一、 数据库 test 新建表emp1

create table emp1(

empno int(4) PRIMARY KEY,

ename varchar(10),

job varchar(9),

hiredate date,

sal float(7,2)

)

二、实例数据

insert into emp1 values(1001,'张三-1','程序员-1','2010-5-4',1000);

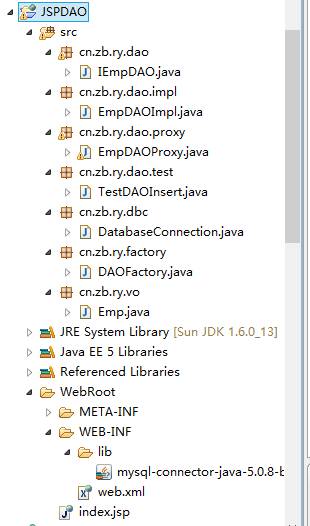
insert into emp1 values(1002,'张三-2','程序员-2','2010-5-4',1500);

insert into emp1 values(1003,'张三-3','程序员-3','2010-5-4',1500);

insert into emp1 values(1004,'张三-4','程序员-4','2010-5-4',2000);

三、DAO开发

**文件夹结构**



图：文件夹结构图

1. 定义vo类

package cn.zb.ry.vo;

import java.util.Date;

public class Emp {

private int empno;

private String ename;

private String job;

private Date hiredate;

private float sal;

对应属性setter、getter方法

。。。。。

}

2.数据库连接类——DatabaseConnection.java

package cn.zb.ry.dbc;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

public static final String DBDRIVER="com.mysql.jdbc.Driver";

public static final String DBURL="jdbc:mysql://localhost:3306/test";

public static final String DBUSER="root";

public static final String DBPASS="123456";

private Connection conn=null;

public DatabaseConnection(){

try {

Class.forName(DBDRIVER);

this.conn=DriverManager.getConnection(DBURL,DBUSER,DBPASS);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public Connection getConnection() {

// TODO Auto-generated method stub

return this.conn;

}

public void close() {

// TODO Auto-generated method stub

try {

this.conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

3.定义DAO操作标准 IEmpDAO.java

package cn.zb.ry.dao;

import java.util.List;

import cn.zb.ry.vo.Emp;

public **interface** IEmpDAO {

public boolean doCreate(Emp emp)throws Exception;

public List<Emp> findAll(String keyWord)throws Exception;

public Emp findById(int empno)throws Exception;

}

4.DAO实现类有两种，一种是真实主题实现类，另外一种是代理操作类，其中真实主题实现类负责具体的数据库操作，但没有数据库的打开和连接操作，只是通过构造方法取得了数据库的连接，而真正负责打开和关闭的操作将由代理来完成。

真实主题实现类——EmpDAOImpl.java

package cn.zb.ry.dao.impl;

import java.sql.\*;

import java.util.\*;

import cn.zb.ry.dao.IEmpDAO;

import cn.zb.ry.vo.Emp;

public class EmpDAOImpl implements IEmpDAO {

private Connection conn=null;

private PreparedStatement pstmt=null;

public EmpDAOImpl(Connection conn){

this.conn=conn;

}

public boolean **doCreate(Emp emp**) throws Exception {

// TODO Auto-generated method stub

boolean flag=false;

String sql="insert into emp1 values(?,?,?,?,?);";

this.pstmt=this.conn.prepareStatement(sql);

pstmt.setInt(1, emp.getEmpno());

pstmt.setString(2, emp.getEname());

pstmt.setString(3,emp.getJob());

pstmt.setDate(4,new java.sql.Date(emp.getHiredate().getTime()));

pstmt.setFloat(5, emp.getSal());

if(pstmt.executeUpdate()>0){

flag=true;

}

this.pstmt.close();

return flag;

}

public List<Emp> findAll(String keyWord) throws Exception {

// TODO Auto-generated method stub

List<Emp> all=new ArrayList<Emp>();

Emp emp=null;

String sql="select \* from emp1 where ename like ? or job like ?";

this.pstmt=this.conn.prepareStatement(sql);

pstmt.setString(1,"%"+keyWord+"%");

pstmt.setString(2,"%"+keyWord+"%");

ResultSet rs=this.pstmt.executeQuery();

while(rs.next()){

emp=new Emp();

emp.setEmpno(rs.getInt(1));

emp.setEname(rs.getString(2));

emp.setJob(rs.getString(3));

emp.setHiredate(rs.getDate(4));

emp.setSal(rs.getFloat(5));

all.add(emp);

}

this.pstmt.close();

return all;

}

public Emp findById(int empno) throws Exception {

// TODO Auto-generated method stub

Emp emp=null;

String sql="select \* from emp1 where empno=?";

this.pstmt=this.conn.prepareStatement(sql);

pstmt.setInt(1,empno);

ResultSet rs=this.pstmt.executeQuery();

while(rs.next()){

emp=new Emp();

emp.setEmpno(rs.getInt(1));

emp.setEname(rs.getString(2));

emp.setJob(rs.getString(3));

emp.setHiredate(rs.getDate(4));

emp.setSal(rs.getFloat(5));

}

this.pstmt.close();

return emp;

}

}

5. 代理操作类——EmpDAOProxy.java

package cn.zb.ry.dao.proxy;

import java.util.List;

import cn.zb.ry.dao.IEmpDAO;

import cn.zb.ry.dao.impl.EmpDAOImpl;

import cn.zb.ry.dbc.DatabaseConnection;

import cn.zb.ry.vo.Emp;

public class EmpDAOProxy implements IEmpDAO {

private DatabaseConnection dbc=null;

private IEmpDAO dao=null;

public EmpDAOProxy()throws Exception{

this.dbc=new DatabaseConnection();

this.dao=new EmpDAOImpl(this.dbc.getConnection());

