

**Faculdade de Engenharia da Universidade do Porto**



## **Base de Dados para plataforma de Streaming de música**



**Bases de Dados 2021/22 – Licenciatura em Engenharia Informática e Computação**

**Turma 13 – Grupo 1305**

Estudantes e autores:

Nuno Pereira: up202007865@up.pt      José Gaspar: up202008561@up.pt

Gonçalo Marques: up202006874@up.pt

**Novembro de 2021**

**Porto, Portugal**

## Índice:

<b>Context.....</b>	<b>p.2</b>
<b>UML Class Diagram.....</b>	<b>p.3</b>
Revised UML Class Diagram.....	p.4
<b>Relational Schema.....</b>	<b>p.5</b>
Normal Form Analysis.....	p.6
<b>Constraints.....</b>	<b>p.6</b>
<b>Queries.....</b>	<b>p.7</b>
<b>Triggers.....</b>	<b>p.7</b>
<b>Alterações.....</b>	<b>p.7</b>

## Contexto:

Foi pedido aos alunos do L.EIC para desenvolver uma solução para uma plataforma de *streaming* de conteúdo auditivo para uma empresa nacional.

Na plataforma existem utilizadores, cuja conta pode ser verificada ou não. Estes têm um nome que é apresentado sempre que outro utilizador visita o seu perfil. Para além disto, os utilizadores da plataforma podem seguir-se entre si, para saber coisas como as músicas preferidas de um dos seus amigos.

Um utilizador pode ser definido como um de dois tipos: artista ou consumidor. Se for um consumidor, apenas se guarda, junto da informação respeitante a um utilizador, o plano de pagamento em vigor para esse consumidor. Por outro lado, se o utilizador for um artista, regista-se o número de utilizadores que ouvem as suas músicas mensalmente, um pequeno texto descritivo do artista (pode ou não existir) e o seu ranking mundial.

Uma música é caracterizada pelo seu nome, a sua duração, se é explícita ou não, o género de música em que se enquadra, o número de *streams* até à data e o seu ano de lançamento. Todas as músicas fazem parte de um e apenas um álbum. Podem ou não também fazer parte de listas de reprodução. Em ambos os casos, uma música tem uma posição definida nessa coleção (álbum/lista de reprodução).

Em relação a uma música, é relevante registar os artistas que participaram a convite na produção e realização dessa mesma música.

Um álbum é caracterizado pelo seu nome, ano de lançamento, duração e número de músicas. Já uma lista de reprodução é descrita pelo seu nome, número de músicas, duração, e se a lista é privada ou não, isto é, se ela é invisível ou não para utilizadores que não sejam o criador.

Um álbum é também caracterizado pelo seu artista principal, propriedade que é partilhada por todas as músicas desse mesmo álbum.

