Advanced Java

Lab Assignment 10

Date: 6/10/2018

Roll no:50

Create a Hibernate web application to search Employees residing in a particular city. The database tables required are as follows:

|  |
| --- |
| **Address** |
| ID |
| Street Name |
| City Name |
| State Name |
| Zip Code |

|  |
| --- |
| **Employee** |
| ID |
| First Name |
| Last Name |
| Address – Foreign Key |

**Hibernate.cfg.xml**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hibernate.connection.driver\_class">com.mysql.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernate\_query?useSSL=false</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">root</property>

<mapping resource="POJOS/Employee.hbm.xml"/>

<mapping resource="POJOS/Address.hbm.xml"/>

</session-factory>

</hibernate-configuration>

**Hibernate.reveng.xml**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-reverse-engineering PUBLIC "-//Hibernate/Hibernate Reverse Engineering DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-reverse-engineering-3.0.dtd">

<hibernate-reverse-engineering>

<schema-selection match-catalog="hibernate\_query"/>

<table-filter match-name="employee"/>

<table-filter match-name="address"/>

</hibernate-reverse-engineering>

**Address.hbm.xml**

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<!-- Generated 6 Oct, 2018 2:34:15 PM by Hibernate Tools 3.6.0 -->

<hibernate-mapping>

<class name="POJOS.Address" table="address" catalog="hibernate\_query">

<id name="addressid" type="int">

<column name="addressid" />

<generator class="assigned" />

</id>

<property name="streetName" type="string">

<column name="StreetName" length="45" />

</property>

<property name="cityName" type="string">

<column name="CityName" length="45" />

</property>

<property name="stateName" type="string">

<column name="StateName" length="45" />

</property>

<property name="zipCode" type="string">

<column name="ZipCode" length="45" />

</property>

<set name="employees" table="employee" inverse="true" lazy="true" fetch="select">

**<key>**

**<column name="Address" />**

**</key>**

**<one-to-many class="POJOS.Employee" />**

**</set>**

</class>

</hibernate-mapping>

**Address.java**

package POJOS;

import java.util.HashSet;

import java.util.Set;

public class Address implements java.io.Serializable {

private int addressid;

private String streetName;

private String cityName;

private String stateName;

private String zipCode;

private Set employees = new HashSet(0);

public Address() {

}

public Address(int addressid) {

this.addressid = addressid;

}

public Address(int addressid, String streetName, String cityName, String stateName, String zipCode, Set employees) {

this.addressid = addressid;

this.streetName = streetName;

this.cityName = cityName;

this.stateName = stateName;

this.zipCode = zipCode;

this.employees = employees;

}

public int getAddressid() {

return this.addressid;

}

public void setAddressid(int addressid) {

this.addressid = addressid;

}

public String getStreetName() {

return this.streetName;

}

public void setStreetName(String streetName) {

this.streetName = streetName;

}

public String getCityName() {

return this.cityName;

}

public void setCityName(String cityName) {

this.cityName = cityName;

}

public String getStateName() {

return this.stateName;

}

public void setStateName(String stateName) {

this.stateName = stateName;

}

public String getZipCode() {

return this.zipCode;

}

public void setZipCode(String zipCode) {

this.zipCode = zipCode;

}

public Set getEmployees() {

return this.employees;

}

public void setEmployees(Set employees) {

this.employees = employees;

}}

**Employee.hbm.xml**

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<!-- Generated 6 Oct, 2018 2:34:15 PM by Hibernate Tools 3.6.0 -->

<hibernate-mapping>

<class name="POJOS.Employee" table="employee" catalog="hibernate\_query">

<id name="employeeId" type="int">

<column name="EmployeeID" />

<generator class="assigned" />

</id>

**<many-to-one name="address" class="POJOS.Address" fetch="select">**

**<column name="Address" />**

**</many-to-one>**

<property name="firstName" type="string">

<column name="FirstName" length="45" />

</property>

<property name="lastName" type="string">

<column name="LastName" length="45" />

</property>

</class>

</hibernate-mapping>

**Employee.java**

package POJOS;

public class Employee implements java.io.Serializable {

private int employeeId;

**private Address address;**

private String firstName;

private String lastName;

public Employee() {

}

public Employee(int employeeId) {

this.employeeId = employeeId;

}

public Employee(int employeeId, Address address, String firstName, String lastName) {

this.employeeId = employeeId;

this.address = address;

this.firstName = firstName;

this.lastName = lastName;

}

public int getEmployeeId() {

return this.employeeId;

}

public void setEmployeeId(int employeeId) {

this.employeeId = employeeId;

}

public Address getAddress() {

return this.address;

}

public void setAddress(Address address) {

this.address = address;

}

public String getFirstName() {

return this.firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return this.lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}}

**Index.html**

<!DOCTYPE html>

<html>

<head>

<title>SearchByCity</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

</head>

<body>

<form method="POST" action="SearchByCity">

<center>

Enter the City Name:<input type="text" name="CityName">

<br><br>

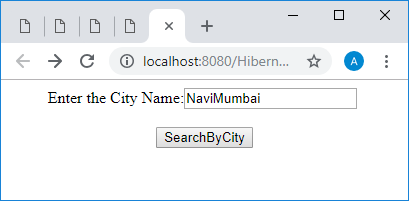
<input type="submit" value="SearchByCity">

