DBMS Architectures

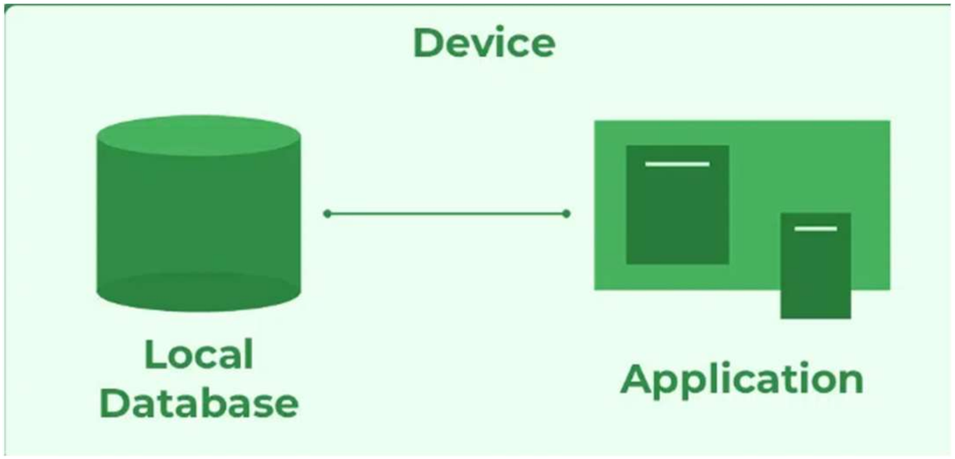
(1-tier, 2-tier, 3-tier)

Types of DBMS Architectures

- There are several types of DBMS Architecture that we use according to the usage requirements. Types of DBMS Architecture are discussed here.
- 1-Tier Architecture
- 2-Tier Architecture
- 3-Tier Architecture

1- Tier Architecture

- In 1-Tier Architecture the **database is directly available to the user**, the user can directly sit on the DBMS and use it that is, the client, server, and Database are all present on the same machine
- **For Example:** to learn SQL we set up an SQL server and the database on the local system. This enables us to directly interact with the relational database and execute operations.
- The industry won’t use this architecture they logically go for 2-Tier and 3-Tier Architecture.

