# 实验四 关联关系映射

## 实验目的

1. 了解数据表之间以及对象之间的三种关联关系；
2. 掌握一对一、一对多、多对多的关联关系映射的配置。

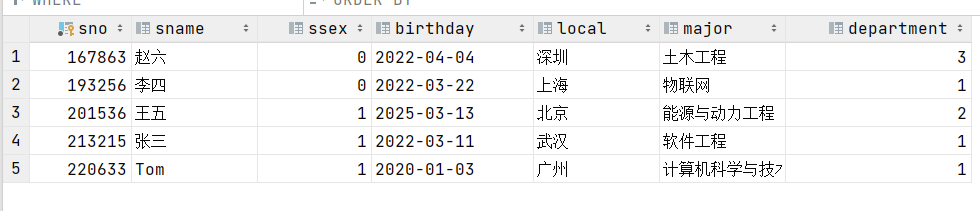
## 实验内容

建学生数据库，创建院系表和学生信息表，院系表中存储院系编号，院系名称，学生信息表中存储学生学号、姓名、性别、出生日期、籍贯、专业、所属院系等信息，在2个表中输入若干数据。创建Java web项目，配置一对多的关联关系，插入一个学生数据，并且查询所有计算机学院的学生。

## 实验代码：

学生表

create table studentsy4  
*(* sno int not null  
 primary key,  
 sname varchar*(*100*)* null,  
 ssex int null,  
 birthday date null,  
 local varchar*(*100*)* null,  
 major varchar*(*100*)* null,  
 department int null,  
 constraint studentsy4\_sno\_uindex  
 unique *(*sno*)  
)*;



院系表

create table departmentsy4  
*(* dno int auto\_increment  
 primary key,  
 dname varchar*(*100*)* null,  
 constraint departmentsy4\_dno\_uindex  
 unique *(*dno*)  
)*;



院系实体类

package com.shf.pojo;  
  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import java.io.Serializable;  
import java.util.List;  
  
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class DepartmentSy4 implements Serializable *{* private Integer dno;  
 private String dname;  
  
 private List*<*StudentSy4*>* students;  
*}*

学生表

package com.shf.pojo;  
  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
import java.io.Serializable;  
import java.util.Date;  
  
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class StudentSy4 implements Serializable *{* private Integer sno;  
 private String sname;  
 private Integer ssex;  
 private Date birthday;  
 private String local;  
 private String major;  
 private Integer department;  
*}*

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*>*

StudentSy4Mapper.java

package com.shf.mapper;  
  
import com.shf.pojo.StudentSy4;  
  
public interface StudentSy4Mapper *{* void addStudent*(*StudentSy4 studentSy4*)*;  
*}*

DepartmentSy4Mapper.java

package com.shf.mapper;  
  
import com.shf.pojo.DepartmentSy4;  
  
import java.util.List;  
  
public interface DepartmentSy4Mapper *{* List*<*DepartmentSy4*>* getDepartment*(*int id*)*;  
*}*

StudentSy4Mapper.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.StudentSy4Mapper"*>  
 <!-- 开启二级缓存-->  
 <*cache*/>  
  
 <!-- 添加-->  
 <*sql id="key"*>  
 <*trim suffixOverrides=","*>  
 <*if test="sno!=0 and sno!=''"*>* sno,  
 *</*if*>  
 <*if test="sname!=null and sname!=''"*>* sname,  
 *</*if*>  
 <*if test="ssex!=0 and ssex!=''"*>* ssex,  
 *</*if*>  
 <*if test="birthday!=null"*>* birthday,  
 *</*if*>  
 <*if test="local!=null and local!=''"*>* local,  
 *</*if*>  
 <*if test="major!=null and major!=''"*>* major,  
 *</*if*>  
 <*if test="department!=null and department!=''"*>* department,  
 *</*if*>  
 </*trim*>  
 </*sql*>  
 <*sql id="values"*>  
 <*trim suffixOverrides=","*>  
 <*if test="sno!=0 and sno!=''"*>* #{sno},  
 *</*if*>  
 <*if test="sname!=null and sname!=''"*>* #{sname},  
 *</*if*>  
 <*if test="ssex!=0 and ssex!=''"*>* #{ssex},  
 *</*if*>  
 <*if test="birthday!=null"*>* #{birthday},  
 *</*if*>  
 <*if test="local!=null and local!=''"*>* #{local},  
 *</*if*>  
 <*if test="major!=null and major!=''"*>* #{major},  
 *</*if*>  
 <*if test="department!=null and department!=''"*>* #{department},  
 *</*if*>  
 </*trim*>  
 </*sql*>  
 <*insert id="addStudent" parameterType="studentSy4"*>* insert into studentsy4 (*<*include refid="key"*/>)* values *(<*include refid="values"*/>)*;  
 *</*insert*>  
  
  
</*mapper*>*

DepartmentSy4Mapper.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.DepartmentSy4Mapper"*>  
 <!-- 开启二级缓存-->  
 <*cache*/>  
  
 <*select id="getDepartment" resultMap="StudentAndDepartmentList" parameterType="int"*>* select \*  
 from studentsy4,  
 departmentsy4  
 where studentsy4.department = departmentsy4.dno  
 and dno = #{dno};  
 *</*select*>  
  
 <*resultMap id="StudentAndDepartmentList" type="departmentSy4"*>  
 <*id column="dno" property="dno"*/>  
 <*result column="dname" property="dname"*/>  
 <*collection property="students" ofType="studentSy4"*>  
 <*id property="sno" column="sno"*/>  
 <*result property="sname" column="sname"*/>  
 <*result property="ssex" column="ssex"*/>  
 <*result property="birthday" column="birthday"*/>  
 <*result property="local" column="local"*/>  
 <*result property="major" column="major"*/>  
 <*result property="department" column="department"*/>  
 </*collection*>  
 </*resultMap*>  
</*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*)*;  
 *}  
}*

