LET'S LEARN

DATABASE MANAGEMENT SYSTEM (DBMS)

-By Riti Kumari

TOPICS TO BE COVERED

1. DBMS Introduction
2. DBMS Architecture
3. Data Abstraction
4. Types of data model
5. ER Model
6. Relational Model
7. Type of keys
8. Normalisation
9. Denormalization
10. Transactions & Concurrency control
11. Indexing(B ,B+ trees)
12. SQL

DATA & INFORMATION

Data : Data refers to a collection of raw facts or figures that can be processed to derive meaning or knowledge. In easier terms we can say any fact that can be stored. *Ex- XYZ, 12*

Information : Processed data is called information.
Ex- Your name, temparature, etc.

DATABASE

Collection of interrelated data is called a database.

- It can be stored in the form of table
- It can be any size

Multimedia database

College database

EXAMPLE

ID	Name	subject
1	Rahul	PHE
2	Raj	ECO
3	Riti	IT

ID	Name	Place
1	Rahul	DELHI
2	Raj	KOLKATA
3	Riti	MUMBAI

FILE SYSTEM

An operating system's approach for organising and storing data on storage units like hard drives is called a file system.
In a file system, data is organised into files.

The major disadvantage of file system is

- Data redundancy
- Poor Memory utilisation
- Data inconsistency
- Data security

DATABASE MANAGEMENT SYSTEM

The acronym DBMS stands for **"Database Management System."**

Users can access databases, save data, retrieve it, update it, and manage it safely and effectively with the use of a software program or combination of programs.

The presence of rules and regulations in the management system is crucial as they are necessary to uphold and maintain the database effectively.

APPLICATION OF DBMS

- Schools and Colleges

- Banks

- Airlines

APPLICATION OF DBMS

- **Schools and Colleges** - DBMS is used to create and maintain a student information system that stores student records, including personal details, academic performance, attendance, and extracurricular activities.

- **Banks** - DBMS is used to maintain a centralised and secure database of customer information, including personal details, account numbers, contact information, and transaction history.

TYPES OF DATABASES

1. Relational Databases (RDBMS)
2. NoSQL Databases
3. Object-Oriented Databases
4. In-Memory Databases
5. Time-Series Databases
6. Spatial Databases
7. Multimedia Databases
8. Columnar Databases
9. XML Databases
10. NewSQL Databases
11. Blockchain Databases

SQL

TYPES OF DATABASES

- **Relational Databases (RDBMS)**- These databases structure data into organized tables that have predefined connections between them. Data manipulation and querying are performed using SQL (Structured Query Language). Well-known instances encompass MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server.

- **NoSQL Databases**- NoSQL databases are created to handle data that doesn't fit neatly into the strict setup of traditional relational databases. Ex- MongoDB(Document Oriented DB)

TYPES OF DATABASES

- **Object-Oriented Databases**- These databases hold objects (data and actions) utilized in object-oriented programming. They work well for applications with intricate data designs, like scientific simulations or multimedia software.

- **In-Memory Databases**- In these databases, data is kept in the primary memory (RAM) rather than on a disk, leading to quicker data retrieval. They're employed in applications that demand instant data processing and top-notch performance.

NEED OF DBMS

DBMS plays a vital role for businesses, institutions, and organizations of all scales in effectively managing their data, ensuring data accuracy and security, and supporting essential decision-making processes.

It serves as the core of contemporary information systems, facilitating efficient data management and serving as a basis for a wide range of applications and services.

ADVANTAGE OF DBMS

- **Data Security**- DBMS implements security mechanisms that regulate access to sensitive information, safeguarding it from unauthorized access and potential data breaches.

- **Data Redundancy and Inconsistency** - DBMS removes data redundancy, minimizing storage needs and ensuring consistency through the maintenance of a unified version of the data.

- **Data Integrity** - DBMS guarantees data integrity by enforcing rules and constraints that prohibit the entry of incorrect or inconsistent data into the database.

ADVANTAGE OF DBMS

- **Data Scalability**- DBMS can handle large datasets and scale to accommodate increasing amounts of data as an organization grows.

- **Data Abstraction**- DBMS offers data abstraction, allowing users and applications to interact with the database without needing to understand its underlying complexities.

