**PROJECT SPECIFICATION**

**Data Warehouse**

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. |
| Staging tables are properly defined. | CREATE statements in sql\_queries.py specify all columns for both the songs and logs staging tables with the right data types and conditions. |
| 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 redshift database, loads log\_data and song\_data into staging tables, and transforms them into the five tables. |
| ETL script properly processes transformations in Python. | INSERT statements are correctly written for each table and handles duplicate records where appropriate. Both staging tables are used to insert data into the songplays table. |

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!**

* Add data quality checks
* Create a dashboard for analytic queries on your new database