DBMS

Database:

A database is a structured collection of data organized for efficient storage, retrieval, and management, commonly used in various applications.

DBMS:

A DBMS (Database Management System) is software that manages and organizes databases, making it easier to store, access, and manipulate data.

Disadvantages in File Processing:

- Data redundancy and inconsistency.
- Difficult in accessing data.
- Data isolation.
- Data integrity.
- Concurrent access is not possible.
- Security Problems.

Advantages of DBMS over file system:

- **Data Integrity:** DBMS systems enforce data integrity through constraints and validations, reducing the risk of data corruption and ensuring accurate, consistent data.
- **Data Security:** DBMS provides access control and authentication mechanisms, allowing you to restrict access to sensitive data, enhancing security.
- **Data Redundancy Reduction:** DBMS minimizes data redundancy, reducing storage requirements and the chance of inconsistencies in duplicate data.
- **Data Reporting:** DBMS allows you to generate reports and perform data analytics, making it easier to gain insights from the data.
- **Data Backups:** DBMS provides mechanisms for data backup and recovery, reducing the risk of data loss.
- **Data Maintenance:** DBMS automates tasks like data maintenance, such as indexing and optimizing data storage.

Application of DBMS: (Explain any two)

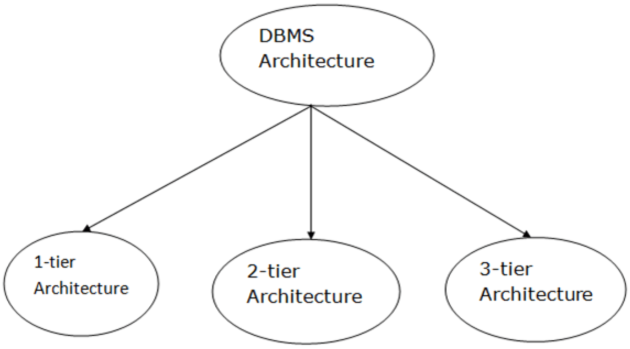
- **Business and Finance**
- **Healthcare:**
- **E-commerce:**
- **Education:**
- **Government:**
- **Telecommunications:**
- **Library Systems:**
- **Restaurant Management System:**
- **Banking:**

- **Online Booking and Reservation Systems:**

Architecture:

A database management system (DBMS) architecture **describes how users interact with a database system**. It helps with the design, implementation, and maintenance of a database.

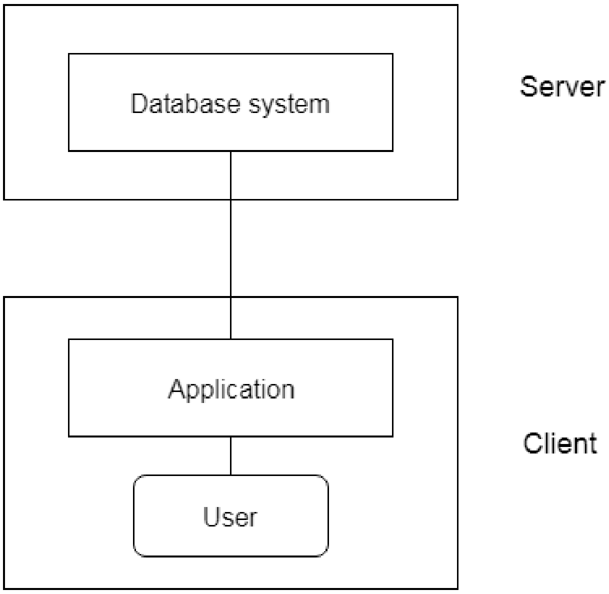
Types of DBMS Architecture



1-Tier Architecture

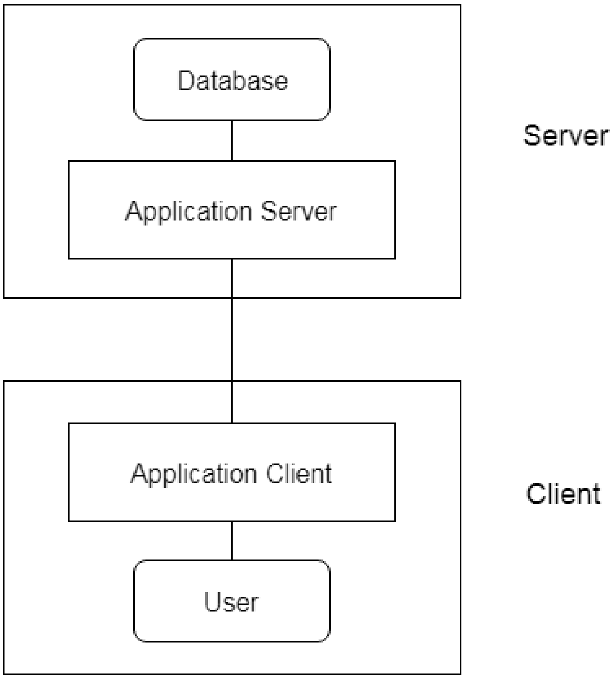
- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-Tier Architecture



- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.

3-Tier Architecture



- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- The 3-Tier architecture is used in case of large web application.

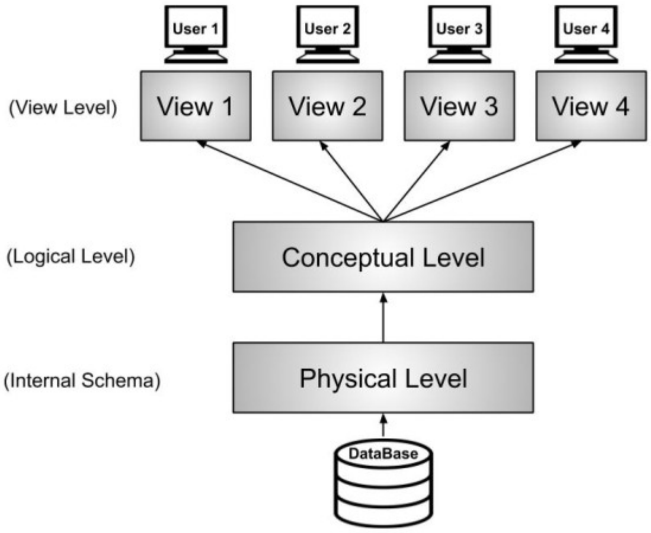
Data Abstraction in DBMS

Data abstraction is **the process of providing only essential information about data to the outside world, while hiding the background details or implementation**

There are three levels of data abstraction in DBMS:

- **Physical or internal level:**
    - This is the lowest level of data abstraction.

    - Defines how the data is stored using various data structures.
    - You can get the complex data structure details at this level.
- **Logical or conceptual level:**
    - This is the middle level of 3-level data abstraction architecture.
    - It describes what data is stored in database.
    - Describes the relationship between the stored data
- **View or external level:**
    - Highest level of data abstraction.
    - This level describes the user interaction with database system.
    - Provides a high-level view of a section of data



Levels of Data Abstraction

Example:

Let’s say we are storing customer information in a customer table.

At **physical level** these records can be described as blocks of storage (bytes, gigabytes, terabytes etc.) in memory. These details are often hidden from the programmers.

At the **logical level** these records can be described as fields and attributes along with their data types, their relationship among each other can be logically implemented. The programmers generally work at this level because they are aware of such things about database systems.

At **view level**, user just interact with system with the help of GUI and enter the details at the screen, they are not aware of how the data is stored and what data is stored; such details are hidden from them.