1. MyBatis介绍---特点
2. 开源的优秀的持久层框架，是轻量级框架------侧重于减小开发的复杂度、不带有侵略性api、不依赖容器、易配置、不消耗太多的内存cpu、启动快、事务功能弱、不具有分布式处理能力

拓展------（1）重量级框架：强调高可伸缩性，带有侵略性api、依赖容器、不具有配置能力、不同的产品部署过程不同不易通用、消耗很多内存cpu、

（2）衡量框架的重轻量：看消耗内存cpu的多少，或者说启动时所需的资源

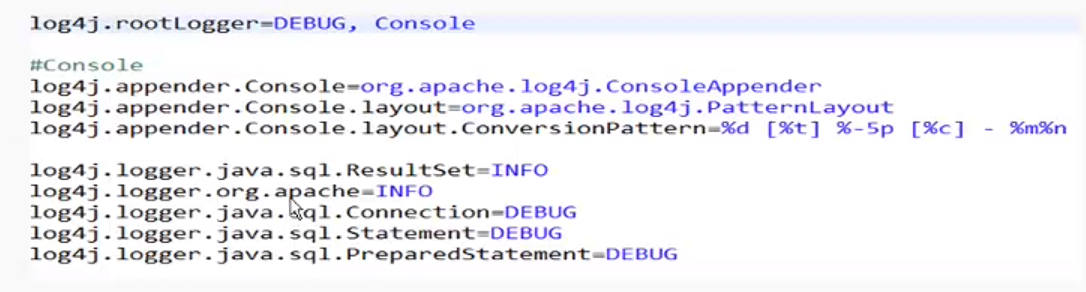
1. sql语句与java代码完全分离，易维护，方便读
2. 是面向配置的编程

拓展------面向过程🡪面向对象🡪面向配置（例子：日志记录，mybatis框架。能够降低耦合性，能够使可维护性、可扩展性大大增强）

1. 良好支持复杂的数据映射
2. 动态sql，更安全更方面（避免了拼装sql语句）
3. MyBatis环境搭建
4. 下载地址：https://github.com/mybatis/mybatis-3/releases
5. 添加jar包



1. 配合日志
2. 原因：提供功能sql语句的输出，方便调试
3. 加入日志文件：log4j.properties
4. 改写日志输出级别：在mybatis中sql语句必须在debug级别才能输出



1. MaBatis工作流程
2. 读取配置文件🡪生成sqlSessionFactory🡪建立sqlSession🡪调用MaBatis提供功能API🡪查询MAP配置🡪返回结果🡪关闭sqlSession
3. 生命周期

SqlSessionFactory:程序级，只有一个，程序开始建立，程序结束消亡

建立：sqlSessionFactory sqlMapper=new

SqlSession:过程级的，可有多个，一个方法结束则关闭

1. map文件(mybatis是sql代码有java代码分离，依托map文件，map文件包含所有的sql代码)

相对路径引用：

<mappers>

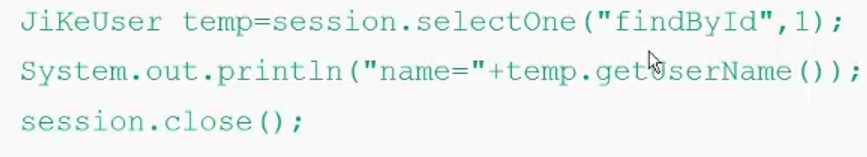
<mapper resource=”.xml” />

</mappers>

绝对路径引用：<mapper url= “file://...xml”/>

包路径引用：<package name=””/>

1. 建立msql数据库，建表，建实体类
2. 查询代码



1. 小试一把
2. 创建数据库、表；配置日志
3. 写相应的实体类
4. 根据实体类和表，写mapper.xml

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace=*"com.ss.mybatis.dao.UserDaos"*>

1. mybatis-config

<?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>

<environments default=*"dev"*>

<environment id=*"dev"*>

<transactionManager type=*"JDBC"*/>

<dataSource type=*"POOLED"*>

<property name=*"driver"* value=*"com.mysql.jdbc.Driver"*/>

<property name=*"url"* value=*"jdbc:mysql://127.0.0.1:3306/test?useUnicode=true&amp;characterEncoding=utf-8"*/>

