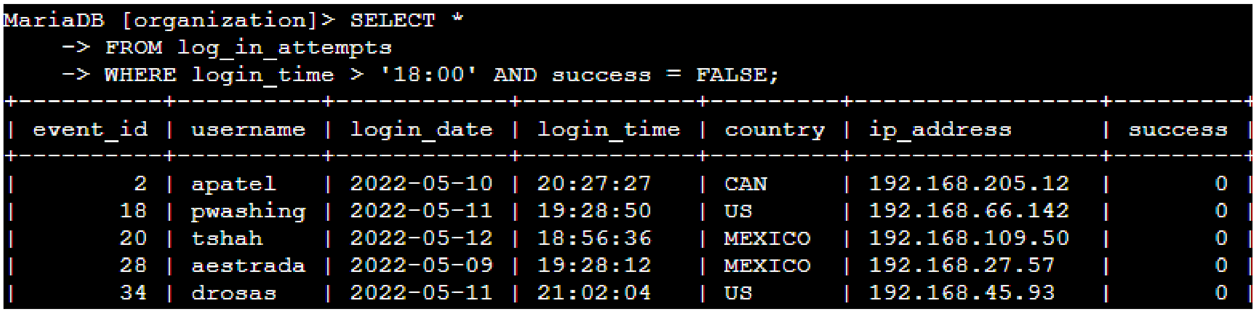
Apply filters to SQL queries

Project description

You are a security professional at a large organization. Part of your job is to investigate security issues to help keep the system secure. You recently discovered some potential security issues that involve login attempts and employee machines.
We will examine the organization's data in their employees and log_in_attempts tables. We will use SQL filters to retrieve records from different datasets and investigate the potential security issues.

Retrieve after hours failed login attempts

You recently discovered a potential security incident that occurred after business hours. I wrote a query that used filters in SQL that identifies all failed login attempts that occurred after 18:00.

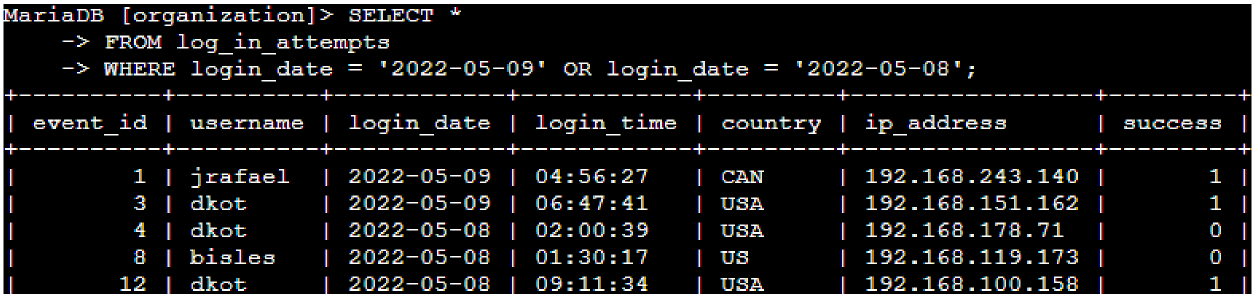


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+---------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

We selected all data from the log_in_attempts table
We used the WHERE and AND commands to filter
The first condition login_time > '18:00' filters login times after 6pm
The second condition AND success = false filters failed login attempts

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. To investigate this event, you want to review all login attempts which occurred on this 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 |       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 |
```

We selected all data from the log_in_attempts table

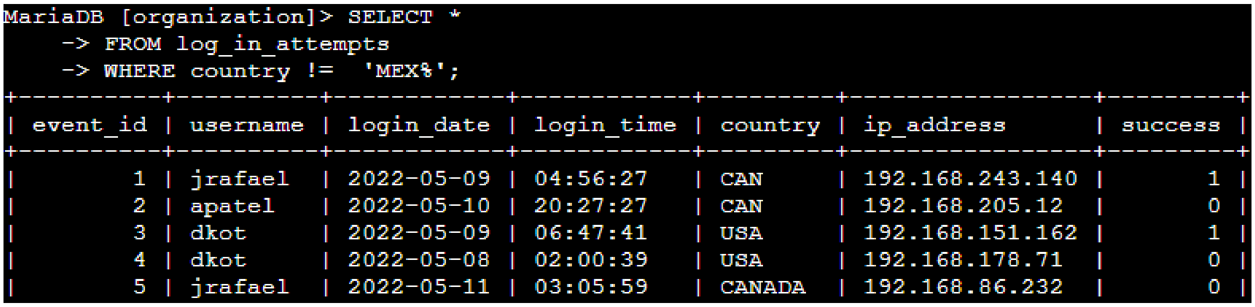
We used the WHERE and OR commands to filter

The first condition login_date '2022-05-09' filters logins on 2022-05-09

The second condition OR '2022-05-08' filters logins on 2022-05-08

Retrieve login attempts outside of Mexico

There's been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. Now, you need to investigate login attempts that occurred outside of Mexico. Use filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico.



