# Apply filters to SQL queries

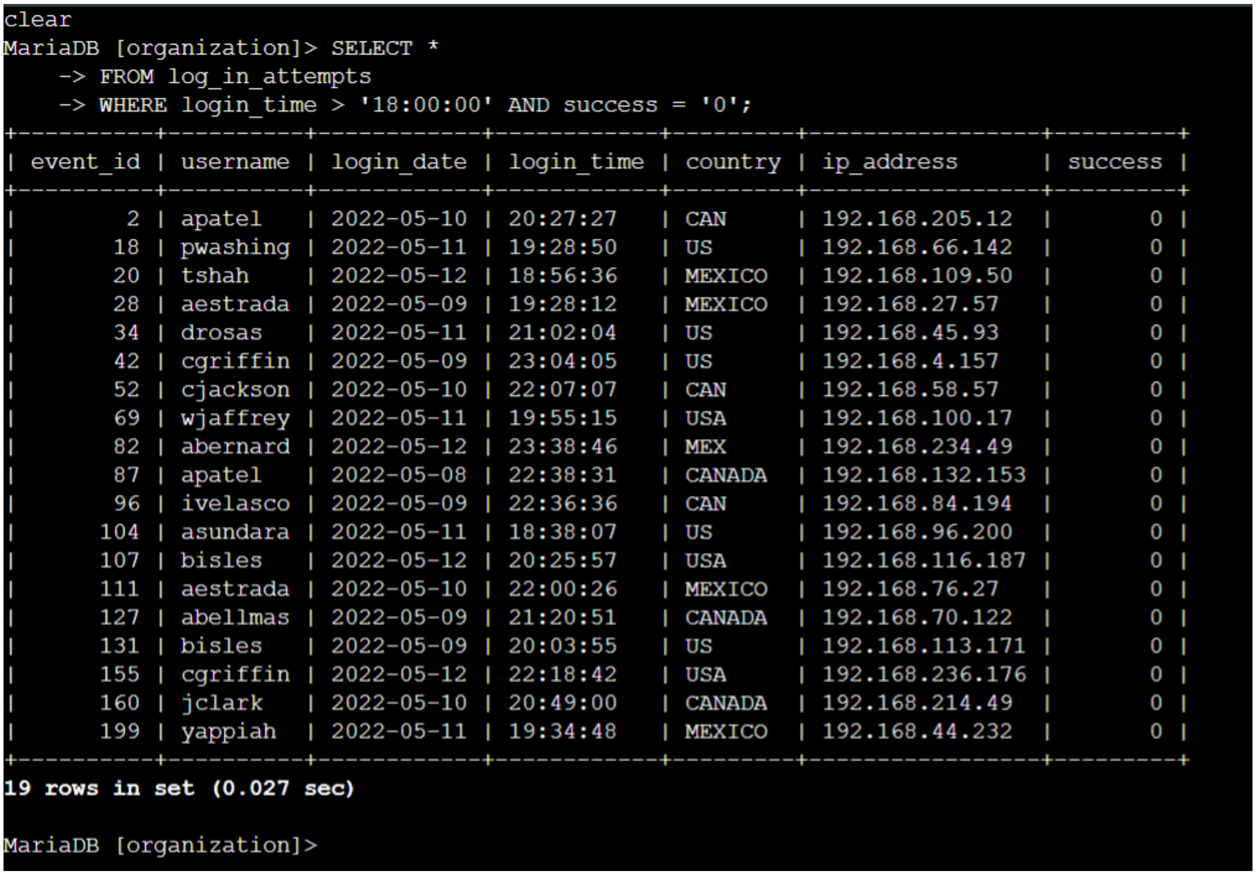
## Project description

I’m a Security Professional at the organization.  My job is to ensure the secure and investigate potential security issues. Below are examples of how SQL queries with filters were used to support security investigation.

## Retrieve after hours failed login attempts

I discovered a potential security that occurred after 18:00 after business hours. I need to investigate this as soon as possible.

