Engenharia de Computadores e Telemática
Departamento de Eletrónica, Telecomunicações e Informática Universidade de Aveiro

Relatório de BD - App de música

28/05/2023

Telmo Sauce-104428
Ricardo Covelo-102668

Índice

Introdução

Decidimos criar um sistema de serviços de música como parte do nosso projeto para a disciplina de Base de Dados. Essa plataforma permitirá que os usuários publiquem e acessem músicas, álbuns, podcasts e playlists. O objetivo principal do desenvolvimento desse software é aplicar, de forma abrangente, os conhecimentos teóricos e práticos adquiridos durante as aulas, desde a concepção e modelagem do banco de dados até a sua gestão e manipulação por meio de sistemas de software.

Iniciar a aplicação

Para executar nosso programa, é necessário executar o arquivo index.py, para o executar é necessário ter os imports “pypyodbc” e “Flask” do python . Antes de iniciar a execução, é importante modificar as credenciais necessárias no próprio arquivo index.py, mais especificamente nas linhas 9, 19 e 20. Certifique-se de substituir as informações adequadas nessas linhas para garantir a conexão correta com os recursos e serviços necessários. Após fazer essas alterações, você estará pronto para iniciar o programa e desfrutar de suas funcionalidades.

Entregáveis

MusicApp.sql- Script com o esquema da base de dados desenvolvida para o projeto.

inserts.sql- Script com alguns inserts que fizemos para testar de maneira inicial a nossa base de dados.

triggers.sql- Script com todos os triggers usados no projeto

udf.sql- Script com todos os udf's usados no projeto

sp.sql- Script com todos os sp's usados no projeto

ProjDB- Contém todos os ficheiros HTML(/template) e o index.py que vai inicializar o site e os ficheiros mencionados anteriormente(/db) .

Desenho da base de dados

Desde o início do desenvolvimento do nosso projeto que o desenho da base de dados que implementámos tem vindo a sofrer alterações no sentido de a tornar mais robusta, menos suscetível a erros e uma melhor representação da realidade a que se propõe. Na próxima figura apresentamos o esquema que desenhamos para este planejamento.

