

11

Cache de queries

Transcrição

Voltando a página inicial da nossa loja podemos utilizar o filtro que desenvolvemos. A intenção é buscar por produtos que possuem a categoria **tecnologia**. Ao realizarmos essa busca são disparados **três queries**, uma para buscar os produtos e duas para buscar a loja de cada produto (do relacionamento @ManyToOne). Será que podemos evitar que essas três queries sejam disparadas toda vez que realizarmos a busca? O que precisamos é armazenar em cache o resultado de uma query feita para determinados parâmetros e, com isso, poderemos buscar no cache ao em vez de ir diretamente ao banco.

Para isso, usamos o **cache de queries** - desligado por padrão. Para ligá-lo, basta adicionar mais uma propriedade na configuração da JPA:

```
public class JpaConfigurator {
    ...
    @Bean
    public LocalContainerEntityManagerFactoryBean getEntityManagerFactory(DataSource dataSource) {
        ...
        props.setProperty("hibernate.cache.use_second_level_cache", "true");
        props.setProperty("hibernate.cache.use_query_cache", "true");
        props.setProperty("hibernate.cache.region.factory_class", "org.hibernate.cache.ehcache.EhCacheRegionFactory");
        ...
    }
}
```



Depois de ativado, precisamos avisar ao Hibernate que queremos que ele salve no cache os resultados de alguma query. E fazemos isso passando uma dica (*hint*) na própria query através do método `setHint` da interface `Query`:

```
query.setHint("org.hibernate.cacheable", "true");
```

Passamos essa dica na query que buscará os produtos com filtro. No método `getProdutos` do `ProdutoDao`:

```
public List<Produto> getProdutos(String nome, Integer categoriaId, Integer lojaId) {
    ...
    TypedQuery<Produto> typedQuery = em.createQuery(query.where(conjuncao));
    typedQuery.setHint("org.hibernate.cacheable", "true");

    return typedQuery.getResultList();
}
```

Vamos testar! Buscaremos pelos produtos de categoria `Tecnologia`. Ao atualizarmos a página, nesse caso, apenas dois selects foram feitos em busca das lojas de cada produto. O que aconteceu? Lembra do problema que tivemos com as categorias? Sim, o cache de queries também **retorna apenas os ids** e por isso será necessário buscar as lojas desses id's.

Sabemos que a query só é disparada quando o objeto não é encontrado no cache, o que devemos fazer? Pedir para o Hibernate armazenar a loja no Cache!

```
@Entity  
@Cache(usage=CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)  
public class Loja {
```

Vamos buscar novamente e atualizar a página logo em seguida. Nesse caso **nenhuma** query será disparada, já que os objetos que possuem aqueles **ids** foram encontrados no cache :)