**MySQL Joining Tables**

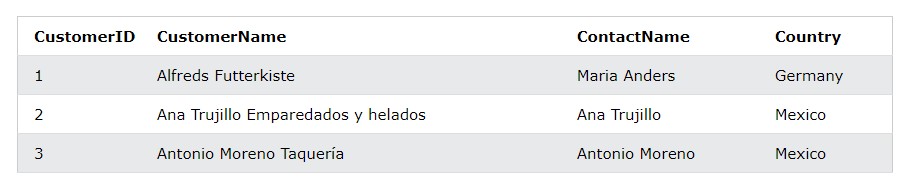
A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Let's look at a selection from the "Orders" table:



Then, look at a selection from the "Customers" table

:



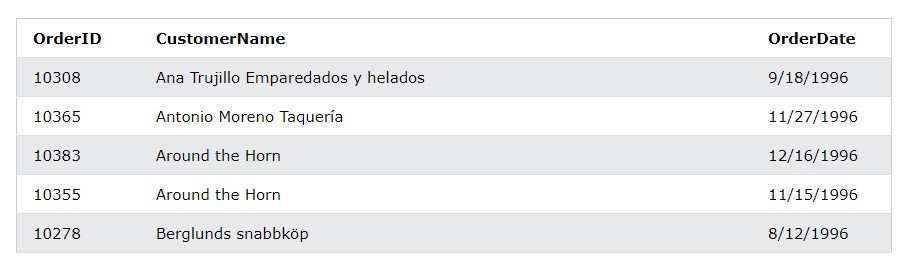
Notice that the "CustomerID" column in the "Orders" table refers to the "CustomerID" in the "Customers" table. The relationship between the two tables above is the "CustomerID" column.

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

|  |
| --- |
| **Example** |
| SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate |

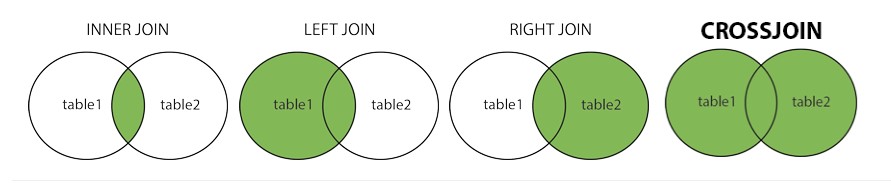
FROM Orders

INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID; and it will produce something like this:



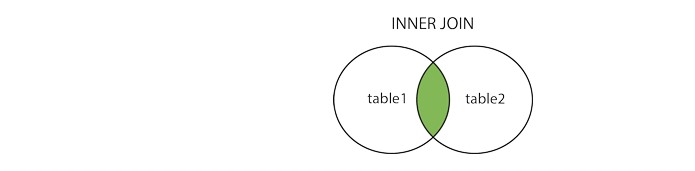
**Supported Types of Joins in MySQL**

* INNER JOIN: Returns records that have matching values in both tables
* LEFT JOIN: Returns all records from the left table, and the matched records from the right table
* RIGHT JOIN: Returns all records from the right table, and the matched records from the left table
* CROSS JOIN: Returns all records from both tables



**MySQL INNER JOIN Keyword**

The INNER JOIN keyword selects records that have matching values in both tables.



