Using Connection Pooling with JNDI

1. JNDI(Java Naming And Directory Interface)
   1)Naming service 및 directory service에 접근하기 위한 API
   2)필요한 data를 name/value 형식으로 저장하고, 필요한 data는 name 값을 이용하여 value값을 얻는 형태
   3)대표적인 경우가 DNS
   4)Web browser에서 영문형식으로 domain name으로 ip값을 얻어오는 방식이 JNDI 와 유사
   5)Database용 JNDI는 "jdbc/데이타베이스버전" 형식으로 지정

2. Lab
   1)server.xml

```
    <Context docBase="JNDIDemo" path="/JNDIDemo" reloadable="true" source="org.eclipse.jst.jee.server:JNDIDemo">
      <Resource auth="Container" driverClassName="oracle.jdbc.driver.OracleDriver" maxActive="10" maxIdle="10"
          maxWait="1000" name="jdbc/myoracle" password="tiger" type="javax.sql.DataSource"
          url="jdbc:oracle:thin:@localhost:1521:XE" username="scott"/>
    </Context>
```

   2)web.xml

```
    <resource-ref>
      <description>JNDI Test</description>
      <res-ref-name>jdbc/myoracle</res-ref-name>
      <res-type>javax.sql.DataSource</res-type>
      <res-auth>Container</res-auth>
    </resource-ref>
```

   3)src/com.example.dao.DBConnection.java

```
    package com.example.dao;

    import java.sql.Connection;
    import java.sql.SQLException;

    import javax.naming.Context;
    import javax.naming.InitialContext;
    import javax.naming.NamingException;
    import javax.sql.DataSource;

    public class DBConnection {
      public static Connection getConnection(){
        Connection conn = null;
        try {
          Context context = new InitialContext();
          Context env = (Context)context.lookup("java:comp/env");
          DataSource ds = (DataSource)env.lookup("jdbc/myoracle");
          conn = ds.getConnection();
        } catch (NamingException e) {
          System.out.println(e.getMessage());
        } catch(SQLException e){
          System.out.println(e.getMessage());
        }
        return conn;
    }
  }
```

4)Create Stored Procedure sp_select

```
  CREATE OR REPLACE PROCEDURE sp_select
  (
     v_deptno   IN    employees.department_id%TYPE,
     employee_records    OUT SYS_REFCURSOR
  )
  AS
  BEGIN
     OPEN employee_records FOR
     SELECT employee_id, first_name, salary,
            TO_CHAR(hire_date, 'YYYY-MM-DD') AS hiredate,
            department_name, city, e.department_id AS deptno
     FROM employees e INNER JOIN departments d ON e.department_id = d.department_id
                   INNER JOIN locations l ON d.location_id = l.location_id
     WHERE e.department_id = v_deptno;
  END;
  /
```

5)Build Path에 oracle driver 추가하기
- -project > right-click > Build Path > Configure Build Path...
- -Libraries tab > Click [Add External JARs...]
- -Select ojdbc6.jar > Click [Apply and Close]

6)WebContent/WEB-INF/lib에 jar file 추가
- -ojdbc6.jar
- -taglibs-standard-impl-1.2.5.jar
- -taglibs-standard-spec-1.2.5.jar

