DT228 3 Software Engineering III Lab 6 (Week 8)

Eclipse Setup for a Dynamic Web Project

Part I – Using Eclipse as a Database Development Tool

1 Set up a Folder Structure

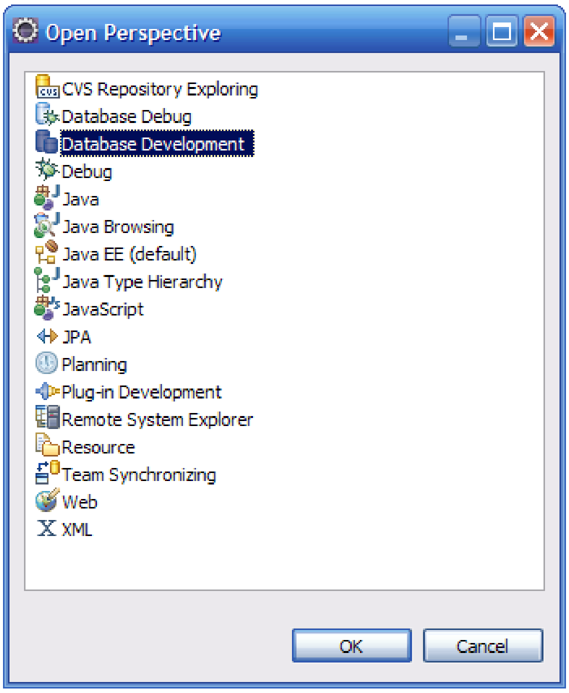
- On your network drive (or laptop), create a folder for your Eclipse lab work (e.g. *SE3\Java*).
- Within that folder create a subfolder called *Libs* to hold any libraries (jar files) that we need.
- Create another subfolder of *SE3\Java* called *Sql* to hold any sql scripts that we need.
- Create another sub folder of *SE3\Java* called *Workspace* to hold our Eclipse projects.

2 Download the MySQL JDBC driver jar file

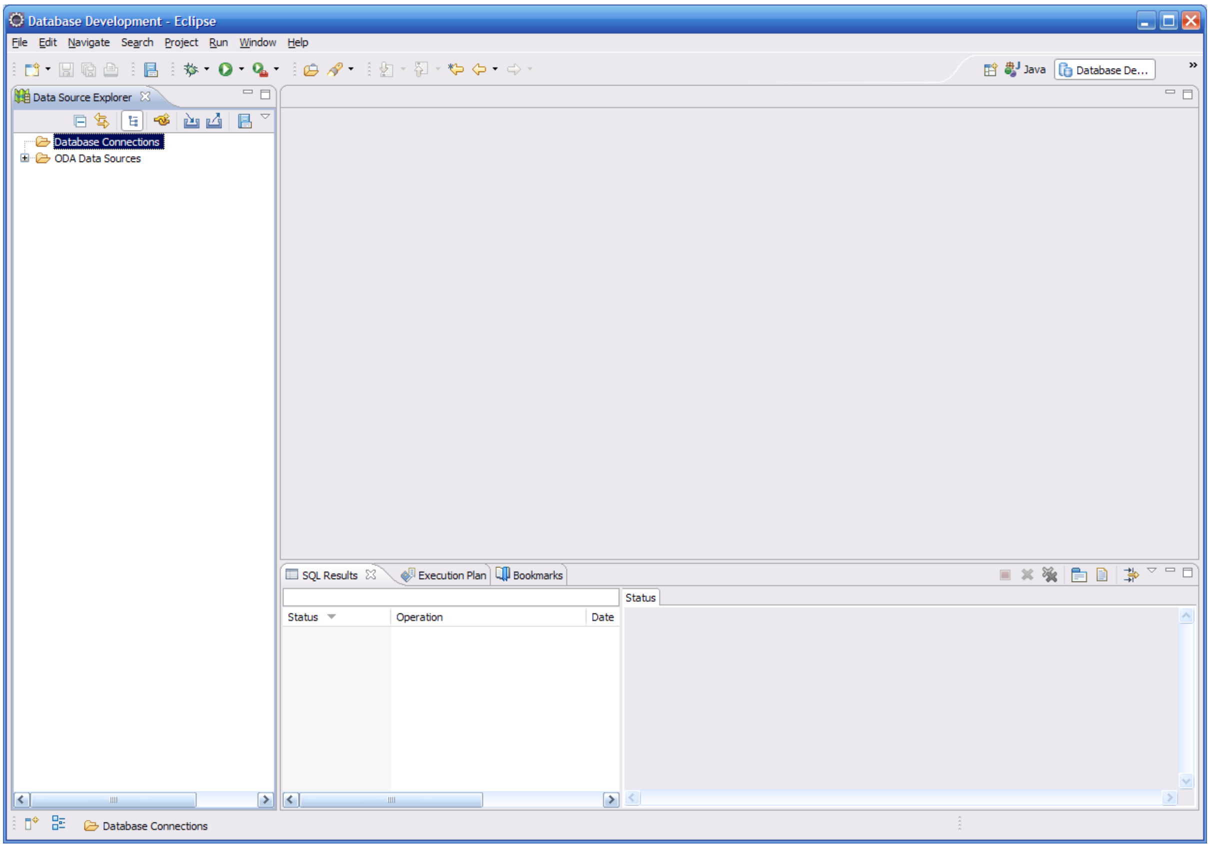
- We will be using the MySQL database that is distributed with the xampp software distribution package on the lab machines.
- To connect to this database from both the Eclipse IDE and our Java code we will need the JDBC driver for the MySQL database - download the *mysql-connector-java-5.1.18-bin.jar* file from the *Lab Materials* folder on webcourses and place it in your *Libs* folder.

