1. Mybatis动态sql是做什么的？都有哪些动态sql？简述下动态sql的执行原理

Mybatis 动态 SQL ，可以让我们在 XML 映射文件内，以 XML 标签的形 式编写动态 SQL ，完成逻辑判断和动态拼接 SQL 的功能

 Mybatis 提供了 9 种动态 SQL 标签：<if/>、<choose/>、<when/>、<otherwise/>、<trim/>、<when/>、<set/>、<foreach/>、<bind/>。

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

1. 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关联对象和collection关联集合对象的延迟加载，association指一对一，collection指一对多查询，可以配置是否启用延迟加载lazyLoadingEnabled=ture|false

原理：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

1. Mybatis都有哪些Executor执行器？他们之前区别是什么？

有三种基本的执行器：SimpleExecutor,ReuseExecutor,BatchExecutor

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，而是放置在Map中，供下一次使用。简言之，就是重复使用

BatchExecutor:执行update,将所有sql都添加到批处理中(addBatch())，等待统一执行（executeBatch()）,他缓存了多个Statement对象，每个Statement对象都是addBatch()完毕后，等待逐一执行executeBatch()批处理

1. 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）？

一级缓存是基于SqlSession级别的，用hashMap存储，查询的时候会根据cacheKey去缓存中查找，如何找到了返回结果，没找到则去数据库查询，并把结果缓存到map中，当删除，修改，添加之后，再次发起查询时，需要重新查数据库

二级缓存是基于Mapper级别的，作用域是Mapper文件中的namespace,当执行insert,update,delete时清除二级缓存，当没有session.close()时，没有二级缓存

useCache:使用在select语句上，true表示使用二级缓存，false查询时不使用，默认true

flushCache:使用在insert,update,delete语句上，true表示清除二级缓存，fasle表示不刷新二级缓存，默认值是true

1. 简述下Mybatis的插件运行原理，以及如何编写一个插件？

1每个创建出来的对象不是直接返回的，而是interceptorChain.pluginAll(parameterHandler);

2,获取到所有的interceptor(拦截器),调用interceptor.plugin（target）,返回target包装后的对象

1. 插件机制，我们可以使用插件为目标创建一个代理对象，（AOP）我们的插件可以为四大对象创建出代理对象，代理对象可以拦截到每一个执行

编写一个类实现interceptor接口,加上下面的注释

@Intercepts({

@Signture(type=Executor.class,method="query",args={})

})

并将插件配置到sqlMapConfig.xml中

<plugins>

<plugin inteceptor="插件类全类名"><plugin>

</plugins>