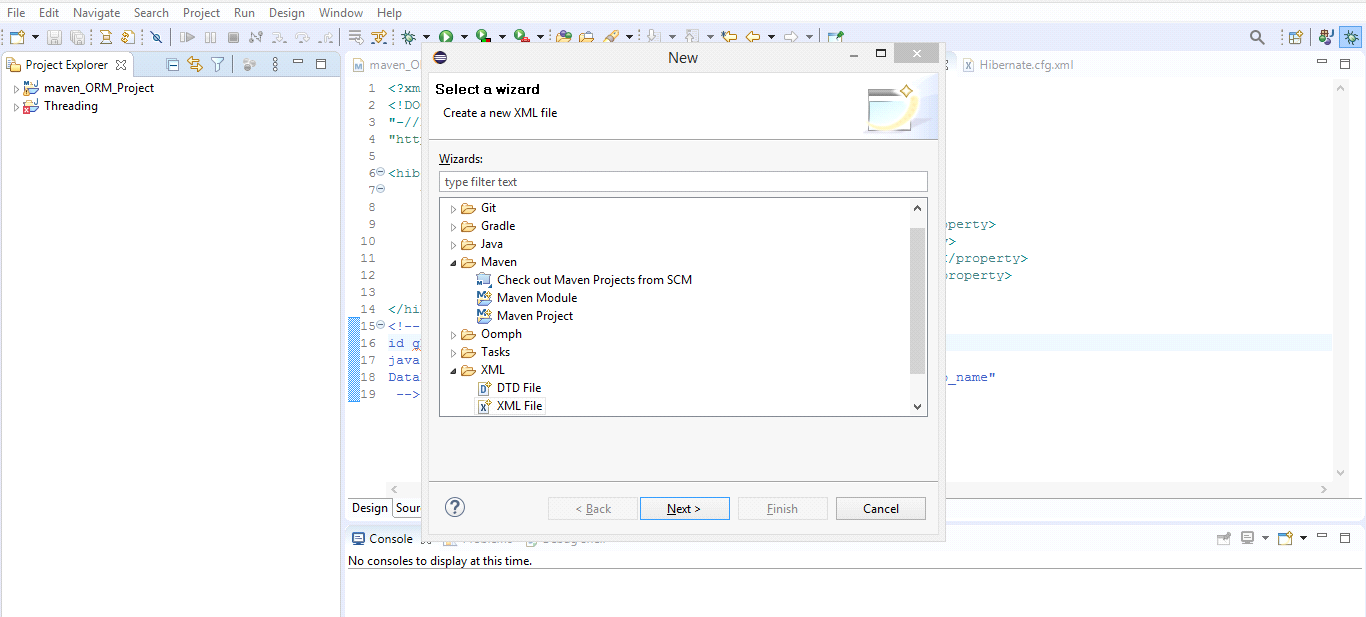
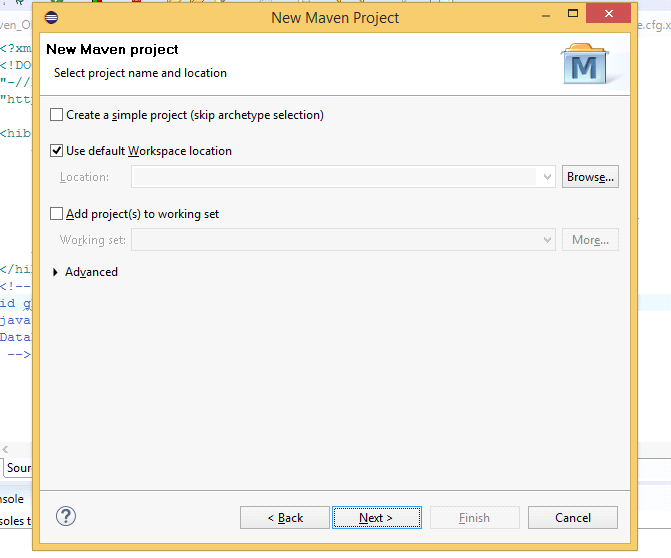
Hibernate

