

# Many to Many Mapping in hibernate by map Example (using xml file)

We can map many to many relation either using set, bag, map etc. Here, we are going to use map for many-to-many mapping. In such case, three tables will be created.

## Example of Many to Many Mapping

You need to create following pages for mapping map elements.

- Question.java
- User.java
- question.hbm.xml
- user.hbm.xml
- hibernate.cfg.xml
- StoreTest.java
- FetchTest.java

### Question.java

```
package com.javatpoint;

import java.util.Map;

public class Question {
    private int id;
    private String name;
    private Map<String,User> answers;

    public Question() {}
    public Question(String name, Map<String, User> answers) {
        super();
        this.name = name;
        this.answers = answers;
    }
}
```

```
public int getId() {  
    return id;  
}  
public void setId(int id) {  
    this.id = id;  
}  
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public Map<String, User> getAnswers() {  
    return answers;  
}  
public void setAnswers(Map<String, User> answers) {  
    this.answers = answers;  
}  
  
}
```

### User.java

```
package com.javatpoint;  
  
public class User {  
    private int id;  
    private String username,email,country;  
  
    public User() {}  
    public User(String username, String email, String country) {  
        super();  
        this.username = username;  
        this.email = email;  
        this.country = country;  
    }  
}
```

```
public int getId() {  
    return id;  
}  
  
public void setId(int id) {  
    this.id = id;  
}  
  
public String getUsername() {  
    return username;  
}  
  
public void setUsername(String username) {  
    this.username = username;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getCountry() {  
    return country;  
}  
  
public void setCountry(String country) {  
    this.country = country;  
}  
}
```

#### question.hbm.xml

```
<?xml version='1.0' encoding='UTF-8'?>  
<!DOCTYPE hibernate-mapping PUBLIC
```

```
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
```

```
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
```

```
<hibernate-mapping>
<class name="com.javatpoint.Question" table="question738">
<id name="id">
<generator class="native"></generator>
</id>
<property name="name"></property>

<map name="answers" table="answer738" cascade="all">
<key column="questionid"></key>
<index column="answer" type="string"></index>
<many-to-many class="com.javatpoint.User" column="userid"></many-to-many>
</map>
</class>

</hibernate-mapping>
```

### user.hbm.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
<class name="com.javatpoint.User" table="user738">
<id name="id">
<generator class="native"></generator>
</id>
<property name="username"></property>
<property name="email"></property>
<property name="country"></property>
</class>

</hibernate-mapping>
```

**hibernate.cfg.xml**

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<!-- Generated by MyEclipse Hibernate Tools.           -->
<hibernate-configuration>

    <session-factory>
        <property name="hbm2ddl.auto">update</property>
        <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
        <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
        <property name="connection.username">system</property>
        <property name="connection.password">oracle</property>
        <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>

        <mapping resource="question.hbm.xml"/>
        <mapping resource="user.hbm.xml"/>
    </session-factory>

</hibernate-configuration>
```

**StoreTest.java**

```
package com.javatpoint;

import java.util.HashMap;
import org.hibernate.*;
import org.hibernate.cfg.*;

public class StoreTest {
    public static void main(String[] args) {
        Session session=new Configuration().configure().buildSessionFactory().openSession();
        Transaction tx=session.beginTransaction();

        HashMap<String,User> map1=new HashMap<String,User>();
```

```
map1.put("java is a programming language",
new User("John Milton","john@gmail.com","usa"));

map1.put("java is a platform",
new User("Ashok Kumar","ashok@gmail.com","india"));

HashMap<String,User> map2=new HashMap<String,User>();
map2.put("servlet technology is a server side programming",

new User("John Milton","john@gmail.com","usa"));
map2.put("Servlet is an Interface",
new User("Ashok Kumar","ashok@gmail.com","india"));

Question question1=new Question("What is java?",map1);
Question question2=new Question("What is servlet?",map2);

session.persist(question1);
session.persist(question2);

tx.commit();
session.close();
System.out.println("successfully stored");
}
}
```

### FetchTest.java

```
package com.javatpoint;
import java.util.*;
import org.hibernate.*;
import org.hibernate.cfg.*;
public class FetchTest {
public static void main(String[] args) {
    Session session=new Configuration().configure().buildSessionFactory().openSession();
    Query query=session.createQuery("from Question ");
    List<Question> list=query.list();
```

```
Iterator<Question> iterator=list.iterator();
while(iterator.hasNext()){
    Question question=iterator.next();
    System.out.println("question id:"+question.getId());
    System.out.println("question name:"+question.getName());
    System.out.println("answers.....");
    Map<String,User> map=question.getAnswers();
    Set<Map.Entry<String,User>> set=map.entrySet();

    Iterator<Map.Entry<String,User>> iteratoranswer=set.iterator();
    while(iteratoranswer.hasNext()){
        Map.Entry<String,User> entry=(Map.Entry<String,User>)iteratoranswer.next();
        System.out.println("answer name:"+entry.getKey());
        System.out.println("answer posted by.....");
        User user=entry.getValue();
        System.out.println("username:"+user.getUsername());
        System.out.println("user emailid:"+user.getEmail());
        System.out.println("user country:"+user.getCountry());
    }
}
session.close();
}
```

download this hibernate example(developed using MyEclipse IDE)

[<<prev](#)[next>>](#)[Share 3](#)