

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THÔNG THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC
KHO DỮ LIỆU VÀ OLAP
ĐỀ TÀI

**PHÂN TÍCH DỮ LIỆU CÁC BÀI HÁT THUỘC NỀN
TẢNG SPOTIFY**

Giảng viên hướng dẫn: ThS. Đỗ Thị Minh Phụng

Lớp: IS217.Q13

Nhóm sinh viên thực hiện: Nhóm 24

1. Đặng Quốc Cường MSSV: 23520192
2. Nguyễn Hữu Đặng Nguyên MSSV: 23521045

TP. Hồ Chí Minh, 2025

Mục lục

Lời cảm ơn	5
Từ ngữ viết tắt và thuật ngữ	6
1 Tổng quan đề tài	7
1.1 Lí do chọn đề tài	7
1.2 Giới thiệu về dataset	7
1.2.1 Tổng quan dataset	7
1.2.2 Mô tả số dòng, cột, thời gian thu thập dữ liệu	7
1.2.3 Hướng chủ đề	8
1.2.4 Mô tả chi tiết các thuộc tính trong dữ liệu	8
1.3 Tiền xử lý	9
1.3.1 Bước 1: Nhập thư viện và đọc file CSV	9
1.3.2 Bước 2: Xử lý giá trị NULL	9
1.3.3 Bước 3 : Chuyển các trường dữ liệu <code>snapshot_date</code> và <code>album_release_date</code> sang dạng <code>datetime</code>	10
1.3.4 Bước 4 : Lặp qua các cột tên, thay ký tự ‘,’ thành ‘;’ để tránh bị lỗi chia cột thuộc tính khi đưa dữ liệu vào Flat File Source	11
1.3.5 Bước 5: Xử lý ký tự đặc biệt	11
1.3.6 Bước 6: Chỉ lọc ra những bản ghi có <code>country</code> thuộc ISO các nước châu Á hoặc là <code>Global</code>	11
1.3.7 Bước 7: Xóa những thuộc tính không sử dụng	12
1.3.8 Bước 8: Lưu dataset đã lọc thành file CSV	12
1.4 Bộ truy vấn	12
1.5 Thiết kế kho dữ liệu	14
1.5.1 Lược đồ kho dữ liệu (Lược đồ bông tuyết)	14
1.5.2 Mô tả chi tiết bảng FactSongSnapshot	14
1.5.3 Mô tả chi tiết bảng Dimension	15
2 Tích hợp dữ liệu vào kho dữ liệu (SSIS)	19
2.1 Chuẩn bị công cụ và cơ sở dữ liệu	19
2.1.1 Chuẩn bị công cụ	19
2.1.2 Chuẩn bị cơ sở dữ liệu	20
2.2 Tạo project SSIS	21
2.3 Tạo các bảng Dim và bảng Fact	22
2.3.1 Tạo bảng DimDate	28
2.3.2 Tạo bảng DimCountry	35
2.3.3 Tạo bảng DimPopularityGroup	37
2.3.4 Tạo bảng DimArtist	40
2.3.5 Tạo bảng DimAlbum_Raw	47
2.3.6 Tạo bảng DimSong_Raw	49
2.3.7 Tạo bảng SongArtist_Raw	52
2.3.8 Tạo bảng FactSongSnapshot_Raw	57
2.3.9 Merge DimAlbum_Raw với DimDate để tạo bảng DimAlbum	60

2.3.10	Merge DimSong_Raw với DimAlbum để tạo bảng DimSong	64
2.3.11	Merge SongArtist_Raw với DimSong và DimArtist tạo SongArtist	68
2.3.12	Tạo bảng Fact1: Merge FactSongSnapshot_Raw với DimDate	73
2.3.13	Tạo bảng Fact2: Merge Fact1 với DimCountry	77
2.3.14	Tạo bảng Fact3: Merge Fact2 với DimSong	81
2.3.15	Tạo bảng FactSongSnapshot từ Fact3	85
2.3.16	Tạo khóa ngoại giữa các bảng	88
2.3.17	Chạy SSIS	91
2.4	Kiểm tra dữ liệu các bảng	95
2.5	Lược đồ sau khi hoàn thành	101
3	Phân tích dữ liệu trực tuyến (SSAS)	102
3.1	Tạo mới Project SSAS	102
3.2	Xác định dữ liệu nguồn (Data Source)	103
3.3	Xác định khung nhìn dữ liệu nguồn (Data Source View)	106
3.4	Xây dựng các khối (Cube) và Deploy Cube	109
3.4.1	Tạo Cube và Dimension	109
3.4.2	Thêm thuộc tính và chỉnh sửa property cho Dimension	114
3.4.3	Phân cấp trong bảng chiều	114
3.4.4	Deploy project SSAS	117
3.5	Thực hiện các câu truy vấn sử dụng SSAS, Pivot table, ngôn ngữ MDX	118
3.5.1	Câu truy vấn 1 - Sử dụng SSAS	118
3.5.2	Câu truy vấn 1 - Sử dụng Pivot Table trong Excel	119
3.5.3	Câu truy vấn 1 - Sử dụng ngôn ngữ MDX	120
3.5.4	Câu truy vấn 2 - Sử dụng SSAS	121
3.5.5	Câu truy vấn 2 - Sử dụng Pivot Table trong Excel	122
3.5.6	Câu truy vấn 2 - Sử dụng ngôn ngữ MDX	122
3.5.7	Câu truy vấn 3 - Sử dụng SSAS	124
3.5.8	Câu truy vấn 3 - Sử dụng Pivot Table trong Excel	125
3.5.9	Câu truy vấn 3 - Sử dụng ngôn ngữ MDX	125
3.5.10	Câu truy vấn 4 - Sử dụng SSAS	126
3.5.11	Câu truy vấn 4 - Sử dụng Pivot Table trong Excel	128
3.5.12	Câu truy vấn 4 - Sử dụng ngôn ngữ MDX	128
3.5.13	Câu truy vấn 5 - Sử dụng SSAS	129
3.5.14	Câu truy vấn 5 - Sử dụng Pivot Table trong Excel	131
3.5.15	Câu truy vấn 5 - Sử dụng ngôn ngữ MDX	131
3.5.16	Câu truy vấn 6 - Sử dụng SSAS	132
3.5.17	Câu truy vấn 6 - Sử dụng Pivot Table trong Excel	133
3.5.18	Câu truy vấn 6 - Sử dụng ngôn ngữ MDX	133
3.5.19	Câu truy vấn 7 - Sử dụng SSAS	134
3.5.20	Câu truy vấn 7 - Sử dụng Pivot Table trong Excel	135
3.5.21	Câu truy vấn 7 - Sử dụng ngôn ngữ MDX	135
3.5.22	Câu truy vấn 8 - Sử dụng SSAS	136
3.5.23	Câu truy vấn 8 - Sử dụng Pivot Table trong Excel	137
3.5.24	Câu truy vấn 8 - Sử dụng ngôn ngữ MDX	137
3.5.25	Câu truy vấn 9 - Sử dụng SSAS	138

3.5.26 Câu truy vấn 9 - Sử dụng Pivot Table trong Excel	139
3.5.27 Câu truy vấn 9 - Sử dụng ngôn ngữ MDX	139
3.5.28 Câu truy vấn 10 - Sử dụng SSAS	140
3.5.29 Câu truy vấn 10 - Sử dụng Pivot Table trong Excel	141
3.5.30 Câu truy vấn 10 - Sử dụng ngôn ngữ MDX	141
3.5.31 Câu truy vấn 11 - Sử dụng SSAS	142
3.5.32 Câu truy vấn 11 - Sử dụng Pivot Table trong Excel	143
3.5.33 Câu truy vấn 11 - Sử dụng ngôn ngữ MDX	143
3.5.34 Câu truy vấn 12 - Sử dụng SSAS	145
3.5.35 Câu truy vấn 12 - Sử dụng Pivot Table trong Excel	146
3.5.36 Câu truy vấn 12 - Sử dụng ngôn ngữ MDX	146
3.5.37 Câu truy vấn 13 - Sử dụng SSAS	147
3.5.38 Câu truy vấn 13 - Sử dụng Pivot Table trong Excel	148
3.5.39 Câu truy vấn 13 - Sử dụng ngôn ngữ MDX	148
3.5.40 Câu truy vấn 14 - Sử dụng SSAS	149
3.5.41 Câu truy vấn 14 - Sử dụng Pivot Table trong Excel	150
3.5.42 Câu truy vấn 14 - Sử dụng ngôn ngữ MDX	150
3.5.43 Câu truy vấn 15 - Sử dụng SSAS	151
3.5.44 Câu truy vấn 15 - Sử dụng Pivot Table trong Excel	151
3.5.45 Câu truy vấn 15 - Sử dụng ngôn ngữ MDX	151
4 QUÁ TRÌNH LẬP BÁO BIỂU (SSRS)	153
4.1 Chuẩn bị công cụ	153
4.1.1 Power BI	153
4.1.2 Google Data Studio (Locker)	153
4.2 Quá trình lập báo biểu bằng công cụ Power BI	154
4.2.1 Báo biểu 1	154
4.2.2 Báo biểu 2	158
4.2.3 Báo biểu 3	161
4.3 Quá trình lập báo biểu bằng Google Data Studio (Looker Studio)	164
4.3.1 Báo biểu 1	164
4.3.2 Báo biểu 2	169
4.3.3 Báo biểu 3	173
5 Quá trình khai phá dữ liệu (Data Mining)	178
5.1 Bối cảnh và bài toán	178
5.1.1 Động lực và bối cảnh hiện tại	178
5.1.2 Bài toán và các câu hỏi được đặt ra	178
5.1.3 Mục tiêu	179
5.2 Dữ liệu và quá trình tiền xử lý	179
5.2.1 Giới thiệu và tổng quan về dataset	179
5.2.2 Quá trình tiền xử lý	179
5.3 Phân tích dữ liệu	180
5.3.1 Tổng quan	180
5.3.2 Phân tích dữ liệu đơn biến	182
5.3.3 Phân tích dữ liệu đa biến	187

5.3.4	Ma trận hiệp tương quan (Correlation Matrix)	191
5.4	Mô hình dự đoán	192
5.4.1	Tổng quan và cách đánh giá	192
5.4.2	Giao thức đánh giá	193
5.4.3	Các bộ metric sử dụng	193
5.4.4	Xử lý và lựa chọn các đặc trưng	195
5.4.5	Logistic Regression	199
5.4.6	Random Forest Classifier	202
5.4.7	XGBoost Classifier	206
5.4.8	Kết luận	209
6	Kết luận	210
6.1	Kết quả đạt được	210
6.1.1	Về mặt dữ liệu và xử lý (ETL):	210
6.1.2	Về mặt phân tích (OLAP & Visualization):	210
6.1.3	Về mặt mô hình hóa (Data Mining):	210
6.2	Thuận lợi	210
6.3	Khó khăn	211
6.4	Hướng phát triển	211
	Tài liệu tham khảo	212

Lời cảm ơn

Sự hỗ trợ và giúp đỡ từ mọi người đã đóng góp không nhỏ vào thành công của chúng em. Đầu tiên, chúng em xin bày tỏ lòng biết ơn chân thành tối toàn thể quý cô và thầy tại Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP.HCM, cũng như quý thầy, cô thuộc Khoa Hệ thống Thông tin. Những kiến thức mà chúng em đã học từ quý thầy cô là nền tảng quan trọng, giúp chúng em hiểu sâu hơn về môn học này.

Chúng em muốn dành lời cảm ơn đặc biệt đến cô Đỗ Thị Minh Phụng - người đã đồng hành và hỗ trợ chúng em suốt hành trình làm đồ án. Sự tận tâm, chỉ bảo và hướng dẫn của cô đã là nguồn động viên quan trọng, giúp chúng em vượt qua những khó khăn trong quá trình nghiên cứu.

Đồ án của nhóm khởi nguồn từ khao khát tìm hiểu về kho dữ liệu, và dưới sự hướng dẫn của cô, chúng em đã có cơ hội áp dụng những kiến thức đã học vào thực tế. Mặc dù đã cố gắng hết sức, nhóm nhận thức rằng có những điểm cần được hoàn thiện. Cuối cùng, chúng em trân trọng sự góp ý và hướng dẫn của cô để từ những thiếu sót, chúng em có thể rút kinh nghiệm và phát triển. Mong rằng, đồ án này không chỉ là một bước đi trong hành trình học tập của chúng em, mà còn là đóng góp nhỏ bé cho lĩnh vực cơ sở dữ liệu.

Chúng em xin chúc cô sức khỏe và niềm vui, và mong rằng cô sẽ tiếp tục truyền đạt kiến thức và kinh nghiệm quý báu cho thế hệ sinh viên sau này. Xin chân thành cảm ơn!

Đặng Quốc Cường, Nguyễn Hữu Đặng Nguyên

Từ ngữ viết tắt và thuật ngữ

STT	Từ viết tắt / thuật ngữ	Ý nghĩa
1	SSIS	SQL Server Integration Services
2	SSAS	SQL Server Analysis Services
3	SSRS	SQL Server Reporting Services

Bảng 1: Bảng từ viết tắt và thuật ngữ.

1 Tổng quan đề tài

1.1 Lí do chọn đề tài

Âm nhạc là một trong những lĩnh vực gắn bó mật thiết với đời sống tinh thần của con người, đặc biệt trong thời đại công nghệ số, nơi mà các nền tảng nghe nhạc trực tuyến đã trở thành phương tiện giải trí quen thuộc của hàng triệu người dùng trên thế giới. Trong số đó, Spotify hiện đang là một trong những dịch vụ phát nhạc trực tuyến phổ biến nhất toàn cầu, sở hữu kho dữ liệu phong phú và phản ánh rõ nét xu hướng thưởng thức âm nhạc của cộng đồng. Bộ dữ liệu “**Top Spotify Songs in 73 Countries (Daily Updated)**” cung cấp danh sách các ca khúc thịnh hành ở nhiều quốc gia khác nhau, được cập nhật hằng ngày, qua đó phản ánh sự thay đổi liên tục của thị hiếu âm nhạc quốc tế.

Việc lựa chọn đề tài dựa trên bộ dữ liệu này xuất phát từ mong muốn tìm hiểu sâu hơn về các xu hướng âm nhạc đang thịnh hành, sự khác biệt và điểm tương đồng trong sở thích nghe nhạc giữa các quốc gia, cũng như sự thay đổi của thị hiếu người nghe theo thời gian. Thông qua phân tích, có thể nhận thấy những bài hát nào có sức lan tỏa mạnh mẽ toàn cầu, bài hát nào chỉ phổ biến trong phạm vi quốc gia hoặc khu vực nhất định. Ngoài ra, dữ liệu còn giúp khám phá các yếu tố tác động đến sự thành công của một ca khúc, chẳng hạn như ngôn ngữ, thể loại nhạc, hay sức ảnh hưởng của nghệ sĩ.

Đề tài này không chỉ mang lại giá trị trong việc học tập và rèn luyện kỹ năng xử lý, phân tích dữ liệu thực tế, mà còn mở ra cái nhìn đa chiều về sự phát triển của ngành công nghiệp âm nhạc hiện đại. Kết quả nghiên cứu có thể trở thành nguồn tham khảo hữu ích cho những ai quan tâm đến lĩnh vực giải trí, truyền thông và marketing, đồng thời giúp người học hiểu rõ hơn về cách dữ liệu có thể phản ánh những xu hướng xã hội trong đời sống hằng ngày.

1.2 Giới thiệu về dataset

1.2.1 Tổng quan dataset

Tên bộ dữ liệu: **Top Spotify Songs in 73 Countries (Daily Updated)**.

Bộ dữ liệu mô tả danh sách các bài hát thịnh hành hàng ngày trên nền tảng Spotify tại 73 quốc gia khác nhau. Mỗi ngày, dữ liệu ghi nhận Top 50 bài hát phổ biến nhất theo từng quốc gia, kèm theo các thông tin liên quan như tên bài hát, nghệ sĩ, album, độ dài, v.v.

Dữ liệu được đăng tải trên Kaggle bởi tác giả *asaniczka*. Đây là nguồn dữ liệu phản ánh xu hướng nghe nhạc toàn cầu cũng như sự khác biệt về thị hiếu âm nhạc giữa các quốc gia.

1.2.2 Mô tả số dòng, cột, thời gian thu thập dữ liệu

Bộ dữ liệu gốc gồm **2,110,316 dòng** và **25 cột**.

Dữ liệu được thu thập và cập nhật định kỳ, phản ánh tình hình các bài hát thịnh hành trên Spotify trong khoảng thời gian gần đây (tính đến thời điểm tải về).

Nguồn tải dataset: [Tại đây](#).

1.2.3 Hướng chủ đề

“Âm nhạc thịnh hành toàn cầu và quốc gia” (Trending Music Analytics).

1.2.4 Mô tả chi tiết các thuộc tính trong dữ liệu

STT	Tên thuộc tính	Kiểu dữ liệu	Ý nghĩa
1	spotify_id	str	Mã định danh duy nhất của bài hát trong cơ sở dữ liệu Spotify.
2	name	str	Tên của bài hát.
3	artists	str	Tên nghệ sĩ/nhóm nhạc; có thể tách bằng dấu phẩy để thành danh sách.
4	daily_rank	int	Thứ hạng hàng ngày (1–50).
5	daily_movement	int	Thay đổi thứ hạng so với ngày trước (âm/dương).
6	weekly_movement	int	Thay đổi thứ hạng so với tuần trước.
7	country	str	Mã ISO quốc gia; NULL nếu là “Global Top 50”.
8	snapshot_date	str	Ngày thu thập dữ liệu từ Spotify API (YYYY-MM-DD).
9	popularity	int	Điểm phổ biến (0–100).
10	is_explicit	bool	Bài hát có nội dung nhạy cảm/explicit lyrics.
11	duration_ms	int	Thời lượng bài hát (ms). Thường chuyển sang phút để phân tích.
12	album_name	str	Tên album chứa bài hát.
13	album_release_date	str	Ngày phát hành album.
14	danceability	float	Độ phù hợp để nhảy (0–1).
15	energy	float	Mức năng lượng và cường độ (0–1).
16	key	int	Tông nhạc (0–11).
17	loudness	float	Độ to trung bình (dB).
18	mode	int	0 = minor, 1 = major.
19	speechiness	float	Mức độ xuất hiện lời nói (0–1).
20	acousticness	float	Mức độ acoustic (0–1).
21	instrumentalness	float	Xác suất không có giọng hát (0–1).
22	liveness	float	Khả năng có khán giả trực tiếp trong bản thu (0–1).
23	valence	float	Độ tích cực cảm xúc của bài hát (0–1).
24	tempo	float	Nhịp độ (beats per minute).
25	time_signature	int	Chữ nhịp (ví dụ: 3, 4).

Bảng 2: Mô tả các thuộc tính trong bộ dữ liệu Spotify

1.3 Tiềm xử lý

Dầu tiên vì bộ dữ liệu ban đầu quá lớn, tận **2,110,316 dòng**. Và cũng như một số trường dữ liệu của bộ dữ liệu chứa một số ký tự đặc biệt, cũng như tồn tại giá trị NULL dẫn đến không thể đưa vào SSIS nên nhóm sẽ thực hiện tiềm xử lý dữ liệu. Nhóm sử dụng ngôn ngữ Python để tiến hành các bước tiềm xử lý dữ liệu dưới đây:

1.3.1 Bước 1: Nhập thư viện và đọc file CSV

```
import pandas as pd
df = pd.read_csv("/kaggle/input/top-spotify-songs-in-73-countries-daily-updated/universal_top_spotify_songs.csv")
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2110316 entries, 0 to 2110315
Data columns (total 25 columns):
 #   Column            Dtype  
 --- 
 0   spotify_id        object  
 1   name              object  
 2   artists            object  
 3   daily_rank        int64  
 4   daily_movement    int64  
 5   weekly_movement   int64  
 6   country            object  
 7   snapshot_date     object  
 8   popularity         int64  
 9   is_explicit       bool   
 10  duration_ms       int64  
 11  album_name        object  
 12  album_release_date object  
 13  danceability      float64 
 14  energy             float64 
 15  key               int64  
 16  loudness           float64 
 17  mode               int64  
 18  speechiness        float64 
 19  acousticness       float64 
 20  instrumentalness  float64 
 21  liveness            float64 
 22  valence             float64 
 23  tempo               float64 
 24  time_signature     int64  
dtypes: bool(1), float64(9), int64(8), object(7)
memory usage: 388.4+ MB
```

Hình 1: Nhập thư viện và đọc file CSV

1.3.2 Bước 2: Xử lý giá trị NULL

```
null_counts_original = df.isnull().sum()
print(null_counts_original)
```

spotify_id	0
name	30
artists	29
daily_rank	0
daily_movement	0
weekly_movement	0
country	28908
snapshot_date	0
popularity	0
is_explicit	0
duration_ms	0
album_name	822
album_release_date	659

Hình 2: Số lượng giá trị NULL tại các trường dữ liệu

Tiến hành xóa những hàng tồn tại ít nhất một trường dữ liệu mang giá trị NULL. Tuy nhiên đối với trường "country" là ISO quốc gia và NULL nếu là "Global Top 50". Do đó với các giá trị NULL ở trường này, ta có thể gán quy ước là Global.

```
df_cleaned = df.dropna(subset=['name', 'artists', 'album_name', 'album_release_date']).copy()

# Diền NULL trong country
df_cleaned['country'] = df_cleaned['country'].fillna('Global')

# Kiểm tra null
null_counts = df_cleaned.isnull().sum()
print(null_counts)
```

Hình 3: Xóa các hàng tồn tại giá trị NULL

1.3.3 Bước 3 : Chuyển các trường dữ liệu snapshot_date và album_release_date sang dạng datetime

Nếu để dữ liệu ngày của trường snapshot_date và album_release_date ở dạng string, hệ quản trị cơ sở dữ liệu sẽ sắp xếp theo thứ tự từ điển (lexicographical order), tức là so sánh ký tự từ trái qua phải chứ không hiểu đây là ngày tháng.

Ví dụ: 2025-9-14 sẽ được so sánh với 2025-10-01. Khi so sánh ký tự sau dấu gạch ngang:

- Ký tự đầu tiên của tháng 9 là '9' có mã ASCII lớn hơn '1' (trong 10).
- Do đó, chuỗi 2025-9-14 sẽ bị xếp sau 2025-10-01, mặc dù về mặt thời gian thì tháng 9 phải đứng trước tháng 10.

Ngược lại, nếu chuyển cột ngày sang dạng datetime, hệ thống sẽ so sánh theo giá trị thời gian thực sự. Nhờ đó, việc sắp xếp, lọc theo khoảng thời gian, hay phân tích theo ngày/tháng/quý/năm trong OLAP đều chính xác.

```
df_cleaned[['album_release_date', 'snapshot_date']] = df_cleaned[['album_release_date', 'snapshot_date']].apply(
    pd.to_datetime, errors='coerce'
)

print(df_cleaned[['album_release_date', 'snapshot_date']].dtypes)
print(df_cleaned[['album_release_date', 'snapshot_date']].head())

album_release_date      datetime64[ns]
snapshot_date           datetime64[ns]
dtype: object
  album_release_date snapshot_date
0        2024-09-26      2025-06-11
1        2025-06-05      2025-06-11
2        2024-12-27      2025-06-11
3        2025-03-07      2025-06-11
4        2024-05-17      2025-06-11
```

Hình 4: Chuyển snapshot_date và album_release_date sang dạng datetime

1.3.4 Bước 4 : Lắp qua các cột tên, thay ký tự ‘,’ thành ‘;’ để tránh bị lỗi chia cột thuộc tính khi đưa dữ liệu vào Flat File Source

```
# Thay dấu phẩy bằng dấu chấm phẩy trong 3 cột
cols = ["artists", "album_name", "name"]
for col in cols:
    df_cleaned[col] = df_cleaned[col].str.replace(",", ";", regex=False)
```

Hình 5: Thay ký tự ‘,’ thành ‘;’

1.3.5 Bước 5: Xử lý ký tự đặc biệt

Việc loại bỏ hoặc thay thế các ký tự đặc biệt trong các trường dữ liệu (ví dụ như name, artists) là cần thiết vì:

- Hỗ trợ quá trình truy vấn và phân tích dữ liệu (tránh lỗi khi viết câu lệnh SQL/MDX hoặc khi lọc dữ liệu).
- Đảm bảo khả năng so sánh và đối chiếu dữ liệu chính xác hơn.

```
import re
import pandas as pd

def clean_text(text):
    if pd.isnull(text):
        return text
    # Giữ lại chữ cái Unicode (\w), số (\d), khoảng trắng, dấu chấm phẩy
    return re.sub(r'[^\\w\\s;]', '', text, flags=re.UNICODE)

# Áp dụng
df_cleaned['name'] = df_cleaned['name'].apply(clean_text)
df_cleaned['artists'] = df_cleaned['artists'].apply(clean_text)
```

Hình 6: Xử lý ký tự đặc biệt

1.3.6 Bước 6: Chỉ lọc ra những bản ghi có country thuộc ISO các nước châu Á hoặc là Global

```
import pandas as pd

def songs_by_criteria_and_date(df: pd.DataFrame) -> int:
    """
    Tìm bài hát có mã ISO thuộc châu Á hoặc nhãn là 'Global'

    Args:
        df (pd.DataFrame): DataFrame chứa dữ liệu bài hát.

    Returns:
        df_new: Các bài hát thỏa mãn điều kiện.
    """

    # Danh sách mã ISO của các quốc gia châu Á
    asian_iso_codes = [
        'AF', 'AM', 'SA', 'AZ', 'BH', 'BD', 'BT', 'BN', 'AE', 'KH', 'TW',
        'TL', 'GE', 'KP', 'HK', 'IN', 'ID', 'IR', 'IQ', 'IL', 'JO', 'KZ',
        'KW', 'KG', 'LA', 'LB', 'MO', 'NV', 'MY', 'WN', 'MM', 'NP', 'JP',
        'OM', 'PK', 'PS', 'PH', 'QA', 'CY', 'SG', 'LK', 'SY', 'TJ', 'TH',
        'TR', 'KP', 'CN', 'TW', 'UZ', 'VN', 'YE'
    ]

    # Lọc các bài hát thỏa mãn điều kiện về quốc gia và năm phát hành
    is_asian_or_global = df['country'].isin(asian_iso_codes) | (df['country'] == 'Global')

    filtered_df = df[is_asian_or_global]

    return filtered_df

df_new = songs_by_criteria_and_date(df_cleaned)
```

Hình 7: Chỉ lọc ra những bản ghi có country thuộc ISO các nước châu Á hoặc là Global

1.3.7 Bước 7: Xóa những thuộc tính không sử dụng

```
df_new = df_new.drop(['speechiness', 'liveness', 'valence'], axis=1)
```

Hình 8: Xóa những thuộc tính không sử dụng

1.3.8 Bước 8: Lưu dataset đã lọc thành file CSV

```
df_new.to_csv("spotify_final.csv", index=False)
```

Hình 9: Lưu dataset đã lọc thành file CSV

Sau khi tiền xử lý, data ban đầu chỉ còn **522,704** dòng.

1.4 Bộ truy vấn

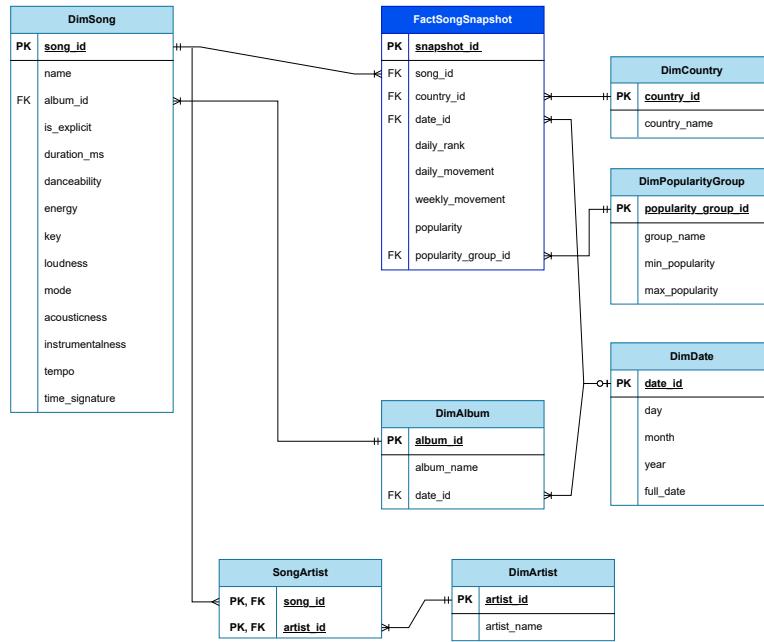
1. **Phân tích top các bài hát theo quốc gia:** Tìm ra 10 bài hát có thứ hạng cao nhất (`daily_rank` từ 1–10) tại mỗi quốc gia, vào ngày 01/01/2025. Nhóm dữ liệu theo `country` và lọc các bản ghi có `daily_rank` ≤ 10 và có `snapshot_date` là 01/01/2025.
2. **Phân tích thời lượng bài hát theo album:** Thống kê thời lượng trung bình (`duration_ms`) của các bài hát xuất hiện trên bảng xếp hạng trong một giai đoạn cụ thể, nhóm theo `album_name`.
3. **So sánh đặc điểm âm nhạc giữa các chế độ (major/minor):** So sánh `danceability` và `energy` trung bình của các bài hát có mặt trên bảng xếp hạng trong khoảng thời gian nhất định, nhóm theo `mode` (trưởng/thứ) để xem xu hướng âm nhạc của từng chế độ.
4. **Phân tích mức độ phổ biến theo nghệ sĩ:** Xác định các nghệ sĩ nào có tác phẩm được ưa chuộng nhất trong ngày. Nhóm dữ liệu theo `artists` và tính điểm phổ biến trung bình (`popularity`).
5. **Phân tích xu hướng thứ hạng hàng ngày:** Xác định những bài hát có sự thay đổi thứ hạng lớn nhất (tăng hoặc giảm) trong một khoảng thời gian cụ thể. Lọc dữ liệu theo `snapshot_date` và phân tích giá trị của `daily_movement`.
6. **Thông kê bài hát theo năm phát hành album:** Phân tích số lượng bài hát và điểm phổ biến trung bình (`popularity`) của các bài hát xuất hiện trên bảng xếp hạng, nhóm theo năm phát hành album để quan sát xu hướng qua từng năm.
7. **Phân tích độ nổi tiếng của bài hát theo độ explicit:** Với từng quốc gia, thống kê tổng điểm `popularity` theo giá trị `is_explicit`.
8. **Phân tích số lượng:** bài hát phân biệt (unique songs) xuất hiện trong bảng xếp hạng của từng quốc gia trong **3 tháng đầu tiên của năm 2025**. Mục tiêu là xác định quy mô danh sách bài hát của từng quốc gia, qua đó thấy được mức độ đa dạng âm nhạc của các thị trường

trong giai đoạn đầu năm.

9. **Phân tích biến động thứ hạng theo cấp thời gian:** (Drill-down theo hierarchy Date) Phân tích sự thay đổi thứ hạng trung bình (*daily_movement*) của các bài hát từ năm → tháng → ngày, để xem chi tiết hơn về mức độ biến động theo thời gian.
10. **Phân tích độ hot của bài hát theo thời gian:** So sánh số lượng bài hát xuất hiện trên bảng xếp hạng theo từng quốc gia và từng tháng trong năm 2024.
11. **Tìm các nhạc sĩ có bài hát nằm trong bảng xếp hạng toàn cầu trong một ngày:** Xác định tên của các nhạc sĩ có bài hát nằm trong top 10 Global của một ngày nhất định.
12. **Tìm album có số lượng bài hát nhiều nhất trong một khoảng thời gian:** Tìm ra tên album có số lượng bài hát nằm trong top 50 nhiều nhất của một quốc gia nhất định trong khoảng thời gian nhất định.
13. **So sánh sự thay đổi thứ hạng hàng tuần (weekly_movement) giữa các quốc gia:** Nhóm theo *country* và tính *weekly_movement* trung bình để xem thị trường nào có sự biến động thứ hạng mạnh nhất.
14. **Phân tích thời lượng bài hát trung bình theo mức độ phổ biến:** Phân nhóm các bài hát dựa trên điểm phổ biến (*popularity*) thành các khoảng (ví dụ: 0-20, 21-40,...) và tính thời lượng trung bình (*duration_ms*) cho mỗi nhóm.
15. **Phân tích số lượng bài hát có chữ nhịp (time_signature) cụ thể:** Đếm số lượng bài hát xuất hiện trên bảng xếp hạng trong một giai đoạn cụ thể, nhóm theo *time_signature* để xác định chữ nhịp được sử dụng phổ biến nhất.

1.5 Thiết kế kho dữ liệu

1.5.1 Lược đồ kho dữ liệu (Lược đồ bông tuyết)



Hình 10: Lược đồ bông tuyết

1.5.2 Mô tả chi tiết bảng FactSongSnapshot

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	snapshot_id	int	PK	Mã định danh duy nhất của bản ghi.
2	song_id	int	FK	Mã bài hát.
3	country_id	int	FK	Mã quốc gia.
4	date_id	int	FK	Mã ngày dữ liệu được thu thập.
5	daily_movement	int		Sự thay đổi thứ hạng của bài hát so với ngày trước đó.
6	weekly_movement	int		Sự thay đổi thứ hạng của bài hát so với tuần trước đó.
7	popularity	int		Điểm phổ biến của bài hát tại thời điểm chụp dữ liệu.
8	popularity_group_id	int	FK	Khóa chính của bảng, định danh duy nhất cho mỗi nhóm mức độ phổ biến.
9	daily_rank	int		Thứ hạng hàng ngày.

Bảng 3: Mô tả chi tiết bảng FactSongSnapshot

1.5.3 Mô tả chi tiết bảng Dimension

1.5.3.1 DimCountry

Gồm 2 thuộc tính, mỗi dòng dữ liệu chứa thông tin của một quốc gia.

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	country_id	int	PK	Mã quốc gia.
2	country_name	Varchar(10)		Tên đầy đủ của quốc gia.

Bảng 4: Mô tả chi tiết bảng DimCountry

1.5.3.2 DimDate

Gồm 5 thuộc tính, mỗi dòng dữ liệu chứa thông tin của một ngày cụ thể.

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	date_id	int	PK	Mã định danh duy nhất của ngày.
2	day	int		Ngày trong tháng.
3	month	int		Tháng trong năm.
4	year	int		Năm đầy đủ.
5	full_date	date		Ngày đầy đủ theo định dạng YYYY-MM-DD.

Bảng 5: Mô tả chi tiết bảng DimDate

1.5.3.3 DimSong

Gồm 15 thuộc tính, mỗi dòng dữ liệu chứa thông tin của một bài hát.

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	song_id	int	PK	Mã định danh duy nhất của bài hát.
2	spotify_id	Nvarchar(500)		Tên của bài hát.
3	name	Nvarchar(500)		Tên của bài hát.
4	album_id	int	FK	Khóa ngoại liên kết đến bảng chiều album, dùng để xác định album chứa bài hát.
5	is_explicit	bool		Bài hát có nội dung nhạy cảm/explicit lyrics.
6	duration_ms	int		Thời lượng của bài hát (đơn vị: mili giây).
7	danceability	float		Mức độ phù hợp để nhảy (giá trị từ 0 đến 1).
8	energy	float		Mức năng lượng và cường độ của bài hát (giá trị từ 0 đến 1).
9	key	int		Tông nhạc của bài hát (giá trị từ 0 đến 11).
10	loudness	float		Độ to trung bình của bài hát (đơn vị: dB).
11	mode	int		Chế độ âm nhạc (0 = thứ, 1 = trưởng).
12	acousticness	float		Mức độ acoustic (giá trị từ 0 đến 1).
13	instrumentalness	float		Xác suất không có giọng hát (giá trị từ 0 đến 1).
14	tempo	float		Nhịp độ của bài hát (đơn vị: beats per minute).
15	time_signature	int		Chữ nhịp của bài hát (ví dụ: 3, 4).

Bảng 6: Mô tả chi tiết bảng DimSong

1.5.3.4 DimAlbum

Gồm 3 thuộc tính, mỗi dòng dữ liệu chứa thông tin của một album.

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	album_id	int	PK	Mã định danh duy nhất của album.
2	album_name	Nvarchar(500)		Tên của album.
3	date_id	int	FK	Khóa ngoại liên kết đến bảng chiều ngày phát hành.

Bảng 7: Mô tả chi tiết bảng DimAlbum

1.5.3.5 DimArtist

Gồm 2 thuộc tính, mỗi dòng dữ liệu chứa thông tin của một tác giả.

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	artist_id	int	PK	Mã định danh duy nhất của nghệ sĩ.
2	artist_name	Nvarchar(500)		Tên của nghệ sĩ.

Bảng 8: Mô tả chi tiết bảng DimArtist

1.5.3.6 SongArtist

Gồm 2 thuộc tính, mỗi dòng dữ liệu chứa thông tin liên kết giữa một bài hát và một tác giả, dùng để giải quyết mối quan hệ nhiều-nhiều giữa hai bảng.

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	song_id	int	PK, FK	Khóa ngoại liên kết đến bảng bài hát.
2	artist_id	int	PK, FK	Khóa ngoại liên kết đến bảng nghệ sĩ.

Bảng 9: Mô tả chi tiết bảng SongArtist

1.5.3.7 DimPopularityGroup

Bảng DimPopularityGroup dùng để phân loại các bài hát thành các nhóm mức độ phổ biến (popularity) khác nhau, giúp thuận tiện cho việc phân tích OLAP theo từng khoảng giá trị. Mỗi dòng dữ liệu trong bảng đại diện cho một nhóm điểm phổ biến (ví dụ: 0–20, 21–40, ...).

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	popularity_group_id	int	PK	Khóa chính của bảng, định danh duy nhất cho mỗi nhóm mức độ phổ biến.
2	group_name	varchar(20)		Tên của nhóm mức độ phổ biến (ví dụ: '0–20', '21–40', ...).
3	min_popularity	int		Giá trị điểm phổ biến nhỏ nhất thuộc nhóm.
4	max_popularity	int		Giá trị điểm phổ biến lớn nhất thuộc nhóm.

Bảng 10: Mô tả chi tiết bảng DimPopularityGroup

2 Tích hợp dữ liệu vào kho dữ liệu (SSIS)

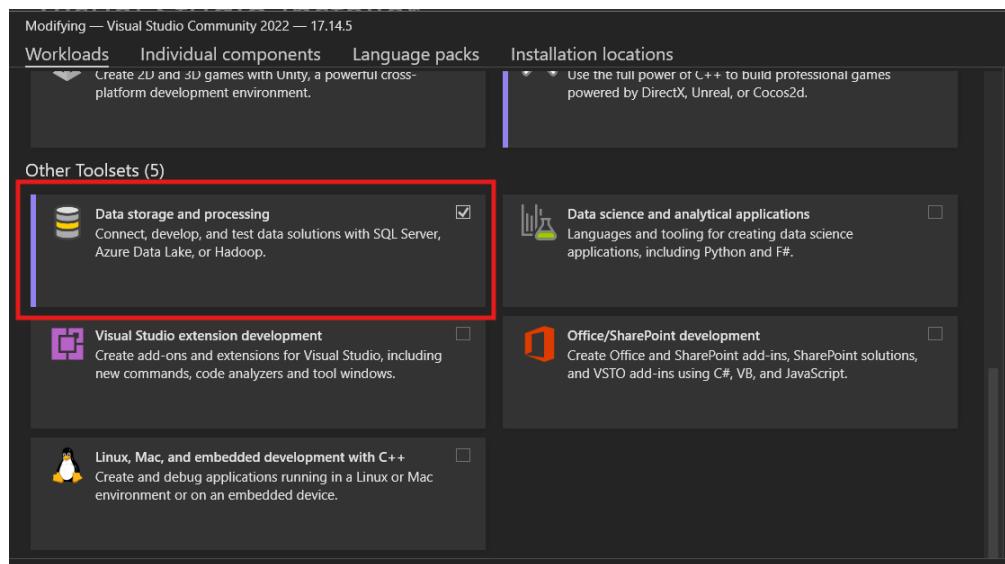
2.1 Chuẩn bị công cụ và cơ sở dữ liệu

2.1.1 Chuẩn bị công cụ

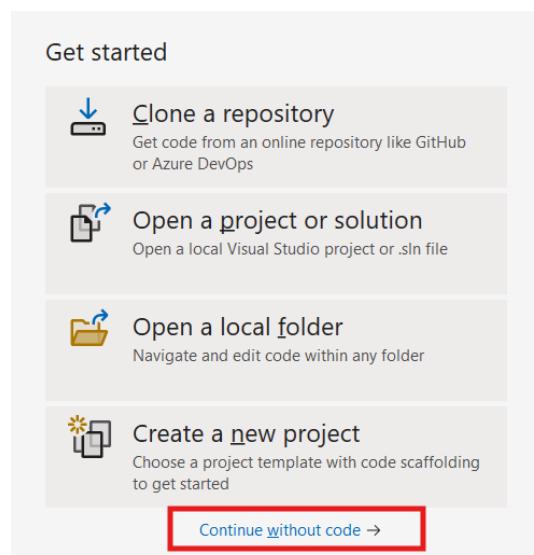
Để thực hiện quá trình SSIS, ta cần cài đặt các công cụ:

- Visual Studio Community 2022.
- SQL Server Integration Services Project.

Bước 1: Trong **Visual Studio Installer**, chọn vào mục “**Data storage and processing**”, cài đặt **SQL Server Data Tools**.

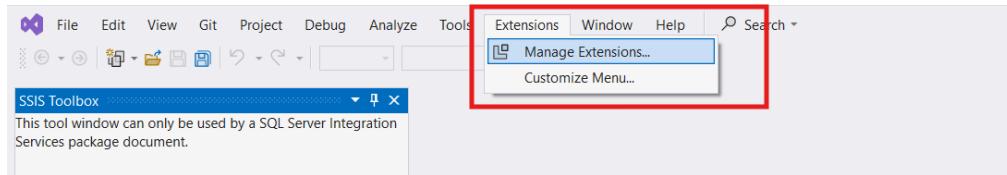


Bước 2: Vào Visual Studio, chọn “Continue without code”.

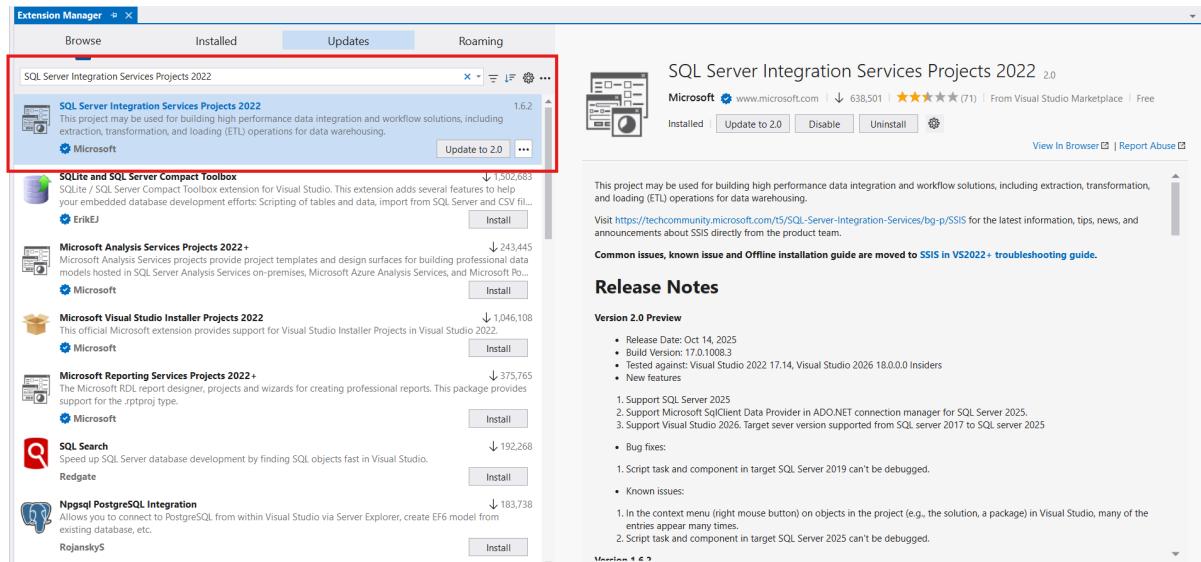


Báo cáo đồ án

Bước 3: Trên thanh công cụ, vào mục “Extensions”, chọn “Manage Extensions”.

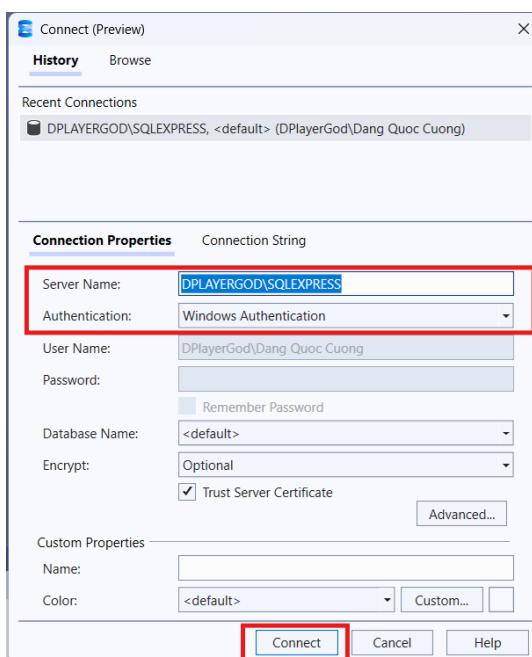


Bước 4: Tìm kiếm và chọn tải “SQL Server Integration Services Projects 2022”.

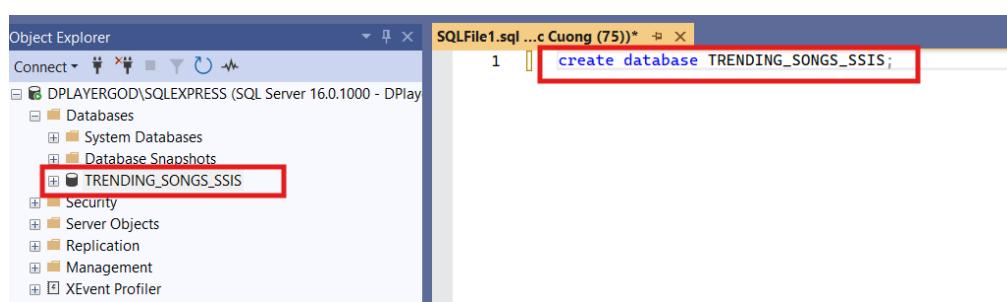


2.1.2 Chuẩn bị cơ sở dữ liệu

Bước 1: Mở SQL Server Management Studio 21 và kết nối với server.

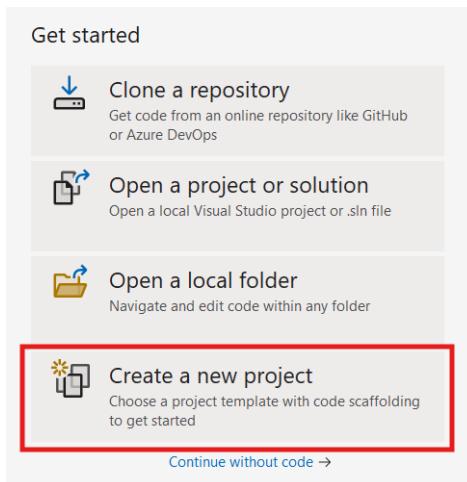


Bước 2: Tạo một database mới có tên **TRENDING_SONGS_SSIS** để lưu các bảng Dim, Fact chứa dữ liệu của bộ dữ liệu.

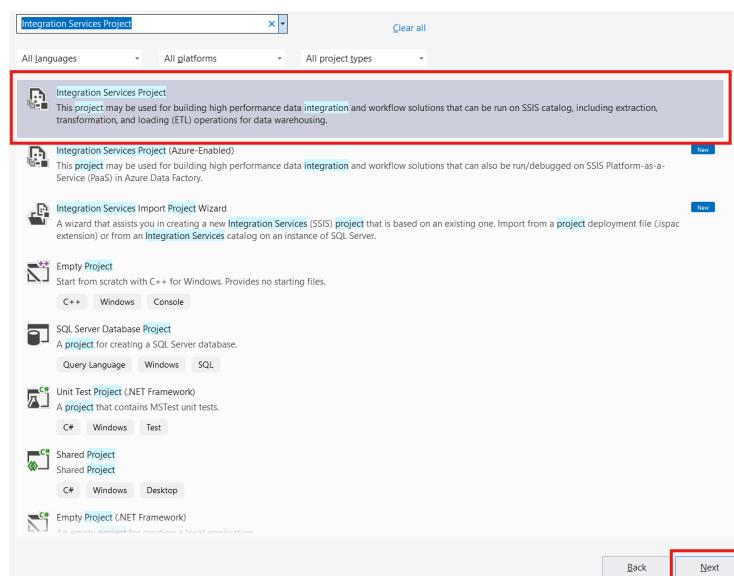


2.2 Tạo project SSIS

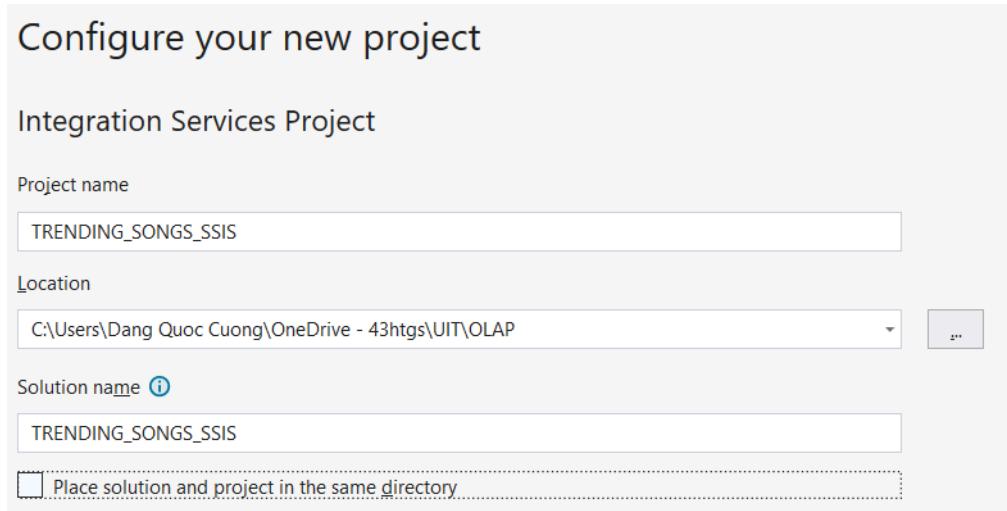
Bước 1: Vào Visual Studio, chọn “Create a new project”.



Bước 2: Tìm và chọn “Integration Services Project”, nhấn “Next” để tiếp tục.

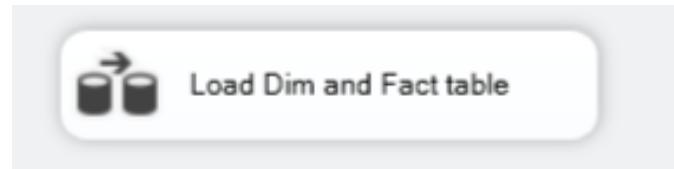


Bước 3: Đặt tên project và đường dẫn.



2.3 Tạo các bảng Dim và bảng Fact

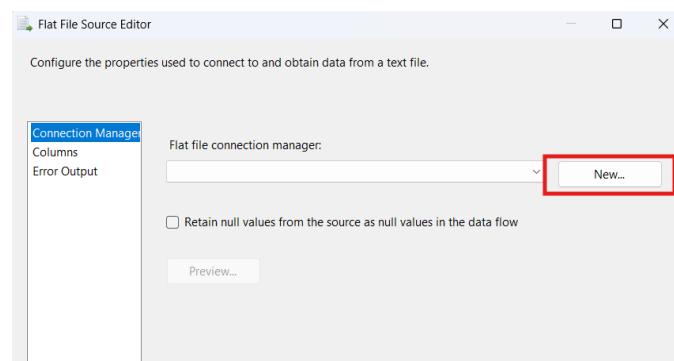
Tạo Data Flow Task có tên “Load Dim and Fact _ raw table” để chuẩn bị cho việc tạo các bảng Dim và Fact.



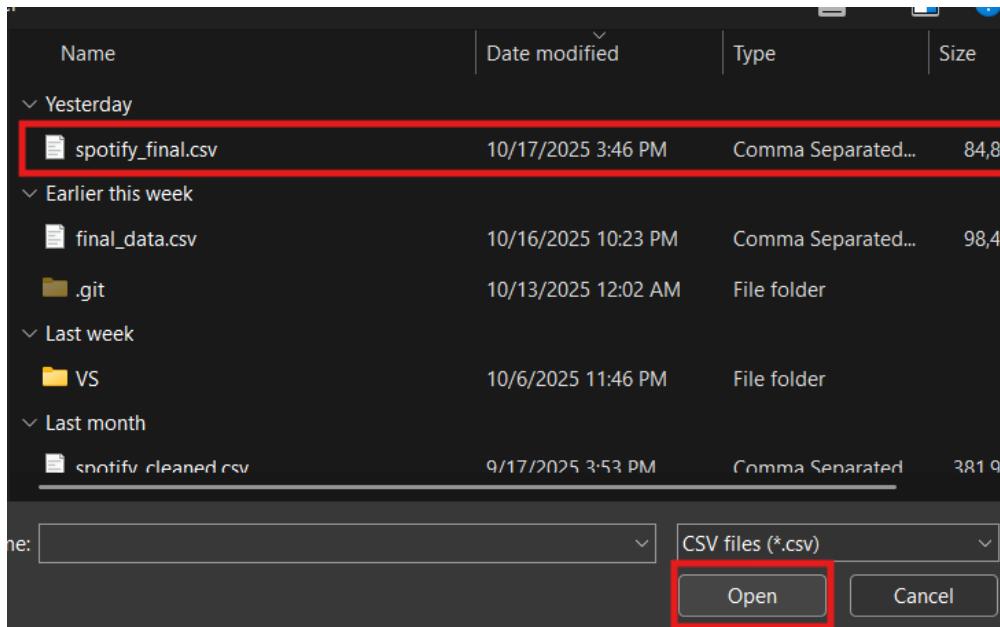
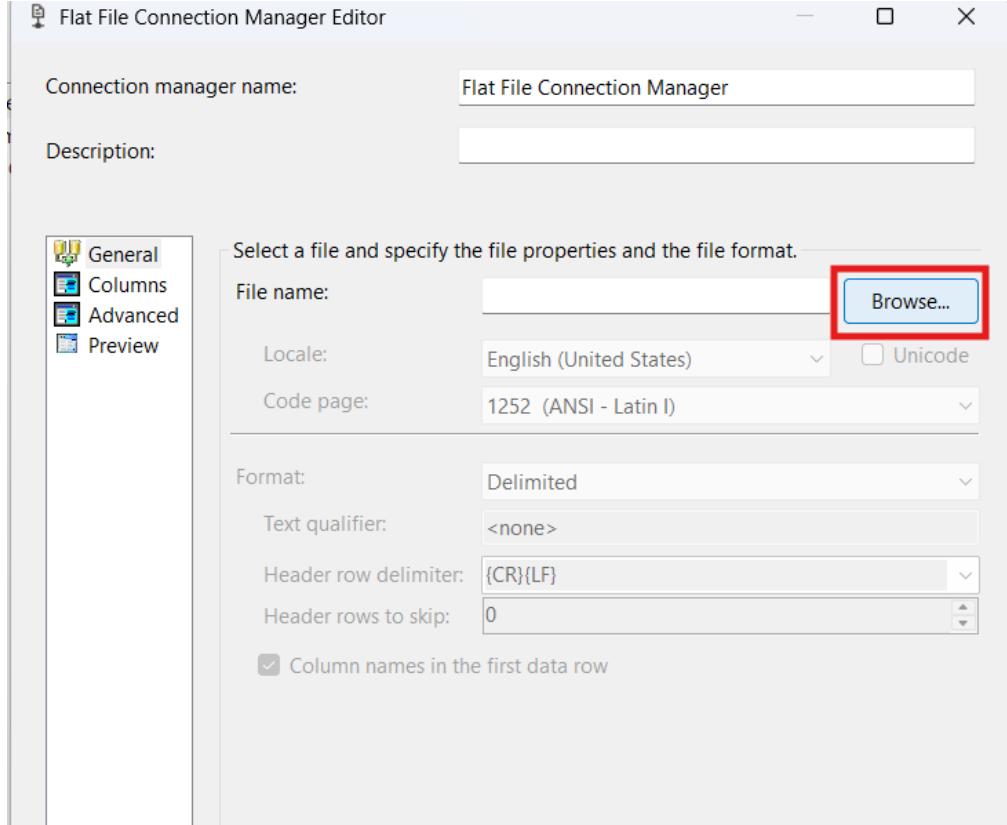
Trong Data Flow, tạo mới một “Flat File Source”.

Nháy đúp hoặc nhấn chuột phải vào “Flat File Source”, chọn “Edit” để hiển thị “Flat File Source Editor”.

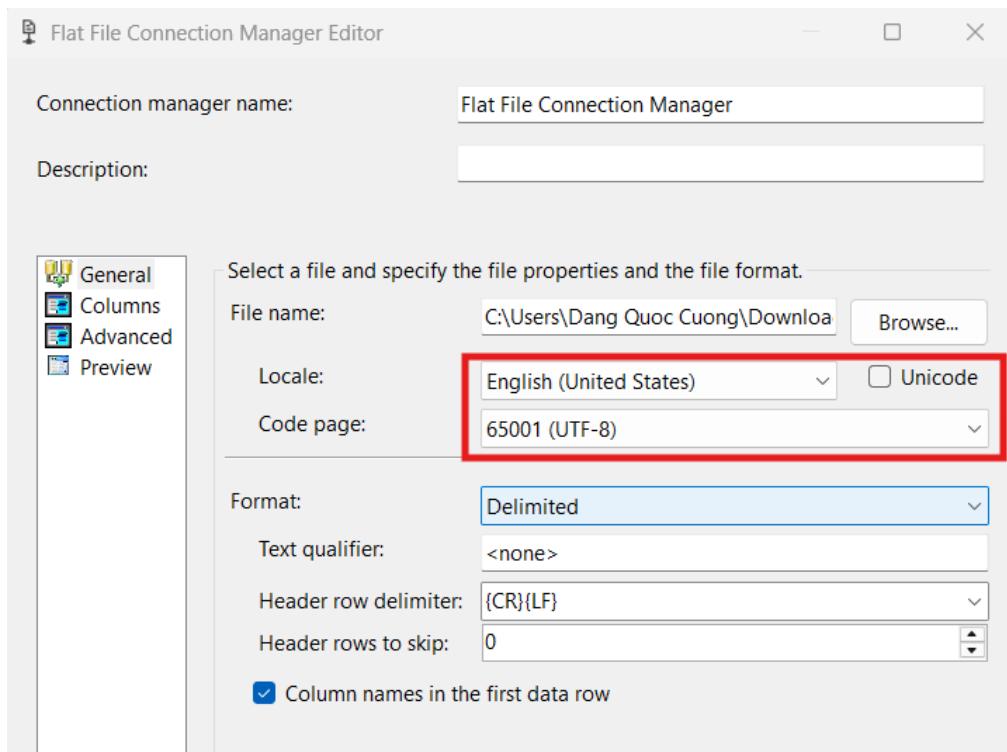
Sau đó chọn vào “New” để tạo mới một “Flat file connection”, kết nối đến file .csv chứa bộ dữ liệu.



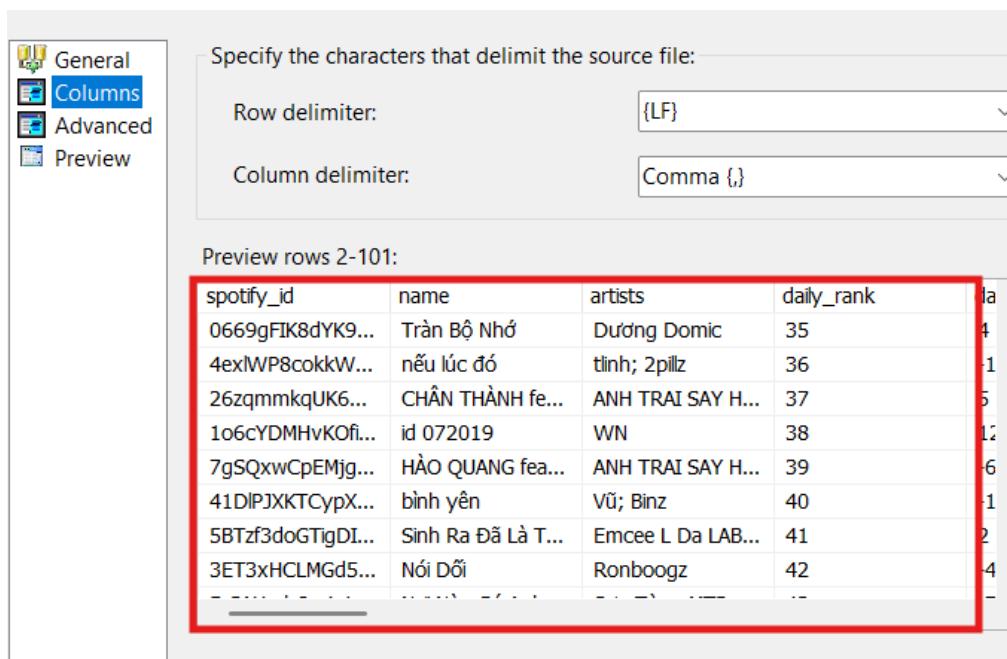
Tiếp theo, chọn “Browse”, chọn file .csv chứa bộ dữ liệu.



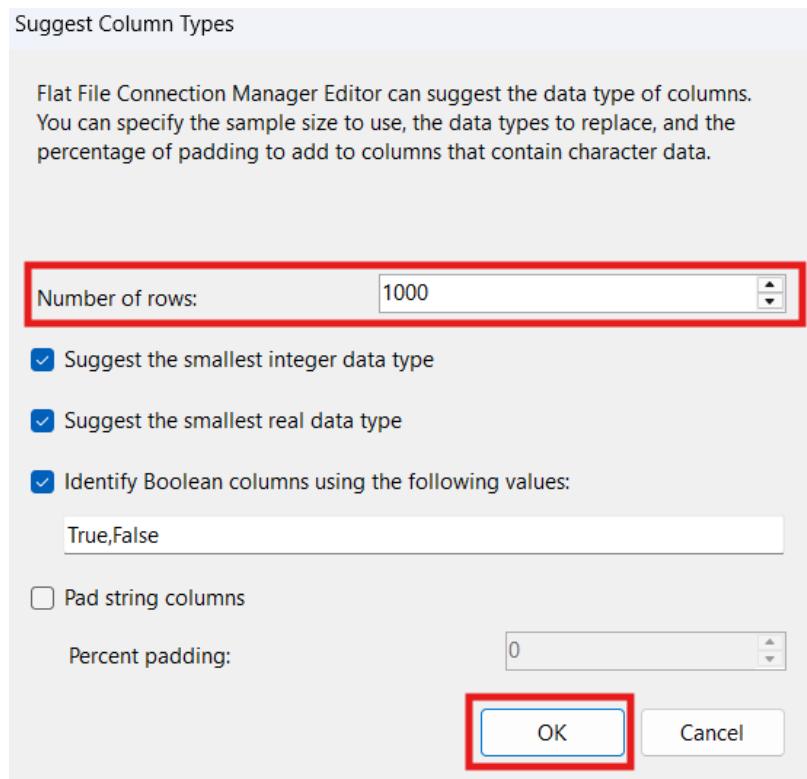
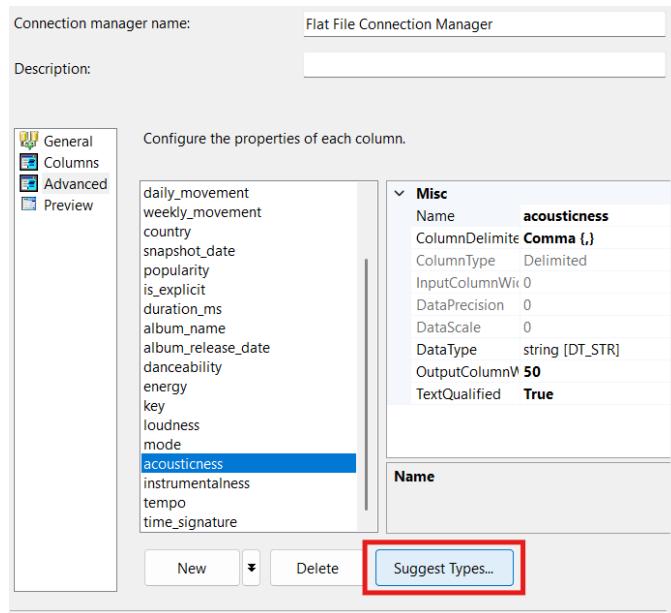
Điều chỉnh code page thành UTF-8



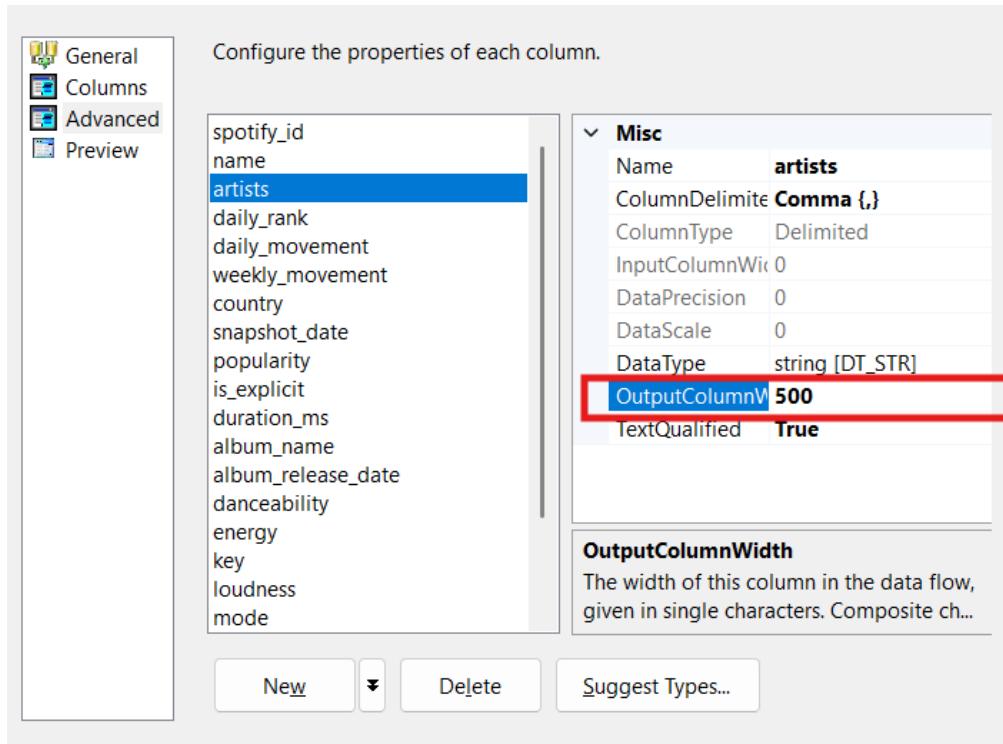
Sau đó, vào mục “Columns” để kiểm tra các cột dữ liệu.



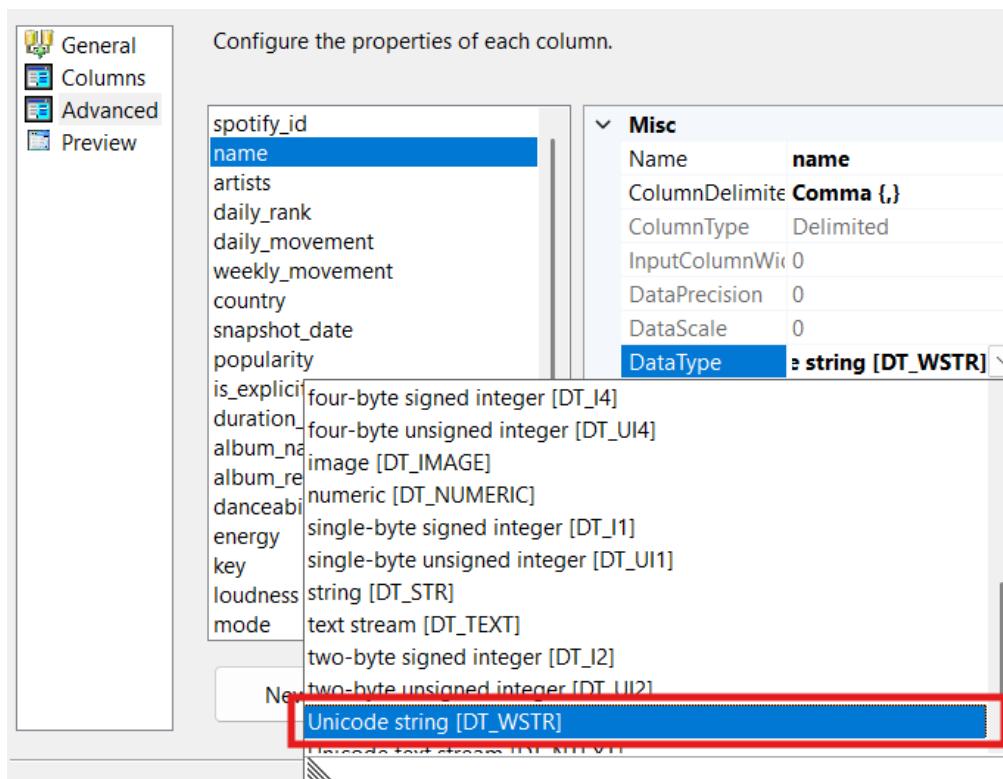
Chọn "Suggest Types" trong mục "Advanced" để đề xuất kiểu dữ liệu cho các thuộc tính.



