Unit 1

Introduction

I MSC CS

GAC CBE

Table of Contents

Introduction

DataBase Systems manage collections of data

Database Systems

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use
- Database systems are used to manage collections of data that are:
  - Highly valuable
  - Relatively large
  - Accessed by multiple users and applications, often at the same time

Database Systems

- A modern database system is a complex software system whose task is to manage a large, complex collection of data.
- Databases touch all aspects of lives

Database System Applications

The software that interacts with end users, applications, and the database itself to capture and analyze the data.

Database System Applications

- **Enterprise Information**
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.
- **Manufacturing**: management of production, inventory, orders, supply chain.
- **Universities**: registration, grades
- **Airlines**: reservations, schedules
- **Document databases**

Database System Applications

- **Banking and finance**
  - customer information, accounts, loans, and banking transactions.
  - Credit card transactions
  - Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data
- **Telecommunication**: records of calls, texts, and data usage, generating monthly bills, maintaining balances on prepaid calling cards

Database System Applications

- **Web-based services**
  - Online retailers: order tracking, customized recommendations
  - Online advertisements
- **Navigation systems**: For maintaining the locations of various places of interest along with the exact routes of roads, train systems, buses, etc.

Purpose of Database Systems

To Overcome the difficulties of File Processing

Purpose of Database Systems

- Data Independence.
- Efficient Data Access.
- Data Integrity and security.
- Data administration.
- Concurrent access and Crash recovery.
- Reduced Application Development Time

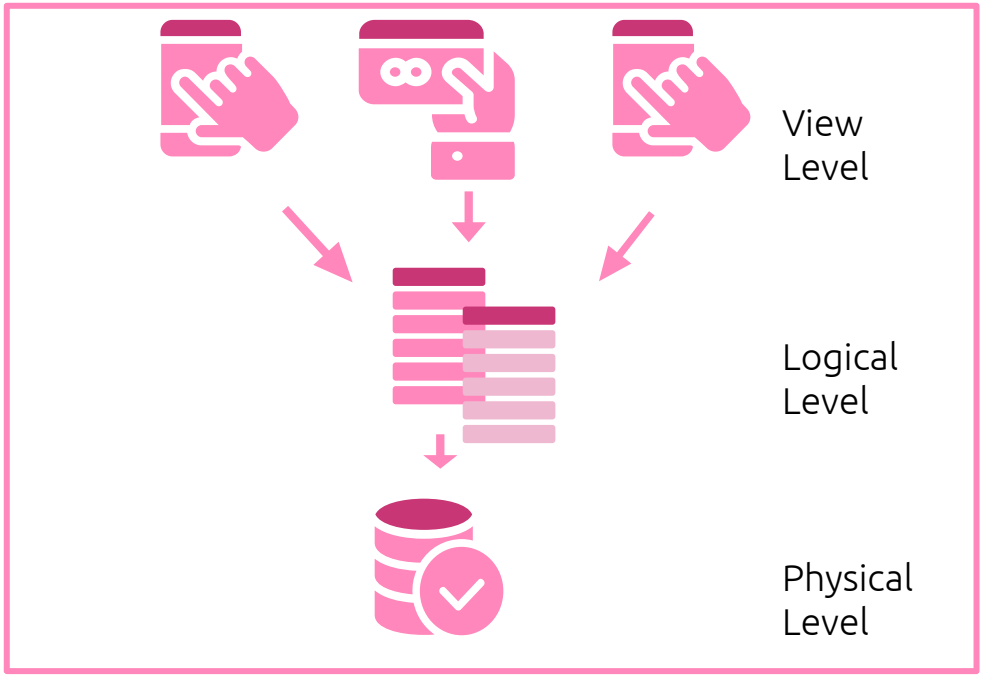
View of Data

Narrates how the data is visualized at each level of data abstraction

View of Data

- A database system is a collection of interrelated data and a set of programs that allow users to access and modify these data.
- A major purpose of a database system is to provide users with an abstract view of the data.
  - **Data models**
    - A collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
  - **Data abstraction**
    - Hide the complexity of data structures to represent data in the database from users through several levels of data abstraction.

