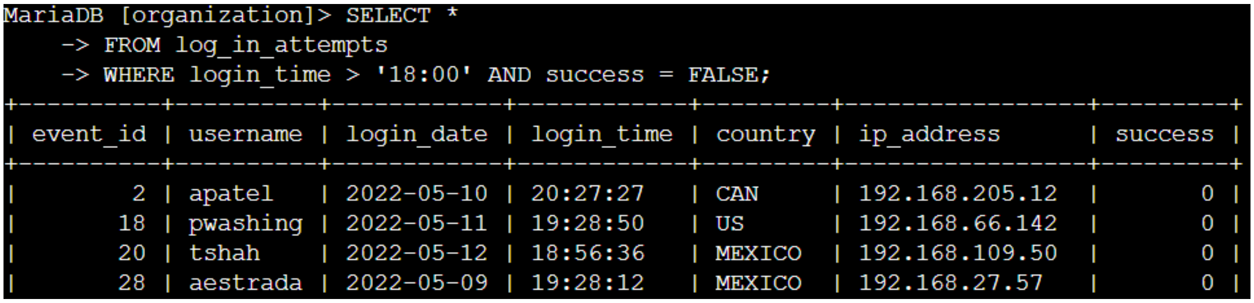
Apply filters to SQL queries

Project description

I was tasked to investigate security issues in my organization and help in keeping the system secure. I discovered some potential security issues that involve login attempts and employee machines. I examined the organizations data in the ‘employees’ and ‘log_in_attempts’ tables using SQL filters to retrieve various records and investigate the potential security issues.

Retrieve after hours failed login attempts

There was a potential security incident that occurred after business hours (after 18:00). All after hours login attempts that failed need to be investigated.

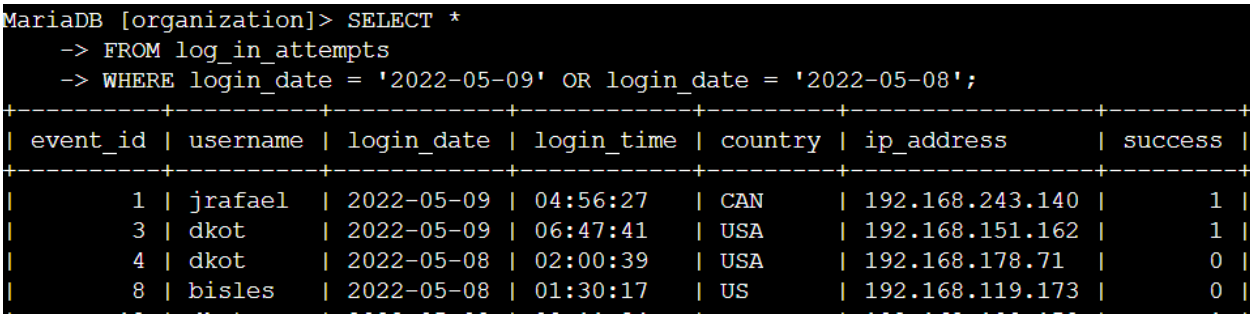


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+---------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

The filter was tailored to output failed login attempts after working hours which is ‘18:00’. Failed login in attempts are indicated by ‘0’. The first condition is login_time > ’18:00’ filters al login attempts after working hours AND the second condition success = FALSE filters for failed attempts.

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.

