COMPUTER SCIENCE

Database Management System

Query Language

Lecture_3

Vijay Agarwal sir

TOPICS TO BE COVERED

01 Basic Operators

02 Derived Operators

1. Selection [σ]
2. Projection [Π]
3. Union [∪]
4. Intersection [∩]
5. MINUS/Set Difference [−]
6. CROSS Product
7. JOIN & its type

Relational Algebra

Basic operators

- π : Projection operator
- σ : Selection operator
- $\times$: Cross-product operator
- $\cup$: Union
- $-$: Set difference
- ρ : Rename operator

Relational Algebra

Derived operators

∩ : Intersection {using "–"}

⋈ : Join {using X, σ}

/ or ÷ : Division {using π, x, –}

Example:

Sailors (S_1)

Sid	Sname	Rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

Sailors (S_2)

Sid	Sname	Rating	age
28	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guppy	5	35.0
58	rusty	10	35.0

Relational Algebra

Basic operators

II. Cross product (×):

- $R \times S$: It result all attributes of R followed by all attributes of S, and each record of R paired with every record of S.
- Degree $(R \times S)$ = Degree (R) + Degree (S)
- $|(R \times S)| = |R| \times |S|$

NOTE:

- Relation R with n tuples and
- Relation S with 0 tuples then
- number of tuples in $R \times S = 0$ tuples

Cross – Product

Reserves (R_1)

Sid	bid	day
22	101	10/10/96
58	103	11/12/96

Sailors (S_1)

Sid	Sname	Rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

R_1 : 2 Tuple
3 Attribute

S_1 : 3 Tuple
4 Attribute

$R_1 \times S_1 = 2 \times 3 = 6$ Tuple
$3 + 4 = 7$ Attribute

S_1 R_1

$\underline{S_1 \times R_1} =$

S1.(Sid)	S1.Sname	S1.Rating	S1.age	R1.(Sid)	R1.bid	R1.day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

JOIN OPERATION

JOIN operation is used to Combined Information From Two or More Relations (Tables)

JOIN operation performed by the Cross Product fallowed by Selection & Projection.

Join Operations

(1) Conditional Join ($\bowtie_c$) [Theta Join]

(2) Equi join

(3) Natural join

Inner Join

(4) Left outer join

(5) Right outer join

(6) Full outer join

Outer Join.

Inner Join

1. Conditional (Theta) Join [$\bowtie_c$]
2. Equi Join
3. Natural JOIN.

CROSSJOIN

: Cross Product [R X S]

OUTER JOIN

1. [LOJ] Left outer Join ($R ⟕ S$) OR $R \bowtie_L S$
2. [ROJ] Right outer Join ($R ⟖ S$) OR $R \bowtie_R S$
3. FULL OUTER JOIN ($R ⟗ S$) OR $R \bowtie S$

FULL OUTER JOIN $\equiv$ OUTER JOIN $(R \overset{\circ}{\bowtie} S)$

OR

$R ⟗ S$

Natural Join ($R \bowtie S$)

$$R \bowtie S = \underbrace{\Pi_{\text{Distinct Attribute}}}_{\text{Step3}} \left[\underbrace{\sigma_{\text{equality Condition on All Common Attributes}}}_{\text{Step2}} \underbrace{(R \times S)}_{\text{Step1}} \right]$$

R(ABC) S(BCD) 3+3=6 Attribute

Distinct

$R \bowtie S$ =

A	B	C	D

Conditional JOIN $(R \bowtie_c S)$

$$R \bowtie_c S = \pi_{\text{ALL Attributes}}\left[\sigma_{\text{Given Condition}}(R \times S)\right]$$

Step3 Step2 Step1

R(ABC) S(BCD)

$R \bowtie_c S =$

RA	RB	RC	SB	SC	SD

Conditional Join

$$R \bowtie_{R_1.Sid > S_1.Sid} S = \pi_{\text{ALL Attribute}}\left[\sigma_{R_1.Sid > S_1.Sid}(R_1 \times S_1)\right]$$

Step3 Step2 Step1

Cross – Product

Reserves (R_1)

Sid	bid	day
22	101	10/10/96
58	103	11/12/96

Sailors (S_1)

