

<<数据库系统实现(英文版)>>

图书基本信息

书名：<<数据库系统实现(英文版)>>

13位ISBN编号：9787111091615

10位ISBN编号：7111091612

出版时间：2002-1-1

出版时间：机械工业出版社

作者：Hector Garcia-Molina

页数：653

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<数据库系统实现(英文版)>>

书籍目录

- 1 Introduction to DBMS Implementation
  - 1.1 Introducing: The Megatron 2000 Database System
    - 1.1.1 Megatron 2000 Implementation Details
    - 1.1.2 How Megatron 2000 Executes Queries
    - 1.1.3 What's Wrong With Megatron 2000?
  - 1.2 Overview of a Database Management System
    - 1.2.1 DataDefinition Language Commands
    - 1.2.2 Overview of Query Processing
    - 1.2.3 Main--Memory Buffers and the Buffer Manager
    - 1.2.4 Thansaction Processing
    - 1.2.5 The Query Processor
  - 1.3 Outline of This Book
    - 1.3.1 Prerequisites
    - 1.3.2 Storage-- M anagement Overview
    - 1.3.3 Query-Proce8sing Overview
    - 1.3.4 Thansaction- P rocessing Overview
    - 1.3.5 Information Integration Overview
  - 1.4 Review of Database Models and Languages
    - 1.4.1 Relational Model Review
    - 1.4.2 SQL Review
    - 1.4.3 Re1ational and Object-Oriented Data
  - 1.5 Summary of Chapter 1
  - 1.6 References for Chapter 1
- 2 Data Storage
  - 2.1 The Memory Hierarchy
    - 2.1.1 Cache
    - 2.1.2 Main Memory
    - 2.1.3 Virtual Memory
    - 2.1.4 Secondary Storage
    - 2.1.5 Tertiary Storage
    - 2.1.6 Volatile and Nonvolatile Storage
    - 2.1.7 Exercises for Section 2.1
  - 2.2 Disks
    - 2.2.1 Mechanics Of Disks
    - 2.2.2 The Disk Controller
    - 2.2.3 Disk Storage Characteristics
    - 2.2.4 Disk Access Characteristics
    - 2.2.5 Writing Blocks
    - 2.2.6 Modifying Blocks
    - 2.2.7 Exercises for Section 2.2
  - 2.3 Using Secondary Storage Effectively
    - 2.3.1 The I/O Model of Computation
    - 2.3.2 Sorting Data in SecondaJry Storage
    - 2.3.3 Merge-Sort
    - 2.3.4 Two-Phase, Multiway Merge--Sort

<<数据库系统实现(英文版)>>

- 2.3.5 Extension of Multiway Merging to Larger Relatbos
- 2.3.6 Exercises for Section 2.3
- 2.4 Improving the Access Time of Secondary Storage
  - 2.4.1 Organizing Data by Cylinders
  - 2.4.2 Using Multiple Disks
  - 2.4.3 Mirroring Disks
  - 2.4.4 Disk Scheduling and the Elevator Algorithm
  - 2.4.5 Prefetching and Large-Scale Buffering
  - 2.4.6 Summary of Strategies and Tradeoffs
  - 2.4.7 Exercises for Section 2.4
- 2.5 Disk Failures
  - 2.5.1 Intermittent Failures
  - 2.5.2 Checksums
  - 2.5.3 Stable Storage
  - 2.5.4 Error-Handling Capabilities of Stable Storage
  - 2.5.5 Exercises for Section 2.5
- 2.6 Recovery from Disk Crashes
  - 2.6.1 The Failure Model for Disks
  - 2.6.2 Mirroring as a Redundancy Technique
  - 2.6.3 Parity Blocks
  - 2.6.4 An Improvement: RAID 5
  - 2.6.5 Coping With Multiple Disk Crashes
  - 2.6.6 Exercises for Section 2.6
- 2.7 Summary of Chapter 2
- 2.8 References for Chapter 2
- 3 Representing Data Elements
  - 3.1 Data Elements and Fields
    - 3.1.1 Representing Relational Database Elements
    - 3.1.2 Representing Objects
    - 3.1.3 Representing Data Elements
  - 3.2 Records
    - 3.2.1 Building Fixed-Length Records
    - 3.2.2 Record Headers
    - 3.2.3 Packing Fixed-Length Records into Blocks
    - 3.2.4 Exercises for Section 3.2
  - 3.3 Representation Block and Record Addresses
    - 3.3.1 Client-Server Systems
    - 3.3.2 Logical and Structured Addresses
    - 3.3.3 Pointer Swizzling
    - 3.3.4 Returning Blocks to Disk
    - 3.3.5 Pinned Records and Blocks
    - 3.3.6 Exercises for Section 3.3
  - 3.4 Variable-Length Data and Records
    - 3.4.1 Records With Variable-Length Fields
    - 3.4.2 Records With Repeating Fields
    - 3.4.3 Variable-Format Records
    - 3.4.4 Records That Do Not Fit in a Block

