Portfolio work, Manuela Kesten, 10.2.2024.

Apply filters to SQL queries

Project description

My organization is working to make their system more secure. My job is to ensure the system is safe, update employee computers as needed and investigate all potential security issues. The following steps show examples of how I used SQL with filters to perform security related tasks.

Retrieve after hours failed login attempts

There was potential security incident which happened after business hours, after 18:00. All after hours login attempts that failed need to be investigated.
In the code below I will demonstrate how I created SQL query to filter for failed login attempts that occurred after business hours.

```
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      | 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 |
```

My query is the first part of the screenshot, and the second part is a portion of the output. This query filters for failed login attempts that occurred after 18:00. First, I started by selecting all data from the `log_in_attempts` table. Then, I used a `WHERE` clause with an `AND` operator to filter my output to show only login attempts that occurred after 18:00 and were unsuccessful.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. I need to investigate all login attempts that occurred on that day or on the day before.
In the code below I demonstrate how I created a SQL query to filter for login attempts that occurred on specific dates.

```
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 |
```

This query returns all login attempts that occurred on 2022-05-09 or 2022-05-08. First, I started by selecting all data from the `log_in_attempts` table. Then, I used a `WHERE` clause with an `OR` operator to filter my results to output only login attempts that occurred on either 2022-05-09 or 2022-05-08. The first condition is `login_date = '2022-05-09'`, which filters for logins on 2022-05-09. The second condition is `login_date = '2022-05-08'`, which filters for logins on 2022-05-08.

Retrieve login attempts outside of USA

After investigating the organization's data on login attempts, I believe there is an issue with the login attempts that occurred outside of USA. I need to investigate these login attempts.
The following code shows how I created SQL query to filter for login attempts that occurred outside of USA.

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'US%';
+----------+----------+------------+------------+---------+-----------------+---------+
| 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 |
|        5 | jrafael  | 2022-05-11 | 03:05:59   | CANADA  | 192.168.86.232  |       0 |
```

The first part of the screenshot is my query, and the second part is a portion of the output. This query returns all login attempts that occurred in countries other than USA. First, I started by selecting all data from the `log_in_attempts` table. Then, I used a `WHERE` clause with `NOT` to filter for countries other than Mexico. I used `LIKE` with `US%` as the pattern to match because the dataset represents USA as `US` and `USA`. The percentage sign (`%`) represents any number of unspecified characters when used with `LIKE`.

Retrieve employees in Marketing

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

Code below demonstrates how I created SQL query to filter for employee machines from employees in the Marketing department in the East building.

```
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 |
```

The first part of the query shows my input, and the second part is portion of output. This query returns all employees in the Marketing department in the East building. First, I started with selecting all data from the `employees`. Then, I used a `WHERE` clause with `AND` to filter for employees who work in Marketing department in the East building. `LIKE` and `East%` filtered `office` column to show only East buildings with specific office numbers.

Retrieve employees in Sales or Finance

Machines in departments in Sales and Finance also needs to be updated.
Following code shows how I created SQL query to filter for employee machines only from these two departments.

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

First, I started with selecting all data from `employees` table. With the rest of query, I filtered to show only departments in `department` column with Sales or Finance.

Retrieve all employees not in IT

Team needs to make one more security update on employees who are not in the department of Information Technology. Code below shows how I created SQL query to filter those machines.

```
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   |
```

The query returns all employees that are not in the Information Technology department. First, I started with selecting all data from `employees` table. Then I used a `WHERE` clause with `NOT` to filter for all employees except those in this department.

Summary

I applied filters to SQL queries to get specific information on login attempts and employee machines. I used two different tables, `employees` and `log_in_attempts`. I used `AND`, `OR` and `NOT` operators to filter for information needed. Also, I used `LIKE` and the percentage sign (`%`) wildcard to filter for patterns.

Table formats

This document describes how the tables used for this portfolio activity are organized. The `organization` database contains the following two tables:

- `log_in_attempts`
- `employees`

log_in_attempts

The `log_in_attempts` table has the following columns:

- `event_id`: The identification number assigned to each login event
- `username`: The username of the employee
- `login_date`: The date the login attempt was recorded
- `login_time`: The time the login attempt was recorded
- `country`: The country where the login attempt occurred
- `ip_address`: The IP address of that employee's machine
- `success`: The success of the login attempt; `FALSE` indicates a failed attempt

employees

The `employees` table has the following columns:

- `employee_id`: The identification number assigned to each employee
- `device_id`: The identification number assigned to each device used by the employee
- `username`: The username of the employee
- `department`: The department the employee is in
- `office`: The office the employee is located in