3 Using Eclipse to Set Up your Sample Database

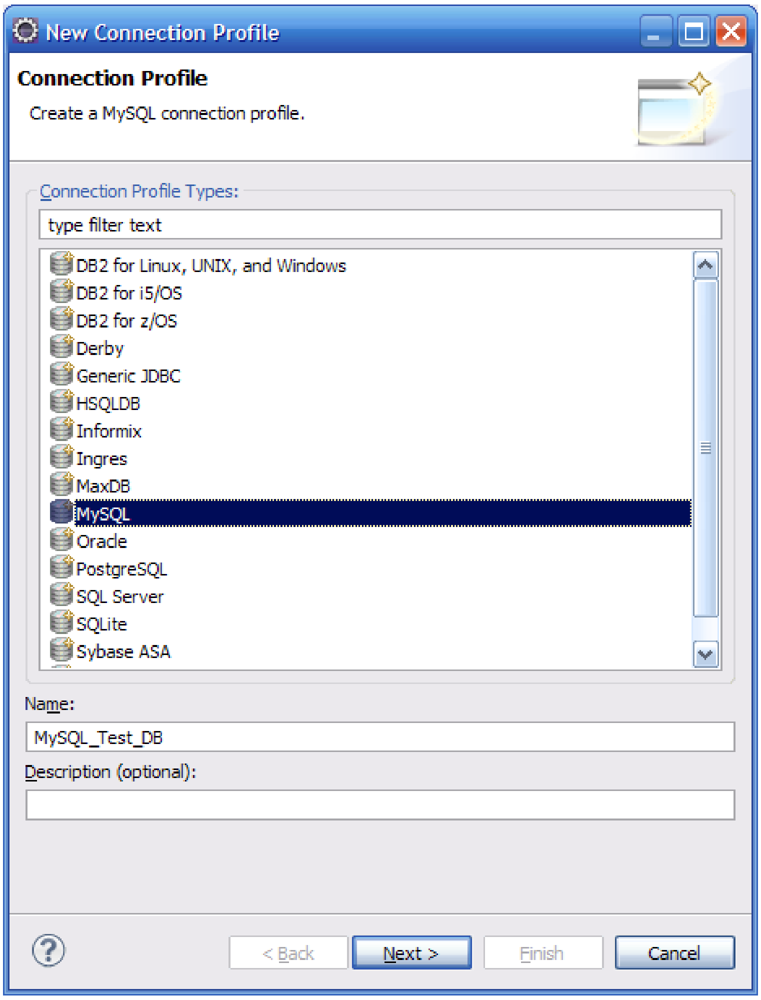
- Start Eclipse (ensure it is the Java Enterprise Edition – *Kepler* release edition on lab machines). Choose the workspace folder you created earlier when starting Eclipse (e.g. SE3\Java\Workspace
- Close the welcome tab if that is what you see.
- In Eclipse, switch to the *Database Development* perspective as follows:
- From the menu bar go to *Window -> Open Perspective -> Other*, you should the screen below:



- 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



Connection Profile

Create a MySQL connection profile.

Connection Profile Types:

type filter text

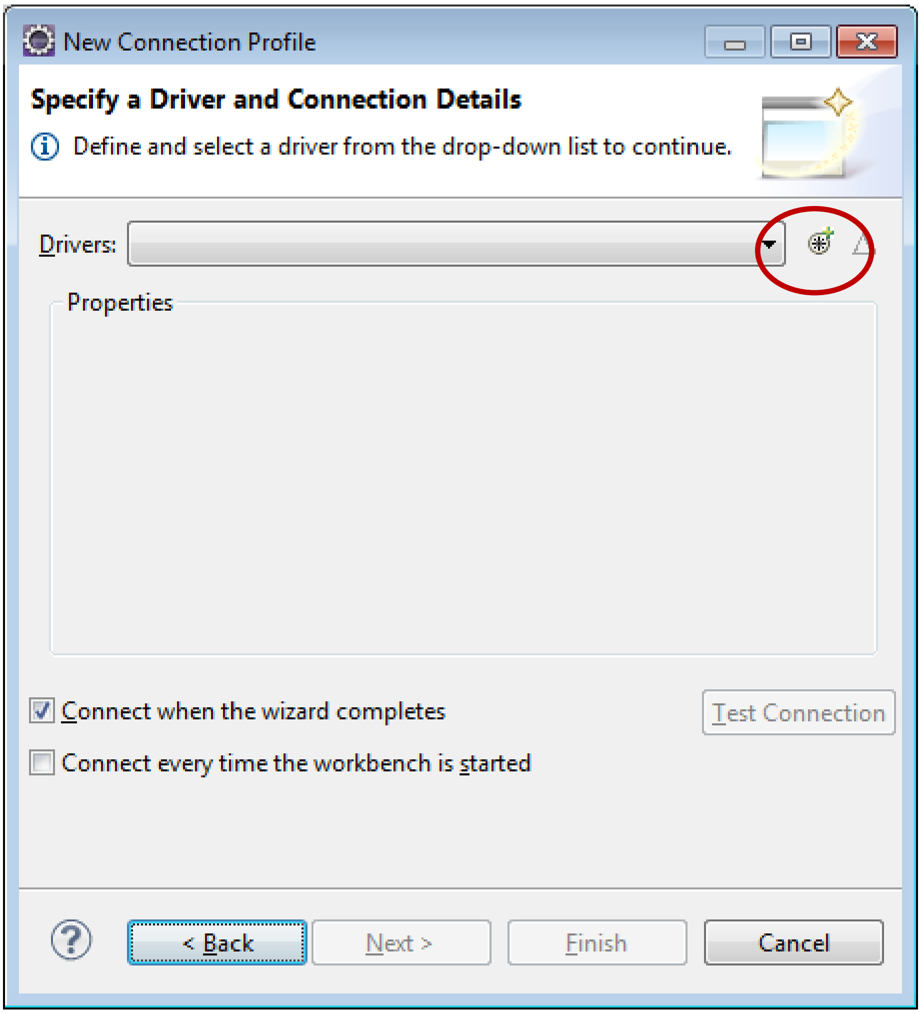
- DB2 for Linux, UNIX, and Windows
- DB2 for i5/OS
- DB2 for z/OS
- Derby
- Generic JDBC
- HSQLDB
- Informix
- Ingres
- MaxDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server
- SQLite
- Sybase ASA

Name:

MySQL_Test_DB

Description (optional):

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



New Connection Profile

Specify a Driver and Connection Details

Define and select a driver from the drop-down list to continue.

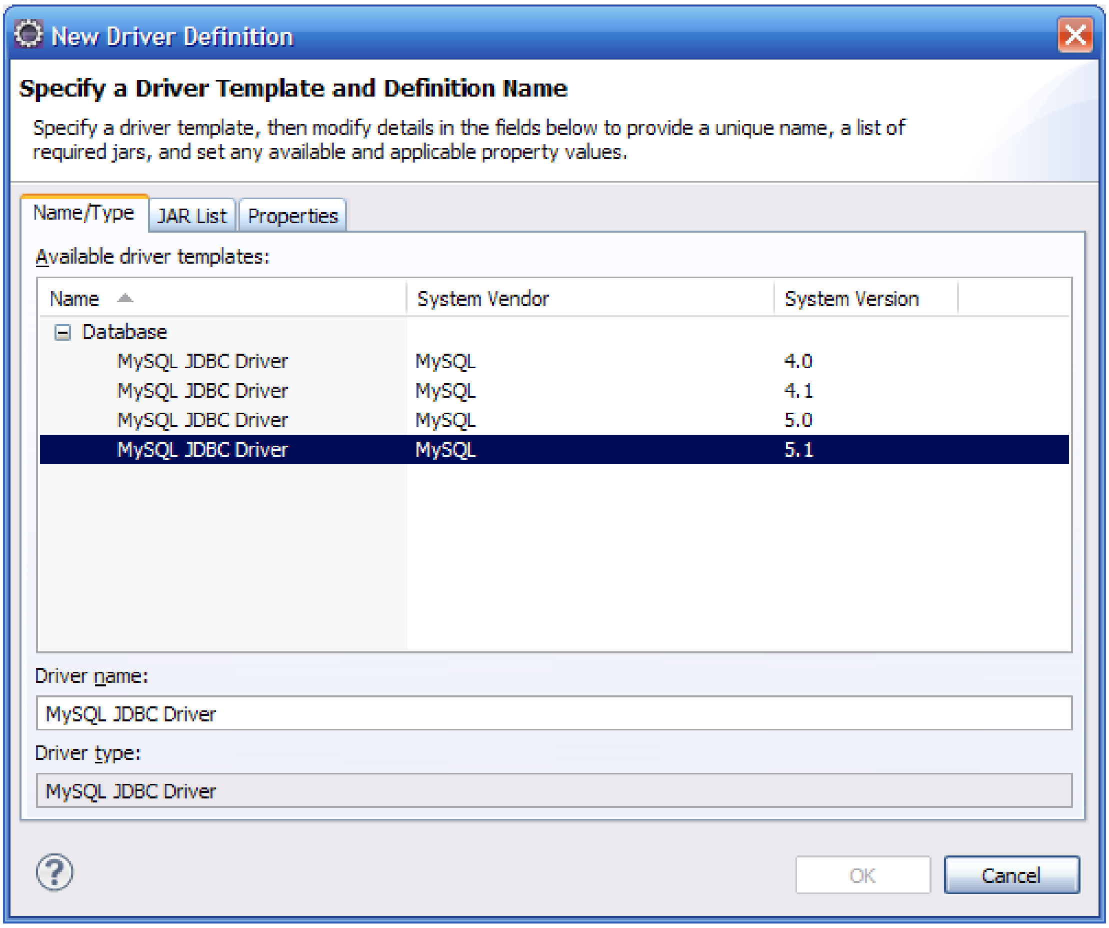
Drivers:

Properties

☑ Connect when the wizard completes

□ Connect every time the workbench is started

- 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:



New Driver Definition

Specify a Driver Template and Definition Name

Specify a driver template, then modify details in the fields below to provide a unique name, a list of required jars, and set any available and applicable property values.

Available driver templates:

Name	System Vendor	System Version
⊟ Database		
MySQL JDBC Driver	MySQL	4.0
MySQL JDBC Driver	MySQL	4.1
MySQL JDBC Driver	MySQL	5.0
MySQL JDBC Driver	MySQL	5.1

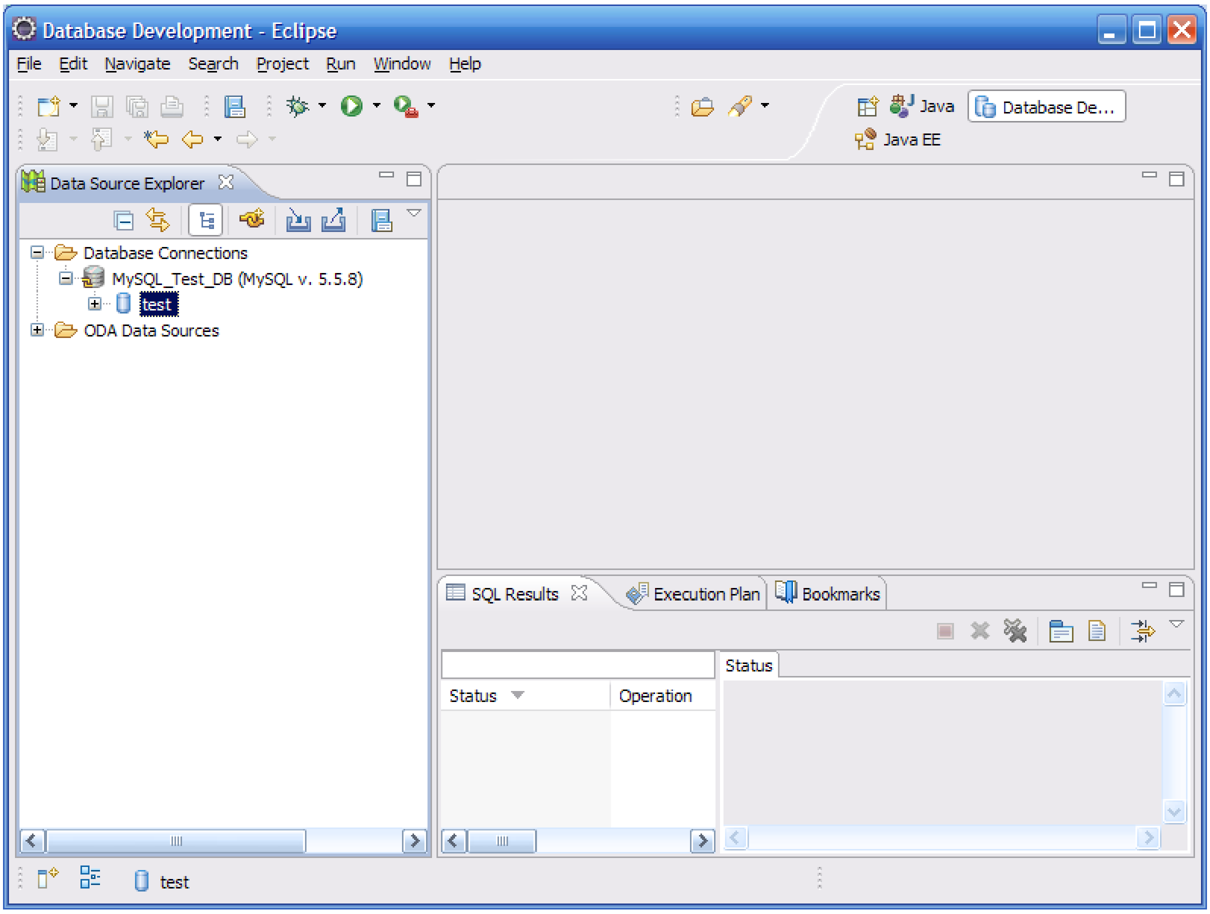
Driver name: MySQL JDBC Driver

Driver type: MySQL JDBC Driver

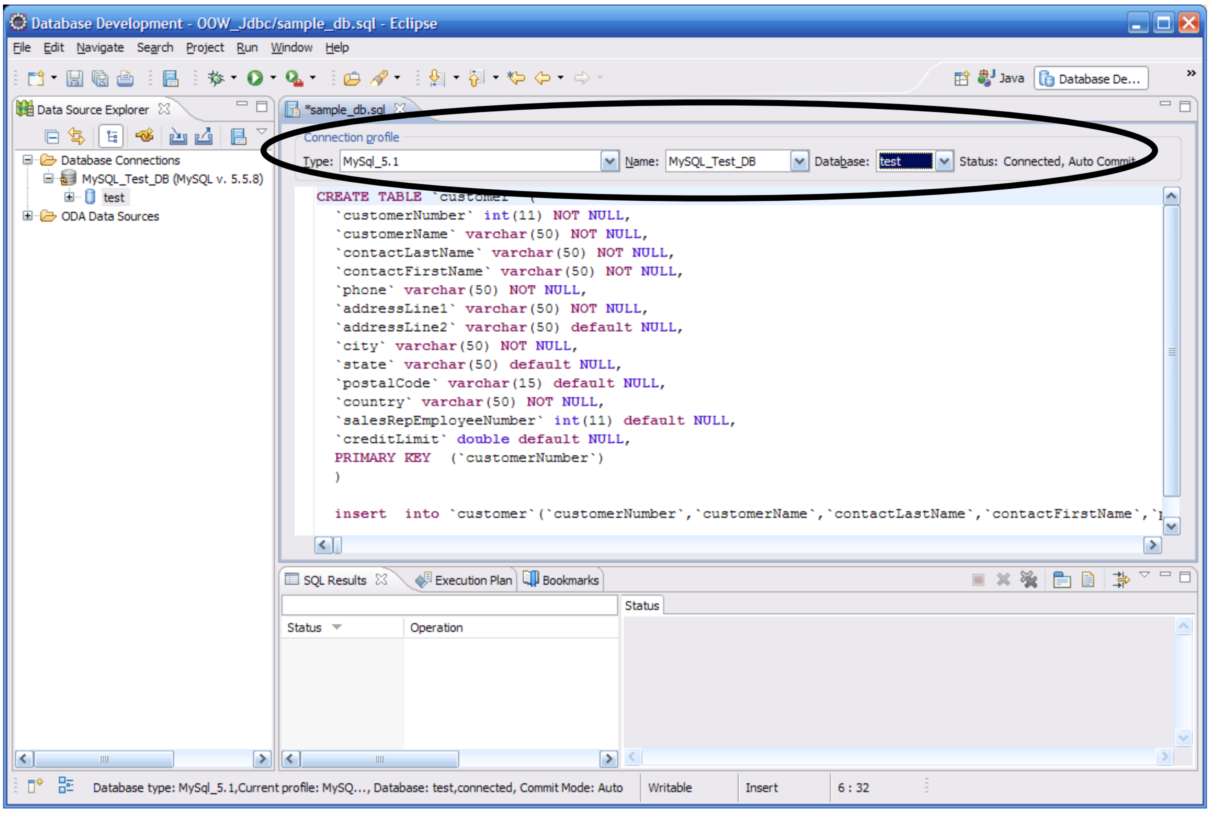
- Select the *MySQL JDBC Driver* in the list of Databases. Select the *JAR List* tab and click the *Add JAR/Zip* button to add the MySQL JDBC jar file you have downloaded from webcourses (see note below). Once you have done this then click OK.

  Note: if you see an existing *JAR* entry that is showing an error, remove it first.

- In the *General* tab ***change the word "database" to "test"*** in both the *Database* field and the *URL* field.
- At this point we need to start the MySQL database – open a Command Prompt and go to xampp\mysql\bin and run the *mysqld* executable.
