Documentation of Cleaning Spotify and Youtube Data Set

Loading the data into Power query editor:

- Clicked Get Data on the Home tab.
- Selected dataset in the Navigator window.
- Afterward clicked on Transform Data to load it into Power Query Editor

Cleaning Steps in Power Query Editor:

a)Reordered and Renamed Columns for Clarity:I have reorganized the columns for improved clarity, positioning Spotify-related columns on the left side and YouTube-related columns on the right side of the table.

b)Removed and Handled Irrelevant Columns: Two irrelevant columns containing random text such as "random," "data," and some arbitrary numbers were identified and removed, as they did not provide any meaningful information. Additionally, another column containing serial numbers was relocated to the beginning of the table and renamed to "Index Number" to enhance clarity and provide a more structured reference for the dataset.

c)Checked and removed duplicacy:I remove duplicates based on the index number, which serves as the primary key for the table. To ensure that the remaining data was unique, I used the column view distribution feature and reviewed both the distinct and unique values within the column.

d)Fix Irregularities in Merged Columns: The **"Spotify_info"** column contained two merged components: the Spotify link and the track ID. I separated the column into two parts: one containing the track link and the other containing the track ID. The column was split using the BAR (|) delimiter. To ensure the data was clean after splitting, I checked for any null values or any random entries that were not valid links. Additionally, I manually clicked on a few of the links to verify that the separation was accurate. For the track IDs, I searched several of them in the official app and confirmed that they had been correctly split.
The "Youtube_info" column contained two merged components: the YouTube link and the title. To separate them, I first split the column based on position, using a straightforward approach. Since the YouTube link contained 44 characters, including a hyphen (-), I initially split the column at that position. Afterward, I split it again using the hyphen as a delimiter, as the hyphen appeared at the end of the link. Finally, I renamed the resulting columns as "YouTube_Link" and "YouTube_Title."

e)Corrected Case Sensitivity and Naming Conventions: To standardize the column

names, I converted all names to lowercase and replaced spaces with hyphens to establish a consistent format across all columns.

f)Identified and Handled Missing Values:Approach: Handling Null Values in 'Likes' and 'Views' Columns

1. **Objective:**
   To address missing (null) values in the "likes" and "views" columns of the YouTube dataset by replacing them with zero, ensuring the dataset remains usable for further analysis.
2. **Rationale:**
   In the context of YouTube data, a null value in the "likes" and "views" columns may represent no user interaction with the video. Therefore, imputing null values with `0` ensures the dataset reflects the absence of engagement rather than leaving it ambiguous.
3. **Steps:**
   - **Step 1: I**dentify all rows in the dataset where the "likes" or "views" columns contain null values.
   - **Step 2:** Replace each null value with `0`, ensuring consistency across both columns**.**
   - **Step 3:** Verify the replacement by checking for any remaining null values in both columns.
4. **Outcome:**
   After replacing nulls with zeros, the dataset is now free from missing values in the "likes" and "views" columns, allowing for accurate computation and analysis.

This approach ensures that the dataset remains complete, with null values interpreted as zero engagement.

g)Address and Fix Invalid Data Entries: To maintain data consistency and ensure that the dataset remained clean and valid, all invalid entries in the 'Views' column were replaced with `null` values.

This was achieved by selecting the 'Views' column, navigating to the Transform tab, and using the Replace Values function. Invalid entries (such as specific non-numeric or negative values) were set to be replaced with `null`.

This replacement ensures that invalid data is clearly flagged as missing (null) rather than being masked by arbitrary values, which helps preserve the integrity of the dataset for further analysis.