Some of the Most Important SQL Commands

* **SELECT** - extracts data from a database
* **UPDATE** - updates data in a database
* **DELETE** - deletes data from a database
* **INSERT INTO** - inserts new data into a database
* **CREATE DATABASE** - creates a new database
* **ALTER DATABASE** - modifies a database
* **CREATE TABLE** - creates a new table
* **ALTER TABLE** - modifies a table
* **DROP TABLE** - deletes a table
* **CREATE INDEX** - creates an index (search key)
* **DROP INDEX** - deletes an index

**The SQL SELECT Statement**

The SELECT statement is used to select data from a database.

### SELECT Syntax

**1)**

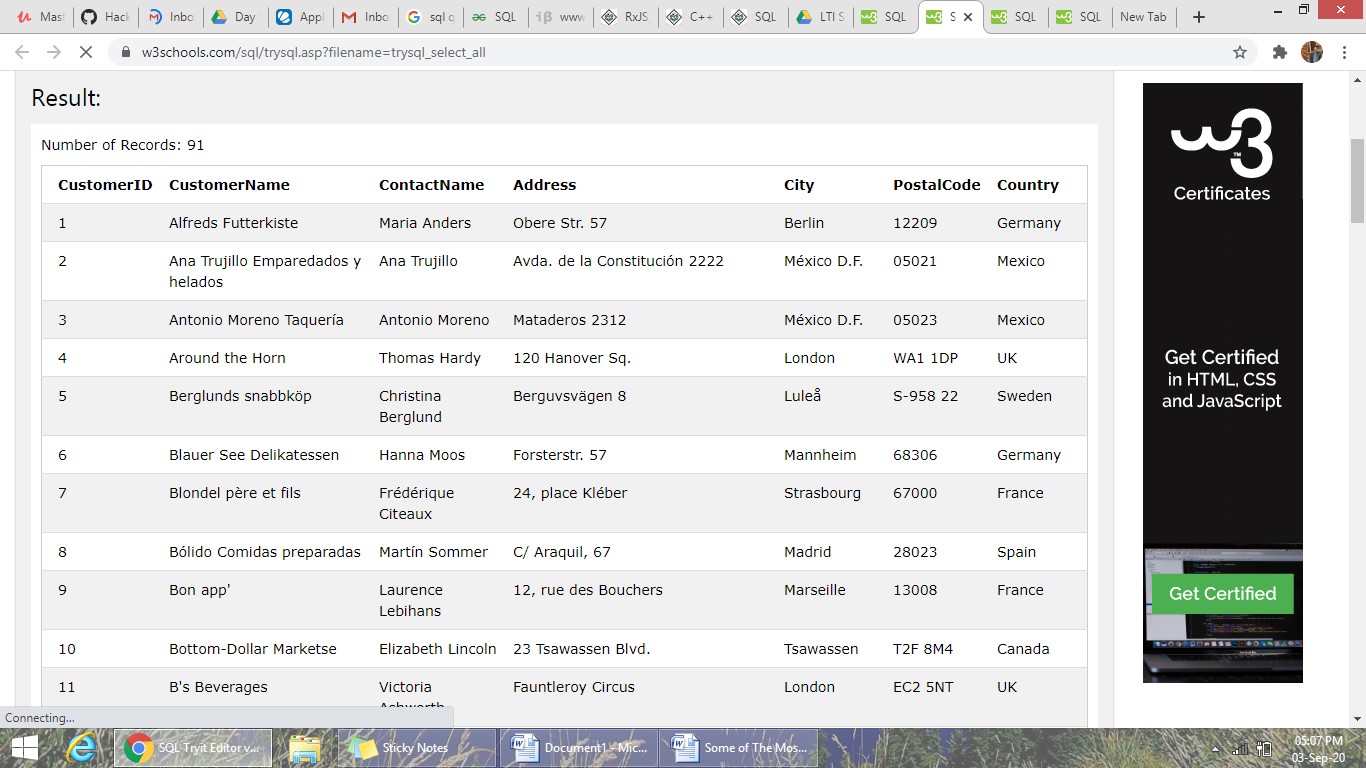
### SELECT column1, column2, ... FROM table\_name;

### Here, column1, column2, ... are the field names of the table you want to select data from.

### 2)

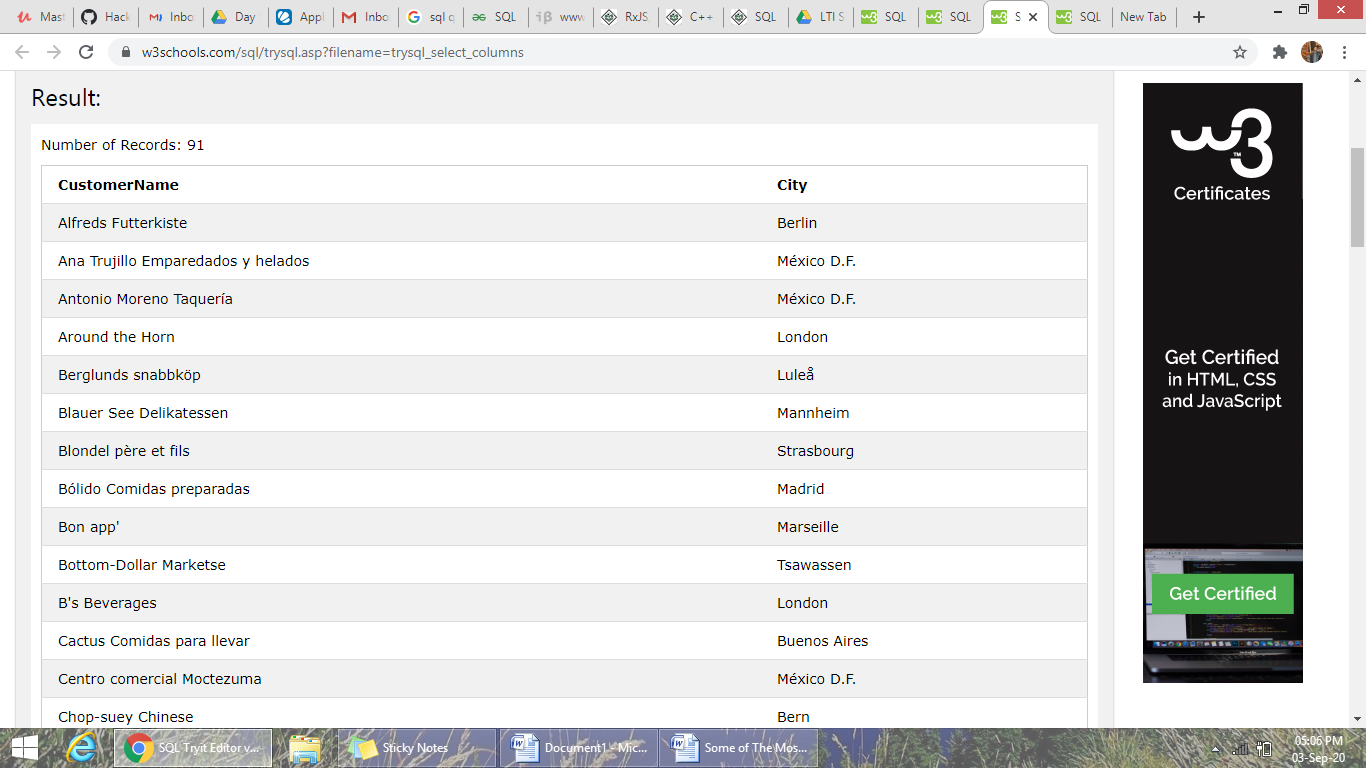
### If you want to select all the fields available in the table, use the following syntax:

### SELECT \* FROM table\_name;



**Example**

SELECT CustomerName, City FROM Customers;