This SQL query demonstrates the use of filters to detect failed login attempts that occurred after normal business hours :



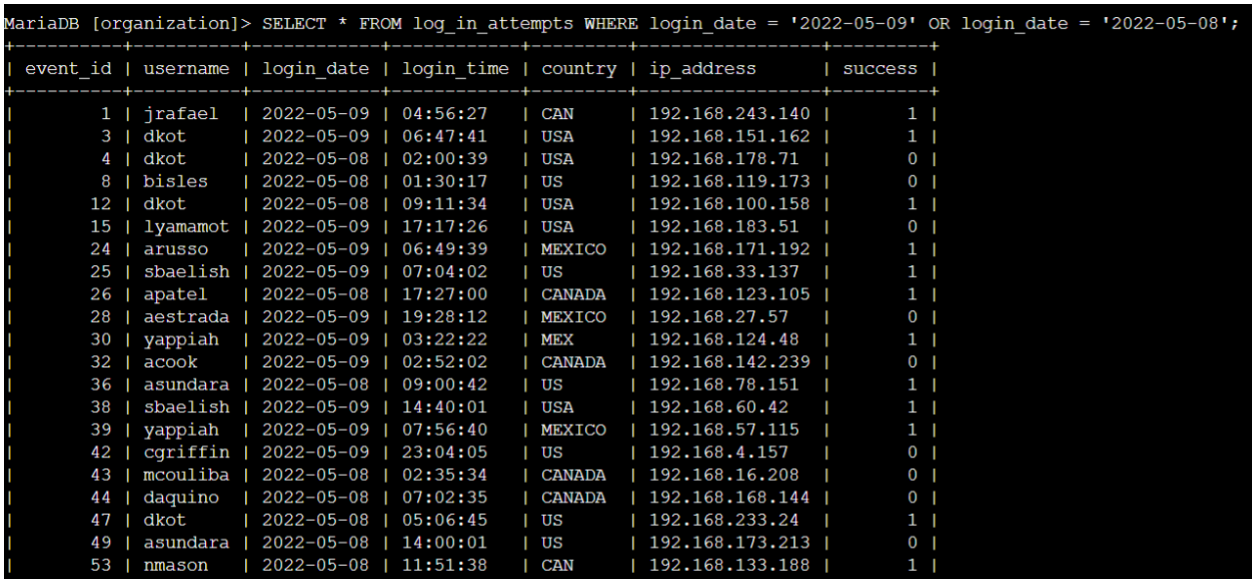
```
clear
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00:00' AND success = '0';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 (0.027 sec)

MariaDB [organization]>
```

On the command line, I queried all data from log_in_attempts table and I used WHERE command with an AND operator. To filter login_time after 18:00:00. The success command contains a value of 0 when login attempt failed, but actually i can use ‘0’ or ‘False’ in the query to identify failed login attempts.

## Retrieve login attempts on specific dates

I need to investigate a suspicious event occurred on 2022-05-09 and i want to review all login attempts which occurred at that time and the day before or 2022-05-09.

This SQL query demonstrates the use of filters to detect failed login attempts that occurred at 2022-05-09 and the day before :



