

Key

Rollno	Name	Age
1	A	20
1	B	20
3	A	18

* Key is attribute or set of attribute.

* uniquely identify a tuple.

* combination of attribute i.e. composite key.

* If two key of attribute is same then we have to make key combination to make unique identify.

Rollno. + Name

1 A

1 B

3 A

* Data base administrator choose the key

DBA is a person or group of person it works on create a attribute.

* In one table key may or may not be more than one.

* Table can be without making key.

* For good database design we should make key.

* Key is a attribute or set of attribute that is used to uniquely identify a tuple.

Ex- Roll no. is a unique key attribute.

Ans * Types of Keys

i) Super Key

ii) Candidate Key

iii) primary Key

iv) Alternative Key

v) composite Key

vi) unique Key

vii) Foreign Key

Website:- Diplomawallah.in

Key

Key are one of the basic requirements of a relational database model. It is widely used to identify the tuples (rows) uniquely in the table.

We also use keys to set up relations amongst various columns and tables of a relational database.

- * Key is attribute or set of attribute.
- * It is the combination of attribute i.e. tuple. Composite Key.
- * The database Administrator choose the key.
- * In one table key may or may not more than one.
- * For good database designer we should make key.

Rollno	Name	Age
1	A	20
2	B	20
3	A	18

• Roll no is a unique key attribute.

Types of Keys:-

- | | |
|---------------------|------------------|
| i) Super Key | v) Composite Key |
| ii) Candidate Key | vi) Unique Key |
| iii) primary Key | vii) Foreign Key |
| iv) Alternative Key | |

Super Key

Student

Age	Rollno	Name	Address
20	1	A	MNO
18	2	B	XYZ
21	3	A	DEF
20	4	C	PQR
19	5	B	NMP

- * The Set of attributes that can uniquely identify a tuple is known as Super Key.
 - * A Super key is a group of single or multiple keys that identifies rows in a table.
 - * It support null values.
 - * Super key is more than any another key.
 - * Adding zero or more attributes to the candidate key generates the super key.
 - * A Candidate key is a super key but the super key is not a candidate key.
 - * Super key value may also be Null.
- Super Key $\rightarrow$ ① Roll no
 ② Address

Superset — Roll no + Name
 Address + Name
 Roll no + Address
 Roll no + Name + Address

→ Formed Super Key

ii) Candidate Key: —

- * It is a attribute or set of attribute.
- * The minimal set of attribute that can uniquely identify a tuple is known as a candidate key.
- * It is a minimal Super Key.
- * It is a super key with no repeated data is called a candidate key.
- * The minimal set of attributes that can uniquely identify a record.
- * It must contain unique values.
- * Every table must have atleast a single candidate key.

(i) Roll no (ii) Address

Roll no + name

↳ Candidate Key X

Super Key ✓

This is not candidate key because Roll no has unique identify and don't want to make new key from collaboration.

Primary Key = unique + not null.

iii) primary Key

- * There can be more than one candidate key in relation out of which one can be chosen as the primary key.
- * It is a unique key.
- * It can identify only one tuple at a time.
- * It has no duplicate values, it has unique values.
- * It cannot be null.

Ex- (i) Roll no } candidate key
(ii) Address }

primary key $\rightarrow$ Roll no.

The left key "Address" is Alternative key.

iv) Alternative Key

- * The candidate key other than the primary key is called an alternative key.
- * All the keys which are not primary keys are called alternative keys.
- * It contains two or more fields to identify two or more records.
- * These values are repeated.

The "Address" is alternative key.

v) Composite Key

* Sometimes a table might not have a single column/attributes that uniquely identifies all the records of a table.

* To uniquely identify rows of a table, a combination of two or more columns/attributes can be used.

* It acts as a primary key if there is no primary key in a table.

* Two or more attributes are used together to make a composite key.

EX - Roll no + Name

vi) unique Key

* It may contain Null value.

* It may be or may be not more than one.

* we can make unique key from Primary key.

* Contain multiple attributes.

Vii) Foreign Key

- * We need minimum two relation for foreign key.
- * There should be one common attribute from taking relation.
- * The Attribute datatype, size should same.
- * Common attribute should have primary key.

primary key

Student			Course		
Rollno	Name	Age	CID	C Name	Roll
26	ABC	18	200	DBMS	27
27	X42	20	300	CHNA	26
30	MNO	18	400	CN	26
			500	PN	28

Referenced relation
(parent) base

Referencing relation (child)

- * The ^{base} relation/table of base always contain primary key.
- * The primary key of base table (referenced relation) will be the foreign key of referencing relation.
- * The data contain by primary key of base relation will be used in foreign key, The data can be repeated but another data cannot.