Three Levels Of Abstraction



Levels of Abstraction

- **Physical level**: describes how a record (e.g., emp) is stored.
- **Logical level**: describes data stored in database, and the relationships among the data.

```
type emp = record
        eno: integer;
    ename : string;
    dno : integer;
    salary : integer;
end;
```

- **View level**: application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

Database Languages

Accessing and updating the data organized by the appropriate data model

Data Definition Language

- Specification notation for defining the database schema

```
create table emp(eno number(8)
primary key,ename varchar2(20),dno
number(8) references
dept(dno),salary number(8);
```

Data Definition Language

- There are basically two types of data-manipulation language
  - **Procedural DML** - require a user to specify what data are needed and how to get those data.
  - **Declarative DML** - require a user to specify what data are needed without specifying how to get those data.
- Declarative DMLs are usually easier to learn and use than are procedural DMLs. Declarative DMLs are also referred to as non-procedural DMLs
- The portion of a DML that involves information retrieval is called a **query** language.

Data Manipulation Language

Language for accessing and updating the data organized by the appropriate data model. DML also known as query language.

Data Manipulation Language

- DDL compiler generates a set of table templates stored in a ***data dictionary***
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization
    - Who can access what

SQL Query Language

SQL query language is nonprocedural. A query takes as input several tables (possibly only one) and always returns a single table.

SQL Query Language

- Eg. find all employee in Marketing dept

```
select ename from emp where dno =10;
```

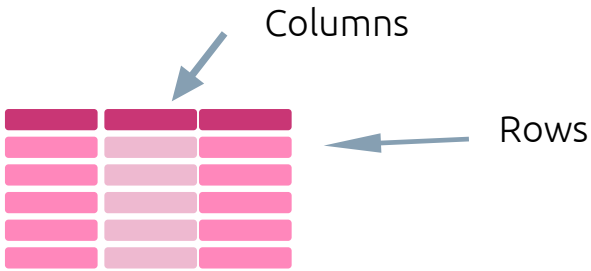
- SQL is **NOT** a Turing machine equivalent language
- To be able to compute complex functions SQL is usually embedded in some higher-level language
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database
-

Relational Databases

Data stored in table form

Relational Model

- All the data is stored in various tables.



Relational Database

EMP Table

ENO	ENAME	DNO	SALARY
13	Brindha	10	15000
22	Parthiban	10	20000
24	Mugilan	14	25000
11	Aswatha	11	10000
25	Madhavan	14	-
26	Kesavan	11	-

Database Design

The process of designing the general structure of the database

Database Design

- Logical Design - Deciding on the database schema. Database design requires that a "good" collection of relation schemas. is found.
  - Business decision - What attributes should be recorded in the database?
  - Computer Science decision - What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- Physical Design - Deciding on the physical layout of the database

Transaction Management

Collection of operations that performs a single logical function in a database application

Transaction Management

- **Transaction-management component** ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- **Concurrency-control manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Data Storage and Querying

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components can be divided into storage manager, query processor and transaction management

