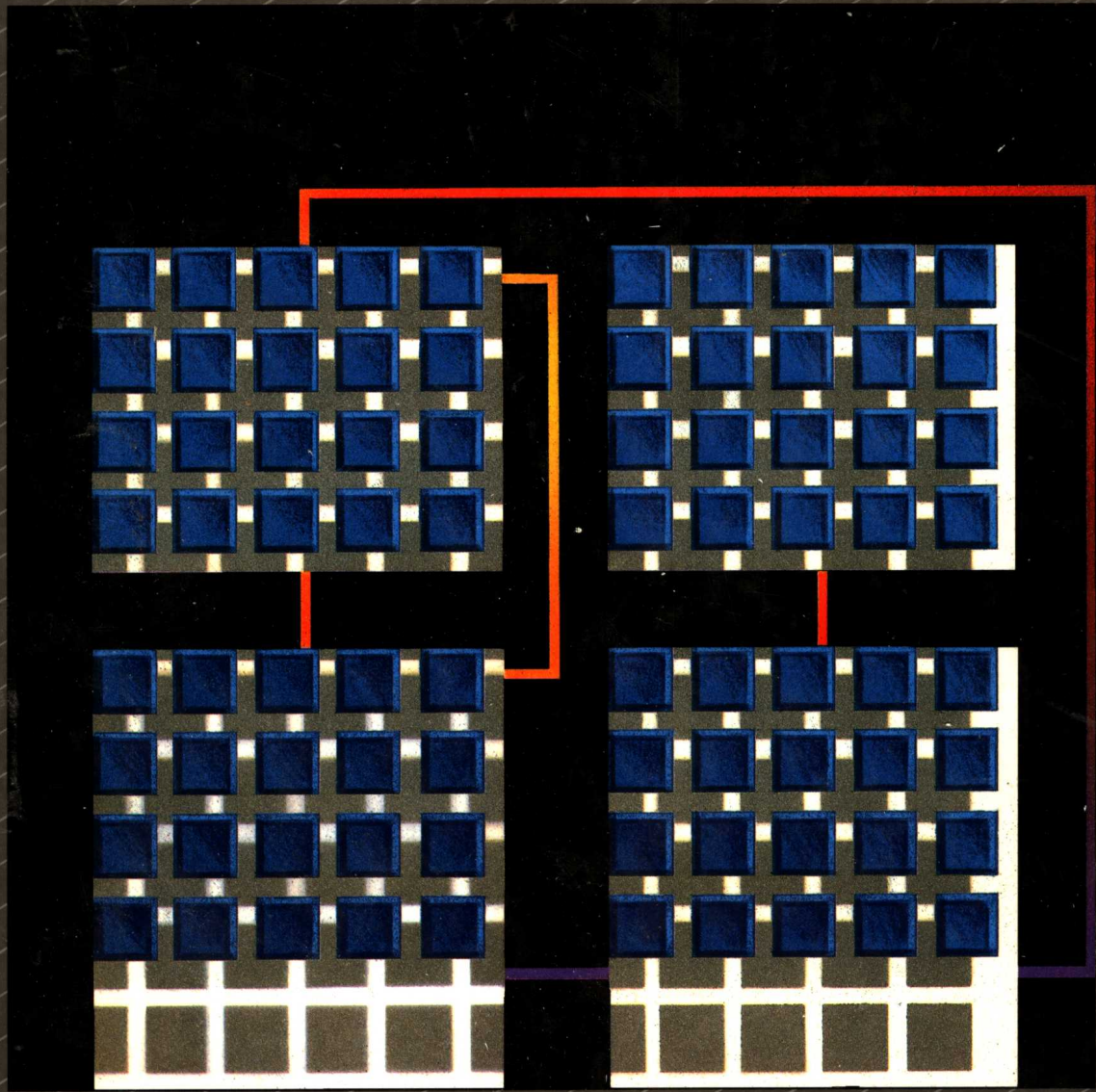


Database Management: Theory and Application



JOHN C. SHEPHERD



Database Management

*Theory and
Application*

John C. Shepherd
Duquesne University

IRWIN

Homewood, IL 60430
Boston, MA 02116

We recognize that certain terms in this book are trademarks, and we have made every effort to print these throughout the text with the capitalization and punctuation used by the holder of the trademark.

© RICHARD D. IRWIN, INC., 1990

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Senior sponsoring editor: *Lawrence E. Alexander*

Developmental editor: *Rebecca J. Johnson*

Project editor: *Waivah Clement*

Production manager: *Carma W. Fazio*

Designer: *Image House*

Artist: *Benoit Design*

Compositor: *J.M. Post Graphics, Corp.*

Typeface: *10/12 Palatino*

Printer: *R.R. Donnelley & Sons Company*

Library of Congress Cataloging-in-Publication Data

Shepherd, John C.

