Apply filters to SQL queries

Project description

This project demonstrates using SQL to analyze login and employee data for security purposes. By applying filters and pattern-matching techniques, I identify suspicious login attempts, review departmental employee records, and gather insights critical for incident investigation and system updates. These tasks highlight the importance of SQL in cybersecurity.

Retrieve after-hours failed login attempts

(In all screenshots the first part is my query and the second part is a portion of the output)

```
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 |
```

Explanation:

- **Filtering for dates and times:** The condition `login_time > '18:00'` filters for login attempts that occurred after 6 PM. Time-based filtering helps identify activity outside regular business hours.
- **Using AND to filter on multiple conditions:** The `AND` operator combines `success = FALSE` (or = 0, indicates failed attempts) with the time filter, ensuring that only failed logins after 6 PM are retrieved.

This query highlights failed login attempts during non-business hours, which could indicate potential unauthorized access.

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';
+----------+----------+------------+------------+---------+----------------+---------+
| 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 |
```

Explanation:

- **Filtering for dates:** The `WHERE login_date = '2022-05-09' OR login_date = '2022-05-08'` clause retrieves records for specific days. This is useful for investigating specific incidents that occurred on known dates.
- **Using OR to filter on multiple conditions:** The `OR` operator ensures records from either of the two specified dates are included in the results.

This query helps investigate login attempts from both days to identify patterns or unusual activity.

Retrieve login attempts outside of Mexico

```
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 |
```

Explanation:

- **Using LIKE to search for a pattern:** The `LIKE` keyword matches values that fit a specified pattern. `NOT LIKE 'MEX%'` excludes countries that start with "MEX," capturing both "MEX" and "MEXICO."
- **Using NOT in filters:** The `NOT` keyword reverses the match from `LIKE`, ensuring the query only includes results not originating from Mexico.

This query ensures login attempts originating from outside Mexico are isolated for further investigation.

Retrieve employees in Marketing

```
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 |
```

Explanation:

- **Using = or to search for a pattern:** `= 'Marketing'` identifies only the marketing department. `LIKE 'East%'` filters for offices in the East building, where the pattern starts with "East-" and can include various office numbers.
- **Using AND to combine conditions:** The `AND` operator ensures only employees meeting both criteria are retrieved.

This query gathers a precise list of Marketing employees in the East building, which is essential for targeted updates.

Retrieve employees in Finance or 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   |
```

Explanation:

- `department = 'Finance'` and `department = 'Sales'` match employees in Finance and Sales.
- The `OR` operator combines both conditions to include either department.

This query lists employees in the Finance and Sales departments for targeted updates.

Retrieve all employees not in IT

```
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   |
```

Explanation:

- The `= 'Information Technology'` clause matches employees in the IT department.
- **Using NOT in filters:** The `NOT` keyword excludes employees whose in the IT department.

This query identifies all employees who are not part of IT, allowing for updates targeting other departments.

Summary

In this project, I used SQL queries to filter and analyze data for cybersecurity investigations and employee updates. I demonstrated the use of the `LIKE` keyword for pattern matching, `NOT` for exclusion, and `AND` and `OR` for combining multiple conditions. Additionally, I applied filters for dates and times to target specific incidents. This project showcases the role of SQL in cybersecurity by effectively retrieving relevant data for analysis and action.