



D HANDS-ON **DATA**BASE

An Introduction to
Database
Design and
Development

Second Edition

STEVE CONGER



OTHER MIS TITLES OF INTEREST

MIS:

Managing Information Technology, 7/e

Brown, DeHayes, Hoffer, Martin & Perkins ©2012

SharePoint for Students

Cole, Fox & Kroenke ©2012

Experiencing MIS, 4/e

Kroenke ©2014

Using MIS, 6/e

Kroenke ©2014

MIS Essentials, 3/e

Kroenke ©2014

Processes, Systems, and Information: An Introduction to MIS

Kroenke & McKinney ©2013

Management Information Systems, 13/e

Laudon & Laudon ©2014

Essentials of Management Information Systems, 10/e

Laudon & Laudon ©2013

IT Strategy, 2/e

McKeen & Smith ©2012

Essentials of Processes, Systems and Information: with SAP tutorials

McKinney & Kroenke ©2014

Information Systems Management In Practice, 8/e

McNurlin, Sprague & Bui ©2009

MIS Cases: Decision Making with Application Software, 4/e

Miller ©2009

Information Systems Today, 6/e

Valacich & Schneider ©2014

Information Systems in Organizations

Wallace ©2013

DATABASE:

Hands-on Database, 2/e

Conger ©2014

Essentials of Database Management

Hoffer, Topi, Ramesh ©2014

Modern Database Management, 11/e

Hoffer, Ramesh & Topi ©2013

Database Systems: Introduction to Databases and Data Warehouses

Jukic, Vrbsky & Nestorov ©2014

Database Concepts, 6/e

Kroenke & Auer ©2013

Database Processing, 13/e

Kroenke & Auer ©2014

SYSTEMS ANALYSIS AND DESIGN:

Object-Oriented Systems Analysis and Design

Ashrafi & Ashrafi ©2009

Modern Systems Analysis and Design, 7/e

Hoffer, George & Valacich ©2014

Systems Analysis and Design, 9/e

Kendall & Kendall ©2014

Essentials of Systems Analysis and Design, 5/e

Valacich, George & Hoffer ©2012

DECISION SUPPORT SYSTEMS:

Decision Support and Business Intelligence Systems, 10/e

Turban, Sharda & Delen ©2014

Business Intelligence, 3/e

Turban, Sharda, Delen & King ©2014

DATA COMMUNICATIONS & NETWORKING:

Applied Networking Labs, 2/e

Boyle ©2014

IT Networking Labs

Cavaiani ©2010

Digital Business Networks

Dooley ©2014

Business Driven Data Communications

Gendron ©2013

Business Data Networks and Security, 9/e

Panko & Panko ©2013

ELECTRONIC COMMERCE:

E-Commerce: Business, Technology, Society, 10/e

Laudon & Traver ©2014

Essentials of E-Commerce

Laudon & Traver ©2014

Electronic Commerce 2014

Turban, King, Lee, Liang & Turban ©2014

Introduction to Electronic Commerce, 3/e

Turban, King & Lang ©2011

ENTERPRISE RESOURCE PLANNING:

Enterprise Systems for Management, 2/e

Motiwalla & Thompson ©2012

PROJECT MANAGEMENT:

Project Management: Process, Technology and Practice

Vaidyanathan ©2013

CORPORATE SECURITY:

Applied Information Security

Boyle ©2010

The Management of Network Security

Carr, Snyder & Bailey ©2010

Corporate Computer Security, 3/e

Boyle & Panko ©2013

HANDS-ON DATABASE

AN INTRODUCTION TO DATABASE DESIGN AND DEVELOPMENT

Second Edition

Steve Conger

Seattle Central Community College

PEARSON

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

Editor in Chief: Stephanie Wall
Executive Editor: Bob Horan
Editorial Project Manager: Kelly Loftus
Editorial Assistant: Kaylee Rotella
Director of Marketing: Maggie Moylan
Senior Marketing Manager: Anne Fahlgren
Marketing Assistant: Gianna Sandri
Senior Managing Editor: Judy Leale
Project Manager: Meghan DeMaio
Creative Director: Jayne Conte
Cover Designer: Bruce Kenselaar
Cover Art: S. E/Fotolia
Media Editor: Alana Coles
Media Project Manager: Lisa Rinaldi
Full-Service Project Management/Composition: Anandkrishnan Natarajan/Integra Software Services
Printer/Binder: Courier Companies
Cover Printer: Lehigh-Phoenix Color
Text Font: 10/12, Palatino LT Std

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

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

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

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

Copyright © 2014, 2012 by Pearson Education, Inc., One Lake Street, Upper Saddle River, New Jersey 07458. All rights reserved. Manufactured in the United States of America. This publication is protected by Copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to 201-236-3290.

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

Library of Congress Cataloging-in-Publication Data

Conger, Steve.

Hands-on database: an introduction to database design and development/Steve Conger, Seattle Central Community College.—2 [edition].

pages cm

Includes index.

ISBN-13: 978-0-13-302441-8 (alk. paper)

ISBN-10: 0-13-302441-5 (alk. paper)

1. Database design. I. Title.

QA76.9.D26C644 2014

005.74'3--dc23

2013011938

10 9 8 7 6 5 4 3 2 1

PEARSON

ISBN 10: 0-13-302441-5

ISBN 13: 978-0-13-302441-8

DEDICATION

To Maureen, Bryan, and Chelsea

This page intentionally left blank

BRIEF CONTENTS

Preface ix

Chapter 1	Who Needs a Database	1
Chapter 2	Gathering Information	20
Chapter 3	Requirements and Business Rules	46
Chapter 4	Database Design	62
Chapter 5	Normalization and Design Review	82
Chapter 6	Physical Design	102
Chapter 7	SQL	125
Chapter 8	Is It Secure?	157
Appendix A	Using Microsoft Access with the Book	178
Appendix B	SQL Server Express	185
Appendix C	Visio	188
Appendix D	Common Relational Patterns	191

Glossary 195

Index 199

This page intentionally left blank

CONTENTS

Preface ix

Chapter 1 WHO NEEDS A DATABASE 1

Overview of Relational Databases and Their Uses	1
The Situation	1
The Opportunity	5
Getting the Scope	7
The First Interview	8
Identifying the Big Topics	10
Writing the Statement of Work	11
Reviewing the Statement of Work	13
The Statement of Work	14
Documentation	15
<i>Things We Have Done</i>	16 • <i>Vocabulary</i> 16
<i>Things to Look Up</i>	17 • <i>Practices</i> 17 • <i>Scenarios</i> 17

Chapter 2 GATHERING INFORMATION 20

Interviews, Observations, and Reviewing Documents	20
Looking at the Documents	20
Cloud Databases	28
Preparing for the Interview	29
The Interview	30
The Questionnaire	32
Tutoring Services Questionnaire	32
Tutors at Work	33
Documentation	36
<i>Things We Have Done</i>	36 • <i>Vocabulary</i> 36
<i>Things to Look Up</i>	36 • <i>Practices</i> 36 • <i>Scenarios</i> 37

Chapter 3 REQUIREMENTS AND BUSINESS RULES 46

Getting Started	46
Review of the Issues	47
Requirements	49
Business Rules	52
Review of Requirements and Business Rules with Terry	54
A Little Bit of Grammar	54
Entities and Attributes	57
Candidate Keys	58
Documentation	59
<i>Things We Have Done</i>	59 • <i>Things to Look Up</i> 59
<i>Vocabulary</i>	60 • <i>Practices</i> 60 • <i>Scenarios</i> 60

