15/03/22

UNIT-2 Relational Model

Relational model represent- how data is stored in relational databases. RDB stores data in the form of relations (tables).

After designing the conceptual model of database using ER diagram we need to convert the conceptual model into relational model, which can be implemented using any RDBMS languages.

RDBMS lang. (Oracle, Mysql).

What is RDBMS -

- stands for Relational Database Management system.
- is a basis for sql and all modern database systems like MS sql server, IBM DB-2, Oracle, Mysql and MS excess.
- It is based on relational model as introduced by EF codd.

Relational Model Concept :-

(i) Relation - A relation is a table with columns and rows.

(ii) Attribute - An attribute is a name col-

(iii) Domain - A domain is the set of valid values for one or more attributes.

(iv) Tuple - is a row of a relation.

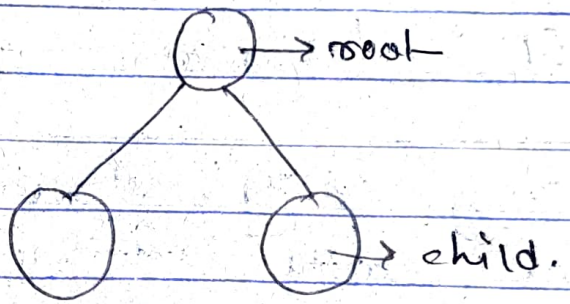
(v) Degree - Total no. of columns or attributes in the relation.

(vi) Cardinality - Total no. of rows present in the table.

Relational Schema -

It represents the name of relation with its attributes (columns).

* Basic of Tree -



primary key

Roll No.	Name	Phone no.
1	Ajay	98 - - -
2	Raj	76 - - -
3	Aman	94 - - -
4	Priya	39 - - -

Degree (column) = 3

Cardinality (Tuple) (row) = 4

properties of

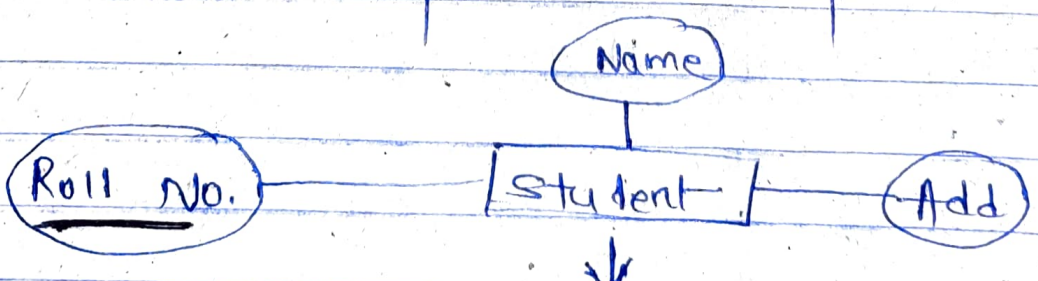
(1) Each relation has unique name.
(2) Each tuple is unique (no duplicate row).
(3) Enteries in any column have the same domain.
(4) Each attribute or column has a unique name.
(5) Attributes value are required to be atomic.
(6) Order of columns and row is irrevelant.

Rules for Conversion :- ER diagram → Relational Tables :-

Rule 1.) "Strong entity set with only simple attributes"
- A strong entity set with only simple attributes will require only one table in relational model.

(i) Attribute of the table will be the attribute of the entity set.

(ii) The primary key of the table will be key attribute of the entity set.



