COMPUTER SCIENCE

Database Management System

Query Language

Lecture_6

Vijay Agarwal sir

TOPICS TO BE COVERED

01 SQL Clauses

02 SQL Operators

Relational Algebra. ✓

SQL

SQL clauses.

SELECT
FROM
WHERE

GROUP By
HAVING

Set operators

UNION & UNION ALL
Intersect & Intersect ALL
MINUS & MINUS ALL.

ANY ✓

ALL ✓

STUDENT (Sid, Name . Branch, Marks)

Q

W.A.Q (Write A Query) to Reterive Sid of the Student Who secured (1st Highest) Highest Marks ?

- RA
- SQL
- Max.

STUDENT

SID	Name	Branch	Marks
S_1	Khushi	CS	80
S_2	Devansh	IT	85
S_3	Rachit	CS	97
S_4	Bhavya	IT	92

Sid
S3

Avg

Relational Algebra

gives Sid who Marks Not Maximum .

Conditional Join

Rename

$$\Pi_{sid}(Student) - \Pi_{sid}\left[Student \underset{marks<M}{\bowtie} \rho_{I,N,B,M}(Student)\right]$$

80
85
97
92

Sid
S_1
S_2
S_3
S_4

−

Sid
S_1
S_2
S_4

=

Sid
S_3

S_1 ⇐ 80 < 80, 85 ✓, 97 ✓, 92 ✓

S_2 85 < 80, 85, 97 ✓, 92 ✓

97 × 80 ×, 85 ×, 97 ×, 92 ×

S_4 92 < 80, 85, 97 ✓, 92

80
85
97
92

Using SQL

Sid
S1
S2
S3
S4

```
SELECT Sid
FROM Student
```

—

EXCEPT MINUS

⇒

Sid
S3

Sid
S1 ✓
S2 ✓
S4 ✓

```
Select T1.Sid
FROM Student As T1, Student As T2
WHERE T1.marks < T2.marks
```

T1	T2
80	80
85	85
97	97
92	92

Using Max function

```
Select Sid
 FROM Student

                            97
WHERE Marks = (Select  max(marks)
                FROM   Student  )
```

Minimum

Relational Algebra

Inner Query gives Sid who Marks Not Minimum

Conditional Join

Rename

$$\Pi_{sid}(\text{Student}) - \Pi_{sid}\left[\text{Student} \bowtie_{marks > M} \rho_{I,N,B,M}(\text{Student})\right]$$

80
85
97
92

Sid
S_1
S_2
S_3
S_4

$-$

sid
S_2
S_3
S_4

o/p

sid
S_1

✗ 80 > 80
85
97
92

S_3 97 > 80 ✓
85 ✓
97
92 ✓

80
85
97
92

✓ S_2 85 > 80 ✓
85
97
92

92 > 80 ✓
85 ✓
97
92 ✓
S_4

80
85
97
92

Home Work

STUDENT (SId, Name . Branch, Marks)

Q

W.A.Q (Write A Query) to Reterive Sid of the Student Who secured (IInd Highest) 2nd Highest Marks ?

RA

SQL

Max.

STUDENT

SID	Name	Branch	Marks
S_1	Khushi	CS	80
S_2	Devansh	IT	85
S_3	Rachit	CS	97
S_4	Bhavya	IT	92

Subblier (Sid Sname Rating)

Parts (Pid Pname Color)

Catalog (Sid Pid Cost)

① Q: W.A.Q to Reteeive Sid whose Parts Cost greater than 90,000?

Sol^n

```
SELECT Sid
FROM Catalog
WHERE Cost >90000.
```

Subblier (Sid Sname Rating)

Parts (Pid Pname Color)

Catalog (Sid Pid Cost)

② Q W.A.Q to Reterive Pid of Red Color Parts ?

