



El futuro digital
es de todos

MinTIC

Ciclo 1

Fundamentos de programación con Python

Sesión 24: Persistencia de datos con Python 2

Programa Ciencias de la Computación e Inteligencia Artificial
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda
Bogotá



UNIVERSIDAD
SERGIO ARBOLEDA

Mision
TIC 2022



Agenda

1. Bases de datos relacionales
2. Instalación y configuración SQLite
3. Creación de bases de datos y tablas
4. SQLite con Python





1. Bases de datos relacionales

- Los sistemas de gestión de bases de datos relacionales (RDBMS) son muy populares y constituyen una parte integral del desarrollo de aplicaciones.
- Existen muchos RDBMS diferentes, por ejemplo, MySQL, PostgreSQL, IBM DB2, Oracle 11g, etc
- Un RDBMS de este tipo es SQLite. SQLite es ampliamente utilizado y es el favorito entre los desarrolladores por muchas razones.





1. Bases de datos relacionales

- Extremadamente ligero (no más de 500 KBs).
- Es sin servidor, lo que significa que no necesita ningún servidor independiente para utilizar sus servicios.
- No tiene una configuración compleja.
- Totalmente transaccional y compatible con la concurrencia.





1. Bases de datos relacionales

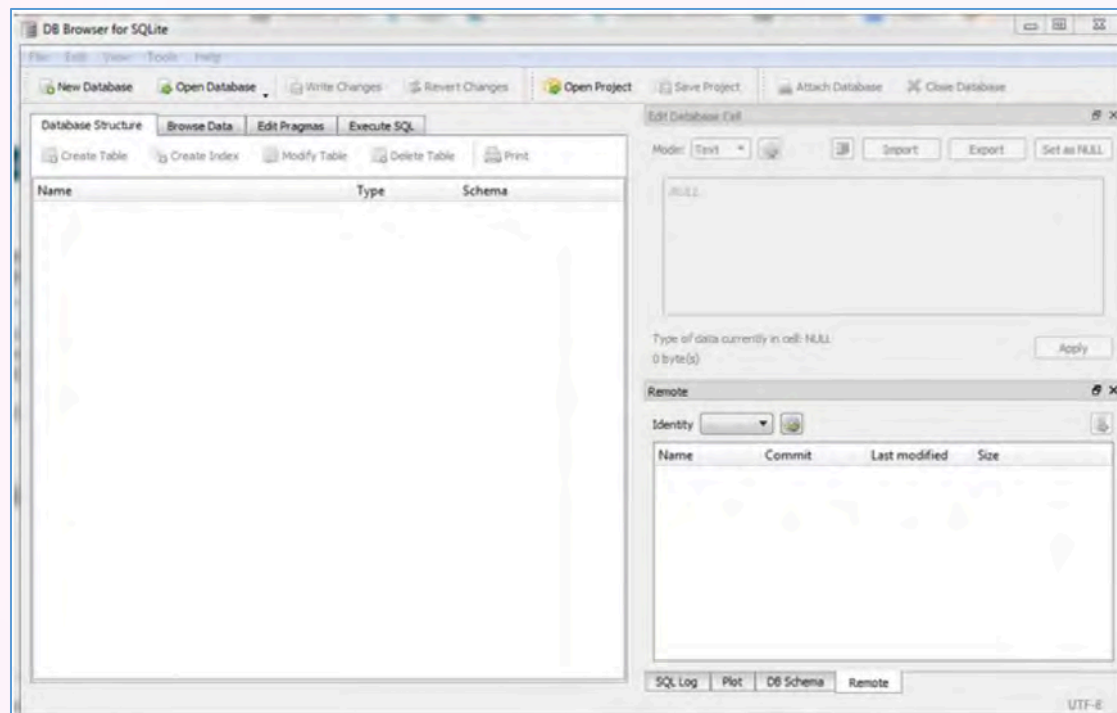
- Sin embargo, también hay algunas limitaciones de SQLite.
- Por ejemplo, no soporta uniones como RIGHT OUTER JOIN y FULL OUTER JOIN. Pero las ventajas son mucho más que las limitaciones.





