Práctica No. 4. Operaciones Básicas con Hibernate – Selección

En esta práctica se trabajarán las actividades para realizar selecciones de la base de datos

Preparación del Entorno

- Abrir el entorno de desarrollo *Eclipse*
- Descargar de
- la página web http://academicos.azc.uam.mx/jfg/pags/tarea_taller_web.html el archivo ***Practica04Base.zip*** que ya contiene las clases básicas, archivos de configuración y mapeo
-

Preparación de la Base de Datos

- Descargar de la página web http://academicos.azc.uam.mx/jfg/pags/tarea_taller_web.html del bloque de **Práctica 4** el archivo ***script_01_generacion_llenado.sql*** el cuál contiene la base de datos y las tablas a utilizar.
- Descargar de la página web http://academicos.azc.uam.mx/jfg/pags/tarea_taller_web.html del bloque **Práctica 4** el archivo ***script_llenado_practica_04.sql*** el cuál contiene los datos a utilizar.
- Abrir el entorno *MySQL Workbench* y ejecutar los *script* para crear la base de datos, las tablas y los datos a utilizar como prueba.

Para facilitar la impresión de los datos, se agregarán los métodos ***toString()*** a cada una de las clases *beans*. Estos métodos no afectan en los procesos de inserción, actualización o consulta.

Uea.java

```
public String toString(){
            String mensaje = "";
            mensaje = claveUea+","+nombre+","+creditos;
            return mensaje;
      }
```

Licenciatura.java

```
public String toString(){
            String mensaje = "";
            mensaje = clave+","+nombre;
            return mensaje;
      }
```

Alumno.java

```
public String toString(){
            String mensaje = "";
            mensaje = matricula+","+nombre+","+primerApellido+","+
            segundoApellido+","+licenciatura;
            return mensaje;
      }
```

AlumnoUea.java

```
public String toString(){
           String mensaje = "";
           mensaje = idAlumnoUea+","+trimestre+","+calificacion+
                      ","+calificacionNumerica+","+alumnoMatricula+
                      ","+ueaClaveUea;
           return mensaje;

     }
```

Selección por Llave Primaria (PK)

Se realizará la selección más simple, con una búsqueda utilizando llave primaria (PK). Se creará la clase **SeleccionesPK** en el paquete **hibernate.operaciones**.

La selección con llave primaria es igual que la utilizada para la actualización.

```
objeto = sesion.get(Clase.class, clave);
```

En donde clave contiene el valor que identifica a un único objeto en la base de datos, en este caso la **PK** buscada. Como primer prueba, se obtendrá una licenciatura.

SeleccionesPK.java

```
package hibernate.operaciones;

import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

import hibernate.beans.Alumno;
import hibernate.beans.Licenciatura;
import hibernate.beans.Uea;
import hibernate.conexion.CrearConexion;

public class SeleccionesPK {

     private final static Logger log =
Logger.getLogger(SeleccionesPK.class);


     public Licenciatura consultarLicenciatura(String clave){

           PropertyConfigurator.configure("logger.properties");
           Licenciatura licenciaturaConsultada;

           Session sesion = null;
           try{
```

```
                sesion = CrearConexion.getSessionFactory().openSession();
            }catch(ExceptionInInitializerError ex){
                System.err.println("No se pudo crear la sesion ");
               throw new ExceptionInInitializerError(ex);
            }

            log.info("CONSULTANDO LICENCIATURA CON PK " + clave);
            licenciaturaConsultada = sesion.get(Licenciatura.class, clave);


            return licenciaturaConsultada;
      }
}
```

Principal.java

```
package hibernate.principal;

import hibernate.beans.Licenciatura;
import hibernate.conexion.CrearConexion;
import hibernate.operaciones.SeleccionesPK;

public class Principal {

      public static void main(String[] args) {
            consultarLicenciatura("101101");
            CrearConexion.closeSessionFactory();
      }

      private static void consultarLicenciatura(String clave){

            SeleccionesPK operaciones = new SeleccionesPK();

            Licenciatura recuperada =
operaciones.consultarLicenciatura(clave);

            if(recuperada != null){
                System.out.println(recuperada.toString());
            }else{
                System.out.println("No se encuentra la clave buscada");
            }

      }

}
```

