# 架构师面试题系列之Mybatis面试

**1、什么是 MyBatis？**  
答：MyBatis 是一个可以自定义 SQL、存储过程和高级映射的持久层框架。

**2、讲下 MyBatis 的缓存**  
答：MyBatis 的缓存分为一级缓存和二级缓存,一级缓存放在 session 里面,默认就有,二级缓存放在它的命名空间里,默认是不打开的,使用二级缓存属性类需要实现 Serializable 序列化接口(可用来保存对象的状态),可在它的映射文件中配置<cache/>

**3、Mybatis 是如何进行分页的？分页插件的原理是什么？**  
答：  
1）Mybatis 使用 RowBounds 对象进行分页，也可以直接编写 sql 实现分页，也可以使用Mybatis 的分页插件。  
2）分页插件的原理：实现 Mybatis 提供的接口，实现自定义插件，在插件的拦截方法内拦截待执行的 sql，然后重写 sql。  
举例：select *from student，拦截 sql 后重写为：select t.* from （select \* from student）t limit 0，10

**4、简述 Mybatis 的插件运行原理，以及如何编写一个插件？**  
答：  
1）Mybatis 仅可以编写针对 ParameterHandler、ResultSetHandler、StatementHandler、Executor 这 4 种接口的插件，Mybatis 通过动态代理，为需要拦截的接口生成代理对象以实现接口方法拦截功能，每当执行这 4 种接口对象的方法时，就会进入拦截方法，具体就是InvocationHandler 的 invoke()方法，当然，只会拦截那些你指定需要拦截的方法。  
2）实现 Mybatis 的 Interceptor 接口并复写 intercept()方法，然后在给插件编写注解，指定要拦截哪一个接口的哪些方法即可，记住，别忘了在配置文件中配置你编写的插件。

**5、Mybatis 动态 sql 是做什么的？都有哪些动态 sql？能简述一下动态 sql 的执行原理不？**  
答：  
1）Mybatis 动态 sql 可以让我们在 Xml 映射文件内，以标签的形式编写动态 sql，完成逻辑判断和动态拼接 sql 的功能。  
2）Mybatis 提供了 9 种动态 sql 标签：trim|where|set|foreach|if|choose|when|otherwise|bind。  
3）其执行原理为，使用 OGNL 从 sql 参数对象中计算表达式的值，根据表达式的值动态拼接 sql，以此来完成动态 sql 的功能。

**6、#{}和${}的区别是什么？**  
答：  
1）#{}是预编译处理，${}是字符串替换。  
2）Mybatis 在处理#{}时，会将 sql 中的#{}替换为?号，调用 PreparedStatement 的 set 方法来赋值；  
3）Mybatis 在处理{}时，就是把时，就是把{}替换成变量的值。  
4）使用#{}可以有效的防止 SQL 注入，提高系统安全性。

**7、为什么说 Mybatis 是半自动 ORM 映射工具？它与全自动的区别在哪里？**  
答：Hibernate 属于全自动 ORM 映射工具，使用 Hibernate 查询关联对象或者关联集合对象时，可以根据对象关系模型直接获取，所以它是全自动的。而 Mybatis 在查询关联对象或关联集合对象时，需要手动编写 sql 来完成，所以，称之为半自动 ORM 映射工具。

**8、Mybatis 是否支持延迟加载？如果支持，它的实现原理是什么？**  
答：  
1）Mybatis 仅支持 association 关联对象和 collection 关联集合对象的延迟加载，association指的就是一对一，collection 指的就是一对多查询。在 Mybatis 配置文件中，可以配置是否启用延迟加载 lazyLoadingEnabled=true|false。  
2）它的原理是，使用 CGLIB 创建目标对象的代理对象，当调用目标方法时，进入拦截器方法，比如调用 a.getB().getName()，拦截器 invoke()方法发现 a.getB()是 null 值，那么就会单独发送事先保存好的查询关联 B 对象的 sql，把 B 查询上来，然后调用 a.setB(b)，于是 a 的对象 b 属性就有值了，接着完成 a.getB().getName()方法的调用。这就是延迟加载的基本原理。