- From Eclipse, click the *Test Connection* button – you should get the following message – *"Ping Succeeded!"*.
- 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 *sql* folder created earlier. You can 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:



Connection profile

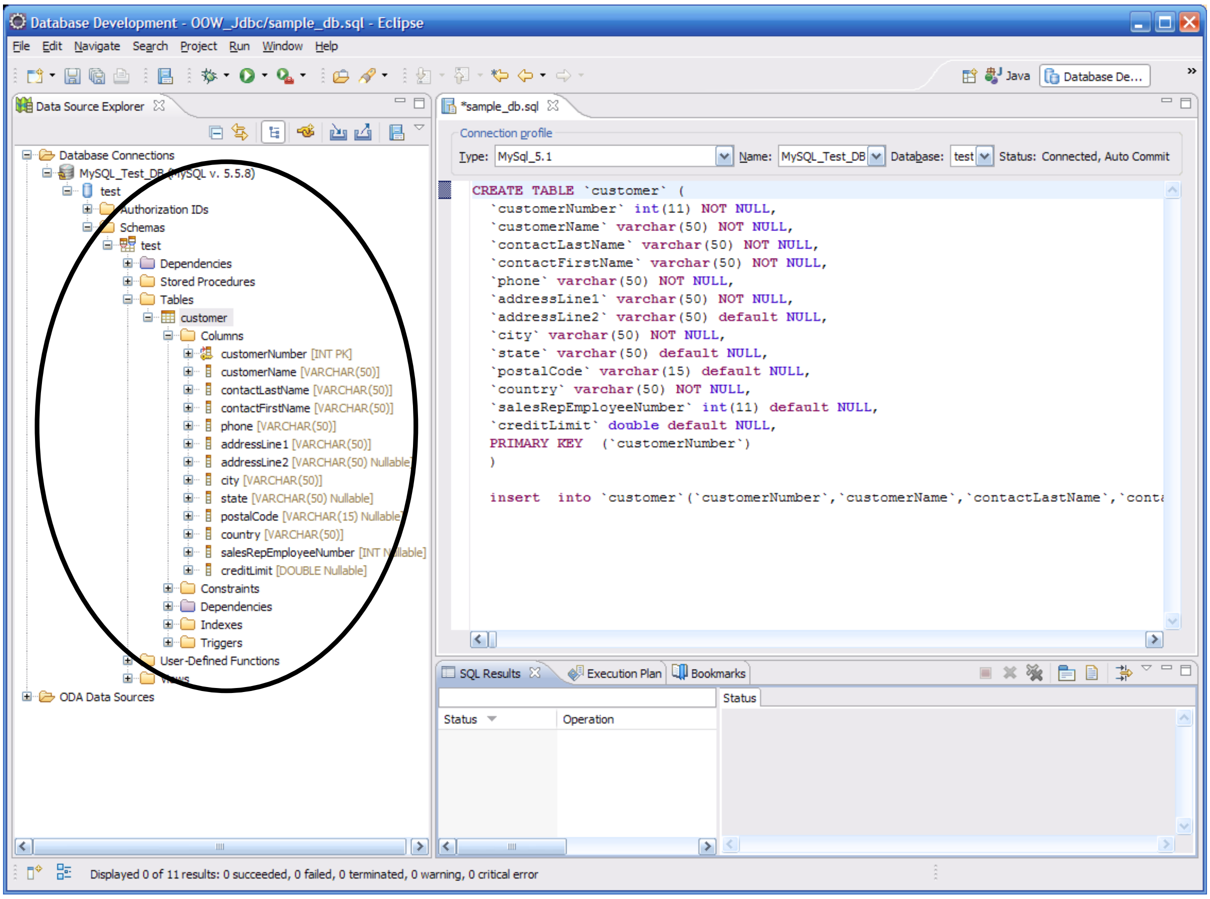
Type: MySql_5.1 Name: MySQL_Test_DB Database: test Status: Connected, Auto Commit

```
CREATE TABLE `customer` (
  `customerNumber` int(11) NOT NULL,
  `customerName` varchar(50) NOT NULL,
  `contactLastName` varchar(50) NOT NULL,
  `contactFirstName` varchar(50) NOT NULL,
  `phone` varchar(50) NOT NULL,
  `addressLine1` varchar(50) NOT NULL,
  `addressLine2` varchar(50) default NULL,
  `city` varchar(50) NOT NULL,
  `state` varchar(50) default NULL,
  `postalCode` varchar(15) default NULL,
  `country` varchar(50) NOT NULL,
  `salesRepEmployeeNumber` int(11) default NULL,
  `creditLimit` double default NULL,
  PRIMARY KEY  (`customerNumber`)
  )

  insert  into `customer`(`customerNumber`,`customerName`,`contactLastName`,`contactFirstName`,`p
```

Database type: MySql_5.1,Current profile: MySQ..., Database: test,connected, Commit Mode: Auto | Writable | Insert | 6 : 32

- 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.