Vẫn trong mục “Advanced” để thay đổi kích thước của các cột dữ liệu kiểu String để tránh mất ký tự của các dữ liệu có độ dài lớn. Sau đó nhấn “OK”



Lý do chuyển đổi từ DT_STR sang DT_WSTR



Trong quá trình xử lý dữ liệu bằng **SQL Server Integration Services (SSIS)**, việc lựa chọn đúng kiểu dữ liệu chuỗi (**string data type**) là rất quan trọng để đảm bảo tính toàn vẹn và khả năng xử lý của hệ thống. Ban đầu, các cột trong **Flat File Source** thường được ánh xạ thành kiểu **DT_STR** (Non-Unicode String). Tuy nhiên, trong thực tế triển khai, việc chuyển đổi sang kiểu **DT_WSTR** (Unicode String) là cần thiết vì các lý do sau:

- **Hỗ trợ ký tự đa ngôn ngữ (Unicode):** Kiểu **DT_STR** phụ thuộc vào **code page** của hệ thống (ví dụ: 1252 cho Latin hoặc 65001 cho UTF-8) và chỉ biểu diễn được các ký tự ASCII hoặc Latin mở rộng. Trong khi đó, kiểu **DT_WSTR** lưu trữ dữ liệu theo chuẩn **UTF-16**, có khả năng biểu diễn đầy đủ các ký tự tiếng Việt, tiếng Trung, Nhật, Hàn và các ngôn ngữ khác mà không bị mất dấu hay sai mã hóa.
- **Đảm bảo tính tương thích khi xử lý dữ liệu:** Các thành phần trong SSIS như **Sort**, **Merge Join**, **Lookup** và **Conditional Split** thường yêu cầu dữ liệu chuỗi ở dạng Unicode để thực hiện so sánh và sắp xếp chính xác. Nếu sử dụng **DT_STR**, SSIS sẽ tự động ép kiểu sang **DT_WSTR** trong quá trình chạy, dẫn đến chi phí xử lý và cảnh báo chuyển đổi ngầm.
- **Tránh lỗi khi nạp dữ liệu vào cơ sở dữ liệu đích:** Trong SQL Server, các cột kiểu **NVARCHAR/NCHAR** tương ứng với dữ liệu Unicode. Nếu dữ liệu trong SSIS vẫn ở dạng **DT_STR**, việc nạp vào bảng đích có thể gây lỗi: "**Cannot convert between Unicode and non-Unicode string data types.**" Chuyển đổi sang **DT_WSTR** giúp đồng nhất với kiểu **NVARCHAR** trong cơ sở dữ liệu, đảm bảo quá trình tải dữ liệu diễn ra suôn sẻ.
- **Giữ nguyên độ chính xác khi đọc file UTF-8:** Dù **DT_STR** có thể đọc đúng dữ liệu khi đặt **code page** là 65001 (UTF-8), nhưng SSIS nội bộ không thật sự xử lý UTF-8 mà chỉ giải mã tạm thời. Việc chuyển sang **DT_WSTR** cho phép dữ liệu được lưu trữ dưới dạng Unicode thực sự trong pipeline, giúp tránh lỗi hiển thị hoặc so sánh sai ký tự.

Tóm lại, việc chuyển đổi từ **DT_STR** sang **DT_WSTR** không chỉ đảm bảo tính chính xác của dữ liệu ngôn ngữ mà còn tăng tính ổn định, giảm lỗi chuyển đổi ngầm và nâng cao hiệu suất xử lý trong các gói SSIS phức tạp.

Cuối cùng, tạo Multicast:

- Kéo **Multicast** vào Data Flow.
- Kết nối mũi tên từ **Flat File Source** → **Multicast**.

Đây là **node trung gian** giúp bạn chia dữ liệu ra nhiều nhánh cho từng bảng sau này.



2.3.1 Tạo bảng DimDate

DimDate	
PK	date_id
	day
	month
	year
	full_date

Bước 1: Trích xuất các cột ngày (Extract date columns)

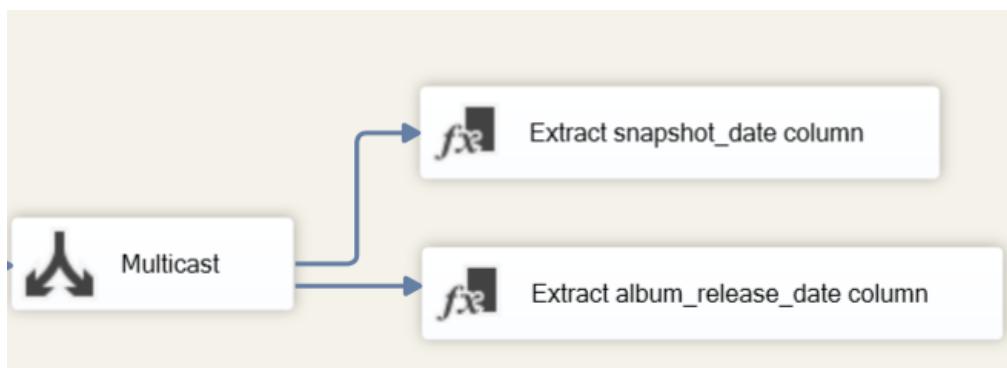
Từ **Multicast**, ta chia dữ liệu thành hai nhánh riêng biệt để trích xuất hai loại ngày: **snapshot_date** và **album_release_date**. Mục tiêu là hợp nhất các loại ngày này về cùng một cấu trúc để tạo bảng **DimDate** thống nhất.

- Nhánh A – Extract snapshot_date column:

- Component: Derived Column.
- Đặt tên: Extract snapshot_date column.
- Giữ lại duy nhất cột snapshot_date.
- Đổi tên output column thành date_value.

- Nhánh B – Extract album_release_date column:

- Sao chép cấu hình từ nhánh A.
- Giữ lại cột album_release_date.
- Đổi tên output column thành date_value.



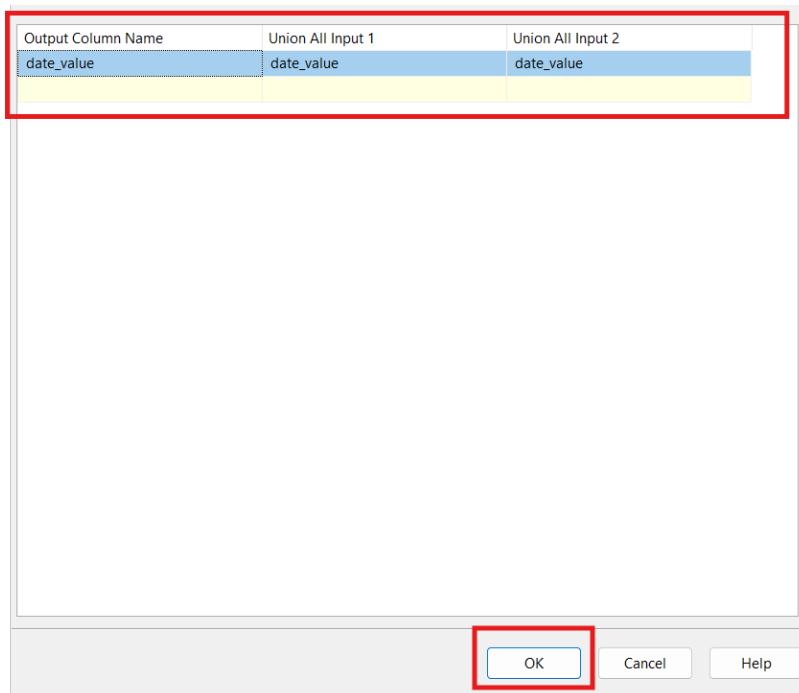
Mục đích: Chuẩn hoá hai trường ngày thành cùng tên cột `date_value` để chuẩn bị cho bước **Union All**.

Derived Column Name	Derived Column	Expression	Data Type	Length
date_value	<add as new column>	snapshot_date	date [DT_DATE]	

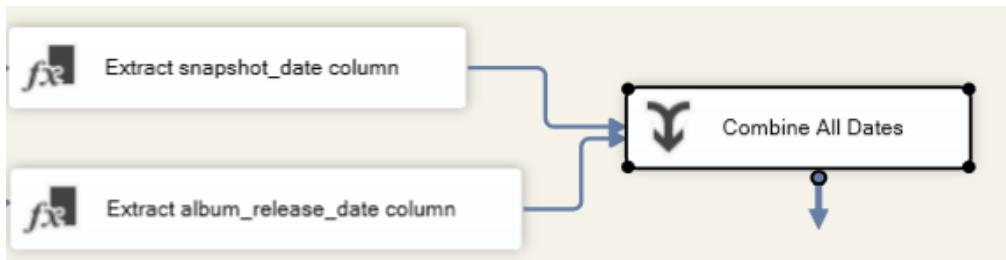
Bước 2: Hợp nhất các loại ngày (Union All)

Thực hiện hợp nhất hai nguồn dữ liệu ngày:

- Component: Union All.
- Đặt tên: Combine All Dates.
- Kết nối đầu vào từ hai nhánh: `Extract snapshot_date column` và `Extract album_release_date column`.
- Xóa bỏ các thuộc tính không cần và Output sẽ chỉ còn một cột duy nhất: `date_value`.



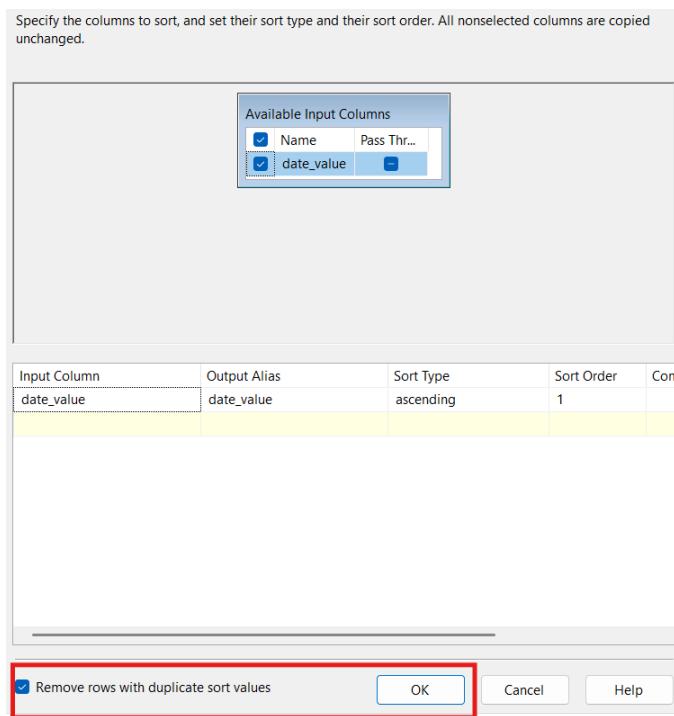
Mục đích: Gộp toàn bộ giá trị ngày (từ **snapshot** và **album**) thành một luồng dữ liệu duy nhất.



Bước 3: Loại trùng lặp (Sort / Remove Duplicates)

Để đảm bảo mỗi ngày chỉ xuất hiện một lần trong bảng **DimDate**, ta cần sắp xếp và loại bỏ các dòng trùng:

- Component: Sort.
- Đặt tên: Remove Duplicates by Date.
- Chọn cột date_value để sắp xếp.
- Tick vào tùy chọn Remove rows with duplicate sort values.



Kết quả: Mỗi giá trị ngày là duy nhất trước khi thêm các cột ngày, tháng, năm.

Bước 4: Tạo các cột dẫn xuất (Derived Column)

Từ giá trị date_value, tạo thêm các thuộc tính ngày/tháng/năm để phục vụ phân tích trong OLAP.

- Component: Derived Column.
- Tạo các cột sau với biểu thức SSIS:

```
full_date : (DT_DBDATE)date_value
day       : DAY(date_value)
month     : MONTH(date_value)
year      : YEAR(date_value)
```

Derived Column Name	Derived Column	Expression	Data Type	Length
day	<add as new column>	DAY(date_value)	four-byte signed integer	4
month	<add as new column>	MONTH(date_value)	four-byte signed integer	4
year	<add as new column>	YEAR(date_value)	four-byte signed integer	4
full_date	<add as new column>	(DT_DBDATE)date_value	database date [DT_DATE]	8

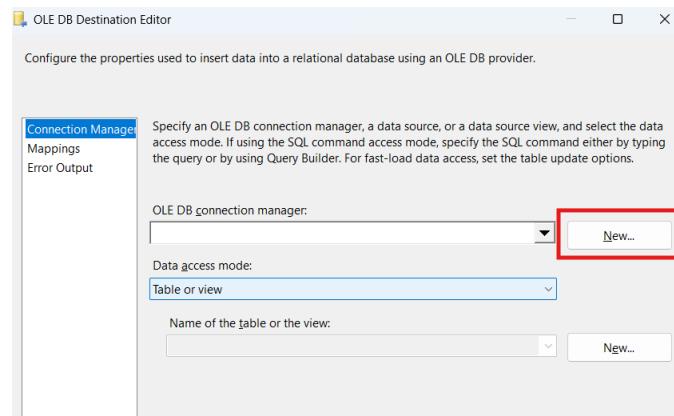
Mục đích: Chuẩn hóa dữ liệu ngày để tạo kho dữ liệu phân tích đa chiều (Dim_Date).

Bước 5: Nạp vào cơ sở dữ liệu (OLE DB Destination)

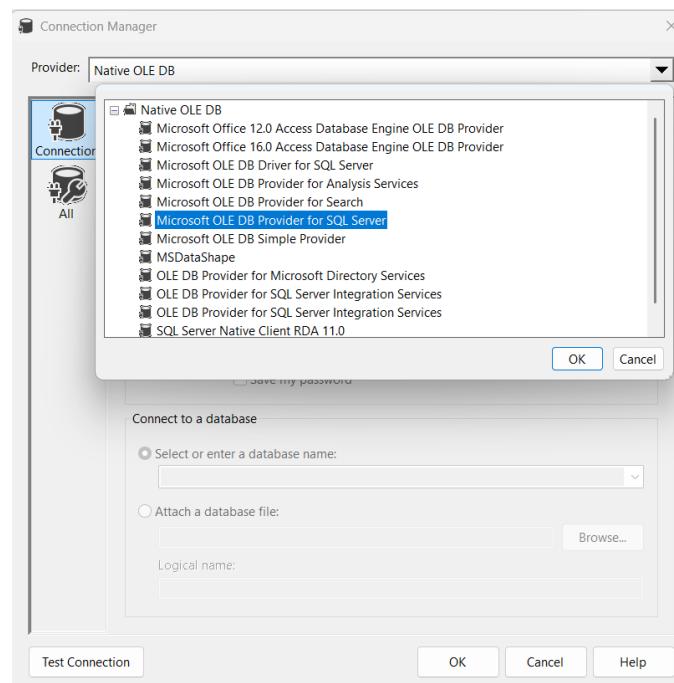
Sau khi có đầy đủ các trường cần thiết, dữ liệu được ghi vào SQL Server:

- Component: OLE DB Destination.
- Đặt tên: Load to DimDate.
- Connection: DPLAYERGOD\SQLEXPRESS.TRENDING_SONGS_SSIS.
- Data access mode: Table or view - fast load.
- Bảng đích: Dim_Date.

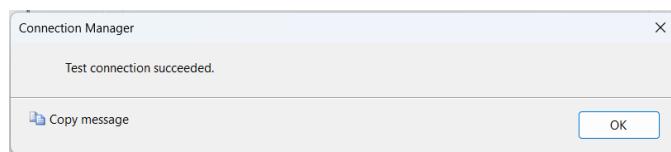
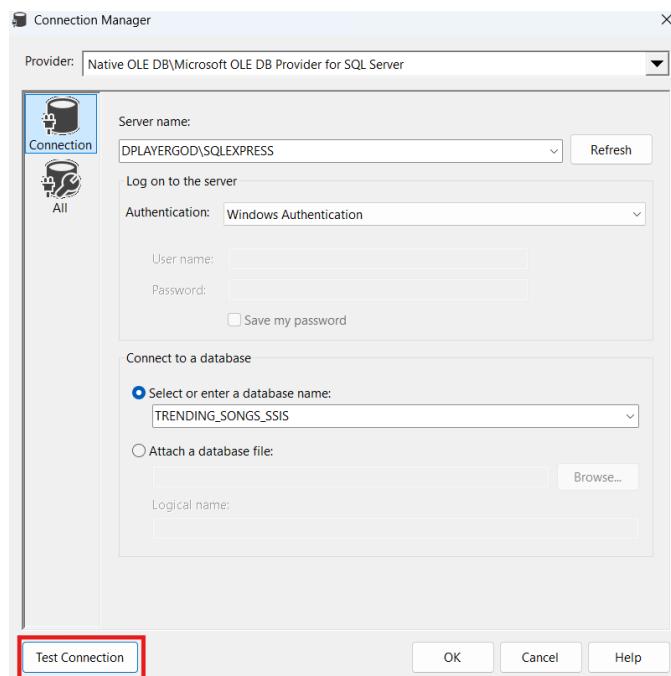
Tạo DimDate từ một OLE DB Destination. Double click vào OLE DB Destination này để tạo một connection mới đến MS SQL Server.



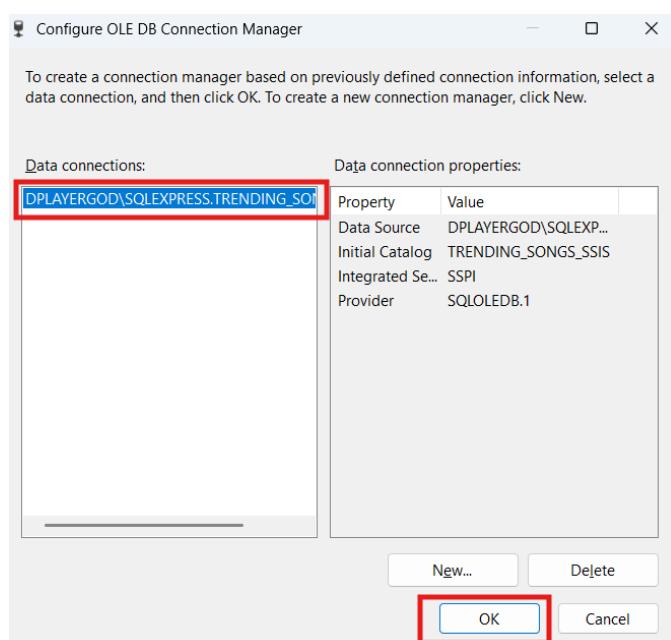
Tiếp tục chọn **New...** để tạo một connection mới:



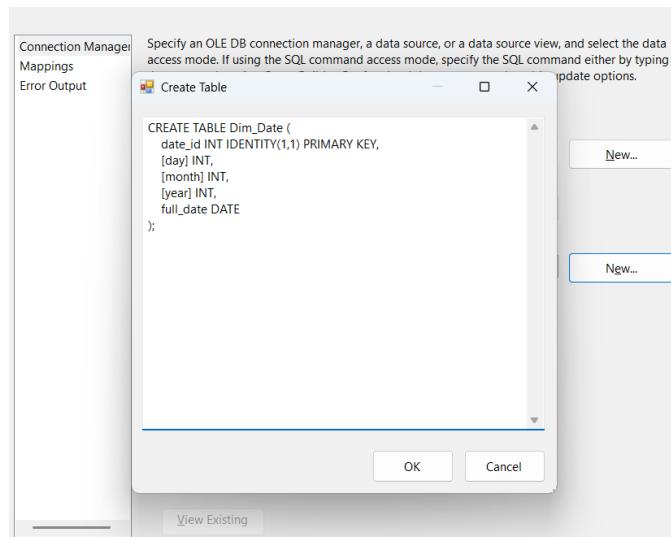
Tiếp theo, chọn tên server name trùng với server name MS SQL Server để ta có thể kết nối đến datawarehouse **TRENDING_SONGS_SSIS**. Kết nối đến server bằng tài khoản window mặc định (**Windows Authentication**).



Chọn **connection** vừa tạo đến MS SQL Server và nhấn **OK**.



Tiếp theo, chọn “New” để tạo bảng DimDate mới trong Database.



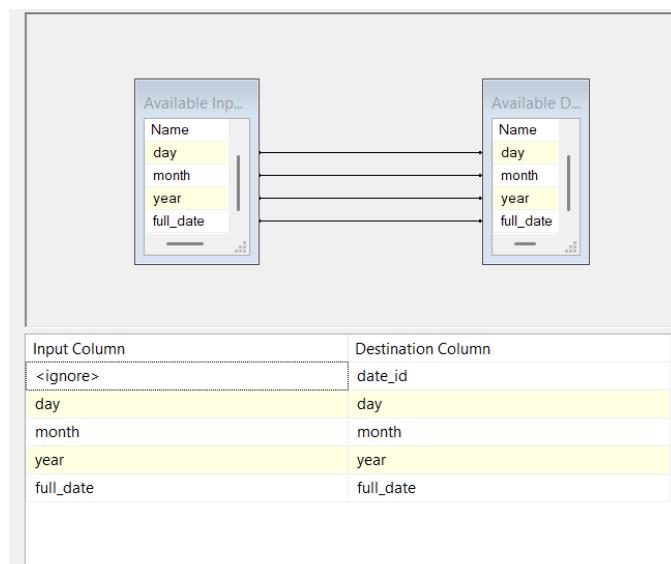
Cấu trúc bảng SQL:

```
CREATE TABLE DimDate (
    date_id INT IDENTITY(1,1) PRIMARY KEY,
    [day] INT,
    [month] INT,
    [year] INT,
    full_date DATE
);
```

Ghi chú: Nếu bảng đã tồn tại, có thể thêm một **Execute SQL Task** trước Data Flow Task với lệnh:

```
TRUNCATE TABLE DimDate;
```

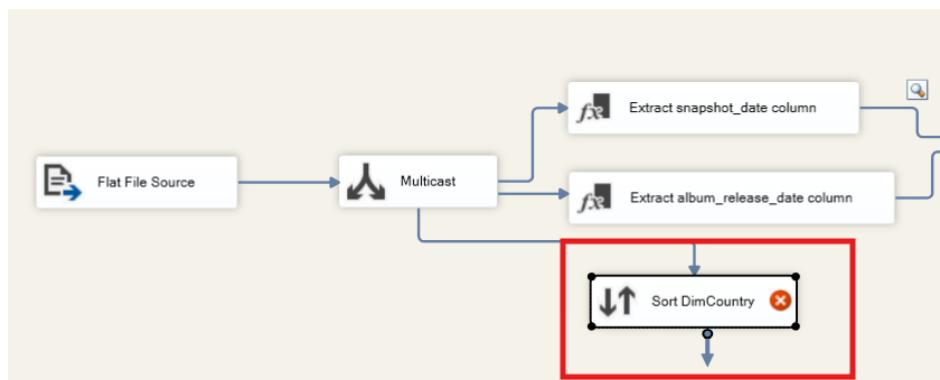
Cuối cùng, vào mục “Mappings” để kiểm tra việc ánh xạ các cột dữ liệu.



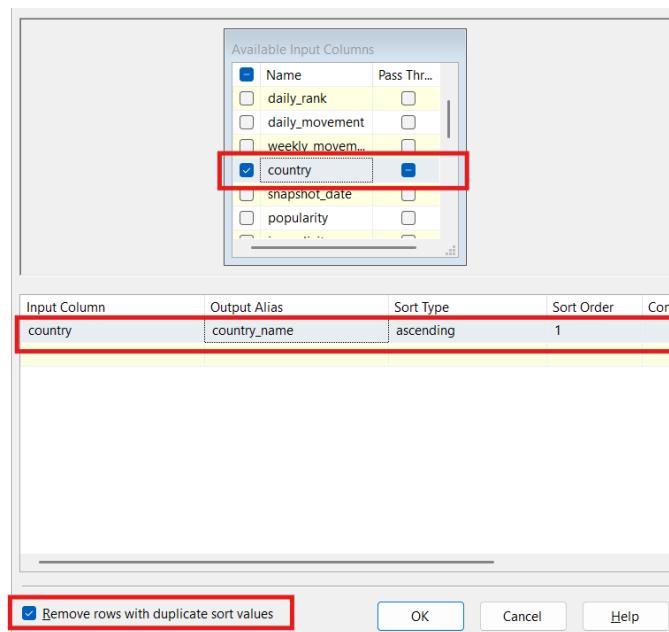
2.3.2 Tạo bảng DimCountry

DimCountry	
* PK	country_id
	country_name

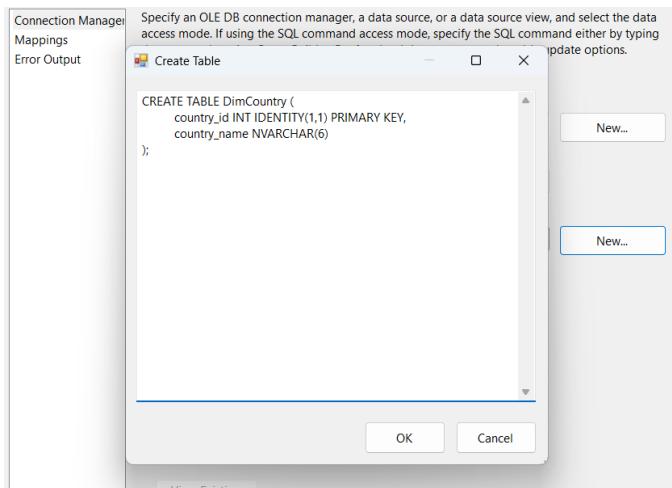
Bước 1: Tạo mới một Sort Transformation có tên Sort DimCountry để lấy ra cột dữ liệu cần thiết cho DimCountry. Nhấn chuột phải và chọn Edit → trong mục Available Input Columns, chọn country làm cột dữ liệu cho Sort DimCountry và đổi tên thành country_name.



Sau đó tích chọn Remove rows with duplicate sort values để loại bỏ các dòng trùng lặp, rồi chọn OK.



Bước 2: Tạo một **OLE DB Destination** có tên **Load to DimCountry**. Trong phần **Connection Manager**, chọn kết nối tới cơ sở dữ liệu đích, nhấn **New...** để tạo mới bảng.



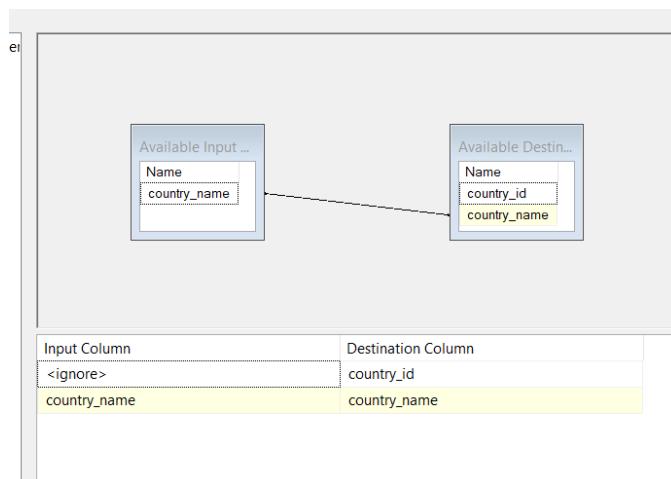
Cấu trúc bảng SQL:

```
CREATE TABLE DimCountry (
    country_id INT IDENTITY(1,1) PRIMARY KEY,
    country_name NVARCHAR(6)
);
```

Ghi chú: Nếu bảng đã tồn tại, có thể thêm một **Execute SQL Task** trước Data Flow Task với lệnh:

```
TRUNCATE TABLE DimCountry;
```

Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.

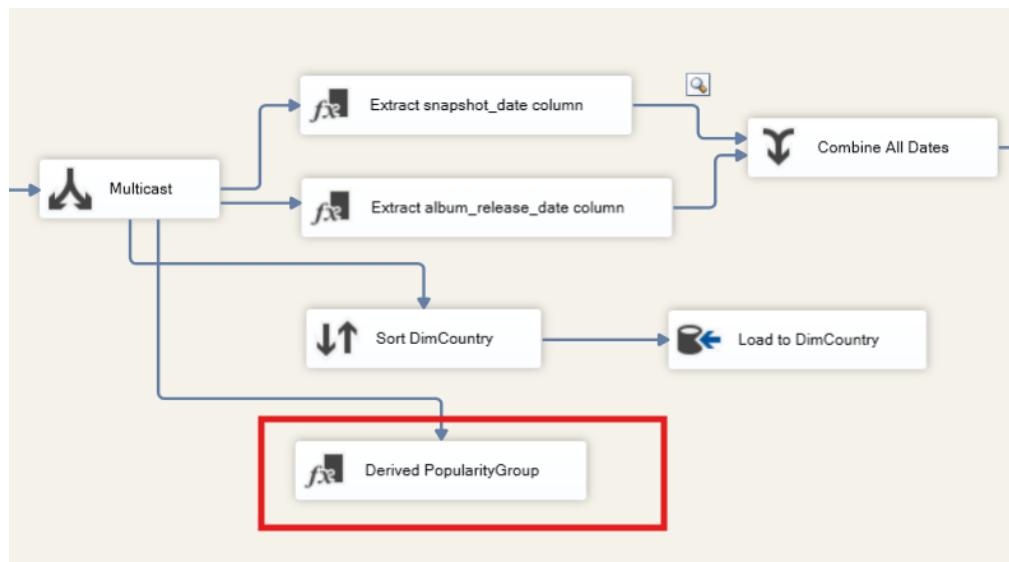


2.3.3 Tạo bảng DimPopularityGroup

DimPopularityGroup	
PK	<u>popularity_group_id</u>
	group_name
	min_popularity
	max_popularity

Bước 1: Thêm một **Derived Column Transformation** có tên Derived PopularityGroup. Trong phần **Derived Column Name**, tạo ba cột mới:

- **group_name**: phân loại mức độ phổ biến của bài hát.
- **min_popularity** và **max_popularity**: biểu thị ngưỡng giá trị tương ứng cho từng nhóm.



Biểu thức được thiết lập trong phần **Expression** như sau:

```

group_name =
  (popularity <= 40) ? "Low" :
  (popularity <= 70) ? "Medium" :
  "High"
  
```

```

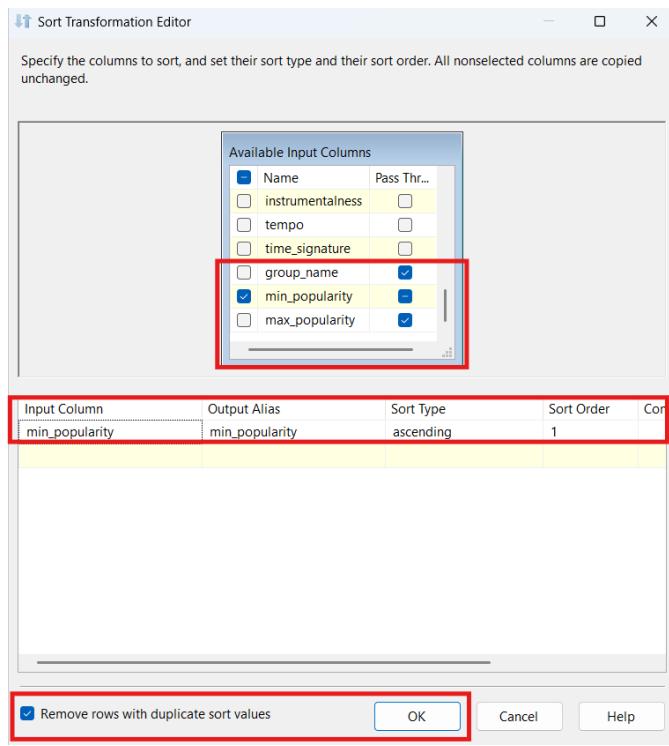
min_popularity =
  
```

```
(popularity <= 40) ? 0 :  
(popularity <= 70) ? 41 :  
71
```

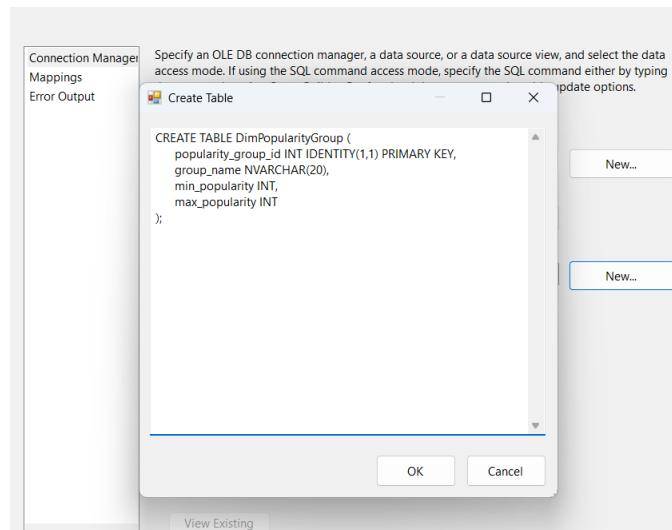
```
max_popularity =  
(popularity <= 40) ? 40 :  
(popularity <= 70) ? 70 :  
100
```

Derived Column Name	Derived Column	Expression	Data Type	Length
group_name	<add as new column>	(popularity <= 40) ? "Low" :(popularity <= 70) ? "Medium" : "High"	Unicode string [DT_WSTR]	6
min_popularity	<add as new column>	(popularity <= 40) ? 0 :(popularity <= 70) ? 41 : 100	four-byte signed integer [DT_I4]	4
max_popularity	<add as new column>	(popularity <= 40) ? 40 :(popularity <= 70) ? 70 : 100	four-byte signed integer [DT_I4]	4

Bước 3: Thêm một Sort Transformation có tên Sort DimPopularityGroup. Trong phần Available Input Columns, chọn các cột group_name, min_popularity và max_popularity đầu ra và chọn min_popularity để sort. Sau đó tích chọn Remove rows with duplicate sort values để loại bỏ các dòng trùng lặp, giữ lại duy nhất ba dòng dữ liệu đại diện cho ba nhóm độ phổ biến.



Bước 4: Tạo một OLE DB Destination có tên Load to DimPopularityGroup. Trong phần Connection Manager, chọn kết nối tới cơ sở dữ liệu đích, nhấn New... để tạo mới bảng DimPopularityGroup.



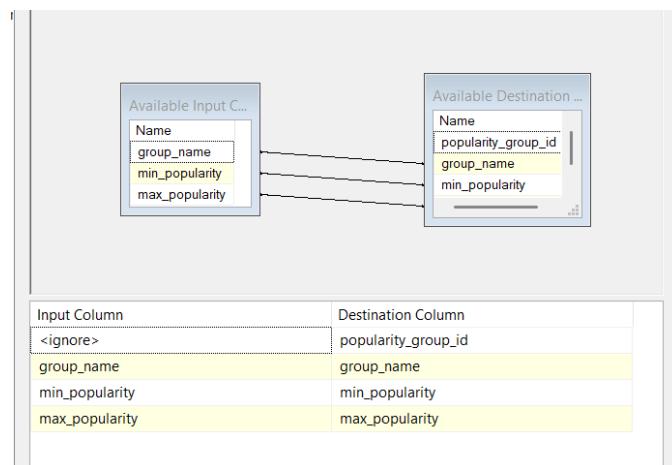
Cấu trúc bảng SQL:

```
CREATE TABLE DimPopularityGroup (
    popularity_group_id INT IDENTITY(1,1) PRIMARY KEY,
    group_name NVARCHAR(20),
    min_popularity INT,
    max_popularity INT
);
```

Ghi chú: Nếu bảng đã tồn tại, có thể thêm một **Execute SQL Task** trước Data Flow Task với lệnh:

```
TRUNCATE TABLE DimPopularityGroup;
```

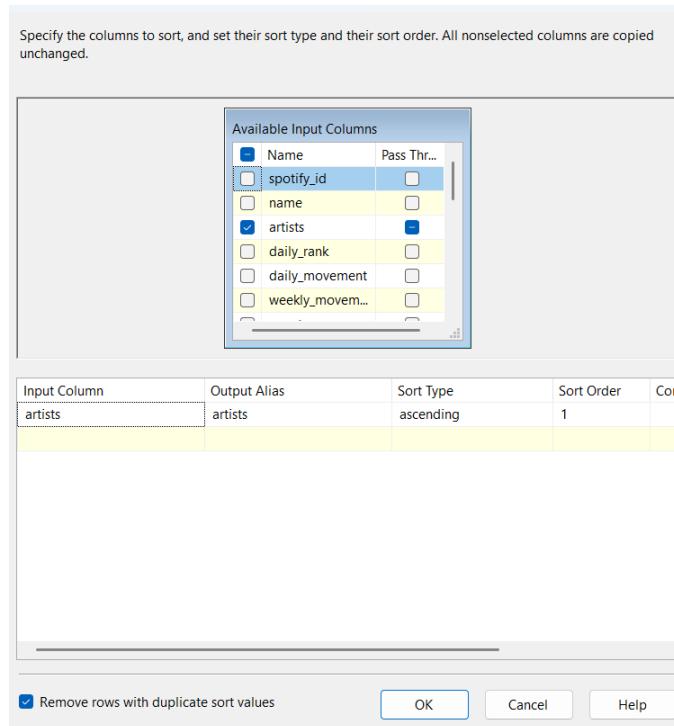
Cuối cùng, vào mục “Mappings” để kiểm tra việc ánh xạ các cột dữ liệu.



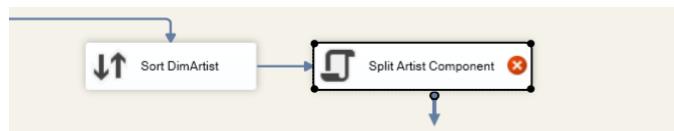
2.3.4 Tạo bảng DimArtist

DimArtist	
PK	<u>artist_id</u>
	artist_name

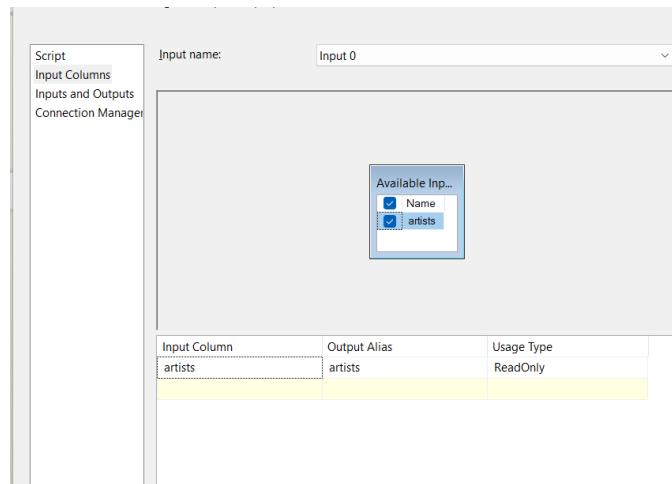
Bước 1: Tạo mới một Sort Transformation có tên Sort DimArtist để lấy ra cột dữ liệu cần thiết cho bảng DimArtist. Trong phần Available Input Columns, chọn cột artists từ nguồn dữ liệu (Multicast). Cột này chứa nhiều tên nghệ sĩ được phân tách bằng dấu ;.



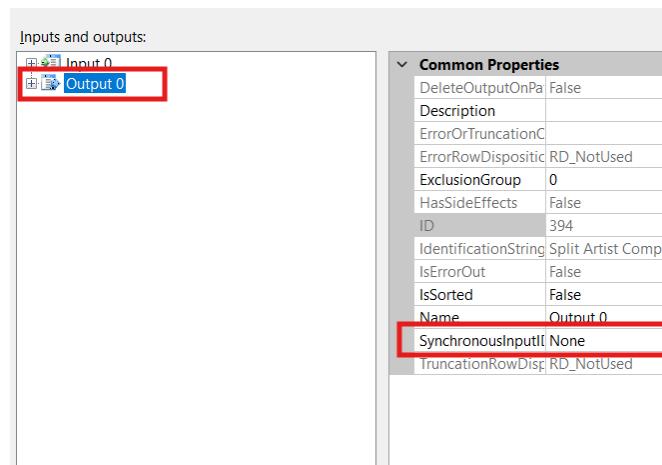
Bước 2: Thêm một Script Component và chọn chế độ Transformation. Đặt tên là Split Artist Component. Thành phần này dùng để tách giá trị trong cột artists thành nhiều dòng riêng biệt.



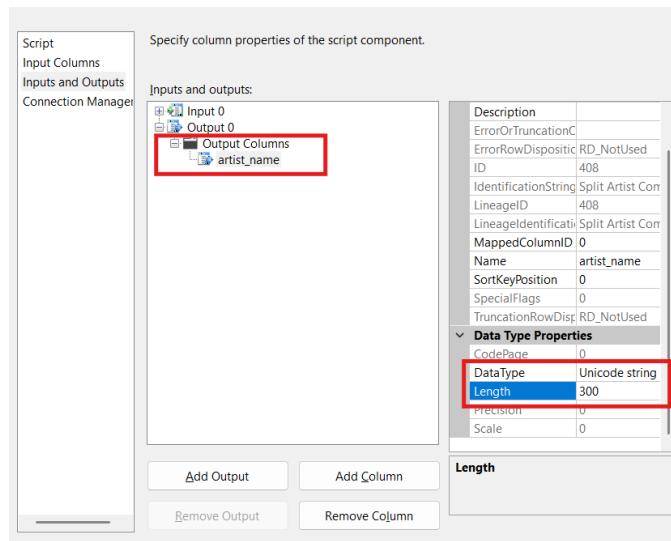
Trong cửa sổ Input Columns, chọn cột artists (ReadOnly = True).



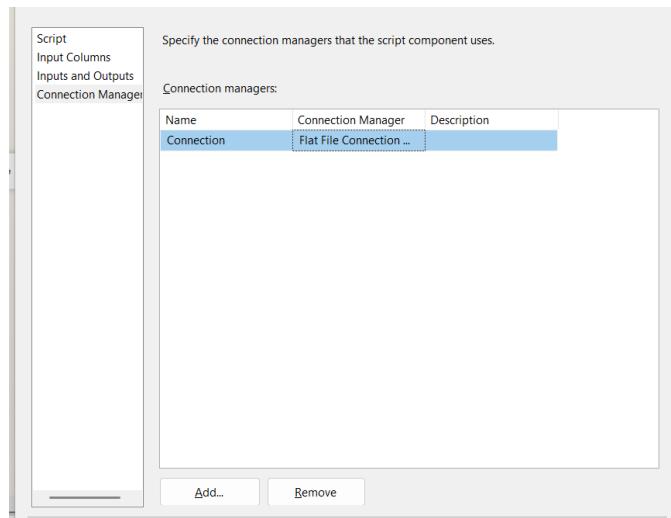
Vào mục “**Inputs and Outputs**”, trong mục “**Common Properties**” đổi **SynchronousInputID** thành **None**



Sau đó, chuyển sang tab **Inputs and Outputs** → chọn **Output Columns** và thêm một cột mới có tên **artist_name** (kiểu DT_WSTR(300)).



Vào mục “**Connection Managers**” để tạo kết nối đến dữ liệu. Nhấn vào “**Add**” để thêm Connection. Trong cột “**Connection Manager**”, chọn **Flat File Connection** đã kết nối với dữ liệu trước đó.



Vào lại mục **Script**, chọn **Edit Script...** để thêm đoạn mã thực hiện việc tách chuỗi. Sử dụng hàm **Split()** để tách giá trị trong cột **artists** thành các phần tử riêng biệt, các phần tử được phân tách bởi dấu ‘;’. Kết quả của phép tách được lưu vào mảng **artistList**.

Tiếp theo, duyệt qua từng phần tử trong mảng và sử dụng hàm **AddRow()** để thêm một dòng mới vào **Output0Buffer**. Mỗi phần tử sẽ được gán vào thuộc tính **artist_name**, đồng thời sử dụng hàm **Trim()** để loại bỏ các khoảng trắng thừa.

```

    ↴ references
public override void Input0_ProcessInputRow(Input0Buffer Row)
{
    if (!Row.artists_IsNull)
    {
        string[] artistList = Row.artists.Split(';');
        foreach (string artist in artistList)
        {
            string trimmed = artist.Trim();
            if (!string.IsNullOrEmpty(trimmed))
            {
                Output0Buffer.AddRow();
                Output0Buffer.artistname = trimmed;
            }
        }
    }
}

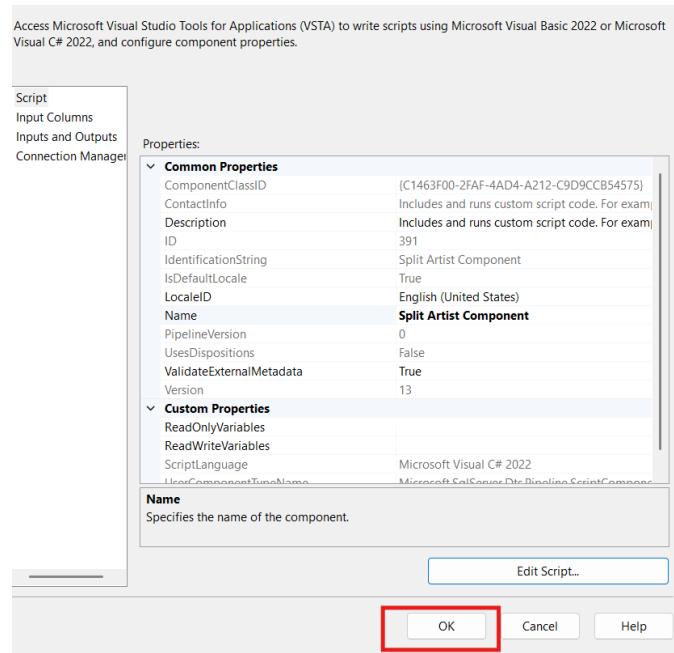
```

```

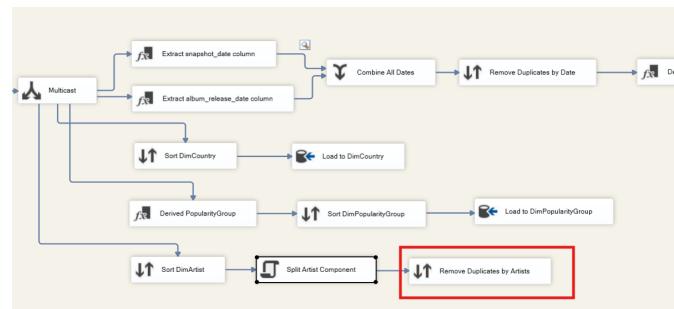
public override void Input0_ProcessInputRow(Input0Buffer Row)
{
    if (!Row.artists_IsNull)
    {
        string[] artistList = Row.artists.Split(';');
        foreach (string artist in artistList)
        {
            string trimmed = artist.Trim();
            if (!string.IsNullOrEmpty(trimmed))
            {
                Output0Buffer.AddRow();
                Output0Buffer.artistname = trimmed;
            }
        }
    }
}

```

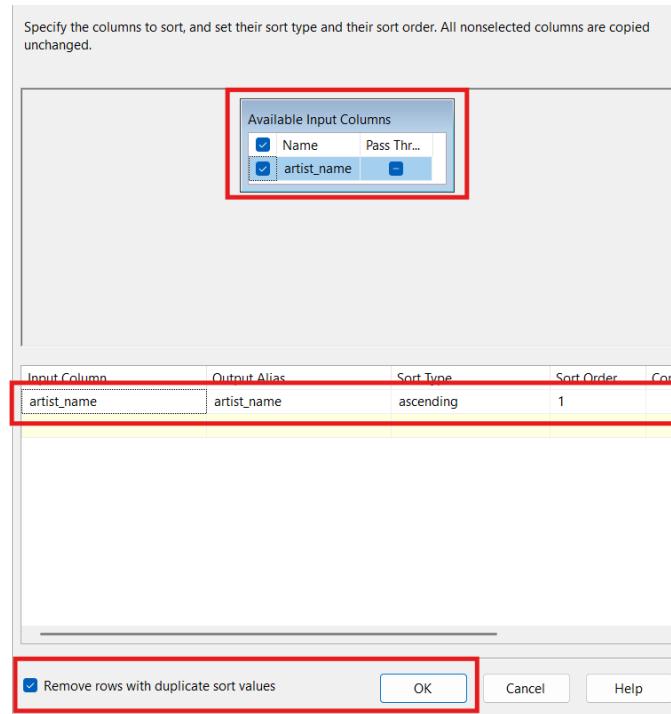
Lưu Script, chọn “OK” để hoàn tất.



Bước 3: Sau khi tách, thêm một Sort Transformation có tên là "Remove Duplicates by Artists" để loại bỏ các dòng trùng lặp.

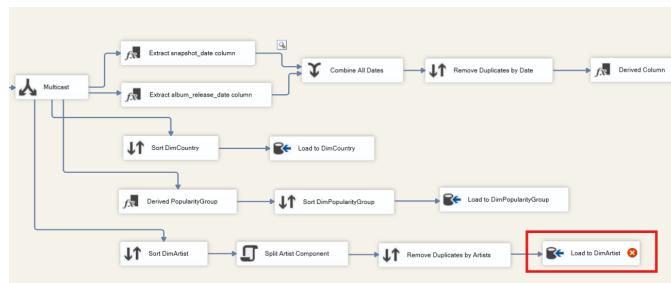


Trong phần Available Input Columns, chọn artist_name và tích chọn Remove rows with duplicate sort values.

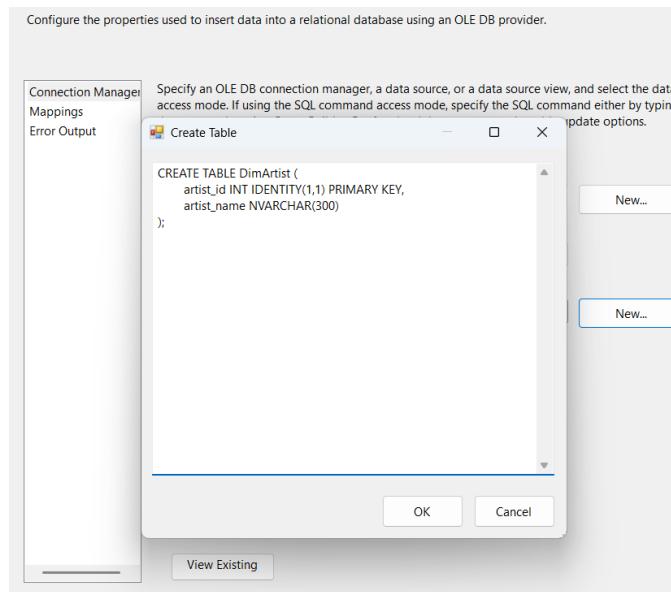


Bước này đảm bảo mỗi nghệ sĩ chỉ xuất hiện một lần trong bảng DimArtist.

Bước 4: Thêm một OLE DB Destination có tên Load to DimArtist.



Trong phần **Connection Manager**, chọn kết nối tới cơ sở dữ liệu đích và nhấn **New...** để tạo mới bảng DimArtist.



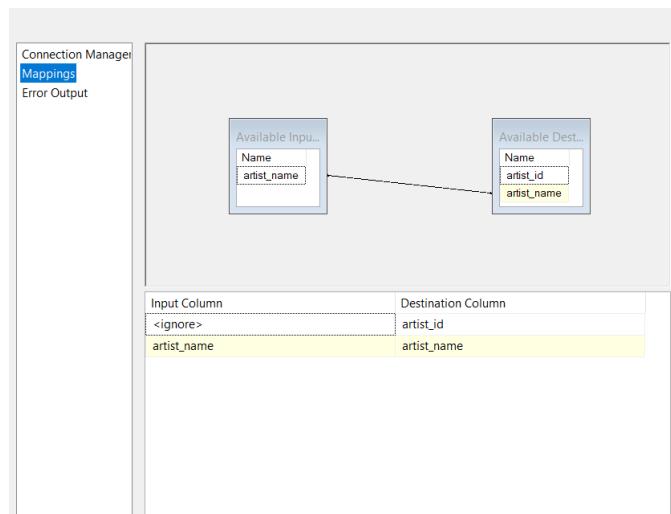
Cấu trúc bảng SQL:

```
CREATE TABLE DimArtist (
    artist_id INT IDENTITY(1,1) PRIMARY KEY,
    artist_name NVARCHAR(300)
);
```

Ghi chú: Nếu bảng đã tồn tại, có thể thêm một **Execute SQL Task** trước Data Flow Task với lệnh:

```
TRUNCATE TABLE DimArtist;
```

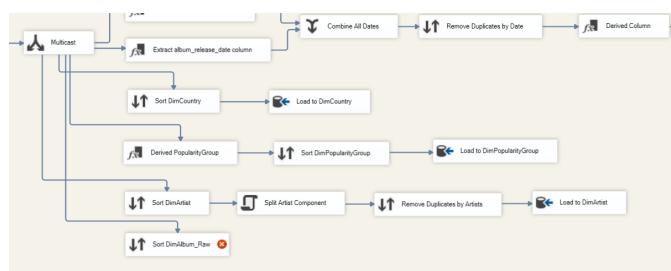
Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.



2.3.5 Tạo bảng DimAlbum _ Raw

Mục đích: Bảng DimAlbum_Raw lưu trữ thông tin gốc về các album từ tập dữ liệu Spotify, bao gồm tên album và ngày phát hành tương ứng. Dữ liệu trong bảng này được dùng để chuẩn hóa và tạo khóa thay thế (album_id) trong bảng DimAlbum ở giai đoạn sau. Mỗi bản ghi trong DimAlbum_Raw đại diện cho một album duy nhất, giúp loại bỏ trùng lặp và đảm bảo tính nhất quán khi liên kết giữa bài hát và album.

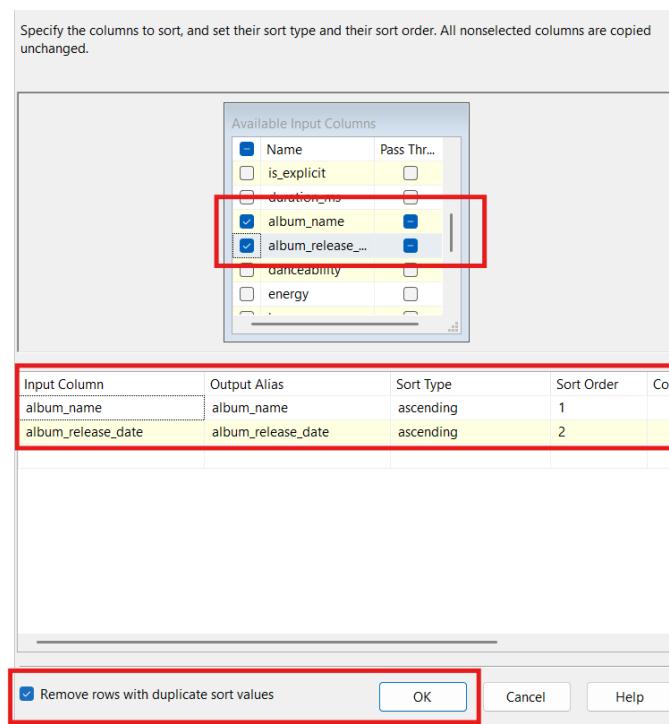
Bước 1: Từ Multicast đầu vào, tạo mới một Sort Transformation có tên Sort DimAlbum_Raw để lấy ra các cột dữ liệu cần thiết cho bảng DimAlbum _ Raw.



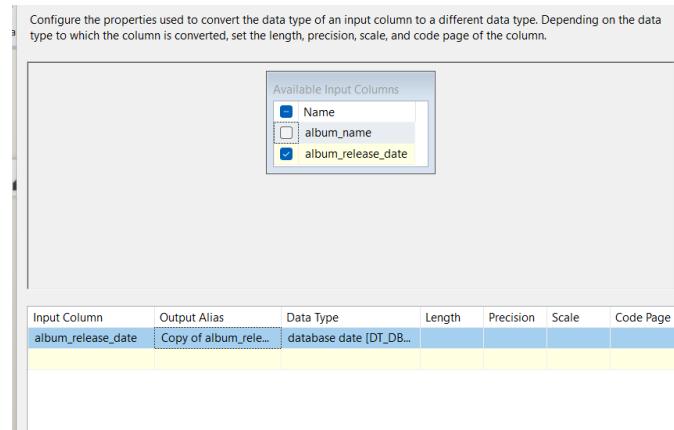
Trong phần Available Input Columns, chọn các cột:

- album_name
- album_release_date

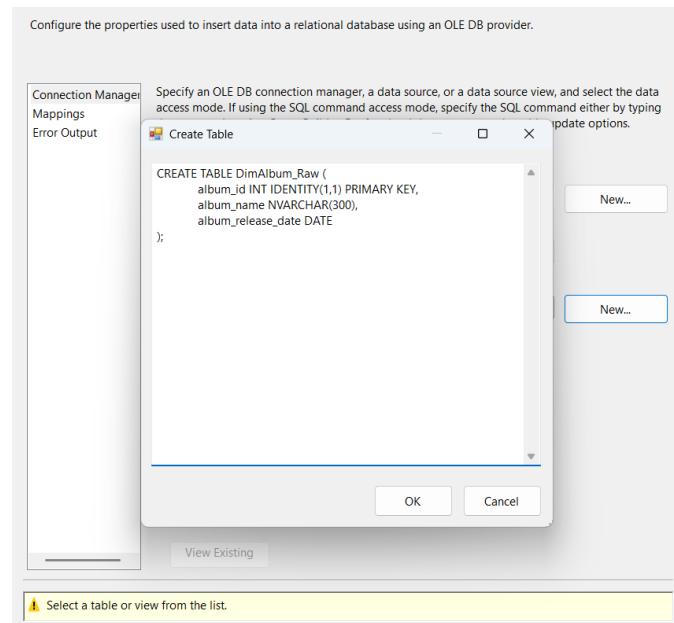
Sau đó, tích chọn Remove rows with duplicate sort values dựa trên cột album_name để loại bỏ các dòng trùng lặp, đảm bảo mỗi album chỉ xuất hiện một lần.



Bước 2: Thêm một **Data Conversion** có tên Convert Date For DimAlbum_Raw. Trong thành phần này, tiến hành **chuyển kiểu dữ liệu** của cột `album_release_date` từ kiểu DT_DATE sang DT_DBDATE để tương thích với kiểu dữ liệu trong SQL Server.



Bước 3: Thêm một **OLE DB Destination** có tên Load to DimAlbum_Raw. Trong phần **Connection Manager**, chọn kết nối đến cơ sở dữ liệu đích, sau đó chọn **New...** để tạo mới bảng DimAlbum_Raw.



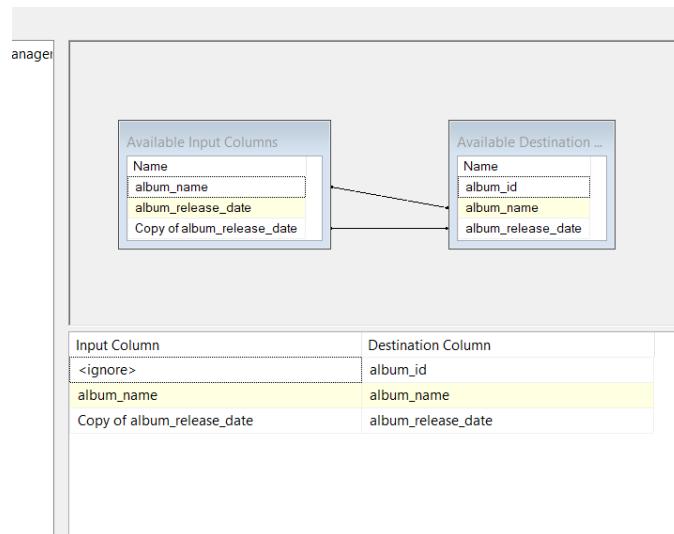
Cấu trúc bảng SQL:

```
CREATE TABLE DimAlbum_Raw (
    album_id INT IDENTITY(1,1) PRIMARY KEY,
    album_name NVARCHAR(500),
    album_release_date DATE
);
```

Ghi chú: Nếu bảng đã tồn tại, có thể thêm một **Execute SQL Task** trước Data Flow Task với lệnh:

```
TRUNCATE TABLE DimAlbum_Raw;
```

Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.



2.3.6 Tạo bảng DimSong_Raw

Mục đích: Bảng DimSong_Raw lưu trữ dữ liệu gốc của các bài hát, bao gồm thông tin nhận dạng (spotify_id, name), thuộc tính âm nhạc (như danceability, energy, tempo, loudness, v.v.), và thông tin liên quan đến album và nghệ sĩ. Mục tiêu của bảng là tập hợp và chuẩn hóa dữ liệu bài hát trước khi tạo bảng DimSong trong Data Warehouse, đảm bảo mỗi bài hát chỉ xuất hiện một lần và có thể liên kết với album, nghệ sĩ, và các fact tương ứng.

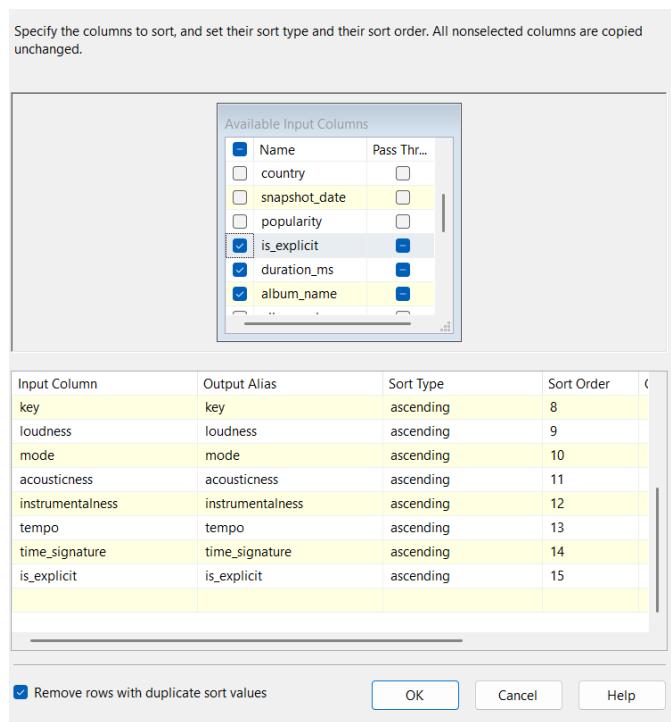
Bước 1: Từ Multicast đầu vào, tạo mới một Sort Transformation có tên Sort DimSong_Raw để lấy ra các cột dữ liệu cần thiết cho bảng DimSong_Raw.



Trong phần **Available Input Columns**, chọn các cột:

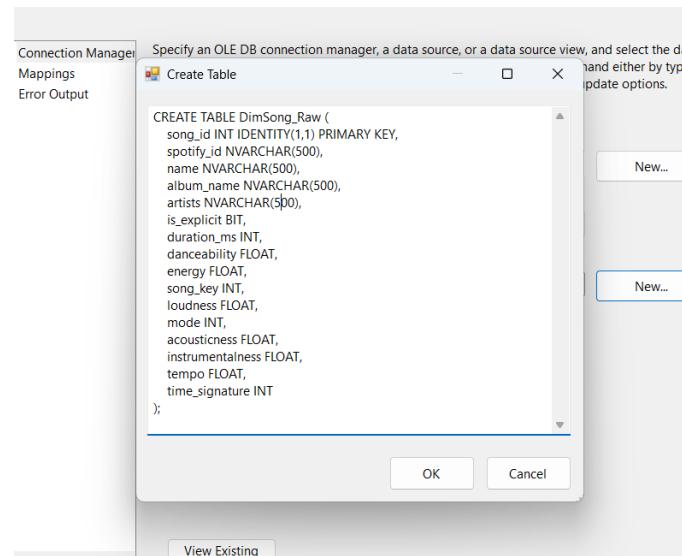
- spotify_id
- name
- album_name
- artists
- is_explicit

- duration_ms
- danceability
- energy
- key
- loudness
- mode
- acousticness
- instrumentalness
- tempo
- time_signature



Sau đó, tích chọn **Remove rows with duplicate sort values** dựa trên cột `spotify_id` để loại bỏ các bản ghi trùng lặp, đảm bảo mỗi bài hát chỉ xuất hiện một lần.

Bước 2: Thêm một **OLE DB Destination** có tên `Load To DimSong_Raw`. Trong phần **Connection Manager**, chọn kết nối đến cơ sở dữ liệu đích, sau đó chọn **New...** để tạo mới bảng `DimSong Raw`.



Cấu trúc bảng SQL:

```

CREATE TABLE DimSong_Raw (
    song_id INT IDENTITY(1,1) PRIMARY KEY,
    spotify_id NVARCHAR(500),
    name NVARCHAR(500),
    album_name NVARCHAR(500),
    artists NVARCHAR(500),
    is_explicit BIT,
    duration_ms INT,
    danceability FLOAT,
    energy FLOAT,
    [key] INT,
    loudness FLOAT,
    mode INT,
    acousticness FLOAT,
    instrumentalness FLOAT,
    tempo FLOAT,
    time_signature INT
);

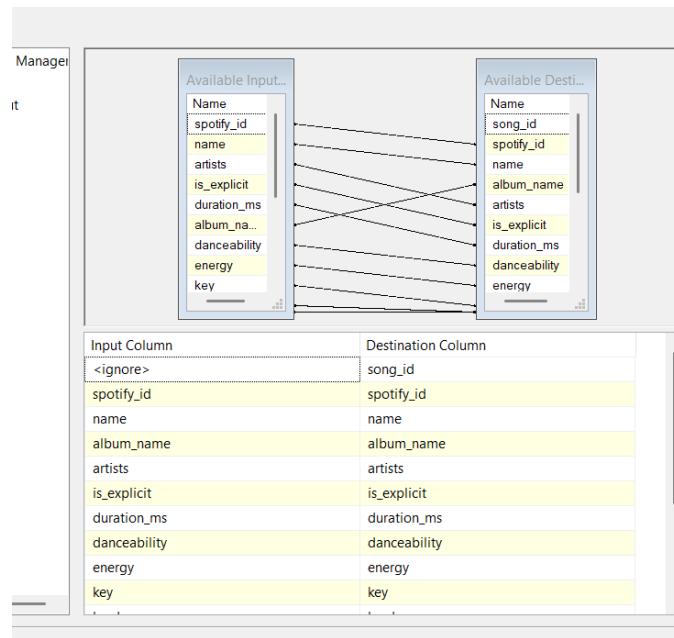
```

Ghi chú:

- Trước khi chạy lại gói SSIS, có thể thêm **Execute SQL Task** với lệnh:

```
TRUNCATE TABLE DimSong_Raw;
```

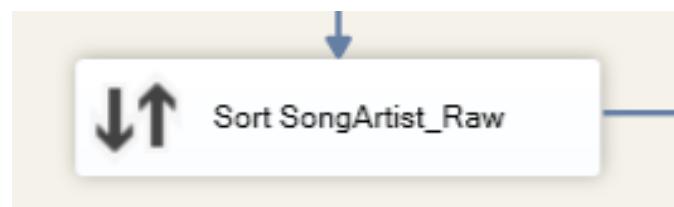
Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.



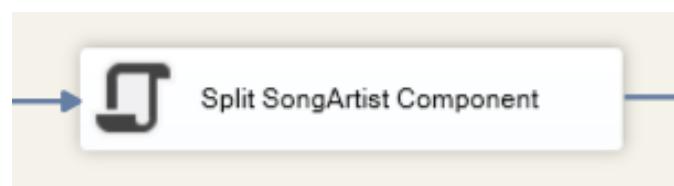
2.3.7 Tạo bảng SongArtist_Raw

Mục đích: Bảng SongArtist_Raw đóng vai trò là bảng cầu tạm thời giữa bài hát và nghệ sĩ. Trong tập dữ liệu Spotify, một bài hát có thể được thể hiện bởi nhiều nghệ sĩ (ví dụ: "Alan Walker; Sasha Alex Sloan"). Bảng này được sử dụng để tách dữ liệu đó thành nhiều dòng riêng biệt, mỗi dòng thể hiện một mối quan hệ giữa một bài hát (song_name) và một nghệ sĩ cụ thể (artist_name).

Bước 1: Từ Multicast đầu vào, tạo một Sort Transformation có tên Sort SongArtist_Raw để xây dựng bảng SongArtist_Raw. Chọn các cột name (sau đó đổi thành song_name và artists. Cột artists chứa danh sách các nghệ sĩ được phân tách bằng dấu ;.



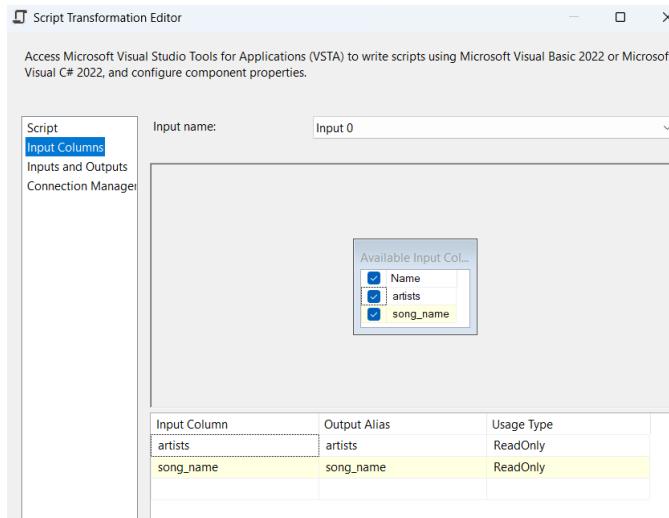
Bước 2: Thêm một Script Component và chọn chế độ Transformation. Đặt tên là Split SongArtist Component.



