ECP 3004: Python for Business Analytics
Department of Economics
College of Business
University of Central Florida
Spring 2021

Assignment 8

Due Sunday, April 25, 2021 at 11:59 PM
in your GitHub repository

Instructions:

Complete this assignment on your private GitHub repo in a folder called `assignment_08`. In this folder, save your answers to Questions 1 and 2 in a file called `my_A8_queries.py`, following the script discussed in class in the file `PP_Ch_17B.py` in the course repository. When you are finished, submit it by uploading your files to your GitHub repo using any one of the approaches outlined in Question 3. You are free to discuss your approach to each question with your classmates but you must upload your own work.

Question 1:

The spreadsheet `US_state_pop_area.csv` is a listing of population figures and land area for US states and territories. In this exercise, you will create a table to store the values of population and land area and then use the table to retreive data in a variety of forms. Write Python code with SQL queries to do the following.

a) Create a new database called `population.db`.

b) Make a database table called `Density` that will hold the name of the state or territory (TEXT), the population (INTEGER), and the land area (REAL).

c) Insert the data from the table above.

d) Retrieve the contents of the table.

e) Retrieve the populations.

f) Retrieve the states that have populations of less than one million.

g) Retrieve the states that have populations of less than one million or greater than five million.

h) Retrieve the states that *do not* have populations of less than one million or greater than five million.

i) Retrieve the populations of states that have a land area greater than 200,000 square miles.

j) Retrieve the states along with their population densities (population divided by land area).

Question 2:

The spreadsheet `US_cap_cities_pop.csv` is a listing of population figures and land area for the capital cities of US states. In this exercise, you will add a new table called `Capitals` to store the values of population and land area. `Capitals` has four columns: State (TEXT), Capital (TEXT), Area (REAL), and Population (INTEGER). Then write Python code with SQL queries to do the following.

a) Retrieve the contents of the table.

b) Retrieve the populations of the states and capitals (in a list of tuples of the form `[state_population, capital_population]`).

c) Retrieve the land area of the states whose capitals that have populations greater than 100,000.

d) Retrieve the states with land densities greater than ten people per square mile and capital city populations more than 500,000.

e) Retrieve the total land area of Canada.

f) Retrieve the average population of the capital cities.

g) Retrieve the lowest population of the capital cities.

h) Retrieve the highest population of the states or territories.

i) Retrieve the states that have land densities within 0.5 persons per square mile of one another. Have each pair of states reported only once.

j) Revisit item (b) above: Retrieve the populations of the states and capitals (in a list of tuples of the form `[state_name, state_population, capital_population]`), except modify the query as follows:

   i) List only the rows corresponding to states but not territories.

   ii) List all rows corresponding to both states and territories. Leave any missing values as `None`.

Question 3:

Push your completed files to your GitHub repository following one of these three methods.

Method 1: In a Browser

Upload your code to your GitHub repo using the interface in a browser.

1. Browse to your `assignment_0X` folder in your repository (the "X" corresponds to Assignment X.).
2. Click on the "Add file" button and select "Upload files" from the drop-down menu.
3. Revise the generic message "Added files via upload" to leave a more specific message. You can also add a description of what you are uploading in the field marked "Add an optional extended description..."
4. Press the button "'Commit changes," leaving the buton set to "Commit directly to the `main` branch."

Method 2: With GitHub Desktop

Upload your code to your GitHub repo using the interface in GitHub Desktop.

1. Save your file within the folder in your repository within the folder referenced in GitHub Desktop.
2. When you see the changes in GitHub Desktop, add a description of the changes you are making in the bottom left panel.
3. Press the button "Commit to main" to commit those changes.
4. Press the button "Push origin" to push the changes to the online repository. After this step, the changes should be visible on a browser, after refreshing the page.

Method 3: At the Command Line

Push your code directly to the repository from the command line in a terminal window, such as GitBash on a Windows machine or Terminal on a Mac.

1. Open GitBash or Terminal and navigate to the folder inside your local copy of your git repo containing your assignments. Any easy way to do this is to right-click and open GitBash within the folder in Explorer. A better way is to navigate with UNIX commands, such as `cd`.
2. Enter `git add .` to stage all of your files to commit to your repo. You can enter `git add my_filename.ext` to add files one at a time, such as `my_functions.py` in this Assignment.
3. Enter `git commit -m "Describe your changes here"`, with an appropriate description, to commit the changes. This packages all the `add`ed changes into a single unit and stages them to `push` to your online repo.
4. Enter `git push origin main` to push the changes to the online repository. After this step, the changes should be visible on a browser, after refreshing the page.