# Apply filters to SQL queries

## Project description

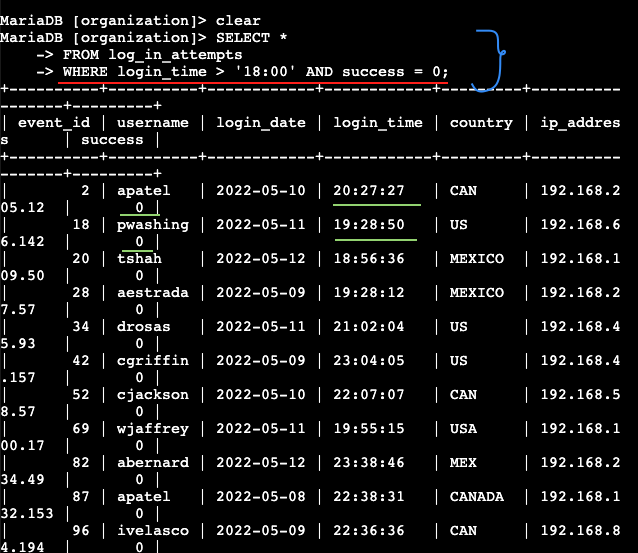
As a security professional at a large organization part of your job is to investigate security issues to help keep the system secure. We recently discovered some potential security issues that involve login attempts and employee machines.

Our task is to examine the organization’s data in their employees and log\_in\_attempts tables. We’ll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues

## Retrieve after hours failed login attempts

The cybersecurity team recently discovered a potential security incident that occurred after business hours. To investigate this, the cybersecurity analyst needs to query the log\_in\_attempts table and review after hours login activity.

The cybersecurity analyst uses filters in SQL to create a query that identifies all failed login attempts that occurred after 18:00. (The time of the login attempt is found in the login\_time column. The success column contains a value of 0 when a login attempt failed; a value of 0 or FALSE in your query to identify failed login attempts.) can be used.



As seen above the cybersecurity analyst entered the command

SELECT \*

FROM log\_in\_attempts

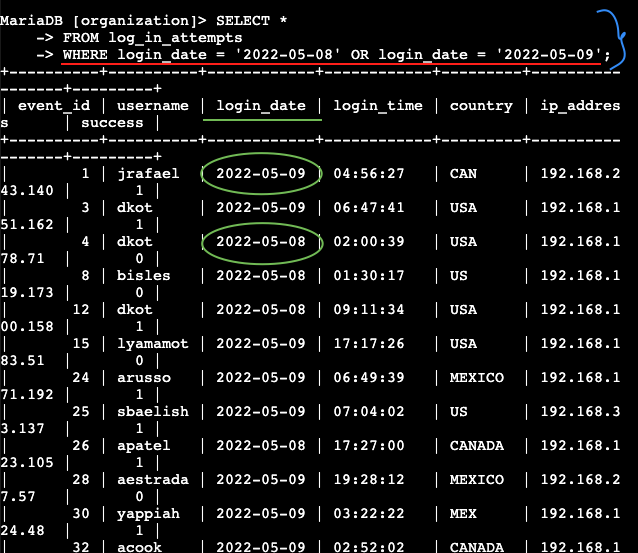
WHERE login\_time > '18;00' AND success = 0;

Including the AND operator to retrieve the failed login attempts that occurred after business hours.

## Retrieve login attempts on specific dates

The team is investigating a suspicious event that occurred on '2022-05-09' and as such want to retrieve all login attempts that occurred on that day and the day before ('2022-05-08').

The login\_date column in the log\_in\_attempts table contains information on the dates when login attempts were made.



The cybersecurity analyst used the OR operator to retrieve the failed login attempts on the specified days with the entry .

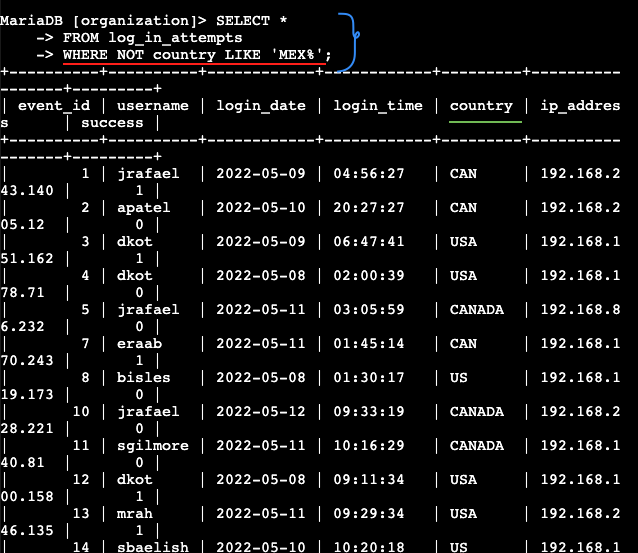
SELECT \*

FROM log\_in\_attempts

WHERE login\_date = 'X' OR login\_date = 'Y';

## Retrieve login attempts outside of Mexico

Now, our team is investigating logins that did not originate in Mexico, and the security analyst needs to find this information. Note that the country field includes entries with 'MEX' and 'MEXICO'. We should use the NOT and LIKE operators and the matching pattern 'MEX%'.



