

## LEGEND\_OF\_RETRO\_HIBERNATE

Pour le fonctionnement du Hibernate:

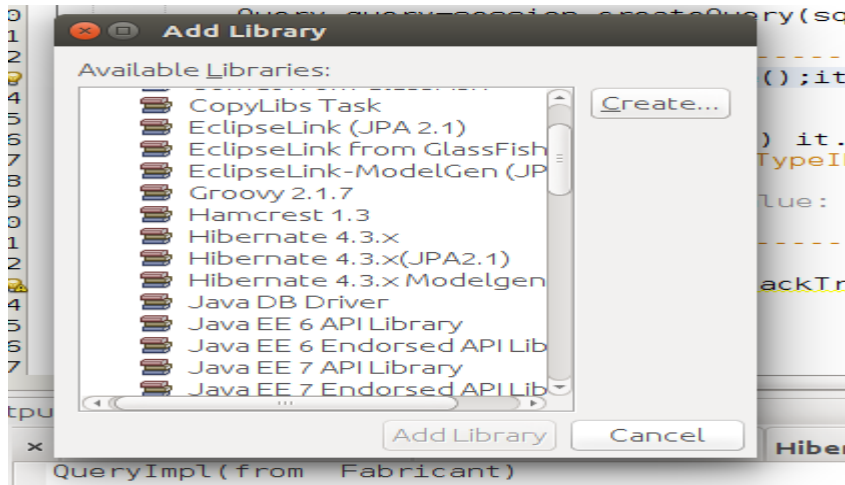
1. Quand vous chargez le code source que vous aurais donné c'est possible d'avoir un message ("mysql-connector-java-5.1.23-bin.jar" file/folder could not be found)

Solution: Run-->Clean/Build.

—>clique droite sur nom du projet->Properties-->Libraries->Add jar/Folder

(Normalement vous trouvez dans dist/lib/ un mysql-connector-java-5.1.23-bin.jar)

2. Si on veut une librairie, cliquez droit sur Libraries-->on a besoin de Hibernate 4.3.x et JPA2.1  
voir image



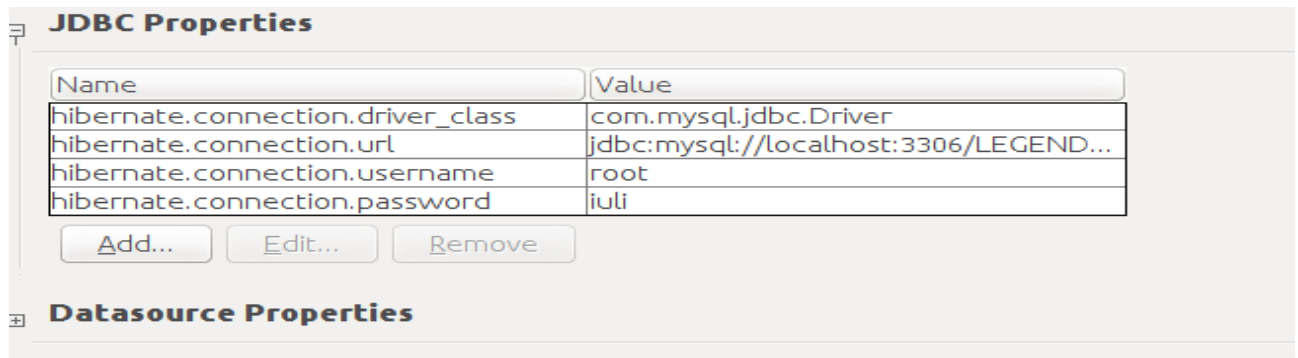
Normalement on ne doit pas avoir un probleme de dependance parce que le rajoute dans code source /dist/lib avec chemin relative (pas absolute).

j'ai testé sur mon petit laptop sur windows apres avoir fait la commande clien/build et chemin relatif du .lib Le resultat est que ça n'a pas demandé aucune dependance.

