

Java Training Center

(No 1 in Training & Placement)



Java Training Center

(No.1 in Training & Placement)

Hibernate-2 Doc.

Master the Content...

Author

Som Prakash Rai

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 |
|-----|------------|-----------|--|-------|-----|
| 1 | Som | Prakash | Som@jtcindia.org | 3333 | 1 |

Addresses

| Aid | Street | City | State |
|-----|--------|-------|-------|
| 1 | SEC-2 | Noida | UP |

Example with hibernate core

B) Hibernate persistence classes

1) Customer.java

```
Public class customer {  
    .....  
    Address;  
}
```

2) Address. java

```
Classaddress {  
    .....  
}
```

C) Hibernate mapping document

1) Customer.hbm.xml

```
<hibernate-mapping package="com.jtcindia.hibernate">  
<class name="Student"table="mystudents">  
<id name="cid"column="cid"type="int">  
<generator coass="increment"/>  
</id>  
<property name="firstName"column="firstName"/>  
<property name="lastName"column="firstName"/>  
<property name="email"column="firstName"/>  
<property name="phone"column="firstName"/>  
</class>  
<hibernate-mapping package="com.jtcindia.hibernate">  
<class name="address"table="addresses">  
<id name="aid"column=" aid"type="int">  
<generator coass="increment"/>  
</id>  
<property name="street"/>  
<property name="city"/>  
<property name="status"/>  
</class>  
</hibernate-mapping>
```

Jtc11: Files required

| | |
|----------------|---------------|
| Jtc11A. java | Jtc11. java |
| Customer. Java | Address. java |

Java Training Center

(No 1 in Training & Placement)

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

Java Training Center

(No 1 in Training & Placement)

| Jtc11A.java | Jtc11B.java |
|---|--|
| <pre>Package com.jtcindia.hibernate; Import java.util.*; /* * @Author:Som PrakashRai * @company:java Training Center * @Visit : www.jtcindia.org **/ Public class Jtc11A{ Public static void main(string{ } args){ Transation tx=null; Try{ sessionFactory sf=Chibernate Util.get Session Factory(); Session session=sf.openSession(); Tx=session.beginTransaction(); Customer cust=new customer("Som","prakash","Som@jtc","1234"); Session.save(cust); Address add=new address("sec-2","Noida", "Uttarpradesh"); Session.save(add); Cust.setAddress(add); Tx.commit(); Session.close(); System.out.println("record inserted"); }catch(Exception e) { e.printStackTrace(); if (tx!=null) tx.rollback(); }}}</pre> | <pre>Package com.jtcindia.hibernate; Import java.util.*; /* * @Author:Som PrakashRai * @company:java Training Center * @Visit : www.jtcindia.org **/ Public class Jtc11B{ Public static void main(string{ } args){ Transation tx=null; Try{ sessionFactory sf=Chibernate Util.getSessionFactory(); session session=sf.openSession(); tx=session.begintransaction(); customer cust=(customer) session.load(customer.class,1); system.out.println(cust.getCid()+"\t"= cust.getFirstName()+"\t"=cust.getLastName()+"\t"+ cust.getEmail()+"\t"+cust.getPhone()); address add= cust.get address(); system.out.printin(add.getAid()+"\t"+add.getCity() +"\t"+add.getStreet()+"\t"+add.getState()); Address add1(Address)session.load(Address.class,1); system.out.printin(add.getAid()+"\t"+add.getCity() +"\t"+add.getStreet()+"\t"+add.getState()); Tx.commit(); Session.close(); System.out.println("record inserted"); }catch(Exception e) { e.printStackTrace(); if (tx!=null) tx.rollback(); }}}</pre> |

Java Training Center

(No 1 in Training & Placement)

| | |
|--|---|
| <pre> Customer.java Package com.jtcindia.hibernate; Import java.util.*; /* * @Author:Som PrakashRai * @company:java Training Center * @Visit : www.jtcindia.org */ Public class customer { Int cid; String firstName; String last name; String email; String phone; Address address; Public customer() { } Public customer (string firstName,string lastName,string email,string phone){ This.firstName=firstName; This.lastName=lastName; This.email=email; This.phone=phone; } //setters and getters } </pre> | <pre> Customer.hbm.xml <hibernate-mapping package="com.jtcindia.hibernate"> <class name="customer"table="mycustomers"> <id name="cid"column="cid"type="int"> <generator class="increment"/> </id> <property name="firstName"column="firstName"/> <property name="lastName"column="lastname"/> <property name="email"column="email"/> <property name="phone"column="phone"/> <many-to-one name="address"class="Address" column ="aid"/> </class> </hibernate-mapping> </pre> |
| <pre> Address.java Package com.jtcindia.hibernate; Import java.util.*; /* * @Author:Som PrakashRai * @company:java Training Center * @Visit : www.jtcindia.org */ Public class customer { Int aid; String street; 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 } </pre> | <pre> Address.hbm.xml <hibernate-mapping package="com.jtcindia.hibernate"> <class name="customer"table="addresses"> <id name="cid"column="aid"type="int"> <generator class="increment"/> </id> <property name="street"/> <property name="city"/> <property name="state"/> </class> </hibernate-mapping> </pre> |

