# Apply filters to SQL queries

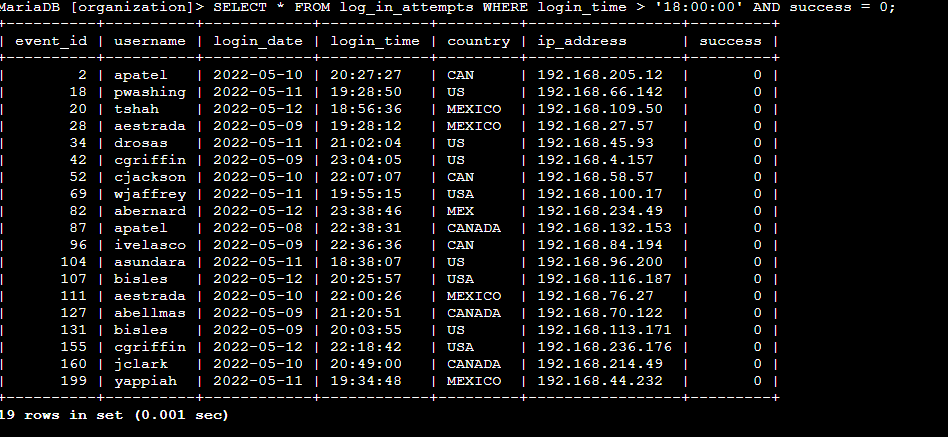
## Project description

The organization is working to make their system more secure. It is my job to ensure the system is safe, investigate all potential security issues, and update employee computers as needed. The following steps provide examples of how I used SQL with filters to perform security-related tasks.

## Retrieve after hours failed login attempts

There was a potential security incident that occurred after business hours (after 18:00). All after hours login attempts that failed need to be investigated. This query filters for failed login attempts that occurred after 18:00. The query consists on selecting (SELECT) from the table log\_in\_attempts which contains the information needed, the condition (WHERE) is that the login time occurs after business hours (login\_time > ‘18:00:00’) and it was successful (success = 0)

(SELECT \* FROM log\_in\_attempts WHERE login\_time > ‘18:00:00’ AND success = 0;)



## 

## Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or on the day before needs to be investigated.The following code demonstrates how I created a SQL query to filter for login attempts that occurred on specific dates. From the table log\_in\_attempts, where the log in attempts of the company are recorded, the query consists on on selecting from the table (SELECT \* FROM log\_in\_attempts) all the entries which meet the condition of the login date being either on September 8 of 2023 (WHERE login\_date = ‘2022-03-09’) or September 9 (OR login\_date = ‘2022-03-08’)

(SELECT \* FROM log\_in\_attempts WHERE login\_date = ‘2022-05-09 OR login\_date = ‘2022-05-08’;)

The query returned 75 rows.

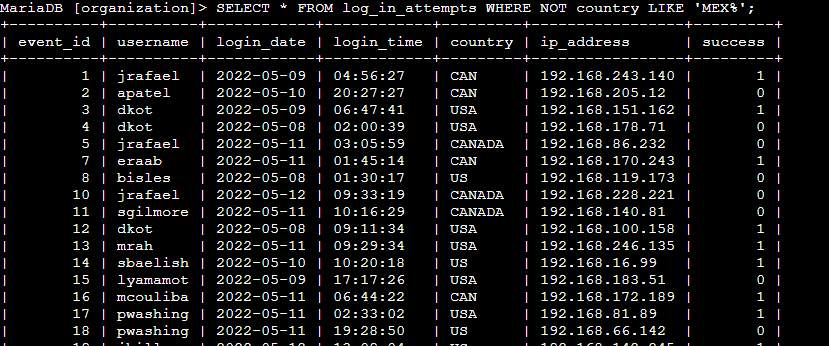


## Retrieve login attempts outside of Mexico

After investigating the organization’s data on login attempts, I believe there is an issue with the login attempts that occurred outside of Mexico. These login attempts should be investigated. The query consists of selecting from the log\_in\_attempts table (SELET \* FROM log\_in\_attempts) all the entries that are not in the country of Mexico (WHERE NOT country LIKE ‘MEX%’).

(SELECT \* FROM log\_in\_attempts WHERE NOT country LIKE ‘MEX%’;)

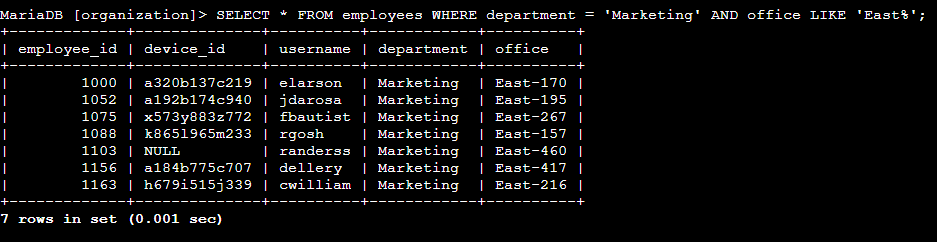
The query returned 144 rows.