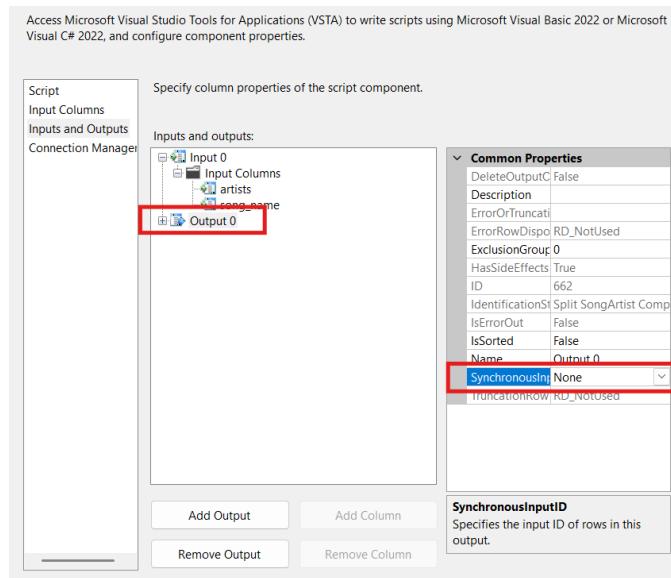
Thành phần này sẽ tách danh sách nghệ sĩ trong cột artists thành nhiều dòng riêng biệt, mỗi dòng tương ứng với một nghệ sĩ trong bài hát.

Trong cửa sổ Input Columns, chọn:

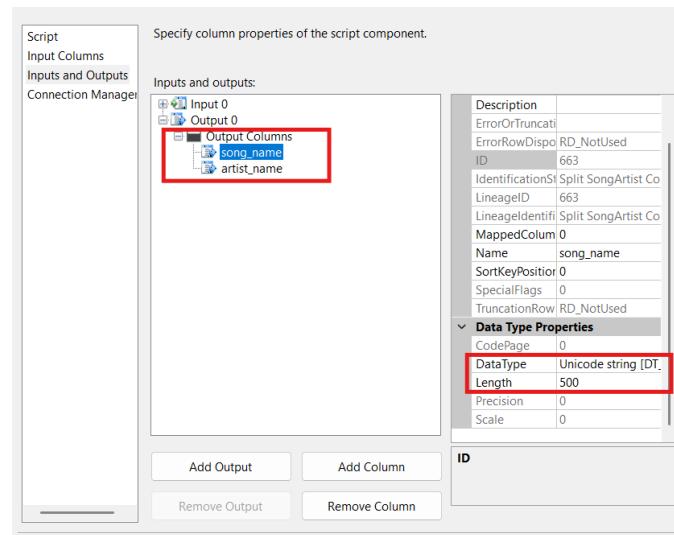
- `song_name` (`ReadOnly = True`)
- `artists` (`ReadOnly = True`)



Vào mục “**Inputs and Outputs**”, trong mục “**Common Properties**” đổi **SynchronousInputID** thành **None**.



Trong tab **Inputs and Outputs**, thêm một cột mới cho phần **Output Columns**: `artist_name` và `song_name` với kiểu dữ liệu `DT_WSTR(500)`.



Tiếp theo, vào mục **Script** → chọn **Edit Script...** và chèn đoạn mã C# để tách danh sách nghệ sĩ:

```

public override void Input0_ProcessInputRow(Input0Buffer Row)
{
    if (!Row.artists_IsNull)
    {
        string[] artistList = Row.artists.Split(';');
        foreach (string artist in artistList)
        {
            string trimmed = artist.Trim();
            if (!string.IsNullOrWhiteSpace(trimmed))
            {
                Output0Buffer.AddRow();
                Output0Buffer.songname = Row.songname;
                Output0Buffer.artistname = trimmed;
            }
        }
    }
}

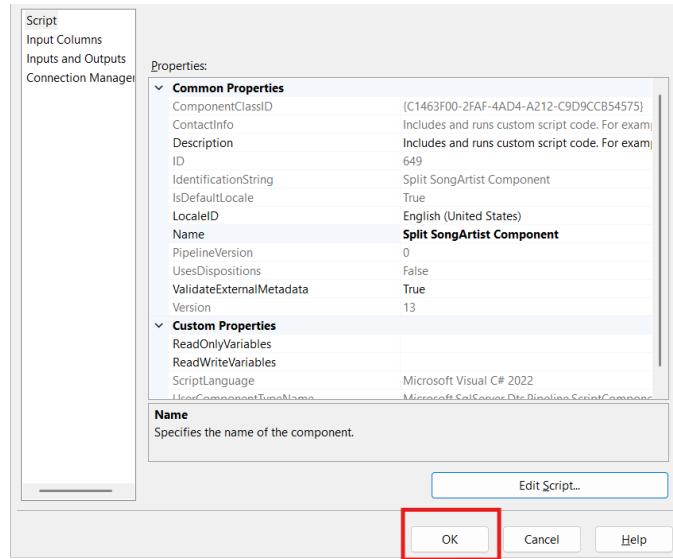
```

```

public override void Input0_ProcessInputRow(Input0Buffer Row)
{
    if (!Row.artists_IsNull)
    {
        string[] artistList = Row.artists.Split(';');
        foreach (string artist in artistList)
        {
            string trimmed = artist.Trim();
            if (!string.IsNullOrWhiteSpace(trimmed))
            {
                Output0Buffer.AddRow();
                Output0Buffer.songname = Row.songname;
                Output0Buffer.artistname = trimmed;
            }
        }
    }
}

```

Lưu Script, nhấn **OK** để hoàn tất việc biên dịch mã.



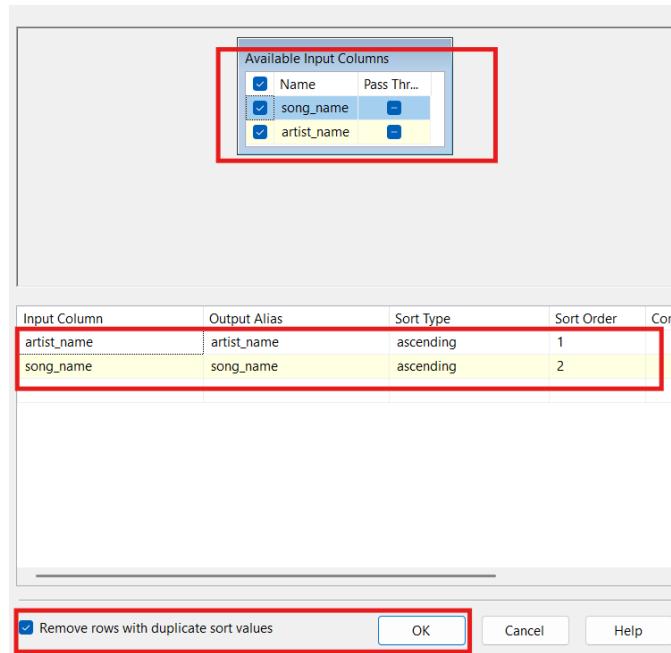
Bước 3: Thêm một Sort Transformation có tên **Remove Duplicates by Song and Artist** để loại bỏ các bản ghi trùng lặp.



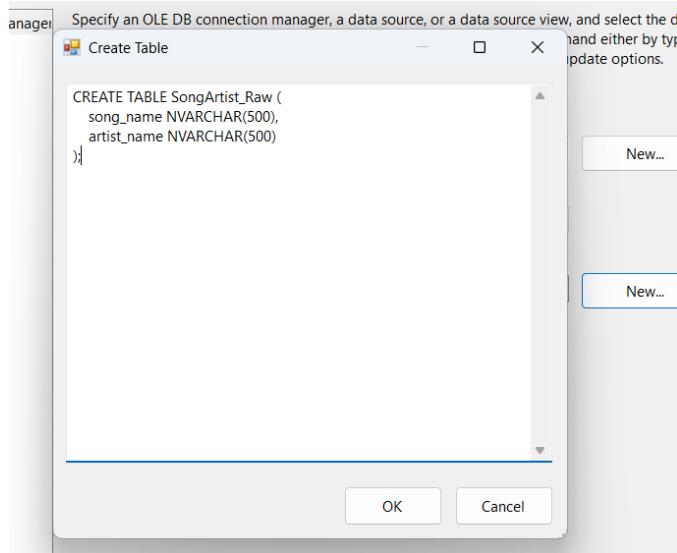
Trong phần **Available Input Columns**, chọn:

- song_name
- artist_name

và tích chọn **Remove rows with duplicate sort values**.



Bước 4: Thêm một **OLE DB Destination** có tên **Load To SongArtist Raw**. Trong phần **Connection Manager**, chọn kết nối đến cơ sở dữ liệu đích và chọn **New...** để tạo mới bảng **SongArtist_Raw**.



Cấu trúc bảng SQL:

```
CREATE TABLE SongArtist_Raw (
    song_name NVARCHAR(500),
    artist_name NVARCHAR(500)
);
```

Ghi chú:

- Không cần khóa chính (PK) trong bảng Raw để tránh lỗi khi dữ liệu trùng.
- Nếu chạy lại package nhiều lần, thêm **Execute SQL Task** với lệnh:

```
TRUNCATE TABLE SongArtist_Raw;
```

Cuối cùng, vào mục “Mappings” để kiểm tra việc ánh xạ các cột dữ liệu.



2.3.8 Tạo bảng FactSongSnapshot _ Raw

Mục đích: Bảng FactSongSnapshot_Raw lưu dữ liệu gốc về thứ hạng, độ phổ biến và chuyển động (movement) của bài hát theo từng ngày và quốc gia. Đây là nguồn dữ liệu trung gian phục vụ cho việc tạo bảng fact chính FactSongSnapshot trong Data Warehouse, nhằm phân tích xu hướng, độ phổ biến và thứ hạng của các bài hát theo thời gian.

Bước 1: Từ Multicast đầu vào, tạo một Sort Transformation có tên Sort_FactSongSnapshot_Raw để xây dựng bảng FactSongSnapshot _ Raw.



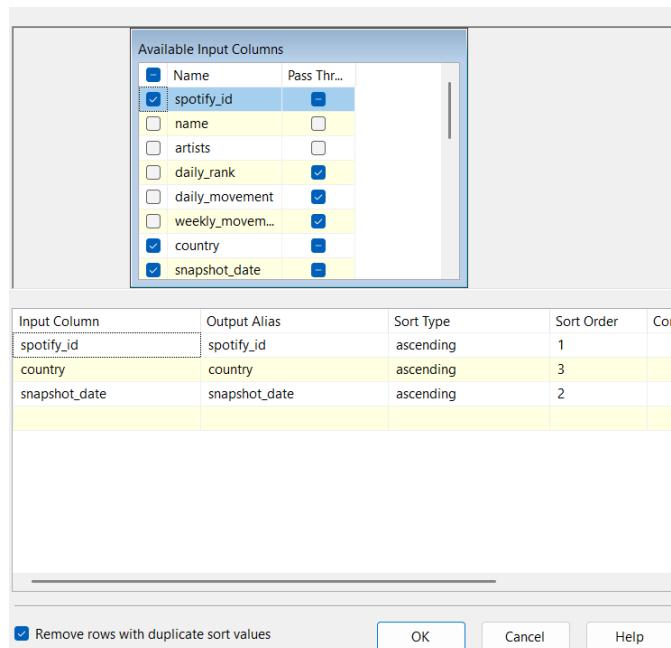
Trong phần Available Input Columns, chọn các cột:

- spotify_id
- country
- snapshot_date
- daily_rank
- daily_movement

- weekly_movement
- popularity

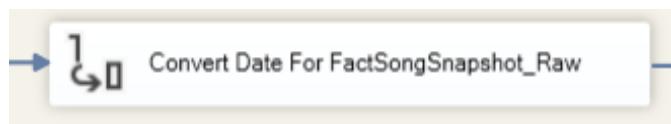
Sau đó chọn các cột sau để Sort:

- spotify_id
- country
- snapshot_date



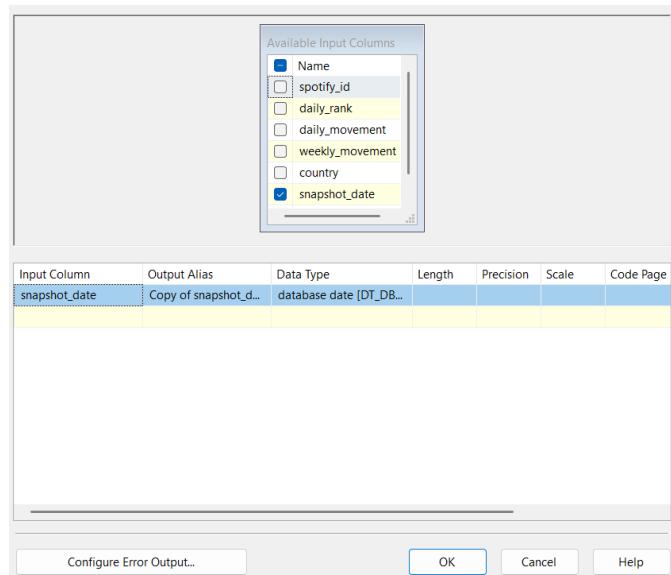
Tích chọn Remove rows with duplicate sort values để loại bỏ các bản ghi trùng lặp cùng bài hát, quốc gia và ngày.

Bước 2: Thêm một Data Conversion Transformation có tên Convert Date For Fact-SongSnapshot_Raw để chuyển đổi kiểu dữ liệu của cột snapshot_date sang định dạng ngày (DT_DBDATE), đảm bảo tương thích với kiểu dữ liệu DATE trong SQL Server.



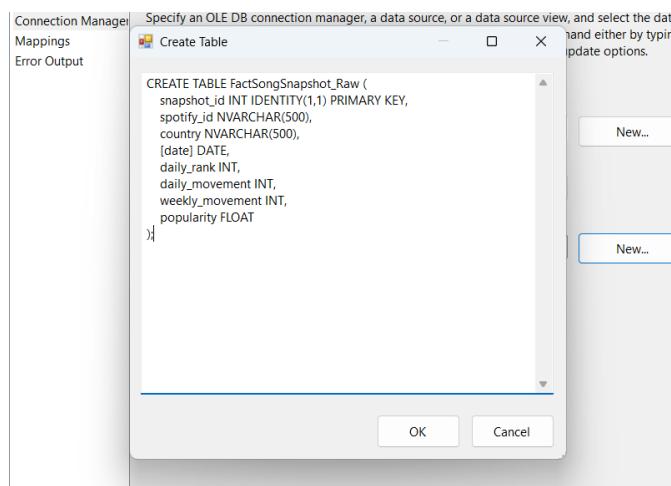
Sau đó chọn các cột:

- spotify_id
- country
- date



Tích chọn **Remove rows with duplicate sort values** để loại bỏ các bản ghi trùng lặp cùng bài hát, quốc gia và ngày.

Bước 4: Thêm **OLE DB Destination** có tên **Load To FactSongSnapshot_Raw**, chọn kết nối tới cơ sở dữ liệu đích và chọn **New...** để tạo mới bảng **FactSongSnapshot Raw**.



Cấu trúc bảng SQL:

```
CREATE TABLE FactSongSnapshot_Raw (
    snapshot_id INT IDENTITY(1,1) PRIMARY KEY,
    spotify_id NVARCHAR(500),
    country NVARCHAR(500),
    snapshot_date DATE,
    daily_rank INT,
    daily_movement INT,
    weekly_movement INT,
    popularity FLOAT
);
```

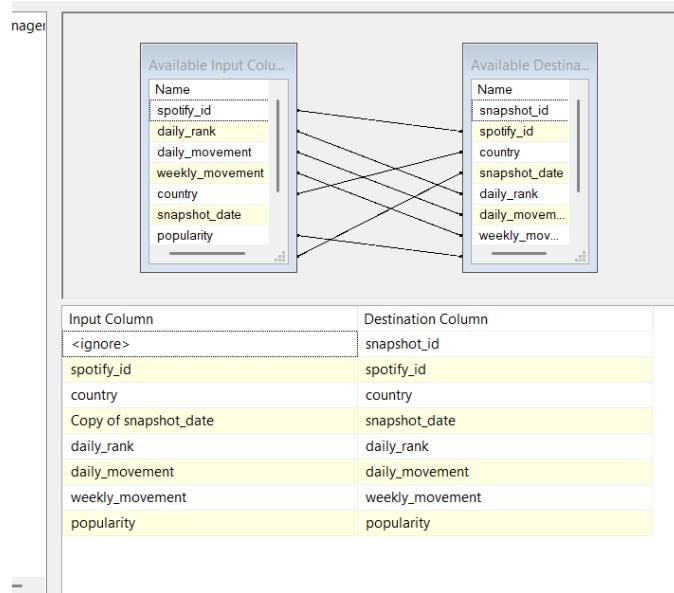
Ghi chú:

- Trong phần **Mappings**, ánh xạ cột **snapshot_date** (sau khi chuyển đổi) với trường **snapshot_date** trong bảng đích.
- Có thể thêm **Execute SQL Task** trước Data Flow để xóa dữ liệu cũ:

```
TRUNCATE TABLE FactSongSnapshot_Raw;
```

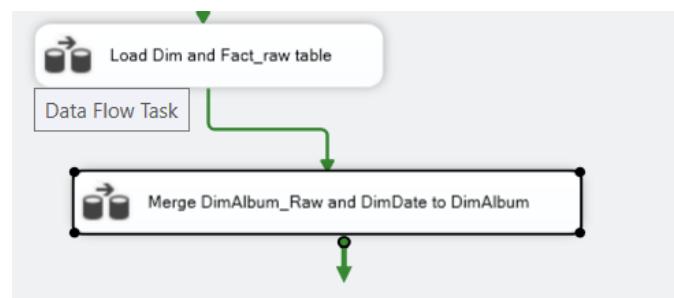
- Trong giai đoạn merge sau, **spotify_id** sẽ được ánh xạ sang **song_id**, còn **country** sẽ ánh xạ sang **country_id** từ **DimCountry**.

Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.



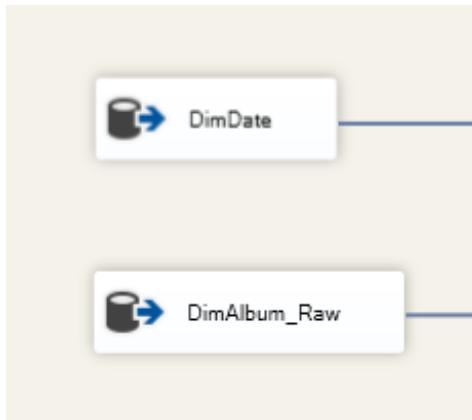
2.3.9 Merge DimAlbum_Raw với DimDate để tạo bảng DimAlbum

Bước 1: Tạo một **Data Flow Task** mới có tên **Merge DimAlbum_Raw and DimDate to DimAlbum**. Task này có nhiệm vụ kết hợp dữ liệu từ bảng **DimAlbum_Raw** với bảng **DimDate** thông qua cột ngày phát hành album để ánh xạ sang khóa thay thế **date_id**. Kết quả cuối cùng được nạp vào bảng **DimAlbum**.



Bước 2: Bên trong Data Flow, tạo hai **OLE DB Source**:

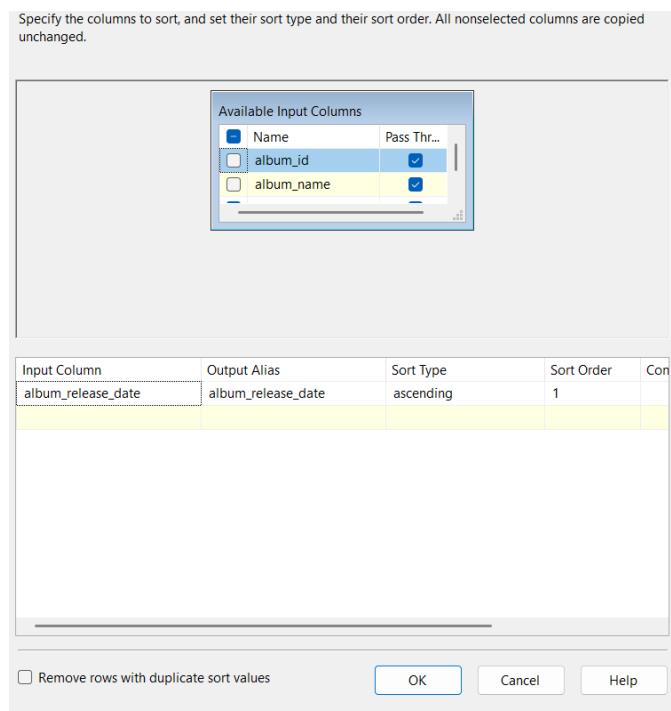
- DimAlbum_Raw
- DimDate



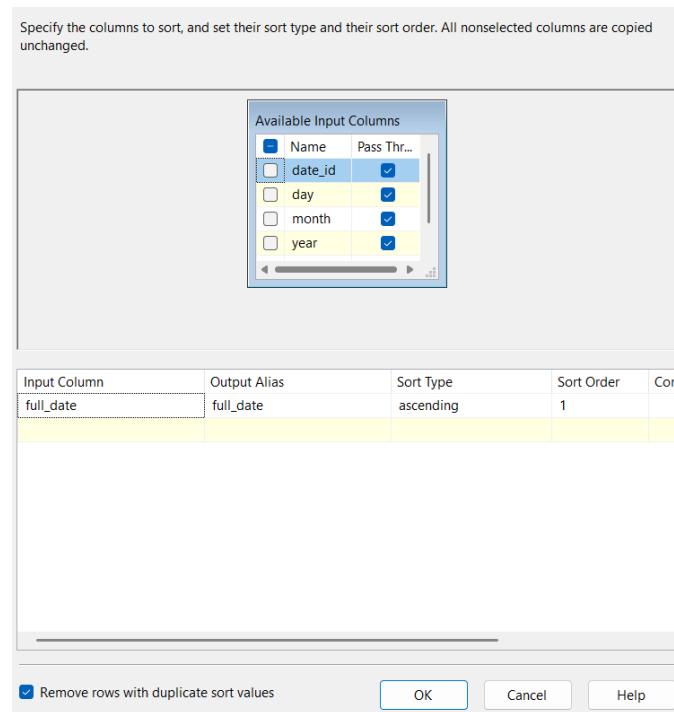
Hai bảng này sẽ được sử dụng làm đầu vào cho quá trình hợp nhất dữ liệu. Cột `album_release_date` từ `DimAlbum_Raw` sẽ được nối (join) với cột `full_date` trong `DimDate`.

Bước 3: Thêm hai **Sort Transformation** để sắp xếp dữ liệu đầu vào cho Merge Join. Cấu hình:

- Sort DimAlbum_Raw: sắp xếp theo cột `album_release_date`

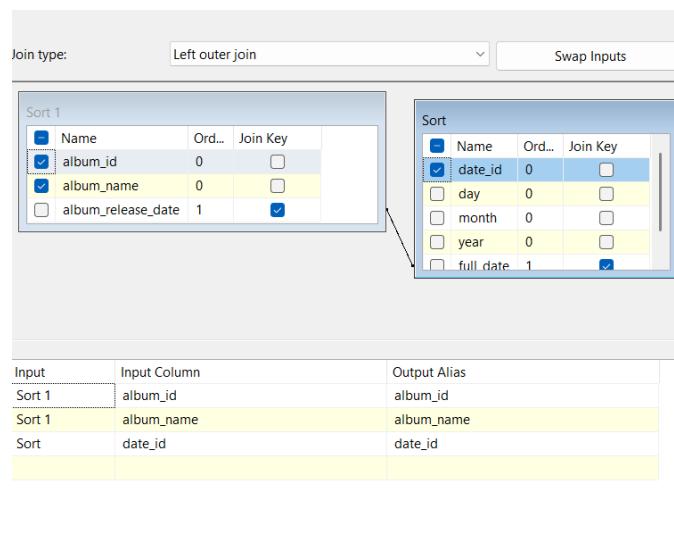


- Sort DimDate: sắp xếp theo cột `full_date`



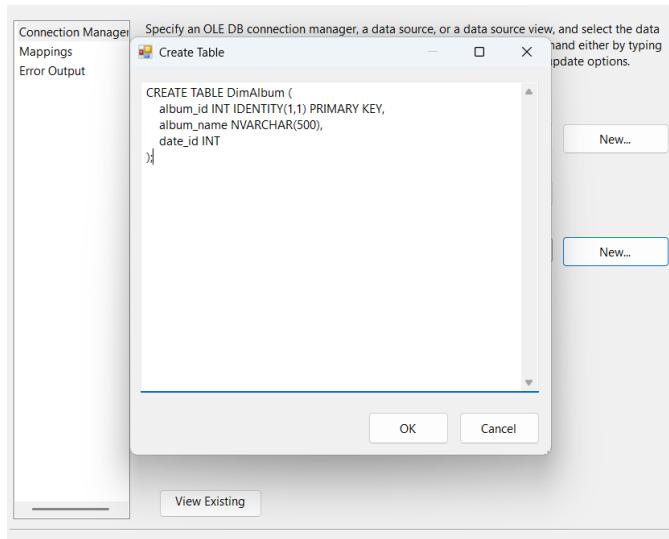
Bước 4: Sau khi Sort, đưa hai đầu ra này vào Merge Join Transformation. Chọn cấu hình:

- Left Input: DimAlbum_Raw
- Right Input: DimDate
- Join Type: Left Outer Join
- Join Key: DimAlbum_Raw.album_release_date = DimDate.full_date



Phép nối này giúp thêm cột `date_id` tương ứng với ngày phát hành album, đồng thời đảm bảo tất cả các album đều được giữ lại dù không khớp ngày trong `DimDate`.

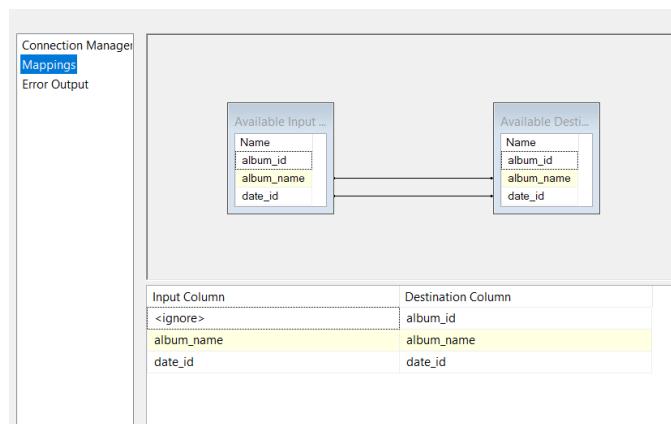
Bước 5: Thêm một **OLE DB Destination** có tên **Load To DimAlbum**, chọn kết nối đến cơ sở dữ liệu đích và chọn **New...** để tạo mới bảng `DimAlbum`.



Cấu trúc bảng SQL:

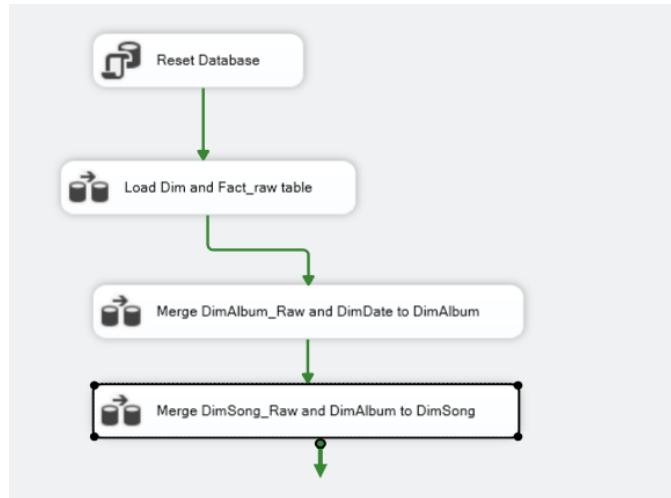
```
CREATE TABLE DimAlbum (
    album_id INT IDENTITY(1,1) PRIMARY KEY,
    album_name NVARCHAR(500),
    date_id INT
);
```

Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.



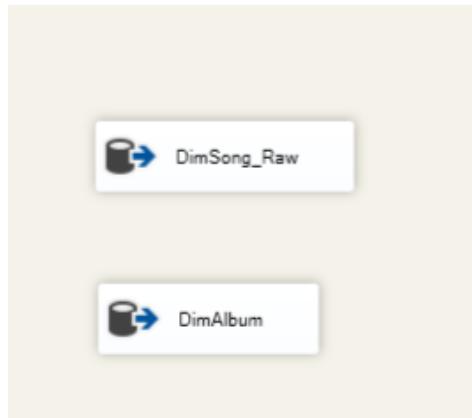
2.3.10 Merge DimSong_Raw với DimAlbum để tạo bảng DimSong

Bước 1: Tạo một Data Flow Task mới có tên **Merge DimSong_Raw and DimAlbum to DimSong**. Task này kết hợp dữ liệu từ DimSong_Raw (thông tin bài hát chi tiết) với DimAlbum (chứa khóa album_id) thông qua cột **album_name**.



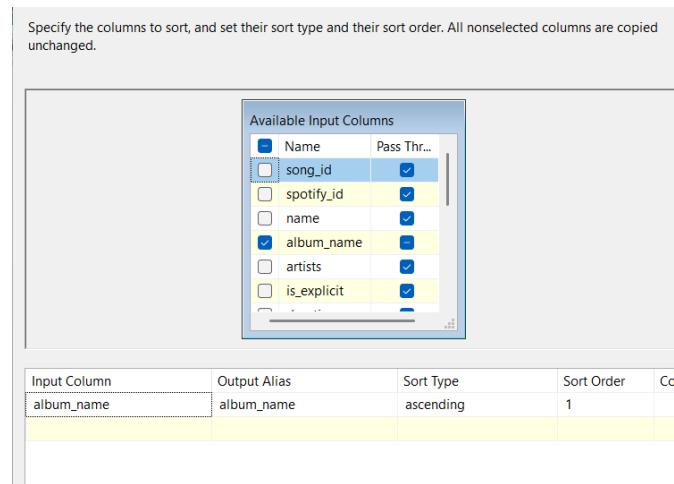
Bước 2: Thêm hai OLE DB Source:

- DimSong_Raw
- DimAlbum

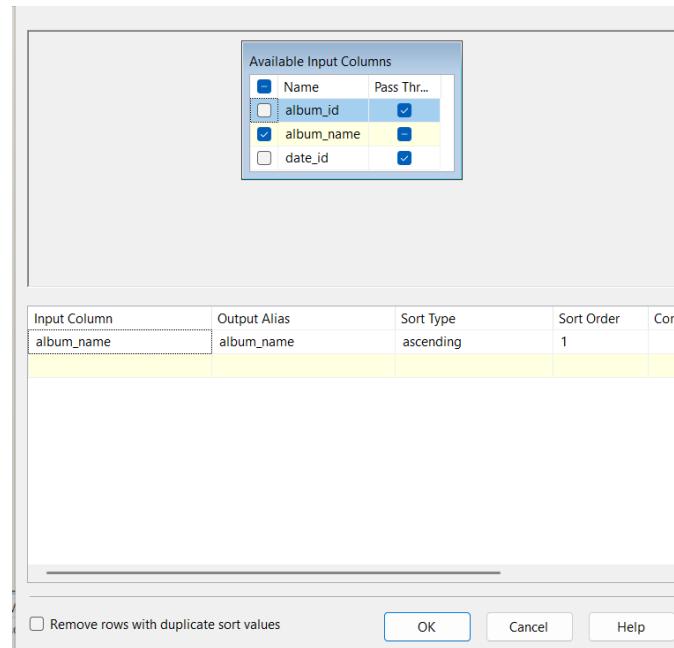


Bước 3: Thêm hai Sort Transformation để sắp xếp dữ liệu đầu vào:

- Sort DimSong_Raw: sắp xếp theo **album_name**



- Sort DimAlbum: sắp xếp theo album_name



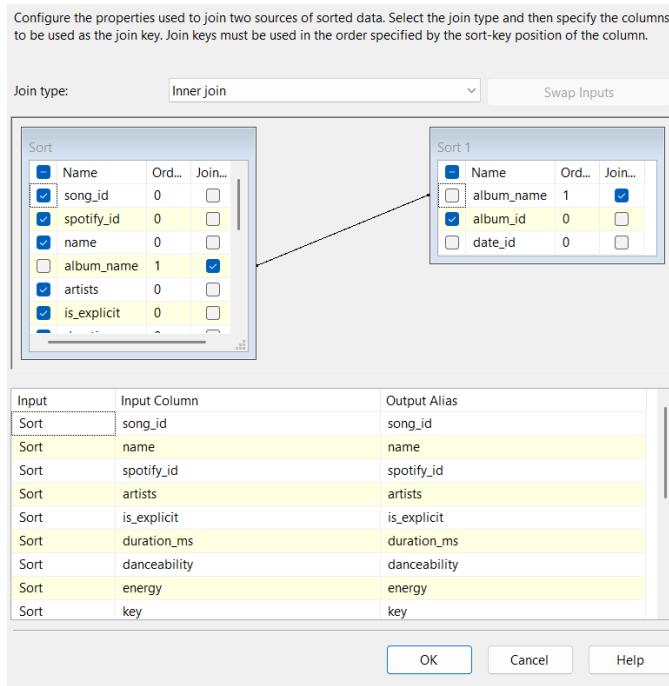
Bước 4: Thêm Merge Join Transformation, cấu hình:

- Left Input: DimSong_Raw
- Right Input: DimAlbum
- Join Type: Inner Join
- Join Key: album_name = album_name

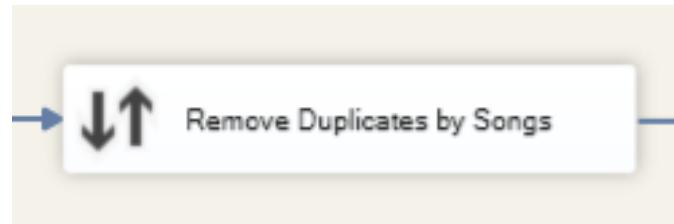
Các cột đầu ra gồm:

- Từ DimSong_Raw: name, spotify_id, is_explicit, duration_ms, danceability, energy, key, loudness, mode, acousticness, instrumentalness, tempo, time_signature.

- Từ DimAlbum: album_id.



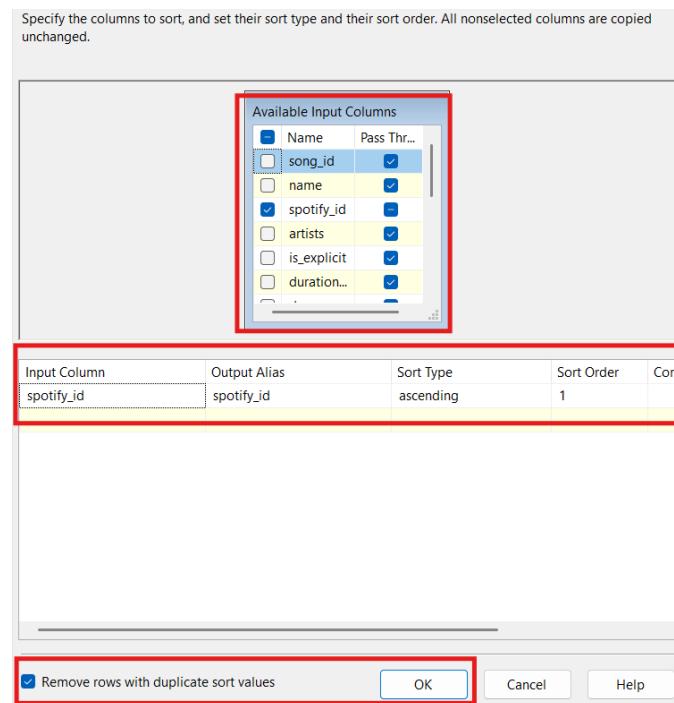
Bước 5: Thêm một Sort Transformation có tên Remove Duplicates by Songs để loại bỏ các bản ghi trùng lặp.



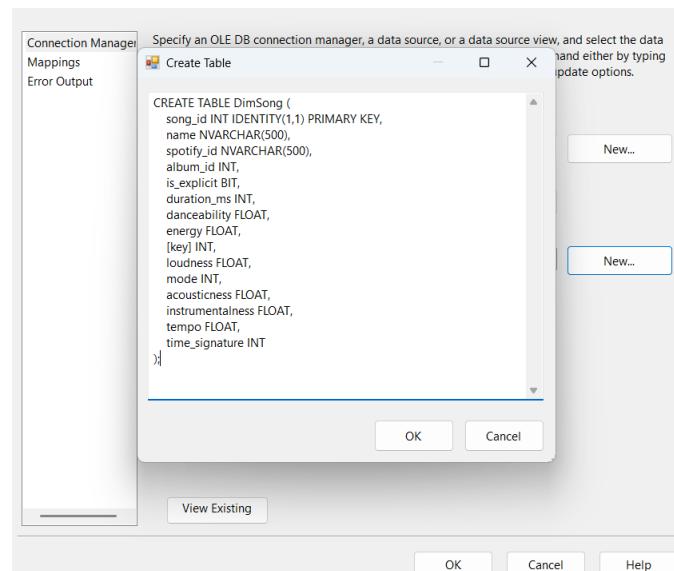
Trong phần Available Input Columns, chọn:

- spotify_id

và tích chọn Remove rows with duplicate sort values.



Bước 6: Thêm một OLE DB Destination có tên Load To DimSong, và tạo bảng DimSong trong SQL Server:



```

CREATE TABLE DimSong (
    song_id INT IDENTITY(1,1) PRIMARY KEY,
    name NVARCHAR(500),
    spotify_id NVARCHAR(500),
    album_id INT,
    is_explicit BIT,
    duration_ms INT,
    danceability FLOAT,

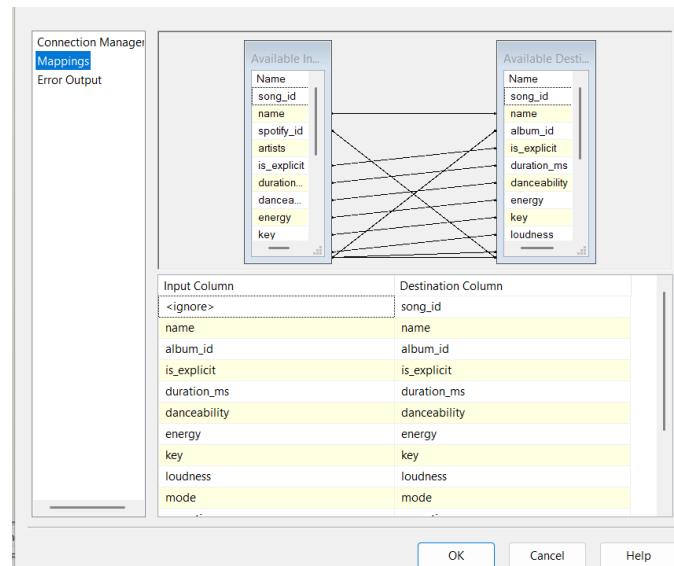
```

```

energy FLOAT,
[key] INT,
loudness FLOAT,
mode INT,
acousticness FLOAT,
instrumentalness FLOAT,
tempo FLOAT,
time_signature INT
);

```

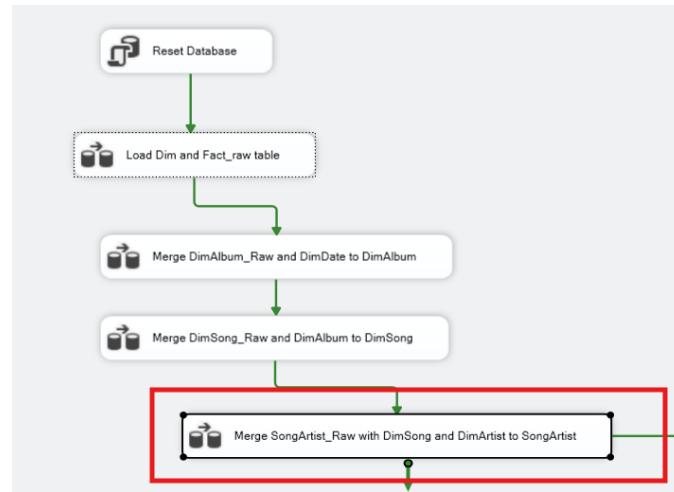
Cuối cùng, vào mục “Mappings” để kiểm tra việc ánh xạ các cột dữ liệu.



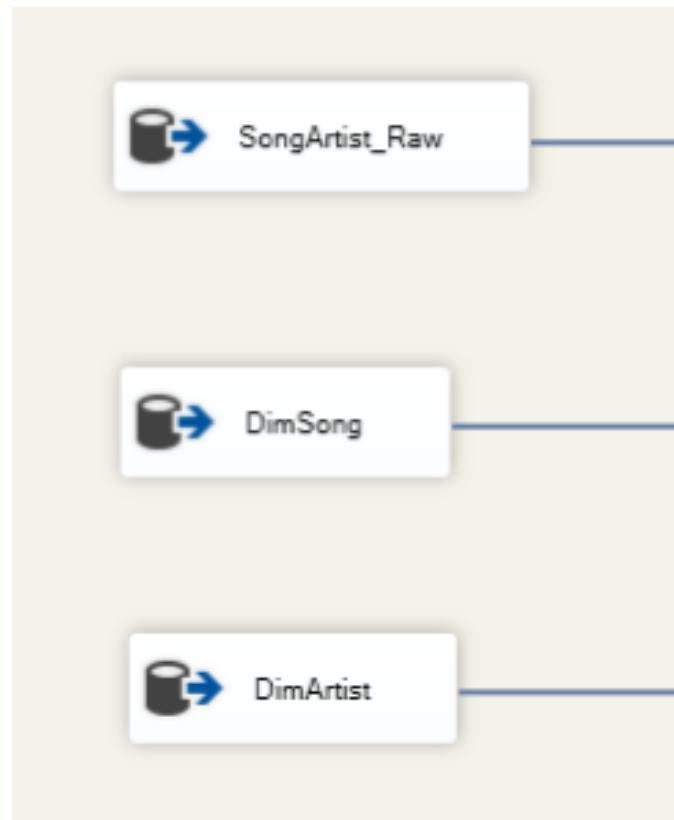
2.3.11 Merge SongArtist_Raw với DimSong và DimArtist tạo SongArtist

Mục đích: Bảng SongArtist thể hiện mối quan hệ N–N giữa bài hát và nghệ sĩ. Mỗi bài hát có thể được thể hiện bởi nhiều nghệ sĩ và ngược lại, một nghệ sĩ có thể tham gia vào nhiều bài hát khác nhau. Dữ liệu được lấy từ bảng SongArtist_Raw và được ánh xạ sang các khóa thay thế tương ứng trong DimSong và DimArtist.

Bước 1: Thêm một Data Flow Task mới có tên **Merge SongArtist_Raw with DimSong and DimArtist to SongArtist**.



Bước 2: Thêm ba OLE DB Source tương ứng với:



- SongArtist_Raw (song_name, artist_name)
- DimSong (song_id, name)
- DimArtist (artist_id, artist_name)

Bước 3: Thêm ba Sort Transformation để sắp xếp dữ liệu trước khi nối:

- SongArtist_Raw: sắp xếp theo song_name

Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

Input Column	Output Alias	Sort Type	Sort Order	Con
song_name	song_name	ascending	1	

- DimSong: sắp xếp theo name

Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

Input Column	Output Alias	Sort Type	Sort Order	Con
name	name	ascending	1	

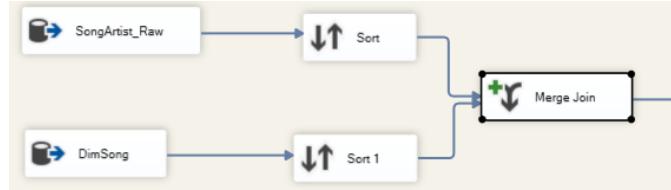
- DimArtist: sắp xếp theo artist_name

Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

Input Column	Output Alias	Sort Type	Sort Order	Con
artist_name	artist_name	ascending	1	

Column name: artist_name
Data type: Unicode string [DT_WSTR]
Length: 300
Scale: 0
Precision: 0
Source Component: DimArtist

Bước 4: Thực hiện Merge Join thứ nhất giữa SongArtist_Raw và DimSong:



- **Left Input:** SongArtist_Raw
- **Right Input:** DimSong
- **Join Key:** song_name = name
- **Join Type:** Inner Join

Đầu ra gồm artist_name và song_id.

Input	Input Column	Output Alias
Sort	artist_name	artist_name
Sort 1	song_id	song_id

Bước 5: Sắp xếp lại đầu ra của Merge Join 1 theo artist_name,

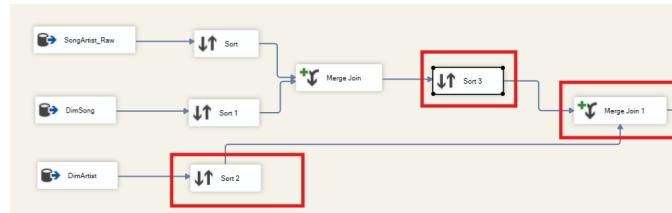
Input Column	Output Alias	Sort Type	Sort Order	Con
artist_name	artist_name	ascending	1	

sau đó thực hiện Merge Join thứ hai với DimArtist:

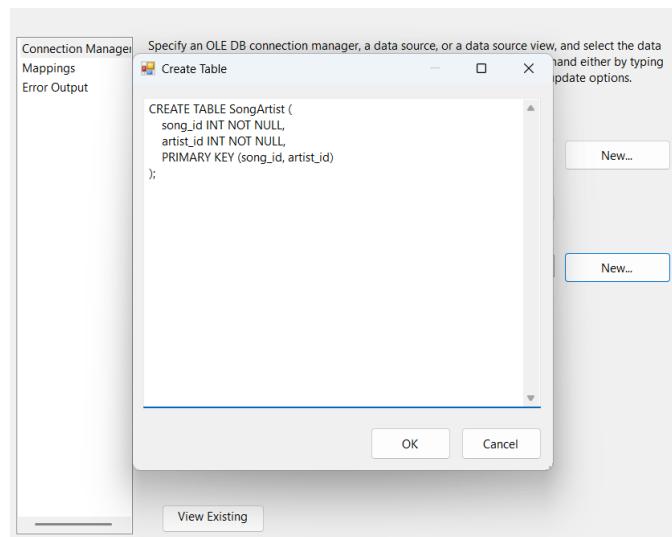
- **Left Input:** kết quả Merge Join 1

- Right Input: DimArtist
- Join Key: artist_name = artist_name
- Join Type: Inner Join

Dầu ra gồm song_id và artist_id.



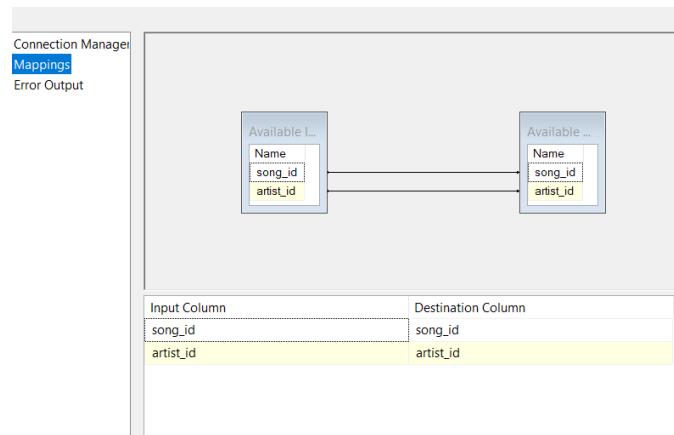
Bước 6: Thêm OLE DB Destination có tên Load To SongArtist. Trong phần Connection Manager, chọn kết nối tới cơ sở dữ liệu đích, và tạo mới bảng SongArtist như sau:



```

CREATE TABLE SongArtist (
    song_id INT NOT NULL,
    artist_id INT NOT NULL,
    PRIMARY KEY (song_id, artist_id)
);
    
```

Cuối cùng, vào mục “Mappings” để kiểm tra việc ánh xạ các cột dữ liệu.

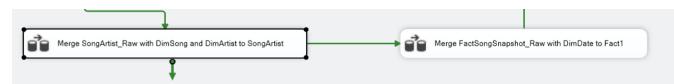


Kết quả: Bảng SongArtist được hình thành, thể hiện mối quan hệ giữa DimSong và DimArtist. Đây là bảng cầu giúp thực hiện các phép phân tích OLAP theo chiều bài hát hoặc nghệ sĩ.

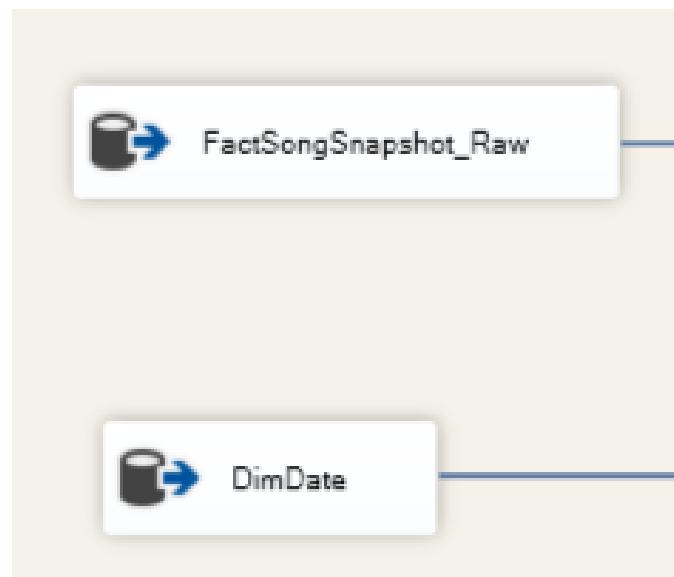
2.3.12 Tạo bảng Fact1: Merge FactSongSnapshot_Raw với DimDate

Mục đích: Bước đầu tiên trong quá trình xây dựng bảng FactSongSnapshot là ánh xạ dữ liệu từ FactSongSnapshot_Raw sang bảng DimDate, nhằm thay thế cột snapshot_date bằng khóa ngoại date_id. Việc chuẩn hóa này giúp dễ dàng tổng hợp và phân tích dữ liệu theo thời gian.

Bước 1: Tạo mới một Data Flow Task có tên **Merge FactSongSnapshot_Raw with DimDate to Fact1**.



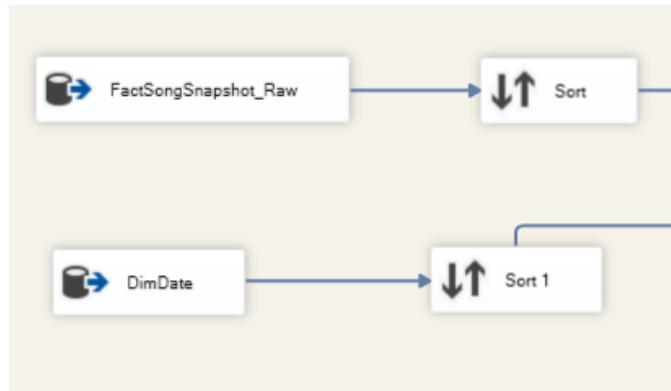
Bước 2: Thêm hai OLE DB Source tương ứng với:



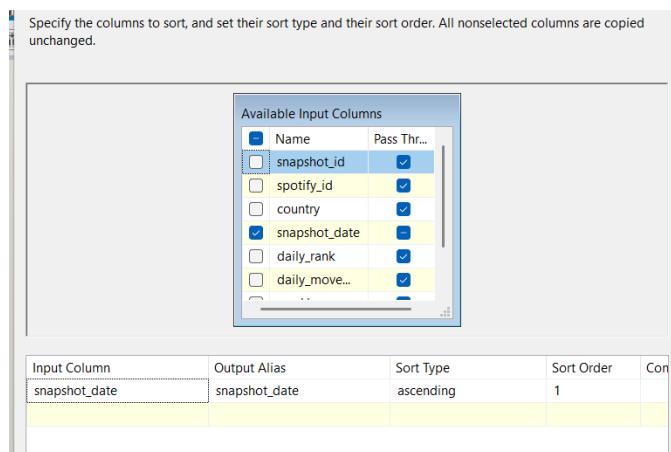
- FactSongSnapshot_Raw (các cột: snapshot_date, country, spotify_id, daily_rank, daily_movement, weekly_movement, popularity)

- DimDate (các cột: date_id, full_date)

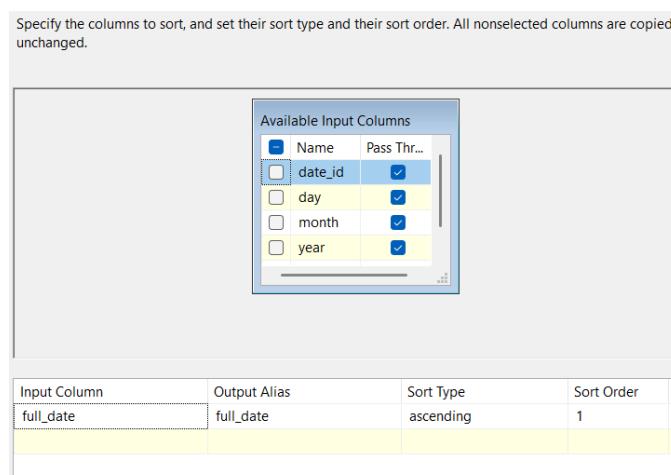
Bước 3: Thêm hai Sort Transformation để sắp xếp dữ liệu trước khi nối:

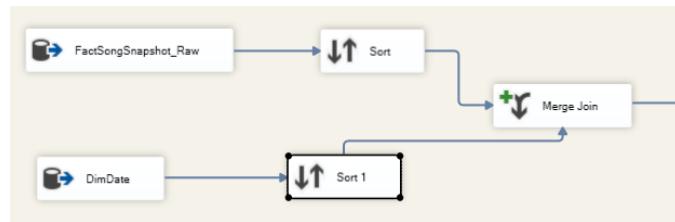


- FactSongSnapshot_Raw: sắp xếp theo snapshot_date

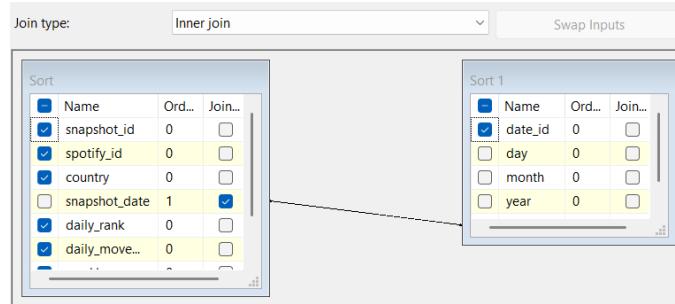


- DimDate: sắp xếp theo full_date



Bước 4: Thực hiện Merge Join giữa hai nguồn:

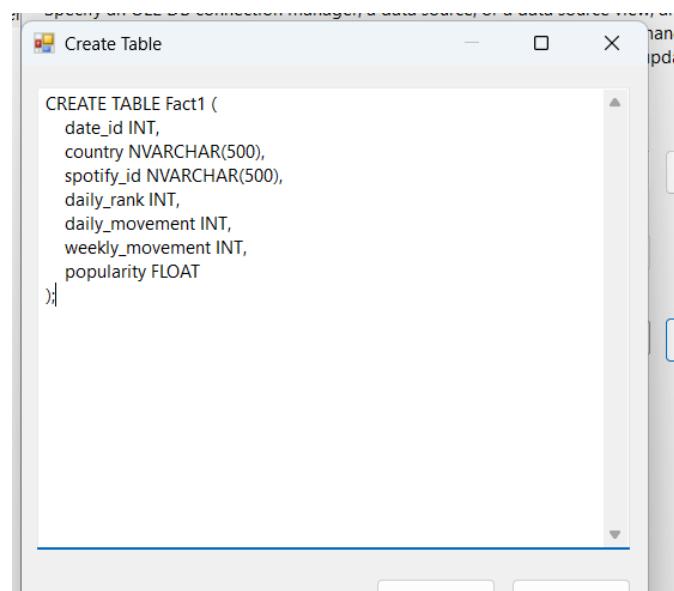
- **Left Input:** FactSongSnapshot_Raw
- **Right Input:** DimDate
- **Join Key:** snapshot_date = full_date
- **Join Type:** Inner Join

**Bước 5:** Chọn các cột đầu ra cần thiết:

- date_id (từ DimDate)
- country
- spotify_id
- daily_rank
- daily_movement
- weekly_movement
- popularity

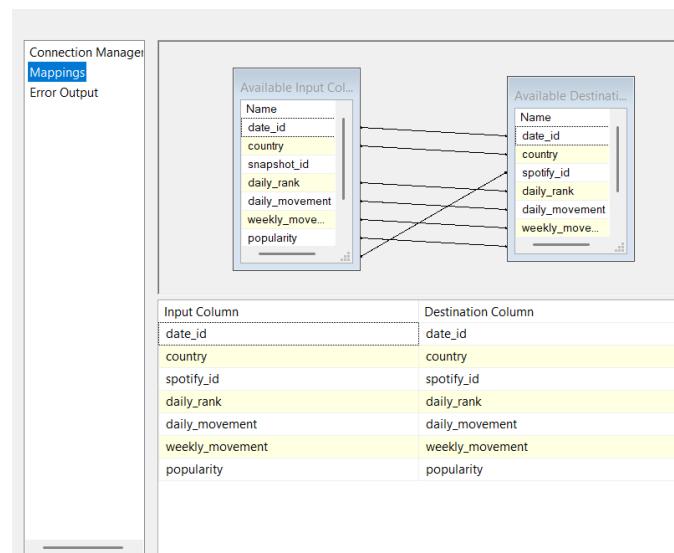
Input	Input Column	Output Alias
Sort 1	date_id	date_id
Sort	country	country
Sort	snapshot_id	snapshot_id
Sort	daily_rank	daily_rank
Sort	daily_movement	daily_movement
Sort	weekly_movement	weekly_movement
Sort	popularity	popularity
Sort	spotify_id	spotify_id

Bước 6: Thêm một **OLE DB Destination** có tên **Load To Fact1**. Trong cửa sổ **Connection Manager**, chọn kết nối tới cơ sở dữ liệu đích và tạo bảng Fact1 bằng câu lệnh SQL sau:



```
CREATE TABLE Fact1 (
    date_id INT,
    country NVARCHAR(500),
    spotify_id NVARCHAR(500),
    daily_rank INT,
    daily_movement INT,
    weekly_movement INT,
    popularity FLOAT
);
```

Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.

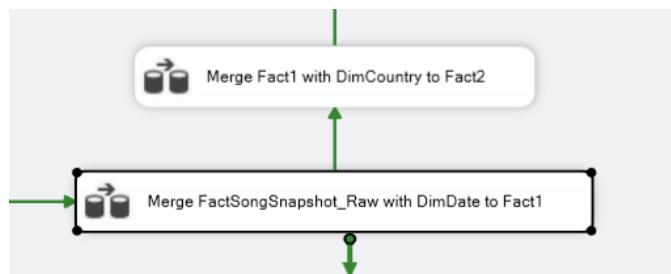


Kết quả: Bảng Fact1 được tạo thành công, trong đó mỗi dòng dữ liệu từ FactSongSnapshot_Raw đã được ánh xạ sang mã ngày tương ứng (`date_id`) trong DimDate. Dữ liệu ở bước này được sử dụng làm đầu vào cho quá trình ánh xạ tiếp theo với DimCountry.

2.3.13 Tạo bảng Fact2: Merge Fact1 với DimCountry

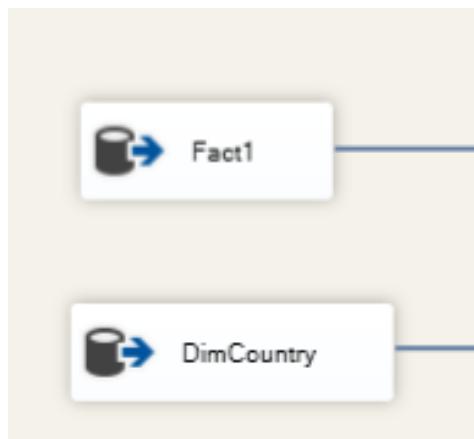
Mục đích: Bước thứ hai trong quá trình xây dựng bảng FactSongSnapshot là ánh xạ dữ liệu từ Fact1 sang bảng DimCountry, nhằm thay thế tên quốc gia bằng khóa ngoại `country_id`. Điều này giúp chuẩn hóa dữ liệu, loại bỏ dữ liệu lặp và hỗ trợ phân tích OLAP theo từng khu vực.

Bước 1: Tạo mới một Data Flow Task có tên **Merge Fact1 with DimCountry to Fact2**.

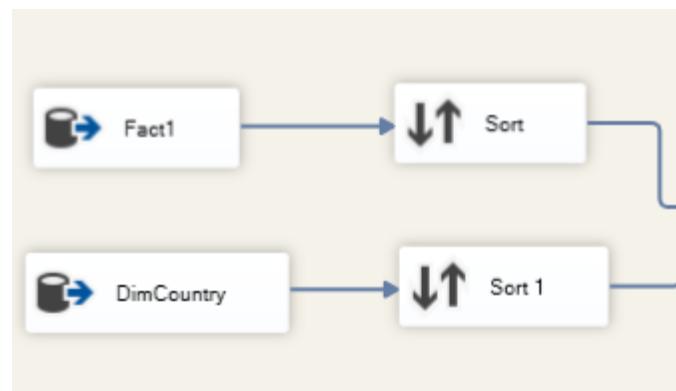


Bước 2: Thêm hai OLE DB Source tương ứng:

- Fact1 (các cột: `date_id`, `country`, `spotify_id`, `daily_rank`, `daily_movement`, `weekly_movement`, `popularity`)
- DimCountry (các cột: `country_id`, `country_name`)



Bước 3: Thêm hai Sort Transformation để đảm bảo dữ liệu được sắp xếp đúng trước khi nối:



- Fact1: sắp xếp theo country

Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

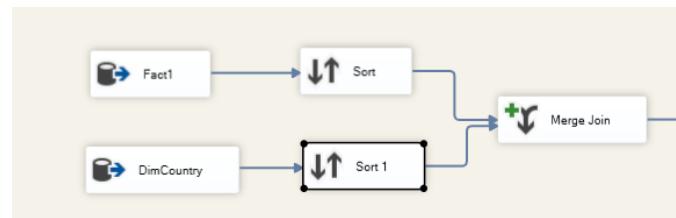
Input Column	Output Alias	Sort Type	Sort Order	Con
country	country	ascending	1	

- DimCountry: sắp xếp theo country_name

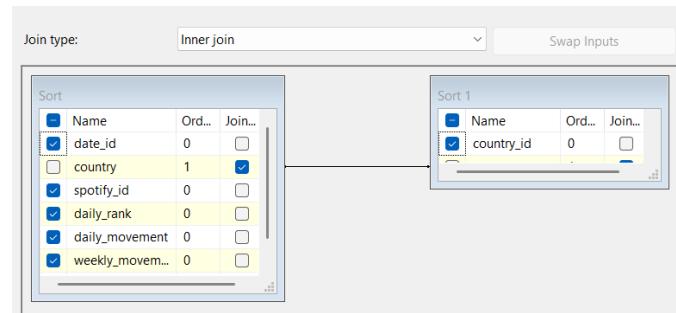
Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

Input Column	Output Alias	Sort Type	Sort Order	Con
country_name	country_name	ascending	1	

Bước 4: Thực hiện Merge Join giữa hai nguồn:



- **Left Input:** Fact1
- **Right Input:** DimCountry
- **Join Key:** country = country_name
- **Join Type:** Inner Join

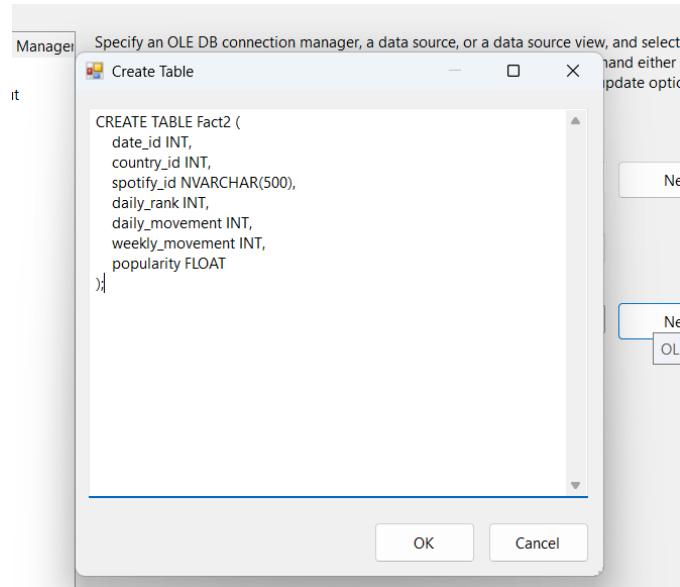


Bước 5: Chọn các cột đầu ra cần thiết:

- date_id
- country_id (từ DimCountry)
- spotify_id
- daily_rank
- daily_movement
- weekly_movement
- popularity

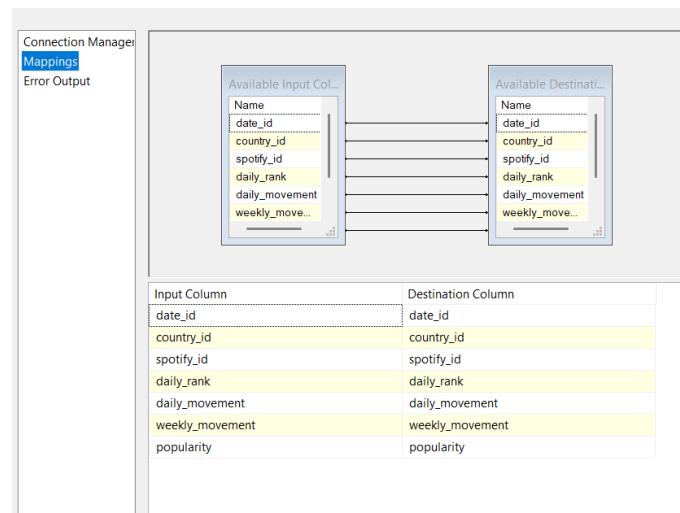
Input	Input Column	Output Alias
Sort	date_id	date_id
Sort 1	country_id	country_id
Sort	spotify_id	spotify_id
Sort	daily_rank	daily_rank
Sort	daily_movement	daily_movement
Sort	weekly_movement	weekly_movement
Sort	popularity	popularity

Bước 6: Thêm một **OLE DB Destination** có tên **Load To Fact2**. Trong phần **Connection Manager**, chọn kết nối tới cơ sở dữ liệu đích và tạo bảng Fact2 bằng lệnh SQL sau:



```
CREATE TABLE Fact2 (
    date_id INT,
    country_id INT,
    spotify_id NVARCHAR(500),
    daily_rank INT,
    daily_movement INT,
    weekly_movement INT,
    popularity FLOAT
);
```

Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.

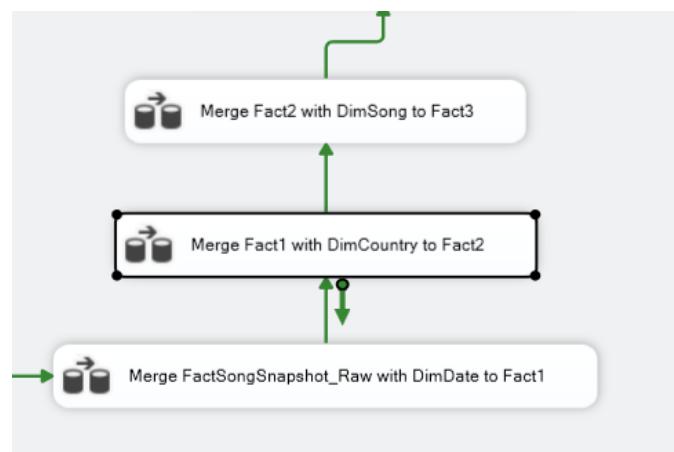


Kết quả: Bảng Fact2 được tạo thành công, trong đó mỗi bản ghi của Fact1 đã được ánh xạ sang mã quốc gia tương ứng (country_id) từ bảng DimCountry. Dữ liệu này sẽ được sử dụng làm đầu vào cho bước tiếp theo — **Merge Fact2 với DimSong để tạo Fact3**.

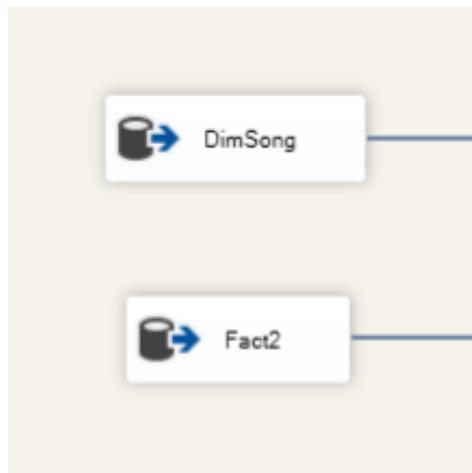
2.3.14 Tạo bảng Fact3: Merge Fact2 với DimSong

Mục đích: Bước thứ ba trong quá trình hình thành bảng FactSongSnapshot là ánh xạ dữ liệu từ Fact2 sang bảng DimSong, nhằm thay thế mã bài hát Spotify bằng khóa ngoại song_id. Việc này giúp liên kết dữ liệu chi tiết của bài hát với các thuộc tính trong DimSong để phục vụ cho các phép phân tích theo từng bài hát, nghệ sĩ, album.

