UNIT-1.

Introduction to DBMS

(Database management system)

DBMS:- It contains information about particular interprice

→ it is a collection of interpretated data
→ set a programs to access there data.
→ it provide an environment that is both convenient and efficient to use

" It is an organised collection of related data typically stored in a disk and accessible by many concurrent uses.

Database are generally sparated into application areas.

for example:- Database may contain human resources data another may contain sales data and accounting data and so on.

Database managed by database management System

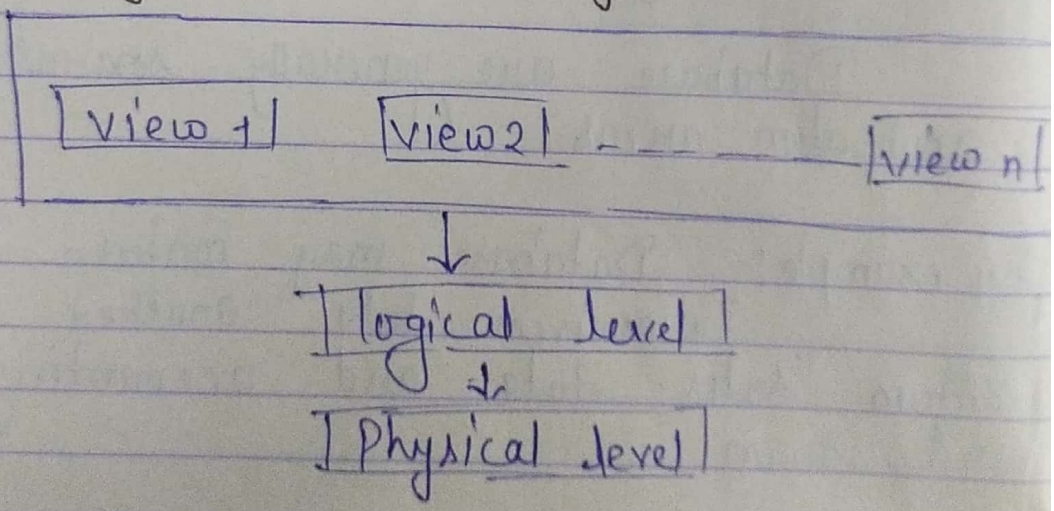
Relation = To read any data base called Relation (table)

Purpose of DBMS :- It is developed to handle the following different of typical file processing system supported of conventional operating system

1) Data Redundency and inconsistency :-
2) Difficulty in accessing data.
3) Data isolation
4) Integrity problems
5) Atomicity of updates.
6) Concurrent access by multiple

Database Applications :- (By own)

View of data / level of abstraction



(i) Physical level → It describes how a record is stored.
(ii) logical level → It describes data stored in database and the

relationship among the data

(iii) View level ⇒ App. programs to hide the details of data-type views can also hide information for security purposes

⇒ Data abstraction:- It is a reduction of a particular body of data to a simplified representation of the whole.

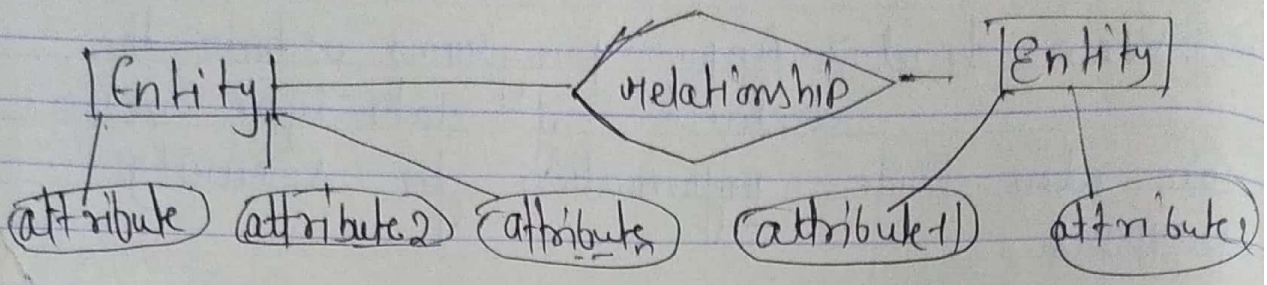
A process to removing characteristic from something in order to reduce it to a set of essential characteristics

Entity - Relationship Model:- ER Model is based on the real world entities and relationships among them.

The E-R model creates entity set relationship set general attributes and constrainted (limitation)

E-R model is best use of a Conceptual design of a database.

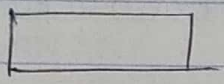
What is view level and data abstraction.



