Objective

Assignment 2 is intended to provide some experience with SQL and data migration. SQL is the language used to interface with most relational database systems (Oracle, MS SQL, SQLite, etc). Data migration is a common step when delivering a new system that handles an existing process; the business needs to have continuity in the available information so data from legacy systems will need to be moved forward into the new platform.

Step 1: Setup

Create a 'sql' directory in your repository. Your work for this assignment will be committed there. Start the postgres daemon and create a database, you will be interacting with the database to complete this assignment. When we grade, the initdb and createdb commands will have been run prior to running your script.

Step 2: Data model

Go to the LOST specification document and become familiar with the data model. Create an sql script named 'create_tables.sql' that will create the tables associated with the data model. Run the create_tables.sql script to generate the tables in that database. You can return the database to a clean state by dropping and then recreating the database.

The datamodel is in the LOST requirements document.

Step 3: Data migration

From the course website, download the legacy data files. The legacy data files should never be committed to your git repository. Write a shell script named 'import_data.sh' that takes in a database name and port number as arguments (e.g. import_data.sh lost_dev 5432). Your script should use curl to download the legacy data from the course website. import_data.sh may call sql and python scripts that you commit to the sql directory. import_data.sh should clean up after itself, any temporary files or tables should be removed before import_data.sh finishes execution.

Legacy data is rarely clean or a good fit for a new data model. Legacy data also rarely comes with much documentation regarding what the data means or how it is related. Expect to spend some time looking at the legacy data and working through how to reshape the data to fit the new data model. You will likely need to write python or SQL scripts to assist with cleaning and reformatting the data.

The legacy data has finally been received from OSNAP. A README.txt file is included that briefly describes each of the contained csv files. **The data files are contained in the tar file here.** You can download the tar file using curl.

Step 4: Documentation

You must include a READE.txt file in the sql directory that enumerates all of the files in the sql directory with a short comment regarding what capability the file provides (e.g. create_tables.sql - SQL to create the LOST database tables).

Step 5: Verify commit and push

Git has a two step process for publishing changes. The first step is to "commit", which adds a change set to the local copy of the repository. The second step is to "push", which copies all of the committed changes to the remote repository. To verify functionality create a new database, run create_tables.sh, run import_data.sh, and then run a query to list all of the current inventory with the site the inventory is currently located at.