

\* Characteristics of DBMS approach :-

- ① Self-describing nature of a Database system :-
- ② Isolation between programs and data, and data abstraction :-
- ③ Support for Multiple views of the data :-
- ④ Sharing of Knowledge and multi-user Transaction processing :-

\* Users of the database system :-

- ① Actors on the scene
- ② Workers behind the scene

① Actors on the scene :-  
 (people whose jobs involve the day to day use of large database)

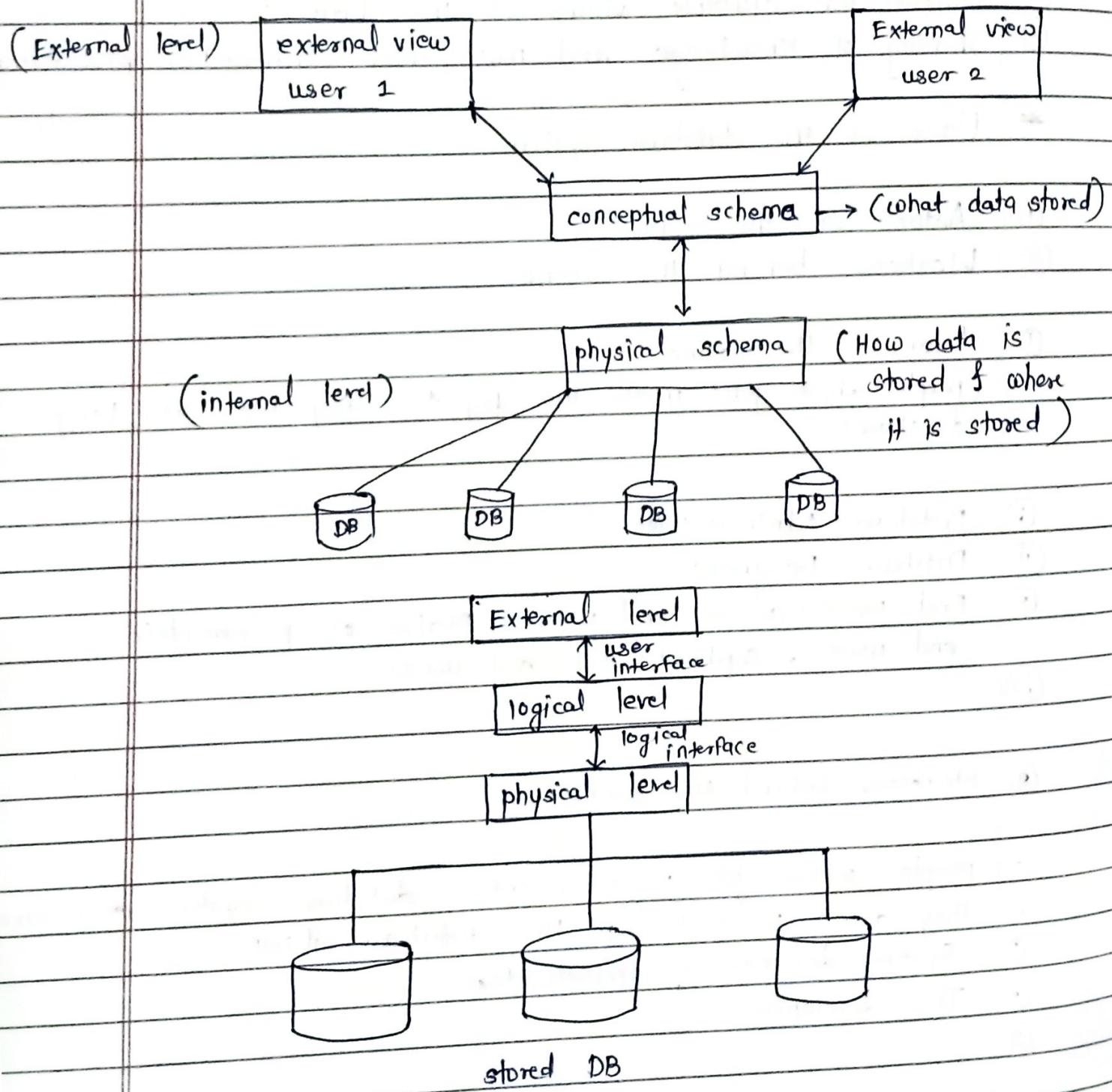
- i) Database Administrators
- ii) Database Designers
- iii) End users - casual end users, Naïve or parametric end users, Sophisticated end users
- iv)

