Schema for Song Play Analysis

Using the song and log datasets, you'll need to create a star schema optimized for queries on song play analysis. This includes the following tables.

Fact Table

1. **songplays** - records in log data associated with song plays i.e. records with page `NextSong`
    - *songplay_id, start_time, user_id, level, song_id, artist_id, session_id, location, user_agent*

Dimension Tables

2. **users** - users in the app
    - *user_id, first_name, last_name, gender, level*

3. **songs** - songs in music database
    - *song_id, title, artist_id, year, duration*

4. **artists** - artists in music database
    - *artist_id, name, location, lattitude, longitude*

5. **time** - timestamps of records in **songplays** broken down into specific units
    - *start_time, hour, day, week, month, year, weekday*

Project Template

To get started with the project, go to the workspace on the next page, where you'll find the project template. You can work on your project with a smaller dataset found in the workspace, and then move on to the bigger dataset on AWS.

Alternatively, you can download the template files in the Resources tab in the classroom and work on this project on your local computer.

The project template includes three files:

- `etl.py` reads data from S3, processes that data using Spark, and writes them back to S3
- `dl.cfg` contains your AWS credentials
- `README.md` provides discussion on your process and decisions

Document Process

Do the following steps in your `README.md` file.

1. Discuss the purpose of this database in context of the startup, Sparkify, and their analytical goals.
2. State and justify your database schema design and ETL pipeline.
3. [Optional] Provide example queries and results for song play analysis.

Here's a guide on Markdown Syntax.

Project Rubric

Read the project rubric before and during development of your project to ensure you meet all specifications.

REMINDER: Do not include your AWS access keys in your code when sharing this project!