DT228-3 Software Engineering III - Lab 4 (Week 5)

Setup / Exercises for the DAO Pattern using *Eclipse*

In this lab sheet, we will use Eclipse both as a *database development tool* and also as a *java code development tool*. We will set up a test MySQL database that we will use to implement the DAO pattern discussed in class.

1 Create a project

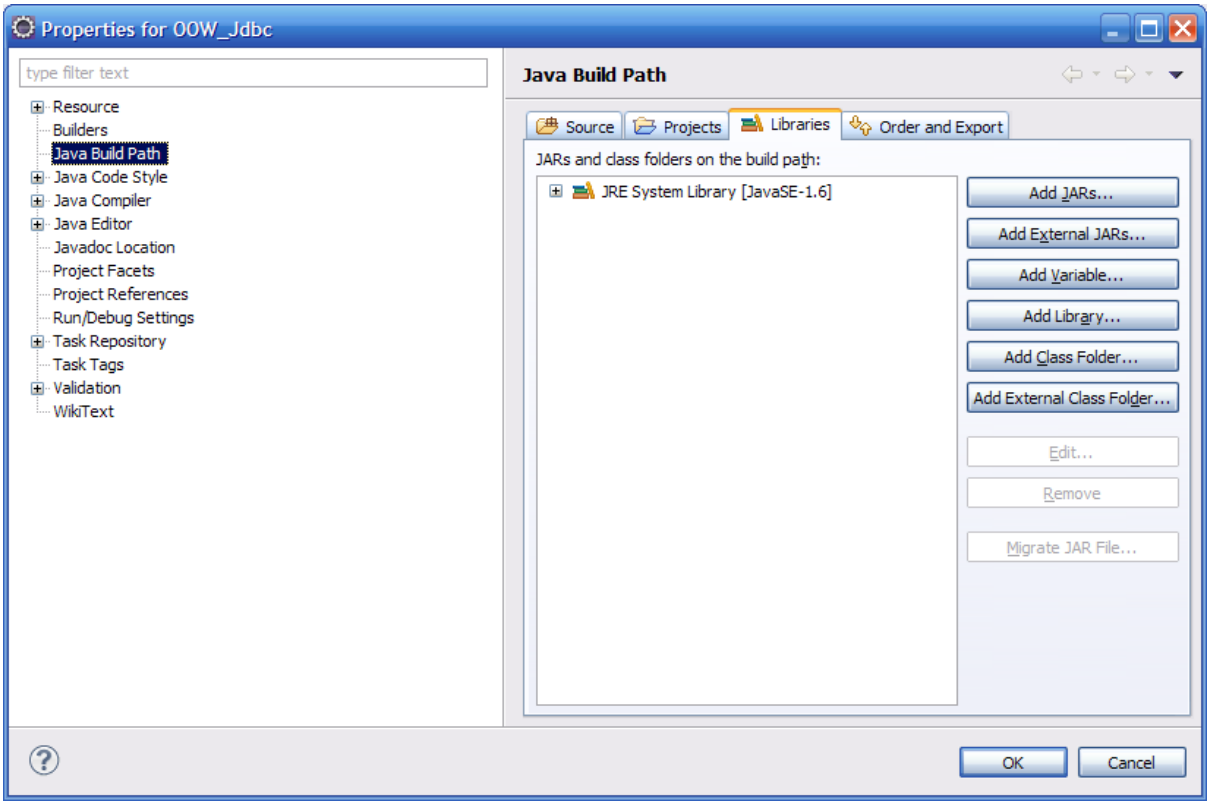
- Start Eclipse (ensure it is Eclipse ***Java Enterprise Edition*** and not RSA or a different Eclipse distribution package).
- Close the welcome tab if that is what you see.
- Create a new *Java Project* in ECLIPSE called *SE3_JDBC_Example (File->New-> Project…->Java->Java Project)*.

