

一、 mybatis 动态 sql 是做什么的？有哪些动态 sql？简述一下动态 sql 的执行原理。

- (1) MyBatis 的动态 SQL 是基于 OGNL 表达式的,它可以帮助我们方便的在 SQL 语句中实现某些逻辑
- (2) If、sql、where、foreach
- (3) 在解析 mapper 文件的时候通过判断动态标签中的值来存储原始 sql。

二、 mybatis 是否有延迟加载

支持

Mybatis 的延迟加载的基本原理是 jdk 动态代理，开启延迟加载后在进行查询的时候，如果匹配到 `fetchType="lazy"`，不会进行关联对象的查询，而是只查询主体数据并将查询结果封装为结果类型的代理对象。在调用懒加载属性的 `get/set` 方法（或者是其他触发懒加载操作的方法）时 mybatis 才能知道这时候应该去加载懒加载属性

三、 Executor 的种类

- (1) mybatis 有三种 executor 执行器，分别为 `simpleexecutor`、`reuseexecutor`、`batchexecutor`。
- (2) `simpleexecutor` 执行器：在每执行一次 `update` 或 `select`，就开启一个 `statement` 对象，用完后就关闭。
- (3) `reuseexecutor` 执行器：在执行 `update` 或 `select` 时以 `sql` 作为 `key` 去查找 `statement`，有就直接使用，没有就创建，使用完毕后不关闭，放入 `Map<String,Statement>` 中，供下次使用。重复使用 `statement`。

(4) batchexecutor 执行器:执行 update (jdbc 批处理不支持 select), 会把所有 sql 添加到批处理中 addbatch (); 等待统一批处理 executorbatch (); 它缓存了多个 statement, 每一个 statement 都是 addbatch (), 后等待进行 executorbatch () 批处理。

四、Mybat 的缓存

- (1) 存储结构：都是 Hashmap
- (2) 范围：一级缓存 SqlSession 级别的，同一 Session 相同的条件第二次查询就会从缓存中查询，而不是从数据库中查询，mybatis 默认是开启一级缓存的；二级缓存是 mapper 级别
- (3) 失效场景：一级缓存在进行更新和插入提交事务的时候回清空缓存；而二级缓存需要手动开启 flushCache 刷新缓存

五、插件的执行原理

- (1) 通过自定义拦截器，指定插件拦截的目标类的目标方法，达到在目标方法每次执行之前完成对目标方法的增强操作。
- (2) 1、实现 Interceptor 接口
 - 2、在是实现类上添加@Interceptors 注解，使该拦截器定位到目标方法。
 - 3、在 mybatis 核心配置文件中配置自定义插件，也就是步骤 2 的实现类。