使用结对编程的理念进行设计文档、项目交接文档、用户手册之类的文档编写：一个人讲，一个人听，遇到问题时问，交流过程中必要时可以画草图，最后听的人负责文档的编写。如：

对于设计文档：设计人员将，开发人员听，开发人员写文档

对于项目交接文档（实际上推荐使用设计文档作为项目交接文档）：开发人员讲，维护人员听，维护人写文档。

对于用户手册：开发人员将，实施人员听，实施人员写文档。

# 看板

## 业务描述

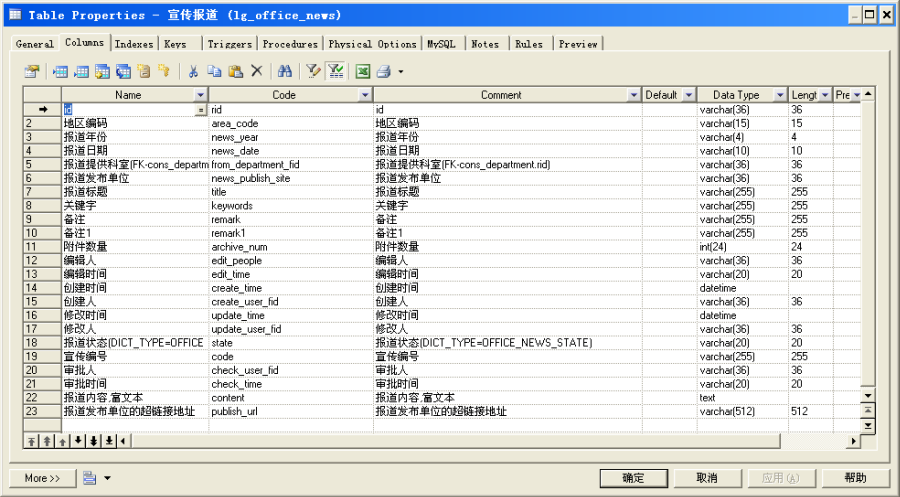
* 几大类用户各自登录成功后，被引导到的首页面，
* 主要针对工作办公模板中的相关功能进行的统计和概述信息。
* 页面的调度信息通过javascript来实现的，
* 文件名为menu\_master.js
* 入口方法init()



## 工作通告

### 公文

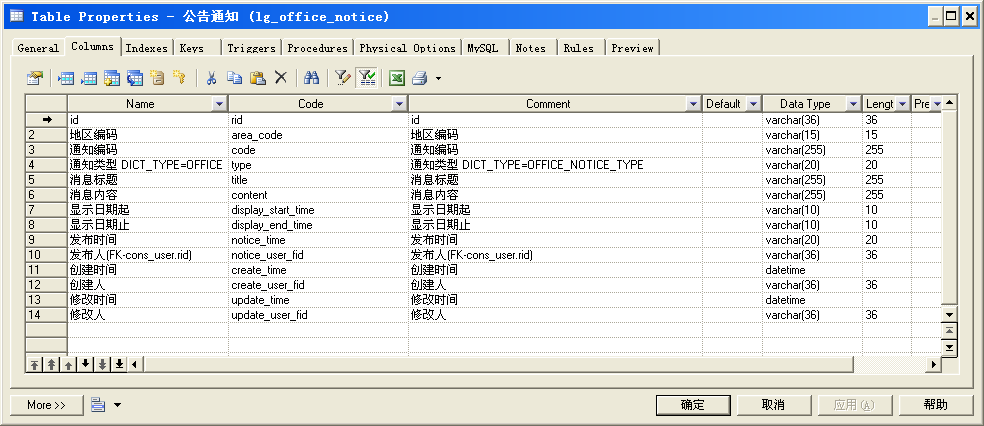
* 服务器的处理action链接地址：/main/index/ajaxFindGongwen
* 实现类：IndexController.ajaxFindGongwen(int, Model)



* 表名称: lg\_office\_document
* 按照文档的日期，倒序方式加载5条件记录进行显示

### 公告

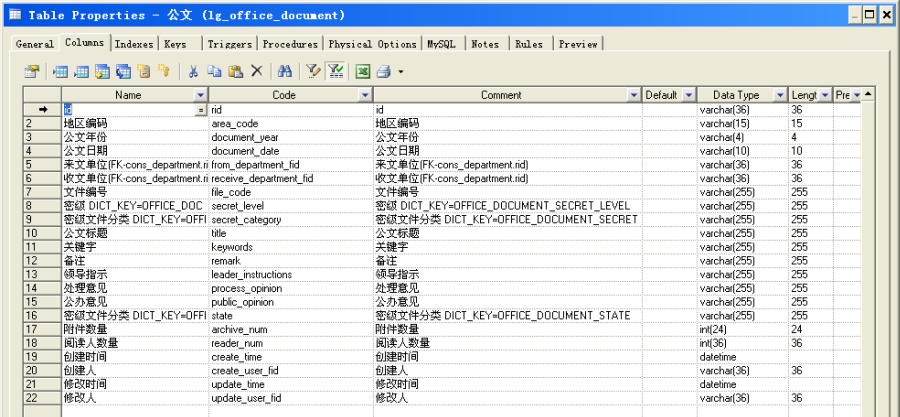
* 服务器的处理action链接地址：/main/index/ajaxFindGonggao
* 实现类：IndexController.ajaxFindGonggao(int, Model)



* 表名称: lg\_office\_document
* 按照通知时间，倒序方式加载5条件记录进行显示

### 报道

* 服务器的处理action链接地址：/main/index/ajaxFindNews
* 实现类：IndexController.ajaxFindNews(int, Model)



* 表名称: lg\_office\_document
* 按照报道时间，倒序方式加载5条件记录进行显示
* IndexController.ajaxFindWorklistByDay(String, int, String, Model)

|  |  |  |
| --- | --- | --- |
| 类型 | 数据标志 | 表名称 |
| 备忘录 | TX | lg\_office\_plan\_memo |
| 领导的行程安排 | RC | LG\_OFFICE\_SCHEDULE |

## 工作日历

* 按月方式显示当前当前登录人的备忘录和导航信息，日历上面暂时不显示每日的备忘信息。
* 表名称：
* js方法名称：doFindWorklist(day, size)
* 服务器的处理action链接地址：/main/index/ajaxFindWorklistByDay

## 右上角代表提醒

* 针对不同的角色，右上角显示的内容是不同的。

|  |  |
| --- | --- |
| 科室类型 | 区别 |
| 财务科 | E:\民政\项目交接文档\马文明\01kanban\0101global\05_right_top_caiwu.png |
| 领导 | E:\民政\项目交接文档\马文明\01kanban\0101global\05_right_top_master.png |
| 其他科室 | E:\民政\项目交接文档\马文明\01kanban\0101global\05_right_top_other.png |

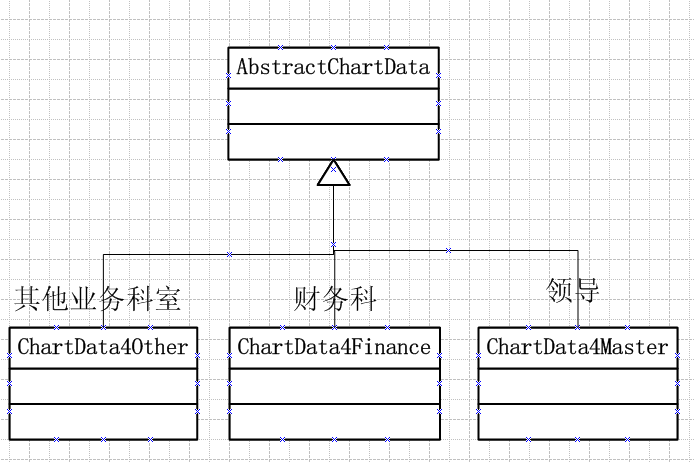
* 领导的审批待办数量
  + 实现类 IndexController.ajaxFindToDoReminder(String, String, Model)
  + 关键sql
    - 资金项目审批的代办数量
  + SELECT COUNT(1)AS SIZE FROM lg\_fund\_project\_department\_summarize
  + WHERE PROJECT\_STATUS=?
  + AND REGISTER\_YEAR\_MONTH =?
  + and AUDIT\_MAN =?
    - 预算审批
  + SELECT COUNT(1)AS SIZE FROM LG\_BUDGET\_DEPARTMENT
  + WHERE 1 = 1
  + AND STAT=?
  + AND EXISTS(SELECT 1 FROM CONS\_DEPARTMENT D WHERE (D.MASTER\_USER\_FID=? OR D.CHARGE\_USER\_FID=?) AND D.RID=DEPARTMENT\_FID)
    - 临时项目审批
  + SELECT COUNT(1)AS SIZE FROM lg\_temporary\_fund\_project
  + WHERE PROJECT\_APPLICATION\_STATUS=?
  + AND REGISTER\_YEAR like ?
  + and AUDITOR\_MAN =?
* 各个科室的预算和发放金额
  + 财务科
    - 实现类 IndexController.ajaxFindFinanceBudgetStat(Model)
  + 查询发放金额
  + SELECT SUM(RESPOND\_HANDLE\_SUM)AS a FROM lg\_fund\_project\_grant c
  + WHERE AREA\_CODE LIKE ?
  + AND REGISTER\_YEAR\_MONTH LIKE ?
  + AND HANDLE\_MARK=?
  + 查询预算金额
  + SELECT BUDGET\_MONEY,STAT FROM LG\_BUDGET\_GLOBAL
  + WHERE AREA\_CODE LIKE ?
  + AND YEAR like ? limit 1
  + 查询科室预算
    - 通过查询科室预算表，来查询这个科室的本年度的科室预算金额情况

## 工作导航

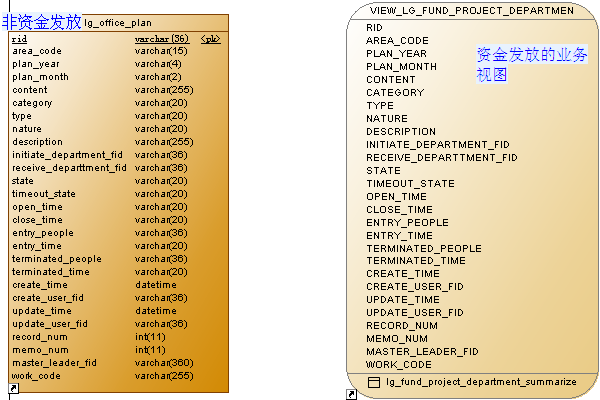
### 概述

* 基于工作的量化情况，进行的统计信息。主要用于
* 服务器的处理action链接地址：/main/index/ajaxChartLoadStat
* 实现类：
  + IndexController.ajaxChartLoadStat(SearchForm, Model)
  + 因为几类用户查询的数据有一些差异，所以真的图像的数据获取实现类进行了单独定义
  + 几个实现类的区别

|  |  |  |  |
| --- | --- | --- | --- |
| 类名称 | 人员类别 | 主要查询表 | 查询条件 |
| ChartData4Master | 领导 | 相同 | 当前领导分管的科室 |
| ChartData4Finance | 财务科 | 相同 | 财务科发起的工作 |
| ChartData4Other | 业务科室 | 相同 | 我接收到的工作 |



