

Projeto de Bases de Dados (CC2005) - parte 2

Última atualização: 12/12/2023

Objetivo

Neste 2ª parte do projeto pretende-se que concretize o modelo de BD proposto na 1ª parte em uma BD SQLite povoada com dados, e que desenvolva uma aplicação Python simples de interface a essa BD.

Deve considerar os seguintes passos no trabalho:

1. **a definição do esquema de tabelas em SQLite**, em correspondência ao modelo relacional proposto anteriormente (e eventualmente revisto);
2. **o povoamento das tabelas com dados** tendo em conta as fontes de dados que identificou na 1ª parte do projeto;
3. e **o desenvolvimento da aplicação Python** em linha com a aplicação MovieStreamApp discutida nas aulas ([slides das aulas teóricas](#), [código no GitHub](#)).

Deverá entregar até **29 de dezembro** via Moodle os seguintes itens:

1. **relatorio2.docx** : relatório da 2ª parte, seguindo o modelo do documento disponibilizado [aqui](#);
2. **relatorio.docx**: relatório de 1ª parte, **necessário se** comentários ao relatório da 1ª parte implicarem revisões à descrição do universo, diagrama UML, ou modelo relacional;
3. **BaseDados.db** : ficheiro SQLite contendo a sua BD;
4. **app.zip** : ficheiro ZIP contendo a implementação da aplicação Python, em linha com o ambiente de código disponibilizado [aqui](#).

Posteriormente à entrega, cada grupo deverá apresentar também o trabalho em ambiente de laboratório com **apresentações previstas a 18 e 19 janeiro de 2024**.

Seguem-se algumas notas importantes a ter em conta na elaboração desta 2ª parte do projeto.

Definição e povoamento das tabelas

As tabelas definidas deverão corresponder à proposta de modelo relacional feita, incluindo a definição de chaves primárias e externas. Os campos das tabelas devem também ter tipos de dados apropriados aos valores representados.

Deverá povoar as tabelas com dados suficientes para exercitar todos os aspetos do esquema, i.e., deverá ter um número razoável de entradas por cada tabela.

Caso seja útil importar dados de ficheiros CSV via consola sqlite3 ou usando o SQLite Studio, veja este tutorial: "[Import a CSV File Into an SQLite Table](#)".

Aplicação Python

IMPORTANTE: Veja notas de configuração e uso da aplicação no arquivo **README.html** disponibilizado no arquivo **app.zip**.

No mesmo espírito que a MovieStreamApp de que falamos nas aulas teóricas:

1. A aplicação deve ter um "endpoint" para a página de entrada /. A página gerada deve permitir acesso às outras funcionalidades (como na página inicial da MovieStreamApp).
2. Para cada tabela **T** que corresponda a uma classe no universo da BD deverão existir "endpoints":
 - **/T/k/** que liste propriedades de um registo individual em **T** com chave primária **k**;
 - **/T/** que liste todas os registos na tabela **T** e conter links que permitam aceder aos registos individuais ;
3. Implementar 3 outros "endpoints" à sua escolha, envolvendo necessariamente pelo menos uma vez:
 - agregação agrupada de informação;
 - junção de informação envolvendo 3 ou mais tabelas;
 - pesquisa de informação;
4. De forma geral, cada página gerada deverá permitir ligação (conter "links HTML") para acesso a dados de outras tabelas em função dos dados existentes.
5. Deverá empregar instruções SQL parameterizadas para inibir injeção de SQL.