Apply filters to SQL queries

Project description

As a security professional at a large organization, I was tasked with investigating potential security issues involving login attempts and employee machines. This project demonstrates my ability to use SQL filters to retrieve specific records from organizational databases and conduct thorough security investigations.

The investigation involved analyzing data from two main tables:

- `employees`: Contains employee information including departments and office locations
- `log_in_attempts:` Contains login attempt records with timestamps, success status, and geographic data

Through this investigation, I utilized various SQL filtering techniques including AND, OR, and NOT operators, pattern matching with LIKE, and date/time filtering to identify security concerns and support organizational security updates.

Retrieve after hours failed login attempts

Objective: Investigate potential security incidents by identifying failed login attempts that occurred after business hours (after 18:00).

```
SELECT *

FROM log_in_attempts

WHERE login_time > '18:00' AND success = 0;
```

This query filters the `log_in_attempts` table to show only records where two conditions are met simultaneously using the AND operator:

`login_time > '18:00'` - Identifies attempts made after 6:00 PM

`success = 0` - Identifies failed login attempts (0 indicates failure)

The AND operator ensures both conditions must be true for a record to be included in the results. This helps identify potentially suspicious activity occurring outside normal business hours, which could indicate unauthorized access attempts or compromised accounts.

Retrieve login attempts on specific dates

Objective: Investigate a suspicious event by reviewing all login attempts on 2022-05-09 and the preceding day (2022-05-08).

```
SELECT *

FROM log_in_attempts

WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

Query Explanation: This query uses the OR operator to retrieve login attempts from either of two specific dates. The OR operator allows the query to return records that match at least one of the specified conditions:

- Records with `login_date = '2022-05-09'`
- Records with `login_date = '2022-05-08'`

This approach is essential for investigating incidents that may have occurred over multiple days or when analyzing patterns of suspicious activity across a specific time period.

Retrieve login attempts outside of Mexico

Objective: Investigate login attempts that occurred outside of Mexico, as the team determined that suspicious activity did not originate from Mexico.

```
SELECT *

FROM log_in_attempts

WHERE NOT country LIKE 'MEX%';
```

Query Explanation: This query combines the NOT operator with the LIKE operator and pattern matching:

- `NOT` excludes records that match the specified condition
- `LIKE 'MEX%'` uses pattern matching where `%` is a wildcard representing any number of characters
- The pattern `'MEX%'` matches any country value starting with "MEX" (including "MEX" and "MEXICO")

This approach ensures we exclude all variations of Mexico entries while including all other countries in our investigation results.

Retrieve employees in Marketing

Objective: Identify Marketing department employees in the East building for targeted security updates.

```
SELECT *

FROM employees

WHERE department = 'Marketing' AND office LIKE 'East%';
```

Query Explanation: This query uses both exact matching and pattern matching:

- `department = 'Marketing'` performs exact matching for the Marketing department
- `office LIKE 'East%'` uses pattern matching to find all offices starting with "East"
- The AND operator ensures both conditions must be met

The `LIKE` operator with `%` wildcard is crucial here because office values include room numbers (like "East-170", "East-320"), and we need to match all East building locations regardless of the specific room number.

Retrieve employees in Finance or Sales

Objective: Identify employees in either Finance or Sales departments for a different security update.

```
SELECT *

FROM employees

WHERE department = 'Finance' OR department = 'Sales';
```

Query Explanation: This query uses the OR operator to retrieve employees from either department:

- Records where `department = 'Finance'`
- Records where `department = 'Sales'`

The OR operator is perfect for this scenario because we need employees from either department, not both simultaneously. This allows us to target multiple departments with a single query for efficiency.

Retrieve all employees not in IT

Objective: Identify all employees outside the Information Technology department who need a security update.

```
SELECT *

FROM employees

WHERE NOT department = 'Information Technology';
```

Query Explanation: This query uses the NOT operator to exclude employees from the IT department:

- `NOT department = 'Information Technology'` returns all records except those with the IT department
- This approach is more efficient than listing all other departments individually

The NOT operator is essential when you need to exclude specific criteria rather than include them, making it perfect for scenarios where one group has already received updates.

Summary

This investigation successfully demonstrated advanced SQL filtering techniques to address various security concerns within the organization. Through the systematic use of AND, OR, and NOT operators, along with pattern matching and date filtering, I was able to:

1. **Identify potential security threats** by analyzing after-hours failed login attempts
2. **Investigate specific incidents** by examining login patterns across targeted dates
3. **Exclude irrelevant data** by filtering out login attempts from Mexico
4. **Support security updates** by precisely targeting employee groups based on department and location criteria

The queries developed during this investigation showcase proficiency in:

- Complex conditional logic using multiple operators
- Pattern matching for flexible data retrieval
- Date and time-based filtering
- Exclusion-based queries for efficient data analysis

These SQL skills are essential for cybersecurity professionals who need to quickly analyze large datasets, identify security anomalies, and support organizational security initiatives. The ability

to write precise, efficient queries enables rapid response to security incidents and proactive security management.