Apply filters to SQL queries

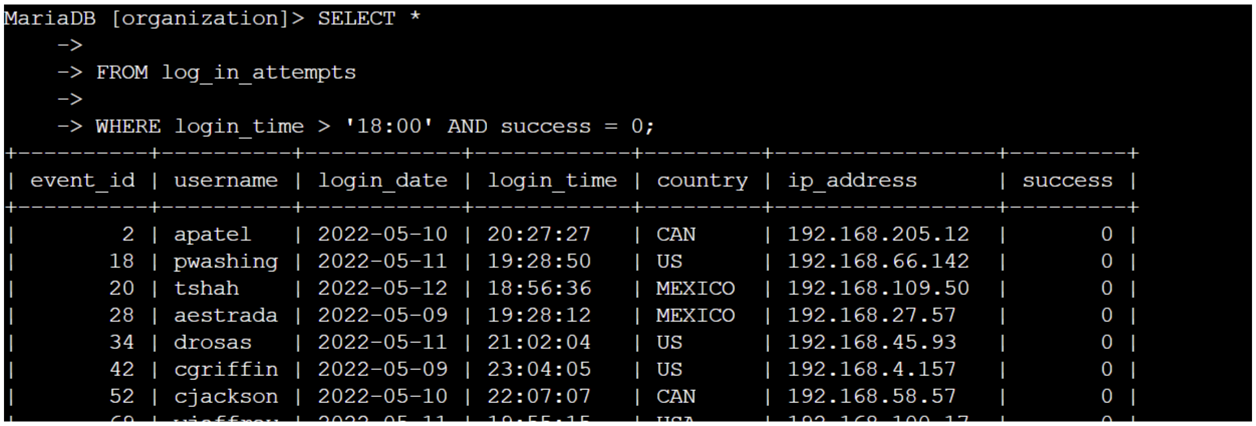
Project description

My role involves strengthening our organization’s system security by identifying and addressing potential vulnerabilities, investigating possible threats, and updating employee computers when necessary. The steps below illustrate how I used SQL with filters to carry out various security-related tasks.

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.

The following code demonstrates how I created a SQL query to filter for failed login attempts that occurred after business hours.



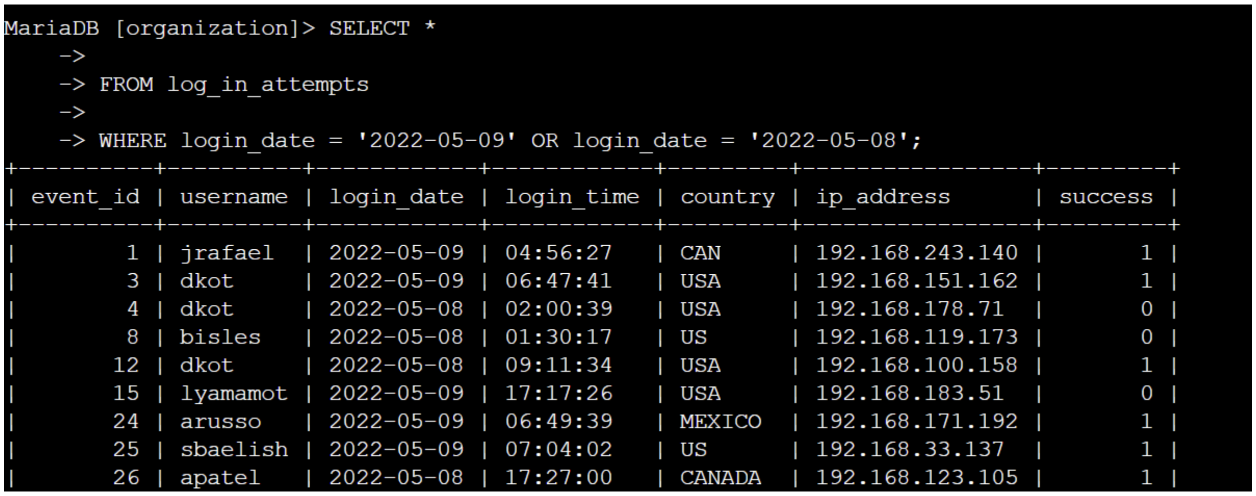
```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_time > '18:00' AND success = 0;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|       52 | cjackson | 2022-05-10 | 22:07:07   | CAN     | 192.168.58.57   |       0 |
```

The first section of the screenshot shows my SQL query, and the second section displays part of the resulting output. This query identifies failed login attempts that took place after 18:00. I began by selecting all records from the `log_in_attempts` table. Then, I applied a `WHERE` clause with an `AND` operator to narrow the results to only those login attempts that occurred after 18:00 and were unsuccessful. The first condition, `login_time > '18:00'`, filters for attempts made after 18:00, while 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.

