

CONCEPTS OF
**DATABASE
MANAGEMENT**

FRIEDRICHSEN • RUFFOLO • MONK
STARKS • PRATT • LAST



10TH EDITION

CONCEPTS OF DATABASE MANAGEMENT

CONCEPTS OF DATABASE MANAGEMENT

Tenth Edition

**Lisa Friedrichsen | Lisa Ruffolo | Ellen F. Monk
Joy L. Starks | Philip J. Pratt | Mary Z. Last**



Australia • Brazil • Mexico • Singapore • United Kingdom • United States

This is an electronic version of the print textbook. Due to electronic rights restrictions, some third party content may be suppressed. Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. The publisher reserves the right to remove content from this title at any time if subsequent rights restrictions require it. For valuable information on pricing, previous editions, changes to current editions, and alternate formats, please visit www.cengage.com/highered to search by ISBN#, author, title, or keyword for materials in your areas of interest.

Important Notice: Media content referenced within the product description or the product text may not be available in the eBook version.

Concepts of Database Management,
Tenth Edition
Lisa Friedrichsen, Lisa Ruffolo,
Ellen F. Monk, Joy L. Starks, Philip J. Pratt,
Mary Z. Last

SVP, Higher Education Product Management:

Erin Joyner

VP, Product Management: Mike Schenk

Product Director: Lauren Murphy

Product Team Manager: Kristin McNary

Product Manager: Jaymie Falconi

Product Assistant: Thomas C. Benedetto

Director, Learning Design: Rebecca von Gillern

Senior Manager, Learning Design: Leigh Hefferon

Learning Designer: Emily Pope

Vice President, Marketing – Science, Technology,
& Math: Jason Sakos

Senior Marketing Director: Michele McTighe

Marketing Manager: Cassie Cloutier

Product Specialist: Mackenzie Paine

Director, Content Creation: Juliet Steiner

Senior Manager, Content Creation: Patty Stephan

Senior Content Manager: Anne Orgren

Director, Digital Production Services:

Krista Kellman

Digital Delivery Lead: Dan Swanson

Developmental Editor: Lisa Ruffolo

Production Service/Composition:

Lumina Datamatics, Inc.

Design Director: Jack Pendleton

Designer: Lizz Anderson

Cover Designer: Lizz Anderson

Cover image: iStockPhoto.com/Viorika

© 2021, 2019 Cengage Learning, Inc.

Unless otherwise noted, all content is © Cengage.

WCN: 02-300

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced or distributed in any form or by any means, except as permitted by U.S. copyright law, without the prior written permission of the copyright owner.

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.

Screenshots for this book were created using Microsoft Access®.

Microsoft® and all Microsoft-based trademarks and logos are registered trademarks of Microsoft Corporation, Inc. in the United States and other countries. Cengage is an independent entity from the Microsoft Corporation, and not affiliated with Microsoft in any manner.

Oracle® and all Oracle-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation, Inc. in the United States and other countries.

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.

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

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

Library of Congress Control Number: 2019958007

Student Edition ISBN: 978-0-357-42208-3

Loose-leaf Edition ISBN: 978-0-357-42209-0

Cengage

200 Pier 4 Boulevard

Boston, MA 02210

USA

Cengage is a leading provider of customized learning solutions with employees residing in nearly 40 different countries and sales in more than 125 countries around the world. Find your local representative at **www.cengage.com.**

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

To learn more about Cengage platforms and services, register or access your online learning solution, or purchase materials for your course, visit

www.cengage.com.

Notice to the Reader

Publisher does not warrant or guarantee any of the products described herein or perform any independent analysis in connection with any of the product information contained herein. Publisher does not assume, and expressly disclaims, any obligation to obtain and include information other than that provided to it by the manufacturer. The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instructions contained herein, the reader willingly assumes all risks in connection with such instructions. The publisher makes no representations or warranties of any kind, including but not limited to, the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and the publisher takes no responsibility with respect to such material. The publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or part, from the readers' use of, or reliance upon, this material.

Printed in the United States of America

Print Number: 01 Print Year: 2020

BRIEF CONTENTS

Preface	xv
Module 1	
<i>Introduction to Database Management</i>	1
Module 2	
<i>The Relational Model: Introduction, QBE, and Relational Algebra</i>	31
Module 3	
<i>The Relational Model: SQL</i>	83
Module 4	
<i>The Relational Model: Advanced Topics</i>	131
Module 5	
<i>Database Design: Normalization</i>	167
Module 6	
<i>Database Design: Relationships</i>	207
Module 7	
<i>Database Management Systems Processes and Services</i>	241
Module 8	
<i>Database Industry Careers</i>	261
Module 9	
<i>Database Industry Trends</i>	281
Appendix A	
<i>Comprehensive Design Example: Douglas College</i>	319
Appendix B	
<i>SQL Reference</i>	349
Appendix C	
<i>FAQ Reference</i>	359
Appendix D	
<i>Introduction to MySQL</i>	361
Appendix E	
<i>A Systems Analysis Approach to Information-Level Requirements</i>	371
Glossary	377
Index	391

CONTENTS

Preface	xv
Module 1	
<i>Introduction to Database Management</i>	1
Introduction	1
JC Consulting Company Background	1
Selecting a Database Solution	4
Defining Database Terminology	4
Storing Data	5
Identifying Database Management Systems	11
Advantages of a Properly Designed Relational Database	14
Key Factors for a Healthy Relational Database	15
Big Data	16
Preparing for a Career in Database Administration and Data Analysis	16
Introduction to the Pitt Fitness Database Case	16
Introduction to the Sports Physical Therapy Database Case	20
Summary	24
Key Terms	24
Module Review Questions	25
Problems	25
Critical Thinking Questions	26
JC Consulting Case Exercises	26
Problems	26
Critical Thinking Questions	27
Pitt Fitness Case Exercises	27
Problems	27
Critical Thinking Questions	29
Sports Physical Therapy Case Exercises	29
Problems	29
Critical Thinking Questions	30
Module 2	
<i>The Relational Model: Introduction, QBE, and Relational Algebra</i>	31
Introduction	31
Examining Relational Databases	31
Relational Database Shorthand	35
Creating Simple Queries and Using Query-By-Example	36
Selecting Fields and Running the Query	37
Saving and Using Queries	38
Using Simple Criteria	40
Parameter Queries	41
Comparison Operators	42
Using Compound Criteria	42
Creating Computed Fields	46
Summarizing with Aggregate Functions and Grouping	49
Sorting Records	52
Sorting on Multiple Keys	54
Joining Tables	56
Joining Multiple Tables	60
Using an Update Query	62
Using a Delete Query	63
Using a Make-Table Query	64