Sid	Sname	Rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

R_1 : 2 Tuple
3 Attribute

S_1 : 3 Tuple
4 Attribute

$R_1 \times S_1 = 2 \times 3 = 6$ Tuple
$3 + 4 = 7$ Attribute

Conditional Join

$$R \bowtie_c S = \sigma_C (R \times S)$$

$\boxed{R_1.Sid > S_1.Sid}$

S_1 Sid

R_1 Sid

$S_1 \times R_1 =$

(Sid)	Sname	Rating	age	(Sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Conditional Join

$S1 \bowtie_{S1.sid < R1.sid} R1$

output

$R \bowtie_{R_1.Sid > S_1.Sid} S$

(Sid)	Sname	Rating	age	(Sid)	bid	day
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	58	103	11/12/96

$\boxed{S1 \bowtie_{S1.sid < R1.sid} R1}$

- Result schema same as that of cross-product.
- Fewer tuples than cross – product, might be able to compute more efficiently.
- Sometimes called a theta –join.

Cross – Product

Reserves (R_1)

Dangling Tuple

Sid	bid	day
22	101	10/10/96
58	103	11/12/96

Sailors (S_1)

Sid	Sname	Rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

Dangling Tuple

$\boxed{R_1.Sid > S_1.Sid}$

R_1 : 2 Tuple
3 Attribute

S_1 : 3 Tuple
4 Attribute

$R_1 \times S_1 = 2 \times 3 = 6$ Tuple
$3 + 4 = 7$ Attribute

1. **Dangling Tuple** : Dangling Tuple (Row) is a tuple that fail to Match (Satisfy) Any Tuple of other Relation in Common Attribute (or) (in Given Condition)

2. **Suburious Tuple** : [Extra Tuple]
   ↳ In Join

How to check Directly Dangling Tuple ?

Cross – Product

Reserves (R_1)

Sid	bid	day
22	101	10/10/96
58	103	11/12/96

Dangling Tuple

Sailors (S_1)

Sid	Sname	Rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

Dangling Tuple

$R_1.Sid > S_1.Sid$

Directly checking of Dangling Tuple

R_1 : 2 Tuple
3 Attribute

S_1 : 3 Tuple
4 Attribute

$R_1 \times S_1 = 2 \times 3 =$ 6 Tuple
$3 + 4 =$ 7 Attribute

Directly ?

R

A	B	C
1	3	5
4	6	7
7	9	11

Dangling Tuple

S

D	E	F	G
4	6	7	8
7	7	8	9

$R \bowtie_{R.B < S.D} S =$

$3 < 4$ True
$3 < 7$ True
$6 < 7$ True
$\left.\begin{array}{l} 9 < 4 \\ 9 < 7 \end{array}\right\}$ False.

Step By Step

R

A	B	C
1	3	5
4	6	7
7	9	11

Dangling Tuple

S

D	E	F	G
4	6	7	8
7	7	8	9

R S

3 Tuple 2 Tuple

3 Attribute 4 Attrib.

$R \times S = 3 \times 2 = 6$ Tuple

$3+4 = 7$ Attributes

$R \bowtie_{R.B < S.D} S =$

S.D > R.B

R.A	R.B	R.C	S.D	S.E	S.F	S.G
✓ 1	3	5	4	6	7	8
✓ 1	3	5	7	7	8	9
✗ 4	6	7	4	6	7	8
✓ 4	6	7	7	7	8	9
✗ 7	9	11	4	6	7	8
✗ 7	9	11	7	7	8	9

R.B < S.D

OR

S.D > R.B

o/p

RA	RB	RC	SD	SE	SF	SG
1	3	5	4	6	7	8
1	3	5	7	7	8	9
4	6	7	7	7	8	9

Equi-Join ($R \bowtie_\theta S$): It is a Subset of Conditional or

Similar to Conditional Join But Here 'only equality Condition' Applied.

$$\boxed{R \bowtie_\theta S = \pi_{\text{Attribute}}\left[\sigma_{\substack{\text{equality}\\\text{Condition}}}(R \times S)\right]}$$

Equi – Join

A special case of condition join where the condition c contains only equalities.

R_1 S_1

$R_1 \times S_1$

(Sid)	Sname	Rating	age	(Sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Equi – Join

$S1 \bowtie_{sid} R1$

A=D

Sid	Sname	Rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

- Result schema similar to cross – product, but only one copy of fields for which equality is specified.
- Natural join: Equijoin on all common fields.