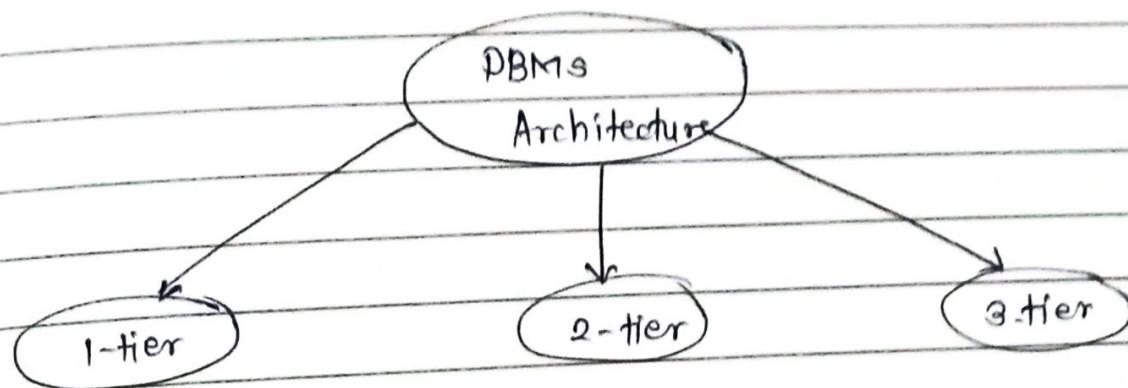
② Workers behind the scene :-

- people whose job is to maintain database system environment.
- They aren't interested in the database itself.

- ① System designers & implementers.
- ② Tool developers
- ③

- \* Advantages of DBMS and Disadvantages of DBMS :-
- \* Data model schema and instance :-
- \* The three-schema architecture is as follows :-  
(This topic is required for data independence).





### \* Course Objectives :-

1. To learn and practice data modelling using the entity-relationship and develop database design.

### \* Degree of relationship :-

① The no. of entity types which took part in the entity relationship is called the degree of relationships.

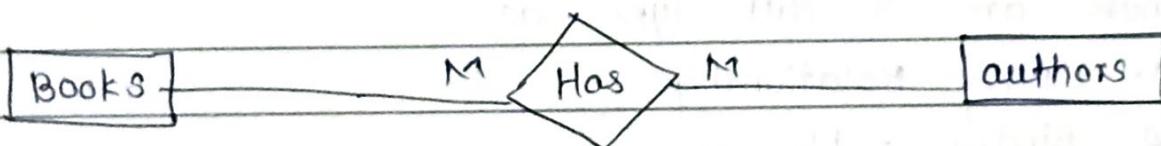
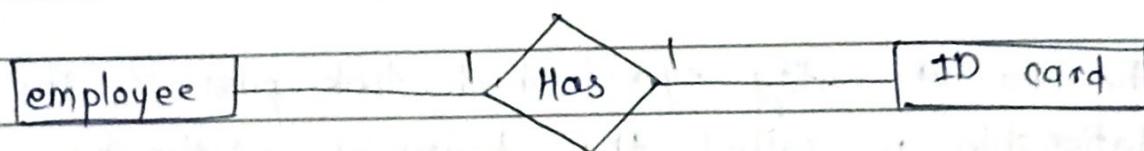
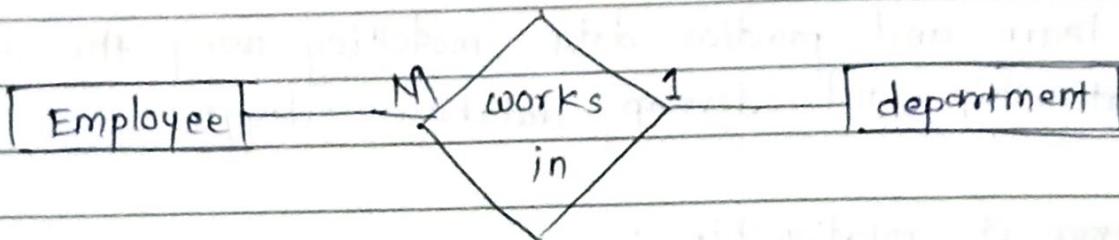
② There are 3 diff. Types as :-

