# **Data Warehousing Process**

There are 3 stages involved for the data warehousing process: Extraction, Transformation and Loading. We have used the following tools to achieve our goal:

**Big Data Distribution System:**

* Xubuntu 18.04.1 (64 bit) Virtual Machine
* Spark-2.4.0-bin-hadoop2.7

**Data Integration Tools:**

* Jupiter Notebook
* Python v3.7.4
* Apache Spark v2.4.5
* Postgres Database v12.0
* DBeaver 7.0.3 Universal Database Manager
* Tableau (data visualization)

**Other tools:**

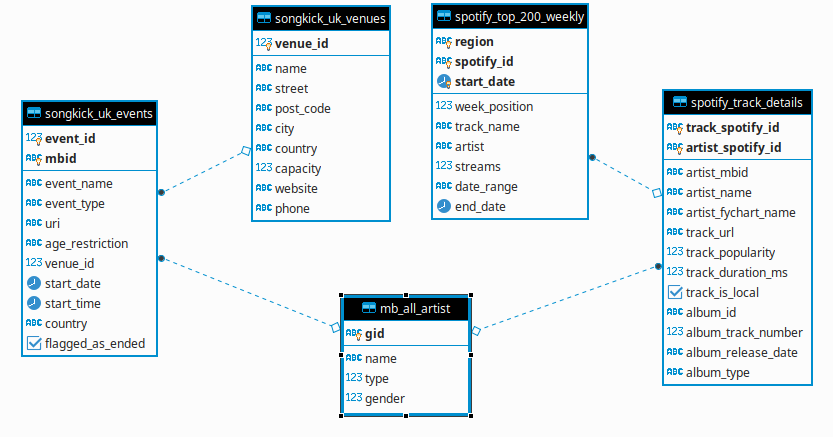
* GitHub: source code control and versioning
* Trello: project management

Each stage of the data warehousing process is documented as follows:

# **Extraction**

We developed a custom python-based data utility app (source codes attached) which we used to access the following APIs: Music Brainz, SpotifyCharts, Spotify, and Songkick; and extract data for the past one year (March 2019 – February 2020). The APIs returns json files which were then transformed and stored as comma separated (csv) files in a directory (Dirty\_Data) on the staging server. Figure 1.1 below illustrates the entity relationship diagram of our datasets (mb\_all\_artist, spotify\_top\_200\_weekly, spotify\_track\_details, songkick\_ukevents and songkick\_ukvenues).

*Figure 1.1: Entity Relationship Diagram of the source data*

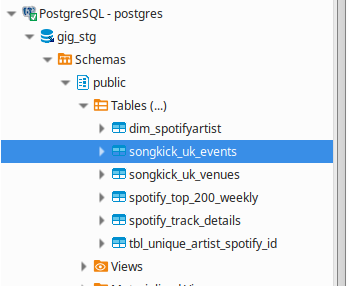


# **Transformation**

This stage involves series of cleaning, loading and transformation activities. The CSV files from the Dirty\_Data directory are then loaded using our custom utility app which implemented the spark dataframes, known for fast data manipulations, to inspect and clean the data. Details of the steps involved in cleaning our data can be found in the data cleaning document. The clean copies of the datasets: mb\_all\_artist, spotify\_top\_200\_weekly, spotify\_track\_details, songkick\_ukevents and songkick\_ukvenues are saved on another directory (Clean\_Data) on the staging server.

Once all our data are clean, it is now time to ingest them into the staging DB (Postgres database) running on our staging server. Once again, using our python-bases data utility app, the csv files are ingested into the database as tables.

