UDACITY

Return to Classroom

DISCUSS ON STUDENT HUB

Data Modeling with Postgres

REVIEW

CODE REVIEW

HISTORY

Requires Changes

3 specifications require changes

Nice work for the first submission! You have README and docstrings documenting the project well.

This project is about star schema and how to make it robust, so we need to improve the loose definitions.

Good luck on your next submission!

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 each of the five tables with the right data types and conditions.

To answer your question, yes, a PRIMARY KEY is required for each table, and you need to at least use NOT

NULL to bind the star relationship. Please add them.

FOREIGN KEY is not required if NOT NULL is in place, but that's even better.

ETL

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.

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 `songid` and `artistid` in the fact table. Those are the only 2 values that the query in the `sql_queries.py` will return that are not-NONE. The rest of the rows will have NONE values for those two variables.

We will re-evaluate after other changes.

INSERT statements are correctly written for each table, and handle existing records where appropriate. `songs` and `artists` tables are used to retrieve the correct information for the `songplays` INSERT.

The rubric requires "handle existing records where appropriate". Please add ON CONFLICT clause and pay attention to `users` table.

It's fine to DO NOTHING for other tables, but `users` need to have `level` updated, as they can switch between free and paid tiers.

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.

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH

Rate this review

START