**What is Database:-**

* The database is a collection of inter-related data.
* Organized in such a manner that it facilitate its users to easily access, manage and upload the data.
* It is also used to organize the data in the form of a table, schema, and views etc.
* **For example:** The college Database organizes the data about the admin, staff, students and faculty etc.

**Database Management System:-**

* Database management system is an application which is used to manage the database.
* Main purpose of DBMS is to allow users to store, retrieve and update data in database efficiently.
* For example: [MySQL](https://www.javatpoint.com/mysql-tutorial), [Oracle](https://www.javatpoint.com/oracle-tutorial) are some very popular DBMS.
* DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
* It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

**Advantages of DBMS:-**

* **Controlled Redundancy:** DBMS supports a mechanism to control the redundancy of data inside the database by integrating all the data into a single database and as data is stored at only one place, the duplicity of data does not happen.
* **Data Sharing:** Sharing of data among multiple users simultaneously can also be done in DBMS as the same database will be shared among all the users and by different application programs.
* **Backup and Recovery Facility:** DBMS minimizes the pain of creating the backup of data again and again by providing a feature of ‘backup and recovery’ which automatically creates the data backup and restores the data whenever required.
* **Enforcement of Integrity Constraints:** Integrity Constraints are very important to be enforced on the data so that the refined data after putting some constraints are stored in the database and this is followed by DBMS.
* **Independence of data:** It simply means that you can change the structure of the data without affecting the structure of any of the application programs.

**Disadvantages of DBMS: -**

* **Increased Cost: -**
  + It require sophisticated hardware and software and highly skilled personnel which is required to operate and manage a database system.
  + Cost of using and maintaining all these things can be very high.
* **Require Proper Maintenance: -**
  + As we all know that database contains companies crucial data which is being accessed from multiple sources, so proper maintenance of security issues must be there.
* **Frequent Upgrade: -** 
  + DBMS vendors provide new functionality in the form of upgraded version of software. Some of these versions require hardware upgrades as well as database users and administrators need to be trained to properly use these functionalities. And all these things costs money.
* **Maintaining Concurrency: -**

**Types of Database: -**

* Relational Database
  + (Databases which use rows and columns. Ex-MySQL)
* NoSQL Database
  + (Database which is used to process unstructured data. Ex-MongoDB)
* Cloud Database
  + (Database service which is built and accessed through a cloud platform. Ex-AWS Amazon Web Services)

**What is RDBMS (Relational Database Management System): -**

* It is a database management system in which data is stored in the form of table that contains rows and column.
* Where each row represents an Entity and column represents attributes.
* It uses SQL for storing, manipulating and maintaining data in database.
* Examples of Relational databases are MySQL, Oracle, DB2 etc.

**What are the different types of languages that are available in the DBMS?**

**Answer:** Basically, there are 3 types of languages in the DBMS as mentioned below:

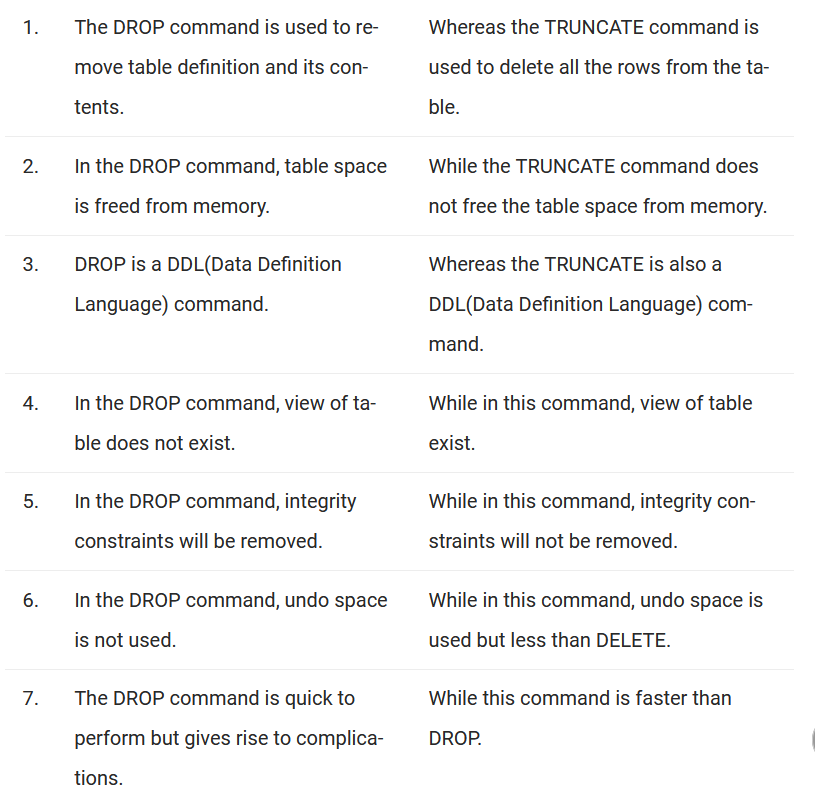
* **DDL:** DDL is **Data Definition Language** which is used to define the database and schema structure by using some set of SQL Queries like **CREATE**, **ALTER**, **TRUNCATE**, **DROP** and **RENAME.**
* **DCL:** DCL is **Data Control Language** which is used to control the access of the users inside the database by using some set of SQL Queries like **GRANT** and **REVOKE.**
* **DML:** DML is **Data Manipulation Language** which is used to do some manipulations in the database like Insertion, Deletion, etc. by using some set of SQL Queries like **SELECT**, **INSERT**, **DELETE** and **UPDATE.**

