



Bases de Dados

Grupo: 203

Elementos do grupo:

- Bruno Fernandes – up200707284
- Igor Silveira – up201505172
- João Mendes – up201505439

Docente:

Carla Teixeira Lopes

Índice

Introdução.....	3
Descrição do contexto.....	4
Diagrama UML	6
Esquema Relacional	7
Análise das Formas Normais	9
Restrições	13

Introdução

O objetivo do trabalho é construir um modelo conceptual para modelar informação de uma aplicação baseada no Spotify:

Os clientes têm um título, nome, morada, código postal, telefone e país.

As contas estão associadas a um tipo de conta.

As contas podem ser grátis (publicidade durante o uso da aplicação), premium (custo mensal de 6.99€, com download offline; sem anúncios e escolha de música na playlist/álbum; saltar música; melhor qualidade de som. Pagamento: cartão de crédito, Paypal e Paysafecard).

Cada artista tem, Albums, Músicas mais populares, Artistas relacionados e Bibliografia.

As músicas têm nome, autor(es), ano e respetivo álbum.

Os álbuns guardam um conjunto de músicas.

Descrição do contexto



O Spotify tem **utilizadores**, onde cada utilizador regista-se com username, email, data de nascimento, país, telefone e por fim, a palavra-passe. Cada utilizador tem uma **subscrição** que no início é free, mas podendo mudar para premium se efetuar um pagamento mensal, pagamento este que envolve 3 formas: cartão de crédito, paypal e paysafecard.

A subscrição premium dá oportunidade de se usufruir de download para poder ouvir música offline, sem anúncios e escolha de música na playlist/álbum, “zapping” e melhoria na qualidade de som.

Quando a subscrição é grátis, a **publicidade** ouvida reflete-se no **país** onde o utilizador se encontra no momento. Cada publicidade tem um id, uma duração e um **publicitário** associado com nome e investimento.

Cada conta está dividida para pesquisa em **artistas**, **playlists** próprias ou de outros utilizadores, géneros e recomendações de músicas e álbuns com base na música ouvida.

Cada utilizador pode escolher como favorito uma **música** individual ou então um **álbum** englobando todas as suas músicas e seguir um dado artista, assim como seguir outros utilizadores.

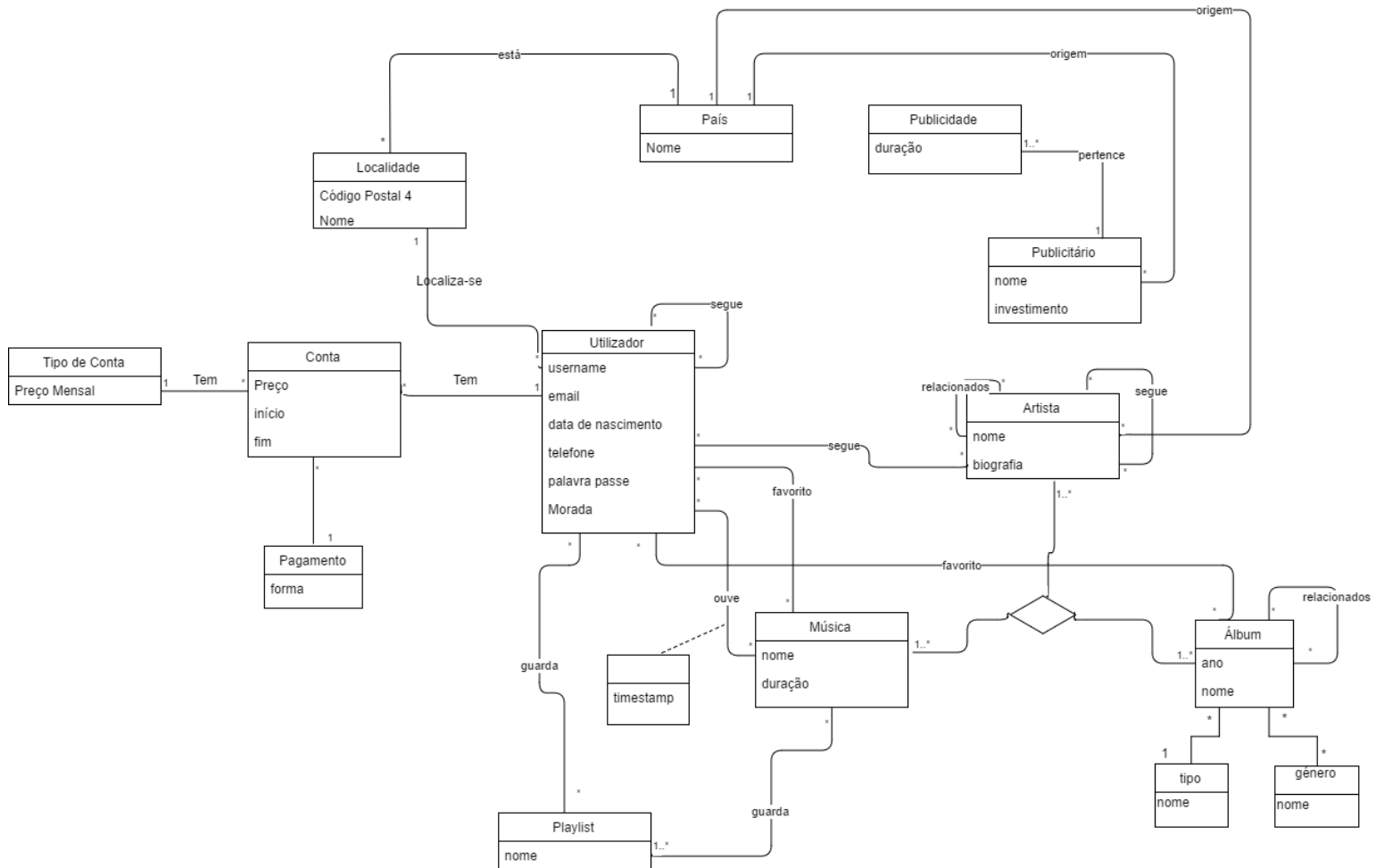
Cada artista tem álbuns em que cada um destes inclui as faixas de música, artistas relacionados, biografia, número de seguidores e número de utilizadores e/ou outros artistas que o artista está a seguir e playlists onde se encontram incluídos.

