COMPUTER SCIENCE

Database Management System

Query Language

Lecture_5

Vijay Agarwal sir

01 Queries

02 SQL Clauses

Relational Algebra

Basic operator

- Selection [σ]
- Projection [Π]
- CROSS Product [X]
- Union [$\cup$]
- EXCEPT / minus / set Difference [$-$]
- Rename [ρ]

Derived operators

- Intersection [$\cap$]
- JOIN & its type [$\bowtie$]
- Division [/ or $\div$]

How to Write Queries in Relational Algebra ?

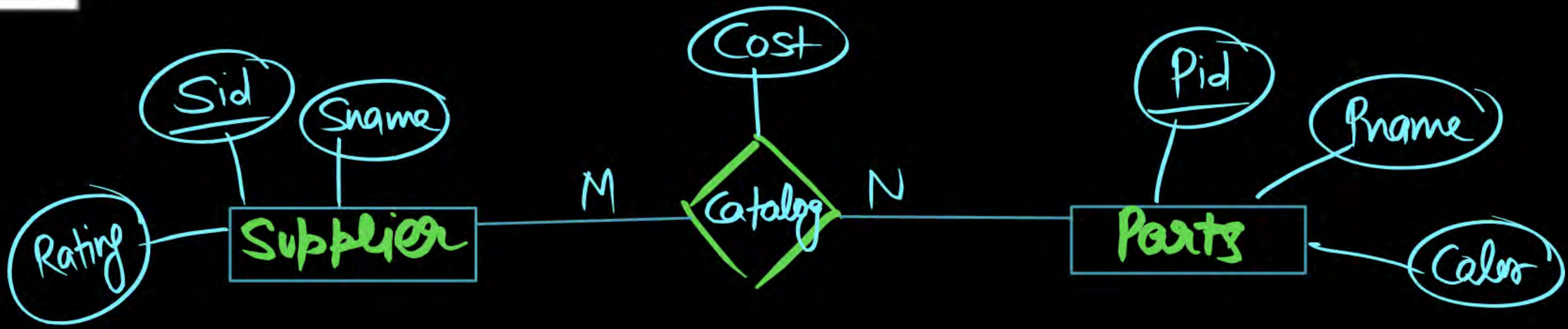
Consider the Database with relations:

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?



Now Start to
write Queries.
& optimize [efficient] Queries.

Consider the Database with relations:

S	Supplier (Sid, Sname, Rating)
P	Parts (Pid, Pname, Color)
S	Catalog (Sid, Pid, Cost)

Q.1 Find the Sid of Supplier whose Rating greater than 9?

Soln

$$\Pi_{Sid}\left[\sigma_{Rating>9}(Subbliier)\right]$$

Q.2 Find the Pid of Red Color Parts?

$$\Pi_{Pid}\left[\sigma_{Color='Red'}(Parts)\right]$$

Q.3

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

$$\Pi_{Sid}\left[\sigma_{Cost>20,000}(Catalog)\right]$$

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

Solution:

2 Table Required.

Parts & Catalog Table.

But WHY THESE TWO Table ?

Consider the Database with relations:

S	Supplier (Sid, Sname, Rating)
P	Parts (Pid, Pname, Color)
S	Catalog (Sid Pid, Cost)

Q.1 WHY Supplier Catalog X

Q.2 WHY Supplier, Parts X

Q.3 WHY Parts, Catalog ✓

X Sailyt If Supplier, Catalog.

Condition Applied on Color Attribute

↳ Present in Parts

Consider the Database with relations:

S	Supplier (Sid, Sname, Rating)
P	Parts (Pid, Pname, Color)
S	Catalog (Sid Pid, Cost)

Q.1 WHY Supplier Catalog X

Q.2 WHY Supplier, Parts X

Q.3 WHY Parts, Catalog ✓

X Sailor

If Supplier & Parts then

No Common Attribute so Not find.

Sid of Red Color Parts.

Consider the Database with relations:

S	Supplier (Sid, Sname, Rating)
P	Parts (Pid, Pname, Color)
S	Catalog (Sid Pid, Cost)

Q.1 WHY Supplier Catalog X

Q.2 WHY Supplier, Parts X

Q.3 WHY Parts, Catalog ✓

Soln 3 WHY Parts, Catalog ✓

In Parts table we find Pid 'Color = Red' then that Pid Match with Catalog table Pid then Project(Find) Sids.

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

Solution:

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

Consider the Database with relations:

S	Supplier (Sid, Sname, Rating)
P	Parts (Pid, Pname, Color)
S	Catalog (Sid Pid, Cost)

Consider the Database with relations:

S	Supplier (Sid, Sname, Rating)
P	Parts (Pid, Pname, Color)
S	Catalog (Sid Pid, Cost)

Q.

Retrieve Sname of Supplier who supplied some Red color parts?

Solution:

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

$$\sigma_{C_1}(\sigma_{C_2}(\sigma_{C_3}(R))) = \sigma_{C_1 \wedge C_2 \wedge C_3}(R) = \sigma_{C_3}(\sigma_{C_1}(\sigma_{C_2}(R)))$$

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

Solution:

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

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

Solution:

Query I :