De manera similar, se crearán métodos para consultar Uea y Alumno, todo basado en una llave primaria. Para la consulta de Uea se tiene:

SeleccionesPK.java

```
public Uea consultarUea(String claveUea){

            PropertyConfigurator.configure("logger.properties");
            Uea ueaConsultada;

            Session sesion = null;
            try{
                  sesion = CrearConexion.getSessionFactory().openSession();
            }catch(ExceptionInInitializerError ex){
                  System.err.println("No se pudo crear la sesion ");
                throw new ExceptionInInitializerError(ex);
            }

            log.info("CONSULTANDO UEA CON CLAVE " + claveUea);
            ueaConsultada = sesion.get(Uea.class, claveUea);

            return ueaConsultada;
      }
```

Principal.java

```
public static void main(String[] args) {

            consultarLicenciatura("101101");
            consultarUea("115102");
            CrearConexion.closeSessionFactory();
}

private static void consultarUea(String claveUea){

            SeleccionesPK operaciones = new SeleccionesPK();

            Uea recuperada = operaciones.consultarUea(claveUea);

            if(recuperada != null){
                  System.out.println(recuperada.toString());
            }else{
                  System.out.println("No se encuentra la clave buscada");
            }

      }
```

Para consultar un Alumno, se tiene:

SeleccionesPK.java

```
public Alumno consultarAlumno(String matricula){

            PropertyConfigurator.configure("logger.properties");
            Alumno alumnoConsultado;

            Session sesion = null;
            try{
                  sesion = CrearConexion.getSessionFactory().openSession();
            }catch(ExceptionInInitializerError ex){
                  System.err.println("No se pudo crear la sesion ");
                throw new ExceptionInInitializerError(ex);
            }

            log.info("CONSULTANDO ALUMNO CON MATRICULA " + matricula);
            alumnoConsultado = sesion.get(Alumno.class, matricula);

            return alumnoConsultado;
      }
```

Principal.java

```
public static void main(String[] args) {

            consultarLicenciatura("101101");
            consultarUea("115102");
            consultarAlumno("23456");
            CrearConexion.closeSessionFactory();
      }
private static void consultarAlumno(String matricula){

      SeleccionesPK operaciones = new SeleccionesPK();

      Alumno recuperado = operaciones.consultarAlumno(matricula);

      if(recuperado != null){
            System.out.println(recuperado.toString());
      }else{
            System.out.println("No se encuentra la clave buscada");
      }

      }
```

La relación entre Alumno y Uea no se consultará en esta parte de la práctica aunque el procedimiento sería el mismo, enviando como PK el número (recordando que es un auto incrementable) de la relación a consultar.

Selección de Múltiples Tuplas

Es posible seleccionar más de un elemento, primero se seleccionarán todos los elementos de una tabla y posteriormente se utilizará un criterio para su selección. Crear la clase **SeleccionesMultiples** en el paquete ***hibernate.operaciones***. Para seleccionar varios elementos, se tienen las siguientes instrucciones:

```
String sentencia = "FROM ruta.ruta.Clase";
//String sentencia = "FROM Clase";
objetoTipoList =  sesion.createQuery(sentencia).list();
```

En este caso se seleccionarán las licenciaturas registradas, no importa si se obtiene solo un elemento, de cualquier manera se puede manejar en una lista.

SeleccionesMultiples.java

```
package hibernate.operaciones;

import hibernate.beans.Licenciatura;
import hibernate.conexion.CrearConexion;
import java.util.List;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.hibernate.Session;

public class SeleccionesMultiples {

	private final static Logger log =
Logger.getLogger(SeleccionesMultiples.class);

	public List<Licenciatura>consultarLicenciaturas(){

		PropertyConfigurator.configure("logger.properties");
		List<Licenciatura>listaLicenciaturas;
		Session sesion = null;
		try{
			sesion = CrearConexion.getSessionFactory().openSession();
		}catch(ExceptionInInitializerError ex){
			System.err.println("No se pudo crear la sesion ");
		    throw new ExceptionInInitializerError(ex);
		}
		log.info("SELECCIONANDO LAS LICENCIATURAS");
		/*El nombre de la clase de la cuál se llenará la lista List*/
		String sentencia = "FROM hibernate.beans.Licenciatura";
		listaLicenciaturas =  sesion.createQuery(sentencia).list();
		return listaLicenciaturas;
	}
}
```

