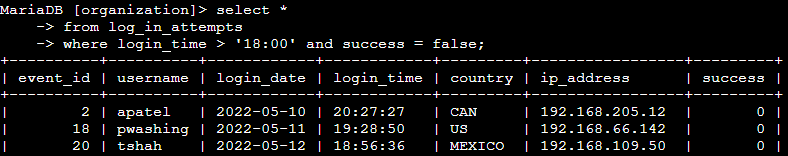
# Apply filters to SQL queries

## Project description

As a cybersecurity analyst at a large corporation, one of my primary responsibilities is to investigate and act on potential security issues to help keep the system secure. Recently there are potential risks stemming from login attempts and employee machines. The following steps provide examples of how I used SQL with filters to execute security related tasks.

## Retrieve after hours failed login attempts

There was a security incident that occurred after business hours (after 18:00). So I moved to investigate all failed login attempts that occurred after business hours. The following screenshot shows how I used SQL filters to see all failed login attempts that occurred after business hours.

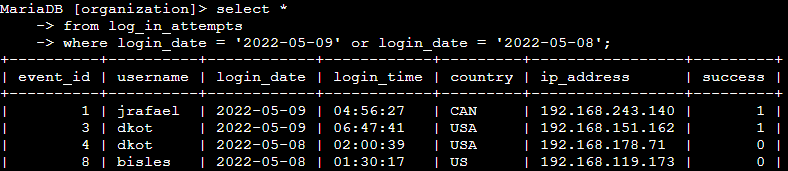


In the presented screenshot, I initiated my query by selecting all information from the "log\_in\_attempts" table. Subsequently, I applied a filtering condition using a WHERE clause along with the AND operator to narrow down the results to show only unsuccessful login attempts that took place after 18:00. The first criterion utilized was login\_time > '18:00', effectively capturing login attempts after the specified time. To further refine the output, the second criterion was employed, success = FALSE, targeting specifically the failed login attempts.

## Retrieve login attempts on specific dates

An incident of interest took place on the date 2022-05-09. It requires thorough investigation, including any login activities that transpired on that particular date or the preceding day.

To address this, I crafted an SQL query to filter login attempts that took place on the specified dates.



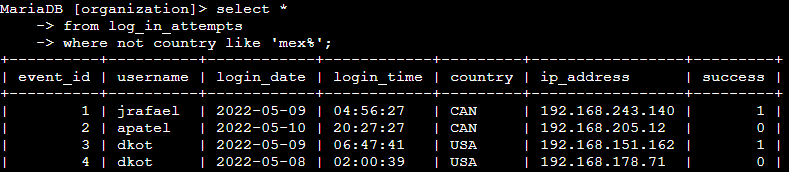
The initial section of the image showcases my query, while the subsequent part displays a portion of the output.

This specific query retrieves all login attempts that took place on either 2022-05-09 or 2022-05-08. My approach commenced by selecting all available data from the "log\_in\_attempts" table. Following that, I applied a WHERE clause using an OR operator to narrow down the results, displaying only login attempts that occurred on either of the specified dates, i.e., 2022-05-09 or 2022-05-08. The first condition, login\_date = '2022-05-09', facilitates filtering for logins that occurred on 2022-05-09, whereas the second condition, login\_date = '2022-05-08', serves to filter for logins that occurred on 2022-05-08.

## Retrieve login attempts outside of Mexico

Upon examining the organization's login attempt records, it has come to my attention that there might be a problem concerning login attempts originating from locations outside of Mexico. I consider it crucial to thoroughly investigate these login attempts.

To address this concern, I have developed an SQL query to isolate and filter login attempts that took place beyond the borders of Mexico.



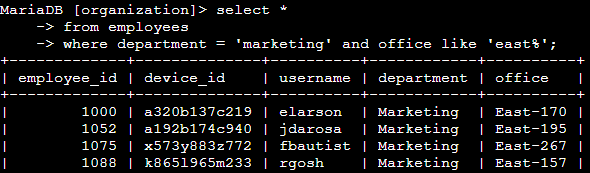
In the presented image, the initial segment comprises my query, while the subsequent part exhibits a section of the output.

