Database Management Systems

Designing and Building Business Applications

GERALD V. POST

Database Management Systems

Designing and Building Business Applications

GERALD V. POST
Western Kentucky University

Irwin McGraw-Hill

Boston Burr Ridge, IL Dubuque, IA Madison, WI New York
San Francisco St. Louis Bangkok Bogotá Caracas Lisbon
London Madrid Mexico City Milan New Delhi Seoul
Singapore Sydney Taipei Toronto

Irwin/McGraw-Hill

A Division of The ***McGraw-Hill*** *Companies*

This book is printed on acid-free paper.

3 4 5 6 7 8 9 DOW/DOW 9 3 2 1 0

ISBN 0-07-289893-3

Vice president and editorial director: *Michael W. Junior*
Senior sponsoring editor: *Rick Williamson*
Developmental editor: *Christine Wright*
Marketing manager: *Jodi Fazio*
Project manager: *Margaret Rathke*
Production supervisor: *Scott M. Hamilton*
Designer: *Kiera Cunningham*
Supplement coordinator: *Rose M. Range*
Compositor: *York Graphic Services, Inc.*
Typeface: *10/12 Garamond Book*
Printer: *R. R. Donnelley & Sons Company*

Library of Congress Cataloging-in-Publication Data

Post, Gerald V.
Database management systems : designing and building business applications / Gerald V. Post.
p. cm.
Includes index.
ISBN 0-07-289893-3
1. Database management. 2. Database design. 3. Business—Databases. 4. Management information systems. I. Title.
QA76.9.D3P675 1999
005.74—dc21 98-21687

www.mhhe.com

Preface

A TALE OF TWO WEB SITES

The Orinoco Music Company is proud of their Web site. The graphics are cool, the audio clips are hot, and initial excitement in the press has brought record numbers of potential customers to the site. Orders are coming in through the Web order form. After a few weeks, some problems arise. Clerks are making mistakes in copying the orders from the Web form into the company's existing mail order system. Customers are canceling orders because many of the shipments are backordered—they are complaining that if they had known the item was not in stock, they would never have ordered it. After a couple of months, the press begins to downgrade the site—noting that the graphics are old and they cannot get audio clips for the new bands. Because of the expense of constantly changing the site, Orinoco Music is thinking about removing the site and returning to a basic company-information site.

Customers have been flocking to the new Web site for Salt Peanuts Music Company. In the month since the site was activated, orders have almost doubled—in terms of the number of customers and in the value of each order. Reviews of the site have noted the limited use of graphics, but they rave about the service provided. In particular, customers can constantly see if an item is in stock. With a couple of clicks, they get background information on any artist and can play short clips from any song. Once customers place an order they can use the UPS tracking system to see when it will arrive. Customers can also call sales representatives who have instant access to all of the customer data. But everyone's favorite feature is that the system tracks individual purchases and suggests similar groups. These selections are based partly on expert opinions, but are primarily driven by grouping sales. Customers can see what products are bought by groups of similar customers. Everyone is happy with the system. Company managers like it because it increases sales. Customers like it because they have instant access to the information they want. Recording artists like it because it gives everyone access to the music and increases sales.

What, you might ask, does this tale of two Web sites have to do with database management systems? The difference between the two Web sites is that the Salt Peanuts Music Company's site is built on a database management system that integrates the company's data and enables them to create a more complete, interactive site. This text will teach students what they need to know about database management systems and how they can be used to solve similar business problems.

INTRODUCTION

While databases are often created and maintained by information technology professionals, more often in today's businesses management professionals in all disciplines are designing and creating their own database applications. Virtually every area of management uses databases: marketing professionals to analyze sales data, human resource managers to evaluate employees, operations managers to track and improve quality, accountants to integrate data across the enterprise, and financial analysts to analyze a firm's performance. That is why it is so important for business students to understand how database management systems are used to design, build, and run a modern database application.

