Apply filters to SQL queries

Everton Oliveira

Project description

SQL (Structured Query Language) is a language used to manage and manipulate relational databases. With SQL, you can query, insert, update, and delete data, as well as create and manage tables. It allows filtering information, joining tables, optimizing queries, and ensuring data security through access permissions. SQL is essential for data analysis, report generation, and cybersecurity investigations.

Retrieve after hours failed login attempts

First, here I am retrieving failed login attempts made after business hours (6:00 PM) from the `log_in_attempts` table. The `login_time` column records the time of the login attempts, and the `success` column indicates whether the login was successful (1 for TRUE, 0 for FALSE).

By using an SQL query with the **AND** clause, you can filter the failed login attempts (value 0 in the `success` column) that occurred after 6:00 PM. The query could look like this:

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00:00' AND success = 0;
```

This command will return all failed login attempts that occurred after business hours.

Retrieve login attempts on specific dates

Then, I am investigating a suspicious event and want to retrieve all login attempts that occurred on two specific dates: **2022-05-09** and **2022-05-08**. The `login_date` column in the `log_in_attempts` table contains the dates when the login attempts took place.

To filter the login attempts on these specified dates, you can use the **OR** clause in an SQL query to retrieve records for both dates. The query would look like this:

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

This query will return all login attempts made on the specified dates, allowing you to analyze the suspicious event.In summary, you used **OR** to retrieve login attempts on the two specific dates by filtering the records based on the `login_date` column.

Retrieve login attempts outside of Mexico

I am investigating logins that did not originate from Mexico. Knowing that the country field can have entries like 'MEX' or 'MEXICO', I used the **NOT** clause with the **LIKE** operator and the pattern **'MEX%'**. This allows me to filter records where the country does not start with "MEX".

The SQL query I ran was:

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
```

With this, I was able to retrieve all login attempts that did not come from Mexico, excluding those where the `country` field starts with 'MEX'.

Retrieve employees in Marketing

```
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 (0.001 sec)
```

I wrote an SQL query to retrieve all the information about employees in the **Marketing** department who are located in offices in the **East** building. To do this, I used the **WHERE** clause with the **AND** and **LIKE** operators. The **LIKE 'East%'** operator allows me to include all offices

that start with "East," such as 'East-170', 'East-320', etc. This way, I was able to filter the correct records from the **employees** table.

Retrieve employees in Finance or Sales

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
```

I wrote an SQL query to select all records from the **employees** table for employees who belong to the **Finance** or **Sales** departments. I used the **WHERE** clause with the **OR** operator to filter for these two departments, ensuring that all employees in these areas were retrieved.

Retrieve all employees not in IT

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE NOT department = 'Information Technology';
```

Finally, I wrote an SQL query using the **NOT** operator to filter records of employees who do not belong to the **Information Technology** department. This allows me to retrieve information from all other departments.

Summary

In this SQL project, I developed queries to filter records of login attempts made after business hours, identify attempts on specific dates, exclude records originating from Mexico, retrieve information about employees in the Finance or Sales departments, and identify employees outside the Information Technology department. I utilized logical operators such as **AND**, **OR**, **NOT**, and **LIKE** to refine and combine conditions, enabling the extraction of specific data according to the project's requirements. These operations enhanced the precision of the queries and contributed to a more efficient data analysis process.