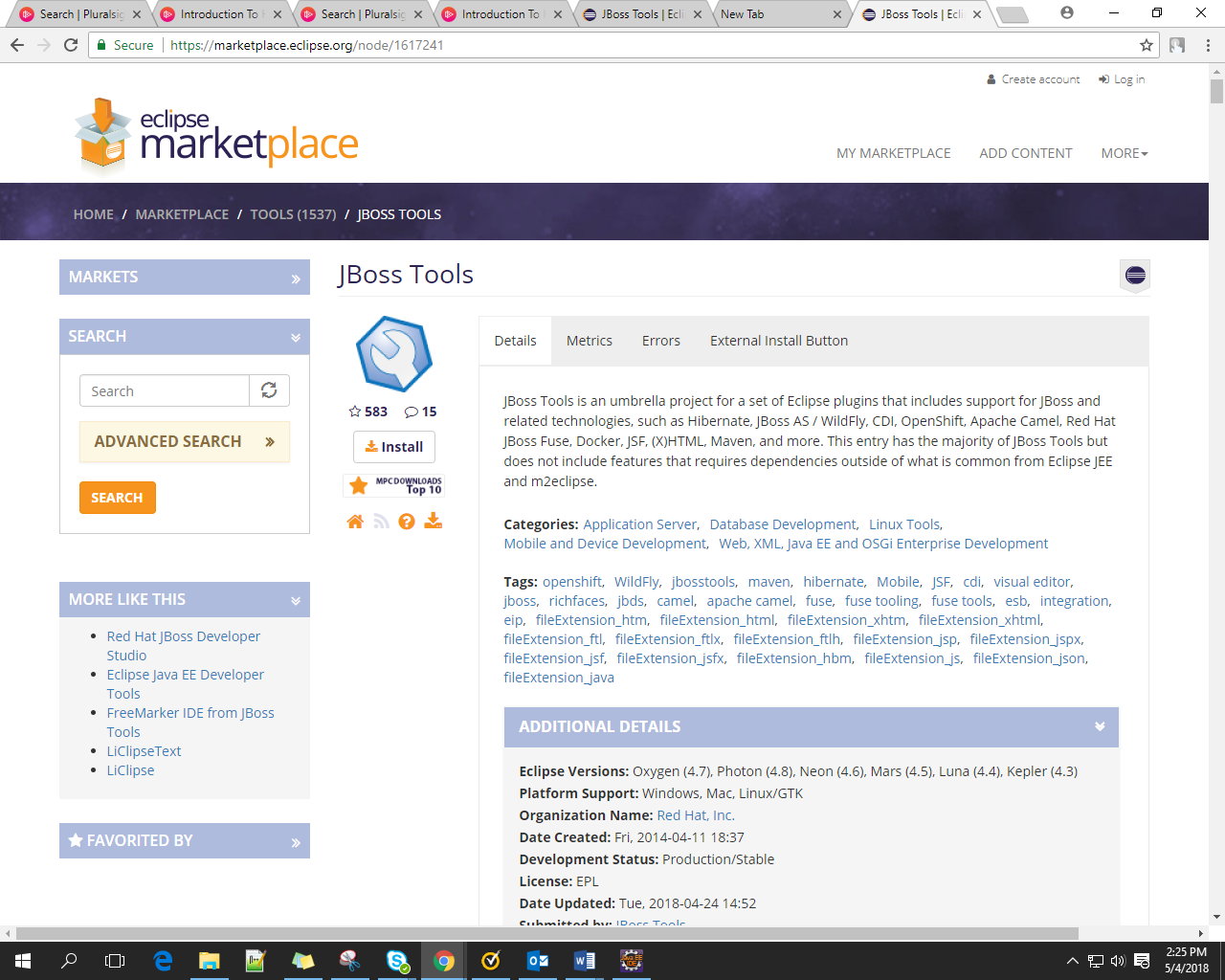
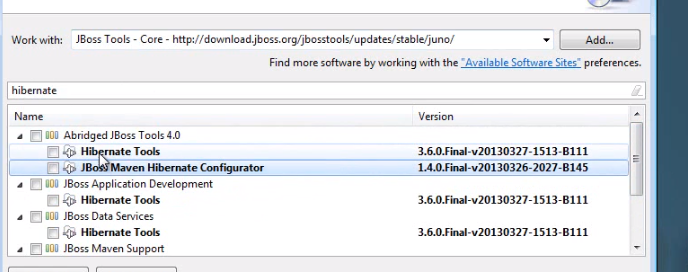
**HIBERNATE**

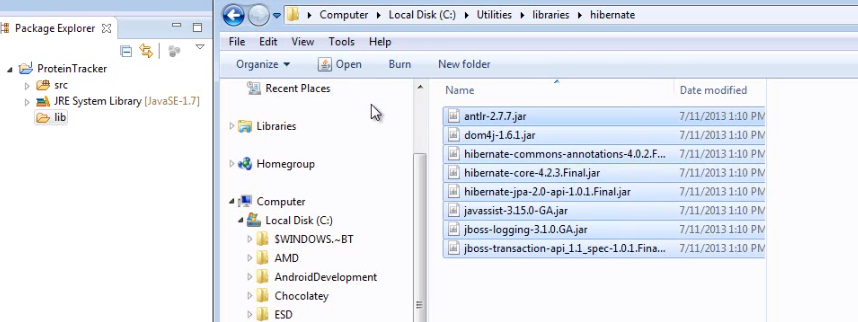
# configuration

## Drap and drop Jboss Tools in to eclipse

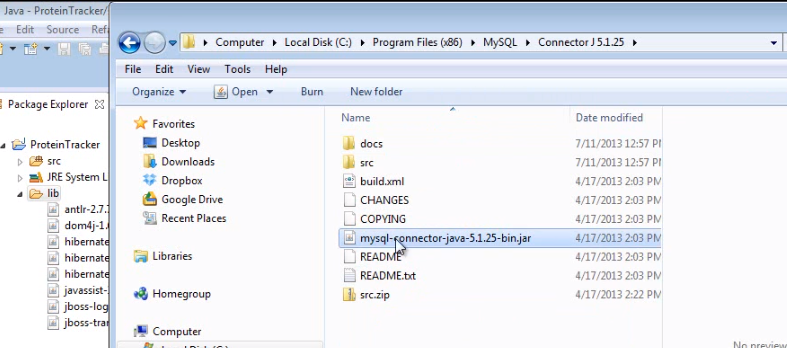




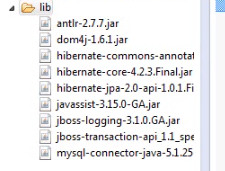
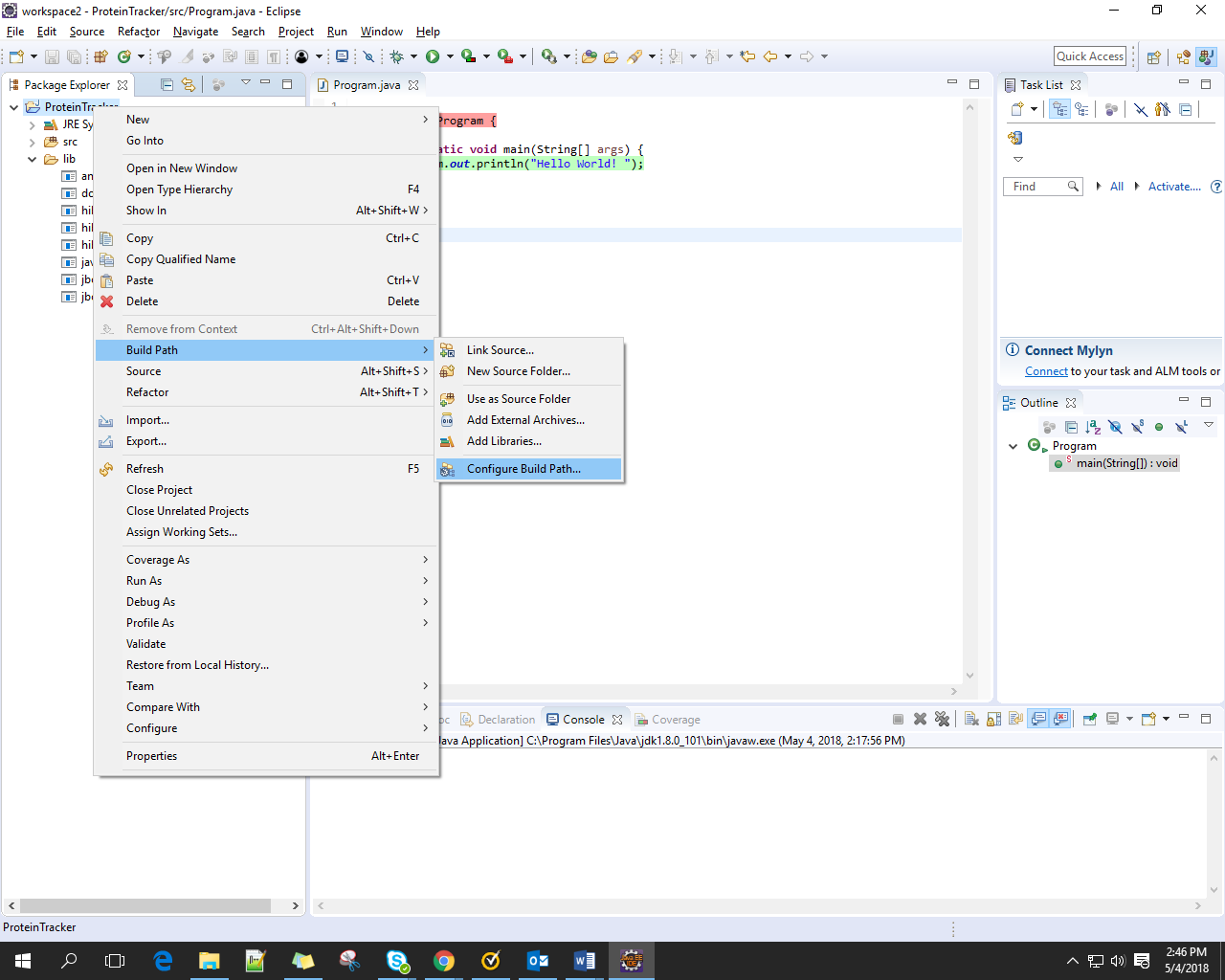
## Add libraries

