EX.NO.4	**JOIN OPERATIONS**

AIM

To execute the join operations.

CREATING THE TABLE

SQL> create table customers(CustomerID int primary key,Name varchar(10),City varchar(12));

Table created.

SQL> create table Orders(OrderID int primary key,CustomerID int,

Product varchar(12));

Table created.

INSERTING VALUES TO THE TABLE

SQL> insert into customers values(1,'Alex','Chennai');

1 row created.

SQL> insert into customers values(2,'Bob','Mumbai');

1 row created.

SQL> insert into customers values(3,'Charlie','Delhi');

1 row created.

SQL> insert into customers values(4,'David','Kolkata');

1 row created.

SQL> insert into customers values(5,'Emma','Bangalore');

1 row created.

SQL> insert into Orders values(101,1,'Pizza');

1 row created.

SQL> insert into Orders values(102,2,'Burger');

1 row created.

SQL> insert into Orders values(103,3,'Pasta');

1 row created.

SQL> insert into Orders values(104,5,'Ice Cream');

1 row created.

SQL> insert into Orders values(105,6,'Sandwich');

1 row created.

DISPLAYING THE TABLE

SQL> select * from customers;

```
CUSTOMERID NAME       CITY
---------- ---------- ------------
         1  Alex       Chennai
         2  Bob        Mumbai
         3  Charlie    Delhi
         4  David      Kolkata
         5  Emma       Bangalore
```

SQL> Select * from Orders;

```
ORDERID CUSTOMERID PRODUCT
---------- ---------- ------------
       101          1 Pizza
       102          2 Burger
       103          3 Pasta
       104          5 Ice Cream
       105          6 Sandwich
```

INNER JOIN

SQL> select Customers.CustomerID,Customers.Name,Orders.OrderID,

Orders.Product from Customers inner join Orders on

Customers.CustomerID=Orders.CustomerID;

CUSTOMERID NAME ORDERID PRODUCT

---------- ---------- ---------- ----------

1 Alex 101 Pizza

2 Bob 102 Burger

3 Charlie 103 Pasta

5 Emma 104 Ice Cream

FULL JOIN

SQL> SELECT Customers.CustomerID, Customers.Name, Orders.OrderID, Orders.Product

FROM Customers

FULL JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

CUSTOMERID NAME ORDERID PRODUCT

---------- ---------- ---------- ----------

1 Alex 101 Pizza

2 Bob 102 Burger

3 Charlie 103 Pasta

4 David

5 Emma 104 Ice Cream

105 Sandwich

6 rows selected.

LEFT JOIN

SQL> select Customers.CustomerID,Customers.Name,Orders.OrderID,

Orders.Product from Customers left join Orders on
Customers.CustomerID=Orders.CustomerID;

CUSTOMERID	NAME	ORDERID	PRODUCT
1	Alex	101	Pizza
2	Bob	102	Burger
3	Charlie	103	Pasta
5	Emma	104	Ice Cream
4	David		

RIGHT JOIN

SQL> select Customers.CustomerID,Customers.Name,Orders.OrderID,

Orders.Product from customers right join Orders on

Customers.CustomerID=Orders.CustomerID;

CUSTOMERID	NAME	ORDERID	PRODUCT
1	Alex	101	Pizza
2	Bob	102	Burger
3	Charlie	103	Pasta
5	Emma	104	Ice Cream
		105	Sandwich

EQUI JOIN

SQL> SELECT customers.CustomerID, customers.Name, customers.City,
Orders.OrderID, Orders.Product

FROM customers

JOIN Orders

ON customers.CustomerID = Orders.CustomerID;

CUSTOMERID	NAME	CITY	ORDERID	PRODUCT
1	Alex	Chennai	101	Pizza
2	Bob	Mumbai	102	Burger
3	Charlie	Delhi	103	Pasta
5	Emma	Bangalore	104	Ice Cream

Natural Join

SQL>SELECT *FROM customers NATURAL JOIN Orders;

CUSTOMERID	NAME	CITY	ORDERID	PRODUCT
1	Alex	Chennai	101	Pizza
2	Bob	Mumbai	102	Burger
3	Charlie	Delhi	103	Pasta
5	Emma	Bangalore	104	Ice Cream

Condition Join

SQL> SELECT Customers.CustomerID, Customers.Name, Orders.OrderID, Orders.Product

FROM Customers

JOIN Orders ON Customers.CustomerID > Orders.CustomerID;

CUSTOMERID	NAME	ORDERID	PRODUCT
2	Bob	101	Pizza
3	Charlie	101	Pizza

4	David	101 Pizza
5	Emma	101 Pizza
3	Charlie	102 Burger
4	David	102 Burger
5	Emma	102 Burger
4	David	103 Pasta
5	Emma	103 Pasta

9 rows selected.

CONTENTS	MARKS ALLOTED	MARKS OBTAINED
Aim,Algorithm,SQL,PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

RESULT

Thus, the inner join, full join, left join, right join, equi join, natural join, and condition join operations were executed successfully. This experiment helped in understanding how SQL joins work to retrieve meaningful insights from relational databases.