Programming Assignment 2

Brandeis University, COSI 127b, Spring 2019

Instructor: Ryan Marcus

For this assignment, you'll be helping a library at Old University modernize their data management workflow. Old University's library system is currently quite a mess: it is stored as a single CSV (comma separated values) file, `data.csv`, where each row represents a book loan. Here's a single row of that file, formatted for readability:

```
Bennett Daniel,                (patron name)
3521007764925597111961645,    (patron card number)
1973,                          (patron join year)
(377) 750-0899,                (patron phone number)
7850851362293096790613312,    (book barcode)
Think Complexity,              (book title)
1729,                          (book year)
"Downey, Allen",               (book author)
1430,                          (author birth year)
Routledge,                     (publisher)
+1-(476)-431-0927,             (publisher phone)
2458153.5,                     (checkout date)
2458161.5,                     (due date)
1                              (returned)
```

This row represents a patron, Bennett Daniel, who joined the library in 1973, checking out a book called Think Complexity by Allen Downey, published by Routledge, on the given date (Julian day). The `returned` column encodes whether or not a book as been returned or not. A `1` is used for returned books, and a `0` used for books that are still out (have not yet been returned).

We know from our knowledge of schema design that this is *not* a good schema for the library! Luckily, we are going to help Old University out and make them a brand-new library system.

Part 1: Schema Design

For the first part of this assignment, you'll design a schema to support the library. You'll do this in two different ways:

- Using an ER diagram. Carefully identify (with prose) the entities and the relationships between them, looking at the data to validate your assumptions. Then, draw (with a computer) an ER diagram representing your proposed schema.
- Using functional dependencies. Carefully enumerate the functional dependencies that exist within the library's data, and explain what each entails.

Then, decompose the schema into a series of relations that are in third normal form (3NF).

Compile both of these designs into a single PDF file, and include it with your submission.

This part of your assignment will be worth 50% of the grade for PA2. We will grade this part of the assignment based on:

- An sensical ER diagram that follows the conventions introduced in class, and is neatly drawn (with a computer).
- A reasonable explaination of the decisions made in the ER diagram
- A correctly denormalized schema in 3NF
- A reasonable explaination of each functional dependency

Part 2: Computer System

Once you've designed a suitable schema, you must build Old University a system to (1) load their existing data into an SQLite-managed database, and (2) help the library execute queries against that database.

You must complete this part of the assignment using Python, and you must write your code using Python 3 (version 3.5 or above). We have provided a file, `main.py`, which contains a skeleton of the functions you need to write, in addition to an interactive menu system. Each function you need to implement starts with a comment explaining what the function should do.

This part of the assignment will be worth 50% of the grade for PA2. This part of the assignment will be graded based on:

- The readability of your Python code
- A correct implementation of the schema you designed
- Correctly using the SQLite Python API (e.g., using `?` in queries)
- Correctly loading the data into your database
- Generating reports that are *neat and easy to read*

Please submit your database file (`library.db`), any Python source files (at least `main.py`), and a PDF containing your schema design (`schema.pdf`) to LATTE. You may ZIP, tar, or otherwise compress them if you wish.