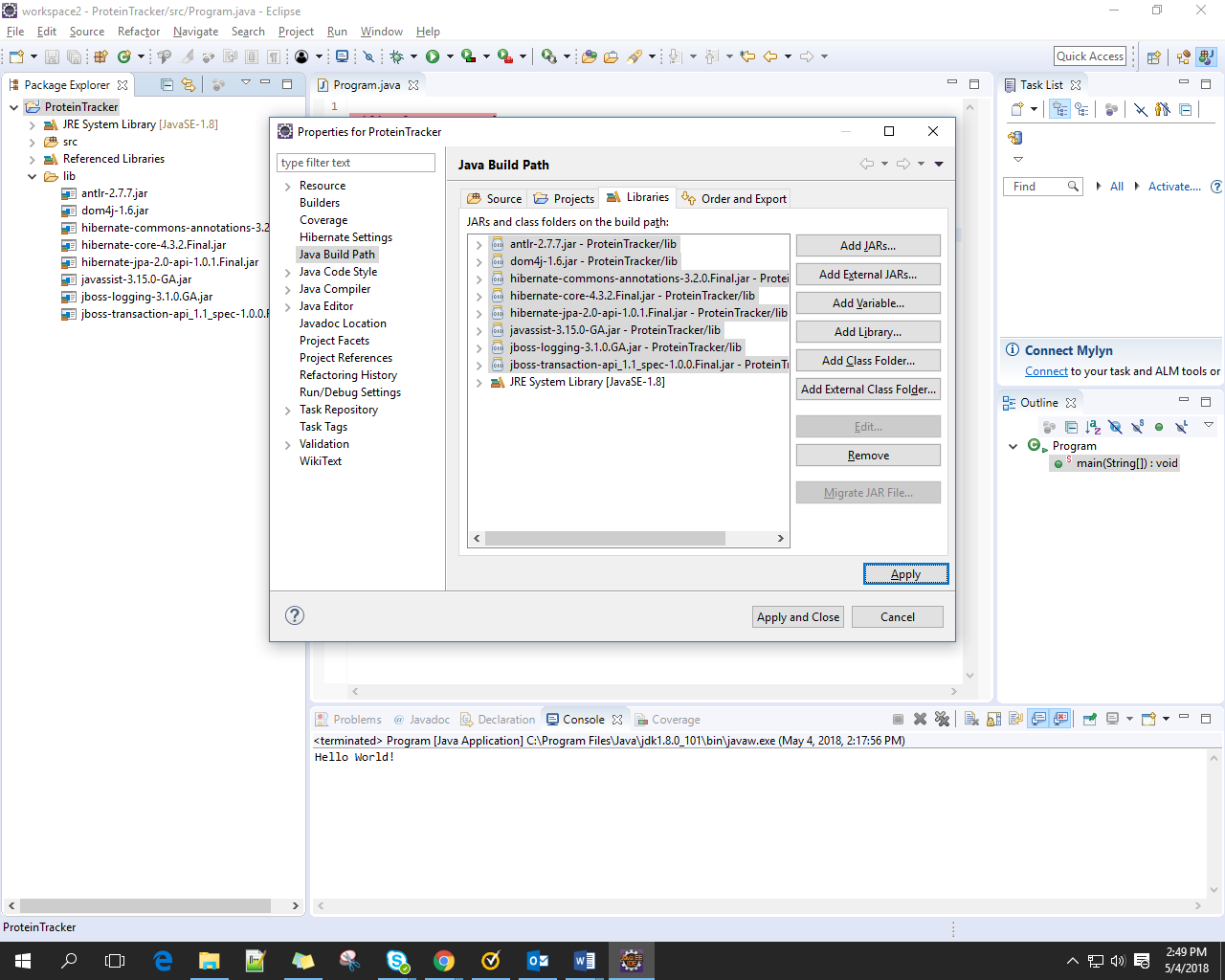


Add file in mysql into lib in eclipse



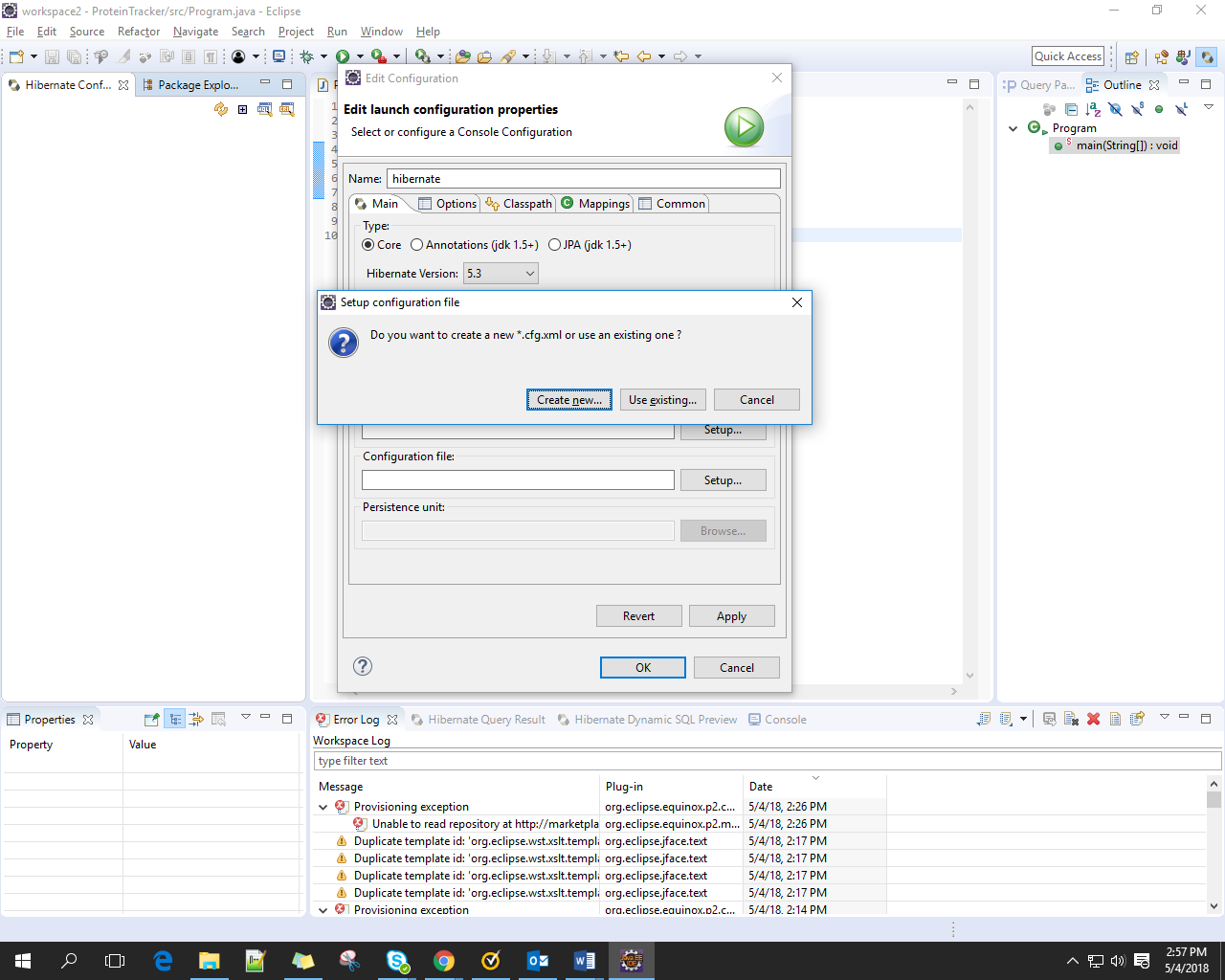
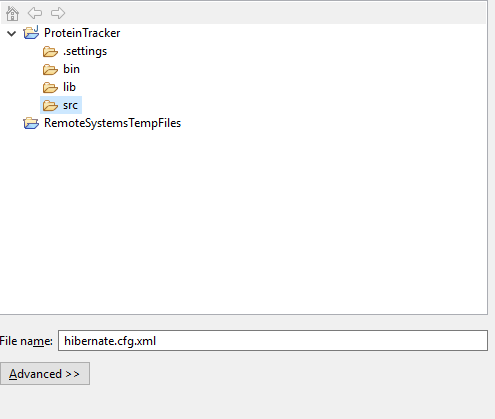
finally