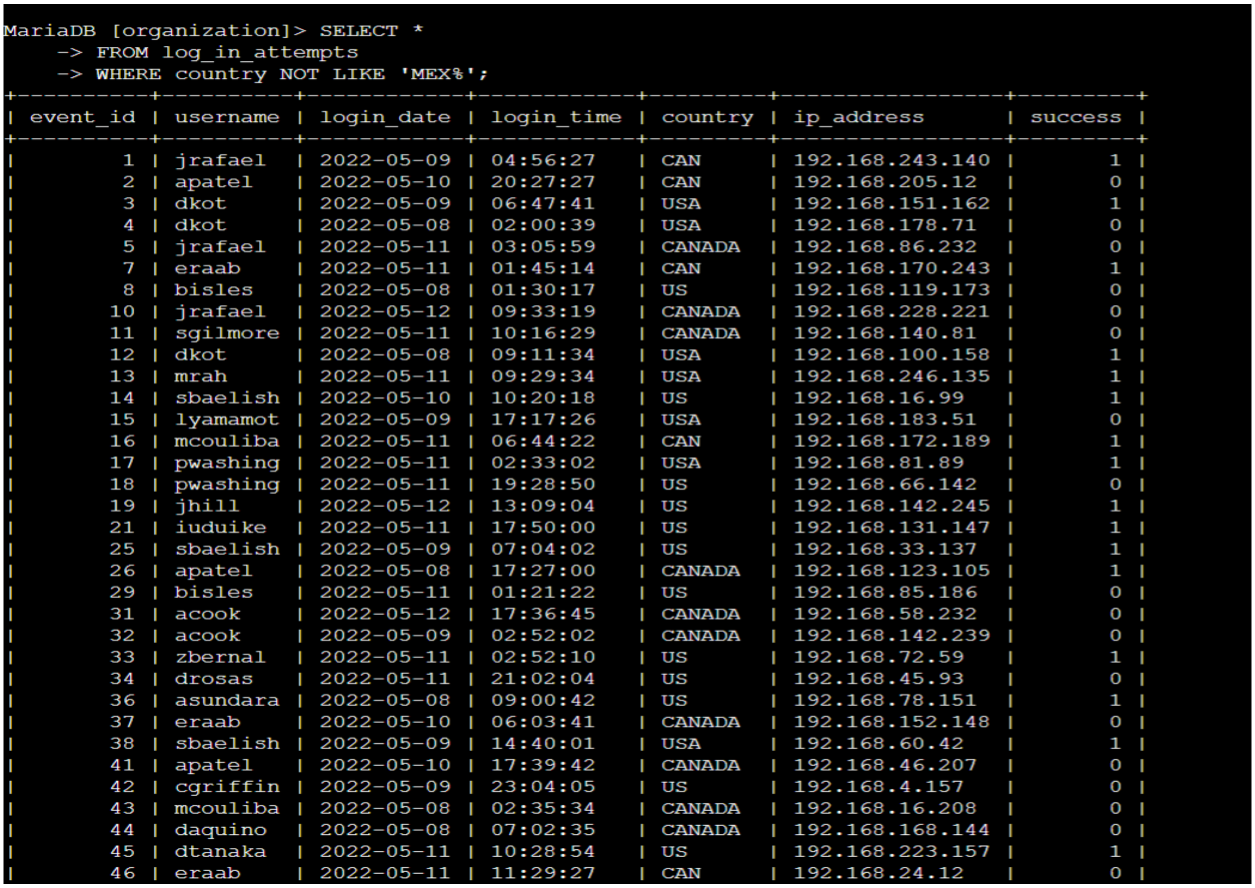
```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|       36 | asundara | 2022-05-08 | 09:00:42   | US      | 192.168.78.151  |       1 |
|       38 | sbaelish | 2022-05-09 | 14:40:01   | USA     | 192.168.60.42   |       1 |
|       39 | yappiah  | 2022-05-09 | 07:56:40   | MEXICO  | 192.168.57.115  |       1 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.157   |       0 |
|       43 | mcouliba | 2022-05-08 | 02:35:34   | CANADA  | 192.168.16.208  |       0 |
|       44 | daquino  | 2022-05-08 | 07:02:35   | CANADA  | 192.168.168.144 |       0 |
|       47 | dkot     | 2022-05-08 | 05:06:45   | US      | 192.168.233.24  |       1 |
|       49 | asundara | 2022-05-08 | 14:00:01   | US      | 192.168.173.213 |       0 |
|       53 | nmason   | 2022-05-08 | 11:51:38   | CAN     | 192.168.133.188 |       1 |
```

On the command line, I queried all data from log_in_attempts table and . I used the WHERE command and an OR operator. To filter login_date on 2022-05-09 or 2022-05-08.

