Apply filters to SQL queries

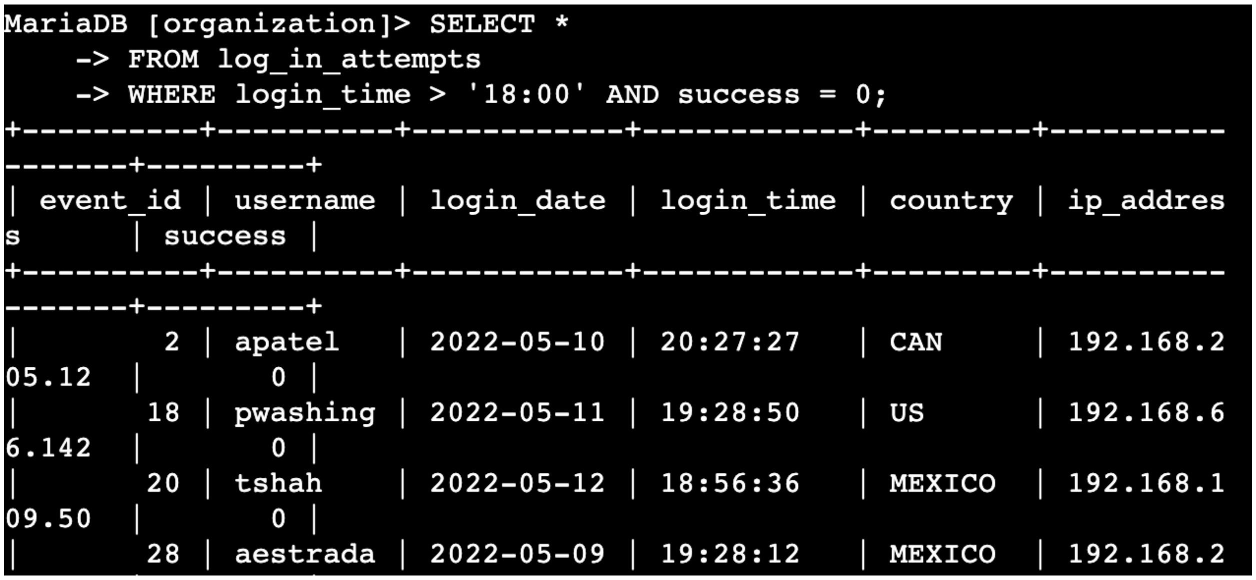
Project description

This project focuses on applying filters to SQL queries to retrieve specific information from a database. It demonstrates how to use SQL commands such as `SELECT`, `FROM`, `WHERE` along with conditions like `AND`, `OR`, `LIKE` and `NOT` to extract targeted data. The goal is to build foundational SQL skills for data filtering and analysis.

Retrieve after hours failed login attempts

In order to retrieve details about login attempts made after the business hours, the command shown in the picture below was used.

