

OBJECT- ORIENTED DATABASE MANAGEMENT

面向对象数据库管理

ALFONS KEMPER
GUIDO MOERKOTTE

清华大学出版社



Prentice-Hall

OBJECT-ORIENTED DATABASE MANAGEMENT

Applications in Engineering
and Computer Science

面向对象数据库管理

(在工程及计算机科学中的应用)

Alfons Kemper/Guido Moerkotte

江苏工业学院图书馆
藏书章

清华大学出版社

Prentice-Hall International Inc.

(京)新登字 158 号

内 容 提 要

本书在工程学科与数据库技术之间架起桥梁,是一部跨学科的指南。书中介绍数据库技术的进展性研究,即关系模型的扩展,而重点则放在革命性研究上,即面向对象数据库。本书第一部分介绍数据库应用的新领域,数据模型建立的基础,工程数据库应用。第二部分概述关系数据库模型,描述关系模型的扩展,对关系概念作出批评性评估,综述数据库进展性研究及开发。第三部分重点介绍面向对象模型的建立和用于相关对象存取的三种语言。第四部分论述控制数据库一致性的模式管理,管理并行用户共享数据库的事务控制概念,控制随时间变化对象进展性版本的支持概念,控制对存储信息存取的授权机制,第五部分重点描述索引和对象布局方法,以增强面向对象数据库应用的性能。第六部分概述现有商用面向对象数据库产品,列出颇有影响的原型开发研究。

本书适用于与数据库技术有关的各类专业人员,软件开发人员及计算机专业有关人员。

©清华大学出版社/Prentice Hall Inc. 1994

版权所有。

本书任何部分之文字,未经出版者书面同意,

不得用任何方式抄袭、节录及翻印。

Original edition published by Prentice Hall Inc. Copyright © 1994. This reprint jointly published by Tsinghua University Press/Prentice Hall. This edition is authorized for sale in mainland China only.

中国〔大陆〕版

Object-Oriented Database Management

面向对象数据库管理

Alfons Kemper/Guido Moerkotte

清华大学出版社 出版
Prentice-Hall Inc.

北京 清华园
清华大学印刷厂印刷
新华书店总店科技发行所发行

开本:787×1092 1/16 印张:44
1994年5月第1版 1994年5月第1次印刷
印数:0001—1000

ISBN 7-302-01507-4/TP·611

定价:92.00元

Preface

During the Eighties relational database systems have gained a predominant role in the commercial, administrative information management sector. At the same time database researchers began to analyze the information management requirements of other, hitherto neglected application domains, such as mechanical engineering, software engineering, VLSI design, architecture, and science—to name just a few. It turned out that relational database systems do not adequately support these so-called *non-standard* applications. This result led the database researchers to develop more suitable functionality for these “advanced” applications.

The research directions can be distinguished in two categories:

- the *evolutionary* approach: In this research and development work the conventional relational model is taken as a platform for extensions and adaptations.
- the *revolutionary* approach: Here, database research is based on a new data model, the *object-oriented model*.

While this book contains some material on the evolutionary approach—i.e., extensions of the relational model—its main thrust is on describing the object-oriented database technology, i.e., the revolutionary database research direction.

The revolutionary database research direction borrowed ideas from programming language developments—namely the object-oriented paradigm—to develop better database support for such advanced application areas. In the recent past, object-oriented database systems emerged as the “next-generation” database technology—in particular for advanced applications, such as engineering.

As mentioned before, database systems were originally developed for the commercial, administrative sector. Therefore, it is not astonishing that many good textbooks exist that cover this aspect of database technology, i.e., the utilization of conventional DBMSs in business and administration. However, there are no books providing an encompassing discussion of concepts, tools, design methodologies and systems for advanced applications. With this book we attempt to provide a *self-contained* text

about object-oriented information management. However, it also includes material on conventional (established) database technology—and extensions thereof—for assessing its strengths and limitations. The development of object-oriented databases was mainly triggered by the needs of the engineering application domains. Consequently, this book covers database concepts from the viewpoint of the engineering community. Therefore, the book can be viewed as an interdisciplinary guide with the goal to bridge the gap between engineering disciplines and database technology.