- Chapter 4 DATABASE DESIGN 62**
 - Entity Relation Diagrams 62
 - Designing the Database 62
 - Documentation 78
 - Things We Have Done* 78 • *Vocabulary* 78
 - Things to Look Up* 78 • *Practices* 78 • *Scenarios* 79

- Chapter 5 NORMALIZATION AND DESIGN REVIEW 82**
 - The Design Review 82
 - Final Content Review 98
 - Documentation 99
 - Things We Have Done* 99 • *Vocabulary* 99
 - Things to Look Up* 100 • *Practices* 100 • *Scenarios* 100

- Chapter 6 PHYSICAL DESIGN 102**
 - Choosing the Management System 102
 - Creating the Database 104
 - Documentation 121
 - Things We Have Done* 121 • *Vocabulary* 121
 - Things to Look Up* 121 • *Practices* 122 • *Scenarios* 123

- Chapter 7 SQL 125**
 - Running Queries 125
 - Testing the Database 131
 - Joins 141
 - Inserts, Updates, and Deletes 145
 - Creating a Trigger 147
 - Documentation 153
 - Things We Have Done* 153 • *SQL Keywords* 154
 - Things to Look Up* 154 • *Vocabulary* 155
 - Practices* 155 • *Scenarios* • 155

- Chapter 8 IS IT SECURE? 157**
 - The Issue 157
 - Where to Start 157
 - Analyzing Security Needs 160
 - Threats 163
 - Finding Solutions 166
 - Documentation 173
 - Things We Have Done* 174 • *Things to Look Up* 174
 - Table of Additional SQL Key words 174
 - Vocabulary* 175 • *Practices* 175 • *Scenarios* 175

- Appendix A: Using Microsoft Access with the Book 178**
- Appendix B: SQL Server Express 185**
- Appendix C: Visio 188**
- Appendix D: Common Relational Patterns 191**

PREFACE

Many students taking an introductory database course need hands-on experience. Typically, they are under pressure to finish quickly with a certificate or degree and get to work. They need to get actual practice in the process of designing and developing databases that they can apply in their future employment. They need to create tables, enter data, and run SQL queries.

This book is designed for them.

Hands-on Database: An Introduction to Database Design and Development focuses on the process of creating a database. It guides the students through the initial conception of the database. It covers gathering of requirements and business rules, the logical and physical design, and the testing of the database. It does this through a continuous narrative that follows a student, Sharon, as she designs and constructs a database to track the tutoring program at her school. It shows some of her missteps as well as her successes. Students get hands-on experience by doing practices and developing scenarios that parallel the narrative.

After completing this book, students will have a good sense of what is involved in developing and creating a database. Following is a list of the book outcomes. A student who has completed this book will be able to

- give a general definition of a relational database
- identify a variety of ways to gather database requirements
- define business rules for a database
- create an entity design for a database
- normalize a design up to Third Normal Form
- develop a database in a given DBMS
- run SQL queries against sample data to test requirements and business rules
- define the general security context of a database and its users
- document the process of database design and development

WHAT'S NEW FOR THE SECOND EDITION

For the second edition of this book, I have corrected any small errors that I found in the first edition. In addition, the following things are new:

- A fifth scenario: **Show Times: Local Shows and Acts**. This scenario gives the students the opportunity to see another type of database, one that deals with schedules and involves the interactions between several disparate stakeholders, specifically between artists, venues for the shows, and fans.
- All screenshots and examples were updated to reflect the most current editions of Microsoft SQL Server, Visio, and Office.
- In Chapter 2, a brief discussion of cloud computing, cloud databases, and cloud data services has been added.
- Also in Chapter 2, a note was added about the interview process and the forms for the WestLake Hospital scenario were simplified and clarified.
- In Chapter 5, "Practices" questions have been totally redone to provide a better demonstration of each stage of normalization.
- In Chapter 7, a new section entitled **Advanced SQL**, which includes discussion of sub queries, UNION, finding and removing duplicated rows, and the use of indexes, has been added.
- Chapter 8 now includes a brief discussion of Big Data and some of its implications for database.

THE SCENARIO APPROACH

The scenario approach is at the heart of the book. It informs both the narrative and the exercises. A scenario in its essence is a story problem. It provides a context from which to work. It is much easier for a student to understand database design if he or she sees it as a solution to a particular set of problems. There is an emphasis on defining business rules and then testing the database design against those rules. The scenarios also provide a sense of process. They give the student some guidance in how to go about defining and developing a database. I would argue that even computer science students could benefit from this approach. It would allow them to experience how the concepts they have learned can be applied to the actual development process.

The scenario that makes up the body of the book describes Sharon, a database student, in the process of creating a database to manage the school's tutoring program. She encounters several problems. The way the tutoring sessions are scheduled is awkward and inefficient. The reports that the manager of the program needs to make are difficult and time consuming to put together. It is also difficult, at times, to track the tutors' hours. Sharon sees a database as a solution to these problems and sets about defining its requirements, designing it, and building a prototype. She enters some sample data and then tests the database using SQL to enter and retrieve the information required. Finally, she looks carefully at the security issues inherent in the database.

At the end of each chapter, after the Practices section, there are five additional scenarios for the student to develop. The **Wild Wood Apartments** scenario involves creating a database to manage a chain of apartment buildings. **Vince's Vintage Vinyl Record Shop** offers a scenario of a small shop owner who needs a database to handle his inventory, sales, and purchases. **Grandfield College** leads students through the process of making a database to track what software the school owns, the licensing for that software, on what machines the software is installed, and which users have access to those machines. The **WestLake Research Hospital** scenario involves creating a database to track a double-blind drug study for a new antidepressant. The **Show Times: Local Shows and Acts** scenario has students creating a database to track local music involving shows, artists, venues, and clients who can be informed when their favorite artists are appearing.

The scenarios are meant to be complex enough to keep the student involved but simple enough not to overwhelm the novice. Each scenario presents different challenges. Students could work on some or all the scenarios, or they could be broken into groups with each group assigned one of the scenarios. The scenarios are open ended, that is, they offer room for student creativity and innovation. The students and the instructor are free to define many of the parameters and business rules as they proceed. But each scenario, in each chapter, has specific deliverables that help keep the students on track.

OTHER FEATURES

Process Driven

The book models the process of developing a database from the beginning through the final stages. It provides students with tools and techniques for discovering requirements and business rules. It also provides them with suggestions for organizing and managing all the complex details that go into developing a database. The book emphasizes the need to understand the data and the relationships among the data. It shows them the value of carefully designing a database before actually implementing it. Then, when the database is first developed, it emphasizes the need to test it, to make sure it meets the requirements and business rules before deploying the database. Finally, it emphasizes the need to secure a database against both accidental and intentional threats.

Normalization

Normalization is an important but complex issue in database development. Anyone who works with databases is expected to have some knowledge of normalization.

For this reason, I believe it is important to introduce the students to the concepts and vocabulary of normalization. But, because this is an introductory book focused on the process of development and design, I have discussed only the first three normal forms. I have found that most databases that achieve at least the Third Normal Form are functional, if not optimal, in design. That being said, I do believe anyone working in databases should become familiar with all the normal forms and principles of normalizations. In the “Things to Look Up” section of Chapter 4, I direct students to look up the other normal forms and pick one of them to explain to other students. Also, in Appendix D, “Common Relational Patterns,” the last example shows an ERD of a database that has been normalized beyond Third Normal Form.

