Mybatis插件本质上就是拦截器，通过拦截mybatis的四大组件并增强其功能。而增强功能本质上是借助于jdk底层的动态代理来实现的，也就是说mybatis的四大组件都是代理对象。

MyBatis的四大组件（核心对象）：

Executor：执行器，负责mybatis的增删改查行为的执行；

StatementHandler：sql语法构建器，用于完成sql的预编译；

ParameterHandler：参数处理器，用于设置参数；

ResultSetHandler：结果集处理器，用于处理返回结果集；

动态对象在调用任一方法时都会去调invoke方法，在invoke方法中可以在原方法被调用之后或之前来完成功能的增强

在调用代理对象的方法时，执行了invoke方法：

public Object invoke(Object proxy, Method method, Object[] args) throws Throwable {

try {

Set<Method> methods = (Set)this.signatureMap.get(method.getDeclaringClass());

return methods != null && methods.contains(method) ?

/\* 若调用的方法在拦截链中，则执行插件中的增强方法（拦截方法） \*/

this.interceptor.intercept(new Invocation(this.target, method, args)) :

method.invoke(this.target, args);

} catch (Exception var5) {

throw ExceptionUtil.unwrapThrowable(var5);

}

}