The book is aimed at two groups of readers:

- For *professionals*—e.g., engineers, software developers—dealing with databases it is a self-contained reference addressing their data modeling and database management needs. The book covers all stages of information management—starting with the conceptual design over implementation design to the physical design. The book provides a thorough description of the state of the art in database development in order to allow engineering users to assess this technology in their area of expertise.
- For *computer scientists* the book serves as an advanced guide to object-oriented database concepts and their use in advanced application domains.

The book is suitable as a text book for an advanced senior or graduate level course on advanced database concepts as well as a self-contained reference guide for professionals. The book is also useful for so-called “technical decision makers” who need to assess forthcoming database technology for their particular application scenarios. Furthermore, it is useful as additional reading in courses involving information processing in technical applications—such as geometric modeling, Computer Aided Design (CAD), Computer Integrated Manufacturing (CIM), VLSI design, software engineering, etc.

The book is self-contained inasmuch as it assumes only basic computer science and engineering prerequisites. The engineering professionals, who want to use this book as a reference in developing database support for technical applications, should have the knowledge of fundamental computer science concepts and a programming language. The computer science reader is familiarized with the database requirements imposed by engineering applications in a dedicated chapter at the beginning.

The book consists of six parts—covering the following topics:

1. Introduction of Basics
2. The Relational Approach
3. Object-Oriented Modeling and Languages
4. Control Concepts

5. Physical Object Base Design

6. Sample Systems

In Part I we first motivate the use of database systems in advanced applications, concentrating on mechanical engineering. Then, the basics of conceptual data modeling and engineering database applications are introduced. The material of the remainder of the book is (largely) illustrated by examples introduced in this part.

In Part II the relational database model is surveyed with respect to engineering application support. A critical assessment of relational concepts is followed by a description of extensions of the relational model. This part overviews the “evolutionary” database research and developments.

Part III is dedicated to introduce the modeling concepts of the object-oriented data model. The discussion of this part is based on the generic object model (GOM) which was developed by the authors. Contrary to relational databases, there is no standard object model—as of the time of writing. Therefore, we chose to base our text on a model that is *generic* inasmuch as it unites the most salient features of object-oriented data modeling in one syntactical framework.

Towards the end of Part III we address interface issues to object-oriented databases. It describes three languages for associative object access; that is, retrieval of objects from the database based on selection predicates.

In Part IV control concepts that are integrated in the object-oriented database systems are introduced. Among these we describe the schema management that controls the database consistency, the transaction control concepts that govern the sharing of the database by concurrent users, the version support concepts that control the evolution of objects over time, and the authorization mechanisms that control the access to the stored information.

In Part V the physical object base design is treated. This part is devoted to describing index and object placement methods to enhance the performance of object-oriented database applications.

The last part (Part VI) surveys the existing commercial object-oriented database products. Some systems, which were the forerunners of object-oriented database technology are treated in detail, while others can only be sketched. This part is concluded by a list of influential research prototypes.

In Figure 1 the dependencies among the six parts of the book are depicted. The thick arrows represent strong dependencies whereas the thin arrows indicate a weak dependency in the sense that the material can be read without a detailed knowledge of the parts from which the arrow emanates.

Figure 2 shows a more detailed dependency graph at the chapter level for Parts III – VI. In particular, it indicates that Chapters 11 – 13 can be skipped on first reading without compromising the understanding of the subsequent material.

