# Apply filters to SQL queries

## Project description

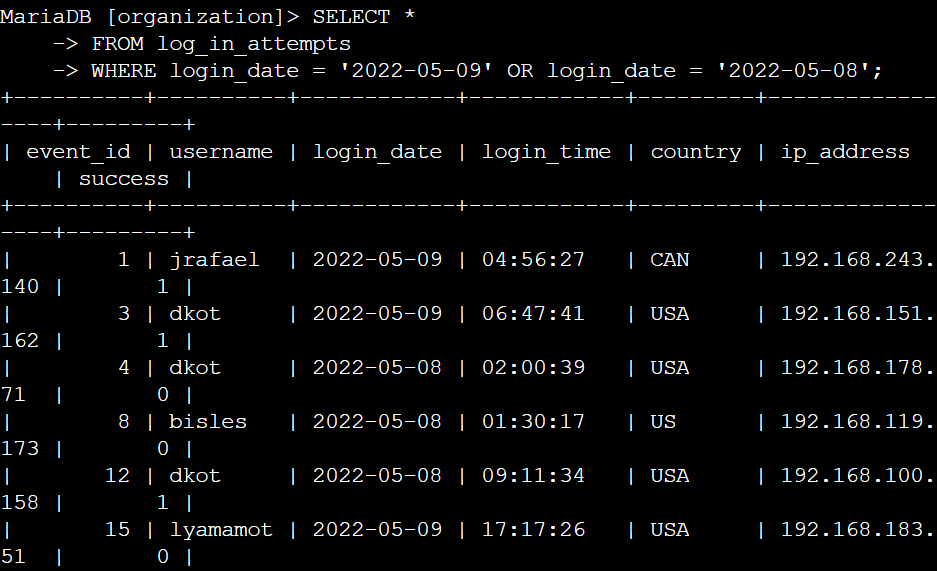
This organization is working to make their system more secure. Using SQL, it’s my job to make sure the system is safe by investigating potential security issues, and update employee computers if required.

## Retrieve after hours failed login attempts



Because there was a potential security incident after business hours (after 18:00), All after-hour login attempts that failed needed to be investigated. The first part of the screenshot is my query, which filters for failed login attempts that occurred after 18:00. First, I started by selecting all data from the log\_in\_attempts table by using SELECT \* . Then, I used a WHERE clause with an AND operator to filter my results to output only login attempts that occurred after 18:00 and were unsuccessful. The first condition is login\_time > '18:00', which filters for the login attempts that occurred after 18:00. The second condition is success = FALSE, which filters for the failed login attempts.

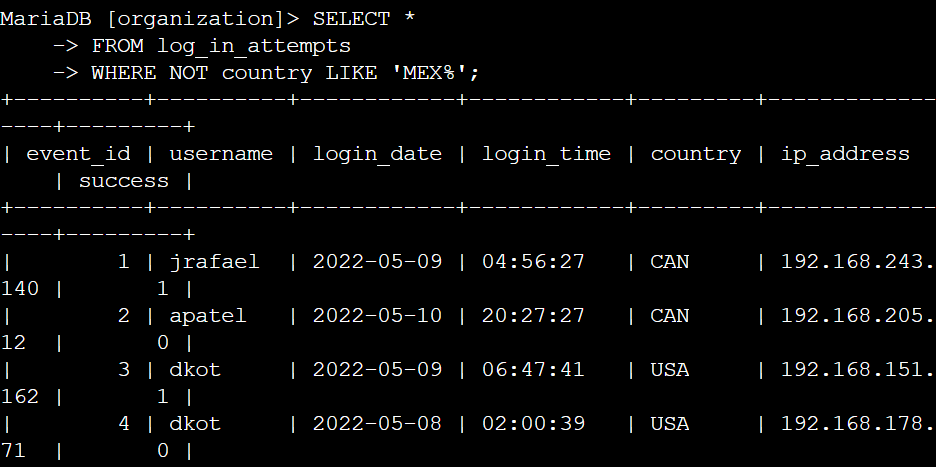
## Retrieve login attempts on specific dates



A suspicious event was recorded that occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or on the day before was decided to be investigated.

The first part of the screenshot is my query, which returns all login attempts that occurred on 2022-05-09 or 2022-05-08. First, I started by selecting all data from the log\_in\_attempts table using SELECT \*. Then, I used a WHERE clause with an OR operator to filter my results to output only login attempts that occurred on either 2022-05-09 or 2022-05-08. The first condition is login\_date = '2022-05-09', which filters for logins on 2022-05-09. The second condition is login\_date = '2022-05-08', which filters for logins on 2022-05-08. By using both we were able to view the login attempt results from both dates

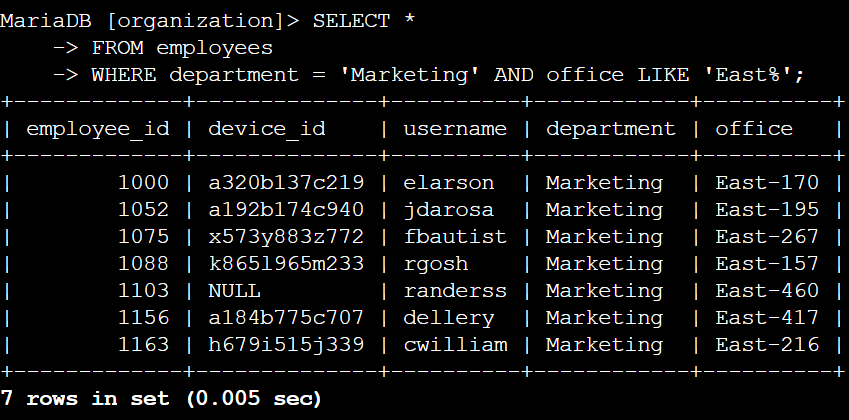
## Retrieve login attempts outside of Mexico



After investigating the organization’s data on login attempts, we wanted to investigate a possible issue with the login attempts that occurred outside of Mexico.

