Apply filters to SQL queries

Project description

My organization is focused on keeping its systems secure. As part of my role, I investigated suspicious login activity and potential vulnerabilities in employee machines. Using SQL queries with filters on the *employees* and *log_in_attempts* tables, I retrieved targeted records to identify unusual activity and determine which devices needed security updates.

Retrieve after hours failed login attempts

A possible security incident occurred after business hours (after 18:00). All failed login attempts during this time required investigation.

The screenshot shows the SQL query used to filter for these events:

```
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
```

The first part of the screenshot is the query, and the second part is a portion of the output. This query returns login attempts that occurred after 18:00 and were unsuccessful. First, all data from the *log_in_attempts* table was selected. Then, a `WHERE` clause with an `AND` operator was applied:

- login_time > '18:00' filters for login attempts after 6:00 PM.
- success = FALSE filters for failed login attempts.

From the output, there are 19 failed login attempts that occurred after 18:00. These records indicate potential suspicious activity that needs further investigation.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or on the day before needs to be investigated.

Retrieve login attempts outside of Mexico

During the investigation, I found potential issues with login attempts that occurred in countries other than Mexico. These attempts need to be reviewed further.

The following screenshot shows the SQL query I used to filter for login attempts outside of Mexico:

```
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
```

The first part of the screenshot is the query, and the second part is a portion of the output. Although this query is filtering by specific dates (2022-05-09 and 2022-05-08), the results reveal login attempts from countries **other than Mexico**, which are of interest for this investigation.

From the output, there are 144 login attempts made outside of Mexico.

Retrieve employees in Marketing

The following code demonstrates how I created a SQL query to filter for login attempts that occurred 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
```

The first part of the screenshot is my query, and the second part is a portion of the output. This query returns all login attempts that occurred on 2022-05-09 or 2022-05-08. First, I started by selecting all data from the *log_in_attempts* table. Then, I used a WHERE clause with an OR operator to filter my results to output only login attempts that occurred on either 2022-05-09 or 2022-05-08. The first condition is login_date = '2022-05-09', which filters for logins on 2022-05-09. The second condition is login_date = '2022-05-08', which filters for logins on 2022-05-08.

Retrieve employees in Finance or Sales

he machines for employees in the Finance and Sales departments also need to be updated. Since a different security update is needed, I have to get information on employees only from these two departments.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Finance or Sales departments.

```
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   |
```

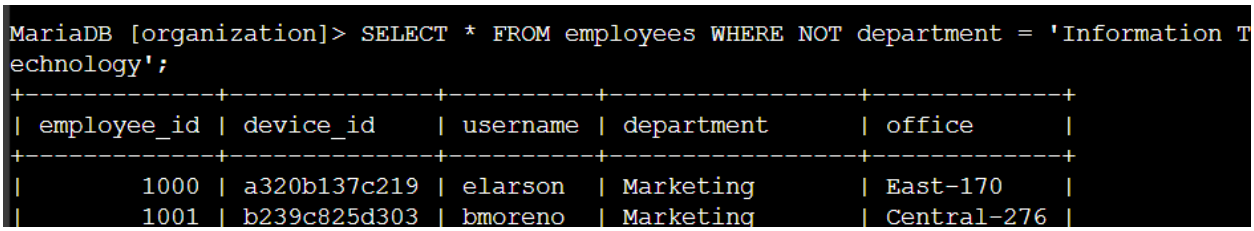
The first part of the screenshot is my query, and the second part is a portion of the output. This query returns all employees in the Finance and Sales departments. First, I started by selecting all data from the *employees* table. Then, I used a WHERE clause with OR to filter for employees who are in the Finance and Sales departments. I used the OR operator instead of AND because I want all employees who are in either department. The first condition is department = 'Finance', which filters for employees

from the Finance department. The second condition is department = 'Sales', which filters for employees from the Sales department.

Retrieve all employees not in IT

My team needs to make one more security update on employees who are not in the Information Technology department. To make the update, I first have to get information on these employees.

The following demonstrates how I created a SQL query to filter for employee machines from employees not in the Information Technology department. The first part of the screenshot is my query, and the second part is a portion of the output. The query returns all employees not in the Information Technology department. First, I started by selecting all data from the *employees* table. Then, I used a WHERE clause with NOT to filter for employees not in this department.



```
MariaDB [organization]> SELECT * FROM employees WHERE NOT department = 'Information T
echnology';
+-------------+--------------+----------+-----------------+-------------+
| employee_id | device_id    | username | department      | office      |
+-------------+--------------+----------+-----------------+-------------+
|        1000 | a320b137c219 | elarson  | Marketing       | East-170    |
|        1001 | b239c825d303 | bmoreno  | Marketing       | Central-276 |
```

Summary

I used SQL filters to retrieve targeted information about login attempts and employee machines from the *log_in_attempts* and *employees* tables. For different tasks, I applied the AND, OR, and NOT operators to narrow down the results. I also used the LIKE operator along with the percentage sign (%) wildcard to search for specific patterns.