

# Aplicación de filtros a consultas SQL

## Descripción del proyecto

Desde hace tiempo la empresa donde trabajo está implementando nuevos sistemas de seguridad más intensos. Las tareas que realizo es asegurar la protección del sistema, investigar cualquier posible problema de seguridad y actualizar los sistemas de los empleados según sea necesario. A continuación, se presentan ejemplos de cómo he utilizado SQL con filtros para llevar a cabo 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.

```
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 muestra mi consulta, mientras que la segunda parte es un fragmento del resultado. Esta consulta se utiliza para filtrar los intentos de inicio de sesión fallidos que ocurrieron después de las 18:00.

Primero, seleccioné todos los datos de la tabla `log_in_attempts` (intentos de inicio de sesión). Luego, apliqué una cláusula `WHERE` con un operador `AND` para refinar mis resultados, de modo que solo se incluyeran los intentos de inicio de sesión fallidos que ocurrieron después de las 18:00.

La primera condición es `login_time > '18:00'`, que filtra los intentos de inicio de sesión que tuvieron lugar después de las 18:00. La segunda condición es `success = FALSE`, que filtra los intentos de inicio de sesión que no tuvieron éxito.

## 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.

```
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 muestra mi consulta, y la segunda parte presenta un fragmento del resultado. Esta consulta devuelve todos los intentos de inicio de sesión que ocurrieron el 09-05-2022 o el 08-05-2022.

Primero, seleccioné todos los datos de la tabla `log_in_attempts` (intentos de inicio de sesión). Luego, utilicé una cláusula `WHERE` con un operador `OR` para filtrar los resultados, obteniendo solo los intentos de inicio de sesión que ocurrieron el 09-05-2022 o el 08-05-2022.

La primera condición es `login_date = '2022-05-09'`, que filtra los inicios de sesión que ocurrieron el 09-05-2022. La segunda condición es `login_date = '2022-05-08'`, que filtra los inicios de sesión que ocurrieron el 08-05-2022.

```
sql
Copiar código
SELECT *
FROM log_in_attempts
WHERE login_date = '2022-05-09'
OR login_date = '2022-05-08';
```

Este código demuestra cómo construí una consulta SQL para filtrar los intentos de inicio de sesión que ocurrieron específicamente en las fechas mencionadas.

## Recupera intentos de inicio de sesión fuera de México

Después de revisar los datos de los intentos de inicio de sesión en nuestra organización, sospecho que hay un problema con los intentos realizados desde fuera de México. Estos requieren una investigación adicional.

El siguiente código muestra cómo formulé una consulta SQL para identificar específicamente los intentos de inicio de sesión que se originaron fuera de México:

```
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 muestra mi consulta, y la segunda parte presenta un fragmento del resultado. Esta consulta devuelve todos los intentos de inicio de sesión que se produjeron fuera de México.

Primero, seleccioné todos los datos de la tabla `log_in_attempts` (intentos de inicio de sesión). Luego, utilicé una cláusula `WHERE` con `NOT` para filtrar los registros de países que no sean México. Empleé `LIKE` con el patrón `MEX%`, ya que el conjunto de datos (dataset) representa a México como `MEX` y `MEXICO`. El signo de porcentaje (%) actúa como un comodín que representa cualquier número de caracteres cuando se usa con `LIKE`.

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

Este código muestra cómo creé una consulta SQL para filtrar los intentos de inicio de sesión que ocurrieron fuera de México.

## Recupera empleados/as en Marketing

El equipo con el que trabajo desea actualizar los sistemas de ciertos empleados del departamento de Marketing. Para hacerlo, necesito obtener información sobre los sistemas de los empleados que requieren la actualización.

El siguiente código muestra cómo formulé una consulta SQL para obtener información sobre los sistemas de los empleados del departamento de Marketing ubicados en el edificio Este (East):

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

Esta consulta selecciona todos los datos de la tabla `employees` para los empleados que pertenecen al departamento de Marketing y que trabajan en una oficina que comienza con 'East'.

```
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 muestra mi consulta, y la segunda parte presenta un fragmento del resultado. Esta consulta devuelve a todos los empleados del departamento de Marketing que trabajan en el edificio Este.

Primero, seleccioné todos los datos de la tabla `employees` (empleados). Luego, utilicé una cláusula `WHERE` con `AND` para filtrar los empleados que trabajan en el departamento de Marketing en el edificio Este (East). Utilicé `LIKE` con el patrón `East%` porque los datos en la columna `office` (oficina) representan el edificio Este junto con el número específico de la oficina.

