COMPUTER SCIENCE

Database Management System

Query Language

Structured Query Language(SQL)

Part-2

Lecture_4

Vijay Agarwal sir

01 SQL Clauses

02 SQL Operators

Relational Algebra.

1. Selection [σ]
2. Projection [Π]
3. Cross Product [X]
4. UNION [U]
5. Set Difference [−]
6. Rename [ρ]

Derived operator.

Intersection [∩]

Join & its type [⋈]

DIVISION. [÷]

Queries

Optimize Queries

GATE Question

Earlier: SEQUEL

↳ Structured English Query language.

SQL [Structured Query language]

RA

↓

Procedural Query Lang.

↓

By Default eliminate Duplicate Value.

SQL

↓

Non Procedural Query Language.

↓

Retain Duplicates.

SQL[Structured Query Language]

- DDL(Data Definition Language): Modification allowed at schema (Definition) level
  - 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 → kill.
- DQL(Data Query Language): Used to Retrieve the Data from DB
  - SELECT
  - FROM
  - WHERE

[ ] ← optional.

SQL

RA

SELECT [DISTINCT] $A_1 A_2 A_3 \ldots A_n$ ≡ Projection [Π]

FROM $R_1 R_2 R_3 \ldots R_m$ ≡ Cross Product [$\times$]

WHERE Condition. ≡ Selection [σ]

R.A

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

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

SELECT IN SQL

Duplicate

Retain

Π in Relational Algebra.

No Duplicates.

(Eliminate Duplicates)

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

SQL Clauses

[ ] ← Optional clause

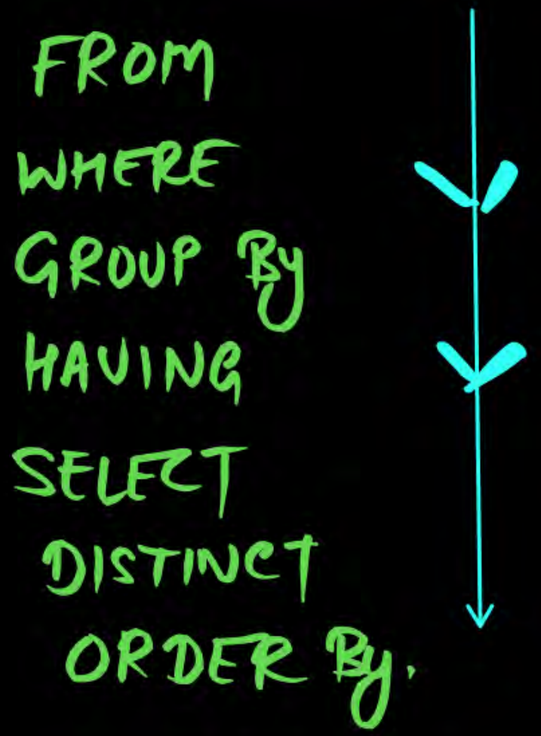
SELECT [DISTINCT] A_1 A_2 A_3 ....A_n

- FROM R_1 R_2 R_3....R_m
- [WHERE P]

[GROUP By Attribute [[HAVING Condition]]]

[ORDER By Attribute [[DESC]]]

Execution Sequence



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)

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

Aggregate operator :

Aggregate operator Always Discard the Null Value.

1. COUNT
2. SUM
3. AVG
4. MIN
5. MAX

Aggregation operator ⇒ Always Discard Null Value

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

(*)
Count (Constant)
⇓
# of Tuples.

