Apply filters to SQL queries

Project description

I am a security professional at a large organization. Part of my job is to investigate security issues to help keep the system secure. I recently discovered some potential security issues that involve login attempts and employee machines.

My task is to examine the organization's data in their employees and log_in_attempts tables. I'll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues.

Retrieve after hours failed login attempts.

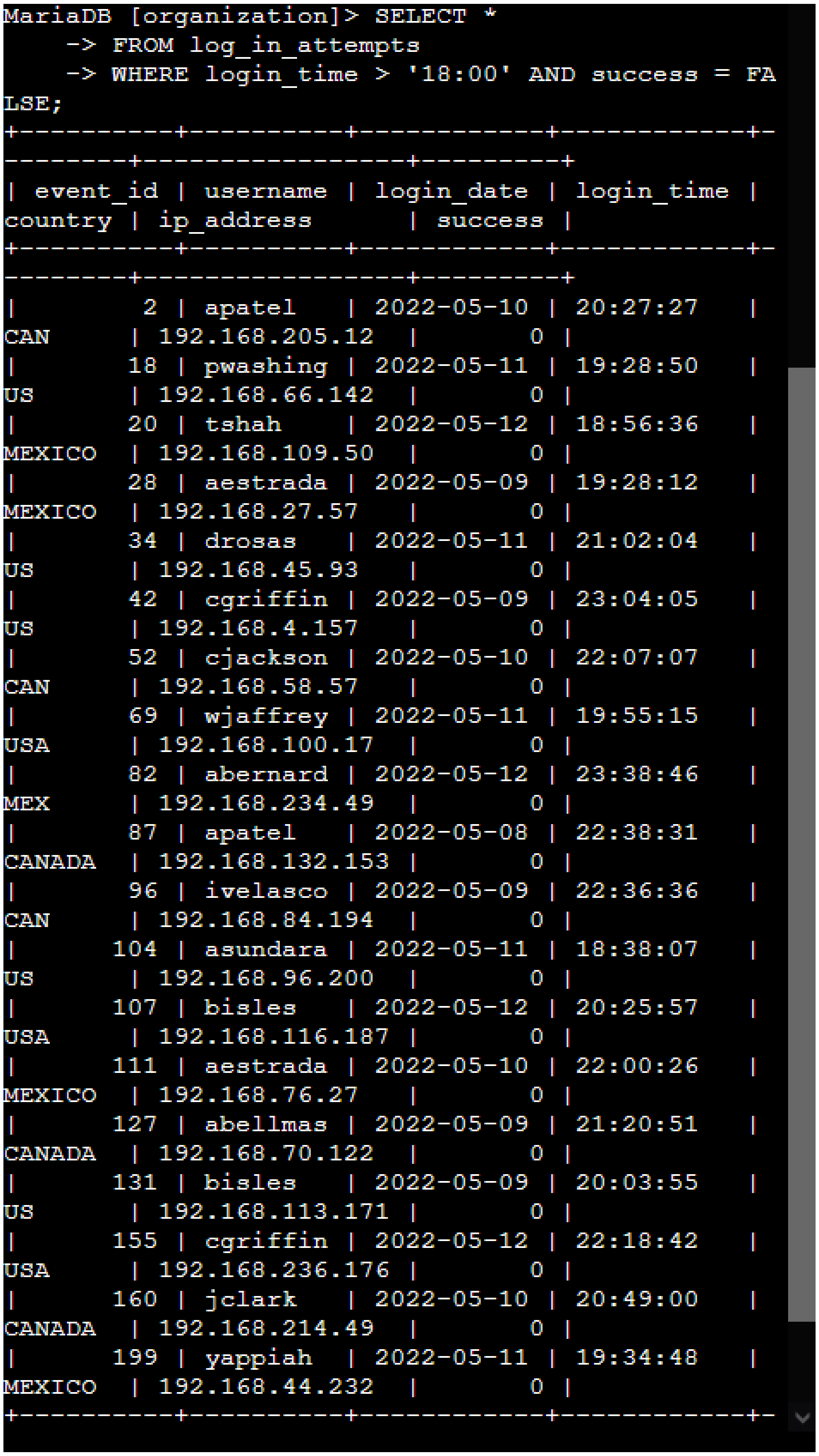
I am investigating failed login attempts made after business hours. I need to extract this information from the login activity, specifically identifying all unsuccessful attempts after 18:00.