2 Download the MySQL JDBC driver jar file

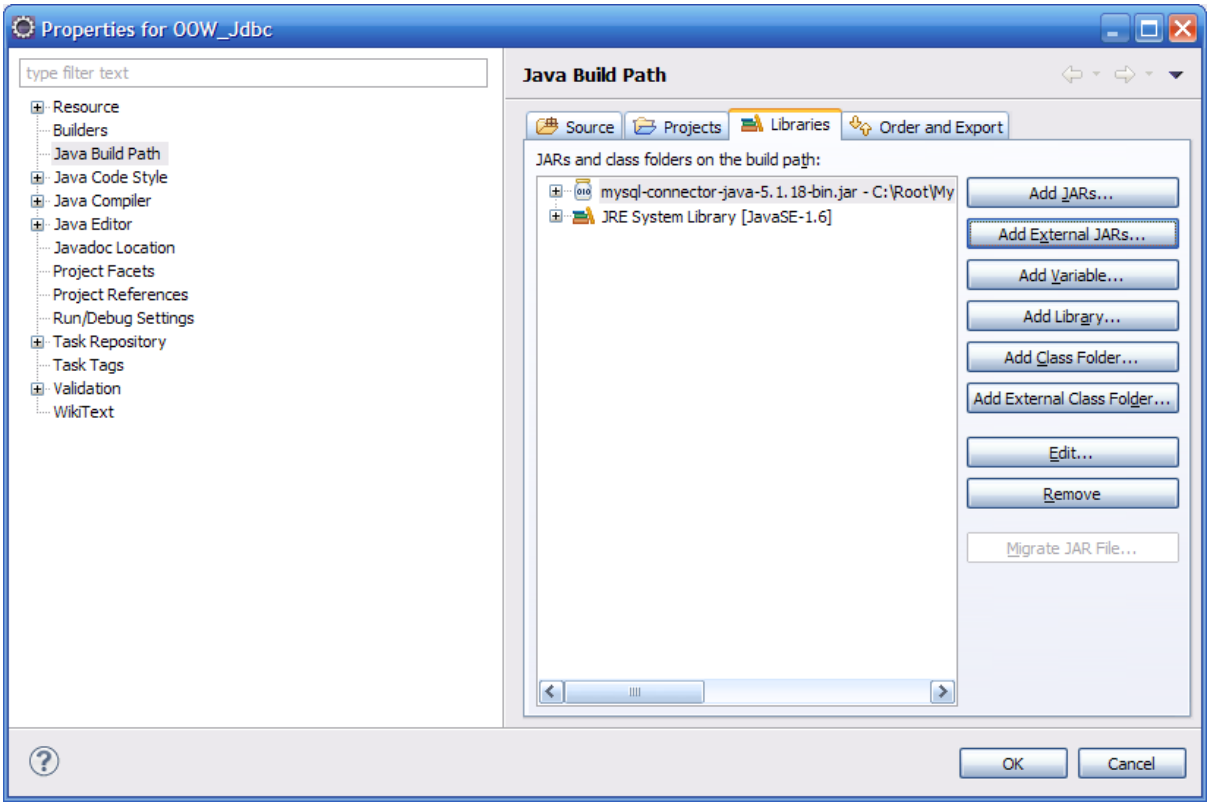
- Download the *mysql-connector-java-5.1.44-bin.jar* file from the Lab Materials folder on webcourses and place in your new project folder (you should have a *SE3_JDBC_Example* sub folder in your eclipse workspace folder).

3 Add the MySQL JDBC driver to your project Build Path as follows

- Select the new project in ECLIPSE. From the menu bar go to *Project -> Properties* and select *Java Build Path* on the left. Select the *Libraries* tab on the right – you should see something like the screenshot below.



- Click *Add External JARs…* and navigate to the location where you saved the MySQL JDBC jar file. Choose the jar file and click OK.
- You should now see the jar file on the right under the *Libraries* tab as below