**Exercise**

1)Write a statement that will select the City column from the Customers table

**SELECT City FROM** Customers;

The SQL SELECT DISTINCT Statement

The SELECT DISTINCT statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

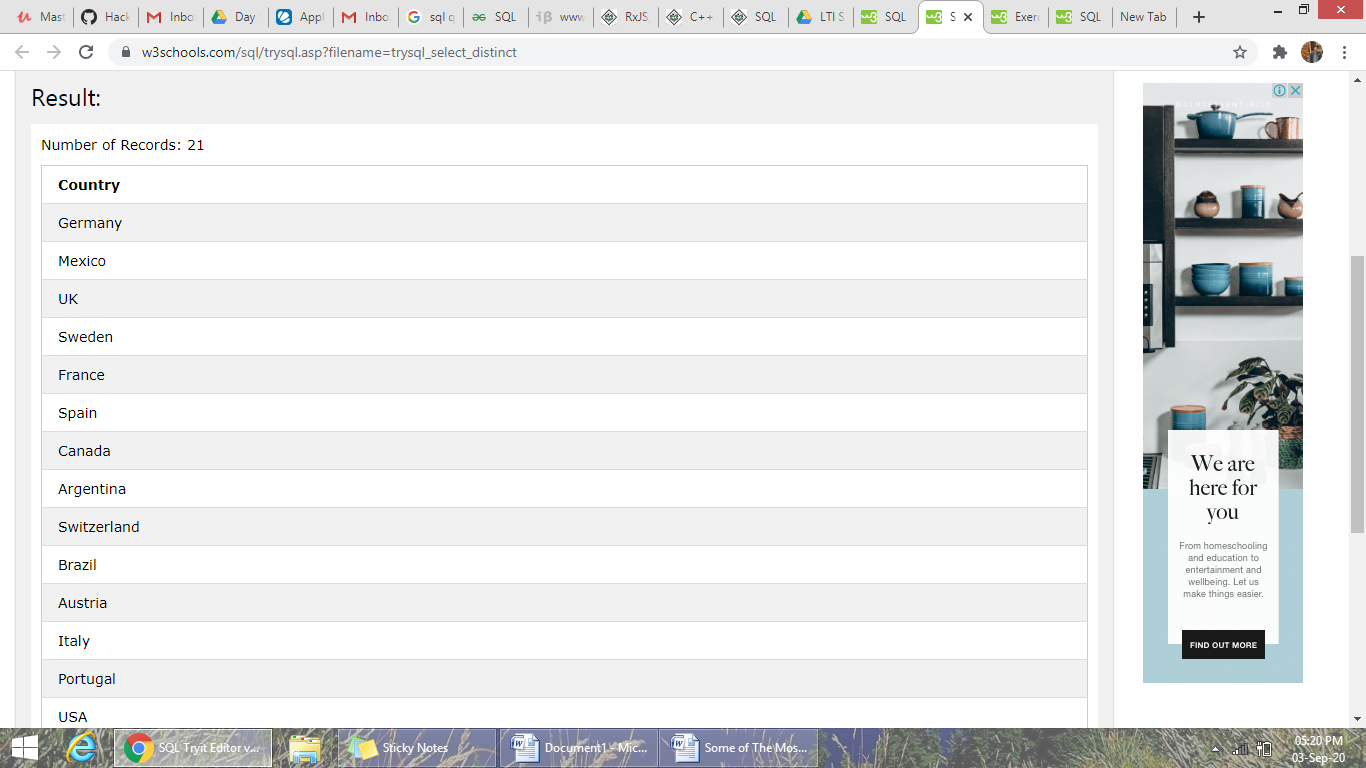
### SELECT DISTINCT Syntax

SELECT DISTINCT column1, column2,  
FROM table\_name;

**Exercise**

1. Select all the different values from the Country column in the Customers table.

**SELECT DISTINCT** Country FROM Customers;

****

SQL WHERE Clause

## The SQL WHERE Clause

The WHERE clause is used to filter records.

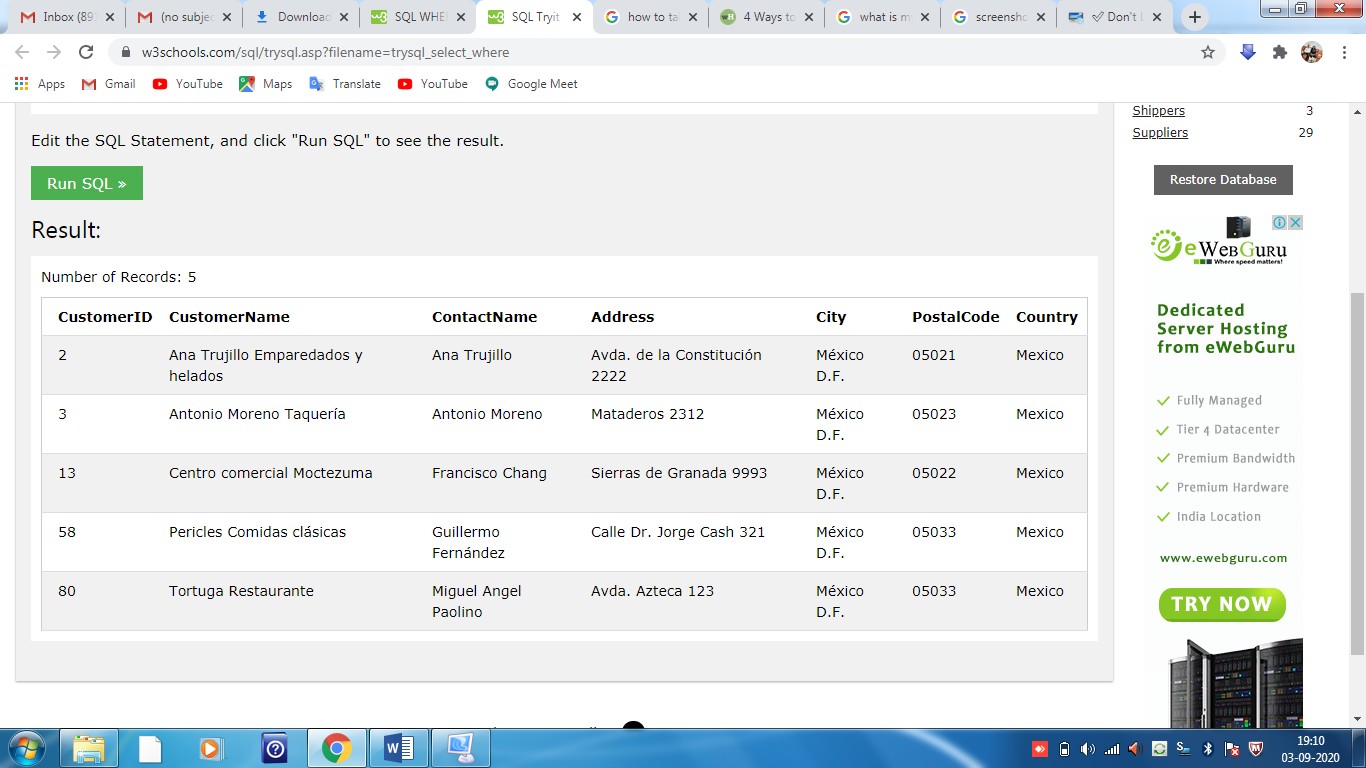
The WHERE clause is used to extract only those records that fulfill a specified condition.