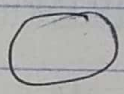
3. Symbols and notation of E-R diagram

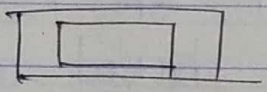
4)
5)
6)

Date

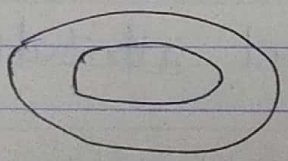
 → Entity

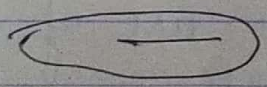
 → Relationship

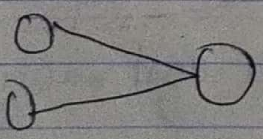
 → Attribute

 → weak entity

 → weak entity relation

 → multivalued

 → key attribute

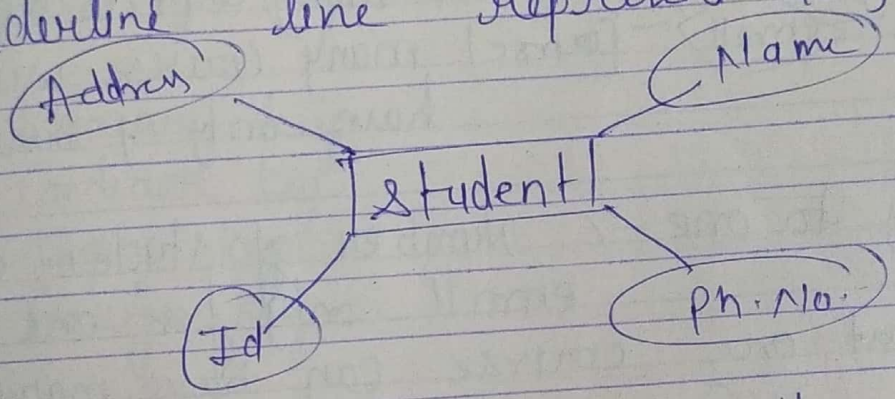
 → Composite attribute

27

Assignment:- E-R diagram (monday due)
Data models.

Weak Entity:- weak entity depends on another entity it doesn't have any key attribute of there own.

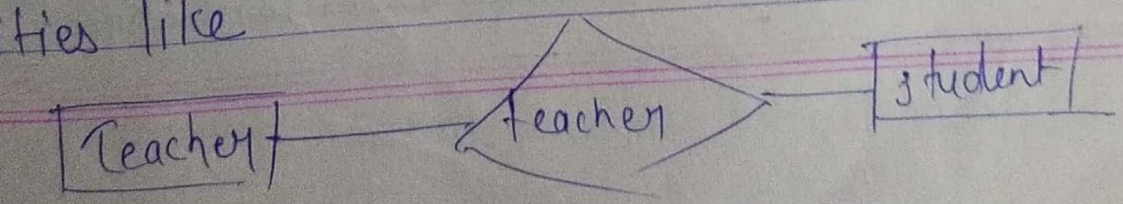
Key attribute:- Key attribute represent the main characteristics of an entity it is used to represent the primary key Ellipse with underline line represent key attribute



Attribute:- An attribute describe a property or characteristics of an entity

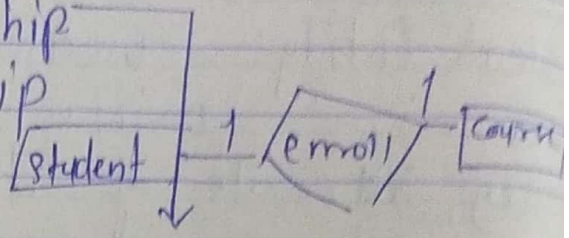
Composite attribute:- An attribute can also have there own attributes these attributes known as composites attribute

Cardinality of relationship:- It describes relations B/w entities like



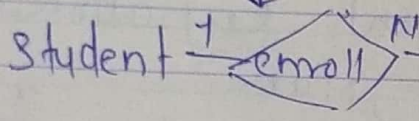
there are 3 types of relationships

1) Binary relationship
ii) Recursive relationship
iii) Ternary relationship

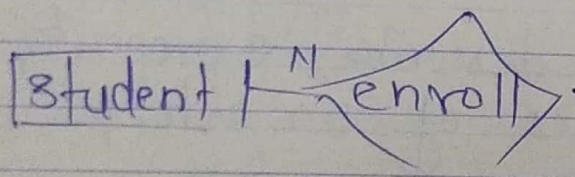


In a binary relationship has one-to-one entity relationship such as → one student can enroll only one course at a time

