# Apply filters to SQL queries

## Project description

In this scenario, I am tasked with examining the organization’s data in the employee and log\_in\_attempts tables. This can be done using SQL filters to retrieve records from the datasets.

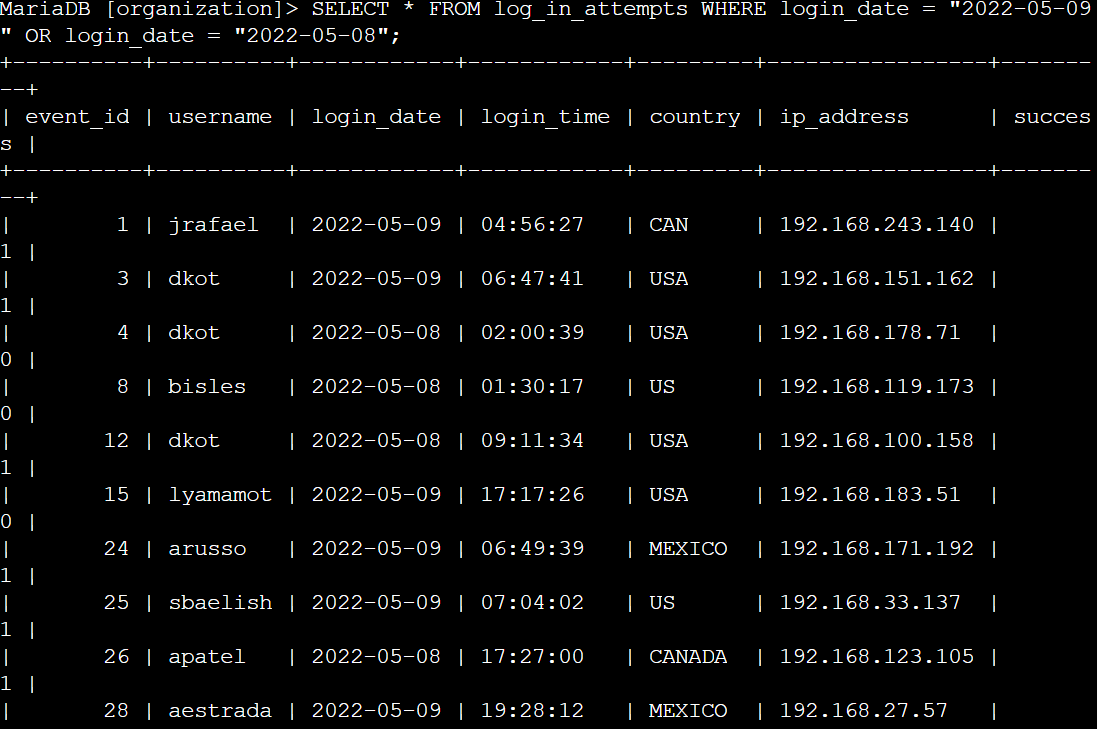
## Retrieve after hours failed login attempts

A potential security incident was seen to have occurred after business hours (18:00). Login attempts after this that failed should be investigated. Using a SQL query, we can filter for failed login attempts after business hours. This is done by selecting all the rows from the log\_in\_attempts category where the login\_time is higher than “18:00” and the login was unsuccessful. We use the “SELECT”, “FROM”, “WHERE”, and “AND” clauses in order to do this query to give us the appropriate results.



## Retrieve login attempts on specific dates

For this segment, we want to look at a potential threat event that happened on 2022-05-09. Therefore, we need to get all the login attempts on this day as well as the day before. We can complete this SQL query by selecting all the rows from log\_in\_attempts and specifically pulling the information from when the login\_date was the day of the incident or the date before. In this scenario using the “SELECT”, “FROM”, WHERE”, and “OR” clauses allows us to retrieve the proper information we need.



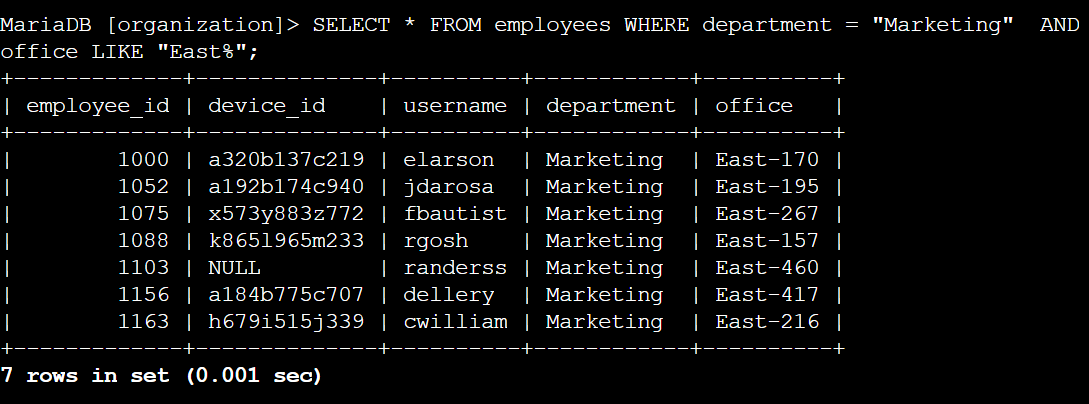
## Retrieve login attempts outside of Mexico

For this part, we need to get information about certain logins that did not originate in Mexico. For this SQL query we have to realize that certain records may list Mexico as “MEX” or as “MEXICO”. Therefore, it is important to use “MEX%” with the “%” which allows us to search for any information that starts with “MEX” but may or may not have additional letters after. In this query, we also use the “NOT” operator which allows us to search for login attempts that are not from Mexico.