* 操作表：
  + 非资金发放的相关工作情况 lg\_office\_plan
  + 资金发放的相关工作情况 VIEW\_LG\_FUND\_PROJECT\_DEPARTMENT\_SUMMARIZE
  + 把两类表的数据union合并起来进行统计。

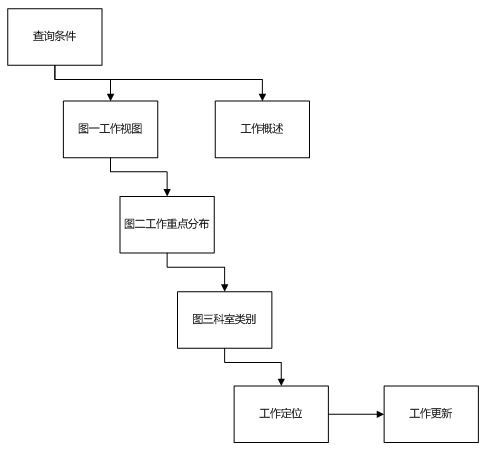


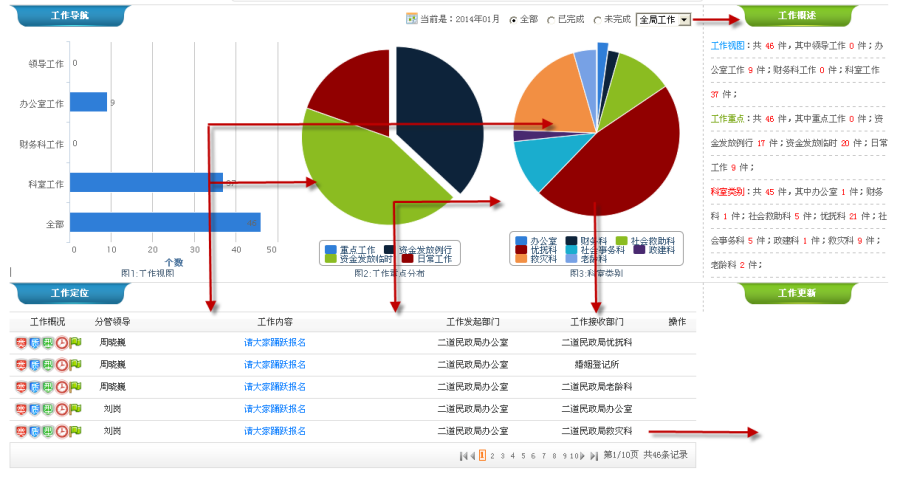
### 使用的控件

* highcharts-3.0.7版
* 同时进行了图形的点击事件触发，向下的方式进行触发加载，逐级缩短数据的检索范围。

### 联动操作的关系

* 每一个触发的动作，都是一个连续的数据的加载过程。
* 比如：点击图1工作视图的某个工作类别，会同时加载图2，图3和工作定位的数据进行同步刷新显示。





### 图一工作视图

* 按照几类工作工作类别进行统计，这几类工作类别定义在数据字典OFFICE\_PLAN\_TYPE中。

SELECT B.TYPE,COUNT(1)SIZE FROM view\_lg\_fund\_project\_department\_summarize B

where xxxsx

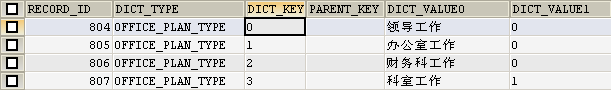
group by b.type

union all

(SELECT A.TYPE,COUNT(1)SIZE FROM LG\_OFFICE\_PLAN A WHERE

a.area\_code LIKE ? and a.plan\_year=? and a.plan\_month=?

group by a.type

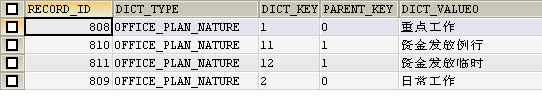


* 字典分组值 :group by OFFICE\_PLAN\_TYPE



### 图二工作重点分布

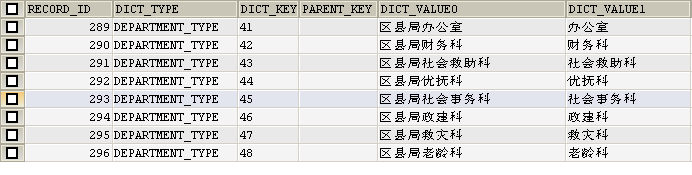
* 字典分组值 :group by OFFICE\_PLAN\_NATURE

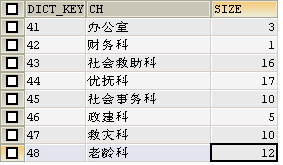




### 图三科室类别

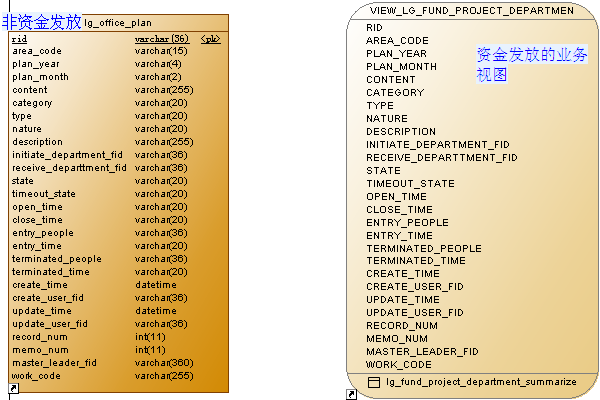
* 只处理科室类别是区县级的部门“DICT\_KEY LIKE '4%'”
* 字典分组值 :group by DEPARTMENT\_TYPE





## 工作定位

* 根据工作导航的三个图标来确定查询的范围，也就是查询条件的不同。
* js方法名称 function doLoad1WorkLocation(pageIndex)
* 服务器的处理action链接地址：/main/index/ajaxQuery1WorkLocation
* 实现类：IndexController.ajaxQuery1WorkLocation(SearchForm, HttpServletRequest, HttpServletResponse)
* 操作表：
  + 非资金发放的相关工作情况 lg\_office\_plan
  + 资金发放的相关工作情况 VIEW\_LG\_FUND\_PROJECT\_DEPARTMENT\_SUMMARIZE
  + 把两类表的数据union合并起来进行统计。

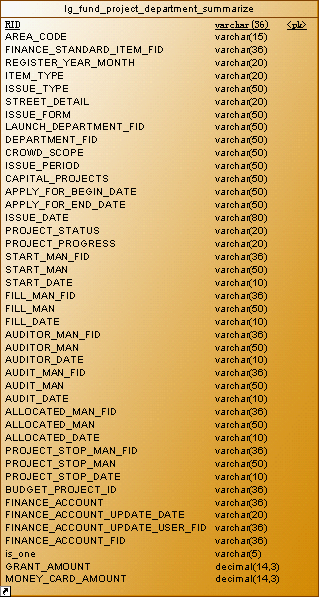


* 只有当前领导是当前记录所在科室的分管领导，则允许该领导输入“备忘录”

## 工作更新

工作更新的内容的加载，取决于左侧的“工作定位”中，每行的的“CATEGORY”列标识的“资金”和“非资金”的区别，进一步加载的右侧的工作更新的内容。

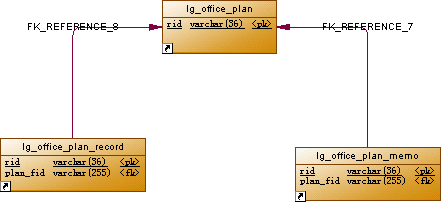
* 资金发放工资
  + 查询表lg\_fund\_project\_department\_summarize，



* + 因为表lg\_fund\_project\_department\_summarize中，对工作进度中的不同的几个阶段都在同一行记录的不同的表字段中进行了记录，因此，在显示的时候，需要根据最新的阶段时间来进行显示。
  + 记录关键阶段的对应配置信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 状态 | 排序 | 部门列 | 人员名称列 | 发生时间列 |
| 项目终止 | 项目终止 | 1 | PROJECT\_STOP\_MAN\_DEPT | PROJECT\_STOP\_MAN | PROJECT\_STOP\_DATE |
| 拨付 | 资金拨付 | 2 | ALLOCATED\_MAN\_DEPT | ALLOCATED\_MAN | ALLOCATED\_DATE |
| 审批 | 项目审核 | 3 | AUDITOR\_MAN\_DEPT | AUDITOR\_MAN | AUDITOR\_DATE |
| 审核 | 项目审批 | 4 | AUDIT\_MAN\_DEPT | AUDIT\_MAN | AUDIT\_DATE |
| 填报 | 项目填报 | 5 | FILL\_MAN\_DEPT | FILL\_MAN | FILL\_DATE |
| 启动 | 项目启动 | 6 | START\_MAN\_DEPT | START\_MAN | START\_DATE |

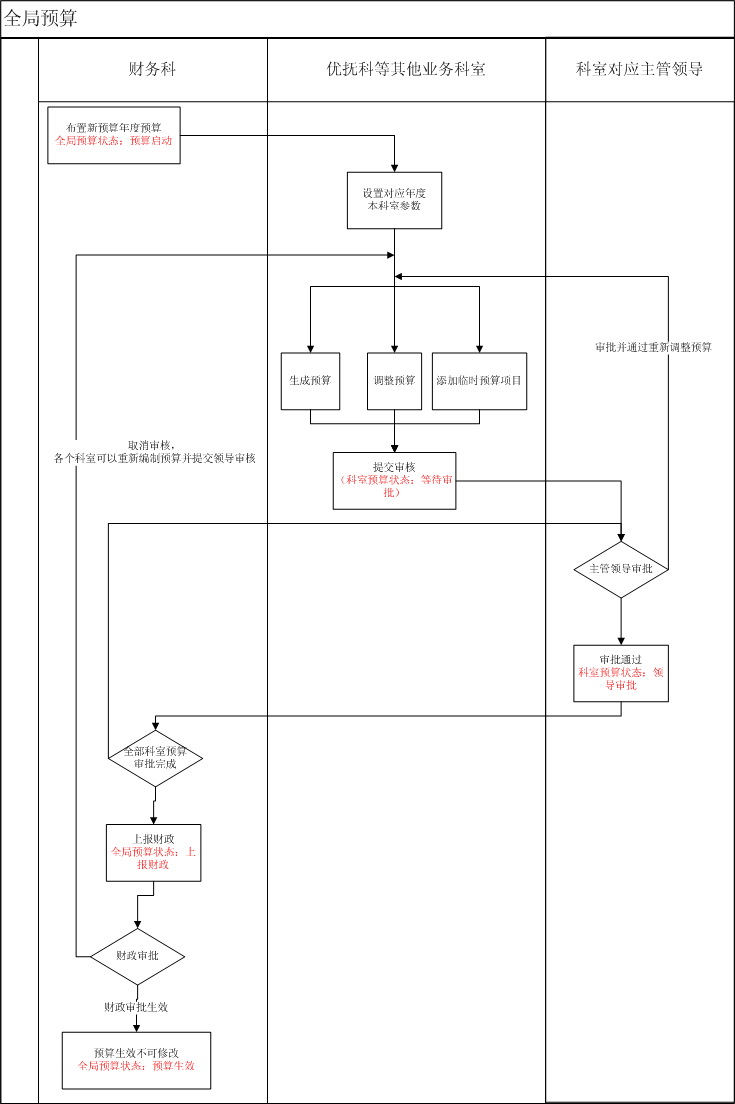
* + 服务器的处理action链接地址：/main/index/ajaxFindWorkUpdate1
  + 如果当前人是这行记录的分管领导，那么运行该领导填写备忘录
* 非资金发放工作
  + 服务器的处理action链接地址：/main/index/ajaxFindWorkUpdate2
  + 查询表lg\_office\_plan\_record