Database management : theory and application / John C. Shepherd.

p. cm.

ISBN 0-256-07829-7

1. Data base management. I. Title

QA76.9.D3S534 1990

005.74—dc20

8927933

CIP

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0 DO 7 6 5 4 3 2 1 0

P R E F A C E

INTENDED AUDIENCE FOR THIS BOOK

Database Management: Theory and Application is appropriate for an applied programmer/analyst-oriented course in database management. It is recommended that students have had at least one COBOL course, as this is the language used in the examples. It is also beneficial if students have already developed sequential and random access programs, which provide a basis for comparing the traditional file-processing methods with the database approach.

This book is best suited to those courses emphasizing database application programming from an applied but thorough approach. It covers each of the three data models (relational, network, and hierarchical) in great detail and describes the theoretical basis for each.

For those courses that take a more managerial, database administration approach, Parts I, II, and VI, which deal with database and design concepts, will be of most benefit. Parts III, IV, and V discuss the data models, beginning with the theoretical foundation of each model, and contain numerous programming examples.

PURPOSE OF THIS BOOK

Today's database environment is changing. For the first time, there is a preferred database methodology and language: the relational approach and Structured Query Language (SQL). However, as recently as early 1989, there were more than three times as many commercial sales of network and hierarchical database packages. Some current texts ignore these models or cover them in just a single chapter. Thus, students may not learn how to develop applications with the network and hierarchical database products used by more than 15,000 organizations.

Using many years of experience as a DBA and many more as a programmer using a database management system (DBMS), I provide intensive coverage of actual commercial packages so that students can use the text as a reference after graduation. For example, instead of merely saying that a program should check for error codes after invoking the DBMS, I provide actual values to be checked. When I worked in this field prior to entering the teaching profession, no books on the market offered appropriate coverage of commercial packages. When faced with the task of developing a program that used a DBMS, students either had to read vendors' reference manuals or attend training sessions in order to gain

1715/5P/05

even a rudimentary knowledge about how to write such programs. This book attempts to fill that vacuum.

FEATURES

Specific features of this text include:

- *A carefully constructed foundation for each of the three data models.* Part I, which discusses the theory behind database management, builds a foundation for the three parts devoted to the data models. Chapter 3 on data structures is particularly complete. All of the database management systems discussed in this book use a data structure for their implementation. By studying data structures, students are better prepared to understand how the packages work, how they are implemented, and why they must be used in the manner described.
- *At least two chapters devoted to each model.* The first chapter opening each model-related part introduces the concepts behind the model and how to define the database. The second chapter shows how to manipulate the resulting database. The relational model will eventually be so dominant that current texts might be able to successfully get away with eliminating a discussion of the network and hierarchical models. The early-1990 reality, however, is that thousands of organizations are still using applications written to take advantage of database systems built on these two models. Parts IV and V provide thorough coverage of the CODASYL network and hierarchical models.
- *Pedagogical symmetry for each data model.* In addition to the traditional sections on data storage and retrieval, the chapters devoted to each model contain sections on data integrity, data backup and recovery, and transaction processing. This enables students to compare and contrast all three models.
- *Integrated microcomputer coverage.* The hierarchical and network parts contain an integrated section, while micro-based versions of relational packages are discussed in Chapter 14.
- *Skills-oriented approach.* This book includes more programming examples for each model than any current text. Whenever possible, I have tried to illustrate more than just a few lines of code. It is vital that students be able to use a DBMS after they graduate. Only by seeing many complete examples can this be accomplished.
- *Fourth-generation language coverage.* One of the database trends I have observed from my consulting work is the tendency of organizations and software developers to use SQL in concert with a fourth-generation language (4GL). Chapter 8 on embedded SQL and Chapter 14 on data efficiencies discuss how SQL can be used in such an environment. While most database applications are written in COBOL, the relational approach requires new languages to fully exploit its power.

