20MCA12C RELATIONAL DATABASE MANAGEMENT SYSTEM

UNIT I: Introduction

FACULTY

Dr. K. ARTHI MCA, M.Phil., Ph.D.,

Assistant Professor,

Postgraduate Department of Computer Applications,

Government Arts College (Autonomous),

Coimbatore-641018.

20MCA12C RELATIONAL DATABASE MANAGEMENT SYSTEM

Objective: To motivate and enable the students in knowing about the concepts of database systems, model and its architecture.

UNIT I: Introduction - Database system applications - purpose of database system - View of data - Database Languages - Relational Databases - Database Design - Data Storage and Querying - Transaction Management - Database Architecture - Database Users and Administrators.

UNIT II: Relational Databases - Relational Model - Structure of Relational Databases - Fundamental Relational Algebra Operations - Additional Relational Algebra Operations. SQL - Background - Data Definition - Basic Structure of SQL Queries - Set Operations - Aggregate Functions - Null values - Nested Sub queries - Views - Modification of the Database.

UNIT III: Database design: Database Design and the E - R Model - Design Phases - Design Alternatives - The Entity Relationship Model - Constraints - Entity Relationship Diagrams - Extended E - R features - Specialization - Generalization - Aggregation - Reduction to Relational Schemas.

UNIT IV: Relational Database Design - Features of Good Relational Designs - Atomic Domains and First Normal Form - Decomposition using Functional Dependencies - Keys and Functional Dependencies – Boyce Codd Normal Form - BCNF and Dependency Preservation - Third Normal Form - Functional Dependency Theory - Lossless Decomposition - Dependency Preservation - BCNF Decomposition Algorithm - 3NF Decomposition - Multivalued Dependencies - Fourth Normal Form - 4NF Decomposition.

UNIT V: Database System Architectures - Centralized and Client / Server Architectures - Centralized Systems - Client / Server Systems - Server System Architectures - Parallel Systems - Distributed Systems - Network Systems.

TEXT BOOKS:

1. Database System Concepts " by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Fifth edition, 2006, Mc Graw Hill International Edition.

REFERENCE BOOKS:

1. "An Introduction to Database Systems "by Bipin c. Desai, West Publishing Company, 1990.
2. "Database Management Systems" by Elmasri and Navathe.

Overview of DBMS

- Database is a collection of data, describing the activities of one or more related organization.
- University database
- Entities – students, faculty, courses & classrooms
- Relationship – between entities
- DBMS is a s/w designed to assist in maintaining and utilizing large collections of data.
- The database is a collection of table, files or datasets.
- Each table is a collection of fields, columns or data items.
- One or more columns in each table may be selected as the primary key.
- There may be additional unique keys or non-unique indexes to assist in data retrieval.
- Columns may be fixed length or variable length.
- Records may be fixed length or variable length.
- Table and column names may be restricted in length (8, 16 or 32 characters).
- Table and column names may be case-sensitive.

Managing data

- Database Design & application development
    - Descriptions of real world enterprise – university
    - Organization of stored data
- Data Analysis
    - How qn is answered
- Concurrency and Robustness
    - Access data concurrently
    - Protection of data
- Efficiency and Scalability
    - DBMS store large datasets and ans qn against this data efficiently

Collection of interrelated data

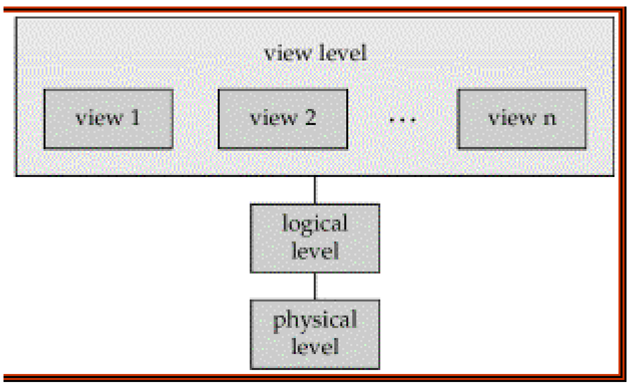
- Set of programs to access the data
- DBMS contains information about a particular enterprise
- DBMS provides an environment that is both *convenient* and *efficient* to use.
- Database Applications:
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions
- Databases touch all aspects of our lives

Purpose of Database System

- In the early days, database applications were built on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become part of program code
    - Hard to add new constraints or change existing

ones

View of Data



Relational Model

A Sample Relational Database

customer-id	customer-name	customer-street	customer-city
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account-number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer-id	account-number
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Data Storage and Querying

A database system is partitioned into modules that deal with each of theresponsibilities of the overall system. The functional components of a database system can be broadly divided into the storage manager and the query processor components.

The *storage manager* is the component of a database system that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system. The storage manager is responsible for the interaction with the file manager. The raw data are stored on the disk using the file system provided by the operating system. The storage manager translates the various DML statements into low-level file-system commands. Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.

The storage manager components include:

• **Authorization and integrity manager**, which tests for the satisfaction of integrity constraints and checks the authority of users to access data.