SQL

Chapter 7 in *Hands-on Database* contains an extensive introduction to SQL. It covers SELECT statements, of course, using a variety of criteria, as well as using scalar functions, especially date and time functions, and various aggregate functions. Inner and outer joins are discussed. INSERT, UPDATE, and DELETE statements are introduced. The chapter also illustrates the use of Views and provides an example of a stored procedure and a trigger. Chapter 8 looks at stored procedures in terms of how they can be used to protect data integrity and security. SQL commands related to Logins and permissions are also introduced.

Perhaps more important than the specific SQL commands presented is the context in which they are introduced. In the text, Sharon uses SQL to test the requirements and business rules of the Tutor Management database. In the scenarios, students use SQL to test the requirements and business rules of the databases they have created. In Chapter 8, they see SQL as a tool for securing a database. By presenting it in this way, students see SQL as a vital part of database development and not just an academic exercise.

Security

Security issues are discussed at several points in the book. It is brought into consideration during the information-gathering phases in Chapters 2 and 3. But it is dealt with in detail in Chapter 8.

Chapter 8 attempts to show the student a structured approach to security. It looks at each user of the database and creates a table that delineates exactly what permissions that user needs on each object in the database. It applies a similar technique for analyzing threats to the database. Then it introduces the concept of roles as collections of permission. It shows how a developer could create an application layer of views and procedures and then assign roles and permissions to those objects rather than to the underlying tables.

Finally, the chapter discusses the importance of disaster management and of creating a set of policies and procedures for recovering from any conceivable disaster.

Software Used by the Book

The book uses Microsoft SQL Express 2012 for the database and Microsoft Visio 2012 for the database diagramming. The SQL Express software is offered free from Microsoft. At the time of writing this introduction, SQL Express is available at <http://www.microsoft.com/express/Database/>. This is, of course, subject to change. But one can always go to the Microsoft site and type SQL Server Express in the Bing search box. This will list the current download URL.

I selected SQL Server Express because it is readily available and because it provides a more realistic and complete database management system experience than Microsoft Access, which is often used in classroom settings. SQL Server Express lets the students experience managing multiple databases in a single management environment. The SQL Express Management Studio also contains a query analyzer that allows students to easily run SQL queries and view the results. Unlike Access, SQL Server Express supports stored procedures and triggers. Finally, again unlike Access, SQL Express provides a rich set of security features that are more typical of commercial

database management systems. If, however, an instructor prefers or must use Microsoft Access, Appendix A explains how to substitute it for SQL Server. The appendix notes the variations in practices and examples in each chapter required for the adaptation.

Other database software such as MySQL or Oracle could also be adopted for use with the book. Although the book uses SQL Server Express, its focus is on the process of developing and designing a database. The principles of this process are applicable to any DBMS.

Microsoft Visio is readily available to students for schools that belong to the Microsoft Developers Network Academic Alliance (MSDNAA). It can also be purchased at a significant discount from places like the Academic Superstore and other academic outlets. Visio offers a range of tools and templates that help make diagramming and modifying diagrams easy and enjoyable for students. Appendix C offers additional instruction in how to use the Database Model template in Visio 2010. Of course, other modeling software could be easily substituted, or students could be asked to simply draw their models on graph paper. What is important are the concepts, not the particular tools.

CHAPTER CONVENTIONS

Each chapter contains several elements other than the narrative about Sharon. These elements are meant to provide greater depth and to provoke the student to think about some of the broader implications of the material.

Things You Should Know

These extended sections provide background and descriptions of various aspects of database development and design. In many ways, they function like the more traditional textbook. They provide definitions, explanations, and examples that provide a deeper, more comprehensive context to the things that Sharon is doing in the narrative.

Things to Think About

These are sidebars that invite the student to consider questions about the processes or topics under discussion. The questions in these sections do not have definite answers. They are meant to encourage thought and discussion.

Cautions

Cautions are found in the margins of the text. Their purpose is to warn the students about potential mistakes or common errors.

Documentation

This section is found at the end of each chapter. It provides a summary of how a student would go about documenting the activities conducted during the chapter.

Things to Look up

This section is also found at the end of each chapter. It guides students to other resources and topics not fully covered in the book.

Vocabulary

Vocabulary is an important part of any discipline. Anyone who wants to work in the database field will be expected to know and understand certain terms.

Vocabulary words are highlighted in margins and are repeated in an exercise at the end of each chapter where the student is asked to match the word with the definition. SQL terms are listed in tables at the ends of Chapters 6 and 8. The terms are also defined in the Glossary at the end of the book.

Practices

Practices are found at the end of each chapter. They are designed to give each student hands-on experience with the materials of the chapter. Most practices are self-contained, but some do build on each other. In particular, the practices for Chapter 5 and 6 are related. In Chapter 5, the students build a Pizza database, and in Chapter 6, they query that database with SQL.

Scenarios

As mentioned earlier, Scenarios are the life of the book. There are five scenarios which students build on throughout the book. Their purpose is to provide students with the full experience of developing a database, from identifying the initial concept to testing the fully built database. For students, the most effective use of these scenarios would be to follow one or more of the scenarios throughout the entire term.

OUTLINE

The book contains eight chapters, four appendixes, and a glossary. It is meant to be just long enough to be covered fully in a single term. Following is an outline of the book with a summary of each chapter's narrative and a list of the outcomes for that chapter.

Chapter 1: Who Needs a Database

NARRATIVE Sharon, a student at a community college, applies to become a tutor for database-related subjects at the school. She discovers they use spiral notebooks and spreadsheets to manage the tutoring information. She suggests to the supervisor that they could benefit from a database and offers to build it. The supervisor agrees to the project. Sharon interviews her and gets a sense of what the overall database will entail and drafts a statement of scope. She and the supervisor discuss the statement and make some modifications.

OUTCOMES

- Define relational databases
- Understand the position of relational databases in the history of databases
- Identify major relational database management systems
- Identify main characteristics of relational databases
- Understand SQL's role in relational database
- Recognize some indications of where a database could be useful
- Define a statement of scope for a given database scenario

Chapter 2: Gathering Information

NARRATIVE Now that she has the scope of the database, Sharon begins to gather information about the data the database will need to capture and process. First, she looks at the spiral notebooks that have been used to schedule tutoring sessions. She also looks at the spreadsheets the supervisor develops for reports and other related documents. Then she arranges an interview with several of the tutors and an additional interview with the supervisor, and creates a questionnaire for students who use the tutoring services. Finally, she spends an afternoon in the computer lab, observing how students schedule tutoring and how the actual tutoring sessions go.

OUTCOMES

- Review documents to discover relevant entities and attributes for database
- Prepare interview questions and follow up
- Prepare questionnaires
- Observe work flow for process and exceptions

Chapter 3: Requirements and Business Rules

NARRATIVE Having gathered all this information, Sharon must figure out what to do with it. She searches through her notes for nouns and lists them. Then she looks at the lists to see if there are additional topics, or subjects. Then she groups which nouns go with which topics. For each topic area, Sharon identifies some candidate keys. Next, she looks through her notes to determine what the business rules of the tutoring program are. She lists the rules and makes notes for further questions. The rules seem complex, and Sharon remembers something from a systems analysis class about UML diagrams called Use Case diagrams. She uses these diagrams to graphically show how each actor—tutor, student, and supervisor—interacts with the database.

