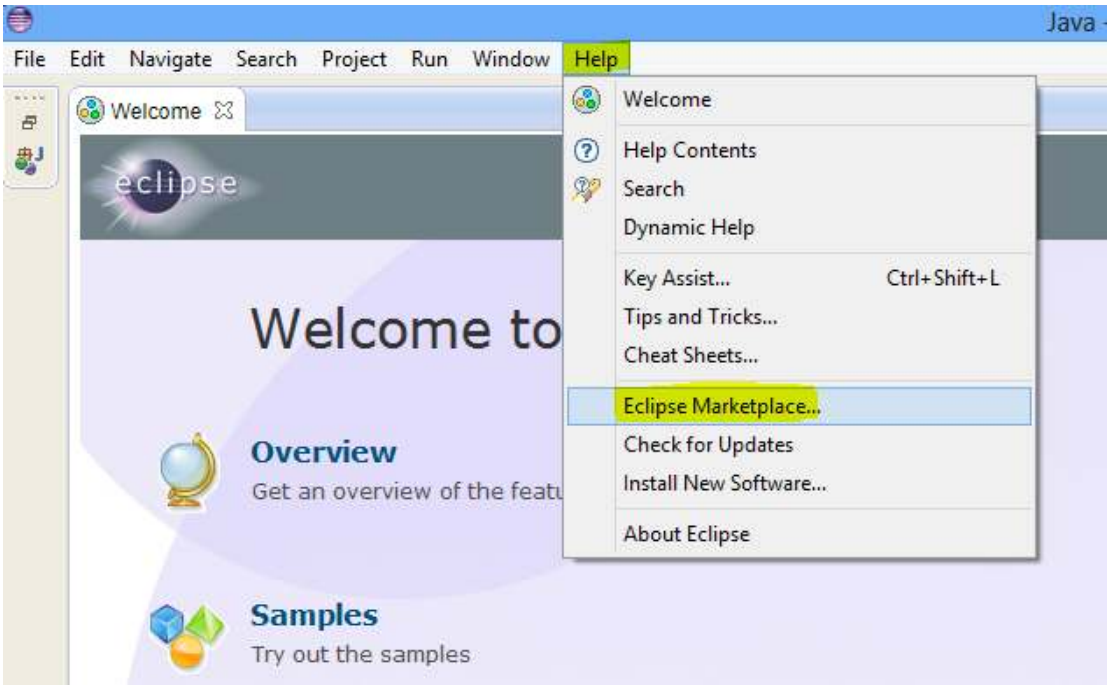
Hibernate Automatically Create Java POJO Classes From Database Tables In Eclipse

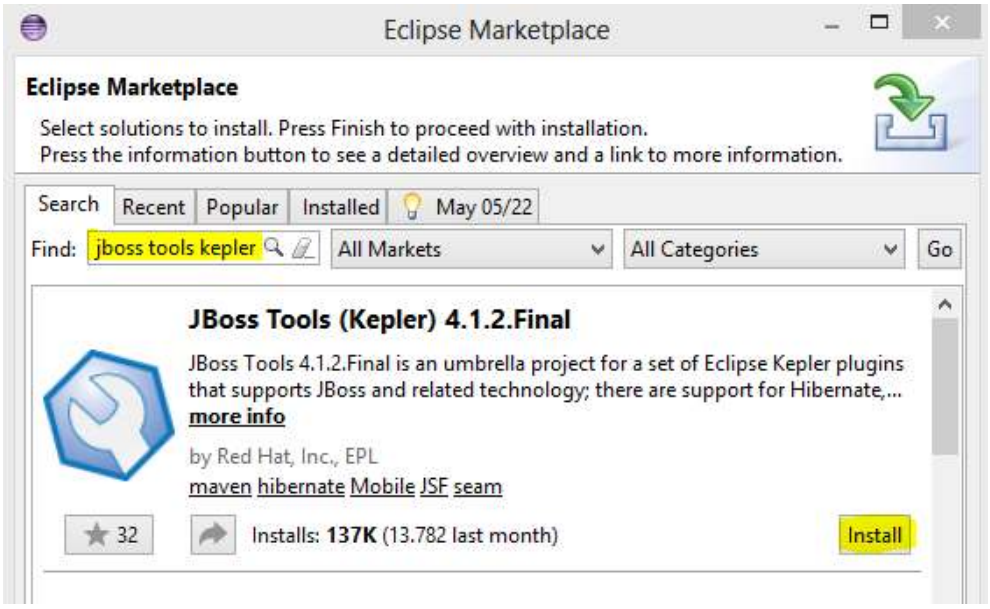
With **Hibernate JBoss Toolbar,** we will do **reverse engineering** to create **Plain Old Java Classes(POJO), Dao** classes and **hbm.xml** files from **database** tables. Thus, we can get rid of converting manually database tables into Java classes. This process is called as Reverse Engineering. Now lets start the process by **using** Eclipse Kepler and Hibernate Toolbar Plugin.