This text is targeted at the primary business database course at the junior level. Students from any business major should understand the material. Although the text favors the database management approach over traditional programming techniques, students will find it easier if they have taken an introductory programming course. This text supports the learning process through clear exposition, many examples, exercises, and sample databases. Although some students might learn how to build an application through a general lecture or discussion, most require examples and hands-on practice supported by comments from a knowledgeable instructor.

GOALS AND PHILOSOPHY

The goal of this text is straightforward: At the end of the text, students should be able to evaluate a business situation and build a database application. The text focuses on the use of relational database technology for building applications as it is what students will encounter in businesses today.

Before they begin building relational databases, students need to master three specific areas of knowledge: database design, SQL, and programming. Databases must be carefully designed to gain the strengths of the DBMS approach. That is why the heart of this text focuses on the two topics crucial to building successful databases: database design (normalization) and SQL (queries). These two topics—standardized across all major database systems—must be covered carefully and thoroughly, particularly because they can be difficult for students.

Database design is the foundation for building applications. A well-designed database can simplify building, maintaining, and expanding an application. An important strength of relational database design is its flexibility. A properly designed database can be expanded to meet changing business conditions. On the other hand, if the design is weak, building an application will be substantially harder and more time-consuming. It is often better to throw away a poorly designed database and start over, than to try fixing or expanding it.

SQL is a powerful, standard query language that is used for virtually every step of application development. One of its greatest strengths is its availability in many different products. Once students learn the foundations of SQL, they will be able to retrieve data from almost any major database system. Many queries in SQL are relatively simple, so the foundations can be learned rapidly. Yet SQL can also be used to answer complex questions.

Another area of knowledge students need to build solid business applications is programming skills. Some applications and some database systems require detailed programming skills. However, in many cases, programming is used sparingly. It can be used as a glue to combine various components or add new features that make the application easier to use.

Most applications experience trade-offs among database design, SQL, and programming. The weaker the design, the less you rely on SQL and the more programming you will need to build the application. Because programming code is more likely to create errors and is harder to change, application developers should rely on proper database design and the power of SQL.

Database designs and queries are relatively standardized across hardware and software platforms, but details and application development depend on the specific DBMS. Microsoft Access has been chosen as the platform for demonstrating application development in this text because of its market dominance, similarity to other packages, availability, and ease of use.

DATABASE DESIGN AND THE UNIFIED MODELING LANGUAGE

For several years, entity-relationship diagrams have been the predominant modeling technique for database design. However, this approach causes problems for instructors (and students) because there are several different diagramming techniques. This text helps solve these problems by incorporating the Unified Modeling Language (UML) method, instead of traditional entity-relationship (ER) diagramming, as the modeling technique for database design. This change will be most apparent in the replacement of the ER diagram notation and terminology with the parallel concepts in UML class diagrams.

UML class diagrams, although very similar to ER diagrams, are superior in several ways. First, they are standardized, so students (and instructors) need learn only one set of notations. Second, they are "cleaner" in the sense that they are easier to read without the bubbles and cryptic notations of traditional ER diagrams. Third, they provide an introduction to object-oriented design, so students will be better prepared for future development issues. Fourth, with the rapid adoption of UML as a standard design methodology, students will be better prepared to move into future jobs. In late 1997, many of the systems design organizations adopted UML as a standard method for designing systems. UML has the support of major authors in systems design (e.g., Booch, Rumbaugh, and Jacobsen) as well as being supported by the major software development firms like IBM, Microsoft, Oracle, and Sterling. In addition, students should have little difficulty transferring their knowledge of the UML method if they need to work with older ER methods.

The basic similarities between ER and class diagrams are (1) entities (classes) are drawn as boxes, (2) binary relationships (associations) are drawn as connecting lines, and (3) n-ary associations (relationships) are drawn as diamonds. Hence the overall structures are similar. The main differences occur in the details. In UML the multiplicity of an association is shown as simple numerical notation instead of a cryptic icon. An example is shown in the following two figures.

UML also has provisions for n-ary associations and allows associations to be defined as classes. There are provisions for naming all associations, including directional names to assist in reading the diagram. Several situations have defined icons for the association ends, such as composition (rarely handled by ER), and subtypes (poorly handled by ER).

