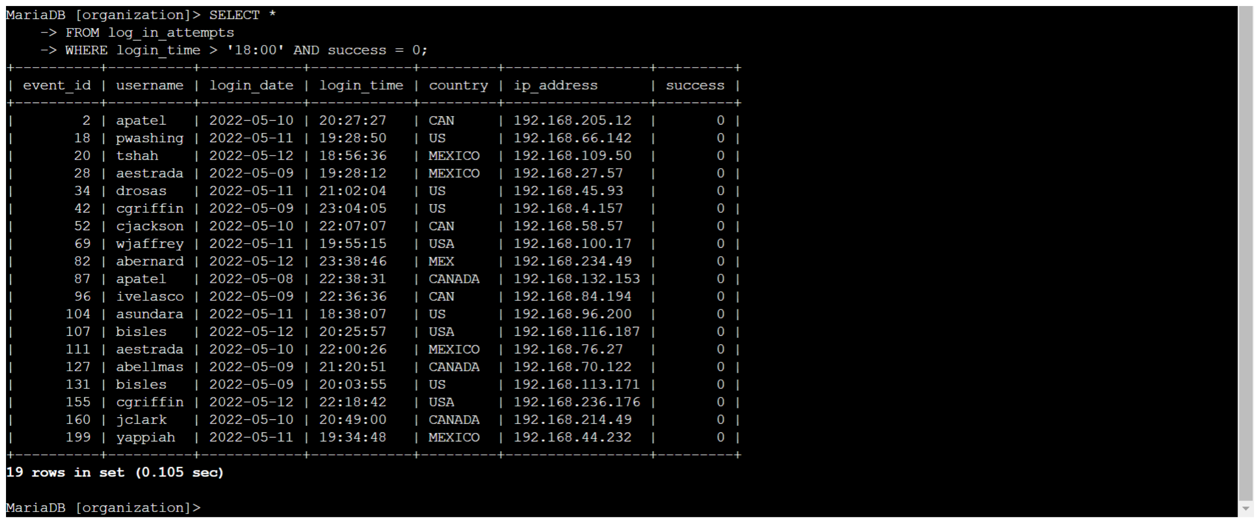
Apply filters to SQL queries

Project description

This project involves investigating potential security issues related to login attempts and employee machines within an organization. Using SQL queries, I will analyze data from the `employees` and `log_in_attempts` tables to identify anomalies, such as repeated failed login attempts, unauthorized access, or unusual activity patterns. The goal is to detect potential security threats and provide actionable insights to enhance the organization's system security.

Retrieve after hours failed login attempts

I discovered a potential security incident related to after-hours login activity. To investigate, I queried the `log_in_attempts` table and applied SQL filters to identify all failed login attempts (success = 0 or FALSE) that occurred after 18:00, based on the `login_time` column.



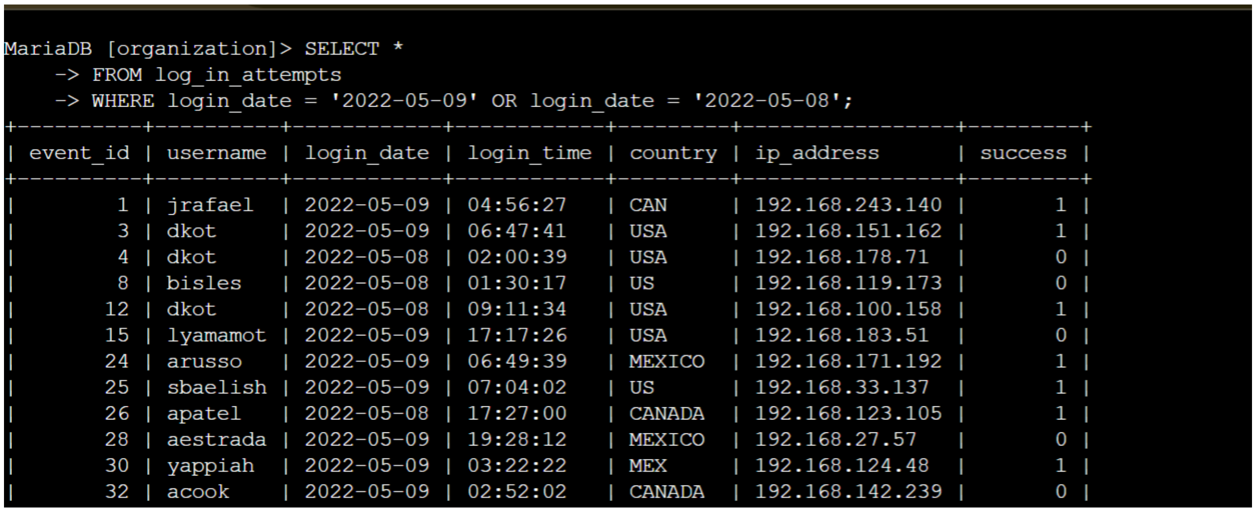
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = 0;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 (0.105 sec)