Solⁿ 2

```
SELECT Pid
  FROM Parts
 WHERE Color = 'Red'
```

Subblier (Sid Sname Rating)

- Parts (Pid Pname Color)
- Catalog (Sid Pid Cost)

Rename Operator

As

SPACE

3 Q W.A.Q to Reteeive Sid of Red Color Part ?

```
SELECT Sid
 FROM Parts P, Catalog As C
 WHERE P.Pid = C.Pid AND
       P.Color = Red
```

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

(4) Q W.A.Q to Retreive Sname who supplied Red Color Part ?

Solⁿ

```
SELECT Sname
 FROM Supplier S, Parts P, Catalog C
 WHERE    P.Pid = C.Pid AND
          C.sid = S.sid AND
          P.Color = Red.
```

OR

```
Color = Red
```

Supplier ($\underline{\text{Sid}}$ Sname Rating)

Parts ($\underline{\text{Pid}}$ Pname Color)

Catalog ($\underline{\text{Sid Pid}}$ Cost)

Supplier ($\underline{\text{Sid}}$ Sname Rating)

Parts ($\underline{\text{Pid}}$ $\underline{\text{Pname}}$ Color)

Catalog ($\underline{\text{Sid Pid}}$ Cost)

Rating > 5, ⇒ Pname.

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

Supplier

Sid	Sname	Rating
S_1	A	
S_2	B	
S_3	C	
S_4	D	
S_5	E	

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

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

Supplier

Sid	Sname	Rating
S_1	A	
S_2	B	
S_3	C	
S_4	D	
S_5	E	

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

$\boxed{\begin{matrix} S_1 \\ S_2 \end{matrix}}$ Ans

Q.1 Retrieve Sid of the Supplier who supplied some Red Color Parts?

Query I:

```
Select      Sid
From        Catalog C, Parts P

WHERE       P.Pid = C.Pid AND
            Color = Red
```

output →

Sid
S_1
S_2

FROM

↓

WHERE

Query II:

```
Select    Sid
From      Catalog
WHERE     Pid = (Select Pid
                 FROM Parts
                 WHERE Color = Red)
```

P_1, P_3

P_1

Pid
P_1
P_3

→ One to many Comparison not Directly allowed

$$\begin{tikzcd} & P_1 \\ P_1 \arrow[ur] \arrow[dr] & \\ & P_3 \end{tikzcd}$$

Independent Inner Query

One to Many Comparison

In : In operator is used to Compare a value with List of value .

↳ Member Ship set .

$\boxed{a=4}$ X : [1, 2, 3, 5]

a In X : False

$\boxed{p=2}$ p In X : True .

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

Supplier

Sid	Sname	Rating
S_1	A	
S_2	B	
S_3	C	
S_4	D	
S_5	E	

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

S_1
S_2

Ans

Query III:

```
Select   Sid
FROM     Catalog   IN   (Select Pid
WHERE    Pid             FROM Parts
                         WHEREColor = Red)
```

P_1
P_3

Pid
P_1
P_2

Output →

Sid
S_1
S_2

P_1 IN (P_1 P_3) : True
→ S_1 ✓

P_2 IN (P_1, P_3) : False

P_3 IN (P_1, P_3) : True
→ S_2 ✓

P_4 IN (P_1, P_3) : False

sid
S_1
S_2

NOT IN Output →

Sid
S_1
S_4

NOT IN

NOT IN.

P_1 NOT IN [P_1 P_3] : False

P_2 NOT IN [P_1, P_3] : True
→ S_1 ✓

P_3 NOT IN (P_1 P_3] : False.

P_4 NOT IN (P_1 P_3) : True S_4 ✓

Query III:

```
Select    Sid
FROM      Catalog
WHERE     Pid    IN   ( Select Pid
                        FROM Parts
                        WHEREColor = Red )
```

P_1
P_2

Pid
P_1
P_2

Output →

Sid
S_1
S_2

NOT IN

Output →

Sid
S_1
S_4

① Normal

② IN | NOT IN

③ EXIST | NOT EXIST.

