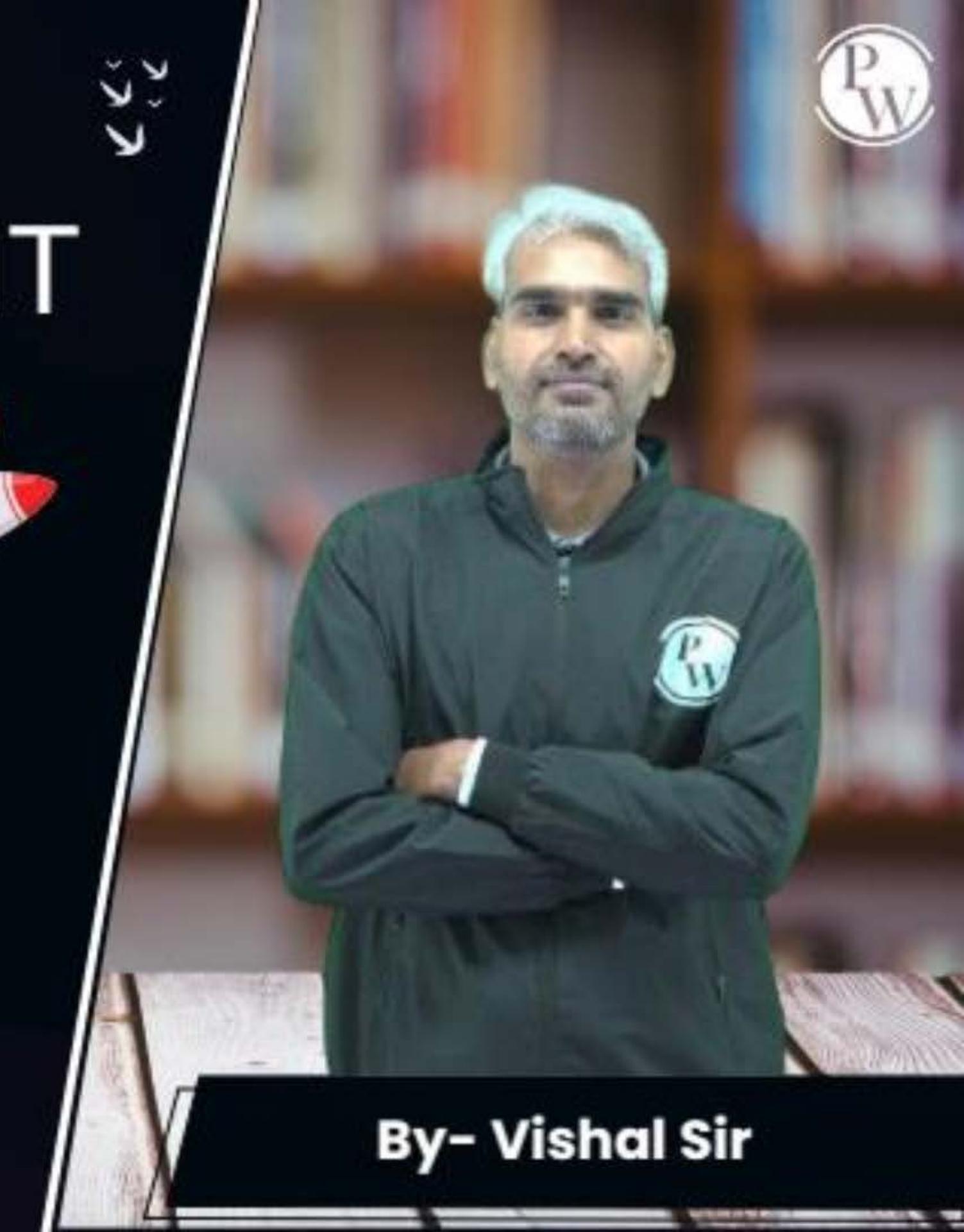


Computer Science & IT

Database Management System

Query Languages

Lecture No. 06



By- Vishal Sir

Recap of Previous Lecture



- Topic** SQL
- Topic** Basic SQL clauses
- Topic** Aggregate functions
- Topic** SQL commands

Topics to be Covered



- * **Topic** SQL clauses
- * **Topic** Nested query (sub-query)
- * **Topic** Operators w.r.t. Nested query
- * **Topic** Order of execution of nested query





Topic : Order of execution

Order of Execution:-

1. From
2. Where
3. Group By
4. Having
5. Select
→ "distinct"
6. Order BY



Topic : SQL clauses



FROM:- From clause is used to select the tables from the database.

