(No 1 in Training & Placement)



Java Training Center

(No.1 in Training & Placement)

Hibernate-2 Doc.

Master the Content...

Author

Som Prakash Rai

(No 1 in Training & Placement)

Association Mapping

- If you want to establish the relationship among different persistence classes, then you have to use association mapping.
- Depending on the cardinality, there are 3 types of association mapping:

One-to-one mapping

One-to-Many mapping

Many-to-many mapping

• Depending on the directionality, you can establish two type of relationship;

Uni – Directional Relationship

Bi – Directional Relationship

- Cardinality represents number of objects participating in relationship on both the sides,
- Directionality represents whether we can access the data in one direction only or both the directions.

EX:

• One-one Bi – directional

One customer can have one address and one address belongs to one customer.

• One – many Bi – Directional

One customer can place many orders and one order placed by one customer.

• Many – many bi – directional

One customer can have many accounts and one account belongs to many customers.

One student can join for many courses and many students can join for one course.

One to one Uni-Directional Mapping

• Consider the relationship between customer and address.

One customer contains one address.

One address belongs to one customer.

A) Tables Required

My customers

cid	First name	Last name	Email	Phone	Aid
	Som	Prakash	Som@jtcindia.org	3333	1
1					

Addresses

Aid	Street	City	State
	SEC-2	Noida	UP
1			

(No 1 in Training & Placement)

Example with hibernate core

B) Hibernate persistence classes

```
    Customer.java
        Public class customer {
        ......
        Address;
        }
        Address. java
        Classaddress}
        .....
    }
```

C) Hibernate mapping document

```
1) Customer.hbm.xml
<a href="chibernate-mapping-package="com.jtcindia.hibernate">
<class name="Student"table="mystudents">
<id name="cid"column="cid"type="int">
<generator coass="increment"/>
</id>
cproperty name="firstName"column="firstName"/>
column=''firstName''/>
cproperty name="email"column="firstName"/>
cproperty name="phone"column="ifirstName"/>
</class>
<a href="chibernate-mapping-package="com.jtcindia.hibernate">
<class name="address"table="addresses">
<id name="aid"column=" aid"type="int">
<generator coass="increment"/>
</id>
cproperty name="street"/>
cproperty name="city"/>
cproperty name="status"/>
</class>
</hibernate-mapping>
```

Jtc11: Files required

Jtc11A. java	Jtc11. java
	Address. java
Customer. Java	

Java Training Center (No 1 in Training & Placement)

	Address, hbm. com
Customer. Hbm. Xml	
	Hibernate. Cfg. xml
CHibernate Util. java	

Java Training Center (No 1 in Training & Placement)

```
Customer.hbm.xml No 1 in Training & Placement <a href="https://www.nbm.nibernate">hibernate</a> Placement <a href="https://www.nbm.nibernate">hibernate</a> Placement
Customer.java
Package com.jtcindia.hibernate;
Import java.util.*;
                                           <class name="customer"table="mycustomers">
                                           <id name="cid"column="cid"type="int">
*@Author:Som PrakashRai
                                           <generator class="increment"/>
*@company:java Training Center
                                           </id>
*@Visit
              : www.jtcindia.org
                                           cproperty name="firstName"/>
**/
                                           column="lestname"/>
Public class customer {
                                           cproperty name="email"column="email"/>
Int cid:
                                           cproperty name="phone"column="phone"/>
String firstName;
                                           <many-to-one name="address"class="Address" column</pre>
String last name;
                                           ="aid/>
String email:
                                           </class>
String phone:
                                           </hibernate-mapping>
Address address;
Public customer() {}
Public customer (string firstName, string
lastName,string email,string phone){
This.firstName=firstName;
This.lostName=lastName;
This.email=email;
This.phone=phone;
//setters and getters
Address.java
                                           Adddress.hbm.xml
Package com.jtcindia.hibernate;
                                           <hibernate-mapping package="com.jtcindia.hibernate">
Import java.util.*;
                                           <class name="customer"table="addresses">
                                           <id name="cid"column="aid"type="int">
*@Author:Som PrakashRai
                                           <generator class="increment"/>
                                           </id>
*@company:java Training Center
*@Visit
              : www.jtcindia.org
                                           property name="street"/>
**/
                                           cproperty name="city"/>
Public class customer {
                                           cproperty name="state"/>
                                           </class>
Int aid;
String street;
                                           </hibernate-mapping>
String city;
String state;
Address address:
Public customer() {}
Public address (string steet, string city,
string state){
This.street=street;
This.city=sity
This.state=state;
 //setters and getters
```