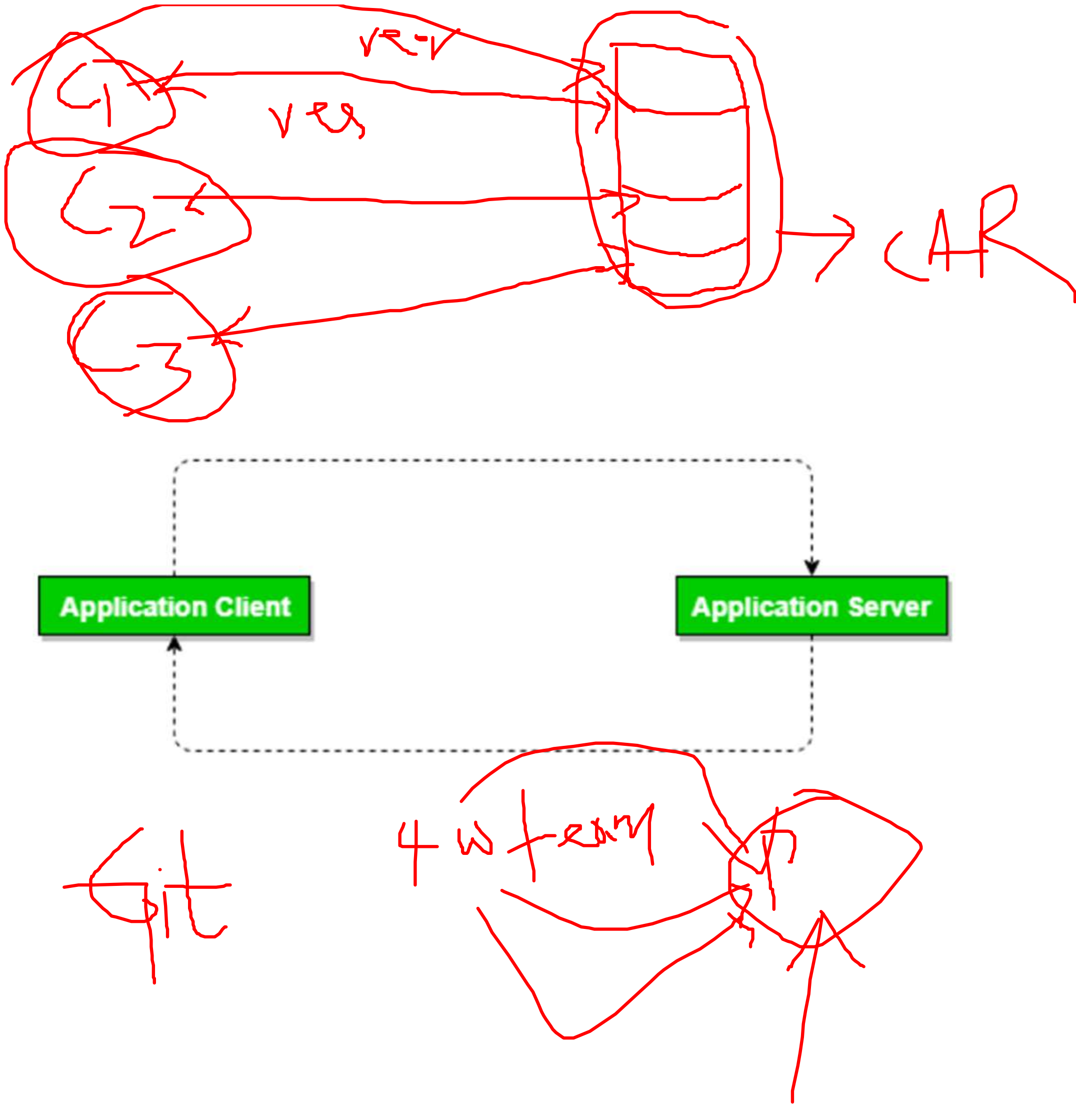


Advantages of 1-Tier Architecture

- **Simple Architecture:** 1-Tier Architecture is the most simple architecture to set up, as only a single machine is required to maintain it.
- **Cost-Effective:** No additional hardware is required for implementing 1-Tier Architecture, which makes it cost-effective.
- **Easy to Implement:** 1-Tier Architecture can be easily deployed, and hence it is mostly used in small projects.

