1. Mybatis

基本特点：

MyBatis 是支持普通 SQL 查询，存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的XML或注解用于配置和原始映射，将接口和 Java 的 POJOs（Plan Old Java Objects，普通的 Java对象）映射成数据库中的记录。

优点：

1、配置灵活，部署简单。

2、SQL语句可以集中管理。

3、持久层结构清晰明了，对应关系且一一对应。

4、灵活解决数据库设计是分表结构。

环境搭建：

1. 注意点

1）、数据查询：

2）、数据新增：

进行数据插入是如果需要返回对应的ID 主键，则必须要如下配置，结合oracle本身的特性（字段没有自动增长的功能），因此我们通常会设置对应的序列和触发器作用于某个字段上

配置如下：

<!-- 新增记录 -->

<insertid=*"insertXjTeacher"*useGeneratedKeys=*"false"*>

<selectKeyresultType=*"long"*order=*"AFTER"*keyProperty=*"xjTeacher.id"*>

SELECT XJ\_TEACHER\_SEQ.currval FROM DUAL

</selectKey>

INSERT INTO XJ\_TEACHER (cdata,birthday,psw,userid,amass\_phone)

VALUES(#{xjTeacher.cdata},#{xjTeacher.birthday}，#{xjTeacher.schoolId})

</insert>

注意，在代码可以直接返回数据成功插入后的ID值。由于数据库本身是采用触发器，因此是配置“XJ\_TEACHER\_SEQ.currval”而不是“XJ\_TEACHER\_SEQ.nextval”。

3）、数据更新：

4）、数据删除：

5）、事务控制代码样例：

/\*\*

\* 事务控制代码

\* @throws Exception

\*/

public void onAddByManualTransactionalProc() throws Exception {

// TODO Auto-generated method stub

DefaultTransactionDefinition def = new DefaultTransactionDefinition();

def.setPropagationBehavior(TransactionDefinition.PROPAGATION\_REQUIRED);

DataSourceTransactionManager txManager = (DataSourceTransactionManager) SpringUtil.getSpringBean("transactionManager");

TransactionStatus status = txManager.getTransaction(def);

try{

TsXjStudentEntry tsXjStudentEntry = new TsXjStudentEntry();

tsXjStudentEntry.setId(this.getId());

tsXjStudentEntry.setStuSequence(this.getStuSequence());

tsXjStudentEntry.setStuNo(this.getStuNo());

tsXjStudentEntry.setName(this.getName());

tsXjStudentEntry.setSex(this.getSex());

TsXjStudentMapper tsXjStudentMapper = (TsXjStudentMapper) SpringUtil.getSpringBean("tsXjStudentMapper");

tsXjStudentMapper.deleteTsXjStudent(this.getDaoAbb(), new String[]{"622210"});

System.out.println("删除成功");

tsXjStudentMapper.insertTsXjStudent(this.getDaoAbb(),tsXjStudentEntry);

System.out.println("新增成功"+tsXjStudentEntry.getId());

tsXjStudentMapper = null;

CsXjStudentInfo csXjStudentInfo = (CsXjStudentInfo) SpringUtil.getSpringBean("csXjStudentInfo");

csXjStudentInfo.setDaoAbb("cs");

csXjStudentInfo.setName("csXjStudentInfoName");

csXjStudentInfo.setSex(1);

csXjStudentInfo.setStuNo(this.getStuNo());

csXjStudentInfo.setStuSequence("csXjStudentInfoSeq");

csXjStudentInfo.add();

System.out.println("csXjStudentInfo新增成功");

csXjStudentInfo = null;

}catch(Exception ex){

txManager.rollback(status);

throw ex;

}

txManager.commit(status);

txManager = null;

def = null;

}

6）、数据批量操作设计：

<!-- 在外部for循环调用一千次 -->  
<insert id="insert" parameterType="sdc.mybatis.test.Student">  
insert into student (id, name, sex,address, telephone, t\_id)  
 values (#{id,jdbcType=INTEGER}, #{name,jdbcType=VARCHAR},  
#{sex,jdbcType=VARCHAR},#{address,jdbcType=VARCHAR},#{telephone,jdbcType=VARCHAR},#{tId,jdbcType=INTEGER})  
</insert>

<!-- 批量，传入一个长度为1000的list -->  
<insert id="insertBatch">  
 insert into student ( <include refid="Base\_Column\_List"/> )   
 values   
<foreach collection="list" item="item" index="index" separator=",">  
(null,#{item.name},#{item.sex},#{item.address},#{item.telephone},#{item.tId})  
</foreach>  
</insert>

1. 开发目录

教师前台web：svn://192.168.1.104:36900/svn/xxt/xxt2\_zj/trunk/04newxxt/webApps/Teacher

业务逻辑模块：svn://192.168.1.104:36900/svn/xxt/xxt2\_zj/trunk/04newxxt/XxtBusinessProject

CoolTool 代码工具的连接地址

内网：<http://zsqtsm.vicp.net:43980/code/>

外网：<http://192.168.4.39:80/code/>