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

**A short introduction**

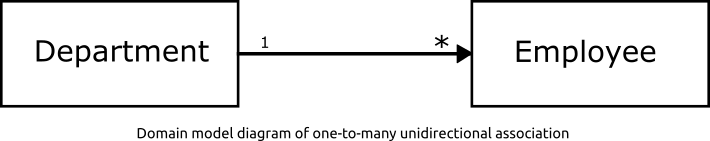
In one-to-many association, source entity has a field that stores one or more target entities. The @OneToMany JPA annotation is used to link the source entity with the target entities.

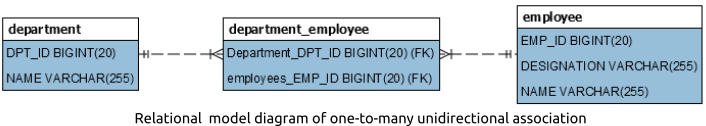
There are two types of one-to-many association -

**Unidirectional** → In this type of association, only source entity has a relationship field that refers to the target entity. We can navigate this type of association from one side.

**Unidirectional one-to-many association example**

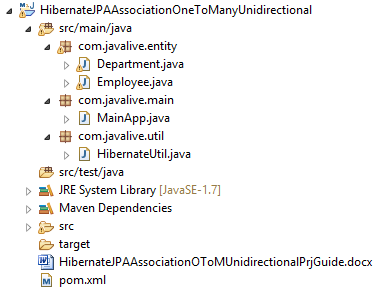
Consider the following domain model and relational model diagrams of one-to-many unidirectional association.





According to the above model diagrams a department can have many employees.

**Project Structure**

****

**Jar dependencies**

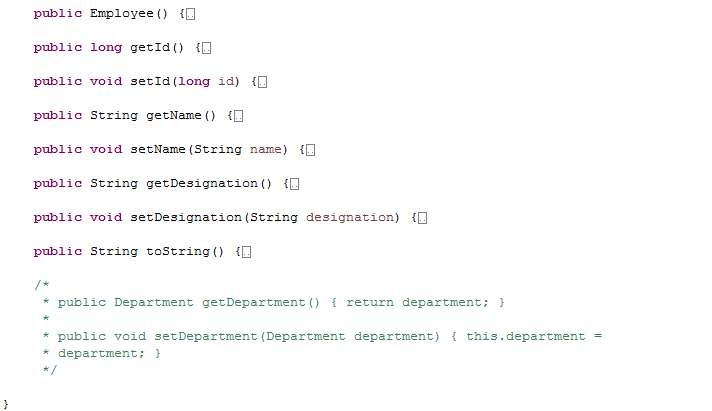
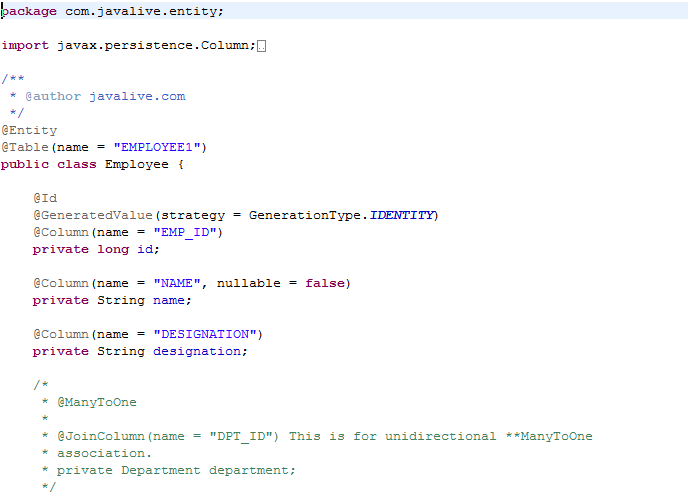
Add the following jar dependencies for Hibernate and MySQL driver in pom.xml file.