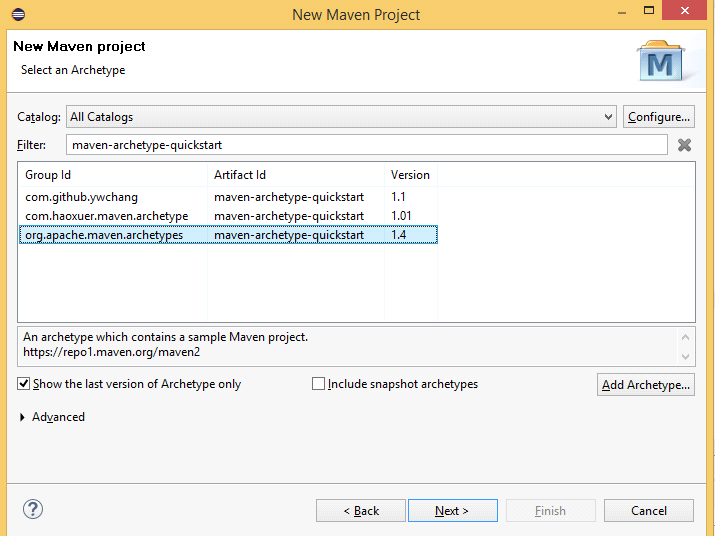
Step 1: Creation of Maven project

Open Eclipse click on Filenew-Others then you will see following window. You need to select Maven Maven Project then click on next.

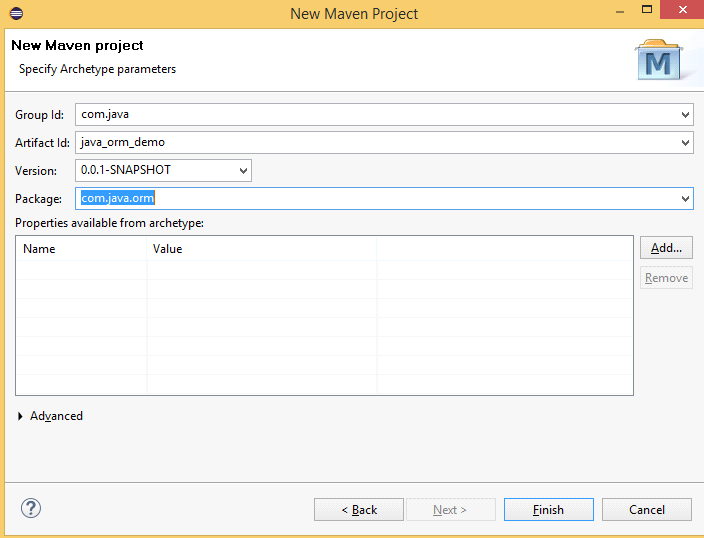


For the following window just check Use default workspace location.

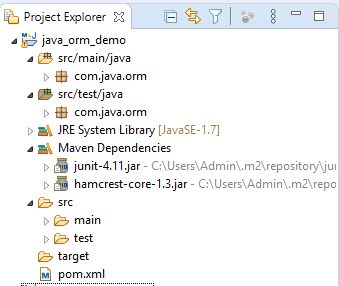


Now the following window will appear. In the Filters you need to search For maven-archetype-quickstart pluggin and make sure to select apache.maven.archetype then just click on next.

Now in the following window Artifact Id is nothing but project name. Here I’m giving project name as ”java\_orm\_demo”. And I’m giving package name as “com.java.orm” after this just click on finish. Eclipse will take some time to create “java\_orm\_demo “ project.



 Look at the default folder structure of the Maven project.



Here wo go!! We have successfully created maven project.

Step 2 : Hibernate setup

Now in this step we are going to setup hibernate since our project depends upon hibernate.

Search For maven dependencies or hibernate on chrome and navigate to first sight i.e. org.hibernate. Now select 5.4.16 Final version because this version is stable and if there are any problems with this version we can find it on google. Now copy following details rom the site. Now open pom.xml file and add the following dependencies as shown in following screenshot.

<!-- <https://mvnrepository.com/artifact/org.hibernate/hibernate-core> -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.4.16.Final</version>

</dependency>



Now search for maven dependency for MySQL and select mysql connector java maven-repository. And select version 8.0.16 and copy following containts. Now paste the following dependency in pom.xml file as shown in following image. After that just press ctrl+sto save it. After saving it you can find these dependencies in Maven Dependencies folder in folder staructure of project

