






07. Evitando queries N + 1

 Date	@15/12/2022
 Categoria	Java e persistência
 Curso	Java e JDBC: trabalhando com um banco de dados

Tópicos

- Projeto da aula anterior
- Relacionamento entre tabelas
- Vantagens da chave estrangeira
- Modelo e o DAO de categorias
- Queries N + 1
- Por que evitar queries N + 1?
- Usando Inner Join
- Informações relacionadas
- Faça que eu fiz
- O que aprendemos?

Vantagens da chave estrangeira

Gabriela, para categorizar os produtos, resolveu criar mais uma coluna na tabela, do tipo `String`, e chamou de `Categoria`. Assim foi possível, no momento da inserção de um produto, já informar qual a sua categoria. Qual a desvantagem dessa abordagem?

- Não daria para padronizar a inserção desta categoria, deixando livre para as pessoas escreverem como quisessem, podendo prejudicar um possível filtro dos produtos por determinada categoria. Como cada um poderia escrever como

bem entendesse, poderia haver diferenciação no nome das categorias (uns podem usar acentos, outros não, por exemplo) e os resultados dos *select's*, filtrados por categoria, estariam errados.

Por que evitar queries N + 1:

Qual é o problema das queries **N + 1** ?

- Utilizam várias *queries*, aumentando a quantidade de acesso ao banco de dados. Quando estamos falando de consultas simples, pode-se não ver o problema, mas conforme a complexidade vai aumentando e precisa-se trazer informações de várias tabelas, o acesso aumenta exponencialmente, atrapalhando a performance do banco de dados.
-

Informações relacionadas

José relacionou suas tabelas, no banco de dados, usando chave estrangeira. Sua dúvida agora é, o que precisa ser feito para trazer as informações relacionadas das duas tabelas pela aplicação?

- José precisará utilizar o *Inner Join*.
 - Com *Inner Join*, José conseguirá trazer as informações, que se relacionam entre si, das duas tabelas.
-

O que aprendemos?

Nesta aula, aprendemos:

- Que quando temos um relacionamento, é preciso ter cuidado para não cair no problema de *queries*

N + 1

- **N + 1** significa executar uma *query* e mais uma nova *query* (**N**) para cada relacionamento
- Queries **N + 1** podem gerar um problema no desempenho
- Queries **N + 1** podem ser evitadas através de *joins* no SQL

- A criar a nossa própria camada de persistência
-