# Apply filters to SQL queries

## Project description

My organization in the wake of a recently failed brute force attack is working to make our system more secure. It is my job to ensure the system is safe, investigate all potential security issues, and provide hardening solutions e.g update employee computers as needed. The following steps provide examples of how I used SQL with filters to perform security-related tasks.

## Retrieve after hours failed login attempts

There were several after business hours (after 18:00). All after hours login attempts that failed need to be investigated.

The following code demonstrates how I created a SQL query to filter for failed login attempts that occurred after business hours.



The first part of the screenshot is my query. This query filters for failed login attempts that occurred after 18:00. First, I started by selecting all data from the log\_in\_attempts table. The SELECT \* statement is used to select all columns from a table, The FROM log\_in\_attempts clause specifies the table from which I want to retrieve data. In this case, it’s the “log\_in\_attempts” table.

Then, In the third line I used a WHERE clause with an AND operator to filter my results to output only login attempts that occurred after 18:00 and were unsuccessful. The first condition is login\_time > '18:00', which filters for the login attempts that occurred after 18:00, the second condition is success = FALSE, which filters for the failed login attempts, the AND keyword combined both conditions

The second part is the portion separated in dash border lines is the output - a table of all login attempts after 18:00. .

## Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or on the day before needs to be investigated.

The following code demonstrates how I created a SQL query to filter for login attempts that occurred on specific dates.



The first part of the screenshot is my query. This query returns all login attempts that occurred on 2022-05-09 or 2022-05-08. First, I started by selecting all data from the log\_in\_attempts table. Then, I used a WHERE clause with an OR operator to filter my results to output only login attempts that occurred on either 2022-05-09 or 2022-05-08. The first condition is login\_date = '2022-05-09', which filters for logins on 2022-05-09. The second condition is login\_date = '2022-05-08', which filters for logins on 2022-05-08.

The OR keyword combined both conditions. sThe second part is a portion of the output separated in dash border lines is the output - a table of all login attempts that occurred on either 2022-05-09 or 2022-05-08 .

.

## Retrieve login attempts outside of Mexico

After investigating the organization’s data on login attempts, I believe there is an issue with the login attempts that occurred outside of Mexico. These login attempts should be investigated.

The following code demonstrates how I created a SQL query to filter for login attempts that occurred outside of Mexico.



The first part of the screenshot is my query, this query returns all login attempts that occurred in countries excluding Mexico. First, I started by selecting all data from the log\_in\_attempts table. Then, I used a WHERE clause with NOT to filter for countries other than Mexico. The NOT command negates the condition that follows it. The LIKE command is used to search for a specified pattern in a column. In this case the pattern is ‘MEX%’. The ‘%’ sign is used to define wildcards (missing letters) both before and after the pattern, so ‘MEX%’ represent “any value that starts with ‘MEX’”. The second part is a portion separated in dash border lines is the output of every online that is not from Mexico.

## Retrieve employees in Marketing

My team wants to update the computers for certain employees in the Marketing department. To do this, I have to get information on which employee machines to update.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Marketing department in the East building.



The first part of the screenshot is my query, this query returns all employees in the Marketing department in the East building. First, I started by selecting all data from the employees table with the commands in the first two lines. Then on the third line, I used a WHERE clause with AND to filter for employees who work in the Marketing department and in the East building. I used LIKE with East% as the pattern to match because the data in the office column represents the East building with the specific office number. The first condition is the department = 'Marketing' portion, which filters for employees in the Marketing department. The second condition is the office LIKE 'East%' portion, which filters for employees in the East building. The AND command combines both conditions to get and output. The second part is a portion of the output, the output is partitioned with a dash border.

## Retrieve employees in Finance or Sales

In our pursuit of robust security, we’ve identified the need for specialized updates on the machines utilized by our colleagues in the Finance and Sales departments. Given the unique security requirements of these departments, it’s imperative that we first gather detailed information about the employees within these sectors. This will enable us to tailor our security updates to meet their specific needs, thereby enhancing our overall security infrastructure.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Finance or Sales departments.



The first part of the screenshot is my query, this query returns all employees in the Finance and Sales departments. First, I started by selecting all data from the employees table. Then, I used a WHERE clause with OR to filter for employees who are in the Finance and Sales departments. I used the OR operator instead of AND because I want all employees who are in either department. The first condition is department = 'Finance', which filters for employees from the Finance department. The second condition is department = 'Sales', which filters for employees from the Sales department. The OR command combined both conditions and get an output of all employees who are in either Finance and Sales departments

The second part The second part is a portion partitioned with a dash border is the output.

## Retrieve all employees not in IT

In order to enhance our security posture, my team is tasked with implementing an additional layer of security measures specifically targeting employees outside the Information Technology department. The first step in this process involves gathering comprehensive data on these individuals, enabling us to tailor our approach effectively.

The following demonstrates how I created a SQL query to filter for employee machines from employees not in the Information Technology department.



The first part of the screenshot is my query, The query returns all employees not in the Information Technology department. First, I started by selecting all data from the employees table. Then, I used a WHERE clause with NOT to filter for employees not in this department. The NOT command negated the department = 'Information Technology' condition and got an output of employees not in the Information Technology department . The second part output outlined in a bash border.

## Summary

I utilized filters in SQL queries to extract specific data related to login attempts and employee machines. The data was drawn from two distinct tables: ‘log\_in\_attempts’ and ‘employees’. To narrow down the information required for each task, I employed the AND, OR, and NOT operators. Additionally, I used the LIKE operator and the percentage sign (%) as a wildcard to filter for specific patterns.