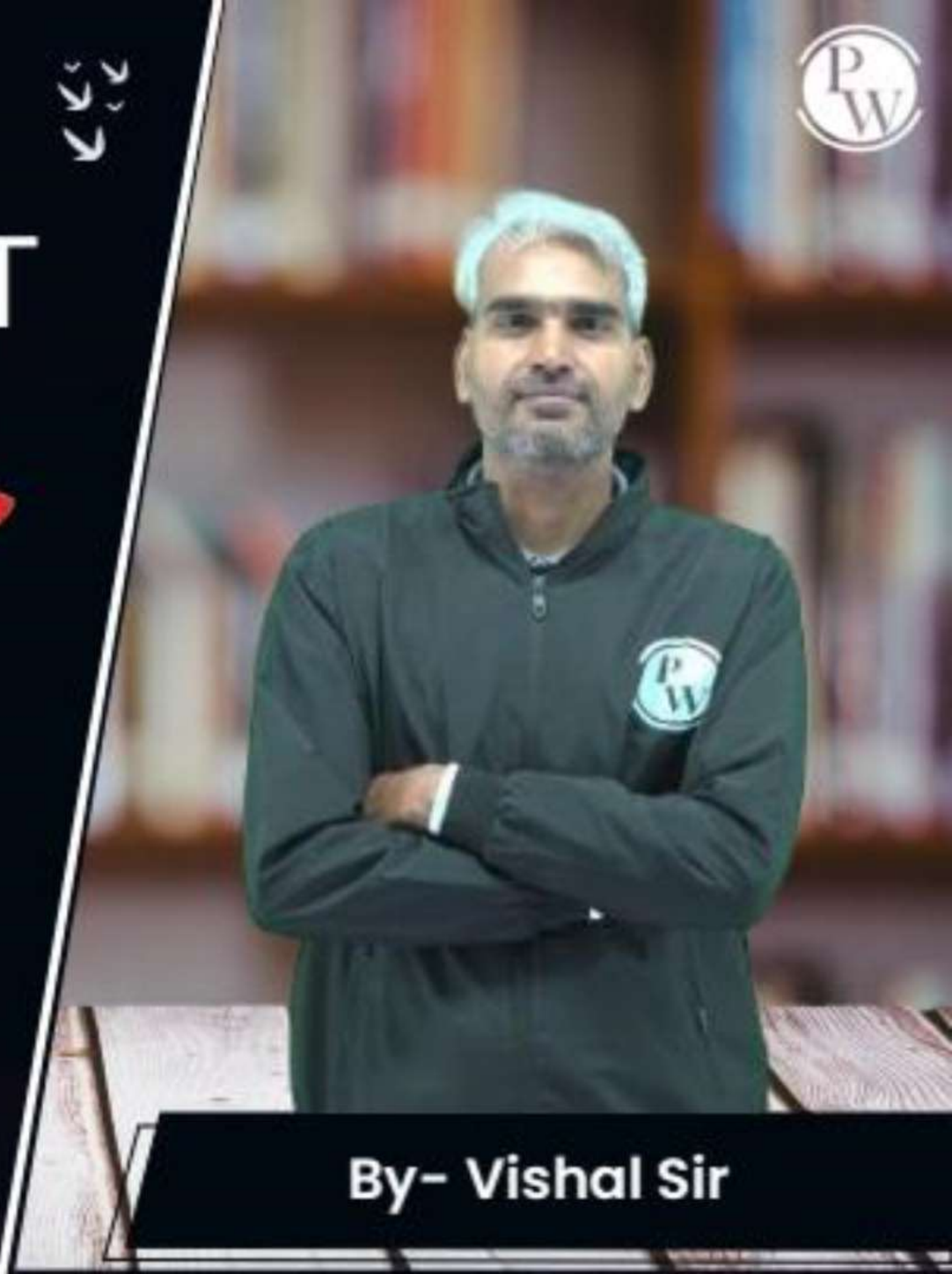
Computer Science & IT

Database Management System

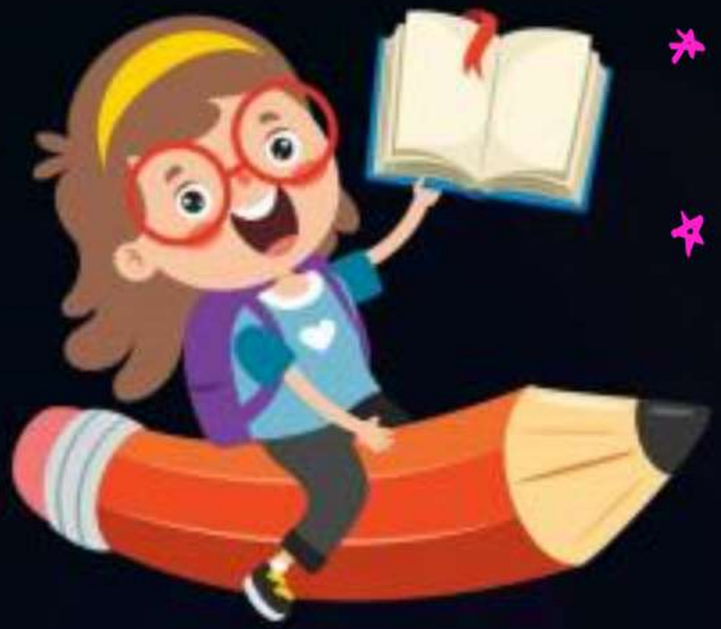
Query Languages

Lecture No. 02



By- Vishal Sir

Recap of Previous Lecture



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

Slide

Topics to be Covered

- Topic: Derived relational algebra operations
- Topic: Join operations
- Topic: Division operation

Slide

Employee (E)

int — Aadhaar_No. | char — Ename

Aadhaar_No.	Ename
123	A
439	A
728	B
385	C

Student (S)

int — Aadhaar_No. | char — Sname

Aadhaar_No.	Sname
430	A
439	A
385	C
400	D

$S \cap E =$

Aadhaar_No.	Sname