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

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

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)

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

90
70
70
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) = 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) $\text{AVG([Distinct]marks)} = \frac{216}{3}$

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

✓ FROM

✓ WHERE

✓ GROUP By

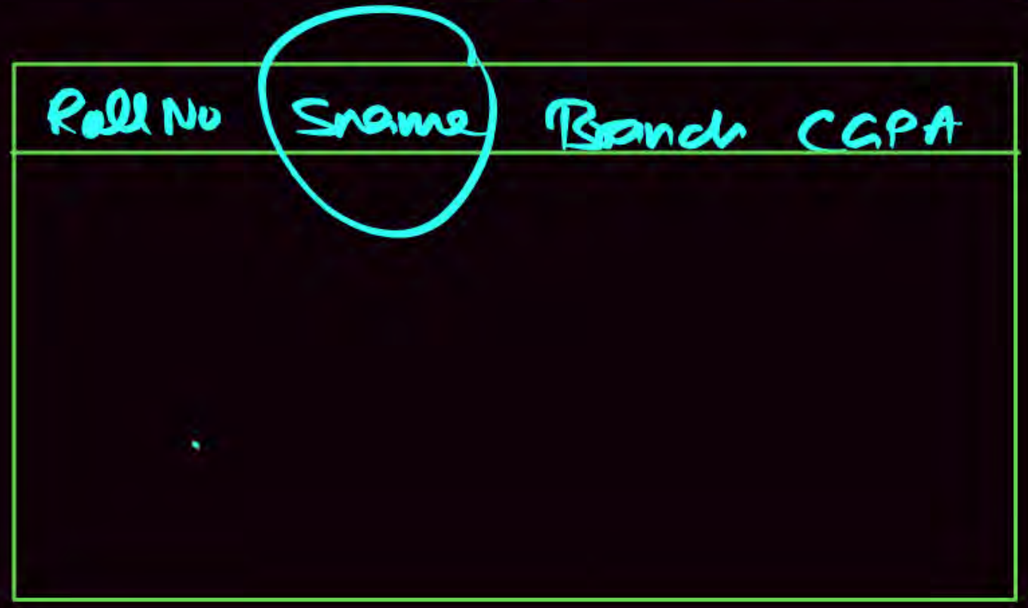
✓ Aggregate operator

HAVING ⇒ Applied on each GROUP.

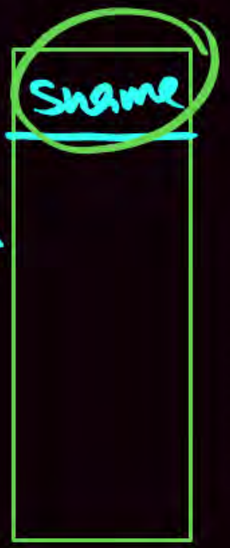
SELECT Sname

FROM Student

WHERE —



O/P



SELECT *

FROM STUDENT

WHERE —

output (O/P) ⇓ O/P

Roll No	Sname	Branch	CGPA

Note Select * means all attribute from that table & the tuples which satisfy condition.

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

$\frac{150}{2} = 75 > 61$

$\frac{130}{2} = 65 > 61$

$\frac{55}{1}$

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

Condition Applied on each Branch (Groups)

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

55.

ANSWER: 55

Q.2

Select min(marks)

FROM Student

WHERE Branch = 'CS'

Sol'n 2

60.

ANSWER: 60

Q.3 Select min(marks)

FROM Student

GROUP By (Branch)

