Apply filters to SQL queries

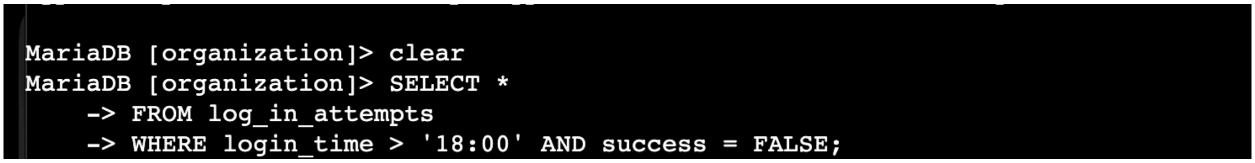
Project description

SQL (Structured Query Language) is a powerful tool used to manage and interact with relational databases. With SQL, our company can retrieve, insert, update, and delete data efficiently, enabling tasks like generating reports, performing data analysis, and managing large datasets. Additionally, SQL allows for complex operations like joining tables, aggregating data, and implementing security controls, making it indispensable for data-driven decision-making in various industries.

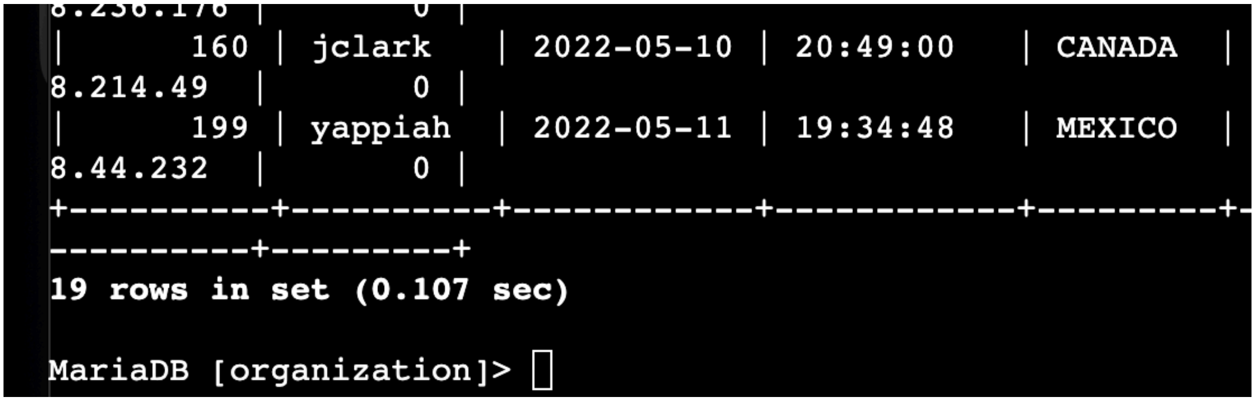
Retrieve after hours failed login attempts

In order to retrieve the failed login attempts that occurred after business hours.This is because there was a potential security breach that occurred. The following will discuss how the code will demonstrate how to retrieve after hours failed login attempts.

The first screenshot represents the input and the second screenshot outputs the information. Select all, from the login attempts, where the time was after 18:00 and failed. Simplistic and straight to the point and quite honestly a lot easier to read and understand. Here we see that 19 rows = answer of failed login attempts after business hours.



```
MariaDB [organization]> clear
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
```

