知识回顾：

1. 三大框架
   1. 旧三大框架：struts2+spring+hibernate
   2. 新三大框架: springmvc+spring+mybatis

大多二三线城市主流开发框架： SSM

一线城市主流开发框架：微服务dubbo+SpringBoot+SpringClound

大数据一定学。

1. 持久层框架mybatis，必然操作数据库

Hibernate持久层框架，ORM框架，解决就是数据库记录集合对象自动转换，它基于HQL，面向对象语句，类SQL。全面面向对象的持久层框架。

Mybatis，半ORM映射框架，解决数据库记录集和对象自动转换，它基于SQL，后半截是面向过程编程方式，前半截面向对象编程方式。

业界：hibernate创建中小型项目，mybatis构建大型项目的。Mybatis轻量封装sql实现，所以它的性能高。

1. 创建mybatis项目，核心组件
   1. sqlMapConfig.xml核心配置文件，环境（测试、开发、运行）、事务（提交、回滚）、数据源（4大参数）、全局配置（别名）
   2. SqlSessionFactory工厂设计模式，特性：线程安全，共享成员变量（初学者写高并发程序时就别写成员变量）
   3. SqlSession对象，特点：线程非安全，必须写在方法内部，局域变量。Mybatis就提供一系列操作数据库的方法。CRUD selectList（可以多条，可以一条，没有）/selectOne（只能有一个值）/insert/update/delete
   4. Mybatis事务，默认开启，手动提交。
2. 创建mybatis查询，user查询所有
   1. 创建数据库mybatisdb，必须设置字符集utf8，防止中文乱码。Dos环境 set names utf8;默认提交环境不支持中文。
   2. 创建Maven项目，两种骨架，设置字符集，设置jdk版本1.7,1.8
   3. 配置核心文件sqlMapConfig.xml，mybatis-config.xml放轻，spring框架整合mybatis，事务、数据源都消失，spring中配置，注解替代，springboot替代
   4. 创建SqlSessionFactoryBuilder.build(is) factory对象
   5. factory.openSession创建SqlSession对象
   6. sqlSession.selectList(ns.id) ns命名空间。作者定义，ns以.分割。ns找到唯一的映射文件，id找这个映射文件中的唯一的方法。
   7. 映射文件，每个操作会有一个标签<select><insert><update><delete>
   8. 参数：parameterType/parameterMap（废除）支持java：java.lang.Integer；自己预定义很多类型int/long/string/date…。复杂参数：User pojo对象，map（参数少，时间段。。。）
   9. 返回值：resultType 简单变量（int/string/map/list/pojo）/resultMap（关联查询，多表查询-mybatis中最强大对象）
   10. 带参，调用方法要调用2个参数，第二个参数就是业务参数值。在映射文件的方法中增加resultType/resultMap/类型设置：int/string。。。如果需要对象变量必须map/pojo

今日任务：

1. 动态SQL语句（mybatis独特发明）

SELECT id,NAME,birthday,address FROM USER

SELECT NAME FROM USER

INSERT INTO USER VALUES(1,2,3,4)

INSERT INTO USER (NAME) VALUES(1)

SQL是写在映射文件中，事先要写好。无法做SQL的优化！

让这些SQL语句，例如字段变成动态的，业务需要几个，就写几个？

DELETE FROM USER WHERE id = 2

DELETE FROM USER WHERE id = 3

DELETE FROM USER WHERE id = 4

DELETE FROM USER WHERE id IN (2,3,4)

SELECT \* FROM USER WHERE AND address='111'

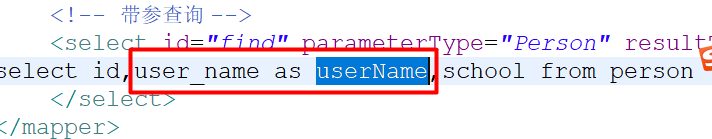
1. 简化编程，映射

数据库字段：user\_name，java属性userName（驼峰规则）ResultMap

1. 数据库4个链接参数写成属性配置文件，如何加载？
2. selectList/selectOne(sqlSession) 不是面向对象设计，User.find(),User.get()，mybatis面向接口开发UserMapper.java 主流开发方式（接口最后还是调用selectList）



Mybatis映射不是表而是结果集！



SELECT id,user\_name AS userName,school FROM person

WHERE user\_name='tony' AND school = '北大'

3种情况：

1. 查询user\_name是tony和school是北大
2. 查询user\_name是tony
3. 查询school是北大

动态SQL语句

1. If

<if test="school != null"> AND school = #{school}</if>

判断如果结果true，就出现标签中的内容；如果结果false，不出现标签中的内容

1. Where

<where>

<if test="userName != null">user\_name=#{userName}</if>

<if test="school != null"> AND school = #{school}</if>

</where>

Where标签会自动去掉最前面的and，or

另外一种技巧：

Where 1=1

1. set

<set>

<if test="userName != null">user\_name=#{userName}, </if>

<if test="school != null">school=#{school}</if>

</set>

4）foreach

DELETE FROM person WHERE id IN (

<foreach collection="ids" item="id" separator=",">

#{id}

</foreach>

)

