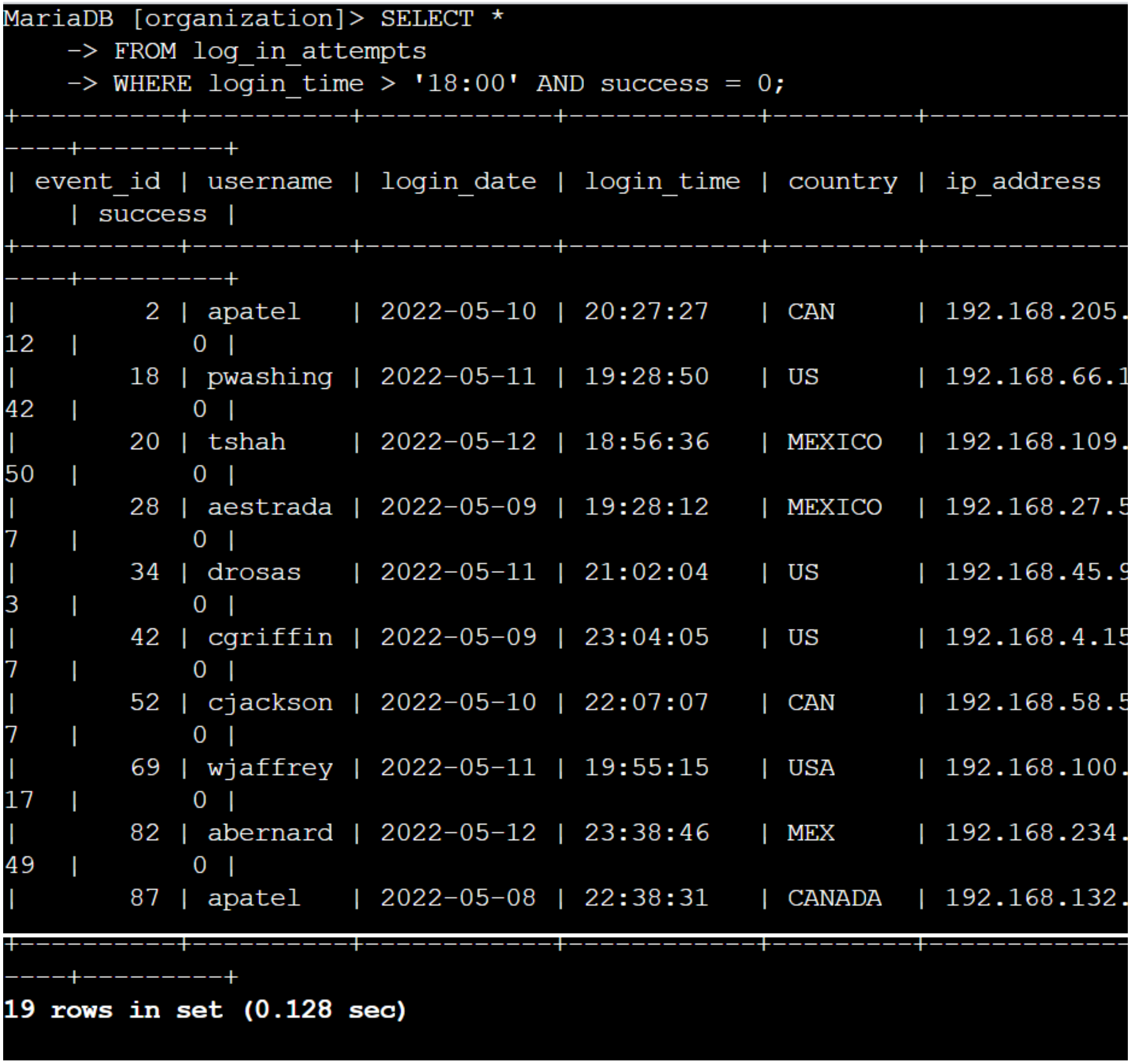
Apply filters to SQL queries

Project description

My security analysis required a task to review the employees and log_in_attempts datasets through SQL filters that employed AND, OR, NOT operators to detect unauthorized access patterns together with untrusted geographic entries and behavior from former staff members or those out of compliance. The investigation process separated uncommon events such as unreasonable login failures and late night system usage to generate practical findings that enhanced security standards for authentication measures and updated clearance regulations to minimize security breaches.

