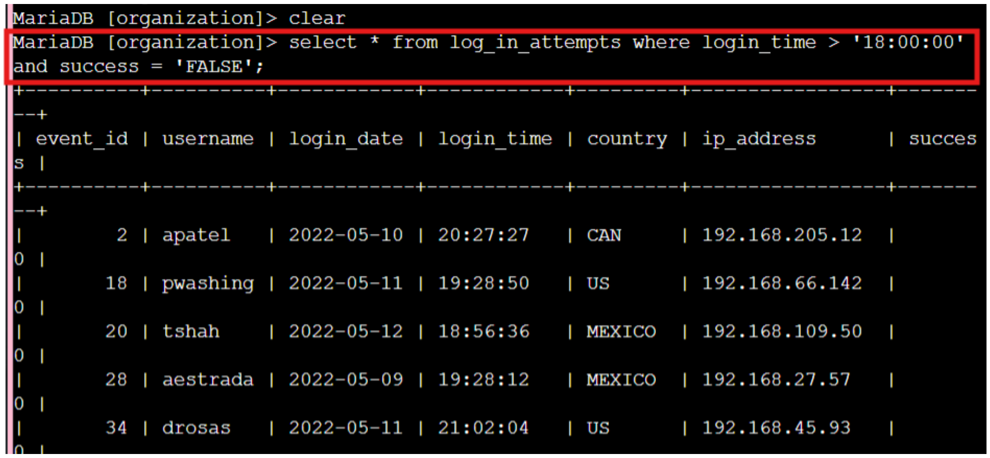
Apply filters to SQL queries

Project description

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

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

Retrieve after hours failed login attempts

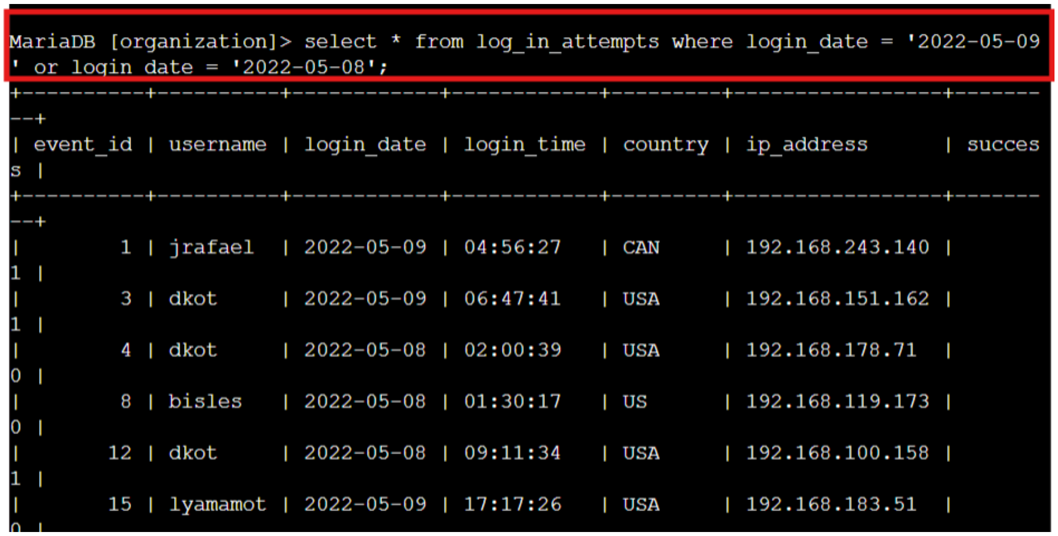


```
MariaDB [organization]> clear
MariaDB [organization]> select * from log_in_attempts where login_time > '18:00:00'
and success = 'FALSE';
+----------+----------+------------+------------+---------+-----------------+-------
--+
| event_id | username | login_date | login_time | country | ip_address      | succes
s |
+----------+----------+------------+------------+---------+-----------------+-------
--+
|        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 |
```

This SQL query was used to filter the log in attempts to find login attempts after (Greater than, or >) the time 18:00:00 (6 PM) and that did not succeed (Success variable is equal to FALSE). The 'and' operation was used since both needed to be true for it to be relevant to the investigation, the "*' symbol was used as all information from the log_in_attempts was wanted, not just certain variables/columns.

