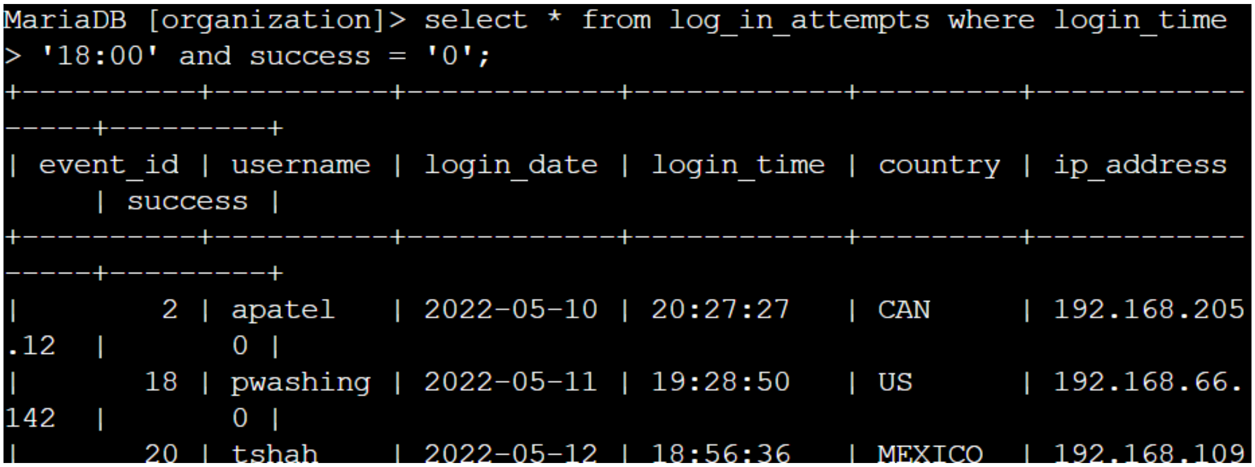
Apply Filters to SQL Queries

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.

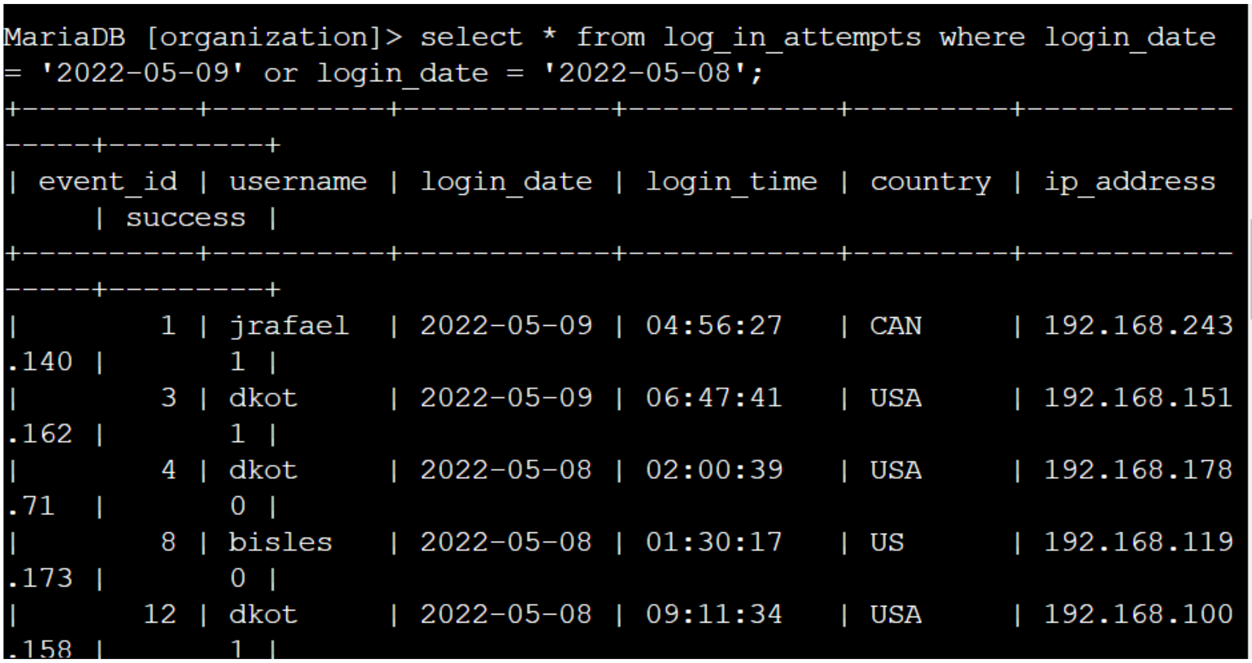
Retrieve after hours failed login attempts



```
MariaDB [organization]> 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
```