2-Tier Architecture

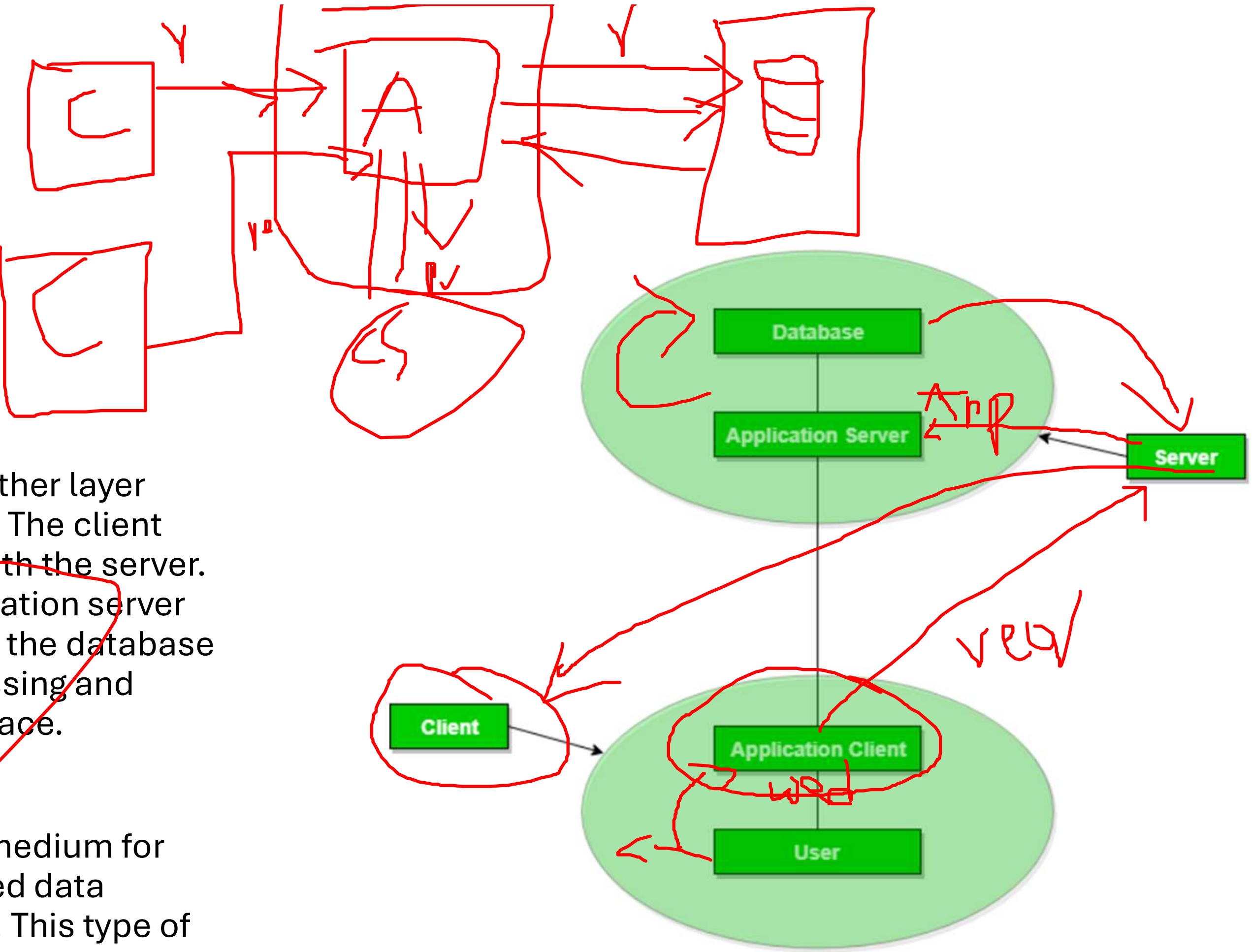
- The 2-tier architecture is similar to a basic client-server model. The application at the client end directly communicates with the database on the server side. APIs like ODBC and JDBC are used for this interaction.
- The server side is responsible for providing query processing and transaction management functionalities
- On the client side, the user interfaces and application programs are run.



Advantages of 2-Tier Architecture

- **Easy to Access:** 2-Tier Architecture makes easy access to the database, which makes fast retrieval.
- **Scalable:** We can scale the database easily, by adding clients or by upgrading hardware.
- **Low Cost:** 2-Tier Architecture is cheaper than 3-Tier Architecture and Multi-Tier Architecture.
- **Easy Deployment:** 2-Tier Architecture is easy to deploy than 3-Tier Architecture.
- **Simple:** 2-Tier Architecture is easily understandable as well as simple because of only two components.

3-Tier Architecture



- In 3-Tier Architecture, there is another layer between the client and the server. The client does not directly communicate with the server. Instead, it interacts with an application server which further communicates with the database system and then the query processing and transaction management takes place.

- This intermediate layer acts as a medium for the exchange of partially processed data between the server and the client. This type of architecture is used in the case of large web applications.

Advantages of 3-Tier Architecture

- **Enhanced scalability:** Scalability is enhanced due to distributed deployment of application servers. Now, individual connections need not be made between the client and server.
- **Data Integrity:** 3-Tier Architecture maintains Data Integrity. Since there is a middle layer between the client and the server, data corruption can be avoided/removed.
- **Security:** 3-Tier Architecture Improves Security. This type of model prevents direct interaction of the client with the server thereby reducing access to unauthorized data.

