Apply filters to SQL queries

Project description

In this activity, I used SQL queries to filter and retrieve specific information from a company's internal database. The goal was to assist in security investigations and system updates by extracting relevant login and employee data. I applied logical operators like `AND`, `OR`, `NOT`, and `LIKE` to create precise filters across different conditions. This helped simulate real-world scenarios a security analyst would face while analyzing suspicious activity and managing IT operations.

Retrieve after hours failed login attempts

To identify potentially suspicious behavior, I retrieved all failed login attempts (`success = 0`) made after business hours, defined as after 18:00.

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

Result: 19 failed login attempts occurred after 18:00.

Retrieve login attempts on specific dates

I filtered login attempts that occurred on two specific dates being investigated: 2022-05-08 and 2022-05-09.

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

Result: 75 login attempts were made on these two days.

Retrieve login attempts outside of Mexico

To examine logins from outside Mexico, I used a `NOT LIKE` pattern to exclude all entries that started with `'MEX'` (matching both 'MEX' and 'MEXICO').

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

Result: 144 login attempts were made outside of Mexico.

Retrieve employees in Marketing

I retrieved records of all employees in the Marketing department who are located in offices within the East building (e.g., East-170, East-320).

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

Username of the first employee in the result: **elarson**

Retrieve employees in Finance or Sales

To support system updates for specific departments, I queried for all employees who belong to either the Finance or the Sales department.

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

Username of the first employee in the Sales department: **lrodiqu**

Retrieve all employees not in IT

To find all employees who were not part of the Information Technology department (already updated), I used a `NOT` condition.

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

Result: 165 employees are not in the Information Technology department.

Summary

This lab activity helped me strengthen my SQL skills in filtering and data retrieval for cybersecurity purposes. By using conditional operators like `AND`, `OR`, `NOT`, and `LIKE`, I was able to extract relevant login records and employee data to assist in investigations and system updates. These are valuable skills for any security analyst working with large data sets in a real-world environment.