Soln 3

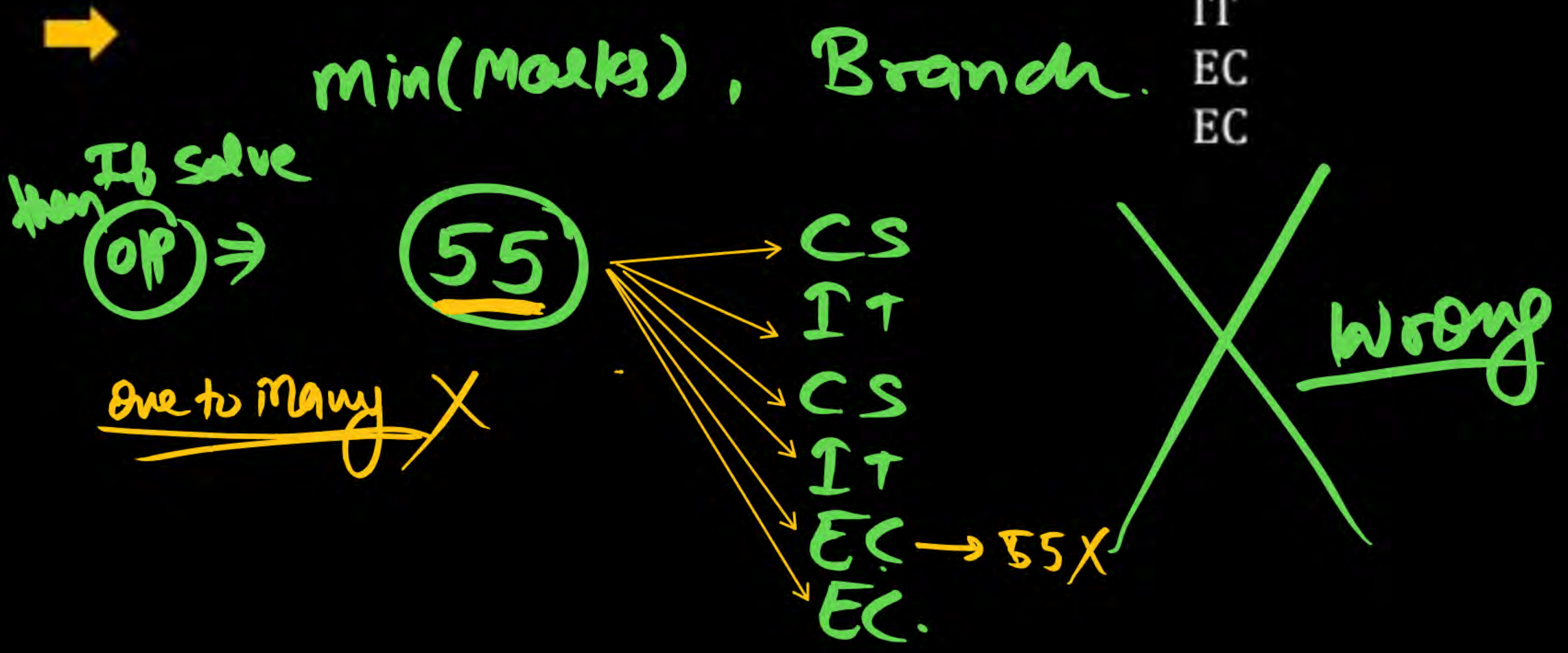
60

60

55

Q.4 Select min(marks), Branch
FROM Student

55	CS
	IT
	CS
	IT
	EC
	EC



Q.4 Select min(marks), Branch
FROM Student

CS
IT
CS
IT
EC
EC

➡ Such Syntax is not allowed in SQL

✗ Wrong Query.

Select min(marks), Branch
FROM Student
GROUP BY (BRANCH)

✓ Correct Query.