Example with hibernate annotation

B) Hibernate persistence classes

(No 1 in Training & Placement)

```
1. Customer.java

@Entity
@ Table (name="mystudents")
Public class student {
.....
@One to one
@Join Column (name="my customers")
Address;
}
2. Address.java
@Entity
@ Table (name=" Address")
Class address
......
}
```

Jtc12: Files requiredJtc12A. javaJtc12B. javaCustomer. JavaAddress. JavaAHibernate Util. javaHibernate. Cfg. xml

```
JTC12A.java
                                                        JTC12B.java
package com.jtcicindia.hibernate;
                                                        package com.jtcicindia.hibernate;
import org.hibernate.*;
                                                        import org.hibernate.*;
*@Author:Som PrakashRai
                                                        *@Author:Som PrakashRai
*@company:java Training center
                                                        *@company:java Training center
*@Visit :WWW.Jtcindia.com
                                                        *@Visit :WWW.Jtcindia.com
public class Jtc12A{
                                                        public class Jtc12B{
public static void main(string[]args){
                                                        public static void main(string[]args){
Transaction tx=null;
                                                        Transaction tx=null;
                                                        try{
SessionFactory sf=CHibernateUtil.getSessionFactory();
                                                        SessionFactory
Session session=sf.openSession();
                                                        sf=CHibernateUtil.getSessionFactory();
tx=session.beginTransaction();
                                                        Session session=sf.openSession();
//same as Jtc11A
                                                        tx=session.beginTransaction();
tx.commit();
                                                         //same as Jtc11B
session.close();
                                                        tx.commit();
System.out.println("record inserted");
                                                        session.close();
)catch(Exception e){
                                                        System.out.println("record inserted");
e.prinStackTrace();
                                                        )catch(Exception e){
if(txl=null)tx.rollback();
                                                        e.prinStackTrace();
}}}
                                                        if(txl=null)tx.rollback();
Customer.java
                                                        Address.java
```

(No 1 in Training & Placement)

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
*@Author:Som PrakashRai
*@company:java Training center
*@Visit
            :www.jtcindia.org
**/
@Entity
@table(name="mycustomers")
Public class customer{
@id
@Column(name="cid")
@GeneratedValue(strategy=GenerationType.AUTO)
Private int cid;
@Column(name="firstName")
Private string cname;
@Column(name="lastName")
Private string cname;
@Column(name="email")
Private string cname;
@Column(name="phone")
Private string cname;
@Column(name="city")
Private string cname;
@Column(name="aid")
Private string cname;
Public customer() {}
Public Customer(String firstName,String email,long
phone, String lastName) {
this.cname=cname;
this.email=email;
this.farstName=farstName;
this lastName=lastName;
     //Setters and Getters
```

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
*@Author:Som PrakashRai
*@company:java Training center
*@Visit
            :www.jtcindia.org
**/
@Entity
@table(name="addresses")
Public class customer{
@id
@Column(name="aid")
@GeneratedValue(strategy=GenerationType.AUTO)
Private int aid;
@Column(name="street")
Private string cname;
@Column(name="city")
Private string cname;
@Column(name="state")
Private string cname;
Public customer() {}
Public Customer(String street, String state, String
this.street=street;
this.city=city;
this.state=state;
     //Setters and Getters
}
```

(No 1 in Training & Placement)

One to one Uni-Directional Mapping

• Consider the relationship between customer and address.

One customer contains one address.

One address belongs to one customer.

D) Tables Required

My customers

cid	First name	Last name	Email	Phone	Aid
1	Som	Novas	Som@jtcindia.org	3333	1

Addresses

Aid	Street	City	State
1	SEC-2	Noida	UP

Example with hibernate core

B) Hibernate persistence classes

3) Customer.java

Public class customer {

.

Address address;

}

4) Address. java

Classaddress }

Customer customer;

}

C) Hibernate mapping document

```
<hibernate-mapping package="com.jtcindia.hibernate">
<class name="Student"table="mystudents">
<id name="sid"column="cid"type="int">
</derival column="int" column="int" column="firstName"/>
</id>
</discriminator column="firstName"/>
<discriminator column="lastName"/>
<discriminator column="mail"/>
<discriminator column="phone"/>
<discriminator column="mail"/>
</discriminator column="mail"/>
</discriminator
```

(No 1 in Training & Placement)

```
</hibernate-mapping>
<hibernate-mapping package="com.jtcindia.hibernate">
<class name="Address"table="Addresses">
<id name="air"column="air"type="int">
<generator class="increment"/>
</id>
<discriminator column="stutype"/>
<property name="city"/>
<property name="status"/>
<property name="customer"class="customer"/>
</class>
</hibernate-mapping>
```

