# Apply filters to SQL queries

## Project description

Nuestro trabajo ahora será **investigar estos incidentes**. Para hacerlo, tendremos que **revisar la información** que la empresa guarda en dos "archivos": uno con los **datos de los empleados** (la tabla employees) y otro con los **registros de los intentos de inicio de sesión** (la tabla log\_in\_attempts).

Nuestra herramienta principal para esta investigación será el **SQL**, un lenguaje que te permite "preguntar" a las bases de datos. Usaremos **filtros de SQL** para encontrar los datos exactos que necesitamos y así **descubrir qué está pasando** con esos problemas de seguridad.

En resumen, seremos los encargados de **escanear los registros de la empresa** para **detectar y resolver problemas de seguridad** relacionados con accesos y equipos, utilizando las habilidades en SQL.

## Retrieve after hours failed login attempts

A continuación, realizaremos un análisis para comprobar cuáles han sido los intentos de inicio de sesión fallidos por los usuarios que han intentado acceder después de las “18:00” hora final de la jornada.

Usaremos consultas SQL como las que tenemos a continuación para filtrar los accesos erróneos después del horario laboral.

Este será el comando para revisar la información especificada:

SELECT \* FROM log\_in\_attempts WHERE login\_time > '18:00' AND success = '0';

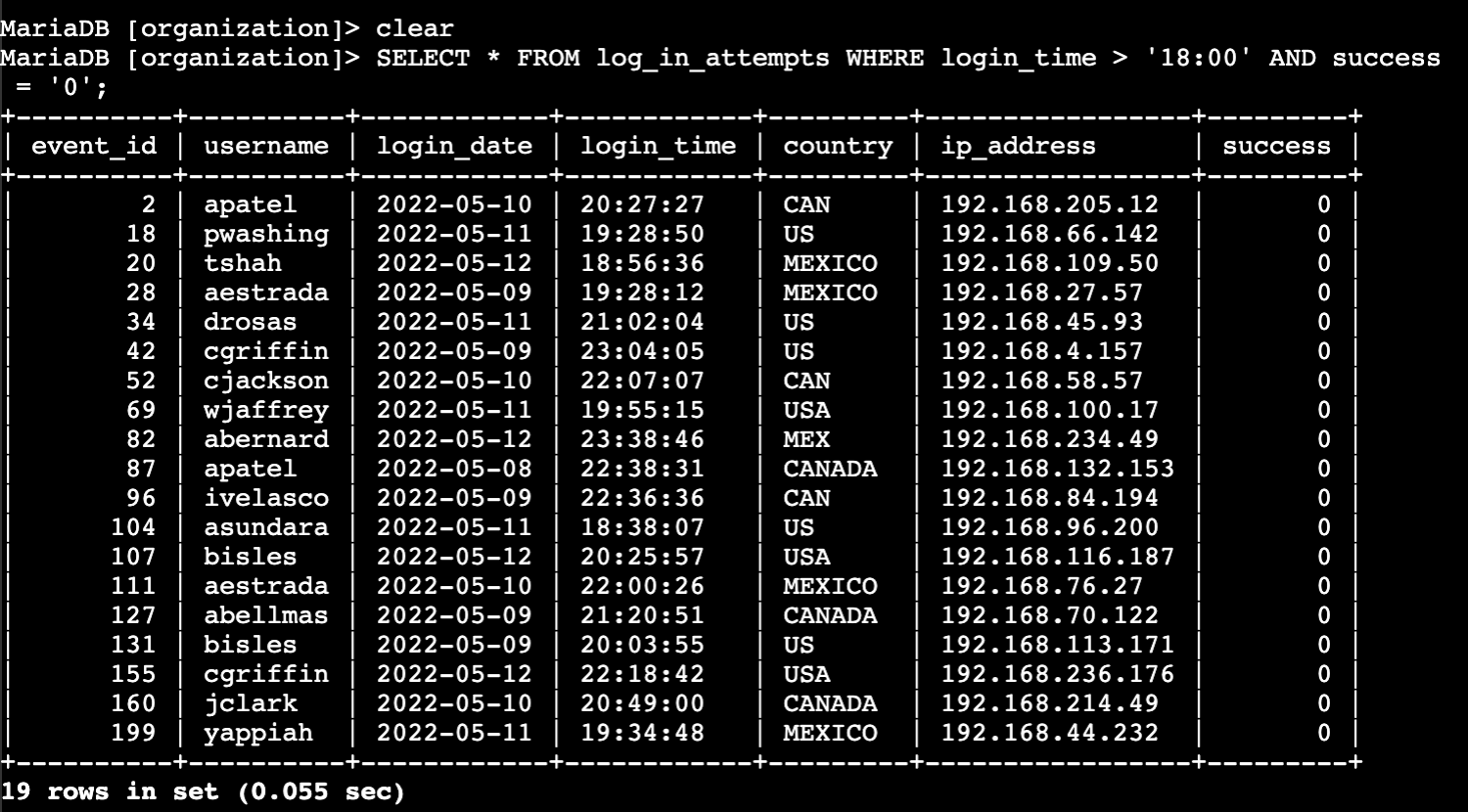
**SELECT \***:

* SELECT: Es la palabra clave que indica que quieres **recuperar o extraer datos** de una base de datos.
* \*: El asterisco es un comodín que significa "todas las columnas". Es decir, quieres **ver toda la información disponible** para cada registro que cumpla las condiciones. Si en lugar de \* pusieras user\_id, login\_time, solo verías esas dos columnas.

**FROM log\_in\_attempts**:

* FROM: Es la palabra clave que especifica de **qué tabla quieres obtener los datos**.
* log\_in\_attempts: Es el **nombre de la tabla** de donde se van a recuperar los registros. Esta tabla contiene la información sobre cada intento de inicio de sesión.

**WHERE login\_time > '18:00' AND success = '0'**:

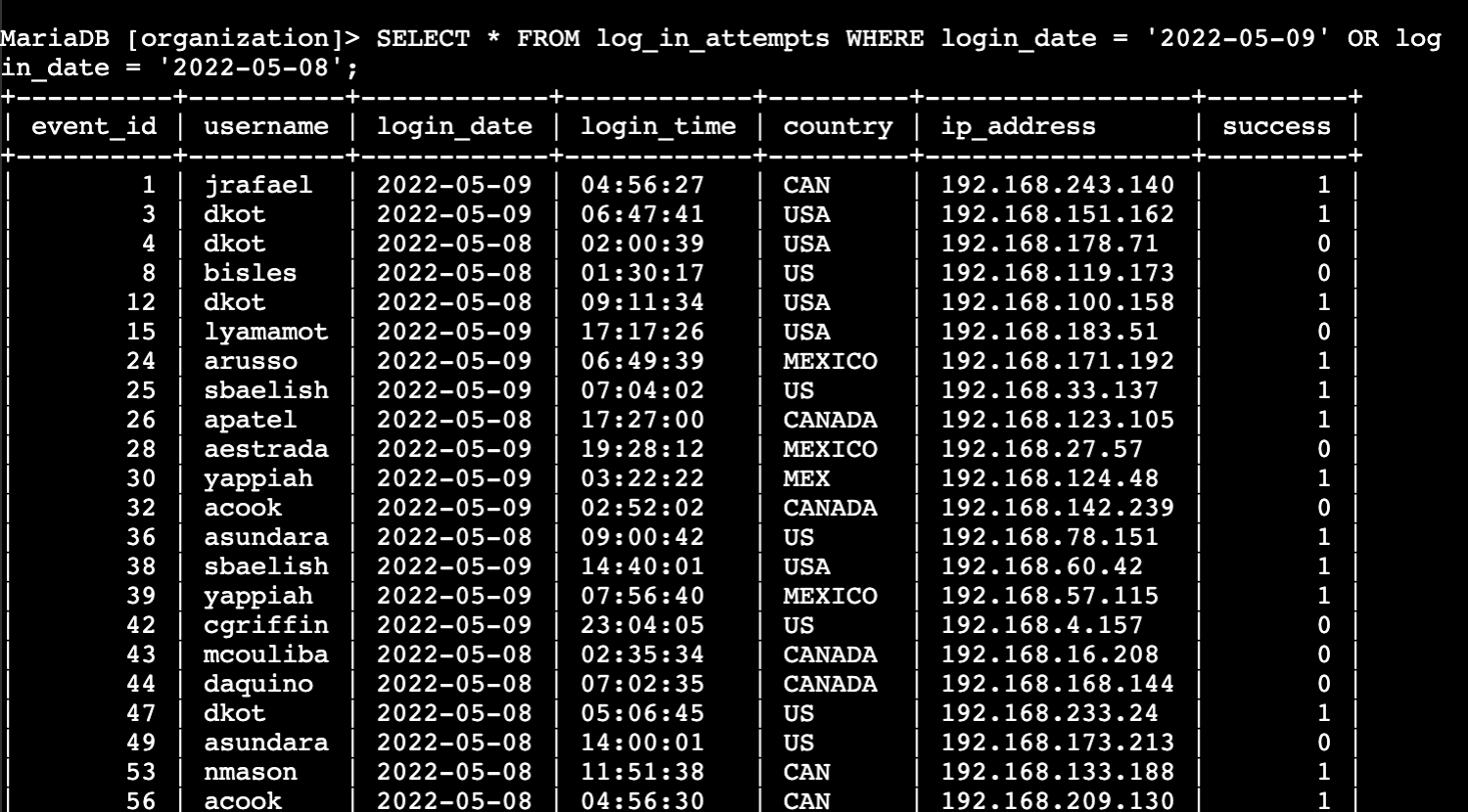
* WHERE: Es la cláusula más importante para filtrar. Indica que solo quieres los registros que **cumplan una o varias condiciones específicas**. Actúa como un filtro.
* login\_time > '18:00': Esta es la **primera condición**.
  + login\_time: Es el **nombre de una columna** en la tabla log\_in\_attempts que almacena la hora del intento de inicio de sesión.
  + >: Es el **operador de comparación "mayor que"**.
  + '18:00': Es el **valor con el que se compara** la login\_time. En este caso, se refiere a las 6 de la tarde (18:00 en formato de 24 horas). Entonces, esta parte selecciona solo los intentos de inicio de sesión que ocurrieron *después* de las 6 PM.
* AND: Este es un **operador lógico** que conecta las dos condiciones. Significa que **ambas condiciones deben ser verdaderas** para que un registro sea incluido en el resultado. Si usamos OR, solo una de las condiciones tendría que ser verdadera.
* success = '0': Esta es la **segunda condición**.
  + success: Es el **nombre de una columna** en la tabla log\_in\_attempts que probablemente indica si el intento de inicio de sesión fue exitoso o no (comúnmente, '1' para éxito y '0' para fallo).
  + =: Es el **operador de comparación "igual a"**.
  + '0': Es el **valor con el que se compara** la columna success. En este contexto, significa que el intento de inicio de sesión **no fue exitoso** (fue un fallo).

