# Data cleaning for Artists datasets

Artists dataset is extracted from MusicBrainz database which is the reference database of Music related data containing artists, events, songs. The database is provided as an opensource database and monthly dumps are available for download. Getting data analysed and observed the missing data, some cleaning were needed. Moreover, in order to improve the ease of use, transformations were needed.

# Dataset Specification

This dataset contains artists’s information and the unique id of the artists which is the main reference of the artist in other applications and studies. This ***gid*** is known as ***mbid*** globally. In addition to ids, name, sort\_name, gender, type, ended, start\_date, end\_date, begin\_area and end\_area are existing the table. The extracted columns is presented below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data Type | Column Description | References |
| 1 | Id | Int | Sequential Id in MusicBrainzDB |  |
| 2 | Gid | Guid | Unique Id of the artists |  |
| 3 | Name | String | Full name |  |
| 4 | Sort\_Name | String | Normalized name of t he artists. In some cases the artist name is reversed. (ie. Lewis Capaldi => capaldi\_lewis) |  |
| 5 | Type | Int | Artist\_type foreign key: 1. Person 2.Group | Artists\_Type table |
| 6 | Area | Int | Area foreign key | Area table |
| 7 | Gender | Int | Gender foreign key | Genders table |
| 8 | Ended | Bit | Active artist flag |  |

The properties are cleaned and transformed into a new temporary table to then used as artists dimension table in the target data warehouse.

# ETL Process

## Extract

Postgres dump downloaded and loaded into PostgreSQL.

## Transform

Python utilized for transformation process. Data loaded into a Dataframe from PostgreSQL using sqlalchemy library. The datafame is then cleansed and transformed using pandas. The steps are as the following:

1. Searching for Null or Empty Values
   1. Gid (mbid)
   2. Name
   3. Type
   4. Area
   5. Gender
   6. Ended
2. Removing records with Null records in columns
   1. Gid
   2. Name
   3. Ended
3. Replacing Null records with default values
   1. Type
   2. Gender
   3. Area
4. Transformation
   1. “Ended” renamed to “is\_active” and Boolean data toggled.

## Transform

Cleaned data (dataframe) is loaded into staging\_db in PostgreSQL.

# References

<https://musicbrainz.org/doc/MusicBrainz_Database/Download>

# Appendix 1 – Codes

DatabaseHelper

**import sqlalchemy**

**from sqlalchemy.sql import text**

**import pandas as pd**

**class DatabaseHelper():**

**def \_\_init\_\_(self, connection\_uri):**

**self.connection\_uri = connection\_uri**

**self.\_db\_engine = sqlalchemy.create\_engine(connection\_uri)**

**def extract\_table\_to\_pandas(self, tablename, columns= None, limit = None):**

**query = "SELECT " + (columns if columns != None else '\*') + " FROM {} order by id".format(tablename) + (' limit ' + str(limit) if limit != None else '')**

**return self.extract\_query\_to\_pandas(query)**

**def extract\_query\_to\_pandas(self, query):**

**return pd.read\_sql(query, self.\_db\_engine)**

**def load\_df\_into\_dwh(self, df, tablename, schema):**

**return df.to\_sql(tablename, self.\_db\_engine, schema=schema, if\_exists="replace", index=False)**

**def run\_command(self, command):**

**with self.\_db\_engine.connect() as con:**

**statement = text("""select version();""")**

**con.execute(statement)**