

## 1. ETL/OTHER TOOLS

We are using the virtual data integration approach. We will not be using ETL as we are not adopting the data warehousing approach. We will also not be using data persistence tools.

## 2. COLLECTING THE DATA

We are simulating the data with the help of python. Python and its libraries such as 'Random' and 'Faker' are used to generate data for the project. This dataset encapsulates various facets of user and artist information, enhancing the depth and authenticity of the simulated data. For users, we generated details such as names, genders, dates of birth, and addresses, crafting a rich set of personal information. On the artist side, we simulated data that includes artist names, genders, dates of birth, and the number of followers on a platform, providing a dynamic representation of their online presence. Additionally, the dataset encompasses details about songs, including song names, genres, and release years, contributing to a holistic and intricate synthetic environment.

### 3. SCHEMAS AND SCHEMA MAPPING

SCHEMA MAPPING BETWEEN GLOBAL\_MUSIC AND SPOTIFY DATABASE:

Global_Music	Spotify
Artist(Artist_name)	Artist(Artist_name)
Artist(DOB)	Artist(birthdate)
Artist(Followers(Spotify))	Artist(Followers)
Genre(Gen_Name)	Genre(Name)
Songs(Song_Name)	Songs(Name)
Songs(Song_Genre)	Songs(Genre)
Songs(Song_Artist)	Songs(Artist)
Songs(LaunchYear)	Songs(ReleaseYear)
Songs(Song_Platform)	“Spotify”
SONG_RECORD(Play_Date)	Plays(Play_Date)
SONG_RECORD(User_DOB)	Customer(birthdate)
SONG_RECORD(User_Gender)	Customer(Gender)
SONG_RECORD(User_Location)	Customer(Address)
SONG_RECORD(Song_Artist)	Artist(Artist_Name)
SONG_RECORD(Song_Genre)	Genre(Name)

## SCHEMA MAPPING BETWEEN GLOBAL MUSIC AND APPLE Music DATABASE

Global_Music	Apple Music
Artist(Artist_name)	Artist(Name)
Artist(DOB)	Artist(Birth_date)
Artist(Followers(Apple Music))	Artist(Followers)
Genre(Gen_Name)	Genre(Genre_Name)
Songs(Song_Name)	Songs(Song_Name)
Songs(Song_Genre)	Songs(Song_Genre)
Songs(Song_Artist)	Songs(Song_Artist)
Songs(LaunchYear)	Songs(Song_Release_Year)
Songs(Song_Platform)	“Apple Music”
SONG_RECORD(Play_Date)	Plays(Play_Date)
SONG_RECORD(User_DOB)	Customer(Birth_date)
SONG_RECORD(User_Gender)	Customer(Gender)
SONG_RECORD(User_Location)	Customer(Location)
SONG_RECORD(Song_Artist)	Artist(Name)
SONG_RECORD(Song_Genre)	Genre(Genre_Name)

## SCHEMA MAPPING BETWEEN GLOBAL MUSIC AND Resso DATABASE

Global_Music	RESSO
Artist(Artist_name)	Singer(name)
Artist(DOB)	Singer(Date_Of_Birth)
Artist(Followers(Resso))	Singer(followers)
Genre(Gen_Name)	Genre(name)
Songs(Song_Name)	Songs(name)
Songs(Song_Genre)	Songs(genre)
Songs(Song_Artist)	Songs(singer)
Songs(LaunchYear)	Songs(releaseyear)
Songs(Song_Platform)	“Resso”
SONG_RECORD(Play_Date)	Plays(playdate)
SONG_RECORD(User_DOB)	User(Date_Of_Birth)
SONG_RECORD(User_Gender)	User(Gender)
SONG_RECORD(User_Location)	User(Address)
SONG_RECORD(Song_Artist)	Singer(name)
SONG_RECORD(Song_Genre)	Genre(Name)

## SCHEMA MAPPING BETWEEN GLOBAL MUSIC AND YOUTUBE MUSIC DATABASE

Global_Music	YOUTUBE MUSIC
Artist(Artist_name)	Artist(name)
Artist(DOB)	Artist(dob)
Artist(Followers(YOUTUBE MUSIC))	Artist(followers)
Genre(Gen_Name)	Genre(name)
Songs(Song_Name)	Songs(name)
Songs(Song_Genre)	Songs(genre#)
Songs(Song_Artist)	Songs(artist#)
Songs(LaunchYear)	Songs(releaseyear)
Songs(Song_Platform)	“Youtube Music”
SONG_RECORD(Play_Date)	Plays(playdate)
SONG_RECORD(User_DOB)	user(dob)
SONG_RECORD(User_Gender)	user(sex)
SONG_RECORD(User_Location)	User(address)
SONG_RECORD(Song_Artist)	Artist(name)
SONG_RECORD(Song_Genre)	Genre(name)

The link for schema diagram is given below:

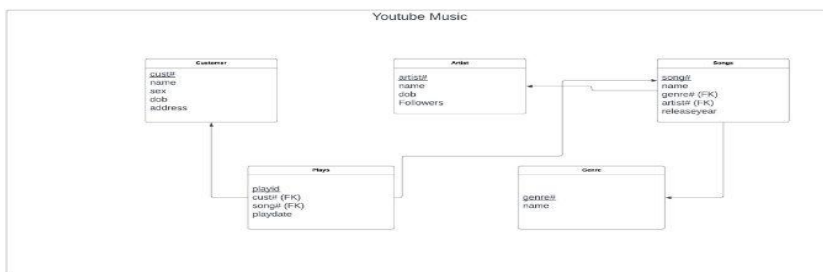
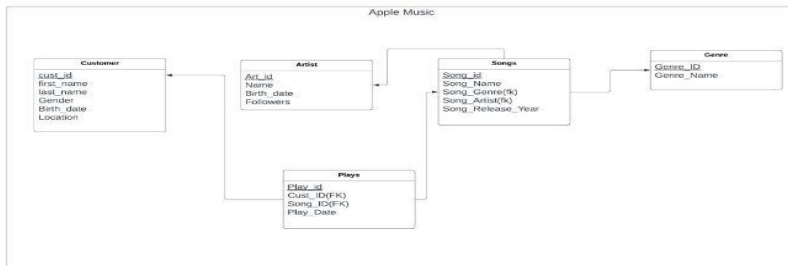
[SCHEMA DIAGRAM LINK](#)

Below is also the image of schema diagram:

## GLOBAL SCHEMA



## LOCAL SCHEMAS



### **How the communication between global and local schema will take place:**

We have used Global As View virtualisation mapping between global and local schema. The user will pass the query to the mediator(global) schema and then the query will be broken down and mapped to respective local schemas through linguistic and structural mapping. After collecting the data from respective local sources, data will be compiled and will be shown to the user. For instance if the user passes a query to get all the songs played from 01/10/23 to 03/10/23 by the females over all the platforms, then the query will breakdown and will collect the data of songs played by females over 01/10/23 to 03/10/23 from all platforms and compile it and show it to user.

Another instance can be if user wants to find songs played in a specific location by users of age group between over all platforms 18-21, then query will be broken down to sub queries and passed to local schemas using linguistic and structural mapping, and data will be collected from local schemas and will be shown to the user.