DBMS NOTES

UNIT 1 - Introduction to Databases

Data -

- Raw form of information. The facts that can be recorded and have implicit meaning known as 'data'.
- Example - 10, UAT, etc.

Information -

- Processed data i.e in structured format.
- Example: Customer ----- name, city, phone number, etc.

Metadata -

- Information about data.
- Example - Size, date, time of an image.

Database -

- It is a logical collection of interrelated data.
- These can be stored in the form of tables.
- A database can be of any size and varying complexity.
- A database may be generated and manipulated manually or it may be computerised.

Traditional & Modern Database -

- Traditional - Contain only numeric and text values (structured data).
- Modern - That can also store data in unstructured form like images, videos, tweets, etc. New DB also called Big data or NOSQL.

Database System -

- It is a computerised system, whose overall purpose is to maintain the information and to make that information available on demand.
- Collection of related data, organised in a structured and meaningful way. Aim is to store data in such a way which provides faster - access, update and manipulation.

Database Management System (DBMS) -

- It is a collection of programs that enables users to create and maintain a database.
- In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.
- Example of DBMS - SQL, MySQL, NOSQL, Oracle
- It manages - data definition, updation, retrieval, administration and security.
- Advantages of DBMS -
  - Data Independence.
  - Efficient Data Access.
  - Data Integrity and security.
  - Data administration.
  - Concurrent access and Crash recovery.
  - Reduced Application Development Time.

- **Data structure used in DBMS -** B or B+ tree
- **Tuple = Record = Row**
- **Attributes = Column**

People who deal with databases -

- Database Administrators (DBA) - The DBA is responsible for authorising access to the database, for Coordinating and monitoring its use and for acquiring software and hardware resources as needed. These are the people who maintain and design the database daily.
- Database Designers - Database designers are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data.
- End Users - People who wish to store and use data in a database. End users are the people whose jobs require access to the database for querying, updating and generating reports, listed as below.
    - Casual End users - These people occasionally access the database, but they may need different information each time.
    - Naive or Parametric End Users - Their job function revolves around constantly querying and updating the database using standard types of queries and updates.
    - Sophisticated End Users - These include Engineers, Scientists, Business analysts and others familiarise themselves to implement their applications to meet their complex requirements.
    - Stand alone End users - These people maintain personal databases by using ready-made program packages that provide easy to use menu based interfaces.
- System Analyst - These people determine the requirements of end users and develop specifications for transactions.
- Application Programmers (Software Engineers) - These people can test, debug, document and maintain the specified transactions.

File System VS DBMS -

- Data Concurrency - If we do manipulation using multiple instances at the sametime we need to take track of all these changes so that manipulation done on data , results in correct value. So this thing is easily handled by DBMS using a locking system.
- Data Integrity - We can put constraints on data.
- Data sharing & security - We can give different access rights.
- Data searching - Easy to search in DBMS.
- Data Inconsistency - Consistence means if some value is used at different places and at one place we changed value then at all places it should change. DBMS maintain this feature.
- Data Redundancy - DBMS check for duplicates.

Data Abstraction -

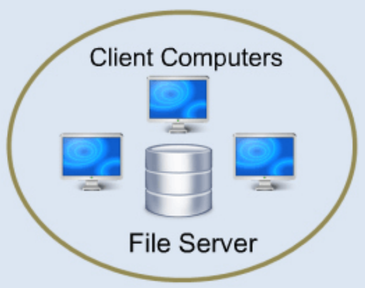
- Refer to hiding details from users at certain levels for authorization and security purposes.
- It consist of 3 levels -
    - Physical (Lowest / Internal) Level - This is a lowest level, which describes how the data is actually stored.
    - Conceptual (Logical / Middle view) Level - It describes the logical structure of a database. Specify what type of data can be stored in DB and relation between them.
    - External (View / Highest) Level - Describes users view of the database. It provides security management.

Data Independencies -

- It says that all the transactions or changes made at obe level are unaffected to all other levels.
- Types -
    - Logical data Independence - It states that the external level is completely unaffected and free from any changes that are made at the conceptual level and vice - versa.
    - Physical Data Independence - It states that the conceptual level is completely unaffected and free from any changes that are made at the internal level and vice - versa.

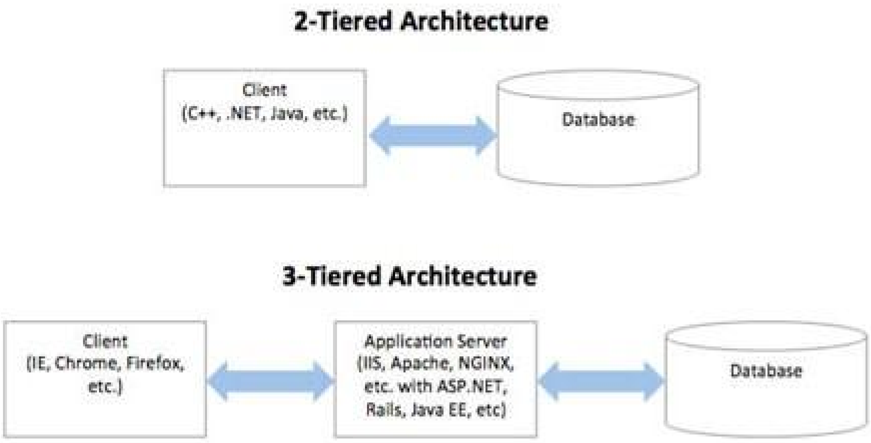
Types of Database Architecture -

- 1 Tier Architecture -
    - In such systems all the required components are all on one server or platform.
    - Advantage - Easy & Simple
    - Disadvantage - Doesn't provide many tools to end users.