1. Unary Relationship
2. Binary → →
3. Ternary → → →
4. N-any → → →

### 1. Unary / Recursive Relationship :-

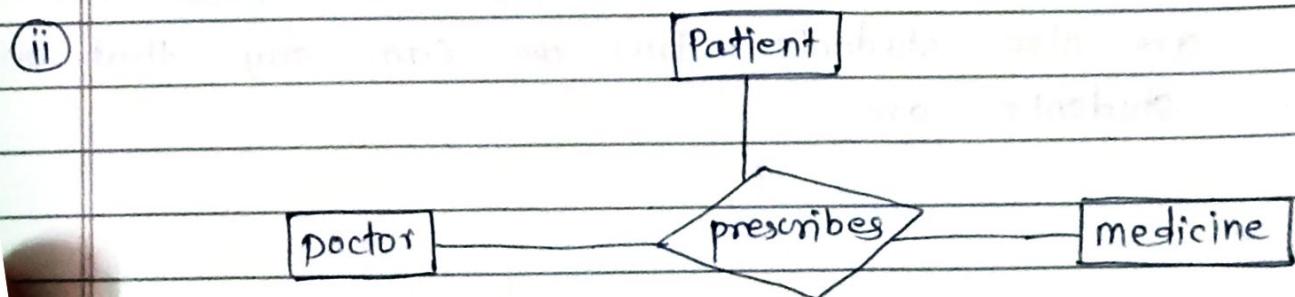
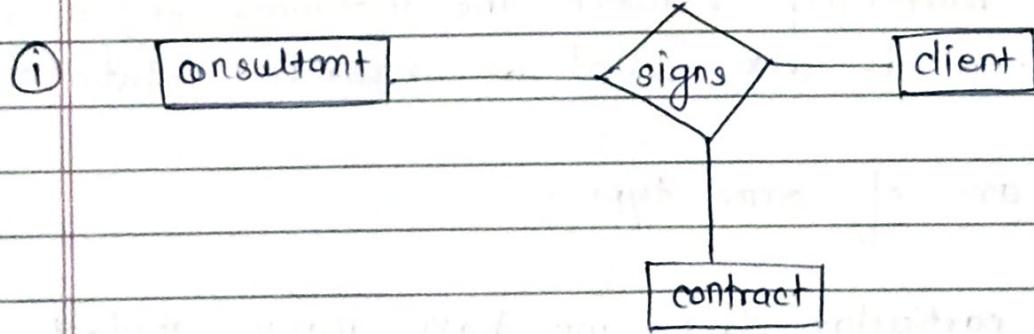
- it is the relationship between the instances of a single entity type. it is also called as recursive relationship.
- 2 entities are of same type.
- e.g. in a particular class , we have many students , there are monitors too , so , here class monitors are also students . Thus we can say that only students are

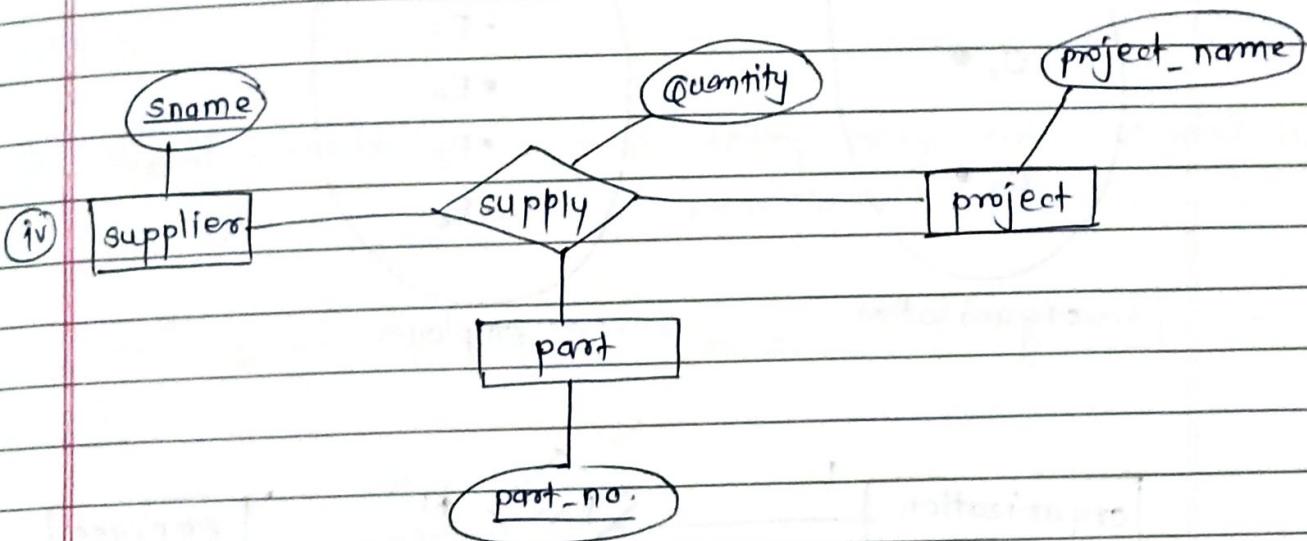
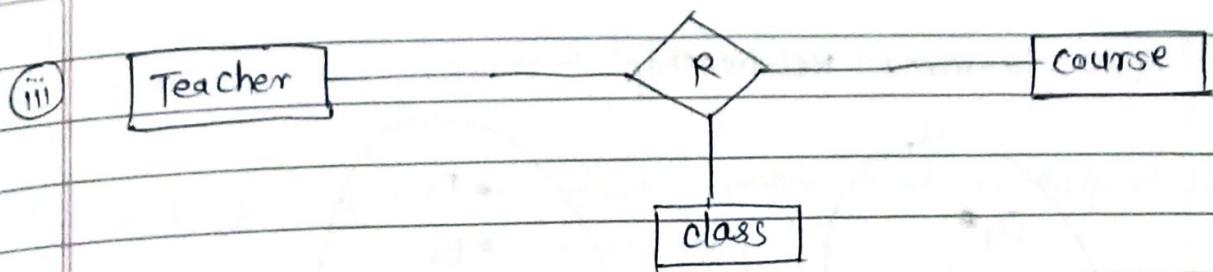
⑧ Binary : Relationship bet<sup>n</sup> two diff. entities



