

Apply Filters to SQL Queries

Project Description:

My organization is working to make their system more secure. It is my job to ensure the system is safe, investigate all potential security issues, and update employee computers as needed. The following steps provide examples of how I used SQL with filters to perform security-related tasks.

Steps and Commands:

1. Retrieve After-Hours Failed Login Attempts:

Objective:

Investigate all failed login attempts that occurred after business hours (after 18:00).

Command:

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

 Copy code

Explanation:

This query filters for failed login attempts that occurred after 18:00. The `WHERE` clause with `AND` operator ensures only login attempts after 18:00 and those that were unsuccessful are selected.

2. Retrieve Login Attempts on Specific Dates:

Objective:

Investigate any login activity that happened on 2022-05-09 or the day before.

Command:

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

 Copy code

Explanation:

This query returns all login attempts on 2022-05-09 or 2022-05-08 using the `WHERE` clause with `OR` operator to filter for the specified dates.

3. Retrieve Login Attempts Outside of Mexico:

Objective:

Investigate login attempts that occurred outside of Mexico.

Command:

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

 Copy code

Explanation:

This query returns all login attempts that occurred in countries other than Mexico. The `NOT LIKE 'MEX%'` pattern matches entries that do not start with 'MEX', covering both 'MEX' and 'MEXICO'.

4. Retrieve Employees in Marketing:

Objective:

Update computers for employees in the Marketing department located in the East building.

Command:

```
sql
```

 Copy code

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

Explanation:

This query returns all employees in the Marketing department in the East building. The `WHERE` clause with `AND` operator ensures that only those in the Marketing department and located in the East building are selected.

5. Retrieve Employees in Finance or Sales:

Objective:

Update computers for employees in the Finance and Sales departments.

Command:

```
sql
```

 Copy code

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

Explanation:

This query returns all employees in the Finance and Sales departments using the `WHERE` clause with `OR` operator to include employees from either department.

6. Retrieve All Employees Not in IT:

Objective:

Make security updates on employees who are not in the Information Technology department.

Command:

```
sql
```

 Copy code

```
SELECT * FROM employees
WHERE department != 'Information Technology';
```

Explanation:

This query returns all employees not in the Information Technology department using the `WHERE` clause with `!=` operator to exclude IT employees.

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 utilized the 'AND', 'OR', and 'NOT' operators to filter the specific information needed for each task. Additionally, I employed 'LIKE' and the percentage sign ('%') wildcard to filter for patterns.