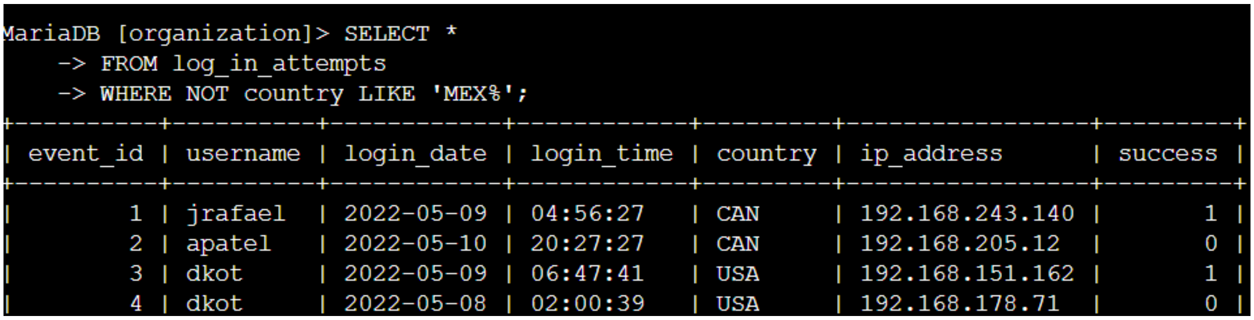


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
```

This filters outputs login activity that happened on ‘2022-05-09’ or before (2022-05-08). The OR operator accomplishes this task. The first condition is login_date = ‘2022-05-09’ which filters for logins on 2022-05-09 while the second condition login_date = ‘2022-05-08’ filters for 2022-05-08.

Retrieve login attempts outside of Mexico

After investigating the organization’s data on login attempts, I believe there is an issue with the login attempts that occurred outside of Mexico. These login attempts should be investigated.

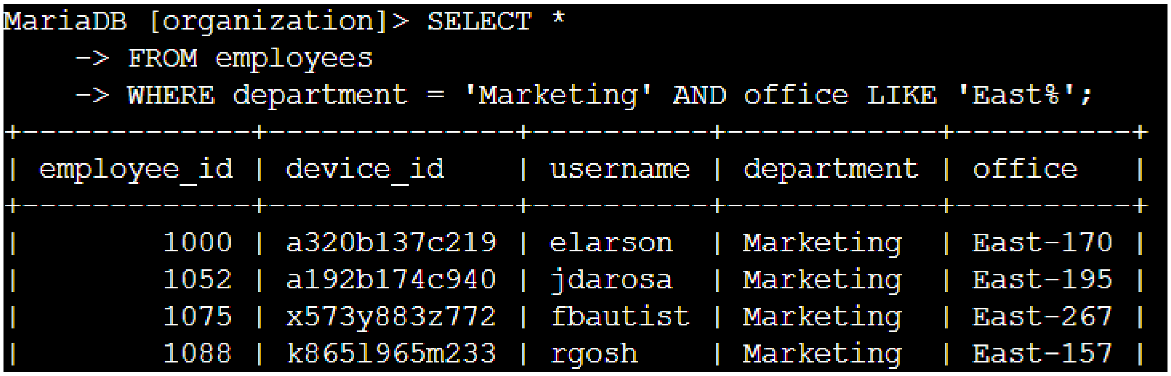


```
ariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
```

This screenshot shows all the logins outside of Mexico. The NOT clause is used to filter for countries other than Mexico and LIKE with MEX% will represent all the dataset that have MEX in the name for the country column which is used to represent Mexico. % is used with LIKE to represent any number of unspecified characters.

Retrieve employees in Marketing

My team wants to update the computers for certain employees in the Marketing department. To do this, I have to get information on which employee machines to update.

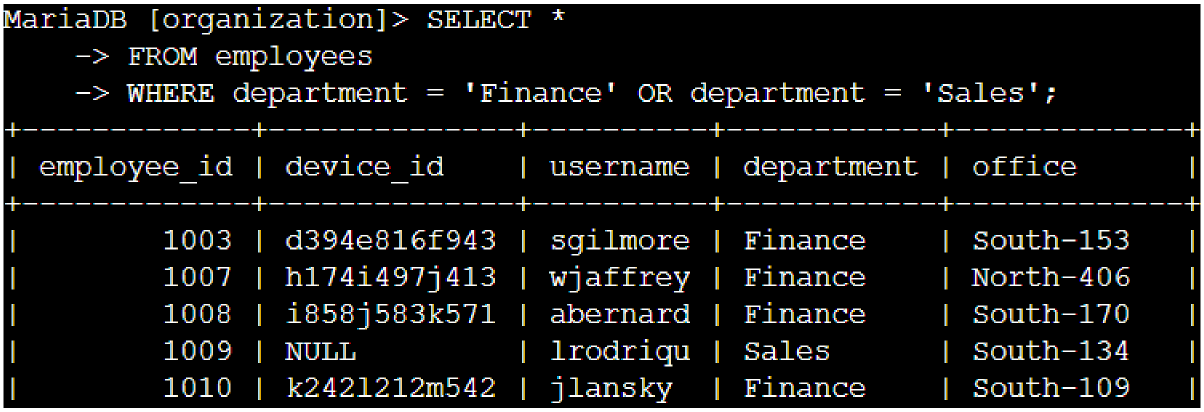


