# Hibernate Many to Many Example using XML

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

## Example of Many to Many Mapping

In this example, we will generate a many to many relation between questions and answers using list.

### Create the Persistent class

There are two persistent classes Question.java and Answer.java. Question class contains Answer class reference and vice versa.

**Question.java**

**package** com.javatpoint;

**import** java.util.List;

**public** **class** Question {

**private** **int** id;

**private** String qname;

**private** List<Answer> answers;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getQname() {

**return** qname;

}

**public** **void** setQname(String qname) {

**this**.qname = qname;

}

**public** List<Answer> getAnswers() {

**return** answers;

}

**public** **void** setAnswers(List<Answer> answers) {

**this**.answers = answers;

}

}

**Answer.java**

**package** com.javatpoint;

**import** java.util.\*;

**public** **class** Answer {

**private** **int** id;

**private** String answername;

**private** String postedBy;

**private** List<Question> questions;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getAnswername() {

**return** answername;

}

**public** **void** setAnswername(String answername) {

**this**.answername = answername;

}

**public** String getPostedBy() {

**return** postedBy;

}

**public** **void** setPostedBy(String postedBy) {

**this**.postedBy = postedBy;

}

**public** List<Question> getQuestions() {

**return** questions;

}

**public** **void** setQuestions(List<Question> questions) {

**this**.questions = questions;

}

}

### 2) Create the Mapping file for the persistent class

Here, we have created the question.hbm.xml and answer.hbm.xml file for defining the list.

**question.hbm.xml**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE hibernate-mapping PUBLIC

          "-//Hibernate/Hibernate Mapping DTD 5.3//EN"

          "http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd">

<hibernate-mapping>

  <**class** name="com.javatpoint.Question" table="ques1911">

        <id name="id" type="int">

            <column name="q\_id" />

            <generator **class**="increment" />

        </id>

        <property name="qname" />

        <list name="answers" table="ques\_ans1911" fetch="select" cascade="all">

            <key column="q\_id" />

               <index column="type"></index>

            <many-to-many **class**="com.javatpoint.Answer" column="ans\_id" />

        </list>

    </**class**>

</hibernate-mapping>

**answer.hbm.xml**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE hibernate-mapping PUBLIC

          "-//Hibernate/Hibernate Mapping DTD 5.3//EN"

          "http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd">

<hibernate-mapping>

<**class** name="com.javatpoint.Answer" table="ans1911">

        <id name="id" type="int">

            <column name="ans\_id" />

            <generator **class**="increment" />

        </id>

        <property name="answername"  />

        <property name="postedBy" />

    </**class**>

</hibernate-mapping>

### 3) Create the configuration file

This file contains information about the database and mapping file.

**hibernate.cfg.xml**

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE hibernate-configuration PUBLIC

          "-//Hibernate/Hibernate Configuration DTD 5.3//EN"

          "http://hibernate.sourceforge.net/hibernate-configuration-5.3.dtd">

<hibernate-configuration>

    <session-factory>

        <property name="hbm2ddl.auto">create</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">jtp</property>

        <property name="connection.driver\_class">oracle.jdbc.driver.OracleDriver</property>

    <mapping resource="question.hbm.xml"/>

    <mapping resource="answer.hbm.xml"/>

    </session-factory>

</hibernate-configuration>

### 4) Create the class to store the data

**StoreData.java**

**package** com.javatpoint;

**import** java.util.ArrayList;

**import** org.hibernate.\*;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** StoreData {

**public** **static** **void** main(String[] args) {

    StandardServiceRegistry ssr=**new** StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();

    Metadata meta=**new** MetadataSources(ssr).getMetadataBuilder().build();

    SessionFactory factory=meta.getSessionFactoryBuilder().build();

    Session session=factory.openSession();

    Transaction t=session.beginTransaction();

    Answer ans1=**new** Answer();

    ans1.setAnswername("Java is a programming language");

    ans1.setPostedBy("Ravi Malik");

    Answer ans2=**new** Answer();

    ans2.setAnswername("Java is a platform");

    ans2.setPostedBy("Sudhir Kumar");

    Question q1=**new** Question();

    q1.setQname("What is Java?");

    ArrayList<Answer> l1=**new** ArrayList<Answer>();

    l1.add(ans1);

    l1.add(ans2);

    q1.setAnswers(l1);

    Answer ans3=**new** Answer();

    ans3.setAnswername("Servlet is an Interface");

    ans3.setPostedBy("Jai Kumar");

    Answer ans4=**new** Answer();

    ans4.setAnswername("Servlet is an API");

    ans4.setPostedBy("Arun");

    Question q2=**new** Question();

    q2.setQname("What is Servlet?");

    ArrayList<Answer> l2=**new** ArrayList<Answer>();

    l2.add(ans3);

    l2.add(ans4);

    q2.setAnswers(l2);

    session.persist(q1);

    session.persist(q2);

    t.commit();

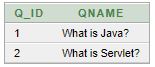
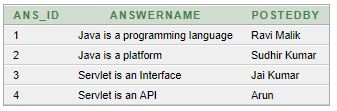
    session.close();

    System.out.println("success");

}

}

### Output

### Download

[Download this Example](https://www.javatpoint.com/src/hb/mtm.zip)