4

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

400 × 500 = 2 Lakh Tuple

Query II

4

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

11 × 500 = 5500 Tuple

Query III

4

$\Pi_{Sid}\left[\sigma_{Color=Red}\left[Parts \bowtie Catalog\right]\right]$

200 Tuple

Color = Red

Query IV

4

Most efficient

$\Pi_{Sid}\left[\sigma_{Color=Red}(Parts) \bowtie Catalog\right]$

11 ⋈

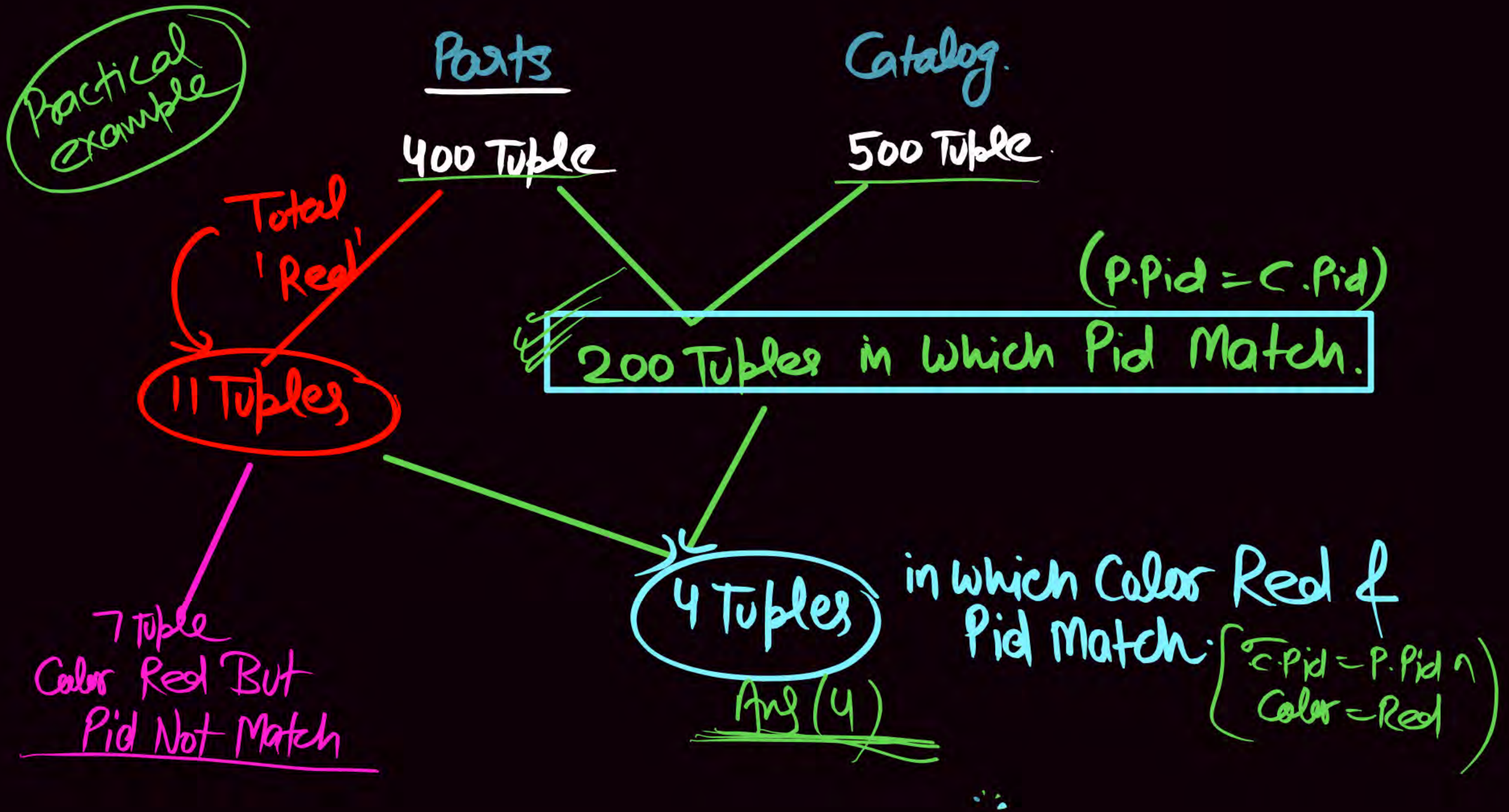
ALL Four Queries give the Same Answer.

But

My Question is Which One is the

Best [Most efficient | optimize] Query

& WHY ?


Practical example
Parts
400 Tuple
Catalog.
500 Tuple
Total 'Red'
11 Tuples
(P.Pid = C.Pid)
200 Tuples in which Pid Match.
7 Tuple Color Red But Pid Not Match
4 Tuples
Ans (4)
in which Color Red & Pid Match.
C.Pid = P.Pid ∧ Color = Red

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

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

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

$\sigma_{A='a' \wedge B='b'}(R \bowtie S) = \sigma_{A='a'}(R) \bowtie \sigma_{B='b'}(S)$ → Most efficient Query.

R (State='WB') S (CGPA > 8)

Engg. Student

AICTE

[25 State
6 UT]

10 Lakh Student

[R] STUDENT
Personal Info

[S] STUDENT
Academic Performance

State = WB

CGPA > 8

$\Pi_{Name}\left(\sigma_{State='WB' \wedge CGPA>8}(R \bowtie S)\right)$

25 State & 6 UT
+ CGPA [0-8, 9-10]

Most-Most efficient Query.

$\Pi_{Name}\left(\sigma_{State='WB'}(R) \bowtie \sigma_{CGPA>8}(S)\right)$

Exclude 24 State & 6 UT

Only WB

Exclude Student CGPA 0 to 8

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:

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

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

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

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

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

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

Q.

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

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

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 $R_1 \bowtie_{\text{(<join condition 1 >), (<join condition 1 >)}} R_2$
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 $R_1 *_{\text{(<join attributes 1 >), (<join attributes 2 >)}} R_2$ 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 $Z = X \cup Y$	$R_1(Z) \div R_2(Y)$

SQL: Non Procedural / Declarative Query language

SQL[Structured Query Language]

- DDL(Data Definition Language): Modification allowed at schema (Definition) level & Constraint
  - CREATE
  - ALTER
  - DROP TABLE
- DML(Data Manipulation Language): Modification allowed at data level
  - INSERT
  - UPDATE
  - DELETE
- DCL(Data Control Language): Control Transactional Operation
  - COMMIT
  - ABORT
- DQL(Data Query Language): Used to Retrieve the Data from DB
  - SELECT
  - FROM
  - WHERE

SQL

optional

RA

SQL

SELECT	[DISTINCT] $A_1\ A_2\ A_3 \ldots A_n$	≡	Projection [Π]
FROM	$R_1\ R_2\ R_3 \ldots R_n$	≡	CROSS Product [X]
WHERE	Condition [Predicate]	≡	Selection [σ]

Relational Algebra :

$$\Pi_{A_1 A_2 A_3 \ldots A_n}\left[\sigma_{Condition}(R_1 \times R_2 \times R_3 \ldots \times R_n)\right]$$

SQL

optional clause.

R.A

SELECT [DISTINCT] A_1 A_2 A_3 A_n... ≡ Projection (π)

FROM R_1 R_2 R_3 ..... R_m ≡ CROSS Product (x)

WHERE Condition ≡ Selection [σ]

R.A: $\pi_{A1A2A3..An}[\sigma_{Condition}(R_1 \times R_2 \times R_3 \ldots \times R_m)]$

Difference between

Q.1 SQL & R.A ?

Q.2 SQL Case Sensitive ? 'key word'

Q.3 Difference b/w Select in SQL & π (Projection) in R.A?

Soln 1 R.A is Procedural Query Language.

SQL is Non Procedural, Declarative Query Language.

Difference between.

Q.1 SQL & R.A ?