Jtc13: .Files required

Jtc13A. java	Jtc13B. java
Jtc13c. java	Customer.Java
Address. java	Customer.Hbm.Xml
Address. Hbm. xml	CHibernateUtil.java
Hibernate. Cfg. xml	

(No 1 in Training & Placement)

JTC13A.java JTC13B.java package com.jtcicindia.hibernate; package com.jtcicindia.hibernate; import org.hibernate.*; import org.hibernate.*; public class Jtc13A{ public class Jtc13B{ public static void main(string[]args){ public static void main(string[]args){ Transaction tx=null; Transaction tx=null; try{ try{ SessionFactory SessionFactory sf=CHibernateUtil.getSessionFactory(); sf=CHibernateUtil.getSessionFactory(); Session session=sf.openSession(); Session session=sf.openSession(); tx=session.beginTransaction(); tx=session.beginTransaction(); customer cust=new Customer cust=(customer) customer("Som","Som","Som@jtc","9036"); Session.load(customer.class,2); System.out.println(cust.getCid()+"\t"+cust.getFirst session.save(cust); address add=new address("sec-2","Noida", Name() +"\"+cust.getLastName()+"\t"+add.getCity()+"\t"+ "Uttarpradesh"); session.save(add); cust.getPhone()); cust.setAddress(add); Address add=cust.getAddress(); System.out.println(add.getAid()+"\t"+add.getCity()+ tx.commit(); system.out.println("record inserted"); "\t"+add.getstreet()+"\t"+add.getState(); }catch(Exception e) { Tx.commit(); e.prinStackTrace(); Session.close(); }catch(Exception e){e.printStackTrace(); if(tx!=null) tx.rollback(); It(tx!=null)tx.rollback();

```
Customeriava in Training & Placement)
JTC13C.java
package com.jtcicindia.hibernate;
                                                     package com.jtcicindia.hibernate;
import org.hibernate.*;
                                                     *@Author:Som PrakashRai
*@Author:Som PrakashRai
                                                     *@Company:Java Training Center
                                                     *@Visit
*@Company:Java Training Center
                                                                   :www.jtcindia.org
*@Visit
              :www.jtcindia.org
**/
                                                     public class customer{
public class Jtc13C{
                                                     private integer cid:
public static void main(string[]args){
                                                     private string first name;
Transaction tx=null;
                                                     private string lastName;
try{
                                                     private string email;
SessionFactory
                                                     private string phone;
sf=CHibernateUtil.getSessionFactory();
                                                     private address address;
Session session=sf.openSession();
tx=session.beginTransaction();
                                                     public customer() {}
Customer cust=(customer)
Session.load(customer.class,2);
                                                     public customer(string firstName, string lastName, string
System.out.println(cust.getCid()+"\t"+cust.getFirst
                                                     email, string phone){
Name()
                                                     this.lfirstNme=firstName;
+"\"+cust.getLastName()+"\t"+add.getCity()+"\t"+
                                                     this.lastName=lastName;
cust.getPhone());
                                                     this.email=email;
Address add=cust.getAddress();
                                                     this.phone=phone;
System.out.println(add.getAid()+"\t"+add.getCity()+
"\t"+add.getstreet()+"\t"+add.getState();
                                                     //seters and getters
Tx.commit();
Session.close();
}catch(Exception e){e.printStackTrace();
It(tx!=null)tx.rollback();
} } }
                                                     Customer.hbm.xml
Address.java
package com.jtcicindia.hibernate;
                                                     <a href="chibernate-mapping package="com.jtcindia.hibernate">
                                                     <class name="customer"table="table="mycustomers">
*@Author:Som PrakashRai
                                                     <id name="cid"column="cid"type="int">
*@Company:Java Training Center
                                                     <generator class="increment"/>
*@Visit
              :www.itcindia.org
                                                     </id>
                                                     cproperty name="firstName"column="firstName"/>
                                                     cproperty name="lastName" column="lastName"/>
public class customer{
private integer aid;
                                                     cproperty name="email"column="phone"/>
private string street;
                                                     <many-to-one
                                                     name="address"class="address"column="aid"/>
private string city;
private string state;
                                                     </class>
Customer customer;
                                                     </hibernate-mapping>
//Custructors
//setters and getters
```

(No 1 in Training & Placement)

Address.hbm.xml

<hibernate-mapping

package="com.jtcindia.hibernate">

<class name="Address"table="addresses">

<id name="aid"column="aid"type="int">

<generator class="increment"/>

</id>

cproperty name="street"/>

cproperty name="city"/>

property name="state"/>

<one-to-one name="customer"class="customer"/>

</class>

</hibernate-mapping>

Example with hibernate annotation

B) Hibernate persistence classes

1. Customer.java

@Entity

@ Table (name="mystudents")