## 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. The query consists on taking the information from the employees table in the database (SELECT \* FROM employees) and returning any employee that is in the Marketing department (WHERE department = ‘Marketing’) and is located in the East building (AND office LIKE ‘East%’). The ‘East%’ is to collect anyone in the east building regardless of the number.

(SELECT \* FROM employees WHERE department = ‘Marketing’ AND office LIKE ‘East%’;)

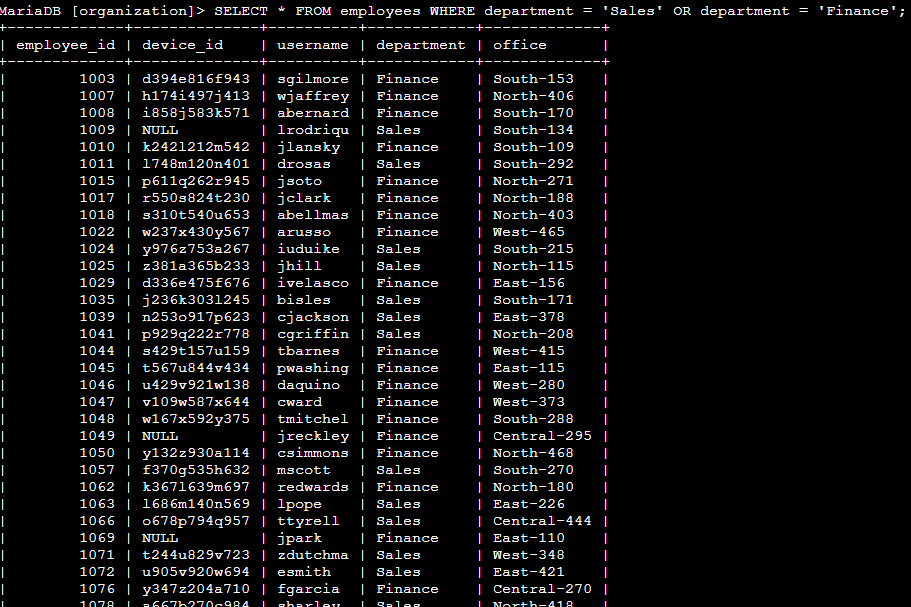


## Retrieve employees in Finance or Sales

The machines for employees in the Finance and Sales departments also need to be updated. Since a different security update is needed, I have to get information on employees only from these two departments. The query consists of taking the information from the employees table in the database (SELECT \* FROM employees) and return the employees which form part of the Sales department (WHERE department = ‘Sales’) or the Finance department (OR department = ‘Finance’).

(SELECT \* FROM employees WHERE department = ‘Sales’ OR department = ‘Finance’;)

The query returned 71 rows.

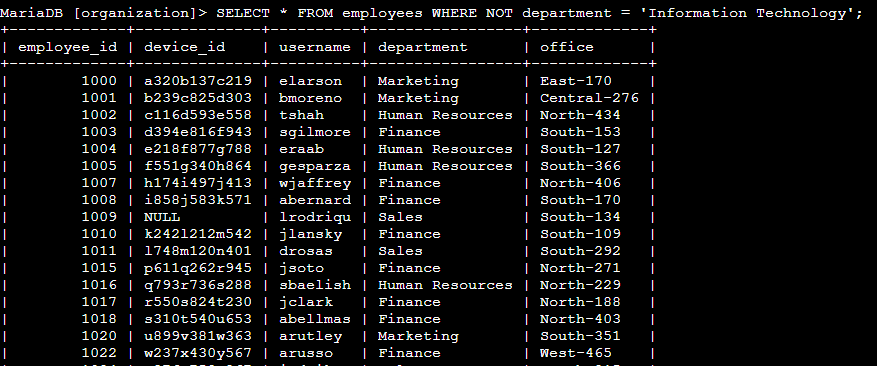


## Retrieve all employees not in IT

The team needs to make one more security update on employees who are not in the IT (Information Technology) department. To make the update, the information of these employees must be retrieved. The query consists of taking the information from the employees table in the database (SELECT \* FROM employees) and return any employee that is not on the IT department (WHERE NOT department = ‘Information Technology’)

(SELECT \* FROM employees WHERE NOT department = ‘Information Technology’;)

The query returned 161 rows as an output.



## Summary

After applying the filters to SQL queries to get specific information on login attempts and employee machines, I used two different tables, log\_in\_attempts and employees, to demonstrate the efficiency of the usage of filters in the SQL language. I used the AND, OR, and NOT operators to filter for the specific information needed for each task. I also used LIKE and the percentage sign (%) wildcard to filter for patterns.