Data Warehouse

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Brilliant Udacity Learner,

Congratulations! 🎉
You've successfully passed all the specifications with an interesting approach! I must admit that the structure of this project implementation is impressive! You should be proud of the work done as you seem to have a good hold on Data Warehouses and AWS to build an ETL pipeline for a database hosted on Redshift.

It was my pleasure reviewing this wonderful project. Please continue with this same spirit of hard work in the projects ahead.

Table Creation

The script, create_tables.py, runs in the terminal without errors. The script successfully connects to the Sparkify database, drops any tables if they exist, and creates the tables.

CREATE statements in sql_queries.py specify all columns for both the songs and logs staging tables with the right data types and conditions.

CREATE statements in sql_queries.py specify all columns for each of the five tables with the right data types and conditions.

ETL

The script, `etl.py`, runs in the terminal without errors. The script connects to the Sparkify redshift database, loads `log_data` and `song_data` into staging tables, and transforms them into the five tables.

INSERT statements are correctly written for each table and handles duplicate records where appropriate. Both staging tables are used to insert data into the songplays table.

Excellent work with your INSERT statements and utilizing DISTINCT appropriately to remove duplicate entries. JOIN clause has been used correctly to insert data into the songplays table from both staging tables. 👏

Extra resources

- Different types of JOINs

- When to use SQL JOINS
- SQL SELECT DISTINCT Statement
- SQL DISTINCT examples

Code Quality

The README file includes a summary of the project, how to run the Python scripts, and an explanation of the files in the repository. Comments are used effectively and each function has a docstring.

Scripts have an intuitive, easy-to-follow structure with code separated into logical functions. Naming for variables and functions follows the PEP8 style guidelines.

Your code is well optimized with intuitive and easy-to-follow structure which follows PEP8 style guidelines and you have limit all lines to a maximum of 79 characters. The Python standard library is conservative and requires limiting lines to 79 characters (and docstrings/comments to 72). You may also use backslash "\" for line continuation.

Suggestions

You may find this link helpful to Check your code for PEP8 requirements.

Here are some additional resource to know more about PEP8 style guidelines:

- PEP 8: Style Guide for Python Code
- How to Write Beautiful Python Code With PEP 8
- PEP-8 Tutorial: Code Standards in Python