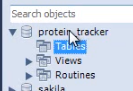


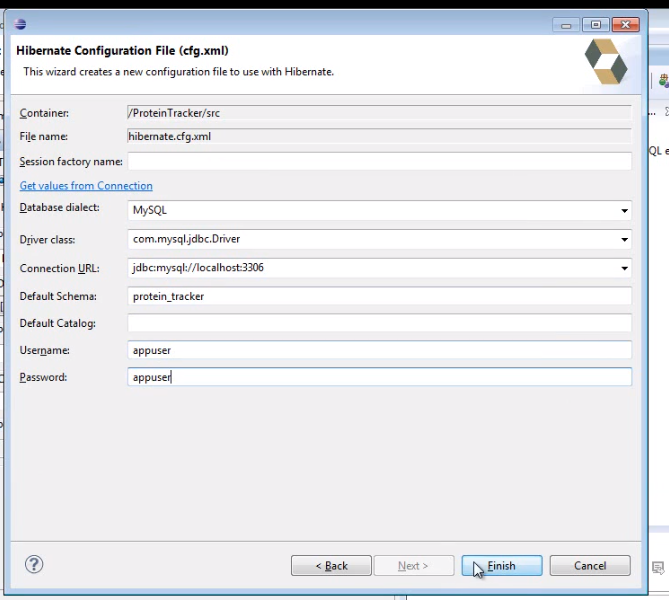


Finally

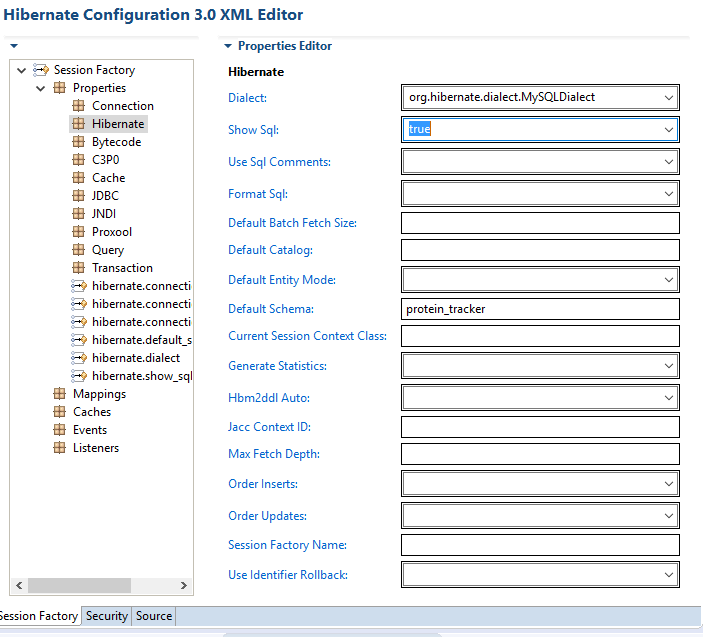
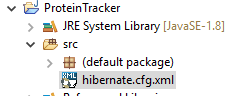