Retrieve login attempts on specific dates

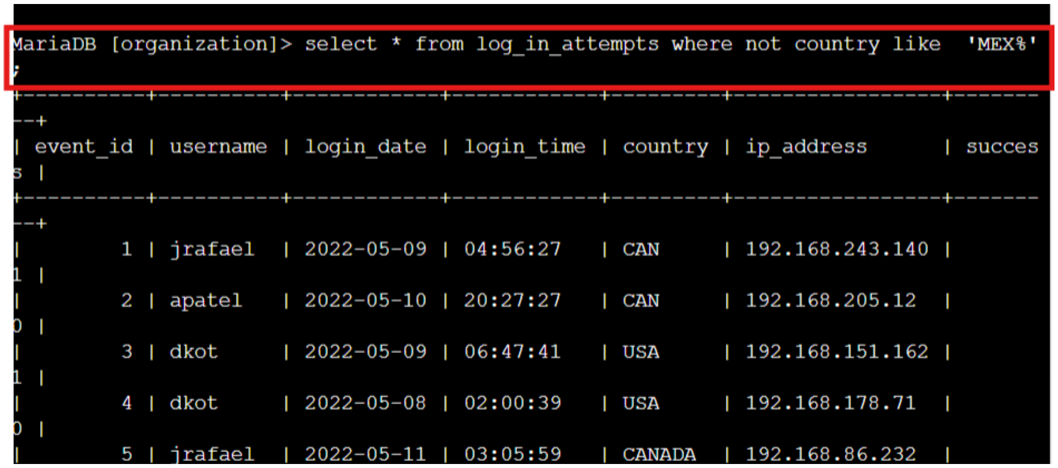
According to the exercise, the suspicious event occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or 2022-05-08 needs to be investigated.



```
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      | succes
s |
+----------+----------+------------+------------+---------+-----------------+-------
--+
|        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 |
```

All information found for the matching query data was needed so the “*” was used. The “or” operation was used since only one of the expressed conditions needed to be met.

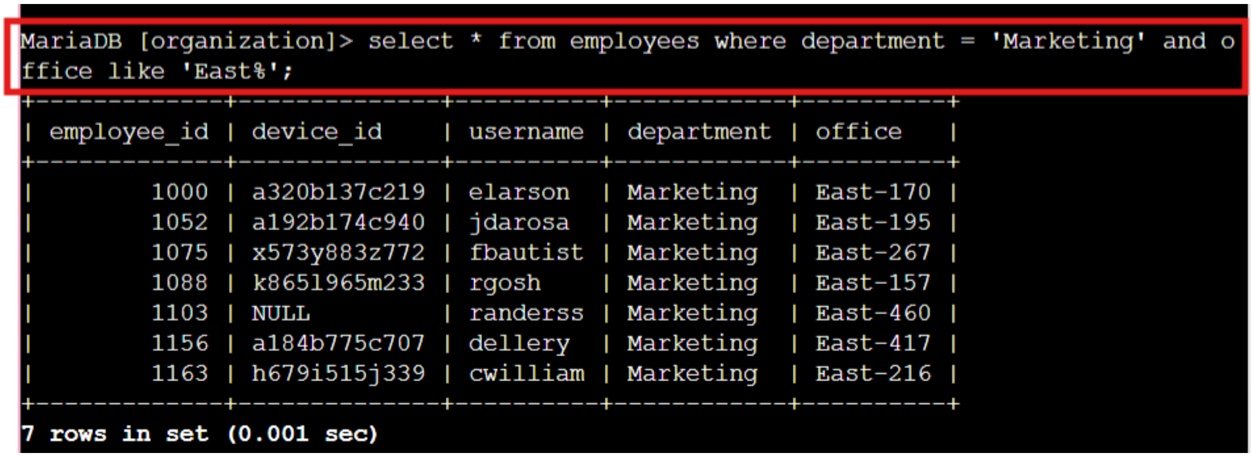
Retrieve login attempts outside of Mexico



```
MariaDB [organization]> select * from log_in_attempts where not country like  'MEX%'
;
+----------+----------+------------+------------+---------+-----------------+-------
--+
| event_id | username | login_date | login_time | country | ip_address      | succes
s |
+----------+----------+------------+------------+---------+-----------------+-------
--+
|        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  |
```

All information found for the matching query data was needed so the “*” was used. The where operation was paired with the not/like operation as everything was wanted other than the country Mexico. Based on previous exercises, it stated that using ‘MEX%’ was better than ‘MEXICO’ as both MEXICO and MEX were used to determine the country in this table, just like USA and US is used.