Q.2 SQL Case Sensitive ? 'keyWood'

Q.3 Difference b/w Select in SQL & π (Projection) in R.A?

Soln NO, SQL is NoT Case Sensitive.

SELECT, Select, SelECT, From, FROM, froM etc.

Difference between.

Q.1 SQL & R.A ?

Q.2 SQL Case Sensitive ?

Q.3 Difference b/w Select in SQL & π (Projection) in R.A?

Soln SELECT in SQL retain Duplicates.

π[Projection] in RA By default eliminate Duplicates.

Q.4 How Many Number of SQL clause is Mandatory?

a) 1

b) 2

c) 3

d) 4 or more than 4

SELECT

From

Π_{Sname} (Student)

e) Select Sname
From Student

∴

Select: Not going to eliminate Duplicate Value.

1) SELECT AB
FROM R

Output →

A	B
1	2
1	2
2	4

R(A B C)

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

2) $\pi_{AB}(R)$ →

A	B
1	2
2	4

3) SELECT [DISTINCT] AB
FROM R

Output →

A	B
1	2
2	4

$\equiv \pi_{AB}(R)$

SQL Clauses

SELECT [DISTINCT] $A_1\ A_2\ A_3 \ldots A_n$

FROM $R_1\ R_2\ R_3 \ldots R_m$

[WHERE P]

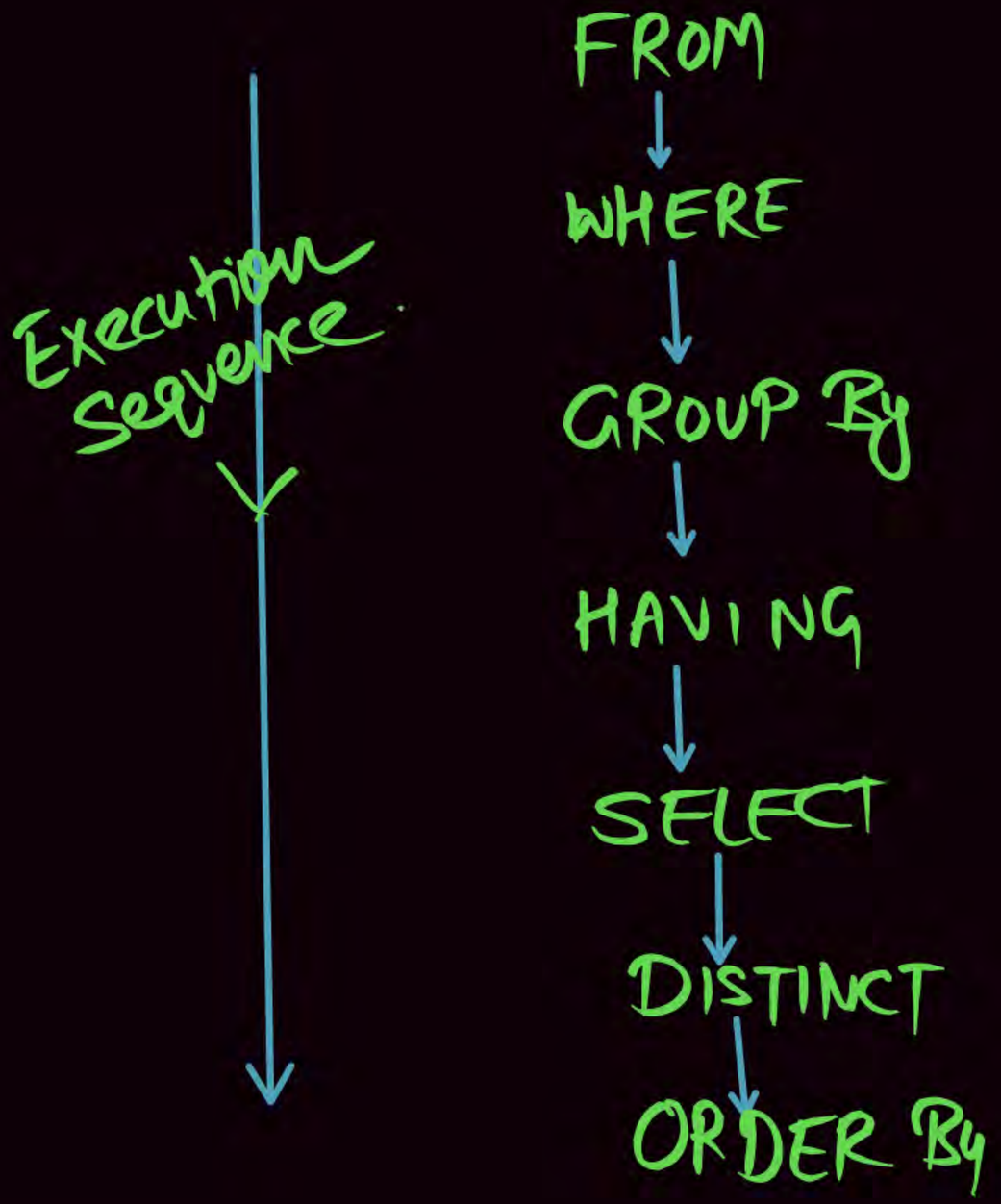
[GROUP By Attribute [[HAVING Condition]]]

[ORDER By Attribute [[DESC]]]

[ ] ← Optional clause

1. FROM
2. WHERE
3. GROUP By
4. HAVING
5. SELECT
6. DISTINCT
7. ORDER By.

ASEC


Execution Sequence
FROM
WHERE
GROUP BY
HAVING
SELECT
DISTINCT
ORDER BY

Query Execution

(1) FROM Clause: It is the first executable Clause. It just simply Relation (or) CROSS Product of Two or more Relation

(2) WHERE Clause: It is the second executable clause. It selects the tuple based on specified condition.

(3) GROUP By Clause: It is the third executable clause if used in the query. It groups the table based on the specified attributes.

