U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Aplicação web simples com interface na BD desenvolvida no primeiro projeto

CC2005 – Base de Dados

Ano letivo 2021/22

Docente
Eduardo Marques

Discentes
José Miguel Alves (up201709215)

Introdução

Com o intuito de desenvolver uma aplicação web, foi utilizada como base a aplicação exemplo fornecida para a DB MovieStream adaptando a mesma à base de dados desenvolvida enteriormente (AppleMusic).

As linguagens de programação utilizadas foram: SQL para pesquisas na base de dados, python para o desenvolvimento do website e html para a criação dos templates. Foram utilizadas as bibliotecas Flask e PyMySQL com o âmbito de auxiliar a criação da mesma.

Funcionalidades e Requisitos

Quanto aos requisitos considerados para esta aplicação, foi necessário em primeiro lugar, garantir a existência de um "endpoint" para a página de entrada (/) de modo a permitir a navegação na DB.

De seguida, foram criadas operações que permitem a listagem de todos os atributos de 3 tabelas onde a página gerada contém "links" para aceder a cada registo individual. As tabelas utilizadas para este propósito foram a MUSIC, ARTIST e PLAYLIST.

Também é possível obter a listagem de outros atributos como é o caso do GENRE, USER e Albums no entanto estes não permitem o acesso aos registos individuais.

Criaram-se funções que permitem a pesquisa de um dado específico através de uma chave. Tanto para a tabela MUSIC como ARTIST foram criadas duas caixas de pesquisa onde é possível procurar os mesmos tanto por Id como pelo Nome.

Houve a necessidade de utilizar uma junção de 3 tabelas para fazer uma procura entre a LIBRARY, o USER e a PLAYLIST já que o IdUser (chave primária do USER) está presente na LIBRARY tal como a IdLibrary (chave primária de LIBRARY) está presente na PLAYLIST sendo assim possível obter o nome do user que criou cada playlist (Fig.1).

Foi realizada também uma junção de três tabelas entre a PLAYLIST, MUSIC e CONTAINS para poder demonstrar o número de músicas presentes em cada playlist. Esta função utilizou também uma agregação agrupada (Group By) necessária para agrupar a contagem de músicas por playlists (Fig.2).

```
163
164 @APP.route('/playlist/')
165 def list_playlist():
166     playlist = db.execute('''
167     SELECT LIBRARY.IdUser, PLAYLIST.Name AS PlaylistName, USER.Name AS UserName, PLAYLIST.IdPlaylist
168     FROM LIBRARY JOIN PLAYLIST ON (PLAYLIST.IdLibrary = LIBRARY.IdLibrary)
169     JOIN USER ON (LIBRARY.IdUser = USER.IdUser)
170     ORDER BY PLAYLIST.NAME
171     ''').fetchall()
172     return render_template('playlist-list.html', playlist=playlist)
173
```

Fig.1

```
    nmr = db.execute(
    '''
    SELECT PLAYLIST.Name, COUNT(MUSIC.IdMusic) AS NmrSongs
    FROM PLAYLIST JOIN CONTAINS ON (CONTAINS.IdPlaylist = PLAYLIST.IdPlaylist)
    JOIN MUSIC ON (CONTAINS.IdMusic = MUSIC.IdMusic)
    GROUP BY PLAYLIST.Name
    ''').fetchall()
    return render_template('playlist-list.html', playlist=playlist, nmr=nmr)
```

Fig.2

Mudanças Aplicadas à Base de Dados Antiga

Foram acrescentados mais dados de maneira a obter resultados que demonstram melhor as funcionalidades a ser aplicadas.

Dúvidas e problemas encontrados

Devido à natureza do atributo Album não foi possível agrupar as músicas por álbum. Também no Genre, devido à falta de um Id conectado à tabela ARTIST, não foi possível a agrupação das músicas por géneros.

Estes problemas poderiam ter sido resolvidos alterando estes dois atributos na base de dados originais, no entanto estes problemas só se tornaram evidentes numa fase muito avançada do trabalho, tornando impossível a sua correção.