Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

LAB NO 3

JOIN OPERATIONS IN SQL

Objective:

- To learn and apply different types of Join Operations in SQL.
- To understand and implement foreign key.

LAB TASKS

1. Consider the following schema:

Sailors (sid(PK), sname, rating, age)
Boats (bid(PK), bname, color)
Reserves (sid (FK), bid(FK), day(date)).
Find all information of sailors who have reserved boat number 101.

Input 01:

```sql
create database lab3;
use lab3;

create table Sailors (sid int primary key, sname varchar(25), rating varchar(25), age int);
insert into Sailors (sid, sname, rating, age) values
(101, 'WAQAR RIASAT ALI' , '9.9%' , 20 ),
(230, 'ANEEQ SHAMS MACHERA' , '8.9%' , 27 ),
(216, 'HUSSAIN MACHERA' , '7.9%' , 26 ),
(219, 'QASIM MACHERA' , '6.9%' , 25 ),
(250, 'SAQIB MACHERA' , '5.9%' , 24 ),
(376, 'HASSAN MACHERA' , '4.9%' , 23 ),
(220, 'ADEEL MACHERA' , '3.9%' , 22 );


create table Boats(B_id int primary key, bname varchar(25), color varchar(25));
insert into Boats (B_id , bname, color) values
(101, 'Surfing Boat', 'Blue'),
(230, 'Fishing Boat', 'Red'),
(216, 'Speed Boat', 'White'),
(219, 'Sail Boat', 'Green'),
```

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

```sql
insert into Boats (B_id , bname, color) values
(101, 'Surfing Boat', 'Blue'),
(230, 'Fishing Boat', 'Red'),
(216, 'Speed Boat', 'White'),
(219, 'Sail Boat', 'Green'),
(250, 'Rescue Boat', 'Yellow'),
(376, 'Rowing Boat', 'Brown'),
(220, 'Luxury Yacht', 'Black');

create table Reserves(sid INT, B_id INT ,day date , foreign key (sid) references Sailors(sid), foreign key (B_id)
 references Boats(B_id) );

INSERT INTO Reserves (sid, B_id, day) VALUES
(101, 101, '2025-04-18'),
(230, 101, '2025-04-19');


SELECT S.sid, S.sname, B.bname, R.day  -- sailors_idi ,sailors_name & Boats_idi, Boats_name & Reserve_day choose kro.
FROM Sailors S                         -- 'S' is the short form of saiors used above
JOIN Reserves R ON S.sid = R.sid -- agar sailors table me se koyi bhi idi match hojaye reserves table ki idi se  or
JOIN Boats B ON R.B_id = B.B_id -- agar Boats table me se koyi bhi idi match hojaye reserves table ki Boat  idi se to
WHERE R.B_id = 101;              -- condition of Boat idi "101"   to  "S.sid, S.sname, B.bname, R.day" show krdo

```

Output 01:

sid	sname	bname	day
101	WAQAR RIASAT ALI	Surfing Boat	2025-04-18
230	ANEEQ SHAMS MACHERA	Surfing Boat	2025-04-19

2. Write a query to find and display the name of boat reserved by Bob

Input 02:

```sql

SELECT  S.sname, B.bname, R.day
FROM Sailors S
JOIN Reserves R ON S.sid = R.sid
JOIN Boats B ON R.B_id = B.B_id -
WHERE R.sid = 101;
```

Output 02:

sname	bname	day
WAQAR RIASAT ALI	Surfing Boat	2025-04-18
WAQAR RIASAT ALI	Surfing Boat	2025-04-18
WAQAR RIASAT ALI	Rescue Boat	2025-04-01
WAQAR RIASAT ALI	Sail Boat	2025-04-01

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

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

Input 03:

```
4    SELECT  S.sname, B.bname, B.color, R.day
5    FROM Sailors S
6    JOIN Reserves R ON S.sid = R.sid
7    JOIN Boats B ON R.B_id = B.B_id
8    WHERE B.color = 'red';
```

Output 03:

	sname	bname	color	day
▶	SAQIB MACHERA	Fishing Boat	Red	2025-04-13

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

Input 04:

```
54    SELECT  S.sname, B.bname, B.color, R.day
55    FROM Sailors S
56    JOIN Reserves R ON S.sid = R.sid
57    JOIN Boats B ON R.B_id = B.B_id
58    WHERE B.color = 'red' OR B.color = 'Green';
```

