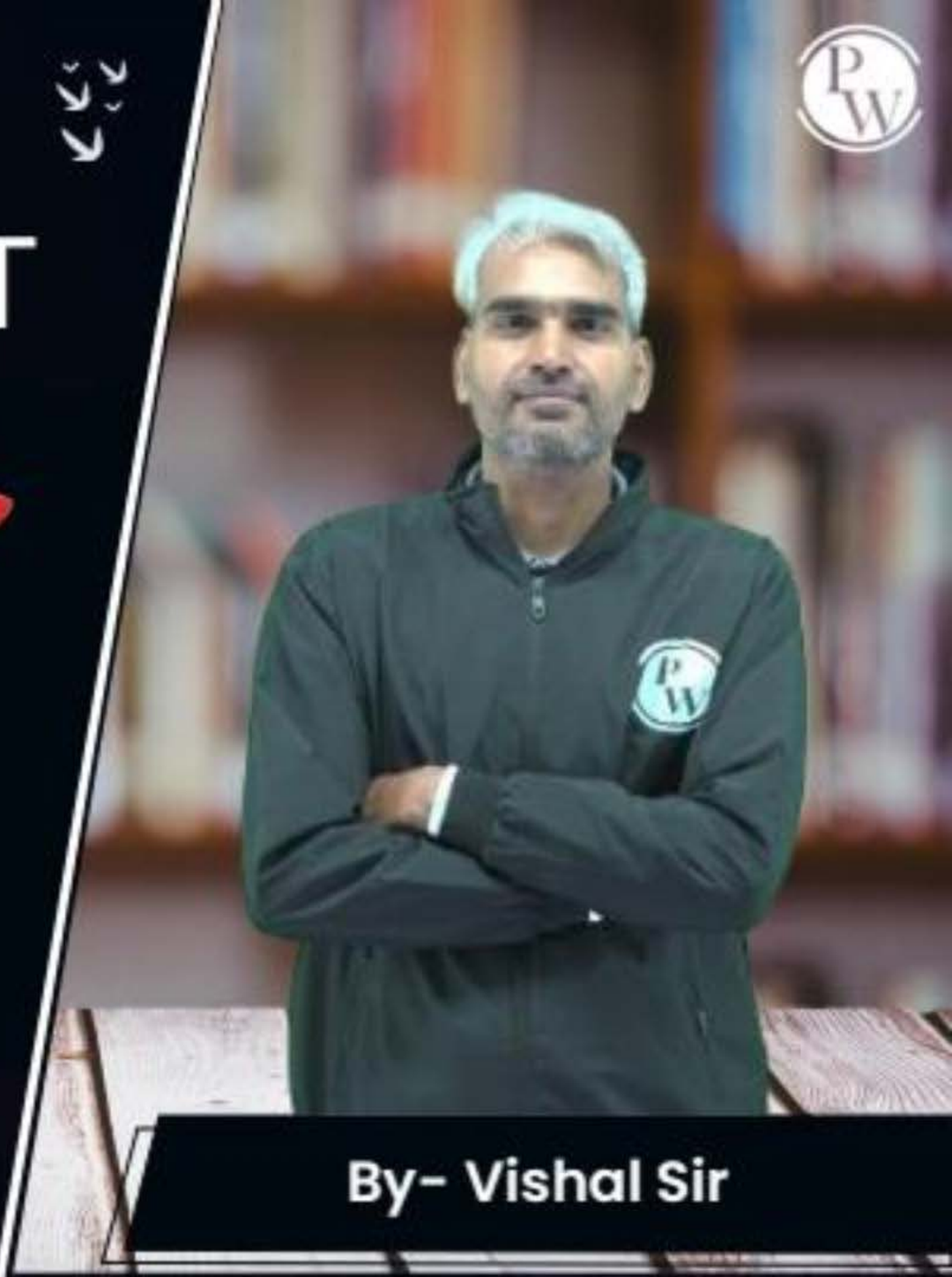
Computer Science & IT

Database Management System

Query Languages

Lecture No. 05



By- Vishal Sir

Recap of Previous Lecture

Topic: Practice questions

Slide

Topics to be Covered

- Topic: SQL
- Topic: SQL commands
- Topic: SQL clauses
- Topic: Aggregate functions

Slide

Supplier

Sid	Sname	Rating
S_1	A	3
S_2	A	5
S_3	B	7
S_4	C	0

Parts

Pid	Pname	Color
P_1	ABC	Red
P_2	XYZ	Gracen
P_3	KBC	Red

Catalog

Sid	Pid	Cost
S_1	P_1	20
S_1	P_2	30
S_2	P_2	30
S_3	P_2	20
S_3	P_3	10

#Q. Retrieve Sid of the suppliers who have supplied most expensive parts.

