# 实验三：Hibernate框架实验

# 项目建立

使用MyEclipse，在上一次实验项目的基础上，手工添加Hibernate框架以及编写相应的代码或者利用相关插件生成代码，并编写客户端，实现数据的基本CRUD功能。

# 客户类的编码实现

将设计的客户端类代码复制到下面的代码框中。

//自动生成的HibernateSessionFactory类 ，太长，而且大家都一样不复制了

学生表对应的 Student类

package cn.edu.scau.cmi.zhouxudong.dao;

/\*\*

\* Student entity. @author MyEclipse Persistence Tools

\*/

public class Student implements java.io.Serializable {

// Fields

private String id;

private String name;

private String sex;

private String class\_;

private String academy;

// Constructors

/\*\* default constructor \*/

public Student() {

}

/\*\* minimal constructor \*/

public Student(String id) {

this.id = id;

}

/\*\* full constructor \*/

public Student(String id, String name, String sex, String class\_,

String academy) {

this.id = id;

this.name = name;

this.sex = sex;

this.class\_ = class\_;

this.academy = academy;

}

// Property accessors

public String getId() {

return this.id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return this.name;

}

public void setName(String name) {

this.name = name;

}

public String getSex() {

return this.sex;

}

public void setSex(String sex) {

this.sex = sex;

}

public String getClass\_() {

return this.class\_;

}

public void setClass\_(String class\_) {

this.class\_ = class\_;

}

public String getAcademy() {

return this.academy;

}

public void setAcademy(String academy) {

this.academy = academy;

}

}

控制增删改查的类DBUtil

**package** cn.edu.scau.cmi.zhouxudong.util;

**import** java.util.ArrayList;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.Transaction;

**import** cn.edu.scau.cmi.zhouxudong.dao.HibernateSessionFactory;

**import** cn.edu.scau.cmi.zhouxudong.dao.Student;

**public** **class** DBUtil {

/\*插入数据\*/

**public** **void** insert(Student stu){

Session session=HibernateSessionFactory.*getSession*();//通过HibernateSessionFactory类的getSession方法获得Session对象

Transaction tx=session.beginTransaction();//开始事务

session.save(stu);

System.*out*.println("数据{ "+"学号：" + stu.getId() + " " + "姓名：" + stu.getName()+" "+ " 性别：" + stu.getSex()+" 班级："+stu.getClass\_()+" 学院："+stu.getAcademy()+" }已插入");

tx.commit();//提交事务

session.close();//关闭session

}

/\*根据hql语句查询\*/

**public** **void** squery(String hql){

Session session=HibernateSessionFactory.*getSession*();//通过HibernateSessionFactory类的getSession方法获得Session对象

Transaction tx=session.beginTransaction();//开始事务

**try** {

ArrayList list=(ArrayList) session.createQuery(hql).list();

} **catch** (Exception e) {

System.*out*.println("错误的hql语句: "+e.getMessage());

System.*exit*(0);

}

ArrayList list=(ArrayList) session.createQuery(hql).list();

System.*out*.println("查询结果如下：");

**for**(**int** i=0;i<list.size();i++){

Student student = (Student) list.get(i);

System.*out*.println("学号：" + student.getId() + " " + "姓名：" + student.getName()+" "+ "性别：" + student.getSex()+" 班级："+student.getClass\_()+" 学院："+student.getAcademy());

}

tx.commit();//提交事务

session.close();//关闭session

}

/\*更新\*/

**public** **void** update(String hql){

Session session=HibernateSessionFactory.*getSession*();//通过HibernateSessionFactory类的getSession方法获得Session对象

Transaction tx=session.beginTransaction();//开始事务

**try** {

Query query = session.createQuery(hql);

query.executeUpdate();

} **catch** (Exception e) {

System.*out*.println("错误的hql语句: "+e.getMessage());

System.*exit*(0);

}

System.*out*.println("更新成功！");

tx.commit();//提交事务

session.close();//关闭session

}

/\*删除根据学生id\*/

**public** **void** delete(String stuID){

Session session=HibernateSessionFactory.*getSession*();//通过HibernateSessionFactory类的getSession方法获得Session对象

Transaction tx=session.beginTransaction();//开始事务

String hql = "delete from Student where id=?";

**try** {

Query query = session.createQuery(hql);

query.setString(0, stuID);

query.executeUpdate();

} **catch** (Exception e) {

System.*out*.println("错误的hql语句: "+e.getMessage());

System.*exit*(0);

}

System.*out*.println("id为"+stuID+"的学生删除成功！");

tx.commit();//提交事务

session.close();//关闭session

}

}

客户端类Client 测试增删改查

**package** cn.edu.scau.cmi.zhouxudong.util;

**import** org.hibernate.engine.query.HQLQueryPlan;

**import** cn.edu.scau.cmi.zhouxudong.dao.Student;

**import** javassist.expr.NewArray;

**public** **class** Client {

**public** **static** **void** main(String[] args){

String hqlQuery = "from Student";

String hqlUpdate = "update Student s set s.name = '诺言' where id = '7777' ";

DBUtil db = **new** DBUtil();

Student stu = **new** Student("7777","clearlove","男","15计算机1班","计算机学院");

db.insert(stu);

//db.squery(hqlQuery);

//db.update(hqlUpdate);

// db.delete("7777");

db.squery(hqlQuery);