3. Entrez dans <default package> -->hibernate.cfg.xml -->Design

(Vous aurez cette image) -vous mettez votre nom d'utilisateur et mot pass de votre MySQL cliquez sur .... et vous pouvez utiliser "Edit".

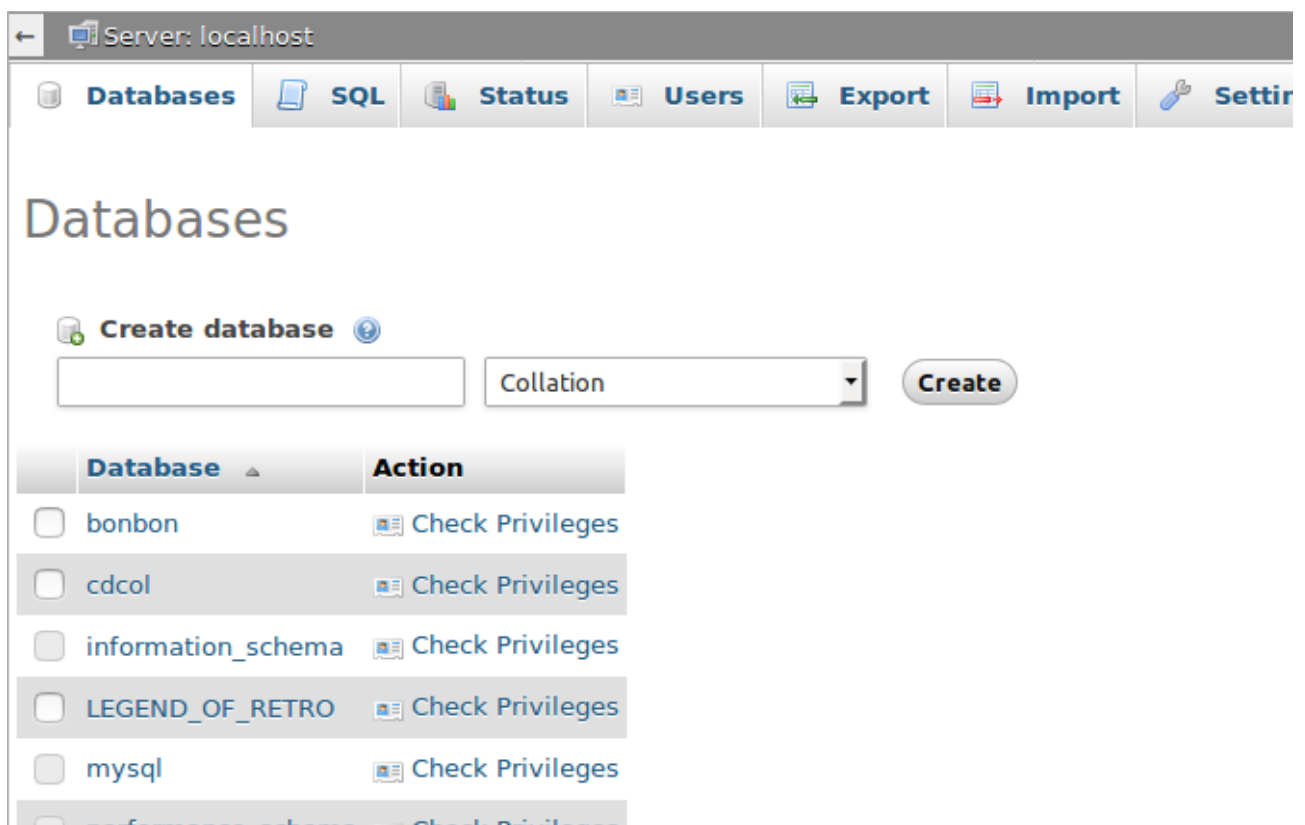
**Iulian  
AMARIEI**



4. Vous créez une base de données avec le nom: LEGEND\_OF\_RETRO et importez le fichier

FABRICANT.sql que je vais vous donner.