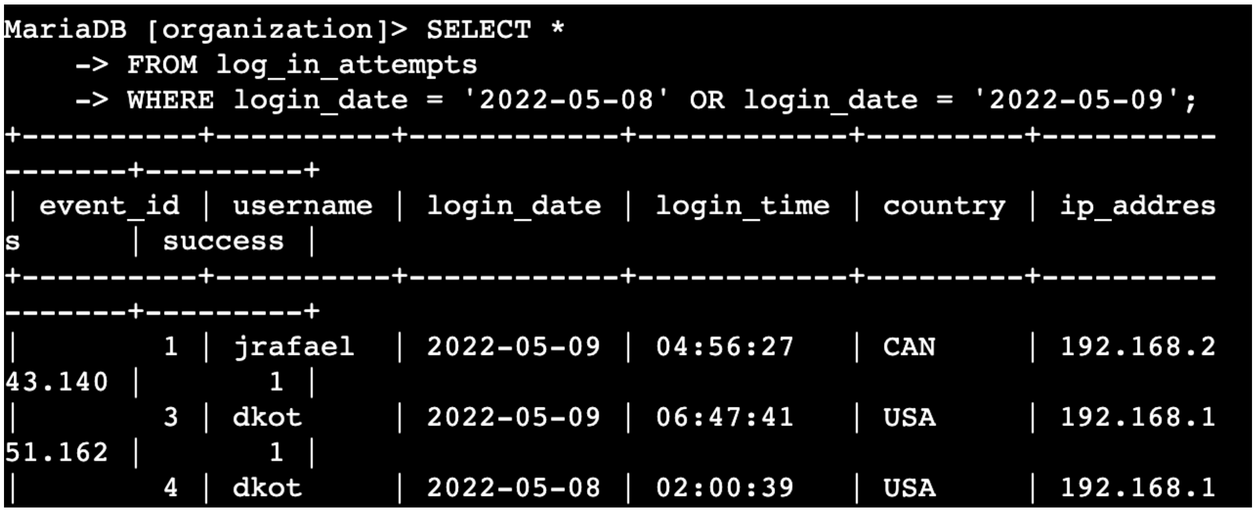
- The `SELECT` command specifies the columns to select the data from. When followed by an asterisk "`*`", it selects the data from all the columns.
- The `FROM` command specifies the table from which the columns are to be selected.
- The `WHERE` command is used to set conditions to get only the data we need. The next part of the query retrieves entries where the login time is after 6:00 PM (i.e., greater than 18:00 hours) and the login attempt was unsuccessful (success =0).



```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = 0;
+----------+----------+------------+------------+---------+----------
-------+---------+
| event_id | username | login_date | login_time | country | ip_addres
s       | success |
+----------+----------+------------+------------+---------+----------
-------+---------+
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.2
05.12   |       0 |
|       18 | pwashing | 2022-05-11 | 19:28:50   | US      | 192.168.6
6.142   |       0 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.1
09.50   |       0 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.2
```

Retrieve login attempts on specific dates

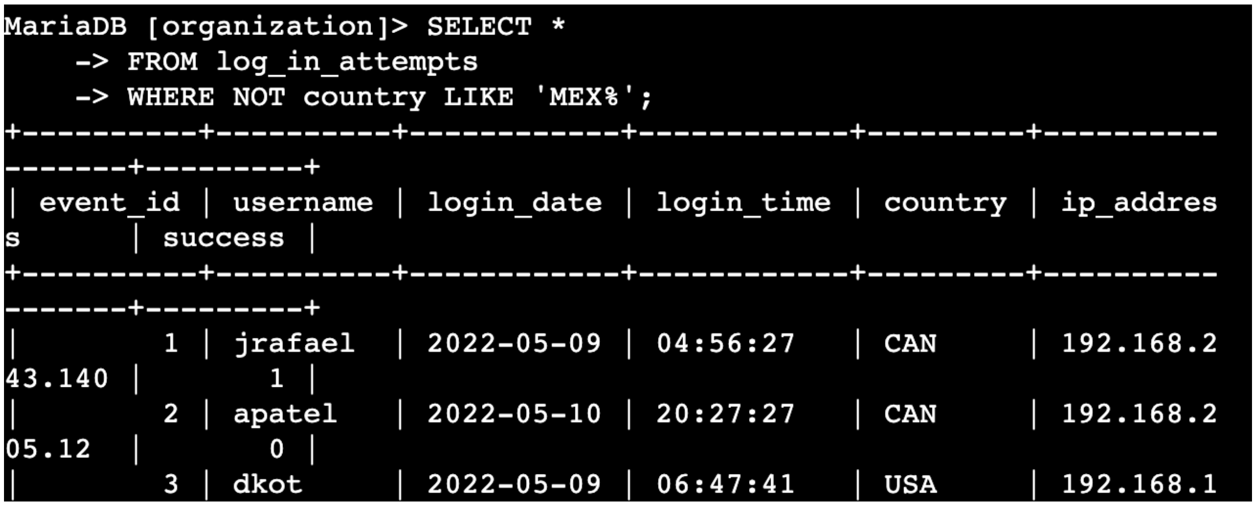
- Since we need data for attempts made on specific dates, we use the "`=`" operator to specify each date.
- In this case, we apply the `OR` condition to retrieve records that match either of the specified dates, ensuring that both conditions are considered.



```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-08' OR login_date = '2022-05-09';
+----------+----------+------------+------------+---------+----------
-------+---------+
| event_id | username | login_date | login_time | country | ip_addres
s       | success |
+----------+----------+------------+------------+---------+----------
-------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.2
43.140 |       1 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.1
51.162 |       1 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.1
```

Retrieve login attempts outside of Mexico

- To retrieve login attempts made from Mexico, we target the country column.
- The `LIKE` condition checks for strings written inside quotes and uses wildcards like "`%`" to match specific patterns. For example, "`MEX%`" will match any word that starts with "MEX".
- In this case, since we want to find login attempts made outside of Mexico, we place the `NOT` condition after the `WHERE` clause.

