COMPUTER SCIENCE

Database Management System

Query Language

Lecture_1

Vijay Agarwal sir

TOPICS TO BE COVERED

01 Relational Algebra

02 Operators

ER MODEL to RDBMS Conversion

Foreign key Concept

(a) CC with Enjoying

(b) CC

(c) C

(d) Doubt

- ✓ FD & Normalization
- ✓ Transaction & Concurrency Control
- ✓ ER MODEL & foreign key Concept
- → Query Language
- File Org & Indexing

Query language

Procedural Query language

Non Procedural Query language

eg Relational Algebra

eg SQL
TRC.

Query Language

Procedural Query Language

Relational Algebra:

Non Procedural Query Language

- SQL
- TRC

Procedural Query Language and Non-procedural Query Language

Procedural Query Language	Non-procedural Query Language
Formulation of how to access data from the database table and what data required to retrieve from DB tables. "Relational Algebra"	Formulation of what data retrieve from DB tables. "Relational Calculus" TRC ✓ DRC ✗ "SQL"

Relational Algebra. (R.A)

- Procedural Query Language.
- Result Always Distinct Value (R.A. By Default eliminate the Duplicate)

Eg

R

SName
A
B
A
C
D
B

$\pi_{sname}(R) \Rightarrow$

sname
A
B
C
D

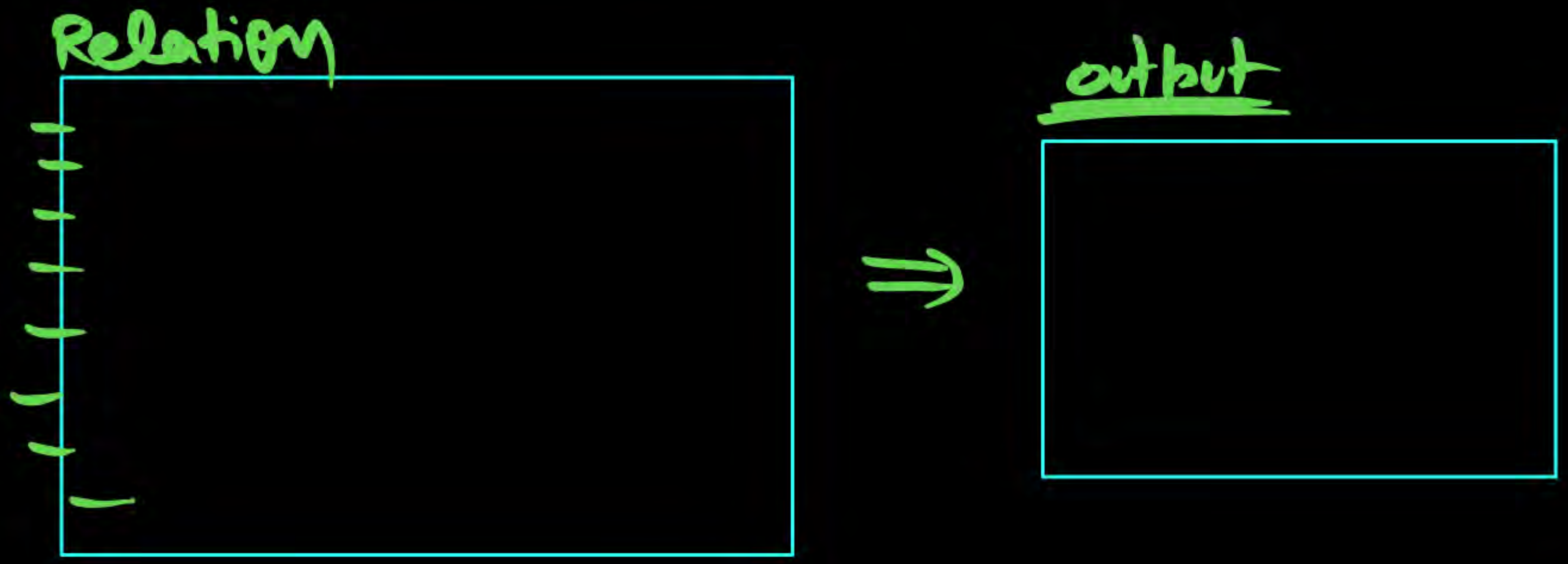
Note

The Basic Idea of Query language is, Query executed on a DB Table Tuple By Tuples, One Tuple at a time.

