Lesson 3 Practical B – Connecting MySQL with Servlet

- Learn to use MySQL Database
- Learn to use Eclipse for Java EE for web application Development
    - Familiarize with the Development Environment
    - Learn how to import the web application project

- Learn to use Eclipse for Java EE for web application Development
    - Understand Web Application Structure
    - Creating a servlet
    - Retrieving user parameter using servlet
    - Perform user request and generate response

Software Requirements

1. Eclipse IDE(Kepler)
2. XAMPP MYSQL
    (i) MySQL Community Server (5.5.27 / 5.6.x)
    (ii) MySQL Connector J (5.1.26)
    (iii) My SQL Workbench (5.2.42 / 6.0)

3. JDK 7 Update 21
4. Tomcat ver 7.0 /8.0

Using MySQL Database

MySQL database is a popular database used in the Open Source community. It is easy to use and does not take up a lot of computing resources, as such, makes it very suitable to be used for learning purpose. In this section, students shall learn the knowledge on how to manage and use MySQL. For all subsequent workshops, it is assumed that the student knows how to operate the database, and detailed steps on the database administration or management shall not be provided.

The school lab has been installed with MySQL Community server v5.5, as well as MySQL Workbench 5.2. To software can be freely downloaded, and the installation is simple. You will need to have administrator rights to install the software, and create a database instance. For details, please consult your tutors accordingly.

Verify that MySQL Community server is running and operation. In the lab, MySQL is managed by a program XAMPP.

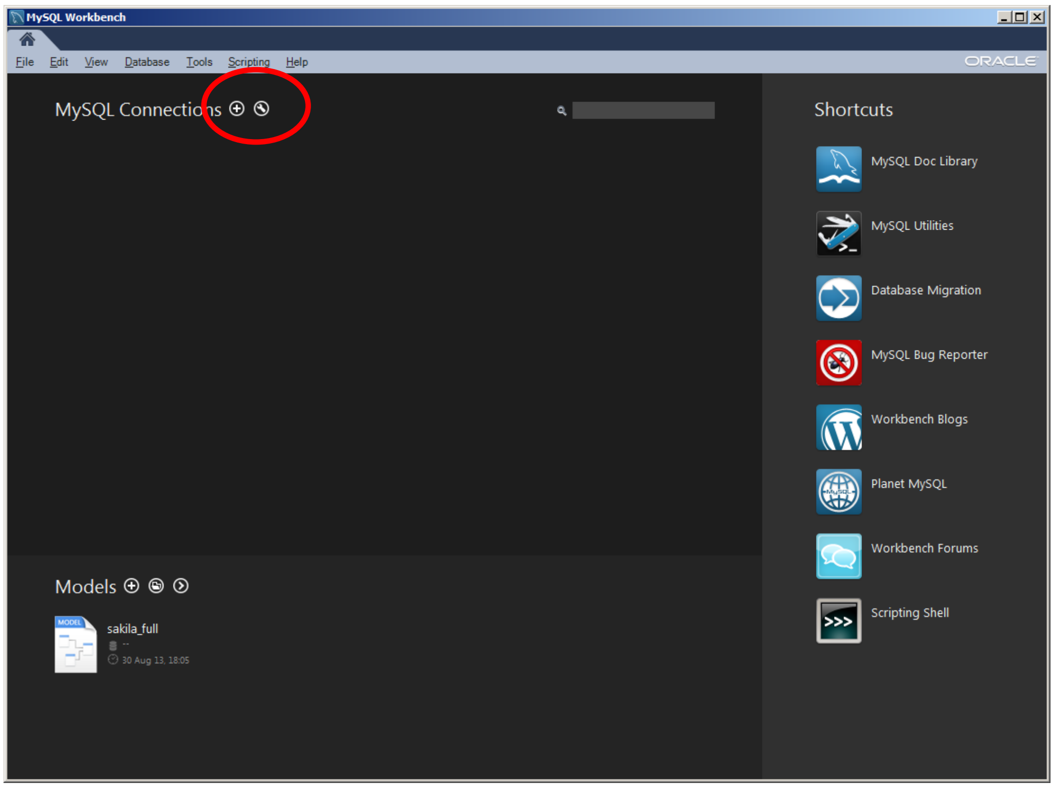
To check if MYSQL server is running on XAMPP : Go to C:\XAMPP. Double click on xampp-control.

Verify that MySQL is started. The default port should be set to 3306.

MySQL Workbench

1) Check if MySQL workbench is available in the system. If not, download and install from NTUlearn.

2) Launch MySQL Workbench 6.0.

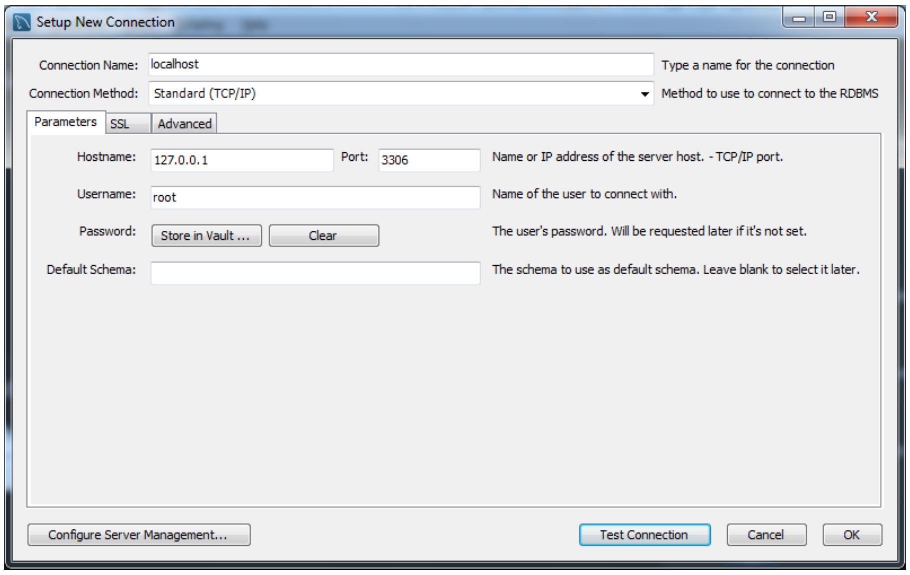


You could use the workbench to design, implement and manage your database server. For the workshop, we will focus on the implementation using the SQL Development section. You may explore the design, administration aspect on your own. If you have not created any connection to the database server, click on New Connection to create a new connection.

3) Click on the (+) icon to add a new connection.

4) To set up a new connection to the local database instance, you may use all the default parameters. Specify a connection name that best describe the connection name. Click on the Test Connection button and if the parameters entered are correct, a dialog will pop up and acknowledge. Click on OK button to save the connection.

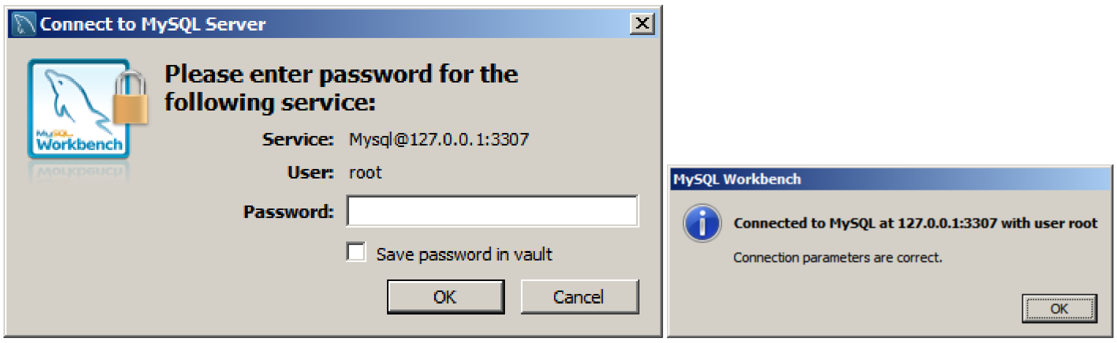
5) Set the connection name to :localhost.
6) Please avoid making changes to the Port No unless necessary. Verify the port number from XAMPP control.


Setup New Connection

Connection Name: localhost — Type a name for the connection
Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: 127.0.0.1 Port: 3306 — Name or IP address of the server host. - TCP/IP port.
Username: root — Name of the user to connect with.
Password: — The user's password. Will be requested later if it's not set.
Default Schema: — The schema to use as default schema. Leave blank to select it later.

7) Perform a "Test Connection". Select "ok" if "Test Connection" is successful.
8) You will be prompted to enter the '**root**' password. The password should be "**toor**".