OUTCOMES

- Use nouns from notes and observations to discover database elements
- Group elements into entities and attributes
- Define business rules
- Develop Use Case diagrams to model requirements

Chapter 4: Database Design

NARRATIVE Sharon is ready to design the database. She looks at her topics lists and diagrams an initial set of entities, using Visio. She analyses the relationships among the entities, adding linking tables wherever she finds a many-to-many relation. Then she adds the other items from her list to the appropriate entities as attributes. For each attribute, she assigns a data type. She reviews the design to ensure that she has captured all the data and the business rules.

OUTCOMES

- Use the database modeling template in Microsoft Visio
- Create entities and add attributes
- Determine the appropriate relationship between entities
- Resolve many-to-many relationships with a linking table

Chapter 5: Normalization and Design Review

NARRATIVE Now, with the help of an instructor, Sharon checks to make sure the database conforms to the rules of normalization. She reviews the database thus far with her supervisor.

OUTCOMES

- Evaluate entities against first three normal forms
- Adjust the relational diagram to reflect normalization

Chapter 6: Physical Design

NARRATIVE Sharon builds a prototype of the database, creating all the tables and setting up the relationships. When she has it set up, she enters 5 or 10 rows of sample data so she can test the database.

OUTCOMES

- Implement a physical design of the database based on the logical ERDs
- Choose appropriate data types for columns
- Enter sample data into tables

Chapter 7: SQL

NARRATIVE Sharon writes some SQL queries to see if she can get the needed information out of the database. She tests for database requirements.

OUTCOMES

- Name the main events in the development of SQL
- Run SELECT queries with a variety of criteria
- Join two or more tables in a query
- Use the aggregate functions COUNT, AVG, SUM, MIN, and MAX
- INSERT, UPDATE, and DELETE records
- Use SQL to test business rules

Chapter 8: Is it Secure?

NARRATIVE In this chapter, Sharon looks at the security needs of the database. It is important to give everyone the access that they require to do the things they need to do. But it is also important to protect the database objects and data from either accidental or intentional damage. Sharon discovers that security is complex and requires careful planning.

OUTCOMES

- Analyze security needs and restrictions for users of the database
- Analyze threats to database integrity
- Understand the concepts of authentication and authorization
- Create logins and users
- Create roles

Appendixes

A: USING MICROSOFT ACCESS WITH THE BOOK A quick overview of using Microsoft Access instead of SQL Server with the book. It looks at each chapter and shows how you would use Access and what adjustments you will need to make to the practices and scenarios.

B: SQL SERVER EXPRESS An overview of how to use the SQL Server Management Studio to create and access databases in SQL Server Express.

C: VISIO An overview of the Visio environment, with a special focus on the database templates.

D: COMMON RELATIONAL PATTERNS A review of some of the most common relational patterns students will encounter in database design such as the Master/Detail relation, weak entities, linking tables, and so on.

GLOSSARY OF TERMS Glossary of all vocabulary terms.

SUPPLEMENTS

The following online resources are available to adopting instructors at www.pearson-highered.com/irc:

Instructor's Manual—It contains a chapter outline and answers to all end-of-chapter questions for each chapter of the text.

PowerPoint Presentations—These feature lecture notes that highlight key text terms and concepts. Professors can customize the presentation by adding their own slides or by editing the existing ones.

Test Item File—An extensive set of multiple choice, true/false, and essay-type questions for each chapter of the text. Questions are ranked according to difficulty level and referenced with page numbers from the text. The Test Item file is available in Microsoft Word format and as the computerized Prentice Hall TestGen software, with WebCT, Blackboard, Angel, D2L, and Moodle-ready conversions.

TestGen—A comprehensive suite of tools for testing and assessment. It allows instructors to easily create and distribute tests for their courses, either by printing and

distributing through traditional methods or by online delivery via a local area network (LAN) server. TestGen features Screen Wizards to assist you as you move through the program, and the software is backed with full technical support.

Image Library—A collection of the text art organized by chapter. This collection includes all of the figures, tables, and screenshots from the book. These images can be used to enhance class lectures and PowerPoint slides.

CourseSmart eTextbooks Online—**CourseSmart** (www.coursesmart.com) is an exciting new *choice* for students looking to save money. As an alternative to purchasing the print textbook, students can purchase an electronic version of the same content and save up to 50% off the suggested list price of the print text. With a CourseSmart etextbook, students can search the text, make notes online, print out reading assignments that incorporate lecture notes, and bookmark important passages for later review.

ACKNOWLEDGMENTS

I would first of all like to acknowledge my patient and enthusiastic students who worked through draft versions of this text and provided invaluable feedback. I would also like to thank Pearson and especially Bob Horan and Kelly Loftus, who provided support, encouragement, and advice throughout the lengthy process of completing this book. I also could not have written the book without the careful and diligent feedback from the reviewers:

Georgia Brown, *Northern Illinois University*
 Geoffrey D. Decker, *Northern Illinois University*
 George Federman, *Santa Barbara City College*
 Bob Folden, *Texas A&M University*
 Jean Hendrix, *University of Arkansas at Monticello*
 Stephen L. Hussey, *St. Louis University*
 Chunming Gao, *Michigan Technological University*
 David Law, *Alfred State College*
 Seongbae Lim, *St. Mary's University*
 Louis Mazzucco, *State University of New York at Cobleskill*
 Tina Ostrander, *Highline Community College*
 Michele Parrish, *Durham Technical Community College*
 Adonica Randall, *Alverno College*
 Ann Rovetto, *Horry-Georgetown Technical College*
 Richard Scudder, *University of Denver*
 Elliot B. Sloane, *Villanova University*
 Lee Tangedahl, *University of Montana*
 Annette Walker, *Craven Community College*
 Loraine Watt, *Mitchell Community College*

Finally, I would like to acknowledge my family, who showed enormous patience with the hours I spent at my computer.

ABOUT THE AUTHOR

When he first started working on his English degree, a professor told Steve Conger that an English major can be used in a variety of ways. His subsequent career proved that. After graduation, he worked for over a year in the Coeur d'Alene Idaho school district, assisting children with learning disabilities. Then, for six years he worked for the U.S. Forest Service as a surveyor's assistant, while going to graduate school in the off-seasons. After graduating, he moved to western Washington, where he worked as a nurse's aide until he was hired to teach at Seattle Central Community College. As a part-time instructor who owned a computer, he realized early that he could teach more sections and earn more money teaching computer classes than he could teaching English composition. Despite this varied career path, Steve has never regretted his English degree or given up his love of writing.

Steve Conger has taught at Seattle Central Community College for over twenty years. He helped design the current successful Information Technology Program, and for the last several years, he has taught database and programming courses using Microsoft SQL Server and .Net programming languages. For several years, he has been a board member for the statewide Working Connections workshops, which offer affordable IT training to college instructors. Currently, Working Connections is sponsored by Bellevue College's Center for Excellence.

Steve Conger has a master's degree in English from the University of Idaho and a bachelor's degree in Literary Studies from Gonzaga University.

Currently, he lives in Eatonville, Washington, with his wife and two children. His two other children live in the area and have kindly provided him and his wife with three grandchildren.