1. First of all, **download** Eclipse Kepler 4.3 then run it

2. After running Eclipse program, click **Help** then choose **Eclipse MarketPlace**

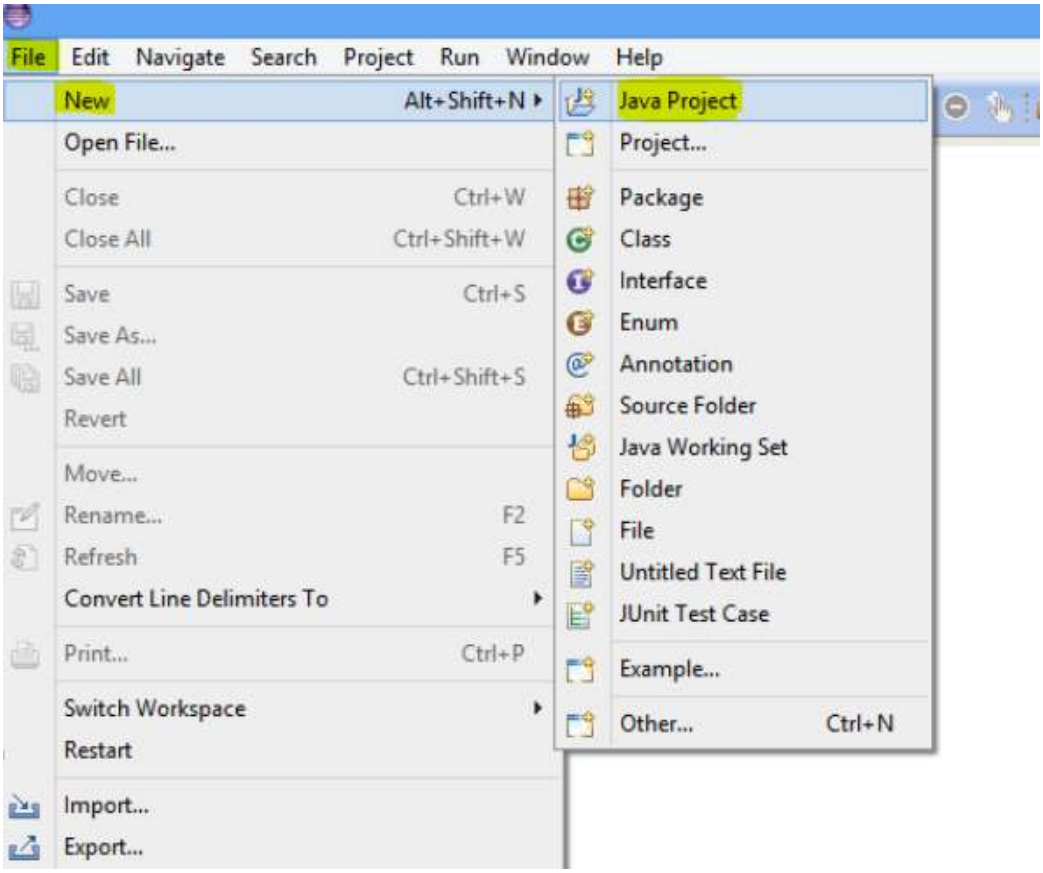


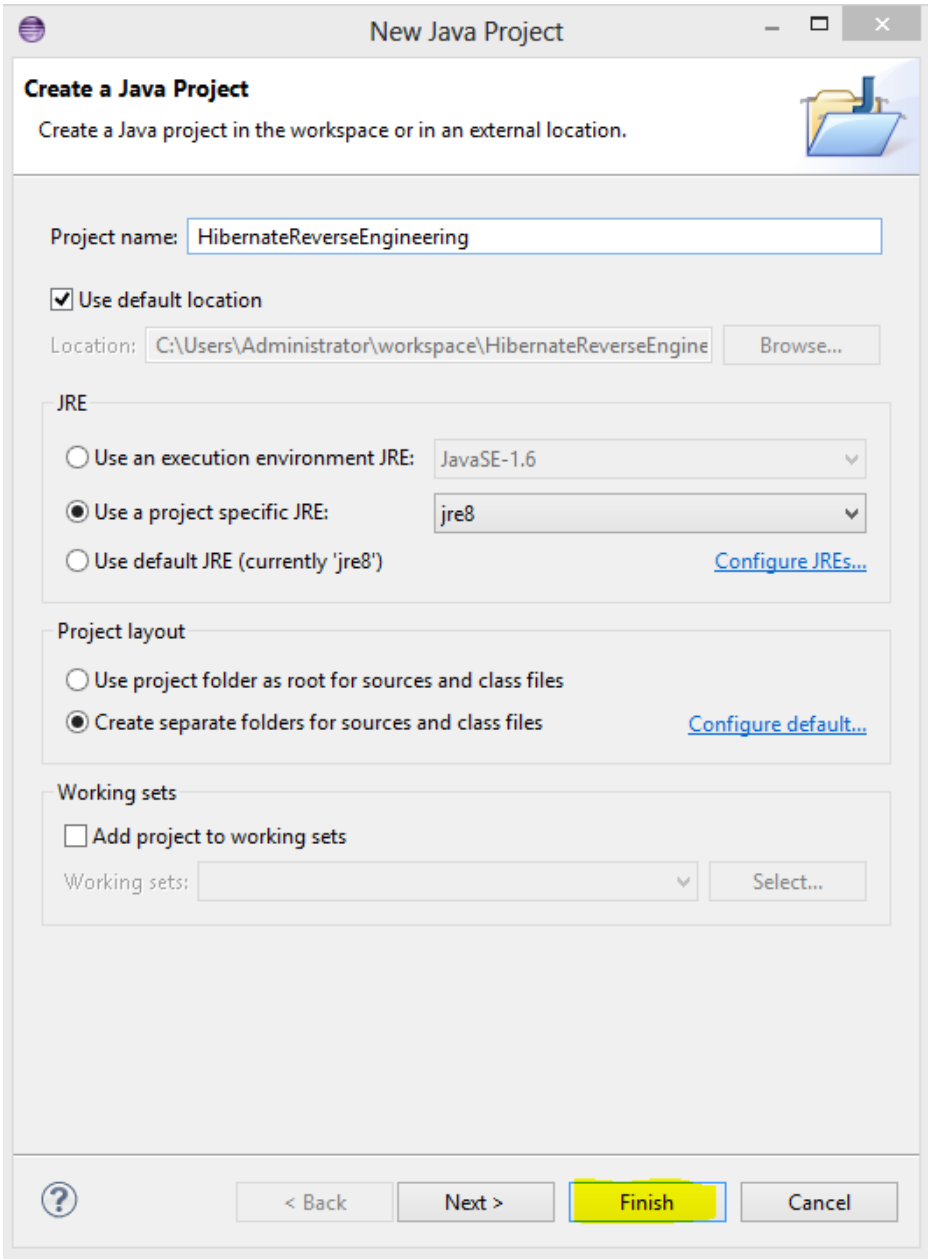
3. Then write **jboss tools kepler** into **Find** area. You can see the results containing **JBoss Tools(Kepler)** plugin. Please install this plugin



