# SQL Learning

W3schools

<https://www.w3schools.com/sql/sql_intro.asp>

## What is SQL?

* Structured Query Language
* Access and manipulate database

## What can SQL do?

SQL can execute queries against a database

SQL can retrieve data from a database

SQL can insert records in a database

SQL can update records in a database

SQL can delete records from a database

SQL can create new databases

SQL can create new tables in a database

SQL can create stored procedures in a database

SQL can create views in a database

SQL can set permissions on tables, procedures, and views

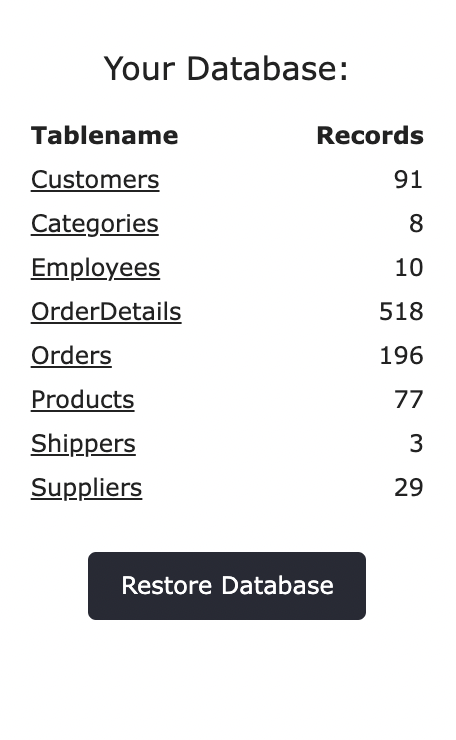
## RDBMS

* Relational Database Management System

## Select column

* Select **everything** from **Customers database**

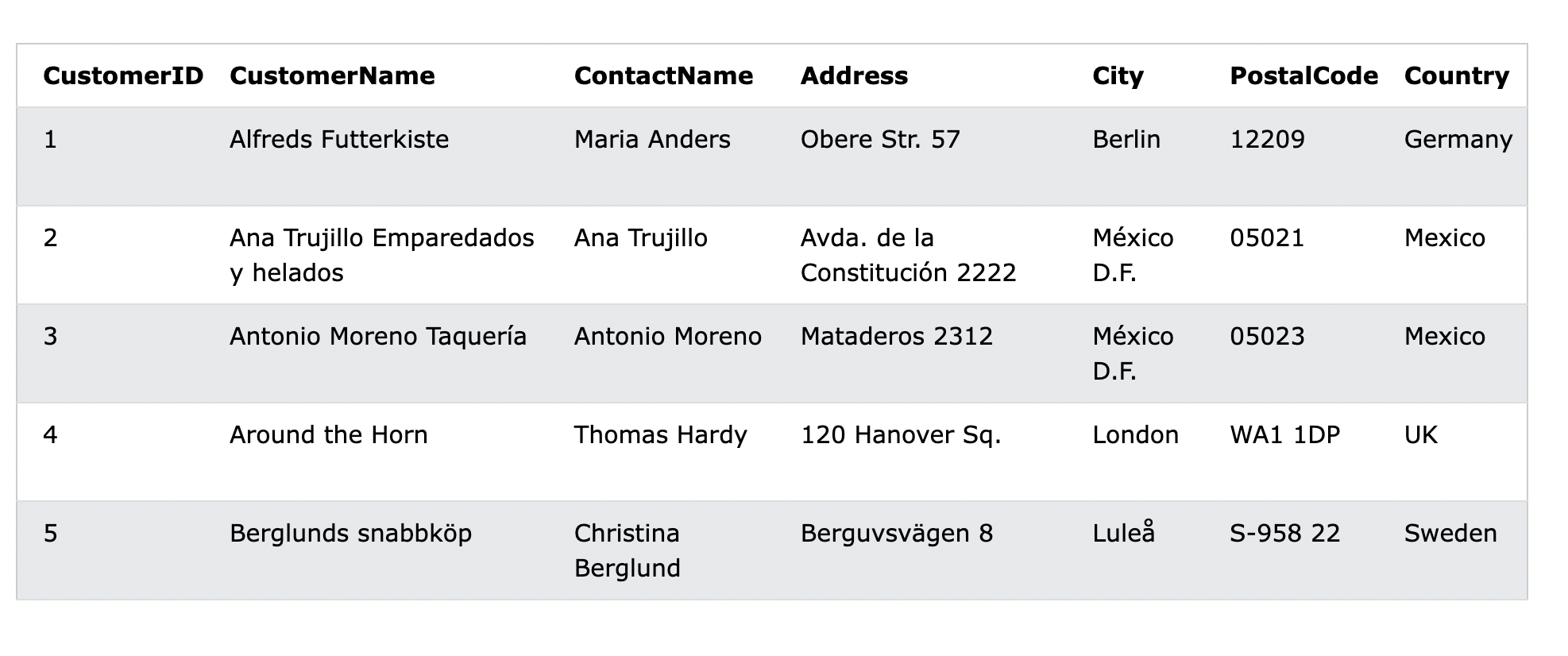
SELECT \* FROM Customers;



# SQL Syntax

## Database table

* Column and rows



## Most common SQL command

**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

## SELECT

SELECT column1, column2, ...

FROM table\_name;

* Select column from table

SELECT \* FROM table\_name;

* Select all column from table

### Select column

SELECT CustomerName,City FROM Customers;



## SELECT DISTINCT Statement

SELECT DISTINCT column1, column2, ...

