5 HW #3: Loading BART data

The objective of this assignment is to create a Python script which loads the BART data into your local database. In order to receive full credit on this assignment you will need to write a Python Script which takes the raw Excel files and loads the "core" ridership data.[6]

The table that holds the data should have the following form:

```
CREATE TABLE cls.bart (
        mon int
        , yr int
        , daytype varchar(15)
        , start varchar(2)
        , term varchar(2)
        , riders float
);
```

Requirements:

- Your code should be callable from a *single* function. While you can have multiple functions (or use objects), the entire script should be run via a single command.
- The code should be in a single text file. No notebooks.
- The code should be robust to being run more than once. If the code is run twice in a row, it should not break, crash or duplicate the data in the database.
- For older time periods the clipper/fastpass data may be broken out, just use the main data and ignore the clipper data.
- You should assume that the code is going to be run on a clean computer. Any implied file structure or libraries that need to be present should be removed.
- The overall structure of the program should be as follows:
  1. Assume that all of the zip files are in a single directory (*dataDir*), which is taken as a parameter in the function.
  2. The code should unzip the files into a directory (*tmpDir*).
  3. The code should process the Excel files, extracting necessary data and reshaping it so that it can be loaded.
  4. A table should be created in your database.
  5. The clean, reshaped and standardized data should then be copied in.
- Things that you will need to standardize:
  - The format for year and month changes over time. Your code should standardize these changes.
  - The number of stations changes over time. If a particular file does not have a station, there is no need to add it.
  - The daytypes ("Weekday", "Saturday" and "Sunday") change their names throughout the data. Make sure that they are standardized. You can ignore the phrase "adjustments." The data was calculated the same way over the entire time period.

[6] For more information about the BART data, please look at `https://www.bart.gov/about/reports/ridership`

- You can assume that the schema has already been created, but you will need to handle the table creation yourself.
- You need to verify that you only load the appropriate files. In other words, make sure to either track files through the process or delete everything within the temp directory before placing files in it.
- Don't use Pandas. It's janky.
- Note that the data in the Excel spreadsheets is presented in a wide format – each column represents the average exits for a particular station. The target table ("cls.bart") is long, not wide; the data will require reshaping before it is copied in.
- The function *ProcessBart* should be called in the following manner:

  ```
  ProcessBart( tmpDir, dataDir, SQLConn=None, schema='cls', table='bart')
  ```

  the parameters of the function:

  - *tmpDir*: Directory where the unzipped files should be stored.
  - *dataDir*: Directory where the zipped files are stored.
  - *SQLConn*: Psycopg2 connection.
  - *schemaName*: The schema where the data should be loaded.
  - *tableName*: The table where the data should be loaded.
- By "core" data, I mean the Weekday, Saturday and Sunday data. Note that in many of the files there are secondary tables or sheets. For example, in the January 2011 data on the "Weekday OD" sheet, the only information that should be copied is B3:AR45.
- Think hard about what code can be repeated and what code should be put into loops or turned into functions. Needlessly repetitive code will be penalized.

Hints:

- Libraries that I used when writing this code:
  - Psycopg2
  - glob
  - xlrd
  - zipfile
  - os
  - shutil

  You can use any other library that can be installed via pip.
- Think hard about what needs to be standardized between years. The difficult part of this code is creating a data structure that allows you to iterate over the years smoothly.
- Please use psycopg2 in order to interface with the database.
- In my code, the create table (using psycopg2) looks like:

```
## Load into DB
SQLCursor = SQLConn.cursor()
SQLCursor.execute("""
  CREATE TABLE %s.%s
  (
  mon int
  , yr int
  , daytype varchar(15)
  , start varchar(2)
  , term varchar(2)
  , riders float
  );""" %(SchemaName, TableName))
  SQLCursor.execute("""COPY %s.%s FROM ’%s’ CSV;"""
          % (SchemaName, TableName, tmpDir + ’toLoad.csv’))
  SQLConn.commit()
```

- Note that I created a CSV file, “toLoad.csv” inside *tmpDir* to put the formatted and reshaped data.
- Finally, when I grade this code, I am going to download your python script to my personal computer. I will then append the following to your script and run it.

  ```
  LCLconnR = psycopg2.connect("dbname=’ncross’ user=’ncross’
           host=’localhost’ password=’XXX’")

  ProcessBart( ’\home\ncross\tmp’, ’\home\ncross\BART’,
           SQLConn=LCLconnR, schema=’cls’, table=’bart’)
  ```

  Assuming that your code runs (and I hope it does), I will then run 3-5 SQL queries on the resulting data to verify that it loaded completely and correctly.
- I will also be reading over the code itself. While I do not expect you to be Python wizards, I do expect you to be able to code efficiently. This means using loops, functions and variables to create well-written code that also contains comments to include readability.
- Please make sure that the code removes files from the temp directory before trying to load or only works on specific files that you choose. If a file is in that directory that you do not expect it should not cause your code to fail.