

## 一、简答题

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？

Mybatis 动态 sql 是做动态 SQL 拼接，可以根据传入的参数动态拼接执行的 SQL，减少 SQL 语句和代码量，增加代码的扩展性和复用性。

动态 SQL：where、if、foreach、include

执行原理：

- 首先在解析 xml 配置文件的时候，会有一个 `SqlSource sqlSource = langDriver.createSqlSource(configuration, context, parameterTypeClass)` 的操作
- `createSqlSource` 底层使用了 `XMLScriptBuilder` 来对 xml 中的标签进行解析
- `XMLScriptBuilder` 调用了 `parseScriptNode()` 的方法，
- 在 `parseScriptNode()` 的方法中有一个 `parseDynamicTags()` 方法，会对 `nodeHandlers` 里的标签根据不同的 handler 来处理不同的标签
- 然后把 `DynamicContext` 结果放回 `SqlSource` 中
- `DynamicSqlSource` 获取 `BoundSql`
- 在 `Executor` 执行的时候，调用 `DynamicSqlSource` 的解析方法，并返回解析好的 `BoundSql` 和已经排好序，需要替换的参数

## 2、Mybatis 是否支持延迟加载？如果支持，它的实现原理是什么？

支持延迟加载

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`。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 查询出来，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。

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

**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。简言之，就是重复使用 `Statement` 对象

**BatchExecutor:** 执行 `update`（没有 `select`，JDBC 批处理不支持 `select`），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。与 JDBC 批处理相同

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）？

**一级缓存：**当执行查询以后，查询的结果会同时存入到 `SqlSession` 提供的一块区域中，该区域的结构是一个 `Map`，当 `SqlSession` 对象消失时，mybatis 的一级缓存也就消失了，同时一级缓存是 `SqlSession` 范围的缓存，当调用 `SqlSession` 的修改、添加、删除、`commit()`、`close` 等方法时，就会清空一级缓存。

二级缓存是 `mapper` 级别的缓存，多个 `SqlSession` 去操作同一个 `mapper` 的 `sql` 语句，多个 `SqlSession` 可以共用二级缓存，二级缓存是跨 `SqlSession`。第一次调用 `mapper` 下的 `sql` 的时候去查询信息，查询到的信息会存放到该 `mapper` 对应的二级缓存区域，第二次调用 `namespace` 下的 `mapper` 映射文件中，相同的 `SQL` 去查询，回去对应的二级缓存内取结果，使用值需要开启 `cache` 标签，在 `select` 上添加 `useCache` 属性为 `true`，在更新和删除时候需要手动开启 `flushCache` 刷新缓存。

5、简述 Mybatis 的插件运行原理，以及如何编写一个插件？

Mybatis 的插件是针对拦截器的一个实现类，在执行 mybatis 时，调用注册的拦截器对方法进行增强或者拦截，