FROM table\_name;

* SELECT DISTINCT return only distinct (different) values

SELECT Country FROM Customers;

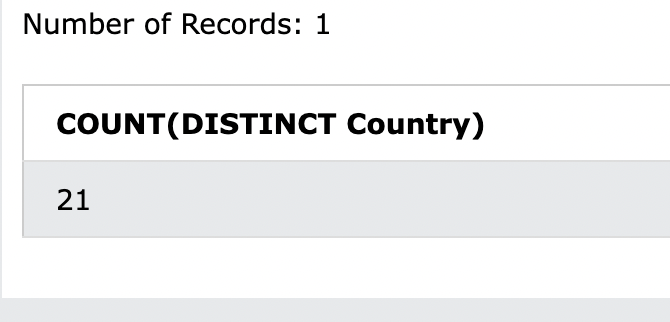


SELECT DISTINCT Country FROM Customers;



SELECT COUNT(DISTINCT Country) FROM Customers;

* Count



## Where

* WHERE clause is used to filter records

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

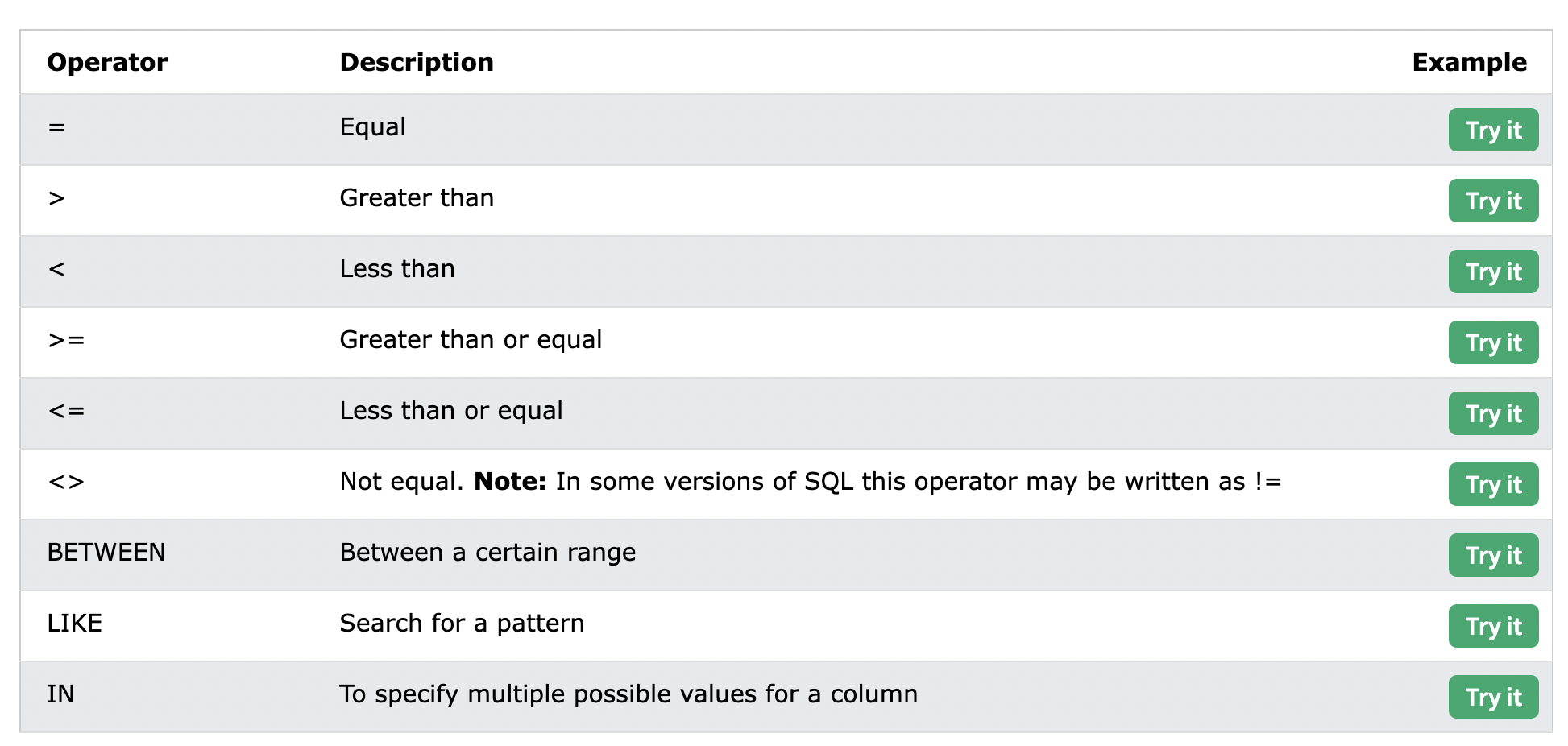
* Filter string

#Filter Country = Mexico

SELECT \* FROM Customers

WHERE Country='Mexico';

**Operators**



## And Or Not (Combine with where)

* Filter base on more than one conditions
* same logic as python
* And = everything need to be true
* or = one need to be true
* not = reverse and

### And syntax

SELECT column1, column2, ...

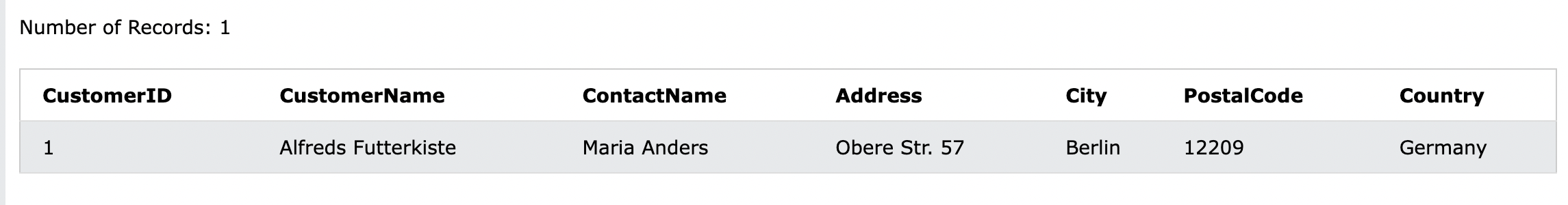
FROM table\_name

WHERE condition1 AND condition2 AND condition3 ...;

**Example**

SELECT \* FROM Customers

WHERE Country='Germany' AND City='Berlin';



### OR syntax

SELECT column1, column2, ...