- *Managerial coverage.* The human element in database management cannot be ignored. The roles of the database administrator (DBA), programmer, analyst, and end user are clearly defined throughout the text. For example, Part II is oriented toward the database administration function, and Part VI discusses where the database market is headed. Finally, each of the three data model–related parts includes thorough coverage of the model’s concepts in a program-free context.
- *Up-to-date information.* Writing a text about a topic as dynamic as database is quite a task. Over the course of a couple of years, a text goes through several drafts. In many subject areas, changes to texts from draft to draft are often aimed at correcting grammatical, organizational, and conceptual errors. This wasn’t the case with this book. Each new draft had to be updated to include new topics. For example, the DB2 coverage was updated for the final time less than one month after the most recent IBM version was released, and the discussions about database machines, distributed databases, and 4GLs reflect research and developments that were less than a month old when I wrote the last draft.
- *Chapter-opening cases.* Many chapter-opening cases illustrate what *not* to do or how the concepts discussed in the chapter could have been used to avoid the problem presented in the case. In this text, each case includes a series of discussion questions to help prepare students for the chapter that follows. Two of the cases are integrated throughout the text so that students can compare problem solutions using a package from each data model.
- *Chapter objectives.* After each opening case, the chapter proceeds with behavioral objectives that students should be able to achieve after reading the chapter. Very few texts in this market provide chapter objectives for students to test themselves.
- *An integrated case.* Based on over seven years of consulting in the hospital industry, I developed an integrated case study, Community Hospital, which demonstrates how to implement concepts discussed in the particular chapter. Several of the case episodes demonstrate a complete COBOL program that illustrates how a DBMS is used. By placing this material in a case format, I could illustrate all the “messy” details without getting bogged down in the chapters. The case approach also offered a unique opportunity to show how an application might look if programmed using a package from each data model.
- *Solved problems.* Most chapters contain numerous problems and accompanying solutions. This technique has proven an effective learning device, as it lets students see how to apply a concept to solve an actual problem. In many cases, the same problems are used in different chapters so that students can see how the problem is solved for each model.
- *Annotated bibliography.* The bibliography at the end of the text contains many references, several of them annotated.

ORGANIZATION

Part I introduces students to the database approach, data, data relationships, and data structures—the mechanisms for implementing data relationships. Because the structures are stored on disk drives, instructors might find it useful to supplement Part I with Appendixes A and B.

Part II covers the design of a database. Chapter 4 presents an approach for developing the conceptual design, a model of the organizations' entities and the relationships among them. The major thrust of this chapter is a discussion of normalization, a technique originally designed for the relational approach that is applied to all of the data models. This approach has seldom been taken, but it makes the chapter applicable regardless of the data model being covered in class. Chapter 5 shows students how to map the conceptual model to one compatible with the organization's DBMS and how to physically implement the transformed design using a data structure.

Part III covers the relational model, probably the most important model in this book. Chapter 6 discusses relational algebra. This method for manipulating relational databases has proven an effective way for students to grasp the concept of set-at-a-time versus the more conventional record-at-a-time processing to which they have become accustomed. This chapter also includes a highly comprehensive discussion of Query-By-Example, an IBM-developed approach that lets end users easily manipulate a relational database. Chapters 7 and 8 discuss Structured Query Language, first in an interactive mode and then embedded in COBOL and 4GL programs. Learning SQL before studying other database management languages can be somewhat misleading. What SQL can do in one statement, the network and hierarchical models would require more than twenty. It may be beneficial to assign the parts out of sequence, saving the easiest for last, so that students can better appreciate the power and benefits of the relational approach.

Parts IV and V discuss the network and hierarchical models, respectively. Both models had their origins in the 1960s and are significantly more complex than the relational approach. Part IV uses the industry standard DBMS developed by CODASYL as the basis for the discussion. The Community Hospital continuing case shows how Cullinet's IDMS/R package, the market leader for the network model, differs from the standard.

Part V discusses IBM's hierarchical package, IMS. As of early 1989, there were still more than twice as many IMS installations as relational ones, so these chapters are particularly important.

Part VI is devoted to management of the database. Chapter 13 discusses the duties and responsibilities of the database administrator (DBA), the individual usually responsible for database management. This chapter also returns to some topics addressed in Chapter 1: concurrency control and backup and recovery. Chapter 14 illustrates some of the current trends in database management, including how organizations are organizing their data for more efficient access.

By distributing the data to the end users, data communications costs as well as response time can be reduced. Another way to gain efficiency is to use a dedicated machine that performs the data access operations, a device usually referred to as a *database machine*. The final technique for improving the efficiency of data retrieval is the use of a 4GL. Two such packages are discussed in the chapter.

