PROJECT SPECIFICATION

Data Modeling with Postgres

Table Creation

CRITERIA	MEETS SPECIFICATIONS
Table creation script runs without errors.	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.
Fact and dimensional tables for a star schema are properly defined.	CREATE statements in `sql_queries.py` specify all columns for each of the five tables with the right data types and conditions.

ETL

CRITERIA	MEETS SPECIFICATIONS
ETL script runs without errors.	The script, `etl.py`, runs in the terminal without errors. The script connects to the Sparkify database, extracts and processes the `log_data` and `song_data`, and loads data into the five tables.

CRITERIA	MEETS SPECIFICATIONS
ETL script properly processes transformations in Python.	INSERT statements are correctly written for each tables and handles existing records where appropriate. `songs` and `artists` tables are used to retrieve the correct information for the `songplays` INSERT.

Code Quality

CRITERIA	MEETS SPECIFICATIONS
The project shows proper use of documentation.	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.
The project code is clean and modular.	Scripts have an intuitive, easy-to-follow structure with code separated into logical functions. Naming for variables and functions follows the PEP8 style guidelines.

Suggestions to Make Your Project Stand Out!

- Insert data using the COPY command to bulk insert log files instead of using INSERT on one row at a time
- Add data quality checks
- Create a dashboard for analytic queries on your new database