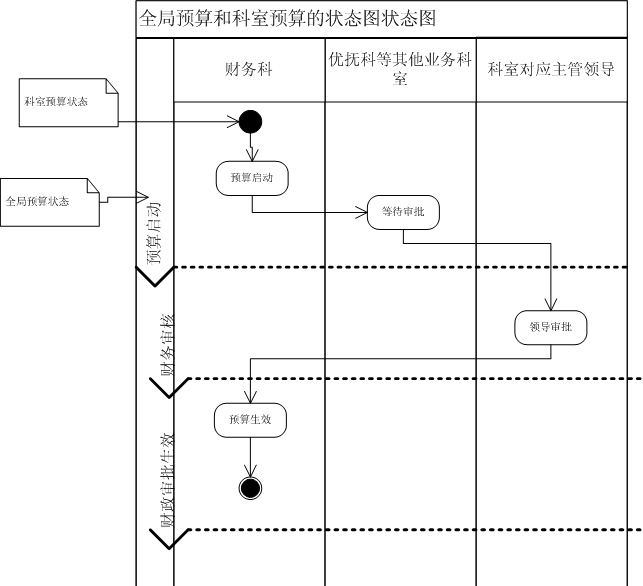
# 预算管理

## 业务流程图

### 跨组织结构流程图



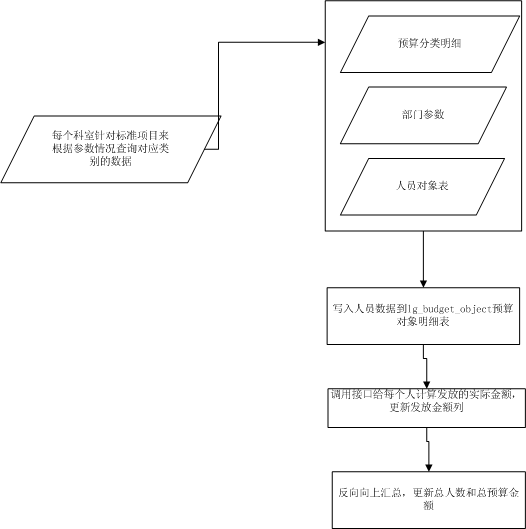
### 状态图



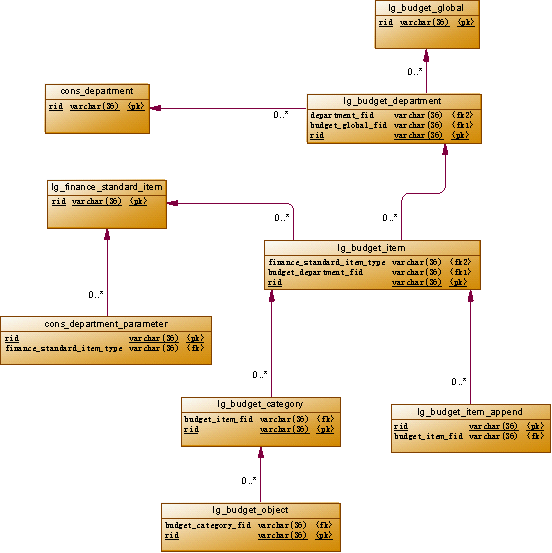
### 操作流程图

* 财务科
  + 每年年底一般为11月份，财务科进行布置预算的操作，布置下一年的预算
* 各个业务科室
  + 配置本科室的参数，主要调整新年度的各个项目的金额等信息。
  + 生成预算，关联对象表，生成每个项目的相关人员的预算金额信息
  + 调整预算
  + 增加临时预算项目
  + 提交主管领导进行审批
* 财务科
  + 业务科室的主管领导全部审批完成后，上报财政，
  + 标记为“财政审批生效”，该年度预算生效。
  + 追加预算（调节，全局预算已生效,年份>=当前年）

### 数据流程图



### E-R图（表间结构）



## 财务科布置预算

### 概述

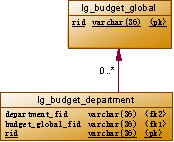
财务科的预算布置，每年年终布置一次，一般在11月份，由财务科发起，根据标准项目定义的项目项目信息为各个业务科室创建预算项目。其他的业务科室根据预算项目，生成各自项目的受众对象。

### 页面

#### 预览



#### 列表页的数据来源

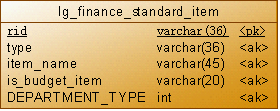


* 通过查询 全局预算表和科室预算表，来得到布置了预算的科室数量，已经各个状态下面的科室的数量情况。
* SELECT RID, AREA\_CODE, YEAR, UNSUBMIT\_DEPARTMENT\_AMOUNT, SUBMITTED\_DEPARTMENT\_AMOUNT, APROVED\_DEPARTMENT\_AMOUNT, BUDGET\_MONEY, STAT,
* (SELECT COUNT(1) FROM LG\_BUDGET\_DEPARTMENT D WHERE G.RID = D.BUDGET\_GLOBAL\_FID) AS COUNT\_ALL
* FROM LG\_BUDGET\_GLOBAL G
* WHERE 1 = 1
* AND g.AREA\_CODE like'220104%'
* ORDER BY YEAR

### 布置预算

#### 布置预算页面的数据来源

* 服务器处理类 Global1GlobalController.ajaxFinanceStandardItems()
* 通过ajax方式从服务器加载标准项目和业务科室的分组信息。
* 查询标准项目表，lg\_finance\_standard\_item

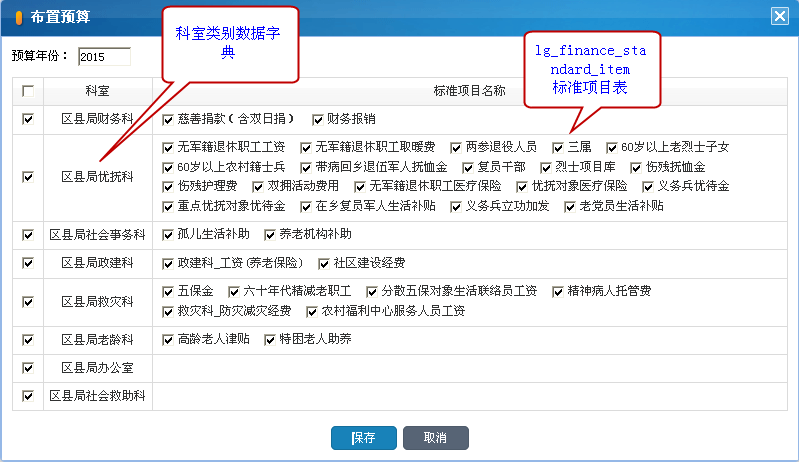


#### 查询规则

* is\_budget\_item是否预算项目为是的项目，非预算项目不处理
* 地区编码相同（todo，未处理）

#### 页面的显示规则

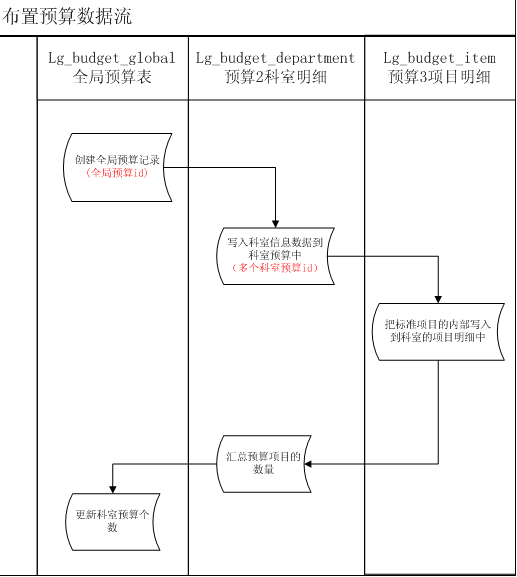
* 对标准项目，安装科室类别分组，相同的科室列表的项目放到一起进行显示。
* 关联数据字典，得到数据字典的编码和中文的对应关系
* 在页面中进行列表展现



#### 保存

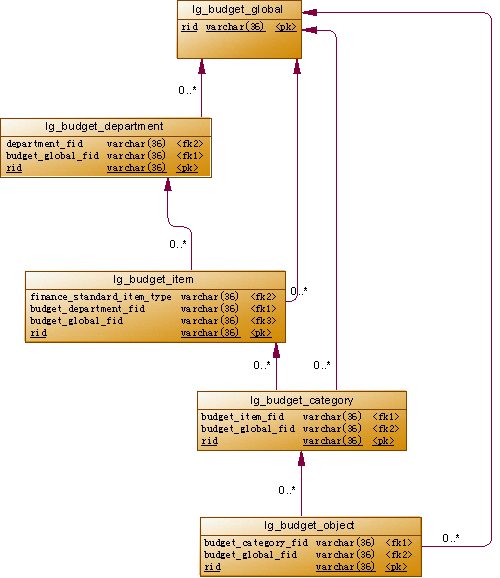
服务器端需要进行的操作：

* 创建全局预算记录，写入【Lg\_budget\_global全局预算表】，得到全局预算id
* 对本单位的业务科室创建科室预算，写入【Lg\_budget\_department预算2科室明细】，同时关联“全局预算id”
* 根据选择的标准项目的项目类别，关联标准项目表，写入每个科室中，能够进行预算的标准项目， 每个标准项目创建一条记录，记录到【预算3项目明细lg\_budget\_item】
* 针对每个科室，统计该科室的项目明细的数量，更新到【Lg\_budget\_department预算2科室明细】中；
* 汇总此全局预算id相关的科室数量，更新到【Lg\_budget\_global全局预算表】。



### 重新布置预算

* 当全局预算状态为“预算启动”状态的时候，财务科能够进行“查询布置预算”的操作。
* 根据全局预算id进行删除操作。



#### 操作的过程为：

* 根据选中的记录，来对这条记录对应的全局预算id的相关数据进行分别删除；
  + 预算5对象明细 lg\_budget\_object
  + 预算4分类明细 lg\_budget\_category
  + 预算3项目明细 lg\_budget\_item
  + 预算2科室明细 lg\_budget\_department
  + 预算1全局 lg\_budget\_global
* 删除完成后，其他的操作步骤同“布置预算”