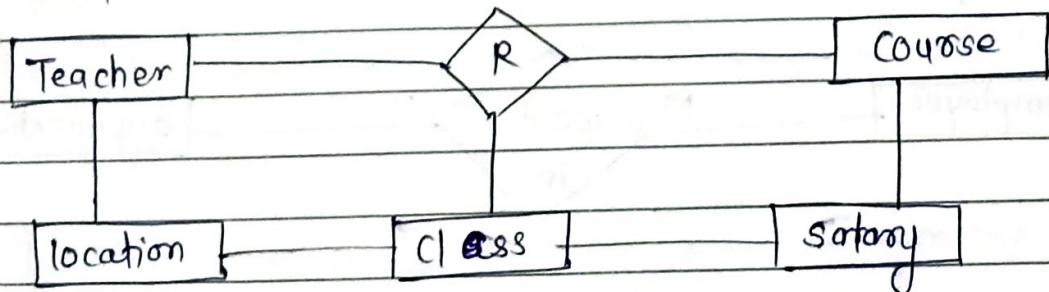
⑨ Ternary relationship

- Relationship bet<sup>n</sup> 3 diff. entities.





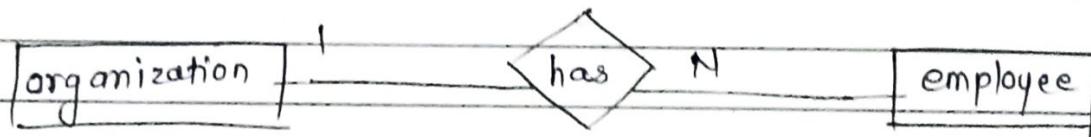
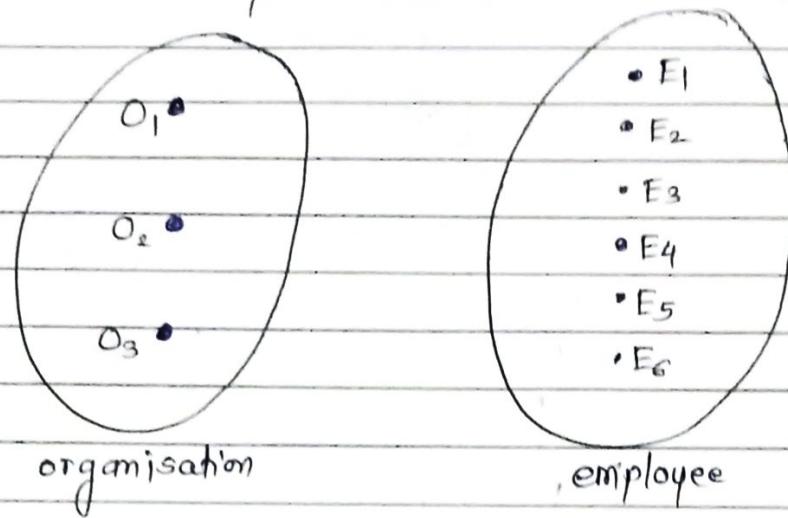
#### ④ N-ary Relationship :