Optimizing Queries	65
Examining Relational Algebra	65
Selection	66
Projection	66
Joining	67
Union	69
Intersection	70
Difference	70
Product	71
Division	71
Summary	72
Key Terms	73
Module Review Questions	74
Problems	74
Critical Thinking Questions	75
JC Consulting Case Exercises: QBE	76
Problems	76
Critical Thinking Questions	77
JC Consulting Case Exercises: Relational Algebra	77
Problems	77
Pitt Fitness Case Exercises	79
Problems	79
Critical Thinking Questions	80
Sports Physical Therapy Case Exercises	81
Problems	81
Critical Thinking Questions	82
Module 3	
<i>The Relational Model: SQL</i>	83
Introduction	83
Getting Started with SQL	84
Opening an SQL Query Window in Access	84
Changing the Font and Font Size in SQL View	85
Creating a Table	85
Naming Conventions	85
Data Types	86
Selecting Data	88
Numeric Criteria	90
Text Criteria	91
Date Criteria	93
Comparing Two Fields	94
Saving SQL Queries	94
Using Compound Conditions: AND Criteria	95
Using Compound Conditions: OR Criteria	95
Using the BETWEEN Operator	96
Using the NOT Operator	98
Creating Calculated Fields	99
Using Wildcards and the LIKE Operator	101
Using the IN Operator	102
Sorting Records	102
Sorting on Multiple Fields	103
Using Aggregate Functions	104
Grouping Records	105
Limiting Records with the HAVING clause	107
Writing Subqueries	108
Joining Tables with the WHERE Clause	109
Joining More Than Two Tables with the WHERE Clause	111
Using the UNION Operator	112

Updating Values with the SQL UPDATE Command	113
Inserting a Record with the SQL INSERT Command	114
Deleting Records with the SQL DELETE Command	116
Saving Query Results as a Table	117
Developing Career Skills: SQL	118
Accessing Free SQL Tutorials	118
Summary	119
Key Terms	120
Module Review Questions	120
Problems	120
Critical Thinking Question	122
JC Consulting Case Exercises	122
Problems	122
Critical Thinking Questions	124
Pitt Fitness Case Exercises	124
Problems	124
Critical Thinking Questions	126
Sports Physical Therapy Case Exercises	126
Problems	126
Critical Thinking Questions	129
Module 4	
<i>The Relational Model: Advanced Topics</i>	131
Introduction	131
Creating and Using Views	131
Using Indexes	134
Examining Database Security Features	137
Preventing Unauthorized Access	138
Safely Distributing Information	139
Providing Physical Security	139
Enforcing Integrity Rules	139
Entity Integrity	139
Referential Integrity	140
Cascade Options	142
Legal-Values Integrity	143
Changing the Structure of a Relational Database	144
Adding a New Field to a Table	144
Modifying Field Properties	145
Deleting a Field	146
Deleting a Table	146
Using SQL JOIN Commands	147
LEFT Joins	149
RIGHT Joins	150
Applying Referential Integrity: Error Messages	152
Applying Referential Integrity: Null Values	153
Using the System Catalog	153
Using Stored Procedures and Triggers	154
Triggers	154
Career Skills: Database Administrators	157
Summary	158
Key Terms	159
Module Review Questions	159
Problems	159
Critical Thinking Question	161
JC Consulting Case Exercises	161
Problems	161
Critical Thinking Questions	162

Pitt Fitness Case Exercises	162
Problems	162
Critical Thinking Questions	164
Sports Physical Therapy Case Exercises	164
Problems	164
Critical Thinking Questions	166

Module 5

<i>Database Design: Normalization</i>	167
Introduction	167
Case Study: Faculty/Student Advising Assignments	167
Data Modification Anomalies	168
Functional Dependence	170
Keys	171
First Normal Form	171
Atomic Values	173
Breaking Out Atomic Values Using Query Design View	173
Creating a Blank Database with Access	174
Importing Excel Data into an Access Database	174
Algorithms	175
Creating Fields	178
Creating New Fields in Table Design View	178
Updating Fields	179
Updating Field Values Using Query Design View	179
Creating the 1NF Table	181
Creating a New Table in Query Design View	181
Using Atomic Values for Quantities	183
Finding Duplicate Records	184
Finding Duplicate Records in Query Design View	184
Second Normal Form	186
Benefits of Normalization	190
Third Normal Form	190
Fourth Normal Form	191
Creating Lookup Tables in Query Design View	192
Beyond Fourth Normal Form	194
Summary	196
Key Terms	196
Module Review Questions	197
Problems	197
Critical Thinking Questions	198
JC Consulting Case Exercises	198
Problems	198
Critical Thinking Questions	199
Pitt Fitness Case Exercises	200
Problems	200
Critical Thinking Questions	202
Sports Physical Therapy Case Exercises	202
Problems	202
Critical Thinking Questions	205

Module 6

<i>Database Design: Relationships</i>	207
Introduction	207
User Views	208
Documenting a Relational Database Design	209
Database Design Language (DDL)	209
Setting Keys and Indexes	211

Entity-Relationship (E-R) Diagrams	213
Crow's Foot Notation	214
Microsoft Access E-R Diagram in the Relationships window	215
The Entity-Relationship Model (ERM)	217
Exploring One-to-Many Relationships in Access	220
Table Datasheet View	220
Subdatasheets	221
Lookup Properties	222
Subforms	225
Working with One-to-Many Relationships in Query Datasheet View	225
Other Relationship Types	229
One-to-One Relationships	229
Many-to-Many Relationships	232
Summary	235
Key Terms	235
Module Review Questions	236
Problems	236
Critical Thinking Questions	237
JC Consulting Case Exercises	237
Problems	237
Critical Thinking Questions	238
Pitt Fitness Case Exercises	238
Problems	238
Critical Thinking Questions	239
Sports Physical Therapy Case Exercises	239
Problems	239
Critical Thinking Questions	240
Module 7	
<i>Database Management Systems Processes and Services</i>	241
Introduction	241
Create, Read, Update, and Delete Data	242
Provide Catalog Services	243
Catalog Services in Microsoft Access	243
Catalog Services in Enterprise Database Management Systems	244
Support Concurrent Updates	244
Concurrent Updates in Microsoft Access	244
Concurrent Updates in Enterprise Database Management Systems	245
Recover Data	246
Recovering Data in Microsoft Access	246
Recovering Data in Enterprise Database Management Systems	247
Forward Recovery	248
Backward Recovery	249
Provide Security Services	250
Encryption	250
Authentication	250
Authorization	250
Views	250
Privacy	250
Provide Data Integrity Features	251
Support Data Independence	252
Adding a Field	252
Changing the Property of a Field	252
Managing Indexes	252
Changing the Name of a Field, Table, or View	252
Adding or Changing a Relationship	252
Support Data Replication	253
Summary	254
Key Terms	254