Public class student {

13

Hibernat-2 Doc Author :SomPrakash Rai $www.jtcindia.org\\ Copyright @JTC^{TM}$

(No 1 in Training & Placement)

@One to one
@Join Column (name="aid")
Address;
}
2. Address.java
@Entity
@ Table (name="Address")
Class address
.....
@One to one
@Join Column (name="aid")
Customer;
Jtc14: .Files required

Jtc14A. java	Jtc14B. java
Jtc14c. java	Customer. java
Address. Java	CHibernateUtil. java
Hibernate, Cfg. xml	

Java Training Center (No 1 in Training & Placement)

JTC14A.java	JTC14B.java
package com.jtcicindia.hibernate; import org.hibernate.*; /* *@Author:Som PrakashRai *@Company:Java Training Center *@Visit :www.jtcindia.org **/ public class Jtc13C{ public static void main(string[]args){ Transaction tx=null; try{ SessionFactory sf=CHibernateUtil.getSessionFactory(); Session session=sf.openSession(); tx=session.beginTransaction(); Customer cust=new Customer("WEBEARN","WEBEARN","webearn@jtc", 9876"); session.save(cust); address add=new address("Sec-2","Noida", "UP"); session.save(add); cust.setAddress(add); tx.commit(); system.out.println("record inserted"); } catch(Exception e) { e.prinStackTrace(); if(tx!=null) tx.rollback(); }}}	package com.jtcicindia.hibernate; import org.hibernate.*; /* *@ Author:Som PrakashRai *@ Company:Java Training Center *@ Visit :www.jtcindia.org **/ public class Jtc13C{ public static void main(string[]args){ Transaction tx=null; try{ SessionFactory sf=CHibernateUtil.getSessionFactory(); Session session=sf.openSession(); tx=session.beginTransaction(); Customer cust=(customer) Session.load(customer.class,1); System.out.println(cust.getCid()+"\t"+" cust.getFirst Name() +"\"+cust.getLastName()+"\t"+add.getEmail() +"\t"+cust.getPhone()); Address add=cust.getAddress(); System.out.println(add.getAid()+"\t"+add.get City()+ "\t"+add.getstreet()+"\t"+add.getState(); Tx.commit(); Session.close(); }catch(Exception e){e.printStackTrace(); It(tx!=null)tx.rollback(); } }
JTC13C.java package com.jtcicindia.hibernate; import org.hibernate.*; /* *@Author:Som PrakashRai *@Company:Java Training Center *@Visit :www.jtcindia.org **/ public class Jtc13C{ public static void main(string[]args){ Transaction tx=null; try{ SessionFactory sf=CHibernateUtil.getSessionFactory();	tx=session.beginTransaction(); Customer cust=(customer) Session.load(customer.class,2); System.out.println(cust.getCid()+"\t"+cust.get First Name() +"\"+cust.getLastName()+"\t"+add.getCity()+ "\t"+cust.getPhone()); Address add=cust.getAddress(); System.out.println(add.getAid()+"\t"+add.get City()+ "\t"+add.getstreet()+"\t"+add.getState(); Tx.commit(); Session.close();

Java Training Center (No 1 in Training & Placement)

Sustomersiava=sf.openSession();	Addressiava(Exception e){e.printStackTrace();
Package com.jtcindia.hibernate;	Packagn (Amininia hihawate;
Import javax.persistence.*;	Import jayax.persistence.*;
r · Jr	/*
/*	*@Author:Som PrakashRai
*@Author:Som PrakashRai	*@company:java Training center
*@company:java Training center	*@Visit :www.jtcindia.org
*@Visit :www.jtcindia.org	**/
**/	@Entity
@Entity	@table(name="addresses")
@table(name="mycustomers")	Public class customer{
Public class customer{	
	@id
@id	@Column(name="aid")
@Column(name="cid")	@GeneratedValue(strategy=GenerationType.AUTO)
@GeneratedValue(strategy=GenerationType.AU	Private int aid;
TO)	
Private int cid;	@Column(name="street")
	Private string cname;
@Column(name="firstName")	
Private string cname;	@Column(name="city")
	Private string cname;
@Column(name="lastName")	
Private string cname;	@Column(name="state")
001 (" '1")	Private string cname;
@Column(name="email")	OC 1 ("' '1")
Private string cname;	@Column(name="aid")
(Column(nomo='nhono'')	Private string cname;
@Column(name="phone") Private string cname;	Public customer() {}
Frivate string chame,	rubile customer() {}
@Column(name="city")	Public Customer(String street, String state, String
Private string cname;	city){
Trivate string chame,	this.street=street;
@Column(name="aid")	this.city=city;
Private string cname;	this.state=state;
,	}
Public customer() { }	//Setters and Getters
Public Customer(String firstName, String email,	}
long phone, String lastName){	
this.cname=cname;	
this.email=email;	
this.farstName=farstName;	
this lastName=lastName;	
}	
//Setters and Getters	
}	

(No 1 in Training & Placement)

One to one Bi-Directional Mapping

Consider the relationship between customer and request.