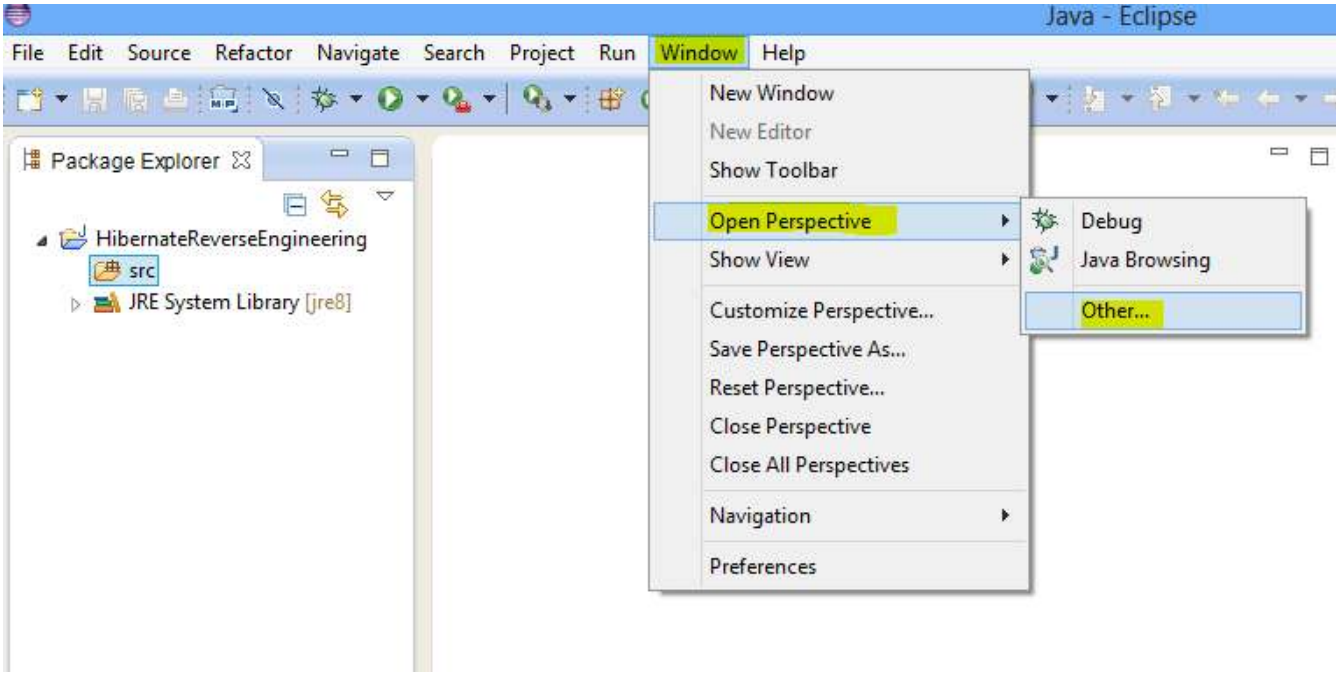
After these processes are completed, **do the following steps:**