- This snapshot shows the SQL query of:
    - Select * from long_in_attempts where login_time > '18:00' and success = '0';
- Using the ">" operator instructed the system to provide the data that was greater than 18:00.
- Using the "=" operator instructed the system to provide only the data that met the failed login attempt criteria.
- This returned a list of 19 rows where there was an after hours failed login attempt.

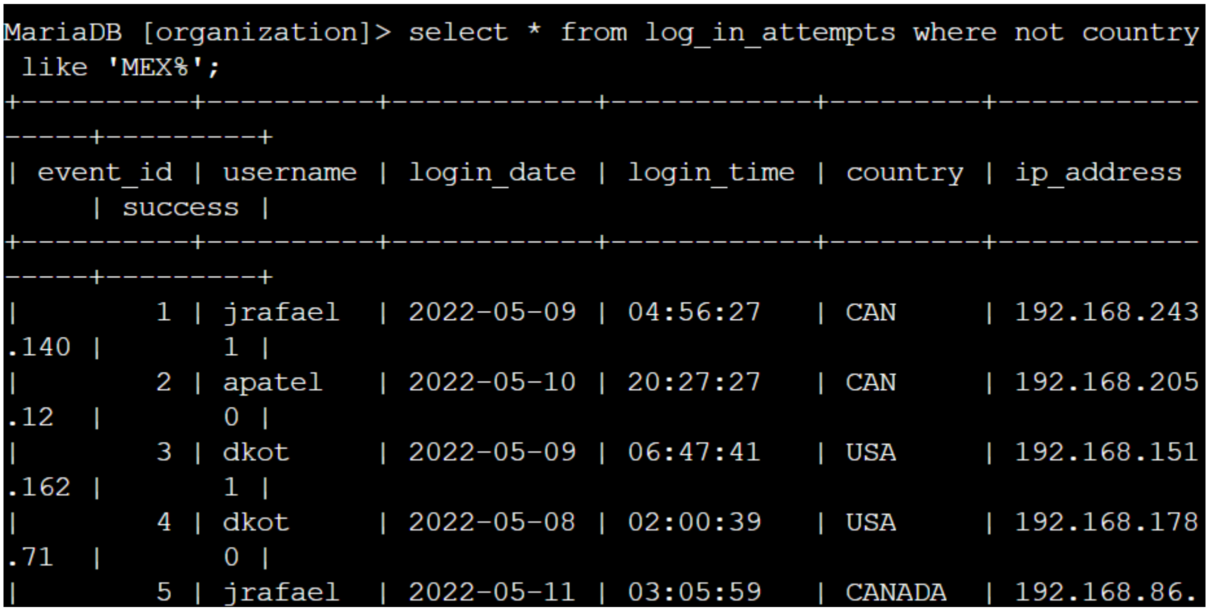
Retrieve login attempts on specific dates



```
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 |
```

- This snapshot shows the SQL query of:
  - select * from log_in_attempts where login_date = '2022-05-09' or login_date = '2022-05-08';
- Utilizing the “or” option specified that either date can be returned in the data
- This returned a list of data with a login date of either 05/09/22 or 05/08/22.

