Chapter 14 — Relational Databases using the DB-API 2.0

Introduction

The Python DB-API 2.0 is a standard way to access databases in your programs. Because of this common way to access data, only minor implementation specific changes would need to be made to your Python code if you were to change the underlying database. The details o the specirfications can be found in document PEP249.

This introductory database access chapter will use the SQLite3 database,. It is built in to most Python implementations and does not require additional setup. SQLite 3 is a self contained, server-less database that is perfect for low to medium volume websites, data analysis, and application data storage.

There is a section at the end of this chapter where a few examples using MySQL are shown. The MySQL database is appropriate for significant multi-user and high volume applications. It required the setup and management of a database server, that is beyond the context of this introduction.

Objectives

Upon completion of this chapter's exercises, you should be able to:

- Blah de blah.
- Baz and Barf.

Prerequisites

This Chapter requires...

A Few Definitions

Connection — A connection is an object that manages the physical communication with the database. A program may have several connections to several different databases. A connection also manages transactions at the database level.

Cursor — A cursor is an object, created from a connection, that manages the results of SQL statements.

Executing Statements

To execute SQL statements, you must 1) include the database module, 2) open a connection to your database and 3) create a cursor to contain the statement as it executes.

This example creates a new database file called "myfile.sqlite3", creates a new table to store books, and inserts a single row into that table. There is a single cursor and execute method on the cursor is called multiple times. The `.rowcount` property is printed out and displays a -1 signifying that create was successful and ones for the inserts to represent that a single row was added.

```
1| import sqlite3
2|
3| db = sqlite3.connect('myfile.sqlite3')
4|
5| c = db.cursor()
6|
7| c.execute('CREATE TABLE books (id INTEGER PRIMARY KEY, title
     TEXT, author TEXT);')
8| print(c.rowcount)
9|
10| c.execute('INSERT INTO books (id, title, author) VALUES
     (?,?,?);', [42, 'Moby Dick', 'Herman Melville'])
11| print(c.rowcount)
12|
13|
14| c.execute('INSERT INTO books (id, title, author) VALUES
     (:id, :title, :author);', {'author':'Douglass Adams',
     'title':'Restaurant at the End of the Universe', 'id':99})
15| print(c.rowcount)
16|
17| db.commit()
18| db.close()
```

```
-1
1
1
```

You will notice that the insert statements use placeholders (? and :name) to hold the data values. These placeholders are replaced with the actual data, passed in the second argument as a list or a dictionary. This insures that values are properly quoted and that special characters are automatically escaped. Using placeholders in your SQL statements make your program much more secure and easier to write.

If you have made changes to data, YOU MUST COMMIT the changes or they will not actually be updated to the database. You will see this being done before the close. Notice that this is not a cursor action, but an action on the connection.

Executing Statements that Return a Single Row

accessing by numeric location in select

```
 1| import sqlite3
 2|
 3| db = sqlite3.connect("myfile.sqlite3")
 4|
 5| c = db.cursor()
 6| c.execute('SELECT * FROM books WHERE id = ?;', [99])
 7| row = c.fetchone()
 8| if row:
 9|     print(row)
10|     print('book', row[0], 'is', row[1])
11| else:
12|     print('no data was returned')
13| db.close()
```

```
(99, 'Restaurant at the End of the Universe', 'Douglass Adams')
book 99 is Restaurant at the End of the Universe
```

Executing Statements that Return Many Rows

Two ways

iterate over the connector

Getting Attributes by Name

using a row factory

Executing "Multi" Statements

Using MySQL

foo

Summary

Goes here

Important Terms

here

Exercises

Here

Word Search

References