Connect to MySQL Server

Please enter password for the following service:

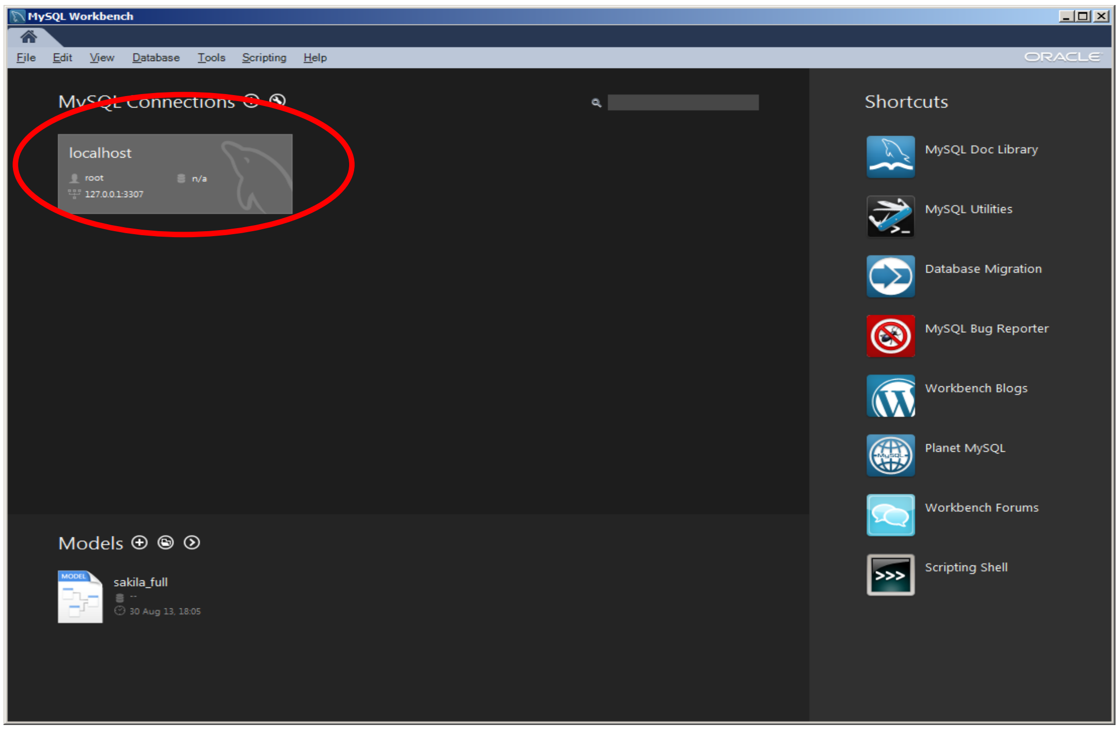
Service: Mysql@127.0.0.1:3307
User: root
Password:

MySQL Workbench

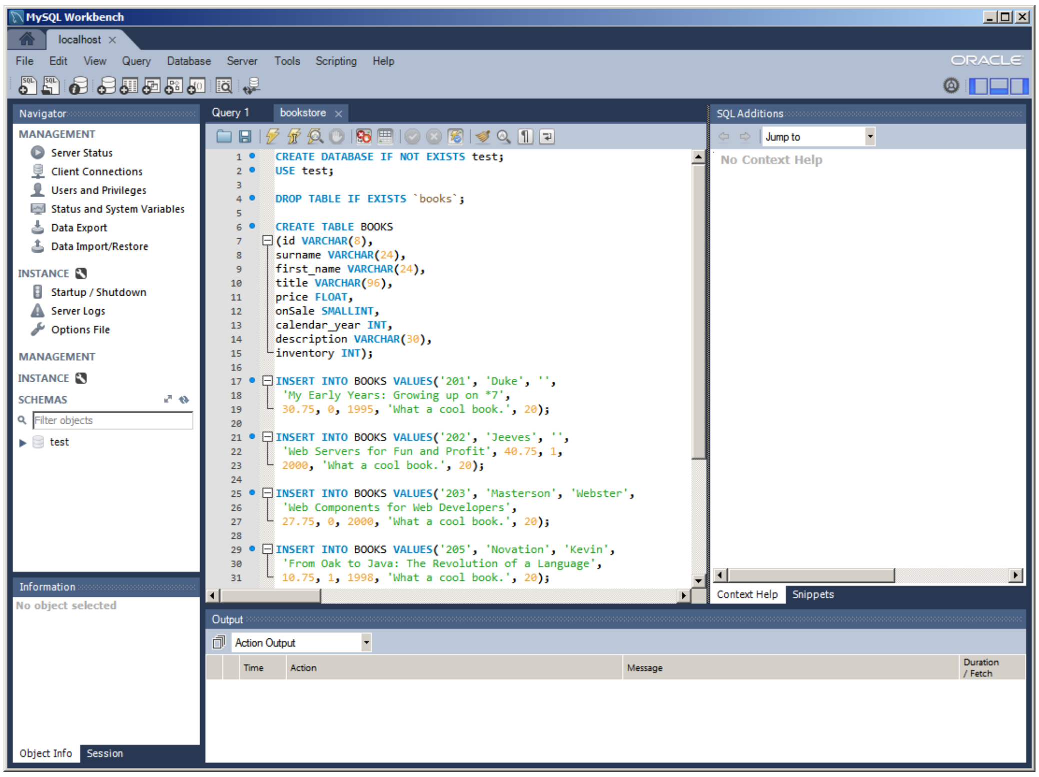
Connected to MySQL at 127.0.0.1:3307 with user root

Connection parameters are correct.

9) MySQL workbench is now connected with MySQL Server. Click on the MySQL connection.



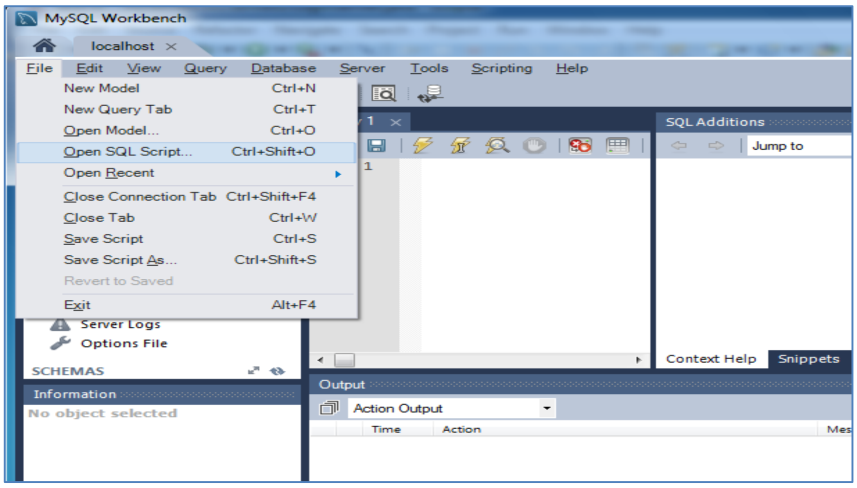
10) MySQL editor will be displayed.
You may use the SQL Editor to issue SQL command to manipulate your schema. You could also load pre-defined script and execute from the editor.



```sql
CREATE DATABASE IF NOT EXISTS test;
USE test;

DROP TABLE IF EXISTS `books`;

CREATE TABLE BOOKS
(id VARCHAR(8),
surname VARCHAR(24),
first_name VARCHAR(24),
title VARCHAR(96),
price FLOAT,
onSale SMALLINT,
calendar_year INT,
description VARCHAR(30),
inventory INT);

INSERT INTO BOOKS VALUES('201', 'Duke', '',
 'My Early Years: Growing up on *7',
 30.75, 0, 1995, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('202', 'Jeeves', '',
 'Web Servers for Fun and Profit', 40.75, 1,
 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('203', 'Masterson', 'Webster',
 'Web Components for Web Developers',
 27.75, 0, 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('205', 'Novation', 'Kevin',
 'From Oak to Java: The Revolution of a Language',
 10.75, 1, 1998, 'What a cool book.', 20);
```

