

Recriação de uma versão simplificada da base de dados Apple Music na linguagem SQL

CC2005 – Base de Dados

Ano letivo 2021/22

Docente

Eduardo Marques

Discentes

José Miguel Alves (up201709215)

Introdução

No âmbito de desenvolver uma base de dados para um universo à escolha, foi recriado o mundo Apple Music devido à familiaridade com o programa em questão. A linguagem de programação utilizada foi SQL e o software usado para o desenvolvimento dos modelos ER e relacional foi o dbdia.

Identificação dos ficheiros do trabalho

- AppleMusicDB.sql contém a base de dados;
- AppleMusic_With_Relations a descrição das relações entre as entidades tipo (Fig.1);
- AppleMusic_without_Relations a descrição das entidades-tipo da base de dados (Fig.2);
- AppleMusic_MR descrição do modelo relacional (Fig.3).

Requisitos da base de dados

Quanto aos requisitos considerados para esta base de dados foram consideradas as seguintes entidades-tipo e respetivos atributos:

- USER com os atributos: IdUser, Name, Email, BirthDate, Sex, Joined, Email e Phone;
- LIBRARY com os atributos: IdLibrary, IdUser;
- ARTIST com os atributos: IdArtist, Name e Genre;
- PLAYLIST com os atributos: IdPlaylist, Name e IdLibrary;
- MUSIC com os atributos: IdMusic, Name, Album e IdArtist.

Modelo ER

No âmbito de transformar os requisitos num modelo ER, foram escolhidas 5 entidades-tipo adequadas à base de dados escolhida. Sendo assim, optou por se utilizar o USER, LIBRARY, ARTIST, PLAYLIST e MUSIC com o objetivo de cumprir os requisitos propostos, como o de exemplificar atributos opcionais, derivados, multivalores e compostos.

Como atributo derivado tem-se o Age no USER, como opcional o Phone no USER e o Album no MUSIC, como multivalores o Genre no ARTIST e por fim como composto o Payment no USER.

Após a criação das entidades tipos e da atribuição de valores decidiu-se criar 7 relações. Primeiro, a relação HAS que liga o USER com a LIBRARY. Esta é uma relação de "1 para 1" (1/1) pois cada utilizador pode possuir apenas uma biblioteca tal como cada biblioteca é utilizada por apenas um utilizador. Para além disso é uma relação total dos dois lados, devido ao facto de todos os utilizadores possuem uma biblioteca e todas as bibliotecas são utilizadas por um utilizador.

Em segundo temos o LISTENS que conecta o USER com a PLAYLIST. Esta é uma relação de "muitos para muitos" (m/n) pois cada utilizador pode ouvir várias playlists, tal como cada playlist pode ser ouvida por vários utilizadores. Para além disso é uma relação parcial dos dois lados, devido ao facto de que um utilizador pode não estar a ouvir nenhuma playlist tal como uma playlist pode não estar a ser ouvida por nenhum utilizador.

Tem-se também o CREATE usado entre a LIBRARY e a PLAYLIST. Esta é uma relação de "um para muitos" (1/n), pois cada biblioteca é pode criar várias playlists, no entanto playlist é criada a partir de uma única biblioteca.

Temos o relacionamento CONTAINS utilizado entre a PLAYLIST e a MUSIC. Esta é uma relação de "muitos para muitos" (m/n) pois cada música pode estar contida em várias playlists tal como cada playlist tem várias músicas. Para além disso, esta relação possui uma participação total por parte da PLAYLIST pois uma playlist tem necessariamente de ter uma música e parcial por parte da MUSIC pois uma música não precisa de estar numa playlist.

O HEARS, que relaciona o USER com a MUSIC é uma relação de "muitos para muitos" (m/n), pelo que um utilizador pode ouvir várias músicas e uma música pode ser ouvida por vários utilizadores. Para além disso é uma relação totalmente parcial porque um utilizador pode não estar a ouvir nenhuma música tal como uma música pode não estar a ser ouvida por nenhum utilizador.

