Apply filters to SQL queries

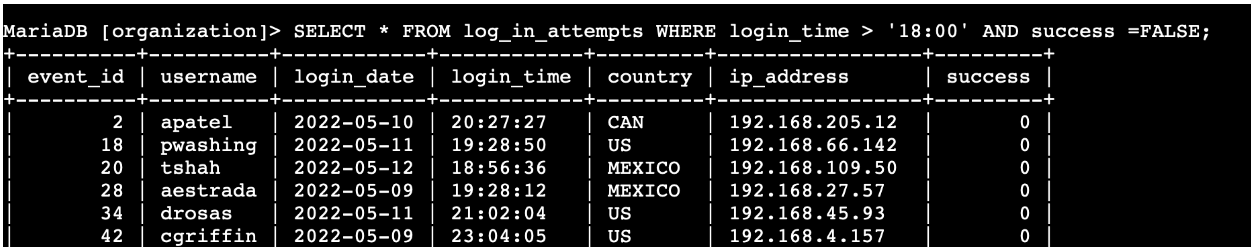
Project description

My team needs to accomplish certain security-related tasks, such as investigating potential security incidents and updating employee machines. Using my familiarity with CLI and knowledge of filtering SQL queries, I can help my team achieve this task.

Retrieve after hours failed login attempts

Following a potential after-hours security breach (post 18:00), all after-hour login attempts needed to be analyzed.

Using a SQL query with a filter, I generated a table of failed after-hour login attempts:



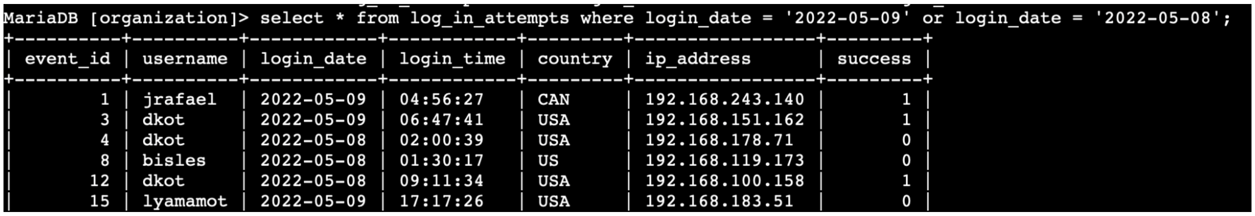
```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_time > '18:00' AND success =FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
```

The first part of the picture is the query, while the second part is part of the output. I started by selecting all data from the `log_in_attempts` table. Then, I used the `WHERE` clause with an `AND` operator to filter to output all unsuccessful login attempts after 18:00, which I filtered for using the `login_time > '18:00'` and `success = FALSE` which respectively produce output such that only log in attempts after 18:00 that are unsuccessful are shown.

Retrieve login attempts on specific dates

A potential suspicious incident occurred on 2022-05-09, which is why any login attempts on or the day before need to be looked into.

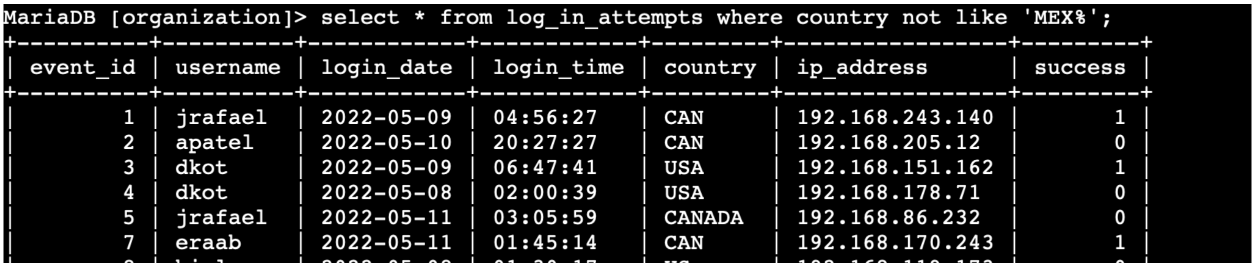
I wrote a SQL query to filter for login attempts that occurred on 2022-05-09 or 2022-05-08



```
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 |
|       12 | dkot     | 2022-05-08 | 09:11:34   | USA     | 192.168.100.158 |       1 |
|       15 | lyamamot | 2022-05-09 | 17:17:26   | USA     | 192.168.183.51  |       0 |
```

Retrieve login attempts outside of Mexico

Upon analyzing data so far, it seems as though the suspicious login activity occurred outside of Mexico, which is what the following SQL query filters for:

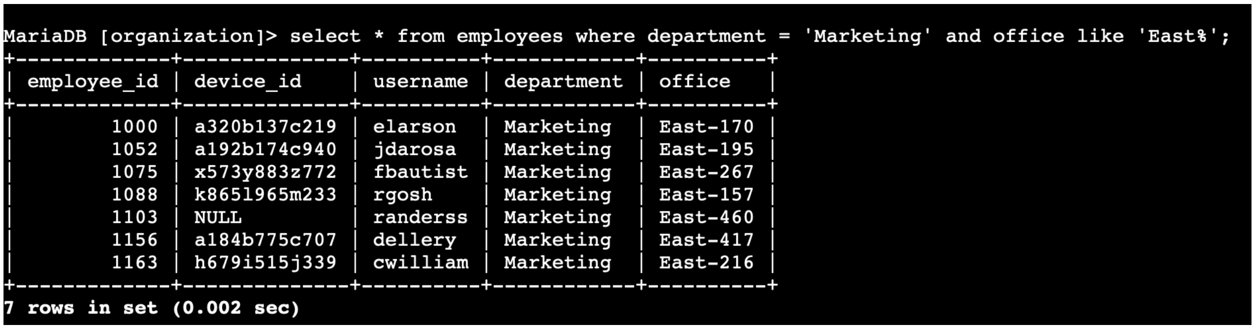


```
MariaDB [organization]> select * from log_in_attempts where country not like 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|        5 | jrafael  | 2022-05-11 | 03:05:59   | CANADA  | 192.168.86.232  |       0 |
|        7 | eraab    | 2022-05-11 | 01:45:14   | CAN     | 192.168.170.243 |       1 |
```

Retrieve employees in Marketing

Some employee computers in the Marketing department need to be updated. To find out which employees need this update, I ran a SQL query.

The following SQL query filters for employee machines from employees in the Marketing department in the East building.

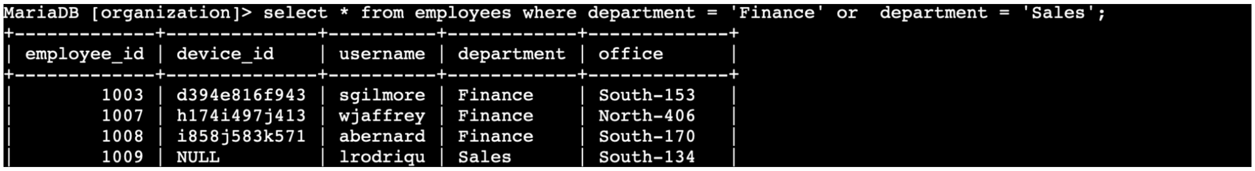


```
MariaDB [organization]> select * from employees where department = 'Marketing' and office like 'East%';
+-------------+--------------+----------+------------+----------+
| employee_id | device_id    | username | department | office   |
+-------------+--------------+----------+------------+----------+
|        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 (0.002 sec)
```

Retrieve employees in Finance or Sales

Employee computers in Finance and Sales also need an update, albeit a different security update. Hence, I needed to run a filtered SQL query to only produce a list of employees from either department.

This SQL query filters for employee machines from either Finance or Sales departments

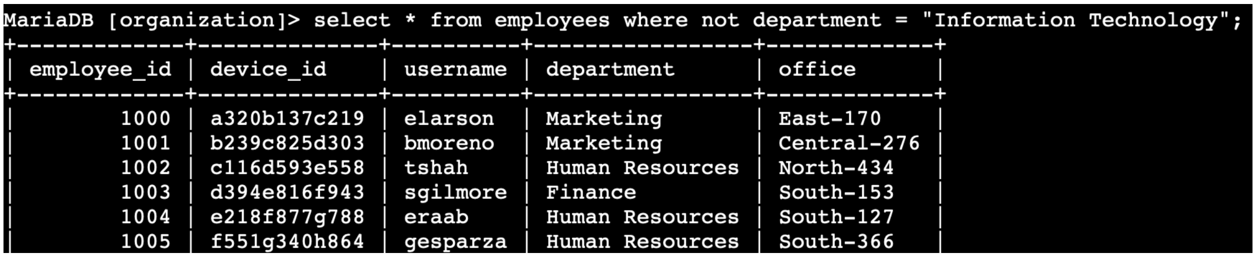


```
MariaDB [organization]> select * from employees where department = 'Finance' or  department = 'Sales';
+-------------+--------------+----------+------------+-------------+
| employee_id | device_id    | username | department | office      |
+-------------+--------------+----------+------------+-------------+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

Retrieve all employees not in IT

My team needs to make one last security update for all non-IT employee machines. To do this, I had to write yet another filtered SQL query to retrieve employee information.

The following SQL query filters for all non-IT employees



```
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   |
|        1003 | d394e816f943 | sgilmore | Finance         | South-153   |
|        1004 | e218f877g788 | eraab    | Human Resources | South-127   |
|        1005 | f551g340h864 | gesparza | Human Resources | South-366   |
```

Summary

I ran filtered SQL queries to retrieve information on login attempts and employee machines. I referred to two tables, namely `log_in_attempts` and `employees`. I also used operators like `AND`, `OR`, and `NOT` to filter my queries. To get more nuanced results by filtering for patterns, I used `LIKE` and the percentage sign (`%`) wildcard.