一、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.Mybatis 动态 sql 是指配置在 mapper.xml 中的 sql，mybatis 可以根据 sql 中的占位符，以及各种标签动态拼凑成最终真正执行的静态 sql。

2.比较常用的动态 sql 标签有

<foreach>循环遍历一个集合，在拼 sql 时，每次把需要的值加到 sql 中，可以指定 seperator，open 和 close。

<If>判断，通常是如果一个值为空时，可能会导致 sql 异常，为了避免异常，使用 if 标签判断不为空时才拼接标签内的 sql

<Set>在 update 时，为了避免 update tb_user set where 这种 sql，没有一个条件满足，应该不需要 set 了，但是还写上了 set。

<Where>可以避免 where and 这种尴尬的 sql

<Include>引用 xml 中一段 sql

3.执行原理。

首先是 mybatis 初始化时，把每个动态 sql 封装到 mappedStatement 对象中的 SqlSource 属性，存到 configuration 类：在 SQLSessionFactoryBuilder.build 创建 SQLSessionFactory 时，XMLStatementBuilder 的 parseStatementNode 会调用 builderAssistant.addMappedStatement 方法，在 addMappedStatement 方法中，把封装好的 mappedstatement 放到核心配置类中 configuration.addMappedStatement(statement)。

然后是在调用时，先拿到之前存好的 BoundSql，并解析执行。首先，传给 sqlsession 一个 mappedstatementId，然后在核心配置类中获取对应的 mappedstatement，然后 executor 通过调用 statementHandler，statementHandler 再调用 parameterHandler 以及 resultSetHandler

执行 sql 并封装返回结果。

二、Mybatis 是否支持延迟加载？如果支持，它的实现原理是什么？

支持。

实现原理是动态代理。Mybatis 在封装 resultset 时，会判断对象的属性是否是引用属性，如果是，就会为该属性的 get 方法创建代理方法，这样在执行 get 方法时，会执行代理方法，从数据库查询。

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

Mybatis 有三种基本的 Executor 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。

四、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）？

一级缓存的底层数据结构是 HashMap。范围是 SqlSession。当 sqlSession 执行了增删改操作后，不管有没有 commit，session 都会失效。

二级缓存的底层结构也是 HashMap，作用范围是 Mapper 的 namespace。当相同 namespace 的 sqlsession 执行 close 或者 commit 方法

时，mybatis 会把查询结果存到二级缓存里，其他相同 namespace 的 sqlsession 就可以直接拿到结果，不用查数据库。

二级缓存失效条件：

1. 如果 sqlsession1 没有执行 commit 或者 close 方法，则二级缓存不生效。
2. 相同 namespace 的 sqlsession 执行了增删改操作。

五、简述 Mybatis 的插件运行原理，以及如何编写一个插件？

在 mybatis 初始化的时候，会将所有的 interceptor 放到 configuration 中，在 openSession 时，会给 sqlsession 对象创建 executor，在 newExecutor 时，会遍历 Executor 对应的 interceptor，然后执行 plugin 方法为当前 Executor 对象创建代理对象，如果有多个 interceptor，则后一个又会代理前一个，形成一个代理 chain，在执行时，会依次执行代理对象，而不是原来的 Executor 本身了。其他三大对象 StatementHandler，ParameterHandler，ResultSetHandler 的插件，也都是在创建各个对象的时候，遍历 interceptor 并生成代理对象，本质上都是动态代理，不再赘述。

自定义拦截器首先要实现 Interceptor 接口，使用@Intercept 注解指定要拦截的对象，以及方法。然后重写 Interceptor 接口的三个方法，在 intercept 方法里实现拦截逻辑，在 plugin 方法里生成包装对象，在 setProperties 中可以设置参数。然后在 SqlMapConfig 核心配置文件中，配置上自定义的 interceptor 就可以了。