439	A
385	C

$E \cup S =$

Aadhaar_No.	Ename
123	A
439	A
728	B
385	C
430	A
400	D

$E - S =$

Aadhaar_No	Ename
123	A
728	B

$S - E =$

Aadhaar_No	Sname
430	A
400	D

Topic : Union, Set difference, Intersection

- Union (A ∪ B)- It contains all tuples from both the relations.
  {But no duplicate tuple}

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

$$A-B = \{x \mid x \in A \text{ and } x \notin B\}$$

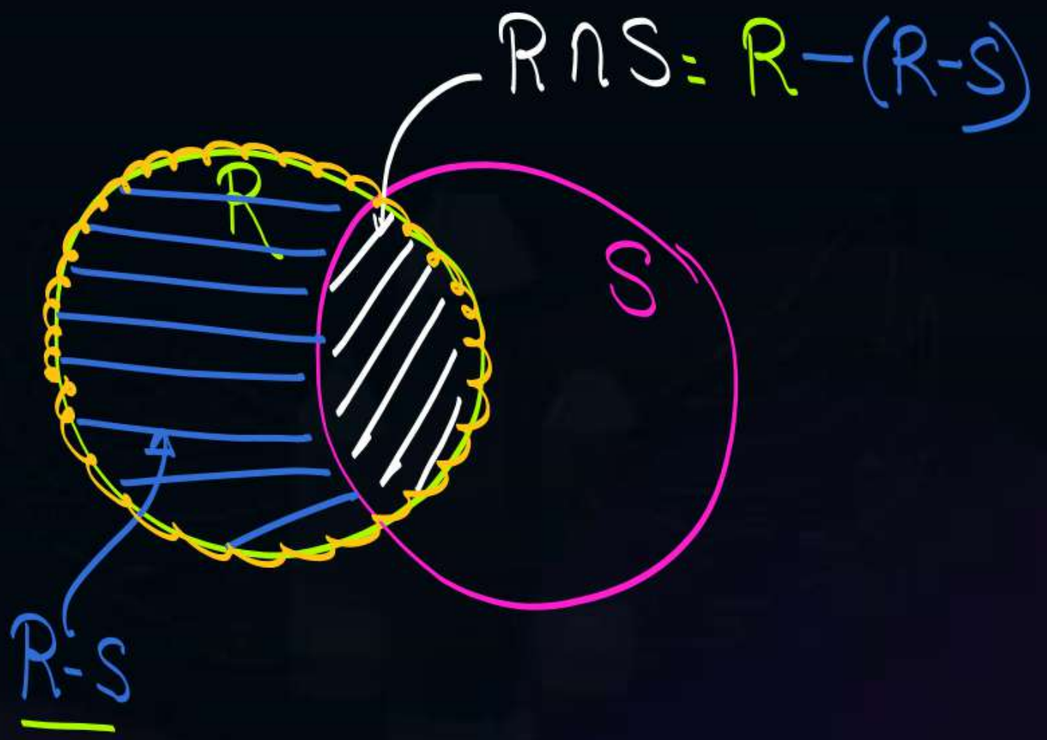
{Note: In general, $A-B \neq B-A$
$A-B = B-A$ only if $A = B$}

- Intersection (A ∩ B)- It contains all the tuples that are contained in both the relations A and B.