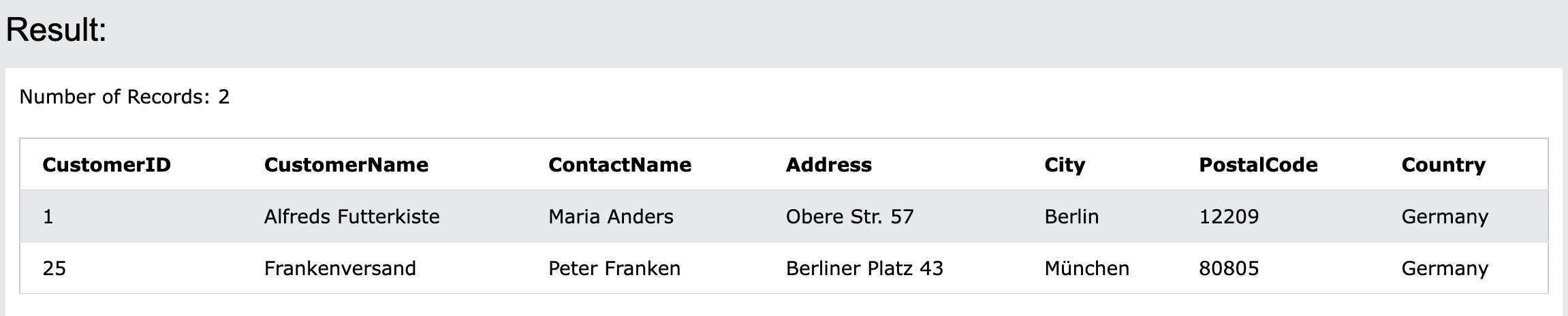
FROM table\_name

WHERE condition1 OR condition2 OR condition3 ...;

**Example**

SELECT \* FROM Customers

WHERE City='Berlin' OR City='München';



### Not Syntax

SELECT column1, column2, ...

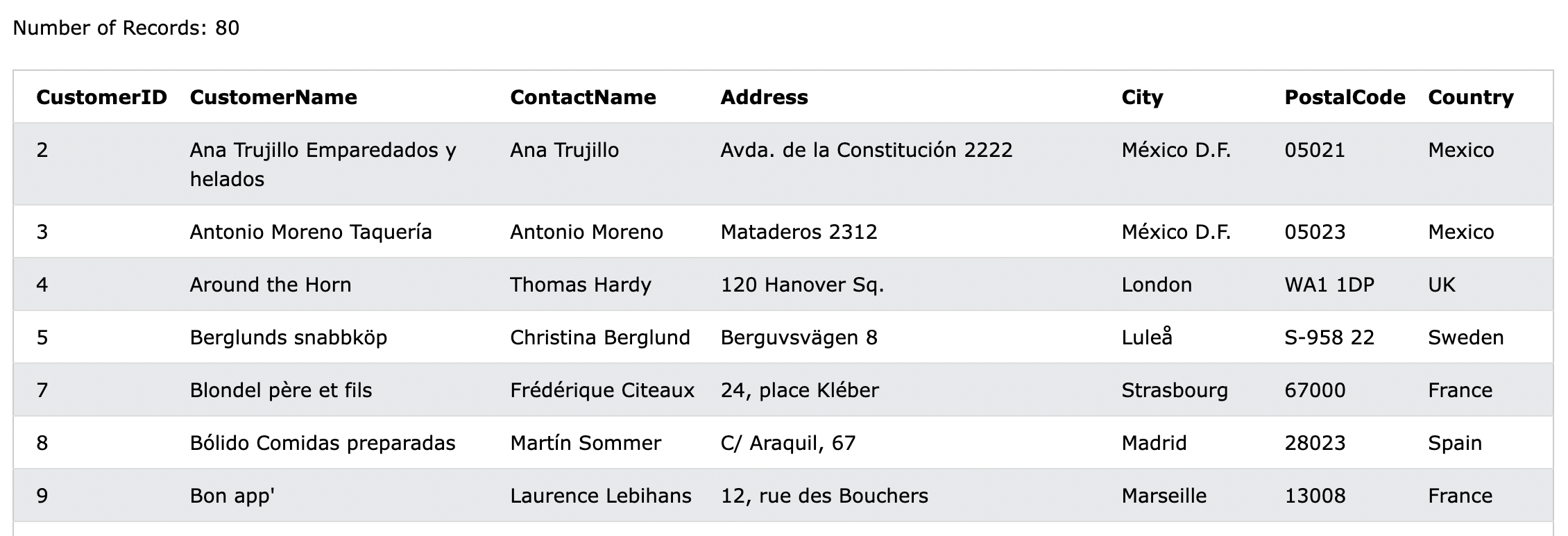
FROM table\_name

WHERE NOT condition;

**Example**

SELECT \* FROM Customers

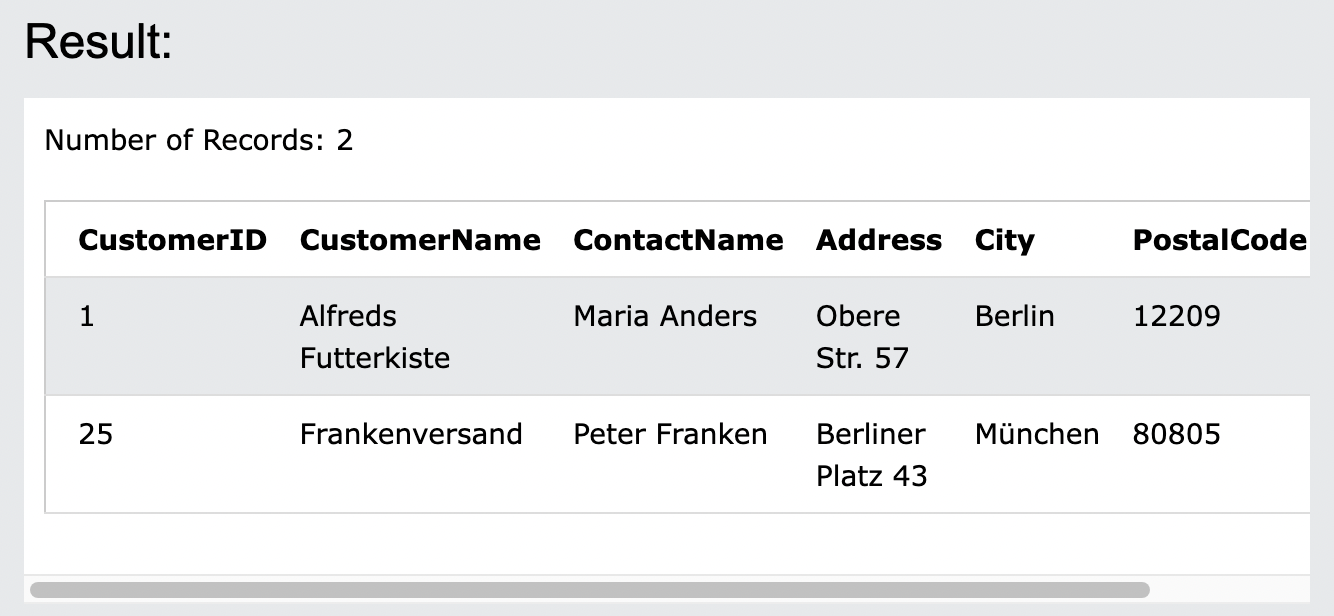
WHERE NOT Country='Germany';



