	386	DL51	5	\$104.95
51614	10/13/2015	260	FD11	1	\$124.95
51617	10/15/2015	586	NL89	4	\$115.99
51617	10/15/2015	586	TW35	3	\$116.95
51619	10/15/2015	126	FD11	2	\$121.95
51623	10/15/2015	586	DR67	5	\$29.95
51623	10/15/2015	586	FH24	12	\$36.95
51623	10/15/2015	586	KD34	10	\$13.10
51625	10/16/2015	796	MT03	8	\$45.79

FIGURE 2-3 ORDERS table without repeating groups

When you remove the repeating groups from Figure 2-2, all of the rows in Figure 2-3 are single-valued. This structure is formally called a **relation**. A **relation** is a two-dimensional table in which the entries in the table are single-valued (each location in the table contains a single entry), each column has a distinct name, all values in the column match this name, the order of the rows and columns is immaterial, and each row contains unique values. A relational database is a collection of relations.

NOTE

Rows in a table (relation) are also called **records** or **tuples**. Columns in a table (relation) are also called **fields** or attributes. This text uses the terms tables, columns, and rows unless the more formal terms of relation, attributes, and tuples are necessary for clarity.

There is a commonly accepted shorthand representation to show the tables and columns in a relational database: for each table, you write the name of the table and then within parentheses list all of the columns in the table. In this representation, each table

appears on its own line. Using this method, you represent the TAL Distributors database as follows:

```
REP (REP_NUM, LAST_NAME, FIRST_NAME, STREET, CITY, STATE,
    POSTAL_CODE, COMMISSION, RATE)
CUSTOMER (CUSTOMER_NUM, CUSTOMER_NAME, STREET, CITY, STATE,
    POSTAL_CODE, BALANCE, CREDIT_LIMIT, REP_NUM)
ORDERS (ORDER_NUM, ORDER_DATE, CUSTOMER_NUM)
ORDER_LINE (ORDER_NUM, ITEM_NUM, NUM_ORDERED, QUOTED_PRICE)
ITEM (ITEM_NUM, DESCRIPTION, ON_HAND, CATEGORY, STOREHOUSE, PRICE)
```

Notice that some tables contain columns with duplicate names. For example, the REP_NUM column appears in both the REP table *and* the CUSTOMER table. Suppose a situation existed wherein someone (or the DBMS) might confuse the two columns. For example, if you write REP_NUM, it is not clear which REP_NUM column you want to use. You need a mechanism for indicating the REP_NUM column to which you are referring. One common approach to solving this problem is to write both the table name and the column name, separated by a period. Thus, you would reference the REP_NUM column in the CUSTOMER table as CUSTOMER.REP_NUM, and the REP_NUM column in the REP table as REP.REP_NUM. Technically, when you reference columns in this format; you say that you **qualify** the names. It is *always* acceptable to qualify column names, even when there is no potential for confusion. If confusion might arise, however, it is *essential* to qualify column names.

FUNCTIONAL DEPENDENCE

The concept of functional dependence is crucial to understanding the rest of the material in this chapter. Functional dependence is a formal name for what is basically a simple idea. To illustrate functional dependence, suppose that the REP table for TAL Distributors is structured as shown in Figure 2-4. The only difference between the REP table shown in Figure 2-4 and the one shown in Figure 2-1 is the addition of an extra column named PAY_CLASS.

REP

REP_NUM	LAST_NAME	FIRST_NAME	STREET	CITY	STATE	POSTAL_CODE	COMMISSION	PAY_CLASS	RATE
15	Campos	Rafael	724 Vinca Dr.	Grove	CA	90092	\$23,457.50	1	0.06
30	Gradey	Megan	632 Liatris St.	Fullton	CA	90085	\$41,317.00	2	0.08
45	Tian	Hui	1785 Tyler Ave.	Northfield	CA	90098	\$27,789.25	1	0.06
60	Sefton	Janet	267 Oakley St.	Congaree	CA	90097	\$0.00	1	0.06