<!-- <https://mvnrepository.com/artifact/mysql/mysql-connector-java> -->

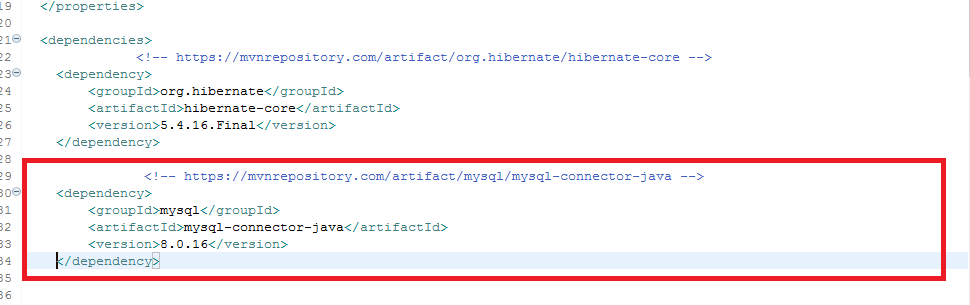
<dependency>

<groupId>mysql</groupId>

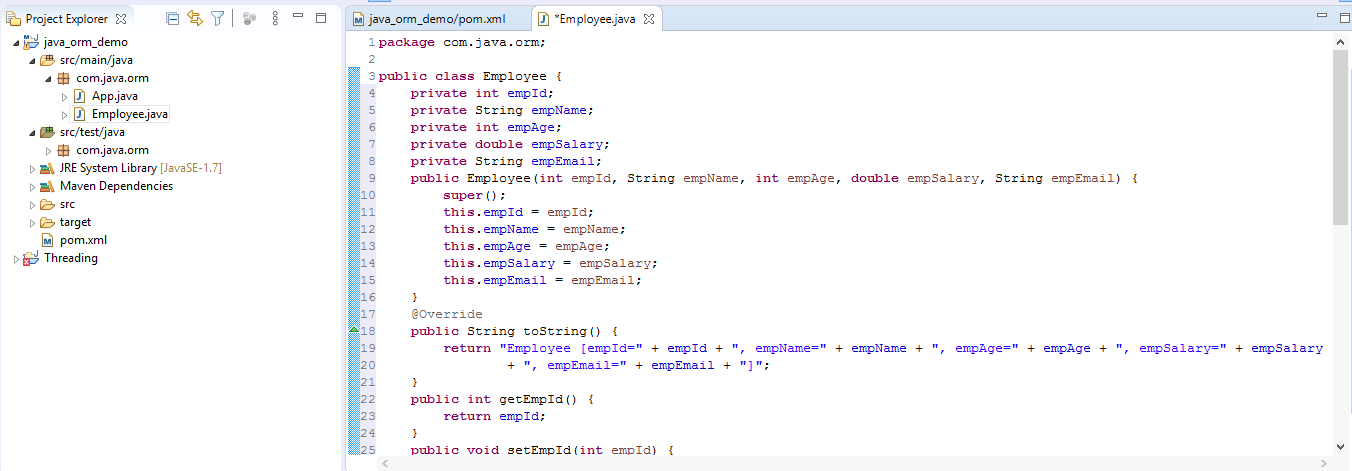
<artifactId>mysql-connector-java</artifactId>

<version>8.0.16</version>

</dependency>



Now in n src/main/java folder we have a package named com.java.orm . create a new class named Employee in this package. And add following properties in the class and create constructer, toString method and getters and setters for the same properties.



Now right click on src/main/java . Now create a new xml file by clicking on newother xml file. And give name as “employee.Hbm.xml” . Hbm stands for hibernate mapping. By creating this file we are telling hibernate that in which table we want to put Employee class in database. Create another xml file in the same src/main/java named “hibernate.cfg.xml” . cfg means this is configuration file. In this file we are going to add the config which is necessary to connect to database. And no need to write this configuration part instead just copy it form internet. Search for hibernate configuration file. Go to tutorials point site and copy following contents and paste it inside cfg file. In the session factory block we need to add the properties which we have created in Employee class as shown in below image.

<?xml version = *"1.0"* encoding = *"utf-8"*?>

<!DOCTYPE hibernate-configuration SYSTEM

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- The following are the mandatory properties for hibernate. -->

<property name=*"hibernate.connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.connection.password"*>root</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/ormdb</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL8Dialect</property>