# INNER JOIN Syntax

SELECT *column\_name(s)*

FROM *table1*

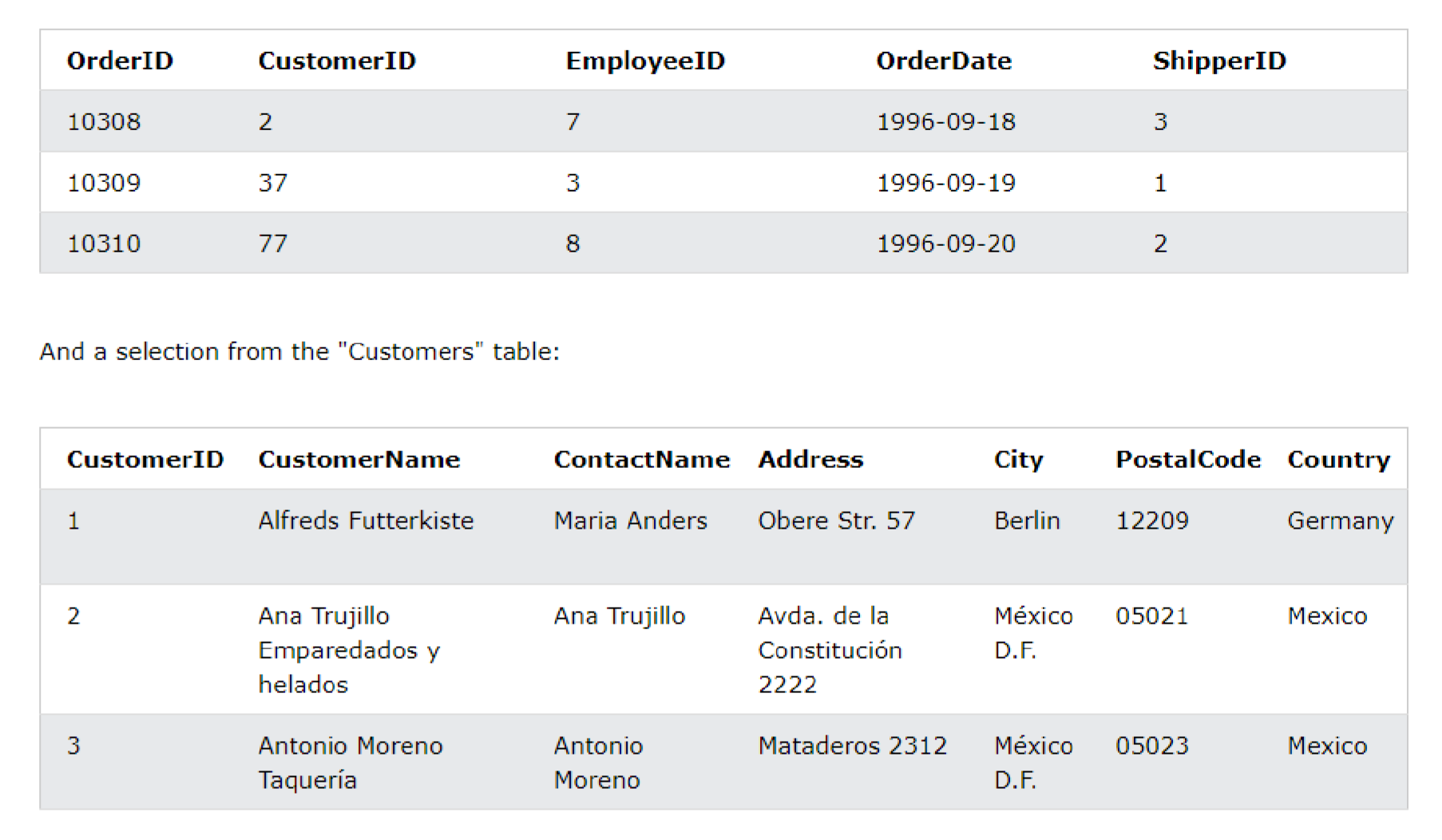
INNER JOIN *table2*

ON *table1.column\_name* = *table2.column\_name*;

**Demo Database**

In this tutorial we will use the well-known Northwind sample database.

Below is a selection from the "Orders" table:



**MySQL INNER JOIN Example**

The following SQL statement selects all orders with customer information:

|  |
| --- |
| **Example** |
| SELECT Orders.OrderID, Customers.CustomerName |

FROM Orders

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

**Note:** The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns. If there are records in the "Orders" table that do not have matches in "Customers", these orders will not be shown!