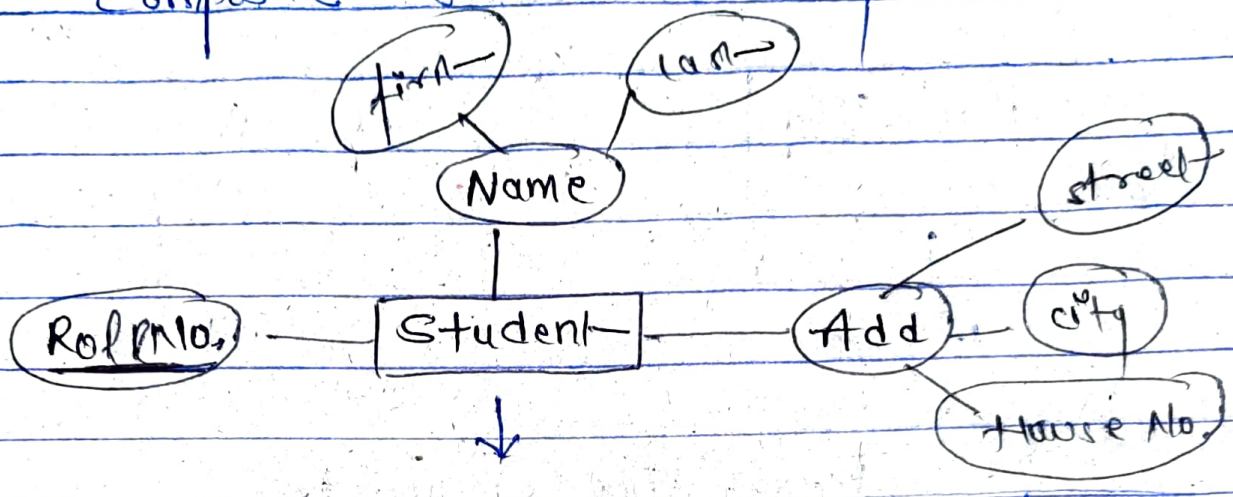
Roll No.	Name	Add

Schema : student (Roll NO, name, Add)

Rule 2: "Strong entity set with composite attribute"

A strong entity set with any no. of composite attribute will require only one table in relational model.

While conversion simple attribute of the composite attribute are taken into account and not the composite attribute itself.



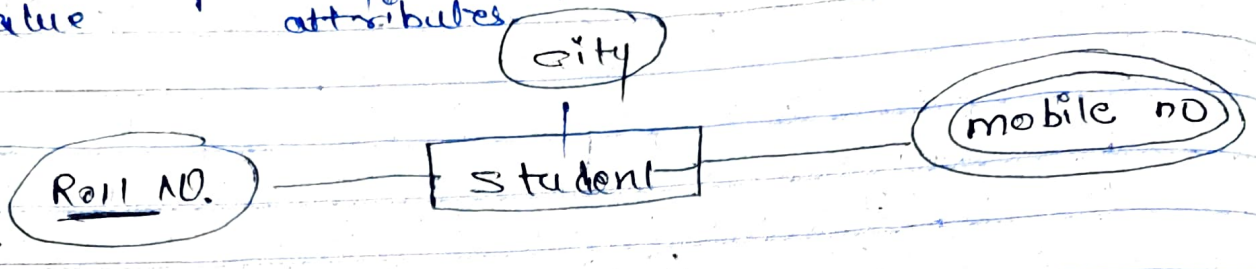
Roll No.	First name	last name	street	city	House no.

Schema: Student (first, last, street, city, house no.)

Rule 3 :- "Strong entity set with multivalue attributes".

A strong entity set with any no. of multi-valued attributes will require two tables in relational model.

One table will contain all the simple attribute with primary key. Other table will contain the primary key and all the multi-value attributes.



Roll NO.	City

(i.)

Roll NO.	Phone no

(ii.)