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

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

```
Select  C.Sid
FROM    Catalog C
WHERE EXISTS
            (Select *
             FROM Part P
             WHERE P.Pid = C.Pid
               AND Color = Red
            )
```

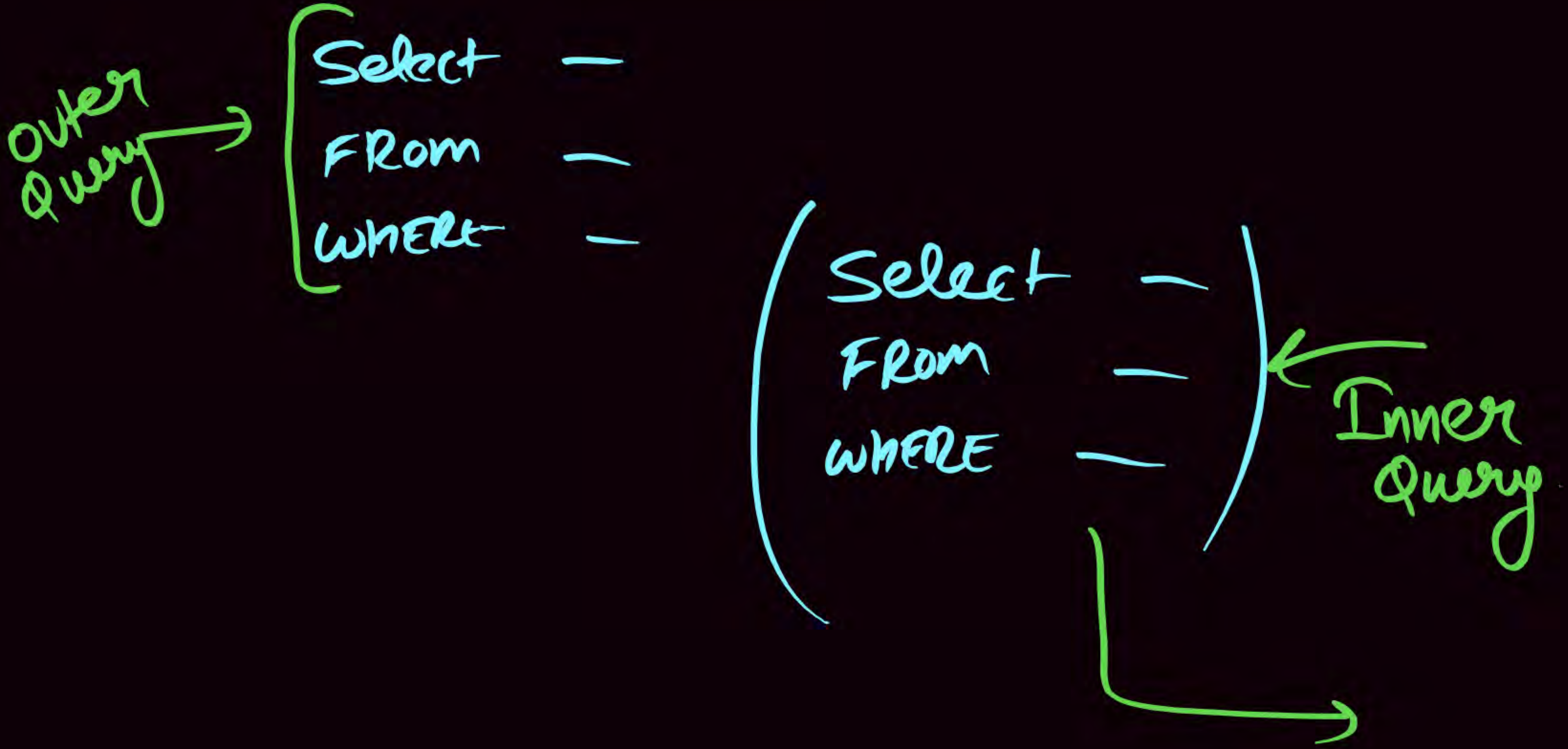
EXISTS: (Checks): Return True if Inner Query Result Non Empty

