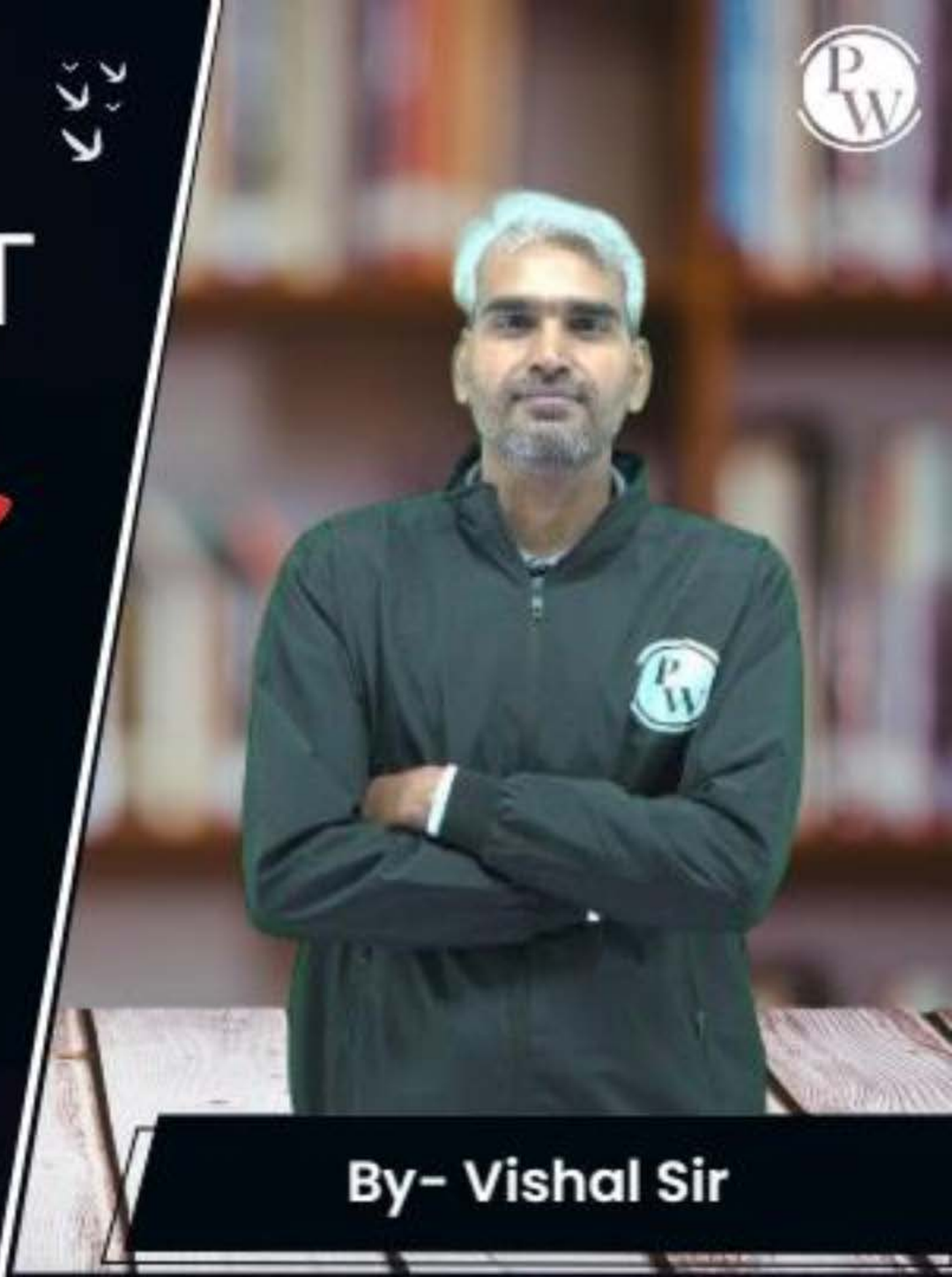
Computer Science & IT

Database Management System

Query Languages

Lecture No. 05



By- Vishal Sir

Recap of Previous Lecture

Topic Practice questions on relational algebra

Topics to be Covered

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

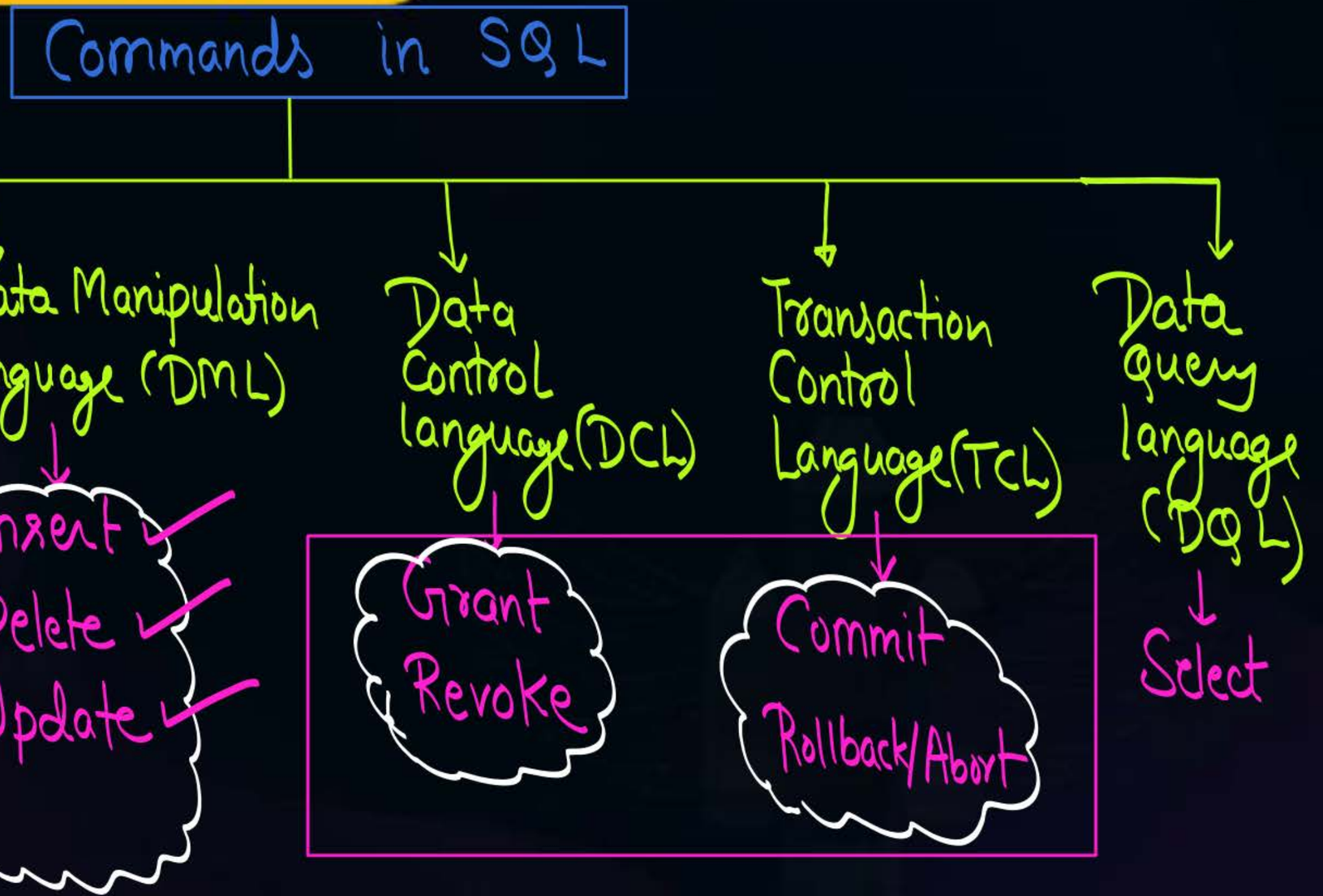
Topic : SQL

Prerequisite : Relational Algebra

Structured query language

- It is a non-procedural query lang.
  - ∴ We must understand the "Syntax" of SQL

Topic : Commands in SQL



Topic : SQL clauses

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

Basic Syntax of SQL Query :

It is optional :
If we do not use 'disinct' then SQL query may produce duplicate tuples

$\pi_{A_1, A_2, \ldots, A_k}$ ≡ ① Select distinct $A_1, A_2, \ldots, A_k$

list of attributes required in the O/P

$R_1 \times R_2 \times \ldots \times R_m$ ≡ ② From $R_1, R_2, R_3, \ldots, R_m$

list of all the tables required for the execution

$\sigma_{(\text{Cond}^n \text{ to select tuples})}$ ≡ ③ Where (Condition to select the tuples)

{ No Equivalent relational Algebra operation }

④ Group by (Attribute/s)

⑤ Having (Cond^n to select the group)

⑥ Order by (Attribute/s) ASC/DESC

Projection (π) in R.A. $\neq$ "Select" in SQL

because duplicate may be present in SQL

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

Topic : Aggregate functions

There are five different aggregate functions in SQL

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

Topic : Aggregate functions

- Select Count (*)
  From STUDENT=
- Select Count (Sid)
  From STUDENT =
- Select Count (distinct Sid)
  from STUDENT=
- Select Count (Marks)
  From STUDENT =
- Select Count (distinct Marks)
  From STUDENT=

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Empty tuple

Topic : Aggregate functions

All

- Count(*) - It returns the total number of tuples in the relation.

Count(*) will also count the empty tuples, i.e., the tuples in which all values all NULL.

