Apply filters to SQL queries

Project description

In this project I demonstrate my ability using SQL for security related tasks. This includes the investigation of all security issues within the various system databases and the updating of employee computers.

Retrieve after hours failed login attempts

In order to retrieve the necessary login times we will need to pull FROM the 'log_in_attempts_table' using the the 'SELECT' keyword and then specify failed attempts (stored as a boolean value under the 'success' column, where 0 is equal to FALSE) past 6:00 PM (or greater than a login time of '18:00:00', under the 'login_time' column). Using the AND operator ensures that both conditions must be true for a given row to be displayed in the table. A total of 19 rows is returned.

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00:00' AND success = 0;
+----------+----------+------------+------------+---------+-----------------+-----
--+
| event_id | username | login_date | login_time | country | ip_address      | succ
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  |
```

Retrieve login attempts on specific dates

Once again we will search the 'log_in_attempts' table for any login attempts made on both the dates '2022-05-09' OR '2022-05-08'. This is done by specifying conditions that can BOTH be true under the 'login_date' column using the OR operator. It is important to always specify which column you are pulling from even if you are pulling from the same column.

```
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 |
```

Retrieve login attempts outside of Mexico

Selecting from the ‘log_in_attempts” table we are going to search for all login attempts that did NOT occur in Mexico. Under the ‘country’ column we can see that the table records both ‘MEX’ and ‘MEXICO’ for login attempts occurring in Mexico. In order to exclude both string values found under the ‘country’ column we tell SQL to NOT return countries that start with the string value of ‘MEX%’ using the ‘LIKE’ operator. The percent symbol is a “wildcard” and represents a placeholder for an uncertain amount of characters within a string from the point of the string value specified.

```
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 |
```

Retrieve employees in Marketing

Similarly, let’s say we have been tasked with updating the Marketing department’s user devices in the East office building. We can do this by searching the ‘employees’ table using the ‘FROM’ clause. Also, by using the AND operator we can include conditions for both the ‘department’ and ‘office’ columns which in this case will be ‘Marketing’ and ‘East%’ respectively.

```
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)
```

Retrieve employees in Finance or Sales

This time we want to INCLUDE both the Finance and Sales departments. To do this we use the OR operator while searching for the specified conditions under the 'department' column.

```
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   |
```

Retrieve all employees not in IT

This time we are tasked with making an update to all employee devices that are NOT in the IT department. To do this we specify to exclude any rows that include "Information Technology" under the department column using the NOT operator.

```
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   |
```

Summary

In these SQL query demonstrations I have shown a familiarity with SQL keywords and an ability to navigate multiple tables within a system. I have also shown my ability to select specific information from those tables or even exclude information when necessary.