# Apply filters to SQL queries

## Project description

In this SQL project, I utilized filtering techniques to investigate potential security issues within the organization's login attempts and employee databases.

## Retrieve after hours failed login attempts

This query filters the log\_in\_attempts table, extracting records where login attempts occurred after 18:00 and resulted in failure. Utilizing the login\_time column for time-based filtration and success column for failed attempts (marked as 0), it identifies potentially suspicious after-hours activity.

SELECT \*

FROM log\_in\_attempts

WHERE login\_time > '18:00'

AND success = 0;

## Retrieve login attempts on specific dates

This SQL query filters the log\_in\_attempts table to extract records where the login attempts occurred on either 2022-05-09 or 2022-05-08. Utilizing the login\_date column, it retrieves login attempts specifically on these two dates, aiding in the investigation of a suspicious event occurring within this two-day timeframe.

SELECT \*

FROM log\_in\_attempts

WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08';

## Retrieve login attempts outside of Mexico

This SQL query filters the log\_in\_attempts table to extract records where the login attempts didn't originate in Mexico. Utilizing the country column, the NOT LIKE 'MEX%' condition ensures that both 'MEX' and 'MEXICO' values are excluded, retrieving login attempts from locations other than Mexico. This query helps in investigating suspicious login activity by excluding attempts originating from within the Mexico region

SELECT \*

FROM log\_in\_attempts

WHERE country NOT LIKE 'MEX%';

## Retrieve employees in Marketing

This SQL query filters the employees table to extract records of employees specifically in the Marketing department, located in offices within the East building. Using the department column to filter 'Marketing' and the office column with the LIKE 'East-%' condition ensures retrieval of employees situated in offices denoted as part of the East building (e.g., East-170, East-320). This query enables the identification of employees in Marketing situated within the specified building for targeted security updates on their machines

SELECT \*

FROM employees

WHERE department = 'Marketing' AND office LIKE 'East-%';

## Retrieve employees in Finance or Sales

This SQL query filters the employees table to extract records of employees belonging to either the Finance or Sales departments. Leveraging the department column, the query utilizes the OR operator to retrieve employees associated with either 'Finance' or 'Sales' departments. This enables the identification of employees targeted for a different security update on their machines, ensuring inclusion of both specified departments in the query results

SELECT \*

FROM employees

WHERE department = 'Finance' OR department = 'Sales';

## Retrieve all employees not in IT

This SQL query filters the employees table to extract records of all employees not belonging to the Information Technology (IT) department. Employing the NOT operator alongside the = (equal) comparison, the query specifically excludes employees associated with the 'Information Technology' department. This query efficiently identifies all employees outside the IT department who require the pending update on their machines.

SELECT \*

FROM employees

WHERE NOT department = 'Information Technology';

## Summary

In this task, I demonstrated proficiency in SQL filtering techniques to address security concerns. Through targeted queries, I retrieved specific datasets, such as failed login attempts, login activities on specific dates, employee information in certain departments, and exceptions like activities outside of specified locations. This approach enables efficient investigation and targeted security measures within the organization