

Hibernate

A Data class

```
package javaapplication3;
public class User {
    private int ID;
    public int getID() { return ID; }
    public void setID(int ID) { this.ID = ID;}

    private String name;
    public String getName() {return name; }
    public void setName(String name) { this.name = name; }

    private String surname;
    public String getSurname() { return surname;}
    public void setSurname(String surname) {this.surname = surname;}

    public String toString(){ return ID+" "+surname+" "+name; }
}
```

HibernateExample

```
package javaapplication3;
```

```
import java.util.Iterator;
```

```
import java.util.List;
```

```
import org.hibernate.Criteria;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
import org.hibernate.Transaction;
```

```
import org.hibernate.cfg.AnnotationConfiguration;
```

```
import org.hibernate.cfg.Configuration
```

HibernateExample

```
public class HibernateExample {  
    private static final SessionFactory sessionFactory;  
    static {  
        try {  
            // 1. Initiation of Hibernate  
            // Create the SessionFactory from standard (hibernate.cfg.xml) config file.  
            Configuration cfg = new AnnotationConfiguration().configure();  
            // 2. Formation of a Hibernate session factory  
            sessionFactory = cfg.buildSessionFactory();  
        } catch (Throwable ex) {  
            // Log the exception.  
            System.err.println("Initial SessionFactory creation failed." + ex);  
            throw new ExceptionInInitializerError(ex);  
        }  
    }  
    public static SessionFactory getSessionFactory() {  
        return sessionFactory;  
    }  
}
```

HibernateExample

```
public static void main(String[] args) {  
    HibernateExample m = new HibernateExample();  
    m.createUsers();  
    m.displayUsers();  
}  
  
public void createUsers() {  
    Session session = getSessionFactory().openSession(); // 3. Opening of Hibernate session  
    Transaction tx = session.beginTransaction(); // 4. Beginning of a transaction  
    // 5. Formation of a user  
    User u = new User();  
    u.setName("John");  
    u.setSurname("Smith");  
    //  
    session.save(u); // 6. Saving the user in a database  
    tx.commit(); // 7. Confirmation of the transaction  
    session.close(); // 8. Closure of the Hibernate session  
}
```

HibernateExample

```
public void displayUsers() {  
    Session session = sessionFactory().openSession(); // 3. Opening of the Hibernate session  
    Transaction tx = session.beginTransaction(); // 4. Beginning of a transaction  
    // 5. Formation a SQL query concerning a list of users to a base  
    Criteria criteria = session.createCriteria(User.class);  
    List users = criteria.list(); // 6. Execution of the SQL query  
    // 7. Iterating after a result of the SQL query  
    for (Iterator it = users.iterator(); it.hasNext();) {  
        User user = (User) it.next();  
        System.out.println(user);  
    }  
    //  
    tx.commit(); // 8. Confirmation of the transaction  
    session.close(); // 9. Closure of the Hibernate session  
}
```

