

Apply filters to SQL queries

Project description

En este proyecto apliqué filtros en consultas SQL para recuperar información específica de una base de datos. Utilicé la cláusula **WHERE**, operadores lógicos como **AND, OR, NOT**, y patrones con **LIKE** para analizar intentos de inicio de sesión y obtener información de empleados según su departamento y ubicación.

Retrieve after hours failed login attempts

Consulta utilizada:

```
sql  
SELECT *  
FROM log_in_attempts  
WHERE login_time > '18:00' AND success = FALSE;
```

Explicación:

Esta consulta recupera todos los intentos de inicio de sesión que ocurrieron **después de las 6:00 p.m.** y que además **fallaron**.

- `login_time > '18:00'` filtra los intentos fuera del horario laboral.
- `success = FALSE` selecciona únicamente los intentos no exitosos.
- `SELECT *` devuelve todos los detalles de esos registros.

Esto permite identificar actividad sospechosa o accesos fallidos fuera del horario habitual.

Retrieve login attempts on specific dates

Consulta utilizada:

sql

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

Explicación:

Esta consulta obtiene todos los intentos de inicio de sesión ocurridos **exactamente** el 8 y 9 de mayo de 2022.

- El operador **OR** permite incluir múltiples fechas específicas.
- **SELECT *** muestra toda la información asociada a esos intentos.

Es útil para investigar actividad puntual en días relevantes.

Retrieve login attempts outside of Mexico

Consulta utilizada:

sql

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

Explicación:

Esta consulta recupera todos los intentos de inicio de sesión que **no provienen de México**.

- **country LIKE 'MEX%'** identifica cualquier país cuyo código o nombre comience con "MEX".
- **NOT** invierte la condición para excluirlos.

Esto ayuda a detectar accesos desde ubicaciones externas o inusuales.

Retrieve employees in Marketing

Consulta utilizada:

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

Explicación:

Esta consulta obtiene a los empleados del departamento de **Marketing** que trabajan en oficinas ubicadas en la **zona Este**.

- `department = 'Marketing'` filtra por departamento.
- `office LIKE 'East%'` selecciona oficinas que comienzan con “East”.
- `AND` exige que ambas condiciones se cumplan.

Es útil para identificar personal de un departamento dentro de una región específica.

Retrieve employees in Finance or Sales

Consulta utilizada:

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

Explicación:

Esta consulta recupera a todos los empleados que pertenecen a **Finance** o **Sales**.

- El operador **OR** permite incluir registros que cumplan cualquiera de las dos condiciones.

Sirve para analizar o comparar información entre ambos departamentos.

Retrieve all employees not in IT

Consulta utilizada:

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

Explicación:

Esta consulta obtiene a todos los empleados que **no pertenecen** al departamento de **Information Technology**.

- **NOT** excluye a los empleados de TI.

Es útil para revisar información del personal de todos los demás departamentos.

Summary

En este proyecto utilicé filtros SQL para recuperar información específica de una base de datos. Aplicué condiciones con **WHERE**, operadores lógicos (**AND, OR, NOT**) y patrones con **LIKE** para analizar intentos de inicio de sesión y clasificar empleados por departamento y ubicación. Estas consultas permitieron obtener datos precisos y relevantes para análisis organizacionales y de seguridad, demostrando cómo los filtros SQL facilitan la extracción de información útil según criterios definidos.