Disadvantages of 3-Tier Architecture

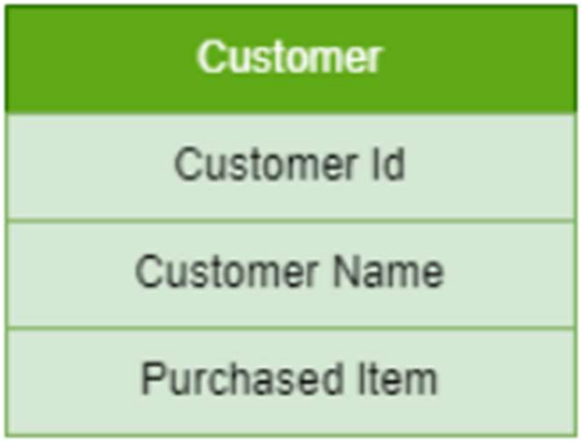
- **More Complex:** 3-Tier Architecture is more complex in comparison to 2-Tier Architecture. Communication Points are also doubled in 3-Tier Architecture.
- **Difficult to Interact:** It becomes difficult for this sort of interaction to take place due to the presence of middle layers.
- **Mark as Read**
- **Report An Issue**

Database Schema representation

Understanding Schema

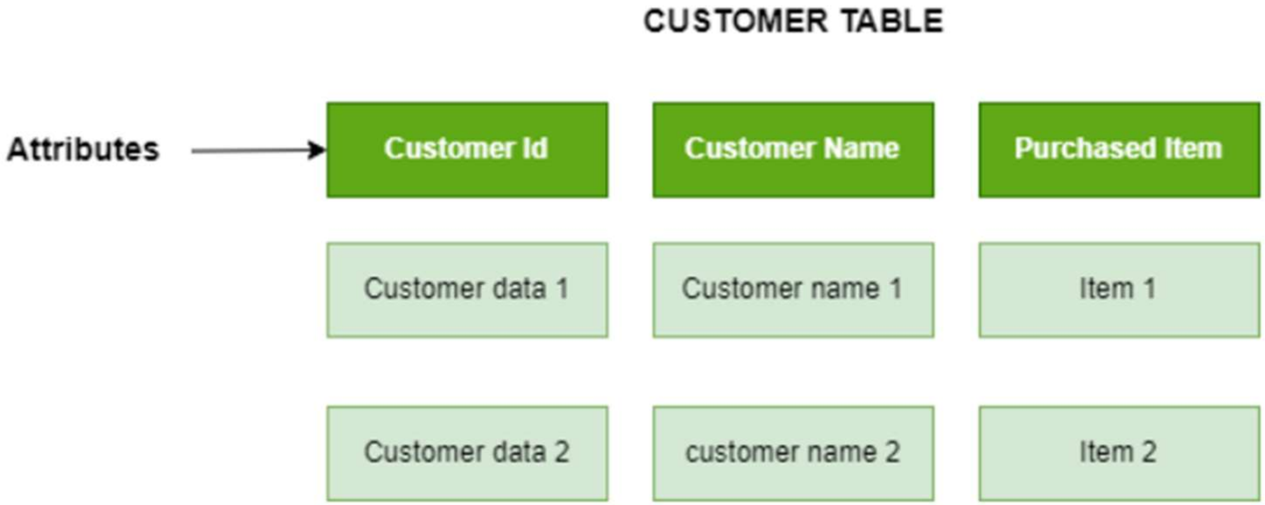
What is Schema?

- The Skeleton of the database is created by the attributes and this skeleton is named as Schema.
- Schema is mentioning the logical constraints like table, primary key etc.
- Schema does not represent the data type of the attributes.



Customer
Customer Id
Customer Name
Purchased Item

Details of a Customer



CUSTOMER TABLE

Attributes

Customer Id	Customer Name	Purchased Item
Customer data 1	Customer name 1	Item 1
Customer data 2	customer name 2	Item 2

Schema of Customer

Database Schema

- A database schema is a **logical representation of data** that shows how the data in a database should be stored logically. It shows how the data is organized and the relationship between the tables.
- Database schema contains table, field, views and relation between different keys like primary key, foreign key.
- Data are stored in the form of files which is unstructured in nature which makes accessing the data difficult. Thus to resolve the issue the data are organized in structured way with the help of database schema.
- Database schema provides the organization of data and the relationship between the stored data.
- Database schema defines a set of guidelines that control the database along with that it provides information about the way of accessing and modifying the data.

