# Apply filters to SQL queries

## Project description

I have discovered some potential security issues that involve login attempts and employee machines. My task is to examine the organization’s data in their **employees** and **log\_in\_attempts** tables and retrieve records using SQL filters to investigate potential security issues.

## Retrieve after hours failed login attempts

metin, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.I discovered that there have been an incident that occurred after business hours (18:00). To investigate the failed attempts after hours I searched the database for failed attempts done after 18:00. The image below demonstrates my actions and its results.

The screenshot includes my query, and the output result of it. This query filters the failed login attempts made after 18:00. First, I selected all the data in the **log\_in\_attempts** table. After that, I filtered the **login\_time** column to display only attempts made after 18:00 with **“login\_time > 18:00”** and to display only failed attempts which were described as 0 in this table with **“and success = 0”**.

## Retrieve login attempts on specific dates

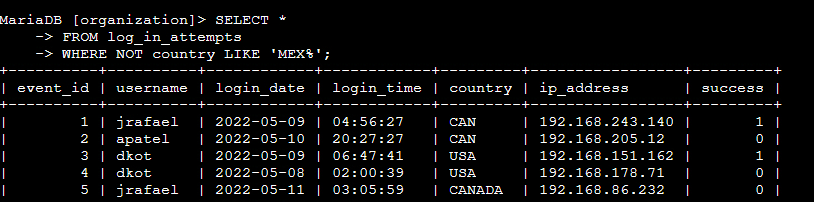
I was informed that there were suspicious activities done on 2022-05-09. So the day that the activity was done and before (2022-05-08) must be investigated.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.Image below shows my query to get the related information and a portion of the output.

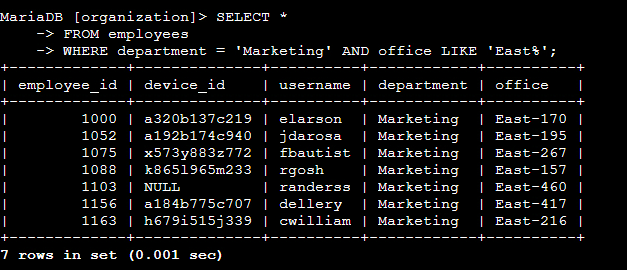
In my query, first I am selecting all the values from the **log\_in\_attempts** table. Then, I am creating a filtering method using **WHERE** clause. I describe a filter that takes **login\_date** as 2022-05-09. At the end of this statement I add **OR** operator to include a second date 2022-05-08. With this method I can display a filtered table with both dates.

## Retrieve login attempts outside of Mexico

After analyzing the data, the team decided that this activity did not originate in Mexico. So, I decided to analyze the attempts outside of Mexico. I also realized that in some record “MEX” was used instead of “MEXICO”, so I had to write my query accordingly. The image below shows my query to get all the attempts outside of Mexico and some of the retrieved data.

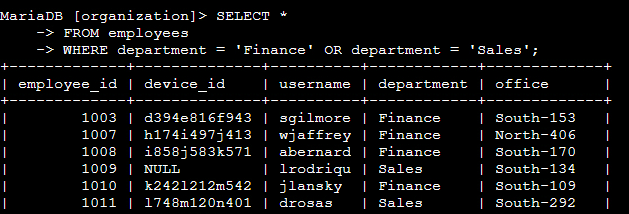
First, I selected all the records from the **log\_in\_attempts** table. After that, I used **WHERE** clause to filter. Then, I started the filter description with **NOT** operator because I want to get attempts that were not from Mexico. As I mentioned before, some records had “MEX” instead of “MEXICO”, so to include both selections I used **‘MEX%’**. The **%** wildcard indicates that there can be other letters or words after the **MEX**, which means that **MEXICO** will also be included. Since I used a wildcard, I replaced the **(=)** with the expression **LIKE.**

## Retrieve employees in Marketing

My team was responsible for performing security updates on specific employee machines in the marketing department. I was tasked to check the employee machines on the East side offices. The image below shows the query that retrieves the information on devices of people working in marketing department in East offices.

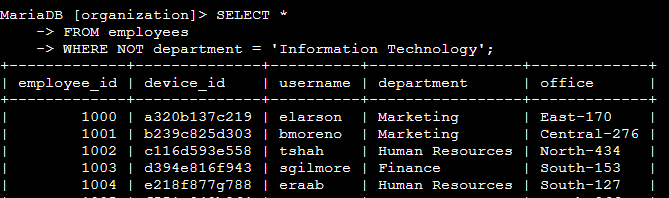
The query I wrote first selects all the information from the **employees** table. Then, I am filtering it using **WHERE** clause specifying the **department** as **‘Marketing’**, and I need both conditions to meet so I am using **AND** operator. After that, I am specifying office as **‘East%’** since the values are varying as **“East-170”**, and using **LIKE** since I used a wildcard. From the results it can be seen that **device\_id** of the employee with the id **1103** is **NULL**.

## Retrieve employees in Finance or Sales

The team wanted to perform another security update on devices of people in finance and sales departments. The image below shows the query I wrote to get the records on people working in finance and sales departments, and some of the records retrieved from the query.

First, I select all the information from the **employees** table. After that, I am describing my filter starting with **WHERE** clause, and specifying that if the **department** is **‘Finance’** or **‘Sales’**, I want the records on the employee.

## Retrieve all employees not in IT

The security team wanted to make one more update to the machines, but the IT team already has these updates. So, I have to get records on every employee working outside of IT department. The image below demonstrates how I retrieved the records on people that are not in IT department.

First, I am selecting all the information from the **employees** table. Then, I am specifying the filter I started with **WHERE** to retrieve records only where the **department** is not **‘Information Technology’** using the expression with **‘NOT’** operator.

## Summary

In this project, first I applied SQL queries to retrieve information on login attempts on employee machines to investigate on suspicious activities. The table I used for this information was **log\_in\_attempts.** Then, I applied SQL queries to retrieve information on employee machines of specific departments to check if they all received the necessary updates. I retrieved information applying filters with **WHERE**, different operators like **AND, OR, NOT** and wildcard percentage sign (%) with **LIKE**.