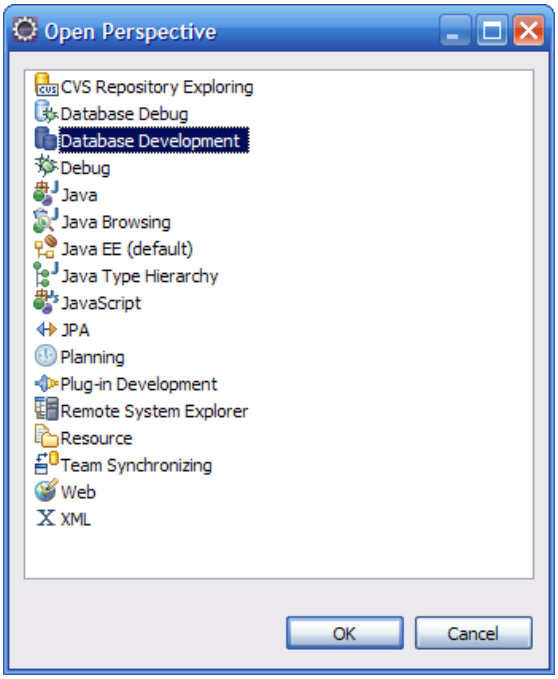
4 Start the MySQL Database Server

Note: *xampp may be installed in a different folder – ask for assistance if you cannot find where it is installed.*

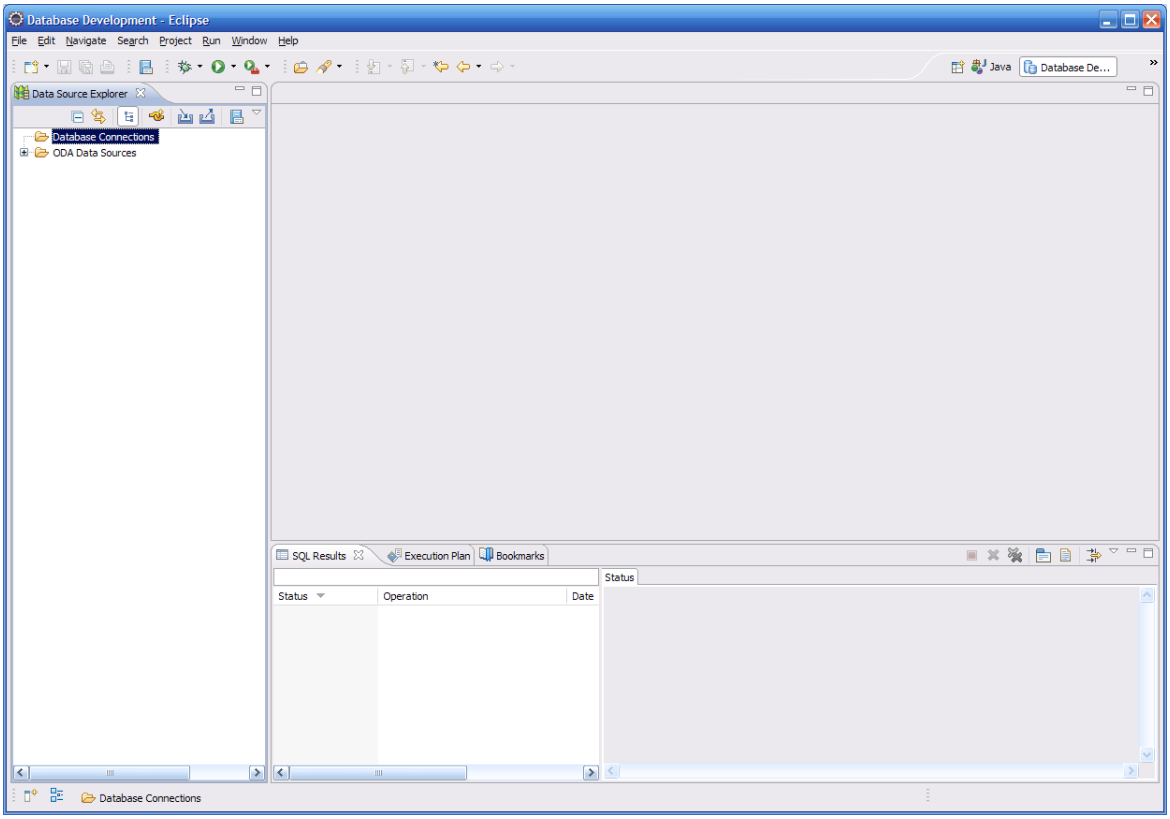
- Go to c:\xampp\mysql\bin and run *mysqld*