### WHERE Syntax

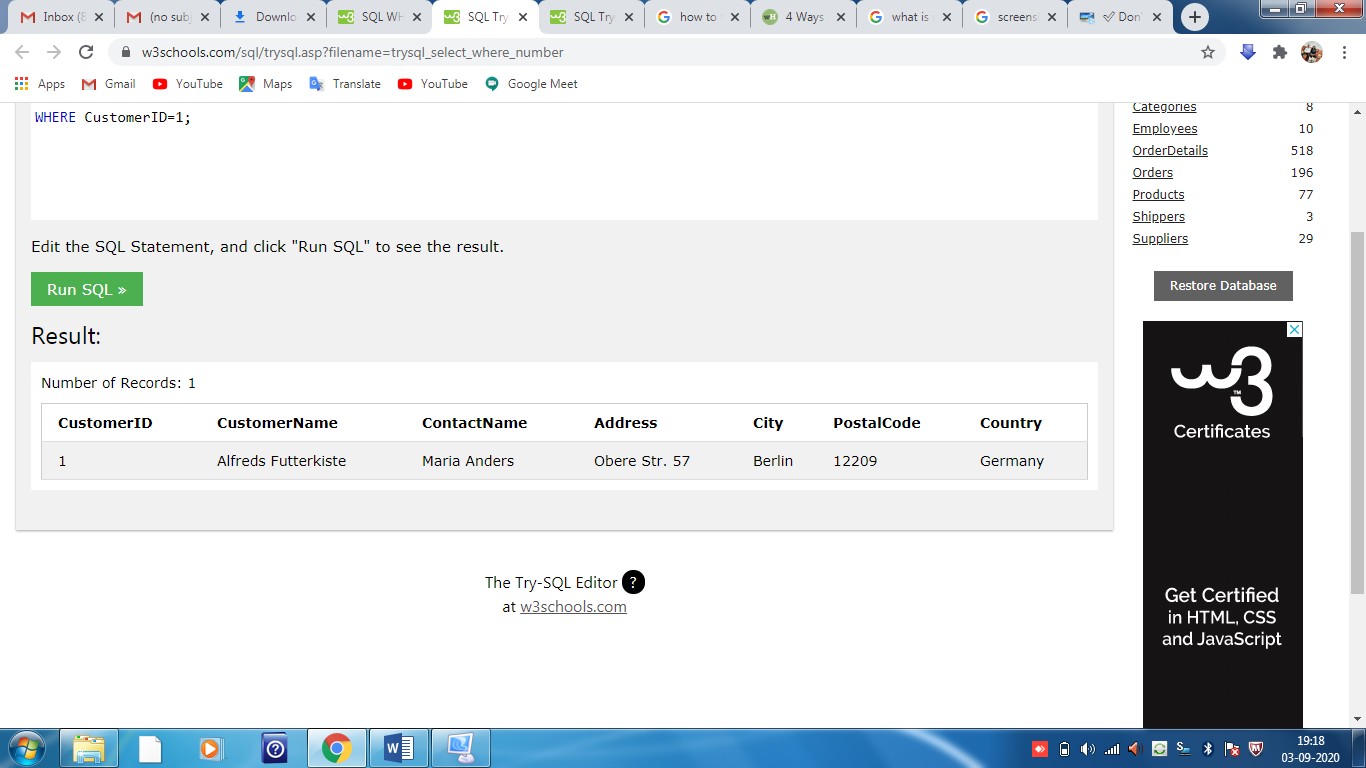
SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

**Exercise**

1. SELECT \* FROM Customers  
   WHERE Country='Mexico';



1. SELECT \* FROM Customers  
   WHERE CustomerID=1;



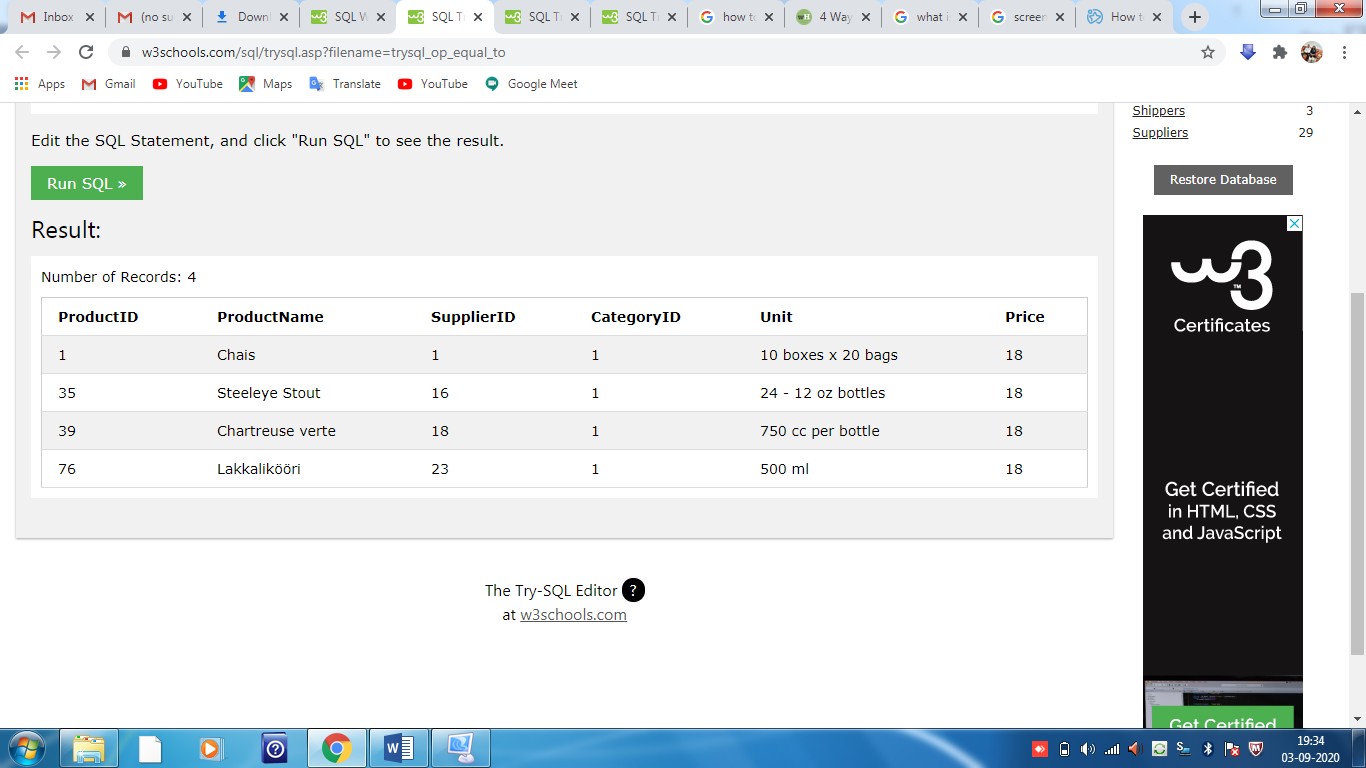
## Operators in the WHERE Clause

The following operators can be used in the WHERE clause:

|  |  |
| --- | --- |
| = | Equal |
| **>** | Greater than |
| **<** | Less than |
| **>=** | Greater than or equal |
| **<=** | Less than or equal |
| **<>** | Not equal. **Note:** In some versions of SQL this operator may be written as != |
| **BETWEEN** | Between a certain range |
| **LIKE** | Search for a pattern |
| **IN** | To specify multiple possible values for a column |