Module Review Questions	255
Problems	255
Critical Thinking Questions	256
JC Consulting Case Exercises	256
Problems	256
Critical Thinking Questions	257
Pitt Fitness Case Exercises	257
Problems	257
Critical Thinking Questions	259
Sports Physical Therapy Case Exercises	259
Problems	259
Critical Thinking Questions	260
Module 8	
<i>Database Industry Careers</i>	261
Introduction	261
Careers in the Database Industry	261
Role of a Database Administrator	262
Duties and Responsibilities of a DBA	262
Database Policy Formulation and Enforcement	263
Access Privileges	263
Security	264
Disaster Planning	264
Archiving	265
Database Technical Functions	266
Database Design	266
SQL and Views	266
Testing	267
Performance Tuning	267
DBMS Maintenance	267
Database Administrative Functions	268
Data Dictionary Management	268
Training	268
Professionals Reporting to the DBA	268
Responsibilities of a Data Analyst	270
Responsibilities of a Data Scientist	270
Database Industry Certifications	271
Summary	274
Key Terms	274
Module Review Questions	275
Problems	275
Critical Thinking Questions	276
JC Consulting Case Exercises	276
Problems	276
Critical Thinking Questions	276
Pitt Fitness Case Exercises	277
Problems	277
Critical Thinking Questions	278
Sports Physical Therapy Case Exercises	278
Problems	278
Critical Thinking Questions	279
Module 9	
<i>Database Industry Trends</i>	281
Introduction	281
Database Architectures	281
Centralized Approach	281
Cloud Computing	283
Personal Computer Revolution	284

Client/Server Architecture	286
Access and Client/Server Architecture	286
Three-Tier Client/Server Architecture	287
Data Warehouses	289
Online Analytical Processing (OLAP)	289
Codd's Rules for OLAP Systems	292
Current OLAP Vendors	292
Distributed Databases	293
Rules for Distributed Databases	293
Summary of Current Database Architecture Implementations	293
Selecting a Relational Database System	294
Software Solution Stacks	297
NoSQL Database Management Systems	299
Object-Oriented Database Management Systems	300
Rules for Object-Oriented Database Management Systems	301
Big Data	302
Google Analytics	302
Data Formats	303
XML	304
JSON	306
Data Visualization Tools	307
Visualization Tools in Microsoft Excel	307
Microsoft Power BI	309
Tableau	309
Summary	311
Key Terms	311
Module Review Questions	313
Problems	313
Critical Thinking Questions	314
JC Consulting Case Exercises	314
Problems	314
Critical Thinking	315
Pitt Fitness Case Exercises	315
Problems	315
Critical Thinking Questions	316
Sports Physical Therapy Case Exercises	317
Problems	317
Critical Thinking Questions	318
Appendix A	
<i>Comprehensive Design Example: Douglas College</i>	319
Douglas College Requirements	319
General Description	319
Report Requirements	319
Update (Transaction) Requirements	323
Douglas College Information-Level Design	323
Final Information-Level Design	340
Exercises	341
Appendix B	
<i>SQL Reference</i>	349
Alter Table	349
Column or Expression List (Select Clause)	349
Computed Fields	350
Functions	350
Conditions	350
Simple Conditions	350
Compound Conditions	350

BETWEEN Conditions	351
LIKE Conditions	351
IN Conditions	351
CREATE INDEX	351
CREATE TABLE	352
CREATE VIEW	352
DATA TYPES	353
DELETE ROWS	353
DROP INDEX	354
DROP TABLE	354
GRANT	354
INSERT	354
INTEGRITY	355
JOIN	355
REVOKE	356
SELECT	356
SELECT INTO	357
SUBQUERIES	357
UNION	358
UPDATE	358
Appendix C	
<i>FAQ Reference</i>	359
Appendix D	
<i>Introduction To MySQL</i>	361
Introduction	361
Downloading and Installing MySQL	361
Running MySQL Workbench and Connecting to MySQL Server	365
Opening an SQL File In MySQL Workbench	366
Running an SQL Script in MySQL Workbench	367
Refreshing Schemas in MySQL Workbench	367
Viewing Table Data in MySQL Workbench	367
Writing SQL in MySQL Workbench	368
Practicing With MySQL Workbench	369
Summary	370
Key Terms	370
Appendix E	
<i>A Systems Analysis Approach to Information-Level Requirements</i>	371
Introduction	371
Information Systems	371
System Requirement Categories	372
Output Requirements	372
Input Requirements	372
Processing Requirements	373
Technical and Constraining Requirements	373
Determining System Requirements	373
Interviews	373
Questionnaires	374
Document Collection	374
Observation	374
Research	374
Transitioning From Systems Analysis to Systems Design	374
Key Terms	375
Critical Thinking Questions	375
Glossary	377
Index	391

ABOUT THIS BOOK

This book is intended for anyone who is interested in the database industry. As a textbook, it is appropriate for business and computer science students in an introductory database concepts course. Traditional database concepts such as data normalization, table relationships, and SQL are thoroughly covered. Emerging database concepts and trends are explained using realistic, current, and practical examples. Anyone who is currently managing or using an existing database will find the book helpful, given that it describes database best practices and how to create and leverage the benefits of a healthy database. Anyone wanting to enter the database industry will enjoy the book given that both traditional and new careers in the field are carefully explored.

This book assumes that students have some familiarity with computers such as solid Microsoft Office and file and folder management skills. Students do not need to have a background in programming, only a desire and curiosity to learn about how to harness the power of databases.

CHANGES TO THE TENTH EDITION

The Tenth Edition includes the following new features and content:

- Full-color screen shots using Access 2019.
- Extensive coverage of the relational model, including hands-on exercises that guide students through the data normalization process, how to build relationships, how to query a database using a QBE tool, and how to write SQL (Structured Query Language) to create, update, and select data from a relational database.
- Hands-on exercises for creating and using Microsoft Access data macros to accomplish the similar functionality to SQL triggers.
- A new module on careers for those interested in database administration, data analysis, data science, and other related careers in the database industry.
- General information about current trends in database management systems, including the management of “big data,” object-oriented database management systems, NoSQL systems, data-driven web apps, and popular software application stacks.
- A new case study, JC Consulting, a web development and data consulting company, used to illustrate skills within each module.
- An updated end-of-module case study, Pitt Fitness.
- Updated exercises for Sports Physical Therapy, the second end-of-module case study.
- New critical-thinking questions and exercises that reinforce problem-solving and analytical skills.
- New data files if using MySQL with the database cases.
- An updated appendix to guide users through the installation of MySQL.

MINDTAP FEATURES

- Integration with SAM and SAM projects, Cengage’s leading-edge, hands-on skills assessment management system. These activities provide auto-grading and feedback of students’ mastery of Microsoft Access.
- Module quiz evaluates students’ understanding of foundational database concepts in each module.
- Quick Lesson concept videos dig deeper into database concepts and innovations to improve students’ comprehension.
- Candid Career videos highlight database and data-focused career paths to inform students of various careers and ways to apply their database skills.

