

Explique o query em SQLAlchemy

O ChatGPT disse:

ChatGPT

No SQLAlchemy, a função `query` permite fazer consultas (queries) em tabelas usando classes mapeadas como ORM (Object-Relational Mapper). Através dela, você acessa o banco de dados e executa instruções SQL de forma mais intuitiva, usando Python em vez de SQL diretamente. Aqui estão os conceitos principais:

Estrutura Básica

Quando você faz uma consulta como:

```
python
Copiar código
session.query(Filmes)
```

o SQLAlchemy cria uma consulta para a tabela que está mapeada pela classe `Filmes`, permitindo que você adicione filtros, ordenações e outros parâmetros de pesquisa.

Para explorar o que cada parte faz, vamos analisar a consulta:

```
python
Copiar código
busca_filme = session.query(Filmes).filter(Filmes.ano == ano).first()
```

1. `session.query(Filmes)`

Aqui, `session.query()` é o ponto de partida para criar uma consulta em uma tabela mapeada por uma classe ORM (neste caso, `Filmes`). Esse comando consulta todos os registros da tabela `Filmes`.

2. `.filter(Filmes.ano == ano)`

O método `.filter()` adiciona uma cláusula `WHERE` à consulta SQL, filtrando resultados com base nas condições definidas. Neste caso, `Filmes.ano == ano` filtra para incluir apenas filmes cujo ano (`Filmes.ano`) corresponde ao valor da variável `ano`.

Equivalente SQL

Essa parte da consulta seria equivalente ao `WHERE ano = <valor>` em SQL, onde `<valor>` é substituído pelo valor de `ano`.

3. `.first()` ou `.one()`

Após definir a condição de filtragem, você pode decidir como deseja recuperar os resultados:

- **.first()**: Retorna o primeiro resultado da consulta ou **None** se não encontrar nada.
- **.one()**: Retorna exatamente um resultado e lança exceções se a consulta retornar zero ou mais de um registro.

Exemplo Completo em SQL

A consulta:

```
python
Copiar código
busca_filme = session.query(Filmes).filter(Filmes.ano == ano).first()
```

corresponde ao SQL:

```
sql
Copiar código
SELECT * FROM filmes WHERE ano = <valor> LIMIT 1;
```

E se usarmos **.one()**, a consulta corresponde a:

```
sql
Copiar código
SELECT * FROM filmes WHERE ano = <valor>;
```

Nesse caso, o SQLAlchemy espera apenas um registro de resultado.

Outros Exemplos Úteis

- **Ordenação:** Para ordenar, use **.order_by()**, como **session.query(Filmes).order_by(Filmes.titulo).**
- **Selecionar Colunas Específicas:** Para selecionar colunas específicas, passe-as como argumentos, como **session.query(Filmes.titulo, Filmes.ano).**
- **Limite de Resultados:** Para limitar o número de registros, use **.limit()**, como **session.query(Filmes).limit(5).**

Resumo

O método **query** no SQLAlchemy permite fazer consultas de forma declarativa e conveniente, usando métodos como **.filter()**, **.order_by()**, **.first()**, e **.one()** para controlar e refinar o que será retornado do banco de dados.