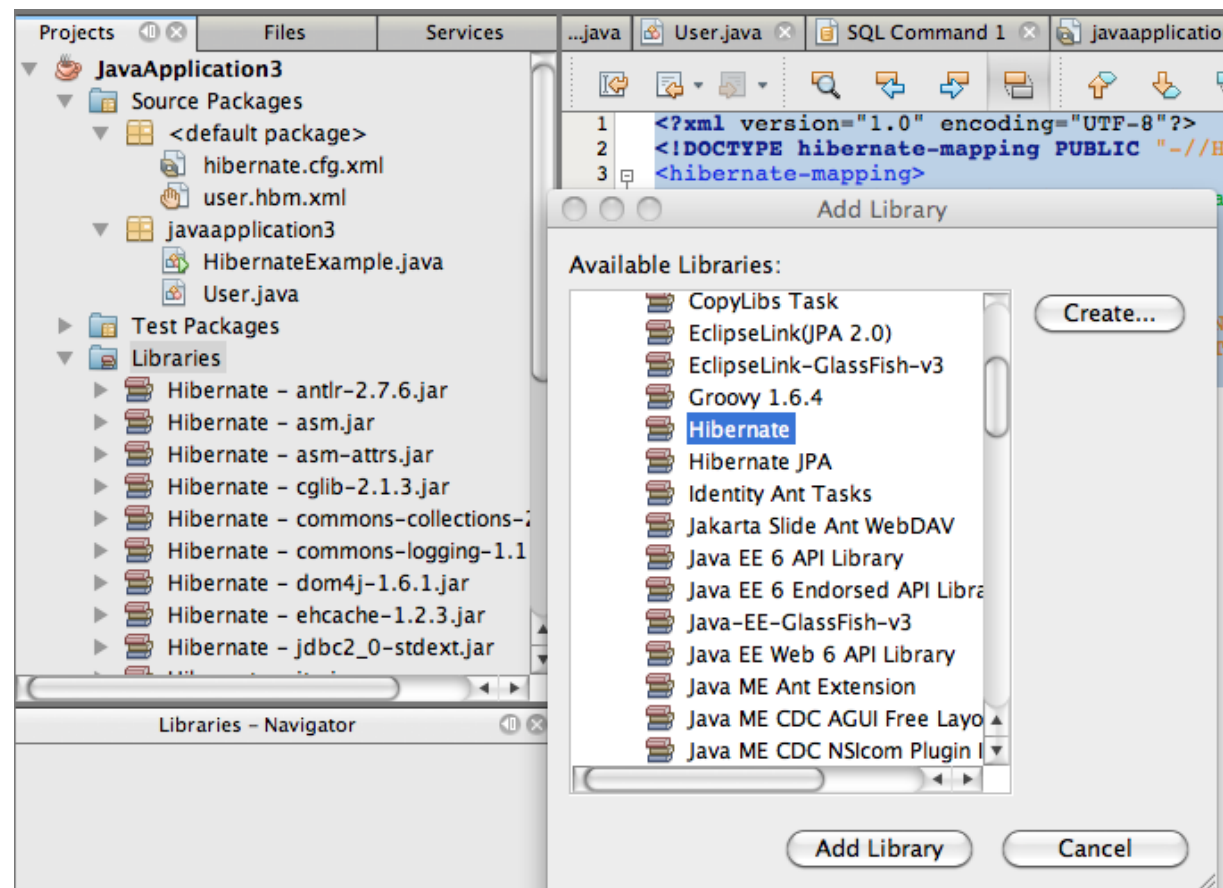
Criteria

```
Criteria crit = session.createCriteria(Insurance.class);  
    crit.add(Restrictions.like("insuranceName", "%a%"));  
//Like condition  
    crit.setMaxResults(5); //
```