## Retrieve login attempts outside of Mexico

There’s been suspicious activity with login attempts, but the team has determined this activity didn’t originate from Mexico. Now I need to investigate login attempt that occurred outside of Mexico.

This SQL query demonstrates the use of filters to detect failed login attempts that occurred outside of Mexico :



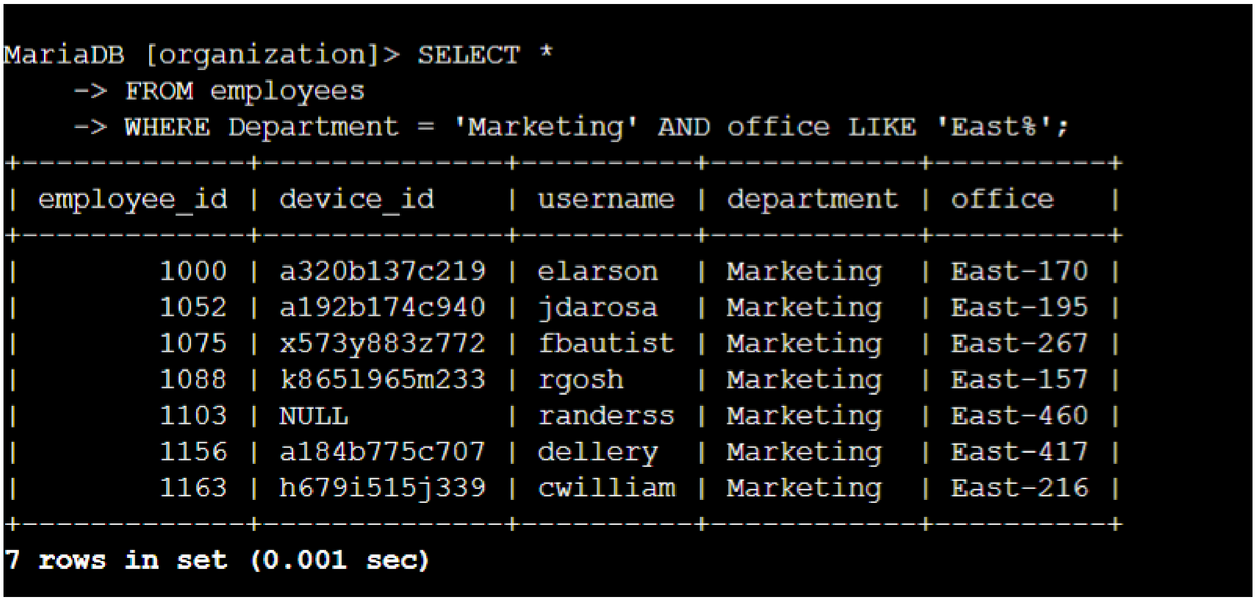
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE country NOT LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|       16 | mcouliba | 2022-05-11 | 06:44:22   | CAN     | 192.168.172.189 |       1 |
|       17 | pwashing | 2022-05-11 | 02:33:02   | USA     | 192.168.81.89   |       1 |
|       18 | pwashing | 2022-05-11 | 19:28:50   | US      | 192.168.66.142  |       0 |
|       19 | jhill    | 2022-05-12 | 13:09:04   | US      | 192.168.142.245 |       1 |
|       21 | iuduike  | 2022-05-11 | 17:50:00   | US      | 192.168.131.147 |       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 |
|       29 | bisles   | 2022-05-11 | 01:21:22   | US      | 192.168.85.186  |       0 |
|       31 | acook    | 2022-05-12 | 17:36:45   | CANADA  | 192.168.58.232  |       0 |
|       32 | acook    | 2022-05-09 | 02:52:02   | CANADA  | 192.168.142.239 |       0 |
|       33 | zbernal  | 2022-05-11 | 02:52:10   | US      | 192.168.72.59   |       1 |
|       34 | drosas   | 2022-05-11 | 21:02:04   | US      | 192.168.45.93   |       0 |
|       36 | asundara | 2022-05-08 | 09:00:42   | US      | 192.168.78.151  |       1 |
|       37 | eraab    | 2022-05-10 | 06:03:41   | CANADA  | 192.168.152.148 |       0 |
