



Desenvolvimento para Servidores-II

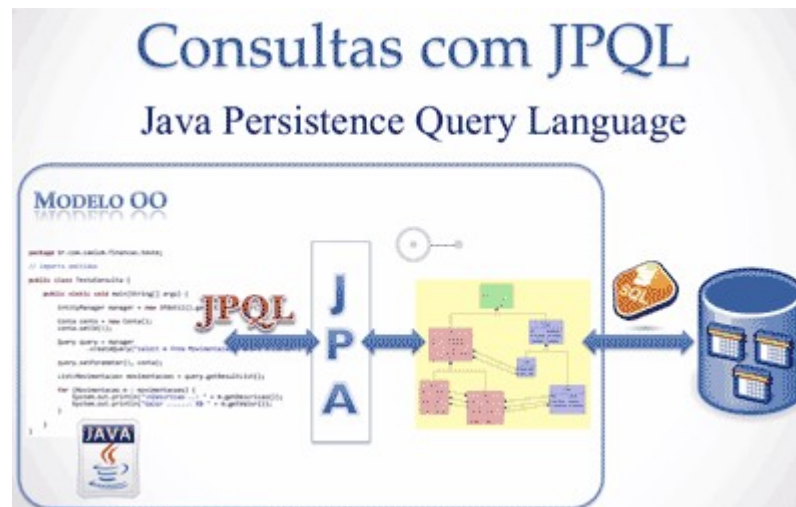
JPQL (Java Persistence Query Language)

Neste tópico abordaremos o JPQL, uma linguagem de consulta semelhante a SQL

Prof. Ciro Cirne Trindade

- Por padrão, todo banco de dados relacional aceita a SQL para realizar consultas e manipulações em modelos relacionais
- Quando adotamos o JPA, uma das coisas que buscamos é nos distanciar o máximo possível do modelo relacional e focar exclusivamente no modelo orientado a objetos e seus estados

- O JPQL é uma linguagem de consulta, assim como o SQL, porém orientada a objetos
- Isso significa que quando estivermos pesquisando dados, não consideramos nomes de tabelas ou colunas, e sim entidades e seus atributos



Criando e executando queries

- Um consulta em JPQL é representada pela interface `Query`
- O método `createQuery()` do `EntityManager` é usado para criar um objeto do tipo `Query`
- `createQuery()` espera como parâmetro a consulta JPQL
- Para recuperar o resultado da consulta, utilizamos o método `getResultList()` da interface `Query`

Exemplo de uma query

```
package finanzas.teste;
// imports omitidos

public class TestaConsulta {
    public static void main(String[] args) {
        EntityManager manager =
            new JPAUtil().getEntityManager();
        Conta conta = new Conta();
        conta.setId(2);
        Query query = manager.createQuery("select m from Movimentacao
m where m.conta.id=" + conta.getId());
        List<Movimentacao> movimentacoes =
            query.getResultList();
        for (Movimentacao m : movimentacoes) {
            System.out.println("\nDescricao ...: " +
m.getDescricao());
            System.out.printf("Valor .....: R$ %.2f\n",
m.getValor());
        }
        manager.close();
    }
}
```

Parâmetros de queries

- Queries podem ter parâmetros semelhantes ao `PreparedStatement`
- Os parâmetros podem ser

- Posicionais

```
Query query = manager.createQuery("select m  
from Movimentacao m where m.conta=?1");  
query.setParameter(1, conta);
```

- Nomeados

```
Query query = manager.createQuery("select m  
from Movimentacao m where  
m.conta=:pConta");  
query.setParameter("pConta", conta);
```

Consultas nomeadas (*named queries*)

- Uma consulta nomeada (*named query*) é uma consulta predefinida que criamos e associamos a uma entidade
- Na implantação da aplicação as consultas nomeadas são armazenadas no `EntityManager`
- Em tempo de execução nós podemos usar o `EntityManager` para recuperar, configurar e executar uma consulta nomeada

A anotação @NamedQuery

- A anotação @NamedQuery possui 2 elementos obrigatórios:
 - name: nome da consulta
 - query: string com a consulta JPQL
- Por exemplo, para definir uma consulta para devolver todas as contas cadastradas faríamos:

```
@Entity
@NamedQuery(name="Conta.listaTodas",
            query="select c from Conta c")
public class Conta implements Serializable {
    ...
}
```


A anotação `@NamedQueries`

- Para associar múltiplas consultas nomeadas a uma entidade é necessário empacotá-las em uma anotação `@NamedQueries`
- Por exemplo:

```
@Entity
@NamedQueries ({
    @NamedQuery (name="Conta.listaTodas",
                  query="select c from Conta c"),
    @NamedQuery (name="Conta.consultaPorNome",
                  query = "select c from Conta c where
c.titular=?1")
})
public class Conta implements Serializable {
    ...
}
```

Usando as consultas nomeadas

- O método `createNamedQuery()` do `EntityManager` é utilizado para recuperar uma consulta nomeada
- Por exemplo:

```
Query query =  
    manager.createNamedQuery(  
        "Conta.listaTodas");  
List<Conta> contas =  
    query.getResultList();
```

Consulta genérica

```
public List<T> listarGenerico(String query,
                               Object ... params) {
    manager =
        JPAEntityManager.getEntityManager();
    Query q =
        manager.createNamedQuery(query);
    for (int i = 0; i < params.length; i++) {
        q.setParameter(i + 1, params[i]);
    }
    List<T> todos = q.getResultList();
    manager.close();
    return todos;
}
```

Referências

- GEARY, David; HORSTMANN, Cay. *Core JavaServer Faces*. 3. ed., Prentice-Hall, 2010.
- ORACLE Corporation. *The Java EE 7 Tutorial*. Disponível em: <https://docs.oracle.com/javaee/7/JEETT.pdf>, 2014.