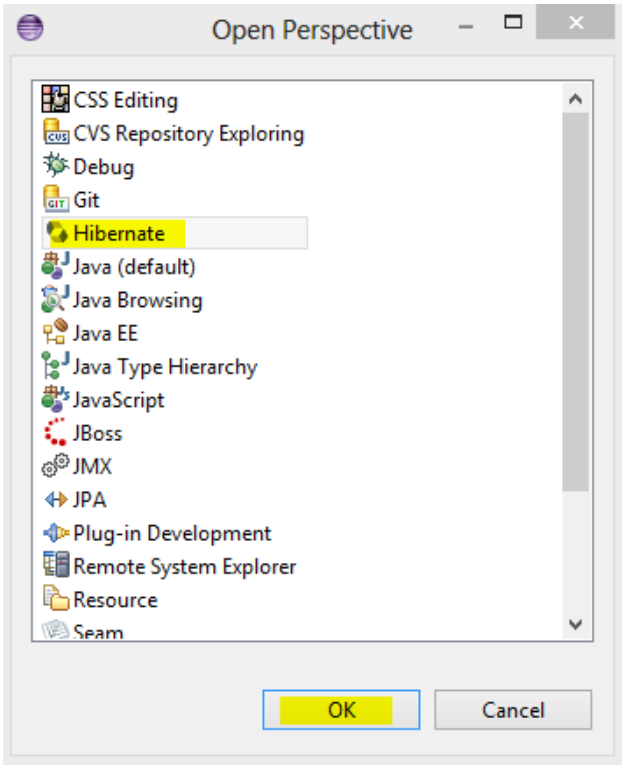
1. Create a new Java Project



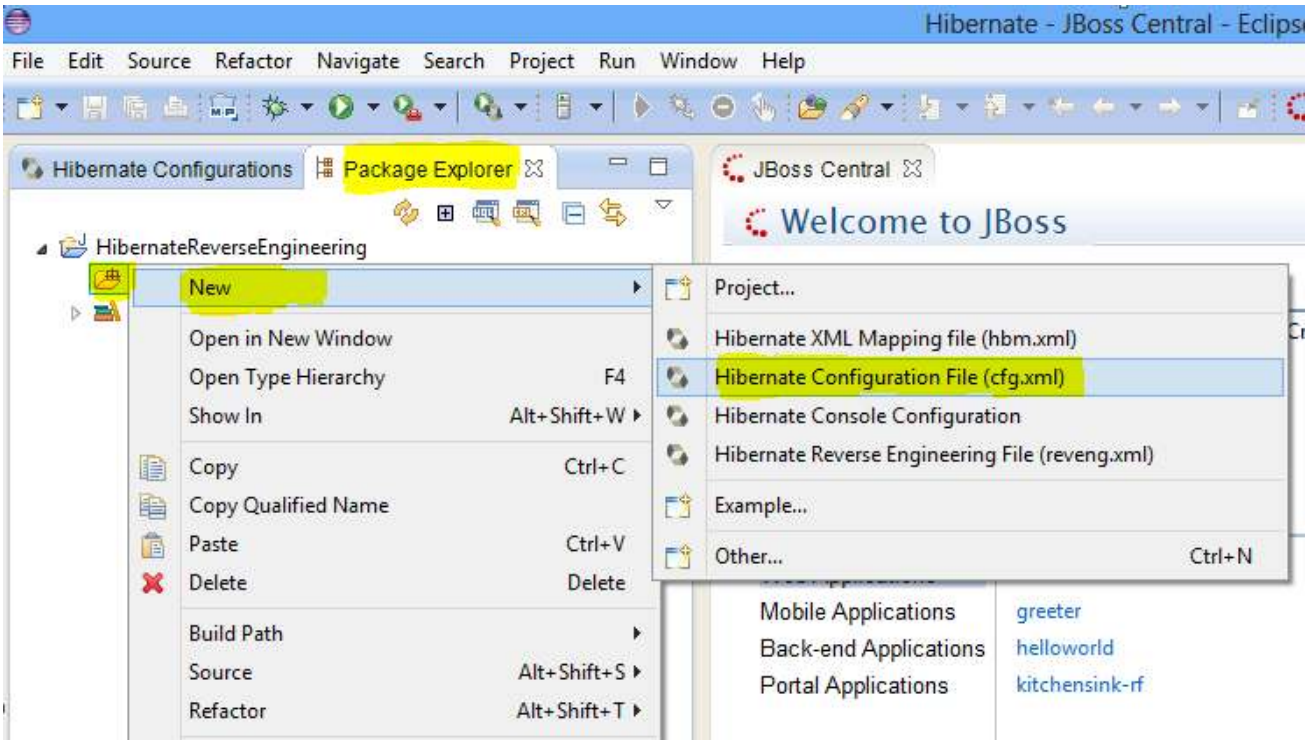


2. Change windows perspective into Hibernate by clicking **Window** then **Open Perspective** buttons.

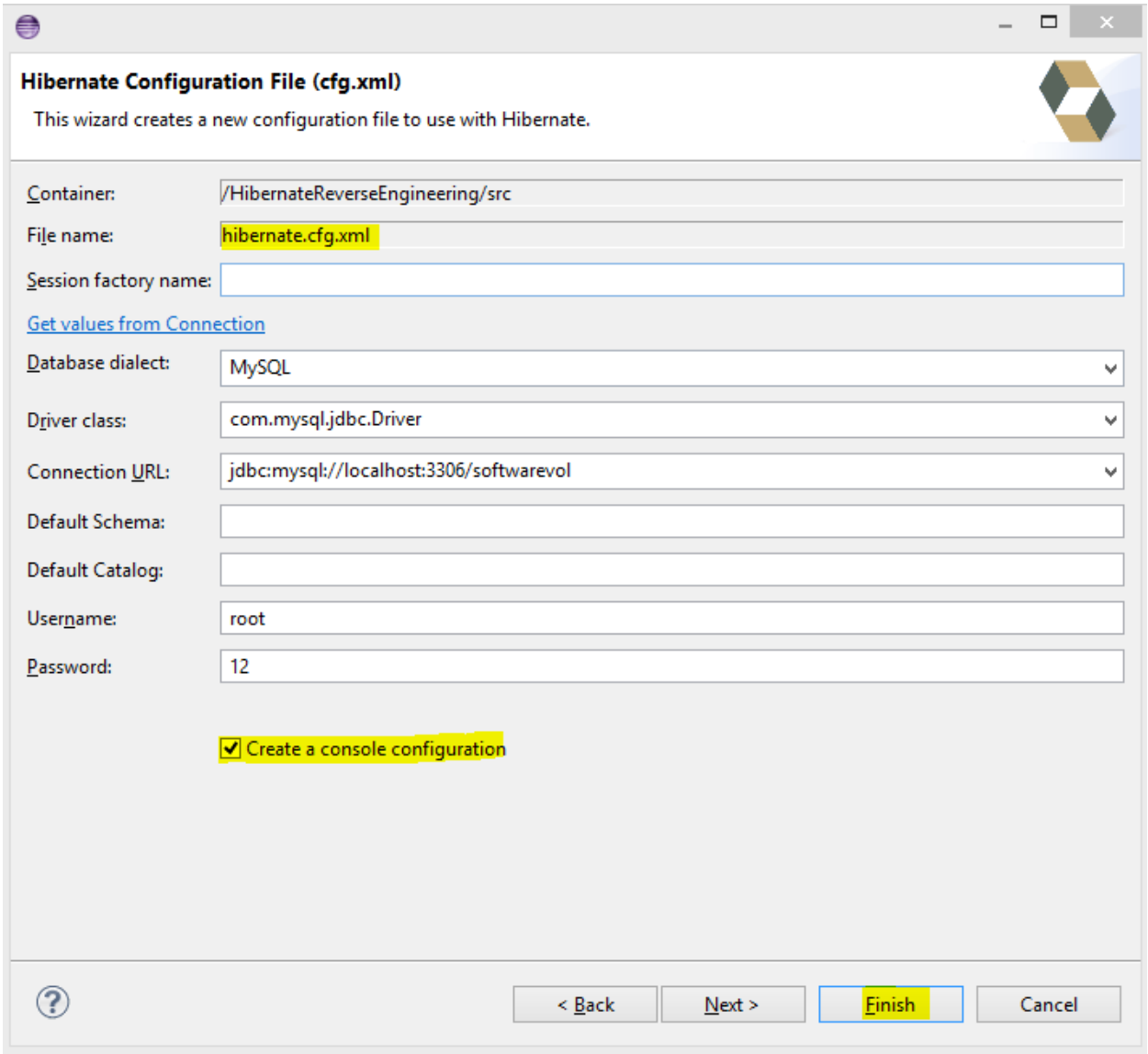




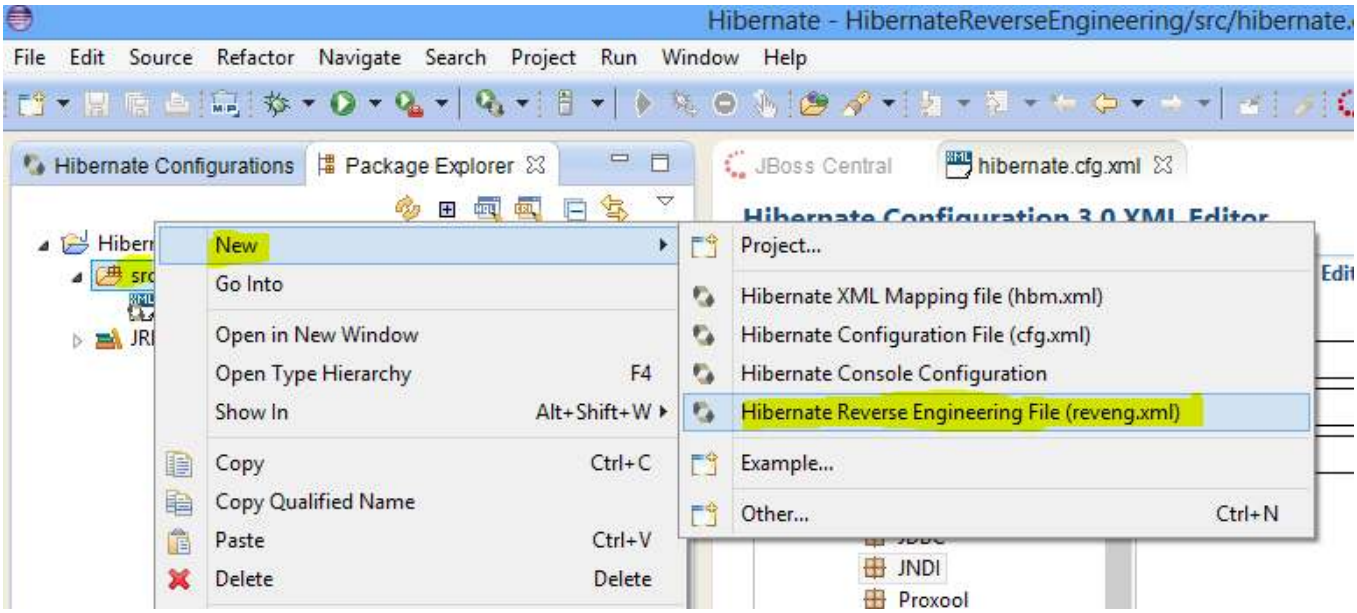