$R(A\,B\,C)$ $\qquad$ $S(B\,C\,D)$

EQUI JOIN

$$R \bowtie_{R.B=S.B} S = \Pi_{\text{Attri}}\left(\sigma_{R.B=S.B}(R \times S)\right)$$

Natural Join

→ Equality Condition on ALL Common Attributes of R & S.

$$R \bowtie S = \Pi_{\substack{\text{Distinct}\\ \text{Attribute}}}\left[\sigma_{\substack{R.B=S.B\,\wedge\\ R.C=S.C}}(R \times S)\right]$$

Natural Join ($R \bowtie S$)

$$R \bowtie S = \underbrace{\pi_{\text{Distinct Attribute}}}_{\text{Step3}} \left[\underbrace{\sigma_{\text{equality Condition on All Common Attributes}}}_{\text{Step2}} \underbrace{(R \times S)}_{\text{Step1}} \right]$$

R(ABC) S(BCD) 3+3=6 Attribute

Distinct

$R \bowtie S =$

A	B	C	D

Join ($\bowtie$)

I. Natural join ($\bowtie$)

$$\boxed{R \bowtie S \equiv \pi_{\text{distinct attributes}}\left(\sigma_{\text{equality between ALL common attributes of R and S}}\,(R \times S)\right)}$$

Step 3 Step 2 Step 1

Example:

- $\boxed{T_1\,(ABC)}$ and $\boxed{T_2\,(BCDE)}$

$$\therefore\quad T_1 \bowtie T_2 = \pi_{ABCDE}\left(\sigma_{T_1.B = T_2.B \,\wedge\, T_1.C = T_2.C}(T_1 \times T_2)\right)$$

- $T_1\,(AB)$ and $T_2\,(CD)$

$$\therefore\quad T_1 \bowtie T_2 \equiv \boxed{T_1 \times T_2} = \pi_{ABCD}\,(T_1 \times T_2)$$

NOTE:

Natural join equal to cross-product if join condition is empty.

$\Downarrow$

If No Common Attribute then

$R \bowtie S = (R \times S)$

Join ($\bowtie$)

II. Conditional Join ($\bowtie_c$)

- $R \bowtie_c S \equiv \sigma_c (R \times S)$

R

A	B	C
1	2	4
3	2	6

S

B	C	D
2	4	8
2	7	4

R: 2 Tuple, 3 Attribute

S: 2 Tuple, 3 Attribute

$\Rightarrow R \times S = 2 \times 2 = 4$ Tuple

$3+3 = 6$ Attribute

Natural Join

$$R \bowtie S = \pi_{\text{Distinct Attribute}} \left[\sigma_{R.B = S.B \wedge R.C = S.C} (R \times S) \right]$$

Step3 Step2 Step1

Step1

$R \times S =$

R.A	R.B	R.C	S.B	S.C	S.D
1	2	4	2	4	8
1	2	4	2	7	8
3	2	6	2	4	8
3	2	6	2	7	8

Step2

R.A	R.B	R.C	S.B	S.C	S.D
1	2	4	2	4	8

Step3

$R \bowtie S =$

A	B	C	D
1	2	4	8

Natural Join ⋈

Inner Join

R

A	B	C
1	2	4
3	2	6

Dangling Tuple

S

B	C	D
2	4	8
2	7	4

Dangling Tuple

$R \bowtie S =$

A	B	C	D
1	2	4	8

$R \times S =$

R.A	R.B	R.C	S.B	S.C	S.D
1	2	4	2	4	8
1	2	4	2	7	4
3	2	6	2	4	8
3	2	6	2	7	4

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

$R \bowtie S =$

A	B	C	D
1	2	4	8

OUTER JOIN

1. Left outer JOIN [LOJ] $R ⟕ S$ OR $R \overset{\circ}{\bowtie}_L S$
2. Right outer Join [ROJ] $R ⟖ S$ OR $R \overset{\circ}{\bowtie}_R S$
3. FULL OUTER JOIN [OUTER JOIN] $R ⟗ S$ OR $R \overset{\circ}{\bowtie} S$

WHY

OUTER-JOIN ?