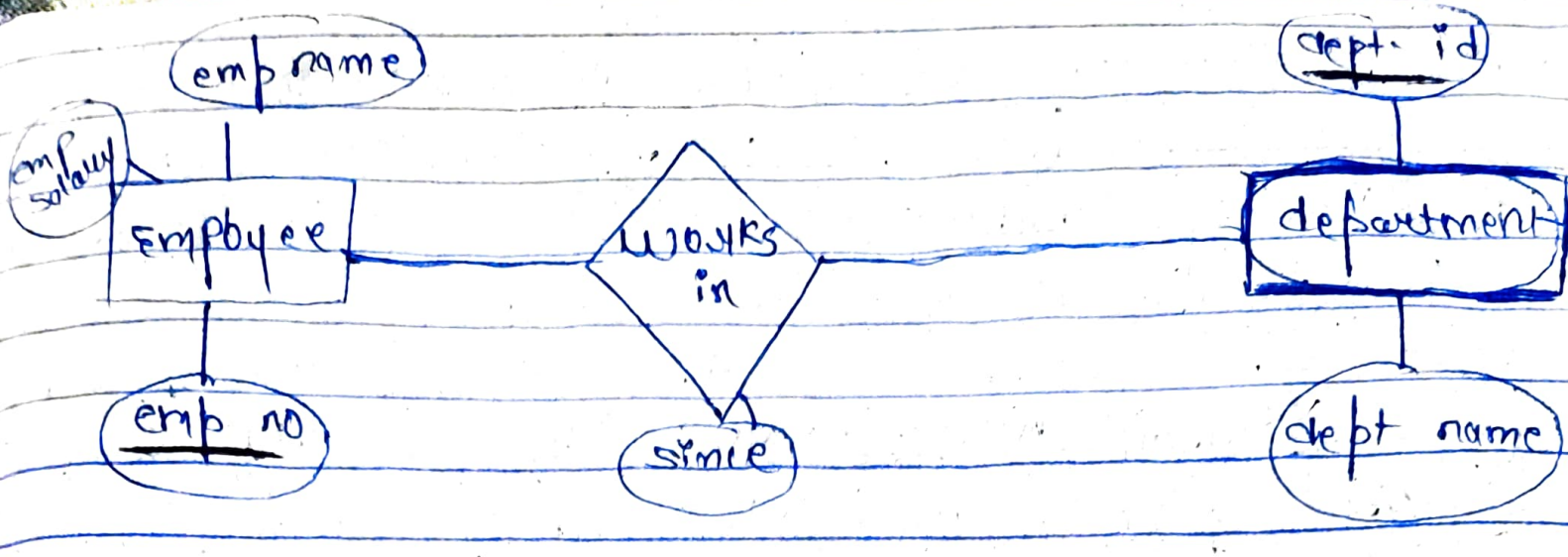
- **Rule 4:—** "Translating relationship set into a table".

(i) A relationship set will require one table in the relational model.

(ii) Attributes of the table are — primary key attribute of the participating entity set.

(iii) It's own descriptive attribute if any.

(iv) Set of non-descriptive attributes will be the primary key.



Employee

Emp no.	emp salary	Emp name

work

Emp no.	since	dept id

dept

dept id	dept name

for given ER diagram 3 tables will be required -

(i) one table for entity set employee,
(ii) one table for entity set dept.
(iii) One table for relationship set work in.

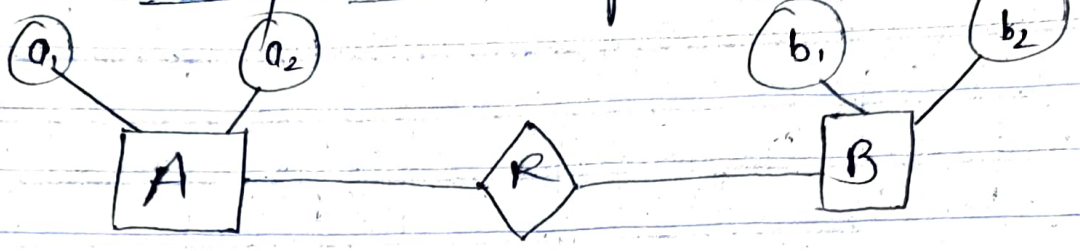
Rule 5 :- "For Binary Relationships with Cardinality ratios".

(i.) Binary Relationship with Cardinality ratio (many - to - many).
(ii.) Binary Relationship with Cardinality ratio (one - to - many).

(ii) Binary relationship with Cardinality ratio. (one to one).

(iv) Binary relationship with Cardinality ratio. (many to one).

(i) Binary Relationship with Cardinality (many to many)