Sid	(Branch)	Marks
S_1	CS	90
S_2	IT	70
S_3	CS	70
S_4	EC	56
S_5	CS	NULL

GROUP By (Branch)

(Branch)

Sid	(Branch)	Marks
S_1	CS	90
S_3	CS	70
S_5	CS	NULL
S_2	IT	70
S_4	EC	56

Aggregation operator ⇒ Always Discard Null Value

1) COUNT ([DISTINCT] Attribute)
2) SUM ([DISTINCT] Attribute)
3) AVG ([Distinct] Attribute)
4) MIN (Attribute)
5) MAX (Attribute)

1) Count(marks) =
2) Count (*) =
3) Count ([DISTINCT]marks) =
4) SUM(marks) =
5) SUM([Distinct]marks) =
6) AVG(marks) =
7) AVG([Distinct]marks) =

Aggregation operator ⇒ Always Discard Null Value

1) COUNT ([DISTINCT] Attribute)
2) SUM ([DISTINCT] Attribute)
3) AVG ([Distinct] Attribute)
4) MIN (Attribute)
5) MAX (Attribute)

1) Count(marks) = 4
2) Count (*) = 5
3) Count ([DISTINCT]marks) = 3
4) SUM(marks) = 286
5) SUM([Distinct]marks) = 216
6) $\text{AVG(marks)} = \frac{286}{4}$
7) $\text{AVG([Distinct]marks)} = \frac{216}{3}$

$$\frac{\text{SUM[DISTINCT]marks}}{\text{COUNT[DISTINCT]marks}} \Rightarrow \frac{216}{3}$$

HAVING: Fourth executable clause (if used in query).

It is used to select the group which satisfy the condition

(condition is for each group).

STUDENT

Sid	Branch	Marks
S_1	CS	60
S_2	IT	70
S_3	CS	90
S_4	IT	60
S_5	EC	55
S_6	EC	NULL

(GROUP By Branch)

Sid	Branch	Marks
S_1	CS	60
S_3	CS	90
S_2	IT	70
S_4	IT	60
S_5	EC	55
S_6	EC	NULL

```
Select * FROM STUDENT
GROUP By (Branch)
HAVING AVG(Marks) > 61
```

Sid	Branch	Marks
S_1	CS	60
S_3	CS	90
S_2	IT	70
S_4	IT	60

Q.1 Select min(marks)

FROM Student

Sid	Branch	Marks
S_1	CS	60
S_2	IT	70
S_3	CS	90
S_4	IT	60
S_5	EC	55
S_6	EC	NULL

ANSWER: 55

Q.2

Select min(marks)

FROM Student

WHERE Branch = 'CS'

Sid	Branch	Marks
S_1	CS	60
S_2	IT	70
S_3	CS	90
S_4	IT	60
S_5	EC	55
S_6	EC	NULL

ANSWER: 60

Q.3

Select min(marks)

FROM Student

GROUP By (Branch)

Sid	Branch	Marks
S_1	CS	60
S_2	IT	70
S_3	CS	90
S_4	IT	60
S_5	EC	55
S_6	EC	NULL

Q.3

Select min(marks)

FROM Student

GROUP By (Branch)

60
60
55

Sid	Branch	Marks
S_1	CS	60
S_2	IT	70
S_3	CS	90
S_4	IT	60
S_5	EC	55
S_6	EC	NULL

Q.4

Select min(marks), Branch

FROM Student

Q.4 Select min(marks), Branch

FROM Student

CS
IT
CS
IT
EC
EC

➡ Such Syntax is not allowed in SQL

NOTE:

When aggregate operator & other Attribute used in select clause is Allowed only of other attribute must be in Group of Clause.

```
Select min (marks) Branch
FROM Student
GROUP By (Branch)
```

Q. Select min(A) B
FROM R
Group By (C) ×
Group by (B) ✓

OTHER Set Operator

Followed by R.A ↓	Not followed By R.A ↓

1) UNION/ UNION ALL

2) INTERSECT/ INTERSECT ALL

3) MINUS / MINUS ALL

R
1
1
1
2
2
4
6
6

S
1
1
2
4
5
5

1) R UNION S

Result Distinct tuple

1
2
4
6
5

2) R UNION ALL S

Result all values

1
1
1
2
2
4
6
6
1
1
2
4
5
5

3) R INTERSECT S

Distinct Common tuples from R & S

1
2
4

4) R INTERSECT ALL S

How many maximum number of times Common in both R & S

1
1
2
4

4) R MINUS S

Distinct tuples from R
those are not there in S

6

5) R MINUS ALL S

Duplicates - # Duplicates
in R in S

1
2
6
6

Q. Retrieve Sid & Marks of the Student who secured Highest Marks?

Note: Aggregate friction can not be in lower clause

Query I:
```
Select Sid max(Marks)
From Student
```

Query II:
```
Select Sid, max(marks)
From Student
Group By (Sid).
```

Query III:
```
Select Sid marks
From Student
Where marks = (Select max(Marks) From Student)
```

Sid	Marks
S_3	90

OTHER SET OPERATOR

1. IN/NOT IN
2. ANY
3. ALL
4. EXISTS/NOT EXISTS

COMPARISION OPERATOR

<, >, < =, < >

Not equal

ANY: Compare a value with each value in a Set & Return true if any value is compared according to given condition.

Example:

$x > ANY\ (10, 20, 30)$

$(x > 10)$ or $(x > 20)$ or $(x > 30)$ $\xrightarrow{\text{output}}$ 11, 12, 13, ...

Example:

$x < ANY\ (10, 20, 30)$

$(x < 10)$ or $(x < 20)$ or $(x < 30)$ $\xrightarrow{\text{output}}$ 29, 28, 27, ...

ALL: Compare a value with each value in a Set & Return true if given condition satisfied for every/each value in the Set.

Example:

$x >$ ALL $(10, 20, 30)$

$(x > 10)$ AND $(x > 20)$ AND $(x > 30)$ $\xrightarrow{\text{output}}$ 31, 32, 33, 34 .....

Example:

$x <$ ALL $(10, 20, 30)$

$(x < 10)$ AND $(x < 20)$ AND $(x < 30)$ $\xrightarrow{\text{output}}$ 9, 8, 7, 6 .....

Q. Find Name of Supplier whose turnover is better than the turnover of any (some) Supplier of Delhi?

```
Select      Sname
From        Supplier
Where       City < > Delhi AND
            turnover > ANY (Select turnover
                            From Supplier
                            Where City = Delhi)
```

Supplier:

Sno	**Sname**	**City**	**turnover**
1	A	Delhi	4 Cr.
2	B	Bang.	5 Cr.
3	C	Delhi	6 Cr.
4.	D	Konchi	7 Cr.