Soln Because in Inner Join some Tuple failed to Satisfying the JOIN Condition [Dangling Tuples]
So loss of Data (Not getting the Complete, Table Data)

OUTER JOIN

By Default

OUTER JOIN = Natural JOIN + Dangling Tuples

OUTER JOIN = Inner Join + Dangling Tuples.

Inner Join

1. Conditional Join
2. Equi Join
3. Natural Join

OUTER JOIN

① Left outer JOIN [LOJ] R ⟕ S OR $R \overset{\circ}{\bowtie}_L S$

② Right outer Join [ROJ] R ⟖ S OR $R \overset{\circ}{\bowtie}_R S$

③ FULL OUTER JOIN [OUTER JOIN] R ⟗ S OR $R \overset{\circ}{\bowtie} S$

① (Ex) Left outer JOIN $(R ⟕ S)$ OR $R \overset{\circ}{\bowtie}_L S$

$R ⟕ S$ = $R \bowtie S$ && Include the The Tuples from Left side Relation R those failed to Satisfy Join Condition.

OR

$R \overset{\circ}{\bowtie}_L S$

OR

$R ⟕ S$ = $\underline{R \bowtie S}$ + Include Dangling Tuple of Left Side Relation R.

OR

$R \overset{\circ}{\bowtie}_L S$

② CONDITIONAL

(LOJ) Left outer JOIN $(R ⟕ S)$ OR $R \overset{\circ}{\bowtie}_L S$.

$R ⟕_c S$ = $R \bowtie_c S$ && (+|Include) The Tuples from Left side Relation R those failed to Satisfy Join Condition.

OR

$R (\overset{\circ}{\bowtie}_c)_L S$.

$R ⟕_c S$ = $R \bowtie_c S$ + Include Dangling Tuple of Left Side Relation R.

OR

$R (\overset{\circ}{\bowtie}_c)_L S$

Join (⋈)

III. Outer Joins:

(a) LEFT OUTER JOIN

$R ⟕ S$: It produces

(Tuple

$(R \bowtie S) \cup$ {Records of R those are failed join condition with remaining attributes null}

(b) RIGHT OUTER JOIN (⟖)

$R ⟖ S$: It produces

$(R \bowtie S) \cup$ {Records of S those are failed join condition with remaining attributes null}

(C) $\boxed{\text{FULL OUTER JOIN (⟗)} \quad R ⟗ S = (R ⟕ S) \cup (R ⟖ S)}$

Natural Join ⋈

R

A	B	C
1	2	4
3	2	6

S

B	C	D
2	4	8
2	7	4

$R \times S =$

R.A	R.B	R.C	S.B	S.C	S.D
1	2	4	2	4	8
1	2	4	2	7	4
3	2	6	2	4	8
3	2	6	2	7	4

$$R \bowtie S = \pi_{ABCD}\left\{\sigma_{\substack{R.B = S.B \wedge \\ R.C = S.C}}(R \times S)\right\}$$

$R \bowtie S =$

A	**B**	**C**	**D**
1	2	4	8

OUTER JOIN

① left outer Join ($R ⟕ S$) OR $R \overset{\circ}{\bowtie}_{L} S$

Left Outer Join [⟕]

(R ⋈ S)

R ⟕ S = R ⋈ S ⊕ && Include Dangling tuple of Relation R

R

A	B	C
1	2	4
3	2	6

Tuple which failed to satisfy Join condition

S

B	C	D
2	4	8
2	7	4

Tuple which failed to satisfy JOIN Condition

(R ⋈ S) =

A	B	C	D
1	2	4	8

R ⟕ S =

A	B	C	D
1	2	4	8
3	2	6	Null

OR $\perp$ NULL

Left Outer Join [⟕]

$(R ⟕ S)$

R

A	B	C
1	2	4
3	2	6

Dangling Tuple

S

B	C	D
2	4	8
2	7	4

Dangling Tuple

$(R \bowtie S) =$

A	B	C	D
1	2	4	8

$R ⟕ S =$

A	B	C	D
1	2	4	8
3	2	6	Null

Right outer Join

$R ⟖ S = R \bowtie S$ && Include Dangling Tuple of S.

