Apply filters to SQL queries

Project description

This organization has a few things that need to be rectified. We will pull up information about employees, their machines, and which departments they belong to. Certain data will need to be investigated due to potential security concerns, as well as updating computers.

We will be filtering out all the necessary data to analyze these potential incidents and ensure the employees and their computers are safe.

Retrieve after hours failed login attempts

The team has investigated failed login attempts after business hours. What we need to do is retrieve this information from the login activity to identify all failed attempts made after 18:00 (6:00pm).

```
MariaDB [organization]> Select *
    -> From log_in_attempts
    -> Where login_time > '18:00' And success = False;
+----------+----------+------------+------------+---------+-----------------+---------+
| 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 (0.108 sec)
```

In order to find the failed login attempts during after hours, we need to input the syntax: `Select * From log_in_attempts Where login_time > ‘18:00’ And success = False;`

This will filter all data from the `log_in_attempts` table with the login times and failed attempts by filtering through the `login_time` and `success` tables.

The success column contains True or False. True and False are displayed as 1 (True) or 0 (False). This is known as boolean values.

At the bottom of the page we can see a displayed total of 19 failed after hour attempts.

Retrieve login attempts on specific dates

Now, the team is investigating a suspicious event that occurred on two specific dates. Those dates being: ‘2022-05-09’ and ‘2022-05-08’. The goal here is to filter out all login attempts for those two days.

```
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 |
```

We enter the syntax needed to perform this, that being: Select * From log_in_attempts Where login_date = ‘2022-05-09’ OR ‘2022-05-08’;

With this, we are selecting all login attempts (Select *) from the login attempts table (From log_in_attempts), then filtering the login dates for only two days (Where login_date = ‘2022-05-09’ OR login_date = ‘2022-05-08’;), those being: ‘2022-05-09’ and ‘2022-05-08’.

```
|      190 | jsoto    | 2022-05-09 | 05:09:21   | USA     | 192.168.25.60   |       0 |
|      191 | cjackson | 2022-05-08 | 06:46:07   | CANADA  | 192.168.7.187   |       0 |
|      193 | lrodriqu | 2022-05-08 | 07:11:29   | US      | 192.168.125.240 |       0 |
|      197 | jsoto    | 2022-05-08 | 09:05:09   | US      | 192.168.36.21   |       0 |
+----------+----------+------------+------------+---------+-----------------+---------+
75 rows in set (0.015 sec)
```

Since there was a vast amount of logins, I have condensed it here. There were a total of 75 login attempts made on the dates of ‘2022-05-09’ and ‘2022-05-08’.

Retrieve login attempts outside of Mexico

The team now needs to investigate login attempts that did not originate in Mexico, telling us there is no security concern coming from that location. Mexico can be populated as ‘MEX’ and ‘MEXICO’ within the country column.

```
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 |
```

In order to do this and to keep the syntax simple, it is easiest to keep ‘MEX’ out of the filter altogether. We do this by:

- Select * From log_in_attempts Where Not country Like ‘MEX%’

- The ‘Not’ operator allows us to remove that option from the filter output
- Since Mexico is outputted as ‘MEX’ and ‘MEXICO’ we need to add the % after MEX. That will allow any characters to display after the letters MEX, in case the whole word MEXICO were to populate as well.

```
|      196 | acook    | 2022-05-10 | 09:56:48   | CAN     | 192.168.52.90   |       0 |
|      197 | jsoto    | 2022-05-08 | 09:05:09   | US      | 192.168.36.21   |       0 |
|      200 | jclark   | 2022-05-12 | 01:11:45   | CANADA  | 192.168.91.103  |       1 |
+----------+----------+------------+------------+---------+-----------------+---------+
144 rows in set (0.013 sec)
```

With this, we are selecting all login attempts made, solely excluding Mexico. This will then filter out a total of 144 login attempts from any country excluding Mexico.

Retrieve employees in Marketing

With this task, the team is updating employee machines and needs to obtain the information of employees in the ‘Marketing’ department and are located in all offices of the East building. In order to find this information, we will need to use the ‘And’ and ‘Like’ operators.

```
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)
```

The syntax used above was: Select * From employees Where department = ‘Marketing’ And office Like ‘East%’;.
What this syntax did was select all (Select *) employees from the employees table (From employees), then target the department of Marketing and all offices in the East building (Where department ‘Marketing’ And office Like ‘East%’).

The ‘And’ operator was needed to ensure both arguments were true. And, since we were targeting the whole East office with multiple building numbers, we needed the ‘Like’ operator. In addition to that, we use ‘East%’ so that we can filter all East office buildings with anything after the keyword: ‘East’.

And with that, we can see all employees to update all the necessary machines.

Retrieve employees in Finance or Sales

We now need to perform a different update to the computers of all employees in the Finance and Sales departments.

```
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   |
```

Here, we are able to see a syntax of: Select * From employees Where department = ‘Finance’ OR department = ‘Sales’;.
What this will do is: Select all (Select *) information from the employees table (From employees), while only displaying the ‘Finance’ or ‘Sales’ departments (Where department = ‘Finance’ OR ‘Sales’;).

```
|        1187 | 1963g637h851 | bbode    | Finance    | East-351    |
|        1188 | g164h566i795 | noshiro  | Finance    | West-252    |
|        1195 | n516o853p957 | orainier | Finance    | East-346    |
+-------------+--------------+----------+------------+-------------+
71 rows in set (0.003 sec)
```

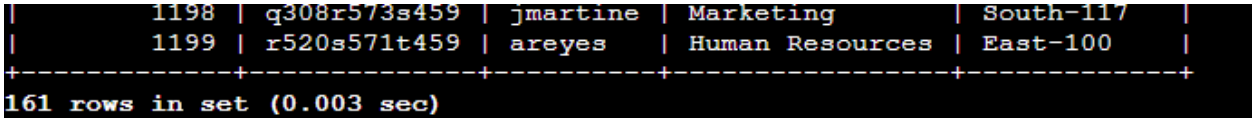
We have now retrieved information on both the Finance and Sales departments so we may perform the appropriate computer updates.

Retrieve all employees not in IT

Lastly, the team needs to perform one last update, but it seems the Information Technology team has already completed theirs. We will need to retrieve all information of employees who are not in the IT department.

```
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 |
```

Here we are using the syntax : Select * From employees Where Not department = ‘Information Technology’;
This will select all (Select *) employees (From employees) that are not in the Information Technology department (Where Not department = ‘Information Technology’).



This will populate a total of 161 employees and their information so that the team may make the proper computer updates. And with that, all necessary tasks have been completed.

Summary

We were able to sort through a wide database with this organization and able to rectify any potential security concerns. We also made sure all the needed departments' computers were updated to ensure a strong security posture.

There were many different operators that were used in this like: `AND`, `OR`, `LIKE`, and `NOT`. All these operators allow us to filter through a wide database and retrieve exactly the information we need.