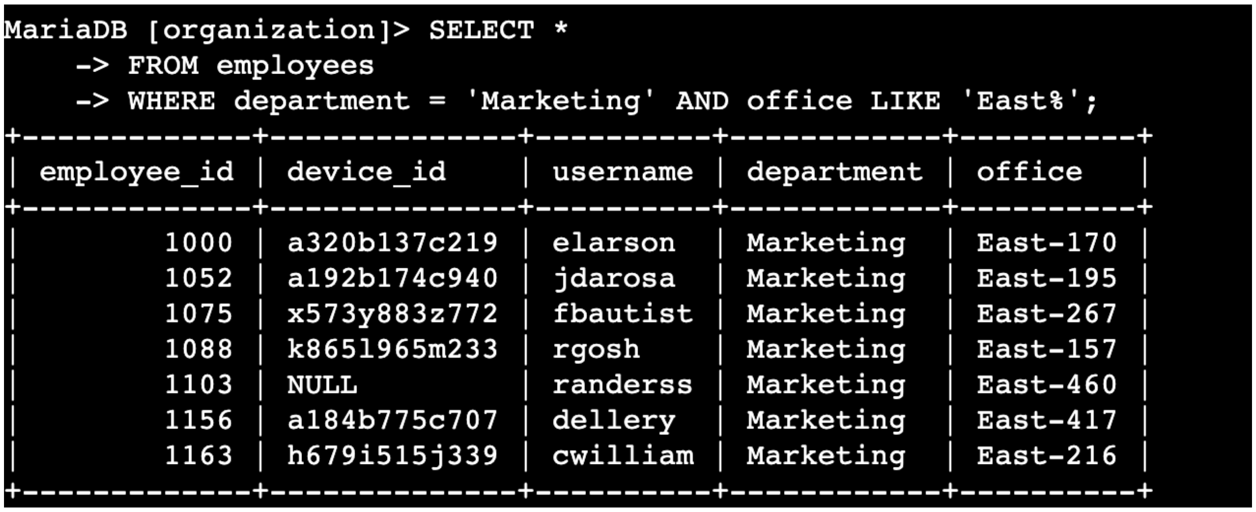


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+----------
-------+---------+
| event_id | username | login_date | login_time | country | ip_addres
s       | success |
+----------+----------+------------+------------+---------+----------
-------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.2
43.140 |       1 |
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.2
05.12  |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.1
```

Retrieve employees in Marketing

In order to get data of the employees, we enter the employees table after the `FROM` clause. Now, to get information about those employees who are in the marketing department as well as their office is in the East, we use the following commands mentioned in the picture given below.

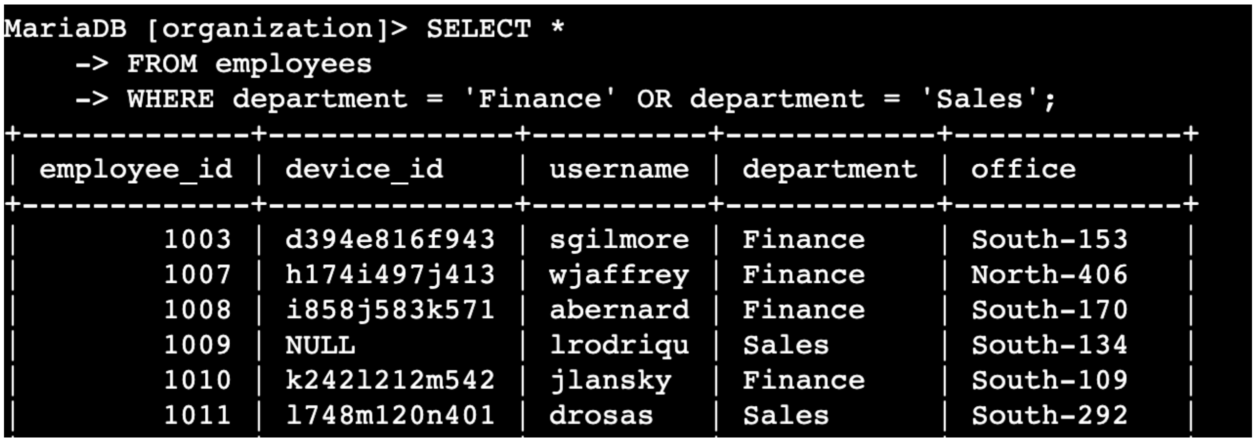
- We enclose the marketing department name in quotes, and use the “`%`” wildcard for offices located in the East.
- The `AND` condition ensures that both criteria are satisfied at the same time.