OR

$R \overset{\circ}{\bowtie}_R S \Rightarrow$

A	B	C	D
1	2	4	8
NULL	2	7	4

Right Outer Join [⋊]

$R \rtimes S =$

A	B	C	D
1	2	4	8
Null	2	7	4

FULL OUTER JOIN.

$$R ⟗ S = R ⟕ S \;\cup\; R ⟖ S$$

OR

$R \overset{\circ}{\bowtie} S$

OR

$R \overset{\circ}{\bowtie} S =$ $R \bowtie S$ && Include Dangling Tuple of Relation R & S.

Full Outer Join [⟗]

Full outer join = Left outer join Union Right outer join

$R ⟗ S = R ⟕ S \cup R ⟖ S$

R ⟕ S

A	B	C	D
1	2	4	8
3	2	6	Null

∪

R ⟖ S

A	B	C	D
1	2	4	8
Null	2	7	4

R ⟗ S =

OR

R ⟗ S

A	B	C	D
1	2	4	8
3	2	6	Null
Null	2	7	4

Left outer JOIN $(R ⟕ S)$ OR $R \overset{\circ}{\bowtie}_L S$.

SUMMARY

Natural JOIN	$R ⟕ S$ OR $R \overset{\circ}{\bowtie}_L S$	$= R \bowtie S$ && Include Dangling Tuple of Left Side Relation R
Conditional left outer JOIN.	$R ⟕_c S$ OR $R (\overset{\circ}{\bowtie}_c)_L S$	$= R \bowtie_c S$ && Include Dangling Tuple of R.

RIGHT OUTER JOIN : $(R ⟖ S)$ OR $R \overset{\circ}{\bowtie}_R S$

Natural ROJ

$R ⟖ S$ OR $R \overset{\circ}{\bowtie}_R S$ $= R \bowtie S$ && Include Tuples from Right Side Relation S Which failed to Satisfy Join Condition. OR (Dangling Tuple of S)

Conditional Right outer JOIN (ROJ)

$R ⟖_c S$ OR $R(\overset{\circ}{\bowtie}_c)_R S$ $= R \bowtie_c S$ + Include Dangle Tuples of Relation S.

FULL OUTER JOIN $(R ⟗ S)$ OR $R \overset{\circ}{⋈} S$

$$\boxed{R ⟗ S = R ⟕ S \cup R ⟖ S}$$

$$\boxed{R \overset{\circ}{⋈} S = R ⋈ S + \text{Dangling Tuple of } R \,\&\, S.}$$

Conditional Full OUTER JOIN.

$$R \overset{\circ}{⋈}_c S \quad = \quad \underset{\text{(Conditional JOIN)}}{R ⋈_c S} + \text{Include Dangling Tuples of Relation } R \,\&\, S.$$

OR

$R ⟗_c S$

Q.

Let R and S be two relations with the following schema

R(P, Q, R1, R2, R3)

S(P, Q, S1, S2)

Where {P, Q} is the key for both schemas. Which of the following queries are equivalent?

I. $\pi_P (R \bowtie S)$

II. $\pi_P(R) \bowtie \pi_P(S)$

III. $\pi_P(\pi_{P,Q} (R) \cap \pi_{P,Q}(S))$

IV. $\pi_P(\pi_{P,Q}(R) - (\pi_{P,Q} (R) - \pi_{P,Q} (S)))$

(A) Only I and II

(B) Only I and III

(C) Only I, II and III

(D) Only I, III and IV

Rename operator [ρ]

$\boxed{\text{STUDENT (Sid, Sname, Age)}}$

1. Table Rename.
2. ALL Attribute Rename.
3. Particular Attribute Rename
4. Table + Attribute Rename

Rename operator [ρ]

STUD. (Sid, Name, Age)

① $\rho(\text{Temp. STUD})$ →

Temp.

Sid	Sname	Age

Renaming Table

② $\rho_{(S, N, A)}\ \text{STUD}$ →

STUDENT

S	N	A

Renaming ALL Attribute

③ $\rho_{\substack{1 \to S \\ 2 \to N}}(\text{STUD})$ →

STUD.

S	N	Age

Renaming (Some Attribute Renaming) Particular Attribute.

Rename operator (ϱ) ρ

It is used to rename table name and attribute names for query processing.

