

SQLAlchemy (ASSÍNCRONO)

1. `busca_filme`

- É uma variável que provavelmente contém o resultado de uma consulta ao banco de dados.
- Geralmente é um objeto do tipo `Result` retornado por um comando de consulta no SQLAlchemy, como:

```
python
Copiar código
busca_filme = session.execute(select(Filme).where(Filme.titulo
== "Título do Filme"))
```

2. `scalars()`

- O método `scalars()` é usado para extrair as colunas ou objetos específicos de um tipo de dado do resultado da consulta.
- Quando você faz uma consulta que retorna múltiplas colunas ou um conjunto de resultados, o `scalars()` filtra e retorna apenas a coluna ou o tipo de dado principal (como uma classe ORM).
- Exemplo: Se você consulta uma tabela que retorna objetos do tipo `Filme`, o `scalars()` retorna uma sequência desses objetos ORM.

3. `all()`

- O método `all()` converte o resultado da consulta em uma lista contendo todos os elementos retornados.
- Ele executa a consulta ao banco de dados e retorna os resultados em uma lista Python.

O fluxo completo:

Se essa linha faz parte de um código que busca filmes em um banco de dados, ela está:

1. **Executando a consulta.**
2. **Filtrando os resultados para retornar apenas os objetos desejados.**
3. **Convertendo-os para uma lista.**

Exemplo completo:

```
python
Copiar código
from sqlalchemy import select

# Supondo que 'Filme' seja um modelo ORM e 'session' seja a sessão do
SQLAlchemy
busca_filme = session.execute(select(Filme).where(Filme.categoria ==
"Ação"))
filmes = busca_filme.scalars().all()

for filme in filmes:
```

```
print(filme.titulo)
```

Resultado:

- A variável `filmes` conterá uma lista de objetos `Filme` cujas categorias sejam "Ação".

4o

Você disse:

`scalars()` tem outros métodos?

O ChatGPT disse:

ChatGPT

O método `scalars()` retorna um objeto do tipo `ScalarResult` no SQLAlchemy, que tem alguns métodos específicos para manipular ou acessar os dados resultantes. Vamos explorar os métodos principais disponíveis:

1. `all()`

- Retorna todos os resultados como uma lista.
- **Exemplo:**

```
python
Copiar código
resultados = busca_filme.scalars().all()
```

2. `first()`

- Retorna o **primeiro** resultado ou `None` se não houver nenhum.
- Útil quando você espera apenas um único item ou precisa do primeiro elemento.
- **Exemplo:**

```
python
Copiar código
primeiro_filme = busca_filme.scalars().first()
```

3. `one()`

- Retorna **exatamente um** resultado.
- Levanta uma exceção:
 - `NoResultFound`: Se não houver resultados.
 - `MultipleResultsFound`: Se houver mais de um resultado.

- **Exemplo:**

```
python
Copiar código
filme_unico = busca_filme.scalars().one()
```

4. `one_or_none()`

- Retorna **exatamente um** resultado ou `None` se não houver nenhum.
- Levanta uma exceção `MultipleResultsFound` se mais de um resultado for encontrado.
- **Exemplo:**

```
python
Copiar código
filme_unico_ou_none = busca_filme.scalars().one_or_none()
```

5. `unique()`

- Garante que os resultados sejam únicos, removendo duplicatas.
- Retorna o mesmo objeto `ScalarResult`, mas com duplicatas filtradas.
- **Exemplo:**

```
python
Copiar código
resultados_unicos = busca_filme.scalars().unique().all()
```

6. `partitions(size)`

- Retorna um iterador que divide os resultados em partes (partições) do tamanho especificado.
- Útil para processar grandes quantidades de dados sem carregá-los todos de uma vez na memória.
- **Exemplo:**

```
python
Copiar código
for batch in busca_filme.scalars().partitions(10):
    for filme in batch:
        print(filme.titulo)
```

Resumo de Usos Comuns:

Método	O que faz	Quando usar
<code>all()</code>	Retorna todos os resultados como uma lista.	Quando você quer todos os dados.
<code>first()</code>	Retorna o primeiro resultado ou <code>None</code> .	Quando precisa apenas do primeiro elemento ou para verificar se há resultados.
<code>one()</code>	Retorna exatamente um resultado. Levanta exceção se houver 0 ou mais de 1 resultado.	Quando você espera exatamente um item e quer garantir isso.
<code>one_or_none()</code>	Retorna um resultado ou <code>None</code> . Levanta exceção se houver mais de um.	Quando você espera 0 ou 1 resultado.
<code>unique()</code>	Filtra duplicatas dos resultados.	Quando pode haver duplicatas nos resultados.
<code>partitions(size)</code>	Retorna resultados em lotes de tamanho especificado.	Para processar grandes conjuntos de dados de forma eficiente.

Esses métodos tornam o `scalars()` muito versátil para acessar e manipular os resultados de consultas.