# Apply filters to SQL queries

## Project description

In this project, I utilized SQL to investigate potential security issues within an organization's database. By applying various filters to SQL queries, I retrieved specific records related to login attempts and employee information. This involved analyzing failed login attempts after business hours, investigating suspicious events on specific dates, identifying login attempts originating outside Mexico, and gathering information about employees in certain departments. These queries provided insights necessary for addressing security concerns and updating employee machines.

## Retrieve after hours failed login attempts

Query:

```SQL

SELECT \* FROM log\_in\_attempts

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

```

This query retrieves all records from the log\_in\_attempts table where login attempts failed (indicated by success = 0) and occurred after 18:00. This helps identify any unauthorized access attempts after business hours.

## Retrieve login attempts on specific dates

Query:

```SQL

SELECT \* FROM log\_in\_attempts

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

```

This query retrieves all records from the log\_in\_attempts table where login attempts occurred on May 8, 2022, or May 9, 2022. This helps investigate suspicious activities on these specific dates.

## Retrieve login attempts outside of Mexico

Query:

```SQL

SELECT \* FROM log\_in\_attempts

WHERE NOT country LIKE ‘MEX%’;

```

This query retrieves all records from the log\_in\_attempts table where the login attempts did not originate from Mexico. The LIKE keyword with MEX% ensures that it matches both 'MEX' and 'MEXICO'. The NOT operator excludes these entries, helping to focus on logins from other countries.

## Retrieve employees in Marketing

Query:

```SQL

SELECT \* FROM employees

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

```

This query retrieves all records from the employees table where the employee belongs to the Marketing department and is located in an office in the East building. The LIKE keyword with East% ensures that it includes all offices starting with 'East-'.

## Retrieve employees in Finance or Sales

Query:

```SQL

SELECT \* FROM employees

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

```

This query retrieves all records from the employees table where the employee belongs to either the Finance or Sales departments. The OR operator is used to include employees from both departments.

## Retrieve all employees not in IT

Query:

```SQL

SELECT \* FROM employees

WHERE department != ‘Information Technology’;

```

This query retrieves all records from the employees table where the employee does not belong to the Information Technology department. The != operator ensures that employees in IT are excluded from the results.

## Summary

Through this project, I applied various SQL filters to retrieve specific records related to login attempts and employee information. By filtering failed login attempts after business hours, identifying suspicious activities on specific dates, excluding logins from Mexico, and focusing on employees in particular departments, I was able to gather necessary data to address potential security issues and update employee machines. This exercise demonstrates the effectiveness of using SQL to perform detailed data analysis in a security context.