

Apply filters to SQL queries

Project description

As part of my role in enhancing the security of our organization's system, I am responsible for ensuring its safety, investigating potential security concerns, and performing necessary updates on employee computers. To accomplish these tasks, I have leveraged SQL queries with filters to address various security-related requirements. The subsequent steps outline specific instances where I applied SQL filtering techniques to carry out these tasks effectively.

Retrieve after hours failed login attempts

Following a potential security incident that took place after business hours (specifically after 18:00), it is crucial to investigate all failed login attempts during that time. To address this, I have devised a SQL query that filters and identifies failed login attempts occurring after the specified business hours.

```
MariaDB [organization]> Select *  
-> from log_in_attempts  
-> WHERE login_time > "18:00"  
-> AND success = 0;
```

This query selects all columns (*) from the log_in_attempts table where the login_time is greater than '18:00:00' (6:00 PM) and the success column is 0

Retrieve login attempts on specific dates

A suspicious event took place on 2022-05-09, requiring investigation of any login activity that occurred on that day or the preceding day. To address this, I have developed a SQL query that filters and identifies login attempts happening on the specified dates.

```
MariaDB [organization]> Select *  
-> From log_in_attempts  
-> Where login_date  
-> Between "2022-05-08" AND "2022-05-09";
```

This query selects all columns (*) from the log_in_attempts table where the login_date was between 2022-05-08 AND 2022-05-09

Retrieve login attempts outside of Mexico

Upon thorough examination of the organization's login attempt data, I have identified a potential issue with login attempts originating from outside of Mexico. It is crucial to conduct an investigation regarding these login attempts. Below is an example of the SQL query I formulated to filter and identify login attempts that occurred outside of Mexico.

```
MariaDB [organization]> Select *
-> from log_in_attempts
-> WHERE country NOT LIKE "MEX" ;
```

This query selects all columns (*) from the log_in_attempts table where the country column does not contain the string 'MEX'

Retrieve employees in Marketing

To facilitate the computer updates for specific employees in the Marketing department, my team requires information about the machines that need to be updated. In order to retrieve this information, I have developed a SQL query that filters and identifies employee machines in the Marketing department within the East building. The following code demonstrates how I constructed this SQL query.

```
MariaDB [organization]> SELECT *  
->  
-> FROM employees  
->  
-> WHERE department LIKE '%Marketing%' AND office LIKE 'East%';
```

This query selects all columns (*) from the employees table where the department column contains the string 'Marketing' (case-insensitive) and the office column starts with 'East'.

Retrieve employees in Finance or Sales

In addition to updating the machines for employees in the Marketing department, it is also necessary to update the machines for employees in the Finance and Sales departments. As a different security update is required for these departments, I need to gather information specifically on employees from Finance and Sales. The following code showcases the SQL query I formulated to filter and retrieve employee machines belonging to the Finance or Sales departments.

```
MariaDB [organization]> Select *  
-> from employees  
-> WHERE department LIKE "Sales"  
-> OR  
-> department LIKE "Finance" ;
```

This query selects all columns (*) from the employees table where the department column has a value of either 'Sales' or 'Finance'.

Retrieve all employees not in IT

To proceed with the final security update on employees who are not part of the Information Technology department, it is essential to gather information about these employees. In order to retrieve this information, I have formulated a SQL query that filters and identifies employee machines belonging to departments other than Information Technology. The following demonstrates the code I used to create this SQL query.

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

This query selects all columns (*) from the employees table where the department column does not contain the string 'Information Technology' (case-insensitive) anywhere within it. The use of the NOT LIKE operator in combination with % before and after 'Information Technology' allows us to exclude any employees in the IT department from the result.

Summary

By applying filters to SQL queries, I successfully obtained targeted information regarding login attempts and employee machines. Two distinct tables, namely `log_in_attempts` and `employees`, were utilized for this purpose. To filter the data effectively, I employed operators such as `AND`, `OR`, and `NOT` to specify the desired criteria for each task. Additionally, I utilized the `LIKE` operator along with the percentage sign (%) wildcard to filter for specific patterns in the data.