* Create configuration ( project current)
* Create table in mysql





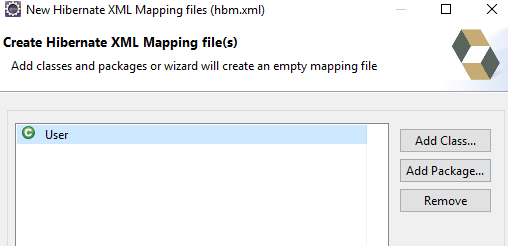
* Result



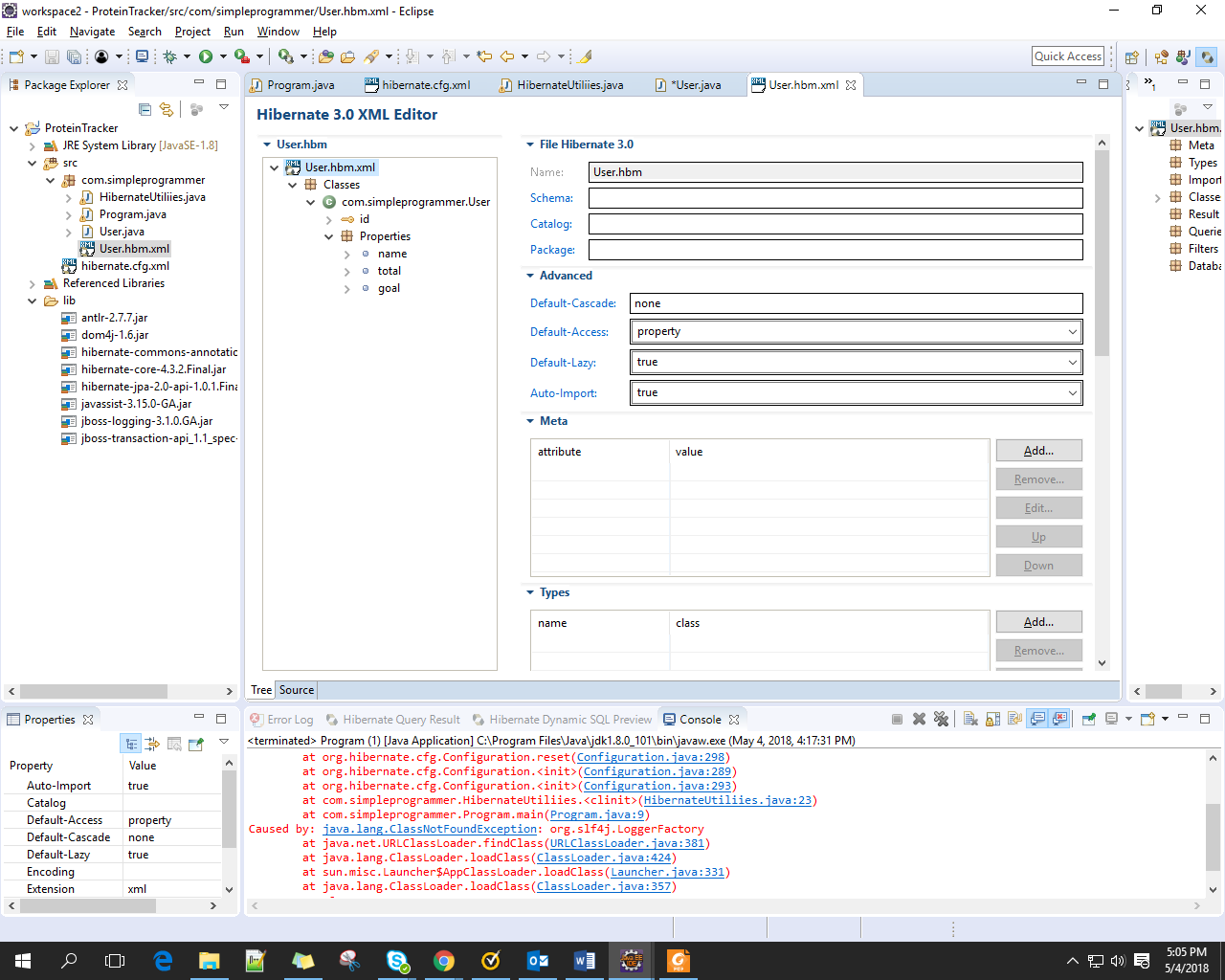
# Basic Mapping

## Create file mapping

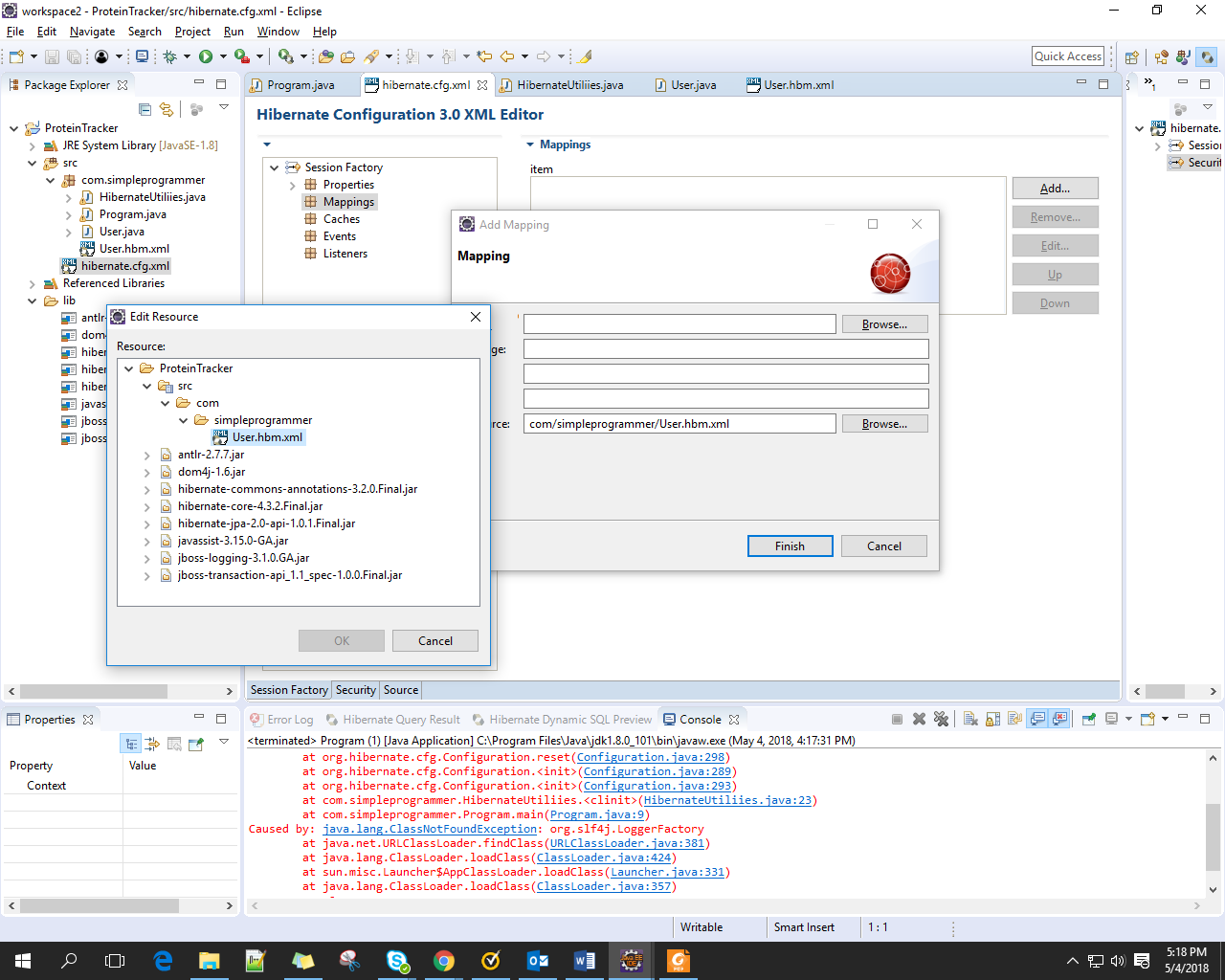


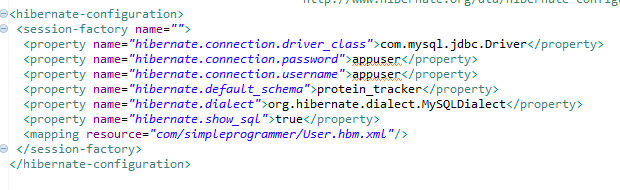


Result

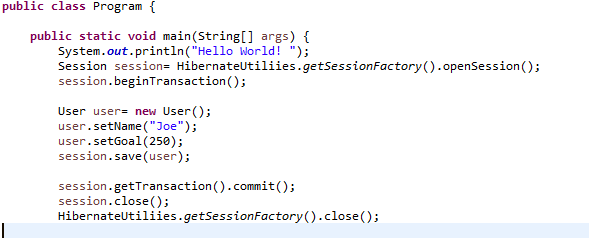




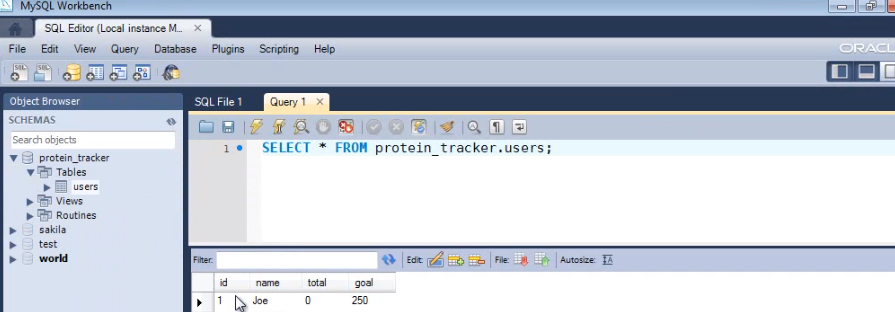




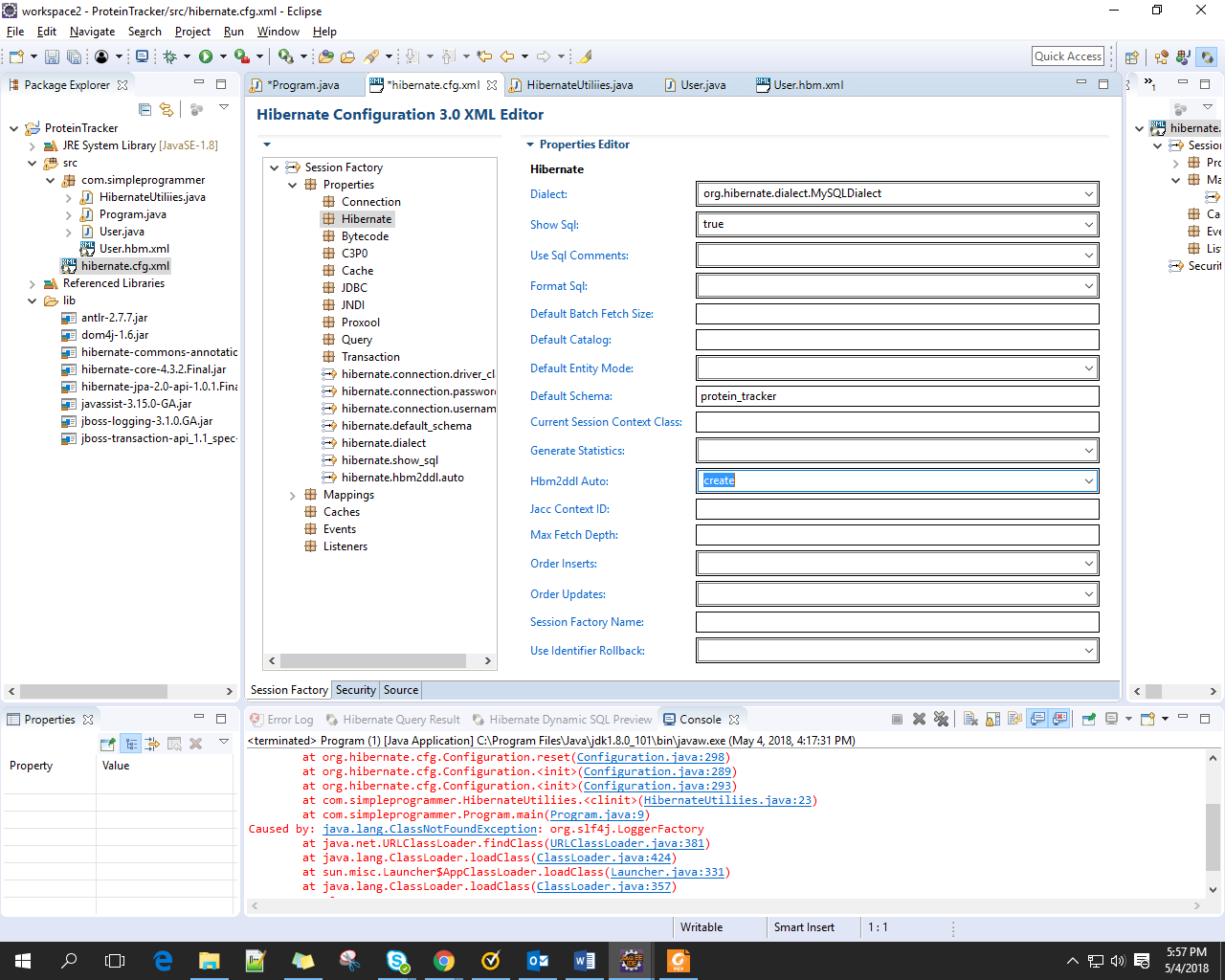
## Save data base

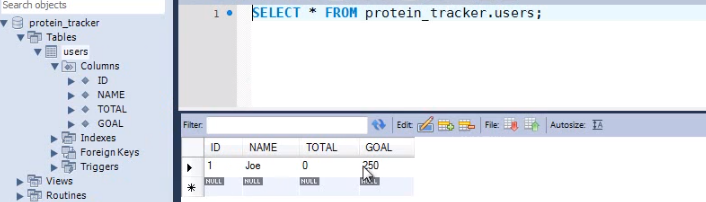


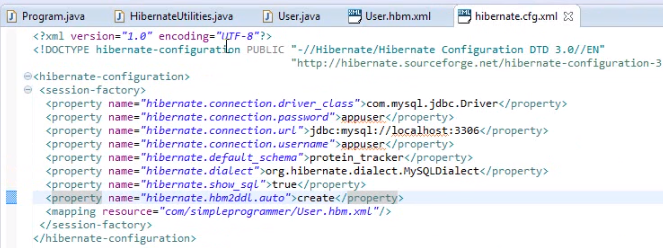
* result



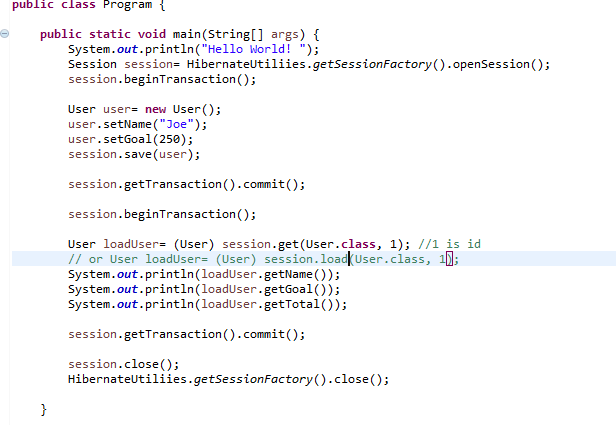
## create table using auto generation(assume not table in database execuse after create file mapping)