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

```
Select   C.Sid
FROM     Catalog C
WHERE EXISTS
              (Select *
               FROM Part P
               WHERE P.Pid = C.Pid)
```


Outer Query
Select —
FROM —
WHERE —
Select —
FROM —
WHERE —
Inner Query

Nested Query

① Normal Nested (Independent) Query

Execution sequence:

Inner Query → outer Query

(Bottom → Top)

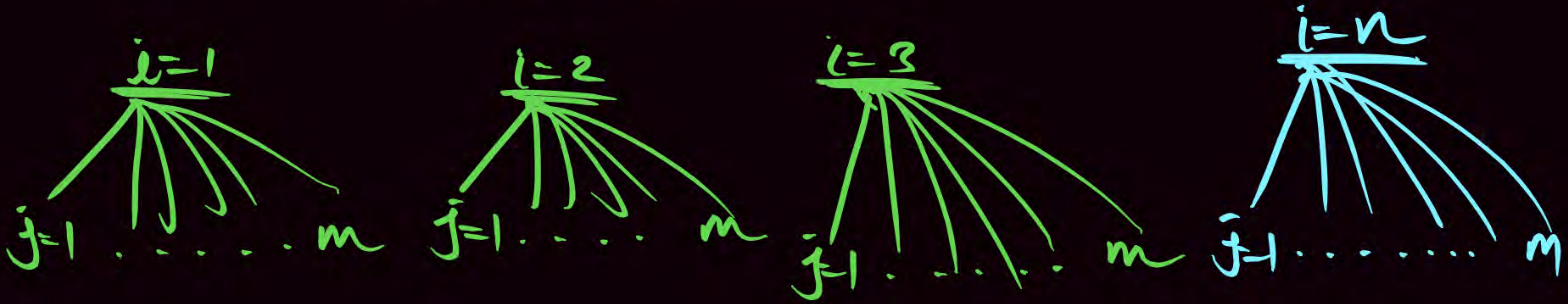
② Co-related Nested Nested Query.

Execution sequence:

Outer Query → Inner Query → outer Query.

TOP → Bottom → Top.