Bước 1: Tạo mới một Data Flow Task có tên **Merge Fact2 with DimSong to Fact3**.

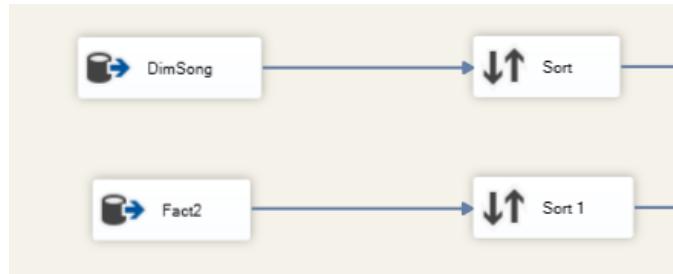


Bước 2: Thêm hai OLE DB Source tương ứng:



- Fact2 (các cột: date_id, country_id, spotify_id, daily_rank, daily_movement, weekly_movement, popularity)
- DimSong (các cột: song_id, spotify_id, name, album_id, is_explicit, duration_ms, energy, tempo, ...)

Bước 3: Thêm hai Sort Transformation để đảm bảo dữ liệu đầu vào được sắp xếp trước khi nối:



- Fact2: sắp xếp theo spotify_id

Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

Available Input Columns	
Name	Pass Thr...
<input checked="" type="checkbox"/> song_id	<input checked="" type="checkbox"/>
<input type="checkbox"/> name	<input checked="" type="checkbox"/>
<input type="checkbox"/> album_id	<input checked="" type="checkbox"/>
<input type="checkbox"/> is_explicit	<input checked="" type="checkbox"/>
<input type="checkbox"/> duration_ms	<input checked="" type="checkbox"/>
<input type="checkbox"/> danceability	<input checked="" type="checkbox"/>

Input Column	Output Alias	Sort Type	Sort Order	Cc
spotify_id	spotify_id	ascending	1	

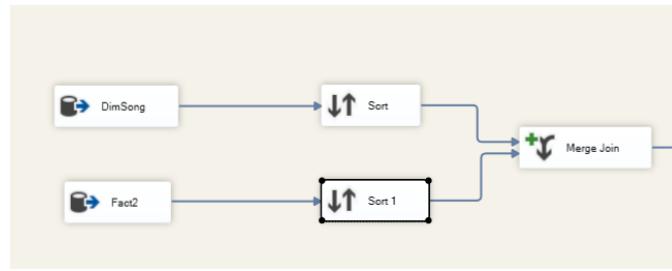
- DimSong: sắp xếp theo spotify_id

Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

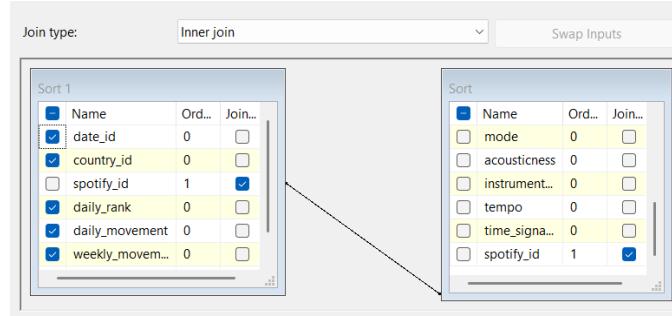
Available Input Columns	
Name	Pass Thr...
<input checked="" type="checkbox"/> date_id	<input checked="" type="checkbox"/>
<input type="checkbox"/> country_id	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> spotify_id	<input checked="" type="checkbox"/>
<input type="checkbox"/> daily_rank	<input checked="" type="checkbox"/>
<input type="checkbox"/> daily_movement	<input checked="" type="checkbox"/>
<input type="checkbox"/> weekly_movement	<input checked="" type="checkbox"/>

Input Column	Output Alias	Sort Type	Sort Order	Cor
spotify_id	spotify_id	ascending	1	

Bước 4: Thực hiện Merge Join giữa hai nguồn:



- **Left Input:** Fact2
- **Right Input:** DimSong
- **Join Key:** Fact2.spotify_id = DimSong.spotify_id
- **Join Type:** Inner Join

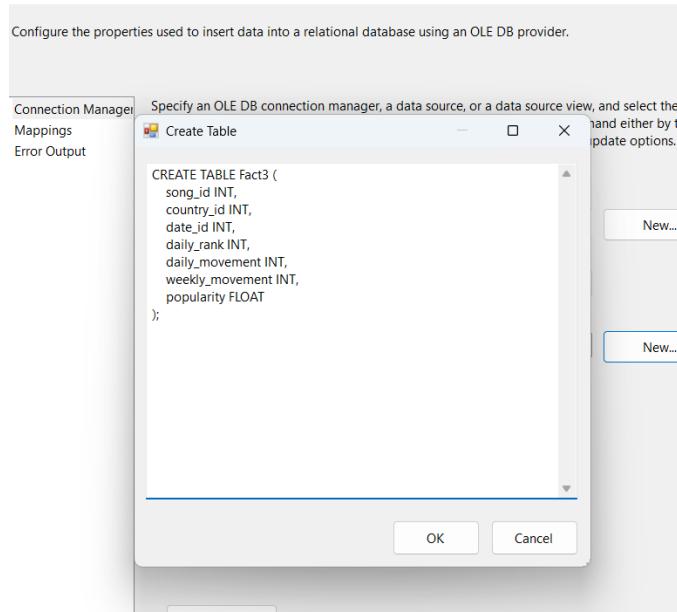


Bước 5: Chọn các cột đầu ra cần thiết:

- song_id (từ DimSong)
- country_id
- date_id
- daily_rank
- daily_movement
- weekly_movement
- popularity

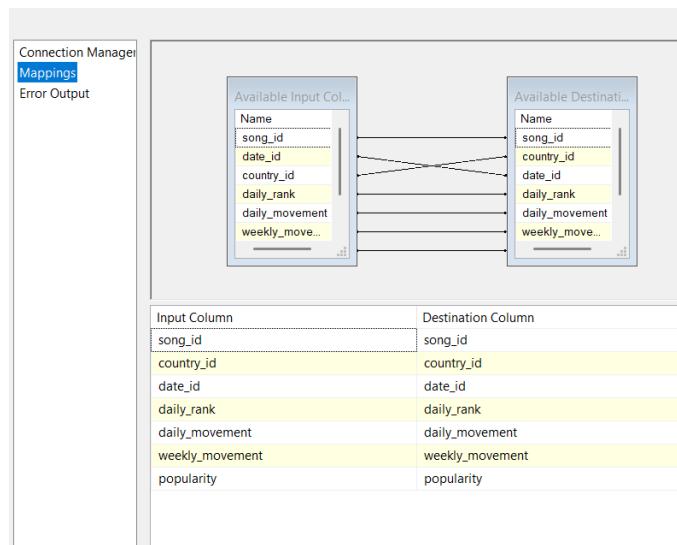
Input	Input Column	Output Alias
Sort	song_id	song_id
Sort 1	date_id	date_id
Sort 1	country_id	country_id
Sort 1	daily_rank	daily_rank
Sort 1	daily_movement	daily_movement
Sort 1	weekly_movement	weekly_movement
Sort 1	popularity	popularity

Bước 6: Thêm một **OLE DB Destination** có tên **Load To Fact3**. Trong **Connection Manager**, chọn cơ sở dữ liệu đích và tạo bảng **Fact3** bằng lệnh SQL sau:



```
CREATE TABLE Fact3 (
    song_id INT,
    country_id INT,
    date_id INT,
    daily_rank INT,
    daily_movement INT,
    weekly_movement INT,
    popularity FLOAT
);
```

Cuối cùng, vào mục “**Mappings**” để kiểm tra việc ánh xạ các cột dữ liệu.

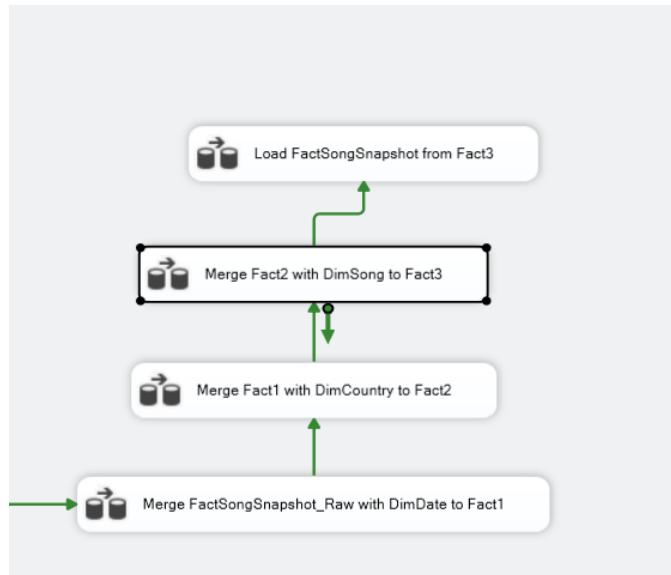


Kết quả: Bảng Fact3 được tạo thành công, trong đó mỗi bản ghi từ Fact2 đã được ánh xạ sang song_id từ DimSong. Dữ liệu ở bước này là đầu vào cho quá trình ánh xạ cuối cùng với DimPopularityGroup.

2.3.15 Tạo bảng FactSongSnapshot từ Fact3

Mục đích: Bước này hoàn thiện quá trình xây dựng bảng FactSongSnapshot – bảng Fact chính trong mô hình kho dữ liệu. Từ bảng Fact3, ta sẽ bổ sung thêm cột popularity_group_id bằng cách phân loại giá trị popularity thành các nhóm phổ biến và sau đó nạp dữ liệu vào bảng FactSongSnapshot.

Bước 1: Tạo một Data Flow Task có tên **Load FactSongSnapshot from Fact3**.



Bước 2: Thêm một OLE DB Source đọc dữ liệu từ bảng Fact3 với các cột:



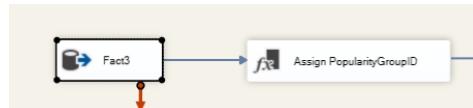
- song_id
- country_id
- date_id
- daily_rank

- daily_movement
- weekly_movement
- popularity

The screenshot shows the 'Available External Column...' dialog box at the top, listing columns: Name, song_id, country_id, date_id, daily_rank, daily_movement, and weekly_movement. Below it is a table mapping external columns to output columns:

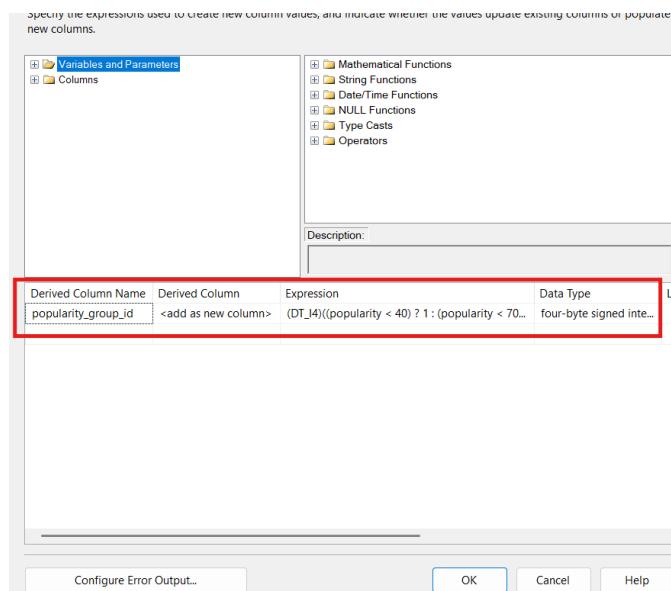
External Column	Output Column
song_id	song_id
country_id	country_id
date_id	date_id
daily_rank	daily_rank
daily_movement	daily_movement
weekly_movement	weekly_movement
popularity	popularity

Bước 3: Thêm một Derived Column Transformation để tạo cột mới popularity_group_id dựa trên giá trị popularity. Đặt tên là Assign PopularityGroupID.



Công thức:

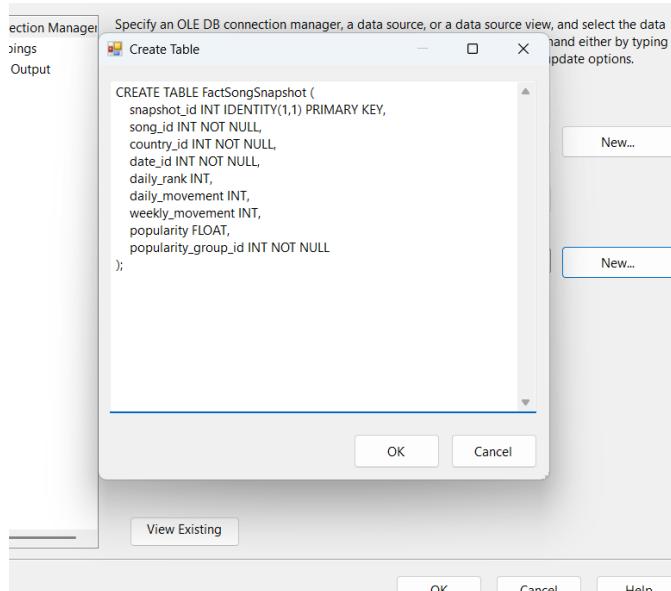
```
(DT_I4) (
    (popularity < 40) ? 1 :
    (popularity < 70) ? 2 :
    3
)
```



Giải thích:

- Nếu popularity < 40 → nhóm 1 (thấp)
- Nếu $41 \leq \text{popularity} < 71$ → nhóm 2 (trung bình)
- Nếu $71 \leq \text{popularity}$ → nhóm 3 (cao)

Bước 4: Thêm một **OLE DB Destination** có tên **Load To FactSongSnapshot**. Chọn kết nối đến cơ sở dữ liệu đích và tạo bảng bằng lệnh SQL sau:



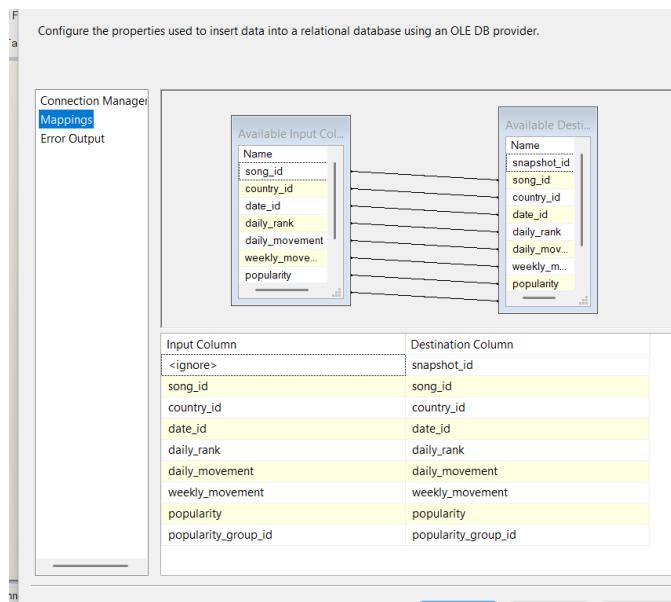
```

CREATE TABLE FactSongSnapshot (
    snapshot_id INT IDENTITY(1,1) PRIMARY KEY,
    song_id INT NOT NULL,

```

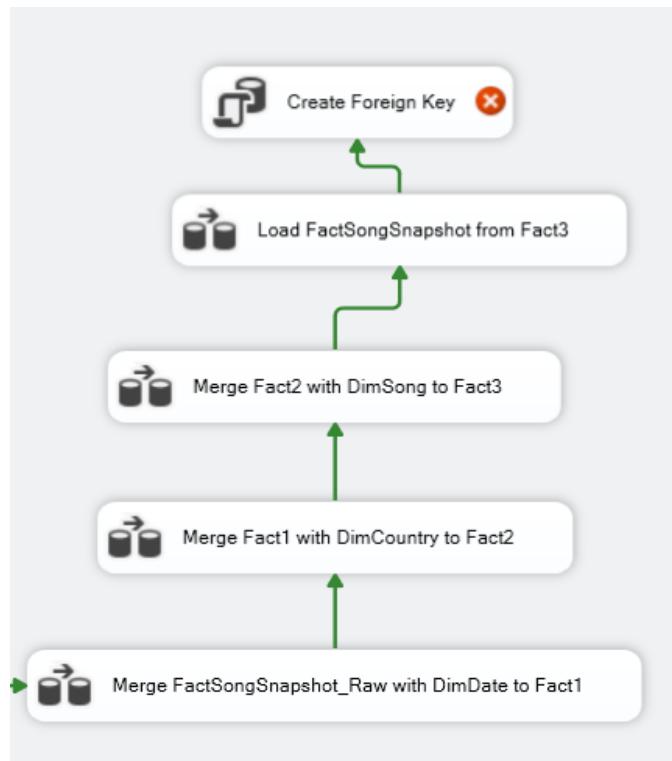
```
country_id INT NOT NULL,  
date_id INT NOT NULL,  
daily_rank INT,  
daily_movement INT,  
weekly_movement INT,  
popularity FLOAT,  
popularity_group_id INT NOT NULL  
);
```

Cuối cùng, vào mục “Mappings” để kiểm tra việc ánh xạ các cột dữ liệu.



2.3.16 Tạo khóa ngoại giữa các bảng

Bước 1: Tạo một “Execute SQL Task” Create Foreign Key.



Bước 2: Nhấn chuột phải, chọn “Edit”.

- Ở mục Connection sẽ chọn kết nối đến database trước đó.
- Ở mục SQLStatement sẽ viết các lệnh SQL tạo khóa ngoại

```

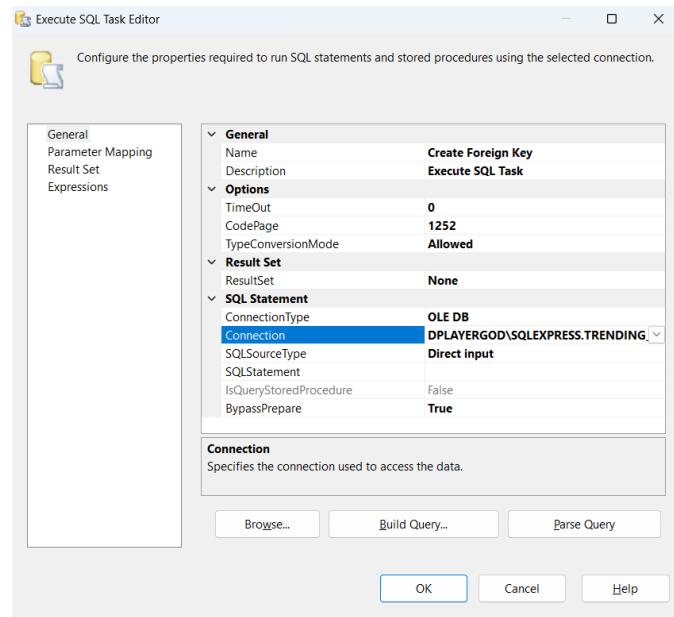
1  -- FOREIGN KEY CONSTRAINTS for Spotify OLAP Database
2
3  -- 1. DimAlbum -> DimDate
4  ALTER TABLE DimAlbum
5  ADD CONSTRAINT FK_DimAlbum_DimDate
6  FOREIGN KEY (date_id) REFERENCES DimDate(date_id);
7
8  -- 2. DimSong -> DimAlbum
9  ALTER TABLE DimSong
10 ADD CONSTRAINT FK_DimSong_DimAlbum
11 FOREIGN KEY (album_id) REFERENCES DimAlbum(album_id);
12
13 -- 3. SongArtist -> DimSong
14 ALTER TABLE SongArtist
15 ADD CONSTRAINT FK_SongArtist_DimSong
16 FOREIGN KEY (song_id) REFERENCES DimSong(song_id);
17
18 -- 4. SongArtist -> DimArtist
19 ALTER TABLE SongArtist
20 ADD CONSTRAINT FK_SongArtist_DimArtist
21 FOREIGN KEY (artist_id) REFERENCES DimArtist(artist_id);
  
```

```

22
23 -- 5. FactSongSnapshot -> DimSong
24 ALTER TABLE FactSongSnapshot
25 ADD CONSTRAINT FK_FactSongSnapshot_DimSong
26 FOREIGN KEY (song_id) REFERENCES DimSong(song_id);
27
28 -- 6. FactSongSnapshot -> DimCountry
29 ALTER TABLE FactSongSnapshot
30 ADD CONSTRAINT FK_FactSongSnapshot_DimCountry
31 FOREIGN KEY (country_id) REFERENCES DimCountry(country_id);
32
33 -- 7. FactSongSnapshot -> DimDate
34 ALTER TABLE FactSongSnapshot
35 ADD CONSTRAINT FK_FactSongSnapshot_DimDate
36 FOREIGN KEY (date_id) REFERENCES DimDate(date_id);
37
38 -- 8. FactSongSnapshot -> DimPopularityGroup
39 ALTER TABLE FactSongSnapshot
40 ADD CONSTRAINT FK_FactSongSnapshot_DimPopularityGroup
41 FOREIGN KEY (popularity_group_id) REFERENCES DimPopularityGroup(
    popularity_group_id);

```

Listing 1: Script tạo ràng buộc khóa ngoại cho các bảng Dim và Fact



Ghi chú: Tạo ràng buộc khóa ngoại sau khi nạp dữ liệu

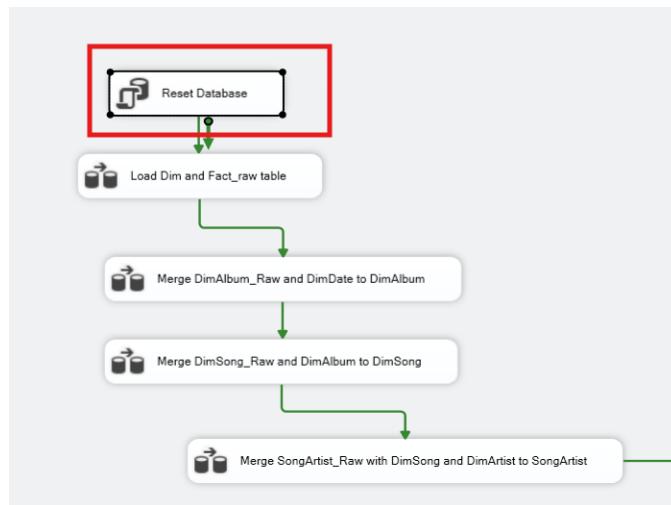
Trong quá trình thiết kế luồng dữ liệu (**Data Flow**) của hệ thống OLAP, việc tạo **ràng buộc khóa ngoại (Foreign Key)** thường được thực hiện **sau khi toàn bộ dữ liệu đã được nạp vào các bảng Dim và Fact**, thay vì khai báo trực tiếp trong giai đoạn tạo bảng ban đầu.

Cách tiếp cận này mang lại một số lợi ích về hiệu năng:

- Giảm chi phí kiểm tra ràng buộc:** Khi khóa ngoại đã tồn tại, mỗi bản ghi được nạp vào bảng Fact sẽ phải kiểm tra tính toàn vẹn tham chiếu (referential integrity) với bảng Dimension tương ứng. Điều này làm chậm đáng kể quá trình nạp dữ liệu khối lượng lớn.
- Tối ưu tốc độ ETL:** Bỏ qua kiểm tra khóa ngoại trong giai đoạn đầu cho phép giai đoạn **Load** thực thi nhanh hơn, đặc biệt khi dữ liệu được nạp từ nhiều nguồn hoặc qua các bước trung gian (như **Fact1**, **Fact2**, **Fact3**).
- Kiểm soát linh hoạt:** Sau khi toàn bộ dữ liệu đã được xác thực và làm sạch ở các bảng trung gian, việc thêm lại khóa ngoại giúp đảm bảo tính toàn vẹn dữ liệu cho các lần truy vấn và phân tích sau này.

2.3.17 Chạy SSIS

Bước 1: Thêm vào một “Execute SQL Task” Reset Database nhằm đảm bảo đổ dữ liệu mới không bị chồng chéo lên dữ liệu cũ mỗi khi chạy project. Thực hiện thêm các câu lệnh tương tự như bước tạo khóa ngoại. Lưu ý với lần chạy đầu tiên, chỉ xóa bỏ các khóa ngoại, không thực hiện các lệnh truncate table.



```

1  -- DROP FOREIGN KEYS (to avoid TRUNCATE conflicts)
2  ALTER TABLE FactSongSnapshot DROP CONSTRAINT FK_FactSongSnapshot_DimSong;
3  ALTER TABLE FactSongSnapshot DROP CONSTRAINT
   FK_FactSongSnapshot_DimCountry;
4  ALTER TABLE FactSongSnapshot DROP CONSTRAINT FK_FactSongSnapshot_DimDate;
5  ALTER TABLE FactSongSnapshot DROP CONSTRAINT
   FK_FactSongSnapshot_DimPopularityGroup;

6
7  ALTER TABLE DimAlbum DROP CONSTRAINT FK_DimAlbum_DimDate;
8  ALTER TABLE DimSong DROP CONSTRAINT FK_DimSong_DimAlbum;
9  ALTER TABLE SongArtist DROP CONSTRAINT FK_SongArtist_DimSong;
10 ALTER TABLE SongArtist DROP CONSTRAINT FK_SongArtist_DimArtist;
11

```

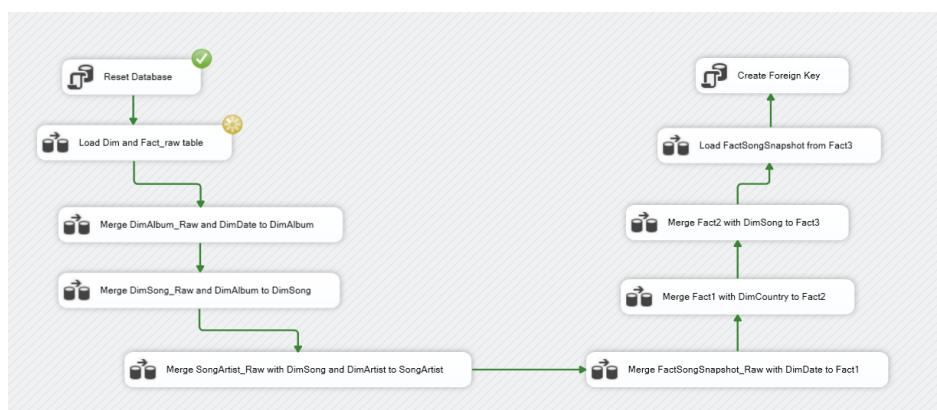
```

12
13 -- TRUNCATE RAW TABLES FIRST
14 TRUNCATE TABLE DimAlbum_Raw;
15 TRUNCATE TABLE DimSong_Raw;
16 TRUNCATE TABLE SongArtist_Raw;
17 TRUNCATE TABLE FactSongSnapshot_Raw;
18
19
20 -- TRUNCATE FACT STAGING TABLES
21 TRUNCATE TABLE Fact1;
22 TRUNCATE TABLE Fact2;
23 TRUNCATE TABLE Fact3;
24
25
26 -- TRUNCATE MAIN DIMENSIONS
27 TRUNCATE TABLE SongArtist;
28 TRUNCATE TABLE DimSong;
29 TRUNCATE TABLE DimAlbum;
30 TRUNCATE TABLE DimArtist;
31 TRUNCATE TABLE DimCountry;
32 TRUNCATE TABLE DimDate;
33 TRUNCATE TABLE DimPopularityGroup;
34
35
36 -- TRUNCATE FACT TABLE (LAST)
37 TRUNCATE TABLE FactSongSnapshot;

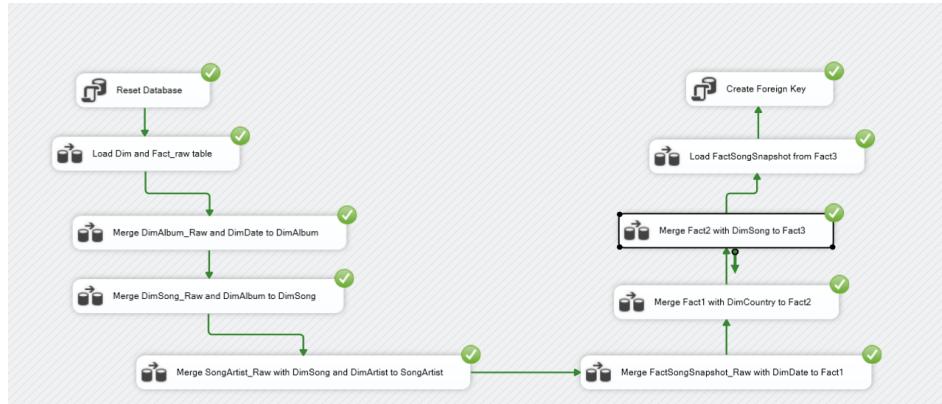
```

Listing 2: Script để Reset Database trước khi chạy SSIS

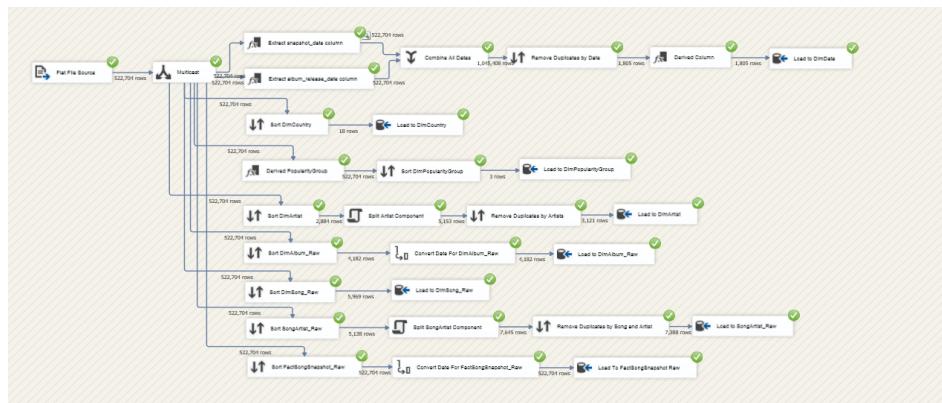
Bước 2: Nhấn vào “Start” để chạy.



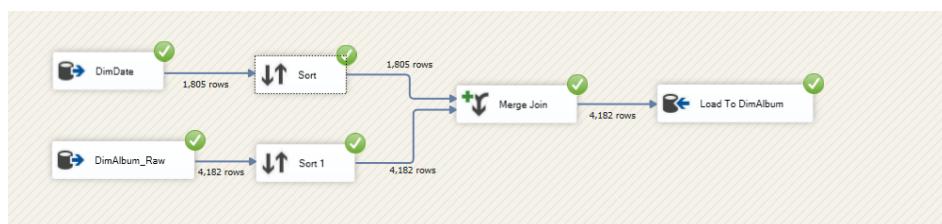
Hình 11: Chạy SSIS Package



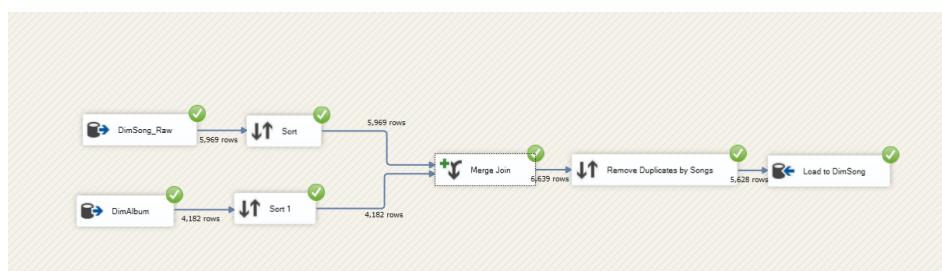
Hình 12: Kết quả chạy SSIS Package thành công



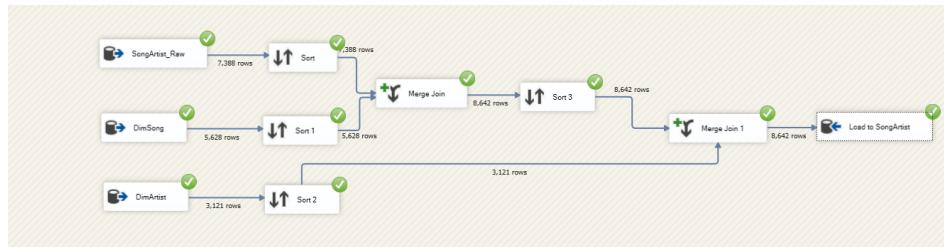
Hình 13: Dữ liệu đã được nạp vào các bảng Dim và Fact _ Raw



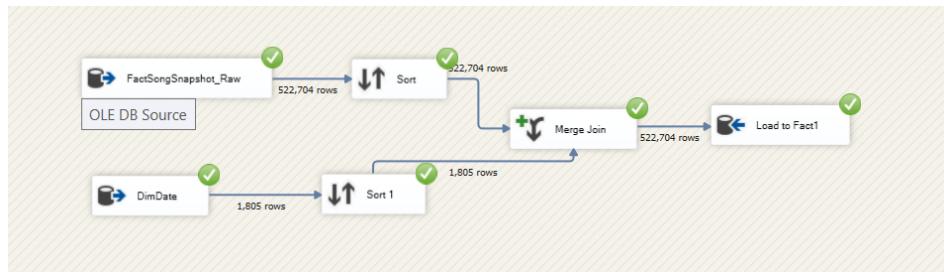
Hình 14: Dữ liệu đã được nạp vào bảng DimAlbum



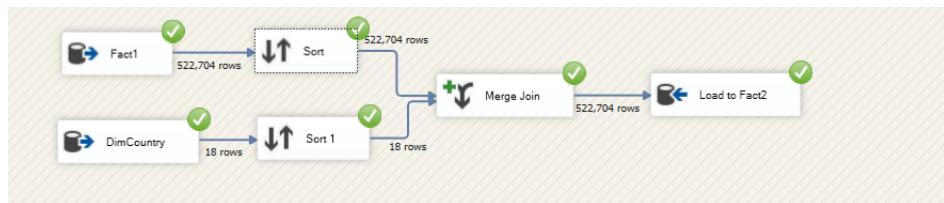
Hình 15: Dữ liệu đã được nạp vào bảng DimSong



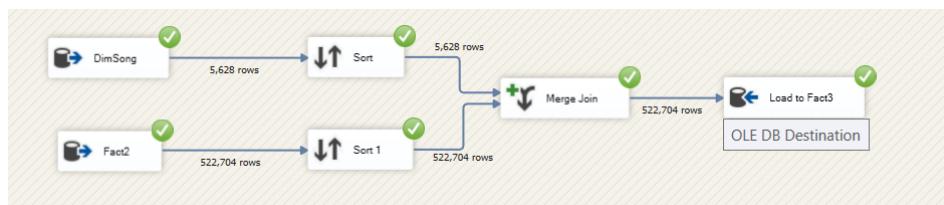
Hình 16: Dữ liệu đã được nạp vào bảng SongArtist



Hình 17: Dữ liệu đã được nạp vào bảng Fact1



Hình 18: Dữ liệu đã được nạp vào bảng Fact2



Hình 19: Dữ liệu đã được nạp vào bảng Fact3



Hình 20: Dữ liệu đã được nạp vào bảng FactSongSnapshot

2.4 Kiểm tra dữ liệu các bảng

Kiểm tra bảng DimDate:

	date_id	day	month	year	full_date
1	1	1	1	1900	1900-01-01
2	2	1	1	1942	1942-01-01
3	3	1	1	1945	1945-01-01
4	4	1	1	1947	1947-01-01
5	5	15	10	1957	1957-10-15
6	6	2	12	1957	1957-12-02
7	7	1	1	1959	1959-01-01
8	8	25	9	1961	1961-09-25
9	9	1	1	1962	1962-01-01
10	10	10	4	1962	1962-04-10
11	11	1	1	1963	1963-01-01
12	12	14	5	1963	1963-05-14
13	13	24	11	1963	1963-11-24
14	14	19	10	1964	1964-10-19
15	15	9	11	1964	1964-11-09
16	16	1	10	1965	1965-10-01
17	17	1	1	1968	1968-01-01
18	18	15	10	1970	1970-10-15
19	19	1	1	1971	1971-01-01
20	20	1	5	1971	1971-05-01
21	21	1	1	1972	1972-01-01
22	22	1	1	1973	1973-01-01
23	23	1	1	1974	1974-01-01
24	24	28	6	1974	1974-06-28
25	25	1	1	1975	1975-01-01
26	26	6	6	1975	1975-06-06
27	27	1	9	1975	1975-09-01
28	28	1	1	1977	1977-01-01

Hình 21: Dữ liệu bảng DimDate

Kiểm tra bảng DimCountry:

The screenshot shows a database interface with two tabs at the top: 'Results' and 'Messages'. The 'Results' tab is selected, displaying a table titled 'DimCountry'. The table has three columns: 'country_id' (containing values from 1 to 18), 'country_name' (containing country abbreviations like AE, Global, HK, ID, IL, IN, JP, KR, KZ, MY, PH, PK, SA, SG, TH, TR, TW, VN), and an empty column on the right.

	country_id	country_name	
1	1	AE	
2	2	Global	
3	3	HK	
4	4	ID	
5	5	IL	
6	6	IN	
7	7	JP	
8	8	KR	
9	9	KZ	
10	10	MY	
11	11	PH	
12	12	PK	
13	13	SA	
14	14	SG	
15	15	TH	
16	16	TR	
17	17	TW	
18	18	VN	

Hình 22: Dữ liệu bảng DimCountry

Kiểm tra bảng DimPopularityGroup:

The screenshot shows a database interface with two tabs at the top: 'Results' and 'Messages'. The 'Results' tab is selected, displaying a table titled 'DimPopularityGroup'. The table has five columns: 'popularity_group_id' (containing values 1, 2, 3), 'group_name' (containing Low, Medium, High), 'min_popularity' (containing 0, 41, 71), and 'max_popularity' (containing 40, 70, 100). There is also an empty column on the right.

	popularity_group_id	group_name	min_popularity	max_popularity	
1	1	Low	0	40	
2	2	Medium	41	70	
3	3	High	71	100	

Hình 23: Dữ liệu bảng DimPopularityGroup

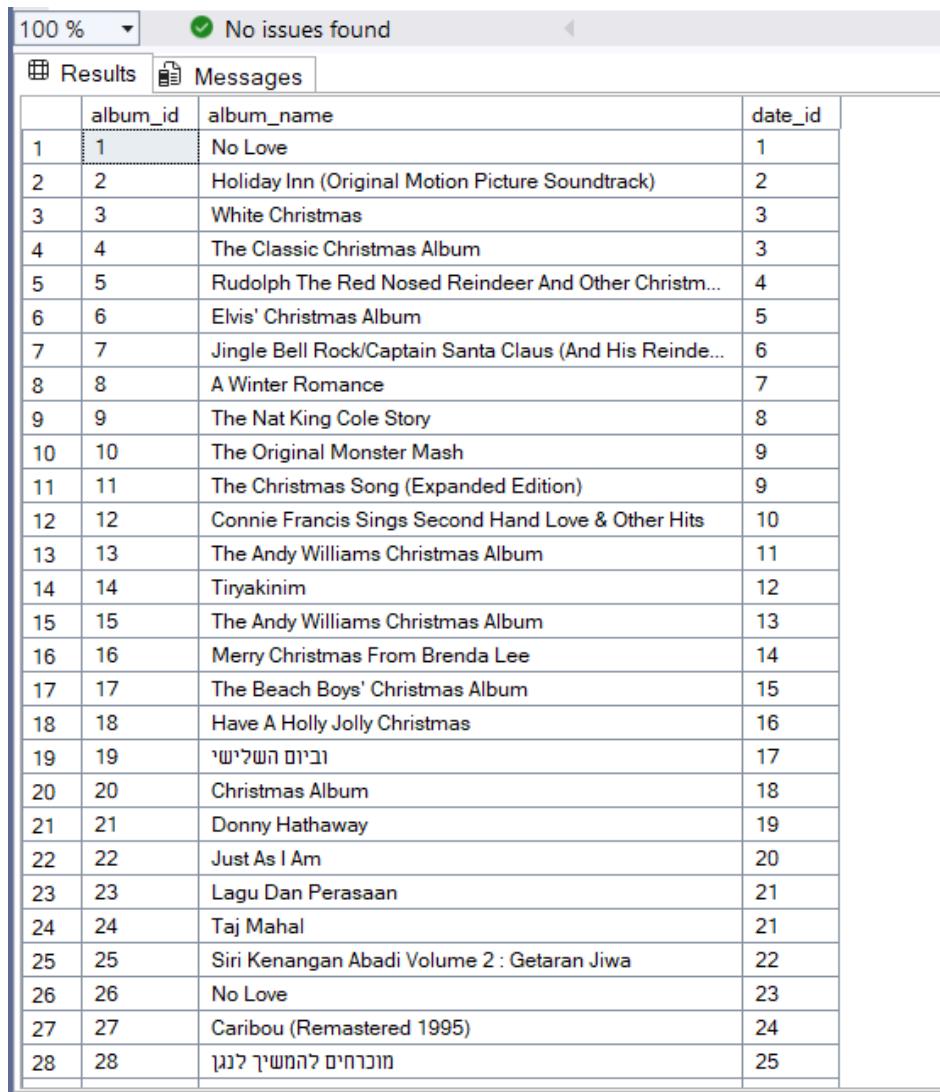
Kiểm tra bảng DimArtist:

The screenshot shows a database interface with a results grid. At the top, a message says "No issues found". Below the grid, a status bar says "Query executed successfully.".

	artist_id	artist_name
1	1	100 First Songs Participants
2	2	10CM
3	3	10FEET
4	4	1119
5	5	13 Killoki
6	6	14 Casper
7	7	1550 Collective
8	8	16 BrT
9	9	16 Typh
10	10	21 Savage
11	11	267
12	12	291i sng
13	13	2Ectasy
14	14	2K
15	15	2KE
16	16	2pillz
17	17	2T FLOW
18	18	3P
19	19	40k
20	20	4batz
21	21	4orty7 Santana
22	22	4ourYou
23	23	52Hz
24	24	53 UNIVERSE
25	25	53a
26	26	53A

Hình 24: Dữ liệu bảng DimArtist

Kiểm tra bảng DimAlbum:



The screenshot shows a software interface for viewing database results. At the top, there is a status bar with "100 %", a green checkmark icon, and the text "No issues found". Below this is a navigation bar with two tabs: "Results" (which is selected) and "Messages". The main area is a table with the following data:

	album_id	album_name	date_id
1	1	No Love	1
2	2	Holiday Inn (Original Motion Picture Soundtrack)	2
3	3	White Christmas	3
4	4	The Classic Christmas Album	3
5	5	Rudolph The Red Nosed Reindeer And Other Christm...	4
6	6	Elvis' Christmas Album	5
7	7	Jingle Bell Rock/Captain Santa Claus (And His Reinde...	6
8	8	A Winter Romance	7
9	9	The Nat King Cole Story	8
10	10	The Original Monster Mash	9
11	11	The Christmas Song (Expanded Edition)	9
12	12	Connie Francis Sings Second Hand Love & Other Hits	10
13	13	The Andy Williams Christmas Album	11
14	14	Tiryakinim	12
15	15	The Andy Williams Christmas Album	13
16	16	Merry Christmas From Brenda Lee	14
17	17	The Beach Boys' Christmas Album	15
18	18	Have A Holly Jolly Christmas	16
19	19	וביום השלייח'	17
20	20	Christmas Album	18
21	21	Donny Hathaway	19
22	22	Just As I Am	20
23	23	Lagu Dan Perasaan	21
24	24	Taj Mahal	21
25	25	Siri Kenangan Abadi Volume 2 : Getaran Jiwa	22
26	26	No Love	23
27	27	Caribou (Remastered 1995)	24
28	28	טוכחות המשין לנוג	25

Hình 25: Dữ liệu bảng DimAlbum

Kiểm tra bảng DimSong:

100 % ▾ No issues found

Results Messages

	song_id	name	album_id	is_explicit	duration	danceability	energy	key	loudness	mode	acousticness	instrumentalness	tempo	time_signature	spotify_id
1	1	Diva Yorgun	1552	0	259703	0.61599999666214	0.88	7	-6.56799993978271	1	0.32899...	1...	134.0...	4	0060GnSzenvzTimeD2oh4
2	2	Tum Kya Mile From Rocky...	1579	0	277500	0.270999979854233	0.72	7	-6.50400018662017	1	0.58600...	0	87.42...	4	00ake0KhnzbZ2MaRLQgqYX
3	3	После дождя	1583	0	169014	0.77000001001356	0.31	9	-14.3940000505409	0	0.78200...	0...	141.9...	4	00eb0d277Ecq4l015d
4	4	風神	3366	0	235543	0.707000017166138	0.63	7	-3.07100009918213	1	0.02309...	0	96.98...	4	00GDUNeJd7qKp3yx0OC
5	5	Bank Gemini	2119	0	260320	0.563600001335144	0.83	1	-3.67600005531311	0	0.25699...	1...	117.9...	4	00GZfEd1gSMedC2jHAT04
6	6	We Made It	2166	1	238000	0.721000015735626	0.59	11	-8.61200046539307	0	0.09399...	9...	137.9...	4	009eB1ZP4HvAVNzKHOz
7	7	K-POP	3919	1	112992	0.64300000667572	0.68	6	-5.97700003651123	0	0.01410...	0...	143.9...	4	001LtefLLeemBlh6Jh
8	8	我愛你對不起謝謝你沒關係	2311	0	305526	0.0412999987602234	0.52	2	-7.1669998169453	1	0.31600...	0	152	4	009Popb3w3eqapLxAe+4+1t
9	9	纯情子彈	3629	0	290267	0.055000007152557	0.35	0	-10.81900024411406	1	0.72299...	0	141.9...	4	0056ywDfBSNu2oFv
10	10	Rum Pum Pum	3800	0	209613	0.753000002098035	0.82	4	-4.04500007629395	1	0.14200...	3...	119.9...	4	00M2zCejhqyB5wWRUkqgjY3
11	11	I wonder	2636	0	157746	0.787000000476837	0.63	6	-3.8819999648242	1	0.50800...	0	124.9...	4	00Q3uYmFKY7TRbenUq7H
12	12	welcome and goodbye	666	0	140643	0.583999991416931	0.71	0	-5.750001907344	0	0.01940...	0.7...	140.0...	4	00RLNHzjkEJcUJcUlfPvPT
13	13	mashiro pure white	3846	0	294140	0.686000002563904	0.69	7	-9.5699999482422	0	0.26699...	0...	125.0...	4	00WTRpkXPXtmIah74vrQi
14	14	Gor Bak	2439	0	218086	0.77999997457491	0.82	7	-6.57299995422363	1	0.17399...	5.0...	154.9...	3	00zpolHLB7zPxWEsYDqB
15	15	HAHVHVHV	3702	1	126255	0.799000002479532	0.53	10	-11.29399971936	0	0.25099...	0...	120.0...	4	0126WdQmnNEazqJUb1wgQH
16	16	Munthrirhar	3964	0	234426	0.782000005245209	0.76	2	-6.5879998207023	1	0.38699...	0.0...	125.0...	4	0133eEuP09SQcqHqgfbS
17	17	Accendio	2677	0	192013	0.75900000333786	0.879	0	-3.9419999122613	0	0.04930...	0	140.0...	4	0139nj8LjLS4YQGYZu
18	18	Chúng Ta Còn Ở Đó Không...	3731	0	245217	0.6800000026236044	0.36	11	-9.52200001205018	0	0.71799...	0	114.9...	4	015cxWefay+eU1ZodBpfn
19	19	Cüt Çüt Çedene	45	0	233946	0.688000007152557	0.52	5	-16.041999816845	0	0.28000...	0	82.5...	4	01f8ba1J7enRnLWa6sauJK
20	20	homebody feat Madman Stan	1619	0	141762	0.746999979019165	0.44	3	-6.94600009918213	0	0.08720...	0	121.9...	4	01NOMvE89ukJeurTDZX2Y
21	21	New Jeans Jersey Club Sl...	3392	0	110005	0.983000009059906	0.59	1	-4.70599985122681	0	0.46999...	0...	125.1...	4	01KwChzvBSPNDj2bzet
22	22	TELEFONO NUEVO	1847	1	354784	0.619000017642975	0.61	2	-6.23799991607666	1	0.40700...	0	125.7...	4	01ppkIDRcmPpuo03yNShRy
23	23	Rockstar Extended	2916	1	164782	0.644999980926514	0.68	1	-4.78999978637695	1	0.06060...	1...	139.9...	4	01rPr0GCTxKbxJd00zwQ
24	24	3D fest Jack Harlow	1812	1	201812	0.852999985218048	0.82	1	-3.26899994087219	1	0.03220...	0	108.0...	4	01gKNWq73UJell0QumE
25	25	Goodbye	3275	0	244000	0.709999978542328	0.20	10	-9.741000017547607	1	0.94800...	0	75.96...	4	01Xepgj5VwHJaNImm
26	26	KTP	1200	1	187120	0.690000027179718	0.90	0	-2.8150000572204	1	0.07060...	0	122.0...	4	01TMLHxBLINd1dJ8khPF
27	27	Lost	1869	0	162475	0.648999989032745	0.66	4	-7.63600001564028	0	0.02300...	5.2...	100.0...	4	01UKgmj21ycGWh2fM6Cn
28	28	想唱就唱	2606	0	260903	0.523000001907349	0.48	9	-5.85400009155273	1	0.02390...	2.0...	185.8...	4	01VbKloQ9QYeSc7eC
29	29	Satyanaas From Chundu C...	2781	0	205088	0.621999979019165	0.83	11	-4.44399976730347	1	0.09830...	0	130.8...	4	01Vpd2o65RTFCbbdBQ
30	30	Sadcan Kalca	1664	1	134049	0.828000009059906	0.81	9	-4.84530000686455	0	0.27399...	1...	142.0...	4	01zEtTdJ6wRoQzXbfbV

Query executed successfully.

DPLAYERGOD\SQL EXPRESS (16.0...) DplayerGod\Dang Quoc C... TRENDING_SONGS_SSIS 0

Hình 26: Dữ liệu bảng DimSong

Kiểm tra bảng SongArtist:

100 % ▾ No issues found

Results Messages

	song_id	artist_id
1	1	1681
2	2	167
3	2	230
4	2	2049
5	2	2392
6	3	2748
7	4	2757
8	5	2319
9	6	892
10	6	1875
11	7	2021
12	8	2818
13	9	780
14	10	74
15	11	596
16	12	751
17	12	1162

Query executed successfully.

Hình 27: Dữ liệu bảng SongArtist

Kiểm tra bảng FactSongSnapshot:

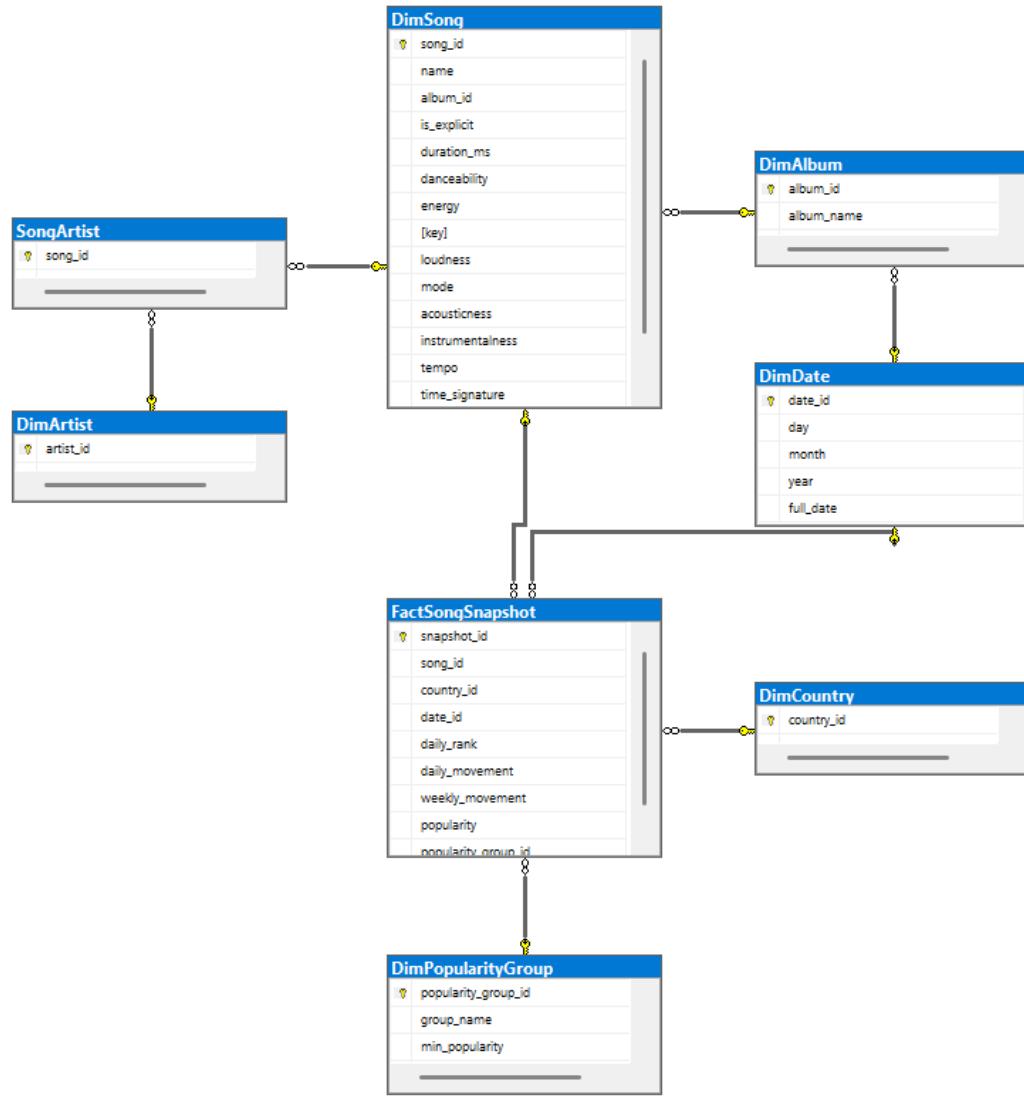
The screenshot shows a database query results window with the following details:

- Query ID: 1
- Progress: 100 %
- Status: No issues found
- Results tab is selected.
- Table structure:
 - Columns: snapshot_id, song_id, country_id, date_id, daily_rank, daily_movement, weekly_movement, popularity, popularity_group_id
 - Primary key: snapshot_id
- Data rows (approximate values):

snapshot_id	song_id	country_id	date_id	daily_rank	daily_movement	weekly_movement	popularity	popularity_group_id
1	1	16	1236	46	3	3	75	3
2	2	1	16	1220	41	-7	77	3
3	3	1	16	1235	49	1	75	3
4	4	1	16	1238	50	-2	75	3
5	5	1	16	1229	49	-2	76	3
6	6	1	16	1237	48	1	75	3
7	7	1	16	1227	45	-4	76	3
8	8	1	16	1234	50	-5	76	3
9	9	1	16	1222	38	-3	77	3
10	10	1	16	1226	45	-3	76	3
11	11	1	16	1242	46	3	75	3
12	12	1	16	1239	50	-4	75	3
13	13	1	16	1221	40	-7	77	3
14	14	1	16	1243	46	0	75	3
15	15	1	16	1244	47	1	75	3
16	16	1	16	1232	46	-4	76	3
17	17	1	16	1231	49	-9	76	3
18	18	1	16	1207	27	23	77	3
19	19	1	16	1211	32	-2	77	3
20	20	1	16	1228	47	-7	76	3
21	21	1	16	1224	40	-2	76	3
22	22	1	16	1218	37	-5	77	3
23	23	1	16	1214	33	-6	77	3
24	24	1	16	1223	39	-1	76	3

Hình 28: Dữ liệu bảng FactSongSnapshot

2.5 Lược đồ sau khi hoàn thành

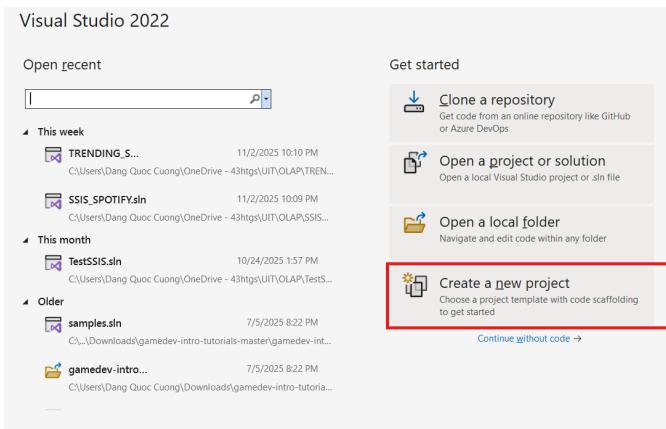


Hình 29: Lược đồ sau khi hoàn thành

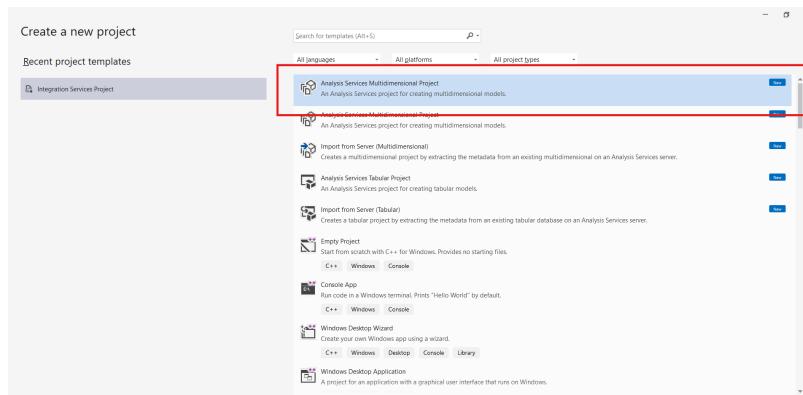
3 Phân tích dữ liệu trực tuyến (SSAS)

3.1 Tạo mới Project SSAS

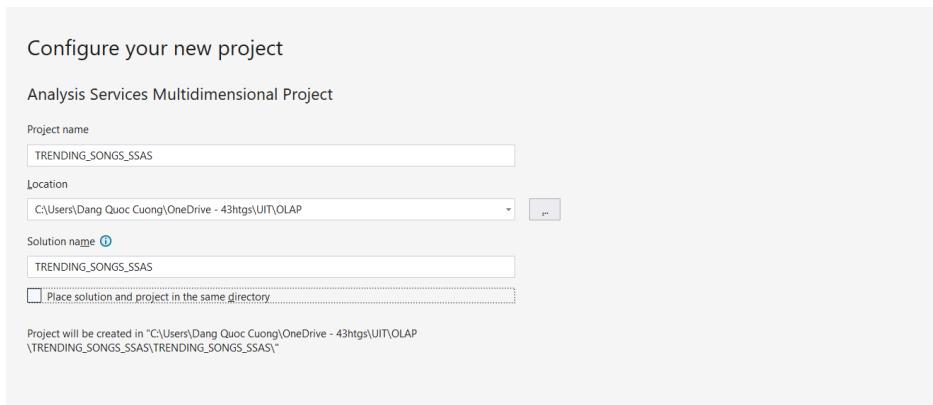
Bước 1: Mở Visual Studio và chọn “Create a new project”.



Bước 2: Chọn “Analysis Services Multidimensional and Data Mining Project” và nhấn “Next”.

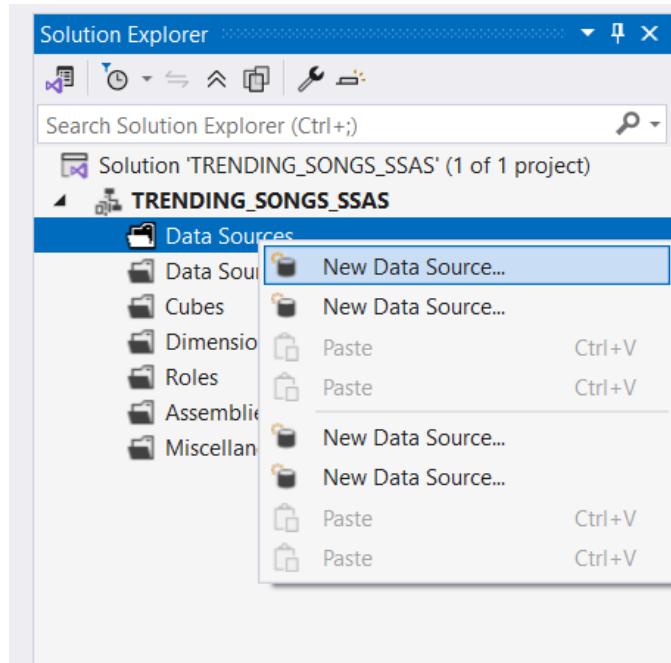


Bước 3: Đặt tên cho project, chọn vị trí lưu trữ và nhấn “Create”.

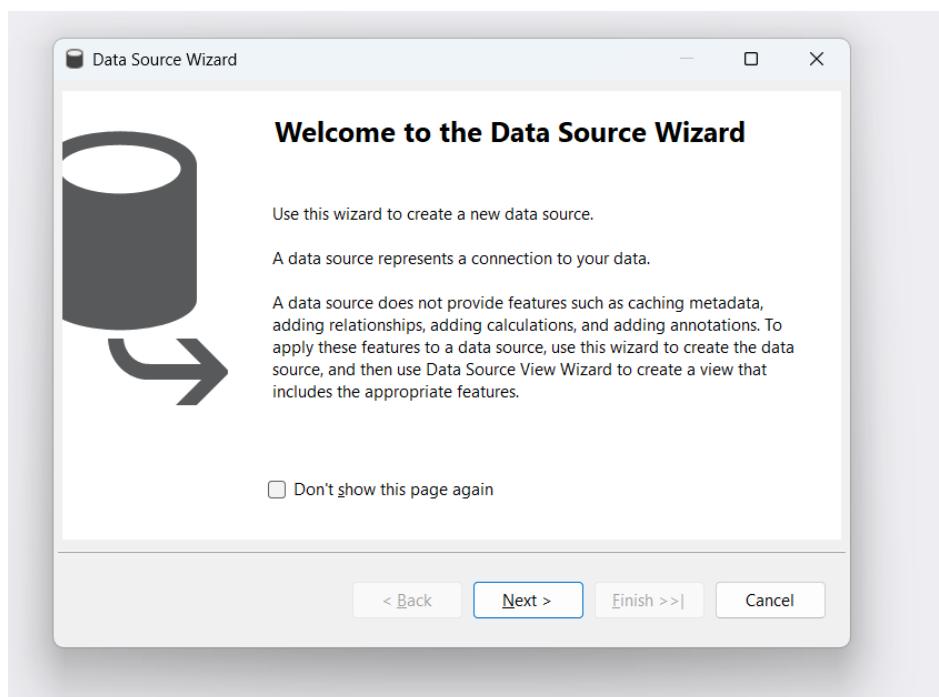


3.2 Xác định dữ liệu nguồn (Data Source)

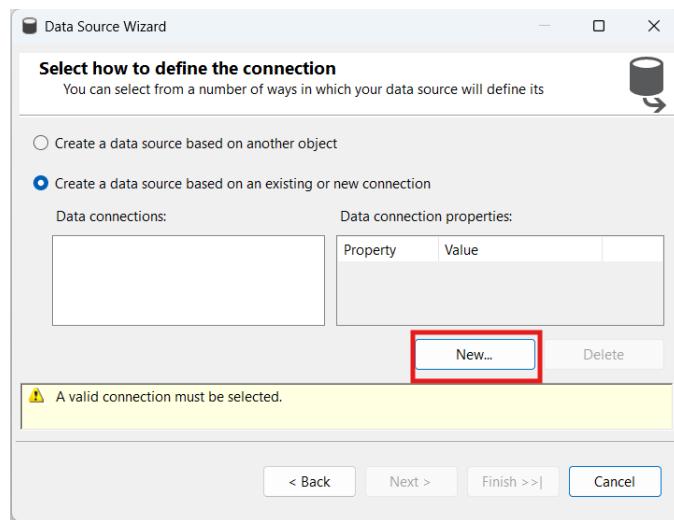
Bước 1: Tại Solution Explorer, right-click vào thư mục Data Sources, chọn "New Data Source".



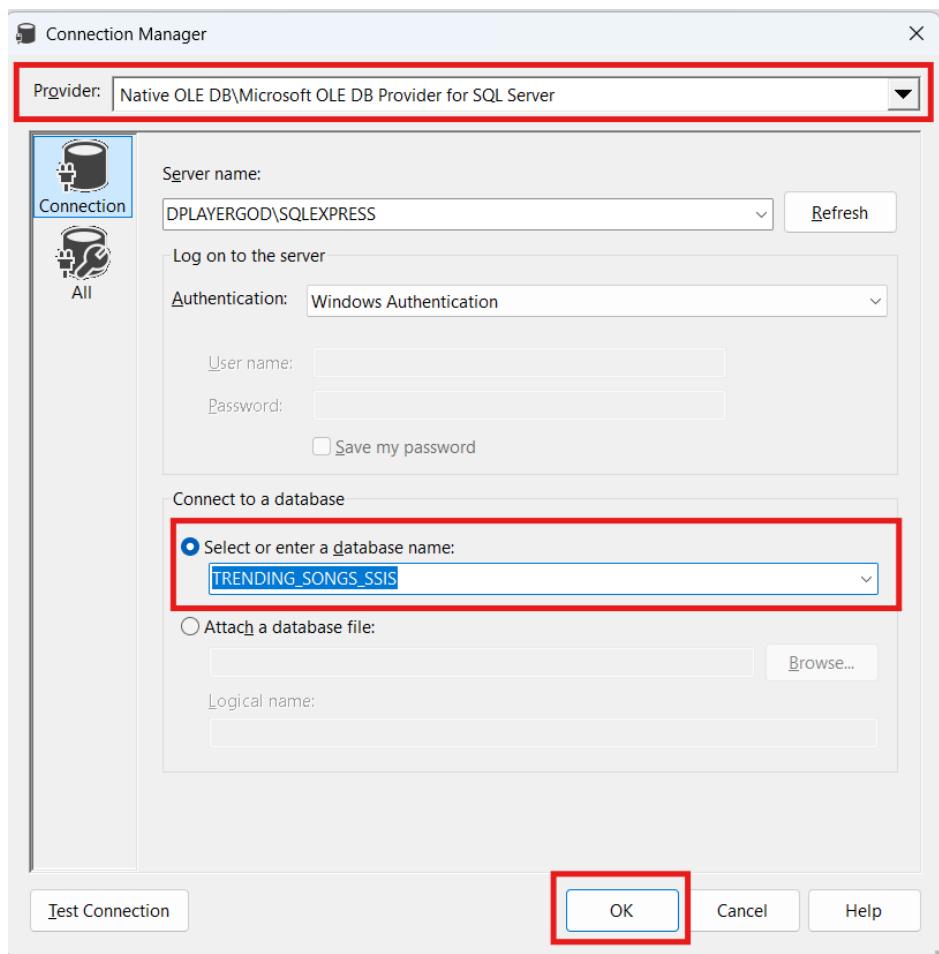
Bước 2: Hộp thoại Data Source Wizard xuất hiện, nhấn “Next”.



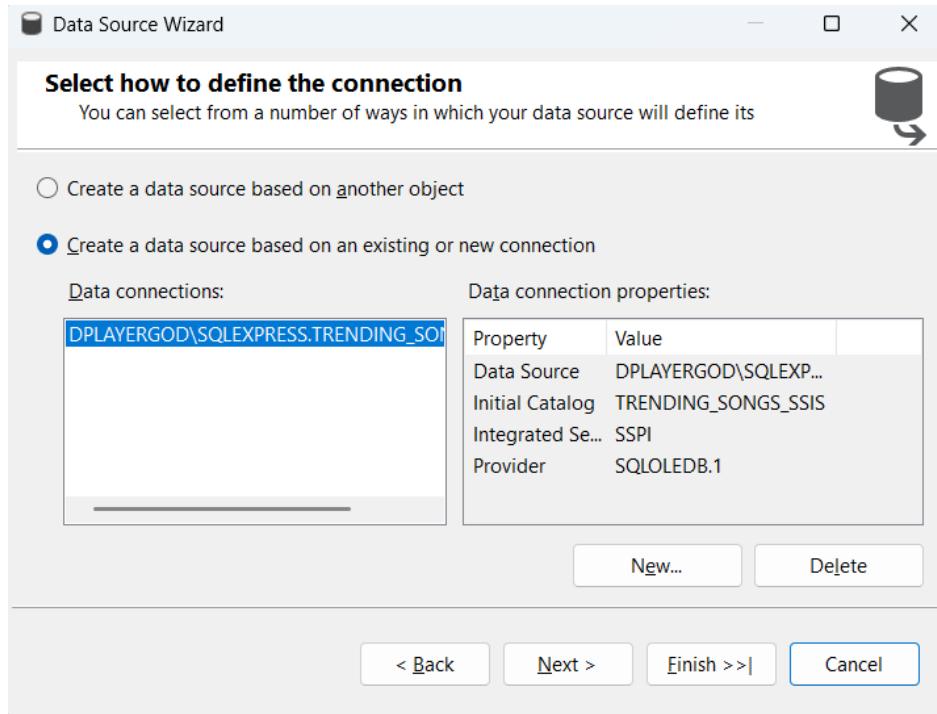
Bước 3: Chọn “New” để tạo kết nối mới đến cơ sở dữ liệu nguồn.



