UDACITY

‹ Return to Classroom

DISCUSS ON STUDENT HUB

Data Modeling with Postgres

REVIEW

CODE REVIEW 11

HISTORY

Meets Specifications

Great job, you are ready to go! 💯Clearly, you have acquired all the important concepts from this project. Wish you all the best for the upcoming projects! 🤞
Tip: If you are interested in knowing better how to analyze the Postgres performance, you are encouraged to read this article. Also, this article is about the pros/cons analysis of the Postgres

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.

Good job! You successfully drop all existing tables and create all necessary tables👍

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

Perfect! You 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 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.

Well done! You correctly implement all required functions in `etl.py`🎉

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.

Good job! Your INSERT statements are correctly written for each table

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.

Great summary and clearly explain the main purpose of this project👍

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

The code is clean and neat!👍And required docstrings are added for each function!

DOWNLOAD PROJECT

11 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review

START