#### 注意事项

* 重新布置预算后，各个业务科室自行生成的数据都会被清空，
* 每次生成的对象是不同的。

### 下一步->业务科室生成预算

需要各个业务科室进行如下的操作

* 生成新年度的部门参数
* 对本科室的预算进行生成操作，具体请参考[业务科室生成预算](file:///E:\%E6%B0%91%E6%94%BF\%E9%A1%B9%E7%9B%AE%E4%BA%A4%E6%8E%A5%E6%96%87%E6%A1%A3\%E9%A9%AC%E6%96%87%E6%98%8E\02fuwuyingyong\0201zongheliucheng\020101budget\02010103\02010103.html)

### TODO未实现的功能

* 财务科取消审核后，按理说，需要各个业务科室能够再次进行编辑操作，但是，因为需要经过领导审批后，财务科才能够上报财政，所以，即使全局预算状态改为取消审批，但是科室预算状态仍然是“领导审批”，并且，领导审批后状态没法回退，因此，科室仍然无法进行“再次生成预算数据”的操作。

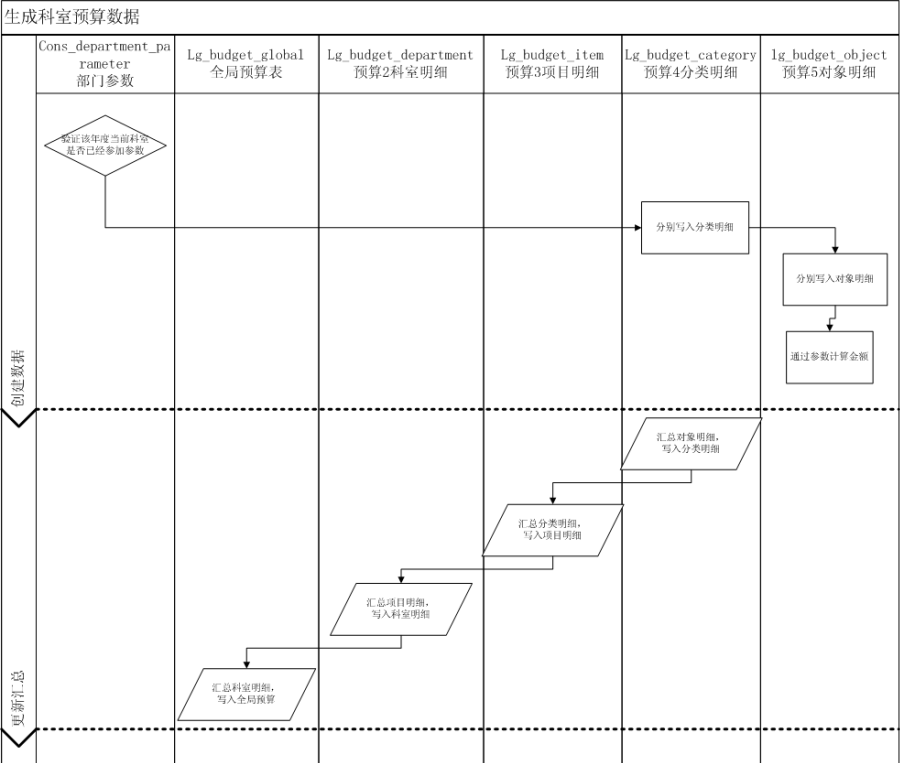
## 业务科室生成预算

### 概述

* 以区县局科室的身份登录系统，在“服务应用”->“预算管理”->“科室预算”模块中进行相关操作。



* 页面分为了四个部分
  + 全局预算，虽然叫全局预算，但是现实的内容，是当前科室的各个年代的预算情况，同财务科“全局预算”中5个页签的第二个页签“科室预算”的内容一致。但不是一套jsp页面。
  + 项目明细，由查看按钮触发进入，传入当前行的记录id和全局预算id。
  + 分类明细，由查看按钮触发进入，传入当前行的记录id和全局预算id。
  + 对象明细，由查看按钮触发进入，传入当前行的记录id和全局预算id。
* 业务科室的预算生成操作包括2个部分
  + 生成操作
  + 逐级数据汇总



### 生成操作

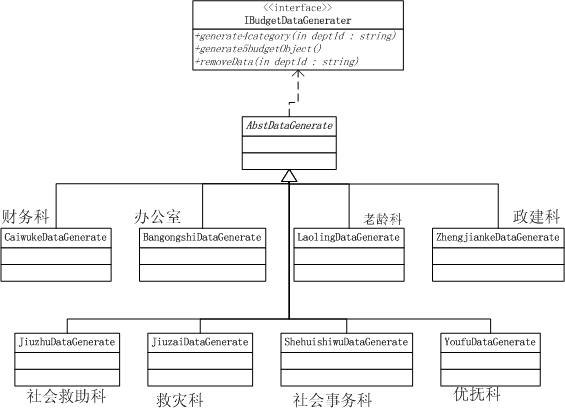
#### 页面入口

* 页面中有2个入口点来提供对当前业务科室的预算数据生成操作。
  + “全局预算”页签的【生成】按钮
  + 从“全局预算”页签点击“查看”后，进入到“项目明细”页签，页面左下角的【重新生成】



#### 科室预算的生成

* 页面的响应处理类：BudgetController.ajaxDoReCreate(String, HttpServletRequest)
* 针对不同科室的业务规则不同，统一创建了IBudgetDataGenerater接口进行数据的生成操作。
  + 针对不同的科室类别，来加载对应的实现类进行操作。



* + 对象的数据生成都是基于标准项目的类别来操作，针对每一个标准项目类别，创建对应的实现类。
  + 能够生成的数据包括
    - 生成预算4分类明细的数据，详见[创建“预算4分类明细”数据的思路](file:///E:\%E6%B0%91%E6%94%BF\%E9%A1%B9%E7%9B%AE%E4%BA%A4%E6%8E%A5%E6%96%87%E6%A1%A3\%E9%A9%AC%E6%96%87%E6%98%8E\02fuwuyingyong\0201zongheliucheng\020101budget\02010103\创建)
    - 生成预算5对象明细的数据，详见[#基于标准项目生成“预算5对象明细”](file:///E:\%E6%B0%91%E6%94%BF\%E9%A1%B9%E7%9B%AE%E4%BA%A4%E6%8E%A5%E6%96%87%E6%A1%A3\%E9%A9%AC%E6%96%87%E6%98%8E\02fuwuyingyong\0201zongheliucheng\020101budget\02010103\02010103.html#基于标准项目生成)
    - 逐级汇总相关分类的预算金额，详见[#逐级汇总数量和预算金额](file:///E:\%E6%B0%91%E6%94%BF\%E9%A1%B9%E7%9B%AE%E4%BA%A4%E6%8E%A5%E6%96%87%E6%A1%A3\%E9%A9%AC%E6%96%87%E6%98%8E\02fuwuyingyong\0201zongheliucheng\020101budget\02010103\02010103.html#逐级汇总数量和预算金额)
    - 完成本科室的生成动作

#### 创建“预算4分类明细”数据的思路

* lg\_budget\_category预算4分类明细表，主要用来分类方式存储预算对象的类别信息情况。



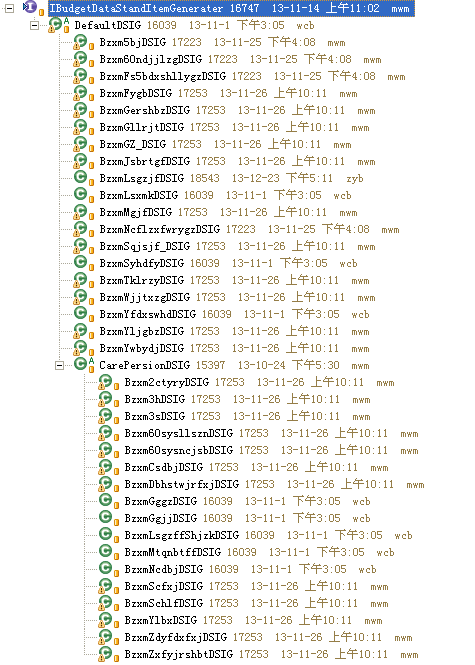
* 主要用来和参数表的相同标准项目进行匹配操作，插入到预算4分类明细表中。



#### 基于标准项目生成“预算5对象明细”

* 不同的业务科室，有不同的标准项目
* 每个标准项目，对应不同的受众对象。
* 因此，针对每一个参与预算的标准项目，开发针对这类项目的对象加载方式，写入到表“lg\_budget\_object预算5对象明细”中。

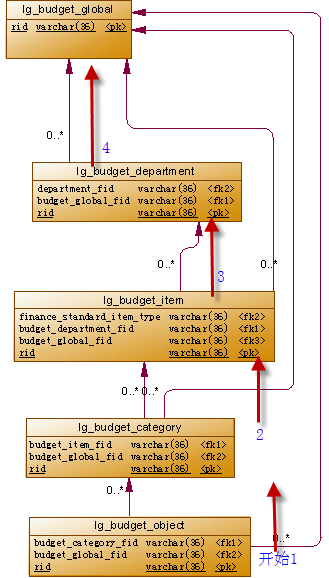




* 生成过程
  + 关联参数表，通过参数的类别和对象表的相关类别关联起来，来确定符合参数定义的对象范围。
  + 通过insert into select 方式直接基于数据库操作，把符合的对象写入到预算对象表中。
  + INSERT INTO lg\_budget\_object ( ...... )
  + select ......
  + FROM 相关对象表 sp, lg\_budget\_category c, cons\_department\_parameter p
  + WHERE 1 = 1
  + AND c.department\_parameter\_fid = p.rid
  + AND SP.area\_code LIKE CONCAT( P.AREA\_CODE,'%') 地区编码
  + AND sp.is\_out\_of\_care=? 过滤条件
  + AND p.finance\_standard\_item\_type =? 标准项目id
  + 基于系统的金额计算函数，对插入的对象计算每个对象需要发放的实际金额

#### 逐级汇总数量和预算金额

* 从下往上逐级进行汇总更新操作
  + 首先对“lg\_budget\_object预算5对象明细”进行汇总，更新到“lg\_budget\_category预算4分类明细”表中的统计列 ，
    - amount数量
    - budget\_money 预算金额
  + 再对“lg\_budget\_category预算4分类明细”进行汇总，更新到“lg\_budget\_item预算3项目明细”表中的数量和金额列：
    - amount数量，
    - budget\_money1 预算金额1
    - BUDGET\_APPEND\_AMOUNT，追加预算笔数
    - BUDGET\_APPEND\_MONEY，追加预算金额
    - budget\_money2 汇总后的预算金额
  + 然后对“lg\_budget\_item预算3项目明细”进行汇总，更新到“lg\_budget\_department 预算2科室明细”表中的数量和金额列
    - itemTypeCount 科室的项目数量
    - budget\_money 预算金额
  + 最后对“lg\_budget\_department 预算2科室明细”进行汇总，更新到“lg\_budget\_global预算1全局”表中
    - budget\_money 预算金额