One customer can place many requests.

One request belongs to one customer only.

A) Tables Required

1) jtccustomers

cid	First name	Last name	Email	Phone
1	Som	Prakash	Som@jtcindia.org	3333

2) requests

Req id	Req date	Description	Status	cid
1	23/7/2015	Hello 1	OK 1	1
2	23/7/2015	Hello 2	OK 2	1

Example with hibernate annotation

B) Hibernate persistence classes

1. Customer.java

Public class customer {

Private collection<request>requests:

...... }

2. Address.java

Public class customer {

Private collection

}

C) Hibernate mapping classes

```
<hibernate-mapping package="com.jtcindia.hibernate">
<class name="Customer"table="jtccustomers">
<id name="sid"column="cid"type="int">
<generator coass="increment"/>
</id>
<discriminator"firstName"column=" firstName"/>
<discriminator"lastName"column=" lastName"/>
<discriminator"email"column=" email"/>
<discriminator"phone"column=" phone"/>
<set name="requests"/>
<key column="cid"/>
<one-to-many class="Request"/>
</set>
</class>
</hibernate-mapping>
2. Request.hbm.xml
<hibernate-mapping package="com.jtcindia.hibernate">
```

(No 1 in Training & Placement)

Jtc15: .Files required

ste 15. If the required	
Jtc15A. java	Jtc15B. java
Jtc15c. java	Customer. java
Request. Java	Customer. Hbm. xml
Request. Hbm. Xml	CHibernate Util. java
Hibernate. Cfg. Xml	

(No 1 in Training & Placement)

```
1.Jtc15A.java
                                                          Jtc15B.iava
package com.jtcicindia.hibernate;
                                                          package com.jtcicindia.hibernate;
import org.hibernate.*;
                                                          import org.hibernate.*;
public class Jtc13A{
                                                          public class Jtc13A{
public static void main(string[]args){
                                                          public static void main(string[]args){
Transaction tx=null;
                                                          Transaction tx=null;
try{
                                                          try{
                                                          SessionFactory
SessionFactory sf=CHibernateUtil.getSessionFactory();
Session session=sf.openSession();
                                                          sf=CHibernateUtil.getSessionFactory();
tx=session.beginTransaction();
                                                          Session session=sf.openSession();
customer cust=new
                                                          tx=session.beginTransaction();
customer("Prakash","rai",prakash@jtc.com,9999);
                                                          Customer cust=(customer)
session.save(cust);
                                                          Session.load(Customer.class,1);
Request req1=new request("22/07/2015","hello1","
                                                          System.out.printIn("**Customer Info");
                                                          System.out.println(cust);
ok1");
                                                          System.out.printIn("**Requests info");
Request req2=new request("23/07/2015", "hello2","
"ok2");
                                                          Collection<Request>col=cust.getRequests();
Session.save(req1);
                                                          For(request reg:col)
                       session.save(req2);
Req1.setCustomer(ust);
                            req2.setCustomer(cust);
                                                          System.out.printIn(req);
Tx.commit();
                                                          Tx.commit();
Session.close();
                                                          Session.close();
System.out.printin("Record Inserted");
                                                          }catch(Exception e)(e.prinStackTrace();
}catch(Exception e){ e.printStackTrace();
                                                          If(tx!=null)tx.rolback();
If (tx !=null) tx.rollback();
}}}
                                                          }
```

