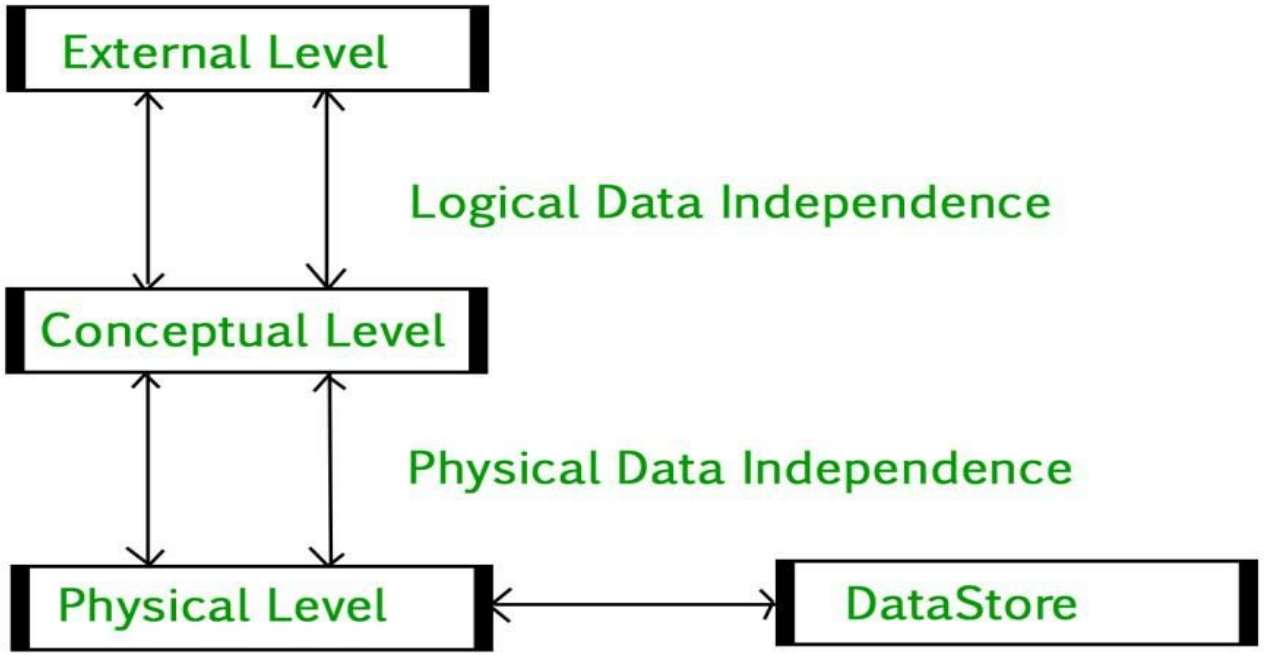
Introduction of 3-Tier Architecture in DBMS

3-Tier Architecture in DBMS

The 3-tier architecture divides an application's components into three tiers or layers. Each layer has its own set of responsibilities.

DBMS 3-Tier architecture divides the complete system into three inter-related but independent modules.



3-Tier Architecture in DBMS

Physical Level: Physical level of a database describes how the data is being stored in secondary storage devices like disks and tapes and also gives insights on additional storage details.

Conceptual Level: At conceptual level, data is represented in the form of various database tables. For Example, STUDENT database may contain STUDENT and COURSE tables which will be visible to users but users are unaware of their storage. Also referred as logical schema, it describes what kind of data is to be stored in the database.

External Level: An external level specifies a view of the data in terms of conceptual level tables. Each external level view is used to cater to the needs of a particular category of users. For Example, FACULTY of a university is interested in looking course details of students, STUDENTS are interested in looking at all details related to academics, accounts, courses and hostel details as well. So, different views can be generated for different users. The main focus of external level is data abstraction.

Data Independence

Data independence means a change of data at one level should not affect another level. Two types of data independence are present in this architecture:

Physical Data Independence: Any change in the physical location of tables and indexes should not affect the conceptual level or external view of data. This data independence is easy to achieve and implemented by most of the DBMS.

Conceptual Data Independence: The data at conceptual level schema and external level schema must be independent. This means a change in conceptual schema should not affect external schema. e.g.; Adding or deleting attributes of a table should not affect the user's view of the table. But this type of independence is difficult to achieve as compared to physical data independence because the changes in conceptual schema are reflected in the user's view.