Collection集合，map方式，ids就是key；

Item代表循环时实例名称 id

Separator 多个值之间分隔符

#{id} 获取循环中每次实例名称

DELETE FROM person WHERE id IN

<foreach collection="ids" item="id" separator="," open="(" close=")">

#{id}

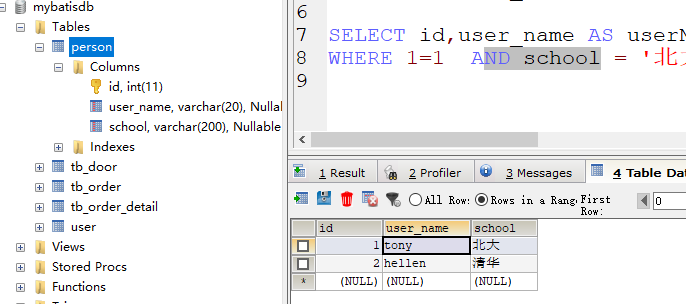
</foreach>

Open就代表开始符号

Close就代表结束符号

练习：

1. 创建person表



1. 创建Person Maven项目
2. 实现person表查询

当java pojo属性和结果集的字段名称不一致时，要配置一个resultMap

<!-- 构建复杂映射关系,id唯一，type返回值放入哪个pojo -->

<resultMap type="Person" id="personRM">

<!-- 主键,property pojo的属性， column数据库表的字段 -->

<id property="id" column="id"/>

<!-- 普通字段，起到了映射规则：数据库结果集字段user\_name，内部会自动调用person.setUserName方法 -->

<result property="userName" column="user\_name"/>

<result property="school" column="school"/>

</resultMap>

<!-- 带参查询 -->

<select id="find" parameterType="Person" resultMap="personRM">

SELECT id,user\_name,school FROM person

where 1=1

<if test="userName != null">user\_name=#{userName}</if>

<if test="school != null"> and school = #{school}</if>

</select>

核心配置文件中标签是有顺序性的：

元素类型为 "configuration" 的内容必须匹配

"(properties?,settings?,typeAliases?,typeHandlers?,objectFactory?,objectWrapperFactory?,plugins?,environments?,databaseIdProvider?,mappers?)"

增加属性文件，jdbc.properties（数据库连接配置）

1. 编写一个jdbc.properties属性文件

driver=com.mysql.jdbc.Driver

url=jdbc:mysql://localhost:3306/mybatisdb?characterEncoding=utf8

username=root

password=root

1. 在核心配置文件中加载这个属性文件sqlMapConfig.xml

<!-- 加载属性文件 -->

<properties resource="jdbc.properties"/>

1. 引用属性文件中的值

<property name="driver" value="${driver}"/>

<property name="url" value="${url}"/>

<property name="username" value="${username}"/>

<property name="password" value="${password}"/>

接口开发

Mybatis早期提供方法selectList/selectOne这种方式不是面向对象思考方式。

推出接口式开发

开发步骤

1. 创建一个接口(Map接口有实现类HashMap)，无需写实现类
2. sqlSession对象来调用这个接口

注意路径：

1. 保证接口文件放在和映射文件同一目录（约定）
2. 命名空间指向映射文件的目录
3. 在全局配置文件中告诉mybatis接口存在，不能同时存在接口和xml方式

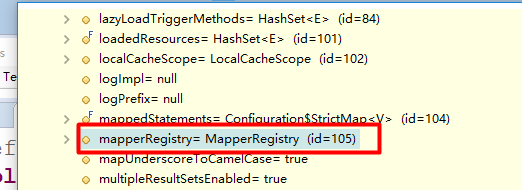
<mappers>

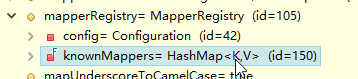
<!-- mapper resource="cn/tedu/person/pojo/PersonMapper.xml"/-->

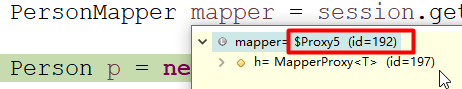
<!-- 全局限定名=包名.类/接口名称 -->

<mapper class="cn.tedu.person.pojo.PersonMapper"/>

</mappers>







$ProxyN 结构，这是一个代理对象，（jdk动态代理实现）

技术：动态代理技术：

1. jdk动态代理
2. cglib动态代理

动态根据接口来创建实现类

接口实际上最终调用的还是xml中的方法

