# **统计涉及的表**

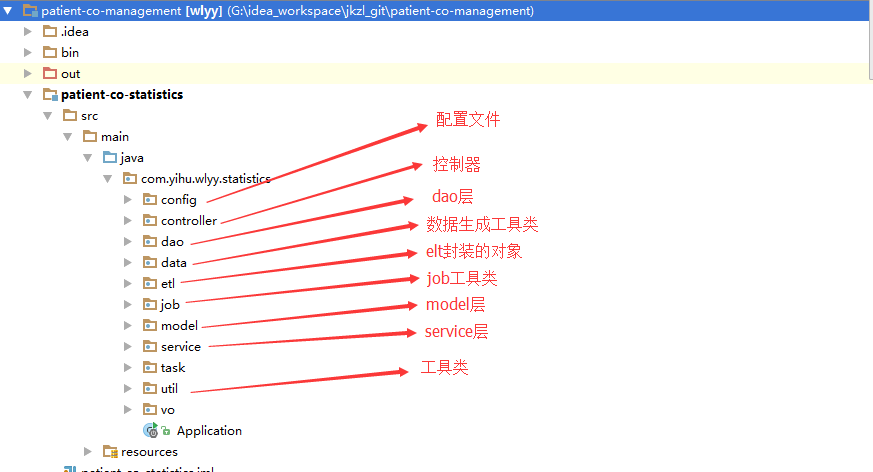
wlyy\_job\_config 任务配置表

wlyy\_job\_log 任务执行日志表

wlyy\_quota 统计配置表

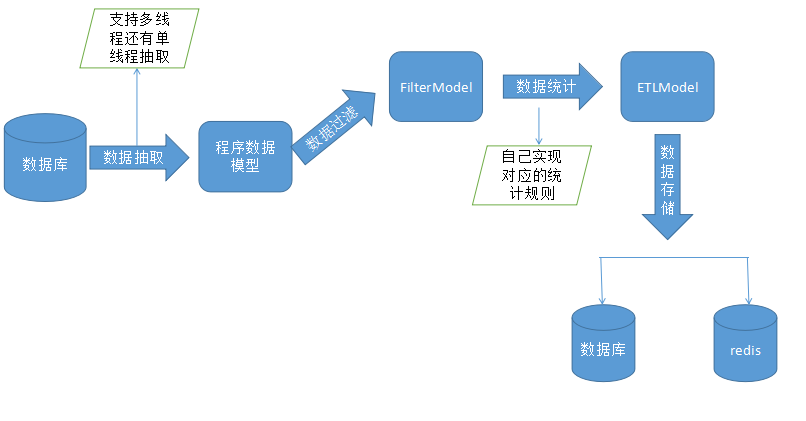
wlyy\_quota\_result 统计结果表

# **统计项目包结构**

[](http://a3.qpic.cn/psb?/V10CK5oY0FaOLu/mTYwcj4njrHdGpnb*vALOUYWi7*ruZcSYPMH7J5URb0!/b/dAoBAAAAAAAA&bo=aQPYAQAAAAADB5E!&rf=viewer_4)

[](http://a2.qpic.cn/psb?/V10CK5oY0FaOLu/8kX27*zcoEKE5F45kQtYeBoNgAxQZyPiiUc4GuP5GSc!/b/dHgBAAAAAAAA&bo=sgMgAgAAAAADB7E!&rf=viewer_4)

# **ETL流程设计分为以下四步**

[](http://a1.qpic.cn/psb?/V10CK5oY0FaOLu/oLXuliGRbaqnL1PV.3VuomyMDfrUXjKn8cKtyC5j3UI!/b/dOEAAAAAAAAA&bo=JgOlAQAAAAADB6M!&rf=viewer_4)

数据抽取------------>数据过滤----------->数据统计------------->数据储存

## 数据抽取

支持普通sql抽取也支持批量异步分页抽取

### 1.普通sql抽取:

参数:

SignFamily: model类型

Sql：sql语句

List<SignFamily> signFamilies= dbExtract.extract(SignFamily.**class**,sql);

### 2.异步分页多线程抽取

参数:

SignFamily: model类型

Sql：sql语句

sqlCount：每次抽取多少条

True：是否多线程抽取

List<SignFamily> t= dbExtract.extractByPage(SignFamily.**class**,sql,sqlCount,**true**);

