Apply Filters to SQL Queries

In this project, I used SQL to support a simulated security investigation.

The goal was to retrieve specific login attempt and employee data using SQL filters such as AND, OR, NOT, LIKE, and date/time conditions.

These queries are critical in identifying suspicious activity, isolating security concerns, and preparing for targeted updates across departments.

The project demonstrates my ability to filter and extract meaningful insights from raw data using structured queries.

Retrieve after hours failed login attempts

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

This query identifies all failed login attempts (where success = 0) that occurred after 6:00 PM.

The AND operator ensures both conditions are met. This helps isolate potential security breaches happening after normal business hours.

Retrieve login attempts on specific dates

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

This query filters all login attempts that occurred on either May 8 or May 9, 2022.

Using OR ensures that both dates are explicitly checked. This is useful for investigating specific incidents on known suspicious dates.

Retrieve login attempts outside of Mexico

```
SELECT *
```

```
FROM log_in_attempts
WHERE country NOT LIKE 'MEX%'
  AND country NOT LIKE 'Mexico%';
```

This query removes all login attempts that originated from Mexico, whether the country code is listed as MEX, Mexico, or any similar variation.

NOT LIKE ensures flexibility in excluding all such entries, which is critical for narrowing down the location of suspicious logins.

Retrieve employees in Marketing

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

This query pulls all employee records where the department is 'Marketing' and the office location starts with "East-", covering variations like East-170 or East-320.

The LIKE operator helps identify office codes by pattern.

Retrieve employees in Finance or Sales

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

This query selects employees who are in either the Finance or Sales departments. Each condition is explicitly written out to satisfy formatting requirements. The OR operator is used to include records matching either department.

Retrieve all employees not in IT

```
SELECT *
FROM employees
WHERE department != 'Information Technology';
```

This query retrieves all employees except those in the Information Technology department.

The != operator filters out any record where the department equals 'Information Technology', allowing the team to focus updates on everyone else.

Summary

This project showcased the use of SQL filtering to conduct a security-focused data investigation.

I used logical operators like AND, OR, and NOT, along with pattern matching using LIKE, and time/date filtering to isolate login and employee data.

These queries enabled precise extraction of data necessary for identifying security events, targeting specific groups for system updates, and excluding irrelevant data.

This project highlights my ability to apply SQL in real-world, security-relevant scenarios.