### 下一步

调整调整项目明细

## 调整项目明细

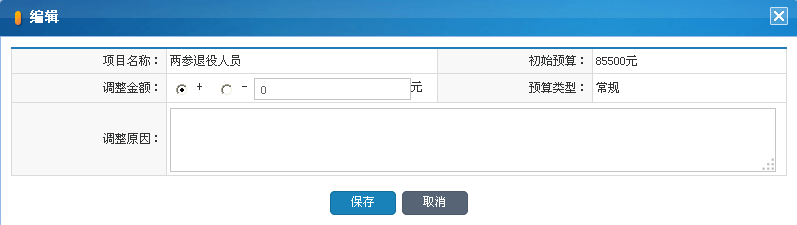
### 概述

由各个业务科室进行操作。主要是调整项目项目的预算金额。

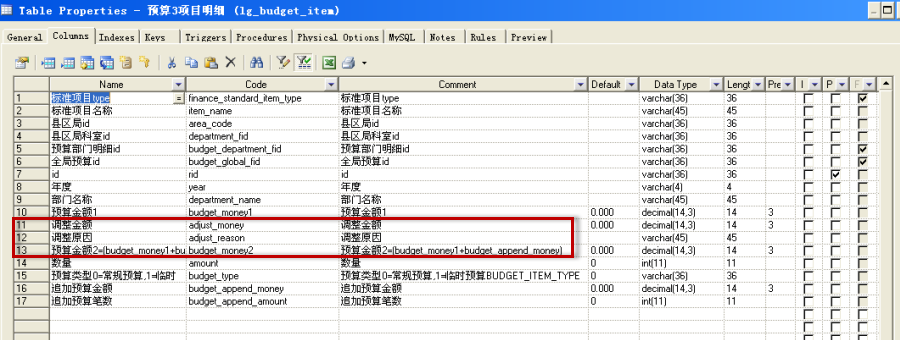


### 编辑

* 实现类 Dept3ItemController.ajaxAdjustBudgetMoney(String, double, String, HttpServletRequest)
* 通过对预算类型为“常规”的项目明细进行“编辑”操作，来达到调整预算金额的目的。
* 调整的对话框



* 需要写入到表“预算3项目明细lg\_budget\_item”中，表的结构为：



* 操作的过程为：
  + 对应页面中输入的“调整金额”和“调整原因”被分别写入到字段“adjust\_money”，“adjust\_reason”中。
  + 同时调用计算调整后金额，写入到“budget\_money2预算金额2”中
    - 公式 budget\_money2=(budget\_money1+budget\_append\_money)
  + 逐级更新预算金额合计，请参考 [逐级汇总数量和预算金额](file:///E:\%E6%B0%91%E6%94%BF\%E9%A1%B9%E7%9B%AE%E4%BA%A4%E6%8E%A5%E6%96%87%E6%A1%A3\%E9%A9%AC%E6%96%87%E6%98%8E\02fuwuyingyong\0201zongheliucheng\020101budget\02010103\02010103.html#逐级汇总数量和预算金额)
    - 更新科室预算合计
    - 更加全局预算合计

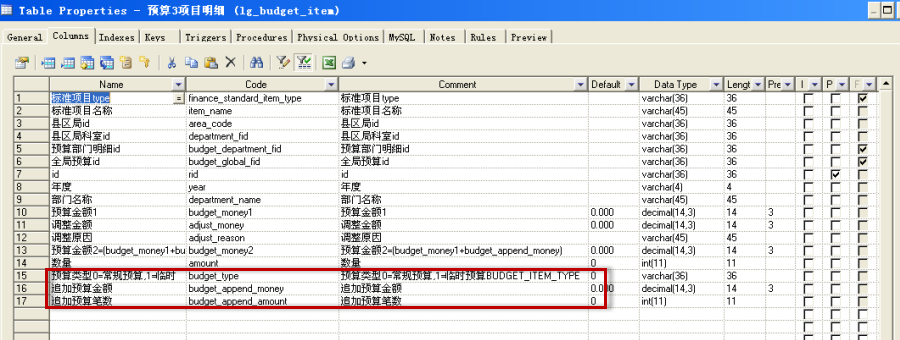
## 添加添加临时预算项目

### 概述

* 对于某些未列在标准项目在预算数据，可以通过【添加临时预算项目】操作来进行添加。



* BudgetController.ajaxDoCreateTempBudget(String, String, double, String, HttpServletRequest)



* 操作的过程为：
  + 直接新建记录写入数据到“预算3项目明细lg\_budget\_item”中，
  + 通过制定budget\_type来区分预算类型0=常规预算,1=临时预算BUDGET\_ITEM\_TYPE
  + 不需要创建分类明细和对象明细数据。
  + 逐级更新预算金额合计，请参考 [逐级汇总数量和预算金额](file:///E:\%E6%B0%91%E6%94%BF\%E9%A1%B9%E7%9B%AE%E4%BA%A4%E6%8E%A5%E6%96%87%E6%A1%A3\%E9%A9%AC%E6%96%87%E6%98%8E\02fuwuyingyong\0201zongheliucheng\020101budget\02010103\02010103.html#逐级汇总数量和预算金额)
    - 更新科室预算合计
    - 更加全局预算合计

## 业务科室提交分管领导审核

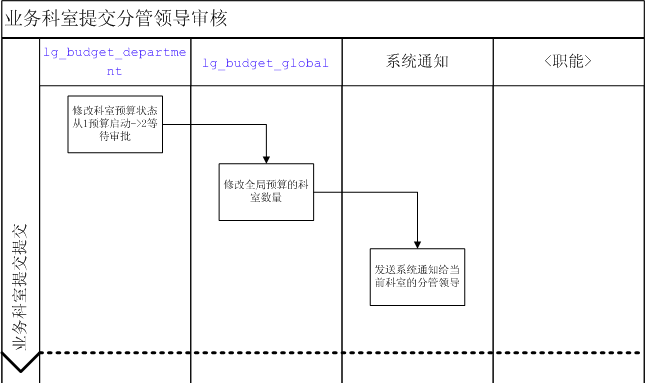
### 概述

* 当本科室的财务预算编制完成后，需要提交给科室的分管领导进行审核！



### 操作

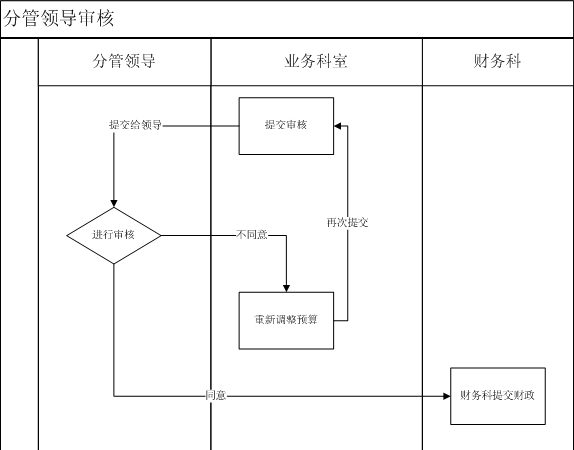
* 在“科室预算”的“全局预算”页面，在本科室的年度预算列表中，预算状态标记为“预算启动”的科室预算，允许通过【提交审核】方式，提交给本科室的分管领导进行审核操作。
* 实现类 Dept2DepartmentController.ajaxDoSubmitDeptData(String, HttpServletRequest)



## 分管领导审核

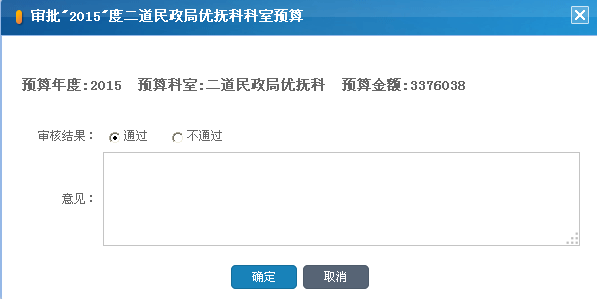
### 概述

* 分管领导能够对自己所负责分管的几个业务科室提交上来的预算进行审核操作，





* 对科室的预算进行审核，审核结果有两种
  + - 通过，预算不可修改，提交给财务科等待上报财政审核
    - 不通过，预算返回业务科室重新调整后，可以再次提交审核



#### 操作

* 实现类： Global2DepartmentController.ajaxDoChargeMasterVerifyed(String, String, String)
* 更新科室预算状态
  + 同意，状态由"2等待审批"->"3领导审批"
  + 不同意，状态由"2等待审批"->"1预算启动"
* 更新全局预算中的提交和审批数量的统计值字段。

## 财务科统一处理

### 概述

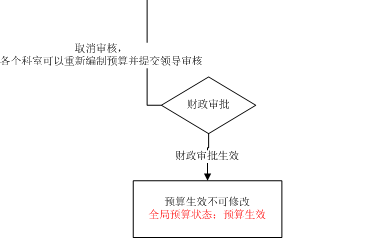
* 几个业务科室的预算被主管领导审核后，最后统一汇总到财务科，由财务科提交给财政进行审批，
* 几个连续的步骤包括
* 上报财政->

### 上报财政

* 提供上报财政操作的条件
  + 相关的业务科室都已经审核通过了，通过 count==lg\_budget\_global.aproved\_department\_amount 进行判断



* + 实现类：Global1GlobalController.doChangeBudgetBeSubmitCZ(String, HttpServletRequest)
  + 动作为 修改全局预算的状态从 "1预算启动"->"2财务审核"



### 取消审核

* 只有当全局预算的状态是“2财务审核”的状态的时候才提供。
* 实现类：Global1GlobalController.doCancelFinancecAudit(String, HttpServletRequest)
* 动作为 修改全局预算的状态从 "2财务审核"->"1预算启动"

### 财政审批生效

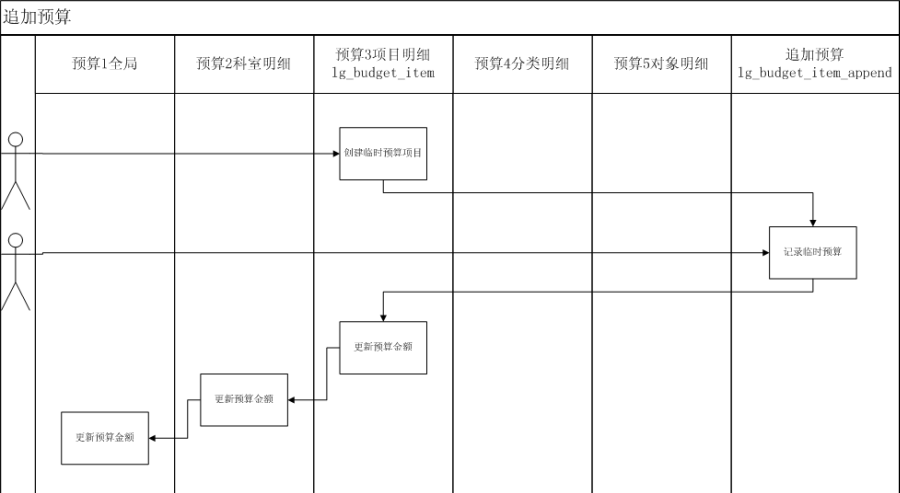
* 只有当全局预算的状态是“2财务审核”的状态的时候才提供。
* 实现类：Global1GlobalController.ajaxDoChangeBudgetReadonly(String, HttpServletRequest)
* 动作为
  + 修改全局预算的状态从 "2财务审核"->"3财政审批生效"
  + 修改科室预算的状态 lg\_budget\_department.stat 为 "4预算生效"