**JOIN Three Tables**

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

# Example

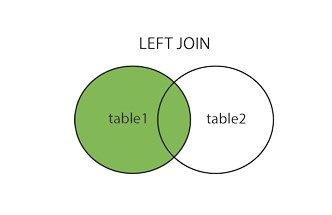
SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName

FROM ((Orders

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID) INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);

**MySQL LEFT JOIN Keyword**

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



# LEFT JOIN Syntax

SELECT *column\_name(s)*

FROM *table1*

LEFT JOIN *table2*

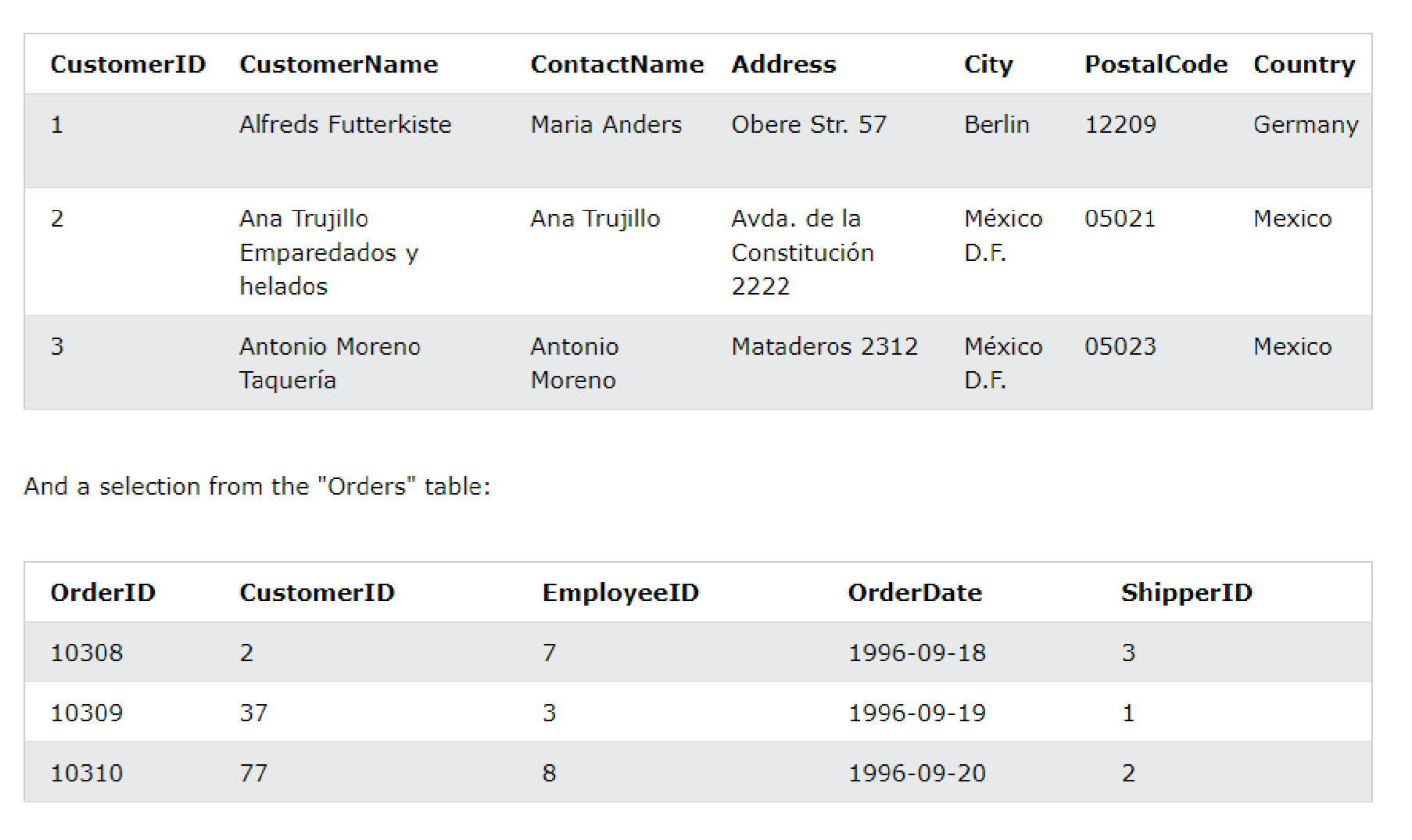
ON *table1.column\_name* = *table2.column\_name*;

**Demo Database**

In this tutorial we will use the well-known Northwind sample database.

