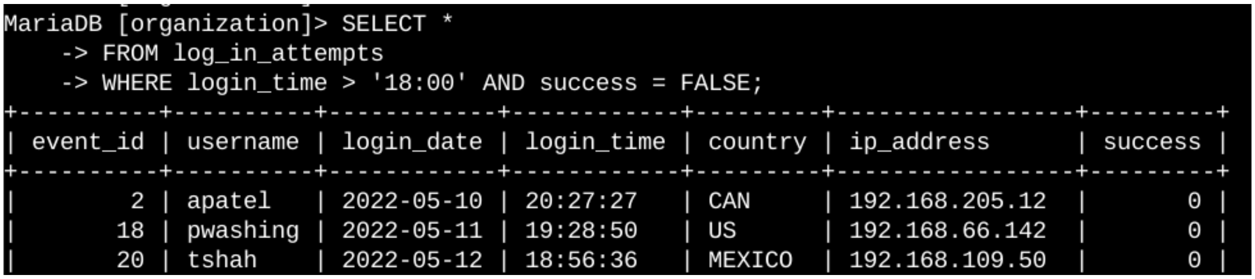
Applying Filters to SQL Queries

Project Overview:
As part of my responsibilities in ensuring the security of our organization's system, I have been tasked with investigating potential security issues and updating employee computers as necessary. Below, I provide examples of how I effectively utilized SQL queries with filters to accomplish these security-related tasks.

Retrieve Failed Login Attempts After Business Hours:
Following a potential security incident that occurred after business hours (post 18:00), it was imperative to investigate all failed login attempts that took place during this time frame.

To address this requirement, I developed the following SQL query to filter and identify failed login attempts that occurred after 18:00:

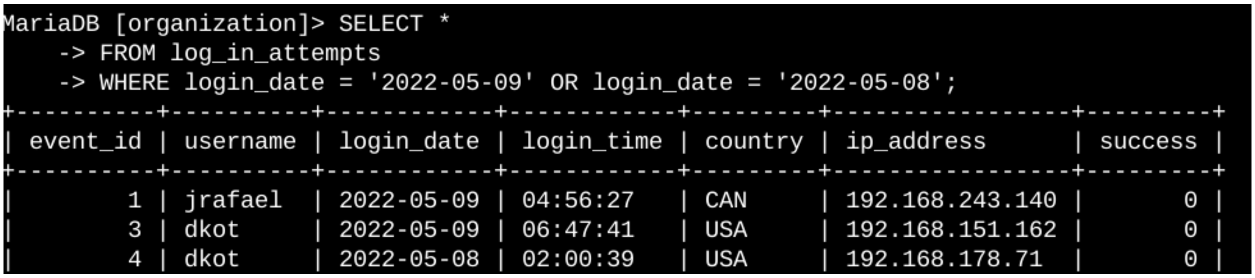


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+---------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

This query retrieves all data from the log_in_attempts table and filters the results to include only unsuccessful login attempts that transpired after 18:00. The first condition, login_time > '18:00', filters for login attempts occurring after 18:00, while the second condition, success = FALSE, filters for failed attempts.

Retrieve Login Attempts on Specific Dates:
A suspicious event took place on 2022-05-09, and it was necessary to investigate any login activity that occurred on that specific date or the day before.

To address this requirement, I designed the following SQL query to filter and retrieve login attempts on the specified dates:

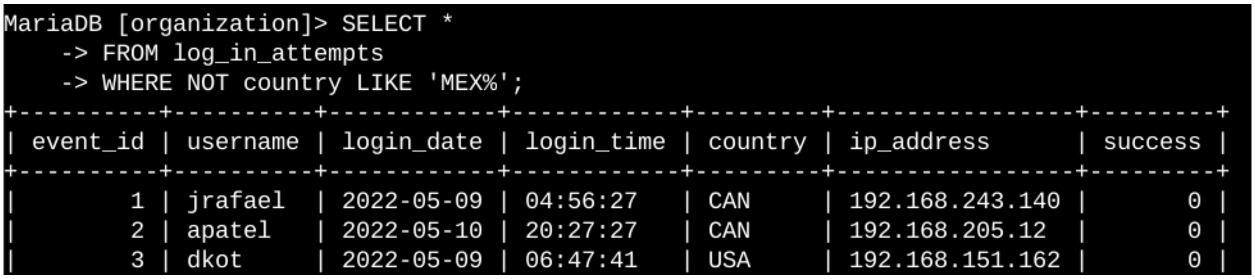


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success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