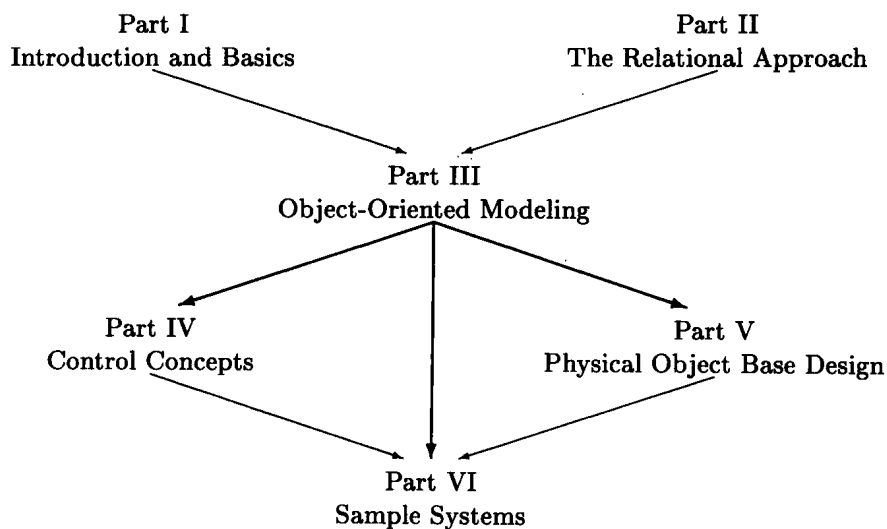


Figure 1: Dependency Graph of the Parts of the Book

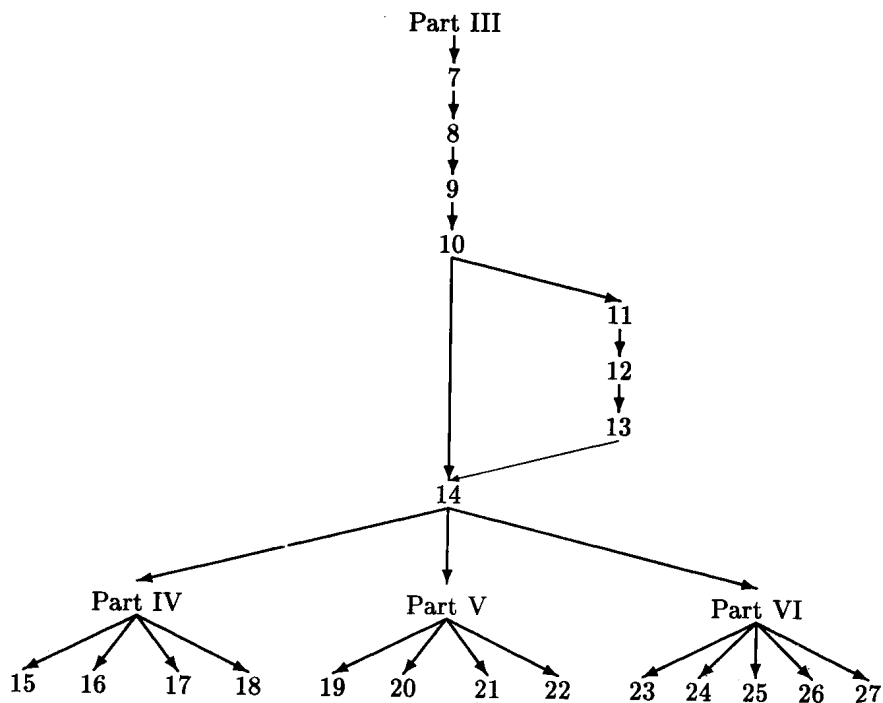


Figure 2: Dependency Graph of Parts III – VI

Contents

Preface	xix
I Introduction	1
1 New Frontiers for Database Application	3
1.1 General Remarks about Computerized Information Systems	3
1.2 The Integration Problem in CAD/CAM	4
1.3 Databases: An Integration Tool for Engineering Applications	5
2 Data Modeling Basics	11
2.1 Database Design Steps	11
2.1.1 The Requirement Analysis	13
2.1.2 The Conceptual and Implementation Design	13
2.2 The Basic Concepts of the ER Model	15
2.2.1 Entities and Entity Sets	15
2.2.2 Relationships and Relationship Types	16
2.2.3 Attributes	16
2.2.4 Key Attributes	17
2.3 Graphical Representation of ER Schemas	17
2.4 Functionality of Relationships	19
2.5 Fundamental Work on Abstraction	20
2.5.1 Aggregation	20
2.5.2 Generalization and Specialization	22
2.5.3 Combining Aggregation and Generalization	24
2.6 Exercises	26
2.7 Annotated Bibliography	28
3 Application Areas	29