ACKNOWLEDGMENTS

I wish to acknowledge the assistance given to me by Cullinet and IBM, both of which provided me with reference materials and answered my questions relating to their database packages. I also want to thank ALCOA for assisting me with the IMS portion of the text. Finally, I wish to express my gratitude to the dedicated reviewers of this text, without whom this book would not have been possible:

James Benjamin
Southern Illinois University
at Edwardsville

Ralph Bisland
University of Southern Mississippi

Norman Brammer
Colorado State University

Helen Casey
Sam Houston State University

William Cornette
Southwest Missouri State

Ralph Duffy
North Seattle Community College

Carl Friedman
University of the District of Columbia

Alden Lorents
Northern Arizona University

Laurie MacDonald
Bryant College

Don Musselman
James Madison University

Richard Ramirez
Arizona State University

Shannon Scanlon
Henry Ford Community College

Steven Schindler
Kent State University

Glenn Smith
James Madison University

Marguerite Summers
Murray State University

Elizabeth Unger
Kansas State University

Susan White
Catonsville Community College

Peggy Wingo
Richland College

John C. Shepherd

C O N T E N T S

PART I Database Fundamentals

1

Chapter 1 Database Overview

2

Minicase: Teck Information Systems, 2

1.1 Database Systems, 3

1.2 The Database Approach: Shareability
and Cooperation, 6

Shareability of Data, 6

Centralized Data Management and Control, 6
Adaptability, 7

1.3 File Processing versus the Database
Approach, 7

Database Approach Summary, 18

1.4 Roles, 18

Database Administrator, 18

Systems Analyst, 19

Programmers, 19

Users, 20

The Database Management System (DBMS), 20

1.5 Summary, 27

Episode 1: Community Hospital:
Background Information, 29

Chapter 2 Data: The Foundation of the Database

32

Minicase: Suburban Hospital, 32

2.1 Data versus Information, 34

2.2 The Real World versus the Abstract
World, 34

The User's Logical World, 35

*The Programmer's Data World: The Data
Dictionary, 38*

*The Physical Stored Data World: Data as Seen
by Programs and Disk Drives, 42*

2.3 Data Element and Field Characters, 43

Name, 43

Type, 44

Representation, 44

Length, 44

Origin: Real versus Virtual, 44

Domain, 44

Value, 45

Data Independence, 45

Key versus Nonkey, 46

Data Integrity, 48

Data Security, 51

2.4 Relationships and Associations:
Relating Entities, 53

One-to-One Relationships, 54

One-to-Many Relationships, 55

Many-to-Many Relationships, 57

Cycles, 58

Loops, 60

Modeling Record/Entity Relationships, 60

2.5 Data Views or Perspectives: A
Taxonomy of Database Types, 63

Historical Perspective, 64

The Conceptual View, 64

The External View, 65

The Physical View, 65

2.6 Data Models, 65

The Relational Data Model, 66

The Hierarchical Data Model, 67

The Network Data Model, 70

2.7 Summary, 71

Episode 2: Community Hospital Adopts
the Database Approach, 73

Chapter 3

Database Structures: Implementing Relationships 77

Minicase: ICU Library, 77

3.1 Pointers, 81

Pointer Types, 82

Implementation of Pointers, 88

Location, 88

3.2 Data Structures: Relationships among Records, 91

3.3 Simple Lists, 92

*Using a Simple List to Sequentially Process
Records Based on a Primary Key, 93*

Adding Records to a Simple List, 94

Deleting Records from a Simple List, 96

*Using a Simple List to Process Secondary
Keys, 98*

How to Find the Head of a List, 99

3.4 Rings, 102

3.5 Two-Way Lists and Rings, 104

3.6 Multilists, 106

3.7 Inverted Data Structures, 108

3.8 Hierarchical Data Structures, 109

Terminology, 110

Physical Representation of Trees, 112

Tree for Indexing: Binary and B-Trees, 114

3.9 Network Data Structures, 118

Simple Network, 121

Complex Networks, 121

3.10 Multilists, Secondary Keys, and Set Types: A Comparison, 128

3.11 Summary, 129

PART II

Database Design 133

Chapter 4

An Introduction to Database Design: The Conceptual Database and Normalization 134

Minicase: Fred's Pharmacy, 134

4.1 Conceptual Database Design: An Introduction, 138

4.2 Step 1. Determine the Scope of the Design: What Are the Boundaries? 139

4.3 Step 2. Determine the Relevant Transaction the Database Must Support: What Does the Organization Do? 140

4.4 Step 3. Determine the Business Rules: Constraints and Domains, 140

4.5 Step 4. Determine the Entities: What Are the Organization's Objects? 142

4.6 Step 5. Determine the Identifier of Each Entity: Finding a Primary Key, 143

4.7 Step 6. Draw the DSD and/or E-R Diagram: Graphical Database Models, 144

4.8 Step 7. Ensure that the Conceptual Design Satisfies the Business Rules and Transactions' Needs: How Good a Model Do We Have? 146