Cada álbum está dividido em diversos tipos (estúdio, ao vivo, compilação, best of's, soundtracks), tem um ou mais géneros musicais (rock, pop, hip hop, country, metal, clássico, etc.), nome do álbum, ano de lançamento e várias faixas de música.

Cada música tem uma duração, número de vezes ouvida e pertencem a um ou mais álbuns.

As playlists podem ser criadas por utilizadores e serem constituídas por diversos tipos de música diferente tendo assim um nome associado escolhido pelo criador. Esse nome pode ser repetido por esse e por outros utilizadores.

Modelo Concetual - Diagrama UML



Bruno Fernandes

Igor Silveira

João Mendes

Esquema Relacional

Utilizador(username, email, data de nascimento, telefone, password, morada)

UtilizadorSegueUtilizador(username1->Utilizador, username2->Utilizador)

UtilizadorLocaliza-se(username->Utilizador, CPGeral->Localidade)

UtilizadorGuardaPlaylist(username->Utilizador, PLID->Playlist)

UtilizadorOuveMúsica(username->Utilizador, MuID->Música, Time_Stamp)

UtilizadorFavoritaMúsica(username->Utilizador, MuID ->Música)

UtilizadorFavoritaÁlbum(Username->Utilizador, AbID->Álbum)

UtilizadorSegueArtista(Username->Utilizador, AID->Artista)

Conta(ContaID, Preço, DataInício, DataFim, Utilizador->Utilizador, , Pagamento->Pagamento, TipoConta->TipoConta)

Pagamento(forma)

TipoConta(PreçoMensual)

Localidade(CPGeral, Nome, PaísID->País)

Playlist(PLID, Nome)

PlaylistGuardaMúsica(PLID->Playlist, MuID ->Música)

Música(MuID, Nome, Duração)

Álbum(AbID, Nome, Ano, TipoID->Tipo)

ÁlbumRelacionaÁlbum(AbDI1->Álbum, AbID2->Álbum)

Bruno Fernandes

Igor Silveira

João Mendes

ÁlbumGênero(AbDI->Álbum, GenID->Gênero)

Artista(AID, Nome, Biografia, PaísID->País)

ArtistaSegueArtista(AID1->Artista1, AID2->Artista)

ArtistaRelacionaArtista(AID1->Artista, AID2->Artista)

MúsicaÁlbumArtista(MuID->Música, AbID->Álbum, AID->Artista)

Publicitário(PID, Nome, Investimento, PaísID->País)

Publicidade(PbID, Duração, PID->Publicitário)

País(PaísID, Nome)

Tipo(TipoID, Nome)

Gênero(GenID, Nome)

Análise das Formas Normais (1st, 2nd, 3rd, Boyce-Codd)

Nenhuma das relações viola a 1ª forma normal, pois em todas, todos os seus atributos são atômicos.

