**实验三： 动态SQL**

## 实验内容：

使用动态SQL进行条件查询、更新以及复杂查询操作。本实验要求利用本章所学知识完成一个学生信息系统，该系统要求实现3个以下功能:

1、多条件查询： 当用户输入的学生姓名不为空，则根据学生姓名进行学生信息的查询； 当用户输入的学生姓名为空，而学生专业不为空，则只根据学生专业进行学生的查询； 当用户输入的学生姓名不为空，且专业不为空，则根据姓名和专业查询学生信息。

当学生姓名和专业都为空，则查询所有学生信息

2、单条件查询：查询出所有id值小于5的学生的信息；

3、学生数据插入：插入三条学生的信息；

创建学生表(tb\_student)并输入3条记录，表结构如下:

CREATE TABLE tb\_student(

id int(32) PRIMARY KEY AUTO\_INCREMENT,

name varchar(50),

major varchar(50),

sno varchar(16) );

## 实验代码

mybatis-config.xml

*<?*xml version="1.0" encoding="UTF-8" *?>  
<!DOCTYPE* configuration  
 PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-config.dtd"*>  
  
<*configuration*>  
  
 <!-- 引入外部配置文件-->  
 <*properties resource="db.properties"*/>  
  
 <*settings*>  
 <*setting name="logImpl" value="LOG4J"*/>  
 <*setting name="mapUnderscoreToCamelCase" value="true"*/>  
 <!-- 开启二级缓存-->  
 <*setting name="cacheEnabled" value="true"*/>  
 </*settings*>  
  
 <!-- 可以给实体类起别名-->  
 <*typeAliases*>  
 <*package name="com.shf.pojo"*/>  
 </*typeAliases*>  
  
 <*environments default="development"*>  
 <*environment id="development"*>  
 <*transactionManager type="JDBC"*/>  
 <*dataSource type="POOLED"*>  
 <*property name="driver" value="${driver}"*/>  
 <*property name="url" value="${url}"*/>  
 <*property name="username" value="${username}"*/>  
 <*property name="password" value="${password}"*/>  
 </*dataSource*>  
 </*environment*>  
 </*environments*>  
  
 <!-- 每一个MapperXML都需要在Mybatis配置中心注册-->  
 <*mappers*>  
 <!-- <mapper resource="com/shf/dao/UserMapper.xml"/>-->  
 <!-- <mapper class="com.shf.dao.UserMapper"/>-->  
 <*package name="com.shf.mapper"*/>  
 </*mappers*>  
</*configuration*>*

tb\_student

@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class tb\_student *{* private int id;  
 private String name;  
 private String major;  
 private String sno;  
*}*

TbStudentMapper

package com.shf.mapper;  
  
  
import com.shf.pojo.tb\_student;  
  
import java.util.List;  
  
public interface TbStudentMapper *{  
 // 多条件查询* List*<*tb\_student*>* getStudentListByConditions*(*tb\_student tbstudent*)*;  
  
 *// 单条件查询* List*<*tb\_student*>* getStudentByLTId*(*Integer id*)*;  
  
 void addStudent*(*tb\_student tb\_student*)*;  
*}*

TbStudentMapper.xml

*<?*xml version="1.0" encoding="UTF-8" *?>  
<!DOCTYPE* mapper  
 PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-mapper.dtd"*>  
  
<*mapper namespace="com.shf.mapper.TbStudentMapper"*>*

*<!-- 开启二级缓存-->  
<*cache*/>*

*<*select id="getStudentListByConditions" parameterType="tb\_student" resultType="tb\_student"*>* select \*  
 from tb\_student  
 *<*where*>  
 <*if test="name!=null and name!=''"*>* and name=#{name}  
 *</*if*>  
 <*if test="major!=null and major!=''"*>* and major=#{major}  
 *</*if*>  
 <*if test="sno!=null and sno!=''"*>* and sno=#{sno}  
 *</*if*>  
 </*where*>  
 </*select*>  
  
 <*select id="getStudentByLTId" parameterType="int" resultType="tb\_student"*>* select \*  
 from tb\_student  
 where id &lt; #{id};  
 *</*select*>  
  