<<数据库系统实现(英文版)>>

- 3.4.5 BLOBS
- 3.4.6 Exercises for Section 3.4
- 3.5 Record Modifications
  - 3.5.1 Insertion
  - 3.5.2 Deletion
  - 3.5.3 Update
  - 3.5.4 Exercises for Section 3.5
- 3.6 Summary of Chapter 3
- 3.7 References for Chapter 3
- 4 Index Structure
  - 4.1 Indexes on Sequential Files
    - 4.1.1 Sequential Files
    - 4.1.2 Dense Indexes
    - 4.1.3 Sparse Indexes
    - 4.1.4 Multiple Levels of Index
    - 4.1.5 Indexes With Duplicate Search Keys
    - 4.1.6 Managing Indexes During Data Modifications
    - 4.1.7 Exercises fOr Section 4.1
  - 4.2 Secondary Indexes
    - 4.2.1 Design of Secondary Indexes
    - 4.2.2 Applications of Secondary Indexes
    - 4.2.3 Indirection in Secondaxy Indexes
    - 4.2.4 Document Retrieval and Inverted Indexes
    - 4.2.5 Exercises fOr Section 4.2
  - 4.3 B-nees
    - 4.3.1 The Structure of B--trees
    - 4.3.2 Applications of B-trees
    - 4.3.3 Lookup in B-Trees
    - 4.3.4 Range Queries
    - 4.3.5 Insertion Into B-nees
    - 4.3.6 Deletion nom B-nees
    - 4.3.7 Efficiency of B-Trees
    - 4.3.8 Exercises fOr Section 4.3
  - 4.4 Hash Tables
    - 4.4.1 Secondary-Storage Hash Tables
    - 4.4.2 Insertion Into a Hash Table
    - 4.4.3 Hash-Table Deletion
    - 4.4.4 Efficiency of Hash Table Indexes
    - 4.4.5 Extensible Hash Tables
    - 4.4.6 Insertion Into Extensible Hash Tables
    - 4.4.7 Linear Hash Tables
    - 4.4.8 Insertion 1nto Linear Hash Tables
    - 4.4.9 Exercises fOr Section 4.4
  - 4.5 Summary Of Chapter 4
  - 4.6 References for Chapter 4
- 5 Multidimensional Indexes
  - 5.1 Applications Needing Multiple Dimensions

