

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？

1.1 Mybatis动态sql可以根据参数动态拼接sql语句，实现不同参数执行不同的sql语句。

1.2 常用的动态sql包括

if-条件拼接

choose/when/otherwise-多分支判断

foreach-迭代拼接

sql片段-提取公共sql片段

where-辅助拼接where语句，避免动态拼接产生的and语法错误

1.3 动态sql是在mybatis全局配置文件解析时，解析其中的mappers节点下的xml文件，根据不同xml标记解析成不同的SqlNode节点对象。在执行时根据参数执行SqlNode，形成最终执行的sql，并构建BoundSql对象。

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

实现原理是在创建结果对象时，如果开启了延迟加载则创建一个结果对象的代理对象，拦截结果对象的get方法，在调用结果对象的get方法时，再执行真正的查询。

3、Mybatis都有哪些Executor执行器？它们之间的区别是什么？

有五种Executor执行器

- BaseExecutor

使用模板方法模式，提供基础方法

- BatchExecutor

批量执行

- ClosedExecutor

已关闭的executor不可再使用其执行操作

- ReuseExecutor

可重用Statement 对象

- SimpleExecutor

默认Executor，每次执行时创建Statement对象

- CachingExecutor

支持二级缓存

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）？

一级缓存是Mybatis默认开启的，其底层使用map作为存储，作用范围为sqlsession级别，在sqlsession调用clear或者close方法时清除缓存，或者调用增删改写操作时同样会使其清空。

二级缓存Mybatis默认关闭，其提供的默认实现同样采用的map作为存储，作用范围为跨sqlsession的mapper级别，多个sqlsession操作同一mapper的sql语句，会共享缓存。其失效只能通过增删改写操作触发清空。由于其默认实现是基于本地数据的，所以在分布式场景下会有脏数据问题，需要提供统一的二级缓存存储实现，比如redis。

5、简述Mybatis的插件运行原理，以及如何编写一个插件？

Mybatis在启动时会扫描所有的插件，根据插件上的注解表明其需要作用的接口方法。然后迭代 `Executor` `StatementHandler` `ParamterHandler` `ResultSetHandler` 四大对象的实现，通过代理对象的方式在其上叠加插件功能。

自定义插件需要完成如下几个步骤：

- 实现Mybatis插件接口方法
- 添加 `@Interceptors` 和 `@Signature` 注解
- 在全局配置文件中配置插件配置