FIGURE 2-4 REP table with a PAY_CLASS column

Suppose that one of the policies at TAL Distributors is that all sales reps in any given pay class earn their commissions at the same rate. To describe this situation, you could say that a sales rep's pay class *determines* his or her commission rate. Alternatively, you could say that a sales rep's commission rate *depends on* his or her pay class. This phrasing uses the words *determines* and *depends on* in the same way that you describe functional dependency. If you wanted to be formal, you would precede either expression with the word *functionally*. For example, you might say, "A sales rep's pay class *functionally determines* his or her commission rate," and "A sales rep's commission rate *functionally depends on* his or her pay class." You can also define functional dependency by saying that when you know a sales rep's pay class, you can determine his or her commission rate.

In a relational database, column B is **functionally dependent** on another column (or a collection of columns), A, if at any point in time a value for A determines a single value for B. You can think of this as follows: when you are given a value for A, do you know that you can find a single value for B? If so, B is functionally dependent on A (often written as $A \rightarrow B$). If B is functionally dependent on A, you also can say that A **functionally determines** B.

At TAL Distributors, is the LAST_NAME column in the REP table functionally dependent on the REP_NUM column? Yes, it is. If you are given a value for REP_NUM, such as 30, there is a *single* LAST_NAME, Gradey, associated with it. This is represented as:
 $REP_NUM \rightarrow LAST_NAME$

Q & A

Question: In the CUSTOMER table, is CUSTOMER_NAME functionally dependent on REP_NUM?

Answer: No. Given the REP_NUM 30, for example, you would not be able to find a single customer name, because 30 appears on more than one row in the table.

Q & A

Question: In the ORDER_LINE table, is NUM_ORDERED functionally dependent on ORDER_NUM?

Answer: No. An ORDER_NUM might be associated with several items in an order, so having just an ORDER_NUM does not provide enough information.

Q & A

Question: Is NUM_ORDERED functionally dependent on ITEM_NUM?

Answer: No. Again, just as with ORDER_NUM, an item number might be associated with several lines in an order, so ITEM_NUM does not provide enough information.

Q & A

Question: On which columns in the ORDER_LINE table is NUM_ORDERED functionally dependent?

Answer: To determine a value for NUM_ORDERED, you need both an order number and an item number. In other words, NUM_ORDERED is functionally dependent on the combination (formally called the **concatenation**) of ORDER_NUM and ITEM_NUM. That is, given an order number *and* an item number, you can find a single value for NUM_ORDERED.

At this point, a question naturally arises: how do you determine functional dependencies? Can you determine them by looking at sample data, for example? The answer is no.

Consider the REP table in Figure 2-5, in which last names are unique. It is very tempting to say that LAST_NAME functionally determines STREET, CITY, STATE, and POSTAL_CODE (or equivalently that STREET, CITY, STATE, and POSTAL_CODE are all functionally dependent on LAST_NAME). After all, given the last name of a rep, you can find the single address.

REP

REP_NUM	LAST_NAME	FIRST_NAME	STREET	CITY	STATE	POSTAL_CODE	COMMISSION	RATE
15	Campos	Rafael	724 Vinca Dr.	Grove	CA	90092	\$23,457.50	0.06
30	Gradey	Megan	632 Liatris St.	Fullton	CA	90085	\$41,317.00	0.08
45	Tian	Hui	1785 Tyler Ave.	Northfield	CA	90098	\$27,789.25	0.06
60	Sefton	Janet	267 Oakley St.	Congaree	CA	90097	\$0.00	0.06

FIGURE 2-5 REP table

What would happen if the last name of rep 60 were also Campos? You would have the situation illustrated in Figure 2-6. Because there are now two reps with the last name of Campos, you can no longer find a single address using a rep's last name—you were misled by the original data. The only way to determine functional dependencies is to examine the user's policies. This process can involve discussions with users, an examination of user documentation, and so on. For example, if managers at TAL Distributors have a policy not to hire two reps with the same last name, then LAST_NAME would indeed determine the other columns. Without such a policy, however, LAST_NAME would not determine the other columns.