## &lt;&lt;数据库系统实现(英文版)&gt;&gt;

- 5.1.1 GWAphic Information System8
- 5.1.2 Data Cubes
- 5.1.3 Multidimensional Queries in SQL
- 5.1.4 Executing Range Queries Using Conventional Indexes
- 5.1.5 Executing Nearest--Neighbor Queries Using Conventional Indexes
- 5.1.6 Other Limitations of Conventional Indexes
- 5.1.7 Overview of Multidimensional Index Structures
- 5.1.8 Exercises for Section 5.1
- 5.2 Hash-Like Structures for Multidimensional Data
  - 5.2.1 Grid Files
  - 5.2.2 Lookup in a Grid File
  - 5.2.3 Insertion Into Grid Files
  - 5.2.4 Performance Of Grid Files
  - 5.2.5 Partitioned Hash Functions
  - 5.2.6 Comparison of Grid Files and Partitioned Hashing
  - 5.2.7 Exercises for Section 5.2
- 5.3 Tree-Like Structures for Multidimensional Data
  - 5.3.1 Multiple-Key Indexes
  - 5.3.2 Performance of Multiple Key Indexes
  - 5.3.3 kd-trees
  - 5.3.4 Operations on kd-trees
  - 5.3.5 Adapting kd-trees to Secondary Storage
  - 5.3.6 Quad Trees
  - 5.3.7 R-Trees
  - 5.3.8 Operations on R-trees
  - 5.3.9 Exercises for Section 5.3
- 5.4 Bitmap Indexes
  - 5.4.1 Motivation for Bitmap Indexes
  - 5.4.2 Compressed Bitmaps
  - 5.4.3 Operating on Run-Length Encoded Bit-Vectors
  - 5.4.4 Managing Bitmap Indexes
  - 5.4.5 Exercises for Section 5.4
- 5.5 Summary of Chapter 5
- 5.6 References for Chapter 5
- 6 Query Execution
  - 6.1 An Algebra for Queries
    - 6.1.1 Union, Intersection, and Difference
    - 6.1.2 The Selection Operator
    - 6.1.3 The Projection Operator
    - 6.1.4 The Product of Relations
    - 6.1.5 Joins
    - 6.1.6 Duplicate Elimination
    - 6.1.7 Grouping and Aggregation
    - 6.1.8 The Sorting Operator
    - 6.1.9 Expression Trees
    - 6.1.10 Exercises for Section 6.1
  - 6.2 Introduction to Physical-Query-Plan Operators

## &lt;&lt;数据库系统实现(英文版)&gt;&gt;

- 6.2.1 Scanning Tables
- 6.2.2 Sorting While Scanning Tables
- 6.2.3 The Model of Computation for Physical Operators
- 6.2.4 Parameters for Measuring Costs
- 6.2.5 I/O Cost for Scan Operators
- 6.2.6 Iterators for Implementation of Physical Operators
- 6.3 One-Pass Algorithms for Database Operations
  - 6.3.1 One-Pass Algorithms for Tuples-at-a-Time Operations
  - 6.3.2 One-Pass Algorithms for Unary, Full-Relation Operai
  - 6.3.3 One-Pass Algorithms for Binary Operations
  - 6.3.4 Exercises for Section 6.3
- 6.4 Nested-Loop Joins
  - 6.4.1 Tuple-Based Nested-Loop Join
  - 6.4.2 An Iterator for Thple--Based Nested--Loop Join
  - 6.4.3 A Block-Based Nested--Loop Join Algorithm
  - 6.4.4 Analysis of Nested-Loop Join
  - 6.4.5 Summary of AlgOrithms so Far
  - 6.4.6 Exercises fOr Section 6.4
- 6.5 TwcaPass Algorithms Based on Sorting
  - 6.5.1 Duplicate Elimination Using Sorting
  - 6.5.2 Grouping and Aggregation Using Sorting
  - 6.5.3 A Sort-Based Union Algorithm
  - 6.5.4 Sort-Based Algorithms for Intersection and Difference
  - 6.5.5 A Simple Sort--Based Join Algorithm
  - 6.5.6 Analysis of Simple Sort-Join
  - 6.5.7 A More Efficient Sort-Based Join
  - 6.5.8 Summary Of Sort-Based Algorithms
  - 6.5.9 Exercises for Section 6.5
- 6.6 Two-Pass AlgOrithms Based on Hashing
  - 6.6.1 Partitioning Relations by Hashing
  - 6.6.2 A Hash-Based Algorithm for Duplicate Elimination
  - 6.6.3 A Hash--Based Algorithm for Grouping and Aggrgation
  - 6.6.4 Hash-Based Algorithms for Union, Intersection, and Dif ference
  - 6.6.5 The Hash-Join Algorithm
  - 6.6.6 Saving Some Disk I/O's
  - 6.6.7 Summary of Hash-Based Algorithms
  - 6.6.8 Exercises for Section 6.6
- 6.7 Index-Based Algorithms
  - 6.7.1 Clustering and Nonclustering Indexes
  - 6.7.2 Index--Based Selection
  - 6.7.3 Joining by Using an Index
  - 6.7.4 Joins Using a Sorted Index
  - 6.7.5 Exercises for Section 6.7
- 6.8 Buffer Management
  - 6.8.1 Buffer Management Architecture
  - 6.8.2 Buffer Manapement Strategies
  - 6.8.3 The Relationship Between Physical Operator Selection and Buffer Management