## 预算追加

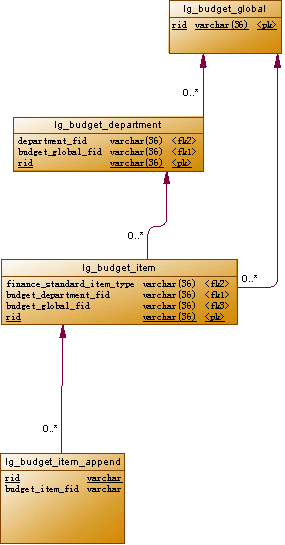
### 概述



* 预算追加的条件
  + 该预算已经生效，lg\_budget\_global.stat="3财政审批生效"
  + 预算年份>=当前年，比如现在是2014年，就无法调整2013年的生效的预算，只能调整当年和之后的意见生效的预算。
* 提供的操作有两种
  + 追加新增预算
  + 增加追加
* 操作流程



* er图



# 综合搜索

## 对象搜索

### 概述

对系统中各个业务科室所管理的对象进行一个统一的查询检索操作。

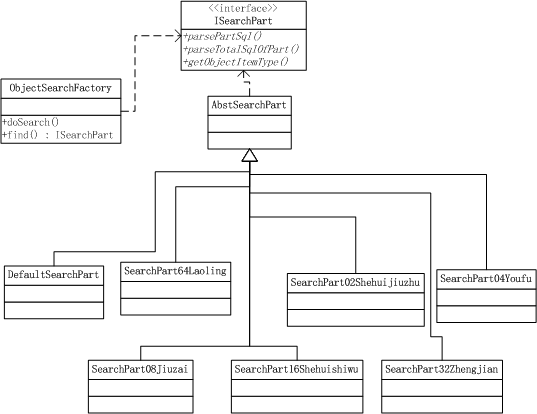


#### 类型

E:\民政\项目交接文档\马文明\02fuwuyingyong\0201zongheliucheng\020103objsearch\02010301\01_type.png

* 相当于查询的范围
* 提供的几个操作类型包括
  + 不限：
  + 社会救助
  + 优抚
  + 救灾慈善
  + 社会事务
  + 政建
  + 老龄

#### 类图



### 具体类型中查询范围的处理

* 页面的表象部分都是集中通过jsp文件进行处理的。基于数据字典进行加载相关的属性类。
* 页面选择的相关数据之间的关系在后台的SubGroup.java类进行分组和组合。
* 结构
  + "表名\_一级\_二级\_三级\_四级"这样的结构，之间用“\_”进行分割，通过这种方式，把相同父级的数据归结到同一个类别中
  + 服务器端根据这种结构，分别把每一个类别的数据进行集中合并。
  + 调用
  + MapList<String, SubGroup> types = SubGroup.parseTypes(form, "",
  + "T1\_SHEHUIJIUZHU\_REN\_UTYPE1\_in\_split",
  + "T1\_SHEHUIJIUZHU\_REN\_UTYPE2\_in\_split",
  + "T1\_SHEHUIJIUZHU\_REN\_UTYPE3\_in\_split");

### 多个类别的复合搜索

程序中采用union all 的方式把每一个独立类别的sql合并为一个整体。每一个实现类各自生成查询本类别的能够独立运行的sql语句后在ObjectSearchFactory.doSearch的调用中被 通过union 方式合并为一个整体的sql。

### 总记录数

原始版本是通过select count(1)from (select a,b,c from table where 1=1 and ...)的方式来计算总记录数的，效率很低。先版本是通过接口方式来实现，有各个具体查询的实现类内部进行记录数量的计算，通过

select count(1)AS TOTAL\_COUNT from table wehre ....

的方式计算本类别的总记录数，然后,在外部的调用中，采用

select sum(TOTAL\_COUNT) AS QC FROM

(

( select count(1)AS TOTAL\_COUNT from table1 wehre .... )

UNION ALL

( select count(1)AS TOTAL\_COUNT from table2 wehre .... )

)V

通过求sum的方式，计算每一个子查询的总计列数量的方式来进行实现。

### 年龄

通过以下的公式来计算出生日期这个查询条件。

(

YEAR(当前时间或者死亡时间) - YEAR(出生日期列) - (RIGHT(当前时间或者死亡时间, 5) < RIGHT(出生日期列, 5))

)

通过以上的方式来得到年龄的公式。.

### 姓名拼音

对于有拼音列的数据表，支持在进行姓名查询的同时，对拼音也一并进行检索操作。

# 参数设置

## 概述

各个业务科室，在编制下一年度的预算的时候，需要对本科室的下一年度的参数进行设置，便于灵活的调整各个金额的变化情况。

### 其他业务科室

* 对于其他业务科室，使用的操作界面都是相同的。



### 优抚科

* 优抚科在“其他业务科室”的基础上，有另外4列的设置信息。分别是
  + 补发时长(年)
  + 补发金额
  + 市政府补发金额
  + 区政府补发金额

# 工作办公

## 工作计划

### 概述

* 工作计划分为2部分：
  + 非资金发放工作：用来对日常的工作办公情况进行布置和记录。
  + 资金发放工作:基于资金的时间发放情况进行的工作情况记录，不提供对此类工作的新增操作，可以到“资金发放”的功能中进行增加操作。
  + 非资金发放的数据库表为lg\_office\_plan
  + 资金发放工作的数据库表为：lg\_fund\_project\_department\_summarize，为了能够和lg\_office\_plan表的结构一致，采用加工后的视图进行关联view\_lg\_fund\_project\_department\_summarize
  + 这两类工作计划的数据，提供了首页页面中，图形的查询统计功能的数据实现。
* 提供的功能包括：

对任意的工作计划的“工作记录”的添加。对工作计划进行备忘录添加。

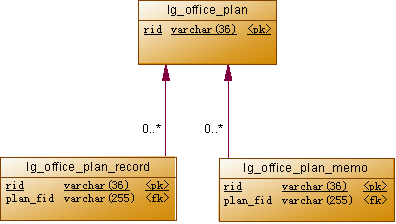
* 计划的添加

用户可以根据级别给本科室或者其他科室分别工作计划。计划的完成与否由工作的发起部门进行标记为完成。添加时，如果涉及到多个部门，则根据部门的数量，为每一个部门创建一条工作记录。

* 备忘录
  + 计划的源或者目的科室中的工作人员，每个人员可以给自己记录备忘录。记录的备忘录只有自己可见，其他人看不到。
* 工作记录
  + 围绕一件工作的开展情况进行记录，这件工作的相关科室的工作人员都可以进行记录，
  + 记录的内容，可以被其他人查阅
* 工作记录的批示
  + 接收部门的分管领导，可以对工作记录信息进行批示操作。
  + 批示的内容会同步显示在“工作记录”的列表中。3

### E-R图（细化流程图）

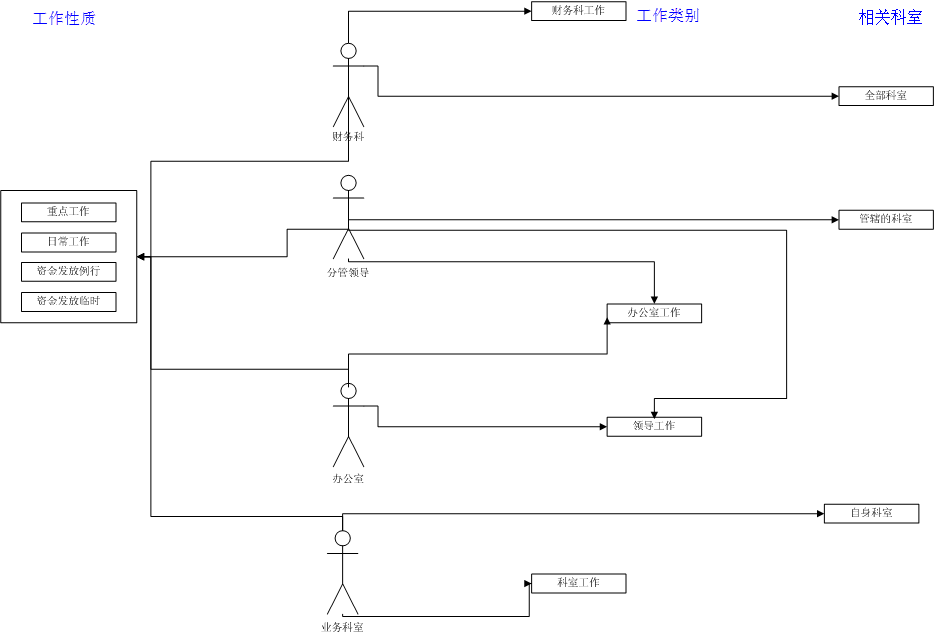
* 非资金发放



* 资金发放
  + 不提供添加操作，纯粹基于实际的资金发放情况进行操作。

### 工作计划的总体流程

* 几类用户的操作范围



* 领导布置工作计划
* 办公室布置工作计划
* 财务科步骤工作计划
* 各个业务科室布置工作计划

### 数据字典

* plan计划中使用的数据字典包括

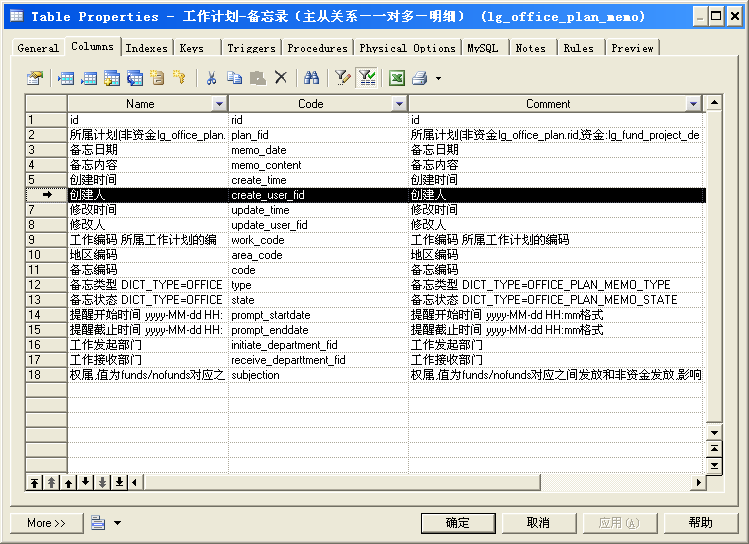
|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 数据字典 | 截图 |
| category | 工作类别 | OFFICE\_PLAN\_CATEGORY | E:\民政\项目交接文档\马文明\03office\0301overview\030101flow\OFFICE_PLAN_CATEGORY.png |
| type | 工作类型 | OFFICE\_PLAN\_TYPE | E:\民政\项目交接文档\马文明\03office\0301overview\030101flow\OFFICE_PLAN_TYPE.png |
| nature | 工作性质 | OFFICE\_PLAN\_NATURE | E:\民政\项目交接文档\马文明\03office\0301overview\030101flow\OFFICE_PLAN_NATURE.png |
| state | 工作状态 | OFFICE\_PLAN\_STATE | E:\民政\项目交接文档\马文明\03office\0301overview\030101flow\OFFICE_PLAN_STATE.png |
| timeout\_state | 超时状态 | OFFICE\_PLAN\_TIMEOUT\_STATE | E:\民政\项目交接文档\马文明\03office\0301overview\030101flow\OFFICE_PLAN_TIMEOUT_STATE.png |

