

# Aplicación de filtros a consultas SQL (Applying filters to SQL queries)

## Descripción del proyecto

Mi organización está trabajando para que su sistema sea más seguro. Mi labor consiste en garantizar que el sistema esté protegido, investigar todos los posibles problemas de seguridad y actualizar las computadoras de los/las empleados/as según sea necesario. Los pasos siguientes ofrecen ejemplos de cómo usé SQL con filtros para realizar tareas de seguridad.

## Recupera intentos de inicio de sesión fallidos después del horario laboral

Se produjo un posible incidente de seguridad después del horario laboral (después de las 18:00). Fue necesario investigar todos los intentos de inicio de sesión fallidos después del horario laboral.

El código siguiente demuestra cómo creé una consulta SQL para filtrar por los intentos de inicio de sesión fallidos que tuvieron lugar después del horario laboral.

## Project Description

My organization is working to make its system more secure. My job is to ensure the system is protected, investigate all potential security issues, and update employee computers as needed. The following steps provide examples of how I used SQL with filters to perform security tasks.

Retrieve failed login attempts after business hours

A potential security incident occurred after business hours (after 6:00 PM). All failed login attempts after business hours needed to be investigated.

The following code demonstrates how I created a SQL query to filter for failed login attempts that occurred after business hours.

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

event_id	username	login_date	login_time	country	ip_address	success
2	apatel	2022-05-10	20:27:27	CAN	192.168.205.12	0
18	pwashing	2022-05-11	19:28:50	US	192.168.66.142	0
20	tshah	2022-05-12	18:56:36	MEXICO	192.168.109.50	0

La primera parte de la captura de pantalla es mi consulta, y la segunda es un fragmento del resultado. Esta consulta filtra por inicios de sesión fallidos que se produjeron después de las 18:00. En primer lugar, comencé por seleccionar todos los datos de la tabla `log_in_attempts` (intentos de inicio de sesión). A continuación, usé una cláusula `WHERE` con un operador `AND` para filtrar mis resultados, de manera de obtener solo los intentos de inicio de sesión fallidos que tuvieron lugar después de las 18:00. La primera condición es `login_time > '18:00'`, que filtra por los intentos de inicio de sesión que se produjeron después de las 18:00. La segunda condición es `success = FALSE`, que filtra por los intentos de inicio de sesión fallidos.

The first part of the screenshot is my query, and the second is a snippet of the result. This query filters for failed logins that occurred after 6:00 PM. First, I started by selecting all the data from the `log_in_attempts` table (login attempts). Next, I used a `WHERE` clause with an `AND` operator to filter my results, returning only failed login attempts that occurred after 6:00 PM. The first condition is `login_time > '18:00'`, which filters for login attempts that occurred after 6:00 PM. The second condition is `success = FALSE`, which filters for failed login attempts.

## Recupera intentos de inicio de sesión en fechas específicas

El 09-05-2022 se produjo un evento sospechoso. Es necesario investigar toda la actividad registrada el 09-05-2022 o el día anterior.

El código a continuación demuestra cómo creé una consulta SQL para filtrar por intentos de inicio de sesión que tuvieron lugar en fechas específicas.

## Retrieve login attempts on specific dates

A suspicious event occurred on September 5, 2022. All activity recorded on or before September 5, 2022 needs to be investigated.

The code below demonstrates how I created a SQL query to filter for login attempts that occurred on specific dates.

```

MariaDB [organization]> SELECT *
-> FROM log_in_attempts
-> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+-----+-----+-----+-----+-----+-----+-----+
| event_id | username | login_date | login_time | country | ip_address | success |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | jrafael | 2022-05-09 | 04:56:27 | CAN | 192.168.243.140 | 0 |
| 3 | dkot | 2022-05-09 | 06:47:41 | USA | 192.168.151.162 | 0 |
| 4 | dkot | 2022-05-08 | 02:00:39 | USA | 192.168.178.71 | 0 |

```

La primera parte de la captura de pantalla es mi consulta, y la segunda es un fragmento del resultado. Esa consulta devuelve todos los intentos de inicio de sesión que tuvieron lugar el 09-05-2022 o el 08-05-2022. En primer lugar, comencé por seleccionar todos los datos de la tabla `log_in_attempts` (intentos de inicio de sesión). A continuación, usé una cláusula `WHERE` con un operador `OR` para filtrar mis resultados, con el fin de obtener solo los intentos de inicio de sesión que tuvieron lugar el 09-05-2022 o el 08-05-2022. La primera condición es `login_date = '2022-05-09'`, que filtra por los inicios de sesión ocurridos el 09-05-2022. La segunda condición es `login_date = '2022-05-08'`, que filtra por los inicios de sesión ocurridos el 08-05-2022.

