Apply filters to SQL queries

Project description

My team identified potential security issues involving login attempts and employee machines. To investigate and resolve these issues, I examined the organization's data in the employees and log_in_attempts tables using SQL filters to retrieve specific records.

Retrieve after hours failed login attempts

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

My organization's business hours end at 6PM. To investigate a security incident that occurred after business hours, I used the SQL query pictured above to filter the login_in_attempts table for unsuccessful attempts made after business hours. Using the operator > excludes all login attempts made during business hours, and by setting the condition of success to = FALSE, I have filtered out successful login attempts. Using the operator AND ensures that the output includes only results that meet both conditions.

Retrieve login attempts on specific dates

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

Upon notification of a suspicious event that occurred on 2022-05-09, I began investigating by reviewing all login attempts on the day of the incident and the day preceding it. Using the operator OR in the SQL statement above produced all the login attempts that occurred on either 2022-05-09 or 2022-05-08.

Retrieve login attempts outside of Mexico

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

After determining that the suspicious login activity did not originate in Mexico, I created a SQL query that would exclude all login attempts that occurred within Mexico. I used the operator NOT to negate the condition of originating in Mexico. Because the country column of the log_in_attempts table has Mexico represented as MEX and MEXICO, I used the operator LIKE and the wildcard % after MEX to exclude results containing either MEX or MEXICO in the country column. The operator LIKE is used with WHERE to search for a pattern in a column. In this case, the pattern would be values in the country column beginning with MEX. Using % indicates that the letters MEX can be followed by any number of additional characters.

Retrieve employees in Marketing

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

My team decided to perform security updates on machines for employees in the Marketing department with offices in the East building. I used the operator AND in the SQL query above to ensure the output satisfies both conditions by listing only information for employees who work in the Marketing department and have offices in the East building. I used the operator LIKE and the wildcard % after East to include results for all employees whose offices begin with “East”, such as East-170, East-320, etc. Using the % allowed me to obtain all results for Marketing employees who work in the East building without having to know the exact numerical characters following the word East in each office name because the % substitutes for any number of other characters.

Retrieve employees in Finance or Sales

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

My team needed to perform security updates on machines for employees in the Sales and Finance departments. I used the operator OR in the query above to return results for employees that work in either department. If I had used the operator AND, only employees with both departments listed in the department column would be returned. Using the operator OR indicates that the results can meet one or both of the conditions.

Retrieve all employees not in IT

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

Employees in the Information Technology department already received an update that needed to be extended to machines in the rest of the organization's departments. I used the query above to identify all employees that need to receive the update by using the operator NOT to exclude all employees who work in Information Technology.

Summary

AND, OR, and NOT are logical operators that can be implemented to specify conditions when filtering numerical or data and time data using SQL queries. I used these operators to query the employees and log_in_attempts tables to identify machines in need of updates and filter login attempts based on specific criteria as part of an investigation into a cybersecurity incident.