Smita – 7738206222

brijeshsmita@gmail.com

JPA – Java Performance API to perform CRUD operation. It is a database framework.

It provides set of Interfaces, classes, method to perform various DB operations

Entity Manager Factory – It is an Interface

In order to manage an entity we require Entity Manager, for entity manager we need entity manager factory. Both of these are provided by Persistence class.

Framework –

* Mapping the object with the table and properties with the column
* JPA will implement ORM – Object Relation Mapping, which will eventually do the line above.
* We will be using XML for connectivity.
* Annotation for mapping.
* @Entity annotation will tell JPA that the particular class should be mapped to a particular table (it will not give the reference to which particular table)
* @Table (name=<table name>) annotation will give the exact table to which the class is mapped to.
* JPA is implemented by the following tools
  + Hibernate
  + Toplink
  + iBatis
  + Eclipse Link
* Persistence will take reference of connection to database from persistence unit name which is defined in persistence.xml
* Hibernate is a ORM framework which will implement JPA
* ORM will perform JDBC
* To create a JPA Project follow the below steps
  + Create core java project (entity)
  + DB
  + Class
  + Project/lib – add all jars to build path
  + Src/META-INF/persistence.xml
* When we have em.persist(object) it will
  + It will go to the object’s class.
  + It will then call the getter to get the data from the main class
  + JPA will then internally call the setters to set the data.

Steps to create a join table program

3 types of collections (interface):

* List (Ordered Collections)
* Set (Unique Collections)
  + The above interface in implemented by the below 2 classes
  + Assert(class). This has sort method
  + Collections(class) This has sort () method
* Java.util has one more Interface which is Map
* Vector Class
  + It is synchronized-thread safe, legacy
* ArrayList
  + We need frequent get and set

Iterator are of 2 types:

* Iterator – one way
* ListIterator – two way

Important points to design JPA

* First decide the parent and child entities
* Unidirectional -> when one entity is persists then only that entity is persist not the child one
* Bi directional – when parent is persisted then child will also persist
* Mapped by - we mention the extra column created in the other table with@joincolumn. It is always given in the parent class
* @PrimaryKeyJoinColumn – When we don’t need an extra column.
* @JoinColumn – it will create a new column in child table
* Mapped by and cascade type is needed for bi directional
* Mapped by rules
* Inheritance Rules
  + @Inheritance annotation will put on parent table only
* JPQL – Java Persistence Query Language
  + It would deal with your classes or Object/ Object based
  + NamedQuery – the way to define a query by giving it a name
  + We can do the following queries through JPQL
* Maven is a build tool has pom.xml
  + Convention over configuration
  + It gives a directory structure
  + Artifact – project name
  + Group id – package name
* Join column will always be there in the owner table which has the foreign key(primary key of the other table)
* Mapped by is always there in the secondary table(where the foreign key is the primary key of owner table) we include the property name of the field where @JoinColumn is given in the primary table