# **Apply filters to SQL queries**

## ***Project description***

In our recent project, we've uncovered a potential security breach that occurred outside of business hours. To investigate further, we needed to query the log\_in\_attempts table and review after-hours login activity. Using SQL filters, we created a query that identified all failed login attempts happening after 18:00. Our investigation aims to enhance security measures and ensure the integrity of our systems.

## ***Table Format***

The organization database contains the following two tables:

● log\_in\_attempts

● employees

## **log\_in\_attempts**

The log\_in\_attempts table has the following columns:

● event\_id: The identification number assigned to each login event

● username: The username of the employee

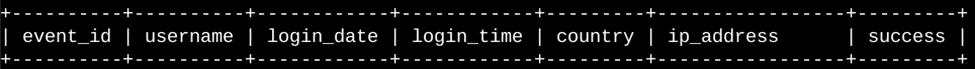
● login\_date: The date the login attempt was recorded

● login\_time: The time the login attempt was recorded

● country: The country where the login attempt occurred

● ip\_address: The IP address of that employee’s machine

● success: The success of the login attempt; FALSE indicates a failed attempt

In the MariaDB shell, these columns are returned as:

## 

## **employees**

The employees table has the following columns:

● employee\_id: The identification number assigned to each employee

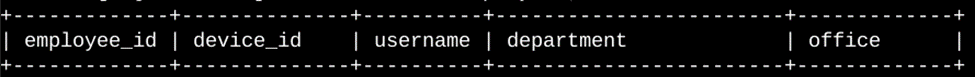
● device\_id: The identification number assigned to each device used by the employee

● username: The username of the employee

● department: The department the employee is in

● office: The office the employee is located in

In the MariaDB shell, these columns are returned as:



## 

## ***Retrieve after hours failed login attempts***

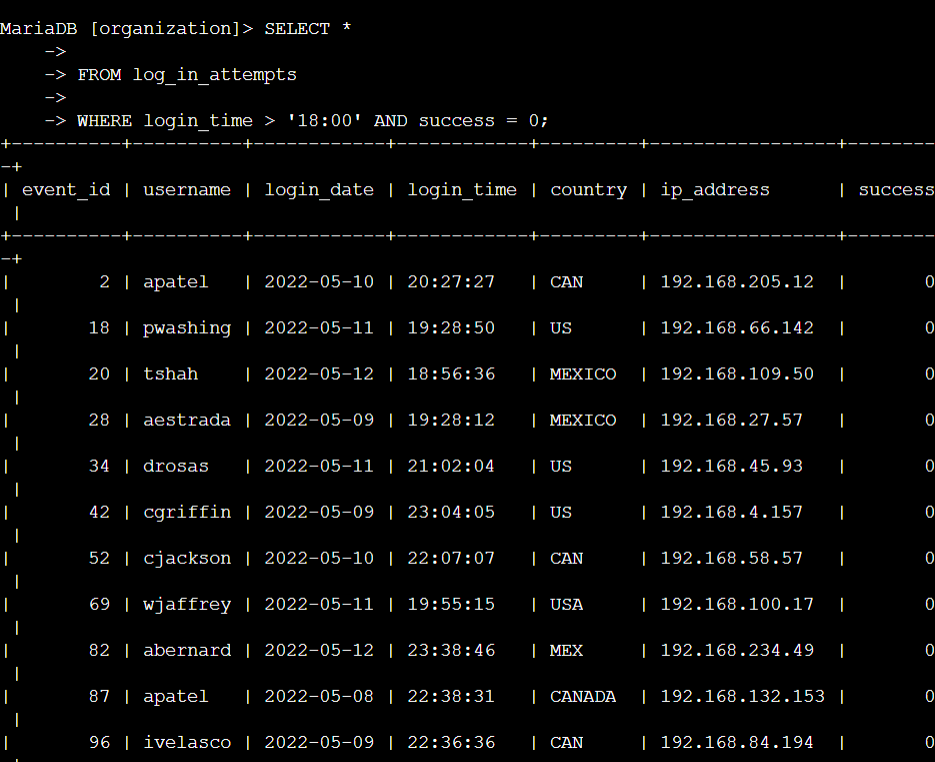
We need to investigate failed login attempts that happened after business hours. To do this, we'll get the information from the login activity. We need to find all unsuccessful attempts made after 18:00.

In the log\_in\_attempts table:

The login\_time column tells me when the login attempts were made. Office hours end at 18:00.

The success column shows whether the login was successful or not, with 1 for TRUE (successful) and 0 for FALSE (unsuccessful).

I'll use the AND operator to find the failed login attempts that occurred after business hours. I need to replace X and Y with the correct values to filter the records I need.



