SQL Joins

SQL JOIN

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Why SQL JOIN is used?

If you want to access more than one table through a select statement.

If you want to combine two or more table then SQL JOIN statement is used .it combines rows of that tables in one table and one can retrieve the information by a SELECT statement.

The joining of two or more tables is based on common field between them.

SQL INNER JOIN also known as simple join is the most common type of join.

Create 2 Tables

```
Create Table Orders(
    OrderID int NOT NULL,
    CustomerID int NOT NULL,
    OrderDate Date
);
```

```
Create Table Customers(
    CustomerID int NOT NULL,
    CustomerName varchar(255),
    ContactName varchar(255),
    Country varchar(255)
);
```

Insert Values

```
INSERT INTO ORDERS
VALUES(10308,2,TO_DATE('1996/09/18','yyyy/mm/dd'));

INSERT INTO ORDERS
VALUES(10309,37,TO_DATE('1996-09-19','yyyy/mm/dd'));

INSERT INTO ORDERS
VALUES(10310,77,TO_DATE('1996-09-20','yyyy/mm/dd'));
```

```
INSERT INTO Customers
VALUES(2,'Saurav','Saurav','Nepal');

INSERT INTO Customers
VALUES(37,'Lokesh','Loki','India');
```

SQL JOIN

Then, we can create the following SQL statement (that contains an INNER JOIN), that selects records that have matching values in both tables:

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

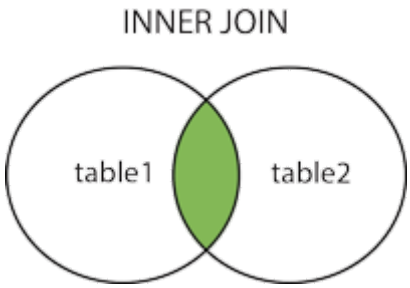
Output

ORDERID	CUSTOMERNAME	ORDERDATE
10308	Saurav	18-SEP-96
10309	Lokesh	19-SEP-96

Different Types of SQL JOINs

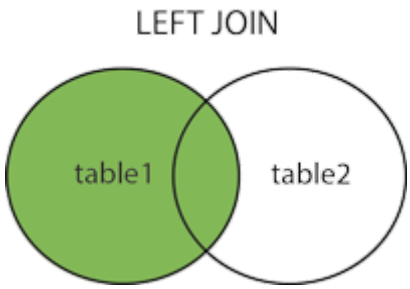
Here are the different types of the JOINs in SQL:

- **(INNER) JOIN**: Returns records that have matching values in both tables.

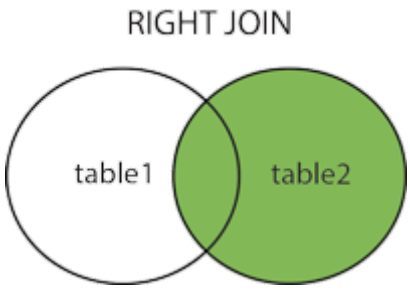


- **LEFT (OUTER) JOIN**(also known as left join): this join returns all the rows from left table combine with the matching rows of the right table. If you get no matching in the right table it returns NULL values.

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

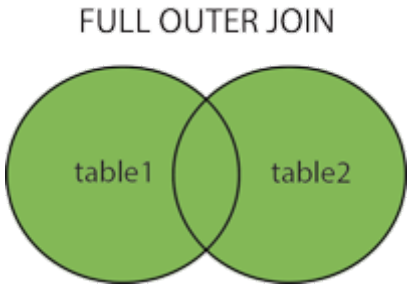


- **RIGHT (OUTER) JOIN**(also known as right join): this join returns all the rows from right table combined with the matching rows of left table .If you get no column matching in the left table it returns null value.




```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

- **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table




```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

Example:

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
Full Outer JOIN Customers ON Orders.CustomerID=Customers.CustomerID
Order By Customers.CustomerName;
```

> **Note:** The FULL OUTER JOIN keyword returns all matching records from both tables whether the other table matches or not. So, if there are rows in "Customers" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Customers", those rows will be listed as well.

SQL Self JOIN

A self JOIN is a regular join, but the table is joined with itself.

Self JOIN Syntax

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

Example:

```
SELECT A.CustomerName AS CustomerName1, A.Country
FROM Customers A, Customers B
WHERE A.CustomerID <> B.CustomerID
AND A.Country != B.Country
ORDER BY A.Country;
```

Output:

CUSTOMERNAME1	COUNTRY
Lokesh	India
Saurav	Nepal