# The MySQL UNION Operator

The UNION operator is used to combine the result-set of two or more SELECT statements.

* Every SELECT statement within UNION must have the same number of columns
* The columns must also have similar data types
* The columns in every SELECT statement must also be in the same order

## UNION Syntax

SELECT *column\_name(s)* FROM *table1*

UNION

SELECT *column\_name(s)* FROM *table2*;

## UNION ALL Syntax

The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL:

SELECT *column\_name(s)* FROM *table1*

UNION ALL

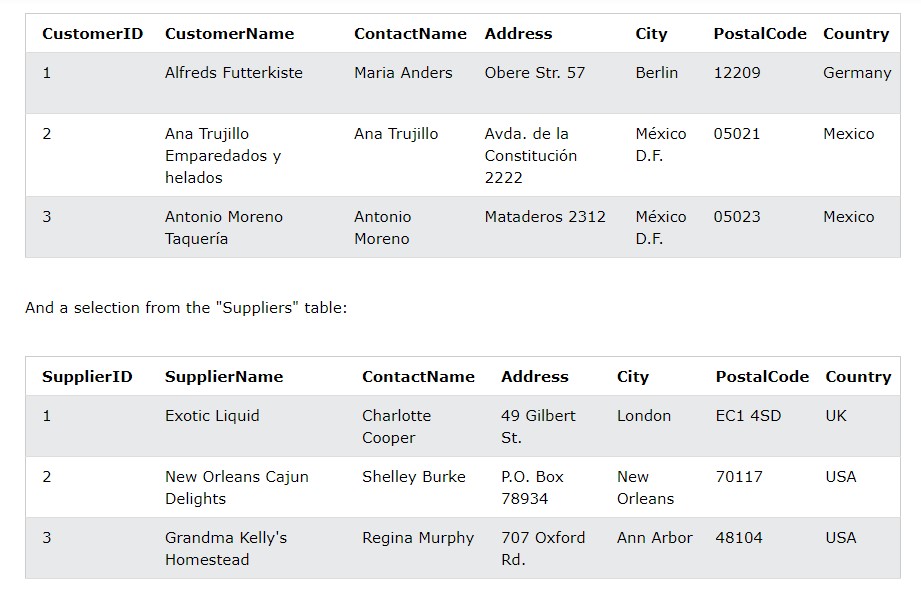
SELECT *column\_name(s)* FROM *table2*;

**Note:** The column names in the result-set are usually equal to the column names in the first SELECT statement.

# Demo Database

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

Below is a selection from the "Customers" table:



# SQL UNION Example

The following SQL statement returns the cities (only distinct values) from both the "Customers" and the "Suppliers" table:

**Example**

SELECT City FROM Customers UNION

SELECT City FROM Suppliers

ORDER BY City;

**Note:** If some customers or suppliers have the same city, each city will only be listed once, because UNION selects only distinct values. Use UNION ALL to also select duplicate values!

# SQL UNION ALL Example

The following SQL statement returns the cities (duplicate values also) from both the "Customers" and the "Suppliers" table:

|  |
| --- |
| **Example** |
| SELECT City FROM Customers |

UNION ALL

SELECT City FROM Suppliers ORDER BY City;

# SQL UNION With WHERE

The following SQL statement returns the German cities (only distinct values) from both the "Customers" and the "Suppliers" table:

|  |  |
| --- | --- |
| **Example** |  |
| SELECT City, Country FROM | Customers |
| WHERE Country='Germany'  UNION |  |
| SELECT City, Country FROM  WHERE Country='Germany' | Suppliers |
| ORDER BY City;  **SQL UNION A LL With WHERE**  The following SQL statement returns the German cities (duplicate values also) from both the "Custo mers" and the "Suppliers" table: | |
| **Example** |  |
| SELECT City, Country FROM | Customers |
| WHERE Country='Germany'  UNION ALL |  |
| SELECT City, Country FROM  WHERE Country='Germany'  ORDER BY City; | Suppliers |
| **Another UNIO N Example**  The following SQL statement lists all customers and suppliers: | |
| **Example** | |
| SELECT 'Customer' AS Type, ContactName, City, Country | |

FROM Customers

UNION

SELECT 'Supplier', ContactName, City, Country FROM Suppliers;

Notice the "AS Type" above - it is an alias. SQL Aliases are used to give a table or a column a temporary name. An alias only exists for the duration of the query. So, here we have created a temporary column named "Type", that list whether the contact person is a "Customer" or a "Supplier".

=============================================ss