Oct 5, 2022
Code Review - Data Warehouse Team (*Data Dogs*)

Deficiencies:
Safety? What happens if a connection is not made? What happens if it is a bad URL?
Error handling is not in Zach's file

Write JSON is not a part of the end product, just a test function
sqlalchemy .Column is not needed

Potential bug:
Is there anything that guarantees that the input data has the correct types and format? There can be problems if INSERT happens
Error handling must be built in in order to keep let the user know that there might be possible loss of data due to incorrect format. try/catch
Implement a log function to let IT know that the client sent incorrect formatted data
Jerry is working on that.

THe columns are supposed to be based on the metadata that is passed in
Each metric should have a column in the table
Supposed to have a dynamic number of columns
Read key value pairs and make columns based on what we find
What comes in from the JSON scribes what columns are named
    Might take a loop construct to implement

Correctness is 95% there
    Mechanisms work correctly

Have not tested with bad data to make sure that bad data is not inserted into tables or if the URL is invalid

Conformity:
Conformant to double-quote
Conformant to whitespace standard
Not conformant to commenting standard
Conforms to variable and class naming standards
Does not comfort to class method standard
Conforms to import standards
In a way did conform to architecture

TO-DO:
Make file conform to commenting standard

Make class methods to camelCase
Implement try/catch, assertions, etc. to prevent bad data from getting into tables
Refactoring class methods
Change how tables methods are created

Standards:
Classes are PascalCase
Class methods are supposed to be camelCase
Variables are snake_case

Notes:
Goals for the code:

> Makes sure it works, it does
> Modules:
> Handshake service module, what runs before the actual data
> Sets up the tables for the data to be put in

Handshake:
Data is from a JSON file
Creates engines, establishes connection and creates metadata
Don't actually need to do this ^^^

Data = handshake_json
Hss
Hss.write
Hss.create

- Create table objects
- Pass to another function that inserts data into tables and tables into database
-

- Conformance to the project coding standard
- Complexity
- History (for example: new, legacy, active development, refactored, etc.)

Optimization should happen later

Sending a JSON file over the internet is not the most optimized method

> A lot of bytes are being transferred