**Explain the concepts of a Primary key and Foreign Key.**

**Answer: Primary Key** is used to uniquely identify the records in a database table while **Foreign Key** is mainly used to link two or more tables together, as this is a particular field(s) in one of the database tables which are the primary key of some other table.

**Example**: There are 2 tables – Employee and Department. Both have one common field/column as ‘**ID’** where ID is the primary key of the **Employee** table while this is the foreign key for the **Department** table.

**Similarities and Difference between Drop and Truncate Command?**

****

**What is the main difference between UNION and UNION ALL?**

**Answer:** UNION and UNION ALL are used to join the data from 2 or more tables but UNION removes duplicate rows and picks the rows which are distinct after combining the data from the tables whereas UNION ALL does not remove the duplicate rows, it just picks all the data from the tables.

**Explain the concept of ACID properties in DBMS? (Remaining)**

**Answer:** ACID properties is the combination of Atomicity, Consistency, Isolation, and Durability properties. These properties are very helpful in allowing a safe and secure way of sharing the data among multiple users.

* **A for Atomicity: -**
  + This property ensures the data operation will complete either with success or with failure. It follows the 'all or nothing' strategy. For example, a transaction will either be committed or will abort.
* **C for Consistency: -** 
  + This property ensures that the consistency is maintained in the database before or after any transaction that takes place inside the database..
  + For Example:- If we perform any transaction and there are two operation associated with that operation transaction (debit and credit) then inconsistency will occur if debit happened but credit didn’t or vice versa.
* **I means Isolation: -** 
  + This property ensures that multiple transactions can occur concurrently without leading to any data inconsistency in database.
* **D means Durability: -** 
  + This property states that the data should always be in a durable state i.e. any data which is in the committed state should be available in the same state even if any failure or restart occurs in the system.

**Explain Entity, Entity Type, and Entity Set in DBMS?**

* **Entity** is an object, place or thing which has its independent existence in the real world and about which data can be stored in a database. **For Example,**any person, book, etc.
* **Entity Type** is a collection of entities that have the same attributes. **For Example,** the STUDENT table contains rows in which each row is an entity holding the attributes like name, age, and id of the students, hence STUDENT is an Entity Type which holds the entities having the same attributes.
* **Entity Set** is a collection of entities of the same type**. For Example,**A collection of the employees of a firm.

**What are the different levels of abstraction in the DBMS? (Remaining)**

**Answer:** There are 3 levels of data abstraction in the DBMS.

**They include:**

* **Physical Level:** This is the lowest level of the data abstraction which states how the data is stored in the database.
* **Logical Level:** This is the next level of the data abstraction which states the type of the data and the relationship among the data that is stored in the database.
* **View Level:** This is the highest level in the data abstraction which shows/states only a part of the database.

**What integrity rules exist in the DBMS? (Replace)**

**Answer:** There are 2 major integrity rules that exist in the DBMS.

**They are:**

* **Entity Integrity:** This states a very important rule that value of a Primary key can never have a NULL value.
* **Referential Integrity:** This rule is related to the Foreign key which states that either the value of a Foreign key is a NULL value or it should be the primary key of any other relation.

