

No contexto do SQLAlchemy, o parâmetro `lazy` na função `relationship` controla o momento em que as associações de objetos relacionados são carregadas do banco de dados. O valor `'joined'` indica que os dados relacionados devem ser carregados usando uma *join* SQL no momento da consulta original, em vez de uma consulta separada (como aconteceria no caso de um carregamento "lazy").

Explicando os valores mais comuns de `lazy` :

- `'select'` (padrão): Carrega o relacionamento apenas quando é acessado pela primeira vez, usando uma consulta separada. Esse é o comportamento "lazy loading" típico.
- `'joined'` : Carrega o relacionamento imediatamente, fazendo um *join* na consulta original, reduzindo o número de consultas quando os dados relacionados são frequentemente necessários.
- `'subquery'` : Similar a `'joined'` , mas usa uma subconsulta para carregar o relacionamento, o que pode ser útil para algumas consultas mais complexas ou para otimização.
- `'dynamic'` : Cria um objeto de consulta em vez de carregar dados imediatamente. Isso é útil quando você quer uma consulta mais complexa ou paginada no relacionamento.

Exemplo com `lazy='joined'` :

No seu caso, o uso de `lazy='joined'` em:

```
python
```

 Copiar código

```
relationship('Conservante', secondary=conservantes_picole, backref='conservante',  
lazy='joined')
```

significa que, quando um objeto principal é carregado, os objetos de `Conservante` associados serão carregados ao mesmo tempo, usando uma única *join* SQL. Isso melhora o desempenho em consultas que sempre precisam desses dados relacionados, evitando múltiplas consultas separadas para cada relação.