De seguida temos o PRODUCES que relaciona a MUSIC com o ARTIST. É uma relação de "um para muitos" (1/n), pois cada música é produzida apenas por um artista, no entanto cada artista pode produzir várias músicas. É uma relação com participação total de uma das entidades, pois uma música tem de necessariamente ser produzida por algum artista tal como cada artista tem de ter produzido pelo menos uma música.

Por fim, temos o relacionamento FOLLOWS utilizado pelo USER consigo mesmo. Esta é uma relação de "muitos para muitos" (m/n) pois cada utilizador pode ser amigo de vários outros utilizadores. Para além disso, esta relação possui uma participação parcial da entidade, pois nem todos os utilizadores têm de seguir outros utilizadores.

Modelo Relacional

Para traduzir o modelo ER obtido para o modelo relacional, começou por se criar a tabela do USER com um atributo chave (IdUser). Esta possuía um atributo composto (PAYMENT) para a qual foi necessário criar uma tabela, em que existia uma ligação através do atributo chave IdUser (chave primária de USER). Os restantes atributos da tabela foram criados posteriormente.

Se seguida foi criada a tabela LIBRARY com o atributo chave IdLibrary. Resolveu-se o relacionamento HAS ao colocar uma chave primária (IdUser) como uma chave externa na LIBRARY, devido à cardinalidade ser 1/1 e haver participação total da LIBRARY.

Foi criada a tabela PLAYLIST com o atributo chave IdPlaylist. Resolveu-se o relacionamento LISTENS criando-se uma tabela específica para a relação com ambas as chaves primárias de USER e PLAYLIST, sendo estas chaves primarias no LISTENS. Resolveu-se também o relacionamento CREATE ao colocar uma chave primária (IdLibrary) como uma chave externa na PLAYLIST devido ao facto de a cardinalidade ser 1/n.

Criou-se também a tabela MUSIC com o atributo chave IdMusic e, de seguida, a tabela CONTAINS com cardinalidade m/n. Assim fez-se uma tabela específica para a relação com ambas as chaves primárias de GAMES e LIBRARY, sendo estas chaves primarias no CONTAINS.

Posteriormente foi concebido o relacionamento HEARS com cardinalidade m/n. Assim, gerou-se uma tabela específica para a relação com ambas as chaves primárias de USER e MUSIC, sendo estas chaves primarias no HEARS.

Prontamente, gerou-se a tabela ARTIST com a chave primaria IdArtist. Com isto, conseguiu-se fazer a relação PRODUCES que é uma relação 1/n. Resolveu-se o relacionamento PRODUCES ao colocar uma chave primária (IdArtist) como uma chave externa no MUSIC devido ao facto de a cardinalidade ser 1/n e haver participação total da ambas as partições.

Para finalizar, criou-se a tabela GENRE com a chave primária IdGenre. Sendo esta uma tabela multivalor relacionada com o ARTIST, houve a necessidade de ser criada posteriormente e não dentro do ARTIST.

Volume de dados

O trabalho foi realizado sem grandes dificuldades. Assim, a base de dados ficou com um total de 173 entradas distribuídos por 11 tabelas da seguinte forma:

Nome	Nº de Entradas
USER	13
PAYMENT	13
LIBRARY	13
PLAYLIST	11
LISTENS	11
ARTIST	21
MUSIC	25
HEARS	7
GENRE	26
CONTAINS	24
FOLLOWS	9
TOTAL	173