3. In the **Package Explorer** tab, right click onto **src** folder, then choose New -> Hibernate Configuration File **(.cfg.xml)**



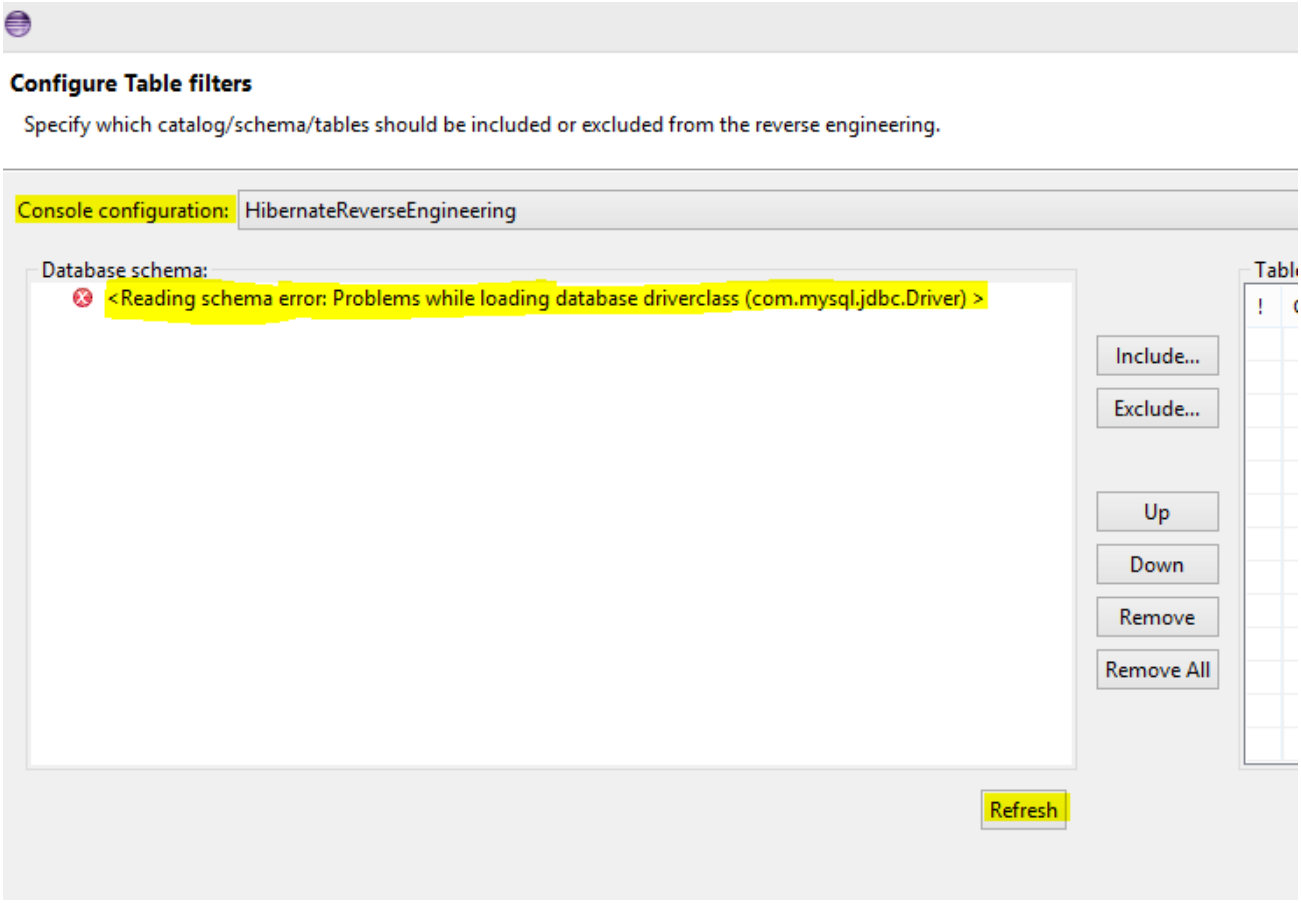
4. Click Next button in Opened Window and then new Window is opened. At this window, we configure database settings to connect to our db and choose **Create Console Configuration** option.



