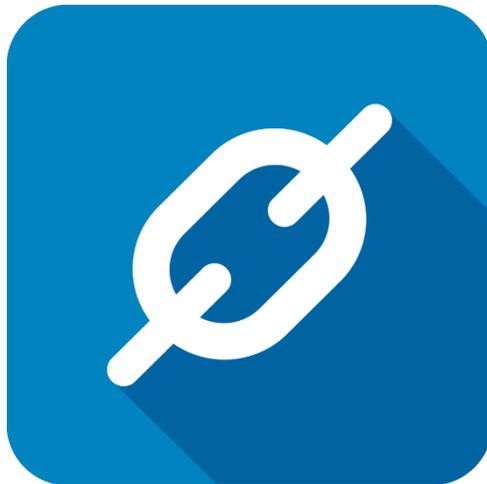


TotalLINK

产品手册



上海朝识智能科技有限公司

2020年8月

LINK 功能列及特殊标识

目 录

LINK 功能列及特殊标识	2
1 功能列概述	5
1.1 功能列 LINKBOOLEAN	6
1.2 功能列 LINKBOOLEAN 的编辑	7
1.3 功能列 LINKTRISTATE	8
1.4 功能列 LINKURL	9
1.5 功能列 LINKFILE	10
1.6 功能列 LINKSUB	12
1.7 功能列 LINKXSUB	15
1.8 功能列 LINKFUN	17
1.9 功能列 LINKBUTTON	17
1.10 功能列 LINKCOLOR	18
1.11 功能列 LINKFONT	18
1.12 功能列 LINKIMAGE	19
1.13 功能列 LINKTRACKBAR	20
1.14 功能列 LINKPROGRESS	20
1.15 功能列 LINKTEXT	21
1.16 功能列 LINKDATE	22
1.17 功能列 LINKTIME	22
1.18 功能列 LINKDATTIM	23
1.19 功能列 LINKCALC	24
1.20 功能列 LINKSPIN	24
1.21 功能列 LINKROWID	25
1.22 功能列 LINKCOLBC	26
1.23 功能列 LINKCOLBGC	27
1.24 功能列 LINKCOLFC	28
1.25 功能列 LINKFTP	28
1.26 功能列 LINKNUM	30
1.27 列标题特殊处理 LINKTI01-LINKTI05	31
1.28 功能列 LINKFORMULA	33
1.29 功能列 LINKSCRIPT	34

2	特殊标识.....	36
2.1	利用添加行添加记录--LINKADDNEWROWS.....	36
2.2	模型中的数据挖掘列--LINKSUBxx_XXX.....	37
2.3	全局变量.....	39
2.4	单元格数据校验--LINKCELLVALID.....	41
2.5	保存时逐行检查--LINKROWVALID.....	43
2.6	挖掘模型执行结束自动关闭并返回刷新列表--LINKAUTOCLOSE.....	45
2.7	查询模型直接加载数据--LINKLOADDATAIMMEDIATE.....	46
2.8	直接进入编辑模式--LINKEDITMODE.....	47
2.9	批量编辑数据--LINKBATCHEDIT.....	48
2.10	保留代码原始模式--LINKKEEPORIGIONAL.....	49
2.11	按钮类附加模型+2000 通知模型.....	53
2.12	Timeout 设置--LINKTIMEOUT:秒数.....	56
2.13	将#内容#替换为空--LINKAUTOREPLACEEMPTYCOLUMNS.....	56
2.14	用户数据权限--LINKMYROWDATA.....	59
2.14.1	控制用户看到的数据.....	60
2.14.2	根据条件控制按钮的可见性 (LINKFUNCTIONLIST).....	61
2.15	LINKFUNC 公式使用介绍.....	65
2.15.1	应用——LINKSCRIPTGET.....	67
2.15.2	应用——LINKSETVALUE/LINKGETVALUE.....	68
2.15.3	过程变量值的保存和获取——LINKSETVAL/LINKGETVAL.....	73
2.16	下拉选择框控制.....	74
2.16.1	LINKLISTXXX.....	74
2.16.2	LINKDROPDOWNXXX.....	77
2.16.3	LINKDROPVALIDXXX.....	80
2.17	附加模型处理多行数据--LINKMULTIROWS.....	81
2.17.1	应用一 发送邮件给多个用户.....	86
2.17.2	应用二 自动处理不同区域客户数据.....	88
2.18	网页集成说明.....	89
2.18.1	网页集成方式.....	91
2.18.2	用户认证.....	94
2.18.3	链接到具体单据.....	95
2.19	从指定的链接获取数据--LINKCONN@XXX.....	96
2.20	查询模型直接加载数据--LINKINPUTBOX.....	101
2.21	标记某些模型不出现在 APP 菜单中.....	102
2.22	获取纯文件名.....	103
3	功能调用.....	105
3.1	LINKACTIONLIST 功能.....	105

文档控制

■ 主要内容

本文整合 LINK 功能列及 LINK 特殊标识，方便开发者统一学习。

■ 更改记录

日期	版本	作者	备注
2017-05	1.0	Randy	初始发布
2020-08-24	1.1	Liz	用户模型权限管理升级-按钮隐藏
2020-09-04	2.0	Jozey	整合所有系统中LINK的关键字
2020-10-14	3.0	Liz	添加“指定链接获取数据 LINKCONN@XXX”内容
2021-02-25	2.3	Werner	标记某些模型不出现在 APP 菜单中
2021-02-25	2.3	Werner	获取纯文件名
2021-03-01	2.3	Werner	过程变量值的保存和获取

■ 支持版本

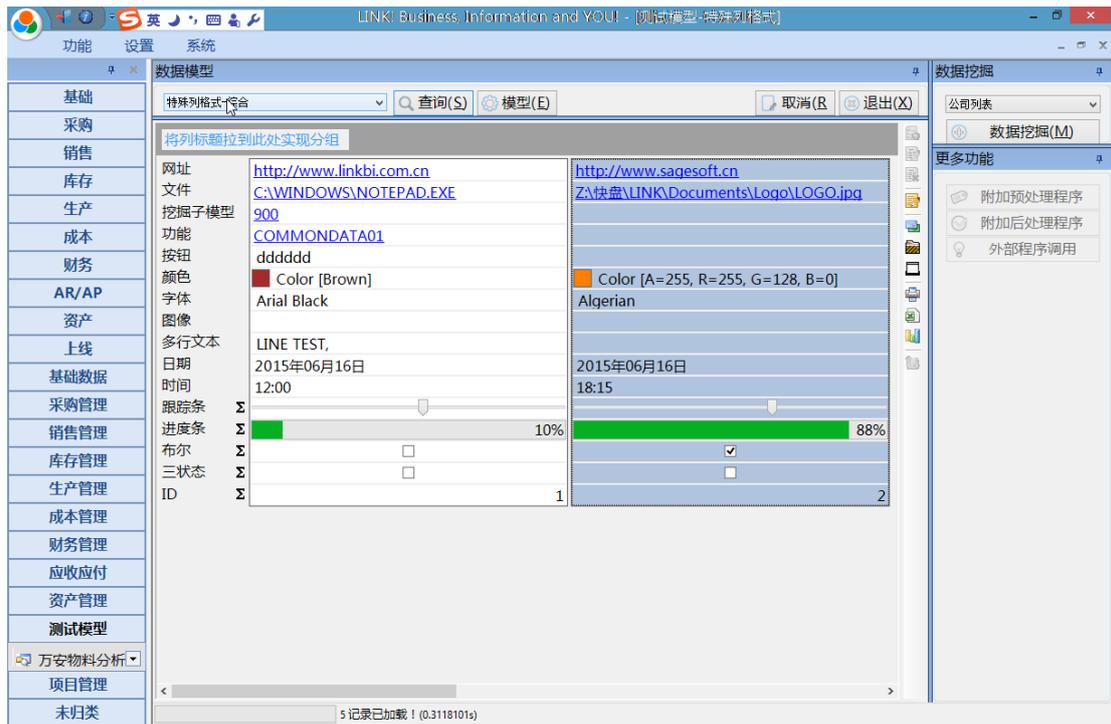
非特殊说明的功能，默认前后版本都支持

序号	T20更新功能点	T20版本号
1	将#内容#替换为空—LINKAUTOREPLACEEMPTYCOLUMNS	
2	Timeout设置--LINKTIMEOUT:秒数	
3	从指定链接获取数据LINKCONN@XXX	
4	标记某些模型不出现在 APP 菜单中	
5	增加参数快捷输入功能	V 8.2.1.0
6	获取纯文件名	V 8.2.3.4
7	LINKSETVAL与LINKGETVAL的使用	V 8.2.3.4
8	过程变量值的保存和获取	V 8.2.3.4
9	日期格式处理—LINKDATE	V 8.2.3.6
10	LinkActionList 执行问题	V 8.2.3.9
11	使用动态LINKCONN	V 8.2.3.15

1 功能列概述

利用 TotalLINK 系统，定义模型时可以设置一些特殊功能列。

只要列标题符合 TotalLINK 系统的特殊功能列规范，系统就能够识别这些特殊的列，并非常容易的实现不同的功能。



系统中定义的特殊功能列举如下：

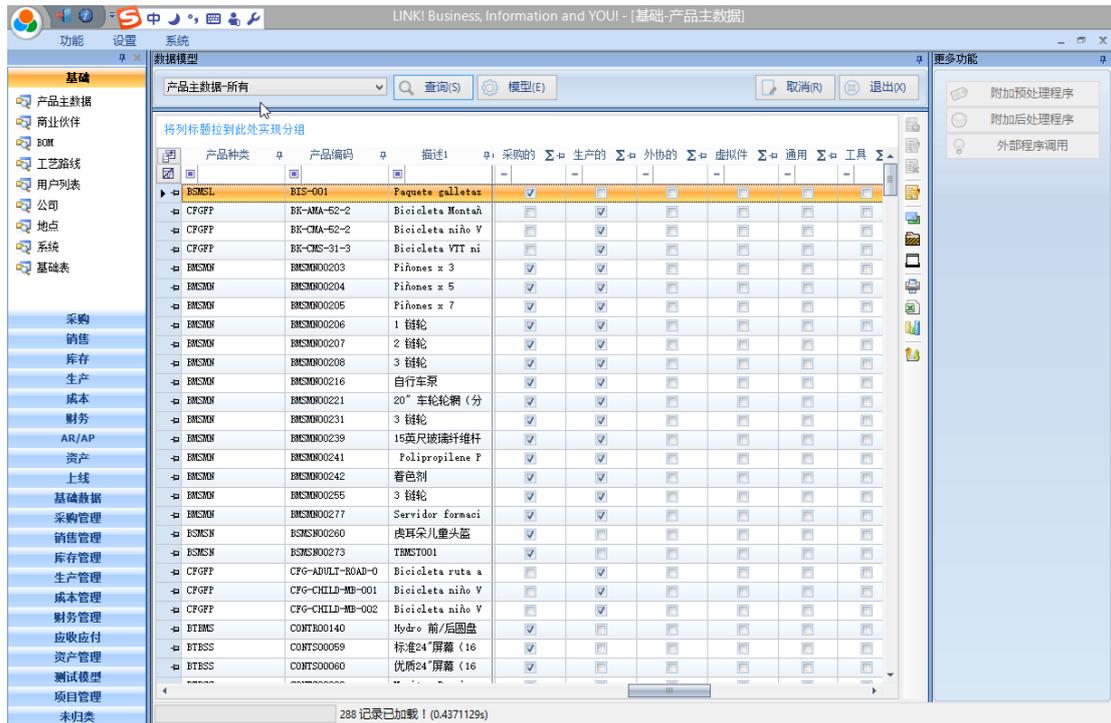
- LINKBOOLEAN 用于处理布尔型（是、否）信息
- LINKTRISTATE 用于处理三种数据状态（是、否、未指定）信息
- LINKURL 用于处理超级链接信息
- LINKFILE 用于处理文件链接信息
- LINKSUB 用于处理数据挖掘到子模型
- LINKXSUB 用于处理多个数据挖掘模型的跳转
- LINKFUN 用于处理功能调用
- LINKBUTTON 用于显示一个按钮，对应功能调用
- LINKCOLOR 用于处理颜色选择

- LINKFONT 用于处理字体选择
 - LINKIMAGE 用于处理图片显示
 - LINKTRACKBAR 用于处理跟踪进度条
 - LINKPROGRESS 用于处理进度条
 - LINKTEXT 用于处理多行文本
 - LINKDATE 用于处理日期
 - LINKTIME 用于处理时间
 - LINKDATTIM 用于显示日期和时间
 - LINKCALC 用于显示一个计算器进行计算及录入数据
 - LINKSPIN 用于显示数据输入的上下箭头
 - LINKROWID 用于设置默认不显示的列
 - LINKCOLBC 单元格背景色
 - LINKCOLBGC 单元格背景渐变色
 - LINKCOLFC 单元格前景(字体)颜色
 - LINKFTP 用来处理 ftp 文件上传
 - LINKNUM 用来处理自动增加编号的数据列
 - LINKTI01-LINKTI05 用来处理特殊列的标题
 - LINKFORMULA 用来处理公式内容的计算
 - LINKSCRIPT 用来对单元格内的脚本进行处理
- LINKSCRIPTRUN@ 可以用于执行脚本
 - LINKSCRIPTGET@ 可以用于根据脚本获取数据

1.1 功能列 LINKBOOLEAN

- 功能列 LINKBOOLEAN，用于处理布尔型（是、否）信息
 - 在数据分析时，如果某列的信息我们希望系统以“是/否”选择框的形式显示，可以使用 LINKBOOLEAN 类型的特殊功能列
 - 比如：列标题为“LINKBOOLEAN 库存管理”
- 系统显示的列标题为“库存管理”
 - 列的格式为“选择框”
 - 如果该列是可以编辑的，当用户点击改变状态时，状态在“是/否”两种状态中循

环改变

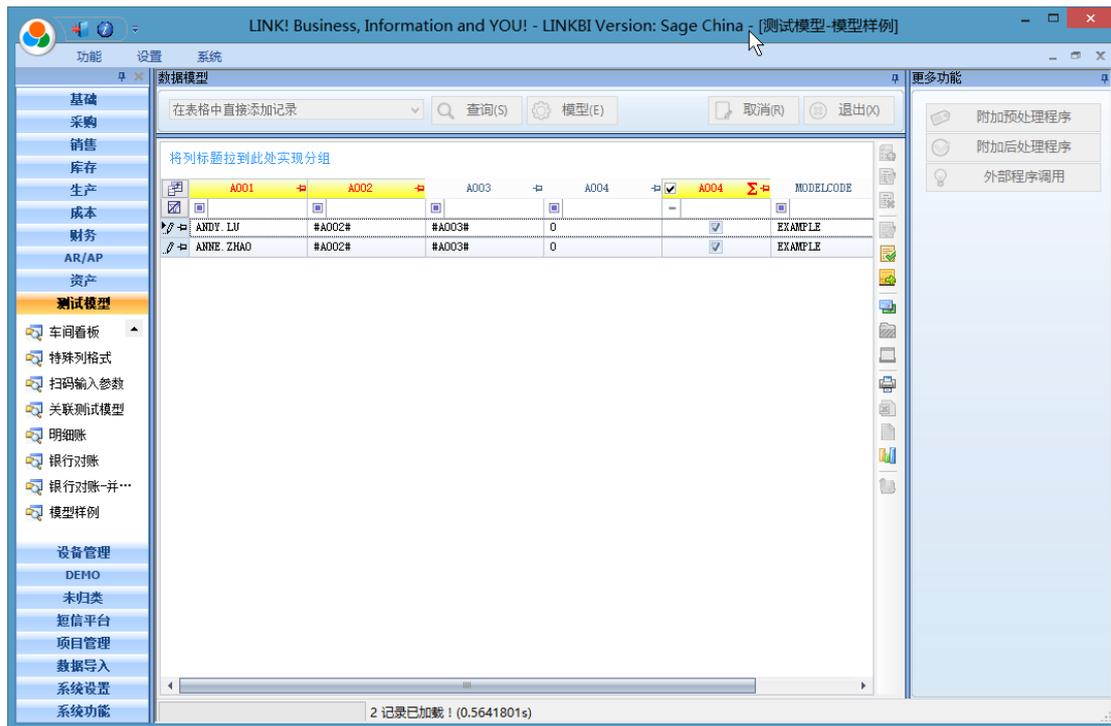


● 备注

- 对于 SQL Server 数据库中的数据，如果列的类型为 BIT，系统会自动识别为“选择框”

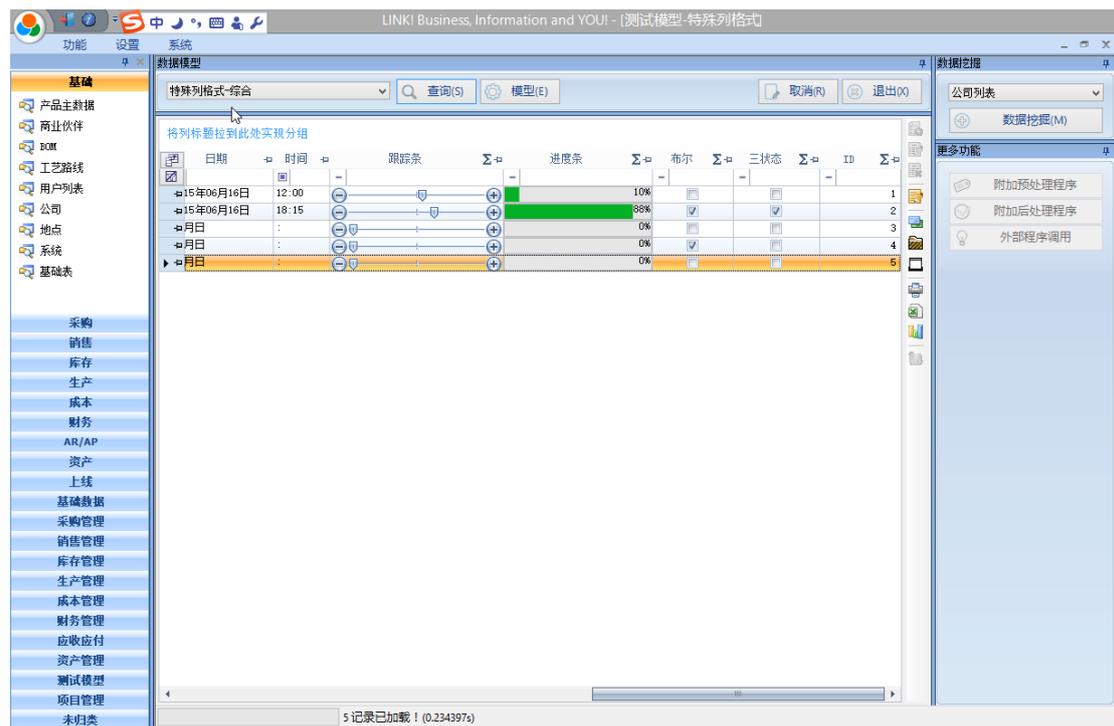
1.2 功能列 LINKBOOLEAN 的编辑

- 使用 LINKBOOLEAN 类型的列的时候，如果该列是可以编辑的，在表格的对应标题中会显示一个 CheckBox 控件
- 对该利用该控件可以对所见的行的状态进行批量修改
- 如下图所示



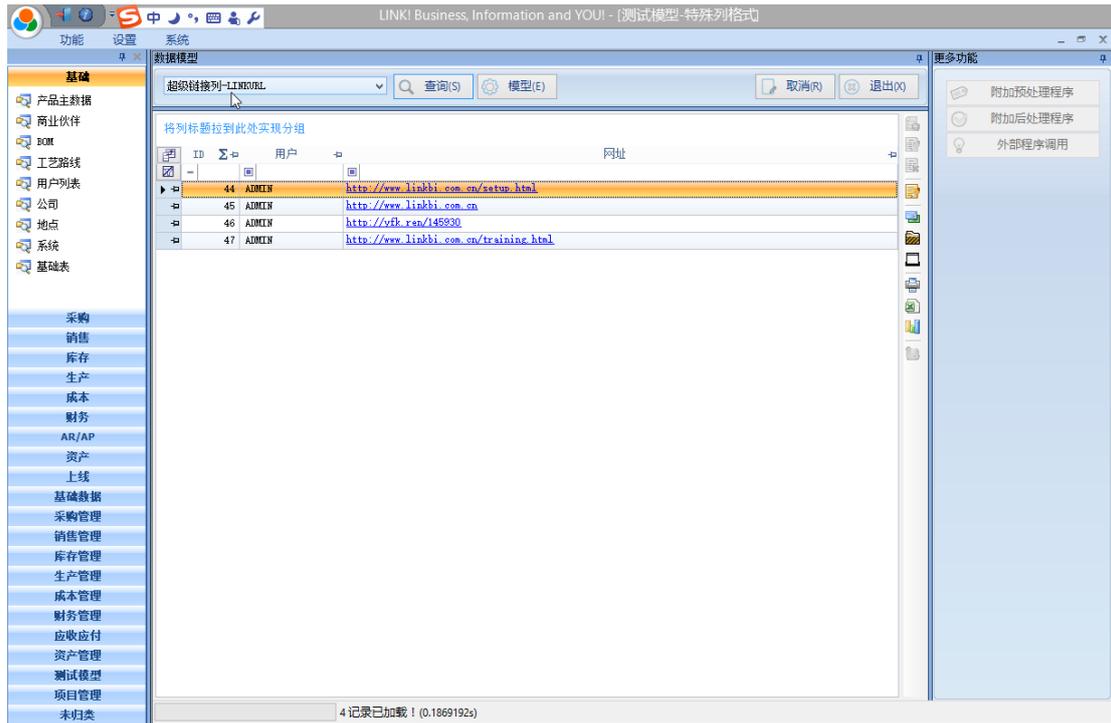
1.3 功能列 LINKTRISTATE

- 用于处理三种数据状态（是、否、未指定）信息
- 在数据分析时，如果某列的信息我们希望系统以“是/否/未指定”选择框的形式显示，可以使用 LINKTRISTATE 类型的特殊功能列
- 比如：列标题为“LINKTRISTATE 库存管理”
 - 系统显示的列标题为“库存管理”
 - 列的格式为“选择框”
 - 如果该列是可以编辑的，当用户点击改变状态时，状态在“是/否/未指定”三种状态中循环改变



1.4 功能列 LINKURL

- 用于处理超级链接信息
- 如果某列的内容为超级链接，可以使用类似“LINKURL 网址”的列标题
 - 系统会自动将其识别为超级链接
 - 点击超级链接可以访问对应的网址



