Extra Lab 5: Hibernate Queries

In this lab, you will write a program that uses Hibernate to query entities.

Objectives:

- To work with the Hibernate Query Language (HQL)

Part 1: Setup

In this part, you will import an project that contains entities and a Web application into which you can enter and execute queries.

Note: You must use JBoss Developer Studio for this lab instead of Eclipse.

Steps:

_1. This lab uses the JBoss Application server, so you need to ensure that the Developer Studio is configured for JBoss:

   a. Switch to the *JEE Perspective* by selecting from the Studio's menu *Window - Open Perspective - Other - Java EE (default)*.

   b. From the menu, choose *Window - Preferences*, then expand the *Server* category and click on the *Runtime Environments* category.

      If you see an entry for *JBoss 7.1.1*, you can skip the rest of this procedure.

   c. If you don't see *JBoss 7.1.1*, click the Add button and scroll down to find and expand the *JBoss Community* category.

      Click on *JBoss 7.1 Runtime* and click Next.

   d. On the next page, for the *Home Directory*, click the Browse button and navigate to where JBoss installed. This directory should be something like:

      ```
      c:\Users\Administrator\java\jboss-as-7.1.1.Final

           or

      c:\java\jboss-as-7.1.1.Final
      ```

      Ask your instructor if you have doubts.

      After selecting the JBoss home directory, press Finish followed by OK.

_2. Next, you must configure a connection from the Developer Studio to JBoss. Please follow these steps even if you were able to skip part of the last step.

   a. Find the *Servers* tab at the bottom of the Studio window and click it.

      If you see an entry for JBoss, right-click on it and choose *Delete* to remove the server.

   b. In the Servers tab, click the link "No servers are available...." to start the wizard.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

c. On the first wizard page, expand the *JBoss Community* category (NOT JBoss Enterprise Middleware) and select *JBoss AS 7.1*, then press Next.

d. On the next wizard page, press Next.

e. On the next wizard page, press Finish.

_3. Next, you need to set up the database for this project:

a. Obtain the *createorders.sql* and *dbCreateExtraLab5.cmd* files from the instructor. Download these files to the *Derby* directory where you should already have files like *dbSQL.cmd* that you've used earlier in lab.

b. Run the dbCreateExtraLab5.cmd script to create and populate the tables needed for this lab. This is a large script, so it will take a while to complete. Please be patient.

_4. Your next job is to import the completed project that defines the entities that you will query. Follow these steps:

a. Obtain the *ExtraLab5.zip* file from the instructor and download it to a directory on your lab computer. Remember where you put it!

b. From the menu, choose *File - Import* to start the wizard.

c. On the first wizard page, select *General - Existing Projects into Workspace*, then press Next.

d. On the next page, for the *Select archive file*, press the Browse button and navigate to where you downloaded the ExtraLab5.zip file, then select that ZIP.

Press Next.

e. On the next wizard page, put a checkmark next to the *ExtraLab5Web* entry, then press Finish.

_5. If your imported project has a red exclamation point on it in the Package Explorer, it might be due to an invalid JRE reference. Fix it with this procedure:

a. In the Package Explorer, right-click on the project and choose *Build Path - Configure Build Path*, then click the *Libraries* tab at the top of the window.

b. In the "JAR and class folders" list, check to see if the JRE entry has a red X on it. If so, double-click on it, then choose the *Workspace Default JRE* entry.

Press Finish followed by Close - the red exclamation point on the project should be gone.

_6. Examine the *domain* package and note that it defines four entity classes. Examine the *domain.Customer*, *domain.Order*, *domain.LineItem* and *domain.Product* entities and familiarize yourself with them.

Note that the entities define relationships. Please sketch a UML class diagram in the space below:

_7. Examine the *queries* package and note that it defines a class named *QueryEvaluatorBean*.

Please examine it and ensure that you understand it.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

_8. Examine the *WebContent* folder and note that it defines the following JSPs:

- **index.jsp** defines an input form into which you can enter JPQL queries against the entities.
- **results.jsp** displays the results of the query that you enter into index.jsp.

Note also that the *servlets* package defines a simple servlet named **servlets.ProcessQueriesServlet** that runs in response to the input form. The servlet essentially invokes the QueryEvaluatorBean's *evaluateSingleResultQuery* or *evaluateMultipleResultQuery* and stores the results in the HTTP session so that the results.jsp can display them.

Part 2: Running Queries

In this part, you will use the provided Web application to test some simple queries against the Customer, Order LineItem and Product entities.

Steps:

_1. To get started, expand *WebContent* and right-click on *index.jsp* and choose *Run As - Run on Server*, and press Finish. Eclipse should deploy the application and display a Web page into which you can enter HQL queries.

_2. The Web page shows a UML diagram of the entities for your convenience, and has the following input widgets:

- A multi-line entry field into which you can type HQL queries. It's OK to press Enter when typing a query to split it across multiple lines for readability.
- A single-line entry field into which you can type a fully qualified class name. This field is required: you will need to figure out the correct "return" type depending on the query that you enter.
- Two radio buttons that let you choose whether the application displays a single or multiple result entities. Again, you will need to select one of these options depending on your query.

_3. Now let's try a simple query:

a. In the *Query* multi-line entry field, enter:

```
FROM Customer
```

This query should return all Customer entities.

b. In the *Target entity class* entry field, enter:

```
domain.Customer
```

Note that your query returns Customer entities and that **domain.Customer** is the fully qualified name of the Customer entity class.

c. Select the *Multiple Results(List)* option. The query should return more than one Customer entity.

d. Press the Submit button -- you should see a table showing all of the Customer entities in the database.

_4. Follow these steps to enter a slightly different query:

a. At the top of the browser pane, press the "left-arrow" icon to go back to the query page.

b. Modify the query, adding text so it looks like:

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

```
FROM Customer c
WHERE c.customerID=2
```

c. The returned class is still Customer, so you don't need to modify the *Target Entity* entry field, but the new query should return a single entity, so select the *Single Result* option.

d. Press the Submit button to see the single entity displayed. The name should be **Sue Jones**.

_5. Repeat the above to first display all of the Product entities, then only the Product entity with productID=2.

Note: Be sure to change the *Target Entity* entry field to **domain.Product**!

Write down your querys' text here:

```
// display all products




// display product #2
```

_6. Now that you've learned a bit about how to use the Web page, your mission, should you choose to accept it, is to craft queries to answer the following questions. Some of these are a bit tricky, so don't feel bad if you can't figure out how to do all of them.

a. Display all orders that are not yet complete.

Hint: Be sure to set the *Target entity class name* correctly!

b. Display all line items that have a quantity greater than 22.

c. Display the count of products (Hint: target entity class should be **java.lang.Long**).

d. Display the average product price (Hint: target entity class should be **java.lang.Double**).

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

e. Display all customers sorted by area code in descending order.

f. Display all products that are lenses, but are not filters.

g. Display all products with above average prices.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.