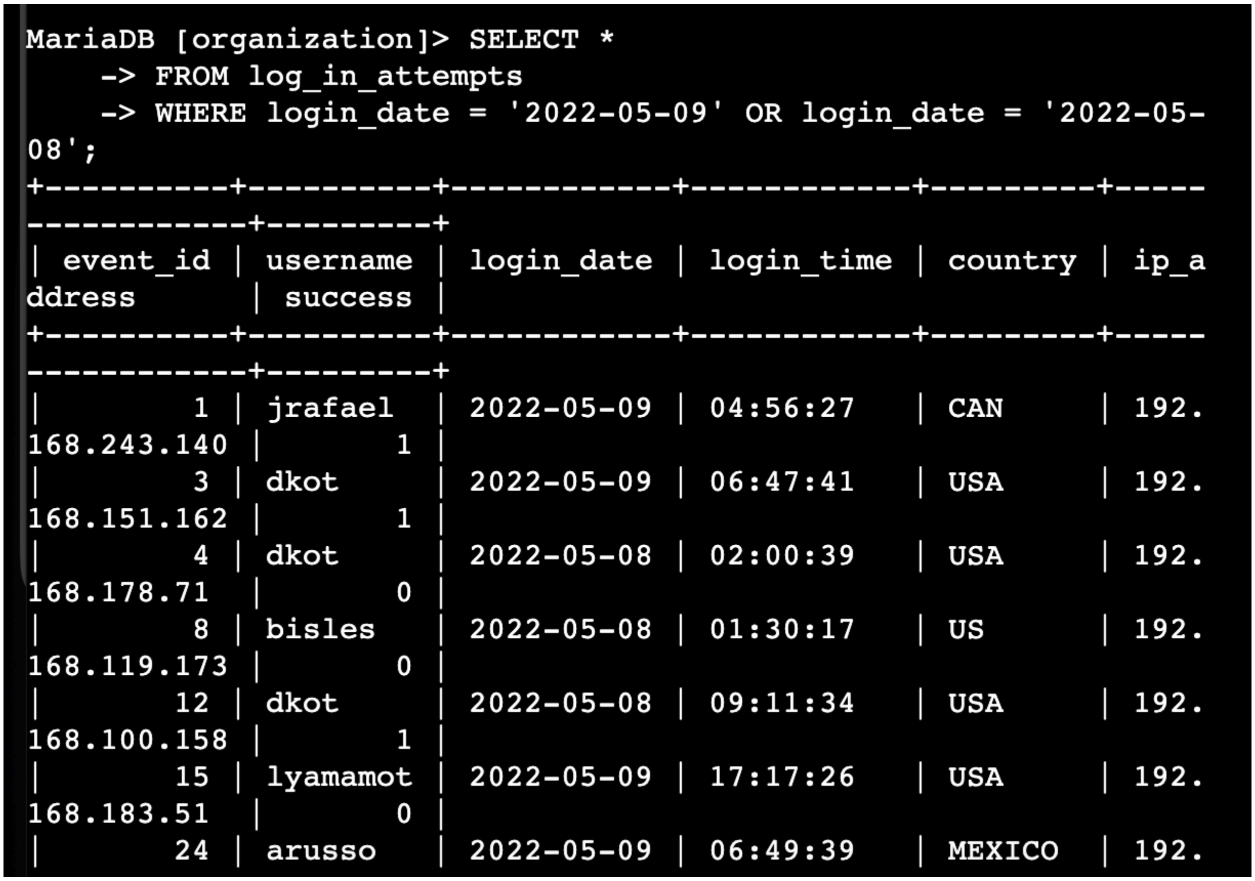


```
8.236.176 |        0 |
|       160 | jclark   | 2022-05-10 | 20:49:00   | CANADA  |
8.214.49  |        0 |
|       199 | yappiah  | 2022-05-11 | 19:34:48   | MEXICO  |
8.44.232  |        0 |
+----------+----------+------------+------------+---------+-
----------+---------+
19 rows in set (0.107 sec)

MariaDB [organization]> 
```

Retrieve login attempts on specific dates

Next, a suspicious event occurred on 09-05-22 and the company wants to further investigate. TIn order to retrieve the following, we must change the programming in the SQL to successfully retrieve login attempts on specific dates. We do this by using login_date instead of time when attempts were made. We will also be using the OR filter to ensure the system is picking up on specific dates. The following:
SELECT *
FROM log_in_attempts
WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
(75 Rows of information=login attempts made on specific dates)



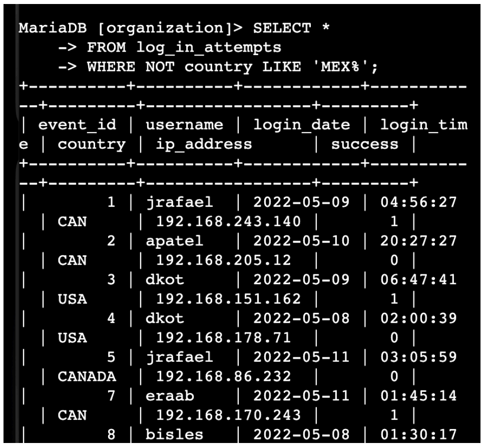
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-
08';
+----------+----------+------------+------------+---------+-----
------------+---------+
| event_id | username | login_date | login_time | country | ip_a
ddress     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