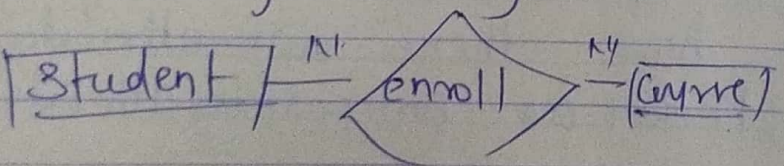
→ one-to-many → 1 student can enroll to many courses but many course will have only 1 student



→ Many to one → Number of student can enroll only for one course but one course can have many student



→ Many to Many →



Recursive relationship:- when an entity is related to itself it is known as recursive relationship

Record → Row - Tuples
Colum - Attribute
entity
expence

Keys:-

A key is an attribute or a set of attribute in a relation that identify of a touple in a relation. The keys are identify in a table to access or to store data quickly and smoothly They are also used to create relationship B/w different table.

Types of keys:-

1) Primary key
2) Candidate key
3) Super key
4) Foreign key

1) Primary key:- is a candidate key that is selected by the data -base designer to identify identify a touples uniquly in a relation.

Some- important point

(i) A relation can have only one primary key

(ii) Each value in primary key attribute must be unique

Most

(iii) Primary cannot contain null value

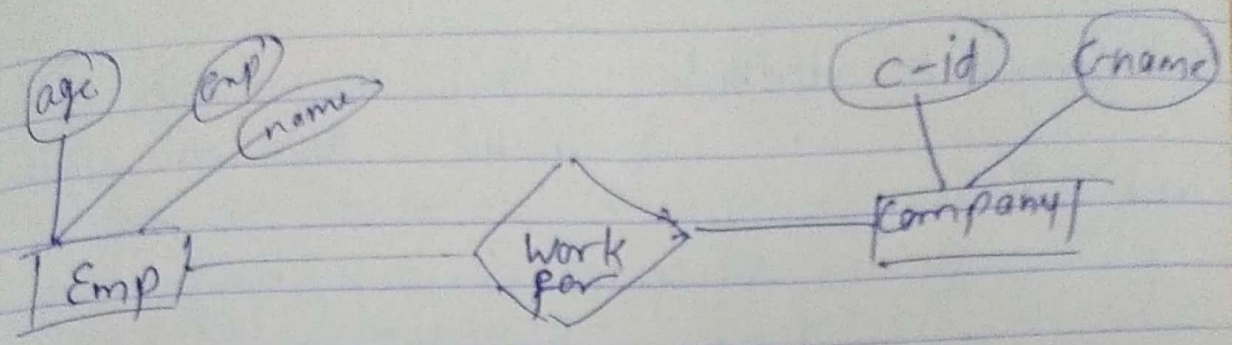
2) Candidate keys:- A candidate keys are defined as the set of fields from which primary keys can be selected.

It is an attribute or set of attribute that can act as a primary key for a table to uniquely identify each record in that table. It consist minimum possible attribute

3) Super key:- a super key is an attribute or is a combination of attribute in a relation that identify a touple uniquely with in the relation it is a super set of candidate key

4) Foreign key:- A foreign key is an attribute or set of attribute in a relation whose value match a primary key in another relation the relation in which foreign key is created is known dependent table or chield table

The relation to which the foreign key refers is known a parent table



emp

emp-id	age	name
P.K		

Work for

Em id	c-id
foreign key	100

Company

C-id	Cnam
100	TCS

relational model

Attribute :- Entities are represented by the properties called attribute all attributes have value

Ex:- a student Entity may have, name, class, age, as attribute ~~typ~~

Types:-

(1) Simple attributes:- simple attribute are atomic value which cannot be devided further.

Ex:- a student phone no. is an atomic value of 10 digits

(ii) Composite attribute :- They are made of more than one simple attribute

Ex :- Students complete name may have. firstname, last name, address.

(iii) Derived attribute :- derived attribute are the attribute that do not exist in the physical database but there values are derived from other attributes present in data base

Ex :- Average salary in Department should not ~~stores~~ saved directly in database but it can be derived.

(iv) Single value :- it contain single value

Example :- Social Security number. (SSN)

(v) Multivalue :- It contain more than one value

Ex :- a person can have more than one value like phone, no, email id etc

Most language

create/alter/drop - DDL => Data Definition language

insert/update/select where (clause) (primary key) - DML => Data manupulation language

manupulation in inserted value

MYSQL:-

(i) How can be connect clint to SQL server.

