

## **Unit -1**

### **Introduction**

#### **Characteristics of the Database Approach**

##### **1. Manages Information**

A database manages all the information in their required fields (according to column definition and ids).

##### **2. Easy Operation Implementation**

All the operations like insert, delete, update, search, etc. by SQL queries. It is a flexible and easy way. For this user required information of column name and appropriate value. Queries make it more powerful.

##### **3. Multiple Views of Database**

A view is a subset of the database or interface. It is defined according to user requirements. Different users of the system may have different views of the same system. Every view contains only the data of interest to a user or a group of users.

##### **4. Data for Specific Purpose**

A database is designed for a specific purpose according to the group of user and application interests. For example, a database of the student management systems is designed to maintain the record of students' marks, fees, and attendance, etc. This data has the specific purpose of maintaining student records. For example, the library system has three types of users, 1. official administration of the college, 2. librarian, and 3. students.

##### **5. Represent Real World Applications**

A database represents real-world applications. If any change in the real world then it is reflected in the database. Example: - railway reservation system - Maintaining records of passengers, waiting list, train arrival and departure time, certain day, etc. related to each train.

##### **6. Logical Relationship Between Records and Data**

A database maintains a logical relationship between records and data. So a user can access various records according to various logical conditions by a single query from the database.

##### **7. Insulation Between Program and Data**

In the database approach, the data structure is stored in the system catalog, not in the programs. If we want to change the structure of a file then no need to change the program. This feature is called program-data independence. It is not found in the file-based system.

##### **8. Self-describing nature of a database system**

A database system is referred to as self-describing by metadata. Metadata defines and describes the data and relationships between tables in the database. This information is used by the DBMS software or database users if needed. This type of nature is not found in file-based systems.

##### **9. Sharing of data and multiuser system**

Current database systems can access by multiple users at the same time. This feature is called concurrency control strategies. These strategies ensure that the data accessed are always correct and integrated.

#### **10. Control of data redundancy**

In the database approach, each data item is stored in only one place in the database. So single data does not repeat more than one time. This feature improves system performance. Redundancy is controlled by application programming and minimized when designing the database.

#### **11. Enforcement of integrity constraints**

database approach provides the ability to define and enforce certain constraints to ensure that users enter valid information and maintain data integrity. A database constraint is a restriction or rule that dictates what can be entered or edited in a table such as a postal code using a certain format or adding a valid city in the City field.

#### **12. Restriction of unauthorized access**

A database approach should provide security to create and control different types of user accounts and restrict unauthorized access. It provides privileges to access or use data from a database these are read-only access (i.e., the ability to read a file but not make changes), and read and write privileges, (both read and modify a file).

#### **13. Transaction processing**

A database approach supports a concurrency control subsystem that ensures that data remains consistent and valid during transaction processing even if several users update the same information.

#### **14. Backup and recovery facilities**

Backup and recovery are methods that are used to protect your data from loss. Backup means storing database copy to another drive or place. If a hard drive fails or is not accessible then it recovers the database from backup places. If a computer system fails in a complex update process, the recovery method is accessed and restored to its original state.

### **Advantages**

#### **1. Controlling data redundancy.**

Redundancy means storing the same type of data on the different parts DBA avoids unnecessary duplication of data and reduces the storage space. Redundancy leads to several problems:- duplication of efforts, storage space waste, and inconsistent data.

#### **2. Minimise data inconsistency**

Inconsistency means different versions of the same data appear in different places. Data consistency is obtained if a data item appears only once. Any update to its value has to be performed only once.

### **3. Restricting unauthorized access**

Data is stored centrally so enforcing forcibly Security constraints is easier.

The system does not provide to accessing same privileges to all users.

Provides a security subsystem to create and control the user accounts such as user ID and password.

Different checks can be established for each type of access as addition modification deletion To each piece of information in the database.

### **4. Improve data access**

DBMS provides quick answers to queries for data manipulation.

### **5. Support multiple views of data**

A view is a subset of a database that contains virtual data derived from the database file but it does not exist in physical form.

Accessed by any user and have different viewpoints for use of data.

Users may not need to be aware of how and where the data they refer to is stored.

### **6. Enforcing integrity constraint**

Various integrity constraints are identified by database design during database design.

these data integrity constraints can be and forced automatically by the DBMS and others may have to be checked by the application programs.