上图中例子的脚本如下：

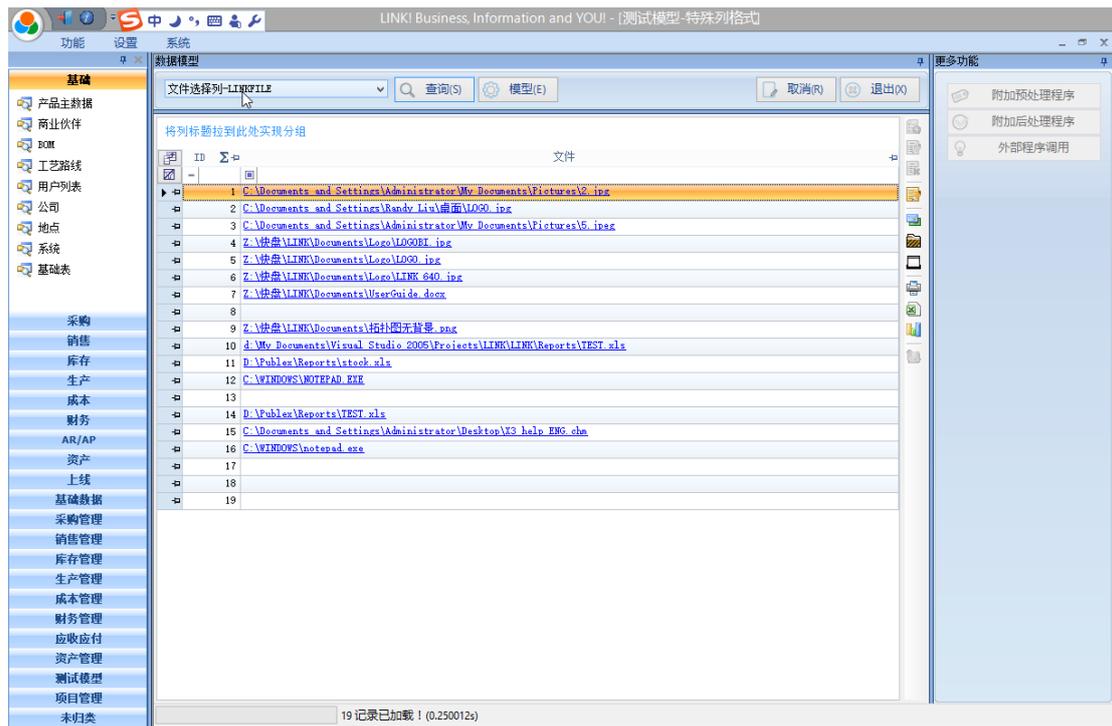
```
SELECT ROWID ID,
       A_0 用户,
       A_1 LINKURL 网址
FROM   YTEMP
WHERE  A_0 = 'ADMIN'
```

- 超级链接列的编辑脚本可以这样写：

```
UPDATE YTEMP
SET    A_1 = N'@LINKURL 网址@'
WHERE ROWID = '#ID#'
```

1.5 功能列 LINKFILE

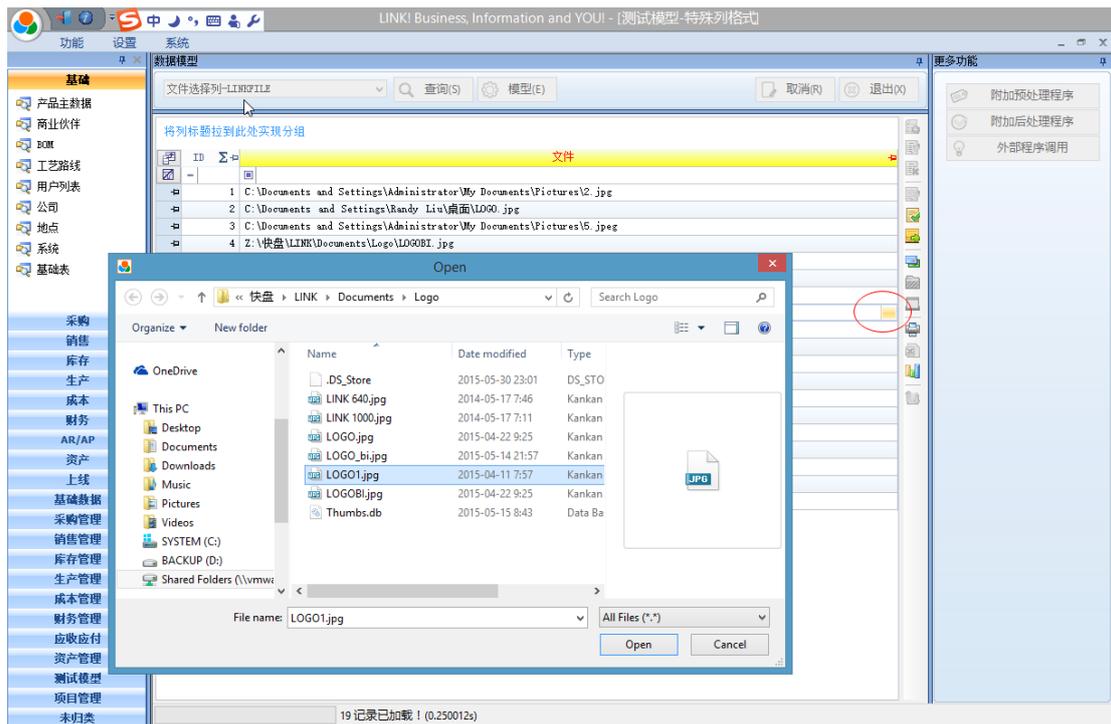
- 用于处理文件链接信息
- 如果某列的内容为文件链接，可以使用类似“LINKFILE 文件”的列标题
 - 系统会自动将其识别为文件链接
 - 点击超级链接可以访问对应的文件



- 上例的模型脚本如下：

```
SELECT ROWID ID,
       A_0 单据,
       A_1 LINKFILE 文件
FROM   YTEMP
WHERE  ROWID < 20
```

- 如果该列是可以编辑的，系统会自动启动文件选择窗口提示用户选择文件

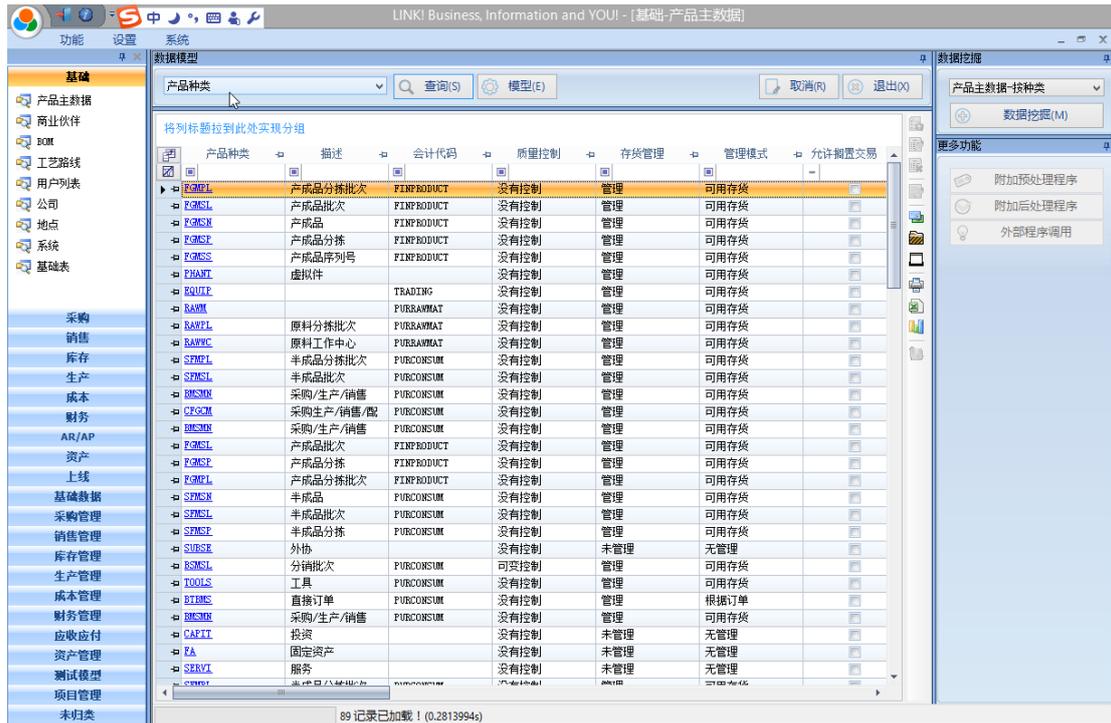


- 该模型的编辑脚本如下：

```
UPDATE YTEMP
SET A_1 = N'@FILE_文件@'
WHERE ROWID = '#ID#'
```

1.6 功能列 LINKSUB

- 用于处理数据挖掘到子模型
- 对于定义了数据挖掘功能的模型，可以将列设置为 LINKSUB 特殊功能列
- 比如，下列模型中包含列 LINKSUB10_产品种类
 - 该列自动显示为超级链接
 - 点击该列可以调用对应的数据挖掘功能



如上图所示，基本功能模型为：

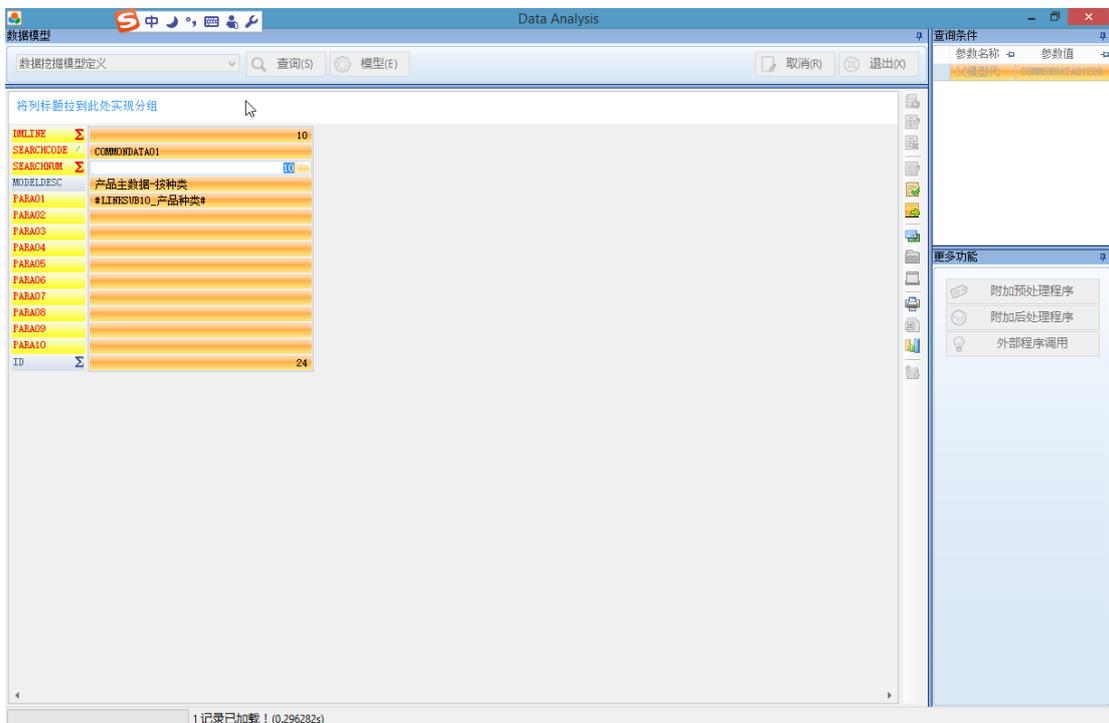
```

SELECT TCLCOD_0 AS LINKSUB10_产品种类,
      T.T AS 描述,
      ACCCOD_0 AS 会计代码,
      L275.LANMES_0 AS 质量控制,
      L215.LANMES_0 AS 存货管理,
      L297.LANMES_0 AS 管理模式,
      I.NEGSTO_0 - 1 AS LINKBOOLEAN 允许搁置交易,
      L2711.LANMES_0 AS 批次管理,
      LOTCOU_0 AS 批次数序列号,
      L210.LANMES_0 AS 序列号管理,
      SERCOU_0 AS 序列号计数,
      STU_0 AS 存货单位,
      PUU_0 AS 采购单位,
      SAU_0 AS 销售单位,
      SSU_0 AS 统计单位,
      VLTCOD_0 AS 计价方式,
      LOCMGTCOD_0 - 1 AS LINKBOOLEAN 库位管理,
      L217.LANMES_0 AS 存货追踪模式,
      L727.LANMES_0 AS 补货模式,
      L250.LANMES_0 AS 建议类型,
      REOPOL_0 AS 补货策略
FROM ITMCATEG I
LEFT OUTER JOIN APLSTD L275
  ON L275.LANCHP_0 = '275'
  AND L275.LAN_0 = 'CHI'
  AND L275.LANNUM_0 = I.QUAFLG_0
LEFT OUTER JOIN APLSTD L215
  ON L215.LANCHP_0 = '215'
  AND L215.LAN_0 = 'CHI'
  AND L215.LANNUM_0 = I.STOMGTCOD_0
LEFT OUTER JOIN APLSTD L297
  ON L297.LANCHP_0 = '297'
  AND L297.LAN_0 = 'CHI'
    
```

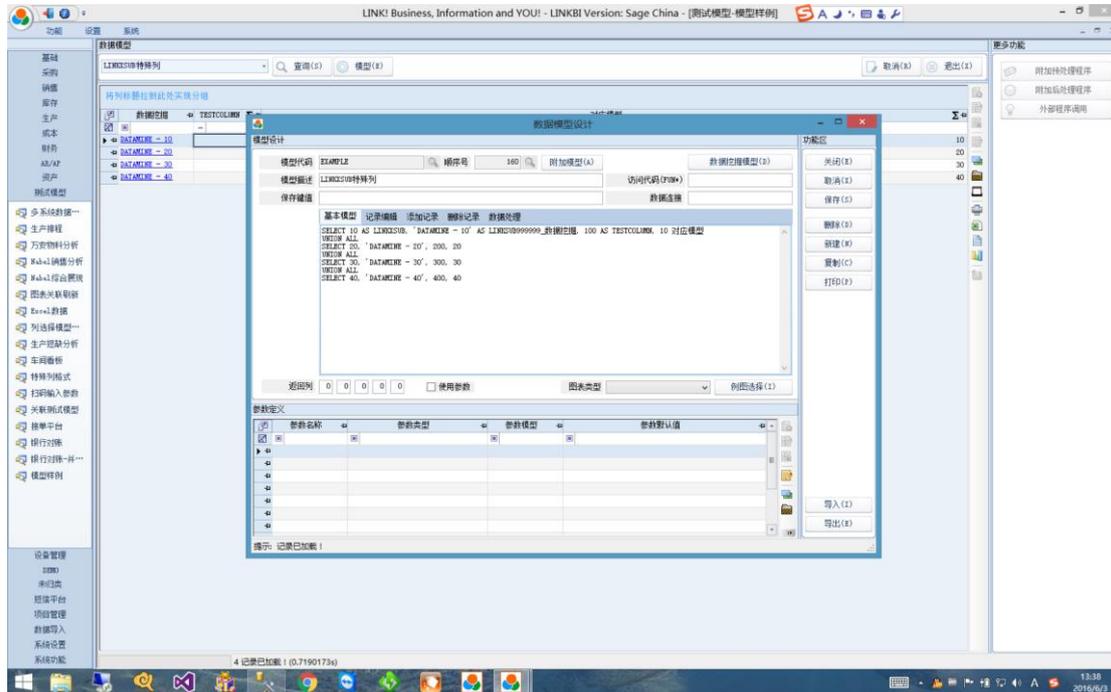
```

AND L297.LANNUM_0 = I.STDFLG_0
LEFT OUTER JOIN APLSTD L2711
    ON L2711.LANCHP_0 = '2711'
    AND L2711.LAN_0 = 'CHI'
    AND L2711.LANNUM_0 = I.LOTMGTCOD_0
LEFT OUTER JOIN APLSTD L210
    ON L210.LANCHP_0 = '210'
    AND L210.LAN_0 = 'CHI'
    AND L210.LANNUM_0 = I.SERMGTCOD_0
LEFT OUTER JOIN APLSTD L217
    ON L217.LANCHP_0 = '217'
    AND L217.LAN_0 = 'CHI'
    AND L217.LANNUM_0 = I.STOCOD_0
LEFT OUTER JOIN APLSTD L727
    ON L727.LANCHP_0 = '727'
    AND L727.LAN_0 = 'CHI'
    AND L727.LANNUM_0 = I.REOMGTCOD_0
LEFT OUTER JOIN APLSTD L250
    ON L250.LANCHP_0 = '250'
    AND L250.LAN_0 = 'CHI'
    AND L250.LANNUM_0 = I.REOCOD_0
LEFT OUTER JOIN (SELECT IDENT1_0 AS C,
    TEXTE_0 AS T
    FROM ATEXTRA
    WHERE CODFIC_0 = 'ITMCATEG'
    AND ZONE_0 = 'TCLAXX'
    AND LANGUE_0 = 'CHI') T
    ON T.C = I.TCLCOD_0
    
```

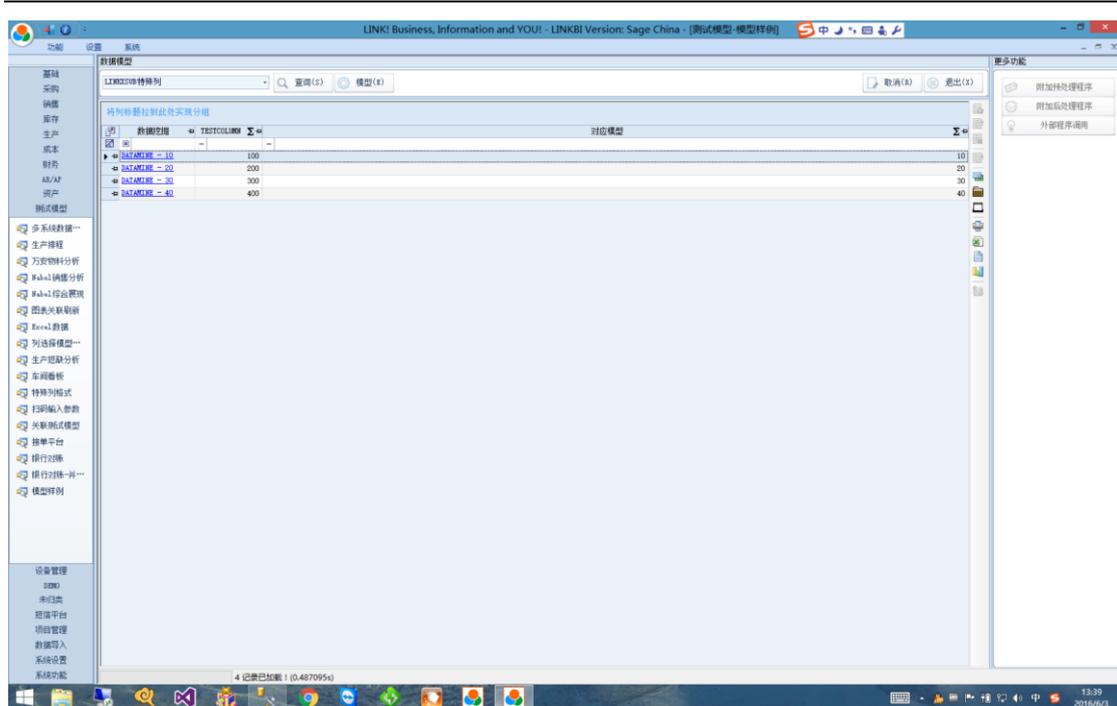
● 该模型对应一个数据挖掘模型，设置如下图所示：



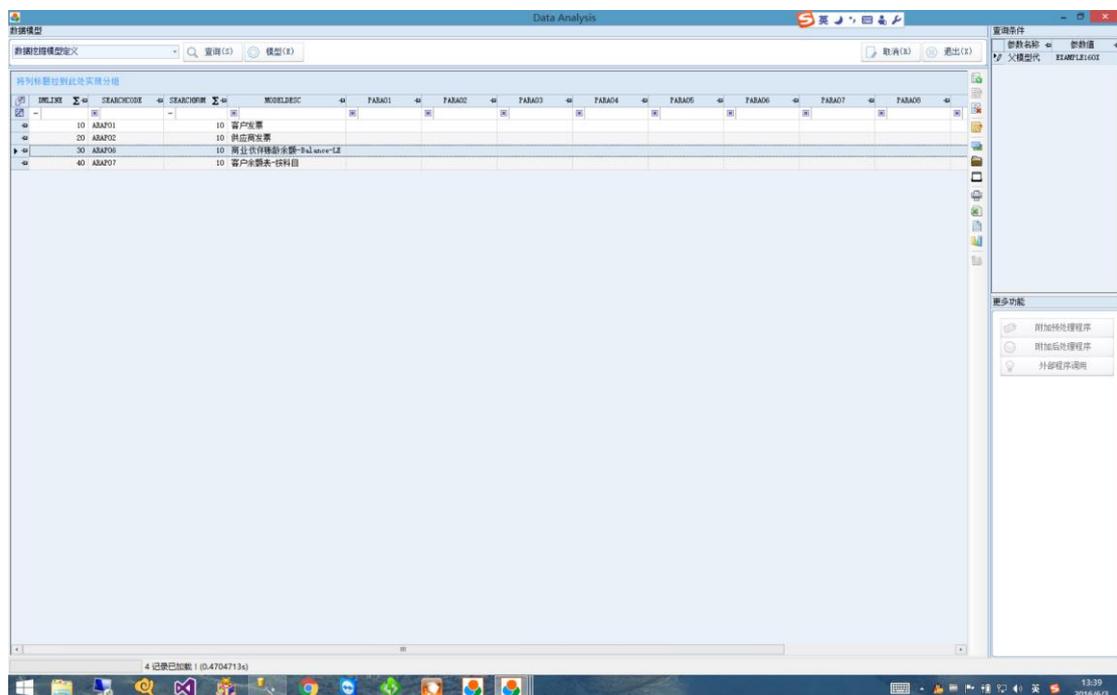
1.7 功能列 LINKXSUB



- 某些情况下，行数据的不同列需要对应不同的数据挖掘模型
- 这时，可以用不同的辅助列，采用 LINKSUB10_XXX，LINKSUB20_XXX 等列分别对应不同的数据挖掘模型
- 这里提供另外一种方式，使用 LINKXSUB 功能列和 LINKSUB99999_XXX 配合的方式实现一个列可以对应不同的数据挖掘模型
- 其中：
 - LINKXSUB 列的内容是需要挖掘的数据模型的编号
 - 当数据挖掘编号为 999999 的时候，尝试取 LINKXSUB 列对应的值作为数据挖掘模型的行号
- 点击“数据挖掘”列的时候，自动获取当前行对应的数据挖掘模型编号，并调用数据挖掘功能



