

≡ 09

## Vários selects

Com o que fizemos até agora, o `cache` já está habilitado. Ao acessar a `index` uma vez e verificar o console, vemos que as duas consultas estão lá como esperado. Porém no segundo acesso, vemos um número ainda maior de selects, dez no total, algo similar a:

```
21:32:44,091 INFO [stdout] (default task-42) Hibernate:
21:32:44,091 INFO [stdout] (default task-42)      select
21:32:44,091 INFO [stdout] (default task-42)          livro0_.id as id1_2_0_,
21:32:44,091 INFO [stdout] (default task-42)          livro0_.capaPath as capaPath2_2_
21:32:44,093 INFO [stdout] (default task-42)          livro0_.dataPublicacao as dataPu
21:32:44,093 INFO [stdout] (default task-42)          livro0_.descricao as descricao4_2
21:32:44,093 INFO [stdout] (default task-42)          livro0_.numeroPaginas as numeroP
21:32:44,093 INFO [stdout] (default task-42)          livro0_.preco as preco6_2_0_,
21:32:44,093 INFO [stdout] (default task-42)          livro0_.titulo as titulo7_2_0_
21:32:44,093 INFO [stdout] (default task-42)      from
21:32:44,093 INFO [stdout] (default task-42)          Livro livro0_
21:32:44,093 INFO [stdout] (default task-42)      where
21:32:44,093 INFO [stdout] (default task-42)          livro0_.id=?
```

Mas o que esperamos é que não seja feito nenhum select. Por que acabamos com um número maior de selects? Como resolver?

Modifique também seu código para resolver o problema.

*Selecione uma alternativa*

**A**

Isso ocorre por que indicamos no `persistence.xml` que queremos que o `cache` seja habilitado apenas de maneira seletiva, e dessa forma a classe `Livro` foram mantidos apenas o `id` dos objetos carregados do banco de dados.

Para resolver isso, temos que utilizar a opção `ALL`:

```
<shared-cache-mode>ENABLE_SELECTIVE</shared-cache-mode>
```

Dessa forma todos os campos da entidade serão cacheados.

**B**

Foi armazenado apenas o `id` da entidade no `cache`, e por conta disso, o Hibernate irá sempre verificar se ele possui cacheado o objeto com aquele `id`. Como o objeto não é encontrado, é feito um select buscando o registro do banco que possui aquele `id`.

A solução é adicionar a anotação `@Cacheable` na classe `Livro`, indicando que toda a entidade deve ser cacheada.

**C**

Foi armazenado apenas o `id` da entidade no `cache`, e por conta disso, o Hibernate irá sempre verificar se ele possui cacheado o objeto com aquele `id`. Como o objeto não é encontrado, é feito um select buscando o registro do banco que possui aquele `id`.

A solução é indicar no `persistence.xml`, por meio da tag `hibernate.cache.class`, que os objetos da entidade `Livro` devem ter todos os campos cacheados:

```
<hibernate.cache.class>br.com.casadocodigo.loja.models.Livro</hibernate.cache.clas
```