## 

## 

## 

## 

## 

## 

## 

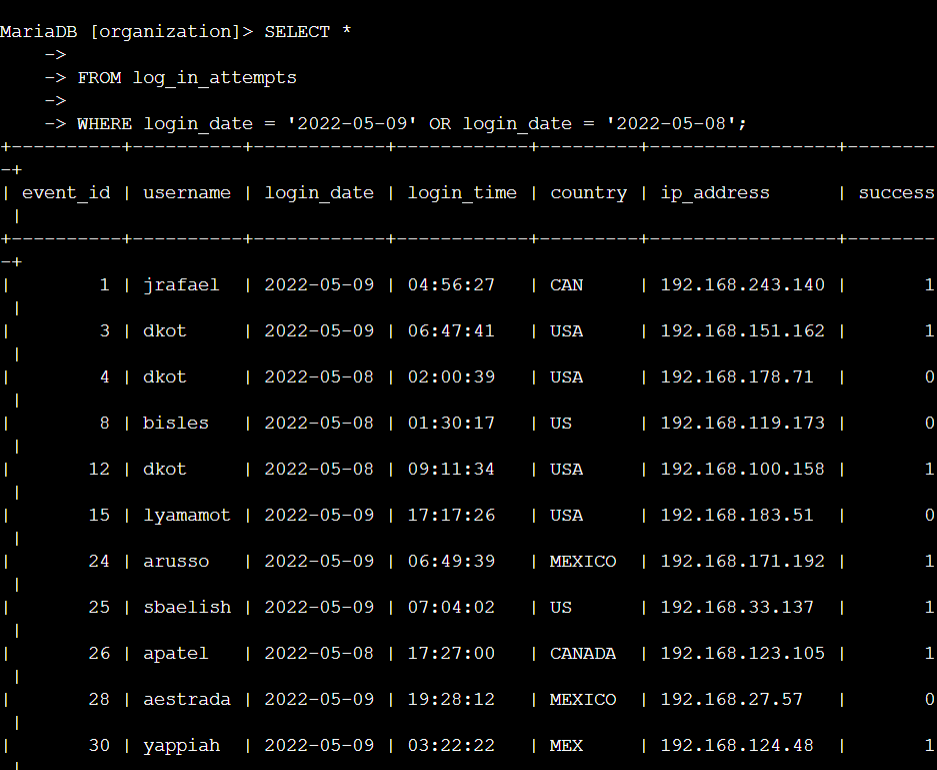
## 

## 

## 

## ***Retrieve login attempts on specific dates***

The team is investigating a suspicious event that occurred on '2022-05-09'. To retrieve all login attempts that occurred on this day and the day before ('2022-05-08').