Database Development - OOW_Jdbc/sample_db.sql - Eclipse
File Edit Navigate Search Project Run Window Help
Java
Database De...
Data Source Explorer
*sample_db.sql
SQL Results
Execution Plan
Bookmarks

```
Database Connections
  MySQL_Test_DB (MySQL v. 5.5.8)
    test
      Authorization IDs
      Schemas
        test
          Dependencies
          Stored Procedures
          Tables
            customer
              Columns
                customerNumber [INT PK]
                customerName [VARCHAR(50)]
                contactLastName [VARCHAR(50)]
                contactFirstName [VARCHAR(50)]
                phone [VARCHAR(50)]
                addressLine1 [VARCHAR(50)]
                addressLine2 [VARCHAR(50) Nullable]
                city [VARCHAR(50)]
                state [VARCHAR(50) Nullable]
                postalCode [VARCHAR(15) Nullable]
                country [VARCHAR(50)]
                salesRepEmployeeNumber [INT Nullable]
                creditLimit [DOUBLE Nullable]
              Constraints
              Dependencies
              Indexes
              Triggers
          User-Defined Functions
          Views
ODA Data Sources
```

Connection profile

Type: MySql_5.1 Name: MySQL_Test_DB Database: test Status: Connected, Auto Commit

```
CREATE TABLE `customer` (
  `customerNumber` int(11) NOT NULL,
  `customerName` varchar(50) NOT NULL,
  `contactLastName` varchar(50) NOT NULL,
  `contactFirstName` varchar(50) NOT NULL,
  `phone` varchar(50) NOT NULL,
  `addressLine1` varchar(50) NOT NULL,
  `addressLine2` varchar(50) default NULL,
  `city` varchar(50) NOT NULL,
  `state` varchar(50) default NULL,
  `postalCode` varchar(15) default NULL,
  `country` varchar(50) NOT NULL,
  `salesRepEmployeeNumber` int(11) default NULL,
  `creditLimit` double default NULL,
  PRIMARY KEY  (`customerNumber`)
  )

  insert  into `customer`(`customerNumber`,`customerName`,`contactLastName`,`conta
```

Status	Operation

Status

Displayed 0 of 11 results: 0 succeeded, 0 failed, 0 terminated, 0 warning, 0 critical error

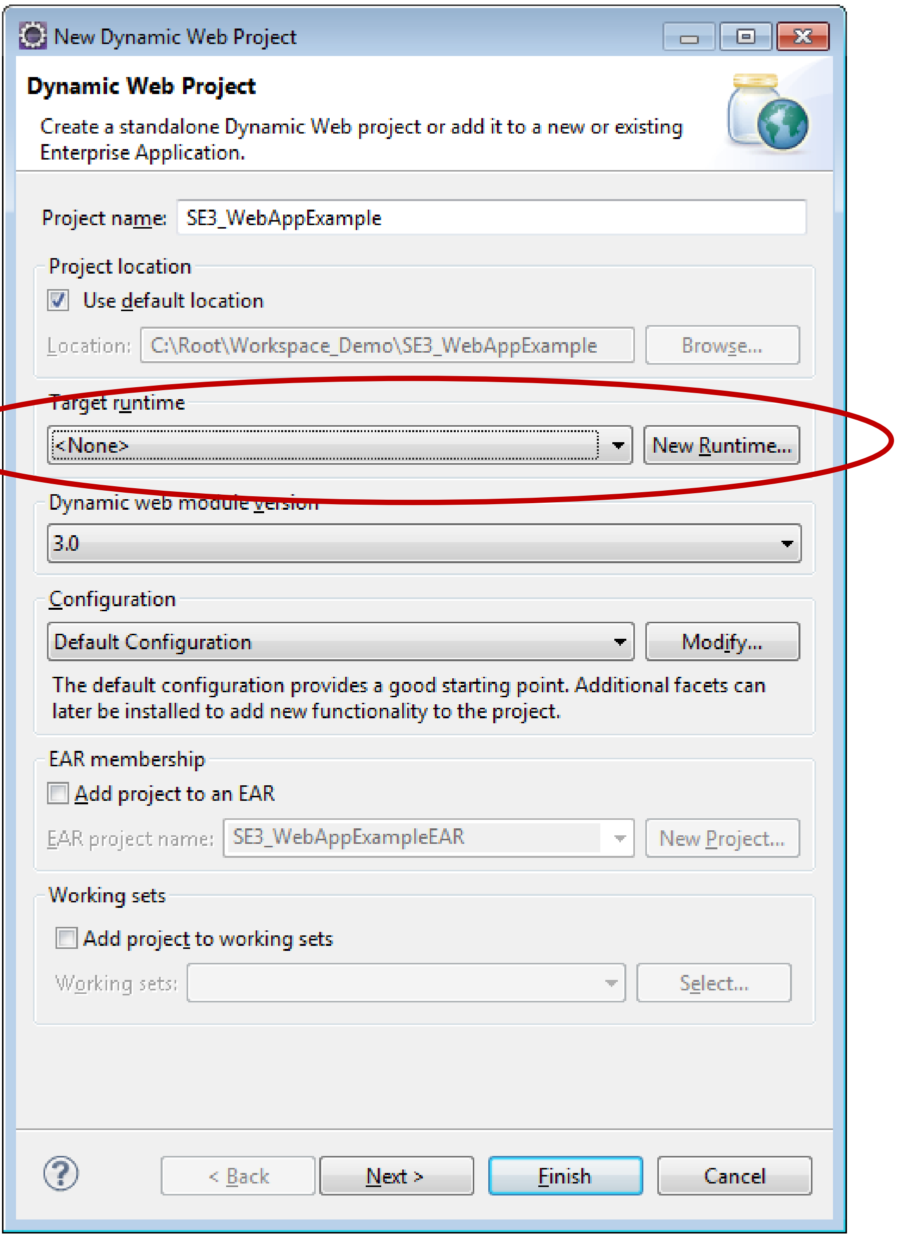
- Open new SQL file *(File->New->SQL File)* and test out some SQL queries…

Note: So far we have just been using Eclipse as a database development tool which is very much separated from what we will do next which is to develop a Dynamic Web Application.