(No 1 in Training & Placement)

```
Jtc15C.java
                                                       Customer.java
package com.jtcicindia.hibernate;
                                                       Package com.jtcindia.hibernate;
import org.hibernate.*;
                                                       Import java.util.set;
public class Jtc13A{
                                                       Public class customer{
public static void main(string[]args){
                                                       Private Integer cid;
                                                                             private string fistName
                                                       Private String lastName Private string email;
Transaction tx=null;
                                                       Private long phone;
try{
SessionFactory sf=CHibernateUtil.getSessionFactory();
                                                       Private set<request>requests;
Session session=sf.openSession();
tx=session.beginTransaction();
                                                       Public Customer() {}
Customer cust=(customer)
Session.load(Customer.class,1);
                                                       Public customer(string firstName, String
System.out.printIn("**Customer Info");
                                                       lastName,string email, long phone){
System.out.println(cust);
                                                       This.firstName=firstName;
System.out.printIn("**Requests info");
                                                       This.lastName=lastName;
Collection<Request>col=cust.getRequests();
                                                       This.email=email;
For(request reg:col)
                                                       This.phone=phone;
System.out.printIn(req);
Tx.commit();
                                                         //setters and getters
Session.close();
                                                       @override
}catch(Exception e)(e.prinStackTrace();
                                                       Public string to string() {
If(tx!=null)tx.rolback();
                                                       cid+","+firstName+","+LastName+","+email+","+ph
                                                       }}}
                                                       Customer.hbm.xml
Request.java
Package com.jtcindia.hibernate;
                                                       <hibernate-mapping
Public class request{
                                                       package="com.jtcindia.hibernate">
Private Integer reqld;
                       private string reqData
Private String description Private string status;
                                                       name="customer"table="itccustomers">
Private string description;
                                                       <id name="cid"column="cid"type="int">
Private Customer customer;
                                                       <generator class="increment"/>
                                                       </id>
Public request() {}
                                                       column="firstName"/>
                                                       Public customer(string reqData, String description,
                                                       cproperty name="email"column='phone'/>
string status){
                                                       cproperty name="phone" column="phone"/>
This. regData = regData;
                                                       <set name="requests">
This. description = description;
                                                       <key column="cid"/>
This. status = status;
                                                       <one-to-many class="request"/>
                                                       </set>
 //setters and getters
@override
                                                       </class>
Public string to string() {
                                                       </hibernate-mapping>
Returncid+","+ reqData +","+ description +","+ status;
}}}
```

(No 1 in Training & Placement)

Request.hbm.xml	<pre><many-to-one <="" class="Customer" name="customer" pre=""></many-to-one></pre>
<pre><hibernate-mapping package="com.jtcindia.hibernate"> <class name="Request" table="requests"> <id column="reqld" increment"="" name="reqld" type="int> <generator class="></id> <pre> <pre></pre></pre></class></hibernate-mapping></pre>	column="cid"/>

Example with hibernate annotation

- B) Hibernate persistence classes
- 1. Customer.java

```
@Entity
@ Table (name="jtccustomers")
Public class student {
.....
@One too Many (mapped by = "customer")
Private collection<Request>requests;
......
}
```

2. Address.java

```
@Entity
@ Table (name=" requests")
Public class requests {
.....
@Many to one
@Join column (name="cid",referenced column name = "cid")
Customer;
.....
}
```

Jtc14: .Files required

Jtc16A. java	Jtc16B. java
Jtc16c. java	Customer. java
Requests. Java	AHibernate Util. java

(No 1 in Training & Placement)

Hibernate. Cfg. Xml

```
Jtc16A.java
                                                     Jtc16B.java
Package com.jtcindia.hibernate;
                                                     Package com.jtcindia.hibernate;
Import org.hibernate.*;
                                                     Import java.util.collection;
                                                     Import org.hibernate.session;
*@Author:Som PrakashRai
                                                     Import org.hibernate.sessionfactory;
*@company:java Training center
                                                     Inport org.hibernate.transaction;
*@Visit
              :www.jtcindia.org
                                                     *@Author:Som PrakashRai
Public class Jtc16{
                                                     *@company:java Training center
Public static void main (string[]args){
                                                     *@Visit
                                                                    :www.jtcindia.org
                                                     **/
Transaction tx=null;
Trv{
                                                     Public class Jtc16B{
                                                     Public static void main(string[]args){
sessionFactory
sf=AHibernateUtil.getSessionFactory();
                                                     Transaction tx=null;
session session=sf.openSession();
                                                     Try{
tx=session.beginTransaction();
                                                     sessionFactory sf=AHibernate
                                                     Util.gerSessionFactory();
//Same as Jtc15A
                                                     Session session=sf.openSession();
                                                     Tx=session.beginTransaction();
Tx.commit();
Session.close();
                                                     //same as Jtc15B
System.out. printin("record Inserted");
}catch(Exception e) {
                                                     Tx.commit();
e.printStackTrace();
                                                     Session.close();
if(tx!=null)
                                                     }catch(Exception e) {
tx.rollback()
                                                     e.printStackTrace();
}}}
                                                     if.rollback();
                                                     }}}
```

Java Training Center (No 1 in Training & Placement)