Retrieve after hours failed login attempts

The log_in_attempts table data shows 19 unsuccessful login attempts that took place after 18:00 when success is assigned a zero value which means failed attempts. The log data indicates repeated failed login attempts which originated from US and Mexican locations as well as Canada. Users apatel as well as pwashing tried to access multiple times. This query analyzes unsuccessful after-hours logins between 18:56 and 23:38 during different times of day. Each unsuccessful login attempt contains record of IP address and geographical location information.



```
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      | 192.168.66.1
42   |       0 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.109.
50   |       0 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.5
7    |       0 |
|       34 | drosas   | 2022-05-11 | 21:02:04   | US      | 192.168.45.9
3    |       0 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.15
7    |       0 |
|       52 | cjackson | 2022-05-10 | 22:07:07   | CAN     | 192.168.58.5
7    |       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.
+----------+----------+------------+------------+---------+------------
----+---------+
19 rows in set (0.128 sec)
```

Retrieve login attempts on specific dates

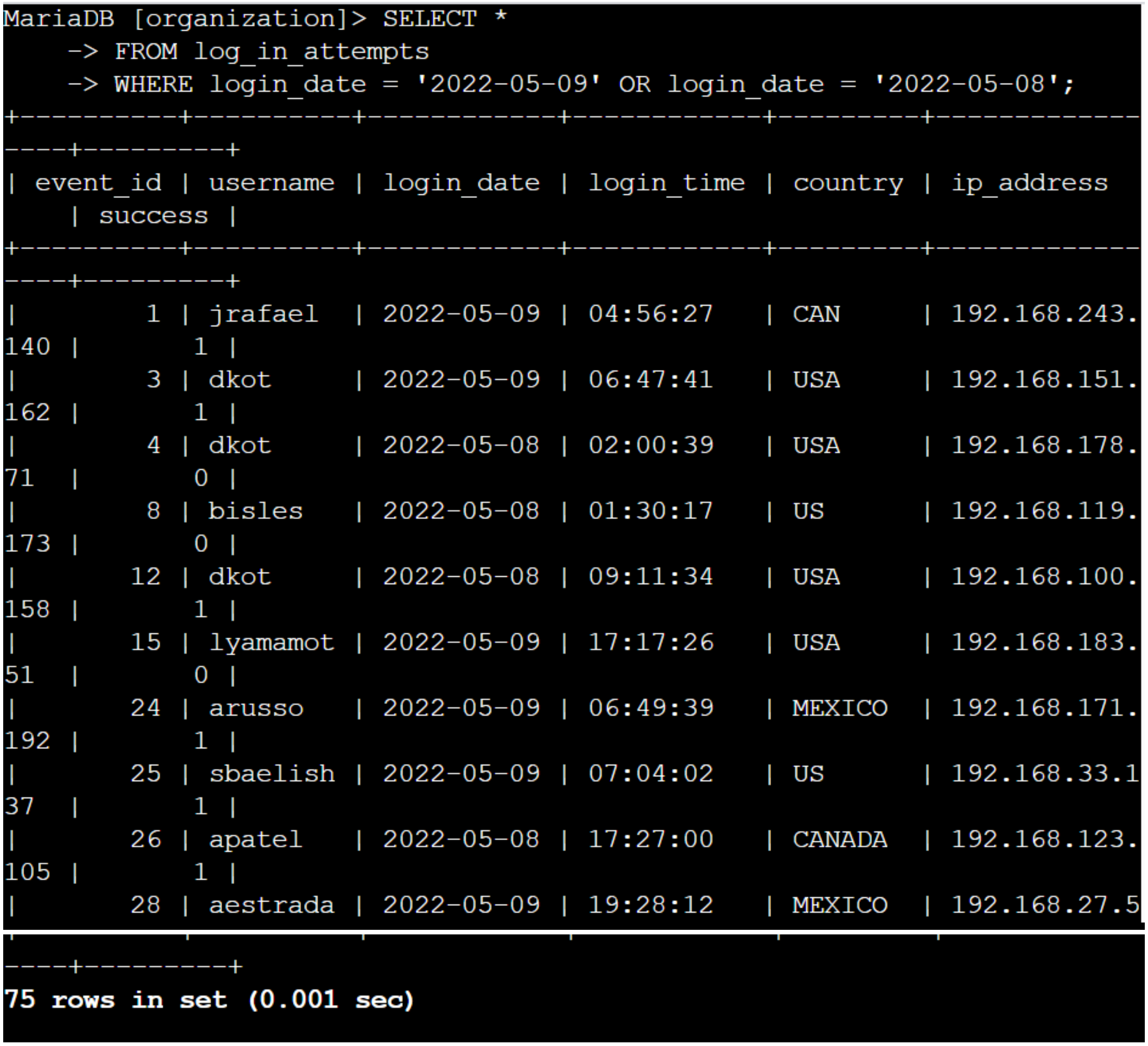
To investigate a security event on **2022-05-09**, I ran an SQL query retrieving all login attempts from both **May 8th and 9th, 2022** using:

SELECT *

FROM log_in_attempts

WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';

The SQL statement selects every entry from the log_in_attempts table which fell between May 8th and May 9th 2022. The analysis reveals that 75 login events happened between May 8th and May 9th whereby users dkot and apatel performed at least two login attempts each. The collected data spans early morning through evening hours (01:30 until 19:28) while including detailed IP address and country information in each recorded attempt. This unfiltered dataset contains all authentication events from successful as well as failed login attempts throughout both selected dates. User activities during these days cover regular operating hours in various time zones based on the diverse login times recorded.



```
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      | 192.168.33.1
37  |       1 |
|       26 | apatel   | 2022-05-08 | 17:27:00   | CANADA  | 192.168.123.
105 |       1 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.5
----+---------+
75 rows in set (0.001 sec)
```

Retrieve login attempts outside of Mexico

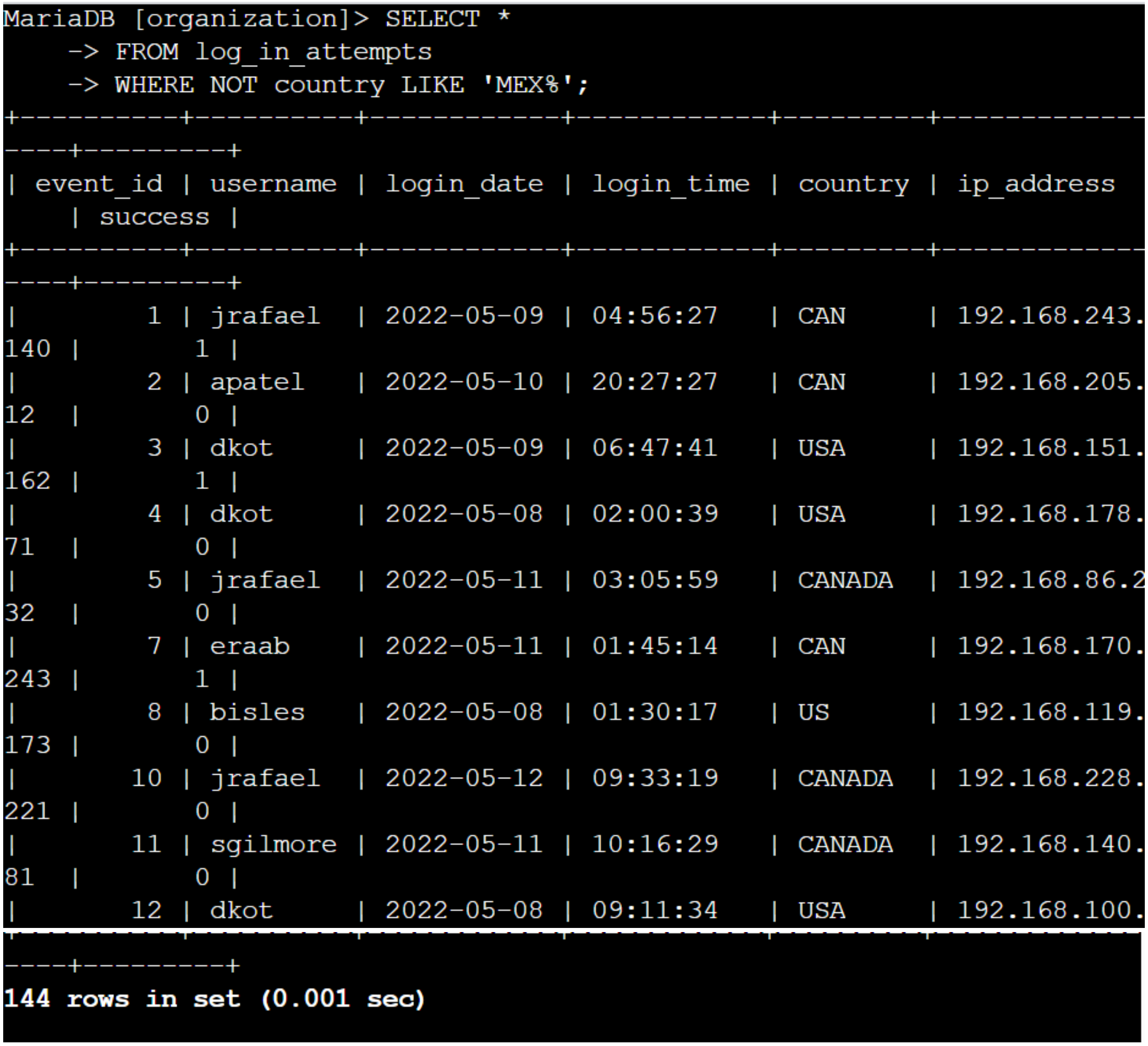
Providing more defined investigation parameters on foreign login activities involved running the following SQL query.

```
SELECT *

FROM log_in_attempts

WHERE NOT country LIKE 'MEX%';
```