## &lt;&lt;数据库系统实现(英文版)&gt;&gt;

- 6.8.4 Exercises for Section 6.8
- 6.9 Algorithms Using More Than Two Passes
  - 6.9.1 Multipass Sort-Based Algorithms
  - 6.9.2 Performance of Multipass, Sort--Based Algorithms
  - 6.9.3 Multipass Hash-Based Algorithms
  - 6.9.4 Performance of Multipass Hash-Based Algorithms
  - 6.9.5 Exercises fOr Section 6.9
- 6.10 Paxallel Algorithms fOr Relational Operations.
  - 6.10.1 Models of Paxallelism
  - 6.10.2 Tuple-at-aTime Operations in Parallel
  - 6.10.3 Parallel Algorithms for Full--Relation Operations
  - 6.10.4 Performance of Parallel Algorithms
  - 6.10.5 Exercises for Section 6.10
- 6.11 SummaJry of Chapter 6
- 6.12 References for ChaPter 6
- 7 The Query Compiler
  - 7.1 Parsing
    - 7.1.1 Syntax Analysis and Parse nees
    - 7.1.2 A Grammar for a Simple Subset of SQL
    - 7.1.3 The Preprocessor
    - 7.1.4 Exercises for Section 7.1
  - 7.2 Algebraic Laws for Improving Query Plans
    - 7.2.1 Commutative and Associative Laws
    - 7.2.2 Laws Involving Selection
    - 7.2.3 Pushing Selections
    - 7.2.4 Laws Involving Projection
    - 7.2.5 Laws About Joins and Products
    - 7.2.6 Laws Involving Duplicate Elimination
    - 7.2.7 Laws Involving Grouping and Aggregation
    - 7.2.8 Exercises for Section 7.2
  - 7.3 From PaJrse Thees to Logical Query Plans
    - 7.3.1 Conversion to Relational Algebra
    - 7.3.2 Removing Subqueries nom Conditions
    - 7.3.3 Improving the Logical Query Plan
    - 7.3.4 Grouping Associative/Commutat ive O perators
    - 7.3.5 Exercises for Section 7.3
  - 7.4 Estimating the Cost of Operations
    - 7.4.1 Estimating Sizes of Illtermediate ffelations
    - 7.4.2 Estimating the Size of a PrOjectiOn
    - 7.4.3 Estimating the Size of a Selectbo
    - 7.4.4 Estimating the Size of a Join
    - 7.4.5 Natural Joins With Multiple Join Attributes
    - 7.4.6 Joins of Many Relations
    - 7.4.7 Estim8ting Sizes fOr Other Operations
    - 7.4.8 Exercises for Section 7.4
  - 7.5 Introduction to Cost-Based Plan Selection
    - 7.5.1 Obtaining Estimates for Size Parameters