Example with hibernate annotation

B) Hibernate persistence classes

Java Training Center

(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 required

| | |
|-----------------------|---------------------|
| Jtc12A. java | Jtc12B. java |
| Customer. Java | Address. Java |
| AHibernate Util. java | Hibernate. Cfg. xml |

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

Java Training Center

(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
    city){
    this.street=street;
    this.city=city;
    this.state=state;
    }
        //Setters and Getters
    }
```


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">  
<generator coass="increment"/>  
</id>  
<discriminator"firstName"column=" firstName"/>  
<discriminator"lastName"column=" lastName"/>  
<discriminator"email"column=" email"/>  
<discriminator"phone"column=" phone"/>  
<many-to-one name="address"class="Address"column=aid"/>  
</class>
```

```
</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 | |

Java Training Center

(No 1 in Training & Placement)

| JTC13A.java | JTC13B.java |
|---|---|
| <pre>package com.jtcindia.hibernate; import org.hibernate.*; public class Jtc13A{ 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("Som","Som","Som@jtc ","9036"); session.save(cust); address add=new address("sec-2","Noida", "Uttarpradesh"); session.save(add); cust.setAddress(add); tx.commit(); system.out.println("record inserted"); }catch(Exception e) { e.prinStackTrace(); if(tx!=null) tx.rollback(); } } }</pre> | <pre>package com.jtcindia.hibernate; import org.hibernate.*; public class Jtc13B{ 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,2); System.out.println(cust.getCid()+"\t"+cust.getFirst Name() +"\t"+cust.getLastName()+"\t"+add.getCity()+"\t"+ cust.getPhone()); Address add=cust.getAddress(); System.out.println(add.getAid()+"\t"+add.getCity()+ "\t"+add.getstreet()+"\t"+add.getState()); Tx.commit(); Session.close(); }catch(Exception e){e.printStackTrace(); It(tx!=null)tx.rollback(); } } }</pre> |

Java Training Center

(No.1 in Training & Placement)

| | |
|--|--|
| <p>JTC13C.java</p> <pre> package com.jtcindia.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,2); System.out.println(cust.getCid()+"\t"+cust.getFirst Name() +"\t"+cust.getLastName()+"\t"+add.getCity()+"\t"+ cust.getPhone()); Address add=cust.getAddress(); System.out.println(add.getAid()+"\t"+add.getCity()+ "\t"+add.getstreet()+"\t"+add.getState()); Tx.commit(); Session.close(); }catch(Exception e){e.printStackTrace(); It(tx!=null)tx.rollback(); } } }</pre> | <p>Customer.java</p> <pre> package com.jtcindia.hibernate; /* * @Author:Som PrakashRai * @Company:Java Training Center * @Visit :www.jtcindia.org */ public class customer{ private integer cid; private string first name; private string lastName; private string email; private string phone; private address address; public customer() {} public customer(string firstName, string lastName, string email, string phone){ this.lfirstNme=firstName; this.lastName=lastName; this.email=email; this.phone=phone; } //seters and getters }</pre> |
| <p>Address.java</p> <pre> package com.jtcindia.hibernate; /* * @Author:Som PrakashRai * @Company:Java Training Center * @Visit :www.jtcindia.org */ public class customer{ private integer aid; private string street; private string city; private string state; Customer customer; //Custructors //setters and getters } } }</pre> | <p>Customer.hbm.xml</p> <pre> <hibernate-mapping package="com.jtcindia.hibernate"> <class name="customer"table="table="mycustomers"> <id name="cid"column="cid"type="int"> <generator class="increment"/> </id> <property name="firstName"column="firstName"/> <property name="lastName" column="lastName"/> <property name="email"column="phone"/> <many-to-one name="address"class="address"column="aid"/> </class> </hibernate-mapping></pre> |

Java Training Center

(No 1 in Training & Placement)

| | |
|--|--|
| | |
|--|--|

| | |
|---|--|
| <u>Address.hbm.xml</u> <pre><hibernate-mapping package="com.jtcindia.hibernate"> <class name="Address"table="addresses"> <id name="aid"column="aid" type="int"> <generator class="increment"/> </id> <property name="street"/> <property name="city"/> <property name="state"/> <one-to-one name="customer" class="customer"/> </class> </hibernate-mapping></pre> | |
|---|--|

Example with hibernate annotation

B) Hibernate persistence classes

1. Customer.java

@Entity

@ Table (name="mystudents")

Public class student {

Java Training Center

(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 <pre>package com.jtcindia.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.printStackTrace(); if(tx!=null) tx.rollback(); }}}</pre> | JTC14B.java <pre>package com.jtcindia.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() +"\t"+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(); }}}</pre> |
| JTC13C.java <pre>package com.jtcindia.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();</pre> | <pre>tx=session.beginTransaction(); Customer cust=(customer) Session.load(customer.class,2); System.out.println(cust.getCid()+"\t"+cust.get First Name() +"\t"+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();</pre> |

Java Training Center

(No 1 in Training & Placement)

| | |
|---|--|
| <pre>Customer.java SessionFactory sf=SessionFactory.openSession(); 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 }</pre> | <pre>Address.java Exception e){e.printStackTrace(); 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; @Column(name="aid") Private string cname; Public customer() { } Public Customer(String street, String state, String city){ this.street=street; this.city=city; this.state=state; } //Setters and Getters }</pre> |
|---|--|

Java Training Center

(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">
```

