Database Management Systems

SECOND EDITION

P.S. GILL

Database Management Systems

Database Management Systems

Second Edition

P.S. Gill
Head, Department of Computer Science and Engineering
Krishna Engineering College
Ghaziabad, UP

I.K. International Publishing House Pvt. Ltd.
NEW DELHI • BANGALORE

Published by
I.K. International Publishing House Pvt. Ltd.
S-25, Green Park Extension
Uphaar Cinema Market
New Delhi–110 016 (India)
E-mail: info@ikinternational.com
Website: www.ikbooks.com

ISBN 978-93-81141-21-2

10 9 8 7 6 5 4 3 2 1

Published by Krishan Makhijani for I.K. International Publishing House Pvt. Ltd., S-25, Green Park Extension, Uphaar Cinema Market, New Delhi–110 016. Printed by Rekha Printers Pvt. Ltd., Okhla Industrial Area, Phase II, New Delhi–110 020.

To my granddaughter
Mannat Rataul and her Ballu Nani

Preface to the Second Edition

Most of the work in today's IT industry is related to Database Management Systems (DBMS). Whether it is migration of legacy databases from network to relational or implementation of new relational databases, there is need to have knowledge of DBMS technology. To understand any technology, one needs to have a strong foundation of its concepts. The DBMS concepts include the concepts of query languages, concepts of database normalization, concurrency control and recovery, etc.

This book is intended to provide an in-depth knowledge of the DBMS concepts at the undergraduate level. To understand the DBMS concepts, a student needs to have basic knowledge of DBMS concepts like process synchronization, process scheduling, etc.

Chapter 1 provides an overview of DBMS concepts like Data Models, Data Abstraction and overall structure of DBMS. The Chapter 2 covers E-R Modeling. A student should be able to model any real-world problem using the concepts.

Chapter 3 provides an overview of the Relational Data Model concepts bringing out clearly the database constraints like Domain Constraint, Key Constraint and Referential Integrity Constraints.

Chapter 4 covers Relational Algebra concepts in detail. Examples have been provided to give a clear description of this procedural query language, since Relational Algebra is a very strong tool for Query Optimization. Chapter 5 covers Tuple Relational Calculus and Chapter 6 covers Domain Relational Calculus.

Chapter 7 covers SQL. Attempt has been made to cover all categories of queries. The concepts of cursors and triggers also have been fully explained using PL/SQL. Probably, this is the first text, wherein a great effort has been made to establish equivalence between Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus and SQL, by providing a large number of equivalent queries in Chapter 7.

Chapter 8 has been newly added; it covers query optimization. The equivalence rules and query optimization strategy using expression trees have been covered in detail, providing ample number of examples.

Chapter 9 covers in detail the Normalization concepts. The student is first familiarized with the concepts of Functional Dependencies (FDs) before introducing the concepts First, Second & Third Normal Forms and Boyce Codd Normal Form (BCNF). Then the text covers the concepts of

Multi-Valued Dependencies (MVDs) before introducing the concept of Fourth Normal Form (4 NF). This is followed by the concepts of Join Dependency (JD) and Fifth Normal Form (5 NF). The text also introduces the latest Sixth Normal Form (6 NF) and Inclusion dependencies. The concepts of loss-less-join decomposition and dependency-preserving decomposition have been fully elaborated. A large number of solved examples have been provided to make the student fully familiar with the concepts. It is hoped that with the help of the information provided in this chapter, a reader will be able to design a flawless database schema.

Chapter 10 covers Transaction processing, focusing on the concepts of Serialization, Conflict Equivalence, View Equivalence, Cascade-less schedules and recoverable schedules.

Chapter 11 covers concurrency control, focusing on lock-based algorithms, time-stamp-based algorithms and other hybrid algorithms. The strengths and limitations of various algorithms, in terms of possibility of deadlocks and starvation have been covered in detail.

Chapter 12 covers deadlock handling, which includes deadlock prevention, deadlock detection and deadlock recovery.

Chapter 13 covers recovery of databases in the event of failure. The concepts like Immediate, deferred update, log-based recovery have been covered in detail.

Chapter 14 covers the issues related to Distributed Databases, like Fragmentation, Replication, Commit of Distributed Transaction Processing and Deadlock Handling in distributed databases.

Chapter 15 has been newly added. It gives introduction to various aspects of Data Warehousing and Data Mining.