**9、MyBatis 与 Hibernate 有哪些不同？**  
答：  
1）Mybatis 和 hibernate 不同，它不完全是一个 ORM 框架，因为 MyBatis 需要程序员自己编写 Sql 语句，不过 mybatis 可以通过 XML 或注解方式灵活配置要运行的 sql 语句，并将java 对象和 sql 语句映射生成最终执行的 sql，最后将 sql 执行的结果再映射生成 java 对象。  
2）Mybatis 学习门槛低，简单易学，程序员直接编写原生态 sql，可严格控制 sql 执行性能，灵活度高，非常适合对关系数据模型要求不高的软件开发，例如互联网软件、企业运营类软件等，因为这类软件需求变化频繁，一但需求变化要求成果输出迅速。但是灵活的前提是 mybatis 无法做到数据库无关性，如果需要实现支持多种数据库的软件则需要自定义多套 sql 映射文件，工作量大。  
3）Hibernate 对象/关系映射能力强，数据库无关性好，对于关系模型要求高的软件（例如需求固定的定制化软件）如果用 hibernate 开发可以节省很多代码，提高效率。但是Hibernate 的缺点是学习门槛高，要精通门槛更高，而且怎么设计 O/R 映射，在性能和对象模型之间如何权衡，以及怎样用好 Hibernate 需要具有很强的经验和能力才行。  
总之，按照用户的需求在有限的资源环境下只要能做出维护性、扩展性良好的软件架构都是好架构，所以框架只有适合才是最好。

**10、MyBatis 的好处是什么？**  
答：  
1）MyBatis 把 sql 语句从 Java 源程序中独立出来，放在单独的 XML 文件中编写，给程序的维护带来了很大便利。  
2）MyBatis 封装了底层 JDBC API 的调用细节，并能自动将结果集转换成 Java Bean 对象，大大简化了 Java 数据库编程的重复工作。  
3）因为 MyBatis 需要程序员自己去编写 sql 语句，程序员可以结合数据库自身的特点灵活控制 sql 语句，因此能够实现比 Hibernate 等全自动 orm 框架更高的查询效率，能够完成复杂查询。

**11、简述 Mybatis 的 Xml 映射文件和 Mybatis 内部数据结构之间的映射关系？**  
答：Mybatis 将所有 Xml 配置信息都封装到 All-In-One 重量级对象 Configuration 内部。在Xml 映射文件中，<parameterMap>标签会被解析为 ParameterMap 对象，其每个子元素会被解析为 ParameterMapping 对象。<resultMap>标签会被解析为 ResultMap 对象，其每个子元素会被解析为 ResultMapping 对象。每一个<select>、<insert>、<update>、<delete>标签均会被解析为 MappedStatement 对象，标签内的 sql 会被解析为 BoundSql 对象。

**12、什么是 MyBatis 的接口绑定,有什么好处？**  
答：接口映射就是在 MyBatis 中任意定义接口,然后把接口里面的方法和 SQL 语句绑定,我们直接调用接口方法就可以,这样比起原来了 SqlSession 提供的方法我们可以有更加灵活的选择和设置.

**13、接口绑定有几种实现方式,分别是怎么实现的?**  
答：接口绑定有两种实现方式,一种是通过注解绑定,就是在接口的方法上面加上@Select@Update 等注解里面包含 Sql 语句来绑定,另外一种就是通过 xml 里面写 SQL 来绑定,在这种情况下,要指定 xml 映射文件里面的 namespace 必须为接口的全路径名.

**14、什么情况下用注解绑定,什么情况下用 xml 绑定？**  
答：当 Sql 语句比较简单时候,用注解绑定；当 SQL 语句比较复杂时候,用 xml 绑定,一般用xml 绑定的比较多

**15、MyBatis 实现一对一有几种方式?具体怎么操作的？**  
答：有联合查询和嵌套查询,联合查询是几个表联合查询,只查询一次,通过在 resultMap 里面配置 association 节点配置一对一的类就可以完成;嵌套查询是先查一个表,根据这个表里面的结果的外键 id,去再另外一个表里面查询数据,也是通过 association 配置,但另外一个表的查询通过 select 属性配置。