La primera condición es `department = 'Marketing'`, que filtra a los empleados que están en el departamento de Marketing. La segunda condición es `office LIKE 'East%'`, que filtra a los empleados que se encuentran en el edificio Este.

```

sql
Copiar código
SELECT *
FROM employees
WHERE department = 'Marketing'
AND office LIKE 'East%';

```

Este código muestra cómo creé una consulta SQL para filtrar a los empleados del departamento de Marketing que trabajan en el edificio Este.

## Recupera empleados/as en Finanzas o Ventas

También es necesario actualizar los sistemas de los empleados de los departamentos de Finanzas y Ventas. Como esta actualización de seguridad es distinta, solo debo obtener información de los empleados de estos dos departamentos.

El siguiente código muestra cómo formulé una consulta SQL para obtener información sobre los sistemas de los empleados de los departamentos de Finanzas o Ventas:

```

sql
Copiar código
SELECT *
FROM employees
WHERE department = 'Finances'
OR department = 'Sales';

```

Esta consulta selecciona todos los datos de la tabla `employees` para los empleados que pertenecen al departamento de Finanzas o al departamento de Ventas.

```

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 muestra mi consulta, y la segunda parte presenta un fragmento del resultado. Esta consulta devuelve todos los empleados de los departamentos de Finanzas y Ventas.

Primero, seleccioné todos los datos de la tabla `employees` (empleados). Luego, utilicé una cláusula `WHERE` con `OR` para filtrar los empleados que trabajan en los departamentos de Finanzas

y Ventas. Utilicé el operador `OR` en lugar de `AND` porque quería obtener todos los empleados de ambos departamentos.

La primera condición es `department = 'Finance'`, que filtra a los empleados del departamento de Finanzas. La segunda condición es `department = 'Sales'`, que filtra a los empleados del departamento de Ventas.

```
sql
Copiar código
SELECT *
FROM employees
WHERE department = 'Finance'
OR department = 'Sales';
```

Este código muestra cómo creé una consulta SQL para filtrar a los empleados de los departamentos de Finanzas o Ventas.

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

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

A continuación, muestro cómo creé una consulta SQL para filtrar por equipos de empleados que no pertenecen al departamento de Tecnología de la Información:

```
sql
Copiar código
SELECT *
FROM employees
WHERE NOT department = 'Information Technology';
```

```
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 muestra mi consulta, y la segunda parte presenta un fragmento del resultado. Esta consulta devuelve todos los empleados que no trabajan en el departamento de Tecnología de la Información.

Primero, seleccioné todos los datos de la tabla `employees` (empleados). Luego, utilicé una cláusula `WHERE` con `NOT` para filtrar a los empleados que no pertenecen a este departamento.

```
sql
Copiar código
SELECT *
FROM employees
WHERE department != 'Information Technology';
```

Este código muestra cómo creé una consulta SQL para obtener información sobre los empleados que no trabajan en el departamento de Tecnología de la Información.

## Resumen

He aplicado filtros a consultas SQL para obtener información específica sobre los intentos de inicio de sesión y los equipos de los empleados. Utilicé dos tablas distintas: `log_in_attempts` (intentos de inicio de sesión) y `employees` (empleados). Apliqué los operadores `AND`, `OR`, y `NOT` para filtrar la información específica requerida para cada tarea. También usé `LIKE` junto con el comodín de porcentaje (%) para filtrar por patrones.

Aquí hay un resumen de cómo se aplicaron los filtros:

- Para filtrar intentos de inicio de sesión fallidos después de las 18:00 horas:

```
sql
Copiar código
SELECT *
FROM log_in_attempts
WHERE TIME(login_date) > '18:00:00'
AND login_status = 'failed';
```

- Para filtrar intentos de inicio de sesión que ocurrieron en fechas específicas:

```
sql
Copiar código
SELECT *
FROM log_in_attempts
WHERE login_date = '2022-05-09'
OR login_date = '2022-05-08';
```

- Para filtrar intentos de inicio de sesión fuera de México:

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

- Para obtener información sobre los equipos de empleados en el departamento de Marketing en el edificio Este:

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

- Para obtener información sobre los equipos de empleados en los departamentos de Finanzas y Ventas:

```
sql
Copiar código
SELECT *
FROM employees
WHERE department = 'Finance'
OR department = 'Sales';
```

- Para obtener información sobre los empleados que no pertenecen al departamento de Tecnología de la Información:

```
sql
Copiar código
SELECT *
FROM employees
WHERE department != 'Information Technology';
```

Estos ejemplos demuestran cómo utilicé diferentes operadores y patrones en SQL para filtrar y obtener la información necesaria de manera eficiente.