

Documentación: Conversor SQL

Descripción general

Este script está diseñado para leer un archivo de Excel, inferir los tipos de datos de sus columnas, y generar dinámicamente una sentencia SQL que contiene tanto la definición de la tabla como los comandos de inserción de los datos. El archivo SQL resultante puede ser utilizado para crear y poblar una base de datos en un sistema de gestión de bases de datos SQL.

Con el tiempo, se agregarán más tipos de archivos que podrán ser convertidos a SQL, tales como **CSV**, **TXT**, **JSON** y **XML**. De esta manera, el script se expandirá para soportar una variedad más amplia de formatos de datos.

Librerías utilizadas

- **pandas**: Se utiliza para manejar y procesar los datos del archivo Excel. Esta librería facilita la lectura del archivo y la manipulación de las columnas y filas.
- **numpy**: Aunque importado, no se utiliza explícitamente en el código, pero podría ser útil para manejar tipos de datos específicos o realizar operaciones en el futuro.
- **tkinter**: Se utiliza para mostrar una interfaz gráfica que permite al usuario seleccionar el archivo de Excel desde el sistema de archivos.

Funciones

1. inferir_tipo_sql(serie)

Esta función infiere el tipo de dato SQL apropiado para una serie (columna) del DataFrame de pandas. La función analiza los valores no nulos de la columna y devuelve un tipo SQL correspondiente.

Parámetros:

- **serie** (pandas.Series): Una columna del DataFrame que contiene los datos a ser evaluados.

Valor de retorno:

- **str**: Devuelve una cadena de texto que representa el tipo de dato SQL inferido para la columna. Los tipos posibles son:

- INT si todos los valores son enteros.
- FLOAT si todos los valores son flotantes.
- NVARCHAR(255) si los valores son de texto o si no se puede determinar un tipo específico.

2. excel_a_sql()

Esta función es la principal del script y se encarga de llevar a cabo la conversión completa del archivo Excel a una sentencia SQL. Primero, permite al usuario seleccionar el archivo Excel, luego lee los datos, genera las sentencias CREATE TABLE y INSERT INTO y guarda el resultado en un archivo SQL.

Parámetros:

No requiere parámetros de entrada, ya que interactúa con el sistema de archivos y pide la entrada del usuario durante la ejecución.

Flujo:

1. **Selección del archivo Excel:** Utiliza tkinter para abrir un cuadro de diálogo de selección de archivo. El archivo debe tener formato .xlsx o .xls.
2. **Lectura del archivo Excel:** Usando pandas.read_excel(), se carga el archivo Excel en un DataFrame.
3. **Inferencia de tipos SQL:** Se invoca la función inferir_tipo_sql() para cada columna del DataFrame, determinando el tipo de dato SQL que debe usarse.
4. **Generación de SQL:**
 - Se genera una sentencia CREATE TABLE para definir la estructura de la tabla SQL.
 - Se crea la sentencia INSERT INTO con los valores de las filas, manejando casos de valores NULL y texto de manera adecuada.
5. **Guardar el archivo SQL:** El SQL generado se guarda en un archivo con el nombre proporcionado por el usuario. La extensión .sql se agrega automáticamente.

Valor de retorno:

No devuelve ningún valor. El resultado final es un archivo .sql que contiene el SQL generado.

Detalles adicionales

- **Manejo de valores nulos:** Si una celda contiene un valor nulo (NaN en pandas), la sentencia SQL insertará NULL en la base de datos.
- **Manejo de valores de texto:** Los valores de texto se insertan correctamente, escapando las comillas simples dentro de los textos mediante la sustitución de ' por ", para evitar errores en las sentencias SQL.
- **Tipos de datos:** El script infiere tres tipos de datos SQL básicos: INT, FLOAT y NVARCHAR(255). Si se encuentran otros tipos de datos, se asigna NVARCHAR(255) como tipo por defecto.

Expansión futura

Con el tiempo, se agregarán más tipos de archivos que podrán ser convertidos a SQL. Entre los formatos que se agregarán están:

- **CSV:** Archivos de valores separados por comas.
- **TXT:** Archivos de texto con datos estructurados.
- **JSON:** Archivos en formato JSON, que es comúnmente utilizado para intercambiar datos entre sistemas.
- **XML:** Archivos en formato XML, utilizados ampliamente en diversas aplicaciones de software.

La incorporación de estos nuevos formatos permitirá a los usuarios convertir una gama más amplia de archivos en sentencias SQL, mejorando la flexibilidad y utilidad del script.

Ejemplo de salida

Si el archivo Excel contiene una tabla con las siguientes columnas:

ID	NOMBRE	EDAD	SALARIO
1	Juan	23	3200.01
2	Ángel	18	2500.50
3	María	51	4500.00

RESULTADO EN SQL

```
CREATE TABLE [archivo_excel] (  
    [ID] INT NULL,  
    [Nombre] NVARCHAR(255) NULL,  
    [Edad] INT NULL,  
    [Salario] FLOAT NULL  
);  
  
INSERT INTO [archivo_excel] ([ID], [Nombre], [Edad], [Salario]) VALUES  
    (1, N'Juan', 30, 2500.5),  
    (2, N'María', 28, 3200.0),  
    (3, N'Pedro', 35, 4500.0);
```

Conclusión

Este script automatiza la conversión de un archivo Excel a una sentencia SQL para crear y poblar una tabla en una base de datos. Es útil para tareas de migración de datos o cuando se necesita convertir rápidamente datos tabulares a un formato que pueda ser procesado por un sistema de bases de datos SQL.

Además, en futuras versiones, se añadirá soporte para más tipos de archivos, como **CSV**, **TXT**, **JSON** y **XML**, lo que ampliará aún más las capacidades del script.