# Lab 5 – HSQLDB Database

## ~~Q1. Java/JDBC application~~

* ~~Download the example project r1userDatabase.zip.~~
* ~~Unzip it and import into eclipse.~~
* ~~Start the HSQL database server.~~
* ~~Start the HSQL database client.~~
* ~~Create USER table (DDL is in doc/readme.txt)~~
* ~~Run Insert.java and Select.java~~

## ~~Q2. J2EE/JDBC application~~

* ~~Shutdown the database server in the previous project.~~
* ~~Create a Dynamic Web application.~~
* ~~Copy the db folder across.~~
* ~~Copy the lib folder across.~~
* ~~Add hsqldb.jar to the Build Path~~
* ~~Copy hsqldb.jar into WebContent/WEB-INF/lib~~
* ~~Start the HSQL database server.~~
* ~~Start the HSQL database client.~~

### ~~Q2.1 User class~~

* ~~Create a User class with properties id, name and address.~~
* ~~(Remember a property is an attribute with getter and setter methods. )~~
* ~~Generate a constructor with parameters name and address (not id).~~

### Q2.2 UserDAO enum

This should get a connection to the database and save user objects to the database.

* ~~Create the new enum UserDAO as per DAO slides~~
* ~~Create a new methodRe in the UserDAO called getConnection() that returns the JDBC Connection object.~~ ~~This will be used by all your save(), list() update() delete() methods.~~
* ~~Create a new method save(User~~ **~~newUser~~**~~). This should get a connection to the database and INSERT a new user with the name and address of the User~~ **~~newUser.~~**
* ~~Create a new method List<User> list(). This should select all users from the database and create a new User object for each row in the returned ResultSet.~~
* Write simple test case to test your DAO:
  + Call the list method check the number of users.
  + Then call the save with a new User class.
  + Then recall the list method and check the new user has been added to the database.

### Q2.3. index.jsp

Should contain a form to enter name and address. POST form data to CreateUserServlet.

### Q2.4. CreateUserServlet

Create a Servlet called CreateUserServlet. It should implement doPost() which

* gets the name and address parameters
* creates a User object
* calls save(u) on the UserDAO [UserDAO.instance.save(u)]
* Forward the User object to showUser.jsp

### Q2.5. showUser.jsp

Prints out the name and address of the User.

Remember you will need to

* Bring in the two JSTL jar files into the project/lib folder and into the project/WebContent/WEB-INF/lib folder.
* Add the JAR files in the project/lib folder to the build path.
* Have the taglib directive line in showUser.jsp (<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>)