Retrieve login attempts outside of Mexico

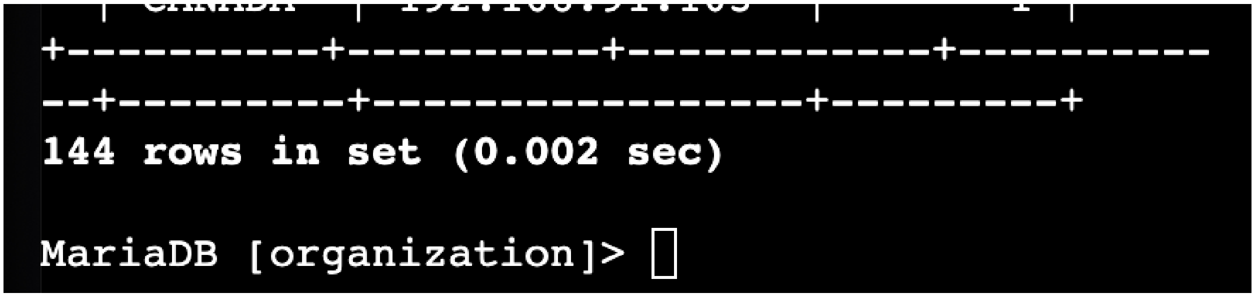
Furthermore, it is believed that there are more issues that need to be addressed and investigated. To retrieve the login attempts outside of Mexico through SQL, we must follow a similar process but change some of the variables. We go ahead and use the following since we are still on attempts,
SELECT *
FROM log_in_attempts

WHERE NOT country LIKE 'MEX%';

The steps here are clearly attempting to see the log in attempts from everyone where the country is not Mexico. When scrolling to the bottom to see the following information, we can see that 144 login attempts were made outside of Mexico.



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



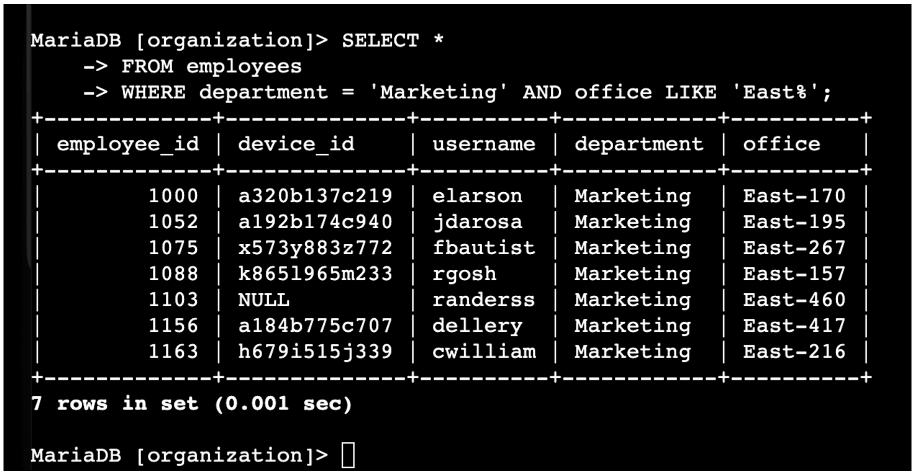
```
  | CANADA  | 192.168.91.103  |       1 |
+----------+----------+------------+----------
--+---------+-----------------+---------+
144 rows in set (0.002 sec)

MariaDB [organization]> 
```

Retrieve employees in Marketing

As per our last retrieval, it is once again similar but different as we are trying to extract different information so must input a different key word into the SQL system.

