# Using SQL to investigate login attempts

## Project description

As a security professional at an organisation, is my job to ensure the system is safe, investigate potential security issues, and update user privileges as required. The analysis below shows how I used SQL to investigate security issues with login attempts.

## Retrieve after hours failed login attempts

There was a potential security incident that occurred after business hours (after 18:00).

I needed to investigate login attempts that occurred after 18:00 that were unsuccessful.

The code below demonstrates the SQL query I used to filter for failed login attempts that occurred after business hours.



The first part of the screenshot is the query, and the second part is the first three rows of the output. This query filters for unsuccessful login attempts that occurred after 18:00.

I selected all columns from the log\_in\_attempts table. Then, in the WHERE clause, I used the AND operator to filter for two parameters:

* login\_time > '18:00', which filters for the login attempts that occurred after 18:00; and
* success = FALSE, which filters for unsuccessful login attempts.

## Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. To investigate, I looked at login activity that happened on 2022-05-09 or 2022-05-08 (the day before).

The code below demonstrates how used a SQL query to filter for login attempts that occurred on the two dates of interest.



The first part of the screenshot is the query, and the second part is the first three rows of the output. This query returns all login attempts that occurred on 2022-05-09 or 2022-05-08.

I selected all data from the log\_in\_attempts table. Then, I used a WHERE clause with an OR operator to include all records which occurred on either of two dates. The parameters were as follows:

* login\_date = '2022-05-09', which filters for logins on 2022-05-09; or
* login\_date = '2022-05-08', which filters for logins on 2022-05-08.

## Retrieve login attempts outside of Mexico

My investigation identified inconsistencies with the login attempts that occurred outside of Mexico. These login attempts should be investigated.

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



The first part of the screenshot is the query, and the second part is the first three rows of the output. This query returns all login attempts that occurred in countries other than Mexico.

I selected all data from the log\_in\_attempts table. Then, I used a WHERE clause with NOT to filter for countries other than Mexico. I used LIKE with MEX% as both MEX and MEXICO are used in the database to represent Mexico. The percentage sign (%) is a wildcard represents any number of unspecified characters when used with LIKE.

## Retrieve employees in Marketing

Employees from Marketing department in the East building needed updates to their computers. I needed to get information on which employee machines to update.

The code below demonstrates how I used SQL to filter for employee data for the abovementioned parameters.



The first part of the screenshot is the query, and the second part is the first three rows of the output. This query returns all employees in the Marketing department in the East building.

First, I started by selecting all data from the employees table. Then, I used a WHERE clause with AND to filter for employees who work in the Marketing department and in the East building. The parameters I entered are as follows

* department = 'Marketing', which filters for employees in the Marketing department; and
* office LIKE 'East%', which filters for employees in the East building. I used this format because office numbers were represented with ‘East’ followed by a hyphen and an office number.

## Retrieve employees in Finance or Sales

Employees in the Finance and Sales departments needed different security updates for their machines. I needed to get data on employees only from these two departments.

The code below demonstrates how I used SQL to filter for employee data for the abovementioned parameters.



The first part of the screenshot is the query, and the second part is the first three rows of the output. 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 either in the Finance department or in the Sales department. The parameters were as folllows:

* department = 'Finance', which filters for employees from the Finance department; or
* department = 'Sales', which filters for employees from the Sales department.

## Retrieve all employees not in IT

Another different security update is required for the machines of employees who are not in the Information Technology department.

I used the below SQL query to filter for employees who fit in the abovementioned group.



The first part of the screenshot is the query, and the second part is the first three rows of the output. The query returns all employees not in the Information Technology department. I started by selecting all data from the employees table. Then, I used a WHERE clause with NOT to filter for employees not in the IT department.

## Summary

I used filters with SQL queries to get specific information on login attempts and employee machines. I used two different tables, log\_in\_attempts and employees. I used the AND, OR, and NOT operators to filter for the specific information needed for each task. I also used LIKE and the percentage sign (%) wildcard to filter for patterns.