Part II – Creating a Dynamic Web Project

1 Create the project

- In Eclipse, switch to the Java EE perspective.
- Create a new Dynamic Web Project in Eclipse called *SE3_WebAppExample (File->New->Dynamic Web Project)*. You should see the following screen:



New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: SE3_WebAppExample

Project location

☑ Use default location

Location: C:\Root\Workspace_Demo\SE3_WebAppExample

Target runtime: <None>

Dynamic web module version: 3.0

Configuration: Default Configuration

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

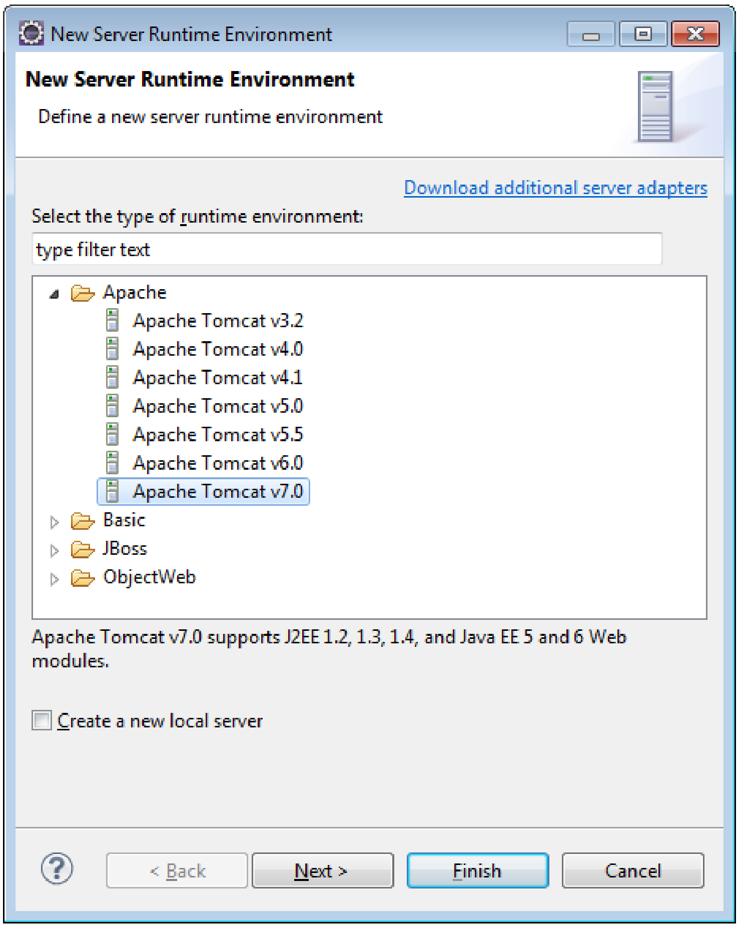
EAR project name: SE3_WebAppExampleEAR

Working sets

☐ Add project to working sets

Working sets:

- If there is no "Apache Tomcat" *Target Runtime,* choose *New Runtime…* you should see the following:


New Server Runtime Environment

New Server Runtime Environment

Define a new server runtime environment

Download additional server adapters

Select the type of runtime environment:

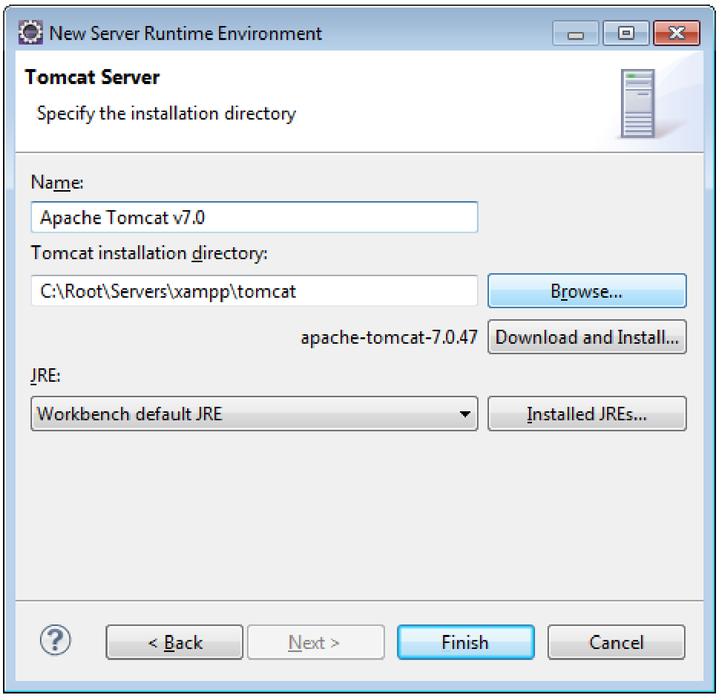
type filter text

- Apache
  - Apache Tomcat v3.2
  - Apache Tomcat v4.0
  - Apache Tomcat v4.1
  - Apache Tomcat v5.0
  - Apache Tomcat v5.5
  - Apache Tomcat v6.0
  - Apache Tomcat v7.0
- Basic
- JBoss
- ObjectWeb

Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.

□ Create a new local server

- Select *Apache Tomcat v7.0* and click *Next>*. You should see the following:


New Server Runtime Environment

Tomcat Server

Specify the installation directory

Name:

Apache Tomcat v7.0

Tomcat installation directory:

C:\Root\Servers\xampp\tomcat

apache-tomcat-7.0.47

JRE:

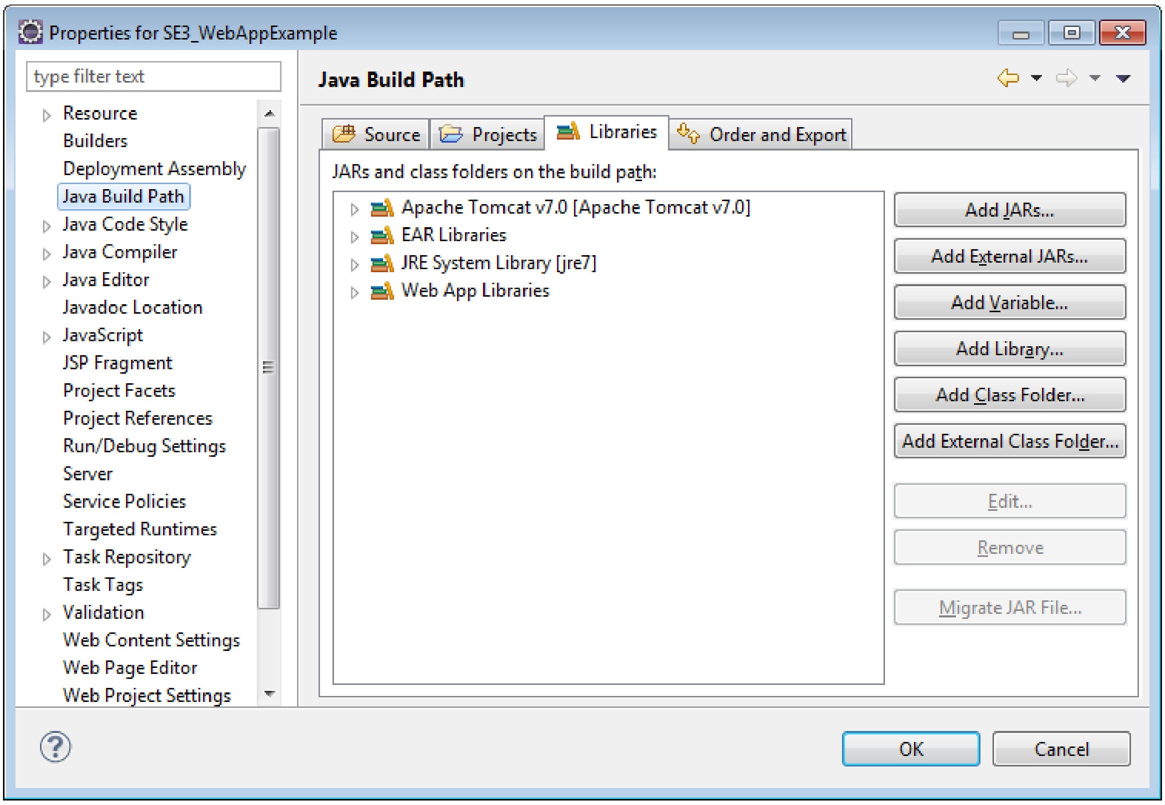
Workbench default JRE

- Using the *Browse* button, navigate to the tomcat folder under the xampp distribution (probably *d:\xampp\tomcat* on the lab machines*)*.
- Click *Finish* to return to previous screen and then click *Finish* again to create the project.

