Apply filters to SQL queries

Project description

As a security professional, my task was to use Structured Query Language (SQL) to investigate potential security incidents and retrieve employee data for machine updates. This involved querying the organization's employees and log_in_attempts tables. The queries demonstrate proficiency with essential filtering operators, including AND, OR, NOT, and LIKE, to efficiently segment and analyze large datasets.

Retrieve after hours failed login attempts

```
SELECT
  *
FROM
  log_in_attempts
WHERE
  success = 0 AND login_time > '18:00:00';
```

This query selects all columns (*) from the log_in_attempts table. The WHERE clause applies two conditions linked by the AND operator:

- success = 0 filters for failed login attempts.
- login_time > '18:00:00' filters for attempts occurring after 6:00 PM.

The AND ensures that only records meeting both the failure and after-hours criteria are returned for investigation.

Retrieve login attempts on specific dates

```
SELECT *
        FROM log_in_attempts
                WHERE login_date = '2022-05-09'
                        OR login_date = '2022-05-08';
```

This query retrieves all data (*) from the log_in_attempts table. The WHERE clause uses the OR operator to filter results by date:

- A record is included if login_date equals '2022-05-09' **OR** '2022-05-08'.

This successfully combines all login attempts from the two target days.

Retrieve login attempts outside of Mexico

```
SELECT * FROM log_in_attempts WHERE country NOT LIKE 'MEX%';
```

This query selects all data (*) from the log_in_attempts table. The WHERE clause uses NOT combined with LIKE to exclude records:

- 'MEX%' matches any country value starting with “MEX” (e.g., MEX, MEXICO).
- NOT LIKE filters out all records where the country is Mexico, isolating other login origins for review.

Retrieve employees in Marketing

```
SELECT * FROM employees WHERE department = 'Marketing' AND office LIKE 'East%';
```

This query selects all information (*) from the employees table. The AND operator ensures both conditions are met:

- department = 'Marketing' specifies the department.
- office LIKE 'East%' matches any office value starting with “East” (e.g., East-170).

Only employees in Marketing **and** the East building are returned.

Retrieve employees in Finance or Sales

```
SELECT * FROM employees WHERE department = 'Sales' OR department = 'Finance';
```

This query selects all data (*) from the employees table. The WHERE clause uses OR to filter for employees in either the 'Sales' **OR** 'Finance' department, providing a consolidated list of employees requiring the update..

Retrieve all employees not in IT

```
SELECT * FROM employees WHERE NOT department = 'Information Technology';
```

This query selects all employee information (*) from the employees table. The WHERE clause uses the NOT operator to exclude employees in the Information Technology department, preparing the final list for the security update.

Summary

This activity demonstrates practical application of SQL to solve common cybersecurity and operational data retrieval challenges. I successfully executed multiple queries using AND, OR, NOT, and LIKE to:

- Investigate security incidents (e.g., after-hours failed logins, non-Mexican login origins)
- Perform operational tasks (e.g., retrieving specific employee groups for machine updates)

This capability is essential for a security professional to efficiently manage and analyze organizational data.