

Práctica No. 5. Operaciones Básicas con Hibernate – Actualización y Eliminación con Criterios

En esta práctica se trabajarán las actividades para realizar operaciones de actualización y eliminación de una o más filas utilizando criterios.

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 *Practica05Base.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 5** 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 5** el archivo *script_llenado_practica_05.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.

Se creará la clase **OperacionesMultiples** en el paquete **hibernate.operaciones**.

Actualización Múltiple con o sin Criterio

En este caso se le sumarán 10 créditos a los créditos de todas las uea registradas en la base de datos, una opción a utilizar puede ser obtener una lista con las llaves primarias de las uea e ir iterando sobre la lista para actualizar una a una.

Otra opción implica construir una sentencia que permita realizar la actualización.

OperacionesMultiples.java

```
package hibernate.operaciones;

import hibernate.conexion.CrearConexion;

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

```

public class OperacionesMultiples {

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

public void actualizarCreditos(){

    PropertyConfigurator.configure("logger.properties");

    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("ACTUALIZANDO LOS CREDITOS LAS LICENCIATURAS");

    String sentencia = "UPDATE Uea SET creditos = creditos +
:creditosTotales";

    sesion.getSession().beginTransaction();
    Query query = sesion.createQuery(sentencia);
    query.setParameter("creditosTotales", 10);
    int a = query.executeUpdate();
    System.out.println("Se afectaron " + a + " filas");
    sesion.getTransaction().commit();
    sesion.close();

    }

}

```

Eliminación Múltiple con Criterio

Se eliminarán los elementos de una tabla que correspondan a un determinado criterio, en este caso se eliminarán los registros de calificación que tengan “S”.

OperacionesMultiples.java

```

public void borrarSuficientes(){

    PropertyConfigurator.configure("logger.properties");

    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("BORRANDO LAS CALIFICACIONES CON S");

String sentencia = "DELETE AlumnoUea WHERE calificacion =
:criterioBorrar";

sesion.getSession().beginTransaction();
Query query = sesion.createQuery(sentencia);
query.setParameter("criterioBorrar", "S");
int a = query.executeUpdate();
System.out.println("Se afectaron " + a + " filas");
sesion.getTransaction().commit();
sesion.close();
}

```

Principal.java

```

package hibernate.principal;

import hibernate.conexion.CrearConexion;
import hibernate.operaciones.OperacionesMultiples;

public class Principal {

    public static void main(String[] args) {

        //actualizarCreditos();
        borrarSuficientes();
        CrearConexion.closeSessionFactory();

    }
    private static void actualizarCreditos(){

        OperacionesMultiples operaciones = new OperacionesMultiples();
        operaciones.actualizarCreditos();
    }
    private static void borrarSuficientes(){
        OperacionesMultiples operaciones = new OperacionesMultiples();
        operaciones.borrarSuficientes();
    }
}

```