**EX.NO:6 NESTED QUERIES**

**AIM:**

To study the various SQL nested queries operations on the database.

**DESCRIPTION:**

Nested query is one of the most useful functionalities of SQL. Nested queries are useful when we want to write complex queries where one query uses the result from another query. Nested queries will have multiple SELECT statements nested together. A SELECT statement nested within another SELECT statement is called a subquery.

**What is a Nested Query in SQL?**

A nested query in SQL contains a query inside another query. The result of the inner query will be used by the outer query. For instance, a nested query can have two **SELECT** statements, one on the inner query and the other on the outer query.

## What are the Types of Nested Queries in SQL?

Nested queries in SQL can be classified into two different types:

1. Independent Nested Queries
2. Co-related Nested Queries

### Independent Nested Queries

In independent nested queries, the execution order is from the innermost query to the outer query. An outer query won't be executed until its inner query completes its execution. The result of the inner query is used by the outer query. Operators such as **IN**, **NOT IN**, **ALL**, and **ANY** are used to write independent nested queries.

The **IN** operator checks if a column value in the outer query's result is **present** in the inner query's result. The final result will have rows that satisfy the **IN** condition.

The **NOT IN** operator checks if a column value in the outer query's result is **not present** in the inner query's result. The final result will have rows that satisfy the **NOT IN** condition.

The **ALL** operator compares a value of the outer query's result with **all the values** of the inner query's result and returns the row if it matches all the values.

The **ANY** operator compares a value of the outer query's result with all the inner query's result values and returns the row if there is a match with **any value**.

### Co-related Nested Queries

In co-related nested queries, the inner query uses the values from the outer query so that the inner query is executed for every row processed by the outer query. The co-related nested queries run slowly because the inner query is executed for every row of the outer query's result.

**How to Write Nested Query in SQL?**

We can write a nested query in SQL by nesting a **SELECT** statement within another **SELECT** statement. The outer **SELECT** statement uses the result of the inner **SELECT** statement for processing.

**The general syntax of nested queries will be:**

*SELECT column\_name [, column\_name ]*

*FROM table1 [, table2 ]*

*WHERE column\_name OPERATOR*

*( SELECT column\_name [, column\_name ]*

*FROM table1 [, table2 ]*

*[WHERE]*

*)*

The **SELECT** query inside the brackets (**()**) is the inner query, and the **SELECT** query outside the brackets is the outer query. The result of the inner query is used by the outer query.

**EXAMPLE:**

**TABLE #1 - employeedata**

*SQL> CREATE TABLE employeedata(id NUMBER PRIMARY KEY, name VARCHAR2(25) NOT NULL, salary NUMBER NOT NULL, role VARCHAR2(15) NOT NULL);*

Table created.

*SQL> INSERT INTO employeedata VALUES (1, 'Augustine Hammond', 10000, 'Developer');*

1 row created.

SQL> INSERT INTO employeedata VALUES (2, 'Perice John', 10000, 'Manager');

1 row created.

*SQL> INSERT INTO employeedata VALUES (3, 'Ragu Delafoy', 30000, 'Developer');*

1 row created.

*SQL> INSERT INTO employeedata VALUES (4, 'Teakwood Saffen', 40000, 'Manager');*

1 row created.

*SQL> INSERT INTO employeedata VALUES (5, 'Freddy Malcom', 50000, 'Developer');*

1 row created.

*SQL> select \* from employeedata;*

**OUTPUT:**

ID NAME SALARY ROLE

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

1 Augustine Hammond 10000 Developer

2 Perice John 10000 Manager

3 Ragu Delafoy 30000 Developer

4 Teakwood Saffen 40000 Manager

5 Freddy Malcom 50000 Developer

**TABLE #2 - awards**

*SQL>CREATE TABLE awards( id NUMBER PRIMARY KEY, employee\_id NUMBER NOT NULL, award\_date DATE NOT NULL );*

Table created.

*SQL> INSERT INTO awards VALUES(1, 1, TO\_DATE('2022-04-01', 'YYYY-MM-DD'));*

1 row created.

*SQL> INSERT INTO awards VALUES(2, 3, TO\_DATE('2022-05-01', 'YYYY-MM-DD'));*

1 row created.

*SQL> select \* from awards;*

**OUTPUT:**

ID EMPLOYEE\_ID AWARD\_DAT

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

1 1 01-APR-22

2 3 01-MAY-22

### Independent Nested Queries

### Example 1: IN

* Select all employees who won an award.

*SQL> SELECT id, name FROM employeedata WHERE id IN (SELECT employee\_id FROM awards);*

**OUTPUT:**

ID NAME

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

1 Augustine Hammond

3 Ragu Delafoy

### Example 2: NOT IN

* Select all employees who never won an award.

*SQL> SELECT id, name FROM employeedata WHERE id NOT IN (SELECT employee\_id FROM awards);*

**OUTPUT:**

ID NAME

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

2 Perice John

4 Teakwood Saffen

5 Freddy Malcom

**Example 3: ALL**

* Select all Developers who earn more than all the Managers

*SQL> SELECT \* FROM employeedata WHERE role = 'Developer' AND salary > ALL (SELECT salary FROM employeedata WHERE role = 'Manager');*

**OUTPUT:**

ID NAME SALARY ROLE

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

5 Freddy Malcom 50000 Developer

**Example 4: ANY**

* Select all Developers who earn more than any Manager

*SQL> SELECT \* FROM employeedata WHERE role = 'Developer' AND salary > ANY (SELECT salary FROM employeedata WHERE role = 'Manager');*

**OUTPUT:**

ID NAME SALARY ROLE

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

3 Ragu Delafoy 30000 Developer

5 Freddy Malcom 50000 Developer

**Co-related Nested Queries**

* Select all employees whose salary is above the average salary of employees in their role.

**Example:**

*SQL> SELECT \* FROM employeedata emp1 WHERE salary > (SELECT AVG(salary) FROM employeedata emp2 WHERE emp1.role = emp2.role);*

**OUTPUT:**

ID NAME SALARY ROLE

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

4 Teakwood Saffen 40000 Manager

5 Freddy Malcom 50000 Developer

**Explanation**

*The manager with id 4 earns more than the average salary of all managers (25000), and the developer with id 5 earns more than the average salary of all developers (30000). The inner query is executed for all rows fetched by the outer query. The role value (emp1.role) of every outer query's row is used by the inner query (emp1.role = emp2.role).*

* We can find the average salary of managers and developers using the below query:

*SQL> SELECT role, AVG(salary) FROM employeedata GROUP BY role;*

**OUTPUT:**

ROLE AVG(SALARY)

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

Developer 30000

Manager 25000

**RESULT:**

Thus the study the various SQL nested queries operations on the database is executed successfully.