## &lt;&lt;数据库系统实现(英文版)&gt;&gt;

- 7.5.2 Incremental Computation of Statistics
- 7.5.3 Heuristics for Reducing the Cost of Logical Query P
- 7.5.4 Approaches to Enumerating Physical Plans
- 7.5.5 Exercises for Section 7.5
- 7.6 Choosing an Order for Joins
  - 7.6.1 Significance of Left and mght Join ArgUments
  - 7.6.2 Join nees
  - 7.6.3 Left-Deep Join nees
  - 7.6.4 Dynarnic Programming to Select a Join Order and Gr
  - 7.6.5 Dynamic Programming With More Detailed Cost fu
  - 7.6.6 A Greedy Algorithm for Selecting a Join Order
  - 7.6.7 Exercises for Section 7.6
- 7.7 Completing the Physical-Query--Plan Selection
  - 7.7.1 Choosing a Selection Method
  - 7.7.2 Choosing a Join Method
  - 7.7.3 Pipelining Versus Materialization
  - 7.7.4 Pipelining Unary Operations
  - 7.7.5 Pipelining Binary Operations
  - 7.7.6 Notation for Physical Query PlaJns
  - 7.7.7 Ordering Of Physical Operations
  - 7.7.8 Exercises for Section 7.7
- 7.8 Summary of Chapter 7
- 7.9 References for ChaPter 7
- 8 Coping With System Failures
  - 8.1 Issues and Models fOr Resilient Operation
    - 8.1.1 Failure Modes
    - 8.1.2 More About nansactions
    - 8.1.3 Correct Execution of nansactions
    - 8.1.4 The Primitive Operations of Transactions
    - 8.1.5 Exercises for Section 8.1
  - 8.2 Undo Logging
    - 8.2.1 Log Records
    - 8.2.2 The UndthLogging Rules
    - 8.2.3 Recovery Using Undo Logging
    - 8.2.4 Checkpointing
    - 8.2.5 Nonquiescent Checkpointing
    - 8.2.6 Exercises for Section 8.2
  - 8.3 Redo Logging
    - 8.3.1 The Redo--Logging Rule
    - 8.3.2 RetiOvery With Redo Logging
    - 8.3.3 Checkpointing a Redo Log
    - 8.3.4 Recovery With a Checkpointed Redo Log
    - 8.3.5 Exercises for Section 8.3
  - 8.4 Undo/Redo Logging
    - 8.4.1 The Undo/Redo Rules
    - 8.4.2 Recovery With Undo/Redo Logging
    - 8.4.3 Checkpointing aJn Undo/Redo Log



<<数据库系统实现(英文版)>>

- 8.4.4 Exercises for Section 8.4
- 8.5 Protecting Against Media Failures
  - 8.5.1 The Archive
  - 8.5.2 Nonquiescent Archiving
  - 8.5.3 Recovery Using an Archive and Log
  - 8.5.4 Exercises for Section 8.5
- 8.6 Summary of Chapter 8
- 8.7 References for Chapter 8
- 9 Concurrency Control
  - 9.1 Serial and Serializable Schedules
    - 9.1.1 Schedules
    - 9.1.2 Serial Schedules
    - 9.1.3 Serializable Schedules
    - 9.1.4 The Effect of Transaction Semantics
    - 9.1.5 A Notation for Transactions and Schedules
    - 9.1.6 Exercises for Section 9.1
  - 9.2 Conflict - Serializability
    - 9.2.1 Conflicts
    - 9.2.2 Precedence Graphs and a Test for Conflict-Serializability
    - 9.2.3 Why the Precedence-Graph Test Works
    - 9.2.4 Exercises for Section 9.2
  - 9.3 Enforcing Serializability by Locks
    - 9.3.1 Locks
    - 9.3.2 The Locking Scheduler
    - 9.3.3 Two-Phase Locking
    - 9.3.4 Why Two-Phase Locking Works
    - 9.3.5 Exercises for Section 9.3
  - 9.4 Locking Systems With Several Lock Modes
    - 9.4.1 Shared and Exclusive Locks
    - 9.4.2 Compatibility Matrices
    - 9.4.3 Upgrading Locks
    - 9.4.4 Update Locks
    - 9.4.5 Increment Locks
    - 9.4.6 Exercises for Section 9.4
  - 9.5 An Architecture for a Locking Scheduler
    - 9.5.1 A Scheduler That Inserts Lock Actions
    - 9.5.2 The Lock Table
    - 9.5.3 Exercises for Section 9.5
  - 9.6 Managing Hierarchies of Database Elements
    - 9.6.1 Locks With Multiple Granularity
    - 9.6.2 Warning Locks
    - 9.6.3 Phantoms and Handling Insertions Correctly
    - 9.6.4 Exercises for Section 9-6
  - 9.7 The Tree Protocol
    - 9.7.1 Motivation for Tree-Based Locking
    - 9.7.2 Rules for Access to Tree-Structured Data
    - 9.7.3 Why the Tree Protocol Works