=> Create data base

Syntax:- create database database-name;
create database collage;

(ii) Syntex for use database

use database-name;
use college;

(iii) Show data

mysql > show databases;

DDL statement:-

Syntax:-

create table table-name
(col1 datatype, col2 datatype ---);

Ex:- create table student (enroll int,
name varchar(10), address varchar(5);
branch varchar(5));

Output

enroll	name	address	branch

Keyboard -> Altered table

(v) Describe table:-

Syntax:-

desc table-name;

mysql> desc student;

field Type Null key default

Q How can be modified the structure of the table.

Ans: Alter table command: you can use the alter table command to add new coulums to a table remove existing coulums and change column name, types and length the syntax are as follows.

Alter table table-name
add [column] column defination
[first / after column];

Ex:-

17. mysql> alter table student add mob int;

~~2) mysql>~~

DDL => Alter / create / drop
DML => insert / delete / update
where (clause) -> Primary key

mysql> alter table student add roll no. int after id;

mysql> alter table student add name char first

V.M

Drop a column :- The drops clause of the alter table statement removes a given column from a table and delete column data. a table must have atleast one column :-

① my sql> after table student drop column address ;
my sql> desc student

*Modify, change, Rename :- (Alter)

Modify clause is used to change the datatype of a existing column.

my sql> alter table name.

modify column_name datatype;
my sql> Alter, table student
modify address varchar(12);

change:-

The change clause is used to modify the attribute name to the a new name in a existing table

Syntax:-

my sql > alter table student
change address location varchar (12);

Rename:-

The rename clause use to a rename the table name.

Syntax:-

my sql > alter table student
rename new employee: name.

Ex:- my sql > alter table student
rename employee:

☆ DML Statement:-

Q How can be insert data into table, delete data and update data into table

So we are using DML statement tune to insert, delete and update the data

[# two most keyword in insert command (into, value)]

(DML command use value)

Insert command:- This command is used to insert data into table

Syntax:-

```
insert into <table name>
values (value1, value2 ---);
```

Ex:-

```
insert into student
value (110, 'Mohan', 'Rewa');
```

Select command for where clause →

Select command is used to extract data from the table and the where clause is used to extract only those record that fullfil a specified conditions

```
Select * from student
where address = JBP;
```

* ⇒ represent complete table

Delete Command:- This command is used to delete data from the table

Syntax:-

```
delete from <table name>
where (<column name> = Some value);
```

Ex:- mysql> delete from student
where id = 10;

delete from student limit 2;

Update Command:- It is used to update data into table.

Syntax:

update table name
set <condition>

Eg:- update student
set id=9 where id=2;

[set is very important in update]

Eg:- mysql> update student set
name = 'Ram' where
id = 20;

change the indexing so,

Eg:- mysql> update student set
id = id+1 rollno, name
20 = 20+1 = 21

~~change the~~
~~Eg:~~

# mysql> update student set
name = 'Ram', limit 3;

* Operators:-

IN
BETWEEN
LIKE
<=
>=
!=
=
distinct

- IN :- IN operator is used to specify a list of value

```
select * from student
where age IN (20, 21, 24)
```

student

id	name	age	city
1	Ram	20	JBL
2	Shyam	18	Delhi
3	mohan	16	Bombay
5	Ayush	21	Bhopal
		24	

* where clause :-

Eg :-

where age = 20 or age = 21 or age = 24

- **BETWEEN:** This operator is used to

ID	Name	Email	Gender	age	city
1	Tom	t@t.com	1	23	London
2	John	j@j.com	1	20	Mumbai
3	Mary	m@m.com	2	21	London
4	John	John@j.com	1	29	JBL
5	Sara	Sara@S.com	2	25	Newyork

Syntax:-

mysql> select * from student where age between 20 and 24

- **LIKE:-** This operator is used to specify pattern.

Q:- Show all the peoples whose city name starts with L

⇒

mysql> select * from student where city like 'L%';

(#%. = place holder)

Q Show all the valid email address from the table.

=>

mysql> Select * from student
where Emal like '%.@%';

[# '%.@.com%';]

Q Show all invalid email address.

• NOT Like operator

mysql> select * from student
where Email NOTLike '%.@%';

Q. Find out those people who lives in london.

• =:-

=> mysql> select * from student
where City = 'London';

Q Find out those people who are not residing in london.

• != operator

=> mysql> select * from student
where city != 'London';

Q Select those people whose age are above 20:

• > operator mysql> select * from student
where age > 20;

distinct - not repeat

09/11/21, Tuesday

- distinct operator :-

→ mysql> select distinct city from student;

→ mysql> select distinct name, city from student; (show all rows)

Output 1

City
London
Newyork
Sydney
mumbai

output 2

Show all name

* Operators and wild cards :-

Operators :-

~~Purpose~~

(i) % → it specifies zero or more character.

