

CONSULTAS

Hibernate tiene el objeto Query que nos da acceso a todas las funcionalidades para poder leer objetos desde la base de datos.

Dependiendo de la consulta realizada, existen distintos tipos de retorno.

- **Lista de Objetos**

El método list() retorna una lista con todos los objetos que ha retornado la consulta. En caso de que no se encuentre ningún resultado se retornará una lista sin ningún elemento.

```
Query q1 = s.createQuery("FROM Profesor");
List<Profesor> listaP = q1.list();
for (Profesor profesor : profesores)
    System.out.println(profesor.toString());
```

O puede ser:

```
for (int i=0;i<listaP.size();i++)
    System.out.println(listaP.get(i).toString());
```

- **Lista de Array de Objetos**

Es posible que las consultas retornen datos escalares en vez de clases completas, por ejemplo, que retorne únicamente el código del profesor y su nombre.

En estos casos el método list() retorna una lista con una Array de objetos con tantos elementos como propiedades se hayan puesto en la Select.

```
Query q2 = s.createQuery("Select p.id,p.nombre FROM Profesor p");
List<Object[]> listaP = q2.list();
for (Object[] datos : listaP)
    System.out.println(datos[0] + "--" + datos[1]);
```

O podría ser

```
for (int i=0;i<listaP.size();i++)
    System.out.println(listaP.get(i)[0] + "--" + listaP.get(i)[1]);
```

- **Lista de Objeto**

Otro caso es cuando se selecciona una única columna de datos escalares en la Select. En este caso, como el array a retornar dentro de la lista solo tendrá un elemento, no se retorna una lista de array de objetos, sino únicamente una lista de elementos List<Object>.

```
Query q3 = s.createQuery("SELECT p.nombre FROM Profesor p");
List<Object> listaP = q3.list();
for (Object datos : listaP)
    System.out.println(datos);
```

O podría ser

```
for (int i=0;i<listaP.size();i++)  
    System.out.println(listaP.get(i));
```

- **UniqueResult**

Muchas veces una consulta retorna cero o un resultado, por lo que es poco práctico que retorne una lista con un único elemento. Para facilitar dicha tarea Hibernate dispone del método **uniqueResult()**.

Este método retornará directamente el único objeto que ha obtenido la consulta. En caso de que no encuentre ninguno se retornará null.

```
Profesor pr=(Profesor)s.createQuery("SELECT p FROM Profesor p  
                                   WHERE p.id=1000").uniqueResult();  
if(pr==null)  
    System.out.println("NO EXISTE");  
else  
    System.out.println("Profesor con Id 1000="+pr.getNombre());
```

Consultas con nombre

Las consultas a la base de datos no deberían escribirse directamente en el código sino que deberían estar en un fichero externo para que puedan ser modificadas fácilmente.

Hibernate tiene una funcionalidad para hacer esto de una forma sencilla y es añadir la etiqueta `<query>`.

En el fichero de mapeo se incluirá la etiqueta `<query>` con la consulta HQL que se desea ejecutar.

Se define una query en el fichero Profesor.hbm.xml.

```
<hibernate-mapping>  
    <class name="POJOS.Profesor" table="profesor" catalog="universidad"  
                                optimistic-lock="version">  
        <id name="id" type="int">  
            <column name="Id" />  
            <generator class="assigned" />  
        </id>  
        <many-to-one name="especialidad" class="POJOS.Especialidad"  
                                fetch="select">  
            <column name="cod" />  
        </many-to-one>  
  
        <property name="nombre" type="string">  
            <column name="Nombre" length="20" />  
        </property>  
        <property name="apellidos" type="string">  
            <column name="Apellidos" length="30" />  
        </property>  
    </class>
```

```
<query name="todoslosProfesores">
    <![CDATA[FROM Profesor]]>
</query>
</hibernate-mapping>
```

Posteriormente, para hacer uso de la consulta se usa el método `getNamedQuery()` en vez de `createQuery()` para obtener el objeto `Query`. El método `getNamedQuery()` recibe como parámetro un `String` que es el nombre de la consulta definida en la etiqueta `query`.

```
Query q = s.getNamedQuery("todoslosProfesores");
List<Profesor> profesores = q.list();
for (int i=0;i<profesores.size();i++)
    System.out.println(profesores.get(i).toString());
```