### **7. Providing backup and recovery**

Recovering from hardware and software failure such as power failure, software error.

saving the data to some other machine periodically is known as backup data.

DBMS recover subsystem either revert back up revert back the database to the state which exists prior to the start of a transaction or resume the transaction from the point it was interesting so that it's completely a fact can be recorded in the database

### **8. Integration of data**

Data in the database is stored in the table.

A single database contains multiple tables and the relationship can be created between the tables. It makes it easy to retrieve and update data.

### **9. Concurrent access and crash recovery**

Data is accessed by only one user at a time.

DBMS protects users from the effects of some system failures

Concurrency control software to ensure that several users are trying to update the same data at the same time.

It should be maintained in a controlled manner

### **10. Reduced application development time**

Develop the application program according to the user's needs.

Other issues like concurrent access security data integrity etc. are handled by the DBMS itself so application development time is reduced.

## **11. Better service to the user**

It improves service data can be shared and the techniques easier to respond to expected information requests.

## **12. Standard can be enforced**

Standards are easier to enforce.

It may be related to the naming of data, the format of data, the structure of data, etc.

It is usually desirable for the purpose of data interchange for migration between systems.

## **13. Data model must be developed**

Building an overall view of an organization's data is usually cost-effective in the long term.

## **14. Data sharing**

Data is shared among multiple users or application programs.

New applications can be developed to use the same store data.

Satisfy the data requirements of a new application without having to create any additional data or with minimal modification.

## **15. Data independence.**

The system data descriptions are separated from the application programs so if any changes in the data structure are not reflected in the application program.

## **16. Transaction processing.**

The DBMS includes a concurrency control subsystem to ensure that several users trying to update the same data do so in a controlled manner so that the result of the update is correct.

## **17. Providing data security.**

Unauthorized persons must not access such confidential data.

The DBA (Database Administrator) is the controller of the database.

He ensures that the proper access procedures are followed.

Performs some checks before permitting access to sensitive data,

Different levels of security according to the significance of various types of data.

For example, a manager in the company can access only the salary structure of employees in his/her department but cannot access the medical history of any employee.

---

## **Disadvantage of DBMS**

### **1. Cost of hardware and software**

A processor with the highest speed of data processing and memory of large size is required to run the DBMS software. DBMS software is also very costly.

### **2. Cost of data conversion**

For this, we have to hire a database and database designer to convert data from one format to another format. A lot of money and developing software's are required

### **3. Cost of staff training**

Complex systems required training for users. Training is required at all levels including programming application development, database administration, and database design. For this lot of amount for the training of staff is required

### **4. Appointing technical staff**

Train technical person such as database administrator, application programmer, database designer, and data entry operators are required to handle the DBMS software of organization and these salaries are very high.

### **5. Database damage**

All data is integrated into a single database. If the database is damaged due to electronic failures or the database is corrupt on the storage media then our valuable data may be lost forever.

### **6. Privacy and security is reduced**

In a database information is stored centrally but it is available to the remote users then it is possible to the destruction of data. So, required technical administrative, and legal measures.

### **Data Model** IMP

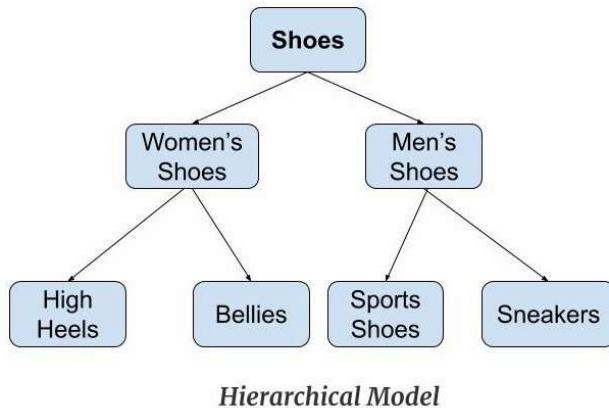
Data Model gives us an idea that how the final system will look like after its complete implementation. It defines the data elements and the relationships between the data elements. Data Models are used to show how data is stored, connected, accessed and updated in the database management system. Here, we use a set of symbols and text to represent the information so that members of the organisation can communicate and understand it. Though there are many data models being used nowadays but the Relational model is the most widely used model. Apart from the Relational model, there are many other types of data models about which we will study in details in this blog. Some of the Data Models in DBMS are:

1. Hierarchical Model
2. Network Model
3. Entity-Relationship Model
4. Relational Model
5. Object-Oriented Data Model
6. Object-Relational Data Model

7. Flat Data Model
8. Semi-Structured Data Model
9. Associative Data Model
10. Context Data Model

### ***Hierarchical Model***

Hierarchical Model was the first DBMS model. This model organises the data in the hierarchical tree structure. The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node. This model easily represents some of the real-world relationships like food recipes, sitemap of a website etc. **Example:** We can represent the relationship between the shoes present on a shopping website in the following way:



### ***Features of a Hierarchical Model***

1. ***One-to-many relationship:*** The data here is organised in a tree-like structure where the one-to-many relationship is between the datatypes. Also, there can be only one path from parent to any node. **Example:** In the above example, if we want to go to the node *sneakers* we only have one path to reach there i.e through men's shoes node.
2. ***Parent-Child Relationship:*** Each child node has a parent node but a parent node can have more than one child node. Multiple parents are not allowed.
3. ***Deletion Problem:*** If a parent node is deleted then the child node is automatically deleted.
4. ***Pointers:*** Pointers are used to link the parent node with the child node and are used to navigate between the stored data. **Example:** In the above example the '*shoes*' node points to the two other nodes '*women shoes*' node and '*men's shoes*' node.

### ***Advantages of Hierarchical Model***

- It is very simple and fast to traverse through a tree-like structure.

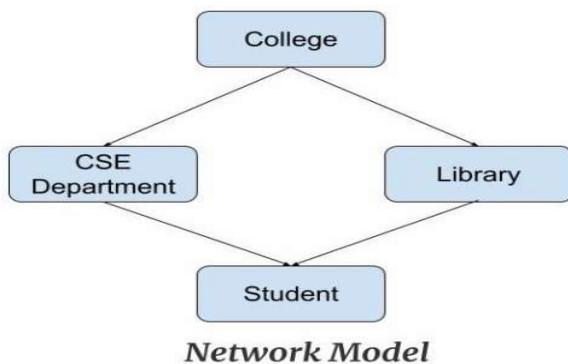
- Any change in the parent node is automatically reflected in the child node so, the integrity of data is maintained.

### *Disadvantages of Hierarchical Model*

- Complex relationships are not supported.
- As it does not support more than one parent of the child node so if we have some complex relationship where a child node needs to have two parent node then that can't be represented using this model.
- If a parent node is deleted then the child node is automatically deleted.

### **Network Model**

This model is an extension of the hierarchical model. It was the most popular model before the relational model. This model is the same as the hierarchical model, the only difference is that a record can have more than one parent. It replaces the hierarchical tree with a graph. **Example:** In the example below we can see that node student has two parents i.e. CSE Department and Library. This was earlier not possible in the hierarchical model.



**Network Model**

### *Features of a Network Model*

1. **Ability to Merge more Relationships:** In this model, as there are more relationships so data is more related. This model has the ability to manage one-to-one relationships as well as many-to-many relationships.
2. **Many paths:** As there are more relationships so there can be more than one path to the same record. This makes data access fast and simple.
3. **Circular Linked List:** The operations on the network model are done with the help of the circular linked list. The current position is maintained with the help of a program and this position navigates through the records according to the relationship.

### *Advantages of Network Model*

- The data can be accessed faster as compared to the hierarchical model. This is because the data is more related in the network model and there can be more than one path to reach a particular node. So the data can be accessed in many ways.
- As there is a parent-child relationship so data integrity is present. Any change in parent record is reflected in the child record.