## 

## Retrieve login attempts on specific dates

Filtraremos dos fechas en concreto para comprobar qué intentos de inicio de sesión se han realizado en ese tiempo.

SELECT \* FROM log\_in\_attempts WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08';



**FROM log\_in\_attempts**: Indica que está buscando la información en la tabla llamada log\_in\_attempts.

**WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08'**: Esta es la condición de filtrado.

* login\_date: Es la columna que contiene la fecha del intento de inicio de sesión.
* = '2022-05-09': La primera parte de la condición que busca registros donde la fecha sea **exactamente el 9 de mayo de 2022**.
* **OR**: Este es el operador clave aquí. Significa que un registro será incluido en el resultado si cumple **al menos una de las dos condiciones**. No tienen que cumplirse ambas.
* login\_date = '2022-05-08': La segunda parte de la condición que busca registros donde la fecha sea **exactamente el 8 de mayo de 2022**.

## Retrieve login attempts outside of Mexico

Ahora filtraremos los contenidos que han sido filtrados fuera del país de México:

Usaremos el siguiente comando para filtrar en la base de datos:

SELECT \* FROM log\_in\_attempts WHERE NOT country LIKE 'MEX%';



## 

## 

## **FROM log\_in\_attempts**: Indica que estamos extrayendo datos de la tabla llamada log\_in\_attempts.

## **WHERE NOT country LIKE 'MEX%'**: Esta es la cláusula de filtrado clave:

## **NOT**: Este operador es un negador. Invierte el resultado de la condición que le sigue. Si la condición es verdadera, NOT la hace falsa, y viceversa.

## **country**: Es el nombre de la columna en la tabla log\_in\_attempts que almacena el país de origen del intento de inicio de sesión.

## **LIKE**: Este operador se usa para buscar patrones de texto. No busca una coincidencia exacta, sino si el texto cumple un patrón.

