Apply filters to SQL queries_2

Activity Overview

In this activity, you will create a new portfolio document to demonstrate your experience using SQL. You can add this document to your cybersecurity portfolio, which you can share with prospective employers or recruiters. To review the importance of building a professional portfolio and options for creating your portfolio, read Create a cybersecurity portfolio.

To create your portfolio document, you will review a scenario and follow a series of steps. This scenario is connected to the lab you have just completed about using the AND, OR, and NOT operators in SQL to filter for information. You will explain the queries you performed in that lab, and this will help you prepare for future job interviews and other steps in the hiring process.

Be sure to complete this activity and answer the questions that follow before moving on. The next course item will provide you with a completed exemplar to compare to your own work.

Scenario

Review the scenario below. Then complete the step-by-step instructions.

You are a security professional at a large organization. Part of your job is to investigate security issues to help keep the system secure. You recently discovered some potential security issues that involve login attempts and employee machines.

Your task is to examine the organization's data in their **`employees`** and **`log_in_attempts`** tables. You'll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues.

Note: This scenario involves the same queries as the ones the Filter with AND, OR, and NOT lab. You can revisit the lab to get screenshots to include in your portfolio document. If you choose, it's also possible to complete this activity without revisiting the lab by typing your queries in the template.

Project description

In this project, the goal is to optimize data retrieval and analysis by implementing advanced filtering techniques in SQL. The project is essential for improving the efficiency of the data management system, enabling users (Cybersecurity analysts) to extract specific information from the databases with greater precision. We will design and implement a set of SQL queries that incorporate various filter clauses, such as '**WHERE**', **AND**, '**OR**', and '**NOT**', and employ operators like = (equals), <> (not equal), and '**LIKE**'. These filters will allow users to extract data based on specified conditions, such as date ranges, department, time range, and office or building location. The project will not only enhance the user experience but also significantly reduce the amount of irrelevant data retrieved, leading to improved query performance and more meaningful insights for decision-makers.

This project will require close collaboration between our database administrators, data analysts, and end-users to understand their specific needs and design SQL queries that align with their data retrieval requirements. We will also provide comprehensive training to users on how to apply filters effectively to extract the information they need. By enhancing our data retrieval capabilities through SQL filters, we aim to empower our organization to make data-driven decisions more efficiently and to gain a competitive advantage in an increasingly data-centric business environment.

Retrieve after-hours failed login attempts

To retrieve failed login attempts after 5:00 PM, we can use an SQL query that filters the login attempts based on the time of day. Here's a query example with an explanation:

Assuming we have a table named **log_in_attempts** with a **login_time** column representing the timestamp of login attempts and a **success** column indicating whether the login was successful, you can use the following SQL query:

```
Mysql>SELECT *
    ->FROM log_in_attempts
    ->WHERE login_time >= '17:00' AND success = FALSE;
```

In this query:

1. **SELECT *** is used to select all columns from the **log_in_attempts** table. We can replace with specific column names if needed.
2. **FROM log_in_attempts** specifies the table you're querying, which is **log_in_attempts** in this case.
3. **WHERE login_time >= 17** is the condition for selecting rows from the table. It uses the The condition checks if the hour is greater than or equal to 17, which represents 5:00 PM or later in a 24-hour time format.
4. **AND success = FALSE** further filters the results to only include failed login attempts (where **success** is **FALSE**).

With this query, we will retrieve all the login attempts from the **log_in_attempts** table that occurred after 5:00 PM and were unsuccessful. You can adjust the time (in this case, 17) to match your specific definition of "after 5:00 PM" if needed.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-10-09. Any login activity that happened on 2022-10-09 needs to be investigated.
we can retrieve login attempts on a specific date using the **OR** operator in your SQL query. Here's an example query with an explanation:
Assuming you have a table named **log_in_attempts** with a **login_time** column representing the timestamp of login attempts and you want to retrieve login attempts that occurred on a specific date, for instance, '2023-10-27', you can use the following SQL query:

```
Mysql>SELECT *
    ->FROM log_in_attempts
    ->WHERE lofin_date = ‘2022-10-27’;
```

In this query:

1. **SELECT *** is used to select all columns from the **log_in_attempts** table.
2. **FROM log_in_attempts** specifies the table we're querying, which is **log_in_attempts** in this case.