 <*sql id="key"*>  
 <*trim suffixOverrides=","*>  
 <*if test="id!=0 and id!=''"*>* id,  
 *</*if*>  
 <*if test="name!=null and name!=''"*>* name,  
 *</*if*>  
 <*if test="major!=null and major!=''"*>* major,  
 *</*if*>  
 <*if test="sno!=null and sno!=''"*>* sno,  
 *</*if*>  
 </*trim*>  
 </*sql*>  
 <*sql id="value"*>  
 <*trim suffixOverrides=","*>  
 <*if test="id!=0 and id!=''"*>* #{id},  
 *</*if*>  
 <*if test="name!=null and name!=''"*>* #{name},  
 *</*if*>  
 <*if test="major!=null and major!=''"*>* #{major},  
 *</*if*>  
 <*if test="sno!=null and sno!=''"*>* #{sno},  
 *</*if*>  
 </*trim*>  
 </*sql*>  
 <*insert id="addStudent" parameterType="tb\_student"*>* insert into tb\_student (*<*include refid="key"*/>)* values *(<*include refid="value"*/>)*;  
 *</*insert*>  
</*mapper*>*

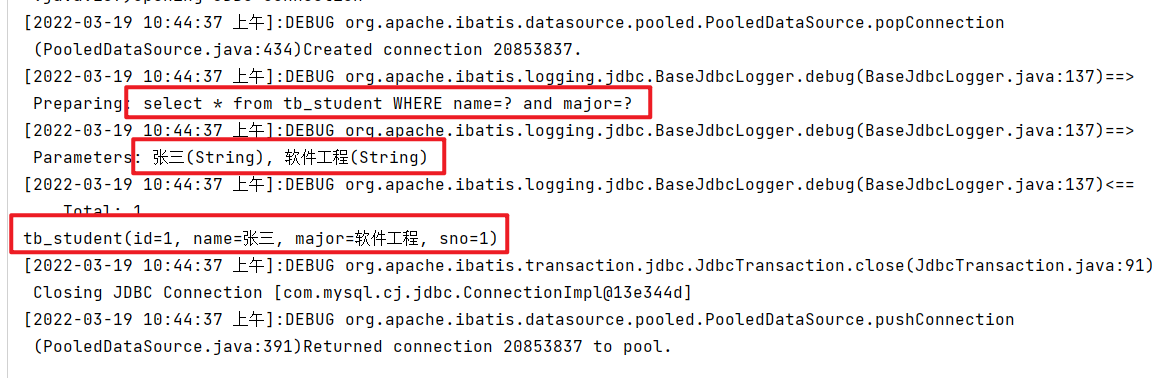
MyBatisUtils.java

package com.shf.utils;  
  
import org.apache.ibatis.io.Resources;  
import org.apache.ibatis.session.SqlSession;  
import org.apache.ibatis.session.SqlSessionFactory;  
import org.apache.ibatis.session.SqlSessionFactoryBuilder;  
  
import java.io.IOException;  
import java.io.InputStream;  
  
public class MyBatisUtils *{* private static SqlSessionFactory *sqlSessionFactory*;  
  
 static *{* try *{* String resource = "mybatis-config.xml";  
 InputStream inputStream = Resources.*getResourceAsStream(*resource*)*;  
 *sqlSessionFactory* = new SqlSessionFactoryBuilder*()*.build*(*inputStream*)*;  
 *}* catch *(*IOException e*) {* e.printStackTrace*()*;  
 *}  
 }* public static SqlSession getSqlSession*() {  
// 自动提交事务* return *sqlSessionFactory*.openSession*(*true*)*;  
 *}  
}*