2. Instalación y configuración

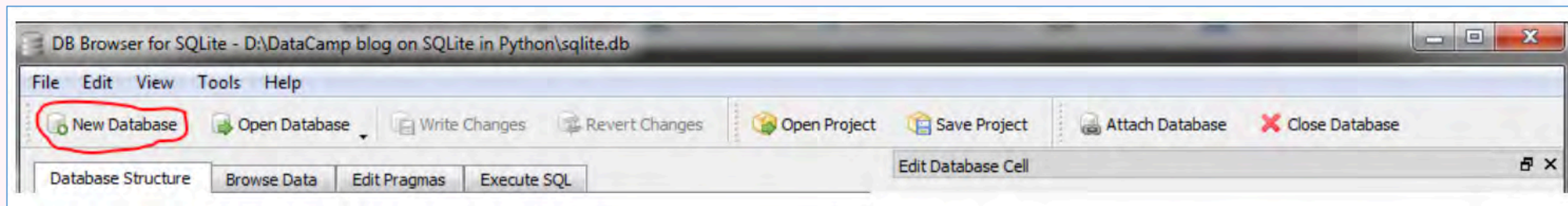
Instalar y configurar SQLite es cuestión de unos minutos. Puedes usar SQLite desde las herramientas de línea de comandos, pero hay una utilidad basada en GUI que te permite usar SQLite a través de una interfaz gráfica decente.





3. Creación de bases de datos y tablas

Una vez que esté listo con la herramienta DB Browser, puede crear una nueva base de datos SQLite para continuar. Para ello, puede hacer clic en la pestaña Nueva base de datos





3. Creación de bases de datos y tablas

- Tras hacer clic en la pestaña Nueva base de datos, se le pedirá que introduzca un nombre para la base de datos.
- Dé un nombre de su elección y proceda.
- Se creará instantáneamente una base de datos vacía con el nombre introducido y se le pedirá que cree una tabla en esa base de datos.





3. Creación de bases de datos y tablas

- Para crear una tabla, es necesario fijar el esquema de la tabla.
- En aras de la comprensión vamos a crear primero una tabla simple llamada consumidores con los siguientes campos y tipos de datos:
 - consumer_id (entero)
 - consumer_full_name (cadena) (no puede ser nula)
 - consumer_email (cadena) (no puede ser nula)
 - consumer_grade (carácter) (no puede ser nulo)





3. Creación de bases de datos y tablas

- Para crear una tabla, haga clic en la pestaña Crear tabla y se le pedirá que introduzca los detalles de la tabla que desea crear