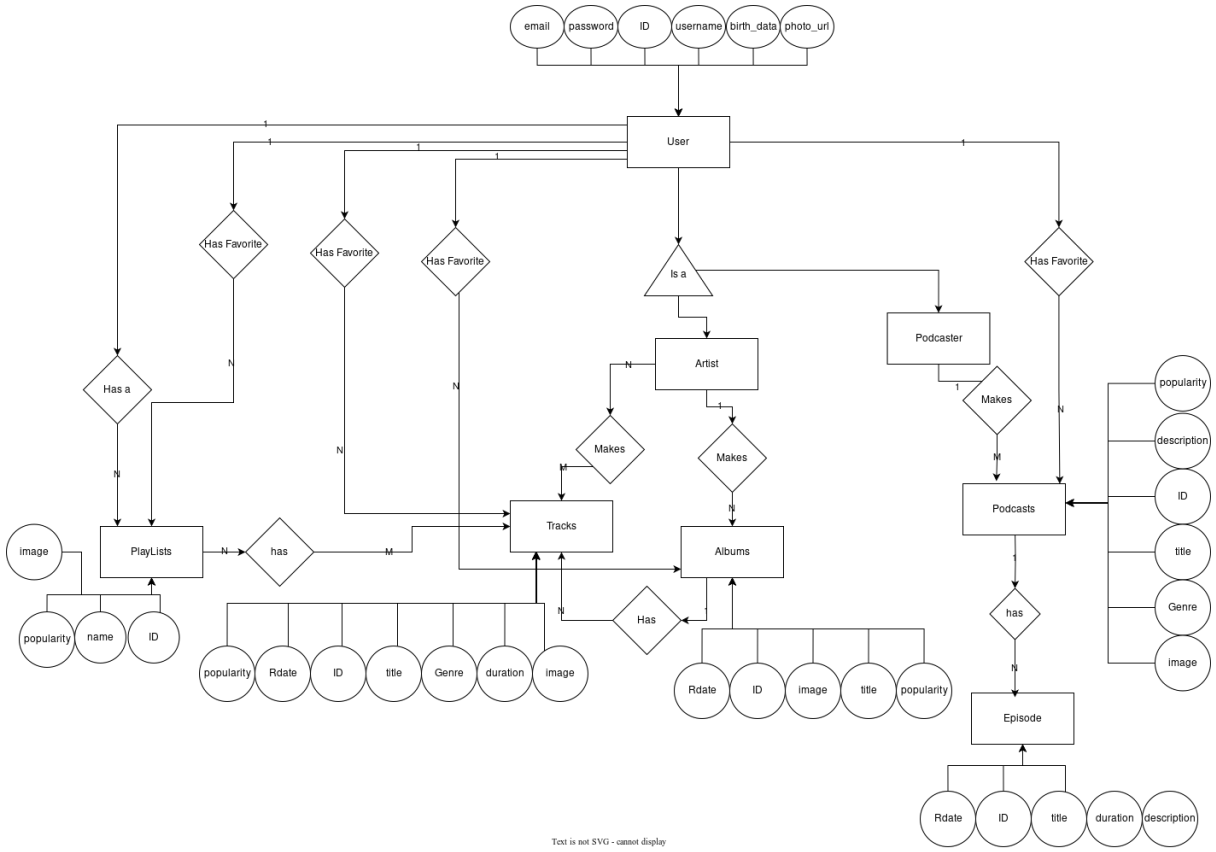

fig1-Diagrama ER

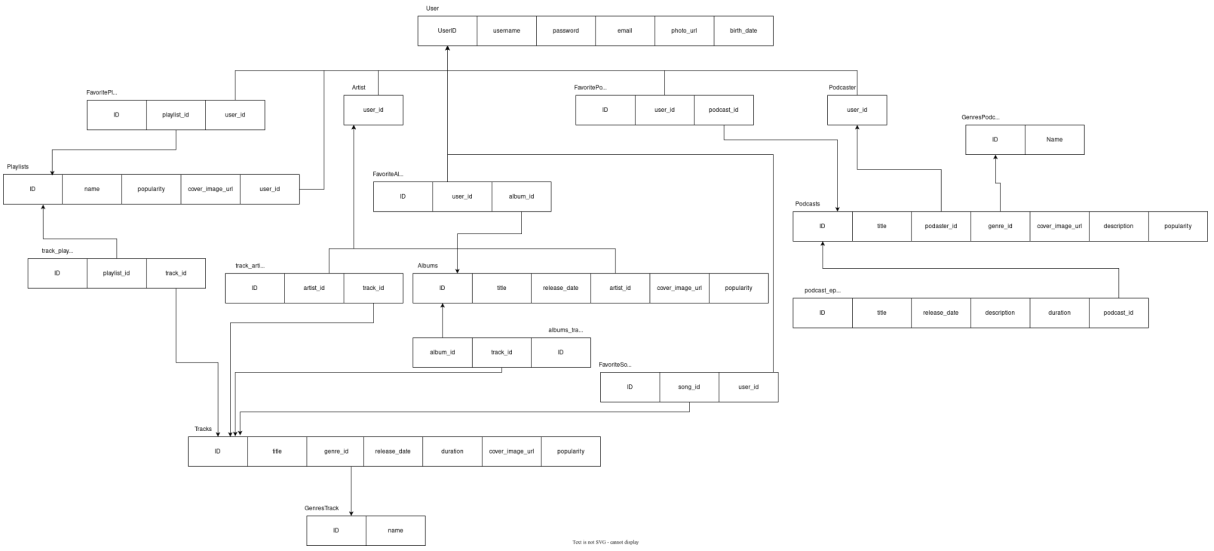

fig2-Esquema Relacional

Índices

Embora possa parecer desnecessário à primeira vista, decidimos implementar um índice no nome das tabelas de "album", "tracks" e "playlists" em nossa aplicação. Essa escolha foi feita com o objetivo de melhorar a velocidade das pesquisas realizadas pelos usuários uma vez que estas são as colunas com mais informação e ao mesmo tempo com o maior número de acessos.Ao criar um índice no nome das tabelas, facilitamos o acesso rápido aos registros relevantes, reduzindo o tempo de busca e tornando a experiência do usuário mais eficiente. Dessa forma, quando os usuários pesquisarem por álbuns, faixas ou playlists específicas, a resposta será entregue de maneira mais ágil.

Udf

No contexto do Flask, embora os UDFs (User-Defined Functions) possam ser evitados, optamos por implementá-los visando a reutilização e melhor visibilidade. Por exemplo, a função "SearchByCategory" recebe uma categoria como parâmetro e retorna uma tabela com o conteúdo necessário. Outro exemplo é a função "GetArtistTracks", que não apenas retorna as faixas de cada artista, mas também substitui o valor do campo "id_genre" da tabela pelo nome correspondente.

Trigger

Em relação aos triggers, eles estão vinculados exclusivamente às tabelas de favoritos e são acionados após uma operação de inserção (insert) ou remoção (remove). Por exemplo, ao inserir um registro na tabela "FavoriteSongs", o valor do campo "popularity" na tabela "Tracks" é incrementado em 1. Da mesma forma, durante uma exclusão, o valor do campo "popularity" é decrementado em 1.

SP

No processo de registro de usuários, foi implementada uma Stored Procedure (SP) que realiza o Insert na tabela "users" para todos os usuários registrados. Além disso, dependendo do tipo de usuário (artista, podcaster ou usuário comum), eles também são adicionados às tabelas correspondentes, como a tabela "artista" ou uma tabela específica para podcasters. Essa SP garante que os registros sejam inseridos nas tabelas adequadas, mantendo a consistência dos dados no banco de dados.

Para além disso foram implementados alguns SP para aumentar a rapidez de algumas “querys”. Como na remoção e inserção de “tracks” favoritas.

Segurança

A fim de assegurar a segurança da base de dados e evitar possíveis ataques de injeção de SQL por parte do usuário, adotamos a prática de parametrizar todas as instâncias da classe SqlCommand.

Notas finais

Devido ao nosso crescente conhecimento e experiência no uso do framework Flask, tomamos a decisão de implementar a frontend do nosso site utilizando HTML, com conexão direta à base de dados através do framework Flask. Embora não fosse um requisito obrigatório para o projeto da disciplina, optamos por criar uma interface gráfica simples. Essa escolha foi motivada pelo nosso desejo de explorar e aprimorar nossas habilidades na construção de interfaces de usuário, além de proporcionar uma experiência mais agradável aos usuários finais.