(ii) _ (under score.) → it specifies exactly one character

(iii) [ ] → any character with in bracket

(iv) [^] → not any character with in bracket

- if we want one character before @ and one character after @ so we use.

```
select * from Student
where Email like '__@__com';
```

Table :-

Salary	ID	Name	Email	Gender	age	city
10,000	1	John	t@t.com	1	23	London
15,000	2	Tom	T@j.com	1	20	New York
20,000	3	Mary	m@m.com	2	21	Sydney
25,000	4	John	John@j.com	1	29	London
45000	5	Sara	Sara@s.com	2	25	Mumbai

Q find all the people whose name start with m, s, R,

=> select * from student where name like '[MSR]%';

Q find those employs whose name does not start with M, S, R,

⇒ Select * from student
where Name like '[^MSR]%';

Q.1. what is rdbms and explain it command?

Q.2 How to create Database & show Database
Q.3 what is DDL and explain it command? in sql?
Q.4 what is DML and explain it command in sql

Q.5 Explain all operator in sql?
Q.6 Explain order by clause in sql?
Q.7 Aggrigate function in sql?
Q:- Joining multiple condition using AND and OR operators:

Q:- Find all people who living in london or mumbai and their age is 25 or above 25.

⇒ select * from student
where (city = 'London' or
city = 'Mumbai') and
age > 25;

Q.8 Explain group by clause with having clause with sql?
Q.9 Explain all types of join in sql?
Q.10 Explain set operation in sql?

Order By clause:-

This clause is used shorting the rows in database

Q If we find attribute name in ascending order by alphabet so we used this by default table is arranged in asending order.

Ans (i) Select * from student (* All row)
order by name (∵ orderby → arrange)

(ii) Select * from student
order by Name DESC;

Q Find those employs whose name are in descending order and age must be ascending order

⇒ Select * from student
order by Name Desc, age Asc;

Q If you want to show top two record from your table

⇒ Select top 2 Name, age from student;

Q If you want to use half of record from your table so we use

⇒ Select top 50 percent * from student;

Q How do you find the most salary of a employ in any organization

⇒ Select top 1 from student order by Sallary Desc;

ID	Name	Grender	Salary	city
1	Tom	M	4000	London
2	Ram	M	3000	Newyork
3	John	M	3500	london
4	Mary	F	4500	london
5.	John	M	5000	Sydney
6.	Jeiq	F	5500	Newyork
7	Sara	F	4600	Sydney
8	Tara	F	7000	[illegible]
9.	Shyam	M	8000	[illegible]
10	James	M	2800	[illegible]

- Aggregate Anchions

1) Sum 2) Avg 3) Max 4) min 5) count

Q. if I want total sallary paying in any organization

1) select sum (salary) from student;

3) Select max (salary) from student;

4) select Avg (salary) from student;

Q. Find min. salary paying in any organizath

⇒ select min (salary) from student;

* Group by clause :-

It is used to grouped a selected set row into a set of summary rows by the value of one or more column. It is always used with one or more Aggrigate function.

Q:- find total salary of an Employed regarding it city

⇒ select city, sum (salary) as Total salary from student group by city;

Output =

city	Total salary
London	27900
Newyork	15500
Sydny	76000

- Group by multiple column :-

Q:- How much salary is paid for male and female employees.

⇒ Select city, Gender, Sum (salary) as Total salary from student group by city, Gender ;

output

city	Gender	Total Salary
Newyork	f	8500
Sydney	f	4800
London	m	2300
Newyork	m	7000
Sydney	m	2000.

Q:- if you want to city in ascending order so be used order by clause with city attribute.

⇒ Select city, Gender, Sum (salary) as Total salary from student group by city, Gender, order by city ;

- Count :-

Select count * from student ;

output ⇒ 10.

Q. Now if you want total empl. in all different cities with total salary so

=>

```
Select Gender, city , sum ( salary )
as total salary, count (ID)
as total employees from student groups
by Gender, city;
```

[Here we are using multiple aggregate function]

Output:-

Gender	city	total salary	total employee
M	London	2700	5
F	Newyork	3800	4
M	Newyork	4000	3
F	Sydney	4000	2
M	Sydney	3000	1

- **Having clause :-**

Having clause Has to come after group by clause if we use where clause or having clause so where clause come first then group by clause always but having clause always come after grouped by

=>

```
Select Gender, city, sum(salary)
as total salary, count(ID)
as total employee from student
group by Gender, city
Having Gender = 'male';
```

Different types of joins

1) Cross join
2) inner join
3) Outer join — left, Right, full

Joins in sql server

Joins are used to retrive data from two or more relative table in gehral tables are relative to each other using foreign key constent

Employee table

ID	Name	Gender	Salary	Department ID
1	Tom	M	4000	1
2	Pam	F	3000	3
3	John	M	3500	1
4	Sam	M	4500	2
5	Todd	M	2800	2
6	Ben	M	7000	1
7	Sara	F	4800	3
8	Valone	F	5500	1
9	James	M	6500	Null
10	Russell	M	8800	Null

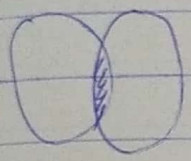
1) Inner join

- It return only the matching row Btw both the table ~~or~~ non matching both are eliminated.