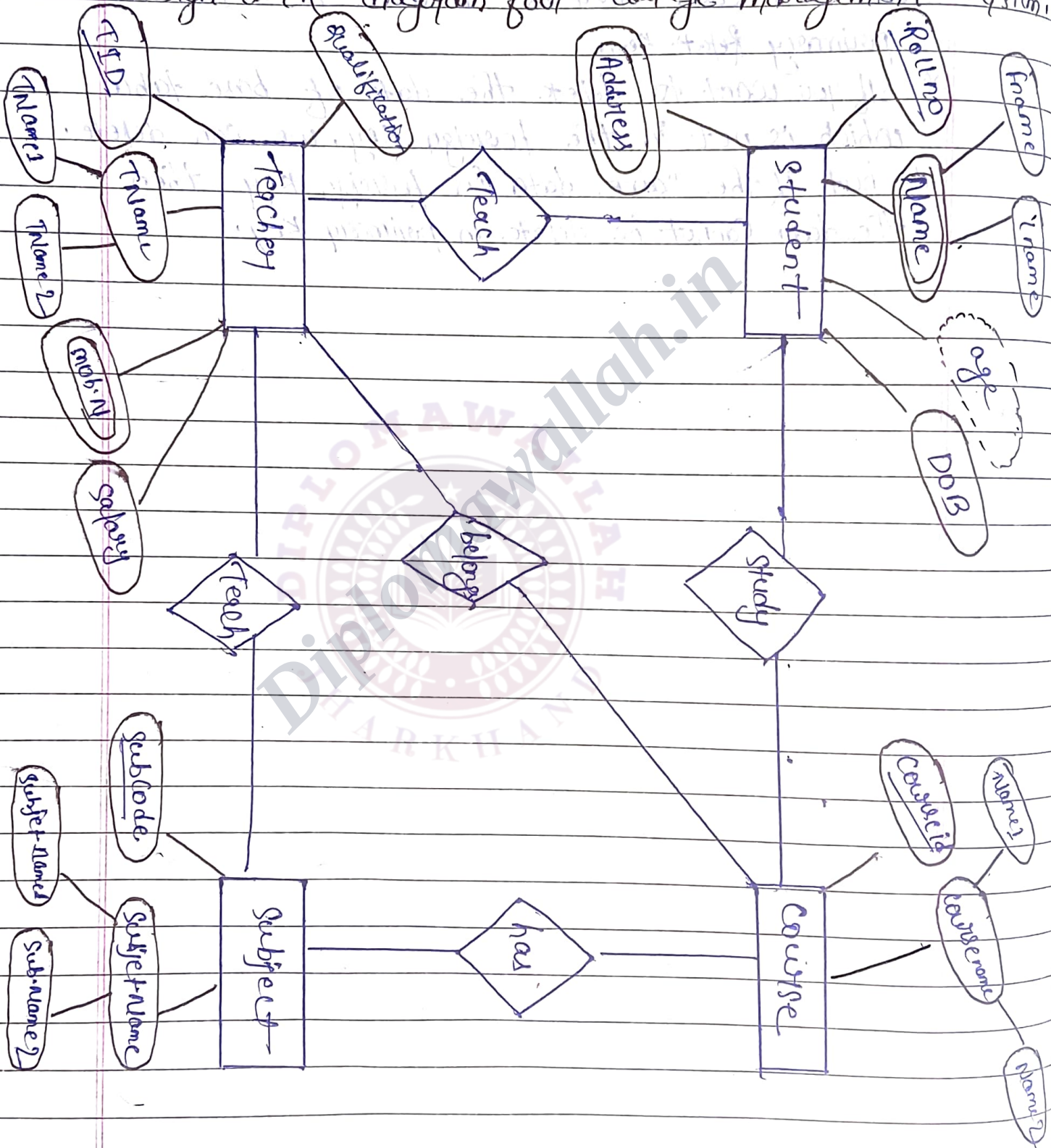
* We can't use 'Roll no' because it is not in primary Relat Key.

* If you want to delete the data of base table which is not in the foreign key, we can delete.

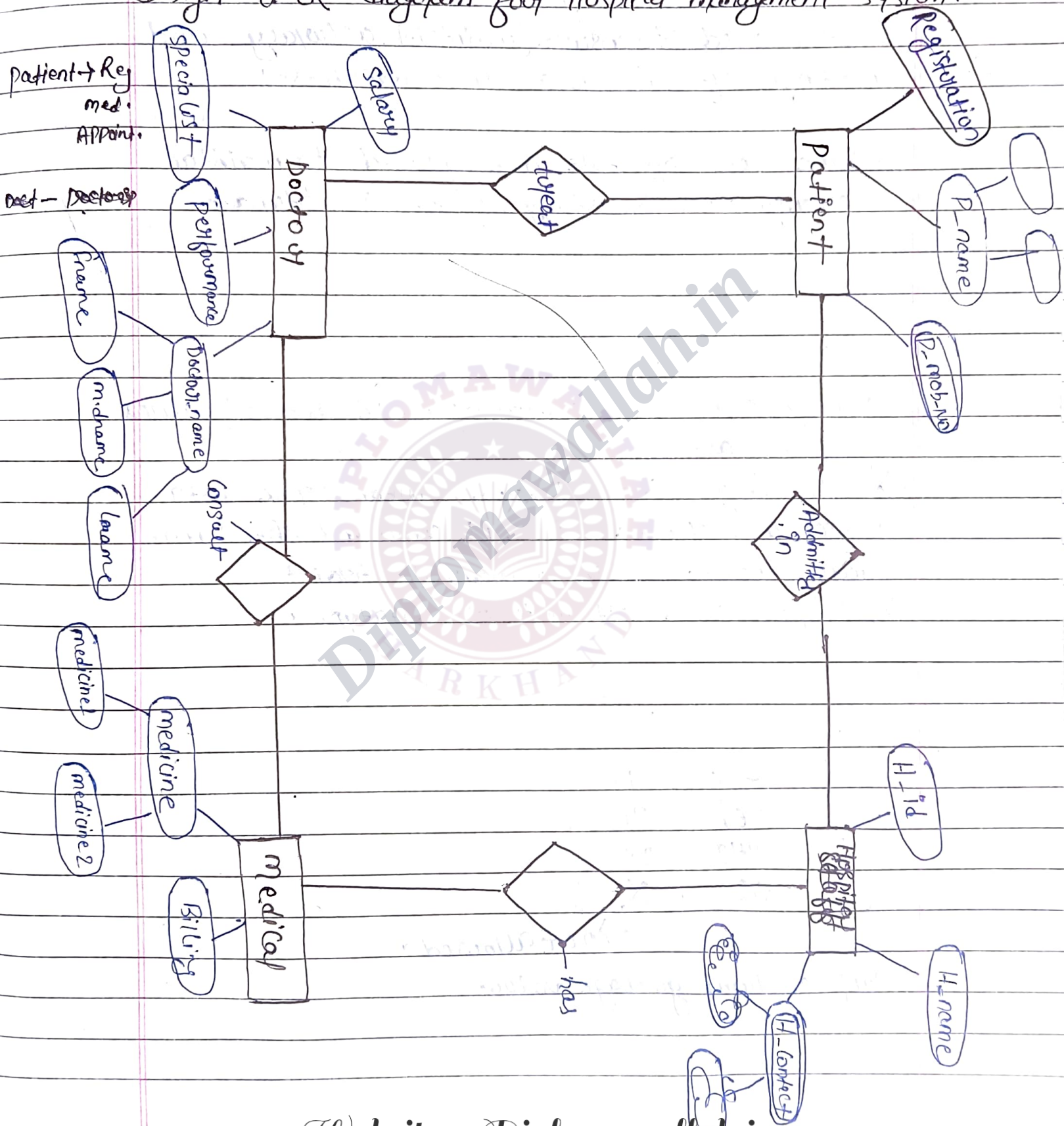
* but if the same data in foreign key then the data cannot be delete in primary key.

ER Diagram

Design a ER diagram for College management System.