Principal.java

```
public static void main(String[] args) {

            //consultarLicenciatura("101101");
            //consultarUea("115102");
            //consultarAlumno("23456");
            consultarListaLicenciaturas();
            CrearConexion.closeSessionFactory();
      }

      private static void consultarListaLicenciaturas(){
            SeleccionesMultiples selecciones = new SeleccionesMultiples();

            List<Licenciatura>lista = selecciones.consultarLicenciaturas();
            if(lista!=null){
                  for(int i=0;i<lista.size();i++)
                        System.out.println(lista.get(i).toString());
            }
            else{
                  System.out.println("No hay elementos");
            }
      }
```

Es posible también seleccionar los elementos que cumplan con un determinado criterio, en este caso se consultarán los alumnos que corresponden a una determinada licenciatura.

El formato de instrucción correspondiente es:

```
String sentencia = "FROM CLASE ALIAS WHERE ALIAS.atributoClase = valor";
```

ALIAS es un alias que se utiliza para no escribir el nombre completo de la clase, notar que se escribe el nombre del *atributo* **NO** el nombre de la columna. Este atributo debe estar mapeado en el archivo ***hbm.xml***.

De esta forma, se seleccionan los alumnos de una licenciatura.

SeleccionesMultiples.java

```
public List<Alumno>consultarAlumnos(String licenciatura){
            PropertyConfigurator.configure("logger.properties");
            List<Alumno>listaAlumnos;
            Session sesion = null;
            try{
                  sesion = CrearConexion.getSessionFactory().openSession();
            }catch(ExceptionInInitializerError ex){
                  System.err.println("No se pudo crear la sesion ");
                throw new ExceptionInInitializerError(ex);
            }
```

```
            log.info("SELECCIONANDO LAOS ALUMNOS DE LA LICENCIATURA " +
licenciatura);
            String sentencia = "FROM hibernate.beans.Alumno ALU WHERE
ALU.licenciatura=101102";
            listaAlumnos =sesion.createQuery(sentencia).list();
            return listaAlumnos;
      }
```

Principal.java

```
      public static void main(String[] args) {

            //consultarLicenciatura("101101");
            //consultarUea("115102");
            //consultarAlumno("23456");
            //consultarListaLicenciaturas();
            consultarListaAlumnos("101102");
            CrearConexion.closeSessionFactory();
      }

private static void consultarListaAlumnos(String licenciatura){
            SeleccionesMultiples selecciones = new SeleccionesMultiples();
            List<Alumno>lista = selecciones.consultarAlumnos(licenciatura);
            if(lista!=null){
                  for(int i=0;i<lista.size();i++)
                        System.out.println(lista.get(i).toString());
            }else{
                  System.out.println("No hay elementos");
            }
      }
```

Aunque se especifica que se envía una clave en particular, el valor se escribió en el código, esto normalmente no es deseable, a continuación se presentan las sentencias para recibir y utilizar una clave "*dinámica*".

La sentencia se construye:

```
String sentencia = "FROM Clase ALIAS WHERE ALIAS.atributoClase =
:nombreParametro";
```

Para esto es necesario crear un objeto ***Query*** (***org.hibernate.query.Query***) y configurar los parámetros necesarios con el método ***setParameter("nombreParametro",valor/variable)***.

