A Simple Web Project

The Python-SQLite API

The most commonly used API calls are listed in the following table:

Syntax API	Description
`sqlite3.connect(database[, timeout, optional args])`	Opens a connection to the SQLite database file. Add the optional argument **`:memory:`** to open a DB connection to a DB residing in RAM rather than on HD. This API call locks the database until the transaction is committed, in the eventuality that there are multiple connections accessing the DB at the same time. The **`timeout`** parameter specifies how long the connection should wait for the lock to go away until raising an exception (the default value is 5 seconds). If the DB doesn't exist, the call to **`connect()`** will create it, optionally using a full path filename for it.
`connection.cursor([cursorClass])`	This API call creates a cursor which will be used with the Python script throughout.
`cursor.execute(sql[, optional parameters])`	Executes an SQL statement. This statement can take parameters for the SQL values, which will be handed off to the SQL by the Python script.
`connection.commit()`	This method commits the current transaction. If not committed, nothing since the last call to **`commit()`** will be visible from other database connections.
`connection.close()`	Closes connection to the database.

A Simple Web Project

We want to exemplify the use of the Python-SQLite API in the context of creating a simple web application. In order to create and test this, in addition to having a working database (Lecture 1), we need a running web server (Apache), and a number of CGI Python scripts that output dynamic HTML, typically residing in the `/Library/WebServer/CGI-Executables` directory on a MacOS system.

Assuming a few basics already known about HTML, we will proceed developing our simple web project by first creating a simple HTML web page. HTML code typically resides in the `~user/public_html` (e.g. `/Users/awise/public_html`) directory, and the web page can be loaded as `http://hostname/*.html` (e.g. `http://127.0.0.1/index.html`).

Our intention is to create a web project centered on the database implemented in Lecture 1, `testDB.db`, containing the three tables, `Musicians`, `MemberOf`, and `SoloArtists`:

```
sqlite> .fullschema
CREATE TABLE Musicians(
ID INT PRIMARY KEY NOT NULL,
Name CHAR(20),
Age INT,
Address CHAR(20),
Num_Albums INT,
Band_ID INT,
FOREIGN KEY(Band_ID) REFERENCES MemberOf(ID)
);
CREATE TABLE MemberOf(ID INT PRIMARY KEY NOT NULL,
Band CHAR(50) NOT NULL,
Musician_ID INT NOT NULL,
Since INT NOT NULL,
FOREIGN KEY(Musician_ID) REFERENCES Musicians(ID)
);
CREATE TABLE SoloArtists(ID INT PRIMARY KEY NOT NULL,
Name CHAR(20),
Age INT,
Address CHAR(20),
Num_Albums INT,
Band_ID INT,
FOREIGN KEY(Band_ID) REFERENCES MemberOf(ID)
);
```

```
CREATE TABLE Logs(
ID INT NOT NULL,
Date TEXT NOT NULL);
CREATE TRIGGER audit AFTER INSERT
ON Musicians
BEGIN
INSERT INTO Logs(ID, Date)
VALUES(new.ID, datetime('now'))
;
END;
/* No STAT tables available */
```

At this point the tables contain:

```
sqlite> .width 3 20 3 20 3 4
sqlite> SELECT * FROM Musicians;
1    Lyle PUENTE          53   Crompond, NY         6    1
2    Tyler JOSEPH         26   Columbus, OH         4    2
3    Josh DUN             27   Columbus, OH         3    2
4    Sara BAREILLES       35   Eureka, CA           2
5    Robin THICKE         35   Los Angeles, CA      1
6    Patti ROTHBERG       50   New York, NY         8    None

sqlite> SELECT * FROM MemberOf;
1    My Brothers Banned   1    1996
2    Twenty One Pilots    2    2009

sqlite> SELECT * FROM SoloArtists;
1    Lyle PUENTE          53   Crompond, NY         2    1
2    Tyler JOSEPH         26   Columbus, OH         1    2
3    Sara BAREILLES       35   Eureka, CA           2
```

A Simple HTML Page: musicians1.html

The first step towards building this project is to write (a simplified version of) the table contents as a static HTML page:

```
<!DOCTYPE html>
<HTML>
        <HEAD>
                <TITLE>HTML Tables</TITLE>
        </HEAD>
        <BODY>
                <H1>Musicians</H1>
```

```
                <TABLE border=1>
                        <TR>
                                <TH>Name</TH>
                                <TH>Age</TH>
                                <TH>Number of Records</TH>
                        </TR>
                        <TR>
                                <TD>Lyle PUENTE</TD>
                                <TD>53</TD>
                                <TD>6</TD>
                        </TR>
                        <TR>
                                <TD>Tyler JOSEPH</TD>
                                <TD>26</TD>
                                <TD>5</TD>
                        </TR>
                        <TR>
                                <TD>Josh DUN</TD>
                                <TD>27</TD>
                                <TD>3</TD>
                        </TR>
                </TABLE>
        </BODY>
</HTML>
```

This outputs the following table in a web page:

Musicians

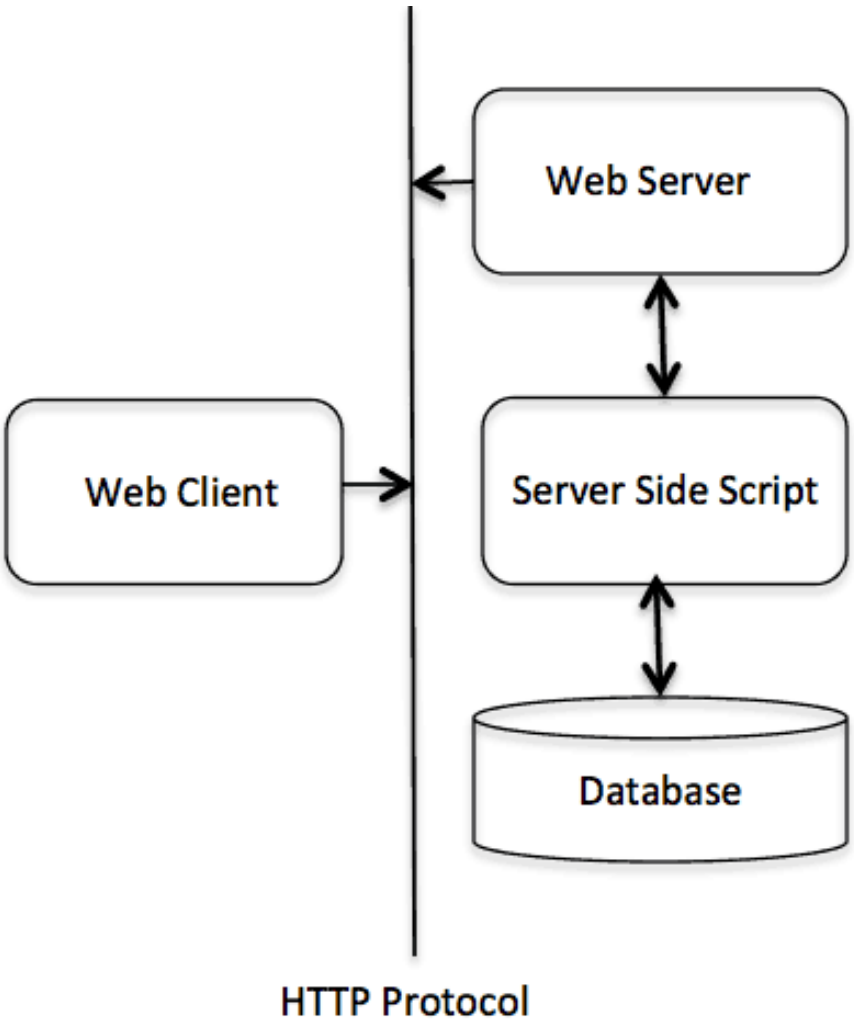
Name	Age	Number of Records
Lyle PUENTE	53	6
Tyler JOSEPH	26	5
Josh DUN	27	3

A Simple CGI Script: musicians1.cgi

Our next step is to produce the same table from a CGI, that is, from a backend Python script querying the SQLite database and outputting dynamic HTML (the word "dynamic" here refers to the contents of the final HTML page depending on the contents being brought up from the database).

A CGI (from Common Gateway Interface=a standard for external gateway programs to interface with HTTP or web servers) script is a program that executes on the server, and whose output is sent for display to the client (browser).

The following diagram, courtesy of tutorialspoint.com, shows the CGI architecture:



In our case, we want to write a Python CGI script that uses the Python-SQLite API to query our SQLite testDB.db musicians database, and uses the result of that query to display the results in an HTML table. As mentioned before, the location of this script on a typical MacOS system is `/Library/WebServer/CGI-Executables`. During the development of our project, we will write several versions of this file:

```
#!/usr/bin/python

import sqlite3

database="/Users/awise/Python/Scripts/testDB.db"
connection=sqlite3.connect(database)
cursor=connection.execute("SELECT * FROM musicians")
print "Content-type: text/html"
print
print "<HTML>"
print "          <HEAD>"
print "                 <TITLE>HTML Tables</TITLE>"
print "          </HEAD>"
print "          <BODY>"
print "                  <H1>Musicians</H1>"
print "                  <TABLE border=1>"
print "                         <TR>"
print "                                 <TH>Name</TH>"
print "                                 <TH>Age</TH>"
print "                                 <TH>Address</TH>"
print "                                 <TH>Number of Records</
TH>"
print "                                 <TH>Band ID</TH>"
print "                         </TR>"
for row in cursor:
        print "<TR><TD>", row[0], "</TD><TD>", row[1], "</
TD><TD>", row[2], "</TD><TD>", row[3], "</TD><TD>", row[4], "</
TD></TR>"
print "                  </TABLE>"
print "          </BODY>"
print "</HTML>"

print "Operation done successfully";
connection.close()
```

When run from the command line, this script produces the following output, which is HTML source code:

```
$ python musicians1.cgi
Content-type: text/html

<HTML>
        <HEAD>
                <TITLE>HTML Tables</TITLE>
            </HEAD>
```

```
            <BODY>
                    <H1>Musicians</H1>
                    <TABLE border=1>
                         <TR>
                                 <TH>Name</TH>
                                 <TH>Age</TH>
                                 <TH>Address</TH>
                                 <TH>Number of Records</TH>
                                 <TH>Band ID</TH>
                         </TR>
<TR><TD> 1 </TD><TD> Lyle PUENTE </TD><TD> 53 </TD><TD>
Crompond, NY </TD><TD> 6 </TD></TR>
<TR><TD> 2 </TD><TD> Tyler JOSEPH </TD><TD> 26 </TD><TD>
Columbus, OH </TD><TD> 4 </TD></TR>
<TR><TD> 3 </TD><TD> Josh DUN </TD><TD> 27 </TD><TD> Columbus,
OH </TD><TD> 3 </TD></TR>
<TR><TD> 4 </TD><TD> Sara BAREILLES </TD><TD> 35 </TD><TD>
Eureka, CA </TD><TD> 2 </TD></TR>
<TR><TD> 5 </TD><TD> Robin THICKE </TD><TD> 35 </TD><TD> Los
Angeles, CA </TD><TD> 1 </TD></TR>
<TR><TD> 6 </TD><TD> Patti ROTHBERG </TD><TD> 50 </TD><TD> New
York, NY </TD><TD> 8 </TD></TR>
            </TABLE>
            </BODY>
</HTML>
Operation done successfully
```