* result

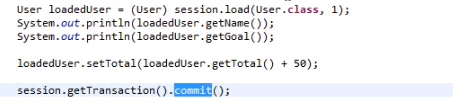
file hibernate.cfg.xml

## Getting data back

* result



## Auto updating (get data then manipulate data )



 Result

# MAPPING RELATIONSHIPS

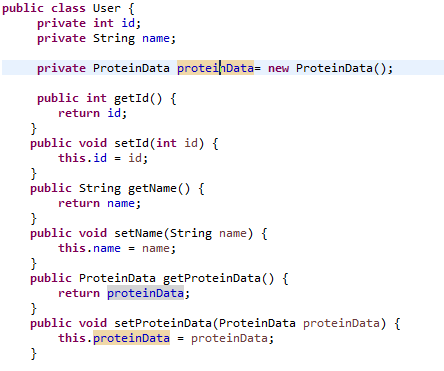
## Distinction value type and entity type

* Value type would be a person’leg we would normal make this piece of data a value type becase it don’t make sense to exist outside of the context of the person another way we would never reference a person’leg by itself we get data from a person that is associated to it, the leg doesn’t stand on it own
* Entity types : person itself was an entity types

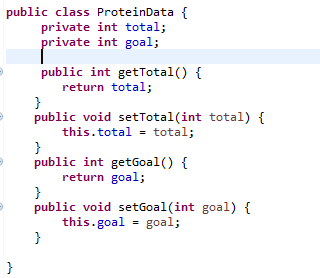
Ex: proteinData in User

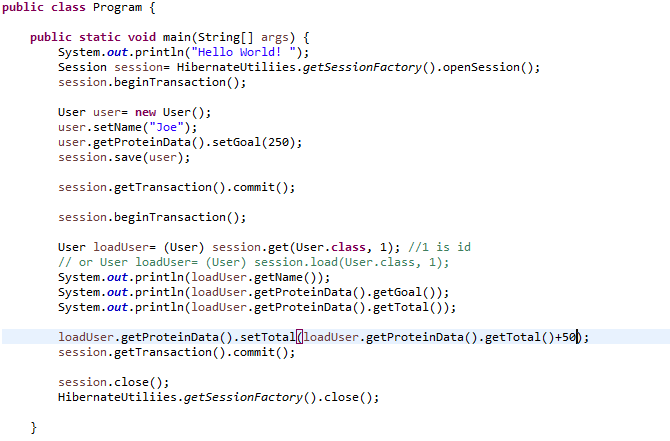
User will have a reference to some protein data, but that protein data won’t make much sense on its on.

## Create component



Create class ProteinData



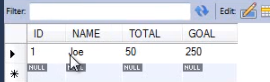


## Mapping the component

Want all of that data to be in a single table

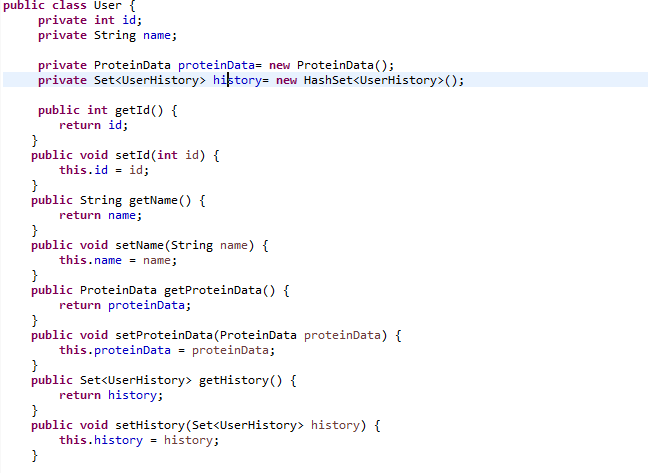


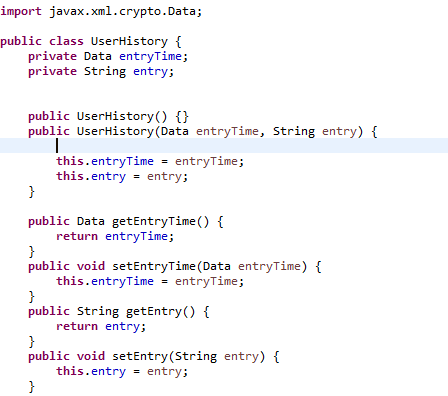
* Result



Don’t change result in SQL just change in java code.

## Create the user history class

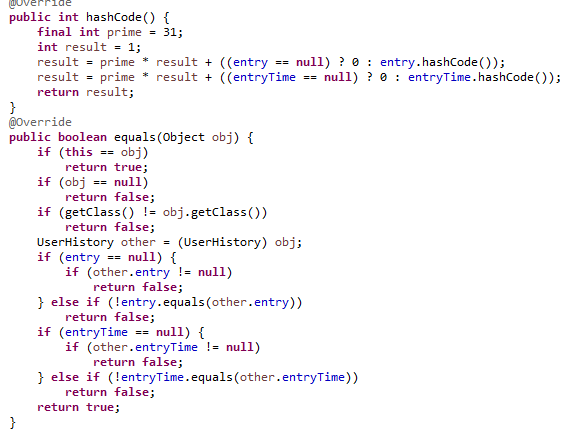




* Create default constructor, hibernate need it to work.



* Create Hashcode Hibernate going to need to build and compare this value types to know if there the same or not Because hibernare is going to need to do some comparisons to see whether or not theses thing are equal because we don’t have a unique id for any of these

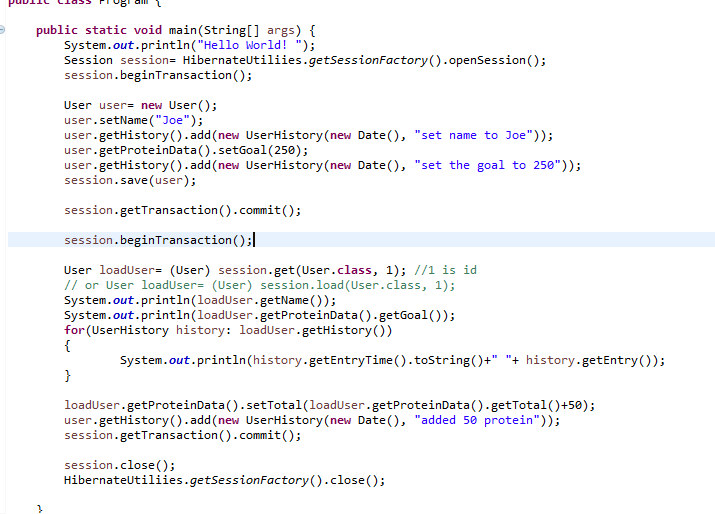
()

## Mapping the set

Value type it doesn’t exist on its own were just using this as a foreign key to the user table so that essentialy mapping this user data to this user history data

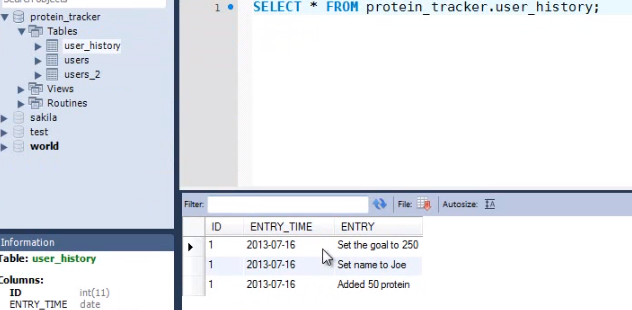


add information into history collection(hashset)