This page intentionally left blank

Who Needs a Database

OVERVIEW OF RELATIONAL DATABASES AND THEIR USES

This chapter introduces Sharon, a college student who is working toward a degree in Database Development and Administration. She signs up to become a tutor and realizes that the tutoring program is in desperate need of a database to track tutoring sessions. She volunteers to develop it, and after some discussions defines a statement of work for the database.

CHAPTER OUTCOMES

By the end of this chapter, you will be able to:

- Define relational databases
- Understand the position of relational databases in the history of databases
- Identify major relational database management systems
- Identify main characteristics of relational databases
- Understand SQL's role in relational database
- Recognize some indications of where a database could be useful
- Define a statement of work for a given database scenario

THE SITUATION

Sharon is a student taking database classes. She is near the end of her program and has done quite well. Like any student, she could really use some extra money and has decided to inquire about tutoring. She has noticed that many students seem to struggle with relational database concepts, particularly in the early classes, and she is fairly sure there would be a demand for her services.

The administrator of the tutoring program at the college is named Terry Lee. Terry invites Sharon into her office and offers her a seat. She smiles.

"So you want to tutor?"

"Yes. I think I would be good at it."

"What subjects do you think you could tutor?"

"I was thinking especially of database-related topics. I can do relational design and SQL. I think I can tutor Microsoft Access, SQL Server, and even other database management systems. I can also do some database programming."

Terry nods. "That's good. We do have some requests for tutoring in those areas, but so far no one to provide the tutoring. Before you can begin, you will need to get recommendations from two instructors who teach in the area you want to tutor. Also, you will need to do a short training session."

Sharon smiles, "That's no problem."

"Good." Terry rises from her seat. "Let me show you how things work."

RELATIONAL DATABASE

A type of database that stores data in tables that are related to each other by means of repeated columns called keys.

RELATIONAL DESIGN

It involves organizing data into tables or entities and then determining the relationships among them. **SQL** is the language relational databases used to create their objects and to modify and retrieve data.

Table 1-1 Equipment Checkout

Member ID	Member Name	Date	Time	Equipment No.
23455	Nancy Martin	2/10/2013	4 PM	2333
45737	Taylor Smith	2/10/2013	4:15 PM	3331
23455	Nancy Martin	2/10/2013	4:45 PM	2221



Things You Should Know

DELIMITED FILES

These files have some sort of character separating columns of data. The delimiter is often a comma or tab, but can be any non-alphanumeric character. In **fixed length** files, the length in characters of each column is the same.

DATA INTEGRITY

It refers to the accuracy and the correctness of the data in the database.

REDUNDANCY

It refers to storing the same data in more than one place in the database.

Databases

A database, at its simplest level, is a collection of related data. It doesn't have to be electronic. The card catalogs that libraries used to have were certainly databases. A scientist's spiral notebook where he or she keeps notes and observations could be considered a database, so too could a phone or address book. When we say "database," though, we usually mean electronic databases, databases that run on computers.

Flat File Databases

The simplest form of an electronic database is the flat file database. Flat files usually consist of a file that stores data in a structured way. A common format for flat file databases is the delimited file. In a delimited file, each piece of data is separated from the next piece by some "delimiter," often a comma or a tab. The end of a row is marked by the new-line character (usually invisible). It is important, if the file is to be read correctly, that each row contain the same number of delimiters. Another kind of flat data file is the fixed-width data file. In such files, all the columns share a fixed width in characters. These flat files can be read by a computer program and manipulated in various ways, but they have almost no protections for data integrity, and they often contain many redundant elements.

Redundancy refers to repeating the same data more than once. It can occur in a number of ways. Data could be repeated over and over again in the same file. For instance, the following example shows an equipment checkout list.

Notice how Nancy Martin's name is repeated, and it would be repeated as many times as she checks out equipment. Another type of redundancy occurs when the same data are stored in different files. For instance, you might have a file of club members that stores Nancy's name and address, and then a separate file for fee payments that repeats her name and address. One problem with this system is that, other than having to type in everything several times, each time you reenter the same data, there is a greater chance of mistyping it or making a mistake of some kind. Another problem occurs when you need to change her address. Say Nancy moves and she notifies the person at the desk in the club about her change of address. The desk clerk changes the address in the membership file, but fails to change it, or to notify someone in billing to change it, in the fee payment file. Now when the club sends out a bill or statement of fees to Nancy, it goes to the wrong address. It is always best to enter each piece of data in one and only one place.

Spreadsheets, such as Excel, can also be used as flat file databases. Spreadsheets offer a great deal more functionality than simple delimited files. Cells can be given a data type such as "numeric" or "date time." This helps ensure that all the entries in a given column are of the same type. You can also define valid ranges for data (e.g., you can stipulate that a valid term grade is between the numbers 0 and 4). Spreadsheets usually contain data tools that make it possible to sort and group data. Most spreadsheets also contain functions that allow the user to query the data. But despite these enhancements, spreadsheets still share many of the redundancy and data integrity problems of other flat file formats.

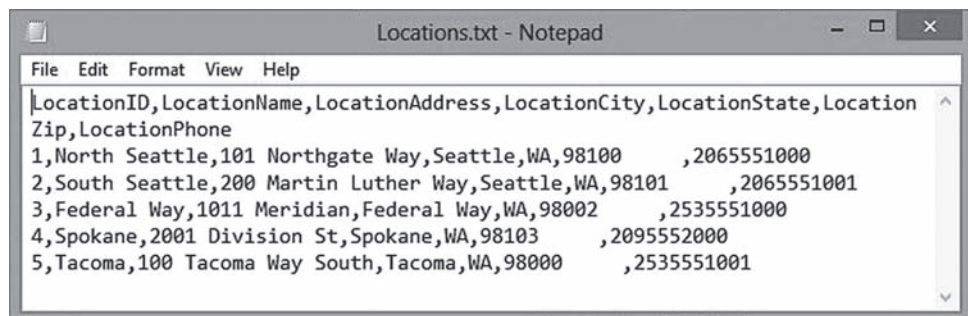


FIGURE 1-1 Delimited Text Example

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G
3	Date	Name	Apartment	LeaseNumber	Amount paid	Late	
4	7/1/2009	Martin Scheller	203	#203011208	\$ 1,200.00		
5	7/1/2009	Roberta Louise	311	#311060108	\$ 1,400.00		
6	7/1/2009	Sue Tam	111		\$ 1,400.00		
7	7/1/2009	Laura Henderson	207	#207020209	\$ 1,350.00		
8	7/1/2009	Thomas Jones	110	#110010109	\$ 1,200.00		
9	7/2/2009	Shannon Hall	205	#205010109	\$ 1,350.00		
10	7/2/2009	Bob Newton	104	#104030209	\$ 1,250.00		
11	7/9/2009	Dennis Smith	209		\$ 1,400.00	X	
12							
13							

FIGURE 1-2 Excel Spreadsheet

Hierarchical Databases

The most common database model before the relational model was the hierarchical database. Hierarchical databases are organized in a tree-like structure. In such a database, one parent table can have many child tables, but no child table can have more than one parent.

