Subquery

A subquery in MySQL is a query, which is nested into another SQL query and embedded with SELECT, INSERT, UPDATE or DELETE (DML) statement along with the various operators. We can also nest the subquery with another subquery. A subquery is known as the inner query, and the query that contains subquery is known as the outer query. The inner query executed first gives the result to the outer query, and then the main/outer query will be performed. MySQL allows us to use subquery anywhere, but it must be closed within parenthesis. All subquery forms and operations supported by the SQL standard will be supported in MySQL also.

Rules to use subqueries

- Subqueries should always use in parentheses.

- If the main query does not have multiple columns for subquery, then a subquery can have only one column in the SELECT command.

- We can use various comparison operators with the subquery, such as >, <, =, IN, ANY, SOME, and ALL. A multiple-row operator is very useful when the subquery returns more than one row.

- We cannot use the ORDER BY clause in a subquery, although it can be used inside the main query.

- If we use a subquery in a set function, it cannot be immediately enclosed in a set function.

Subquery Syntax

SELECT column_list (s) FROM table_name

WHERE column_name OPERATOR

(SELECT column_list (s) FROM table_name [WHERE])

MySQL Row Subqueries

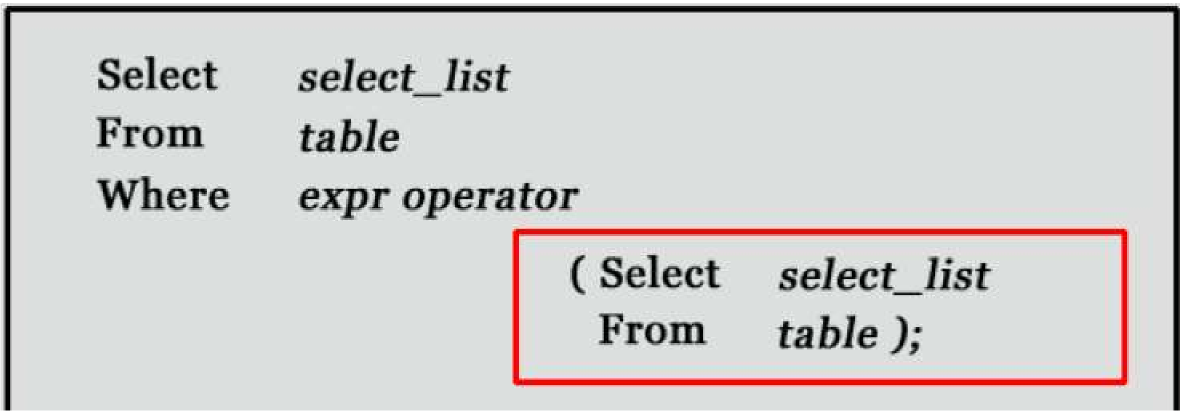
Single Row Subquery : An inner query which returns only one row called Single row subquery

Multi Row Subquery : Multiple-row subqueries are nested queries that can return more than one row of results to the parent query.

Types of Subqueries

- **Where Subquery**
  The most common type of subquery uses an inner SELECT subquery on the right side of a WHERE comparison expression. The subquery will be found in the WHERE clause. this type of query, when used in a >, <, =, >=,or <= conditional expression, requires a subquery that returns only one single value (one column, one row).

The value generated by the subquery must be of a “comparable” data type; if the attribute to the left of the comparison symbol is a character type, the subquery must return a character string.



```
Select    select_list
From      table
Where     expr operator
                          ( Select    select_list
                            From      table );
```

Example :

1. SELECT first_name, salary, department_id FROM employees WHERE salary = (SELECT MIN (salary) FROM employees);

2. Select first_name,last_name,salary from employees where salary > (select salary from employees where first_name = ‘Alok’);

3. SELECT emp_name, city, income FROM employees WHERE income = (SELECT MAX(income) FROM employees);

- **IN and NOT IN SubQueries**

  If the subquery produces more than one value, we need to use the IN or NOT IN operator with the WHERE clause. Suppose we have a table named "Student" and "Student2" that contains the following data:

SELECT emp_name FROM employee WHERE eno IN (SELECT eno FROM department WHERE dname = ‘finance’);

SELECT emp_name FROM employee WHERE eno NOT IN (SELECT eno FROM department WHERE dname = ‘finance’);

- **Having Sub Queries**

  A HAVING clause is used when the group results of a query need to be restricted based on some condition. If a subquery's result must be compared with a group function, you must nest the inner query in the outer query's HAVING clause.

SELECT department_id, MIN (salary) FROM employees GROUP BY department_id HAVING MIN (salary) < (SELECT AVG (salary) FROM employees)