Consultas

Introducción

- Hibernate soporta un lenguaje de consulta orientado a objetos denominado **HQL** (Hibernate Query Language)
- Las consultas HQL y SQL son representadas con una instancia de **org.hibernate.Query.**
- La interfaz **Query** ofrece métodos para ligar parámetros, manejo del conjunto resultado y para la ejecución de la consulta real.
- Siempre se obtiene una Query utilizando el objeto **Session** actual.

Algunos métodos de Query

- **Iterator iterate()** → Devuelve en un objeto Iterator el resultado de la consulta
- **List list()** → Devuelve el resultado de la consulta en un List
- **Query setFetchSize (int size)** → Fija el número de resultados a recuperar en cad acceso a la base de datos al valor indicado en size
- **int executeUpdate()** → Ejecuta la sentencia de modificación o borrado. Devuelve el número de entidades afectadas
- **String getQueryString()** → Devuelve la consulta en un string
- **Object uniqueResult()** → Devuelve un objeto (cuando sabemos que la consulta devuelve un objeto) o nulo si la consulta no devuelve resultados
- **Query setCharacter(int posición, char valor)** o también **Query setCharacter(String nombre, char valor)** → Asigna el valor indicado en el método a un parámetro de tipo char. Posición indica la posición del parámetro empezando en 0. Nombre es el nombre (:nombre) del parámetro

- **Query setDate (int posición, Date fecha)** o también, **Query setDate (String nombre, Date fecha)** → Asigna la fecha a un parámetro de tipo DATE
- **Query setDouble (int posición, double valor)** o también **Query setInteger (String nombre, double valor)** → Asigna valor a un parámetro de tipo FLOAT
- **Query setInteger(int posición, String valor)** o también **Query setInteger(String nombre, String valor)** → Asigna valor a un parámetro de tipo entero
- **Query setString(int posición, String valor)** o también **Query setString(String nombre, String valor)** → Asigna valor a un parámetro de tipo VARCHAR

- **Query setParameterList (String nombre, Collection valores)** → Asigna una colección de valores al parámetro cuyo nombre se indica en nombre
- **Query setParameter (int posicion, Object valor)** → Asigna el valor al parámetro indicado en posición
- **Query setParameter (String nombre, Object valor)** → Asigna el valor al parámetro indicado en nombre
- **int executeUpdate()** → Ejecuta una sentencia UPDATE o DELETE, devuelve el número de entidades afectadas.
- API de Hibernate → Enlace

Primera consulta

- Para realizar una consulta usaremos el método **createQuery()** de la interfaz **SharedSessionContract**, se le pasará en un String la consulta HQL
- Por ejemplo, para hacer una consulta sobre la tabla depart, mapeada con la clase Depart, se escribe:

```
Query q = session.createQuery(“from Depart”);
```

- Para recuperar los datos de la consulta usaremos **list():**

```
List <Depart> lista = q.list();
```

- El método **list()** devuelve una colección de todos los resultados de la consulta. En la colección se encuentran instanciadas todas las entidades que corresponden al resultado de la ejecución de la consulta.
- Si la cantidad de resultados es extensa, el retraso del acceso a la bbdd será notorio.
- El método **iterate()** devuelve un iterador Java para recuperar los resultados de la consulta. En este caso Hibernate ejecuta la consulta obteniendo solo los ids de las entidades, y en cada llamada al método Iterator.next() ejecuta la consulta propia para obtener la entidad completa.

Ejemplo: filas de la tabla depart

```
import primero.*;
import java.util.Iterator;
import java.util.List;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
public class Main_emple {
	public static void main(String[] args){
		SessionFactory sesion = HibernateUtil.getSessionFactory();
		Session session = sesion.openSession();
		Query q = session.createQuery("from Depart");
		List <Depart> lista =q.list();
		Iterator <Depart> iter = lista.iterator();
		System.out.printf("Número de departamentos: %d%n", lista.size());
		while (iter.hasNext()){
			Depart dep = (Depart) iter.next();
			System.out.printf("%d, %s%n", dep.getDeptNo(), dep.getDnombre());
		}
		session.close();
		System.exit(0);
	}
}
```

- El ejemplo anterior con el método Iterator quedaría:

```
Query q = session.createQuery(“from Depart”);
q.setFetchSize(10);
Iterator iter = q.iterate();
while (iter.hasNext()){
                        Depart dep = (Depart) iter.next();
                        System.out.printf("%d, %s%n", dep.getDeptNo(), dep.getDnombre());
                }
```

problemas Hibernate_V_1

1. Visualiza el apellido y salario de los empleados del departamento 20. (Ayuda: la consulta HQL necesaria es: “from Emple as e where e.depart.deptNo=20”)
2. Visualiza los datos del señor “ARROYO”
3. Visualiza los nombres de los empleados junto con el departamento al que pertenecen. (Ayuda: consulta en el manual de HQL cómo realizar un JOIN. Observarás que es muy parecido a SQL)
4. Calcula el salario medio de todos los empleados. (Ayuda: busca en el manual de HQL cómo utilizar las funciones de agregación. Verás que es lo mismo que en SQL)
5. Muestra el salario medio y el número de empleados por departamento