3.1	Geometric Modeling	29
3.1.1	Representation Schemes for Rigid Solids	29
3.1.2	Geometrical Transformations	35
3.2	Computer-Integrated Manufacturing (CIM)	37
3.2.1	Structure and Characteristic Parameters of a Robot	37
3.2.2	Transformation between Different Coordinate Systems	39
3.2.3	Modeling the Kinematics of a Robot Axis	42
3.2.4	Simulation of Assembly Operations	44
3.3	Computer-Aided VLSI Design	45
3.3.1	The Different Design Dimensions	47
3.3.2	Versioned Objects	52
3.3.3	Cell Instantiation	52
3.4	Exercises	53
3.5	Annotated Bibliography	53

II The Relational Approach 55

4	The Relational Database Model	57
4.1	Definition of the Relational System	58
4.1.1	Relational Schema	59
4.1.2	Relational Key	59
4.2	Mapping a Conceptual Schema	60
4.2.1	Representing Entities in Relations	60
4.2.2	Representing Relationships in Relations	62
4.2.3	Schema Refinement	64
4.3	Sample Relational Database Extension	65
4.4	SQL: The Standard Relational Language	65
4.4.1	Data Definition	65
4.4.2	Schema Alteration	68
4.4.3	Indexing	68
4.4.4	The Syntax of the Query Language	71
4.4.5	The Join Operation	72
4.4.6	Transitive Closure	75
4.4.7	Aggregate Operators	77
4.4.8	Nested Queries: Set Operations in the where Clause	78
4.4.9	Group Operations	79
4.4.10	Views	80
4.4.11	Insertion, Deletion, and Update of Data	81
4.5	A Programming Language Interface of SQL	81
4.5.1	General Concepts	82

4.5.2	Host Language and Communication Variables	82
4.5.3	SQL Statements without Host Variables	83
4.5.4	Using Host Language Variables	84
4.5.5	Embedded Queries	84
4.6	Exercises	87
4.7	Annotated Bibliography	91
5	Assessment of Conventional Database Technology	93
5.1	A Representative Engineering Example	93
5.2	Weak Points of the Relational Model	95
5.2.1	Segmentation	95
5.2.2	Identifier Attributes	97
5.2.3	Lack of Data Abstraction	98
5.2.4	Only Built-in Data Types	99
5.2.5	Modeling Object Behavior	99
5.3	The "Impedance Mismatch"	100
5.3.1	One-Way Motion: Sequential Access	101
5.3.2	Nonupdatable Joins	101
5.4	Discussion and Outlook	102
5.5	Exercises	105
5.6	Annotated Bibliography	106
6	Extensions of Relational DBMSs	107
6.1	Modeling Generalization and Aggregation	107
6.1.1	Syntactic Extensions of the Relational Model	108
6.1.2	An Example Application of Generalization and Aggregation	110
6.1.3	Schema Definition	110
6.1.4	Data Manipulation	115
6.1.5	Discussion	115
6.2	Molecular Objects	116
6.2.1	Disjoint, Nonrecursive Molecules	116
6.2.2	Nondisjoint, Nonrecursive Molecular Objects	117
6.2.3	Recursive; Disjoint Molecular Objects	117
6.2.4	Recursive, Nondisjoint Molecular Object Types	117
6.2.5	Discussion of Molecular Objects	119
6.3	A General Reference Mechanism	119
6.3.1	Modeling Constructs	120
6.3.2	Extended Query Language	121
6.3.3	Lack of Type Safety	123
6.4	Abstract Data Types in the Relational Model	124
6.5	Hierarchical Object Support in XSQL	127