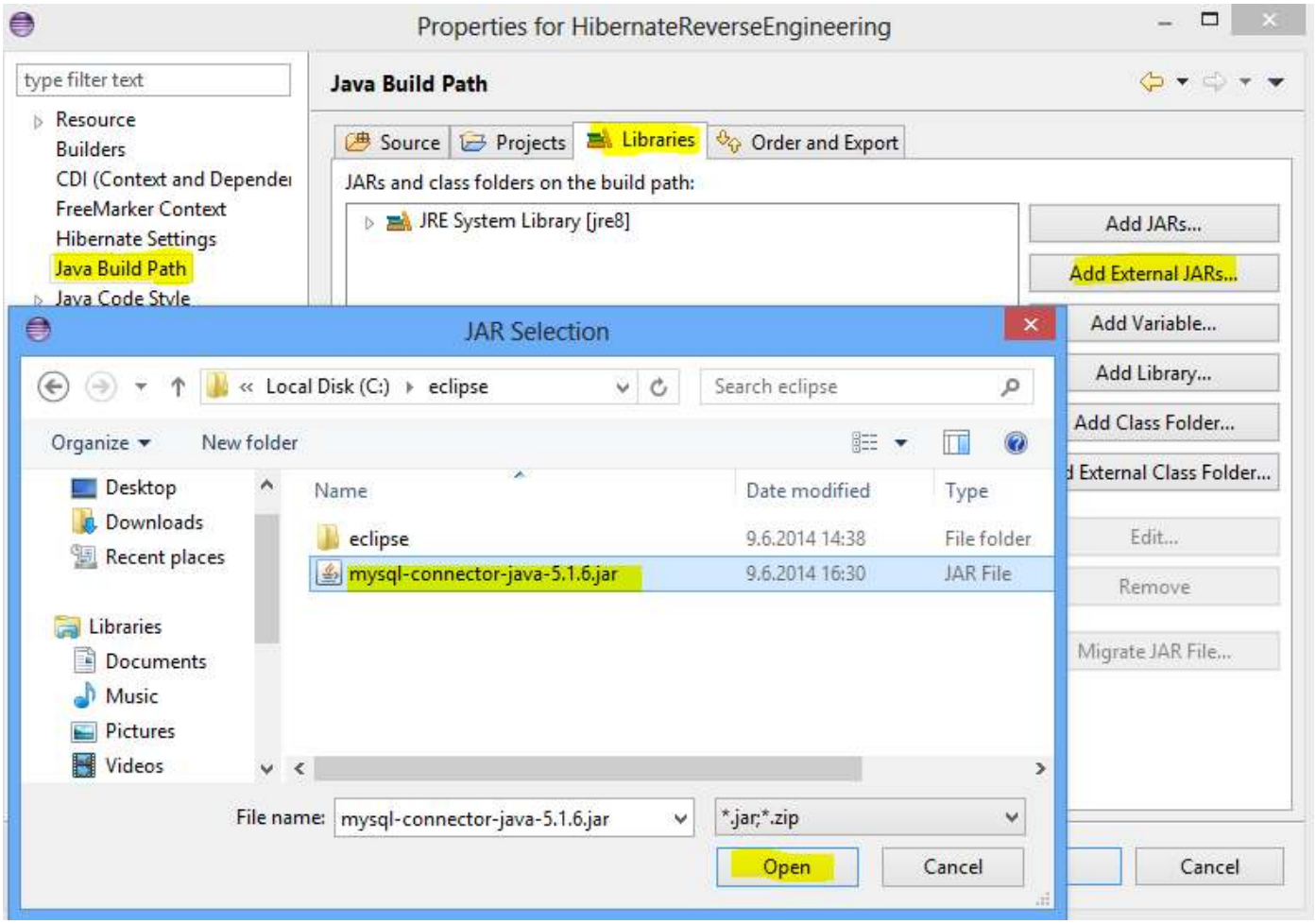
5. Again right click onto **src** folder then New -> Hibernate Reverse Engineering File(reveng.xml) is choosen



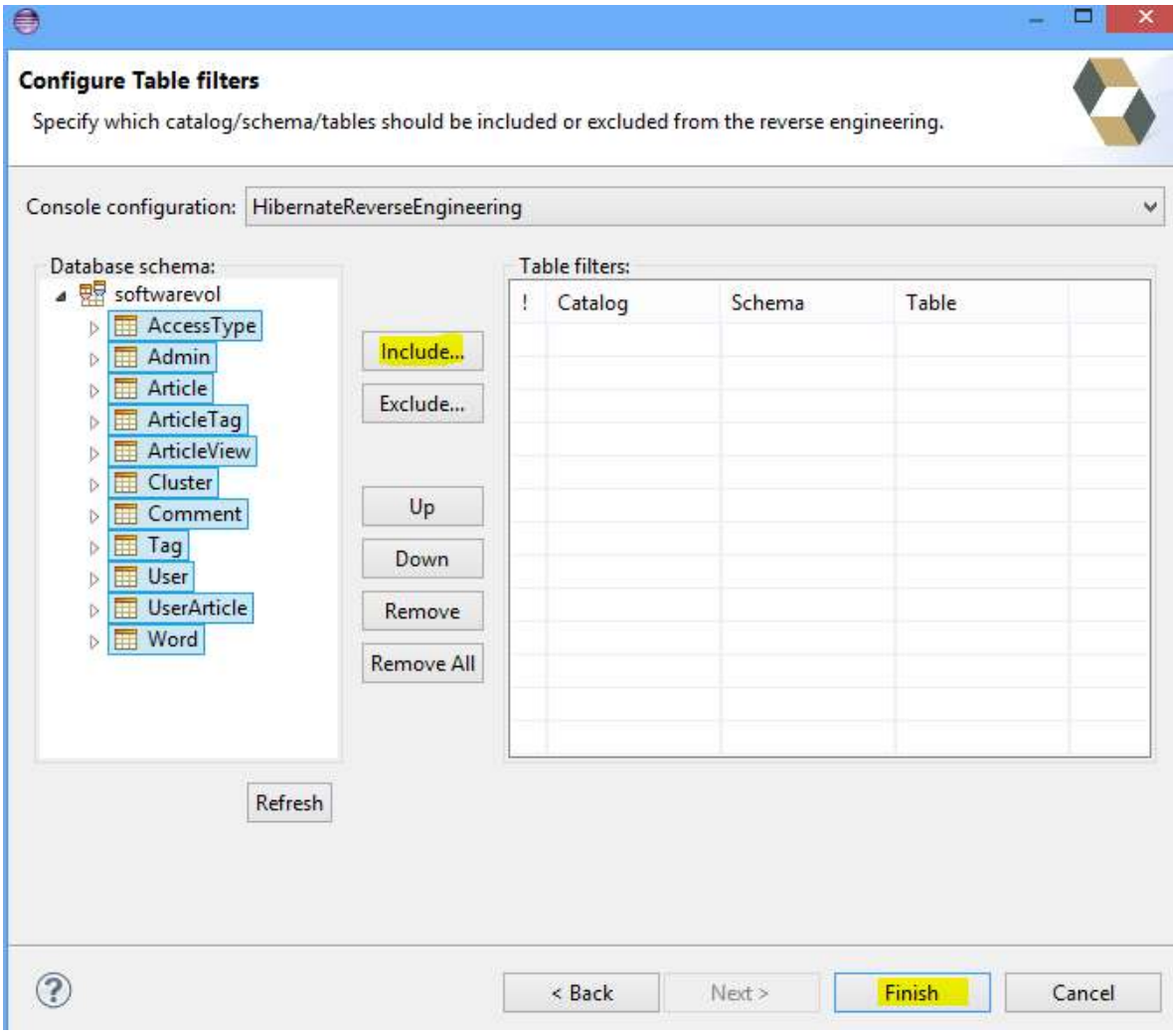
6. Click Next button in Opened Window and later new Window is opened. Choose the configuration created at step 4 in the **Console Configuration** section. Then click Refresh button, however you will see an exception such as **loading database driver class** error. To solve this problem, you have to add mysql connection provider jar file as External Library.