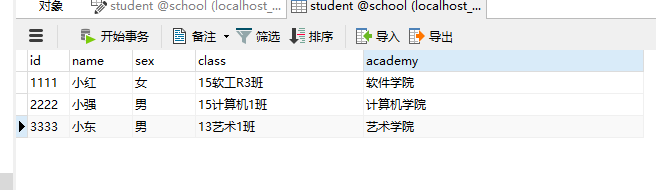
}

}

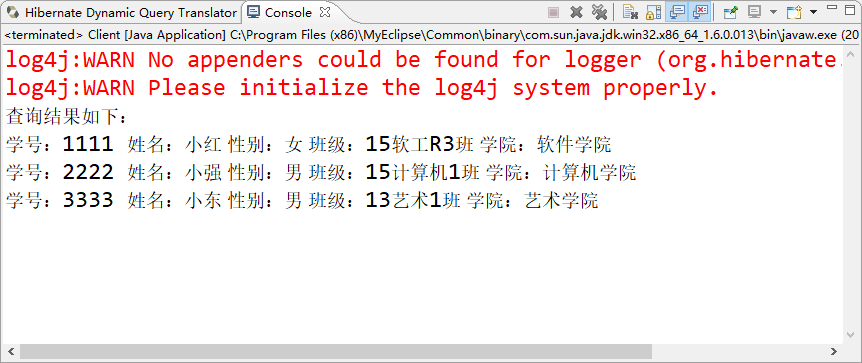
# 运行效果

截屏展示运行的效果，注意适当的说明和图的编号

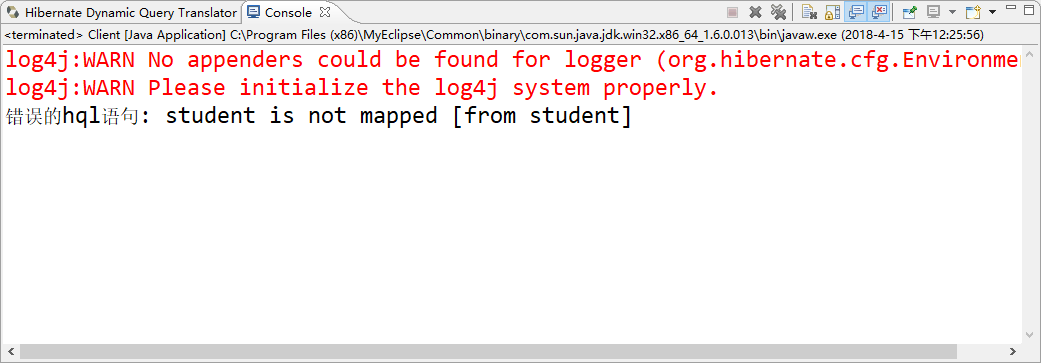
一开始数据库的数据有



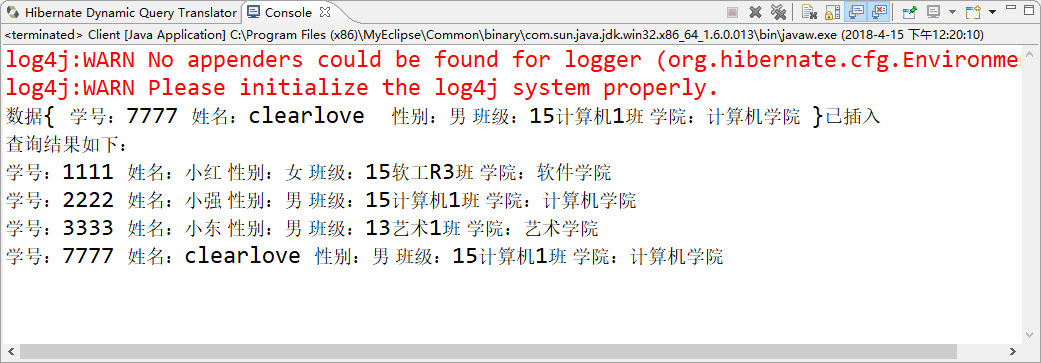
(1)查询数据库 db.squery(hqlQuery);



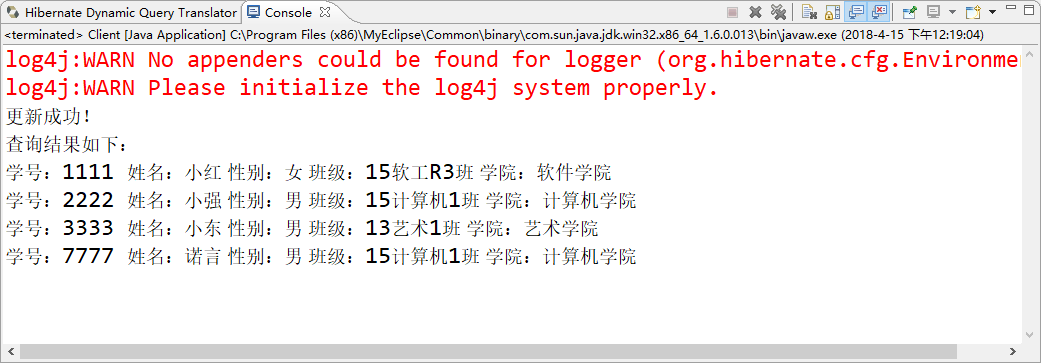
1. 异常抛出



1. 插入数据 db.insert(stu);



1. 更新数据 db.update(hqlUpdate);



1. 删除数据 db.delete("7777");