Any suggestions, from the valuable readers are welcome at pavittergill@hotmail.com.

P.S. Gill

Acknowledgements

First of all, I convey my sincere thanks to my esteemed teacher Prof. L.M. Patnaik, who taught me this subject at IISc Bangalore. Also, I record by gratitude to the authors of highly acknowledged textbooks on DBMS; like Date CJ, "An Introduction to Database Systems", Addison Wesley; Silberschatz, Korth & Sudarshan, "Database System Concepts", McGraw-Hill; Elmasri & Navathe, "Fundamentals of Database Systems", Pearson; Arun K Majumdar & Bhattacharyya, "Database Management Systems", TMH, from whose works, I learnt the fundamentals of this subject.

P.S. Gill

Contents

Preface to the Second Edition *vii*
Acknowledgements *ix*

1. DBMS Overview **1**
Database System 2
Database Management System (DBMS) 3
Some Important Concepts in DBMS 3
File Processing Systems to Handle Information 6
Limitations of a File Processing System 6
Features of DBMS 7
Data Models 8
Schema and Instance 12
Data Abstraction 13
Data Independence 13
Database Languages 14
Procedural vs Non-Procedural DMLs 14
Overall Structure of DBMS 15
DBMS vs File Processing System 17
Exercises *19*
2. Entity-Relationship (ER) Modeling **21**
Attribute Types 22
E-R Modeling Notations 22
Relationship Constraints 23
Key of an Entity Set 26
Concept of Weak Entity Set 28
Features of Extended E-R Model 29
Specialization Constraints 30
Reduction of E-R Schema to Tables 34
E-R Diagram of an Airline Reservation System 44
Exercises *46*
3. Relational Data Model **48**
Relational Database Concepts 48
A Sample Database "Academics DB" 54
Exercises *56*

4. Relational Algebra **57**
Basic Operations 57
Additional Operations 57
Extended RA Operations 57
Additional Operations of Relational Algebra 63
Extended Relational-Algebra Expressions 67
Database Modification 69
Views 71
Exercises *72*
5. Tuple Relational Calculus **76**
Formal Definition of TRC 76
Safe TRC Expressions 78
Exercises *79*
6. Domain Relational Calculus **81**
Formal Definition of DRC 81
Safe Expression in DRC 83
7. Structured Query Language **85**
Characteristics of SQL 85
Merits of SQL 85
SQL Data Types & Literals 86
Categories of SQL Commands 88
SQL Operators 89
Tables, Views & Indexes 89
Queries & Sub-queries 91
Aggregate Functions 92
Insert, Update & Delete Operations 92
Joins 93
Union, Intersect & Minus 93
Cursors in SQL 94
Navigating through SQL 95
DDL 95
DML 101
PL/SQL 108
Triggers 110
Cursors 111
Assertions 113
Some Solved Problems 114
Exercises *132*
8. Query Optimization **134**
Introduction 134
Equivalent Queries in SQL 134

Varying Time and Space Complexity of Query Evaluation Plans 136
Equivalence Rules for Query Optimization 137
Query Optimization Strategy using Expression Trees 140
Exercises *144*
9. Normalization of Relational Schemas **145**
Concept of Functional Dependencies (FDs) 145
Normalization 158
Boyce Codd Normal Form (BCNF) 165
Multi-Valued Dependencies (MVDs and 4 NF) 173
BCNF to 5 NF 175
Join Dependency (JD) 181
Sixth Normal Form (6 NF) 185
Concept of Inclusion Dependencies 186
Some Solved Examples 187
Points to Remember 206
Exercises *207*
10. Transaction Processing Concepts **209**
Transaction 209
Commit 209
Abort 209
ACID Properties 209
Transaction Schedule 213
Serial Schedule 213
Concurrent Schedules 214
Equivalent Schedules 215
Serializable Schedules 215
Serialization of Concurrent Schedules 216
View-Serializable Schedule 221
Cascading Roll-backs 221
Cascade-less Schedules 222
Recoverable Schedules 222
Some Solved Examples 222
Points to Remember 227
Exercises *227*
11. Concurrency Control **230**
Goal of Concurrency Control Schemes 230
Concurrency Control Protocols 230
Time-Stamp Based Algorithms 240
Starvation-Free Algorithms 244
Points to Remember 248
Exercises *249*