MariaDB [organization]>
```

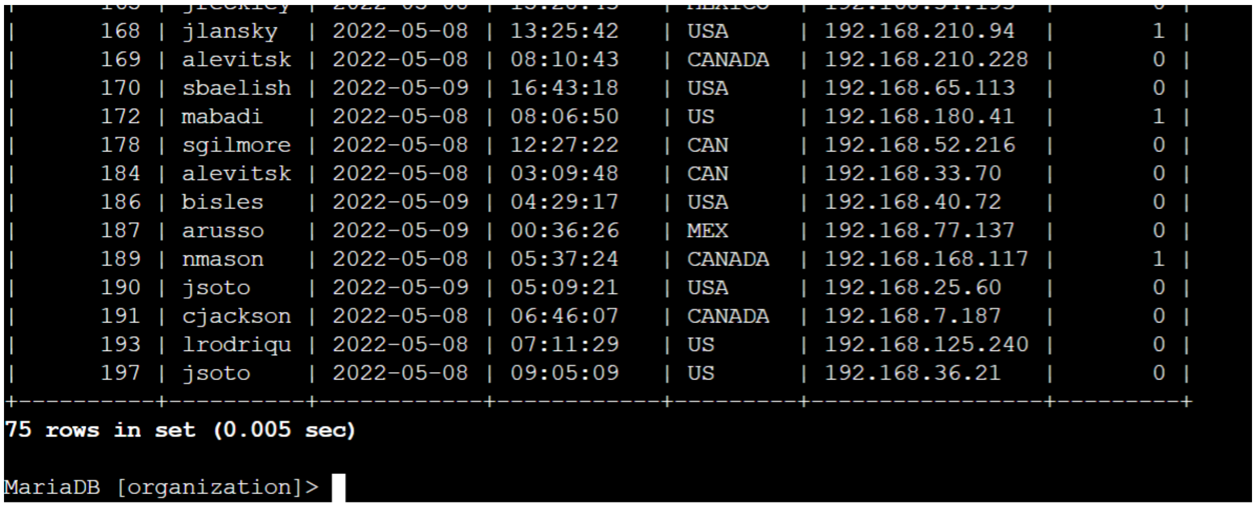
The results are based on login times after 18:00. I used the `WHERE` clause with the > operator to specify login attempts occurring after this time, indicating activity outside of working hours. Additionally, I added another condition using the `AND` operator to ensure both conditions were met simultaneously: login times after 18:00 and failed login attempts. For failed logins, I used `AND success = 0;`, where `0` indicates a false login.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09, so I wanted to investigate login attempts on that day and the previous day (2022-05-08). To do this, I used an SQL query to filter for login attempts that occurred on either 2022-05-09 or 2022-05-08, using the `login_date` column to identify the specific dates. This query helped me focus on the relevant login activity for my investigation.



