جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الانظمة الطبية الذكية

المرحلة الثانية

Fourth Lecture

Subject: Database Systems

Class: Second

Lecturers: Dr. Dunia Hamid Hameed, M.Sc. Qusai Al-Durrah

Lecture 4: Database Systems

Lecture Title: DBMS Architecture

Table of Contents

DBMS Architecture 1-level, 2-Level, 3-Level

A Database stores a lot of critical information to access data quickly and securely. Hence it is important to select the correct architecture for efficient data management. Database Management System (DBMS) architecture is crucial for efficient data management and system performance. It helps users to get their requests done while connecting to the database. It focuses on how the database is designed, built and maintained, shaping how users access and interact with it.

Types of DBMS Architecture

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

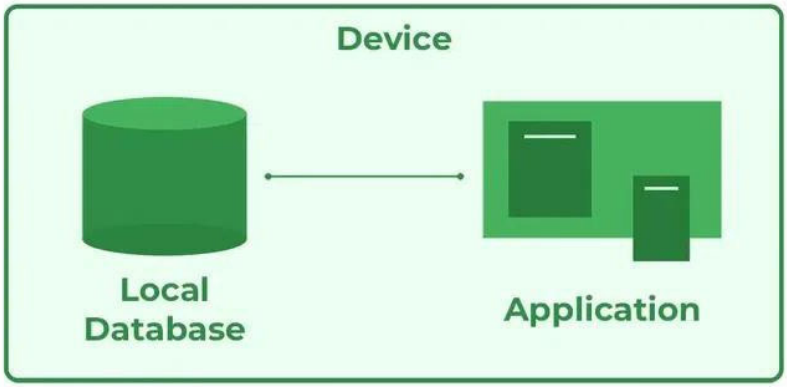
1-Tier Architecture

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. This setup is simple and is often used in personal or standalone applications where the user interacts directly with the database.

For Example: A Microsoft Excel spreadsheet is a great example of one-tier architecture.

- Everything—the user interface, application logic and data is handled on a single system.
- The user directly interacts with the application, performs operations like calculations or data entry and stores data locally on the same machine.

This architecture is simple and works well for personal, standalone applications where no external server or network connection is needed.



DBMS 1-Tier Architecture

Advantages of 1-Tier Architecture

Below mentioned are the advantages of 1-Tier Architecture.

- **Simple Architecture:** 1-Tier Architecture is the simplest 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

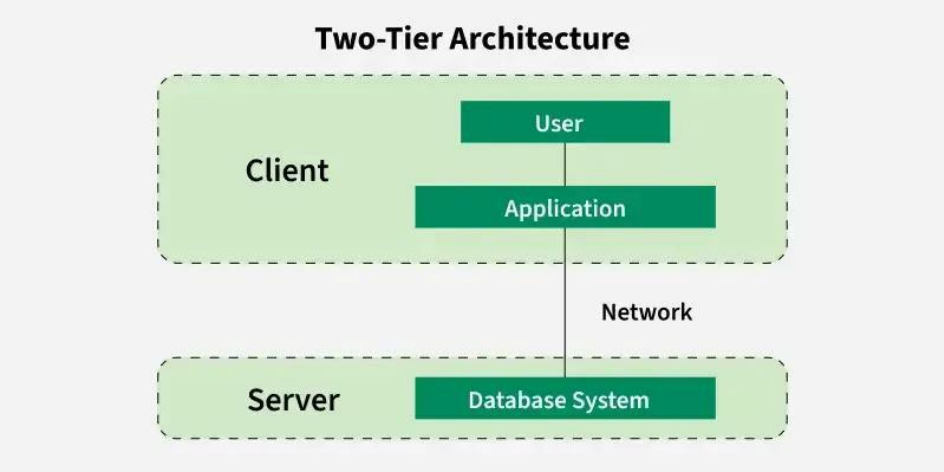
The 2-tier architecture is similar to a basic client-server model. The application at the client end directly communicates with the database on the server side. APIs like ODBC and JDBC are used for this interaction. The server side is responsible for providing query processing and transaction management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

For Example: A Library Management System used in schools or small organizations is a classic example of two-tier architecture.

1. **Client Layer (Tier 1):** This is the user interface that library staff or users interact with. For example they might use a desktop application to search for books, issue them, or check due dates.

2. **Database Layer (Tier 2):** The database server stores all the library records such as book details, user information, and transaction logs.

The client layer sends a request (like searching for a book) to the database layer which processes it and sends back the result. This separation allows the client to focus on the user interface, while the server handles data storage and retrieval.



DBMS 2-Tier Architecture

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 upgrading hardware.
- Low Cost: 2-Tier Architecture is cheaper than 3-Tier Architecture and Multi-Tier Architecture.
- Easy Deployment: 2-Tier Architecture is easier 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

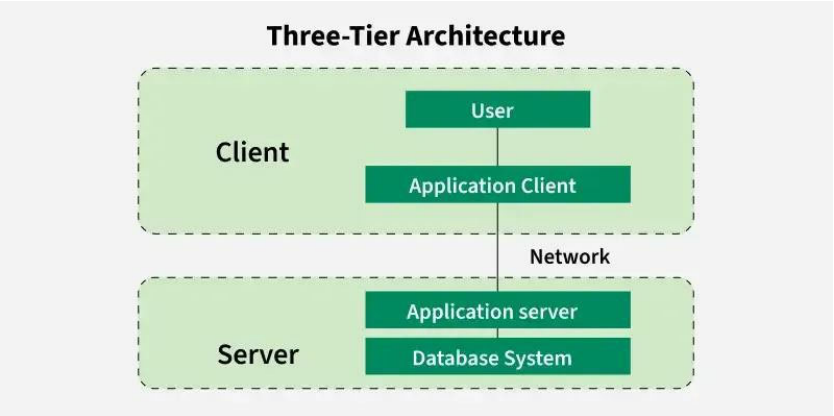
In 3-Tier Architecture, there is another layer between the client and the server. The client does not directly communicate with the server. Instead, it interacts with an application server which further communicates with the database system and then the query processing and transaction management takes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

For Example: E-commerce Store

User: You visit an online store, search for a product and add it to your cart.

Processing: The system checks if the product is in stock, calculates the total price and applies any discounts.

Database: The product details, your cart and order history are stored in the database for future reference.



DBMS 3-Tier Architecture

Advantages of 3-Tier Architecture

- Enhanced scalability: Scalability is enhanced due to the 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.
- 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.