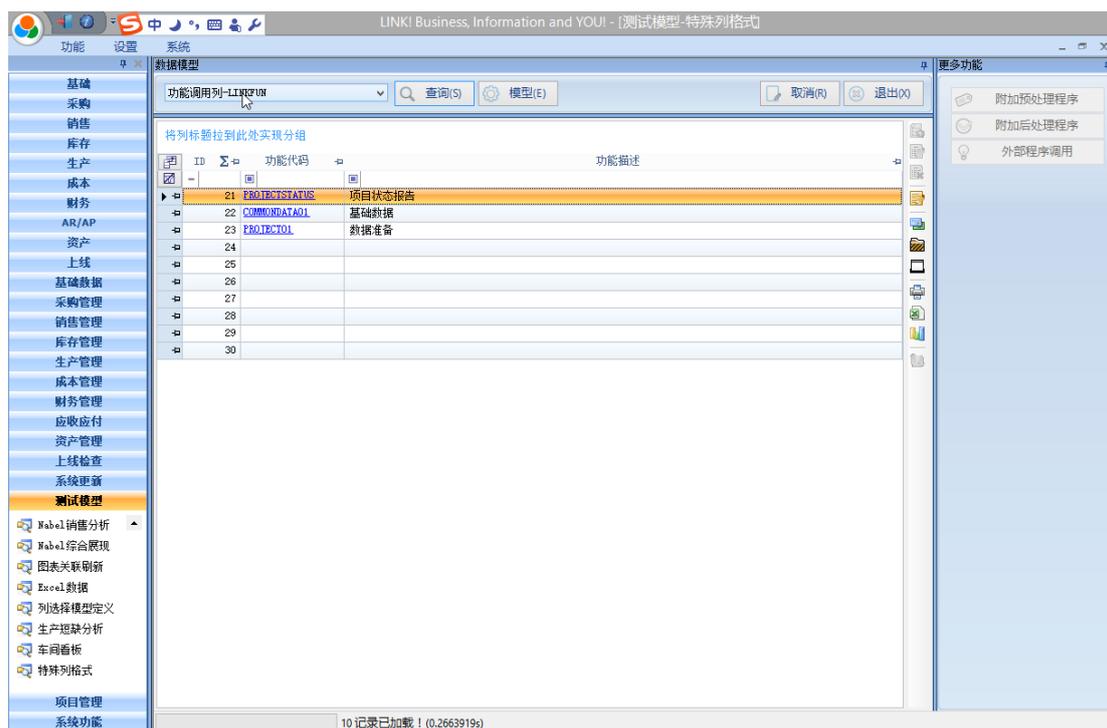
- 数据模型查询的结果如图所示
- 点击同一个列“数据挖掘”时，可以根据数据挖掘模型编号分别打开不同的数据挖掘内容



- 数据挖掘模型的配置方式如上图所示，与常规的数据挖掘模型配置一致
- 如上图所示

1.8 功能列 LINKFUN

- 用于处理功能调用
- 可以用该列处理其他功能模型的调用
- 比如，下列模型中包含列“LINKFUN 功能代码”
- 该列自动显示为超级链接
- 点击该列可以调用对应的数据分析功能



- 上例的模型脚本如下：

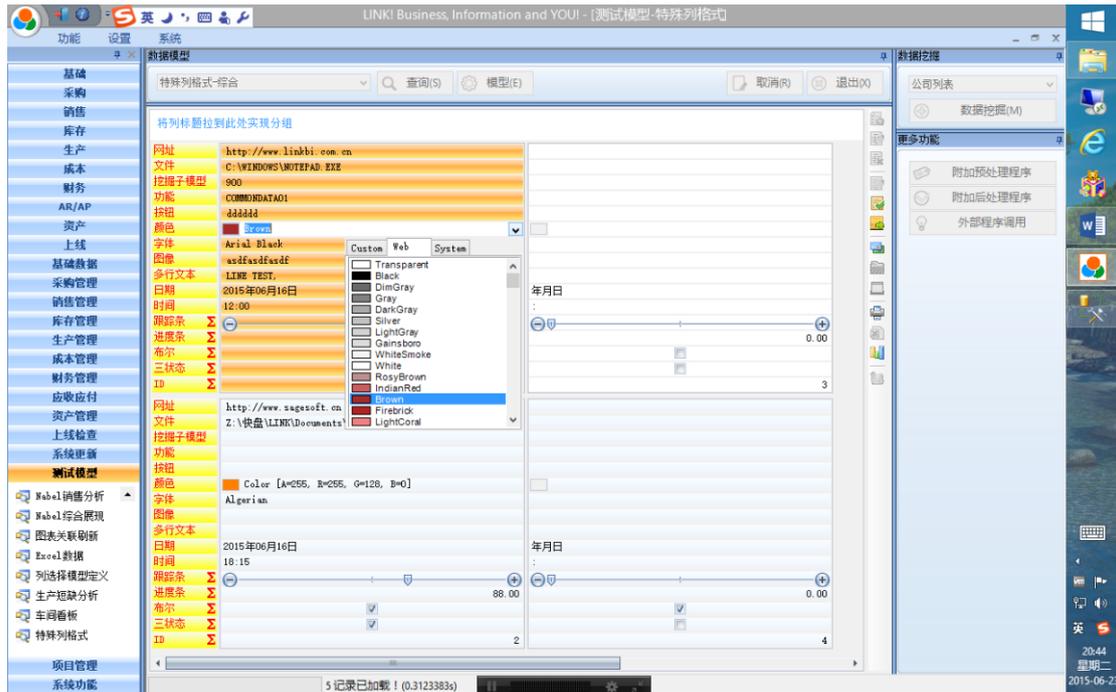
```
SELECT ROWID ID,
       A_0 单据,
       A_1 LINKFUN 功能代码,
       A_2 AS 功能描述
FROM   YTEMP
WHERE  ROWID BETWEEN 21 AND 30
```

