Apply filters to SQL queries

Project description

I was tasked to make a system more secure by investigating potential security issues and updating employee computers as needed.

Retrieve after hours failed login attempts

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_time > '18:00' AND success = 0;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12  |       0 |
|       18 | pwashing | 2022-05-11 | 19:28:50   | US      | 192.168.66.142  |       0 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.109.50  |       0 |
```

I was tasked to investigate a series of failed login attempts that were made after business hours (18:00). I created a SQL query to filter all of the data by first, selecting from `log_in_attempts` table, then using the WHERE clause and AND operator to filter. `login_time > '18:00'` filters all of the login times that occurred after 18:00. `success = 0` filters for failed login attempts.

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';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       1 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       1 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
```

I was tasked to investigate a suspicious event that occurred on a specific date (2022-05-09). It was also necessary to investigate the day before (2022-05-08). I created a SQL query to filter all of the login attempts that took place on 2022-05-08 or 2022-05-09. First, I selected all data from the `log_in_attempts table`. Then, I used the `WHERE` clause and `OR` operator to filter the results. `login_date = '2022-05-09'` filters login attempts that happened on 2022-05-09. `login_date = '2022-05-08'` filters for the other data, 2022-05-08.

Retrieve login attempts outside of Mexico

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE NOT country LIKE 'Mex%';
+----------+----------+------------+------------+---------+----------------+---------+
| event_id | username | login_date | login_time | country | ip_address     | success |
+----------+----------+------------+------------+---------+----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       1 |
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12  |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       1 |
```

I was tasked to investigate login attempts that occurred outside of Mexico. I created a SQL query to filter out all login attempts that are labeled as a country remotely close to the way Mexico is written, like Mex. First, I selected all data from the `log_in_attempts table`. Then, I used the `WHERE` clause with `NOT` to filter for countries that are not Mexico. I used the `LIKE` with `MEX%` in order to account for the database representing Mexico as both Mexico and Mex. The percentage sign (%) represents any unspecified characters.

Retrieve employees in Marketing

```
MariaDB [organization]> SELECT *
    ->
    -> FROM employees
    ->
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+--------------+----------+------------+----------+
| employee_id | device_id    | username | department | office   |
+-------------+--------------+----------+------------+----------+
|        1000 | a320b137c219 | elarson  | Marketing  | East-170 |
|        1052 | a192b174c940 | jdarosa  | Marketing  | East-195 |
|        1075 | x573y883z772 | fbautist | Marketing  | East-267 |
```

I was tasked to update computers for employees who are in the marketing department and located in an East building. First, I selected all data from the `employees` table. Then, I used the `WHERE` clause along with the `AND` operator to filter employees who work in the Marketing department and office being located in an East building. I used `department = 'Marketing'` to filter employees who work in Marketing. I also used office `LIKE 'East%'` to filter the database that represents any office that relates to the east side. The percentage sign (%) represents any unspecified characters.

Retrieve employees in Finance or Sales

```
MariaDB [organization]> SELECT *
    ->
    -> FROM employees
    ->
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+-------------+----------+------------+-------------+
| employee_id | device_id   | username | department | office      |
+-------------+-------------+----------+------------+-------------+
|        1003 | d394e816f943 | sgilmore | Finance    | South-153   |
|        1007 | h174i497j413 | wjaffrey | Finance    | North-406   |
|        1008 | i858j583k571 | abernard | Finance    | South-170   |
|        1009 | NULL        | lrodriqu | Sales      | South-134   |
```

I was tasked to update computers for employees who are in the Finance or Sales department. I first selected all the data from the `employees` table. Then, I used the `WHERE` clause along with the `OR` operator to filter employees who work in either the Finance or Sales department. Because I used the `OR` operator, I need to filter through the department column two times: `department = 'Finance'` and `department = 'Sales'`.

Retrieve all employees not in IT

```
MariaDB [organization]> SELECT *
    ->
    -> FROM employees
    ->
    -> WHERE NOT department = 'Information Technology';
+-------------+-------------+----------+----------------+-------------+
| employee_id | device_id   | username | department     | office      |
+-------------+-------------+----------+----------------+-------------+
|        1000 | a320b137c219 | elarson  | Marketing      | East-170    |
|        1001 | b239c825d303 | bmoreno  | Marketing      | Central-276 |
|        1002 | c116d593e558 | tshah    | Human Resources | North-434   |
```

I was tasked to make one more security update on employees who are not in the Information Technology department. First, I selected all the data from the `employees` table. Then, I used the `WHERE` clause along with `NOT` to filter out any employee in the Information Technology department.

Summary

I created SQL queries to filter out specific information in the database to find discrepancies or know which computers need updating. I used the tables, `employees` and `log_in_attempts` to get the information and used the `AND`, `OR`, and `NOT` operators to filter them out for each task. For special cases, I used the `LIKE` and percent sign wildcard to filter for patterns.