## &lt;&lt;数据库系统实现(英文版)&gt;&gt;

- 9.7.4 Exercises for Section 9.7
- 9.8 Concurrency Control by TimeStamps
  - 9.8.1 Timestamps
  - 9.8.2 Physically Unrealizable Behaviors
  - 9.8.3 Problems With Dirty Data
  - 9.8.4 The Rules for Timestamp-Based Scheduling
  - 9.8.5 Multiversion Timestamps
  - 9.8.6 Timestamps and Locking
  - 9.8.7 Exercises for Section 9.8
- 9.9 Concurrency Control by Validation
  - 9.9.1 Architecture of a Validation-Based Scheduler
  - 9.9.2 The Validation Rules
  - 9.9.3 Comparison Of Three Concurrency-Control Mechanisms
  - 9.9.4 Exercises for Section 9.9
- 9.10 Summary Of Chapter 9
- 9.11 References for Chapter 9
- 10 More About Transaction Management
  - 10.1 Transactions that Read Uncommitted Data
    - 10.1.1 The Dirty-Data Problem
    - 10.1.2 Cascading Rollback
    - 10.1.3 Managing Rollbacks
    - 10.1.4 Group Commit
    - 10.1.5 Logical Logging
    - 10.1.6 Exercises for Section 10.1
  - 10.2 View Serializability
    - 10.2.1 View Equivalence
    - 10.2.2 Polygraphs and the Test for View-Serializability
    - 10.2.3 Testing for View-Serializability
    - 10.2.4 Exercises for Section 10.2
  - 10.3 Resolving Deadlocks
    - 10.3.1 Deadlock Detection by Timeout
    - 10.3.2 The Waits-For Graph
    - 10.3.3 Deadlock Prevention by Ordering Elements
    - 10.3.4 Detecting Deadlocks by Timestamps
    - 10.3.5 Comparison of Deadlock Management Methods
    - 10.3.6 Exercises for Section 10.3
  - 10.4 Distributed Databases
    - 10.4.1 Distribution of Data
    - 10.4.2 Distributed Transactions
    - 10.4.3 Data Replication
    - 10.4.4 Distributed Query Optimization
    - 10.4.5 Exercises for Section 10.4
  - 10.5 Distributed Commit
    - 10.5.1 Supporting Distributed Atomicity
    - 10.5.2 Two-Phase Commit
    - 10.5.3 Recovery of Distributed Transactions
    - 10.5.4 Exercises for Section 10.5

<<数据库系统实现(英文版)>>

- 10.6 Distributed Locking
  - 10.6.1 Centralized Lock Systems
  - 10.6.2 A Cost Model for Distributed Locking Algorithms
  - 10.6.3 Locking Replicated Elements
  - 10.6.4 Primary-CoPy Locking
  - 10.6.5 Global Locks Wom Local Locks
  - 10.6.6 Exercises for Section 10.6
- 10.7 Long--Duration nansactions
  - 10.7.1 Problems of Long nansactions
  - 10.7.2 sasas
  - 10.7.3 Compensating nansactions
  - 10.7.4 Why Compensating nansactions Work
  - 10.7.5 Exercises for Section 10.7
- 10.8 Summary of ChaPter 10
- 10.9 References for ChaPter 10
- 11 Information Integration
  - 11.1 Modes of Information Illtegration
    - 11.1.1 Problems of Information Integration
    - 11.1.2 Federated Database Systems
    - 11.1.3 Data Waehouses
    - 11.1.4 Mediators
    - 11.1.5 Exercises for Section 11.1
  - 11.2 WraPpers in Mediator-Based Systems
    - 11.2.1 Templates for Query Patterns
    - 11.2.2 WraPper Generators
    - 11.2.3 Filters
    - 11.2.4 Other Operations at the Wrapper
    - 11.2.5 Exercises for Section 11.2
  - 11.3 On--Line Analytic Processing
    - 11.3.1 OLAP Applications
    - 11.3.2 A Multidimensional View of OLAP Data
    - 11.3.3 StaJr Schemas
    - 11.3.4 Slicing and Dicing
    - 11.3.5 Exercises for Section 11.3
  - 11.4 Data Cubes
    - 11.4.1 The Cube Operator
    - 11.4.2 Cube Implementation by Materialized Views
    - 11.4.3 The Lattice of Views
    - 11.4.4 Exercises for Section 11.4
  - 11.5 Data Mining
    - 11.5.1 Data-Mining Applications
    - 11.5.2 Association-Rule Mining
    - 11.5.3 The A-Priori Algorithm
- 11.6 Summary of Chapter 11
- 11.7 References for Chapter 11
- Index



版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>