Select *
From Student
= (Student)
All
R.A.

O/p will be Complete Student table

Student

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC



Topic : SQL clauses



FROM:- From clause is used to select the tables from the database.

Select Sid
From Student

Select distinct Sid
From Student

Sid
S ₁
S ₂
S ₃
S ₄
S ₅
S ₆

Student

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC



Topic : SQL clauses



FROM:- From clause is used to select the tables from the database.

Select Sname
From Student

Select distinct Sname
From Student

Sname
A
A
B
A
C
C

Sname
A
B
C

Student			
Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC



Topic : SQL clauses



WHERE:- Used to select the tuples from the database table based on conditions specified with WHERE clause.

Retrieves Sid of Students who scored more than 40 Marks

Select Sid
From Student
Where Marks > 40

Sid
S3
S4

Student

Sid	Sname	Marks	Branch
X S1	A	40	CS
X S2	A	20	IT
✓ S3	B	60	CS
✓ S4	A	60	EC
X S5	C	40	IT
X S6	C	NULL	EC



Topic : SQL clauses



WHERE:- Used to select the tuples from the database table based on conditions specified with WHERE clause.

Retrievre Sids of students whose marks are more than or equal to 20 and less than or equal to 40

Select Sid
From Student
Where (Marks \geq 20 AND Marks \leq 40)

O/p:

Sid
S1
S2
S5

Student

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC



Topic : SQL clauses



WHERE:- Used to select the tuples from the database table based on conditions specified with WHERE clause.

Retrievre Sids of students whose marks are more than or equal to 20 and less than or equal to 40

Select Sid
From Student

Where Marks between 20 AND 40

Both inclusive

Student

Sid	Sname	Marks	Branch
✓ S1	A	40	CS
✓ S2	A	20	IT
✗ S3	B	60	CS
✗ S4	A	60	EC
✓ S5	C	40	IT
✗ S6	C	NULL	EC



Topic : SQL clauses



GROUP BY:- GROUP BY clause is used to group the result of WHERE clause.

If Where Clause
is not present, then
{all rows will be used
for grouping}

Query: Retrieve names of all branches along with maximum marks in that branch.

Select Branch, Max(Marks)
From Student

invalid query

Student

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC

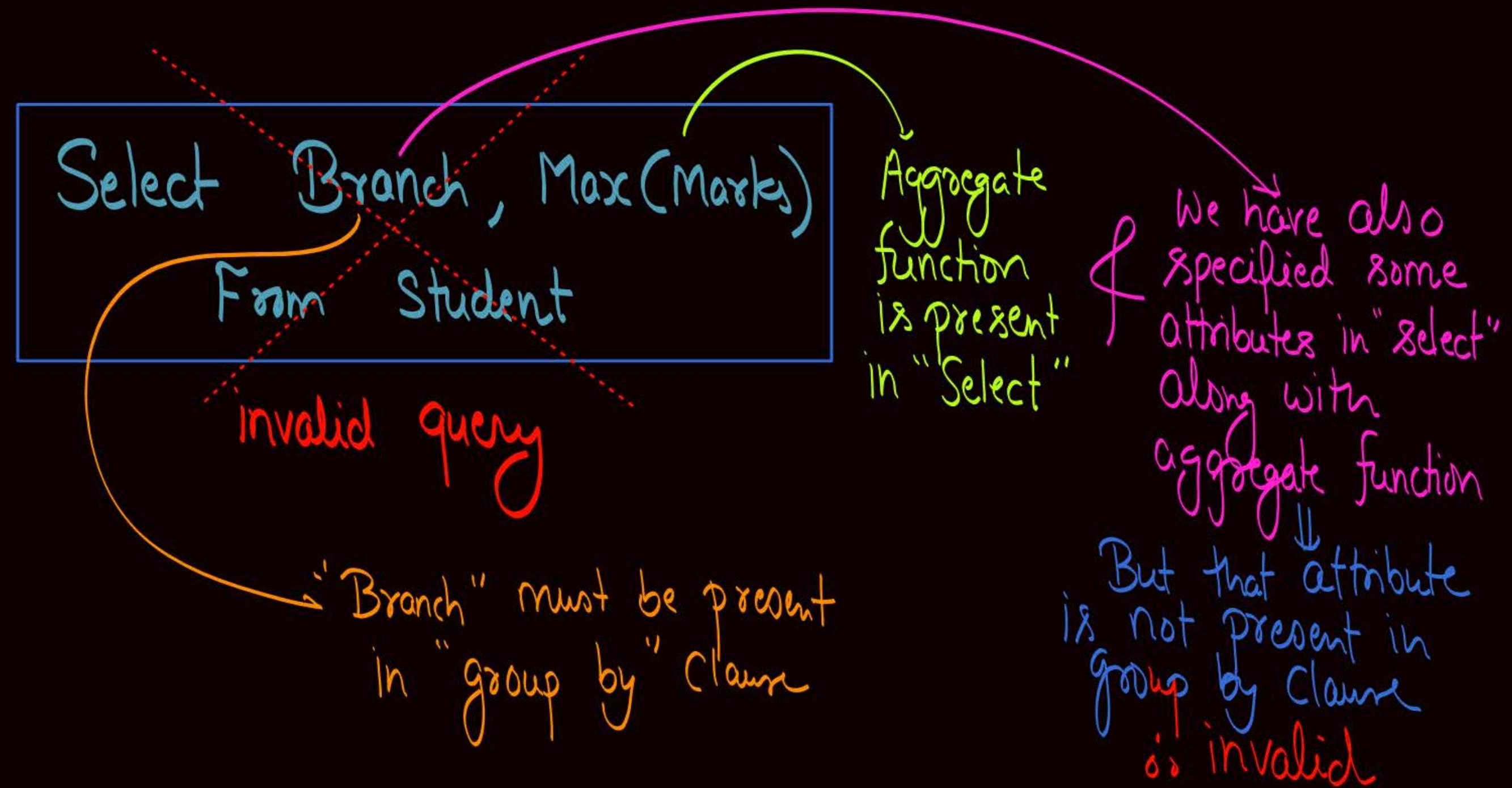
* Select Branch \Rightarrow O/P:

Branch
CS
IT
CS
EC
IT
EC

* Select Max(Marks)
From Student \Rightarrow O/P:

Max(Marks)
60

Q: Retrieve names of all branches along with maximum marks in that branch.



Note:- If any attribute/attributes are present in select clause along with an aggregate function, then all those attributes must be present in group by clause as well.

Eg:

```
Select Branch, Max(Marks)  
From Student  
Group by (Branch)
```

← it is a valid query

Note:-

If any attribute/attributes are present in group by clause, and an aggregate function is specified with select clause, then the attributes present in group by clause need not be specified in select clause along with aggregate function.

Eg:

```
Select Max(Marks)  
From Student  
Group by (Branch)
```

← it is a valid query

Note:- If any attribute/attributes are present in select clause along with an aggregate function, then all those attributes must be present in group by clause as well.

Eg.

```
Select Branch, Max(Marks)  
From Student  
Group by (Sname)
```

← it is an invalid query

Note:- If any attribute/attributes are present in select clause along with an aggregate function, then all those attributes must be present in group by clause as well.

Eg.

```
Select Branch, Max(Marks)  
From Student  
Group by (Sname,Branch)
```

← it is a
Valid query



Topic : SQL clauses



NOTE:-

1. We can not select any attribute in SELECT clause along with aggregate function if those attributes are not present in GROUP BY clause.
2. If aggregate function is used along with GROUP BY clause, then aggregate function is applied on each group.

{ But if "group by" Clause is not present,
then Aggregate function will be applied on Complete
table }

Q: Retrieve names of all branches along with maximum marks in that branch.

Correct query:

```
Select Branch , Max(Marks)  
From Student  
Group by (Branch)
```



Topic : SQL clauses



GROUP BY:- GROUP BY clause is used to group the result of WHERE clause.

Query: Retrieve names of all branches along with maximum marks in that branch.

- ③ Select Branch, Max(Marks)
- ① From Student
- ② Group by (Branch)

① Student

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC

Branch	Max(Marks)
CS	60
IT	40
EC	60



Topic : SQL clauses

P
W

HAVING:- HAVING condition is applied on each group.

Query: Retrieve branch names with average marks more than or equal to 40.

O/P =

Branch
CS
EC

 Select Branch
 From Student
 Group by (Branch)
 Having (Avg(Marks) ≥ 40)

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC

Avg(Marks)
 50 ≥ 40 30 ≥ 40 60 ≥ 40
 ✗ ✗ ✓

discarded by
Aggregate fn



Topic : SQL clauses



ORDER BY:- This clause is used to sort the result in ascending or descending order based on values of attribute specified with ORDER BY clause.

By default order is ascending order.