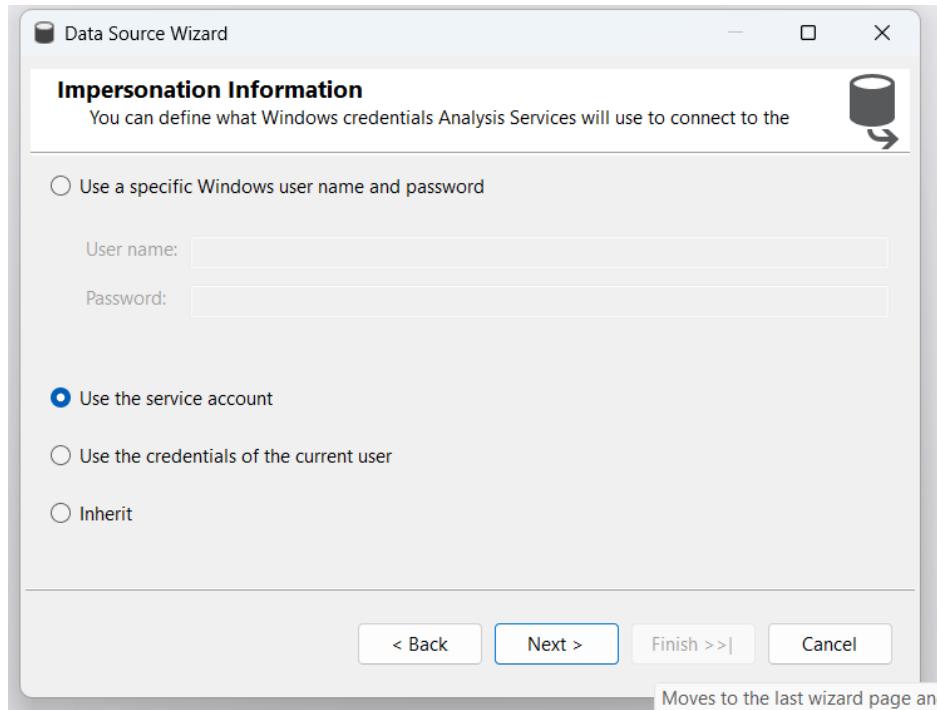
Bước 4: Trong hộp thoại Connection Manager, nhập thông tin kết nối đến cơ sở dữ liệu nguồn và nhấn “OK”.



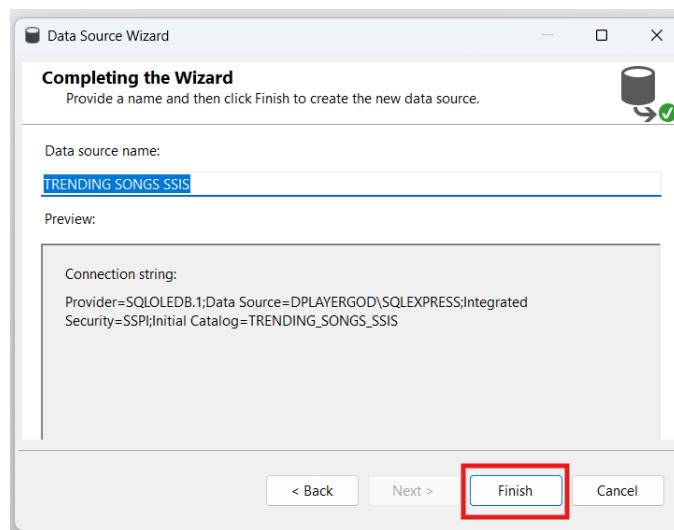
Bước 5: Quay lại Data Source Wizard, nhấn “Next” và sau đó nhấn “Finish” để hoàn tất việc tạo Data Source.



Bước 6: Chọn “Use the service account”, sau đó chọn "Next" để tiếp tục.

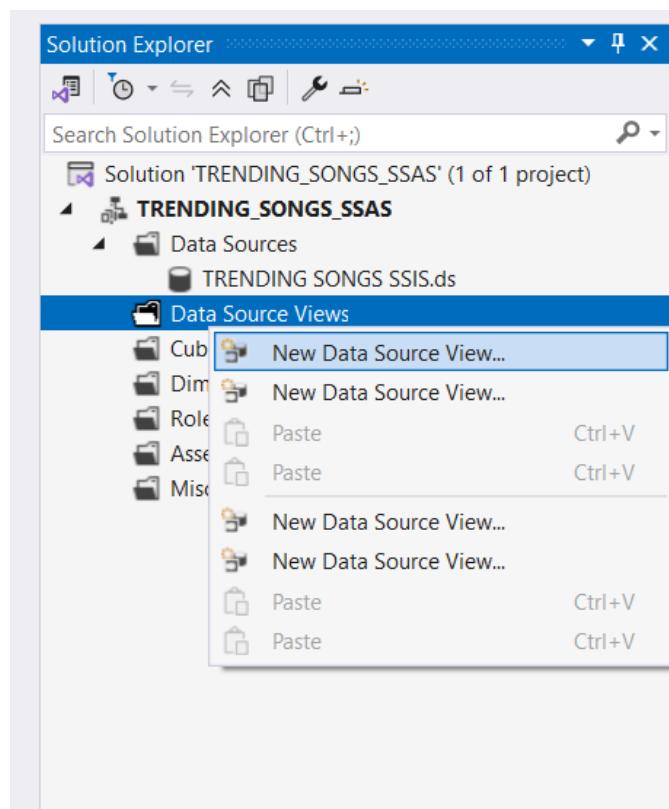


Bước 7: Nhấn “Finish” để hoàn tất việc tạo Data Source.

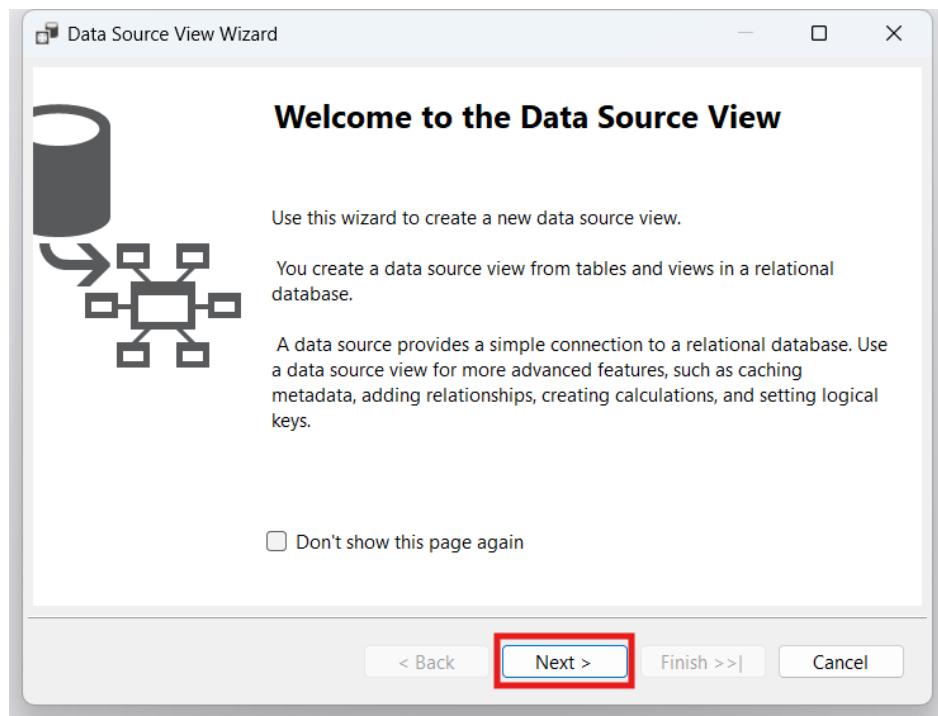


3.3 Xác định khung nhìn dữ liệu nguồn (Data Source View)

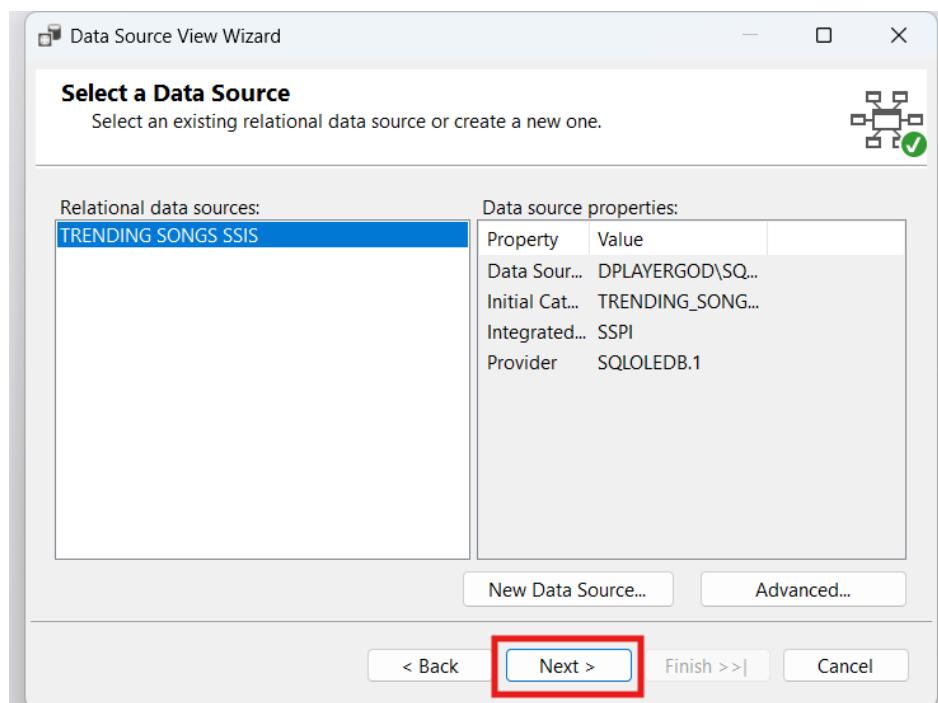
Bước 1: Tại Solution Explorer, right-click vào thư mục Data Source Views, chọn "New Data Source View".



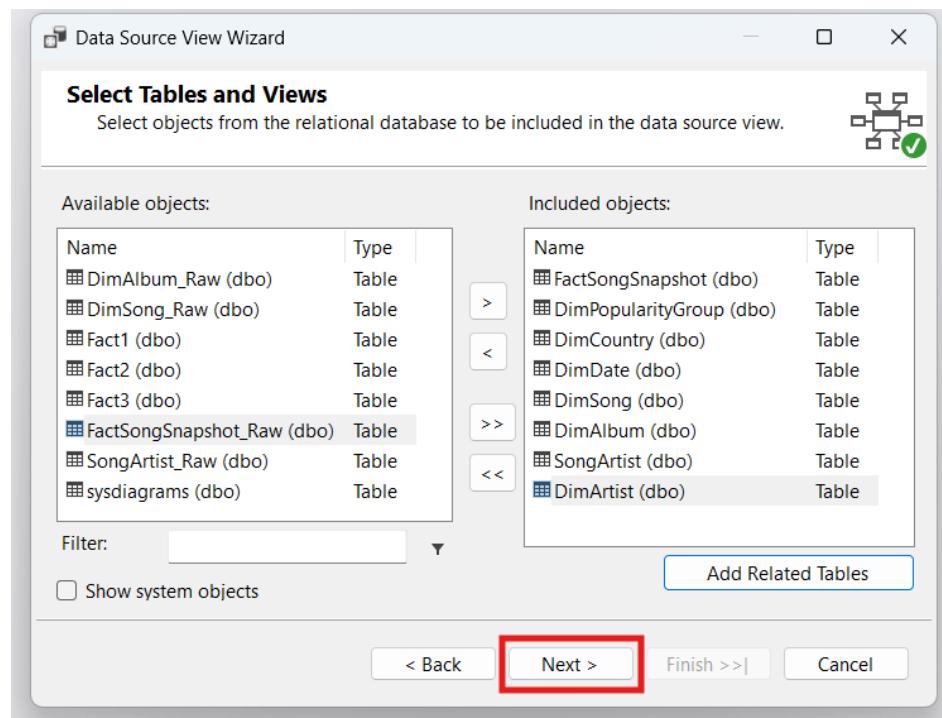
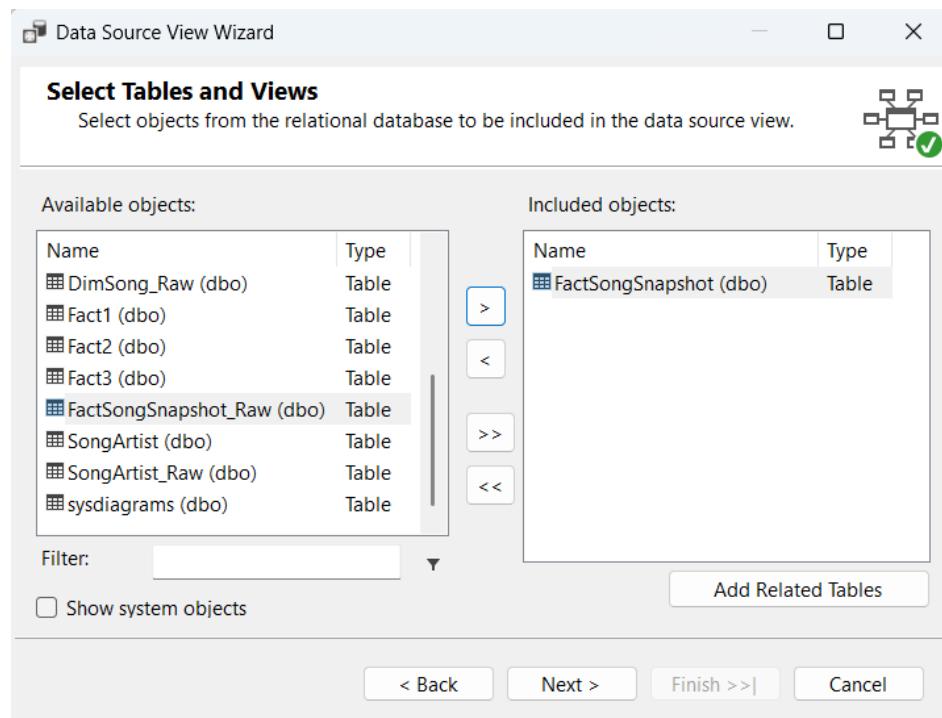
Bước 2: Hộp thoại Data Source View Wizard xuất hiện, nhấn “Next”.



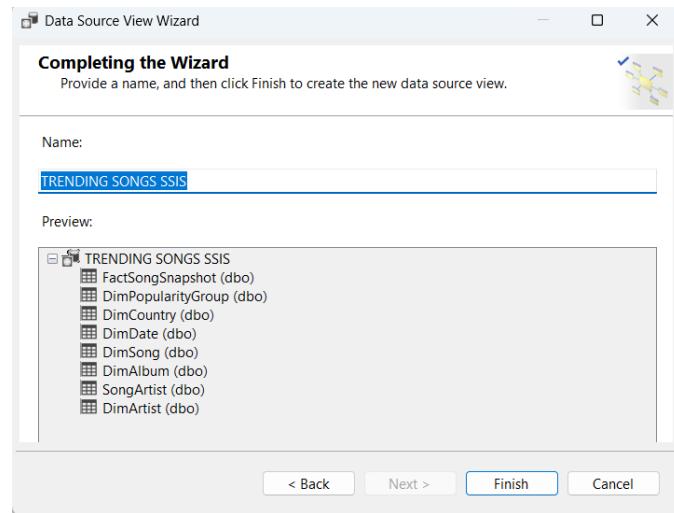
Bước 3: Chọn Data Source đã tạo ở bước trước và nhấn “Next”.



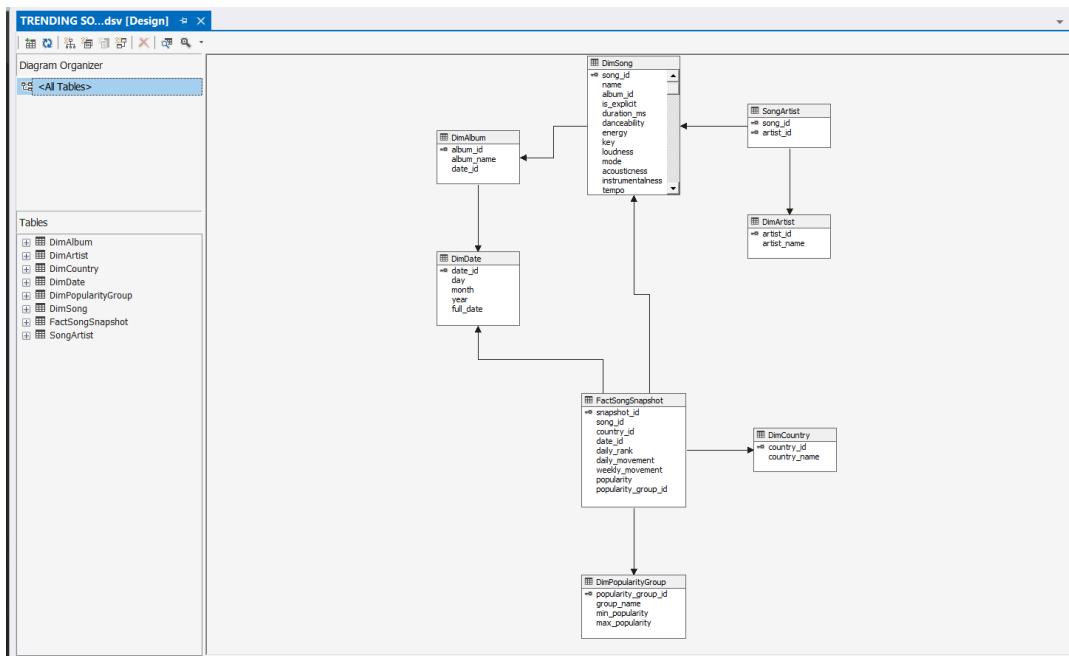
Bước 4: Chọn các bảng cần thiết từ danh sách Available Tables và nhấn nút mũi tên để chuyển chúng sang danh sách Included Tables. Sau đó, nhấn “Next”.



Bước 5: Nhấn “Finish” để hoàn tất việc tạo Data Source View.



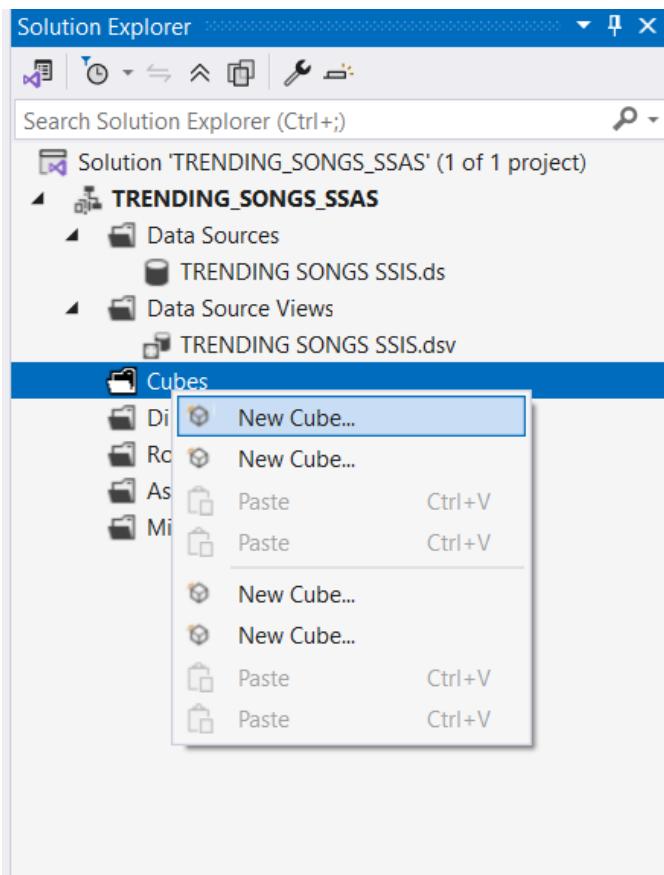
Bước 6: Kiểm tra Data Source View đã tạo trong Solution Explorer.



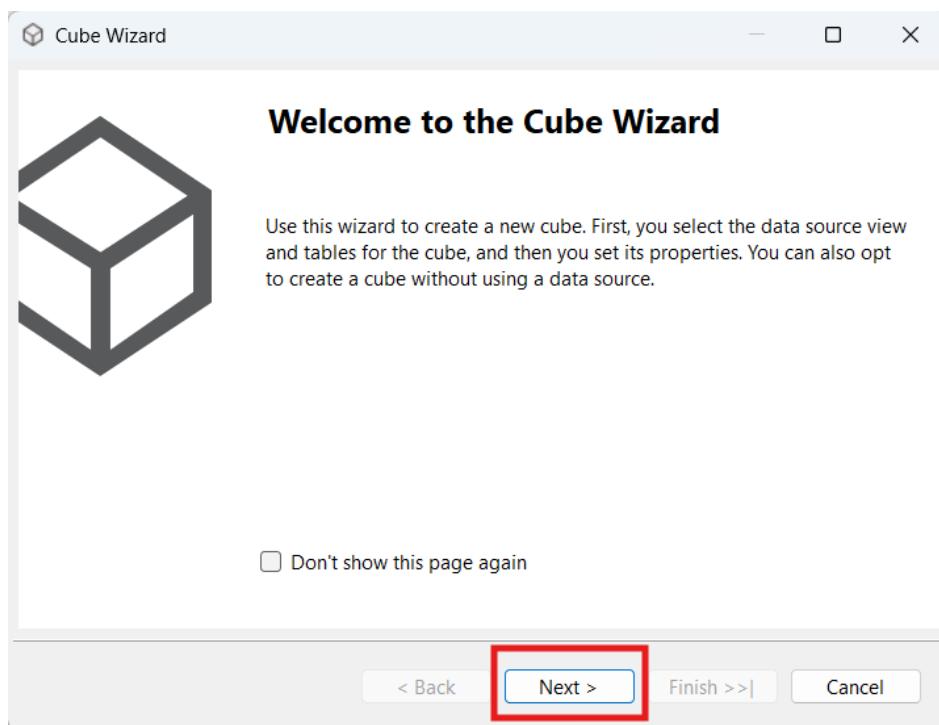
3.4 Xây dựng các khối (Cube) và Deploy Cube

3.4.1 Tạo Cube và Dimension

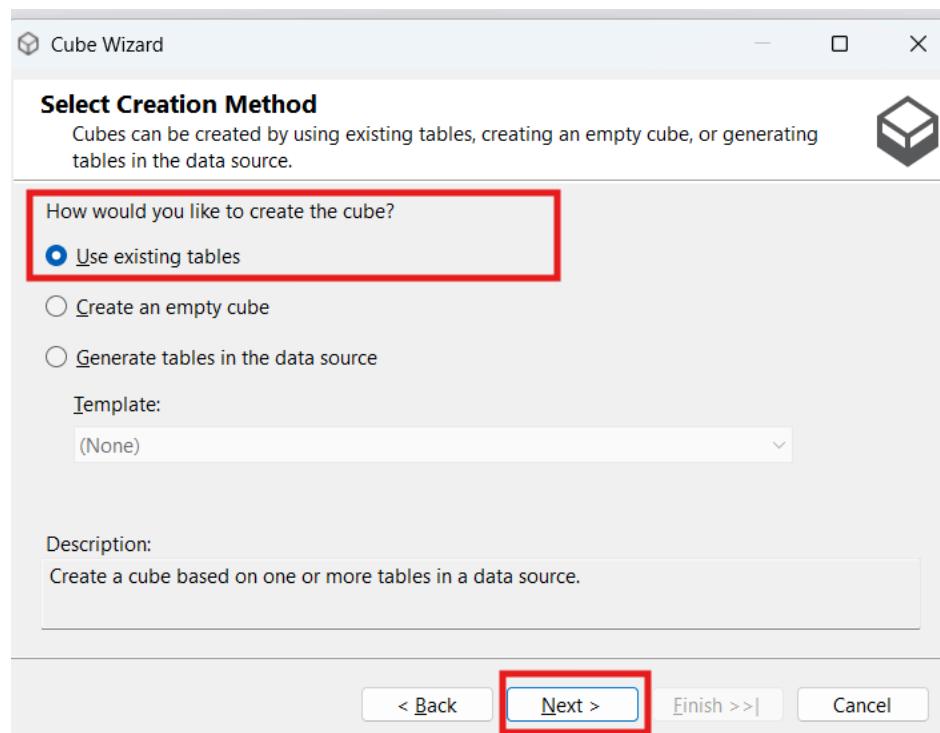
Bước 1: Tại Solution Explorer, right-click vào thư mục Cubes, chọn "New Cube".



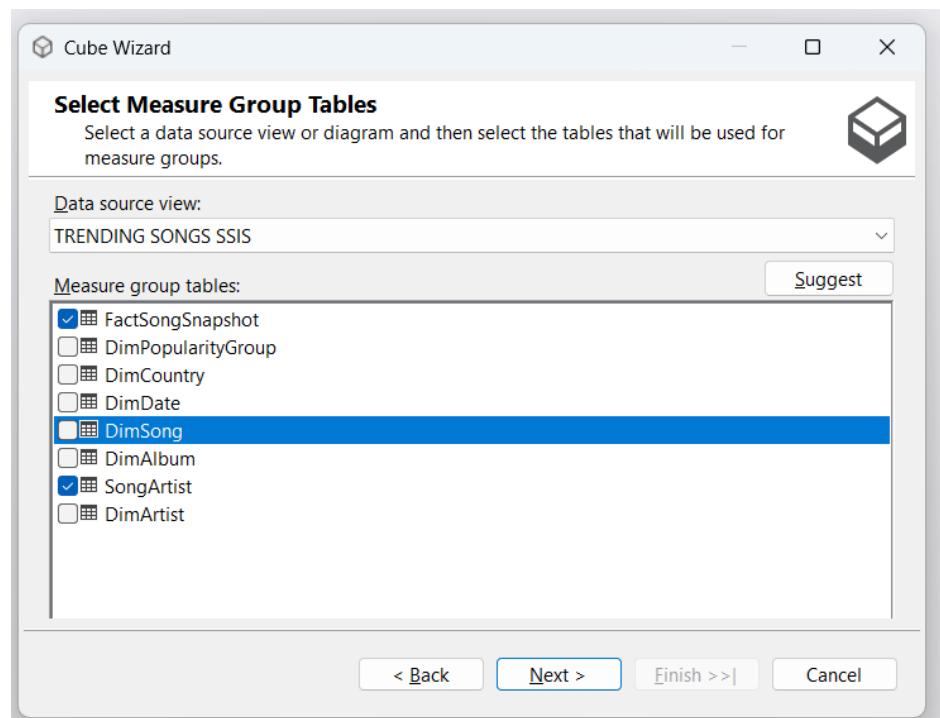
Bước 2: Hộp thoại Cube Wizard xuất hiện, nhấn “Next”.



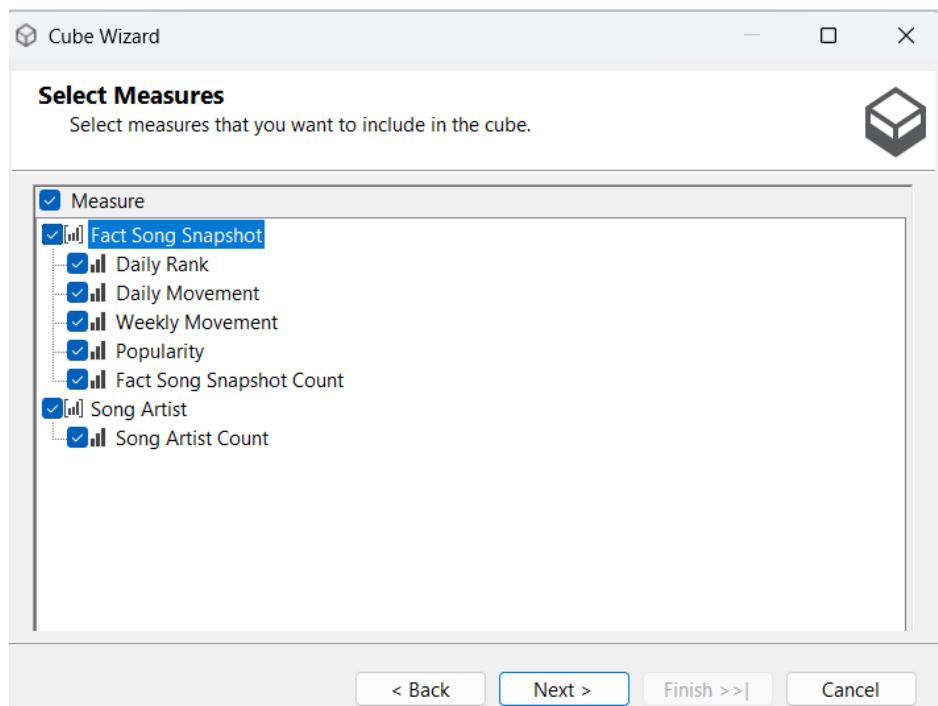
Bước 3: Chọn “Use existing tables” và nhấn “Next”.



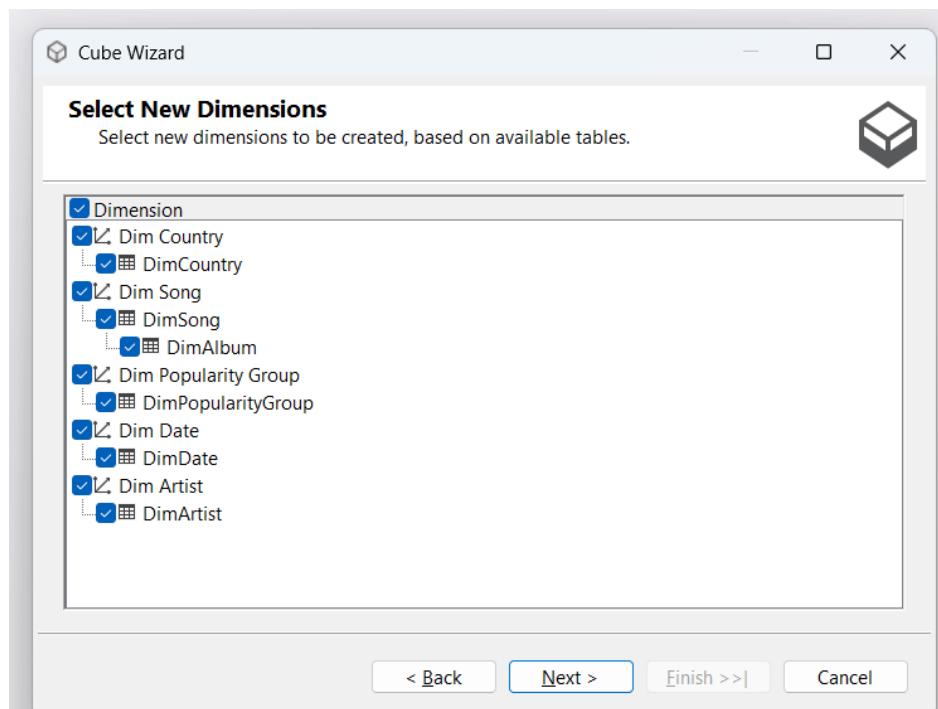
Bước 4: : Chọn bảng Fact và các bảng Bridge để phân chia các measure group.



Bước 5: Chọn những độ đo để xuất, sau đó chọn Next để tiếp tục



Bước 6: Chọn các bảng Dimension cần thiết và nhấn nút mũi tên để chuyển chúng sang danh sách Selected Dimensions. Sau đó, nhấn “Next”.



Bước 7: Nhấn “Finish” để hoàn tất việc tạo Cube và Dimension.

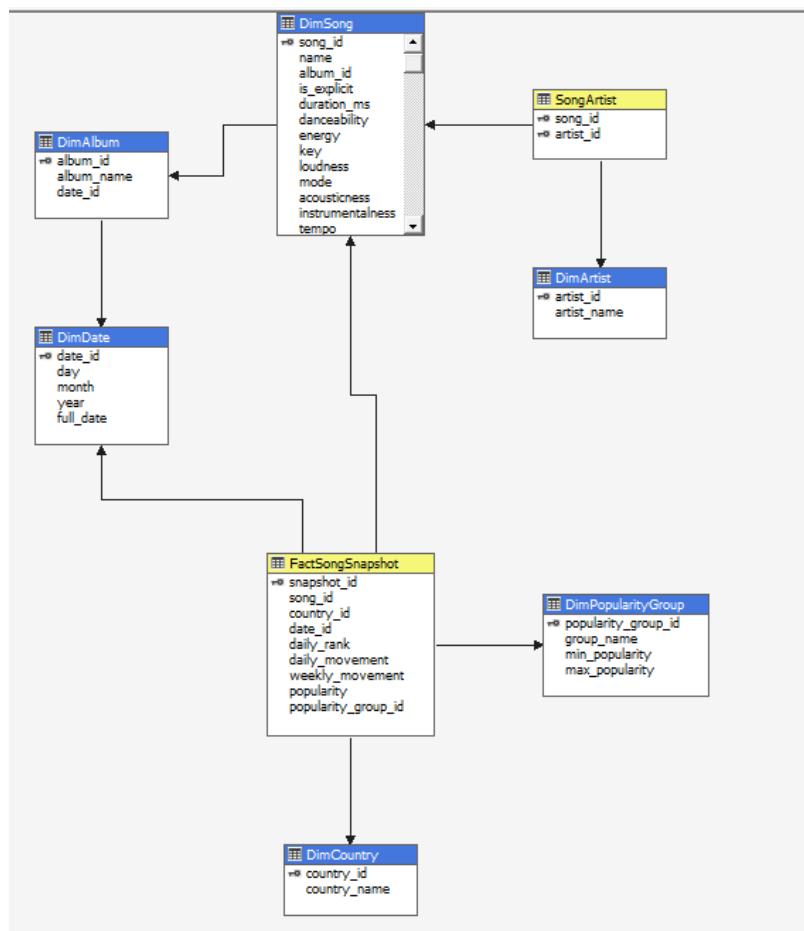
Cube name:

TRENDING SONGS SSIS

Preview:

- Measure groups
 - Fact Song Snapshot
 - Daily Rank
 - Daily Movement
 - Weekly Movement
 - Popularity
 - Fact Song Snapshot Count
 - Song Artist
 - Song Artist Count
- Dimensions
 - Dim Country
 - Dim Song
 - Dim Popularity Group
 - Dim Date
 - Dim Artist

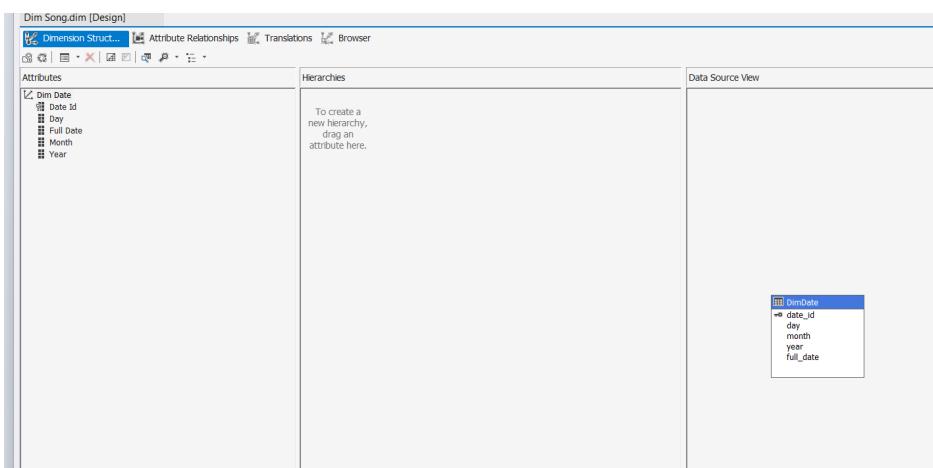
Sau khi hoàn tất, bạn sẽ thấy Cube và các Dimension đã được tạo trong Solution Explorer.



3.4.2 Thêm thuộc tính và chỉnh sửa property cho Dimension

Kéo thả các thuộc tính từ Data Source View vào Dimension Designer để thêm chúng vào Dimension.

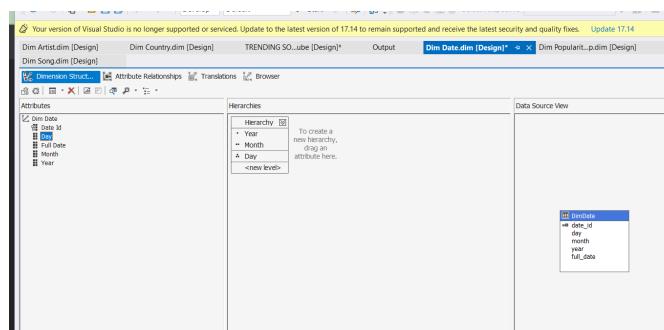
Ví dụ: Ở Dimension "DimDate", ta thêm các thuộc tính như "day", "month", "year", v.v.



3.4.3 Phân cấp trong bảng chiều

Thực hiện tạo các thuộc tính phân cấp (Hierarchies) và định nghĩa Attribute Relationships cho bảng Dim Date. Tạo Hierarchy phân cấp theo Year → Month → Day:

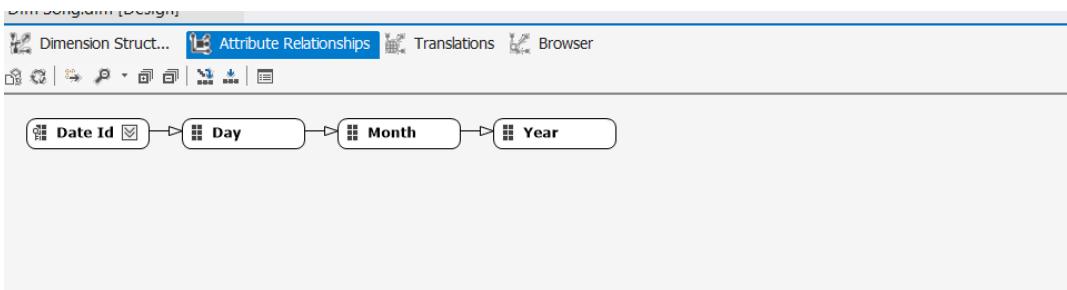
Bước 1: Kéo thả các thuộc tính từ Dimension Structure vào khu vực Hierarchies để tạo phân cấp.



Bước 2: Đặt tên cho phân cấp mới, ví dụ: "Y_M_D".

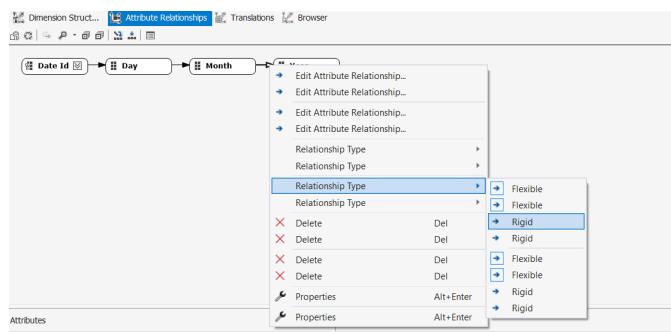


Bước 3: Tại tab Attribute Relationships, định nghĩa các mối quan hệ. Thực hiện kéo thả phân cấp từ nhỏ đến lớn theo thứ tự từ trái sang phải.



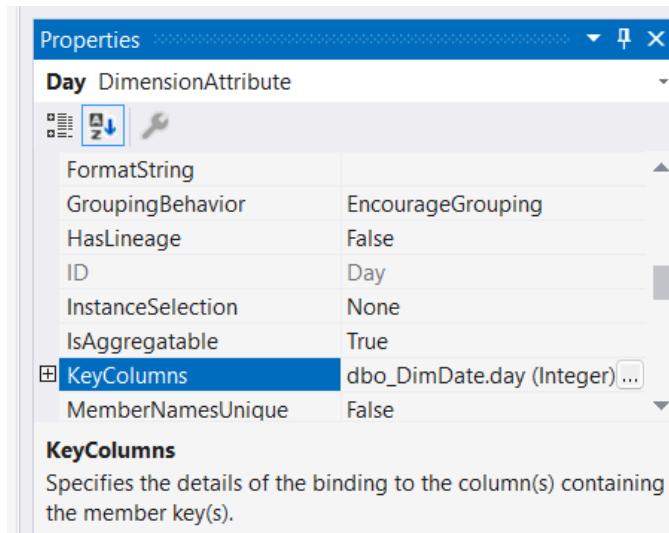
Bước 4: Chỉnh sửa Relationship Type thành Rigid.

Lí do: Vì các thuộc tính về ngày tháng năm không thay đổi theo thời gian.

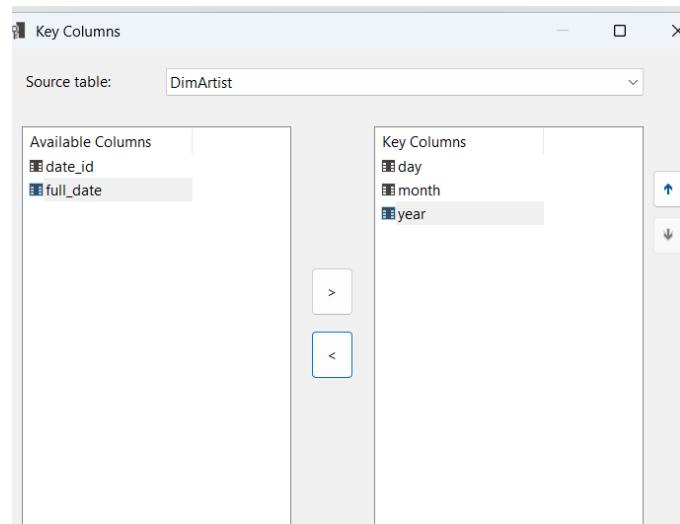


Bước 5: Chỉnh sửa cột (KeyColumns) và tên cột (Name Column) của thuộc tính Day. Vì thuộc tính này sẽ lấy khóa cột gồm chính nó và những thuộc tính cấp cao hơn.

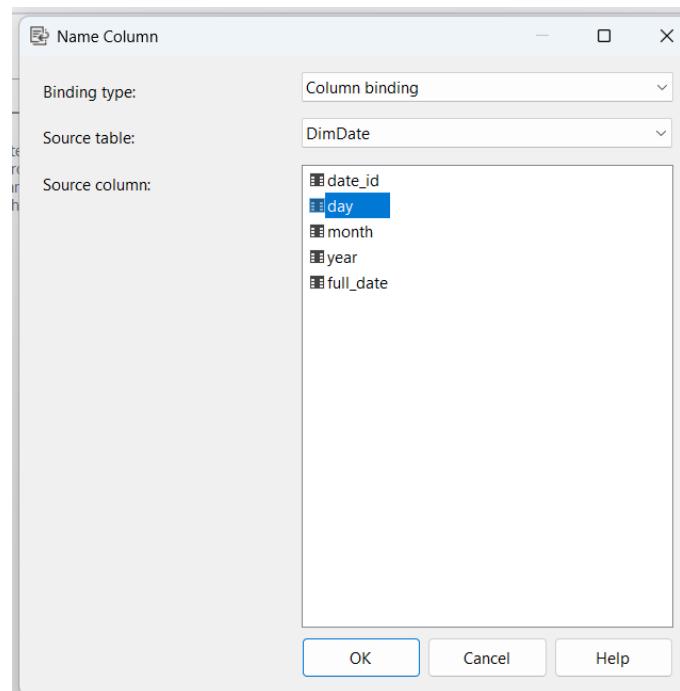
Chuyển sang tab Dimension Structure, cột Attributes, right-click vào thuộc tính Day và chọn Properties.



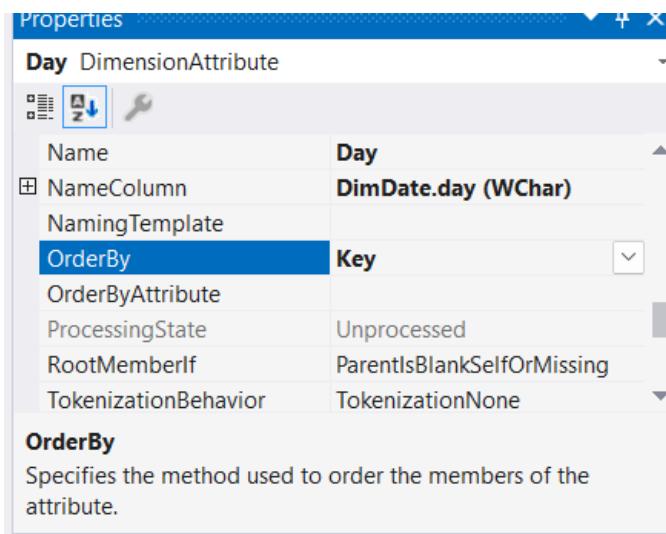
Tại cửa sổ Properties, chọn KeyColumns. Thêm các thuộc tính cấp cao hơn vào KeyColumns, sau đó chọn OK để hoàn tất:



Tại cửa sổ Properties, chọn NameColumn và chọn tên thuộc tính sẽ hiển thị trên Hierachy là "day".



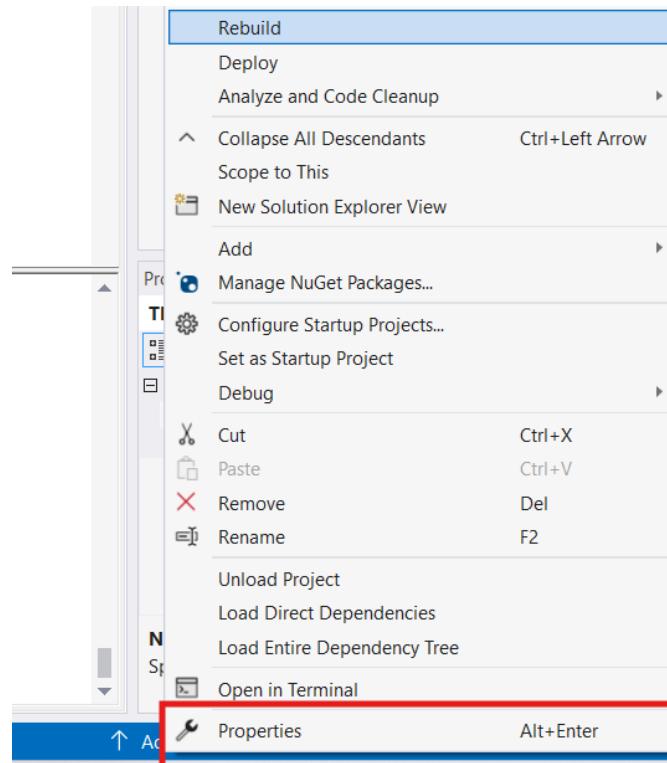
Chỉnh thuộc tính OrderBy thành Key để "day" được sắp xếp theo thứ tự tăng dần:



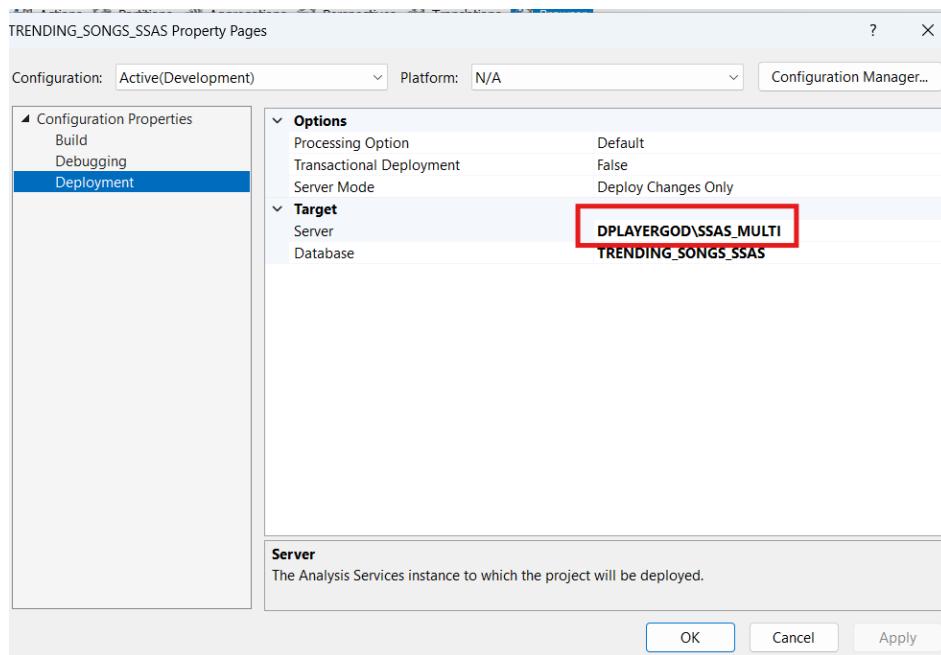
Bước 6: Lặp lại các bước trên cho các thuộc tính Month để hoàn tất việc tạo phân cấp "Y_M_D".

3.4.4 Deploy project SSAS

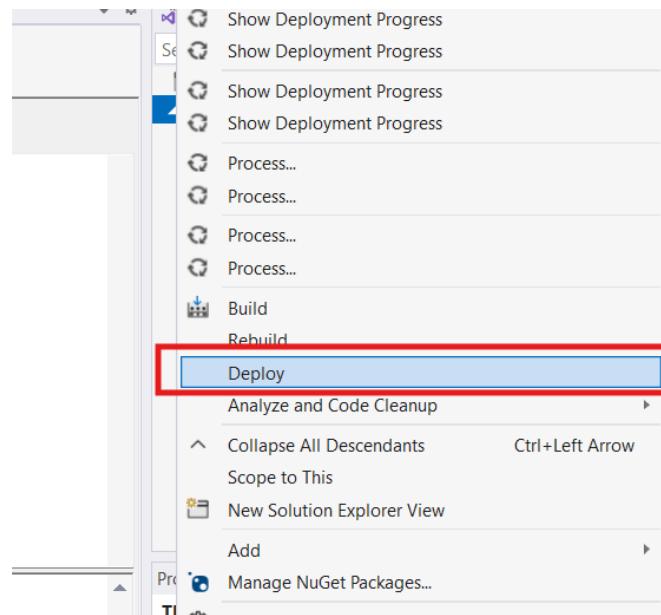
Bước 1: Tại Solution Explorer, right-click ở tên project và chọn Properties để chỉnh sửa kết nối đến Analysis Service của SQL Server.



Bước 2: Ở cửa sổ vừa mở, đi đến Deployment và thực hiện đổi tên Server theo tên trong SQL Server.



Bước 3: Right-click vào project đang hiện hành và chọn Deploy project.



3.5 Thực hiện các câu truy vấn sử dụng SSAS, Pivot table, ngôn ngữ MDX

3.5.1 Câu truy vấn 1 - Sử dụng SSAS

Nội dung câu truy vấn: Tìm ra 10 bài hát có thứ hạng cao nhất (daily_rank từ 1–10) tại mỗi quốc gia, vào ngày 01/01/2025. Nhóm dữ liệu theo country và lọc các bản ghi có daily_rank ≤ 10 và có snapshot_date là 01/01/2025.

Bước 1: Trong Cube, vào mục Calculation, tạo mới nameset:



Bước 2: Sang Browser, chỉnh sang Design Mode và thêm đoạn MDX sau vào ô Query.

Country Name	Name	Daily Rank
TW	Whip...	10
TW	Who	2
TW	Wint...	1
TW	Wint...	8
VN	3D f...	5
VN	APT	8
VN	bính ...	7
VN	Exit ...	6
VN	Mát ...	3
VN	Seve...	1
VN	toxic...	10
VN	Who	2
VN	Wint...	4
VN	Wro...	9

3.5.2 Câu truy vấn 1 - Sử dụng Pivot Table trong Excel

Artist	Song	Country	Daily Rank
APT	DRIP	VN	5
DRIP	HOME SWEET HOME feat TAEYANG DAESUNG	VN	8
DRIP	Runnin' Wild	VN	7
toxic till the end	toxic till the end	VN	3
Whiplash	Whiplash	VN	10
Who	Who	VN	2
Winter Ahead with PARK HYO SHIN	Winter Ahead with PARK HYO SHIN YUNSEOKCHEOL TRIO Ver	VN	1
Winter Ahead with PARK HYO SHIN	Winter Ahead with PARK HYO SHIN YUNSEOKCHEOL TRIO Ver	VN	8
3D feat Jack Harlow	3D feat Jack Harlow	VN	5
APT	APT	VN	9
bính	bính	VN	6
East Side	East Side	VN	3
Mát	Mát	VN	3
Seven feat Latto Explicit Ver	Seven feat Latto Explicit Ver	VN	1
toxic till the end	toxic till the end	VN	10
Who	Who	VN	2
Winter Ahead with PARK HYO SHIN	Winter Ahead with PARK HYO SHIN	VN	4
Wrong Times	Wrong Times	VN	9

3.5.3 Câu truy vấn 1 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
SET Top10Songs AS
GENERATE(
    [Dim Country]. [Country Name]. [Country Name]. MEMBERS,
    {
        [Dim Country]. [Country Name]. CURRENTMEMBER
    } *
FILTER(
    NONEMPTY(
        [Dim Song]. [Name]. [Name]. MEMBERS,
        ( [Measures]. [Daily Rank],
            [Dim Date]. [Full Date]. CURRENTMEMBER,
            [Dim Country]. [Country Name]. CURRENTMEMBER )
    ),
    ( [Measures]. [Daily Rank],
        [Dim Date]. [Full Date]. CURRENTMEMBER,
        [Dim Country]. [Country Name]. CURRENTMEMBER ) <= 10
)
)

SELECT
{ [Measures]. [Daily Rank] } ON COLUMNS,
NON EMPTY Top10Songs ON ROWS
FROM
[TRENDING SONGS SSIS]
WHERE
( [Dim Date]. [Full Date]. [2025-01-01] );

```

The screenshot shows the SSAS Browser interface with a query editor at the top and a results grid below. The query editor contains the following DAX code:

```

    ) <= 10
)
)
SELECT
{ [Measures].[Daily Rank] } ON COLUMNS,
NON EMPTY [Top10Songs] ON ROWS
FROM [TRENDING SONGS SSIS]
WHERE ([Dim Date].[Full Date].[2025-01-01]);

```

The results grid displays a list of songs with their daily ranks. A red box highlights a specific group of rows:

		Daily Rank
TW	Running Wild	3
TW	toxic till the end	5
TW	Whiplash	10
TW	Who	2
TW	Winter Ahead with PARK HYO SHIN	1
TW	Winter Ahead with PARK HYO SHIN YUNSFOKCHEOL TRIO Ver	8
VN	3D feat Jack Harlow	5
VN	APT	8
VN	bình yên	7
VN	Exit Sign	6
VN	Mặt Kết Nối	3
VN	Seven feat Latto Explicit Ver	1
VN	toxic till the end	10
VN	Who	2
VN	Winter Ahead with PARK HYO SHIN	4
VN	Wrong Times	9

Nhận xét: Mặc dù đáp ứng yêu cầu của truy vấn. Tuy nhiên câu truy vấn khá phức tạp và khó áp dụng việc kéo thả trong SSAS Browser.

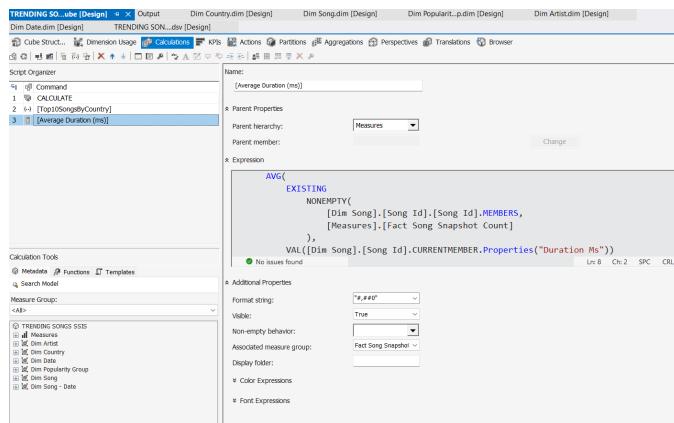
3.5.4 Câu truy vấn 2 - Sử dụng SSAS

Nội dung câu truy vấn: Thống kê thời lượng trung bình (duration_ms) của các bài hát xuất hiện trên bảng xếp hạng trong một giai đoạn cụ thể, nhóm theo album_name.

Ví dụ: Tính thời lượng trung bình của các bài hát xuất hiện trong bảng xếp hạng vào tháng 1 năm 2025, nhóm theo tên album.

Bước 1: Trong Cube, vào mục Calculation, tạo mới Calculated Member:

Báo cáo đồ án



Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Album Name	Average Duration (ms)
"Aaj Ki Raat (From ""Street 2"")	228.620
"Ayay Na (From ""Street 2"")	178.780
"Akhiyan Gulab (From ""Teri Baaton Mein Aisa Ujhha Jiyaa"")	171.147
"ANH TRAI ""SAY HIT"" (Live Stage 2")	245.454
"ANH TRAI ""SAY HIT"" (Live Stage 3")	252.428
"ANH TRAI ""SAY HIT"" (Live Stage 4")	276.529
"ANH TRAI ""SAY HIT"" Chung Kêt 1"	210.000
"ANH TRAI ""SAY HIT"" Tập 14"	252.625
"Chai Diye Tum Kahan (From ""Kabhi Main Kabhi Tum"")	275.200
"Chai Diye Tum Kahan (From ""Devara Part 1"")	223.520
"Di Gita Tròi Rực Rỡ (From ""Di Gita Tròi Rực Rỡ"")	220.838
"Godari Gattu Meetha (From ""Sankranthi Vasthunam"")	250.083
"Jaani Samjho Na (From ""Bhoot Bhulayaa 3"")	212.108
"Kale Hua (From ""Kabir Singh"")	234.722
"Khobosurat (From ""Street 2"")	244.120
"Mere Sohneya (From ""Kabir Singh"")	193.358
"Naina (From ""Crew"")	180.000
"O Saathi (From ""Baaghi 2"")	251.818
"Peelings (From ""Ujhaa 2 The Rule"") [TELUGU]"	247.137
"Raajha (From ""Do Patti"")	240.000
"Ranjha (From ""Shershaah"")	228.858
"Sajni (From ""Lapataa Ladies"")	170.044
"Satranga (From ""ANIMAL"")	271.169
"Tauhi Khabab (From ""Bad News"")	188.166
"The Trouble is... (From ""Misaal kaao"")	207.110
"Tu Hain Toh Main Hoon (From ""Sky Force"")	190.714
"Tuje Kina Chahne Lage (From ""Kabir Singh"")	247.792
"Tuje Se (From ""Teri Baaton Mein Aisa Ujhha Jiyaa"")	284.779
"Tumhein Rehnaa Sun (From ""Street 2"")	264.069
"別重逢 (電影)?" 别重逢"主视觉	230.000
	151.600

3.5.5 Câu truy vấn 2 - Sử dụng Pivot Table trong Excel

3.5.6 Câu truy vấn 2 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

WITH

```

MEMBER [Measures]. [Average Duration (ms)] AS
    AVG(
        EXISTING
        NONEMPTY(
            [Dim Song]. [Song Id]. [Song Id]. MEMBERS,
            [Measures]. [Fact Song Snapshot Count]
        ),
        VAL([Dim Song]. [Song Id]. CURRENTMEMBER.Properties("Duration Ms"))
    ),
    FORMAT_STRING = "#,##0"

SELECT
    { [Measures]. [Average Duration (ms)] } ON COLUMNS,
    NON EMPTY
        [Dim Song]. [Album Name]. [Album Name]. MEMBERS
    ON ROWS

FROM
    [TRENDING SONGS SSIS]

WHERE
    ( [Dim Date]. [Y_M_D]. [Month] .&[1]&[2025] );

```

The screenshot shows the SSAS Browser interface with a query editor at the top and a results grid below. The query editor contains the MDX code from the previous section. The results grid displays a list of songs along with their average duration in milliseconds.

	Average Duration (ms)
"Aaj Ki Raat (From ""Stree 2"")"	228.620
"Aayi Nai (From ""Stree 2"")"	178.780
"Akhiyaan Gulaab (From ""Teri Baaton Mein Aisa Ujha Jiya"")"	171.147
"ANH TRAI ""SAY HI"" (Live Stage 2)"	245.454
"ANH TRAI ""SAY HI"" (Live Stage 3)"	252.428
"ANH TRAI ""SAY HI"" (Live Stage 4)"	276.529
"ANH TRAI ""SAY HI""; Chung Kết 1"	210.000
"ANH TRAI ""SAY HI""; Tập 14"	252.625
"Chal Diye Tum Kahan (From ""Kabhi Main Kabhi Tum"")"	275.200
"Chuttamalle (From ""Devara Part 1"")"	222.063
"Đi Giữa Trời Rực Rỡ (From ""Đi Giữa Trời Rực Rỡ"")"	220.839
"Godari Gattu Meedha (From ""Sankranthiki VasthuNam"")"	250.083
"Jaana Samjhona (From ""Bhool Bhulaiyaa 3"")"	212.108
"Kaise Hua (From ""Kabir Singh"")"	234.722
"Khoobsurat (From ""Stree 2"")"	244.583
"Mere Sohneya (From ""Kabir Singh"")"	193.355
"Naina (From ""Crew"")"	180.000

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.7 Câu truy vấn 3 - Sử dụng SSAS

Nội dung câu truy vấn: So sánh danceability và energy trung bình của các bài hát có mặt trên bảng xếp hạng trong khoảng thời gian nhất định, nhóm theo mode (trưởng/thứ) để xem xu hướng âm nhạc của từng chế độ.

Ví dụ: So sánh danceability và energy trung bình của các bài hát xuất hiện trong bảng xếp hạng vào tháng 1, 2, 3 năm 2025, nhóm theo chế độ (mode).

Bước 1: Trong Cube, vào mục Calculation, tạo mới Calculated Member:

```


    Name: [Average Danceability]
    Parent Properties: Measures
    Parent member: 
    Expression:
        AVG(
            EXISTING
            NONEMPTY(
                [Dim Song].[Song Id].[Song Id].MEMBERS,
                [Measures].[Fact Song Snapshot Count]
            ),
            /* Lấy property 'Danceability' của bài hát */
            VAL([Dim Song].[Song Id].CURRENTMEMBER.Properties("Danceability"))
        )
        No issues found
    Additional Properties:
        Format string: "#.###"
        Value: True
        Non-empty behavior: Associated measure group: Fact Song Snapshot
        Display folder: Color Expressions
        Font Expressions


```

```


    Name: [Average Energy]
    Parent Properties: Measures
    Parent member: 
    Expression:
        AVG(
            EXISTING
            NONEMPTY(
                [Dim Song].[Song Id].[Song Id].MEMBERS,
                [Measures].[Fact Song Snapshot Count]
            ),
            /* Lấy property 'Energy' của bài hát */
            VAL([Dim Song].[Song Id].CURRENTMEMBER.Properties("Energy"))
        )
        No issues found
    Additional Properties:
        Format string: "#.###"
        Value: True
        Non-empty behavior: Associated measure group: Fact Song Snapshot
        Display folder: Color Expressions
        Font Expressions


```

Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Mode	Average Danceability	Average Energy
0	0.64048480476467	0.6351374814...
1	0.607074516060105	0.5979336822...

3.5.8 Câu truy vấn 3 - Sử dụng Pivot Table trong Excel

The screenshot shows an Excel spreadsheet with a Pivot Table. The Pivot Table has 'Dim Date.Y_M_D (Nhiều khoản mục)' in row 1, 'Nhận Hàng' in row 2, and 'Tổng Cuối' in row 6. The columns are labeled 'Average Danceability' and 'Average Energy'. The data rows show values for months 0 and 1, and summary totals at the bottom.

	A	B	C	D
1	Dim Date.Y_M_D (Nhiều khoản mục)			
2	Nhận Hàng	Average Danceability	Average Energy	
3	0	0.640	0.635	
4	1	0.607	0.598	
5	Tổng Cuối	0.005	0.013	
6				
7				
8				
9				

3.5.9 Câu truy vấn 3 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
MEMBER [Measures]. [Average Danceability] AS
    AVG(
        EXISTING
        NONEMPTY(
            [Dim Song]. [Song Id]. [Song Id]. MEMBERS,
            [Measures]. [Fact Song Snapshot Count]
        ),
        VAL([Dim Song]. [Song Id]. CURRENTMEMBER.Properties("Danceability"))
    ),
    FORMAT_STRING = "0.000"

MEMBER [Measures]. [Average Energy] AS
    AVG(
        EXISTING
        NONEMPTY(
            [Dim Song]. [Song Id]. [Song Id]. MEMBERS,
            [Measures]. [Fact Song Snapshot Count]
        ),
        VAL([Dim Song]. [Song Id]. CURRENTMEMBER.Properties("Energy"))
    ),
    FORMAT_STRING = "0.000"

SELECT
{
    [Measures]. [Average Danceability],
    [Measures]. [Average Energy]
} ON COLUMNS,

```

Báo cáo đồ án

```

NON EMPTY
    [Dim Song].[Mode].[Mode].MEMBERS
ON ROWS

FROM
    [TRENDING SONGS SSIS]
WHERE
    ( [Dim Date].[Y_M_D].[Month].&[1]&[2025] : [Dim Date].[Y_M_D].[Month].&[3]&[2025] );

```

	Average Danceability	Average Energy
0	0.640	0.635
1	0.607	0.598

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.10 Câu truy vấn 4 - Sử dụng SSAS

Nội dung câu truy vấn: Xác định các nghệ sĩ nào có tác phẩm được ưa chuộng nhất trong ngày. Nhóm dữ liệu theo artists và tính điểm phổ biến trung bình (popularity).

Ví dụ: Tìm top 5 các nghệ sĩ có điểm phổ biến trung bình cao nhất vào ngày 01/01/2025.

