Apply filters to SQL queries

Project description

I investigated security issues to help the system secure. I used SQL filters to find conditions to observe some potential security issues that involve login attempts and employee machines by examining the employees and log_in_attempts table.

Retrieve after hours failed login attempts

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
```

I recently discovered a potential security incident that occurred after business hours (18:00) of login activity. To investigate this further, I applied `SELECT *` so that I can have all the columns return to me. Next, I then need to add `FROM log_in_atempts` to specify which table I will view from, which is called log_in_attempts. Lastly, to filter out conditions where I only look at datas where login time is after 18:00 and the success attempt failed, I would type `WHERE login_time > '18:00' AND success = FALSE;` or I could replace FALSE with 0.

Retrieve login attempts on specific dates

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------+
```

I want to investigate a suspicious event that occurred on 2022-05-09. In order to do this, I must review all login attempts which occurred on this day and the day before (2022-05-09). I `SELECT *` so that I can have all the columns return to me. Next, I then need to add `FROM log_in_atempts` to specify which table I will view from, which is called log_in_attempts. Lastly, I want to select datas satisfy either conditions using `WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';`

Retrieve login attempts outside of Mexico

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE NOT country LIKE 'MEX%';
```

There's been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. In order to investigate login attempts that occurred outside of Mexico, I would use `WHERE NOT country LIKE 'MEX%';` . This 'NOT' negates the condition and 'LIKE' searches for a pattern that goes with and 'MEX%' is looking for anything that matches MEX or includes any characters after MEX which is Mexico.

Retrieve employees in Marketing

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
```

The team wants to perform security updates on specific employee machines in the Marketing department. I want to return all columns from the employees table and filter conditions where only employees are in the East building from the Marketing department. To return all the columns from the employees table. I would do the following

`> SELECT *`
`> FROM employees`

To filter in the conditions `WHERE department = 'Marketing' AND office LIKE 'East%';`

- AND satisfies both conditions
- LIKE satisfies patterns that goes with WHERE
- 'East%' has % as a placeholder to include any amount of strings/characters
    - Example: 'East-170', 'East-320'

Retrieve employees in Finance or Sales

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
```

My team's needs to perform a different security update on machines for employees in the Sales and Finance departments. To do this I would `> SELECT * FROM employees` where I could return all columns from the employees table. To apply the filters for employees in the

Sales and Finance department, I would include “OR” to include both employees from Finance and sales departments.

Retrieve all employees not in IT

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE NOT department = 'Information Technology';
```

My team needs to make one more update to employee machines. It was given that the employees who are in the Information Technology department already had this update, but other departments need it. My queries goes to as follows `SELECT * FROM employees` and I would include `WHERE NOT department = ‘Information Technology’;` this NOT represents to not include any employees who are in the Information Technology department.

Summary

For login attempts, I investigated the potential security issues by reviewing login attempts after hours, Mexico, and on specified dates. Attempts after hours would be suspicious because employees would be off work after specified hours, log in attempts from Mexico would be suspicious since I would assume that their offices aren’t located in Mexico, and login-attempts from specified dates would be suspicious because employees don’t generally work holidays and weekends.

For employees' machines, I took a look to see which criteria I should perform security updates on. Applying filters which would be much more convenient and reduce human error because I was able to see all the data to conduct security updates to.