Chapter-1

Introduction to DBMS

CRUD:

Create, Read, Update and Delete.

- **Data:** Information stored in memory in a particular manner
- **Database:** Collection of structured information or data.
- **DBMS:** Collection of set of application programs used to access, update and manage data.
- **Advantages:**
  - Data Sharing
  - Data Security
  - Data Searching
  - Backup
  - Maintenance
- **Disadvantages:**
  - Complex Structure of DBMS
  - High Maintenance
  - Compatibility
  - Cost of hardware & software

- **Difference between file system and DBMS:**

Basis	**File System**	**DBMS**
Structure	The file system is a way of arranging the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
Data Independence	There is no data independence.	In DBMS data independence exists.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The user has to write procedures for managing databases	The user not required to write procedures.
Sharing	Data is distributed in many files. So, not easy to share data	Due to centralized nature sharing is easy
Data Abstraction	It give details of storage and representation of data	It hides the internal details of Database
Integrity Constraints	Integrity Constraints are difficult to implement	Integrity constraints are easy to implement

- **Instances and schemas:**
  - **Schemas:**
    - The overall design of the database.
    - Does not change frequently.
  - **Instances:**
    - Collection of Information stored at a particular moment.
    - Change frequently.

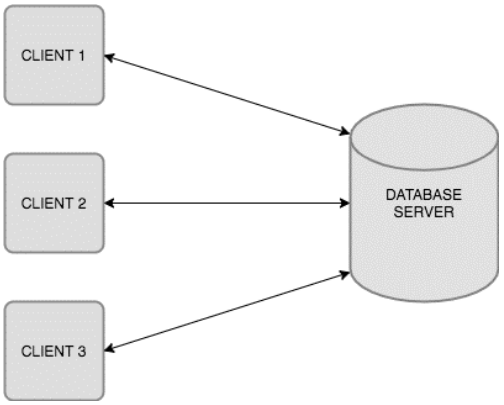
EXAMPLE:

Let A = -5 be a variable.

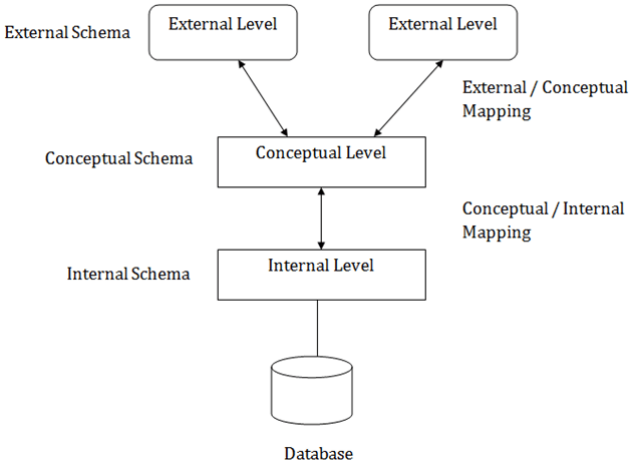
A can be changed frequently such that A = 10, A = 1089 and so on. But even though the value of the variable changes frequently the data type of A remains the same. A schema is like data type of the variable while an instance is like the value of the variable.

- **Database System Architecture:**
  - The database system architecture can be broadly classified into two main types:
    - Two-tier architecture
    - Three-tier architecture (ANSI-SPARC architecture)

- **Two-tier Architecture:**
  - Also known as the client server architecture, consists of two layers client layer and server layer.
  - **Client Layer:**
    - Responsible for user interaction,
    - such as displaying the user interface, collecting user input, and performing data validation.
    - Handles presentation logic and user interface components.
  - **Sever layer:**
    - Also known as database server,
    - Manage database and handles tasks related to data storage, retrieval, and manipulation.
    - Interacts directly with DBMS.



- **Three- schema/ Three-tier/ ANSI-SPARC Architecture:**
  - **Goal:** To separate the user applications and the physical database.
  - **3 levels:**
    - **Internal level:**
      1. Describes the physical storage structure of the database.
      2. Describes complete details of data storage and access paths.
    - **Conceptual level:**
      1. Hides the details of the physical storage structure and concentrates on describing entities, data types, relationships, constraints, etc.
    - **External level:**
      1. Describes the part of the database that a user is interested in and hides the rest of the database from the user group.
  - **Mapping:** This process of transforming request and result between these levels is called as mapping.



- **Characteristics of Data Base approach:**
  - The database approach also known as the database management system (DBMS) approach, is a method of storing, managing, and organizing data in a structured manner.

