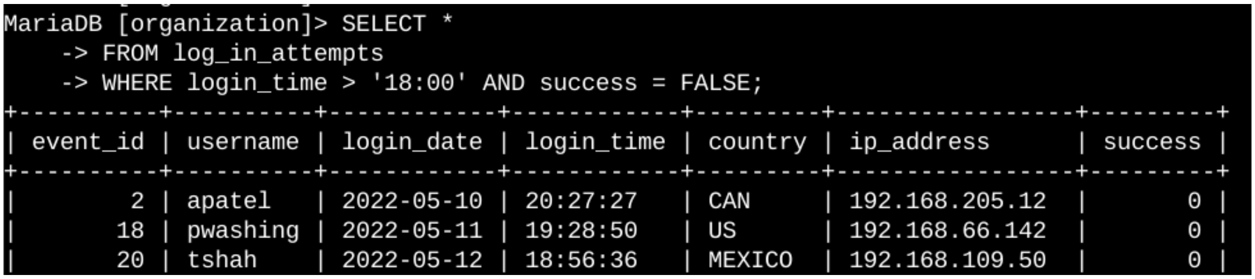
Apply filters to SQL queries

Project description

I investigate security issues to keep our system safe. I found suspicious login activity, so I'm checking the employees and log_in_attempts tables using SQL filters to dig deeper.

Retrieve after hours failed login attempts

A possible security issue happened outside regular working hours (after 6:00 PM). I need to look into all failed login attempts during that time. Here's the SQL query I used to filter those specific records.



```
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 |
```

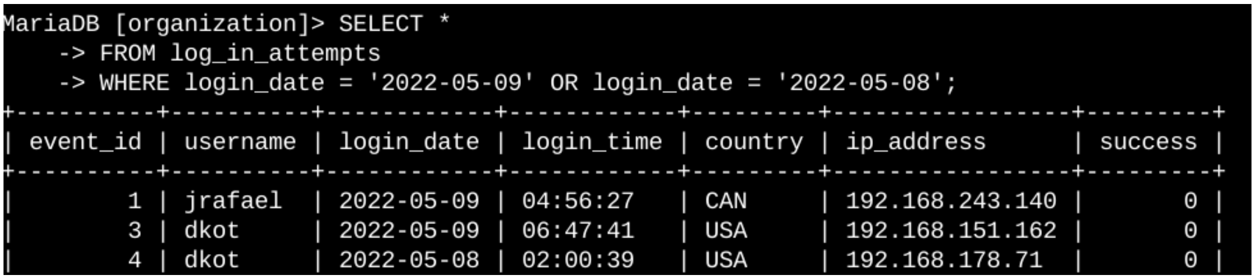
The screenshot shows my SQL query and part of the resulting data. I began by selecting all records from the log_in_attempts table. Then, I added a WHERE clause with two conditions using AND:

- login_time > '18:00' to capture attempts after business hours
- success = FALSE to isolate failed logins

This query helps me focus on potentially suspicious activity that happened after 6 PM.

Retrieve login attempts on specific dates

A suspicious event took place on 2022-05-09, so I need to review login activity from that day and the day before. To do this, I wrote a SQL query that filters for login attempts on 2022-05-08 and 2022-05-09. Here's how I set it up to target those 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 |       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 |
```

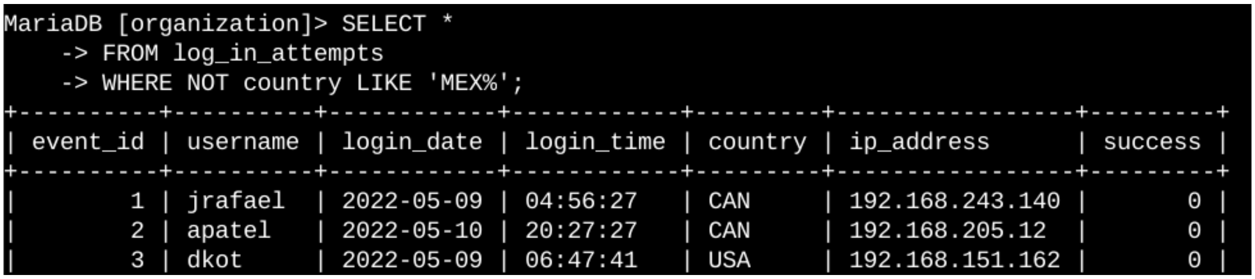
The screenshot shows my SQL query and part of the output. I began by selecting all records from the log_in_attempts table. Then, I added a WHERE clause with an OR operator to filter for login attempts that happened on either 2022-05-09 or 2022-05-08. Specifically, I used:

- login_date = '2022-05-09' to capture logins on May 9
- login_date = '2022-05-08' to include logins from the day before

This helps me focus on activity around the time of the suspicious event.

Retrieve login attempts outside of Mexico

I found a potential issue with login attempts that happened outside of Mexico. These entries need further investigation. To identify them, I wrote a SQL query that filters for login attempts where the location is not Mexico. Here's how I set it up.



