1. 动态SQL是做什么的？有哪些动态SQL？简述执行原理？

动态SQL是可以在Mapper.xml文件中用标签的形式来编写，根据表达式的值完成动态拼接SQL的目的。

Mybatis 提供了 9 种动态 sql 标签：trim | where | set | foreach | if | choose

| when | otherwise | bind。

MyBatis采用代理的方式来生成DAO文件，在创建SQLSessionFactory的时候读取xml文件，每一个insert/delete/update/select都会生成一个MappedStatment对象。在生成该对象的时候会解析动态SQL的标签，把解析后的sql语句存储在BoundSQL之中。

2.延迟加载？实现原理？

在数据集中有一对多或多对多的情况下，比如用户和账户的场景，一个用户可以有多个账户，在查询用户的时候不用查询其名下的账户，只有真正用到的时候才会去查询账户，这样减小了数据库的压力。

其实现是在第一次查询的时候不查询集合中对应的数据，并且存到缓存中，第二次再次执行查询的时候从缓存中加载该数据的信息在进行数据库的查询。

3.有哪些Executor执行器？区别？

从继承关系来看Executor结构有一个抽象实现类BaseExecutor，其提供一些通用方法，BaseExecutor存在三个子类：

BatchExecutor:提供批处理功能

SimpleExecutor：默认实现类，Statment对象使用时创建，使用完关闭

ReuseExecutor：Statment对象可以重复使用

4.一级，二级缓存？（存储结构，范围，失效场景）

一级缓存底层是一在PerpetualCache类中的HashMap，作用范围是每一个SqlSession中，失效场景:SqlSession.clearCache()或者执行增删改操作；

二级缓存底层是TransactionalCache类中的HashMap，作用范围是应用范围，也就是跨SqlSession中，失效场景：执行实务操作（增删改）

5.插件运行原理和如何编写？

通过拦截器链来实行四大对象的拦截增强：在MyBatis创建Executor，ParameterHandler，ResultSetHandler，StatmentHandler的时候都调用了“interceptorChain.pluginAll”方法通过反射来实现方法增强。

创建一个类实现Interceptor 接口，在@Signature 中说明拦截的是哪个接口，然后在sqlMapConfig.xml中说明该类的位置