Apply filters to SQL queries

Project description

The main goal of this project is to use SQL with filters to investigate potential security issues. Below are the results of a simulated incident investigations where I query a database with MariaDB.

Retrieve after hours failed login attempts

Here a potential security incident occurred after business hours (18:00-) and I investigated by querying the database for all after hours failed login attempts.

```
MariaDB [organization]> clear
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      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.57   |       0 |
|       34 | drosas   | 2022-05-11 | 21:02:04   | US      | 192.168.45.93   |       0 |
```

The first few lines in the screenshot are my query and the second part is a portion of the returned output. Firstly, I extracted all data from the `log_in_attempts` table by using the `SELECT *` statement. Then a `WHERE` clause was used together with and `AND` operator to filter the results to display only unsuccessful logins after 18:00. The two conditions I used were `login_time > '18:00'` to filter out all login attempts that occurred before 18:00 and `success = FALSE` in order to limit the output to failed login attempts only.

Retrieve login attempts on specific dates

In the next scenario a suspicious event occurred on 2022-05-09 and I decided to investigate by querying the database for all login activity that happened on 2022-05-09 and on the day before. My SQL query to filter for logins on specific dates was as follows:

```
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      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
|       12 | dkot     | 2022-05-08 | 09:11:34   | USA     | 192.168.100.158 |       1 |
```

This next query returns all login attempts that occurred on 2022-05-09 or 2022-05-08. Here I selected all entries in the `log_in_attempts` table and used a `WHERE` clause with an `OR` operator to filter the output for logins on either 2022-05-09 or 2022-05-08. I used the `OR` operator instead of `AND` in order to get all logins from both dates. The first condition is `login_date = '2022-05-09'` and filters for login attempts on the day of the suspicious event while the second condition is `login_date = '2022-05-08'` and filters for logins on the day before.

Retrieve login attempts outside of Mexico

During its investigation the team determines that the suspicious activity did not originate in Mexico, so I am tasked with filtering for all login attempts outside of Mexico. I do this using the following SQL query:

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
|        5 | jrafael  | 2022-05-11 | 03:05:59   | CANADA  | 192.168.86.232  |       0 |
```

I got this output by selecting all data from the `log_in_attempts` table and applying a `WHERE` clause with `NOT` to filter for countries other than Mexico. Because the dataset represents Mexico as both `MEX` and `MEXICO`, I used the `LIKE` operator with the pattern `MEX%` to filter out both results. In this case, the percent sign wildcard (`%`) represents any number of unspecified characters after `MEX`.

Retrieve employees in Marketing

In this scenario my team's goal is to update the OS of machines belonging only to employees in the Marketing department located in the East building. I used the following SQL query to identify which machines need to be updated:

```
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 |
```

I selected all data from the `employees` table by using the `SELECT *` statement. Then I used the `WHERE` clause with the `AND` operator to filter for employees who work in Marketing at the East building. By adding the `LIKE` operator with `East%` as the pattern, I managed to display all offices in the East building in the appropriate column. This was necessary because in the database all offices are indexed with their respective unique numbers. Therefore, the two conditions for getting the correct information were `department = 'Marketing'` and `office LIKE 'East%'`.

Retrieve employees in Finance or Sales

A different update was also requested for the Finance and Sales departments. Here I need to limit my results to only these two departments. The following code demonstrates how I went about this:

```
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   |
|        1009 | NULL         | lrodriqu | Sales      | South-134   |
|        1010 | k2421212m542 | jlansky  | Finance    | South-109   |
```

Once again, I started by selecting all data from the `employees` table. Then, I used the `OR` operator with the `WHERE` clause because I need to extract the information all employees who are in either department. The first condition is `department = 'Finance'`, which filters for employees from the Finance department. The second condition is `department = 'Sales'`, which filters for employees from the Sales department.

Retrieve all employees not in IT

The last security update my team requested was for all employees outside of the IT department. This can easily be achieved with the following 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   |
|        1003 | d394e816f943 | sgilmore | Finance         | South-153   |
|        1004 | e218f877g788 | eraab    | Human Resources | South-127   |
```

The first part of the screenshot is the query, and the second part is a portion of the output. Here I utilized a `WHERE` clause with `NOT` operator to filter to filter out everyone from the `employees` table who does not work in Information Technology.

Summary

In this project I used filtered SQL queries to get specific information on login attempts and employee machines. The two tables used were `log_in_attempts` and `employees`. The `AND`, `OR`, and `NOT` operators were used to filter for the specific information I needed for each scenario. Additionally, the `LIKE` operator and the percent sign (`%`) wildcard were utilized to filter for patterns.