The log_in_attempts table contains a login_time column that provides information on when login attempts were made. Our office hours conclude at '18:00'.

The success column in the log_in_attempts table contains values of TRUE or FALSE, indicating the success or failure of the login attempts. In MySQL, Boolean values are represented as 1 for TRUE and 0 for FALSE. Therefore, I will use the AND operator to retrieve the failed login attempts that occurred after business hours. It's important to note that values of TRUE and FALSE are not placed in single quotes since they are not string data but rather Boolean data.

I am selecting all records from the log_in_attempts table where the login_time is later than '18:00' and the login attempt was unsuccessful (success = FALSE).

This SQL query aims to retrieve information about failed login attempts that occurred after business hours, which end at '18:00'. The conditions in the WHERE clause ensure that only records meeting both criteria (login_time after '18:00' and unsuccessful login) will be included in the result set. The asterisk (*) is used as a wildcard to indicate that all columns should be included in the output.

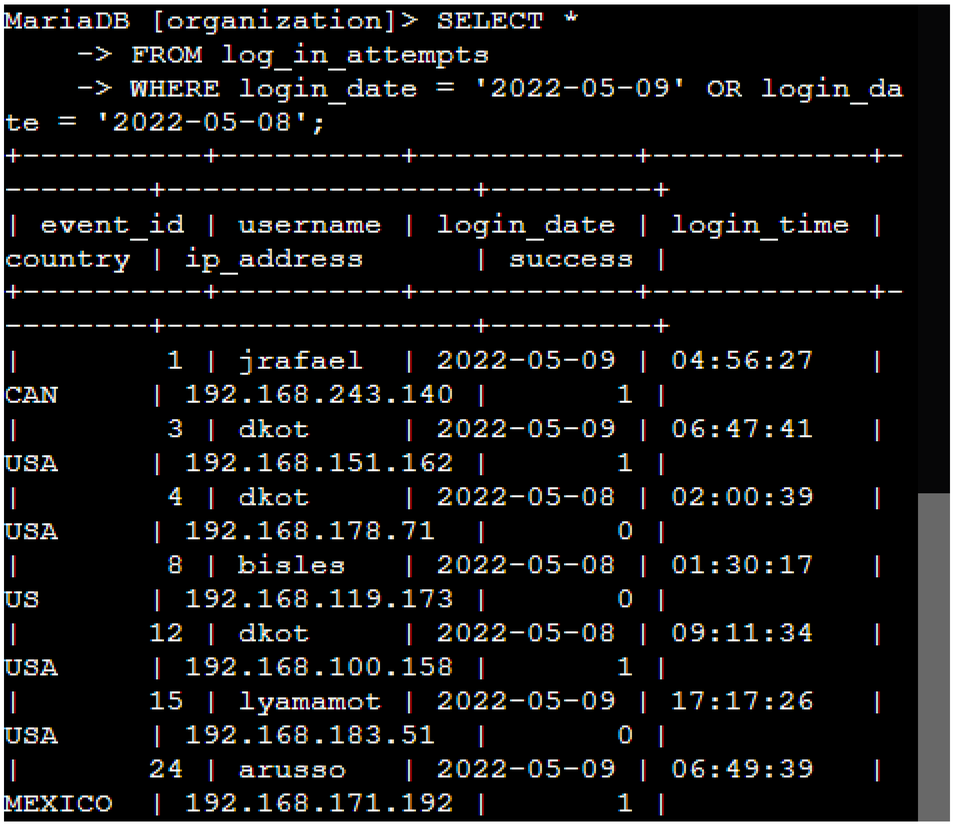


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FA
LSE;
+----------+----------+------------+------------+-
--------+-----------------+---------+
| event_id | username | login_date | login_time |
country | ip_address      | success |
+----------+----------+------------+------------+-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

Retrieve login attempts on specific dates.