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



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

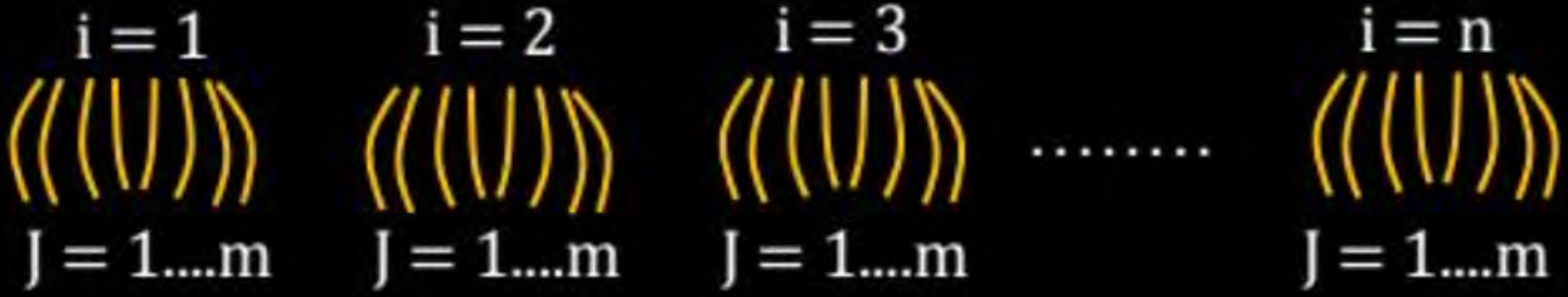
Bottom → Top

Corelated Nested
Query

Outer → Inner → Outer

Top → Bottom → Top

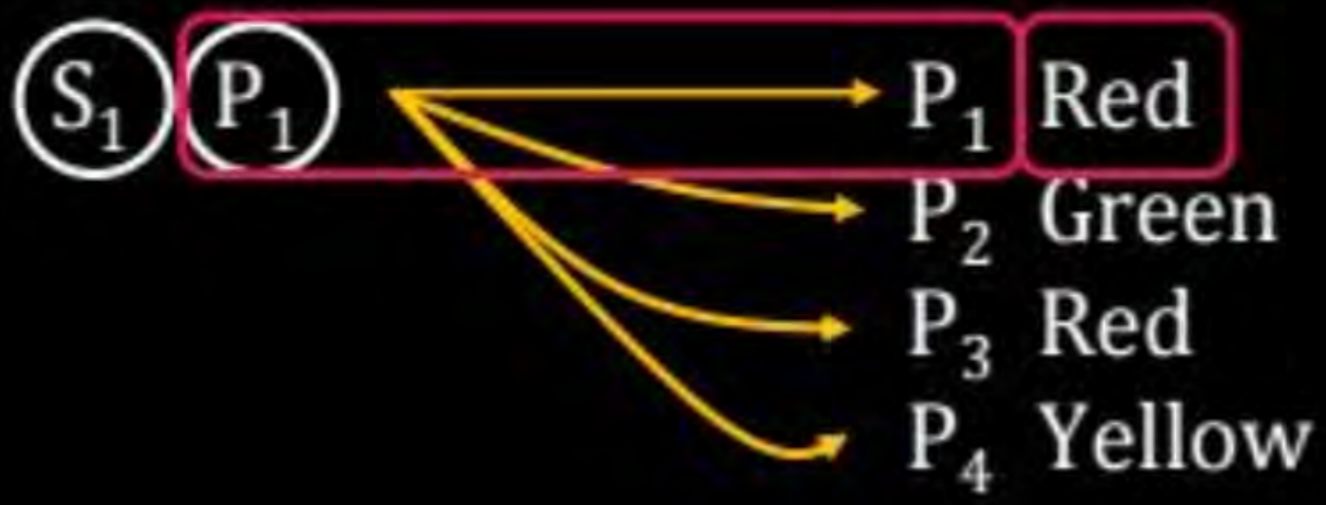
```
for(i = 1; i < = n; i++)
 for(j = 1; j <= m, j++)
```



i = 1 i = 2 i = 3 …….. i = n

J = 1....m J = 1....m J = 1....m J = 1....m

Ist Iteration:



S_1 P_1

P_1 Red

P_2 Green

P_3 Red

P_4 Yellow

[Pid Match & color = Red]

1 Tuple Return

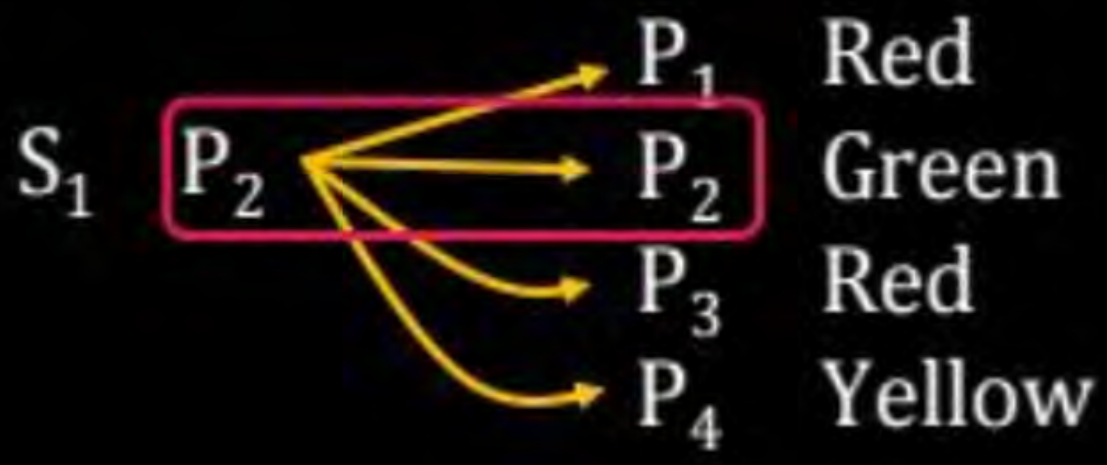
Catalog

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:



S_1 P_2

P_1 Red

P_2 Green

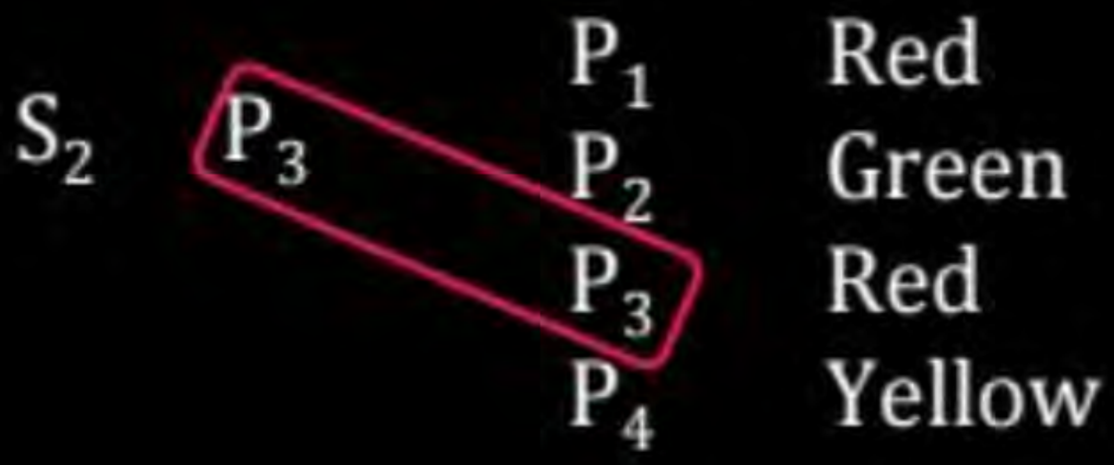
P_3 Red

P_4 Yellow

[Pid Match but color not Red]

0 Tuple Return

IIIrd Iteration:



S_2 P_3

P_1 Red

P_2 Green

P_3 Red

P_4 Yellow

[Pid Match & Color Red]

1 Tuple Return

IV^th Iteration:

S_4 P_4

P_1	Red
P_2	Green
P_3	Red
P_4	Yellow

[Pid Match but color not Red]

$\boxed{\text{0 Tuple Return}}$

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?