(Chez moi si j'ai essayé d'importer directement la BDD ça donne une erreur, pour cette raison vous créez la BDD vide et après importez la table.)

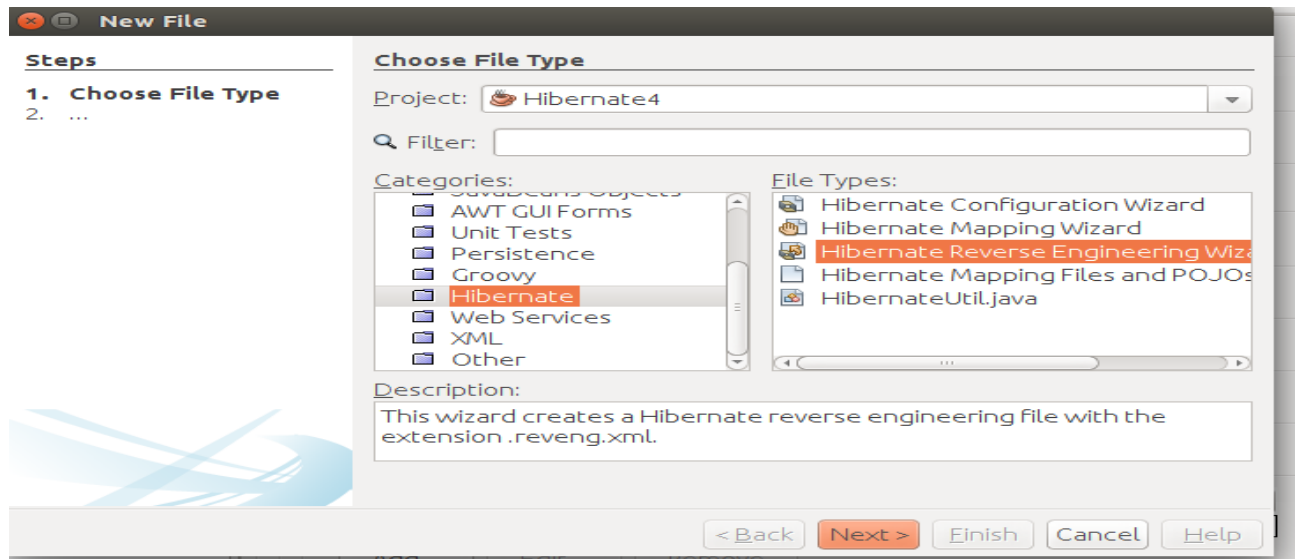


Après cette étape normalement ça doit marcher et ajoutez vers la BDD et aussi de BDD vers "main" et affichez dans output chaque élément avec la commande run.