We ran the following SQL query

SELECT \*

FROM log\_in\_attempts

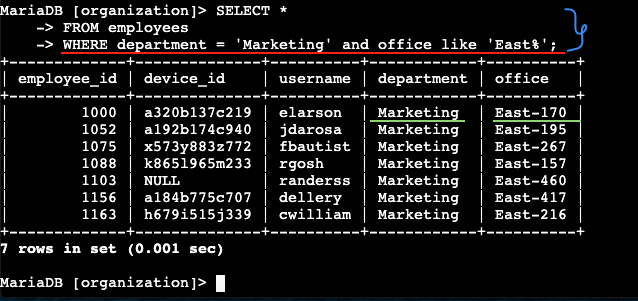
WHERE NOT country LIKE 'MEX%';

To retrieve login attempts that did not originate in Mexico.

## Retrieve employees in Marketing

The cybersecurity team is updating employee machines, and the SOC analyst needs to obtain the information about employees in the 'Marketing' department who are located in all offices in the East building (such as 'East-170' or 'East-320').

We write a SQL query to retrieve this information from the employees table.



I selected all columns and included filters on the department and office columns to return only the needed records. This through the query

SELECT \*

FROM employees

WHERE department = 'Marketing' AND office like 'East%’;

## Retrieve employees in Finance or Sales

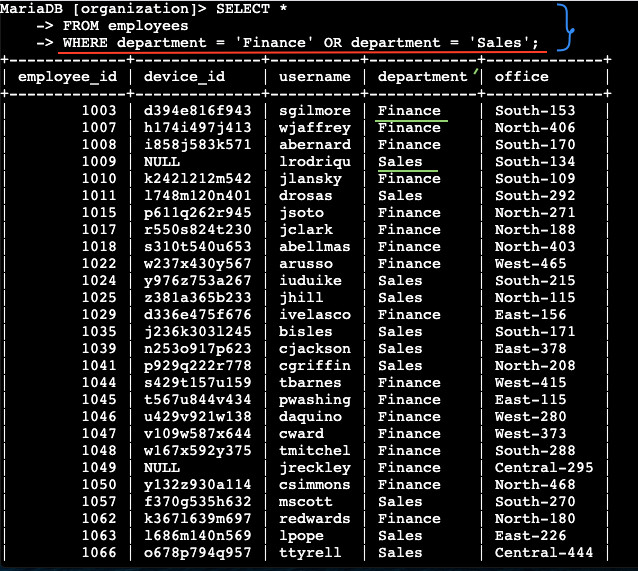
Now, the cybersecurity team needs to perform a different update to the computers of all employees in the Finance or the Sales department, and the cybersecurity analyst needs to locate information on these employees.

I Write a SQL query to retrieve records for employees in the 'Finance' or the 'Sales' department.

SELECT \*

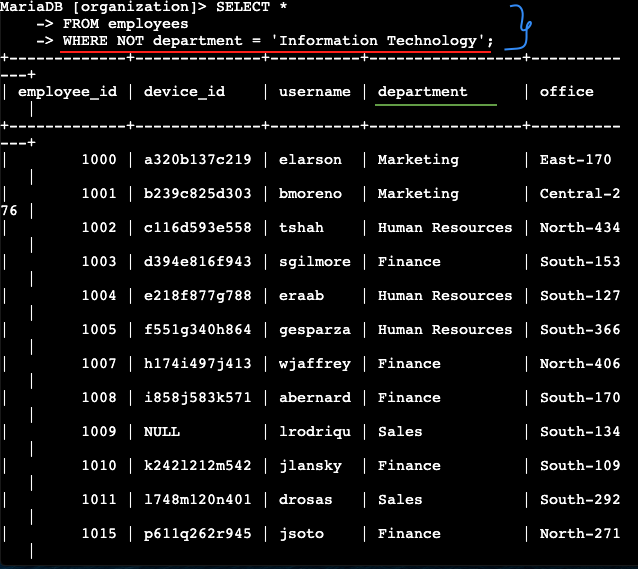
FROM employees

WHERE department = 'Finance' OR department = 'Sales';



## Retrieve all employees not in IT

The team needs to make one more update. This update was already made to employee computers in the Information Technology department. The team needs information about employees who are not in that department and I used the NOT operator to identify these employees.



We wrote the following SQL query

SELECT \*

FROM employees

WHERE NOT department = ‘Information Technology’;

To retrieve records for employees who are not in the 'Information Technology' department.

## Summary

In this cybersecurity project, SQL queries and filters were used to investigate potential security issues related to login attempts and employee records. By examining after-hours login activity, specific date-based events, international login origins, and department-specific employee details, the project showcases essential data analysis skills for maintaining system security.