```
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 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.57   |       0 |
|       30 | yappiah  | 2022-05-09 | 03:22:22   | MEX     | 192.168.124.48  |       1 |
|       32 | acook    | 2022-05-09 | 02:52:02   | CANADA  | 192.168.142.239 |       0 |
```



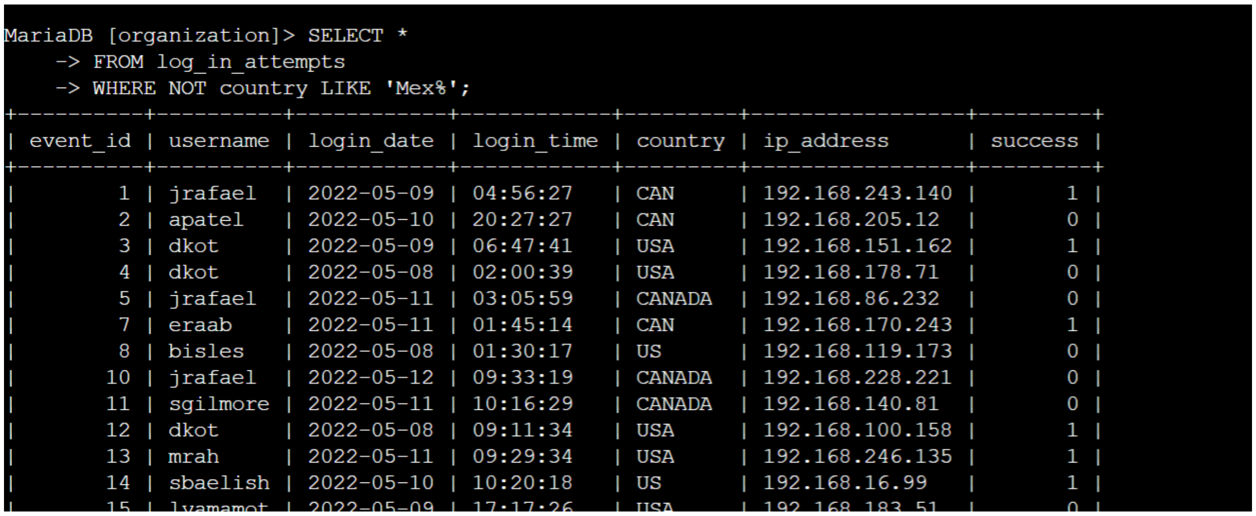
```
|      168 | jlansky  | 2022-05-08 | 13:25:42   | USA     | 192.168.210.94  |       1 |
|      169 | alevitsk | 2022-05-08 | 08:10:43   | CANADA  | 192.168.210.228 |       0 |
|      170 | sbaelish | 2022-05-09 | 16:43:18   | USA     | 192.168.65.113  |       0 |
|      172 | mabadi   | 2022-05-08 | 08:06:50   | US      | 192.168.180.41  |       1 |
|      178 | sgilmore | 2022-05-08 | 12:27:22   | CAN     | 192.168.52.216  |       0 |
|      184 | alevitsk | 2022-05-08 | 03:09:48   | CAN     | 192.168.33.70   |       0 |
|      186 | bisles   | 2022-05-09 | 04:29:17   | USA     | 192.168.40.72   |       0 |
|      187 | arusso   | 2022-05-09 | 00:36:26   | MEX     | 192.168.77.137  |       0 |
|      189 | nmason   | 2022-05-08 | 05:37:24   | CANADA  | 192.168.168.117 |       1 |
|      190 | jsoto    | 2022-05-09 | 05:09:21   | USA     | 192.168.25.60   |       0 |
|      191 | cjackson | 2022-05-08 | 06:46:07   | CANADA  | 192.168.7.187   |       0 |
|      193 | lrodriqu | 2022-05-08 | 07:11:29   | US      | 192.168.125.240 |       0 |
|      197 | jsoto    | 2022-05-08 | 09:05:09   | US      | 192.168.36.21   |       0 |
+----------+----------+------------+------------+---------+-----------------+---------+
75 rows in set (0.005 sec)

MariaDB [organization]>
```

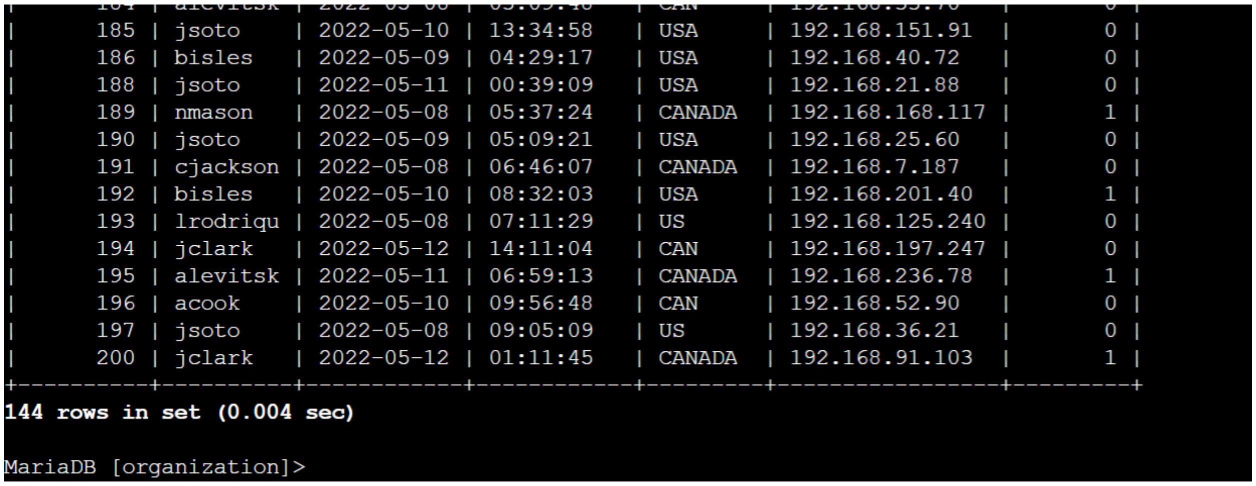
I selected the `log_in_attempts` table and used the `WHERE` clause with the `OR` operator to filter the results and include the specified dates, allowing either condition to be met. As a result, the output showed that there were 75 login attempts on those two days.

Retrieve login attempts outside of Mexico

There was suspicious activity with login attempts, but the team determined that it didn't originate from Mexico. To investigate login attempts outside of Mexico, I created an SQL query that filters out any records with the country values of "MEX" or "MEXICO" in the `country` column.