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



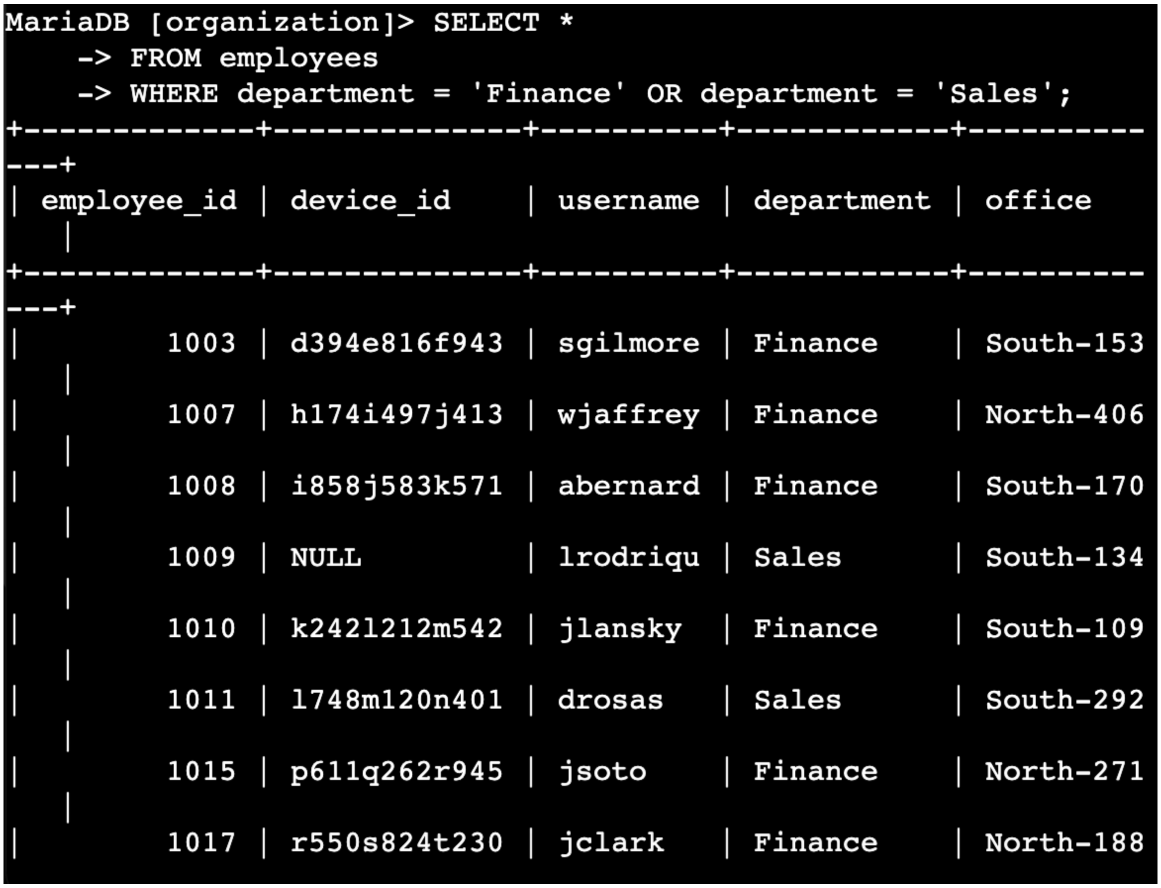
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+--------------+----------+------------+----------+
| employee_id | device_id    | username | department | office   |
+-------------+--------------+----------+------------+----------+
|        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 (0.001 sec)