I am investigating a suspicious event that occurred on '2022-05-09', and I need to retrieve all login attempts on this day and the day before ('2022-05-08').

The log_in_attempts table contains a login_date column, which provides information on the dates when login attempts were made.



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

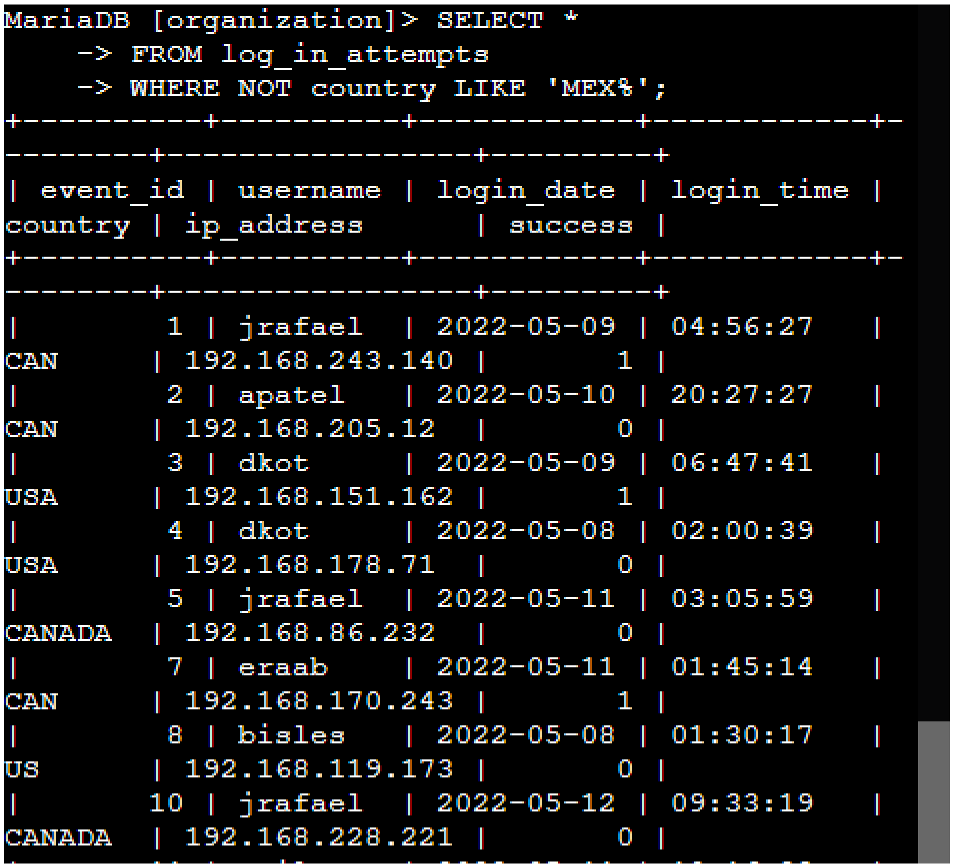
To achieve this, I will use the OR operator to retrieve the failed login attempts on the specified days. This allows me to combine the conditions for '2022-05-09' and '2022-05-08' in order to gather relevant information regarding the suspicious event.

I am selecting all records from the log_in_attempts table where the login_date is either '2022-05-09' or '2022-05-08'.

This SQL query is designed to retrieve login attempts that occurred on the specified dates, '2022-05-09' and the day before, '2022-05-08'. The OR operator is used to combine these conditions, ensuring that records matching either date are included in the result set. The asterisk (*) is employed as a wildcard to include all columns in the output.

