Apply filters to SQL queries

Project description

You recently discovered a potential security incident that occurred after business hours. To investigate this, you need to query the **`log_in_attempts`** table and review after hours login activity. Use filters in SQL to create a query that identifies all failed login attempts that occurred after 18:00. (The time of the login attempt is found in the **`login_time`** column. The **`success`** column contains a value of `0` when a login attempt failed; you can use either a value of `0` or **`FALSE`** in your query to identify failed login attempts.)

Retrieve after hours failed login attempts

There was a potential security incident that occurred after business hours (18:00). All after hours login attempts that failed needed to be investigated.

```
SELECT *
FROM log_in_attempts
WHERE login_time > '18:00' AND success = False;
```

```
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 |
```

Data was selected from the `log_in_attempts` table. Then, I used a `WHERE` clause with an `AND` operator to filter the results to display only login attempts that occurred after 18:00 and were unsuccessful. The first condition is `login_time > '18:00'` which filters times outside of 18:00. `The` second condition is `success = FALSE`, which filters for the failed login attempts.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or on the day before needed to be investigated.

```
SELECT *
FROM log_in_attempts
WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

```
ariaDB [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 |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       0 |
|        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. Data was selected from the `log_in_attempts` table. A `WHERE` clause with an `OR` operator to filter results to display only login attempts that occurred on either 2022-05-09 or 2022-05-08.

Retrieve login attempts outside of Mexico

After investigating the organization's data on login attempts, I believe there is an issue with the login attempts that occurred outside of Mexico. These login attempts were investigated.

```
SELECT *
FROM log_in_attempts
WHERE NOT country LIKE 'Mex%';
```

```
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 |       0 |
|        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 |       0 |
```

This query returns all login attempts that occurred in countries other than Mexico. Data was selected from the `log_in_attempts` table. A `WHERE` clause with `NOT` to filter for countries other than Mexico. The data was filtered using the `LIKE` operator with `MEX%` as the pattern to match because the dataset represents Mexico as `MEX` and `MEXICO`.

Retrieve employees in Marketing

The team needed to update the computers for certain employees in the Marketing department. A list was gathered containing the devices in the Marketing department and the East Building.

```
SELECT *
FROM employees;
WHERE department = 'Marketing' AND office LIKE 'East%' ;
```

```
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 |
```

This query returns all employees in the Marketing department in the East building. Data was selected from the `employees` table. The `WHERE` clause with `AND` to filter for employees who work in the Marketing department and in the East building. The data was filtered using the `LIKE` operator with `East%` as the pattern to match the East building.

Retrieve employees in Finance or Sales

```
SELECT *
FROM employees;
WHERE department = 'Finance' OR department = 'Sales' ;
```

```
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 | i858i583k571 | abernard | Finance    | South-170   |
```

This query returns all employees in the Finance and Sales departments. Data was selected from the `employees` table. A `WHERE` clause with `OR` to filter for employees who are in the Finance and Sales departments. I used the `OR` operator rather `AND` because we are filtering for either the Finance or Sales department.

Retrieve all employees not in IT

```
SELECT *
FROM employees;
WHERE NOT department = LIKE ‘Information Technology’ ;
```

```
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 not in the Information Technology department. The data was selected from the `employees` table. A `WHERE` clause with `NOT` to filter for employees not in the IT department.

Summary

SQL queries with filters to retrieve specific information on login attempts and employee machines. Two tables were utilized, `log_in_attempts` and `employees`. Operators `LIKE, AND, OR,` and `NOT` operators to filter for the specific information needed for each task. The percentage sign (`%`) wildcard to filter for patterns was also used.