Apply filters to SQL queries

Project description

As a cybersecurity analyst, it is my job to investigate possible security issues and ensure system safety. I recently discovered some potential security issues. The following steps show how I apply filters with SQL to perform security tasks.

Retrieve after-hours failed login attempts

There were failed login attempts that happened after business hours.

The screenshot demonstrates the SQL query with filters to pull after-hours failed login attempts after `'18:00'`. Following the query are the results of the failed login attempts. I started by selecting all data from the **`log_in_attempts`** table, then applied a **`WHERE`** clause with an **`AND`** operator to filter for unsuccessful login attempts after 6 PM. The first condition, **`login_time > '18:00'`**, captures attempts made after 6 PM, and the second, **`success = FALSE`**, ensures only the failed attempts are included.

```
MariaDB [organization]> clear
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| 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 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.57   |       0 |
|       34 | drosas   | 2022-05-11 | 21:02:04   | US      | 192.168.45.93   |       0 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.157   |       0 |
```

Retrieve login attempts on specific dates

A suspicious event took place on 2022-05-09. All login activity from that day, as well as the day prior, should be thoroughly investigated.
The screenshot below displays my query and then is followed by the output. The first part of the screenshot shows my query, and the second part displays a portion of the output. This query retrieves all login attempts from either 2022-05-09 or 2022-05-08. I started by selecting all data from the log_in_attempts table, then applied a WHERE clause with an OR operator to filter for logins on these two dates. The first condition, login_date = '2022-05-09', captures

logins from May 9th, and the second, login_date = '2022-05-08', captures logins from May 8th

```
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 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
```

Retrieve login attempts outside of Mexico

After investigating data on login attempts, there may be a possible issue with login attempt that happened outside of Mexico.

The screenshot below shows SQL query to filter login attempts that occurred outside of Mexico. I ran the following query below to pull **`log_in_attempts`** that did not come from Mexico using **`WHERE NOT country LIKE 'Mex%'`**. Percent sign (%) signifies any unspecified characters when using **`LIKE.`**

```
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 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
```

Retrieve employees in Marketing

I need to retrieve info from the **`department`** and **`office`** column in the **`employees`** table.I ran the following SQL query to obtain returned values of the **`employees`** table:

```
MariaDB [organization]> SELECT *
    ->
    -> FROM employees;
+-------------+--------------+----------+-----------------------+-------------+
| 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   |
|        1003 | d394e816f943 | sgilmore | Finance               | South-153   |
```

My team and I are updating employee machines. So, we need to obtain the info about employees in the 'Marketing' department who are located in offices in the East building.

I use SQL query by selecting all columns from the **`employees`** table. Then use filters on **`department = 'marketing' and office LIKE 'East%'`** to retrieve the needed records:

```
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 |
|        1088 | k8651965m233 | rgosh    | Marketing  | East-157 |
|        1103 | NULL         | randerss | Marketing  | East-460 |
|        1156 | a184b775c707 | dellery  | Marketing  | East-417 |
|        1163 | h679i515j339 | cwilliam | Marketing  | East-216 |
+-------------+-------------+----------+------------+----------+
7 rows in set (0.023 sec)
```

Retrieve employees in Finance or Sales

We need to perform and update the computers of all the employees in the finance and sales department.
Below is a screenshot of the SQL query that I used to retrieve records of employees from '**finance**' and '**sales**' department.

```
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   |
```

Retrieve all employees not in IT

We need to make one more update. Employee computers in the IT department were already updated. We need to retrieve info on employees that are **NOT** in the IT department.

Screenshot below displays the SQL query that I used to obtain records of employees that are NOT in the IT department. Followed by the returned values

```
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   |
```

Summary

I used filters to SQL queries to obtain specific info on login attempts, employees and departments. I applied logical operators such as AND, NOT, OR. I also used LIKE and % to filter for patterns.