Apply filters to SQL queries

Project description

My organization is working to make their system more secure. It is my job to ensure the system is safe, investigates all potential security issues, and updates employee computers as needed. The following steps provide examples of how I used SQL with filters to perform security-related tasks.

Retrieve after hours failed login attempts

The Purpose: Extract rows based on specific time values or ranges, using the AND operator.

The query below shows how to retrieve all the data from the 'log_in_attempts' table, and filter only unsuccessful login(can be seen as 0) attempts after 18:00.

```
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 (0.001 sec)
```

Retrieve login attempts on specific dates

Purpose: Extract rows based on specific date values or ranges, using the OR operator.

The query below shows how to retrieve all the data from the 'log_in_attempts' table, and filter the login_date to 2022/05/08/09.

```
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_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 |
|       15 | lyamamot | 2022-05-09 | 17:17:26   | USA     | 192.168.183.51  |       0 |
|       24 | arusso   | 2022-05-09 | 06:49:39   | MEXICO  | 192.168.171.192 |       1 |
|       25 | sbaelish | 2022-05-09 | 07:04:02   | US      | 192.168.33.137  |       1 |
```

Retrieve login attempts outside of Mexico

Purpose: The LIKE operator allows you to find rows where text or character data matches a specific pattern.

The query below shows how to retrieve all the data from the 'log_in_attempts' table, and remove any result from the country column that starts with 'MEX'.

```
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  | 192.168.86.232  |       0 |
```

Retrieve employees in Marketing

Purpose: Using the AND operator at the same time to use two filters at the same time.

The query below shows how to retrieve all the data from the "employees" table, shows only the marketing department, and offices that start with 'EAST'.

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department LIKE '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

Purpose: Use the OR operator to use only one of two filters for the same column.

The query below shows how to retrieve all the data from the 'employees' table, and shows only the Finance and Sales departments

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department LIKE 'Finance'
    -> OR department LIKE '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 |
```

Retrieve all employees not in IT

Purpose: Use the NOT operator to clean the output of certain data.

The query below shows how to retrieve all the data from the employees' table, apart from data related to the information technology department.

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department NOT LIKE '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   |
```

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.