**What is Database:-**

* The database is a collection of inter-related data. It 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, 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 a software which is used to manage the database.
* For example: [MySQL](https://www.javatpoint.com/mysql-tutorial), [Oracle](https://www.javatpoint.com/oracle-tutorial) are very popular commercial database.
* 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:-**

* **Minimized Data Inconsistency: -**
  + Data inconsistency occurs between files when different versions of the same data appear in different places.
  + For Example, data inconsistency occurs when a student name is saved as “Shubham Kumar” in some place and “Shubham Singh” on other places.
  + If database is properly designed then data inconsistency can be greatly reduced which result in minimized data inconsistency.
* **Improve Data Sharing: -** 
  + DBMS helps to create an environment in which end users have better access to well-organized data. Such access also allow authorized user to share data within organization.
* **Better Data Security: -**
  + As number of users increases day by day data transferring rate also increases which in turn increase the risks related to data security. A Database Management System (DBMS) provide a better platform to organizations so that they can improve Data Security.
* **Better data integration: -**
  + Due to Database Management System we have an access to well managed and synchronized form of data thus it makes data handling very easy and gives integrated view of how a particular organization is working
* **Faster Data Access: -** 
  + DBMS helps to produce quick answers to database queries which makes accessing data faster and more accurate.
* It also provide other facilities like control data redundancy, Reduce development time, Provide Backup and so on.

**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
* NoSQL Database
* Centralized Database
* Cloud Database

**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.
* It uses SQL for storing, manipulating and maintaining data in database.
* Examples of Relational databases are MySQL, Oracle, DB2 etc.

**Properties of Relational Database (ACID): -**

* **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 if we perform any operation over the data, its value before and after operation must be consistent.
  + 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 ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs.
* If we conclude try to conclude it, then come to point that **ACID** properties provide mechanism to ensure correctness and consistency of data such that each transaction produces consistent results.

**What is NULL Values: -**

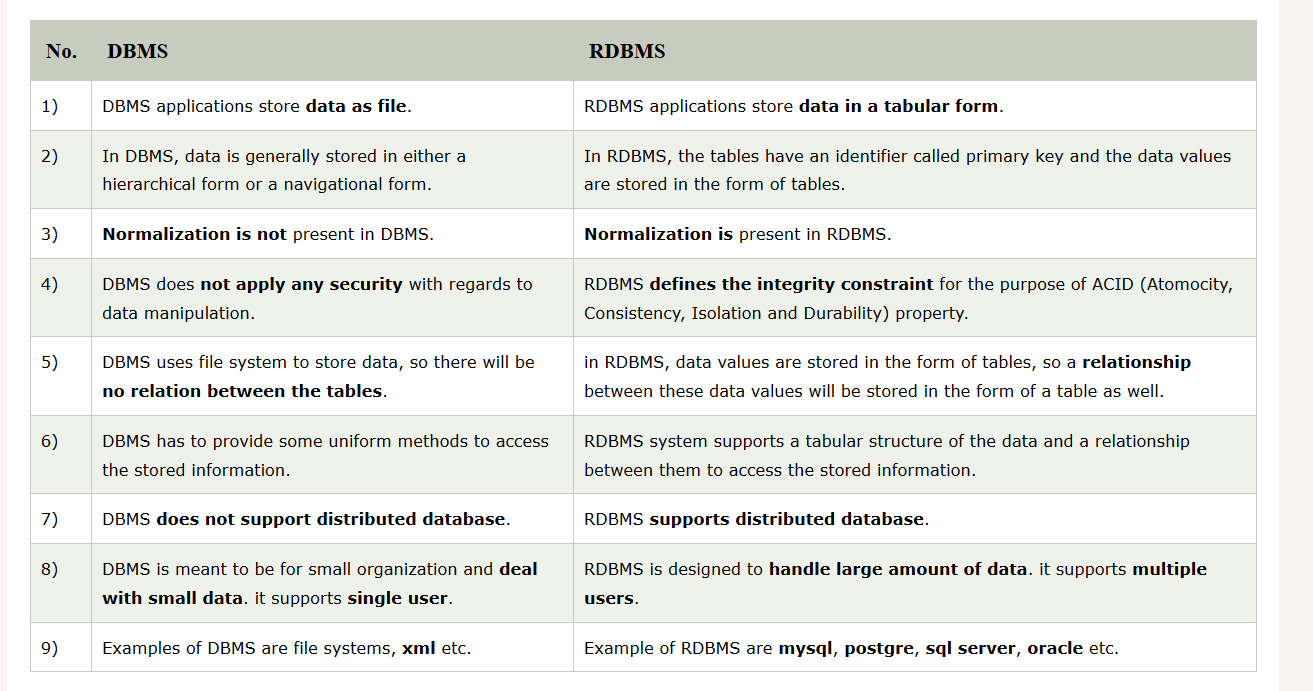
* Null value in a table specifies that the field has been left blank during creation of record.