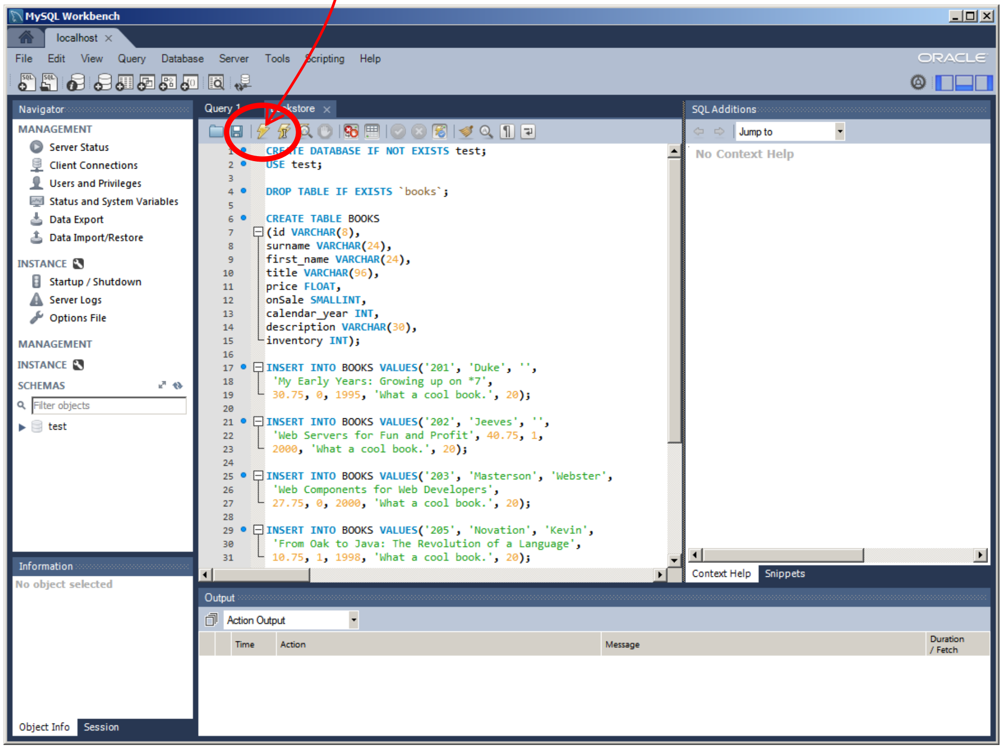
To load the SQL script from your file system,
select **File→Open SQL Script** from the menu and browse to folder to load the script, **bookstore.sql**. (Get script from ntuLearn)

11) Select File→Open SQL Script → bookstore.sql



12) The script will create a database (or schema) called **test** if it has not been created, and create a table called **books** and populate the table with some data.

13) Click on the **Execute** button to execute the script. If there is any error, the output panel at the bottom will display the error message. Verify that there is no error message. Verify that the table is created in test schema. You might need to refresh Object browser panel for the new table to be reflected.

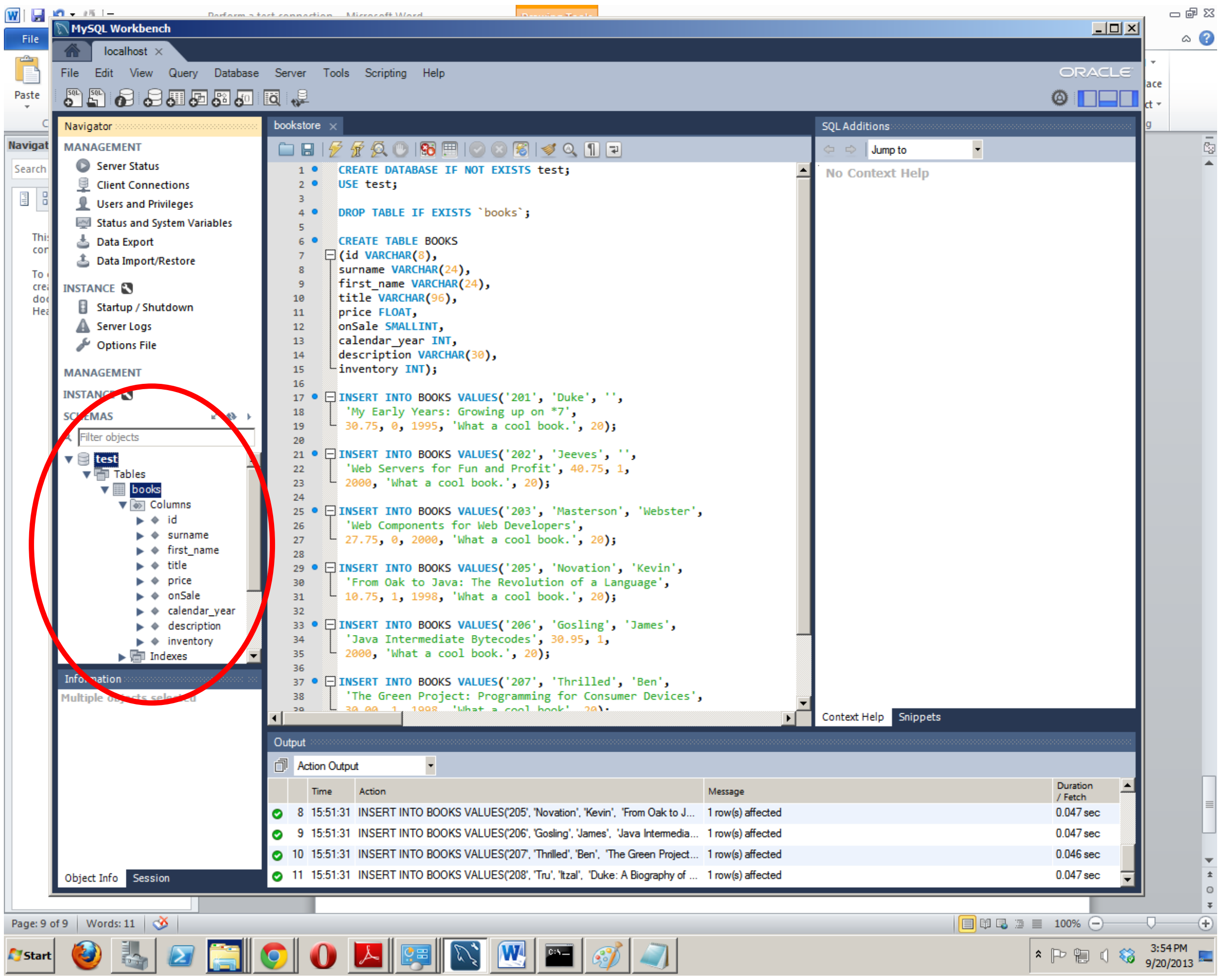


```sql
CREATE DATABASE IF NOT EXISTS test;
USE test;

DROP TABLE IF EXISTS `books`;

CREATE TABLE BOOKS
(id VARCHAR(8),
surname VARCHAR(24),
first_name VARCHAR(24),
title VARCHAR(96),
price FLOAT,
onSale SMALLINT,
calendar_year INT,
description VARCHAR(30),
inventory INT);

INSERT INTO BOOKS VALUES('201', 'Duke', '',
 'My Early Years: Growing up on *7',
 30.75, 0, 1995, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('202', 'Jeeves', '',
 'Web Servers for Fun and Profit', 40.75, 1,
 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('203', 'Masterson', 'Webster',
 'Web Components for Web Developers',
 27.75, 0, 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('205', 'Novation', 'Kevin',
 'From Oak to Java: The Revolution of a Language',
 10.75, 1, 1998, 'What a cool book.', 20);
```

bookstore.sql

```
CREATE DATABASE IF NOT EXISTS test;
USE test;

DROP TABLE IF EXISTS `books`;

CREATE TABLE BOOKS
(id VARCHAR(8),
surname VARCHAR(24),
first_name VARCHAR(24),
title VARCHAR(96),
price FLOAT,
onSale SMALLINT,
calendar_year INT,
description VARCHAR(30),
inventory INT);

INSERT INTO BOOKS VALUES('201', 'Duke', '',
 'My Early Years: Growing up on *7',
 30.75, 0, 1995, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('202', 'Jeeves', '',
 'Web Servers for Fun and Profit', 40.75, 1,
 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('203', 'Masterson', 'Webster',
 'Web Components for Web Developers', 27.75, 0,
 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('205', 'Novation', 'Kevin',
 'From Oak to Java: The Revolution of a Language',
 10.75, 1, 1998, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('206', 'Gosling', 'James',
 'Java Intermediate Bytecodes', 30.95, 1,
 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('207', 'Thrilled', 'Ben',
 'The Green Project: Programming for Consumer Devices',
 30.00, 1, 1998, 'What a cool book', 20);

INSERT INTO BOOKS VALUES('208', 'Tru', 'Itzal',
 'Duke: A Biography of the Java Evangelist',
 45.00, 0, 2001, 'What a cool book.', 20);
```

14) Verify from the Object Browser that a database with the name „test" together with a table name „books" were created successfully.