**16、Mybatis 能执行一对一、一对多的关联查询吗？都有哪些实现方式，以及它们之间的区别？**  
答：能，Mybatis 不仅可以执行一对一、一对多的关联查询，还可以执行多对一，多对多的关联查询，多对一查询，其实就是一对一查询，只需要把 selectOne()修改为 selectList()即可；多对多查询，其实就是一对多查询，只需要把 selectOne()修改为 selectList()即可。关联对象查询，有两种实现方式，一种是单独发送一个 sql 去查询关联对象，赋给主对象，然后返回主对象。另一种是使用嵌套查询，嵌套查询的含义为使用 join 查询，一部分列是 A 对象的属性值，另外一部分列是关联对象 B 的属性值，好处是只发一个 sql 查询，就可以把主对象和其关联对象查出来。

**17、MyBatis 里面的动态 Sql 是怎么设定的?用什么语法?**  
答：MyBatis 里面的动态 Sql 一般是通过 if 节点来实现,通过 OGNL 语法来实现,但是如果要写的完整,必须配合 where,trim 节点,where 节点是判断包含节点有内容就插入 where,否则不插入,trim 节点是用来判断如果动态语句是以 and 或 or 开始,那么会自动把这个 and 或者 or取掉。

**18、Mybatis 是如何将 sql 执行结果封装为目标对象并返回的？都有哪些映射形式？**  
答：  
第一种是使用<resultMap>标签，逐一定义列名和对象属性名之间的映射关系。  
第二种是使用 sql 列的别名功能，将列别名书写为对象属性名，比如 T\_NAME AS NAME，对象属性名一般是 name，小写，但是列名不区分大小写，Mybatis 会忽略列名大小写，智能找到与之对应对象属性名，你甚至可以写成 T\_NAME AS NaMe，Mybatis 一样可以正常工作。  
有了列名与属性名的映射关系后，Mybatis 通过反射创建对象，同时使用反射给对象的属性逐一赋值并返回，那些找不到映射关系的属性，是无法完成赋值的。

**19、Xml 映射文件中，除了常见的 select|insert|updae|delete 标签之外，还有哪些标签？**  
答：还有很多其他的标签，<resultMap>、<parameterMap>、<sql>、<include>、<selectKey>，加上动态 sql 的 9 个标签，trim|where|set|foreach|if|choose|when|otherwise|bind 等，其中<sql>为 sql 片段标签，通过<include>标签引入 sql 片段，<selectKey>为不支持自增的主键生成策略标签。

**20、当实体类中的属性名和表中的字段名不一样，如果将查询的结果封装到指定 pojo？**  
答：  
1）通过在查询的 sql 语句中定义字段名的别名。  
2）通过<resultMap>来映射字段名和实体类属性名的一一对应的关系。

**21、模糊查询 like 语句该怎么写**  
答：  
1）在 java 中拼接通配符，通过#{}赋值  
2）在 Sql 语句中拼接通配符 （不安全 会引起 Sql 注入）

**22、通常一个 Xml 映射文件，都会写一个 Dao 接口与之对应, Dao 的工作原理，是否可以重载？**  
答：不能重载，因为通过 Dao 寻找 Xml 对应的 sql 的时候全限名+方法名的保存和寻找策略。接口工作原理为 jdk 动态代理原理，运行时会为 dao 生成 proxy，代理对象会拦截接口方法，去执行对应的 sql 返回数据。

**23、Mybatis 映射文件中，如果 A 标签通过 include 引用了 B 标签的内容，请问，B 标签能否定义在 A 标签的后面，还是说必须定义在 A 标签的前面？**  
答：虽然 Mybatis 解析 Xml 映射文件是按照顺序解析的，但是，被引用的 B 标签依然可以定义在任何地方，Mybatis 都可以正确识别。原理是，Mybatis 解析 A 标签，发现 A 标签引用了 B 标签，但是 B 标签尚未解析到，尚不存在，此时，Mybatis 会将 A 标签标记为未解析状态，然后继续解析余下的标签，包含 B 标签，待所有标签解析完毕，Mybatis 会重新解析那些被标记为未解析的标签，此时再解析 A 标签时，B 标签已经存在，A 标签也就可以正常解析完成了。

