# Apply filters to SQL queries

## Project description

In this project, I undertook the role of a security professional within a large organization. My primary responsibility was to identify and address potential security issues to ensure the organization's system remains secure. The project involved investigating security concerns related to login attempts and employee machines. Utilizing SQL filters, I analyzed data from the employees and log\_in\_attempts tables to retrieve specific records and gain insights into the security incidents.

## Retrieve after hours failed login attempts

SELECT \*

FROM log\_in\_attempts

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

Explanation:

The SELECT \* statement retrieves all columns from the log\_in\_attempts table.

The WHERE clause is used to filter the results based on conditions.

login\_time > '18:00:00' filters for login attempts that occurred after 18:00.

success = 0 filters for failed login attempts, where the success column has a value of 0.

The combination of these conditions ensures that the query retrieves records of failed login attempts that occurred after 18:00.

## Retrieve login attempts on specific dates

SELECT \*

FROM log\_in\_attempts

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

Explanation:

The SELECT \* statement retrieves all columns from the log\_in\_attempts table.

The WHERE clause is used to filter the results based on conditions.

login\_date = '2022-05-09' OR login\_date = '2022-05-08' filters for login attempts that occurred on either May 9, 2022, or May 8, 2022.

The query combines the two conditions using the OR operator, which ensures that the query retrieves records of login attempts that occurred on the specified dates.

## Retrieve login attempts outside of Mexico

SELECT \*

FROM log\_in\_attempts

WHERE country NOT LIKE 'MEX%';

Explanation:

The SELECT \* statement retrieves all columns from the log\_in\_attempts table.

The WHERE clause is used to filter the results based on conditions.

country NOT LIKE 'MEX%' filters out login attempts where the country starts with 'MEX' (MEX or MEXICO).

The query uses the NOT LIKE operator with the % wildcard to match any country that doesn't start with 'MEX'. This way, the query retrieves records of login attempts that occurred outside of Mexico.

## Retrieve employees in Marketing

SELECT \*

FROM employees

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

Explanation:

The SELECT \* statement retrieves all columns from the employees table.

The WHERE clause is used to filter the results based on conditions.

department = 'Marketing' filters out employees who are not in the Marketing department.

office LIKE 'East-%' filters out employees whose office starts with 'East-', which represents the East building.

The query combines the AND operator to filter for employees in the Marketing department and offices in the East building using the LIKE operator with the % wildcard.

## Retrieve employees in Finance or Sales

SELECT \*

FROM employees

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

Explanation:

The SELECT \* statement retrieves all columns from the employees table.

The WHERE clause is used to filter the results based on conditions.

department = 'Sales' OR department = 'Finance' filters for employees in either the Sales or Finance departments.

The query combines the OR operator to filter for employees in either the Sales or Finance departments.

## Retrieve all employees not in IT

SELECT \*

FROM employees

WHERE department != 'Information Technology';

Explanation:

The SELECT \* statement retrieves all columns from the employees table.

The WHERE clause is used to filter the results based on conditions.

department != 'Information Technology' filters for employees whose department is not equal to 'Information Technology'.

The query utilizes the != operator (not equal) to filter out employees who are not in the IT department.

## Summary

Throughout the project, I utilized SQL queries and filters to retrieve crucial information from the organization's database tables. By crafting targeted queries, I investigated different aspects of security concerns, including failed login attempts, employee machine updates, and department-specific actions. My analysis helped identify patterns and potential risks, allowing the organization to take proactive measures to enhance security. This project showcases my proficiency in SQL filtering techniques and demonstrates my ability to contribute to maintaining a secure and resilient system within a complex organizational environment.