```
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 |
|       13 | mrah     | 2022-05-11 | 09:29:34   | USA     | 192.168.246.135 |       1 |
|       14 | sbaelish | 2022-05-10 | 10:20:18   | US      | 192.168.16.99   |       1 |
|       15 | lyamamot | 2022-05-09 | 17:17:26   | USA     | 192.168.183.51  |       0 |
```



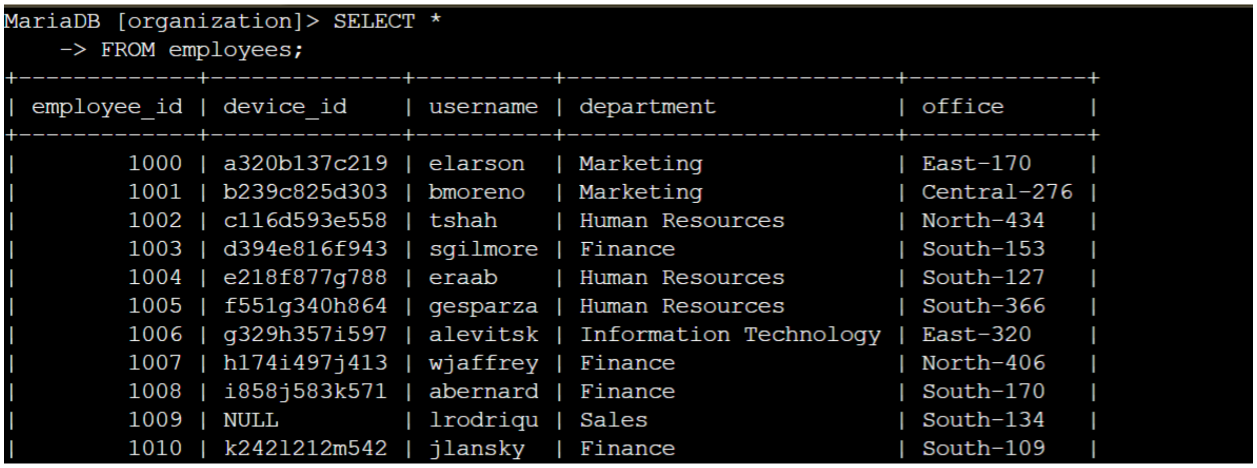
```
|      185 | jsoto    | 2022-05-10 | 13:34:58   | USA     | 192.168.151.91  |       0 |
|      186 | bisles   | 2022-05-09 | 04:29:17   | USA     | 192.168.40.72   |       0 |
|      188 | jsoto    | 2022-05-11 | 00:39:09   | USA     | 192.168.21.88   |       0 |
|      189 | nmason   | 2022-05-08 | 05:37:24   | CANADA  | 192.168.168.117 |       1 |
|      190 | jsoto    | 2022-05-09 | 05:09:21   | USA     | 192.168.25.60   |       0 |
|      191 | cjackson | 2022-05-08 | 06:46:07   | CANADA  | 192.168.7.187   |       0 |
|      192 | bisles   | 2022-05-10 | 08:32:03   | USA     | 192.168.201.40  |       1 |
|      193 | lrodriqu | 2022-05-08 | 07:11:29   | US      | 192.168.125.240 |       0 |
|      194 | jclark   | 2022-05-12 | 14:11:04   | CAN     | 192.168.197.247 |       0 |
|      195 | alevitsk | 2022-05-11 | 06:59:13   | CANADA  | 192.168.236.78  |       1 |
|      196 | acook    | 2022-05-10 | 09:56:48   | CAN     | 192.168.52.90   |       0 |
|      197 | jsoto    | 2022-05-08 | 09:05:09   | US      | 192.168.36.21   |       0 |
|      200 | jclark   | 2022-05-12 | 01:11:45   | CANADA  | 192.168.91.103  |       1 |
+----------+----------+------------+------------+---------+-----------------+---------+
144 rows in set (0.004 sec)

