Apply filters to SQL queries

Project description

My organization is working on strengthening cybersecurity across the company. Part of my responsibility is to use SQL to examine system activity and locate the exact employees and login events that require further review or updates. By applying SQL filters, including AND, OR, NOT, and pattern matching with LIKE, I am able to extract only the information necessary to support system security decisions and perform required updates efficiently.

Retrieve after hours failed login attempts

```
SELECT *
FROM log_in_attempts
WHERE login_time > '18:00'
  AND success = FALSE;
```

This query filters for failed login attempts occurring after 18:00.
I selected all data from the log_in_attempts table and then applied a WHERE clause using the AND operator:

- login_time > '18:00' filters only the login attempts performed after business hours.
- success = FALSE restricts the output to attempts that were not successful.
  Together, these conditions help identify failed logins that could represent unauthorized access attempts.

Retrieve login attempts on specific dates

A suspicious activity occurred on **2022-05-09**, so all login activity from that day and the previous day must be reviewed.

```
SELECT *

FROM log_in_attempts
```

```
WHERE login_date = '2022-05-09'

   OR login_date = '2022-05-08';
```

This query returns all login attempts from 2022-05-09 or 2022-05-08.
I selected all data from the log_in_attempts table and used the OR operator inside the WHERE clause:

- login_date = '2022-05-09' filters for logins on the day of the incident.

- login_date = '2022-05-08' includes logins from the day before, in case the incident originated earlier.
  Using OR ensures that login results from either date are included.

Retrieve login attempts outside of Mexico

While reviewing login activity, I discovered potential issues involving attempts made outside of Mexico. These attempts need to be analyzed.

```
SELECT *

FROM log_in_attempts

WHERE country NOT LIKE 'MEX%';
```

This query returns all login attempts that occurred in a country other than Mexico.
I selected all data from the log_in_attempts table and used a WHERE clause with NOT LIKE:

- The pattern LIKE 'MEX%' matches both MEX and MEXICO in the country column.

- Using NOT LIKE excludes those records and returns only countries outside Mexico.
  The % wildcard matches any characters following MEX.

Retrieve employees in Marketing

The Marketing department needs a computer update. To perform the update, I must determine exactly which employees are located in the East building.

```
SELECT *

FROM employees

WHERE department = 'Marketing'

  AND office LIKE 'East%';
```

This query returns all employees in the Marketing department who work in the East building.
I selected all data from the employees table and applied a WHERE clause with the AND operator:

- department = 'Marketing' filters to Marketing employees only.

- office LIKE 'East%' identifies only those who work in the East building (since their office location starts with “East”).
  Using LIKE allows us to match the building name regardless of the specific room number afterward.

Retrieve employees in Finance or Sales

A separate security update is required for employees in the Finance and Sales departments. Only employees from these two departments should be included.

```
SELECT *

FROM employees

WHERE department = 'Finance'

   OR department = 'Sales';
```

This query returns all employees who work in either Finance or Sales.
I selected all data from the employees table and used the OR operator within the WHERE clause:

- department = 'Finance' filters to Finance employees.

- department = 'Sales' filters to Sales employees.
  OR is used because employees from either department need to be included, not both.

Retrieve all employees not in IT

My last task is to update computers for employees who are **not** in the Information Technology department. To complete this update, I first need to identify these employees.

SELECT *

FROM employees

WHERE department NOT LIKE '%Information Technology%';

This query returns all employees who are not part of the Information Technology department.
I selected all data from the employees table and then applied a WHERE clause using NOT:

- department NOT LIKE '%Information Technology%' removes anyone whose department matches Information Technology.
  The wildcard % ensures that the pattern matches any department value containing that full phrase, even if additional text appears before or after.

Summary

In this project, I used SQL to filter information from both the log_in_attempts and employees tables in order to support cybersecurity decisions and complete required security updates. I applied a combination of AND, OR, and NOT operators to retrieve only the relevant records for each task. I also used pattern matching with LIKE and % to search for partial strings. These SQL skills helped isolate failed and suspicious login events and identify employees who needed system updates based on their departments and office locations.