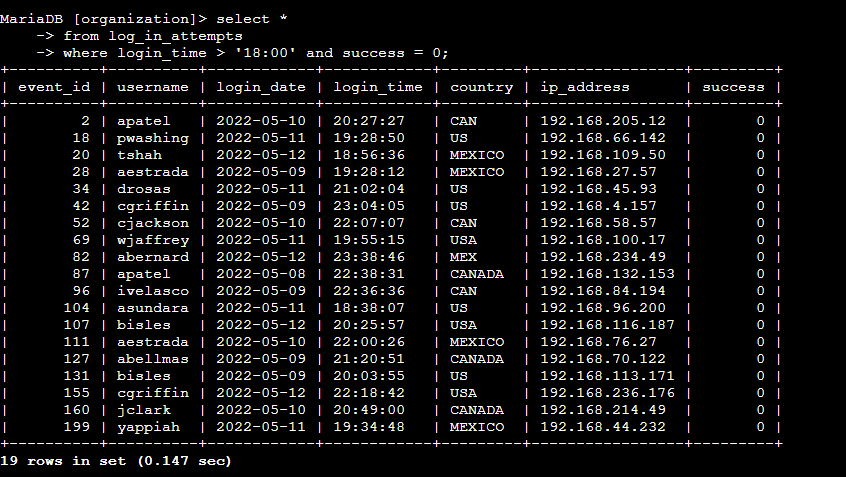
# **Apply filters to SQL queries**

## **Project description**

## In this project, SQL is used to filter and retrieve relevant data for cybersecurity investigations. By applying specific filters, I can identify suspicious login activity, isolate data from specific departments, and ensure efficient data management to enhance security measures.

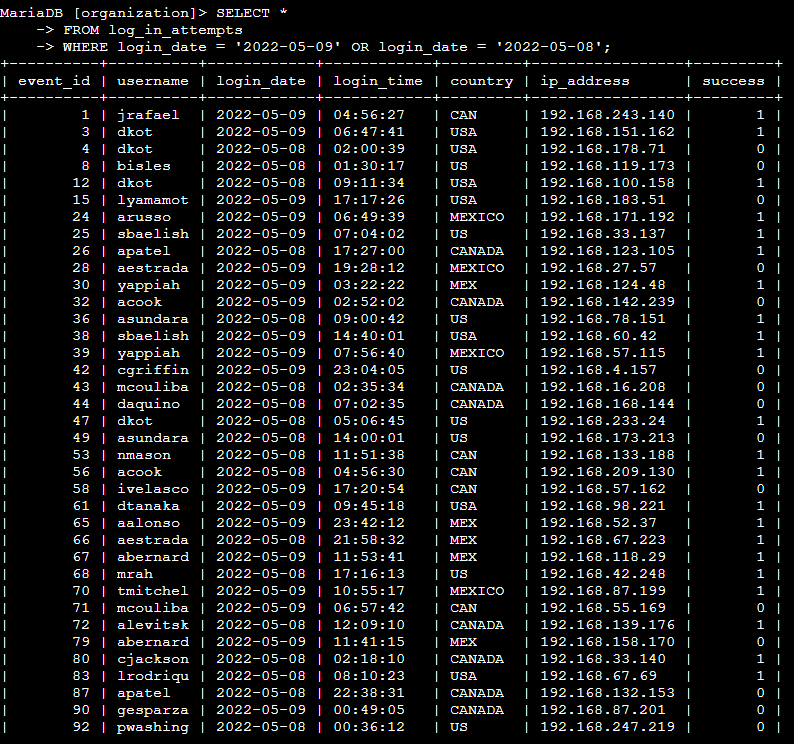
## **Retrieve after hours failed login attempts** To investigate potential unauthorized access after working hours, I used SQL to filter login attempts based on the time of the attempt and the status of the login.

The login\_time column in the log\_in\_attempts table contains information on when login attempts were made. Office hours end at '18:00'.

The success column in the log\_in\_attempts table contains values of TRUE or FALSE to indicate whether the login was successful. MySQL stores Boolean values as 1 for TRUE, and 0 for FALSE. This means that TRUE is represented as 1, and FALSE represented as 0 in the success column.  


## **Retrieve login attempts on specific dates** To review all login attempts that occurred on **2022-05-09** and **2022-05-08**, I was using the SQL WHERE clause with the OR operator to filter the login\_date column for the specific dates. Here is the SQL query and an explanation of how it works:

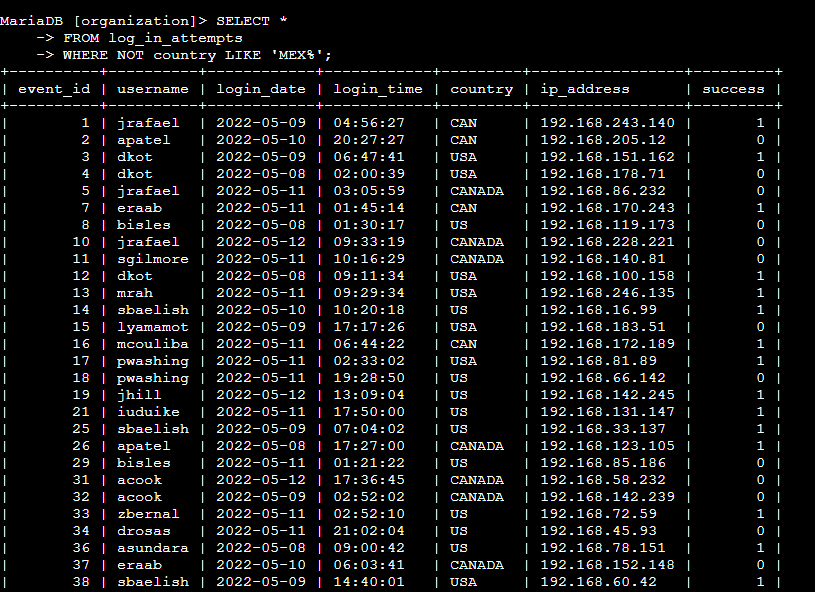
## This query will return all login attempts made on **2022-05-09** and **2022-05-08**, providing the necessary information to investigate the suspicious event.



**Retrieve login attempts outside of Mexico**  
  
To find login attempts that did not originate from Mexico, you can use the NOT and LIKE operators with the matching pattern 'MEX%', which will filter out any logins where the country is either 'MEX' or 'MEXICO'. Here's the query and an explanation of how it works:

**WHERE country NOT LIKE 'MEX%'**: This condition filters out login attempts where the country starts with 'MEX', which includes both 'MEX' and 'MEXICO'. The NOT LIKE ensures that only login attempts from countries other than Mexico are returned.

The pattern 'MEX%' matches any value that starts with 'MEX', and using NOT LIKE excludes those values. Therefore, this query will retrieve all login attempts that did not originate from Mexico.



## **Retrieve employees in Marketing** To retrieve all employees in the Marketing department who are located in the East building, you can create a SQL query using the LIKE keyword to filter for office locations that start with "East-" and the WHERE clause to filter by department. Here's the query and an explanation of how it works: **AND office LIKE 'East-%'**: Further filters the results to include only those employees whose office is in the East building. The LIKE 'East-%' uses the wildcard % to match any office location that starts with "East-" and can be followed by any number of characters (e.g., East-170, East-320).

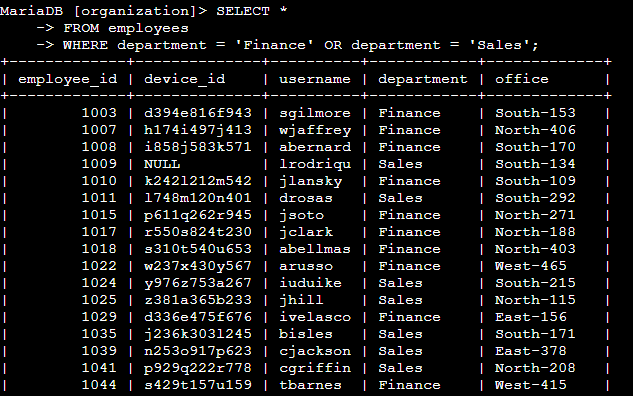
## marketing dept

## **Retrieve employees in Finance or Sales**

**WHERE department = 'Finance'**: This condition filters the results to only include employees who work in the Finance department.

**OR department = 'Sales'**: This condition adds an alternative option, where employees who work in the Sales department are also included. The OR operator ensures that employees from either the Finance or Sales departments are retrieved.

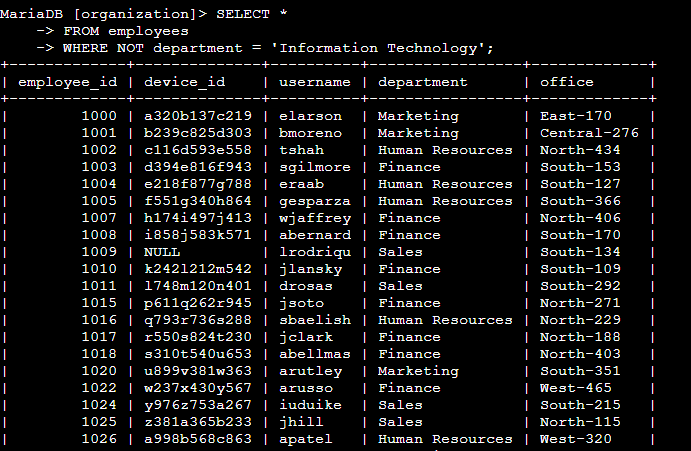
This query will return a list of all employees who belong to either the Finance or Sales department.



**Retrieve all employees not in IT**

**WHERE NOT department = 'Information Technology'**: This condition filters out employees whose department is 'Information Technology'. The NOT operator excludes any rows that match this department, meaning only employees from other departments will be returned.

This query will return all employees from departments **other than** Information Technology. It's useful when you need to focus on employees from different departments and exclude IT-specific staff.



## 

## **Summary**

This project demonstrates how SQL can be used to apply various filters to extract meaningful data for cybersecurity purposes. From isolating failed login attempts to retrieving department-specific employee data, SQL provides a flexible and powerful tool for managing data in a security context. The queries in this project show practical examples of filtering login attempts by time, location, date, and department, all essential for effective cybersecurity analysis.