The SQL query identifies all login attempts except Mexican ones which it achieves through the condition WHERE NOT country LIKE 'MEX%'. The analyzed data exhibits both successful logins and failed logins that originated from Canada, the US, and several other countries. The produced output contains user authentication patterns showing jrafael from Canada and dkot from the US along with other global users which enables investigators to concentrate on external threats against the Mexican system. Regardless of spelling variation MEX or MEXICO the query excludes all Mexican records through the LIKE operator with wildcard %.



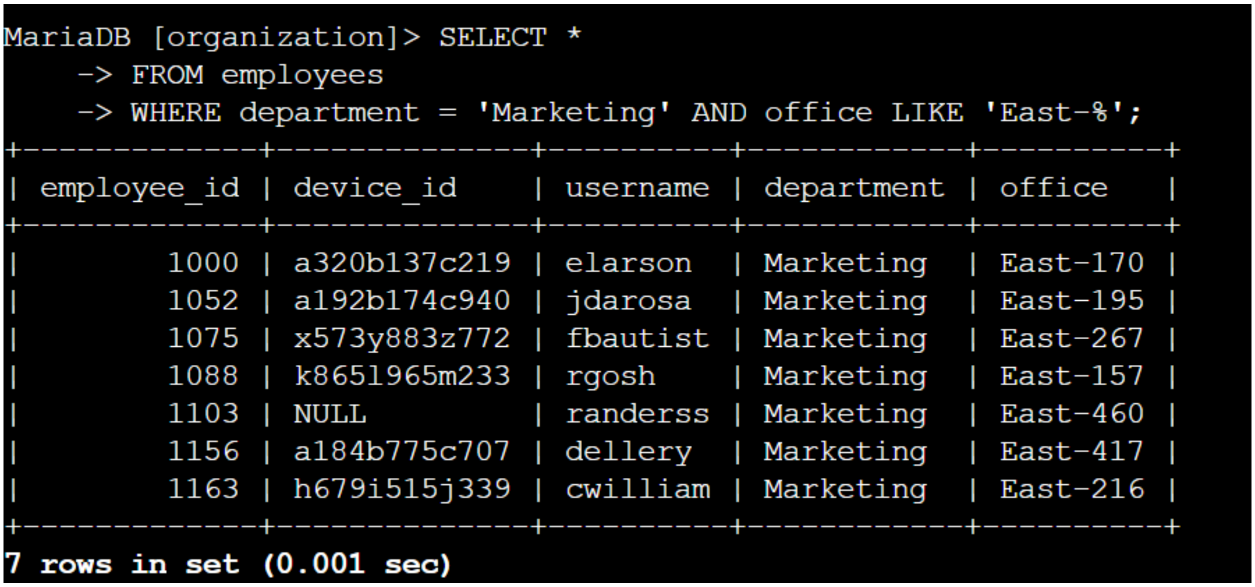
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+------------
----+---------+
| event_id | username | login_date | login_time | country | ip_address
    | success |
+----------+----------+------------+------------+---------+------------
----+---------+
|        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  | 192.168.86.2
32  |       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.
+----------+----------+------------+------------+---------+------------
----+---------+
144 rows in set (0.001 sec)
```

Retrieve employees in Marketing

The required data about Marketing employees who work in East-building offices was retrieved with the following SQL query.

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