Relational Algebra

(Always generate distinct tuples)

Relational algebra refers to a procedural query language that takes relation instances as input and returns relation instances as output



Relational Algebra (R.A)

Operators

- Basic operator
- Derived operator.

Relational Algebra

Basic operators

1. π : Projection operator [π]
2. σ : Selection operator [σ]
3. × : Cross-product operator [x]
4. U : Union [U]
5. – : Set difference (Minus) [–]
6. ρ : Rename operator [ρ]

Relational Algebra

Derived operators

1. $\cap$: Intersection {using "_"} $[\cap]$
2. $\bowtie$: Join {using X, σ} $(\bowtie)$
3. / or $\div$: Division {using π, x, –} $(- \times /)$

$$\boxed{R \cap S \equiv R - (R - S)}$$

R : [1, 2, 3, 4, 5, 6]

S = [4, 5, 6, 7, 8]

$R \cap S = [4, 5, 6]$

R – S = [1, 2, 3]

R – (R – S) = [1, 2, 3, 4, 5, 6] – [1, 2, 3] = [4, 5, 6]

$$\boxed{R \cap S \equiv R - (R - S)}$$

BASIC OPERATORS.

① selection [σ] : It Select a Tuples Based on Specified Condition.

Syntax

$$\sigma_{Condition}(Relation)$$

eg

$$\sigma_{CGPA>9}(Student)$$

$$\sigma_{State=WB}(Employee)$$

② Projection [Π]: It Select Attribute or Attribute List from the Relation.

Syntax

$$\Pi_{\substack{Attribute \text{ or } \\ Attribute\ List}}(Relation)$$

eg

(i) $\Pi_{sname}(Student)$

(ii) $\Pi_{sname\ Branch\ CGPA}(STUDENT)$

(iii) $\Pi_{ABCD}(R)$

(iv) $\Pi_{A}(R)$

Projection

① $\Pi_{RollNo}(STUDENT)$

② $\Pi_{Sname}(STUDENT)$

③ $\Pi_{Branch}(STUDENT)$

STUDENT.

Roll No	Sname	Branch	CGPA
1	A	CS	9.1
2	B	IT	9.2
3	C	CS	9.3
4	A	IT	9.4
5	D	EC	9.6
6	C	CS	9.7

Sol^n 1

RollNo
1
2
3
4
5
6

Sol^n 2

Sname
A
B
C
D

Sol^n ③

Branch
CS
IT
EC

Selection

STUDENT.

Roll no	Sname	Branch	CGPA
1	A	CS	9.1
2	B	IT	9.2
3	C	CS	9.3
4	A	IT	9.4
5	D	EC	9.6
6	C	CS	9.7

① $\Pi_{RollNo}\left[\sigma_{CGPA > 9.2}(Student)\right]$

Sol^n 1

RollNO
3
4
5
6

② $\Pi_{Sname}\left[\sigma_{CGPA > 9.2}(Student)\right]$

Sol^n 2

Sname
C
A
D

BASIC OPERATORS.

① selection (σ) :

② Projection (π)

Relational Algebra

Basic operators

I. π : Projection

- $\pi_{\text{Attribute name}}$ (R): It is used to project required attribute from relation R.

→ Tuples | Rows.

- $\sigma_{\text{Condition(P)}}$ (R): It is used to select records from relation R, those satisfied the condition (P).

Example:

R

A	B	C
8	4	5
2	4	5
7	4	6
3	5	5

① $\pi_{B,C}(R)$:

Projection

B	C
4	5
4	6
5	5

② $\sigma_{A>5}(R)$:

A	B	C
8	4	5
7	4	6

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

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

Selection (σ)

σ : Tuple.

