Extraction, Transformation, and Load Technical Report

Top 100 Music ETL Project

**TABLE OF CONTENTS**

1. Introduction 3
   1. Summary 3
   2. Scope 3
   3. Technologies and resource contributions 3
   4. Definitions, Acronyms and Abbreviations 3
2. ETL Details 4
   1. Data Import/Extract Sources and Method 4
   2. Data Acquisition 4
   3. Data Transform 4
   4. Data Integrity 4
   5. Data Refresh Frequency 4
   6. Data Security 4
   7. Data Loading and Availability 5
3. Data Quality 6

|  |  |
| --- | --- |
| **1.** | **INTRODUCTION** |

*The purpose of the Extraction, Transformation, and Load (ETL) Technical Report is to capture details that pertain specifically to ETL portion of the data pipeline that is to be used in a data science project. This however does keep in mind the final target objective while performing the ETL.*

# 1.1 Summary

The purpose of the Top 100 Music ETL Project is to examine the Top 100 Singles and Artists of all time (by Sales) and find their current popularity and genre/country breakdown.

# 1.2 Scope

This section explicitly outlines the disparate data sources that are to be integrated, which components of the overall data science project is in the scope for this initiative and also lists out the components of the data science project that are not in scope here.

Data Sources:

* Wikipedia – Top 100 All time Singles and Artists – CSV format
* Spotify API - current popularity of the Top 100 All time Singles and Artists - imported API data into a data frame (Documentation: <https://developer.spotify.com/documentation/web-api/>)

# 1.3 Technologies and resource contributions

This section lists out the team members and their contributions towards the ETL initiative. Use this section to also outline (or list) the tech stack used to obtain the final outcome.

Team members:

* Ross Nicholson – API research and E, T coding
* Patrick Pruckler – Power point Presentation, API research
* Shreyashi Mukhopadhyay – Postgres Table creation and Power point (L in ETL)
* Hima Devulapalli – Project Report, API research

Technical Stack:

* Python API
* Numpy
* Matplotlib
* Pandas/Python
* Postgres/SQL

# 1.4 Definitions, Acronyms and Abbreviations

List acronyms and terms that need to be defined in this section, such as ETL: Extract, Transform and

Load

* E – Extract
* T – Transform
* L – Load
* API – Application Programming Interface
* Pandas – Framework
* Python – Programming Language
* Postgres – Database Application to import and export data into a database
* SQL – Structured Query Language

|  |  |
| --- | --- |
| **2.** | **ETL DETAILS** |

*This section outlines a more detailed description of the processes utilized/proposed to achieve the objectives of this initiative.*

# 2.1 Data Import/Extract Sources and Method

This section provides information about the data and its source. For example, API names and URLs, key parameters available and its subset which will be preserved (loaded). Data extraction protocols (API, FTP, Web scraping etc.), any permissions required to access the said extraction dataset and any restriction placed on the usage and distribution of the acquired dataset.

* Wikipedia – Imported readily available data of top 100 singles and artists of all time into a CSV.
* Spotify API - Using Spotipy, a lightweight Python library for the Spotify Web API [https://spotipy.readthedocs.io/en/latest/#](https://spotipy.readthedocs.io/en/latest/).
  + Search parameters for API - API Call using ***search*** method, querying for "***Title, Artist***"
  + We limited data to just the top response for the Artist and Title Combination while searching using the parameters in the API call

# 2.2 Data Acquisition

This section outlines the data needed, such as range and if the data is static or dynamic and needs continuous update. Outline the process to obtain again or update the dataset. The formatting and any special attributes about the data the one should be mindful of while obtaining and processing the raw dataset. How to decide on the selection of data while re-obtaining or updating. Discuss, here the dimension of the obtained dataset and if updated what is the project growth rate of the data. Lastly, address any issues or pre-requisites that needs to be cleared prior to getting the data?

* Wiki data was a static data and readily available and it was used and imported into a CSV and required only minor cleaning.
* API – user registration is required to get the unique API Key for the user to extract data using the API. All requests to Web API require authentication. This is achieved by sending a valid OAuth access token in the request header.
* The API data was obtained by calling the Spotify API using Search parameters – Title and Artist
* The API data can be called again and again to obtain the latest information

# 2.3 Data Transform

In this section address any data transformation that needs to be performed to modify, clean, filter or create existing and new parameters. Address any technical analysis performed, include design specification or data models used (example linear interpolation etc.), and any calculations performed for any newly derived fields.

* Merged the Wiki and Spotify Data frames and reformatted the data
* Clean up the ‘Genre’ column to create more generalized Genre buckets
* If there are multiple artists on album, store as one value separated by commas

# 2.4 Data Integrity

In this section discuss the reliability of the extraction source data (e.g., missing data, dates stored as text, invalid code values, text fields with odd characters, etc.). Address the frequency with which the data sources are updated and if it is necessary to update the local data at the same frequency. Lastly, how if any notification can be received when the source data is updated; and what if any notification will be sent to the internal team when the local dataset is updated.

***Wikipedia data*** – Reliable and updated frequently by Wikipedia Community.

***Spotify API data*** – Search and get information about any artist, album, track, or playlist on Spotify based on a search query.

* Search based on user’s market
* Search based on any market
* Only issue was that sometimes top result did not represent most popular version of the title when searching singles.

# 2.5 Data Refresh Frequency

This section explicitly lists the frequency with which this ETL process will refresh the local dataset (Daily, Weekly, Monthly, Quarterly, Semi-Annually, etc.).

***Wikipedia*** – Data consists of many years of sales progress to break into the top 100 so will update on an adhoc basis when a single / artist breaks into this list.

***Spotify API*** - audience, song, and playlist stats update once a day at approximately 3 PM EST. Days in Spotify for Artists are based on the UTC time zone (e.g. if you release a track at 7 PM EST, there won't be any data for that release day in Spotify for Artists as it's 00:00 UTC).

# 2.6 Data Security

This section discusses any data anonymity and security requirements need to be satisfied. Address any federally mandated HIPAA considerations, any need to build in additional privacy, Encryption, Data masking, Auditing, Backups etc.

* All requests to Web API require authentication. This is achieved by sending a valid OAuth access token in the request header.

# 2.7 Data Loading and Availability

This section addresses the data schema and during of data retention. Discuss the interface that will allow your Client/Users to access the data.

Latest Data Load will happen based on making daily calls using the Spotify API. ***Spotify API*** - audience, song, and playlist stats update once a day at approximately 3 PM EST.

Wiki data is not frequently updated unlike the Spotify data

|  |  |
| --- | --- |
| **3.** | **DATA QUALITY** |

Address in this section success criteria for this project. Summarize the parameter KPIs such as Totals and expected counts. What user acceptance testing was performed and what were the outcomes. What is the recommended site acceptance testing that your client can perform to ensure the expected outcomes meets their expectations?

Success criteria was achieving a better understanding the popularity and country/genre break down of the most famous singles of all time, this was achieved once we got all of the data we needed and visualized the information via matplotlib. In terms of accepted testing, the only thing that should be reviewed is whether the APIs can finish a loop through all artists without timing out. Sometimes latency caused these requests to time out.