The SQL statement selects marketing department workers who hold seats in East-building areas based on department = 'Marketing' and 'office LIKE 'East-%''. This query uses East-% LIKE operator to select all offices starting with East while the AND statement maintains multiple condition requirements. The security update receipt contains seven marketing department staff members who have device IDs assigned to them although one worker's ID is missing. The output shows the distribution pattern of East-building offices from East-170 to East-460 and demonstrates different device ID formats which need specialized update procedures.



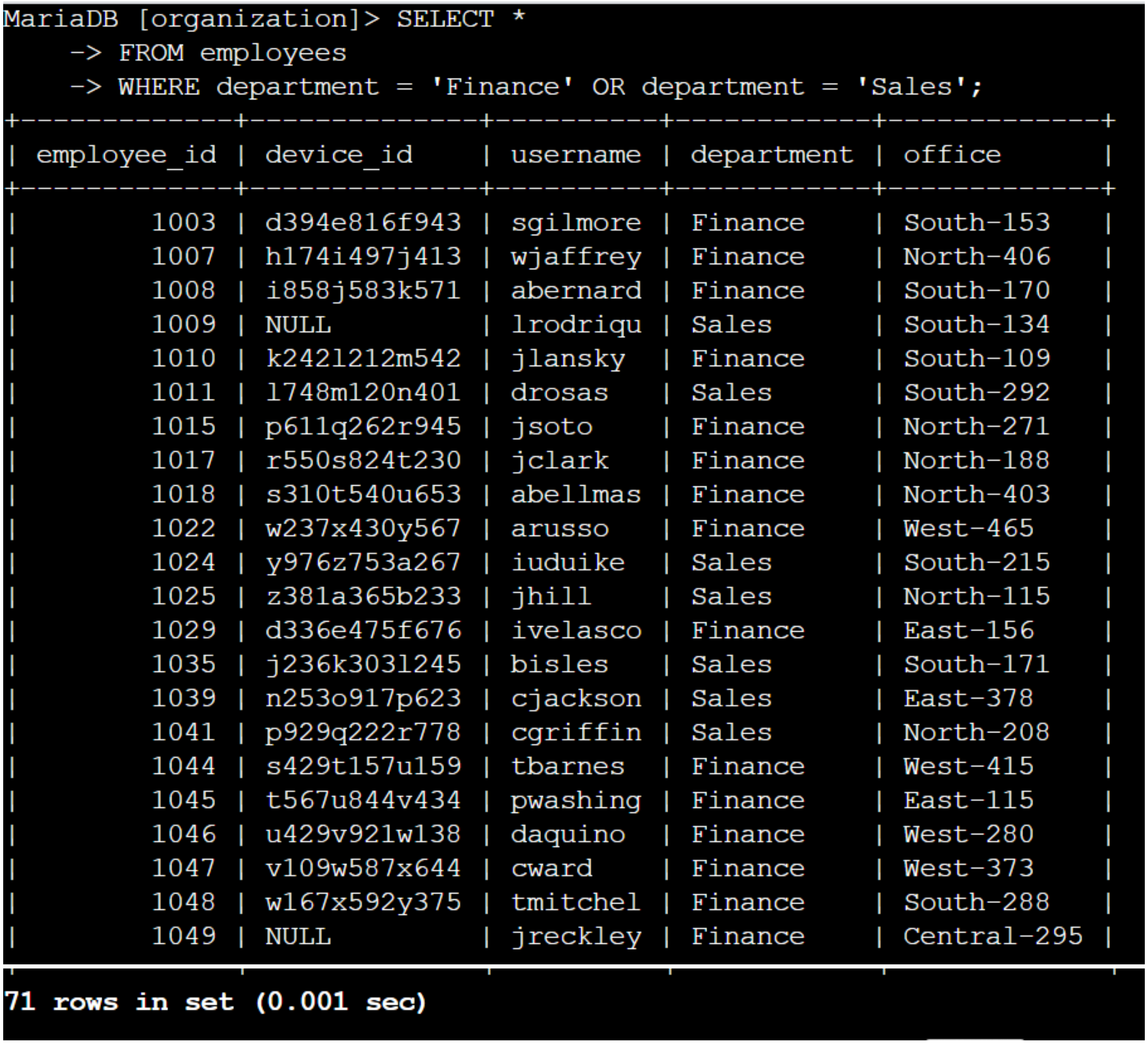
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East-%';
+-------------+-------------+----------+------------+----------+
| employee_id | device_id   | username | department | office   |
+-------------+-------------+----------+------------+----------+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

Retrieve employees in Finance or Sales

The query:
SELECT *
FROM employees
WHERE department = 'Finance' OR department = 'Sales';

The sql statement fetches the entire employee dataset which includes information from the employees table for departments consisting of Finance or Sales. Through its utilization of the OR operator this query brings back all records matching the finance and sales departments. All columns of matching employees are displayed by the SELECT * component within the result including employee IDs and device IDs as well as usernames and office assignments.

The query applies particularly to updating or changing system settings for the machines of personnel in Finance or Sales departments. Through departmental filtering the IT team can rapidly select appropriate user groups while avoiding impact on users from Marketing or Human Resources.