## 数据过滤

功能：根据对应的业务过滤除脏数据，过滤结束后返回对应的过滤模型

目前已经存在的业务过滤器

### 咨询业务过滤器(ConsultDataFilter)

参数:

consultTeams:数据抽取返回的list

Sql:数据抽取的sql（日志记录用）

Yesterday：统计哪天的指标（日志记录用）

FilterModel etlModels= **consultDataFilter**.filter(consultTeams,sql,**yesterday**);

### 签约业务过滤器(SignDataFilter)

一级维度

cacheModel.getSignFamilies():从缓存中获取签约list

Sql:从缓存中获取执行的sql（日志记录用）

Yesterday：统计哪天的指标（日志记录用）

FilterModel etlModels= **signDataFilter**.filter(cacheModel.getSignFamilies(),cacheModel.getSql(),**yesterday**);

二级维度:

cacheModel.getSignFamilies():从缓存中获取签约list

Sql:从缓存中获取执行的sql（日志记录用）

SignDataFilter.level2Sex:二级维度的key(通过这个往etl中设置二级维度的值)

1:性别 2:年龄 3:分组 4:费用 5:健康服务分布 6:年龄 7:年龄疾病

Yesterday：统计哪天的指标（日志记录用）

FilterModel etlModels= **signDataFilter**.filter(cacheModel.getSignFamilies(),SignDataFilter.level2Sex,cacheModel.getSql(),**yesterday**);

三级维度:

cacheModel.getSignFamilies():从缓存中获取签约list

cacheModel.getSql():从缓存中获取执行的sql（日志记录用）

SignDataFilter.level2Age:二级维度的key(通过这个往etl中设置二级维度的值)

1:性别 2:年龄 3:分组 4:费用 5:健康服务分布 6:年龄 7:年龄疾病

SignDataFilter.level3Disease

1:疾病

**yesterday**：统计哪天的指标（日志记录用）

FilterModel etlModels= **signDataFilter**.filter(cacheModel.getSignFamilies(),SignDataFilter.level2Age,SignDataFilter.level3Disease,cacheModel.getSql(),**yesterday**);

### 随访业务过滤器(FollowUpDataFilter)

plans:数据抽取返回的list

Sql:数据抽取的sql（日志记录用）

Yesterday：统计哪天的指标（日志记录用）

FilterModel etlModels = **followUpDataFilter**.filter(plans, sql, **yesterday**);

### 健康指导业务过滤器(HealthGuideDataFilter)

consultTeams:数据抽取返回的list

Sql:数据抽取的sql（日志记录用）

Yesterday：统计哪天的指标（日志记录用）

FilterModel etlModels= **healthGuideDataFilter**.filter(consultTeams,sql,**yesterday**);

## 数据统计

统计分为三种：

1级维度的统计:Level1Role

2级维度的统计Level2Role

3级维度的统计:Level3Role

目前一级维度默认是4个等级 市 区 机构 团队（可无）

### Level1Role用法:

包含团队

参数

etlModels.getEtlModelList():数据过滤返回的list

List<Map<String, List<ETLModel>>> returnDatas= **levelRole**.elt(etlModels.getEtlModelList());

不包含团队

参数

etlModels.getEtlModelList():数据过滤返回的list

List<Map<String, List<ETLModel>>> returnDatas= **levelRole**.eltNoTeam(etlModels.getEtlModelList());

### Level2Role用法:

//统计数据 一级维度  
List<Map<String, List<ETLModel>>> returnDatas= **levelRole**.elt(etlModels.getEtlModelList());  
//统计数据 二级维度  
List<Map<String, Map<String, List<ETLModel>>>> level2Data= **level2Role**.elt(returnDatas);

### Level3Role用法:

*//统计数据 一级维度*List<Map<String, List<ETLModel>>> returnDatas= **levelRole**.elt(etlModels.getEtlModelList());  
*//统计数据 二级维度*List<Map<String, Map<String, List<ETLModel>>>> patientSexRoleData= **level2Role**.elt(returnDatas);  
*//统计数据 三级维度*List<Map<String, Map<String, Map<String, List<ETLModel>>>>> returnD=level3Role.elt(patientSexRoleData);

## 数据存储

### 数据库（DBStorage）

类:DBStorage

用法:

1级维度的保存

returnDatas:统计结束的数据

Yesterday：统计哪天的指标（日志记录用）

**wlyyQuota：指标对象**

**dbStorage**.saveByLevel1(returnDatas,**yesterday**,**wlyyQuota**);

2级维度的保存

level2Data:统计结束的数据

yesterday：统计哪天的指标（日志记录用）

wlyyQuota：指标对象

foreachNum **标识二级维度循环的次数** 包含尾 例如 年龄就传6 性别就传3   
 type 1:性别 2:年龄 3:分组 4:费用 5:健康服务分布 6:年龄 7:年龄疾病  
 start 起始位置 默认是1

dbStorage.saveByLevel2(level2Data,yesterday,wlyyQuota,foreachNum ,type ,start );

3级维度的保存

returnD:统计结束的数据

yesterday：统计哪天的指标（日志记录用）

wlyyQuota：指标对象

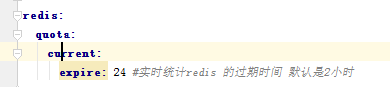
foreachNum2标识二级维度循环的次数 包含尾 例如 年龄就传6 性别就传3

foreachNum3标识三级维度循环的次数 包含尾 例如 年龄就传6 性别就传3   
 type2:性别 2:年龄 3:分组 4:费用 5:健康服务分布 6:年龄 7:年龄疾病  
 type3:1 疾病分组

dbStorage.saveByLevel3(returnD,yesterday,wlyyQuota,foreachNum2,foreachNum3,type2,type3);

### Redis(RedisStorage)

Redis中的数据的**key都有设置生命周期:**可以在yml中配置



**Redis中的存储格式**

**1级维度**

quota:指标code:1级维度等级:业务的code:timeKey时间戳

例如

指标code为1的市级的市是350200的数目 quota:1:4:350200:1477663200727

指标code为1的区级的区是350203的数目 :quota:1:3:350203:1477663200727

指标code为1的机构级的机构是3502030100数目 :quota:1:2:3502030100:1477663200727

指标code为1的团队级的团队100是数目 :quota:1:1:100:1491976800176

**2级维度**

quota:指标code:1级维度等级:1级维度的业务code(4市3区 2机构 1团队):2级维度等级:timeKey时间戳

例如

指标code为7的市级是市是350200二级维度业务代码 1(普通人群)的数目

quota:7:4:350200:1:1491969600205

指标code为7的区级的区是350206二级维度业务代码2(慢病65岁以下人群)的数目

quota:7:3:350206:2:1491984000234

指标code为7的机构级的机构是3502030100二级维度业务代码 0(未分組)数目

quota:7:2:3502030500:0:1491969600205

指标code为7的团队级的团队是127二级维度业务代码 0(未分組)数目

quota:7:1:127:0:1491976800176

**3级维度**

quota:指标code:1级维度等级:1级维度的业务code(4市3区 2机构 1团队):2级维度业务:3级维度业务:timeKey时间戳

例如

指标code为12的市级是市是350200二级维度业务代码 1 三级维度业务代码 1 的数目

quota:12:4:350200:1:1:1492041600237

指标code为12的区级的区是350206二级维度业务代码 1 三级维度业务代码 1的数目

quota:12:3:350206:1:1:1492041600237

指标code为12的机构级的机构是3502030100二级维度业务代码 1 三级维度业务代码 1数目

quota:12:2:3502030500:1:1:1492041600237

指标code为12的团队级的团队是127二级维度业务代码 1 三级维度业务代码 1数目

quota:12:2:127:1:1:1492041600237

# **简单的新增指标步奏如下**

## 1在2个表中新增数据

wlyy\_job\_config 任务配置表

wlyy\_quota 统计配置表





## 2代码参考已经存在的指标

由于目前指标生成是分成2部分

1是quartz定时统计的**隔天**的指标

1. 是实时的每个2小时统计**截止到今天**的指标

需要**新增定时指标**

如果是一级指标代码参考UnSignJob.class

如果是二级指标代码参考SignExpenseStatusJob.class

如果是三级指标代码参考SignAgeGroupDiseaseJob.class

还需要**新增实时的统计指标**

在类中 CurrentDayAllQuotaJob.class中新增



对应的新增方法既可