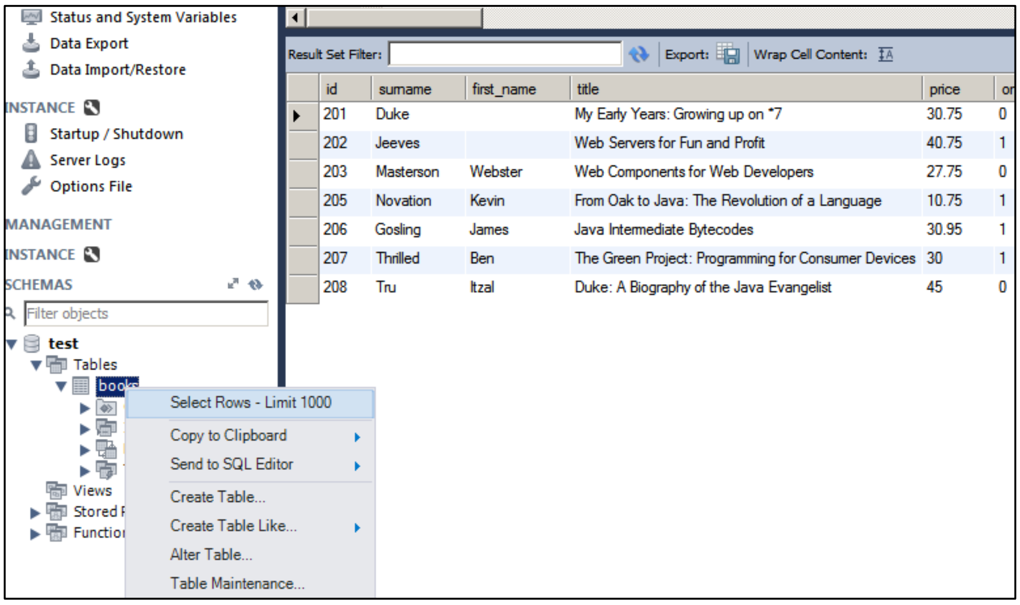


```
CREATE DATABASE IF NOT EXISTS test;
USE test;

DROP TABLE IF EXISTS `books`;

CREATE TABLE BOOKS
(id VARCHAR(8),
surname VARCHAR(24),
first_name VARCHAR(24),
title VARCHAR(96),
price FLOAT,
onSale SMALLINT,
calendar_year INT,
description VARCHAR(30),
inventory INT);

INSERT INTO BOOKS VALUES('201', 'Duke', '',
 'My Early Years: Growing up on *7',
 30.75, 0, 1995, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('202', 'Jeeves', '',
 'Web Servers for Fun and Profit', 40.75, 1,
 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('203', 'Masterson', 'Webster',
 'Web Components for Web Developers',
 27.75, 0, 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('205', 'Novation', 'Kevin',
 'From Oak to Java: The Revolution of a Language',
 10.75, 1, 1998, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('206', 'Gosling', 'James',
 'Java Intermediate Bytecodes', 30.95, 1,
 2000, 'What a cool book.', 20);

INSERT INTO BOOKS VALUES('207', 'Thrilled', 'Ben',
 'The Green Project: Programming for Consumer Devices',
```

Output

```
   Time      Action                                                          Message            Duration / Fetch
8  15:51:31  INSERT INTO BOOKS VALUES('205', 'Novation', 'Kevin', 'From Oak to J...   1 row(s) affected  0.047 sec
9  15:51:31  INSERT INTO BOOKS VALUES('206', 'Gosling', 'James', 'Java Intermedia...  1 row(s) affected  0.047 sec
10 15:51:31  INSERT INTO BOOKS VALUES('207', 'Thrilled', 'Ben', 'The Green Project... 1 row(s) affected  0.046 sec
11 15:51:31  INSERT INTO BOOKS VALUES('208', 'Tru', 'Itzal', 'Duke: A Biography of ... 1 row(s) affected  0.047 sec
```

15) Verify that the data are all loaded into the table. Right-click "Books" → Select Rows.



id	surname	first_name	title	price	or
201	Duke		My Early Years: Growing up on *7	30.75	0
202	Jeeves		Web Servers for Fun and Profit	40.75	1
203	Masterson	Webster	Web Components for Web Developers	27.75	0
205	Novation	Kevin	From Oak to Java: The Revolution of a Language	10.75	1
206	Gosling	James	Java Intermediate Bytecodes	30.95	1
207	Thrilled	Ben	The Green Project: Programming for Consumer Devices	30	1
208	Tru	Itzal	Duke: A Biography of the Java Evangelist	45	0

16) Create a database table “accounts” inside your bookstore database, with 2 fields : id and password
17) Right-click “Tables” → Create Table. Create the 2 DB fields/Columns “id” and “password”. Choose, VARCHAR(45) as datatype and set “id” as the primary key – “PK”.

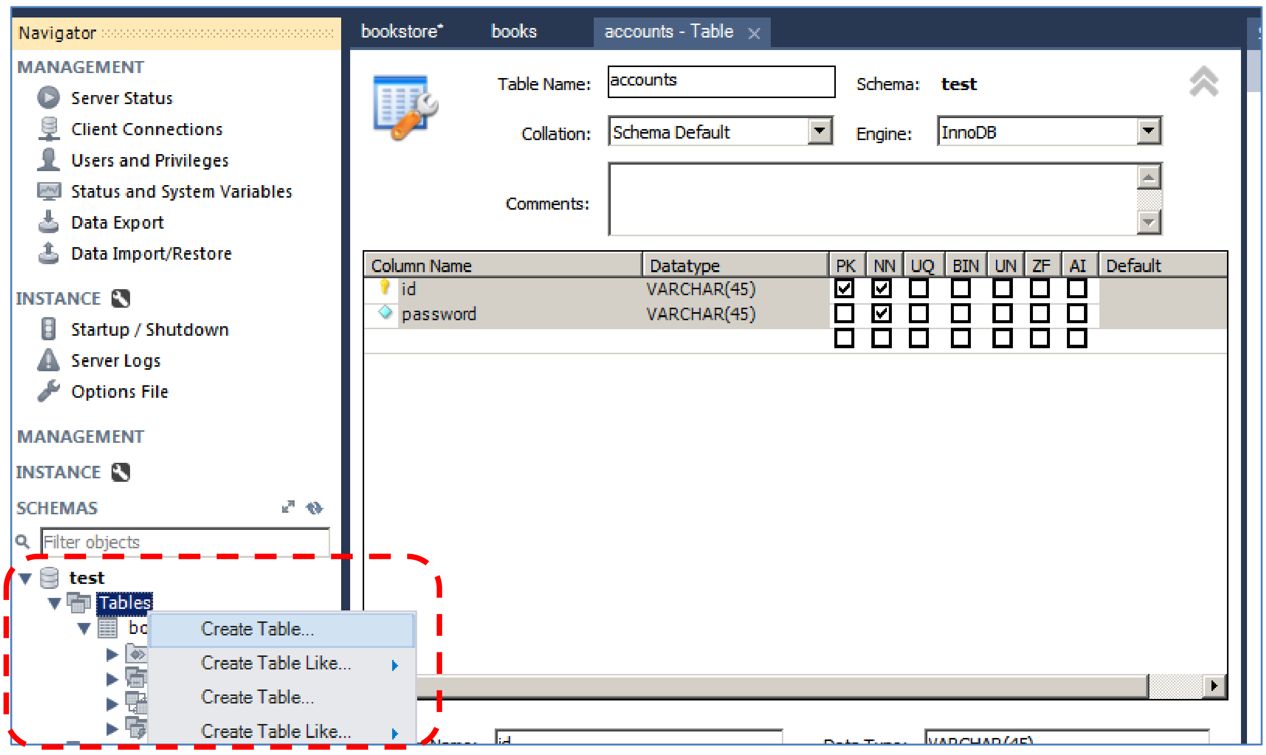


Table Name: accounts Schema: test
Collation: Schema Default Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	VARCHAR(45)	☑	☑	☐	☐	☐	☐	☐	
password	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	

