**Apply filters to SQL queries**

**Project description**

This project involves querying a database to investigate potential security issues. By applying various SQL filters, I can retrieve specific information from the log\_in\_attempts and employees tables to identify suspicious activities and gather data for security updates.

**Retrieve after hours failed login attempts**

This query retrieves all records of failed login attempts that occurred after 18:00 (6:00 PM). It selects all columns from the log\_in\_attempts table. The WHERE clause filters these results based on two conditions combined with an AND operator. The first condition, login\_time > '18:00', selects attempts made after this time. The second condition, success = FALSE, selects only the attempts that were not successful.

SQL

SELECT \*

FROM log\_in\_attempts

WHERE login\_time > '18:00' AND success = FALSE;

**Retrieve login attempts on specific dates**

To investigate a suspicious event, this query retrieves all login attempts that occurred on either 2022-05-08 or 2022-05-09. It selects every column from the log\_in\_attempts table. The WHERE clause is used with an OR operator to filter for records where the login\_date is equal to '2022-05-09' or '2022-05-08', effectively combining the logs from both days into a single output.

SQL

SELECT \*

FROM log\_in\_attempts

WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08';

**Retrieve login attempts outside of Mexico**

This query identifies all login attempts that originated from outside of Mexico. It selects all columns from the log\_in\_attempts table. The WHERE clause filters the data using the NOT LIKE operator to exclude any entries where the country column begins with 'MEX'. This approach successfully filters out values such as 'MEX' and 'MEXICO'.

SQL

SELECT \*

FROM log\_in\_attempts

WHERE country NOT LIKE 'MEX%';

**Retrieve employees in Marketing**

This query gathers information on employees in the Marketing department who work in any of the East building offices. It selects all data from the employees table. The WHERE clause uses an AND operator to combine two conditions. The first condition, department = 'Marketing', isolates employees from that specific department. The second condition, office LIKE 'East%', uses the LIKE operator with a wildcard % to find all employees whose office location starts with 'East'.

SQL

SELECT \*

FROM employees

WHERE department = 'Marketing' AND office LIKE 'East%';

**Retrieve employees in Finance or Sales**

To prepare for a security update, this query retrieves a list of all employees from the Finance and Sales departments. It selects all columns from the employees table. An OR operator within the WHERE clause filters for employees where the department is either 'Finance' or 'Sales'.

SQL

SELECT \*

FROM employees

WHERE department = 'Finance' OR department = 'Sales';

**Retrieve all employees not in IT**

This query retrieves information for all employees who are not part of the IT department. It selects all columns from the employees table. The WHERE clause applies the NOT operator to the department column to exclude any employee whose department is 'IT'.

SQL

SELECT \*

FROM employees

WHERE NOT department = 'IT';

**Summary**

In this project, I utilized SQL filtering to investigate security-related events and gather information from an organization's database. I constructed queries using WHERE clauses in combination with logical operators such as AND, OR, and NOT to retrieve specific datasets. I filtered the log\_in\_attempts table to identify suspicious login patterns, such as failed attempts after business hours, logins on specific dates, and attempts from outside a designated country. I also queried the employees table to get lists of staff based on their department and office location to prepare for security updates. These tasks demonstrate the practical application of SQL in a cybersecurity context for monitoring and system administration.