# **Project: SQL Query and Filtering for Security Analysis**

## **Project Description**

In this project, SQL was used to retrieve critical information from an organization’s database. The objective was to investigate potential security incidents related to login attempts and employee information. Filters were applied to SQL queries to identify specific patterns and potential issues, aiding in decision-making regarding security updates and system management.

## **SQL Usage Details**

* **Using LIKE for Pattern Matching:** The LIKE keyword is used with the % symbol to search for patterns in text strings. This is useful for filtering data that contains certain substrings or specific patterns.
* **Filtering by Dates and Times:** Filtering by dates and times allows for the extraction of records within a specific time range. In SQL, this is achieved using comparison operators and appropriate date and time formats.
* **Using AND and OR for Multiple Conditions:** The AND and OR operators are used to combine multiple conditions in an SQL query. AND requires all conditions to be true, while OR allows for any of the conditions to be true.
* **Using NOT in Filters:** The NOT operator is used to exclude certain values or patterns in an SQL query. This helps filter out records that do not meet a specific condition.

## **SQL Queries and Explanations**

### **1. Retrieve After Hours Failed Login Attempts**

**SQL Query:**

sql

Copiar código

SELECT event\_id, login\_time

FROM log\_in\_attempts

WHERE success = 0

AND login\_time > '18:00:00';

**Explanation:** This query identifies all failed login attempts (success = 0) that occurred after 18:00:00. The AND filter ensures that only failed attempts outside of regular working hours are selected.

**Screenshot:** [Include a screenshot of the executed query and the obtained results here.]

### **2. Retrieve Login Attempts on Specific Dates**

**SQL Query:**

sql

Copiar código

SELECT event\_id, login\_date

FROM log\_in\_attempts

WHERE login\_date = '2022-05-09'

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

**Explanation:** This query retrieves event\_id and login\_date of login attempts that occurred on 08-05-2022 and 09-05-2022. The OR operator filters records from both dates.

**Screenshot:** [Include a screenshot of the executed query and the obtained results here.]

### **3. Retrieve Login Attempts Outside of Mexico**

**SQL Query:**

sql

Copiar código

SELECT event\_id, country

FROM log\_in\_attempts

WHERE country NOT LIKE '%MEX%'

AND country NOT LIKE '%MEXICO%';

**Explanation:** This query selects event\_id and country of login attempts that occurred outside Mexico. The NOT LIKE operator is used to exclude records with values containing 'MEX' or 'MEXICO'.

**Screenshot:** [Include a screenshot of the executed query and the obtained results here.]

### **4. Retrieve Employees in Marketing**

**SQL Query:**

sql

Copiar código

SELECT employee\_id, username, office

FROM employees

WHERE department LIKE '%Marketing%'

AND office LIKE 'East-%';

**Explanation:** This query retrieves employee\_id, username, and office of employees in the Marketing department located in the East building. The LIKE operator is used to match patterns in the department and office.

**Screenshot:** [Include a screenshot of the executed query and the obtained results here.]

### **5. Retrieve Employees in Finance or Sales**

**SQL Query:**

sql

Copiar código

SELECT employee\_id, username, department

FROM employees

WHERE department LIKE '%Sales%'

OR department LIKE '%Finance%';

**Explanation:** This query lists employee\_id, username, and department of employees in the Sales or Finance departments. The OR operator combines both filters to get results from either department.

**Screenshot:** [Include a screenshot of the executed query and the obtained results here.]

### **6. Retrieve All Employees Not in IT**

**SQL Query:**

sql

Copiar código

SELECT employee\_id, username, department

FROM employees

WHERE department NOT LIKE '%IT%';

**Explanation:** This query selects employee\_id, username, and department of employees whose department does not include 'IT'. The NOT LIKE operator is used to exclude all records related to IT.

**Screenshot:** [Include a screenshot of the executed query and the obtained results here.]

## **Summary**

This document details how to apply SQL filters to investigate and analyze data from an organization’s database. Queries were performed to identify failed login attempts outside of working hours, suspicious activities on specific dates, and employee details across different departments. Queries were also designed to find information outside specific locations, such as Mexico.

## **Conclusion**

SQL queries are powerful tools for data analysis and identifying security issues. Applying specific filters allows analysts to extract relevant information and make informed decisions regarding security updates and system management. Skills in constructing and executing precise queries are essential for maintaining the integrity and security of information systems.

## **Recommendations**

* **Implement Monitoring:** Use similar queries for real-time monitoring of unusual activities and security failures.
* **Proactive Updates:** Perform security updates and patches based on the data obtained from queries.
* **Regular Review:** Conduct trend analysis and periodic reviews of access logs to detect suspicious behavior patterns.

# **Proyecto: Consulta y Filtrado de Datos en SQL para Análisis de Seguridad**

## **Descripción del Proyecto**

En este proyecto, se ha utilizado SQL para recuperar información crítica de la base de datos de una organización. El objetivo es investigar posibles incidentes de seguridad relacionados con los intentos de inicio de sesión y la información de los empleados. Se han aplicado filtros en las consultas SQL para identificar patrones específicos y problemas potenciales, facilitando la toma de decisiones sobre actualizaciones de seguridad y la gestión de sistemas.