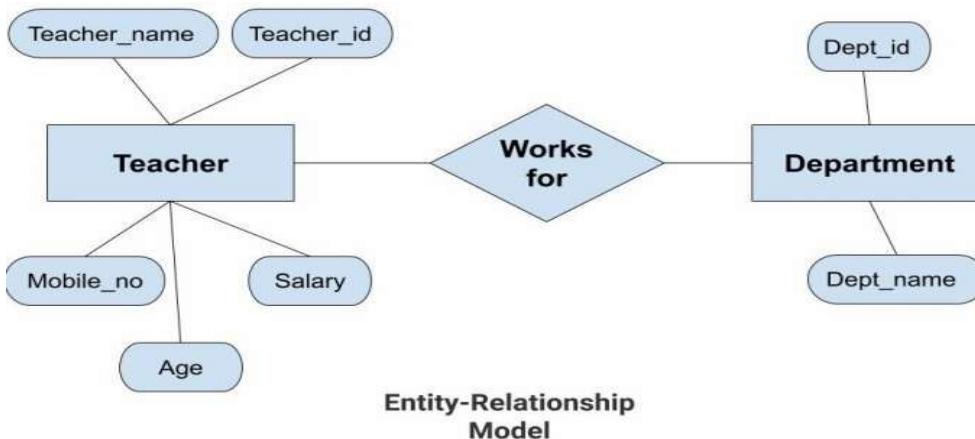
### *Disadvantages of Network Model*

- As more and more relationships need to be handled the system might get complex. So, a user must be having detailed knowledge of the model to work with the model.
- Any change like updation, deletion, insertion is very complex.

### **Entity-Relationship Model**

Entity-Relationship Model or simply ER Model is a high-level data model diagram. In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand. It is also very easy for the developers to understand the system by just looking at the ER diagram. We use the ER diagram as a visual tool to represent an ER Model. ER diagram has the following three components:

- **Entities:** Entity is a real-world thing. It can be a person, place, or even a concept. *Example:* Teachers, Students, Course, Building, Department, etc are some of the entities of a School Management System.
- **Attributes:** An entity contains a real-world property called attribute. This is the characteristics of that attribute. *Example:* The entity teacher has the property like teacher id, salary, age, etc.
- **Relationship:** Relationship tells how two attributes are related. *Example:* Teacher works for a department.



In the above diagram, the entities are Teacher and Department. The attributes of **Teacher** entity are Teacher\_Name, Teacher\_id, Age, Salary, Mobile\_Number. The attributes of

entity **Department** entity are Dept\_id, Dept\_name. The two entities are connected using the relationship. Here, each teacher works for a department.

### *Features of ER Model*

- **Graphical Representation for Better Understanding:** It is very easy and simple to understand so it can be used by the developers to communicate with the stakeholders.
- **ER Diagram:** ER diagram is used as a visual tool for representing the model.
- **Database Design:** This model helps the database designers to build the database and is widely used in database design.

### *Advantages of ER Model*

- **Simple:** Conceptually ER Model is very easy to build. If we know the relationship between the attributes and the entities we can easily build the ER Diagram for the model.
- **Effective Communication Tool:** This model is used widely by the database designers for communicating their ideas.
- **Easy Conversion to any Model:** This model maps well to the relational model and can be easily converted relational model by converting the ER model to the table. This model can also be converted to any other model like network model, hierarchical model etc.

### *Disadvantages of ER Model*

- **No industry standard for notation:** There is no industry standard for developing an ER model. So one developer might use notations which are not understood by other developers.
- **Hidden information:** Some information might be lost or hidden in the ER model. As it is a high-level view so there are chances that some details of information might be hidden.

### *Relational Model*

Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables. So, the tables are also called *relations* in the relational model. **Example:** In this example, we have an Employee table.

Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

**EMPLOYEE TABLE**

### *Features of Relational Model*

- **Tuples:** Each row in the table is called tuple. A row contains all the information about any instance of the object. In the above example, each row has all the information about any specific individual like the first row has information about John.
- **Attribute or field:** Attributes are the property which defines the table or relation. The values of the attribute should be from the same domain. In the above example, we have different attributes of the *employee* like Salary, Mobile\_no, etc.

### *Advantages of Relational Model*

- **Simple:** This model is more simple as compared to the network and hierarchical model.
- **Scalable:** This model can be easily scaled as we can add as many rows and columns we want.
- **Structural Independence:** We can make changes in database structure without changing the way to access the data. When we can make changes to the database structure without affecting the capability to DBMS to access the data we can say that structural independence has been achieved.