This sounds abstract, and it is. One way to visualize it is to think of the Windows (or, for that matter, the Mac or Linux) file system. The file system has a hierarchical structure. You have a directory, under which there can be subdirectories, and in those subdirectories, there can be other subdirectories or files. You navigate through them by following a *path*.

```
C:\Users\ITStudent\Documents\myfile.txt
```

This tree-like organization is very logical and easy to navigate but it does present some of the same problems of redundancy, data integrity, and comparability of data. It is not uncommon for the same data to be repeated in more than one place in the tree. Whenever a data is repeated, there is a risk of error and inconsistency. It can also be very difficult to compare a piece of data from one branch of the database with a piece from an entirely different branch of the database.

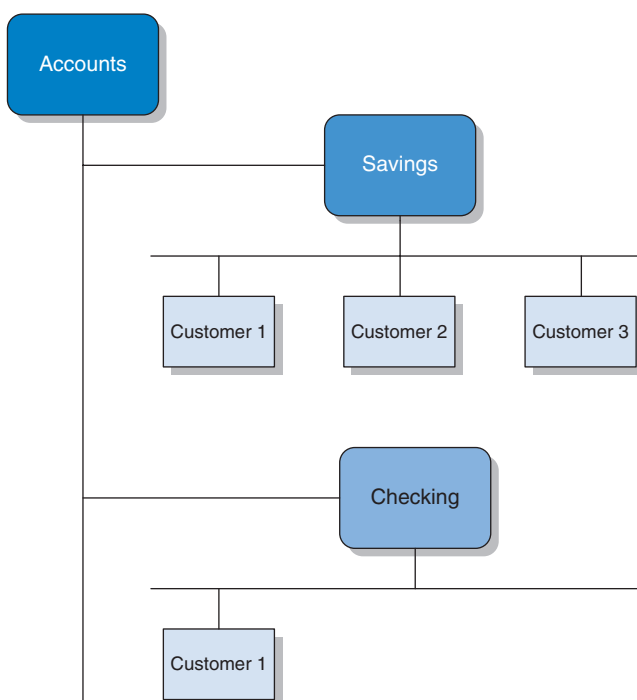


FIGURE 1-3 Hierarchical Database Model

THINGS TO THINK ABOUT

Hierarchical databases are still in use in many institutions. This is especially true of large institutions such as banks and insurance companies that adopted database technologies early.

These institutions invested heavily in the development of these databases and have committed decades of data to their files. Although database technologies have improved, they are reluctant to commit the time and money and to incur the risk of redeveloping their databases

and translating their vast stores of existing data into new formats.

The basic philosophy is, if it still works, let well enough alone. Most companies are conservative about their databases, for understandable reasons.

What do you think companies like Microsoft or Oracle have to do to convince companies to upgrade to their newest database products?

Relational Databases

By far, the most popular type of database for at least the last 30 years is the *relational database*. The idea for relational databases came from a man named Edgar F. Codd in 1970. He worked for IBM, and he wrote a paper on, at that time, a new theoretical design for databases. This design would be based on the mathematics of set theory and predicate logic. He formulated the basics of the relational design in 12 rules. The first rule, called the “information rule,” states, “All information in a relational database is represented explicitly at the logical level and in exactly one way—values in tables.”

Briefly, in the relational model data would be organized into tables. Even the information about the tables themselves is stored in tables. These tables then define the relationships among themselves by means of repeating an attribute or column from one table in another table. These repeating columns would be called “keys.” He also specified that the logical design of a database should be separate and independent of physical design considerations such as file types, data storage, and disk writing and reading functions. He specified that there should be a “data sublanguage” that can perform all data-related tasks. SQL has evolved into this language. We will discuss it more thoroughly in a later chapter. For a discussion of Codd’s 12 rules, see Wikipedia at http://en.wikipedia.org/wiki/Codd's_12_rules

This may sound complex, and it certainly can be, but it solved many of the problems that plagued the databases of the day. One of those problems was data redundancy. As mentioned earlier, redundancy refers to the need to store the same data in more than one place in the database.

In a relational database, the redundancy is minimized. A bank would enter the customer’s data only once, in one place. Any changes would be made only in one place. The only redundancy allowed is the repetition of a key column (or columns) that is used to create relationships among the tables. This significantly reduces the chances of error and protects the integrity of the data in the database.

Another problem the relational design helped solve was that of relating data from different parts of the database. In many of the previous database designs, a programmer had to write a routine in a language like Fortran or Cobol to extract the data from various parts of the database and compare them. In a well-designed relational database, every piece of data can be compared or joined with any other piece of data. The relational design was a huge step forward in flexibility.

The chief drawback of a relational database is the inherent complexity of the design. It is fairly easy to design a bad database that will not do what a client wants it to do. In a bad database design you may find that you cannot enter the data you need to enter. This is often the result of an error in how the relationship was created. It may also not be possible to retrieve the data that you need. Because of the complexity of relational design, it is crucial that you follow a design process that clarifies both the nature of the data you wish to store and the structure of the database. That is what this book is designed to help you with.

The chief advantages for a well-designed relational database are data integrity and flexibility. These two advantages have made it the most commonly used database model for the past 30 years or so.

KEYS

In relational databases, each table usually has one column designated as a **primary key**. This key uniquely identifies each row in the table. This primary key becomes a **foreign key** when it is repeated in another table to create a link between the tables.

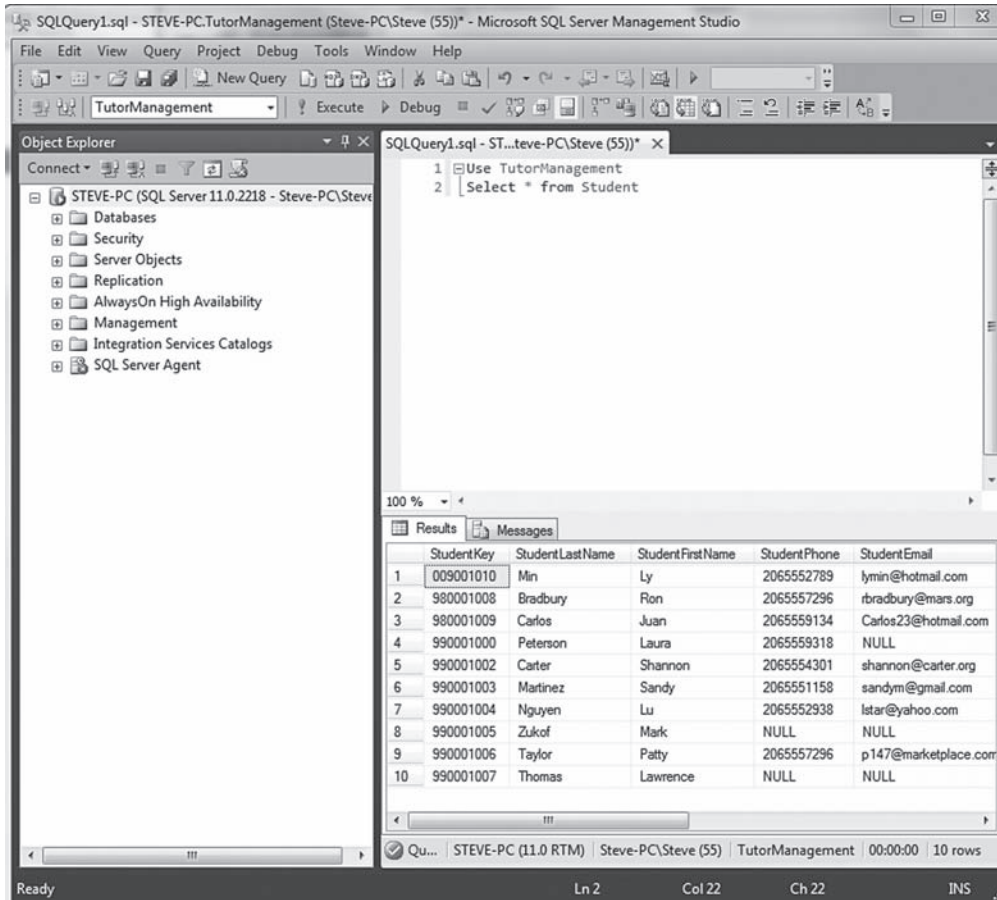


FIGURE 1-4 SQL Server Relational Database Manager Showing an SQL Query and Results