Também não existem violações à 2ª forma normal porque cada atributo não-chave é dependente da chave inteira.

Utilizador(username, email, data de nascimento, telefone, password, morada)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Username é super chave.

UtilizadorSegueUtilizador(username1->Utilizador, username2->Utilizador)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

UtilizadorLocaliza-se(username->Utilizador, CPGeral->Localidade)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

UtilizadorGuardaPlaylist(username->Utilizador, PLID->Playlist)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

UtilizadorOuveMúsica(username->Utilizador, MuID->Música, Time_Stamp)

- Exceção. Viola a 2ª Forma normal, o atributo Time_Stamp não tem dependência funcional de chave primária. Consequentemente viola a 3ª forma normal e forma normal de Boyce-Codd.

UtilizadorFavoritaMúsica(username->Utilizador, MuID->Música)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

UtilizadorFavoritaÁlbum(Username->Utilizador, AbID->Álbum)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

UtilizadorSegueArtista(Username->Utilizador, AID->Artista)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Conta(ContaID, Preço, DataInício, DataFim, Utilizador->Utilizador, Pagamento->Pagamento, TipoConta->TipoConta)

- 3ª Forma normal: Viola. Cada atributo não-chave não é completamente independente dos outros atributos não-chave. Consequentemente viola a BCNF.

Pagamento(forma)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais.

TipoConta(PreçoMensal)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais.

Localidade(CPGeral, Nome, PaísID->País)

- 3ª Forma normal: Viola, existe dependência transitiva. Consequentemente viola BCNF.

Playlist(PLID, Nome)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais.

PlaylistGuardaMúsica(PLID->Playlist, MuID->Música)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Música(MuID, Nome, Duração)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância em dependências funcionais.

Álbum(AbID, Nome, Ano, TipoID->Tipo)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais.

ÁlbumRelacionaÁlbum(AbDI1->Álbum, AbID2->Álbum)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

ÁlbumGênero(AbDI->Álbum, GenID->Gênero)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Artista(AID, Nome, Biografia, PaísID->País)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais.

ArtistaSegueArtista(AID1->Artista1, AID2->Artista)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

ArtistaRelacionaArtista(AID1->Artista, AID2->Artista)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Música(AlbumID, ArtistaID, MusicaID->Música, AlbumID->Álbum, ArtistaID->Artista)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Publicitário(PID, Nome, Investimento, PaísID->País)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais.

Publicidade(PbID, Duração, PID->Publicitário)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais.

País(PaísID, Nome)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Tipo(TipoID, Nome)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Gênero(GenID, Nome)

- 3ª Forma normal: Não viola, não existe dependência transitiva.
- BC Normal Form: Não viola, não existe redundância nas dependências funcionais. Chaves no lado esquerdo.

Restrições

Utilizador(username, email, data de nascimento, telefone, password, morada)

-Username é a chave primária da relação (PRIMARY KEY) que não aceita valores nulos e repetidos, pois não deve haver dois Utilizadores com o mesmo UserName.

-Email deve ser único (UNIQUE) e obrigatoriamente preenchido (NOT NULL)

-DataNascimento e password deve ser obrigatórias (NOT NULL)

-Caso não seja preenchido o campo telefone deve tomar o valor "N/D" (DEFAULT telefone="N/D"), este campo é único por Utilizador (UNIQUE)

-Caso não seja preenchido o campo morada deve tomar o valor "N/D" (DEFAULT morada="N/D")

UtilizadorSegueUtilizador(username1->Utilizador, username2->Utilizador)

-username1 e username2 são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia aos mesmos Utilizadores. Fazem também referencia às chaves primárias da tabela Utilizador (FOREIGN KEY)

UtilizadorLocaliza-se(username->Utilizador, CPGeral->Localidade)

-username e CPGeral são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Utilizador e Localidade. Fazem também referencia às chaves primárias das tabelas Utilizador e Localidade (FOREIGN KEY).