5 Use ECLIPSE to set up your sample database

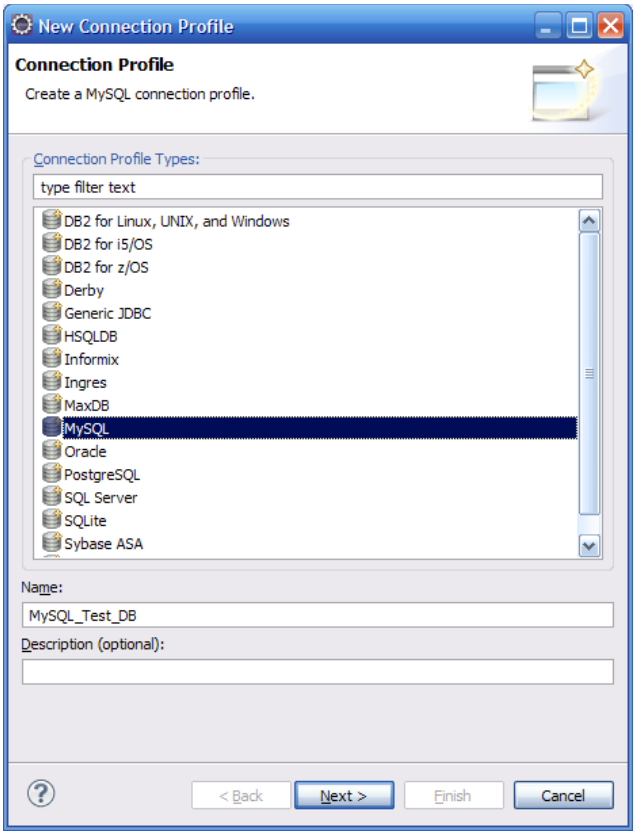
- In ECLIPSE, switch to the *Database Development* perspective as follows:
- From the menu bar go to *Window -> Open Perspective -> Other*, you should the screen below (if you do not see the *Database Development* perpsective check the eclipse version and ensure it is a *Java Enterprise Edition*):



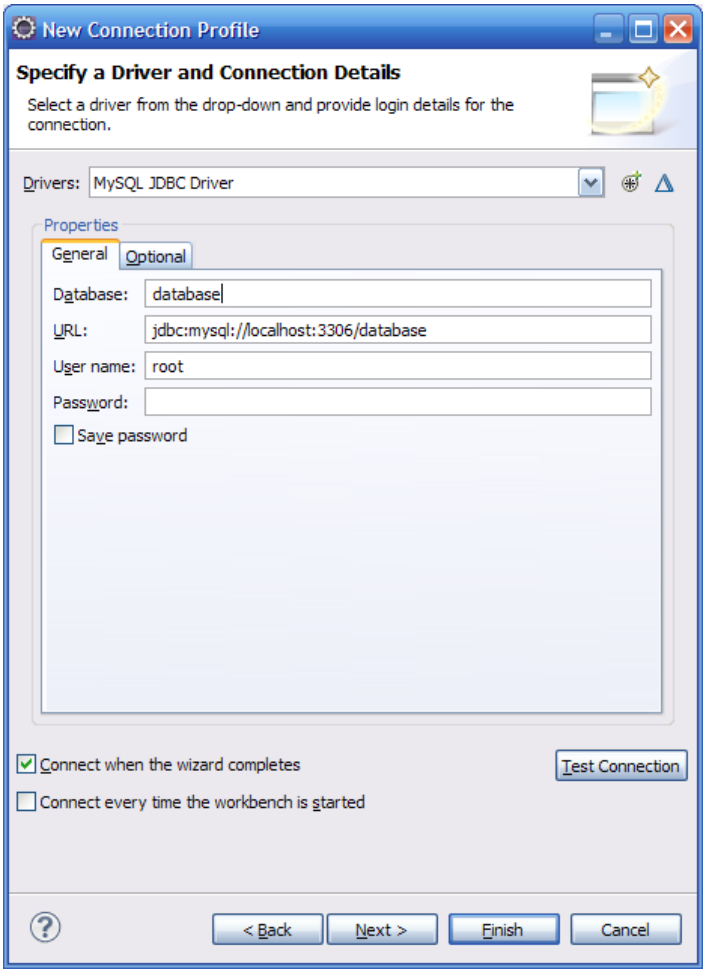
- Choose the *Database Development* perspective. You should see the window below:



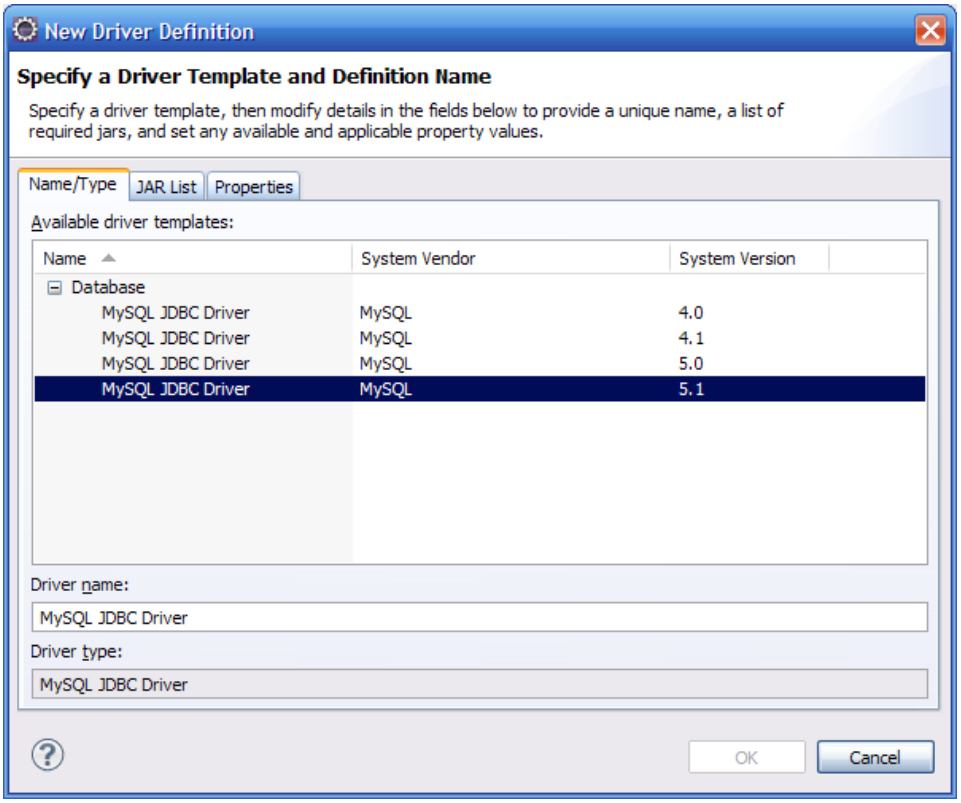
- Right-click on *Database Connections* and select *New*, you should see the screen below:



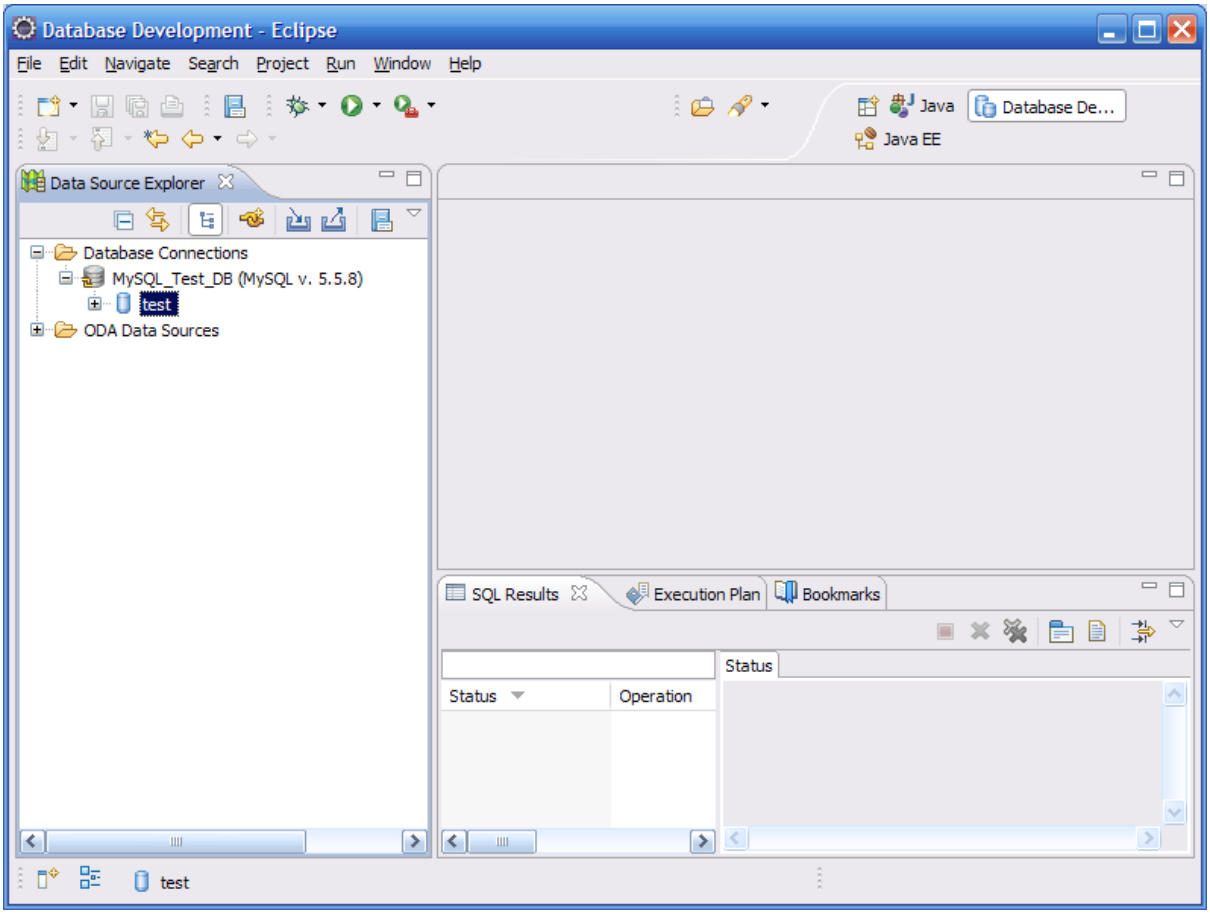
- Select *MySQL* and enter the name *MySQL_Test_DB*
- Click *Next,* you should see the screen below:



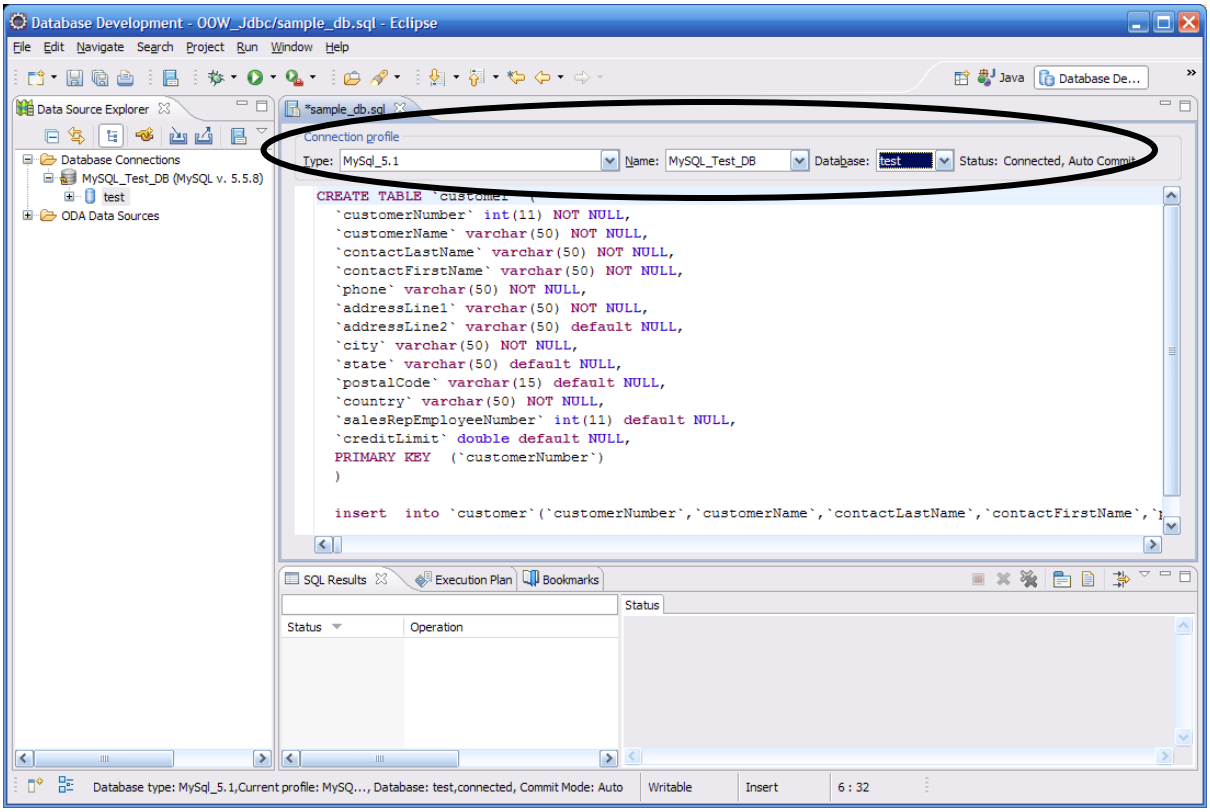
- You will probably have no *Driver* currently defined so click the *New Driver Definition* button to the right of the Drivers dropdown box, you should see the screen below:



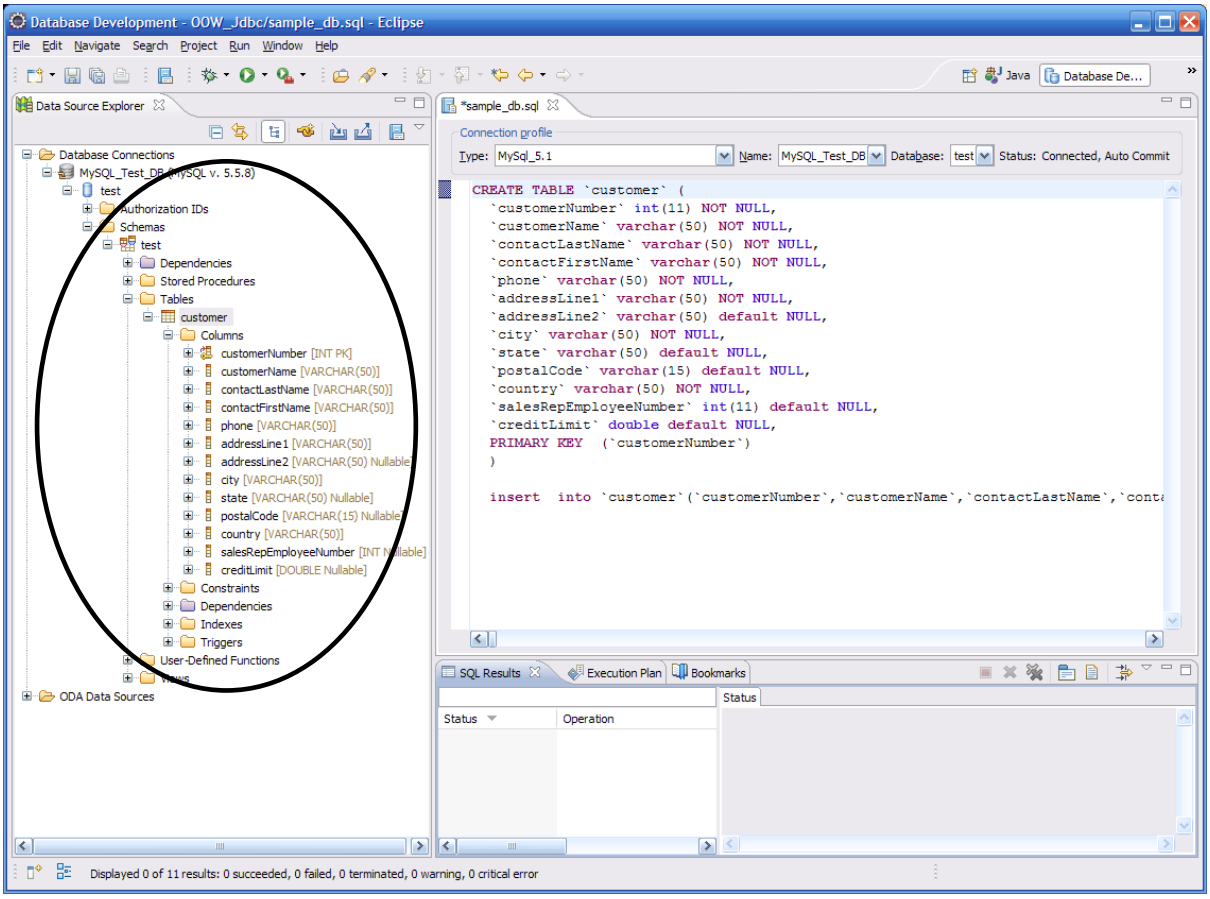
- Select the *MySQL JDBC Driver* in the list of Databases. Select the *JAR List* tab and click the *Add JAR/Zip* button to add the MySQL JDBC jar file you have downloaded from webcourses (see note below). **Note**: if you see an existing *JAR* entry that is showing an error, remove it first. Once you have done the above then click OK.
- In the *General* tab ***change the word "database" to test*** (in both the *Database* field and the *URL* field) and click *Test Connection* to test your connection – you should see a *Ping Succeeded* message.
- Click *Finish*.
- You should now see a database connection called *test* in ECLIPSE as below