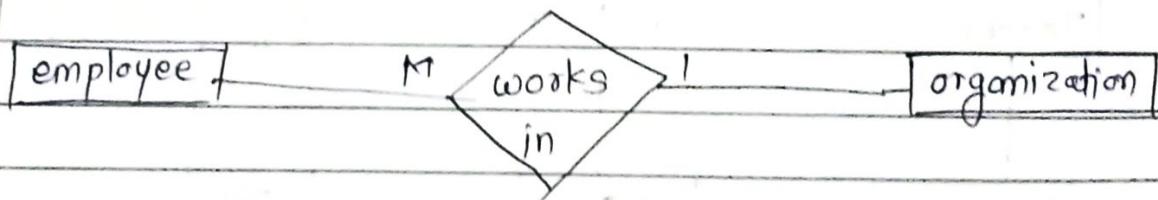
#### ① One to One Relationship :

One employee is assigned with only 1 parking space or 1 parking space is assigned to only 1 employee. Hence it is a 1:1 relationship and cardinality is one to one.

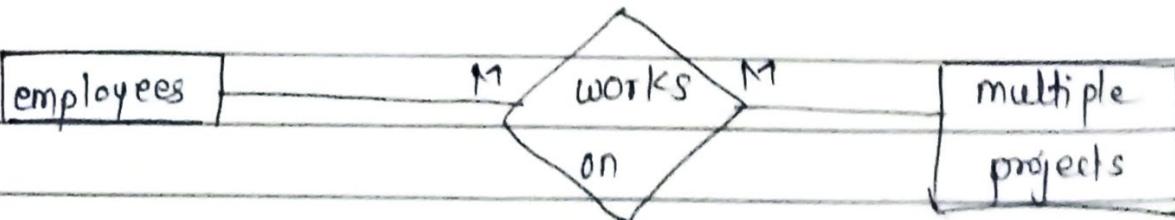
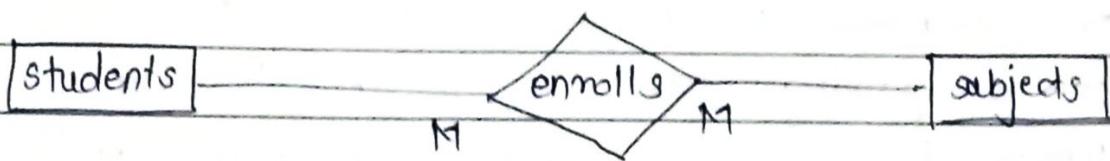
② One to many Relationship :-



③ Many to one :-



④ many to many :-

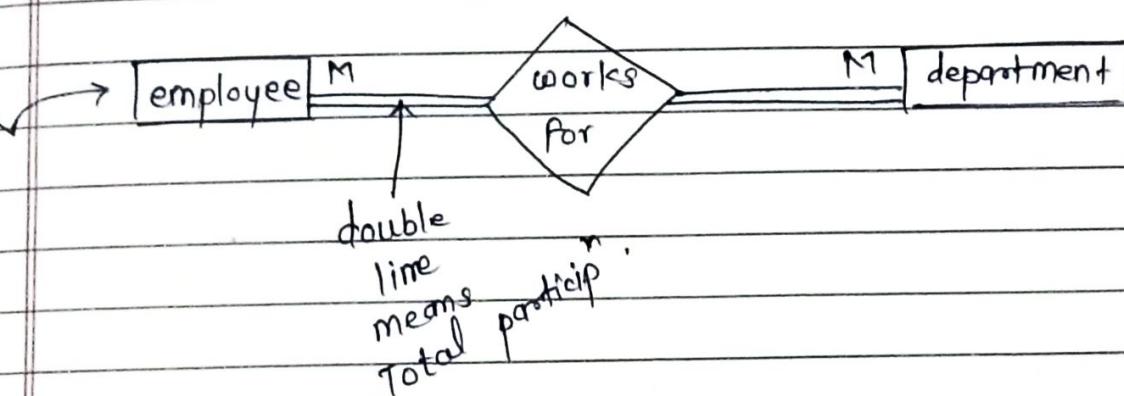


\* participation constraints / minimum cardinality constraint :-

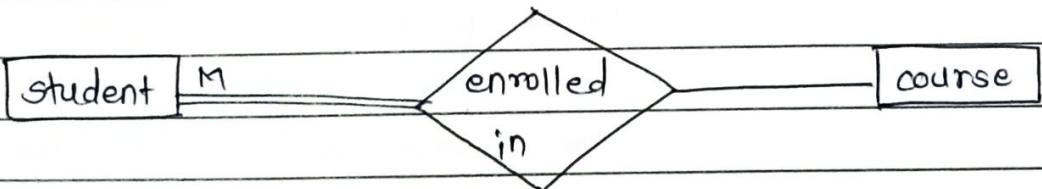
① Total participation :- each entity must participate in the relationship.