Otherwise → for Ascending order : ASC
→ for Descending order : DESC

Student

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC

► Select Sid, Sname
From Student
Where (Marks ≥ 40) $\Rightarrow O/P$

Sid	Sname
S ₁	A
S ₃	B
S ₄	A
S ₅	C

Select Sid, Sname
From Student $\Rightarrow O/P =$
Where (Marks ≥ 40)
Order by (Sname) /ASC
By default
Ascending order

Sid	Sname
S ₁	A
S ₄	A
S ₃	B
S ₅	C

Select Sid, Sname
From Student $\Rightarrow O/P =$
Where (Marks ≥ 40)
Order by (Sname) DESC

Sid	Sname
S ₅	C
S ₃	B
S ₄	A
S ₁	A

* Retrieve Sids of all the Students
Who scored maximum Marks.

```
Select Sid
From Student
Where (Marks = Max(Marks))
```

We can not
use aggregate
function directly
within the Where Cond

Student			
Sid	Sname	Marks	Branch
S1	A	40	CS
S2	A	20	IT
S3	B	60	CS
S4	A	60	EC
S5	C	40	IT
S6	C	NULL	EC

* Retrieve Sids of all the students who scored maximum Marks

Select Sid
From Student
Main-query
(or)
Outer query

Where Marks = (Select Max(Marks)
From Student)

Inner query
(Sub-query)

query to get the
Max(Marks)
from the
Complete table



Topic : Nested queries



"Sub-query"

↳ query within query

Select Sid
From Student
Where (Marks = (Select Max (Marks)
From Student))

Main query
query within Main query
(i.e. Sub-query)

Nested Query

✓ Independent Nested query
(Independent Sub-query)

* If inner query can be executed independently, then it is called independent sub-query

✓ Correlated Nested query
(Correlated Sub-query)

→ When execution of inner query requires value of an attribute from the relation specified in Outer query, then it is called Correlated Sub-query

Nested Query

Independent Nested query (Independent Sub-query)

e.g. Select Sid
From Student

Main Query

Inner query

Where (Marks) = (Select Max(Marks)
From Student)

This query can
be executed
independently

∴ Independent
Sub-query

Correlated Nested query (Correlated Sub-query)

e.g: Select R.A
From R
Where ... (Select S.C
From S
Where (R.A=S.B))

Main Query

Inner query

Value of Attribute 'A' is required
from relation R (i.e., from the relation
specified in outer query)

∴ Correlated Sub-query

(1) Order of execution w.r.t. Independent Sub-query

- ④ Inner query will be executed first and it will produce its output,
- ④ then outer query will use the output produced by inner query for its execution.

* Retrieve Sids of all the Students

Who scored maximum Marks.

Select Sid
From Student

Where (Marks = (Select Max(Marks)
From Student))

O/P of this query = 60

And then
Our query will compare
the marks of each tuple
with "60"

Op :- Sid
S3
S4

Student

Sid	Sname	Marks	Branch
X S1	A	40	CS
X S2	A	20	IT
✓ S3	B	60	CS
✓ S4	A	60	EC
X S5	C	40	IT
X S6	C	NULL	EC

Correlated Sub-query :-

operand of this operator is also the o/p of inner query.

Main query
Select *
From R

Where operator

This operator
will return
True or False

Inner query
Select *
From S
Where ($R.A = S.B$)

- If operator returns true, then "Where" Cond is true
- If operator returns false, then 'Where' Cond is false

Order of Execution w.r.t. Correlated Subquery :-

- ⑥ — Select *
- ① — From R
- ⑤ — Where Operator

R A P Q

①	0		

S	A	B	X	Y
①	0			
②	0			

Select * —————— ④
 From S —————— ②
 Where ($R.A = S.B$) —————— ③

②, ③ & ④ will be executed on complete table 'S' for each tuple of relation 'R'. i.e. Relation in Outer query



Topic : Operators

* IN, ANY, ALL and EXISTS are the operators that are generally used with Sub-query Concept

IN operator

Operator "IN" is used to check whether the concerned tuple is a member of a set of tuples produced by the inner query or not?

Complement of 'IN' is 'NOT IN'

Check whether

X

IN

{ 2, 5, 8, 9, 12, 13 }

Our concerned
tuple from
Outer table

Set of tuples
Produced by
inner query

- for eg.

Select Eid
From EMP

(i) If $X = 8$, then operator 'IN' will return true

(ii) If $X = 10$, then operator 'IN' will return false

Check whether

X

IN

$\{(1, A), (3, D), (4, D), (7, F)\}$

"tuple from
Outer table"

for eg

Select Sid, Sname
From Students

(i) If $X = (4, D)$, then IN returns True

(ii) If $X = (F, 7)$, then IN returns false.