- Download the *sample_db.sql* file from webcourses. Place it in your Project folder. We will use this to create a table in the sample *test* database of MySQL as follows:
- From the *Database Development* perspective, choose (from the menu bar) *File -> Open File...*
- Navigate to the *sample_db.sql* file and open it.
- Set the Connection Profile settings as you see below:



- Right-click in the editor and choose *Execute All*
- You should now be able to explore the new *customer* table in the *test* database using the *Data Source Explorer* to the left of the window as below.



- Open new SQL file *(File->New->SQL File)* and test out some SQL queries…

6 Test out the example JDBC connectivity code

- Download the example *JDBCExample.java* source code from webcourses. Create a class in your ECLIPSE java project with the same name *(right-click on the 'src' folder and choose New -> Class)*. Use the code from the example in your new class.
- Test out running the code from ECLIPSE. Right-click in the editor and choose *Run As -> Java Application*

Exercises

1. Write a java class called *Customer* which has a class attribute for every column in the CUSTOMER table. Write a getter & setter for each attribute (**tip:** after you have declared the attributes at the top of the class, right-click in your editor and choose *Source -> Generate Getters and Setters...*).

2. Write a java interface called *CustomerDao (right-click on the 'src' folder and choose New -> Interface)*. This java interface should specify the following methods (note the use of the *Customer* class):
   a. `public Vector<Customer> selectCustomersByName(String name);`
   b. `public Customer findCustomerById(int customerNumber);`

3. Write a java class called *MySqlCustomerDao* which implements the *CustomerDao* interface that you have written. This DAO class should also implement the following method:

```
private Connection getConnection()
```

This method should be invoked by the two database access methods to get the connection to the database that they will need to execute their SQL.

4. Write a client class called *JDBCExercise* which contains a single static *main* method. Write the code to instantiate a *CustomerDao* object. Use the *CustomerDao* object to:
    a. find and display the customer with customer number *347*
    b. find and display all customers whose name contain the word *"Mini"*

If you have time:

5. Add a method to your *MySqlCustomerDao* class called *updateCustomer*. This method should allow the *creditLimit* of a customer to be updated. Modify your *JDBCExercise* client to update the *creditLimit* on customer 347 to 60000.