Below is a selection from the "Customers" table:

**MySQL LEFT JOIN Example**



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

|  |
| --- |
| **Example** |
| SELECT Customers.CustomerName, Orders.OrderID |

FROM Customers

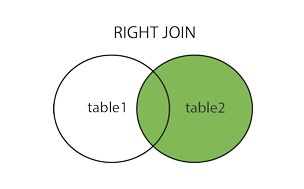
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID ORDER BY Customers.CustomerName;

**Note:** The LEFT JOIN keyword returns all records from the left table

(Customers), even if there are no matches in the right table (Orders).

**MySQL RIGHT JOIN Keyword**

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records (if any) from the left table (table1).



# RIGHT JOIN Syntax

SELECT *column\_name(s)*

FROM *table1*

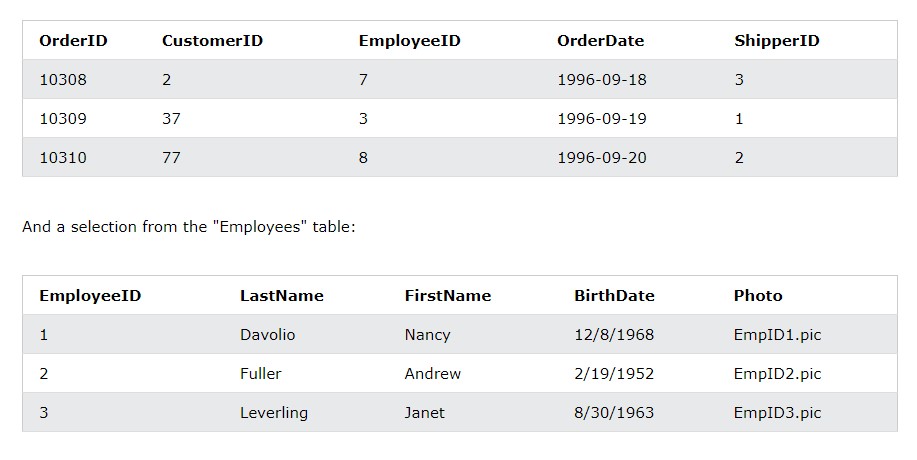
RIGHT JOIN *table2*

ON *table1.column\_name* = *table2.column\_name*;

**Demo Database**

In this tutorial we will use the well-known Northwind sample database.

Below is a selection from the "Orders" table:



**MySQL RIGHT JOIN Example**

The following SQL statement will return all employees, and any orders they might have placed:

|  |
| --- |
| **Example** |
| SELECT Orders.OrderID, Employees.LastName, Employees.FirstName |

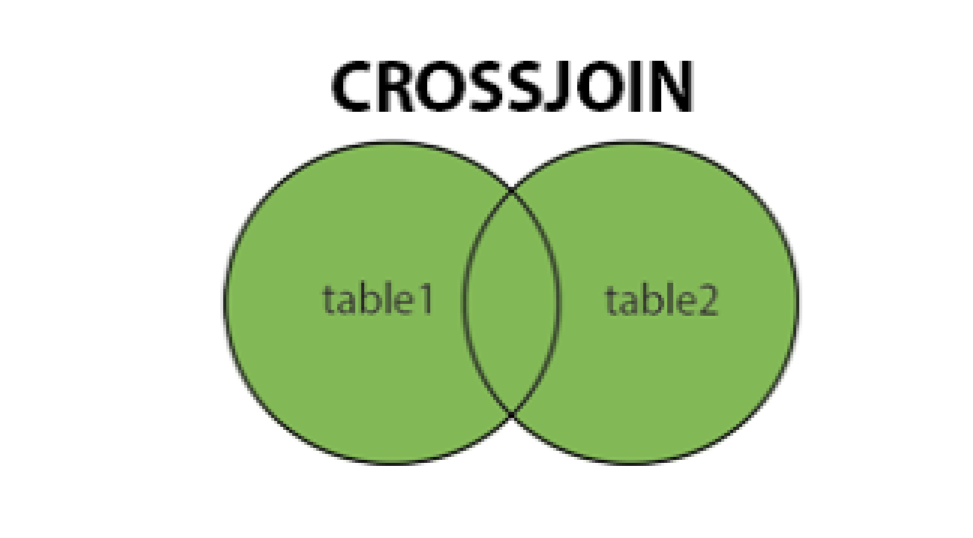
FROM Orders

RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID ORDER BY Orders.OrderID;

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

**SQL CROSS JOIN Keyword**

The CROSS JOIN keyword returns all records from both tables (table1 and table2).