}

public boolean doCreate(Emp emp) throws Exception {

// TODO Auto-generated method stub

boolean flag=false;

try {

if(this.dao.findById(emp.getEmpno())==null){

flag=this.dao.doCreate(emp);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return flag;

}

public List<Emp> findAll(String keyWord) throws Exception {

// TODO Auto-generated method stub

List<Emp> all=null;

try {

all=this.dao.findAll(keyWord);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

this.dbc.close();

}

return all;

}

public Emp findById(int empno) throws Exception {

// TODO Auto-generated method stub

Emp emp=null;

try {

emp=this.dao.findById(empno);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

this.dbc.close();

}

return emp;

}

}

5.DAO工厂类——DAOFactory.java

package cn.zb.ry.factory;

import cn.zb.ry.dao.IEmpDAO;

import cn.zb.ry.dao.proxy.EmpDAOProxy;

public class DAOFactory {

public static IEmpDAO getIEmpDAOInstance() throws Exception{

return new EmpDAOProxy();

}

}

6.测试DAO插入功能——TestDAOInsert

package cn.zb.ry.dao.test;

import cn.zb.ry.factory.DAOFactory;

import cn.zb.ry.vo.Emp;

public class TestDAOInsert {

public static void main(String[] args) throws Exception {

// TODO Auto-generated method stub

Emp emp=null;

for(int x=5;x<12;x++){

emp=new Emp();

emp.setEmpno(1000+x);

emp.setEname("张三-"+x);

emp.setJob("程序员-"+x);

emp.setHiredate(new java.util.Date());

emp.setSal(500\*x);

**DAOFactory.getIEmpDAOInstance().doCreate(emp);**

}

}

}



图：程序运行效果图

7. 测试DAO查询操作功能——TestDAOSelect.java

package cn.zb.ry.dao.test;

import java.util.\*;

import cn.zb.ry.factory.DAOFactory;

import cn.zb.ry.vo.Emp;

public class TestDAOSelect {

/\*\*

\* @param args

\*/

public static void main(String[] args) throws Exception{

// TODO Auto-generated method stub

List<Emp> all=DAOFactory.getIEmpDAOInstance().findAll("张三-1");

Iterator<Emp> iter=all.iterator();

while(iter.hasNext()){

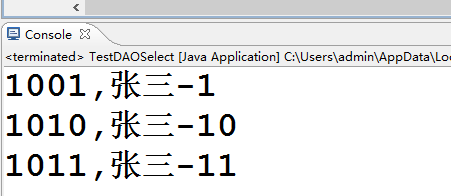
Emp emp=iter.next();

System.out.println(emp.getEmpno()+","+emp.getEname());

}

}

}



图：程序运行结果(后台输出)

8.JSP调用DAO

emp\_insert.jsp

<body>

<form action="emp\_insert\_do.jsp" method="post">

雇员编号：<input type="text" name="empno"><br>

雇员姓名：<input type="text" name="ename"><br>

雇员职位：<input type="text" name="job"><br>

雇佣日期：<input type="text" name="hiredate"><br>

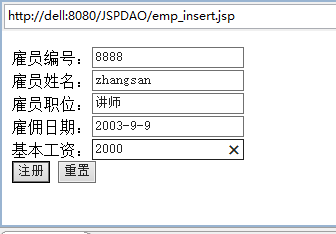
基本工资：<input type="text" name="sal"><br>

<input type="submit" value="注册">

<input type="reset" value="重置">

</form><br>

</body>



图：增加雇员表单

9. emp\_insert\_do.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="gb2312"%>

<%@ page import="cn.zb.ry.factory.\*,cn.zb.ry.vo.\*" %>

<%@ page import="java.text.\*"%>

<body>

<%

request.setCharacterEncoding("gb2312");

Emp emp=new Emp();

emp.setEmpno(Integer.parseInt(request.getParameter("empno")));

emp.setEname(request.getParameter("ename"));

emp.setJob(request.getParameter("job"));

emp.setHiredate(new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("hiredate")));

emp.setSal(Float.parseFloat(request.getParameter("sal")));

if(DAOFactory.getIEmpDAOInstance().doCreate(emp)){

%>

<h1>雇员信息添加成功！</h1>

<%

}else{

%>

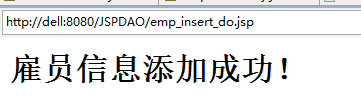
<h1>雇员信息添加失败！</h1>

<%

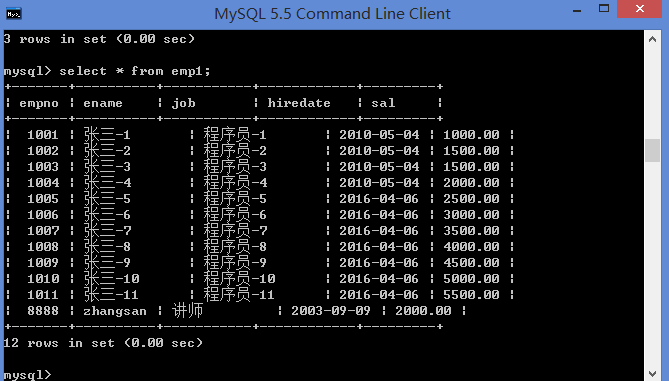
}

%> <br>

</body>



图：雇员信息增加成功页面



图：Mysql数据库emp1表中

10.注意添加的JSP文件目录结构