When uploaded as a web page, with the address http://127.0.0.1/cgi-bin/musicians1.cgi, this Python script produces the following output:

Musicians

Name	Age	Address	Number of Records	Band ID
1	Lyle PUENTE	53	Crompond, NY	6
2	Tyler JOSEPH	26	Columbus, OH	4
3	Josh DUN	27	Columbus, OH	3
4	Sara BAREILLES	35	Eureka, CA	2
5	Robin THICKE	35	Los Angeles, CA	1
6	Patti ROTHBERG	50	New York, NY	8

Operation done successfully

When writing CGI scripts, it is always a good idea to check the Apache error log file, located at `/var/log/apache2/access.log`. This will help find HTML formatting errors.

Another CGI Script: memberof1.cgi

This script does for table `testDB.MemberOf` what `musicians1.cgi` does for table `testDB.Musicians`, i.e. it displays in an HTML table the contents of `MemberOf`:

```
#!/usr/bin/python

import sqlite3

database="/Users/awise/Python/Scripts/testDB.db"
connection=sqlite3.connect(database)
cursor=connection.execute("SELECT * FROM MemberOf")
print "Content-type: text/html"
print
print "<HTML>"
print "          <HEAD>"
print "                 <TITLE>Music Bands</TITLE>"
print "          </HEAD>"
print "          <BODY>"
print "                  <H1>Music Bands</H1>"
print "                  <TABLE border=1>"
print "                         <TR>"
print "                                 <TH>ID</TH>"
print "                                 <TH>Band</TH>"
print "                                 <TH>Musician ID</TH>"
print "                                 <TH>Since</TH>"
print "                         </TR>"
for row in cursor:
        print "<TR><TD>", row[0], "</TD><TD>", row[1], "</
TD><TD>", row[2], "</TD><TD>", row[3], "</TD></TR>"
print "                  </TABLE>"
print "          </BODY>"
print "</HTML>"

print "Operation done successfully";
connection.close()
```

The HTML output when running this script from the command line is:

```
$ python memberof1.cgi
```

```
Content-type: text/html

<HTML>
        <HEAD>
                <TITLE>Music Bands</TITLE>
            </HEAD>
            <BODY>
                    <H1>Music Bands</H1>
                    <TABLE border=1>
                        <TR>
                                <TH>ID</TH>
                                <TH>Band</TH>
                                <TH>Musician ID</TH>
                                <TH>Since</TH>
                        </TR>
<TR><TD> 1 </TD><TD> My Brothers Banned </TD><TD> 1 </TD><TD>
1996 </TD></TR>
<TR><TD> 2 </TD><TD> Twenty One Pilots </TD><TD> 2 </TD><TD>
2009 </TD></TR>
            </TABLE>
            </BODY>
</HTML>
Operation done successfully
```

The browser will display the following table:

Music Bands

ID	Band	Musician ID	Since
1	My Brothers Banned	1	1996
2	Twenty One Pilots	2	2009

Operation done successfully

Database Update Using a Web Form: musicians2.html and formdata.cgi

Our next step is to provide a way for the database to be updated from a web page. We know that the database query for adding a table row, for example, in the testDB.Musicians table, is an INSERT operation. To make this simple, we would first write a Python-SQLite script that updates the database, and only then write the CGI that outputs dynamic HTML reflecting a web user-initiated update.

The Python database query file that adds a row to the table, `sqliteapi.cgi` (we could have named it `sqliteapi.py`, since it is not a CGI script, but just a Python script updating the database), looks like this:

```
#!/usr/bin/python

import sqlite3

connection=sqlite3.connect('/Users/awise/Python/Scripts/
testDB.db')
print "Opened database successfully";
connection.cursor()

musicianName='Adam LEVINE'
musicianAge=35
musicianAddress='New York, NY'
musicianRecords=5
musicianBandID=3

connection.execute("INSERT INTO Musicians(ID, Name, Age,
Address, Num_Albums, Band_ID) VALUES('%s', '%s', '%s', '%s',
'%s', '%s')" % (6, musicianName, musicianAge, musicianAddress,
musicianRecords, musicianBandID))
connection.commit()

cursor=connection.execute("SELECT * FROM Musicians")

for row in cursor:
        print "ID = ", row[0]
        print "Name = ", row[1]
        print "Age = ", row[2]
        print "Address = ", row[3]
        print "Num_Albums = ", row[4]
        print "Band_ID = ", row[5], "\n"

print "Operation done successfully";
connection.close()
```

