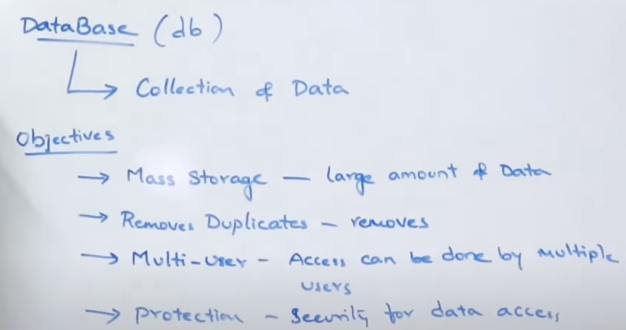
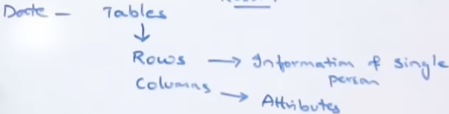
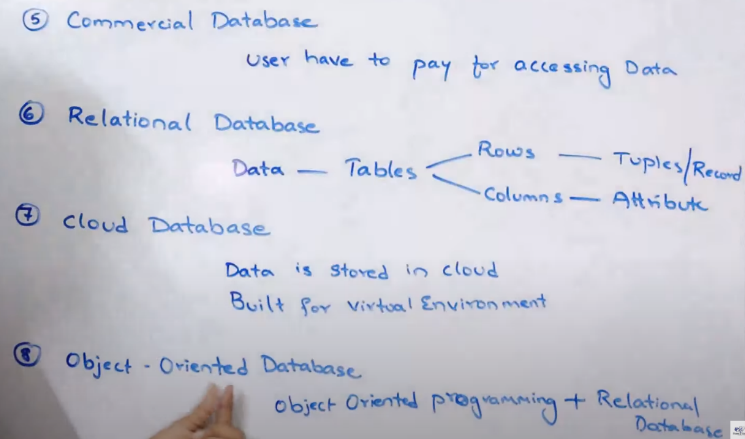
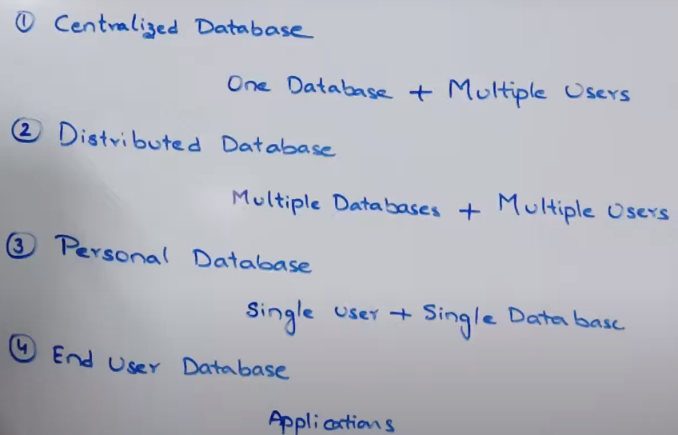
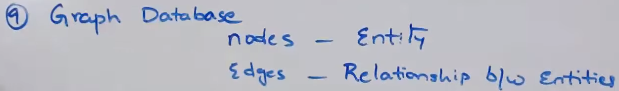
**DBMS**

Data is an unorganised/unprocessed Information and Database is used to organize the and store the Data and DBMS is a software/tool used to manipulate/manage the Database (Eg, Oracle,Mongo DB, My SQL,etc).

****

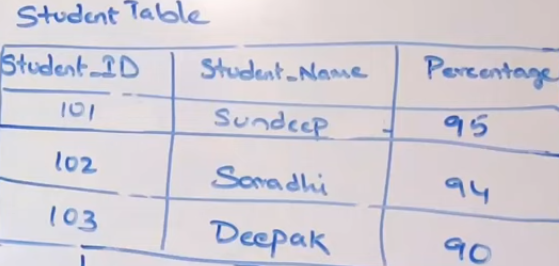
In DBMS complete data is stored in the form of Tables.

******Types of Databases**

****

**RDBMS**(Relational Database Management System)

In RDBMS data is stored in the form of tables where each row is called as Tuples/Records and Columns as Attribute.

**Tuples rules**: 1. Duplicates are not allowed in Tuples mtlb assume tuple 1 me sundeep ka data h and same data not allowed in other Tuples.

2. Tuples should follow same format.(Eg tuple me id, name, percent h to agle tuple me bhi yahi order se hona chahiye)

**Attribute rules:** 1. Should have name (vo attribute kya store kr raha h). Above Attribute 1 stores Student\_Id, 2 -> name, 3-> Percent.

2. To identify a tuple there is a unique attribute called Primary key which stores only unique and not null value. Eg, above id is a primary key since id is unique for every student but not the name and percent.