MariaDB [organization]>
```

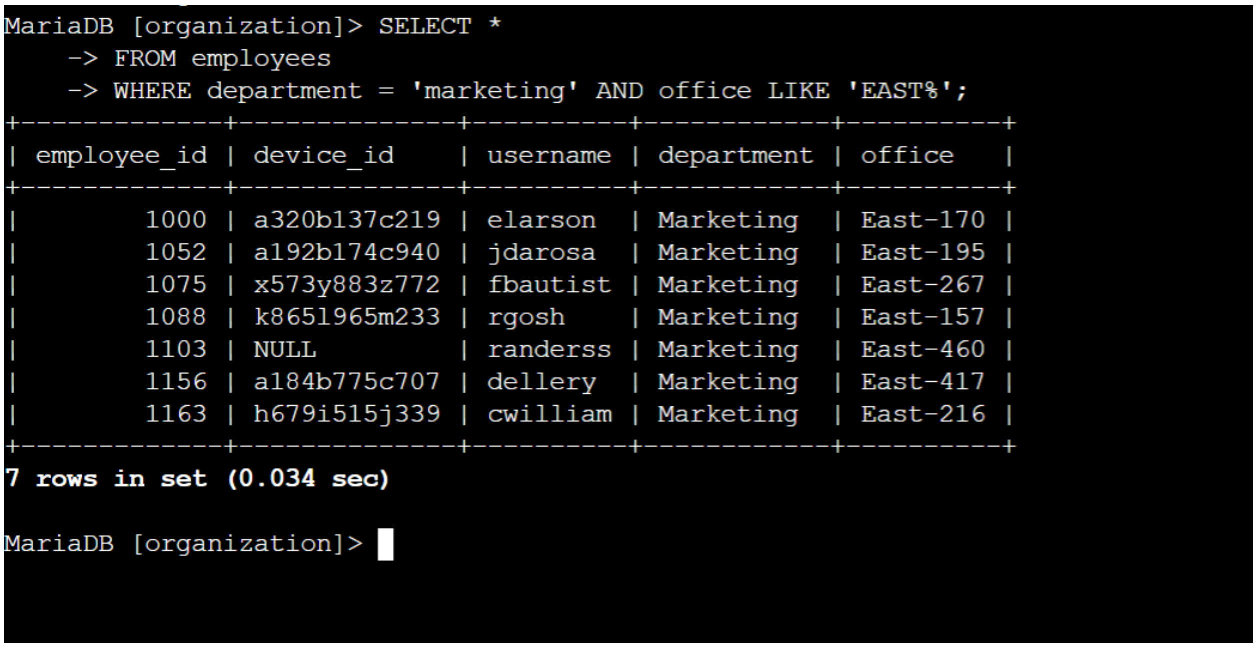
I used the `WHERE` clause and the `NOT` operator to filter the results and retrieve login attempts from outside of Mexico. Since the `country` column contains both "MEX" and "MEXICO" as values for Mexico, I used the `LIKE` keyword with `%` to ensure the query accurately reflects this.

Retrieve employees in Marketing

I was tasked with identifying employees in the Marketing department across all offices in the East building for security updates. To do this, I created an SQL query using filters to select employees from the `employees` table, where the `department` column contained "Marketing" and the `office` column contained values like "East-170" and "East-320." I used the `LIKE` keyword with % to ensure the query filtered for all offices in the East building.