Bước 1: Trong Cube, vào mục Calculation, tạo mới nameset:

Báo cáo đồ án

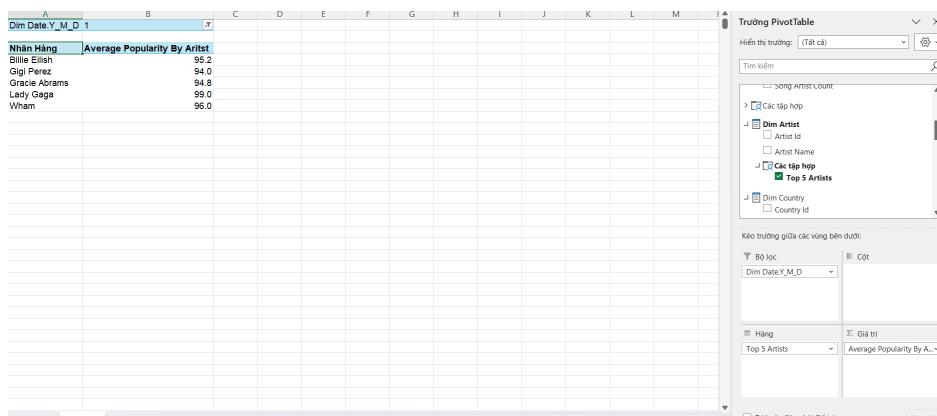
Bước 2: Tạo mới Named Set để lấy top 5 artists có điểm phổ biến trung bình cao nhất.

```
[Dim Artist].[Artist Name].[Artist Name].MEMBERS,
5,
[Measures].[Average Popularity By Artist]
```

Bước 3: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Artist Name	Average Popularity By Artist
Billie Eilish	95.2380952380952
Gigi Peret	94
Grace Abr...	94.8333333333333
Lady Gaga	99
Wham	96

3.5.11 Câu truy vấn 4 - Sử dụng Pivot Table trong Excel



3.5.12 Câu truy vấn 4 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
MEMBER [Measures].[Average Popularity By Aritst] AS
    DIVIDE(
        [Measures].[Popularity],
        [Measures].[Fact Song Snapshot Count]
    ),
    FORMAT_STRING = "#,##0.0"
SELECT
    { [Measures].[Average Popularity By Aritst] } ON COLUMNS,
    NON EMPTY
    TOPCOUNT(
        [Dim Artist].[Artist Name].[Artist Name].MEMBERS,
        5,
        [Measures].[Average Popularity By Aritst]
    ) ON ROWS
FROM
    [TRENDING SONGS SSIS]
WHERE
    ( [Dim Date].[Y_M_D].[Day].&[1]&[1]&[2025] );

```

Average Popularity By Artist	
Lady Gaga	99.0
Wham	96.0
Billie Eilish	95.2
Gracie Abrams	94.8
Gigi Perez	94.0

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau. Tuy nhiên trong SSAS Browser và Pivot Table, ta khó khăn trong việc sắp xếp thứ hạng cho các nghệ sĩ.

3.5.13 Câu truy vấn 5 - Sử dụng SSAS

Nội dung câu truy vấn: Xác định những bài hát có sự thay đổi thứ hạng lớn nhất (tăng hoặc giảm) trong một khoảng thời gian cụ thể. Lọc dữ liệu theo `snapshot_date` và phân tích giá trị của `daily_movement`.

Ví dụ: Tìm 10 bài hát có sự thay đổi thứ hạng lớn nhất trong tháng 1 năm 2025.

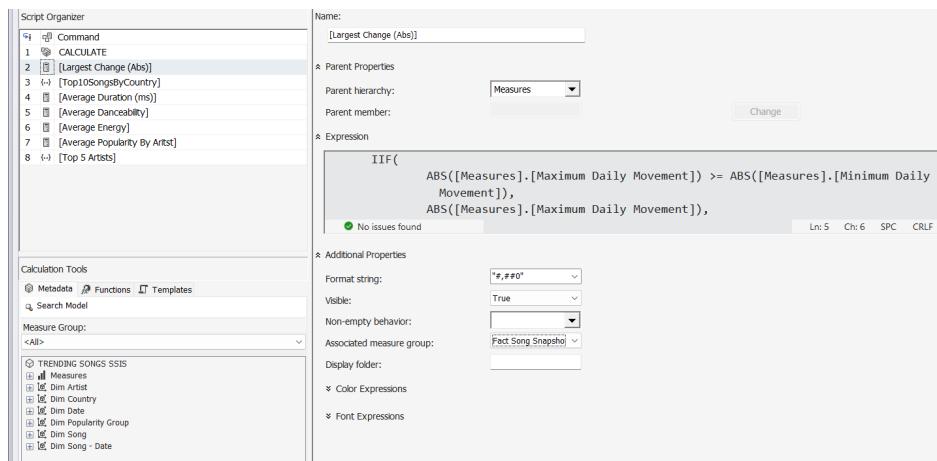
Ý tưởng: Tìm ra 10 bài hát có giá trị tuyệt đối của `daily_movement` lớn nhất trong khoảng thời gian từ 01/01/2025 đến 31/01/2025.

Bước 1: Tạo thêm hai measure từ `daily_movement` là `Minimum Daily Movement` và `Maximum Daily Movement`.

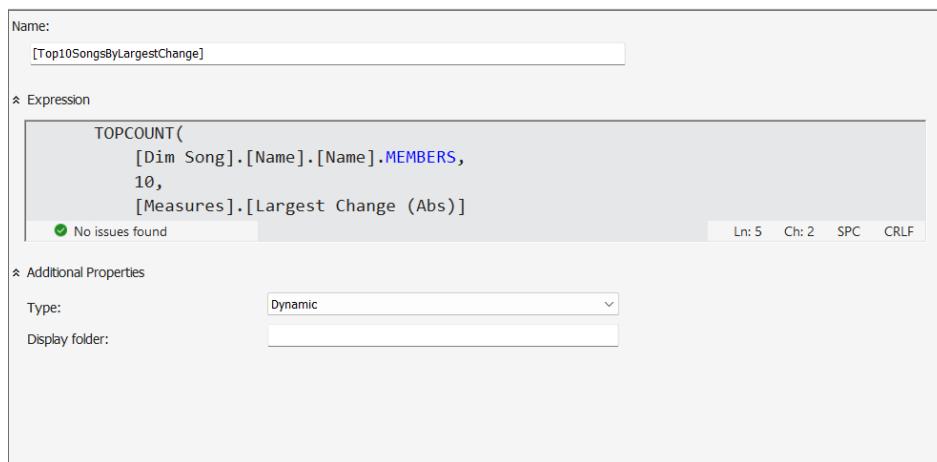
Name	Measure Group	Data Type	Aggregation
Daily Rank	Fact Song Snapshot	Integer	Sum
Daily Movement	Fact Song Snapshot	Integer	Sum
Weekly Movement	Fact Song Snapshot	Integer	Sum
Popularity	Fact Song Snapshot	Double	Sum
Fact Song Snapshot...	Fact Song Snapshot	Integer	Count
Minimum Daily Move...	Fact Song Snapshot	Integer	Min
Maximum Daily Mov...	Fact Song Snapshot	Integer	Max
Song Artist Count	Song Artist	Integer	Count

Bước 2: Vào mục Calculation, tạo mới calculated member:

Báo cáo đồ án



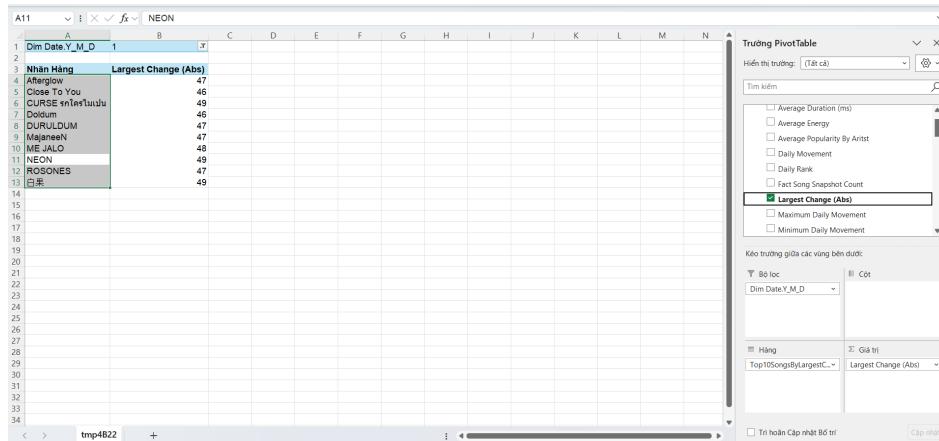
Bước 3: Tạo mới Named Set để lấy top 10 bài hát có sự thay đổi thứ hạng lớn nhất.



Bước 4: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Name	Largest Change (Abs)
Afterglow	47
Close To You	46
CURSE	49
Dokum	46
DURULUDUM	47
MajaneN	47
ME JALO	48
NEON	49
ROSONES	47
白果	49

3.5.14 Câu truy vấn 5 - Sử dụng Pivot Table trong Excel



3.5.15 Câu truy vấn 5 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

WITH

```
MEMBER [Measures]. [Largest Change (Abs)] AS
    IIF(
        ABS([Measures]. [Maximum Daily Movement]) >= ABS([Measures]. [Minimum Daily Movement])
        ABS([Measures]. [Maximum Daily Movement]),
        ABS([Measures]. [Minimum Daily Movement])
    ),
    FORMAT_STRING = "#,##0"
```

SELECT

```
{ [Measures]. [Largest Change (Abs)] } ON COLUMNS,
TOPCOUNT(
    [Dim Song]. [Name]. [Name]. MEMBERS,
    10,
    [Measures]. [Largest Change (Abs)]
) ON ROWS
FROM [TRENDING SONGS SSIS]
WHERE
    ( [Dim Date]. [Y_M_D]. [Month]. &[1]&[2025] );
```

The screenshot shows the SSAS Browser interface. At the top, there is an MDX query:

```
[Measures].[Largest Change (Abs)]
) ON ROWS
FROM [TRENDING SONGS SSIS]
WHERE
( [Dim Date].[Y_M_D].[Month].&[1]&[2025] );
```

The status bar at the top indicates "83 % No issues found". Below the query, there are two tabs: "Messages" and "Results". The "Results" tab is selected and displays a table titled "Largest Change (Abs)" with the following data:

Song	Largest Change (Abs)
CURSE ใจไม่เป็น	49
NEON	49
白果	49
ME JALO	48
Afterglow	47
DURULDUM	47
MajaneeN	47
ROSONES	47
Close To You	46
Doldum	46

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.16 Câu truy vấn 6 - Sử dụng SSAS

Nội dung câu truy vấn: Phân tích số lượng bài hát và điểm phổ biến trung bình (popularity) của các bài hát xuất hiện trên bảng xếp hạng, nhóm theo năm phát hành album để quan sát xu hướng qua từng năm.

Ví dụ: Phân tích số lượng bài hát và điểm phổ biến trung bình của các bài hát xuất hiện trong bảng xếp hạng, nhóm theo năm phát hành album từ năm 2000 đến năm 2025.

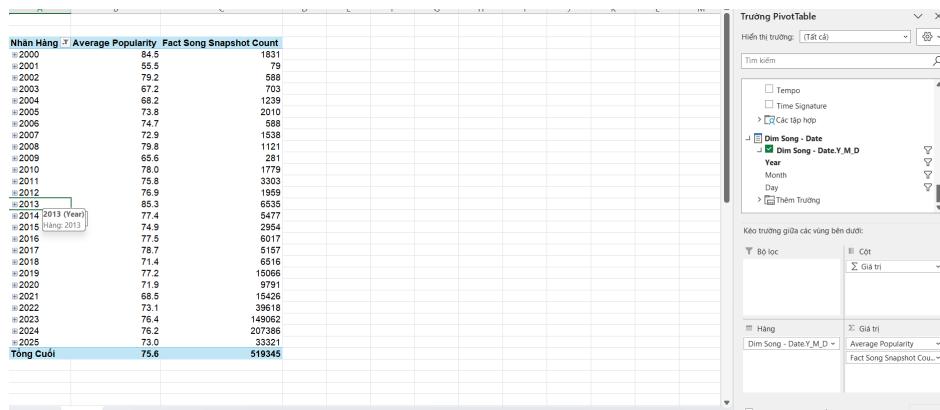
Ghi chú: Sử dụng lại calculation member Average Popularity đã tạo ở câu truy vấn 4.

Bước 1: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

The screenshot shows the SSAS Browser interface with the cube structure open. The "Dimensions" pane on the left lists various dimensions like Dim Song, Dim Date, etc. The "Measures" pane shows a single measure "Fact Song Snapshot Count". The main area displays a data grid with columns: Year, Fact Song Snapshot Count, and Average Popularity. The "Filter Expression" for the Year dimension is set to "2000 : 2025". The data grid shows the following data:

Year	Fact Song Snapshot Count	Average Popularity
2000	1831	84.5363189513927
2001	79	55.493670860759
2002	588	79.193877510204
2003	703	67.1635946372689
2004	1239	68.224374495609
2005	2010	73.8014925371314
2006	588	74.7159863945578
2007	1538	72.8940182054616
2008	1121	79.8064228367529
2009	281	65.622758007118
2010	1779	78.0342889263631
2011	3303	75.7674659400545
2012	1959	76.885145482389
2013	6535	85.3427697010607
2014	5477	77.4422128902684
2015	2954	74.9211238997969
2016	6017	77.5329898620575
2017	5157	78.7335660267597
2018	6516	71.3502148557397
2019	15066	77.1638789326961
2020	9791	71.8673271371668
2021	15426	68.4611046285492
2022	39618	73.0550002524105
2023	149062	76.4253330828783
2024	207386	76.1876838359388

3.5.17 Câu truy vấn 6 - Sử dụng Pivot Table trong Excel



3.5.18 Câu truy vấn 6 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

SELECT
{
    [Measures].[Fact Song Snapshot Count],
    [Measures].[Average Popularity]
} ON COLUMNS,
NON EMPTY
(
    [Dim Song - Date].[Y_M_D].[Year].&[2000]
    :
    [Dim Song - Date].[Y_M_D].[Year].&[2025]
)
ON ROWS
FROM
[TRENDING SONGS SSIS];

```

The screenshot shows the SSAS Browser interface. At the top, there is an MDX query window containing the following code:

```

SELECT
{
    [Measures].[Fact Song Snapshot Count],
    [Measures].[Average Popularity]
} ON COLUMNS,
NON EMPTY
(
    [Dim Song - Date].[Y_M_D].[Year].&[2000]
    :
    [Dim Song - Date].[Y_M_D].[Year].&[2025]
) ON ROWS
FROM
[TRENDING SONGS SSIS];

```

Below the query window is a results grid titled "Messages Results". The grid has two columns: "Fact Song Snapshot Count" and "Average Popularity". The data is as follows:

	Fact Song Snapshot Count	Average Popularity
2010	1779	78.0
2011	3303	75.8
2012	1959	76.9
2013	6535	85.3
2014	5477	77.4
2015	2954	74.9
2016	6017	77.5
2017	5157	78.7
2018	6516	71.4
2019	15066	77.2
2020	9791	71.9
2021	15426	68.5
2022	39618	73.1
2023	149062	76.4
2024	207386	76.2
2025	33321	73.0

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.19 Câu truy vấn 7 - Sử dụng SSAS

Nội dung câu truy vấn: Với từng quốc gia, thống kê tổng điểm **popularity** theo giá trị **is_explicit**.

Bước 1: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

The screenshot shows the SSAS Browser MDX editor. The query is as follows:

```

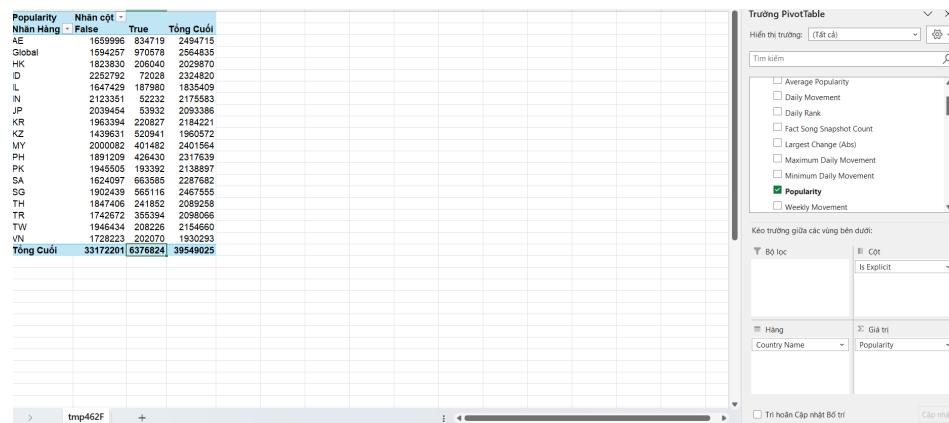
SELECT
Is Explicit AS Is Explicit,
Country Name AS Country Name,
Popularity AS Popularity
FROM
[TRENDING SONGS SSIS]
WHERE
    Is Explicit = TRUE;

```

The left pane shows the cube structure with dimensions like Dim Date, Dim Popularity Group, Dim Song, etc., and measures like Acousticness, Danceability, Duration Ms, Energy, Instrumentalness, Key, Loudness, Mode, Name, and Song Id.

Ghi chú: Nếu như chỉ dùng kéo thả thông thường, ta sẽ không thể hiện thị được giá trị **is_explicit** như một dạng bảng xoay được.

3.5.20 Câu truy vấn 7 - Sử dụng Pivot Table trong Excel



3.5.21 Câu truy vấn 7 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

SELECT

NON EMPTY
[Dim Song].[Is Explicit].[Is Explicit].MEMBERS ON COLUMNS,

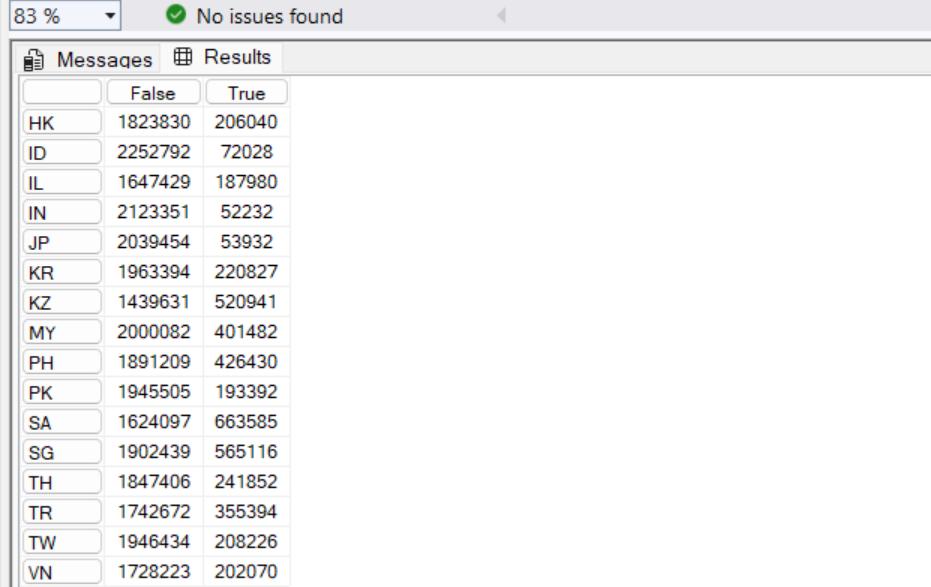
NON EMPTY
[Dim Country].[Country Name].[Country Name].MEMBERS ON ROWS

FROM

[TRENDING SONGS SSIS]

WHERE

[Measures].[Popularity] ;



The screenshot shows a table with two columns: "False" and "True". The rows represent different countries with their corresponding values. The data is as follows:

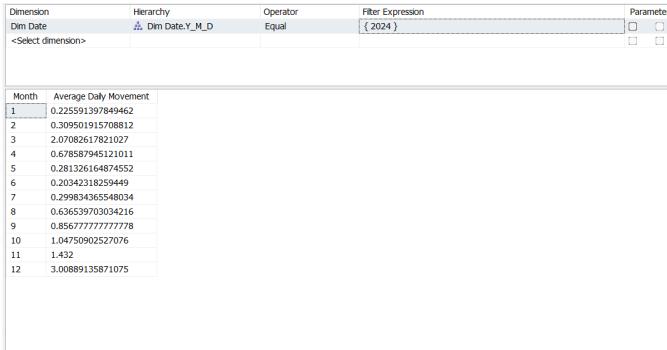
	False	True
HK	1823830	206040
ID	2252792	72028
IL	1647429	187980
IN	2123351	52232
JP	2039454	53932
KR	1963394	220827
KZ	1439631	520941
MY	2000082	401482
PH	1891209	426430
PK	1945505	193392
SA	1624097	663585
SG	1902439	565116
TH	1847406	241852
TR	1742672	355394
TW	1946434	208226
VN	1728223	202070

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau. Tuy nhiên trong SSAS Browser, không thể hiện rõ bảng xoay như trong Pivot Table và MDX.

3.5.22 Câu truy vấn 8 - Sử dụng SSAS

Nội dung câu truy vấn: (Drill-down theo hierarchy Date) Phân tích sự thay đổi thứ hạng trung bình (daily_movement) của các bài hát từ năm → tháng trong năm 2024, để xem chi tiết hơn về mức độ biến động theo thời gian.

Bước 1: Sang Calculation, tạo mới calculated member:



The screenshot shows the SSAS Calculation view. A new calculated member is being defined with the following details:

- Dimension:** Dim Date
- Hierarchy:** Dim Date.Y_M_D
- Operator:** Equal
- Filter Expression:** {2024}
- Parameters:** None

The resulting calculated member is named "Average Daily Movement" and contains the following data:

Month	Average Daily Movement
1	0.225591397849462
2	0.309501915708812
3	2.07082617821027
4	0.678857945121011
5	0.281326164874552
6	0.20342318259449
7	0.2998343655468034
8	0.636539703034216
9	0.856777777777778
10	1.04750902527076
11	1.432
12	3.00889135871075

Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

The screenshot shows a Microsoft Power BI interface. At the top, there is a filter pane for the 'Dim Date' dimension, set to 'Equal' with the value '{ 2024 }'. Below the filter pane is a PivotTable with the following data:

Month	Average Daily Movement
1	0.225591397849462
2	0.309501915708812
3	2.07082617821027
4	0.678587945121011
5	0.281326164874552
6	0.20342318259449
7	0.299834365548034
8	0.636539703034216
9	0.856777777777778
10	1.04750902527076
11	1.432
12	3.00889135871075

3.5.23 Câu truy vấn 8 - Sử dụng Pivot Table trong Excel



3.5.24 Câu truy vấn 8 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
MEMBER [Measures].[Average Daily Movement] AS
DIVIDE(
    [Measures].[Daily Movement],
    [Measures].[Fact Song Snapshot Count]
),
FORMAT_STRING = "0.00"

```

SELECT

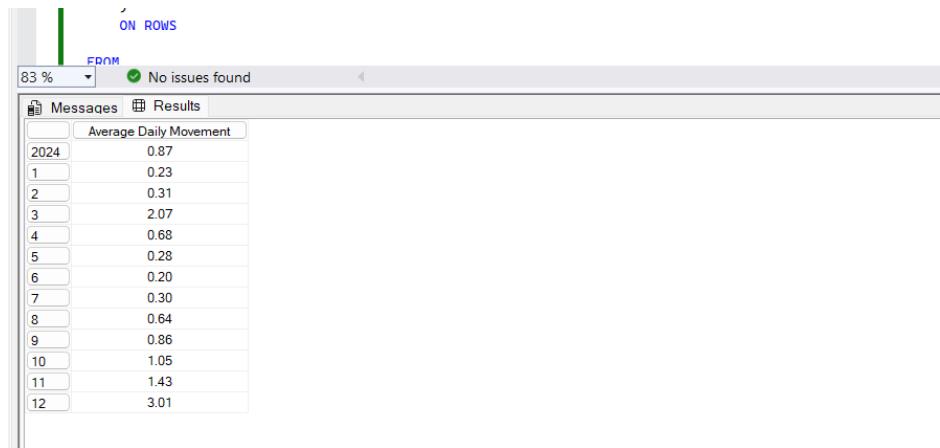
```
{ [Measures].[Average Daily Movement] } ON COLUMNS,
```

NON EMPTY

DRILLDOWNMEMBER(

```
{ [Dim Date].[Y_M_D].[Year].&[2024] },
{ [Dim Date].[Y_M_D].[Year].&[2024] }
)
ON ROWS

FROM
[TRENDING SONGS SSIS];
```

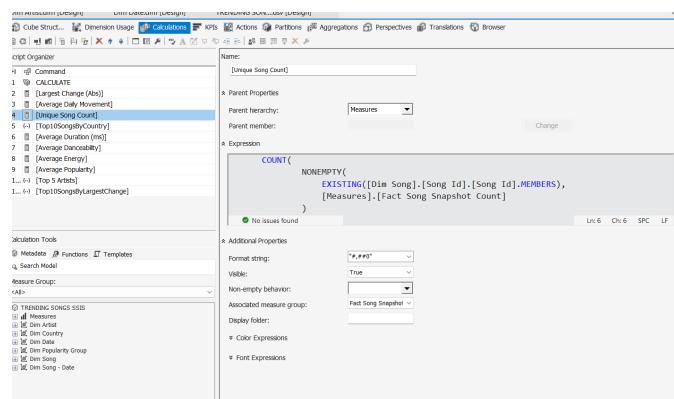


Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.25 Câu truy vấn 9 - Sử dụng SSAS

Nội dung câu truy vấn: So sánh số lượng bài hát xuất hiện trên bảng xếp hạng theo từng quốc gia và từng tháng trong năm 2024.

Bước 1: Tạo mới Calculation Member:

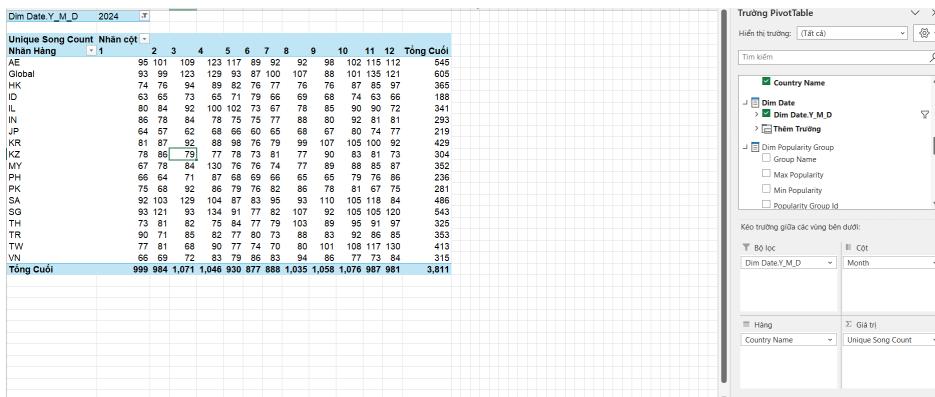


Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Báo cáo đồ án

Dimension	Hierarchy	Operator	Filter Expression	Parameters
Dim Date	Dim Date.Y_M_D	Equal	{ 2024 }	
<Select dimension>				
Country Name	Month	Unique Song Count		
AE	1	95		
AE	2	101		
AE	3	109		
AE	4	123		
AE	5	117		
AE	6	89		
AE	7	92		
AE	8	92		
AE	9	98		
AE	10	102		
AE	11	115		
AE	12	112		
Global	1	93		
Global	2	99		
Global	3	123		
Global	4	129		
Global	5	93		
Global	6	87		
Global	7	100		
Global	8	107		
Global	9	88		
Global	10	101		
Global	11	135		
Global	12	121		
HK	1	74		

3.5.26 Câu truy vấn 9 - Sử dụng Pivot Table trong Excel



3.5.27 Câu truy vấn 9 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
MEMBER [Measures].[Unique Song Count] AS
COUNT(
NONEMPTY(
EXISTING([Dim Song].[Song Id].[Song Id].MEMBERS),
[Measures].[Fact Song Snapshot Count]
)
),
FORMAT_STRING = "#,##0"

SET [Months in 2024] AS
[Dim Date].[Y_M_D].[Year].&[2024].CHILDREN
  
```

```
SELECT
```

```
NON EMPTY [Months in 2024]
ON 0,
```

```
NON EMPTY
    [Dim Country]. [Country Name]. [Country Name].MEMBERS
ON 1
```

```
FROM
```

```
[TRENDING SONGS SSIS]
```

```
WHERE
```

```
( [Measures]. [Unique Song Count] )
```

```
;
```

The screenshot shows the SSAS Browser interface with a title bar 'NON EMPTY' and a status bar indicating '83 %' and 'No issues found'. Below the title bar is a toolbar with icons for Refresh, Save, and Help. The main area is divided into 'Messages' and 'Results' tabs, with the 'Results' tab selected. The results grid has 13 columns labeled 1 through 12, representing months. Rows represent countries: IL, IN, JP, KR, KZ, MY, PH, PK, SA, SG, TH, TR, TW, and VN. Each cell contains a numerical value representing the unique song count for that country in that month.

	1	2	3	4	5	6	7	8	9	10	11	12
IL	80	84	92	100	102	73	67	78	85	90	90	72
IN	86	78	84	78	75	75	77	88	80	92	81	81
JP	64	57	62	68	66	60	65	68	67	80	74	77
KR	81	87	92	88	98	76	79	99	107	105	100	92
KZ	78	86	79	77	78	73	81	77	90	83	81	73
MY	67	78	84	130	76	76	74	77	89	88	85	87
PH	66	64	71	87	68	69	66	65	65	79	76	86
PK	75	68	92	86	79	76	82	86	78	81	67	75
SA	92	103	129	104	87	83	95	93	110	105	118	84
SG	93	121	93	134	91	77	82	107	92	105	105	120
TH	73	81	82	75	84	77	79	103	89	95	91	97
TR	90	71	85	82	77	80	73	88	83	92	86	85
TW	77	81	68	90	77	74	70	80	101	108	117	130
VN	66	69	72	83	79	86	83	94	86	77	73	84

3.5.28 Câu truy vấn 10 - Sử dụng SSAS

Nội dung câu truy vấn: Phân tích số lượng bài hát phân biệt (unique songs) xuất hiện trong bảng xếp hạng của từng quốc gia trong **3 tháng đầu tiên của năm 2025**. Mục tiêu là xác định quy mô danh sách bài hát của từng quốc gia, qua đó thấy được mức độ đa dạng âm nhạc của các thị trường trong giai đoạn đầu năm.

Ghi chú: Sử dụng lại calculation member Unique Song Count đã tạo ở câu truy vấn 9.

Bước 1: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Country Name	Unique Song Count
AE	211
Global	214
HK	151
ID	87
IL	131
IN	149
JP	99
KR	147
KZ	152
MY	171
PH	124
PK	150
SA	197
SG	190
TH	176
TR	140
TW	170
VN	137

3.5.29 Câu truy vấn 10 - Sử dụng Pivot Table trong Excel

Nhân Hàng	Unique Song Count
AE	211
Global	214
HK	151
ID	87
IL	131
IN	149
JP	99
KR	147
KZ	152
MY	171
PH	124
PK	150
SA	197
SG	190
TH	176
TR	140
TW	170
VN	137
Tổng Cuối	1,723

3.5.30 Câu truy vấn 10 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

SELECT
{ [Measures].[Unique Song Count] } ON COLUMNS,
NON EMPTY [Dim Country].[Country Name].[Country Name].MEMBERS ON ROWS
FROM [TRENDING SONGS SSIS]
WHERE
( [Dim Date].[Y_M_D].[Month].&[1]&[2025] :
[Dim Date].[Y_M_D].[Month].&[3]&[2025] );

```

The screenshot shows the SSAS Browser interface. At the top, there is an MDX query window containing the following code:

```

SELECT
    { [Measures].[Unique Song Count] } ON COLUMNS,
    NON EMPTY [Dim Country].[Country Name].[Country Name].MEMBERS ON ROWS
FROM [TRENDING SONGS SSIS]
WHERE
    ( [Dim Date].[Y_M_D].[Month].&[1]&[2025] :
    [Dim Date].[Y_M_D].[Month].&[3]&[2025] );

```

Below the query window is a message bar indicating "No issues found". Underneath the message bar, there are two tabs: "Messages" and "Results". The "Results" tab is selected and displays a table titled "Unique Song Count" with the following data:

	Unique Song Count
AE	211
Global	214
HK	151
ID	87
IL	131
IN	149
JP	99
KR	147
KZ	152
MY	171
PH	124
PK	150
SA	197
SG	190
TH	176
TR	140
TW	170

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.31 Câu truy vấn 11 - Sử dụng SSAS

Nội dung câu truy vấn: Tìm các nhạc sĩ có bài hát nằm trong Xác định tên của các nhạc sĩ có bài hát nằm trong top 10 Global của một ngày nhất định.

Ví dụ: Tìm các nhạc sĩ có bài hát nằm trong top 10 Global vào ngày 28/03/2025.

Bước 1: Tạo mới Named Set để lấy danh sách các bài hát trong top 10.

The screenshot shows the SSAS Script Organizer. On the left, there is a tree view of the script structure under the "CALCULATE" node. A new item, "[Top10Songs]", has been added to the list. The main pane displays the definition of this Named Set:

```

FILTER(
    NONEMPTY(
        [Dim Song].[Name].[Name].MEMBERS,
        (
            [Measures].[Daily Rank],
            [Dim Date].[Full Date].CURRENTMEMBER,
            [Dim Country].[Country Name].CURRENTMEMBER
        )
    ),
    (
        [Measures].[Daily Rank],
        [Dim Date].[Full Date].CURRENTMEMBER,
        [Dim Country].[Country Name].CURRENTMEMBER
    ) <= 10
)

```

At the bottom of the main pane, there is a status bar with "Ln: 15 Ch: 6 SPC CRLF". Below the main pane, there is a section for "Additional Properties" with "Type: Dynamic" and "Display folder:".

Báo cáo đồ án

Bước 2: Sang Browser, chuyển sang Design Mode và thêm đoạn MDX sau vào ô Query.

```

SELECT
    { [Measures].[Daily Rank] } ON COLUMNS,
    NON EMPTY
        (
            [Dim Artist].[Artist Name].[Artist Name].MEMBERS *
            [Top10Songs]
        ) ON ROWS
    FROM
        [TRENDING SONGS SSIS]
    WHERE
        (
            [Dim Date].[Full Date].[2025-03-28],
            [Dim Country].[Country Name].&[Global]
        );
    
```

Artist Name	Name	Daily Rank
Alex Warren	Ordinary	5
Bad Bunny	BAILE INOLVIDABLE	10
Bad Bunny	DTMF	7
Billie Eilish	BIRDS OF A FEATHER	3
Bruno Mars	APT	2
Bruno Mars	Die With A Smile	1
Doechii	Anxiety	4
Gracie Abrams	Thats So True	8
JENNIE	like JENNIE	9
Kendrick Lamar	luther with sza	6
Lady Gaga	Die With A Smile	1
ROSÉ	APT	2
SZA	luther with sza	6

3.5.32 Câu truy vấn 11 - Sử dụng Pivot Table trong Excel

The screenshot shows the 'PivotTable Fields' pane and the main worksheet area. The fields listed in the pane include:

- Rows: Dim Date.Y_M_D, Global, Country Name, Dim Artist, Artist Id, Artist Name, Dim Country.
- Columns: Song Artist, Song Artist Count.
- Values: Daily Rank.

3.5.33 Câu truy vấn 11 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
SET [Top10Songs] AS
FILTER(
    NONEMPTY(
        [Dim Song].[Name].[Name].MEMBERS,
        (
            [Measures].[Daily Rank],
    
```

```

        [Dim Date]. [Full Date]. CURRENTMEMBER,
        [Dim Country]. [Country Name]. CURRENTMEMBER
    )
),
(
    [Measures]. [Daily Rank],
    [Dim Date]. [Full Date]. CURRENTMEMBER,
    [Dim Country]. [Country Name]. CURRENTMEMBER
) <= 10
)

SELECT
{ [Measures]. [Daily Rank] } ON COLUMNS,
NON EMPTY
(
    [Dim Artist]. [Artist Name]. [Artist Name]. MEMBERS *
    [Top10Songs]
) ON ROWS
FROM
[TRENDING SONGS SSIS]
WHERE
(
    [Dim Date]. [Full Date]. [2025-03-28],
    [Dim Country]. [Country Name]. & [Global]
);

```

The screenshot shows the SSAS Browser interface. At the top, there is an MDX query window with the following code:

```
JUN HUWS
FROM
[TRENDING SONGS SSIS]
WHERE
(
    [Dim Date].[Full Date].[2025-03-28],
    [Dim Country].[Country Name].&[Global]
);
```

Below the query window is a status bar indicating "82 %". To the right of the status bar is a message box stating "No issues found". Below these are two tabs: "Messages" and "Results". The "Results" tab is selected, displaying a table with the following data:

		Daily Rank
Alex Warren	Ordinary	5
Bad Bunny	BAILE INOLVIDABLE	10
Bad Bunny	DiMF	7
Billie Eilish	BIRDS OF A FEATHER	3
Bruno Mars	APT	2
Bruno Mars	Die With A Smile	1
Doechii	Anxiety	4
Gracie Abrams	Thats So True	8
JENNIE	like JENNIE	9
Kendrick Lamar	Luther with sza	6
Lady Gaga	Die With A Smile	1
ROSÉ	APT	2
SZA	Luther with sza	6

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau. Tuy nhiên trong SSAS Browser, ta khó khăn trong việc sử dụng Named Set để kéo thả.

3.5.34 Câu truy vấn 12 - Sử dụng SSAS

Nội dung câu truy vấn: Tìm ra tên album có số lượng bài hát nằm trong top 50 nhiều nhất của một quốc gia nhất định trong khoảng thời gian nhất định.

Ví dụ: Tìm ra tên album có số lượng bài hát nằm trong top 50 nhiều nhất của quốc gia VN trong tháng 2 năm 2025.

Bước 1: Sử dụng lại measure Unique Song Count đã tạo ở câu truy vấn 9 và tạo ra Named Set để lấy danh sách top album có số lượng bài hát nằm trong top 50 nhiều nhất.

The screenshot shows the SSAS Studio interface with the "Named Sets" node selected in the left navigation pane. A new Named Set is being created with the following properties:

- Name:** [Top5Album]
- Expression:**

```
TOPCOUNT(
    [Dim Song].[Album Name].[Album Name].MEMBERS,
    5,
    [Measures].[Unique Song Count]
```
- Type:** Dynamic
- Additional Properties:** Display folder: (empty)

At the bottom of the expression editor, there is a message box stating "No issues found".

Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Báo cáo đồ án

The screenshot shows the 'Dimension' configuration for the 'Top5Album' named set. It includes the following details:

Dimension	Hierarchy	Operator	Filter Expression	Parameters
Dim Country	Country Name	Equal	{ VN }	<input type="checkbox"/> <input type="checkbox"/>
Dim Date	Dim Date.Y_M_D	Equal	{ 2 }	<input type="checkbox"/> <input type="checkbox"/>
Dim Song	Album Name	In	Top5Album	<input type="checkbox"/> <input type="checkbox"/>
<Select dimension>				

Below the configuration table is a preview table showing the results of the filter expression:

Album Name	Unique Song Count
AI Càng Phải Bắt Đầu Từ Đầu Đó	3
AMORTAGE	4
Bảo Tàng Của Nuôi Tiếc	3
m-tp M-TP	3
rosie	4

Lưu ý: Cần phải kéo thả các thuộc tính Country Name và Date vào trước khi kéo thả Named Set Top5Album. Nếu không sẽ không thể lọc đúng kết quả theo quốc gia và thời gian được.

3.5.35 Câu truy vấn 12 - Sử dụng Pivot Table trong Excel

The screenshot shows an Excel PivotTable with the following data:

Country Name	VN	
Dim Date.Y_M_D	2	
Nhóm Hàng	Unique Song Count	
AI Càng Phải Bắt Đầu Từ Đầu Đó	3	
AMORTAGE	4	
Bảo Tàng Của Nuôi Tiếc	3	
m-tp M-TP	3	
rosie	4	

To the right of the PivotTable is the 'Trường PivotTable' (PivotTable Fields) pane, which displays the following settings:

- Hiển thị trường: (Tất cả)
- Tìm kiếm:
- Unique Song Count (selected)
- Kéo trưởng giữa các vùng bên dưới:

 - Bộ lọc: Country Name, Dim Date.Y_M_D
 - Hàng: Top5Album
 - Cột: Unique Song Count

3.5.36 Câu truy vấn 12 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
SET [Top5Album1] AS
TOPCOUNT(
    [Dim Song].[Album Name].[Album Name].MEMBERS,
    5,
    [Measures].[Unique Song Count]
)

```

```

SELECT
{ [Measures].[Unique Song Count] } ON COLUMNS,
[Top5Album1] ON ROWS
FROM [TRENDING SONGS SSIS]

```

```

WHERE (
    [Dim Country].[Country Name].&[VN],
    [Dim Date].[Y_M_D].[Month].&[2]&[2025]
);

```

The screenshot shows the SSAS Studio interface. At the top, there is a DAX query:

```
LTOPSONG WITH ROWS
FROM [TRENDING SONGS SSIS]
WHERE (
    [Dim Country].[Country Name].&[VN],
    [Dim Date].[Y_M_D].[Month].&[2]&[2025]
);
```

Below the query is a results grid titled "Results". It displays a table with the following data:

	Unique Song Count
AMORTAGE	4
rosie	4
Ai Cũng Phải Bắt Đầu Từ Đầu Đó	3
Bảo Tàng Của Nuôi Tiếc	3
m-tp M-TP	3

3.5.37 Câu truy vấn 13 - Sử dụng SSAS

Nội dung câu truy vấn: Nhóm theo country và tính weekly_movement trung bình để xem thị trường nào có sự biến động thứ hạng mạnh nhất.

Ví dụ: Tính weekly_movement trung bình của các bài hát xuất hiện trong bảng xếp hạng từ tháng 4 đến tháng 6 năm 2025, nhóm theo quốc gia.

Bước 1: Tạo mới Calculation Member:

The screenshot shows the "Calculation Member" creation dialog in SSAS Studio. The "Name" field is set to "[Average Weekly Movement]". The "Expression" field contains the following DAX code:

```
[Measures].[Weekly Movement],  
[Measures].[Fact Song Snapshot Count]
```

The "Additional Properties" section includes:

- Format string: "0.00"
- Visible: True
- Non-empty behavior: Fact Song Snapshot
- Associated measure group: Fact Song Snapshot
- Display folder: (empty)

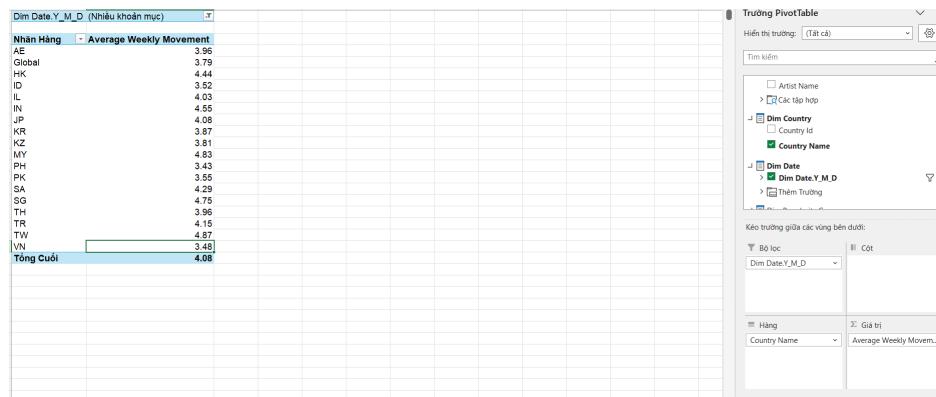
Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Báo cáo đồ án

The screenshot shows a Microsoft Analysis Services Dimension browser window. At the top, there are tabs for 'Dimension' (selected), 'Hierarchy' (Dim Date.Y_M_D), 'Operator Range (Inclusive)', 'Filter Expression' (4 : 6), and 'Parameters'. Below the tabs is a table with two columns: 'Country Name' and 'Average Weekly Movement'. The data rows are:

Country Name	Average Weekly Movement
AE	3.9603125
Global	3.78569197125898
HK	4.4415625
ID	3.51625
IL	4.02875
IN	4.5453125
JP	4.0827349359975
KR	3.8665625
KZ	3.80599812558575
MY	4.82698313554029
PH	3.43
PK	3.549375
SA	4.29285045270059
SG	4.7546875
TH	3.961875
TR	4.15432677288347
TW	4.86602123672705
VN	3.4796875

3.5.38 Câu truy vấn 13 - Sử dụng Pivot Table trong Excel



3.5.39 Câu truy vấn 13 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

```

WITH
MEMBER [Measures].[Average Weekly Movement] AS
    DIVIDE(
        [Measures].[Weekly Movement],
        [Measures].[Fact Song Snapshot Count]
    ),
    FORMAT_STRING = "0.00"

SELECT
{ [Measures].[Average Weekly Movement] } ON COLUMNS,
NON EMPTY
[Dim Country].[Country Name].[Country Name].MEMBERS
ON ROWS
  
```

```

FROM
[TRENDING SONGS SSIS]

WHERE
(
    [Dim Date].[Y_M_D].[Month].&[4]&[2025] : [Dim Date].[Y_M_D].[Month].&[6]&[2025]
);

```

Country	Average Weekly Movement
HK	4.44
ID	3.52
IL	4.03
IN	4.55
JP	4.08
KR	3.87
KZ	3.81
MY	4.83
PH	3.43
PK	3.55
SA	4.29
SG	4.75
TH	3.96
TR	4.15
TW	4.87
VN	3.48

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.40 Câu truy vấn 14 - Sử dụng SSAS

Nội dung câu truy vấn: Phân nhóm các bài hát dựa trên điểm phổ biến (popularity) thành các khoảng (ví dụ: 0-20, 21-40,...) và tính thời lượng trung bình (duration_ms) cho mỗi nhóm.

Ví dụ: Phân nhóm các bài hát xuất hiện trong bảng xếp hạng vào tháng 5 năm 2025 dựa trên điểm phổ biến thành các khoảng (0-40, 41-70, 71-100) và tính thời lượng trung bình cho mỗi nhóm.

Bước 1: Sử dụng lại calculation member Average Duration (ms) đã tạo ở câu truy vấn 2.

Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Group Name	Average Duration (ms)
High	214100.142857143
Low	200424.539215686
Medium	217968.326714801

3.5.41 Câu truy vấn 14 - Sử dụng Pivot Table trong Excel



3.5.42 Câu truy vấn 14 - Sử dụng ngôn ngữ MDX

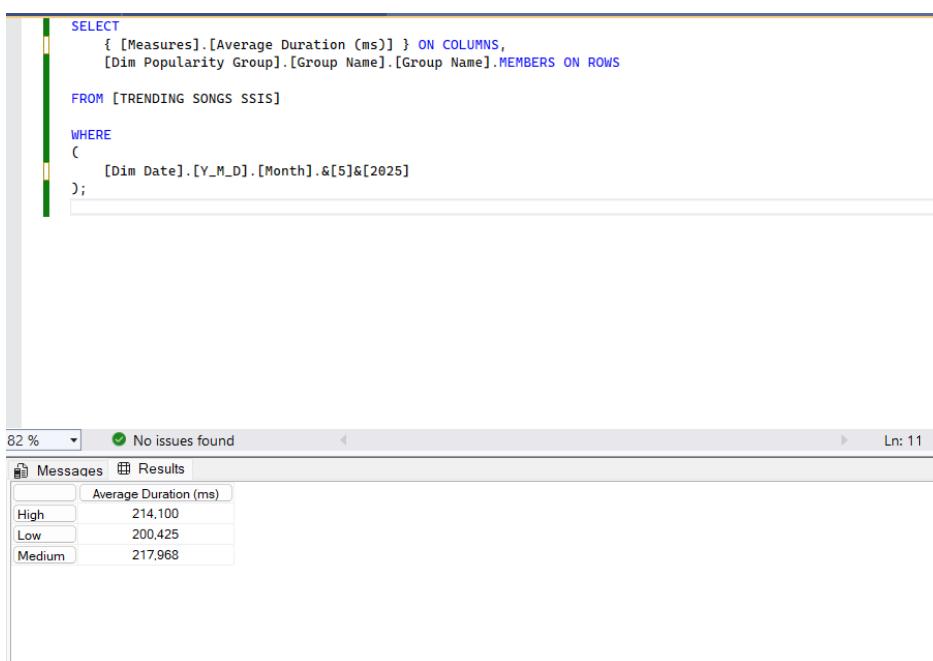
Câu truy vấn MDX:

```

SELECT
{ [Measures].[Average Duration (ms)] } ON COLUMNS,
[Dim Popularity Group].[Group Name].[Group Name].MEMBERS ON ROWS

FROM [TRENDING SONGS SSIS]

WHERE
(
    [Dim Date].[Y_M_D].[Month].&[5]&[2025]
);
  
```



Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

3.5.43 Câu truy vấn 15 - Sử dụng SSAS

Nội dung câu truy vấn: Đếm số lượng bài hát xuất hiện trên bảng xếp hạng trong một giai đoạn cụ thể, nhóm theo `time_signature` để xác định chữ nhịp được sử dụng phổ biến nhất.

Ví dụ: Đếm số lượng bài hát xuất hiện trong bảng xếp hạng từ tháng 6 đến tháng 8 năm 2024, nhóm theo `time_signature`.

Bước 1: Sử dụng lại measure Unique Song Count đã tạo ở câu truy vấn 9.

Bước 2: Sang Browser, kéo thả các Dimension và Measure cần thiết vào Rows và Columns.

Time Signature	Unique Song Count
1	5
3	78
4	1335
5	13
Unknown	0

3.5.44 Câu truy vấn 15 - Sử dụng Pivot Table trong Excel

3.5.45 Câu truy vấn 15 - Sử dụng ngôn ngữ MDX

Câu truy vấn MDX:

SELECT

```
{ [Measures].[Unique Song Count] } ON COLUMNS,
```

NON EMPTY

```
[Dim Song].[Time Signature].[Time Signature].MEMBERS ON ROWS
```

```
FROM
[TRENDING SONGS SSIS]

WHERE
(
    [Dim Date].[Y_M_D].[Month].&[6]&[2024] :
    [Dim Date].[Y_M_D].[Month].&[8]&[2024]
);
```

The screenshot shows the SSAS Browser interface. At the top, there is a code editor window containing an MDX query:

```
SELECT
{ [Measures].[Unique Song Count] } ON COLUMNS,
NON EMPTY
[Dim Song].[Time Signature].[Time Signature].MEMBERS ON ROWS
FROM
[TRENDING SONGS SSIS]
WHERE
(
    [Dim Date].[Y_M_D].[Month].&[6]&[2024] :
    [Dim Date].[Y_M_D].[Month].&[8]&[2024]
);
```

Below the code editor is a results pane. The status bar indicates "82 %", "No issues found", and "Ln:". The results pane has tabs for "Messages" and "Results". The "Results" tab is selected and displays a table:

	Unique Song Count
1	5
3	78
4	1,335
5	13
Unknown	0

Nhận xét: Kết quả trả về giữa SSAS Browser, Pivot Table và MDX đều giống nhau.

4 QUÁ TRÌNH LẬP BÁO BIỂU (SSRS)

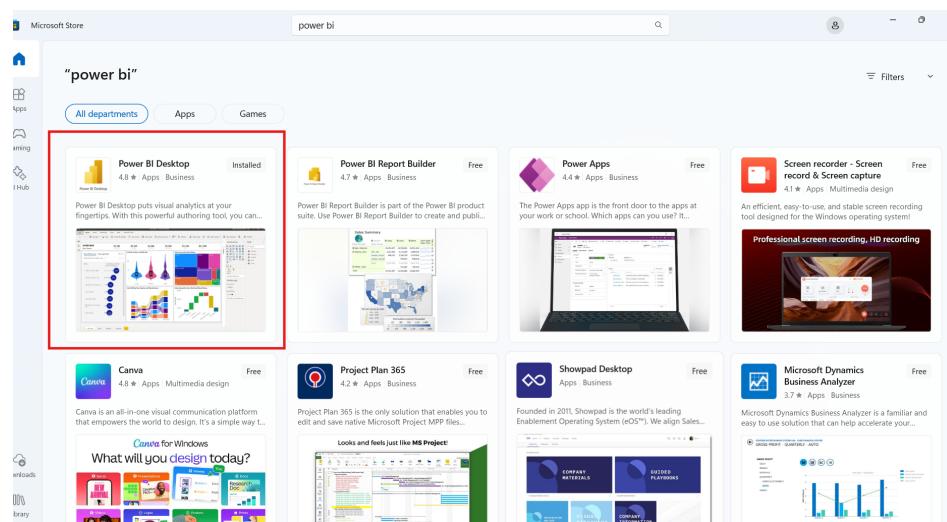
4.1 Chuẩn bị công cụ

Để thực hiện quá trình lập báo cáo (SSRS), chúng ta cần chuẩn bị các công cụ sau:

- Power BI.
- Google Data Studio (Locker).

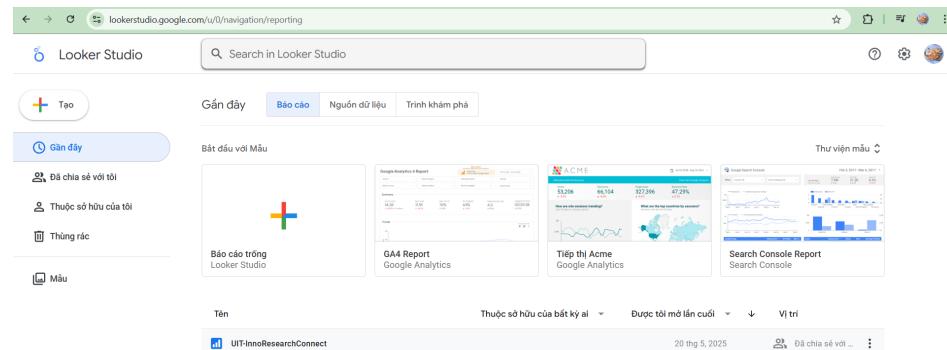
4.1.1 Power BI

Vào Microsoft Store, tìm kiếm Power BI Desktop và tải về máy tính.



4.1.2 Google Data Studio (Locker)

Truy cập vào trang web <https://datastudio.google.com/> và đăng nhập bằng tài khoản Google của bạn.



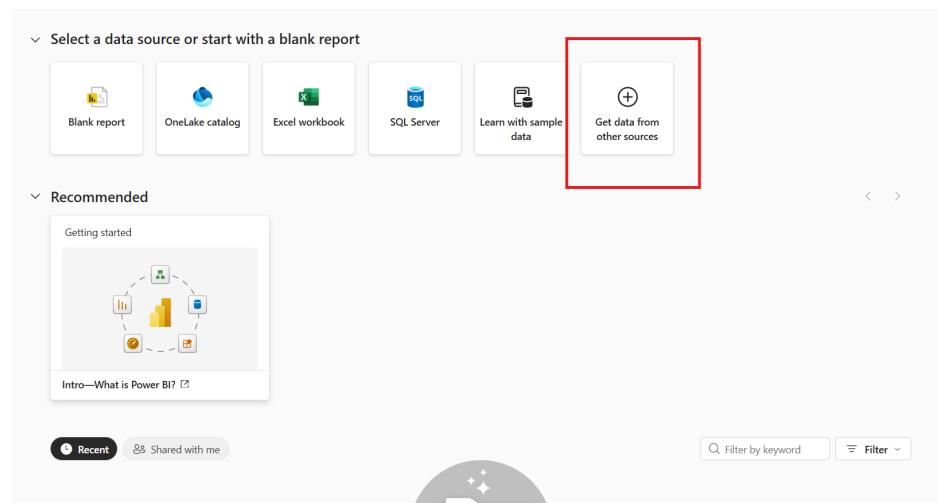
4.2 Quá trình lập báo biểu bằng công cụ Power BI

4.2.1 Báo biểu 1

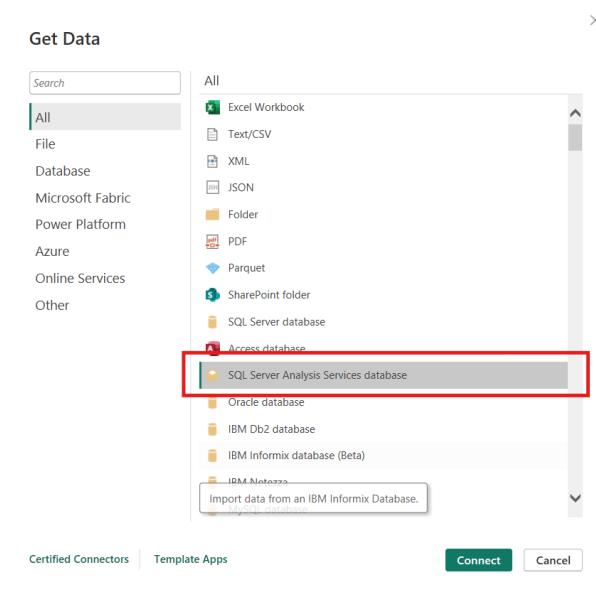
Nội dung (Câu truy vấn 10): Phân tích số lượng bài hát phân biệt (unique songs) xuất hiện trong bảng xếp hạng của từng quốc gia trong **3 tháng đầu tiên của năm 2025**.

Mục tiêu: Xác định quy mô danh sách bài hát của từng quốc gia, qua đó thấy được mức độ đa dạng âm nhạc của các thị trường trong giai đoạn đầu năm.

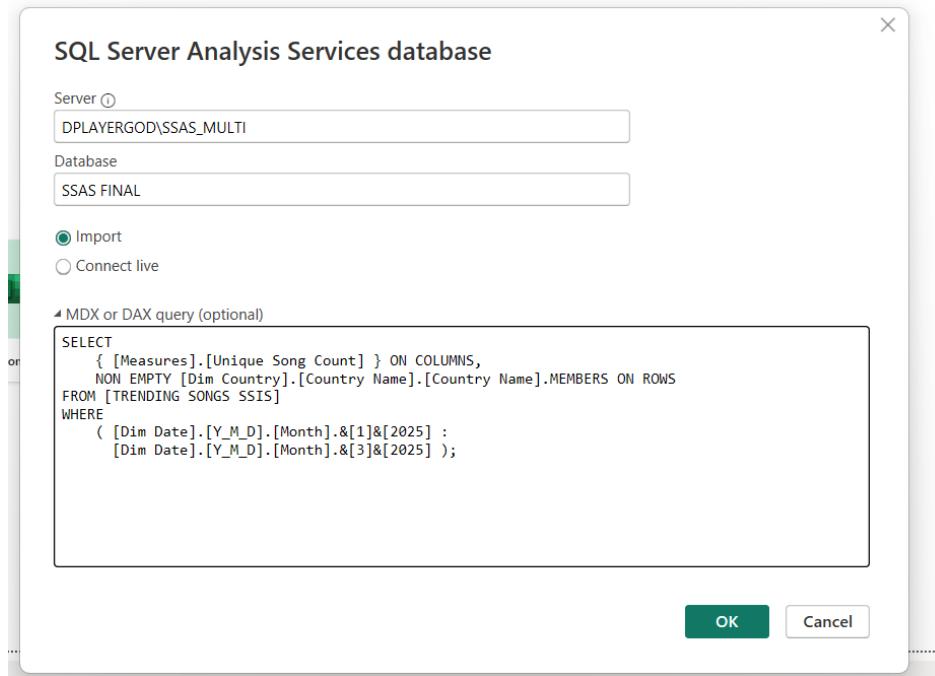
Bước 1: Tại màn hình làm việc của **Power BI**, chọn **Get data from another sources**



Bước 2: Chọn nguồn lấy data là **SQL Server Analysis Services database**. Nhấn **Connect** để kết nối.



Bước 3: Nhập thông tin Server và Database đã tạo trong quá trình xây dựng kho dữ liệu. Sau đó nhập câu truy vấn **MDX**. Cuối cùng, Nhấn **OK** để tiếp tục.



Bước 4: Cửa sổ review dữ liệu hiện ra. Nhấn **Transform Data** để biến đổi dữ liệu.

The screenshot shows a table titled "[Dim Country].[Country Name].[Country Name].[MEMBERS].[Measures].[Unique Song Count]" with the following data:

[Dim Country].[Country Name].[Country Name].[MEMBERS]	[Measures].[Unique Song Count]
AE	211
Global	214
HK	151
ID	87
IL	131
IN	149
JP	99
KR	147
KZ	152
MY	171
PH	124
PK	150
SA	197
SG	190
TH	176
TR	140
TW	170
VN	137

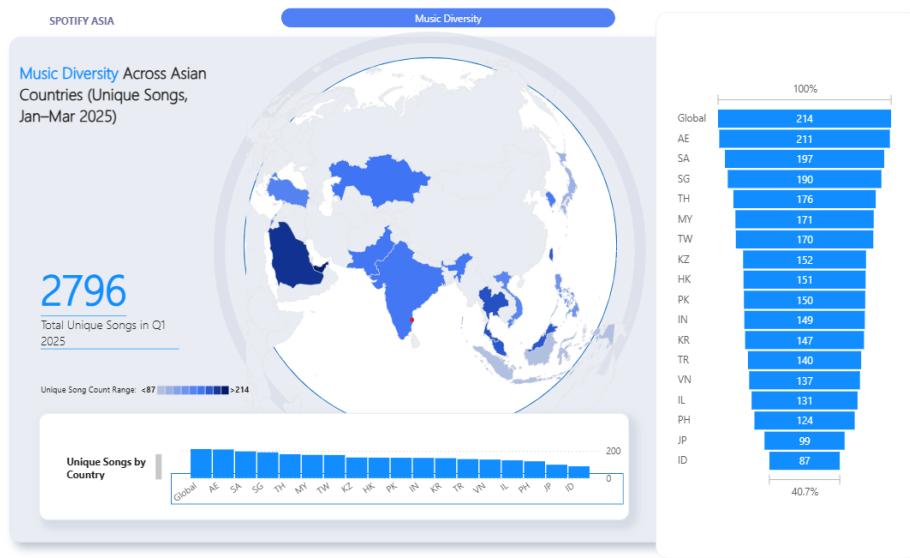
At the bottom right are "Load", "Transform Data" (which is highlighted with a red box), and "Cancel" buttons.

Bước 5: Tại cửa sổ Power Query Editor, đổi tên cột cho phù hợp. Chuyển đổi cột dữ liệu Unique Song Count thành Whole Number. Sau đó, nhấn **Close & Apply** để áp dụng thay đổi và đóng cửa sổ.

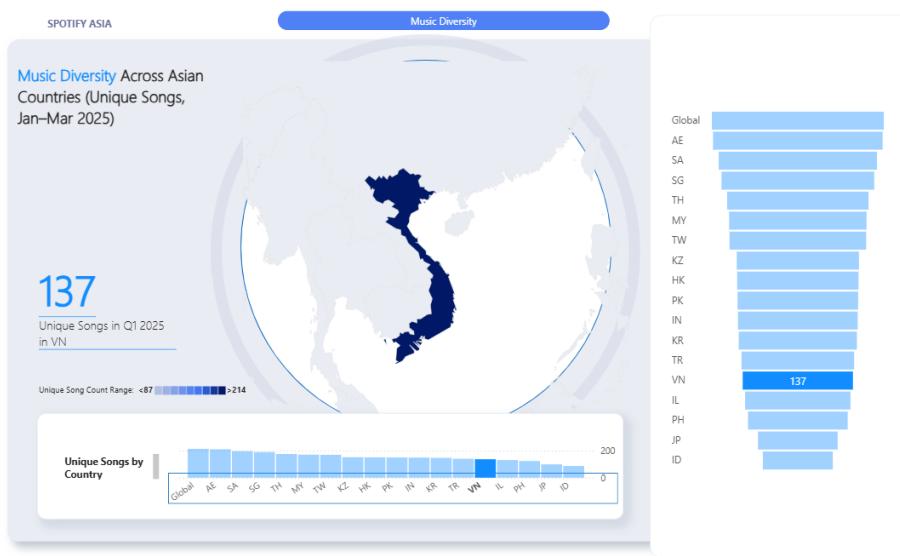
Báo cáo đồ án

Country Name	Unique Song Count
AE	211
Global	214
HK	151
ID	87
IL	131
IN	149
JP	99
KR	147
KZ	152
MY	171
PH	124
PK	150
SA	197
SG	190
TH	176
TR	140
TW	170
VN	137

Bước 6: Thiết kế report. Có thể thêm chart để minh họa trực quan hơn.



Hình 30: Báo biểu 1 - Hiển thị toàn cảnh tất cả các quốc gia



Hình 31: Báo cáo 1 - Hiển thị chi tiết quốc gia Việt Nam

Kết luận:

- Báo cáo tập trung vào việc đánh giá “**độ đa dạng âm nhạc**” của các quốc gia Châu Á trong Quý 1 năm 2025, được đo lường thông qua số lượng bài hát độc nhất xuất hiện trong các bảng xếp hạng. Trong toàn khu vực, có tổng cộng 2.796 bài hát được ghi nhận, phản ánh mức độ thay đổi và luân chuyển bài hát ở mức cao. Chỉ số Global đạt 214 bài hát độc nhất và được sử dụng như mốc so sánh để đánh giá mức độ cởi mở của từng thị trường.
- Dựa trên biểu đồ thanh so sánh số lượng bài hát, các quốc gia có thể được chia thành ba nhóm hành vi rõ rệt. Nhóm thứ nhất là các thị trường có mức độ đa dạng cao, bao gồm UAE (211), Saudi Arabia (197) và Singapore (190). Ba thị trường này có số lượng bài hát độc nhất gần tương đương hoặc chỉ thấp hơn không đáng kể so với Global. Điều này cho thấy bảng xếp hạng của họ thay đổi rất nhanh, người dùng liên tục cận các bài hát mới và **không bị phụ thuộc quá nhiều** vào một nhóm bài hát cố định.
- Nhóm thứ hai là các thị trường có mức độ đa dạng trung bình, gồm Thái Lan (176), Malaysia (171), Đài Loan (170), Hàn Quốc (147) và Việt Nam (137). Các quốc gia này thể hiện mức độ thay đổi vừa phải, trong đó xuất hiện xen kẽ nhiều bài hát mới nhưng vẫn duy trì một số bài hát ổn định trong thời gian dài. Việt Nam với 137 bài hát nằm ở mức trung bình thấp của nhóm này, cho thấy mức độ biến động của bảng xếp hạng không quá nhanh và thói quen nghe nhạc của người dùng mang tính tập trung rõ rệt.
- Nhóm thứ ba gồm các quốc gia có mức độ đa dạng thấp, bao gồm Nhật Bản (99) và Indonesia (87). Đây là hai thị trường có số lượng bài hát độc nhất thấp nhất trong khu vực. Số liệu cho thấy bảng xếp hạng của các nước này ít thay đổi, các bài hát khi đạt thứ hạng cao thường duy trì trong thời gian dài. Chỉ số của Indonesia chỉ bằng khoảng 40% so với Global, phản ánh mức độ biến động rất thấp.
- Từ góc độ chiến lược, dữ liệu cho thấy các thị trường như UAE, Saudi Arabia và Singapore phù hợp với việc cập nhật danh sách phát liên tục, do người dùng có nhu cầu khám phá cao.

Trong khi đó, những thị trường như Nhật Bản, Indonesia lại **ưu tiên tính ổn định**, phù hợp với chiến lược tập trung vào các bài hát nổi bật có vòng đời dài. Thái Lan và Đài Loan là các thị trường có độ mở tốt và mức độ đa dạng cao, tạo điều kiện thuận lợi hơn cho việc đưa các bài hát mới tiếp cận người nghe. Việt Nam với mức độ đa dạng trung bình cho thấy cần có thời gian để người dùng tiếp nhận các bài hát mới trước khi chúng có thể lọt vào bảng xếp hạng.

4.2.2 Báo biểu 2

Nội dung: Báo cáo độ phổ biến của top 5 nghệ sĩ trong bảng xếp hạng vào ngày 1/1/2025.

Mục tiêu: Xác định mức độ ảnh hưởng và phổ biến của các nghệ sĩ hàng đầu trong thị trường âm nhạc.

Bước 1: Xây dựng câu truy vấn MDX để lấy dữ liệu từ kho dữ liệu.

```

SELECT
{
    [Measures].[Average Popularity],
    [Measures].[Fact Song Snapshot Count],
    [Measures].[Average Weekly Movement]
} ON COLUMNS,
NON EMPTY
TOPCOUNT(
    [Dim Artist].[Artist Name].[Artist Name].MEMBERS,
    10,
    [Measures].[Average Popularity]
) ON ROWS

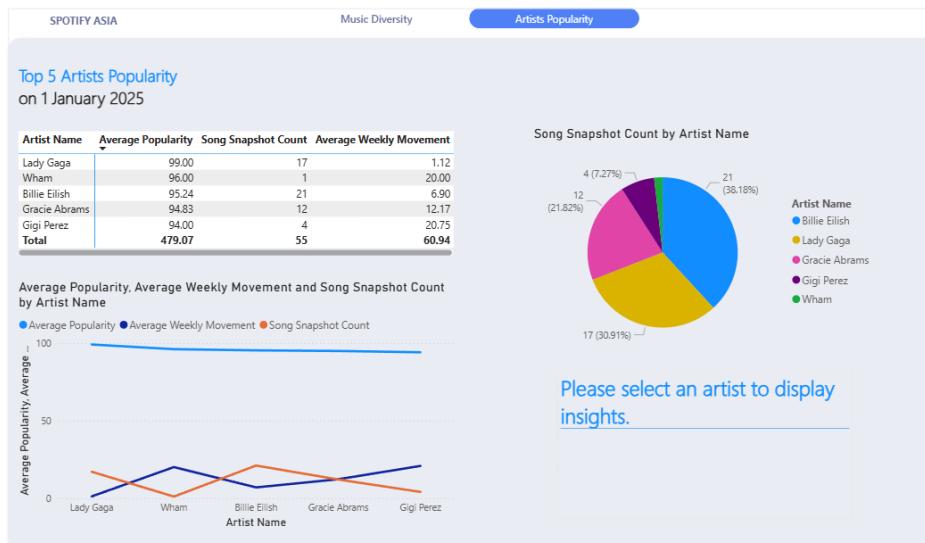
FROM
[TRENDING SONGS SSIS]

WHERE
( [Dim Date].[Y_M_D].[Day].&[1]&[1]&[2025] );

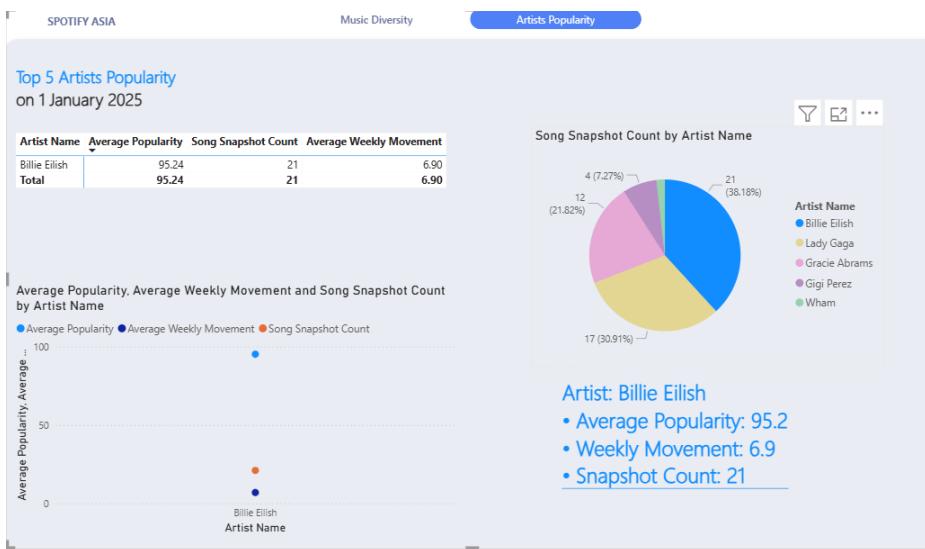
```

Bước 2: Thực hiện các bước tương tự như trong Báo biểu 1 để kết nối và lấy dữ liệu từ kho dữ liệu vào Power BI.

Bước 3: Thiết kế report. Có thể thêm chart để minh họa trực quan hơn.



Hình 32: Báo biểu 2 - Hiển thị độ phổ biến của top 5 nghệ sĩ



Hình 33: Báo biểu 2 - Hiển thị chi tiết nghệ sĩ

Kết luận:

1. Bối Cảnh Dữ Liệu

Dữ liệu được ghi nhận vào ngày 01/01/2025, một thời điểm đặc biệt khi các bản hit Giáng sinh, nhạc lễ hội và các bài pop thịnh hành cùng xuất hiện trên bảng xếp hạng. Ba chỉ số quan trọng được sử dụng để phân tích là:

- **Average Popularity:** phản ánh mức độ nổi tiếng trung bình của nghệ sĩ.
- **Song Snapshot Count:** số lượng bài hát của nghệ sĩ xuất hiện trong bảng xếp hạng, cho thấy độ phủ và mức độ quan tâm của người nghe.

- **Average Weekly Movement:** mức độ biến động thứ hạng qua tuần, thể hiện xu hướng tăng/giảm và mức ổn định.

Dựa trên ba yếu tố này, ta tiến hành phân tích chi tiết từng nghệ sĩ trong Top 5.

2. Phân Tích Chi Tiết Từng Nghệ Sĩ

A. Lady Gaga

- **Chỉ số:** Popularity cao nhất (99.00), Movement thấp nhất (1.12), và 17 bài hát xuất hiện.
- **Nhận định:** Lady Gaga thể hiện sự thống trị tuyệt đối trên bảng xếp hạng với mức độ ổn định rất cao. Chỉ số Movement cực thấp cho thấy thứ hạng của cô gần như không thay đổi, duy trì ở nhóm đầu trong thời gian dài.

B. Wham!

- **Chỉ số:** Popularity rất cao (96.00), chỉ có 1 bài hát, Movement cao (20.00).
- **Nhận định:** Đây là trường hợp điển hình của hiện tượng mùa vụ. Bài hát duy nhất xuất hiện, có thể là "Last Christmas", với mức tăng hạng nhanh vào cuối tháng 12.

C. Billie Eilish

- **Chỉ số:** Song Snapshot Count cao nhất (21), Popularity cao (95.24).
- **Nhận định:** Billie Eilish là nghệ sĩ có mức độ bao phủ lớn nhất. Việc có 21 bài hát lọt vào dữ liệu cho thấy người nghe đang phát nhạc toàn album hoặc playlist, thay vì chỉ nghe một vài bài đơn lẻ.
- **Kết luận:** Billie đang sở hữu thị phần người nghe mạnh nhất với mức độ gắn kết (engagement) cao.

D. Gigi Perez

- **Chỉ số:** Popularity 94.00 (thấp nhất nhóm nhưng vẫn cao), Movement cao nhất (20.75), Song Count chỉ 4.
- **Nhận định:** Gigi Perez đang có tốc độ tăng hạng nhanh nhất, dấu hiệu mạnh của một bài hát viral (có thể từ TikTok hoặc xu hướng mạng xã hội).
- **Giải thích:** Số lượng bài hát thấp cho thấy sự nổi tiếng tập trung vào một vài hit chủ lực, chưa phải mức độ ổn định hay bao phủ rộng như Lady Gaga hoặc Billie Eilish.

E. Gracie Abrams

- **Chỉ số:** Popularity 94.83, Song Count 12, Movement trung bình (12.17).
- **Nhận định:** Gracie Abrams có vị thế ổn định với độ bao phủ tốt và mức độ biến động vừa phải. Điều này cho thấy cô đang duy trì sự phát triển đều trong cộng đồng người nghe.

4.2.3 Báo cáo đồ án

Nội dung: (Drill-down theo hierarchy Date) Phân tích sự thay đổi thứ hạng trung bình (daily_movement) của các bài hát trong bảng xếp hạng theo từng tháng trong năm 2024. (Câu truy vấn 8).

Mục tiêu: Hiểu rõ xu hướng biến động của bảng xếp hạng âm nhạc qua từng tháng, từ đó nhận diện các giai đoạn có sự thay đổi mạnh mẽ hoặc ổn định.

Bước 1: Xây dựng câu truy vấn MDX để lấy dữ liệu từ kho dữ liệu.