ORGANIZATION OF THE TEXTBOOK AND KEY FEATURES

Updated Case Studies

Module 1 covers essential database terminology and examines the benefits of and key factors for a healthy relational database system. It also introduces the JC Consulting, Pitt Fitness, and Sports Physical Therapy case studies and databases that are used throughout the textbook and end-of-module exercises.

Detailed Coverage of the Relational Model, Query-By-Example (QBE), and SQL

Module 2 includes in-depth, hands-on exercises to select and summarize data from a relational database using Access's QBE (Query By Example), Query Design View. Module 2 also covers relational algebra, foundational information for SQL. Module 3 is an in-depth look at selecting and summarizing data with SQL. Module 4 covers advanced topics for the relational model such as indexes, data integrity, security, inner and outer joins, triggers, and data macros.

Normalization Coverage

Module 5 dives into the data normalization process, taking a new list of nonnormalized data from first to second to third (Boyce-Codd normal form) to fourth normal form using hands-on exercises. The module describes the update anomalies associated with lower normal forms. Access queries and tools are used to take the data through the normalization process. Access is used to create new, normalized tables, data is analyzed and updated, fields and data types are properly defined, primary and foreign key fields are created, and lookup tables are created.

Database Design

Module 6 continues the process of building a healthy relational database by focusing on table relationships, again using hands-on exercises to illustrate the concepts. Database Design Language (DDL), E-R diagrams, and entity-relationship models are all used to document and implement one-to-many relationships between the tables of data that were properly normalized in Module 5.

Functions Provided by a Database Management System

Module 7 covers traditional database management processes and concerns such as the data recovery processes, security issues, data integrity and concurrency issues, data replication, and database documentation features.

Careers in the Database Industry

Module 8 explores the jobs and careers in the database industry starting with the traditional career of a database administrator (DBA), as well as the emerging areas of data analysts and data scientists. Valuable educational credentials and industry certifications are identified.

Trends

Module 9 compares and contrasts historical mainframe database management systems with current data management trends such as distributed database management systems, client/server systems, data warehouses, object-oriented database management systems, web access to databases, XML, and JSON.

TEACHING TOOLS

When this book is used in an academic setting, instructors may obtain the following teaching tools from Cengage Learning through their sales representative or by visiting www.cengage.com:

- **Instructor's Manual.** The Instructor's Manual includes suggestions and strategies for using this text. It includes many ideas for classroom activities and graded projects.

For instructors who want to use an Access text as a companion to the Tenth Edition, consider *Microsoft Access 2019: Comprehensive* by Friedrichsen, also published by Cengage.

- **Data and Solution Files.** Data and solution files are available at www.cengage.com. Data files consist of copies of the JC Consulting, Pitt Fitness, and Sports Physical Therapy databases that are usable in Access 2010, Access 2013, Access 2016, Access 2019, and script files to create the tables and data in these databases in other systems, such as MySQL.
- **PowerPoint Presentations.** Microsoft PowerPoint slides are included for each module as a teaching aid for classroom presentations, to make available to students on a network for module review, or to be printed for classroom distribution. Instructors can add their own slides for additional topics they introduce to the class. The presentations are available at www.cengagebrain.com.

Cengage Learning Testing Powered by Cognero is a flexible, online system that allows you to:

- author, edit, and manage test bank content from multiple Cengage Learning solutions
- create multiple test versions in an instant
- deliver tests from your LMS, your classroom, or wherever you want

GENERAL NOTES TO THE STUDENT

Within each major section, special questions or “Your Turn” exercises have been embedded. Sometimes the purpose of these exercises is to ensure that you understand crucial material before you proceed. In other cases, the questions are designed to stretch your understanding into real world application of the concepts. Read the question or exercise, try to answer the question or complete the exercise on your own, and then compare your work against the answer that is provided.

You also will find complementary SAM projects in MindTap, which allow you to apply the concepts learned in a meaningful hands-on project. These critical thinking exercises help you solidify the process and well as solve the problem.

The end-of-module material consists of a summary, a list of key terms, review questions, and exercises for the JC Consulting, Pitt Fitness, and Sports Physical Therapy databases. The summary briefly describes the material covered in the module. The review questions require you to recall and apply the important material in the module. Review questions and exercises include critical-thinking questions to challenge your problem-solving and analytical skills.

ACKNOWLEDGMENTS

We would like to acknowledge all of the talented professionals who made contributions during the creation of this book. We also want to thank those professors and students who use this book to teach and learn. Stay curious!

MODULE 1

INTRODUCTION TO DATABASE MANAGEMENT

LEARNING OBJECTIVES

- Examine JC Consulting (JCC), the company used for many of the examples throughout the text
- Define basic database terminology
- Describe database management systems (DBMSs)
- Explain the advantages and key factors for a healthy relational database system
- Prepare for a career in database administration
- Review Pitt Fitness, a company used in a case that appears at the end of each module
- Review Sports Physical Therapy, a company used in another case that appears at the end of each module

INTRODUCTION

In this module, you will examine the requirements of JC Consulting (JCC), a company that will be used in many examples in this text. You will learn how JCC initially stored its data, what problems employees encountered with that storage method, and why management decided to employ a database management system (DBMS). You will also study the basic terminology and concepts of relational databases, database management systems, and big data. You will learn the advantages and key factors of a properly designed relational database. Finally, you will examine the database requirements for Pitt Fitness and Sports Physical Therapy, the companies featured in the cases that appear at the end of each module.

JC CONSULTING COMPANY BACKGROUND

JC Consulting (JCC) is a digital development and consulting business. The founder, Jacqueline Cabrero, started the business in the mid-1990s when the Internet became publicly available. Jacqueline grew the business from a sole proprietorship that built static webpages for small businesses to a firm with more than 20 employees. JCC's services range from building websites and web apps to back-end database conversions and programming.

Initially, Jacqueline kept track of her clients and project bids in a spreadsheet. As the company grew, she used a homegrown project estimator program to bid new projects. Jacqueline has now determined that the company's recent growth means it is no longer feasible to use those programs to maintain its data.

What led JCC to this decision? One of the company's spreadsheets, shown in Figure 1-1, displays project estimates, and illustrates JCC's problems with the spreadsheet approach. For each estimate, the spreadsheet displays the number and name of the client, the project estimate number and date, the task ID, a description of the task, and a quoted price for that task. Tri-Lakes Realtors received two different project estimates (ProjectIDs 1 and 31). In the first project estimate, Tri-Lakes Realtors needed general help to establish online goals. In the second estimate, the agency needed help with relational database design and data conversion. The result was seven lines in the spreadsheet, two project estimate numbers, and several task IDs.