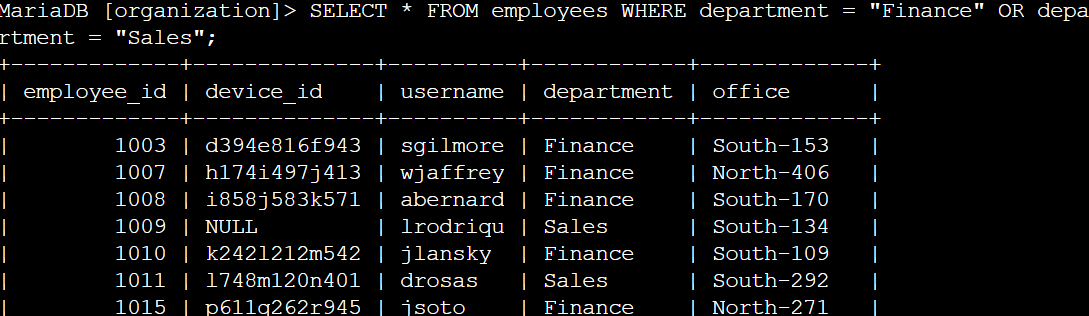
## Retrieve employees in Marketing

In this task, we need to retrieve all the information about employees in the marketing department whose offices are in the East building such as “East-170” or “East-320”. Due to multiple East building numbers, we must use the “LIKE” operator to clarify that the office has to start with “East” but can have different characters after such as “East-170” or “East-320”.



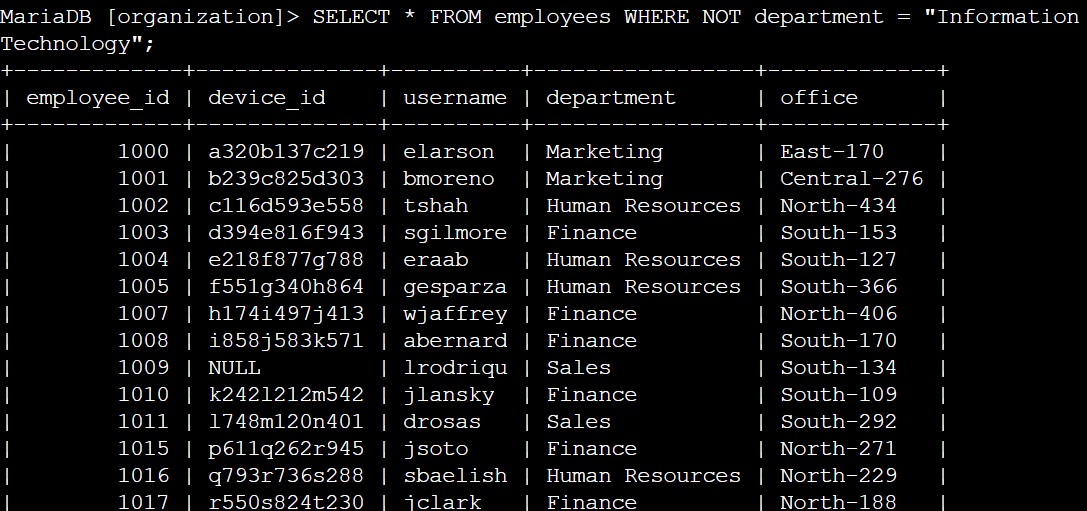
## Retrieve employees in Finance or Sales

Here we need a list of employees in the Finance or Sales department. We do this by following the same format as the previously used SQL queries except we use the “OR” operator. This helps us pull the information for employees in the Finance or the Sales department at the same time.



## Retrieve all employees not in IT

For this final task, we are told that the employees in the IT department have received an update that the rest of the departments still need. In order to see who still needs the update, we need to pull the information for everyone who is not in IT. In order to do this, we use the “NOT” operator which helps us specify the only department we don’t want the employees from.



## Summary

This scenario was completed using SQL queries which helped to retrieve specific information about login attempts, dates, and employee department information. By using operators such as “AND”, “OR”, and “NOT” to further narrow the search to include the desired information from the dataset. Using additional operators like “LIKE” and “%” help to filter for different patterns. This assignment taught me how to better use SQL to efficiently pull the desired information from a dataset by using different operators and specifications.