Projecto de Bases de Dados (CC2005) - parte 2

1. Elementos do grupo

Grupo nº [32]

Nº mecanográfico	**Nome**
202207036	Magda Costa
202206258	Rafael Pacheco
202207203	Sofia Machado

2. Ajustes ao modelo da BD

Trocámos as designações “Classe-associação” por “Tabela auxiliar” e “chaves secundárias” por “chaves externas”. Explicámos também o porquê de uma personagem só poder estar num filme.

3. Povoamento de tabelas

Para o povoamento das tabelas, utilizámos ficheiros CSV que construímos a partir dos dados do dataset anteriormente mencionado.

Porém, para um povoamento eficiente e correto das tabelas, fizemos uma análise mais profunda do nosso dataset e procedemos à sua limpeza e organização, a fim de o enriquecer e evitar a perda dados.

Os dados acrescentados manualmente foram cuidadosamente escolhidos para nos certificarmos de que eram consistentes e reais. Nomeadamente, esses dados são:

- **Character Species:** espécie do personagem;

- **Character Role:** papel de cada personagem;
- **Actor Gender:** género do ator;
- **Genre Name and Description:** nome do género do filme e a respetiva descrição;
- **Director Gender:** género do diretor;

Finalmente, fizemos com que o nome das colunas dos CSV correspondessem aos nomes das colunas das tabelas para um povoamento mais eficiente.

Mais concretamente, utilizámos sqlite e a função “Import into the table”, atribuindo a cada tabela o respetivo ficheiro CSV com todas as alterações previamente referidas (com encoding UTF-8).

Nome da tabela	**Nº de entradas**
Actor	459
Character	570
Director	31
Genre	10
Movie	49

4. Aplicação Python

Uma vez que a descrição dos endpoints é repetitiva, optámos por agrupar aqueles que possuem as mesmas funcionalidades com o intuito organizar a informação de forma mais eficiente e legível. Deste modo, todos aqueles que pretendem listar todos os dados de uma determinada tabela do nosso dataset, por exemplo, irão encontrar-se agrupados.

“Endpoint”	Funcionalidade
/	Página de entrada.
/search/	Página que contém links que redirecionam para outras páginas que permitem realizar a pesquisa de características.
/movies/	Página que contém uma tabela com todos os filmes ordenados por ordem alfabética, assim como pesquisas de filme por nome, id ou ano de lançamento.
/characters/ **/actors/** **/directors/** **/genres/**	Páginas que contêm uma tabela com todos os personagens/atores de voz/diretores/géneros, respetivamente, ordenados por ordem alfabética, assim como pesquisas dos mesmos por nome ou id.
/movies/name/<expr>/ **/characters/name/<expr>/** **/actors/name/<expr>/** **/directors/name/<expr>/** **/genres/name/<expr>/**	Mostra o resultado da pesquisa por nome em forma de tabela
/movies/year/<expr>/	Mostra o resultado da pesquisa por ano em forma de tabela
/movies/id/<int:expr>/ **/characters/id/<int:expr>/** **/actors/id/<int:expr>/** **/directors/id/<int:expr>/** **/genres/id/<int:expr>/**	Páginas com informação do filme/personagem/ator/diretor/género, respetivamente, após selecionar o seu id
/movies/info/<int:id>/ **/characters/info/<int:id>/** **/actors/info/<int:id>/** **/directors/info/<int:id>/** **/genres/info/<int:id>/**	Páginas com informação do filme/personagem/ator/diretor/género, respetivamente, após selecionar o seu id
/species/info/<expr>/	Página que contém informação sobre a espécie do personagem, assim como uma tabela contendo os personagens da mesma espécie
/roles/info/<expr>/	Página que contém informação sobre o papel do personagem, assim como uma tabela contendo os personagens que realizam também esse papel
/gender/info/<expr>/	Página que contém informação sobre o género do ator ou diretor, assim como uma tabela contendo os membros da equipa técnica (atores ou diretores) com o mesmo género

Relativamente aos códigos SQL

A ilustrar a utilização de agregação agrupada de informação, temos uma tabela que exibe o número de entradas de cada tabela da nossa base de dados, representada na página de entrada:

SELECT * FROM (SELECT COUNT(*) n_movies FROM MOVIE) JOIN (SELECT COUNT(*) n_characters FROM Character) JOIN (SELECT COUNT(*) n_actors FROM ACTOR) JOIN (SELECT COUNT(*) n_directors FROM Director) JOIN (SELECT COUNT(*) n_genres FROM GENRE)	Este código SQL utiliza a função de agregação COUNT que faz a contagem de registos por tabela escolhida. Adicionalmente, realiza-se junção entre todas as tabelas para mostrar o número de entradas de cada tabela da base de dados.

A título de exemplo da utilização da junção de 3 ou mais tabelas, segue abaixo um excerto do código utilizado para exibir a informação de um filme:

SELECT a.Name AS n_A, a.IdActor AS id_A, c.Name AS n_C, c.IdCharacter AS id_C FROM Actor a JOIN Character c ON a.IdActor=c.IdActor JOIN Movie m on c.IdMovie=m.IdMovie WHERE m.IdMovie = ?	Este código SQL executa o JOIN (junção) de três tabelas, “Actor”, “Character” e “Movie”, o que nos permite aceder a informações que consideramos pertinentes relativamente à informação de um filme mas que não se encontram presentes na tabela “Movie”: - Nome e Id dos atores desse filme - Nome e Id dos personagens desse filme

Finalmente, para demonstrar as várias maneiras como podemos pesquisar por informação, temos:

SELECT IdMovie, Name, "Date of Release" as ReleaseDate FROM MOVIE WHERE Name LIKE ? SELECT IdMovie, Name, "Date of Release" as ReleaseDate, Gross, IdDirector FROM Movie ORDER BY Name	Estas queries, quando usadas em conjunto, permitem, a partir de uma função, que os usuários pesquisem filmes pelo nome (Query 1) e exibindo detalhes adicionais sobre todos os filmes, ordenando-os pelo nome (Query 2). Este tipo de queries é usado para pesquisas por nome nas tabelas movies, actors, directors, characters e genres.
SELECT IdMovie, Name, "Date of Release" as ReleaseDate FROM MOVIE WHERE ReleaseDate LIKE ? SELECT IdMovie, Name, "Date of Release" as ReleaseDate, Gross, IdDirector FROM Movie ORDER BY Name	Estas queries, quando usadas em conjunto, permitem, a partir de uma função, que os usuários pesquisem filmes pelo ano de lançamento (Query 1) e exibem detalhes adicionais sobre todos os filmes, ordenando-os pelo nome (Query 2).
SELECT IdMovie, Name, "Date of Release" as ReleaseDate FROM MOVIE WHERE IdMovie == ? SELECT IdMovie, Name, "Date of Release" as ReleaseDate, Gross, IdDirector FROM Movie ORDER BY IdMovie	Estas queries, quando usadas em conjunto, permitem, a partir de uma função, que os usuários pesquisem filmes pelo ID (Query 1) e exibem detalhes adicionais sobre todos os filmes, ordenando-os pelo ID (Query 2). Este tipo de queries é usado para pesquisas por ID nas tabelas movies, actors, directors, characters e genres.