DISADVANTAGE OF DBMS

- **Cost**- Acquiring, deploying, and sustaining DBMS software can incur significant costs. Furthermore, the hardware essential for the proficient operation of a DBMS can also lead to substantial expenses.

- **Scale Projects**- When dealing with modest applications and minimal data storage requirements, adopting a comprehensive DBMS could introduce avoidable intricacies and additional burdens. In these instances, more streamlined data storage alternatives could be better suited.

DISADVANTAGE OF DBMS

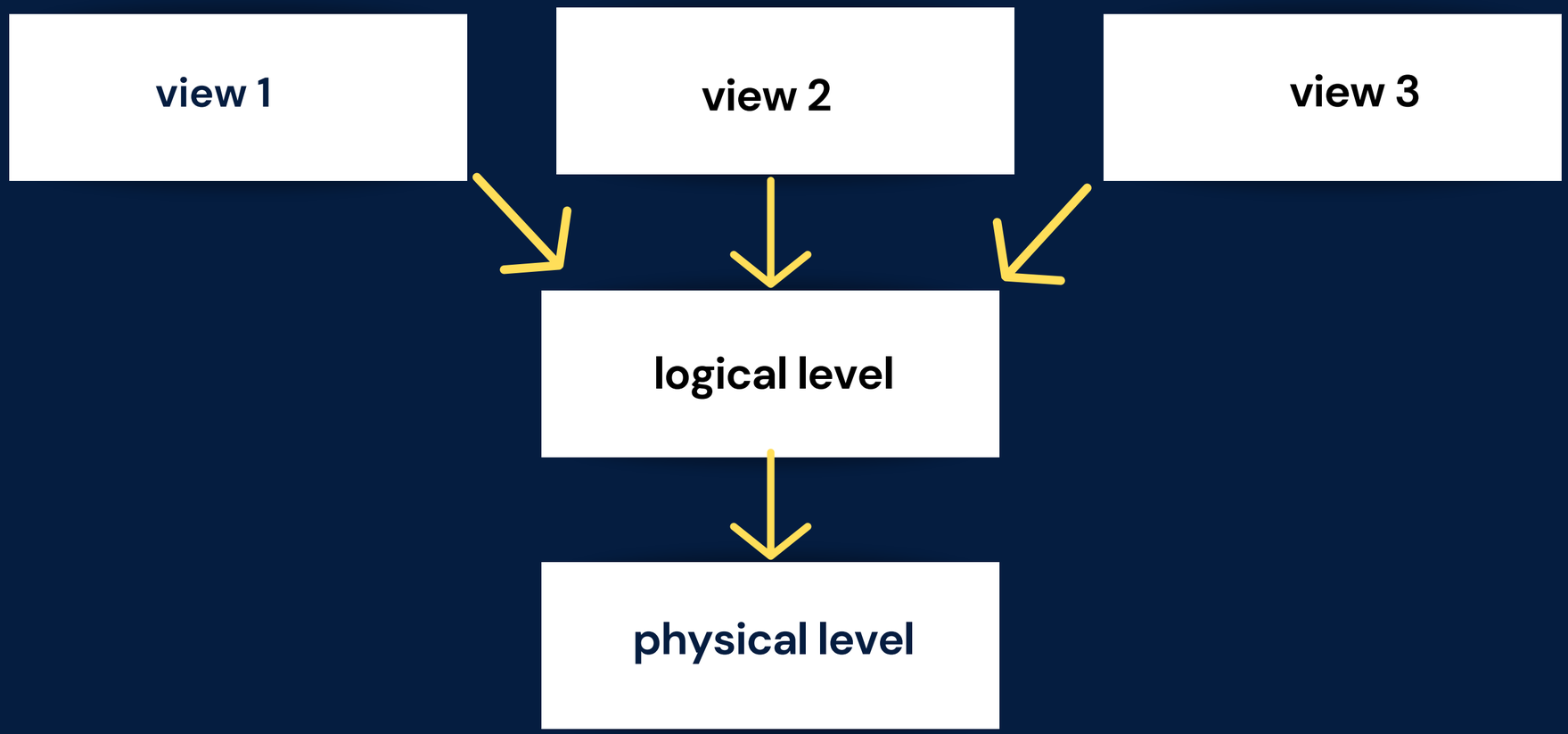
- Vendor Lock-In- Once you've chosen a specific DBMS, it can be challenging to switch to a different one due to differences in data formats, query languages, and other technical aspects. This can lead to vendor lock-in, where you are dependent on a particular vendor's technology and pricing.

