**QUESTIONS**

1. **Database Schema**A database schema is the logical representation of a database.

It represents the organization of data and provides information about the relationships between the tables in a given database.

A database schema contains schema objects that may include **tables, fields, packages, views, relationships, primary key, foreign key.**

Types of Database Schema

The database schema is divided into three types, which are:

1. **Physical Schema**
2. **Logical Schema**
3. **View Schema**

### **Physical Database Schema**

It describes the database designed at Physical level.

A physical database schema specifies how the data is stored physically on a storage system in the form of Files and Indices.

### **Logical Database Schema**

It describes the database designed at logical level.

The logical schema represents how the data is stored in the form of tables and how the attributes of a table are linked together.

The Logical database schema specifies all the logical constraints that need to be applied to the stored data.

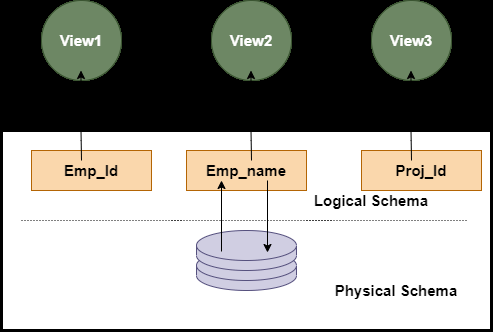
The logical schema represents how the data is stored in the form of tables and how the attributes of a table are linked together.

At this level, programmers and administrators work, and the implementation of the data structure is hidden at this level.

### **View Schema**

It describes the database designed at View level.

The view level design of a database is known as **view schema**. This schema generally describes the end-user interaction with the database systems.



1. Database users and DBA  
     
   1. Native User

2. Application Program User

3. End User

**DBA**

* DBA is a person who works on creating, maintaining, querying, and tuning the database of the organization.
* They are also responsible for maintaining data security and integrity.
* They specialize in database development.

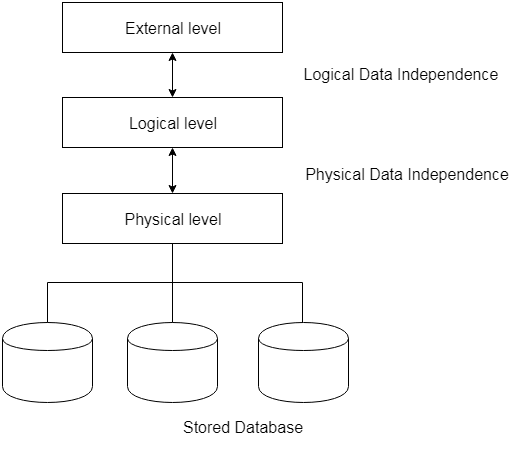
Data Independence

## The ability to modify the schema definition in one level without affecting a schema definition in the next higher level is call data Independence. 1. Logical Data Independence

* The ability 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

* The ability to change the 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.



Explain Mapping

# **Mappings**

Process of transforming request and results between three level it's called mapping.

There are the two types of mappings:

1. Conceptual-Internal Mapping
2. External-Conceptual Mapping

### **1. Conceptual-Internal Mapping:**

* It defines the correspondence between the **conceptual view and the store database.**
* It relates conceptual schema with internal schema.
* It specifies how conceptual record and fields are represented at the internal level.
* There could be one mapping between conceptual and internal levels.

### **2. External/Conceptual Mapping:**

* It defines the correspondence between a particular external view and conceptual view.
* It relates each external schema with conceptual schema.
* Example: fields can have different data types; fields and record name can be changed; several conceptual fields can be combined into a single external field.
* There could be several mapping between external and conceptual levels.

1. Mapping Cardinality Constraints

It tells the number of entities to with another entity can be associated through a 'relationship set' . Mapping cardinality uses binary relationship sets.

Design issues in E-R model

### **1) Use of Entity Set vs Attributes**

### **2) Use of Entity Set vs. Relationship Sets**

### **3) Use of Binary vs n-ary Relationship Sets**

### **4) Placing Relationship Attributes**

DBMS

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently.

It is also used to organize the data in the form of a table, schema, views, and reports, etc

