COMPUTER SCIENCE

Database Management System

Query Language

Lecture_1

Vijay Agarwal sir

TOPICS TO BE COVERED

01 Relational Algebra

02 Operators

1. FD & Normalization
2. Transaction & Concurrency Control
3. ER Model & Foreign Key Concept

Remaining GATE Question.

4. Query Language. - 20 to 30 Q.
5. File org & Indexing - 10 to 15 Q

50 Q

92% GATE Question

Query language

(Formal)
Procedural Query Language
[eg Relation Algebra]

(Informal)
Non Procedural Query Language
[eg SQL
TRC]

Query Language

Procedural Query Language

- WHAT to Reterive From DB
- HOW to Reterive from DB

Relational Algebra:

Non Procedural Query Language

WHAT to Reterive from DB.

- SQL (Structured Query Language)
- TRC (Tuple Relational Calculus)

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" DRC TRC ✓ "SQL"

Relational Algebra

- Procedural Query language
- By Default eliminate Duplicate value.

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

Note

Basic Idea of Query Language is Query executed on DB Table Tuple by Tuple (Row by Row) One Tuple at a time.

Relation Algebra.

R: [1, 2, 3, 4, 5, 6]
S: [4, 5, 6, 7, 8]

R − S = [1, 2, 3]

$\boxed{R-(R-S) = [4, 5, 6] \equiv R \cap S}$

Basic operator

1. Selection [σ]
2. Projection [π]
3. CROSS Product [X]
4. UNION [U]
5. Minus / set Difference [−]
6. Rename [ρ]

Derived operator

1. JOIN ($\bowtie$)
2. DIVISION [/]
3. Intersection [$\cap$]

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

Relational Algebra

Basic operators

1. π : Projection operator
2. σ : Selection operator
3. × : Cross-product operator/Cartesian Product
4. U : Union
5. – : Set difference
6. ϱ : Rename operator

Relational Algebra

Derived operators

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

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

Projection [Π]

(Column)

It Select [Project] Attribute or Attribute List from the Relation

Syntax

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

eg

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

$\Pi_{sname\,Contact}(\text{STUDENT})$

$\Pi_A(R)$

$\Pi_{ABCD}(R)$

Selection [σ]

(Row) / Record

It Select the Tuple / Record from the Relation Based on Specified Condition

Syntax

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

eg $\sigma_{CGPA>9}(\text{STUDENT})$

eg $\sigma_{State='MP'}(\text{STUDENT})$

eg $\sigma_{Branch='CS'}(\text{Student})$

Q W.A.Q [Write an Query] to Reteive Name of Student whose CGPA > 9.

STUDENT

Rollno	Name	Branch	Gender	Contact	CGPA
1	Ajay	CS.	M.	–	8
2	Ankit	IT	M.		9
3	Biswjeet	CS.	M	–	9.1
4	Sohini	IT	F	–	9.2
5	Pankaj	CS.	M.	–	8.5
6	Gagan	IT	M.	–	9.4

Solⁿ

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

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

O/p. →

Name
Biswjeet
Sohini
Gagan

Q W.A.Q [Write an Query] to Reteive Name & Branch of Student whose CGPA > 9.

STUDENT

Rollno	Name	Branch	Gender	Contact	CGPA
1	Ajay	CS.	M.	–	8
2	Ankit	IT	M.		9
3	Biswjeet	CS.	M	–	9.1
4	Sohini	IT	F	–	9.2
5	Pankaj	CS.	M.	–	8.5
6	Gagan	IT	M.	–	9.4

Sol^n

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

$$\pi_{\text{name Branch}}\left[\sigma_{CGPA>9}(\text{Student})\right]$$

O/P. →

Name	Branch
Biswjeet	CS
Sohini	IT
Gagan	IT

Same o/p

$$\left[\sigma_{c_3}\left(\sigma_{c_2}\left(\sigma_{c_1}(R)\right)\right)\right] \equiv \sigma_{c_1}\left(\sigma_{c_3}\left(\sigma_{c_2}(R)\right)\right)$$

Q. CGPA > 9 . Gender = M . Branch = CS

c_1 c_2 c_3

o/p → $\boxed{\text{Biswjeet}}$

$\sigma_{c_1 \wedge c_2 \wedge c_3}$

$\sigma_{CGPA > 9 \wedge Gender = 'M' \wedge Branch = 'CS'}$

Relational Algebra

Basic operators

I. π : Projection

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

Example:

GIVEN Table

R

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

①

$\pi_{B,C}(R)$:

O/P

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)

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

Sailors (S_2)

<u>Sid</u>	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)

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

Output:

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

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

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 $\pi_{age}(S_2)$

Sailors (S_2)

<u>Sid</u>	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)

