# How to Create a Database and INSERT Some Data

Creating a database in SQLite is really easy, but the process requires that you know a little SQL to do it. Here's some code that will create a database to hold music albums:

```
import sqlite3

conn = sqlite3.connect("mydatabase.db") # or use :memory: to put it in RAM

cursor = conn.cursor()

# create a table
cursor.execute("""CREATE TABLE albums
                  (title text, artist text, release_date text,
                   publisher text, media_type text)
               """)

# This is SQLite's way of getting the tables created so far
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
print(cursor.fetchall())
```

First we have to import the **sqlite3** module and create a connection to the database. You can pass it a file path, file name or just use use the special string ":memory:" to create the database in memory. In our case, we created it on disk in a file called **mydatabase.db**. Next we create a cursor object, which allows you to interact with the database and add records, among other things. Here we use SQL syntax to create a table named **albums** with 5 text fields: title, artist, release_date, publisher and media_type. SQLite only supports five **data types**: null, integer, real, text and blob. Let's build on this code and insert some data into our new table!

**Note: If you run the CREATE TABLE command and the database already exists, you will receive an error message.**

```
# insert some data
cursor.execute("""INSERT INTO albums
                  VALUES ('Glow', 'Andy Hunter', '7/24/2012',
                          'Xplore Records', 'MP3')"""
               )

# save data to database
conn.commit()

# insert multiple records using the more secure "?" method
albums = [('Exodus', 'Andy Hunter', '7/9/2002', 'Sparrow Records', 'CD'),
          ('Until We Have Faces', 'Red', '2/1/2011', 'Essential Records', 'CD'),
          ('The End is Where We Begin', 'Thousand Foot Krutch', '4/17/2012', 'TFKmusic', 'CD'
          ('The Good Life', 'Trip Lee', '4/10/2012', 'Reach Records', 'CD')]
cursor.executemany("INSERT INTO albums VALUES (?,?,?,?,?)", albums)
conn.commit()

# Show rows inserted so far
cursor.execute("SELECT * from albums;")
print(cursor.fetchall())
```

Here we use the INSERT INTO SQL command to insert a record into our database. Note that each item had to have single quotes around it. This can get complicated when you need to insert strings that include single quotes in them. Anyway, to save the record to the database, we have to **commit** it. The next piece of code shows how to add multiple records at once by using the cursor's **executemany** method. Note that we're using question marks (?) instead of string substitution (%s) to insert the values. Using string substitution is NOT safe and should not be used as it can allow SQL injection attacks to occur. The question mark method is much better and using SQLAlchemy is even better because it does all the escaping for you so you won't have to mess with the annoyances of converting embedded single quotes into something that SQLite will accept.