* 工作记录中使用的数据字典包括

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 数据字典 | 截图 |
| type | 备忘类型 | OFFICE\_PLAN\_MEMO\_TYPE | E:\民政\项目交接文档\马文明\03office\0301overview\030101flow\OFFICE_PLAN_MEMO_TYPE.png |
| state | 备忘状态 | OFFICE\_PLAN\_MEMO\_STATE | E:\民政\项目交接文档\马文明\03office\0301overview\030101flow\OFFICE_PLAN_MEMO_STATE.png |

### 备忘录

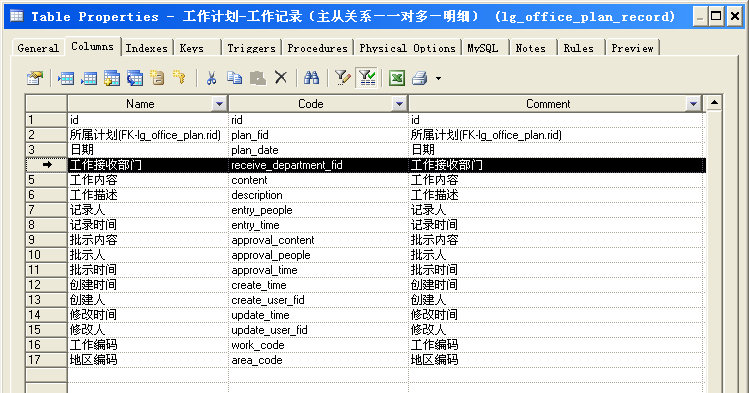
* 备忘录记录的信息是私人数据，其他人看不到。因此数据列表展现的过程中，需要把当前用户的userid也作为查询条件进行查询。
  + and create\_user\_fid=?





### 工作记录

* 操作表"lg\_office\_plan\_record";



## 公文管理

### 概述

* 相关表包括：

|  |  |
| --- | --- |
| 表名称 | 概述 |
| lg\_office\_document | 文档主表对象 |
| lg\_office\_document\_readers | 文档密级阅读权限表 |
| lg\_office\_document\_archives | 文件附件文件信息表 |
| lg\_archives | 附件档案 |

### 密级

密级分为3类，分别是：

* 无密级
* 秘密，并指定“密级文件分类”
* 机密，并指定“密级文件分类”
* 绝密，并指定“密级文件分类”

### 密级文件分类

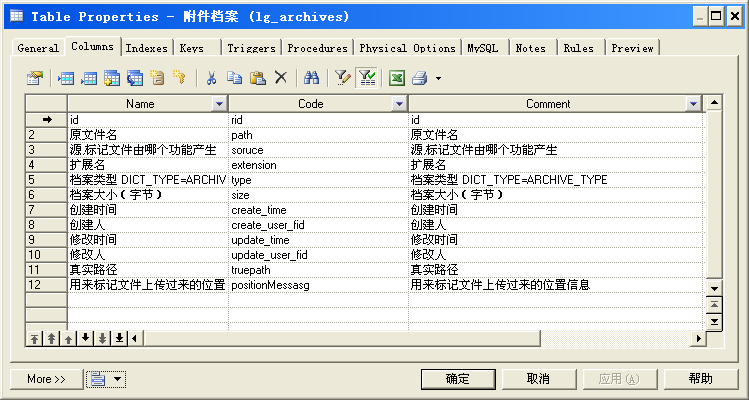
* 仅仅作为一个分类标识，标准这个密集对应的类别，无其他使用意义

### 可以阅读该密级文件的人

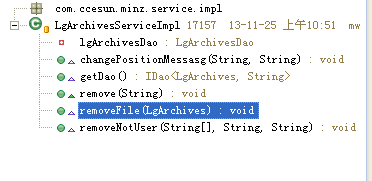
* 针对密集进行设置，该文件哪些人能够阅读，只有被设置为可以阅读的人才能够阅读这些文件。
* 其他人无法检索到该文件。
* 能够阅读文档的人员权限，数据被写入到表lg\_office\_document\_readers中

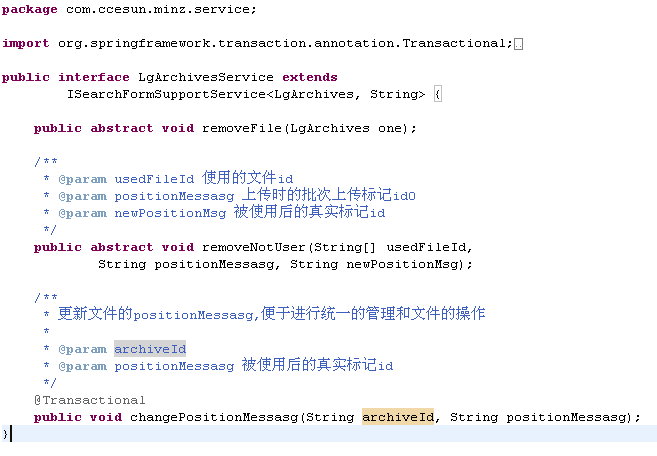
### 附件上传

* 使用的jquery插件uploadify
* 因为服务器对request进行了拦截，所以这里采用定制的jsp来进行文件的上传操作。
* /js/libs/ueditor-1.2.6.1/jsp/LgArchiveFileUp.jsp
* 表名称 lg\_archives



* 可以配置的参数包括
  + 'source' :文件来源，比如'officeDocumnt'
  + 'positionMessasg' :用来正向标识这个文档是从那个源id上面上传的，便于被定为。
  + 'currentUid' :当前的操作人员，用来记录上传者
* LgArchiveFileUp.jsp
  + 进行自定义的文档的上传操作
  + 上传文档到服务器，保存的上传文件夹，
  + 保存文档的相关信息，同时拾取'positionMessasg'信息，记录到lg\_archives表中
  + 文件上传成功后，返回文件id和相关路径信息给前端页面进行动态显示。
* 对上传文件的确认
  + 附件的文件数据，只是作为从属关系存在的，
  + 当主数据保存完成后，需要根据positionMessasg信息对附件进行更新，来标记这个附件从属于那个记录对象的id
  + 同时对于已经成功上传，但是未被采用的附件，执行完全删除操作。
* 删除操作
  + 删除分为2个动作
    - 对物理文件进行删除
    - 对附件表记录进行删除
  + 系统中提供的删除接口
    - 根据文档id删除
    - 对未使用的上传后的文档生成





### 公文查阅

* 重点就是通过sql语句来过滤密集，对于有密集的文档能够根据阅读人来进行查询到。
* SELECT
* (SELECT department\_fid FROM cons\_user WHERE rid = b.create\_user\_fid) AS CREATEDEPTID,
* RID, AREA\_CODE, DOCUMENT\_YEAR, DOCUMENT\_DATE, FROM\_DEPARTMENT\_FID,
* RECEIVE\_DEPARTMENT\_FID, FILE\_CODE, SECRET\_LEVEL, SECRET\_CATEGORY, TITLE,
* KEYWORDS, REMARK, LEADER\_INSTRUCTIONS, PROCESS\_OPINION, PUBLIC\_OPINION,
* STATE, ARCHIVE\_NUM, READER\_NUM, CREATE\_TIME, CREATE\_USER\_FID, UPDATE\_TIME, UPDATE\_USER\_FID
* FROM
* LG\_OFFICE\_DOCUMENT b
* WHERE 1 = 1
* AND (
* EXISTS (SELECT 1 FROM lg\_office\_document\_readers r WHERE r.read\_user\_fid =? AND r.document\_fid = b.rid)
* )
* OR (1 = 1 AND secret\_Level = '0')
* ORDER BY create\_time DESC

LIMIT 0, 10

## 消息收发

### 概述

* er图,目前设计的数据结构是一张消息主表lg\_office\_message，一张接收者子表lg\_office\_message\_rec
  + 因此一条消息，可以有n多个消息的接收者
  + 可以记录每个接收者的阅读情况。

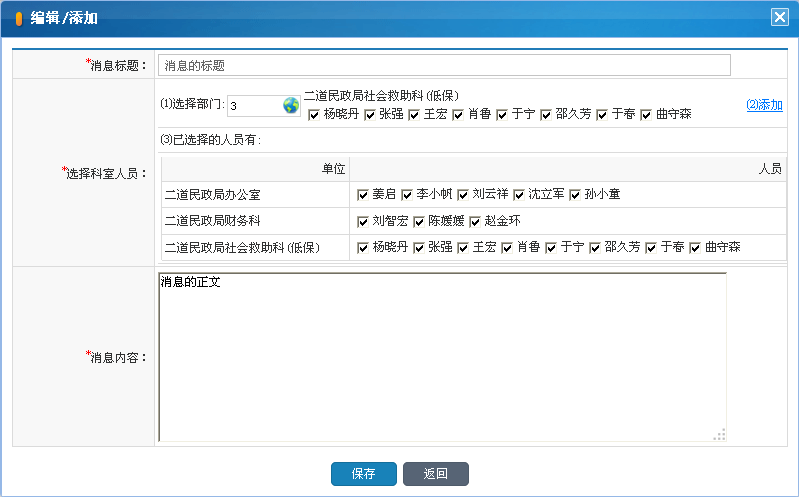


### 开发接口



* MessageHelper类，提供了几个调用接口来设置消息的正文和消息的接收人信息。
* 给科室发送消息
* public static String sendMessage2dept(String fromUserId,String titleSegment, String contentSegment,Map<String, Object> paramMap, String url, String... recDeptId)
* public static String sendMessage2dept(String fromUserId, String title, String content, String url, String... recDeptId)
* 给具体用户发送消息
* public static String sendMessage2users(String fromUserId,String titleSegment, String contentSegment,Map<String, Object> paramMap, String url, String... receiveUserIds)
* public static String sendMessage2users(String fromUserId, String title, String content, String url, String... receiveUserIds)
* 内容的表达式，这样的表达式,占位符区分大小写，占位符用变量代替，同时提供Map对象来设置占位符中的值。
* String msg="消息时间：【${now}】\n${deptName} ${userName} 布置 ${year}年年度预算。\n访问位置：【服务应用->预算管理->全局预算】";
* //对占位符进行赋值设置
* Map<String, Object> map = new HashMap<String, Object>();
* map.put("deptName", loginUser.getDepartment().getName());
* map.put("userName", loginUser.getName());
* map.put("year", year);
* map.put("now", MinzUtils.currentDateTime());
* 每个目标接收者的自己的消息都是通过关联接收者子表lg\_office\_message\_rec来实现的。
* 写入过程为：
  + 构建需要发送的消息的正文
  + 根据部门id等信息，加载部门中需要接收消息的用户的id
  + 统计消息的接收者信息
  + 写入消息正文到“消息主表lg\_office\_message”中
  + 写入接收人员信息到“接收者子表lg\_office\_message\_rec”中
  + 完成消息的存储操作。

