

1、Mybatis 动态 sql 是做什么的？都有哪些动态 sql？简述一下动态 sql 的执行原理？

(1) Mybatis 以 XML 标签的形式编写动态 SQL，可以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：<if/>、<choose/>、<when/>、<otherwise/>、<trim/>、<when/>、<set/>、<foreach/>、<bind/>。

(3) 其执行原理为，使用 OGNL 的表达式，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。

2、Mybatis 是否支持延迟加载？如果支持，它的实现原理是什么？

Mybatis 支持动态加载，延迟加载就是当在真正需要数据的时候，才真正执行数据加载操作。可以在配置文件中设置 lazyLoadingEnabled=true|false 控制是否启用延迟加载，在一对多和多对多的场景下可以使用延迟加载，其原理是通过创建目标对象的代理对象，在调用目标对象的方法是，进入拦截器的 invoke 方法，在 invoke 方法内加载对象。

3、Mybatis 都有哪些 Executor 执行器？它们之间的区别是什么？

Mybatis 都有 SimpleExecutor、ReuseExecutor、BatchExecutor 三个 executor 执行器。

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。

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 Map 内，供下一次使用。

BatchExecutor：执行 update 将所有 sql 都添加到批处理中 addBatch()，

等待统一执行 `executeBatch()` ,它缓存了多个 `Statement` 对象 ,每个 `Statement` 对象都是 `addBatch()` 完毕后 ,等待逐一执行 `executeBatch()` 批处理。

4、简述下 **Mybatis** 的一级、二级缓存（分别从存储结构、范围、失效场景。三个方面来作答）？

(1)Mybatis 一级缓存是 `sqlSession` 级别的缓存 ,默认开启 ,在 `sqlSession` 关闭或者执行了增删改操作后 ,缓存被清理 ,也可以手动清理 ,其存储结构为 `HashMap`。

(2)Mybatis 二级缓存是 `Mapper` 级别的缓存 ,可以跨 `sqlSession` 存在 ,在 `Mapper` 下的某一个 `session` 执行了增删改操作后 ,缓存将被清理。二级缓存需要手动开启 ,二级缓存开启后实体类需实现 `Serializable` 接口 ,因为对象可能被缓存在硬盘中 ,需反序列化获取。二级缓存不支持分布式场景 ,可能出现脏读的情况。Mybatis 二级缓存需要结合 `redis` 来支持分布式缓存。

5、简述 **Mybatis** 的插件运行原理 , 以及如何编写一个插件？

Mybatis 仅可以编写针对 `ParameterHandler`、`ResultSetHandler`、`StatementHandler`、`Executor` 这 4 种接口的插件 , Mybatis 使用 JDK 的动态代理 ,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 `InvocationHandler` 的 `invoke()` 方法。

可以使用注解的方式指定需要拦截的方法。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。