SeleccionesMultiples.java

```
	public List<Alumno>consultarAlumnos(String licenciatura){

		PropertyConfigurator.configure("logger.properties");
		List<Alumno>listaAlumnos;
		Session sesion = null;
		try{
			sesion = CrearConexion.getSessionFactory().openSession();
		}catch(ExceptionInInitializerError ex){
			System.err.println("No se pudo crear la sesion ");
		    throw new ExceptionInInitializerError(ex);
		}
		log.info("SELECCIONANDO LAOS ALUMNOS DE LA LICENCIATURA " +
licenciatura);
		String sentencia = "FROM hibernate.beans.Alumno ALU WHERE
ALU.licenciatura = :clave";
		Query query = sesion.createQuery(sentencia);
		query.setParameter("clave", licenciatura);
		listaAlumnos = query.list();
		return listaAlumnos;
	}
```

Principal.java

```
public static void main(String[] args) {

		//consultarLicenciatura("101101");
		//consultarUea("115102");
		//consultarAlumno("23456");
		//consultarListaLicenciaturas();
		consultarListaAlumnos("101102");
		CrearConexion.closeSessionFactory();
	}
	private static void consultarListaAlumnos(String licenciatura){
		SeleccionesMultiples selecciones = new SeleccionesMultiples();

		List<Alumno>lista = selecciones.consultarAlumnos(licenciatura);
		if(lista!=null){
			if(lista.size()!=0){
				for(int i=0;i<lista.size();i++)
					System.out.println(lista.get(i).toString());
			}else{
				System.out.println("No hay elementos");
			}

		}
	}
```

Otros Modificadores

De la misma forma, es posible utilizar otros modificadores, incluyendo conectores lógicos y criterios de ordenamiento.

SeleccionesMultiples.java

```
      public List<Alumno>consultarAlumnos(String licenciatura){

            PropertyConfigurator.configure("logger.properties");
            List<Alumno>listaAlumnos;

            Session sesion = null;
            try{
                  sesion = CrearConexion.getSessionFactory().openSession();
            }catch(ExceptionInInitializerError ex){
                  System.err.println("No se pudo crear la sesion ");
                throw new ExceptionInInitializerError(ex);
            }

            log.info("SELECCIONANDO LAOS ALUMNOS DE LA LICENCIATURA " +
licenciatura);
            String sentencia = "FROM hibernate.beans.Alumno ALU ORDER BY
ALU.licenciatura";
            Query query = sesion.createQuery(sentencia);
            //query.setParameter("clave", licenciatura);

            listaAlumnos = query.list();

            return listaAlumnos;
      }
```

En este caso, como no se utiliza ningún parámetro, no se les debe dar valor alguno.

Selección entre Múltiples Tablas

Es común que se tengan que presentar datos que combinen información de varias tablas. En esta parte de la práctica se revisará como obtener esa información.

Se crearán los siguientes elementos:

- Un paquete llamado **hibernate.compuestas**
- Una clase en el paquete **hibernate.compuestas** llamada **AlumnoLicenciatura**
- Una clase en el paquete **hibernate.operaciones** llamada **SeleccionesEntreTablas**

AlumnoLicenciatura.java

```
package hibernate.compuestas;

public class AlumnoLicenciatura {

	private String matricula;
	private String nombre;
	private String primerApellido;
	private String segundoApellido;
	private String licenciatura;

	public String getMatricula() {
		return matricula;
	}
	public void setMatricula(String matricula) {
		this.matricula = matricula;
	}
	public String getNombre() {
		return nombre;
	}
	public void setNombre(String nombre) {
		this.nombre = nombre;
	}
	public String getPrimerApellido() {
		return primerApellido;
	}
	public void setPrimerApellido(String primerApellido) {
		this.primerApellido = primerApellido;
	}
	public String getSegundoApellido() {
		return segundoApellido;
	}
	public void setSegundoApellido(String segundoApellido) {
		this.segundoApellido = segundoApellido;
	}


	public String getLicenciatura() {
```

```
            return licenciatura;
      }
      public void setLicenciatura(String licenciatura) {
            this.licenciatura = licenciatura;
      }

      public String toString(){
            String mensaje ="";

            mensaje = matricula + ", " + nombre
+","+primerApellido+","+segundoApellido+","+licenciatura;

            return mensaje;
      }

}
```

La clase **AlumnoLicenciatura** contiene los elementos a obtener de la consulta de dos tablas. El principal problema es que no se tiene una tabla que contenga todos los elementos por lo que no se cuenta con un mapeo. En este caso se debe llenar la clase a utiliza a partir de los resultados regresados por la sentencia.