### Combination

SELECT \* FROM Customers

WHERE Country='Germany' AND (City='Berlin' OR City='München');



## ORDER BY

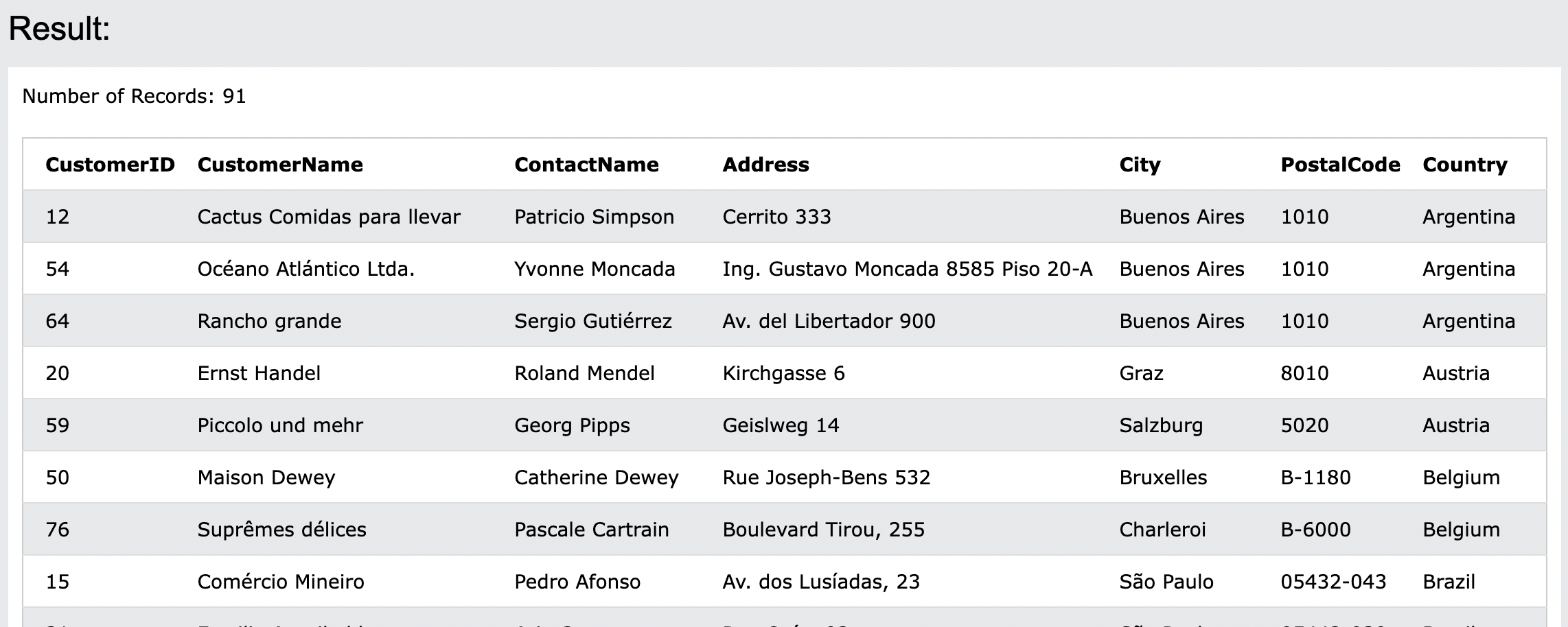
SELECT column1, column2, ...

FROM table\_name

ORDER BY column1, column2, ... ASC|DESC;

SELECT \* FROM Customers

ORDER BY Country;

