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, investigate all potential security issues, and update 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

A possible security incident happened after business hours (after 18:00). We need to investigate all failed login attempts that occurred during that time.

The following code demonstrates how I created a SQL query to filter for failed login attempts that occurred after business hours.

```
   -> FROM log_in_attempts
   -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+-
--------+
 event_id | username | login_date | login_time | country | ip_address      |
uccess |
+----------+----------+------------+------------+---------+-----------------+-
--------+
        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   |
```

The first part of the screenshot shows my query, and the second part displays a section of the output. This query focuses on failed login attempts that took place after 18:00. I began by selecting all data from the `log_in_attempts` table. Then, I added a **WHERE** clause with an **AND** operator to narrow the results to only include unsuccessful login attempts that occurred after 18:00. The first condition is `login_time > '18:00'`, which filters for attempts made after that time. The second condition is success = FALSE, which identifies the failed login attempts.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or on the day before needs to be investigated.

The following code shows how I wrote a SQL query to filter login attempts based on specific dates.

```
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 |
|        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 |
```

The first part of the screenshot shows my query, while the second part displays a portion of the output. This query retrieves all login attempts that occurred on 2022-05-09 or 2022-05-08. I started by selecting all data from the `log_in_attempts` table, then used a **WHERE** clause with an **OR** operator to filter the results to include only login attempts from either 2022-05-09 or 2022-05-08. The first condition is `login_date = '2022-05-09'`, which filters for logins on 2022-05-09. The second condition is `login_date = '2022-05-08'`, which filters for logins on 2022-05-08.

Retrieve login attempts outside of Mexico

After reviewing the organization's data on login attempts, I suspect there may be an issue with the attempts made outside of Mexico. These login attempts should be investigated.
The following code shows how I wrote a SQL query to filter for login attempts that occurred outside of Mexico.

```
MariaDB [organization]>
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  |
```

The first part of the screenshot displays my query, while the second part shows a section of the output. This query retrieves all login attempts from countries other than Mexico. I began by selecting all data from the `log_in_attempts` table. Then, I used a **WHERE** clause with **NOT** to filter for countries that are not Mexico. I used **LIKE** with the pattern **MEX%** to match entries, as the dataset identifies Mexico as both MEX and MEXICO. The percentage sign (`%`) is used with **LIKE** to represent any number of unspecified characters.

Retrieve employees in Marketing

My team needs to update the computers for specific employees in the Marketing department. To proceed, I need to gather information about which employee machines require updates.

The following code shows how I wrote a SQL query to filter for employee machines belonging to those in the Marketing department located in the East building.

```
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 (0.002 sec)
```

The first part of the screenshot shows my query, while the second part displays a section of the output. This query retrieves all employees in the Marketing department located in the East building. I began by selecting all data from the `employees` table. Then, I applied a **WHERE** clause with an **AND** operator to filter for employees who work in the Marketing department and in the East building. I used **LIKE** with the pattern `'East%'` to match entries, as the `office` column includes specific office numbers for the East building. The first condition, `department = 'Marketing'`, filters for employees in the Marketing department, while the second condition, `office LIKE 'East%'`, filters for those in the East building.

Retrieve employees in Finance or Sales

The machines for employees in the Finance and Sales departments also need to be updated. Since a different security update is needed, I have to get information on employees only from these two departments.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Finance or Sales departments.

```
MariaDB [organization]> clear
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 | l748m120n401 | drosas   | Sales      | South-292   |
```

The first part of the screenshot displays my query, while the second part shows a section of the output. This query retrieves all employees in the Finance and Sales departments. I began by selecting all data from the `employees` table. Then, I applied a **WHERE** clause with the **OR** operator to filter for employees in either the Finance or Sales departments. I used **OR** instead of **AND** because I want to include employees from both departments. The first condition, `department = 'Finance'`, filters for those in the Finance department, while the second condition, `department = 'Sales'`, filters for employees in the Sales department.

Retrieve all employees not in IT

My team needs to make one more security update on employees who are not in the Information Technology department. To make the update, I first have to get information on these employees.

The following demonstrates how I created a SQL query to filter for employee machines from employees not in the Information Technology 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 |
|        1002 | c116d593e558 | tshah    | Human Resources | North-434   |
|        1003 | d394e816f943 | sgilmore | Finance         | South-153   |
|        1004 | e218f877g788 | eraab    | Human Resources | South-127   |
|        1005 | f551g340h864 | gesparza | Human Resources | South-366   |
|        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 | l748m120n401 | drosas   | Sales           | South-292   |
|        1015 | p611q262r945 | jsoto    | Finance         | North-271   |
|        1016 | q793r736s288 | sbaelish | Human Resources | North-229   |
|        1017 | r550s824t230 | jclark   | Finance         | North-188   |
```

The first part of the screenshot shows my query, while the second part displays a section of the output. This query retrieves all employees who are not in the Information Technology department. I began by selecting all data from the `employees` table and then used a **WHERE** clause with **NOT** to filter out those in the IT department.

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.