U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Bases de Dados

Plataforma Spotify

António Cruz, Joana Ramos e João Malheiro

MIEICT1G3 Bases de Dados, 2017/2018

Descrição

Este projeto retrata um serviço de streaming de música, à semelhança do Spotify.

Para usufruírem da plataforma todos os **utilizadores** têm que se registar com um nome de utilizador (*username*), email (que não se pode repetir) e também com o seu nome, **país** de origem e uma foto (opcional). É atribuído ao utilizador um ID. Todos os utilizadores podem guardar (porque são, à partida, imutáveis) **músicas** e **álbuns** na sua conta e seguir (acompanhar a atividade) **playlists**, **artistas** e outros utilizadores. Interessa saber qual a música que um dado utilizador está a ouvir no momento, e de todas as músicas que ouviu é relevante saber a data (incluindo o instante temporal) e se a passou à frente.

Um **artista** (referido no singular podendo ser na verdade um conjunto de artistas, tais como uma banda, que partilham apenas uma conta profissional), tem um ID, nome artístico, uma biografia, país de origem, **género**(s) e pelo menos uma música.

Uma **música** é representada pelo seu nome, ID, duração, género(s), artista(s) que a interpretam e pelo álbum/álbuns em que se inclui.

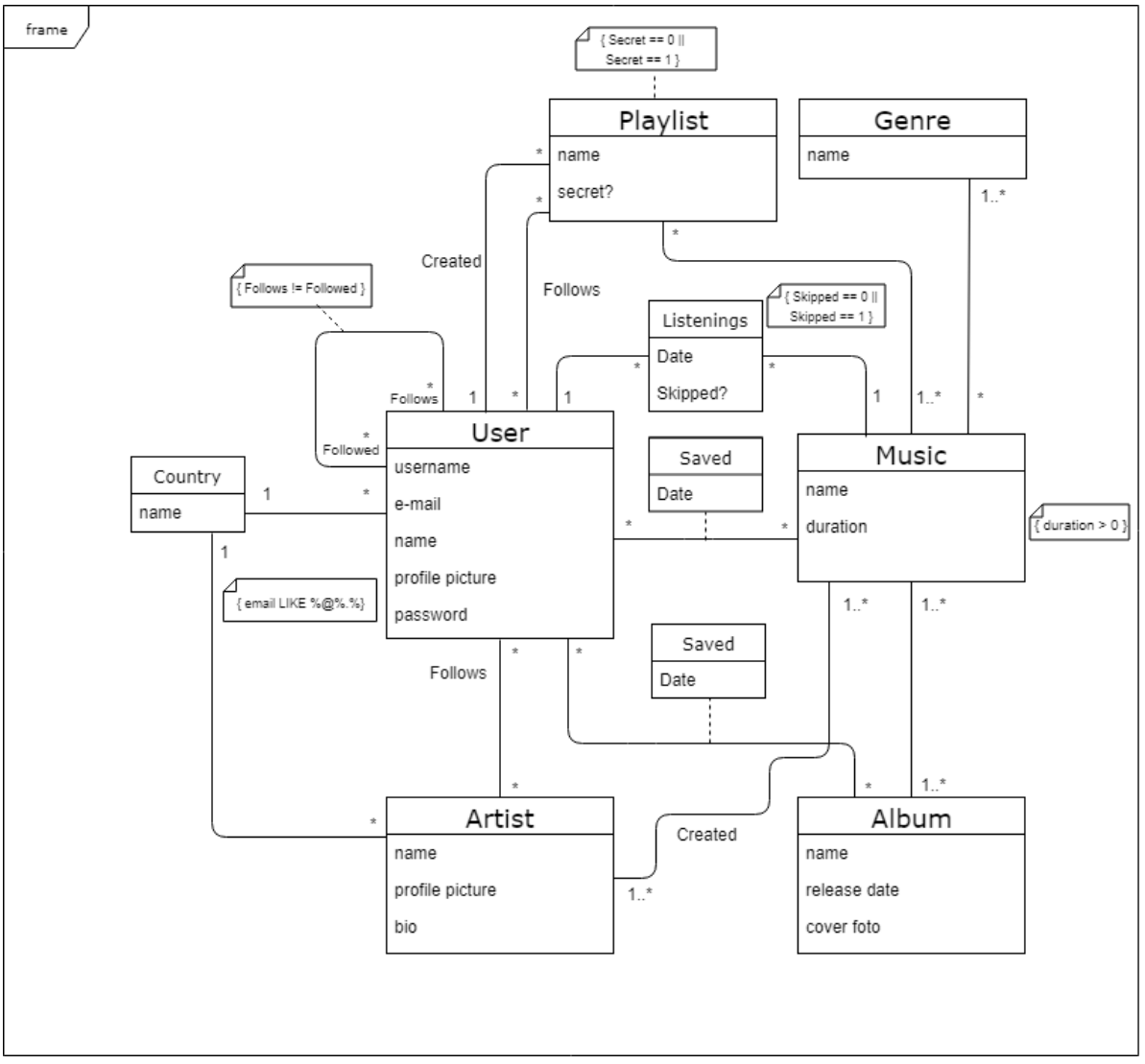
Um **álbum** é composto por pelo menos uma música, e interessa guardar-se a informação relativa ao seu nome, ID, data de lançamento e capa do álbum.