Retrieve login attempts outside of Mexico

I am selecting all records from the log_in_attempts table where the country field does not start with 'MEX'. This is achieved by using the NOT operator in combination with the LIKE operator and the matching pattern 'MEX%'.



```
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 |
```

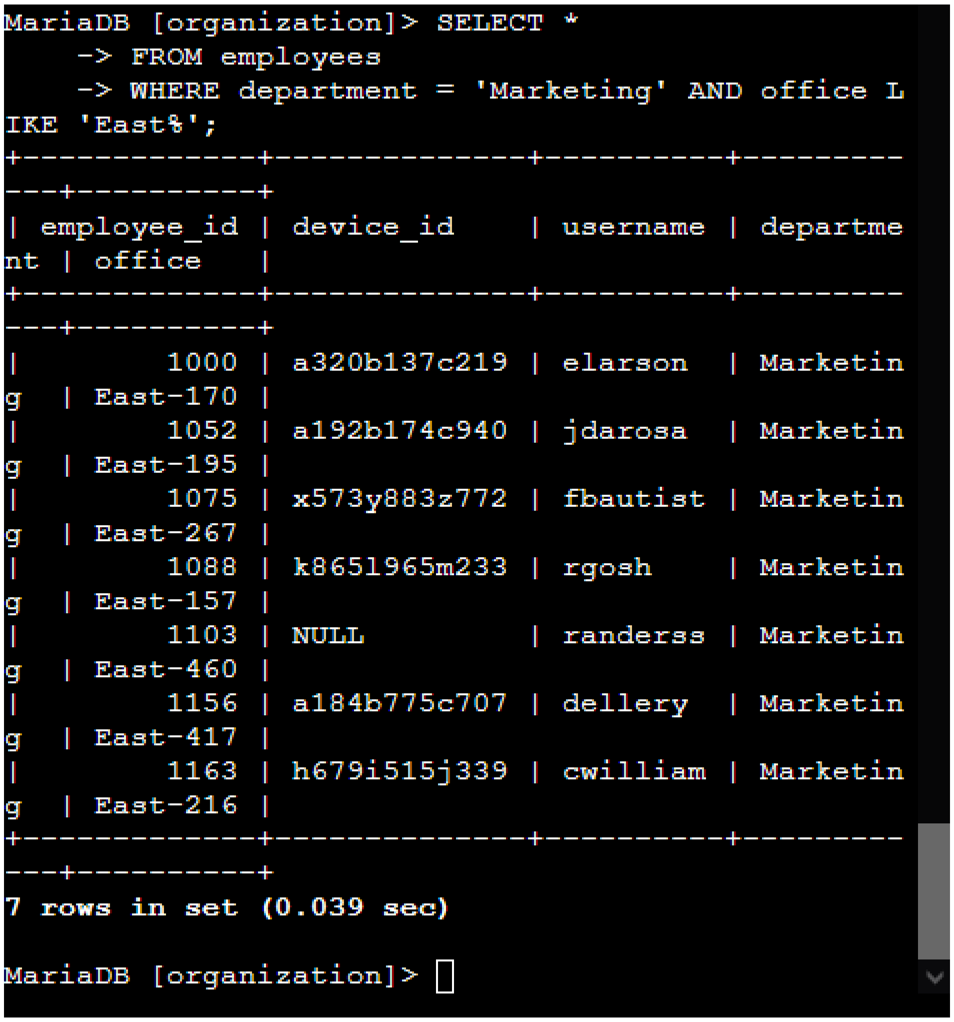
The SQL query aims to identify login attempts that did not originate in Mexico. The condition in the WHERE clause ensures that only records with country values not starting with 'MEX' are included in the result set. This accounts for entries with both 'MEX' and 'MEXICO' in the country field.