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

Output:

age
35.0
55.5

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

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

Output:

Sname	Rating
yuppy	9
lubber	8
guppy	5
rusty	10

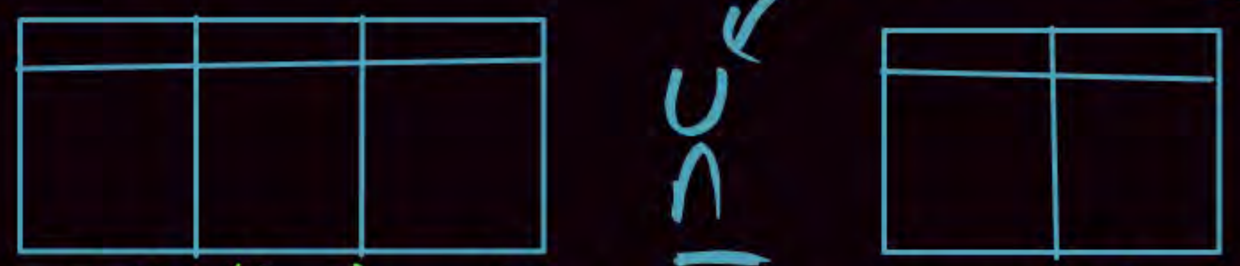
Set operators : [∪, ∩, −]

(∩)

Union . Intersection , Minus(−)

R & S be the Two Relation . is Union Combitable iff

① Number of Attribute [Degree/ Arity] Should be same .



Not possible to Apply .

② Range of Attribute Must be Similar .

RollNo	Name

∪ ∩ −

Branch	Contact

Not possible .

Now Apply

Registration JAVA	
Sname	Contact

$\cup$

$\cap$

$-$

Registration .Net	
Sname	Contact

$\left.\begin{array}{l}\cup\\ \cap\\ -\end{array}\right\}$ Now Apply.

Arity (Degree) : Same

Range : similar.

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

(Range)

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

Now Apply

Registration
JAVA

Sname	Contact

$\cup$

$\cap$

$-$

Registration
.Net

Sname	Contact

$\cup$, $\cap$, $-$ } Now Apply.

Arity (Degree) : Same

Range (Domain) : similar.

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}

Example

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

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

1. Set operation on relation:

R

A
2
2
2
3

S

A
2
2
4

2
3

2
4

$S - R : \{x \mid x \in S \wedge x \notin R\}$

4

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

A
2
3
4

OR

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

A
3

AND

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

A
2

R

2
2
2
3

2
3

S

2
2
4

2
4

1. $R \cup S$: 2, 3, 4 $\{x \mid x \in R \lor x \in S\}$
2. $R \cap S$: 2 $\{x \mid x \in R \land x \in S\}$
3. $R - S$: 3 $\{x \mid x \in R \land x \notin S\}$
4. $S - R$: 4 $\{x \mid x \in S \land x \notin R\}$

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

Minimum Maximum

(1) Range of tuples in $R \cup S = \max(M, N)$ to $M + N$

(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

R → 5 Tuple

C lang

A
B
C
D
E

S → 4 Tuple

JAVA lang

(i)		(ii)		(iii)
A	OR	D	OR	P
B		E		Q
C		P		R
D		Q		S

		Minimum		Maximum
1	$R \cup S$	$\max(M, N)$	to	$M+N$
2	$R \cap S$	ϕ	to	$\min(M, N)$
3	$R - S$	ϕ	to	M
4	$S - R$	ϕ	to	N

R (5 Tuple) S (4 Tuple)

	min		max
$R \cup S$:	5	7	$5+4$
	$\max(M, N)$	to	$M+N$
$R \cap S$	ϕ	4	$\min(M, N)$

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

$S_1 - S_2 =$ 22 Dustin

$S_2 - S_1 =$ 28 yuppy 9 35
44 Guppy 5 35

$S_1 \cap S_2 =$ 31
58

(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 × 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 × 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)

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

- T_1 (AB) and T_2 (CD)

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

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 ($\bowtie$)

III. Outer Joins:

(a) LEFT OUTER JOIN

$R ⟕ S$: It produces

$(R \bowtie S) \cup$ {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) FULL OUTER JOIN (⟗)

$R ⟗ S = (R ⟕ S) \cup (R ⟖ S)$

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

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 [$\rtimes$]

$R \rtimes S =$

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

All attributes renaming

(III) $\rho_{\substack{Sid \rightarrow I \\ age \rightarrow A}}$ (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?

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

Solution Π_{Sid} (Enrolled)

Division

Q. Retrieve all student who are Enrolled every course?

Solution

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

Find

2^{nd} attribute must be same.

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

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

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 _____.

THANK YOU!