$\Pi \Rightarrow$ Attribute.

Q.1 $\sigma_{\text{rating} > 8}(S_2)$

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

Query language.

Today 8:30PM New Subject Starts.

OS By GOD OF CS

Dr. K. Sir

Selection (σ)

Ans.1 $\sigma_{\text{rating} > 8}(S_2)$

Sailors (R_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

Output:

Sid	Sname	Rating	age
28	yuppy	9	35.0
58	rusty	10	35.0

Q.2 $\pi_{sname,\ rating}(\sigma_{rating > 8}(S_2))$

Ans 2

Output:

Sname	Rating
yuppy	9
rusty	10

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

Projection(π)

Q.2 $\pi_{age}(S_2)$

Age
35.0
55.5

Name
C_1: yuppy
C_2 Rating > 8
C_3: Age ≥ 35

Sailors (R_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

$\sigma_{Name=yppy}(\sigma_{age \geq 35}(\sigma_{Rating>8}(S_2))) \equiv \sigma_{age \geq 35}(\sigma_{name=yppy}(\sigma_{Rating>8}(S_2)))$

(1) $\sigma_{c_1}(\sigma_{c_2}(\sigma_{c_3}(\sigma_{c_4}(R)))) \equiv \sigma_{c_4}(\sigma_{c_2}(\sigma_{c_3}(\sigma_{c_1}(R))))$

Because σ [selection] Condition is applied on Tuples.

$$\sigma_{c_1}(\sigma_{c_2}(S2))$$

c_1 : Rating > 8

c_2 : Age ≥ 35.

$$\left[\sigma_{Rating > 8}\left(\sigma_{Age \geq 35}\right)\right] \rightarrow \text{o/p.}$$

28	yubby	9	35.0
58	Rusty	10	35.0

$$\sigma_{c_2}(\sigma_{c_1}(S2))$$

$$\sigma_{Age \geq 35}\left[\sigma_{Rating > 8}(S2)\right] \xrightarrow{\text{o/p}}$$

28	yubby	9	35.0
58	Rusty	10	35.0

Q>2

$\sigma_{C_1}(\sigma_{C_2}(S_2))$

C_1 : Rating > 8

C_2 : Age > 38

$\left[\sigma_{Rating > 8}\left(\sigma_{Age \geq 38}\right)\right]$ → O/P → Empty

---

$\sigma_{C_2}(\sigma_{C_1}(S_2))$

$\left[\sigma_{age \geq 38}\left(\sigma_{Rating > 8}(S_2)\right)\right]$ → O/P → Empty

age > 38 **AND** Rating > 8

Projection(π)

Ans.2 $\pi_{age}(S_2)$

Q.3

$\pi_{Sname, Age}(S_2)$

o/p.

Sname	Age
yuppy	35
Lubber	55.5
guppy	35.0
rusty	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

Output:

age
35.0
55.5

Q. $\pi_{\text{sname, rating}}(S_2)$

Output:

Sname	Rating
yuppy	9
lubber	8
guppy	5
rusty	10

Important Point

Q.1 $\sigma_{C_1}(\sigma_{C_2}(\sigma_{C_3}(\sigma_{C_4}(R)))) \equiv \sigma_{C_4}(\sigma_{C_2}(\sigma_{C_1}(\sigma_{C_3}(R))))$

Soln 1 YES Both are Equivalent.

C_1 : Condition No 1

Q.2 $\underbrace{\pi_{A_1}}_{②}\left[\underbrace{\sigma_{C_1}(R)}_{①}\right] \equiv \sigma_{C_1}\left[\pi_{A_1}(R)\right]$

C_2 : Condition No 2.