Example:

(I) Stud (Sid, Sname, age)

ϱ(Temp, Stud) : $\boxed{\text{Temp (Sid, Sname, age)}}$

(II) $\varrho_{I, N, A}$ (Stud) : Stud (I, N, A)

All attributes renaming

(III) $\varrho_{\substack{sid \rightarrow I \\ age \rightarrow A}}$ (Stud) : Stud (I, Sname, A)

Some attribute renaming

Assignment operator.

$\underline{Temp_1} \longleftarrow \left[\begin{array}{c} \text{Any} \\ \text{Result} \end{array} \right]$

Division

- It is used to retrieve attribute value of R which has paired with every attribute value of other relation S.
- $\pi_{AB}(R)/\pi_{B}(S)$: It will retrieve values of attribute 'A' from R for which there must be pairing 'B' value for every 'B' of S.

Expansion of '/' by using basic operator

- Example: Retrieve sid's who enrolled every course.
- Result:

  $\pi_{sidcid}(\text{Enroll}) / \pi_{cid}(\text{Course})$

  Step 1: Sid's not enrolled every course of course relation.

  (Sid's enrolled proper subset of course)

  $\pi_{sid}((\pi_{sid}(\text{Enroll}) \times \pi_{cid}(\text{course})) - \pi_{sidcid}(\text{Enroll}))$
- Step 2:

  [sid's enrolled every course] = [sid's enrolled some course] – [sid's not enrolled every course]

$\therefore \quad \pi_{sidcid}(E) / \pi_{cid}(c) = \pi_{sid}(E) - \pi_{sid}((\pi_{sid}(E) \times \pi_{cid}(C) - \pi_{sidcid}(E))$

Division

Q. Retrieve all student who are Enrolled **Some course** or **Any course** or at least one course?

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

Course
Cid
C_1
C_2
C_3

Solution Π_{Sid} (Enrolled)

Division

Q. Retrieve all student who are Enrolled every course?

Solution

$$\boxed{\Pi_{Sid.Cid}\,(\text{Enrolled}) \,/\, \Pi_{Cid}\,(\text{Course})}$$

Find

2^{nd} attribute must be same.

Enrolled	
<u>Sid</u>	<u>Cid</u>
S_1	C_1
S_1	C_2
S_1	C_3
S_2	C_1
S_2	C_3
S_3	C_1

Course
Cid
C_1
C_2
C_3

Division

$$\Pi_{Sid}(\text{Enrolled}) - \Pi_{Sid}[\Pi_{Sid}(\text{Enrolled}) \times \Pi_{Cid}(\text{Course}) - \text{Enrolled}]$$

Division

$$\boxed{\Pi_{AB}(R) \,/\, \Pi_{B}(S) = \Pi_{A}(R) - \Pi_{A}\left[\Pi_{A}(R) \times \Pi_{B}(S) - R\,\right]}$$

Find

Connection

$$\Pi_{ABCD}(R) \,/\, \Pi_{CD}(S) \Rightarrow \Pi_{AB}(R) - \Pi_{AB}\left[\Pi_{AB}(R) \times \Pi_{CD}(S) - R\,\right]$$

Q. Consider the following three relations in a relational database:

Employee (eId, Name), Brand (bId, bName), Own(eId, bId)

Which of the following relational algebra expressions return the set of eIds who own all the brands? [GATE: 2022]

(A) $\pi_{eId}(\pi_{eId,\,bId}(Own / \pi_{bId}(Brand))$

(B) $\pi_{eId}(Own) - \pi_{eId}((\pi_{eId}(Own) \times \pi_{bId}(Brand)) - \pi_{eId,\,bId}(Own))$

(C) $\pi_{eId}(\pi_{eId,\,bId}(Own) / \pi_{bId}(Own))$

(D) $\pi_{eId}((\pi_{eId}(Own) \times \pi_{bId}(Own) / \pi_{bId}(Brand))$

Consider the two relation Suppliers and Parts are given below.

Suppliers

S_{no}	P_{no}
S_1	P_1
S_1	P_2
S_1	P_3
S_1	P_4
S_2	P_1
S_2	P_2
S_3	P_2
S_4	P_2
S_4	P_4

Parts

P_{no}
P_2
P_4

$\pi_{S_{no}P_{no}}$ (Suppliers) / $\pi_{P_{no}}$ (Parts)

The number of tuples are there in the result when the above relational algebra query executes is _____.