7)src/com.example.vo.EmployeeVO.java

```
  package com.example.vo;

  public class EmployeeVO {
    private int employee_id;
    private String first_name;
    private double salary;
    private String hiredate;
    private String department_name;
    private String city;
    private int departno;

    public EmployeeVO() {}

    public int getEmployee_id() {
      return employee_id;
    }

    public void setEmployee_id(int employee_id) {
      this.employee_id = employee_id;
    }

    public String getFirst_name() {
        return first_name;
    }

    public void setFirst_name(String first_name) {
        this.first_name = first_name;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public String getHiredate() {
        return hiredate;
    }

    public void setHiredate(String hiredate) {
        this.hiredate = hiredate;
    }

    public String getDepartment_name() {
        return department_name;
    }

    public void setDepartment_name(String department_name) {
        this.department_name = department_name;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public int getDepartno() {
        return departno;
    }

    public void setDepartno(int departno) {
        this.departno = departno;
    }

    @Override
    public String toString() {
        return "EmployeeVO [employee_id=" + employee_id + ", first_name=" + first_name + ", salary=" + salary
                + ", hiredate=" + hiredate + ", department_name=" + department_name + ", city=" + city
                + ", department_id=" + departno + "]";
    }

  }

8)src/com.example.dao.EmployeeDao.java
  package com.example.dao;

  import java.sql.CallableStatement;
  import java.sql.Connection;
  import java.sql.ResultSet;
  import java.sql.SQLException;
  import java.util.ArrayList;

  import com.example.vo.EmployeeVO;

  public class EmployeeDao {
    public static ArrayList<EmployeeVO> selectAll(int deptno) throws SQLException{
      ArrayList<EmployeeVO> list = new ArrayList<EmployeeVO>();
      Connection conn = DBConnection.getConnection();
      CallableStatement cstmt = conn.prepareCall("{ call sp_select(?, ?) }");
      cstmt.setInt(1, deptno);
      cstmt.registerOutParameter(2, oracle.jdbc.OracleTypes.CURSOR);
      cstmt.executeUpdate();
      ResultSet rs = (ResultSet)cstmt.getObject(2);
      while(rs.next()) {
        EmployeeVO emp = new EmployeeVO();
        emp.setEmployee_id(rs.getInt("employee_id"));
        emp.setFirst_name(rs.getString("first_name"));
        emp.setSalary(rs.getDouble("salary"));
        emp.setHiredate(rs.getString("hiredate"));
        emp.setDepartment_name(rs.getString("department_name"));
        emp.setCity(rs.getString("city"));
        emp.setDepartno(rs.getInt("deptno"));
        list.add(emp);
      }
      if(rs != null) rs.close();
      if(cstmt != null) cstmt.close();
      return list;
    }
  }

9)src/com.example.service.EmployeeService.java
  package com.example.service;

  import java.sql.SQLException;
  import java.util.ArrayList;

  import com.example.dao.EmployeeDao;
  import com.example.vo.EmployeeVO;

  public class EmployeeService {
    private int deptno;
    private ArrayList<EmployeeVO> list;

    public void setDeptno(int deptno) {
      this.deptno = deptno;
    }

    public ArrayList<EmployeeVO> getList() {
      ArrayList<EmployeeVO> list = null;
      try {
        list = EmployeeDao.selectAll(this.deptno);
      }catch(SQLException ex) {
        System.out.println(ex);
      }
      return list;
    }
  }
```

10) dbtest.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.ArrayList, com.example.vo.EmployeeVO" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<jsp:useBean id="service" class="com.example.service.EmployeeService" />
<c:set target="${service}" property="deptno" value="${empty param.deptno ? 10 : param.deptno}"/>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>사원명단</title>
  </head>
  <body>
    <h1>사원 명단(부서번호 : <c:out value="${empty param.deptno ? 10 : param.deptno}" />)</h1>
    <table border="1">
      <thead>
        <tr>
          <th>사원번호</th><th>사원이름</th><th>봉급</th>
          <th>입사일자</th><th>부서이름</th><th>부서위치</th>
          <th>부서번호</th>
        </tr>
      </thead>
      <tbody>
        <c:forEach items="${service.list}" var="emp">
        <tr>
          <td>${emp['employee_id']}</td><td>${emp['first_name']}</td><td>${emp.salary}</td>
          <td>${emp.hiredate}</td><td>${emp['department_name']}</td><td>${emp.city}</td>
          <td>${emp.departno}</td>
        </tr>
        </c:forEach>
      </tbody>
    </table>
  </body>
</html>
```