|       38 | sbaelish | 2022-05-09 | 14:40:01   | USA     | 192.168.60.42   |       1 |
|       41 | apatel   | 2022-05-10 | 17:39:42   | CANADA  | 192.168.46.207  |       0 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.157   |       0 |
|       43 | mcouliba | 2022-05-08 | 02:35:34   | CANADA  | 192.168.16.208  |       0 |
|       44 | daquino  | 2022-05-08 | 07:02:35   | CANADA  | 192.168.168.144 |       0 |
|       45 | dtanaka  | 2022-05-11 | 10:28:54   | US      | 192.168.223.157 |       1 |
|       46 | eraab    | 2022-05-11 | 11:29:27   | CAN     | 192.168.24.12   |       0 |
```

On the command line, I queried all data from log_in_attempts and used WHERE command and an NOT operator to filter country outside Mexico and also used LIKE operator to find a word after ‘MEX’.

## Retrieve employees in Marketing

My team wants to perform security updates on specific employee machines in the Marketing department. I’m responsible for getting information on these employee machines and need to query the employees table.

The query below shows how SQL was used to identify employee machines for Marketing staff located in the East building :



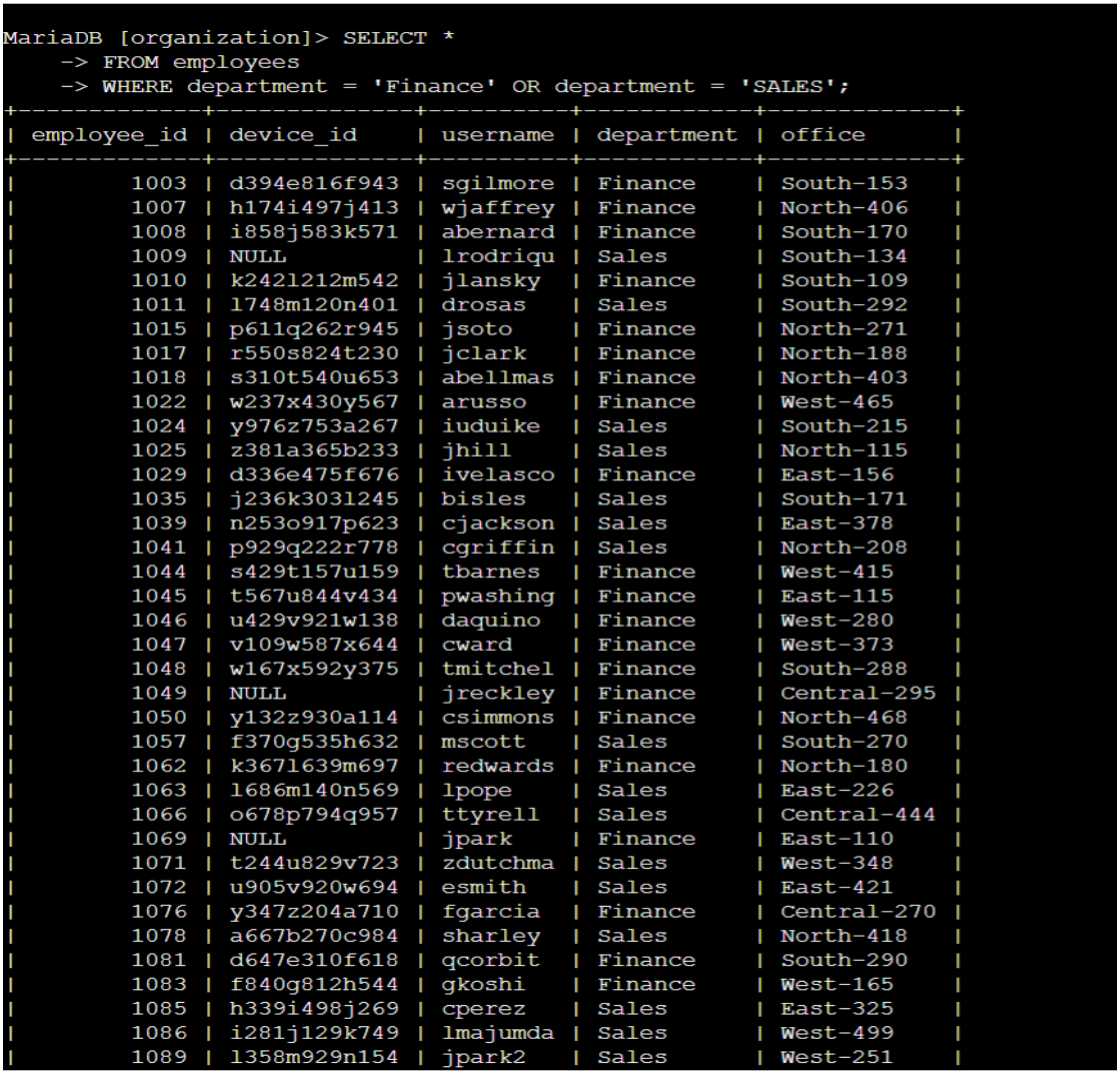
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE Department = 'Marketing' AND office LIKE 'East%';
+-------------+--------------+----------+------------+----------+
| employee_id | device_id    | username | department | office   |
+-------------+--------------+----------+------------+----------+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