6.5.1	The Modeling Constructs for Complex Objects	127
6.5.2	Attributes of Type <i>reference</i>	131
6.5.3	Boundary Representation in XSQL	132
6.5.4	Evaluation of the XSQL Model	133
6.6	The Nested Relational Model NF ²	135
6.6.1	The Basic Modeling Constructs of the Nested Relational Model	136
6.6.2	Formal Definition of NF ² Relations	138
6.6.3	Representation of NF ² Relations by Attributes	139
6.6.4	The Extended NF ² Model	141
6.6.5	Data Definition Language	143
6.6.6	Modeling Boundary Representation without Redundant Vertices	146
6.6.7	Query Language	147
6.6.8	Discussion	150
6.7	Exercises	151
6.8	Annotated Bibliography	155

III Object-Oriented Modeling and Languages 157

7	Objects and Types	159
7.1	Distinction between Objects and Values	159
7.2	Object Types	160
7.2.1	The Type Definition Frame	160
7.3	Tuple-Structured Object Types	162
7.4	Instantiation	165
7.5	Object Identity	169
7.5.1	Motivation for Object Identity	170
7.5.2	Logical Object Identity	171
7.6	Shared Subobjects	172
7.7	Referencing and Dereferencing	173
7.8	Collection Types	174
7.8.1	Set-Structured Object Types	175
7.8.2	Object Type Extensions as Implicit Sets	177
7.8.3	List-Structured Object Types	177
7.9	Typing	180
7.10	Persistence	184
7.10.1	Persistence of Types	184
7.10.2	Persistence of Objects	185
7.10.3	Persistence of Variables	185

7.11	Garbage Collection	186
7.11.1	Inaccessible Objects	186
7.11.2	Detection of Inaccessible Objects	187
7.11.3	Explicit Deletion of Objects from the Database	187
7.12	Sorts	188
7.12.1	Complex Sorts	189
7.12.2	Enumeration Sorts	191
7.13	Exercises	191
7.14	Annotated Bibliography	194
8	Specifying the Behavior of Objects	197
8.1	Classification of Operations	197
8.2	Type-Associated Operations	198
8.2.1	Operation Declaration	199
8.2.2	Operation Definition	201
8.2.3	Invocation of Type-Associated Operations	202
8.3	Bottom-Up Type Definition	203
8.4	Encapsulation	203
8.5	Initialization	207
8.6	Refining the <i>delete</i> Operation by Destructors	209
8.7	Overloading	210
8.8	Value-Receiving and Value-Returning Operations	213
8.8.1	Built-In VCOs and VTOs	213
8.8.2	User-Defined VCOs	213
8.9	Free Operations and Operators	215
8.9.1	Declaration and Definition of Free Operations	215
8.9.2	Declaration and Definition of Operators	216
8.10	Defining Operations Outside the Type Definition Frame	217
8.11	Parameter Passing	218
8.12	Operations as Parameters	219
8.12.1	Motivation	219
8.12.2	Passing Selection Predicates as Parameter	220
8.13	Exercises	221
8.14	Annotated Bibliography	224
9	Implementing Relationships	227
9.1	Binary Relationships without Attributes	227
9.1.1	1 : 1 Relationship Types	228
9.1.2	1 : N Relationship Types	231
9.1.3	N : M Relationship Types	232
9.2	Binary Relationships with Attributes	233