UtilizadorGuardaPlaylist(username->Utilizador, PLID->Playlist)

-username e PLID são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Utilizador e Playlist. Fazem também referencia às chaves primárias das tabelas Utilizador e Playlist (FOREIGN KEY).

UtilizadorOuveMúsica(username->Utilizador, MuID->Música, Time_Stamp)

-username e MuID são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Utilizador e Musica. Fazem também referencia às chaves primárias das tabelas Utilizador e Musica (FOREIGN KEY).

-Time_Stamp nunca tomará valores nulos (NOT NULL) e será sempre distinto para cada tuplo entre o mesmo username e MID;

UtilizadorFavoritaMúsica(username->Utilizador, MuID->Música)

-username e MuID são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Utilizador e Musica. Fazem também referencia às chaves primárias das tabelas Utilizador e Musica (FOREIGN KEY).

UtilizadorFavoritaÁlbum(Username->Utilizador, AbID->Álbum)

-username e AbID são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Utilizador e Álbum. Fazem também referencia às chaves primárias das tabelas Utilizador e Álbum (FOREIGN KEY).

UtilizadorSegueArtista(Username->Utilizador, AID->Artista)

-username e AID são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Utilizador e Artista. Fazem também referencia às chaves primárias das tabelas Utilizador e Artista (FOREIGN KEY).

Conta(ContaID, Preço, DataInício, DataFim, Utilizador->Utilizador, Pagamento->Pagamento, TipoConta->TipoConta)

-CID é a chave primária da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e que não aceita valores nulos nem repetidos, pois não devem haver contas com o mesmo CID;

-Preço deve ter valores não negativos (CHECK Preço>=0), e necessariamente não nulo (NOT NULL)

-DataInício deve ser menor que DataFim (CHECK DataInício<DataFim)

-Utilizador faz uma referência a uma chave primaria da tabela Utilizador (FOREIGN KEY) e deve haver sempre um cliente relacionado com a Conta (NOT NULL)

-Pagamento faz uma referência a uma chave primária da tabela Pagamento (FOREIGN KEY) e deve haver sempre um tipo de pagamento relacionado com a Conta (NOT NULL) e que tome um dos seguintes valores “gratuito”, “CartãoCredito”, “Paypal” ou “PaysafeCard” (CHECK Pagamento in (‘gratuito’, ‘CartãoCrédito’, ‘Paypal’, ‘PaysafeCard’))

-TipoConta faz uma referência a uma chave primaria da tabela TipoConta (FOREIGN KEY) e deve haver sempre um tipo de conta associado à conta (NOT NULL) que deve tomar um dos seguintes valores “gratuita”, “premium” e “família” (CHECK TipoConta in (‘gratuita’, ‘premium’, ‘família’))

Pagamento(forma)

-Forma é chave primária da classe (PRIMARY KEY) e toma obrigatoriamente um dos seguintes valores “gratuito”, “CartãoCrédito”, “Paypal” ou “PaysafeCard” (CHECK Forma in (‘gratuito’, ‘CartãoCrédito’, ‘Paypal’, ‘PaysafeCard’))

TipoConta(PreçoMensal)

-PreçoMensal é chave primária da classe (PRIMARY KEY) e toma obrigatoriamente um dos seguintes valores “0.0”, “6.99” ou “10.99” (CHECK PreçoMensal in (‘0.0’, ‘6.99’, ‘10.99’))

Localidade(CPGeral, Nome, PaísID->País)

-CPGeral é chave primária da relação (PRIMARY KEY) e não aceita valores nulos nem repetidos, pois cada localidade tem o seu código postal respetivo.

-Nome não aceita valores nulos (NOT NULL)

-País faz referência a uma chave primária da tabela País (FOREIGN KEY) e deve haver sempre um país associado a uma localidade (NOT NULL)

Playlist(PLID, Nome)

-PLID é chave primária da relação (PRIMARY KEY) que é auto incrementada com a criação de mais playlists na base de dados e não aceita valores nulos nem repetidos, pois cada playlist é distinta de todas as outras dentro da base de dados.

-Caso Nome não seja especificado o campo será preenchido com o ID interno da playlist (DEFAULT Nome='PLID')

PlaylistGuardaMúsica(PLID->Playlist, MuID->Música)

-PLID e MuID são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referência à mesma Playlist e Música. Fazem também referência às chaves primárias das tabelas Playlist e Música (FOREIGN KEY).

