

School of Computer Science, UPES, Dehradun.

A

LABORATORY FILE

On

DATABASE MANAGEMENT SYSTEM (DBMS) LAB

B.TECH. -III Semester

AUG. – NOV.- 2024.

Submitted by:

Name: Harsh Vardhan Saini SAP ID: 500120369 Roll No: R2142230056

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
 2
      -- 500119331
 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)
     );
12
13
       -- Part Table
14 ● ⊖ CREATE TABLE Part (
      pcode INT PRIMARY KEY,
15
16
      weigh DECIMAL(10, 2),
      color VARCHAR(20),
17
      cost DECIMAL(10, 2),
18
      sellingprice DECIMAL(10, 2)
19
20
      - );
       -- Supplier_Part Table (Many-to-Many relationship)
22 • 

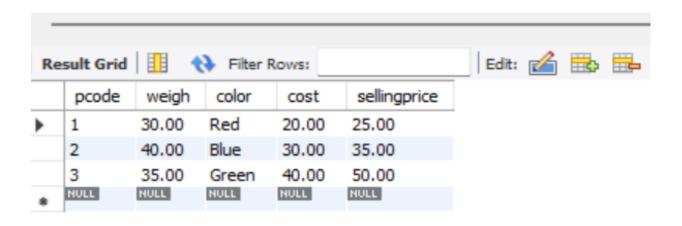
CREATE TABLE Supplier_Part (
      scode INT,
23
       pcode INT,
24
       qty INT,
25
26
       PRIMARY KEY (scode, pcode),
```

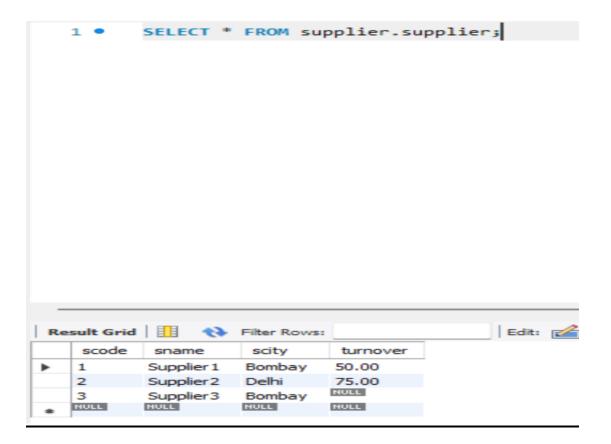
```
27
       FOREIGN KEY (scode) REFERENCES Supplier(scode),
       FOREIGN KEY (pcode) REFERENCES Part(pcode)
29
       -- Insert data into Supplier
30
       INSERT INTO Supplier VALUES
31 •
       (1, 'Supplier1', 'Bombay', 50.00),
32
       (2, 'Supplier2', 'Delhi', 75.00),
33
       (3, 'Supplier3', 'Bombay', NULL);
       -- Insert data into Part
35
36 • INSERT INTO Part VALUES
       (1, 30, 'Red', 20.00, 25.00),
       (2, 40, 'Blue', 30.00, 35.00),
38
       (3, 35, 'Green', 40.00, 50.00);
39
       -- Insert data into Supplier Part
41 • INSERT INTO Supplier_Part VALUES
       (1, 1, 100),
       (1, 2, 200),
44
       (2, 3, 150),
45
      (3, 2, 120);
47 • SELECT scode, pcode
      FROM Supplier_Part
       ORDER BY scode ASC;
50
51 • SELECT *
52
       FROM Supplier
```

```
WHERE scity = 'Bombay' AND turnover = 50;
53
54
       SELECT COUNT(*) AS total_suppliers
55 •
       FROM Supplier;
57
58
       SELECT pcode
      FROM Part
       WHERE weigh BETWEEN 25 AND 35;
61
62 • SELECT scode
63
      FROM Supplier
      WHERE turnover IS NULL;
64
65
      SELECT pcode
67
      FROM Part
      WHERE cost IN (20, 30, 40);
68
70 • SELECT SUM(qty) AS total_quantity
71
       FROM Supplier_Part
      WHERE pcode = 2;
72
73
74 •
      SELECT sname
75
       FROM Supplier
   WHERE scode IN (SELECT scode
       FROM Supplier_Part
77
78
       WHERE pcode = 2);
```

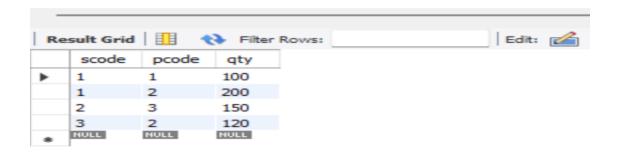
```
80 • SELECT pcode
81   FROM Part
82   WHERE cost > (SELECT AVG(cost) FROM Part);
83
84 • SELECT scode, turnover
85   FROM Supplier
86   ORDER BY turnover DESC;
```

1 • SELECT * FROM supplier.part;





5ELECT * FROM supplier.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.