<property name=*"username"* value=*"root"*/>

<property name=*"password"* value=*"111111"*/>

</dataSource>

</environment>

</environments>

<mappers>

<mapper resource=*"com/ss/mybatis/dao/UserDao.xml"*/>

</mappers>

</configuration>

五、MyBatis基础操作

1. id找到映射唯一标识；

parmeterType,传入sql的参数类型（TypeAliases标记定义别名，在parameterType中就可以不用写实体类的全路径）；

statementType=”PREPARED”；预处理

keyProperty:指定主键、主键定义

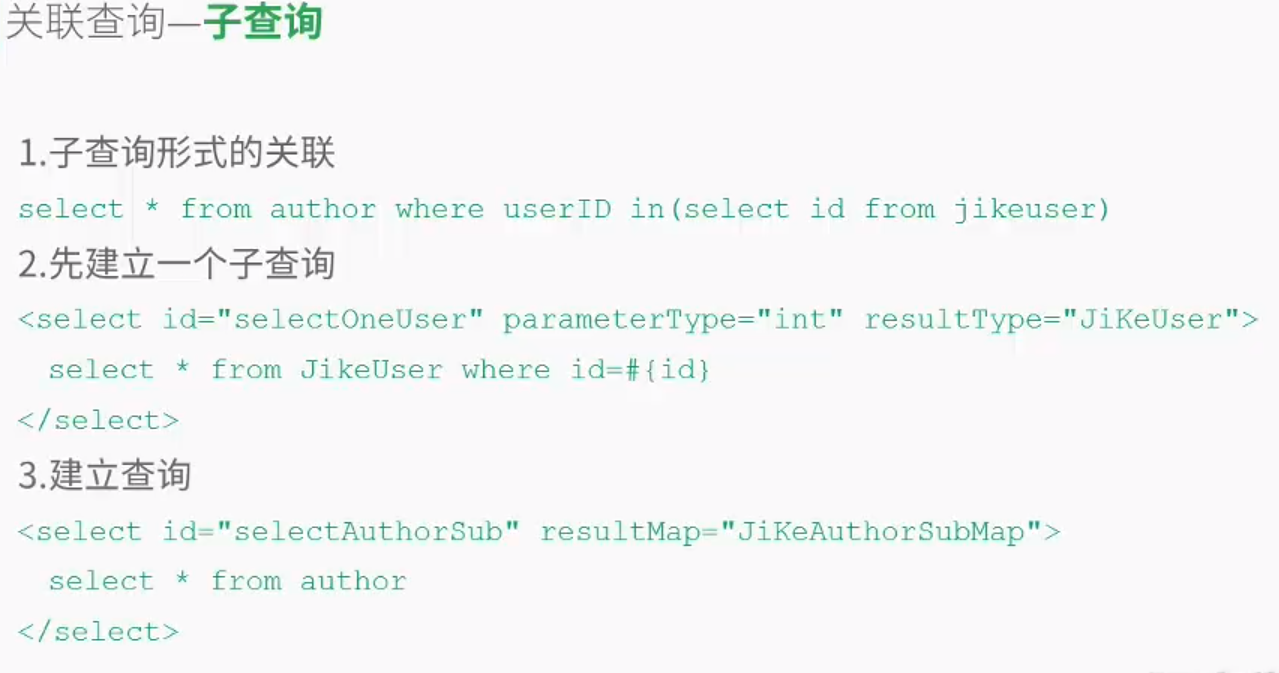
useGeneratedKeys:主键自增

#{username,jdbcType=varchar}:自动数据类型转换，常不用

1. 查询

（1）联合查询：一次查询，占用资源大，较子查询推荐

（2）子查询：N+1次查询，耗时，占用资源可大可小

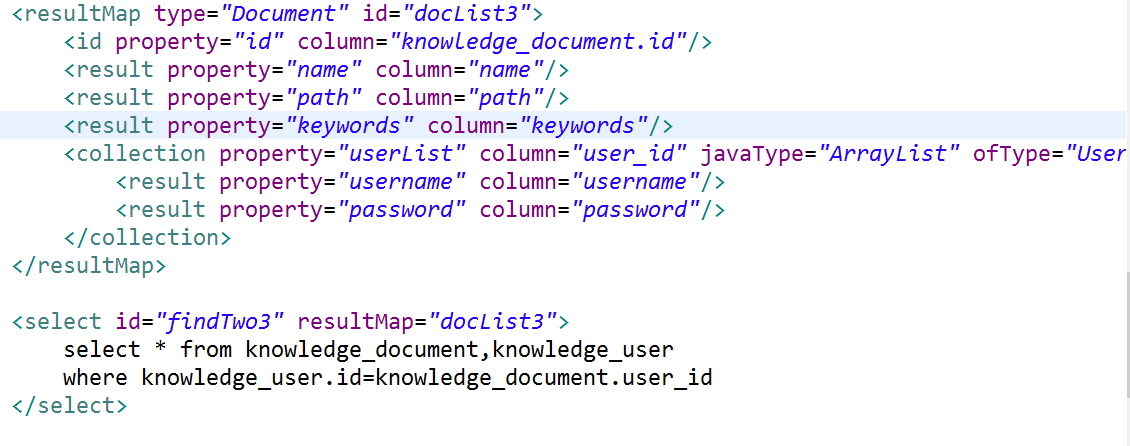


懒加载应用，更推荐：用到时才加载，非懒加载一次执行加载

（3）集合查询:

定义：在数学中可以对集合做交并差运算，在 SQL 中同样可以对查询结果集做交并差操作

应用：用于当查询的对象当中含有另外一个对象的集合引用的时候使用。



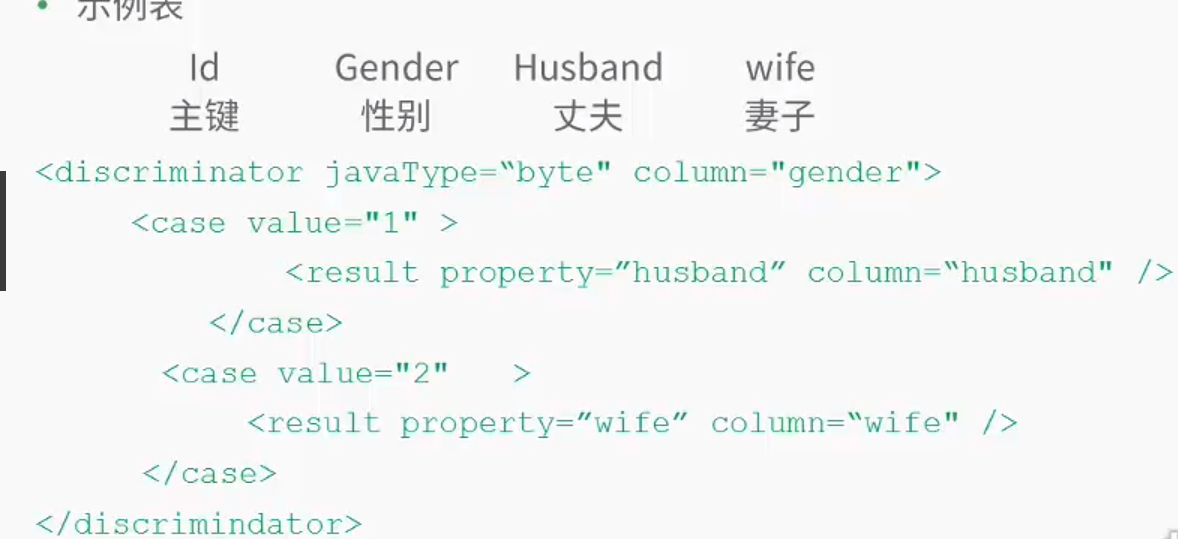
六、MyBatis鉴别器

鉴别器discriminator标记：

javaType:鉴别器对应字段的类型

column:字段

case子标记：根据字段的值执行不同的操作



七、实战思路

1、单独配置：

(1)、sqlSessionFactoryUtil中

读取mabatis-config.xml的配置：

（Reader, InputStream）: Resources.getResourceAsStream(CONFIG);

创建sqlSessionFactory工厂:SqlSessionFactoryBuilder

判断sqlSession是否开启了：ThreadLocal<SqlSession> local=new …🡪local.get()

打开sqlSession: sqlSessionFactory.openSession()🡪可放置新的sqlSession:local.set(sqlSession);

设置mapper映射器：public static <T> T getMapper(Class<T> t)

(2)、测试使用

通过xml映射dao层接口，拿到接口，传入相关的参数条件等直接调用使用

2、与spring的整合

八、事务

1、READ\_UNCOMMITTED：未提交读，当前事务可以读到另一个事务未提交的（INSERT未提交）数据（脏读）

2、READ\_COMMITTED，已提交读，当前事务只可以读另一个事务已提交的数据（避免脏读,但出现不可重复读，即同一事务中可以读取不同的数据，这些数据被另一事务UPDATE或DELETE）

3、REPEATABLE\_READ，可重复读，避免脏读、不可重复读，但出现幻读（同一事务中，相同的检索条件，读取到新的数据，即另一事务INSERT并提交的数据）

4、SERIALIZABLE，串行化，事务按顺序执行，防止脏读、不可重复读、幻读

脏读：当前事务可以读到另一个事务未提交的（INSERT未提交）数据

不可重复读：同一事务中读取到了UPDATE或DELETE已经提交的数据

幻读：同一事务中，相同的检索条件，读取到新的数据，即另一事务INSERT并提交的数据