

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,一级缓存在 session 里面,默认就有,二级缓存在它的命名空间里,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

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\${ }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，一旦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，标签会被解析为 ParameterMap 对象，其每个子元素会被解析为 ParameterMapping 对象。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