## **'MEX%'**: Este es el patrón que se busca.

## 'MEX': Son los primeros tres caracteres del país.

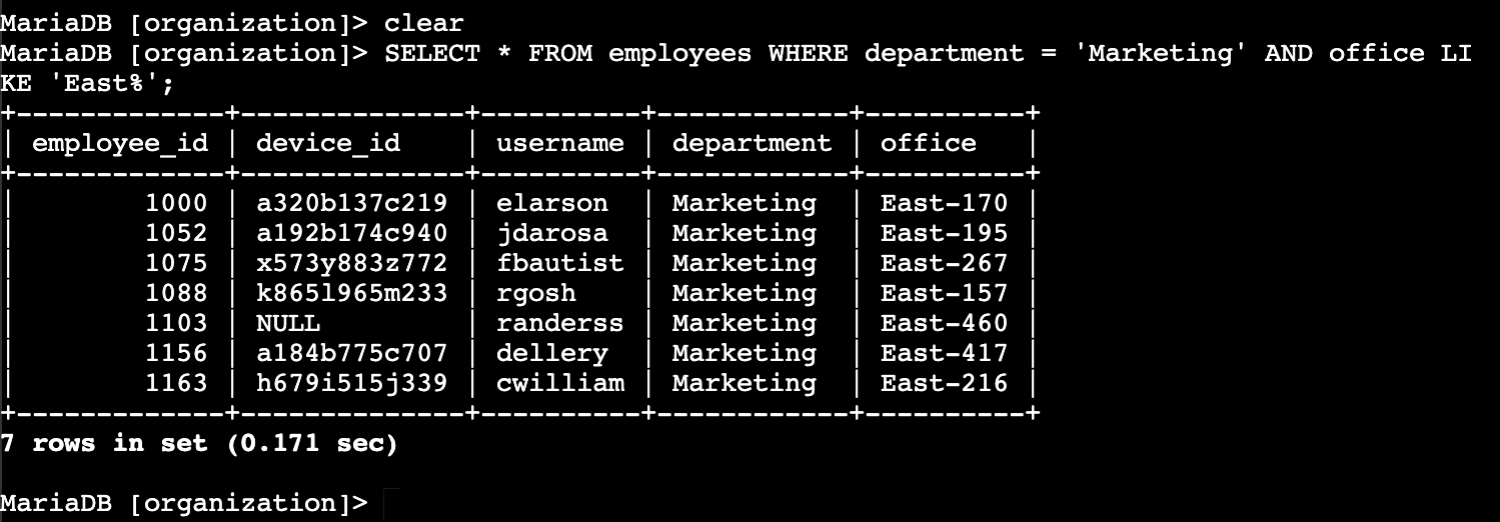
## %: Es un comodín que significa "cualquier secuencia de cero o más caracteres”.

## 

## Retrieve employees in Marketing

Ahora filtraremos los usuarios de Marketing dentro las oficinas que se encuentran en el Este.

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



**FROM employees**: Especifica que estamos extrayendo datos de la tabla llamada employees, que contiene la información de los empleados.

