Applying Filters to SQL Queries

Project description

As a security professional, part of my job is to investigate security issues to help keep systems secure. In this scenario, I discovered some potential security issues that involve login attempts and employee machines.

My task was to examine the organization's data in their **employees** and **log_in_attempts** data tables. I used SQL filters to retrieve records from different datasets and investigated the potential security issues.

Note: The tables and their formats can be seen in the "Table Formats" document within this folder.

Retrieve after hours failed login attempts

To retrieve the number of failed login attempts after office hours end, I used the command **SELECT * FROM log_in_attempts WHERE login_time > '18:00' AND success = 0;.** Office hours end at 18:00 and a failed login attempt is recorded as a 0 in the success column. The command and output can be seen below. The number of failed login attempts after hours is 19.

```
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  | 192.168.109.50  |       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 (0.329 sec)
```

Retrieve login attempts on specific dates

Now the team is investigating a suspicious event that occurred on 2022-05-09 and wants me to retrieve all login attempts that occurred on this day and the day before (2022-05-08). To do this, I used the command **SELECT * FROM log_in_attempts WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';**. The output is too large for an image; however, the command can be seen below. The number of login attempts on these dates was 75 attempts.

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

Retrieve login attempts outside of Mexico

Now the team is investigating logins that did not originate in Mexico and want me to find this information. Since the country field includes entries with 'MEX' and 'MEXICO', the NOT and LIKE operators and the matching pattern 'MEX%' were used. To retrieve these login attempts, I used the command **SELECT * FROM log_in_attempts WHERE NOT country LIKE 'MEX%';**. The output is too large for an image; however, the command can be seen below. The number of login attempts outside of Mexico was 144 attempts.

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE NOT country LIKE 'MEX%';
```

Retrieve employees in Marketing

For the remaining tasks I need to retrieve the information from the department and office columns in the employees table. The team is updating employee machines and I need to obtain the information about employees in the Marketing department who are all located in all offices in the East building (such as 'East-170' or 'East-320').

To achieve this, I used the command **SELECT * FROM employees WHERE department = 'Marketing' AND office LIKE 'East%';.** As seen below, there were 7 employees in the Marketing department in the East office building.

```
MariaDB [organization]> SELECT * FROM employees WHERE department = 'Ma
rketing' 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)
```

Retrieve employees in Finance or Sales

Now, the team needs to perform a different update to the computers of all employees in the Finance or the Sales department, and need me to locate this information. I completed this task by using the command **SELECT * FROM employees WHERE department = 'Finance' OR department = 'Sales';** as seen below. There were 71 employees returned.

```
MariaDB [organization]> SELECT * FROM employees WHERE department = 'Finance' OR department = 'Sales';
```

Retrieve all employees not in IT

Lastly, the team needs to make one more update and needs the information about employees who are not in the IT department. I did this using the command SELECT * FROM employees WHERE NOT department = 'Information Technology';. The number of employees returned was 161 employees.

```
MariaDB [organization]> SELECT * FROM employees WHERE NOT department = 'Information Technology';
```

Summary

I applied filters to SQL queries to get specific information on login attempts and employee machines. I used two different tables, log_in_attempts and employees. I used the AND, OR, and NOT operators to filter for the specific information needed for each task. I also used LIKE and the percentage sign (%) wildcard to filter for patterns.