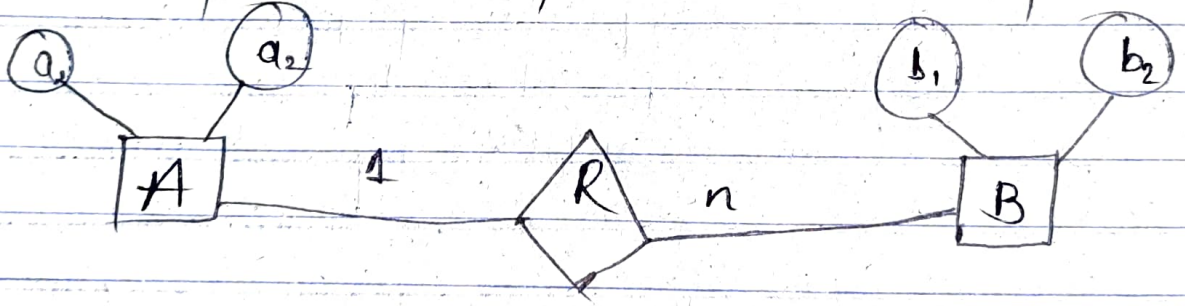


Schema- A (a_1, a_2)

B (b_1, b_2)

R (a_1, b_1)

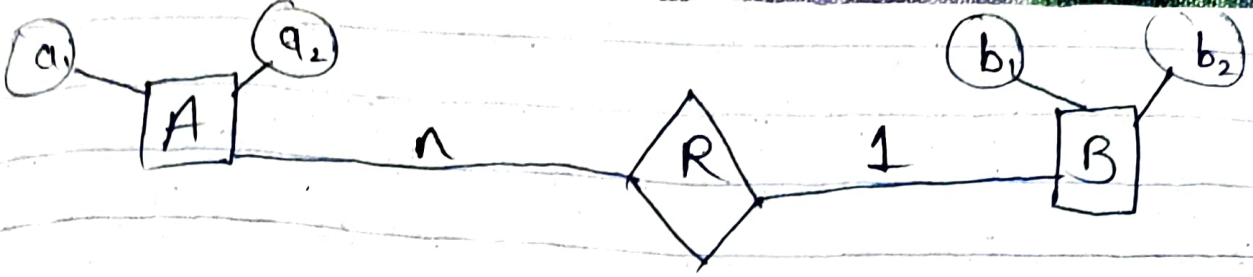
(ii) Binary Relationship with Cardinality Ratio (one to many)



Schema- A (a_1, a_2).

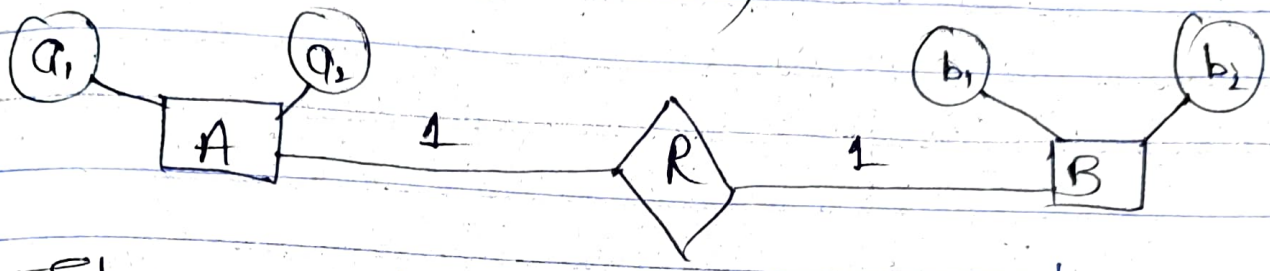
BR (a, b_1, b_2).

(iii) Binary Relationship with Cardinality Ratio (many-to-one)-



Schema - AR (a_1, a_2, b_1)
B (b_1, b_2)

(iv) Binary Relationship with Cardinality (one to one) :-



Schema - AR (a_1, a_2, b_1) or BR (b_1, b_2, a_1)
B (b_1, b_2) or A (a_1, a_2)

KEYS :-

Emp id	Name	Adhar no.	Email	dept no.
1	Ravi	—	—	1
2	Neha	—	—	2
3	Aman	—	—	2
4	Sachin	—	—	3

A Key is an attribute or set of attribute that uniquely identifies any record (tuple) from the table.

Key is used to uniquely identify and record or group of data from the table. It is also used to establish and identify relationships between table.

Types of Keys:-

(i) Super
(ii) Candidate
(iii) Primary
(iv) Attribute
(v) Foreign
(vi) Composite.

Super Key :- It is a combination of all possible attributes that can uniquely identify the rows or tuple in the given relation.

