### 逆向工程总述

自动生成

* JavaBean类文件，
* Dao接口文件，
* 自动生成增删改查的sql映射，Mapper映射文件
* 豪华版还有example封装条件类文件

简单版，就是简单的增删改查

豪华版，就是动态sql形成的增删改查，有生成好的条件类，封装后，按条件类中属性进行查询，条件特别复杂

### 导包

<!-- mybatis-generator版本控制 -->

<mybatis.generator.version>1.3.5</mybatis.generator.version>

<dependency>

<groupId>org.mybatis.generator</groupId>

<artifactId>mybatis-generator-core</artifactId>

<version>${mybatis.generator.version}</version>

</dependency>

### 编写一个配置文件

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE generatorConfiguration

PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"

"http://mybatis.org/dtd/mybatis-generator-config\_1\_0.dtd">

<generatorConfiguration>

<!--

**targetRuntime="MyBatis3Simple":生成简单版的CRUD**

**MyBatis3:豪华版**

-->

<context id=*"DB2Tables"* targetRuntime=*"MyBatis3Simple"*>

<!-- 是否阻止生成注释,true为阻止生成注释，false为不阻止生成注释 -->

<commentGenerator>

<property name=*"suppressAllComments"* value=*"true"*></property>

</commentGenerator>

<!-- jdbcConnection：指定如何连接到目标数据库 -->

<jdbcConnection driverClass=*"com.mysql.jdbc.Driver"*

connectionURL=*"jdbc:mysql://localhost:3306/mybatis"*

userId=*"root"*

password=*"123456"*>

</jdbcConnection>

<javaTypeResolver >

<property name=*"forceBigDecimals"* value=*"false"* />

</javaTypeResolver>

<!-- **javaModelGenerator：指定javaBean的生成策略**

**targetPackage="test.model"：目标包名**

**targetProject="\MBGTestProject\src"：目标工程（包的上一级）**

-->

<javaModelGenerator targetPackage=*"com.mybatis.entites"*

targetProject=*".\src\main\java"*>

<property name=*"enableSubPackages"* value=*"true"* />

<property name=*"trimStrings"* value=*"true"* />

</javaModelGenerator>

<!-- sqlMapGenerator：sql映射生成策略： -->

<sqlMapGenerator targetPackage=*"mybatis-mapper"*

targetProject=*".\src\main\resources"*>

<property name=*"enableSubPackages"* value=*"true"* />

</sqlMapGenerator>

<!-- javaClientGenerator:指定mapper接口所在的位置 -->

<javaClientGenerator type=*"XMLMAPPER"* targetPackage=*"com.mybatis.dao"*

targetProject=*".\src\main\java"*>

<property name=*"enableSubPackages"* value=*"true"* />

</javaClientGenerator>

<!-- 指定要逆向分析哪些表：根据表要创建javaBean，domainObjectName为bean的类名 -->

<table tableName=*"tbl\_department"* domainObjectName=*"Department"*></table>

<table tableName=*"tbl\_employee"* domainObjectName=*"Employee"*></table>

</context>

</generatorConfiguration>

### 执行，生成文件

List<String> warnings = **new** ArrayList<String>();

**boolean** overwrite = **true**;

//这里为mbg.xml文件的路径，在工程路径下

File configFile = **new** File("mbg.xml");

ConfigurationParser cp = **new** ConfigurationParser(warnings);

Configuration config = cp.parseConfiguration(configFile);

DefaultShellCallback callback = **new** DefaultShellCallback(overwrite);

MyBatisGenerator myBatisGenerator = **new** MyBatisGenerator(config,

callback, warnings);

myBatisGenerator.generate(**null**);

### 豪华版使用

SqlSessionFactory sqlSessionFactory = getSqlSessionFactory();

SqlSession openSession = sqlSessionFactory.openSession();

EmployeeMapper mapper = openSession.getMapper(EmployeeMapper.**class**);

条件筛选对象

EmployeeExample example=**new** EmployeeExample();

//创建(条件)1

Criteria criteria = example.createCriteria();

//and Email like %a%

criteria.andEmailLike("%a%");

//and Id>2

criteria.andIdGreaterThan(2);

//创建（条件）2

Criteria criteria2 = example.createCriteria();

把条件2或者到条件对象中

example.or(criteria2);

criteria2.andGenderEqualTo("0");

List<Employee> selectByExample = mapper.selectByExample(example);

System.***out***.println(selectByExample);

最终生成的sql为

**select id, last\_name, gender, email, department\_id**

**from tbl\_employee**

**WHERE ( email like ? and id > ? ) or( gender = ? )**