# Apply filters to SQL queries

## Project description

## **Title:** Advanced SQL Query Filters for Data Analysis

## **Description:** As a security analyst, retrieving precise information from databases is crucial for making informed decisions and identifying potential vulnerabilities. In this project, I explored how to efficiently filter and analyze data using SQL's logical operators: AND, OR, and NOT. These operators allowed me to create complex conditions to extract specific data sets from a database, focusing on scenarios where multiple criteria needed to be met or excluded.

## **Key Highlights:**

## AND Operator: Used to retrieve records that match multiple conditions simultaneously.

## OR Operator: Applied to gather records that meet at least one of several conditions.

## NOT Operator: Leveraged to exclude specific records from the query results that do not match a particular condition.

## Use Cases: Filtering logs to detect anomalous patterns, identifying users who meet specific security requirements, and excluding irrelevant data to improve focus during threat assessments.

## **Skills Gained:**

## Crafting efficient SQL queries for secure data analysis.

## Practical application of filtering techniques to streamline complex data retrieval processes.

## Enhanced understanding of database operations in security-focused environments.

## This project demonstrates my ability to use SQL for refined, data-driven decision-making in security analysis.

We are running our queries in Maria DB with the DB name as organization and have the DB structure as below.

The organization database contains the following two tables:

* log\_in\_attempts
* employees

log\_in\_attempts

The log\_in\_attempts table has the following columns:

* event\_id: The identification number assigned to each login event
* username: The username of the employee
* login\_date: The date the login attempt was recorded
* login\_time: The time the login attempt was recorded
* country: The country where the login attempt occurred
* ip\_address: The IP address of that employee’s machine
* success: The success of the login attempt; FALSE indicates a failed attempt

In the MariaDB shell, these columns are returned as:



employees

The employees table has the following columns:

* employee\_id: The identification number assigned to each employee
* device\_id: The identification number assigned to each device used by the employee
* username: The username of the employee
* department: The department the employee is in
* office: The office the employee is located in

In the MariaDB shell, these columns are returned as:



## Retrieve after hours failed login attempts

In this project, we aim to retrieve data from the log\_in\_attempts table, focusing on all failed login attempts that occurred after office hours (i.e., after 18:00). To achieve this, we will extract the hour from the login\_time column and filter the results based on the login status, which is recorded in the success column.

The SQL query designed to accomplish this will target:

* Failed login attempts (where success is marked as 'false' or equivalent).
* Login attempts made after 18:00, using the login\_time column to extract the hour of each attempt.

***SELECT \* FROM log\_in\_attempts WHERE login\_time > ‘18:00’ AND success = FALSE;***

A screen shot of a computer

Description automatically generated19

After successfully running the SQL query, we were able to retrieve **19 records** that matched our criteria. This query allowed us to quickly pinpoint potential security issues by focusing on failed login attempts outside of normal office hours, which are more likely to be unauthorized access attempts.

## Retrieve login attempts on specific dates

In this next task, we aim to retrieve all login attempts made on either **May 8, 2022**, or **May 9, 2022** from the log\_in\_attempts table. By filtering the records based on these specific dates, we can analyze user activity during this timeframe and investigate any unusual login behavior.

The query will filter the login\_time column to include only records where the date is either 2022-05-08 or 2022-05-09. This approach helps us quickly isolate and examine login patterns on those two days, which could be crucial for detecting potential security breaches or anomalies.

SELECT \* FROM log\_in\_attempts WHERE login\_date = ‘2022-05-08’ OR login\_date = ‘2022-05-09’;

A screen shot of a computer

Description automatically generated

## After successfully running the SQL query, we retrieved a total of 75 records that matched the specified dates—May 8, 2022, and May 9, 2022. This result allows us to analyze all login attempts that occurred during this period, providing valuable insight into user activity and potential security events within these two days.

## This successful query execution confirms the accuracy of our filter criteria, helping us isolate relevant data for further investigation.

## Retrieve login attempts outside of Mexico

In this task, we aim to retrieve all login attempts that were made **outside of Mexico** from the log\_in\_attempts table. By filtering for login attempts originating from locations other than Mexico, we can focus on potential security threats from external regions, which may pose a higher risk. Here we are using the LIKE operator with wildcard symbol (%). Helps us to find the pattern in the entry as some of the entries are with MEX and some have full entry as MEXICO.

The query will utilize the location column to exclude any records where the country is Mexico. This allows us to narrow down login attempts from international users, which could be crucial for identifying unauthorized access attempts from regions outside our normal operating area.