REP

REP_NUM	LAST_NAME	FIRST_NAME	STREET	CITY	STATE	POSTAL_CODE	COMMISSION	RATE
15	Campos	Rafael	724 Vinca Dr.	Grove	CA	90092	\$23,457.50	0.06
30	Gradey	Megan	632 Liatris St.	Fullton	CA	90085	\$41,317.00	0.08
45	Tian	Hui	1785 Tyler Ave.	Northfield	CA	90098	\$27,789.25	0.06
60	Campos	Janet	267 Oakley St.	Congaree	CA	90097	\$0.00	0.06

FIGURE 2-6 REP table with two reps named Campos

PRIMARY KEYS

Another important database design concept is the primary key. In the simplest terms, the **primary key** is the unique identifier for a table. For example, the REP_NUM column is the unique identifier for the REP table. Given a rep number in the table, such as 30, there will only be one row on which that rep number occurs. Thus, the rep number 30 uniquely identifies a row (in this case, the second row).

In this text, the definition of primary key needs to be more precise than just a unique identifier for a table. Specifically, column A (or a collection of columns) is the primary key for a table if:

Property 1. *All* columns in the table are functionally dependent on A.

Property 2. No subcollection of the columns in A (assuming A is a collection of columns and not just a single column) also has property 1.

Q & A

Question: Is the CATEGORY column the primary key for the ITEM table?

Answer: No, because the other columns are not functionally dependent on CATEGORY. Given the category GME, for example, you cannot determine an item number, description, or anything else, because there are several rows on which the category is GME.

Q & A

Question: Is the CUSTOMER_NUM column the primary key for the CUSTOMER table?

Answer: Yes, because TAL Distributors assigns unique customer numbers. A specific customer number cannot appear on more than one row. Thus, all columns in the CUSTOMER table are functionally dependent on CUSTOMER_NUM.

Q & A

Question: Is the ORDER_NUM column the primary key for the ORDER_LINE table?

Answer: No, because it does not functionally determine either NUM_ORDERED or QUOTED_PRICE.

Q & A

Question: Is the combination of the ORDER_NUM and ITEM_NUM columns the primary key for the ORDER_LINE table?

Answer: Yes, because you can determine all columns by this combination of columns, and, further, neither the ORDER_NUM nor the ITEM_NUM alone has this property.

Q & A

Question: Is the combination of the ITEM_NUM and DESCRIPTION columns the primary key for the ITEM table?

Answer: No. Although it is true that you can determine all columns in the ITEM table by this combination, ITEM_NUM alone also has this property.

You can indicate a table's primary key with a shorthand representation of a database by underlining the column or collection of columns that comprise the primary key. The complete shorthand representation for the TAL Distributors database is:

```
REP (REP_NUM, LAST_NAME, FIRST_NAME, STREET, CITY, STATE,
      POSTAL_CODE, COMMISSION, RATE)
CUSTOMER (CUSTOMER_NUM, CUSTOMER_NAME, STREET, CITY, STATE,
          POSTAL_CODE, BALANCE, CREDIT_LIMIT, REP_NUM)
ORDERS (ORDER_NUM, ORDER_DATE, CUSTOMER_NUM)
ORDER_LINE (ORDER_NUM, ITEM_NUM, NUM_ORDERED, QUOTED_PRICE)
ITEM (ITEM_NUM, DESCRIPTION, ON_HAND, CATEGORY, STOREHOUSE, PRICE)
```

NOTE

Sometimes you might identify one or more columns that you can use as a table's primary key. For example, if the TAL Distributors database also included an EMPLOYEE table that contains employee numbers and Social Security numbers, either the employee number or the Social Security number could serve as the table's primary key. In this case, both columns are referred to as candidate keys. Like a primary key, a **candidate key** is a column or collection of columns on which all columns in the table are functionally dependent—the definition for primary key really defines candidate key as well. From all the candidate keys, you would choose one to be the primary key.