# Apply filters to SQL queries

## Project description

As a security professional it is my duty to investigate security issues to help keep our systems secure. I must review potential security issues that involve login attempts and employee machines using SQL to filter though our employees and log\_in\_attempts tables.

## Retrieve after hours failed login attempts

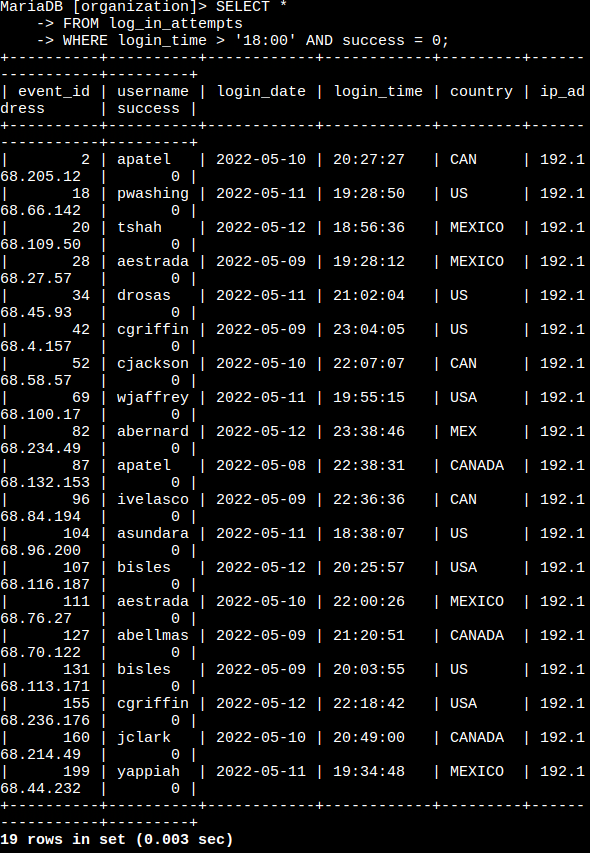
Our team is investigating failed login attempts after business hours. To identify failed login attempts after 6pm we query our SQL table with:

SELECT \*

-> FROM log\_in\_attempts

-> WHERE login\_time > '18:00' AND success = 0;

The result from the query above shows us there is 19 failed login attempts after 6pm:



## Retrieve login attempts on specific dates

Our team is investigating a suspicious event that occurred on 2022-05-09 and we want to retrieve all login attempts from this day and the day before.

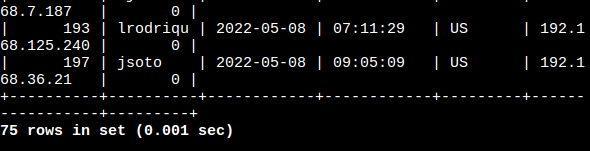
To achieve this, we use:

SELECT \*

FROM log\_in\_attempts

WHERE login\_date = '2022-05-08' OR login\_date = '2022-05-07';

Our result from the query above returns us 75 rows of data:



## Retrieve login attempts outside of Mexico

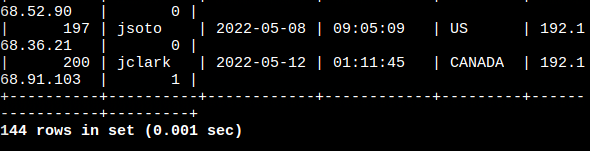
Now our team is investigating logins from locations excluding Mexico. To find this information we use the following query:

SELECT \*

FROM log\_in\_attempts

WHERE not country LIKE 'MEX%';

From the query we receive 144 rows of data from the log\_in\_attempts table where the country string does not start with the characters ‘MEX’ to exclude entries relating to Mexico.



## Retrieve employees in Marketing

Our team wants to gather information from all employees in the Marketing department residing in the East Building for a security update.

To achieve this we use the query:

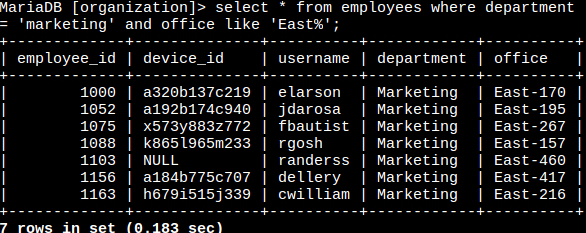
SELECT \*

FROM employees

WHERE department = ‘Marketing’ AND office LIKE ‘East%’;

In the WHERE clause of query above we combine two filters with the AND keyword. One of the filters is for an exact matching using = and the other using an approximation with the LIKE keyword. With our LIKE approximation we’re able to return all offices that start with “East” using the % wildcard at the end of the word in our filter.

The data we receive from running the filter only gives us entries from the Marketing department and offices that begin with ‘East’:



## Retrieve employees in Finance or Sales

Our team now needs to perform a different security update on machines for employees in the Sales and Finance departments.

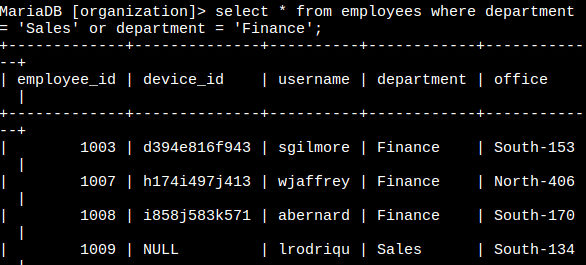
In order to pull that information we use the following query:

SELECT \*

FROM employees

WHERE department = ‘Sales’ OR department = ‘Finance’;

Although the filter was for the same column in this case we still had to specify Sales and Finance one at a time separating them with the OR keyword so we could include results from both:



## Retrieve all employees not in IT

The team needs to make one more update. This update was already made for the users of the Information Technology department so we need to apply the update to every other department.

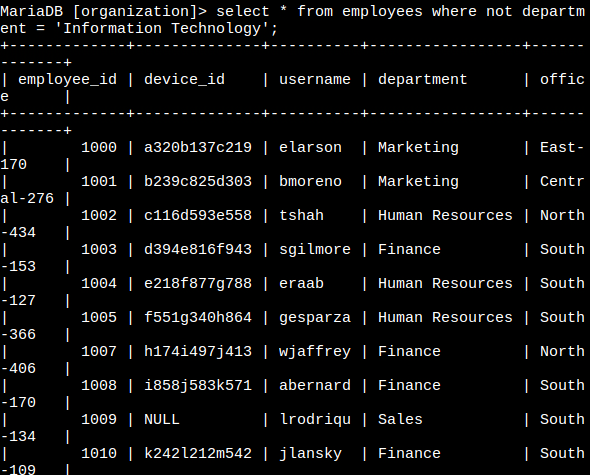
The query we need for that is:

SELECT \*

FROM employees

WHERE NOT department = ‘Information Technology’;

In the above query, we utilize the NOT keyword to exclude the Information Technology department from appearing in our search results:



## Summary

The purpose of this activity was to demonstrate a few of the ways we can filter information using SQL in several real world scenarios. Using the WHERE clause in tandem with the AND, OR, and NOT keywords allows us to request data using a wide range of criteria. Using the % wildcard and LIKE allows us to even further expand our search queries by allowing us to approximate our searches.