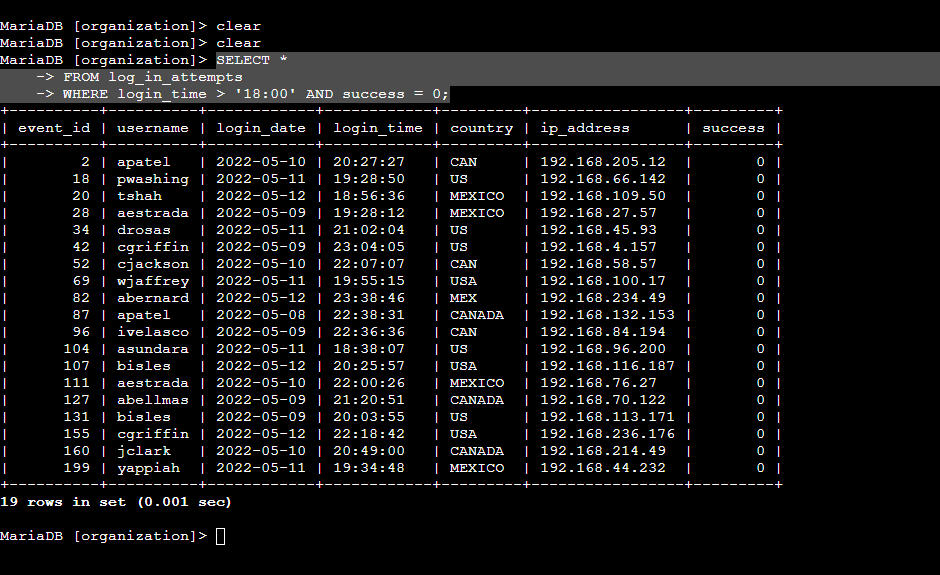
# Apply filters to SQL queries

## Project description

This project is examining an organization’s data related to employee login attempts. Using SQL, I am using queries to filter the data to retrieve the relevant information.

## Retrieve after hours failed login attempts



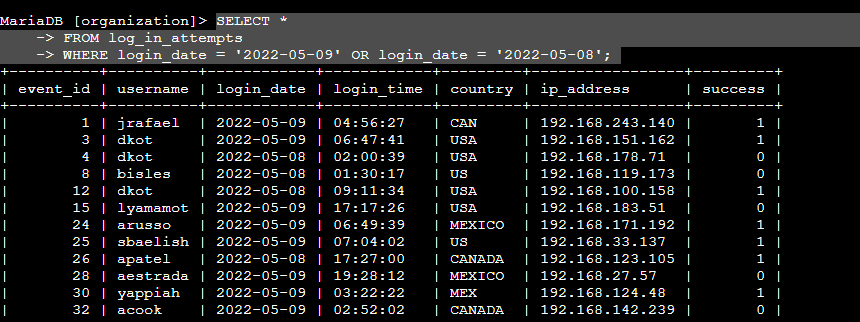
In order to retrieve after hours failed login attempts, I used the following query:

SELECT \* I want to retrieve all of the data that meets the specified conditions.

-> FROM log\_in\_attempts This is the table I want to retrieve the data from.

-> WHERE login\_time > ‘18:00’ AND success = 0; I want to retrieve the login attempts that happened after 18:00, AND also failed. The 0 represents the boolean false.

## Retrieve login attempts on specific dates



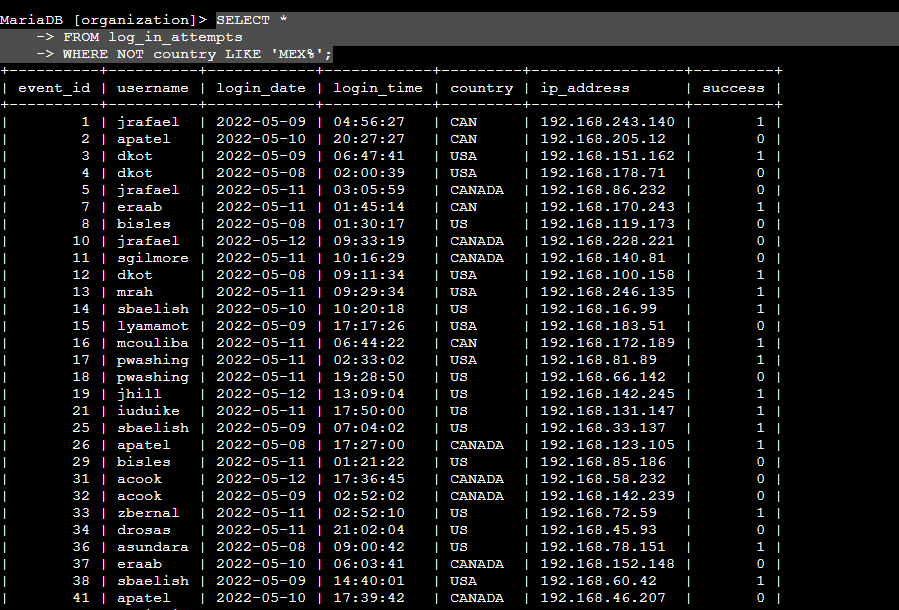
In order to retrieve the login attempts that occurred on either May 8th, 2022 or May 9th, 2022, I used the following query:

SELECT \* I want to retrieve all of the data that meets the specified criteria.

-> FROM log\_in\_attempts This is the table I want to retrieve data from.

-> WHERE login\_ date = ‘2022-05-09’ OR login\_date = ‘2022-05-08’; I only want to retrieve the login attempts that occurred on May 9th, 2022, or May 8th 2022. The OR operator is an inclusive operator so it will return all logins that occurred on those 2 days.

## Retrieve login attempts outside of Mexico



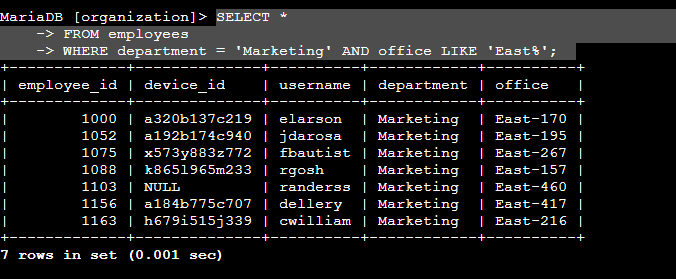
In order to retrieve the login attempts that occurred outside of Mexico I used the following query:

SELECT \* I want to retrieve all data that meets the specified criteria.

-> FROM log\_in\_attempts This is the table I want to retrieve data from.

-> WHERE NOT country LIKE ‘MEX%’; I use the NOT operator because I do not want to retrieve any results that match a specific country. Because Mexico is sometimes abbreviated, I use the LIKE operator and ‘MEX%’ because I do not want to include any values that start with MEX. The percent sign after MEX ensures that any characters after MEX are accounted for.

## Retrieve employees in Marketing



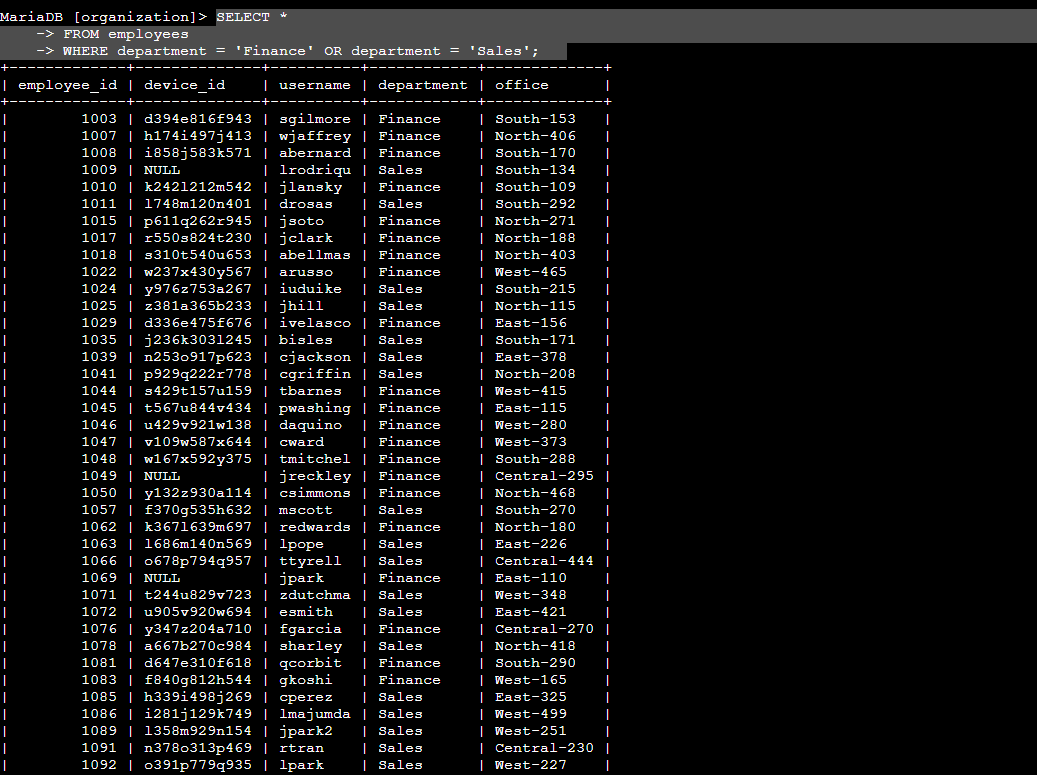
In order to retrieve all employees that work in the Marketing department I used the following query:

SELECT \* I want to retrieve all of the data that meets the specified criteria.

-> FROM employees This is the table I want to retrieve data from.

-> WHERE department = ‘Marketing’ AND office LIKE ‘East%’; I only want to retrieve employees that are in Marketing, and work in an office that starts with East. Similar to the last query, they need to match both conditions. Because the offices start with East and have different values afterwards, I use the LIKE operator and ‘East%’ to retrieve all employees who work in an office that begins with East.

## Retrieve employees in Finance or Sales



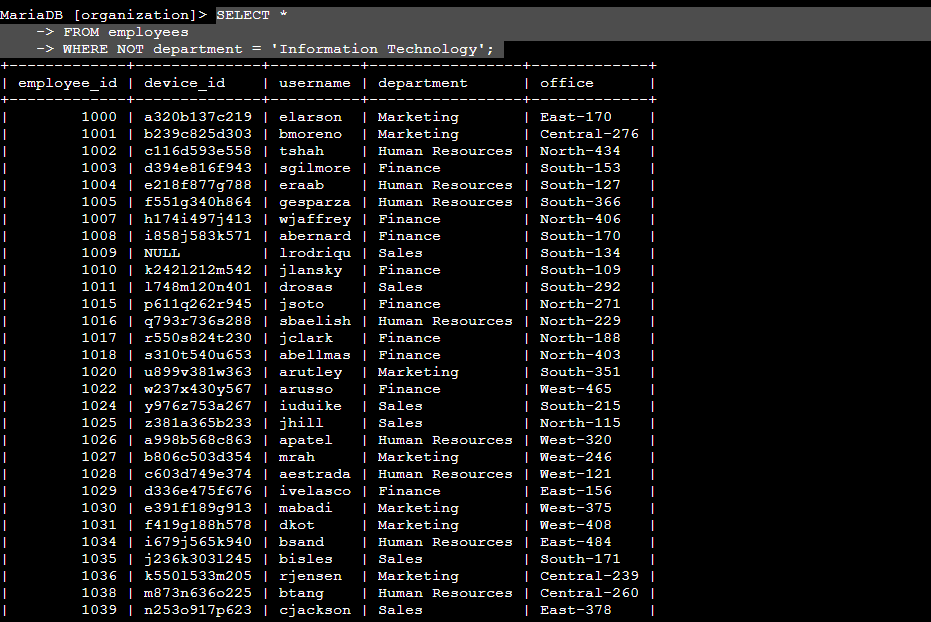
In order to retrieve all of the employees that work in Finance or Sales, I use the following query:

SELECT \* I want to retrieve all data that meets the specified criteria.

-> FROM employees This is the table I want to retrieve data from.

-> WHERE department = ‘Finance’ OR department = ‘Sales’; I want to retrieve the data for all of the employees that work in either Finance or Sales. The OR operator ensures that the data returned will meet one of those two conditions.

## Retrieve all employees not in IT



In order to retrieve all employees that are not in the IT department I use the following query:

SELECT \* I want to retrieve all of the data that meets the specified criteria.

-> FROM employees This is the table I want to retrieve data from

-> WHERE NOT department = ‘Information Technology’; I use the NOT operator to ensure that all rows that have Information Technology in the department column are excluded from the results.

## Summary

By using SQL filters to return only the specific information I was looking for, I was able to quickly narrow down large data results into only the relevant information.