*Please note that we haven't explained full-fledged annotation based Hibernate project here. Rather we have focused on explaining how bidirectional many to many association is mapped in Hibernate.*

*To know more about annotation based basic Hibernate project please thoroughly go through project* ***' Hibernate5StandaloneWithFullJavaConfig'*** *if you are not familiar with it.*

**What this project dose ?**

This project explains how bidirectional many to many association is mapped in Hibernate by using annotations.

**A short introduction**

In many-to-many association, source entity has a field that stores collection of target entities. The @*ManyToMany* JPA annotation is used to link the source entity with the target entity.

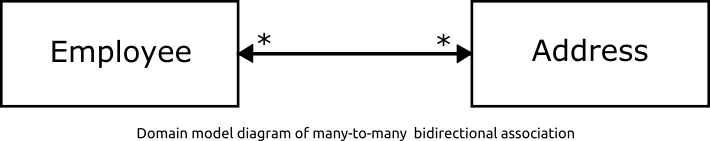
A many-to-many association always uses an intermediate join table to store the association that joins two entities. The join table is defined using the @JoinTable JPA annotation.

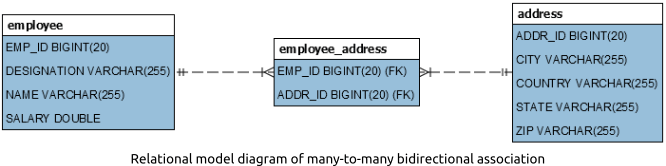
The many-to-many association can be either unidirectional or bidirectional.

In bidirectional association, each entity (i.e. source and target) has a relationship field that refers to each other. We can navigate this type of association from both sides.

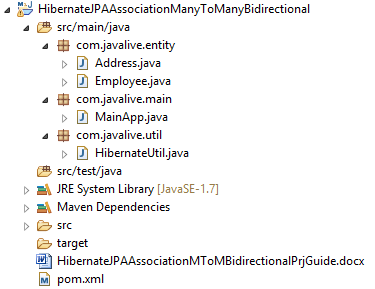
Bidirectional many-to-many association example

Following are the domain model and relational model diagrams of many-to-many bidirectional association.





**Project Structure**

****

**Jar dependencies**

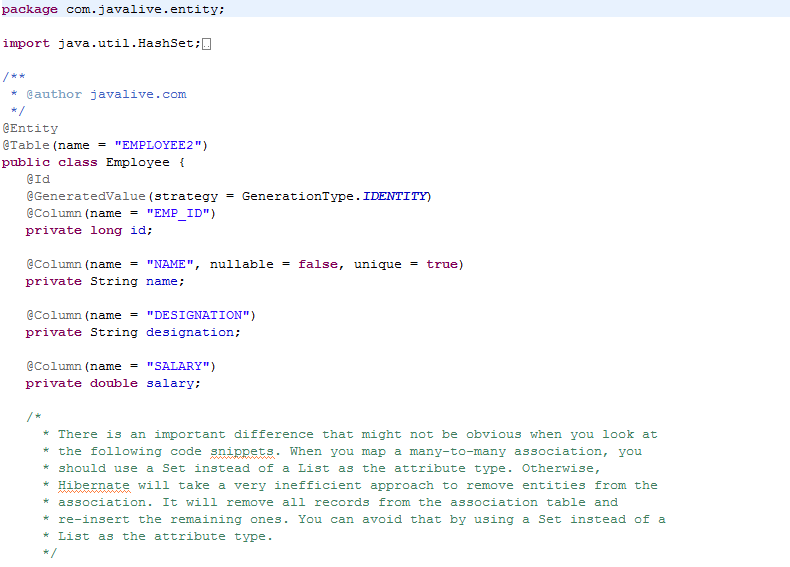
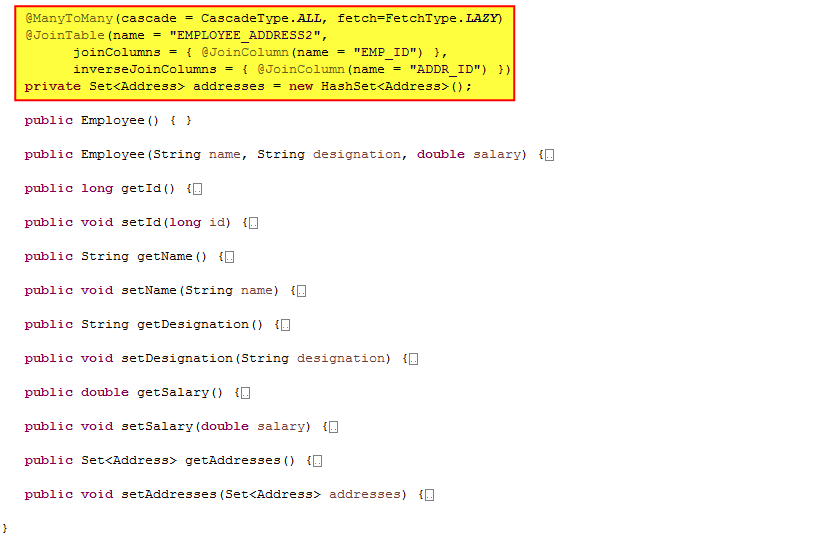
Add the following jar dependencies for Hibernate and MySQL driver in pom.xml file.