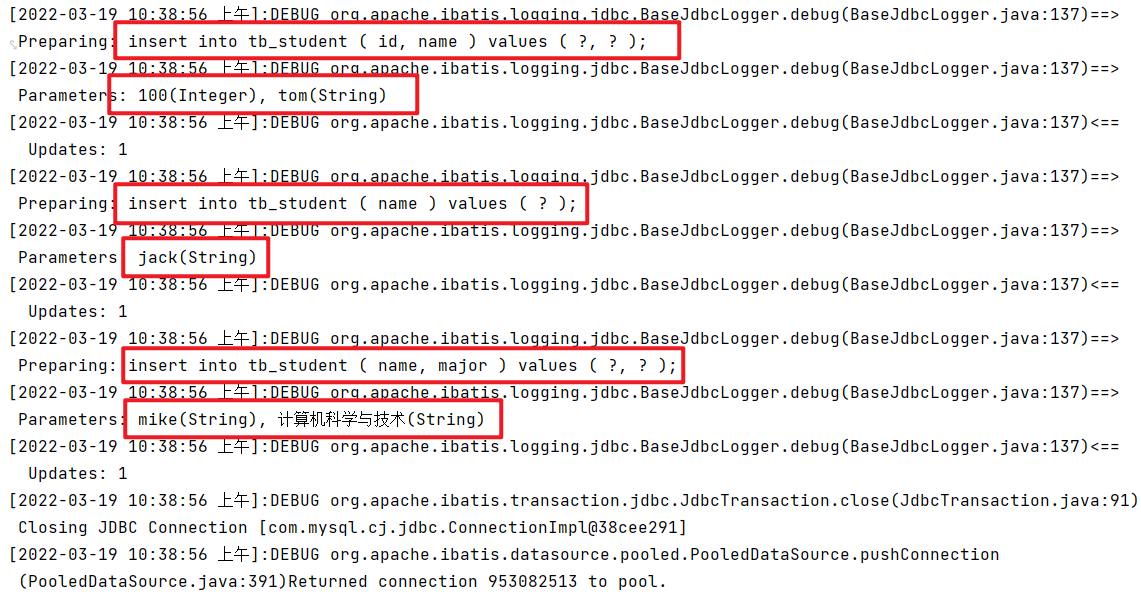
tbstudentTest.java

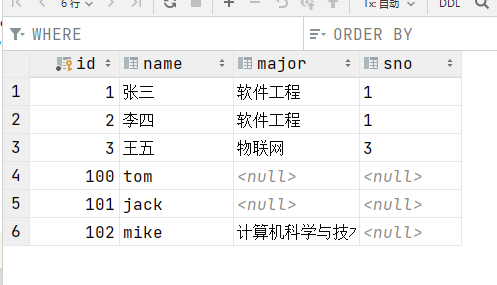
package com.shf.test;  
  
import com.shf.mapper.TbStudentMapper;  
import com.shf.pojo.tb\_student;  
import com.shf.utils.MyBatisUtils;  
import org.apache.ibatis.session.SqlSession;  
import org.junit.Test;  
  
import java.util.List;  
  
public class tbstudentTest *{* @Test  
 public void getStudentListByConditions*() {* SqlSession sqlSession = MyBatisUtils.*getSqlSession()*;  
 TbStudentMapper mapper = sqlSession.getMapper*(*TbStudentMapper.class*)*;  
  
 tb\_student tbstudent = new tb\_student*()*;  
 tbstudent.setName*(*"张三"*)*;  
 tbstudent.setMajor*(*"软件工程"*)*;  
 List*<*tb\_student*>* tbstudentListByConditions = mapper.getStudentListByConditions*(*tbstudent*)*;  
 tbstudentListByConditions.forEach*(*System.*out*::println*)*;  
  
 sqlSession.close*()*;  
 *}* @Test  
 public void getStudentById*() {* SqlSession sqlSession = MyBatisUtils.*getSqlSession()*;  
 TbStudentMapper mapper = sqlSession.getMapper*(*TbStudentMapper.class*)*;  
 List*<*tb\_student*>* studentById = mapper.getStudentByLTId*(*5*)*;  
 studentById.forEach*(*System.*out*::println*)*;  
  
 sqlSession.close*()*;  
 *}* @Test  
 public void addStudent*() {* SqlSession sqlSession = MyBatisUtils.*getSqlSession()*;  
 TbStudentMapper mapper = sqlSession.getMapper*(*TbStudentMapper.class*)*;  
  
 tb\_student student1 = new tb\_student*()*;  
 student1.setId*(*100*)*;  
 student1.setName*(*"tom"*)*;  
 mapper.addStudent*(*student1*)*;  
  
 tb\_student student2 = new tb\_student*()*;  
 student2.setName*(*"jack"*)*;  
 mapper.addStudent*(*student2*)*;  
  
 tb\_student student3 = new tb\_student*()*;  
 student3.setName*(*"mike"*)*;  
 student3.setMajor*(*"计算机科学与技术"*)*;  
 mapper.addStudent*(*student3*)*;  
  
 sqlSession.close*()*;  
 *}  
}*

## 实验截图：









## 实验小结：

通过本次实验我学会使用mybatis中的动态sql，学习了if、choose（when、otherwise）、trim（where，set）、foreach；以及sql片段，使用include标签进行引入。