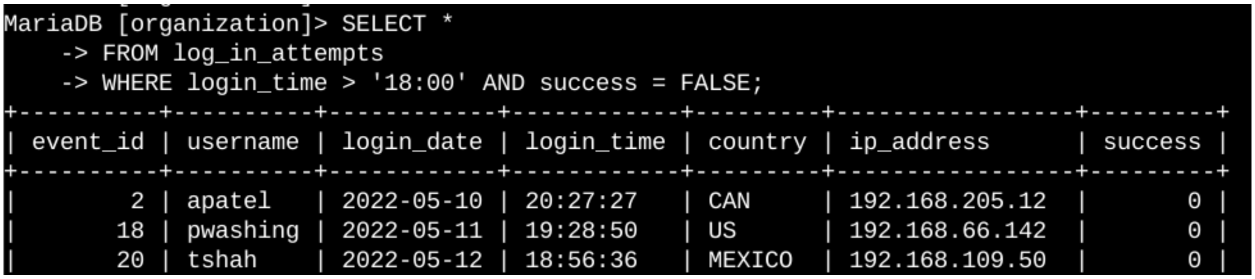
Apply filters to SQL queries

Project description

My organization is working on making our system more secure. My role is to keep the system safe, look into any possible security issues, and update employee computers when needed. The steps below show examples of how I used SQL with filters to carry out security-related tasks.

Retrieve after hours failed login attempts

A potential security incident happened after business hours (after 18:00). Any failed login attempts that occur after hours need to be investigated. The code below shows how I wrote a SQL query to filter for failed login attempts that took place outside of normal business hours.

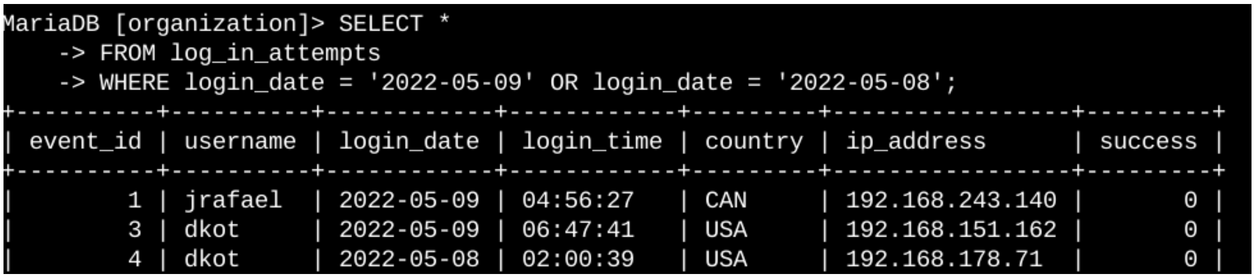


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 first part of the screenshot shows my query, and the second part shows part of the output. This query looks for failed login attempts that happened after 18:00. I started by selecting all data from the `log_in_attempts` table. Then I added a `WHERE` clause with an `AND` operator to narrow the results to only login attempts that happened after 18:00 and were unsuccessful. The first condition, `login_time > '18:00'`, finds login attempts made after 18:00. The second condition, `success = FALSE`, finds the failed login attempts.

Retrieve login attempts on specific dates

A suspicious event took place on 2022-05-09. Any login activity from that date or the previous day needs to be reviewed. The code below shows how I wrote a SQL query to filter login attempts by specific dates.