DepartmentSy4MapperTest.java

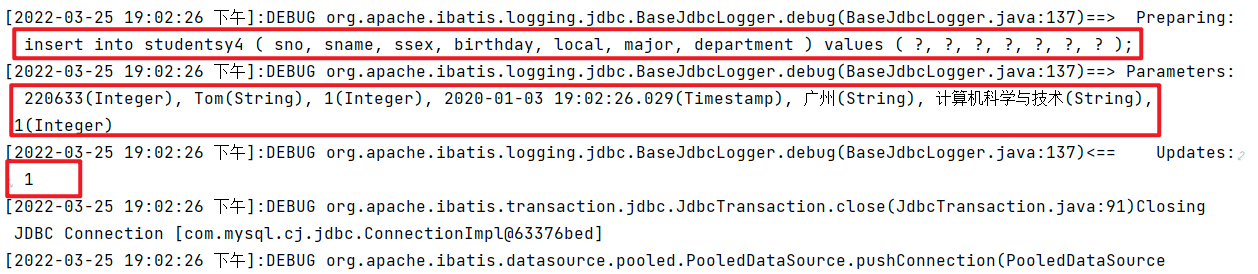
package com.shf.mapper;  
  
import com.shf.pojo.DepartmentSy4;  
import com.shf.utils.MyBatisUtils;  
import org.apache.ibatis.session.SqlSession;  
import org.junit.Test;  
  
import java.util.List;  
  
public class DepartmentSy4MapperTest *{* @Test  
 public void getStudent*() {* SqlSession sqlSession = MyBatisUtils.*getSqlSession()*;  
 DepartmentSy4Mapper mapper = sqlSession.getMapper*(*DepartmentSy4Mapper.class*)*;  
 List*<*DepartmentSy4*>* department = mapper.getDepartment*(*1*)*;  
 department.forEach*(*System.*out*::println*)*;  
  
 sqlSession.close*()*;  
 *}  
}*

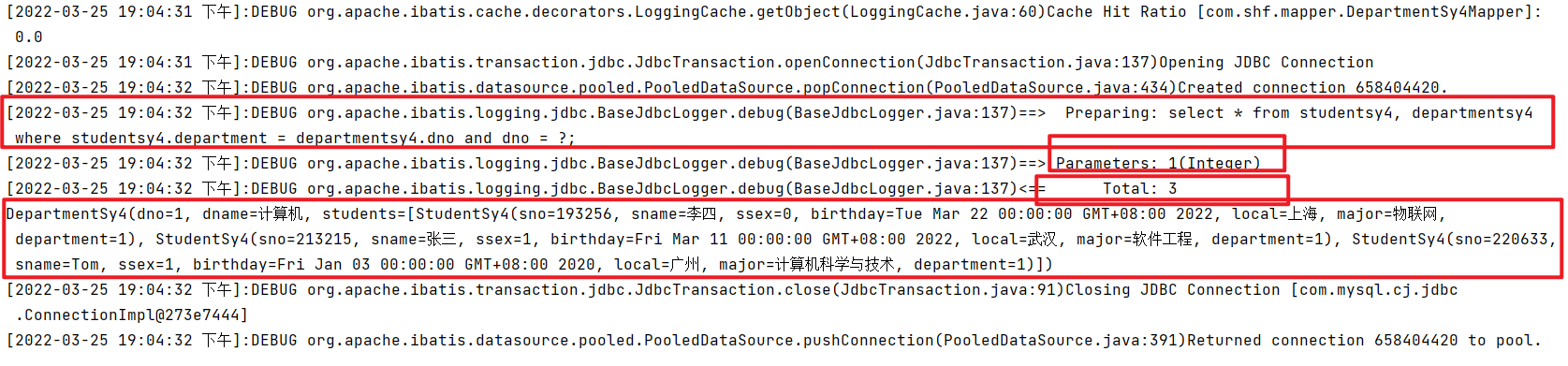
StudentSy4MapperTest.java

package com.shf.mapper;  
  
import com.shf.pojo.StudentSy4;  
import com.shf.utils.MyBatisUtils;  
import org.apache.ibatis.session.SqlSession;  
import org.junit.Test;  
  
import java.util.Calendar;  
import java.util.Date;  
  
  
public class StudentSy4MapperTest *{* @Test  
 public void addStudent*() {* SqlSession sqlSession = MyBatisUtils.*getSqlSession()*;  
 StudentSy4Mapper mapper = sqlSession.getMapper*(*StudentSy4Mapper.class*)*;  
  
 Calendar cal = Calendar.*getInstance()*;  
 cal.set*(*Calendar.*YEAR*, 2020*)*;  
 cal.set*(*Calendar.*MONTH*, Calendar.*JANUARY)*;  
 cal.set*(*Calendar.*DAY\_OF\_MONTH*, 3*)*;  
 Date dateRepresentation = cal.getTime*()*;  
 StudentSy4 studentSy4 = new StudentSy4*(* 220633,  
 "Tom",  
 1,  
 dateRepresentation,  
 "广州",  
 "计算机科学与技术",  
 1  
 *)*;  
 mapper.addStudent*(*studentSy4*)*;  
  
 sqlSession.close*()*;  
 *}  
}*

## 运行截图

插入一个学生数据

查询所有计算机学院的学生



# 实验小结

通过这次我学会了使用mybatis的关联映射，掌握一对一关系、一对多关系和多对多关系的使用方法。