12. Deadlock Handling in Transaction Processing **252**
Deadlock in Transaction Processing 252
Deadlock Prevention 253
Graph-Based Locking to Prevent Deadlocks 253
Deadlock Detection & Recovery 255
Exercises *258*
13. Recovery **260**
Commit Transaction 260
Roll-back Transaction 261
System Log 261
Transaction Recovery 261
System Recovery 262
Modes of Database Updates 262
Recovery from System Failures 263
Exercises *269*
14. Distributed Database Systems **270**
Node or Site 270
Characteristics of a Distributed DBMS 270
Data Replication 270
Data Fragmentation 271
Transparency Requirements 272
Distributed Transactions 272
Transaction Commit Protocols 273
Locking Protocols 276
Deadlock Handling in Distributed Databases 276
Exercises *280*
15. Brief Introduction to Data Warehousing and Data Mining **281**
Data Warehousing 281
Normalized vs Dimensional Approach for Storage of Data 281
Top-down vs Bottom-up Design Methodologies 282
Hybrid Design 283
Data Warehouses vs Operational Databases 283
Data Mining 285
Role of Data Mining in Science and Engineering 288
Exercises *289*
Index *291*

1

DBMS Overview

A database is a collection of logically related data; and a Database Management System (DBMS) is a set of programs specifically designed to create, maintain, update and access databases. Together the DBMS and database form a Database System.

The use of computers for management of databases is almost as old as the existence of digital computers. The older Database Management Systems used to be tightly coupled to the platforms and highly customized to the nature of underlying databases. But as computers gained in speed and memory space, the DBMSs became more generic and general-purpose.

In 1968, IBM came out with a DBMS known as Information Management System (IMS) on IBM/360 mainframe computer. The IMS was designed for the Apollo Program and it was based on the Hierarchical Model. The Hierarchical Model stores data items in the form of tree; wherein a data node can have any number of child nodes but only one parent node. It facilitates modeling of one-to-many relationships amongst the data items; but fails to model many-to-many relationships. The Hierarchical Databases were superseded by CODASYL Databases, which were based on the Network Model, wherein the data items are linked to each other as a generic graph (network). It facilitated the modeling of many-to-many relationships amongst the data items. Both the hierarchical and network databases make use of pointers to link the data items. Traversing through the pointers, while searching for data items, is a messy job.

In 1970, E.F. Codd proposed a new model for databases, known as "Relational Model". In Relational Databases, information is stored in flat (2-D) tables. Each table (called a relation) has a unique name and is created on a specified schema. The schema is ordered list of attributes, with each attribute representing a property of the information to be stored. In the table, each attribute is represented by a column and it has a specified domain (a set of valid values that the attribute can assume). Each table is a set of rows (known as tuples), with each row representing information of one data item. Each tuple has an ordered list of values (known as fields) with each field pertaining to corresponding attribute. The tables are related to each other through the concept of Primary Key-Foreign Key relationships, without use of any pointers. In the physical storage, each table is stored as a file, with each tuple forming a record in the file. A query language known as Structured Query Language (SQL) is used for creation, maintenance, update and access of tables. Within a short span, the relational model became highly popular due to its simple implementation, user-friendly query structures and strong mathematical foundations. Popular Relational DBMS available from known vendors are ORACLE, DB-2 (of IBM) and SQL Server (of Microsoft).

DATABASE SYSTEM

A Database System comprises a Database and a DBMS to create, maintain, update and access the database:

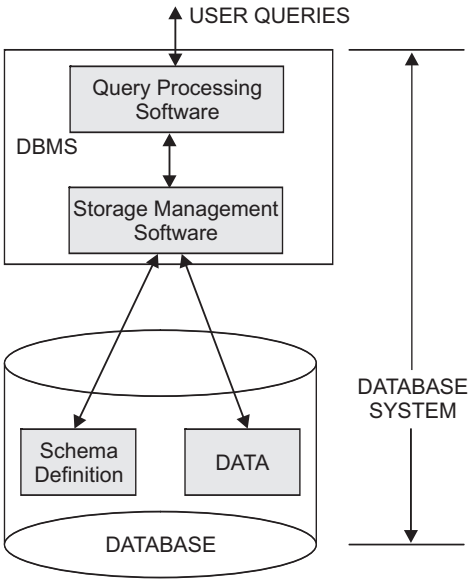


Fig. 1.1

Database

A database is a collection of logically related data that can be recorded. The information stored in the database must have the following implicit properties:

(a) It must represent some aspect of real-world; like a college or a company, etc. The aspect represented by a database is called its "Mini-world".