② Partial participation :- each entity may not participate in the relationship.

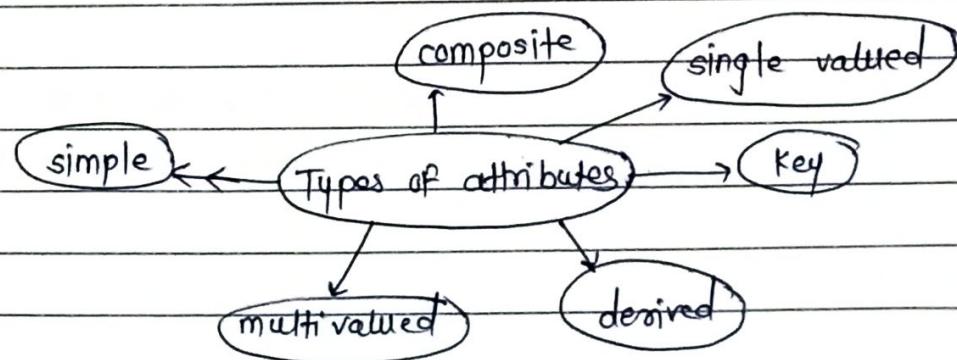
② Partial participation :- each entity may not participate in the relationship.



e.g.



# Types of Attributes :-



What are keys in DBMS ?

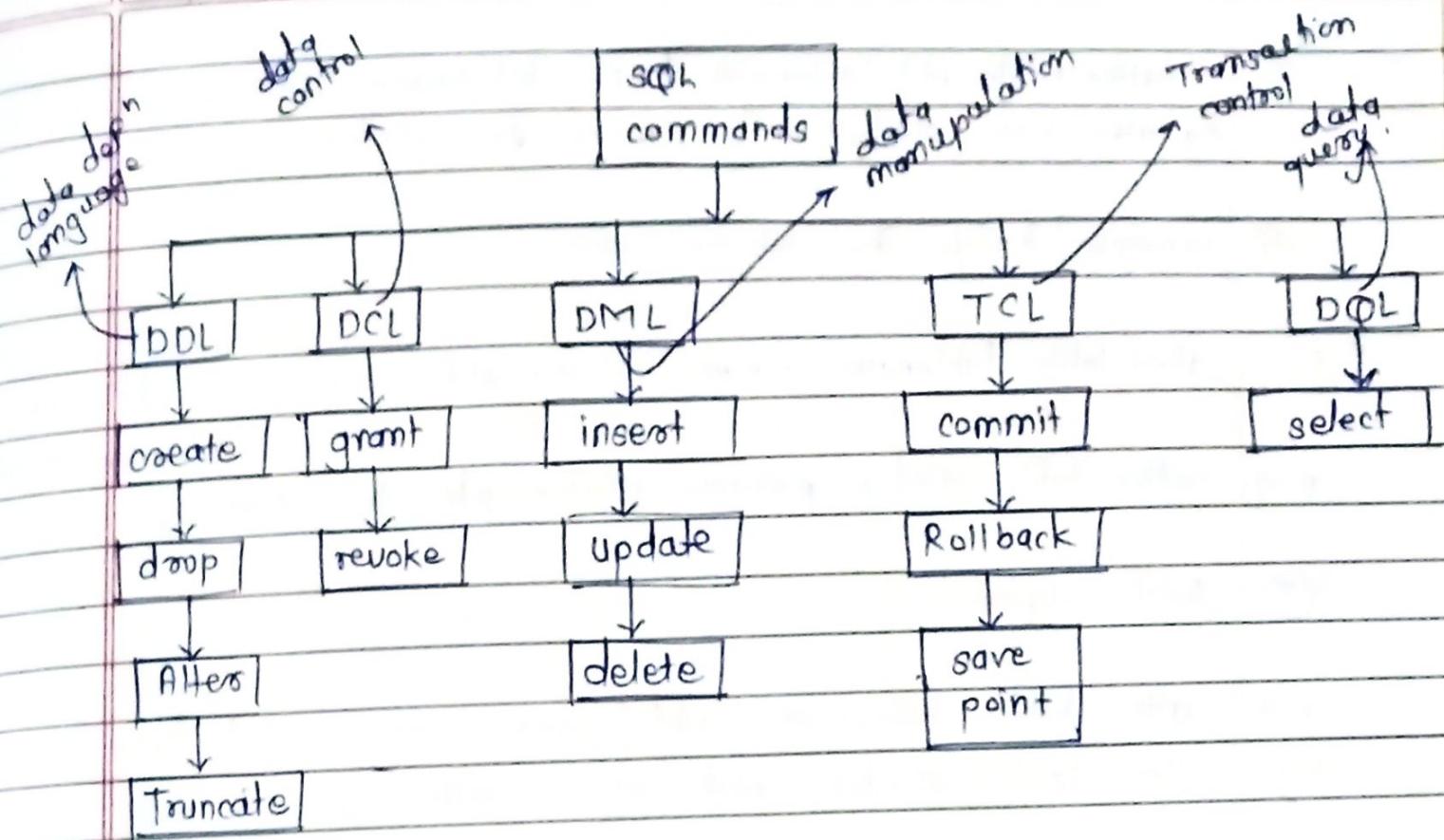
- D KEYS in DBMS is an attribute or set of attributes which helps you to identify a row (tuple) in a relation (table)
- ② They allow you to find the relation between two tables.
- ③ Keys helps you uniquely identify a row in a table by a combination of one or more columns in that table.
- ④ Key is also helpful for finding unique record or row from the table.