② Department table

Id	Department name	location	Department ID
1	IT	london	
2	CSE	delhi	
3	AI	mumbai	
4	EC	newyork	

• Inner join



Matching - Non-matching
rows - rows
only - eliminated

Select Name, Gender, Salary
Department Name
from employee
join Department
ON employee Department ID
= Departmentid

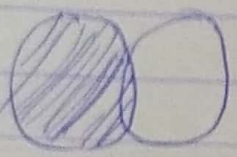
Output:- left two row 9,10

* if you write only join then it will be take inner join.

- Left join / Left outer join

if we want not only the matching row but not matching row also so we use in employee table whether they assign department or not

Matching + non matching
Rows Rows.



employee. Department

⇒ Select name, Gender, Salary,
Department name from employe
left join Department on employee.
Department Id = Department.id ;

- It returns all.

#
Right join / Right outer: It returns all the matching rows + non matching rows from the right table

Select name, Gender, Salary,
Department name
from employee
Right join Department
On employee.Department ID = Department.ID;

- Full outer / Full join :- This join show matching rows from the both table and non matching rows from the right table and the left table

⇒ select .Name, Gender, Sabry, Department Name from employe
full outer join Department on employee Department ID = Department.id

output it show 11 ~~record~~ rows.
11 Null Null Null other department

- Cross Join :- It produce the cartesian product of two table which involve in this join

Ex:- we have 10 rows in employes table and four rows in department table show a cross join Bhw these two table produce ~~40~~ rows

⇒ Select Name, Gender, Salary
Department Name from employee
Cross join Department;

- **Set operation:**

The intersection operators:

If you are need to retrive the common records from both the left and the right table then we used intersect operators

- It firstly introduce in SQL server 2005
- The no. and the order of the columns must be same in both queries.
- The data-type must be same or atleast compatible.

Table A

Id	Name	Gender
1	Mark	m
2	mary	f
3	Steve	m

Table B

Id	alame	Gender
2	mary	f
3	steve	m

Select id, alame, gender from Table A
intersect
Select id, name, Gender from Table B

Q. Diff B/w intersect and inner join?

[distinct keyword]
intersect focuses on distinct and inner join focus on comparation rows.

All operations relational model }

- Union and Union all :-

table indid Customers

id	name	Email
1	Raj	R@R.com
2	Sam	S@S.com

Table UK customers

id	name	Email
1	Ben	B@B.com
2	Sam	S@S.com

Select id, name, Email from tblindia customers
Union
Select id, name, Email from tblUKcustomers;

#
It remove duplicate row and ouput in shorted manner

Output

id	name	Email
1	Ben	B@B.com
1	Raj	R@R.com
2	Sam	S@S.com

Unionall

Select id, name, Email from tblindia customers
Union all
Select id, name, Email from tbl UK customers;

output not remove duplicate & do not ~~remove~~ repeated row.

DBMS
2, 3 unit

→ Relational Algebra

A querie language is a language in which user request information from the database it can be classified procedural and Non-Procedural

The Relational Algebra is a procedural querie language in consist set of relation that take on or two relation to describe another relation both the operant and output are relation and so the output from one operation can turn into the input to Another operat

So operation a relational ~~Alg~~ Algebra are :-

	Operations	Symbol
1) Select		
2) Project	(1) projection	∏
3) Union	(2) Selection	σ
4) Set different	(3) Renaming	ρ
5) Cartesian product	(4) Union	∪
6) Rename	(5) Intersection	∩
	(6) Cartesian product	X
	(7) Join	⋈
	(8) left outer join	⟕
	(9) Right outer join	⟖
	(10) Full outer join	⟗
	(11) Assignment	←

Select show the data
project retrive the rows

(i) Select Operation :- (σ)

It select tuples that satisfy the given predicate from a relation :- Syntax :- $\sigma_{condition}(R)$

Q Obtain information of student who CGPA is 9+

$\Rightarrow$ $\sigma_{CGPA>9}(Student)$

(ii) Project operation :- (Π) It is used to retrive data from columns.