Design a ER diagram for Hospital management system.



Integrity Constraints

→ It is used to ensure/maintain accuracy and consistency of the data in a database.

* Some sets of rule are formed to maintain accuracy and consistency of data in database.

Types:-

① Domain Constraints:-

→ Defined as the definition of valid set of values for an attribute.

→ It defines the datatype means the data which are stored in DDL command the same data should be in Domain Constraint.

→ It apply on a attribute of datatype.

int	char	int
Rollno	Name	Age
1	ABC	12
2	DEF	20
3	UHI	22
4	XY2	X

↘ not allowed

Null contains garbage value

② Entity Integrity Constraints

→ It is apply on only primary key attribute.

→ If primary key contains null value then we cannot identify those rows.

→ Table can contain null value in it except primary key field.

Roll-no	Name
1	A
2	B
□	C

→ null value. ✗

It is not allowed because it is containing primary key as null value.

③ Key Constraints:-

→ Keys are the entity set that are used to identify an entity within its entity set uniquely.

→ An entity set can contain multiple keys,

but out of them one key will be primary key.

→ primary key is always unique, it does not contain any null value in table.

Rollno	Name	age
1	A	20
2	B	30
□	D	18

not allowed ✗

(iv) Referential Constraint:-

- It apply on a table of attribute.
- It is for checking the data.
- you cannot delete a record (data) from a primary table (base table) if matching data exist in a related table.
(we can delete from child table).
- you cannot update, ~~from~~ or change a primary key value in the primary table if that records has related records.
- you cannot insert a value in the foreign key of the related table that does not exist in the primary key of the primary table.
- we can enter a null value in the foreign key, specifying that the records are unrelated.

Entity - Relationship

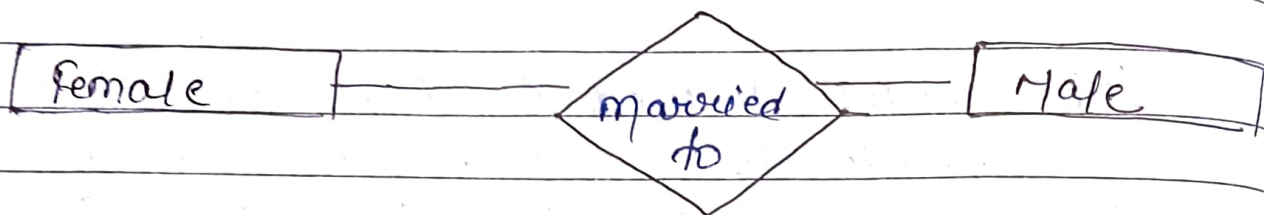
- ER models stands for an Entity - Relationship model.
- This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database.
- Develops a very simple and easy design view of data.
- In ER model we show the data relationship through diagram.

* Components of ER Diagram ER model

↓ Entity	↓ Attribute	↓ Relation
↳ weak entity ↳ strong entity	↳ Key Key attribute ↳ Composite attribute ↳ Single-valued attribute ↳ Multivalued attribute ↳ Derived attribute	↳ one to one ↳ one to many ↳ many to one ↳ many to many

Entity:-

An entity is object, class, person or place.
In ER diagram entity can be represented as rectangle.



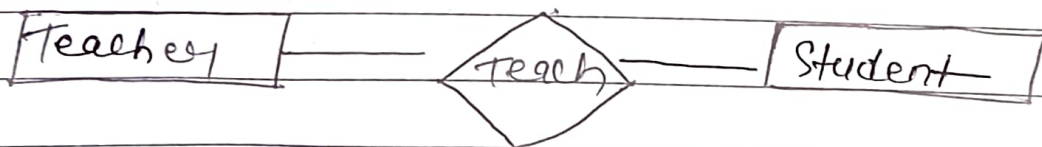
a. weak Entity:-

An entity depends on another entity called a weak entity.
→ It doesn't contain any key attribute.
→ Represented by a double rectangle.



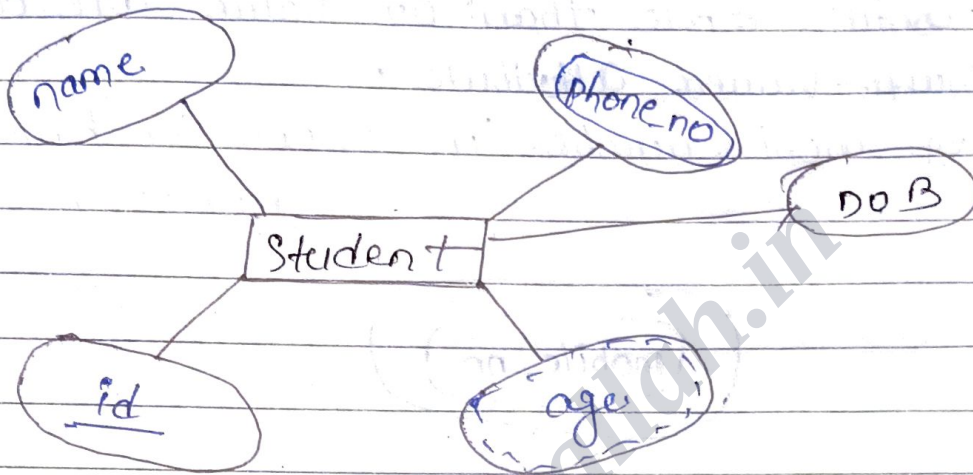
b. Strong Entity:-

An entity depends upon their self.
→ It contains any key attribute or primary key.
→ Represented by a single rectangle.



2. Attribute :-

- It describes the property of an entity.
- Represented by the shape of ellipse.

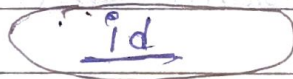


• ~~Simple attribute~~

• Key attribute :-

It is an attribute or set of attributes that uniquely identifies a row in a table.

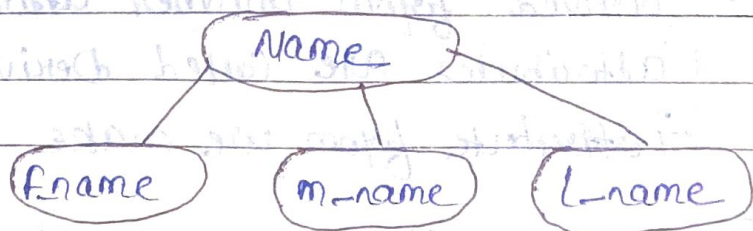
• ~~Comp~~



• Composite ~~key~~ attribute :-

Those attributes which can have more than one attributes is called composite attribute.

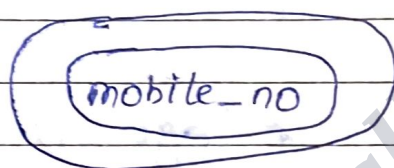
Represented by :-



* Multi-Valued attribute:—

Those attributes which can have more than one entry or which contain more than one value are called Multi-Valued attributes.

eg. mobile number (one person can have many numbers)

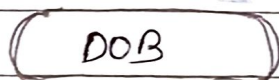


* Single-valued attribute:—

Those attributes which can have exactly one value are known as Single Valued attributes.

→ They contain only single value.

eg - DOB - Birth date.



* Derived attribute:—

The value of attribute derived from another attribute, those attributes are called Derived attribute.

→ attribute from we make a new attribute.

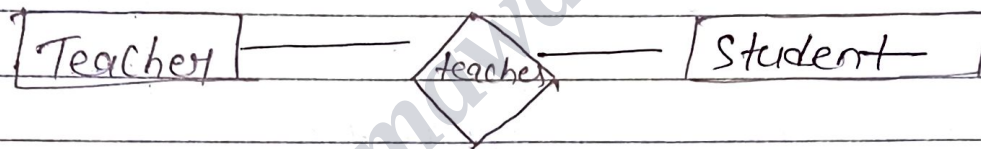
Eg - age () is derived attribute -

age derived from DOB

3. Relationship

A Relationship is used to describe the relation b/w entities.

→ Diamond or rhombus is used to represent the relationship.

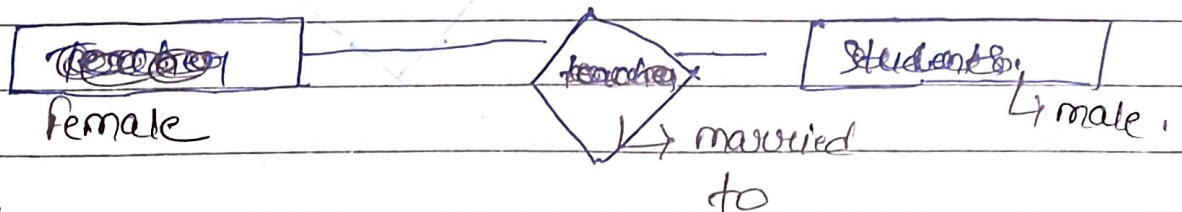


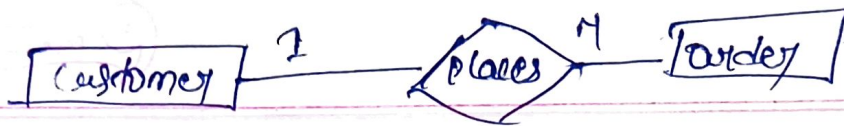
Types of relationship are as follows:

1. One-to One Relationship (1-1) :-

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

e.g - A female can marry to one male, and a male can marry to one female.

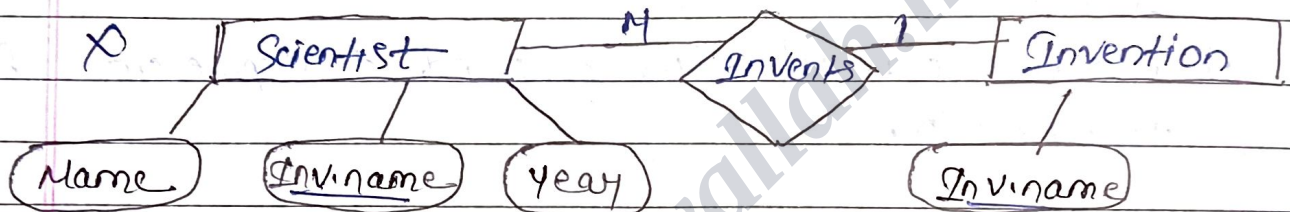




② One to many relationship:

when one instance of the entity on the left, and more than one instance of an entity on the right associates with relationship then it's known as a one-to-many relationship.

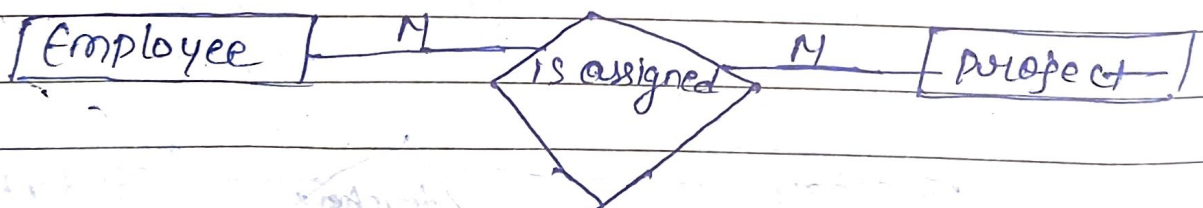
e.g - Scientist can invent many invention.



③ many to many relationship: —

when more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it's known as many-to-many relationship.

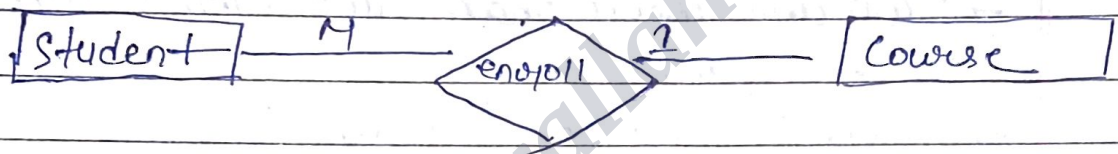
e.g - Employee can assign by many projects and project can have many employees.



④ many to one relationship:-

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

eg - Student enrolls for only one course, but a course can have many students.

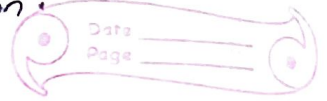


* Symbols used in ER model:-

* Rectangle		Entities in ER model
* Ellipse		Attributes in ER model
* Diamond		Relationship
* Line		Connection b/w entity and relationship.
* double rect. weak entity		Double rectangle weak entity
* Double ellipse		Multi-valued attributes
* underlined att.		Primary key
* dotted ellipse		Derived attributes.

Row level redundancy - whole field of common type.

Column level redundancy - Some field are common.



Normalisation

→ Normalization is a technique to remove our reduce redundancy from a table.

→ There are two types of redundancy

(1) Row level redundancy

(ii) Column level redundancy

(1) Row Level Redundancy

→ Two and more than two rows are exactly same.

→ Relation Name - "Student"

Sid	Sname	Age
1	Ram	20
2	Shyam	25
1	Ram	20

→ Remove the Row level duplicacy

- we use the concept of primary key
- Primary key (unique + Not Null)

(2) Column Level Redundancy

→ Two and more than two column are exactly same.

→ Relation Name - "Student"

Sid	Sname	Cid	Cname	Fid	Fname	Salary
1	Ram	C1	DBMS	F1	John	30000
2	Shyam	C2	JAVA	F2	Bob	40000
3	Ravi	C1	DBMS	F1	John	30000
4	Rohan	C1	DBMS	F1	John	30000

Where,

Sid = Student id

Sname = Student Name

Cid = college id

Cname = college Name

Fid = faculty Id

Fname = faculty Name.

→ Problem on Column Level Redundancy

- Insertion Anomaly
- Deletion Anomaly
- Updation Anomaly

→ Anomaly means problem (Special)

• Insertion Anomaly

Sid	Sname	Cid	Cname	Fid	Fname	Salary
1	Ram	C1	DBMS	F1	John	30000
2	Shyam	C2	JAVA	F2	Bob	40000
3	Ravi	C1	DBMS	F1	John	30000
4	Rohan	C1	DBMS	F1	John	30000
		C10	C++			

• Deletion Anomaly

- Delete from Student where Sid = 1
- Delete from Student where Sid = 2

Sid	Sname	Cid	Cname	Fid	Fname	Salary
.						
.						
3	Ravi	C1	DBMS	F1	John	30000
4	Rohan	C1	DBMS	F1	John	30000
.						

• updation Anomaly

- Update Student Set Sname = "Ravi Kumar" where Sid = 3
- Change Salary of F1 from 30000 to 400000

Sid	Sname	Cid	Cname	Fid	Fname	Salary
1	Ram	C1	DBMS	F1	John	30000
2	shyam	C2	JAVA	F2	Bob	400000
3	Ravi	C1	DBMS	F1	John	50000
4	Rohan	C1	DBMS	F1	John	300000
5	Sagar	C3	C++	F2	John	400000

Remove the column level Redundancy

1.) Divide the table into multiple table.

Student

<u>Sid</u>	Sname
1	Ram
2	shyam
3	Ravi
4	Rohan

College

<u>Cid</u>	Cname
C1	DBMS
C2	JAVA

Faculty

<u>Fid</u>	Fname	Salary
F1	John	30000
F2	Bob	40000

* Normalization:-

It is the process of minimizing redundancy from a relation or set of relation. Redundancy in relation may cause insertion, deletion and update anomalies.

So, it helps to minimize the redundancy in relation. Normal forms are used to eliminate or reduce redundancy in database tables.

1NF (First Normal Form)

→ A relation is in first normal form if every attribute in that relation is single valued attribute.

(The table should not contain any multi-valued attributes)

Relation Student in table is not in 1NF because of multi-valued attribute Course.

Student

Roll no	Name	Course
1	Ram	C/C++
2	Rohan	Java
3	Shyam	C/DBMS

→ primary key - Roll no.

Relation Student in table is in 1NF because every attribute in that relation is single valued attribute.

- Relation Name - Student
- primary Key - Roll no

Roll no	Name	Course
<u>1</u>	Ram	C
1	Ram	CH
2	Rohan	Java
3	Shyam	C
3	Shyam	DBMS

2NF (second Normal form)

- A relation or table must be in first normal form.
- All non-prime attribute should be fully functional dependent on candidate key. (There should be no partial dependency in the relation).

Relation Teacher in table is not in 2NF.

- Candidate Keys : {Teacher id, subject}
- Non prime attribute : Teacher - age.

Teacher-id + subject → age, ck = $\frac{Tid + subject}{PA}$

Teacher-id → age

LHS → ck of Part RHS → APA

1 C++
2 DBMS
3 C
→ OS

Teacher-id	Subject	Age
1	C	30
1	C++	30
2	DBMS	30
3	C	40
3	OS	40

The table is in 1NF because each attribute has atomic values.

However, it is not in 2NF because non-prime attribute teacher-age is dependent on teacher-id alone which is a proper subset of candidate key.

* To make this table in 2NF we can break it in two tables (To remove partial dependencies we divided it into two parts).

Teacher - details table:

Teacher-id	age
1	30
2	30
3	40

Teacher - Subject table:

Teacher - Id	Subject
1	C
1	C++
2	DBMS
3	C
3	OS

* Prime attribute:-

prime attributes are attributes that are part of a candidate key. It is also known as key attributes.

prime attributes have unique values within a relation.

Eg - Teacher id.

* Non-prime attribute:-

* Non-prime attributes are attributes that are not part of any candidate key. It is also known as non-key attributes.

* Non-prime attributes may have duplicate values within a relation.

Eg - Subject, age.

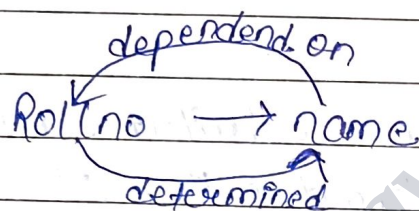
* Functional Dependencies:- when candidate determined non-prime attribute, then we say that relation is functional dependencies.

Rollno / Name / age

prime $\rightarrow$ Rollno + name $\rightarrow$ CK (Composite Key)

Rollno. ?
name } Prime attribute

Age ? - non-prime.



CK $\Rightarrow$ Rollno $\rightarrow$ PA
name $\rightarrow$ NPA
age $\rightarrow$ NPA

CK $\rightarrow$ Roll + Name

PA $\rightarrow$ Rollno

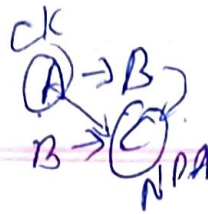
PA $\rightarrow$ Name

NPA $\rightarrow$ age

Roll $\rightarrow$ age

$\rightarrow$ {partial NPA}

CK $\rightarrow$ NPA
NPA $\rightarrow$ NPA



$A \rightarrow C$



3NF-

- Table or relation must be in 2NF.
- There should be no transitive dependency in table
- A Relation in 3NF if it is in 2NF and no non-key attribute is transitive dependent on the primary key.

①	②	③ (PK)	④	⑤
Roll no	S Name	S State	S Country	Age
1	A	Bihar	India	18
2	B	Chhatisgarh	India	20
3	C	Bihar	India	18

Candidate key $\rightarrow$ Roll no

Roll no, S name Functional dependencies -
S State age

Roll no $\rightarrow$ determined S Name ✓ Roll no $\rightarrow$ S Name
 Roll no $\rightarrow$ determined S State ✓ Roll no $\rightarrow$ S State
 S State $\rightarrow$ determined S Country ✗ S State $\rightarrow$ S Country
 Roll no $\rightarrow$ Age ✓

Now, ✗ Non-prime att. Should not be the transitive form of candidate key.

Roll no - S name, S State, age ✗ To check transitive dependency we have 2 functional dependency.

CK (PK) $\rightarrow$ 1 $\rightarrow$ 2 $\rightarrow$ (NPA)
 (NPA) $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ (NPA)

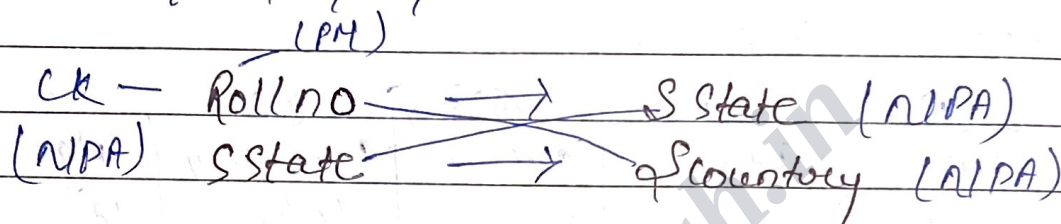
Each non-key directly depends on key Here, 1 determined 2 and 2 determined 3 i.e. 1 determined 3.

Functional dependency is a relationship two attributes in database table.
It means the value of one attribute uniquely determines the value of another attribute.



Transitive dependency:-

A transitive dependency occurs when a non-key attribute depends on the another non-key attribute rather than directly on the primary key.



S Country transitive dependency on Roll no

Non-Key attribute is dependence on Candidate Key.
i.e. $\varnothing$

non-key attribute — S State, S Country.

Functional Dependencies

Defines a relation b/w two attribute.

Determinant attribute — (X) → (Y) —> dependent attribute.

X → determined y or y

y is dependent on X.

Functional dependencies is used to check the data is student or not.

Roll no → S name

Roll no. defined 25 → Sagar
at there is no 34 → Sagar
student data.

St id	Name	Course	St id $\rightarrow$ Name
101	AB	CSE	St id $\rightarrow$ Course
102	CD	IT	means name and course are functionally dependent on st id.
103	AC	CSE	

Types of Functional Dependencies

(I) Trivial Functional Dependency:-

If $X \rightarrow Y$ and Y is a subset of X . The intersection of $X \cap Y \neq \emptyset$ (null).

Roll $\rightarrow$ Roll, Name $\rightarrow$ Roll
 $\rightarrow$ Roll determined Roll.

$X = \{1, 2, 3\}$

$Y = \{1\}$

Subset means the property should match.

St id $\rightarrow$ St id, Name

(II) Non-Trivial Functional Dependency:-

If X determines Y and Y is not subset of X .
 $(X \rightarrow Y)$

$X \cap Y = \emptyset$

Rollno $\rightarrow$ name

State $\rightarrow$ Country

St id $\rightarrow$ name

* properties

(I) Reflexivity:- If $X \rightarrow Y$ and Y is subset of X then $X \rightarrow Y$

Eg $\rightarrow$ Roll $\rightarrow$ Rollno

St id $\rightarrow$

(II) Argumentation:- If $X \rightarrow Y$ then $XZ \rightarrow YZ$

eg. if Roll $\rightarrow$ name

then {Roll, age} $\rightarrow$ {name, age}

(iii) Transitive:— If $x \rightarrow y$ and $y \rightarrow z$ then $x \rightarrow z$.
 eg - Roll $\rightarrow$ name
 class $\rightarrow$ Branch
 Roll $\rightarrow$ Branch.

(iv) Union:— If $x \rightarrow y$ and $x \rightarrow z$ then $x \rightarrow yz$
 eg - Rollno $\rightarrow$ sname
 Rollno $\rightarrow$ age
 Rollno $\rightarrow$ {sname, age}

(v) Decomposition:— If $x \rightarrow yz$ then $x \rightarrow y$ and $x \rightarrow z$
 Rollno $\rightarrow$ {sname, age}
 Rollno $\rightarrow$ sname
 Rollno $\rightarrow$ age

(vi) Composition:— If $x \rightarrow y$ and $A \rightarrow B$ then $xA \rightarrow yB$
 eg - Rollno $\rightarrow$ sname
 age $\rightarrow$ year
 {Rollno, age} $\rightarrow$ {sname, year}

* If the LHS contain Candidate Key then it's 3NF. OR
 * If the RHS contain prime att. then it's 3NF.
 CK $\rightarrow$ PA/NPA $\rightarrow$ CK/PA/NPA $\rightarrow$ PA

BCNF (Boyce Codd Normal Form)

BCNF defines the left hand side always contain a candidate key.

* BCNF is advanced version of 3NF.

A table is in BCNF if every functional dependencies $X \rightarrow Y$, X is a candidate key or a super key of the table.

$P \rightarrow R$
 $P \rightarrow N$

Student

Rollno	Name	Adhar no	Age
1	ABC	123	18
2	CDE	456	19
3	EFG	789	20 18
4	GHI	101	21

Relation name - Student

Candidate Key $\rightarrow$ Rollno.

Adhar no.

Functional dependencies

Rollno. $\rightarrow$ name

Adhar no. $\rightarrow$ age

Rollno $\rightarrow$ Name, Rollno $\rightarrow$ Adhar no

Adhar no $\rightarrow$ Age

In the above table the relation is in BCNF because determinant attribute is candidate key.

* This is a

* For BCNF, the table should be in 3NF, and for every functional dependency in left hand side is a candidate or super key.

Multivalued Dependency:-

- multivalued dependency occurs when two attributes in a table are independent of each other but both depend on a third attribute.
- A multivalued dependency consists of at least two attributes that are dependent on a third attribute. That's why it always requires at least three attributes.

$$X \twoheadrightarrow Y \quad \twoheadrightarrow Z \quad \twoheadrightarrow YZ$$

Bike manufacturer

Bike-model	manuf. year	Color
M2001	2008	White
M2001	2008	Black
M3001	2013	White
M3001	2013	Black
M4006	2019	White
M4006	2019	Black

Bike model $\rightarrow$ manuf-year

Bike model $\rightarrow$ Color

Bike model determines manufacture year and Bike model determines color but manufacture year and color independent with each other.

4NF

* A relation is in 4NF iff and only if the following condition are satisfied.

- * i) It should be in BCNF.
- * ii) The table should not have any multivalued dependency.

Student		
Rollno	Course	Hobby
21	CN	cricket
34	physics	Singing
22	EEE	Dancing
32	DBMS	Hockey

Rollno $\rightarrow$ Course

Rollno $\rightarrow$ Hobby

Here, Rollno determined Course and Rollno determined Hobby but Course and Hobby are independent with each other i.e. it is not a 4NF.

Date _____
Page _____

De composition

lossy
↳ with the loss, destory
cannot provide original
data.

(If the table form original)

CSE

Subject	Teacher	Sem
DBMS	ARC	3
DBMS	DEF	3
CHMA	DEF	3
CHMA	XYZ	2
CN	PQR	1

R1	
Subject	sem
DBMS	3
CHNA	3
CHNA	2
CN	1

Teacher	Score
ABC	3
DEF	3
XYZ	2
PQR	1

Subject	Teacher
DBMS	ABC
DBMS	DEF
CHMA	DEF
CHMA	XYZ
CAI	PQR

$$R_1 \boxtimes R_2 = R_4$$

$$R_1 \text{ AND } R_2 = R_4$$



Subject	Sem	Teacher	Sem
DBMS	3	ABC	3 ✓
DBMS	3	DEF	3 ✓
DBMS	3	XYZ	2 ✗
DBMS	3	PQR	1 ✗
CHMA	3	ABC	3 ✓
CHMA	3	DEF	3 ✓
CHMA	3	XYZ	2 ✗
CHMA	3	PQR	1 ✗
CHMA	3	ABC	3 ✗
CHMA	3	DEF	3 ✗
CHMA	2	XYZ	2 ✓
CHMA	2	PQR	1 ✗
CN	3	ABC	3 ✗
CN	3	DEF	3 ✗
CN	2	XYZ	2 ✗
CN	1	PQR	1 ✓

(R₄)

Subject	Sem	Teacher
DBMS	3	ABC
DBMS	3	DEF
CHMA	3	ABC
CHMA	3	DEF
CHMA	2	XYZ
CN	1	PQR

Then it is joined with R₃

Subject	Teacher	Semester	Subject	Teacher
DBMS	ABC	3		
DBMS	DEF	3		
CHMA	DEF	3		
CHMA	XYZ	2		
CN	PQR	1		

→ A relation in 4NF is only if satisfied when all conditions.

→ It should satisfy all conditions of 4NF.
→ It cannot contain any joint dependencies and joining should be lossless.

Relational Algebra

Relational Algebra provides a theoretical foundation for relational database and SQL.

The main purpose using relational algebra is to define operators that transform one or more input relations into a output relation.

These operators accept relation as input and produce relation as output.

Operators

- i) Selection (σ) (Select * from table-name where condition)
- ii) Projection (π)
- iii) Union ($\cup$)
- iv) Intersection ($\cap$)
- v) Minus ($-$)
- vi) Rename (ρ)
- vii) Cartesian product or cross product ($\times$)
- viii) Division ($\div$)
- ix) Join ($\bowtie$)

*

i) Selection (σ) :-

an SQL

Select * from table-name where condition;

DQ1

Selection - To select tuple — for full
Projection - To select attribute.
↳ To access one or two attribute data

Roll no	Name	Marks
101	A	10
102	B	20
103	C	30
104	D	40
105	E	20

- ⇒ The selection operation selects tuples that satisfy a given predicate.
- ⇒ It is used to select required tuples of a relation.
- ⇒ It is denoted by Sigma (σ)

Notation :- $\sigma_P(R)$

where ① σ (selection) is used for selection predicate ② R is used for relation ③ P is used for propositional logic formula which may be connector AND OR NOT.

Notes - These relational ^{Algebra} operators can be used as relational operators.

Like $=, \neq, <, >, \leq, \geq$ etc.

eg. Retrieve the student details where marks is greater than 25.

$\sigma_{\text{marks} > 25}(\text{student})$

Roll no	name	marks
103	C	30
104	D	40

ii) projection (π):-

It is used to ~~can~~ project required tuple attribute data from relation.

It is denoted by (π)

Notation:-

$\pi_{A_1, A_2, \dots, A_n}(R)$

where $A_1, A_2, \dots, A_n$ is use as an attribute name.

Also R is a relation.

Eg - fetch the student name.

$\pi_{\text{Name}}(\text{student})$

Name
A
B
C
D

In SQL

syntax:- Select attname1, att2 --- attn from table_name
Where Condition;

Find the name of student whose marks is greater than 25.

