DBMS Artitecture

1.What is Abstraction?

-> Abstraction means showing only the important details and hiding unnecessary internal working.

Example of Abstraction?

-> When you use a mobile phone, you press buttons and use apps, but you don’t see the internal circuits or code working behind it.

That is abstraction.

2. Physical Level / Internal Level

Definition:

Describes how data is actually stored in memory (files, indexes, blocks).

Example:

A table's data is stored in the form of B+ trees, hash files, or compressed blocks inside the disk.

The user never sees this. Only the DBMS handles it.

3. Logical Level / Conceptual Level (WHAT data is stored)

Definition:

Describes what data the database contains and how tables are related.

Example:

A database schema like:

Student(id, name, age, department)

Department(dept_id, dept_name)

This level tells us tables and relationships, but not how data is stored physically.

4. View Level / External Level (User-specific view)

Definition:

Shows only required data to each user; hides the rest of the database.

Example:

A student only sees his marks and profile.

A teacher sees student marks but not salary details.

An accountant sees only fee-related data.

Each user gets a different “view”.

5.Schema (Structure of the Database)

Definition:

A schema is the overall design or blueprint of the database.

It describes how the data is organized, what tables exist, what attributes they have, and how they are related.

Schema does not change frequently.

Think of it like:

A building map (structure is fixed).

You don’t change the map every day.

Example:

Database Schema:

Student(id, name, age, department)

Course(course_id, course_name)

Enroll(student_id, course_id)

These tables and their structure represent the schema.

Even if data inside tables changes, the schema remains the same unless we modify the structure.

6. Instance (Current Data Stored in the Database)

Definition:

An instance is the actual data stored in the database at a particular moment.

Instances change frequently when data is inserted, updated, or deleted.

It's like a snapshot of the database at a given time.

Think of it like:

People living inside a building — this can change anytime, even though the building structure (schema) is fixed.

Example:

Current data in Student table:

id | name | age | department

---

1 | Ram | 20 | CSE

2 | Riya | 21 | IT

This is an instance, because it is the current content.

Tomorrow if we add a student:

3 | Aman | 19 | ECE

The instance changes, but the schema remains the same.

7. Data Models (with examples)

a. Definition

A Data Model provides a way to describe the design of a database at the logical level.

b. Purpose

A Data Model gives the conceptual tools to describe:

how data is structured,

how data is related,

what the data means (semantics),

and what rules/constraints must be followed.

It forms the foundation for how the DB is designed.

c. Examples of Data Models

ER Model (Entity–Relationship Model)

Uses entities, attributes, relationships.

Example: Student–Course ER diagram.

1.Relational Model

Represents data in tables (relations).

Example: Student(id, name, age).

2.Object-Oriented Model

Stores data as objects containing both data and methods.

Example: A Student object with properties + functions.

Object–Relational Model

Combines tables with object features.

Example: A relational DB that supports custom types or classes.

8. Database Languages (with examples)

a. Data Definition Language (DDL)

Used to define and modify the database schema.

Example DDL commands:

CREATE → create tables

ALTER → modify tables

DROP → delete tables

b. Data Manipulation Language (DML)

Used to retrieve, insert, update, and delete data stored in the database.

Example DML commands:

SELECT → retrieve data

INSERT → add new data

UPDATE → modify data

DELETE → remove data

c. Practical Use

In real systems, both DDL and DML features are available in one language,

for example: SQL contains both.

d. More on DDL

DDL also defines consistency constraints that must be checked whenever the database is updated.

Example:

PRIMARY KEY,

FOREIGN KEY,

UNIQUE,

NOT NULL.

e. More on DML

i. Data manipulation involves:

Retrieving information → SELECT * FROM Student;

Inserting information → INSERT INTO Student VALUES (1, 'Ram', 20);

Deleting information → DELETE FROM Student WHERE id = 1;

Updating information → UPDATE Student SET age = 21 WHERE id = 1;

ii. Query Language

A part of DML used only for retrieval.

Example: SQL SELECT statements.

9. How is Database Accessed from Application Programs?

a. Application Interaction

Applications written in languages like C, C++, Java, Python, etc., interact with the database to store or retrieve data.

b. Example

A banking system's payroll module written in Java or C++ accesses the database by executing DML statements like:

SELECT to fetch account details

INSERT to store transaction logs

UPDATE to modify balances

The application sends these SQL statements to the DBMS and receives the output.

c. API / Interface Used

Programming languages use APIs to communicate with the database.

Common APIs:

ODBC (Open Database Connectivity)

Used with languages like C, C++

Platform-independent way to send SQL statements to the DB

JDBC (Java Database Connectivity)

Used in Java programs

Allows Java code to connect, send SQL queries, and get results

10. Database Administrator (DBA)

a. Definition

A Database Administrator (DBA) is the person who has central control over the database and the programs that access the data.

They ensure the database runs smoothly, safely, and efficiently.

b. Functions of DBA

i. Schema Definition

DBA creates and manages the database schema (tables, relationships, constraints).

ii. Storage Structure & Access Methods

Decides how data will be stored, indexed, and accessed (e.g., B+ trees, hashing).

iii. Schema & Physical Modifications

Makes changes when needed such as adding new tables, altering table structures, or reorganizing storage.

iv. Authorization Control

Controls who can access, modify, or view data.

(Manages roles, permissions, login accounts.)

v. Routine Maintenance

Periodic Backups

Ensures data can be recovered in case of failure.

Security Patches

Applies updates to protect the DB from threats.

Upgrades

Installs new versions or improves performance as required.

11. DBMS Application Architectures

When we use a database, there are:

Client machines → where users work

Server machines → where the DBMS runs

Based on how work is divided, DBMS systems follow three main architectures:

1. One-Tier Architecture (T1)

Definition

Also called Single-tier or Standalone Architecture.

The DBMS, application, and user interface all run on the same machine.

Example

A student using Oracle/SQL Server installed on their own laptop.

Local MS Access database.

Use Case

Used for learning, testing, or personal systems.

2. Two-Tier Architecture (T2) — Client–Server

Definition

The application is split into client and server.

Client sends SQL queries; the DBMS (server) processes them.

Structure

Client (UI + Application Logic) ⇔ Server (DBMS + Database)

Example

A company payroll application where:

Frontend made in Java/Python/C# is on client computers.

Database (MySQL/Oracle/SQL Server) is on a server machine.

Use Case

Banking systems, office applications, medium-size organizations.

3. Three-Tier Architecture (T3) — Most Modern

Definition

Has three layers:

Presentation Layer (Client/UI)

Application/Business Logic Layer (Middle tier)

Database Layer (DB Server)

Structure

Client (UI) → Application Server (Business Logic, APIs) → Database Server

Example

Web applications:

Frontend: React, Angular, HTML/CSS/JS

Backend: Java, Node.js, Python, .NET

Database: MySQL, MongoDB, PostgreSQL

Use Case

E-commerce, social media apps, large-scale websites, enterprise systems.

Summary Table

Architecture Layers Example

1-Tier (T1) All in one system SQL installed on same PC

2-Tier (T2) Client + DB Server ATM system, office apps

3-Tier (T3) UI + App Server + DB Server Amazon, Flipkart, banking apps