```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE country !=  '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 |
```

We selected all data from the log_in_attempts table

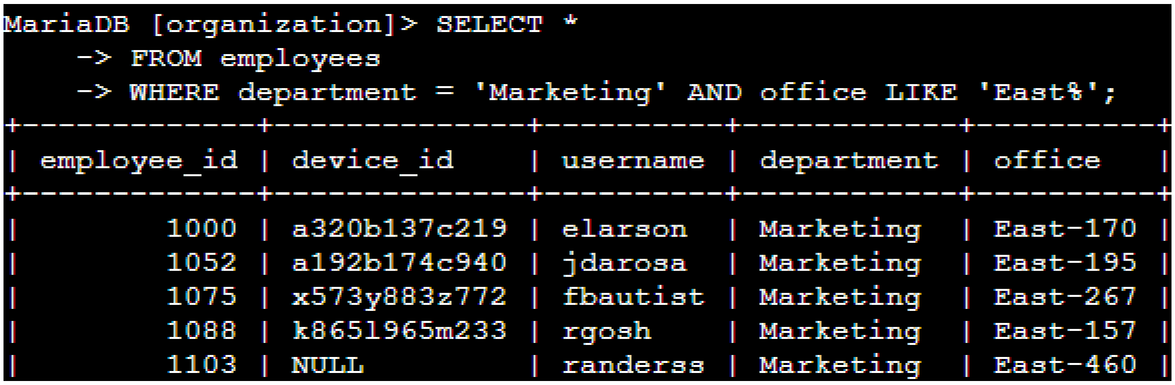
We used the WHERE to filter countries

!= means not equal to

'MEX%' means filter a string that match for MEX and MEXICO. The % substitutes for any number of other characters. % is used as a wildcard.

Retrieve employees in Marketing

Your team wants to perform security updates on specific employee machines in the Marketing department. You're responsible for getting information on these employee machines and will need to query the employees table. Use filters in SQL to create a query that identifies all employees in the Marketing department for all offices 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  |
```

We selected all data from the employees table

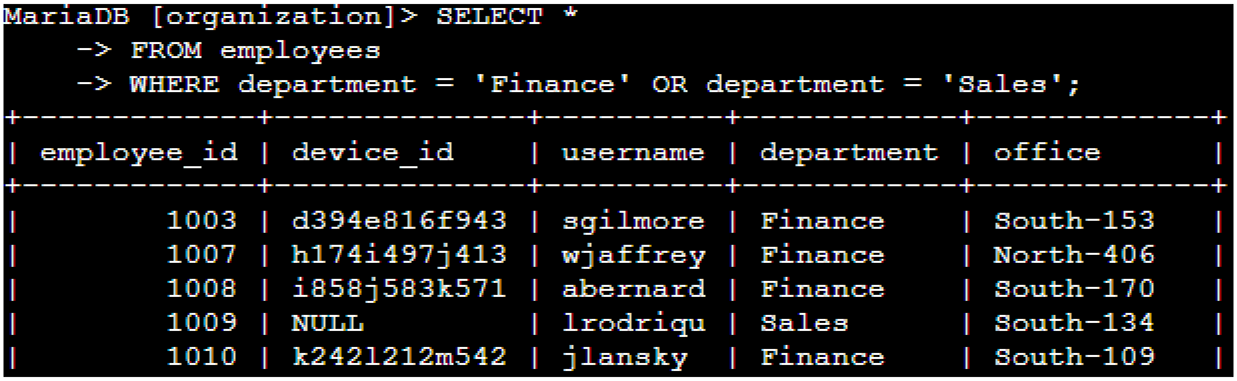
WHERE and AND filters out employees from the Markeing department in the East office

The first condition was department = ‘Marketing’

The second condition was office LIKE ‘East%’ which filters offices in the East office with various numbers

Retrieve employees in Finance or Sales

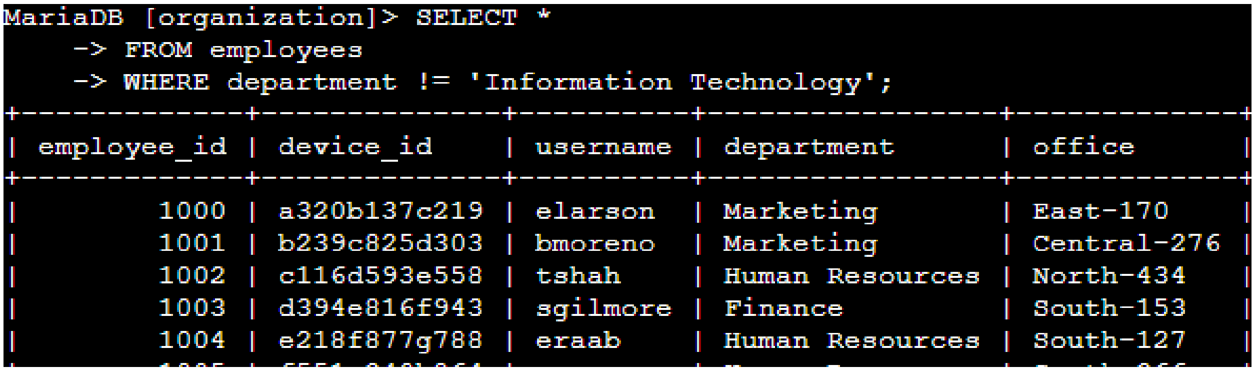
Your team now needs to perform a different security update on machines for employees in the Sales and Finance departments. Use filters in SQL to create a query that identifies all employees in the Sales or Finance departments.



```
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    |
```

Retrieve all employees not in IT

Your team needs to make one more update to employee machines. The employees who are in the Information Technology department already had this update, but employees in all other departments need it. Use filters in SQL to create a query which identifies all employees not in the IT department.



```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE 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    |
```

Summary

I applied filters to SQL queries to get information on login attempts and employees.
I used the login_in_attempts and employees tables.
I used the AND, OR , and != operators while also using LIKE and the % wildcard to filter for patterns.