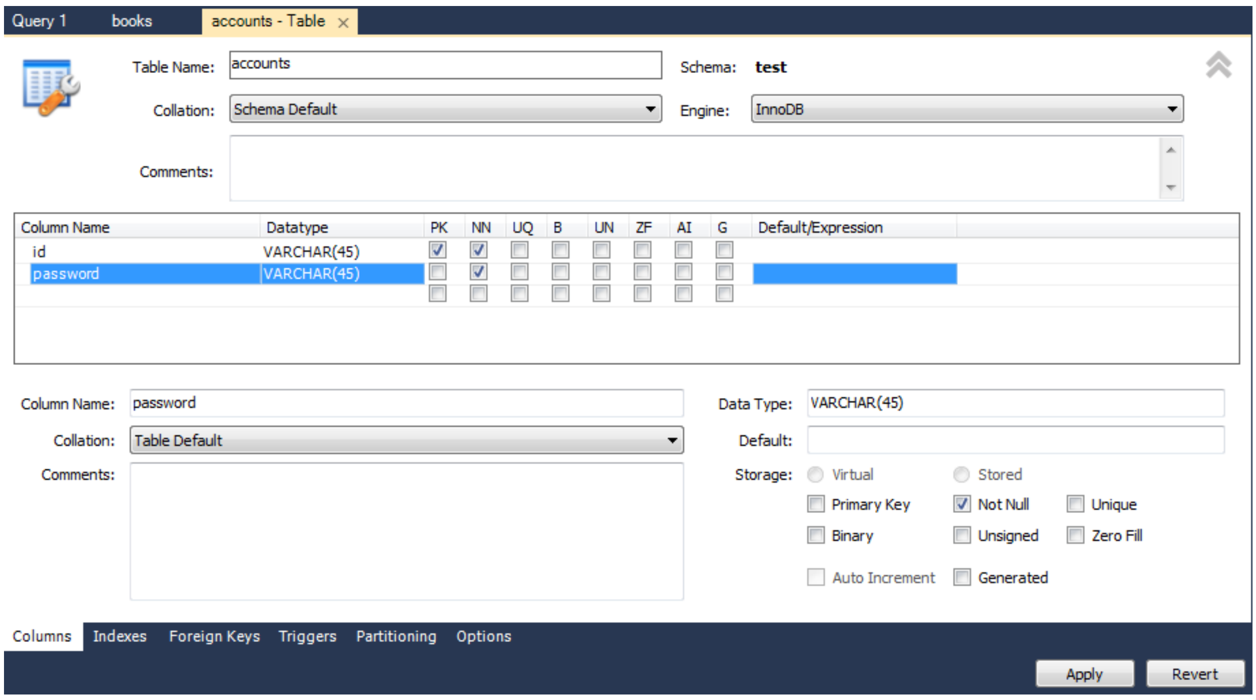


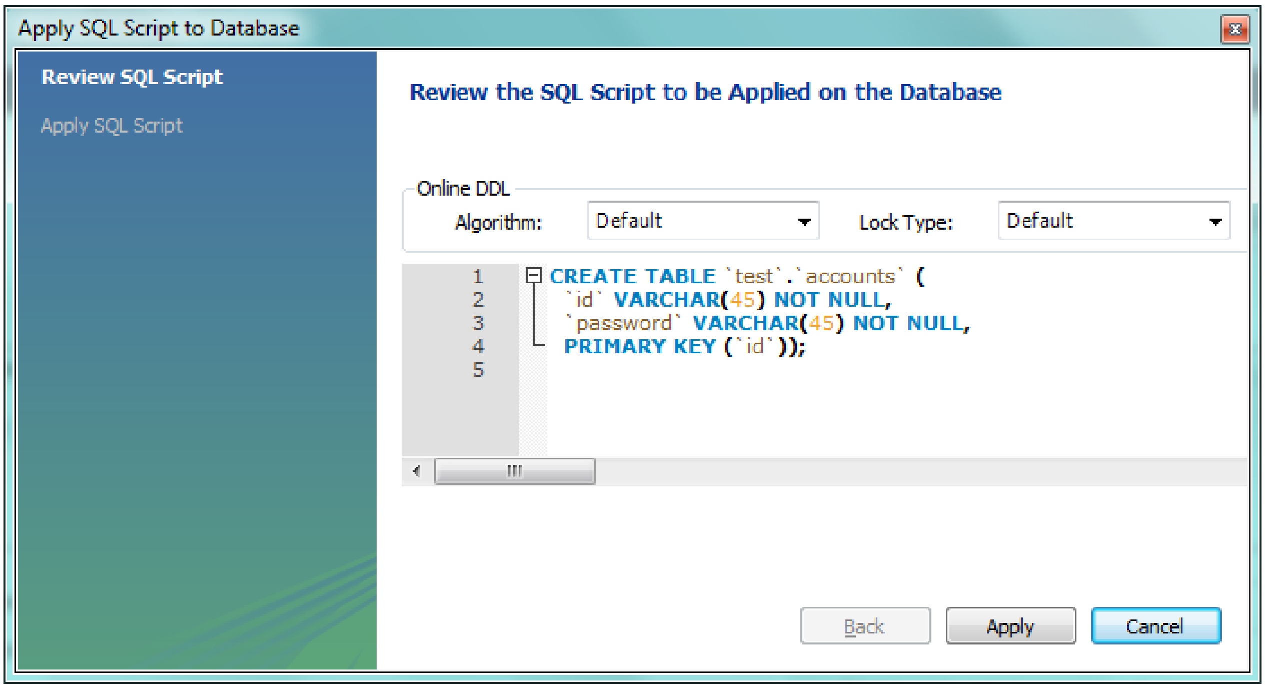
Table Name: accounts Schema: test
Collation: Schema Default Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	VARCHAR(45)	☑	☑	☐	☐	☐	☐	☐	☐	
password	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Column Name: password Data Type: VARCHAR(45)
Collation: Table Default Default:
Comments: Storage: ○ Virtual ○ Stored
☐ Primary Key ☑ Not Null ☐ Unique
☐ Binary ☐ Unsigned ☐ Zero Fill
☐ Auto Increment ☐ Generated

18) Click “Apply” when you are done creating the columns.

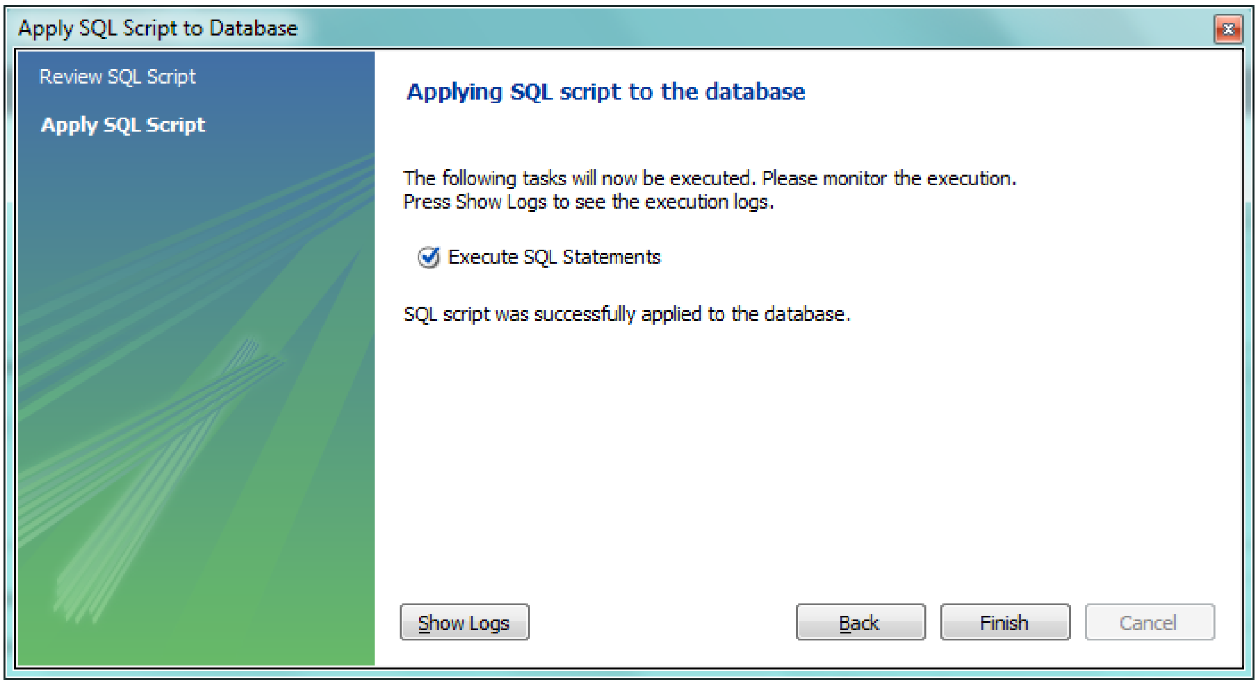
19) Select "Apply" again when asked to apply the SQL script.