1.9 功能列 LINKBUTTON

- 用于显示一个按钮，对应功能调用
- 此功能为后续功能扩展保留

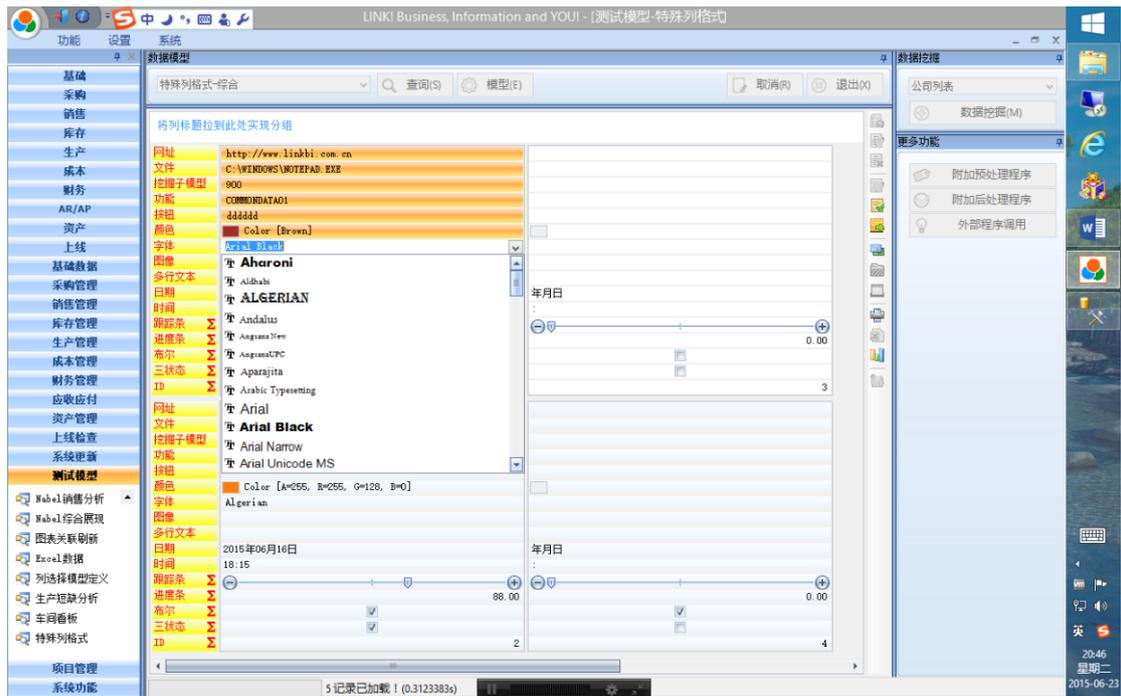
1.10 功能列 LINKCOLOR

- 用于处理颜色选择
- 如果所示，如果列类似“LINKCOLOR 颜色”格式，系统会自动识别为颜色列
- 如果该列是可以编辑的，系统自动打开颜色选择功能



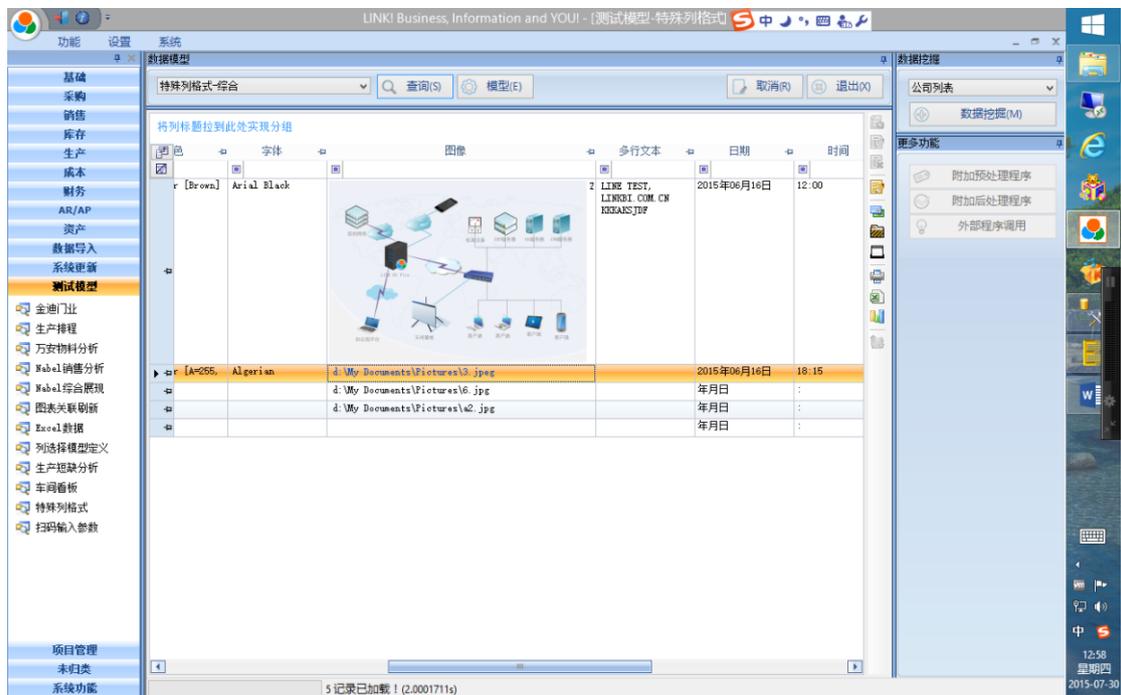
1.11 功能列 LINKFONT

- 用于处理字体选择
- 如图所示，如果列类似“LINKFONT 字体”格式，系统会自动识别为字体列
- 如果该列是可以编辑的，系统自动打开字体选择功能



1.12 功能列 LINKIMAGE

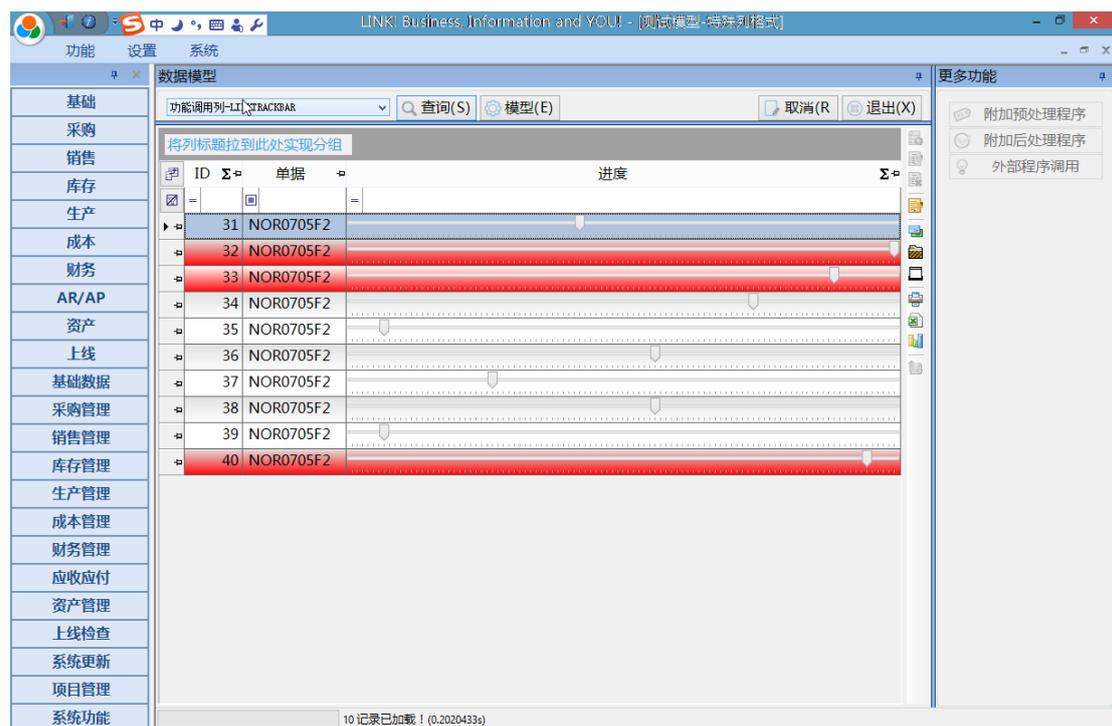
- 用于处理图片信息显示



- 如图所示，如果列类似“LINKIMAGE 图片”格式，系统会自动识别为图像列

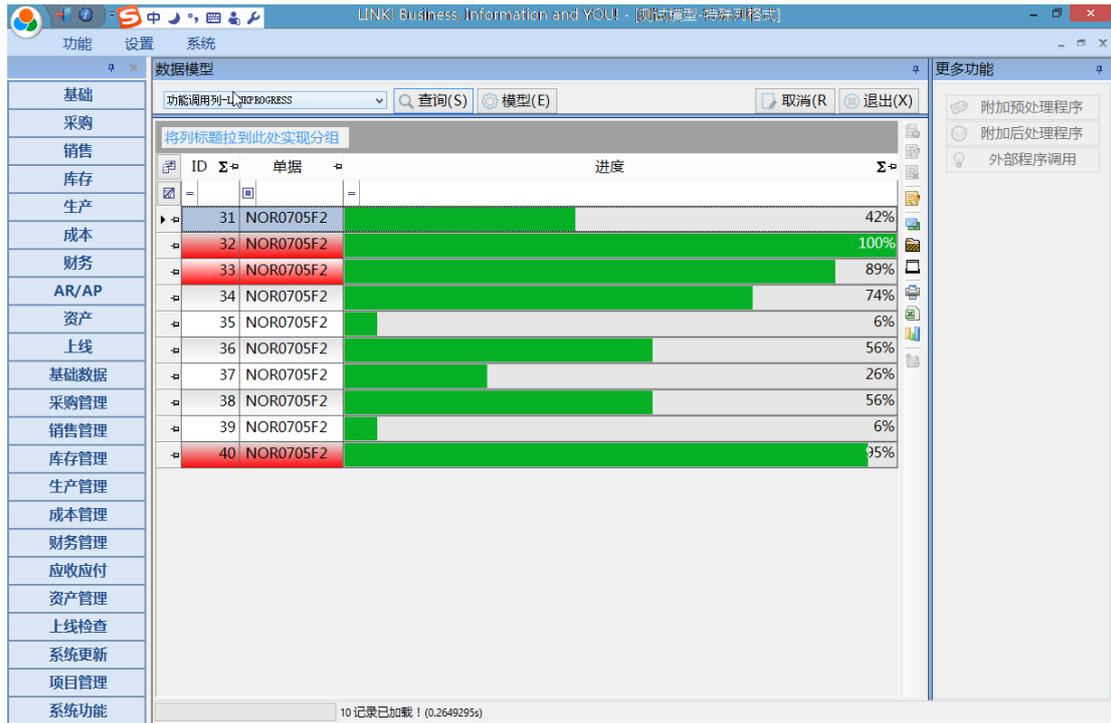
1.13 功能列 LINKTRACKBAR

- 用于处理跟踪进度条
- 如果所示，如果列类似“LINKTRACKBAR 跟踪条”格式，系统会自动识别为跟踪条
- 跟踪条的数据为 0-100



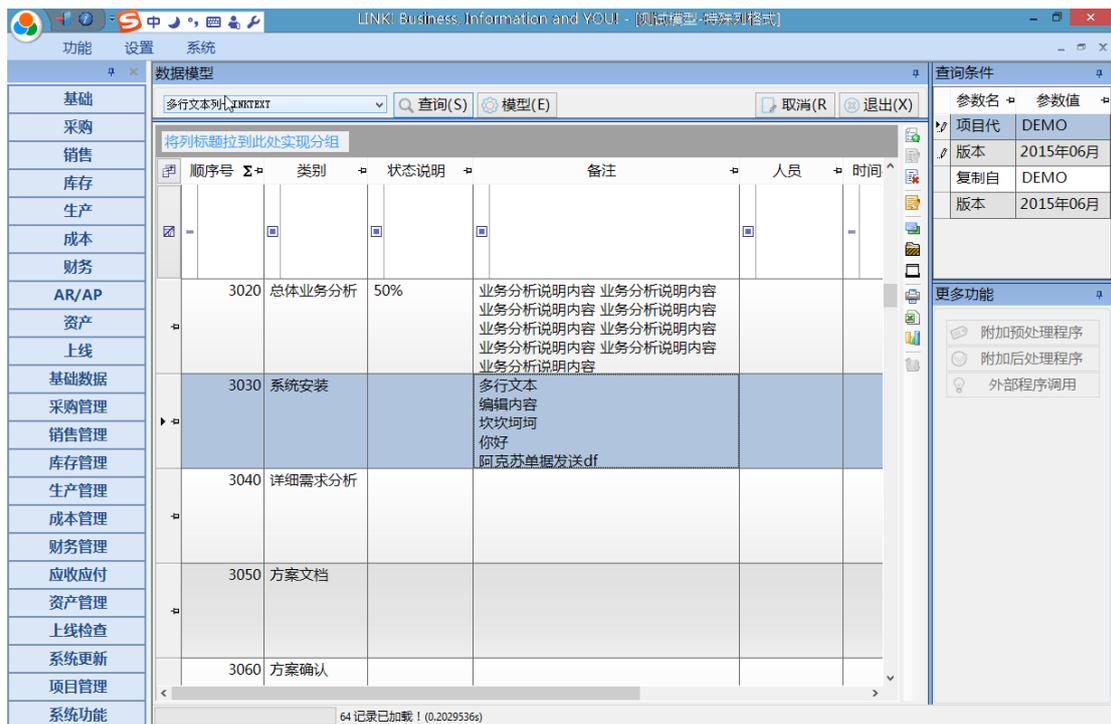
1.14 功能列 LINKPROGRESS

- 用于处理进度条
- 如果所示，如果列类似“LINKPROGRESS 进度条”格式，系统会自动识别为进度条
- 进度条的数据为 0-100



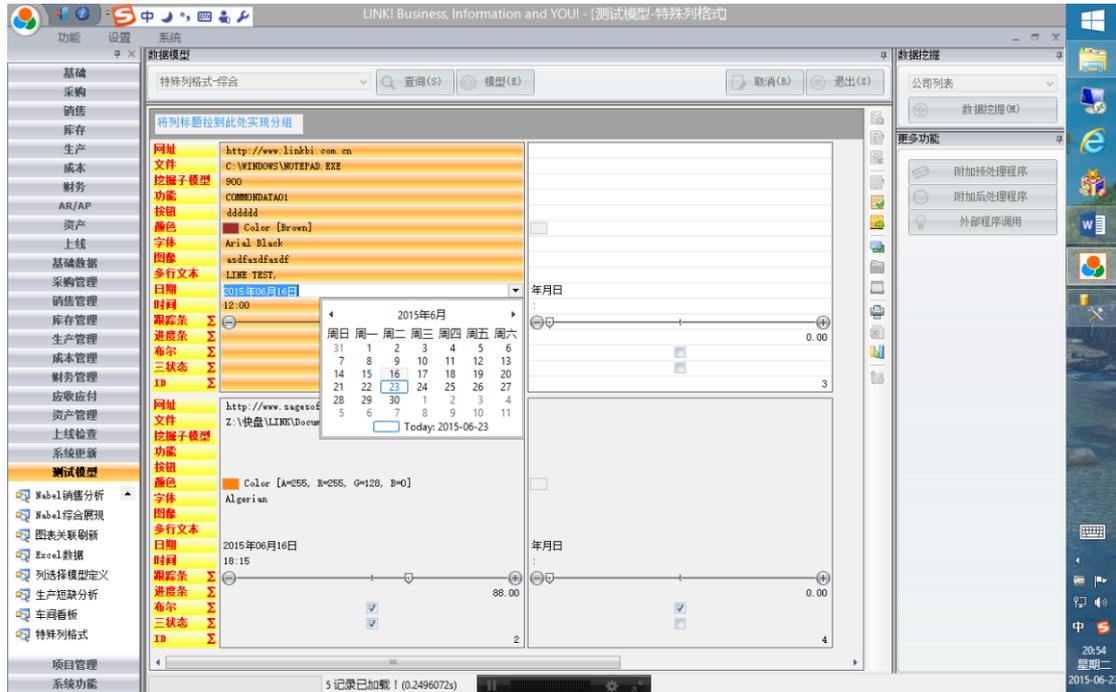
1.15 功能列 LINKTEXT

- 用于处理多行文本
- 当输入的内容为多行文本时，可以使用类似“LINKTEXT 备注”类型的列标题



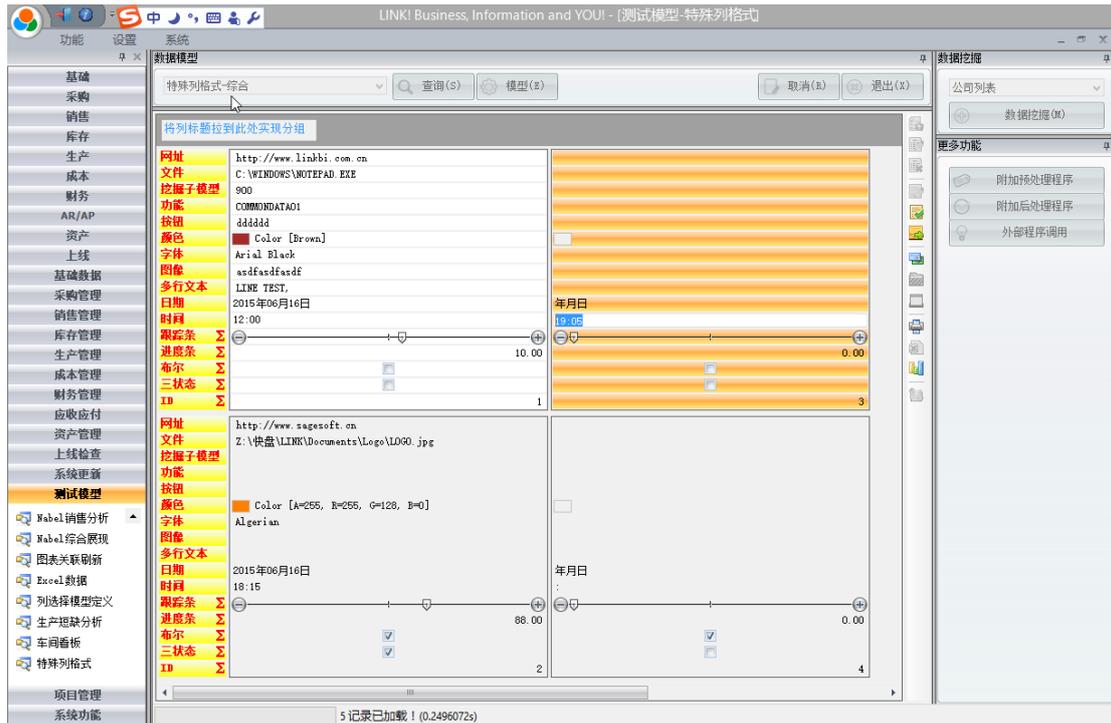
1.16 功能列 LINKDATE

- 用于处理日期
- 系统会自动将日期类型的内容识别为日期，也可以使用类似“LINKDATE 日期”格式的列标题，让系统强制识别为日期列



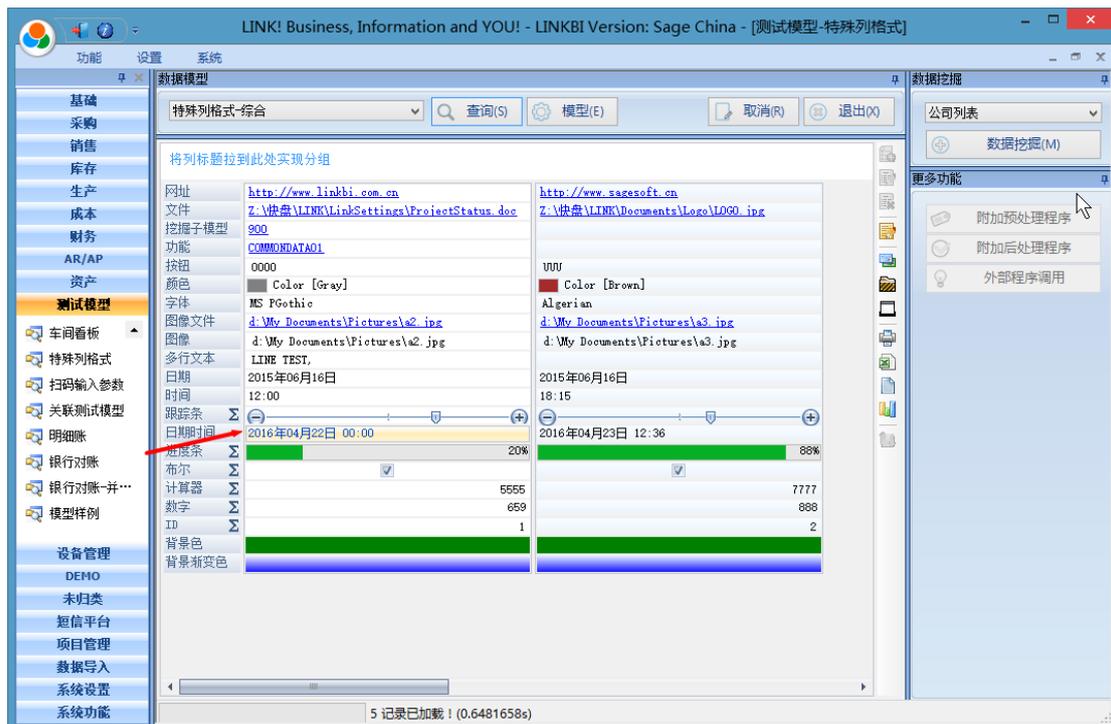
1.17 功能列 LINKTIME

- 用于处理时间
- 用类似“LINKTIME 时间”作为列标题时，系统自动将其识别为形如 HH:MM 的时间格式



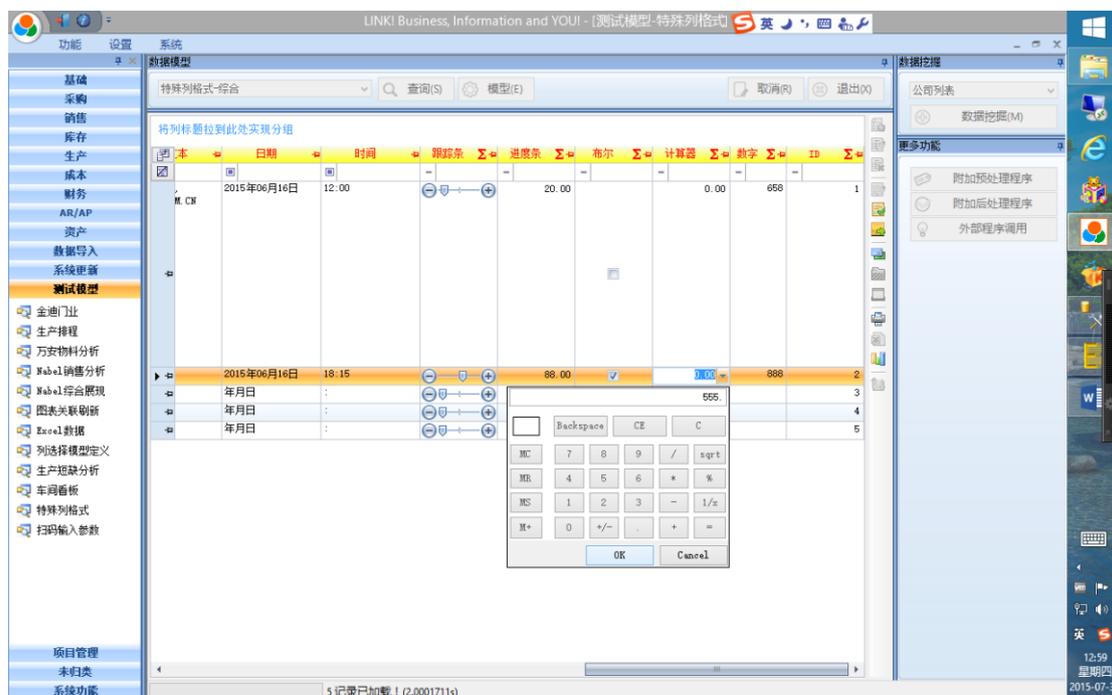
1.18 功能列 LINKDATTIM

- 用于显示和输入日期及时间
- 如图所示，当使用“LINKDATTIM 日期时间”作为列标题时，系统将自动将其识别为形如 YYYY-MM-DD HH:MM 的日期时间格式



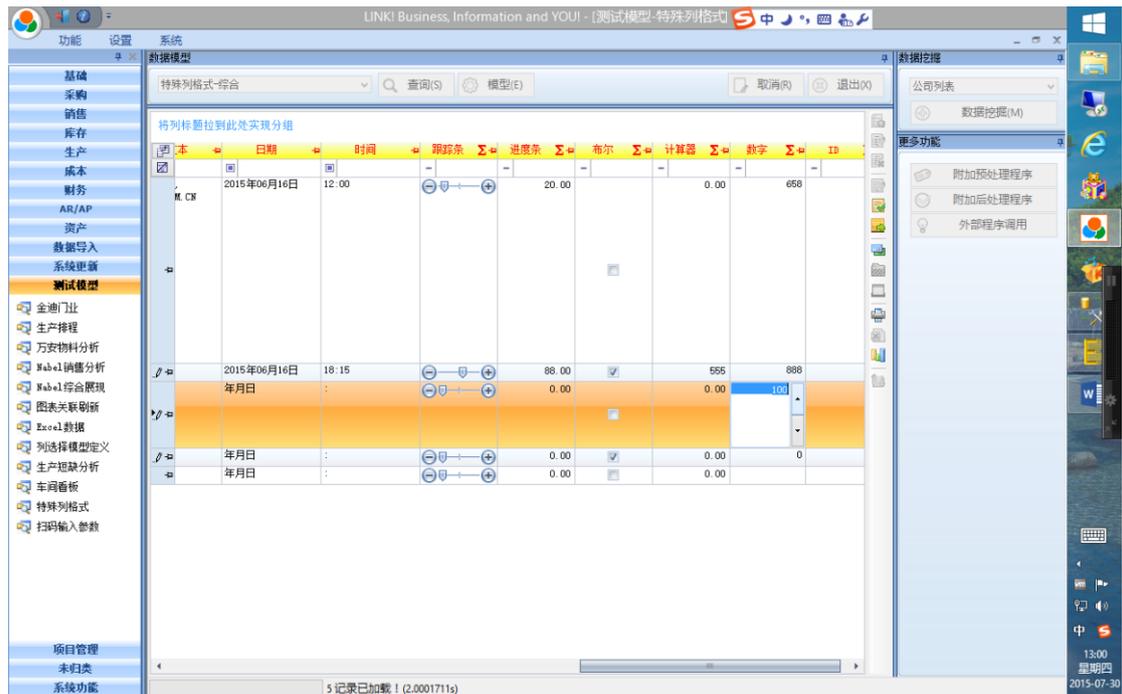
1.19 功能列 LINKCALC

- 用于处理数字输入
- 如图所示，用类似“LINKCALC 计算器”作为列标题时，系统自动将其设定为可以打开计算器输入的单元格



1.20 功能列 LINKSPIN

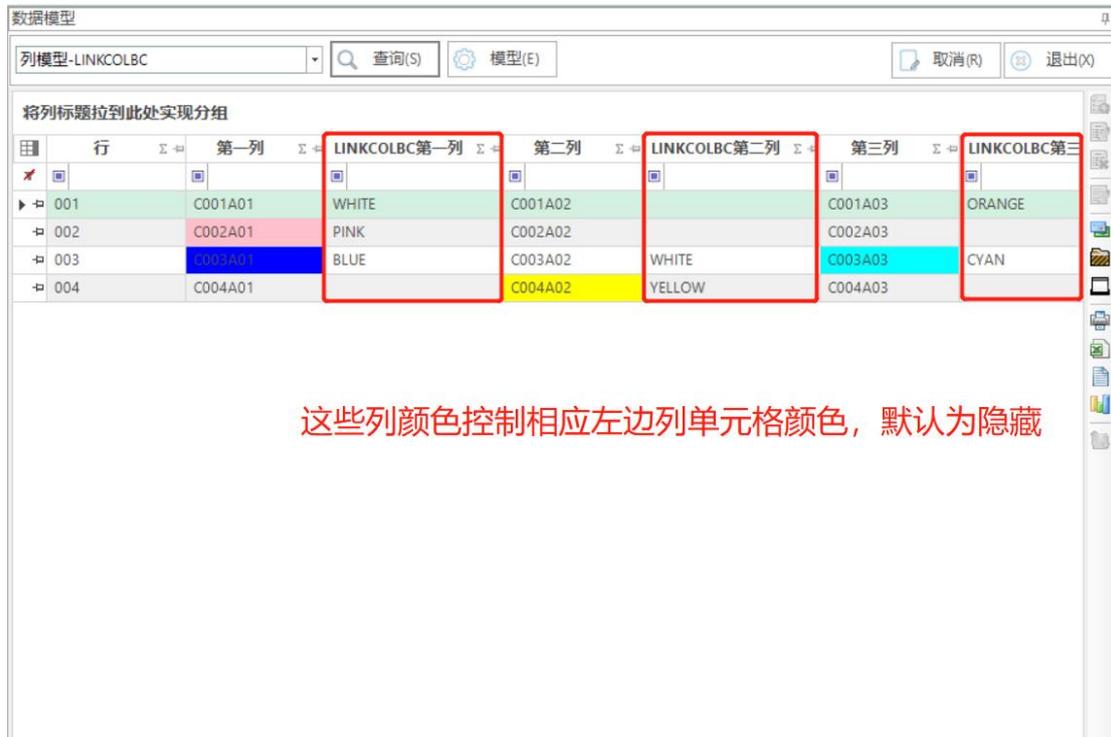
- 用于处理数字输入
- 如图所示，用类似“LINKSPIN 数字”作为列标题时，系统自动将其设定为可以通过上下箭头修改输入内容的单元格



1.21 功能列 LINKROWID

- 该列用于数据表中自动隐藏的 ROWID
- 当列标题以 LINKROWID 开始时，对应列默认不显示，比如
 - LINKROWID
 - LINKROWID1
 - LINKROWID002
 - LINKROWID 列标识

1.22 功能列 LINKCOLBC



- 如图所示，可以通过标识 LINKCOLBCxxx 来控制单元格的背景色显示
- 使用数据模型查询数据的时候，数据列的内容是“颜色名称”（参见颜色列表），若为空或 NULL 则保持原来的单元格颜色
- 在表格中，颜色名称以实际的颜色显示
- 上图所使用的脚本参考如下：

```

SELECT
'001' 行,
'C001A01' 第一列,
'WHITE' LINKCOLBC 第一列,
'C001A02' 第二列,
'' LINKCOLBC 第二列,
'C001A03' 第三列,
'ORANGE' LINKCOLBC 第三列,
'C001A04' 第四列,
'YELLOW' LINKCOLBC 第四列
UNION ALL
SELECT
'002' 行,
'C002A01' 第一列,
'PINK' LINKCOLBC 第一列,
'C002A02' 第二列,
NULL LINKCOLBC 第二列,
'C002A03' 第三列,
'' LINKCOLBC 第三列,

```

```
'C002A04' 第四列,
''        LINKCOLBC 第四列
UNION ALL
SELECT
'003'    行,
'C003A01' 第一列,
'BLUE'   LINKCOLBC 第一列,
'C003A02' 第二列,
'WHITE'  LINKCOLBC 第二列,
'C003A03' 第三列,
'CYAN'   LINKCOLBC 第三列,
'C003A04' 第四列,
''        LINKCOLBC 第四列
UNION ALL
SELECT
'004'    行,
'C004A01' 第一列,
''        LINKCOLBC 第一列,
'C004A02' 第二列,
'YELLOW'  LINKCOLBC 第二列,
'C004A03' 第三列,
''        LINKCOLBC 第三列,
'C004A04' 第四列,
''        LINKCOLBC 第四列;
```

1.23 功能列 LINKCOLBGC

数据模型

列模型-LINKCOLBGC渐变色

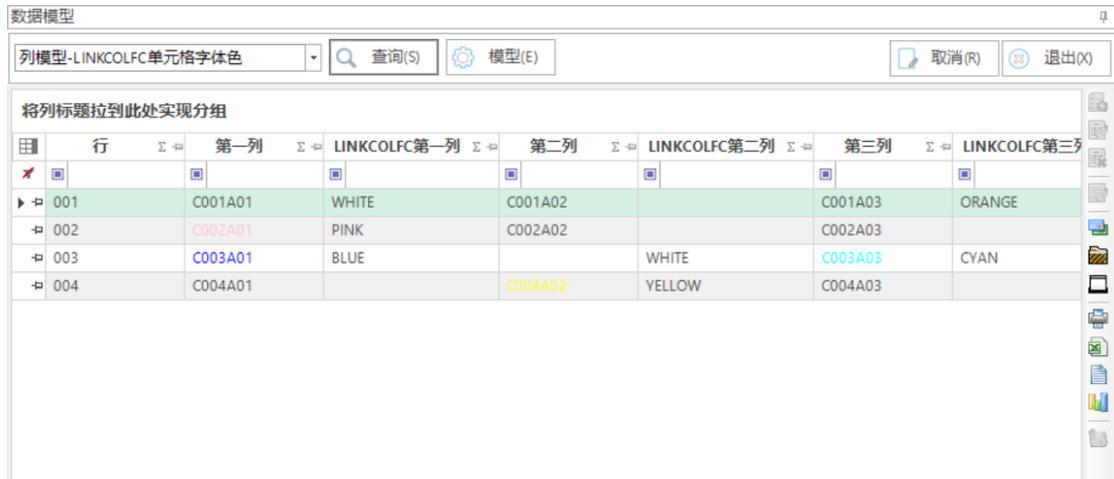
查询(S) 模型(E) 取消(R) 退出(X)

将列标题拉到此处实现分组

行	第一列	LINKCOLBGC第一列	第二列	LINKCOLBGC第二列	第三列	LINKCOLBGC
001	C001A01	WHITE	C001A02		C001A03	ORANGE
002	C002A01	PINK	C002A02		C002A03	
003	C003A01	BLUE	C003A02	WHITE	C003A03	CYAN
004	C004A01		C004A02	YELLOW	C004A03	

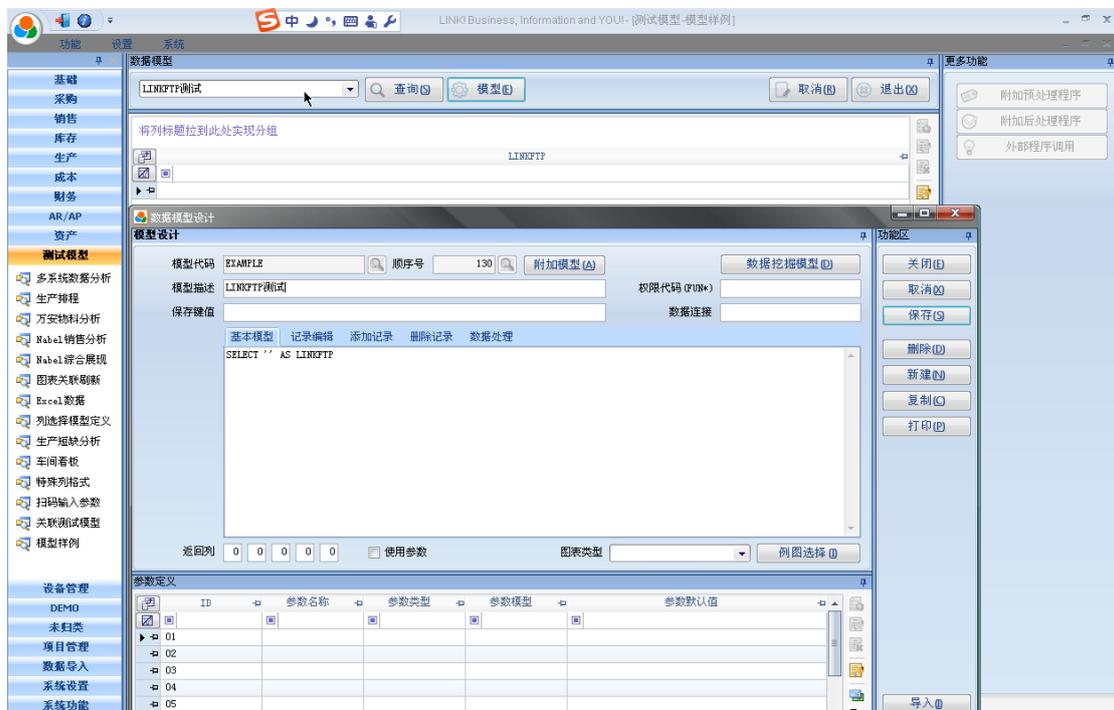
- LINKCOLBGC 功能列的作用于 LINKCOLBC 相似，当使用 LINKCOLBGC 时，所显示的颜色与原来的背景色过渡显示
- 如上图所示

1.24 功能列 LINKCOLFC



- LINKCOLFC 功能列的作用是控制单元格前景(字体)颜色
- 如上图所示

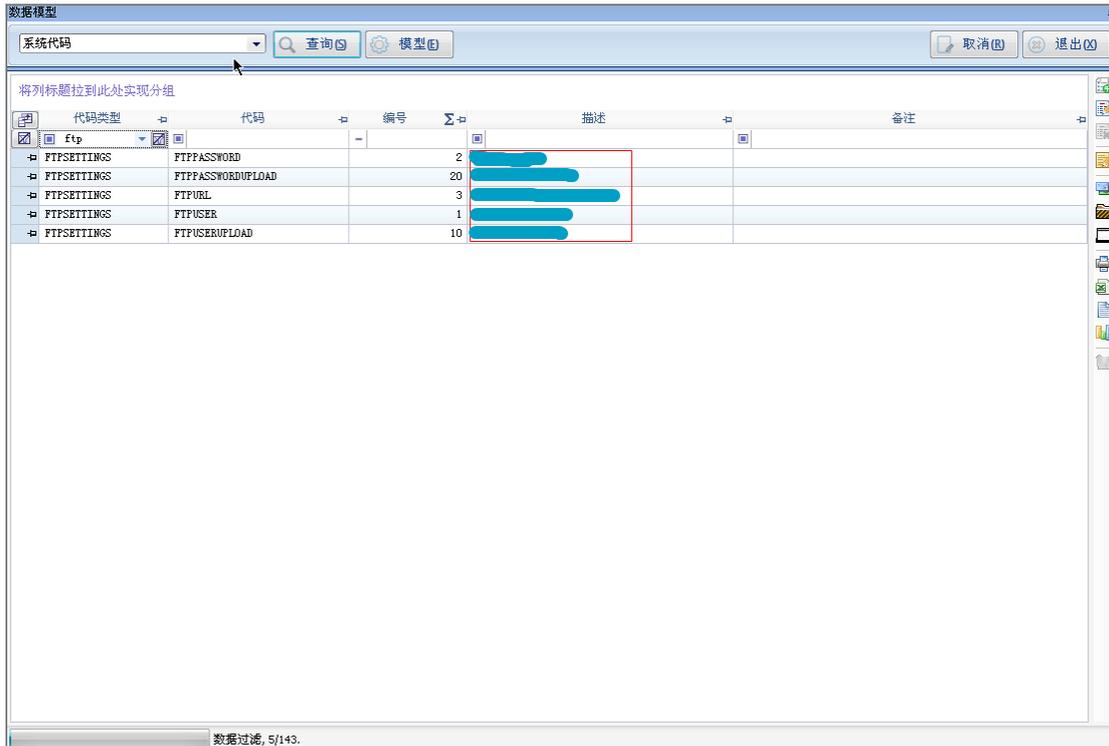
1.25 功能列 LINKFTP



- 当模型中存在 LINKFTP 命名的功能列时，系统会根据需要自动进行 FTP 文件上传处理
- 比如，上述模型的脚本为：

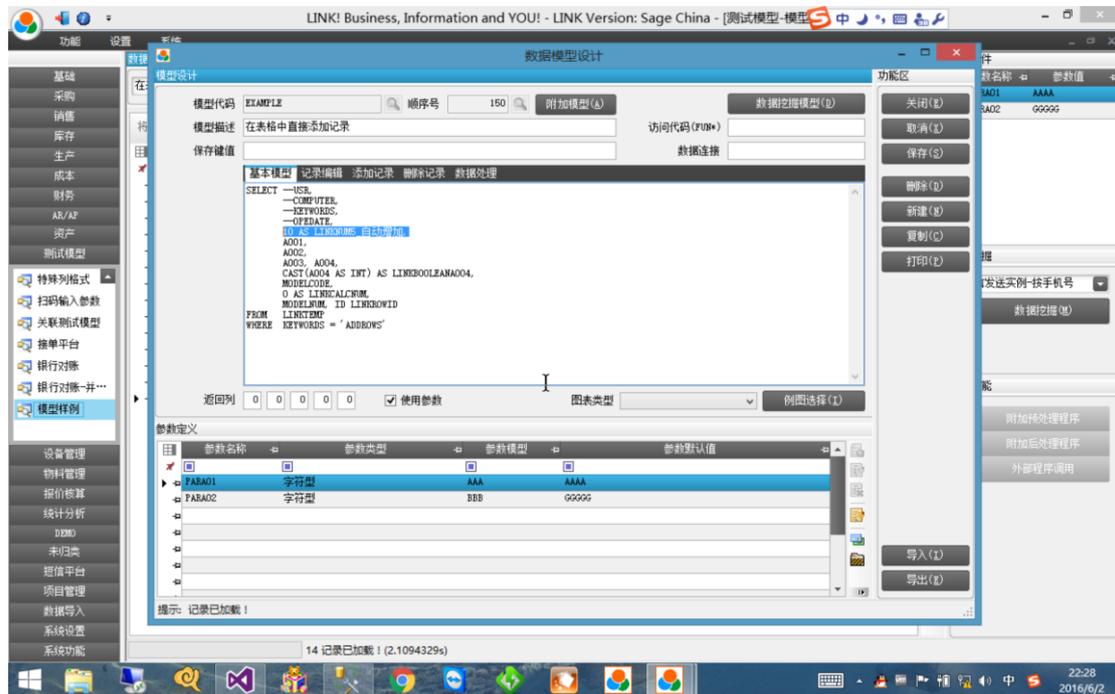
SELECT " AS LINKFTP

- 模型的更新脚本为：
--@LINKFTP@
- 这样，当用户编辑表格的内容时，该列可以像 LINKFILE 一样，允许用户选择文件
- 当执行保存时，系统会自动上传文件到对应的 ftp 服务器