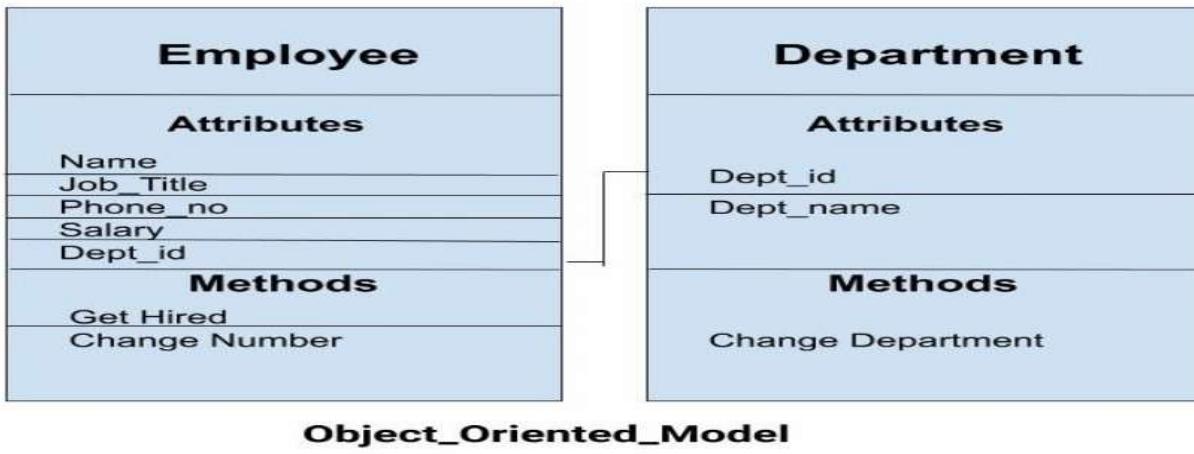
### *Disadvantages of Relational Model*

- **Hardware Overheads:** For hiding the complexities and making things easier for the user this model requires more powerful hardware computers and data storage devices.
- **Bad Design:** As the relational model is very easy to design and use. So the users don't need to know how the data is stored in order to access it. This ease of design can lead to the development of a poor database which would slow down if the database grows.

But all these disadvantages are minor as compared to the advantages of the relational model. These problems can be avoided with the help of proper implementation and organisation.

### *Object-Oriented Data Model*

The real-world problems are more closely represented through the object-oriented data model. In this model, both the data and relationship are present in a single structure known as an object. We can store audio, video, images, etc in the database which was not possible in the relational model (although you can store audio and video in relational database, it is advised not to store in the relational database). In this model, two or more objects are connected through links. We use this link to relate one object to other objects. This can be understood by the example given below.



In the above example, we have two objects Employee and Department. All the data and relationships of each object are contained as a single unit. The attributes like Name, Job\_title of the employee and the methods which will be performed by that object are stored as a single object. The two objects are connected through a common attribute i.e the Department\_id and the communication between these two will be done with the help of this common id.

### ***Object-Relational Model***

As the name suggests it is a combination of both the relational model and the object-oriented model. This model was built to fill the gap between object-oriented model and the relational model. We can have many advanced features like we can make complex data types according to our requirements using the existing data types. The problem with this model is that this can get complex and difficult to handle. So, proper understanding of this model is required.

### ***Flat Data Model***

It is a simple model in which the database is represented as a table consisting of rows and columns. To access any data, the computer has to read the entire table. This makes the model slow and inefficient.

### ***Semi-Structured Model***

Semi-structured model is an evolved form of the relational model. We cannot differentiate between data and schema in this model. **Example:** Web-Based data sources which we can't

differentiate between the schema and data of the website. In this model, some entities may have missing attributes while others may have an extra attribute. This model gives flexibility in storing the data. It also gives flexibility to the attributes. **Example:** If we are storing any value in any attribute then that value can be either atomic value or a collection of values.

### **Associative Data Model**

Associative Data Model is a model in which the data is divided into two parts. Everything which has independent existence is called as an *entity* and the relationship among these entities are called *association*. The data divided into two parts are called items and links.

- **Item:** Items contain the name and the identifier(some numeric value).
- **Links:** Links contain the identifier, source, verb and subject.

**Example:** Let us say we have a statement "The world cup is being hosted by London from 30 May 2020". In this data two links need to be stored:

1. The world cup is being hosted by London. The source here is 'the world cup', the verb 'is being' and the target is 'London'.
2. ...from 30 May 2020. The source here is the previous link, the verb is 'from' and the target is '30 May 2020'.