The first part of the screenshot is my query, and the second is a snippet of the result. That query returns all login attempts that occurred on either 2022-05-09 or 2022-05-08. First, I started by selecting all the data from the `log_in_attempts` table (login attempts). Next, I used a `WHERE` clause with an `OR` operator to filter my results, returning only login attempts that occurred on either 2022-05-09 or 2022-05-08. The first condition is `login_date = '2022-05-09'`, which filters for logins that occurred on 2022-05-09. The second condition is `login_date = '2022-05-08'`, which filters for logins that occurred on 2022-05-08.

## Recupera intentos de inicio de sesión fuera de México

Luego de haber investigado los datos de intentos de inicio de sesión en la organización, sospecho que existe un problema con los intentos de inicio de sesión realizados fuera de México. Estos intentos de inicio de sesión deben ser investigados..

El código siguiente demuestra cómo creé una consulta SQL para filtrar por intentos de inicio de sesión ocurridos fuera de México.

## Retrieve login attempts outside of Mexico

After investigating the organization's login attempt data, I suspect there is a problem with login attempts made outside of Mexico. These login attempts need to be investigated.

The following code demonstrates how I created a SQL query to filter for login attempts that occurred outside of Mexico.

```
MariaDB [organization]> SELECT *  
-> FROM log_in_attempts  
-> WHERE NOT country LIKE 'MEX%';
```

event_id	username	login_date	login_time	country	ip_address	success
1	jrafael	2022-05-09	04:56:27	CAN	192.168.243.140	0
2	apatel	2022-05-10	20:27:27	CAN	192.168.205.12	0
3	dkot	2022-05-09	06:47:41	USA	192.168.151.162	0

La primera parte de la captura de pantalla es mi consulta, y la segunda es un fragmento del resultado. Esta consulta devuelve todos los intentos de inicio de sesión que tuvieron lugar fuera de México. En primer lugar, comencé por seleccionar todos los datos de la tabla `log_in_attempts` (intentos de inicio de sesión). A continuación, usé una cláusula `WHERE` con `NOT` para filtrar por países que no son México. Usé `LIKE` con `MEX%` como el patrón de coincidencia, porque el conjunto de datos (dataset) representa a México como `MEX` y `MEXICO`. El signo de porcentaje (%) representa cualquier número de caracteres no especificados cuando se usan con `LIKE`.

The first part of the screenshot is my query, and the second is a snippet of the result. This query returns all login attempts that occurred outside of Mexico. First, I started by selecting all the data from the `log_in_attempts` table (login attempts). Next, I used a `WHERE` clause with `NOT` to filter for countries other than Mexico. I used `LIKE` with `MEX%` as the matching pattern because the dataset represents Mexico as both `MEX` and `MEXICO`. The percent sign (%) represents any number of unspecified characters when used with `LIKE`.

## Recupera empleados/as en Marketing

Mi equipo quiere actualizar las computadoras para ciertos/as empleados/as del departamento de Marketing. Para hacerlo, necesito obtener información sobre los equipos de los/las empleados/as que debo actualizar.

El código siguiente demuestra cómo creé una consulta SQL para filtrar por equipo de empleados/as en el departamento de Marketing en el edificio Este (East).

## Retrieve employees in Marketing

My team wants to update computers for certain employees in the Marketing department. To do this, I need to get information about the computers of the employees I need to update.

The following code demonstrates how I created a SQL query to filter by team of employees in the Marketing department in the East building.

```
MariaDB [organization]> SELECT *  
  -> FROM employees  
  -> WHERE department = 'Marketing' AND office LIKE 'East%';
```

employee_id	device_id	username	department	office
1000	a320b137c219	elarson	Marketing	East-170
1052	a192b174c940	jdarosa	Marketing	East-195
1075	x573y883z772	fbautist	Marketing	East-267

La primera parte de la captura de pantalla es mi consulta, y la segunda es un fragmento del resultado. Esta consulta devuelve a todos/as los/las empleados/as del departamento de Marketing en el edificio Este. En primer lugar, comencé por seleccionar todos los datos de la tabla `employees` (empleados/as). A continuación usé una cláusula `WHERE` con `AND` para filtrar por empleados/as que trabajan en el departamento de Marketing en el edificio Este (East). Usé `LIKE` con `East%` como el patrón de coincidencia porque los datos en la columna `office` (oficina) representan el edificio Este (East) con el número específico de la oficina. La primera condición es el fragmento `department = 'Marketing'`, que filtra por empleados en el departamento de Marketing. La segunda condición es el fragmento `office LIKE 'East%'`, que filtra por empleados en el edificio Este (East).

