Apply filters to SQL queries

Project description

In this scenario, my job is to analyze data from two tables (“log_in_attempts” and “employees”) and investigate any potential security issues and update employee computers when required. The main goal is to make the organization's systems more secure. To complete this task, I applied filters to SQL queries and performed the following tasks:

Retrieve after hours failed login attempts

I applied the following filter for failed login attempts that occurred after business hours:

```
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 |
```

I started by selecting all data with `*` from the `log_in_attempts` table. I then used `WHERE` and `AND` to filter for only login attempts after business hours (`> ‘18:00’`) and were unsuccessful (`success = FALSE`).

Retrieve login attempts on specific dates

I applied the following filter for login attempts that occurred on either 2022-05-09 or 2022-05-08:

```
SELECT *
-> FROM log_in_attempts
-> WHERE login_date = ‘2022-05-09’ OR login_date = ‘2022-05-08’;
```

```
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 |       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 |
```

I started by selecting all data with `*` from the `log_in_attempts` table. I then used `WHERE` and `OR` to filter for only login attempts on either 2022-05-09 (`'2022-05-09'`) or 2022-05-08 (`'2022-05-08'`).

Retrieve login attempts outside of Mexico

I applied the following filter for login attempts that occurred outside Mexico:

```
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 |
```

I started by selecting all data with `*` from the `log_in_attempts` table. I then used `WHERE` and `NOT` to filter for login attempts from countries other than Mexico. I used `LIKE` with `MEX%` as Mexico is represented by both `MEX` and `MEXICO` in the table.

Retrieve employees in Marketing

I applied the following filter for employee machines from employees in the Marketing department in 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 |
```

I started by selecting all data with `*` from the `employees` table. I then used `WHERE` and `AND` to filter for employees who work in the Marketing department (`department = 'Marketing'`) and are located in the East Building (`office LIKE 'EAST%'`). I used `EAST%` as each office entry includes the specific office number.

Retrieve employees in Finance or Sales

I applied the following filter for employee machines from employees in the Finance or Sales departments:

```
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 | i858j583k571 | abernard | Finance    | South-170   |
```

I started by selecting all data with `*` from the `employees` table. I then used `WHERE` and `OR` to filter for employees from either the Finance department (`department = 'Finance'`) or the Sales department (`department = 'Sales'`).

Retrieve all employees not in IT

I applied the following filter for employee machines from employees not in the Information Technology department:

```
SELECT *
```

-> `FROM employees`
-> `WHERE NOT department = '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   |
```

I started by selecting all data with `*` from the `employees` table. I then used `WHERE` and `NOT` to filter for employees from departments other than Information Technology.

Summary

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