Intermediate Python **Homework 3**
Due: 11:59 PM Sunday, November 22, 2020 **This is an individual assignment.**

1. Download the file 'JustLeeBooks.txt'. This file contains SQL statements, one per line. For readability, the file contains blank lines. Open the file and read the commands; if the line is not blank (len > 0), then append it to a list of str values named ***commands***. Display commands. Note: all table and column names should be in upper case.

2. Open a sqlite3 database named 'LeeBooks.db' and get its cursor. Then write a for loop to execute each command in commands to create the database. Then do connection.commit(). Note: if you have to re-run these commands, manually delete LeeBooks.db and start from scratch.

3. Create the following display to ensure that the tables were created correctly. To get table names, use the query `'SELECT name FROM sqlite_master WHERE type="table"'`; then fetch the returned rows; each row's entry [0] contains the table name. To get column names, use the query `'PRAGMA table_info(' + row[0] + ')'` where row[0] is the table name; then fetch the returned rows; each row's entry [1] contains the column name for that table. Hint: this requires nested for loops.

```
Table: EMPLOYEES
Columns: |  EMPNO | LNAME | FNAME | JOB | HIREDATE | DEPTNO | MTHSAL | MGR |

Table: CUSTOMERS
Columns: |  CUSTOMERNUM | LASTNAME | FIRSTNAME | ADDRESS | CITY | STATE | ZIP | REFERRED
| REGION | EMAIL |

Table: ORDERS
Columns: |  ORDERNUM | CUSTOMERNUM | ORDERDATE | SHIPDATE | SHIPSTREET | SHIPCITY |
SHIPSTATE | SHIPZIP | SHIPCOST |

Table: PUBLISHER
Columns: |  PUBID | NAME | CONTACT | PHONE |

Table: AUTHOR
Columns: |  AUTHORID | LNAME | FNAME |

Table: BOOKS
Columns: |  ISBN | TITLE | PUBDATE | PUBID | COST | RETAIL | DISCOUNT | CATEGORY |

Table: ORDERITEMS
Columns: |  ORDERNUM | ITEMNUM | ISBN | QUANTITY | PAIDEACH |

Table: BOOKAUTHOR
Columns: |  ISBN | AUTHORID |

Table: PROMOTION
Columns: |  GIFT | MINRETAIL | MAXRETAIL |
```

4. Create a DataFrame called customersDF from the CUSTOMERS table, using just the LASTNAME, FIRSTNAME, and STATE fields. Display it. Then sort it on the STATE field and display that. Then sort it by both STATE and LASTNAME (put them in [ ] as the parameter to sort_values( )). Display that.

5. Create a DataFrame called orderItemsDF from the ORDERITEMS table, using just the ORDERNUM, QUANTITY, and PAIDEACH fields. Make sure it has those column names. Display it. Add a new column, TOTAL, which is the QUANTITY field times the PAIDEACH field. Display the table. Then use it to compute the overall TOTAL and display that as a money value with a label.

6. Use the syntax `CREATE TABLE newtable AS <some SELECT query>` to create a new db table called ***BOOKLIST***, where the SELECT query is a join of BOOKS and BOOKAUTHOR on ISBN joined to AUTHOR on AUTHORID, keeping only the author's LNAME and FNAME and the book's TITLE. To display this table, execute the query SELECT * FROM BOOKLIST and fetch the tuples.

7. Create a DataFrame called bookListDF from the data in #6, the BOOKLIST table. Make sure it has the same column names as BOOKLIST. Display bookListDF. Sort bookListDF on LNAME, FNAME (see the hint in #4) and display it again. Group the table by LNAME, select that column, and count( ) the entries – display the result. What's wrong with these counts? Answer in a comment in your code.

8. Repeat #3, but add the number of rows in each table to the display – use the query SELECT * FROM <table>. One entry is shown below:

```
Table: EMPLOYEES
Columns: |  EMPNO | LNAME | FNAME | JOB | HIREDATE | DEPTNO | MTHSAL | MGR |
# Rows =  5
```