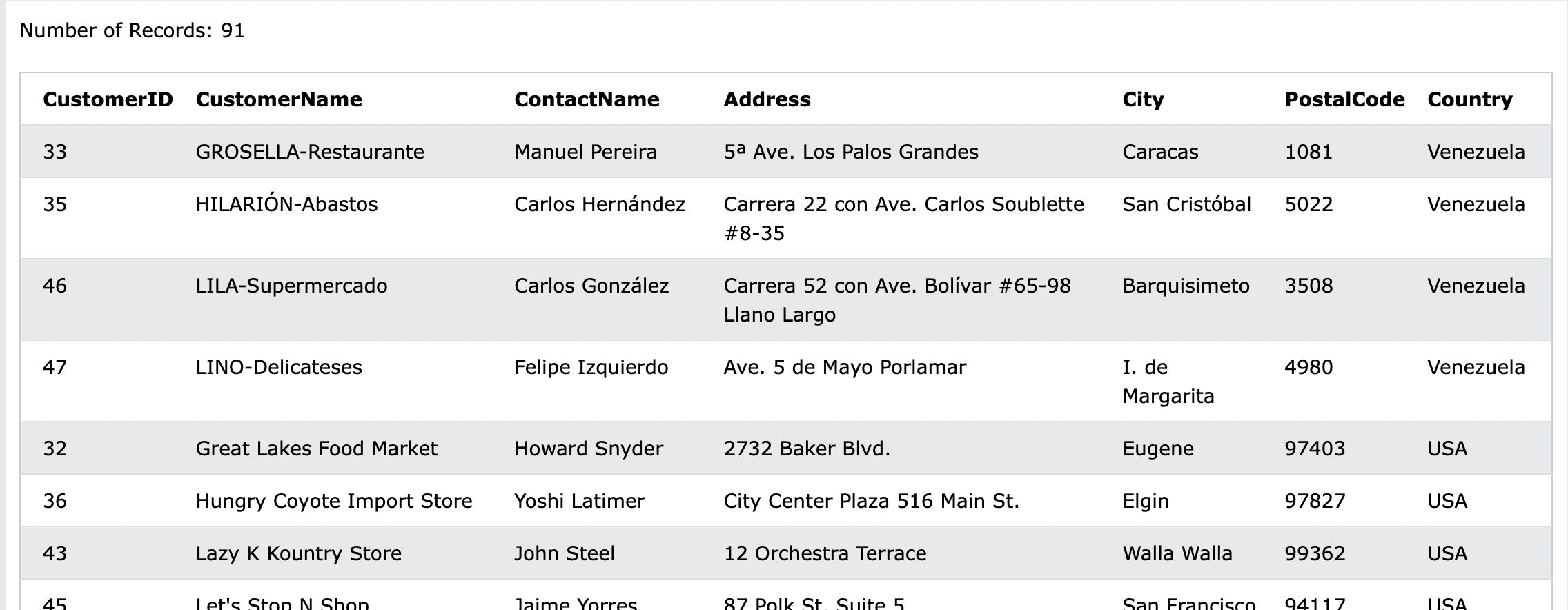


* Alphabetical order

### ORDER BY DESC

SELECT \* FROM Customers

ORDER BY Country DESC;



* Reverse alphabetical order

### INSERT INTO

insert new records in a table.

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

* Choose specific column and value

INSERT INTO table\_name

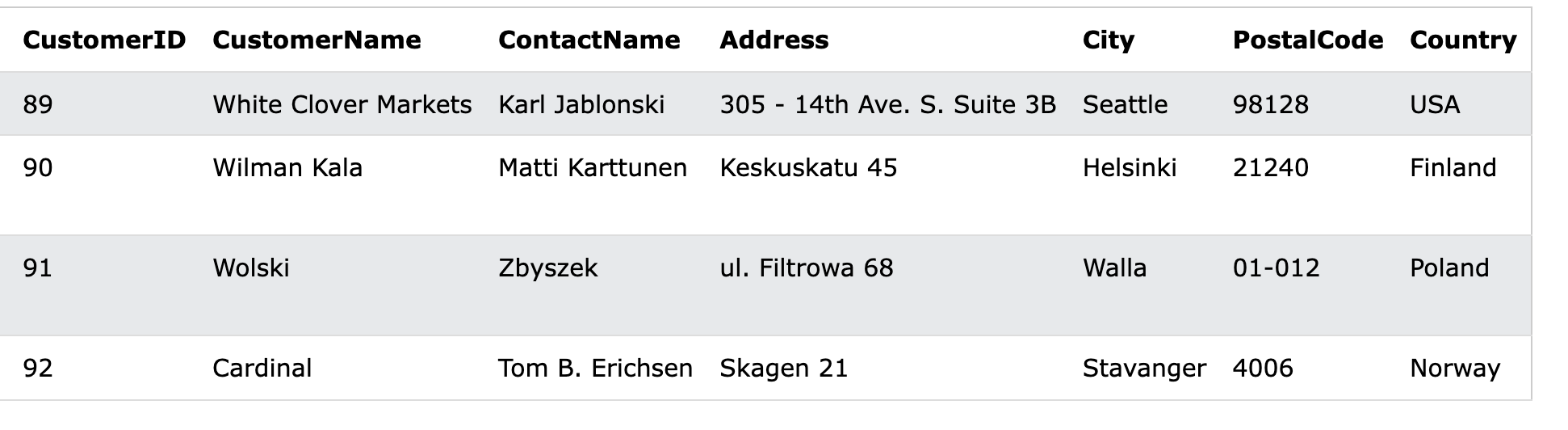
VALUES (value1, value2, value3, ...);

* assign automatically

**Example**

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

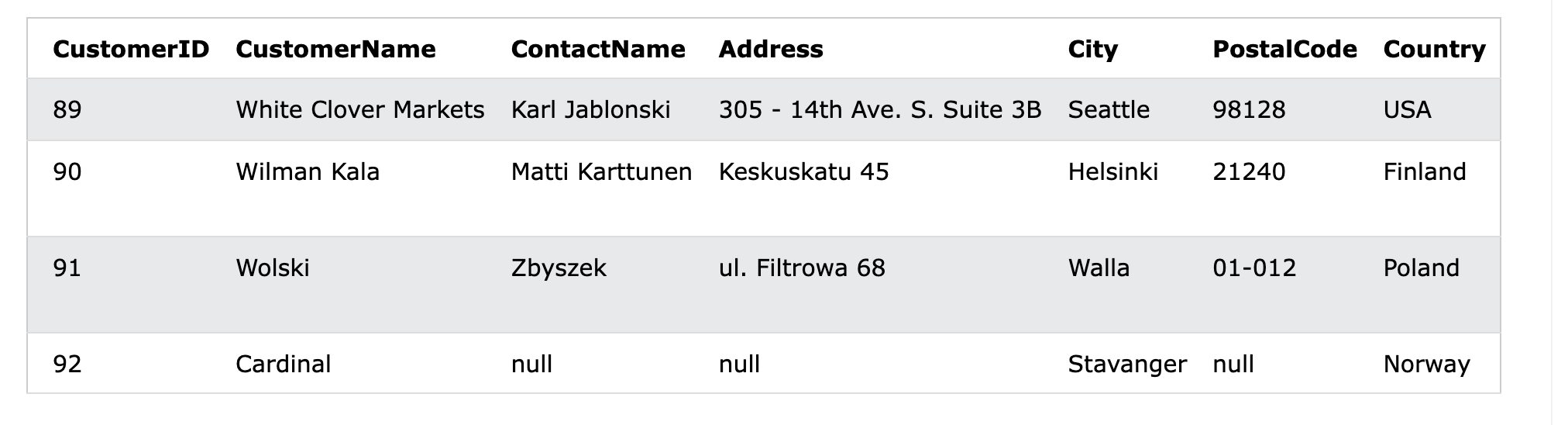
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');



