Apply filters to SQL queries

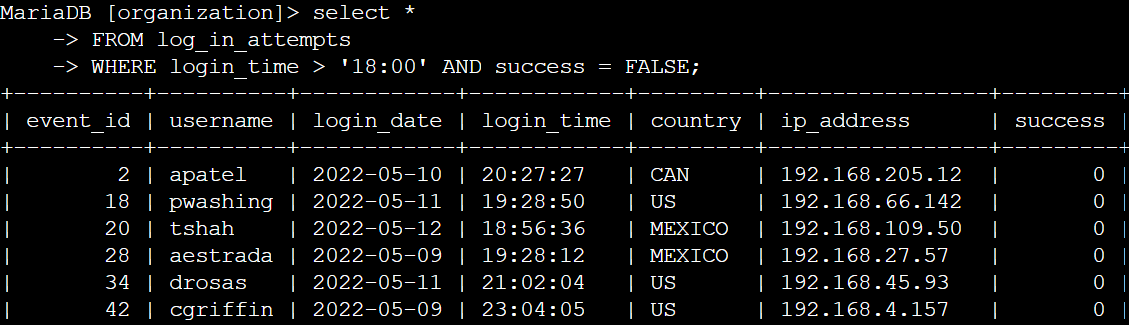
# Apply filters to SQL queries

## Project description

My role is to investigate login attempts and employee machines to identify potential security threats. Using SQL queries, I analyze data from the employees and log\_in\_attempts tables to detect anomalies and enhance system security.

## Retrieve after hours failed login attempts

Security incident that occurred after business hours (18:00). After hours login attempts that failed need to be investigated.



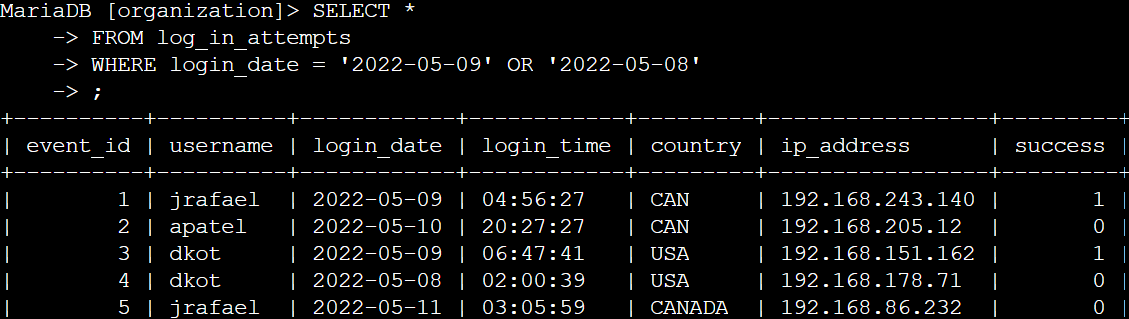
The first part of the screenshot shows my query, while the second part displays a portion of the output. This query filters for failed login attempts that occurred after 18:00. I began by selecting all data from the log\_in\_attempts table. Then, I applied a WHERE clause with an AND operator to narrow the results to only unsuccessful login attempts after 18:00.

The first condition, login\_time > '18:00', filters login attempts that occurred after 18:00. The second condition, success = FALSE, ensures that only failed login attempts are included in the output.

## 

## 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 first part of the screenshot shows my query, while the second part displays a portion of the output. This query retrieves all login attempts that took place on 2022-05-09 or 2022-05-08. I began by selecting all data from the log\_in\_attempts table. Then, I applied a WHERE clause with an OR operator to filter the results, ensuring only login attempts from these two dates were included.The first condition, login\_date = '2022-05-09', filters for logins on May 9, 2022. The second condition, login\_date = '2022-05-08', filters for logins on May 8, 2022.

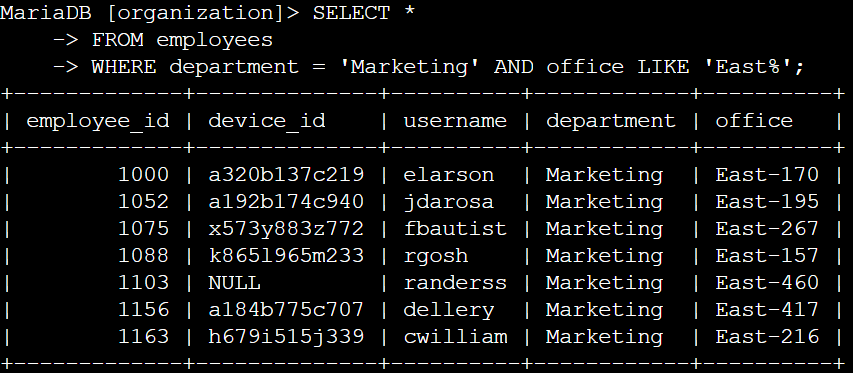
## Retrieve login attempts outside of Mexico

After investigating the organization’s data on login attempts, I believe there is an issue with the login attempts that occurred outside of Mexico.



