|  |  |
| --- | --- |
| **Name** | **Hatim Yusuf Sawai** |
| **UID no.** | **2021300108** |
| **Experiment No.** | **6** |

|  |  |
| --- | --- |
| **AIM:** | **To implement Sub-Queries in sql** |
| **PROBLEM STATEMENT:** | Perform 4 complex & 6 simple subqueries on existing tables of the database |
| **THEORY:** | **SUBQUERIES**  Subquery or Inner query or Nested query is a query in a query. SQL subquery is usually added in the WHERE Clause of the SQL statement. Most of the time, a subquery is used when you know how to search for a value using a SELECT statement, but do not know the exact value in the database. Subqueries are an alternate way of returning data from multiple tables. Subqueries can be used with the following SQL statements along with the comparison operators like =, <, >, >=, <= etc.  **The following are the rules to use subqueries:**  1. Subqueries should always use in parentheses.  2. If the main query does not have multiple columns for subquery, then a subquery can have only one column in the SELECT command.  3. We can use various comparison operators with the subquery, such as >, <, =, IN, ANY, SOME, and ALL. A multiple-row operator is very useful when the subquery returns more than one row.  4. We cannot use the ORDER BY clause in a subquery, although it can be used inside the main query.  5. If we use a subquery in a set function, it cannot be immediately enclosed in a set function.  **The following are the advantages of using subqueries:**  1. The subqueries make the queries in a structured form that allows us to isolate each part of a statement.  2. The subqueries provide alternative ways to query the data from the table; otherwise, we need to use complex joins and unions.  3. The subqueries are more readable than complex join or union statements.  **MySQL Subquery Syntax**  The following is the basic syntax to use the subquery in MySQL:  **SELECT** column\_list (s) **FROM**  table\_name  **WHERE**  column\_name OPERATOR  (**SELECT** column\_list (s)  **FROM** table\_name [**WHERE**])  **MySQL Correlated Subqueries**  A correlated subquery in MySQL is a subquery that depends on the outer query. It uses the data from the outer query or contains a reference to a parent query that also appears in the outer query. MySQL evaluates it once from each row in the outer query. |
| **QUERIES:** | **Complex Queries:**  1. Show max salary for each location and for only those doctors who’s address match their patient’s address:  SELECT Address,MAX(Salary) FROM doctor  WHERE Address IN (SELECT Address FROM patient  WHERE Address=doctor.Address)  GROUP BY Address;  2. View Max ages of patients whose assigned doctors have salaries more than 50,000:  SELECT MAX(Age) FROM patient  WHERE D\_id IN (SELECT D\_id FROM doctor  WHERE Salary>50000)  GROUP BY D\_id;  3. Show all details of doctors with minimum & maximum salary:  SELECT \* FROM doctor  WHERE Salary=(SELECT MIN(Salary) FROM doctor)  OR Salary=(SELECT MAX(Salary) FROM doctor);  4. Show details of patient whose assigned doctor stays in Colaba:  SELECT \* FROM patient  WHERE D\_id IN (SELECT D\_id FROM doctor  WHERE Address="Colaba");  **Simple Queries:**  1. Show details of patient whose assigned doctor has minimum salary:  SELECT \* FROM patient  WHERE D\_id=(SELECT D\_id FROM doctor  WHERE Salary=(SELECT MAX(Salary) FROM doctor));  2. Display count of patients whose address is same as their assigned doctor:  SELECT COUNT(P\_id) FROM patient  WHERE D\_id IN (SELECT D\_id FROM doctor  WHERE Address=patient.Address);  3, show max average salary from average of all salaries grouped location wise:  SELECT MAX(Salary) FROM (SELECT Address,AVG(Salary) FROM doctor GROUP BY Address);  4. Get min salary of doctor whose address matches with his patient’s address:  SELECT Address,MIN(Salary) FROM doctor  WHERE Address IN (SELECT Address FROM patient  WHERE Address=doctor.Address)  GROUP BY Address; |
| **RESULT:**  **Doctor Table:**  **Patient Table:** | |
| **CONCLUSION:** | In this experiment, we learned how perform simple as well as complex subqueries on given data of the database. |