This is represented using the table as follows:

<b>Items</b>	
<b>Identifiers</b>	<b>Name</b>
89	The world cup
95	Is being hosted
40	By London
44	from
10	30 May 2020

<b>Links</b>			
<b>Identifiers</b>	<b>Source</b>	<b>Verb</b>	<b>Target</b>
70	89	95	40
75	70	44	10

**ASSOCIATIVE MODEL**

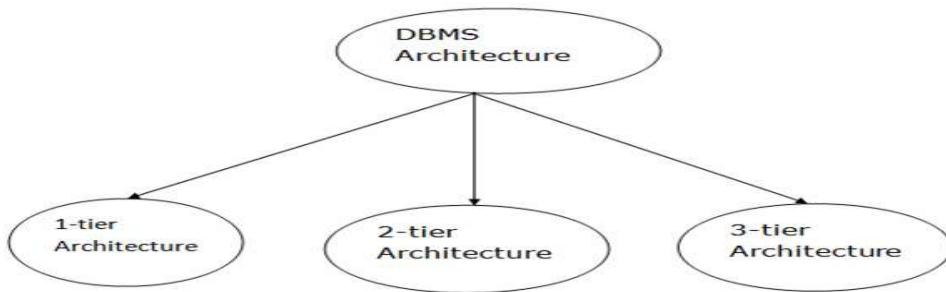
### **Context Data Model**

Context Data Model is a collection of several models. This consists of models like network model, relational models etc. Using this model we can do various types of tasks which are not possible using any model alone.

# DBMS Architecture

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.

## Types of DBMS Architecture



Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: **2-tier architecture** and **3-tier architecture**.

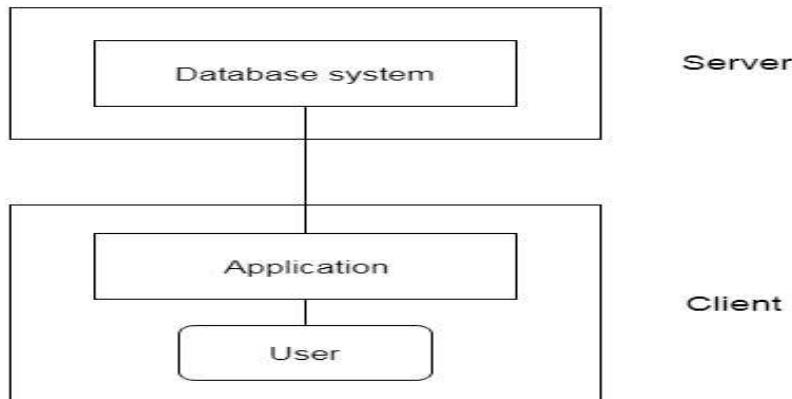
### 1-Tier Architecture

- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

### 2-Tier Architecture

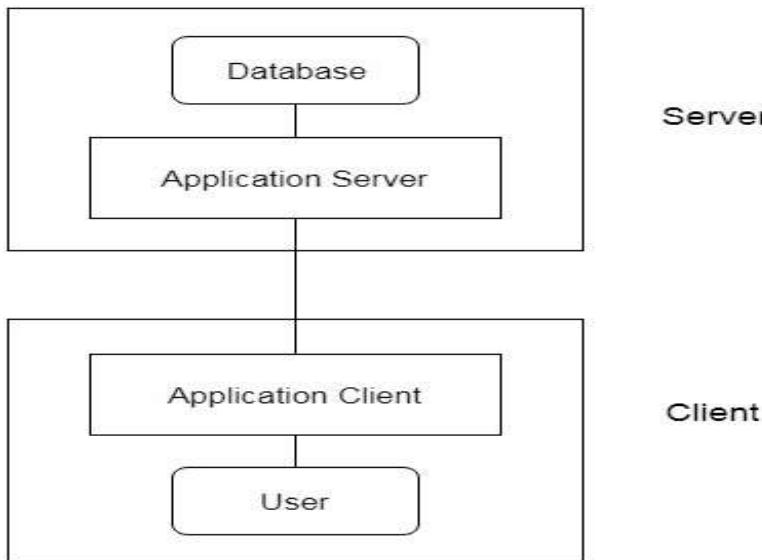
- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: **ODBC**, **JDBC** are used.
- The user interfaces and application programs are run on the client-side.

- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.