ClientID	ClientName	ProjectID	ProjectStartDate	TaskID	Description	Estimate
1	Tri-Lakes Realtors	1	2/6/19	MEET00	Initial customer meeting	0.00
1	Tri-Lakes Realtors	1	2/6/19	PLAN01	Establish goals with client	200.00
1	Tri-Lakes Realtors	31	3/1/20	MEET00	Initial customer meeting	0.00
1	Tri-Lakes Realtors	31	3/1/20	DB01	Design relational database	1,000.00
1	Tri-Lakes Realtors	31	3/1/20	MEET01	Meet with client	100.00
1	Tri-Lakes Realtors	31	3/1/20	DB02	Convert data	5,000.00
1	Tri-Lakes Realtors	31	3/1/20	MEET01	Meet with client	100.00
2	Project Lead The Way	7	9/2/19	MEET00	Initial customer meeting	0.00
2	Project Lead The Way	7	9/2/19	CODE04	Code SQL	750.00
2	Project Lead The Way	7	9/2/19	TEST01	Test technology	300.00
2	Project Lead The Way	7	9/2/19	TEST02	Test performance	300.00
2	Project Lead The Way	7	9/2/19	MEET01	Meet with client	200.00
3	Midstates Auto Auction	3	3/11/19	MEET00	Initial customer meeting	0.00
3	Midstates Auto Auction	3	3/11/19	PLAN01	Establish goals with client	240.00
3	Midstates Auto Auction	8	1/6/20	PLAN01	Establish goals with client	200.00
3	Midstates Auto Auction	8	1/6/20	MEET00	Initial customer meeting	0.00
3	Midstates Auto Auction	8	1/6/20	CODE10	Apply SEO	1,500.00
4	Bretz & Hanna Law Firm	4	4/30/20	PLAN01	Establish goals with client	240.00

FIGURE 1-1 Project estimates spreadsheet

Data redundancy is one problem that employees have with the project estimates spreadsheet. Data redundancy, sometimes shortened to **redundancy**, is the duplication of data, or the storing of the same data in more than one place. In the project estimates spreadsheet, redundancy occurs in the ClientName column because the name of a client is listed for each line item in each project estimate. Redundancy also occurs in other columns, such as in the ProjectID column when a project has more than one task, or in the TaskID and Description columns when two projects contain the same tasks.

Q & A 1-1

Question: What problems does redundancy cause?

Answer: Redundant data yields a higher frequency of data errors and inconsistencies, which in turn leads to poorer decision making. For example, if you entered “Tri-Lakes Realtors” and “Tri Lakes Realtors” on separate rows in the ClientName column, you would be unsure about the correct version of this client’s name. Summarized data about this client would not be complete or correct given the client would be misinterpreted as two different companies because of the two different spellings.

When you need to change data, redundancy also makes your changes more cumbersome and time-consuming. For example, to change a client’s name, you would need to update it in each row where it appears. Even if you use a global find-and-replace feature, multiple changes require more editing time than updating the client name in one location.

Finally, while storage space is relatively inexpensive, redundancy wastes space because you’re storing the same data in multiple places. This extra space results in larger spreadsheets that require more space in memory and storage. Larger-than-necessary files also take longer to save and open.

Difficulty accessing related data is another problem that employees at JCC encounter with their spreadsheets. For example, if you want to see a client’s address, you must open and search another spreadsheet that contains this data because the client’s address is not currently stored in the project estimates spreadsheet.

Spreadsheets also have limited security features to protect data from being accessed by unauthorized users. In addition, a spreadsheet’s data-sharing features prevent multiple employees from updating data in one spreadsheet at the same time. Finally, if JCC estimates continue to increase at their planned rate,

spreadsheets have inherent size limitations that will eventually force the company to split the project estimates into multiple spreadsheets. Splitting the project estimates into more than one spreadsheet would create further redundancy, data organization, and reporting problems. For these reasons, JCC decided to replace the estimating spreadsheet with a **database**, a collection of data organized in a manner that allows access, retrieval, and use of that data.

After making the decision, management has determined that JCC must maintain the following information about its employees, clients, tasks, and project estimates:

- For employees: Employee ID, last name, first name, hire date, title, and salary
- For clients: Client ID, name, address, and government status
- For projects: Project ID, start date, task IDs, task descriptions, costs, project notes, and task notes

Figure 1-2 shows a sample project estimate.

JC Consulting Project Estimate		Creating data-driven web solutions since 1995	
2000 Pine Four Blvd, Boston, MA 02210 (617) 868-8900			
Brief project description	ProjectID	Estimated Start Date	Estimate
Tri-Lakes Realtors Three existing databases need to be merged into one.	31	1/1/2020	
Tasks			
Initial customer meeting			0
Design relational database			1,000
Meet with client			100
Convert data			5,000
Meet with client			100
			\$1,325

FIGURE 1-2 Sample project estimate

- The top of the estimate contains the company name, JC Consulting, the words “Project Estimate,” and company contact information.
- The body of the estimate contains the name of the client for which the project estimate has been created, a brief description of the project, the project ID, an estimated project start date, and one or more line items. Each line item contains a task description and the estimated price for that task.
- The bottom of the estimate contains the total estimated price for the project.

Internally, JCC also must store the following items for each client’s estimate:

- For each project estimate, JCC must store the client’s address as well as the employee assigned as the project leader for the project.
- For each task line item, JCC not only stores the project ID but also the task ID, the estimated completion date for that task, and task notes. If the task is considered complex or risky, a factor is applied to increase the estimate. The task description and task category (coding, meeting, planning, and so forth) are also stored for each task.
- The overall project estimate total is not stored. Instead, the total is calculated whenever a project estimate is printed or displayed on the screen.

The problem facing JCC is common to many businesses and individuals that need to store and retrieve data in an efficient and organized way. JCC is interested in several areas of information such as employees, clients, estimates, and tasks. A school is interested in students, faculty, and classes; a real estate agency is interested in clients, houses, and agents; a distributor is interested in customers, orders, and inventory; and a car dealership is interested in clients, vehicles, and manufacturers.

The difficult question is not how to manage single categories or lists of information, but how to manage the lists of data and the relationships between the lists. For example, besides being interested in client and project estimate information, JCC also wants to know which clients have received more than one project estimate. The company wants to know which employees are assigned as the lead to which projects and which tasks are most commonly added to which projects.

Likewise, a school is not only interested in students and classes but also which students are enrolled in which classes. A real estate agency is not only interested in their lists of agents and homes for sale; they also want to know which agents are listing or selling the most homes. A distributor wants to know which customers are ordering specific inventory items, and a car dealership not only wants to know about their customer base and car inventory but also which customers are buying multiple cars over time.

SELECTING A DATABASE SOLUTION

After studying the alternatives to using spreadsheet software, JCC decided to switch to a relational database system. A **relational database** is a structure that contains data about many categories of information as well as the relationships between those categories. The JCC database, for example, will contain information about employees, clients, project estimates, and tasks. It also will provide facts that relate employees to the projects they manage, clients to their project estimates, and the project estimates to the tasks that are contained within each project.

With a relational database, JCC will be able to retrieve a particular project estimate and identify which client and tasks belong to that estimate. In addition, employees can start with a client and find all project estimates, including the individual tasks within the project that have been prepared for that client. Using a relational database, JCC can use the data to produce a variety of regular periodic or ad hoc reports to summarize and analyze the data in an endless number of ways.

