Video-12

Topics to cover:

- Subquery in SQL

# Subquery

A subquery in SQL is a query nested inside another query. It is often called an inner query, while the main query is the outer query.

Subqueries allow you to use the result of one query as input to another, making your SQL more powerful and flexible.

Syntax:

```
SELECT column_list
FROM table
WHERE expression operator (SELECT column_list FROM table WHERE condition);
```

- **Outer query** → the main query that uses the result of the subquery.
- **Inner query (subquery)** → the query inside parentheses, executed first.
- **Operator** → defines how the outer query compares with the subquery result (IN, NOT IN, ANY, ALL, EXISTS, etc.).

# Creating employees table to understand different Commands

```
CREATE TABLE employees (
    emp_id     INT PRIMARY KEY,
    name       VARCHAR(50),
    department VARCHAR(50),
    salary     INT
);

-- Sample data
INSERT INTO employees VALUES
(1, 'Amit', 'HR', 40000),
(2, 'Priya', 'IT', 60000),
(3, 'John', 'Finance', 55000),
(4, 'Emma', 'IT', 70000),
(5, 'Rahul', 'HR', 30000),
(6, 'Neha', 'Finance', 65000);

SELECT * FROM employees;
```

-- Find employees who are getting more salary then Priya:

-- Step-1

```sql
SELECT salary
FROM employees
WHERE name = "Priya";
```

-- Step-2

```sql
SELECT *
FROM employees
WHERE salary > 60000;
```

-- Combine step

```sql
SELECT *
FROM employees
WHERE salary > (SELECT salary FROM employees WHERE name = "Priya");
```

# IN: The IN operator allows you to specify multiple values in a WHERE clause. It returns true if a value matches any value in a list.

```
-- Find employees who are in HR or IT department
SELECT *
FROM employees
WHERE department IN ("HR" , "IT");
```

# NOT IN: The NOT IN operator excludes the values in the list. It returns true if a value does not match any value in the list.

```
-- Find employees who are not in HR or IT department
SELECT *
FROM employees
WHERE department NOT IN ("HR" , "IT");
```

-- One more example of IN : Find employees in the same department as Priya:

```
SELECT *
FROM employees
WHERE department IN (SELECT department FROM employees WHERE name = 'Priya');
```

-- One more example of NOT IN : Find employees not in the same department as Priya:

```
SELECT *
FROM employees
WHERE department NOT IN (SELECT department FROM employees WHERE name = 'Priya');
```

# ANY: The ANY operator returns true if any subquery value meets the condition.

-- Find employees earning more than any HR employee:

```
SELECT *
FROM employees
WHERE salary > ANY (SELECT salary FROM employees WHERE department = 'HR');
```

-- note: Equivalent to “greater than the minimum HR salary.”

# ALL: The ALL operator returns true if all subquery value meets the condition.

-- Find employees earning more than all HR employees:

```
SELECT * FROM employees

WHERE salary > ALL (SELECT salary FROM employees WHERE department = 'HR');
```

# EXISTS: The EXISTS operator returns true if the subquery returns one or more records.

-- Find employees who belong to a department that has at least one employee earning above 60,000:

```
SELECT DISTINCT e.department

FROM employees e

WHERE EXISTS (SELECT 1 FROM employees WHERE department = e.department AND salary > 60000);
```

# NOT EXISTS: The NOT EXISTS operator returns true if the subquery returns no records.

-- Find departments where no employee earns above 60,000:

```
SELECT DISTINCT e.department

FROM employees e

WHERE NOT EXISTS (SELECT 1 FROM employees WHERE department = e.department AND salary > 60000);
```