HIBERNATE - O/R MAPPINGS

So far we have seen very basic O/R mapping using hibernate but there are three most important mapping topics which we have to learn in detail. These are the mapping of collections, the mapping of associations between entity classes and Component Mappings.

Collections Mappings:

If an entity or class has collection of values for a particular variable, then we can map those values using any one of the collection interfaces available in java. Hibernate can persist instances of **java.util.Map, java.util.Set, java.util.SortedMap, java.util.SortedSet, java.util.List**, and any **array** of persistent entities or values.

Collection type	Mapping and Description
java.util.Set	This is mapped with a <set> element and initialized with java.util.HashSet
java.util.SortedSet	This is mapped with a <set> element and initialized with java.util.TreeSet. The **sort** attribute can be set to either a comparator or natural ordering.
java.util.List	This is mapped with a <list> element and initialized with java.util.ArrayList
java.util.Collection	This is mapped with a <bag> or <ibag> element and initialized with java.util.ArrayList
java.util.Map	This is mapped with a <map> element and initialized with java.util.HashMap
java.util.SortedMap	This is mapped with a <map> element and initialized with java.util.TreeMap. The **sort** attribute can be set to either a comparator or natural ordering.

Arrays are supported by Hibernate with <primitive-array> for Java primitive value types and <array> for everything else. However, they are rarely used so I'm not going to discuss them in this tutorial.

If you want to map a user defined collection interfaces which is not directly supported by Hibernate, you need to tell Hibernate about the semantics of your custom collections which is not very easy and not recommend to be used.

Association Mappings:

The mapping of associations between entity classes and the relationships between tables is the soul of ORM. Following are the four ways in which the cardinality of the relationship between the objects can be expressed. An association mapping can be unidirectional as well as bidirectional.

Mapping type	Description
Many-to-One	Mapping many-to-one relationship using Hibernate
One-to-One	Mapping one-to-one relationship using Hibernate
One-to-Many	Mapping one-to-many relationship using Hibernate
Many-to-Many	Mapping many-to-many relationship using Hibernate

Component Mappings:

It is very much possible that an Entity class can have a reference to another class as a member variable. If the referred class does not have it's own life cycle and completely depends on the life cycle of the owning entity class, then the referred class hence therefore is called as the Component class.

The mapping of Collection of Components is also possible in a similar way just as the mapping of regular Collections with minor configuration differences. We will see these two mappings in detail with examples.

Mapping type	Description
Component Mappings	Mapping for a class having a reference to another class as a member variable.