Consider the Database with relations:

S Supplier (Sid, Sname, Rating)

P Parts (Pid, Pname, Color)

S Catalog (Sid Pid, Cost)

Q. Find the Sid of Supplier whose Rating greater than 9?

Q.

Find the Pid of Red Color Parts?

Q. Retrieve Sid of Supplier whose cost is greater than 20,000?

Q. Retrieve Sid of Supplier who supplied some Red color parts?

Solution: $\Pi_{Sid}\left[\sigma_{\substack{P.Pid=C.Pid \\ P.Color=Red}} \wedge (Catalog \times Parts)\right]$

Note: Let an Attribute A belongs to R only then

$\sigma_{A = \text{'a'}}(R \bowtie S) = \sigma_{A = \text{'a'}}(R) \bowtie S \rightarrow$ More efficiency query

Note: Let an Attribute A belongs to R only and Attribute B belongs to S only then

$\sigma_{A = \text{'a'} \wedge B = \text{'b'}}(R \bowtie S) = \sigma_{A = \text{'a'}}(R) \bowtie \sigma_{B = \text{'b'}}(S)$

Q.

Consider the following relation schemas:

b-Schema = (b-name, b-city, assets)

a-Schema = (a-num, b-name, bal)

d-Schema = (c-name, a-number)

Let branch, account and depositor be respectively instances of the above schemas. Assume that account and depositor relations are much bigger than the branch relation.

Consider the following query:

$$\Pi_{\text{c-name}} \left(\sigma_{\text{b-city} = \text{"agra"} \wedge \text{ bal} < 0} (\text{branch} \bowtie \text{account} \bowtie \text{depositor})\right)$$

Q. Which one of the following queries is the most efficient version of the above query? [GATE-2007: 2 Marks]

(A) $\Pi_{c\text{-}name}\,(\sigma_{bal < 0}\,(\sigma_{b\text{-}city = \text{"Agra"}}\ \text{branch} \bowtie \text{account}) \bowtie \text{depositor})$

(B) $\Pi_{c\text{-}name}\,(\sigma_{b\text{-}city = \text{"Agra"}}\ \text{branch} \bowtie (\sigma_{bal < 0 =}\ \text{account}) \bowtie \text{depositor})$

(C) $\Pi_{c\text{-}name}\,(\sigma_{b\text{-}city = \text{"Agra"}}\ \text{branch} \bowtie \sigma_{b\text{-}city = \text{"Agra} \wedge bal < 0}\ \text{account} \bowtie \text{depositor})$

(D) $\Pi_{c\text{-}name}\,(\sigma_{b\text{-}city = \text{"Agra"}}\ \text{branch} \bowtie (\sigma_{b\text{-}city = \text{"Agra} \wedge bal < 0}\ \text{account} \bowtie \text{depositor}))$

Q.

Consider two relations $R_1(A, B)$ with the tuples (1, 5), (3, 7)and R_2 (A, C) = (1, 7)(4, 9)

(OUTER JOIN)

Assume that R(A, B, C) is the full natural outer join of R_1 and R_2. Consider the following tuples of the form (A, B, C); a = (1, 5, null), b = (1, null, 7), c = (3, null, 9), d = (4, 7, null), e = (1, 5, 7), f = (3, 7, null), g = (4, null, 9). Which one of the following statements is correct? [GATE-2015: 1 Mark]

A) R contains a, b, e, f, g, but not c, d

B) R contains all of a, b, c, d, e, f, g

C) R contains e, f, g, but not a, b

D) R contains e but not f, g

Q.

Consider the following relations given below:

R

A	B
6	6
7	6
8	8

S

C	D
6	7
8	9
8	10

$$\Pi_{AD}(R \times S) - P_{A \leftarrow B}(\Pi_{BD}(R \bowtie_{B=C} S))$$

Number of tuples return by the above query when it is executed on the above instance of relation R and S is ___

Summary

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{<\text{selection condition}>}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{<\text{attribute list}>}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{<\text{join condition}>} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{<\text{join condition}>} R_2$, OR $R_1 \bowtie_{(<\text{join condition 1}>),\ (<\text{join condition 1}>)} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{<\text{join condition}>} R_2$, OR $R_1 *_{(<\text{join attributes 1}>),\ (<\text{join attributes 2}>)} R_2$ OR $R_1 * R_2$

OPERATION	PURPOSE	NOTATION
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 and that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$	$R_1(Z) \div R_2(Y)$

NAT

Consider a database that has the relation schema CR(StudentName, CourseName). An instance of the schema CR is as given below:

The following query is made on the database.