## **Detalles de Uso de SQL**

* **Uso de LIKE para Buscar Patrones:** La palabra clave LIKE se utiliza con el símbolo % para buscar patrones en cadenas de texto. Esto es útil para filtrar datos que contienen ciertas subcadenas o patrones específicos.
* **Filtrado por Fechas y Horas:** El filtrado por fechas y horas permite extraer registros dentro de un rango de tiempo específico. En SQL, esto se hace usando operadores de comparación y formatos de fecha y hora apropiados.
* **Uso de AND y OR para Filtrar Múltiples Condiciones:** Los operadores AND y OR se utilizan para combinar múltiples condiciones en una consulta SQL. AND requiere que todas las condiciones sean verdaderas, mientras que OR permite que cualquiera de las condiciones sea verdadera.
* **Uso de NOT en Filtros:** El operador NOT se usa para excluir ciertos valores o patrones en una consulta SQL. Esto ayuda a filtrar registros que no cumplen con una condición específica.

## **Consultas SQL y Explicaciones**

### **1. Recupera Intentos Fallidos de Inicio de Sesión Fuera del Horario Laboral**

**Consulta SQL:**

sql

SELECT event\_id, login\_time

FROM log\_in\_attempts

WHERE success = 0

AND login\_time > '18:00:00';

**Explicación:** Esta consulta identifica todos los intentos de inicio de sesión fallidos (success = 0) que ocurrieron después de las 18:00:00. El filtro AND asegura que solo se seleccionen los intentos fallidos fuera del horario laboral establecido.

**Captura de Pantalla:** [Aquí debes incluir una captura de pantalla de la consulta ejecutada y los resultados obtenidos.]

### **2. Recupera Intentos de Inicio de Sesión en Fechas Específicas**

**Consulta SQL:**

sql

SELECT event\_id, login\_date

FROM log\_in\_attempts

WHERE login\_date = '2022-05-09'

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

**Explicación:** Esta consulta extrae los event\_id y login\_date de los intentos de inicio de sesión que ocurrieron el 08-05-2022 y el 09-05-2022. El operador OR permite filtrar registros de ambos días.

**Captura de Pantalla:** [Aquí debes incluir una captura de pantalla de la consulta ejecutada y los resultados obtenidos.]

### **3. Recupera Intentos de Inicio de Sesión Fuera de México**

**Consulta SQL:**

sql

SELECT event\_id, country

FROM log\_in\_attempts

WHERE country NOT LIKE '%MEX%'

AND country NOT LIKE '%MEXICO%';

**Explicación:** Esta consulta selecciona los event\_id y country de los intentos de inicio de sesión que ocurrieron fuera de México. Se usa NOT LIKE para excluir registros con valores que contienen 'MEX' o 'MEXICO'.

**Captura de Pantalla:** [Aquí debes incluir una captura de pantalla de la consulta ejecutada y los resultados obtenidos.]

### **4. Recupera Empleados en Marketing**

**Consulta SQL:**

sql

SELECT employee\_id, username, office

FROM employees

WHERE department LIKE '%Marketing%'

AND office LIKE 'East-%';

**Explicación:** Esta consulta recupera el employee\_id, username, y office de empleados en el departamento de Marketing que están ubicados en el edificio Este. El operador LIKE se utiliza para buscar coincidencias de patrón en el departamento y la oficina.

**Captura de Pantalla:** [Aquí debes incluir una captura de pantalla de la consulta ejecutada y los resultados obtenidos.]

### **5. Recupera Empleados en Finanzas o Ventas**

**Consulta SQL:**

sql

SELECT employee\_id, username, department

FROM employees

WHERE department LIKE '%Sales%'

OR department LIKE '%Finance%';

**Explicación:** Esta consulta lista los employee\_id, username, y department de empleados en los departamentos de Ventas o Finanzas. El operador OR permite combinar ambos filtros para obtener resultados de cualquiera de los dos departamentos.

**Captura de Pantalla:** [Aquí debes incluir una captura de pantalla de la consulta ejecutada y los resultados obtenidos.]

### **6. Recupera Todos los Empleados No en TI**

**Consulta SQL:**

sql

SELECT employee\_id, username, department

FROM employees

WHERE department NOT LIKE '%IT%';

**Explicación:** Esta consulta selecciona el employee\_id, username, y department de empleados cuyo departamento no incluye 'IT'. El operador NOT LIKE asegura que se excluyan todos los registros relacionados con TI.

**Captura de Pantalla:** [Aquí debes incluir una captura de pantalla de la consulta ejecutada y los resultados obtenidos.]

## **Resumen**

Este documento detalla cómo aplicar filtros SQL para investigar y analizar datos de una base de datos de organización. Se han realizado consultas para identificar intentos de inicio de sesión fallidos fuera del horario laboral, actividades sospechosas en fechas específicas, y detalles de empleados en distintos departamentos. Las consultas también han sido diseñadas para buscar información fuera de ubicaciones específicas, como México.

## **Conclusión**

Las consultas SQL son herramientas poderosas para el análisis de datos y la identificación de problemas de seguridad. Al aplicar filtros específicos, los analistas pueden extraer información relevante y tomar decisiones informadas sobre actualizaciones de seguridad y gestión de sistemas. Las habilidades para construir y ejecutar consultas precisas son esenciales para mantener la integridad y seguridad de los sistemas de información.

## **Recomendaciones**

* **Implementación de Monitoreo:** Use consultas similares para el monitoreo en tiempo real de actividades inusuales y fallos de seguridad.
* **Actualizaciones Proactivas:** Realice actualizaciones y parches de seguridad en los sistemas basándose en los datos obtenidos de las consultas.
* **Revisión Periódica:** Realice análisis de tendencias y revisiones periódicas de los registros de acceso para detectar patrones de comportamiento sospechosos.