

SQL Introduction >

SQL SELECT (I) ^

✓

SQL SELECT

✓

SQL AND, OR, NOT

✓

SQL SELECT DISTINCT

✓

SQL SELECT AS

✓

SQL LIMIT, TOP, FETCH FIRST

✓

SQL IN Operator

✓

SQL BETWEEN Operator

✓

SQL IS NULL and NOT NULL

✓

SQL MIN() and MAX()

✓

SQL COUNT()

✓

SQL SUM() and AVG()

SQL SELECT (II) >

SQL JOIN >

SQL DATABASE & TABLE >

SQL Insert, Update and Delete >

SQL Constraints >

SQL Additional Topics >

Related Topics

SQL SELECT

SQL SELECT INTO Statement

SQL Subquery

SQL INSERT INTO SELECT Statement

SQL AND, OR, and NOT Operators

SQL GROUP BY

SQL IN Operator

In this tutorial, we'll learn about the SQL IN operator with the help of examples.

The `IN` operator is used with the [WHERE](#) clause to match values in a list. For example,

```
SELECT first_name, country
FROM Customers
WHERE country IN ('USA', 'UK');
```

Run Code >>

Here, the SQL command selects rows if the `country` is either **USA** or **UK**.



SQL IN Operator With Columns

The `IN` operator can also be used to select rows in which a certain value exists in the given field. Let's see an example to clarify it.

```
SELECT first_name, country
FROM Customers
WHERE 'USA' IN (country);
```

Run Code >>

Here, the SQL command selects the rows if the **USA** value exists in the `country` field.