The output from the command line will show the contents of the database updated with the new entry, while a query to the database from the sqlite3 prompt will show the same, in column mode:

```
$ python sqliteapi.cgi
Opened database successfully
```

```
ID =  1
Name =  Lyle PUENTE
Age =  53
Address =  Crompond, NY
Num_Albums =  6
Band_ID =  1

ID =  2
Name =  Tyler JOSEPH
Age =  26
Address =  Columbus, OH
Num_Albums =  4
Band_ID =  2

ID =  3
Name =  Josh DUN
Age =  27
Address =  Columbus, OH
Num_Albums =  3
Band_ID =  2

ID =  4
Name =  Sara BAREILLES
Age =  35
Address =  Eureka, CA
Num_Albums =  2
Band_ID =  None

ID =  5
Name =  Robin THICKE
Age =  35
Address =  Los Angeles, CA
Num_Albums =  1
Band_ID =  None

ID =  6
Name =  Patti ROTHBERG
Age =  50
Address =  New York, NY
Num_Albums =  8
Band_ID =  None

ID =  7
Name =  Adam LEVINE
Age =  35
Address =  New York, NY
```

```
Num_Albums =  5
Band_ID =  3
Operation done successfully
```

…And the corresponding SQLite query from the sqlite3 prompt:

```
sqlite> SELECT * FROM Musicians;
1    Lyle PUENTE          53   Crompond, NY          6    1
2    Tyler JOSEPH         26   Columbus, OH          4    2
3    Josh DUN             27   Columbus, OH          3    2
4    Sara BAREILLES       35   Eureka, CA            2
5    Robin THICKE         35   Los Angeles, CA       1
6    Patti ROTHBERG       50   New York, NY          8    None
7    Adam LEVINE          35   New York, NY          5    3
```

Once we wrote the Python script that updated the database, we want to add the capability to initiate this update from a **web form**. Our design will be a web page displaying the current contents of the `testDB.Musicians` table, as well as a web form allowing the user to enter a new musician to this table: the `musicians2.html` file. The web form will submit the data to the server via the POST method, and the data will be available for retrieval by a new script, the `formdata.cgi` file, which could (but does not) show the contents of the entire updated table: in a first phase, it just outputs the form data in a new page.

So we have 2 files:

- `musicians2.html`—shows current contents of the Musicians table and contains the web form for the new entry
- `formdata.cgi`—retrieves the data from the web form and outputs it in a dynamic HTML page

The `musicians2.html` file looks like this:

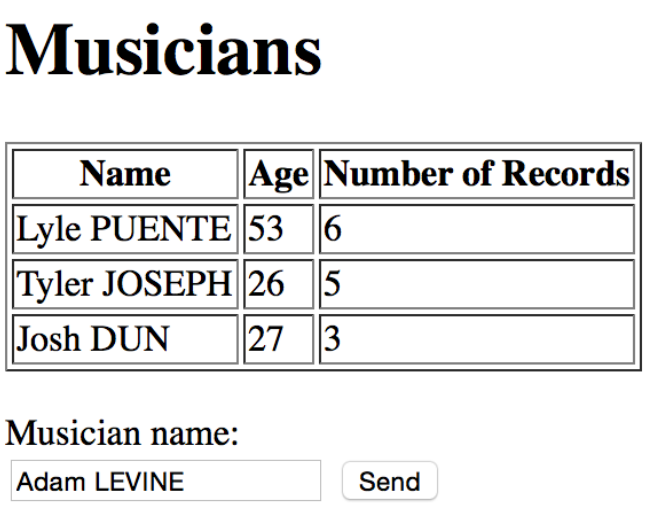
```
$ vi musicians2.html
<!DOCTYPE html>
<HTML>
        <SCRIPT TYPE="text/javascript">
        <!--
        function InputHandler(inputname)
        {
                x=document.musicianform.inputname.value;
                document.musicianform.submit(x);
        }
```

```
        -->
        </SCRIPT>
        <HEAD>
                <TITLE>HTML Tables</TITLE>
        </HEAD>
        <BODY>
                <H1>Musicians</H1>
                <TABLE border=1>
                        <TR>
                                <TH>Name</TH>
                                <TH>Age</TH>
                                <TH>Number of Records</TH>
                        </TR>
                        <TR>
                                <TD>Lyle PUENTE</TD>
                                <TD>53</TD>
                                <TD>6</TD>
                        </TR>
                        <TR>
                                <TD>Tyler JOSEPH</TD>
                                <TD>26</TD>
                                <TD>5</TD>
                        </TR>
                        <TR>
                                <TD>Josh DUN</TD>
                                <TD>27</TD>
                                <TD>3</TD>
                        </TR>
                </TABLE>
                <BR>
                <LABEL for='musicianform'>Musician name: </
LABEL>
                <FORM
                        name="musicianform"
                        method="POST"
                        action="http://127.0.0.1/cgi-bin/
formdata.cgi">
                <INPUT
                        name="inputname"
                        type="text"
                        value="inputName"

onsubmit="javascript:InputHandler('inputname');"
                />
                <INPUT type="submit" value="Send">
                </FORM>
```

```
        </BODY>
</HTML>
```

The corresponding web page looks like this (notice the form):



Musicians

Name	Age	Number of Records
Lyle PUENTE	53	6
Tyler JOSEPH	26	5
Josh DUN	27	3

Musician name:

Adam LEVINE Send

The data from the web form is submitted to the server by calling a JavaScript function. The JavaScript code is invoked by the web form, and implemented as a script within the HTML page (at the beginning).

Once the form is filled out and the "Send" button clicked, the HTML page redirects to a CGI script (file `formdata.cgi`), which in the simplest form just retrieves the form data and displays in as dynamic HTML.