**What is E-R model in the DBMS?**

**Answer:** E-R model is known as an **Entity-Relationship model** in the DBMS which is based on the concept of the Entities and the relationship that exists among these entities.

**What is a functional dependency in the DBMS?**

**Answer:** This is basically a constraint which is useful in describing the relationship among the different attributes in a relation.

**Example:** If there is some relation ‘R1’ which has 2 attributes as Y and Z then the functional dependency among these 2 attributes can be shown as **Y->Z** which states that Z is functionally dependent on Y.

**What is 1NF in the DBMS?**

**Answer:** 1NF is known as the **First Normal Form**.

This is the easiest form of the normalization process which states that the **d**omain of an attribute should have only atomic values. The objective of this is to remove the duplicate columns that are present in the table.

**What is 2NF in the DBMS?**

**Answer:** 2NF is the **Second Normal Form**.

Any table is said to have in the 2NF if it satisfies the following 2 conditions:

* A table is in the 1NF.
* Each non-prime attribute of a table is said to be functionally dependent in totality on the primary key.

**What is 3NF in the DBMS?**

**Answer:** 3NF is the **Third Normal Form.**

Any table is said to have in the 3NF if it satisfies the following 2 conditions:

* A table is in the 2NF.
* Each non-prime attribute of a table is said to be non-transitively dependent on every key of the table.

**What is BCNF in the DBMS?**

**Answer:** BCNF is the **Boyce Codd Normal Form** which is stricter than the 3NF**.**

Any table is said to have in the BCNF if it satisfies the following 2 conditions:

* A table is in the 3NF.
* For each of the functional dependency X->Y that exists, X is the super key of a table.

**Q #23) What is a CLAUSE in terms of SQL?**

**Answer:** This is used with the SQL queries to fetch specific data as per the requirements on the basis of the conditions that are put in the SQL. This is very helpful in picking the selective records from the complete set of the records.

**For Example,** There is a query which has WHERE condition or the query with the HAVING clause.

**Q #24) How can you get the alternate records from the table in the SQL?**

**Answer:** **If you want to fetch the odd numbers then the following query can be used:**

|  |
| --- |
| SELECT EmpId from (SELECT rowno, EmpId from Emp) WHERE mod(rowno,2)=1; |

**If you want to fetch the even numbers, then the following query can be used:**

|  |
| --- |
| SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=0; |

**Q #25) How is the pattern matching done in the SQL?**

**Answer:** With the help of the LIKE operator, pattern matching is possible in the SQL.’**%**’ is used with the LIKE operator when it matches with the 0 or more characters and **‘\_**’ is used to match the one particular character.

**Example:**

|  |
| --- |
| SELECT \* from Emp WHERE name like ‘b%’; |
| SELECT \* from Emp WHERE name like ‘hans\_’; |

**Q #26) What is a join in the SQL?**

**Answer:** A Join is one of the SQL statements which is used to join the data or the rows from 2 or more tables on the basis of a common field/column among them.

**Q #27) What are different types of joins in SQL? (practice)**

**Answer: There are 4 types of SQL Joins:**

* **Inner Join:** This type of join is used to fetch the data among the tables which are common in both the tables.
* **Left Join:** This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.
* **Right Join:** This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
* **Full Join:** This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values.

**Q #28) What is meant by trigger?**

**Answer:** Trigger is one of the very important codes or programs which get executed **automatically in response to the events** that occur in a table or a view. **For Example,** If a new record is inserted in an employee database then the data gets created automatically in the related tables like salary, department and roles tables.

**Q #29) Explain the Stored Procedure.**

**Answer:** Stored Procedure is a group of SQL statements in the form of a function that has some unique name and is stored in relational database management systems (RDBMS) and can be accessed whenever required.

**Q #30) What is RDBMS?**

**Answer:** RDBMS is the Relational Database Management System which contains data in the form of the tables and data is accessed on the basis of the common fields among the tables.

**Q #31) What are the different type of relationships in the DBMS?**

**Answer: Relationships in DBMS depicts an association between the tables.**

**Different types of relationships are:**

* **One-to-One:** This basically states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables. **Example:** Among a married couple, both wife and husband can have only one spouse.
* **One-to-Many:** This states that there can be many relationships for one i.e. a primary key table hold only one record which can have many, one or none records in the related table. **Example:** A Mother can have many children.
* **Many-to-Many:** This states that both the tables can be related to many other tables. **Example:** One can have many siblings and so do they have.

