Relatório de Programação e Base de Dados
Mestrado em Ciência de Dados

Inês Silva Renatha Vieira

January 17, 2024

1 Introdução

2 Universo

O cenário selecionado para análise neste projeto é o ecossistema da aplicação Spotify, uma plataforma de streaming de música amplamente difundida globalmente. No Spotify, cada usuário desfruta de acesso a uma vasta coleção de músicas, organizadas em álbuns de diversos artistas. Tanto os álbuns quanto os artistas são categorizados por géneros musicais.

Foi também ponderada a popularidade dos artistas numa escala de valores entre 0 e 100, calculada por um algoritmo que considera, sobretudo, o número total de reproduções das faixas e a atualidade dessas reproduções. Em termos gerais, as músicas reproduzidas com frequência no momento tenderão a apresentar uma pontuação de popularidade mais elevada do que aquelas que foram reproduzidas num passado distante.

Os usuários têm a capacidade de seguir tanto artistas como outros utilizadores, adicionando uma dimensão colaborativa à experiência musical que é disponibilizada pelo Spotify. Esta interação social não só permite o partilhar de preferências musicais, mas também contribui para a descoberta de novas músicas.

O propósito é caracterizar e modelar a dinâmica que existe entre utilizadores, álbuns e artistas no contexto do Spotify, mediante isso os dados empregados neste estudo foram extraídos de uma lista de reprodução musical e adaptados para a elaboração da base de dados.

3 Modelo Entidade-Relacionamento (ER)

- Entidades:
  - TRACK(TrackId, TrackName, TracDuration, TrackPopularity);
  - ALBUM(AlbumId, AlbumName, TotalTracks, AlbumPopularity, AlbumReleaseDate);
  - ARTIST(ArtistId, ArtistName, ArtistPopularity);
  - GENRE(GenreId, GenreName);
  - USER(UserId, UserName, Locality(Country, State, City), Email?, NumTel);
- Atributos:
  - Opcional: Email na entidade USER.
  - Derivado: TotalTracks em ALBUM, que contabiliza o total de músicas pertencentes ao álbum.
  - Multi-valor: NumTel em USER que possibilita o usuário associar a sua conta mais de um número de telefone.
  - Composto: Locality em USER que agrega os atributos Country, State e City.
- Relacionamentos:
  - Uma música está exclusivamente associada a um álbum, podendo este conter de 1 a n músicas.
  - Uma música pode ser produzida por um ou mais artistas, sendo que um artista deve obrigatoriamente ter pelo menos uma música e pode ter inúmeras.

- Foi estabelecido que todas as músicas produzida por um artista pertencem a um álbum, garantindo assim que todos os artistas tenham de 1 a n álbuns..
- Cada álbum está vinculado a um único género, podendo um género abranger vários álbuns, mas sendo obrigatório ter pelo menos um álbum.
- Um utilizador pode optar por não seguir nenhum artista, assim como pode seguir todos os artistas na plataforma. Cada artista, por sua vez, pode ser seguido por 0 a n utilizadores
- Um utilizador pode seguir de 0 até n utilizadores, bem como adquirir de 0 a n seguidores.

• Restrições de relacionamentos:

Relacionamento	Restr. Cardinalidade	Restr. Participação	Restr. Estrutural
MADE BY(TRACK, ARTIST)	N:M	total/total	(1,N) >(1,M)
BELONGS TO(TRACK, ALBUM)	N:1	total/total	(1,N) >(1,1)
PRODUCED BY(ALBUM, ARTIST)	N:1	total/total	(1,N) >(1,1)
PERTAIN TO(ALBUM, GENRE)	N:1	total/total	(1,N) >(1,1)
FOLLOWS_ARTIST(USER, ARTIST)	N:M	parcial/parcial	(0,N) >(0,M)
FOLLOWS_USER(USER, USER)	N:M	parcial/parcial	(0,N) >(0,M)