Retrieve employees in Marketing

I am tasked with retrieving information from the 'department' and 'office' columns in the employees table. To view the columns and values in the employees table, I can execute the SQL query:

SELECT * FROM employees;

As my team is currently updating employee machines, I need to gather information about employees in the 'Marketing' department who are situated in all offices within the East building, such as 'East-170' or 'East-320'. To achieve this, I am selecting all columns from the employees table where the department is 'Marketing' and the office starts with 'East'.



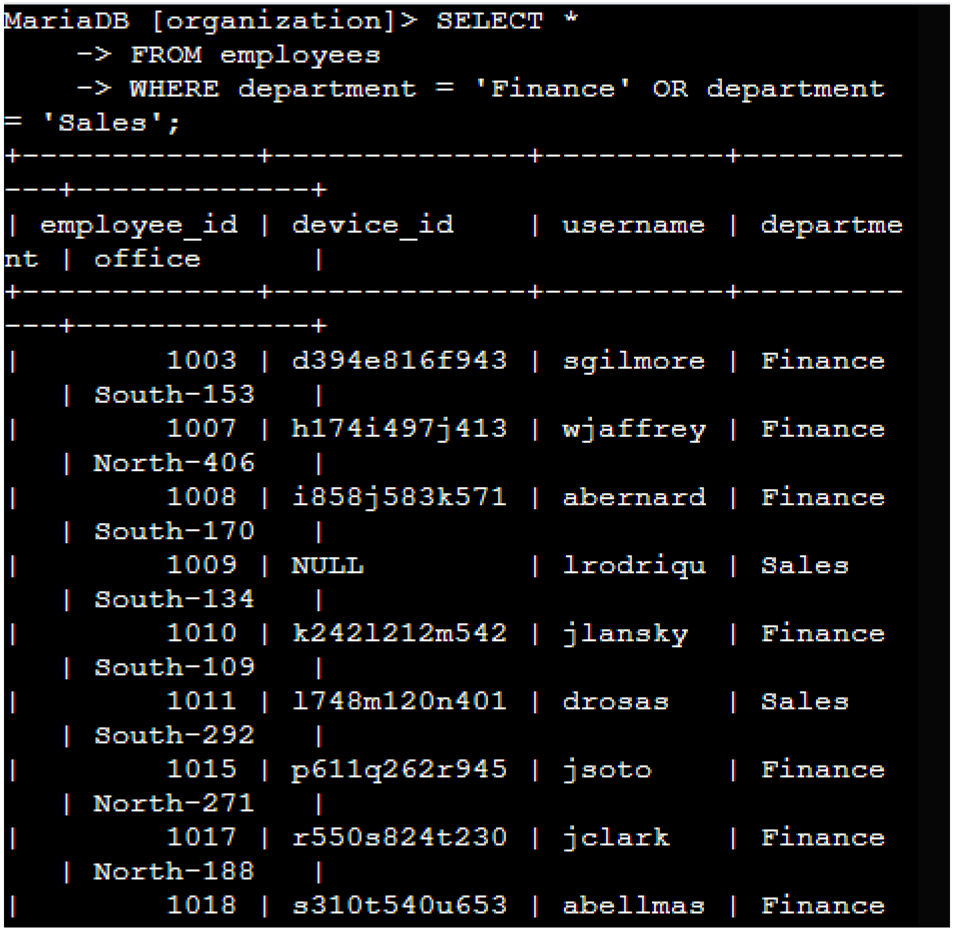
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office L
IKE 'East%';
+-------------+---------------+----------+---------
---+-----------+
| employee_id | device_id     | username | departme
nt | office    |
+-------------+---------------+----------+---------
---+-----------+
|        1000 | a320b137c219  | elarson  | Marketin
g  | East-170  |
|        1052 | a192b174c940  | jdarosa  | Marketin
g  | East-195  |
|        1075 | x573y883z772  | fbautist | Marketin
g  | East-267  |
|        1088 | k8651965m233  | rgosh    | Marketin
g  | East-157  |
|        1103 | NULL          | randerss | Marketin
g  | East-460  |
|        1156 | a184b775c707  | dellery  | Marketin
g  | East-417  |
|        1163 | h679i515j339  | cwilliam | Marketin
g  | East-216  |
+-------------+---------------+----------+---------
---+-----------+
7 rows in set (0.039 sec)

