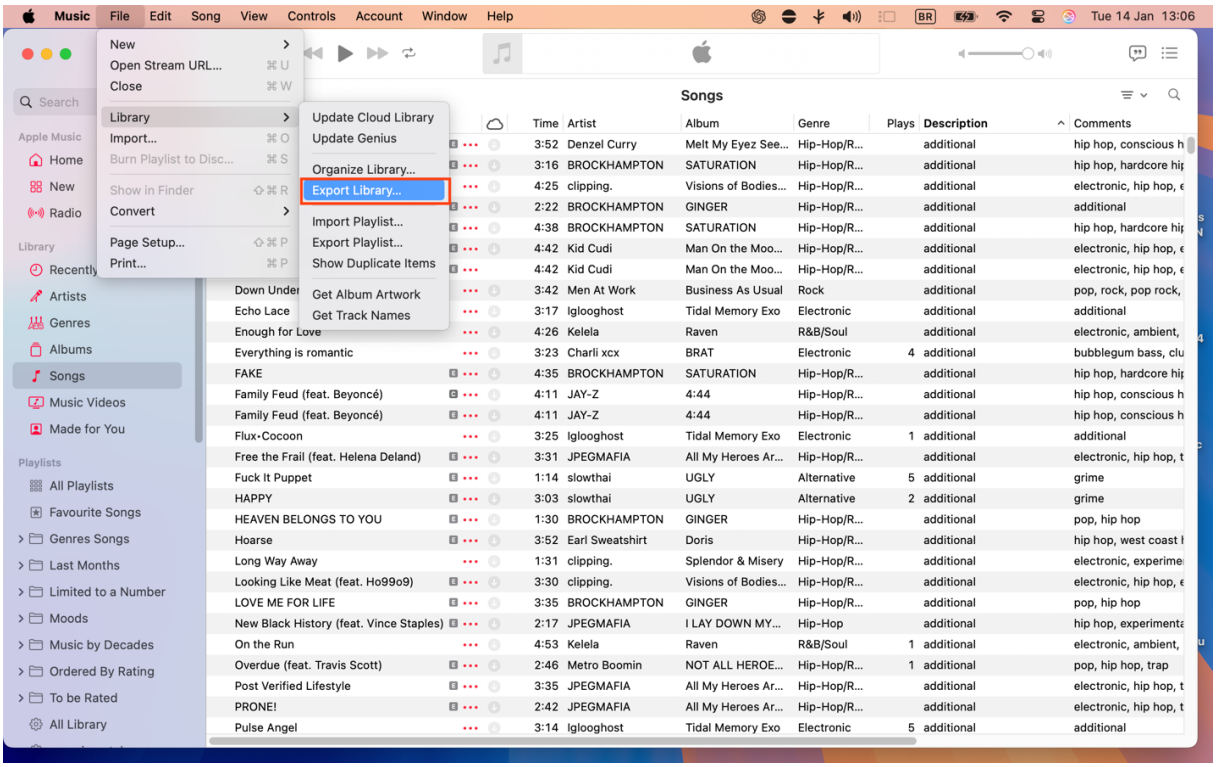
PopulateAppleMusicV2 – DOCUMENTATION

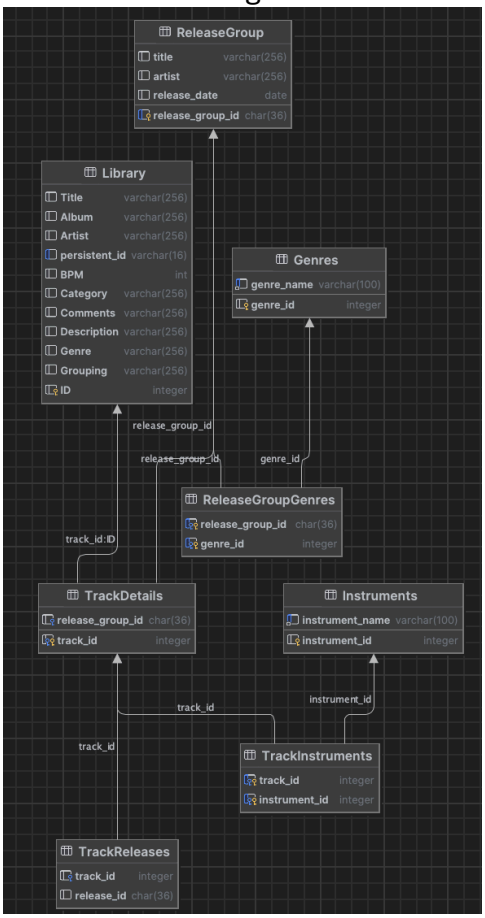
Usage:



1. Export Library to a file named library.xml
2. Place this file to the data folder in the project folder
3. Run main.py
4. After main.py is done running all the song metadata will be transferred to Apple Music desktop and it will be ready for creating Smart Playlists based on it, and also viewing song subgenres and instruments.

Summary of Files:

1. **getFromAPI.py:** Contains the API retrieving logic. It gets data from specific queries in the MusicBrainz API and also will get the BPM from Deezer API in the future.
2. **apiToDb.py:** Contains the PopulateDatabase class that is responsible for populating the SQLITE database based on the parsed data in the library table.
3. **appleScripting.py:** Applescripting.py will be responsible for communicating with the Apple Music Mac to modify the values for the description (instruments), and comments (subgenres)
4. **create_tables.sql:** Contains SQL code responsible for creating the SQLITE database.
5. **dbManager.py:** Contains a function that parses the library.xml file to SQL
6. **main.py:** Contains the main algorithm responsible for the flow of the application. Running this will achieve the goal of importing subgenres and instruments for given tracks to Apple Music.
7. **prepareDataToImport.py:** It will separate by a comma the subgenres and instruments for a given track and then update it in the library table.



Relational Database Schema