

JOSE GUADALUPE GUERRERO SANCHEZ

I. Con ayuda de la clase "Creación de tablas" realiza una y considera:

- Usa la librería sqlite3.
- El archivo de la base de datos se llamará "musica.db"
- Los campos serán: id, canción, artista, álbum

```
1 import sqlite3
2
3 conexion = sqlite3.connect("musica.db")
4 consulta=conexion.cursor()
5
6 tabla = "CREATE TABLE Music (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, \"
7       \"Artista VARCHAR (30) NOT NULL, \"
8       \"Album VARCHAR (40) NOT NULL, \"
9       \"Cancion VARCHAR (29) NOT NULL);\"
10
11 print (tabla)
12
13 if(consulta.execute(tabla)):
14     print ("La tabla fue creada")
15 else: print ("La tabla no fue creada")
16
17 consulta.close()
18 conexion.commit()
19 conexion.close()
20
21
```

DB Browser for SQLite - D:\Usuarios\Guerrero guadalpe\Escritorio\ITQ\Quinto Semestre (en linea)\Program

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragma Execute SQL

Table: Music

id	Artista	Album	Cancion
Filter	Filter	Filter	Filter

II. Ahora con ayuda de la clase "inserción de registros" crea una función llamada "registros" para insertar los campos:

- cancion
- artista
- álbum

```
1 import sqlite3
2
3 def Insert():
4     db1 = sqlite3.connect("musica.db")
5     consulta = db1.cursor()
6     print("Insert information")
7
8     artist = input("Artist's name: ")
9     album = input("Album: ")
10    song = input("Song: ")
11
12
13
14    strConsulta = "INSERT INTO Music (Artista, Album, Cancion) VALUES \
15    ('"+artist+"', '"+album+"', '"+song+"')\"
16
17    print (strConsulta)
18
19    consulta.execute(strConsulta)
20    print("consulta")
21    consulta.close()
22    db1.commit()
23    db1.close()
24
25    Insert()
```

DB Browser for SQLite - D:\Usuarios\Guerrero guadalpe\Escritorio\ITQ\Quinto Semestre (en linea)\Programacion Avanzada\Capacitate para el empleo POO\ProgramasPython\Tablas_Musica_sqlite3\Musica.py

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragma Execute SQL

Table: Music

id	Artista	Album	Cancion
Filter	Filter	Filter	Filter
1	Stray Kids	IN	Back Door
2	PENTAGON	[12:00]	Daisy

Go to: 1

In [46]:

III. Es turno de hacer una consulta para verificar que fue lo que se escribió sobre la tabla. Para lograrlo consulta la clase "Consulta de registros" y haz lo siguiente:

- Crea una función llamada "consultas".
- Ocupa la sintaxis para hacer consultas.
- Haz un arreglo donde guardes los campos de la tabla en una lista. Esto es para lograr una buena organización.
- Imprime la lista y verifica el resultado en la consola.

Spyder (Python 3.8)

```

Archivo  Editar  Buscar  Código fuente  Ejecutar  Depurar  Terminales  Proyectos  Herramientas  Ver Ayuda

...Programacion Avanzada\Capacitate para el empleo POO\ProgramasPython\Tablas_Musica_sqlite3\Music.py
Operations_JoseGuadalupeGuerreroSanchez.py  tkinter.py  Tablas_sqlite3.py  Music.py*

1  import sqlite3
2
3  def Insert():
4      db1 = sqlite3.connect("musica.db")
5      consulta = db1.cursor()
6      print("\n\nInsert information")
7
8      artist = input("Artist's name: ")
9      album = input("Album: ")
10     song = input("Song: ")
11
12
13
14     strConsulta = "INSERT INTO Music (Artista, Album, Cancion) VALUES \
15 ('"+artist+"','"+album+"','"+song+"')"
16
17     #print (strConsulta)
18
19     consulta.execute(strConsulta)
20     print("consulta")
21     consulta.close()
22     db1.commit()
23     db1.close()
24
25
26 def Consult():
27     db2 = sqlite3.connect("musica.db")
28     #print("Consult Funtion")
29     db2.row_factory = sqlite3.Row
30     consulta = db2.cursor()
31     consulta.execute("select * from Music")
32     filas = consulta.fetchall()
33     lista=[]
34     for fila in filas:
35         s={}
36         s["Artista"] = str(fila["Artista"])
37         s["Album"] = str(fila ["Album"])
38         s["Cancion"] = str(fila["Cancion"])
39         lista.append(s)
40     consulta.close()
41     db2.close()
42     return (lista)
43
44 def Menu():
45     option = int(input("\n 1. Insert data \n 2. Consult data \
46 \n 0. exit \n\t select and option "))
47
48     if option == 1:
49         Insert()
50         Menu()
51     elif option == 2:
52         listSongs = Consult()
53         for songs in listSongs:
54             print("\n\nArtist: ",songs["Artista"])
55             print("Album:      ",songs["Album"])
56             print("Song:        ",songs["Cancion"] )
57     elif option == 0:
58         print("End")
59
60 Menu()

```

Terminal 1/A

```

1. Insert data
2. Consult data
0. exit
select and option 2

```

Artist: Stray Kids
Album: IN
Song: Back Door

Artist: PENTAGON
Album: [12:00]
Song: Daisy

Artist: LOONA
Album: 12:00
Song: VOICE

Artist: TWICE
Album: MORE & MORE
Song: MORE AND MORE

In [60]:

DB Browser for SQLite - D:\Usuarios\Guerrero guadalupe\Escritorio\ITQ\Quinto Semestre (en linea)\Programacion A

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragma Execute SQL Edit Database Cell

Table: Music

id	Artista	Album	Cancion
1	Stray Kids	IN	Back Door
2	PENTAGON	[12:00]	Daisy
3	LOONA	12:00	VOICE
4	TWICE	MORE & ...	MORE AND MORE

Mode: Text

Type of data currently in cell: Text / Numeric
1 character(s)

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name