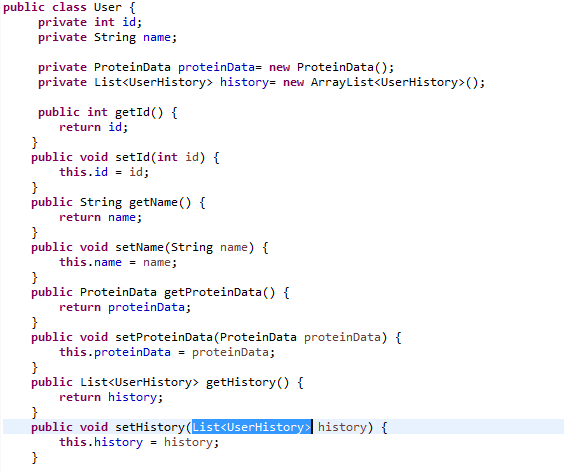
Result after run

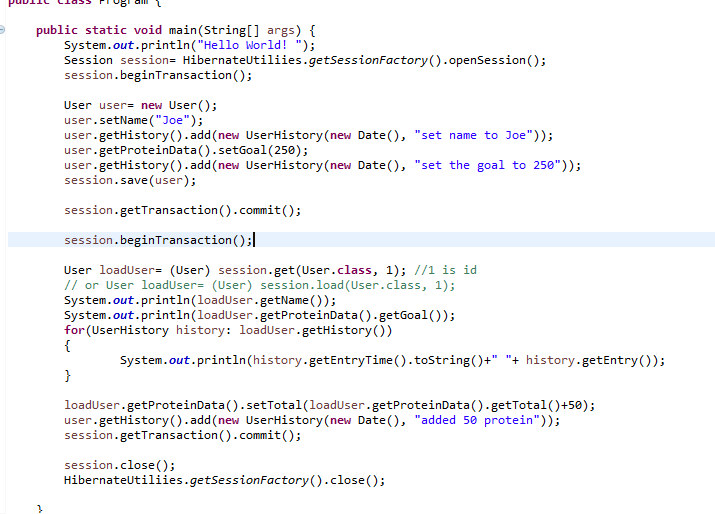
Automate create table USER\_HISTORY



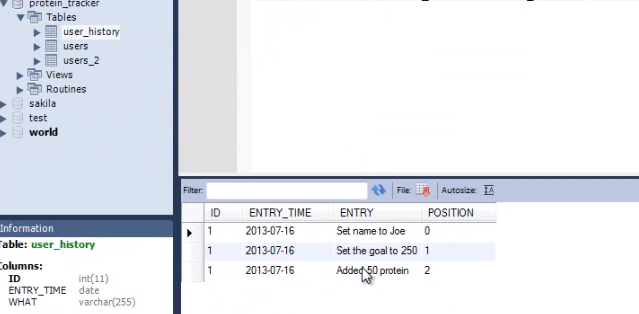
## Mapping A lists



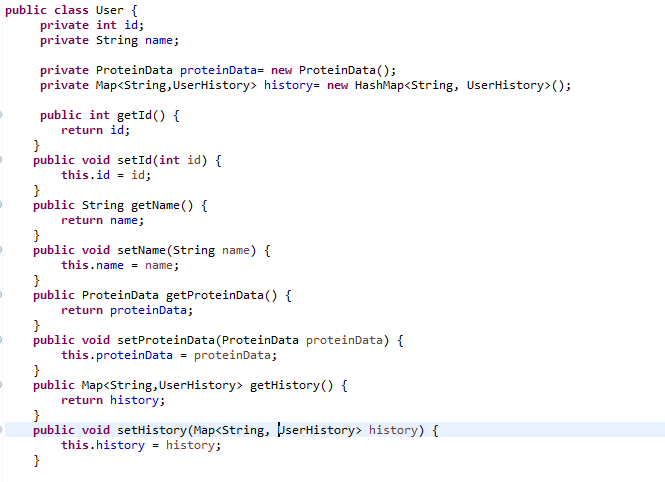


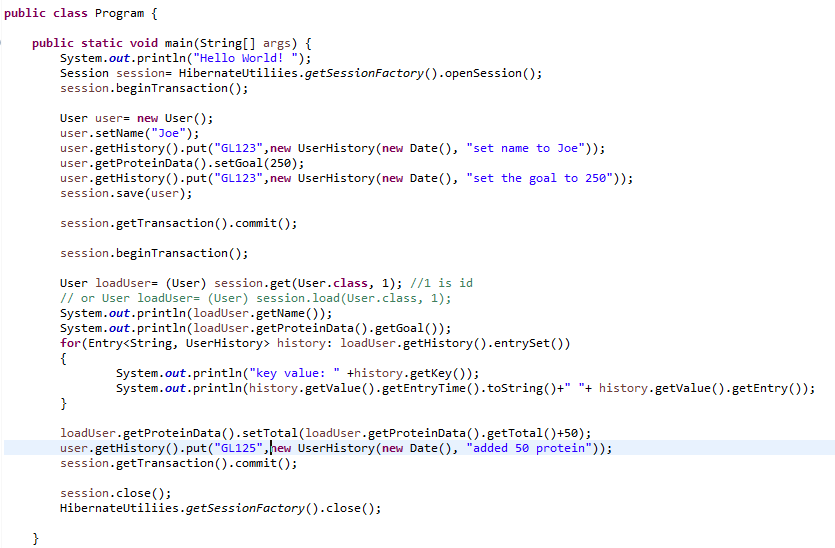


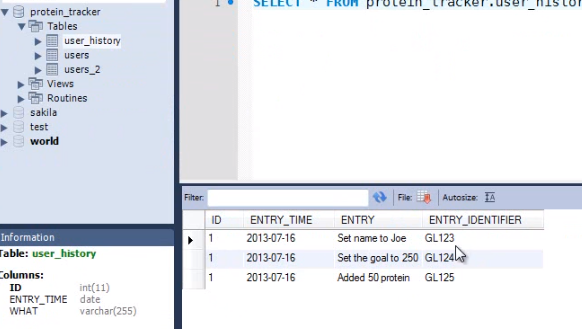
Result



## Mapping a map



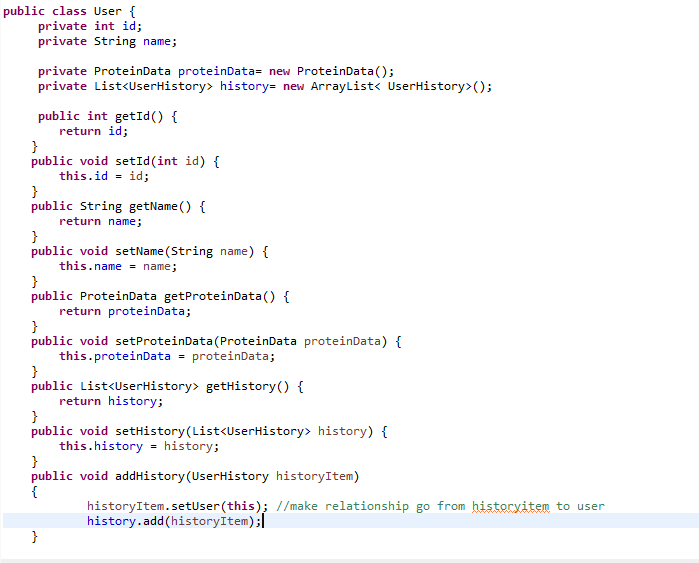


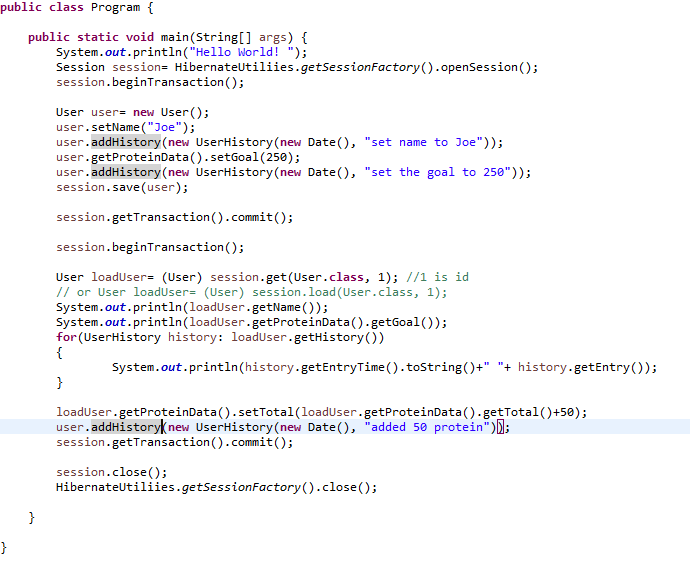


## Changing UserHistory to enity value

Create relationship bewean User and UserHistoty (one two many)



