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

A Database store a lot of critical information to access data quickly and securely. Hence it is important to select the correct architecture for efficient data management. DBMS Architecture helps users to get their requests done while connecting to the database. We choose database architecture depending on several factors like the size of the database, number of users, and relationships between the users. There are two types of database models that we generally use, are logical model and physical model. Several types of architecture are there in the database which we will deal with in the next section.

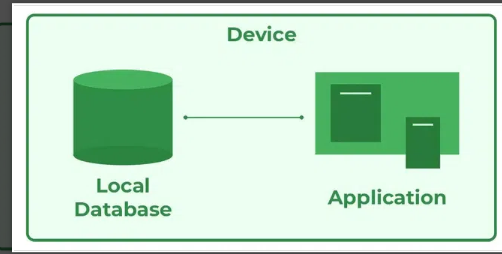
**Types of DBMS Architecture**

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. 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.



*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 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](https://www.geeksforgeeks.org/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 in order to communicate with the DBMS.   
An advantage of this type is that maintenance and understanding are easier, and compatible with existing systems. However, this model gives poor performance when there are a large number of users.



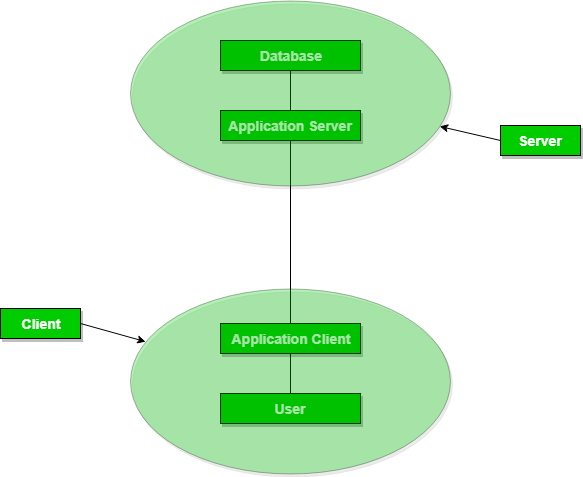
*DBMS 2-Tier Architecture*

**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 by upgrading hardware.
* **Low Cost:** 2-Tier Architecture is cheaper than 3-Tier Architecture and [Multi-Tier Architecture](https://www.geeksforgeeks.org/multi-tier-architecture-of-data-warehouse/).
* **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](https://www.geeksforgeeks.org/introduction-of-3-tier-architecture-in-dbms-set-2/), 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.



*DBMS 3-Tier Architecture*

**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.

## View of data in DBMS:

## View of data three-schema architecture

## **View of data**in DBMS narrate how the data is visualized at each level of data abstraction? **Data abstraction** allow developers to keep complex data structures away from the users. The developers achieve this by hiding the complex data structures through **levels of abstraction**.

Content: View of Data in DBMS

1. [Data Abstraction](https://binaryterms.com/view-of-data.html#DataAbstraction)
2. [Data Independence](https://binaryterms.com/view-of-data.html#DataIndependence)
3. [Instance and Schema](https://binaryterms.com/view-of-data.html#InstanceandSchema)

### Data Abstraction

Data abstraction is **hiding the complex data structure** in order to **simplify the user’s interface** of the system. It is done because many of the users interacting with the database system are not that much computer trained to understand the complex data structures of the database system.

To achieve data abstraction, we will discuss a **Three-Schema architecture** which abstracts the database at three levels discussed below:

#### Three-Schema Architecture:

The main objective of this architecture is to have an effective separation between the **user interface** and the **physical database**. So, the user never has to be concerned regarding the internal storage of the database and it has a simplified interaction with the database system.

**The three-schema architecture defines the view of data at three levels:**

1. Physical level (internal level)
2. Logical level (conceptual level)
3. View level (external level)

**1. Physical Level/ Internal Level**

The physical or the internal level schema describes **how the data is stored in the hardware**. It also describes how the data can be accessed. The physical level shows the data abstraction at the lowest level and it has **complex data structures**. Only the database administrator operates at this level.

**2. Logical Level/ Conceptual Level**

It is a level above the physical level. Here, the data is stored in the form of the **entity set**, **entities**, their **data types**, the **relationship** among the entity sets, **user operations** performed to retrieve or modify the data and certain **constraints on the data**. Well adding constraints to the view of data adds the security. As users are restricted to access some particular parts of the database.

It is the developer and database administrator who operates at the logical or the conceptual level.

**3. View Level/ User level/ External level**

It is the highest level of data abstraction and exhibits only a part of the whole database. It exhibits the data in which the user is interested. The view level can describe many views of the same data. Here, the user retrieves the information using different application from the database.