MariaDB [organization]>
```

This SQL query is designed to retrieve information about employees who belong to the 'Marketing' department and are located in offices within the East building, such as 'East-170' or 'East-320'. The LIKE operator with the pattern 'East%' ensures that offices starting with 'East' are included, and the AND operator combines this condition with the requirement for the 'Marketing' department. The result set will contain all relevant details for these specified criteria.

Retrieve employees in Finance or Sales

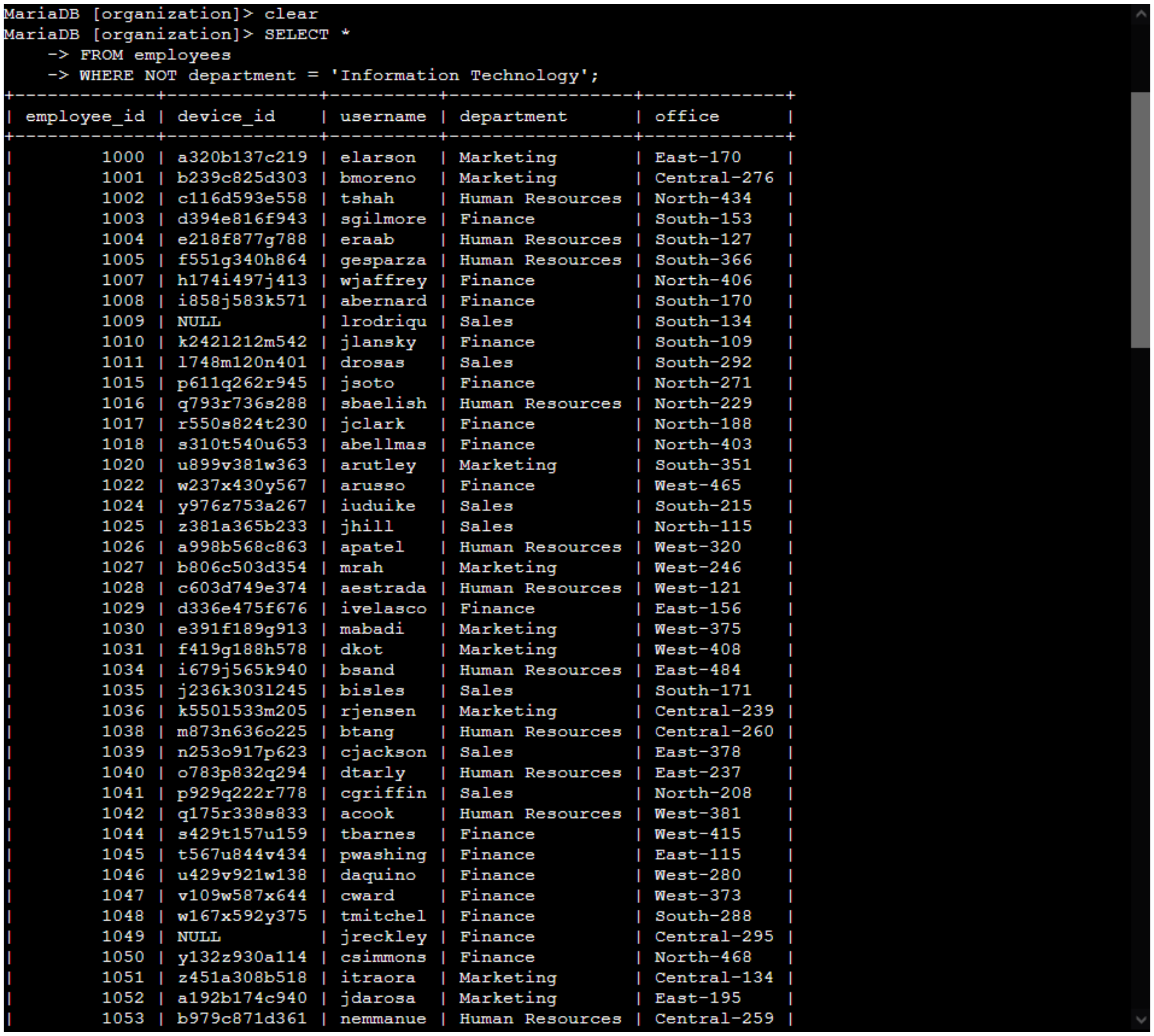
I am constructing a SQL query to retrieve records for employees in either the 'Finance' or the 'Sales' department. Despite both conditions being based on the same column ('department'), it is necessary to explicitly specify the column in both conditions.