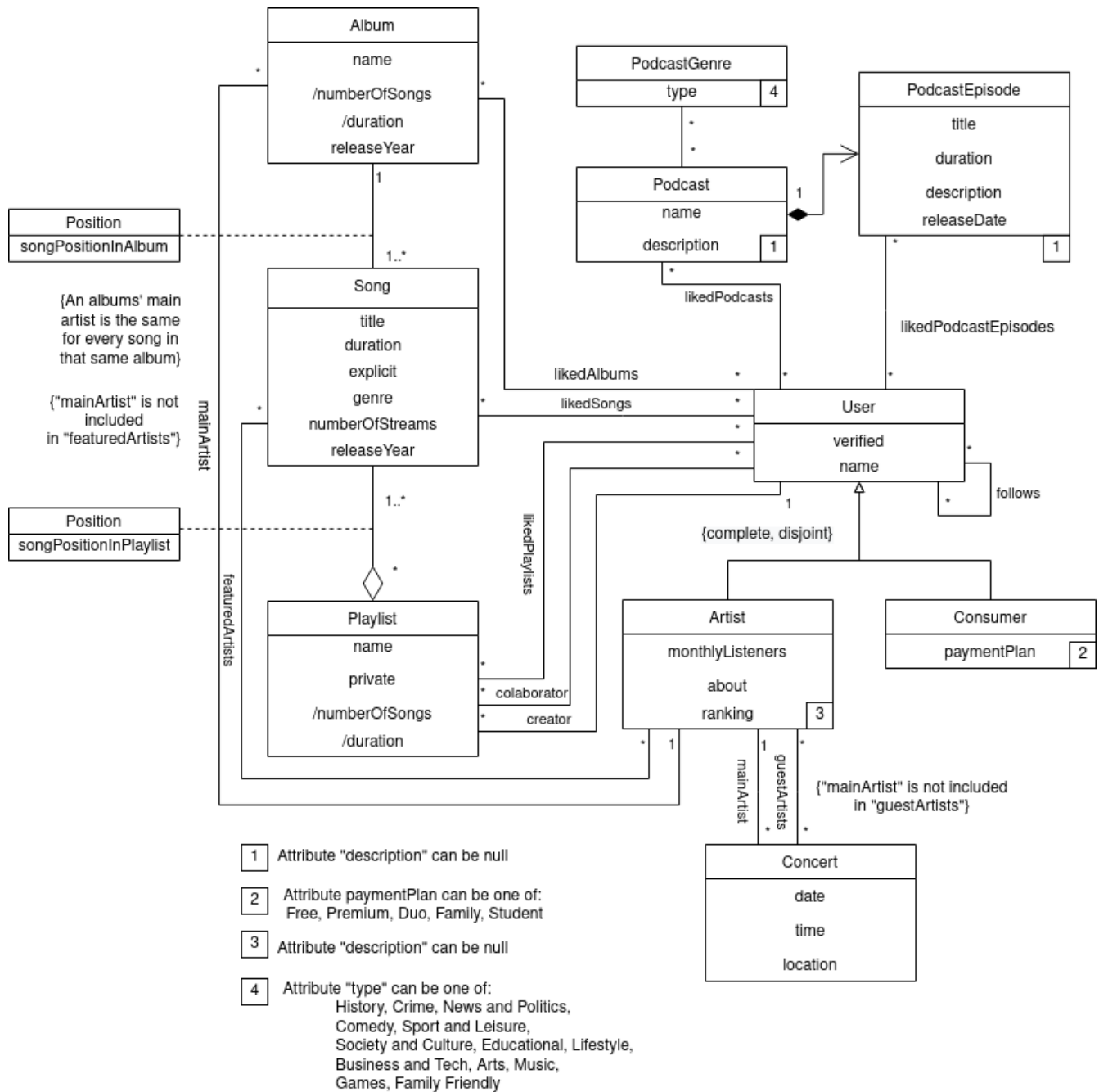
Um utilizador pode criar uma lista de reprodução, sendo distinguido dos restantes participantes dessa mesma lista de reprodução, que são designados como colaboradores. Um utilizador pode marcar uma lista de reprodução como “liked”. Também pode marcar músicas e álbuns como “liked”.

Um artista pode tocar/cantar em concertos, que são marcados pela data, o local e a hora do dia em que ocorrem, o artista principal e todos os artistas convidados. O artista principal não está incluído nos artistas convidados.

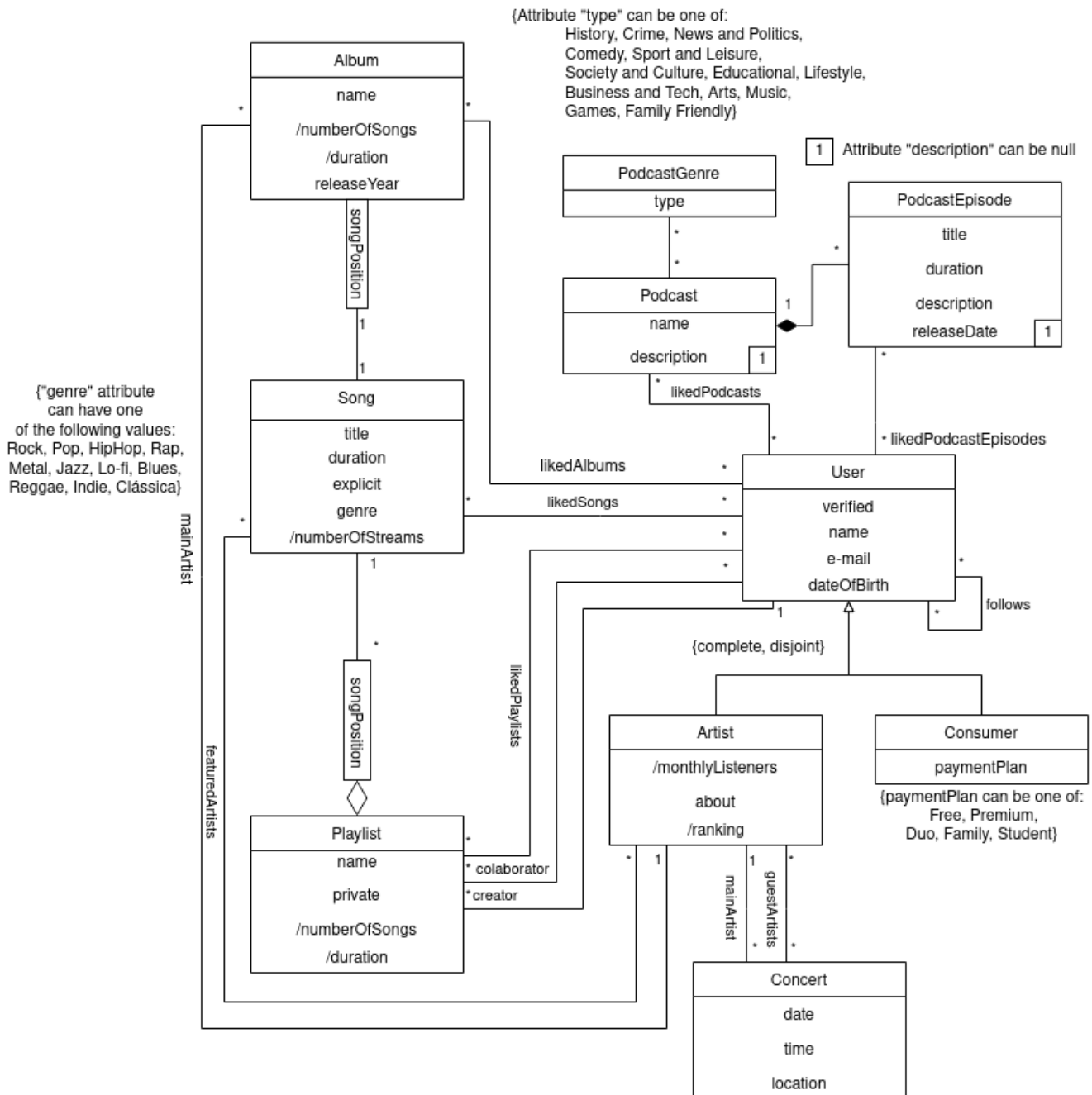
Para além de músicas, álbuns e listas de reprodução, existem também *podcasts*, que são marcados pelo seu nome e descrição opcional. Um *podcast* pode abordar tópicos relacionados com vários assuntos e portanto enquadrar-se em vários géneros de *podcast*. Um *podcast* é composto por vários episódios, que são caracterizados pela sua duração, o seu nome, uma descrição opcional e a data em que vão ser disponibilizados.

