Set One to many

If the persistent class has set object that contains the entity reference, we need to use one-to-many association to map the set element. We can map this list object by either **set**.

It is non-index based and will not allow duplicate elements.

Let's see the persistent class that has set objects. In this case, there can be many answers for a question and each answer may have its own informations that is why we have used set element to represent a collection of answers.

```
package com.javatpoint;


import java.util.List;


public class Question {
private int id;
private String qname;
private Set<Answer> answers;
//getters and setters


}
```

The Answer class has its own informations such as id, answername, postedBy etc.

```
package com.javatpoint;


public class Answer {
private int id;
private String answername;
private String postedBy;
//getters and setters


}
}
```

The Question class has set object that have entity reference (i.e. Answer class object). In such case, we need to use **one-to-many** of set to map this object. Let's see how we can map it.

```
<set name="answers" cascade="all">
        <key column="qid"></key>
        <one-to-many class="com.javatpoint.Answer"/>
</set>
```

Example of mapping set in collection mapping by one to many association

To understand this example, you may see the bag one-to-many relation example. We have changed only bag to set in the hbm file and ArrayList to HashSet in the Store class.

download this hibernate example (developed using myeclipse IDE)

<<prev next>>