$\pi_{\text{Attribute}}$ [Condition Applied on Tuple]

C_1: Rating > 8

A_1: Name.

$$\pi_{Pname}\left(\sigma_{Color=Red}(Parts)\right)$$

o/p

A
D
C

$$\sigma_{Color=Red}\left[\pi_{Pname}(Parts)\right]$$

Pid	Pname	Color	Cost
P1	A	Red	5K
P2	B	Green	6K
P3	C	yellow	7K
P4	A	Blue	8K
P5	D	Red	9K
P6	C	Red	9K

$$\pi_{sname}\left[\sigma_{CGPA>9}(STUDENT)\right]$$

Sname
B
E

Sname	Branch	CGPA
A	CS	9
B	IT	10
C	CS	8
D	IT	5
E	CS	10
F	IT	9

NOW

$$\sigma_{CGPA>9}\left[\pi_{sname}(STUDENT)\right]$$

$\sigma_{CGPA>9}$

sname
A
B
C
D
E
F

$\Rightarrow$ Not Possible to Apply.

2marks GATE

Is the L.H.S Can be Replaced by R.H.S or Not ?

Sol^n YES.

$$\sigma_{C_1}\left[\Pi_{A_1}(R)\right] \equiv \Pi_{A_1}\left[\sigma_{C_1}(R)\right]$$

first filter the Attribute then apply the Condition on same Attribute

$$\sigma_{Rating>8}\left[\underbrace{\Pi_{A_1}(R)}_{Rating}\right]$$

A_1 : Rating

C_1 : Rating > 8.

$$\underbrace{\Pi_{A_1}}_{Rating}\left[\sigma_{Rating>8}(R)\right]$$

Name Rating) Rating. Name

We know

$$\pi_{A_1}\left(\sigma_{C_1}(R)\right) \not\equiv \sigma_{C_1}\left(\pi_{A_1}(R)\right)$$

$\Rightarrow$ Both are Not equivalent

But

L.H.S can be Replaced by R.H.S

In these case

But L.H.S Can be Replaced by R.H.S

$$\sigma_{C_1}\left(\pi_{A_1}(R)\right) \equiv \pi_{A_1}\left(\sigma_{C_1}(R)\right)$$

In L.H.S Here Condition $[C_1]$ Applied on Only A_1 Attribute.

But Here R.H.S Can not be Replaced by L.H.S.

Set operators

1. UNION [U]
2. Intersect [∩]
3. MINUS (−]
   (Set Difference)

R & S be Union Compatible iff

① Arity (# of Attribute) of R & S must be same.

Arity: 3

R

∪

∩

S

Arity: 2

Not possible to Apply.

② Range of Attribute Must be Similar.

R

Sid	Sname

∪

∩

S

Branch	CGPA

Not possible to Apply

Registration Class.

R

Sid	Sname

$\cup$

$\cap$

$-$

Reg. JAVA Lang.

S

Sid	Sname

Now possible to Apply.

Set operator

U: Union operator

– : Except or minus

∩ : Intersection operator

- To apply set operations relations must be union compatible.
- R and S relations are union compatible
- If and only if-

(i) Arity of R equal to Arity of S and

(ii) Domain of attributes of R must be same as domain of attributes of s respectively.

Example

Example 1:

$\pi_{Sid\ Sname}(\dots\dots\dots) \cap \pi_{Sid}(\dots\dots\dots)$

{Arity not same so, set operation not allowed}

Example 2:

$\pi_{Sid\ Sname}(\dots\dots\dots) \cap \pi_{Sid\ Marks}(\dots\dots\dots)$

{Arity same but Sname domain is different from marks so, not allowed}

Arity same
Domain Similar } then allowed.