Um utilizador pode, também, marcar um *podcast* inteiro ou episódios individuais de um *podcast* como “liked”.

## UML Class Diagram



## Revised UML Class Diagram



## Esquema relacional

User (id, name, email UNIQUE, verified, dateOfBirth);

FDs: { {email} → {id, name, verified, dateOfBirth},  
          {id} → {email, name, verified, dateOfBirth} }

Artist (id→User.id, monthlyListeners, about?);

FDs: { {id} → {monthlyListeners, about} }

Consumer (id→User.id, paymentPlan);

FDs: { {id} → {paymentPlan} }

Concert (id, date, time, place, mainArtist→Artist);

FDs: { {id} → {date, time, place, mainArtist} }

GuestArtist (concertId→Concert, artistId→Artist);

Podcast (id, name, description?);

FDs: { {id} → {name, description} }

PodcastGenre (id, type);

FDs: { {id} → {type} }

TypesOfPodcast (podcastId→Podcast, podcastGenreId→PodcastGenre);

PodcastEpisode (podcastId→Podcast, episodeNumber, title, duration, description?, releaseDate);

FDs: { {podcastId, episodeNumber} → {title, duration, description, releaseDate} }

LikedPodcasts (podcastId→Podcast, userId→User);

LikedPodcastEpisodes (podcastId→PodcastEpisode, episodeNumber→PodcastEpisode, userId→User);

Album (id, name, numberOfSongs, duration, releaseYear, mainArtist→Artist);

FDs: { {id} → {name, numberOfSongs, duration, releaseYear, mainArtist} }

Song (id, name, duration, explicit, genre, numberOfStreams, album→Album.id, positionInAlbum);

FDs: { {id} → {name, duration, explicit, genre, numberOfStreams, album, positionInAlbum} }

Playlist (id, name, private, numberOfSongs, duration, creator→User);

FDs: { {id} → {name, private, numberOfSongs, duration, creator} }

SongPositionInPlaylist (songId→Song, playlistId→Playlist, songPosition);

FDs: { {songId, playlistId} → {songPosition} }

LikedAlbum (albumId→Album, userId→User);

LikedSong (songId→Song, userId→User);

LikedPlaylist (playlistId→Playlist, userId→User);

Colaborator (playlistId→Playlist, userId→User);

FeaturedArtists (songId→Song, artistId→Artist);

Follows (userId1→User.id, userId2→User.id);

## Esquema relacional – Análise das Formas Normais

Nenhuma relação viola quer a 3ª Forma Normal quer a Forma Normal de Boyce-Codd porque, quando têm FDs, apenas contêm a FD  $\{\langle \text{primaryKey} \rangle\} \rightarrow \{\langle \text{restantes atributos da relação} \rangle\}$ . Como  $\{\langle \text{primaryKey} \rangle\}$  é uma superchave da relação, esta não encorre em nenhuma violação das Formas Normais antes mencionadas.