More details of the UML approach are shown in Chapters 2 and 3. Only a small fraction of the UML diagrams, notation, and terminology will be used in this text. You can find the full specification on the Web at http://www.rational.com/uml/.

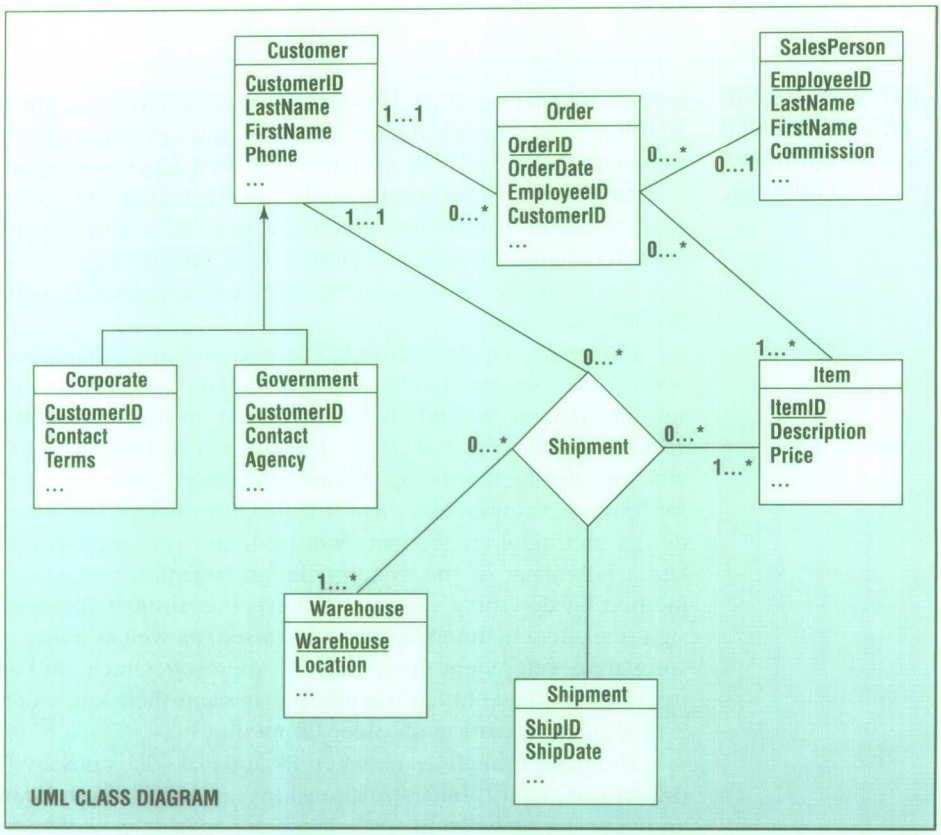
ORGANIZATION

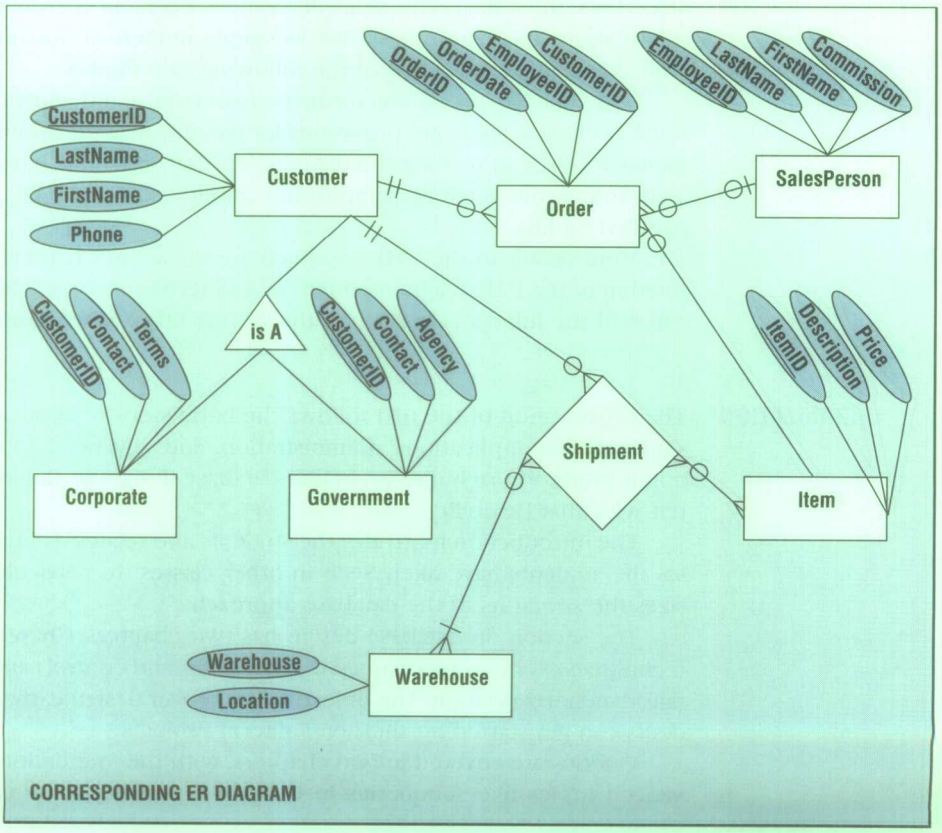
The organization of the text follows the basic steps of application development: design, queries, applications, administration, and advanced topics. Some instructors might prefer to teach queries before database design, so the initial chapters are written with that flexibility.

The introduction motivates the students and relates database applications to topics the students have likely seen in other classes. In particular, the chapter emphasizes the strengths of the database approach.

The section on database design has two chapters: Chapter 2 on general design techniques (systems techniques, diagramming, and control) and Chapter 3, which details data normalization. The objective is to cover design early in the term so that students can get started on their end-of-term projects.

Queries are covered in two chapters, with the foundations in Chapter 4 and advanced topics like subqueries in Chapter 5. Although the two chapters could be


Customer
CustomerID
LastName
FirstName
Phone
...
Order
OrderID
OrderDate
EmployeeID
CustomerID
...
SalesPerson
EmployeeID
LastName
FirstName