```
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 |
+-------------+-------------+----------+------------+----------+
```

Retrieve employees in Finance or Sales

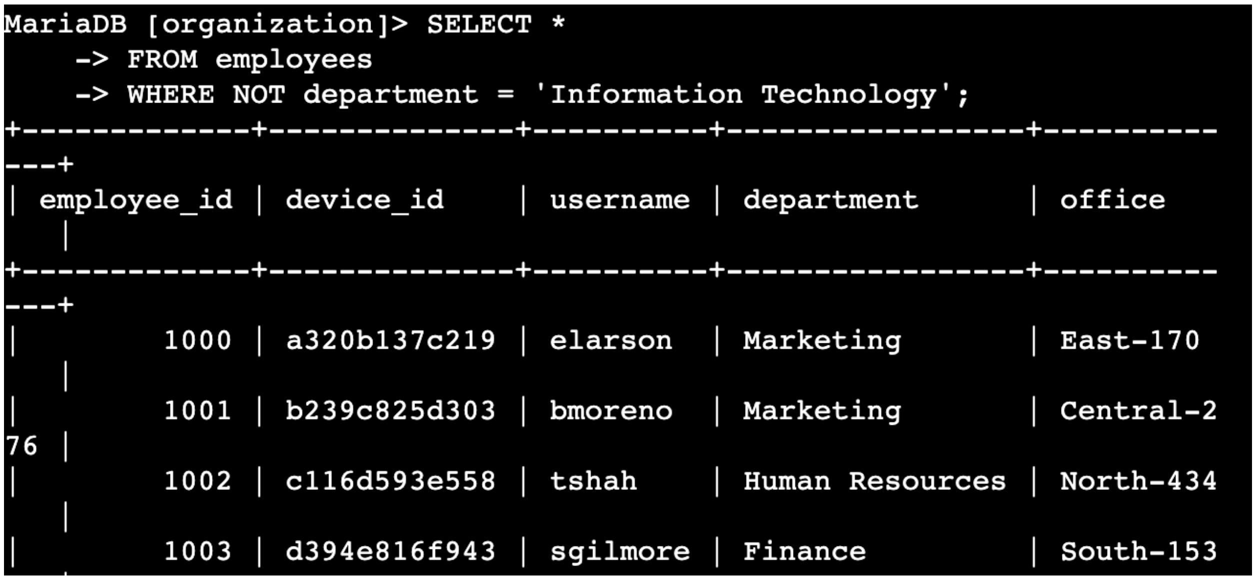
- The `OR` condition will be used in this scenario to get the entries from the finance department as well as the sales department.



```
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   |
```

Retrieve all employees not in IT

- To get the data about the employees who aren't working in the IT department, we simply add a `NOT` condition in front of the `WHERE` condition.



```
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-2
76 |
|        1002 | c116d593e558 | tshah    | Human Resources | North-434
   |
|        1003 | d394e816f943 | sgilmore | Finance         | South-153
```

Summary

The project showcases practical applications of SQL filtering techniques by writing queries for real-world scenarios. These include retrieving after-hours failed logins, identifying login attempts on given dates, and filtering employee data based on department or geographical location. The use of logical operators and pattern matching enhances the precision of data retrieval. Overall, the project strengthens query-building skills for effective data analysis in databases.