**Basic Terminology in RDBMS**

**Data Item/ Cell Data** Intersection of Tuple and attribute. (hr ek dabbe me stored data). Eg 101, Deepak, 95,etc. It must have single value only. Eg Deepak,Ramesh ek cell me nhi ho sakte.

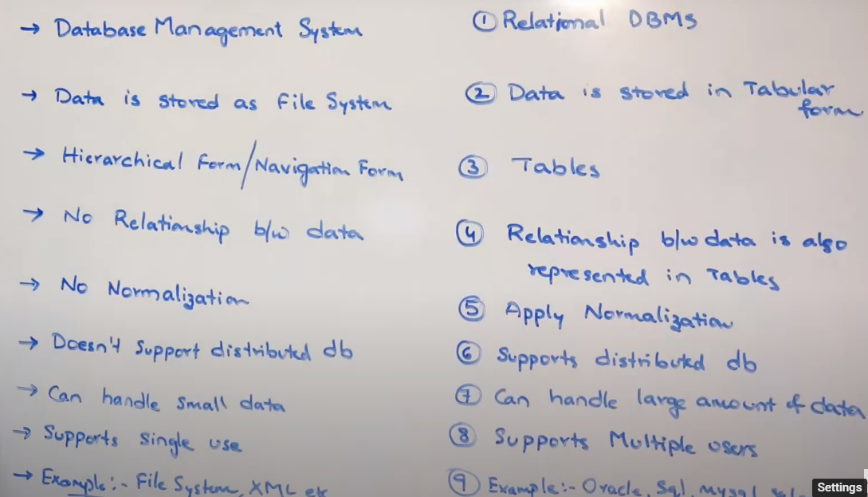
**Cardinality:**  No. of Tuples in a table. Eg above 3

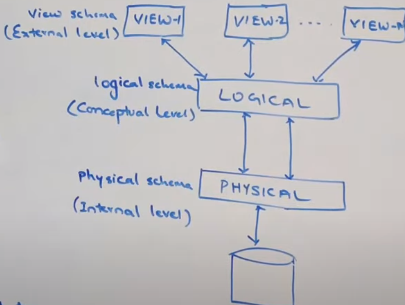
**Degree :**  No. of Attributes in a table. Eg above 3

**Domain:** Range of values from which value for an attribute is selected. Eg, In percent Domain -> (0,100)

**NULL Value:** Cell without any value is set to NULL. It doesn’t mean 0 or blank space. Primary key can’t be set to NULL.

**Difference between DBMS and RDBMS**

****

**Schema & Instance**

Schema is a structure/design which the database follows

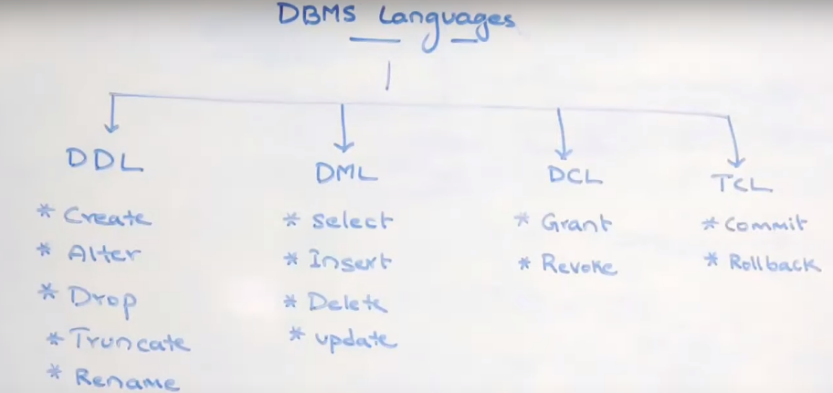
Types of Schemas:

1. **Physical Schema** :- It describes how the data is stored in the Storage system. Eg file system, indices etc
2. **Logical Schema** :- defines the structure of the data itself and the relationships between the various attributes, tables, and entries. (Table kaisa dikhta h database me ye samjh le).
3. **View Schema:** How the data is visible to the end user. (Interface ki tarah samjh le)

Instance is a view of the database at a particular instance of time. (abhi database dekhne jaau to bol sakta hu abhi kya dikh raha h vo mera instance h).

It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information.