```
MariaDB [organization]> SELECT *
    -> FROM employees;
+-------------+---------------+----------+------------------------+-------------+
| 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   |
|        1006 | g329h357i597  | alevitsk | Information Technology | East-320    |
|        1007 | h174i497j413  | wjaffrey | Finance                | North-406   |
|        1008 | i858j583k571  | abernard | Finance                | South-170   |
|        1009 | NULL          | lrodriqu | Sales                  | South-134   |
|        1010 | k2421212m542  | jlansky  | Finance                | South-109   |
```



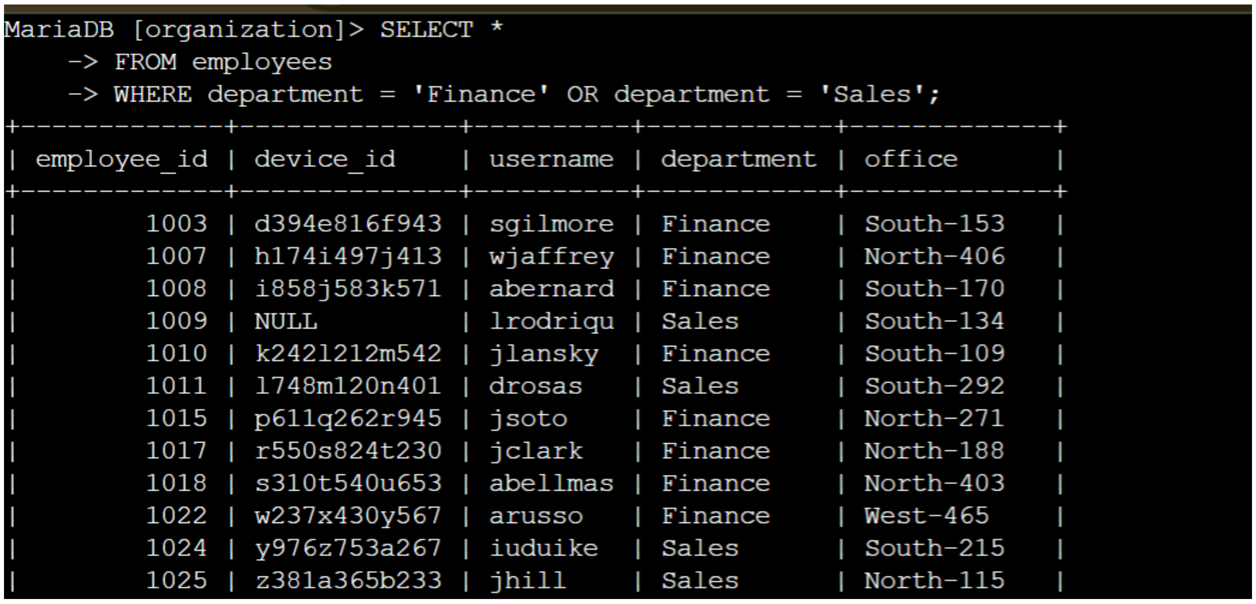
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'marketing' AND office LIKE 'EAST%';
+-------------+---------------+----------+------------+----------+
| employee_id | device_id     | username | department | office   |
+-------------+---------------+----------+------------+----------+
|        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 (0.034 sec)

MariaDB [organization]>
```

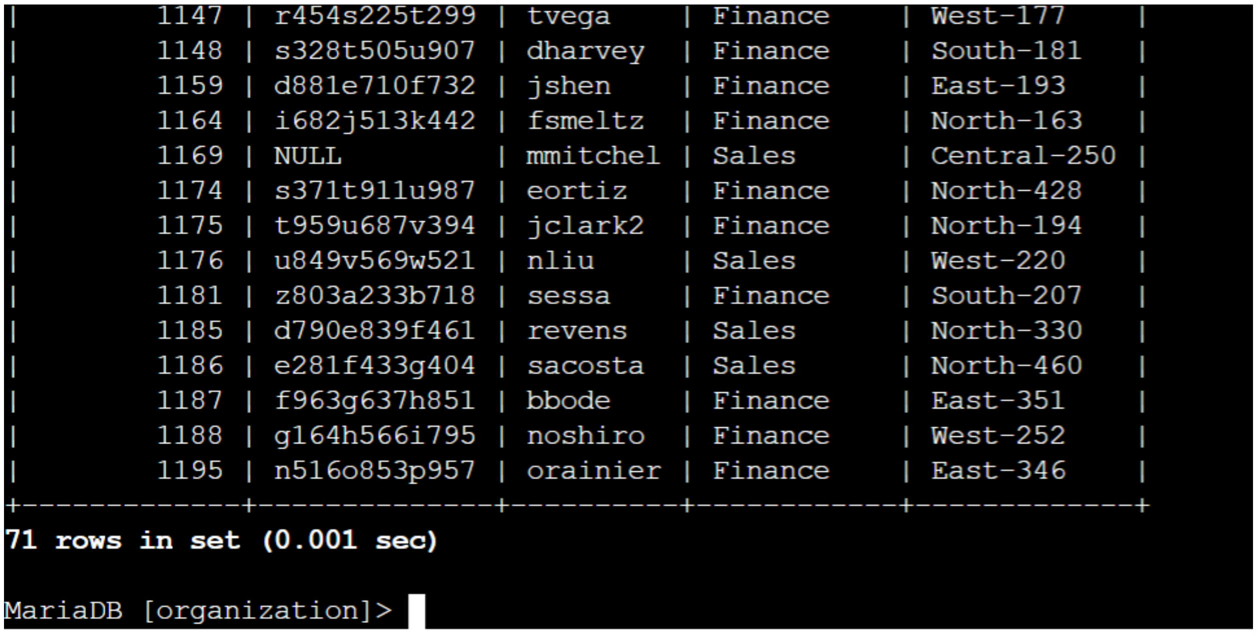
First, I ran `SELECT * FROM employees;` to view the columns and values in the `employees` table. Then, I executed another query to retrieve all employees in the Marketing department who are located in the East building. I used the `WHERE` clause to filter employees based on the Marketing department and the East building by including the condition `AND office LIKE 'EAST%';`. As a result, I identified 7 employees who matched the criteria.