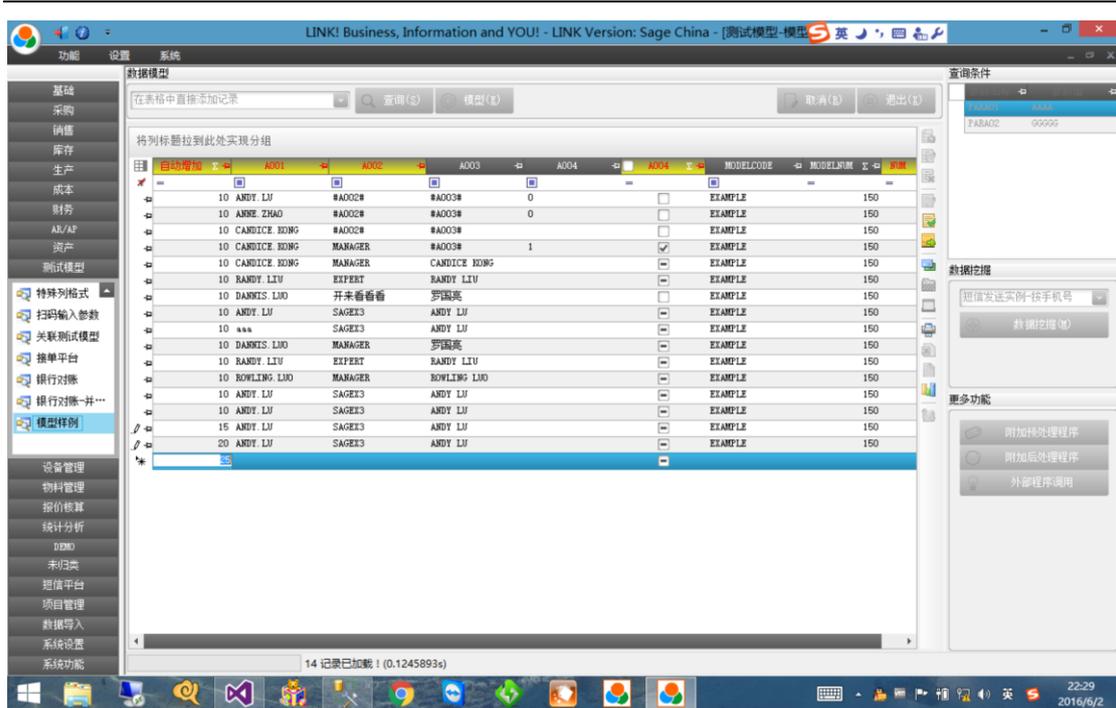


- 通过系统代码进行 ftp 上传的相关设置，包括：
 - FTPUSER——访问用户
 - FTPUSERUPLOAD——上传用户
 - FTPPASSWORD——访问密码
 - FTPPASSWORDUPLOAD——上传密码
 - FTPURL——FTP 服务器地址

1.26 功能列 LINKNUM



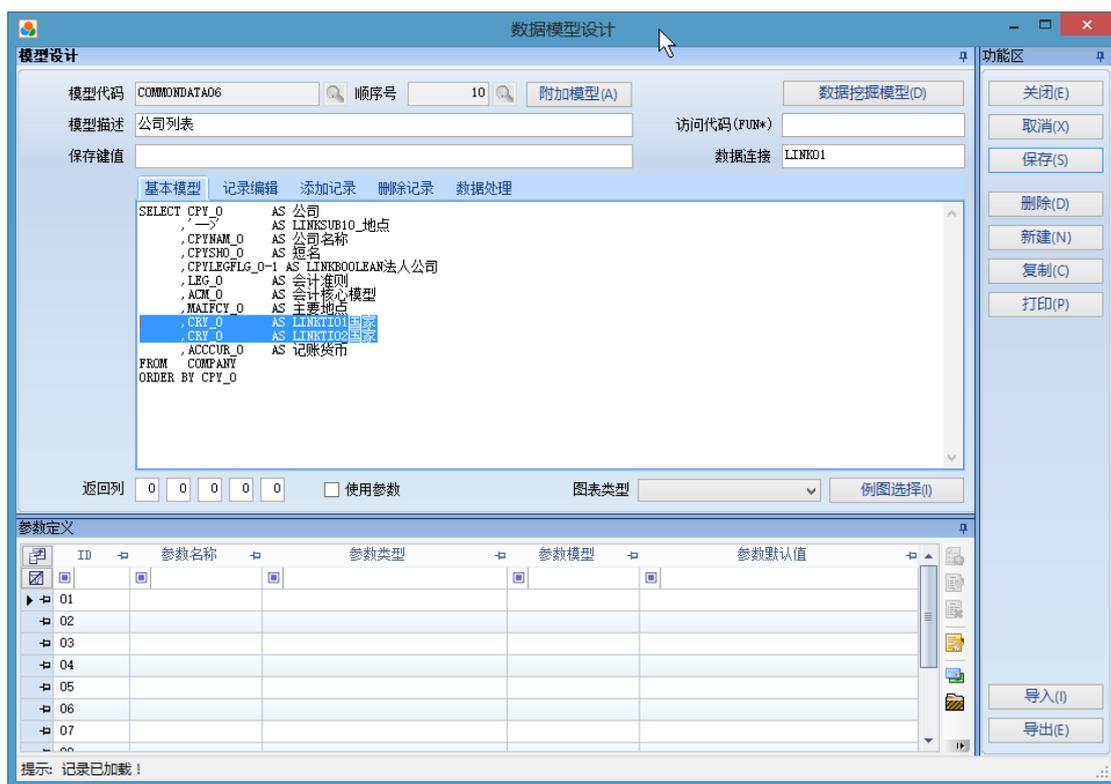
- 功能列 LINKNUM，用于处理编辑时的行号自动增长
- 列的使用方法为设置列的标题为类似 LINKNUM5_XXX
- 其中，
 - 数字 5，表示当自动添加记录的时候，新增加的行在上一行对应数据的基础上自动增加 5
 - 即：当前新添加行的数据=上一行数据+5



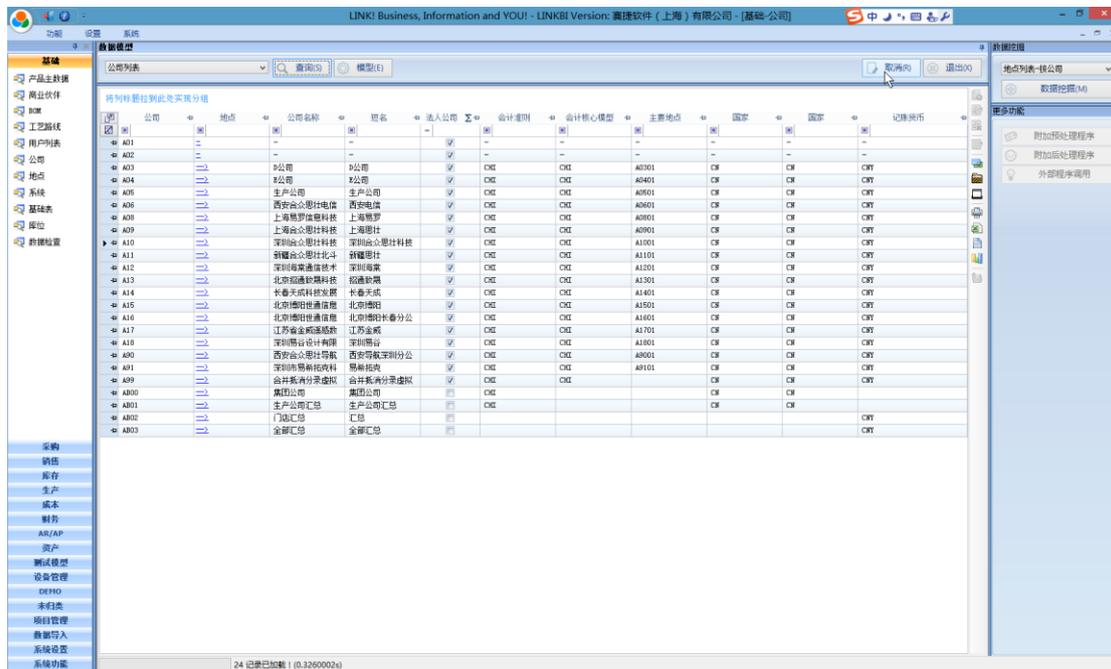
- 执行时的效果如图所示

1.27 列标题特殊处理 LINKTI01-LINKTI05

- 对于某些特殊的列，有时希望几个不同的列显示同样的标题（Caption）
- 可以分别使用 LINKTI01-LINKTI05 作为其列标题的前缀

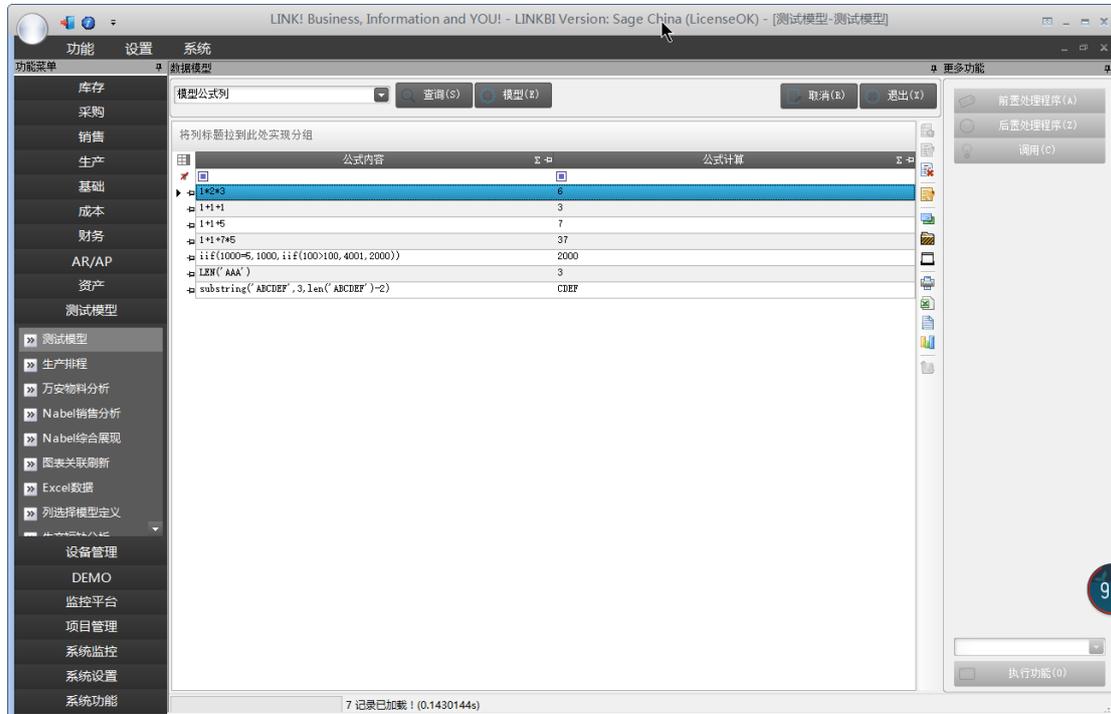


- 如上图所示，比如有两列分别为“LINKTIO1 国家、LINKTIO2 国家”
- 执行的结果如下图所示



1.28 功能列 LINKFORMULA

- 当数据信息中的“公式的内容”，可以用这个特殊的列来计算“公式的值”



- 如图所示，该模型获取的结果中有一列的内容是根据“公式内容”计算的结果。

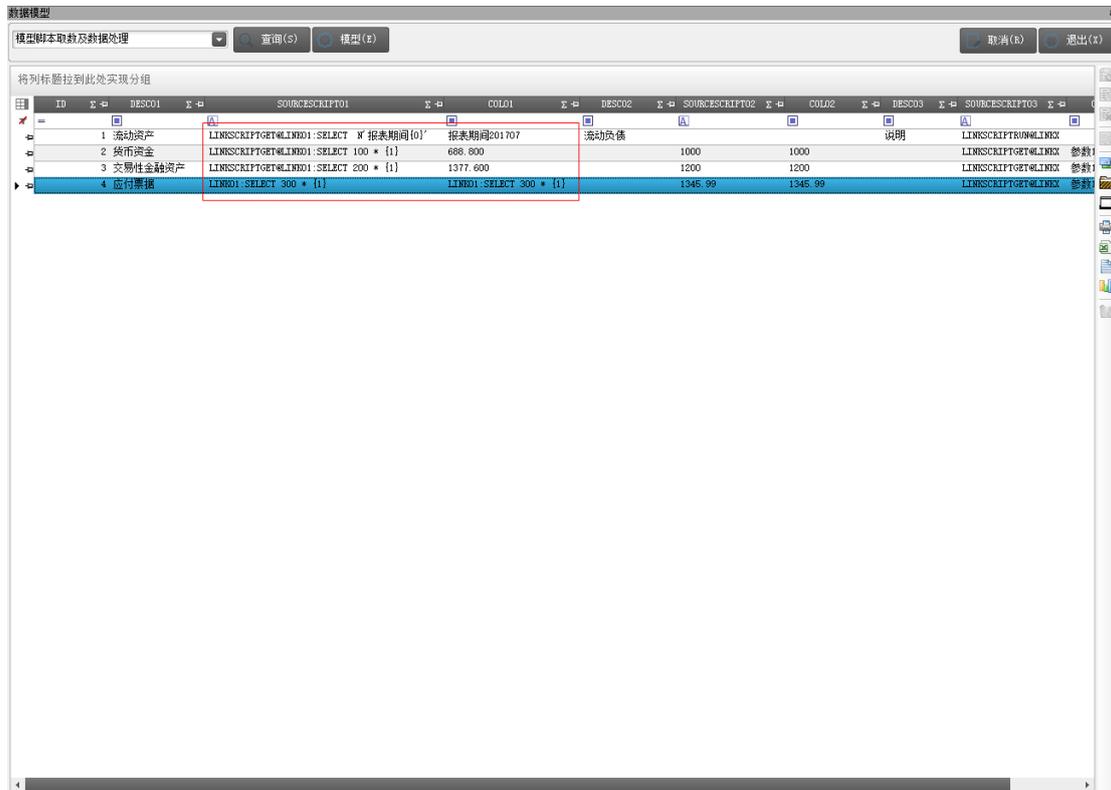
- 模型的脚本如下：

```
SELECT '1*2*3' 公式内容, '1*2*3' AS LINKFORMULA 公式计算
UNION ALL
SELECT '1+1+1', '1+1+1'
UNION ALL
SELECT '1+1+5', '1+1+5'
UNION ALL
SELECT '1+1+7*5', '1+1+7*5'
UNION ALL
SELECT 'iif(1000=5,1000,iif(100>100,4001,2000))',
'iif(1000=5,1000,iif(100>100,4001,2000))'
UNION ALL
SELECT 'LEN('AAA')', 'LEN('AAA')'
UNION ALL
SELECT 'substring('ABCDEF',3,len('ABCDEF')-2)',
'substring('ABCDEF',3,len('ABCDEF')-2)'
```

- 在系统中执行上述模型，对于“公式计算”这一列，可以直接获取计算结果

1.29 功能列 LINKSCRIPT

- 某些情况下, 我们希望在单元格中填写一段数据处理脚本 (SQL Script), 用于根据脚本执行的结果, 获取或者影响单元格的数据。这时, 可以使用 LINKSCRIPT 功能列。



- 如图所示, 相关的脚本如下:

```
SELECT ID
      , DESC01
      , LINKSCRIPTCOL01 SOURCESCRIPT01
      , LINKSCRIPTCOL01
      , DESC02
      , LINKSCRIPTCOL02 SOURCESCRIPT02
      , LINKSCRIPTCOL02
      , DESC03
      , LINKSCRIPTCOL03 SOURCESCRIPT03
      , LINKSCRIPTCOL03
FROM   LINKSCRIPTDEMO
ORDER BY ID
```

- 在上述脚本中, 有某些列的列名 (最终输出的列名或别名) 中是以 LINKSCRIPT 开头的, 系统在执行到这些列时, 会检查列的内容, 并对脚本进行处理

- 该列的内容中, 可以是具体的内容或者是一段可执行的数据处理脚本, 如以下几种情况

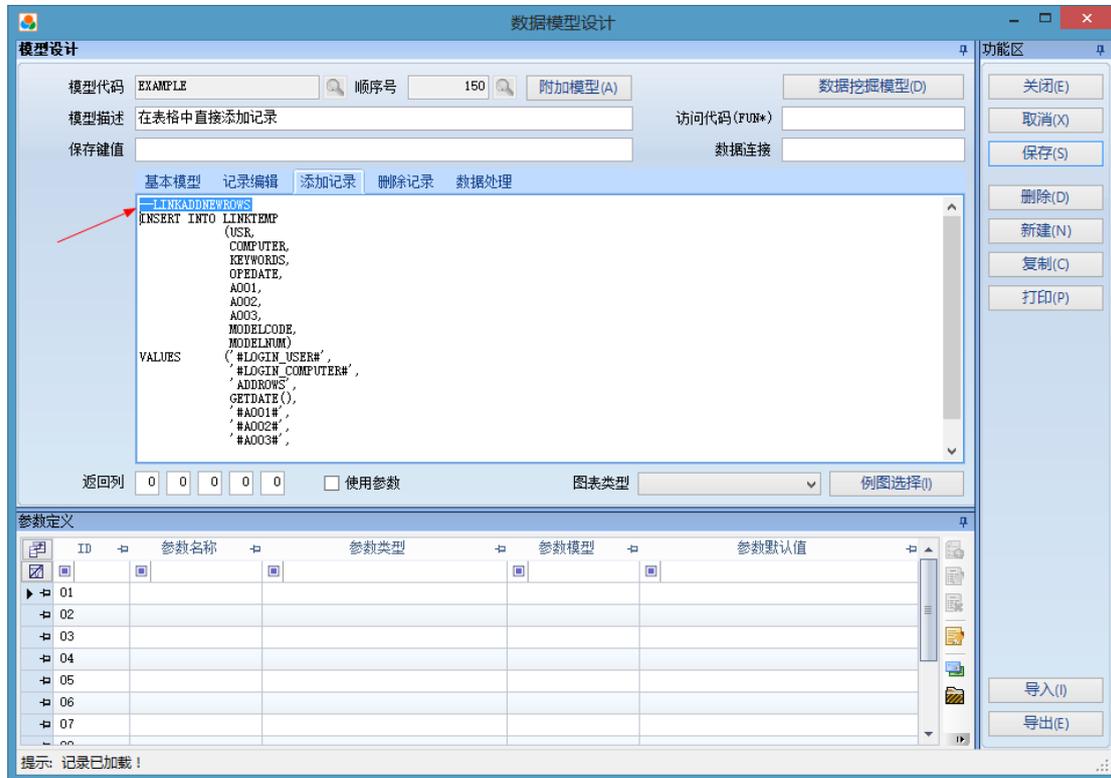
■ LINKSCRIPTRUN@CONNCODE:SQL SCRIPT 格式

- ◆ 比如: LINKSCRIPTRUN@LINKX3DEMO:SELECT N'说明的内容...'