# Different types of Database Users:

1. **Database Administrator (DBA) :** Database Administrator (DBA) is a person/team who defines the schema and also controls the 3 levels of database. The DBA will then create a new account id and password for the user if he/she need to access the database. DBA is also responsible for providing security to the database and he allows only the authorized users to access/modify the data base. DBA is responsible for the problems such as security breaches and poor system response time.
   * DBA also monitors the recovery and backup and provide technical support.
   * The DBA has a DBA account in the DBMS which called a system or superuser account.
   * DBA repairs damage caused due to hardware and/or software failures.
   * DBA is the one having privileges to perform DCL (Data Control Language) operations such as GRANT and REVOKE, to allow/restrict a particular user from accessing the database.
2. **Naive / Parametric End Users :** Parametric End Users are the unsophisticated who don’t have any DBMS knowledge but they frequently use the database applications in their daily life to get the desired results. For examples, Railway’s ticket booking users are naive users. Clerks in any bank is a naive user because they don’t have any DBMS knowledge but they still use the database and perform their given task.
3. **System Analyst :**  
   System Analyst is a user who analyzes the requirements of parametric end users. They check whether all the requirements of end users are satisfied.
4. **Sophisticated Users :** Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own database applications according to their requirement. They don’t write the program code but they interact the database by writing SQL queries directly through the query processor.
5. **Database Designers :** Data Base Designers are the users who design the structure of database which includes tables, indexes, views, triggers, stored procedures and constraints which are usually enforced before the database is created or populated with data. He/she controls what data must be stored and how the data items to be related. It is responsibility of Database Designers to understand the requirements of different user groups and then create a design which satisfies the need of all the user groups.
6. **Application Programmers :** Application Programmers also referred as System Analysts or simply Software Engineers, are the back-end programmers who writes the code for the application programs. They are the computer professionals. These programs could be written in Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc. Application programmers design, debug, test, and maintain set of programs called “canned transactions” for the Naive (parametric) users in order to interact with database.
7. **Casual Users / Temporary Users :** Casual Users are the users who occasionally use/access the database but each time when they access the database they require the new information, for example, Middle or higher level manager.
8. **Specialized users :**Specialized users are sophisticated users who write   
   specialized database application that does not fit into the traditional data-  
   processing framework. Among these applications are computer aided-design   
   systems, knowledge-base and expert systems etc.

**DATABASE ADMINISTRATOR ROLES AND RESPONSIBILITIES**

**1. Software Installation and Maintenance**

A DBA is frequently involved in the initial installation and configuration of a new Oracle, SQL Server, or other databases. The system administrator configures the database server’s hardware and implements the operating system, after which the DBA installs and configures the database software. The DBA is in charge of ongoing maintenance, such as updates and patches.

**2. Managing Data Integrity**

DBAs primarily handle the overall integrity of a company’s database. They make sure that the Data integrity is carefully managed because it protects data from unauthorized use. DBAs manage data relationships to ensure data consistency.

**3. Takes Care of Data Extraction, Transformation, and Loading**

DBAs are responsible for Data extraction, transformation, and loading, also known as (ETL), which refers to the efficient import of large amount of data extracted from multiple systems into a data warehouse environment. The external data is cleaned and transformed to fit the required format before being imported into [**a central repository**](https://whatagraph.com/data-transfer/google-bigquery/).

**4. Monitoring Performance**

Only implementing a database is not the task of the database administrator. Once the database is implemented, they are required to monitor databases for performance issues. If a system component slows down processing, the DBA may need to change the software configuration or add more hardware capacity. There are numerous monitoring tools available, and understanding what they need to track to improve the system is part of the DBA’s job.

**5. Data Handling**

Each company’s success today revolves around massive databases. Companies nowadays maintain massive databases containing unstructured data types such as images, documents, or sound and video files. Managing an extensive database (VLDB) may necessitate higher-level skills, as well as additional monitoring and tuning, which a DBA possesses.

**6. Create a Database Backup Plan**

DBAs create backup and recovery plans and procedures as per the industry standards. Not only that, but DBAs make certain that all necessary steps are taken. DBAs are responsible for ensuring that everything is completed on time, in addition to taking the required precautions to keep data safe.

**7. Database Recovery**

The DBA’s responsibility in the event of a server failure or other type of data loss is to restore lost data to the system using existing backups.

Different types of failures may necessitate different recovery strategies, and a DBA performs his duties while keeping the necessary requirements in mind. Furthermore, as technology advances, it becomes crucial for a DBA to backup databases to the cloud.

**8. Database Security**

One of the most critical responsibilities of a DBA is identifying and correcting any flaws in the database software. No system is entirely secure; however, DBAs mitigate risks by implementing best practices. A DBA must be able to identify potential flaws in the database software and the overall system of the company and take appropriate steps to mitigate risks.

**9. Database Integrity**

DBAs are primarily responsible for the overall integrity of a company’s database. This includes putting the database in place, keeping it safe from loss and corruption, making it easily accessible, ensuring it works properly, and constantly tweaking it for ease of use and maximum productivity. In addition, the database administrator is also in charge of training eligible employees on how to access and use the database so that they can perform their duties.

**10. Database Accessibility**

Setting up employee access is a critical component of database security. DBAs decide who has access and what kind of access they have. They create a subschema to control database accessibility.