<property name=*"hibernate.hbm2ddl.auto"*>update</property>

<property name=*"hibernate.show\_sql"*>true</property>

<!-- <mapping resource="employee.hbm.xml"/> -->

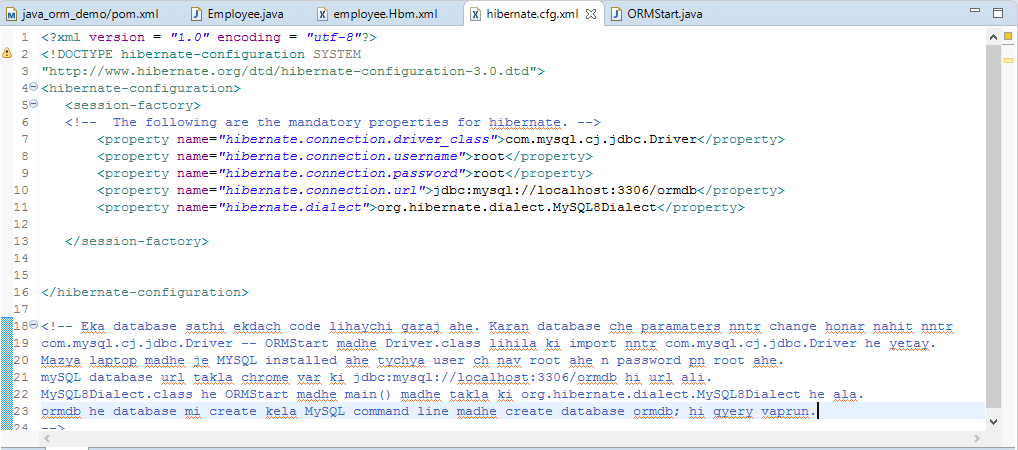
<!-- <mapping resource="product.hbm.xml"/> -->

<mapping class=*"com.java.orm.basic.Employee"*/>

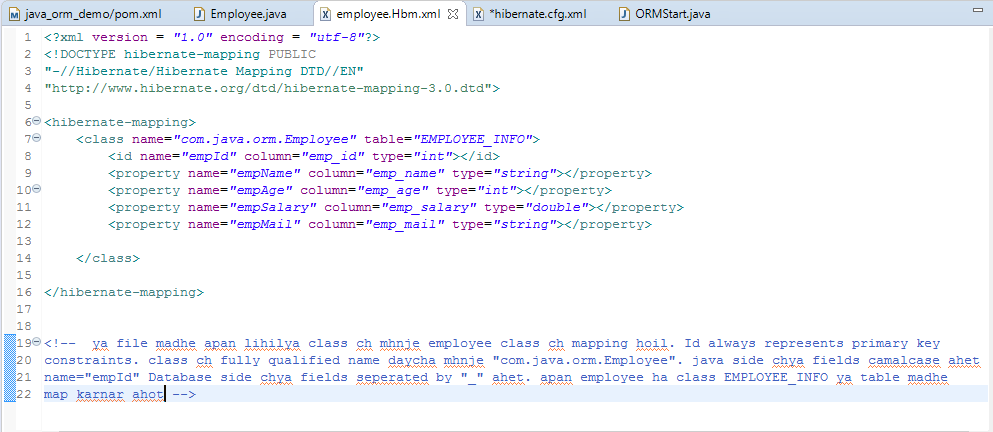
</session-factory>

</hibernate-configuration>

Now side by side create a new class named ORMStart in com.java.orm package. Now open MySQL commandline interface. And fire following commands.Create database ormdb; Use ormdb; Show tables;

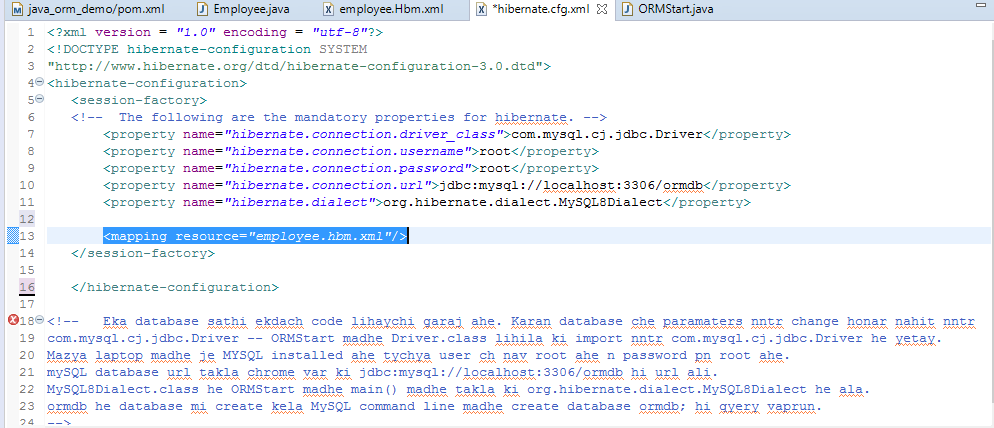


Now again search for hibernate mapping file and copy following containts and add it in employee.Hbm.xml file . Add the contents as per the following screenshot.

