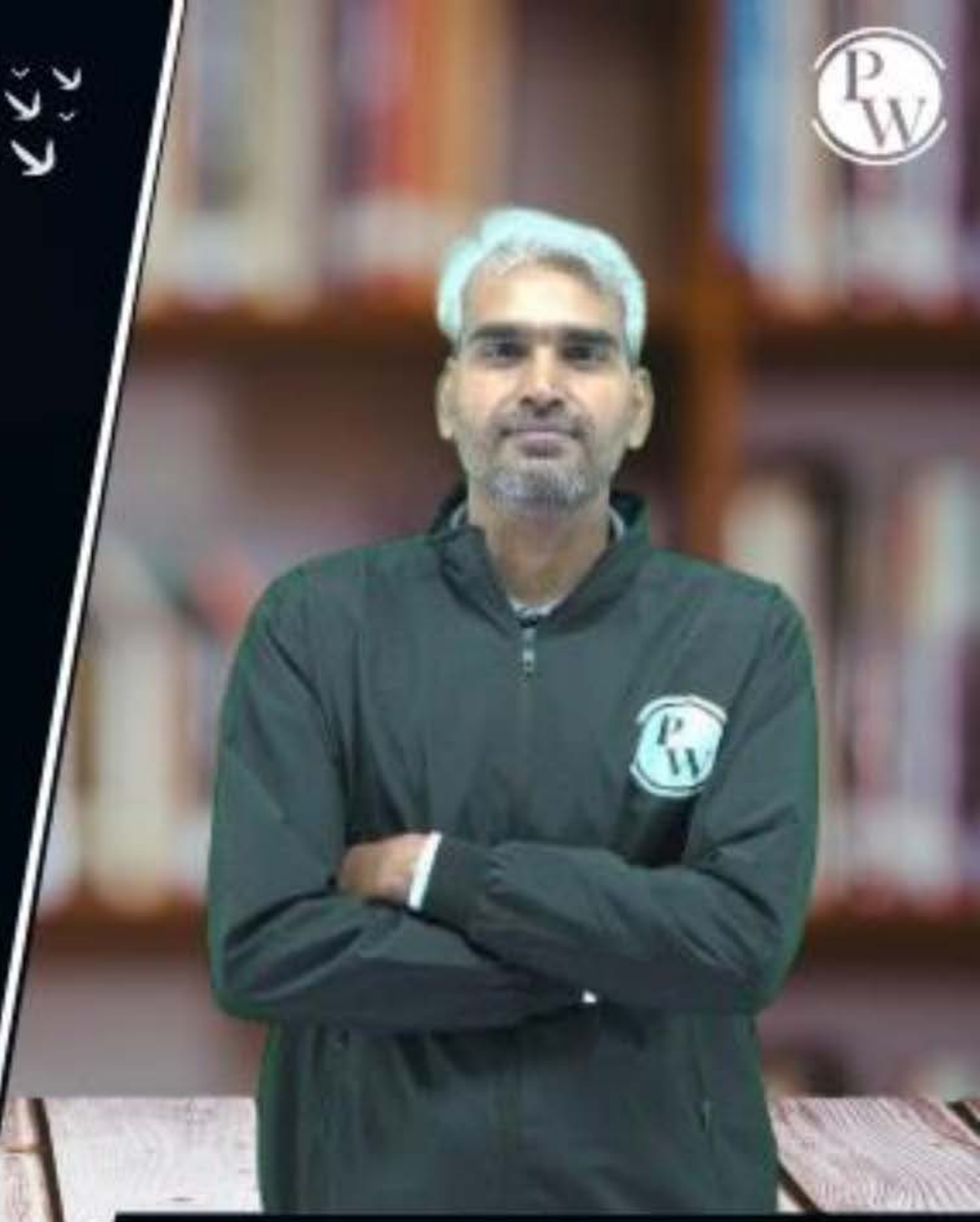
Computer Science & IT

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

Topics to be Covered

Topic Join operations

Topic Division operation

Topic : Derived Relational Algebra operators

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

* It is a set opⁿ, it can be performed only if relation are union compatible

⇓

Intersection is a derived relational algebra opⁿ, it is derived using "Set difference"

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

Topic : Join Operations

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

Using Cross Join each tuple of 1st table will be joined with each tuple of 2nd table

Topic : Types Join Operations

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

{Using inner join, we select only those tuples from both the relations that satisfy the join condition}

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

{In outer join we can also select the tuples from one or both the relations, that failed the join Cond^n}

Topic : Inner join

Inner join, includes only those tuples that satisfy the join condition.

Topic : Theta join

(Also Known as Conditional Join)

The general case of JOIN operation is called a Theta join. It is denoted by symbol θ. i.e. $\bowtie_{\theta}$

it can be any condition

- Theta join can use $\underbrace{\text{any conditions}}_{<,\ \leq,\ >,\ \geq,\ =,\ <>}$ in the selection criteria.

