# ETL: Data Wrangling

* Transform Spotify HTML table into dataframe
* Spotify table column contains track and artist
* Use split() to create separate functions for track title and artist
* Drop empty column
* Rename column to "position"
* Re-order column
* Insert new columns (source\_id, chart\_id, region\_id) to align dataframe with proposed database table "tracks"

# ETL: Project Activities

* Update web scale function to capture Spotify Viral 50 charts as well as Top 200 charts
* Ensure web scrape function creates dataframe with the desired fields in the "tracks" table. As illustrated in the ERD diagram (ERD\_diagram.png)
* Build script (Create\_Tables.sql) to create tables in table with foreign key constraints
* Create CSV's for supporting tables (e.g. Region, Source & Chart Type)
* Pre-populate supporting tables by importing CSV's using Dbeaver
* Export web scrape and load database functions from Jupyter Notebooks
* Create [webscrape\_test.py](http://webscrape_test.py/) to web scrape Spotify chat and populate "tracks" table in GCP SQL server
* Create Spotify class (spotify.py) with web scrape and load database functions
* Validate operation from CLI
* Validate "tracks" table contains 750 records using DBeaver (Top 200 & Viral 50 charts for Global, US, UK)

# ETL: Outstanding Tasks

* Python script to load database does not empty "tracks" table before loading
* Difficulty creating a method to perform (DELETE FROM tracks ;)
* Schedule python script to run periodically with GCP Cloud Scheduler
* Write up