  - Here are some key characteristics of the database approach:
    - Data Independence
    - Data Integrity
    - Data Security
    - Data sharing and concurrency Control
    - Data Querying and manipulation
    - Data scalability and Performance
    - Data Durability and recovery

- **Data Independence:**
  - **Definition:** Capacity to change the schema at one level of a database system without having to change the schema at the next higher level. E.g., if schema at internal level is changed then schema at conceptual and Exter

  - **Types:**
    - **Logical Data Independence:**
      1. Ability to modify the conceptual schema without changing the external Schema or application programs.

    - **Physical Data Independence:**
      1. Ability to modify the internal schema without changing the conceptual schema.
      2. Changes may be needed to improve performance.

- **Data Integrity:**

  - **Definition:** It is a concept and process that ensures the accuracy, completeness, consistency, and validity of an organization's data.

  - **Integrity Constraints:**
    - These are set of rules.
    - It is used to maintain the quality of information.
    - They ensure that the data insertion, updating and other processes have to be performed in such a way that data integrity is not affected.
    - It is used to guard against accidental damage to the database.

    - Types of Integrity Constraints:
      1. **Domain Constraint:**
         a. Can be defined as the definition of a valid set of values for an attribute.
         b. The data type of domain includes string, character, integer, time, date, currency, etc. the value of the attribute must be available in the corresponding domain.

      EMPLOYEE

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

      2. **Entity constraint:**
         a. The entity constraint states that the primary key value can't be null.
         b. A table can contain a null value other than the primary key field.

      (Table 1)

EMP_NAME	NAME	AGE	D_No
1	Jack	20	11
2	Harry	40	24
3	John	27	18
4	Devil	38	13

Foreign key

Not allowed as D_No 18 is not defined as a Primary key of table 2 and In table 1, D_No is a foreign key defined

Relationships

(Table 2)

Primary Key

D_No	D_Location
11	Mumbai
24	Delhi
13	Noida

      3. **Referential Integrity Constraint:**
         a. A referential constraint is specified between two tables.
         b. If a foreign key in table 1 refers to the primary key of table 2 then every value of the foreign key in table 1 must be available in table 2.

      4. **Key Constraint:**
         a. Keys are the entity set that is used to identity within its entity set uniquely.
         b. An entity set can have multiple keys, but out of which one key will be the primary key.

c. A primary key can contain a unique and null value in the relational table.

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1002	Morgan	8th	22

Not allowed. Because all row must be unique

- **Data Security:**
  - **Definition:** This database approach provides mechanisms to protect data from unauthorized access and manipulation.
  - User authentication, authorization, and access control features allow administrators to define and enforce security policies at various levels, ensuring that only authorized users can access the data.

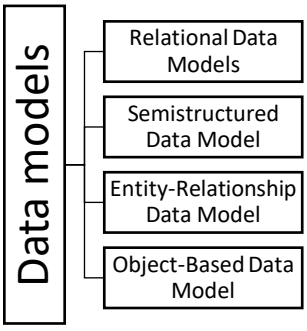
- **Data Sharing and Concurrency Control:**
  - **Definition:** DBMSs allow multiple users and applications to access and modify the data concurrently while ensuring data consistency.
  - Concurrency control mechanisms, such as locking and transaction management, prevent conflicts and ensure that changes made by one user do not interfere with the work of others.

- **Data Querying and Manipulation:**
  - **Definition:** The database approach offers powerful query languages, such as SQL (Structured Query Language), to retrieve, manipulate, and analyse data.
  - These languages provide a standardized and declarative way to express complex queries and operations on the data.

- **Data Scalability and Performance:**
  - **Definition:** DBMSs are designed to handle large volumes of data and support high-performance operations.
  - They provide optimization techniques, indexing mechanisms, and query optimization strategies to enhance data retrieval and processing speed, even with vast amounts of data.

- **Data Durability and Recovery:**
  - **Definition:** DBMSs incorporate features to ensure data durability and recovery in case of failures or system crashes.
  - Transaction logging, backup and restore mechanisms, and recovery procedures are implemented to safeguard data integrity and enable data restoration.

- **RDBMS** (Relational Database Management system)**:**
  - All modern database management systems like SQL, MS SQL Server, ORACLE, My-SQL and Microsoft Access are based on RDBMS.
  - IT's called RDBMS because it is based on the relational model introduced by E.F. Codd.
  - Data is represented in forms of tuples. It is the most commonly used database. RDBMS database uses tables to store data.
  - It is easily accessible due to a collection of organised set of tables, data can be accessed easily in RDBMS.



- **Data Models:** Data Model is the modelling of the data description, data semantics, and consistency constraints of the data.
  It provides the conceptual tools for describing the design of the database at each level of data abstraction.

