# Apply filters to SQL queries

## Project description

As a security professional at a large organization, my task is to investigate potential security incidents by analyzing login attempts and employee data using SQL queries. Through SQL filtering, I retrieve relevant records to identify suspicious login attempts, determine login locations, and gather employee information necessary for security updates. These queries help ensure the system remains secure by pinpointing irregular activities and supporting proactive measures.

## Retrieve after hours failed login attempts

To identify failed login attempts that occurred after business hours (18:00), I used the following SQL query:

```
SELECT *
FROM log_in_attempts
WHERE login_time > '18:00' AND success = FALSE;
```

```
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 (0.142 sec)
```

This query retrieves all records from the **log_in_attempts** table where the **login_time** is later than 18:00 and the **success** column is 0, indicating a failed login attempt. This helps identify unauthorized access attempts occurring after normal working hours.

## Retrieve login attempts on specific dates

To review all login attempts on May 8 and May 9, 2022, I used the following SQL query:

```
SELECT *
FROM log_in_attempts
WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

This query selects all records where the login_date matches either '2022-05-08' or '2022-05-09'. This allows investigation of suspicious activity that occurred on those specific dates.

## Retrieve login attempts outside of Mexico

To find login attempts that did not originate from Mexico, I used the following SQL query:

```
SELECT *
FROM log_in_attempts
WHERE NOT country LIKE 'MEX%';
```

This query retrieves all login attempts where the **country** column does not start with 'MEX' or 'MEXICO', ensuring that we filter out logins originating from Mexico and focus on potentially unauthorized foreign access.

## Retrieve employees in Marketing

To find employees in the Marketing department working in the East building, I used the following SQL query:

```
SELECT *
FROM employees
WHERE department = 'Marketing' AND office LIKE 'East%';
```

```
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 (0.031 sec)
```

This query selects all employees whose department contains 'Marketing' and whose office starts with 'East-', ensuring that only those located in the East building are included in the results.

## Retrieve employees in Finance or Sales

To identify employees working in either the Finance or Sales departments, I used the following SQL query:

```
SELECT *
FROM employees
WHERE department = 'Finance' OR department = 'Sales';
```

This query retrieves all records where the department column contains either 'Finance' or 'Sales', allowing the team to target security updates for these employees.

## Retrieve all employees not in IT

To exclude employees from the Information Technology department, I used the following SQL query:

```
SELECT *
FROM employees
WHERE NOT department = 'Information Technology';
```

This query selects all employees whose department does not contain 'Information Technology', ensuring that only employees from other departments are included for security updates.

## Summary

In this project, I used SQL queries to investigate security issues related to login attempts and employee data. I retrieved records of failed login attempts after hours, analyzed suspicious activity on specific dates, and filtered login attempts originating outside Mexico. Additionally, I identified employees in specific departments and locations to facilitate targeted security updates. These SQL-based investigations help maintain system integrity and strengthen security measures within the organization.