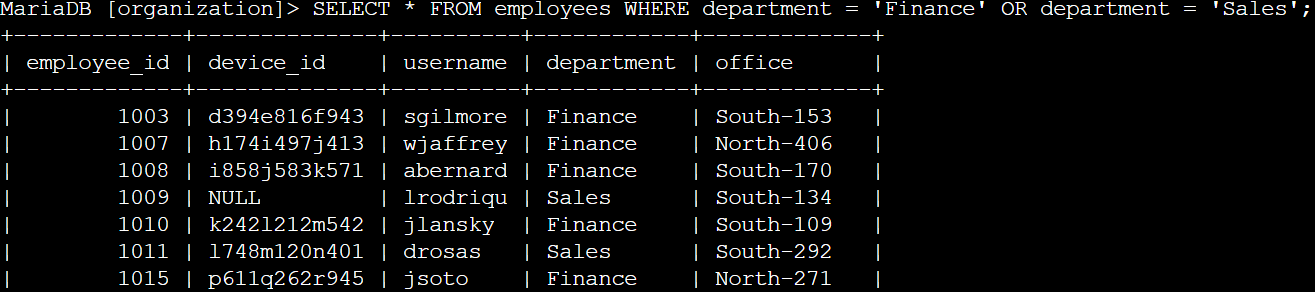
The first part of the screenshot shows my query, and the second part displays the output. This query retrieves login attempts from countries other than Mexico. I used NOT LIKE 'MEX%' to exclude entries labeled as MEX or MEXICO, with % acting as a wildcard for any following characters.

## Retrieve employees in Marketing



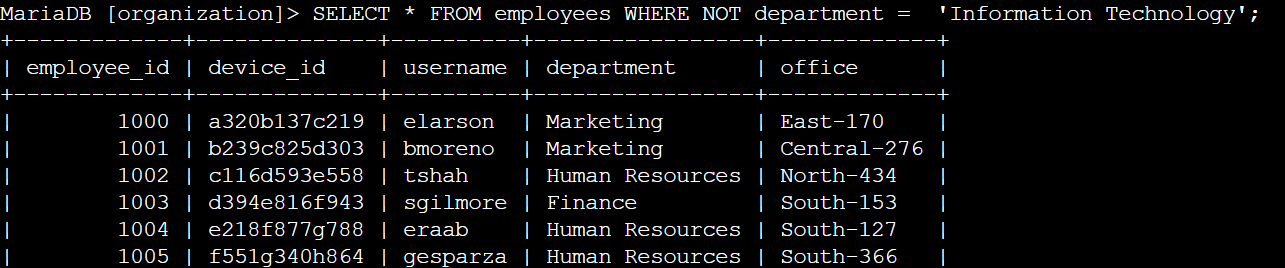
The first part of the screenshot shows my query, and the second part displays the output. This query retrieves all Marketing department employees located in the East building. I selected data from the employees table and used a WHERE clause with AND to filter results.The first condition, department = 'Marketing', selects employees in Marketing. The second condition, office LIKE 'East%', filters for those in the East building, where LIKE 'East%' accounts for specific office numbers.

Retrieve employees in Finance or Sales



## The first part of the screenshot shows my query, and the second part displays the output. This query retrieves all employees from the Finance and Sales departments. I selected data from the employees table and used a WHERE clause with OR to include employees from either department.The first condition, department = 'Finance', filters for Finance employees, while the second condition, department = 'Sales', filters for Sales employees. Using OR ensures that employees from both departments are included.

## Retrieve all employees not in IT



## The first part of the screenshot shows my query, and the second part displays the output. This query retrieves all employees excluding those in the Information Technology department. I selected data from the employees table and used a WHERE clause with NOT to filter out employees from this department.

## Summary

I applied filters to SQL queries to retrieve specific information on login attempts and employee machines using the log\_in\_attempts and employees tables. I used AND, OR, and NOT operators to refine results based on different criteria. Additionally, I applied LIKE with the % wildcard to filter data based on patterns.