On the command line, I queried all data from Employees table and used WHERE command to set the condition, Department for retrieve data from Marketing department, used an AND operator and retrieve data from office used LIKE operator to filter to East building.

## Retrieve employees in Finance or Sales

My team need to perform a different security update on machines for employees in the Sales and Finance departments.

The query below shows how SQL was used to identify employee machines for Finance or Sales :



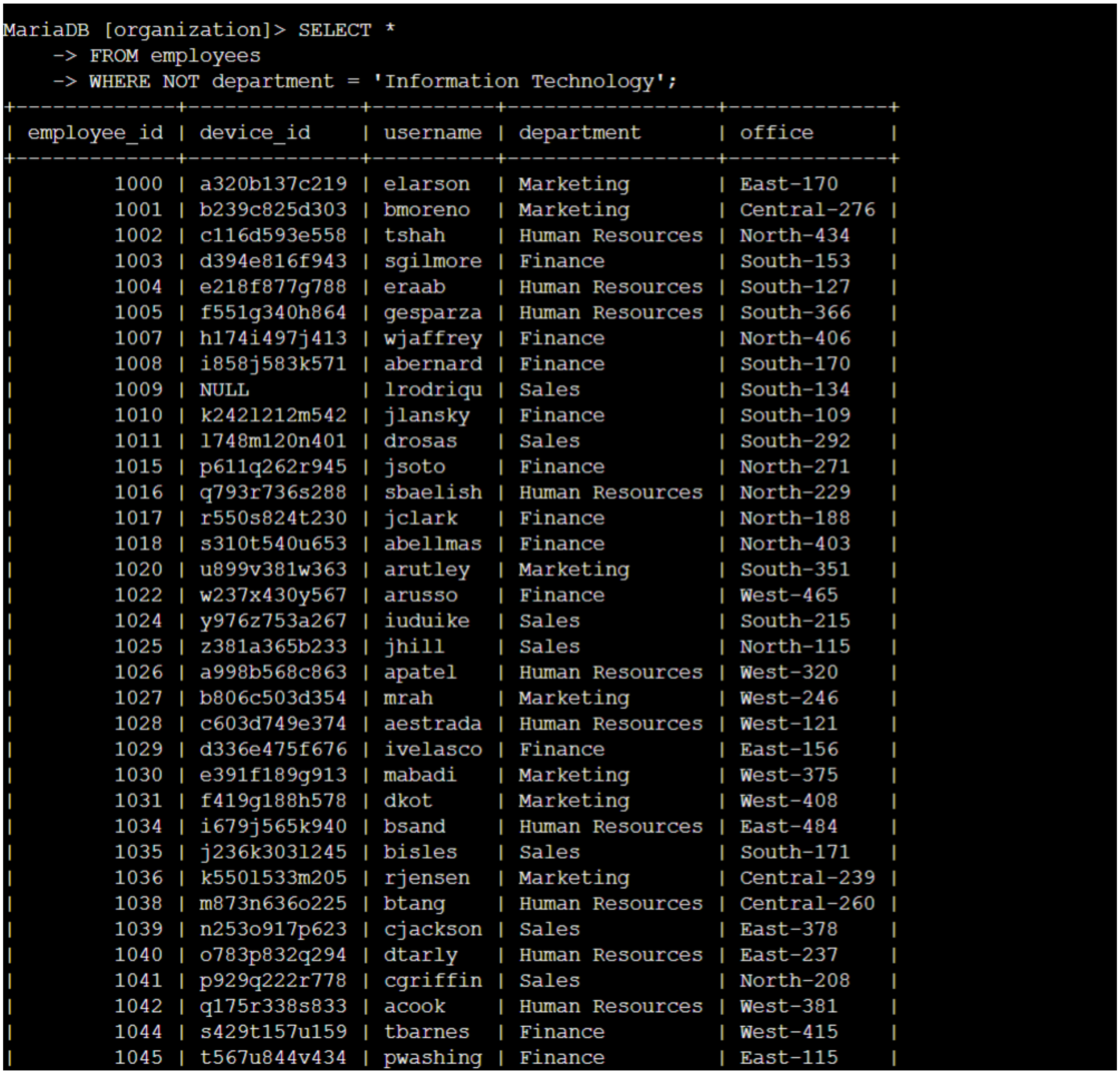
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'SALES';
+-------------+---------------+----------+-------------+---------------+
| employee_id | device_id     | username | department  | office        |
+-------------+---------------+----------+-------------+---------------+
|        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 | 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     |
|        1029 | d336e475f676  | ivelasco | Finance     | East-156      |
|        1035 | j236k3031245  | 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   |
|        1050 | y132z930a114  | csimmons | Finance     | North-468     |
|        1057 | f370g535h632  | mscott   | Sales       | South-270     |
|        1062 | k3671639m697  | redwards | Finance     | North-180     |
|        1063 | 1686m140n569  | lpope    | Sales       | East-226      |
|        1066 | o678p794q957  | ttyrell  | Sales       | Central-444   |
|        1069 | NULL          | jpark    | Finance     | East-110      |
|        1071 | t244u829v723  | zdutchma | Sales       | West-348      |
