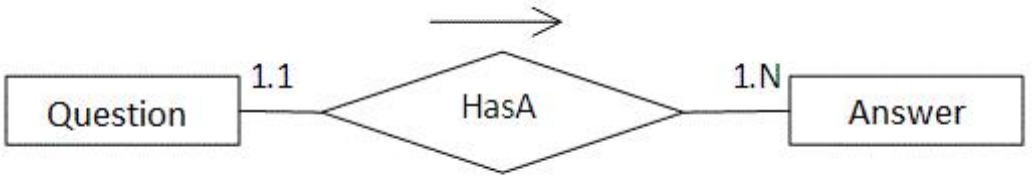
Mapping List in Collection Mapping (using xml file)

If our persistent class has List object, we can map the List easily either by <list> element of class in mapping file or by annotation.

Here, we are using the scenario of Forum where one question has multiple answers.



Let's see how we can implement the list in the mapping file:

```
<class name="com.javatpoint.Question" table="q100">
    ...
      <list name="answers" table="ans100">
      <key column="qid"></key>
      <index column="type"></index>
      <element column="answer" type="string"></element>
      </list>

    ...
</class>
```

List and Map are index based collection, so an extra column will be created in the table for indexing.

Example of mapping list in collection mapping

In this example, we are going to see full example of collection mapping by list. This is the example of List that stores string value not entity reference that is why are going to use **element** instead of **one-to-many** within the list element.

1) create the Persistent class

This persistent class defines properties of the class including List.

```
package com.javatpoint;

import java.util.List;

public class Question {
private int id;
private String qname;
private List<String> answers;

//getters and setters

}
```

2) create the Mapping file for the persistent class

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

```
<?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="q100">
   <id name="id">
     <generator class="increment"></generator>
   </id>
   <property name="qname"></property>

   <list name="answers" table="ans100">
     <key column="qid"></key>
     <index column="type"></index>
     <element column="answer" type="string"></element>
   </list>
```

```
 </class>

</hibernate-mapping>
```

3) create the configuration file

This file contains information about the database and mapping file.

```
<?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"/>
   </session-factory>

</hibernate-configuration>
```

4) Create the class to store the data

In this class we are storing the data of the question class.

```
package com.javatpoint;

import java.util.ArrayList;
```

```
import org.hibernate.*;
import org.hibernate.cfg.*;

public class StoreData {
 public static void main(String[] args) {
    Session session=new Configuration().configure("hibernate.cfg.xml")
                    .buildSessionFactory().openSession();
    Transaction t=session.beginTransaction();

    ArrayList<String> list1=new ArrayList<String>();
    list1.add("java is a programming language");
    list1.add("java is a platform");

    ArrayList<String> list2=new ArrayList<String>();
    list2.add("Servlet is an Interface");
    list2.add("Servlet is an API");

    Question question1=new Question();
    question1.setQname("What is Java?");
    question1.setAnswers(list1);

    Question question2=new Question();
    question2.setQname("What is Servlet?");
    question2.setAnswers(list2);

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

    t.commit();
    session.close();
    System.out.println("success");
 }
}
```

How to fetch the data of List

Here, we have used HQL to fetch all the records of Question class including answers. In such case, it fetches the data from two tables that are functional dependent.

```
package com.javatpoint;

import java.util.*;

import org.hibernate.*;
import org.hibernate.cfg.*;

public class FetchData {
public static void main(String[] args) {

    Session session=new Configuration().configure("hibernate.cfg.xml")
                    .buildSessionFactory().openSession();

    Query query=session.createQuery("from Question");
    List<Question> list=query.list();

    Iterator<Question> itr=list.iterator();
    while(itr.hasNext()){
        Question q=itr.next();
        System.out.println("Question Name: "+q.getQname());

        //printing answers
        List<String> list2=q.getAnswers();
        Iterator<String> itr2=list2.iterator();
        while(itr2.hasNext()){
            System.out.println(itr2.next());
        }

    }
    session.close();
    System.out.println("success");
```

```
}
}
```

download this hibernate example (developed using MyEclipse IDE)
download this hibernate example (developed using Eclipse IDE)

← prev next →