Review the SQL Script to be Applied on the Database

Online DDL
Algorithm: Default
Lock Type: Default

```
CREATE TABLE `test`.`accounts` (
  `id` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`));
```



Applying SQL script to the database

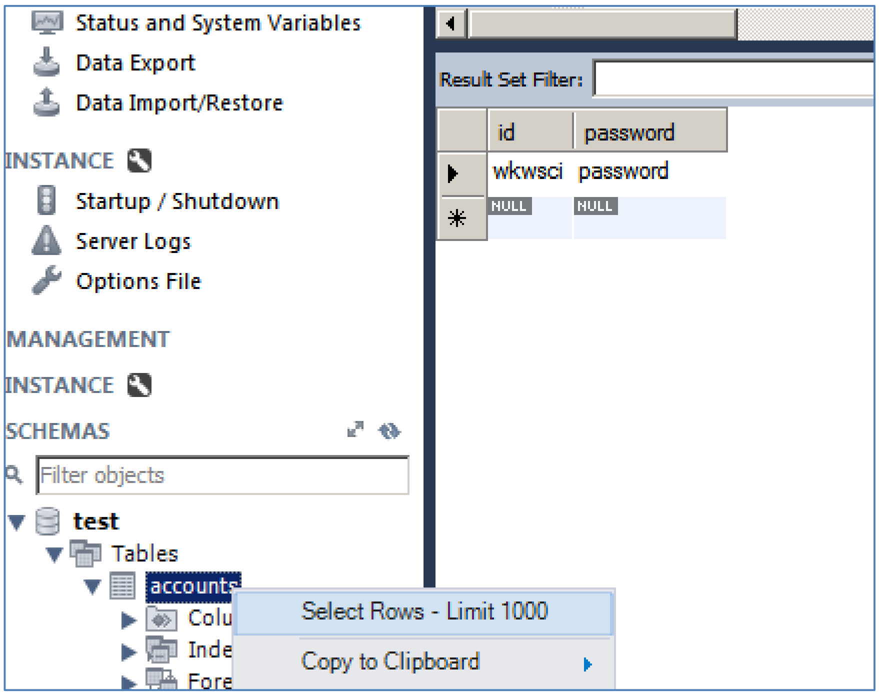
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

✓ Execute SQL Statements

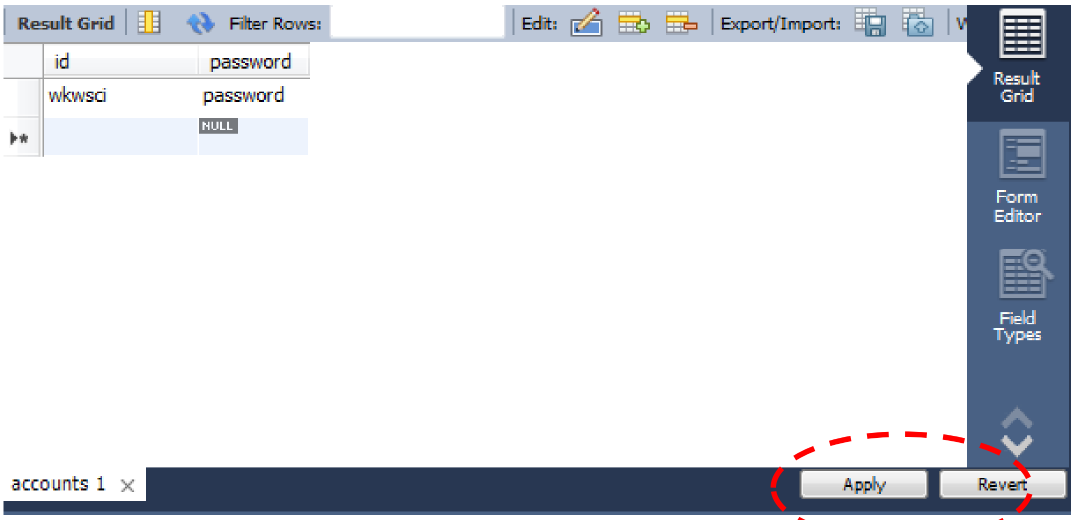
SQL script was successfully applied to the database.

20) Click "Finish"

21) Next add data into the accounts table. Right-click “accounts” → Select Rows.
22) Add in the id as “wkwsci” and password as “password”.



	id	password
▶	wkwsci	password
✳	NULL	NULL



	id	password
	wkwsci	password
▶✳		NULL

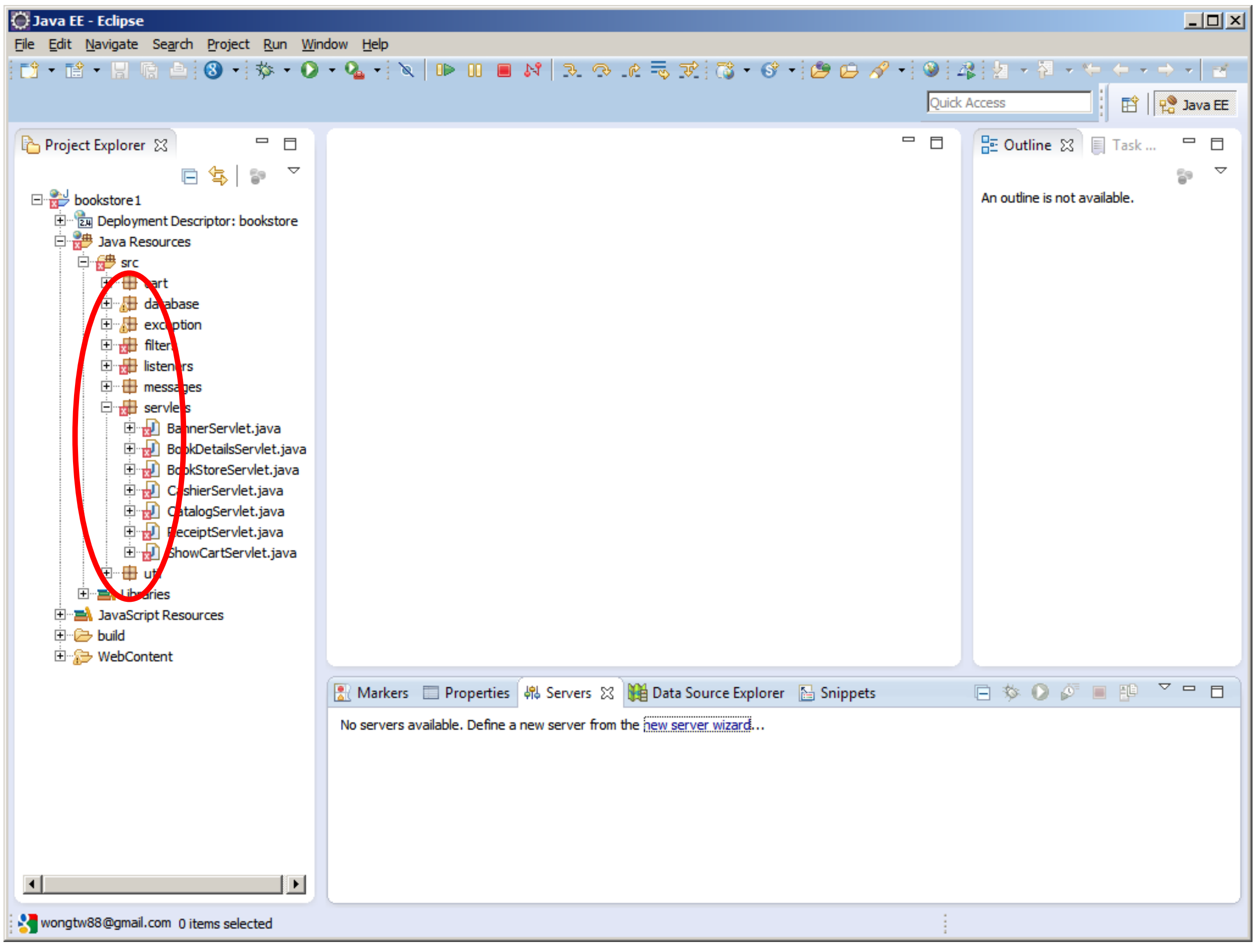
23) Select “Apply”.
24) We have successfully configured MySQL DB and populated the DB with some data using MySQL Workbench.