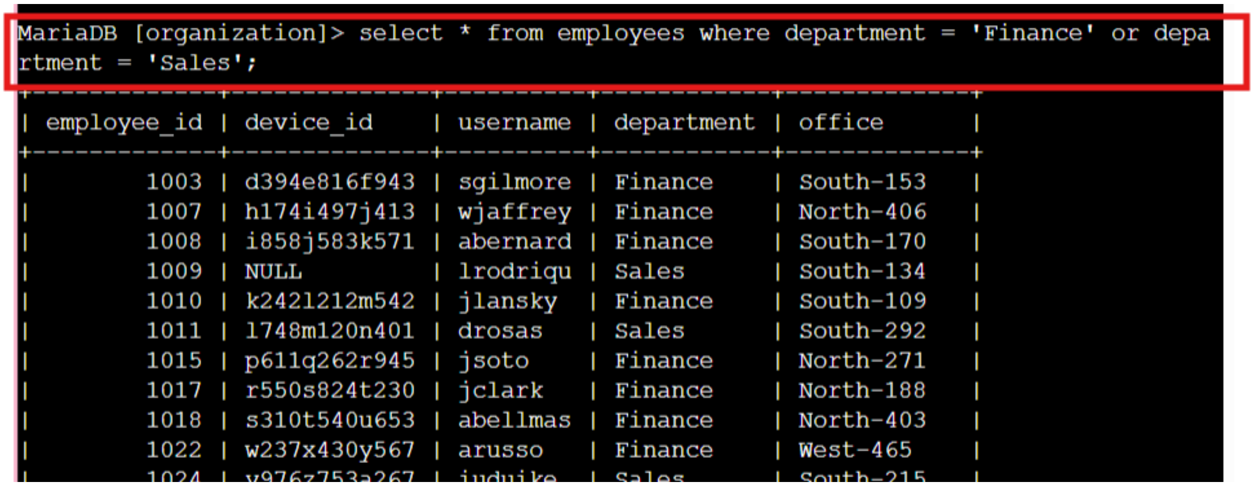
Retrieve employees in Marketing



```
MariaDB [organization]> select * from employees where department = 'Marketing' and o
ffice 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)
```

All information found for the matching query data was needed so the “*” was used. The employees then had to be from Marketing and in the east office, so the “and” operation was used as both needed to be true. For the office query, the “like” operation had to be used with the % wild card so it pulled all employees from the east section of the building no matter the office number.

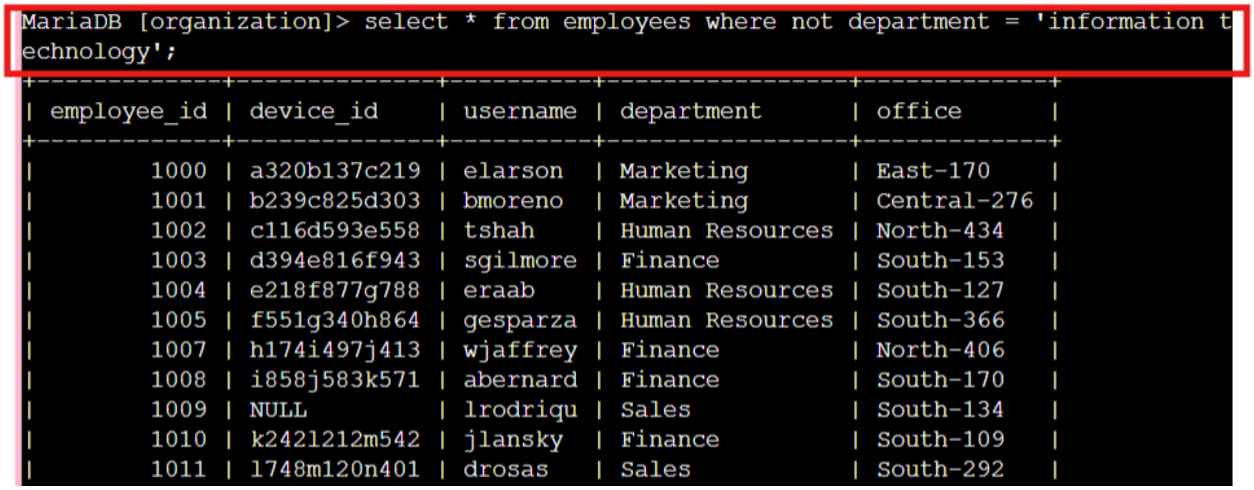
Retrieve employees in Finance or Sales



```
MariaDB [organization]> select * from employees where department = 'Finance' or depa
rtment = '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   |
```

All information found for the matching query data was needed so the “*” was used. The where and or operations were used to find all employees that were in finance or in sales, only one needed to be true for the employee to be included in the query.

Retrieve all employees not in IT



```
MariaDB [organization]> select * from employees where not department = 'information t
echnology';
+-------------+--------------+----------+-----------------+-------------+
| 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   |
```

All information found for the matching query data was needed so the “*” was used. The where and not operations were used as all employees not found in the IT department were needed in this query, this should include every other employee that is not marked as being in the Information Technology Department in the Employee table.

Summary

In this investigation I used the filters of ‘not’, ‘and’, and ‘or’ to find the correct needed information for the investigation across two tables, the ‘employees’ table and the ‘log_in_attempts’ table. I also use the ‘where’ and ‘like’ operations plus symbols like ‘*’ and ‘%’ which are wildcards, used for the select all function and the partial match functions.