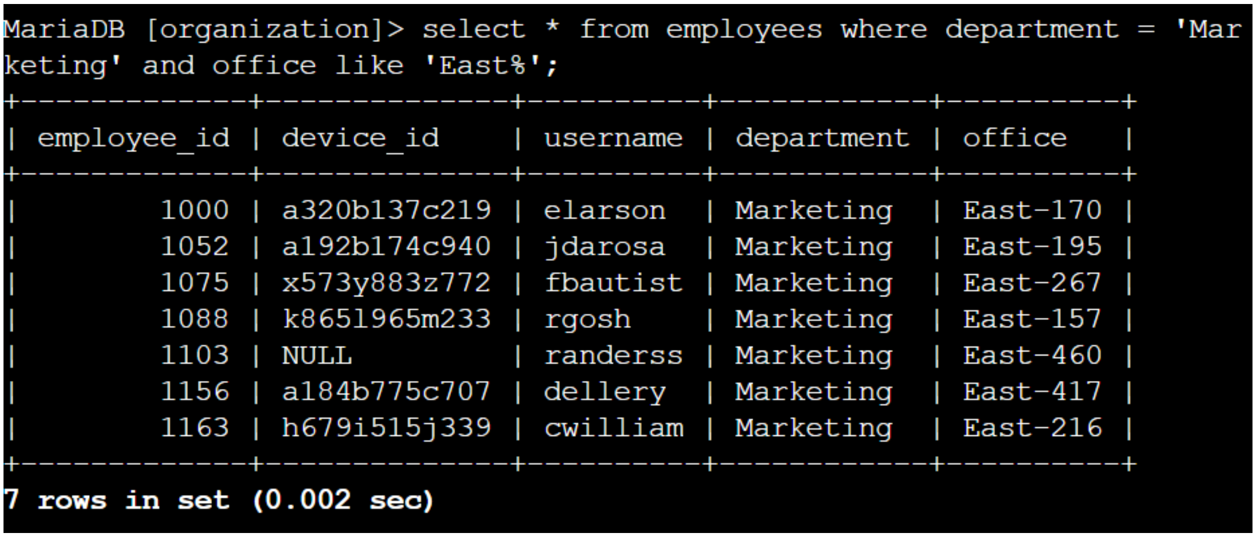
Retrieve login attempts outside of Mexico



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

- This snapshot shows a SQL query of:
    - select * from log_in_attempts where not country like 'MEX%';
- Utilizing the “not” operational allows all data to be displayed except that which was detailed.
- Utilizing the “%” operational tells the system to return data that includes and follows the parameters entered before the “%?” (no matter the number of characters).
- This returned a set of data that had every country except Mexico (or MEX) listed.

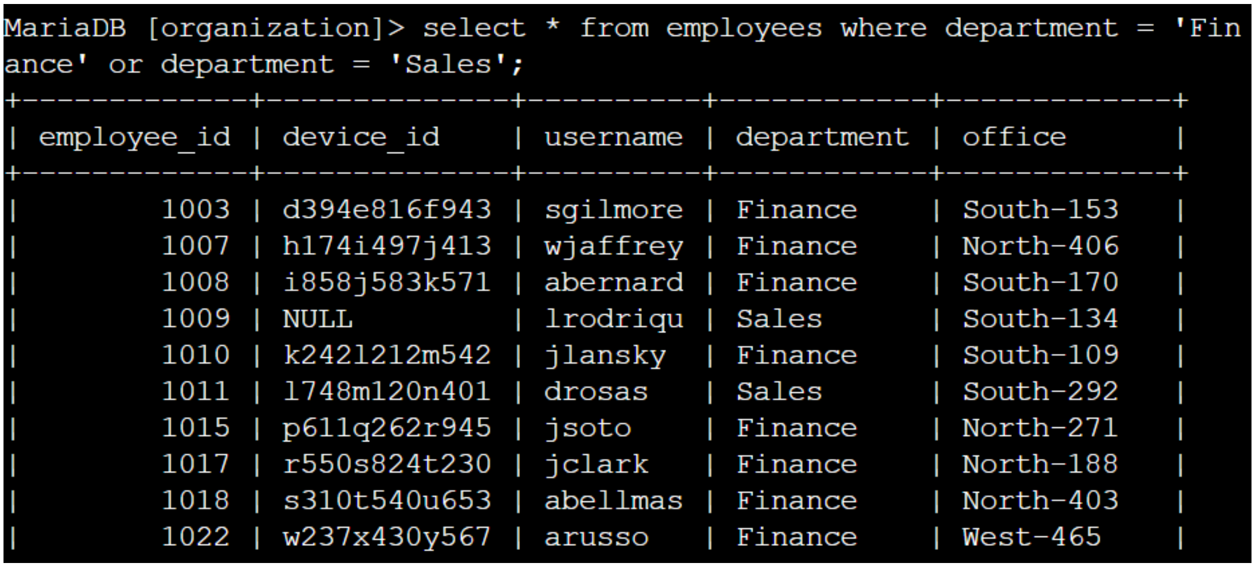
Retrieve employees in Marketing



```
MariaDB [organization]> select * from employees where department = 'Mar
keting' 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 (0.002 sec)
```