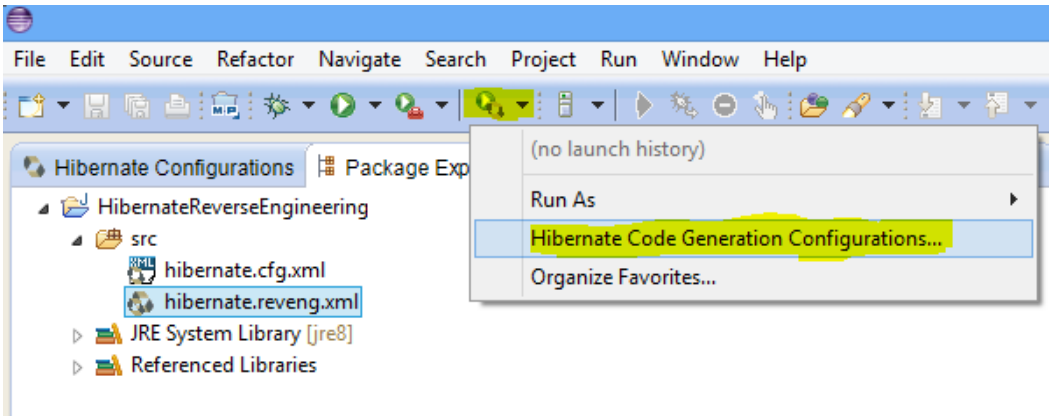
7. To download necessary jar file please **click** (http://www.softwarevol.com/sources/docs/setup/hibernate/reverseEngineering/mysql-connector-java-5.1.6.jar) . After downloading add this jar file into External Jars section by clicking File->Properties



8. Now back to **step 6.** Click **Refresh** button, you can see now tables in the database. After choosing the tables which will be converted into Java POJO classes, click **Include** button and lastl click **Finish** button



9. After clicking **Finish** button click **Arrow** icon as seen in the following image. Now open **Hibernate Code Generation Configuration** area.



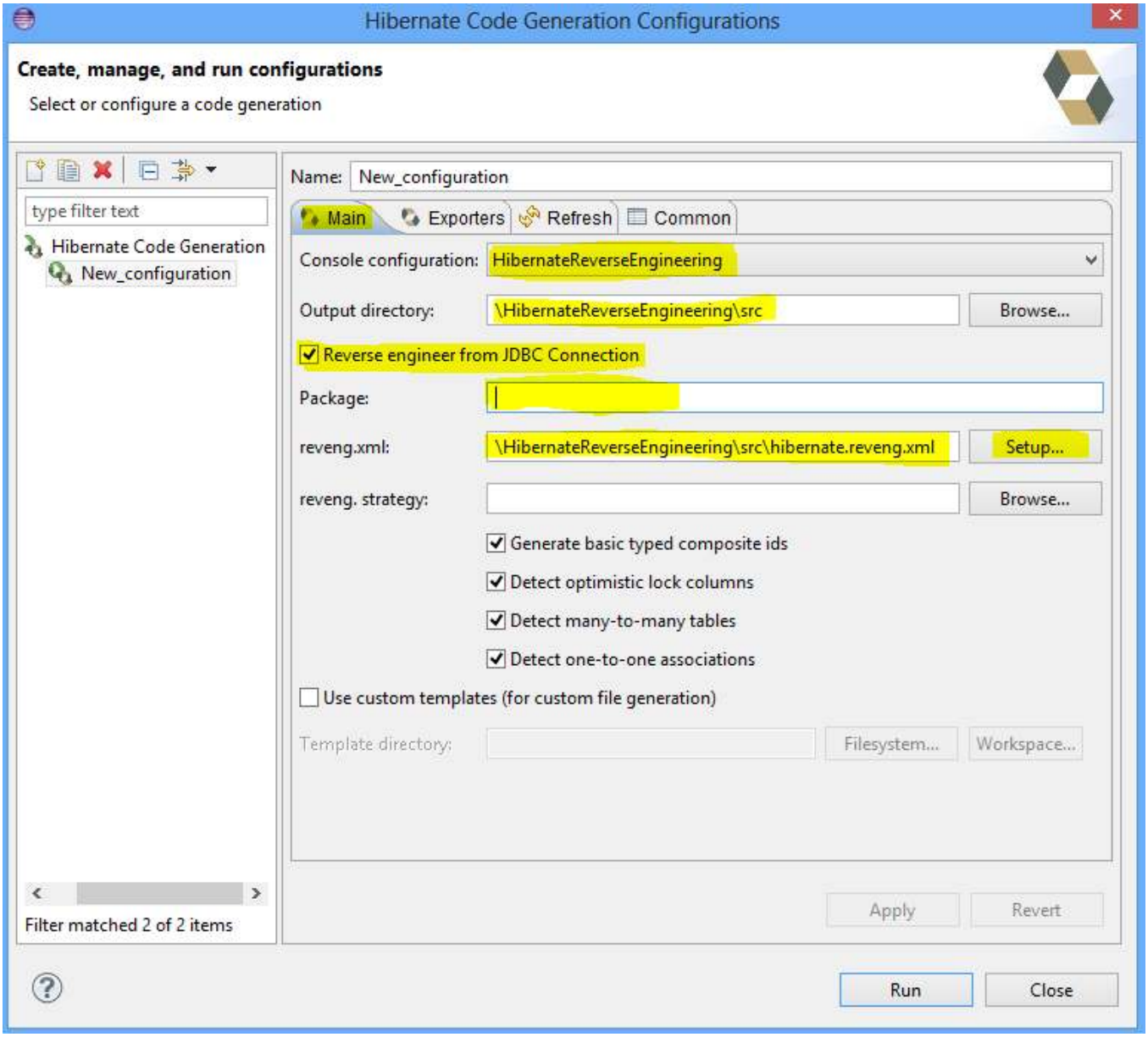
10. **Hibernate Code Generation Configuration** screen is seen in the following image. At the **Main** tab choose the configuration created in the **step 4.** And choose **src** folder path in the **Output** directory. Note that you should also choose **Reverse engineer from JDBC Connection** option, otherwise Java classes will not be created. You can write packet name in the src folder into **Package** section. We will leave this section as empty. When **package** name is specified, created Java classes' **package** is set to this package. Then click **Setup** button and choose **reveng.xml** file path. **reveng.xml** file is created after **step 5.**