# Keys in DBMS :-

- ① Superkey :- it is a group single or multiple keys which identifies rows in a table.
  - it is superset of candidate key.
  - A table can have many super-keys.
  - uniquely identify the row from the table which are combination of all possible attributes .
- ② Candidate key / minimal  $\rightarrow$  super key / super key with no redundant attributes :-
- It must contains unique values
- It must not contains null values .
- ③ Primary Key (unique + Not null) :- it is a column or group of columns in a table that uniquely identify every row in that table .

- (4) Alternate Keys :- it is a column or group of columns in a table that uniquely identify every row in that table.
- (5) Foreign key :- it is a column that creates a relationship bet<sup>n</sup> two tables.

## Structured Query Language (SQL).



① create :-

ex. :- create table branch  
 (branch-name char(15);  
 branch-city char(30);  
 assets integer);

② drop And Alter Table :-

~~drop-table~~

drop table tablename ; (delete the table)

③ Alter command :-

(rename table , column)

(delete table , column)

(add table , column)

(change the datatype)

alter table old tablename to new tablename ;  
 e.g. alter table student info rename to student ;

④ rename / change the column name :-

e.g. alter table tablename rename column old columnname to new  
 columnname;

e.g. alter table student rename column ph to phoneno;

⑤ Add column :-

e.g. alter table tablename add (column name - data type) ;

e.g. alter table student add (prn- number) ;  
 (12)

⑥ Delete column :-

e.g. Alter table tablename drop column columnname

⑦ Modify the datatype of attribute .

e.g. Alter table tablename modify (column name new datatype);

⑧ Truncate : delete all the rows :-

Syntax :-

TRUNCATE Table tablename ;

e.g. :- TRUNCATE Table Employee ;

⑨ Data manipulation language :-  
 language not auto committed .

\* Data Query language :-

(1) DML - INSERT :-

• Inserting Data / Values into a table :-

INSERT INTO student values (1, 'vishakha');

INSERT INTO student values (2, 'John');

- all rows in a table are inserted using the insert command.

e.g:- - Inserting All values

insert into emp values (7315, 'John', 'clerk', 7801, '10-Jun-97', 2500, NULL, 30);

- inserting only some values

(2) Update command :-

- 1) it is used to modify column values in table.
- 2) values of single column or a

syntax :-

UPDATE student

SET name = 'vishakha'  
[Where PRN = 30];

generally update  
command is  
used with

where clause .

### ③ DELETE :-

- Rows can be deleted using DELETE command.
- syntax :-

```

SELECT emp-name          DELETE from student
  FROM employee           where PRN = 30 ;
 WHERE PRN = 30 ;
  
```

### \* Data Query language :-

Syntax :-

```

SELECT emp-name
  FROM employee
 WHERE PRN = 30 ;
  
```

- it is used to fetch the data from database.

### \* Data control language :-

- grant and remove are the commands used in DCL.

#### ① Grant command :-

1stly to give the permission to user, we have to use this to access command from user.

#### ② Revoke command :-

- To remove a data it is used

```

REVOKE privilege-name on objectname from user ;
  
```

## \* Transaction control language :-

- it is only used with DML commands to insert, delete or update data.

