

School of Computer Science, UPES, Dehradun.

## A

# LABORATORY FILE

On

# DATABASE MANAGEMENT SYSTEM (DBMS) LAB

B.TECH. -III Semester

# **Submitted by:**

Name: Mayank Negi SAP ID: 500120279 Roll No: R2142230039

Batch: 2

# **Experiment 06**

# To understand and use SQL Sub-Query

#### AIM:

To understand the use of SQL subquery.

#### **Problem Statement:**

- 1. Create the tables
- 2. Populate the tables with sample data
- 3. Perform some SQL Queries

#### **THEORY:**

Structured query language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values.

#### **COMMAND USED:**

- 1. DROP TABLE: Deletes a table and all its data.
- 2. PRIMARY KEY: Defines a column or set of columns as the unique identifier for rows in a table.
- 3. FOREIGN KEY: Establishes a relationship between columns in different tables.
- 4. ORDER BY: Sorts the result set by one or more columns.
- 5. WHERE: Filters records based on specific conditions.
- 6. GROUP BY: Groups rows that have the same values into summary rows.
- 7. HAVING: Filters records after the GROUP BY clause.
- 8. JOIN: Combines rows from two or more tables based on a related column.
- 9. CREATE INDEX: Creates an index on a table to speed up searches.

## **RESULTS:**

```
-- Ayush Vashishth
      -- 500119331
 2
 3
4 • CREATE DATABASE Supplier;
 5 • USE Supplier;
       -- Supplier Table
7 ● ⊖ CREATE TABLE Supplier (
      scode INT PRIMARY KEY,
      sname VARCHAR(50),
9
      scity VARCHAR(50),
10
       turnover DECIMAL(10, 2)
11
12
     - );
13
       -- Part Table
14 • G CREATE TABLE Part (
      pcode INT PRIMARY KEY,
15
      weigh DECIMAL(10, 2),
16
      color VARCHAR(20),
17
      cost DECIMAL(10, 2),
18
      sellingprice DECIMAL(10, 2)
19
20
      );
21
       -- Supplier_Part Table (Many-to-Many relationship)
22 • 

CREATE TABLE Supplier_Part (
       scode INT,
23
       pcode INT,
24
       qty INT,
25
       PRIMARY KEY (scode, pcode),
26
```

```
FOREIGN KEY (scode) REFERENCES Supplier(scode),
27
       FOREIGN KEY (pcode) REFERENCES Part(pcode)
28
       -- Insert data into Supplier
30
31 •
      INSERT INTO Supplier VALUES
       (1, 'Supplier1', 'Bombay', 50.00),
           'Supplier2', 'Delhi', 75.00),
33
       (2,
       (3, 'Supplier3', 'Bombay', NULL);
34
       -- Insert data into Part
36 •
      INSERT INTO Part VALUES
       (1, 30, 'Red', 20.00, 25.00),
37
       (2, 40, 'Blue', 30.00, 35.00),
       (3, 35, 'Green', 40.00, 50.00);
39
       -- Insert data into Supplier Part
41 • INSERT INTO Supplier Part VALUES
       (1, 1, 100),
42
       (1, 2, 200),
43
       (2, 3, 150),
45
      (3, 2, 120);
46
47 • SELECT scode, pcode
      FROM Supplier_Part
48
49
       ORDER BY scode ASC;
50
51 • SELECT *
       FROM Supplier
52
```

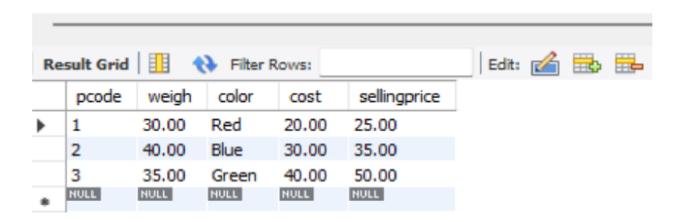
```
WHERE scity = 'Bombay' AND turnover = 50;
53
55 •
      SELECT COUNT(*) AS total_suppliers
       FROM Supplier;
56
57
58 • SELECT pcode
      FROM Part
       WHERE weigh BETWEEN 25 AND 35;
60
61
62 • SELECT scode
      FROM Supplier
63
      WHERE turnover IS NULL;
65
66 • SELECT pcode
67
      FROM Part
68
      WHERE cost IN (20, 30, 40);
70 • SELECT SUM(qty) AS total_quantity
71
      FROM Supplier_Part
72
      WHERE pcode = 2;
73
      SELECT sname
       FROM Supplier

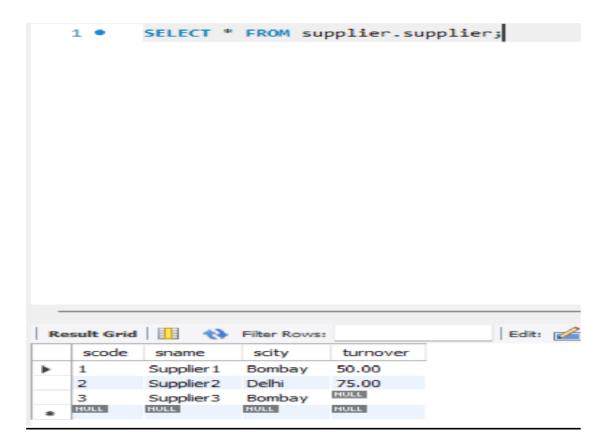
⊖ WHERE scode IN (SELECT scode)

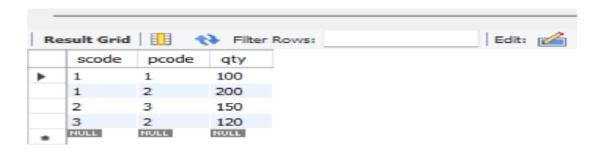
76
77
      FROM Supplier_Part
78
      WHERE pcode = 2);
```

```
80 • SELECT pcode
81 FROM Part
82 WHERE cost > (SELECT AVG(cost) FROM Part);
83
84 • SELECT scode, turnover
85 FROM Supplier
86 ORDER BY turnover DESC;
```

SELECT \* FROM supplier.part;







## **Conclusion:**

In this experiment, the use of SQL subqueries was explored to retrieve specific data from relational databases. Through the creation and population of tables, subqueries were used to filter and organize data based on various conditions, such as joins, groupings, and orderings. Subqueries enhance query flexibility by allowing complex data extraction within a single statement. This experiment provided hands-on experience with key SQL commands like DROP, PRIMARY KEY, FOREIGN KEY, ORDER BY, WHERE, GROUP BY, and JOIN, deepening the understanding of relational database management and data querying techniques.