# Mybatis 技术内幕

# Mybatis快速入门

## 1.2 常见的持久化框架

HIbernate

Hibernate除了能够实现对象模型与关系模型的映射，还可以帮组开发人员屏蔽不同的数据库产品中SQL语句的细微差异。开发人员不再直接编写SQL语句，只需要使用Hibernate封装了数据库层面的全部操作，开发人员不再直接编写SQL语句，只要使用HIbernate提供简单易懂的API即可完成数据库的操作。例如，Hibernate提供的Criteria是一个完全面向对象、可扩展的条件查询API，使用它查询数据库时完全不需要考虑数据库底层如何实现、SQL语句如何编写。除了Criteria，Hibernate还提供了一种称为HQL的语言，它是一种面向对象的查询语言。

另外，HIbernate的API没有入侵性，业务逻辑不需要继承HIbernate的任何接口。Hibernate默认提供了一级缓存和二级缓存，有利于提高系统性能，降低数据库压力。其他优点：支持透明的持久化、延迟加载、由对象模型自动生层数据库表等。

Hibernate的缺点：并不是数据库中所有的概念都能在面向对象中找到合适的映射。例如，索引、存储过程、函数等，尤其是索引，它地数据库传的性能帮助很大。我们很难修改Hibernate生成的SQL语句，当数据库数量比较大、数据库结构比较复杂时，HIbernate生成SQL语句会非常复杂，让生成的SQL正确的使用索引比较困难，这就会导致出现大量满查询的情况。在有些大数据量、高并发、低延迟的场景下，HIbernate并不是特别适合。HIbernate对批量处理支持并不是很友好。

JPA

JPA仅仅是一个持久化的规范，它并没有提供具体的实现。其他持久化厂商提供JPA规范具体实现，例如，HIbernate，EclipseLink等都提供了JPA规范的具体实现。

Spring JDBC

Spring JDBC并不能算是一个ORM框架，它仅仅是使用末班方式对原生JDBC进行了一层非常薄的封装。使用Spring JDBC可以帮助开发人员屏蔽创建数据库连接对、Statement独享、异常处理以及事务管理的重复性代码。

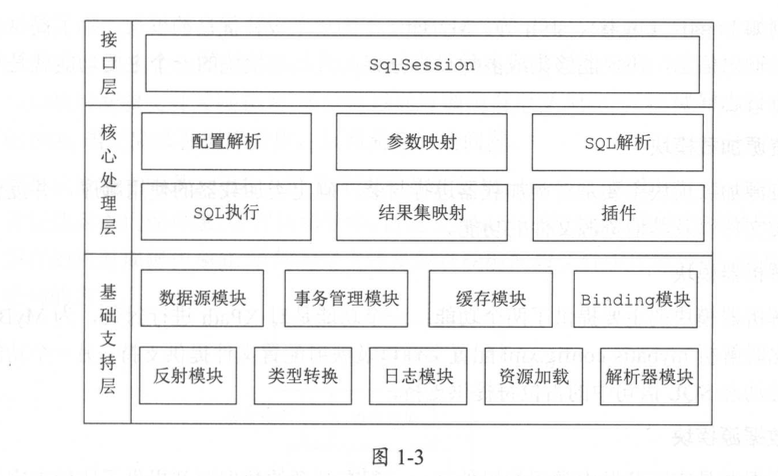
Spring JDBC中没有映射文件、对象查询语言、缓存等概念，而是直接执行原生SQL语句。Spring JDBC中提供了多种Template类，可以将对象中的属性映射成SQL中语句中绑定参数。

Mybatis

相相交于HIbernate，Mybatis更加轻量级，可控性也更高，在适应Mybatis时我们直接在映射配置文件中编写待执行的原生SQL语句，这就给了我们直接优化SQL语句的机会。

## 1.4 Mybatis整体架构

MyBatis的整体架构分为三层，分别是基础支持层、核心处理层和接口层。



### 1.4.1 基础支持层

**反射模块**

该模块对Java原生的反射进行了良好的封装，停工了更加简洁易用的API，方便上层使用，并且对反射操作进行了一些列优化，例如缓存了类的元数据，提高了反射操作的性能。

**类型转换模块**

Mybatis为了监护配置文件提供了别名机制，该机制是类型装换模块的主要功能之一。类型转换模块的另一个功能是实现JDBC类型与java类型之间的装换，该功能在为SQL语句绑定实参以及映射查询结果集时都会被涉及。

**日志模块**

**资源加载模块**

主要是对类加载器进行封装，确定类加载器的使用顺序，并提供了加载类文件以及其他资源文件的功能

**解析器模块**

主要提供了两个功能：一个功能是对XPath进行封装，为Mybatis初始化解析mybatis-config.xml配置文件以及映射配置文件提供支持；另外一个功能是为处理动态SQL语句中的占位符提供支持

**数据源模块**

**事务管理**

在很多场景中，Mybatis会与Spring框架集成，并由Spring框架管理事务

**缓存模块**

mybatis中提供了一级缓存和二级缓存，而这两级缓存都是依赖于基础支持层中的缓存模块实现的。Mybatis中自带的这两级缓存与Mybatis以及整个应用是运行在同一个JVM中的，共享同一块对内存。如果这两级缓存中的数据量较大，则可能影响系统中其他功能的运行，所以当需要缓存大量数据时，优先考虑使用Redis，Memcache等缓存产品

**Binding模块**

### 1.4.2 核心处理层