### Restrições

- Não existem dois utilizadores diferentes com o mesmo ID e mesmo *email*; (PRIMARY KEY + UNIQUE)
- Por defeito, um utilizador não é verificado; (DEFAULT FALSE)
- Um artista tem sempre ouvintes mensais superiores ou iguais a 0; (CHECK (monthlyListeners >= 0))
- Um consumer tem, por defeito, um plano de pagamento “Free”, podendo mudar para um de "Premium", "Duo", "Family" ou "Student"; (CHECK (paymentPlan IN ("Premium", "Free", "Duo", "Family", "Student"))) DEFAULT "Free")
- Na relação PodcastGenre, o atributo *genre* tem que ser único, só podendo ter os valores "History", "Crime", "News and Politics", "Comedy", "Sport and Leisure", "Society and Culture", "Educational", "Lifestyle", "Business and Tech", "Arts", "Music", "Games" ou "Family Friendly".
- O número de um episódio de podcast é sempre maior ou igual a 1; (CHECK (episodeNumber >= 1))
- Um álbum tem sempre pelo menos uma música; (CHECK (numberOfSongs >= 1) DEFAULT 1)
- Um álbum tem uma duração superior a 0; (CHECK (duration > 0))
- Uma música tem uma duração superior a 0; (CHECK (duration > 0))
- Por defeito, uma musica não é explícita; (DEFAULT FALSE)
- O género de uma música é um de “Rock”, “Pop”, “HipHop”, “Rap”, “Metal”, “Jazz”, “Lo-fi”, “Blues”, “Reggae”, “Indie”, “Clássica”.
- O número de *streams* de uma música é sempre 0 ou maior; (CHECK (numberOfStreams >= 0))
- A posição de uma música no seu álbum é 1 ou maior; (CHECK (positionInAlbum >= 1))
- Por defeito, uma playlist é privada; (DEFAULT TRUE)
- O número de músicas e a duração de uma *playlist* é sempre positivo; (CHECK(numberOfSongs >= 0) CHECK (duration >= 0))
- A posição de uma música numa *playlist* deve ser pelo menos 1; (CHECK(position >= 1))
- Nenhum atributo deve ser nulo com excessão das descrições de *podcasts* e episodios de *podcasts*; (NOT NULL)
- Os atributos *duration* e *numberOfSongs* das relações Playlist e Album vão sendo alterados à medida que se adicionam músicas a cada uma delas; (IMPLEMENTAÇÃO USANDO TRIGGERS)
- Existe um *ranking* entre os vários artistas que depende do número de ouvintes mensais de cada um; (ATRIBUTO VIRTUAL)

## Queries

- 1- Qual foi o género musical mais ouvido?
- 2- Qual o ranking dos artistas (por número de ouvintes)?
- 3- Qual a música mais ouvida de cada um dos 5 artistas mais ouvidos?
- 4- Qual a media de ouvintes de cada album?
- 5- Quais as músicas favoritas em comum entre utilizadores com plano gratuito e utilizadores com plano pago?
- 6- Quantos seguidores tem cada artista?
- 7- Que músicas podem ser recomendadas ao utilizador 1, baseado nas músicas que ele gosta?
- 8- Quais os concertos que acontecerão em Wembley?
- 9- Com que frequência é que saem episódios novos no Podcast “The Jordan B. Peterson Podcast”?
- 10- Qual a idade média dos colaboradores da Playlist 1?

## Triggers

- 1- Atualizar automaticamente a duração de álbuns, playlists e o número de músicas dessas tabelas sempre que lhes é adicionada/removida uma música. Também atualiza o número de episódios de um Podcast quando lhe é adicionado/removido um episódio.
- 2- Sempre que um podcast for eliminado, é necessário efetuar também a remoção dos episódios desse mesmo Podcast, assim como todas as referências ao mesmo que possam estar presentes na base de dados, nomeadamente nas tabelas TypesOfPodcast, LikedPodcastEpisode, LikedPodcast.
- 3- Verifica se, ao inserir uma música na tabela, o seu género musical pertence a um dos géneros suportados pela base de dados.

## Alterações

Para a submissão da terceira entrega, foram efetuadas alterações nos ficheiros criar.sql e povoar.sql, sendo que, para a avaliação das queries e dos triggers é essencial a utilização das versões atualizadas dos ficheiros mencionados.