- Super Key is a super set of candidate Key.
- A Table can have many super Keys.
- A super Key may have additional attribute that are not needed for unique identity.

- Super Key are-

(i) Emp id
(ii) Adhaar no.
(iii) Email id
(iv) Emp id and Adhar no.
(v) emp id and email id.
(vi) Emp id, adhaar no. and email id.
(vii) Emp id, adhaar no, email id, name and dept-no.

CANDIDATE KEY :-

A candidate Key is an attribute or a set of attribute which can uniquely identify a tuple.

Candidate Key is a minimal superKey, or a superKey with no duplicate attributes. It is called as minimal super key becoz we select a candidate Key from a set of super Key such that the selected candidate Key is the minimum attribute require to uniquely identify the table.

Candidate Key are defined as distinct set of attributes from which primary Key can be selected.

Candidate Key are not allowed to have null values.

Emp id	Name	E-mail	Dept-id	adhar

Here, Email, Emp id, Adhaar are unique.

∴ they are candidate Key.

(In mse ek choose krke primary Key banayege)

PRIMARY KEY:—

It is one of the candidate key selected by database designer to uniquely identify the tuple in the relation.

The value of primary key can never be null.

The value of primary key must always be unique.

The value of primary key can never be changed i.e no updation is possible.

A relation allowed to have only one primary key.

ALTERNATE KEY:—

- Out of all candidate key only one get selected as primary key remaining are known as Alternate key.
- In Employee table, Empid = primary key rest of the attributes aadhar no., email are consider as alternate key.

COMPOSITE KEY:—

- ~~Set of attributes~~ The key that has one or more than one attribute is known as Composite or Compound key.

Foreign Key :—

- It is used to link two tables together.
- It is an attribute or set of attributes in one table that refers to the primary key of another table or in same table.
- The purpose of foreign key is to maintain different-ial integrity of data.
- It can take only those values which are present in the primary key of the referenced relation.
- Foreign key may have a name other than that of primary key.
- F.K can take null values.
- There is no restriction on foreign key to be unique.
- Referenced relation also called base table/ master/ primary table and Referencing table also called foreign table.

Employee table (Referencing model)

Emp id	Name	Adhaar card	e-mail	Dep id
1	Aman	—	—	2
2	priya	—	—	3
3	prince	—	—	

Foreign key

Primary Key →

Dep id	Dep name
1	Sales
2	Male S
	mmm

★ INTEGRITY CONSTRAINT -

- They are used to ensure accuracy and consistency of the data in a relational database. These are set of rules that the database is not permitted to violate.

- Constraints may apply to each attribute or they may apply to relationship bet^n table.

- Integrity Constraints ensure that changes (update, deletion, insertion) made to database by authorized user do not result in the loss of data consistency.

eg- Blood group must be A or B or AB or O only.

- There are 4 types of I.C -
  (i) Domain Constraint.
  (ii) Key "
  (iii) Entity Integrity "
  (iv) Referential Integrity "

(i) DOMAIN CONSTRAINT -

= It defines the domain or the valid set of values for an attribute.

- 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.

(ii) Entity Integrity Constraint:-
The EIC states that primary key value can't be null. This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value then we can't identify those rows.

A table can contain a null value other than primary field.

(iii.) KEY CONSTRAINT:-
An entity set can have multiple key so candidate key (minimal key) but out of which one key will be the primary key.

Key constraints specifies that in any relation all the values of primary key must be unique.

The value of primary key not be null.

(iv.) DIFFERENTIAL INTEGRITY CONSTRAINT:-
DIC specifies between two tables.

Differential IC is enforced when a foreign key

References the primary key of a table.

- In DLC, if a foreign key in table 1 refers to primary key of table 2 - then either every value of the foreign key in table 1 must be present in primary key value of table 2. or must be null.

Rules -

(i) You can't delete a record from a primary table if matching record exist in related table.

(ii) You can't change a primary key value in the primary table if that record has related records.

(iii) You can't insert a value in the foreign key field if the related table that doesn't exist in the primary key of the primary table.

However, you can enter a null value in the foreign key specifying that the records are unrelated.

RELATIONAL QUERY LANGUAGE :-

It is a lang in which User request inf. from the database. eg - Sql

There are two types of query lang :-

(i) Procedural lang.
(ii) Non- Procedural lang.