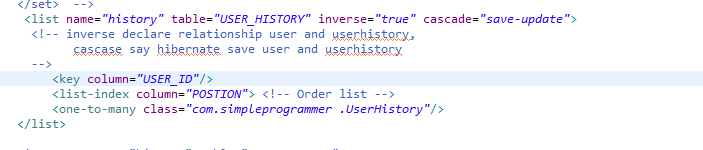




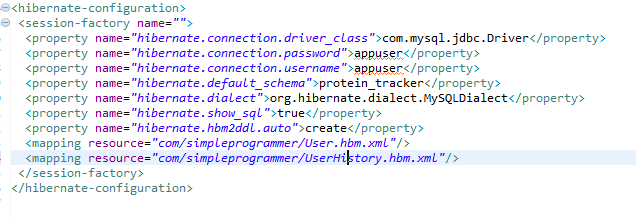
## Mapping

Create ProteinData.hbm.xml

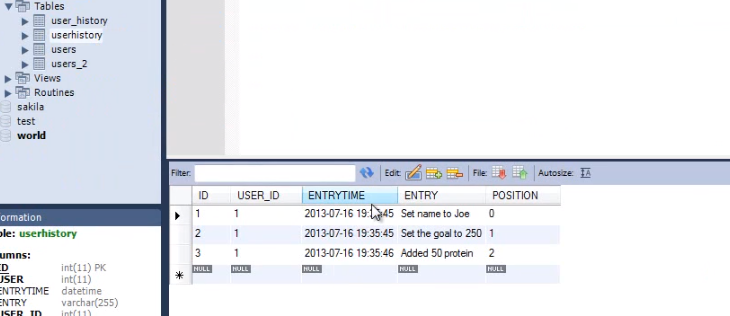
Change file User.hbm.xml

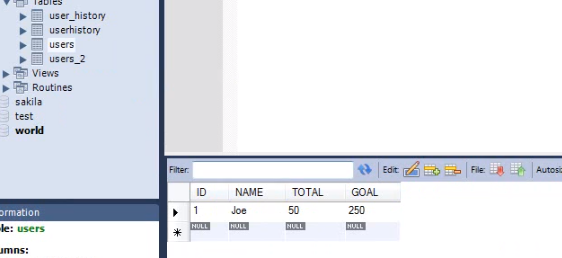


Change file hibernate.cfg.xml

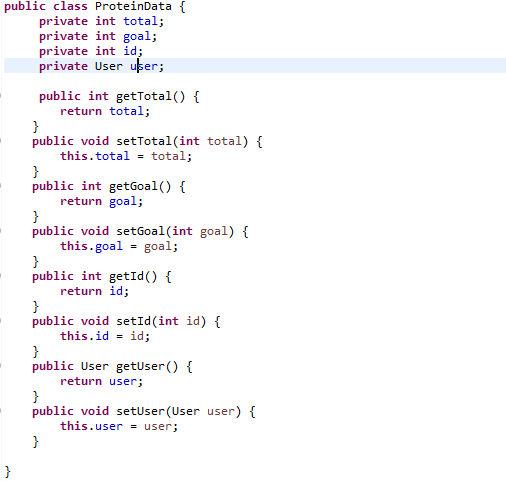


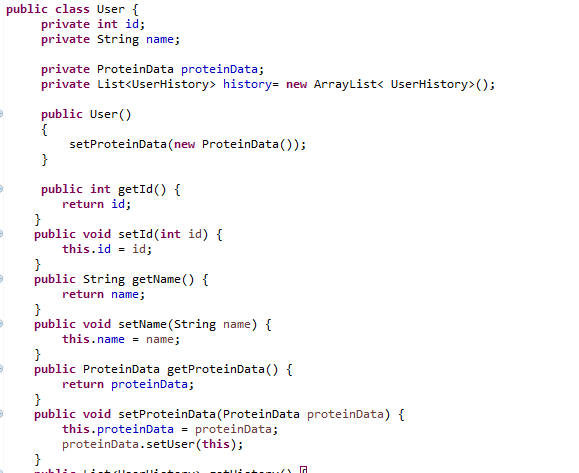
Result





## Create relationship beween user and proteindata(one-to-one)

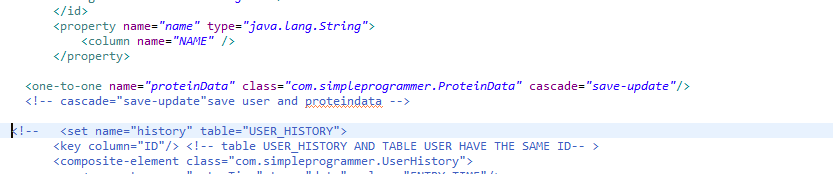


create contructor

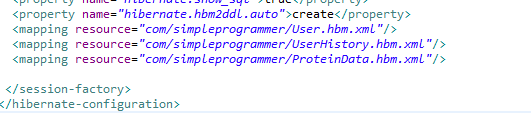
## Creat file mapping ProteinData.hbm.xml



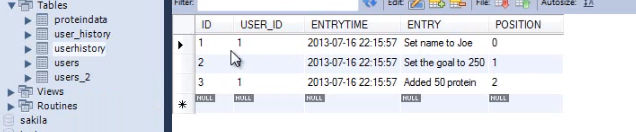
Change file user.hbm.xml

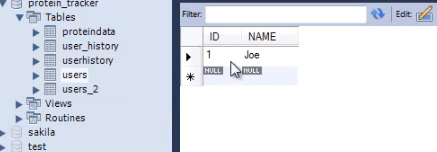


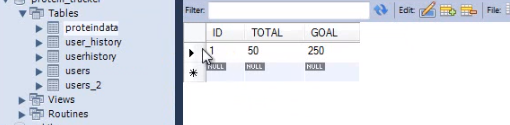
Change file hibernate.hbml.xml



Result



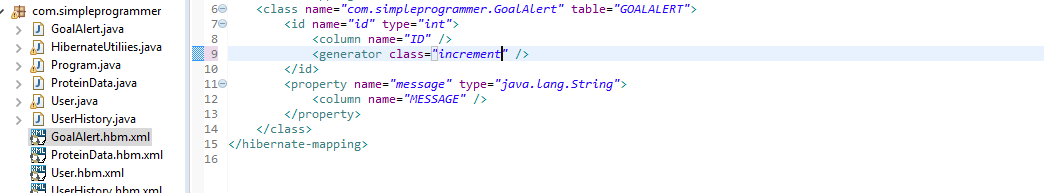


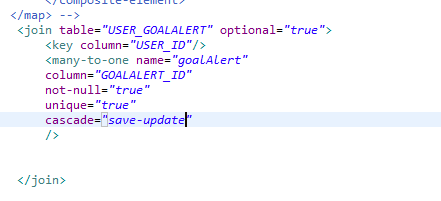


## Create mapping a joint table

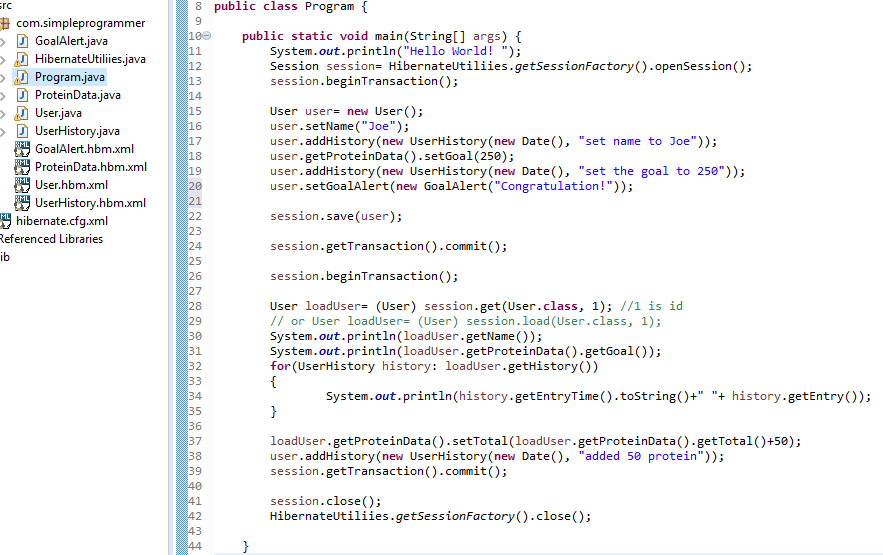
* Create class GoalAlert



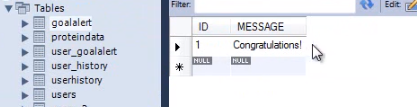
* Create file mapping GoalAlert.hbm.xml
* create veriable in class User and setter, getter
* adjust file mapping user.hbm.xml add information

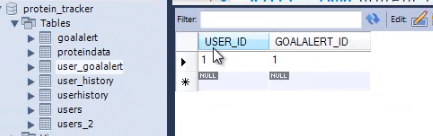


* Add formation in Programe

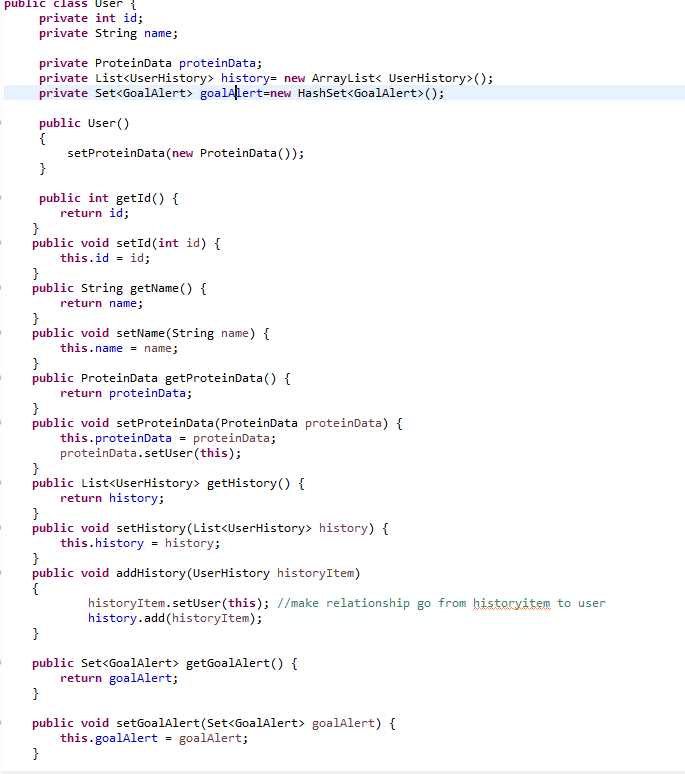


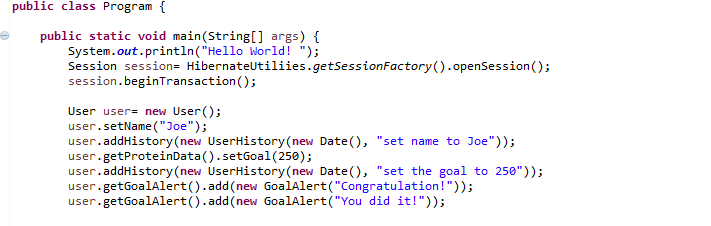
* Result automate create table goalalert and user\_goalalert