Uma ***playlist*** é também composta por pelo menos uma música, tem um nome e ID e está associada a um utilizador responsável pela

sua criação e aos utilizadores que a seguem. Pode ainda ser pública (todas as pessoas conseguem encontrá-la) ou privada (apenas o seu criador tem acesso a ela).

Um **género** é representado pelo seu nome e ID.

Diagrama UML



Modelo Relacional

User(ID, email,username,name,profile-pic,password,country->Country)

Music(ID,name,duration)

Album(ID,name,releasedate,cover-photo)

Artist(ID,name,profile-pic,bio,country->Country)

PlayList(ID,name,secret?,creator->User)

Genre(ID,name)

Country(ID,name)

Listenings(ID,date,skipped?,user->User,music->Music)

SavedMusic(user->User,music->Music,date)

SavedAlbum(user->User,album>Album,date)

FollowUser(follows->User, followed->User)

FollowPlayList(user->User,playlist->Playlist)

FollowArtist(follows->Artist,followed->User)

BelongsPlayList(music->Music,playlist->Playlist)

BelongsGenre(music->Music,genre->Genre)

BelongsAlbum(music->Music,album->Album)

CreatedMusic(artist->Artist,music->Music)

Dependências Funcionais

User(<u>ID</u>,
email,username,name,profile-pic,password,country->Country)
FDs:
ID->email,username,name,profile-pic,password,country
email->ID,username,name,profile-pic,password,country

Chaves da relação: {ID}, {email}

Music(<u>ID</u>,name,duration)
FDs:
ID->name,duration

Chaves da relação: {ID}

Album(<u>ID</u>,name,releaseDate,cover-photo)
FDs:
ID->name,releaseDate,cover-photo
name->ID,releaseDate,cover-photo

Chaves da relação: {ID}, {name}

Artist(<u>ID</u>,name,profile-pic,bio,country->Country)
FDs:
ID->name,profile-pic,bio,country
name->ID,profile-pic,bio,country

Chaves da relação: {ID}, {name}

PlayList(ID,name,secret?,creator->User)
FDs:
ID->name,secret?,creator
creator,name -> ID,secret?

Chaves da relação: {ID}, {creator, name}

Genre(ID,name)
FDs:
ID->name

Chaves da relação: {ID}

Country(ID,name)
FDs:
ID->name

Chaves da relação: {ID}

Listenings(ID,date,skipped?,user->User,music->Music)
FDs:
ID->date,skipped,user,music
user,music,date -> ID, skipped

Chaves da relação: {ID}, {user, music, date}

SavedMusic(user->User,music->Music,date)
FDs:
user,music -> date

Chaves da relação: {user, music}

SavedAlbum(user->User,album>Album,date)
FDs:
user,album->date

Chaves da relação: {user, album}

Todas as dependências funcionais das relações que se seguem são triviais.