```
WITH
MEMBER [Measures]. [Average Daily Movement] AS
    DIVIDE(
        [Measures]. [Daily Movement],
        [Measures]. [Fact Song Snapshot Count]
    ),
    FORMAT_STRING = "0.00"

SELECT
    { [Measures]. [Average Daily Movement] } ON COLUMNS,
    NON EMPTY
    DRILLDOWNMEMBER(
        { [Dim Date]. [Y_M_D]. [Year] .&[2024] },
        { [Dim Date]. [Y_M_D]. [Year] .&[2024] }
    )
    ON ROWS

FROM
    [TRENDING SONGS SSIS];
```

Bước 2: Thực hiện các bước tương tự như trong Báo cáo 1 để kết nối và lấy dữ liệu từ kho dữ liệu vào Power BI.

Bước 3: Thiết kế report. Có thể thêm chart để minh họa trực quan hơn.



Hình 34: Báo biểu 3 - Hiển thị sự thay đổi thứ hạng trung bình theo tháng



Hình 35: Báo biểu 3 - Hiển thị chi tiết tháng 01



Hình 36: Báo biểu 3 - Hiển thị chi tiết tháng 1, 2, 3

Kết luận. Dựa trên biểu đồ Average Daily Movement theo từng tháng trong năm 2024 và bảng giá trị tương ứng, có thể rút ra một số nhận định quan trọng như sau:

- **Hai giai đoạn biến động mạnh nhất là tháng 3 và tháng 12:** Tháng 3 đạt mức 2.07 và tháng 12 đạt mức 3.00, cho thấy đây là những thời điểm mà thứ hạng bài hát thay đổi nhiều nhất trong năm. Mặc dù không trùng với các mùa sự kiện lớn, mức biến động cao có thể phản ánh sự xuất hiện của nhiều bài hát mới, chiến dịch truyền thông, hoặc hiện tượng viral trong các giai đoạn này.
- **Giai đoạn giảm dần sau tháng 3:** Từ tháng 4 đến tháng 7, Average Daily Movement giảm liên tục và chạm mức thấp nhất vào tháng 7 (0.20), cho thấy thị trường âm nhạc trong giai đoạn này tương đối ổn định, ít thay đổi về vị trí xếp hạng.
- **Xu hướng tăng trở lại từ tháng 8 đến tháng 11:** Sau giai đoạn ổn định giữa năm, mức dịch chuyển tăng đều, đạt 1.43 vào tháng 11, thể hiện sự hoạt động trở lại của thị trường trước thời điểm cuối năm.
- **Tháng 12 là điểm bùng nổ mạnh nhất:** Đây là tháng có Average Daily Movement cao nhất (3.00), phản ánh sự cạnh tranh mạnh giữa các bài hát mới trong mùa lễ hội cuối năm — vốn là thời điểm ra mắt nhiều sản phẩm âm nhạc.
- **Giá trị trung bình cả năm là 0.92:** Điều này cho thấy phần lớn các tháng biến động ở mức thấp đến trung bình, và chỉ một vài thời điểm có sự thay đổi thứ hạng mạnh.

Tổng quan, dữ liệu cho thấy thị trường âm nhạc năm 2024 có các đợt biến động mạnh theo từng đợt: đỉnh đầu tiên vào tháng 3, sau đó là giai đoạn ổn định giữa năm, và cuối cùng là bùng nổ mạnh vào tháng 12. Đây là các tín hiệu quan trọng để nhận diện các thời điểm cạnh tranh cao và các chu kỳ biến động trong bảng xếp hạng.

4.3 Quá trình lập báo biểu bằng Google Data Studio (Looker Studio)

4.3.1 Báo biểu 1

Nội dung: Thông kê thời lượng trung bình (ms) của các bài hát xuất hiện trên bảng xếp hạng, được nhóm theo album (truy vấn 2).

Mục đích: Báo biểu nhằm thể hiện phân phối độ dài trung bình giữa các album trên bảng xếp hạng. Thông tin này giúp nghệ sĩ và công ty nắm bắt xu hướng nghe nhạc của người dùng, từ đó đưa ra quyết định phù hợp về độ dài sản phẩm âm nhạc.

Quy trình xây dựng báo cáo:

Dữ liệu gốc:

	Tên bài hát	Average Duration (ms)
1	"Aaj Ki Raat (From ""Stree 2""")	228,620
2	"Ayay Nai (From ""Stree 2""")	178,780
3	"Akhiyaan Gulaab (From ""Teri Baaton Mein Aisa Uljha Jiya""")	171,147
5	"ANH TRAI ""SAY HI"" (Live Stage 2)"	245,454
6	"ANH TRAI ""SAY HI"" (Live Stage 3)"	252,428
7	"ANH TRAI ""SAY HI"" (Live Stage 4)"	276,529
8	"ANH TRAI ""SAY HI""; Chung Kết 1"	210,000
9	"ANH TRAI ""SAY HI""; Tập 14"	252,625
10	"Chal Diye Tum Kahan (From ""Kabhi Main Kabhi Tum""")	275,200
11	"Chuttamalle (From ""Devara Part 1""")	222,063
12	"Đi Giữa Trời Rực Rỡ (From ""Đi Giữa Trời Rực Rỡ""")	220,839
13	"Godari Gattu Meedha (From ""Sankranthiki Vasthunam""")	250,083
14	"Jaana Samjhona (From ""Bhool Bhulaiyaa 3""")	212,108
15	"Kaise Hua (From ""Kabir Singh""")	234,722
16	"Khoobsurat (From ""Stree 2""")	244,583
17	"Mere Sohneya (From ""Kabir Singh""")	193,355
18	"Naina (From ""Crew""")	180,000
19	"O Saathi (From ""Baaghi 2""")	251,818
20	"Peelings (From ""Pushpa 2 The Rule""") [TELUGU]"	247,137
21	"Raanjhana (From ""Do Patti""")	240,066

Hình 37: Dữ liệu gốc ở dạng XLSX

Dữ liệu ban đầu chỉ bao gồm hai cột: *Tên album* và *Độ dài trung bình*. Để dễ phân tích hơn, ta tạo thêm hai trường mới:

- **Phân loại độ dài:** chia thành ba nhóm
 - Dưới 3 phút
 - Từ 3–5 phút
 - Trên 5 phút
- **Duration bin:** nhóm độ dài theo từng khoảng 20 giây.

The screenshot shows a data visualization interface with a context menu open over a dimension field named "Average Duration (ms)". The menu options include "Add calculated field", "Add group", and "Add bin". The "Add calculated field" option is highlighted. To the right of the menu, there are buttons for "Add a parameter" and "Fil". Below the menu, the interface displays two dimensions and one metric:

- Dimensions (2):**
 - Average Duration (ms) (Number)
 - Tên bài hát (Text)
- Metrics (1):**
 - Record Count (Number)

Hình 38: Thêm trường mới bằng *Add calculated field*

The dialog has two main sections: "Field Name" and "Field ID".

Field Name: e.g. New Calculated Field — Phân loại độ dài

Field ID: Field Id — calc_or3wvir8xd

Formula:

```

1 CASE
2 WHEN [Average Duration (ms)] < 180000 THEN "Ngắn (<3 phút)"
3 WHEN [Average Duration (ms)] >= 180000 AND [Average Duration (ms)] <= 300000 THEN "Tiêu chuẩn (3-5 phút)"
4 WHEN [Average Duration (ms)] > 300000 THEN "Dài (>5 phút)"
5 ELSE "Khác"
6 END
  
```

Hình 39: Xây dựng trường ‘Phân loại độ dài’

Báo cáo đồ án

Create a new bin by selecting an existing field. [Learn more](#)

New field name* — **Bining Duration(20s)**

123 Average Duration (ms)

Bin field format* — Integer "x to y"

Average Duration (ms)

Min value 67596 Max value 586075

Refresh field info

Bin type

Equal Sized

Custom Sized

Customize the bin sizes and ranges. Bins are automatically created for values in the data that fall outside the specified ranges.

Bin Size* 20000

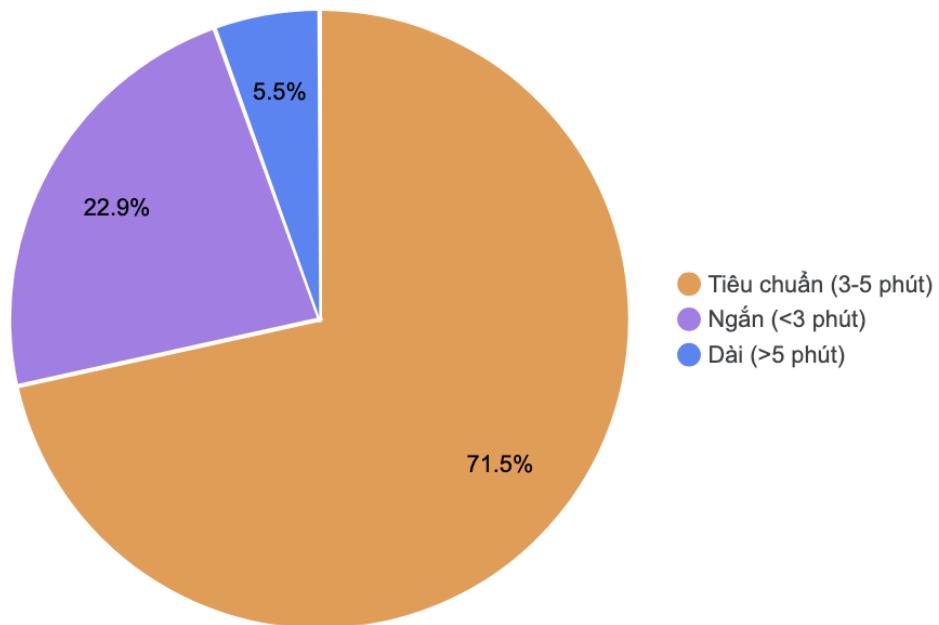
Bin min value* 67596

Bin max value* 586075

Bin remaining values outside the min and max as separate bins

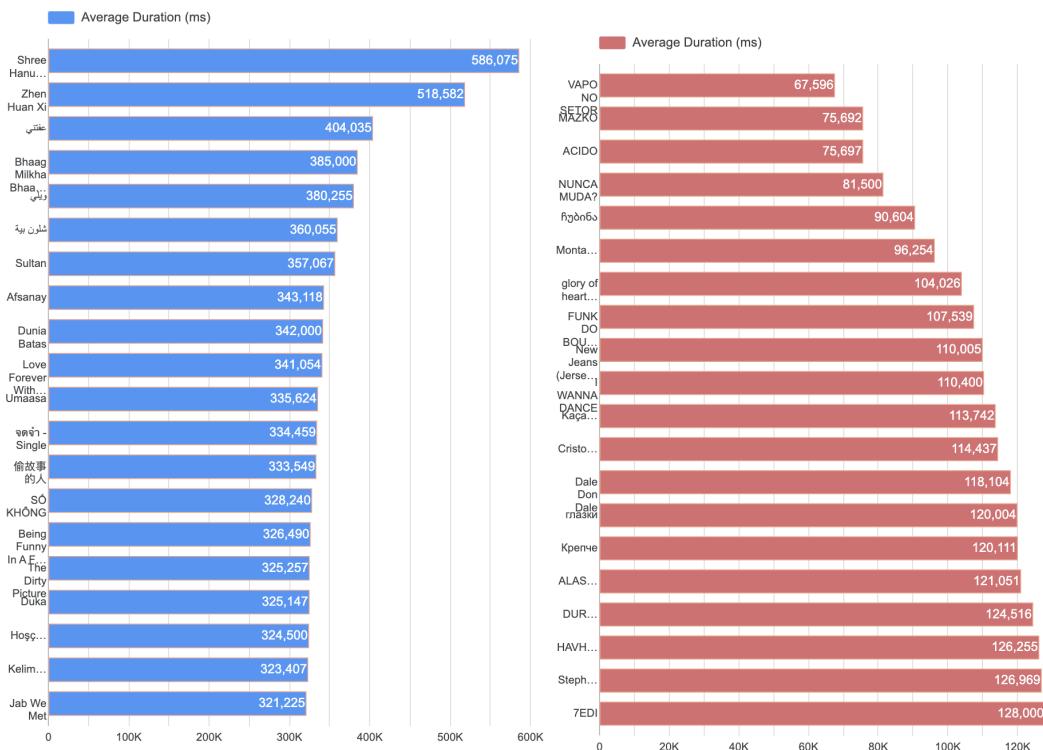
Hình 40: Xây dựng trường ‘Duration bin’

Các biểu đồ:

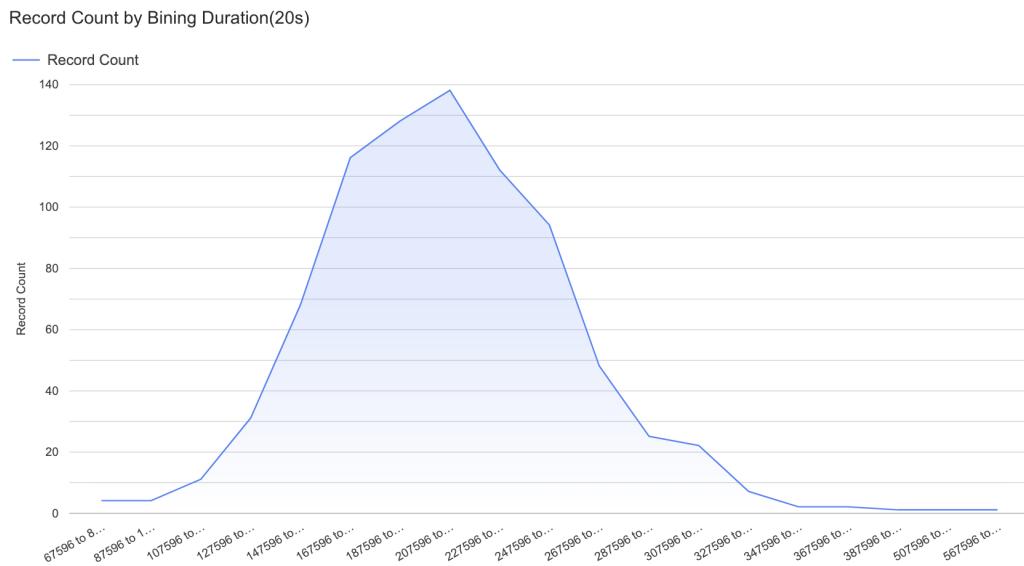


Hình 41: Biểu đồ tròn thể hiện tỷ trọng các loại độ dài

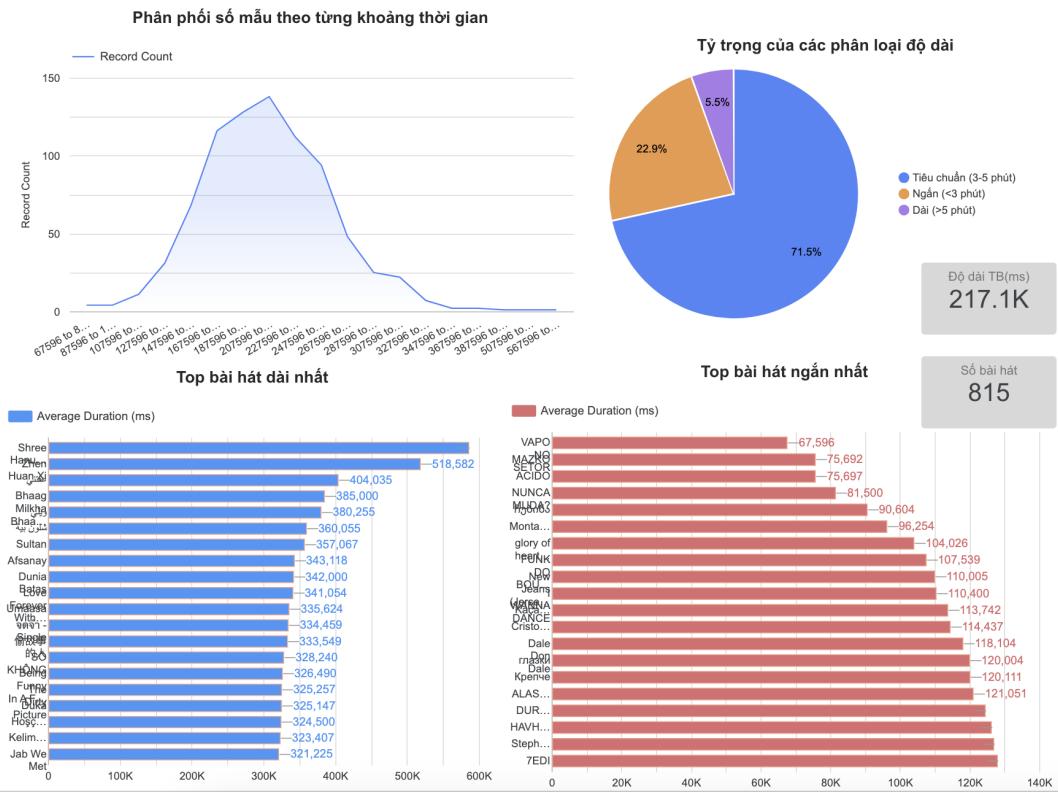
Báo cáo đồ án



Hình 42: Biểu đồ cột ngang thể hiện top 20 album dài nhất và ngắn nhất



Hình 43: Biểu đồ đường theo duration bin



Hình 44: Tổng hợp bieu đồ

Kết luận: Qua quá trình phân tích dữ liệu về thời lượng (duration) của các bài hát, chúng ta có thể rút ra những nhận định quan trọng về cấu trúc sản phẩm âm nhạc trên thị trường hiện nay:

- Sự thống trị của tiêu chuẩn:** Biểu đồ tròn thể hiện rõ ràng rằng nhóm bài hát có thời lượng "Tiêu chuẩn" (3-5 phút) chiếm ưu thế tuyệt đối với tỷ trọng **71.5%**. Đây là độ dài lý tưởng cho việc phát sóng trên radio và duy trì sự tập trung của người nghe. Nhóm bài hát ngắn (<3 phút) chiếm **22.9%**, phản ánh xu hướng nhạc TikTok/ngắn gọn đang lên ngôi, trong khi các bài hát dài (>5 phút) chỉ chiếm thiểu số **5.5%**.
- Phân phối Chuẩn:** Biểu đồ tần suất phân nhóm theo mỗi 20 giây cho thấy phân bố thời lượng bài hát tuân theo quy luật hình chuông. Mật độ bài hát tập trung dày đặc nhất ở khoảng **187.000ms - 227.000ms** (tương đương khoảng 3 phút 07 giây đến 3 phút 47 giây). Số lượng bài hát giảm dần đều về hai phía của đỉnh chuông này.
- Các ngoại lệ:** Biểu đồ thanh ngang giúp xác định rõ các thái cực của dữ liệu.
 - Cực đại:* Bài hát dài nhất ghi nhận là "Shree Hanuman..." với thời lượng lên tới **586.075 ms** (9.7 phút).
 - Cực tiểu:* Bài hát ngắn nhất là "VAPO VAPO..." chỉ kéo dài **67.596 ms** (1.1 phút).

Tuy nhiên, các trường hợp này rất hiếm và nằm xa so với mức trung bình của tập dữ liệu.

- Tổng kết xu hướng:** Dữ liệu khẳng định rằng mặc dù có sự đa dạng về thể loại và nghệ sĩ, nhưng cấu trúc thời gian của một bài hát phổ biến vẫn tuân thủ chặt chẽ quy chuẩn từ 3 đến 4 phút để tối ưu hóa khả năng tiếp cận đại chúng.

4.3.2 Báo biểu 2

Nội dung: Phân tích số lượng bài hát và điểm phổ biến trung bình (*Average Popularity*) của các bài hát xuất hiện trên bảng xếp hạng, được nhóm theo năm phát hành album nhằm quan sát xu hướng thay đổi qua từng năm (Truy vấn 6).

Mục đích: Báo biểu giúp nhận diện những giai đoạn âm nhạc có sự bùng nổ về số lượng bài hát hoặc sự thay đổi trong mức độ phổ biến. Qua đó, các hãng âm nhạc và nghệ sĩ có thể đánh giá xu hướng thưởng thức của người nghe theo từng thời kỳ.

Quy trình xây dựng báo cáo:

Dữ liệu gốc

	A	B	C
1	Năm	Fact Song Snapshot Count	Average Popularity
2	2000	1831	84.5
3	2001	79	55.5
4	2002	588	79.2
5	2003	703	67.2
6	2004	1239	68.2
7	2005	2010	73.8
8	2006	588	74.7
9	2007	1538	72.9
10	2008	1121	79.8
11	2009	281	65.6
12	2010	1779	78.0
13	2011	3303	75.8
14	2012	1959	76.9
15	2013	6535	85.3
16	2014	5477	77.4

Hình 45: Dữ liệu gốc ở dạng XLSX

Dữ liệu gồm ba cột chính:

- Năm
- Fact Song Snapshot Count: số lượng mẫu bài hát xuất hiện trong snapshot
- Average Popularity: điểm phổ biến trung bình

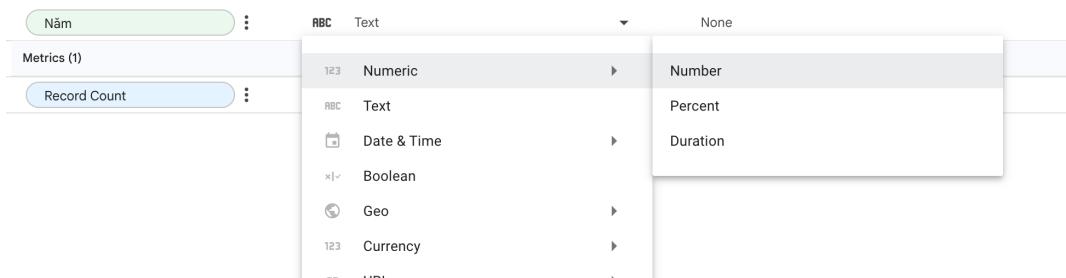
Do cột *Năm* mang kiểu dữ liệu **text**, ta chuyển đổi về dạng **number** để phục vụ phân tích và trực quan hóa. Ngoài ra, ta thêm một trường mới “**Thời đại**” để phân nhóm theo các mốc thời gian:

- 2000–2010
- 2010–2020

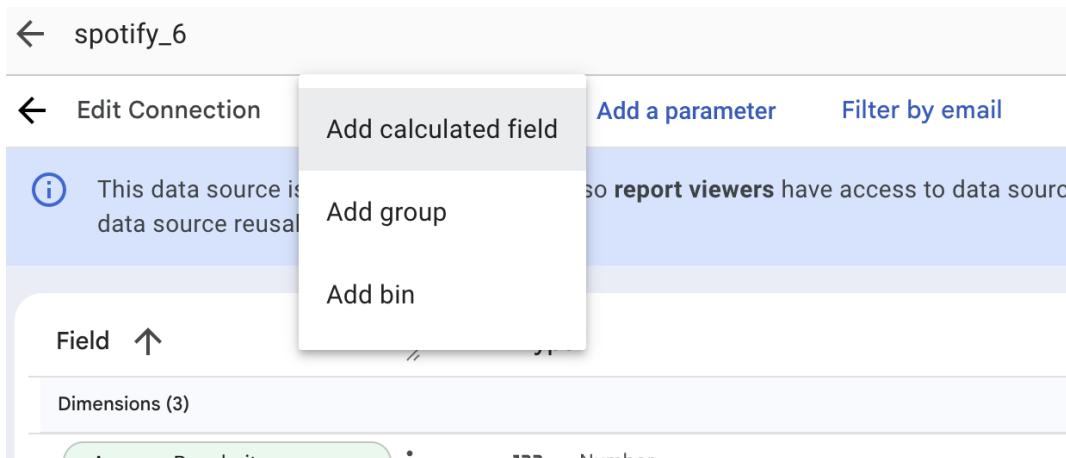
Báo cáo đồ án

- 2020–2023
- 2024–2025

Chỉnh sửa dữ liệu:



Hình 46: Chỉnh sửa dữ liệu bằng biểu tượng bút chì

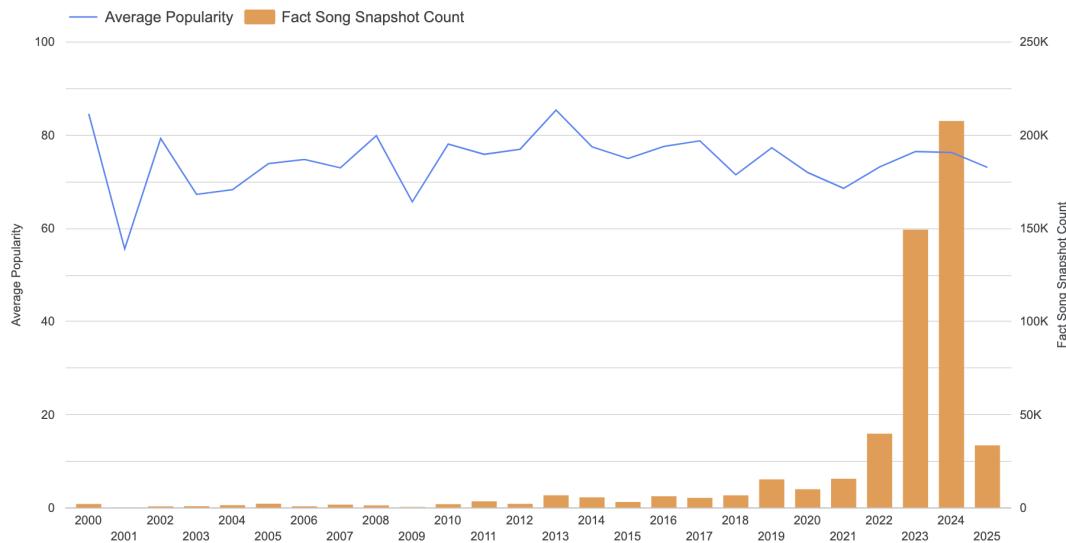


Hình 47: Chuyển kiểu dữ liệu của trường “Năm” sang dạng số

Field Name e.g. New Calculated Field — Thời đại	Field ID Field Id — calc_5dy52tq8xd
Formula	
<pre> 1 CASE 2 WHEN Năm >= 2000 AND Năm <= 2010 THEN "2000 - 2010" 3 WHEN Năm >= 2011 AND Năm <= 2020 THEN "2011 - 2020" 4 WHEN Năm >= 2021 AND Năm <= 2023 THEN "2021 - 2023" 5 WHEN Năm >= 2024 AND Năm <= 2025 THEN "2024 - 2025" 6 ELSE "Giai đoạn khác" 7 END </pre>	

Hình 48: Thêm trường phân loại “Thời đại”

Các biểu đồ:

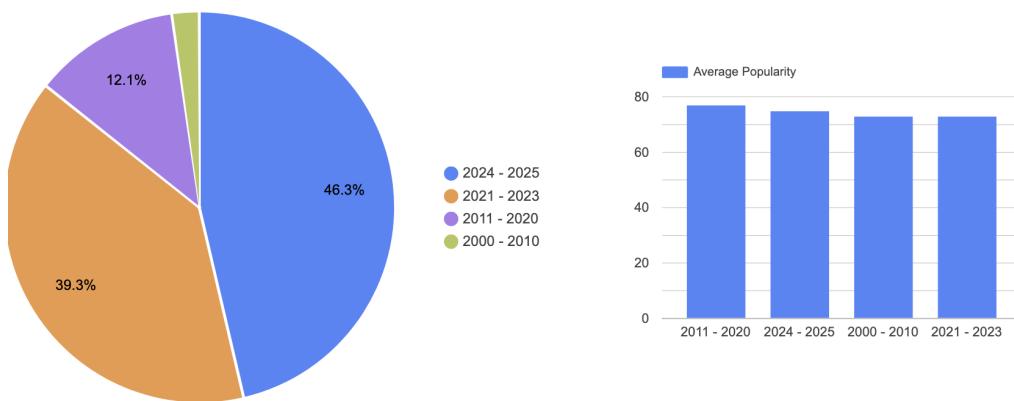


Hình 49: Biểu đồ kết hợp thể hiện Average Popularity và số lượng mẫu theo từng năm (2020–2025)

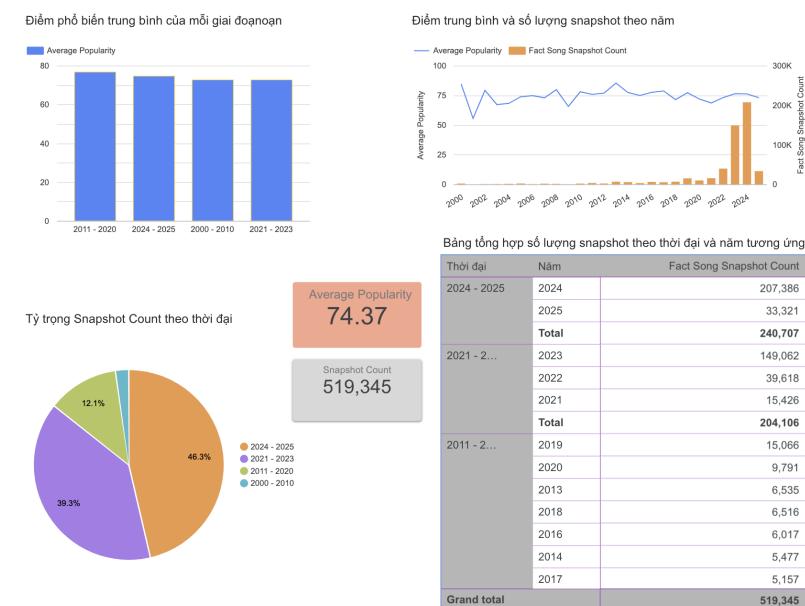
Bảng tổng hợp số lượng snapshot theo thời đại và năm tương ứng

Thời đại	Năm	Fact Song Snapshot Count
2024 - 2025	2024	207,386
	2025	33,321
	Total	240,707
2021 - 2024	2023	149,062
	2022	39,618
	2021	15,426
	Total	204,106
2011 - 2020	2019	15,066
	2020	9,791
	2013	6,535
	2018	6,516
	2016	6,017
	2014	5,477
	2017	5,157
	Grand total	519,345

Hình 50: Biểu đồ tổng hợp số lượng snapshot theo thời đại và năm tương ứng



Hình 51: Biểu đồ tròn (số lượng mẫu) và biểu đồ cột (Average Popularity) theo từng thời đại



Hình 52: Tổng hợp biểu đồ

Biểu đồ tròn cho thấy tỷ trọng số lượng bài hát theo thời kỳ, trong khi biểu đồ cột thể hiện mức độ phổ biến trung bình của từng thời đại.

Kết luận: Dựa trên các biểu đồ phân tích xu hướng bài hát theo năm phát hành (từ dữ liệu Spotify 2024-2025), chúng ta có thể rút ra các nhận định quan trọng sau:

- Khả năng lan truyền của âm nhạc:** Biểu đồ cột và biểu đồ tròn cho thấy sự chênh lệch không lồ về số lượng bài hát. Giai đoạn **2024-2025** chiếm tới **46.3%** và giai đoạn **2021-2023** chiếm **39.3%** tổng lượng bài hát trên bảng xếp hạng. Điều này cho thấy những bài hát mới ra có khả năng tiếp cận công chúng dễ dàng, cho thấy mức độ phát triển của các nền tảng số.
- Chất lượng không phụ thuộc vào số lượng:** Mặc dù số lượng bài hát phát hành những năm 2000-2020 thấp hơn rất nhiều so với hiện tại, nhưng **Điểm phổ biến trung bình** của

chúng lại không hề thua kém, thậm chí cao hơn. Cụ thể, biểu đồ cột theo giai đoạn chỉ ra rằng nhóm **2011-2020** có điểm phổ biến trung bình cao nhất (xấp xỉ 80), cao hơn cả nhóm nhạc mới 2024-2025.

- **Hiệu ứng hit:** Biểu đồ bong bóng và biểu đồ đường (Combo chart) minh chứng rằng các bài hát cũ (từ 2000-2015) còn tồn tại trong dữ liệu đến nay đều là những bài hát hit/classic với chỉ số popularity rất cao (dao động ổn định từ 70-85). Ngược lại, nhạc mới tuy nhiều về lượng nhưng điểm phổ biến trung bình bị kéo xuống do sự cạnh tranh của quá nhiều bài hát mới ra mắt cùng lúc.
- **Xu hướng tiêu thụ:** Người dùng có xu hướng nghe kết hợp: phần lớn thời gian dành cho lượng lớn các ca khúc mới ra mắt (theo trào lưu), nhưng vẫn duy trì sự yêu thích ổn định đối với các bài hát chất lượng cao từ thập kỷ trước.

4.3.3 Báo biểu 3

Nội dung: Với từng quốc gia, thống kê tổng điểm *popularity* của các bài hát trên bảng xếp hạng, được phân loại theo mức độ *explicit* (bạo lực/ngôn từ nhạy cảm).

Mục đích: Báo biểu nhằm đánh giá ảnh hưởng của mức độ explicit đến độ phổ biến của bài hát ở từng thị trường. Điều này giúp quan sát mức độ chấp nhận của người nghe theo từng quốc gia cũng như sự khác biệt văn hoá trong thói quen tiêu thụ âm nhạc.

Quy trình xây dựng báo cáo:

Dữ liệu gốc:

1	Quốc gia	False	True	Tổng Cuối
2	AE	1659996	834719	2494715
3	Global	1594257	970578	2564835
4	HK	1823830	206040	2029870
5	ID	2252792	72028	2324820
6	IL	1647429	187980	1835409
7	IN	2123351	52232	2175583
8	JP	2039454	53932	2093386
9	KR	1963394	220827	2184221
10	KZ	1439631	520941	1960572
11	MY	2000082	401482	2401564

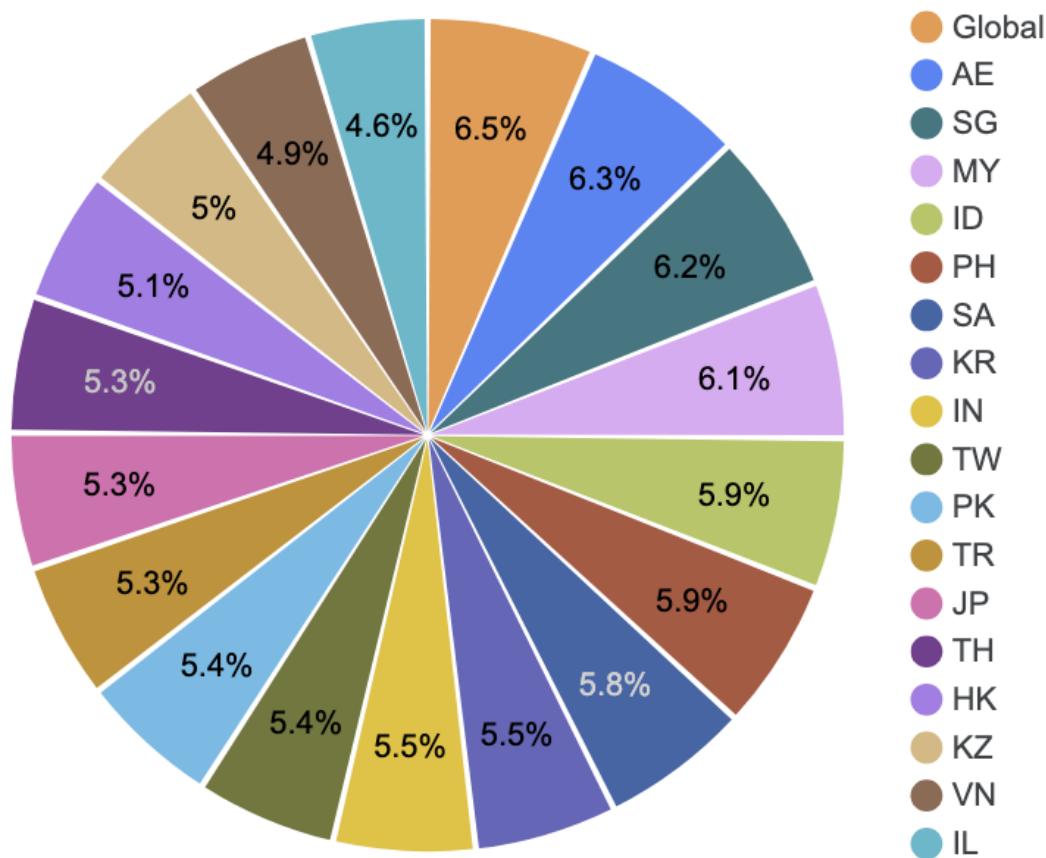
Hình 53: Dữ liệu gốc ở dạng XLSX

Dữ liệu ban đầu bao gồm bốn cột:

- **Country:** tên quốc gia
- **True:** tổng điểm popularity của các bài hát explicit
- **False:** tổng điểm popularity của các bài hát không explicit
- **Total:** tổng điểm popularity của toàn bộ bài hát (True + False)

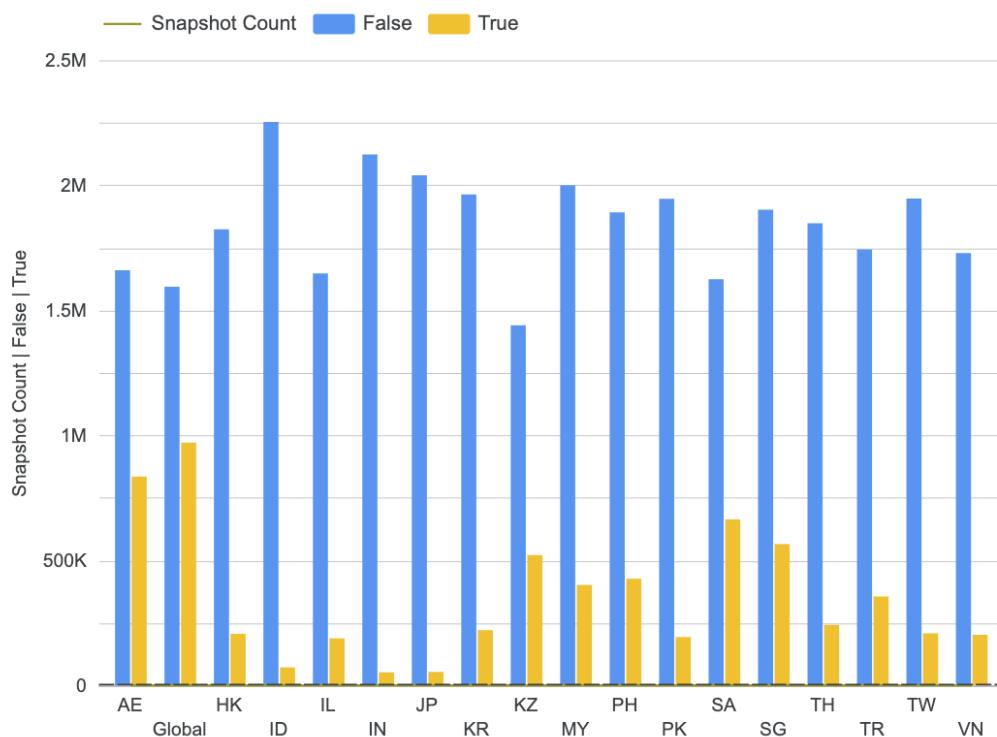
Dữ liệu đã ở dạng phù hợp cho việc trực quan hóa, không cần xử lý thêm.

Các biểu đồ trong báo cáo:

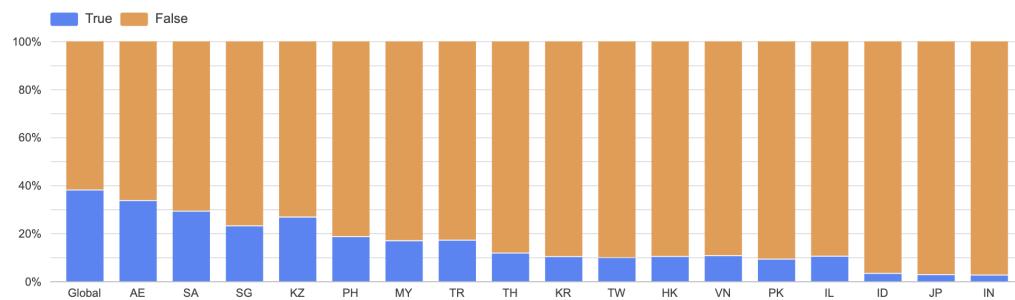


Hình 54: Biểu đồ tròn thể hiện tỷ trọng tổng điểm popularity theo từng quốc gia

Báo cáo đồ án



Hình 55: Biểu đồ cột so sánh tổng điểm popularity của bài hát explicit và non-explicit theo quốc gia



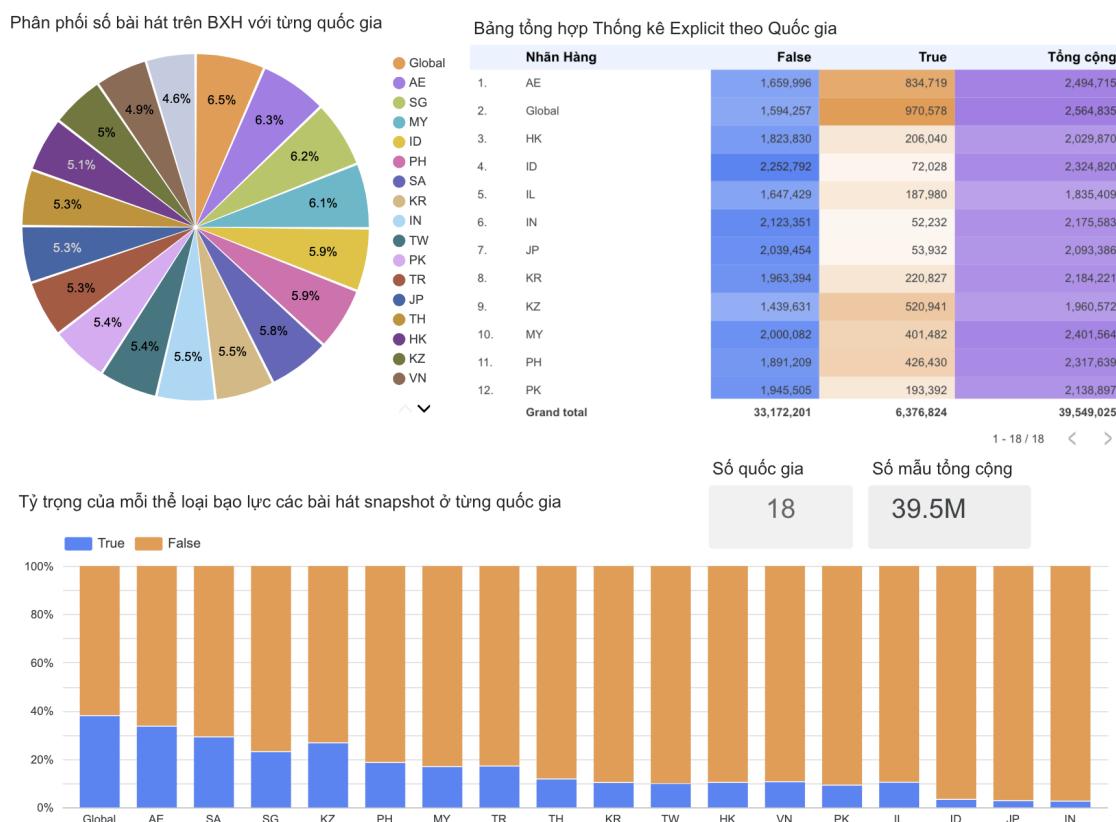
Hình 56: Biểu đồ cột dạng stacked thể hiện tỷ lệ phần trăm explicit và non-explicit theo từng quốc gia

Bảng tổng hợp Thống kê Explicit theo Quốc gia

Nhân Hàng	False	True	Tổng cộng
1. AE	1,659,996	834,719	2,494,715
2. Global	1,594,257	970,578	2,564,835
3. HK	1,823,830	206,040	2,029,870
4. ID	2,252,792	72,028	2,324,820
5. IL	1,647,429	187,980	1,835,409
6. IN	2,123,351	52,232	2,175,583
7. JP	2,039,454	53,932	2,093,386
8. KR	1,963,394	220,827	2,184,221
9. KZ	1,439,631	520,941	1,960,572
10. MY	2,000,082	401,482	2,401,564
11. PH	1,891,209	426,430	2,317,639
12. PK	1,945,505	193,392	2,138,897
Grand total	33,172,201	6,376,824	39,549,025

1 - 18 / 18 < >

Hình 57: Bảng tổng hợp Explicit theo từng quốc gia



Hình 58: Tổng hợp biểu đồ

Kết luận:

Dựa trên các biểu đồ trực quan hóa, có thể rút ra các nhận định sau:

- **Phân phối tổng điểm popularity theo quốc gia .**

Biểu đồ cho thấy tổng điểm popularity phân bổ tương đối đồng đều giữa các quốc gia, không có thị trường nào chiếm tỷ trọng quá vượt trội. Một số quốc gia có tỷ trọng nổi bật hơn mức trung bình. Điều này phản ánh lượng người nghe và mức độ tiêu thụ âm nhạc khác nhau giữa các khu vực, nhưng nhìn chung không có sự chênh lệch quá lớn.

- **So sánh tổng popularity giữa explicit và non-explicit.**

Ở phần lớn các quốc gia, tổng điểm popularity của nhóm non-explicit luôn cao hơn explicit. Một số thị trường có mức độ nghe explicit tương đối cao như AE, Global, KZ, PH, MY. Ngược lại, các quốc gia như JP, IN, ID có điểm explicit rất thấp, cho thấy sự hạn chế trong mức độ tiếp nhận nội dung bạo lực hay nhạy cảm. Xu hướng này phản ánh sự khác biệt về văn hóa, mức độ kiểm duyệt và thói quen nghe nhạc của từng thị trường.

- **Tỷ lệ explicit trong tổng popularity.**

Biểu đồ tỷ lệ cho thấy rõ mức độ chấp nhận explicit giữa các quốc gia. Nhóm chấp nhận explicit cao (30–40%) gồm Global, AE, SA, SG, KZ. Nhóm có tỷ lệ trung bình (15–25%) gồm PH, MY, TR, KR, TW. Các quốc gia có tỷ lệ explicit thấp hơn 10% gồm HK, VN, PK, IL, ID, JP, IN, trong đó JP và IN có mức explicit thấp nhất. Điều này phản ánh sự bảo thủ văn hóa hoặc chính sách kiểm duyệt nghiêm ngặt về nội dung âm nhạc.

- **Tổng kết chung.**

Nội dung explicit chỉ chiếm tỷ lệ nhỏ trong tổng popularity ở hầu hết các quốc gia, ngoại trừ một số thị trường cởi mở hơn như AE, SA, SG. Các quốc gia Đông Á và Nam Á (đặc biệt là JP, IN, ID, VN) có tỷ lệ explicit rất thấp, cho thấy thị hiếu ưu tiên nội dung an toàn, dễ phổ cập hoặc có kiểm duyệt mạnh.

Về mặt ứng dụng, các nghệ sĩ hoặc nhà phân phối có thể ưu tiên phát hành nội dung explicit tại những thị trường cởi mở hơn, trong khi tại các thị trường thận trọng như JP, IN, VN, việc phát hành các bài hát non-explicit sẽ phù hợp hơn và có khả năng đạt mức độ phổ biến cao hơn.

5 Quá trình khai phá dữ liệu (Data Mining)

5.1 Bối cảnh và bài toán

5.1.1 Động lực và bối cảnh hiện tại

Sự bùng nổ của Streaming: Spotify hiện giữ vai trò thống trị trong thị trường âm nhạc toàn cầu và khu vực Châu Á. Dữ liệu streaming mang lại một thước đo khách quan về mức độ đón nhận của khán giả đối với các ca khúc.

Thách thức về dự đoán: Đối với các hãng thu âm, nghệ sĩ và nhà tiếp thị, việc xác định một bài hát có khả năng trở thành “Hit” dựa trên các đặc trưng có sẵn là một bài toán khó nhưng mang lại giá trị kinh tế lớn.

Khoảng trống thị trường: Việc phân tích xu hướng âm nhạc không chỉ ở cấp độ toàn cầu mà còn tại khu vực Châu Á (nơi có sự khác biệt đáng kể về văn hóa và sở thích âm nhạc so với phương Tây) là rất cần thiết.

Việc áp dụng các kỹ thuật khai phá dữ liệu trong lĩnh vực âm nhạc mở ra cơ hội đổi mới, thích ứng và nâng cao lợi thế cạnh tranh cho ngành công nghiệp âm nhạc.

Vì vậy, thông qua phân tích bộ dữ liệu này, dự án hướng đến việc khám phá các yếu tố cơ bản góp phần vào sự thành công của các bài hát, đồng thời xây dựng mô hình dự đoán đủ tốt để cung cấp cái nhìn ban đầu cho người dùng.

5.1.2 Bài toán và các câu hỏi được đặt ra

5.1.2.1. Phân tích mối quan hệ giữa các đặc trưng và mức độ nổi tiếng

Bài toán đầu tiên nhằm xác định và định lượng mối quan hệ giữa các đặc trưng âm nhạc của bài hát và mức độ phổ biến của chúng.

- **Mối quan hệ đơn biến:** Phân tích từng đặc trưng âm học riêng lẻ và ảnh hưởng của chúng đối với điểm popularity.
- **Mối quan hệ đa biến:** Khám phá sự tương tác giữa hai hoặc nhiều đặc trưng cùng lúc. Ví dụ: sự kết hợp giữa loudness và energy ảnh hưởng đến độ phổ biến ra sao, đặc biệt trong các thị trường khác nhau.

5.1.2.2. Định hình xu hướng và sự khác biệt theo khu vực

Câu hỏi trọng tâm: Liệu các yếu tố quyết định thành công ở cấp độ toàn cầu có còn đúng khi xét riêng thị trường Châu Á?

- **Phân tích so sánh vĩ mô:** Xác định các đặc trưng âm học nổi bật (ví dụ: high tempo, low acousticness) của các bài hát thuộc nhóm “Hit Class” tại thị trường Châu Á so với thị trường toàn cầu.

- **Đặc trưng thời gian:** Phân tích vai trò của các đặc trưng thời gian (như days since release, daily movement) trong việc giải thích xu hướng ngắn hạn và dài hạn.

5.1.2.3. Xây dựng mô hình dự đoán khả năng đạt Hit

Đây là trọng tâm của dự án, nhằm chuyển hóa các phân tích thành công cụ dự đoán có tính ứng dụng.

- **Mô hình phân loại đa nhiệm:** Tối ưu hóa các mô hình Machine Learning, đặc biệt là các mô hình dạng cây như Random Forest và XGBoost, để dự đoán khả năng một bài hát thuộc nhóm popularity class (Low, Medium, Hit).
- **Độ tin cậy và khả năng giải thích:** Dánh giá hiệu suất mô hình (Accuracy, F1-score) và áp dụng các kỹ thuật diễn giải đặc trưng để hiểu rõ các yếu tố quan trọng.

5.1.3 Mục tiêu

Mục tiêu OLAP: Cung cấp cái nhìn toàn cảnh đa chiều về xu hướng âm nhạc Châu Á và Thế giới.

Mục tiêu Data Mining: Xây dựng các mô hình máy học để: Dự báo tiềm năng thương mại (Popularity) của một bản demo trước khi phát hành. Phân khúc thị trường để có chiến lược marketing trúng đích.

5.2 Dữ liệu và quá trình tiền xử lý

5.2.1 Giới thiệu và tổng quan về dataset

Tên bộ dữ liệu: Top Spotify Songs in 73 Countries (Daily Updated).

Bộ dữ liệu ghi nhận danh sách các bài hát thịnh hành hàng ngày trên Spotify tại 73 quốc gia khác nhau. Mỗi ngày, dữ liệu lưu trữ Top 50 bài hát phổ biến nhất ở từng quốc gia cùng với các thông tin mô tả liên quan.

Bộ dữ liệu gốc gồm **2,110,316 dòng** và **25 cột**.

5.2.2 Quá trình tiền xử lý

Quy trình tiền xử lý bao gồm các bước chính sau (Quá trình tiền xử lý được mô tả chi tiết tại Mục 1.2 của đồ án):

- Xử lý giá trị NULL.
- Chuyển đổi các trường thời gian (như snapshot date, album release date) về định dạng `datetime`.
- Chuẩn hóa định dạng chuỗi: thay dấu “,” bằng “;” trong các cột tên để tránh lỗi phân tách.
- Xử lý ký tự đặc biệt.

- Giới hạn dữ liệu: Dữ liệu ban đầu rất lớn, vì vậy nhóm chỉ chọn các mẫu thuộc khu vực Châu Á và Global.
- Loại bỏ các thuộc tính không rõ ràng như speechiness, liveness, valence.

Sau khi tiền xử lý, bộ dữ liệu còn lại **522,704 dòng** và **22 cột**.

5.3 Phân tích dữ liệu

Quan sát tổng quan và trực quan hóa dữ liệu trước khi xây dựng mô hình là một bước quan trọng nhằm hiểu rõ cấu trúc, phân phối và mối quan hệ giữa các thuộc tính trong dữ liệu.

5.3.1 Tổng quan

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

warnings.filterwarnings("ignore")
```

Hình 59: Import các thư viện cần thiết

Dữ liệu được nạp vào biến df và quan sát 5 dòng đầu tiên bằng lệnh head().

	spotify_id	name	artists	daily_rank	daily_movement	weekly_movement	country	snapshot_date	popularity	is_explicit	...	album_release_date	danceability	energy	key	loudness
0	2RkZ5LkEzeHGRsmDqKwmaJ	Ordinary	Alex Warren	1	1	0	Global	2025-06-11	95	False	...	2024-09-26	0.368	0.694	2	-6.141
1	42UBPzRMh5yyz0EDPr6fr1	Manchild	Sabrina Carpenter	2	-1	48	Global	2025-06-11	89	True	...	2025-06-05	0.731	0.685	7	-5.087
2	0FTmksd2dxIE5e3rWyJxs6	back to friends	sombr	3	0	1	Global	2025-06-11	98	False	...	2024-12-27	0.436	0.723	1	-2.291
3	7so0lg0zP2Sbgs2d7a1SZ	Die With A Smile	Lady Gaga; Bruno Mars	4	0	-1	Global	2025-06-11	91	False	...	2025-03-07	0.519	0.601	6	-7.727
4	6d0tVTDiauQNBQED0tIAB	BIRDS OF A FEATHER	Billie Eilish	5	1	0	Global	2025-06-11	100	False	...	2024-05-17	0.747	0.507	2	-10.171

5 rows × 22 columns

Hình 60: Load dữ liệu và xem mẫu đầu tiên

Tiếp theo, sử dụng info() để quan sát thông tin các cột như kiểu dữ liệu, giá trị NULL, số lượng dòng và cột.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 522704 entries, 0 to 522703
Data columns (total 22 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   spotify_id      522704 non-null   object 
 1   name             522695 non-null   object 
 2   artists          522704 non-null   object 
 3   daily_rank       522704 non-null   int64  
 4   daily_movement   522704 non-null   int64  
 5   weekly_movement  522704 non-null   int64  
 6   country          522704 non-null   object 
 7   snapshot_date    522704 non-null   object 
 8   popularity       522704 non-null   int64  
 9   is_explicit      522704 non-null   bool   
 10  duration_ms     522704 non-null   int64  
 11  album_name       522704 non-null   object 
 12  album_release_date 522704 non-null   object 
 13  danceability     522704 non-null   float64
 14  energy            522704 non-null   float64
 15  key               522704 non-null   int64  
 16  loudness          522704 non-null   float64
 17  mode              522704 non-null   int64  
 18  acousticness      522704 non-null   float64
 19  instrumentalness 522704 non-null   float64
 20  tempo              522704 non-null   float64
 21  time_signature    522704 non-null   int64  
dtypes: bool(1), float64(6), int64(8), object(7)
memory usage: 84.2+ MB
```

Hình 61: Tổng quan thông tin dữ liệu

Dữ liệu sau tiền xử lý gồm **522,704 dòng** và **22 cột**. Không có giá trị NULL. Kiểu dữ liệu bao gồm: `bool (1), float64 (6), int64 (8), object (7)`.

Tiếp đó, lệnh `describe()` được sử dụng để xem xét các thống kê mô tả của các đặc trưng số học, gồm: giá trị trung bình (mean), lớn nhất (max), nhỏ nhất (min), độ lệch chuẩn (std), và số lượng quan sát.

```
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
<code>daily_rank</code>	522704.0	25.496476	14.431403	1.000000	13.0000	25.00	38.000000	50.000
<code>daily_movement</code>	522704.0	0.857327	6.441476	-49.000000	-1.0000	0.000	2.000000	49.000
<code>weekly_movement</code>	522704.0	2.494096	11.273914	-49.000000	-3.0000	0.000	4.000000	49.000
<code>popularity</code>	522704.0	75.662373	14.145782	0.000000	66.0000	77.000	87.000000	100.000
<code>duration_ms</code>	522704.0	207200.846397	47747.703457	41487.000000	173746.00000	204078.000	237485.000000	933407.000
<code>danceability</code>	522704.0	0.631018	0.145013	0.093900	0.5240	0.644	0.744000	0.988
<code>energy</code>	522704.0	0.610951	0.175153	0.001740	0.4840	0.618	0.747000	0.993
<code>key</code>	522704.0	5.371384	3.622043	0.000000	2.0000	6.000	9.000000	11.000
<code>loudness</code>	522704.0	-6.957477	2.849924	-54.341000	-8.4450	-6.588	-5.038000	2.605
<code>mode</code>	522704.0	0.628979	0.483078	0.000000	0.0000	1.000	1.000000	1.000
<code>acousticness</code>	522704.0	0.331332	0.285505	0.000008	0.0693	0.255	0.562000	0.996
<code>instrumentalness</code>	522704.0	0.018902	0.100149	0.000000	0.0000	0.0000	0.000069	0.995
<code>tempo</code>	522704.0	121.340284	28.407271	46.718000	99.9740	119.992	139.972000	213.503
<code>time_signature</code>	522704.0	3.919693	0.359057	1.000000	4.0000	4.000	4.000000	5.000

Hình 62: Thống kê mô tả cho các đặc trưng số học

5.3.2 Phân tích dữ liệu đơn biến

Các đặc trưng phân loại

```
# countplot for discrete feature
col_to_plot = ['country', 'key', 'mode', 'is_explicit', 'time_signature']

num_cols = 2
num_rows = (len(col_to_plot) + num_cols - 1) // num_cols

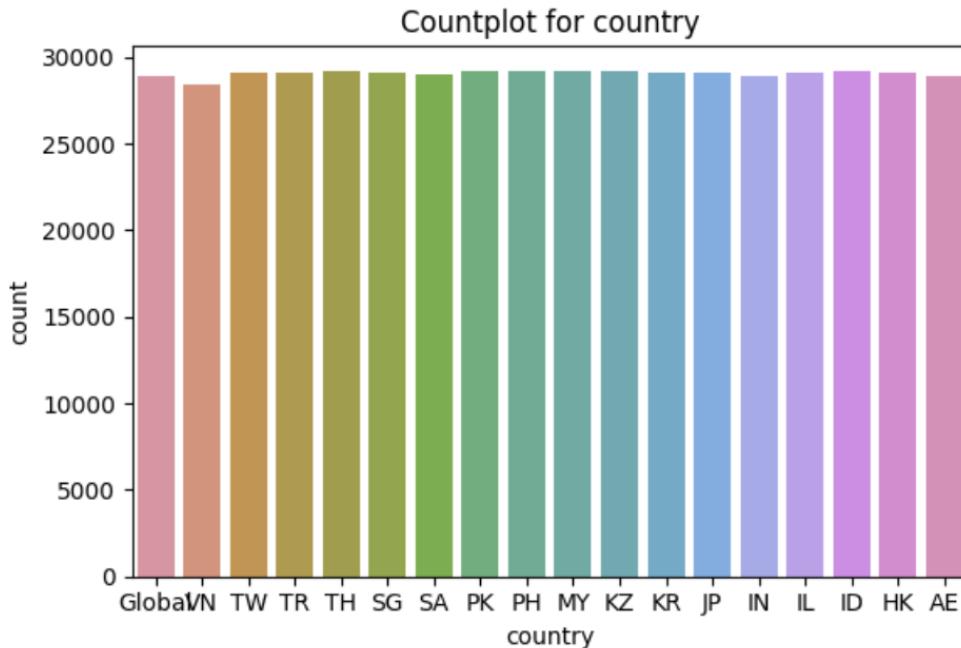
plt.figure(figsize=(12, num_rows * 4))

for i, col in enumerate(col_to_plot, 1):
    plt.subplot(num_rows, num_cols, i)
    sns.countplot(data=df, x=col)
    plt.title(f'Countplot for {col}')

plt.tight_layout()
plt.show()
```

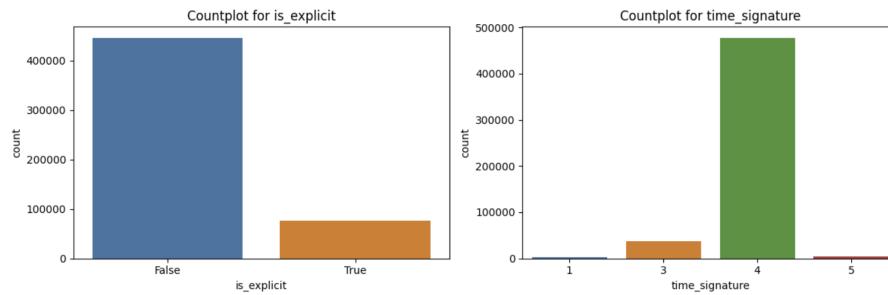
Hình 63: Code trực quan hóa các đặc trưng phân loại

Country



- Dữ liệu phân phối tương đối đồng đều giữa các quốc gia, không có hiện tượng mất cân bằng nghiêm trọng.
- Mỗi quốc gia có khoảng 28.000 mẫu, giúp mô hình học được xu hướng riêng biệt theo từng thị trường mà không bị thiên lệch.

Is _ explicit và Time _ signature



- Hai đặc trưng này có phân phối lệch mạnh.
- **Is _ explicit**: phần lớn giá trị là `False`, phản ánh xu hướng các bảng xếp hạng ưu tiên ca khúc có nội dung phù hợp với đại chúng.
- **Time _ signature**: đa số bài hát sử dụng nhịp 4/4, chuẩn công nghiệp của Pop và Dance – hai thể loại phổ biến trong Top Charts toàn cầu và Châu Á.

Các đặc trưng số học

Dưới đây là trực quan hóa phân phối và kiểm tra giá trị ngoại lai (outliers):

```
# hist plot + kde for continuous feature
col_to_plot = [
    'duration_ms', 'loudness', 'tempo', 'popularity', 'daily_movement',
    'weekly_movement', 'acousticness', 'danceability',
    'instrumentalness', 'energy'
]

num_cols = 2
num_rows = (len(col_to_plot) + num_cols - 1) // num_cols

plt.figure(figsize=(12, num_rows * 4))

for i, col in enumerate(col_to_plot, 1):
    plt.subplot(num_rows, num_cols, i)
    sns.histplot(df[col], kde=True, bins=10)
    plt.title(f'Histogram + KDE for {col}')

plt.tight_layout()
plt.show()
```

Hình 64: Code vẽ phân phối các đặc trưng số học liên tục

Popularity

```

# hist plot + kde for continuous feature
col_to_plot = [
    'duration_ms', 'loudness', 'tempo', 'popularity', 'daily_movement',
    'weekly_movement', 'acousticness', 'danceability',
    'instrumentalness', 'energy'
]

num_cols = 2
num_rows = (len(col_to_plot) + num_cols - 1) // num_cols

plt.figure(figsize=(12, num_rows * 4))

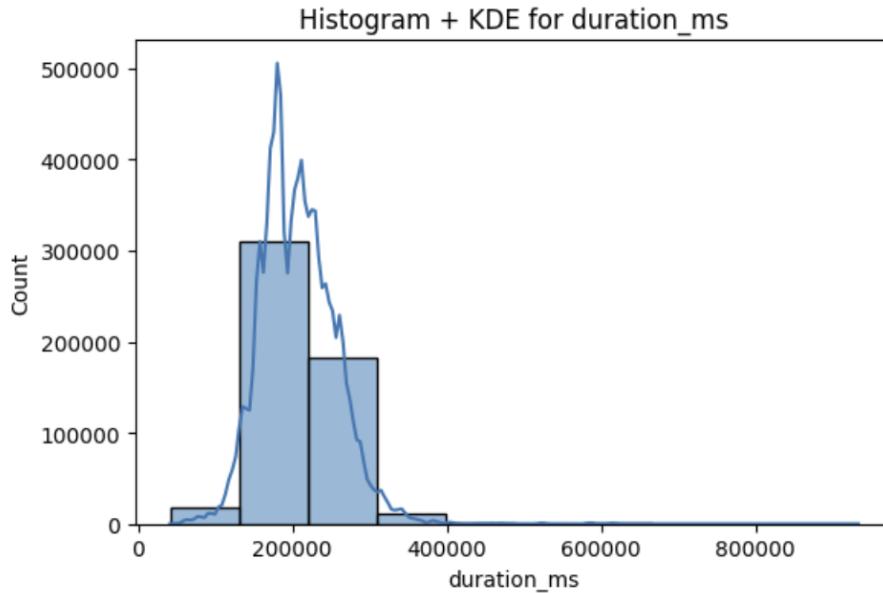
for i, col in enumerate(col_to_plot, 1):
    plt.subplot(num_rows, num_cols, i)
    sns.histplot(df[col], kde=True, bins=10)
    plt.title(f'Histogram + KDE for {col}')

plt.tight_layout()
plt.show()

```

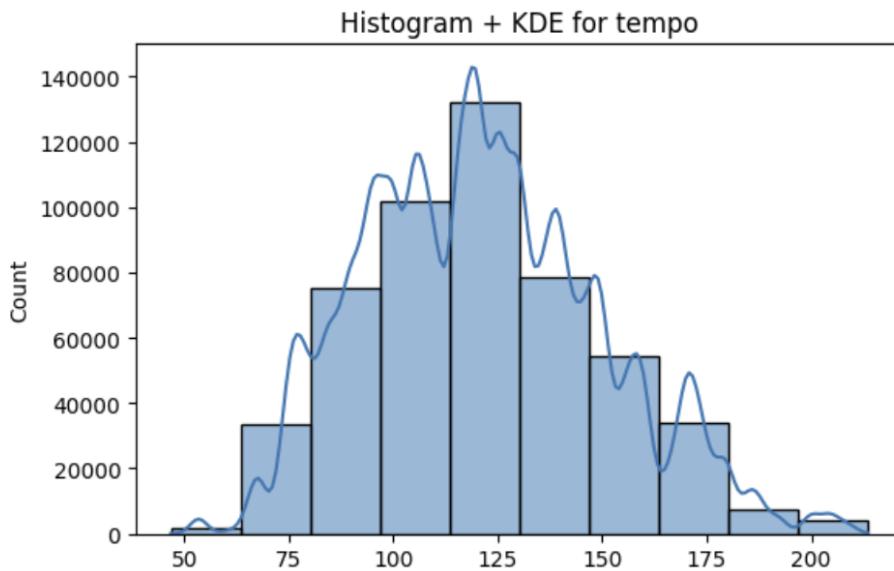
- Giá trị nằm trong khoảng 0–100, đúng với thang đo của Spotify.
- Phân phối lệch phải, tập trung ở vùng 60–100.
- Do dữ liệu được thu thập từ các bảng xếp hạng, nên chỉ những bài hát đã tương đối nổi tiếng mới xuất hiện.

