**Applying filters to SQL queries**

**Project Description**

In this project, I investigated potential security issues related to login attempts and employee machine updates at a large organization. Using SQL, I retrieved and filtered data from the log\_in\_attempts and employees tables to identify suspicious activities and ensure that necessary updates were applied to employee machines in specific departments. The focus was on utilizing various SQL operators, including AND, OR, NOT, and LIKE, to efficiently query and analyze the data.

**Retrieve After Hours Failed Login Attempts**

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

**SQL Query:**



**Explanation:** The first part of the screenshot is my query, and the second part is a portion of the output. This query retrieves all login attempts from the log\_in\_attempts table that occurred after 18:00 and were marked as failed (indicated by success = FALSE). The condition login\_time > '18:00' filters the records to include only those that happened after business hours, making it essential for identifying any suspicious after-hours activity.

**Retrieve Login Attempts on Specific Dates**

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

**SQL Query:**



**Explanation:** The first part of the screenshot is my query, and the second part is a portion of the output. 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.

**Retrieve Login Attempts Outside of Mexico**

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

**SQL Query:**



**Explanation:** The first part of the screenshot is my query, and the second part is a portion of the output.This query returns all login attempts that occurred in countries other than 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. 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.**

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.

**SQL Query:**



**Explanation:** The first part of the screenshot is my query, and the second part is a portion 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. 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.

**Retrieve Employees in Finance or Sales**

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

**SQL Query:**



**Explanation:** The first part of the screenshot is my query, and the second part is a portion 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 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.

**Retrieve All Employees Not in IT**

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

**SQL Query:**



**Explanation:** The first part of the screenshot is my query, and the second part is a portion of the output. 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.

**Summary**

I applied filters to 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.