Student (S)

Sid	Sname
2	A
3	B
4	A

Employee (E)

Eid	Ename
2	B
4	A
5	B

We can use any Cond^n in theta join

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

it is the join Cond^n

$\equiv \left(\sigma_{S.Sid < E.Eid} (S \times E) \right)$

No projection op^n is used

$\therefore$ All attributes of "S×E" will be present in output.

O/P :

S.Sid	S.Sname	E.Eid	E.Ename
2	A	4	A
2	A	5	B
3	B	4	A
3	B	5	B
4	A	5	B

Topic : Equijoin

{ Equi-join is a type of theta join, where Cond^n is Equality Cond^n }

An equijoin is a theta join using the equality operator.

In equijoin the condition is always $\underbrace{\text{equality condition.}}$

It need not be on the Common attributes

It is a type of theta join.

Student (S)

Sid	Sname
2	A
3	B
4	A

Employee (E)

Eid	Ename
2	B
4	A
5	B

Attribute need not be Common attribute

Student $\bowtie_{(\text{Student.Sid} = \text{Employee.Eid})}$ Employee $\equiv \left(\sigma_{S.Sid = E.Eid} (S \times E) \right)$

Equality Cond^n
∴ It can be Called Equi-join

O/P :

S.Sid	S.Sname	E.Eid	E.Ename
2	A	2	B
4	A	4	A

✓ Student (S)

Sid	Sname
2	A
3	B
4	A

✓ Enroll (E)

Sid	Cid
2	C_1
4	C_2
5	C_3

Attribut may be Common attribute as well

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

Equi-join

$$\equiv \left(\sigma_{S.sid = E.Sid}(S \times E)\right)$$

O/p :

S.Sid	S.Sname	E.Sid	E.Cid
2	A	2	C_1
4	A	4	C_2

It is not natural join.
In Natural join we do not specify join Condn.

If we perform Equi-join, then Common attributes from each relation will be present in output.

Note:- If we use theta join or Equi-join then we need to specify the join condition Explicitly, Which is not the Case with natural join.

In natural join we do not specify any Cond^n, because by default Cond^n for natural join is Equality Condition 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 (Stu-id, Cid)

There is no common attribute between Student & Enroll w.r.t. Natural join

- ## Natural Join ($\bowtie$)

Natural join is a derived relational algebra opⁿ which can be derived using three basic relational algebra opⁿ. ie; Cross product, Selection and Projection.

Derivation of Natural join

$R \bowtie S =$ Step-1 : obtain $R \times S$

Step-2 : Select the tuples from $R \times S$ based on Equality Condⁿ on all common attributes of R and S.

Step-3 : From the output of Step-2 project distinct attributes

Student (S)

Sid	Sname
2	A
3	B
4	A

Enroll (E)

Sid	Cid
2	C_1
4	C_2
5	C_3

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

* No $cond^n$ is specified

∴ Natural join

O/P =

"Sid" will be present only once in the O/P

Sid	Sname	Cid
2	A	C_1
4	A	C_2

eg: $R(A\ B\ C)$ & $S(B\ C\ D)$

$$R \bowtie S = \pi_{A,B,C,D}\left(\sigma_{\underbrace{R.B = S.B \,\wedge\, R.C = S.C}_{\text{Equality Cond}^n \text{ on all common attributes}}}(R \times S)\right)$$

Equality Cond^n on all common attributes ✓

eg. $R(A, B, C)$ & $S(D, E)$

$$R \bowtie S = \pi_{A,B,C,D,E}\left(\underbrace{\qquad}_{\text{No Common attributes b/w R\&S}}(R \times S)\right) \equiv R \times S$$

No Common attributes b/w R&S

∴ No selection Condⁿ.

↓

And hence all the tuples of R×S will get selected

And finally all attributes of R×S will be projected.

∴ O/p will be exactly same as R×S

If there is no Common attribute between R and S, then Natural join of R and S will be equal to "R×S"

Student (S)

Sid	Sname
2	A
3	B
4	A

Employee (E)

Eid	Ename
2	B
4	A
5	B

$$\underbrace{\overset{(S)}{\text{Student}} \bowtie \overset{(E)}{\text{Employee}}} \equiv S \times E$$

No Common attributes b/w Student & Enroll

Topic : Outer join

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

Topic : Left Outer Join (⟕)

$R ⟕ S$ = All tuples of $R \bowtie S$, along with the tuples from left hand side relation {i.e. Relⁿ R} that failed the join Condⁿ

Topic : Right Outer Join (⟖)

$R ⟖ S$ = All tuples of $R \bowtie S$ along with the tuples of right hand side relation {i.e, Relⁿ S} that failed the join condition

Topic : Full Outer Join ($⟗$)

$R ⟗ S$: All tuples of $R \bowtie S$ along with the tuples from both side relation that failed the join condition

R :-

A	B	C
1	2	3
1	4	5

$R \bowtie S$ =

A	B	C	D
1	2	3	4

S :-

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

$R ⟕ S$ =

A	B	C	D
1	2	3	4
1	4	5	NULL

$R ⟖ S$ =

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

$R ⟗ S$ =

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

Division

Topic : Division (÷)

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

eg. Consider following three relations.