R

Sid	Sname

$\cup$

$\cap$

S

Sid	Gender

$R \cap S$

No

Example

$\pi_{Sid\ Sname}(\dots\dots\dots) \cap \pi_{\overset{Sid}{Stud\ ID},\ Stud\ name}(\dots\dots\dots)$

{Arity and domains are same so, allowed for set operation}

1. Set operation on relation:

R

A
2
2
2
3

2
3

S

B
2
2
4

2
4

1. $R \cup S: \{x / x \in R \lor x \in S\} \equiv$ $\boxed{\begin{matrix}2\\3\\4\end{matrix}}$

2. $R - S: \{x / x \in R \land x \notin S\} \equiv$ $\boxed{3}$

3. $R \cap S: \{x / x \in R \land x \in S\} \equiv$ $\boxed{2}$

Example

$\pi_{\underline{Sid\ Sname}}(\dots\dots\dots) \cap \pi_{\overset{Sid}{Stud\ ID},\ \underline{Stud\ name}}(\dots\dots\dots)$

{Arity and domains are same so, allowed for set operation}

1. Set operation on relation:

R

A
2
2
2
3

2
3

S

B
2
2
4

2
4

R − S

$\begin{matrix}2\\3\end{matrix} - \begin{matrix}2\\4\end{matrix} = 3$

$\boxed{\begin{matrix}2\\4\end{matrix}} \cap \boxed{\begin{matrix}2\\3\end{matrix}} = 2$

1 $R \cup S: \{x / x \in R \vee x \in S\} \equiv$

A
2
3
4

$\vee$: OR

2 $R - S: \{x / x \in R \wedge x \notin S\} \equiv$

A
3

AND

$R \cap S: \{x / x \in R \wedge x \in S\} \equiv$

A
2

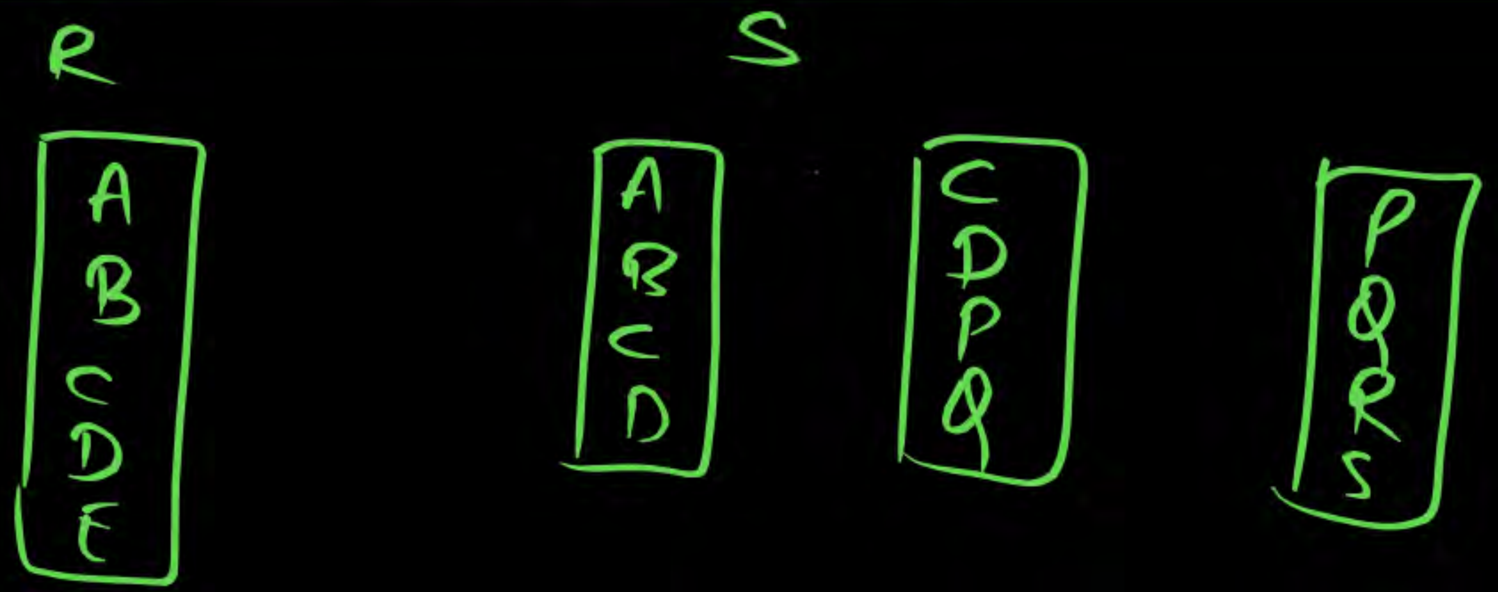
Assume Relation R & Relation S consist M & N Tuple Respectively