3. **WHERE DATE(login_date) = '2022-10-27'** is the condition for selecting rows from the table. This condition checks if the date part of the **login_date** column matches the specific date '2023-10-27'.

Retrieve login attempts outside of Canada

After Investigating the organization’s data on login attempts, I believe there is an issue with the login attempts that occurred outside of Canada. These login attempts should be investigated.

To investigate the incident I have to retrieve login attempts that are outside of a specific country, I can use a database or log analysis tool and write a query that filters the login attempts based on the user's IP address or geolocation data. Here's a simplified. So, I need to construct a SQL query using "NOT" and "LIKE" to filter login attempts outside of Canada:

```
Mysql>SELECT *
    ->FROM login_attempts
    ->WHERE user_country NOT LIKE ‘Canada%’;
```

In this query, I assume there is a database table called "login_attempts" that contains information about each login attempt, including the user's IP address and geolocation data. The query filters for entries where the "user_country" column does not start with 'Canada'. This query would return all login attempts from users with IP addresses that are not associated with Canada.

The actual implementation would depend on the structure of my database and how I store geolocation data. Additionally, for more accurate geolocation filtering, I may need to use a dedicated geolocation database or service to map IP addresses to countries.

Retrieve employees in Marketing

To retrieve employees in the marketing department located in specific buildings (West, East, North, or South), we would need to have a database or data source that contains information about employees, their departments, and their building locations. You can use SQL to query this data. Here's a generalized SQL query that you can modify based on your database structure:

Assuming there is a table named "employees" with columns "department" and "building":

```
Mysql>SELECT *

    ->FROM employees
```

```
    ->WHERE department = ‘Marketing’ AND building LIKE ‘West%’;
```

In this query, we're selecting all columns from the "employees" table where the "department" column is 'Marketing' and the "building" column matches any of the specified values (West, East, North, or South). This will return employees in the marketing department who work in any of the specified buildings. We use also AND, and LIKE clauses.

Retrieve employees in Finance

To retrieve employees in the Finance department, I can use a SQL query similar to the one I provided for the Sales department. Assuming there is a table named "employees" with a "department" column, can modify the query as follows:

```
Mysql>SELECT *

    ->FROM employees

    ->WHERE department = ‘Finance’;
```

In this query, I’m selecting all columns from the "employees" table where the "department" column is equal to 'Finance'. This will return a list of employees who belong to the Finance department.

Retrieve employees in Finance or Sales

To retrieve employees in both the Sales and Finance departments, you can use a SQL query with the `IN` clause to specify multiple departments. Here's how you can do it:

department = ‘Finance’ OR department = ‘Sales’;
We can also use OR clause for both Finance and Sales.

In this query, we are selecting all columns from the "employees" table where the "department" column is either 'Sales' or Assuming you have a table named "employees" with a "department" column:

```
Mysql>SELECT *
    ->FROM employees
    ->WHERE department IN ( ‘Sales’ , ‘Finance’ );
```

Or We can use:

```
Mysql>SELECT *
    ->FROM employees
    ->WHERE 'Finance'. This will return a list of employees who belong to either the Sales department or the Finance department.
```

Retrieve all employees working in IT

To retrieve all employees working in the Information Technology (IT) department, you can use a SQL query like this:

Assuming you have a table named "employees" with a "department" column:

```
Mysql>SELECT *
    >FROM employees
    >WHERE department = ‘Information Technology’;
```

In this query, we are selecting all columns from the "employees" table where the "department" column is equal to 'Information Technology'. This will return a list of employees who work in the IT department.

Please adjust the table and column names to match your specific database schema if they are different.

Retrieve all employees not in IT

using the **NOT** operator in a query to retrieve all employees who are not working in the Information Technology (IT) department. Here's how it works:

```
Mysql>SELECT *
    ->FROM employees
    ->WHERE department <> 'Information Technology';
```

Or

```
Mysql>SELECT *
    ->FROM employees
    ->WHERE NOT department = 'Information Technology';
```

In this query, we are selecting all columns from the "employees" table where the "department" column is not equal to 'Information Technology'. This will return a list of employees who work in departments other than IT.