ALL: (WHERE City = Delhi)

output

Sname
D

x > ANY (4.6)

(x > 4) or (x > 6) = 5, 6. 7

B, C, D

Q. Retrieve eid who get more salary than any employee of dept = 5?

EMP

Eid	dno	Salary
E_1	5	50 k
E_2	3	20 k
E_3	5	30 k
E_4	3	40 k
E_5	4	60 k

Query I:

```
Select      Eid
From        Emp
WHERE       dno < > 5
AND         Salary > ANY (Select Salary From Emp Where dno=5)
```

30k, 50k

output →

Eid
E_4
E_5

Query II:

```
Select      Eid
From        Emp
WHERE       dno < > 5
AND         Salary > (Select min(Salary) From Emp Where dno='5')
```

30k, 50k

output →

Eid
E_4
E_5

All employee of dno = 5

Query I:

```
Select    Eid
From      Emp
WHERE     dno < > 5
AND       Salary > ALL (Select Salary
                        From Emp
                        Where dno=5)
```

output →

Eid
E_5

Query II:

```
Select    Eid
From      Emp
WHERE     dno < > 5
AND       Salary > 50 (Select max (Salary)
                       From Emp
                       Where dno='5')
```

output →

Eid
E_5

Supplier (Sid, Sname, Rating)
Parts (Pid, Pname, Color)
Catalog (Sid, Pid, Cost)

Q. Retrieve Sid of the Supplier who supplied some Red Color Parts?

Query I:

Select Sid
From Catalog C, Parts P

output → | S_1 | S_2 |

WHERE P.Pid = C.Pid

Color = Red

Query II:

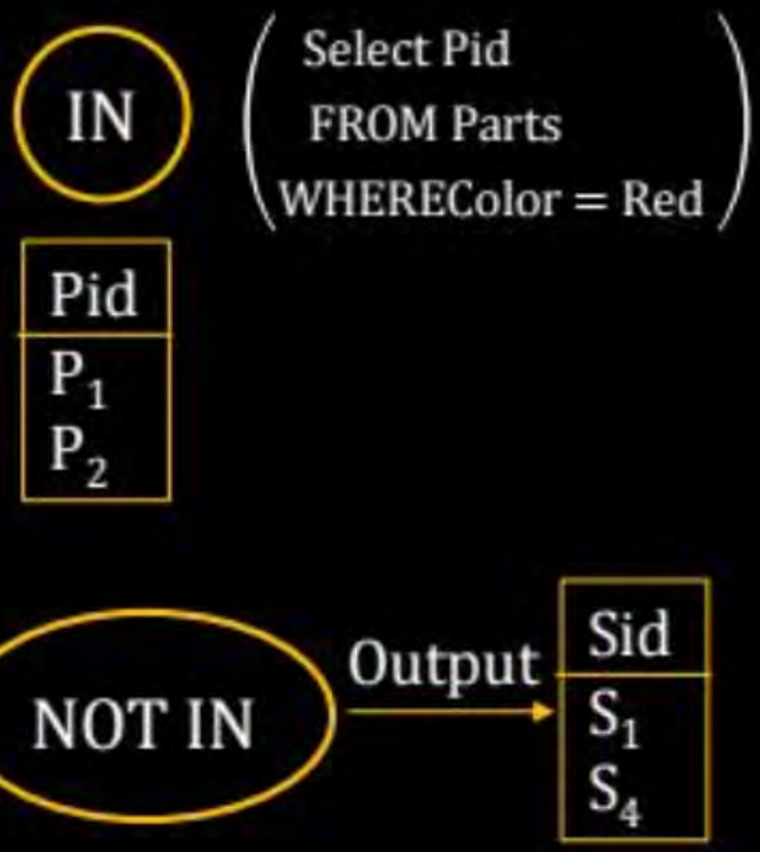
Select Sid
From Catalog

WHERE Pid = (Select Pid
FROM Parts
WHERE Color = Red)

One to many
Comparison not
Directly allowed

Query III:

Select Sid
FROM Catalog
WHERE Pid

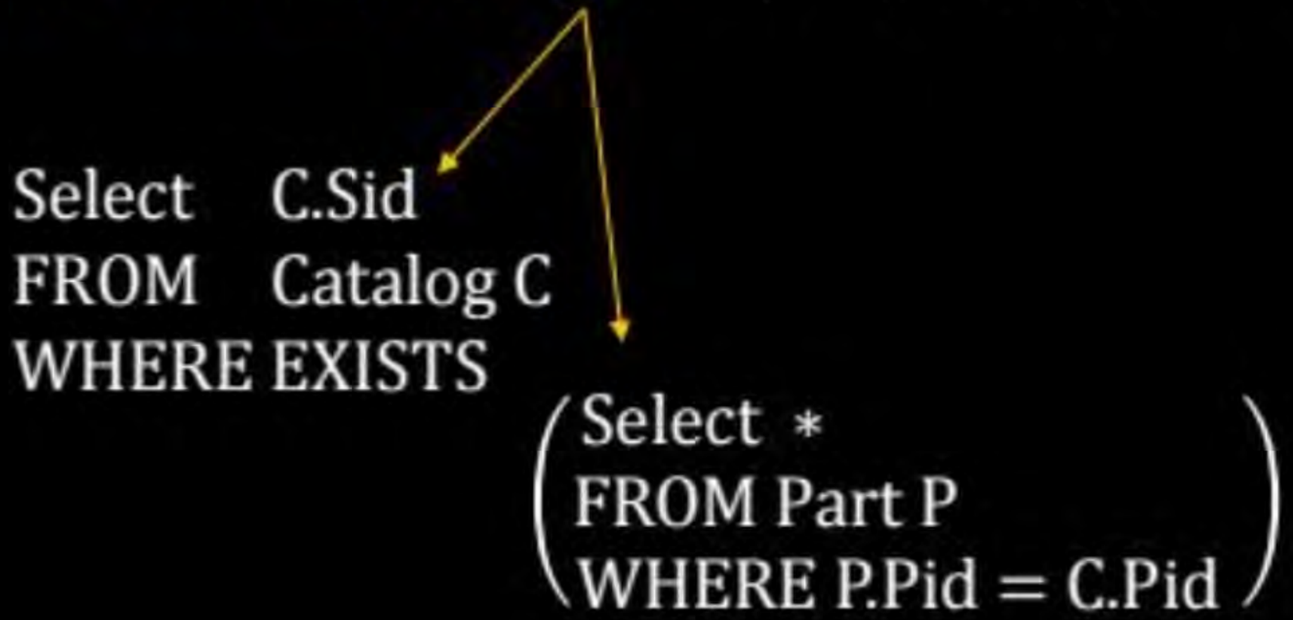


EXISTS: (Checks): Return True if Inner Query Result

EXISTS: (Checks): Return True if Inner Query Result Not Empty

NOT EXIST: Return True if Inner Result Empty

Correlated Nested Query: Inner Query Using attribute defined in Outer Query



Corelated Nested Query

```
Select   C.sid
FROM     Catalog
WHERE EXISTS ( Select*
               FROM Parts P
               WHEREP P.Pid = C.Pid
               AND Color = Red )
```

