Lab 3 Handout: Objects and SQLite Databases

Due: 11 November 2019

1 Objectives

- Reuse and adapt the code you developed in Lab2 such that it could interact with SQLite databases.
- Use the sqlite3 module, a Python interface to SQLite databases, to permanently store information.

2 Task description

Instructor - Course - Student - SQLite.

This lab is a follow-up to Lab 2, where you created three classes: Instructor, Course and Student.

A student derived from `Student` class can select or drop courses. An instructor derived from `Instructor` class can teach courses. A course derived from `Course` class can enroll or remove students.

In Lab 2, all data are stored in memory. After the program finishes, the data are gone. In this lab, we will store them permanently in a database, which we can view, edit or analyze later. To this end, we use the `sqlite3` library in Python, which includes methods for interacting with SQLite databases. SQLite is a lightweight, portable relational database management system.

We import the `sqlite3` library in Python in the following way: `import sqlite3 as sqlite`.

Refer to this SQLite Tutorial to get started. In particular, SQL statements such as INSERT, INSERT OR REPLACE, INSERT OR IGNORE, DELETE, SELECT and JOIN are useful for this lab. Refer to The Python Standard Library for more information about the `sqlite3` module.

You decide what tables and table columns to include in your SQLite database. Make sure that creating an object from a class adds a record to one of these tables, and calling a class method (such as `select`, `drop`, `enroll`, `remove`, and `teach`) updates a table.

Define a special method, `__str__()`, for each class to retrieve information from relevant database tables and return that information as a string. For example, let `s` be an object of `Student` class. `print(s)` displays the student name (student number) followed by a list of courses he has selected. Each course contains its course number, course name, the semester in which this course is offered, and the instructor's name. Let `c` be an object of `Course` class. `print(c)` displays the course number, course name, semester and instructor's name followed by a list of enrolled students. Let `t` be an object of `Instructor` class. `print(t)` displays the instructor's name, followed by a list courses under his name. Include course number, course name, semester, and time and location for each course when calling `print(t)`.

You should prevent the database tables from adding duplicate records.

You can manually create a database and then a few empty tables with DB Browser for SQLite.

The starter code for this lab can be downloaded from our course homepage.

3 Requirements

- Do the lab in a group with a maximum of 2 students.
- Make sure the commented-out statements below `if __name__ == '__main__'` in the provided starter code produce reasonable results.
- I only accept object-oriented implementation.
- The partial work is due on October 28 (one for each group). Please submit your source code, Lab3.py, to LRR.

- The completed work is due on November 11. Submit the following files to LRR: source code (Lab3.py) and the portable database (`school.sqlite3`). It is important to work on your unfinished lab work as I may check your progress and evaluate evidence of efforts based on that.

- Prefix `YourFullNames-YourStudentNumbers-` to all your file names.