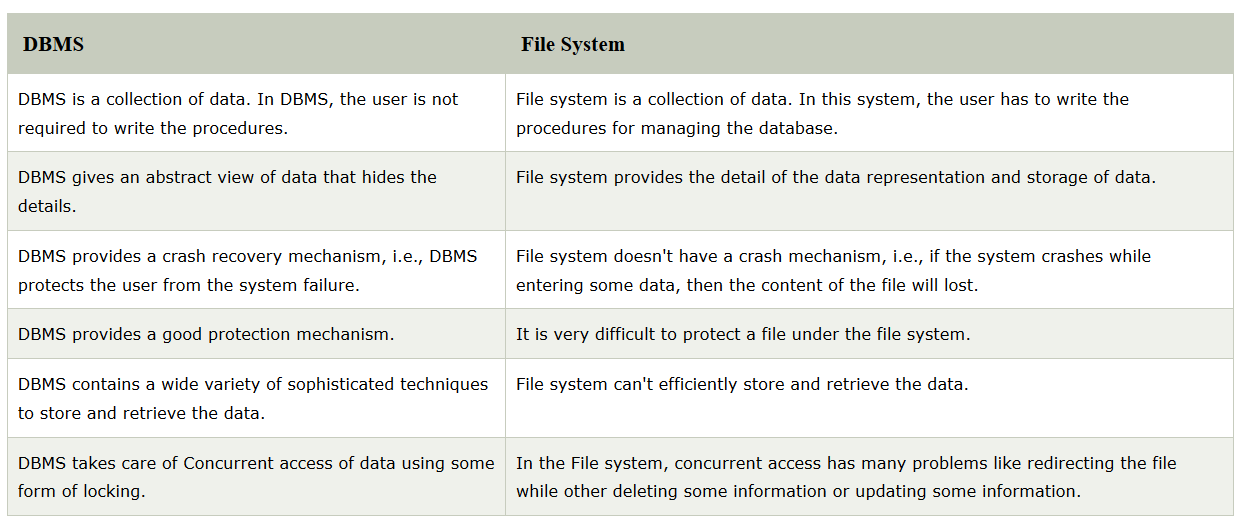
**Data Integrity: -**

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: -**

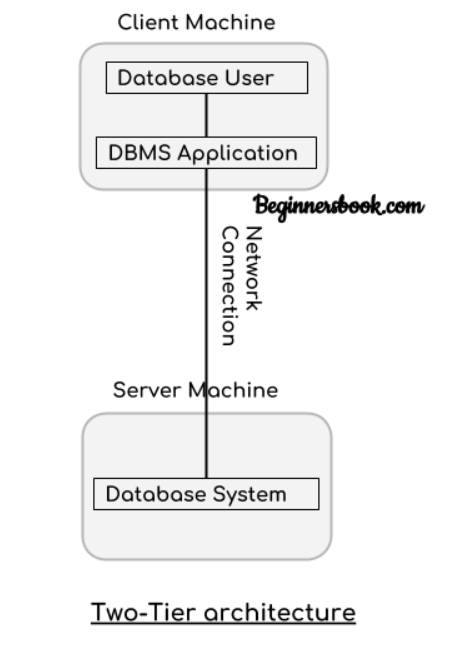
* DBMS Architecture help us to understand the components of database system and relation among them.

**Types of DBMS Architecture:-**

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

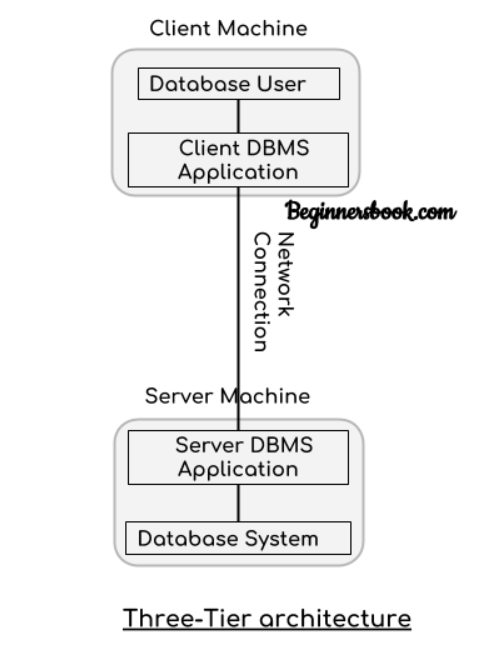
**Single Tier Architecture:-**

* 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: -**

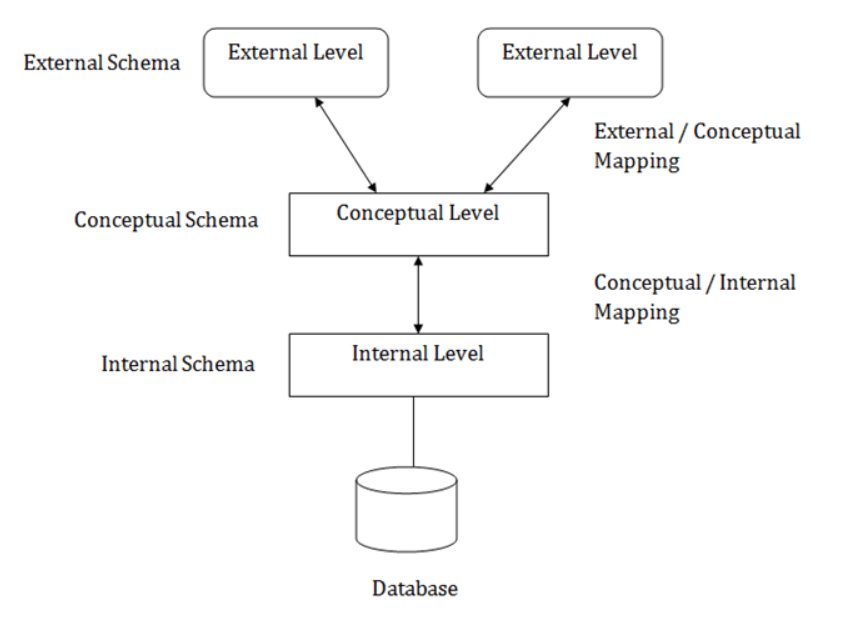
* In Two-tier architecture, there are two components these are: server and a client.
* Database is present at server and on client side we have database user and DBMS software. These two are connected with each other using a reliable network connection.
* Whenever user want to access data available in 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.