Índice

Almacenamiento de datos

En este tutorial vamos a explicar cómo almacenar datos a partir de diccionarios Python en una base de datos relacional, para la que en este caso usaremos el conector compatible con los gestores MariaDB y MySQL.

Requisitos previos

Se necesitan los siguientes requisitos previos:

1. Tener descargado MariaDB en nuestro equipo habiendo configurado una conexión local con usuario y contraseñas conocidos.
2. Tener alguna interfaz de usuario para visualizar nuestra base de datos como pueda ser DBeaver o similares.
3. Haber creado una base de datos para almacenar nuestros datos.
4. Tener creada en nuestra base de datos una tabla con las columnas correspondientes a los campos de nuestros diccionarios y tipos deseados, además de una columna id que actuará como primary key.

Inserción de datos con conector mysql-python-conector

En esta sección explicaremos los pasos necesarios para insertar nuestros datos desde Python a nuestra base de datos usando el conector mysql-python.

1. Instalar el paquete necesario

Asegúrate de tener instalada la biblioteca mysql-connector-python para poder conectarte a la base de datos MariaDB:

```
pip install mysql-connector-python
```

2. Configurar la conexión a la base de datos

Usa BeautifulSoup para analizar el contenido HTML de la página:

```
import mysql.connector

# Configuración de la conexión a la base de datos
config = {
    'user': 'tu_usuario',
    'password': 'tu_contraseña',
    'host': 'tu_host',  # Puede ser 'localhost' si la base de datos está en tu máquina
    'database': 'tu_base_de_datos',
}
```

En este bloque, importamos el módulo mysql.connector y configuramos los parámetros necesarios para establecer la conexión con la base de datos MariaDB. Asegúrate de reemplazar 'tu_usuario', 'tu_contraseña', 'tu_host', y 'tu_base_de_datos' con los valores específicos de tu base de datos.

3. Establecer la conexión y crear un cursor

Aquí, creamos una conexión a la base de datos utilizando la configuración proporcionada. conn es el objeto de conexión y cursor es un objeto cursor que usaremos para ejecutar comandos SQL.

```
# Conectar a la base de datos
conn = mysql.connector.connect(**config)
cursor = conn.cursor()
```

4. Preparar los datos que deseas insertar

Tener preparado los datos que vamos a insertar, por ejemplo en formato diccionario.

```
# Diccionario que queremos insertar en la tabla
nuevo_registro = {'nombre': 'Juan', 'edad': 25}
```

5. Construir y ejecutar la consulta de inserción

Aquí, creamos una cadena de consulta SQL (insert_query) que especifica la estructura de la inserción. Luego, utilizamos el método execute del cursor para ejecutar la consulta, pasando los valores del diccionario (nuevo_registro['nombre'] y nuevo_registro['edad']) como parámetros.

```
# Insertar el diccionario en la tabla
insert_query = "INSERT INTO tu_tabla (nombre, edad) VALUES (%s, %s)"
cursor.execute(insert_query, (nuevo_registro['nombre'],
nuevo_registro['edad']))
```

6. Forma alternativa para los diccionarios

Para realizar la inserción a partir de las claves de un diccionario, se puede modificar la sentencia para que los parámetros de la consulta los enlace directamente con los valores del diccionario, enlazándolos por el nombre de las claves.

De esta manera:

- se sustituye **%s → %(clave_diccionario)s**
- a la sentencia execute se le pasa el diccionario completo.

```
# Insertar el diccionario en la tabla
insert_query = "INSERT INTO tu_tabla (nombre, edad) VALUES (%(nombre)s,
%(edad)s)"
cursor.execute(insert_query, nuevo_registro)
```

7. Guardar cambios y cerrar la conexión

Finalmente, usamos commit para guardar los cambios en la base de datos y cerramos la conexión con close. Es crucial cerrar la conexión después de realizar las operaciones en la base de datos para liberar recursos. Este paso asegura que los datos se guarden correctamente en la base de datos.

```
# Guardar los cambios y cerrar la conexión
conn.commit()
conn.close()
```

Recuperación de datos

A continuación se muestra un ejemplo de cómo se podrían extraer los datos de nuestra base de datos y meterlos en lista de diccionarios , para volver al formato inicial:

```
import mysql.connector

# Conéctate a la base de datos
conexion = mysql.connector.connect(
    host="tu_host",
    user="tu_usuario",
    password="tu_contraseña",
    database="tu_base_de_datos"
)

# Crea un cursor para ejecutar consultas SQL
cursor = conexion.cursor(dictionary=True)

# Ejecuta una consulta SQL para seleccionar datos de una tabla
consulta = "SELECT * FROM tu_tabla"
cursor.execute(consulta)

# Obtiene los nombres de las columnas
columnas = cursor.column_names

# Obtiene todos los resultados de la consulta
resultados = cursor.fetchall()

# Cierra la conexión y el cursor
cursor.close()
conexion.close()

# Construye la lista de diccionarios con los nombres de las columnas como claves
lista_de_diccionarios = []

for resultado in resultados:
    diccionario = {}
    for columna, valor in zip(columnas, resultado):
        diccionario[columna] = valor
    lista_de_diccionarios.append(diccionario)
```

Páginas de referencia y ayuda

Página oficial de la dependencia del conector:
https://pypi.org/project/mysql-connector-python/

Tutorial de W3Schools:
https://www.w3schools.com/python/python_mysql_getstarted.asp