Inner Query using Attributes defined in the Outer Query

Nested Queries

(Independent)
Normal Nest Query

Inner → Outer

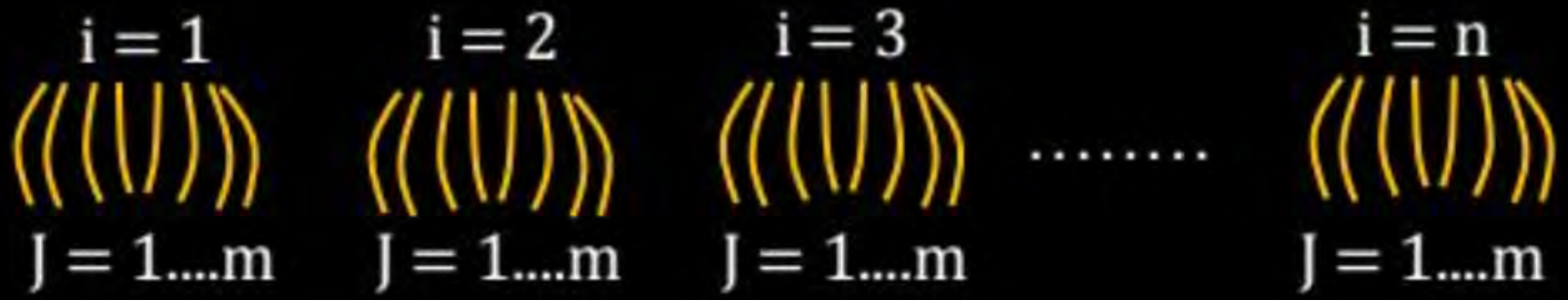
Bottom → Top

Corelated Nested
Query

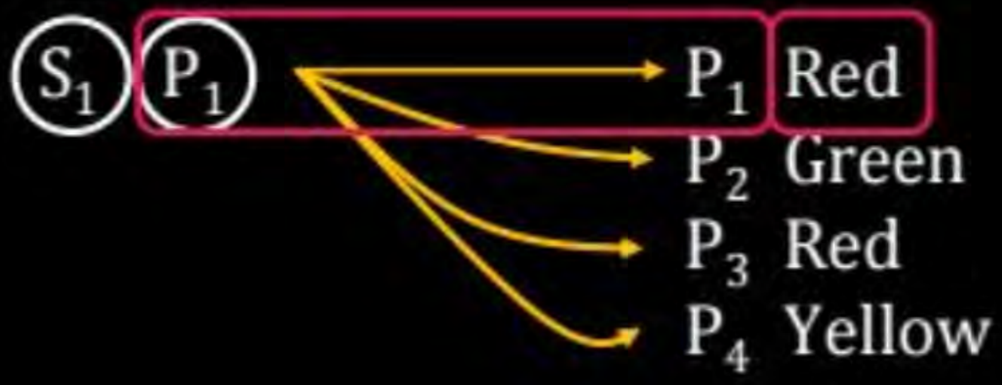
Outer → Inner → Outer

Top → Bottom → Top

```
for(i = 1; i < = n; i++)
  for(j = 1; j <= m, j++)
```


i = 1
i = 2
i = 3
i = n
J = 1....m
J = 1....m
J = 1....m
J = 1....m

I[st] Iteration:



[Pid Match & color = Red]

1 Tuple Return

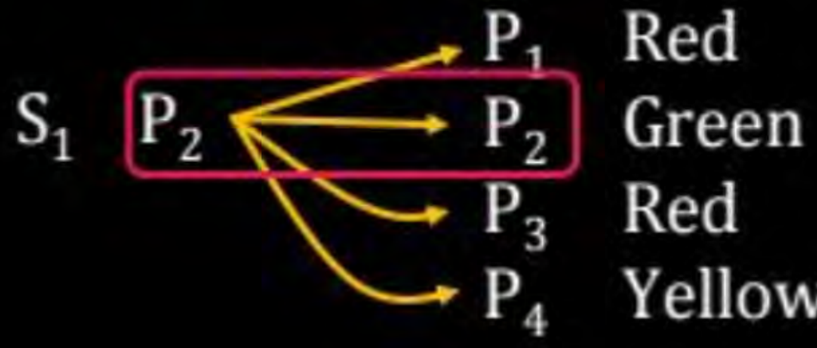
Catalog

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_3
S_4	P_4

Parts

Pid	Color
P_1	Red
P_2	Green
P_3	Red
P_4	Yellow

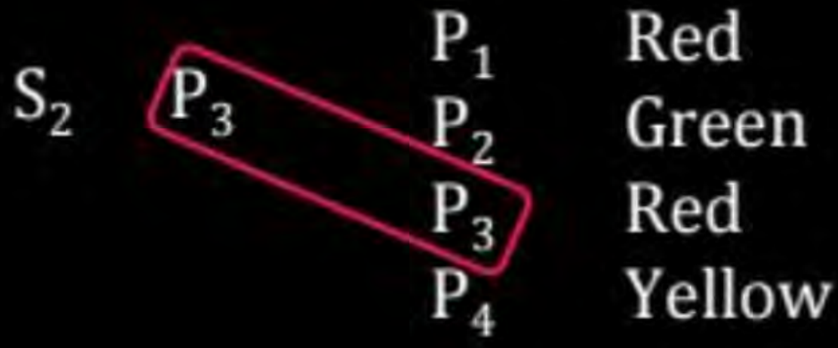
IInd Iteration:



[Pid Match but color not Red]

0 Tuple Return

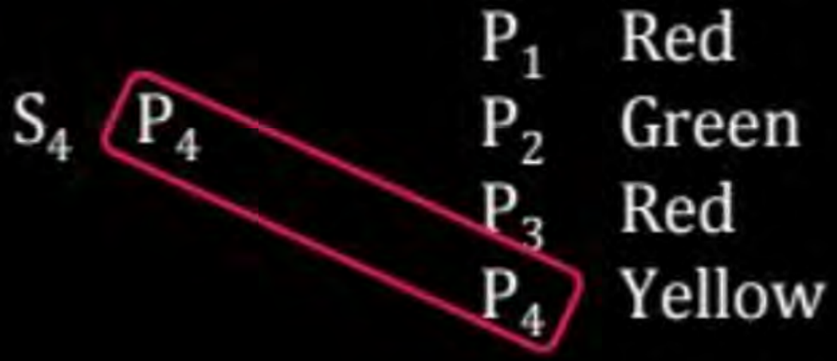
IIIrd Iteration:



[Pid Match & Color Red]

1 Tuple Return

IVth Iteration:



[Pid Match but color not Red]

0 Tuple Return

EXISTS

o/p

Sid
S_1
S_2

If NOT EXIST then output

Sid
S_1
S_4

Before EXIST & NOT EXISTS No Attribute is required.

