Projeto SQL

Megadados

Introdução

Continuando nosso projeto, vamos agora introduzir o banco de dados usando um ORM.

Entrega

Para entregar esta atividade (handout 02), crie uma **tag** no padrão `v2.x.y` (ex: v2.0.1, v2.2.3). Apenas a última tag `v2.*` será considerada.

Exemplo de comandos:

```
git tag -a v2.0.1 -m "Versão 2.0.1"
git push origin v2.0.1
```

Data de entrega: 05/05/2024

Object-Relational Mapping

Vocês devem ter observado que existem alguns paralelos entre um modelo relacional e uma arquitetura de classes composta de PODs (“Piece-Of-Data”) apenas:

- Uma tabela é similar a uma classe
- Uma coluna de tabela é similar a um campo de uma classe
- Um objeto é similar a uma linha de uma tabela
- Uma chave primária é similar a uma referência (e.g. um ponteiro) para um objeto
- Uma chave estrangeira é similar a um campo em um objeto que referencia outro objeto

Tendo essa observação em mente surgiram várias bibliotecas que permitem criar objetos em nossas aplicações, e que são diretamente mapeados para linhas em tabelas de um banco de dados relacional. Essas bibliotecas são chamadas de bibliotecas *Object-Relational Mapping (ORM)*.

Em Python, uma das bibliotecas ORM mais conhecidas é o SQLAlchemy (https://www.sqlalchemy.org/).

Os ORMs se prestam a uma outra função: abstrair o detalhe de qual banco de dados está sendo usado na aplicação. Vocês devem ter observado também que existem vários SGBDs relacionais e que estes são apenas parcialmente

compatíveis entre si, em termos de características e de especificação da sua variante de SQL. Porém, apesar das incompatibilidades em termos de detalhes, quase todos fazem a mesma coisa – são compatíveis em conceito. Uma biblioteca ORM também serve para abstrair esses detalhes.

Como essa abstração de detalhes é construída? Uma biblioteca ORM define uma maneira de descrever os dados que é independente de SGBD, é feita em nível de aplicação. Internamente, a biblioteca traduz as construções feitas pelo usuário da biblioteca em comandos específicos para o SGBD escolhido pelo usuário. A Figura 1 mostra a arquitetura do SQLAlchemy. Podemos ver que a camada ORM não se comunica diretamente com o SGBD, mas sim com uma abstração do SGBD que é formada pelos componentes “Schema/Types”, “SQL Expression Language” e “Engine”. Esta última componente representa diretamente o SGBD, e é responsável pela comunicação com o SGBD usando bibliotecas específicas para cada um destes (componente “DBAPI”).

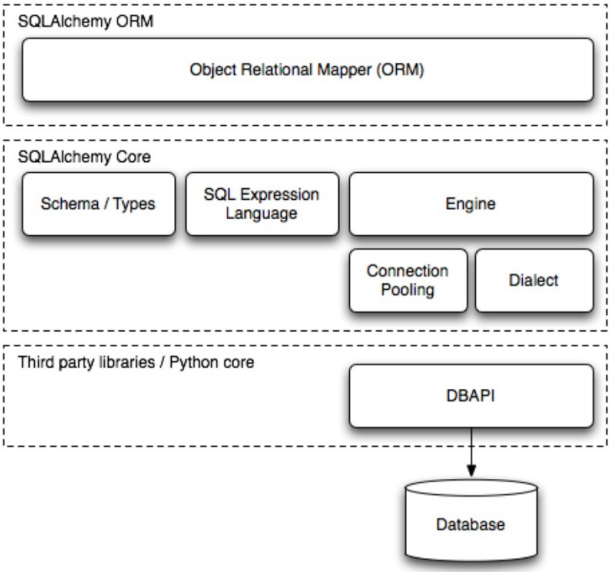

Figura 1: Arquitetura do SQLAlchemy. Fonte: Michael Bayer. SQLAlchemy. In Amy Brown and Greg Wilson, editors, The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks 2012 http://aosabook.org

Atividade

Vamos incorporar o MySQL à nossa aplicação usando o SQLAlchemy.

Tarefa 1

Assista o vídeo “Introduction to SQLAlchemy” (https://www.youtube.com/watch?v=woKYyhLCcnU), que é uma palestra do criador do SQLAlchemy na PyCon 2013 (meio velhinho, mas tá ok)

Sim, é um video de mais de 2 horas. Pega um café, ache uma posição confortável, reserve o tempo para assistir tudo!

Tarefa 2

Faça a sessão de banco de dados do tutorial do FastAPI, se você ainda não a fez.

Tarefa 3 (entregável)

Incorpore o **uso do banco de dados** no projeto usando o **SQLAlchemy**.

ATENÇÃO: Não inclua as credenciais de acesso ao banco de dados diretamente no seu código! Procure uma solução adequada para o gerenciamento de "secrets" no seu projeto, ok? Sugestão: criar um **.env** que deve estar no **.gitignore** e NUNCA ser enviado ao github! Pesquise no google por **.env.example**

Faça a entrega seguindo o padrão de tags proposto.

Rubrica

Conceito	Descrição
I	• Não fez, ou entregou groselha
D	• ORM implementado mas com falhas • Credenciais mal-gerenciadas (e.g. secrets committed no github). • Não deixou no README o Diagrama ER • Não fez um vídeo (deixar link no README) descrevendo e demonstrando as atualizações do handout 02 na API
A	• Atingiu conceito D • ORM implementado sem falhas e credenciais bem gerenciadas

Sim, esta rúbrica é tudo-ou-nada: ou funciona, ou não funciona!