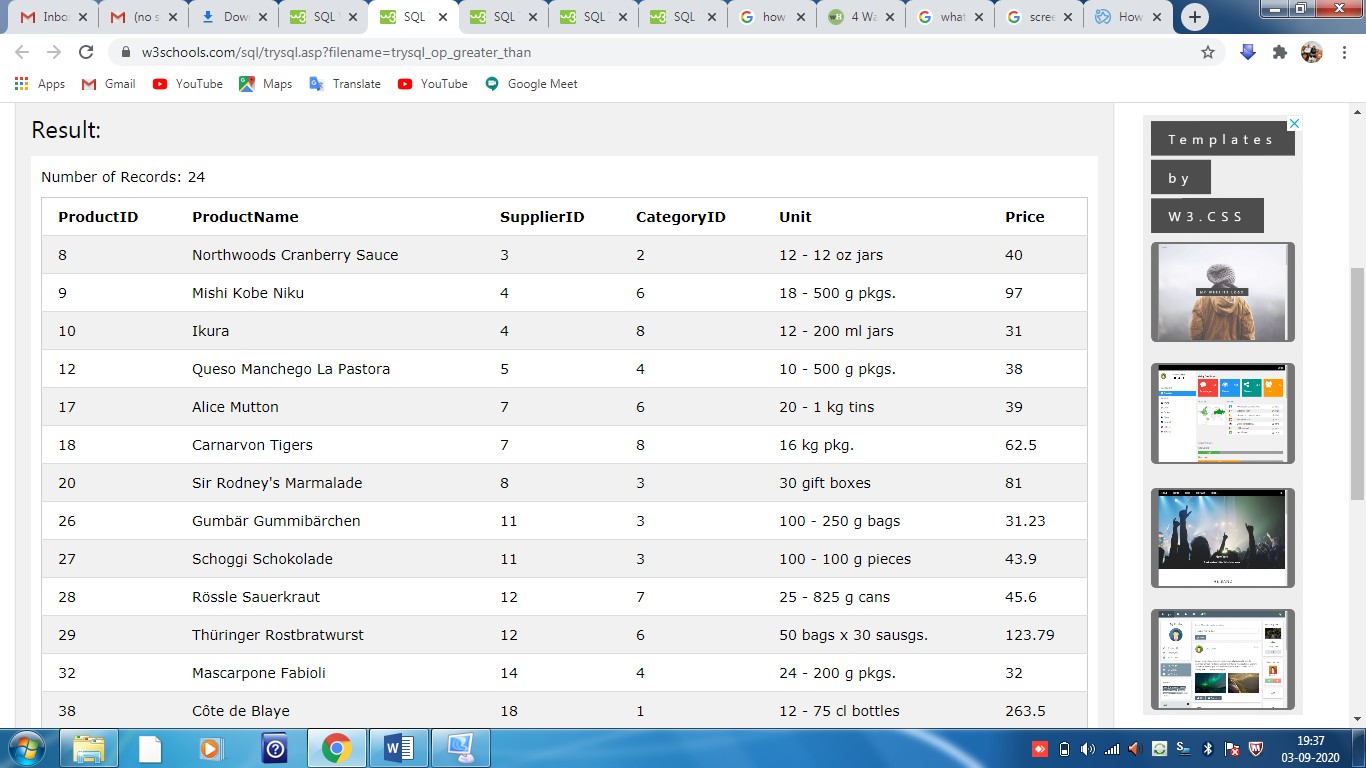
**SELECT \* FROM Products**

**WHERE Price = 18;**

****

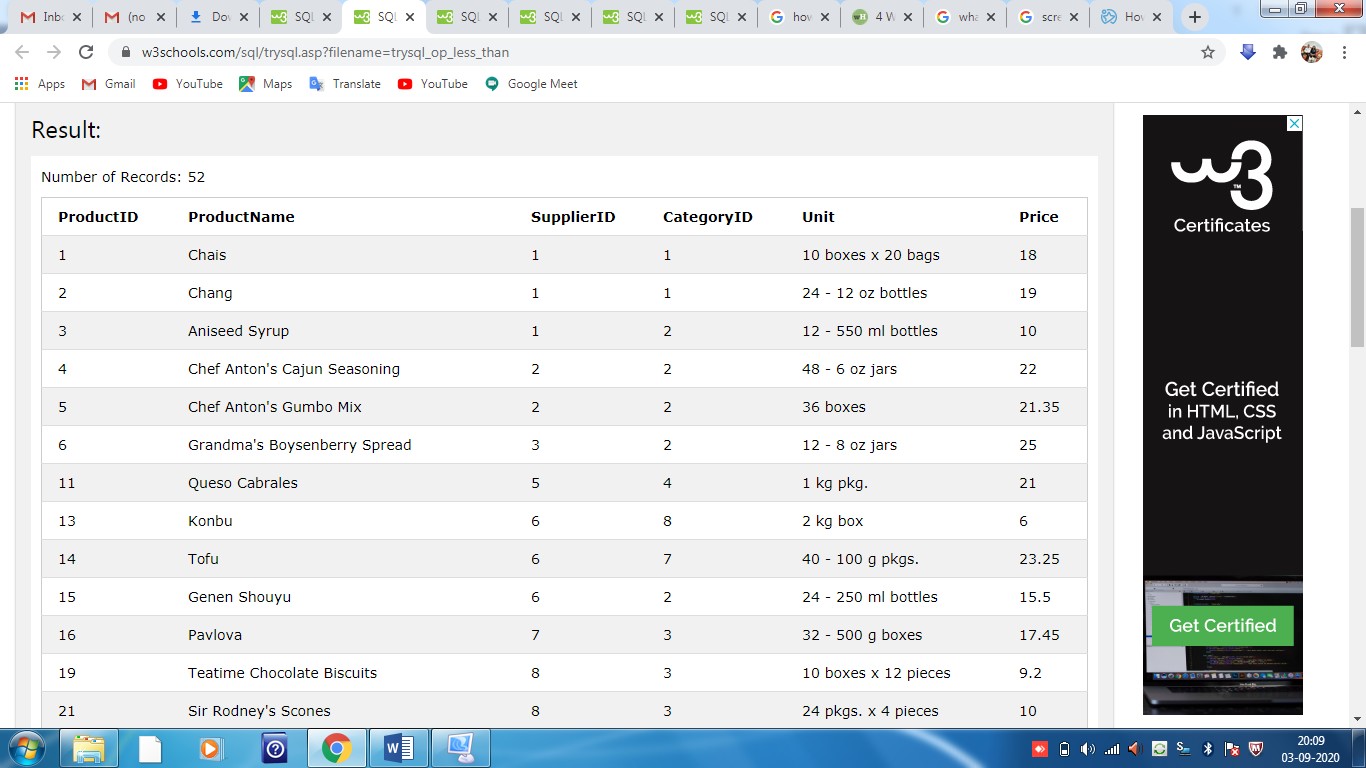
**SELECT \* FROM Products**

**WHERE Price > 30;**

****

**SELECT \* FROM Products**

**WHERE Price < 30;**

****

**SELECT \* FROM Products**

**WHERE Price >= 30;**

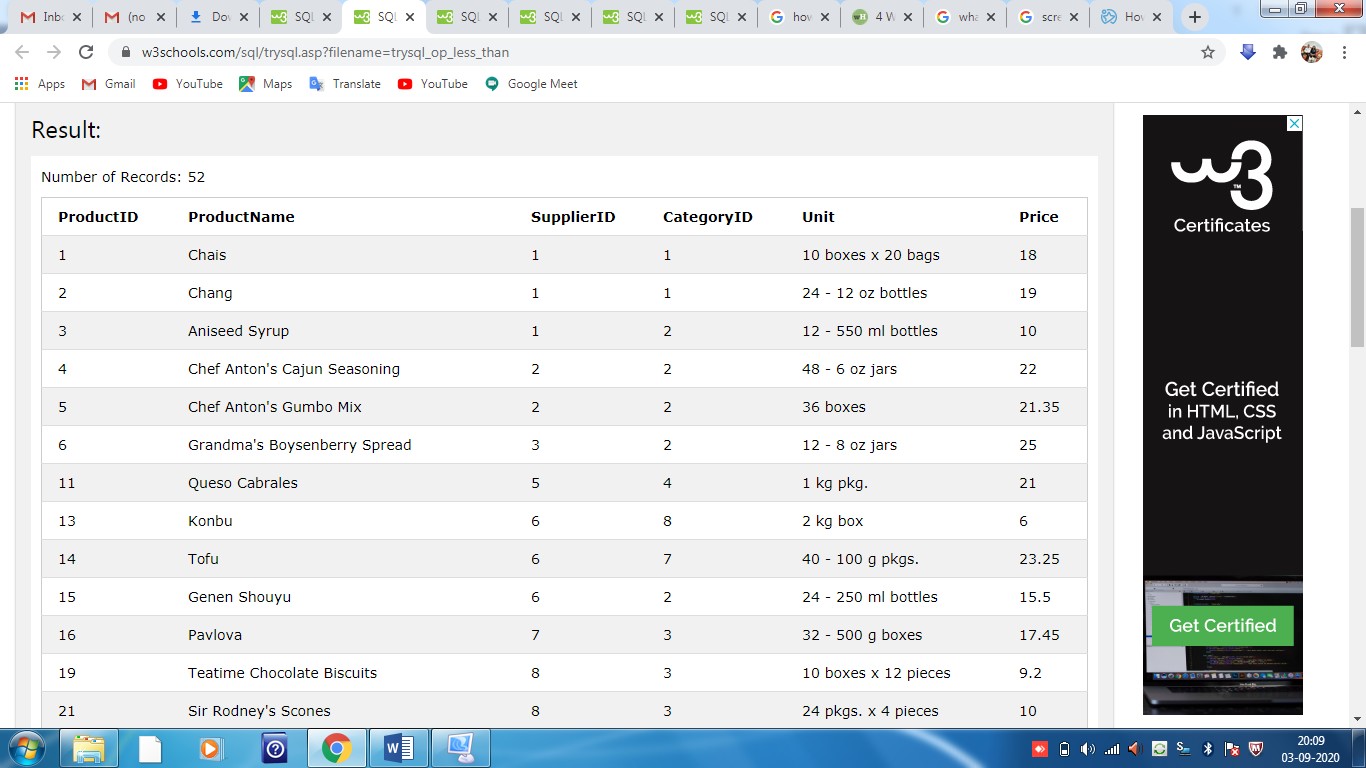
**SELECT \* FROM Products**

**WHERE Price <= 30;**

**SELECT \* FROM Products**

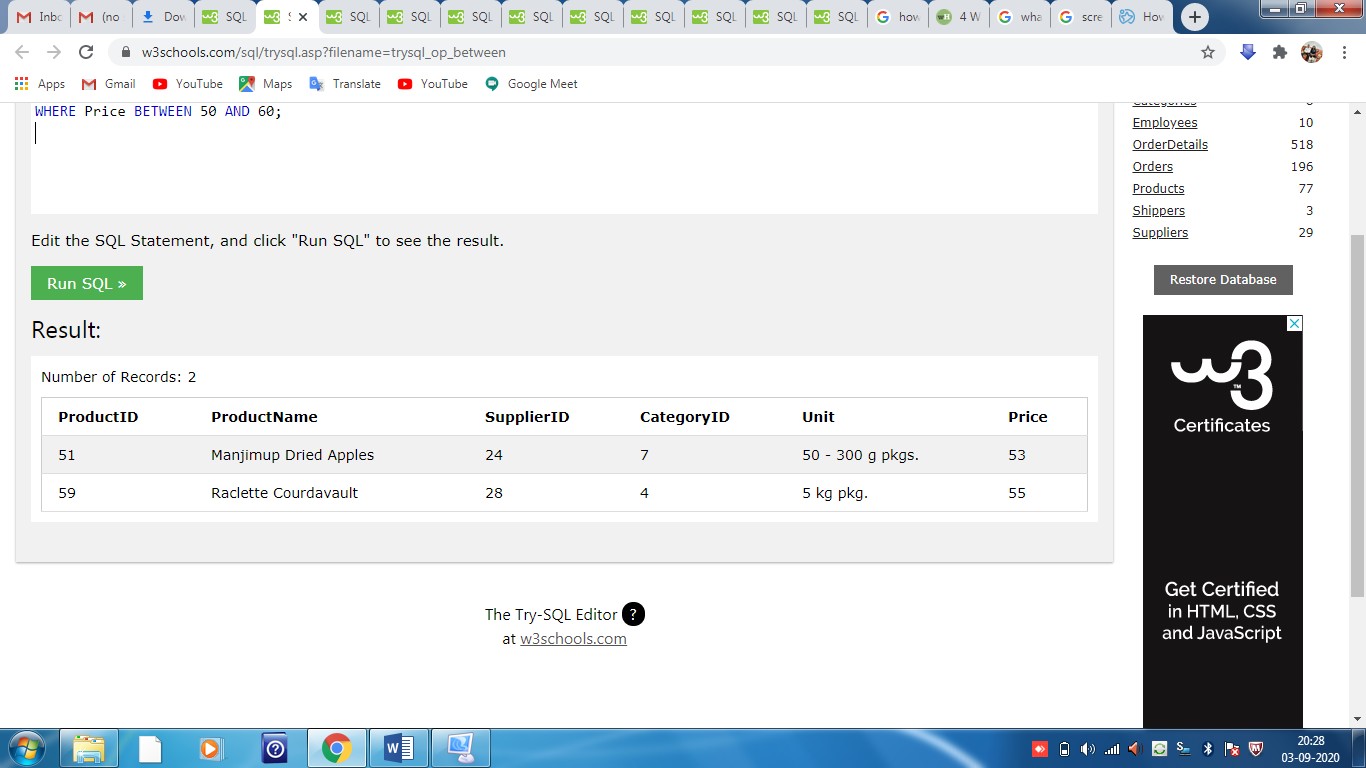