Storage Manager

A program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

Storage Manager

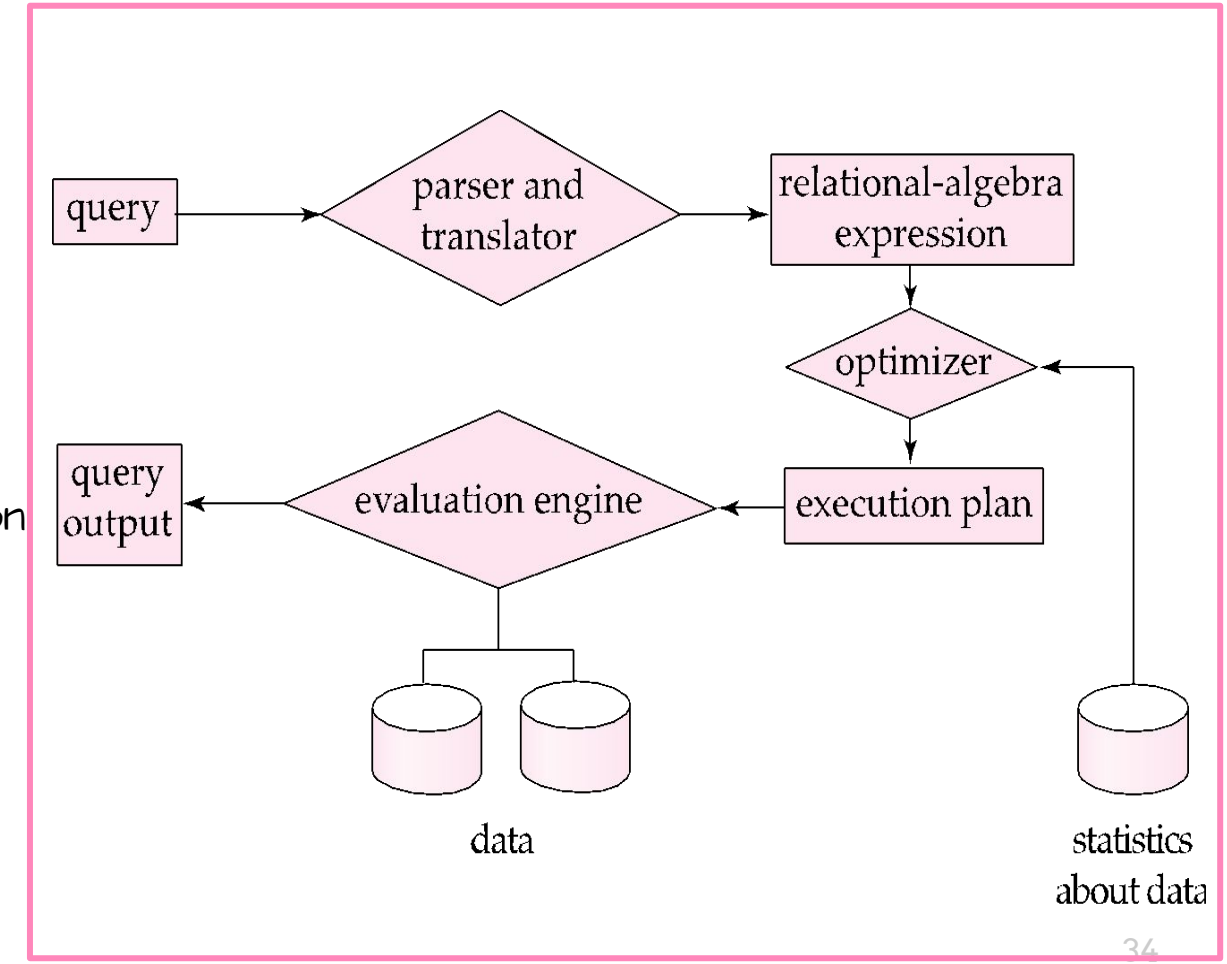
- The storage manager is responsible to the following tasks:
  - Interaction with the OS file manager
  - Efficient storing, retrieving and updating of data
- The storage manager components include:
  - Authorization and integrity manager
  - Transaction manager
  - File manager
  - Buffer manager

Query Processor

- The query processor components include:
  - DDL interpreter - interprets DDL statements and records the definitions in the data dictionary.
  - DML compiler - translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.
    - The DML compiler performs query optimization; that is, it picks the lowest cost evaluation plan from among the various alternatives.
  - Query evaluation engine - executes low-level instructions generated by the DML compiler.

Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation

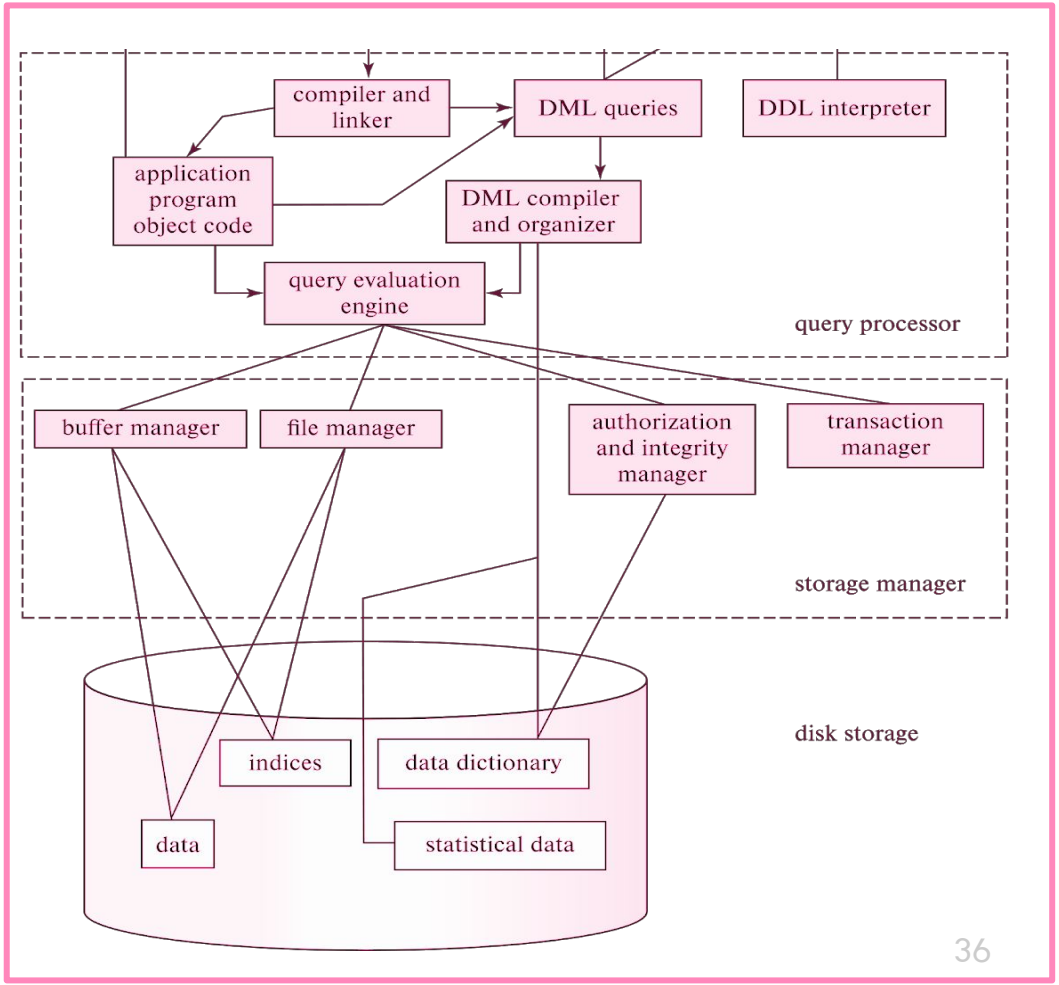


Database Architecture

Seen as a single tier or multi-tier.

Database Architecture

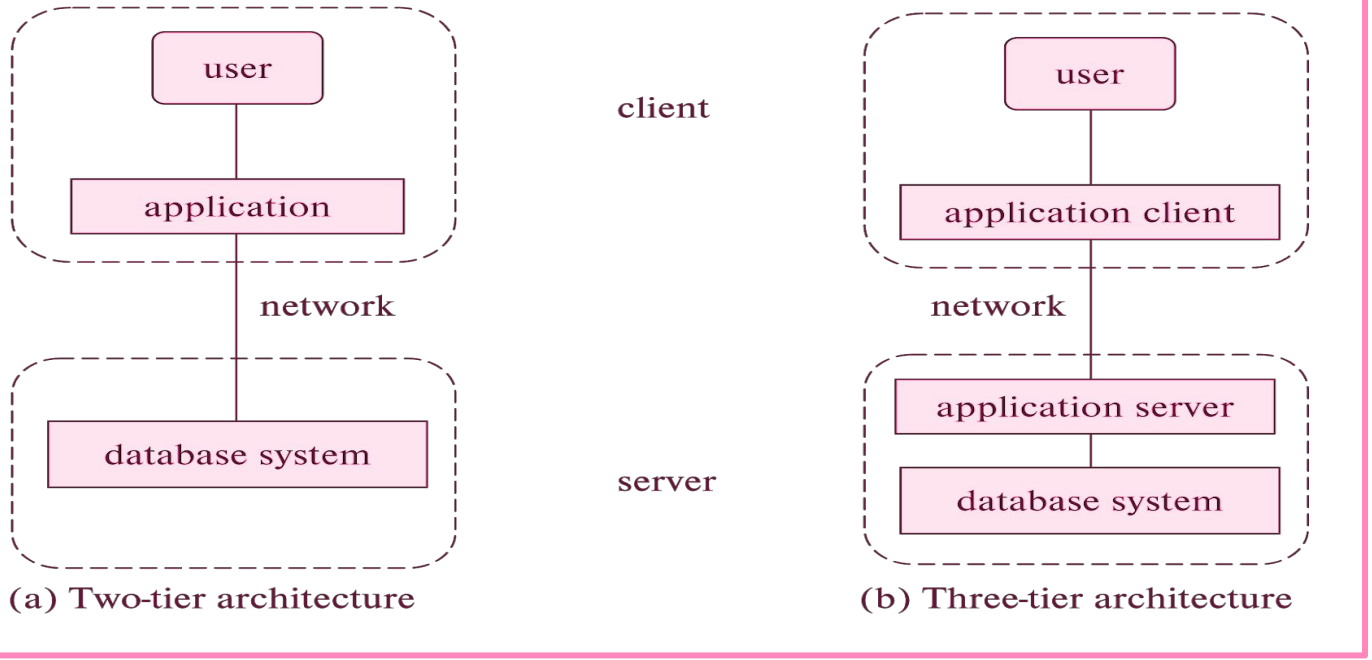
- Centralized databases
  - One to a few cores, shared memory
- Client-server,
  - One server machine executes work on behalf of multiple client machines.