A database instance is a state of operational database with data at any given time. It contains a snapshot of the database. Database instances tend to change with time. A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed

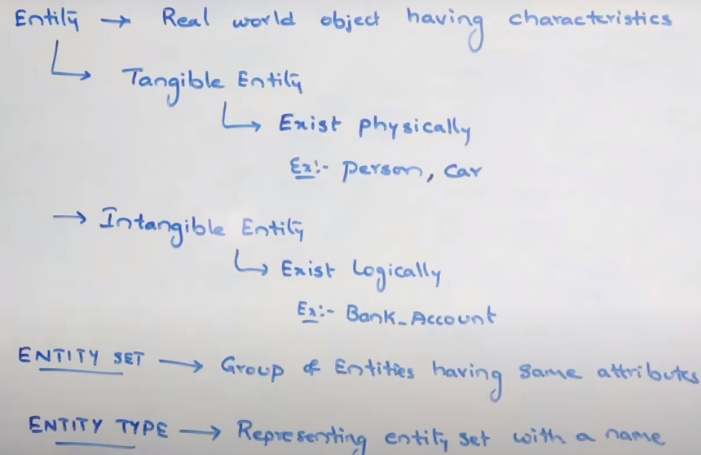
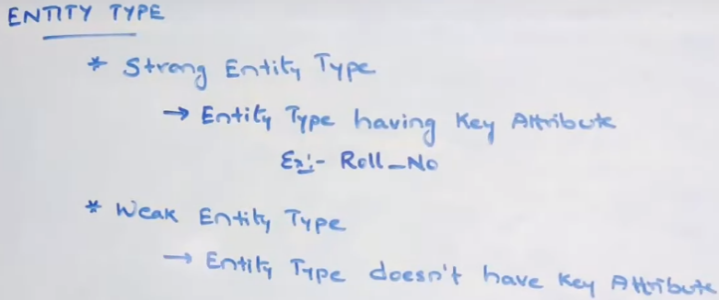
**DBMS languages**

1. Data Definition Langauge (DDL)

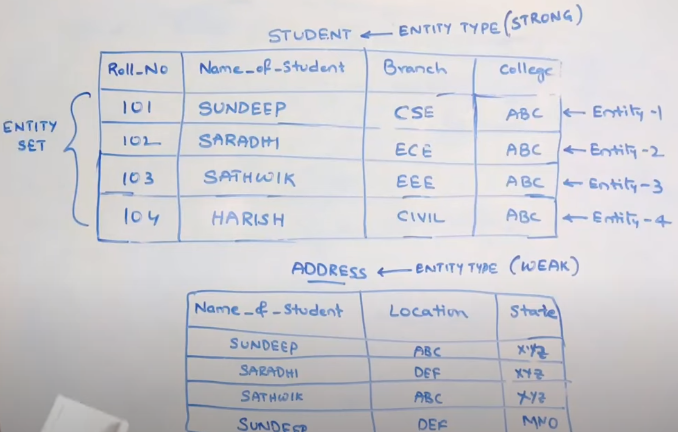
## Data Manipulation Language (DML)

## Data Control Language (DCL)

## Transaction Control Language (TCL)

**Entity**

Note: Weak Entity takes help of Strong Entity to identify the records.

Eg,

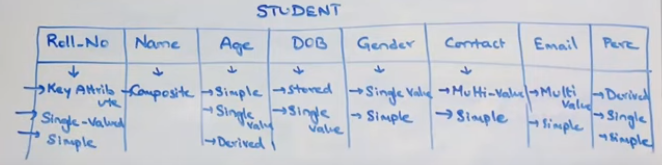
Here each row can be called as entity and saare rows have same attributes therefore we can call it Entity set. Entity set ka naam h STUDENT to ye hua Entitiy type.

Since 1st table has primary key. Therefore, Strong Entity type and 2nd table is Weak Entity type.

**Attribute**

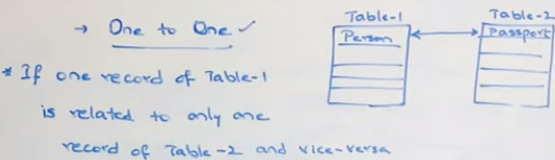
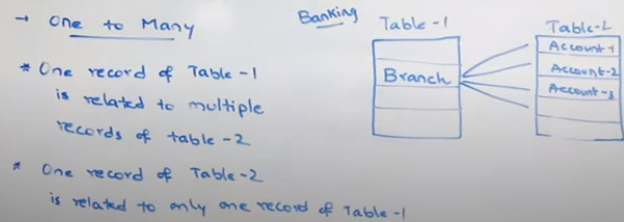
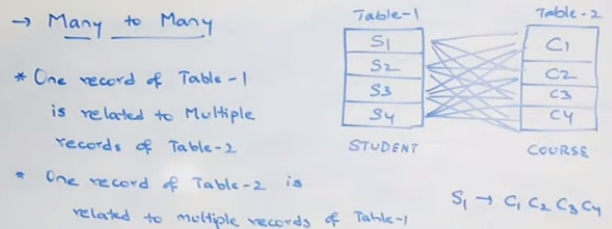
It is a property or characteristics that describes an entity.)