- **Relational Data Model:**
  - **Definition:** A relational database consists of a collection of tables, each of which is assigned a unique name.
    Tables are also called relations. Primarily used by commercial data processing applications.

  - **Characteristics:**
    - Data is represented in rows and in columns called relations.
    - Data is stored in tables having relationships between them called Relational model.
    - The relational model supports the operations like data definition, Data manipulation, and Transaction management.
    - Each column has a distinct name and they are representing attributes.
    - Each row represents a single entity.

  - **Important Terminologies:**
    - **Attributes:** Attributes are the properties that define an entity.
      E.g.: - RollNo, Name, Address.

- **Relational Schema:** A relational schema defines the structure of the relation and represents the name of the relation and represents the name of the relation with its attributes.
  E.g.: Student (RollNo, Name, Address, Phone, Age) is the relational schema for student.

- **Tuple:** Each row in the relation is known as tuple.

- **Degree:** The number of attributes in the relation is known as the degree of the relation.
  The Student attribute has 5 degree.

- **Cardinality:** The number of tuples in a relation is known as cardinality.

- **Column:** The column represents the set of values for a particular attribute.

- **Null Values:** The value which is not known or unavailable is called a Null Value.

- **Relational Key:** Keys that are used to identify the rows uniquely or also help in identifying tables.

  These are of the following types:

  1. **Primary Key:** A key that helps uniquely identify every tuple in a table.
     a. No duplicate values allowed.
     b. No null values are present in column with primary key.

  2. **Candidate Key:** A set of one or more columns that can uniquely identify a row within a table.
     A table can have multiple candidate key but only have one primary key.

  3. **Super Key:** A set of those keys that identify a row or a tuple uniquely.
     A super key is the super set of candidate key.
     It means a candidate key can only be obtained from super key only.

  4. **Foreign Key:** Is a column that creates a relationship between two tables.
     The purpose of foreign key is to maintain data integrity and allow navigation between two different instances of an entity

  5. **Alternate Key:** All Keys which are not primary keys are known as alternate key.

  6. **Composite Key:** A combination of two or more columns that uniquely identify rows in a table.
     The combination of columns guarantees uniqueness, though individually uniqueness is not guaranteed.

Figures	Symbols	Represents
Rectangle		Entities in ER Model
Ellipse		Attributes in ER Model
Diamond		Relationships among Entities
Line		Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse		Multi-Valued Attributes
Double Rectangle		Weak Entity

1	Ram	Delhi	7889239033	19

- **Constraints in relational model:** While designing the relational model, we define some conditions which must hold for data present in the database are called constraints.

RollNo
1
2
3
4

  - **Domain constraints:**
    These are attribute-level constraints.
    An attribute can only take values that lie inside the domain range.

  - **Key integrity:**
    Every relation in database should have at least one set of attributes that defines a tuple uniquely.

- **Referential Integrity:**
  When one attribute of a relation can only take values from another attribute of the same relation or any other relation, it is called referential integrity.

- **Anomalies in the relational Model:**
  An anomaly is an irregularity or something which deviates from the expected or normal state.
  When designing databases, we identify three types of anomalies: Insert, Update and Delete.

- **ER Model:**
  Entity relational model is a model for identifying entities to be represented in the database and representation in the database and representation of how data is related.

  ER models are used to model real world objects like a person, a car, or a company, and relation between these real-world objects.

  ER diagram is the structural format of the database used to represents the E-R model in a database, which makes them easy to be converted into relations (tables).

- **Components of ER model:**
  - ER model consists of entities, Attributes, and relationships among Entities in a database system.

  - **Entity:**
    - An entity may be an object with physical existence.

    - **Strong entity:**

    A strong entity is a type of entity that has a key attribute.

    Strong entity does not depend on other entity in the schema.

    - **Weak entity:**
      An entity type has a key attribute that uniquely identifies each entity in the entity set.

      But some entity type exists for which key attributes can't be defined.

    - **Example:** A company may store the information of dependents (Parents, children, spouse) of an employee. But the dependents don't have existed without the employee. So dependent will be weak entity and employee will be identifying entity type of dependent, which means it is strong entity type.

  - **Attributes:**
    Attributes are the properties that define the entity type.
    E.g.: RollNo, Name, DOB, Age, Address, and MobileNo are the attributes that define entity type student.

    - **Key attribute:** The attribute which uniquely identifies each entity in the entity set is called the key attribute.

    - **Multivalued Attribute:** An attribute consisting of more than one value for a given entity.

    - **Derived Attribute:** An attribute that can be derived from other attributes of the entity type is known

    - **Composite attribute:** An attribute composed of many other attributes is called