Retrieve employees in Finance or Sales

I was tasked with identifying employees in the Sales or Finance departments for a security update. I created an SQL query using filters to retrieve all employees from the `employees` table where the `department` column contained "Sales" or "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 | 1748m120n401  | drosas   | Sales      | South-292   |
|        1015 | p611q262r945  | jsoto    | Finance    | North-271   |
|        1017 | r550s824t230  | jclark   | Finance    | North-188   |
|        1018 | s310t540u653  | abellmas | Finance    | North-403   |
|        1022 | w237x430y567  | arusso   | Finance    | West-465    |
|        1024 | y976z753a267  | iuduike  | Sales      | South-215   |
|        1025 | z381a365b233  | jhill    | Sales      | North-115   |
```



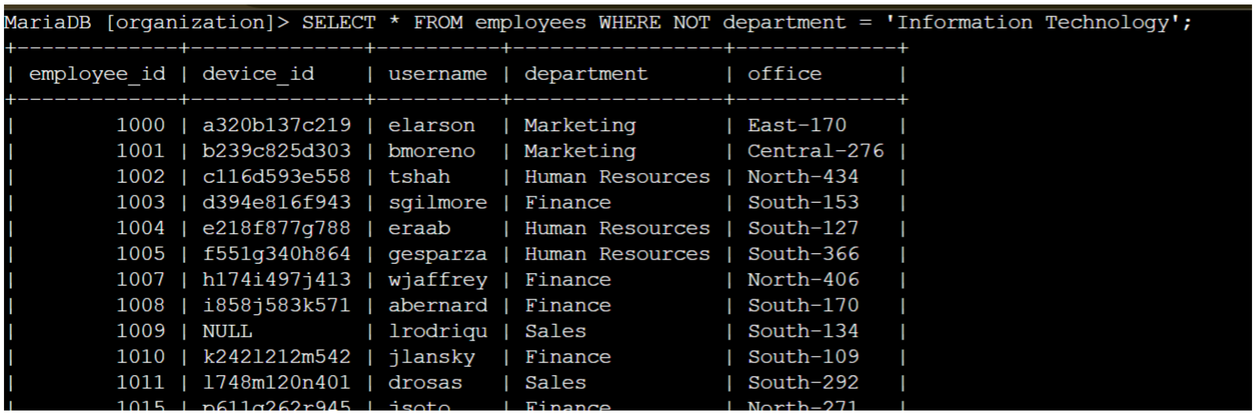
```
|        1147 | r454s225t299  | tvega    | Finance    | West-177    |
|        1148 | s328t505u907  | dharvey  | Finance    | South-181   |
|        1159 | d881e710f732  | jshen    | Finance    | East-193    |
|        1164 | i682j513k442  | fsmeltz  | Finance    | North-163   |
|        1169 | NULL          | mmitchel | Sales      | Central-250 |
|        1174 | s371t911u987  | eortiz   | Finance    | North-428   |
|        1175 | t959u687v394  | jclark2  | Finance    | North-194   |
|        1176 | u849v569w521  | nliu     | Sales      | West-220    |
|        1181 | z803a233b718  | sessa    | Finance    | South-207   |
|        1185 | d790e839f461  | revens   | Sales      | North-330   |
|        1186 | e281f433g404  | sacosta  | Sales      | North-460   |
|        1187 | f963g637h851  | bbode    | Finance    | East-351    |
|        1188 | g164h566i795  | noshiro  | Finance    | West-252    |
|        1195 | n516o853p957  | orainier | Finance    | East-346    |
+-------------+---------------+----------+------------+-------------+
71 rows in set (0.001 sec)