Customer ID(PK)	Last Name	First Name	Address	City	State
C41098X3	Carson	Lewis	121 Center Street	Seattle	WA
CV1099B1	Madison	Sarah	1324 Broadway	Seattle	WA
D345XU24	Brown	Lisa	2201 Second Ave	Seattle	WA

Transaction ID	Transaction Type	Transaction Date	Customer ID(FK)	Amount
10002345	Deposit	2009-2-12 10:25:06	C41098X3	1245.76
10002346	Deposit	2009-2-12 10:27:13	CV1099B1	500.00
10002347	Withdrawal	2009-2-13 14:45:57	C41098X3	200.00

FIGURE 1-5 Primary Key-Foreign Key Relations Between a Customer Table and a Transaction Table

THE OPPORTUNITY

They walk from Terry’s office down the hall to the computer lab. Terry stops at the front desk. “The computer lab is one of our designated tutoring areas, and I suspect the one where most of your sessions would be scheduled.” She picks up a clipboard containing several pieces of paper. “We have 2 pages for each week, an AM one and a PM one. At the beginning of the month, tutors enter their availability for each day, what times they are available that day, and the courses they can tutor for. Students sign up

for particular sessions. Tutoring is free for the students as long as they are enrolled in the class for which they are getting tutored.”

“How do you check that?”

“Right now, it is mostly a matter of trust.”

“How long is each tutoring session?”

“Tutoring sessions are for 30 minutes each, and a tutor can only do 30 sessions or 15 hours a week.”

“What if you set up a time slot and nobody signs up?”

“As long as you show up when scheduled, we will pay you for the time. The pay, by the way, is \$10.50 an hour.”

“Thanks.” Sharon looks over the notebook. “Just out of curiosity, what do you do with the schedules at the end of the month?”

“Actually, I take them back to my office every 2 weeks and type them into various spreadsheets to make reports to the people who pay for the tutoring, and to determine the pay for the tutors themselves.”

Sharon turns to Terry and says, “You know, you could really use a database. It would make it much simpler to track schedules and availability, and it could make doing your reports much easier.”

Terry sighs. “I’ve known that for some time, but we just can’t find anyone willing to take on the task. The school’s database administrator is much too busy and no one else feels competent or has the time to take on the task.”

Sharon hesitates a little and then says, “I might be able to put a database together.”

Terry looks hopeful. “Really? That would be wonderful. We even have some money in our budget so we could pay you something for your work.”

“I am still learning database,” Sharon cautions, “but I am pretty sure I could make you something that would meet most of your needs.”

“Good, why don’t you come by tomorrow about this time and we will talk about it.”

“I will be there.”

THINGS TO THINK ABOUT

There are many situations that could be improved with the addition of a database. Whenever there is a large amount of complex data to handle, a database is likely to provide the best solution.

There are times, however, when the data involved is modest in scope and complexity that a

relational database may be an overkill. Relational databases are complex to develop and maintain.

The benefits when dealing with large amounts of data are worth the costs in development time and maintenance. But, sometimes, the best solution is simply a spreadsheet such as Excel.



Things You Should Know

RDBMS

A relational database management system (RDBMS) is, as its name suggests, a system for managing relational databases. As a minimum, an RDBMS needs to allow a user to create one or more databases and the objects associated with that database such as tables, relationships, views, and queries. It also needs to support basic maintenance such as backing up the database and restoring it from a backup file. Moreover, it needs to support security, making sure that users and groups have access only to the databases and data that they are authorized to use.

Most commercial RDBMSs offer many features beyond these basic ones. Most include tools for monitoring and optimizing the performance of their databases. Many include reporting services to format and present the results of queries. Some even include complex business intelligence packages for analyzing business trends and patterns. Following is a table of the most common RDBMSs with a link to their home Web sites.

Table 1-2 Common Relational Database Management Systems

RDBMS	Comments	URL
ORACLE	The first and the biggest commercial RDMS. Powers many of the world's largest companies	http://www.Oracle.com
SQL Server	Microsoft's RDMS product. Ships in many versions designed for different company needs. Also powers many large enterprises	http://www.microsoft.com/sql/default.msp
DB2	IBMs RDBMS	http://www306.ibm.com/software/data/db2/9/
MySQL	The most popular open source RDBMS currently owned by SUN	http://www.MySql.com
PostGres SQL	Another free, open source RDBMS. It is older and some would say more powerful than MySQL	http://www.postgresql.org/
ACCESS	Microsoft's desktop database	http://office.microsoft.com/en-us/access/default.aspx?ofcresset=1

GETTING THE SCOPE

After Sharon leaves Terry's office, she goes to one of the instructors, a professor named Bill Collins, from whom she hopes to get a recommendation. He is sitting in his office and smiles when he opens the door for her. "Come in. How can I help you today?" Sharon tells Collins about her plan to tutor and asks for a recommendation. Collins says he will be happy to provide one. Then Sharon tells him about the possibility of making a database.

She says, "I've got a thousand ideas about how the database should look and what should be in it."

Bill cautions her, "Be careful not to get ahead of yourself. You need to remember you are not making this database for yourself. You are making it for a client. You need to listen carefully to what Terry and the other people who will use the database say about what they need and not get trapped by preconceived notions. The first thing you need to do is get as clear an idea as possible about what the database is intended to do."

"A statement of scope?"

"Yes, that would be a good place to start, but I would go farther and make a complete statement of work. That would include the scope, but it would also contain some discussion of the background, the objectives of the project, and a tentative timeline. I have some samples I can share with you. Listen, if you need any help or advice on this project, feel free to ask me."

"Thank you. Thank you very much."



Caution

It is easy to get carried away with your own excitement about a database project. You may be able to see several possible solutions and want to start designing right away. But it is critically important that you delay designing until you have a clear idea of what the client wants and needs. Patience and the ability to listen are among the most important skills of a database developer.

STATEMENT OF SCOPE

A **statement of scope** is a short statement of one or more paragraphs that says in clear, but general, terms what the project will do. A **statement of work** is a more complete statement about the objectives and timeline of the project.

Things You Should Know

Statement of Work

A statement of work is a preliminary document that describes, in general, the work that needs to be done on a project. Often this is prepared by the people who want the work to be done and offered to contractors for bidding. But sometimes, as in this case, it can be used as an initial clarification of task at hand.

It is important to have something like a statement of work for any major project so that everyone knows what is expected. Without it, people often find, sometimes late in the process, that different individuals have very different expectations about what the project should contain. A statement of work is also a good reference throughout the project to keep everyone on track and focused. The statement is preliminary and can be altered as the needs of the project change or grow. But, by referring to the statement of work, you can guarantee that any changes or additions are a matter of discussion and not just assumed by one of the parties.