$\sqsubset \neq (7, F)$

Note :-

If set of tuples is empty {i.e., O/P produced by inner query is Empty}, then operator 'IN' will always return False

and hence operator 'NOT IN' will always return true if inner query result is Empty.

Q: Consider the following relational scheme

Supplier (Sid, Sname, Rating)

Parts (Pid, Pname, Color)

Catalog (Sid, Pid, Cost)

* Retrieve Sid of suppliers who supplied some Red

Color Parts

Part	Color
p1	Red
p2	Green
p3	Red

Catalog	
S1	P1
S1	P2
S2	P2
S3	P2
S3	P3

R.A:-

$\pi_{Catalog.Sid} \left(\begin{array}{l} \sigma_{Catalog.Pid = Parts.Pid} (Catalog \times Parts) \\ \wedge \\ Parts.Color = 'Red' \end{array} \right)$

SQL: (Without IN)

Select C.Sid

Catalog
renamed
as C

From Catalog C , Parts P
Where (C.Pid = P.Pid AND P.Color = 'Red')

SQL: (Using IN)

Select Sid From Catalog

Where Pid IN (Select Pid From Parts)
Where Color = 'Red'

Parts
renamed as P

'ANY' operator

Operator 'ANY' is used along with Comparison operators

<, ≤, >, ≥, =, <>

↑
Not equal

Check whether $x < \text{ANY}$

$x = 10$	x	x	x	x	✓
$x = 20$	x	x	x	x	x

{ 2, 5, 8, 9, 12, 14 }

If $x = 10$, then "ANY" will return True

If $x = 20$, then "ANY" will return False

'ANY' operator

Operator 'ANY' is used along with Comparison Operators

<, <=, >, >=, =, <>

Note: Operator 'ANY' will return True, if and only if at least one tuple in the set of tuples satisfy the Comparison Condition

Note: If set of tuples is empty, then Operator 'ANY' will always return False

'ALL' operator

Operator 'ALL' is also used along with Comparison Operators

<, <=, >, >=, =, <>

Check whether $x = 10$, $x < \text{ALL}$

$x = 10$, $x \in \{2, 5, 8, 9, 12, 14\}$

$x = 1$, $\underline{\quad \quad \quad \quad \quad \quad}$

If $x = 10$, then ALL returns False

If $x = 1$, then ALL returns True

'ALL' operator

Operator 'ALL' is also used along with Comparison Operators

<, <=, >, >=, =, <>

- * Operator 'ALL' can return False if and only if at least one tuple in the set of tuples fails the Comparison Condition

- * If set of tuples is Empty, then operator 'ALL' will always return True

EXISTS operator

- EXISTS returns True if and only if inner query result is not Empty.
 - If inner query result is not Empty, then EXISTS will return True &
 - If inner query result is Empty, then EXISTS will return False.

Complement of 'EXISTS' is 'NOT EXISTS'

H.W.

#e.g.

```
SELECT C.sid
FROM Catalog C
```

```
WHERE EXISTS (SELECT *
FROM Parts P
```

WHERE (P.Pid=C.Pid AND P.color='RED'))

Correlated
Sub-query

Parts	
Pid	Color
P ₁	Red
P ₂	Green
P ₃	Red

Catalog	
Sid	Pid
S ₁	P ₁
S ₁	P ₂
S ₂	P ₂
S ₃	P ₂
S ₃	P ₃

What output will be produced by the above query if it is executed on the given instances of Parts & Catalog table

H.W.
#e.g.

```
SELECT C1.sid  
FROM Catalog C1  
WHERE NOT EXISTS ( SELECT P.Pid
```

Catalog

Sid	Pid
S ₁	P ₁
S ₁	P ₂
S ₂	P ₂

Parts

Pid
P ₁
P ₂

FROM Parts P

```
WHERE NOT EXISTS (SELECT C2.Sid  
FROM Catalog C2  
WHERE(C2.Pid=P.Pid AND C2.Sid=C1.Sid)))
```

What output is produced by above SQL query:

- A. Sids of suppliers who supplied some parts
- B. Sids of suppliers who supplied only proper subset of parts from all parts
- C. Sids of suppliers who supplied all parts
- D. Sids of suppliers who did not supply any part



2 mins Summary



- Topic** SQL clauses
- Topic** Nested query (sub-query)
- Topic** Operators w.r.t. Nested query
- Topic** Order of execution of nested query

THANK - YOU