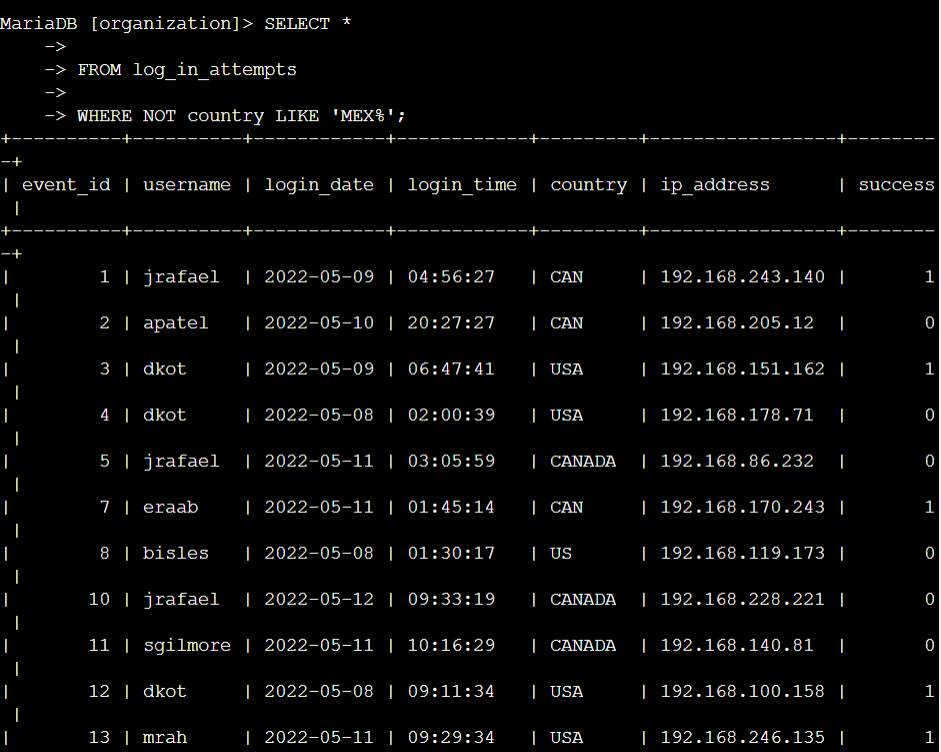


## 

## 

## ***Retrieve login attempts outside of Mexico***

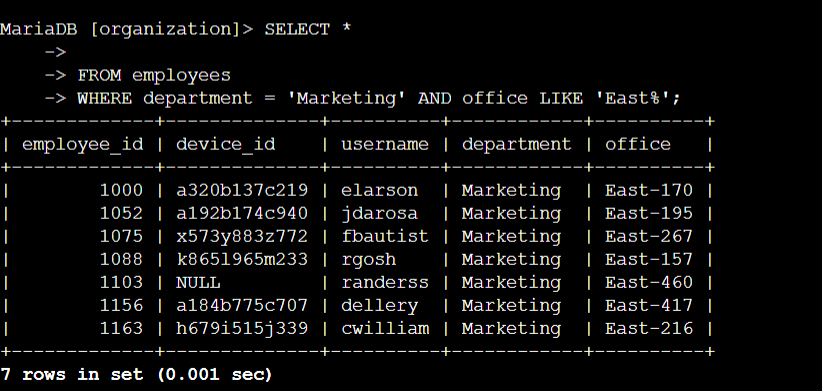
Suspicious activity has been detected with login attempts, but the team has determined that this activity did not originate in Mexico. Now, it is necessary to investigate login attempts that occurred outside of Mexico. Use filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico. When referring to Mexico, the country column contains values of both MEX and MEXICO, so the LIKE keyword with % should be used to ensure the query reflects this.



## 

## ***Retrieve employees in Marketing***

The team intends to execute security updates on designated employee machines within the Marketing department. Responsibility lies with acquiring information pertaining to these employee machines, necessitating a query on the employees table. Utilize SQL filters to formulate a query identifying all employees within the Marketing department across offices situated in the East building. (The department of the employee is identifiable in the department column, encompassing values such as Marketing. The office is discernible in the office column, featuring values like East-170, East-320, and North-434. Employ the LIKE keyword with % to narrow down the search for the East building.)



## 

## 

## 

## 

## 

## 

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

The team now needs to perform a different security update on machines for employees in the Sales and Finance departments. Use filters in SQL to create a query that identifies all employees in the Sales or Finance departments. (The department of the employee is found in the department column, which contains values that include Sales and Finance.)

## 

## 

## 

## 

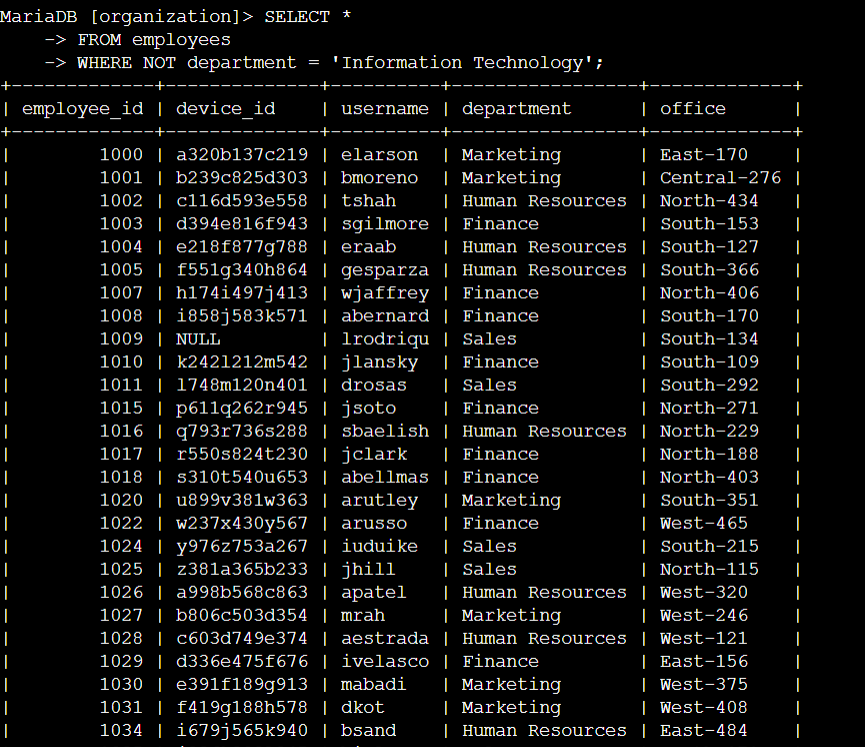
## 

## 

## 

## 

## ***Retrieve all employees not in the IT***

The team needs to make one more update to employee machines. Employees in the Information Technology department have already had this update, but employees in all other departments need it. Use filters in SQL to create a query which identifies all employees not in the IT department. (The department of the employee is found in the department column, which contains values that include Information Technology.)

## 

## 

## ***Summary***

In prior tasks, various SQL queries were executed to investigate security incidents and perform updates on employee machines. These queries involved filtering login attempts, identifying suspicious activity, and targeting specific departments and locations. The scenario presented now requires another update to employee machines, focusing on those not in the IT department. Building upon previous query techniques, filters in SQL will be applied to identify all employees outside the IT department, ensuring comprehensive coverage for the security update.