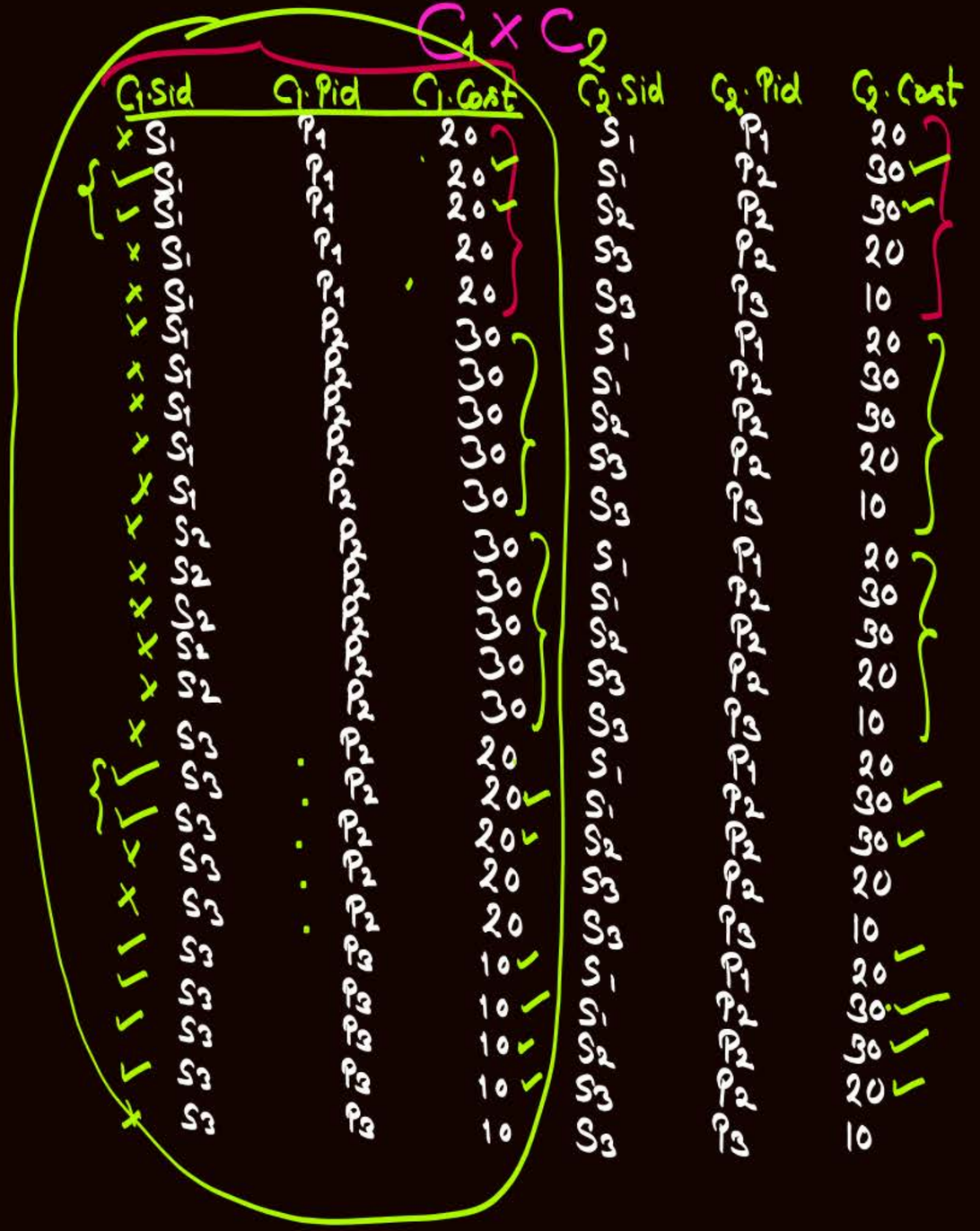
$$\Pi_{sid}\left[\left(\begin{array}{c}\text{All tuples}\\ \text{of}\\ \text{Catalog table}\end{array}\right) - \left(\begin{array}{c}\text{Tuples of the catalog table}\\ \text{in which Cost is not maximum}\end{array}\right)\right]$$

Sid of the suppliers who have supplied most expensive parts.

It will give all the tuples from the catalog table, in which Cost is maximum

Catalog

Sid	Pid	Cost
S_1	P_1	20
S_1	P_2	30
S_2	P_2	30
S_3	P_2	20
S_3	P_3	10



#Q. Retrieve Sid of the suppliers who have supplied most expensive parts.