Defining Database Terminology

Some terms and concepts in the database environment are important to know. The terms *entity*, *attribute*, and *relationship* are fundamental when discussing databases. An **entity** is a person, place, event, item, or other transaction for which you want to store and process data. The entities of interest to JCC, for example, are employees, clients, project estimates, and tasks. Entities are represented by a **table** of data in relational database systems.

An **attribute** is a characteristic or property of an entity. For the entity *employee*, for example, attributes might include such characteristics as first and last name, employee number, and date of hire. For JCC, the attributes of interest for the *client* entity include client name, street, city, state, zip code, and whether the client is a government body. An attribute is also called a **field** or **column** in many database systems.

Figure 1-3 shows two entities, Clients and Projects, along with the attributes for each entity. The Clients entity has seven attributes: ClientID, ClientName, Street, City, State, Zip, and Government (whether the client is any type of government institution). Attributes are similar to columns in a spreadsheet. The Projects entity (which represents project estimates) has five attributes: ProjectID, ProjectStartDate, ClientID, EmployeeID, and ProjectNotes. Entity (table) names and attribute (field) names should be easy to understand, concise, indicative of their content, and contain no spaces or other special characters.

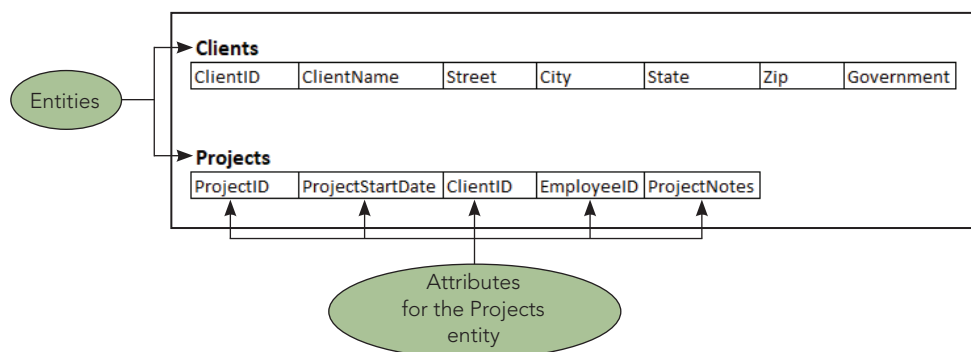


FIGURE 1-3 Entities and attributes

A **relationship** is an association between entities. For example, there is an association between clients and projects. A client is *related to* all of its projects, and a project is *related to* its client.

This relationship is called a **one-to-many relationship** because each client may be associated with *many* projects, but each project is associated with only *one* client. In this type of relationship, the word *many* is

used differently from everyday English because it does not always indicate a large number. In this context, the term *many* means that a client can be associated with *any* number of projects. That is, a given client can be associated with zero, one, or more projects.

A one-to-many relationship often is represented visually as shown in Figure 1-4. In such a diagram, entities and attributes are represented in precisely the same way as they are shown in Figure 1-3. A line connecting the entities represents the relationship. The *one* entity of the relationship (in this case, Clients) does not have an arrow on its end of the line, and the *many* entity in the relationship (in this case, Projects) is indicated by a single-headed arrow.

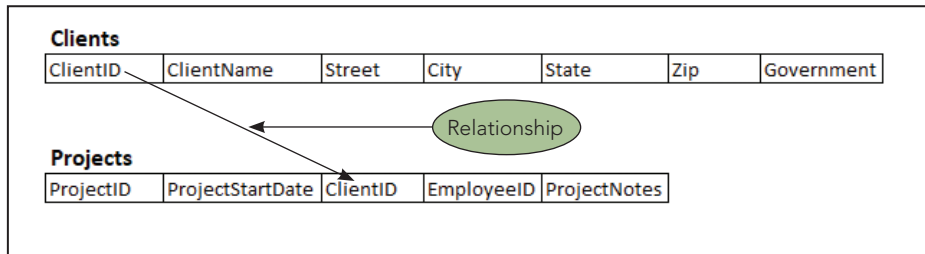


FIGURE 1-4 One-to-many relationship

Q & A 1-2

Question: What happens when the relationship between two entities is best defined as “many-to-many” because one record in one entity relates to many records in the other entity and vice versa? For example, at a college, one student may be related to many classes, and one class is also related to many students.

Answer: A **many-to-many relationship** cannot be directly created in a relational database. To accommodate this relationship between two tables, a third table must be inserted, which is used on the “many” side of two one-to-many relationships with the original two tables. In this case, an Enrollments table could be added between the Students and Classes tables. One student may enroll in many classes. One class may have many enrollments. Two tables that are on the “one” side of a one-to-many relationship with the same table have, by definition, a many-to-many relationship with each other.

STORING DATA

A spreadsheet that is used to store data, often called a **data file**, typically stores data as one large table. Data stored this way is also referred to as a **flat file** because lists in a spreadsheet have no relationships with other lists. A relational database, however, not only stores information about multiple entities in multiple tables but also identifies the relationships between those tables.

For example, in addition to storing information about projects and clients, the JCC database will hold information relating clients to the various project estimates that were created for that client, which employee is assigned as the project leader for that particular project, and more. A relational database can store information about multiple types of entities, the attributes of those entities, and the relationships among the entities.

How does a relational database handle these entities, attributes, and relationships among entities? Entities and attributes are fairly straightforward. Each entity has its own table. The JCC database, for example, will have one table for employees, one table for clients, one table for the project estimates, and so on. The attributes of an entity become the columns in the table. Within each table, a **row** of data corresponds to one record. A **record** is a group of fields (attributes) that describe one item in the table (entity).

What about relationships between entities? At JCC, there is a one-to-many relationship between clients and projects. But how is this relationship established in a relational database system? It is handled by using a common field in the two tables to tie the related records from each table

together. Consider Figure 1-4. The ClientID column in the Clients table and the ClientID column in the Projects table are used to build the relationship between clients and projects. Given a particular ClientID, you can use these columns to determine all the projects that have been estimated for that client; given a ProjectID, you can use the ClientID columns to find the client for which that project estimate was created.

How will JCC store its data via tables in a database? Figure 1-5 shows sample data for JCC.

