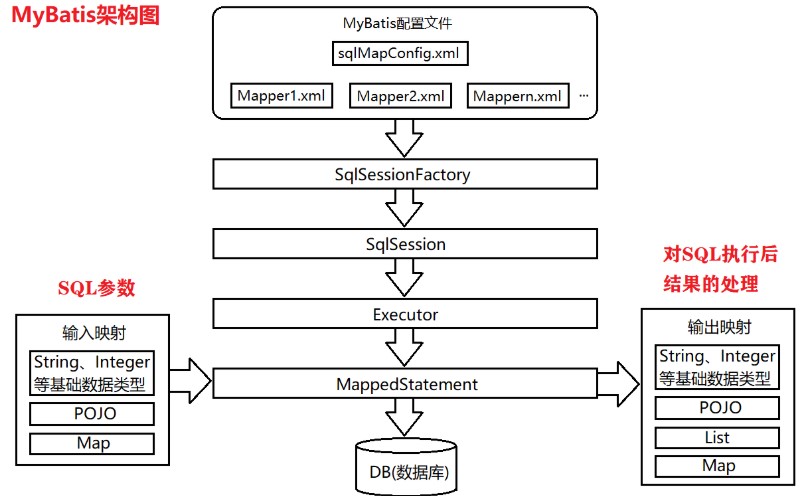
[持久层(数据层,Dao层) MyBatis框架](https://www.cnblogs.com/jingweixioabai/p/13221162.html)

**所谓的持久层就是把数据可以永久保持的存储到设备中,不像放到内存中那样断电就消失,一般来说,持久层为直接的理解就是对数据库的各种操作,如CRUD(增加,删除,修改,查询),更新等操作**

**持久层,就是把持久的动作封装成一个独立的层,这是为了降低功能代码之间的关联.创建一个更清晰的抽象,提高代码的内聚力,降低代码的耦合度,提高可维护性和复用性.**

MyBatis是一个优秀的持久层框架，它对jdbc的操作数据库的过程进行封装，使开发者只需要关注SQL本身，而不需要花费精力去处理例如注册驱动、创建connection、创建statement、手动设置参数、结果集检索等jdbc繁杂的过程代码。

Mybatis通过xml或注解的方式将要执行的各种statement（statement、preparedStatemnt）配置起来，并通过java对象和statement中的sql进行映射生成最终执行的sql语句，最后由mybatis框架执行sql并将结果映射成java对象并返回。



(1)mybatis-config.xml是Mybatis的核心配置文件，通过其中的配置可以生成SqlSessionFactory,也就是SqlSession工厂

(2)基于SqlSessionFactory可以生成SqlSession对象

(3)SqlSession是一个既可以发送SQL去执行，并返回结果，类似于JDBC中的Connection对象，也是Mybatis中至关重要的一个对象。

(4)Executor是SqlSession底层的对象，用于执行SQL语句

(5)MapperStatement对象也是SqlSession底层的对象，用于接收输入映射（SQL语句中的参数），以及做输出映射（即将SQL查询的结果映射成相应的结果）

总之，Mybatis对JDBC访问数据库的过程进行了封装，简化了JDBC代码，解决JDBC将结果集封装为Java对象的麻烦。

MyBatis是一个优秀的持久层框架，它对jdbc的操作数据库的过程进行封装，使开发者只需要关注SQL本身，而不需要花费精力去处理例如注册驱动、创建connection、创建statement、手动设置参数、结果集检索等jdbc繁杂的过程代码。

Mybatis通过xml或注解的方式将要执行的各种statement（statement、preparedStatemnt）配置起来，并通过java对象和statement中的sql进行映射生成最终执行的sql语句，最后由mybatis框架执行sql并将结果映射成java对象并返回。

总之，Mybatis对JDBC访问数据库的过程进行了封装，简化了JDBC代码，解决JDBC将结果集封装为Java对象的麻烦。