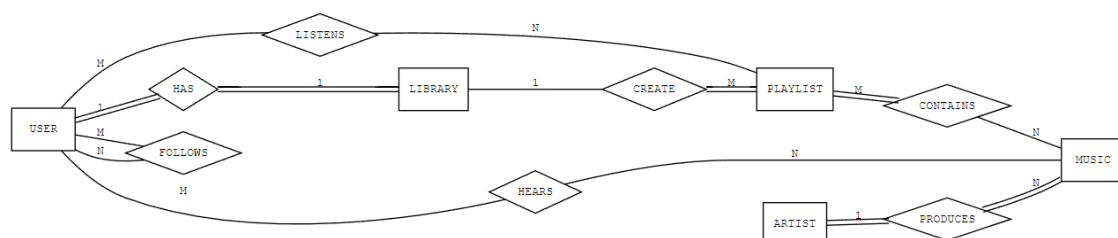


Fig.1

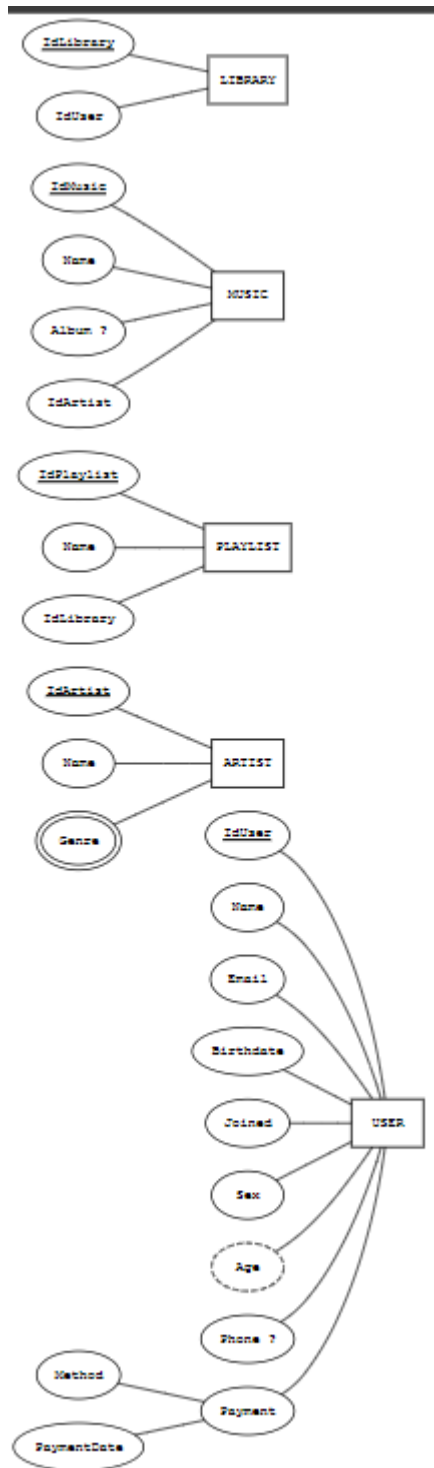


Fig.2

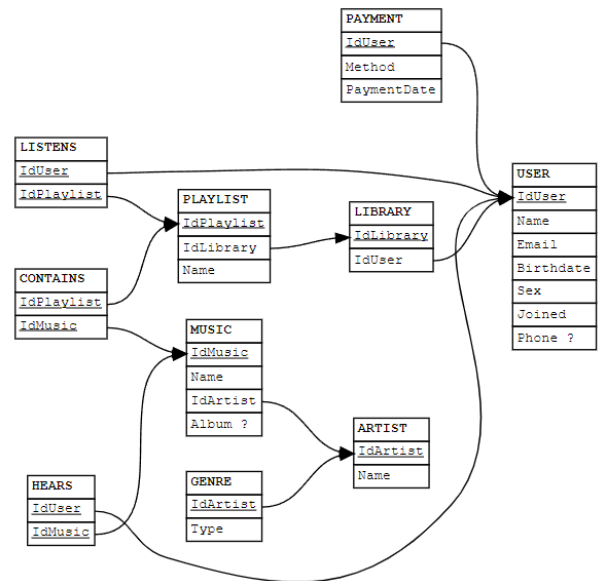


Fig.3