**What is NULL Values: -**

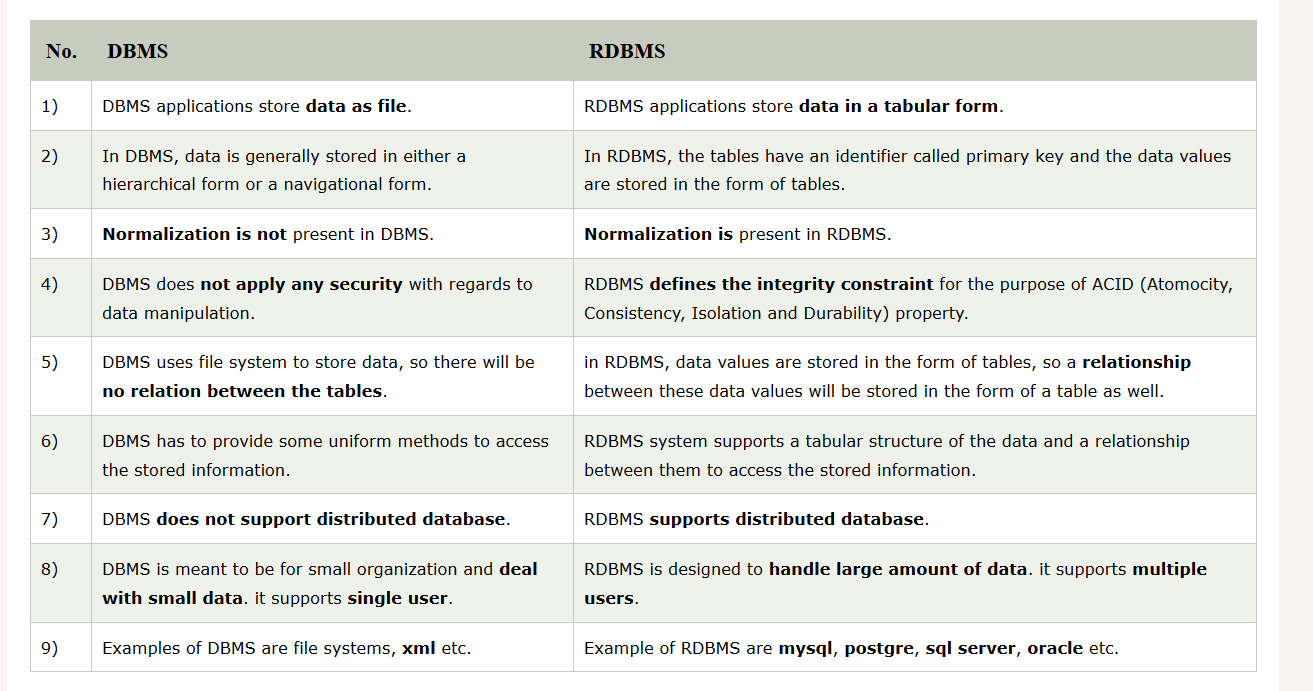
* Null value in a table specifies that the field has been left blank during creation of record.

