Apply filters to SQL queries

Project description

You are a security professional at a large organization. Part of your job is to investigate security issues to help keep the system secure. You recently discovered some potential security issues that involve login attempts and employee machines.
Your task is to examine the organization's data in their employees and log_in_attempts tables. You'll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues.

Retrieve after hours failed login attempts

You recently discovered a potential security incident that occurred after business hours. To investigate this, you need to query the log_in_attempts table and review after hours login activity. Use filters in SQL to create a query that identifies all failed login attempts that occurred after 18:00. (The time of the login attempt is found in the login_time column. The success column contains a value of 0 when a login attempt failed; you can use either a value of 0 or FALSE in your query to identify failed login attempts.)

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

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. To investigate this event, you want to review all login attempts which occurred on this day and the day before. Use filters in SQL to create a query that identifies all login attempts that occurred on 2022-05-09 or 2022-05-08. (The date of the login attempt is found in the login_date column.)

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

Retrieve login attempts outside of Mexico

There's been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. Now, you need to investigate login attempts that occurred outside of Mexico. Use filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico. (When referring to Mexico, the country column contains values of

both MEX and MEXICO, and you need to use the LIKE keyword with % to make sure your query reflects this.)

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

Retrieve employees in Marketing

Your team wants to perform security updates on specific employee machines in the Marketing department. You're responsible for getting information on these employee machines and will need to query the employees table. Use filters in SQL to create a query that identifies all employees in the Marketing department for all offices in the East building.
(The department of the employee is found in the department column, which contains values that include Marketing. The office is found in the office column. Some examples of values in this column are East-170, East-320, and North-434. You'll need to use the LIKE keyword with % to filter for the East building.)

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

Retrieve employees in Finance or Sales

Your team now needs to perform a different security update on machines for employees in the Sales and Finance departments. Use filters in SQL to create a query that identifies all employees in the Sales or Finance departments. (The department of the employee is found in the department column, which contains values that include Sales and Finance.)

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

Retrieve all employees not in IT

Your team needs to make one more update to employee machines. The employees who are in the Information Technology department already had this update, but employees in all other departments need it. Use filters in SQL to create a query which identifies all employees not in the IT department. (The department of the employee is found in the department column, which contains values that include Information Technology.)

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

Summary

This document outlines several SQL queries designed to investigate potential security issues within an organization's database. The tasks include:

- **Retrieving After Hours Failed Login Attempts:** A query to identify login attempts that failed after 18:00, using the log_in Insteads table, where login_time and success columns are key for filtering.
- **Investigating Specific Dates:** Queries to fetch all login attempts from 2022-05-08 and 2022-05-09 to investigate a suspicious event, using the login_date column.
- **Filtering by Location:** A query to find login attempts outside Mexico, accounting for variations in country codes like 'MEX' and 'MEXICO' in the country column.
- **Department-Specific Queries:**
    - Query to list all employees in the Marketing department located in the East building, using the department and office columns.
    - Query to retrieve employees from both Sales and Finance departments.
    - Query to find all employees not in the Information Technology department, ensuring security updates are applied where needed.

These queries demonstrate the use of SQL filtering techniques like WHERE, AND, OR, and LIKE to narrow down data for security analysis. This approach helps in pinpointing potential security breaches or ensuring specific security protocols are applied across relevant employee groups.