# DBMS

⇒ What is database ?

A database is a structured collection of data, facilitating easy acess, management & updates, generally stored in a computer system.

⇒ Difference between file system & DBMS :

| | File System | DBMS |
|---|---|---|
| 1. Data Access | Slow access of data due to unstructured querying capabilities. | Faster data acess due to Structured querying capabilities. |
| 2. Data Isolation | Challanges in corelation of data accross seperate files. | Facilitates data corelation, reducing data isolation. |
| 3. Data Integrity | Risk of unknown data alteration & deletion | Prevents unauthorized data alteration. |
| 4. Atomicity Problem | Risk of data inconsistancy due to incomplete operations. | Ensures that operations are either completed fully or not at all. |
| 5. Concurrent Access | Inconsistancies due to simultaneous data access | Can manage multiple users to access data simultaneous |

⇒ View of Database:
(i) Physical level
(ii) Logical level
(iii) View level

1. Physical level: (i) Data storage & access in hardware
(ii) Complex structure
(iii) Managed by database administrator
(iv) Most ~~seq~~ secured, cannot be access without authorization.

2. Logical level: (i) Consists of entity sets & their relation
(ii) Connection between stored data in the database.

3. View level: (i) Data abstraction
(ii) Displays only portion of database required by the user
(iii) Allows user to access the data from the database.

⇒ Instances:
Collection of data stored in the database at a specific moment. Shows the current state of records & transactions

⇒ Database Schema:
Refers to overall design of the database, structure of tables & its contents. It defines how data is organized & how the relationships of data are handled. Basically, a blueprint of how a database is structured.

Date
Page

⇒ Types of Database :

1. Commercial Database : Used in business sector to handle large volume of transactions & customer data. Eg, CRM system.

2. Multimedia Database : Stores data type such as images, audio & video files. Eg, Youtube, Adobe Experience Manager

3. Deductive Database : Uses logic programming to derive information stored in a database, allowing more complex & analytical queries.

4. Temporal Database : Keeps track of changing data over time, allowing queries concerning time-based data.

5. Geological Information System (GIS) : Stores, organizes, & analyzes geographical data, useful in mapping projects. Eg, Google Maps.

⇒ Database Administrator :

1. Schema Definition : Administrator outlines the original database schema.
2. Storage Structure & Access Methods : Responsible for forming appropriate storage structures & access methods.
3. Schema & Database Modification : Involves altering of database.
4. Granting authorization for Data Access : Administrator can monitor & give access to different database users.
5. Integrity Constraints : Maintains integrity & consistency throughout the database.

⇒ ER Diagram :
"Entity Relationship" diagram which helps the user to understand complex databases easily using block diagrams. It also shows the relation between different datas in the database.

⇒ Entity :
Smallest piece of information which can be stored in the database, which is also differentiable from other entity

"An entity is a thing or an object which is distinguishable from other object based on the value of the attribute it posseses"

Types of Entity :
(i) Tangible : Physically exists IRL. Eg, Locker
(ii) Intangible : Exists logically. Eg, Account

⇒ Entity Set: Collection of same type of entities that share same properties or attributes.

[ Entity ] ⟶ Entity Set Representation

⇒ Attributes: The unit that defines the properties & characteristics of entities.

⇒ Types of Attributes:

1. Single Valued: Attributes having single value at any instance. Eg, DOB, Aadhar No.

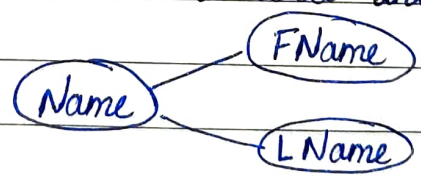
(DOB)

2. Multi Valued: Attributes which can have more than one value. Eg. Phone No., email

(Email)

3. Simple: Attribute which cannot be further divided into sub parts.

4. Composite: Attributes which can be divided into sub parts. Eg, Name.




5. Stored: Attributes whose values are permanantly stored in db.

6. Derived: Values of these attributes can be derived from other attributes. Eg, Age can be derived from DOB.

(Age)

⇒ Relationship: Association between two or more entities of same or different entity set.

⇒ Mapping Cardinalities:

(i) One to One (1:1)
(ii) One to Many (1:M)
(iii) Many to One (M:1)
(iv) Many to Many (M:M)

⇒ Strong & Weak Entity:

(i) Strong Entity: Has a primary key. Strong

(ii) Weak Entity: Does not have a primary key. Weak

⇒ Keys:

1. Super Key: Set of attributes using which we can identify each tuple uniquely is called super key.

2. Candidate Key: Minimal " " " " " " " " called candidate key.

3. Primary Key: It is a unique identifier by which you can access a particular record. It should not contain null values. There should be exactly one Primary Key per table.

4. Foreign Key: It acts as a reference to the primary key of same or other table to represent relationship.

5. Secondary Key: It can also act as a primary key to speed up data retreival. It does not necessarily contains unique values.

6. Composite Key: Key composed of more than one attribute.

## ⇒ Introduction to SQL:

- It is a domain specific language.
- Used in programming & managing data in DBMS.
- It can define structure of database, modify, specify security constraints.

## ⇒ Classification of database language:

1. Data Definition Language (DDL):
   - Set of commands used to create, modify & delete database structure & not data.
   - It is used by Database Administrator.
   - CREATE, ALTER, DROP, TRUNCATE, COMMENT are few commands of DDL.

2. Data Manipulation Language (DML):
   - Set of commands that enables users to access or modify data.
   - There are two types of DML:
     (i) Procedural: Requires user to ~~sy~~ specify what data is needed & how to get those data.
     (ii) Non-Procedural: Requires user to specify what data is needed without specifying how to get those data.
   - INSERT, UPDATE, DELETE, are few DML commands.

⇒ Create a Table:

```
CREATE TABLE Students (
    StudentID   INT         Primary Key,
    FirstName   VARCHAR (50),
    LastName    VARCHAR (50),
    Email       VARCHAR (100)
);
```

1. Adding new column:

```
ALTER TABLE Students
ADD PhoneNumber VARCHAR (15);
```

2. Drop / Delete a column:

```
ALTER TABLE Students
DROP COLUMN PhoneNumber;
```

3. ~~Modif~~ Modifying an existing column:

```
ALTER TABLE Students
MODIFY COLUMN PhoneNumber (20);
```

4. Renaming a column:

```
ALTER TABLE Students,
RENAME COLUMN PhoneNumber TO ContactNumber;
```

5. Renaming a table :

```
ALTER TABLE Students
RENAME TO Staff ;
```

6. Drop a table :

```
DROP TABLE Students ;
```

7. Adding Foreign Key:

```
CREATE TABLE Orders (
  OrderID INT Primary Key,
  CustomerID INT,
  Foreign Key (CustomerID) REFERENCES Customer (CustomerID)
);
```

⇒ Inserting Values:

1. INSERT INTO Students (StudentID, FirstName, LastName)
VALUES
(1, 'Amit', 'Sharma'),
(2, 'Ojas', 'Chatur');

⇒ Deleting Values:

1. DELETE FROM Students
WHERE StudentID = 1;

⇒ Show table values:

1. ~~Select~~ SELECT * FROM Students
WHERE StudentID = 1;