

Documentación: Script para eliminación de datos en SQL Server

Introducción

Este script está diseñado para conectarse a una base de datos SQL Server utilizando la biblioteca pymssql de Python.

Su propósito es limpiar los datos de varias tablas específicas deshabilitando temporalmente las restricciones de claves foráneas, eliminando los registros y luego reactivando las restricciones.

Requisitos previos

Instalación de dependencias

Asegúrate de tener instalada la biblioteca pymssql:

```
pip install pymssql
```

Configuración de base de datos

Antes de ejecutar el script, necesitas:

1. Tener un servidor de SQL Server configurado.
2. Configurar las credenciales de acceso a la base de datos.

Código

```
import pymssql

# Conectar a SQL Server utilizando pymssql

conn = pymssql.connect(

    server="LAPTOP-007NV287", # Dirección del servidor
```

```

user="williamxlr",          # Nombre de usuario

password="Al3xW$1978",     # Contraseña

database="monokera"        # Nombre de la base de datos

)

cursor = conn.cursor()

def delete_rows():

    try:

        cursor.execute("EXEC sp_MSforeachtable 'ALTER TABLE ? NOCHECK CONSTRAINT all'")

        print("Restricciones de claves foráneas deshabilitadas.")

        tables_to_delete_from = [

            'agents',

            'premium',

            'payments',

            'insured',

            'claims',

            'policy',

        ]

        for table in tables_to_delete_from:

            cursor.execute(f"DELETE FROM {table}")

            print(f"Datos de la tabla {table} eliminados correctamente.")

        conn.commit()

```

```
        cursor.execute("EXEC sp_MSforeachtable 'ALTER TABLE ? CHECK CONSTRAINT all'")

        print("Restricciones de claves foráneas habilitadas.")

except Exception as e:

    print(f"Error: {e}")

    conn.rollback()

# Llamar a la función

delete_rows()

# Cerrar la conexión

cursor.close()

conn.close()
```

Consideraciones de seguridad

- Manejo de credenciales: No incluyas credenciales sensibles en el código fuente. Usa variables de entorno para configurarlas de manera segura.
- Transacciones: Se utiliza `conn.commit()` para confirmar los cambios y `conn.rollback()` en caso de errores.
- Validación: Verifica las tablas y datos que deseas eliminar para evitar eliminar información crítica accidentalmente.

Posibles errores

1. Fallo en la conexión:

- Verifica la dirección del servidor, usuario y contraseña.
- Asegúrate de que el servidor esté accesible desde tu máquina.

2. Restricciones de claves foráneas:

- Si hay tablas no incluidas que dependen de las seleccionadas, podrían ocurrir conflictos.

3. Permisos insuficientes:

- El usuario configurado debe tener permisos para deshabilitar restricciones y eliminar datos.

Licencia

Este código es de uso libre, pero debe adaptarse a tus necesidades específicas. Asegúrate de probarlo en un entorno controlado antes de implementarlo en producción.