- ◆ 这时，系统根据需要在 CONNCODE 对应的数据链接中执行相应的脚本，可以进行必要的数据处理
- ◆ 此时，单元格的内容保持空白
- LINKSCRIPTGET@CONNCODE:SQL SCRIPT 格式
 - ◆ 比如：LINKSCRIPTGET@LINK01:SELECT 200 * {1}
 - ◆ 这时，系统根据需要在 CONNCODE 对应的数据链接中执行相应的脚本，可以进行必要的数据处理，同时，根据执行的结果，将返回的数据填写在相应的单元格中
 - ◆ 此时，该单元格中的数据为 1377.600 (=200*6.888)
- 其他内容
 - ◆ 当单元格的内容不符合上述格式的时候，系统将其视为内容，而不会作为脚本执行
- 说明
 - 单元格中的脚本可以包含参数，在系统执行时，会以实际模型参数的值替代，用于获取实际的数据
 - 单元格中的脚本可以包含全局变量，在系统执行时，会以实际变量的值替代，用于获取实际的数据

2 特殊标识

2.1 利用添加行添加记录--LINKADDNEWROWS

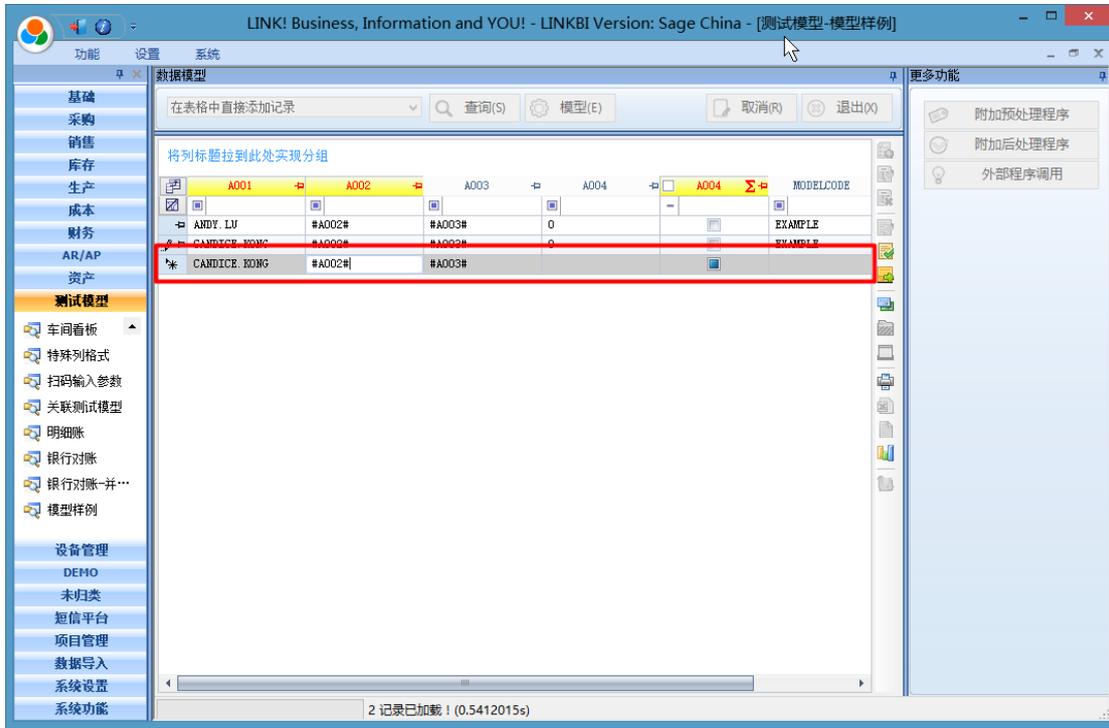


- 当模型的添加记录脚本的第一行以特殊的标记开始时，系统任务该模型在编辑时可以开启行添加功能
- 这个特殊的标记是：**--LINKADDNEWROWS**
- 开启此功能后，在数据的最后以后允许用户输入新的记录
- 可以编辑的列仍然由记录编辑模型定义



- 表格中的数据编辑时，会自动获得上一行的数据内容，并且允许用户编辑

- 保存数据时，系统会自动判断是添加的新行记录还是已经存在的记录，并对其进行分别处理
- 如下图所示



2.2 模型中的数据挖掘列--LINKSUBxx_XXX



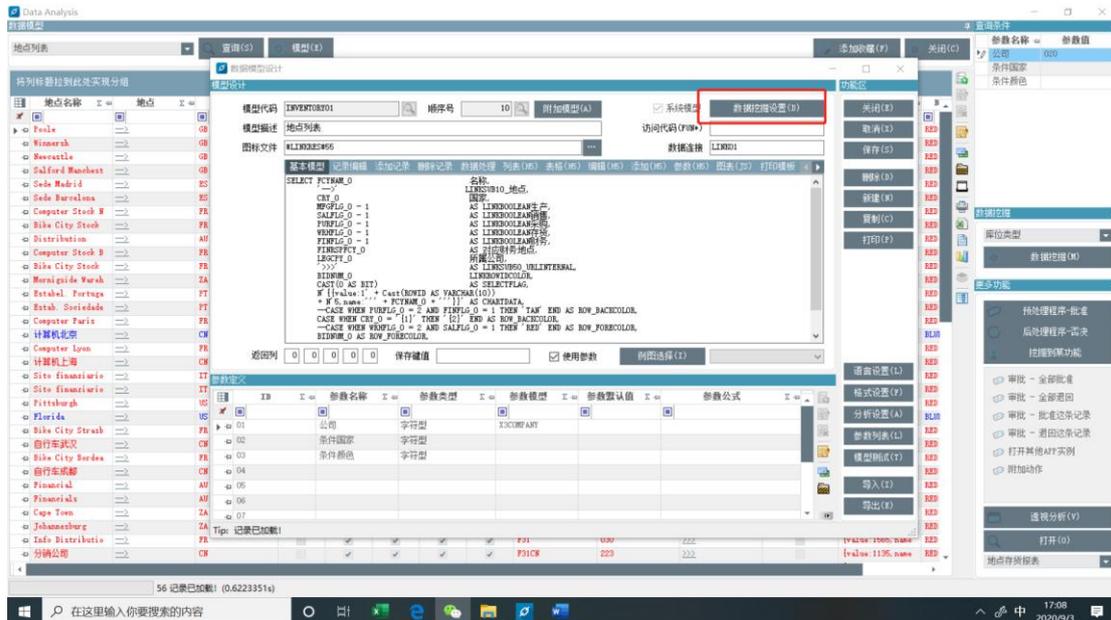
- 如上图所示，当为模型定义了数据挖掘之后，还可以在数据表行中定义数据挖掘列

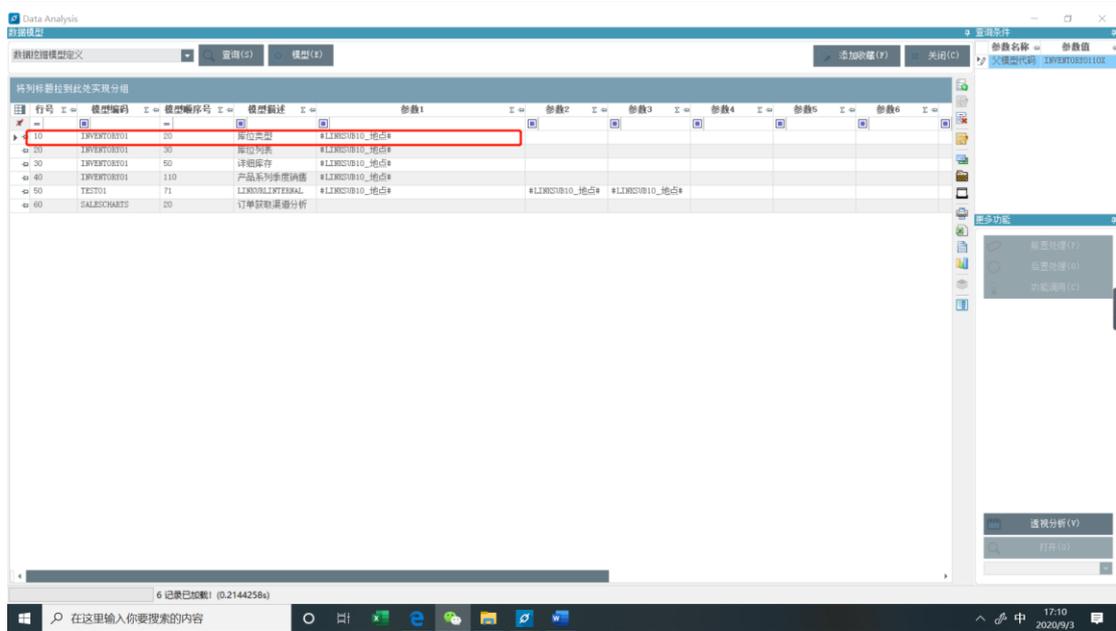
- 上例中
 - 产品地点一列以“→”标记，点击该信息可以数据挖掘到对应地点下的产品地点信息
 - 库存列也是一个数据挖掘列，点击可以查看该地点下的库存数据
- 上述数据挖掘列是以“LINKSUBxx_XXXX”标识的
- 模型的脚本如下：

```

SELECT FCYNAM_0          名称,
'-->'                  LINKSUB10_地点,
CRY_0                  国家,
MFGFLG_0 - 1          AS LINKBOOLEAN 生产,
SALFLG_0 - 1          AS LINKBOOLEAN 销售,
PURFLG_0 - 1          AS LINKBOOLEAN 采购,
WRHFLG_0 - 1          AS LINKBOOLEAN 存货,
FINFLG_0 - 1          AS LINKBOOLEAN 财务,
FINRSPFCY_0           AS 对应财务地点,
LEGPCPY_0             所属公司,
'>>>'                 AS LINKSUB50_URLINTERNAL,
BIDNUM_0              LINKROWIDCOLOR,
CAST(0 AS BIT)        AS SELECTFLAG,
N'{{value:1' + Cast(ROWID AS VARCHAR(10))
+ N'5,name:'' + FCYNAM_0 + ''}}' AS CHARTDATA,
--CASE WHEN PURFLG_0 = 2 AND FINFLG_0 = 1 THEN 'TAN' END AS ROW_BACKCOLOR,
CASE WHEN CRY_0 = '{1}' THEN '{2}' END AS ROW_BACKCOLOR,
--CASE WHEN WRHFLG_0 = 2 AND SALFLG_0 = 1 THEN 'RED' END AS ROW_FORECOLOR,
BIDNUM_0 AS ROW_FORECOLOR,
BIDNUM_0
FROM FACILITY
ORDER BY FCY_0
    
```

- 点击“数据挖掘”，在跳转的挖掘页面定义该字段挖掘到哪个模型





2.3 全局变量

- 灵活运用特殊的全局变量可以使得模型的编制更加简单
- 常用：
 - #LOGIN_USER#——当前登录的用户代码
 - #LOGIN_ROLE#——当前登录的用户对应的角色代码
 - #LOGIN_ID#——当前登录用户对应的 ID
 - #LOGIN_COMPUTER#——当前登录的电脑名称
 - #LINKSERVER#——当前访问的服务器基本地址
 - #LINKCUSCODE#——服务器指定的客户代码
 - #CURR_MODELCODE#——当前执行的模型的模型代码
 - #CURR_MODELNUM#——当前执行的模型的模型编号
 - #ATTACH_MODELCODE#——当前执行的模型的附加模型代码
 - #LINKLOCATION#——当前地理位置信息（字符串）
 - #LINKLOCATIONJSON#——当前地理位置信息（JSON 格式数据）
 - #LINKID_RESULT#——为统一的 GUID 号码，每次执行附加模型的动作，相当于 NEWID()
 - #LINKID_ACTION#——为统一的 GUID 号码，每次执行附加模型的动作，相当于 NEWID()
 - #LINKREPORTPATH#——ActiveReport 报表路径，默认为系统默认文件夹下的 Reports 文件夹（系统右键左上角图--我的文件-Reports）；使用方式举例：
"reportFile": "#LINKREPORTPATH#\\daily.rdlx", 具体使用见文档《报表设计应用》
 - #LINKCONN.LINK01#--替代数据源连接语句

- 与时间有关:

- #LINK_TODAY#——当前日期 (yyyy-mm-dd)
- #LINK_YESTERDAY#——昨天 (yyyy-mm-dd)
- #LINK_TOMORROW#——明天 (yyyy-mm-dd)
- #LINK_THISYEAR#——今年 (yyyy)
- #LINK_THISMONTH#——当月 (mm)
- #LINK_THISYEARMONTH#——当前年月 (yyyy-mm)
- #LINK_THISYEARMONTH6#——当前年月 (yyyymm)
- #LINK_THISYEARMONTH#-01——今年当前月份的第一天 (yyyy-mm-01)
- #LINK_FIRSTDAY_THISMONTH#——当前月份的第一天 (yyyy-mm-01)
- #LINK_LASTDAY_THISMONTH#——当前月份的最后一天 (yyyy-mm-xx)
- #LINK_LASTYEAR#——去年 (yyyy)
- #LINK_LASTMONTH#——上个月 (mm)
- #LINK_LASTYEARMONTH#——当前月份减去一年 (yyyy-mm)
- #LINK_LASTYEARMONTH6#——当前月份减去一年 (yyyymm)
- #LINK_FIRSTDAY_LASTMONTH#——上个月的第一天 (yyyy-mm-01)
- #LINK_LASTDAY_LASTMONTH#——上个月最后一天 (yyyy-mm-xx)
- #LINK_TIMETICKS#——时间戳
- #LINK_TIMEHM#——当前时分 (hh:mm)
- #LINK_TIMEHMS#——当前时分秒 (hh:mm:ss)
- #LINK_FULLTIME#——当前时间 (yyyy-MM-dd hh:mm:ss.ffff)

- 日期格式处理 LINKDATE

- #LINKDATE(0,0,0,yyyyMM)#——获取当前年月
- #LINKDATE(0,0,0,yyyyMMdd)#——获取当前年月日
- #LINKDATE(0,0,0,yyyy-MM-dd)#——获取当前年月, 格式为 yyyy-MM-dd
- #LINKDATE(0,-1,0,yyyyMM)#——当前年月的上一个月
- #LINKDATE(-1,0,0,yyyyMM)#——去年的当前日期
- #LINKDATE(0,0,-1,yyyyMMdd)#——昨天的日期
- #LINKDATE(0,0,1,yyyyMMdd)#——明天的日期
- #LINKDATE(10,0,0,yyyy-MM-dd)#——10年的当前日期
- #LINKDATE(0,0,0,yyMM)#——当前年月, 格式为 yyMM
- #LINKDATE(0,0,0,yyyy-MM-01)#——当前年月+固定日期 01

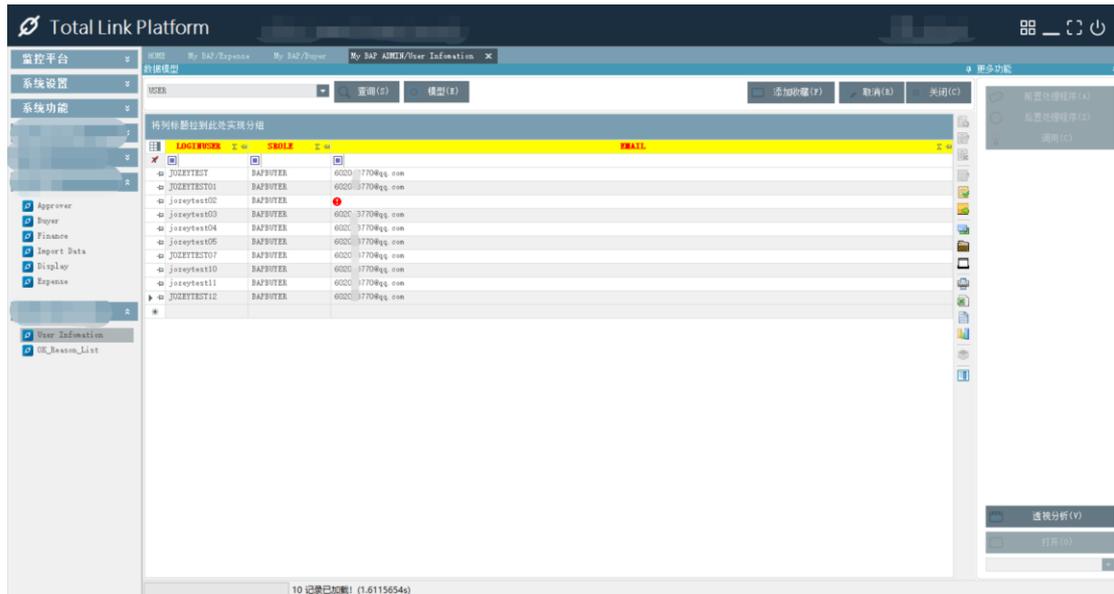
说明:

Y、M、D 分别表示向前或向后移动的年份、月份、日期数字, 可以是负数, 或为正数, 则表示当前年、月或天加上该数。

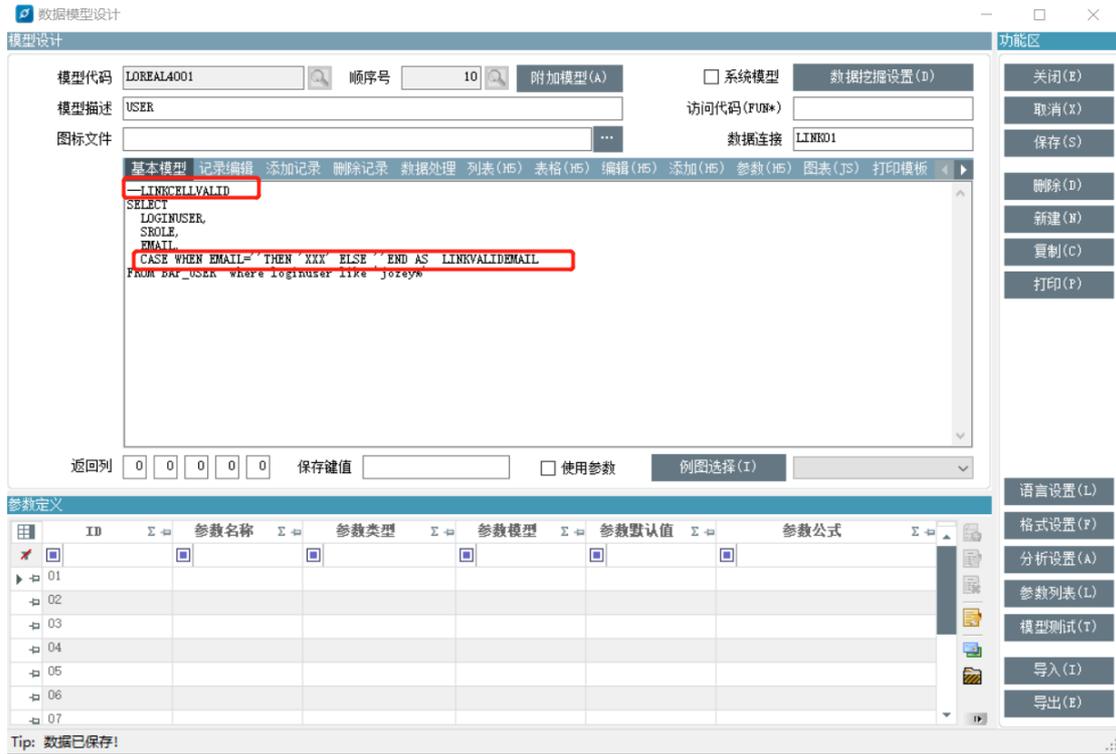
定义日期格式当 Y、M、D 为 0 时, 这表示当前日期, 若遇到其他的日期格式, 可参考网址: <https://docs.microsoft.com>。

2.4 单元格数据校验--LINKCELLVALID

- 触发该校验机制，需要在代码最上方添加关键代码“**LINKCELLVALID**”
- 查询代码新增一个判断字段 **LINKVALIDXXX(XXX 为校验字段名)**，如果该判断字段值为空，则数据验证通过；如果该判断字段值不为空，系统会提示错误，并且再次编辑时不允许离开错误行



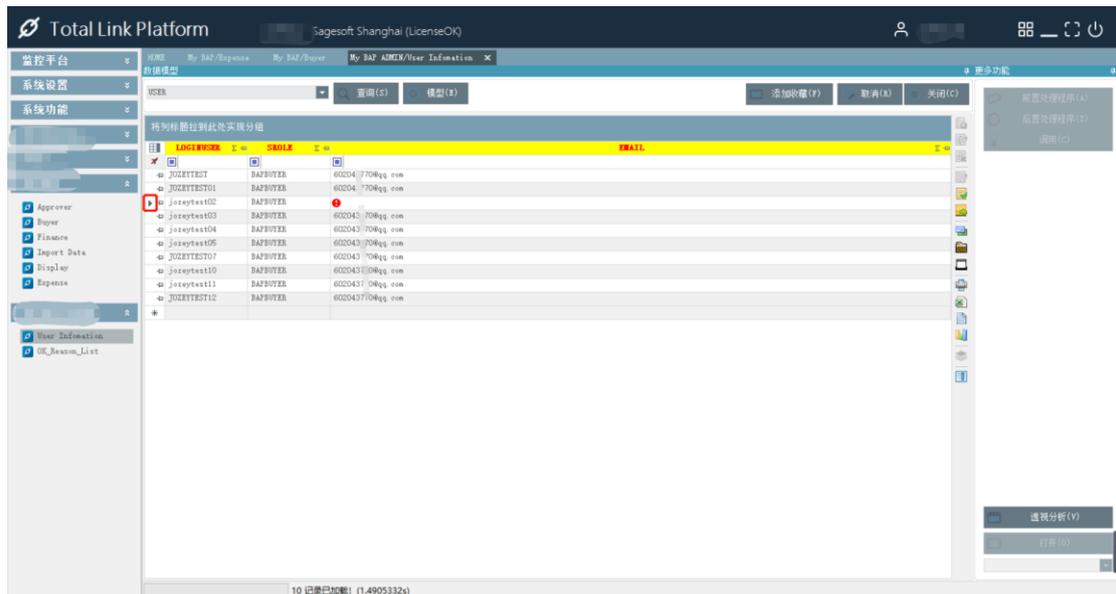
- 该模型判断 **EMAIL** 是否为空，为空则报错，不为空则验证通过



参考代码:

```
-- LINKCELLVALID
SELECT
  LOGINUSER,
  SROLE,
  EMAIL,
  CASE WHEN EMAIL=' ' THEN 'XXX' ELSE ' ' END AS LINKVALIDEMAIL
FROM BAP_USER where loginuser like 'jozey%
```

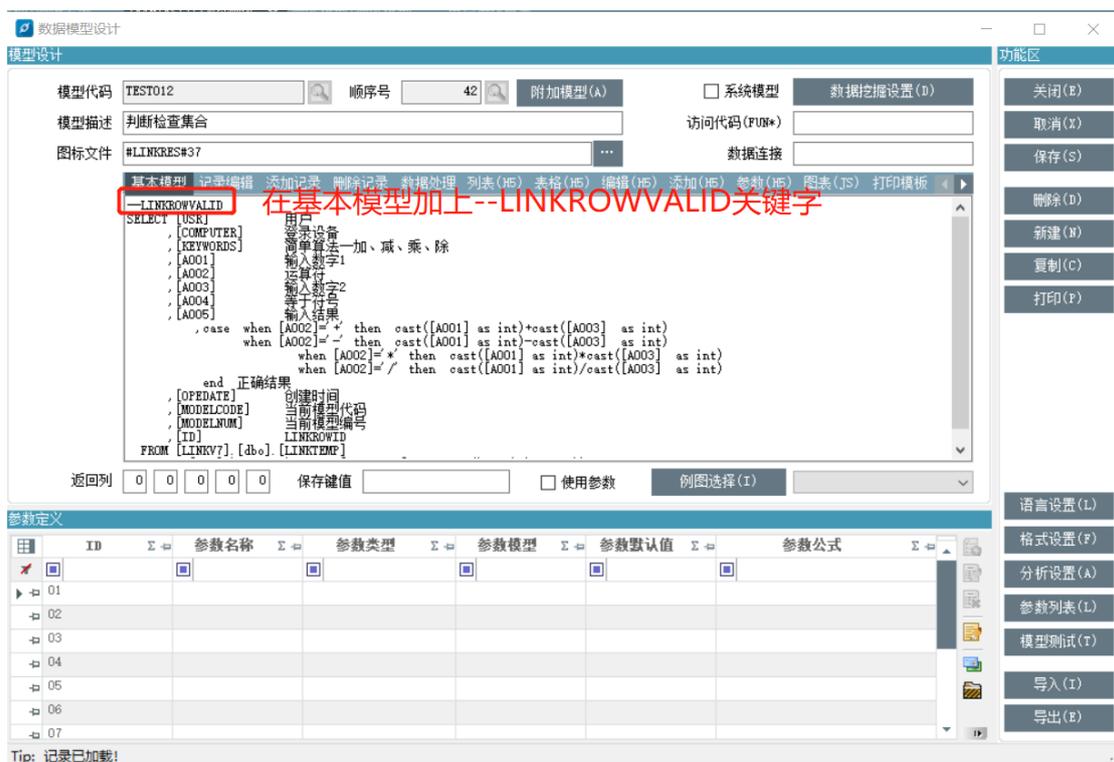
- 效果如下，查询数据时邮箱为空的单元格前提示“!”，并且在编辑时仅能编辑该行



2.5 保存时逐行检查--LINKROWVALID

- 触发该校验机制，需要在基本模型添加关键代码“--LINKROWVALID”，用于离开行时对行数据进行保存验证，验证成功可以离开行，验证不成功提示信息，并不允许离开。
- 具体实例如下：

