Resumen de Mapeo

(POR EJEMPLO EN NAME="CLASE" SE ESCRIBE JUNTO SIN ESPACIOS!)

ejemplo Docente

/\*table="docente": se puede escribir en minuscula por mas que en la tabla en DB este con mayuscula. No hace diferencia entre mayus y minus

\*/

<class name="org.integrados.data.usuarios.Docente" table="docente" catalog="integrados\_db" optimistic-lock="version">

<id name="id" tipe="Integer" column="id">

<generator class="identity" />

/\*Relaciones con otras tablas:

si la relación es one-to-one= es lo mismo poner many-to-one que one-to-one. En la BD sabe que relación corresponde

\*/

//Ejemplo relación docente con domicilio.Hecho desde docente.hbm.xml (un docente tiene un domicilio)

<many-to-one name="domicilio" class="org.integrados.data.usuarios.Domicilio" fetch="join"<

<column name="id\_Domicilio" />

/\*fetch="join" : reduce el número de consultas que se generan en la base de datos.

\*/

//Atributos de la clase:

<property name="usuario" type="string" column="usuario" length="20" />

<property name="nombre" type="string" column="nombre" length="45" not-null="true" /> //si se NO se agrega not-null por defecto toma "false"

//MAPEAR LISTAS

Hay 3 tipos de mapeo con listas:

\*<bag>: es una colección desordenada, que puede contener elementos duplicados. Eso significa que si persistes una bolsa con algún orden de elementos, no puedes esperar que se retenga el mismo orden cuando se recupera la colección.

EJEMPLO: <bag name="employees" table="employee" inverse="true" cascade="all-delete-orphan lazy="false" fetch="join">

<key> <column name="employee\_id" not-null="true" /> </key>

<one-to-many class="net.viralpatel.hibernate.Employee" /> </bag>

\*<set>: es similar a <bag> excepto que solo puede almacenar objetos únicos. Eso significa que no se pueden contener elementos duplicados en un conjunto. Cuando agrega el mismo elemento a un conjunto por segunda vez, reemplazará al anterior. Un conjunto no está ordenado por defecto, pero podemos pedir que se ordene.

EJEMPLO(docente tiene una lista de actividades):

<set name="actividades" table="actividad" inverse="true" cascade="all-delete-orphan" lazy="false" fetch="join">

<key>

<column name="id\_Docente" not-null="true" />

</key>

<one-to-many class="org.integrados.data.usuarios.Actividad" />

</set>

\*<list>: puede contener duplicados pero con un orden conservado.

EJEMPLO(en hbm alumno, el alumno tiene una lista de registro actividades) :

<list name="registroActividades" inverse="true" cascade="all-delete-orphan" fetch="join" lazy="false">

<key column="id\_alumno" not-null="true"/>

<index column="id" />

<one-to-many class="org.integrados.data.RegistroActividad"/>

</list>

En registroActividad.hbm.xml:

<many-to-one name="alumno" column="id\_alumno" class="org.integrados.data.Alumno"/>

/\* inverse=”true” : se crea para definir qué lado es el propietario para mantener la relación (Propietario de la lista). Solo se usa en relaciones muchos a uno y uno a muchos.

fetch="join" : reduce el número de consultas que se generan en la base de datos.

cascade="all-delete-orphan" : si se borra un alumno borra en cascada las filas de registroActividad asociadas a ese alumno.

Lazy=”false” : al momento de recuperar el alumno, recupera toda la lista de registroActividad. Si está en “true” al obtener el alumno de la base no carga la lista de registroActividad. SIEMPRE DEBE ESTAR EN FALSE.

\*/