The formdata.cgi script looks like this:

```
#!/usr/bin/python

import cgi
import cgitb; cgitb.enable()     # for troubleshooting

formData=cgi.FieldStorage()
musicianName=formData.getvalue('inputname')
#musicianName="Adriana Wise"

print "Content-type: text/html"
print
print "<HTML>"
print "          <HEAD>"
```

```
print "                <TITLE>HTML Tables</TITLE>"
print "          </HEAD>"
print "          <BODY>"
print "          <P>", musicianName, "</P>"
print "          </BODY>"
print "</HTML>"
```

The output as a web page is:

One Little Step at a Time: DB Update from an HTML Form

The next step is to upgrade the same to a multiple text box form and reflect the changes in the target CGI page. Our new files are:

- musicians3.html
- formdata2.cgi

```
<!DOCTYPE html>
<HTML>
        <SCRIPT TYPE="text/javascript">
        <!--
        function InputHandler()
        {
                //
inputvalue=document.musicianform.inputname.value;
                //document.musicianform.submit(inputvalue);
                v1=document.musicianform.musicianname.value;
                document.musicianform.submit(v1);
                v2=document.musicianform.musicianage.value;
                document.musicianform.submit(v2);
                v3=document.musicianform.musicianrecords.value;
                document.musicianform.submit(v3);
        }
        -->
        </SCRIPT>
        <HEAD>
```

```
                <TITLE>HTML Tables</TITLE>
        </HEAD>
        <BODY>
                <H1>Musicians</H1>
                <TABLE border=1>
                        <TR>
                                <TH>Name</TH>
                                <TH>Age</TH>
                                <TH>Number of Records</TH>
                        </TR>
                        <TR>
                                <TD>Lyle PUENTE</TD>
                                <TD>53</TD>
                                <TD>6</TD>
                        </TR>
                        <TR>
                                <TD>Tyler JOSEPH</TD>
                                <TD>26</TD>
                                <TD>5</TD>
                        </TR>
                        <TR>
                                <TD>Josh DUN</TD>
                                <TD>27</TD>
                                <TD>3</TD>
                        </TR>
                </TABLE>
                <BR>
                <FORM
                        name="musicianform"
                        method="POST"
                        action="http://127.0.0.1/cgi-bin/
formdata2.cgi"
                >
                <LABEL for='musicianname'>Musician name: </
LABEL>
                <INPUT
                        name="musicianname"
                        type="text"
                />
                <BR>
                <LABEL for='musicianage'>Musician age: </LABEL>
                <INPUT
                        name="musicianage"
                        type="text"
                />
                <BR>
```

```
                <LABEL for='musicianrecords'>Musician records:
</LABEL>
                <INPUT
                        name="musicianrecords"
                        type="text"
                />
                <INPUT
                        type="submit"
                        value="Update DB"
                />
                </FORM>
        </BODY>
</HTML>
```

As a web page, this looks like this:

Musicians

Name	Age	Number of Records
Lyle PUENTE	53	6
Tyler JOSEPH	26	5
Josh DUN	27	3

Musician name:
Musician age:
Musician records: Update DB

The new version of the target CGI is:

```
#!/usr/bin/python

import cgi
import cgitb; cgitb.enable()     # for troubleshooting

formData=cgi.FieldStorage()
musicianName=formData.getvalue('musicianname')
musicianAge=formData.getvalue('musicianage')
musicianRecords=formData.getvalue('musicianrecords')
#musicianName="Adriana Wise"
```

```
print "Content-type: text/html"
print
print "<HTML>"
print "          <HEAD>"
print "                  <TITLE>HTML Tables</TITLE>"
print "          </HEAD>"
print "          <BODY>"
print "          <P>", musicianName, "</P>"
print "          <P>", musicianAge, "</P>"
print "          <P>", musicianRecords, "</P>"
print "          </BODY>"
print "</HTML>"
```

And the resulting web page looks like this:

In the next version of editing this pair of files, we want to not only capture the data input from the HTML form, but to also use the resulting data values from the CGI and initiate a database query, in this case to insert a new musician record into the database. The two new files are:

- musicians4.html
- formdata3.cgi

```
<!DOCTYPE html>
<HTML>
        <SCRIPT TYPE="text/javascript">
        <!--
        function InputHandler()
        {
                //
inputvalue=document.musicianform.inputname.value;
                //document.musicianform.submit(inputvalue);
                v1=document.musicianform.musicianname.value;
                document.musicianform.submit(v1);
                v2=document.musicianform.musicianage.value;
                document.musicianform.submit(v2);
                v3=document.musicianform.musicianaddress.value;
                document.musicianform.submit(v3);
                v4=document.musicianform.musicianrecords.value;
                document.musicianform.submit(v4);
                v5=document.musicianform.musicianbandID.value;
```

```
        document.musicianform.submit(v5);
}
-->
</SCRIPT>
<HEAD>
        <TITLE>HTML Tables</TITLE>
</HEAD>
<BODY>
        <H1>Musicians</H1>
        <TABLE border=1>
                <TR>
                        <TH>Name</TH>
                        <TH>Age</TH>
                        <TH>Address</TH>
                        <TH>Number of Records</TH>
                        <TH>Band ID</TH>
                </TR>
                <TR>
                        <TD>Lyle PUENTE</TD>
                        <TD>53</TD>
                        <TD>Crompond, NY</TD>
                        <TD>6</TD>
                        <TD>1</TD>
                </TR>
                <TR>
                        <TD>Tyler JOSEPH</TD>
                        <TD>26</TD>
                        <TD>Columbus, OH</TD>
                        <TD>5</TD>
                        <TD>2</TD>
                </TR>
                <TR>
                        <TD>Josh DUN</TD>
                        <TD>27</TD>
                        <TD>Columbus, OH</TD>
                        <TD>3</TD>
                        <TD>2</TD>
                </TR>
                <TR>
                        <TD>Sara BAREILLES</TD>
                        <TD>35</TD>
                        <TD>Eureka, CA</TD>
                        <TD>3</TD>
                        <TD></TD>
                </TR>
                <TR>
```

```
                                    <TD>Robin THICKE</TD>
                                    <TD>35</TD>
                                    <TD>Los Angeles, CA</TD>
                                    <TD>1</TD>
                                    <TD></TD>
                                </TR>
                        </TABLE>
                        <BR>
                    <FORM
                                name="musicianform"
                                method="POST"
                                action="http://127.0.0.1/cgi-bin/
formdata3.cgi"
                        >
                        <LABEL for='musicianname'>Musician name: </
LABEL>
                        <INPUT
                                name="musicianname"
                                type="text"
                        />
                        <BR>
                        <LABEL for='musicianage'>Musician age: </LABEL>
                        <INPUT
                                name="musicianage"
                                type="text"
                        />
                        <BR>
                        <LABEL for='musicianaddress'>Musician address:
</LABEL>
                        <INPUT
                                name="musicianaddress"
                                type="text"
                        />
                        <BR>
                        <LABEL for='musicianrecords'>Musician records:
</LABEL>
                        <INPUT
                                name="musicianrecords"
                                type="text"
                        />
                        <BR>
                        <LABEL for='musicianbandID'>Musician band ID: </
LABEL>
                        <INPUT
                                name="musicianbandID"
                                type="text"
```

```
                />
                <INPUT
                        type="submit"
                        value="Update DB"
                />
                </FORM>
        </BODY>
</HTML>
```