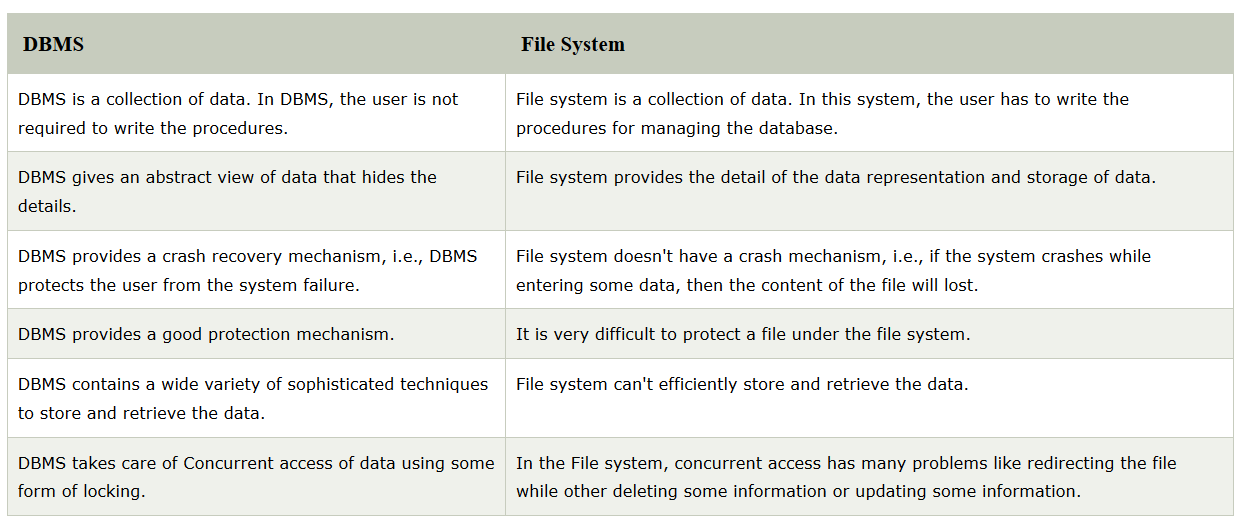
**Data Integrity: -**

There are four categories of data integrity in RDBMS:-

* **Entity Integrity: -** It specifies that there should be no duplicate rows in a table.
* **Domain Integrity:** - It enforces user to enter valid data as per the required format or type of data or range of values.
* **Referential Integrity:** -rows which are being used by other records cannot be deleted.
* **User-defined Integrity: -** It allows you to define specific rules which don’t fall under above categories. (**REMAINING)**

**Difference between DBMS and RDBMS**



**Difference between DBMS and File System: -**

**DBMS Architecture: -**

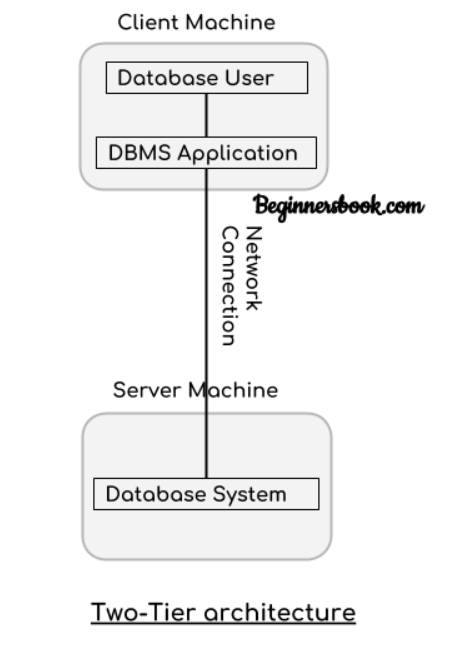
* DBMS Architecture help us to understand the components of database system and relation among them.

**Types of DBMS Architecture:-**

* Single Tier Architecture
* Two Tier Architecture
* Three Tier Architecture

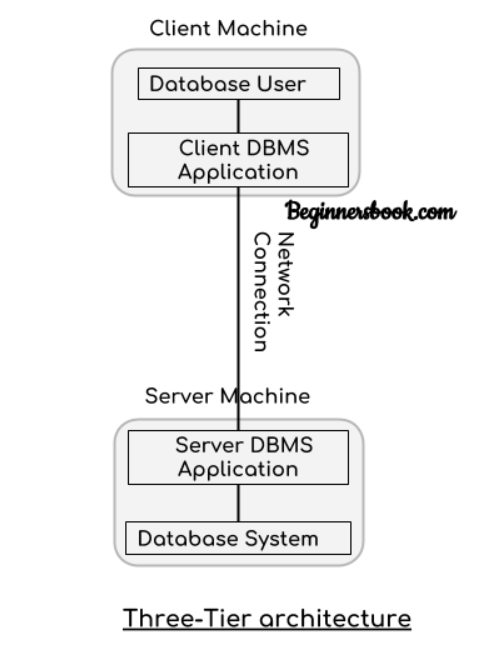
**Single Tier Architecture:-**

* In Single Tier Architecture, the database is directly available on the client machine, any request made by client to perform action on database don’t need any network connection.
* For Example, you want to fetch record from database and database is directly available on your system, so the request to fetch record will be done by your system itself. This type of system is generally referred as local database system.