Commission
...
Corporate
CustomerID
Contact
Terms
...
Government
CustomerID
Contact
Agency
...
Item
ItemID
Description
Price
...
Shipment
Warehouse
Warehouse
Location
...
Shipment
ShipID
ShipDate
...
1...1
0...*
0...1
1...*
UML CLASS DIAGRAM


CustomerID
LastName
FirstName
Phone
Customer
OrderID
OrderDate
EmployeeID
CustomerID
Order
EmployeeID
LastName
FirstName
Commission
SalesPerson
is A
CustomerID
Contact
Terms
Corporate
CustomerID
Contact
Agency
Government
Shipment
ItemID
Description
Price
Item
Warehouse
Location
Warehouse
CORRESPONDING ER DIAGRAM

Chapter 1: Introduction
Part 1: Systems Design
Chapter 2: Database Design
Chapter 3: Data Normalization
Part 2: Queries
Chapter 4: Data Queries
Chapter 5: Advanced Queries and Subqueries
Appendix: SQL Syntax
Part 3: Applications
Chapter 6: Forms and Reports
Chapter 7: Calculations and Data Manipulation
Appendix: Introduction to Programming
Chapter 8: Application Development
Appendix: Database Applications in Visual Basic
Part 4: Database Administration
Chapter 9: Physical Design and Performance
Chapter 10: Database Administration
Part 5: Distribution and Integration of Data
Chapter 11: Distributed Databases and The Internet
Appendix: Introduction to Oracle PL/SQL
Chapter 12: Object-Oriented Databases and Integrated Applications
Appendix: Database Projects

combined, the split provides for a second set of end-of-chapter material (particularly exercises).