The following table delineates a few of the elements that can appear in a statement of work.

Table 1-3 Statement of Work Elements

Element	Description
History	Describes the reason for the project, usually a problem with the current system or an opportunity to provide new services. May describe the various steps and efforts that led to the current state of the project
Scope	Provides a general statement of the requirements and expectations of the project. It states only the high-level requirements and does not get into specifics. It does not go into detail about how things are to be done. It may include some general constraints such as time or budget limits
Objectives	The things the project is intended to achieve. Objectives aren't about creating specific elements of the database, for instance, but about what the database is supposed to achieve, that is, why the client wants the database in the first place
Tasks and Deliverables	Breaks the project into discrete tasks. Each task should have an estimated duration and concrete deliverables

THE FIRST INTERVIEW

The next day Sharon sits in Terry's office. She has brought a notebook in which she has written down some of the key questions she knows she will need to ask. Sharon knows it is important to be prepared and focused for any interview. She has also brought a diagram of a database she created for a nonprofit to show Terry as an example of work she has done on database creation.

Terry says, "Thanks for coming in. You have no idea how long and how much we've wanted a database for the tutoring program. We have to generate several reports each term to justify our funding. It has gotten so that creating reports takes most of my time. It keeps us from doing things to improve the program. We also really need to be able to track what works and what doesn't better."

Sharon nods, "I really hope I can help. I've brought an example of a database I made for Capital Charities to show that I do have some experience creating databases. We did this as part of a project for a Database class"

Terry looks at the diagram as Sharon explains it.

"Capital Charities provides funds for basic utilities, food, and occasional repairs for poor families on a onetime, emergency basis. They needed to be able to track their contributors and their contributions. That was one part of the database. That data is stored in the contributor and contribution tables. That line between them indicates a one-to-many relationship. It uses what is called "crow's feet" notation. It shows that each contributor has contributed at least once and may have contributed many times. The crow's foot, those three lines, points to the many sides of the relationship. The other part of the database tracks the types and amounts of assistance given to each client. The client information is entered into the Client table."

She points to the ClientNotes entity, "There can be 0 or many notes about any client. Each client receives assistance at least once. That was a business rule of the charity. They wanted to list as clients only those they had actually given assistance to. Each act of assistance is associated with a particular councilor and can involve several different types of assistance. That is the reason for the AssistanceDetail table."

"It looks complex."

"It is a little. But I also built some forms and reports that made it such that the Capital Charities staff didn't have to navigate the database directly. It made it a lot easier to use."

"Well, it certainly looks like you should be capable of doing this for us. What do you need from me?"

"You have already started suggesting some of the things I want to talk about today—things you want the database to do. What I need to get from you today is a clear sense of what you want the database to do for you. I don't need the specifics yet, just

CROW'S FEET NOTATION

A type of entity relation diagram where the relationships are depicted using lines and 0s. These are more descriptive of relationships than the diagrams using simple arrows.

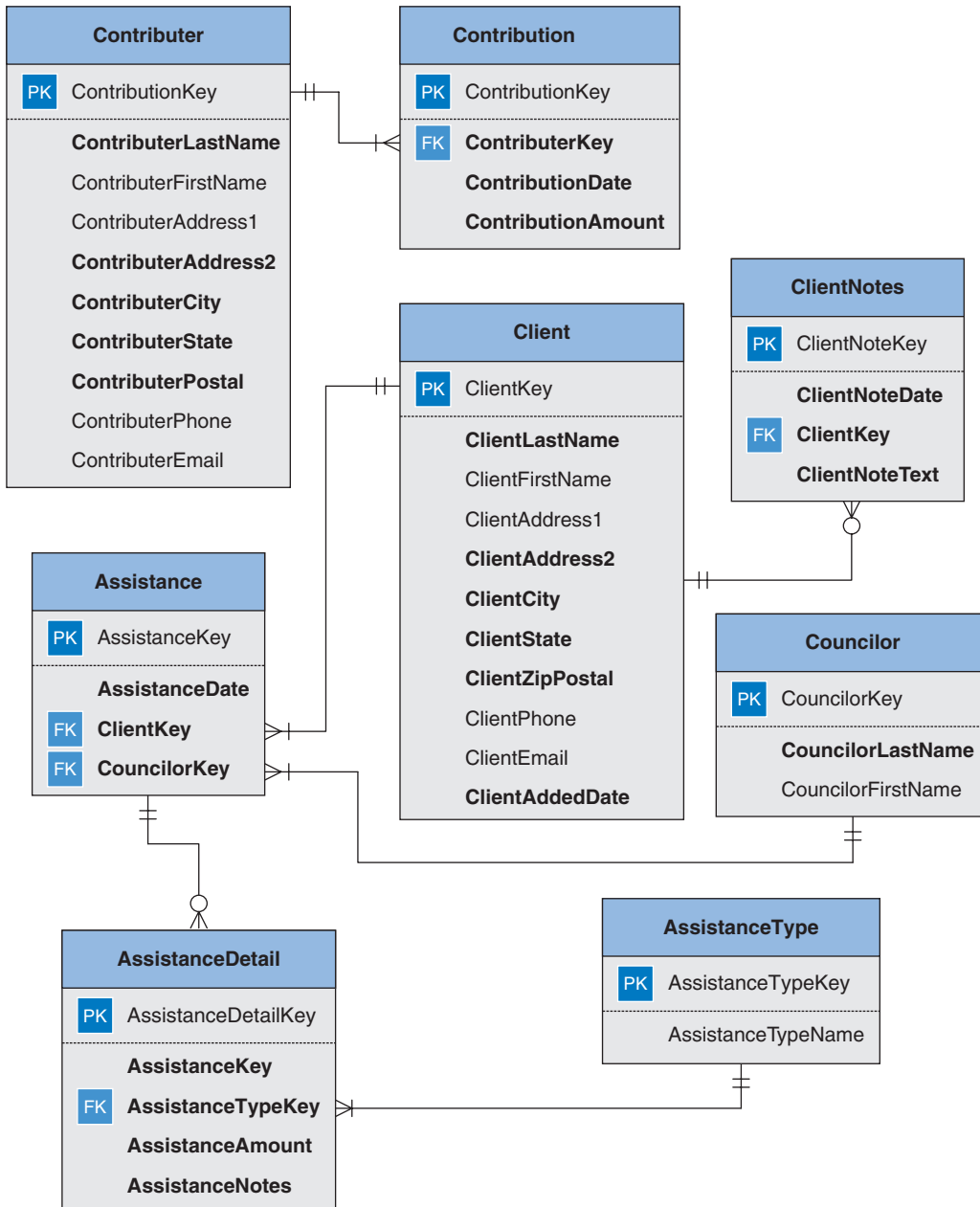


FIGURE 1-6 Sample Entity Diagram for a Nonprofit

general statements of what you want to see and what the database needs to do to be useful to you.”

Terry hesitates, “OK... Where do I start?”

“You already suggested a couple of things. You need to track what works and what doesn’t. How would you determine that something is working or not working?”

Things You Should Know

You should always go to an interview prepared. In this initial interview, you should be prepared to help the client get started on the right track and have questions that help focus them on the important aspects of the database. But you don’t want to guide them toward some preconceived notion of what the database should be. Rather, your questions should help them guide you to a clearer understanding of what they need out of a database.