**WHERE Price <> 18;**

**Note - <> means !=(not equal)**

****

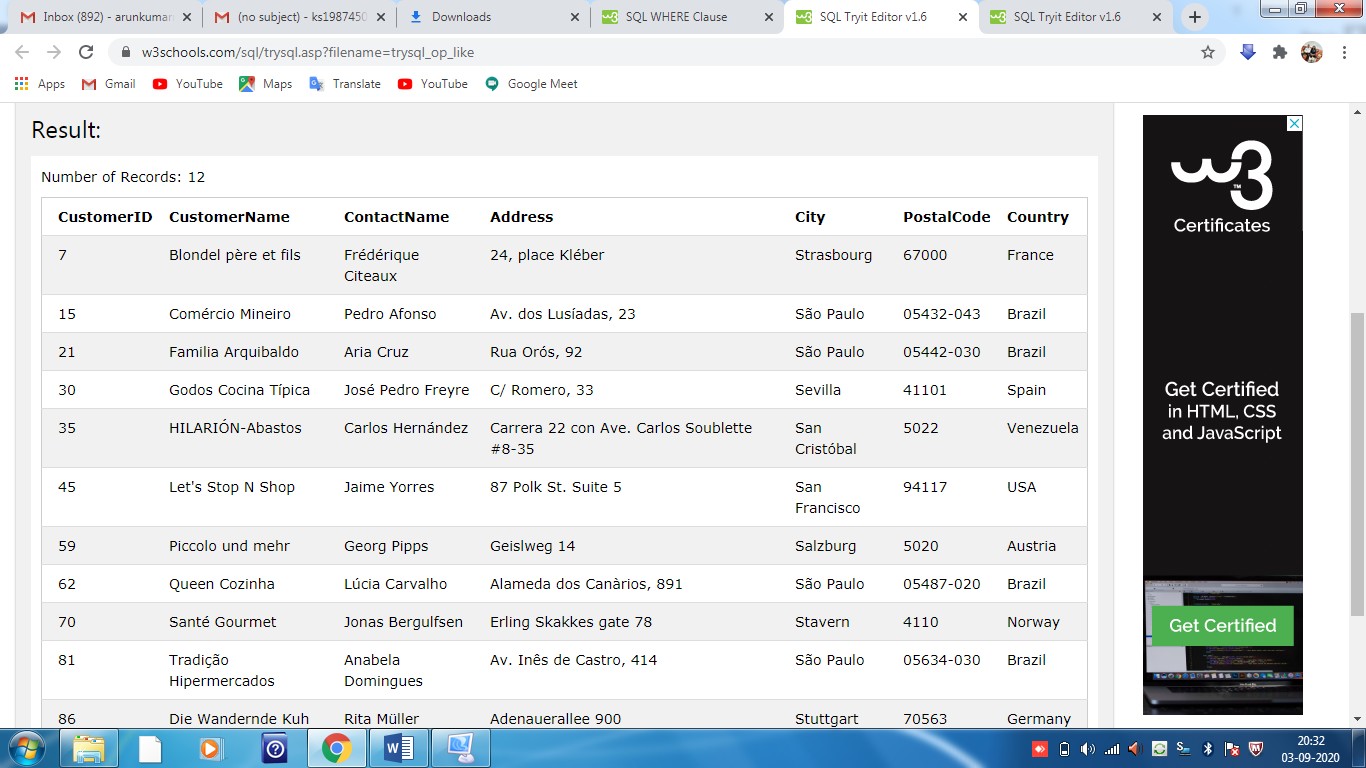
**SELECT \* FROM Products**

**WHERE Price BETWEEN 50 AND 60;**

****

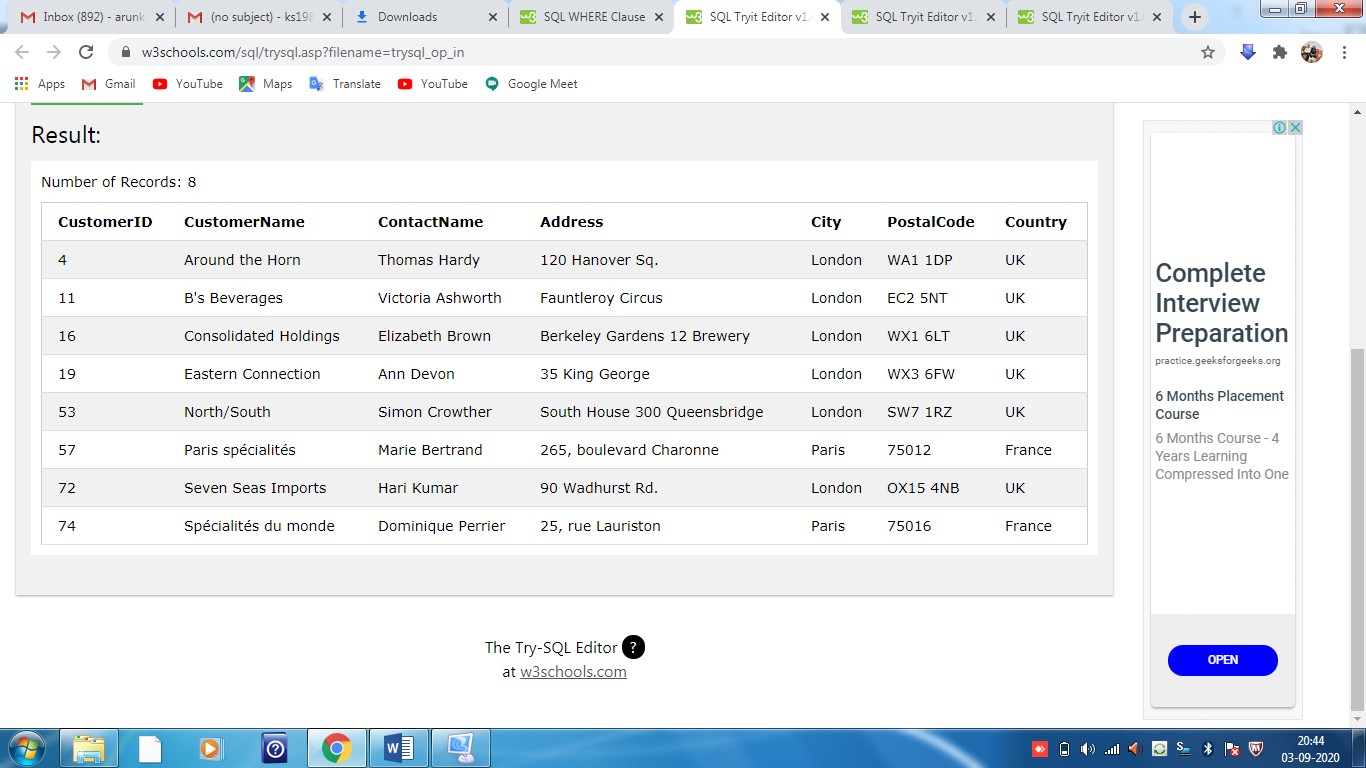
**SELECT \* FROM Customers**

**WHERE City LIKE 's%';**

****

**SELECT \* FROM Customers**

**WHERE City IN ('Paris','London');**

****

The SQL AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

