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; **Address** | **MAX(Salary)** Filter... | Filter... Marol | 500000 Andheri | 720000 Colaba | 450000 Bhayandar | 100000

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;
```

MAX(Age)
Filter...
65
30
35
40
45
50
60
55

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);
```

D_id	Dname	Ph_no	Salary	Field	Address
Filter.	Filter...	Filter...	Filter...	Filter...	Filter...
2	pramod	8965735643	720000	Neurologist	Andheri
10	Dhruv	5672356257	50000	Neurologist	Bhayandar

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");
```

P_id	Pname	Age	Address	Ph_no	D_id
Filt	Filter...	Filter	Filter...	Filter...	Filter
6	Virinchi	50	bhayandar	9876543210	7
8	Kaif	60	Bandra	9876543210	8
7	Udit	55	Dahisar	9876543210	11

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));
```

P_id	Pname	Age	Address	Ph_no	D_id
Filter...	Filter...	Filte	Filter...	Filter...	Filter...
2	Raj	30	Parel	9876543210	2

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);
```

COUNT(P_id)
Filter...
2

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;
```

Address	MIN(Salary)
Filter...	Filter...
Marol	100000
Andheri	200000
Colaba	320000
Bhayandar	50000

RESULT:

Doctor Table:

D_id	Dname	Ph_no	Salary	Field	Address
Filte	Filter...	Filter...	Filter..	Filter...	Filter...
1	akash	5748364582	500000	Cardiologist	Marol
2	pramod	8965735643	720000	Neurologist	Andheri
3	hansraj	6758392011	200000	Orthopedic	Andheri
4	ritu	9876567814	350000	dermatologist	Marol
5	viraj	7898657788	100000	dentist	Marol
6	rohit	9956443218	560000	ophthalmologist	Andheri
7	Iyer	9887854563	320000	gynecologist	Colaba
8	sachin	9876543210	450000	pediatrician	Colaba
9	sagar	9876543210	450000	pediatrician	Andheri
10	Dhruv	5672356257	50000	Neurologist	Bhayandar
11	Kaif	9348569346	400000	dentist	Colaba
12	Virinchi	9348569346	100000	dentist	Bhayandar

Patient Table:

P_id	Pname	Age	Address	Ph_no	D_id
Filte	Filter..	Filt	Filter...	Filter...	Filte
1	Rahul	25	Andheri	9876543210	1
2	Raj	30	Parel	9876543210	2
3	Pranay	35	Colaba	9876543210	3
4	Dev	40	Santacruz	9876543210	4
5	Hatim	45	Marol	9876543210	5
6	Virinchi	50	bhayandar	9876543210	7
7	Udit	55	Dahisar	9876543210	11
8	Kaif	60	Bandra	9876543210	8
9	Anish	65	Borivali	9876543210	1
10	Husain	21	Marol	1234567890	4

CONCLUSION:	In this experiment, we learned how perform simple as well as complex subqueries on given data of the database.