Part 3 describes the development of database applications, beginning with the essentials of form and report development in Chapter 6. Chapter 7 provides a foundation in writing programs in a database environment. It focuses on Microsoft Access, but the general concepts can easily be applied to other systems, such as PowerBuilder. Chapter 8 discusses application-finishing concepts like transactions, menus, toolbars, and help files.

Part 4 examines various topics in database administration. Chapter 9 investigates aspects of physical design and database performance including indexes, clustering, and choice of storage methods. It focuses on understanding database options to improve application performance. Chapter 10 examines management issues emphasizing planning, implementation, and security.

Part 5 expands applications into larger solutions. Chapter 11 covers several issues involving distributed databases and discusses developing databases for use over the Internet. It describes how to integrate a database with Microsoft Windows NT Active Server Pages. Chapter 12 introduces the topic of integrating data objects across applications, such as using a database in a groupware environment to share spreadsheets or driving a word processing report system with a database application program. Chapter 12 also describes the use of object-oriented databases and SQL3.

Additionally, three chapters have appendixes that discuss programming concepts that are more technical. The appendix to Chapter 7 introduces programming using Microsoft's Visual Basic for Applications, which underlies Microsoft Access.

The appendix to Chapter 8 briefly describes how to create database applications in the stand-alone product Microsoft Visual Basic. The appendix to Chapter 11 introduces programming concepts in Oracle's PL/SQL, which is similar to the proposals for SQL3.

PEDAGOGY

The educational goal of the text is straightforward and emphasized in every chapter: By the end of the text, students should be able to build business applications using a DBMS. The text uses examples to apply the concepts described in the text. Students should be encouraged to apply the knowledge from each chapter by solving the exercises and working on their final projects.

Each chapter contains several sections to assist in understanding the material and in applying it to the design and creation of business applications:

- **What You Will Learn in This Chapter.** A series of questions that highlight the important issues.
- **Overview.** A student's perspective of the chapter contents.
- **Chapter Summary.** A brief review of the chapter topics.
- **A Developer's View.** A short summary of how the material in the chapter applies to building applications.
- **Key Words.** A list of words introduced in the chapter. A full glossary is provided at the end of the text.
- **Additional Reading.** References for more detailed investigation of the topics.
- **Web Site References.** Some sites that provide detailed information on the topic. Some are newsgroups where developers share questions and tips.
- **Review Questions.** Designed as a study guide for the exams.
- **Exercises.** Problems that apply the concepts presented in the chapter. Most require the use of a DBMS.
- **Projects.** Several longer projects are presented in an appendix at the end of the text. They are suitable for an end-of-term project.
- **Sample Databases.** Two sample databases are provided to illustrate the concepts. Sally's Pet Store illustrates a database in the early design stages, whereas Rolling Thunder Bicycles presents a more finished application, complete with realistic data. Exercises for both databases are provided in the chapters and called out with icons.

FEATURES OF THE TEXT

1. Focus on modern business application development.
   - Database design explained in terms of business modeling.
   - Application hands-on emphasis with many examples and exercises.
   - Emphasis on modern graphical user interface applications.
   - Chapters on database programming and application development.
   - Appendixes on programming and development details.
2. Hot topics.
   - Description and use of the unified modeling language (UML) for modeling and system diagrams. This new standard will soon be required for all designers.

- In-depth discussion of security topics in a database environment.
- Development of databases for the Internet and intranets.
- Emphasis on SQL 92, with an introduction to SQL3.
- Integrated applications and objects in databases.

3. Development examples in Microsoft Access and Visual Basic.
4. Applied business exercises and cases.
   - Many database design problems.
   - Exercises covering all aspects of application development.
   - Sample cases suitable for end-of-term projects.
5. A complete sample database application (Rolling Thunder Bicycles).
   - Fully functional business database.
   - Sample data and data generator routines.
   - Program code to illustrate common database operations.
6. A second database (Sally's Pet Store) for comparison and additional assignments.
7. Lecture notes as PowerPoint slide show.