```
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 |
```

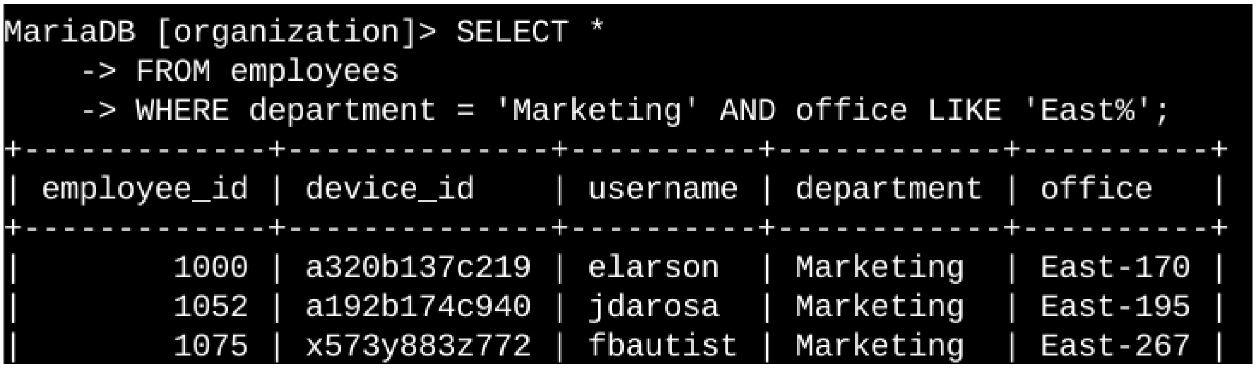
The screenshot shows my SQL query and part of the output. I began by selecting all records from the log_in_attempts table. Then, I added a WHERE clause using NOT LIKE 'MEX%' to exclude entries from Mexico. This pattern filters out both "MEX" and "MEXICO", since % matches any characters that follow. The result displays login attempts from other countries.

Retrieve employees in Marketing

The screenshot includes my SQL query and part of the output. I started by selecting all records from the employees table. Then, I used a WHERE clause with two conditions to filter for employees in the Marketing department located in the East building. Specifically:

- department = 'Marketing' targets the right team
- building = 'East' narrows it down to their location

This query helps identify which employee machines need updating.



```
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 |
```

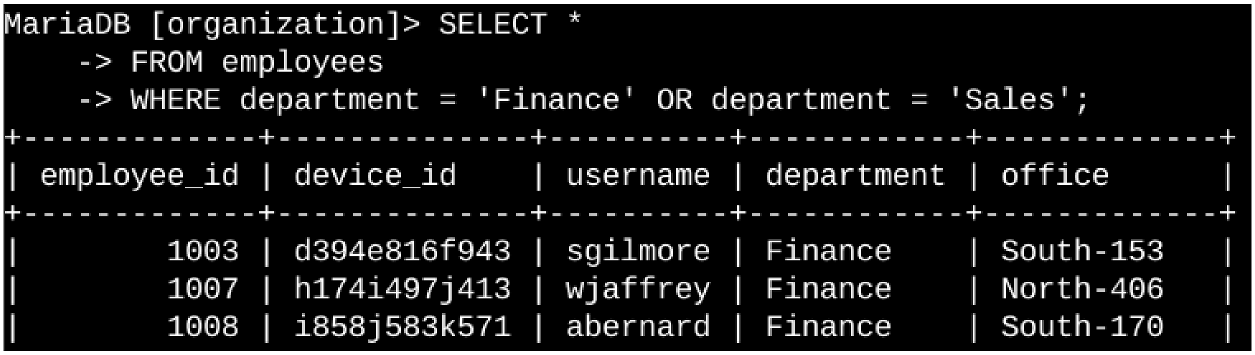
The screenshot shows my SQL query and part of the output. I began by selecting all records from the employees table. Then, I added a WHERE clause with an AND operator to filter for employees in the Marketing department who are located in the East building. I used department = 'Marketing' to target the right team, and office LIKE 'East%' to match office entries that start with "East", since they include specific room numbers.

Retrieve employees in Finance or Sales

The screenshot shows my SQL query and part of the output. I started by selecting all records from the employees table. Then, I used a WHERE clause with an OR operator to filter for employees in either the Finance or Sales departments. Specifically:

- department = 'Finance' targets Finance staff
- department = 'Sales' includes Sales staff

This query helps identify which machines need the specific security update.

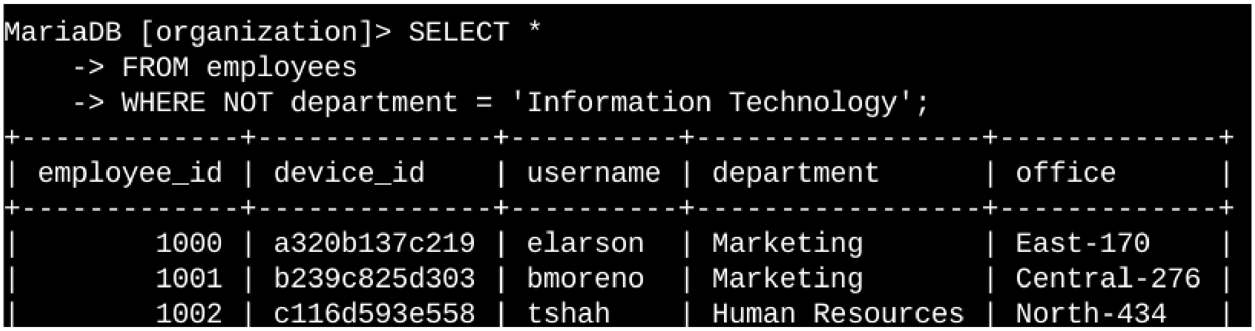


```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+-------------+----------+------------+-------------+
| employee_id | device_id   | username | department | office      |
+-------------+-------------+----------+------------+-------------+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

Retrieve all employees not in IT

The screenshot shows my SQL query and part of the output. I started by selecting all records from the employees table. Then, I used a WHERE clause with != to filter out employees from the

Information Technology department. Specifically, I used department != 'Information Technology' to return only those who are in other departments. This helps identify which machines need the final security update.



```
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 SQL filters to extract targeted data from the log_in_attempts and employees tables. To narrow down results, I applied logical operators like AND, OR, and NOT depending on the task. I also used the LIKE operator with the % wildcard to match specific patterns in the data. This helped me pinpoint login activity and employee machine details efficiently.