检查输入的计算结果与正确的结算结果做对比检查，输入的计算结果不正确时，给出提示，并不允许离开改行，直到修正正确结果。

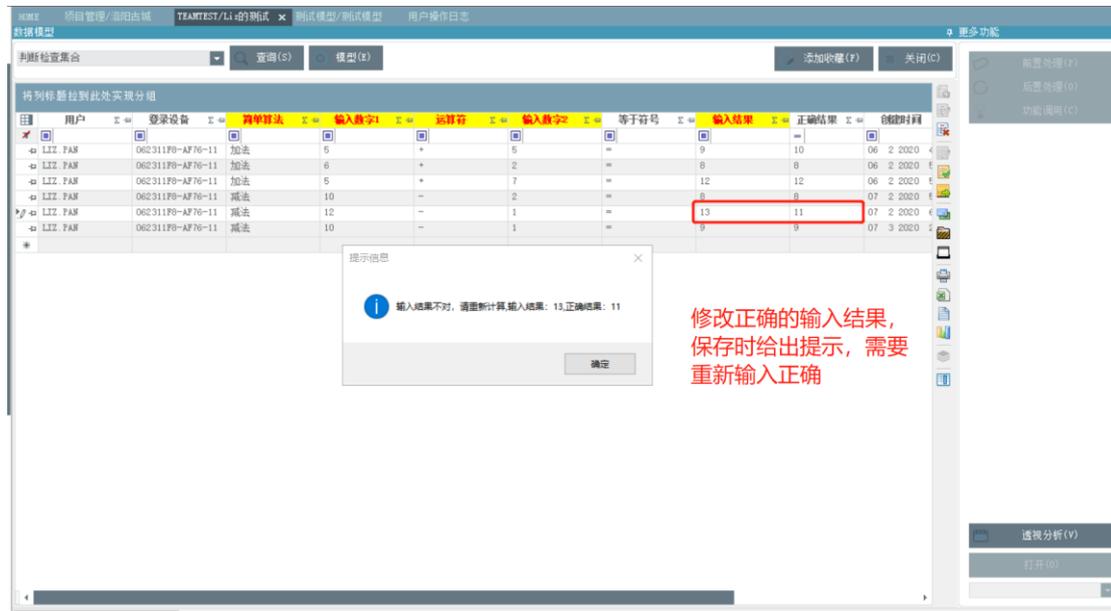




参考代码：

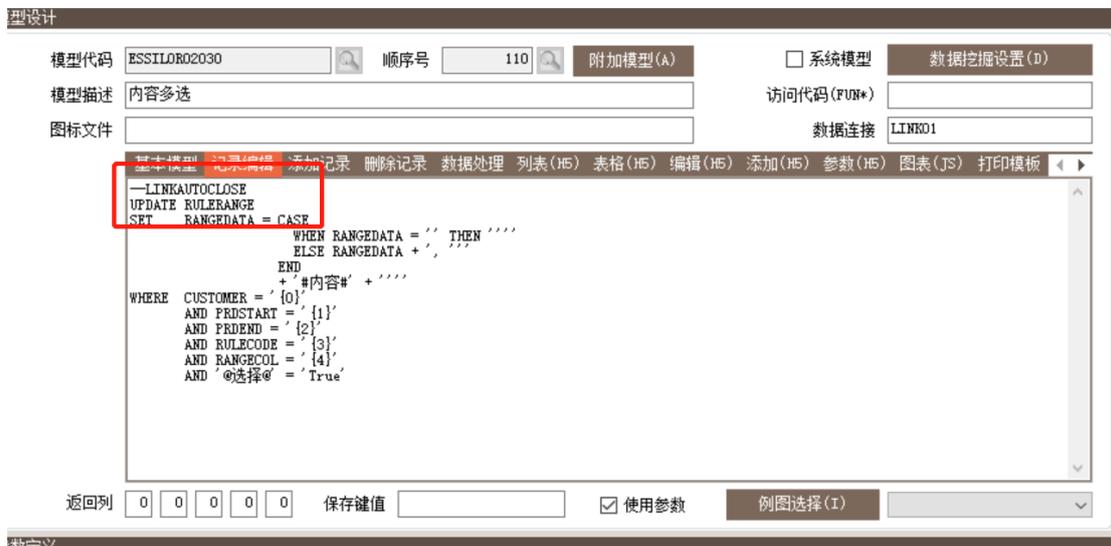
```

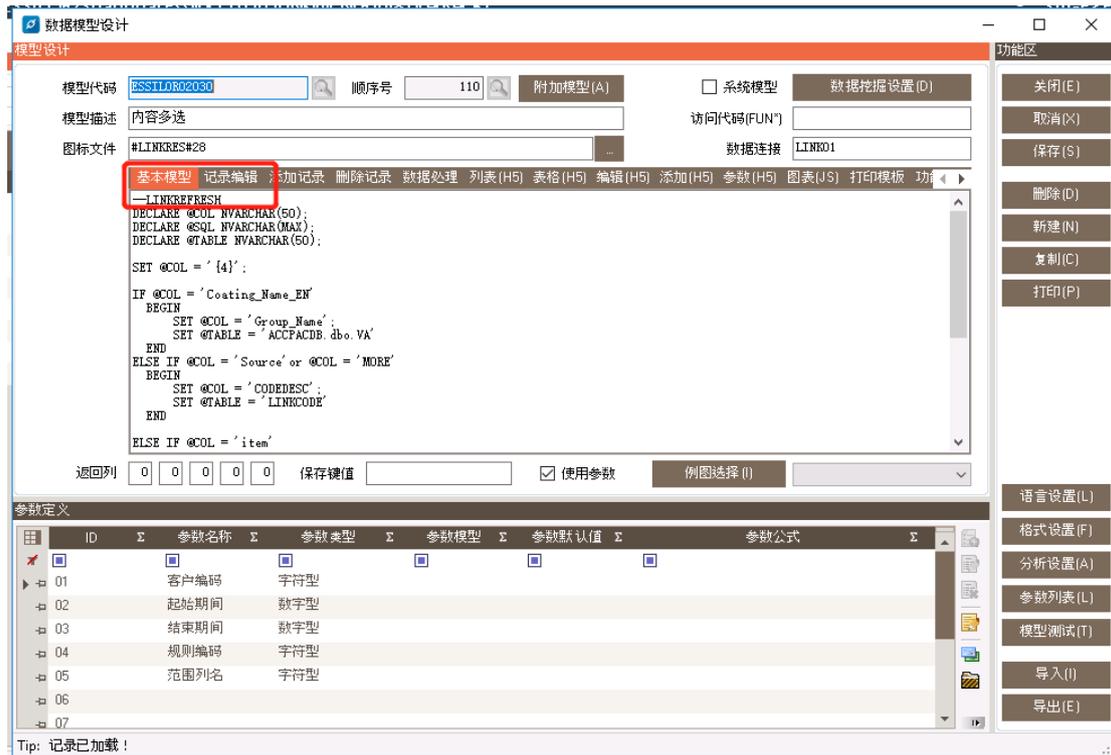
{"type":3,"message":"输入结果不对，请重新计算,输入结果：#输入结果#,正确结果：#正确结果#","condition":"LINKFORMULA:#输入结果#/#正确结果#"}
    
```





2.6 挖掘模型执行结束自动关闭并返回刷新列表 -- LINKAUTOCLOSE

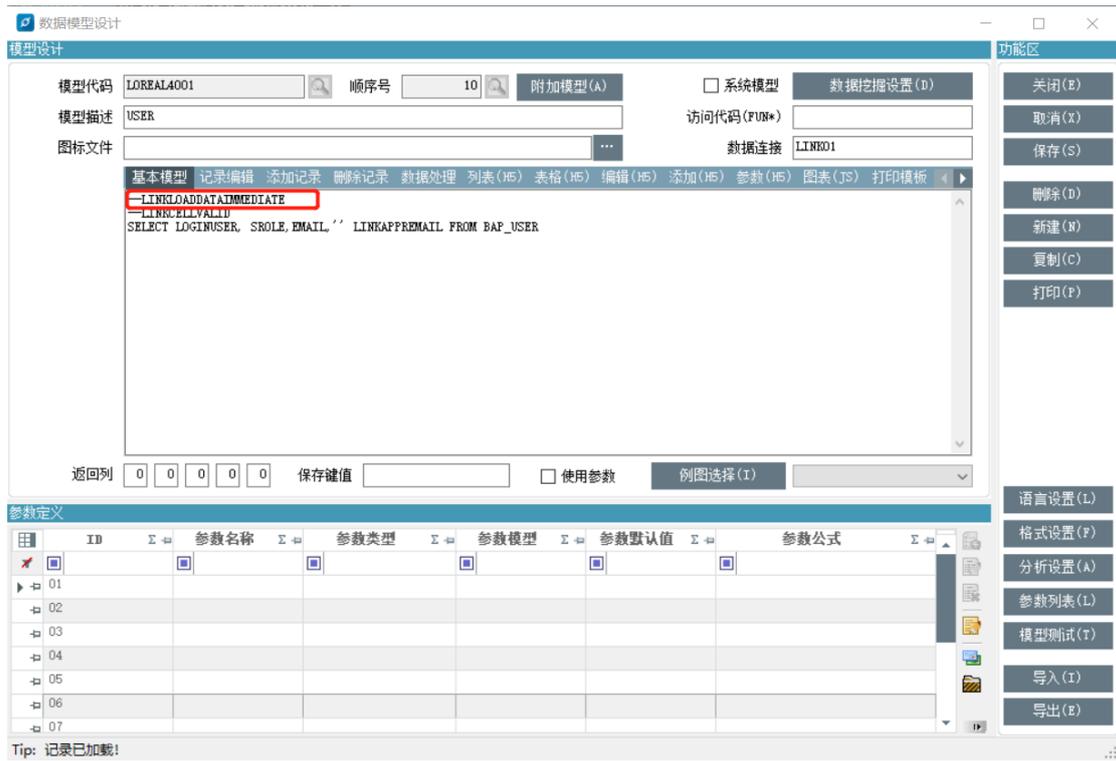




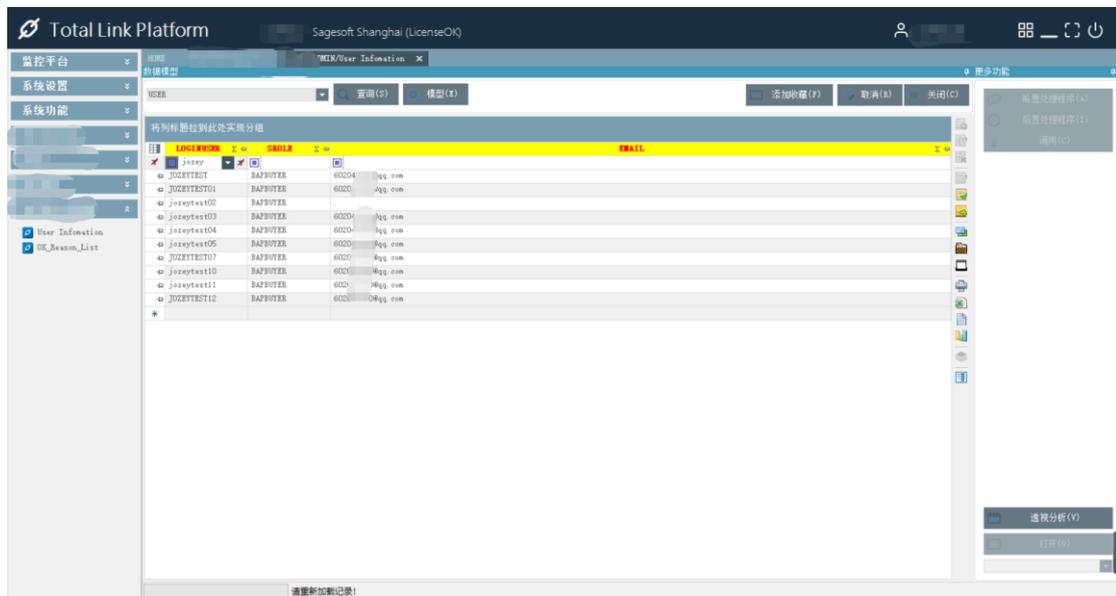
- **--LINKAUTOCLOSE**:挖掘后的模型（记录编辑）加入这行代码，意为执行结束后自动关闭该模型并返回上一个查询模型，并搭配**--LINKREFRESH**（基本模型添加）一起使用实现自动刷新。