- This snapshot shows a SQL query of:
    - Select * from employees where department = ‘Marketing’ and office like ‘East%’;
- Utilizing the “=” operator instructed the system to provide only the data that met the “Marketing” requirements in the department column.
- Utilizing the “like” and “%” together instructed the system to provide only the office numbers of those who worked in the East office.
- This query returned employees who work in the marketing department that work in the East office.

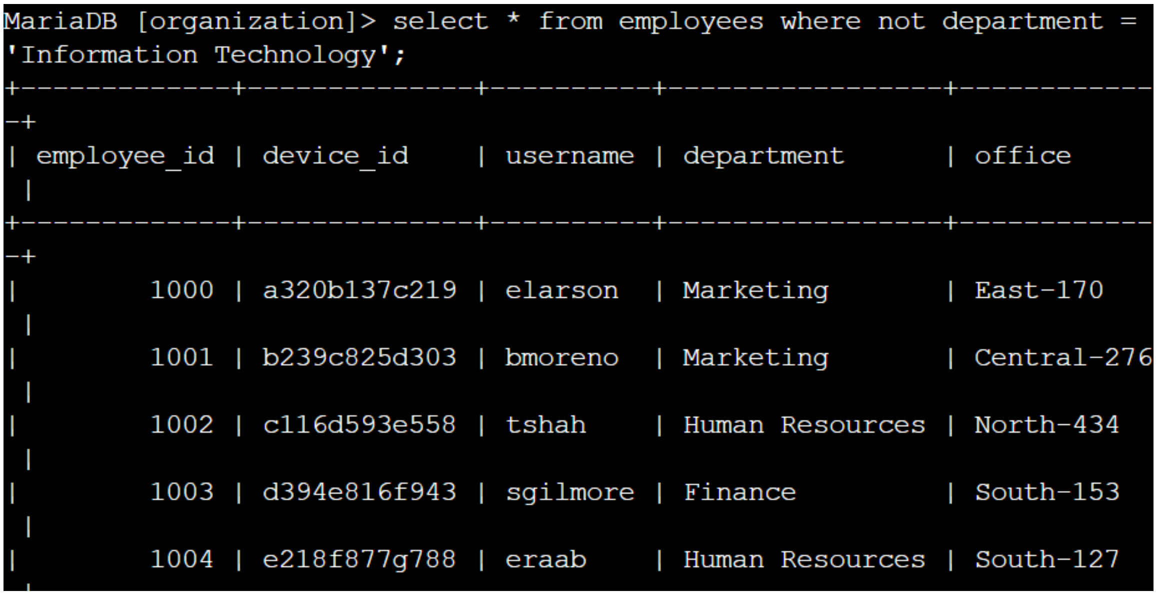
Retrieve employees in Finance or Sales



```
MariaDB [organization]> select * from employees where department = 'Fin
ance' 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    |
```

- This snapshot shows a SQL query of:
    - Select * from employees where department = ‘Finance’ or department = ‘Sales’;
- Utilizing the “or” option specified that either department can be returned in the data
- This query returned a list of employees that work either in the Finance or Sales departments.

Retrieve all employees not in IT



```
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
```

- This snapshot shows a SQL query of:
    - Select * from employees where not department = ‘Information Technology’;
- Utilizing the ‘not” option instructed the system to return data for everything except the specific department entered exactly.
- This query returned a table of employees that do not work in the Information Technology department.

Summary

I applied different filters to the SQL queries to get information on login attempts and on employees machines. I was able to complete this by utilizing many different filter options; AND, OR, NOT and LIKE. I also was able to narrow down information using the “%” wildcard to filter for different patterns.