Student (S)

Sid	Sname
S_1	A
S_2	A
S_3	B
S_4	C

Enroll (E)

Sid	Cid	fee
S_1	C_1	-
S_1	C_2	-
S_2	C_2	-
S_1	C_3	-
S_2	C_3	-
S_3	C_3	-

Course (C)

Cid	Cname
C_1	OS
C_2	COA
C_3	DBMS

Query:- Retrieve Sids of the students who have enrolled for all courses.

Student (S)

Sid	Sname
S_1	A
S_2	A
S_3	B
S_4	C

Enroll (E)

Sid	Cid	fee
S_1	C_1	
S_1	C_2	
S_2	C_2	
S_1	C_3	
S_2	C_3	
S_3	C_3	

Course (C)

Cid	Cname
C_1	OS
C_2	COA
C_3	DBMS

* Retrieve Sids of the students who have Enrolled for all Courses.

→ We are actually looking for the Sids in the Enroll table that are associated with all 'Cids' of Course table

Desired result can be produced by the following relational algebra Expression

$$\pi_{sid, cid}(E) \div \pi_{cid}(C) \Rightarrow$$ o/p:

Sid
S_1

It will produce the Sids that are associated with all Cid of Course table

Enroll (E)

Sid	Cid
S_1	C_1
S_1	C_2
S_2	C_2
S_1	C_3
S_2	C_3
S_3	C_3

$$\underbrace{\pi_{sid}\ (E)}_{} \Rightarrow \quad O/P =$$

Sid
S_1
S_2
S_3

This R.A. Expression will produce the Sids of the Students who have enroll for some Course
{at least 1}

Course(C)

Cid	Cname
C_1	OS
C_2	COA
C_3	DBMS

$$\underbrace{\pi_{cid}\ (C)}\implies O/P =$$

Cid
C_1
C_2
C_3

This R.A. Expression will produce Cids of all courses.

* $\pi_{sid}(E) =$

Sid
S_1
S_2
S_3

* $\pi_{cid}(C) =$

Cid
C_1
C_2
C_3

$\underbrace{\pi_{sid}(E) \times \pi_{cid}(C)}$ $\Rightarrow$ o/p =

E.Sid	C.Cid
S_1	C_1
S_1	C_2
S_1	C_3
S_2	C_1
S_2	C_2
S_2	C_3
S_3	C_1
S_3	C_2
S_3	C_3

This R.A. Expression will produce a universal relation, in which Every student who has Enrolled for some courses is combined with all courses.

$\pi_{sid,cid}(E) \Rightarrow$ O/P =

Sid	Cid
S_1	C_1
S_1	C_2
S_2	C_2
S_1	C_3
S_2	C_3
S_3	C_3

$$\left(\pi_{sid}(E) \times \pi_{cid}(C)\right) - \pi_{sid,cid}(E) = \text{o/p} =$$

$\Rightarrow$ o/p =

E.Sid	C.Cid
S_1	C_1
S_1	C_2
S_1	C_3
S_2	C_1
S_2	C_2
S_2	C_3
S_3	C_1
S_3	C_2
S_3	C_3

Sid	Cid
S_1	C_1
S_1	C_2
S_2	C_2
S_1	C_3
S_2	C_3
S_3	C_3

Sid	Sid
S_2	C_1
S_3	C_1
S_3	C_2

* Retrieve Sids of the students who have Enrolled for all Courses.

Ans: $\pi_{sid,cid}(E) \div \pi_{cid}(C)$

$$= \left[\pi_{sid}(E) - \pi_{sid}\left[\left(\pi_{sid}(E) \times \pi_{cid}(C) \right) - \pi_{sid,cid}(E) \right] \right]$$

Sid
S_1
S_2
S_3

Sids of Students who entrolled for at least one Course

Sid
S_1

Sids of the students who enrolled for all Courses.

Sid
S_2
S_3

→ o/p will be Sids of the students who did not Enroll for all Courses.

Sid	Cid
S_2	C_1
S_3	C_1
S_3	C_2

→ In the o/p we will get the Sids of students who did not Enroll for all courses along with the Cids of the Courses for which they did not Enroll.

* Retrieve Sids of the students who have Enrolled for all Courses.

Ans: $\pi_{sid,cid}(E) \div \pi_{cid}(C)$

$$= \left[\pi_{sid}(E) - \pi_{sid}\left[\left(\pi_{sid}(E) \times \pi_{cid}(C) \right) - \pi_{sid,cid}(E) \right] \right]$$

Order of Execution ⇒ 1 8 7 2 4 3 6 5

Note :-

① $\pi_{A,B,C}(R) \div \pi_{C}(S) \Rightarrow$ O/P will contain the value (A, B) from relation R, that are associated with all values of attribute 'C' in relation S.

② $\pi_{A,B,C}(R) \div \pi_{B,C}(S) \Rightarrow$ O/P will contain values of attribute 'A' from relation R, that are associated with all values of (B, C) in relation S.

2 mins Summary

Topic — Join operations

Topic — Division operation

Slide

THANK - YOU

Slide