$T1 \leftarrow \pi_{CourseName}(\sigma_{StudentName='SA'}(CR))$

$T2 \leftarrow CR \div T1;$

The number of rows in T2 is ________.

[GATE-2017-CS: 2M]

CR	
Student Name	Course Name
SA	CA
SA	CB
SA	CC
SB	CB
SB	CC
SC	CA
SC	CB
SC	CC
SD	CA
SD	CB

Student Name	Course Name
SD	CC
SD	CD
SE	CD
SE	CA
SE	CB
SF	CA
SF	CB
SF	CC

MCQ

The following relation records the age of 500 employees of a company, where empNo {indicating the employee number} is the key:

empAge(<u>empNo</u>, age)

Consider the following relational algebra expression:

$\Pi_{empNo}(\text{empAge} \bowtie_{(age>age1)} \rho_{empNo1,age1}(\text{empAge}))$

What does the above expression generate? [GATE-2020-CS: 1M]

(A) Employee numbers of only those employees whose age is the maximum

(B) Employee numbers of only those employees whose age is more than the age of exactly one other employee

(C) Employee numbers of all employees whose age is not the minimum

(D) Employee numbers of all employees whose age is the minimum

NAT

Consider the following relations P(X, Y, Z), Q(X, Y, T) and R(Y, V)

P		
X	Y	Z
X1	Y1	Z1
X1	Y1	Z2
X2	Y2	Z2
X2	Y4	Z4

Q		
X	Y	T
X2	Y1	2
X1	Y2	5
X1	Y1	6
X3	Y3	1

R	
Y	V
Y1	V1
Y3	V2
Y2	V3
Y2	V2

How many tuples will be returned by the following relational algebra query?

$[\Pi_X(\sigma_{(P.Y=R.Y \wedge R.V=V2)}(P\times R))-\Pi_X(\sigma_{(Q.Y=R.Y \wedge Q.T>2)}(Q\times R))]$;

[GATE-2019-CS: 2M]

MCQ

Suppose $R_1(\underline{A}, B)$ and $R_2(\underline{C}, D)$ are two relation schemes. Let r_1 and r_2 be the corresponding relation instances. B is a foreign key that refers to C in R_2. If data in r_1 and r_2 satisfy referential integrity constraints, which of the following is ALWAYS TRUE?

[GATE-2013-CS: 2M]

(A) $\Pi_B(r_1)-\Pi_C(r_2)=\phi$

(B) $\Pi_C(r_2)-\Pi_B(r_1)=\phi$

(C) $\Pi_B(r_1)=\Pi_C(r_2)$

(D) $\Pi_B(r_1)-\Pi_C(r_2) \neq \phi$

NAT

Consider the following table named Student in a relational database. The primary key of this table is rollNum.

Student

Roll Num	Name	Gender	Marks
1	Naman	M	62
2	Aliya	F	70
3	Aliya	F	80
4	James	M	82
5	Swati	F	65

The SQL query below is executed on this database.

```
SELECT *
FROM Student
WHERE gender = 'F' AND marks > 65;
```

The number of rows returned by the query is

[GATE-2023-CS: 2M]

MCQ

Consider the following relation A, B and C:

A		
ID	Name	Age
12	Arun	60
15	Shreya	24
99	Rohit	11

B		
ID	Name	Age
15	Shreya	24
25	Hari	40
98	Rohit	20
99	Rohit	11

C		
ID	Phone	Area
10	2200	02
99	2100	01

How many tuples does the result of the following relational algebra expression contain? Assume that the schema of $A \cup B$ is the same as that of A.

$(A \cup B) \bowtie_{A.Id > 40 \vee C.Id < 15} C$ [GATE-2012-CS: 2M]

A) 7

B) 4

C) 5

D) 9

THANK YOU!