Table 1: Restições de Relacionamento

Modelo ER

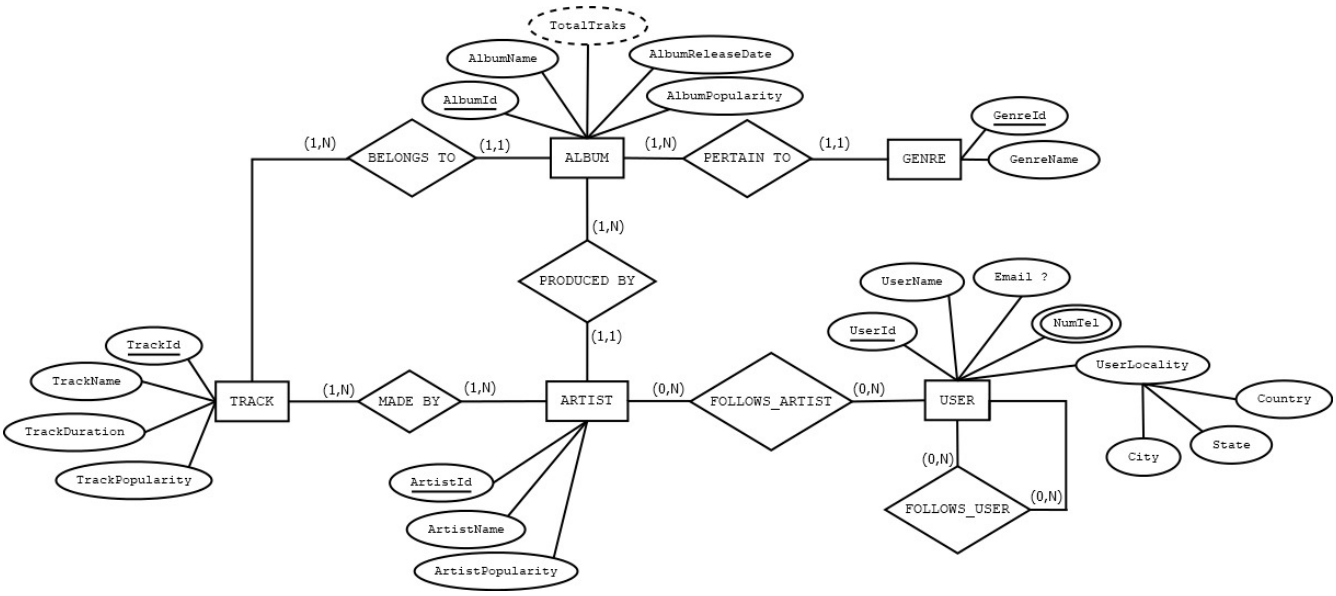

Figure 1: Modelo Entidade-Relacionamento da base de dados Spotify

4 Modelo Relacional

A conversão de um modelo Entidade-Relacionamento (ER) para um esquema SQL envolve a tradução das entidades, relacionamentos e atributos do modelo ER para tabelas, chaves primárias e estrangeiras no SQL.

Os relacionamentos BELONGS TO(TRACK, ALBUM), PRODUCED BY(ALBUM, ARTIST) e PERTAIN TO(ALBUM, GENRE) caracterizam-se por serem de vários para um (N,1) e possuem participação total para

ambas as entidades relacionadas. Neste sentido, foram atribuídas chaves estrangeiras nas entidades TRACK e ALBUM com o intuito de indicar a que álbum a faixa pertence, por qual artista o álbum foi produzido e em qual género o álbum se insere.

Por outro lado, MADE BY (TRACK, ARTIST), FOLLOWS_ARTIST (USER, ARTIST) e FOLLOWS_USER (USER, USER) representam relacionamentos de cardinalidade M:N. A representação destes relacionamentos exige a construção de tabelas de "referência cruzada". Estas tabelas incluem chaves que identificam a relação como chaves primárias, e ambas são designadas como chaves estrangeiras, indicando a relação entre as entidades.