4.9 Step 8. Add Attributes to Entities: A Difficult, Error-Prone Step, 148

Modification Anomalies, 149

Functional Dependencies, 151

*Eliminating Modification Anomalies: Rules for
Assigning Attributes to Entities, 156*

Adding Link Attributes, 169

4.10 Step 9. Add Anticipated Attributes: Back from the Future, 171

4.11 Step 10. Revise and Refine the Design, 171

- 4.12 Step 11. Identify User Views: The Database through the End User's Eyes, 171
- 4.13 A Bottom-Up Alternative, 172
- 4.14 Summary, 176
- Episode 3: Barb Redesigns Her Database, 181

Chapter 5 Logical and Physical Database Design 193

- 5.1 Relational Database Design, 194
 - Mapping to the Logical Database, 194*
 - Physical Design of Relational Database, 196*
- 5.2 Network Database Design, 198
 - Mapping to the Logical Database, 198*
 - Physical Design of Network Database, 202*
- 5.3 The Hierarchical Model, 210
 - Mapping to the Logical Database, 210*
 - Hierarchical (IMS) Physical Database Design, 216*
- 5.4 Summary, 218

PART III The Relational Model 223

Chapter 6 An Introduction to the Relational Model: Concepts and Manipulation via Relational Algebra and Calculus and Query-By-Example 224

- 6.1 Features of the Relational Model, 225
- 6.2 A Historical Perspective, 227
 - Relational Calculus Implementations, 227*
 - Relational Algebra Implementations, 228*
 - Transform-Oriented Implementations, 228*
 - Graphics-Oriented Implementations, 228*

6.3 Terminology and Concepts, 229

- Relations, 230*
- Occurrences of Relations, 230*
- Base and View Tables, 230*
- Relational Rules, 231*

6.4 Relational Assertions, 231

- Referential Integrity Assertions, 231*
- Primary Key Assertions, 232*

6.5 Relational Algebra, 232

- The SELECT Operator, 233*
- The PROJECT Operator, 238*
- The JOIN Operator, 240*
- Natural Joins and Equijoins, 241*
- The UNION Operator, 244*
- The INTERSECTION Operator, 247*
- The DIFFERENCE Operator, 247*
- The PRODUCT Operator, 249*
- Nesting Operators, 251*

6.6 Relational Calculus, 256

6.7 End User Interaction with DB2: QBE, 257

- Identifying the Desired Tables, 258*
- Retrieval, 258*
- Retrieval of Selected Rows: Implementing Algebra's SELECT Operator, 259*
- Adding New Rows: The I. Operator, 266*
- Changing Existing Rows: The U. Operator, 266*
- Deleting Rows: The D. Operator, 268*

6.8 Summary, 269

Chapter 7 An Introduction to SQL Using Database 2 (DB2) 273

- 7.1 DB2 Concepts, 275
 - Table Types, 275*
 - Objects, 276*
- 7.2 The Data Description Language, 278
 - Defining Objects: The CREATE Statement, 278*

- Deleting Objects: The DROP Statement, 286*
 - Changing a Table's Structure: The ALTER TABLE Statement, 287*
 - 7.3 Data Retrieval from a Single Table: The SELECT-FROM-WHERE Statement, 288
 - Predicates, 289*
 - SQL and the Relational PROJECT, 292*
 - Built-in SQL Functions, 296*
 - Concatenation, 299*
 - DATE and TIME Functions, 299*
 - SQL and the Relational Select: SQL with Predicates, 300*
 - Sorting Queries: the ORDER BY Clause, 304*
 - Simple Queries with Calculations and Literals, 305*
 - Table Aliases, 306*
 - Retrieval of NULL Values, 306*
 - SQL and UNION: The SQL UNION Statement, 306*
 - 7.4 Querying Multiple Tables: SQL and Join, 309
 - 7.5 Nested Select-from-Where Blocks: Subselects, 313
 - 7.6 Queries Using All or Any, 317
 - 7.7 Queries Using GROUP BY, 320
 - 7.8 Adding Rows to a Table: The Insert Statement, 322
 - 7.9 Deleting Rows from a Table: The Delete Statement, 323
 - 7.10 Changing a Table's Row Values: The Update Statement, 324
 - 7.11 Deleting Columns from Tables, 325
 - 7.12 Use of EXISTS and NOT EXISTS, 326
 - 7.13 Correlated subqueries, 329
 - 7.14 Creating Views: Logical Tables and Domain Constraints, 329
 - 7.15 Security: User-IDs, GRANT, and REVOKE, 333
 - 7.16 Data Integrity: Entity Integrity, Referential Integrity, and Currency Control, 335
 - Entity Integrity, 335*
 - Specifying Referential Integrity Rules, 335*
 - Concurrency Control, 337*
 - 7.17 Querying the System Tables, 339
 - 7.18 ANSI/IBM SQL Differences, 340
 - 7.19 Summary, 340
- Chapter 8**
Embedded SQL: COBOL and 4GLs **345**
- 8.1 The Embedded SQL Environment, 346
 - Create and Test the SQL Portion of the Host Program, 352*
 - Declare the DB2 Tables and the Associated Host Structures and Set up the Communications Area: DATA DIVISION Requirements, 352*
 - Embed the Debugged SQL Statements into the Host Program, 361*
 - Precompile the Host Program, 363*
 - Compile and Link-Edit the Host Program, 364*
 - Bind the Host Program: Creating the Application Plan, 364*
 - Execute the Host Program, 365*
 - 8.2 Single-Row Processing: Embedded SQL without Cursors, 365
 - Retrieval of One Row: Host Structures with the INTO Clause, 365*
 - Handling NULL Values: The NULL Indicator, 366*
 - Updating a Row, 368*
 - Adding a Row, 369*
 - 8.3 Multiple Row Processing: Embedded SQL with and without Cursors, 371
 - Retrieval of Multiple Rows, 371*
 - Updating Multiple Rows, 375*
 - Deleting Multiple Rows, 379*
 - 8.4 Dynamic SQL, 380