Música(MuID, Nome, Duração)

- MuID é chave primária da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceita valores nulos nem repetidos, pois cada música é distinta de todas as outras dentro da base de dados.

-Nome não aceita valores nulos (NOT NULL)

-Duração em segundos não aceita valores nulos (NOT NULL) nem não positivos (CHECK Duração>0)

Álbum(AbID, Nome, Ano, TipoID->Tipo)

-AbID é chave primária da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceita valores nulos nem repetidos, pois cada Álbum tem o seu numero de identificação interno.

-Nome não aceita valores nulos (NOT NULL)

-Ano não aceita valores nulos (NOT NULL) nem não inferiores a 1900 (CHECK Ano>1900)

-TipoID faz referência a uma chave primária da tabela Tipo (FOREIGN KEY) e deve haver sempre um Tipo associado a um Álbum (NOT NULL)

ÁlbumRelacionaÁlbum(AbID1->Álbum, AbID2->Álbum)

-AbID1 e AbID2 são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia aos mesmos Álbuns. Fazem também referencia às chaves primárias da tabela Álbum (FOREIGN KEY).

ÁlbumGênero(AbID->Álbum, GenID->Gênero)

-AbID e GenID chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Álbum e Gênero. Fazem também referencia às chaves primárias das tabelas Álbum e Gênero (FOREIGN KEY).

Artista(AID, Nome, Biografia, PaísID->País)

-AID é chave primária da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceita valores nulos nem repetidos, pois cada artista tem o seu código interno único.

-Nome não aceita valores nulos (NOT NULL)

- caso de não ser preenchida a Biografia deverá apresentar o valor “Sem Informação” (DEFAULT Biografia=”Sem Informação”)

-País faz referência a uma chave primária da tabela País (FOREIGN KEY) e deve haver sempre um país associado a um artista para fins de sugestões (NOT NULL)

ArtistaSegueArtista(AID1->Artista1, AID2->Artista)

-AID1 e AID2 chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia aos mesmos Artistas. Fazem também referencia às chaves primárias da tabela Artista (FOREIGN KEY).

ArtistaRelacionaArtista(AID1->Artista, AID2->Artista)

-AID1 e AID2 chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia aos mesmos Artistas. Fazem também referencia às chaves primárias da tabela Artista (FOREIGN KEY).

MúsicaÁlbumArtista(MuID->Música, AbID->Álbum, AID->Artista)

-MID, AbID e AID são chaves primárias (PRIMARY KEY) não aceitando valores nulos nem repetidos, logo não deve haver mais do que apenas um tuplo que faça referencia ao mesmo Artista, Álbum e Música. Fazem também referencia às chaves primárias das tabelas Artista, Álbum e Música (FOREIGN KEY).

Publicitário(PID, Nome, Investimento, PaísID->País)

-PID é chave primária da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceita valores nulos nem repetidos, pois cada publicitário tem o seu numero de identificação na base de dados
-Nome não aceita valores nulos (NOT NULL)
-Investimento não toma valores nulos (NOT NULL) nem negativos (CHECK Investimento >0)
-País faz referência a uma chave primária da tabela País (FOREIGN KEY) e deve haver sempre um país associado a um publicitário (NOT NULL)

Publicidade(PbID, Duração, PID->Publicitário)

-PbID é a chave primária da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceita valores nulos nem repetidos pois não poderá existir duas publicidades com identificações iguais internamente
-Duração não pode tomar valores nulos (NOT NULL) nem valores não positivos (CHECK Duração>0)
-PID faz referência a uma chave primária da tabela Publicitário (FOREIGN KEY) e deve haver sempre um Publicitário associado a uma Publicidade (NOT NULL)

País(PaísID, Nome)

-PaísID e Nome são as chaves primárias da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceitam valores nulos nem repetidos pois não existem dois países que sejam o mesmo.

Bruno Fernandes

Igor Silveira

João Mendes

Tipo(TipoID, Nome)

-TipoID e Nome são as chaves primárias da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceitam valores nulos nem repetidos pois não existem dois tipos que sejam o mesmo.

Género(GenID, Nome)

-GenID e Nome são as chaves primárias da relação (PRIMARY KEY) que é auto incrementada com a criação de mais contas na base de dados e não aceitam valores nulos nem repetidos pois não existem dois géneros que sejam o mesmo.