Output 04:

	sname	bname	color	day
	WAQAR RIASAT ALI	Sail Boat	Green	2025-04-01
▶	SAQIB MACHERA	Fishing Boat	Red	2025-04-13

5. Create a table Projects with the following schema. (P_id, Pro_name, Dep_id). Now write a query to show which projects have been assigned to the ‘HR’ department. 6

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

Input 05:

```
64                    -- TASK NO: 5
65
66  create table Projects(P_id int Primary key , Pro_name varchar(25), Dep_id int, Assigned_department varchar(20) );
67
68  INSERT INTO Projects (P_id, Pro_name, Dep_id, Assigned_department) VALUES
69  (1, 'GFS', 101, 'HR department'),
70  (2, 'ARY LAGUNA', 102, 'Finance department'),
71  (3, 'BAHRIA TOWN KARACHI', 103, 'Sales department'),
72  (5, 'DHA ISLAMABAD', 104, 'Marketing department'),
73  (6, 'HABIB METRO', 105, 'HR department'),
74  (7, 'KDA LEASE', 106, 'Audit department'),
75  (8, 'SCHEME33', 107, 'Technical department');
76
77  select * from Projects
78   where Assigned_department = 'HR department' ;
```

Output 05:

P_id	Pro_name	Dep_id	Assigned_department
1	GFS	101	HR department
6	HABIB METRO	105	HR department
NULL	NULL	NULL	NULL

6. Write a query to display project names along with their allocated department names. Also display the names of those projects which have not been assigned to any department yet.

Input 06:

```
19  create table department (
20      Assigned_department varchar(25),
21      P_id int,
22      foreign key(P_id) references Projects(P_id));
23
24  insert into department (Assigned_department, P_id) values
25  ('Finance department', 1),
26  ('Sales department', 2),
27  ('HR department', 3),
28  (null, 5),
29  ('Marketing department', 6),
30  ('Legal department', 7),
31  (null, 9);
32
33  select Projects.Pro_name, department.Assigned_department
34  from Projects
35  Left join
36  department
37  on
38  Projects.P_id = department.P_id;
```

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

Output 06:

Pro_name	Assigned_department
GFS	Finance department
ARY LAGUNA	Sales department
BAHRIA TOWN KARACHI	HR department
DHA ISLAMABAD	NULL
HABIB METRO	Marketing department
KDA LEASE	Legal department
SCHEME33	NULL
NAYA NAZIMABAD	NULL
ASF CITY	NULL

7. Create tables for the student, teacher and course scenario as shown in the manual. Write a query to display the course name along with the allocated teacher's name for that course.

Input 07

```
CREATE TABLE Teachers (
    teacher_id INT PRIMARY KEY,
    teacher_name VARCHAR(100)
);

CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(100),
    teacher_id INT,
    FOREIGN KEY (teacher_id) REFERENCES Teachers(teacher_id)
);

CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    student_name VARCHAR(100)
);

INSERT INTO Teachers VALUES (1, 'Sir Usman');
INSERT INTO Teachers VALUES (2, 'Miss Ayesha');
INSERT INTO Teachers VALUES (3, 'Sir Bilal');

INSERT INTO Courses VALUES (101, 'Database Systems', 1);
INSERT INTO Courses VALUES (102, 'Computer Networks', 2);
INSERT INTO Courses VALUES (103, 'Web Development', 3);
INSERT INTO Courses VALUES (104, 'Artificial Intelligence', NULL);

INSERT INTO Students VALUES (1, 'Ali Raza');
INSERT INTO Students VALUES (2, 'Fatima Khan');
INSERT INTO Students VALUES (3, 'Hassan Javed');

SELECT C.course_name, T.teacher_name FROM Courses C LEFT JOIN Teachers T ON C.teacher_id = T.teacher_id;
```

Output 07:

100 %

Results Messages

	course_name	teacher_name
1	Database Systems	Sir Usman
2	Computer Networks	Miss Ayesha
3	Web Development	Sir Bilal
4	Artificial Intelligence	NULL