# Apply filters to SQL queries

## Project description

I am a security analyst working at a larger organization, tasked with keeping the system secure. I have recently discovered some potential security issues having to do with suspicious failed login attempts, or machines held throughout the organization – various departments, various employees – that could pose a security risk.

## Retrieve after hours failed login attempts

There was a potential security accident that happened after hours (18:00). To investigate this, I ran a SQL query in the Maria DB table for records of all login attempts after 18:00. This is the query I used. A star stands for “retrieve all”, and of course all numeric values need to have (‘’) around them since in our organization we take the threat of SQL injection seriously and having the retrieved value(s) converted to strings also reduces the risk for excessive server load stemming from that.

SELECT \*

FROM log\_in\_attempts

WHERE login\_time > '18:00' AND success = ‘0’;

The reason for the last zero (0) is that login success is a Boolean value that is either TRUE (stored as 1) or FALSE (stored as 0). There were 19 such attempts.

See also: Failed attempts to log in after hours.

## Retrieve login attempts on specific dates

There was a suspicious event on 2022-05-09. To investigate this, I wanted to review all login attempts on this day, and the day before. I ran a SQL query in the Maria DB table for records of all login attempts on those two days. This is the query I used, to retrieve both using the OR keyword.

SELECT \*

FROM log\_in\_attempts

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

Note that all login attempts are of interest in this case.

See also: Retrieve login attempts on specific dates.

## Retrieve login attempts outside of Mexico

There has been much suspicious activity with login attempts, but my organization has determined that those did originate from outside of Mexico. To retrieve those login attempts that needed further investigation, I ran a query on the company Maria DB in the table log\_in\_attempts. To make sure I did not include any attempts originating from Mexico by mistake, I added a % sign since the system log the country both as MEXICO and as MEX. LIKE is a keyword that allows for searches using similar values rather than exact values, and of course NOT is a keyword that filters out all those values.

SELECT \* FROM log\_in\_attempts

WHERE NOT country LIKE 'MEX%';

See attached file: Retrieve login attempts outside of Mexico

The snapshot covers a part of the list. The full list shows 144 such attempts.

## Retrieve employees in Marketing

There were security updates needed for some, but not all, machines in the Marketing department. Those needing attention were all situated somewhere in the East building. To find them I ran a query in the employee table using this query. The dollar sign is to ensure that no affected machine in East building with all its offices gets missed, and no other marketing personnel gets needly affected. AND is the keyword that ensure that both conditions must be met to retrieve that value.

SELECT \*

FROM employees

WHERE department = 'Marketing' AND office LIKE 'East%';

See also attached file: Retrieve employee in Marketing

## Retrieve employees in Finance or Sales

Our team needs to perform an additional security update for all employees in the Sales and in the Finance departments. To find the employees in those departments I ran a Maria DB SQL query in the employee table and filtered it for employees in the Finance and Sales departments, respectively. OR is a keyword that ensures that if any condition is met then that value should be retrieved.

SELECT \*

FROM employees

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

See also attached file: Retrieve employees in Finance or Sales

The snapshot covers a part of the list. The full list is of 71 employees.

## Retrieve all employees not in IT

There was a need to perform a security update on all computer machines. Every employee in Information Technology department had already received that update, so I ran an SQL query to find all other employees in the organization so that those machines too could be updated. This was the query in the used in the employee data table, using the NOT keyword to filter out all employees in the IT department.

SELECT \*

FROM employees

WHERE NOT department = 'Information Technology';

See also attached file: Retrieve all employee not in IT.

The snapshot covers a part of the list. The full list is of 161 employees.

## Summary

I applied filters to my SQL queries, AND, OR and NOT to get more precise answers. I also used LIKE and the % sign to search for patterns in my searches.