***SELECT \* FROM log\_in\_attempts WHERE country LIKE ‘MEX%’;***

A screen shot of a computer screen

Description automatically generated

After successfully running the query, we identified **144 login attempts** that were made from locations **outside of Mexico**. These records are crucial for further analysis as they represent potential external access attempts, which may require closer scrutiny for security purposes.

This successful query execution allows us to focus on login attempts from foreign regions, helping us better assess and mitigate risks from unauthorized access originating outside our primary operating country.

## Retrieve employees in Marketing AND OFFICE located in East Side

In this activity, we will retrieve records of employees who are part of both the Marketing department and the Office location situated in the East Side. By applying the AND operator, we can ensure that our query only returns employees meeting both criteria simultaneously.

The query will filter the data from the employees table, targeting those whose department is Marketing and who are also located in the East Side. This approach allows us to narrow down our focus to a specific subset of employees, facilitating more targeted analysis and insights into departmental operations in that location.

SELECT \* FROM employees WHERE department = ‘Marketing’ AND office LIKE ‘East%’;

A screenshot of a computer program

Description automatically generated

## After successfully executing the query, we identified a total of 7 records of employees who are in the Marketing department and are located in the East Side office. This result provides valuable insights into the specific team composition in that region, allowing for targeted analysis and potential follow-up actions.

## The identification of these 7 employees can aid in understanding the distribution of roles within the Marketing department and help facilitate communication and collaboration strategies for this specific location.

## Retrieve employees in Finance or Sales

In this activity, we will retrieve records of employees who work in either the **Finance** department or the **Sales** department. By utilizing the **OR** operator in our query, we can ensure that our search captures all employees belonging to either of these two departments.

The query will filter data from the employees table, targeting those whose department is either **Finance** or **Sales**. This approach allows us to gather insights into employee distribution across these key areas, facilitating better resource allocation and departmental analysis.

***SELECT \* FROM employees WHERE department = ‘Finanace’ or department = ‘Sales’;***

A screenshot of a computer screen

Description automatically generated

After successfully executing the query, we found a total of **71 records** of employees who belong to either the **Finance** or **Sales** departments. This result provides a comprehensive view of the workforce in these critical areas, enabling us to analyze employee distribution and assess departmental performance effectively.

Identifying these 71 employees will help in making informed decisions regarding resource allocation, team dynamics, and strategic planning within the Finance and Sales departments.

## Retrieve all employees not in Information Technology

In this activity, we will retrieve records of all employees who are **not** working in the **Information Technology (IT)** department. By using the **NOT** operator in our query, we can effectively filter out those employees associated with the IT department.

The query will target the employees table and exclude any records where the department is labeled as **Information Technology**. This approach allows us to focus on employees from other departments, providing insights into the overall workforce composition outside of IT.

SELECT \* FROM employees WHERE NOT department = ‘Information Technology’;

A screenshot of a computer screen

Description automatically generated

After successfully executing the query, we identified a total of **161 records** of employees who are **not** working in the **Information Technology (IT)** department. This result gives us a clear view of the workforce composition outside of IT, allowing for further analysis of employee distribution across other departments.

These 161 employees represent a diverse range of roles and functions, which can be valuable for understanding departmental dynamics and making strategic decisions regarding workforce management.

## Summary

Throughout this series of activities, we explored various SQL queries to analyze and retrieve specific employee and login attempt records from the database, enhancing our understanding of user behavior and departmental structures.

1. **Failed Login Attempts After Office Hours**: We retrieved data from the log\_in\_attempts table to identify failed login attempts made after 18:00. The successful query yielded **19 records**, highlighting potential security risks outside normal working hours.
2. **Login Attempts on Specific Dates**: We focused on retrieving all login attempts made on **May 8, 2022**, and **May 9, 2022**. This query resulted in **75 records**, allowing us to analyze user activity during these key dates.
3. **Login Attempts Outside Mexico**: We aimed to find all login attempts made from locations **outside of Mexico**, leading to the identification of **144 records**. This analysis is crucial for assessing security threats from external sources.
4. **Employees in Marketing and Office on East Side**: We retrieved records of employees working in both the **Marketing** department and located in the **East Side** office, resulting in **7 records**. This outcome provided insights into team composition in that specific location.
5. **Employees in Finance or Sales**: By querying for employees in either the **Finance** or **Sales** departments, we found **71 records**. This data helps us understand workforce distribution across these essential departments.
6. **Employees Not in Information Technology**: Lastly, we retrieved all employees who are **not** working in the **Information Technology (IT)** department, which resulted in **161 records**. This query allowed us to focus on employees across various other departments.

These activities demonstrate the application of SQL filtering techniques using the **AND**, **OR**, and **NOT** operators to extract meaningful insights from the database. The successful execution of each query underscores the effectiveness of SQL in analyzing data for security and organizational purposes.