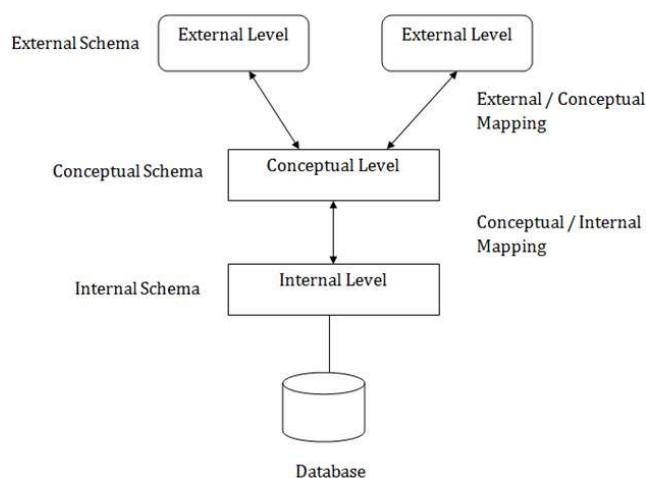
### 3-Tier Architecture

- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- The 3-Tier architecture is used in case of large web application.



## Three schema Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.
- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.



- It shows the DBMS architecture.

- Mapping is used to transform the request and response between various database levels of architecture.
- Mapping is not good for small DBMS because it takes more time.
- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

## Objectives of Three schema Architecture

The main objective of three level architecture is to enable multiple users to access the same data with a personalized view while storing the underlying data only once. Thus it separates the user's view from the physical structure of the database. This separation is desirable for the following reasons:

- Different users need different views of the same data.
- The approach in which a particular user needs to see the data may change over time.
- The users of the database should not worry about the physical implementation and internal workings of the database such as data compression and encryption techniques, hashing, optimization of the internal structures etc.
- All users should be able to access the same data according to their requirements.
- DBA should be able to change the conceptual structure of the database without affecting the user's
- Internal structure of the database should be unaffected by changes to physical aspects of the storage.

### 1. Internal Level

**Internal view**

STORED_EMPLOYEE record length 60	
Empno	: 4 decimal offset 0 unique
Ename	: String length 15 offset 4
Salary	: 8,2 decimal offset 19
Deptno	: 4 decimal offset 27
Post	: string length 15 offset 31

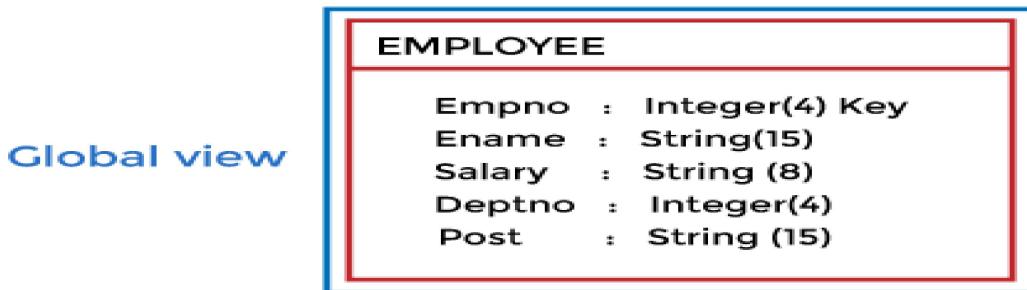
- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.

- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

The internal level is generally concerned with the following activities:

- Storage space allocations.  
**For Example:** B-Trees, Hashing etc.
- Access paths.  
**For Example:** Specification of primary and secondary keys, indexes, pointers and sequencing.
- Data compression and encryption techniques.
- Optimization of internal structures.
- Representation of stored fields.

## 2. Conceptual Level



- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

## 3. External Level



- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

## Mapping between Views

The three levels of DBMS architecture don't exist independently of each other. There must be correspondence between the three levels i.e. how they actually correspond with each other. DBMS is responsible for correspondence between the three types of schema. This correspondence is called Mapping.

**There are basically two types of mapping in the database architecture:**

- Conceptual/ Internal Mapping
- External / Conceptual Mapping

### Conceptual/ Internal Mapping

The Conceptual/ Internal Mapping lies between the conceptual level and the internal level. Its role is to define the correspondence between the records and fields of the conceptual level and files and data structures of the internal level.

### External/ Conceptual Mapping

The external/Conceptual Mapping lies between the external level and the Conceptual level. Its role is to define the correspondence between a particular external and the conceptual view.

## Data Independence

- Data independence can be explained using the three-schema architecture.
- Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

### 1. Logical Data Independence

- Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.

- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

## 2. Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.

there are two types of data independence:

