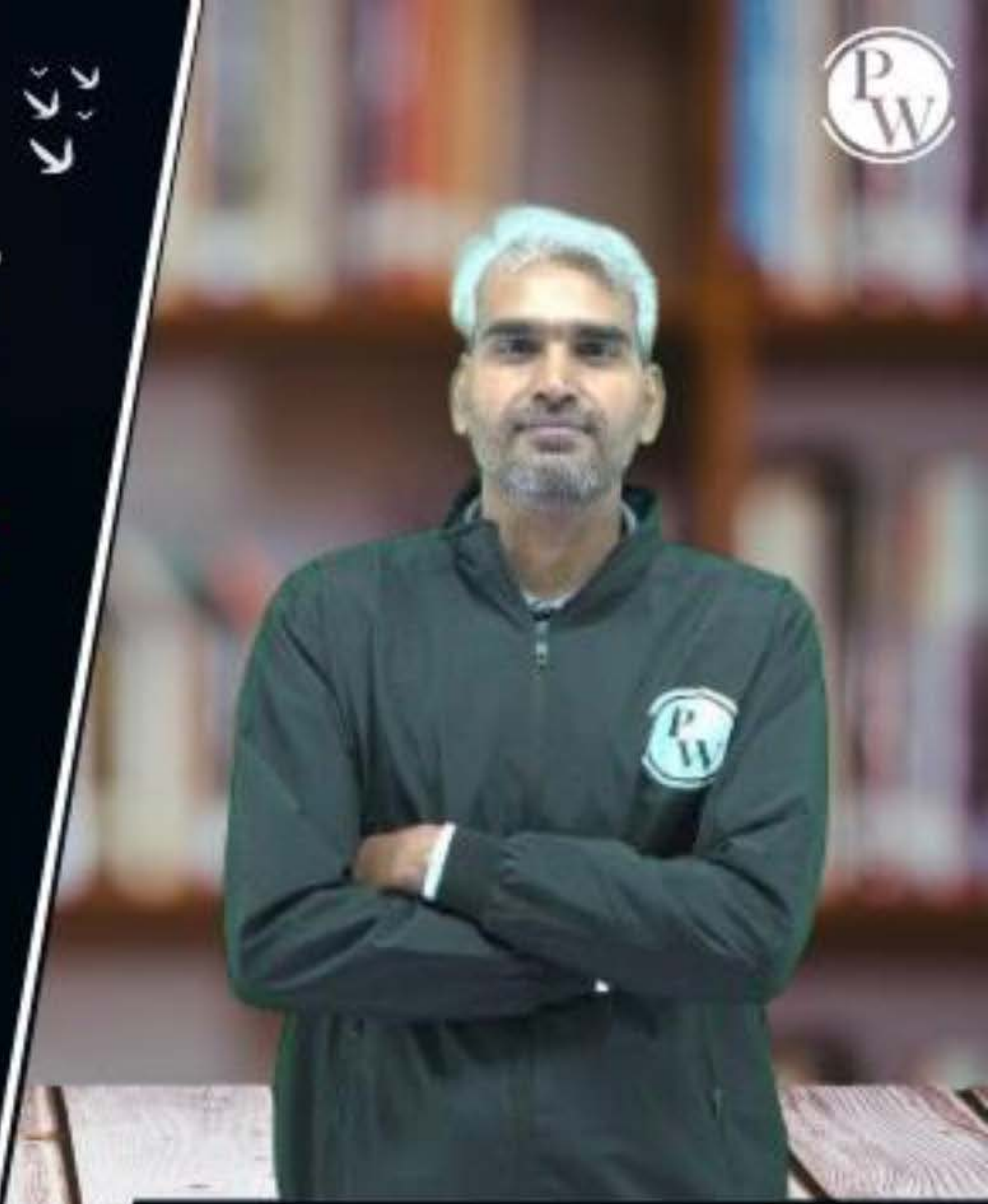
Computer Science & IT

Database Management System

Query Languages

Lecture No. 01



By- Vishal Sir

Recap of Previous Lecture

Topic Properties of multi-valued dependency

Topic 4NF

Topics to be Covered

- **Topic** Query languages
- **Topic** Relational Algebra
- **Topic** Basic relational algebra operations

Topic : Query languages

Query languages

Procedural query language

we define the procedure to retrieve the required information from the database

eg: $\boxed{\text{Relational Algebra}}$

Non-procedural query language

We use the syntax Provided by query language. to retrieve the required information from the database

eg: - Structured Query language (SQL)
- Tuple Relational Calculus (TRC)
- Domain Relational Calculus (DRC)

Relational Algebra

Note :- Relational algebra query Condition Evaluates tuple by tuple, taken only one tuple at a time.

↳ ∴ If we want to compare two or more tuples of the same table or different tables, then we need combine those tuples into a single tuple using "Join operation".

⇓

eg Cross Join (Cross Product), Natural Join or any other type of Join opⁿ

Note :- Relational algebra query will always produce distinct tuples.

{i.e., duplicate tuples will never be present in the output of Relational algebra query}

Topic : Relational Algebra

Relational Algebra is a procedural query language used to query the relational database tables to access data.

Relational Algebra operation can be classified into two types:

1) Basic Relational Algebra Operations
2) Derived Relational Algebra operations

Topic : Basic Relational Algebra operators

1. Projection(π)
2. Selection (σ)
3. Cross Product ($\times$)

Already done

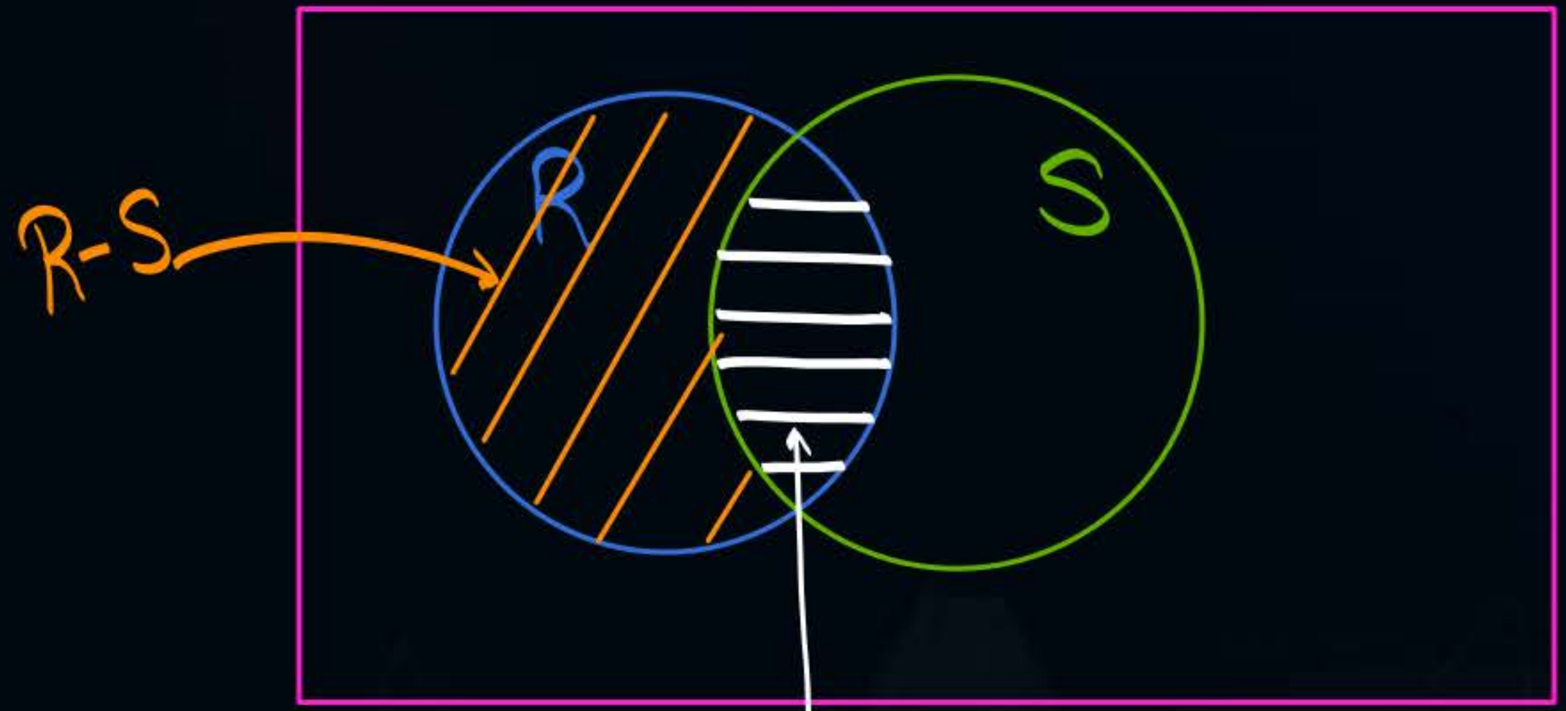
4. Union ($\cup$)
5. Set Difference ($-$)

Set operations

6. Rename (ρ)

Topic : Derived Relational Algebra operators

1. Intersection (∩)
2. Join Operations ("⋈")
3. Division Operation (÷)



$R \cap S = R - (R-S)$

Topic : Projection (π)

It is used to project the column data from a relation based on the attributes specified with projection operation.