DATA ABSTRACTION

Database systems are built with complex ways of organizing data. To make it easier for people to use the database, the creators hide the complicated stuff that users don't need to worry about. This hiding of unnecessary things from users is called data abstraction.

DATA ABSTRACTION

There are three levels of Abstraction



TYPES OF LEVEL

- **Physical level-** This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.

- **Logical level-** This is the middle level of 3-level data abstraction architecture. It describes what data is stored in database.

- **View level-** Highest level of data abstraction. This level describes the user interaction with database system.

SCHEMA & INSTANCE

Let us first learn about some basic concepts:

Schema- A schema is a logical container or structure that organizes and defines the structure of a database.

It defines how data is organized, what data types are used, what constraints are applied, and the relationships between different pieces of data. A schema acts as a blueprint for the database, ensuring data integrity, consistency, and efficient data retrieval.

SCHEMA & INSTANCE

Types of Schema :

1. **Physical Schema–** A physical schema defines how data is stored on the underlying hardware, including details such as storage format, file organization, indexing methods, and data placement.

SCHEMA & INSTANCE

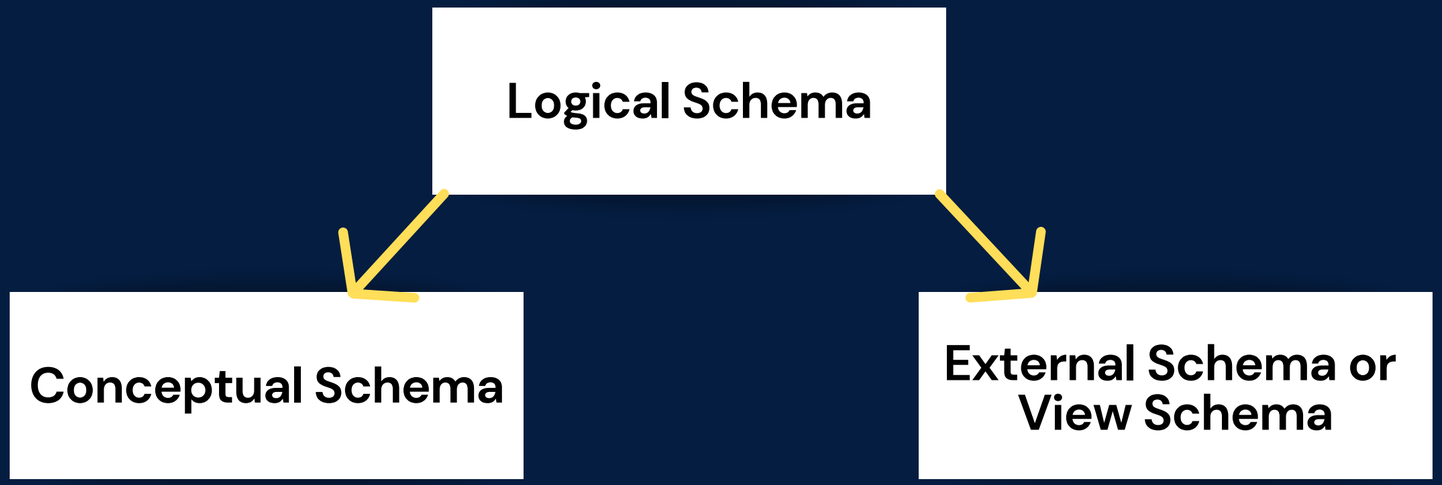
Characteristics of **Physical Schema** :

- Its primary focus lies in enhancing the storage and retrieval of data to boost performance.
- Modifications made to the physical schema demand meticulous planning and can potentially affect the overall performance of the database.
- Example: Deciding to use clustered indexes on specific columns for faster retrieval.

SCHEMA & INSTANCE

Types of Schema :

2. **Logical Schema-** A logical schema defines the database's structure from a logical or conceptual perspective, without considering how the data is physically stored.



SCHEMA & INSTANCE

Types of Logical Schema :

- **Conceptual Schema:** The conceptual schema represents the overall view of the entire database. It defines the high-level structure and relationships between all data elements.