<http://docs.jboss.org/hibernate/core/3.3/reference/en/html/querycriteria.html>

Netbeans

1 – add library to the project

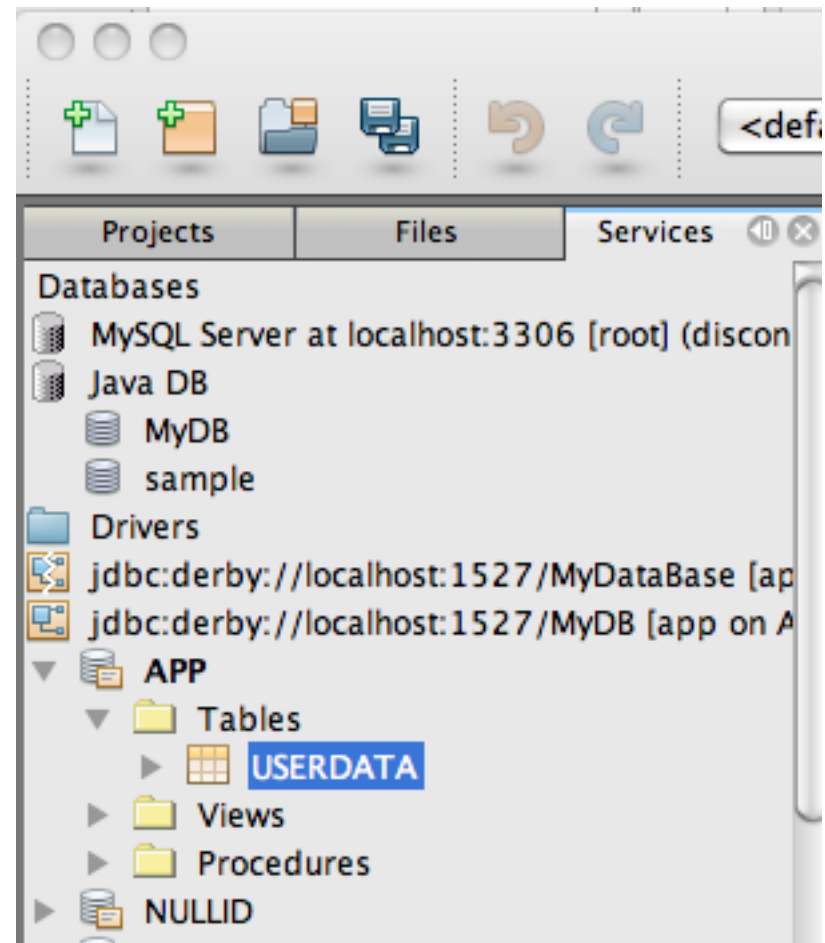


Netbeans

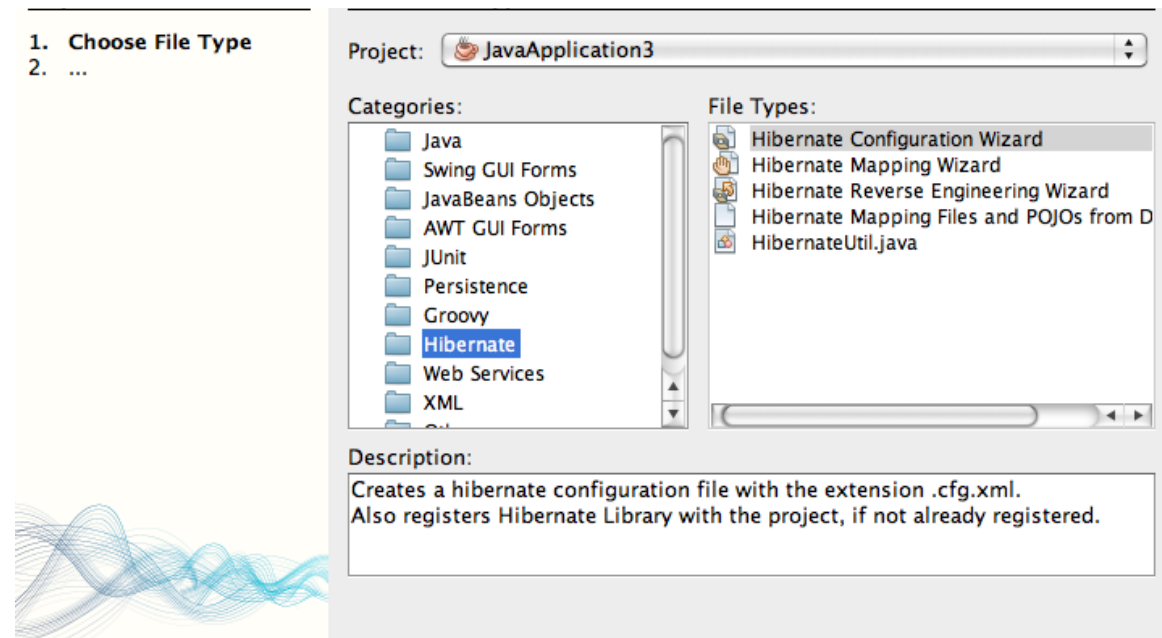
2 – choose a DBMS

3 – create a DB

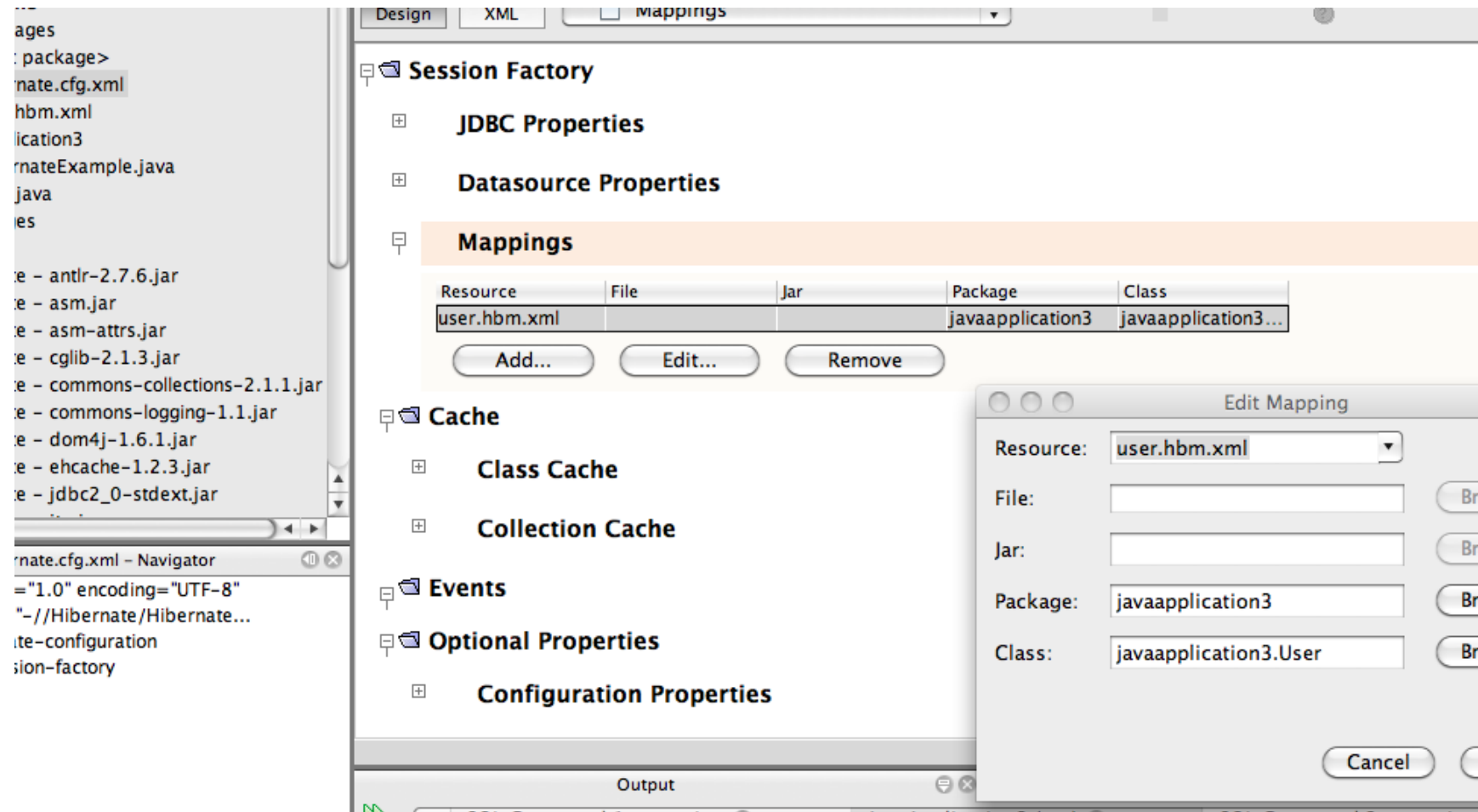
4 – create Tables



Create hibernate.cfg.xml



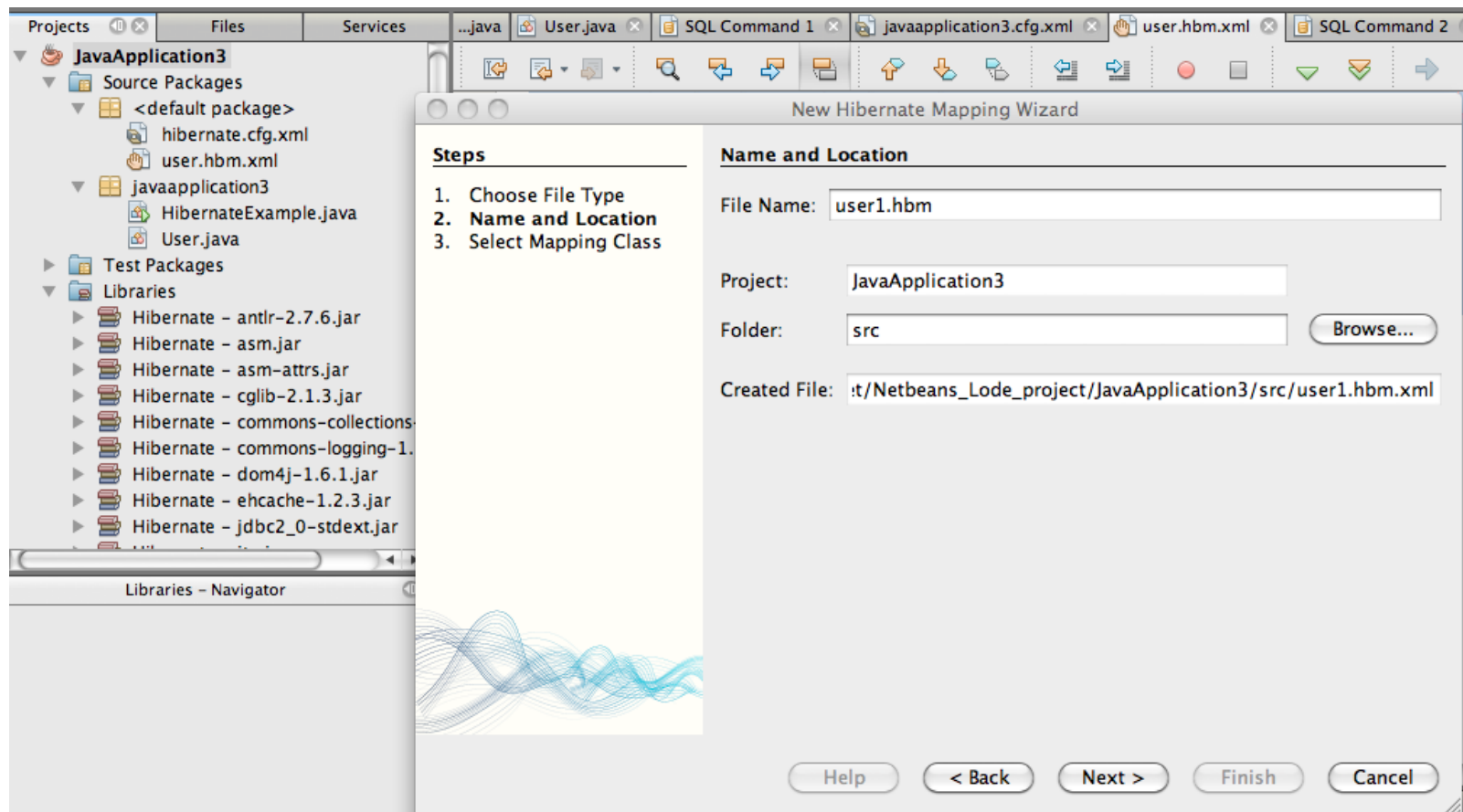
Create hibernate.cfg.xml



Create hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.DerbyDialect</property>
    <property name="hibernate.connection.driver_class">org.apache.derby.jdbc.ClientDriver
  </property>
    <property name="hibernate.connection.url">jdbc:derby://localhost:1527/MyDB
  </property>
    <property name="hibernate.connection.username">app</property>
    <property name="hibernate.connection.password">app</property>
    <mapping class="javaapplication3.User" file="" jar="" package="javaapplication3"
      resource="user.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

Create mapping



Create mapping

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate
Mapping DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-
mapping-3.0.dtd">
<hibernate-mapping>
  <class name="javaapplication3.User" table="USERDATA">
    <id name="ID" type="int">
      <generator class="increment"/>
    </id>
    <property name="name" column="THENAME" type="string"/>
    <property name="surname" column="THESURNAME"
type="string"/>
  </class>
</hibernate-mapping>
```

Extended tutorial

[http://www.roseindia.net/hibernate/
index.shtml](http://www.roseindia.net/hibernate/index.shtml)