* The AND operator displays a record if all the conditions separated by AND are TRUE.
* The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

### AND Syntax

SELECT column1, column2, ...  
FROM table\_name  
WHERE condition1 AND condition2 AND condition3 ...;

### OR Syntax

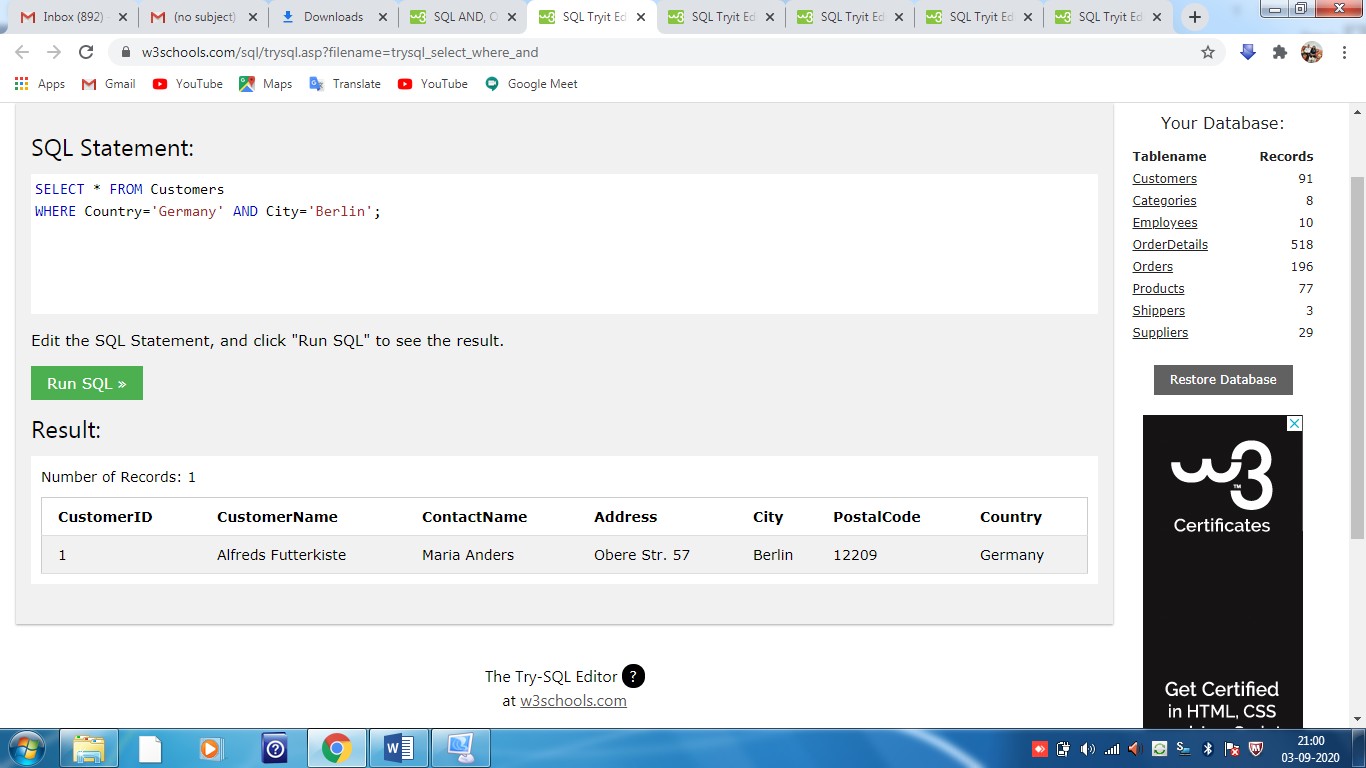
SELECT column1, column2, ...  
FROM table\_name  
WHERE condition1 OR condition2 OR condition3 ...;

### NOT Syntax

SELECT column1, column2, ...  
FROM table\_name  
WHERE NOT condition;