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

Note: Here we are adding the MySQL JDBC implementation jar file to our project classpath so that our Java code can utilise it to access the MySQL database.

- Add a folder to your project called *libs* (right-click on the project in the Project Explorer in Eclipse and choose *New -> Folder*).
- Copy the MySQL JDBC jar file that you downloaded earlier to the new project folder (you can drag the file from windows file explorer into the Eclipse project explorer).
- Click on the project in Eclipse and 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.



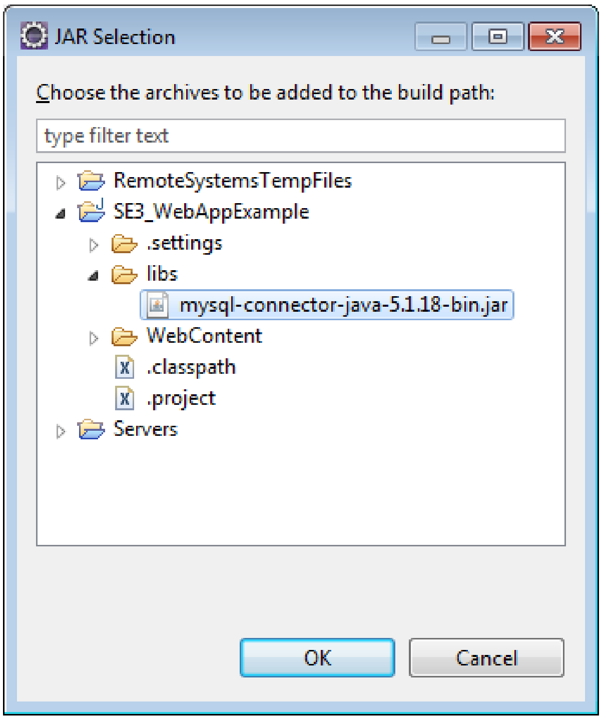
Properties for SE3_WebAppExample

Java Build Path

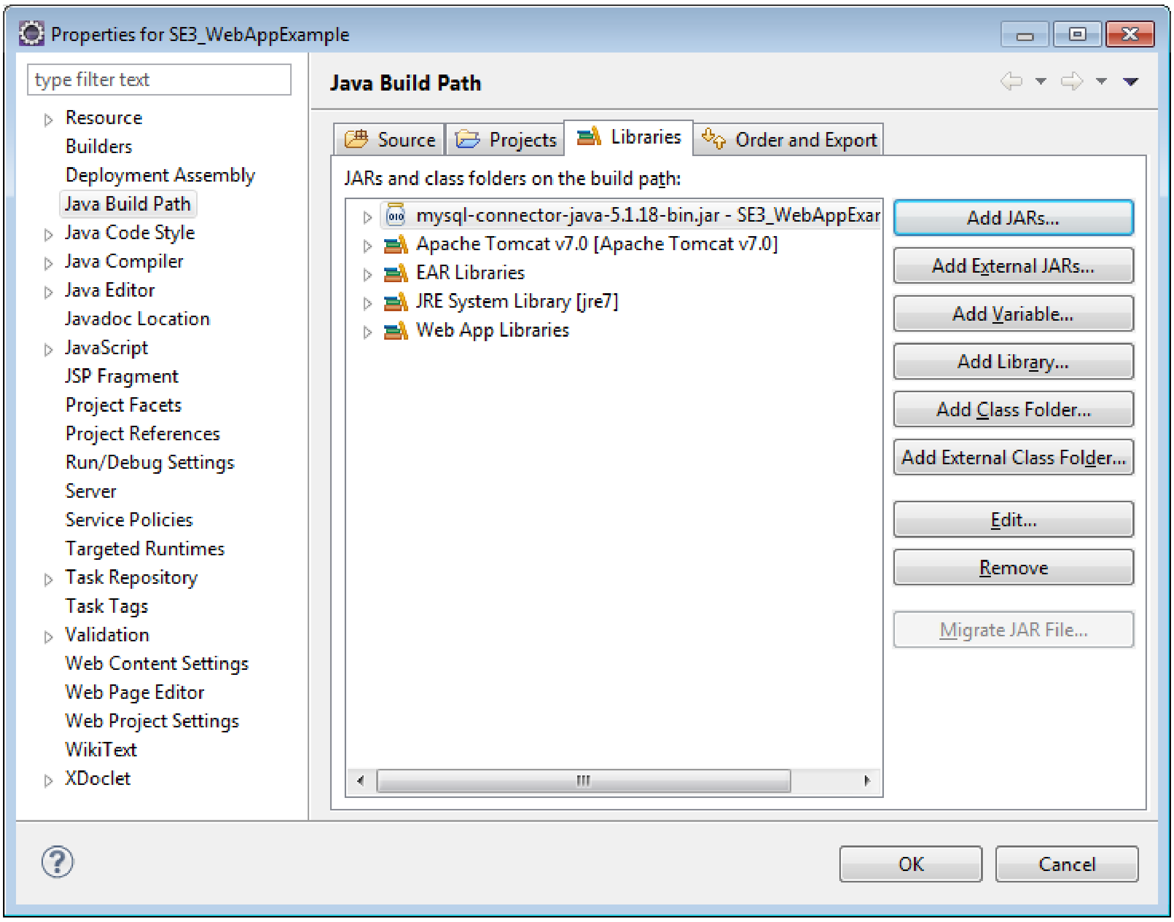
JARs and class folders on the build path:

- Apache Tomcat v7.0 [Apache Tomcat v7.0]
- EAR Libraries
- JRE System Library [jre7]
- Web App Libraries

- Click *Add JARs…* and navigate to the MySQL JDBC jar file as below:

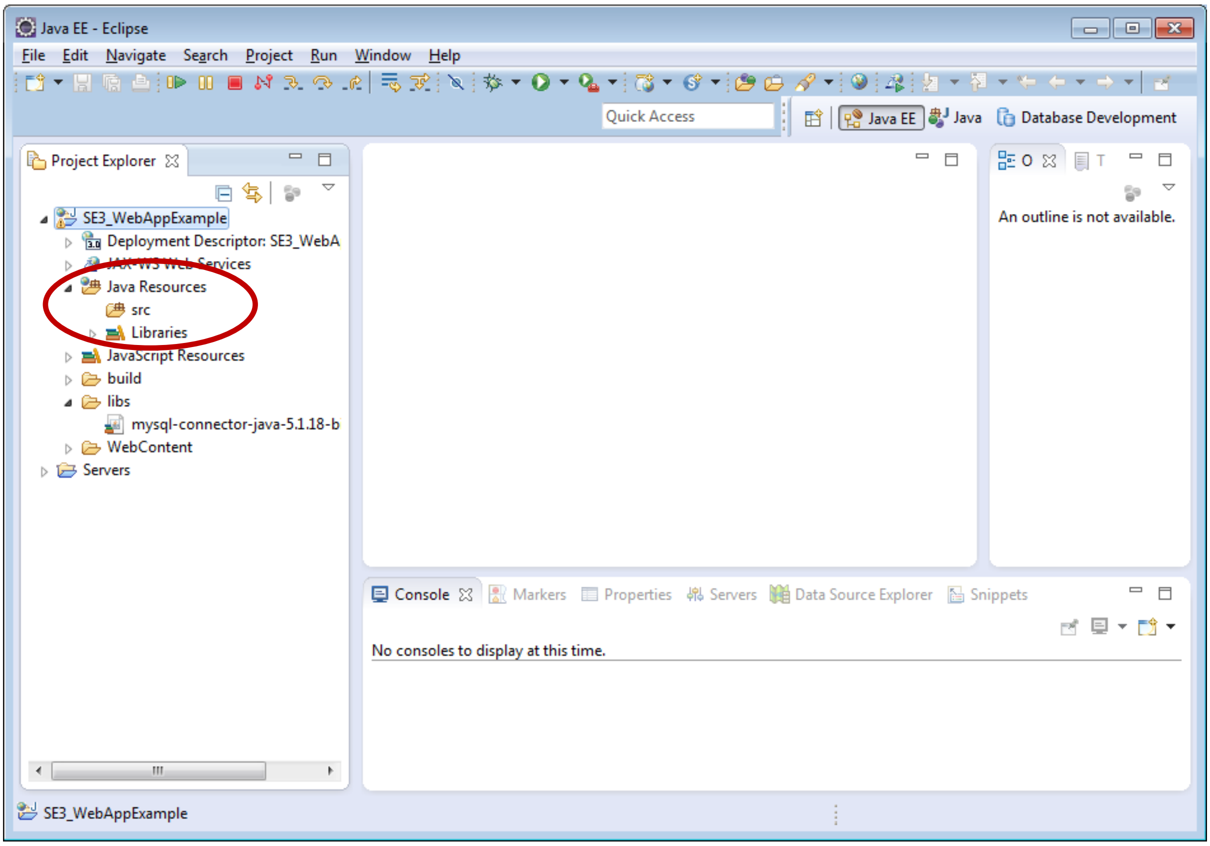


- Choose the jar file and click OK.
- You should now see the jar file on the right under the *Libraries* tab as below.
- Click *OK* to finish.



3 Add a Package Structure

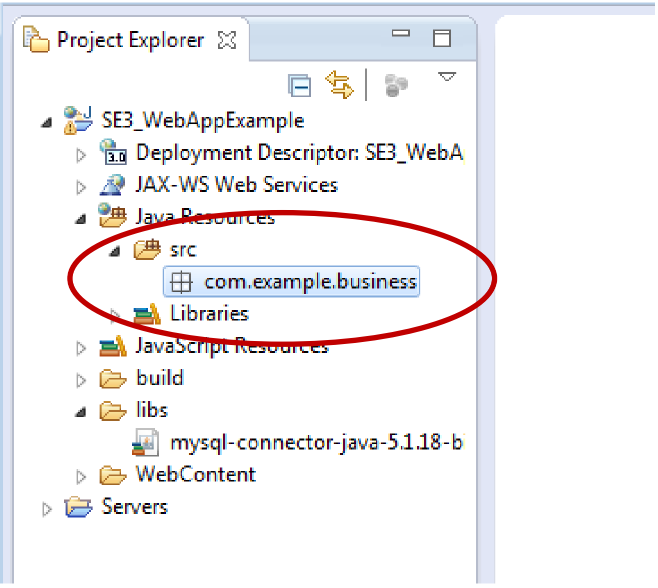
- Expand the Java Resources folder in the Eclipse Project Explorer so that you see the *src* folder that was automatically created – this will hold our java source code:



An outline is not available.

No consoles to display at this time.

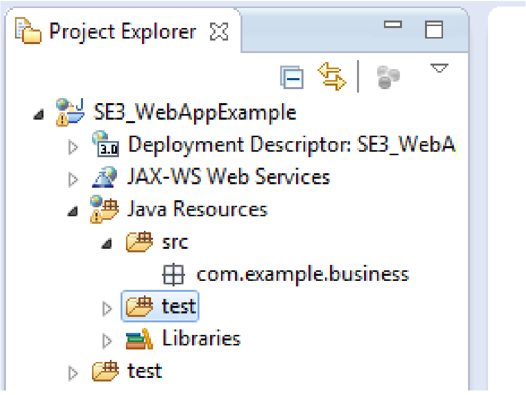
- Right-click on the *src* folder and choose *New -> Package*. Name it *com.example.business*.
- You should see the following:



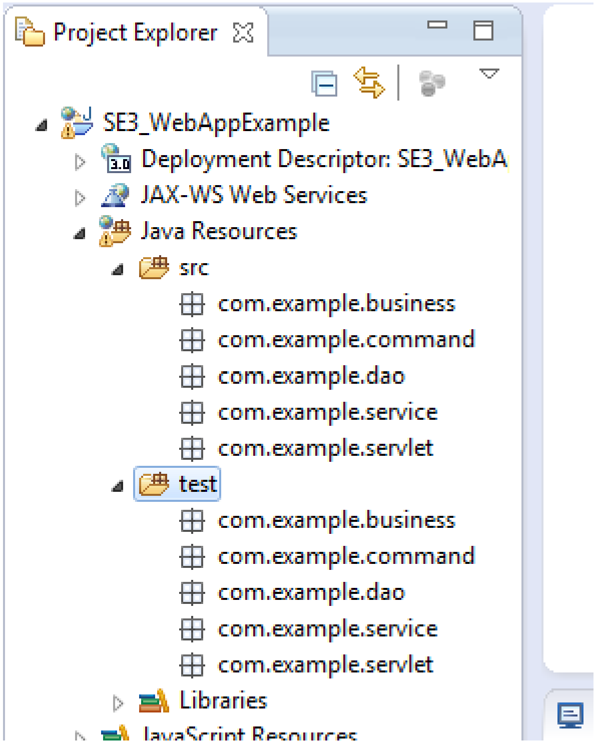
- Also create the following packages:

  - com.example.dao
  - com.example.service
  - com.example.command
  - com.example.servlet

- Create another source folder alongside our current *src* folder (right-click on the project in the Project Explorer and choose *New -> Source Folder* and name it *test* – you should now have the following:



- Replicate the package structure that you created in the *src* folder so you have the following:



4 Test Database Connection

Note: Here we will just execute some java code to connect to the database and retrieve some data – this is not part of our application, just a checkpoint to see our project is set up correctly.

- Download the *JDBCExample.java* source file from webcourses.
- Copy it into the root of the *test* source folder (this will create a default package).
- Open it in the editor and right-click -> Run As -> Java Application.
- Have a look at the output in the console view.