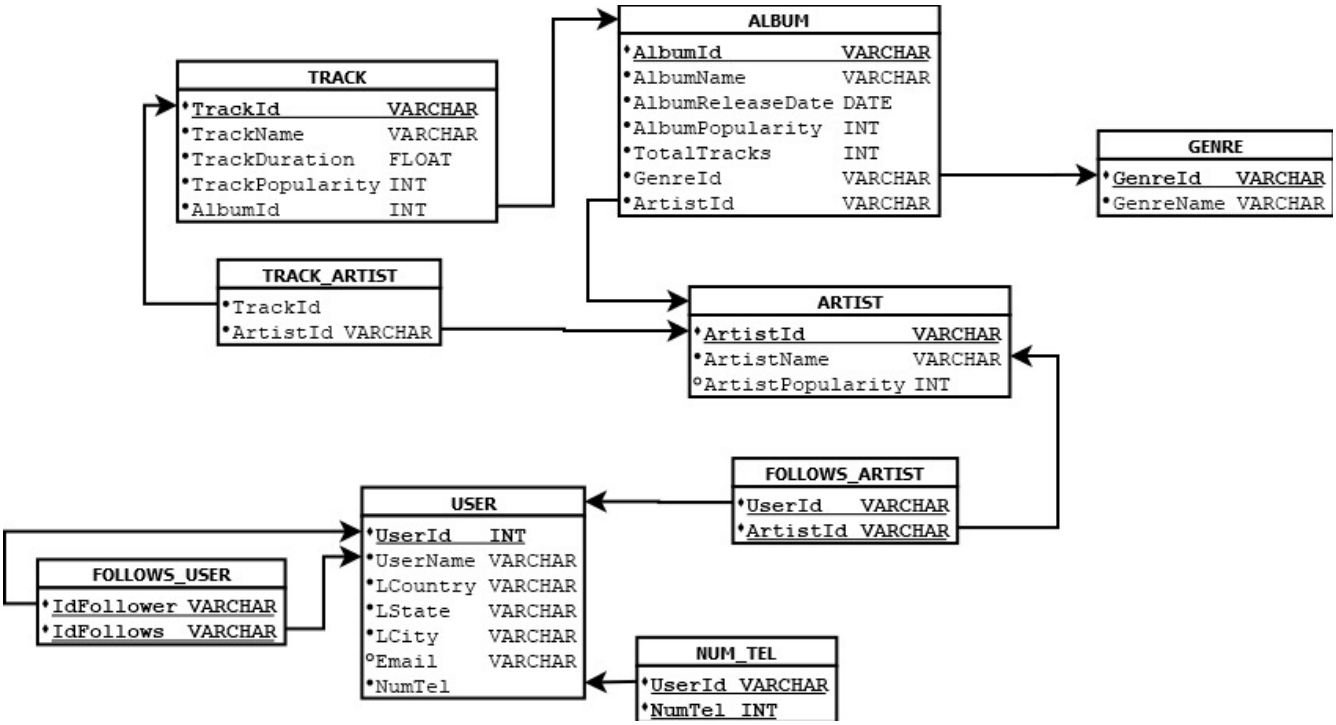

Figure 2: Modelo Relacional base de dados Spotify

5 Construção da Base de Dados

5.1 Python

Para a construção da base de dados, foram recolhidas informações de uma playlist do Spotify. Para esse fim, a playlist foi exportada como um ficheiro CSV, e procedeu-se à análise dos dados. Cada variável foi examinada detalhadamente a fim de identificar possíveis erros de formatação, tipos de dados incorretos, valores nulos e entradas duplicadas. As variáveis associadas ao dataset exportado estão listadas na imagem abaixo:

```
Data columns (total 22 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   addedAt                   1101 non-null   object
 1   addedBy                   1101 non-null   object
 2   albumArtistsNames         1101 non-null   object
 3   albumName                 1101 non-null   object
 4   albumPopularity           1101 non-null   int64
 5   albumRecordLabel          1101 non-null   object
 6   albumReleaseDate          1101 non-null   object
 7   albumType                 1101 non-null   object
 8   albumUpc                  1101 non-null   int64
 9   albumUrl                  1101 non-null   object
 10  artistFollowers           1101 non-null   int64
 11  artistGenres              1072 non-null   object
 12  artistName                1101 non-null   object
 13  artistPopularity          1101 non-null   int64
 14  artistUrl                 1101 non-null   object
 15  isLikedByUser             1101 non-null   bool
 16  trackDuration             1101 non-null   object
 17  trackFeatureAcousticness  1101 non-null   float64
 18  trackName                 1101 non-null   object
 19  trackNumber               1101 non-null   int64
 20  trackPopularity           1101 non-null   int64
 21  trackUrl                  1101 non-null   object
dtypes: bool(1), float64(1), int64(6), object(14)
```

Figure 3: Informações das variáveis refentes ao Dataset original

Considerando que as variáveis TrackId e AlbumId não estavam originalmente contidas no dataset, mas foram julgadas pertinentes para a incorporação no esquema da base de dados, optou-se por empregar a API disponibilizada pelo Spotify for Developers. Através do módulo Spotipy, realizou-se uma pesquisa utilizando o nome da faixa musical, o nome do artista, o nome do álbum e o nome do artista do álbum, para identificar o ID de cada música e o ID de cada álbum. Adicionalmente, aplicou-se o mesmo método para obter o ID do artista, utilizando o nome de cada artista para tal efeito. Dessa forma, possibilitou-se a inclusão dessas variáveis como chaves primárias nas tabelas correspondentes do banco de dados, uma vez que o ID representa o elemento ideal para distinguir e identificar as entidades. Numa fase subsequente, foram removidos todos os valores nulos e todas as entradas duplicadas, a fim de evitar conflitos durante a inserção no SQL. Através do módulo UUID, sigla para Universally Unique Identifier, foram geradas as colunas UserId, UserFollowers e UserFollowing, de modo a criar informações de relacionamento numa interface entre o Artista e o Utilizador. Este módulo, amplamente utilizado para identificar informações em sistemas de computação, é representado por 32 dígitos hexadecimais, exibidos em cinco grupos separados por hifens. Foram apenas consideradas as colunas de maior interesse.