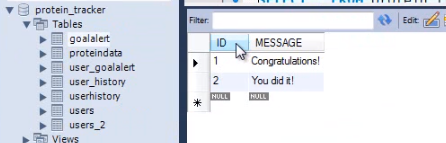
## Mapping To many

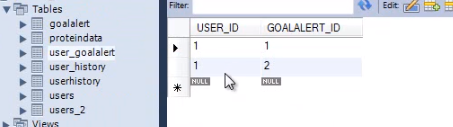
* Change class User
* change Class Program

change file user.hbm.xml

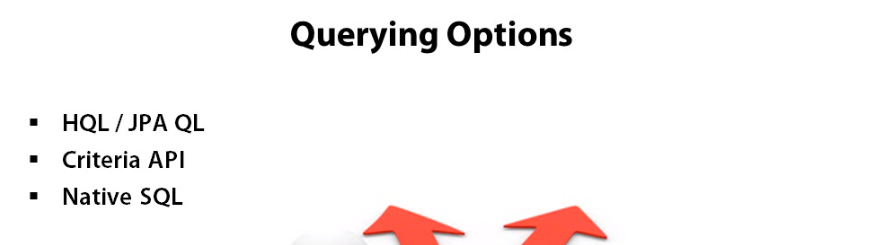


Result

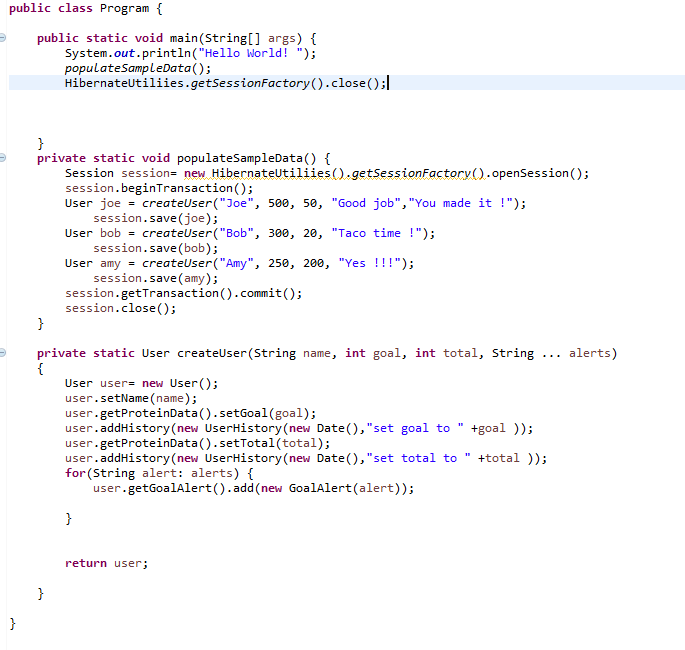


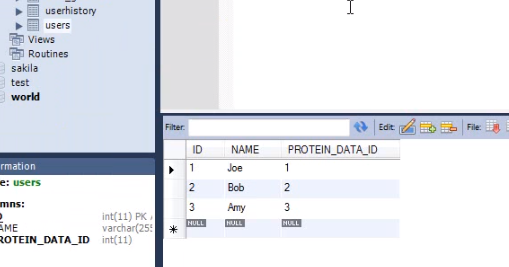


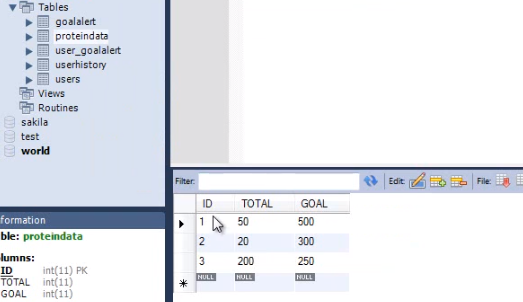
# QUERYING

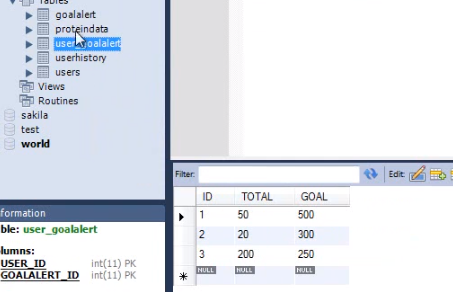


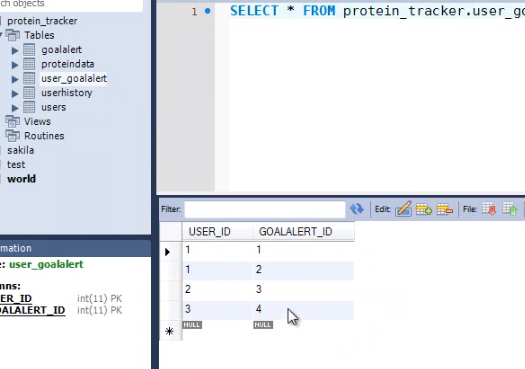
## Create some database

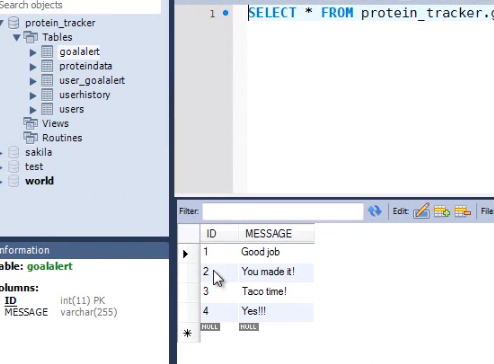
 Result



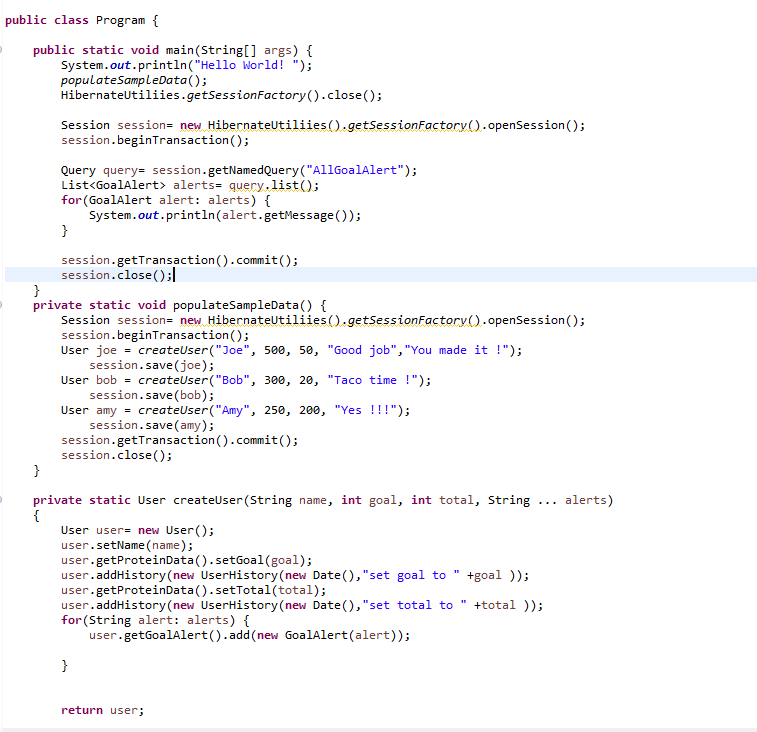




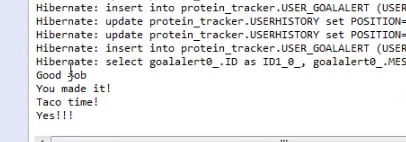




## Query by name

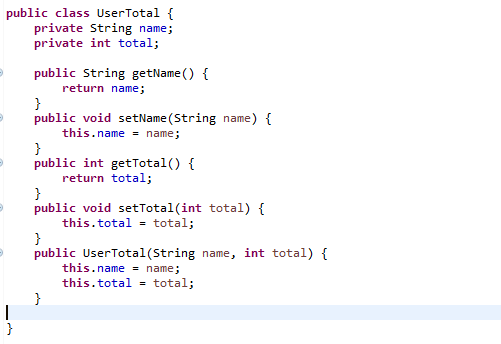


Result



## Create value from query

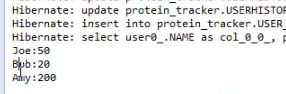
* Create class UserTotal



* Create value for UserTotal and print out



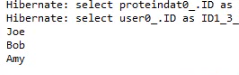
Result



## Query use criteria



Result



* Add condition by criteria



# Advanced topics