Employees

EmployeeID	LastName	FirstName	HireDate	Title	Salary
19	Kohn	Ali	01-Jan-20	Project Leader	\$5,000.00
22	Kaplan	Franco	01-Feb-20	Programmer	\$5,500.00
35	Prohm	Nada	29-Feb-20	Customer Support Specialist	\$4,000.00
47	Alvarez	Benito	31-Mar-20	Front End Developer	\$5,200.00
51	Shields	Simone	30-Apr-20	Network Specialist	\$7,000.00
52	Novak	Stefan	01-Jan-19	Project Leader	\$8,000.00
53	Anad	Sergei	01-Jan-19	Front End Developer	\$5,300.00
54	Allen	Sasha	01-Jan-19	Programmer	\$7,000.00
55	Winter	Wendy	31-Dec-20	Front End Developer	\$4,300.00
56	Reddy	Kamal	01-Sep-19	Programmer	\$6,200.00
57	Yang	Tam	30-Apr-21	Front End Developer	\$5,000.00
58	Young	Solomon	01-Jan-19	Programmer	\$5,500.00
59	Santana	Carmen	01-Jan-19	Front End Developer	\$4,800.00
60	Lu	Chang	01-Mar-19	Database Developer	\$7,900.00
61	Smirnov	Tovah	01-Oct-19	Programmer	\$6,000.00
62	Turner	Jake	31-Mar-21	Database Developer	\$7,800.00
63	Geller	Nathan	01-Jan-19	Project Leader	\$8,100.00
64	Lopez	Miguel	01-Jan-19	Programmer	\$6,200.00
65	Garcia	Hector	01-Apr-23	UI Designer	\$7,000.00
66	Roth	Elena	31-Oct-20	Network Specialist	\$7,000.00
67	Horvat	Nigel	30-Apr-24	UI Designer	\$6,300.00

Clients

ClientID	ClientName	Street	Zip	Government
1	Tri-Lakes Realtors	135 E Jefferson St	02447	FALSE
2	Project Lead The Way	762 Saratoga Blvd	02446	TRUE
3	Midstates Auto Auction	9787 S Campbell Ln	01355	FALSE
4	Bretz & Hanna Law Firm	8101 N Olive Dr	01431	FALSE
5	Aspire Associates	5673 South Ave	01431	FALSE
6	Bounteous	9898 Ohio Ave	02770	FALSE
7	Netsmart Solutions	4091 Brentwood Ln	01354	FALSE
8	Loren Group	9565 Ridge Rd	02466	FALSE
9	Associated Grocers	231 Tecumsa Rd	02532	FALSE
10	Jobot Developers	1368 E 1000 St	02330	FALSE
11	Harper State Bank	1865 Forrest Dr	01571	FALSE
12	MarketPoint Sales	826 Hosta St	01983	FALSE
13	SecureCom Wireless	5280 Industrial Dr	01852	FALSE
14	The HELPCard	840 Boonville Ave	02466	TRUE
15	Jillian Henry & Associates	815 E California St	02113	FALSE
16	Pediatric Group	4940 W Farm Rd	02113	FALSE
17	SkyFactor	1736 Sunshine Dr	02726	FALSE
18	NuCamp	2500 E Kearny St	01431	FALSE
19	Wu Electric	5520 S Michigan	02447	FALSE
20	Juxly Engineering	4238 Rumsfield Rd	02148	FALSE
21	Carta Training	2445 N Airport Dr	02446	FALSE

FIGURE 1-5 Sample data for JCC (continued)

Projects

ProjectID	ProjectStartDate	ClientID	EmployeeID	ProjectNotes
1	06-Feb-19	1	52	Client wants digital solutions to help rebrand company name to emphasize commercial real estate.
2	07-Feb-19	2	63	Client needs help converting, organizing, and managing various sources/formats of donor and donation data.
3	11-Mar-19	3	52	Client wants to establish SEO goals.
4	10-Apr-20	4	52	Client wants to set up an internal server as well as help with a domain name.
7	02-Sep-19	2	63	Client has used the database for several months and now needs new report
8	06-Jan-20	3	52	Develop and implement website SEO strategy.
9	10-Feb-20	6	63	Needs help to manage and organize internal data.
10	31-Mar-21	7	19	Develop new website content.
11	30-Apr-20	9	19	Client needs internal database to manage personnel.
13	30-Nov-20	10	64	Client needs subcontracting help installing a new database for a WordPress site.
14	09-Dec-20	15	19	Client needs new functionality for current JavaScript application.
15	21-Dec-20	14	19	Client needs new functionality for current Ruby/Rails application.
16	04-Jan-21	11	52	Client needs help with server security.
17	15-Feb-21	12	52	Current online sales solution is unreliable.
18	14-Apr-21	6	63	Client needs internal database to manage inventory.
19	04-Jun-21	13	52	Client needs new functionality for current C# / ASP.NET application.
20	30-Jul-21	1	63	Client needs full website reskin.
21	31-Aug-21	16	19	Client needs help with data analytics.
22	30-Sep-21	20	19	Client needs an online reference database
23	12-Nov-21	18	63	Client needs new blog and current pages updated to include responsive web design principles for mobile devices.

ProjectLineItems

ProjectLineItemID	ProjectID	TaskID	TaskDate	Quantity	Factor	ProjectLineItemNotes
1	1	MEET00	06-Feb-19	1	1.00	
2	1	PLAN01	06-Feb-19	1	1.00	
4	2	MEET00	07-Feb-19	1	1.00	
5	2	PLAN01	07-Feb-19	1	1.00	
6	2	DB01	15-Mar-19	1	1.30	Data is stored in multiple spreadsheets.
7	2	DB02	15-Apr-19	20	1.30	Data is not consistent between spreadsheets.
8	3	MEET00	11-Mar-19	1	1.00	
9	3	PLAN01	11-Mar-19	1	1.20	Owner is difficult to pin down.
10	4	MEET00	10-Apr-20	1	1.00	
11	4	PLAN01	10-Apr-20	1	1.20	Two principal attorneys must agree.
12	4	SERV01	11-May-20	1	1.00	
13	4	SERV02	10-Jun-20	1	1.30	Security is a paramount issue.
17	11	MEET00	30-Apr-20	1	1.00	
18	11	PLAN01	30-Apr-20	1	1.00	
19	9	MEET00	10-Feb-20	1	1.00	
20	9	PLAN01	10-Feb-20	1	1.00	
25	9	PLAN10	17-Feb-20	1	1.00	
26	18	MEET00	14-Apr-21	1	1.00	
27	20	MEET00	30-Jul-21	1	1.00	
28	20	PLAN01	30-Jul-21	1	1.00	
29	20	PLAN02	30-Jul-21	1	1.00	

FIGURE 1-5 Sample data for JCC (continued)