Developing a LoginServlet

1) Load a sample Web application from NTU Learn : **bookstore1.war**

   Bookstore1.war is a sample online bookstore less the login feature. We will implement a LoginServlet (**Controller**), a database class (**Model**) and 1 JSPs : login.jsp (**View**) to complete the application. Set ***id=wkwsci ; password=password***

2) Launch Eclipse. Import bookstore1.war.
3) File → Import → Web (War File)



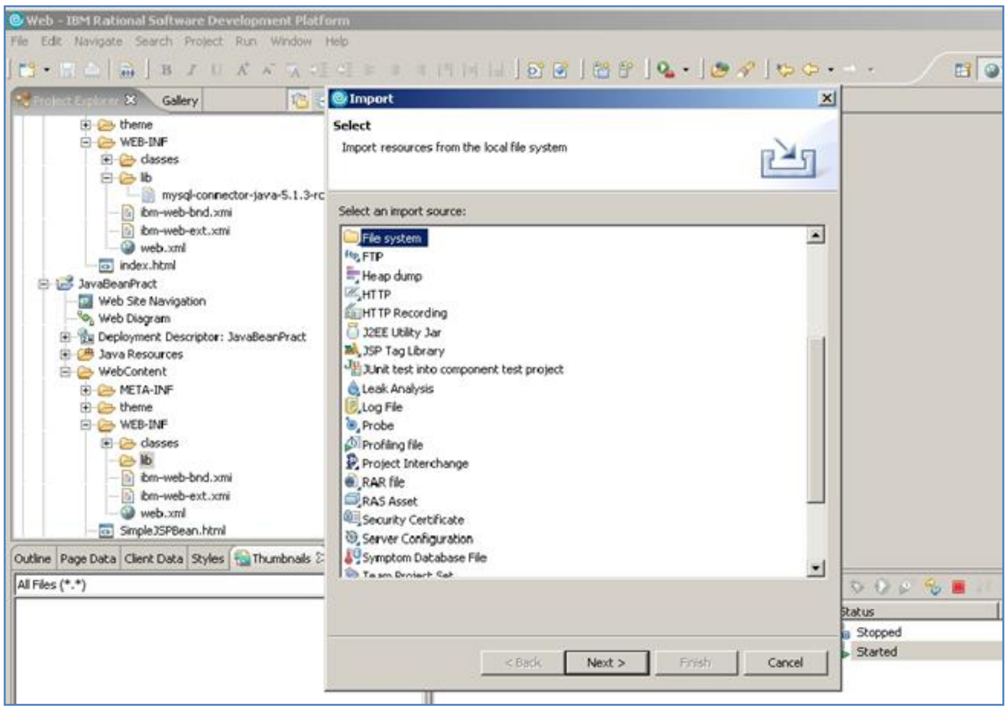
An outline is not available.

No servers available. Define a new server from the new server wizard...

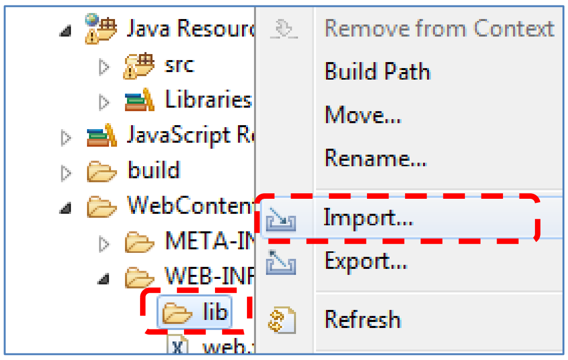
4) Most of the errors are due to missing Tomcat Server libraries. (***Remember how to make those errors dissappear?***)

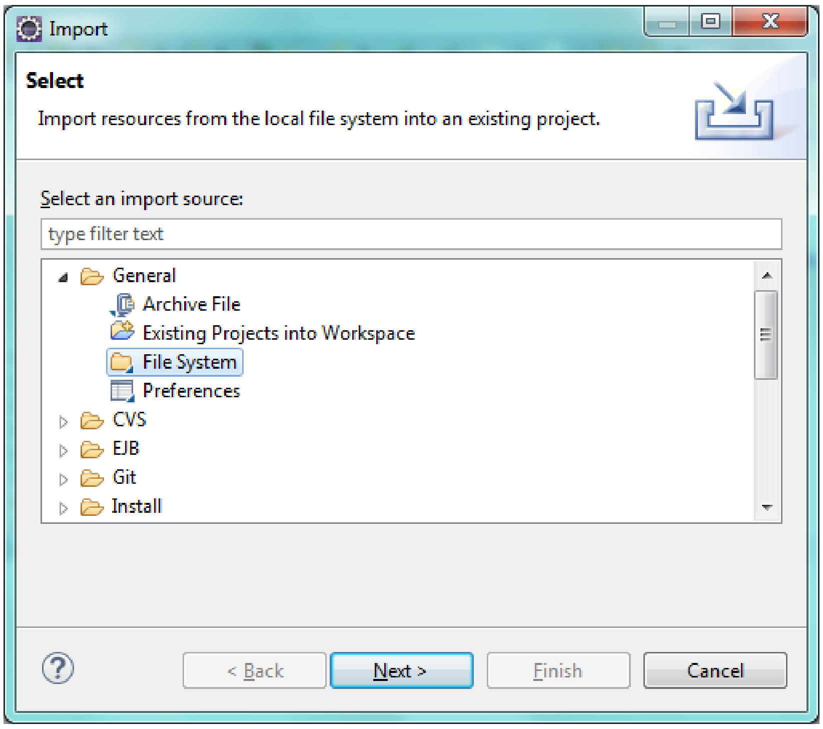
Setting up JDBC Driver

5) Set up the JDBC driver by importing the "MySQL Connector J" libraries into your web application.
6) Import mysql-connector-java-5.1.26-bin.jar into the **WEB-INF/lib** of your web application as shown below:



7) **Right click** on the **WEB-INF / lib** folder and click Import → (General)→ File System.




Import

Select

Import resources from the local file system into an existing project.

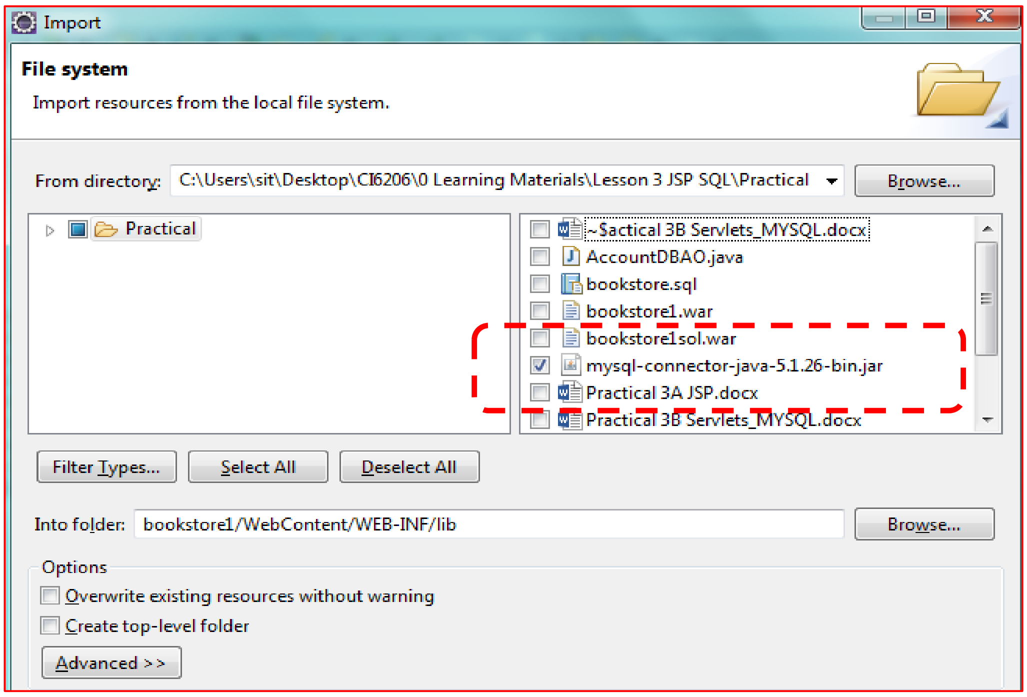
Select an import source:

type filter text

- General
  - Archive File
  - Existing Projects into Workspace
  - File System
  - Preferences
- CVS
- EJB
- Git
- Install

< Back | Next > | Finish | Cancel

8) Select “Next”
9) Browse and select the directory containing the jar file. Check the jar file.
10) Click “Finish”.


Import

File system

Import resources from the local file system.