```
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   |
|        1018 | s310t540u653 | abellmas | Finance    | North-403   |
|        1022 | w237x430y567 | arusso   | Finance    | West-465    |
|        1024 | y976z753a267 | iuduike  | Sales      | South-215   |
|        1025 | z381a365b233 | jhill    | Sales      | North-115   |
|        1029 | d336e475f676 | ivelasco | Finance    | East-156    |
|        1035 | j236k303l245 | bisles   | Sales      | South-171   |
|        1039 | n253o917p623 | cjackson | Sales      | East-378    |
|        1041 | p929q222r778 | cgriffin | Sales      | North-208   |
|        1044 | s429t157u159 | tbarnes  | Finance    | West-415    |
|        1045 | t567u844v434 | pwashing | Finance    | East-115    |
|        1046 | u429v921w138 | daquino  | Finance    | West-280    |
|        1047 | v109w587x644 | cward    | Finance    | West-373    |
|        1048 | w167x592y375 | tmitchel | Finance    | South-288   |
|        1049 | NULL         | jreckley | Finance    | Central-295 |

71 rows in set (0.001 sec)
```

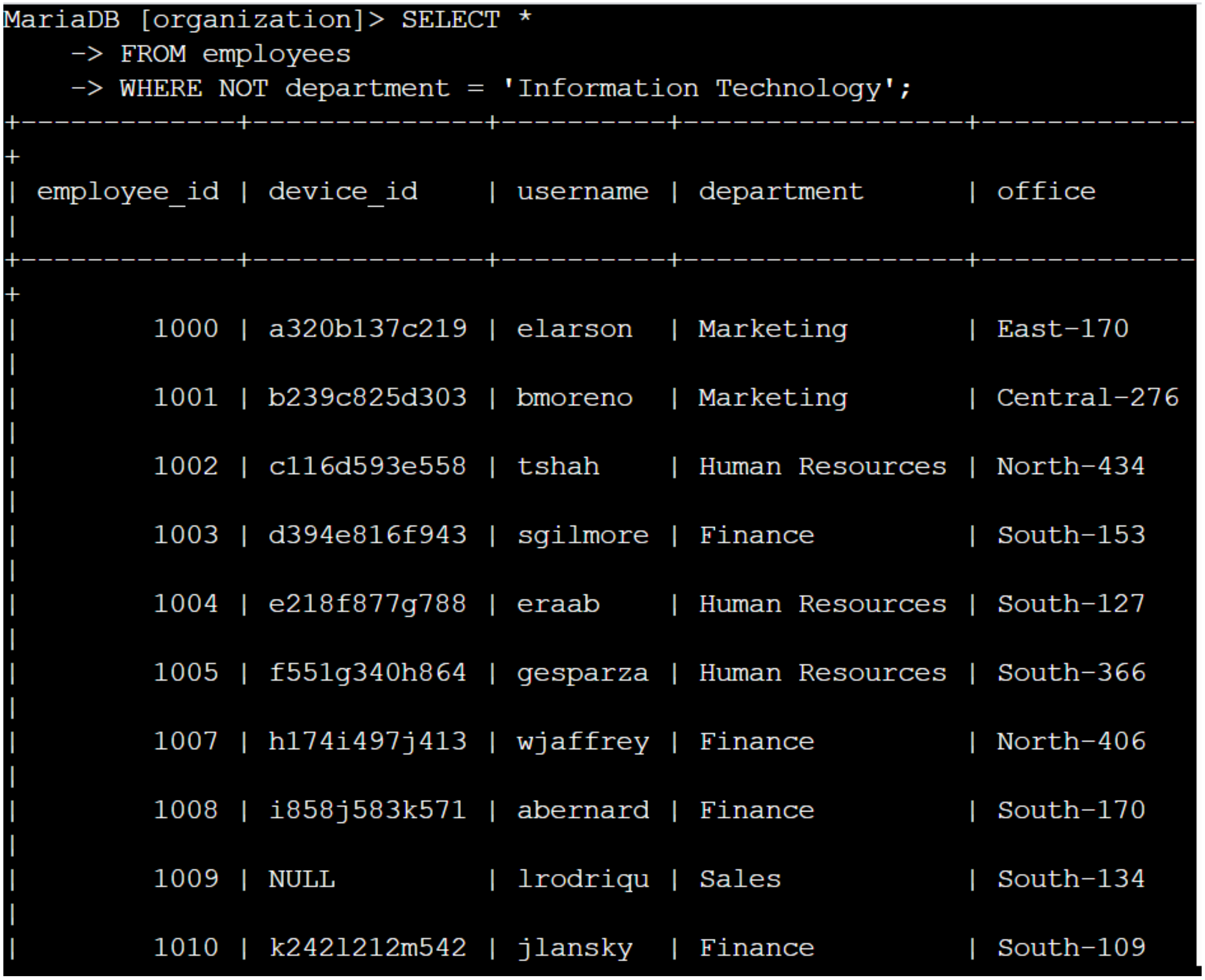
Retrieve all employees not in IT

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

The SQL request selects every record from employees that does not belong to the 'Information Technology' department. The NOT operator applies exclusivity to rows when matching conditions to eliminate data containing 'Information Technology' department.

The query contains the condition WHERE NOT department = 'Information Technology' that selects all employees from departments besides Information Technology. The query provides benefit when updating departments but excluding Information Technology which has already received its necessary changes.

The query searched for 161 devices which belong to staff members outside IT who require the update software.



```
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
```

Summary

The presented analysis uses SQL queries to study login patterns and employee records that helps with system security updates. The incident shows 19 failed login attempts that occurred during after-hours operation from Mexico as well as Canada and the United States by users *apatel* and *pwashing*. Insight from May 8 and 9 login activity revealed 75 different events that showed successful and unsuccessful attempts thus tracing abnormal user actions. A different query excluded Mexican login attempts in order to identify user activities coming from international locations. The update management process used individual queries to select Marketing workers in East-building space, full Finance and Sales divisions and every staff member outside of IT. The related queries verify correct deployment of updates to staff members while keeping protected departments and users who have already received the updates.