# Apply filters to SQL queries

## Project description

SQL allows a Cybersecurity Analyst to search through Tables that have logs and data for specific patterns and uses. An Analyst will use these search method to isolate and pull out data using specific search parameters to filter the table until the results they are looking for is displayed. Conditions can be used as well within the search filtering to allow for specific targeted items within the tables.

## Retrieve after hours failed login attempts

## 

Within this attempt, we used the following commands in SQL:  
  
SELECT \*

FROM log\_in\_attempts

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

Within this we accomplished selecting all of the items within the table of ‘log\_in\_attempts’ where we filtered only the results of any login time that was greater than the time of 18:00 and we also filtered only for Unsuccessful attempts by using the Boolean argument where success is 0

## Retrieve login attempts on specific dates

## 

Within this attempt, we used the following SQL Arguments:  
  
SELECT \*  
FROM log\_in\_attempts  
WHERE login\_date = ‘2022-05-08’ OR login\_date = ‘2022-05-09’;  
  
Within this we accomplished selecting all of the items within the table of ‘log\_in\_attempts’ where we filtered the Login Dates of either ‘2022-05-08’ OR the login date of ‘2022-05-09’.

## Retrieve login attempts outside of Mexico

## 

Within this attempt we used the following SQL Arguments:

SELECT \*   
FROM log\_in\_attempts

WHERE NOT country like ‘MEX%’;  
  
Within this we accomplished selecting all of the items within the table of ‘log\_in\_attempts’ where we filtered out any country using the ‘NOT’ condition that started with the following 3 letters: MEX The % sign is used here as a wildcard as some countries are MEX whereas others are listed as MEXICO. This allowed to cover both items while ensuring the SQL was as clean as possible.

## Retrieve employees in Marketing

## 

Within this attempt we used the following SQL Arguments:

SELECT \*  
FROM employees  
WHERE department = ‘Marketing’ AND office LIKE ‘East%’;  
  
Within this we accomplished selecting all the items within the table of ‘employees’ where we filtered to only employees in Marketing by using the ‘department =’ and also using the AND condition along with LIKE to get all results similar to ‘East’ within the office column. Also we used the % to ensure that all ‘East’ Office numbers were included.

## Retrieve employees in Finance or Sales

## 

Within this attempt we used the following SQL Arguments:

SELECT \*  
FROM employees  
WHERE department = ‘Finance’ OR department = ‘Sales’;  
  
Within this we accomplished selecting all the items within the table of ‘employees’ where we filtered to only employees in Finance or Sales using the OR operator with the specific filters of each department.

## Retrieve all employees not in IT

## 

Within this attempt we used the following SQL Arguments:

SELECT \*  
FROM employees  
WHERE NOT department = ‘Information Technology’;  
  
Within this we accomplished selecting all the items within the table of ‘employees’ where we filtered out any employees using the ‘NOT’ condition, ensuring that any employee under the ‘Information Technology’ Department would be filtered out of the results, leaving all others.

## Summary

Within these previous tasks, it has been shown that I was able to not only filter out a table for specific timeframes, along with dates, but also retrieve based on if we want to filter OUT a specific term within the search parameters or filter in IN for specific items. Also using wildcards as well as using greater than, equals or less than to ensure we can extend the search timeframe or numeral frame.