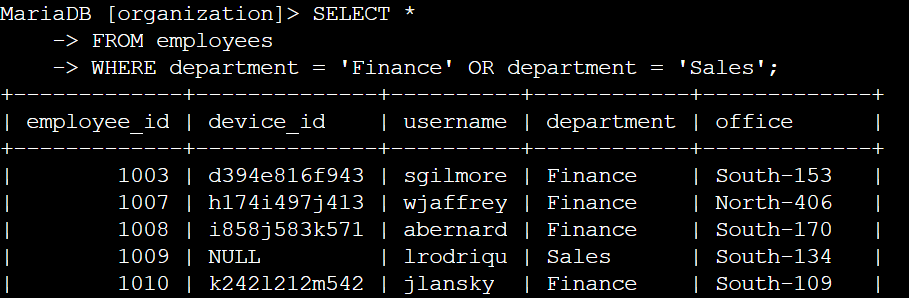
The first part of the screenshot is my query, which returns all login attempts that occurred in countries other than Mexico. First, I started by selecting all data from the log\_in\_attempts table using SELECT \* . Then, I used a WHERE clause with NOT to filter for countries other than Mexico. I used LIKE with MEX% as the pattern to match because the dataset represents Mexico as MEX and MEXICO. The % represents any number of unspecified characters when used with LIKE.

## Retrieve employees in Marketing



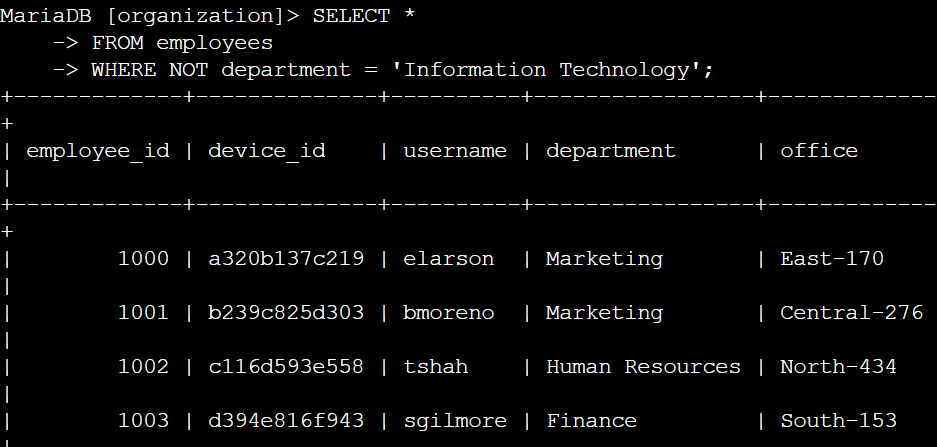
My team wants to update the computers for certain employees in the Marketing department. To do this, I have to get information on which employee machines to update for the East building. The first part of the screenshot is my query, which returns all employees in the Marketing department in the East building. First, I started by selecting all data from the employees table by using SELECT \*. Then, I used a WHERE clause with AND to filter for employees who work in the Marketing department and in the East building. I used LIKE with East% as the pattern to match because the data in the office column represents the East building with the specific office number. The first condition is the department = 'Marketing' portion, which filters for employees in the Marketing department. The second condition is the office LIKE 'East%' portion, which filters for employees in the East building.

## Retrieve employees in Finance or Sales



The machines for employees in both the Finance and Sales departments also needed to be updated. Since a different security update is needed, I had to get information on employees only from these two departments. The first part of the screenshot is my query, which returned all employees in the Finance and Sales departments. First, I started by selecting all data from the employees table using SELECT \*. Then, I used a WHERE clause with OR to filter for employees who are in the Finance and Sales departments. I used the OR operator instead of AND because I want all employees who are in either department. The first condition is department = 'Finance', which filters for employees from the Finance department. The second condition is department = 'Sales', which filters for employees from the Sales department.

## Retrieve all employees not in IT



My team needed a security update on all employees who are not in the Information Technology department. To make the update, I first had to get information on these employees. The first part of the screenshot is my query, which returns all employees not in the Information Technology department. First, I started by selecting all data from the employees table using SELECT \*. Then, I used a WHERE clause with NOT to filter for employees not in this department.

## Summary

By applying filters to SQL queries, I was able to get specific information on login attempts and employee machines. From specified tables, such as log\_in\_attempts and employees. I then used the AND, OR, and NOT operators to filter for the specific information needed for each task. I also used LIKE and the % wildcard to filter for patterns that I wanted to find in my results.