MariaDB [organization]> 
```

Retrieve employees in Finance or Sales

In order to perform an update, the company must first retrieve all employees in sales or finance. The following is an SQL query to retrieve records for the following

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



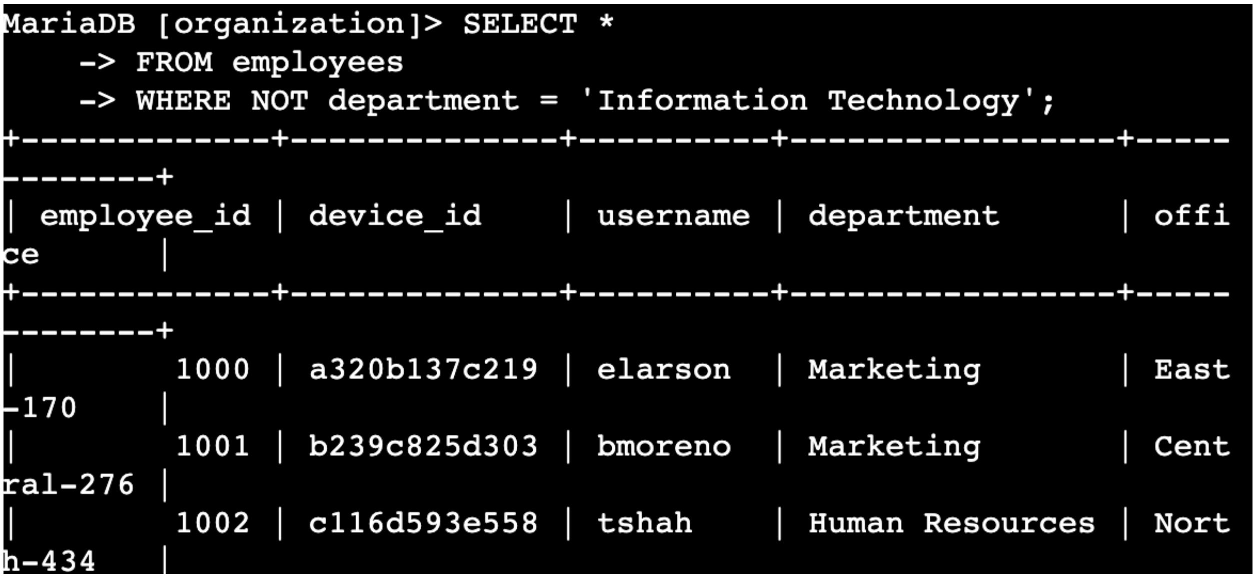
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+--------------+----------+------------+----------
---+
| employee_id | device_id    | username | department | office
   |
+-------------+--------------+----------+------------+----------
---+
|        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 | k242l212m542 | jlansky  | Finance    | South-109
   |
|        1011 | l748m120n401 | drosas   | Sales      | South-292
   |
|        1015 | p611q262r945 | jsoto    | Finance    | North-271
   |
|        1017 | r550s824t230 | jclark   | Finance    | North-188
```

Retrieved!

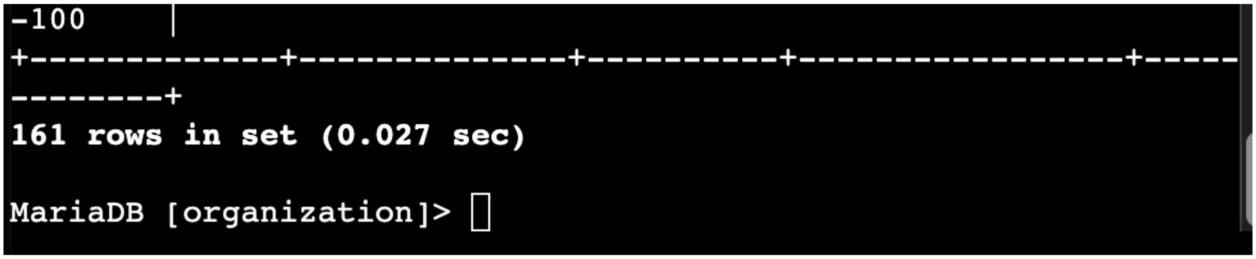
Retrieve all employees **not** in IT

Lastly, the company needs to retrieve all employes in IT for the last update. The team needs to following information about employees not in the department of IT. TThe query used to retrieve the following records is as follows:

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



```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE NOT department = 'Information Technology';
+-------------+-------------+----------+-----------------+-----
--------+
| employee_id | device_id   | username | department      | offi
ce      |
+-------------+-------------+----------+-----------------+-----
--------+
|        1000 | a320b137c219 | elarson  | Marketing       | East
-170    |
|        1001 | b239c825d303 | bmoreno  | Marketing       | Cent
ral-276 |
|        1002 | c116d593e558 | tshah    | Human Resources | Nort
h-434   |
```



```
-100    |
+-------------+-------------+----------+-----------------+-----
--------+
161 rows in set (0.027 sec)

MariaDB [organization]> 
```

Summary

In conclusion, SQL (Structured Query Language) is a standard programming language used for interacting with relational databases. It provides the functionality to create, retrieve, update, and delete data stored in database tables. SQL is essential for managing data and is widely used across industries for tasks such as the following that have been provided. SQL queries to retrieve information from a database have now been completed with specific AND, OR , and NOT operators to filter out specific SQL queries.