**Entity class**

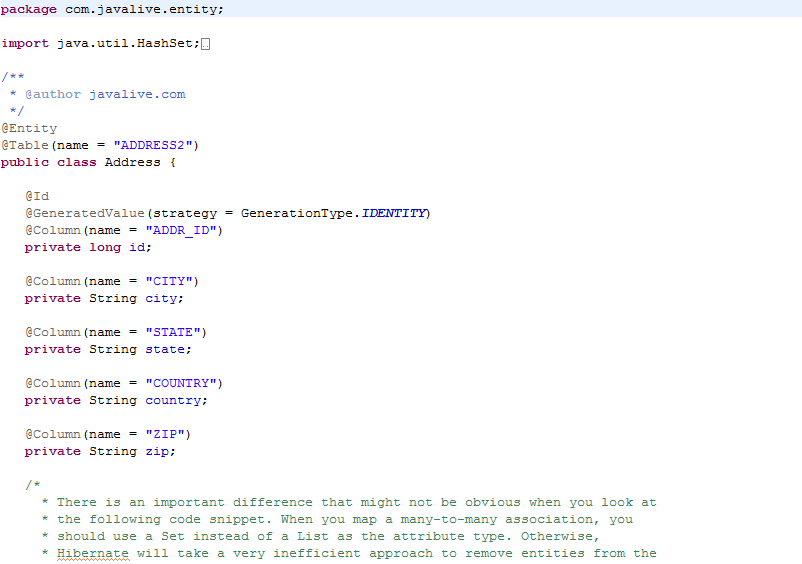
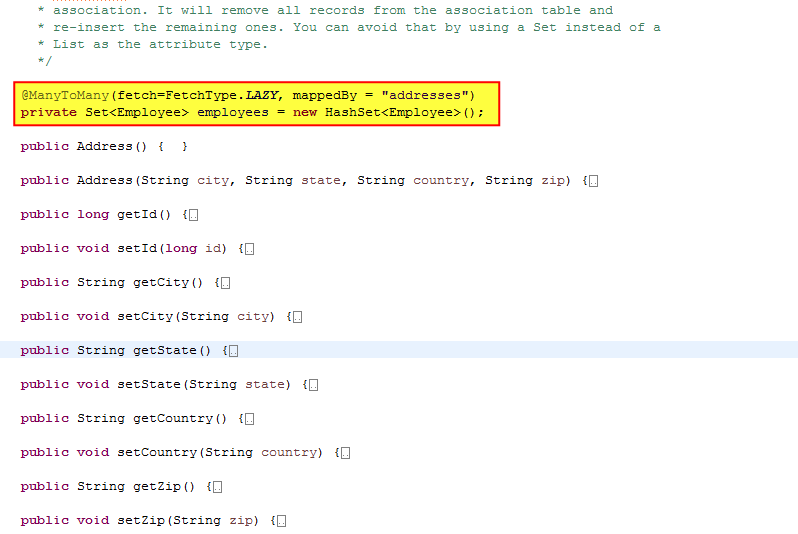
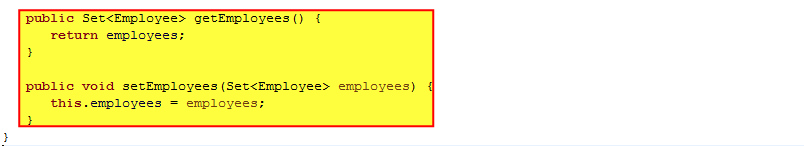
To define a many-to-many bidirectional association, the @ManyToMany annotation must be used in the target and source entities. One entity must be defined as the owner and the other must use the mappedBy attribute to define its mapping.

**Employee.java**

**** ****

The @JoinTable annotation is used to create the EMPLOYEE\_ADDRESS2 join table. This table defines an EMP\_ID foreign key to the source entity’s table primary key and an ADDR\_ID foreign key to the target entity’s table primary key.