Table

consumers

▼ Advanced

Fields

Add field Remove field Move field up Move field down

Name	Type	NN	PK	AI	U	Default	Check
consumer_id	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
consumer_full_name	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
consumer_email	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
consumer_grade	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```
1 CREATE TABLE "consumers" (  
2     "consumer_id" INTEGER PRIMARY KEY AUTOINCREMENT,  
3     "consumer_full_name" TEXT NOT NULL,  
4     "consumer_email" TEXT NOT NULL,  
5     "consumer_grade" TEXT NOT NULL  
6 );
```





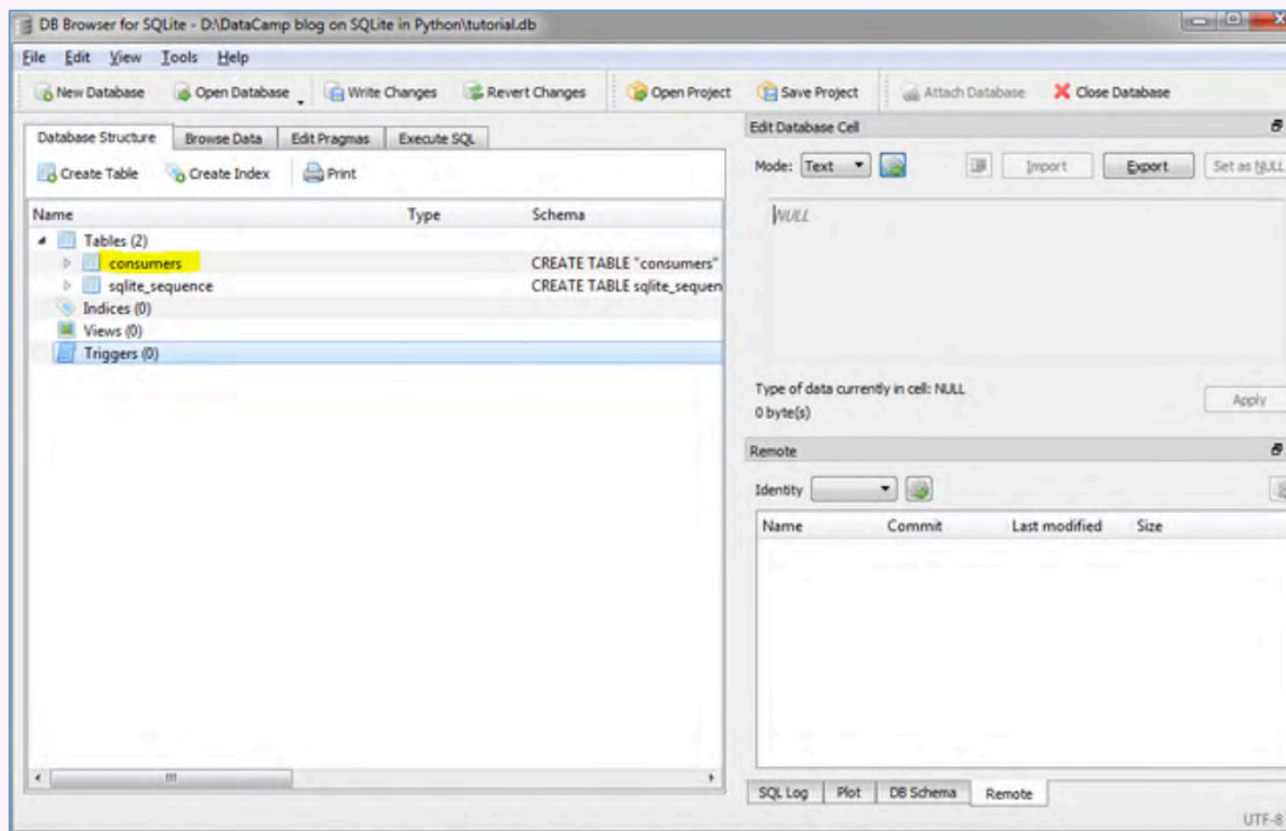
3. Creación de bases de datos y tablas

- Si observa detenidamente la figura anterior, verá que contiene los detalles exactos que quería incorporar a los consumidores de la tabla.
- También puede ver el respectivo SQL para crear la tabla.
- La herramienta DB Browser le permite hacer esto de manera muy eficiente.





3. Creación de bases de datos y tablas





3. Creación de bases de datos y tablas

- La tabla *sqlite_sequence* está ahí porque si se especifica que el campo `consumer_id` se autoincrementa, SQLite crea una tabla aparte para mantener las secuencias.
- También puede ejecutar otras consultas SQL yendo a la sección Ejecutar SQL.
- Ahora, antes de interactuar con las bases de datos SQLite utilizando Python, vamos a ver cómo se puede importar un archivo .csv en una base de datos SQLite y utilizarlo para el análisis.





3. Importación de un archivo .csv a una base de datos SQLite

- Para importar un archivo .csv a la base de datos que ha creado, sólo tiene que seguir esta navegación Archivo -> Importar -> Tabla desde archivo CSV.
- Puede utilizar este un archivo .csv para este fin. Contiene detalles sobre diferentes países de todo el mundo. Navegue hasta el archivo y obtendrá un cuadro de diálogo como el siguiente

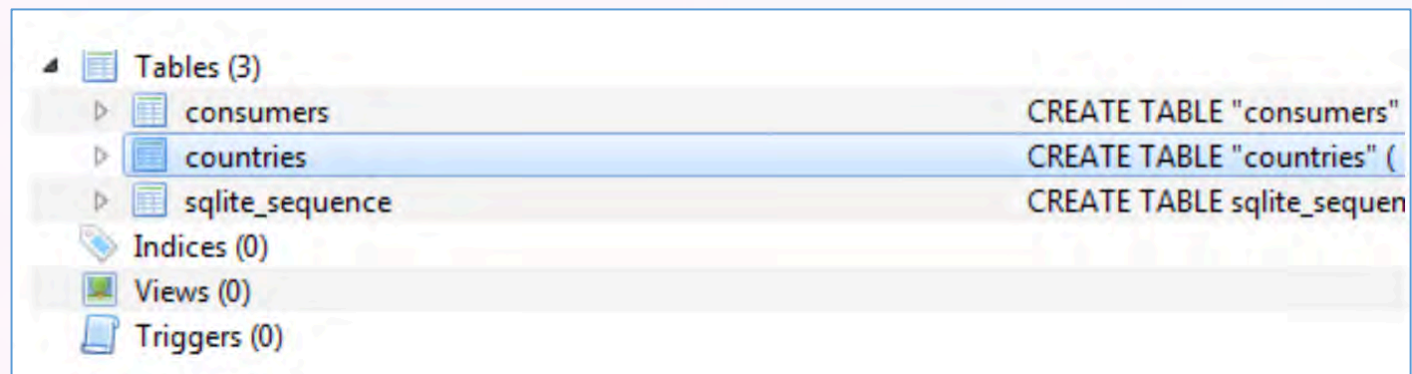
	code	country_name	continent	region	ndependence_yea	local_name	gov_form	
1	AFG	Afghanistan	Asia	Southern and C...	1919	Afganistan/Afq...	Islamic Emirate	Kabv
2	NLD	Netherlands	Europe	Western Europe	1581	Nederland	Constitutional ...	Ams
3	ALB	Albania	Europe	Southern Europe	1912	Shqiperia	Republic	Tira
4	DZA	Algeria	Africa	Northern Africa	1962	Al-Jaza'ir/Algerie	Republic	Algi
5	ASM	American Samoa	Oceania	Polynesia		Amerika Samoa	US Territory	Pag
6	AND	Andorra	Europe	Southern Europe	1278	Andorra	Parliamentary ...	And
7	AGO	Angola	Africa	Central Africa	1975	Angola	Republic	Luar
8	ATG	Antigua and Ba...	North America	Caribbean	1981	Antigua and Ba...	Constitutional ...	Sain
9	ARE	United Arab Em...	Asia	Middle East	1971	Al-Imarat al-'Ar...	Emirate Federat...	Abu
10	ARG	Argentina	South America	South America	1816	Argentina	Federal Republic	Bue





3. Importación de un archivo .csv a una base de datos SQLite

- DB Browser le permite especificar muchas cosas aquí, incluyendo los nombres de la tabla. Asegúrese de marcar la opción Nombres de columnas en la primera línea para que SQLite pueda extraer los nombres de las columnas automáticamente.
- Debería poder ver una entrada para la tabla





4. SQLite en Python

- Ahora se tiene una base de datos y una tabla lista para trabajar.
- Para poder interactuar con una base de datos SQLite usando Python, se necesita el módulo sqlite3
- Ahora, te conectarás a la base de datos que has creado utilizando el método connect() proporcionado por sqlite3. Esto devuelve un objeto Connection.
- Hay que darle la ruta de la base de datos al método connect. Las bases de datos se guardan generalmente con la extensión .db.