- 2 Tier Architecture -
    - In this we have divided the whole architecture into client and database tier. It is a client-server model.
    - The client where data needs to be delivered to somewhere.
    - The server is where data is stored, updated and processed and communicated between these 2 happens.
    - ODBC, JDBC used for the interaction.
    - Example of 2-Tier architecture - Excel, MS Office, etc.
    - Advantage - Direct & faster communication
    - Disadvantage - Can't be used with dynamic web applications.

- 3 Tier Architecture -
    - It consists of 3 tiers - Database (client) tier, Application tier, User (server) tier.
    - Database Tier - Contain raw data along with DBMS system language and its queries that allow different operations on the DB.
    - Application Tier - Act as an intermediate layer between the database and users.
    - User Tier - Here we view all DB results. HTML/ CSS also used to modify how this data looks to end user.
    - Advantage - Security,To provide separation, Data independency, Integrity
    - Disadvantage - Complexity in implementation, Delay in response.



Data Models in DBMS -

- Hierarchical Database Model -
  - Data arranged in tree based data format.
  - It only follows one to many relationships.
- Network Model -
  - Data arranged in graph-like format.
  - Here we can have many to many relationships.
- Entity - relationship Model -
  - In this we break data into different components based on characteristics in terms of entity and attributes.
  - It consists of - entity, attributes and relationship.
- Relational Model -
  - Data stored in 2D tabular form.
  - Relations establish using at least one common field.
  - It may sometimes lead to a problem called ‘Island of Info’. When you forget to create relations between 2 tables.
- Object Oriented Model -
  - Here we treat everything asobject.
  - This is actually a collection of related data.
  - Here data and relationships are kept in a single structure.

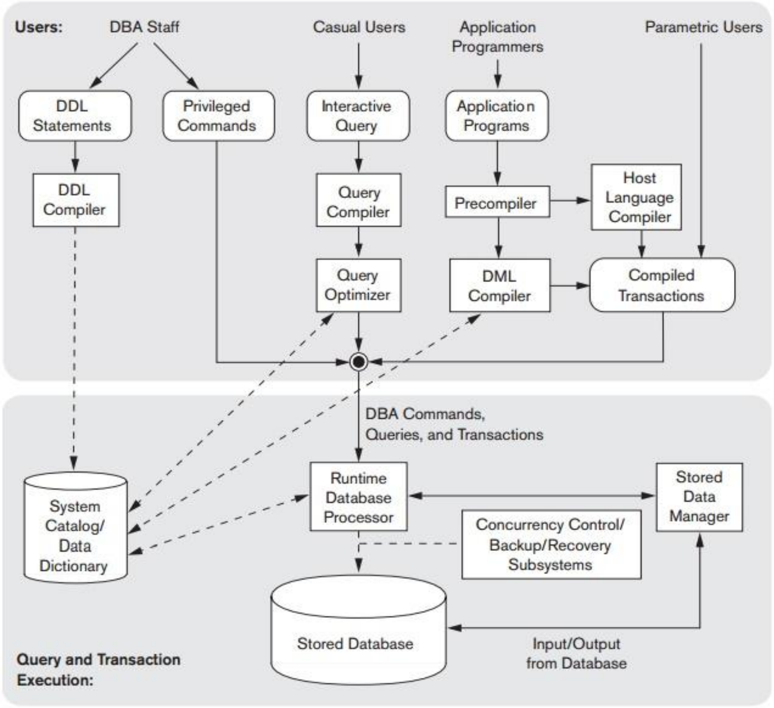
- **Database Schema -** It is the description of a database which is specified during database design. If changed, it is called **schema evolution.**

- **Schema Diagram -** A displayed schema.

- **Database Snapshot** - The data in the database at a particular moment in time.

- **DBMS Languages -**
  - Data Definition Language - Used to classify the database structure or schema. It is a type of language that allows the DBA or user to depict and name those entities, attributes, and relationships that are required for the application along with any associated integrity and security constraints. Example - CREATE, ALTER.
  - Data Manipulation Language - The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. Example - INSERT, UPDATE.
  - Data Control Language - The Data Control Language (DCL) is used to control privilege in Databases. To perform any operation in the database, such as for creating tables, sequences, or views, we need privileges. Example - Grant, Revoke
  - Transaction Control Language - Used to run the changes made by DML statements. It allows statements to be grouped into logical transactions. Example - COMMIT, ROLLBACK.
  - Data Query Language **-** Used to extract data. Example - SELECT.

- **DBMS Component Modules -**



- The figure is divided into two halves. The top half of the figure refers to the various users of the database environment and their interfaces. The lower half shows the internals of the DBMS responsible for storage of data and processing of transactions.
- The database and the DBMS catalogue are usually stored on disk. Access to the disk is primarily controlled by the operating system (OS) which includes disk input/Output. A higher level stored data manager module of DBMS controls access to DBMS information that is stored on the disk.
- If we consider the top half of the figure, It shows interfaces to DBA staff, casual users, application programmers and parametric users
- The DDL compiler processes schema definitions, specified in the DDL, and stores the description of the schema in the DBMS Catalogue..The catalogue includes information such as names and sizes of the files, data types of data of data items. Storage details of each file, mapping information among schemas and constraints.
- Casual users and persons with occasional need of information from the database interact using some interface which is interactive query interface. The queries are parsed, analysed for correctness of the operations for the model, the names of the data elements and so on by a query compiler that compiles them into internal form.
- The internal query is subjected to query optimization. The query optimizer is concerned with rearrangement and possible recording of operations, eliminations of redundancies.
- Application programmers write programs in host languages. The pre-compiler extracts DML commands from an application program.