**-- Subquery in SQL**

-------------------------------------------------------------------------------------------------

/\*

A subquery is a SQL query nested inside a larger query.

A subquery can be located in :

- A SELECT clause

- A FROM clause

- A WHERE clause

- A HAVING clause

The subquery can be nested inside a SELECT, INSERT, UPDATE, or DELETE statement or inside another subquery.

A subquery is usually added within the WHERE Clause of another SQL SELECT statement.

You can use the comparison operators, such as >, <, or =. The comparison operator can also be a multiple-row operator, such as IN, ANY, or ALL.

A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.

The inner query executes first before its parent query so that the results of an inner query can be passed to the outer query.

You can use a subquery in a SELECT, INSERT, DELETE, or UPDATE statement to perform the following tasks:

Compare an expression to the result of the query.

Determine if an expression is included in the results of the query.

Check whether the query selects any rows.

Syntax :

sql subquery syntax

The subquery (inner query) executes once before the main query (outer query) executes.

The main query (outer query) use the subquery result.

\*/

-- SQL Subqueries Locations :

-- In the SELECT Clause : Used to return a single value or a set of values.

SELECT first\_name, (SELECT distinct department\_name FROM departments where department\_name = 'Payroll') AS department\_name

FROM employees;

select \* from employees;

select \* from departments;

-- In the FROM Clause : Treated as a derived table or inline view.

SELECT e.first\_name

FROM (SELECT first\_name, salary FROM employees WHERE salary > 5000) as e;

-- In the WHERE Clause : Used to filter the results.

SELECT first\_name

FROM employees

WHERE department\_id IN (SELECT department\_id FROM departments WHERE location\_id>1500);

-- In the HAVING Clause : Used to filter groups.

SELECT department\_id, AVG(salary)

FROM employees

GROUP BY department\_id

HAVING AVG(salary) > (SELECT AVG(salary) FROM employees)

ORDER BY department\_id;

/\*

Subqueries: General Rules

A subquery SELECT statement is almost similar to the SELECT statement and it is used to begin a regular or outer query. Here is the syntax of a subquery:

Syntax:

(SELECT [DISTINCT] subquery\_select\_argument

FROM {table\_name | view\_name}

{table\_name | view\_name} ...

[WHERE search\_conditions]

[GROUP BY aggregate\_expression [, aggregate\_expression] ...]

[HAVING search\_conditions])

Subqueries: Guidelines

There are some guidelines to consider when using subqueries :

A subquery must be enclosed in parentheses.()

A subquery must be placed on the right side of the comparison operator.

Subqueries cannot manipulate their results internally, therefore ORDER BY clause cannot be added into a subquery. You can use an ORDER BY clause in the main SELECT statement (outer query) which will be the last clause.

Use single-row operators with single-row subqueries.

If a subquery (inner query) returns a null value to the outer query, the outer query will not return any rows when using certain comparison operators in a WHERE clause.

Type of Subqueries

Scalar Subquery: Returns a single value.

Column Subquery: Returns a single column of values

Multiple column subqueries : Returns one or more columns.

Single row subquery : Returns a single row of values.

Multiple row subquery : Returns one or more rows.

Table Subquery: Returns a result set that can be treated as a table

Correlated subqueries : Reference one or more columns in the outer SQL statement. The subquery is known as a correlated subquery because the subquery is related to the outer SQL statement.

Nested subqueries : Subqueries are placed within another subquery.

\*/

-- Scalar Subquery:

-- It returns a single value.

SELECT first\_name

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

-- Column Subquery:

-- Returns a single column of values.

SELECT first\_name

FROM employees

WHERE department\_id IN (SELECT department\_id FROM departments WHERE location\_id > 1500);

-- Single Row Subquery:

-- Returns a single row of values.

SELECT first\_name

FROM employees

WHERE salary = (SELECT MAX(salary) FROM employees);

-- Table Subquery:

-- Returns a result set that can be treated as a table.

SELECT first\_name,last\_name

FROM employees

WHERE department\_id IN (SELECT department\_id FROM departments);

select \* from employees;

select \* from departments;

-- Multi Column Subquery:

SELECT first\_name,last\_name,salary

FROM employees

WHERE (department\_id,manager\_id) IN (SELECT department\_id,manager\_id FROM departments);

-- Correlated vs. Non-Correlated Subqueries

-- Non-Correlated Subquery:

-- Independent and can be executed alone.

SELECT first\_name

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

-- Correlated Subquery:

-- Depends on the outer query and is executed for each row processed by the outer query.

SELECT first\_name,department\_id,salary

FROM employees e1

WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e1.department\_id = e2.department\_id);