

## 1、什么是逆向工程：


Mybatis的一个主要特点就是需要程序员自己写SQL，如果表太多的话，难免很麻烦，所以Mybatis官方提供了一个逆向工程，可以针对表单自动生成Mybatis所执行的代码（包括mapper.xml、mapper.java,po）在实际开发中，最常见的功能就是通过数据库表生成代码。


## 2、使用逆向工程：


3、使用Mybatis的逆向工程需要导入mybatis-generator-core-1.3.2.jar下面开始总结一下Mybatis逆向工程的使用步骤。


### 3.1、创建一个工程：

引入一下几个Jar包：

 log4j-1.2.16.jar

 mybatis-3.2.3.jar

 mybatis-generator-core-1.3.2.jar

 mysql-connector-java-5.1.28-bin.jar

完成配置文件：

generateConfig.xml:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE generatorConfiguration
3     PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4     "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5 <generatorConfiguration>
6     <context id="testTables" targetRuntime="Mybatis3">
7         <commentGenerator>
8             <!-- 是否自动生成注释: true or false -->
9             <property name="suppressAllComments" value="true"/>
10        </commentGenerator>
11        <!-- 数据库连接的信息: 驱动类, 连接地址, 用户名, 密码 -->
12        <jdbcConnection driverClass="com.mysql.jdbc.Driver"
13            connectionURL="jdbc:mysql://localhost:3306/mybatis"
14                userId="root"
15                password="password">
16        </jdbcConnection>
17        <!-- 默认false,把JDBC DECIMAL和NUMERIC类型解析为Integer, 为true
            时把二者解析为Java.math.BigDecimal-->
```

```

18     <javaTypeResolver>
19         <property name="forceBigDecimals" value="false"/>
20     </javaTypeResolver>
21     <javaModelGenerator targetPackage="com.cauchy.mybatis.po"
22         targetProject=".\\src">
23         <!-- enableSubPackages:是否让schema作为包的后缀 -->
24         <property name="enableSubPackages" value="false"/>
25         <!-- 从数据库返回的值清理前后的空格 -->
26         <property name="trimString" value="true"/>
27     </javaModelGenerator>
28     <sqlMapGenerator targetPackage="com.cauchy.mybatis.mapper"
29         targetProject=".\\src">
30         <!-- enableSubpackages:是否让schema作为后缀 -->
31         <property name="enableSubPackages" value="false"/>
32     </sqlMapGenerator>
33     <!-- targetPackage:mapper接口生成的位置 -->
34     <javaClientGenerator targetPackage="com.cauchy.mybatis.mapper"
35         type="XMLMAPPER"
36         targetProject=".\\src">
37         <!-- enableSubPackages:是否让schema作为包的后缀 -->
38         <property name="enableSubPackages" value="false"/>
39     </javaClientGenerator>
40     <!-- 指定数据库的表 -->
41     <table schema="" tableName="user"/>
42     <table schema="" tableName="orders"/>
43     <table schema="" tableName="item"/>
44     <table schema="" tableName="orderdetail"/>
45 </context>
46 </generatorConfiguration>

```

在默认包下定义核心代码即可。

GenerateSqlmap.java:

```

1
2
3 import java.io.File;
4 import java.util.ArrayList;
5 import java.util.List;
6

```

```
7 import org.mybatis.generator.api.MyBatisGenerator;
8 import org.mybatis.generator.config.Configuration;
9 import org.mybatis.generator.config.xml.ConfigurationParser;
10 import org.mybatis.generator.internal.DefaultShellCallback;
11
12 public class GenerateSqlMap {
13     public void generate()throws Exception{
14         List<String> warnings = new ArrayList<String>();
15         boolean overwrite = true;
16         // 指定逆向工程配置文件
17         File configFile = new File("generateConfig.xml");
18         ConfigurationParser cp = new ConfigurationParser(warnings);
19         Configuration config = cp.parseConfiguration(configFile);
20         DefaultShellCallback callback = new
DefaultShellCallback(overwrite);
21         MyBatisGenerator myBatisGenerator = new
MyBatisGenerator(config,callback,warnings);
22         myBatisGenerator.generate(null);
23     }
24     public static void main(String[] args)throws Exception{
25         try {
26             GenerateSqlMap generateSqlMap = new GenerateSqlMap();
27             generateSqlMap.generate();
28         }catch(Exception e) {
29             e.printStackTrace();
30         }
31     }
32 }
33
```

执行方法，刷新代码即可实现逆向生成代码。