Projecto de BD - parte 2

Bases de Dados (CC2005), DCC / FCUP

Eduardo R. B. Marques, DCC / FCUP

Sumário

Nesta segunda parte do projecto pretende-se que conceba uma aplicação web simples que faça interface com a BD que desenvolveu na primeira parte do projecto.

A aplicação deve ser escrita em Python 3 e usar as bibliotecas Flask e PyMySQL, de acordo com o ambiente de código fornecido.

Requisitos

1. A aplicação deve ter um "endpoint" para a página de entrada `/`. Esta deve permiti uma navegação na BD usando as outras funcionalidades (como na página inicial da aplicação MovieStream).

2. Permitir para pelo menos 3 tabelas `T` as operações de:

    - Listagem de todos os registos ex. `/T/`, em que a página gerada deve conter "links" para aceder a cada registo individual (ver a seguir);
    - Listagem das propriedades de um registo individual em `T`, por ex. `/T/<k>` onde `k` é um valor chave que identifica um registo em `T`. A página gerada deverá permitir ligação para acesso a dados de outras tabelas (ex. como ilustrado para filmes na aplicação MovieStream).

3. Permitir uma operação de pesquisa para pelo menos uma tabela (ex. como ilustrado para filmes no caso da aplicação exemplo).

4. Mostrar resultados de 3 consultas à sua escolha para dados relevantes na sua BD, envolvendo necessariamente pelo menos uma vez:

    - agregação agrupada (i.e. com `GROUP BY`);
    - junções de 3 ou mais tabelas;
    - uma combinação de junções e agregações;

5. Inibir a presença de Bobby Tables, i.e., injeção de SQL!

Nota: O projecto requer apenas a leitura de dados de BD, mas fora do âmbito do projecto pode explorar operações de actualização da BD se quiser (inserções, remoções, actualizações).

Realização e entrega

O trabalho deverá ser realizado pelo mesmo grupo de alunos e ser entregue até ao dia **30 de Dezembro de 2021** inclusive.

Deverá entregar de forma a anunciar posteriormente:

1. **um arquivo ZIP com o código da sua aplicação e também o código SQL da sua BD mesmo que não tenha havido alteração face à parte 1**.
2. **Relatório em formato PDF** onde constem de forma clara:
    - a identificação do grupo;
    - descrição das funcionalidades implementadas na aplicação - não precisa de incluir excertos de código fonte, mas algumas imagens de captura de ecrã poderão ajudar à compreensão do seu trabalho;
    - caso tenha modificado a BD face à primeira parte do projecto, faça um pequeno sumário dessas mudanças.

Desenvolvimento da aplicação

Instalação de software

Precisa de ter o Python 3 e o gestor de pacotes pip instalado. Experimente executar `python3 --version` e `pip3 --version` para saber se já estão instalados. Em caso negativo, pode por exemplo em Ubuntu executar:

```
sudo apt-get install python3 python3-pip
```

Tendo Python 3 e pip instalados, deve instalar as bibliotecas Python `Flask`, `PyMySQL`, e `cryptography` em Python, executando o comando:

```
pip3 install --user Flask==1.1.4 PyMySQL==1.0.2 cryptography==36.0.0
```

Código disponível

Baixe o arquivo bdproj2.zip que contém o esqueleto para a aplicação.

Configuração de acesso à sua BD

Em `bdproj2` edite o ficheiro `db.py` no que se refere à configuração da sua BD, modificando os parâmetros `DB` (nome da base de dados), `USER` (nome do utilizador) e `PASSWORD` (senha do utilizador). Em **computadores dos laboratório do DCC** esses valores terão de ser `guest` para `USER` e `DB`, e `aDammGoodP@ssw0rd` para `PASSWORD`.

Teste o acesso executando:

```
python3 test_db_connection.py NOME_DE_UMA_TABELA
```

Se a configuração do acesso à BD estiver correcto, deverá ser listado o conteúdo da tabela `NOME_DE_UMA_TABELA`, por ex. se a BD configurada for a MovieStream:

```
$ python3 test_db_connection.py REGION
SELECT * FROM REGION
5 results ...
{'RegionId': 6, 'Name': 'Other countries', 'RegionManager': 17}
{'RegionId': 7, 'Name': 'America', 'RegionManager': 16}
{'RegionId': 8, 'Name': 'Asia', 'RegionManager': 15}
{'RegionId': 9, 'Name': 'Europe', 'RegionManager': 17}
{'RegionId': 10, 'Name': 'Africa', 'RegionManager': 15}
```

Execução do servidor da aplicação

Teste agora o servidor executando `python3 server.py`, ex.:

```
$ python3 server.py
2021-05-18 21:40:46 - INFO - Connected to database guest
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
2021-12-08 21:40:46 - INFO -  * Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)
...
```

De seguida abra no seu browser **http://127.0.0.1:9000** ou **http://localhost:9000**. Deverá ver uma página com uma mensagem **Hello World!**.

Programação

A estrutura é similar à da aplicação MovieStream que vimos nas aulas:

- Deve editar o código Python da aplicação em `app.py`.
- Deve colocar as templates de geração de HTML na pasta `templates`.
- Coloque o SQL da sua BD na pasta `sql` com o nome `db.sql`.

Referências

- Aplicações BD com SQL embebido (slides das teóricas)
- MovieStream - aplicação exemplo
- PyMySQL
- Linguagem HTML

- Flask
- Jinja templates