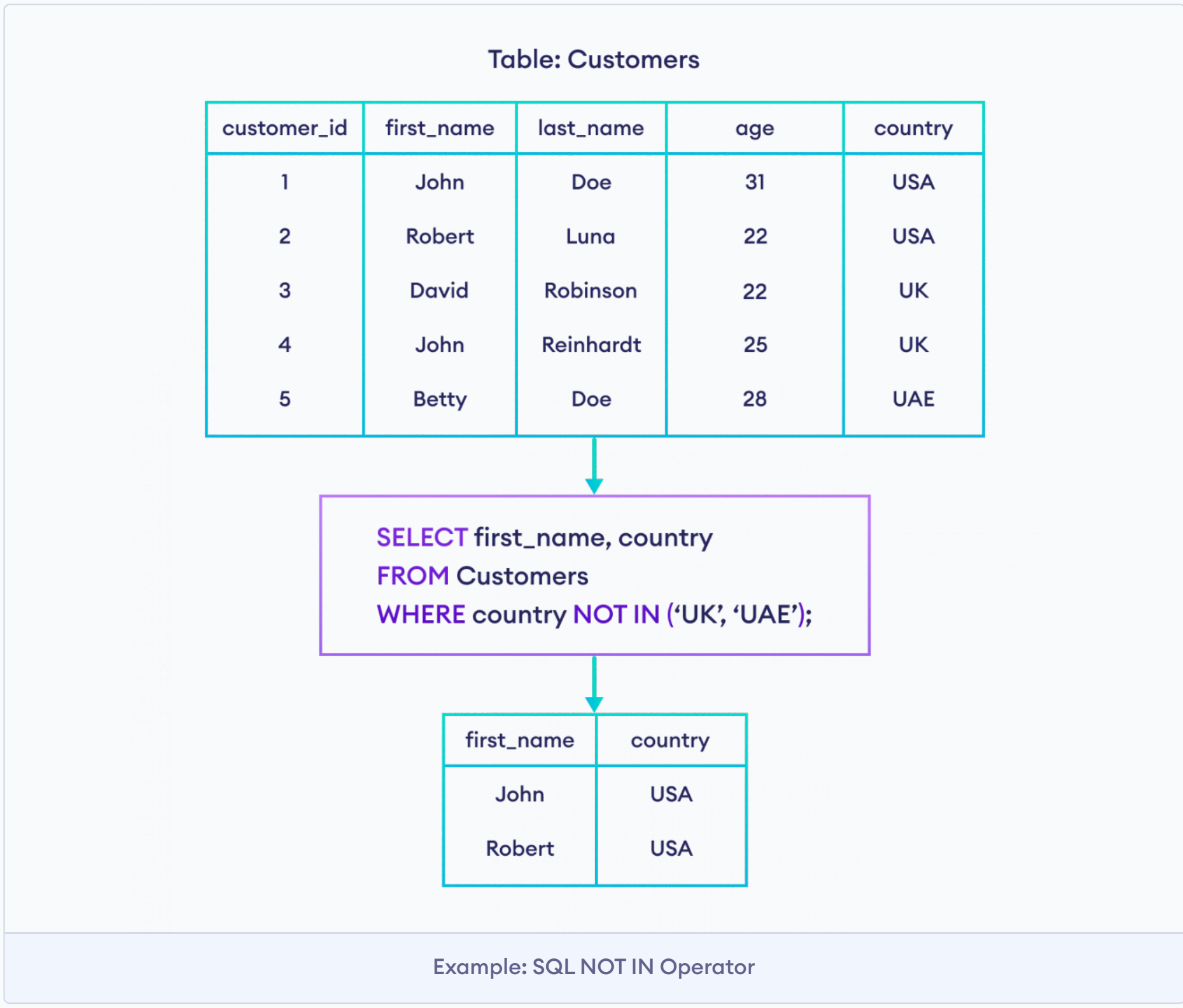
SQL NOT IN Operator

The `NOT IN` operator returns is used to exclude the rows that match values in the list. It returns all the rows except the excluded rows. For example,

```
SELECT first_name, country
FROM Customers
WHERE country NOT IN ('UK', 'UAE');
```

Run Code >>

Here, the SQL command selects rows if **UK** or **UAE** is not in the `country` column.



Note: The working of `IN` operator is reversed by `NOT` Operator. They are basically two operators combined. To learn more, visit [SQL AND, OR, and NOT Operators](#).

SQL IN Operator With Duplicate Values

By the way, the `IN` operator ignores duplicate values in the list. For example,

This code

```
SELECT first_name, country
FROM Customers
WHERE country IN ('USA', 'UK', 'USA');
```

Run Code >>

is equivalent to

```
SELECT first_name, country
FROM Customers
WHERE country IN ('USA', 'UK');
```

Run Code >>

SQL IN Operator With Subquery

Suppose we want details of customers who have placed an order. Here's how we can do that using a subquery,

```
SELECT customer_id, first_name
FROM Customers
WHERE customer_id IN (
  SELECT customer_id
  FROM Orders
);
```

Run Code >>

Here, the SQL command

- selects `customer_id` from `Orders` table
- select rows from `Customers` table where `customer_id` is in the result set of subquery

To learn more, visit [SQL Subquery](#).

Previous Tutorial:
SQL LIMIT, TOP, FETCH FIRST

Next Tutorial:
SQL BETWEEN →

Did you find this article helpful?

😊

😞

Related Tutorials

- Programming
SQL SELECT
- Programming
SQL Subquery
- Programming
SQL SELECT INTO Statement
- Programming
SQL INSERT INTO SELECT Statement

Programiz



Tutorials

- Python 3 Tutorial
- JavaScript Tutorial
- SQL Tutorial
- C Tutorial
- Java Tutorial
- Kotlin Tutorial
- C++ Tutorial
- Swift Tutorial
- C# Tutorial
- Go Tutorial
- DSA Tutorial

Examples

- Python Examples
- JavaScript Examples
- C Examples
- Java Examples
- Kotlin Examples
- C++ Examples

Company

- About
- Advertising
- Privacy Policy
- Terms & Conditions
- Contact
- Blog
- Youtube

Apps

- Learn Python
- Learn C Programming
- Learn Java