Jtc16c.java	// Same as Jtc15C
Package com.jtcindia.hibernate; Import org.hibernate.*; Public class jtc16C{ Public static void main(string[]args) { Transaction tx=null; Try{ sessionFactory sf=AHibernateUtil.get SessionFactory(); Sessionsession=sf.openSession(); Tx=session.beginTransaction();	Tx.commit(); Session.close(); }catch(Exception e) {e.printStackTrace(); If(tx!=null) tx.rollback(); } } }

(No 1 in Training & Placement)

Customer.java Request .java Package com.jtcindia.hibernate; Package com.jtcindia.hibernate; Import javax.persistence.*; Import java.util.collection; Import javax.persistence.*; *@Author:Som PrakashRai *@Author:Som PrakashRai *@company:java training center *@company:java training center *@Visit :www.jtcindia.org **/ *@Visit :www.jtcindia.org **/ @Entity @Table(name="itccustomers") @Entity @Table(name="jtccustomers") Public class customer { Public class customer { @id @Column(name="cid") @id @Column(name="cid") @GeneratedValue(strategy=generationType.AUTO) @GeneratedValue(strategy=generationType.AUTO) Private integer cid: Private integer cid; @Column(name="reqDate") @Column(name="firstName") Private string reqDate; Private string firstName; @column(name="description") @column(name="lastName") Private string description; Private string lastName; @column(name="status") @column(name="email") Private string status; Private string email; @many To One @column(name="phone") @join Private long phone; Column(name="cid",referencedColumnName="cid") Private customer customer; Public customer() {} @OneToMany (mappedBy="customer") Private collection<Request>requests; Public customer (string reqDate, string description, Public customer() {} String status){ Public customer (string firstName, string lastName, This. reqDate = reqDate; String email, long phone){ This. description = description; This.firstName=firstName; This. status = status; This. lastName=lastName: This.email=email; //setters and getters @override This.phone=phone; Public string to string() { Return cid+","+ reqDate +","+ description +","+ //setters and getters @override status Public string to string() { cid+","+firstName+","+lastName+","+email+","+phone;

(No 1 in Training & Placement)

Many t	too ma	nv Bi-I	Direction	nal Ma	apping

• Consider the relationship between customer and account.

One customer can place many account.

One request belongs to one customer customers.

consider the relationship between student and course

One student can join for many courses.

One course can have many students

B) Tables Required

1) itccustomers

Sid	Sname	Email	Phone

2) Jtccourses

cid	Cname	Duration	Cost

3) Stu coo

cid	Sid

Example with hibernate core

B) Hibernate persistence classes

1. Student.java

```
Public class student {
Private set <course> courses;
......
}
```

2. Course.java

Public class course {

Private set<student>students;

.....}

(No 1 in Training & Placement)

Hibernate mapping document

```
Student.hbm.xml
<a href="chibernate-mapping-package="com.jtcindia.hibernate">
<class name="Student"table="jtcstudents">
<id name="sid"column="sid"type="int">
<generator coass="increment"/>
</id>
cproperty name="sname"/>
cproperty name="email"/>
cproperty name="phone"type=long"/>
<set name="courses"table="stu cou">
<key column="sid"/>
<many-to-many class="course"column=column="cid"/>
</set>
</class>
</hibernate-mapping>
Course.hbm.xml
<hibernate-mapping package="com.jtcindia.hibernate">
<class name="course"table="jtcstudents">
<id name="cid"column=" cid"type="int">
<generator coass="increment"/>
</id>
cproperty name="cname"/>
cproperty name="duration"type=int"/>
cproperty name="cost"type=stu cou"/>
<key column="sid"/>
<many-to-many class="course" student=column="sid"/>
</set>
</class>
</hibernate-mapping>
```

C) Hibernate mapping Document

Jtc17: .Files required

Jtc17A. java	Jtc17B. java
Jtc17c. java	Student. Java
Course. Java	Student.java
Course. Hbm. Xml	CHibernate Util. java
Hibernate. Cfg. Xml	

(No 1 in Training & Placement)