</center>

</form>

</body>

</html>



**SearchCity.java**

import POJOS.Employee;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

String CityName=request.getParameter("CityName");

try (PrintWriter out = response.getWriter())

{

Configuration c=new Configuration();

c.configure("hibernate.cfg.xml");

SessionFactory sf=c.buildSessionFactory();

Session s=sf.openSession();

out.println("Session");

String hql="FROM Employee E where E.address.cityName='"+CityName+"'";

out.println("Query");

Query query=s.createQuery(hql);

List<Employee> results=query.list();

out.println("List");

/\* TODO output your page here. You may use following sample code. \*/

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet SearchByCity</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Search BY City"+CityName+"</h1>");

out.println("<table border='1'>");

for(Employee EMP:results)

{

out.println("<tr>");

out.println("<td>");

out.println(EMP.getFirstName());

out.println("</td>");

out.println("<td>");

out.println(EMP.getLastName());

out.println("</td>");}

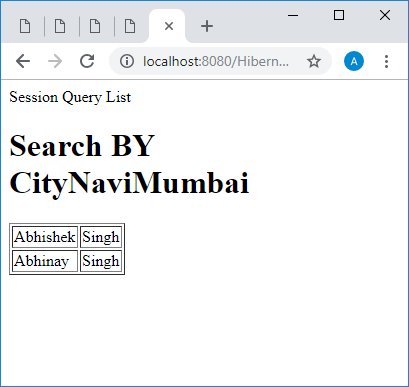
out.println("<table>");

out.println("</body>");

out.println("</html>");

}

}



**OrderByLastName.html**

<!DOCTYPE html>

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

</head>

<body>

<form method="POST" action="OrderByLastName">

<center>

Enter the City Name:<input type="text" name="CityName" value="Pune">

<br><br>

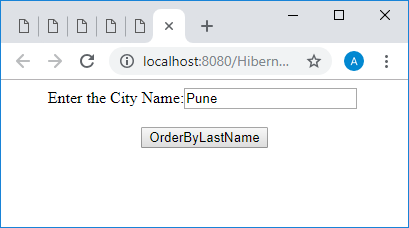
<input type="submit" value="OrderByLastName">

</center>

</form>

</body>

</html>



**OrderByLastName.java**

import POJOS.Employee;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

String CityName=request.getParameter("CityName");

try (PrintWriter out = response.getWriter()) {

Configuration c=new Configuration();

c.configure("hibernate.cfg.xml");

SessionFactory sf=c.buildSessionFactory();

Session s=sf.openSession();

String hql="FROM Employee E where E.address.cityName='"+CityName+"' order by E.lastName";

Query query=s.createQuery(hql);

List<Employee> results=query.list();

/\* TODO output your page here. You may use following sample code. \*/

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet OrderByLastName</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Servlet OrderByLastName"+CityName+"</h1>");

out.println("<table border='1'>");

for(Employee EMP:results)

{

out.println("<tr>");

out.println("<td>");

out.println(EMP.getFirstName());

out.println("</td>");

out.println("<td>");

out.println(EMP.getLastName());

out.println("</td>");

out.println("</tr>");

}

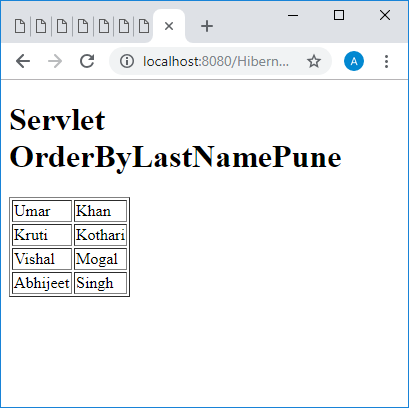
out.println("</table>");

out.println("</body>");

out.println("</html>");

}

}



**SearchState.html**

<!DOCTYPE html>

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

</head>

<body>

<form method="POST" action="SearchByState">

<center>

Enter the State Name:<input type="text" name="StateName">

<br><br>

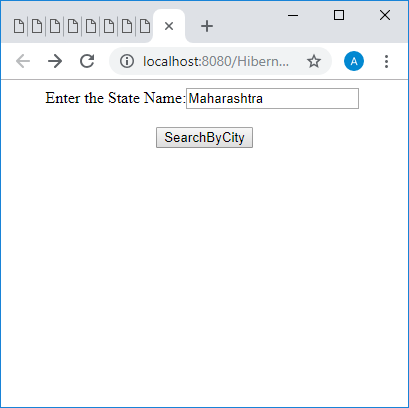
<input type="submit" value="SearchByCity">

</center>

</form>

</body>

</html>



**SearchByState.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class SearchByState extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

String StateName=request.getParameter("StateName");

try (PrintWriter out = response.getWriter())

{

Configuration c=new Configuration();

c.configure("hibernate.cfg.xml");

SessionFactory sf=c.buildSessionFactory();

Session s=sf.openSession();

String hql="Select count(\*) FROM Employee E where E.address.stateName='"+StateName+"'";

Long count=(Long) s.createQuery(hql).uniqueResult();

/\* TODO output your page here. You may use following sample code. \*/

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet SearchByState</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Number of the Employee Belonging to State "+StateName+" </h1>");

out.println("<h1>"+count+"<h1>");

out.println("</body>");

out.println("</html>");

}

}