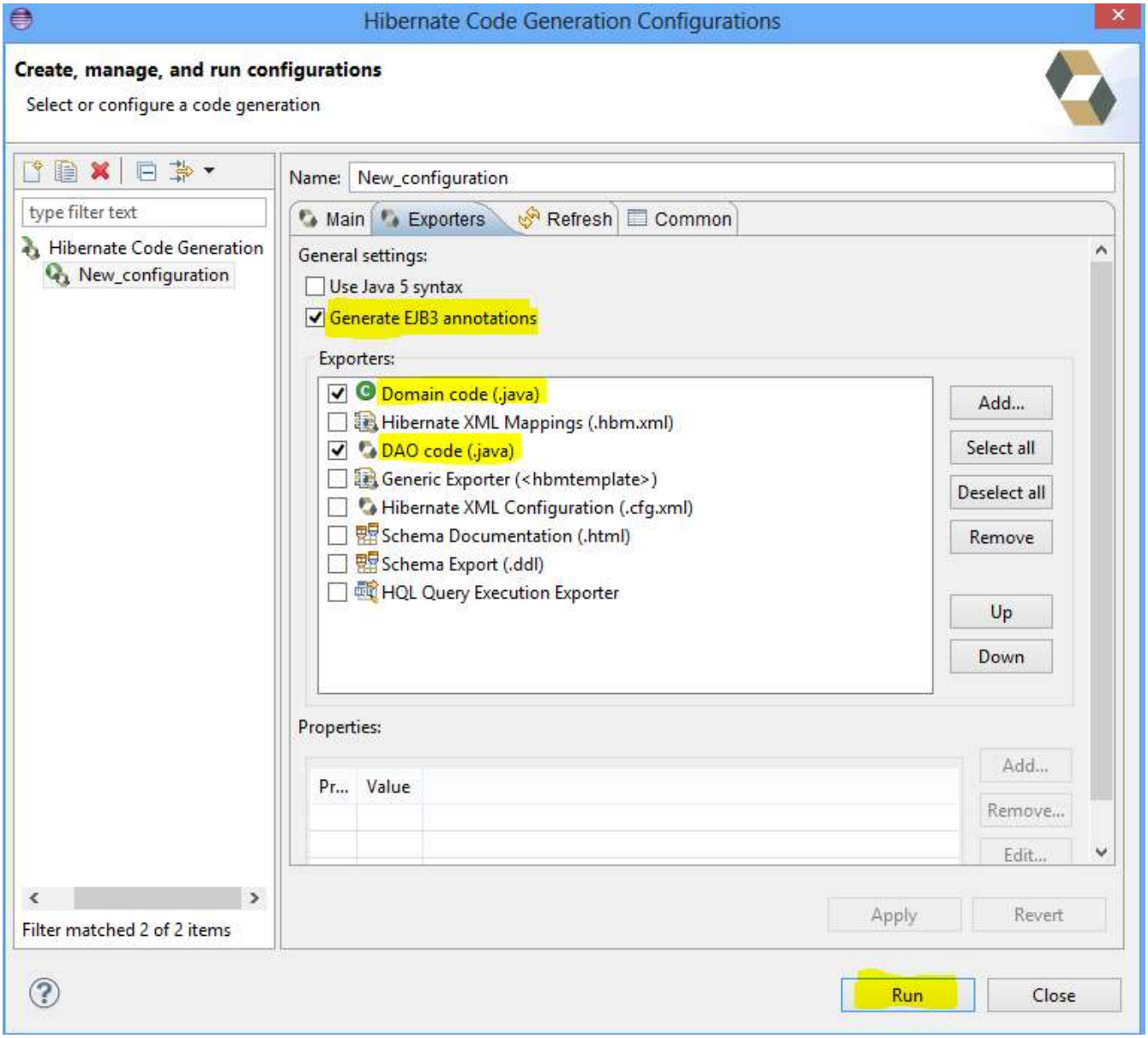
After Main tab, Go to **Exporters** tab. This section is used to specify automatically created Java classes properties by choosing **EJB(** annotation **type** classes) or **Java 5** syntax type. If you choose Java 5 syntax, you have to add **.hbm.xml** files for each **class** to configure hibernate database mapping. Some important options in the Exporters tab is:

Domain code(.java): represents tables in the database as Java classes

Hibernate XML Mapping(.hbm.xml): When annotation is not used, it is used to define relationship between Java classes and tables in the database.

DAO code(.java): This file contains codes used to operate database operations such as insert, update, delete, etc.





Lastly, click **Run** button to start the reverse engineering process.