CMPSC 312
Database Systems
Fall 2020

Lab 5 Assignment:
Coding Abstraction for Interaction with SQlite
Submit deliverables through your assignment GitHib repository.
Place the builder and Python code in `src/`, and the report in the `writing` directories

Objectives

To learn how to write code in Python to automate database management. Your program will ask for user input and then will create the correct commands to pass to the `sqlite3` library for making insertions and table edits. Your working database will be your choice to make. Sample Python code to help you get started is provided.

GitHub Starter Link

https://classroom.github.com/a/lRKi6qd3

To use this link, please follow the steps below.

- Click on the link and accept the assignment
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab,
- Clone this repository (bearing your name) and work locally
- As you are working on your lab, you are to commit and push regularly. The commands are the following.
  - `git add -A`
  - `git commit -m ‘‘Your notes about commit here’’`
  - `git push`

Handed out: 14^{th} Oct 2020

Introduction

The term, *abstraction* is commonly used in computer science to describe the simplicity created by a program that does something very complicated. This simplicity may result from a software that is able automate complicated tasks by automatically writing and inputting robust instructions for users to accomplish tasks quickly and with little effort on their part. Such software is commonly used in databases to allow users to interact with large databases without having to know how to write SQL code.

When working with databases, it is common to see abstraction helping users to work with data in areas of; inserting, updating, deleting and retrieving information. To simplify this work, users interface with user-friendly software that connects to the database to complete the *heavy-lifting* work for them. Unknown to the user, all the commands that are received by the software are translated into the language of the database and the work is completed without the user ever having to touch the core programming code of the database.

In previous labs, we have studied how to build advanced databases where primary and foreign keys were programmed into the schema. We also learned how to populate these databases with data. More recently, we spent some time in class to write Python code to connect to a SQlite3 database to make automatic queries and insertions. In this lab, you will demonstrate abstraction by creating a simple database, which will be controlled by a simple Python manager program. Your Python program will translate user commands into SQL code to facilitate work with the database for the users. For this, you will have to use a menu-driven system to interact with the user. Demo code is provided with this assignment to give some ideas about how such a menu system might work in Python.

Tasks

1. Create a database having at least three (3) tables using a `build-file`.
   (a) Choose the database's function: The database you create can manage any data you choose (i.e., sports, films, music, etc).
   (b) Each table must have primary keys to connect the three tables. Foreign keys are suggested but not absolutely necessary for this assignment.
   (c) Your database must be created by a `build-file`.
   (d) Please provide several rows of data in each table using your `build-file`.
   (e) Your menu-driven Python code will allow the user to further populate the database's tables and to complete modifications and queries.
2. Write a simple Python program to interact with the database using a menu system for the user. Note, there is is the file: `demoCode.py` included with this lab to give you ideas on how to complete the menu system in Python. The main functions of your manager program are listed below.
   (a) **Query all tables in the database**: Allow the user to run a `SELECT * FROM ...` query of tables the database. Note: Please check class notes for code to get you started.

(b) **Selected table queries**: Allow the user to run a specific query over a table in the database such as, `SELECT * FROM myTable`. For this, you would have to present a list of all tables in the database and allow the user to choose one.

- As an added challenge, present the user with a list of tables in the database and then ask which table to query for **selected attributes**.

(c) **Insertions**: Ask the user for data and a table into which the data is inserted.

(d) **Alterations**: Ask the user for a table name for a data alteration. In specific, ask the user to provide the name of a table, two attributes, and a string or value. One attribute is to be used to make a conditional statement in the query of the selected table, and the other attribute is to be changed to the provided value or string.

1 Summary of the Required Deliverables

1. This is an open-ended lab and you are free to choose which ever solution you like to accomplish the above goals which are written in blue.
2. File `src/dbManager.py`: a program to interact with your database and to complete the tasks mentioned above
3. File `src/dbBuilder.txt`: a builder file for your own database. This database can have any function and contain any data. IT must have at least three (3) tables.
4. File `writing/report.md`: A markdown file containing an introduction to your databas, a description of how your code works and some copy-and-pasted output from the running the code.

In adherence to the Honor Code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. Deliverables that are nearly identical to the work of others will be taken as evidence of violating Allegheny College's Honor Code.