$\pi_{name}(\sigma_{marks > 25}(\text{Student}))$

name
C
D

Q. Find the Roll number of student whose marks is equal to 30.

$\pi_{\text{Rollno}}(\sigma_{marks = 30}(\text{Student}))$

Rollno
103

Q. Find the name of student whose marks is equal to 20 and Rollno = 105.

$\pi_{name}(\sigma_{marks = 20 \text{ AND } Rollno = 105}(\text{Student}))$

Name
E

iii) union:-

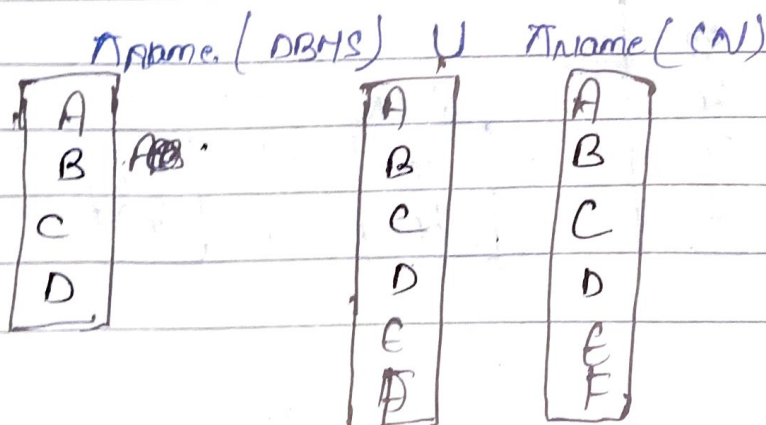
Notation:- $R_1 \cup R_2$

Note:- A set operation must hold the following condition:-
 Relation R_1 and Relation R_2 must have the same
 number of attribute. (Domain name also be same i.e. Datatype)
 (4pt)

* Duplicate tuple are eliminated.

DBMS		CNI	
Name	DBMS: Marks	Name	CNI: Marks
A	70	A	50
B	50	B	40
C	60	C	17
D	65	E	55
		F	60

Find the name of student which is enrolled in DBMS
 and CNI & either DBMS or CNI



Intersection ($R_1 \cap R_2$)

It contains all tuples (attribute) that are in both relations.

Q. Find the name of Student which is enrolled in both Subject.

$$\pi_{\text{Name}}(\text{DBMS}) \cap \pi_{\text{Name}}(\text{CN})$$

A
B
C

minus ($-$)

$$R_1 \cup R_2 = R_2 \cup R_1$$

$$R_1 \cap R_2 = R_2 \cap R_1$$

$$R_1 - R_2 \neq R_2 - R_1$$

R_1	R_2
10	20
20	50
30	70

$R_1 - R_2$	$R_2 - R_1$
10	50
30	70

$$[R_1 - R_2]$$

Suppose there are two relation R_1 & R_2 . The set difference operation contains all tuples that are in R_1 but not in R_2 .

Q. Find the name of student which is enrolled in ~~DBMS~~ DBMS & ~~DBMS~~ DBMS.

$$\pi_{\text{name}}(\text{DBMS}) - \pi_{\text{name}}(\text{CAU})$$

Name
D

- D is not common

(V) Rename (R)

This operation is used to rename the file.

It is denoted by (P) also.

P new relation name, old relation name

eg- $P(\text{stu}, \text{Student})$

Attribute name

$P(\text{relation name}) (a_1, a_2, a_3 \dots a_n)$

eg- $P(\text{Student}) (\text{name}, \text{B-violino})$

Before

Student	
name	Rollno

After

Student	
name	B Rollno

(vii) Cross product (X)

The cross product is used to combine each row in one table with each row in another table.

It is denoted by X

Note:- In cross product the no. of rows and number of column = $a+b$.

Where as

M - no. of rows in first relation.

N - No. of rows in second relation.

a - no. of column in first relation.

b - no. of column in second relation.

employee		
eid	ename	dept
1	M42	A
2	MNO	R

Department	
deptno	deptname
A	Cst
B	elec
C	mech

employee X department

Eid	Ename	Edept	Deptno	Deptname
1	Y42	A	A	CSE
1	Y42	A	B	Elec
1	Y42	A	C	Mech
2	MNO	R	A	CSE
2	MNO	R	B	Elec
2	MNO	R	C	Mech

(vii) Division ($\div$) for all.

The division operators is used for query which involve 'for all'.

It is denoted by ($\div$)

Notation:- $R1 \div R2$

Name	Course
A	Btch
B	Diploma
B	B. tech
C	Btceh

R2

Course
Btceh
Diploma

Q. Find the name of student of data who enroll in all course for all.

$$R1 \div R2$$

Name
B

(IV) Join

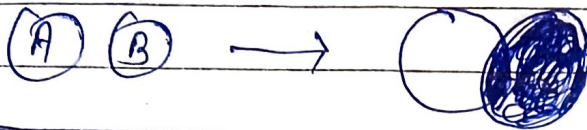
A join operation combine related tuple from different relation, if and only if a given join condition is satisfied.

It is denoted by $\bowtie$

Types of join

- (i) outer join
- (ii) Inner join
- (iii) Right join
- (iv) Full join

(ii) Right outer join

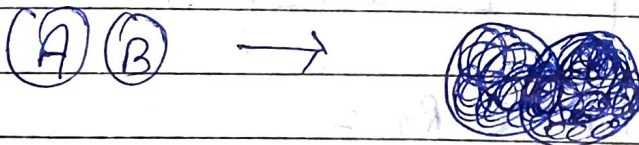


Num	Square	Cube
2	4	8
3	9	27
5	Null	125

$R_1 \bowtie R_2$

(iii) Full outer join

Num	Square	Cube
2	4	8
3	9	27
4	16	Null
5	Null	125



$R_1 \bowtie R_2$

• Natural join

works as equijoin but not need to write condition i.e. equal sign.

It is denoted $R_1 \bowtie R_2$

R1				
Num	Square		Num	Cube
2	4		2	8
3	9		3	27

R1 $\bowtie$ R2				
Num	Square	num	Cube	
2	4	2	8	✓
2	4	3	27	✗
3	9	2	8	✗
3	9	3	27	✓

Num	Square	Cube
2	4	8
3	9	27

(ii) Inner join

(i) theta join

It is denoted by $R_1 \bowtie R_2$.

$$R_3 \bowtie R_1 \cdot \text{col}_2 \geq R_2 \cdot \text{col}_2$$

R ₃				
col ₁	col ₂	col ₁	col ₂	
1	1	1	1	X
1	1	1	3	X
1	2	1	1	✓
1	2	1	3	X

R ₃		
col ₁	col ₂	Ans
1	2	

(ii) Equi join

It is denoted by $R_1 \bowtie = R_2$

R ₁		R ₂	
col ₁	col ₂	col ₁	col ₂
1	1	1	1
1	2	1	3

$$R_3 \bowtie R_1 \cdot \text{col}_2 = R_2 \cdot \text{col}_2$$

R2

col1	col2	col1	col2	
1	1	1	1	✓
1	1	1	3	✗
1	2	1	1	✗
1	2	1	3	✗

col1	col2
1	1

AB

Diplomawallah.in