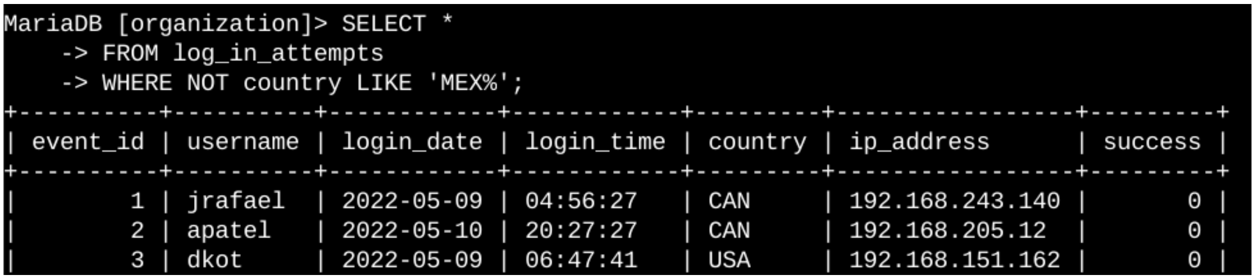


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 first part of the screenshot shows my query, and the second part shows part of the output. This query returns all login attempts from 2022-05-09 or 2022-05-08. I began by selecting all data from the `log_in_attempts` table. Then I added a `WHERE` clause with an `OR` operator to limit the results to logins that happened on either of those two dates. The first condition, `login_date = '2022-05-09'`, filters for logins on 2022-05-09. The second condition, `login_date = '2022-05-08'`, filters for logins on 2022-05-08.

Retrieve login attempts outside of Mexico

After reviewing the organization's login attempt data, I noticed a potential issue with attempts made outside of Mexico. These attempts should be investigated further. The code below shows how I wrote a SQL query to filter for login attempts that originated outside of Mexico.

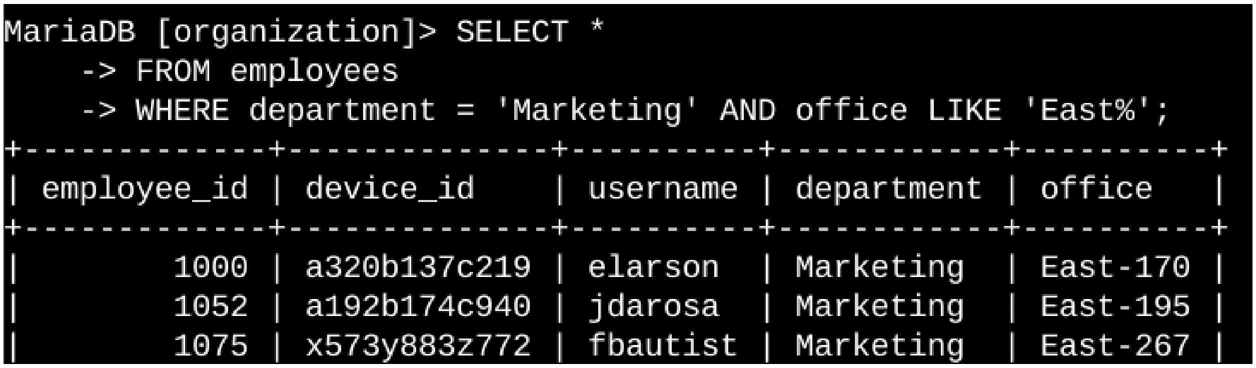


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 first part of the screenshot shows my query, and the second part shows part of the output. This query returns all login attempts that occurred outside of Mexico. I started by selecting all data from the `log_in_attempts` table. Then I added a `WHERE` clause with `NOT` to filter for countries other than Mexico. I used `LIKE 'MEX%'` as the pattern because the dataset represents Mexico as both `MEX` and `MEXICO`. The `%` symbol matches any number of characters when used with `LIKE`.

Retrieve employees in Marketing

My team needs to update the computers for certain employees in the Marketing department. To do this, I first needed to identify which employee machines required updates. The code below shows how I wrote a SQL query to filter for employee machines belonging to Marketing department employees in the East building.