The following code demonstrates how I created a SQL query to filter for login attempts that occurred on specific dates.



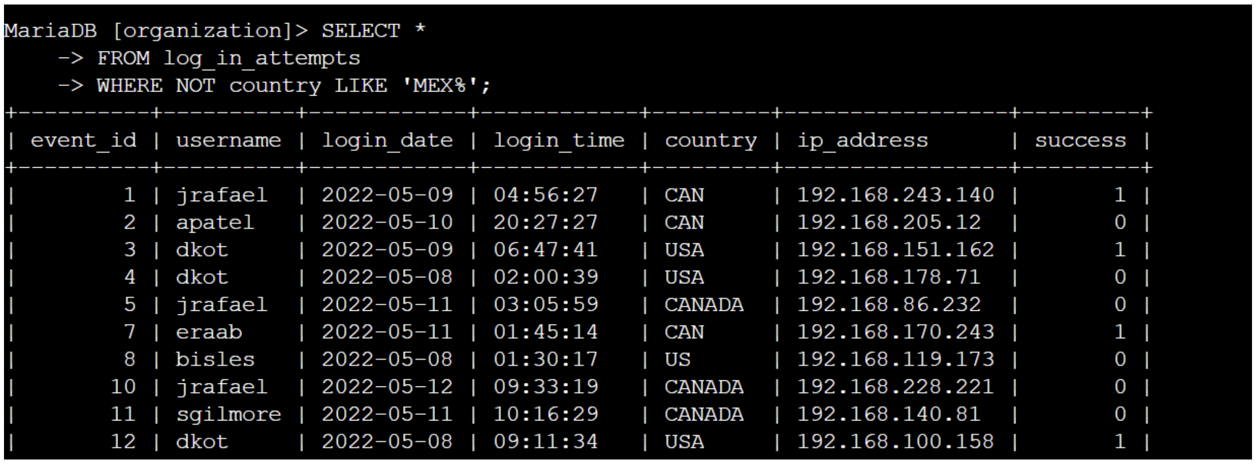
```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       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 |
|       24 | arusso   | 2022-05-09 | 06:49:39   | MEXICO  | 192.168.171.192 |       1 |
|       25 | sbaelish | 2022-05-09 | 07:04:02   | US      | 192.168.33.137  |       1 |
|       26 | apatel   | 2022-05-08 | 17:27:00   | CANADA  | 192.168.123.105 |       1 |
```

The first part of the screenshot shows my SQL query, and the second part displays a portion of the output. This query retrieves all login attempts that took place on either 2022-05-09 or 2022-05-08. I began by selecting all records from the `log_in_attempts` table. Next, I applied a `WHERE` clause with an `OR` operator to filter the results to only those attempts occurring on the specified dates. The first condition, `login_date = '2022-05-09'`, selects logins from 2022-05-09, while the second condition, `login_date = '2022-05-08'`, selects logins from 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.

The following code demonstrates how I created a SQL query to filter for login attempts that occurred outside of Mexico.