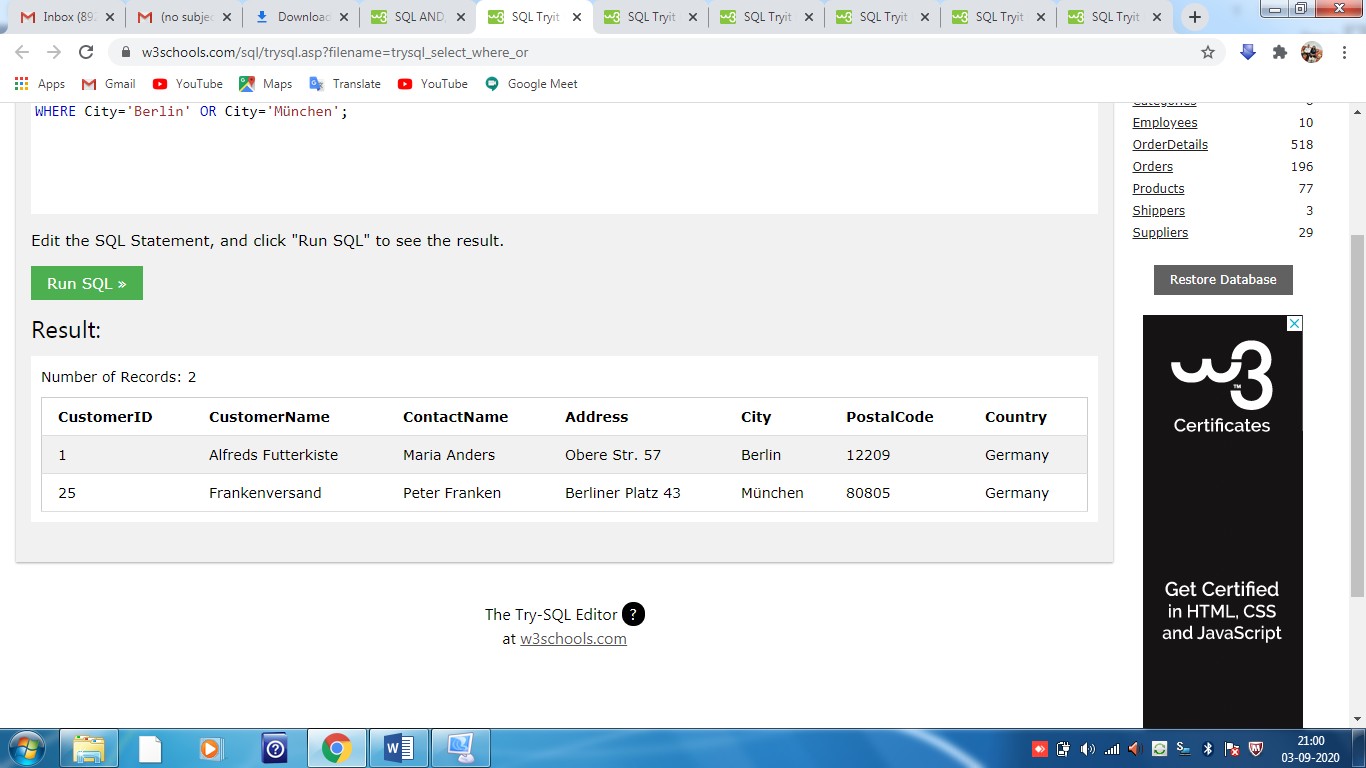
## AND Example

SELECT \* FROM Customers  
WHERE Country='Germany' AND City='Berlin';



## OR Example

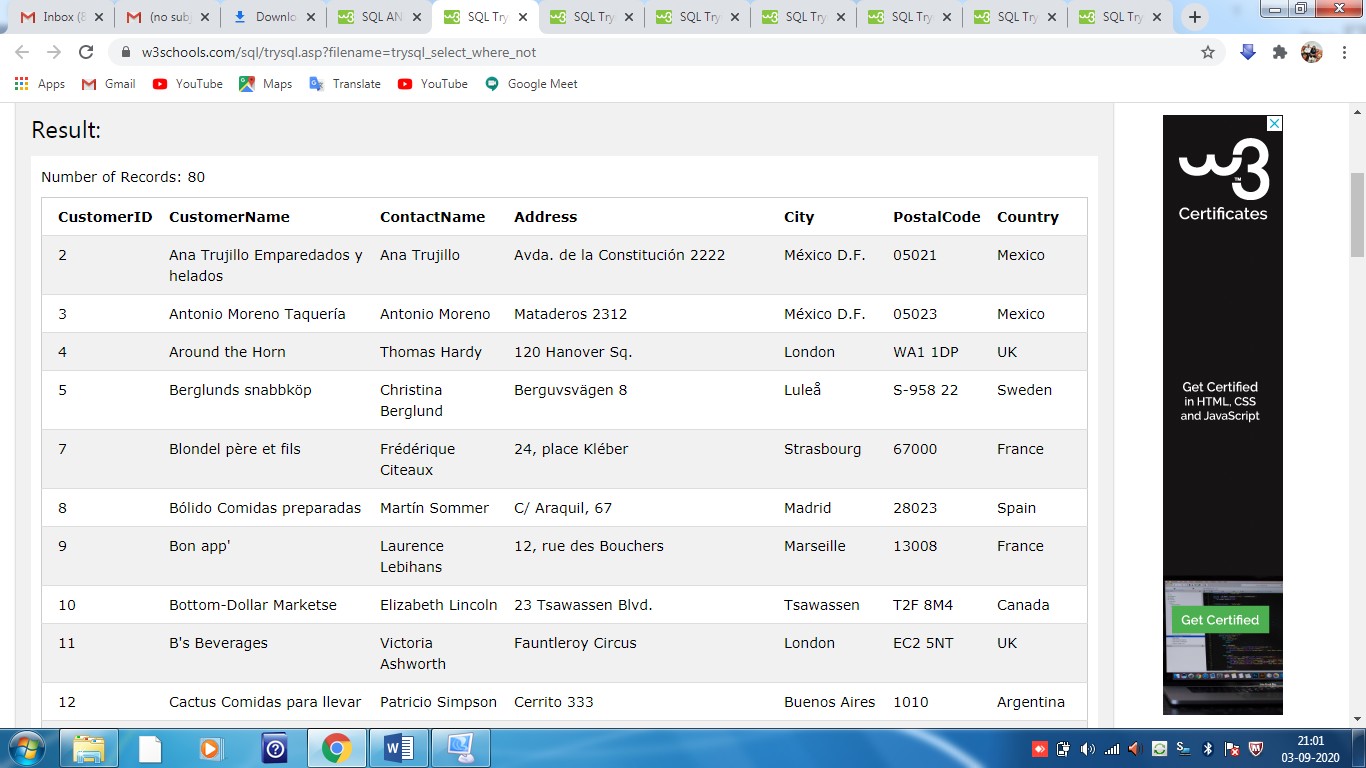
SELECT \* FROM Customers  
WHERE City='Berlin' OR City='München';

****

## NOT Example

The following SQL statement selects all fields from "Customers" where country is NOT "Germany":

SELECT \* FROM Customers  
WHERE NOT Country='Germany';

****

SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

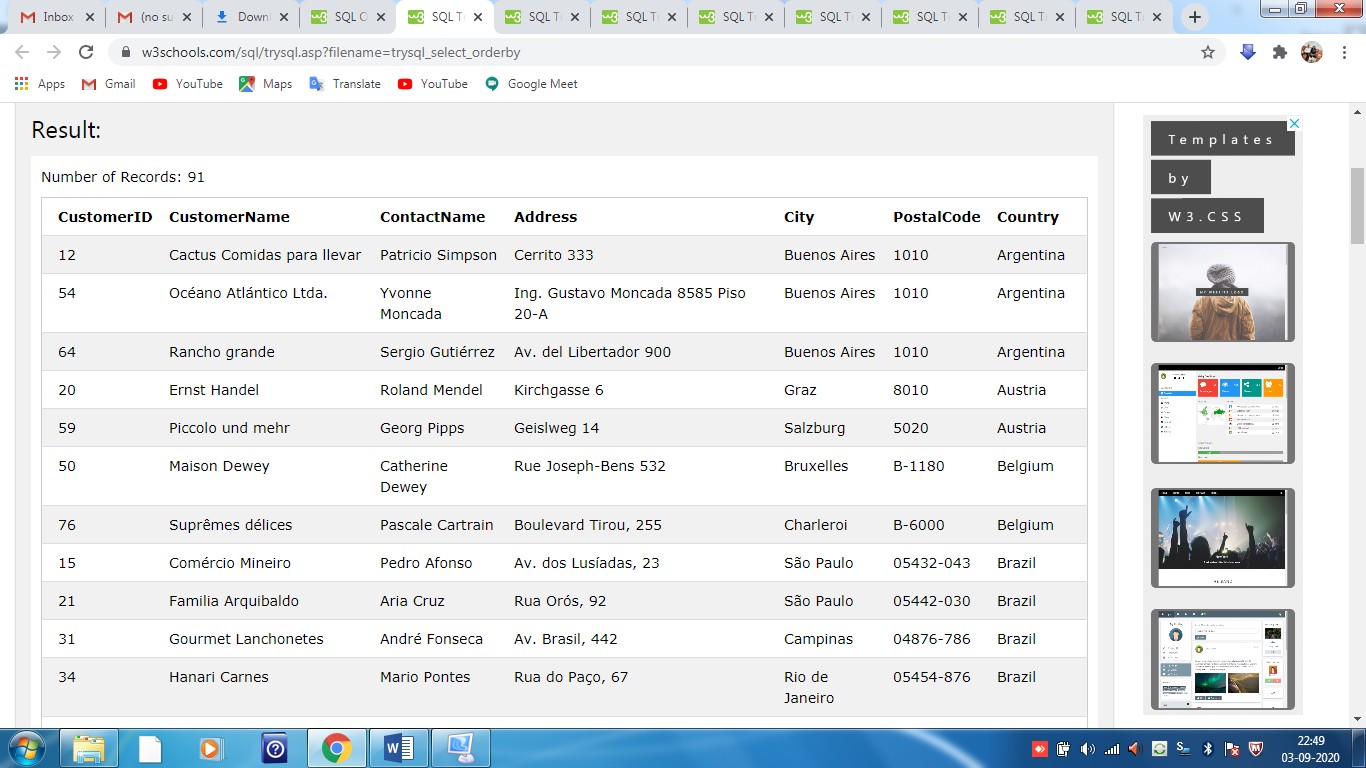
The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