Collection of interrelated data

Set of programs to access the data

An environment that is both convenient and efficient to use

|  |  |  |
| --- | --- | --- |
| **Basis** | **DBMS Approach** | **File System Approach** |
| **Meaning** | DBMS is a collection of data. In DBMS, the user is not required to write the procedures. | The file system is a collection of data. In this system, the user has to write the procedures for managing the database. |
| **Sharing of data** | Due to the centralized approach, data sharing is easy. | Data is distributed in many files, and it may be of different formats, so it isn't easy to share data. |
| **Data Abstraction** | DBMS gives an abstract view of data that hides the details. | The file system provides the detail of the data representation and storage of data. |
| **Security and Protection** | DBMS provides a good protection mechanism. | It isn't easy to protect a file under the file system. |
| **Recovery Mechanism** | DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from system failure. | The file system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will be lost. |
| **Manipulation Techniques** | DBMS contains a wide variety of sophisticated techniques to store and retrieve the data. | The file system can't efficiently store and retrieve the data. |
| **Concurrency Problems** | DBMS takes care of Concurrent access of data using some form of locking. | In the File system, concurrent access has many problems like redirecting the file while deleting some information or updating some information. |
| **Where to use** | Database approach used in large systems which interrelate many files. | File system approach used in large systems which interrelate many files. |
| **Cost** | The database system is expensive to design. | The file system approach is cheaper to design. |
| **Data Redundancy and Inconsistency** | Due to the centralization of the database, the problems of data redundancy and inconsistency are controlled. | In this, the files and application programs are created by different programmers so that there exists a lot of duplication of data which may lead to inconsistency. |
| **Structure** | The database structure is complex to design. | The file system approach has a simple structure. |
| **Data Independence** | In this system, Data Independence exists, and it can be of two types.   * Logical Data Independence * Physical Data Independence | In the File system approach, there exists no Data Independence. |
| **Integrity Constraints** | Integrity Constraints are easy to apply. | Integrity Constraints are difficult to implement in file system. |
| **Data Models** | In the database approach, 3 types of data models exist:   * Hierarchal data models * Network data models * Relational data models | In the file system approach, there is no concept of data models exists. |
| **Flexibility** | Changes are often a necessity to the content of the data stored in any system, and these changes are more easily with a database approach. | The flexibility of the system is less as compared to the DBMS approach. |
| **Examples** | Oracle, SQL Server, Sybase etc. | Cobol, C++ etc. |

Applications of Database management

Banking: transactions

Airlines: reservations, schedules

Universities: registration, grades

Sales: customers, products, purchases

Online retailers: order tracking, customized recommendations

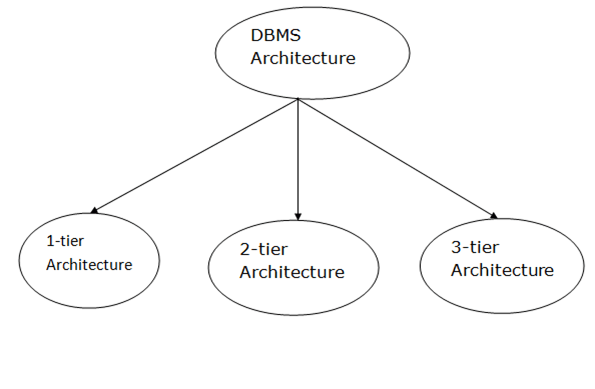
Manufacturing: production, inventory, orders, supply chain

Human resources: employee records, salaries, tax deductions

# **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.

Types of DBMS Architecture

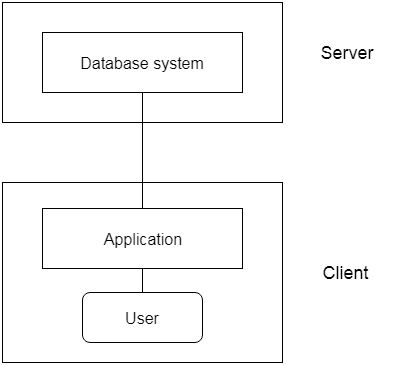


### **1-Tier Architecture**

* 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**

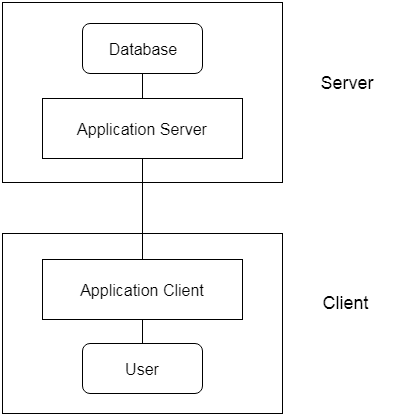
* The 2-Tier architecture is same as basic client-server.
* In the two-tier architecture, applications on the client end can directly communicate with the database at the server side.
* The user interfaces and application programs are run on the client-side.
* The server side is responsible to provide the functionalities like: query processing and transaction management.
* To communicate with the DBMS, client-side application establishes a connection with the server side.



**Fig: 2-tier Architecture**

### **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.
* 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.



**Fig: 3-tier Architecture**

Explain Relational Constraints. Types of it.

Relational Integrity constraints are the set of rules and restrictions which are used to maintain the data consistency and integrity of a Relation (Table).

Key Constraint

**A key constraint is that constraint which uniquely identifies the tuple of a relation.** A Relation should have at least one key constrain

Referential Integrity constraint

**The referential integrity constraint is a type of constraint that exists between two tables for maintaining the data consistency in both the tables.**

This type of integrity constraint depends on the concept of Foreign key.