**WHERE department = 'Marketing' AND office LIKE 'East%'**: Esta es la parte que filtra los datos.

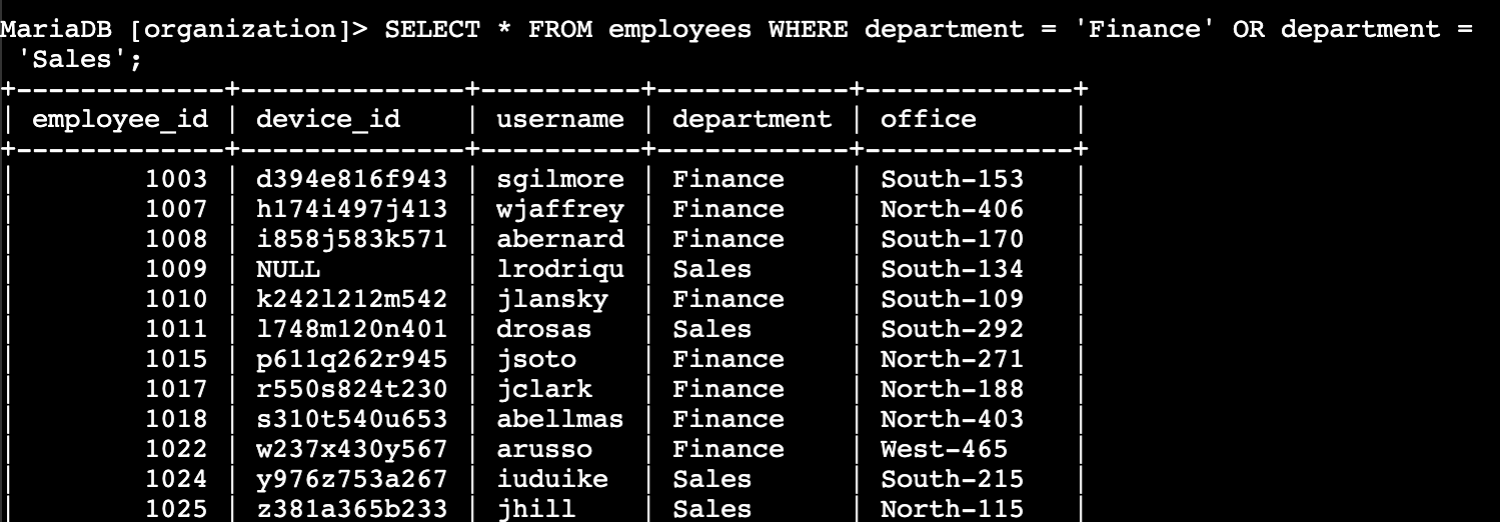
* **department = 'Marketing'**: Esta es la primera condición. Busca registros donde el valor en la columna department sea exactamente 'Marketing'.
* **AND**: Este operador lógico nos dice que **ambas condiciones deben ser verdaderas** para que un registro sea incluido. Si una no se cumple, el registro se descarta.
* **office LIKE 'East%'**: Esta es la segunda condición.
  + office: Es la columna que contiene la ubicación de la oficina del empleado.
  + LIKE: Es un operador que se utiliza para buscar patrones de texto, no una coincidencia exacta.
  + 'East%': Es el patrón que estamos buscando. 'East' significa que la cadena debe comenzar con esas letras, y % es un comodín que representa cualquier secuencia de cero o más caracteres. Así, coincidirá con 'East Wing', 'East Coast', 'East Office', etc.

## Retrieve employees in Finance or Sales

Filtramos en la base de datos para realizar una actualización de seguridad en las máquinas de los departamentos Finanzas y Ventas:

Ejecutaremos el siguiente comando:

SELECT \* FROM employees WHERE department = 'Finance' OR department = 'Sales';

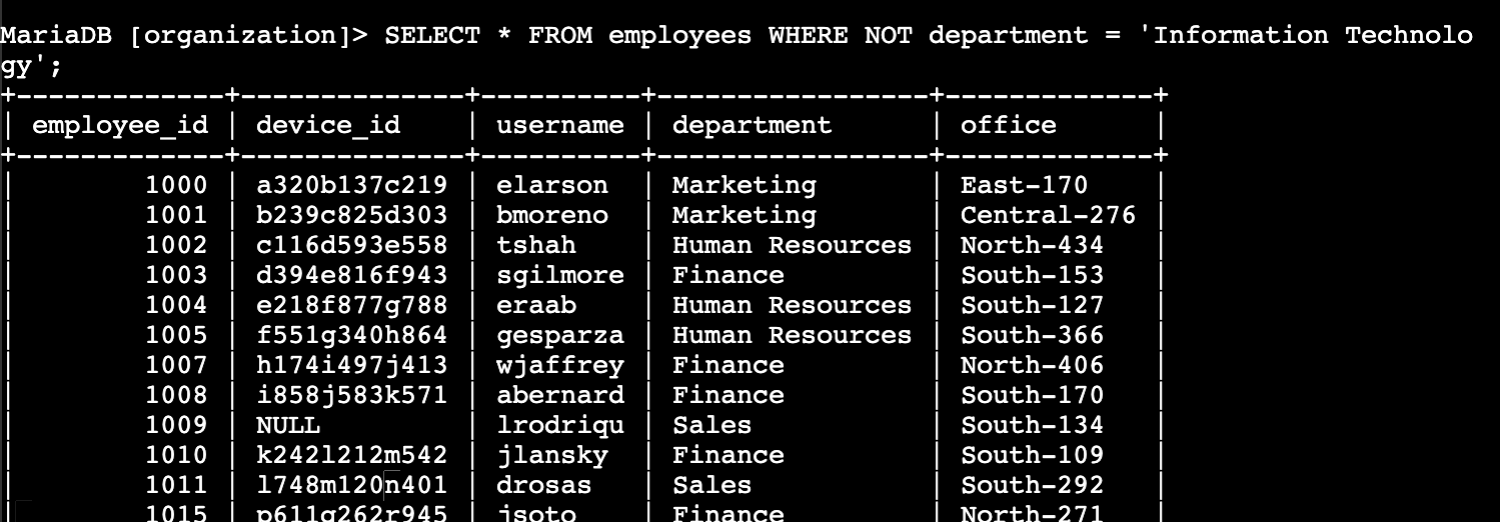


**FROM employees**: Específica que estamos extrayendo los datos de la tabla llamada employees, que contiene la información de los empleados.