Name of relation

e.g., $\pi_{\langle\text{attribute list}\rangle}(R)$

list of attributes required in o/p of query

Projection operator need not obey commutative property i.e.

$$\pi_{\langle\text{list2}\rangle}(\pi_{\langle\text{list1}\rangle}(R)) \neq \pi_{\langle\text{list1}\rangle}(\pi_{\langle\text{list2}\rangle}(R))$$

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

① Retrieve all records from relation R

i.e; all attributes are required in o/p

$$\pi_{Sid,\,Cid,\,Branch}(R) \equiv (R)$$

Both will produce Complete relation R as output

If we do not use Projection opⁿ then all the attributes of relational Schema R will be Present in o/p

② Retrieve Sids of all students from relation R.

$$\pi_{Sid}(R) \Rightarrow o/p$$

Sid
S_1
S_2
S_3

③ Retrieve all Cids from relation R

$$\pi_{Cid}(R) \Rightarrow o/p$$

Cid
C_1
C_2
~~C_1~~

duplicate tuple will not be present.

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

(4)

$\pi_{Sid,\,Cid}(R) \Rightarrow$ o/p

Sid	Cid
S_1	C_1
S_2	C_2
S_3	C_1

tuple wise they are distinct

(5)

$$\pi_{\underbrace{Sid}_{\text{List 2}}}\left(\pi_{\underbrace{Sid,\,Cid}_{\text{List 1}}}(R)\right) \neq \underbrace{\pi_{\underbrace{Sid,\,Cid}_{\text{list 1}}}\left(\pi_{\underbrace{Sid}_{\text{list2}}}(R)\right)}_{\text{invalid query}}$$

Topic : Selection(σ)

It is used to select the tuples from underlying relation based on the predicate condition specified with selection operation.

$$\sigma_{\text{<selection\_condition>}}(R)$$

NOTE:

$$\sigma_{A \wedge B}(R) = \sigma_{B \wedge A}(R)$$

OR

$$\sigma_B(\sigma_A(R)) = \sigma_A(\sigma_B(R))$$

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

(1) $\pi_{Sid, Cid, Branch}(R) \equiv (R)$

We did not specify any selection, ∴ All the tuples of relation R are selected

(2) Retrieve records of the students who enrolled for Cource with Cid = C_1

i.e., in the o/p all attributes are required

$\sigma_{Cid = 'C_1'}(R) \Rightarrow$ o/p

Sid	Cid	Branch
S_1	C_1	CS
S_3	C_1	CS

i.e. Complete record will be present in O/p

Projection is not used, ∴ all attributes will be present in output.

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

③ Retrieve Sids of the students who enrolled for Cource with Cid = C_1

$$\pi_{Sid}\left(\sigma_{Cid='C_1'}(R)\right) \Rightarrow \text{O/P}:$$

Sid
S_1
S_3

R

Sid	Cid	Branch
S_1	C_1 ✓	CS ✗
S_2	C_2 ✗	IT
S_3	C_1 ✓	CS ✗

4. Retrieve records in which Cid is 'C_1' and Branch is 'IT'

$$\sigma_{\underbrace{Cid = 'C_1' \wedge Branch = 'IT'}_{\text{"AND"}}}(R)$$

$\Rightarrow$ O/P =

Sid	Cid	Branch

Out will be Empty relation.

{Schema will exist, but no tuple will be Present}

$$\sigma_{Branch = 'IT'}\left(\sigma_{Cid = 'C_1'}(R)\right)$$

$$\sigma_{Branch = 'IT'}(R) \cap \sigma_{Cid = 'C_1'}(R)$$

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

(5) Retrieve records in which Cid is 'C_1' or Branch is 'IT'

$$\sigma_{\underbrace{Cid = 'C_1'}\ \vee\ \underbrace{Branch = 'IT'}}(R)$$

OR

$\Rightarrow$ O/P =

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

$$\sigma_{Cid = 'C_1'}(R) \cup \sigma_{Branch = 'IT'}(R)$$

Topic : NOTE

let 'A' & 'B' are Conditions.