### Only in Specified Columns

INSERT INTO Customers (CustomerName, City, Country)

VALUES ('Cardinal', 'Stavanger', 'Norway');



## NULL Values

* null = a field with no value

**Find null column**

SELECT column\_names

FROM table\_name

WHERE column\_name IS NULL;

**Find non-null**

SELECT column\_names

FROM table\_name

WHERE column\_name IS NOT NULL;

## UPDATE

modify the existing records in a table.

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

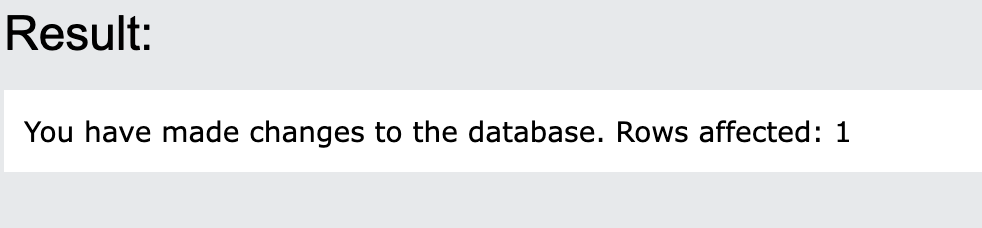
WHERE condition;

**Update a single row**

UPDATE Customers

SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'

WHERE CustomerID = 1;



## DELETE

* delete existing records

DELETE FROM table\_name WHERE condition;

DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

* Delete Row where CusomterName = ‘Alfreds Futterkiste’

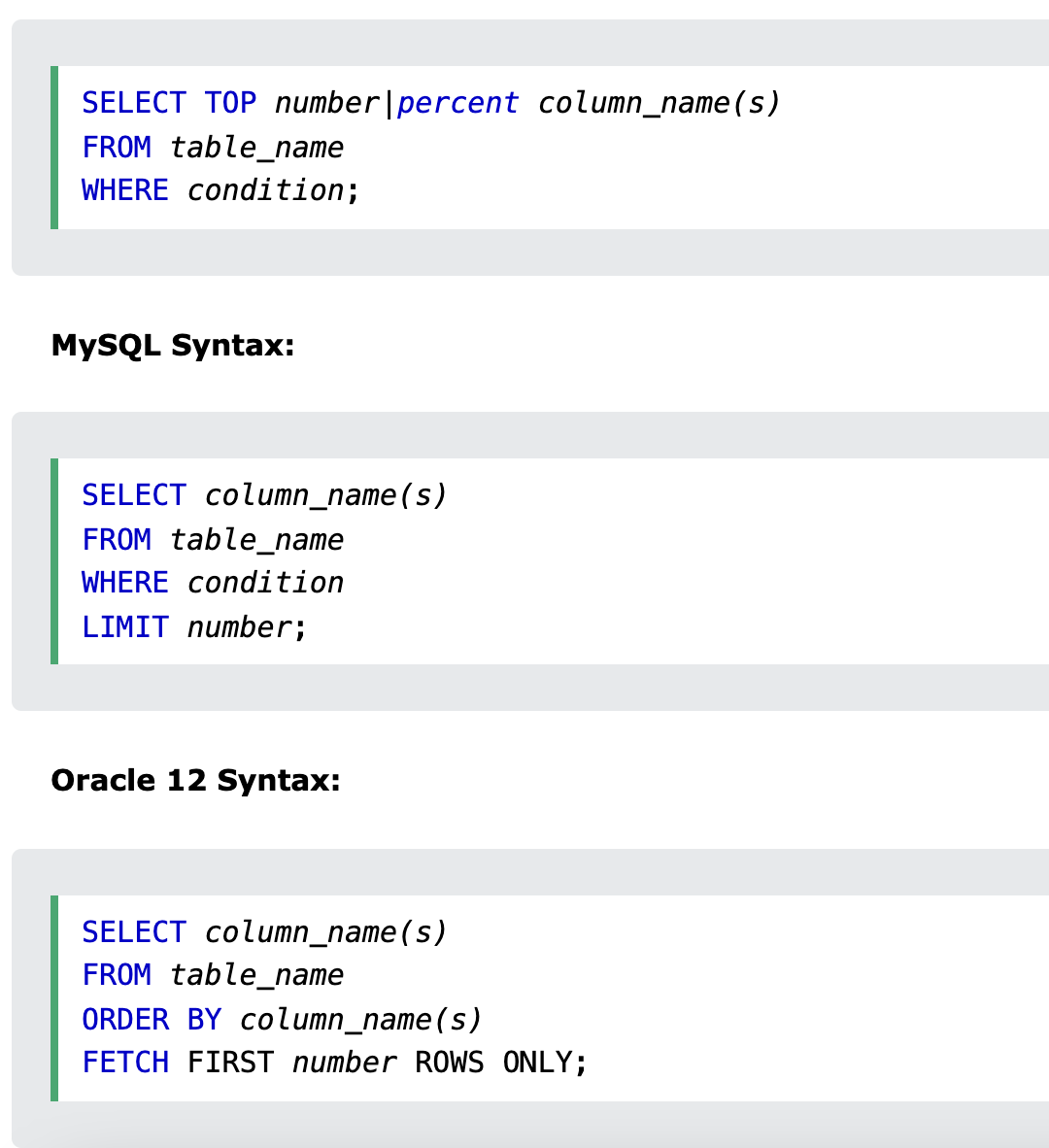
**Delete all rows and table**

DELETE FROM table\_name;