JTC17A.java JTC17B.java package com.jtcicindia.hibernate; package com.jtcicindia.hibernate; import java.util.*; import java.util.*; import org.hibernate.*; import org.hibernate.*; *@Author:Som PrakashRai *@Author:Som PrakashRai *@Company:Java Training Center *@Company:Java Training Center *@Visit :www.jtcindia.org *@Visit :www.jtcindia.org **/ **/ public class Jtc17A{ public class Jtc17B{ public static void main(string[]args){ public static void main(string[]args){ Transaction tx=null: Transaction tx=null: try{ try{ SessionFactory SessionFactory sf=CHibernateUtil.getSessionFactory(); sf=CHibernateUtil.getSessionFactory(); Session session=sf.openSession(); Session session=sf.openSession(); tx=session.beginTransaction(); tx=session.beginTransaction(); Student s1= new student("Som","Som@jtc.com",9999) Student s1=(student)session.load(student.class,1); Student s2= new student System.out.printIn(**student info") (Prakash", prakash@jtc.com, System.out.printin(s1); System.out.prinitin(**courses info)(8888): Student s3=new Set<course> cous=s1.getCourses(); student(prakash", prakash@jtc.com,7777); For (course c:cous)(Session.save(s1); session.save(s2); System.out.printin©; Session.save(s3); Course c1= new course("java",new Integer(9),new double(999); Tx.commit(); Course c2=new course("jsp",new integer(8),new double(888); }catech(exception e) { Course C3=new course("EJB .new Integer(7),new e.printStackTrace(); Double(777); if(tx!=null) Session.save(s1); session.save(s2); tx.rollback(); Session.save(s3); Set<course>cs2=new hashSet<course>(); } Cs2.add(c1); cs2.add(c2); cs2.add(c3);} S2.setCourses(cs2); Tx.commit(); Session.close(); System .out.println(record inserted"); }catch(Exception e) {e.printStack Trace(); If(tx!=null)tx.rollback();

Student.java

(No 1 in Training & Placement)

Jtc17c.java package com.jtcicindia.hibernate; import java.util.*; import org.hibernate.*; *@Author:Som PrakashRai *@Company:Java Training Center *@Visit :www.jtcindia.org **/ public class Jtc17A{ public static void main(string[]args){ Transaction tx=null; try{ SessionFactory sf=CHibernateUtil.getSessionFactory(); Session session=sf.openSession(); tx=session.beginTransaction(); course c = (course)session.load(course.class,1); syatem.out.println(**courses info"); system.out.println©; system.out.printin(**student info"); set<student> stus=c.get student() for (student s; stus) system.out.println(s); tx.commit(); session.close(); }catch9Exception e) { e.printStack Trace(); if(tx!=null) tx.rollback(); }}}

```
package com.jtcicindia.hibernate;
*@Author:Som PrakashRai
*@Company:Java Training Center
*@Visit
              :www.jtcindia.org
**/
public class customer{
private integer sid;
private string sname;
private string email;
private string phone;
private set<course>courses;
public student() { }
public student (string sname, string email, long
phonje)
this.sname=sname;
this.email.=email:
this.phone=phone;
//setters and getters
@override
Public string to string() {
Return sid+","+sname+","=email+","d+phone;
}}}
```

(No 1 in Training & Placement)

Example with hibernate annotation

B) Hibernate persistence classes

1. Student.java

2. Address.java

```
@Entity
@ Table (name=" requests")
Public class requests {
.....
     @Many to many (mapped by = "courses");
     Private set<Student>students;
     ......
}
```

Jtc18: .Files required

Jtc18A. java	Jtc18B. java
Jtc18c. java	Student. java
Course. java	AHibernate Util. java
Hibernate.Cfg.Xml	

(No 1 in Training & Placement)

```
Student.java
Package com.jtcindia.hibernate;
Import java.util.*:
Import javax.persistence.*;
*@Author:Som PrakashRai
*@company:java Training center
*@Visit
            :www.jtcindia.org
**/
@Entity
@table(name="jtcstudents")
Public class customer{
@id
@GeneratedValue(strategy=GenerationType.AUT
@Column(name="sname")
Private sname
@Column(name="email")
Private string email;
@Column(name="phone")
Private string phone;
Public customer() { }
Public Customer(String cname, String email, long
phone, String city, double bal){
this.cname=cname;
this.email=email;
this.phone=phone;
     //Setters and Getters
@override
Public string to string() {
Return sid+","+sname+","=email+","d+phone;
```

```
Student.java
Package com.jtcindia.hibernate;
Import java.util.*:
Import javax.persistence.*;
*@Author:Som PrakashRai
*@company:java Training center
*@Visit
            :www.jtcindia.org
**/
@Entity
@table(name="jtccourses")
Public class customer{
@GeneratedValue(strategy=GenerationType.AUT
@Column(name="cname")
Private cname
@Column(name="duration")
Private string duration;
@Column(name="cost")
Private string cost;
Public customer() {}
Public Customer(String cname, String email, long
phone, String city, double bal) {
this.cname=cname;
this.email=email;
this.phone=phone;
     //Setters and Getters
@override
Public string to string() {
Return sid+","+sname+","=email+","d+phone;
}
}
```

	Java Training Center (No 1 in Training & Placement)		