**Address.java**

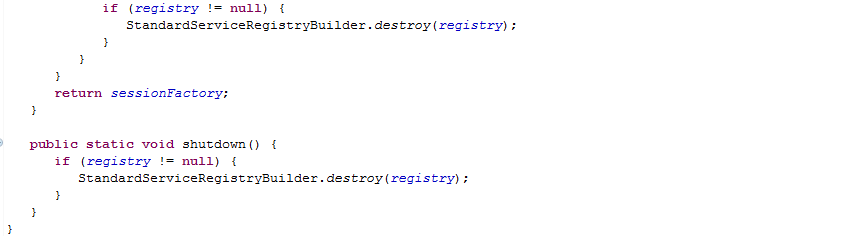
  

**Hibernate utility class**

Create a helper class HibernateUtil to bootstrap hibernate.

Map the Employee and Address entities using the #MetadataSources.addAnnotatedClass() method.

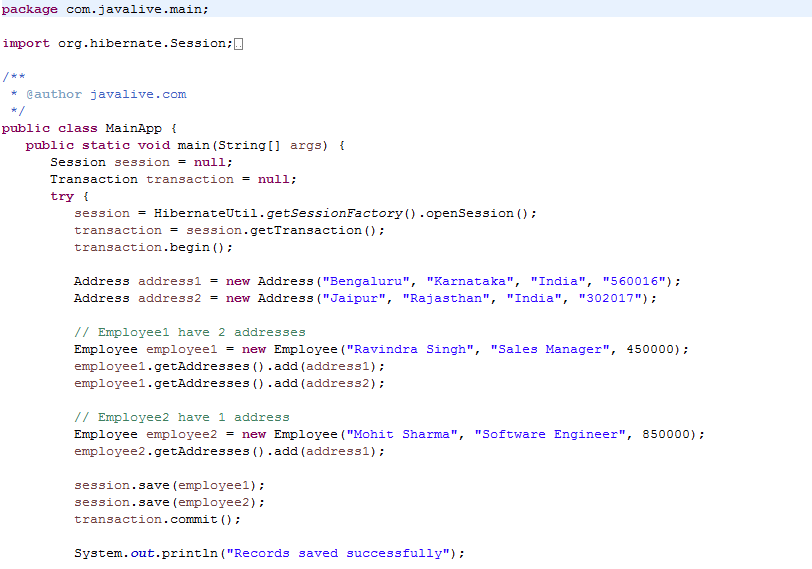
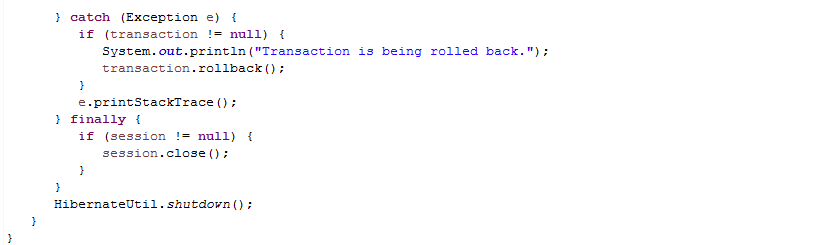
**HibernateUtil.java**

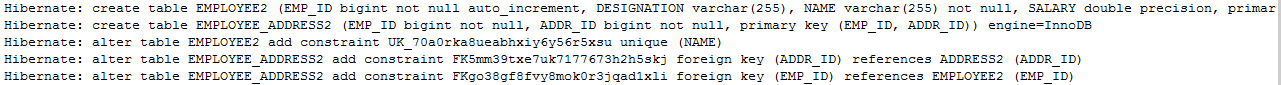
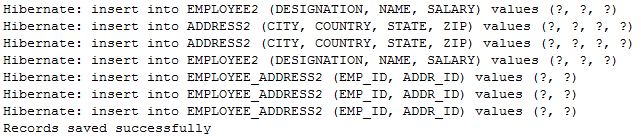
**Main class**

Create the MainApp class to run the application.

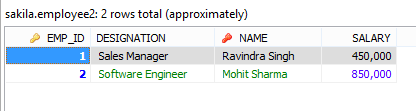
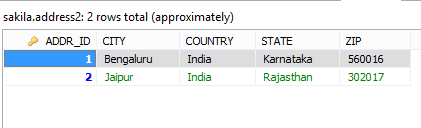
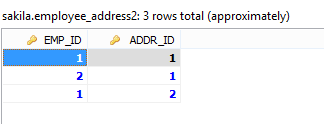
**MainApp.java**

**Output**

**  **

On executing the MainApp class, you will see the following records in EMPLOYEE2, ADDRESS2 and EMPLOYEE\_ADDRESS2 tables.

**  **