**WHERE department = 'Finance' OR department = 'Sales'**: Esta es la parte que filtra los datos.

* **department = 'Finance'**: Esta es la primera condición. Busca registros donde el valor en la columna department sea exactamente 'Finance'.
* **OR**: Este es el operador clave aquí. Nos dice que un registro será incluido en el resultado si cumple **al menos una de las dos condiciones**. No es necesario que se cumplan ambas.
* **department = 'Sales'**: Esta es la segunda condición. Busca registros donde el valor en la columna department sea exactamente 'Sales'.

## Retrieve all employees not in IT

Sacaremos los registros en los que los trabajadores no se encuentran en el departamento IT:

## El comando usado es el siguiente:

SELECT \* FROM employees WHERE NOT department = 'Information Technology';

**FROM employees**: Específica que estamos extrayendo los datos de la tabla llamada employees, la cual contiene la información de los empleados.

**WHERE NOT department = 'Information Technology'**: Esta es la parte que filtra los datos.

* **NOT**: Este operador es un negador. Invierte el resultado de la condición que le sigue. Si la condición 'department = 'Information Technology'' es verdadera para un empleado (es decir, sí es de TI), NOT la convierte en falsa, y ese empleado no se incluirá. Si la condición es falsa (es decir, no es de TI), NOT la convierte en verdadera, y el empleado sí se incluirá.
* **department = 'Information Technology'**: Esta es la condición que se niega. Busca registros donde el valor en la columna department sea exactamente 'Information Technology'.

## Summary

## **Resumen de la Investigación de Seguridad con SQL**

Como profesionales de seguridad, nuestro objetivo es **investigar incidentes y mantener seguros los sistemas** de la organización. Para ello, utilizamos SQL como nuestra herramienta principal para **analizar datos** de las tablas employees (empleados) y log\_in\_attempts (intentos de inicio de sesión).

Hemos explorado varios **filtros SQL** para identificar posibles problemas de seguridad y patrones relevantes:

* **Intentos de Inicio de Sesión Fallidos Fuera de Horario**: Hemos buscado **intentos de inicio de sesión fallidos que ocurrieron después de las 6:00 PM** (SELECT \* FROM log\_in\_attempts WHERE login\_time > '18:00' AND success = '0';). Esto nos ayuda a detectar actividades sospechosas fuera del horario laboral habitual.
* **Intentos de Inicio de Sesión en Fechas Específicas**: También hemos recuperado todos los **intentos de inicio de sesión en el 8 y 9 de mayo de 2022** (SELECT \* FROM log\_in\_attempts WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08';). Esto es útil para revisar la actividad en días concretos.
* **Intentos de Inicio de Sesión Fuera de México**: Hemos filtrado los intentos de inicio de sesión que **no provienen de México** (SELECT \* FROM log\_in\_attempts WHERE NOT country LIKE 'MEX%';), permitiéndonos identificar accesos desde ubicaciones geográficas inesperadas.
* **Empleados de Marketing en Oficinas del Este**: Además, hemos localizado a los **empleados del departamento de Marketing que trabajan en oficinas con nombre que empieza por 'East'** (SELECT \* FROM employees WHERE department = 'Marketing' AND office LIKE 'East%';).
* **Empleados de Finanzas o Ventas**: Hemos identificado a todos los **empleados que pertenecen al departamento de Finanzas o al de Ventas** (SELECT \* FROM employees WHERE department = 'Finance' OR department = 'Sales';), lo cual es relevante para actualizaciones de seguridad dirigidas.
* **Todos los Empleados Excepto los de TI**: Finalmente, hemos obtenido un listado de **todos los empleados que no pertenecen al departamento de Tecnología de la Información** (SELECT \* FROM employees WHERE NOT department = 'Information Technology';), útil para segmentar análisis o comunicaciones.

En resumen, utilizando estas consultas SQL, podemos **escanear y filtrar grandes volúmenes de registros** de la empresa para **detectar y resolver problemas de seguridad** relacionados con accesos de usuarios y equipos de empleados.