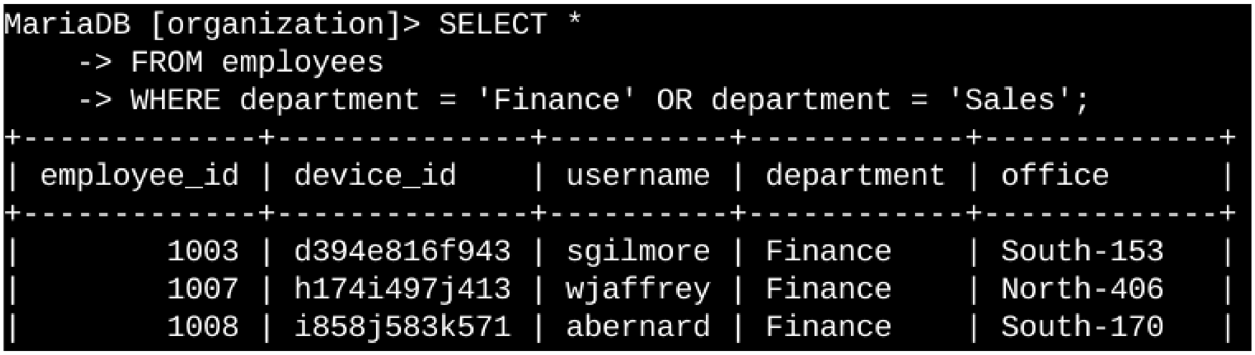


```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+-------------+----------+------------+----------+
| employee_id | device_id   | username | department | office   |
+-------------+-------------+----------+------------+----------+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

The first part of the screenshot shows my query, and the second part shows part of the output. This query returns all employees in the Marketing department who work in the East building. I started by selecting all data from the `employees` table. Then I added a `WHERE` clause with `AND` to filter for employees in the Marketing department **and** in the East building. I used `LIKE 'East%'` because the office column includes specific office numbers within the East building. The first condition, `department = 'Marketing'`, filters for Marketing employees, and the second condition, `office LIKE 'East%'`, filters for employees located in the East building.

Retrieve employees in Finance or Sales

The machines for employees in the Finance and Sales departments also need updates. Because a different security update is required, I needed to identify only the employees in these two departments. The code below shows how I wrote a SQL query to filter for employee machines belonging to employees in the Finance or Sales departments.

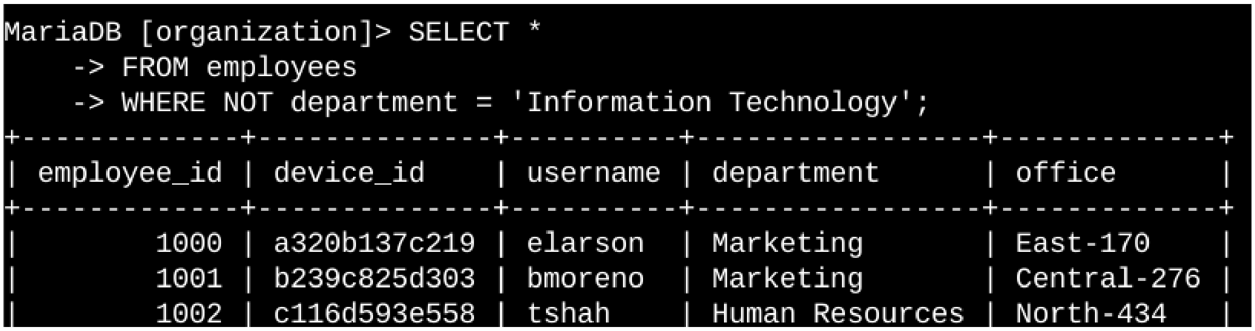


```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+--------------+----------+------------+-------------+
| employee_id | device_id    | username | department | office      |
+-------------+--------------+----------+------------+-------------+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

The first part of the screenshot shows my query, and the second part shows part of the output. This query returns all employees in the Finance and Sales departments. I started by selecting all data from the `employees` table. Then I added a `WHERE` clause with `OR` to filter for employees in either the Finance or Sales departments. I used `OR` instead of `AND` because I wanted to include employees from **either** department. The first condition, `department = 'Finance'`, filters for Finance employees, and the second condition, `department = 'Sales'`, filters for Sales employees.

Retrieve all employees not in IT

My team needs to perform one more security update for employees who are **not** in the Information Technology department. To prepare for this update, I first needed to identify these employees. The code below shows how I wrote a SQL query to filter for employee machines belonging to employees outside the Information Technology department.



```
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   |
```

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.

Summary

I applied filters in SQL queries to retrieve specific information on login attempts and employee machines. I worked with two tables: `log_in_attempts` and `employees`. To narrow down the results, I used the `AND`, `OR`, and `NOT` operators for each task. I also used `LIKE` with the `%` wildcard to filter for specific patterns in the data.