### 新建消息



* 消息的存储操作，同开发接口定义的消息存储一致。

### 加载我的消息

* 系统通过当前人的userid作为查询关键字，关联“接收者子表lg\_office\_message\_rec”来判断哪些是发送给我的消息。
  + 根据发送时间进行排序

# 优抚对象管理

## 业务概述

优抚对象包括多个类别,如果修改优抚对象的优抚类别，则会自动将原来的优抚对象信息保存到历史中。如果优抚对象因死亡、迁出等原因，不再享受优抚待遇，则可通过停止享受待遇操作来实现。如果目前有正在处理的相关资金发放项目，则不能进行停止享受待遇操作。如果停止享受待遇的原因是优抚对象死亡，则需要计算抚恤金。

## 数据结构



lg\_care\_person：优抚对象

lg\_care\_person\_history：优抚对象历史。修改优抚对象类别时，需要将修改前的优抚对象信息保存到优抚对象历史表中，因此，一个优抚对象可能有0或多个历史记录。

lg\_care\_death\_money：抚恤金发放登记。优抚对象死亡时要发放抚恤金，因此一个优抚对象可能有0或1次抚恤金发放记录。

## 业务逻辑

### 新增



保存优抚对象。

### 修改



更新优抚对象；如果修改了优抚对象的类别，则需将修改前的优抚对象信息保存到历史表中。

### 停止享受待遇

#### 显示停止享受待遇对话框



检查有没有正在处理的（项目进度！=项目终止）优抚对象资金发放项目（包括例行和临时），如果有，则不能执行该操作。相关数据结构说明参见资金发放。

#### 选择停止享受待遇原因



如果停止享受待遇原因为优抚对象死亡，显示当年已经发放的资金（包括例行和临时）和当年所有相关例行资金发放项目剩余应发的资金。相关数据结构说明参见资金发放。

#### 保存



更新优抚对象的是否停止享受待遇、停止享受待遇日期、停止享受待遇原因；如果优抚对象死亡，保存抚恤金发放。

# 日常支出报销

## 业务概述



首先，由报销人通过日常支出报销功能填写报销申请，财务人员对等待财务科审核的报销申请进行审核，如果审核通过，则进入等待支付状态；否则，申请人对报销内容进行编辑，重新进入到等待审核状态。出纳人员通过银行转账或现金支付的方式完成支付后，将相应的报销申请设置为已支付状态。

## 数据结构

lg\_finance\_reimburse：日常支出报销

## 状态转换



## 业务逻辑

### 日常支出报销

插入到日常支出报销表。

### 财务审核

审核通过：设置日常支出报销状态为财务科审核已通过等待出纳支付。

审核未通过：设置日常支出报销状态为财务科审核未通过已退回。

### 出纳支付

设置日常支出报销状态为已支付。

# 财务帐户管理

## 业务概述

每个月财务科根据各业务科室提交的请款报告向区财政局申请资金，财政局会将款项打到相应的账户中，通过财务帐户管理可以对这些账户信息及使用状态进行维护；一般情况下，相关的资金项目对应同一个账户，目前是一个科室对应一个账户，可以通过标准项目账户管理来维护资金项目于账户之间的对应关系，在每年年初生成全年资金项目时，将会按照这个对应关系为当年的资金发放项目指定账户。如果在实际款项拨付时使用的账户发生变化，可以通过资金项目账户管理来指定具体资金项目实际使用的账户。目前为止，为资金项目指定财务帐户的目的主要是将来设计统计分析功能时作为一个维度来使用。

## 数据结构

lg\_financel\_account：财务帐户，财政局为申请的资金拨款时使用的账户。

lg\_finance\_standard\_item：标准（资金发放）项目，参见预算管理的数据结构说明。

lg\_fund\_project\_department\_summarize：公共-资金项目科室汇总表-月度，参见资金发放的数据结构说明。

## 业务逻辑

参考代码实现。

# 例行资金发放

## 业务概述

### 简述

财务会根据不同发放时间的发放项目，在年初统一生成当年所需要发放的资金项目，然后在每一个月份生成当月所需要发放的资金项目。生成的月度资金项目由业务科室用户核对，领导用户审核，财务用户拨付、一直到项目终止。

### 业务流程



### 状态转换

* 资金发放进度跟踪



* 资金发放乡镇进度跟踪



## 数据结构



* lg\_finance\_standard\_item：标准资金项目，对应实际业务中的一个资金发放项目。
* lg\_routine\_fund\_project：例行资金项目，标准项目在某一年的发放计划，包括该项目这一年分为几次发放，每次发放日期是什么。一个标准资金项目对应多个理性资金项目，每年一个。
* lg\_fund\_project\_department\_summarize：资金发放进度跟踪，用于跟踪一个项目在某个月的发放过程。有的项目不是按月发的。
* lg\_fund\_project\_town\_summarize：资金发放街道进度跟踪。如果资金发放流程包含街道一级角色，则需为每个街道生成对应的资金发放进度跟踪。
* lg\_fund\_project\_grant：银行转账信息。用于生成银行转账用的银行模板文件。
* lg\_fund\_project\_cash\_plan：请款、付款报告文书数据，用于生成请款报告文书。一个标准项目可能包含多个请款文书。
* \*\*\_grant\_detail：各种资金发放对象的资金发放明细。资金发放对象的详细信息，不同的对象对应不同的明细表。
* report\_\*\*：资金发放项目请款报告中涉及到的各种报表。

## 业务逻辑

### 生成资金项目

即生成本年度的例行资金发放计划：发放那些项目，什么时候请款，什么时候发放。标准项目中＜是否为例行项目＞为＜是＞的才可以生成，生成的例行资金项目的请款起止日期和发放日期默认来自标准资金项目，可以通过编辑功能进行修改。

当年的预算生成并生效之后，才可以生成资金项目；一旦生成之后不能重新生成。



生成公共-例行资金项目年度表数据的验证有：全局预算表是否存在指定年份的数据、全局预算的状态是否为财务审批生效（3）、当前年份是否已经生成了年度资金项目

### 启动例行资金发放流程

根据当年的资金发放计划生成本月的所有资金发放进度跟踪，对于乡镇参与的流程，同时生成资金发放乡镇进度跟踪。



生成资金项目时，会根据标准项目的发放周期和每一个标准项目的发放时间，来判断指定月份是否生成当前发放项目

### 街道生成请款数据

根据资金项目的发放对象和相关参数信息生成资金发放明细数据和各种报表，同时将乡镇资金发放进度跟踪（lg\_fund\_project\_town\_summarize）的进度(project\_progress)设置为数据已生成(2)。



### 街道取消生成

街道生成请款数据之后，未提交科室审核之前，或科室审核未通过，可以取消生成，修改有问题的数据，然后重新生成。

取消生成是生成请款数据的逆过程，会删除生成的数据，并将街道资金发放进度跟踪（lg\_fund\_project\_town\_summarize）的进度(project\_progress)恢复成生成请款数据之前的状态。

### 街道提交科室审核

将乡镇资金发放进度跟踪的项目进度设置为待审核，项目状态设置为等待审批。

### 补发

数据核对过程中对有错误发放信息的数据进行补发。



### 科室生成请款数据

#### 流程包含街道

所有街道都已提交审核之后才可以生成请款数据。

生成区级的报表数据，设置资金发放进度跟踪的项目进度（project\_progress）为数据已生成。



#### 流程不包含街道

生成区级的报表数据、资金发放明细数据；设置资金发放进度跟踪的项目进度(project\_progress)为数据已生成；乡镇资金发放进度跟踪的项目状态（project\_status）为领导审批、项目进度(project\_progress)为已审核。



### 科室生成请款报告

设置资金发放进度跟踪的项目进度（project\_progress）为<（已生成）请款报告>；生成银行转账数据；生成请款、付款报告文书数据。一个项目可能有多份请款、付款文书。



### 科室提交审核

设置资金发放进度跟踪的项目进度为待审核，项目状态为等待审批；通知科室主管领导进行审批。



### 主管领导审核

设置资金发放进度跟踪的项目进度为已审核，项目状态为领导审批；发送消息给财务用户；发送消息给科室用户。



### 财务支付

设置资金发放进度跟踪的项目进度为财务拨付，项目状态为财务拨付；发送消息给负责审核的主管领导；发送消息给科室用户。

如果存在支付失败的情况，需要将失败的银行转账信息标记为未成功。



### 标记未成功的银行转账记录

设置银行转账记录（lg\_fund\_project\_grant）的处理标志(handle\_mark)为失败。



### 遗留处理

如果存在处理失败需要重新和信息再处理的银行转账信息，可以进入遗留处理环节。

操作：设置资金发放进度跟踪的项目进度为遗留处理；发送消息给负责审核的主管领导；发送消息给科室用户。



### 处理未成功的银行转账记录

在此期间可以修改银行模板，对支付失败的对象重新支付，成功后将相应的转帐信息（lg\_fund\_project\_grant）处理标志设置为 (handle\_mark)。

### 更新银行卡号

对于财务发放有问题的对象（银行卡号错误），更新银行卡号。



更新公共发放明细表银行卡号的同时将数据汇总中查询的发放明细表银行卡号字段一起更新。

### 项目终止

如果财务支付完全成功或存在支付失败的情况但已处理，则可以标记项目终止。

设置资金发放进度跟踪的项目进度为项目终止，项目状态为项目终止。



# 高龄老人

## 业务概述

### 业务描述

高龄老人是指80周岁以上的老年人，享受高龄老人津贴，其中80-89周岁的高龄老人津贴由社会救助科以分类施保的形式进行发放，90周岁以上的高龄老人津贴由老龄科负责发放。高龄老人津贴每个季度发放一次，发放未来3个月的钱。工作人员每个月要通过各种形式对高龄老人进行探访，已确认其仍可以享受津贴。探访只针对90周岁以上的老人。如果通过探访得知高龄老人已去世或迁出，则停止向该人发放高龄津贴，同时，需要返还多发的钱。

## 数据结构



lg\_old\_person：高龄老人，用于高龄老人登记；津贴返还信息也记录在该表中。

lg\_old\_person\_change：高龄老人变动。记录高龄老人停发、99岁变百岁，每次变动一条记录。

lg\_old\_person\_visit：高龄老人探访记录，记录每个月的探访结果，一次一条记录

## 高龄老人状态机



## 增加

将用户输入信息保存到高龄老人表中。



## 90-99变百岁

更改老人类型，保存变动情况。



## 停发

更改老人停发状态；保存变动情况，根据停发的高龄老人类型确定变动类型。



## 探访记录

* 记录探访记录：insert into lg\_old\_person\_visit
* 如果停发，添加变动情况：insert into lg\_old\_person\_change
* 如果登记返款，更新探访对象的返款信息：update lg\_old\_person



## 返款登记

可以针对停发的高龄老人进行返款登记：

* 更新高龄老人：update lg\_old\_person，返款状态为已登记并提交



## 高龄老人返款财务处理

财务对老龄科高龄老人津贴返还登记进行处理：

* 如果确认收到款项，则标记已返款：update lg\_old\_person，返款状态为已返款；
* 如果登记有误，则退回到老龄科重新处理：update lg\_old\_person，返款状态为审核未通过：



# 尾巴