**1、本章面试题**

            什么是Mybatis？

mybatis 工作原理

**2、知识点**

**2.1、课程回顾**

                maven工作原理

maven常用命令

**2.2、本章重点**

         1, ORM 思想

2, mybatis 原理

3, mybatis 配置详解

**3、具体内容**

**3.1 ORM思想**

对象关系映射（Object Relational Mapping，简称ORM） ORM是通过使用描述对象和数据库之间映射的元数据，元数据一般采用XML格式，并且存放在专门的对象一映射文件中。将程序中的对象自动持久化到关系数据库中。

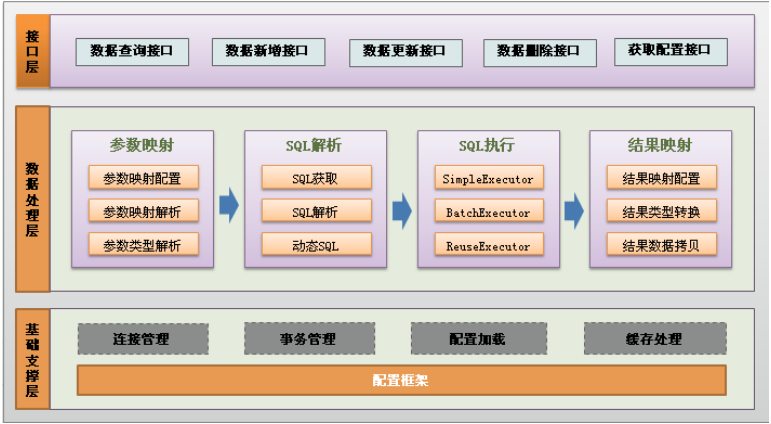
**3.2 mybatis简介**

**mybatis官网：**

[**https://mybatis.org/mybatis-3/zh/getting-started.html**](https://mybatis.org/mybatis-3/zh/getting-started.html)

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生类型、接口和 Java 的 POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。

**3.3 结构图**



架构分为三层：

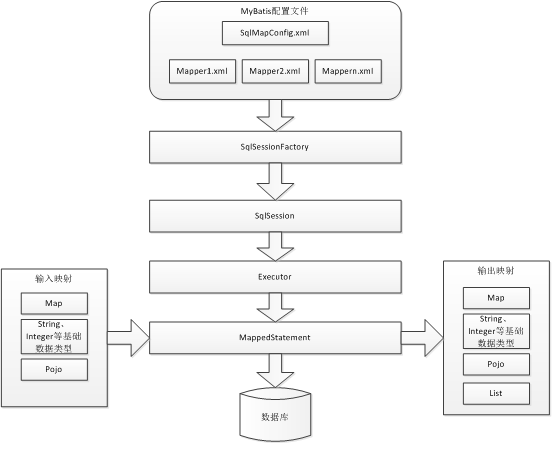
接口层：给应用程序提供一系列的数据接口。（CRUD) ;SqlSession数据处理层：