From directory: C:\Users\sit\Desktop\CI6206\0 Learning Materials\Lesson 3 JSP SQL\Practical

- Practical

- [ ] ~$actical 3B Servlets_MYSQL.docx
- [ ] AccountDBAO.java
- [ ] bookstore.sql
- [ ] bookstore1.war
- [ ] bookstore1sol.war
- [x] mysql-connector-java-5.1.26-bin.jar
- [ ] Practical 3A JSP.docx
- [ ] Practical 3B Servlets_MYSQL.docx

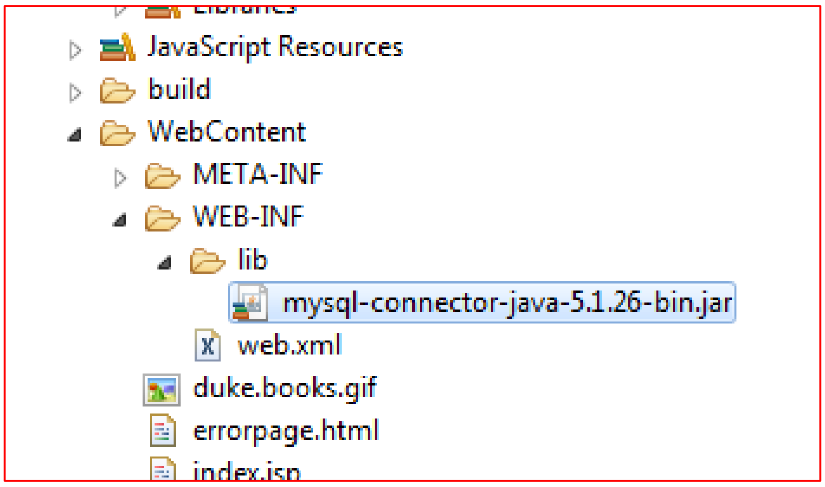
Filter Types... | Select All | Deselect All

Into folder: bookstore1/WebContent/WEB-INF/lib

Options
- [ ] Overwrite existing resources without warning
- [ ] Create top-level folder

Advanced >>

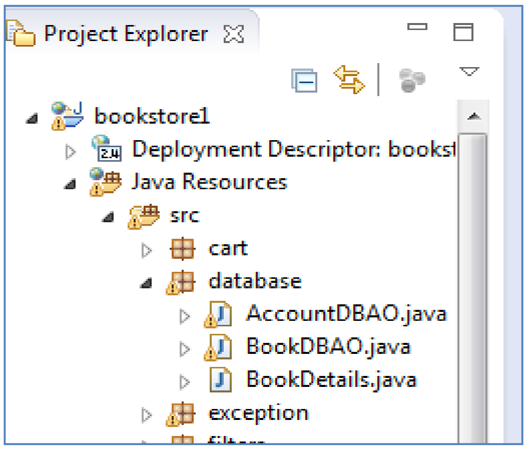
11) The jar file required by Eclipse to connect to the MySQL database is loaded into the WEB-INF/lib folder.



12) Browse and select the directory containing the jar file. Check the jar file.

Create the AccountDBAO.java class

1) Add a database class AccountDBAO.java into java package “database” by ‘drag-dropping’ the file from a file explorer into the Eclipse. Inspect and examine AccountDBAO.java.
2) What is the purpose of AccountDBAO.java ? ***Change the login and password to match the SQL Server in your machine.*** *Make necessary changes to BookDBAO.java as well.*

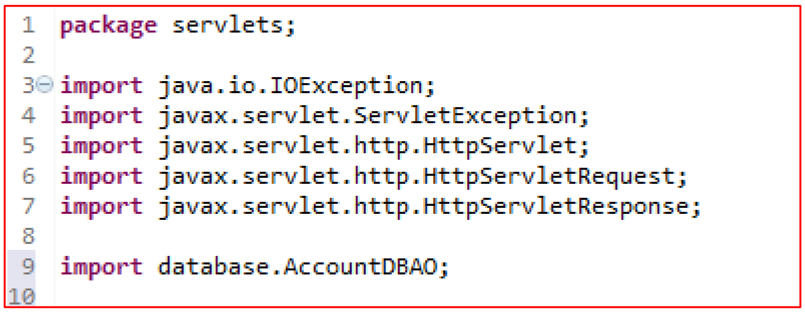


3) Verify the database connection in AccountDBAO.java and BookDBAO.java.

```
// Database configuration
public static String url = "jdbc:mysql://localhost:3306/test";
public static String dbdriver = "com.mysql.jdbc.Driver";
public static String username = "root";
public static String password = "toor";
```

4) Below shows a snippet of codes for login.html and LoginServlet. User enters userid in the “id” field, and the password in the “password” field. The parameter name/value pairs are appended to the url and sent to the “LoginServlet” through HTTP GET command. LoginServlet handles the GET request through the doGet() method. The id/password pair are extracted from the request through the getParameter() API. If the password and id matches the ones found in the database (accounts), the servlet will forward the request to /bookstore, otherwise, the request will be forwarded to the login.jsp page again.

Include a reference to the Database class file “AccountDBAO” : *import database.AccountDBAO;*



```
package servlets;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import database.AccountDBAO;

```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {


            String id = request.getParameter("id");
            String password = request.getParameter("password");
            boolean result = false ;
            //ensure that out.close() do not come before response object

           try {
                   AccountDBAO account = new AccountDBAO();
                   result = account.authenticate(id, password);
           }
           catch (Exception e)
           {
                   e.printStackTrace();

           }

           if (result){
                   request.getRequestDispatcher("/bookstore").forward(request,response);
                   return;
            }
           else {
                   response.sendRedirect("login.jsp");
                   return;
           }
}
```

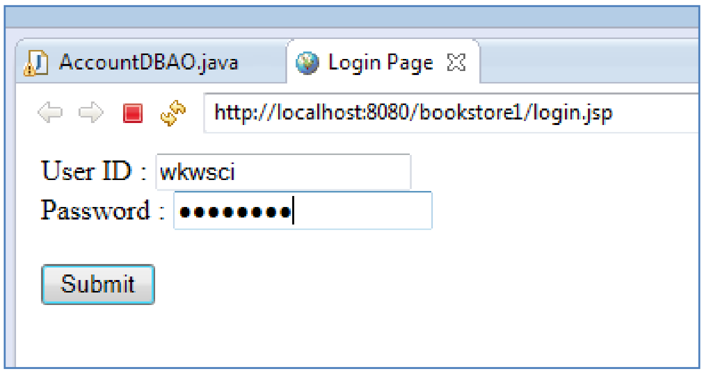
1) Make necessary changes to BookDBAO.java as well.
2) Create a login.jsp to allow user to send id and password to LoginServlet (Refer to practical from previous lesson). Add the following lines of codes to your login.jsp.

```
<form action="LoginServlet" method="GET">
User ID : <input type="text" name="id" size="20"><br>
Password : <input type="password" name="password" size="20">
<br><br>
<input type="submit" value="Submit">
</form>
```

3) Test run your login.jsp and LoginServlet.jsp. Remember to set your Server configuration to Tomcat 7.0 before running your application in Eclipse.

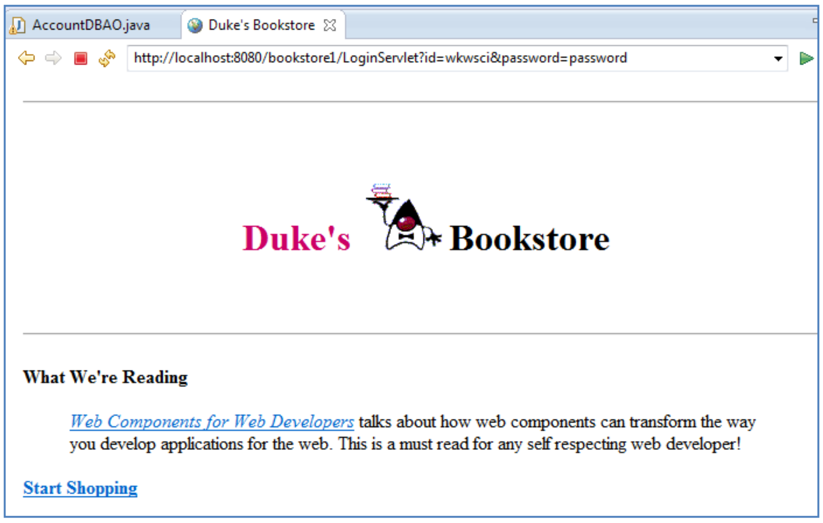
4) Success operation

login.jsp



http://localhost:8080/bookstore1/login.jsp

User ID : wkwsci
Password : ●●●●●●●●



http://localhost:8080/bookstore1/LoginServlet?id=wkwsci&password=password

What We're Reading

Web Components for Web Developers talks about how web components can transform the way you develop applications for the web. This is a must read for any self respecting web developer!

Start Shopping

5) Possible error : Access denied

```
Exception in BookDBAO: java.lang.ClassNotFoundException: com.mysql.jdbc.Driver Couldn't
create bookstore database bean: Couldn't open connection to database:
com.mysql.jdbc.Driver
Exception in BookDBAO: java.sql.SQLException: Access denied for user 'root'@'localhost'
(using password: YES)
Couldn't create bookstore database bean: Couldn't open connection to database: Access
denied for user 'root'@'localhost' (using password: YES)
```

Check the credential on sql server and make the necessary changes to BookDBAO.java and AccountDBAO.java. Verify the password use in the codes matches the one used in your MySQL account.