Apply filters to SQL queries

Project description

In my organization, I am in charge of security. My organization wants to make its systems more secure due to a previous discovery of security issues with access attempts and employees' machines. To address this, I have used the following SQL filters on the `employees` and `log_in_attempts` tables.

Retrieve after hours failed login attempts

There was a potential security incident after working hours (after 18:00). Therefore, all access attempts after work hours must be investigated.

```
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      | succe
ss |
+----------+----------+------------+------------+---------+-----------------+------
---+
|        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 |
```

The first part shown in the screenshot is my SQL query, followed by part of the output. This query is formed by selecting all data (`SELECT *`) from the `log_in_attempts table` (`FROM log_in_attempts`). Then, a `WHERE` clause is used with an `AND` operator to filter the output. The first condition (`login_time > '18:00'`) shows the access attempts after 18:00, followed by the second condition (`success = FALSE`), which displays failed access attempts. It must be clarified that, in MySQL, the Boolean values are stored as `1` for `TRUE`, and `0` for `FALSE`.

Retrieve login attempts on specific dates

My team discovered that a suspicious event occurred on 2022-05-09. To investigate this, all login attempts that occurred on this day and the day before 2022-05-08 have been retrieved. The next screenshot shows the SQL query I used to filter for login attempts 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      | succe
ss |
+----------+----------+------------+------------+---------+-----------------+------
---+
|        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 |
```

In the first part, my SQL query is shown, followed by part of the output. In the output, the data of all login attempts that occurred on 2022-05-09 or 2022-05-08 is displayed. The code starts with selecting all columns from the log_in_attempts table. Following this, I perform filtering using the `WHERE` and `OR` clauses to retrieve the access data for both dates. The two conditions (`login_date = '2022-05-09', login_date = '2022-05-08'`)specify the dates to be filtered.

Retrieve login attempts outside of Mexico

After investigating both dates, it is believed that there is an issue with the login attempts that occurred outside of Mexico. Therefore, these were investigated next. The following code shows my SQL query used to retrieve the data of login attempts outside of Mexico.

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+------
---+
| event_id | username | login_date | login_time | country | ip_address      | succe
ss |
+----------+----------+------------+------------+---------+-----------------+------
---+
|        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 |
```

The first part of the screenshot shows my SQL query, followed by part of the output. First, all columns from the `log_in_attempts` table are selected. Then, the data is filtered to include only countries outside of Mexico using the `WHERE NOT` clause (to exclude the country indicated in the "country" column) and the `LIKE` clause (to specify the pattern that will not be displayed). The selected pattern is `MEX%`. Since Mexico appears in the database as both `MEX`

and `MEXICO`, the percentage sign (`%`) must be used to represent an unspecified number of characters following the `MEX` pattern.

Retrieve employees in Marketing

My team wants to update the employees' machines, specifically those in the Marketing department located in the East building offices. To determine which machines to update, I need to search for information using these filters. The following code shows the SQL query I used to extract this information.

```
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 |
```

The first part of the screenshot shows the SQL query used, followed by part of the output. First, I selected all the columns from the `employees` table. After that, to filter, I used the `WHERE` clause (`department = 'Marketing'`) with `AND`, to specify the selection of the Marketing department along with the selection of the East building offices. For the search of specific offices, I used `LIKE` with the pattern `East%` (`office LIKE 'East%'`). This will display in the office column only the data that contains the `"East"` pattern followed by an unspecified number of characters.

Retrieve employees in Finance or Sales

After this, the machines of employees in the Finance or Sales departments also need different updates. To achieve that, the information from these two departments is required. The following code shows the SQL query used to retrieve this data.

```
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   |
```

In the first part my query is shown, followed by part of the output. First, I selected all the columns from the `employees` table. Then, I used the `WHERE` clause with `OR`. The operator `OR` was used to display data that belongs to one department or the other. The first condition indicates the employees from the Finance department (`department = 'Finance'`) and the second condition indicates the employees from the Sales department (`department = 'Sales'`).

Retrieve all employees not in IT

Finally, my team needs to perform one last update for employees who do not belong to the Information Technology department. The information for these updates is obtained using the following SQL query:

```
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   |
```

In the first part, my query is shown, followed by part of the output. The query returns all the data of employees who do not belong to the Information Technology department. First, all columns from the `employees` table are selected. After that, the query is filtered using the `WHERE NOT` clause to retrieve the data of employees who do not work in this department.

Summary

To meet the assigned objectives of obtaining information about login attempts and employee machines, a series of SQL queries were used. Two tables were investigated: `log_in_attempts` and `employees`. To accomplish this, I used the `WHERE` clause along with

the `AND`, `OR`, and `NOT` operators to filter the information. Additionally, the `LIKE` clause and the percentage sign `(%)` wildcard were used to filter based on character patterns.