Java Training Center

(No 1 in Training & Placement)

```
<class name="Request"table="requests">
<id name="reqId"column="air" reqId ="int">
<generator class="increment"/>
</id>
<discriminator column="reqDate"/>
<property name="description"/>
<property name="status"/>
<many-to-one name="customer" class="Customer" column="cid"/>
</class>
</hibernate-mapping>
```

Jtc15: .Files required

| | |
|---------------------|-----------------------|
| Jtc15A. java | Jtc15B. java |
| Jtc15c. java | Customer. java |
| Request. Java | Customer. Hbm. xml |
| Request. Hbm. Xml | CHibernate Util. java |
| Hibernate. Cfg. Xml | |

Java Training Center

(No 1 in Training & Placement)

```
1.Jtc15A.java
package com.jtcindia.hibernate;
import org.hibernate.*;
public class Jtc13A{
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("Prakash","rai",prakash@jtc.com,9999);
session.save(cust);
Request req1=new request("22/07/2015","hello1","
ok1");
Request req2=new request("23/07/2015","hello2","
ok2");
Session.save(req1);    session.save(req2);
Req1.setCustomer(ust);    req2.setCustomer(cust);
Tx.commit();
Session.close();
System.out.println("Record Inserted");
}catch(Exception e){ e.printStackTrace();
If (tx !=null) tx.rollback();
}}}
```

```
Jtc15B.java
package com.jtcindia.hibernate;
import org.hibernate.*;
public class Jtc13A{
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("**Customer Info");
System.out.println(cust);
System.out.println("**Requests info");
Collection<Request>col=cust.getRequests();
For(request req:col)
System.out.println(req);
Tx.commit();
Session.close();
}catch(Exception e){e.printStackTrace();
If(tx!=null)tx.rollback();
}
}
}
```

Java Training Center

(No 1 in Training & Placement)

Jtc15C.java

```
package com.jtcindia.hibernate;
import org.hibernate.*;
public class Jtc13A{
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("**Customer Info");
System.out.println(cust);
System.out.println("**Requests info");
Collection<Request>col=cust.getRequests();
For(request req:col)
System.out.println(req);
Tx.commit();
Session.close();
}catch(Exception e)(e.prinStackTrace());
If(tx!=null)tx.rollback();
}
}
}
```

Request.java

```
Package com.jtcindia.hibernate;
Public class request{
Private Integer reqId;    private string reqData
Private String description Private string status;
Private string description;
Private Customer customer;

Public request() {}

Public customer(string reqData, String description ,
string status){
This. reqData = reqData;
This. description = description ;
This. status = status;
}
//setters and getters
@Override
Public string to string() {
Returncid+"", "+ reqData +", "+ description +", "+ status;
}}}
```

Customer.java

```
Package com.jtcindia.hibernate;
Import java.util.set;
Public class customer{
Private Integer cid;    private string fistName
Private String lastName Private string email;
Private long phone;
Private set<request>requests;

Public Customer() {}

Public customer(string firstName, String
lastName,string email, long phone){
This.firstName=firstName;
This.lastName=lastName;
This.email=email;
This.phone=phone;
}
//setters and getters
@Override
Public string to string() {
Return
cid+", "+firstName+", "+LastName+", "+email+", "+ph
one;
}}}
```