The web page with the new form:

Musicians

Name	Age	Address	Number of Records	Band ID
Lyle PUENTE	53	Crompond, NY	6	1
Tyler JOSEPH	26	Columbus, OH	5	2
Josh DUN	27	Columbus, OH	3	2
Sara BAREILLES	35	Eureka, CA	3	
Robin THICKE	35	Los Angeles, CA	1	

Musician name:
Musician age:
Musician address:
Musician records:
Musician band ID: Update DB

The new target CGI is `formdata4.cgi`:

```
#!/usr/bin/python

import cgi
import cgitb; cgitb.enable()     # for troubleshooting
import sqlite3

formData=cgi.FieldStorage()
musicianName=formData.getvalue('musicianname')
musicianAge=formData.getvalue('musicianage')
musicianAddress=formData.getvalue('musicianaddress')
musicianRecords=formData.getvalue('musicianrecords')
musicianBandID=formData.getvalue('musicianbandID')

print "Content-type: text/html"
```

```
print
print "<HTML>"
print "          <HEAD>"
print "                 <TITLE>HTML Tables</TITLE>"
print "          </HEAD>"
print "          <BODY>"
print "          <P>", musicianName, "</P>"
print "          <P>", musicianAge, "</P>"
print "          <P>", musicianAddress, "</P>"
print "          <P>", musicianRecords, "</P>"
print "          <P>", musicianBandID, "</P>"
print "          </BODY>"
print "</HTML>"

connection=sqlite3.connect('/Users/awise/Python/Scripts/
testDB.db')
print "Opened database successfully";

#connection.execute("UPDATE Musicians SET ID=6 Name='Adam
LEVINE' Age='35' Address='New York, NY', Num_Albums='5',
Band_ID='3'")
sql="""" INSERT INTO Musicians (ID, Name, Age, Address,
Num_Albums, Band_ID) VALUES('%s', '%s', '%s', '%s', '%s', '%s')
""" % (6, musicianName, musicianAge, musicianAddress,
musicianRecords, musicianBandID)
connection.execute(sql)
#connection.execute("INSERT INTO Musicians(ID, Name, Age,
Address, Num_Albums, Band_ID) VALUES(%s, %s, %s, %s, %s, %s)" %
(6, musicianName, musicianAge, musicianAddress, musicianRecords,
musicianBandID))
connection.commit()
print "Total number of rows updated :", connection.total_changes
cursor=connection.execute("SELECT * FROM Musicians")
for row in cursor:
        print "ID = ", row[0]
        print "Name = ", row[1]
        print "Age = ", row[2]
        print "Address = ", row[3]
        print "Num_Albums = ", row[4]
        print "Band_ID = ", row[5], "\n"

print "Operation done successfully";
connection.close()
```

Assuming we want to re-test inserting the record Adam Levine into the database, we would get a **uniqueness constraint error**:

Musicians

Name	Age	Address	Number of Records	Band ID
Lyle PUENTE	53	Crompond, NY	6	1
Tyler JOSEPH	26	Columbus, OH	5	2
Josh DUN	27	Columbus, OH	3	2
Sara BAREILLES	35	Eureka, CA	3	
Robin THICKE	35	Los Angeles, CA	1	

Musician name: Adam LEVINE
Musician age: 35
Musician address: New York, NY
Musician records: 8
Musician band ID: 3 Update DB

Adam LEVINE

35

New York, NY

8

3

Opened database successfully

<class 'sqlite3.IntegrityError'>

A problem occurred in a Python script. Here is the sequence of function calls le

/Library/WebServer/CGI-Executables/formdata3.cgi in ()

```
    32 #connection.execute("UPDATE Musicians SET ID=6 Name='Adam
    33 sql=""" INSERT INTO Musicians (ID, Name, Age, Address, Nur
=>  34 connection.execute(sql)
    35 #connection.execute("INSERT INTO Musicians(ID, Name, Age,
    36 connection.commit
```

connection = <sqlite3.Connection object>, connection.**execute** = <built-in method execute of sql

