Apply filters to SQL queries

Project description

You are a security professional at a large organization. Part of your job is to investigate security issues to help keep the system secure. You recently discovered some potential security issues that involve login attempts and employee machines.

Your task is to examine the organization's data in their employees and log_in_attempts tables. You'll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues.

Retrieve after hours failed login attempts

"Your team is investigating failed login attempts that were made after business hours. You want to retrieve this information from the login activity. You'll identify all unsuccessful attempts after 18:00."

Here's what I've done to complete this task:

```
MariaDB [organization]> clear
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 (0.002 sec)
```

The screenshot shows my query and part of the output. This query looks for failed login attempts after 6 PM. I started by selecting all data from the log_in_attempts table. Then, I used a WHERE clause with an AND to only show logins after 6 PM that failed. The first condition, `login_time > '18:00'`, filters logins after 6 PM, and the second, `success = FALSE`, shows the failed ones.

Retrieve login attempts on specific dates

"Your team is investigating a suspicious event that occurred on `'2022-05-09'`. You want to retrieve all login attempts that occurred on this day and the day before (`'2022-05-08'`)."

Here's what I've done to complete this task:

```
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 |
```

The screenshot shows my query and part of the output. This query finds all login attempts from May 9, 2022, or May 8, 2022. First, I selected all the data from the log_in_attempts table. Then, I used a WHERE clause with an OR to show only logins from those two dates. The first condition is `login_date = '2022-05-09'` for May 9, and the second is `login_date = '2022-05-08'` for May 8.

Retrieve login attempts outside of Mexico

"Now, your team is investigating logins that did not originate in Mexico, and you need to find this information."

Here's what I've done to complete this task:

```
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 |
```

The screenshot shows my query and part of the output. This query finds all login attempts from countries other than Mexico. First, I selected all data from the log_in_attempts table. Then, I used a WHERE clause with NOT to leave out Mexico. I used LIKE with "MEX%" to match both MEX and MEXICO since the dataset uses both. The % symbol means any number of characters after "MEX".

Retrieve employees in Marketing

"Your team is updating employee machines, and you need to obtain the information about employees in the `'Marketing'` department who are located in all offices in the East building (such as `'East-170'` or `'East-320'`)."

Here's what I've done to complete this task:

```
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 (0.046 sec)
```

The screenshot shows my query and part of the output. This query finds all employees in the Marketing department who work in the East building. First, I selected all data from the employees table. Then, I used a WHERE clause with AND to filter for Marketing employees in the East building. I used LIKE with "East%" because the office column includes the East building and office numbers. The first condition, `department = 'Marketing'`, filters for Marketing, and the second, `office LIKE 'East%'`, filters for the East building.

Retrieve employees in Finance or Sales

"Now, your team needs to perform a different update to the computers of all employees in the Finance or the Sales department, and you need to locate information on these employees."

Here's what I've done to complete this task:

```
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   |
```

The screenshot shows my query and part of the output. This query finds all employees in the Finance or Sales departments. First, I selected all the data from the employees table. Then, I used a WHERE clause with OR to get employees from either department. I used OR instead of AND because I want employees from both departments. The first condition, `department = 'Finance'`, gets Finance employees, and the second, `department = 'Sales'`, gets Sales employees.

Retrieve all employees not in IT

"Your team needs to make one more update. This update was already made to employee computers in the Information Technology department. The team needs information about employees who are not in that department."

Here's what I've done to complete this task:

```
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   |
```

The screenshot shows my query and some of the results. The query finds all employees who are not in the Information Technology department. First, I selected all data from the employees table. Then, I used a WHERE clause with NOT to filter out anyone in that department.

Summary

In summary, I used filters in SQL queries to find specific information about login attempts and employee machines. I worked with two tables: log_in_attempts and employees. I used the AND, OR, and NOT operators to get the right data for each task. I also used LIKE with the % wildcard to filter for patterns.