(i) PROCEDURAL - In this lang. User instruct the system to perform a series of operations to produce the desired result.

User tells what data to be retrived from database and how to retrive it.

(ii) NON- PROCEDURAL (Declarative) lang -

User instruct the system to obtain the desired result without telling the step by step process.

User tells what data to be retrived from database but doesn't tell how to retrive it.

TWO PURE QUERY LAN :-

(i) Relational Algebra < Tuple / Domain
(ii) Relational Calculus

Relational Algebra -

- It is a procedural query lang. It is more operational, very useful for representing execution plan.
- It takes a relation as an input and generates a relation as an output.
- It is a lang. for expressing relational database query.
- It uses operators for performing query. Operators can be unary or binary.

- Types of operator in Relation Algebra -

(i) Basic or fundamental operators.
(ii) Additional or derived operators.

Relational algebra operations work on one or more relations to define another relation without changing original relation.

(i) Basic operators -

(i) Selection
(ii) Projection.
(iii) Union
(iv) Set difference.
(v) Cartesian product
(vi) Rename.

(ii) Additional operators:-

(a) Natural

(b) left, right, full outer join.

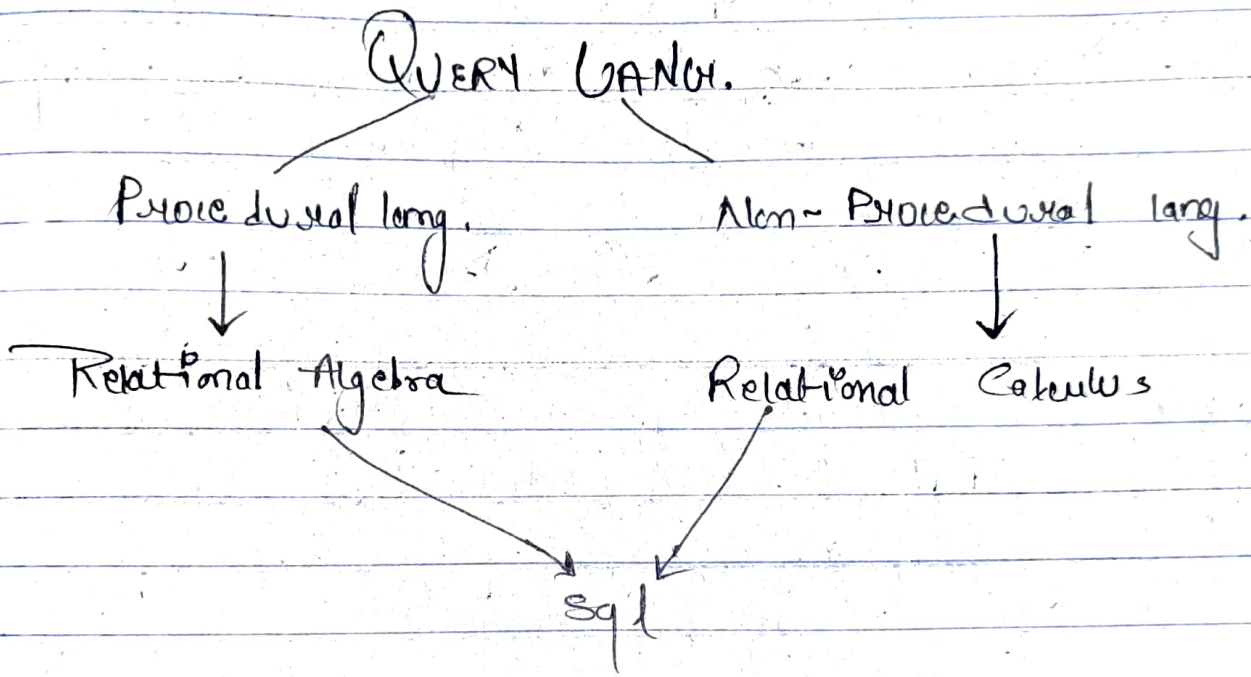
(c) Set Intersection.

(d) Division

(e) Assignment.

