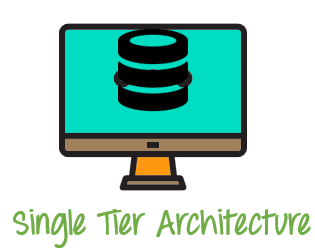
**Application Architecture**

Application architecture helps in design, development, implementation, and maintenance of a database. A database stores critical information for a business. Selecting the correct Database Architecture helps in quick and secure access to this data.

**1-Tier Architecture**

The simplest of Database Architecture are **1 tier**where the Client, Server, and Database all reside on the same machine. Anytime you install a DB in your system and access it to practise SQL queries it is 1 tier architecture. But such architecture is rarely used in production.



This is the 1-Tier Architecture.

**1-tier DBMS** architecture also exist, this is when the database is directly available to the user for using it to store data. Generally such a setup is used for local application development, where programmers communicate directly with the database for quick response.

**Example**- Ms-Office.

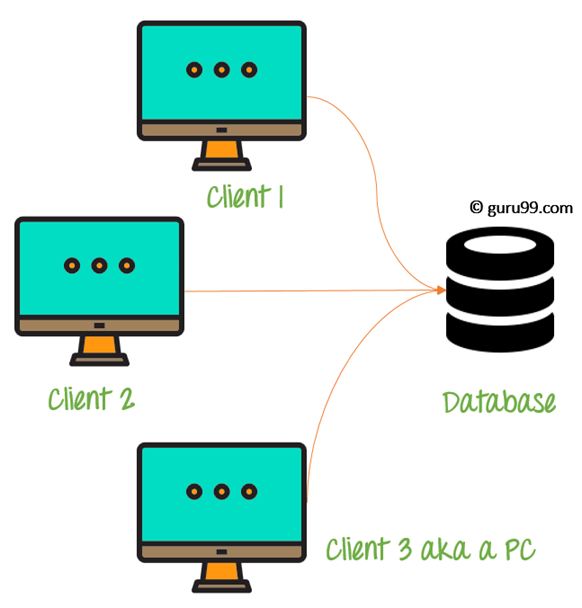
**2-Tier Architecture**

A two-tier architecture is a database architecture where

1. Presentation layer runs on a client (PC, Mobile, Tablet, etc)
2. Data is stored on a Server.

An application interface which is called ODBC (Open Database Connectivity) an API which allows the client-side program to call the DBMS. Today most of the DBMS offers ODBC drivers for their DBMS. 2 tier architecture provides added security to the DBMS as it is not exposed to the end user directly.

**Example** of Two-tier Architecture is a Contact Management System created using MS- Access.



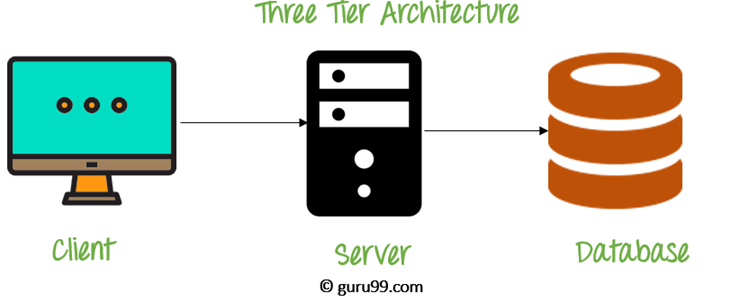
**3-Tier Architecture**

3-tier schema is an extension of the 2-tier architecture. 3-tier architecture has following layers

Presentation layer (your PC, Tablet, Mobile, etc.)

Application layer (server)

Database Server



This DBMS architecture contains an Application layer between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user.

The application layer(business logic layer) also processes functional logic, constraint, and rules before passing data to the user or down to the DBMS.

**The goal of Three-teir architecture is:**

* To separate the user applications and physical database
* Proposed to support DBMS characteristics
* Program-data independence
* Support of multiple views of the data

**n-Tier Architecture**

An **N-Tier Application** program is one that is distributed among three or more separate computers in a distributed network.

The most common form of n-tier is the 3-tier Application, and it is classified into three categories.

* User interface programming in the user's computer
* Business logic in a more centralized computer, and
* Required data in a computer that manages a database.

This architecture model provides Software Developers to create Reusable application/systems with maximum flexibility.

In **N-tier, "N"**refers to a number of tiers or layers are being used like – **2-tier, 3-tier or 4-tier, etc**. It is also called “**Multi-Tier** **Architecture”**.

**The n-tier architecture** is an industry-proven software architecture model. It is suitable to support enterprise level client-server applications by providing solutions to scalability, security, fault tolerance, reusability, and maintainability. It helps developers to create flexible and reusable applications.

A diagrammatic representation of an n-tier system depicts here – presentation, application, and database layers.

