**Creating Relations: Joining Tables**

**Instructions**

In this practice problem, you will write join queries to retrieve specified datasets.

You will write your queries in the relevant .sql files to get the tests to pass.

Only one query should be written per file. Carefully check your syntax if you encounter errors. In particular, the name of the tables are very important because the automated tests depend on them to work properly.

**Set up a database (optional)**

This setup is optional, but if you would like a way to debug your queries, you can setup a database and connect it to DBeaver on your local machine. You can then execute any queries you write for this assignment and verify their outputs in DBeaver on your local machine.

If you have already set up DBeaver on your local machine to connect to an ElephantSQL-hosted database called thinkful\_music\_events for the previous checkpoint assignment, then you can just use the same setup to test out your queries for this assignment. Otherwise, follow the steps below:

1. Set up a new instance called thinkful\_music\_events on ElephantSQL. The instructions for creating a new database instance can be found in the "Creating and deleting databases" checkpoint of the previous module.
2. Connect DBeaver to your database instance and rename the database connection to thinkful\_music\_events for easy reference. The instructions for connecting DBeaver to your instance can be found in the "Installing DBeaver" checkpoint of the previous module.
3. Now, you can execute your queries in DBeaver.
4. After creating all the tables described below, you can run the src/seed.sql script in DBeaver to seed your database. Then you can use the SELECT \* FROM <table\_name> command to retrieve the records from the tables and check that the tables were properly populated. As you're creating your queries for this assignment, you can execute your queries in DBeaver to see if they're retrieving the datasets properly.

**Inner Join Queries**

For each of the following files, create a query as described.

1. src/get-artists-and-songs.sql: Write a join query that will return a list of artists and their songs. Include all columns from both the artists and songs tables in your result.
2. src/get-artists-and-songs-selected-columns.sql: Write a join query that will return a list of artists and their songs. Include only the artist id, artist name, and song name in your result.
3. src/get-artists-by-song-name.sql: Write a join query that will return a list of artists with song names that start with "The". Include only the artist name, song name and album name in your result.
4. src/get-artists-and-concerts.sql: Write a join query that will return all a list of artists and concerts they are performing at. Include only the artist name, concert name, concert date, and scheduled performance start and end times in your result.