```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+--------------+----------+------------+-----------+
| employee_id | device_id    | username | department | office    |
+-------------+--------------+----------+------------+-----------+
|        1000 | a320b137c219 | elarson  | Marketing  | East-170  |
|        1052 | a192b174c940 | jdarosa  | Marketing  | East-195  |
|        1075 | x573y883z772 | fbautist | Marketing  | East-267  |
|        1088 | k8651965m233 | rgosh    | Marketing  | East-157  |
```

This snippet shows the employees in the Marketing department who are in the East offices. The first condition department = ‘Marketing’ outputs the employees in the Marketing department AND the second condition office LIKE ‘East%’ further details the employees in that department who are in the East offices thus the LIKE clause.

Retrieve employees in Finance or Sales

The 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.

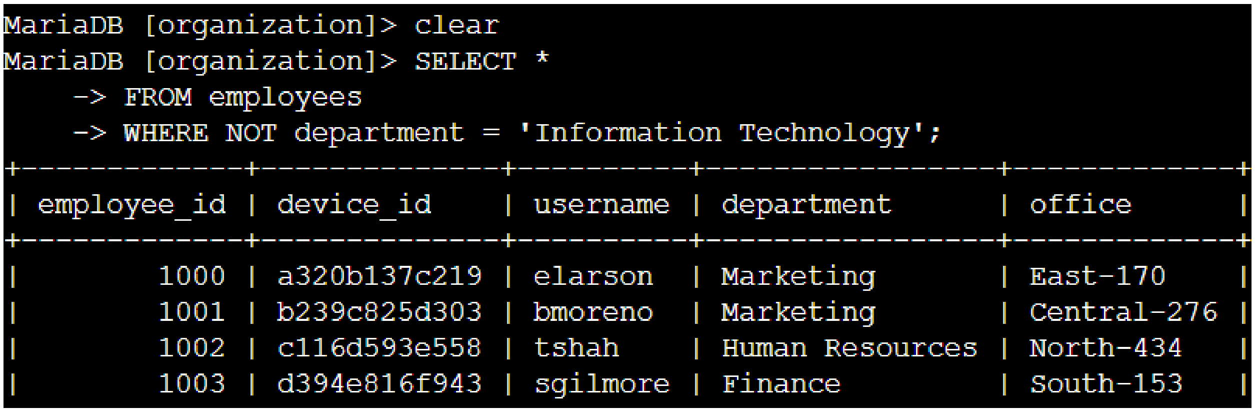


```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+--------------+----------+-------------+-------------+
| employee_id | device_id    | username | department  | office      |
+-------------+--------------+----------+-------------+-------------+
|        1003 | d394e816f943 | sgilmore | Finance     | South-153   |
|        1007 | h174i497j413 | wjaffrey | Finance     | North-406   |
|        1008 | i858j583k571 | abernard | Finance     | South-170   |
|        1009 | NULL         | lrodriqu | Sales       | South-134   |
|        1010 | k2421212m542 | jlansky  | Finance     | South-109   |
```

This illustrates the all the employees in the Finance and Sales departments who need updates. This query returns all employees in the Finance and Sales departments. All the data is collected from the 'employees' table. The OR clause is meant to output employees from either department. The first condition is department = 'Finance' which filters for Finance department employees and the second condition is department = 'Sales' which filters for Sales department employees. Note that each condition has to be filtered separately thus the clause.

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.



```
MariaDB [organization]> clear
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   |
```

This query displays outputs for all the employees that are not in the Information Technology department. The WHERE clause with the NOT accomplishes that with the condition department = 'Information Technology' specifying where not to filter.

Summary

I applied filters to SQL queries to get specific information on login attempts and employee machines. I used two different tables, log_in_attempts and employees. I used the AND, OR and NOT operators to filter for the specific information needed for each task. I also used LIKE and the percentage sign % wildcard to filter for patterns.