Types of Database Schemas

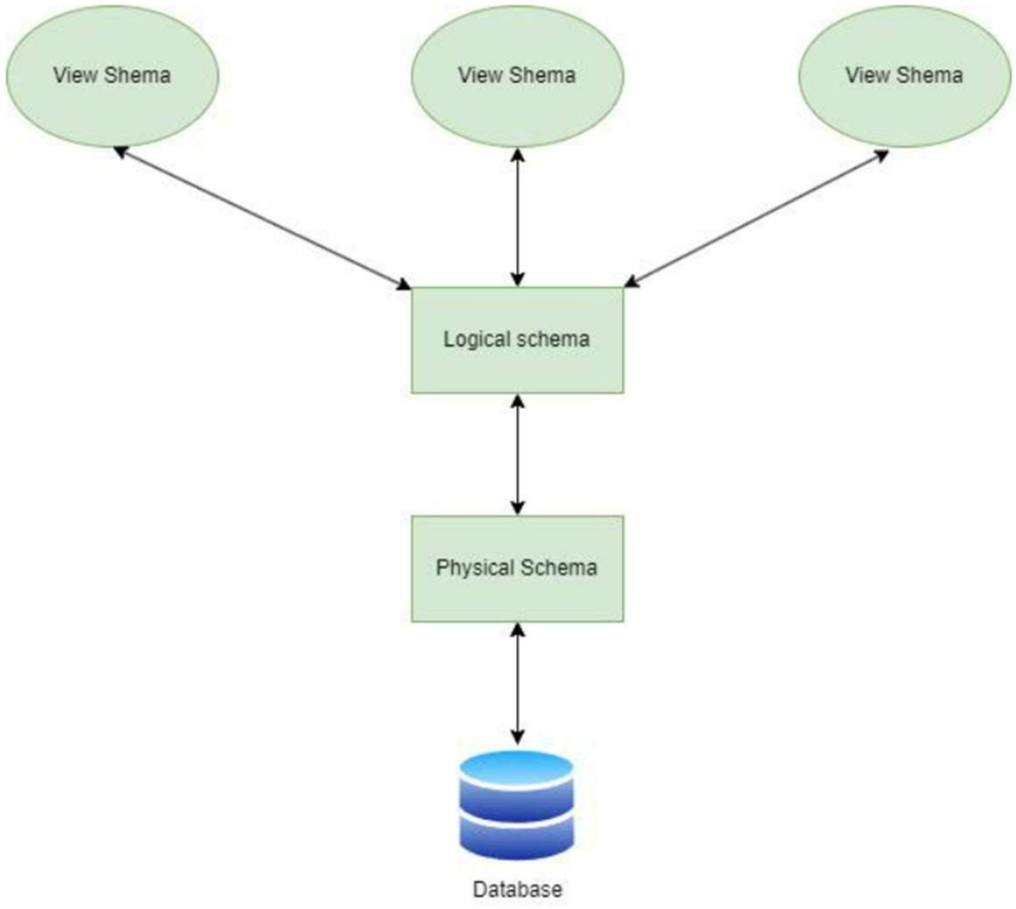
- There are 3 types of database schema:
- **Physical Database Schema:**
  - A Physical schema defines, how the data or information is stored physically in the storage systems in the form of files & indices. This is the actual code or syntax needed to create the structure of a database, we can say that when we design a database at a physical level, it’s called physical schema.
  - The Database administrator chooses where and how to store the data in the different blocks of storage.

Logical Database Schema:

- A logical database schema defines all the logical constraints that need to be applied to the stored data, and also describes tables, views, entity relationships, and integrity constraints.
- The Logical schema describes how the data is stored in the form of tables & how the attributes of a table are connected.
- Using **ER modelling** the relationship between the components of the data is maintained.
- In logical schema different integrity constraints are defined in order to maintain the quality of insertion and update the data.

View Database Schema:

- It is a view level design which is able to define the interaction between end-user and database.
- User is able to interact with the database with the help of the interface without knowing much about the stored mechanism of data in database.



Three layer Schema Diagram