FollowUser(follows->User, followed->User)

FollowPlayList(user->User,playlist->Playlist)

FollowArtist(follows->Artist,followed->User)

BelongsPlayList(music->Music,playlist->Playlist)

BelongsGenre(music->Music,genre->Genre)

BelongsAlbum(music->Music,album->Album)

CreatedMusic(artist->Artist,music->Music)

Formas normais

Com a informação anterior podemos concluir que todas as relações do modelo estão na *Boyce-Codd Normal Form* uma vez que o lado esquerdo de todas as dependências funcionais não triviais são super chaves das relações correspondentes. Assim sendo, podemos dizer que também estão na 3ª Forma Normal.

Restrições

Na criação da base de dados foram implementadas restrições para garantir a manutenção da integridade da mesma.

➔ Restrição NOT NULL:

Considera-se que as relações à frente mencionadas não fariam sentido sem os atributos referidos, pelo que se manifesta a obrigatoriedade de estes existirem através da restrição NOT NULL:

Na classe **Album**: name, date

Na classe **Artist**: name, country

Na classe **Country**: name

Na classe **Genre**: name

Na classe **Listenings**: date, user, music

Na classe **Music**: name, duration

Na classe **Playlist**: name

Na classe **SavedAlbum**: date

Na classe **SavedMusic**: date

Na classe **User**: email, name, password, country

➔ Restrição chave - UNIQUE

Não podem existir dois álbuns com o mesmo nome pelo que o atributo nome da respetiva relação é UNIQUE. Ao contrário dos álbuns, as músicas podem repetir nomes pelo que não foram implementadas com esta restrição.

Não podem existir dois artistas com o mesmo nome pelo que o atributo nome da respetiva relação é UNIQUE.

Não podem existir dois países com o mesmo nome pelo que o atributo nome da respetiva relação é UNIQUE.
Não podem existir dois géneros musicais com o mesmo nome pelo que o atributo nome da respetiva relação é UNIQUE.

➔ Restrição chave - PRIMARY KEY

Os atributos mencionados à frente são chaves primárias das relações a que pertencem, pelo que foram implementados com a restrição PRIMARY KEY:

Na classe **Album**: ID

Na classe **Artist**: ID

Na classe **BelongsAlbum**: music, album

Na classe **BelongsGenre**: music, genre

Na classe **BelongsPlaylist**: music, playlist

Na classe **Country**: ID

Na classe **CreatedMusic**: artist, music

Na classe **FollowArtist**: user, artist

Na classe **FollowPlaylist**: user, playlist

Na classe **FollowUser**: followed, follows

Na classe **Genre**: ID

Na classe **Listenings**: ID

Na classe **Music**: ID

Na classe **Playlist**: ID

Na classe **SavedAlbum**: user, album

Na classe **SavedMusic**: user, music

Na classe **User**: ID

➔ Restrição de integridade referencial - REFERENCES

O atributo country da relação **Artista** e **User** é uma chave estrangeira que aponta para a relação **Country**. É o resultado da modelação de uma associação *many to one*.

A relação **Listenings**, criada para armazenar informação acerca da atividade de um utilizador em relação às músicas, contém duas chaves estrangeiras, uma para **User** e outra para **Music**.

A relação **Playlist** tem uma chave estrangeira, creator, que aponta para **User**, representa o único criador da playlist. É o resultado da modelação de uma associação *many to one*.

Da definição do modelo relacional resultaram várias relações extra das associações *many to many*. Estas relações têm chaves estrangeiras para as relações que associa. Este foi o caso das seguintes relações: **BelongsAlbum**, **BelongsGenre**, **BelongsPlaylist**, **CreatedMusic**, **FollowArtist**, **FollowPlaylist**, **FollowUser**, **SavedAlbum**, **SavedMusic**.

➔ CHECK

Um email deve ser do tipo %@%.%, esta restrição foi implementada através de CHECK.

O atributo secret da relação **Playlist** e o atributo skipped da relação **Listenings** devem ser usados como booleanos e, como tal, só podem ter o valor 0 ou 1. Esta restrição foi implementada através de CHECK.

O atributo duration da relação **Music** referente à duração de uma música tem que ser superior a 0. Esta restrição foi implementada através de CHECK.