$\pi_{Sid}\{(Catalog) - \pi_{C_1.Sid, C_1.Pid, C_1.Cost}(\sigma_{C_1.Cost < C_2.Cost}(C_1 \times C_2))\}$

S_1
S_2

Suppliers who have supplied most Expensive parts

Sid	Pid	Cost
S_1	P_1	20
S_1	P_2	30
S_2	P_2	30
S_3	P_2	20
S_3	P_3	10

Sid	Pid	Cost
S_1	P_1	20
S_3	P_2	20
S_3	P_3	10

We will get the tuples from catalog table in which Cost is not maximum

S_1 P_2 30
S_2 P_2 30

#Q. Retrieve Sid of the suppliers who have supplied most expensive parts.

$$\pi_{sid}(Catalog) - \pi_{sid}\left(\pi_{C_1.Sid,\, C_1.Pid,\, C_1.Cost}\left(\sigma_{C_1.Cost < C_2.Cost}(C_1 \times C_2)\right)\right)$$

it is wrong.

Topic : SQL

{ Prerequisite: Relational Algebra }

- Structured query language
  - It is non-procedural query language
    - We need to understand "Syntax"

Topic : Commands in SQL

Commands in SQL

- Create
  - Drop
  - Alter
  - → Data definition Language (DDL)
- Insert
  - Delete
  - Update
  - Truncate
  - → Data Manipulation Language (DML)
- Grant
  - Revoke
  - → Data Control Language (DCL)
- Commit
  - Rollback
  - → Transaction Control Language (TCL)
- Select (IMP.)
  - → Data Query language (DQL)

Topic : SQL clauses

Some Basic SQL Clauses are,

(i) From

(ii) Where

(iii) Group By

(iv) Having

(v) Order by

Topic : SQL clauses

"distinct" is optional.
If we do not use 'distinct', then SQL query output may contain duplicate tuples

$\pi_{A_1, A_2, \ldots A_m}$ ≡ Select [distinct] $A_1, A_2, \ldots A_m$ — list of attributes required in O/P

$R_1 \times R_2 \times \ldots \times R_n$ ≡ From $R_1, R_2, \ldots, R_n$ — Relations required to get the O/P

$\sigma_{(\text{Cond}^n \text{ to select tuples})}$ ≡ Where (Condition to select tuples) — This Cond^n will be applied on each tuple of relation.

No Equivalent op^n in Relational Algebra ≡

Group by (Attributes)

Having (Condition) — This Condition will be applied on each group. (Provided group by Clause is present)

Order by (Attribute) DESC/ASC

- Order of Execution

1. From
2. Where
3. Group by
4. Having
5. Select
6. Order by

Select of SQL $\neq$ Projection (π) of relational algebra

Select distinct of SQL $\equiv$ Projection (π) of R.A.

Topic : Aggregate functions

{ Aggregate functions are not available in Relational Algebra }

There are five aggregate functions in SQL

1. Count
2. Sum
3. Avg
4. Min
5. Max

Topic : Aggregate functions

- Count (*) = It will return the total number of tuples in the relation.

  {Note: Count (*) will also count the Empty tuple i.e. a tuple in which all values are NULL}

- Count (Attribute) : It will count the total number of non-NULL values in the column corresponding to the attribute specified with Count function

Note :-

1. NULL is a non-zero unknown value.
2. No two NULL are Equal.
3. NULL values are always discarded by aggregate functions. {Except for Count (*)}
4. Arithmatic operation with NULL value will always produce NULL. {i.e: NULL + 100 = NULL}

Topic : Aggregate functions

* Sum (Attribute) : It will return the summation of all non-NULL values of attribute specified with aggregate function SUM

→ Sum (distinct Attribute): It will return the summation of all distinct non-NULL values of attribute specified with aggregate function

Topic : Aggregate functions

* Avg (Attribute) : It will return the avg value w.r.t. specified attribute.

$$\text{Avg (Attribute)} = \frac{\text{Sum (attribute)}}{\text{Count (attribute)}}$$

- Avg (distinct Attribute) = $\frac{\text{sum (distinct attribute)}}{\text{Count (distinct attribute)}}$

Topic : Aggregate functions

- Min (attribute) : It will return the minimum of all the values corresponding to the specified attribute

- Max (attribute) : It will return the maximum of all the values corresponding to the specified attribute.