- $\sigma_{A \wedge B}(R) = \sigma_{B \wedge A}(R)$

OR

$\sigma_B(\sigma_A(R)) = \sigma_A(\sigma_B(R))$

(OR)

$\sigma_A(R) \cap \sigma_B(R)$

$\sigma_B(R) \cap \sigma_A(R)$

- $\sigma_{A \vee B}(R) \equiv \sigma_{B \vee A}(R)$

(or)

$\sigma_A(R) \cup \sigma_B(R)$

(or)

$\sigma_B(R) \cup \sigma_A(R)$

Topic : Cross Product (×) / Cross Join / Cartesian Join

Cross-product is a binary operation. Let R and S are any two relation, then cross product R × S will result in all attributes of R followed by all attribute of S with all possible combinations of tuples from R and S.

→ i.e. each tuple of R will join with each tuple of S 'x' attributes

R

m tuples

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

'y' attributes

S

n tuples

Sid	Sname
S_1	Ram
S_2	Mohan
S_3	Ram

'x+y' attributes

$R \times S =$

(m*n) tuples

R.Sid	R.Cid	R.Branch	S.Sid	S.Sname
S_1	C_1	CS	S_1	Ram
S_1	C_1	CS	S_2	Mohan
S_1	C_1	CS	S_3	Ram
S_2	C_2	IT	S_1	Ram
S_2	C_2	IT	S_2	Mohan
S_2	C_2	IT	S_3	Ram
S_3	C_1	CS	S_1	Ram
S_3	C_1	CS	S_2	Mohan
S_3	C_1	CS	S_3	Ram

Topic : Union, Set difference, Intersection

- Union, Set Difference and Intersection are the Set operations.

- To use set theory operators on any two relations, those relations must be union compatible.

- The union compatibility of relations implies that the participating relations must fulfil the following conditions.

1. Same degree, i.e. The two relations must have the same number of attributes.
2. Same domain of each corresponding attributes of relations