## TOP, LIMIT, FETCH FIRST or ROWNUM

**SELECT TOP**

* Specify the number of records to return



SELECT TOP 3 \* FROM Customers;

* Select top 3 from customers

**TOP PERCENT**

SELECT TOP 50 PERCENT \* FROM Customers;

## MIN() and MAX()

* MIN() return the smallest value

SELECT MIN(column\_name)

FROM table\_name

WHERE condition;

* MAX() returns the largest value

SELECT MAX(column\_name)

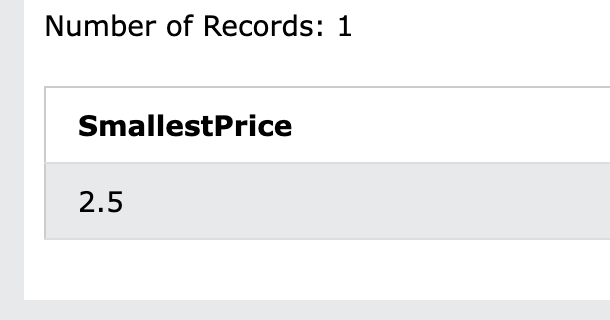
FROM table\_name

WHERE condition;

**Example**

SELECT MIN(Price) AS SmallestPrice

FROM Products;



## COUNT(),AVG() and SUM()

### COUNT()

SELECT COUNT(column\_name)

FROM table\_name

WHERE condition;

#### AVG()

SELECT AVG(column\_name)

FROM table\_name

WHERE condition;

### SUM()

SELECT SUM(column\_name)

FROM table\_name

WHERE condition;

**example**

SELECT COUNT(ProductID)

FROM Products;



* Count product id from products

## LIKE Operator

LIKE

* use in a WHERE
* To search a **specified pattern**

SELECT column1, column2, ...

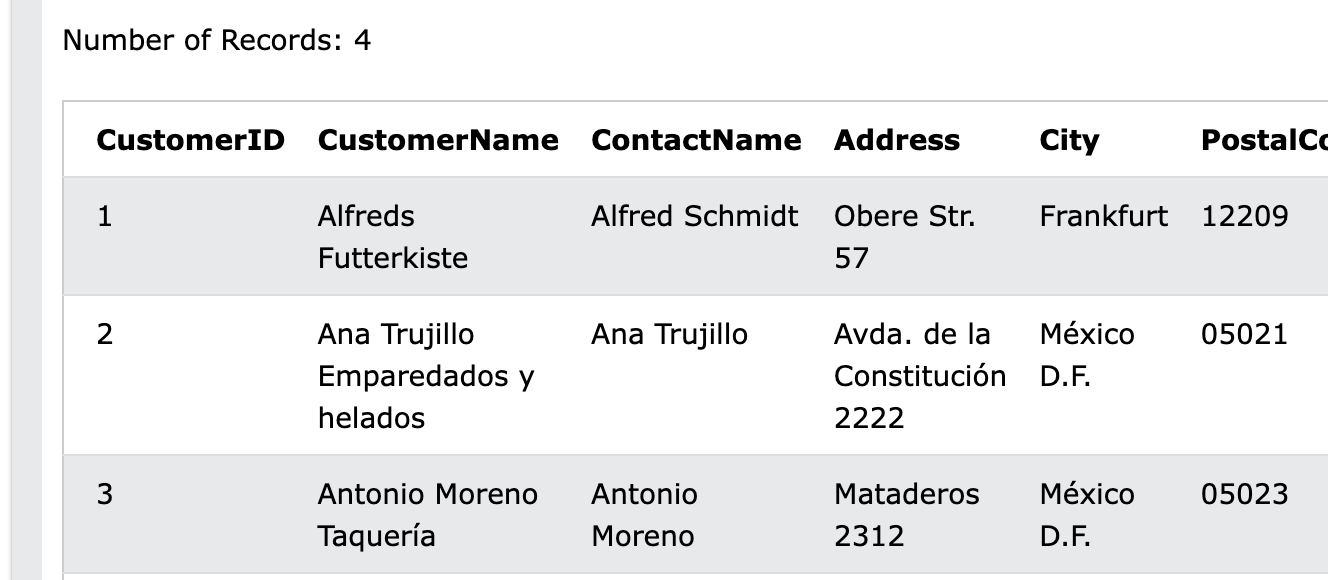
FROM table\_name

WHERE columnN LIKE pattern;

**Starting with a**

SELECT \* FROM Customers

WHERE CustomerName LIKE 'a%';



**End with a**

SELECT \* FROM Customers

WHERE CustomerName LIKE '%a';

