In this lesson, we dive deeper into SQL and how to filter data with **multiple conditions**. As a **security analyst**, it's common to encounter situations where multiple factors need to be considered simultaneously to identify vulnerabilities or patterns.

**Using the AND Operator**

The **AND** operator in SQL allows us to filter for data that satisfies **multiple conditions at the same time**. For example, if we want to find machines using a **specific email client** and running a **specific operating system**, both conditions must be met simultaneously. Think of it like selecting **large and fresh apples** from a fruit cart — the apples must meet both conditions to be included.

**Example:**

Let's say you want to find machines that meet two conditions:

* Running **Operating System 1**
* Using **Email Client 1**

Your query would look like this:

SELECT \* FROM machines WHERE operating\_system = 'OS 1' AND email\_client = 'Email Client 1';

This query will only return machines that meet both conditions.

**Using the OR Operator**

The **OR** operator is used when **either** of the conditions must be met. For example, if you need to find machines running **OS 1 or OS 3**, the query would return machines that satisfy **either** of the conditions or **both**.

**Example:**

SELECT \* FROM machines WHERE operating\_system = 'OS 1' OR operating\_system = 'OS 3';

This query will return machines running either **OS 1** or **OS 3**, or both.

**Using the NOT Operator**

The **NOT** operator negates a condition. It selects **all data except** for those that match the condition. Think of it like finding all the fruits **except** apples. This is particularly useful when you want to exclude certain data from your results.

**Example:**

If you want to exclude machines running **OS 3** and get all others, the query would be:

SELECT \* FROM machines WHERE NOT operating\_system = 'OS 3';

This will return all machines **except** those running **OS 3**.

**Conclusion**

Now you have learned how to filter data with **AND**, **OR**, and **NOT** operators, enabling you to craft more complex queries and work with multiple conditions. You can combine these operators in various ways to address specific scenarios, such as identifying vulnerabilities or filtering data based on multiple criteria.

In the next video, we’ll explore how to combine and join **two tables** together to expand the queries you can perform!

# More on filters with AND, OR, and NOT

Previously, you explored how to add filters containing the AND, OR, and NOT operators to your SQL queries. In this reading, you'll continue to explore how these operators can help you refine your queries.

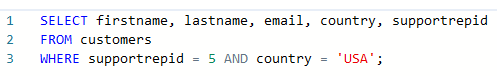
## Logical operators

AND, OR, and NOT allow you to filter your queries to return the specific information that will help you in your work as a security analyst. They are all considered logical operators.

### AND

First, AND is used to filter on two conditions. AND specifies that both conditions must be met simultaneously.

As an example, a cybersecurity concern might affect only those customer accounts that meet both the condition of being handled by a support representative with an ID of 5 and the condition of being located in the USA. To find the names and emails of those specific customers, you should place the two conditions on either side of the AND operator in the WHERE clause:

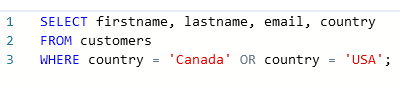


Running this query returns four rows of information about the customers. You can use this information to contact them about the security concern.

### OR

The OR operator also connects two conditions, but OR specifies that either condition can be met. It returns results where the first condition, the second condition, or both are met.

For example, if you are responsible for finding all customers who are either in the USA or Canada so that you can communicate information about a security update, you can use an OR operator to find all the needed records. As the following query demonstrates, you should place the two conditions on either side of the OR operator in the WHERE clause:



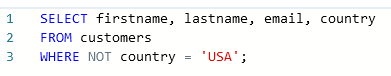
The query returns all customers in either the US or Canada.

**Note:** Even if both conditions are based on the same column, you need to write out both full conditions. For instance, the query in the previous example contains the filter WHERE country = 'Canada' OR country = 'USA'.

### NOT

Unlike the previous two operators, the NOT operator only works on a single condition, and not on multiple ones. The NOT operator negates a condition. This means that SQL returns all records that don’t match the condition specified in the query.

For example, if a cybersecurity issue doesn't affect customers in the USA but might affect those in other countries, you can return all customers who are not in the USA. This would be more efficient than creating individual conditions for all of the other countries. To use the NOT operator for this task, write the following query and place NOT directly after WHERE:



SQL returns every entry where the customers are not from the USA.

**Pro tip:** Another way of finding values that are not equal to a certain value is by using the <> operator or the != operator. For example, WHERE country <> 'USA' and WHERE country != 'USA' are the same filters as WHERE NOT country = 'USA'.

## Combining logical operators

Logical operators can be combined in filters. For example, if you know that both the USA and Canada are not affected by a cybersecurity issue, you can combine operators to return customers in all countries besides these two. In the following query, NOT is placed before the first condition, it's joined to a second condition with AND, and then NOT is also placed before that second condition. You can run it to explore what it returns:

## 

## Key takeaways

Logical operators allow you to create more specific filters that target the security-related information you need. The AND operator requires two conditions to be true simultaneously, the OR operator requires either one or both conditions to be true, and the NOT operator negates a condition. Logical operators can be combined together to create even more specific queries.