# Apply filters to SQL queries

## Project Description: Security Incident Investigation with SQL

In this project, I acted as a security analyst for a large organization, investigating potential security issues related to suspicious login activity and employee devices. Using the organization's **employees** and **log_in_attempts** databases, I applied SQL filtering techniques to extract, analyze, and interpret records relevant to possible unauthorized access attempts.

The project involved:

- Reviewing organizational data to identify unusual login patterns
- Querying and filtering large datasets using SQL
- Cross-referencing employee information with login attempt logs
- Investigating anomalies to support proactive threat detection

This project demonstrates my ability to use SQL for security-focused data analysis, perform structured investigations, and contribute to maintaining organizational system integrity.

### Retrieve after hours failed login attempts

[My first task is to investigate a potential security incident that occurred after business hours, I queried the **log_in_attempts** table to identify all failed login attempts that took place after **18:00**. Using SQL filters, I selected records where login_time is later than '18:00:00' and the success column equals **0** (indicating a failed attempt). This query isolates suspicious after-hours activity and supports deeper security analysis.]

```
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 |
|       69 | wjaffrey | 2022-05-11 | 19:55:15   | USA     | 192.168.100.17  |       0 |
|       82 | abernard | 2022-05-12 | 23:38:46   | MEX     | 192.168.234.49  |       0 |
|       87 | apatel   | 2022-05-08 | 22:38:31   | CANADA  | 192.168.132.153 |       0 |
|       96 | ivelasco | 2022-05-09 | 22:36:36   | CAN     | 192.168.84.194  |       0 |
|      104 | asundara | 2022-05-11 | 18:38:07   | US      | 192.168.96.200  |       0 |
|      107 | bisles   | 2022-05-12 | 20:25:57   | USA     | 192.168.116.187 |       0 |
|      111 | aestrada | 2022-05-10 | 22:00:26   | MEXICO  | 192.168.76.27   |       0 |
|      127 | abellmas | 2022-05-09 | 21:20:51   | CANADA  | 192.168.70.122  |       0 |
|      131 | bisles   | 2022-05-09 | 20:03:55   | US      | 192.168.113.171 |       0 |
|      155 | cgriffin | 2022-05-12 | 22:18:42   | USA     | 192.168.236.176 |       0 |
|      160 | jclark   | 2022-05-10 | 20:49:00   | CANADA  | 192.168.214.49  |       0 |
|      199 | yappiah  | 2022-05-11 | 19:34:48   | MEXICO  | 192.168.44.232  |       0 |
+----------+----------+------------+------------+---------+-----------------+---------+
19 rows in set (0.084 sec)
```

As you can see we have flagged up all the failed login attempts after 18:00hrs for us to investigate.

## Retrieve login attempts on specific dates

[A suspicious event occurred on 2022-05-09. To investigate this event, I will review all login attempts which occurred on this day and the day before. I will use filters in SQL to create a query that identifies all login attempts that occurred on 2022-05-09 or 2022-05-08. (The date of the login attempt is found in the **login_date** column.)]

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-08' OR login_date = '2022-05-09';
+----------+----------+------------+------------+---------+-----------------+---------+
| 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 |
```

As you can see I have used the OR operator to find the login attempts for the TWO days in question

## 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, I need to investigate login attempts that occurred outside of Mexico. I will use filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico. (When referring to Mexico, the **country** column contains values of both **MEX** and **MEXICO**, and so I need to use the **LIKE** keyword with **%** to make sure my query reflects this.)]