**Two Tier Architecture: -**

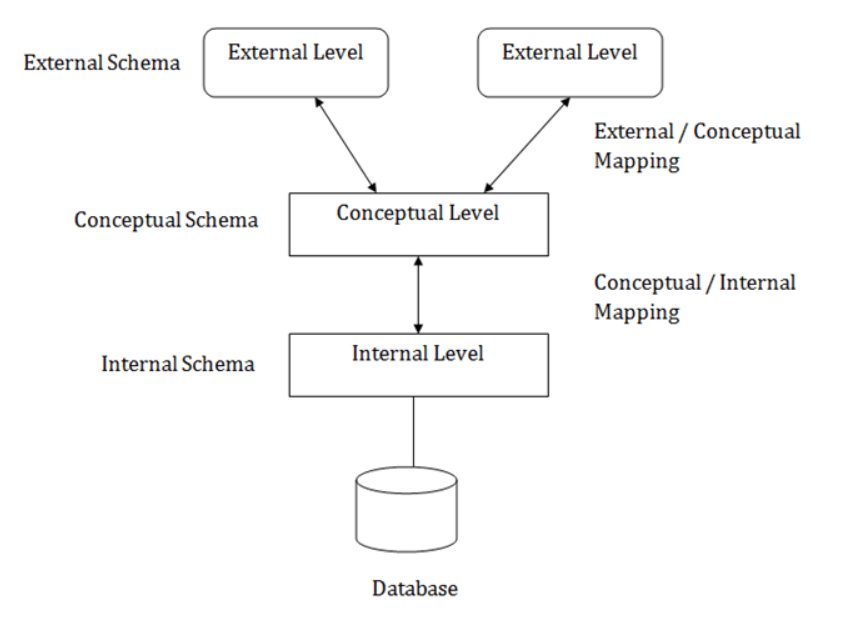
* In Two-tier architecture, there are two components these are: server and a client.
* Database is present at server and on client side we have database user and DBMS software. These two are connected with each other using a reliable network connection.
* Whenever user want to access data available in database, it sends request to server using DBMS software by writing a query in SQL language and in turn server perform action and respond with the appropriate response.

**Three Tier Architecture: -**

* In Three-Tier Architecture, Again there are two components: Server and client, on client we have Database user and client DBMS application, on server side we have Database system and server DBMS application. They are connected with each other using a reliable network connection.
* Whenever user want to perform any action on database DBMS application on client side is used to send request to DBMS application on server side. Then it will perform action on database internally.

**Three Schema (structure) Architecture: -**

* This Architecture is used to describe the structure of a specific database system.
* Main motive of using this architecture is to separate the user application and physical database.
* For Example: - We all use Gmail and store important stuff in it like pictures, videos, mails etc. But we don’t know where exactly this data is stored physically. That’s how it make create abstraction or independence between user and data.
* It basically divide database into three different categories.
  + External level
  + Conceptual level
  + Internal level



**Internal Schema: -**

* Internal level has a schema which describe the physical storage structure of the database.
* How the data is being stored in database, how encryption of data will happen, how much space to be allocated all these things are decided at internal level.

**External Schema: -**

* External Schema includes multiple external view.
* Each external view is separate from each other and fulfil the needs of particular category of users.
* External view describes the database part that a particular user group is interested and hides the remaining database from that user group.
* For Example: - we have a university management system in which student and faculty both can login, but both of them will have a different view of database. A student will be able to check his attendance, marks and fee related details and so on while a faculty will be able to upload student marks, check his salary and so on. That’s how both of them are getting different views and we only showing information to a particular as per need.

**Conceptual Schema: -**

* It describes the design of database at the conceptual level.
* It describes the structure of whole database as well as what data is to be stored in it and what relationship exists among the data.
* It basically describes the data types, relationships and various constraints applicable on the data.
* Programmer and database administrators work at this level.

**Data Independence (abstraction): -**

* Data Independence means a change of data at one level should not affect another level. Two types of data independence are:-
  + Physical data independence
  + Conceptual data independence

**Physical Data Independence: -**

* Any change in the physical location of tables and indexes should not affect the conceptual level or external view of data.
* It is used to separate conceptual levels from internal levels.
* This data independence is easy to achieve and implemented by most of the DBMS.

**Conceptual (logical) Data Independence: -**

* The data at conceptual level schema and external level schema must be independent.
* This means any change in conceptual schema should not affect external schema. It is used to separate external level from conceptual view.
* For example: - Adding or deleting attributes of a table should not affect the user’s view of the table.
* But this type of independence is difficult to achieve as compared to physical data independence because changes in conceptual schema are reflected in the user’s view.