MariaDB [organization]>
```

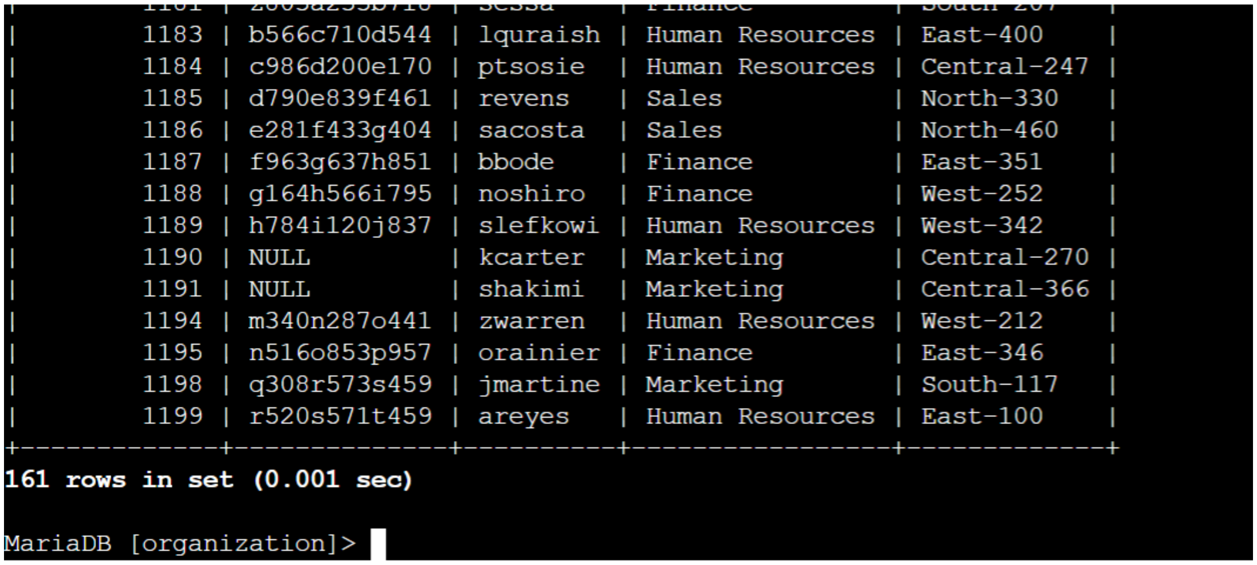
I used the `WHERE` clause and the `OR` operator to query employees working in either the Finance or Sales department. As a result, the output showed that 71 people worked in either the Finance or Sales department.

Retrieve all employees not in IT

I was tasked with identifying employees who still need a machine update, as those in the Information Technology department already received it. I created an SQL query using filters to exclude employees from the IT department by applying a condition in the `department` column.



```
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   |
|        1009 | NULL         | lrodriqu | Sales           | South-134   |
|        1010 | k2421212m542 | jlansky  | Finance         | South-109   |
|        1011 | 1748m120n401 | drosas   | Sales           | South-292   |
|        1015 | p611q262r945 | isoto    | Finance         | North-271   |
```



```
|        1183 | b566c710d544 | lquraish | Human Resources | East-400    |
|        1184 | c986d200e170 | ptsosie  | Human Resources | Central-247 |
|        1185 | d790e839f461 | revens   | Sales           | North-330   |
|        1186 | e281f433g404 | sacosta  | Sales           | North-460   |
|        1187 | f963g637h851 | bbode    | Finance         | East-351    |
|        1188 | g164h566i795 | noshiro  | Finance         | West-252    |
|        1189 | h784i120j837 | slefkowi | Human Resources | West-342    |
|        1190 | NULL         | kcarter  | Marketing       | Central-270 |
|        1191 | NULL         | shakimi  | Marketing       | Central-366 |
|        1194 | m340n287o441 | zwarren  | Human Resources | West-212    |
|        1195 | n516o853p957 | orainier | Finance         | East-346    |
|        1198 | q308r573s459 | jmartine | Marketing       | South-117   |
|        1199 | r520s571t459 | areyes   | Human Resources | East-100    |
+-------------+-------------+----------+-----------------+-------------+
161 rows in set (0.001 sec)

MariaDB [organization]>
```

First, I selected all data from the `employees` table. Then, I used the `NOT` operator in the `WHERE` clause to exclude employees from the Information Technology department.

Summary

In this project, I successfully investigated various security-related scenarios using SQL queries. I analyzed login attempts to identify suspicious activity, including failed logins after working hours, login attempts outside specific regions, and activity during specific dates. I also retrieved and filtered data to identify employees based on department and location for targeted security updates.

Through these tasks, I was able to:

- Identify login attempts after hours and isolate failed attempts.
- Exclude login attempts originating from a specific country to narrow down suspicious activity.

- Filter login attempts by date to investigate incidents occurring on or around specific days.
- Retrieve employees in certain departments and buildings for machine updates.
- Exclude employees in the Information Technology department who had already received updates.

This project demonstrated my ability to effectively use SQL filters, logical operators, and conditions to analyze data, investigate potential security threats, and support security updates. The queries provided actionable insights to enhance organizational security measures.