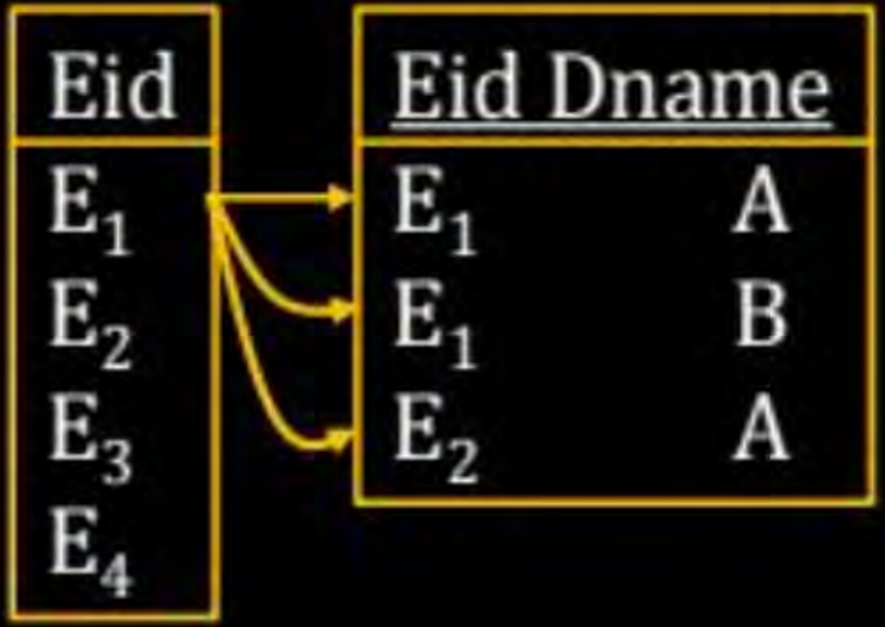
A) Q_1 ✓ Q_2 ×

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
```



Eid
E_1
E_2
E_3
E_4

Eid	Dname
E_1	A
E_1	B
E_2	A

Query II:

```
Select      Eid
FROM        Emp E
WHERE       NOT EXISTS (Select *
                        FROM Dep D
                        WHERE E. Eid = D. Eid)
```

Query I:

$E_1 <> E_1 \rightarrow F$

$E_1 <> E_1 \rightarrow F$

$E_1 <> E_2 \rightarrow T$

$E_2 <> E_1 \rightarrow T$

$E_2 <> E_1 \rightarrow T$

$E_2 <> E_2 \rightarrow F$

$E_3 <> E_1 \rightarrow T$

$E_3 <> E_1 \rightarrow T$

$E_3 <> E_2 \rightarrow T$

$E_4 <> E_1 \rightarrow T$

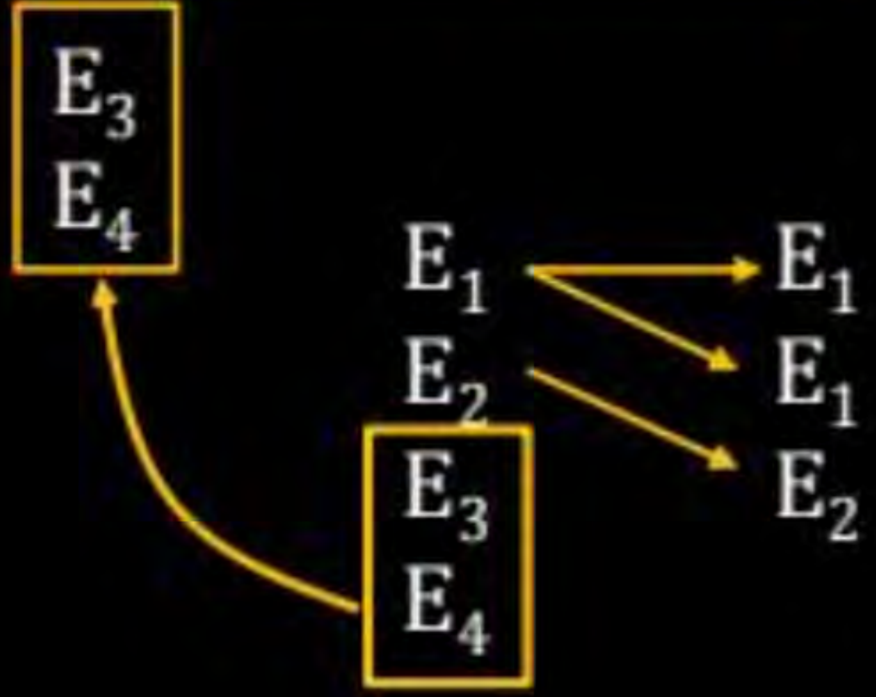
$E_4 <> E_1 \rightarrow T$

$E_4 <> E_2 \rightarrow T$

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



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:

e2
e3

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

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

eid	Age
e1	30
e2	40
e3	50

eid	Age
e1	30
e2	40
e3	50

(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

THANK YOU!