Apply filters to SQL queries

Project description

In this portfolio activity, I use SQL filters (specifically the WHERE, AND, OR, NOT, and LIKE clauses) to query an organization's database containing log_in_attempts and employees tables. The objective is to apply these filters to investigate potential security incidents by retrieving specific subsets of data, such as failed logins outside of business hours, login attempts on specific dates, and employee information based on department and office location. This demonstrates proficiency in using conditional logic to narrow down result sets for targeted data analysis and investigation.

Retrieve after hours failed login attempts

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

Explanation:

This query retrieves all columns (*) from the log_in_attempts table. I use the WHERE clause to apply two conditions that must both be true for a row to be included in the results, connected by the AND operator:

login_time > '18:00:00': Filters for login attempts that occurred after 18:00:00 (6 PM).

success = FALSE: Filters for only failed login attempts. Using FALSE is equivalent to using 0 for a boolean/integer column indicating failure.

This effectively isolates failed login activity that occurred after business hours for security investigation.

Retrieve login attempts on specific dates

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

Explanation:

This query selects all columns (*) from the log_in_attempts table. I use the WHERE clause to filter the results based on the login_date.

The query uses the OR operator to connect two conditions: login_date = '2022-05-09' OR login_date = '2022-05-08'. A row is returned if at least one of these date conditions is true.

Retrieve login attempts outside of Mexico

SELECT * FROM log_in_attempts WHERE country NOT LIKE 'MEX%';

Explanation:

This query selects all columns (*) from the log_in_attempts table. I use the WHERE clause with the NOT LIKE operators to exclude login attempts originating from Mexico.

The LIKE keyword is used for pattern matching. Since the country values can be MEX or MEXICO, I use the pattern 'MEX%'. The % wildcard matches any sequence of zero or more characters.

This pattern ensures that any value starting with "MEX" (i.e., "MEX" itself or "MEXICO") is matched.

The NOT operator is applied to the LIKE expression: NOT LIKE 'MEX%'. This reverses the condition, returning rows where the country value does not match the pattern 'MEX%'. This successfully retrieves all login attempts that occurred outside of Mexico.

Retrieve employees in Marketing

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

Explanation:

This query retrieves all columns (*) from the employees table. I use the WHERE clause with two conditions connected by the AND operator, meaning both must be true:

department LIKE '%Marketing%': Filters for employees in the Marketing department. Since department names may contain "Marketing" as a part of a larger string (e.g., 'Digital Marketing'), I use the pattern '%Marketing%'. The % wildcard on both sides matches any characters before or after the word "Marketing".

office LIKE 'East%': Filters for employees located in any office in the East building. I use the pattern 'East%' to match any office location that starts with "East" (e.g., 'East-170', 'East-320').

This query successfully returns data for all employees who are in the Marketing department and are located in an East building office.

Retrieve employees in Finance or Sales

SELECT * FROM employees WHERE department LIKE '%Sales%' OR department LIKE '%Finance%';

Explanation:

This query selects all columns (*) from the employees table. I use the WHERE clause with the OR operator to connect the two department conditions. A row is included if the employee's department is either Sales or Finance.

department LIKE '%Sales%': Filters for employees in the Sales department (using % for flexibility, like 'Internal Sales').

department LIKE '%Finance%': Filters for employees in the Finance department (using % for flexibility).

Using OR ensures the result set includes employees from both the Sales department and the Finance department.

Retrieve all employees not in IT

SELECT * FROM employees WHERE department NOT LIKE '%Information Technology%';

Explanation:

This query retrieves all columns (*) from the employees table. I use the WHERE clause with the NOT LIKE operators to exclude employees in the IT department.

The pattern is defined as '%Information Technology%' to account for any variations in the department name that contain "Information Technology" (e.g., 'Global Information Technology').

Applying the NOT operator to the LIKE expression reverses the filter, ensuring the query returns employees whose department does not match the Information Technology pattern. This successfully identifies all employees not in the IT department.

Summary

This activity effectively utilized various SQL filtering techniques to retrieve targeted data from the log_in_attempts and employees tables.

- **Filtering for Dates and Times**: Used comparison operators (>) for time filtering (login_time > '18:00:00') and equality (=) for date filtering (login_date = '2022-05-09' OR login_date = '2022-05-08').
- **Using AND and OR**:
  - **AND** was crucial for combining multiple necessary conditions (e.g., after-hours **AND** failed login attempt; Marketing **AND** East building office).
  - **OR** was used to include data that satisfies one of several conditions (e.g., Sales **OR** Finance department).
- **Using LIKE to Search for a Pattern**: Employed the **LIKE** keyword with the **%** wildcard to perform flexible, partial pattern matching, which is essential when column values have variations or include other text. Examples include matching any office in the East building ('East%') or matching country codes for Mexico ('MEX%').
- **Using NOT in Filters**: The **NOT** operator was applied with LIKE (NOT LIKE) to **exclude** specific patterns (e.g., excluding login attempts from Mexico, or excluding employees in the IT department), which is a key technique for focusing investigations on all *other* data.

Collectively, these SQL filters allow for precise data retrieval, making security incident investigation and system update preparation efficient and accurate.