- 8.5 Automatic Error Handling: The Whenever Statement, 383
- 8.6 Transaction Processing Statements: Lock, Commit, and Rollback, 385
- 8.7 SQL and 4GLS, 386
- 8.8 Summary, 394
- Episode 4: A Purchase Order Program Is Written Using DB2, 397

PART IV The CODASYL Network Model 409

Chapter 9 The CODASYL Network Model: Theory and Implementation 410

- Minicase: Microtech Luggage Company, 410
- 9.1 Historical Perspective, 413
- 9.2 The CODASYL Architecture, 414
 - Record Types: The Conceptual Files or Entities, 417*
 - Data Items: The Fields, or Attributes, 418*
 - Set Types: Named 1:M Relationships, 419*
 - Currency Indicators: Keeping Track of the Most Recently Accessed Logical Unit, 424*
- 9.3 Writing the Schema: Using the DDL to Describe the Conceptual Database, 426
 - The SCHEMA Entry, 431*
 - The Record Entries, 432*
 - The SET Entry: Defining Set Types, 436*
- 9.4 DDL Integrity Enforcement, 445
 - Domain Constraints, 445*
 - Entity Integrity, 446*
 - Referential Integrity Constraint Enforcement: Foreign Keys, 446*
 - Data Security: Passwords, 446*
 - Data (User) Views, 446*

- 9.5 Pre-1981 DDL Differences, 447
- 9.6 External Databases: Subschemas, 448
 - Designing the External Databases: Determining User Requirements, 448*
 - Defining the External Databases: The Subschema DDL, 452*
- 9.7 Summary, 453
- Episode 5: Defining the Schema Using IDMS, 457

Chapter 10 The CODASYL Data Manipulation Language 473

Minicase: Microtech Luggage Company—Continued, 473

- 10.1 Invoking the Subschema: How to Access the Desired User View, 475
- 10.2 The Special Registers: Communicating with the DBMS, 476
 - DB-STATUS: Determining the Success or Failure of a DML Call, 476*
- 10.3 Housekeeping Commands: Ready and Finish, 478
- 10.4 Retrieval Commands: FIND and GET, 480
 - Random Access, 481*
 - Sequential Retrieval, 481*
 - Finding the Owner of a Set Occurrence: FIND OWNER, 496*
- 10.5 Adding New Records: The Store Command, 497
 - Storing Records that Are Owners Exclusively or Are Members of Singular Sets Only, 499*
 - Storing Records that Have Owners, 502*
- 10.6 Maintenance of Existing Records: The MODIFY, DISCONNECT, CONNECT, and ERASE Commands, 506
 - Changing Data Item Values: The MODIFY Command, 506*