(b) It must have a logically coherent collection of data, inherent meaning (semantics), which must be well understood.

(c) The repository of data must be designed, developed and implemented for a specific purpose. There must be an intended group of users, who must have some pre-conceived applications of the data.

A Database System has the following major organs:

- **Sources of information,** from where it derives its data.
- **Some related real-world events,** which influence its data.
- **Some intended users,** who would be interested in its data.

For example, in the college database, sources of information will be students, faculty, labs, etc. The real-world events affecting the information in the database will be admissions, exams, results & placements etc. The set of intended users will be faculty, students, administrative staff, etc.

DATABASE MANAGEMENT SYSTEM (DBMS)

A Database Management System (DBMS) refers to a set of programs for defining, creation, maintenance and manipulation of a database. A DBMS must facilitate the following major functions:-

- **Defining of Database Schema:** The DBMS must facilitate defining the database structure, i.e., defining of data types, relationships amongst the data and specification of the integrity constraints to be enforced on the database. It should also facilitate specifying the access rights of authorized users.
- **Manipulation of the Database:** The DBMS must facilitate the following functions related to update and access of stored information:-

  Insertion of new data

  Update of changed data

  Deletion of defunct data

  Access of data for read and report generation
- **Concurrency Control and Sharing of Data:** The DBMS must enable concurrent execution of transactions, with a view to improve the utilization of system resources like CPU. To make it possible, the DBMS must permit sharing of data items amongst the concurrent transactions, while preserving the database consistency and integrity.
- **Database Security:** The DBMS must protect the database against unauthorized/malicious access. For this, DBMS must support features for grant/revoke of access rights to the users, as decided by the Database Administrator. Each authorized user is permitted to access the database only to the extent of its granted access rights. Any attempt by a user to exceed its access rights is denied by the system
- **Database Recovery:** The DBMS should maintain a log of all the updates performed by active transactions, in a log file maintained on a non-volatile media (hard disk). The entries in the log file facilitate recovery of the system, in the event of failures. Also, DBMS must support facilities for taking of periodic backup of the database. The backups are safely archived, which are used to recover the system in the event of catastrophic failures like Disk Crash.

SOME IMPORTANT CONCEPTS IN DBMS

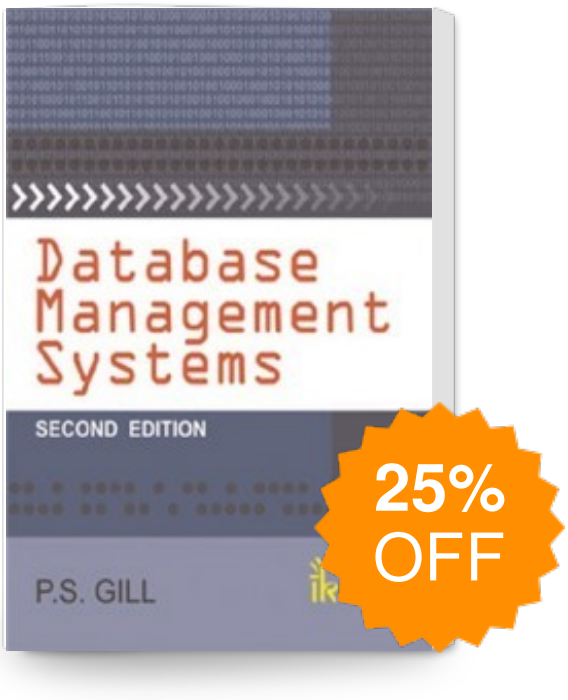
Transaction

A Transaction refers to a unit of work in DBMS. It is basically a sequence of Data Manipulation operations. The main requirement is that a transaction must be executed atomically (indivisibly), i.e. it must be executed either fully or not at all. If the system fails during execution of a transaction, then the transaction must be rolled back to its initial state (i.e., effect of its partial execution must be undone. After roll-back, it will be as though the transaction had never commenced execution).

Consider a table representing Bank Account (Account_Number, Balance)

Suppose there is a transaction "Transfer Rs 1000/= from Account_Number 101 to Account_Number 203"

Database Management Systems



Publisher : IK International ISBN : 9789381141212 Author : P. S. Gill