# Apply filters to SQL queries

## Project description:

As a cybersecurity analyst, it is my responsibility to ensure my organization’s systems, analyze and investigate all potential security issues and takes necessary actions as outlined in the playbook. The following steps provide examples of how I used SQL with filters to analyze logs and data to perform security-related tasks.

## Retrieve after hours failed login attempts:

A screenshot of a computer screen

Description automatically generated

A potential security incident occurred after business hours, so all after hours failed login attempts needed to be investigated. I ran the query highlighted in the screenshot above to filter data in the “*log\_in\_attempts”* table for all failed login attempts after 18:00. I used “WHERE” with an “AND” operator to filter the data for the output shown in the screenshot. The first condition is login\_time > '18:00', which filters for the login attempts that occurred after 18:00. The second condition is success = 0, which filters for the failed login attempts.

## Retrieve login attempts on specific dates:

A screen shot of a computer

Description automatically generated

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.

To start the investigation, I ran the query highlighted in the screenshot above to filter for login attempts that occurred on 2022-05-09 or 2022-05-08 selecting all data from the “*log\_in\_attempts”* tables. I used a WHERE clause with an OR operator to filter my results to get the output in the screenshot. 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.

[Add content here.]

## Retrieve login attempts outside of Mexico:

A screen shot of a computer

Description automatically generated

After analysis of the filtered login attempts data, it is my believe that there is an issue with login attempts outside of Mexico. To investigate this, I filtered the data in the *“log\_in\_attempts”* table for login attempts from outside of Mexico by running the query highlighted in the screenshot above. The query was set up to select all data from the *“log\_in\_attempts”* table using the WHERE clause with NOT to filter for countries other than Mexico. I used LIKE with MEX% as the pattern to match because the dataset represents Mexico as MEX and MEXICO. The percentage sign (%) represents any number of unspecified characters when used with LIKE.

## Retrieve employees in Marketing:

A screenshot of a computer screen

Description automatically generated

A critical update needs to be added to a program used by the members of the marketing team. To get this started, I ran the highlighted query to filter for member of this group. The query selected all data from *“employee”* table using the WHERE keyword with = to set department as Marketing.

## Retrieve employees in Finance or Sales:

A screen shot of a computer

Description automatically generated

The machines for employees in the Finance and Sales departments also need to be updated. Since a different security update is needed. To filter for all those in this group, I ran the query highlighted above to select all data from the “*employee”* table using the WHERE clause with = to set the conditions. The OR operator sets the conditions as employees in either of both departments.

## Retrieve all employees not in IT:

A screenshot of a computer screen

Description automatically generated

Another patch update was needed for all employees except those in the Information Technology department. To apply this filter to the data in the “*employee”* table, I ran the highlighted query using the WHERE with = operator to set condition as all departments except Information Technology.

## Summary:

Using the AND, OR, and NOT operators, I was able to apply appropriate filters in my SQL queries on the data in “*log\_in\_attempts”* and “*employee”* tables to achieve desired results. I also used LIKE and the percentage sign (%) wildcard to filter for patterns.