NOTE:- [Select, project, rename are unary operators]

[Union, diff, cartesian product are Binary op]



SELECTION OPERATOR :-

- It is unary operator that perform selection operation. It select tuple or row that satisfies given condition from a relation. It is denoted by sigma (σ).

Notation - $\sigma_p(r)$

r = relation / table name

p = condition / predication.

where p = logical connectives

i.e. $\wedge$ And

$\vee$ OR

$!$ NOT

and relational operator like : $=, \neq, <, >, \leq, \geq$.

The where clause of sql command corresponds to relation select operator.

* In sql = select * from D.

Query :- select a type from student table whose age is greater then 17.

$\sigma_{age>17}(student)$

(e.) select student whose roll no. is 2

$\sigma_{rollno.=2}(student)$

e.) select student whose name is D.

$\sigma_{name="D"}(student)$

(11.) select student- whose age is greater than 17 who lives in delhi

$\sigma_{age>17 \wedge address = "Delhi"}$ (student)

PROJECTION OPERATOR:-

- It is unary operator in relational algebra.
- It project or display the particular columns or attributes. It deletes column that are not in the projection list.
- It is denoted by π.

or $\pi_{A_1, A_2 \ldots A_n}(r)$

or

$\pi_{\text{attribute list}}$ (Relation name)

- Duplicate rows are automatically eliminated from result. The SQL select command corresponds to relational project operator.

Select $A_1, A_2 \ldots A_n$ from tablename.

eg (1.) Display the columns Roll No., and Name from relation student.

$\pi_{\text{Roll No, Name}}$ (student)

(2.) Display age of student from student table.

$\pi_{age}(student)$

(3.) Display the roll no., name and student whose age is greater than 17.

$\pi_{name, rollno.}(\sigma_{age>17}(student))$

SET OPERATORS :-

Union, intersection and difference are binary operators as they take two input relations.

To use set operator on two relations the two relations must be compatible. Two relations are compatible —

(i) if both the relations must have same no. of attributes or columns.

(ii) Corresponding attribute or column have the same domain or type.

(iii) Duplicate tuples are eliminated automatically.

(1) UNION - Suppose R and S are two relations then the union operation selects all the tuples that are either in relation R or S or in both relation R and S.

for union operation to be valid the following condition must hold -

1) Two relation R and S both have same no. of attribute or column.

2) corresponding attribute or column have the same domain or type.

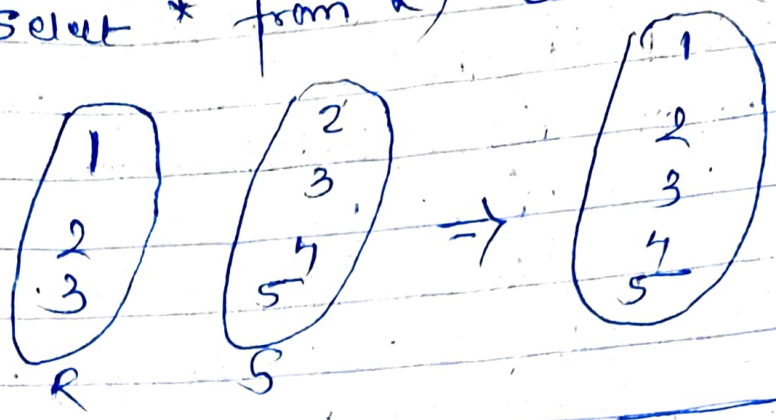
3) The attribute of R and S must occur in the same order.

Symbol = $\cup$

Notation = $R \cup S$

In SQL- (select * from R) UNION (select * from S)

-



-

R

Roll. No.	Name
1	A
2	B
3	C
4	D

S

Emp. no.	Name
2	B
8	G
9	H

RUS

Roll. No.	Name
1	A
2	B
3	C
4	D
8	G
9	H

find the name of author who have either written a book or an article?

Π_{author} (Book) $\cup$ Π_{author} (article)

(11.) INTERSECTION -

Suppose R and S are two relations, the set intersection operation selects all the tuples that are in both relations R and S.

