

# 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.

## Retrieve after hours failed login attempts

My team is investigating failed login attempts that were made after business hours. I want to retrieve this information from the login activity. I can identify all unsuccessful attempts after 18:00 with this SQL statement:

```
MariaDB [organization]> SELECT *  
-> FROM log_in_attempts  
-> WHERE login_time > '18:00:00' AND success = FALSE  
-> ;
```

## Retrieve login attempts on specific dates

My team is investigating a suspicious event that occurred on '2022-05-09'. I want to retrieve all login attempts that occurred on this day and the day before ('2022-05-08'). I did so using this SQL statement:

```
MariaDB [organization]> SELECT *  
-> FROM log_in_attempts  
-> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

## Retrieve login attempts outside of Mexico

Now, my team is investigating logins that did not originate in Mexico, and I need to find this information. I note that the country field includes entries with 'MEX' and 'MEXICO'. I should use the **NOT** and **LIKE** operators and the matching pattern 'MEX%'.

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

## Retrieve employees in Marketing

My team is updating employee machines, and I need to obtain the information about employees in the 'Marketing' department who are located in all offices in the East building (such as 'East-170' or 'East-320'). I did so using this SQL statement:

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

## Retrieve employees in Finance or Sales

Now, my team needs to perform a different update to the computers of all employees in the Finance or the Sales department, and I need to locate information on these employees.

```
MariaDB [organization]> SELECT * FROM employees  
-> WHERE department = 'Sales' OR department = 'Finance';
```

## Retrieve all employees not in IT

My team needs to make one more update. This update was already made to employee computers in the Information Technology department. The team needs information about employees who are not in that department. I should use the **NOT** operator to identify these employees.

```
MariaDB [organization]> SELECT * FROM employees  
-> WHERE NOT department = 'Information Technology';
```

## Summary

I looked up login attempts after hours, those on two specific days, and login attempts outside of Mexico. Then I looked up employees from different departments that needed updates to their computers. I used the **WHERE** command with various filters, such as **AND**, **OR** and **NOT**.