

Experiment No: 7

Title : Study & Implementation of

- Sub queries
- Views

Objective:

- To perform nested Queries and joining Queries using DML command
- To understand the implementation of views.

LAB PRACTICE ASSIGNMENT:

Consider the following schema:

Sailors (sid, sname, rating, age)

Boats (bid, bname, color)

Reserves (sid, bid, day(date))

Write subquery statement for the following queries.

1. Find all information of sailors who have reserved boat number 101.

```
mysql> SELECT *
-> FROM sailors
-> WHERE sid IN(SELECT sid FROM Reserves WHERE bid=101);
```

sid	sname	rating	age
1	Bob	5	25
2	Alice	3	30
3	Charlie	5	22

2. Find the name of boat reserved by Bob.

```
mysql> SELECT bname
-> FROM Boats
-> WHERE bid IN (SELECT bid FROM Reserves WHERE sid = (SELECT sid FROM Sailors WHERE sname = 'Bob'));
```

bname
Dinghy
Sailboat
Yacht

3. Find the names of sailors who have reserved a red boat, and list in the order of age.

```
mysql> SELECT sname
-> FROM Sailors
-> WHERE sid IN (SELECT sid FROM Reserves WHERE bid IN (SELECT bid FROM Boats WHERE color = 'red'))
-> ORDER BY age;
+-----+
| sname |
+-----+
| Charlie |
| Bob    |
| David  |
| Alice  |
+-----+
```

4. Find the names of sailors who have reserved at least one boat

```
mysql> SELECT sname
-> FROM Sailors
-> WHERE sid IN (SELECT DISTINCT sid FROM Reserves);
+-----+
| sname |
+-----+
| Bob   |
| Alice |
| Charlie |
| David |
| Eve   |
+-----+
```

5. Find the ids and names of sailors who have reserved two different boats on the same day.

```
mysql> SELECT sid, sname
-> FROM Sailors
-> WHERE sid IN (
-> SELECT sid
-> FROM Reserves
-> GROUP BY sid, day
-> HAVING COUNT(DISTINCT bid) >= 2
-> );
+----+-----+
| sid | sname |
+----+-----+
| 1   | Bob   |
| 2   | Alice |
+----+-----+
```

6. Find the ids of sailors who have reserved a red boat or a green boat.

```
mysql> SELECT DISTINCT sid
-> FROM Reserves
-> WHERE bid IN (SELECT bid FROM Boats WHERE color IN ('red', 'green'));
+----+
| sid |
+----+
| 1   |
| 2   |
| 3   |
| 4   |
+----+
```

7. Find the name and the age of the youngest sailor.

```
mysql> SELECT sname, age
-> FROM Sailors
-> WHERE age = (SELECT MIN(age) FROM Sailors);
+-----+-----+
| sname | age |
+-----+-----+
| Charlie | 22 |
+-----+-----+
1 row in set (0.01 sec)
```

8. Count the number of different sailor names.

```
mysql> SELECT COUNT(DISTINCT sname)
-> FROM Sailors;
+-----+
| COUNT(DISTINCT sname) |
+-----+
| 5 |
+-----+
```

9. Find the average age of sailors for each rating level.

```
mysql> SELECT rating, AVG(age)
-> FROM Sailors
-> GROUP BY rating;
+-----+-----+
| rating | AVG(age) |
+-----+-----+
| 5 | 23.5000 |
| 3 | 30.0000 |
| 4 | 28.0000 |
| 2 | 35.0000 |
+-----+-----+
```

10. Find the average age of sailors for each rating level that has at least two sailors.

```
mysql> SELECT rating, AVG(age)
-> FROM Sailors
-> GROUP BY rating
-> HAVING COUNT(sid) >= 2;
+-----+-----+
| rating | AVG(age) |
+-----+-----+
| 5 | 23.5000 |
+-----+-----+
```