**Experiment No.5**

**Lab Practice Assignment**

**Consider the following schema:**

**Sailors (sid, sname, rating, age) Boats (bid, bname, color) Reserves (sid, bid, day(date))**

1. **Creating tables**

mysql> use Jayabrota;

Database changed

mysql> CREATE TABLE Sailors (sid INT PRIMARY KEY, sname VARCHAR(100), rating INT, age INT);

Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Boats (bid INT PRIMARY KEY, bname VARCHAR(100), color VARCHAR(50));

Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Reserves (sid INT, bid INT, day DATE, PRIMARY KEY (sid, bid, day), FOREIGN KEY (sid) REFERENCES Sailors(sid), FOREIGN KEY (bid) REFERENCES Boats(bid));

Query OK, 0 rows affected (0.07 sec)

1. **Inserting values**

mysql> INSERT INTO Sailors (sid, sname, rating, age) VALUES (1, ‘Bob’, 10, 25), (2, ‘Alice’, 5, 30), (3, ‘Charlie’, 7, 22), (4, ‘Diana’, 10, 28);

Query OK, 4 rows affected (0.02 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Boats (bid, bname, color) VALUES (101, ‘Boat1’, ‘red’), (102, ‘Boat2’, ‘blue’), (103, ‘Boat3’, ‘green’), (104, ‘Boat4’, ‘red’);

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Reserves (sid, bid, day) VALUES (1, 101, ‘2024-08-01’), (1, 102, ‘2024-08-01’), (2, 103, ‘2024-08-02’), (3, 101, ‘2024-08-03’), (3, 104, ‘2024-08-03'), (4, 104, '2024-08-04');

Query OK, 6 rows affected (0.01 sec)

Records: 6 Duplicates: 0 Warnings: 0

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

mysql> SELECT S.\* FROM Sailors S JOIN Reserves R ON S.sid = R.sid WHERE R.bid = 101;

+-----+---------+--------+------+

| sid | sname | rating | age |

+-----+---------+--------+------+

| 1 | Bob | 10 | 25 |

| 3 | Charlie | 7 | 22 |

+-----+---------+--------+------+

2 rows in set (0.01 sec)

1. **Find the name of boat reserved by Bob.**

mysql> SELECT B.bname FROM Boats B JOIN Reserves R ON B.bid = R.bid JOIN Sailors S ON R.sid = S.sid WHERE S.sname = 'Bob';

+-------+

| bname |

+-------+

| Boat1 |

| Boat2 |

+-------+

2 rows in set (0.00 sec)

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

mysql> SELECT sname FROM (SELECT S.sname, S.age FROM Sailors S JOIN Reserves R ON S.sid = R.sid JOIN Boats B ON R.bid = B.bid WHERE B.color = 'red' ORDER BY S.age) AS subquery GROUP BY sname;

+---------+

| sname |

+---------+

| Bob |

| Charlie |

| Diana |

+---------+

3 rows in set (0.00 sec)

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

mysql> SELECT DISTINCT S.sname FROM Sailors S JOIN Reserves R ON S.sid = R.sid;

+---------+

| sname |

+---------+

| Bob |

| Alice |

| Charlie |

| Diana |

+---------+

4 rows in set (0.00 sec)

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

mysql> SELECT R1.sid, S.sname FROM Reserves R1 JOIN Reserves R2 ON R1.sid = R2.sid AND R1.day = R2.day AND R1.bid <> R2.bid JOIN Sailors S ON R1.sid = S.sid GROUP BY R1.sid, S.sname;

+-----+---------+

| sid | sname |

+-----+---------+

| 1 | Bob |

| 3 | Charlie |

+-----+---------+

2 rows in set (0.01 sec)

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

mysql> SELECT DISTINCT R.sid FROM Reserves R JOIN Boats B ON R.bid = B.bid WHERE B.color IN ('red', 'green');

+-----+

| sid |

+-----+

| 1 |

| 3 |

| 2 |

| 4 |

+-----+

4 rows in set (0.00 sec)

1. **Find the name and the age of the youngest sailor.**

mysql> SELECT sname, age FROM Sailors ORDER BY age ASC LIMIT 1;

+---------+------+

| sname | age |

+---------+------+

| Charlie | 22 |

+---------+------+

1 row in set (0.01 sec)

1. **Count the number of different sailor names.**

mysql> SELECT COUNT(DISTINCT sname) AS different\_sailor\_names FROM Sailors;

+------------------------+

| different\_sailor\_names |

+------------------------+

| 4 |

+------------------------+

1 row in set (0.01 sec)

1. **Find the average age of sailors for each rating level.**

mysql> SELECT rating, AVG(age) AS average\_age FROM Sailors GROUP BY rating;

+--------+-------------+

| rating | average\_age |

+--------+-------------+

| 10 | 26.5000 |

| 5 | 30.0000 |

| 7 | 22.0000 |

+--------+-------------+

3 rows in set (0.01 sec)

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

mysql> SELECT rating, AVG(age) AS average\_age FROM Sailors GROUP BY rating HAVING COUNT(\*) >= 2;

+--------+-------------+

| rating | average\_age |

+--------+-------------+

| 10 | 26.5000 |

+--------+-------------+

1 row in set (0.00 sec)