• **Transaction manager**, which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.

• **File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

• **Buffer manager**,which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory. The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.

• **Data dictionary**, which stores metadata about the structure of the database, in particular the schema of the database.

• **Indices**, which can provide fast access to data items. Like the index in this textbook, a database index provides pointers to those data items that hold a particular value. For example, we could use an index to find the instructor record with a particular ID, or all instructor records with a particular name. Hashing is an alternative to indexing that is faster in some but not all cases.

2. The Query Processor

The query processor components include:

• **DDL interpreter**,which interprets DDL statements and records the definitions in the data dictionary.

• **DML compiler**,which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

Transaction Management

A **transaction** is a collection of operations that performs a single logical function in a database application. Each transaction is a unit of both atomicity and consistency. Thus, we require that transactions do not violate any databaseconsistency constraints. That is, if the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates. However, during the execution of a transaction, it may be necessary temporarily to allow inconsistency, since either the debit of A or the credit of B must be done before the other. This temporary inconsistency, although necessary, may lead to difficulty if a failure occurs.

Database Architecture

The architecture of a database system is greatly influenced by the underlying computer system on which the database system runs. Database systems can be centralized, or client-server, where one server machine executes work on behalf of multiple client machines. Database systems can also be designed to exploit parallel computer architectures. Distributed databases span multiple geographically separated machines.

The issues include how to store data, how to ensure atomicity of transactions that execute at multiple sites, how to perform concurrency control, and how to provide high availability in the presence of failures. Distributed query processing and directory systems are also described in this chapter. Most users of a database system today are not present at the site of the database system, but connect to it through a network. We can therefore differentiate between client machines, on which remote database users work, and server machines, on which the database system runs.

Database applications are usually partitioned into two or three parts, as in Figure 1.6. In a two-tier architecture, the application resides at the client machine, where it invokes database system functionality at the server machine through query language statements. Application program interface standards like ODBC and JDBC are used for interaction between the client and the server.