9.3	<i>n</i> -ary Relationships	234
9.4	Examples from Our <i>Company</i> Database	235
9.4.1	1 : 1 Relationship	235
9.4.2	1 : <i>N</i> Relationships	237
9.4.3	Recursive 1 : <i>N</i> Relationship	238
9.4.4	<i>N</i> : <i>M</i> Relationship	242
9.4.5	Ternary Relationship	243
9.5	Implementing Inverse Relationships	244
9.5.1	Single-Valued Attributes Implementing 1 : 1 Relationships	244
9.5.2	Multivalued Inverse Attributes	246
9.6	Composite Object Support	248
9.6.1	Classification of Composite Object References	248
9.6.2	Dependent Exclusive Composite Object Reference	249
9.6.3	Independent Exclusive Composite Object Reference	249
9.6.4	Dependent Shared Composite Object Reference	250
9.6.5	Independent Shared Composite Object Reference	250
9.7	Exercises	251
9.8	Annotated Bibliography	254
10	Inheritance	255
10.1	Motivation	255
10.2	The General Idea of Inheritance and Subtyping	257
10.3	Substitutability	260
10.4	Terminology	263
10.5	ANY: The Common Root of All Type Hierarchies	264
10.6	A Comprehensive Example of Inheritance and Specialization	267
10.7	Dynamic (Late) Binding of Refined Operations	271
10.8	Typing Rules under Subtyping	274
10.8.1	Typing Rules under Substitutability	274
10.8.2	Examples Illustrating the Typing Rules	276
10.9	Anchor Type Constraints	277
10.10	Legal Operation Refinement	279
10.10.1	Typing Requirements	279
10.10.2	An Anecdotal Example to Illustrate the Refinement Conditions	281
10.11	Restrictions to Preserve Strong Typing	283
10.11.1	Retyping of Attributes Is Illegal	283
10.11.2	Refinement on Sets and Lists Violates Strong Typing	284
10.12	Glossary of Inheritance Terminology	287
10.13	Exercises	289

10.14 Annotated Bibliography	291
11 Virtual Types and Type Case	293
11.1 Virtual Types	293
11.2 The typecase Construct	299
11.3 Exercises	301
11.4 Annotated Bibliography	302
12 Polymorphic Operations and Generic Types	303
12.1 Polymorphic Operations	303
12.1.1 Motivation for Polymorphism	304
12.1.2 Type Variables	306
12.1.3 Type Expressions and Type Boundaries	308
12.1.4 Polymorphic Operation Declarations	309
12.1.5 An Anecdotal Example	310
12.1.6 Behavioral Type Requirements	313
12.1.7 Built-In Polymorphic Operations	315
12.2 Generic Types	316
12.2.1 Motivation	316
12.2.2 Sample Generic Type in GOM	317
12.2.3 Augmenting the Functionality of an Instantiated Generic Type	321
12.3 Exercises	322
12.4 Annotated Bibliography	323
13 Multiple Inheritance and Multiple Substitutability	325
13.1 Multiple Inheritance	325
13.1.1 Introduction to Multiple Inheritance	326
13.1.2 Illustration of Operation Ambiguity	329
13.1.3 Resolving Ambiguity under Multiple Inheritance	331
13.1.4 Type Conflicts	334
13.1.5 Benefits of Multiple Inheritance	336
13.2 Multiple Substitutability	338
13.2.1 Illustrating the Shortcomings of Multiple Inheritance	338
13.2.2 Multiple Substitutability	340
13.2.3 Representative Problems	342
13.2.4 Treating Parts as a Whole	344
13.2.5 Discussion	346
13.3 Exercises	346
13.4 Annotated Bibliography	347

14 Associative Object Access	349
14.1 Associative Selection of Objects	349
14.1.1 Selection Predicate without Parameter	350
14.1.2 Selection Predicates with Parameters	353
14.1.3 Iterators	354
14.2 GOMql: A Declarative Query Language	355
14.2.1 The Example Object Base <i>DeptStore</i>	355
14.2.2 Base Concepts of GOMql	357
14.2.3 Example Queries in GOMql	358
14.2.4 Quantifiers	361
14.2.5 Nested Queries	362
14.2.6 Invocation of Operations and Aggregate Functions	362
14.3 GOMsql: An SQL-Based Object Query Language	363
14.3.1 The Example Schema: Boundary Representation	363
14.3.2 The Basic Query Constructs	364
14.3.3 Range Variables	366
14.3.4 Nested Query Expressions	366
14.3.5 Binding Range Variables to Set-Valued Attributes	368
14.4 Query Nesting and Path Expressions	369
14.5 Different Join Types	370
14.6 Exercises	373
14.7 Annotated Bibliography	375

