

SQLAlchemy

um tour pelo ORM mais utilizado dentro do ecossistema python

João Vitor Rezende Moura - 1221184773



UNIT - Universidade Tiradentes

Abstract

abstract

Keywords: keywords

1 SQL e Dados Relacionais.

SQL (Structured Query Language) é uma linguagem para fazer consultas em bancos de dados depermite que nós façamos consultas dos dados presentes nesses bancos e nas tabelas que estão presentes dentro desses bancos de dados. Ela é relativamente semelhante entre os **Sistemas de Gerenciamento de Bancos de Dados(SGBDs)**, como: Oracle, MySQL, MariaDB e outros.

Cada um deles entretanto, possui sua característica específica e podem ou não possuir um certo nível de compatibilidade com os outros sistemas. O SQL é uma forma padronizada do uso, criação, e manejoamento dos dados, e ele permite uma maior facilidade também da integridade dos dados e facilita como a consulta desses addos pode ser feita.

O SQL é utilizado em diversas áreas como uma forma de armazenar os dados de forma organizada quando são colhidos e tratados, algumas das áreas as quais o SQL e os SGBDs são:

- Desenvolvimento de Aplicações Web
- Análise de Dados e BI
- Ciência de Dados e mineração de Dados
- Aplicações Móveis
- Educação e Pesquisa
- Gestão de Finanças e Contabilidade

1.1 Componentes de um sistema SQL

Um sistema SQL é composto por vários componentes para permitir a criação, manutenção e consulta de bancos de dados relacionais, esses componentes são:

- **Tabela:** As tabelas são estruturas fundamentais, pois elas funcionam como "pools" de dados, ou seja, armazenam dados em um grupo que é feito de formato tabular, com colunas representando atributos e linhas representando registros.
- **Procedimentos Armazenados:** Os procedimentos armazenados são blocos de código SQL que podem ser definidos e armazenados no banco de dados. Eles são usados para realizar tarefas específicas com cálculos complexos, validações de dados e processamentos personalizados.
- **Instruções:** As instruções servem para manipular os dados, e os comandos SQL permitem que os usuários realizem diversas ações, conforme as ações, e especifiquem a forma como os dados são manipulados por meio do conjunto dessas instruções por meio de determinadas lógicas.

2 Introdução ao SQLAlchemy

2.1 Do motor, ao mapa

O SQLAlchemy é um conjunto de ferramentas SQL e um ORM que podem ser utilizados para trabalhar com bases de dados dentro do python. Seus componentes são divididos em duas estruturas primárias, que é o **core**, onde fica a parte mais direcionada à comunicação e integração com a base de dados, e o **orm**, onde fica a parte dos modelos e das abstrações de dados que podem ser usados para enviar os dados de uma determinada forma ao banco de dados.

Por definição técnica, temos que o **SQLAlchemy Core** é a parte direcionada à arquitetura fundacional do SQLAlchemy, como um conjunto de ferramentas para a base de dados. Ela provê ferramentas que são direcionadas à facilitar a integração com a base de dados, como: ferramentas para gerenciamento de conectividade, interatividade com as queries da base de dados, e permitir a construção de queries SQL diretamente.

Por outro lado, o **SQLAlchemy ORM** é a parte direcionada ao **object relational mapping**, ou seja, provê técnicas dentro do sistema para facilitar a comunicação entre o banco de dados relacional e a linguagem de programação por meio de um "dialetto em comum". O ORM provê também uma camada adicional de configurações que permitem a definição de classes para serem mapeadas dentro das tabelas da base de dados, também como o mecanismo de **sessão**, que permite a persistência de objetos.

Podemos dizer com isso então, que o SQLAlchemy possui duas partes específicas, o motor de funcionamento que possibilita sua comunicação com o banco de dados, e o mapa, que permite a comunicação dos objetos de forma relacional para serem inseridos dentro do banco de dados.

2.2 Estabelecendo Conectividade

A conexão do SQLAlchemy e do banco de dados é feito por meio da **engine**, e esse objeto atua como o a fonte da conexão com o banco de dados em específico. Ela normalmente é um objeto global que é criado somente uma vez em particular para a conexão com o servidor da base de dados, e a sua URL é configurada usando string que descrevem como a conexão deve ocorrer. Um exemplo de como isso ocorre, é mostrado a seguir numa conexão com um banco de dados sqlite3:

```
from sqlalchemy import create_engine
engine = create_engine("sqlite+pysqlite:///memory:", echo="True")
```

A string que é passada dentro do método `create_engine` permite o entendimento de que tipo de base de dados ele está se comunicando, qual driver do SQLAlchemy deve ser usado para fazer a conexão, que pode ser referenciada como **DBAPI**. E permite especificarmos a localização exata de onde fica a base de dados, que nesse caso especificamos que deve ficar dentro da memória.

OBS: O método usado não inicia de fato a conexão com a base de dados a partir do momento o qual ela é criada, ela só é feita a partir do momento em que alguma tarefa lhe é ordenada, usando por tanto o padrão de inicialização tardia.

2.3 Usando a conexão

O objetivo da engine é exclusivamente para o usuário, fornecer a conectividade para a base de dados. Essa conexão é o objeto que podemos usar para interagir com a base e podemos fazer isso com o gerenciador de contexto, como na seguinte forma:

```
from sqlalchemy import text

with engine.connect() as conn:
    result = conn.execute(text("select 'hello world'"))
    print(result.all())
```

2.4 Commitando mudanças