Topic : Intersection

Intersection is a derived relational algebra operation.
it is derived using "Set difference" operation

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



Topic : Rename (ρ)

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

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

Student

Student_id	Name_of_student

$\Rightarrow \rho_{\text{Final\_yr\_student}}(\text{Student}) \Rightarrow$

Final_yr_student

Student_id	Name_of_student

$\Downarrow \rho_{(\text{Sid}, \text{Sname})}(\text{Student})$

New name for 1st attribute

New name for 2nd attribute

Student

Sid	Sname

$\rho_{\text{Final\_yr\_student}(\text{Sid}, \text{Sname})}(\text{Student})$

Final_yr_student

Sid	Sname

Topic : Basic Relational Algebra operators

1. Projection(π)
2. Selection (σ)
3. Cross Product (×)
4. Union (U)
5. Set Difference (−)
6. Rename (ρ)

Topic : Derived Relational Algebra operators

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

Topic : Join Operations

Join operations are used to join relational tables based on some condition.

Cross Product ≡ Cross Join

In cross join, we join all tuples of first relation with all tuples of second relation.

Topic : Types Join Operations

Broadly join operations are Classified into two types:.

(1) Inner join
(2) Outer join

1) Inner join { In inner join we only select the tuples from both the relation that satisfied "join Condition" }
   a. Theta join
   b. Equi join
   c. Natural join

2) Outer join { We may also select the tuples from one or both the relation that did not satisfy the "join Condition" }
   a. Left outer join
   b. Right outer join
   c. Full outer join

Topic : Inner join

Inner join, includes only those tuples that satisfy the matching criteria.

"Condition"

Topic : Theta join / Conditional Join

The general case of JOIN operation is called a Theta join. It is denoted by symbol θ

- Theta join can use any condition in the selection criteria.

$<$, $>$, $\leq$, $\geq$, $=$, $<>$ (Not equal)

$R_1 \bowtie_\theta R_2$

θ can be any Condition

eg:

Student

Sid	Sname
2	A
4	B
3	A

Employee

Eid	Ename
2	B
4	A
5	B

$$Student \bowtie_{(Student.Sid < Employee.Eid)} Employee$$

(Student.Sid < Employee.Eid) — join Condition "⊙"

↳ This join Condⁿ can be any Condⁿ.

$$= \Pi_{\text{all attributes of Student \& Employee}} \left(\sigma_{Student.Sid < Employee.Eid} (Student \times Employee) \right)$$

{Not distinct}

Student.Sid	Student.Sname	Employee.Eid	Employee.Ename
2	A	4	A
2	A	5	B
4	B	5	B
3	A	4	A
3	A	5	B

Topic : Equijoin

An equijoin is a theta join using the equality operator.
In equijoin the condition is always equality condition.

It is a type of theta join. (in which Condition is Equality Cond^{n})

need not be on
Same or Common attribute

eg:

Student

Sid	Sname
2	A
4	B
3	A

Employee

Eid	Ename
2	B
4	A
5	B

$$Student \bowtie_{(Student.Sid = Employee.Eid)} Employee$$

↑ Equality Cond^n ∴ Equijoin

$$= \Pi_{\text{All attributes of Student \& all attributes of Employee}} \left(\sigma_{Student.Sid = Employee.Eid} (Student \times Employee) \right)$$

Student.Sid	Student.Sname	Employee.Eid	Employee.Ename
2	A	2	B
4	B	4	A

eg:

Student

Sid	Sname
2	A
4	B
3	A

Enroll

Sid	Cid
2	C_1
2	C_2
3	C_2

Using Equijoin, Common attributes from both the relation will be present in o/p.

$$Student \bowtie_{(Student.Sid = Enroll.Sid)} Enroll$$

(Student.Sid = Enroll.Sid ← Equality Cond^n ∴ Equijoin)

$$= \Pi_{\text{All attributes of Student \& all attributes of Enroll}} \left(\sigma_{Student.Sid = Enroll.Sid} (Student \times Enroll) \right)$$

=

Student.Sid	Student.Sname	Enroll.Sid	Enroll.Cid
2	A	2	C_1
2	A	2	C_2
3	A	3	C_2

Note: In theta join (or) Equijoin, we need to specify the join condition Explicitly, which is not the case with Natural join.