<i>Removing a Member Record from a Set Occurrence: The DISCONNECT Command, 508</i>	
<i>Inserting a Record into a Set Occurrence: The CONNECT Command, 509</i>	
<i>Deleting Records: The ERASE Command, 512</i>	
10.7 Transaction Processing Commands: COMMIT and ROLLBACK, 515	
<i>Establishing a Quiet Point: The COMMIT Command, 515</i>	
<i>Removing Updates Since the Last Quiet Point: The ROLLBACK Command, 515</i>	
<i>Use of COMMIT and ROLLBACK with Transaction Processing, 516</i>	
10.8 Retaining Currency: Ensuring that the DBMS Does Not Update One or More Currency Indicators, 516	
10.9 Declaratives and the USE Command, 520	
10.10 Concurrency Control, 523	
10.11 Pre-1981 CODASYL DML Differences, 526	
10.12 TAB: A Microcomputer Implementation of IDMS, 527	
10.13 Summary, 527	
Episode 6: A Report Is Written Using IDMS/R, 530	
 PART V	
The Hierarchial Data Model	543
 Chapter 11	
The Hierarchial Data Model and IMS	544
11.1 Historical Perspective, 544	
11.2 IMS Concepts and Terminology, 546	
<i>Segments and Segment Occurrences, 547</i>	
<i>Relationships, 548</i>	
<i>Hierarchic Sequence: Hierarchic Order plus KSF, 548</i>	
<i>Physical and Logical Databases, 553</i>	
11.3 Implementing Relationships Using Pointers, 556	
<i>Hierarchical Pointers, 556</i>	
<i>First-Child Pointers, 557</i>	
<i>Twin Pointers, 558</i>	
<i>Summary, 558</i>	
11.4 Logical Relationships, 559	
11.5 IMS Access Methods, 561	
<i>HSAM, 561</i>	
<i>HISAM, 562</i>	
<i>HDAM, 562</i>	
<i>HIDAM, 562</i>	
<i>MSDB, 563</i>	
<i>DEDB, 563</i>	
<i>INDEX, 563</i>	
11.6 Defining the Physical Database: The DBDGEN Utility, 563	
<i>Specifying the PDB Name, 564</i>	
<i>Defining the Data Set, 564</i>	
<i>Defining the Segments and Their Hierarchic Order, 566</i>	
<i>Defining the Fields, 567</i>	
<i>Defining a HIDAM Index Database, 568</i>	
11.7 Defining the IMS Logical (External) Databases: The PSBGEN Utility, 569	
11.8 Secondary Indexes, 572	
11.9 Fast Path, 574	
<i>The Main Storage Data Base, 574</i>	
<i>The Data Entry Data Base, 574</i>	
11.10 Summary, 574	
Episode 7: Defining the IMS Conceptual Database, 579	
 Chapter 12	
Manipulating the Hierarchial Model: DL/I	586
12.1 The ENTRY and GOBACK	

- Statements: Establishing and Terminating Communications with IMS, 588
- 12.2 The PCB Mask: The Application/IMS Communications Area, 588
- The DBD Name*, 591
 - The Segment Level Number*, 591
 - The Status Code*, 592
 - The Processing Options*, 592
 - The Reserved for IMS Field*, 593
 - The Segment Name*, 593
 - The Length of the Key Feedback Area*, 593
 - The Number of Sensitive Segments*, 593
 - The Key Feedback Area*, 593
- 12.3 The I/O Areas: Record Definitions (UWAS) for IMS Segment Types, 594
- 12.4 The DL/I COBOL Call, 594
- Function*, 594
 - PCB Mask Name*, 596
 - I/O Name*, 596
 - Segment Search Arguments (SSAs)*, 596
- 12.5 Kinds of Calls, 599
- Unqualified Calls*, 599
 - Qualified Calls*, 599
- 12.6 Direct Retrieval: The GET UNIQUE (GU) Function, 600
- Direct Retrieval of the First Root: The Unqualified GU Call*, 600
 - Direct Retrieval of the First Occurrence of Any Segment: An Unqualified SSA Call*, 600
 - Direct Retrieval of a Specified Occurrence: A Qualified SSA Call*, 601
 - GU Status Codes*, 601
 - GU with Multiple SSAs*, 601
 - GU with Command Codes*, 602
- 12.7 Sequential Retrieval: The GET NEXT (GN) Function, 603
- GN Status Codes*, 603
 - Unqualified Calls: Retrieval of Every Segment in a Database*, 604
 - Unqualified SSA Calls: Retrieving All the Segments of a Given Type*, 606
 - Qualified SSA Calls: Accessing Specific Segment Occurrences for a Given Type*, 607
- 12.8 Sequential Retrieval of Children: The GET NEXT within Parent (GNP) Function, 607
- GNP Status Codes*, 608
 - Establishing Parentage*, 608
 - Unqualified Calls: Retrieval of All Children*, 608
 - Unqualified SSA Calls: Retrieving All the Children of a Specific Type*, 610
 - Qualified SSA Calls: Retrieval of Specified Child Occurrences*, 613
- 12.9 Adding New Segments: The INSERT (ISRT) Function, 614
- The ISRT Status Codes*, 616
 - Inserting a Child for an Existing Parent*, 616
 - Adding a Child When the Parent Hasn't Been Stored Yet: The Path Call*, 617
- 12.10 Updating Segments: The REPLACE (REPL) Function, 617
- REPL Status Codes*, 618
- 12.11 Deleting Segments: The DELETE (DLET) Function, 619
- 12.12 Data Integrity: The CHECKPOINT (CHKP) Function and IMS Logs, 619
- 12.13 Rolling Back Updates: The ROLL and ROLB Functions, 620
- 12.14 IMSVS86: IMS for the Microcomputer, 620
- 12.15 Summary, 621
- Episode 8: A Program that Uses DL/I to Print an IV Order Report, 624