Entity Integrity constraint

**An entity integrity constraint states that the value of the primary key attribute cannot be NULL.**

Domain constraint

**Domain constraint is that constraint which defines the set of rules for an attribute of 'Relation'**

Domain Constraints are the user-defined data types such as Date, String, Integer, Currency, character, etc.

Tuple Uniqueness Constraint

Tuple Uniqueness Constraint: It specifies that each tuple in a relation must be identified uniquely

What is the need of Key in Database? Differentiate Super Key, Candidate Key, Primary Key and Foreign Key with example.

**KEYS in DBMS** is an attribute or set of attributes which helps you to identify a row(tuple) in a relation(table).

* It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.

1. Primary Key

 A **primary key** is a minimal set of attributes (columns) in a table that uniquely identifies tuples (rows) in that table.

Primary Key rules

* A primary key may have one or more attributes.
* There is only one primary key in the relation (table).
* A primary key attribute value cannot be NULL.

1. Super Key

* A **Super key** is a set of attributes or more in a table that uniquely identifies each records in that table.

1. Candidate Key

* A candidate key is a subset of a super key.
* A candidate key is a single attribute or the least combination of attributes that uniquely identifies each record in the table.
* Every candidate key is a super key but every super key is not a candidate key.

1. Foreign Key

* A Foreign Key is a database key that is used to link two tables together.
* A foreign key is a column or set of columns in one table whose values must have matching values in the primary key of another (or the same) table.

1. Alternate Key

* Alternate keys are those candidate keys which are not the Primary key. There can be only one Primary key for a table.
* Therefore all the remaining Candidate keys are known as Alternate or Secondary keys.

Explain specialization and generalization feature of E-R diagram with example.

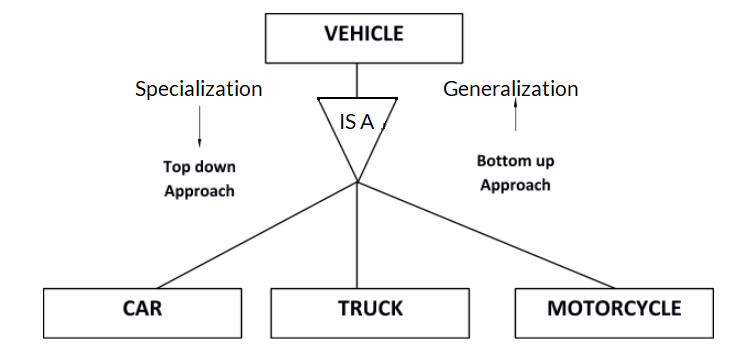
**Specialization** is a top-down approach in which a higher-level entity is divided into multiple specialized lower-level entities.

Specialization is usually used to find subsets of an entity that has a additional attributes.

Generalization is a bottom-up approach in which multiple lower-level entities are combined to form a single higher-level entity.

Generalization is usually used to find common attributes among entities to form a generalized entity.

It can also be thought of as the opposite of specialization.



How GROUP BY clause works?

The GROUP BY Statement in SQL is used to arrange identical data into groups with the help of some functions.

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

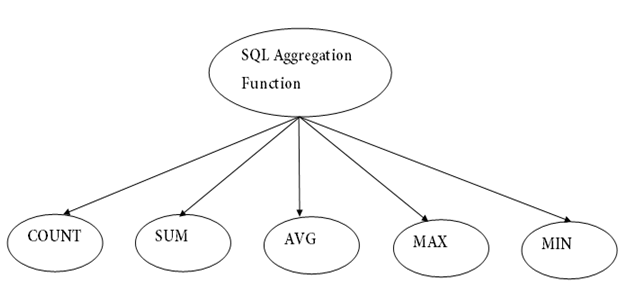
What is the difference between WHERE and HAVING?

The **WHERE clause** is used to fetch the data which specify the given condition.

HAVING clause is used to specify a condition for filtering values from a group.

Aggregate functions

* SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value.
* It is also used to summarize the data.



### **1. COUNT FUNCTION**

* COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.
* COUNT function uses the COUNT(\*) that returns the count of all the rows in a specified table. COUNT(\*) considers duplicate and Null.

**Syntax**

1. COUNT(\*)
2. or
3. COUNT( [ALL|DISTINCT] expression )

### **2. SUM Function**

Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

**Syntax**

1. SUM()
2. or
3. SUM( [ALL|DISTINCT] expression )

### **3. AVG function**

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

**Syntax**

1. AVG()
2. or
3. AVG( [ALL|DISTINCT] expression )

### **4. MAX Function**

MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

**Syntax**

1. MAX()
2. or
3. MAX( [ALL|DISTINCT] expression )

### **5. MIN Function**

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.

**Syntax**

1. MIN()
2. or
3. MIN( [ALL|DISTINCT] expression )