- for a set Intersection to be valid the following condition holds -

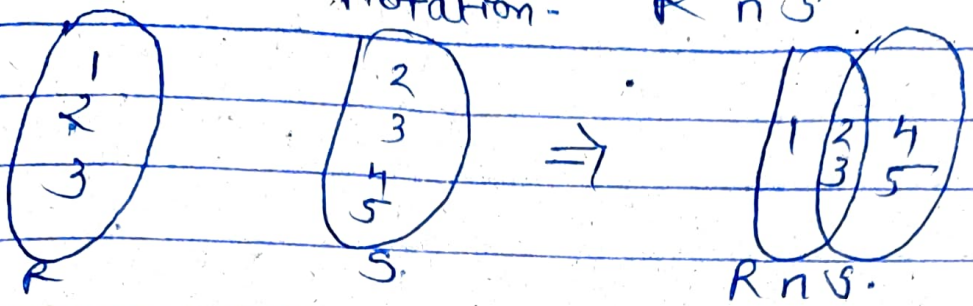
(1.) Two relation R and S must have same no. of attribute or column.

(ii.) Corresponding attribute or column have the same domain or type.

(iii) The attributes of R and S must be in the same order.

Symbol – ∩

Notation – R ∩ S



Q find the name of the Author who have written a Book and an Article both?

$$\pi_{author}(Books) \cap \pi_{author}(Article)$$

SET DIFFERENCE:-

- Suppose R and S are two relations, the set-difference operation select all the tuples that are present in relation R but not in second relation S.

- The foll. condition must hold:-

(i) Two relation R and S, both have same no. of column and attribute.

(ii) Corresponding attribute or column have same domain.

Symbol = −
Notation = R − S.

Stud:

Roll No.	Name
1	A
2	B
3	C
4	D

Emp

Empid	Name
2	B
8	G
9	A

⇒

Roll No.	Name
1	A
3	C
4	D

Stud − Employee

CROSS PRODUCT (Cartesian)

- It is the fundamental operation in relational algebra.
It combines info of two different relation into one table.
It is also called cartesian / cross product.

Symbol = X
Notation = R X S

In Sql = Select * from R, S;

R1

A	B
α	1
β	2

R2

C	D	E
α	10	α
β	10	α
β	20	b
γ	10	b

$R_1 \times R_2$

A	B	C	D	E
a	1	α	10	a
α	1	β	10	a
a	1	β	20	b
β	2	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
		γ	10	b

conditions:-

(i) if Relation R_1 and R_2 have a and b attribute then resulting relation will have (a+b) attribute from both the input relation.

(ii) if relation on R_1 and R_2 have a_1 and a_2 tuples then resulting relation will have $n_1 \times n_2$ pair or tuples.

Attribute	a	b	a+b
tuples	n_1	n_2	$n_1 \times n_2$

(iii) if both input relation have same attribute having same name then change the name of the attribute with the name of relation name.

R_1

A	B
a	1
b	2

R_2

B	D	E

A	R_1.B	R_2.B	D	E

DIVISIONAL OPERATOR :-

↳ can be expressed in terms of cross product, set difference and projection.

↳ is suited to queries that include Keywords, OR, All, EVERY.

eg ① find the person that has an account in all the banks of a particular city.

② find students who have registered for every course.

JOINS OPERATORS

Join is a combination of cartesian product followed by selection process.

Join = Cartesian product + Selection.

Cartesian product of two relations a x b gives us all the possible tuples that are paired together.

But it might not be feasible in certain cases to take cartesian product where we encounter huge relations with thousands of tuple having a large no. of attribute.

Join operation pair two tuples from diff. Relation if and only if a join condition is satisfied.

Symbol = ⋈

eg - $A \bowtie_c B = \sigma_c (A \times B)$

Types of Join operation:-

- Inner join
  - Theta join (θ)
  - equi join
  - Natural join
- Outer join
  - left outer join
  - Right outer join
  - Full outer join.

- **Inner Join** -

It contains only those tuples that satisfy the matching condition.

- **Outer Join**-

(i) Extension of join.

(ii) It contains matching tuple that satisfy the matching condition along with some of all tuples that do not satisfy the matching condition.

Join	Cartesian
(i) Combination of tuple that satisfy the matching condition.	All possible combination of tuples from the relation.
(ii) Fewer tuples than cross product might be able to compute efficiency.	Huge no. of tuples and costly to manage.