

# Data Modeling with Postgres

REVIEW

CODE REVIEW

HISTORY

## Meets Specifications

Dear Student,

You have done a good work integrating the changes suggested in the previous review.

Going through your code was a pleasure as it is really well written in a good and modular style.

Congratulations on successfully completing your project.

Cheers!

## Table Creation

✓	The script, <code>create_tables.py</code> , runs in the terminal without errors. The script successfully connects to the Sparkify database, drops any tables if they exist, and creates the tables.
The create_table script runs without errors. Good Job!	
✓	CREATE statements in <code>sql_queries.py</code> specify all columns for each of the five tables with the right data types and conditions.
Good job. Correct DataTypes have been chosen for each column and not NULL constraint has been successfully enforced!!	

## ETL

✓	<p>The script, <code>etl.py</code>, runs in the terminal without errors. The script connects to the Sparkify database, extracts and processes the <code>log_data</code> and <code>song_data</code>, and loads data into the five tables.</p> <p>Since this is a subset of the much larger dataset, the solution dataset will only have 1 row with values for value containing ID for both <code>songid</code> and <code>artistid</code> in the fact table. Those are the only 2 values that the query in the <code>sql_queries.py</code> will return that are not-NONE. The rest of the rows will have NONE values for those two variables.</p>
The pipeline runs without errors. Good Job!	
✓	INSERT statements are correctly written for each table, and handle existing records where appropriate. <code>songs</code> and <code>artists</code> tables are used to retrieve the correct information for the <code>songplays</code> INSERT.
<p>For user table, Don't you think the <code>level</code> should be updated when a new record for an existing user with different level pops up?</p> <p>You could have used <code>excluded</code> table for the same</p>	

## 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.
<p>You should also include docstrings for all the functions in the etl file as well.</p> <p>Additionally, you can make your readme for informative and interesting by including a schema diagram and/or providing a screenshot of results, etc.</p>	
✓	Scripts have an intuitive, easy-to-follow structure with code separated into logical functions. Naming for variables and functions follows the PEP8 style guidelines.

DOWNLOAD PROJECT