### ORDER BY Syntax

SELECT column1, column2, …  
FROM table\_name  
ORDER BY column1, column2, ... ASC|DESC;

Example 1 - ORDER BY

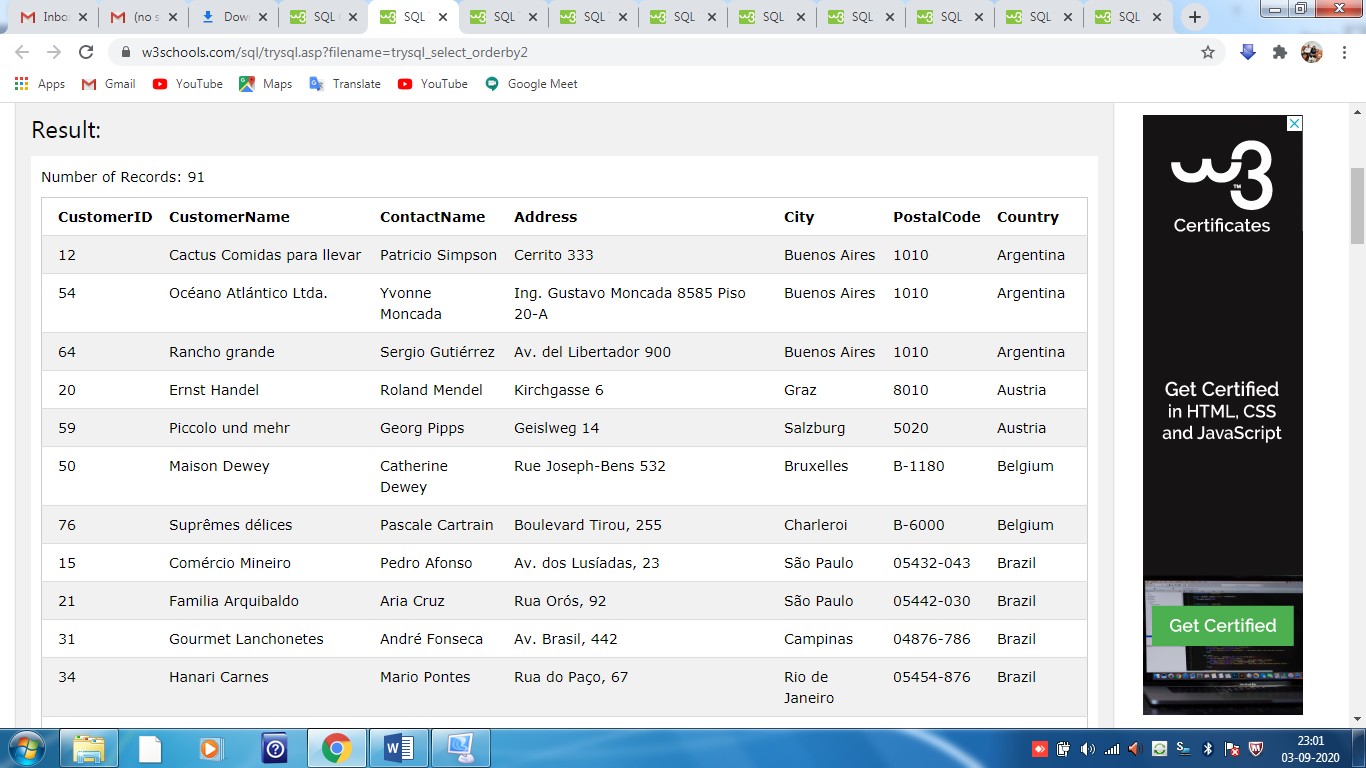
SELECT \* FROM Customers  
ORDER BY Country;



## Example 2 - ORDER BY Several Columns

## The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column. This means that it orders by Country, but if some rows have the same Country, it orders them by CustomerName:

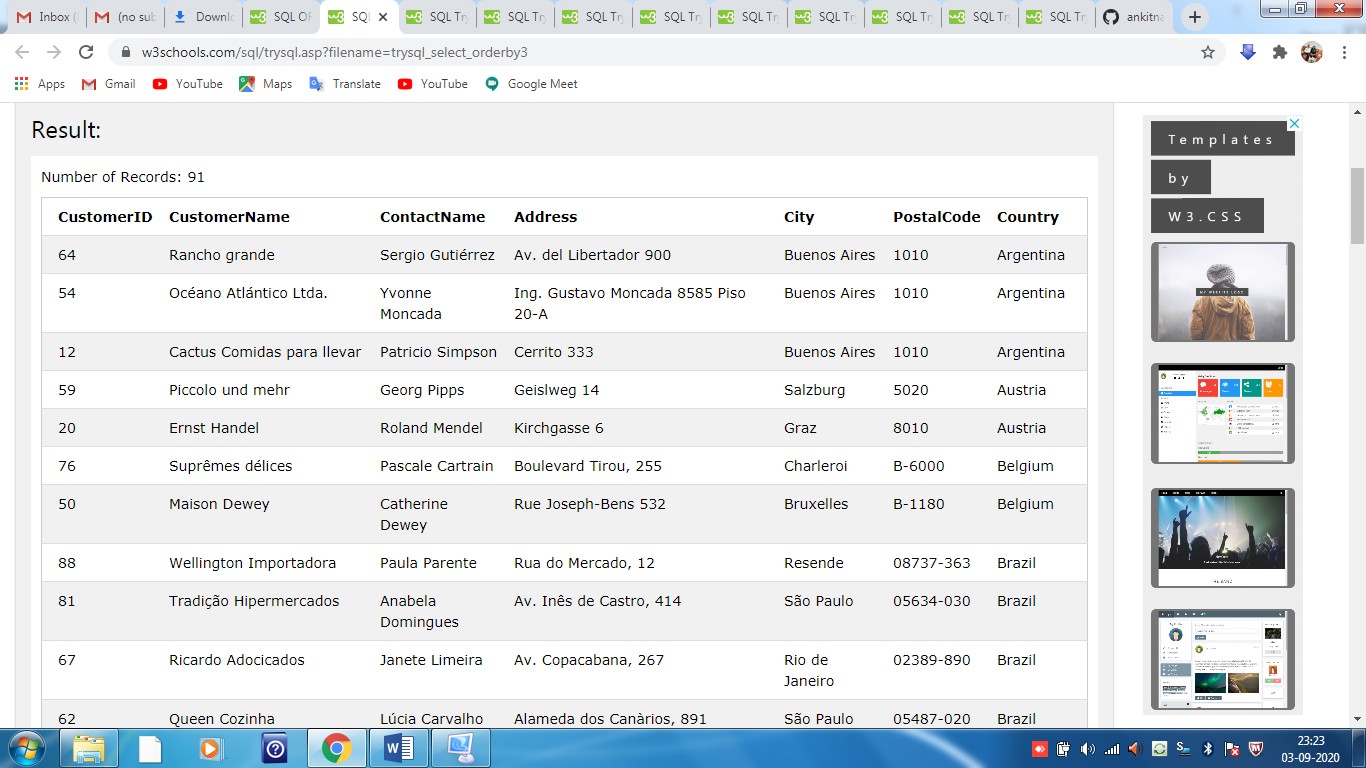
## SELECT \* FROM Customers ORDER BY Country, CustomerName;



## Example 2 - ORDER BY Several Columns

The following SQL statement selects all customers from the "Customers" table, sorted **ascending** by the **"Country"** and **descending** by the **"CustomerName"** column:

SELECT \* FROM Customers  
ORDER BY Country ASC, CustomerName DESC;

****