Para construir la sentencia, se tiene:

```
String sentencia = "SELECT ALIAS1.atributo1, ALIAS1.atributo2,
ALIASN.atributoN FROM Clase1 as ALIAS1, ClaseN as ALIASN";
```

En donde:

- **Clase1** y **ClaseN** son clases mapeadas a su tabla correspondiente
- **ALIAS1** y **ALIASN** son nombres que facilitan el manejo de las clases
- **atributo1**, **atributo2** y **atributoN** son atributos de las clases (**NO** usar los nombres de columnas) que están mapeadas a una tabla

Se construyen los objetos necesarios

```
Query query = sesion.createQuery(sentencia);
lista = query.list();
```

Aquí se obtiene una lista de arreglos de objetos (**Object []**), cada elemento de la lista corresponde a la información que se tendría en la tupla obtenida por la base de datos y cada elemento del objeto corresponde a una columna (en el orden especificado por los nombres en la sentencia).

SeleccionesEntreTablas.java

```
package hibernate.operaciones;

import java.util.LinkedList;
import java.util.List;

import org.apache.log4j.Logger;
import org.hibernate.Session;
import org.hibernate.query.Query;

import hibernate.compuestas.AlumnoLicenciatura;
import hibernate.conexion.CrearConexion;

public class SeleccionesEntreTablas {
     private final static Logger log =
Logger.getLogger(SeleccionesEntreTablas.class);

     public List<AlumnoLicenciatura>alumnoLicenciatura(){

          List<Object[]>lista;
          List<AlumnoLicenciatura>listaAl=new LinkedList();
          Session sesion = null;
          try{
               sesion = CrearConexion.getSessionFactory().openSession();
          }catch(ExceptionInInitializerError ex){
               System.err.println("No se pudo crear la sesion ");
              throw new ExceptionInInitializerError(ex);
          }
          String sentencia = "SELECT ALU.matricula, ALU.nombre,
ALU.primerApellido, "+ "ALU.segundoApellido, LIC.nombre FROM Alumno as
ALU, Licenciatura"+ " as LIC WHERE LIC.clave = ALU.licenciatura";

          Query query = sesion.createQuery(sentencia);
          lista = query.list();

          for(int i=0;i<lista.size();i++){

               Object[] datosRecuperados = lista.get(i);

               AlumnoLicenciatura alumnoLic = new AlumnoLicenciatura();
               alumnoLic.setMatricula((String)datosRecuperados[0]);
               alumnoLic.setNombre((String)datosRecuperados[1]);
               alumnoLic.setPrimerApellido((String)datosRecuperados[2]);
               alumnoLic.setSegundoApellido((String)datosRecuperados[3]);
               alumnoLic.setLicenciatura((String)datosRecuperados[4]);
               listaAl.add(alumnoLic);
          }
          return listaAl;
     }
}
```

También es posible agregar modificadores para cambiar el comportamiento de la sentencias

SeleccionesEntreTablas.java

```
String sentencia = "SELECT ALU.matricula, ALU.nombre, ALU.primerApellido,
"+ "ALU.segundoApellido, LIC.nombre FROM Alumno as ALU, Licenciatura"+ "
as LIC WHERE LIC.clave = ALU.licenciatura AND ALU.licenciatura = :clave
ORDER BY ALU.matricula";
          Query query = sesion.createQuery(sentencia);
          query.setParameter("clave", "A101101");
```

Principal.java

```
public static void main(String[] args) {

          //consultarLicenciatura("101101");
          //consultarUea("115102");
          //consultarAlumno("23456");

          //consultarListaLicenciaturas();
          //consultarListaAlumnos("101102");
          consultarVarias();

          CrearConexion.closeSessionFactory();

     }

     private static void consultarVarias(){
          SeleccionesEntreTablas selecciones = new
SeleccionesEntreTablas();
          List<AlumnoLicenciatura> lista =
selecciones.alumnoLicenciatura();


          if(lista.size()!=0){
               for(int i=0;i<lista.size();i++){
                    System.out.println(lista.get(i).toString());
               }
          }else{
               System.out.println("NO HAY ELEMENTOS");
          }


     }
```