Before IN & NOT IN Attribute is Required.

Q. Given Relative Schema

Emp(Eid, Ename, Salary)

Department(Eid, dname, code)

Retrieve Employee ID who have no Department?

Which is true?

A) Q_1 ✓ Q_2 ×

B) Q_2 ✓ Q_1 ×

C) Q_1 ✓ Q_2 ✓

D) Q_1 × Q_2 ×

Query I:
```
Select  Eid
FROM    Emp E, Dep D
WHERE E.Eid <> D.Eid
```

Query II: Select Eid
FROM Emp E
WHERE NOT EXISTS (Select * FROM Dep D WHERE E. Eid = D. Eid)

Query I:

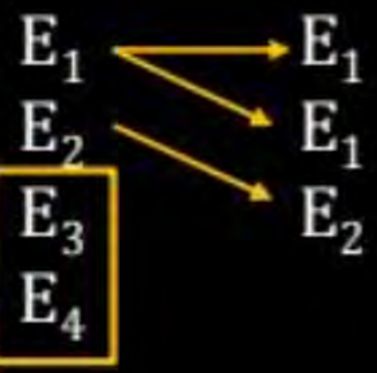
$E_1 <> E_1 \rightarrow F$
$E_1 <> E_1 \rightarrow F$
$E_1 <> E_2 \rightarrow T$
$E_2 <> E_1 \rightarrow T$
$E_2 <> E_1 \rightarrow T$
$E_2 <> E_2 \rightarrow F$
$E_3 <> E_1 \rightarrow T$
$E_3 <> E_1 \rightarrow T$
$E_3 <> E_2 \rightarrow T$
$E_4 <> E_1 \rightarrow T$
$E_4 <> E_1 \rightarrow T$
$E_4 <> E_2 \rightarrow T$

Output of Query I

Eid
E_1
E_2
E_2
E_3
E_3
E_3
E_4
E_4
E_4

Output of Query II

E_3
E_4



NULL:

- Non Zero
- Unknown
- Un existed
- No Two Null are equal

Q. Retrieve Eid who does not have any Passport Details?

```
Select   Eid
FROM     Emp
WHERE PPro = NULL
```

Employee

eid	ename	PassPortNo
E1	A	20
E2	B	NULL
E3	B	16

NOTE: for comparison with NULL SQL support IS/ IS Not Clause

```
WHERE PPro IS NULL
```

Q. Retrieve Eid who are having some passport details?

```
Select  Eid
FROM    Emp
WHERE PPro IS NOT NULL
```

Regular Expression:

%: Zero or More Character

_ : Exactly One character

Q. Retrieve Student whose Name Start with 'S' & end with M & at least 5 character?

→ 'S _ _ _ % M'

```
Select *
FROM Student
WHERE Sname = 'S _ _ _ % M'
```

Output: Pattern → 'S _ _ _ % M'

LIKE: is used to compare to specify certain search Condition for a Pattern in a column.

```
Select *
FROM Student
WHERE Sname LIKE 'S _ _ _ % M'
```

Q. Retrieve all student whose Name NOT Start with ‘C’?

```
Select    *
FROM    Student
WHERE Sname NOT LIKE ‘C%’
```

```
Select    *
FROM    Student
WHERE Sname
```

LIKE	A% →	Starts with A
	%J →	end with J
	%I% →	Contain join I
	'_ _ _ _' →	All 4 length Name
	'S _ _ _' →	Starts with S & exactly 4 length character
	'S _ _ _ %' →	Starts with S & at least length

$R \times S$:	CROSS Join
$R \cup S$:	Union Join
$R \bowtie S$:	Inner Join
$R \overset{\bowtie}{} S$:	Outer Join

ORDER By: Order By Clause is used to sort the Rows.

Company (Name, invoice no)

Q. Display all the company in alphabetical order of their Name?

```
Query:  Select    *
        FROM    Company
        Order by Name asc;
```

Q. If Reverse alphabetical order then
Order by Name Desc

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:

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

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

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

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

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

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

Q.

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

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(empNo, 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(<u>A</u>, B) and R_2(<u>C</u>, 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

$A \cup B \equiv A$

Consider the following relation A, B and C:

A		
ID	Name	Age
12	Arun	60
15	Shreya	24
99	Rohit	11
25	Hari	40
98	Rohit	20

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.

$(A \cup B) \bowtie_{A.Id \geq 40 \vee C.Id < 15} C$

[GATE-2012-CS: 2M]

A 7

B 4

C 5

D 9

Q.

Consider the following relation

Cinema (theatre, address, capacity)

Which of the following options will be needed at the end of the SQL query

```
SELECT P1.address
FROM Cinema P1
```

such that it always finds the addresses of theaters with maximum capacity?

[MCQ: 2015]

A. WHERE P1.capacity > = ALL (select P2.capacity from Cinema P2)

B. WHERE P1.capacity > = Any (select P2.capacity from Cinema P2)

C. WHERE P1.capacity > ALL (select max (P2.capacity) from Cinema P2)

D. WHERE P1.capacity > Any (select max (P2.capacity) from Cinema P2)

Q.

Database table by name Loan_Records is given below:
What is the output of the following SQL query?

```
SELECT count(*)
FROM (
        SELECT Borrower, Bank_Manager
        FROM Loan_Records) AS S NATURAL JOIN
        (SELECT Bank-Manager, Loan_Amount
         FROM Loan_Records) AS T
        );
```

[MCQ: 2011:2M]

A 3

B 9

C 5

D 6

Borrower	Bank_Manager	Loan_Amaount
Ramesh	Sunderajan	10000.00
Suresh	Ramgopal	5000.00
Mahesh	Sunderajan	7000.00

Q. SELECT operation in SQL is equivalent to [MCQ: 2015-1M]

A The selection operation in relational algebra.

B The selection operation in relational algebra, except that SELECT in SQL retains duplicates.

C The projection operation in relational algebra.

D The projection operation in relational algebra, except that SELECT in SQL retains duplicates.