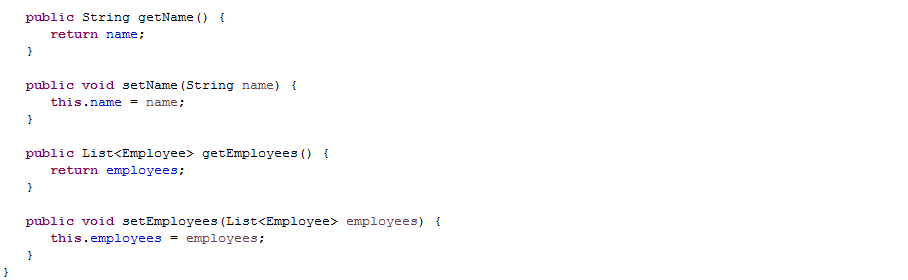
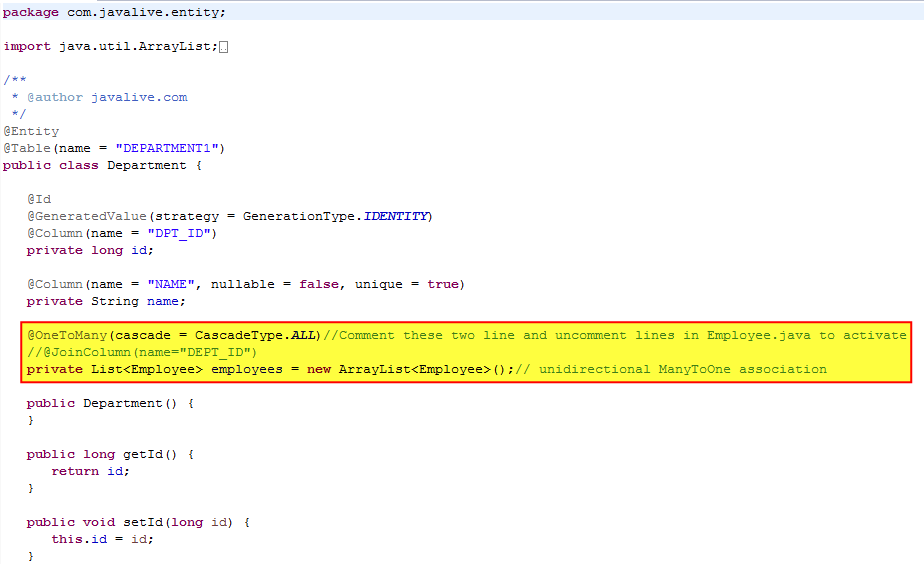
**Entity class**

Create two @Entity classes - Employee and Department, to map with EMPLOYEE1 and DEPARTMENT1 tables respectively.

**Employee.java**

****

**Department.java**

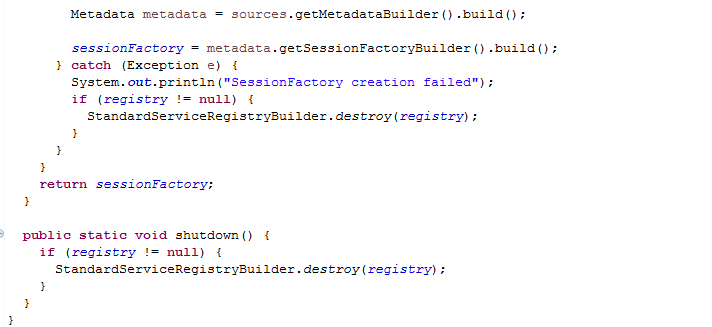
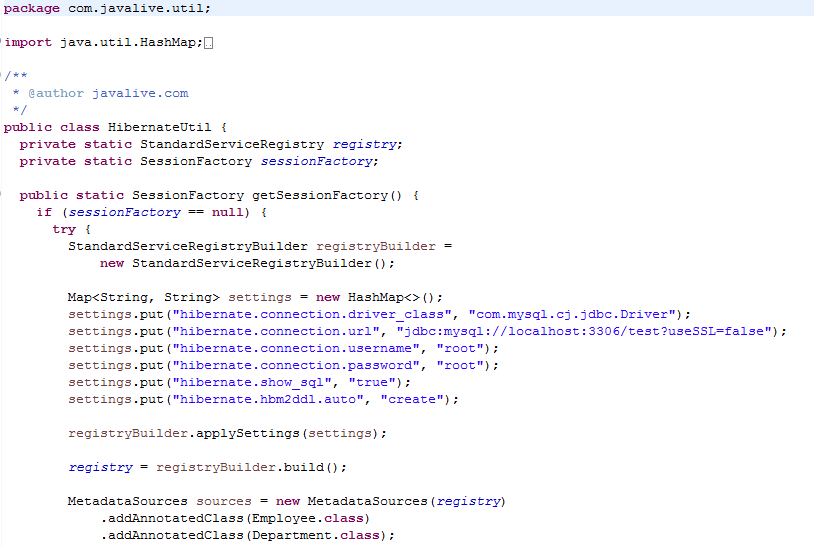


**Hibernate utility class**

Create a helper class HibernateUtil to bootstrap hibernate.

