**Schema ,Instance and Schema Architecture**

**Schema:** A schema represents the overall design and layout of the database objects, such as tables, views, indexes, and constraints. A schema is a logical container or blueprint that defines the structure, organization, and relationships of a database. A schema defines the schema objects and their attributes, data types, and relationships. It provides a framework for organizing and representing data in a consistent and structured manner.

**Instance:** It represents the actual data stored in the database system, including the content of tables, indexes, and other database objects. Each running database system has its own instance, which includes the memory structures and processes needed to manage and manipulate the data.

**Schema Architecture:** Schema architecture refers to the design and organization of schemas within a database system. It determines how schemas are structured, related, and accessed. There are different types of schema architectures, including:

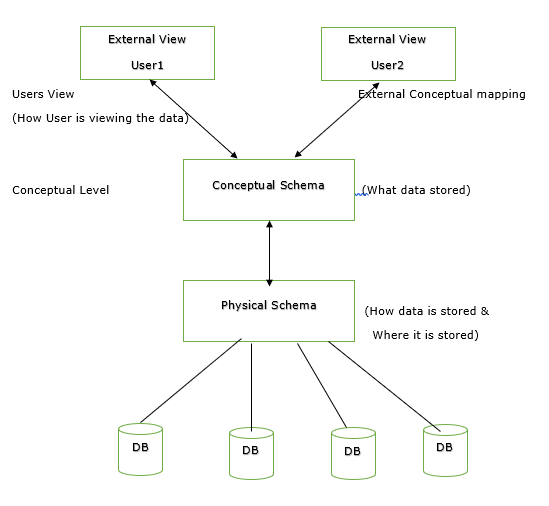
1. *Single-schema architecture*: In this architecture, there is only one schema that encompasses all the database objects. It is commonly used in small-scale applications or systems where there is a single logical unit of data.
2. *Three-schema architecture*: The three-schema architecture divides the database into three-level used to create a separation between the physical database and the user application. In simple terms, this architecture hides the details of physical storage from the user.

This architecture contains three layers of database management system, which are as follows −

1. External level

2. Conceptual level

3. Internal level



**Database System Architecture**

The database system architecture can be broadly classified into two main types: two-tier architecture and three-tier architecture (ANSI-SPARC architecture). Let's take a look at each of them:

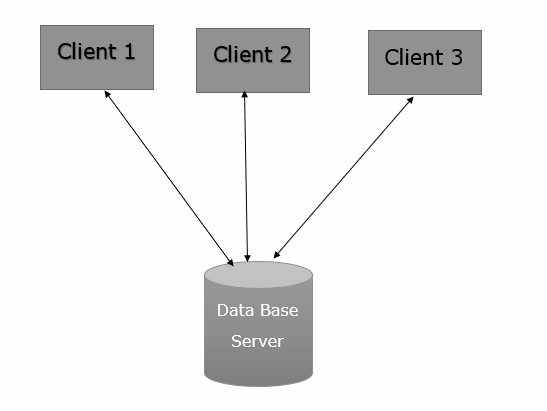
**1. Two-tier Architecture:**

The two-tier architecture, also known as the client-server architecture, consists of two layers: the client layer and the server layer. In this architecture, the client layer directly communicates with the database server.

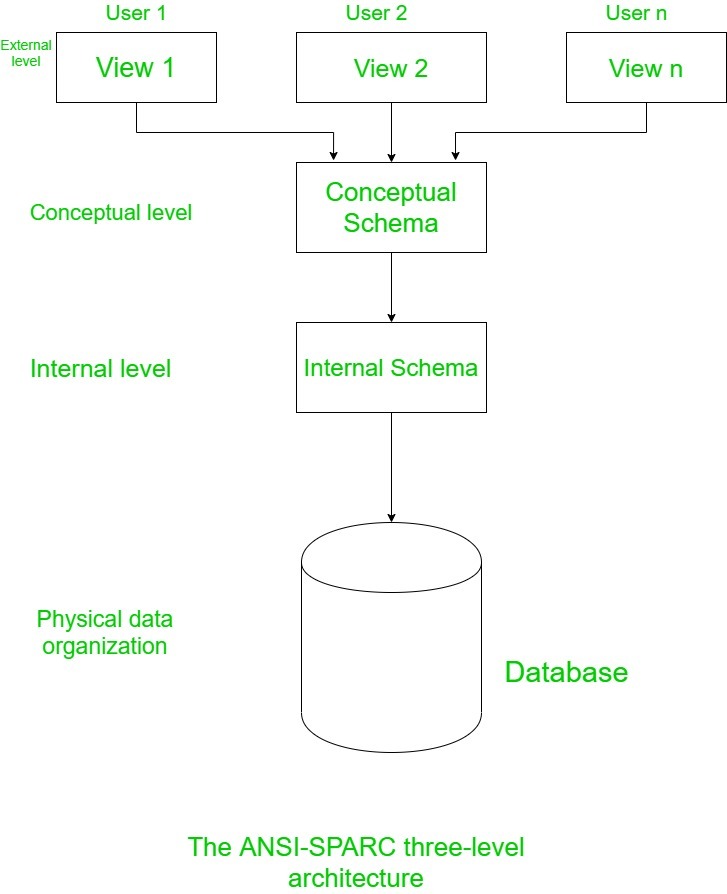
The client layer is responsible for user interaction, such as displaying the user interface, collecting user input, and performing data validation. It handles the presentation logic and user interface components.

The server layer, also known as the database server, manages the database and handles tasks related to data storage, retrieval, and manipulation. It interacts directly with the database management system (DBMS) to process queries, execute transactions, and manage data integrity and security.

In the two-tier architecture, both the application logic and data access logic reside on the client side. This architecture is relatively simple but can suffer from issues such as scalability and maintainability, as all the processing tasks are handled on the client side.



**2. Three Level ANSI-SPARC Architecture:**



The ANSI-SPARC (American National Standards Institute, Standards Planning And Requirements Committee) architecture is a widely accepted framework for database system design. It defines three levels of abstraction: external level, conceptual level, and internal level.

- External Level: This level represents the user views or user interfaces. It describes how each user sees the data and defines the individual user's perception of the database. Each user can have a different external view tailored to their specific needs, hiding unnecessary details of the database schema.

- Conceptual Level: The conceptual level represents the overall logical view of the entire database for the entire organization. It describes the entire database schema, including all entities, attributes, and their relationships. It provides a global, integrated view of the database that is independent of any specific application.

- Internal Level: The internal level, also known as the physical level, deals with the physical representation of data in the database. It describes how the data is stored and accessed at the lowest level, such as data structures, indexes, and storage organization. It focuses on optimization techniques for efficient storage and retrieval of data.

The ANSI-SPARC architecture separates the user views (external level) from the overall logical structure (conceptual level) and the physical storage details (internal level). This separation provides flexibility, independence, and data abstraction, allowing changes in one level without affecting the other levels.

By using the ANSI-SPARC architecture, database designers can modify or enhance the database schema, optimize storage structures, or change user interfaces without affecting the entire system, leading to better scalability, maintainability, and adaptability.

It's important to note that the ANSI-SPARC architecture is a conceptual framework and does not dictate a specific implementation or deployment strategy. Actual database management systems (DBMS) may vary in their implementation while adhering to the fundamental principles of the ANSI-SPARC architecture.

MVC Architecture-