**r in second position**

SELECT \* FROM Customers

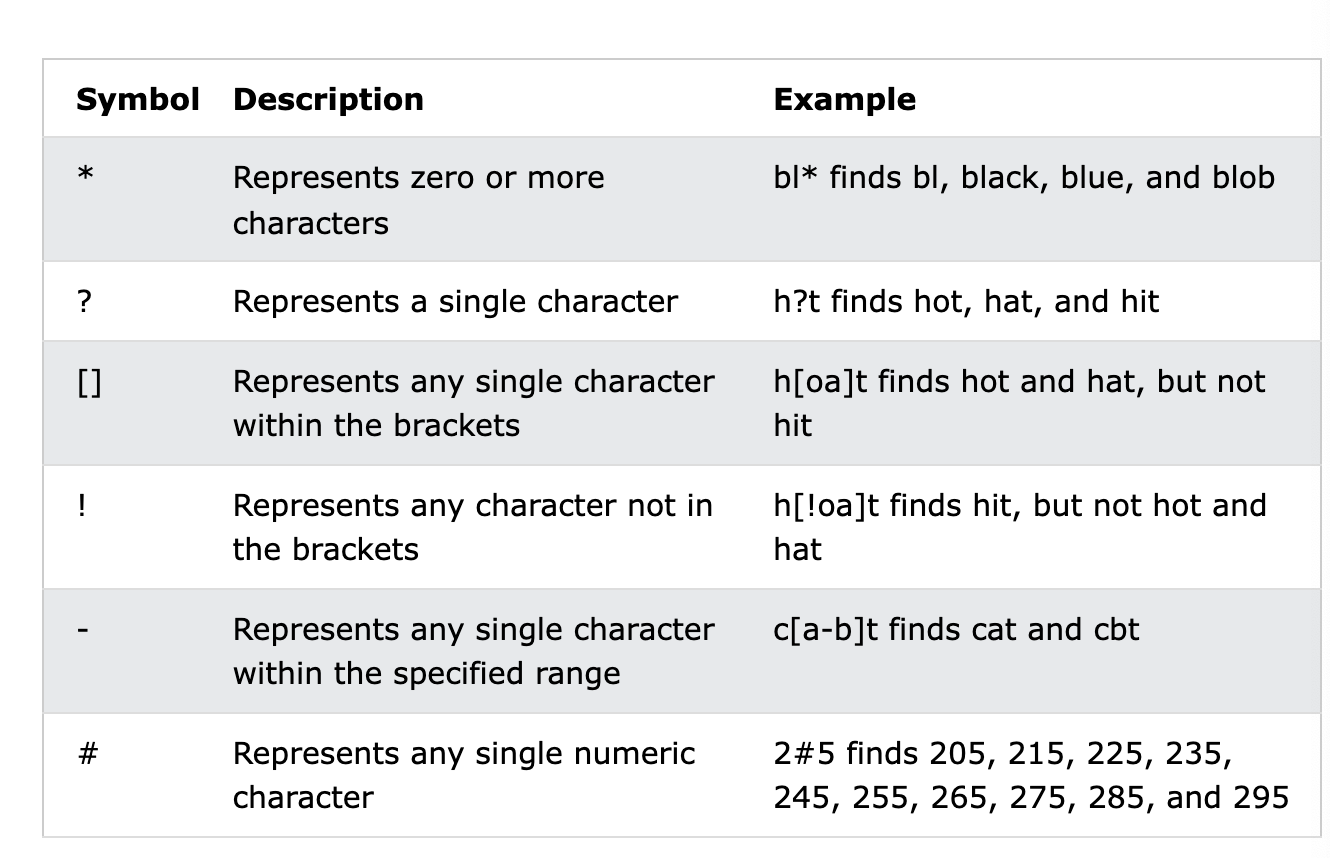
WHERE CustomerName LIKE '\_r%';

**not a**

SELECT \* FROM Customers

WHERE CustomerName NOT LIKE 'a%';

## Wildcards



## IN

* IN operator > specify multiple values in a WHERE clause
* shorthand for multiple **OR** conditions

SELECT column\_name(s)

FROM table\_name

WHERE column\_name IN (value1, value2, ...);

Or

SELECT column\_name(s)

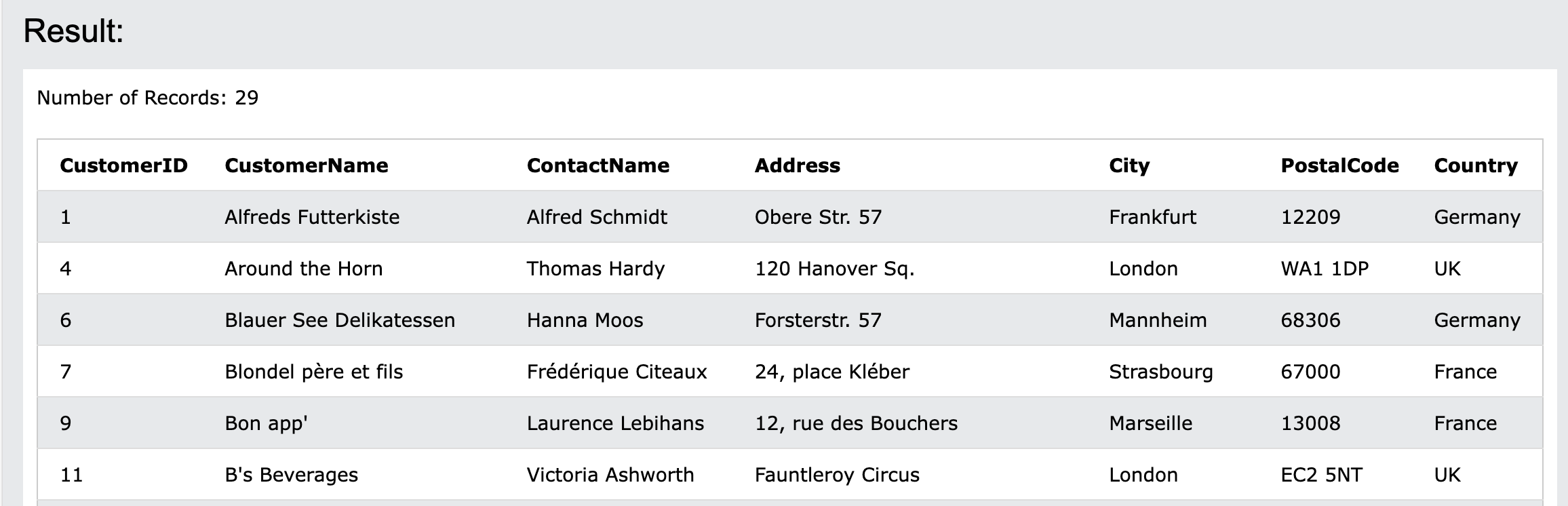
FROM table\_name

WHERE column\_name IN (SELECT STATEMENT);

**Example**

SELECT \* FROM Customers

WHERE Country IN ('Germany', 'France', 'UK');



## Between

* selects values within a given range
  + numbers, text or dates
  + begin and end values

SELECT column\_name(s)

FROM table\_name

WHERE column\_name BETWEEN value1 AND value2;

**Example**

SELECT \* FROM Products

WHERE Price BETWEEN 10 AND 20;

SELECT \* FROM Products

WHERE ProductName BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'

ORDER BY ProductName;

## Aliases

* give a table, column a temporary name
* Make column names more readable

SELECT column\_name AS alias\_name

FROM table\_name;

SELECT column\_name(s)

FROM table\_name AS alias\_name;