<class 'sqlite3.IntegrityError'>: UNIQUE constraint failed: Musicians.ID
args = ('UNIQUE constraint failed: Musicians.ID',)
message = 'UNIQUE constraint failed: Musicians.ID'

Instead, let's insert a new artist name, Alecia Beth Moore (a.k.a. Pink):

Musicians

Name	Age	Address	Number of Records	Band ID
Lyle PUENTE	53	Crompond, NY	6	1
Tyler JOSEPH	26	Columbus, OH	5	2
Josh DUN	27	Columbus, OH	3	2
Sara BAREILLES	35	Eureka, CA	3	
Robin THICKE	35	Los Angeles, CA	1	

Musician name: Alecia MOORE
Musician age: 36
Musician address: Doylestown, PA
Musician records: 4
Musician band ID: Update DB

Alecia MOORE

36

Doylestown, PA

4

None

Opened database successfully Total number of rows updated : 2 ID = 1 Name = Lyle PUENTE Age = 53 Address = Crompond, NY Num_Albums = 6 Band_ID = 1 ID = 2 Name = Tyler JOSEPH Age = 26 Address = Columbus, OH Num_Albums = 4 Band_ID = 2 ID = 3 Name = Josh DUN Age = 27 Address = Columbus, OH Num_Albums = 3 Band_ID = 2 ID = 4 Name = Sara BAREILLES Age = 35 Address = Eureka, CA Num_Albums = 2 Band_ID = None ID = 5 Name = Robin THICKE Age = 35 Address = Los Angeles, CA Num_Albums = 1 Band_ID = None ID = 6 Name = Patti ROTHBERG Age = 50 Address = New York, NY Num_Albums = 8 Band_ID = None ID = 7 Name = Adam LEVINE Age = 35 Address = New York, NY Num_Albums = 5 Band_ID = 3 ID = 8 Name = Alecia MOORE Age = 36 Address = Doylestown, PA Num_Albums = 4 Band_ID = None Operation done successfully

This is just plain text, displayed from within an HTML page, but the output reflects the newly updated contents of the database. The next step from here would be to re-do the CGI script that updates the database so it will now include HTML table formatting for the DB table information.

The new target CGI is `formdata5.cgi`:

```
import cgitb; cgitb.enable()     # for troubleshooting
import sqlite3

formData=cgi.FieldStorage()
musicianName=formData.getvalue('musicianname')
musicianAge=formData.getvalue('musicianage')
musicianAddress=formData.getvalue('musicianaddress')
musicianRecords=formData.getvalue('musicianrecords')
```

```
musicianBandID=formData.getvalue('musicianbandID')

print "Content-type: text/html"
print
print "<HTML>"
print "         <HEAD>"
print "                <TITLE>HTML Tables</TITLE>"
print "         </HEAD>"
print "         <BODY>"
print "         <P>", musicianName, "</P>"
print "         <P>", musicianAge, "</P>"
print "         <P>", musicianAddress, "</P>"
print "         <P>", musicianRecords, "</P>"
print "         <P>", musicianBandID, "</P>"
print "         </BODY>"
print "</HTML>"

connection=sqlite3.connect('/Users/awise/Python/Scripts/
testDB.db')
print "Opened database successfully";
connection.cursor()

connection.execute("INSERT INTO Musicians(ID, Name, Age,
Address, Num_Albums, Band_ID) VALUES('%s', '%s', '%s', '%s',
'%s', '%s')" % (8, musicianName, musicianAge, musicianAddress,
musicianRecords, musicianBandID))
connection.commit()
print "Total number of rows updated :", connection.total_changes

cursor=connection.execute("SELECT * FROM Musicians")
connection.commit()
print '                  <TABLE border="1" cellpadding="5"
cellspacing="5" bordercolor="green" bgcolor="orange">'
print "                           <TR>"
print "                                  <TH>Name</TH>"
print "                                  <TH>Age</TH>"
print "                                  <TH>Address</TH>"
print "                                  <TH>Number of Records</
TH>"
print "                                  <TH>Band ID</TH>"
print "                           </TR>"
for row in cursor:
        print "                  <TR>"
        print "                          <TD>", row[1], "</TD>"
        print "                          <TD>", row[2], "</TD>"
        print "                          <TD>", row[3], "</TD>"
```

```
        print "                                <TD>", row[4], "</TD>"
        print "                                <TD>", row[5], "</TD>"
        print "                        </TR>"
print "                    </TABLE>"
for row in cursor:
        print "ID = ", row[0]
        print "Name = ", row[1]
        print "Age = ", row[2]
        print "Address = ", row[3]
        print "Num_Albums = ", row[4]
        print "Band_ID = ", row[5], "\n"

print "Operation done successfully";
connection.close()
```

The web page generated by `formdata5.cgi` is:

Alecia MOORE

36

Doylestown, PA

4

None

Opened database successfully Total number of rows updated : 2

Name	Age	Address	Number of Records	Band ID
Lyle PUENTE	53	Crompond, NY	6	1
Tyler JOSEPH	26	Columbus, OH	4	2
Josh DUN	27	Columbus, OH	3	2
Sara BAREILLES	35	Eureka, CA	2	None
Robin THICKE	35	Los Angeles, CA	1	None
Patti ROTHBERG	50	New York, NY	8	None
Adam LEVINE	35	New York, NY	5	3
Alecia MOORE	36	Doylestown, PA	4	None

Operation done successfully

One Little Step at a Time: DB Update from a CGI-Generated HTML Form

Our next step is to achieve the same DB update, but this time departing from a CGI script, outputting dynamic HTML for the musicians table from our `testDB.musicians` database. To make things more interesting, the SQL transaction we want to commit will be an update, rather than an insert, by highlighting as hyperlinks all the musicians' names, and redirecting on click to a CGI script that captures the musician's name and offers a database edit.