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

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Topic : Aggregate functions

- Select Count (*) From STUDENT = 7
- Select Count (Sid) From STUDENT = 6
- Select Count (distinct Sid) from STUDENT = 6
- Select Count (Marks) From STUDENT = 5
- Select Count (distinct Marks) From STUDENT = 4

It will Count Non-NULL distinct Sids

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Topic : NOTE

- NULL is a non-zero unknown value.
- No two NULL are equal. { we can not Compare with NULL using normal Comarison operators }
- NULL values are always discarded by aggregate functions except for count(*).
- Arithmetic operation with NULL will always produce NULL

NULL + 50 = NULL

Topic : Aggregate functions

- Sum(Attribute) - It will return the summation of all non NULL values in the column corresponding to the attribute specified with aggregate function Sum().

- Sum(distinct Attribute) - It will return the summation of all distinct non NULL values in the column corresponding to the attribute specified with aggregate function Sum().

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Topic : Aggregate functions

- Select Sum (Marks) From STUDENT = 250

- Select Sum (distinct Marks) From STUDENT = 190

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

$$\text{Avg} = \frac{\text{Summation}}{\text{No. of terms}}$$

Topic : Aggregate functions

- Avg(Attribute) - It will return the average of all non NULL values with respect to specified attribute.

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

- Avg(distinct Attribute) - It will return average of all distinct non NULL values with respect to specified attribute.

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

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Topic : Aggregate functions

- Select Avg (Marks) From STUDENT = Sum(Marks) / Count(Marks)

$$= \frac{250}{5} = 50$$

- Select Avg (distinct Marks) From STUDENT = Sum(distinct Marks) / Count(distinct Marks)

$$= \frac{190}{4} = 47.5$$

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Topic : Aggregate functions

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

- Min(Attribute) - It will return the minimum of all the values in the column corresponding to the specified attribute.

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Topic : Aggregate functions

- Select Max (Marks) From STUDENT = 70

- Select Min (Marks) From STUDENT = 20

STUDENT

Sid	Sname	Marks	Branch
S1	A	40	CS
S2	B	60	IT
S3	A	20	CS
S4	C	NULL	IT
S5	B	60	EC
S6	D	70	EC
NULL	NULL	NULL	NULL

Topic : Commands in SQL

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

DML
- 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,
                );
```

Syntax:

```
Create table
    New_table_name
    ( Attribute1  data_type  Constraint,
      Attribute2  datatype   Constraint,
      :           :          :
                             );
```

"it is optional"

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

table_name

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

to add a new Column

Name of new attribute required

data_type

Student

Sid	Sname	Marks

* To add a new Column we use "ADD"
* To delete an existing Column we use "DROP"

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 { Complete table will be deleted from database }

DROP TABLE *Student*;

- TRUNCATE TABLE - deletes complete data (i.e. all records) from the table without deleting the structure of the table

TRUNCATE TABLE Student;

Topic : Commands in SQL

- INSERT - inserts new data(row/rows) into a database table.

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

Student

Sid	Sname	Marks	Branch

Topic : Commands in SQL

- INSERT - inserts new data(row/rows) into a database table.

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

Student

Sid	Sname	Marks	Branch
S1	A	35	CS

We can insert multiple rows using a single insert command. ⇒

```
Insert Into table_name
Values( (Att1, Att2, - - - Attk),    ← 1st row
        (Att1, Att2, - - - - Attk),  ← 2nd row
          :      :            :
       );
```

Topic : Commands in SQL

- DELETE - deletes data(row/rows) from a database table

Delete from Student
↘ Where (Marks = 30)

} ⇒ All the records from the Student table in which "Marks = 30" will be deleted

Topic : Commands in SQL

- UPDATE - updates data in a database table

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

Student

Sid	Sname	Marks	Branch
S1	A	35	CS

→ Updat table_name

Set Updation required

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 : SQL clauses

- **FROM** - The FROM clause in SQL is used to select the database tables

{All the tables specified along with From clause will be joined}

- **WHERE** - The WHERE clause in SQL is used to select the tuples from the database tables based on conditions specified with WHERE clause.

{where condition is applied on each tuple}

{If we do not use the "Where(Cond")" then all tuples will be selected}

- **GROUP BY** - GROUP BY clause is used to group the result of WHERE clause based on attributes specified with GROUP BY clause.

If "Where(Cond")" is not present, then "group by" will be applied on all tuples.

- **HAVING** - HAVING clause can be used with or without GROUP BY clause. If group by clause is present, then condition specified with having clause will be used to select the groups that satisfy the specified condition.

"It is debatable"

Generally, Having clause should be used when "Group by" is present

{① Having "Cond"" is applied on each group.
② If Having Cond" is not present then all the groups will be selected}

- **ORDER BY** - The ORDER BY clause in SQL is used for sorting the records in the O/p based on attributes specified with ORDER BY clause.

Topic : Order of execution

Order of Execution:-

1. From
2. Where
3. Group By
4. Having
5. Select
   - distinct (optional)
6. Order BY

2 mins Summary

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

Slide

THANK - YOU

Slide