END-OF-TERM PROJECTS

Several projects are described in the appendix at the end of the text. These cases are suitable for end-of-term projects. Students should be able to build a complete application in one term. The grading focus should be on the final project. However, the instructor should evaluate at least two intermediate stages: (1) a list of the normalized tables collected shortly after Chapter 3 is completed and (2) a design preview consisting of at least two major forms and two reports collected shortly after Chapter 6.

Some instructors may choose to assign the projects as group assignments. However, it is often wiser to avoid this approach and require individual work. The project is a key learning tool. If some members of the group avoid working on the project, they will lose an important learning opportunity.

INSTRUCTIONAL SUPPORT

An Instructor CD-ROM is available to adopters and contains the following:

- A test bank prepared by G.W. Willis of Baylor University with multiple choice, short answer questions, and short projects is available for use with the Irwin/McGraw-Hill electronic test bank software.
- Lecture notes and overheads are available as slide shows in Microsoft PowerPoint format. The slides contain all the figures and additional notes. The slides are organized into lectures and can be rearranged to suit individual preferences.
- Several databases and exercises are available. The instructor can add new data, modify the exercises, or use them to expand the discussion in the text.
- Sally's Pet Store database application is provided in Microsoft Access format and is used extensively in the text to illustrate topics. The Pet Store example is in an earlier stage of design than Rolling Thunder for two main reasons. (1) Students can compare the applications and gain insight

into the development process. (2) Students can be given assignments to provide additional features to the Pet Store application.

- The Rolling Thunder database application is available in Microsoft Access format. It is a self-contained application that illustrates many of the concepts and enables students to examine many aspects of a complete database application, including the code that drives the application.
- An Internet site for direct contact with the author: http://www.mhhe.com/business/mis/post.
- An Internet site for contact with the publisher: http://www.mhhe.com.

SUPPLEMENTAL BOOKS

The purpose of this book is to show students how to design and build business applications. To illustrate the concepts, several examples and applications use existing DBMS. Most examples are based on Microsoft Access; a few use Oracle and Microsoft SQL Server. However, to learn to use these tools in-depth, students may wish to read a supplementary book that explains the detailed features of a particular database system.

Irwin/McGraw-Hill has many supplemental books that can be bundled with this book as a package. Books are available to help students learn the details of Access, Oracle, and Visual Basic.

ACKNOWLEDGMENTS

Creating a new approach to teaching database management required the efforts and support of many people. The database class can be a difficult course to teach but one of the most enjoyable. It requires considerable dedication by instructors to develop methods to teach the material. The dedication of the reviewers who shared their time and expertise to improve this book is greatly appreciated. They are: Susan Athey, Colorado State University; Gerald C. Canfield, University of Maryland; Connie W. Crook, University of North Carolina-Charlotte; Tom Farrell, Dakota State University; John W. Gudenas, Aurora University; Thomas Hilton, Utah State University; Chris Jones, University of Washington; Bhushan Kapoor, California State University-Fullerton; John C. Malley, University of Central Arkansas; Bruce McLaren, Indiana State University; Rajesh Mirani, University of Baltimore; Hsueh-Chi Joshua Shih, National Yunlin University; and G. W. Willis, Baylor University.

Feedback from students has refined the teaching methodologies and improved the text. I am particularly indebted to the "students" at the 1997 National Computer Educators Institute. Their dedication, camaraderie, and professionalism are an inspiration to all instructors. Their input led to several improvements in the text. Thanks to Lou Adelson, Clarion University; Kellie Keeling, University of North Texas; Alyson Livingston, Tarrant County Jr. College; Scott Lord, Bainbridge College; Donald Musselman, James Madison University; Beth Paver, Grand Rapids Community College; Doug Stoddard, Clovis Community College; and Ben White, Bainbridge College.

It is always a pleasure to work with the staff at Irwin/McGraw-Hill. Rick Williamson's guidance and support were crucial to producing this text. Christine Wright's dedication and humor kept the project on track. The entire staff made my job much easier and more enjoyable. I am also grateful to G. W. Willis of Baylor University for assistance in preparing the instructor supplements.

Brief Contents

Contents