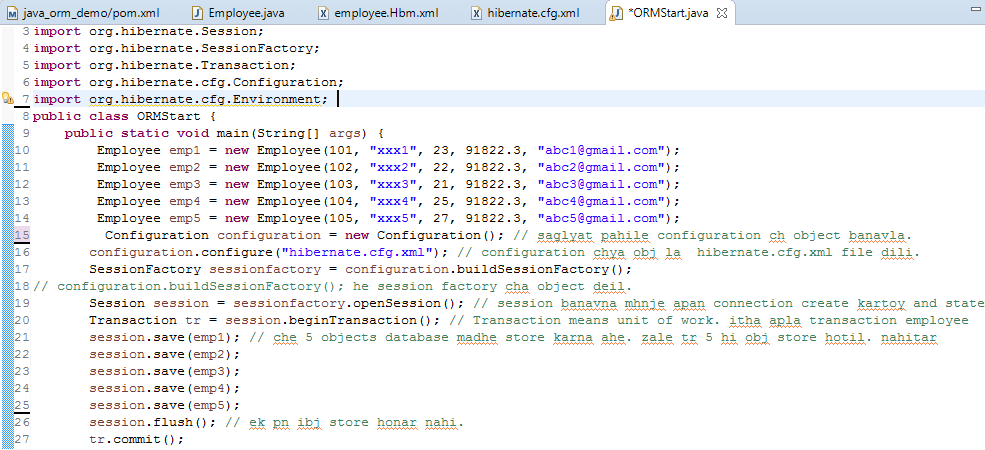


And give this above configuration to our session factory i.e. add a single line which is mentioned below in hibernate.cfg.xml file after properties. Refer the following screenshot for the same.

<mapping resource=*"employee.hbm.xml"*/>



Now go to ORMStart class and add following contents to it.

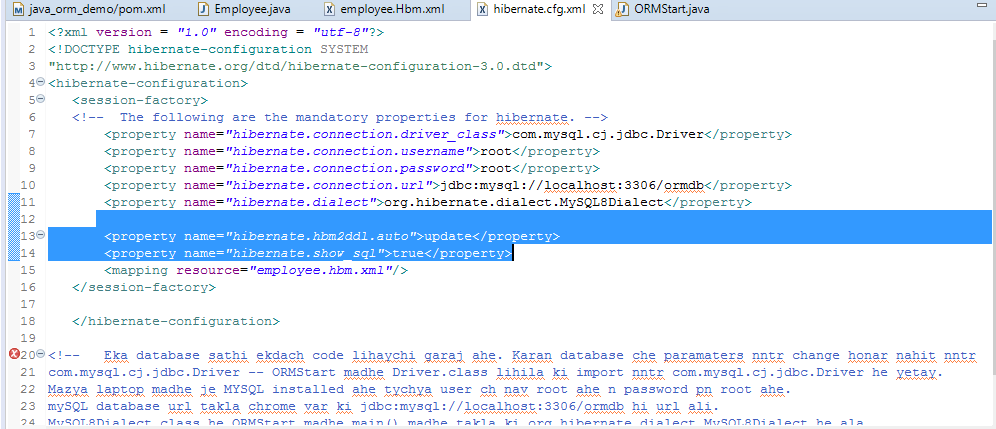


Now If you want that hibernate should create tables on it’s own then you need to add following property to hibernate.cfg.xml file.

<property name=*"hibernate.hbm2ddl.auto"*>update</property>

And also add the following property in the same file so that we can see what queries hibernate is firing.

<property name=*"hibernate.show\_sql"*>true</property>

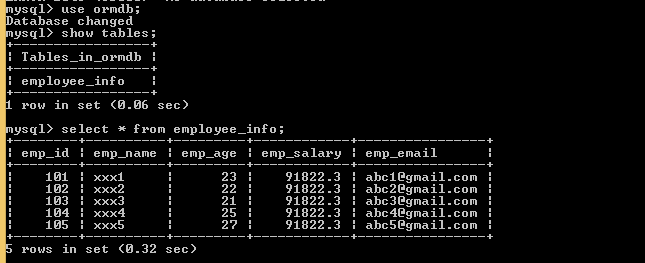


Now finally go to MySQL command line and fire following commands you can see 5 employees are successfully added into database. Refer following image.

use ormdb;

show tables;

select \* from employee\_info



Now go to ORMStart and run the program. You will get o/p like following image. Thus hibernate is successfully inserted emp1.