# Apply filters to SQL queries

## Project description

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

My task is to examine the organization’s data in their employees and log\_in\_attempts tables. I’ll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues.

By using SQL filters to retrieve records from the employees and log\_in\_attempts tables, I can gather information about login events, usernames, timestamps, locations, success rates, and employee details. Through SQL queries and analysis, I aim to identify any suspicious login patterns, unauthorized access attempts, or potential vulnerabilities in the system.

## Retrieve after hours failed login attempts

SELECT \*

FROM log\_in\_attempts

WHERE success = 0

AND login\_time > '18:00';

* Describe your query and how it works:

SELECT \*: This selects all columns from the log\_in\_attempts table for the matching records.

FROM log\_in\_attempts: Specifies the table from which the data is retrieved.

WHERE success = 0: This condition filters the records where the success column has a value of 0, indicating a failed login attempt.

AND login\_time > '18:00': This condition filters the records where the login\_time is greater than '18:00', meaning the login attempt occurred after 18:00.

## Retrieve login attempts on specific dates

SELECT \*

FROM log\_in\_attempts

WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08';

* Describe your query and how it works:

SELECT \*: This selects all columns from the log\_in\_attempts table for the matching records.

FROM log\_in\_attempts: Specifies the table from which the data is retrieved.

WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08': This condition filters the records where the login\_date column matches either '2022-05-09' or '2022-05-08'. The OR operator is used to specify multiple conditions.

## Retrieve login attempts outside of Mexico

SELECT \*

FROM log\_in\_attempts

WHERE country NOT LIKE '%MEX%';

* Describe your query and how it works:

SELECT \*: This selects all columns from the log\_in\_attempts table for the matching records.

FROM log\_in\_attempts: Specifies the table from which the data is retrieved.

WHERE country NOT LIKE '%MEX%': This condition filters the records where the country column does not contain the string 'MEX' (case-insensitive). The % symbol is a wildcard character that matches any sequence of characters. By using %MEX%, we ensure that values like 'MEX' and 'MEXICO' are excluded from the result.

## Retrieve employees in Marketing

SELECT \*

FROM employees

WHERE department = 'Marketing'

AND office LIKE 'East%';

* Describe your query and how it works:

SELECT \*: This selects all columns from the employees table for the matching records.

FROM employees: Specifies the table from which the data is retrieved.

WHERE department = 'Marketing': This condition filters the records where the department column has a value of 'Marketing', indicating employees in the Marketing department.

AND office LIKE 'East%': This condition filters the records where the office column starts with 'East', using the LIKE keyword and the % wildcard. This ensures that all offices in the East building are included in the results. For example, 'East-170', 'East-320', and 'East-434' would match this condition.

## Retrieve employees in Finance or Sales

SELECT \*

FROM employees

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

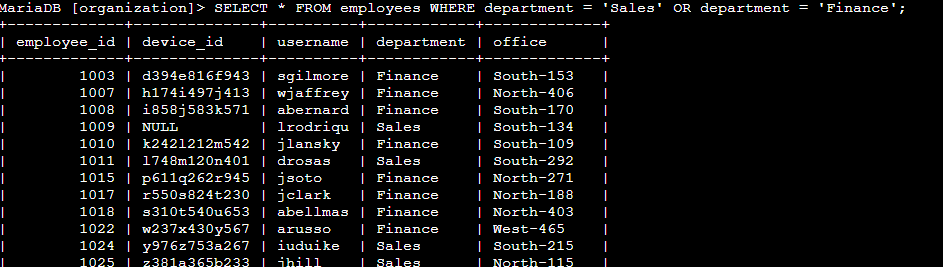
* Describe your query and how it works:

SELECT \*: This selects all columns from the employees table for the matching records.

FROM employees: Specifies the table from which the data is retrieved.

WHERE department = 'Sales' OR department = 'Finance': This condition filters the records where the department column has a value of 'Sales' or 'Finance', indicating employees in the Sales or Finance departments.

By using the OR operator between the conditions, the query retrieves all employees who belong to either the Sales or Finance department. The result will include all the columns from the employees table for those matching records.

Retrieve all employees not in IT

SELECT \*

FROM employees

WHERE department != 'Information Technology';

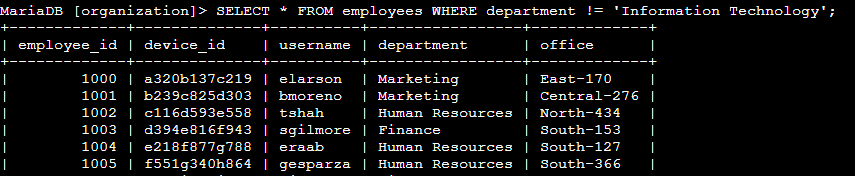
* Describe your query and how it works:

SELECT \*: This selects all columns from the employees table for the matching records.

FROM employees: Specifies the table from which the data is retrieved.

WHERE department != 'Information Technology': This condition filters the records where the department column does not equal 'Information Technology'.

By using the != operator, the query retrieves all employees whose department is not 'Information Technology'. The result will include all the columns from the employees table for those matching records.



## **Summary**

The previous tasks related to SQL filters with AND, OR, and NOT are directly applicable to this scenario. By combining different filter conditions such as usernames, login dates, countries, success status, and IP addresses, I can effectively narrow down and extract relevant data from the log\_in\_attempts table.

Furthermore, by joining the log\_in\_attempts table with the employees table using common fields like usernames or employee IDs, I can gather additional employee-specific information to aid in my investigation. These SQL techniques helped me uncover patterns, anomalies, and potential security issues within the organization's login and employee data.