Customer.hbm.xml

```
<hibernate-mapping
package="com.jtcindia.hibernate">
<class
name="customer"table="table=jtccustomers">
<id name="cid"column="cid"type="int">
<generator class="increment"/>
</id>
<property name="firstName"column="firstName"/>
<property name="lastName" column="lastName"/>
<property name="email"column="phone"/>
<property name="phone" column="phone"/>
<set name="requests">
<key column="cid"/>
<one-to-many class="request"/>
</set>

</class>
</hibernate-mapping>
```

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

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 |

Java Training Center

(No 1 in Training & Placement)

Hibernate. Cfg. Xml

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

Java Training Center

(No 1 in Training & Placement)

| | |
|---|--|
| Jtc16c.java | // Same as Jtc15C |
| <pre>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(); Session session=sf.openSession(); Tx=session.beginTransaction();</pre> | <pre>Tx.commit(); Session.close(); }catch(Exception e) {e.printStackTrace(); If(tx!=null) tx.rollback(); } } }</pre> |

Java Training Center

(No 1 in Training & Placement)

Customer.java

```
Package com.jtcindia.hibernate;
Import java.util.collection;
Import javax.persistence.*;
/*
 * @Author:Som PrakashRai
 * @company:java training center
 * @Visit      :www.jtcindia.org
 */
@Entity
@Table(name="jtccustomers")
Public class customer {
    @id
    @Column(name="cid")
    @GeneratedValue(strategy=generationType.AUTO)
    Private integer cid;

    @Column(name="firstName")
    Private string firstName;

    @column(name="lastName")
    Private string lastName;

    @column(name="email")
    Private string email;

    @column(name="phone")
    Private long phone;

    @OneToMany (mappedBy="customer")
    Private collection<Request>requests;
    Public customer() {}
    Public customer (string firstName, string lastName,
    String email, long phone){
    This.firstName=firstName;
    This.lastName=lastName;
    This.email=email;
    This.phone=phone;
    }
    //setters and getters
    @override
    Public string to string() {
    Return
    cid+"",""+firstName+"",""+lastName+"",""+email+"",""+phone;
    }
    }
```

Request .java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
 * @Author:Som PrakashRai
 * @company:java training center
 * @Visit      :www.jtcindia.org
 */
@Entity
@Table(name="jtccustomers")
Public class customer {
    @id
    @Column(name="cid")
    @GeneratedValue(strategy=generationType.AUTO)
    Private integer cid;

    @Column(name="reqDate")
    Private string reqDate;

    @column(name="description")
    Private string description;

    @column(name="status")
    Private string status;

    @many To One
    @join
    Column(name="cid",referencedColumnName="cid")
    Private customer customer;
    Public customer() {}
    Public customer (string reqDate, string description,
    String status){
    This. reqDate = reqDate;
    This. description = description;
    This. status = status;
    }
    //setters and getters
    @override
    Public string to string() {
    Return cid+"",""+ reqDate +"",""+ description +"",""+
    status
    }
    }
```


Many too many Bi-Directional Mapping:

- 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) jtccustomers

| 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;  
  
    .....  
}
```

Hibernate mapping document

Student.hbm.xml

```
<hibernate-mapping package="com.jtcindia.hibernate">
<class name="Student"table="jtcstudents">
<id name="sid"column="sid"type="int">
<generator coass="increment"/>
</id>
<property name="sname"/>
<property name="email"/>
<property 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>
<property name="cname"/>
<property name="duration"type=int"/>
<property 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 | |

Java Training Center

(No 1 in Training & Placement)

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

Java Training Center

(No 1 in Training & Placement)

Jtc17c.java

```
package com.jtcindia.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.printStackTrace();
if(tx!=null)
tx.rollback();
}}}
```

Student.java

```
package com.jtcindia.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;
}}}
```

Example with hibernate annotation

B) Hibernate persistence classes

1. Student.java

```
@Entity
@Table (name="jtccustomers")
Public class student {
.....
@Many too many
@Join table (name = "stu coo",
Join columns=
    @Join column (name ="sid". referenced column name ="sid"),
Inverse join columns=
    @join column (name ="cid" referenced Column Name ="cid")
Private set<course>courses;
.....
}
```

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 | |

Java Training Center

(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.AUTO)
    @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 toString() {
    Return sid+","+sname+","+email+","+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{
    @id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @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 toString() {
    Return sid+","+sname+","+email+","+phone;
    }
    }
    }
```