(1) Commit command :- it is used to save all the transactions.

Syntax :- COMMIT ;

ex :- DELETE FROM CUSTOMERS  
WHERE AGE = 20 ;  
COMMIT ;

(2) ROLLBACK :- it is used to undo transactions that have not already been saved to the database.

Syntax :- ROLLBACK ;

ex :- DELETE FROM CUSTOMERS  
WHERE AGE = 20 ;  
ROLLBACK ;

(3) SAVEPOINT :- it is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax :-

SAVEPOINT SAVEPOINT\_NAME ;

ex :- DELETE from student  
WHERE AGE = 20 ;  
SAVEPOINT SP<sub>1</sub> ;

\* SQL Set Operation :-

- The SQL set operation is used to combine the two or more SQL Select statements.
- The number and order of columns must be the same.
- Data types must be compatible.

\* Union :- eliminate the duplicate rows (same) .

Syntax :-

```

(*) Select column-name from table 1
Union
Select column-name from table 2 ;
(*)
```

\* Union All :- without removing duplication and sorting data .

Syntax :-

```

(*) Select column-name from table 1
Union All
Select column-name from table 2 ;
(*)
```

\* Intersect :-

- Returns the common rows from both .

Syntax :-

SELECT column-name from table 1

INTERSECT

SELECT column-name from table 2 ;

\* Minus / Except :- display the rows present in 1<sup>st</sup> query  
absent in 2<sup>nd</sup> query.  
it has no duplicate and data arranged in ascending  
order by default.

Syntax :- (\*)

SELECT column-name From table 1

MINUS

SELECT column-name from table 2 ;  
(\*)

\* SQL order by keyword :-

- Order by syntax :-

SELECT column 1 , column 2 , ...

from table-name

Order By column 1 , column 2 , ... ASC | DESC ;

e.g.	Country	Name
	India	Chamay
	china	Neha
	Afghanisthan	Sujt
	America	Priya
	India	Amol
	America	Hitesh

Select \* from customers

ORDER by Country ;

\* Group by Clause : Arrange the identical data into rows.

Select column 1 , column 2

From table-name

where (condition)

Group by column 1 , column 2

Order by column 1 , column 2

Query 1:  
Select Department , sum (salary) as salary  
FROM Employee  
Group by department ;

\* Having Clause :-

It enable you to specify conditions that filters which group results appear in the result .

SELECT column 1 , column 2

FROM table 1 , table 2

WHERE (condition)

Group By column 1 , column 2

HAVING Clause

### \* Domain constraints :- (attribute)

- Domain constraints can be defined as defin' of a valid set of values for an attribute .
- The data type of domain includes string , character, integer , time , date , currency , etc . The value of attribute must be available in the corresponding domain .

(number) (varchar) (varchar) (number)

ID	Name	Semester	Age
1000	Tom	1 <sup>st</sup>	17
1001	Johnson	2 <sup>nd</sup>	24
1002	Leonardo	5 <sup>th</sup>	21
1003	Kate	3 <sup>rd</sup>	19
1004	Morgan	8 <sup>th</sup>	A

→ not allow because  
it is varchar .

### \* Entity integrity constraints :- (attribute)

- The entity integrity constraints states that 1<sup>o</sup> key value can't be null .
- This is because the 1<sup>o</sup> key value is used to identify individual rows in relation and if the 1<sup>o</sup> key has a null value , then we can't identify those rows .
- A table can contain a null value then we can't identify those rows .

A table can contain a null value other than the 1<sup>o</sup> key field .

### \* Key constraints (attribute) :-

- Keys are the entity set that is used to identify an entity within its entity set uniquely.
- An entity set can have multiple keys, but out of which one key will be the 1<sup>o</sup> key.
- A 1<sup>o</sup> key can contain a unique and not null value in the relational table.

(1<sup>o</sup>key)

ID	Name	SEM	age
1000	vish	1 <sup>st</sup>	17
1001	siddhi	2 <sup>nd</sup>	24
1002	neha	5 <sup>th</sup>	21
1003	isha	3 <sup>rd</sup>	19
1002	leena	8 <sup>th</sup>	22

↑  
not  
allowed

because  
already  
used.

### \* Referential integrity constraints : (Relation) :-

- it is specified bet" 2 tables.
- This constraint is enforced when a foreign key references the primary key of table.
- in the referential integrity constraints foreign key is in Table 1 refers to the 1<sup>o</sup> key of Table 2 . then every value of Foreign key in Table 1 must be null or be available in another Table 2 .