Apply filters to SQL queries

Project description

At my current workplace, we're working hard to improve our system's security. As part of the cybersecurity team, my responsibility includes making sure systems are safe, investigating any security concerns, and updating employees' devices. I've used SQL queries to help with several security tasks by filtering specific data we needed. Here's what I did:

Retrieve after hours failed login attempts

We noticed some unusual activity after regular work hours (after 6 PM). To help with the investigation, I wrote a SQL query to find all failed login attempts that happened after 18:00.

```
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 |
```

In the screenshot, the first part is my SQL query and the second part shows the output. I selected everything from the log_in_attempts table, then used a WHERE clause with an AND condition to narrow it down. One condition checked if the login happened after 18:00 (login_time > '18:00') and the other looked for failed logins (success = FALSE).

Retrieve login attempts on specific dates

A suspicious login happened on May 9, 2022, so I also checked all login activity from that day and the day before.

```
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 |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       0 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
```

To do this, I used a SQL query that filters results for those two dates: login_date = '2022-05-09' or login_date = '2022-05-08'. I again selected all data from the log_in_attempts table and used the WHERE clause with an OR operator to pull records from both dates.

Retrieve login attempts outside of Mexico

We discovered some logins might have come from outside Mexico, which could be a red flag. I created a query to pull all login attempts from countries other than 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 |       0 |
|        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 |       0 |
```

In the code, I used NOT LIKE 'MEX%' to find entries that don't start with "MEX" or "MEXICO". The % symbol works as a wildcard to match anything after "MEX".

Retrieve employees in Marketing

My team needed to update systems for people working in Marketing in the East building. I wrote a SQL query to get those specific employees.

I filtered the data from the employees table using WHERE department = 'Marketing' AND office LIKE 'East%'. This made sure we only got Marketing employees working in East offices.

```
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 |
```

Retrieve employees in Finance or Sales

We also needed to do a separate update for staff in either the Finance or Sales departments. I wrote a query to get just those employees.

I used WHERE department = 'Finance' OR department = 'Sales' to get everyone from either of those two departments. The OR condition ensures both are included.

```
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   |
```

Retrieve all employees not in IT

Finally, we had one more update to do — this time for employees who are not part of the IT department. I used a query with a NOT condition: WHERE department != 'Information Technology'.

```
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

In this project, I used SQL to filter data in two tables: log_in_attempts and employees. I applied filters using AND, OR, and NOT to find the exact records I needed for each task. I also used LIKE and % to match patterns. These filtering skills helped me perform real tasks related to security investigations and system updates.