Again, we need two files:

- `musicians6.cgi`—to output dynamic HTML consisting of the current database entries
- `musicianupdate.cgi`—to capture the hyperlink form data and display it onto a new web page; later to use it to issue a new query to the database

At this point, the database has been updated with the Alecia MOORE entry. The old version of the musicians CGI file, `musicians1.cgi`, outputs the following content:

Musicians

Name	Age	Address	Number of Records	Band ID
1	Lyle PUENTE	53	Crompond, NY	6
2	Tyler JOSEPH	26	Columbus, OH	4
3	Josh DUN	27	Columbus, OH	3
4	Sara BAREILLES	35	Eureka, CA	2
5	Robin THICKE	35	Los Angeles, CA	1
6	Patti ROTHBERG	50	New York, NY	8
7	Adam LEVINE	35	New York, NY	5
8	Alecia MOORE	36	Doylestown, PA	4

Operation done successfully

Our modification to the original `musicians.cgi` file creates hyperlinks for the musician names and attaches forms to them, in order to retrieve information and forward it to the HTTP server and the target CGI. In our simplest implementation, we just display the form values captured from the dynamic HTML form in the new CGI script.

After this step, that value can be used to query the database, for instance, to do a record update on that musician.

The `musicians6.cgi` file looks like this:

```
#!/usr/bin/python

import sqlite3
import cgi

database="/Users/awise/Python/Scripts/testDB.db"
connection=sqlite3.connect(database)
cursor=connection.execute("SELECT * FROM musicians")
print "Content-type: text/html"
print
print "<HTML>"
print "         <HEAD>"
print "                 <TITLE>HTML Tables</TITLE>"
print '                 <SCRIPT language="JavaScript"
type="text/javascript">'
print "                 <!--"
print "                 function getmusician(selectedname)"
print "                 {"
print "
document.musicianform.musicianinput.value=selectedname;"
print "
document.musicianform.submit();"
#print "
document.forms[musicianform].value=selectedname;"
#print "
document.forms[musicianform].submit(selectedname);"
print "                 }"
print "                 -->"
print "                 </SCRIPT>"
print "         </HEAD>"
print "         <BODY>"
print "                 <H1>Musicians</H1>"
print '                 <TABLE border="1" cellpadding="5"
cellspacing="5" bordercolor="green" bgcolor="orange">'
print "                         <TR>"
print "                                 <TH>Name</TH>"
print "                                 <TH>Age</TH>"
print "                                 <TH>Address</TH>"
print "                                 <TH>Number of Records</
TH>"
```

```
print "                                    <TH>Band ID</TH>"
print "                            </TR>”
print "                             <FORM"
print '                                     name="musicianform"'
print '                                     method="POST"'
print '                                     action="http://
127.0.0.1/cgi-bin/musicianupdate.cgi"'
print '                                     target="_self">'
print '                             <INPUT”
print '                                     type="hidden"
print '                                     name="musicianinput">'
i=0
for row in cursor:
        print "                     <TR>"
        print "                             <TD>"
        print """                                    <A
href="javascript:getmusician('%s')">%s</A>""" % (row[1], row[1])
        #print """                                   <A href="#"
onclick="javascript:document.forms['musicianform'].submit();">
%s</A>""" % row[1]
        print "                             </TD>"
        print "                             <TD>", row[2], "</TD>"
        print "                             <TD>", row[3], "</TD>"
        print "                             <TD>", row[4], "</TD>"
        print "                             <TD>", row[5], "</TD>"
        print "                     </TR>"
        i=i+1
print "                             </INPUT>"
print "                             </FORM>"
print "                    </TABLE>"
print "          </BODY>"
print "</HTML>"

print "Operation done successfully";
connection.close()
```

The associated CGI (the target `musicianupdate.cgi`) just outputs the form name in an HTML page:

```
#!/usr/bin/python

import cgi
import cgitb; cgitb.enable()     # for troubleshooting
import sqlite3
```

```
formData=cgi.FieldStorage()
musicianName=formData.getvalue('musicianinput')
#print "%s" % musicianName

print "Content-type: text/html"
print
print "<HTML>"
print "          <HEAD>"
print "                  <TITLE>Musician Update</TITLE>"
print "          </HEAD>"
print "          <BODY>"
print "                   <H2>%s</H2>" % musicianName
print "          </BODY>"
print "</HTML>"
```

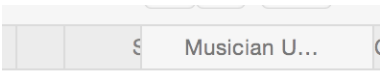
The web pages will look like this:

Musicians

Name	Age	Address	Number of Records	Band ID
Lyle PUENTE	53	Crompond, NY	6	1
Tyler JOSEPH	26	Columbus, OH	4	2
Josh DUN	27	Columbus, OH	3	2
Sara BAREILLES	35	Eureka, CA	2	None
Robin THICKE	35	Los Angeles, CA	1	None
Patti ROTHBERG	50	New York, NY	8	None
Adam LEVINE	35	New York, NY	5	3
Alecia MOORE	36	Doylestown, PA	4	None

Operation done successfully

…with the first hyperlink pointing to:



Lyle PUENTE