3 Tier Schema Architecture in DBMS

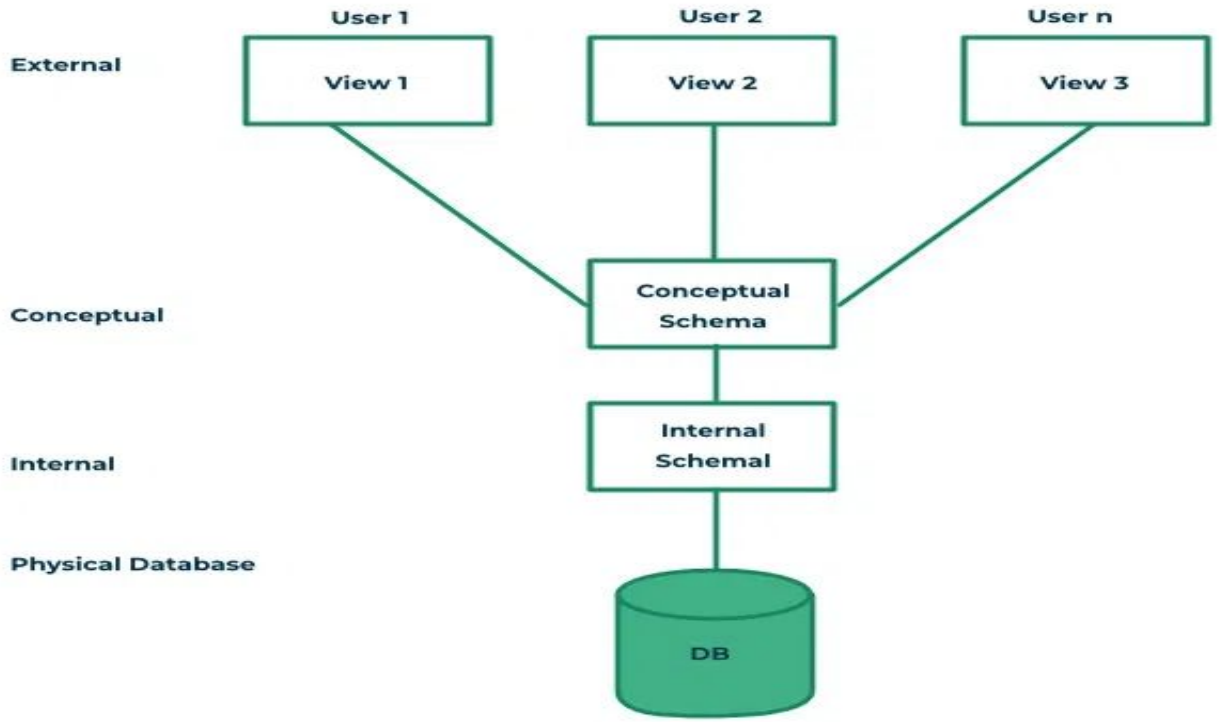
In DBMS, the 3-tier architecture is a client-server architecture that separates the user interface, application processing, and data management into three distinct tiers or layers.

Presentation Tier: The presentation tier is the user interface or client layer of the application. It is responsible for presenting data to the user and receiving input from the user. This tier can be a web browser, mobile app, or desktop application.

Application Tier: The application tier is the middle layer of the 3-tier architecture. It is responsible for processing and managing the business logic of the application. This tier communicates with the presentation tier to receive user input and communicates with the data management tier to retrieve or store data. This tier may include application servers, web servers, or APIs.

Data Management Tier: The data management tier is the bottom layer of the 3-tier architecture. It is responsible for managing and storing data. This tier can include databases, data warehouses, or data lakes. The data management tier communicates with the application tier to receive or store data.

3 Tier Schema Architecture in DBMS



Benefits of 3-Tier Architecture

The 3-tier architecture in DBMS provides several benefits, including:

Scalability: The architecture separates the application processing and data management layers, which allows for easy scalability of each layer independently.

Flexibility: The architecture allows for the replacement or upgrade of one layer without affecting the other layers.

Security: The architecture provides an additional layer of security, as the data management tier can be isolated from the application and presentation tiers, reducing the risk of unauthorized access.

Thank you