Apply filters to SQL queries

Project description

In this project, I need to obtain specific information about employees, their machines, and the departments they belong to from the database. My team needs to investigate potential security issues and update computers. I am responsible for filtering the required information in the database.

Retrieve after hours failed login attempts

There was a potential security incident that occurred after business hours (after 18:00). All after 0-hours login attempts that failed need to be investigated.

I used this command to retrieve after hours failed login attempts: `SELECT *  FROM log_in_attempts WHERE login_time > '18:00' AND success = 0;`

```
MariaDB [organization]> clear
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     | su
ccess |
+----------+----------+------------+------------+---------+----------------+---
------+
|        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 |
```

Explanation:
SELECT Statement: SELECT *: This part of the query is used to specify which columns from the table should be included in the result set. The asterisk (*) is a wildcard character that means "all columns". So, this query is asking for all columns of the selected rows to be returned.
FROM log_in_attempts: This specifies the source table from which the data should be retrieved. In this case, the table is named log_in_attempts.
WHERE login_time > '18:00' AND success = 0: This is the filter condition that determines which rows from the log_in_attempts table should be included in the result set.

- Login_time > '18:00': This condition filters the records to include only those where the login_time is later than 6:00 PM. The login_time field presumably stores the time when a login attempt was made. The comparison > means "greater than".
- AND: This is a logical operator that combines multiple conditions in the WHERE clause. For a row to satisfy this combined condition, it must satisfy both of the individual conditions.
- success = 0: This condition filters the records to include only those where the success field is equal to 0. Assuming that success is a field indicating whether the login attempt was successful, a value of 0 likely represents a failed attempt.

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.

I used this command to retrieve login attempts on specific dates:

```
SELECT *
-> FROM log_in_attempts
-> WHERE login_date = '2022-05-08' OR login_date =
'2022-05-09';
```

```
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      | su
ccess |
+----------+----------+------------+------------+---------+-----------------+---
------+
|        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 |
```

- SELECT *: This part of the query specifies which columns from the table should be included in the result set. The asterisk (*) is a wildcard character that signifies "all columns". Therefore, this query requests all columns of the selected rows to be returned.
- FROM log_in_attempts: This indicates the source table from which the data should be retrieved. In this case, the table is named log_in_attempts.
- WHERE login_date = '2022-05-08' OR login_date = '2022-05-09': This clause sets the conditions for filtering the data.
- The query filters the records to include only those where the login_date column matches either '2022-05-08' or '2022-05-09'.

- login_date = '2022-05-08': This condition selects rows where the login_date is exactly May 8, 2022.
- login_date = '2022-05-09': Similarly, this condition selects rows where the login_date is exactly May 9, 2022.
- OR: This is a logical operator used to combine two conditions. A row will satisfy the overall condition if it meets either of the individual conditions.

Retrieve login attempts outside of Mexico

After investigating the organization's data on login attempts, I believe there is an issue with the login attempts that occurred outside of Mexico. These login attempts should be investigated.

I used this command to retrieve login attempts outside of Mexico:

```
    SELECT *
   -> FROM log_in_attempts
   -> WHERE NOT country LIKE 'MEX%';
```

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

- SELECT *: This part of the query specifies which columns from the table should be included in the result set. The asterisk (*) is a wildcard character that means "all columns". So, this query is asking for all columns of the selected rows to be returned.
- FROM log_in_attempts: This indicates the source table from which the data should be retrieved. The table is named log_in_attempts, which likely contains records of login attempts made in an application or system.
- WHERE NOT country LIKE 'MEX%': This is the filter condition that determines which rows from the log_in_attempts table should be included in the result set.
- NOT: This is a negation operator. It inverts the condition that follows it. In this case, it is used to exclude rows that match the subsequent LIKE condition.
- country LIKE 'MEX%': This condition uses the LIKE operator for pattern matching. It filters the records to include only those where the country column starts with 'MEX'.

The % symbol is a wildcard character in SQL that represents any sequence of characters. So 'MEX%' will match any value that begins with 'MEX', such as 'Mexico', 'MEX123', etc.

Retrieve employees in Marketing

My team wants to update the computers for certain employees in the Marketing department.
To do this, I have to get information on which employee machines to update.

I used this command to retrieve employees in marketing:

```
SELECT *
FROM employees
WHERE department = 'Marketing'
AND office LIKE 'East-%';
```

```
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 |
```

- SELECT *: Selects all columns from the matching rows in the employees table.
- FROM employees: Specifies that the data should be retrieved from the employees table.
- WHERE department = 'Marketing': Filters the rows to include only those where the department is 'Marketing'.
- AND office LIKE 'East-%': Adds an additional filter to include only those rows where the office starts with 'East-'. The % is a wildcard character in SQL that matches any sequence of characters following 'East-'.

This query will return all records from the employees table where the department is 'Marketing' and the office is in the East building, as indicated by the office code starting with 'East-'.

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.

I used this command to Retrieve employees in Finance or Sales:

```
SELECT *
FROM employees
WHERE department = 'Finance'
OR department = 'Sales';
```

```
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   |
```

- SELECT *: This selects all columns from the employees table.
- FROM employees: This specifies that the data is being retrieved from the employees table.
- WHERE department = 'Finance': This condition filters the records to include only those where the department column is 'Finance'.
- OR department = 'Sales': This condition further extends the filter to also include records where the department column is 'Sales'.

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.

I used this command to retrieve all employees not in IT:

```
SELECT *
-> FROM employees
```

```
-> WHERE NOT department = 'Information Technology';
```

```
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   |
```

- SELECT *: This selects all columns from the employees table.
- FROM employees: This specifies that the data is being retrieved from the employees table.
- WHERE NOT department = 'Information Technology': This condition filters the records to include only those where the department is not 'Information Technology'. The NOT operator negates the condition that follows, thus excluding the 'Information Technology' department.

Summary

I have used multiple operators, the SELECT statement, wildcard characters, and clauses to obtain information such as employees that are not in IT, employees who are in the finance or sales department, employees who are in the marketing department, and login attempts made outside of Mexico. I also retrieved information from specific dates and after hours failed login attempts.