- **External/View Schema:** An external schema defines the user-specific views of the database. It focuses on the portions of the database that are relevant to specific user roles or applications.

SCHEMA & INSTANCE

Characteristics of **Logical Schema** :

- It delineates how data is structured into tables, the interconnections between these tables, and the restrictions placed on the data.
- Logical schemas prioritize data modeling and database design over considerations related to hardware or storage specifics.
- Example: Defining tables, specifying primary and foreign keys, and creating views for data access.

SCHEMA & INSTANCE

Instance - The information residing within a database at a specific point in time is referred to as the database's "instance."

Within a given database schema, the declarations of variables within its tables pertain to that specific database. The term "instance" in this context denotes the current values of these variables at a particular moment in time for that database.

DBMS ARCHITECTURE

Database Management System architecture, refers to the structural framework and organization of a database management system. It defines how the various components of the system work together to store, manage, and retrieve data efficiently.

DBMS ARCHITECTURE

Types of DBMS ARCHITECTURE :

There are several types of DBMS Architecture.
Choice of architecture depends on factors such as the type of database (e.g., relational, NoSQL) and the specific needs of an application.

- **1-Tier Architecture**
- **2-Tier Architecture**
- **3-Tier Architecture**

DBMS ARCHITECTURE

1-Tier Architecture - In 1 tier architecture the entire database application, including the user interface, application logic, and data storage, resides on a single machine or computer.

ex- An illustration of a straightforward single-tier architecture can be seen when you install a database on your system and use it to practice SQL queries.

DBMS ARCHITECTURE

2-Tier Architecture - In 2 Tier Architecture the presentation layer runs on a client (PC, Mobile, Tablet, etc.), and data is stored on a server.

Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly. It also provides direct and faster communication.

DBMS ARCHITECTURE

2-Tier Architecture

Client (User/Application)

Server(db)

DBMS ARCHITECTURE

- **3-Tier Architecture** - It separates the application into three logically distinct layers presentation, application, and data layer

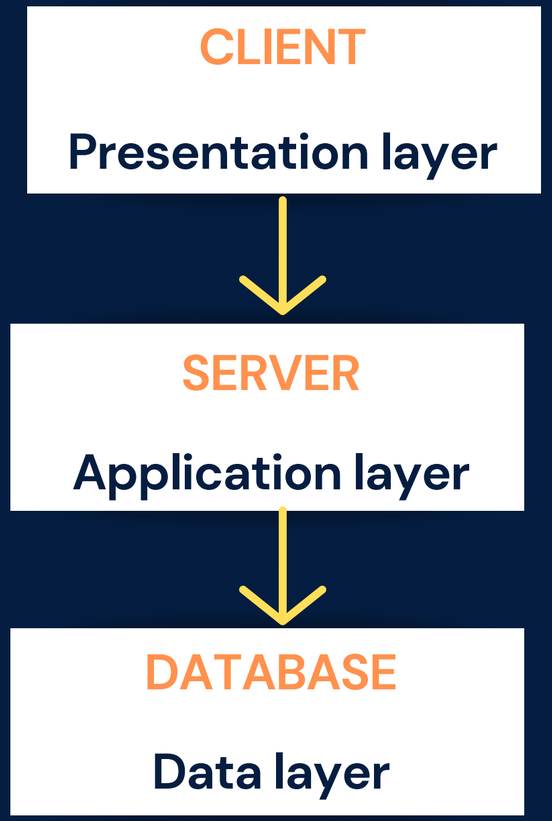
- **Presentation layer**- It handles the user interface.
  ex- your PC, Tablet, Mobile, etc

- **Application layer** - It manages business logic
  ex- server

- **Data layer**- It manages data storage and processing.
  ex- Database Server

DBMS ARCHITECTURE

- 3-Tier Architecture



DBMS ARCHITECTURE

Advantages of 3-tier-architecture

- Scalability: Easily adjust each tier to handle changing user demands.
- Modularity and Maintainability: Simplify maintenance by separating responsibilities.
- Security: Protect sensitive data with an additional layer.
- Performance: Optimize presentation and application tiers for better performance.

Disadvantages of 3-tier-architecture

- The disadvantages of 3-Tier Architecture include increased complexity, potential latency issues, longer development time, resource overhead, and the possibility of bottlenecks.