5.1.1 Adaptação dos dados para uma Base de Dados

Numa etapa final, procedeu-se à criação de todas as tabelas conforme representadas no Modelo Relacional, com o objetivo de serem posteriormente exportadas como ficheiros CSV, visando assim simplificar o processo de introdução dos dados no SQL.

Face à extensão do dataset e à identificação de algumas violações de integridade associadas a determinadas chaves primárias, optou-se por extrair apenas uma amostra deste. Esta amostra foi, então, manualmente inserida na tabelas SQL correspondentes, juntamente com os respetivos valores. Esta abordagem foi adotada com o propósito de conferir maior maleabilidade aos dados, permitindo assim uma exemplificação mais eficaz de possíveis consultas(Queries).

6 Exemplos de Queries SQL

A elaboração das consultas SQL tem como propósito proporcionar uma visão abrangente e organizada da base de dados musical, seguindo uma abordagem formal. A primeira consulta, ao criar uma tabela que lista todos os géneros associados a cada artista, oferece um panorama claro dos estilos presentes no seu repertório. Já a segunda consulta, ao apresentar uma tabela de músicas ordenadas por popularidade, acompanhadas das informações sobre artistas, álbuns e géneros, procura proporcionar uma experiência analítica ao utilizador, permitindo a identificação

das músicas mais populares e a contextualização destas nos seus respetivos álbuns e géneros musicais. A terceira consulta visa quantificar o número de seguidores de cada artista, fornecendo insights sobre a popularidade e o impacto das suas obras no público.

Para além das consultas de análise, o processo de expansão e atualização da base de dados foi abordado através de queries de inserção. As queries INSERT INTO utilizadas para adicionar novos géneros, artistas e músicas demonstram a flexibilidade do sistema em lidar com mudanças e atualizações constantes na indústria musical. Ao seguir as relações predefinidas entre tabelas, como exemplificado nas queries de inserção, garantimos a integridade e consistência dos dados. Foi elaborada uma outra query exploratória para identificar os utilizadores que seguem um artista específico e apresentar o país de origem desses seguidores. Esta análise contribui para uma compreensão mais aprofundada do alcance geográfico e do público de determinado artista, fornecendo insights valiosos para estratégias de marketing e promoção. É de salientar que os exemplos de queries apresentados são apenas ilustrativos e representam possíveis abordagens na execução das consultas em questão.

7 Conclusão

Este projeto representou uma imersão profunda no cenário musical da plataforma Spotify, com o propósito de modelar e compreender as complexas interações entre utilizadores, álbuns e artistas. A escolha do Spotify como objeto de estudo possibilitou uma análise aprofundada da dinâmica social e musical presente nesta plataforma global de streaming.

A utilização da API do Spotify, juntamente com a extração e adaptação meticulosa dos dados provenientes de uma playlist, destacou a importância crucial da manipulação cuidadosa que é necessário ter quando se lida com este tipo de dados. A opção pela inserção manual de apenas uma parte reduzida, representativa de todo o dataset, permitiu uma abordagem mais flexível e eficaz na demonstração de consultas SQL, enfatizando a necessidade essencial de adaptação e refinamento dos dados de modo a proporcionar uma maior robustez aos dados. A etapa de desenvolvimento dos modelos lógicos não apenasnfacilitou a tradução de conceitos complexos em entidades, atributos e relacionamentos, mas também serviu como um guia sólido para o processo subsequente de construção da base de dados. A importância da criação destes modelos tornou-se evidente ao longo do projeto devido ao facto de providenciarem uma clara estrutura conceitual para representar as interações entre os diferentes atributos.

Os desafios enfrentados neste projeto desempenharam um papel fundamental na obtenção de uma compreensão mais profunda e abrangente do processo envolvido no desenvolvimento de uma base de dados. Permitiu a obtenção de uma visão mais ampla sobre as complexidades da modelagem de dados e as nuances envolvidas na construção de uma base sólida para análise musical no contexto do mundo Spotify.

8 Referências

Spotify. (2024). Spotify for Developers. Recuperado de https://developer.spotify.com/documentation/web-api