# CROSS JOIN Syntax

SELECT *column\_name(s)*

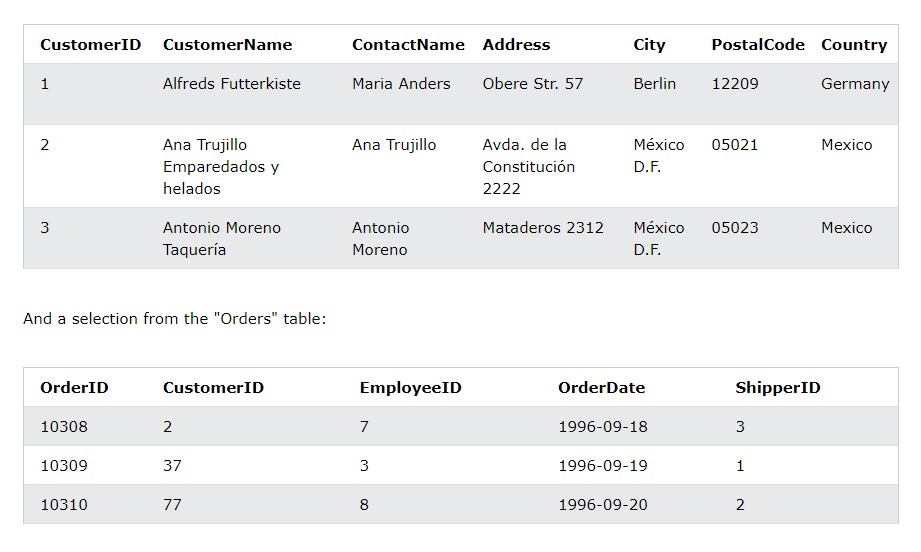
FROM *table1*

CROSS JOIN *table2*;

|  |
| --- |
| **Note:** CROSS JOIN can potentially return very large result-sets! |
| **Demo Database** |

In this tutorial we will use the well-known Northwind sample database.

Below is a selection from the "Customers" table:



**MySQL CROSS JOIN Example**

The following SQL statement selects all customers, and all orders:

|  |
| --- |
| **Example** |
| SELECT Customers.CustomerName, Orders.OrderID |

FROM Customers

CROSS JOIN Orders;

**Note:** The CROSS 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.

If you add a WHERE clause (if table1 and table2 has a relationship), the CROSS JOIN will produce the same result as the INNER JOIN clause:

|  |
| --- |
| **Example** |
| SELECT Customers.CustomerName, Orders.OrderID |

FROM Customers

CROSS JOIN Orders

WHERE Customers.CustomerID=Orders.CustomerID;

**MySQL 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*;

*T1* and *T2* are different table aliases for the same table.

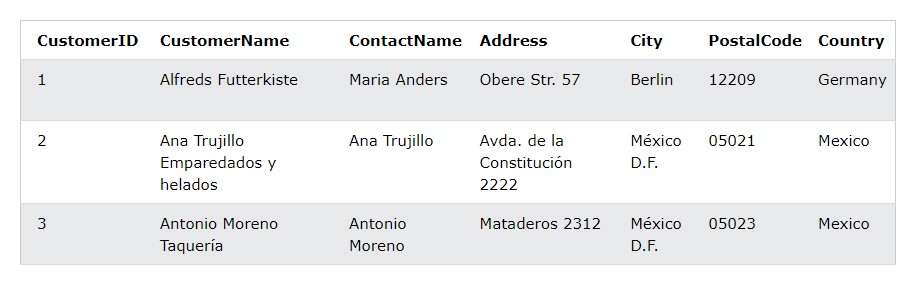
**Demo Database**

In this tutorial we will use the well-known Northwind sample database.

Below is a selection from the "Customers" table:

**MySQL**

**Self Join Example**



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;