PART VI	
Database Management	633
 Chapter 13	
Database Administration	634
Minicase: Telecorp,	634
13.1 Database Administration Functions,	636
<i>Administrative Responsibilities,</i>	637
<i>Technical Responsibilities,</i>	653
13.2 Concurrency Control,	660
13.3 Organizing the Database	
Administration Function,	662
<i>Diffused Responsibility,</i>	663
<i>Staff Organization,</i>	663
<i>Functional Organization,</i>	663
<i>Line Organization,</i>	665
13.4 Summary,	665
 Chapter 14	
Database Efficiencies: Data	
Organization and Access	667
Minicase: Tri-State Gas and	
Electric,	667
14.1 Overview of Distributed Databases,	669
14.2 Distributed Database Goals,	674
14.3 Partitioning a Database,	675
<i>Horizontal Partitioning,</i>	675
<i>Vertical Partitioning,</i>	678
14.4 Replicating a Database,	679
<i>The Master/Slave Technique,</i>	679
<i>The Two-Phase Commit,</i>	679
14.5 Advantages of a Distributed	
Database,	680
<i>More Efficient Processing,</i>	680
<i>Improved Accessibility,</i>	680
<i>Local Control of Data,</i>	680
<i>Improved System Reliability,</i>	681
14.6 Disadvantages of a Distributed	
Database,	681
<i>Updating Problems,</i>	681
<i>Query Resolution,</i>	681
<i>More Complex Recovery Procedures,</i>	682
<i>Performance Degradation,</i>	682
14.7 The Role of Microcomputers in a	
Distributed Environment,	683
<i>LAN DDBMs,</i>	683
<i>The Microcomputer as a Terminal,</i>	684
14.8 Database Machines,	686
14.9 The 4GL Environment: INFORMIX-	
4GL and ORACLE,	688
<i>INFORMIX-4GL,</i>	688
<i>ORACLE,</i>	705
14.10 Summary,	707
 Appendix A:	
Disk-Based Input/Output	710
A.1 The CPU,	710
A.2 The Operating System,	711
A.3 Channels,	712
A.4 Access Methods,	713
<i>Basic Access Methods,</i>	714
<i>Queued Access Methods: Blocking and</i>	
<i>Buffering,</i>	714
<i>Virtual Storage Access Method,</i>	717
A.5 Interface Equipment: Getting Data	
from Here to There,	717
<i>Channels,</i>	717
<i>Controllers,</i>	717
<i>The Role of the Interface Equipment: Putting It</i>	
<i>All Together,</i>	718
A.6 The Disk System,	719
<i>The Disk Drive or Spindle: The Device,</i>	719
<i>The Disk Pack: The Medium,</i>	719
<i>The Access Mechanism,</i>	723
<i>Data Storage,</i>	731
A.7 Track Capacities,	736
A.8 Summary,	737

Appendix B:	
File Organization and Access:	
Traditional Storage and Retrieval of	
Data	741
B.1 File Characteristics, 741	
B.2 File Organization versus File Access:	
An Introduction, 743	
Access, 743	
Organization, 745	
B.3 Sequential Organization and Access:	
Key Sequence Storage, 747	
Sequential Organization, 748	
Processing Sequential Files, 749	
B.4 Indexed Sequential Organization and	
Access: Flexibility with a Price, 750	
Prime Data Area, 751	
The Cylinder Index, 751	
Track Indexes, 753	
Overflow Areas, 754	
B.5 Direct Organization, 757	
Physical Organization, 758	
Direct Addressing and Processing, 760	
Indirect Addressing and Processing, 760	
B.6 Virtual Storage Access Method, 767	
VSAM Organizations, 769	
Accessing a VSAM, KSDS: A Theoretical	
Approach, 771	
Adding Records to a KSDS, 773	
Advantages of VSAM, 775	
B.7 Summary, 775	
Annotated Bibliography	778
Glossary	G-1
Index	I-1