Note:- In Equijoin Equality condition can be applied on Common attributes or different attributes, there is no restriction.,

But in Natural join, ① we don't need to specify the join Cond^{n}.
& ② Join Cond^{n} is always Equality Cond^{n} on all Common attributes.

Topic : Natural join (⋈)

Natural join is performed based on the equality condition on all common attributes (column) between the relations. The name and type of common attributes must be same.
Additionally, a natural join removes the duplicate columns involved in the equality comparison so only one of each compared column remains

Student (Sid, Sname) & Enroll (Stud_id, Cid)

No Common attribute b/w Student & Enroll.

* Natural Join ($\bowtie$) :-

Natural join is a derived relational algebra op^n, which is derived using three basic relational algebra op^{n}s (i) Cross Product (ii) selection (iii) Projection

$R \bowtie S$ = R Natural join S

Step 1: Obtain R x S

Step 2: from "R x S" select the tuples based on equality Condn on all common attributes of R & S.

Step 3: from the output of Step-2, Project distinct attributes of R & S.

① $R(A,B,C)$, $S(A,D,E)$

$$R \bowtie S = \pi_{A,B,C,D,E}\left(\sigma_{R.A=S.A}(R \times S)\right)$$

② $R(A,B,C)$ & $S(B,C,D)$

$$R \bowtie S = \pi_{A,B,C,D}\left(\sigma_{(R.B=S.B \wedge R.C=S.C)}(R \times S)\right)$$

③ $R(A,B,C)$ & $S(D,E)$

$$R \bowtie S = \pi_{A,B,C,D,E}(\quad\quad(R \times S)) \equiv R \times S$$

No common attribute
∴ No selection condition
↳ Hence all tuples of R×S will be selected.

If there are No Common attributes b/w the relations, then o/p of Natural join will be same as Cross Product of those relations

eg:

Student

Sid	Sname
2	A
4	B
3	A

Employee

Eid	Ename
2	B
4	A
5	B

Student ⋈ Employee ≡ Student × Employee

Because No Common attribute

eg:

Student

Sid	Sname
2	A
4	B
3	A

Enroll

Sid	Cid
2	C1
2	C2
3	C2

Student ⋈ Enroll

Sid	Sname	Cid
2	A	C1
2	A	C2
3	A	C2

$$\Pi_{Sid, Sname, Cid}\left(\sigma_{Student.Sid = Enroll.Sid}\left(Student \times Enroll\right)\right)$$

Topic : Types Join Operations

1) Inner join
   a. Theta join
   b. Equi join
   c. Natural join

2) Outer join
   a. Left outer join
   b. Right outer join
   c. Full outer join

Topic : Outer join

In an outer join, along with tuples that satisfy the matching criteria, we also include tuples that do not match the criteria (condition) either from left hand side relation, or from right hand side relation or both.

Topic : Left Outer Join (⟕)

$R \overset{L}{\bowtie} S \equiv$ All tuples of $R \bowtie S$, and tuples from left hand side relation (i.e. 'R') that failed the join condition

R

A	B	C
1	2	3
1	4	5

S

B	C	D
2	3	4
5	6	9
3	7	8

$R \bowtie S =$

A	B	C	D	
1	2	3	4	} tuple from R ⋈ S
1	4	5	NULL	

Topic : Right Outer Join (⟖)

$R \overset{R}{\bowtie} S$ = All tuples of $R \bowtie S$, and the tuples from right hand side relation {i.e.: 'S'} that failed the join condition

R

A	B	C
1	2	3
1	4	5

S

B	C	D
2	3	4
5	6	9
3	7	8

$R \bowtie S =$

A	B	C	D
1	2	3	4
NULL	5	6	9
NULL	3	7	8

} tuple from $R \bowtie S$

Topic : Full Outer Join (⟗)

$R \text{ ⟗ } S$ = All tuples of $R \bowtie S$, and tuples from both left hand side relation & right hand side relation that failed join condition

R

A	B	C
1	2	3
1	4	5

S

B	C	D
2	3	4
5	6	9
3	7	8

R ⟗ S =

A	B	C	D
1	2	3	4
1	4	5	NULL
NULL	5	6	9
NULL	3	7	8

} from R ⟗ S

Q: Result of which of the following is Super-set of other three

(a) Natural join
(b) Left outer join
(c) Right outer join
(d) Full outer join

Topic : Division (÷)

Division operation is used whenever the query is with respect to every or all.

2 mins Summary

Topic — Derived relational algebra operations

Topic — Join operations

Topic — Division operation

} Next Class

Slide

THANK - YOU