Q.

Consider the following database table named top_scorer.
Consider the following SQL Query:

```
SELECT ta.player FROM top_scorer AS ta
WHERE ta.goals > ALL (SELECT tb.goals
      FROM top_scorer AS tb
      WHERE tb.country = 'Spain')
AND ta.goals > ANY (SELECT tc.goals
      FROM top_scorer AS tc
      WHERE tc.country = 'Germany')
```

The number of tuples returned by the above SQL query is _____.

[NAT:2017–2M]

top_scorer		
player	country	goals
Klose	Germany	16
Ronaldo	Brazil	15
G Muller	Germany	14
Fountaine	France	13
Pele	Brazil	12
Klinsmann	Germany	11
Kocsis	Hungary	11
Batistuta	Argentina	10
Cubillas	Peru	10
Lato	Poland	10
Lineker	England	10
T Muller	Germany	10
Rahn	Germany	10

Q.

Consider the following relations:

Student	
Roll_No	Student_Name
1	Raj
2	Rohit
3	Raj

Performance		
Roll_No	Course	Marks
1	Math	80
1	English	70
2	Math	75
3	English	80
2	Physics	65
3	Math	80

Consider the following SQL query.

```
SELECT S . Student_Name, Sum(P. Marks)
FROM Student S, Performance P
WHERE S. Roll_No = P.Roll_No
Group BY S. Student_Name
```

The number of rows that will be returned by the SQL query is _____.

[NAT: 2015–2M]

Q.

Consider the relational database with the following four schemes and their respective instances.

Student(sNo, sName, dNo) Dept(dNo, dName)
Course(cNo, cName, dNo) Register(sNo, cNo)

Student

sNo	sName	dNo
S01	James	D01
S02	Rocky	D01
S03	Jackson	D02
S04	Jane	D01
S05	Milli	D02

Dept

dNo	dName
D01	CSE
D02	EEE

Course

cNo	cName	dNo
C11	DS	D01
C12	OS	D01
C21	DE	D02
C22	PT	D02
C23	CV	D03

Register

sNo	cNo
S01	C11
S01	C12
S02	C11
S03	C21
S03	C22
S03	C23
S04	C11
S04	C12
S05	C11
S05	C21

Question Continues in Next Slide

SQL Query:

```
SELECT * FROM Student AS S WHERE NOT EXIST
    (SELECT cNo FROM Course WHERE dNo = “D01”.
        EXCEPT
SELECT cNo FROM Register WHERE sNo = S.sNo)
```

The number of rows returned by the above SQL query is _____.

[NAT: 2022: 2M]

Q.

Consider a database that has the relation schema EMP(EmpId, EmpName, and DeptName). An instance of the schema EMP and a SQL query on it are given below:

```
SELECT AVG(EC.Num)
FROM EC
WHERE(DeptName, Num) IN
        (SELECT DeptName, COUNT(EmpId)AS
                                EC(DeptName, Num)
        FROM EMP
        GROUP BY DeptName)
```

The output of executing the SQL query is _____.

[NAT:2017-1M]

EMP		
EmpId	EmpName	DeptName
1	XYA	AA
2	XYB	AA
3	XYC	AA
4	XYD	AA
5	XYE	AB
6	XYF	AB
7	XYG	AB
8	XYH	AC
9	XYI	AC
10	XYJ	AC
11	XYK	AD
12	XYL	AD
13	XYM	AE

Q.

A relational database contains two tables Student and Performance as shown below:

Student

Roll_no	Student_name
1	Amit
2	Priya
3	Vinit
4	Rohan
5	Smita

Performance

Roll_no	Student_code	Marks
1	A	86
1	B	95
1	C	90
2	A	89
2	C	92
3	C	80

The primary key of the Student table is Roll_no. For the Performance table, the columns Roll_no. and Subject_code together from the primary key. Consider the SQL query given below:

SELECT S.Student_name, sum(P.Marks) FROM Student S, Performance P
WHERE P. Marks > 84 GROUP BY S. Student_name;

The number of rows returned by the above SQL query is _____.

[NAT: 2019–2M]

Q.

Consider the following database table named water_schemes:
The number of tuples returned by the following SQL query is:

water_schemes		
scheme_no	district_name	capacity
1	Ajmer	20
1	Bikaner	10
2	Bikaner	10
3	Bikaner	20
1	Churu	10
2	Churu	20
1	Dungargarh	10

```
with total(name, capacity) as
select district_name, sum(capacity)
from water_schemes
group by district_name
with total_avg(capacity) as
select avg(capacity)
from total
select name
  from total, total_avg
  where total.capacity ≥ total_avg.capacity
```

[NAT:2016-2M]

Q.

Suppose $R_1(\underline{A}, B)$ and $R_2(\underline{C}, D)$ are two relation schemas. 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?

[MCQ: 2012–2M]

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

Q.

Consider a relational database containing the following schemas.

Catalogue		
sno	pno	cost
S1	P1	150
S1	P2	50
S1	P3	100
S2	P4	200
S2	P5	250
S3	P1	250
S3	P2	150
S3	P5	300
S3	P4	250

Suppliers		
sno	sname	location
S1	M/s Royal furniture	Delhi
S2	M/s Balaji furniture	Bangalore
S3	M/s Premium furniture	Chennai

Parts		
pno	**pname**	**part_spec**
P1	Table	Wood
P2	Chair	Wood
P3	Table	Steel
P4	Almirah	Steel
P5	Almirah	Wood

Question Continues in Next Slide

The primary key of each table is indicated by underlining the constituent field

```
SELECT s.no, s.name
FROM Suppliers s, Catalogue c
WHERE s.sno = c.sno AND
        cost > (SELECT AVG (cost)
                FROM Catalogue
                WHERE pno = 'p4'
               GROUP BY pno);
```

The number of rows returned by the above SQL query is [MCQ: 2020-2M]

(A) 4
(B) 5
(C) 0
(D) 2

MCQ

Consider a database table T containing two columns X and Y each of type integer. After the creation of the table, one record (X=1, Y=1) is inserted in the table.

Let MX and MY denote the respective maximum values of X and Y among all records in the table at any point in time. Using MX and MY, new records are inserted in the table 128 times with X and Y values being MX+1, 2*MY+1 respectively. It may be noted that each time after the insertion, values of MX and MY change.

What will be the output of the following SQL query after the steps mentioned above are carried out?

SELECT Y FROM T WHERE X=7;

[GATE-2011-CS: 2M]

A 127

B 255

C 129

D 257

THANK YOU!