Student

Sid	Sname	Marks	Branch
S_1	A	40	CS
S_2	B	60	IT
S_3	A	20	CS
S_4	C	NULL	IT
S_5	B	60	EC
S_6	D	70	EC
NULL	NULL	NULL	NULL

* Select Count(*)
  From Student = 7

* Select Count(Sid)
  From Student = 6

* Select Count(Marks)
  From Student = 5

* Select Count distinct Marks
  From Student = 4

NULL will be discarded

* Select Sum (Marks)
  From Student = 250

* Select Sum (distinct marks)
  From Student = 190

Student

Sid	Sname	Marks	Branch
S_1	A	40	CS
S_2	B	60	IT
S_3	A	20	CS
S_4	C	NULL	IT
S_5	B	60	EC
S_6	D	70	EC
NULL	NULL	NULL	NULL

Student

Sid	Sname	Marks	Branch
S_1	A	40	CS
S_2	B	60	IT
S_3	A	20	CS
S_4	C	NULL	IT
S_5	B	60	EC
S_6	D	70	EC
NULL	NULL	NULL	NULL

→ Select Avg (Marks)
From Student

$$= \frac{\text{Sum (Marks)}}{\text{Count (Marks)}} = \frac{250}{5}$$

$$= 50$$

→ Select Avg (distinct marks)
From Student

$$= \frac{\text{Sum (distinct Marks)}}{\text{Count (distinct Marks)}} = \frac{190}{4} = 47.5$$

* Select max(Marks)
  From Student = 70

* Select Min(Marks)
  From Student = 20

Student

Sid	Sname	Marks	Branch
S_1	A	40	CS
S_2	B	60	IT
S_3	A	20	CS
S_4	C	NULL	IT
S_5	B	60	EC
S_6	D	70	EC
NULL	NULL	NULL	NULL

Topic : Commands in SQL

- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table

- INSERT - inserts new data into a database
- DELETE - deletes data from a database
- UPDATE - updates data in a database

- SELECT - extracts data from a database

Topic : Commands in SQL

- CREATE TABLE - creates a new table

```
CREATE TABLE Student (
        Sid varchar(25),
        Sname varchar(25),
        Marks int,
                    );
```

Topic : Commands in SQL

- CREATE TABLE - creates a new table

```
CREATE TABLE Student (
        Sid varchar(25),
        Sname varchar(25),
        Marks int,
                );
```

STUDENT

Sid	Sname	Marks

Topic : Commands in SQL

- ALTER TABLE - modifies a table

ALTER TABLE *Student*
ADD *Branch varchar(25);*

STUDENT

Sid	Sname	Marks

Topic : Commands in SQL

- ALTER TABLE - modifies a table

ALTER TABLE *Student*
ADD *Branch varchar(25);*

STUDENT

Sid	Sname	Marks	Branch

Topic : Commands in SQL

- DROP TABLE - deletes a table

DROP TABLE *Student*;

- TRUNCATE TABLE - deletes complete data from the table without deleting the structure of the table

TRUNCATE TABLE Student;

Topic : Commands in SQL

- INSERT - inserts new data into a database

- INSERT INTO *Student* VALUES (*S1*, *A*, *35*, CS);

STUDENT

Sid	Sname	Marks	Branch

Topic : Commands in SQL

- INSERT - inserts new data into a database

- INSERT INTO *Student* VALUES (*S1*, *A*, *35*, CS);

STUDENT

Sid	Sname	Marks	Branch
S1	A	35	CS

Topic : Commands in SQL

- DELETE - deletes data from a database

Topic : Commands in SQL

- UPDATE - updates data in a database

UPDATE *Student*
SET *Marks = Marks + 5;*

STUDENT

Sid	Sname	Marks	Branch
S1	A	35+5=40	CS

Topic : Commands in SQL

- UPDATE - updates data in a database

UPDATE *Student*
SET *Marks = Marks + 5;*

STUDENT

Sid	Sname	Marks	Branch
S1	A	35	CS

Topic : SQL clauses

- FROM - The FROM clause in SQL is used to select the database tables
- WHERE - The WHERE clause in SQL is used to retrieve the data from the database based on conditions specified with WHERE clause.
- GROUP BY - GROUP BY clause is used to group the result of WHERE clause.
- HAVING - HAVING clause can be used in a GROUP BY clause. It is used to specify a search condition for a group in the database tables.
- ORDER BY - The ORDER BY clause in SQL is used for sorting the records of the database

Topic : SQL clauses

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

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 retrieve the data from the database based on conditions specified with WHERE clause.

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.

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

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

NOTE:-

1. We can not select any attribute in SELECT clause along with aggregate function until those attributes are present in GROUP BY clause.

2. If aggregate function is used along with GROUP BY clause, then aggregate function is applied on each group.

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.

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

HAVING:- HAVING condition is applied on each group.

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

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

NOTE:-

1. WHERE condition is applied on each tuple whereas HAVING condition is applied on each group.
2. We can use HAVING condition without GROUP BY clause, but in that case HAVING condition will be applied on each tuple. i.e., without GROUP BY clause HAVING clause will degenerate into WHERE clause.

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.

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 : Order of execution

Order of Execution:-

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

2 mins Summary

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

THANK - YOU