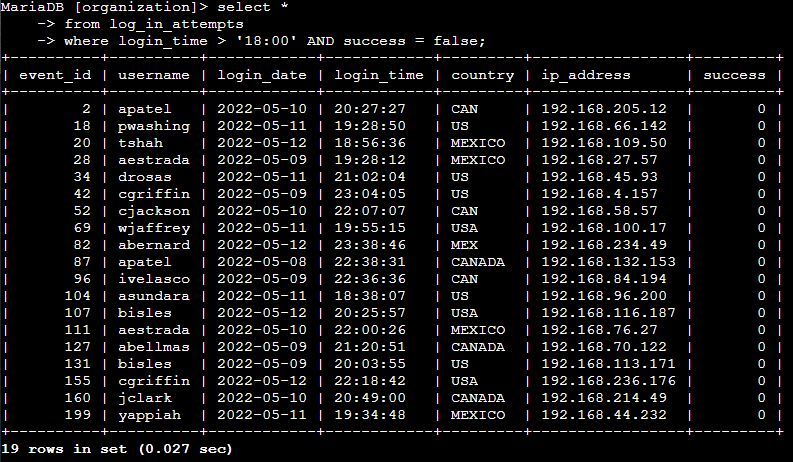
# Apply filters to SQL queries

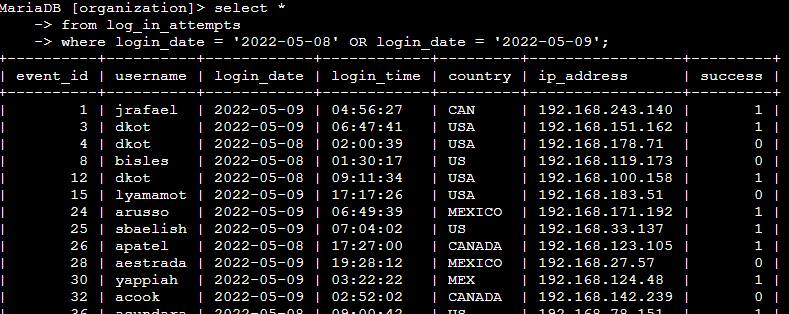
## Project description

I will be using SQL to find out specific information about employees, their machines, and their departments.

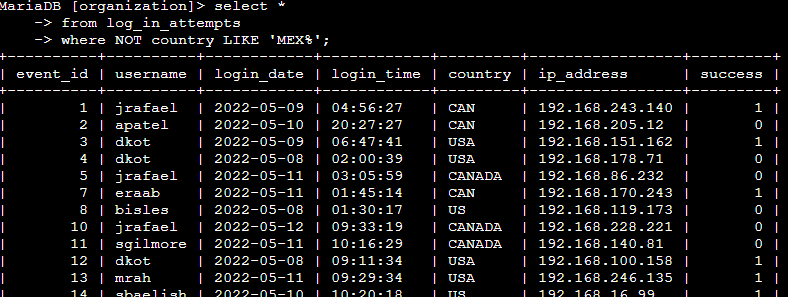
## Retrieve after hours failed login attempts

After hours in this scenario means after 6:00 P.M. (18:00). The first part of my query, select \*, tells the system which data I would like to see, with an asterisk denoting that I would like to see all the data from a given table. The from part of the query specifies the table I want to return the selected data from; in this case it is log\_in\_attempts. The operator AND in the where part of the query allows me to use 2 conditions in this query, those being: login\_time > '18:00', tells the system to only output results later than 18:00, and success = false, this tells the system to only output results where the login attempt was failed.

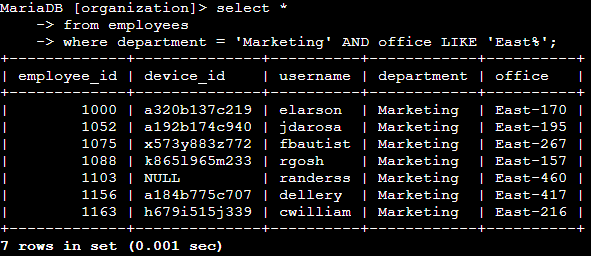
## Retrieve login attempts on specific dates

The dates for this scenario are May 8th and 9th, 2022. The list continues on for a total of 75 rows only including results from the 2 specified dates, but I cut the screenshot short in order to keep this document shorter. Again, I selected all data from the log\_in\_attempts table. The where line uses OR instead of AND, and uses 2 conditions of login\_date to see all data from those 2 specific days.

## Retrieve login attempts outside of Mexico

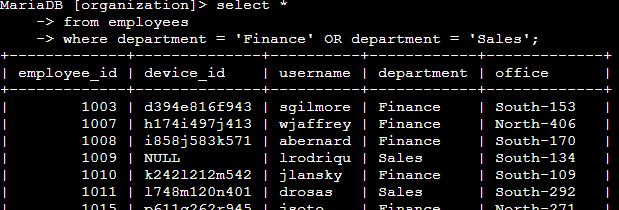
 Screenshot was cut short again, total of 144 rows with none of them containing a variation of Mexico. Once again, I used the same inputs for my first and second lines. This time the where line uses NOT to tell the system that I am looking for results that include everything except the next parameter country LIKE ‘MEX%’. country simply denotes a column in the table, LIKE ‘MEX%’ specifies the contents of that column that I would like to exclude. % is used because the contents of country are sometimes abbreviated, so the percent symbol tells the system that we want the characters before it in that exact order, but there can be any combination of characters after. This allows to filter out results that include either MEX or MEXICO.

## Retrieve employees in Marketing

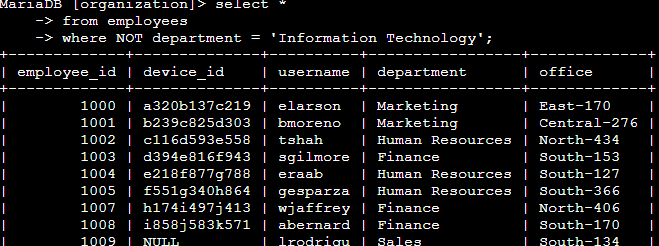
The marketing department is in the East building.

I selected all data again, but this time I am pulling from the employees table. I used AND and LIKE again in the WHERE line to filter results for the marketing department with the department = 'Marketing', and offices in the East building with office LIKE 'East%'. I also used % again because offices are all named with the building name first, followed by the room number,

## Retrieve employees in Finance or Sales

 I shortened the screenshot again, total of 71 rows only containing employees in the finance or sales departments. I’ve explained all of the concepts used in this SQL query already, but to reiterate: I selected all data from the employees table and filtered it with department = 'Finance' and department = 'Sales' using the OR operator which left me with only employees in the finance or sales departments.

## Retrieve all employees not in IT

 Shortened screenshot again, 161 rows without any of them containing an employee from IT. Again, selecting all data from the employees table, and using the NOT operator along with the condition, department = ‘Information Technology’, in the WHERE line, I was able to view all employees except for those in IT.

## Summary

I made use of filters in SQL queries to gather specific information in various databases. The operators like NOT, OR, AND, and LIKE were very important to use as they allow me to be much more specific in my queries.