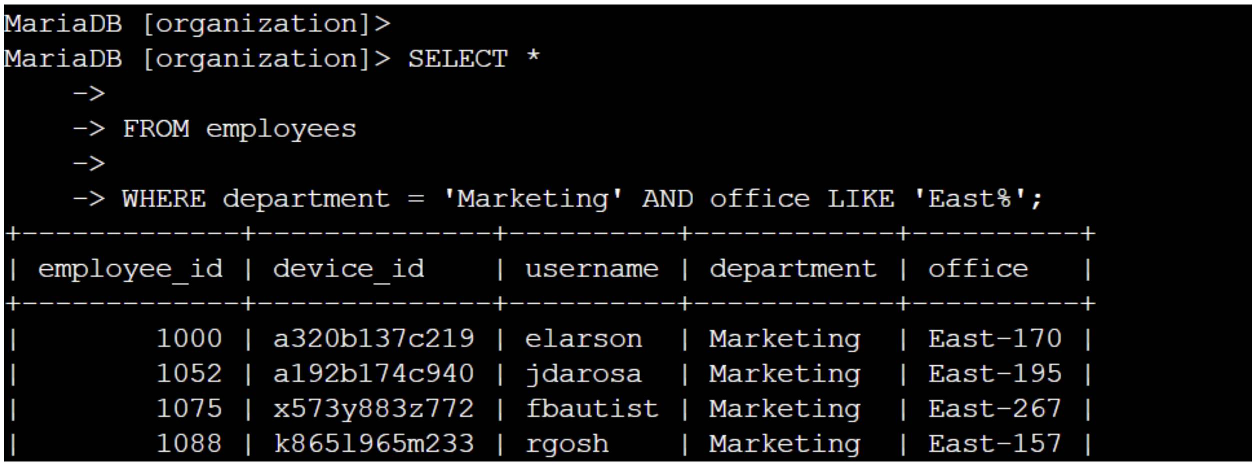
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       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 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
|       10 | jrafael  | 2022-05-12 | 09:33:19   | CANADA  | 192.168.228.221 |       0 |
|       11 | sgilmore | 2022-05-11 | 10:16:29   | CANADA  | 192.168.140.81  |       0 |
|       12 | dkot     | 2022-05-08 | 09:11:34   | USA     | 192.168.100.158 |       1 |
```

The first part of the screenshot shows my SQL query, and the second part displays part of the output. This query retrieves all login attempts from countries other than Mexico. I began by selecting all records from the `log_in_attempts` table, then applied a `WHERE` clause with `NOT` to exclude Mexico. To account for the fact that the dataset records Mexico as both `MEX` and `MEXICO`, I used `LIKE 'MEX%'` as the matching pattern. The percent sign (%) in `LIKE` represents 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.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Marketing department in the East building.



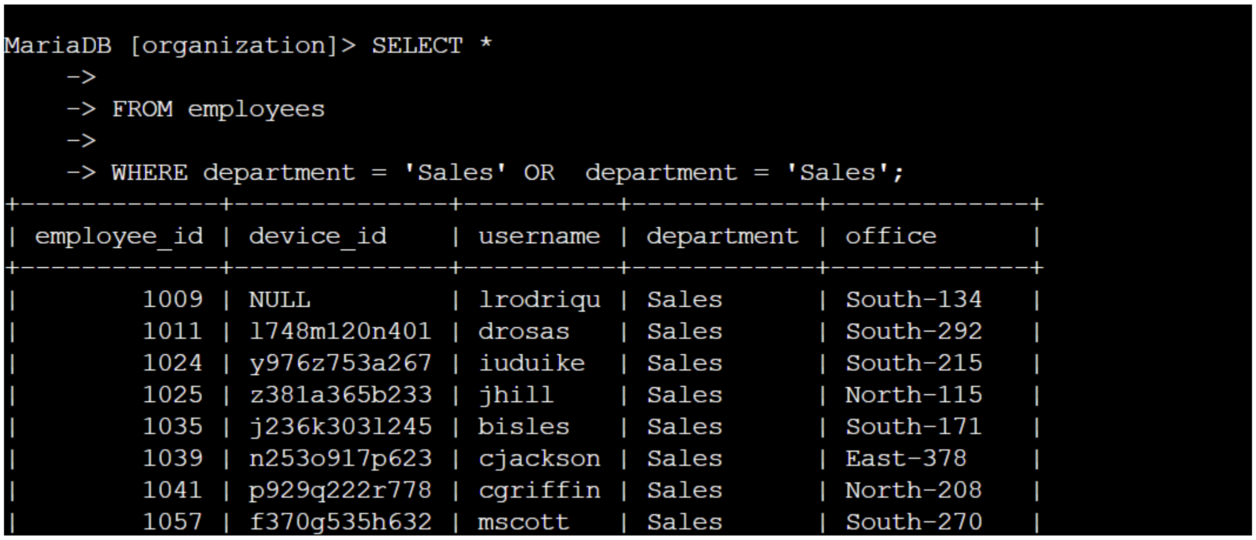
```
MariaDB [organization]>
MariaDB [organization]> SELECT *
    ->
    -> FROM employees
    ->
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+---------------+----------+------------+----------+
| employee_id | device_id     | username | department | office   |
+-------------+---------------+----------+------------+----------+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

The first part of the screenshot shows my SQL query, and the second part displays part of the output. This query retrieves all employees who work in the Marketing department and are located in the East building. I began by selecting all records from the `employees` table. Then, I applied a `WHERE` clause with an `AND` operator to filter for employees who meet both criteria. The first condition, `department = 'Marketing'`, selects employees in the Marketing department. The second condition, `office LIKE 'East%'`, matches any office in the East building, using `LIKE` with `East%` to account for the building name followed by a specific office number.