and

"data type" {i.e. the type of values that they can acquire}

R

Sid	Cid	Branch

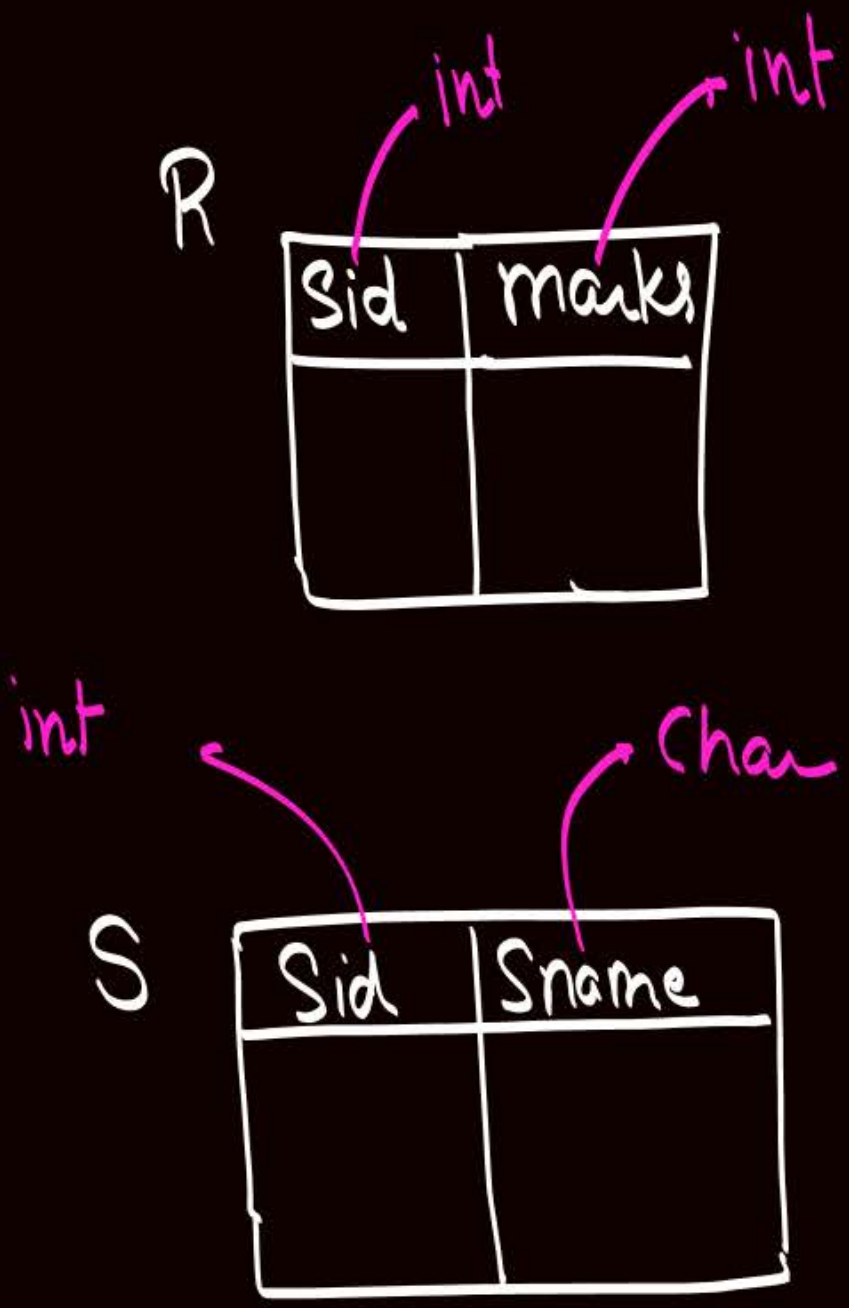
S

Sid	Sname

No. of attributes in R $\neq$ No. of attributes in S

$\downarrow$

$\therefore$ Not Union Compatible



① No. of attributes in R = No. of attributes in S ✓

② (i) Domain of 1st attribute of R = Domain of 1st attribute of S ✓

(ii) Domain of 2nd attribute of R ≠ Domain of 2nd attribute of S

∴ Not Union Compatible

int Char

R

Sid	Branch

int Char

S

Sid	Sname

① No. of attributes in R = No. of attributes in S

② (i) Domain of 1st attribute of R = Domain of 1st attribute of S

(ii) Domain of 2nd attribute of R = Domain of 2nd attribute of S

∴ R & S are Union Compatible

Note: (1) Set operations can be performed on two relations if and only if relations are Union Compatible

(2) After the set operation the resulting relation will take the names of its attributes from left hand side relation.

i.e, (i) In $R \cup S$, names of attributes will be same as names of attributes in Relⁿ R.

and (ii) In $S \cup R$, names of attributes will be same as names of attributes in Relⁿ S

Topic : Union, Set difference, Intersection

- Union $(A \cup B)$- It contains unique tuples from both the relations.

- Difference $(A - B)$- It contains all the tuples that are contained in the relation A but are not present in the relation B

- Intersection $(A \cap B)$- It contains all the tuples that are contained in both the relations A as well as in relation B.

Employee(E)

int Char

Eid	Ename
1	A
3	B
5	A
8	C

Student (S)

Int Char

Sid	Sname
2	D
3	B
5	A
7	C

$E \cup S =$

Eid	Ename
1	A
3	B
5	A
8	C
2	D
7	C

$S - E =$

Sid	Sname
2	D
7	C

$S \cap E =$

$\parallel$

$S-(S-E)$

Sid	Sname
3	B
5	A

$E - S =$

Eid	Ename
1	A
8	C

Topic : Rename (ρ)

Consider the following relational Schema
Students (Stu-id, Stu-name)

Rename operation can be used to rename attribute (Column) of the relation, name of the relation or both.

- Renaming a relation: $\rho_{\underbrace{FinalYrStudents}_{\text{New name for relation}}}(\underbrace{Students}_{\text{Name of relation}})$

- Renaming attributes: $\rho_{(\underbrace{SID}_{\text{New name for 1st attribute}}, \underbrace{Sname}_{\text{New name for 2nd attribute}})}(Students)$

- Renaming both: $\rho_{\underbrace{FinalYrStudents}_{\text{new table name}} (SID, Sname)}(Students)$

Topic : Rename (ρ)

Consider the following relational Schema

Students (Stu_id, Stu_name)

Rename operation can be used to rename attribute (Column) of the relation, name of the relation or both.

O/p = FinalYrStudents(Stu-id, Stu_name)

- Renaming a relation: $\rho_{FinalYrStudents}(Students)$

Students (SID, Sname)

- Renaming attributes: $\rho_{(SID,\ Sname)}(Students)$

FinalYrStudents (SID, Sname)

- Renaming both: $\rho_{FinalYrStudents\ (SID,\ Sname)}(Students)$

H.W. :- Read about

1. Difference b/w Equi-join & Natural join
2. Difference b/w 'inner join' & 'outer join'
3. Derivation of "division" relational algebra opn.

2 mins Summary

- **Topic** Query Languages
- **Topic** Relational algebra
- **Topic** Basic relational algebra operations

Slide

THANK - YOU

Slide