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)
```

O/P.

min(marks)	Branch
60	CS
60	IT
55	EC

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	

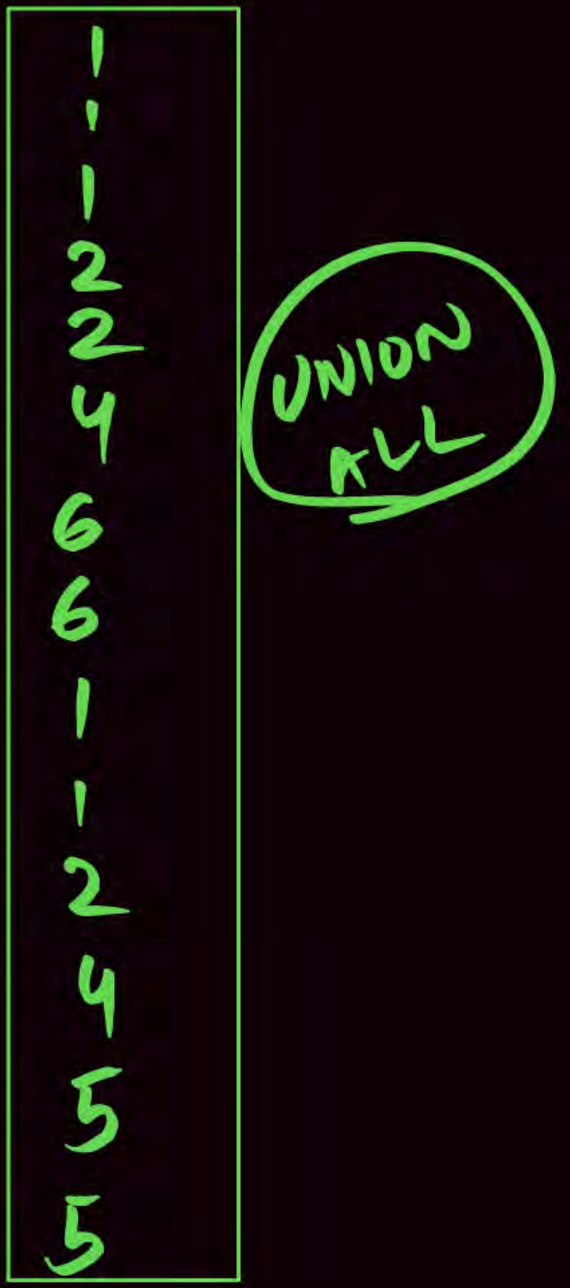
R.A

R
1
1
1
2
2
4
6
6

S
1
1
2
4
5
5

1. R UNION S
   R UNION ALL S

UNION.



UNION ALL

2. R INTERSECT S
   R INTERSECT ALL S.

3. R MINUS S.
   R MINUS ALL S.

Set operator in SQL.

① UNION	& UNION ALL.
② Intersect	& Intersect ALL.
③ MINUS	& MINUS ALL.
followed by RA	Not followed by R.A

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 in R — # Duplicates 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:

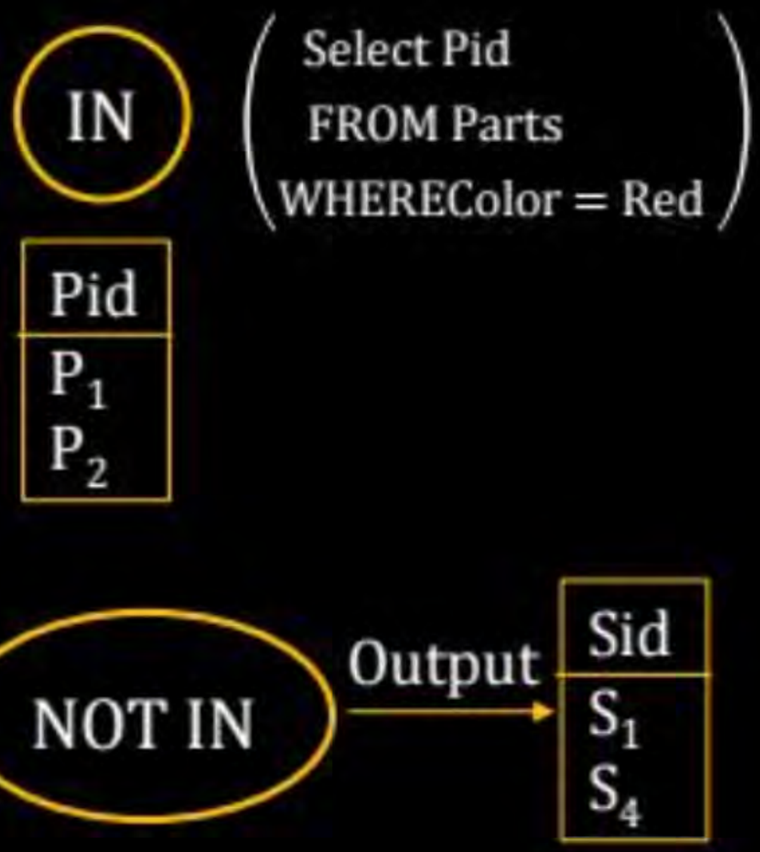
Select Sid
From Catalog

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

One to many
Comparison not
Directly allowed

Query III:

Select Sid
FROM Catalog
WHERE Pid



Any Doubt ?

THANK YOU!