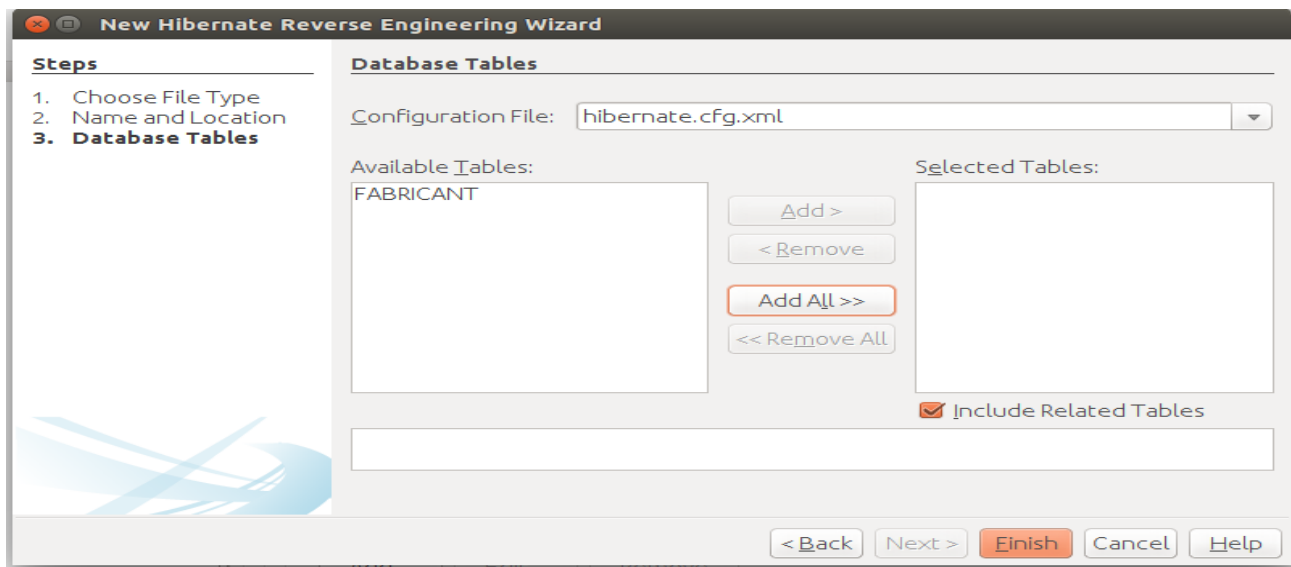
**5.Important.** Vous voulez ajouter des classes dans projet.

-Vous créez la table avec des attributs mais vide.

-cliquez droite Source Packages et voir l'image: "Hibernate Reverse Engineering Wizard"



-next->



-Add: le table est le resultat voir l'image en bas :une classe jolie qui a été crée qui automatiquement :)

6.Si il  
y a  
une  
erreur

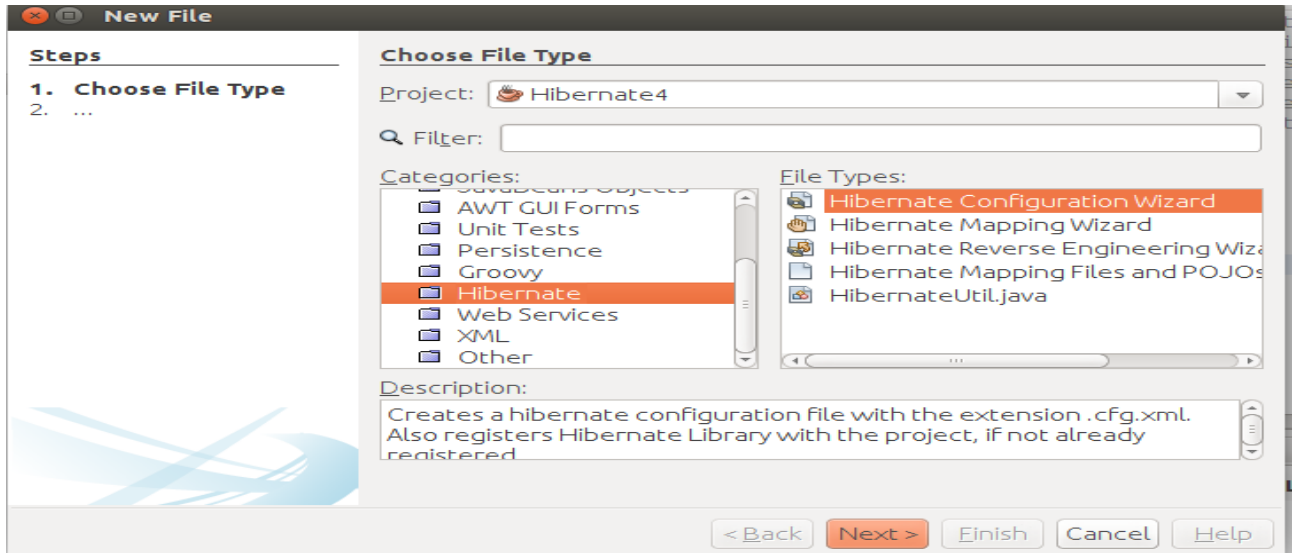
```
public class Fabricant implements java.io.Serializable {  
    private Integer idFabricant;  
    private String nomFabricant;  
    public Fabricant() {  
    }  
    public Fabricant(String nomFabricant) {  
        this.nomFabricant = nomFabricant;  
    }  
    public Integer getIdFabricant() {  
        return this.idFabricant;  
    }  
    public void setIdFabricant(Integer idFabricant) {  
        this.idFabricant = idFabricant;  
    }  
}
```