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.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Finance or Sales departments.



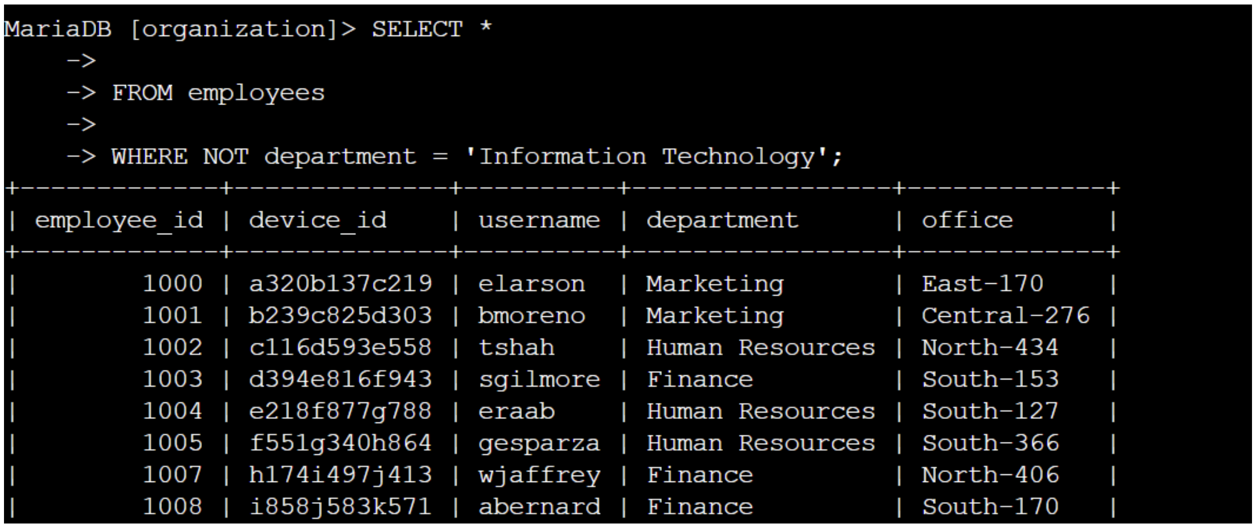
```
MariaDB [organization]> SELECT *
    ->
    -> FROM employees
    ->
    -> WHERE department = 'Sales' OR  department = 'Sales';
+-------------+---------------+----------+------------+-------------+
| employee_id | device_id     | username | department | office      |
+-------------+---------------+----------+------------+-------------+
|        1009 | NULL          | lrodriqu | Sales      | South-134   |
|        1011 | 1748m120n401  | drosas   | Sales      | South-292   |
|        1024 | y976z753a267  | iuduike  | Sales      | South-215   |
|        1025 | z381a365b233  | jhill    | Sales      | North-115   |
|        1035 | j236k3031245  | bisles   | Sales      | South-171   |
|        1039 | n253o917p623  | cjackson | Sales      | East-378    |
|        1041 | p929q222r778  | cgriffin | Sales      | North-208   |
|        1057 | f370g535h632  | mscott   | Sales      | South-270   |
```

The first part of the screenshot shows my SQL query, and the second part displays part of the output. This query retrieves all employees who work in either the Finance or Sales department. I began by selecting all records from the `employees` table, then applied a `WHERE` clause with the `OR` operator to include employees from both departments. The first condition, `department = 'Finance'`, filters for Finance department employees, while the second condition, `department = 'Sales'`, filters for Sales department employees.

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.

The following demonstrates how I created a SQL query to filter for employee machines from employees not in 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   |
|        1003 | d394e816f943 | sgilmore | Finance         | South-153   |
|        1004 | e218f877g788 | eraab    | Human Resources | South-127   |
|        1005 | f551g340h864 | gesparza | Human Resources | South-366   |
|        1007 | h174i497j413 | wjaffrey | Finance         | North-406   |
|        1008 | i858j583k571 | abernard | Finance         | South-170   |
```

The first part of the screenshot shows my SQL query, and the second part displays part of the output. This query retrieves all employees who are not in the Information Technology department. I began by selecting all records from the `employees` table, then applied a `WHERE` clause with `NOT` to exclude employees from that department.

Summary

I applied filters to SQL queries to extract specific information about login attempts and employee machines, working with two different tables: `log_in_attempts` and `employees`. Depending on the task, I used the `AND`, `OR`, and `NOT` operators to refine the results, as well as the `LIKE` operator with the `%` wildcard to match patterns.