The syntax are as follows :-

$\Pi_{A1, A2 \dots An}(R)$

$\Rightarrow$ $\Pi_{name}[\sigma_{cgpA>9}(Student)]$

it show name of those student whose CGPA > 9

(iii) Cross product :- It genrall followed by Select operation the syntax are as follows

$\Pi_{name}[\sigma_{grade=A}(Student \times Grade)]$

Student	
Name	rollno.
A	1
B	2
C	3

Grade	
Rollno.	Grade
2	A
4	A

(iv) Rename:-

Syntax:- $\rho_{employee-table}(employee)$

Employee

name	emplid
A	123
B	431

- Aggregation:- It is a abstraction for building
- Generalization:- A generalization entities are together represent a more generalised view

The process of generalizing entities where the generalized entities contain the properties at all the entities is called generalisation.

In generalization a no. of entities are accomodative together into generalized entities which based on their similar characteristics. Generalization is a bottom up approach in which two level entities combined to form a higher level entities

ER Model:- Due to the most demanding data base requirement different data models are used to represent the

requirement of new complex application then various semantic model have been proposed. So the ER model supported with additional Semantic concept is called enhanced entity Relationship model.

→ There are most important and useful added concept of EER model are

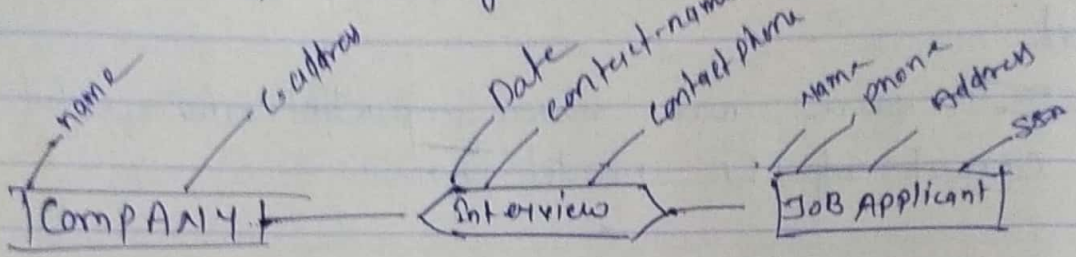
1) Generalization
2) Specialization
3) Aggregation

Specialization:- In specialisation a group of entities is divide into subgroup based their characteristics

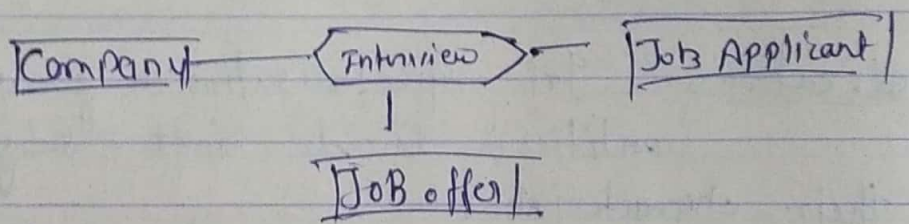
Aggregation:- Aggregation is a abstraction for building composite object for their components object
It is an abstraction through which relation are treated as higher level entities Be consider ER Scheme in following figure which store information about interviews By applicant to various companies The entity company.

Company is an aggregation of component objects: C-name and C-address, where job applicant is an aggregat of SSN, name phone, address.

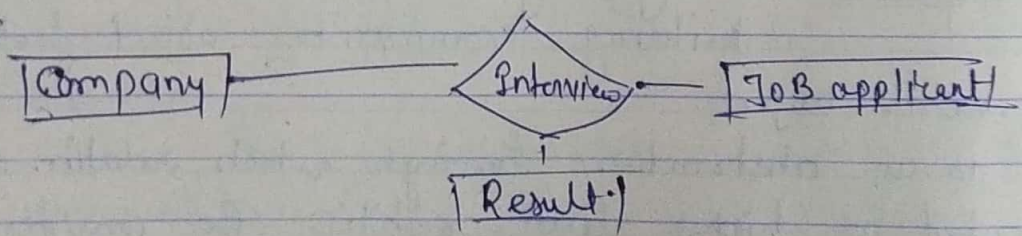
whenever the relationship attribute contact name, contact, phone, Date, represent the name phone number of the person in company who responsible for interview



Now suppose that some interviews result in job offer where as other do not here we like



But it is incorrect because it requires each interview relationship instance to have a job offer.