## Iulian AMARIEI

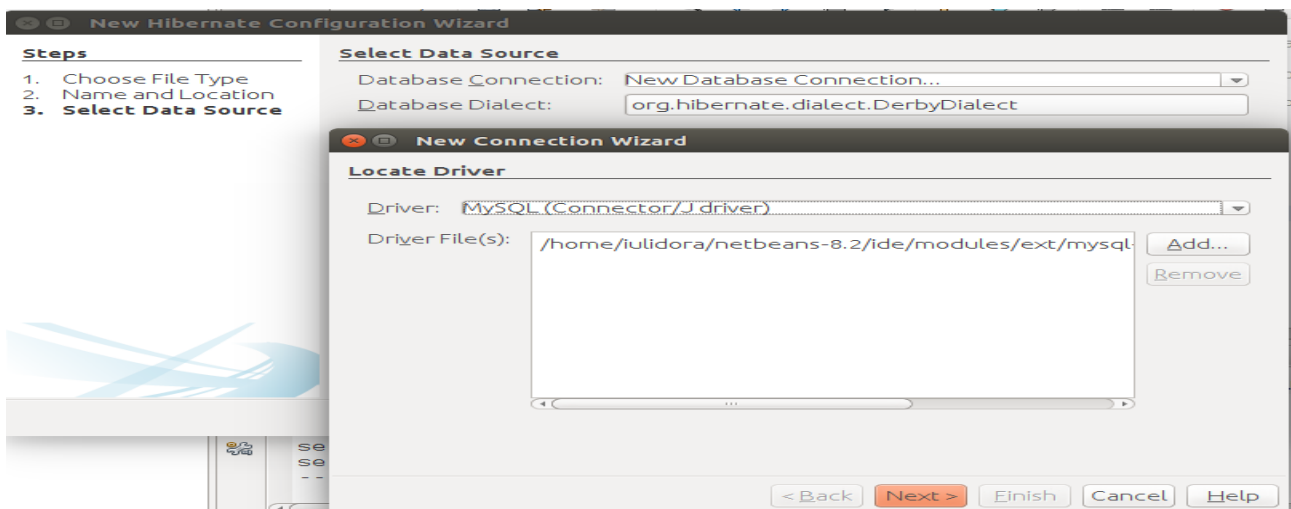
lié au host( le port normal est 3306 comme d'habitude)

vous devrez créer une nouvelle connexion:

Cliquez droit sur "Sources Packages" ---> New -->Other--> voir l'image



-->Next-->Next-->vous arrivez ici:Select Data Source vous sélectionnez : voir l'image:



->next.

Voir 2-em image:important si le port n'est pas 3306

(voir avec une commande query en MySQL:

`SHOW VARIABLES WHERE Variable_name = 'port'`) le vrai port et vous changez le port et Database: nom BDD----> testez la connexion "Test Connection"

--->le resultat: "Connection Succeded" .

Julian  
AMARIEI

Host:

localhost

Port:

3306

Database:

bonbon

User Name:

root

Password:

\*\*\*\*

☐ Remember password

Connection Properties

Test Connection

JDBC URL:

jdbc:mysql://localhost:3306/bonbon?zeroDateTimeBehavior=convertToNull

Connection Succeeded.

< Back

Next >

Finish

Cancel

Help

dist:

BUILD SUCCESSFUL (total time: 3 seconds)