The first part of the screenshot is my query, and the second is a snippet of the result. This query returns all employees in the Marketing department in the East building. First, I started by selecting all the data from the `employees` table. Next, I used a `WHERE` clause with `AND` to filter for employees who work in the Marketing department in the East building. I used `LIKE` with `East%` as the matching pattern because the data in the `office` column represents the East building with the specific office number. The first condition is the snippet `department = 'Marketing'`, which filters for employees in the Marketing department. The second condition is the snippet `office LIKE 'East%'`, which filters for employees in the East building.

## Recupera empleados/as en Finanzas o Ventas

También es necesario actualizar los equipos de los/las empleados/as de los departamentos de Finanzas y Ventas. Como se necesita una actualización de seguridad distinta, solo debo obtener información de empleados/as de esos dos departamentos.

El código siguiente demuestra cómo creé una consulta SQL para filtrar por equipos de empleados de los departamentos de Finanzas o Ventas.

## Retrieve employees in Finance or Sales

It's also necessary to update the teams of employees in the Finance and Sales departments. Since a separate security update is required, I only need to retrieve information for employees in those two departments.

The following code demonstrates how I created a SQL query to filter by teams of employees in the Finance or Sales departments.

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

employee_id	device_id	username	department	office
1003	d394e816f943	sgilmore	Finance	South-153
1007	h174i497j413	wjaffrey	Finance	North-406
1008	i858j583k571	abernard	Finance	South-170

La primera parte de la captura de pantalla es mi consulta, y la segunda es un fragmento del resultado. Esta consulta devuelve todos/as los/as empleados/as de los departamentos de Finanzas y Ventas. En primer lugar, comencé por seleccionar todos los datos de la tabla `employees` (empleados/as). A continuación, usé una cláusula `WHERE` con `OR` para filtrar por empleados/as que trabajan en los departamentos de Finanzas y Ventas. Usé el operador `OR` en lugar de `AND` porque quería obtener todos/as los/las empleados/as de ambos departamentos. La primera condición es `department = 'Finance'`, que filtra por empleados/as del departamento de Finanzas. La segunda condición es `department = 'Sales'`, que filtra por empleados/as del departamento de Ventas.

The first part of the screenshot is my query, and the second is a snippet of the result. This query returns all employees in the Finance and Sales departments. First, I started by selecting all the data from the `employees` table. Next, I used a `WHERE` clause with `OR` to filter for employees who work in both the Finance and Sales departments. I used the `OR` operator instead of `AND`

because I wanted to get all employees from both departments. The first condition is department = 'Finance', which filters for employees in the Finance department. The second condition is department = 'Sales', which filters for employees in the Sales department.

## Recupera a todos/as los/las empleados/as que no trabajan en TI

Mi equipo necesita realizar otra actualización de seguridad para empleados/as que no trabajan en el departamento de Tecnología de la Información. Para realizar la actualización, primero debo obtener información sobre estos/as empleados/as.

A continuación, demuestro cómo creé una consulta SQL para filtrar por equipos de empleados/as que no trabajan en el departamento de Tecnología de la Información.

## Retrieve all non-IT employees

My team needs to perform another security update for employees who don't work in the Information Technology department. To perform the update, I first need to gather information about these employees.

Below, I demonstrate how I created a SQL query to filter by teams of employees who don't work in the Information Technology department.

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

employee_id	device_id	username	department	office
1000	a320b137c219	elarson	Marketing	East-170
1001	b239c825d303	bmoreno	Marketing	Central-276
1002	c116d593e558	tshah	Human Resources	North-434

La primera parte de la captura de pantalla es mi consulta, y la segunda es un fragmento del resultado. La consulta devuelve todos/as los/las empleados/as que no trabajan en el departamento de Tecnología de la Información. En primer lugar, comencé por seleccionar todos los datos de la tabla `employees` (empleados/as). A continuación, usé una cláusula `WHERE` con `NOT` para filtrar por empleados/as que no trabajan en este departamento.

The first part of the screenshot is my query, and the second is a snippet of the result. The query returns all employees who do not work in the Information Technology department. First, I started

by selecting all the data from the employees table. Next, I used a WHERE clause with NOT to filter for employees who do not work in this department.

## Resumen

Apliqué filtros a consultas SQL para obtener información específica sobre los intentos de inicio de sesión y los equipos de los/las empleados/as. Utilicé dos tablas distintas, `log_in_attempts` (intentos de inicio de sesión) y `employees` (empleados/as). Usé los operadores AND, OR y NOT para filtrar por la información específica que necesitaba para cada tarea. También utilicé LIKE y el comodín de signo de porcentaje (%) para filtrar por patrones.

## Summary

I applied filters to SQL queries to obtain specific information about employee login attempts and computers. I used two different tables, `log_in_attempts` (login attempts) and `employees` (employees). I used the AND, OR, and NOT operators to filter for the specific information I needed for each task. I also used LIKE and the percentage sign (%) wildcard to filter by patterns.