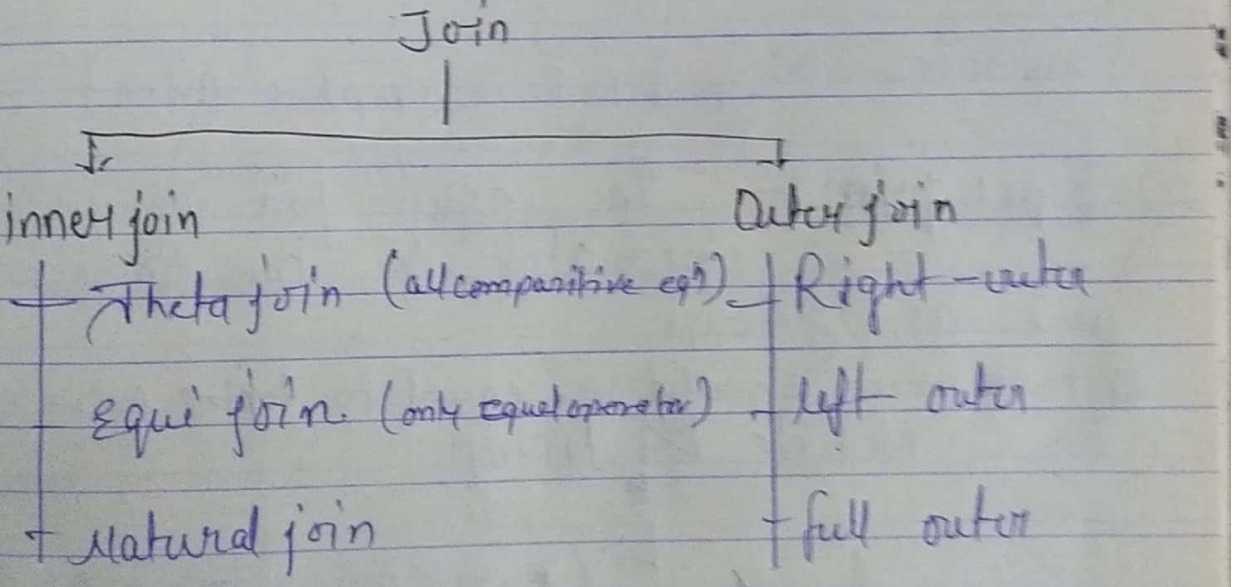
But this is not allow because in E-R model donot allow relationship among relationship so one way to represent the situation to create a higher level aggregation class composed of company, JOB Applicant and Interview and relate this Class to JOB offer, which shown in fig (d) So. described.

Join in Relational Algebra:-

It is used to compile related tuples from two relations into single tuples it is denoted by this symbol '⋈' and this is similar to cross or cartision product

There are different types of join which are as follows:-

1) inner join
2) Outer join



- Theta join :- It is defined as join which has comparitive operator's like (=, <, >, ≤, ≥, !)

Syntax:-

$\bowtie_\theta (R)$

→ comparision conditions {=, <, >, ≤, ≥, !}

Q. I want to purchase both mobile or laptop but mobile price should be less than laptop price.

Mobile

model	Price
vivo	10 K
apple	50 K
Samsung	20 K

Laptop

model	Price
Dell	30 K
acer	20 K
Asus	10 K

$\Rightarrow$ Mobile $\bowtie_{\theta}$ laptop

↳ (mobile Price < laptop . Price)

O.R

mobile $\bowtie$ (mobile Price < laptop . Price)

2) Equi join :- In equi join we used operator to the condition are as follows

mob. price should be equal to laptop price.

$\Rightarrow$ Mobile $\bowtie$ laptop

↳ (mobile price = laptop : Price)

OR

mobile $\bowtie$

(mobile price = laptop Price)

Left outer join :- $\text{L} \bowtie$

Example of left outer join

R

Ename	Did
A	1
B	2
C	3

S

Did	Dname
1	d_1
2	d_2
4	d_4

Ename	Did	Dname
A	1	d_1
B	2	d_2
C	3	Null

Right outer join :- $R \bowtie S$

Matching row for both table & non matching row for right table

R

ename	did
A	1
B	2
C	3

S

Did	Dname
1	d_1
2	d_2
4	d_4

ename	Did	Dname
A	1	d_1
B	2	d_2
Null	4	d_4

Primary key always not similar

R ⋈

3) full outer join:- matching row from both table & non matching row for both table

Ename	Did	Dname
A	1	d1
B	2	d2
C	3	Null
Null	4	d4

Q. for each department find Department name, name of HOD and phone no. of HOD find DName, HOD, P.No.)

Dept (Dep-No - Dname, HOD)
prof (Pid, Depid, Pname, ph-no)
through θ join.

⇒

$\Pi_{Dname,\ Pname,\ Ph.no}\left[Dept \bowtie_{\theta} Prof\right]$

(HOD = Pid) and (Dep-No = DePid)