Database Architecture

- Parallel databases
  - Many core shared memory
  - Shared disk
  - Shared nothing
- Distributed databases
  - Geographical distribution
  - Schema/data heterogeneity

Database Architecture



(a) Two-tier architecture

(b) Three-tier architecture

Database Architecture

Database applications are usually partitioned into two or three parts

- Two-tier architecture - the application resides at the client machine, where it invokes database system functionality at the server machine
- Three-tier architecture - the client machine acts as a front end and does not contain any direct database calls.
  - The client end communicates with an application server, usually through a forms interface.
  - The application server in turn communicates with a database system to access data.

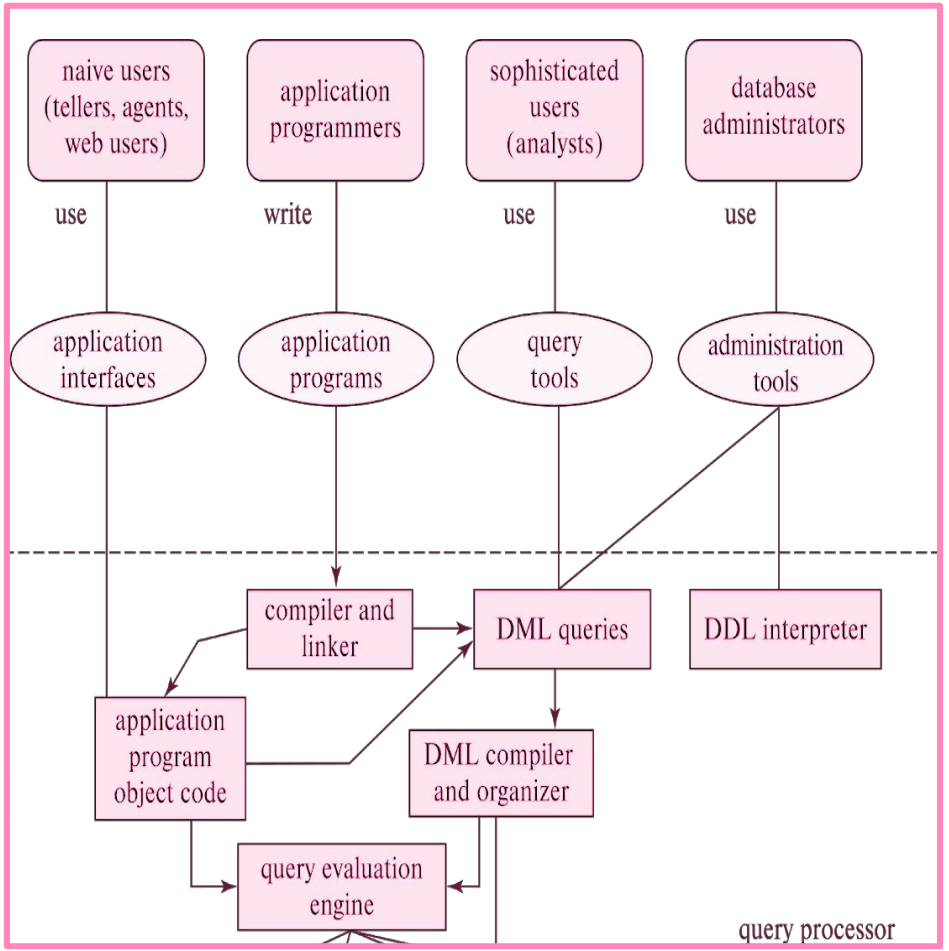
Database Users and Administrators

Seen as a single tier or multi-tier.

Database Users

Types of Database Users

- Naive Users
- Application Users
- Sophisticated Users
- Database Administrators



Database Administrator

A person who has central control over the system is called a database administrator (DBA).

Database Administrators

Functions of a DBA include:

- Schema definition
- Storage structure and access-method definition
- Schema and physical-organization modification
- Granting of authorization for data access
- Routine maintenance
- Periodically backing up the database
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required
- Monitoring jobs running on the database

Thanks!

Does anyone have any queries?

buvann@gmail.com

Credits

- Abraham Silberschatz ,Henry F. Korth and S. Sudarshan,”Database System Concepts”, Seventh Edition, McGRAW Hill Education
- http://slidesgo.com/