IV Control Concepts 377

15 Transaction Control	379
15.1 The Classical Notion of Transaction	379
15.2 Locking	384
15.2.1 Lock Modes	384
15.2.2 Two-Phase Locking	386
15.2.3 Deadlocks	387
15.3 Optimistic Concurrency Control	388
15.4 Multiple Granularity Locking	390
15.5 Recovery	394
15.6 Assessment of Classical Concurrency Control	396
15.7 Requirements of Design Transactions	396
15.8 Concurrency Control in Client/Server Architectures	399
15.9 Group Transactions	403
15.10 Exploiting Versioning for Concurrency Control	405
15.11 Exercises	407

15.12 Annotated Bibliography	409
16 Authorization	411
16.1 Overview	411
16.2 Implicit Authorization	414
16.2.1 Authorization Subjects	414
16.2.2 Authorization Operations	416
16.2.3 Authorization Objects	416
16.3 Type Hierarchies	421
16.4 Exercises	421
16.5 Annotated Bibliography	422
17 Version Control	423
17.1 Motivation	423
17.2 Organizing the Version Set	424
17.3 Reference Mechanisms and Configurations	427
17.4 Version Hierarchies	430
17.5 Change Propagation and Notification	431
17.6 Versioning in the Three-Level Database Hierarchy	435
17.7 A Simple Implementation	437
17.8 Exercises	438
17.9 Annotated Bibliography	440
18 Schema Management	441
18.1 The Notion of Schema	441
18.2 The Schema Definition Frame	443
18.3 Information Hiding for Schemata	444
18.4 Name Spaces of Schemata	445
18.5 Importing Schemata	446
18.6 Schema Consistency	449
18.7 A Taxonomy of Schema Evolution Operations	450
18.8 Conversion and Masking	452
18.8.1 Conversion	452
18.8.2 Masking	453
18.9 Exercises	453
18.10 Annotated Bibliography	456
V Physical Object Base Design	457
19 Clustering	459

19.1	Introduction	459
19.1.1	Motivation for Clustering	459
19.1.2	Reclustering Possibilities	462
19.2	An Introduction to Sequence-Based Clustering	463
19.3	PreSort Orders	466
19.4	Traversal Strategies	468
19.5	Exercises	474
19.6	Annotated Bibliography	476
20	Index Structures	479
20.1	The Running Example	479
20.2	The Basics of Indexing	480
20.3	Binary Access Support Relations	482
20.3.1	Atomic Attributes	483
20.3.2	Complex Attributes	483
20.3.3	Collection-Valued Attributes	484
20.4	Path Expressions	484
20.4.1	The Syntax	484
20.4.2	The Semantics	486
20.5	Join Operators	486
20.6	Extensions of Access Support Relations	488
20.6.1	The Canonical Extension	489
20.6.2	The Full Extension	490
20.6.3	The Left-Complete Extension	491
20.6.4	The Right-Complete Extension	492
20.7	The Storage Model of ASRs	493
20.8	Applicability of ASR Extensions	495
20.9	Physical Database Design	496
20.10	Exercises	497
20.11	Annotated Bibliography	498
21	Function Materialization	501
21.1	The Running Example	501
21.2	Motivation for Function Materialization	503
21.3	Generalized Materialization Relations	505
21.4	Retrieval of Materialized Results	507
21.4.1	Storage Representation of GMRs	509
21.5	Invalidation and Rematerialization of Function Results	509
21.6	Detection of Invalidated Function Results	511
21.6.1	Maintaining Reverse References	511
21.6.2	The Update Notification Mechanism	514