接口层传递参数，sql命令，在数据处理层进行处理，返回对应的结果映射。

基础支撑层：提供最基础的底层的操作：连接管理（连接池），事务管理（增、删、改），配置加载

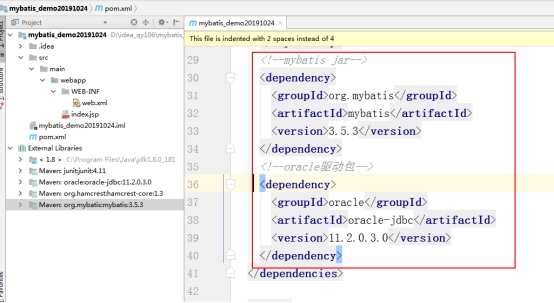
（读取配置信息），缓存（一级缓存，二级缓存）。

**3.4 mybatis 工作原理**

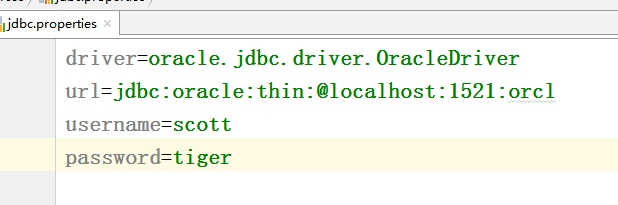


**3.5 mybatis 快速入门（完成增删改查操作）**

**3.4.1 创建项目引入jar**

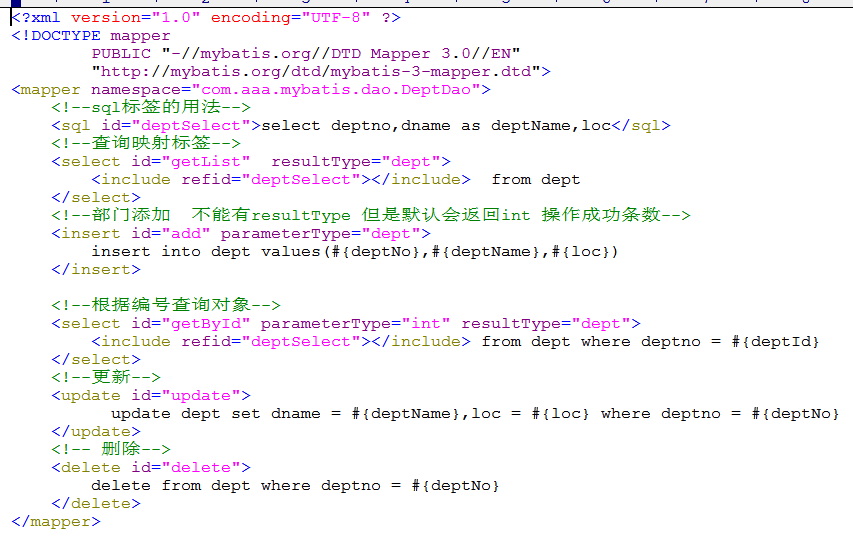


**3.4.2 创建mybatis主配置文件，创建数据源配置文件**





**3.4.3 创建mybatis Mapper文件**



**3.4.4 创建Mapper文件中用到的实体及接口，完善Mapper文件**



**3.4.5 创建编写SqlSessionFactoryUtil类**

加载主配置

生成SqlSessionFactory

生成sqlSession



**3.4.6 编写测试类**









**3. 6 关键类介绍:**

**SqlSessionFactoryBuilder**

这个类可以被实例化、使用和丢弃，一旦创建了 SqlSessionFactory，就不再需要它了。 因此 SqlSessionFactoryBuilder 实例的最佳作用域是方法作用域（也就是局部方法变量）。 你可以重用 SqlSessionFactoryBuilder 来创建多个 SqlSessionFactory 实例，但是最好还是不要让其一直存在，以保证所有的 XML 解析资源可以被释放给更重要的事情。

**SqlSessionFactory**

SqlSessionFactory 一旦被创建就应该在应用的运行期间一直存在，没有任何理由丢弃它或重新创建另一个实例。 使用 SqlSessionFactory 的最佳实践是在应用运行期间不要重复创建多次，多次重建 SqlSessionFactory 被视为一种代码“坏味道（bad smell）”。因此 SqlSessionFactory 的最佳作用域是应用作用域。 有很多方法可以做到，最简单的就是使用单例模式或者静态单例模式。

**SqlSession**

每个线程都应该有它自己的 SqlSession 实例。SqlSession 的实例不是线程安全的，因此是不能被共享的，所以它的最佳的作用域是请求或方法作用域。 绝对不能将 SqlSession 实例的引用放在一个类的静态域，甚至一个类的实例变量也不行。 也绝不能将 SqlSession 实例的引用放在任何类型的托管作用域中，比如 Servlet 框架中的 HttpSession。 如果你现在正在使用一种 Web 框架，要考虑 SqlSession 放在一个和 HTTP 请求对象相似的作用域中。 换句话说，每次收到的 HTTP 请求，就可以打开一个 SqlSession，返回一个响应，就关闭它。 这个关闭操作是很重要的，你应该把这个关闭操作放到 finally 块中以确保每次都能执行关闭。

**3.5 mybatis 配置文件详解**

**3.5.1 主配置详解：**

<?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>

<!--属性-->

<properties resource="jdbc.properties"></properties>

<!--全局的配置-->

<settings>

<setting name="cacheEnabled" value="true"/>

<setting name="logImpl" value="STDOUT\_LOGGING" />

</settings>

<!--类型别名-->

<typeAliases>

<package name="com.aaa.entity" />

</typeAliases>

<!--环境：数据库，-->

<environments default="development">

<!--单个的数据库配置-->

<environment id="development">

<!--事务管理器-->

<transactionManager type="JDBC"/>

<!--数据源：连接池技术-->

<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>

<mappers>

<mapper resource="com/aaa/dao/DeptMapper.xml"/>

</mappers>

</configuration>

**3.5.2 mapper文件详解：**

  可配置标签：

cache – 对给定命名空间的缓存配置。

cache-ref – 对其他命名空间缓存配置的引用（已经不用了）。

resultMap – 是最复杂也是最强大的元素，用来描述如何从数据库结果集中来加载对象。

parameterMap – 已被废弃！老式风格的参数映射。更好的办法是使用内联参数，此元素可能在将来被移除。文档中不会介绍此元素。

sql – 可被其他语句引用的可重用语句块。

insert – 映射插入语句

update – 映射更新语句

delete – 映射删除语句

select – 映射查询语句

标签的属性描述：

1）id：命名空间中唯一的标识符，可以来引用该条语句 也是接口的方法名称

2）parameterType：这里设置传入该条语句的参数类的完全限定名或者普通java类型

3）resultType：返回结果的类的完全限定名或者普通java类型

4）resultMap：外部resultMap的命名引用，结果集的映射是Mybatis最强大的特性

5）flushCache：如果设置为true,则任何时候该语句被调用，都会导致本地缓存和二级缓存都被清空。默认是false

6）useCache：如果设置为true，将会导致本条语句的结果被二级缓存。默认是true

7）timeout：抛出异常之前，驱动程序等待数据库返回请求结果的秒数

**4、本章总结**

**4.1 总结本章知识点**

**4.2 面试题答案**

**4.3 预习下一章重点**