(1) Range of tuples in $R \cup S = \overset{\text{Minimum}}{\max(M, N)}$ to $\overset{\text{Maximum}}{M + N}$

(2) Range of tuples in $R \cap S = \phi$ to $\min(M, N)$

(3) Range of tuples in $R - S = \phi$ to M

(4) Range of tuples in $S - R = \phi$ to N



Union Operation

- Notation: $r \cup s$
- Defined as :

$$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$$

- For $r \cup s$ to be valid.
  1. r, s must have the same arity (same number of attributes)
  2. The attribute domains must be compatible (example: 2nd column or r deals with the same type of values as does the 2nd column of s)

Example:

To find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both.

$$\pi_{course\_id}(\sigma_{semester = \text{"Fall"} \wedge year = 2009}(section)) \cup$$
$$\pi_{course\_id}(\sigma_{semester = \text{"Spring"} \wedge year = 2010}(section))$$

Set Difference Operation

- Notation: r – s
- Defined as :

$$r - s = \{t \mid t \in r \text{ and } t \notin s\}$$

- Set differences must be taken between compatible relations.
  - r and s must have the same arity
  - attribute domains of r and s must be compatible

Example:

Example: to find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester

$$\pi_{course\_id}(\sigma_{semester = \text{"Fall"} \wedge\, year = 2009}\ (section)) -$$
$$\pi_{course\_id}(\sigma_{semester = \text{"Spring"} \wedge\, year = 2010}\ (section))$$

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

(1) Union $S1 \cup S2$

(1) Union

Sid	Sname	Rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$S1 \cup S2$

(2) Set Difference

Sid	Sname	Rating	age
22	dustin	7	45.0

S1 – S2

(3) Intersection

Sid	Sname	Rating	age
31	lubber	8	55.5
58	rusty	10	35.0

$S1 \cap S2$

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

Join ($\bowtie$)

I. Natural join ($\bowtie$)

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

Example:

- T_1 (ABC) and T_2 (BCDE)

$$\therefore \quad T_1 \bowtie T_2 = \pi_{ABCDE}\left(\begin{matrix}\sigma_{T_1 \cdot B} = T_{2 \cdot B}(T_1 \times T_2) \\ \cap T_1 \cdot C = T_2 \cdot C\end{matrix}\right)$$

- T_1 (AB) and T_2 (CD)

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

NOTE:

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

Join ($\bowtie$)

II. Conditional Join ($\bowtie_c$)

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

Join (⋈)

III. Outer Joins:

(a) LEFT OUTER JOIN

R ⟕ S : It produces

(R ⋈ S) ∪ {Records of R those are failed join condition with remaining attributes null}

(b) RIGHT OUTER JOIN (⟖)

R ⟖ S : It produces

(R ⋈ S) ∪ {Records of S those are failed join condition with remaining attributes null}

(C) FULL OUTER JOIN (⟗)

R ⟗ S = (R ⟕ S) ∪ (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

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

$R \bowtie S =$

A	B	C	D
1	2	4	8

Left Outer Join [⟕]

$(R ⟕ S)$

R

A	B	C
1	2	4
3	2	6

S

B	C	D
2	4	8
2	7	4

$(R ⟕ S) =$

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 [⟖]

$R ⟖ S =$

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

Full Outer Join [⟗]

Full outer join = Left outer join Union Right outer join

$R ⟗ S = R ⟕ S \cup R ⟖ S$

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

$\cup$

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

$R ⟗ S =$

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

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)

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

Example:

(I) Stud (Sid, Sname, age)

ϱ(Temp, Stud) : 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

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?

Solution Π_{Sid} (Enrolled)

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_1
C_3

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_1
C_3

Division

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

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 _____.

Any Doubt ?

THANK YOU!