```
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 |
```

I have used the NOT operator along with the LIKE 'Mex%' query to find all the log in attempts that are from outside the country of Mexico.

## Retrieve employees in Marketing

[Now the team wants to perform security updates on specific employee machines in the Marketing department. I am responsible for getting information on these employee machines and will need to query the **employees** table. I will use filters in SQL to create a query that identifies all employees in the Marketing department for all offices in the **East** building.

(The department of the employee is found in the **department** column, which contains values that include **Marketing**. The office is found in the office column. Some examples of values in this column are **East-170**, **East-320**, and **North-434**. I will need to use the **LIKE** keyword with **%** to filter for 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 |
|        1156 | a184b775c707 | dellery  | Marketing  | East-417 |
|        1163 | h679i515j339 | cwilliam | Marketing  | East-216 |
+-------------+--------------+----------+------------+----------+
7 rows in set (0.056 sec)
```

From the employees table, I filtered out all those in the marketing department who have their office in the East building

## Retrieve employees in Finance or Sales

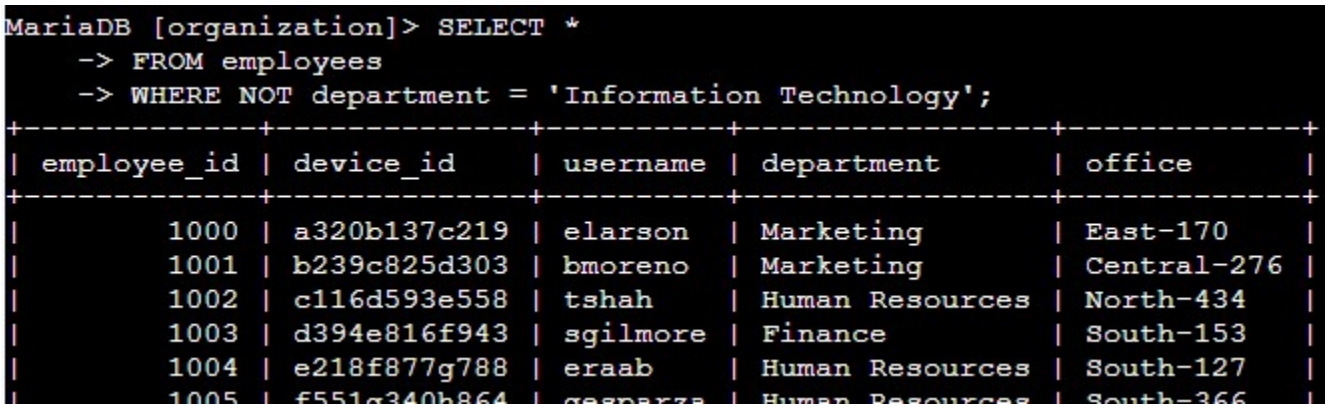
[Now the team needs to perform a different security update on machines for employees in the Sales and Finance departments. I will use filters in SQL to create a query that identifies all employees in the Sales or Finance departments. (The department of the employee is found in the **department** column, which contains values that include **Sales** and **Finance**.)]

```
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   |
|        1011 | l748m120n401 | drosas   | Sales      | South-292   |
```

Here we have filtered out all those in the Sales and Finance departments.

## Retrieve all employees not in IT

[Now the team needs to make one more update to employee machines. The employees who are in the Information Technology department have already had this update, but employees in all the other departments need it. I will use filters in SQL to create a query which identifies all employees NOT in the IT department. (The department of the employees is found in the **department** column, which contains values that include **Information Technology**.)]



```
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   |
```

To do this, I have used the NOT operator to filter out those who are not in the Information Technology departments.

# Summary

[In this project, I worked as a security analyst investigating a series of potential security incidents involving suspicious login activity and employee devices. Using SQL, I queried and filtered data from the **log_in_attempts** and **employees** tables to uncover patterns that could indicate unauthorized access. I examined failed login attempts after business hours, retrieved login activity across specific dates related to a suspicious event, and isolated login attempts originating outside of Mexico using operators such as OR, NOT, and LIKE. These analyses allowed me to highlight unusual behavior and support the team's broader security investigation.

In addition to reviewing login activity, I also analyzed employee data to support system security updates. I filtered records to identify employees in the Marketing department located in the East building, those working in either Sales or Finance, and all employees outside of the Information Technology department. By applying SQL filters to segment users by department and office location, I ensured accurate targeting for security maintenance. Overall, this project demonstrates my ability to apply SQL to real-world security scenarios, investigate anomalies, and contribute to maintaining a secure organizational environment.]