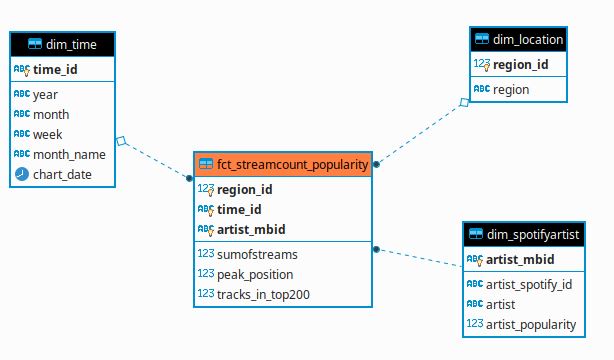
*Figure 1.2: Postgres Database showing source data*



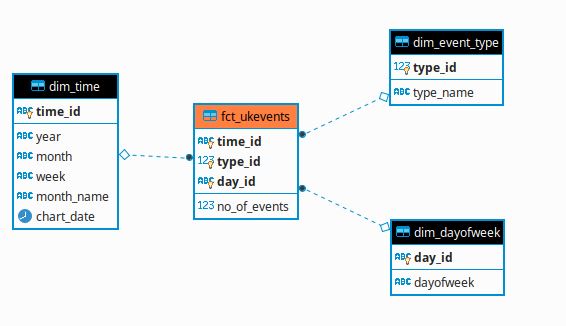
The transformation process can now begin with the creation of our dimension and fact table in relation to our business questions.

**Business Question 1:** Provide a monthly breakdown of artists by number of streams, number of tracks in the weekly top-200 chart and track peak position on spotify in the UK between Mar-2019 and Feb-2020.

*Figure 1.2.1: ERD for Business Question 1*



**Business Question 2:** Provide a monthly analysis of number of live music events by event type (festival and live concerts), day of week and event start-time in the UK between Mar-2019 and Feb-2020

*Figure 1.2.2: Star Diagram for Business Question 2*

**Business Question 3:** Provide monthly analysis of UK live music events showing number of events per artist groups (spotify-trending and others) and event type between Mar-2019 and Feb-2020

*Figure 1.2.3: Constellation Diagram for Business Question 3*

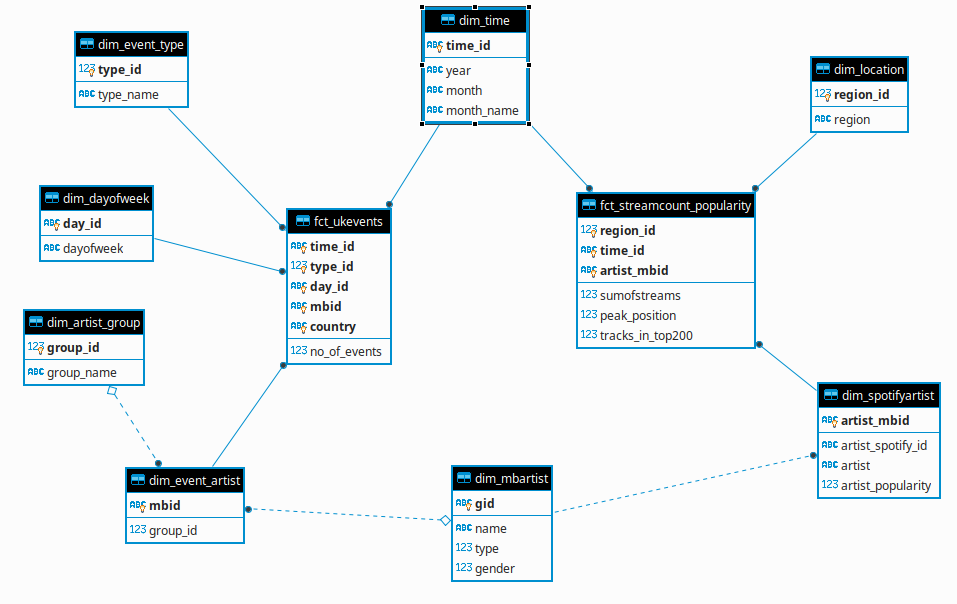


Figure 1.2.3 above illustrates a constellation which is a result of the merging Business questions 1 and 3. The idea is to combine the two fact tables (fct\_ukevents and fct\_streamcount\_poularity) to present a constellation that can provide adequate response to the business question.

# **Loading**

The transformed dimension and fact tables are then loaded, using our custom data utility app, to another Postgres DB instance on the staging server, which serves as the Datawarehouse. Connection to the Datawarehouse is granted to business users and applications for analytics and visualization.

***NB****Source codes for the python-based data utility app can be found in the source\_codes folder under the project directory.*