

# SQL Joins

## Inner Join, Left Join, Right Join, Full Join

W3Schools

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

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

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

OrderID	CustomerName	OrderDate
10278	Berglunds snabbköp	1996-08-12
10280	Berglunds snabbköp	1996-08-14
10308	Ana Trujillo Emparedados y helados	1996-09-18
10355	Around the Horn	1996-11-15
10365	Antonio Moreno Taquería	1996-11-27
10383	Around the Horn	1996-12-16
10384	Berglunds snabbköp	1996-12-16

# JOIN Three Tables

The following SQL statement selects all orders with customer and shipper information:

## Example

```
SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName
FROM ((Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);
```

```
SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName
FROM ((Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);
```

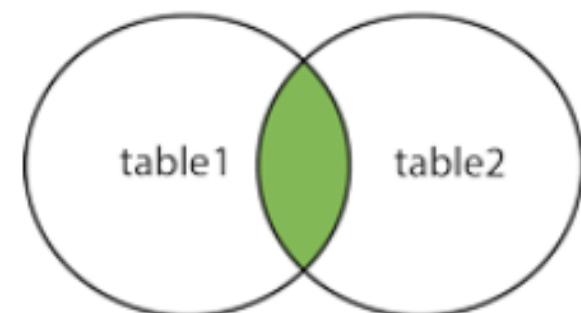
OrderID	CustomerName	ShipperName
10280	Berglunds snabbköp	Speedy Express
10355	Around the Horn	Speedy Express
10278	Berglunds snabbköp	United Package
10365	Antonio Moreno Taquería	United Package
10308	Ana Trujillo Emparedados y helados	Federal Shipping
10383	Around the Horn	Federal Shipping
10384	Berglunds snabbköp	Federal Shipping

# Different Types of SQL JOINS

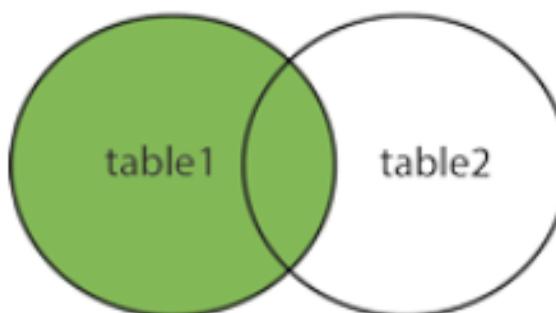
Here are the different types of the JOINs in SQL:

- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table

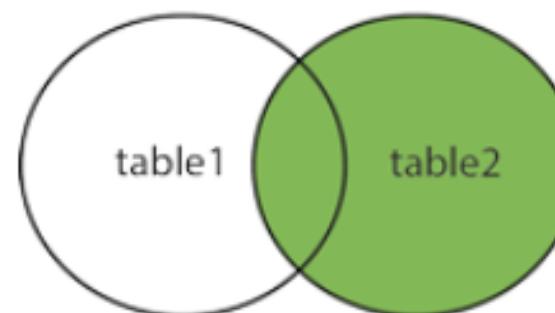
INNER JOIN



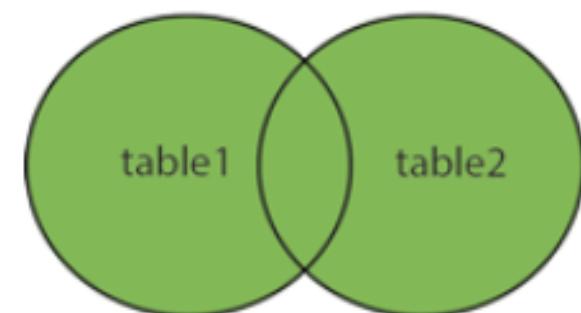
LEFT JOIN



RIGHT JOIN



FULL OUTER JOIN



# SQL LEFT JOIN

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2).

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

**Note:** The LEFT JOIN keyword returns all records from the left table (Customers), even if there are no matches in the right table (Orders).

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

The following SQL statement will select all customers, and any orders they might have:

## SQL Statement:

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

CustomerName	OrderID
Alfreds Futterkiste	NULL
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
Around the Horn	10383
Around the Horn	10355
Berglunds snabbköp	10280
Berglunds snabbköp	10384
Berglunds snabbköp	10278
Eugene	NULL
Jane	NULL

# SQL RIGHT JOIN

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

**Note:** The RIGHT JOIN keyword returns all records from the right table (Employees), even if there are no matches in the left table (Orders).

OrderID	CustomerName
NULL	Jane
NULL	Eugene
NULL	Alfreds Futterkiste
10278	Berglunds snabbköp
10280	Berglunds snabbköp
10308	Ana Trujillo Emparedados y helados
10355	Around the Horn
10365	Antonio Moreno Taquería
10383	Around the Horn
10384	Berglunds snabbköp

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
RIGHT JOIN Customers ON Orders.CustomerID =
Customers.CustomerID
ORDER BY Orders.OrderID;
```

# SQL FULL OUTER JOIN

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

**Tip:** FULL OUTER JOIN and FULL JOIN are the same

**Note:** FULL OUTER JOIN can potentially return very large result-sets!

MySQL does not support FULL JOIN, so you have to combine LEFT JOIN UNION RIGHT JOIN to get an equivalent. It gives the results of A union B. It returns all records from both tables. Those columns which exist in only one table will contain NULL in the opposite table.

**Any way to use this syntax on MySQL?**

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

Provide the SQL statement for a full join.

```
SELECT Customers.CustomerName FROM customers left join  
orders on Customers.CustomerID = Orders.CustomerID  
UNION  
SELECT Orders.OrderID FROM customers RIGHT join orders  
on Customers.CustomerID = Orders.CustomerID;
```

CustomerName

Berglunds snabbköp

Ana Trujillo Emparedados y helados

Around the Horn

Antonio Moreno Taquería

Alfreds Futterkiste

Eugene

Jane

10278

10280

10308

10355

10365

10383

10384

## Provide the SQL statement for a full join.

CustomerName	OrderID	
Berglunds snabbköp	10278	<pre>SELECT Customers.CustomerName, Orders.OrderID FROM customers <b>left join</b> orders on Customers.CustomerID = Orders.CustomerID</pre>
Berglunds snabbköp	10280	<pre>UNION</pre>
Ana Trujillo Emparedados y helados	10308	<pre>SELECT Orders.OrderID, Customers.CustomerName FROM customers <b>RIGHT join</b> orders on Customers.CustomerID = Orders.CustomerID;</pre>
Around the Horn	10355	
Antonio Moreno Taquería	10365	
Around the Horn	10383	
Berglunds snabbköp	10384	
Alfreds Futterkiste	NULL	
Eugene	NULL	
Jane	NULL	
10278	Berglunds snabbköp	
10280	Berglunds snabbköp	
10308	Ana Trujillo Emparedados y helados	
10355	Around the Horn	
10365	Antonio Moreno Taquería	
10383	Around the Horn	
10384	Berglunds snabbköp	

# Self Join

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

<b>CustomerID</b>	<b>CustomerName</b>	<b>ContactName</b>	<b>Address</b>	<b>City</b>	<b>PostalCode</b>	<b>Country</b>
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

The following SQL statement matches customers that are from the same city:

## Example

```
SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City
FROM Customers A, Customers B
WHERE A.CustomerID <> B.CustomerID
AND A.City = B.City
ORDER BY A.City;
```

CustomerName1	CustomerName2	City
Ana Trujillo Emparedados y helados	Antonio Moreno Taquería	México D.F.
Antonio Moreno Taquería	Ana Trujillo Emparedados y helados	México D.F.