Types of Attributes:

1. **Single Valued:** Attribute which can hold single value only. Eg age, gender, roll no etc
2. **Multivalued:** Attribute which can hold multiple values. Eg, email, skillset.
3. **Compound/Composite:** Attribute that can be subdivided into two or more other attributes. Eg name can be divided into First name, middle name and last name.
4. **Single/Atomic:** Attribute that can’t be subdivided into smaller parts. Eg age, mobile, email, etc.
5. **Derived:** An attribute that can be derived from another Attribute. Eg age can be derived from DOB, percentage from marks, etc.
6. **Stored:** data that remain constant and fixed for an entity instance. These values help in deriving the derived attributes.
7. **Key:** used to identify an entity uniquely. Every primary key is a key attribute. Eg roll no, pid, etc.

**Relationship**

Relating one table with one or many table is termed as Relatioship.

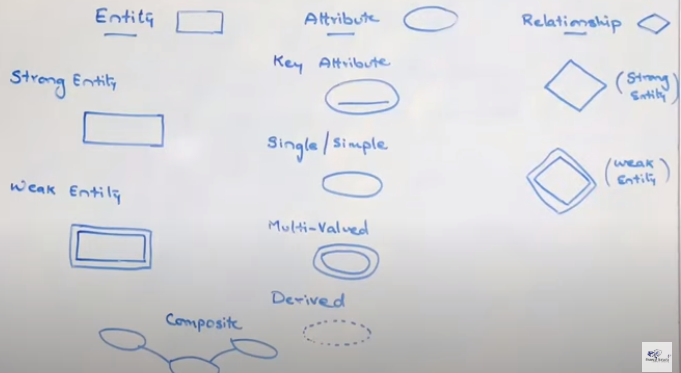
There are 3 types of Relationship.

**ER-Model**

Pictorial representation of a database showing Entity type and their relationships.

Components of ER-Model: 1. Entity 2. Attribute 3. Relationship

**Notations in ER-Model:**

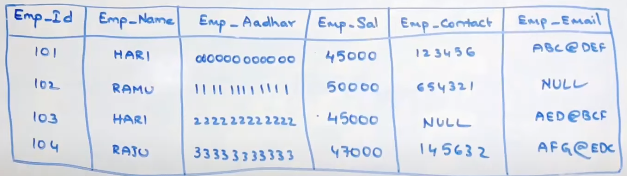


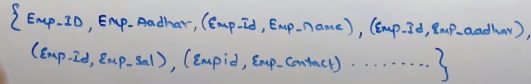
**Keys**

It is an Attribute/set of attributes which uniquely identifies a tuple.

Types of Keys.

1. **Super Key**: Attribute/set of attributes which uniquely identifies a tuple.
2. **Candidate Key**: Minimal set/combination of super keys.
3. **Primary Key**: A key from candidate key which is considered as Primary key. Note: any 1 key can be considered as Primary Key.
4. **Alternate Key**: All candidate key except primary key.
5. **Unique Key**: Same as Primary key but can have NULL value.
6. **Foreign Key**: Attribute in a table which points at Primary Key from another table. It basically, relates one table with one or many table. (Ellaborated more in SQL notes).

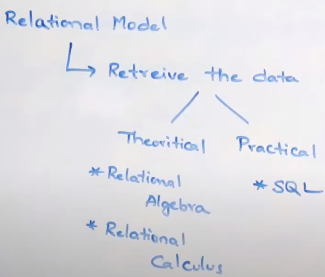
Consider the below mentioned table for eg,

 **Super Key =>**

**Candidate key** => **{Emp\_Id and Emp\_Aadhar}** (since they are not repeated. The set of Emp\_Id and Emp\_name and other sets can’t be called as Candidate key as Emp\_Id is already used. Basically, we cannot use duplicate attributes as candidate key).

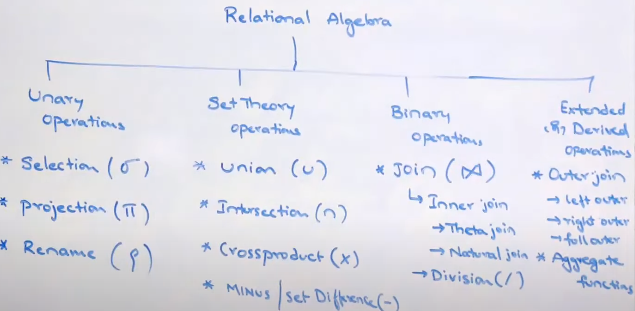
**Note**: Combinations of keys can also be termed as Candidate key bss they must not be repeated before.

**Unique Key: =>** Emp\_Contact and Emp\_Email.

**Relational Algebra**

It’s a procedural query language that accepts a Relation as input and outputs another Relation.It is a theoritical language and base for SQL.

**Types of Operators in Relational Algebra**

****