# Apply filters to SQL queries

## Project description

[Describe what you accomplish through SQL.]

## Retrieve after hours failed login attempts

A potential security incident took place outside of regular business hours, specifically after 6:00 PM. We need to conduct an investigation into all unsuccessful login attempts made during this time frame.

The provided code illustrates how I crafted an SQL query to identify failed login attempts that happened after the standard business hours.

A screenshot of a computer screen

Description automatically generated

The screenshot's first part displays my query, while the second part exhibits a segment of the output. This query is designed to sift through unsuccessful login attempts that took place after 18:00. To start, I began by selecting all the data from the log\_in\_attempts table. Then, I utilized a WHERE clause with an AND operator to filter the results, showing only login attempts that occurred after 18:00 and were unsuccessful. The first condition, login\_time > '18:00', pinpoints login attempts happening after 18:00. The second condition, success = FALSE, identifies those login attempts that didn't succeed.

## Retrieve login attempts on specific dates

An unusual event took place on 2022-05-09, prompting the need to investigate any login activity that occurred on that date or the preceding day.

The provided code showcases how I constructed a SQL query to target login attempts on particular dates.

A screen shot of a computer

Description automatically generated

The provided query is split into two parts, with the second section displaying a segment of the query's output. This query retrieves all login attempts that took place on either 2022-05-09 or 2022-05-08.

To achieve this, I initiated the query by selecting all data from the log\_in\_attempts table. Following this, I utilized a WHERE clause with an OR operator to refine the results, ensuring that only login attempts from either 2022-05-09 or 2022-05-08 are included. The first condition, login\_date = '2022-05-09', filters for logins on 2022-05-09, while the second condition, login\_date = '2022-05-08', targets logins on 2022-05-08.

## Retrieve login attempts outside of Mexico

Upon examining the organization's login attempt data, it has come to my attention that there may be an issue related to login attempts originating from locations outside of Mexico. It is imperative that these login attempts be thoroughly investigated.

Below, I present the SQL code that I devised to filter and identify login attempts originating from outside of Mexico.

A screen shot of a computer

Description automatically generated

The provided SQL query is designed to extract login attempts that originate from countries outside of Mexico. To achieve this, I initiated the query by selecting all relevant data from the log\_in\_attempts table. Subsequently, I employed a WHERE clause with the NOT operator to filter the results, ensuring that only entries from countries other than Mexico are displayed. The LIKE operator was utilized with the pattern "MEX%" to accommodate variations in how Mexico is represented in the dataset, including both "MEX" and "MEXICO." The percentage sign (%) serves as a wildcard character, matching any number of unspecified characters when used in conjunction with LIKE.

## Retrieve employees in Marketing

I was tasked with gathering information on the employee machines that need to be updated for the Marketing department in the East building. To accomplish this, I developed a SQL query as illustrated below: SELECT \*

>FROM employee\_machines

>WHERE department = 'Marketing'

>AND building = 'East';

This query effectively filters the employee machines, ensuring that only those belonging to the Marketing department in the East building are included in the results.

A black screen with white text

Description automatically generated

The query in the first part of the screenshot selects all data from the employees table and then uses a WHERE clause with the AND operator to filter for employees who are part of the Marketing department and located in the East building. The specific filter conditions are as follows:

1. **department = 'Marketing'**: This condition filters for employees in the Marketing department.
2. **office LIKE 'East%'**: This condition filters for employees located in the East building. It uses the LIKE operator with the pattern 'East%' to match office numbers that start with 'East'.

## Retrieve employees in Finance or Sales

The query in the first part of the screenshot selects all data from the employees table and then uses a WHERE clause with the OR operator to filter for employees who are part of either the Finance or Sales departments. By using the OR operator between these conditions, the query returns employees who belong to either the Finance or Sales departments.

A screenshot of a computer screen

Description automatically generated

The query displayed in the first part of the screenshot is designed to retrieve all employee data from the employees table. To narrow down the results, a WHERE clause is employed, utilizing the OR operator to filter for employees who belong to either the Finance or Sales departments. The choice of the OR operator, as opposed to AND, is made to ensure that all employees from either department are included.

By incorporating the OR operator between these conditions, the query effectively identifies employees who are part of either the Finance or Sales departments.

## Retrieve all employees not in IT

The query presented in the first part of the screenshot aims to retrieve information about employee machines from employees who are not part of the Information Technology department. To achieve this, a WHERE clause is employed with the NOT operator to filter out employees in the IT department.

Here's a breakdown of the conditions applied:

1. **department != 'Information Technology'**: This condition filters for employees whose department is not "Information Technology," effectively excluding IT department employees from the results.

By using the NOT operator, the query effectively identifies employees outside of the Information Technology department, facilitating the necessary security update.A screenshot of a computer screen

Description automatically generated

The query displayed in the first section of the screenshot is designed to retrieve data for employees who are not part of the Information Technology department. The query, as described, returns a list of employees who are not affiliated with the Information Technology department, which is essential for making a specific security update for this group of employees.

## Summary

I applied filters to SQL queries to retrieve specific information on login attempts and employee machines. Two different tables, log\_in\_attempts, and employees were utilized. I employed the AND, OR, and NOT operators to tailor the filters for each task's specific requirements. Additionally, I harnessed the power of the LIKE operator alongside the wildcard percentage sign (%) to filter for patterns in the data.