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

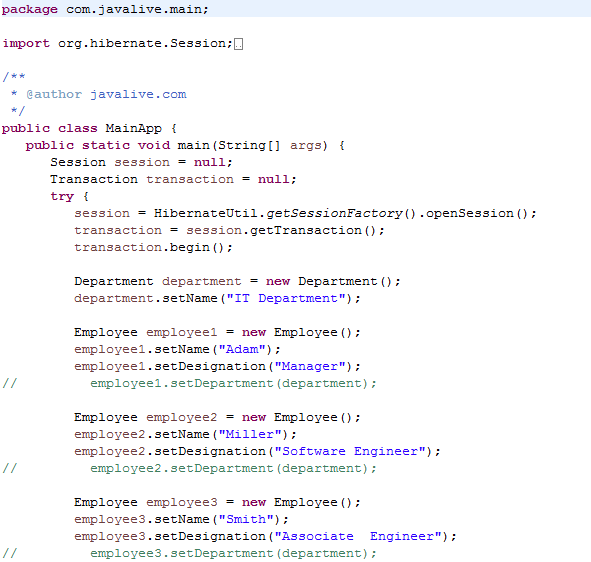
**HibernateUtil.java**

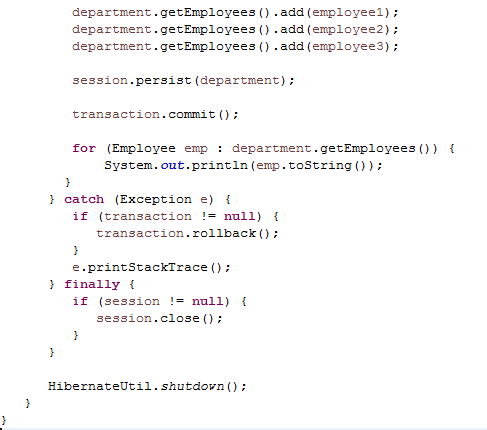


**Main class**

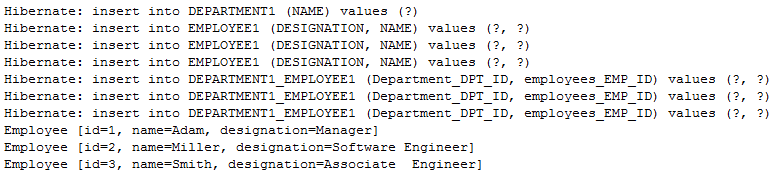
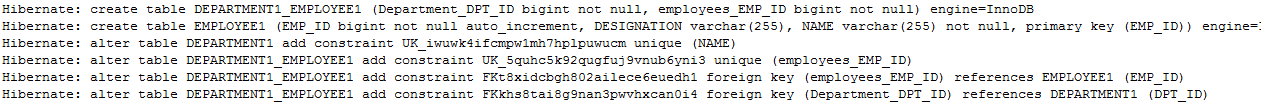
Create the MainApp class to run the application.

**MainApp.java**

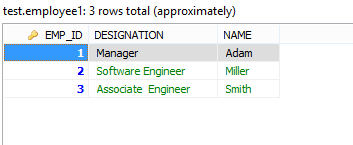


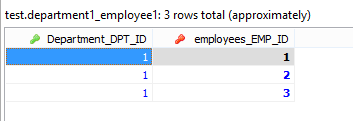
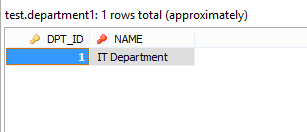


**Output**



On executing the MainApp class, you will see the following records in EMPLOYEE1,DEPARTMENT1 tables.



****