**24、Mybatis 的 Xml 映射文件中，不同的 Xml 映射文件，id 是否可以重复？**  
答：不同的 Xml 映射文件，如果配置了 namespace，那么 id 可以重复；如果没有配置namespace，那么 id 不能重复；毕竟 namespace 不是必须的，只是最佳实践而已。原因就是 namespace+id 是作为 Map<String, MappedStatement>的 key 使用的，如果没有namespace，就剩下 id，那么，id 重复会导致数据互相覆盖。有了 namespace，自然 id 就可以重复，namespace 不同，namespace+id 自然也就不同。25、Mybatis 中如何执行批处理？  
答：使用 BatchExecutor 完成批处理。

**26、Mybatis 都有哪些 Executor 执行器？它们之间的区别是什么？**  
答：Mybatis 有三种基本的 Executor 执行器，SimpleExecutor、ReuseExecutor、BatchExecutor。1）SimpleExecutor：每执行一次 update 或 select，就开启一个 Statement 对象，用完立刻关闭 Statement 对象。2）ReuseExecutor：执行 update 或 select，以 sql 作为key 查找 Statement 对象，存在就使用，不存在就创建，用完后，不关闭 Statement 对象，而是放置于 Map3）BatchExecutor：完成批处理。

**27、Mybatis 中如何指定使用哪一种 Executor 执行器？**  
答：在 Mybatis 配置文件中，可以指定默认的 ExecutorType 执行器类型，也可以手动给DefaultSqlSessionFactory 的创建 SqlSession 的方法传递 ExecutorType 类型参数。

**28、Mybatis 执行批量插入，能返回数据库主键列表吗？**  
答：能，JDBC 都能，Mybatis 当然也能。

**29、Mybatis 是否可以映射 Enum 枚举类？**  
答：Mybatis 可以映射枚举类，不单可以映射枚举类，Mybatis 可以映射任何对象到表的一列上。映射方式为自定义一个 TypeHandler，实现 TypeHandler 的 setParameter()和getResult()接口方法。TypeHandler 有两个作用，一是完成从 javaType 至 jdbcType 的转换，二是完成 jdbcType 至 javaType 的转换，体现为 setParameter()和 getResult()两个方法，分别代表设置 sql 问号占位符参数和获取列查询结果。

**30、如何获取自动生成的(主)键值？**  
答：配置文件设置 usegeneratedkeys 为 true

**31、在 mapper 中如何传递多个参数？**  
答：  
1）直接在方法中传递参数，xml 文件用#{0} #{1}来获取  
2）使用 @param 注解:这样可以直接在 xml 文件中通过#{name}来获取

**32、resultType resultMap 的区别？**  
答：  
1）类的名字和数据库相同时，可以直接设置 resultType 参数为 Pojo 类  
2）若不同，需要设置 resultMap 将结果名字和 Pojo 名字进行转换

**33、使用 MyBatis 的 mapper 接口调用时有哪些要求？**  
答：  
1）Mapper 接口方法名和 mapper.xml 中定义的每个 sql 的 id 相同  
2）Mapper 接口方法的输入参数类型和 mapper.xml 中定义的每个 sql 的 parameterType 的类型相同  
3）Mapper 接口方法的输出参数类型和 mapper.xml 中定义的每个 sql 的 resultType 的类型相同  
4）Mapper.xml 文件中的 namespace 即是 mapper 接口的类路径。

**34、Mybatis 比 IBatis 比较大的几个改进是什么？**  
答：  
1）有接口绑定,包括注解绑定 sql 和 xml 绑定 Sql  
2）动态 sql 由原来的节点配置变成 OGNL 表达式 3） 在一对一,一对多的时候引进了association,在一对多的时候引入了 collection 节点,不过都是在 resultMap 里面配置

**35、IBatis 和 MyBatis 在核心处理类分别叫什么？**  
答：IBatis 里面的核心处理类叫 SqlMapClient,MyBatis 里面的核心处理类叫做 SqlSession。

**36、IBatis 和 MyBatis 在细节上的不同有哪些？**  
答：  
1）在 sql 里面变量命名由 原来的#变量# 变成了#{变量}  
2）原来的变量变量变成了${变量}  
3）原来在 sql 节点里面的 class 都换名字叫  type  
4）原来的 queryForObject queryForList 变成了 selectOne selectList5）原来的别名设置在映射文件里面放在了核心配置文件里