```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department
= 'Sales';
+-------------+-------------+----------+---------
---+--------------+
| employee_id | device_id    | username | departme
nt | 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
```

Retrieve all employees not in IT

I am tasked with retrieving information about employees who are not in the Information Technology department. To accomplish this, I will use the NOT operator in a SQL query.



```
MariaDB [organization]> clear
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 | jsoto    | Finance         | North-271   |
|        1016 | q793r736s288 | sbaelish | Human Resources | North-229   |
|        1017 | r550s824t230 | jclark   | Finance         | North-188   |
|        1018 | s310t540u653 | abellmas | Finance         | North-403   |
|        1020 | u899v381w363 | arutley  | Marketing       | South-351   |
|        1022 | w237x430y567 | arusso   | Finance         | West-465    |
|        1024 | y976z753a267 | iuduike  | Sales           | South-215   |
|        1025 | z381a365b233 | jhill    | Sales           | North-115   |
|        1026 | a998b568c863 | apatel   | Human Resources | West-320    |
|        1027 | b806c503d354 | mrah     | Marketing       | West-246    |
|        1028 | c603d749e374 | aestrada | Human Resources | West-121    |
|        1029 | d336e475f676 | ivelasco | Finance         | East-156    |
|        1030 | e391f189g913 | mabadi   | Marketing       | West-375    |
|        1031 | f419g188h578 | dkot     | Marketing       | West-408    |
|        1034 | i679j565k940 | bsand    | Human Resources | East-484    |
|        1035 | j236k3031245 | bisles   | Sales           | South-171   |
|        1036 | k5501533m205 | rjensen  | Marketing       | Central-239 |
|        1038 | m873n636o225 | btang    | Human Resources | Central-260 |
|        1039 | n253o917p623 | cjackson | Sales           | East-378    |
|        1040 | o783p832q294 | dtarly   | Human Resources | East-237    |
|        1041 | p929q222r778 | cgriffin | Sales           | North-208   |
|        1042 | q175r338s833 | acook    | Human Resources | West-381    |
|        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 |
|        1050 | y132z930a114 | csimmons | Finance         | North-468   |
|        1051 | z451a308b518 | itraora  | Marketing       | Central-134 |
|        1052 | a192b174c940 | jdarosa  | Marketing       | East-195    |
|        1053 | b979c871d361 | nemmanue | Human Resources | Central-259 |
```

I am selecting all columns from the employees table where the department is not equal to 'Information Technology'. This SQL query is designed to retrieve information about employees who do not belong to the Information Technology department. The NOT operator is used to negate the condition, ensuring that only records with departments other than 'Information Technology' are included in the result set.

Summary

In the role of a security analyst, the analysis of data is a frequent requirement. Effectively retrieving specific information from the database often involves considering multiple factors. In utilizing SQL, I have gained practical experience in:

- Executing SQL queries to extract information from a database.
- Applying AND, OR, and NOT operators to refine SQL queries, allowing for more intricate and precise filtering of data.