



[Return to Classroom](#)

Data Modeling with Postgres

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear student,

You have managed to correctly implement this ETL process with python and Postgresql 🙌

If you would like to practice SQL more, I recommend the w3school website:

<https://www.w3schools.com/sql/>

Also, I recommend you learn more about the differences between SQL and NoSql databases by checking this video:

https://www.youtube.com/watch?v=QwevGzVu_zk

Keep up the good work

Have a great day,

Cheers :)

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.

Awesome 🙌

The previous reviewer had validated this specification as 'meets requirements'. I totally agree on this decision. Well done 🚀

Suggested Read 📖

You might have hear of other Relational Database Management Systems (RDBMs) than postgresql. MySQL and Sqlite are among the most famous too. Here is a nice blog post on DigitalOcean that compares each one of these systems for pros and cons.

Enjoy reading

<https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>

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

Well done 🙌

You have appropriately specified the primary and foreign keys for the tables as required. ★

Moreover, you have chosen correct data types for each one of the columns.

Suggested Read 📖

Choosing the right data type is a very important aspect of the data base design. Especially when storage and memory resource are limited. Please refer to this awesome official tutorial on choosing correct data types in postgresql.

Enjoy this read

<https://www.postgresqltutorial.com/postgresql-data-types/>

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.

It's okay if there are some null values for song titles and artist names in the `songplays` table. There is only 1 actual row that will have a songid and an artistid.

Awesome !!

1. In order to check out this rubric specification, I have run the `etl.py` script successfully

1. Then I ran the `test1.ipynb` jupyter notebook and saw that insertion has been done successfully 🙌

WELL DONE 🚀 You meet this specification !!

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.

Awesome 🙌

The previous reviewer had validated this specification as 'meets requirements'. I totally agree on this decision. Well done 🚀

Suggested Read 📖

Here is the official tutorials about the INSERT INTO statement with conflict management:

<https://www.postgresqltutorial.com/postgresql-upsert/>

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.

Awesome 🙌

The previous reviewer had validated this specification as 'meets requirements'. I totally agree on this decision. Well done 🚀

Suggested Read 📖

With markdown, you can write normal text, but you can also do much more.

And yeah, you code add cool Emoji, here is a complete list of all markdown emoji that you can use in your future projects or markdown-based blog posts:

<https://gist.github.com/rxaviers/7360908>

And here is a nice cheat sheet to discover some more tips in using Markdown:

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

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

Awesome 🙌

The previous reviewer had validated this specification as 'meets requirements'. I totally agree on this decision. Well done 🚀

Suggestion 📖

For your knowledge, here is a really nice resource that teach you the popular google-style docstrings used by millions of developers around the world

https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

Rate this review

START