MySQL

Subquery, ALL, ANY, CO-Related Subquery, CTE

SELECT, WHERE ,AND, OR, NOT ,ORDER BY,LIMIT, MIN(), MAX(), COUNT(), AVG(), SUM(), Wildcards, IN, BETWEEN, Aliases

Subquery

A **subquery** is a query written inside another SQL query (like SELECT, INSERT, UPDATE, or DELETE) to help fetch data based on the result of the inner query. It runs first and provides data to the outer query. Subqueries can return a single value, a single row, or multiple rows, and can be used in the WHERE, FROM, or SELECT clauses. When a subquery depends on values from the outer query, it is called a **correlated subquery**.

Ex:-

```
SELECT name, salary
FROM Employees
WHERE salary > (SELECT AVG(salary) FROM Employees);
```

ANY and ALL Operators

ANY Operator:

- ANY returns **true** if **any one value** from the subquery satisfies the condition.
- Think of it as: **"At least one match is enough."**

Syntax:-

```
SELECT column_name
FROM table_name
WHERE column_name operator ANY (subquery);
```

EX:-

```
SELECT name
FROM Employees
WHERE salary > ANY (SELECT salary FROM Employees WHERE department = 'HR');
```

ANY and ALL Operators

2. ALL Operator:

- ALL returns **true** only if **all values** from the subquery satisfy the condition.
- Think of it as: **"Every value must match the condition."**

Syntax:-

```
SELECT column_name
FROM table_name
WHERE column_name operator ALL (subquery);
```

Example:-

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

Correlated Subquery

A **correlated subquery** is a subquery that **depends on the outer query** for its value. It runs **once for every row** selected by the outer query.

Example:-

```
SELECT name, salary
FROM Employees E1
WHERE salary > (
    SELECT AVG(salary)
    FROM Employees E2
    WHERE E1.department = E2.department
);
```

CTE (Common Table Expression)

A **Common Table Expression (CTE)** is a temporary named result set created using the WITH keyword, used to simplify complex SQL queries by breaking them into readable parts; it can be used in SELECT, INSERT, UPDATE, and DELETE statements, supports recursion, and exists only for the duration of the query in which it is defined.

Syntax:-

```
WITH cte_name AS (
    SELECT column1, column2
    FROM table_name
    WHERE condition
)
SELECT *
FROM cte_name
WHERE another_condition;
```

Example:-

```
WITH HighEarners AS (
    SELECT name, salary
    FROM Employees
    WHERE salary > 50000
)
SELECT *
FROM HighEarners
WHERE name LIKE 'A%';
```