This query retrieves all data from the log_in_attempts table and filters the results to include login attempts that took place either on 2022-05-09 or 2022-05-08. The first condition, login_date = '2022-05-09', filters for login attempts on 2022-05-09, while the second condition, login_date = '2022-05-08', filters for login attempts on 2022-05-08.

Retrieve Login Attempts Outside of Mexico:
After conducting an analysis of the organization's login attempt data, it became apparent that there might be issues with attempts originating from locations outside of Mexico. It was crucial to investigate these login activities further.

To address this requirement, I formulated the following SQL query to filter and retrieve login attempts outside of Mexico:

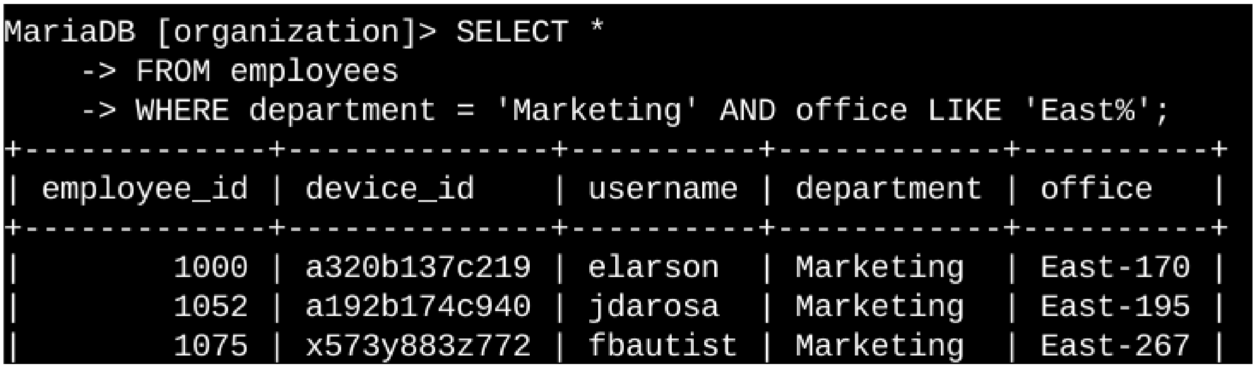


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+--------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

This query retrieves all data from the log_in_attempts table and filters the results to include login attempts from countries other than Mexico. The condition country NOT LIKE 'MEX%' filters for countries that do not match the pattern 'MEX%'. The '%' symbol acts as a wildcard representing any number of unspecified characters when used with the LIKE operator.

Retrieve Employee Machines in the Marketing Department:
Our team aims to update the computers of specific employees working in the Marketing department. To facilitate this process, it was necessary to gather information about the machines used by these employees.

To address this requirement, I devised the following SQL query to filter and retrieve employee machines belonging to the Marketing department:

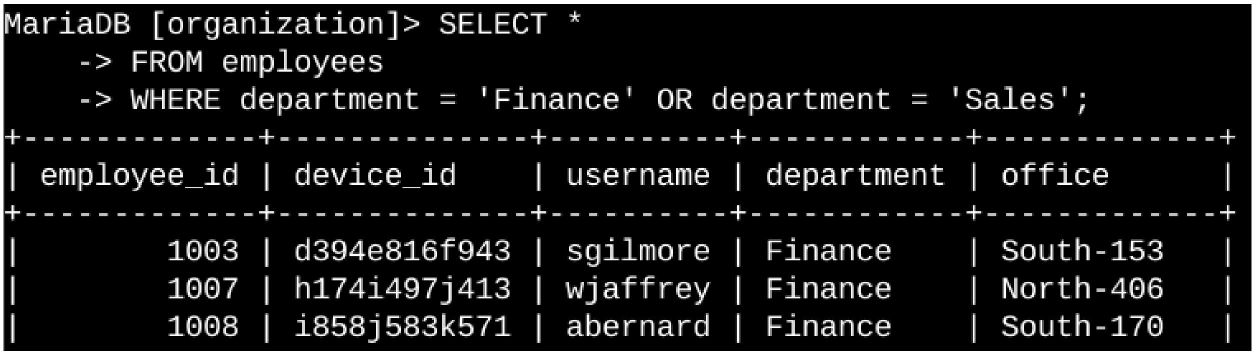


```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+-------------+----------+------------+----------+
| employee_id | device_id   | username | department | office   |
+-------------+-------------+----------+------------+----------+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

This query retrieves all data from the employees table and filters the results to include employees working in the Marketing department located in the East building. The first condition, department = 'Marketing', filters for employees in the Marketing department, while the second condition, office LIKE 'East%', filters for employees situated in the East building. The '%' symbol, used with the LIKE operator, represents any number of unspecified characters.

Retrieve Employee Machines in the Finance or Sales Departments:
To carry out specific security updates, it was necessary to obtain information about employee machines belonging to the Finance and Sales departments.

To address this requirement, I constructed the following SQL query to filter and retrieve employee machines from the Finance or Sales departments:

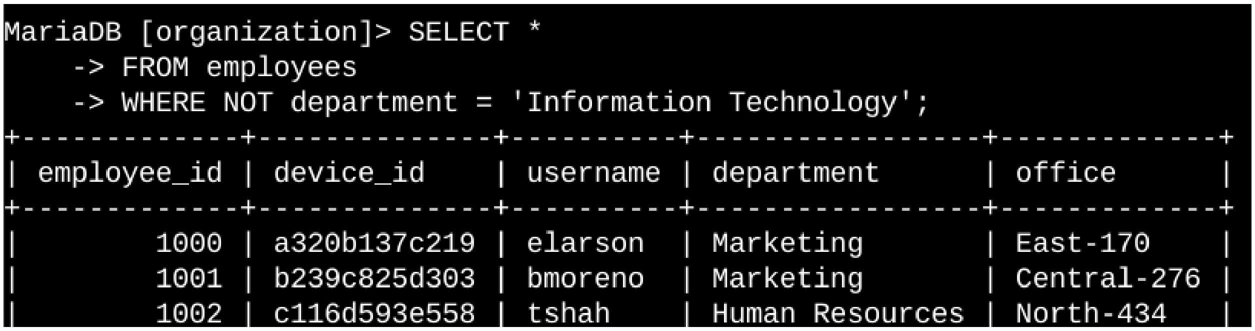


```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+-------------+----------+------------+-------------+
| employee_id | device_id   | username | department | office      |
+-------------+-------------+----------+------------+-------------+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

This query retrieves all data from the employees table and filters the results to include employees from either the Finance or Sales departments. The condition department = 'Finance' OR department = 'Sales' filters for employees from either department.

Retrieve Employee Machines Outside of the IT Department:
One final security update was required for employees not associated with the Information Technology department. To facilitate this update, it was necessary to gather information on these employees.

To address this requirement, I developed the following SQL query to filter and retrieve employee machines not affiliated with 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   |
```

This query retrieves all data from the employees table and filters the results to include employees who are not part of the Information Technology department. The condition NOT department = 'Information Technology' filters for employees not associated with this specific department.

Summary:
By skillfully applying filters to SQL queries, I successfully retrieved specific information regarding login attempts and employee machines. By employing different tables, such as log_in_attempts and employees, and utilizing operators such as AND, OR, and NOT, I was able to filter and extract the desired information for each task. Additionally, I utilized the LIKE operator with the '%' wildcard symbol to filter for specific patterns.