Experiment – 8

Student Name: Garv Khurana	UID: 21BCS6615
Branch: BE CSE AIML	Section/Group: 21AML-9A
Semester: 03	Date of Performance: 1st Nov 2022
Subject Name: DBMS	Subject Code: 21CSH-243

AIM:- To implement the concept of Sub Query

What is subquery in SQL?

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

A subquery may occur in :

- A SELECT clause
- A FROM clause
- A WHERE 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 :

```
SELECT    select_list
FROM      table
WHERE     expr operator
                (SELECT    select_list
                FROM       table);
```

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

- The main query (outer query) use the subquery result. SQL Subqueries Example :

In this section, you will learn the requirements of using subqueries. We have the following two tables 'student' and 'marks' with common field 'StudentID'.

StudentID	Name
V001	Abe
V002	Abhay
V003	Acelin
V004	Adelphos

StudentID	Total_marks
V001	95
V002	80
V003	74
V004	81

student marks Now we want to write a query to identify all students who get better marks than that of the student who's StudentID is 'V002', but we do not know the marks of 'V002'.

-
-

To solve the problem, we require two queries. One query returns the marks (stored in Total_marks field) of 'V002' and a second query identifies the students who get better marks than the result of the first query.

First query:

```
SELECT *
FROM `marks`
WHERE studentid = 'V002';
```

Copy

Query result:

StudentID	Total_marks
V002	80

The result of the query is 80.

- Using the result of this query, here we have written another query to identify the students who get better marks than 80. Here is the query :

Second query:

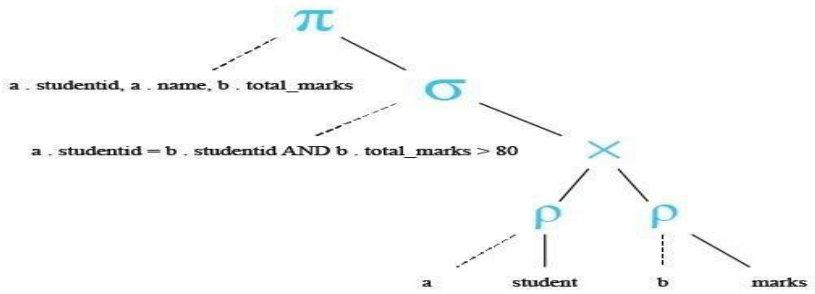
```
SELECT a.studentid, a.name, b.total_marks
FROM student a, marks b
WHERE a.studentid = b.studentid
AND b.total_marks >80;
```

Copy

Relational Algebra Expression:

$$\pi_{a.studentid,\ a.name,\ b.total\_marks}\ \sigma_{a.studentid = b.studentid\ AND\ b.total\_marks > 80}\ (\rho_{a}\ student \times \rho_{b}\ marks)$$

Relational Algebra Tree:



Query result:

studentid	name	total_marks
V001	Abe	95
V004	Adelphos	81

Above two queries identified students who get the better number than the student who's StudentID is 'V002' (Abhay).

You can combine the above two queries by placing one query inside the other. The subquery (also called the 'inner query') is the query inside the parentheses. See the following code and query result :

SQL Code:

```
SELECT a.studentid, a.name, b.total_marks
FROM student a, marks b
WHERE a.studentid = b.studentid AND b.total_marks >
(SELECT total_marks
FROM marks
WHERE studentid = 'V002');
```

Copy

Query result:

studentid	name	total_marks
V001	Abe	95
V004	Adelphos	81

Pictorial Presentation of SQL Subquery:
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
- Single row subquery : Returns zero or one row.
- Multiple row subquery : Returns one or more rows.
- Multiple column subqueries : Returns one or more columns.
- 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.

In the next session, we have thoroughly discussed the above topics. Apart from the above type of subqueries, you can use a subquery inside INSERT, UPDATE and DELETE statement. Here is a brief discussion:

Subqueries with INSERT statement

INSERT statement can be used with subqueries. Here are the syntax and an example of subqueries using INSERT statement.

Syntax:

```
INSERT INTO table_name [ (column1 [, column2 ]) ]
SELECT [ *|column1 [, column2 ]
FROM table1 [, table2 ]
[ WHERE VALUE OPERATOR ];
```

If we want to insert those orders from 'orders' table which have the advance_amount 2000 or 5000 into 'neworder' table the following SQL can be used:

Sample table: orders SQL
Code:

```
INSERT INTO neworder
SELECT * FROM orders
WHERE advance_amount in(2000,5000);
```

Output:

```
2 row(s) inserted.

0.71 seconds
```

Subqueries with UPDATE statement

In a UPDATE statement, you can set new column value equal to the result returned by a single row subquery. Here are the syntax and an example of subqueries using UPDATE statement.

Syntax:

```
UPDATE table SET column_name = new_value
[ WHERE OPERATOR [ VALUE ]
(SELECT COLUMN_NAME
FROM TABLE_NAME)      [ WHERE) ]
```

Output:-

```
    SQL Code: UPDATE neworder
SET ord_date='15-JAN-10'
WHERE ord_amount-advance_amount<
(SELECT MIN(ord_amount) FROM orders);
```

To see more details of subqueries using UPDATE statement click here. Subqueries with

DELETE statement

DELETE statement can be used with subqueries. Here are the syntax and an example of subqueries using DELETE statement.

Syntax:

```
DELETE FROM TABLE_NAME



[ WHERE OPERATOR [ VALUE ]
(SELECT COLUMN_NAME        FROM TABLE_NAME)  [ WHERE) ]
```

SQL Code:

```
DELETE FROM neworder
WHERE advance_amount<
(SELECT MAX(advance_amount) FROM orders);
```

Output:

```
34 row(s) deleted.

0.04 seconds
```

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			