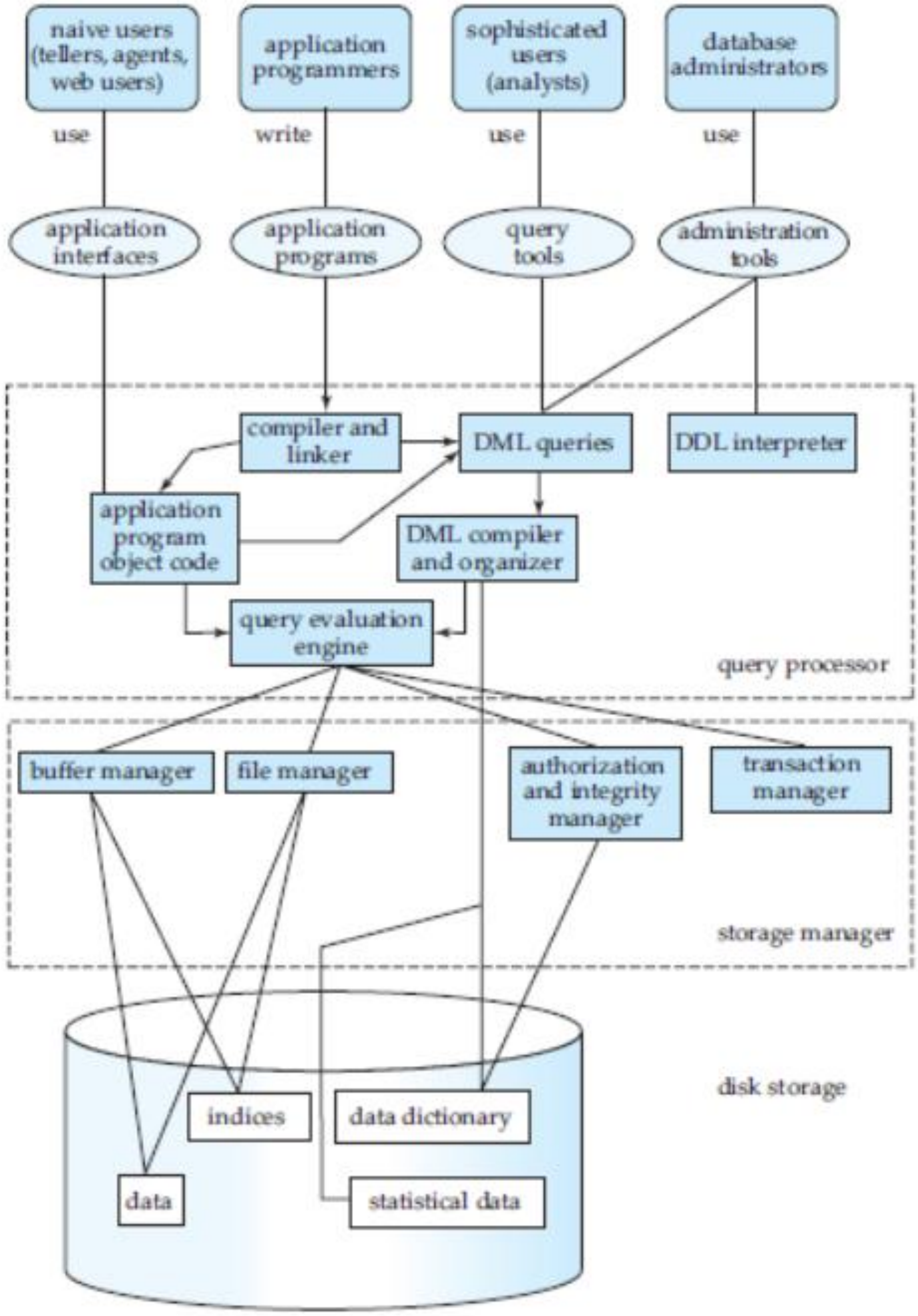


Figure 1.5 System structure.

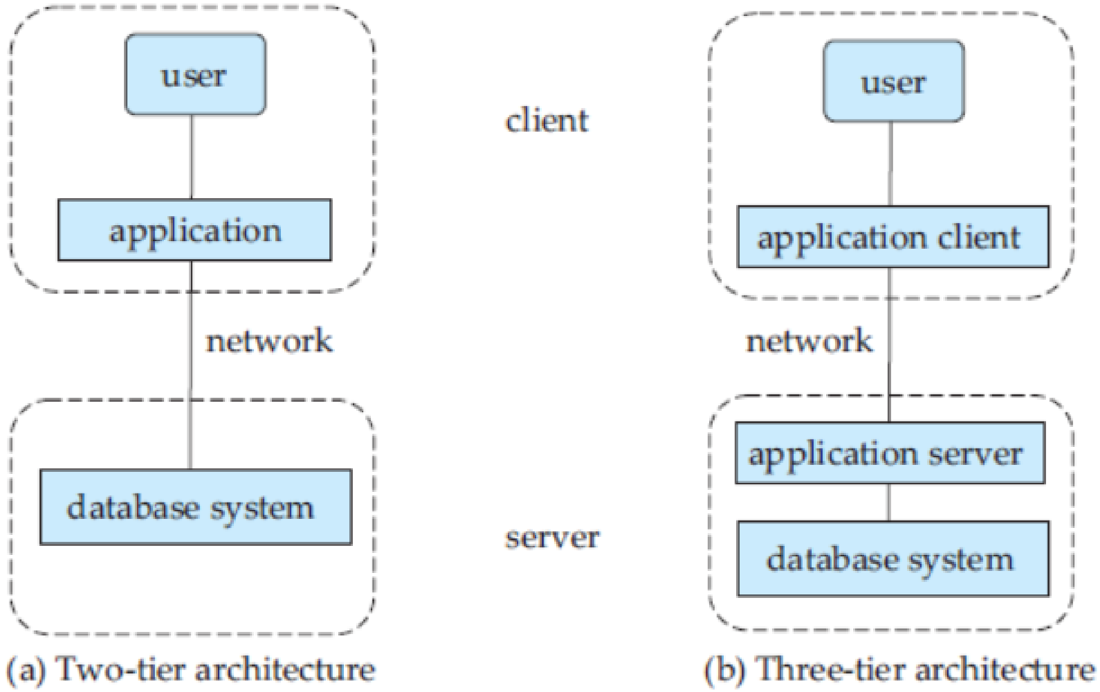


Figure 1.6 Two-tier and three-tier architectures.

Database Users and Administrators

A primary goal of a database system is to retrieve information from and store new information into the database. People who work with a database can be categorized as database users or database administrators.

1. Database Users and User Interfaces

There are four different types of database-system users, differentiated by the way they expect to interact with the system. Different types of user interfaces have been designed for the different types of users.

• **Naive users** are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.

• **Application programmers** are computer professionalswho write application programs. Application programmers can choose frommany tools to develop user interfaces. **Rapid application development (RAD)** tools are tools that enable an application programmer to construct forms and reportswith minimal programming effort.

• **Sophisticated users** interact with the system without writing programs. Instead, they form their requests either using a database query language or by using tools such as data analysis software. Analysts who submit queries to explore data in the database fall in this category.

• **Specialized users** are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework. Among these applications are computer-aided design systems, knowledgebase and expert systems, systems that store data with complex data types (for example, graphics data and audio data), and environment-modeling systems.

2. Database Administrator

One of the main reasons for using DBMSs is tohave central control of both thedata and the programs that access those data. A person who has such central control over the system is called a **database administrator (DBA**). The functions of a DBA include:

• **Schema definition.** The DBA creates the original database schema by executing a set of data definition statements in the DDL.

• **Schema and physical-organization modification.** TheDBAcarries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

• **Granting of authorization for data access.** By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.

• **Routine maintenance.** Examples of the database administrator's routine maintenance activities are:

1. Periodically backing up the database, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.

2. Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.

THANK YOU

This content is taken from the text books and reference books prescribed in the syllabus.