This specific query is designed to retrieve all login attempts that took place in countries distinct from Mexico. To accomplish this, I began by selecting all available data from the "log\_in\_attempts" table. Subsequently, I employed a WHERE clause with the NOT operator to exclude occurrences from Mexico. To achieve this, I utilized the LIKE operator along with the pattern "MEX%" as the matching criterion. This pattern is employed since the dataset represents Mexico as "MEX" and "MEXICO." The percentage sign (%) in the pattern serves to encompass any number of unspecified characters when used with the LIKE operator.

## Retrieve employees in Marketing

In response to my team's objective of updating the computers assigned to specific individuals within the Marketing department, I embarked on obtaining relevant information about the employee machines that require updates.

To accomplish this task, I crafted an SQL query with the aim of filtering out employee machines linked to individuals in the Marketing department, specifically those located in the East building.



The initial part of the image displays my query, while the subsequent section exhibits a portion of the output.

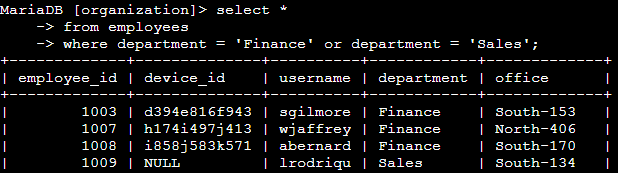
This specific query is designed to fetch all employees who belong to the Marketing department and work in the East building. To achieve this, I initiated by selecting all available data from the employees table. Subsequently, I utilized a WHERE clause with the AND operator to narrow down the results, focusing on employees who meet the criteria of being part of the Marketing department and stationed in the East building.

To identify employees in the East building, I employed the LIKE keyword with the pattern "East%" since the office column represents the East building with specific office numbers. The first condition, department = 'Marketing', serves to filter for employees in the Marketing department, while the second condition, office LIKE 'East%', targets employees in the East building.

## Retrieve employees in Finance or Sales

To address the update requirements for computers assigned to employees in the Finance and Sales departments, it is essential to gather relevant information solely pertaining to individuals within these two departments.

To achieve this objective, I have formulated an SQL query designed to filter employee machines belonging to individuals in either the Finance or Sales departments.



The initial section of the image showcases my query, while the subsequent part displays a section of the output.

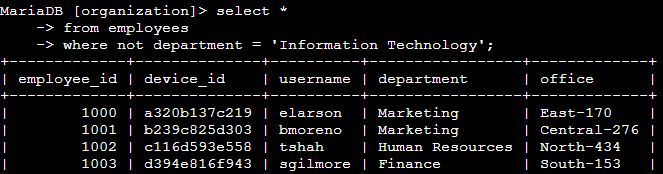
This specific query is designed to retrieve all employees belonging to both the Finance and Sales departments. To accomplish this, I initiated by selecting all available data from the employees table. Subsequently, I utilized a WHERE clause with the OR operator to narrow down the results, specifically targeting employees who are part of either the Finance or Sales departments.

Using the OR operator in place of AND allows for including all employees who belong to either department. The first condition, department = 'Finance', effectively filters for employees from the Finance department, while the second condition, department = 'Sales', serves to filter for employees from the Sales department.

## Retrieve all employees not in IT

To fulfill our team's requirement of implementing an additional security update for employees who do not belong to the Information Technology department, it is essential to gather relevant information exclusively related to these employees.

To achieve this objective, I have devised an SQL query designed to filter employee machines associated with individuals outside the Information Technology department.



The initial segment of the image displays my query, while the subsequent section exhibits a portion of the output.

This specific query is designed to retrieve all employees who are not part of the Information Technology department. To accomplish this, I began by selecting all available data from the employees table. Subsequently, I employed a WHERE clause with the NOT operator to narrow down the results, focusing on employees who do not belong to the mentioned department.

## Summary

I utilized filtering techniques in SQL queries to obtain particular details about login attempts and employee machines. This involved working with two distinct tables, namely log\_in\_attempts and employees. To achieve the desired results for each task, I employed the AND, OR, and NOT operators, tailoring the filters accordingly. Additionally, I utilized the LIKE operator along with the percentage sign (%) wildcard to efficiently filter data based on specific patterns.