

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？

答：

动态sql 是指 在进行sql操作的时候，传入的参数对象或者参数值，根据匹配的条件，动态的去判断是否为空 循环，拼接等情况。

动态sql 9个 标签 if|where |bind|choose | when| otherwise|trim|set|foreach

动态sql执行原理

内部使用OGNL 的表达式，从sql参数中计算表达式的值，根据表达式的值 动态的拼接SQL，以此来完成 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仅支持association(一对一) 关联对象和collection（一对多）关联对象的延迟加载。在配置文件中 设置 lazyLoadingEnabled=true|false.

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 a.getB().getName()：当发现 a.getB() 方法是null值 就回单独发送事先保存好的查询B对象的sql 把B查询出来 并调用 a.setB(b)，于是a的对象B属性就有值了，接着调用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？

1 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。

2 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，而是放置在Map中，供下一次使用。

3 BatchExecutor: 执行update (不包括select)，将所有的sql都添加到批处理中（addBatch()），等待统一执行，它缓存多个Statement对象，每个 Statement对象都是addBatch()完毕后，等待 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）？

一级缓存：存储结构 HashMap 存储结构

范围 在同一个sqlSession 中

失效场景 sqlSession.close 或者 手动调用 sqlSession.clearCache()方法

二级缓存 默认由 PerpetualCache类 实现 存储结构 查询到的信息会存放到该mapper对应的二级缓存区域 底层其实也是 HashMap 存储结构

范围：整个mapper中

失效场景：系统处于分布式架构中 二级缓存

5、简述Mybatis的插件运行原理，以及如何编写一个插件？

1> mybatis 创建1Executor,StatmentHandler,ParameterHandler,ResultSetHandler

这4个接口对象的时候 不是直接返回的，而是 调用 interceptorChain.pluginAll(parameterHandler)

2 获取到所有的拦截器

调用interceptor.plugin(target),返回target包装后的对象

3 使用Jdk的动态代理为需要拦截的接口生成代理对象，然后实现接口的拦截方法。

编写插件

1 创建插件类 实现interceptor接口并且使用注解标注拦截对象与方法

2在配置文件中 写入plugins标签