**NESTED SUBQUERIES**

**EXPT NO: 6 DATE:** 31-10-2022

**AIM:** To study nested subqueries in SQL

**THEORY**

A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.

A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.

There are a few rules that subqueries must follow −

* Subqueries must be enclosed within parentheses.
* A subquery can have only one column in the SELECT clause, unless multiple columns are in the main query for the subquery to compare its selected columns.
* An ORDER BY command cannot be used in a subquery, although the main query can use an ORDER BY. The GROUP BY command can be used to perform the same function as the ORDER BY in a subquery.
* Subqueries that return more than one row can only be used with multiple value operators such as the IN operator.
* The SELECT list cannot include any references to values that evaluate to a BLOB, ARRAY, CLOB, or NCLOB.
* A subquery cannot be immediately enclosed in a set function.
* The BETWEEN operator cannot be used with a subquery. However, the BETWEEN operator can be used within the subquery.

In independent nested queries, query execution starts from innermost query to outermost queries. The execution of inner query is independent of outer query, but the result of inner query is used in execution of outer query. Various operators like IN, NOT IN, ANY, ALL etc are used in writing independent nested queries.

Using **C\_ID** of step 1 for finding **S\_ID**

Select **S\_ID** from **STUDENT\_COURSE** where **C\_ID** IN

(SELECT **C\_ID** from **COURSE** where **C\_NAME** = ‘DSA’ or **C\_NAME**=’DBMS’);

Select **S\_ID** from **STUDENT** where **S\_ID** NOT IN

(Select **S\_ID** from **STUDENT\_COURSE** where **C\_ID** IN

(SELECT **C\_ID** from **COURSE** where **C\_NAME**=’DSA’ or **C\_NAME**=’DBMS’));

Co-related Nested Queries: In co-related nested queries, the output of inner query depends on the row which is being currently executed in outer query. e.g.; If we want to find out S\_NAME of STUDENTs who are enrolled in C\_ID ‘C1’, it can be done with the help of co-related nested query as:

Select S\_NAME from STUDENT S where EXISTS

( select \* from STUDENT\_COURSE SC where S.S\_ID=SC.S\_ID and SC.C\_ID=’C1’);

**QUERIES**

1) Retrieve the publisher with maximum number of books published.

2) Which book has the maximum number of copies ?

3) Retrieve the book with maximum copies at ‘central’ branch.

4) Retrieve the books borrowed by people from ‘Farmagudi’

5) Find the author with maximum number of books written.

6) Find the library branch with minimum books(not the number of copies.)

7) List all the books written by the author of the book ‘The mystery’.

8) Find the borrower who has borrowed maximum number of books.

9) Find the borrower who has maximum number of books checked out (not yet

returned).

10) List the borrowers who have borrowed a book published by publisher with the same

address as the borrower.

11) Retrieve the names of the borrowers who have borrowed books from ‘panjim’;

branch but not from ‘farmagudi’ branch.

12) Retrieve the names of the borrowers who have borrowed books from ‘panjim’

branch as well as from’farmagudi’ branch.

13) Retrieve the details of the borrower who has borrowerd all books on databases.

**CONCLUSION**

The concept of Nested Subqueries was understood and various application scenarios were discussed.