```
import sqlite3

conn = sqlite3.connect('tutorial.db')
```





4. SQLite en Python

- Una vez que tengas una Conexión a la base de datos, puedes crear un objeto Cursor y llamar a su método execute() para realizar comandos SQL.

```
cur = conn.cursor()
cur.execute('SELECT * from countries')

<sqlite3.Cursor at 0x398ace0>
```





4. SQLite en Python

Después de ejecutar la sentencia SELECT, se puede:

- Tratar el objeto cursor cur como un iterador llamar al método fetchone() para mostrar una sola fila
- Llamar al método fetchall() para mostrar una lista de filas.

```
print(cur.fetchone())

('AFG', 'Afghanistan', 'Asia', 'Southern and Central Asia'

print(cur.fetchall())

[('NLD', 'Netherlands', 'Europe', 'Western Europe', '1581'
```





4. SQLite en Python

- Puedes hacer que la salida del método fetchall() sea ligeramente más bonita iterando sobre cada una de las filas.

```
for row in cur.execute('SELECT * FROM countries'):  
    print(row)
```





4. SQLite en Python

- Veamos ahora cómo se añade una cláusula where a la consulta y se ejecuta. Obtengamos los detalles del país donde el código = 'AFG'. fetchall() sea ligeramente más bonita iterando sobre cada una de las filas.

```
code = ('AFG',)

cur.execute('SELECT * FROM countries WHERE code = ?', code)

print(cur.fetchone())
```





4. SQLite en Python

- Se pueden insertar registros en una tabla, ya sea uno por uno o muchos registros a la vez. Para ello, vamos a utilizar la tabla de consumidores. Por ahora no contiene ningún registro. Vamos a rellenarla desde aquí.

```
# One by one  
  
cur.execute("INSERT INTO consumers VALUES (1,'John Doe','j  
  
for row in cur.execute('SELECT * FROM consumers'):  
    print(row)
```





Preguntas