TaskMasterList

TaskID	Description	CategoryID	Per	Estimate
CODE01	Code PHP	Coding	Hour	\$150.00
CODE02	Code C# in ASP.NET	Coding	Hour	\$150.00
CODE03	Code Ruby on Rails	Coding	Hour	\$150.00
CODE04	Code SQL	Coding	Hour	\$150.00
CODE05	Code HTML	Coding	Hour	\$100.00
CODE06	Code CSS	Coding	Hour	\$100.00
CODE07	Code JavaScript	Coding	Hour	\$125.00
CODE08	Perform analytics	Coding	Hour	\$100.00
CODE09	Select technology stack	Coding	Hour	\$200.00
CODE10	Apply SEO	Coding	Hour	\$125.00
CODE12	Create prototype	Coding	Hour	\$150.00
CODE13	Code WordPress	Coding	Hour	\$100.00
CODE14	Code Python	Coding	Hour	\$150.00
CODE15	Create shopping cart	Coding	Hour	\$125.00
CODE16	Code other	Coding	Hour	\$150.00
DB01	Design relational database	Database	Project	\$1,000.00
DB02	Convert data	Database	Hour	\$125.00
DB03	Install MySQL database	Database	Project	\$500.00
DB04	Install SQL Server database	Database	Project	\$500.00
DB05	Install Access database	Database	Project	\$400.00
MEET00	Initial customer meeting	Meeting	Project	\$0.00

FIGURE 1-5 Sample data for JCC (*continued*)

In the Employees table, each employee has a unique EmployeeID number in the first column. The name of the employee with the EmployeeID value of 19 in the first record is Ali Kohn. His hire date was 01-Jan-20, his title is Project Leader, and his monthly salary is \$5,000.00.

The Clients table contains one record for each client, which is uniquely identified by a ClientID number in the first column. The client name, street, zip, and whether the client is a governmental entity are also stored for each record.

In the Projects table, a unique ProjectID number for each project is positioned in the first column. The project start date and project notes are also recorded in fields named ProjectStartDate and ProjectNotes. The ClientID field contains a number that connects the Projects table with the Clients table. In the first record for ProjectID 1, the ClientID value is also 1, which connects with Tri-Lakes Realtors in the Clients table. The EmployeeID value of 52 connects with Stefan Novak in the Employees table.

In the table named ProjectLineItems, each record represents one task for each project. The ProjectID value connects each record to a specific project in the Projects table. Note that the first two records contain a ProjectID value of 1 connecting them with the first project in the Projects table, which in turn connects them with the Tri-Lakes Realtors record in the Clients table. The TaskID column connects each line item with a record in the TaskMasterList table that further describes that task. The ProjectLineItems table also contains fields named TaskDate, Quantity, Factor, and ProjectLineItemNotes, which further describe each task for that project. The Quantity field is used to identify the estimated hours for the hourly tasks. The Factor field is a multiplier that represents additional risk or complexity. For example, 1.1 = 10% increase in perceived complexity or risk for that task. Both the Quantity and Factor fields are used to calculate the price for that task.

The TaskMasterList table uniquely identifies the different tasks that may appear on a project estimate with the TaskID field, and further describes each task with the Description, CategoryID, Per (per hour or per project), and Estimate fields. The Estimate field contains the dollar amount for that task. It is multiplied by the Quantity and Factor fields in the ProjectLineItems table to calculate the total estimated cost for that line item.

The table named ProjectLineItems might seem strange at first glance. Why do you need a separate table for the project line items? Couldn't the project line items be included in the Projects table? The Projects table *could* be structured as shown in Figure 1-6. Notice that this table contains the same projects and line items as those shown in Figure 1-5, with the same fields and data. However, the TaskID, TaskDate, Quantity, Factor, and ProjectLineItemNotes fields contain multiple entries.

ProjectID	ProjectStartDate	ClientID	EmployeeID	ProjectNotes	TaskID	TaskDate	Quantity	Factor	ProjectLineItemNotes
1	06-Feb-19	1	52	Client wants digital solutions to emphasize commercial real estate.	MEET00	06-Feb-19	1	1.00	
					PLAN01	06-Feb-19	1	1.00	
2	07-Feb-19	10	63	Client needs help converting, organizing, and managing donor and donation data.	MEET00	07-Feb-19	1	1.00	Data is stored in multiple spreadsheets. Data is not consistent between spreadsheets. Code SQL to code 4 queries.
					PLAN01	07-Feb-19	1	1.00	
					DB01	15-Mar-19	1	1.30	
					DB02	15-Apr-19	20	1.30	
					CODE04	15-May-19	4	1.00	
					TEST01	03-Jun-19	8	1.00	
					TEST02	03-Jun-19	8	1.00	
					MEET01	03-Jun-19	2	1.00	
					SUPP03	03-Jun-19	8	1.00	
3	11-Mar-19	3	52	Client wants to establish SEO goals.	MEET00	11-Mar-19	1	1.00	Owner is difficult to pin down.
					PLAN01	11-Mar-19	1	1.20	
4	10-Apr-20	4	52	Client wants to set up an internal server as well as help with a domain name.	MEET00	10-Apr-20	1	1.00	Two principal attorneys must agree. Security is a paramount issue.
					PLAN01	10-Apr-20	1	1.20	
					SERV01	11-May-20	1	1.00	
					SERV02	10-Jun-20	1	1.30	
					TEST01	15-Jun-20	16	1.00	
					TEST02	15-Jun-20	16	1.00	
					SUPP03	15-Jun-20	4	1.00	

FIGURE 1-6 Alternative Projects table structure

Q & A 1-3

Question: How is the information in Figure 1-5 represented in Figure 1-6?

Answer: Examine the ProjectLineItems table shown in Figure 1-5 and note the first two records are connected with ProjectID 1 in the Projects table. In Figure 1-6, the entire project estimate for ProjectID 1 is entered in one record. Two task IDs and the other fields that describe the two tasks for that project are entered together in the fields that describe the tasks.

Q & A 1-4

Question: Why does ProjectID 2 have such a large row in Figure 1-6?

Answer: Figure 1-6 shows one row (record) for each project (as opposed to one row for each task). Given that the estimate for ProjectID 2 has nine different tasks, ProjectID 2 requires nine different entries in each of the columns (fields) that describe the tasks for that project.

Figure 1-5 shows a single entry in each field of the ProjectLineItems table. In Figure 1-6, the fields that describe tasks contain multiple entries such as the TaskID, TaskDate, Quantity, Factor, and ProjectLineItemNotes. For example, ProjectID 1 consists of two tasks, and therefore two entries are placed in the TaskID, TaskDate, Quantity, and Factor fields because those fields describe the two tasks for that project. Other projects contain many more tasks and would have many more entries in those fields.

In general, tables that contain more than one piece of information per attribute (column or field) create several problems that can be eliminated with a proper relational database design. The following are some warning signs that your entities, attributes, and relationships are not properly designed:

- You need to enter more than one value in a particular field (see Figure 1-6).
- You are asked to enter two or more pieces of information in a field. For example, using one field to enter both first and last names means you cannot quickly and easily search, sort, and filter on either part of a person's name.
- You are asked to enter both values and units of measure in the same field. Entering numbers and text in the same field generally prevents you from calculating on the numeric part of the data.
- You find yourself adding new columns to handle multiple values for the same type of data. For example, to track employee salaries over time, you wouldn't want to create additional attributes in the Employees table with names such as Salary1, Salary2, and Salary3. A better approach would be to create a Salaries table and relate it to the Employees table. One employee record