Duration (ms)



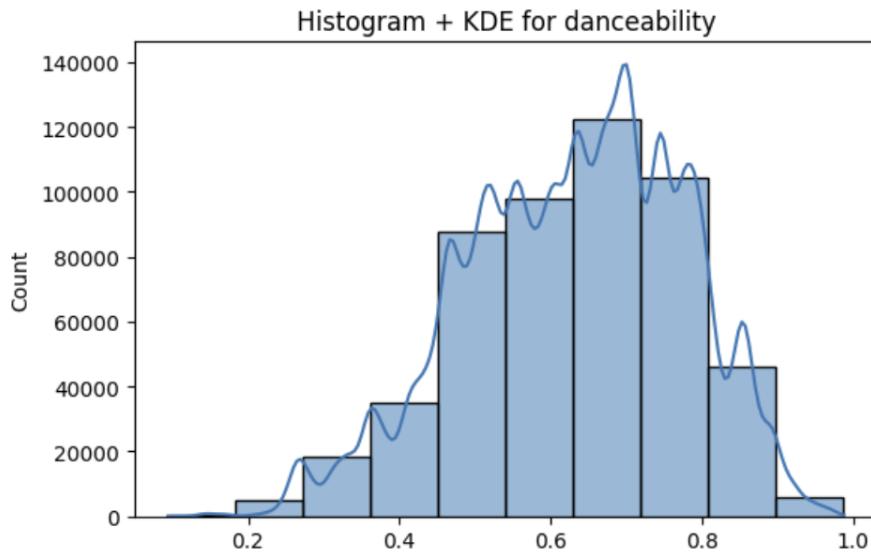
- Thời lượng trải rộng từ khoảng 100.000 ms đến 800.000 ms.
- Đa số bài hát có độ dài 150.000–300.000 ms.
- Các bài hát hiện đại có xu hướng ngắn hơn, phù hợp hành vi nghe nhạc trên nền tảng streaming và video ngắn.

Tempo



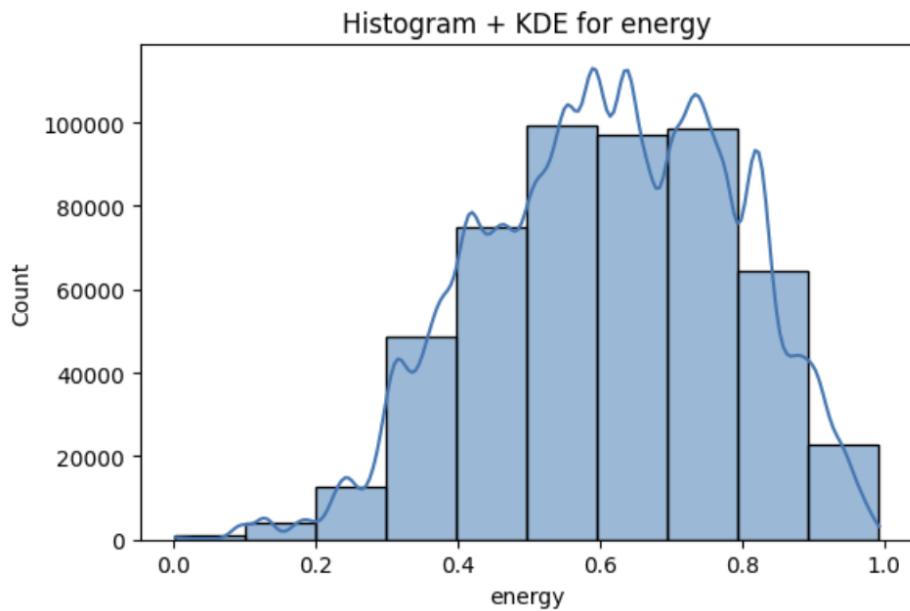
- Phân phối tương đối đều, nằm trong khoảng 50–200 BPM.
- Tập trung nhiều ở vùng 100–150 BPM – đặc trưng của nhạc Pop/Dance.
- Phản ánh sự đa dạng thể loại trong bảng xếp hạng.

Danceability



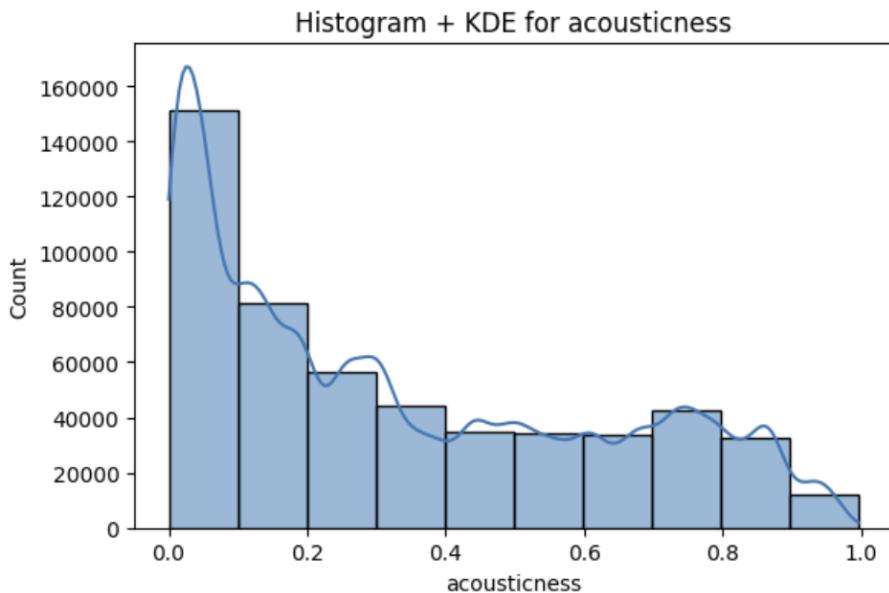
- Phân phối lệch phải trong khoảng 0–1.
- Tập trung ở mức 0.5–0.7.
- Các bài hát phổ biến thường có nhịp ổn định, dễ nghe và phù hợp với playlist giải trí hoặc luyện tập – đặc trưng nổi bật trong nhạc Pop châu Á và K-pop.

Energy



- Phân phối khá đồng đều trên toàn khoảng 0–1.
- Phần lớn giá trị nằm trong khoảng 0.3–0.9.
- Thị trường âm nhạc có sự cân bằng giữa ca khúc sôi động (energy cao) và ballad/chill (energy thấp hơn).

Acousticness



- Phân phối lệch trái mạnh.
- Tập trung cao ở vùng gần 0 và giảm dần về phía 1.

- Diều này phản ánh sự thống trị của các bài hát sản xuất điện tử so với các bản acoustic truyền thống trên nền tảng streaming.

5.3.3 Phân tích dữ liệu đa biến

Phần này thực hiện phân tích sự tương quan giữa các đặc trưng âm nhạc và điểm popularity, đồng thời mô tả các mối quan hệ quan trọng giữa các cặp biến.

```

target_col = 'popularity'
compare_col = ['duration_ms', 'loudness', 'tempo', 'daily_movement',
               'acousticness', 'danceability', 'instrumentalness', 'energy']

num_cols = 2
num_rows = (len(compare_col) + num_cols - 1) // num_cols
plt.figure(figsize=(15, num_rows * 5))

for i, col in enumerate(compare_col, 1):
    plt.subplot(num_rows, num_cols, i)
    df['bin'] = pd.qcut(df[col], q=15, duplicates='drop')

    # Vẽ Boxplot cho từng giờ
    sns.boxplot(data=df, x='bin', y=target_col, palette='viridis')

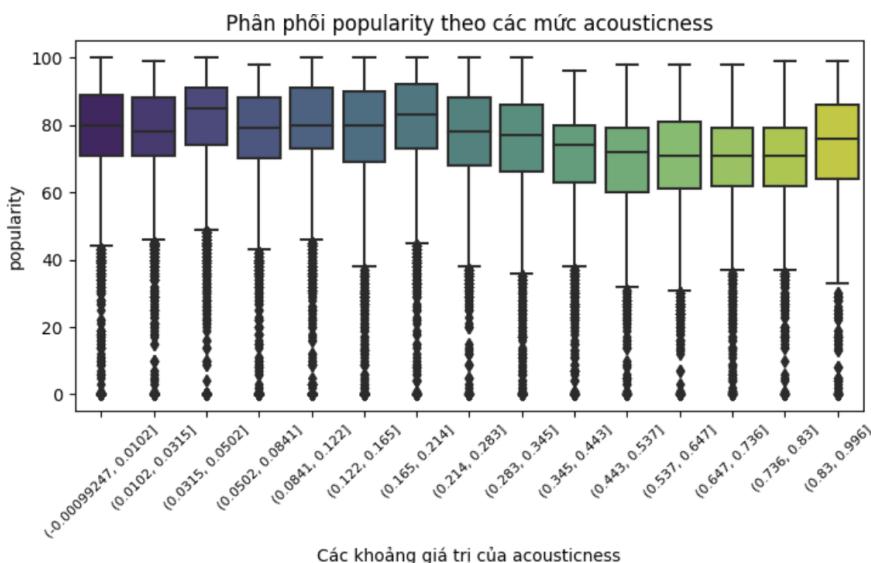
    # Trang trí lại trục X cho dễ đọc (chi lấy giá trị đại diện)
    plt.title(f'Phân phối {target_col} theo các mức {col}')
    plt.xticks(rotation=45, fontsize=8)
    plt.xlabel(f'Các khoảng giá trị của {col}')

plt.tight_layout()
plt.show()

```

Hình 65: Code trực quan hóa các đặc trưng liên quan đến popularity

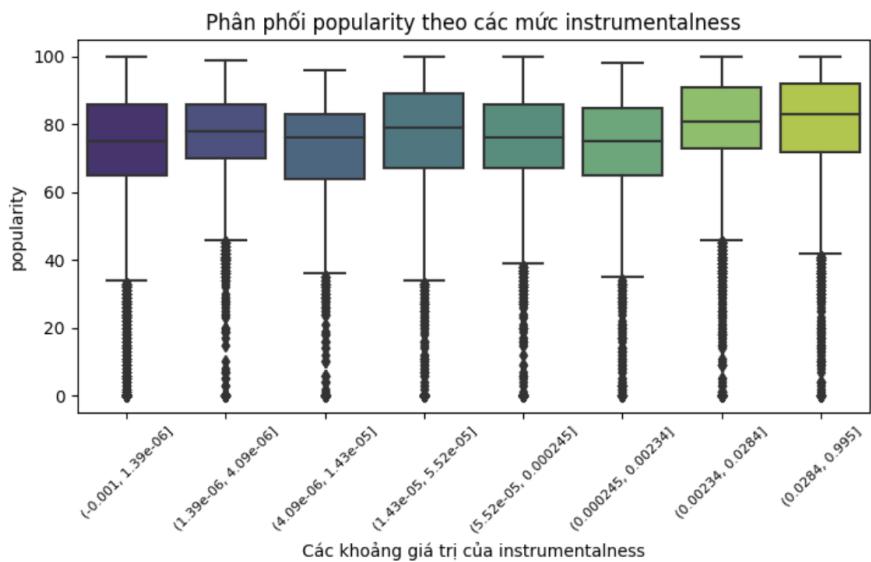
Popularity – Acousticness



- Các bài hát có acousticness thấp (ghi lệch trái) thường có điểm popularity trung bình cao hơn.

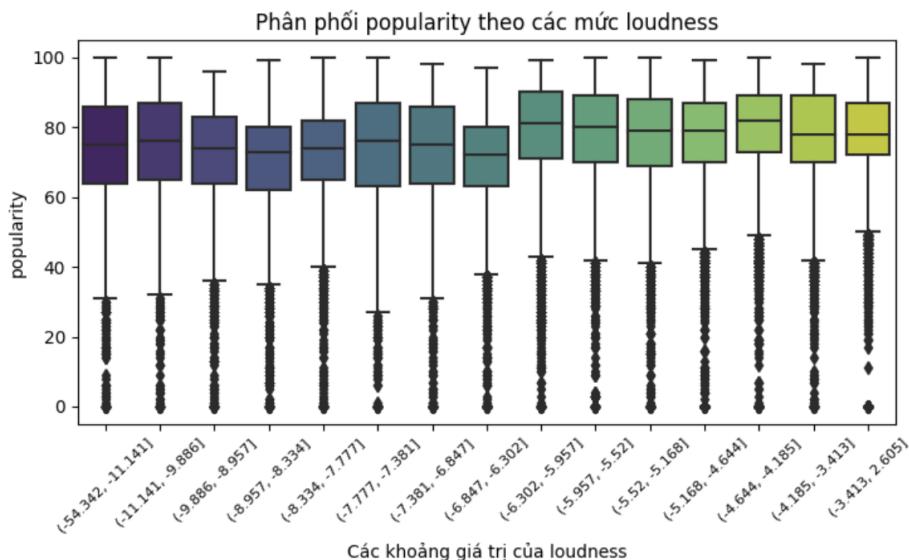
- Điều này cho thấy những bài hát mang phong cách Pop, Hip-hop, EDM, hoặc nhạc phòng thu với nhiều nhạc cụ điện tử (điểm acousticness gần 0) có xu hướng phổ biến hơn.

Popularity – Instrumentalness



- Các bài hát có instrumentalness cao có xu hướng đạt điểm popularity cao hơn.

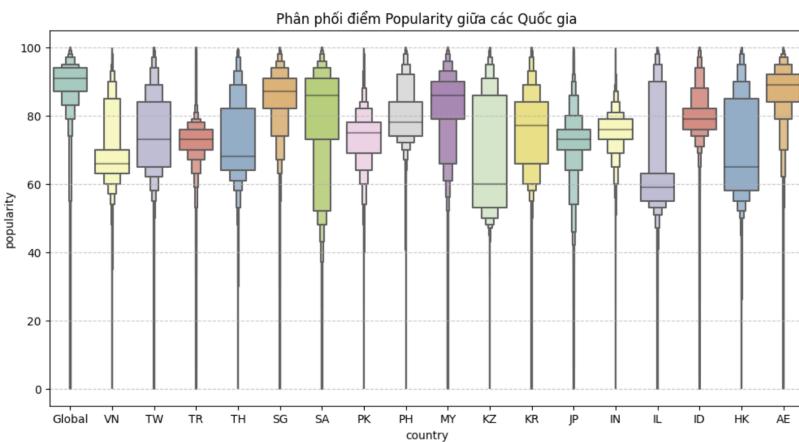
Popularity – Loudness



- Những bài hát có mức loudness cao (từ -6.3 đến -3 dB) thường có điểm popularity cao hơn.
- Điều này phản ánh chiến lược sản xuất âm nhạc hiện đại: các bài hát được nén và tăng âm lượng mạnh nhằm tạo tác động mạnh trên radio và nền tảng streaming.

Popularity – Country

```
plt.figure(figsize=(12, 6))
sns.boxenplot(data=df, x='country', y='popularity', palette='Set3')
plt.title('Phân phối điểm Popularity giữa các Quốc gia')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

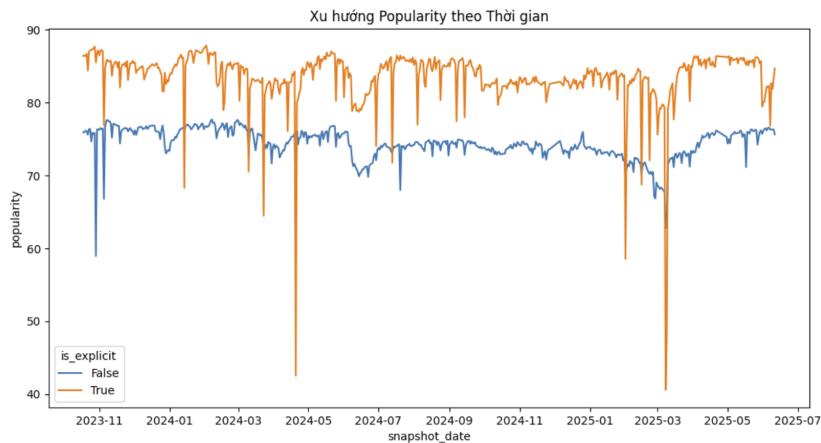


Hình 66: Code và kết quả trực quan hóa popularity theo quốc gia

- Điểm popularity của bảng xếp hạng Global nhìn chung cao hơn đa số các nước tại khu vực châu Á.
- Một số quốc gia có điểm trung bình thấp: IL, VN, TG.
- Điều này xảy ra vì bảng Global chứa các bài hát “siêu hit” với độ phổ biến rất cao, trong khi bảng xếp hạng quốc gia bị chi phối bởi gu âm nhạc địa phương.

Popularity – Snapshot Date

```
plt.figure(figsize=(12, 6))
# Chuyển snapshot_date sang datetime nếu chưa chuyển
df['snapshot_date'] = pd.to_datetime(df['snapshot_date'])
sns.lineplot(data=df, x='snapshot_date', y='popularity', hue='is_explicit', errorbar=None)
plt.title('Xu hướng Popularity theo Thời gian')
plt.show()
```

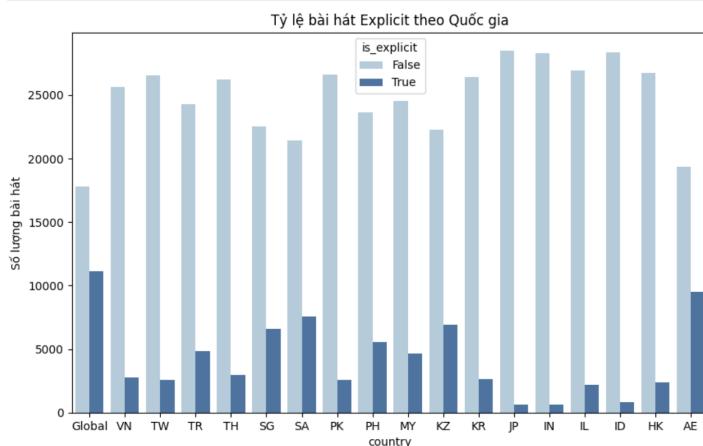


Hình 67: Biến động điểm popularity theo thời gian

- Số ngày có điểm popularity khá ổn định, dao động nhỏ.
- Một số ngày xuất hiện sự sụt giảm lớn (trên 30 điểm), cho thấy khả năng xuất hiện các yếu tố ngoại lai như:
 - ngày lễ lớn,
 - thay đổi cơ chế thu thập dữ liệu,
 - nhiều ca khúc mới đồng loạt ra mắt.

Explicit Content – Country

```
plt.figure(figsize=(10, 6))
# Vẽ biểu đồ đếm có phân nhóm
sns.countplot(data=df, x='country', hue='is_explicit', palette='Paired')
plt.title('Tỷ lệ bài hát Explicit theo Quốc gia')
plt.ylabel('Số lượng bài hát')
plt.show()
```



Hình 68: Phân bố số bài hát có yếu tố nhạy cảm theo quốc gia

- Số lượng bài hát không có yếu tố bạo lực luôn chiếm đa số ở tất cả quốc gia.
- Các khu vực có tỷ lệ bài hát explicit cao: Global, AE, KZ, SA, TG.
- Các quốc gia có tỷ lệ rất thấp: JP, ID, IN.
- Điều này phản ánh sự khác biệt văn hóa và mức độ kiểm duyệt nội dung giữa các thị trường âm nhạc khu vực.

Trung bình các đặc trưng theo Country

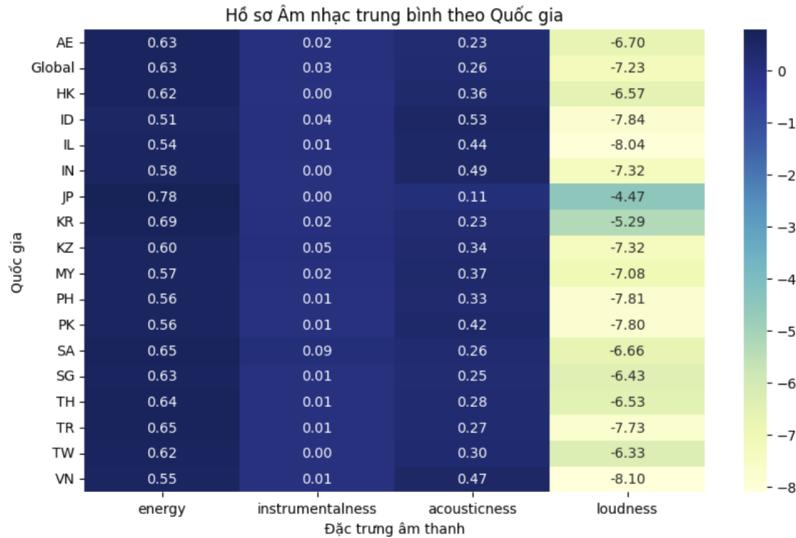
```

features = ['energy', 'instrumentalness', 'acousticness', 'loudness']

# Tính trung bình theo quốc gia
country_profile = df.groupby('country')[features].mean()

plt.figure(figsize=(10, 6))
sns.heatmap(country_profile, annot=True, cmap='YlGnBu', fmt=".2f")
plt.title('Hồ sơ Âm nhạc trung bình theo Quốc gia')
plt.ylabel('Quốc gia')
plt.xlabel('Đặc trưng âm thanh')
plt.show()

```



Hình 69: Giá trị trung bình của một số đặc trưng theo từng quốc gia

- Nhìn chung, các đặc trưng số giữa các quốc gia khá tương đồng.
- Tuy nhiên, có nhiều khác biệt đáng chú ý:
 - instrumentalness của SA cao vượt trội (0.09),
 - energy tại JP rất cao,
 - acousticness tại JP rất thấp,
 - loudness tại JP và KR thấp hơn hẳn.
- Điều này phản ánh gu âm nhạc đặc thù của từng thị trường, đặc biệt là Nhật Bản và Hàn Quốc.

5.3.4 Ma trận hiệp tương quan (Correlation Matrix)

Ma trận tương quan thể hiện mức độ tuyến tính giữa các đặc trưng số, với giá trị trong khoảng $[-1, 1]$. Giá trị gần 1 biểu thị quan hệ thuận mạnh, gần -1 biểu thị quan hệ nghịch mạnh.

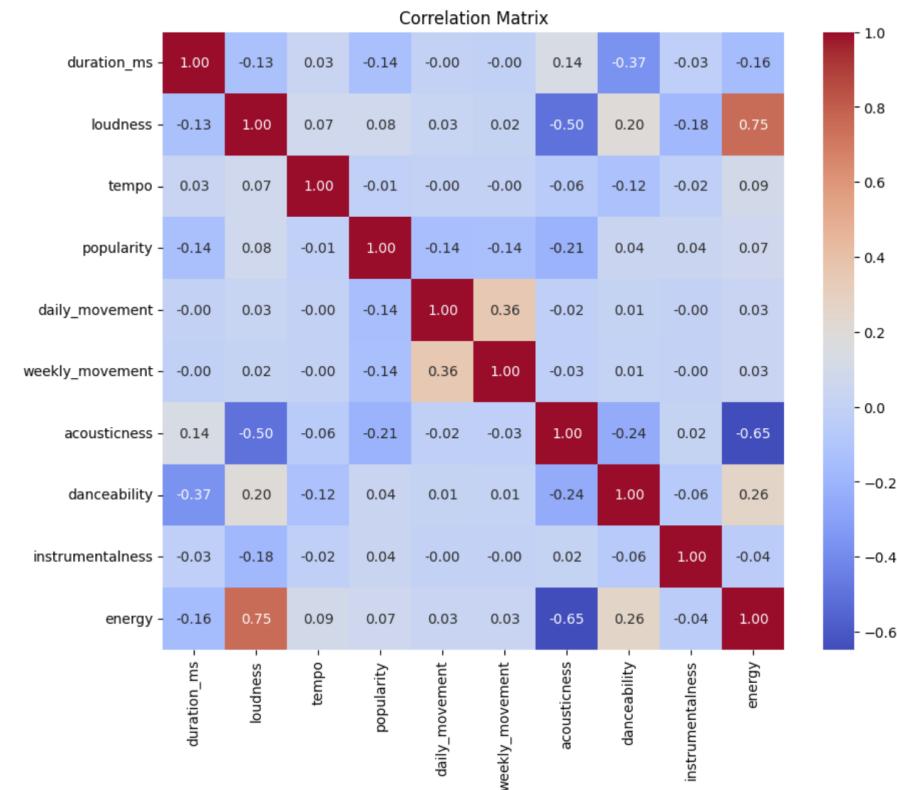
```

# correlation matrix
numeric_cols = ['duration_ms', 'loudness', 'tempo', 'popularity', 'daily_movement', 'weekly_movement', 'acousticness', 'danceability', 'instrumentalness', 'energy']
corr_matrix = df[numeric_cols].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()

```

Hình 70: Code tạo ma trận hiệp tương quan



Hình 71: Ma trận hiệp tương quan giữa các đặc trưng

Nhận xét

- **Energy – Loudness** tương quan cao (0.75): bài hát lớn tiếng thường được cảm nhận là giàu năng lượng.
- **Loudness – Acousticness** tương quan nghịch mạnh (-0.5): nhạc acoustic có âm lượng thấp rất thấp.
- **Energy – Acousticness** tương quan nghịch (0.65): nhạc năng lượng cao gần như không thể là nhạc acoustic thuần.
- Các cặp đặc trưng khác đa số tương quan yếu (trong khoảng [-0.2, 0.2]).

Điều này cho thấy các đặc trưng nhìn chung khá độc lập, rất thuận lợi cho mô hình Machine Learning vì hạn chế đa cộng tuyến và giúp mô hình học hiệu quả hơn.

5.4 Mô hình dự đoán

5.4.1 Tổng quan và cách đánh giá

Bài toán

Bài toán chính của đồ án là: **dự đoán khả năng một bài hát trở thành “Hit” trong tương lai** dựa trên các đặc trưng âm thanh cùng với thông tin thời gian và địa lý.

Bài toán được mô tả dưới dạng phân loại đa lớp với ba nhãn (Low, Medium, Hit) suy ra từ biến liên tục **popularity** theo ba ngưỡng phân chia (ví dụ: 0–60, 61–80, 81–100).

Dữ liệu

- **Input:** các đặc trưng âm thanh (numeric), cùng các biến thời gian và địa lý đã được tiền xử lý.
- **Target:** popularity_class — nhãn phân loại 3 lớp.

Các mô hình được thử nghiệm

Để so sánh hiệu năng và độ phức tạp, nhóm sử dụng ba mô hình phân loại có mức phức tạp tăng dần:

1. Logistic Regression
2. XGBoost Classifier
3. Random Forest Classifier

Mục tiêu là tối ưu và so sánh các mô hình này trên cùng một quy trình đánh giá, rồi chọn mô hình phù hợp nhất theo tập hợp các metric đã định nghĩa dưới đây.

5.4.2 Giao thức đánh giá

- **Giữ thứ tự thời gian:** chia bộ dữ liệu theo thời gian (Train / Test theo time series), đảm bảo dữ liệu tập test đều là tương lai của tập train. Kết quả báo cáo chính là kết quả trên tập test.
- **Tinh chỉnh siêu tham số:** Sử dụng GridSearchCV để thử nhiều bộ siêu tham số nhằm tìm ra bộ số tốt nhất.
- **Chuẩn hóa và pipelines:** dùng Pipeline (scaler + model) để đảm bảo quy trình tiền xử lý phù hợp.

5.4.3 Các bộ metric sử dụng

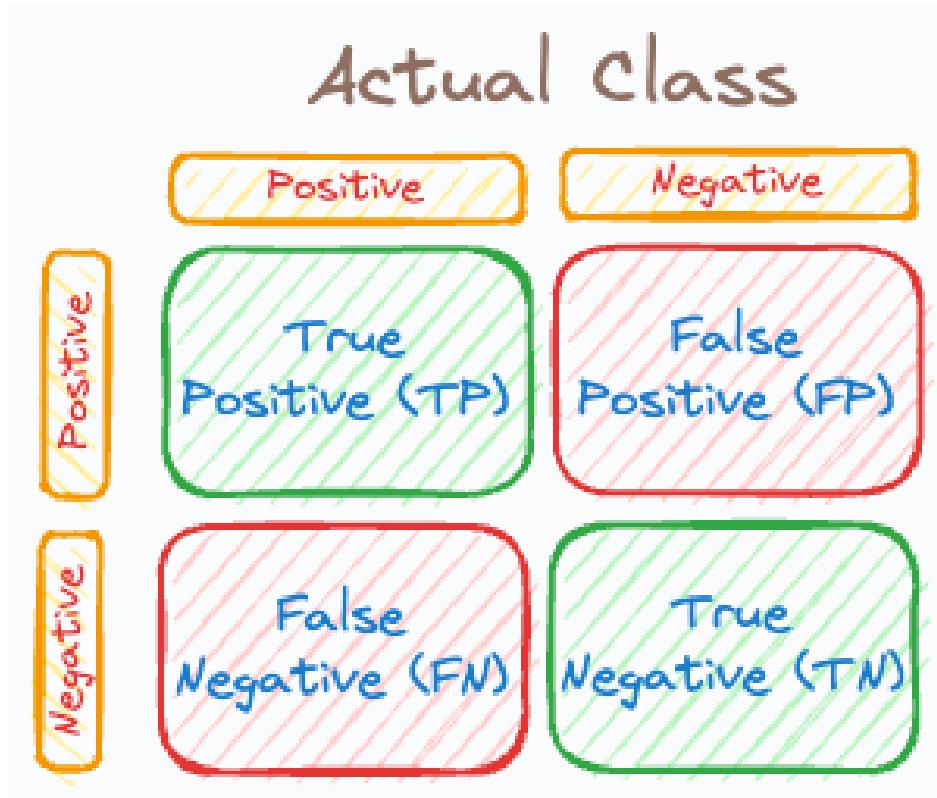
Để đánh giá toàn diện mô hình phân loại đa lớp, báo cáo sẽ bao gồm các chỉ số sau: **Accuracy (train vs test)**, **ROC**, **Classification report (precision, recall, F1, support)**. Dưới đây là định nghĩa toán học và ý nghĩa thực tiễn của từng metric.

5.4.3.1. Accuracy

$$\text{Accuracy} = \frac{\text{Số dự đoán đúng}}{\text{Tổng số mẫu}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Ý nghĩa: tỉ lệ dự đoán chính xác trên tất cả các lớp. Dễ hiểu và phổ biến.

5.4.3.2. Precision, Recall, F1-score (theo từng lớp)



Cho một lớp cụ thể, định nghĩa:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Precision:** trong số những mẫu dự đoán là lớp X, tỉ lệ đoán đúng.
- **Recall :** trong số các mẫu thực sự thuộc lớp X, tỉ lệ được mô hình bắt đúng.
- **F1-score:** là trung bình điều hòa của precision và recall, dùng cân bằng giữa hai chỉ số.

5.4.3.3. Các aggregation cho multi-class

Với multi-class ta đánh giá các aggregation sau:

- **Macro-average:** trung bình đơn giản các chỉ số trên từng lớp.
- **Micro-average:** tổng TP, FP, FN trên tất cả các lớp rồi tính chỉ số (phù hợp khi muốn trọng số theo tần suất mẫu).
- **Weighted-average:** trung bình các chỉ số theo trọng số là support mỗi lớp (cân bằng giữa macro và micro).

5.4.3.4. ROC Curve

Với bài toán nhãn nhiều lớp, ta dùng chiến lược **One-vs-Rest (OvR)**:

- Với OvR, mỗi lớp được xét là “positive” so với phần còn lại; từ đó vẽ đường ROC cho từng lớp.
- **AUC (Area Under Curve):** là diện tích dưới đường ROC, giá trị nằm trong $[0, 1]$. AUC lớn hơn cho thấy model phân biệt càng tốt và dự đoán càng chính xác.

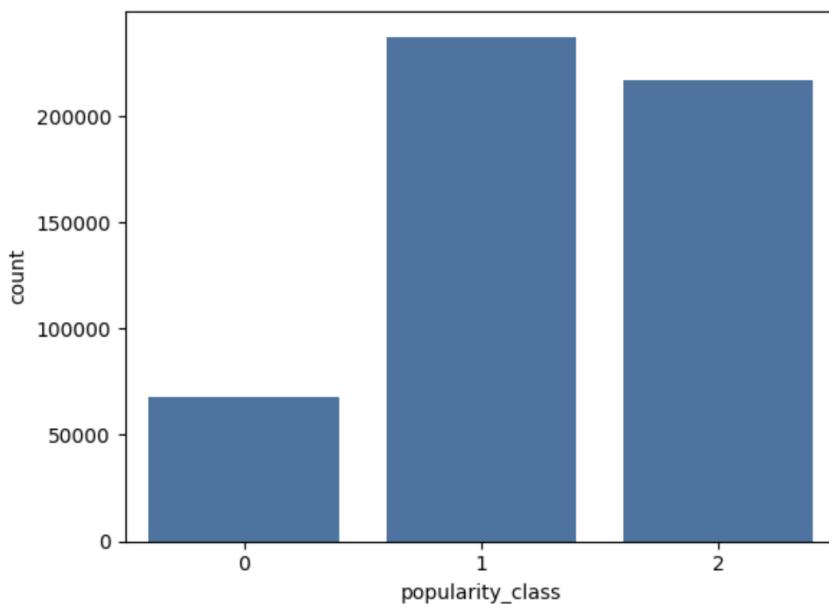
5.4.4 Xử lý và lựa chọn các đặc trưng

Tạo đặc trưng mục tiêu (Target)

```
def classify_popularity(score):
    if score >= 80:
        return 2 # Hit
    elif score >= 60:
        return 1 # Average
    else:
        return 0 # Low

df['popularity_class'] = df['popularity'].apply(classify_popularity)
```

Từ biến liên tục popularity, nhóm xây dựng nhãn phân loại popularity_class gồm 3 mức độ dựa trên ba ngưỡng 60 – 80 – 100.



Hình 72: Phân phối các lớp

Phân phối các lớp tương đối cân bằng, không xuất hiện hiện tượng mất cân bằng nghiêm trọng.

Xử lý dữ liệu thời gian

```

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

df = df.sort_values(by=['spotify_id', 'snapshot_date'])

df['snapshot_date'] = pd.to_datetime(df['snapshot_date'])
df['album_release_date'] = pd.to_datetime(df['album_release_date'])

# Đặc trưng từ Snapshot Date (Thời điểm xếp hạng)
df['snapshot_month'] = df['snapshot_date'].dt.month
df['snapshot_dayofweek'] = df['snapshot_date'].dt.dayofweek
df['is_weekend'] = (df['snapshot_dayofweek'] >= 5).astype(int)

# Số ngày bài hát ra mắt tính đến ngày xếp hạng (Độ mới/tuổi của bài hát)
df['days_since_release'] = (df['snapshot_date'] - df['album_release_date']).dt.days

# Tạo đặc trưng độ trễ cho daily_movement
df['daily_movement_lag1'] = df.groupby('spotify_id')['daily_movement'].shift(1).fillna(0)

```

Hình 73: Xử lý dữ liệu thời gian

Chuyển `snapshot_date` và `album_release_date` sang dạng `datetime` để thuận tiện cho việc xử lý.

Từ `snapshot_date` trích xuất thêm các đặc trưng: `day_of_week`, `month` và `is_weekend`.

Tạo đặc trưng `days_since_release` bằng hiệu giữa thời điểm snapshot và ngày phát hành album.

Tạo đặc trưng độ trễ cho `daily_movement` nhằm đưa thông tin xu hướng gần nhất của bài hát vào mô hình.

Xử lý dữ liệu dạng object và bool

```

df['is_explicit'] = df['is_explicit'].astype(int)

ohe_cols = ['country']
df = pd.get_dummies(df, columns=ohe_cols, drop_first=True)

bool_cols = df.select_dtypes(include=['bool']).columns

for col in bool_cols:
    df[col] = df[col].astype(int)

df = df.sort_values(by='snapshot_date')

```

Hình 74: Label dữ liệu object và chuyển kiểu dữ liệu bool sang int

Da số các trường dạng object không mang thông tin hữu ích; ngoại lệ là `country`.

Vì `country` là biến phân loại không có tính thứ tự, nhóm sử dụng One-Hot Encoding để chuyển thành các biến nhị phân.

Các feature dạng bool được ánh xạ sang 0/1 để thuận lợi cho mô hình, đặc biệt là Logistic Regression.

Loại bỏ các cột không cần thiết

```
cols_to_drop = [
    'spotify_id', 'name', 'artists', 'album_name',
    'snapshot_date', 'album_release_date', 'popularity'
]
df = df.drop(columns=cols_to_drop)
df.info()
```

Hình 75: Loại bỏ các cột không cần thiết

Loại bỏ các trường không mang giá trị mô hình hóa như ID, name.

Loại bỏ album_release_date, snapshot_date và country vì đã được chuyển hóa thành các đặc trưng mới.

Loại bỏ popularity để tránh mô hình học nội suy trực tiếp từ giá trị gốc.

Dữ liệu sau xử lý

```
Data columns (total 37 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   daily_rank       522704 non-null   int64  
 1   daily_movement   522704 non-null   int64  
 2   weekly_movement  522704 non-null   int64  
 3   is_explicit     522704 non-null   int64  
 4   duration_ms     522704 non-null   int64  
 5   danceability    522704 non-null   float64 
 6   energy          522704 non-null   float64 
 7   key              522704 non-null   int64  
 8   loudness         522704 non-null   float64 
 9   mode             522704 non-null   int64  
 10  acousticness    522704 non-null   float64 
 11  instrumentalness 522704 non-null   float64 
 12  tempo            522704 non-null   float64 
 13  time_signature  522704 non-null   int64  
 14  popularity_class 522704 non-null   int64  
 15  snapshot_month  522704 non-null   int32  
 16  snapshot_dayofweek 522704 non-null   int32  
 17  is_weekend       522704 non-null   int64  
 18  days_since_release 522704 non-null   int64  
 19  daily_movement_lag1 522704 non-null   float64 
 20  country_Global  522704 non-null   int64  
 21  country_HK      522704 non-null   int64  
 22  country_ID      522704 non-null   int64  
 23  country_IL      522704 non-null   int64  
 24  country_IN      522704 non-null   int64  
 25  country_JP      522704 non-null   int64  
 26  country_KR      522704 non-null   int64  
 27  country_KZ      522704 non-null   int64  
 28  country_MY      522704 non-null   int64  
 29  country_PH      522704 non-null   int64  
 30  country_PK      522704 non-null   int64  
 31  country_SA      522704 non-null   int64  
 32  country_SG      522704 non-null   int64  
 33  country_TH      522704 non-null   int64  
 34  country_TR      522704 non-null   int64  
 35  country_TW      522704 non-null   int64  
 36  country_VN      522704 non-null   int64  
dtypes: float64(7), int32(2), int64(28)
memory usage: 147.6 MB
```

Hình 76: Dữ liệu sau khi xử lý

Bộ dữ liệu sau xử lý gồm 37 đặc trưng dạng số (int/float), phù hợp để đưa vào các mô hình phân loại.

Chia dữ liệu

```

train_ratio = 0.8
split_index = int(len(df) * train_ratio)

train_df = df.iloc[:split_index]
test_df = df.iloc[split_index:]

X_train = train_df.drop(columns=['popularity_class'])
y_train = train_df['popularity_class']

X_test = test_df.drop(columns=['popularity_class'])
y_test = test_df['popularity_class']

```

Hình 77: Chia dữ liệu

Dữ liệu được sắp xếp theo `snapshot_date`, sau đó chia theo tỉ lệ 80–20 cho tập train-test. Vì mục tiêu là dự đoán mức độ phổ biến trong tương lai, tập test buộc phải là các mẫu xuất hiện sau thời điểm của tập train nhằm mô phỏng đúng bối cảnh dự báo.

5.4.5 Logistic Regression

Logistic Regression là một thuật toán phân loại tuyến tính (Linear Classification) dựa trên hàm Sigmoid (Logit Function). Thay vì dự đoán trực tiếp nhãn lớp, mô hình ước lượng xác suất một mẫu thuộc về một lớp cụ thể; từ đó gán nhãn dựa trên ngưỡng phân loại. Với tính chất tuyến tính, Logistic Regression thường được sử dụng như một mô hình cơ sở (Baseline Model) để so sánh với các mô hình phi tuyến phức tạp hơn.

Xây dựng và huấn luyện mô hình

```

1 from sklearn.pipeline import Pipeline
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.linear_model import LogisticRegression
4 from sklearn.model_selection import GridSearchCV, TimeSeriesSplit
5
6 # 1. TimeSeriesSplit cho dữ liệu time series
7 tscv = TimeSeriesSplit(n_splits=4)
8
9 # 2. Pipeline scale dữ liệu -> dự đoán
10 pipeline = Pipeline([
11     ("scaler", StandardScaler()),
12     ("lr", LogisticRegression(max_iter=1000, multi_class="auto"))
13 ])
14
15 # 3. Parameter Grid cho Logistic Regression
16 param_grid = {
17     "lr_C": [0.01, 0.1, 1.0, 10],
18     "lr_penalty": ["l2"],
19     "lr_solver": ["lbfgs", "saga"],
20 }
21
22 # 4. GridSearchCV
23 grid_search = GridSearchCV(
24     estimator=pipeline,
25     param_grid=param_grid,
26     cv=tscv,
27     scoring="accuracy",
28     n_jobs=-1,
29     verbose=2
30 )
31 # 5. Fit
32 grid_search.fit(X_train, y_train)

```

Hình 78: Mã nguồn xây dựng và huấn luyện mô hình Logistic Regression

Trước khi huấn luyện mô hình, dữ liệu được chuẩn hóa bằng **StandardScaler** nhằm đảm bảo các thuộc tính có phân phối phù hợp với thuật toán tối ưu dựa trên Gradient Descent.

Bộ tham số của mô hình được lựa chọn thông qua **GridSearchCV**, thử nghiệm toàn bộ các tổ hợp siêu tham số được định nghĩa trước. Nhóm đã tinh chỉnh các tham số:

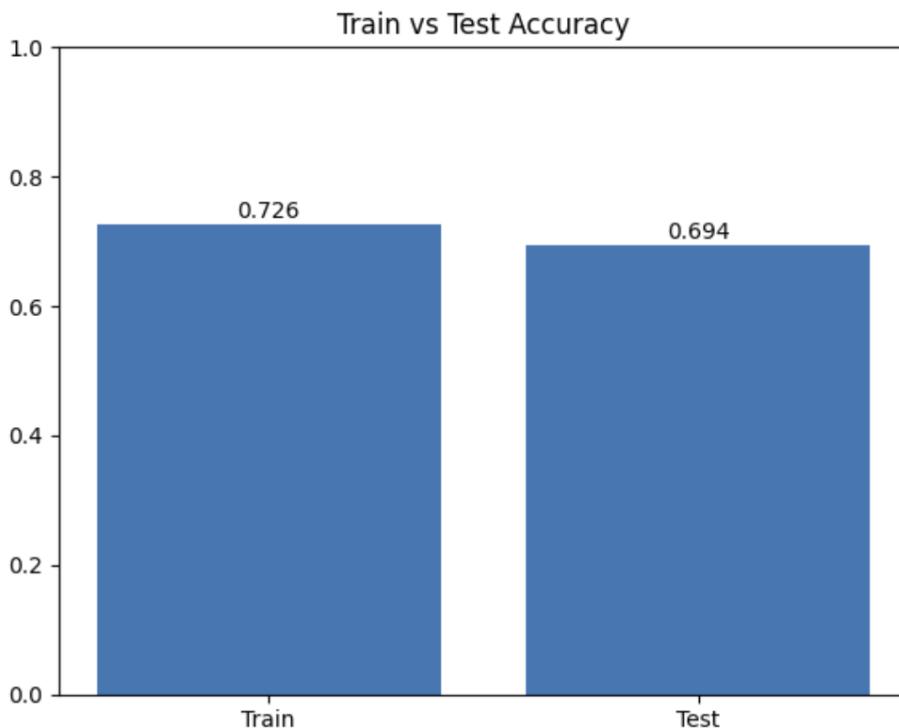
- C
- penalty
- solver

Để phù hợp với dữ liệu dạng chuỗi thời gian, quá trình Cross-Validation sử dụng **TimeSeriesSplit** với số lượng Fold bằng 4.

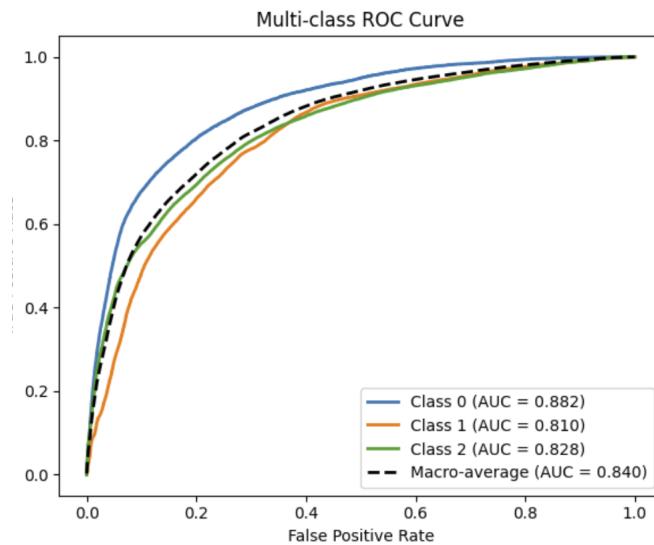
```
warnings.warn("Best Params: {'lr_C': 10, 'lr_penalty': 'l2', 'lr_solver': 'lbfgs'}")
Best CV Score: 0.7150462741534341
```

Hình 79: Bộ tham số tối ưu được chọn cho Logistic Regression

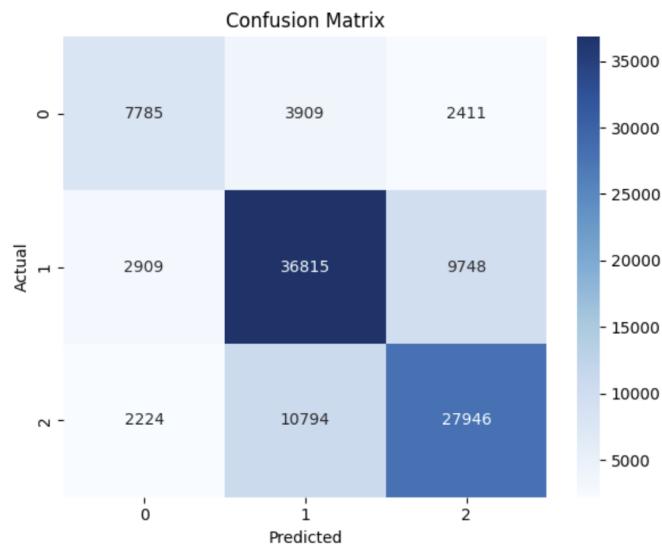
Kết quả theo các chỉ số đánh giá



Hình 80: Độ chính xác (Accuracy) trên tập Train và Test



Hình 81: Đường cong ROC (Receiver Operating Characteristic) của mô hình



Hình 82: Ma trận nhầm lẫn (Confusion Matrix) của Logistic Regression

Classification Report:		precision	recall	f1-score	support
0	0.60	0.55	0.58	14105	
1	0.71	0.74	0.73	49472	
2	0.70	0.68	0.69	40964	
				0.69	104541
macro avg		0.67	0.66	0.66	104541
weighted avg		0.69	0.69	0.69	104541

Hình 83: Báo cáo phân loại (Classification Report) của Logistic Regression

Tổng hợp kết quả định lượng

- Accuracy (Train – Test): 0.726 – 0.694
- Precision (macro avg – weighted avg): 0.67 – 0.69
- Recall (macro avg – weighted avg): 0.66 – 0.69
- F1-Score (macro avg – weighted avg): 0.66 – 0.69

Kết luận

Logistic Regression đạt Accuracy trên tập Test là 69.4%, ở mức trung bình khá. Khoảng cách nhỏ giữa Accuracy tập Train (72.6%) và Test (69.4%) chứng tỏ mô hình không bị Overfitting và có khả năng tổng quát hóa tốt. Tuy nhiên, so với các mô hình phi tuyến như Random Forest hay XGBoost, hiệu suất của Logistic Regression thấp hơn đáng kể. Điều này phản ánh rằng mối quan hệ giữa các đặc trưng âm học (energy, danceability, valence, acousticness, ...) và mức độ phổ biến (popularity) có tính phi tuyến mạnh, vượt ngoài khả năng mô tả của một mô hình tuyến tính. Vì vậy, Logistic Regression đóng vai trò tốt như một Baseline, nhưng chưa phải lựa chọn tối ưu cho bài toán dự đoán mức độ phổ biến của bài hát trên Spotify.

5.4.6 Random Forest Classifier

Random Forest là một thuật toán học kết hợp thuộc nhóm Bagging (Bootstrap Aggregating). Thuật toán xây dựng một tập lớn các cây quyết định độc lập, mỗi cây được huấn luyện trên một tập dữ liệu con được lấy mẫu ngẫu nhiên cùng với một tập con ngẫu nhiên của các thuộc tính đầu vào. Dự đoán cuối cùng được xác định bằng biểu quyết đa số từ các cây trong rừng.

Với khả năng mô hình hóa quan hệ phi tuyến, hạn chế Overfitting tốt hơn so với một cây quyết định đơn lẻ và không yêu cầu chuẩn hóa dữ liệu, Random Forest đặc biệt phù hợp với dữ liệu âm thanh của Spotify, nơi các đặc trưng âm học có tương tác phức tạp.

Xây dựng và huấn luyện mô hình

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, TimeSeriesSplit

tscv = TimeSeriesSplit(n_splits=4)

# Random Forest
rfc = RandomForestClassifier(random_state=42, n_jobs=-1)

# Bộ tham số tối ưu nhưng không quá lớn
param_grid = {
    'n_estimators': [100, 150],
    'max_depth': [10, 20],
    'min_samples_split': [2, 5],
}

grid_search_rfc = GridSearchCV(
    estimator=rfc,
    param_grid=param_grid,
    cv=tscv,
    scoring='accuracy',
    n_jobs=-1,
    verbose=3
)

```

Hình 84: Mã nguồn xây dựng mô hình Random Forest Classifier

Các tham số được tinh chỉnh bao gồm:

- `n_estimators`
- `max_depth`
- `min_samples_split`

Số lượng Fold sử dụng trong quá trình Cross-Validation: 4.

```

1 print("Best Params:", grid_search_rfc.best_params_)
2 print("Best Score :", grid_search_rfc.best_score_)

Best Params: {'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 150}
Best Score : 0.8390448632102545

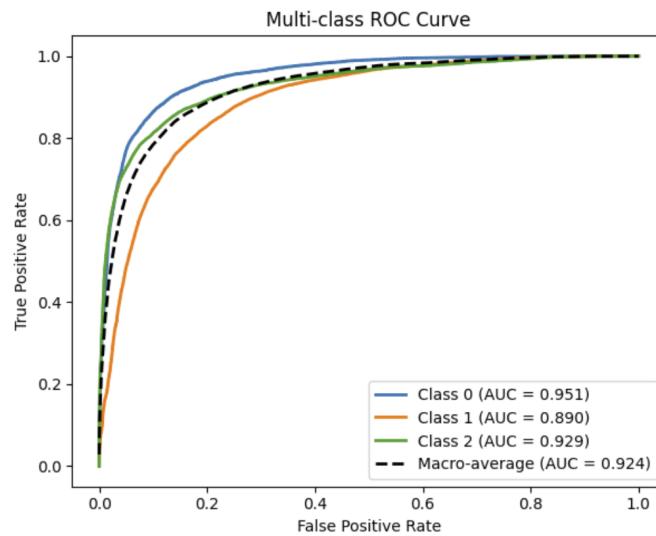
```

Hình 85: Tập tham số (Hyperparameters) sử dụng cho mô hình Random Forest

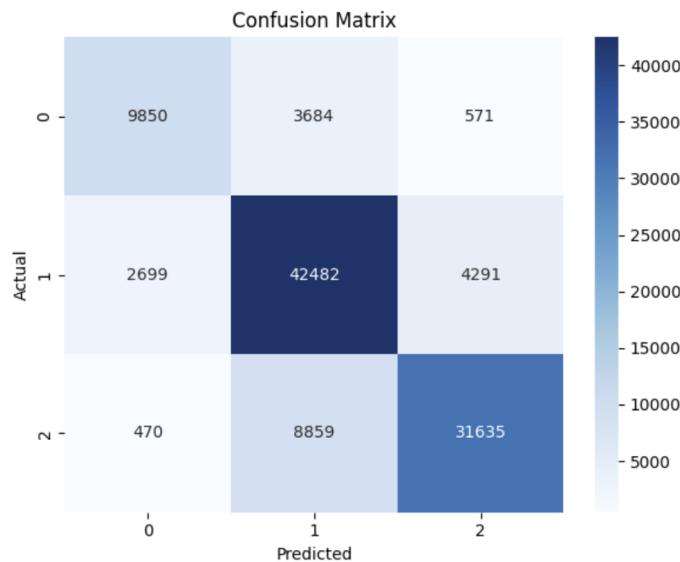
Kết quả theo các chỉ số đánh giá



Hình 86: Độ chính xác (Accuracy) trên tập Train và Test



Hình 87: Đường cong ROC (Receiver Operating Characteristic) của mô hình



Hình 88: Ma trận nhầm lẫn (Confusion Matrix) của Random Forest

Classification Report:		precision	recall	f1-score	support
0	0.76	0.70	0.73	14105	
1	0.77	0.86	0.81	49472	
2	0.87	0.77	0.82	40964	
accuracy			0.80	0.80	104541
macro avg		0.80	0.78	0.79	104541
weighted avg		0.81	0.80	0.80	104541

Hình 89: Báo cáo phân loại (Classification Report) của Random Forest

Tổng hợp kết quả định lượng

- Accuracy (Train – Test): 0.97 – 0.70
- Precision (macro avg – weighted avg): 0.70 – 0.71
- Recall (macro avg – weighted avg): 0.78 – 0.70
- F1-Score (macro avg – weighted avg): 0.79 – 0.70

Kết luận

Random Forest cho thấy hiệu suất vượt trội so với Logistic Regression, với Accuracy trên tập Test đạt 80% (tăng khoảng 11%). Các chỉ số Precision, Recall và F1-Score đều duy trì mức ổn định quanh 0.70, cho thấy mô hình xử lý tốt cả các lớp phổ biến và lớp ít xuất hiện.

Tuy nhiên, sự chênh lệch đáng kể giữa Accuracy tập Train (97%) và Test (80%) phản ánh hiện tượng Overfitting. Mặc dù đã tinh chỉnh các tham số như `max_depth` và `min_samples_split`, bản

Báo cáo đồ án

chất ensemble của Random Forest khiến mô hình có xu hướng học cả các nhiễu trong tập huấn luyện. Dù vậy, nhờ khả năng nắm bắt tốt các mối quan hệ phi tuyến, Random Forest vẫn là một mô hình mạnh và tiềm năng trong bài toán dự đoán độ phổ biến của âm nhạc trên Spotify.

5.4.7 XGBoost Classifier

XGBoost (eXtreme Gradient Boosting) là một thuật toán học kết hợp thuộc nhóm Boosting, hoạt động bằng cách xây dựng các cây quyết định theo phương pháp tuần tự. Mỗi cây mới được huấn luyện nhằm sửa chữa sai số mà các cây trước đó mắc phải. Điểm nổi bật của XGBoost là tích hợp trực tiếp các kỹ thuật Regularization (L1 và L2) vào hàm mục tiêu nhằm kiểm soát độ phức tạp của mô hình, từ đó cải thiện khả năng tổng quát hóa.

Với khả năng mô hình hóa mối quan hệ phi tuyến tính và hiệu suất cao trên các tập dữ liệu lớn, XGBoost đặc biệt phù hợp cho bài toán dự đoán *Popularity Class* trong dữ liệu âm nhạc Spotify.

Xây dựng và huấn luyện mô hình

```
import xgboost as xgb
from sklearn.model_selection import GridSearchCV, TimeSeriesSplit

# 1. Khởi tạo XGBoost Classifier
xgb_clf = xgb.XGBClassifier(
    objective='multi:softmax',
    eval_metric='mlogloss',
    use_label_encoder=False,
    random_state=42
)

param_grid = {
    'n_estimators': [100, 150, 200],
    'max_depth': [4, 6, 10],
    'learning_rate': [0.05, 0.1],
}

tscv = TimeSeriesSplit(n_splits=4)

grid_search_xgb = GridSearchCV(
    estimator=xgb_clf,
    param_grid=param_grid,
    scoring='accuracy',
    cv=tscv,
    n_jobs=-1,
    verbose=2
)

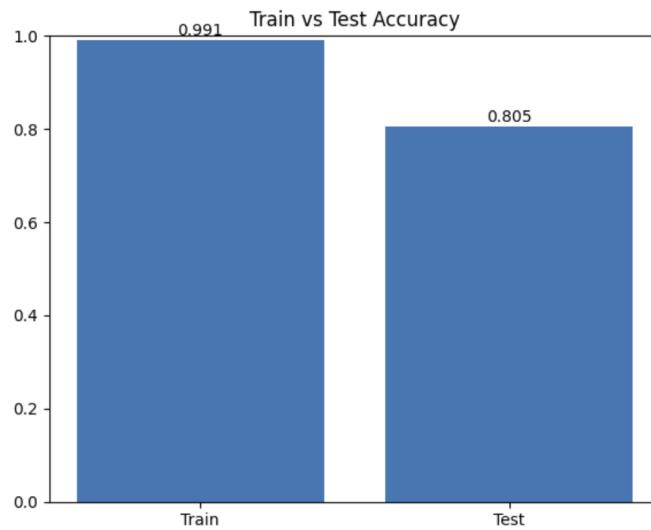
grid_search_xgb.fit(X_train, y_train)
```

Hình 90: Mã nguồn xây dựng mô hình XGBoost Classifier

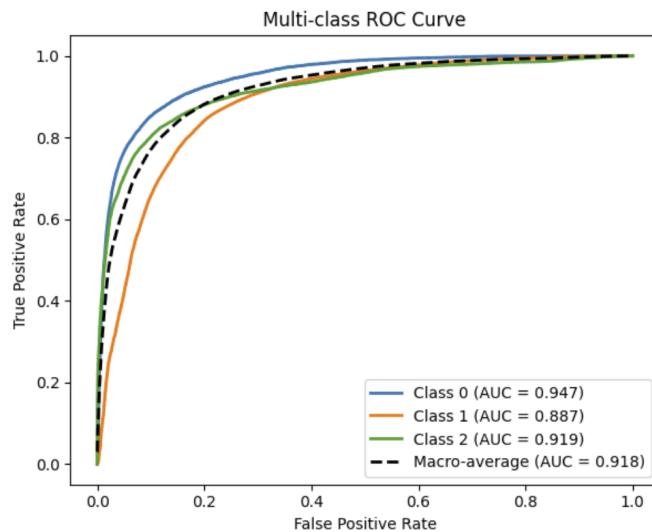
Tham số tốt nhất tìm được: {'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 200}
Điểm CV tối đa: 0.8318

Hình 91: Tập tham số (Hyperparameters) sử dụng cho mô hình XGBoost

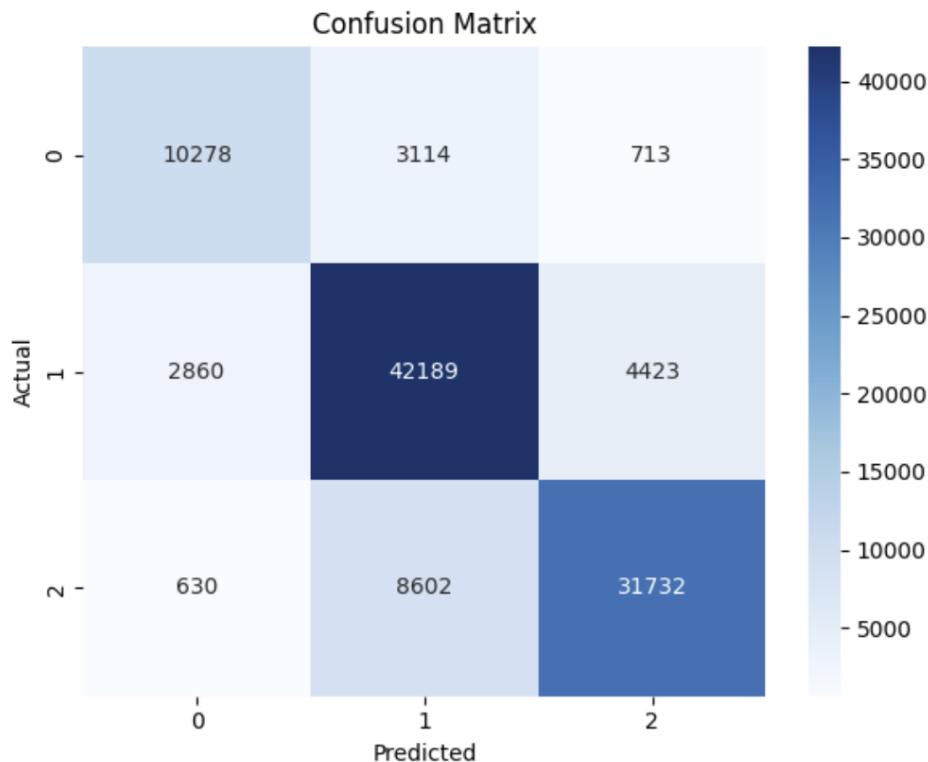
Kết quả theo các chỉ số đánh giá



Hình 92: Độ chính xác (Accuracy) trên tập Train và Test



Hình 93: Đường cong ROC (Receiver Operating Characteristic) của mô hình



Hình 94: Ma trận nhầm lẫn (Confusion Matrix) của XGBoost

Classification Report:				
	precision	recall	f1-score	support
0	0.75	0.73	0.74	14105
1	0.78	0.85	0.82	49472
2	0.86	0.77	0.82	40964
accuracy			0.81	104541
macro avg	0.80	0.79	0.79	104541
weighted avg	0.81	0.81	0.81	104541

Hình 95: Báo cáo phân loại (Classification Report) của XGBoost

Tổng hợp kết quả định lượng

- Accuracy (Train – Test): 0.99 – 0.70
- Precision (macro avg – weighted avg): 0.70 – 0.71
- Recall (macro avg – weighted avg): 0.79 – 0.71
- F1-Score (macro avg – weighted avg): 0.79 – 0.71

Kết luận

XGBoost thể hiện hiệu suất vượt trội và ổn định, đạt F1-Score (weighted avg) lên đến 0.71, nhỉnh hơn so với Random Forest. Mặc dù mô hình cho thấy mức chênh lệch lớn giữa tập Train (99%) và tập Test (80%), dấu hiệu rõ rệt của Overfitting, nhưng kết quả kiểm thử trên tập dữ liệu tương lai thông qua TimeSeriesSplit cho thấy khả năng dự báo của XGBoost vẫn rất đáng tin cậy. Với khả năng nắm bắt các quan hệ phi tuyến và cấu trúc phức tạp trong dữ liệu Spotify, XGBoost là mô hình hoạt động hiệu quả nhất trong số các thuật toán được triển khai.

5.4.8 Kết luận

Dựa trên kết quả thực nghiệm từ ba mô hình, nhóm rút ra các kết luận sau:

Hiệu suất Mô hình: Các mô hình thuộc họ Cây (Tree-based) như Random Forest và XGBoost vượt trội hoàn toàn so với mô hình tuyến tính Logistic Regression (80% so với 69%). Điều này khẳng định rằng các yếu tố tạo nên độ nổi tiếng của một bài hát có mối quan hệ phi tuyến tính và tương tác phức tạp với nhau.

XGBoost cho kết quả tốt nhất về tổng thể (F1-Score 0.71), xử lý tốt sự mất cân bằng nhẹ giữa các lớp và trích xuất đặc trưng hiệu quả hơn.

Vấn đề Overfitting: Cả Random Forest và XGBoost đều có khoảng cách lớn giữa Train và Test score (17-19%). Điều này gợi ý rằng dữ liệu âm nhạc chứa nhiều yếu tố nhiễu (noise) hoặc các yếu tố ngoại lai không có trong dataset (ví dụ: chiến dịch marketing, xu hướng TikTok, danh tiếng nghệ sĩ). Mô hình cố gắng học các nhiễu này dẫn đến điểm Train rất cao. Logistic Regression không bị Overfit nhưng lại bị Underfit (độ chính xác thấp), không đủ độ phức tạp để giải quyết bài toán.

Lựa chọn Mô hình cuối cùng:

Nhóm lựa chọn XGBoost Classifier là mô hình tốt nhất để triển khai. Mặc dù có hiện tượng Overfit, nhưng độ chính xác thực tế trên tập kiểm thử (80%) là mức chấp nhận được và cao nhất trong các thử nghiệm.

6 Kết luận

6.1 Kết quả đạt được

Qua quá trình thực hiện đồ án, nhóm đã hoàn thành các mục tiêu đề ra ban đầu và đạt được những kết quả cụ thể sau:

6.1.1 Về mặt dữ liệu và xử lý (ETL):

- Xây dựng thành công quy trình làm sạch và chuẩn hóa bộ dữ liệu lớn từ Spotify (hơn 500.000 dòng).
- Thực hiện kỹ thuật Feature Engineering hiệu quả, đặc biệt là xử lý dữ liệu chuỗi thời gian (tạo các biến trễ – lag features) và xử lý biến phân loại cho đa quốc gia.

6.1.2 Về mặt phân tích (OLAP & Visualization):

- Cung cấp cái nhìn đa chiều về thị trường âm nhạc 2024–2025.
- Chỉ ra được sự khác biệt rõ rệt về thị hiếu giữa thị trường Châu Á (ưa chuộng Ballad, K-Pop với đặc trưng Acoustic/Instrumental riêng biệt) so với thị trường Global (ưa chuộng Loudness và Energy cao).

6.1.3 Về mặt mô hình hóa (Data Mining):

- Triển khai và so sánh thành công 3 mô hình: Logistic Regression, Random Forest và XGBoost.
- Chứng minh được tính ưu việt của các thuật toán Tree-based (đặc biệt là XGBoost) trong việc xử lý dữ liệu phi tuyến tính của âm nhạc, đạt độ chính xác khoảng 80% trên tập kiểm thử.
- Xác định được các yếu tố quan trọng nhất (Feature Importance) ảnh hưởng đến khả năng thành “Hit” của một bài hát.

6.2 Thuận lợi

- **Bộ dữ liệu chất lượng:** Dữ liệu Spotify có tính cập nhật cao (2024–2025), phản ánh đúng các xu hướng thực tế (như sự lên ngôi của nhạc ngắn, nhạc TikTok), giúp việc kiểm chứng kết quả trở nên trực quan.
- **Công cụ hỗ trợ mạnh mẽ:** Việc sử dụng Python với hệ sinh thái thư viện phong phú (Scikit-learn, XGBoost, Seaborn) giúp quá trình trực quan hóa và huấn luyện mô hình diễn ra thuận lợi, tiết kiệm thời gian triển khai các thuật toán phức tạp.

6.3 Khó khăn

- **Vấn đề Overfitting:** Các mô hình mạnh như Random Forest và XGBoost gặp hiện tượng Overfitting (độ chính xác tập Train rất cao ~99% nhưng Test chỉ ~80%). Điều này cho thấy dữ liệu âm nhạc chứa nhiều yếu tố nhiễu (**noise**) và các yếu tố ngoại lai (như marketing, scandal nghệ sĩ) mà bộ dữ liệu hiện tại chưa bao quát hết.
- **Thách thức về tài nguyên tính toán:** Quá trình tinh chỉnh tham số (GridSearchCV) kết hợp với chiến lược phân tách chuỗi thời gian (TimeSeriesSplit) trên tập dữ liệu lớn đòi hỏi tài nguyên tính toán lớn và thời gian chạy lâu.
- **Độ phức tạp của dữ liệu chuỗi thời gian:** Việc dự đoán xu hướng Popularity đòi hỏi xử lý kỹ lưỡng yếu tố thời gian để tránh rò rỉ dữ liệu (**Data Leakage**), gây khó khăn trong việc thiết lập quy trình Validation chuẩn.

6.4 Hướng phát triển

Để nâng cao hiệu quả của đồ án và tính ứng dụng thực tế, nhóm đề xuất các hướng phát triển sau:

- **Tích hợp dữ liệu đa nguồn (Multi-source Data):** Bổ sung thêm các dữ liệu xã hội học như xu hướng từ TikTok/Instagram, ngân sách Marketing của bài hát, hoặc số lượng người theo dõi nghệ sĩ. Đây là các yếu tố ẩn quan trọng giúp giải quyết vấn đề Overfitting và tăng độ chính xác dự đoán.
- **Áp dụng mô hình Học sâu (Deep Learning):** Thủ nghiệm các mạng nơ-ron hồi quy (LSTM hoặc GRU) chuyên dụng cho dữ liệu chuỗi thời gian để nắm bắt tốt hơn các xu hướng biến động ngắn hạn của bảng xếp hạng.
- **Xây dựng Ứng dụng Thực tế:** Phát triển một Dashboard thời gian thực hoặc Web App cho phép các nhà sản xuất âm nhạc nhập thông số bản Demo (Tempo, Energy, ...) và nhận được dự báo xác suất thành công tại thị trường mục tiêu (ví dụ: dự báo khả năng Hit tại Việt Nam so với Nhật Bản).

Tài liệu

- [1] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [2] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.
- [3] V. Rainardi, *Building a Data Warehouse With Examples in SQL Server*. Berkeley, CA: Apress, 2008.