Homework 11

Re-submit Assignment

Due Apr 15 by 10pm **Points** 100 **Submitting** a file upload

Your student information system has been chosen from a long list of competitors to be used by Stevens as their student data management system! To do that we need to move our solution to a relational database.

You'll need to download and install two free tools for this assignment.

1. We're going to use SQLite, a local, file based Relational Database Management System that provides a powerful subset of SQL with minimal set up hassles. Go to the **SQLlite Download Page** (https://sqlite.org/download.html), download and install a copy of SQLite for your machine. I suggest that you **download the precompiled binaries** rather than building SQLite from source code.

2. We're also using **JetBrains DataGRIP** (https://www.jetbrains.com/student/). Note that DataGRIP is a proprietary solution with a paid license but it is available free to students (Be sure to get the student license). DataGRIP provides a powerful GUI interface to most Relational Database Systems. (You're also welcome to choose a different IDE if you prefer.)

Once you've installed SQLite and DataGRIP:

1. Create a new SQLite database following the instructions from the lecture
2. Download the **students.txt**, **instructors.txt**, **majors.txt**, **grades.txt** from Canvas. These files are different from the previous versions in HW09 and HW10 so be sure to **download these new files**. You'll import these files into your database and use them for HW11.
3. Use DataGrip to open your new SQLite database and import the four files into your database as separate tables. The files have a header row with the column names and the fields are separated by **tabs**.
   - **(%24CANVAS_COURSE_REFERENCE%24/modules/items/i7ce6070196a380e8a7af4e3472678fad) students.txt:**
     - The first row is a header
     - Change the type of **CWID** to **TEXT** from **INTEGER**
     - Make **CWID** a primary key. Each **CWID** appears exactly once for each student so **CWID** is a good Primary Key for the **students** table.
   - **instructors.txt**:
     - The first row is a header
     - Change the type of **CWID** to **TEXT** from **INTEGER** and make it a primary key
   - **majors.txt**:
     - The first row is a header
     - Do **NOT** make **Major** a primary key. If you do make it a primary key, then only one course from each major will be imported into the table because the primary key must be unique across all rows.
   - **grades.txt:**
     - The first row is a header
     - Change the type of **StudentCWID** to **TEXT** from **INTEGER**
     - Change the type of **InstructorCWID** to **TEXT** from **INTEGER**

- Do NOT make either CWID a primary key. If you do make it a primary key, then only one grade from each student will be imported into the table because the primary key must be unique across all rows.

4. Run queries to answer the following questions. Submit screen dumps of DataGRIP to ***show your query and results*** for each of the following queries:
   1. What is the name of the student with CWID='10115' (NOTE: if you don't find any matching records then verify that the CWID has type TEXT in all four tables. Integers don't match strings in SQL or Python.)
   2. What is the total number of students by major? Hint: you'll need count(*) and 'group by'. See the slide on "**Aggregate functions and GROUP BY".**
   3. What is the most frequent grade for SSW 810 across all students?
   4. Display the name and cwid of each student along with the total number of courses taken by the student. Hint: You'll need to join the **students** and **grades** tables on **StudentCWID** and **CWID**.
   5. Display each student's name, CWID, course, grade, and the instructor's name for all students and grades. The result should be sorted by the student's name. Hint: You'll need to join the **grades** and **students** tables on **StudentCWID** and **CWID** and join the **instructors** table using the instructor's **CWID**. E.g. Bezos, J, CWID 10115, earned an 'A' in SSW 810 taught by Rowland.
5. Create a new branch in your GitHub repository for HW11 from your HW10 assignment and add the following new features to your code:
   1. Update your code to use the new data files that use '\t' to separate the fields and each file has a header record
   2. Add a new student_grades_table_db(self, db_path) method to your Repository class to create a **new** student grades PrettyTable that retrieves the data for the table from the database you created above using 'db_path' to specify the path of your SQLite database file. Use Python calls to execute the **student grades** summary query you defined above and use the data from executing the query to generate and display a student grades PrettyTable with the results.
   3. Add a new automated test to verify that the data retrieved from the database matches the expected rows.

- Add your database file to your GitHub repository.

Here's the output from my HW11 solution:

Majors Summary

Major	Required Courses	Electives
SFEN	['SSW 540', 'SSW 555', 'SSW 810']	['CS 501', 'CS 546']
CS	['CS 546', 'CS 570']	['SSW 565', 'SSW 810']

Student Summary

CWID	Name	Major	Completed Courses	Remaining Required	Remaining Electives	GPA
10103	Jobs, S	SFEN	['CS 501', 'SSW 810']	['SSW 540', 'SSW 555']	[]	3.38
10115	Bezos, J	SFEN	['SSW 810']	['SSW 540', 'SSW 555']	['CS 501', 'CS 546']	2.0
10183	Musk, E	SFEN	['SSW 555', 'SSW 810']	['SSW 540']	['CS 501', 'CS 546']	4.0
11714	Gates, B	CS	['CS 546', 'CS 570', 'SSW 810']	[]	[]	3.5

Instructor Summary

CWID	Name	Dept	Course	Students
98764	Cohen, R	SFEN	CS 546	1
98763	Rowland, J	SFEN	SSW 810	4
98763	Rowland, J	SFEN	SSW 555	1
98762	Hawking, S	CS	CS 501	1
98762	Hawking, S	CS	CS 546	1
98762	Hawking, S	CS	CS 570	1

Student Grade Summary

Name	CWID	Course	Grade	Instructor
Bezos, J	10115	SSW 810	A	Rowland, J
Bezos, J	10115	CS 546	F	Hawking, S
Gates, B	11714	CS 546	A	Cohen, R
Gates, B	11714	SSW 810	B-	Rowland, J
Gates, B	11714	CS 570	A-	Hawking, S
Jobs, S	10103	SSW 810	A-	Rowland, J
Jobs, S	10103	CS 501	B	Hawking, S
Musk, E	10183	SSW 555	A	Rowland, J
Musk, E	10183	SSW 810	A	Rowland, J

Deliverables:

1. The URL of the new branch in your GitHub repository with the revised source code and database file
2. Upload the queries you ran to answer part 4.1-6 above and screen dumps of the results of running the queries
3. Upload your new HW11 source code that uses the new data files and database to generate the student grades table.
4. Upload your SQLlite database file

Please let me know if you have any questions.