|        1072 | u905v920w694  | esmith   | Sales       | East-421      |
|        1076 | y347z204a710  | fgarcia  | Finance     | Central-270   |
|        1078 | a667b270c984  | sharley  | Sales       | North-418     |
|        1081 | d647e310f618  | qcorbit  | Finance     | South-290     |
|        1083 | f840g812h544  | gkoshi   | Finance     | West-165      |
|        1085 | h339i498j269  | cperez   | Sales       | East-325      |
|        1086 | i281j129k749  | lmajumda | Sales       | West-499      |
|        1089 | 1358m929n154  | jpark2   | Sales       | West-251      |
```

On the command line, I queried  all data from employees table, used WHERE command to set the condition, used OR operator and Department for retrieve data from Finance department or Sales.

## Retrieve all employees not in IT

My team needs to make one more update to employee machines. The employees who are not in the Information Technology department already had this update, but employees in all other departments need it.

The query below shows how SQL was used to identify employees not from Information Technology :



```
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    |
```

On the command line, I queried all data from employees, used WHERE to set the condition, used an NOT operator to filter department outside of Information Technology.

## Summary

I analyzed login activity and employee device data by querying the `log_in_attempts` and `employees` tables. To narrow down results, I combined multiple conditions using logical operators and pattern matching, ensuring only relevant records were returned for each security task.