2.7 查询模型直接加载数据—— LINKLOADDATAIMMEDIATE

- 在查询模型加入“**--LINKLOADDATAIMMEDIATE**”，可以实现打开页面直接加载数据

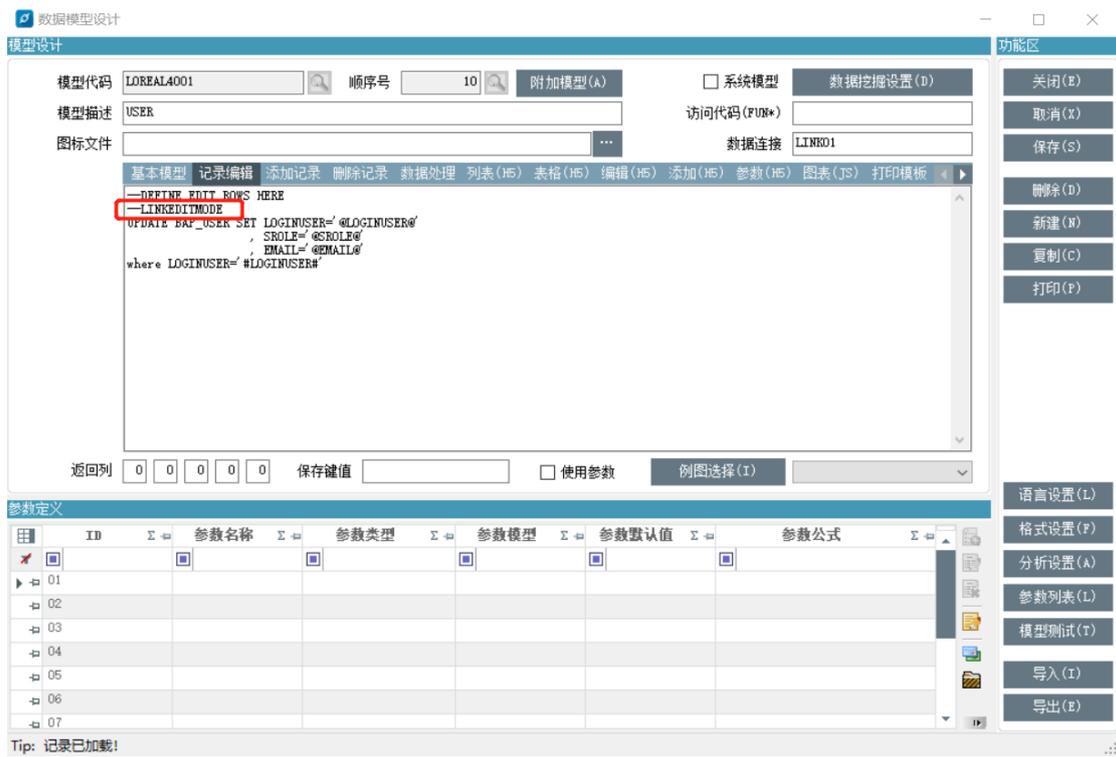


- 实现效果如下，用户进入页面不需要点击“查询”按钮便可以直接看到数据

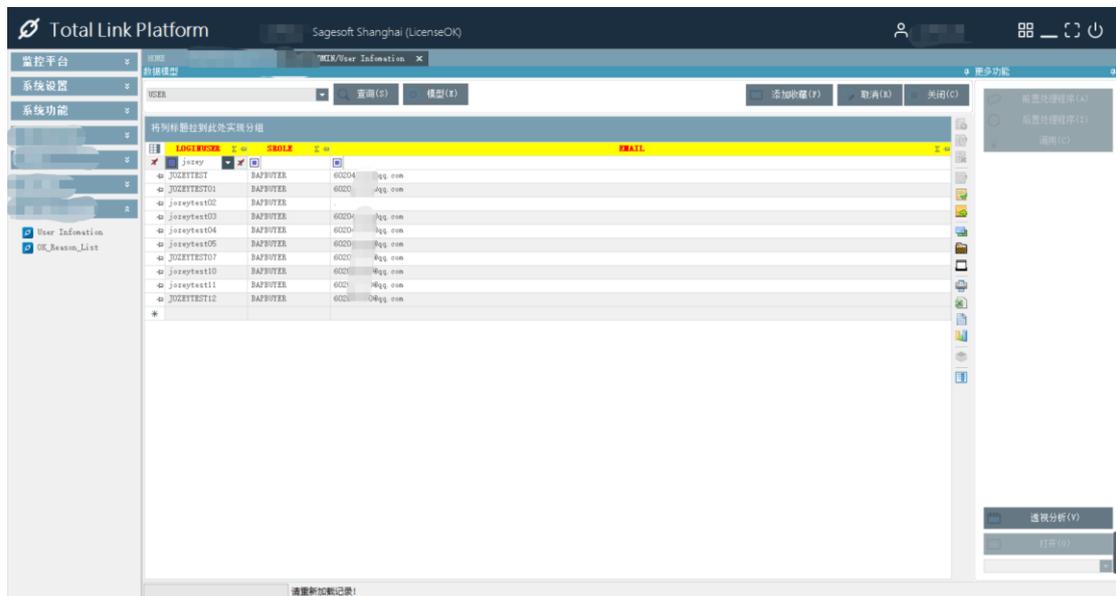


2.8 直接进入编辑模式--LINKEDITMODE

- 在编辑模型加入 "--LINKEDITMODE"，可以实现打开页面直接进入编辑模式

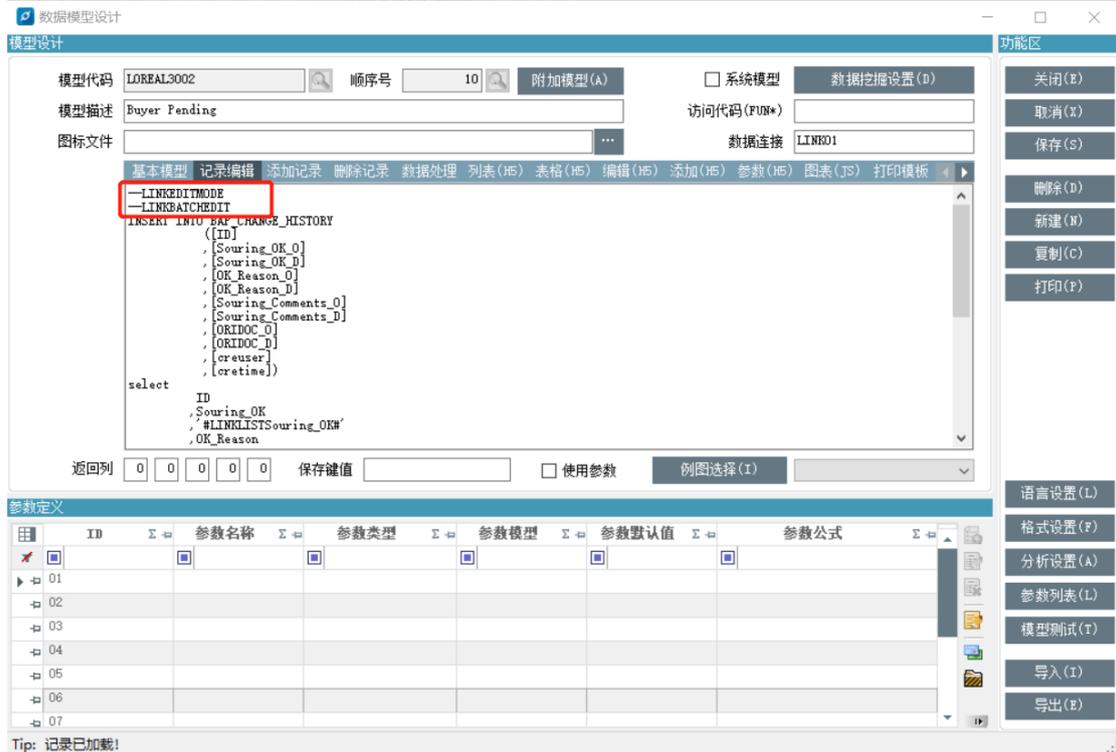


- 实现效果如下，用户进入页面不需要点击编辑按钮便可以实现编辑

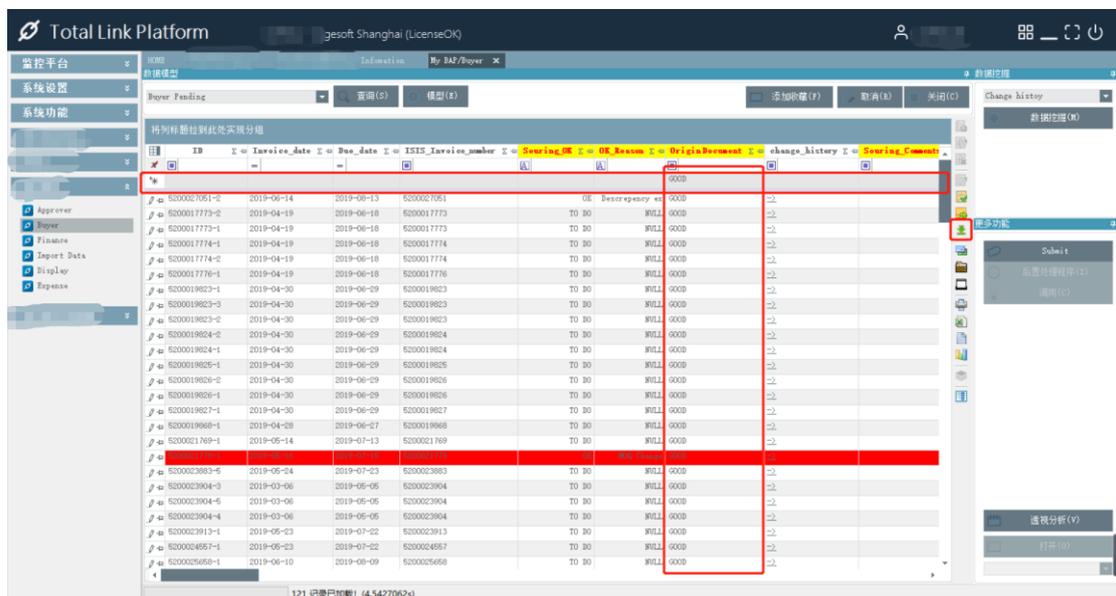


2.9 批量编辑数据--LINKBATCHEDIT

- 在编辑模型加入“--LINKBATCHEDIT”和“--LINKEDITMODE”，可以实现批量录入



- 实现效果如下：在灰色空白行可编辑列输入值，点击“批量”按钮即可

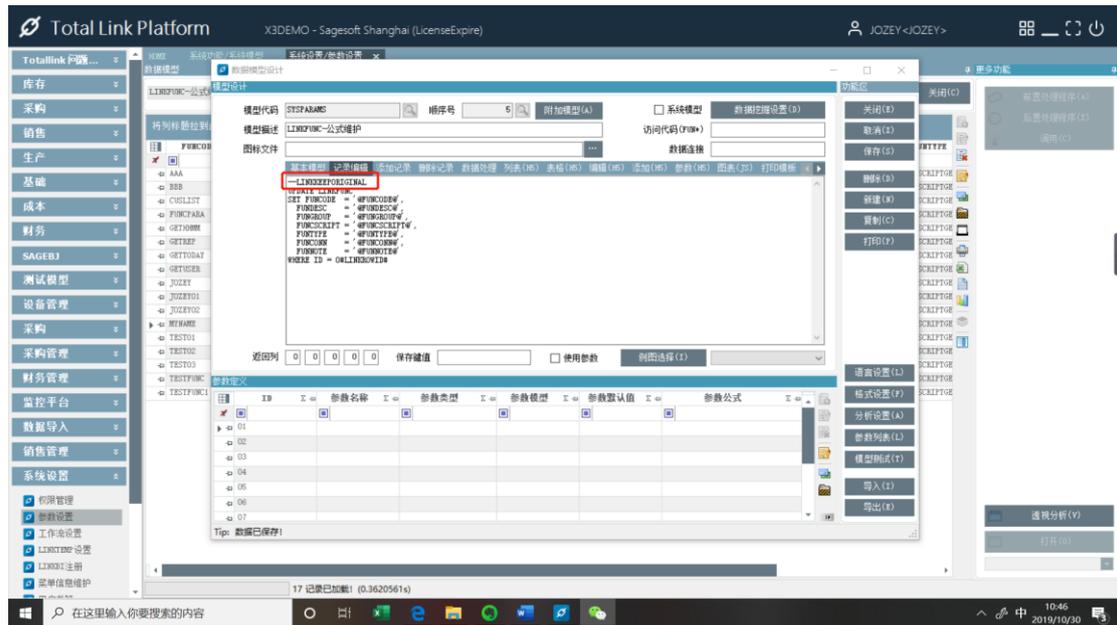


- 注:批量编辑和添加功能不能同时存在

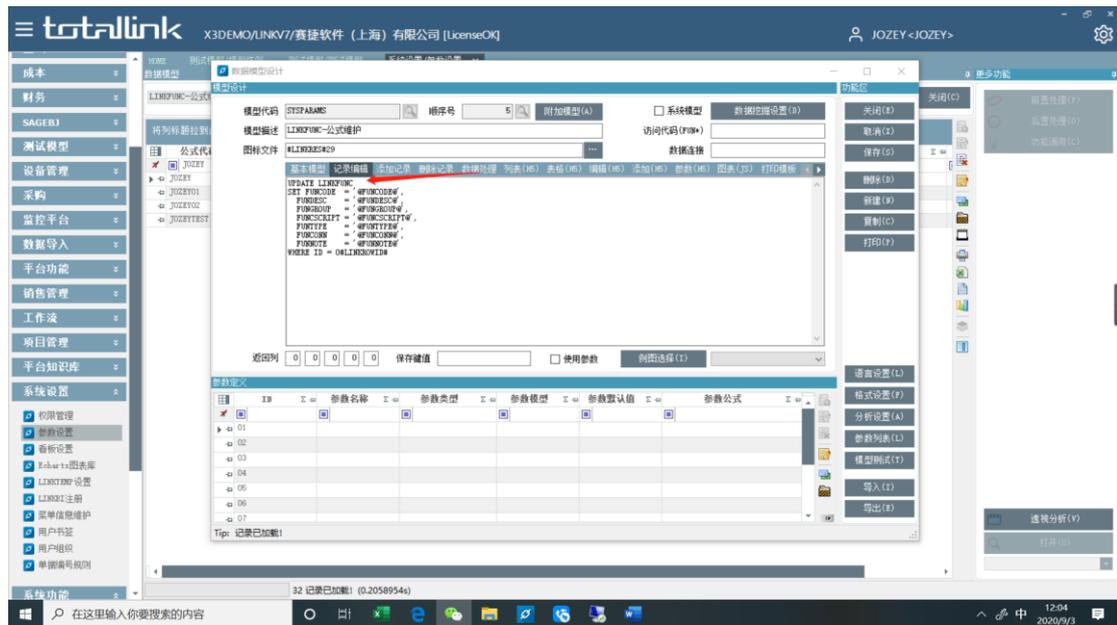
2.10 保留代码原始模式--LINKKEEPORIGINAL

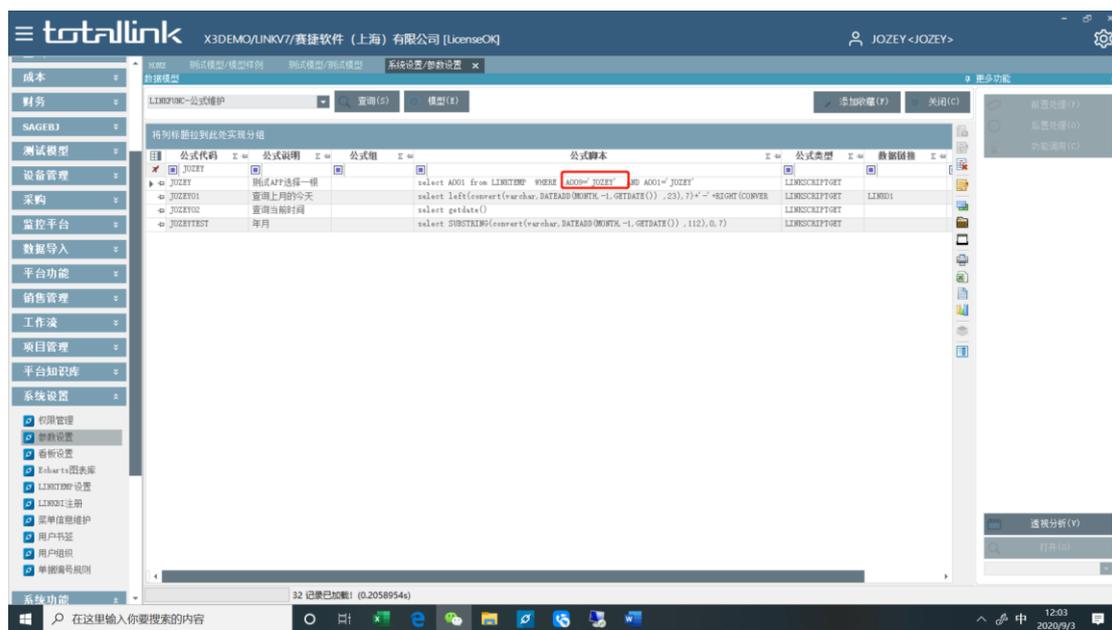
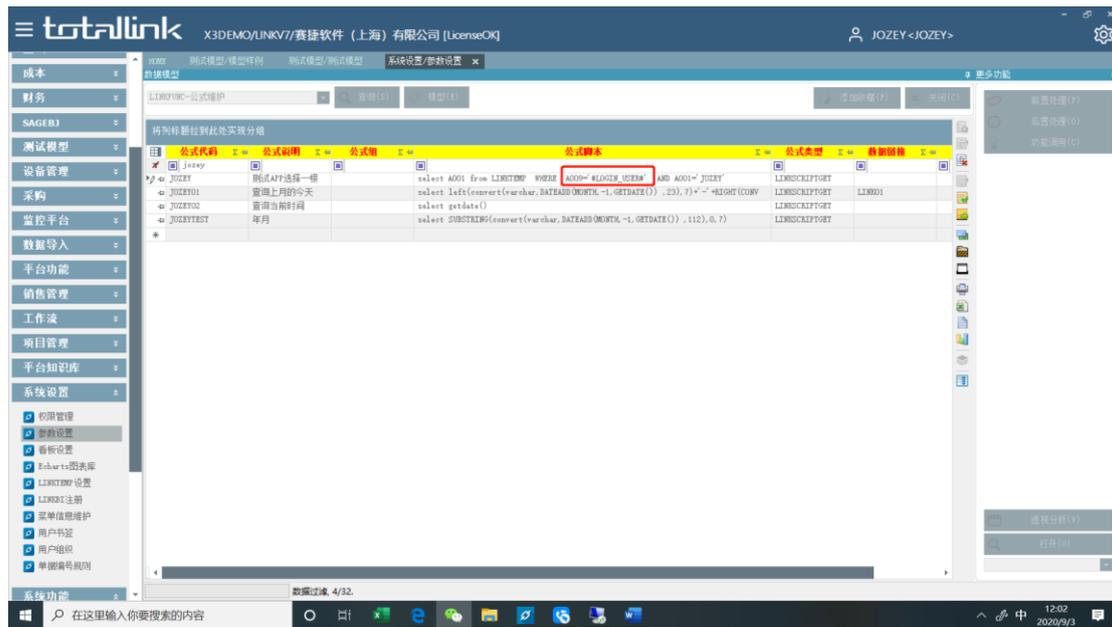
在维护系统模型时，涉及到全局变量、自定义变量、参数时，在模型的添加编辑代码上加 “--LINKKEEPORIGINAL”，可以解决添加编辑时，“全局变量、自定义变量、参数” 代码

被系统自动调换成值的问题。

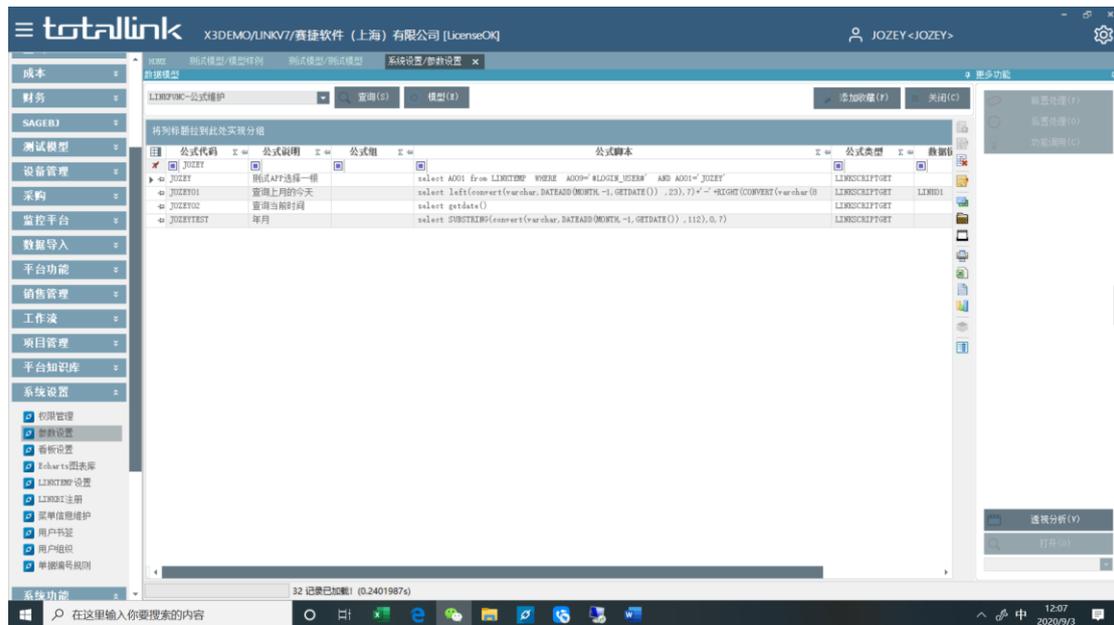
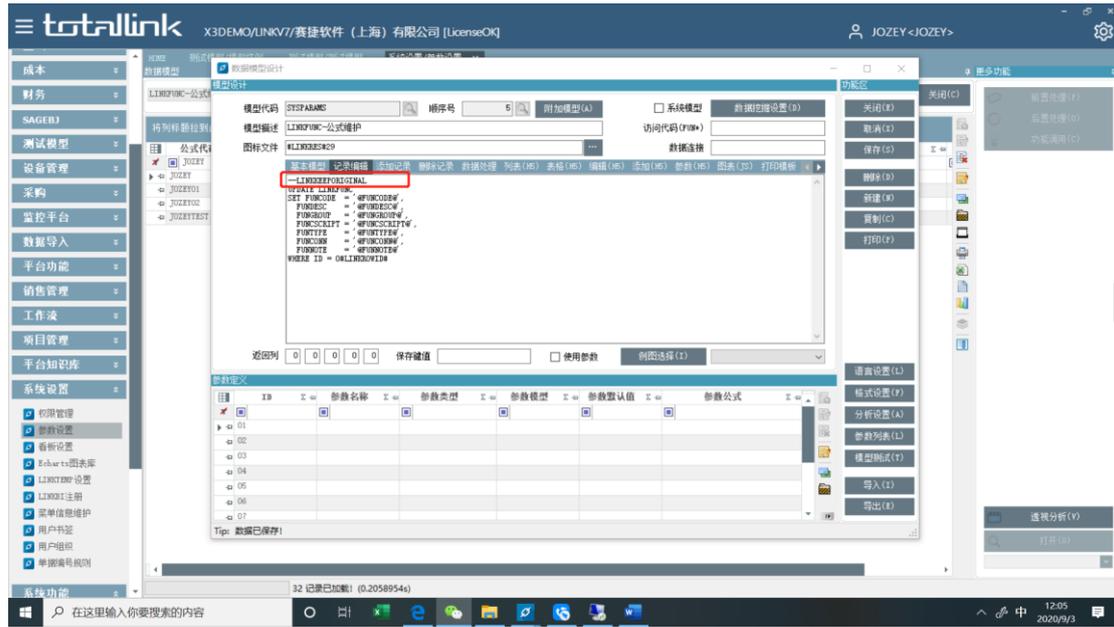


- 如下图，在编辑的代码未加上面的关键字，在保存的时候代码被系统替换成了当前用户



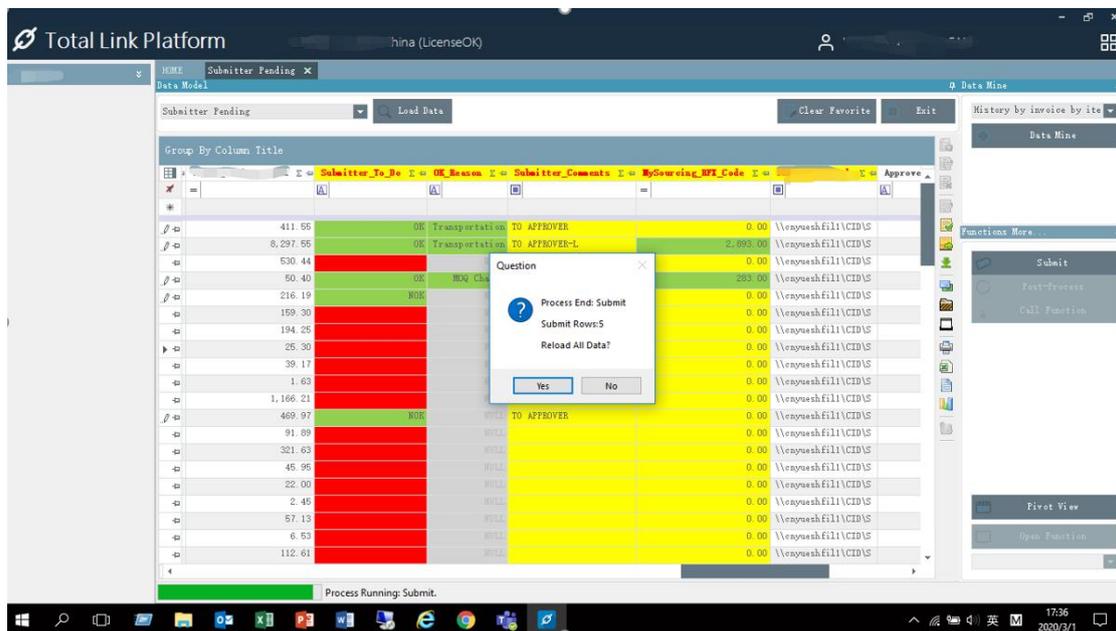
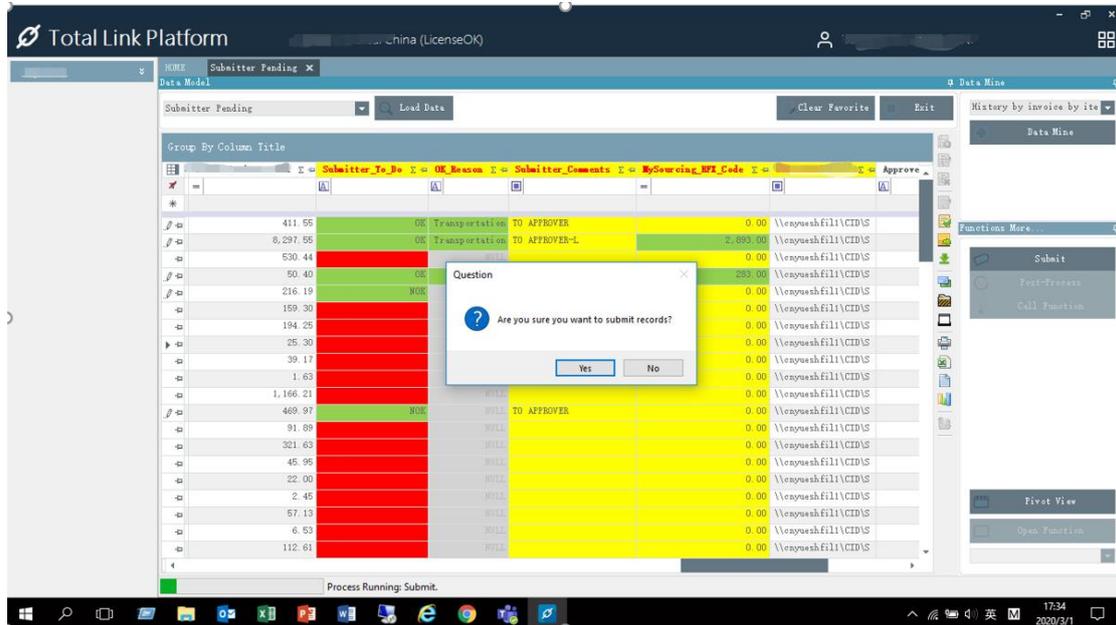


- 在编辑模块加入以上代码，保存的是“#LOGIN_USER#”



2.11 按钮类附加模型+2000 通知模型

- 效果如下：点击提交按钮，系统执行完后返回提交的行数



功能介绍：

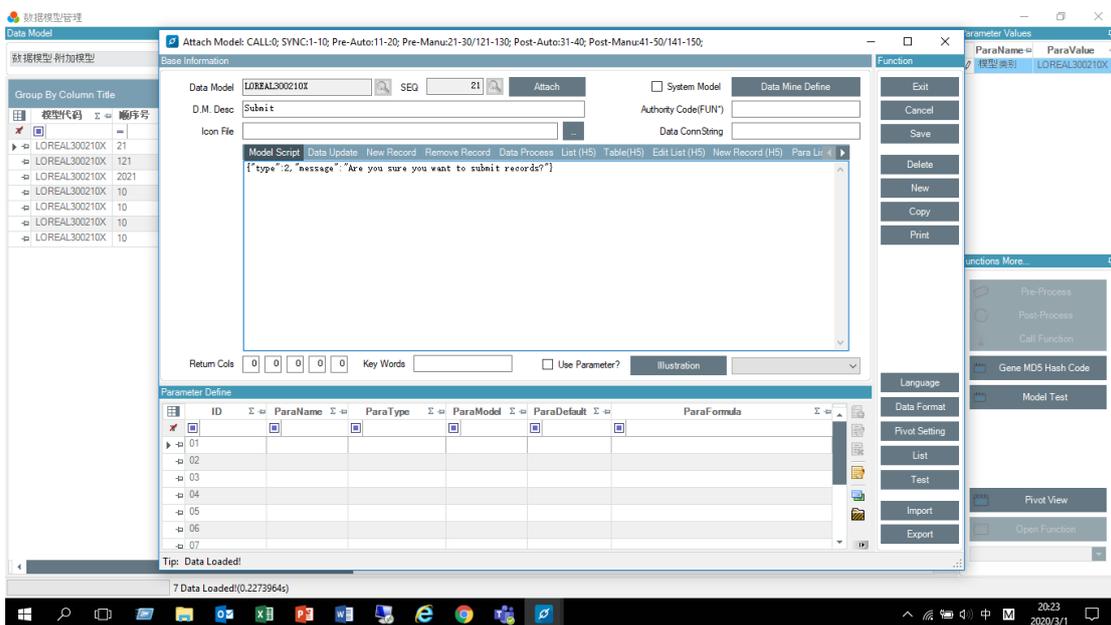
附加模型编号+2000 的模型可以用来做信息提示功能。

附加模型为 21、41、501 类型的附加模型，对应的提示模型为 2021、2041、2501。

当附加模型为以下的编号时，对应的提示模型如下：

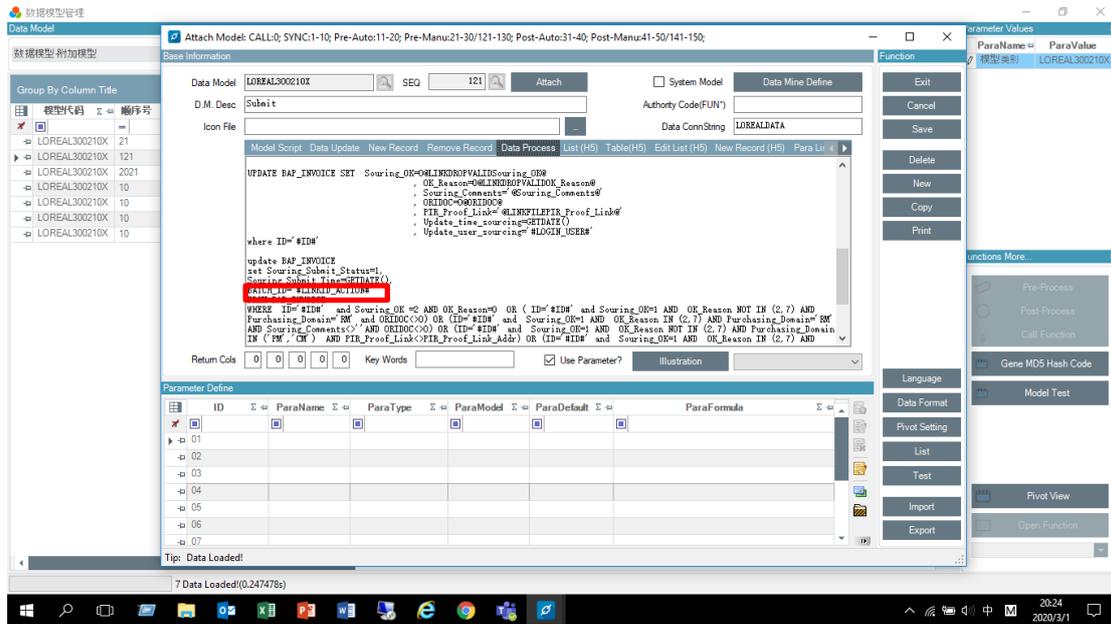
编号	提示模型
121-130	2021
502	2502
22	2021
42-50	2041

- 21 模型，人为判断动作模型，询问是否要执行后面的动作，是则继续，否则不执行

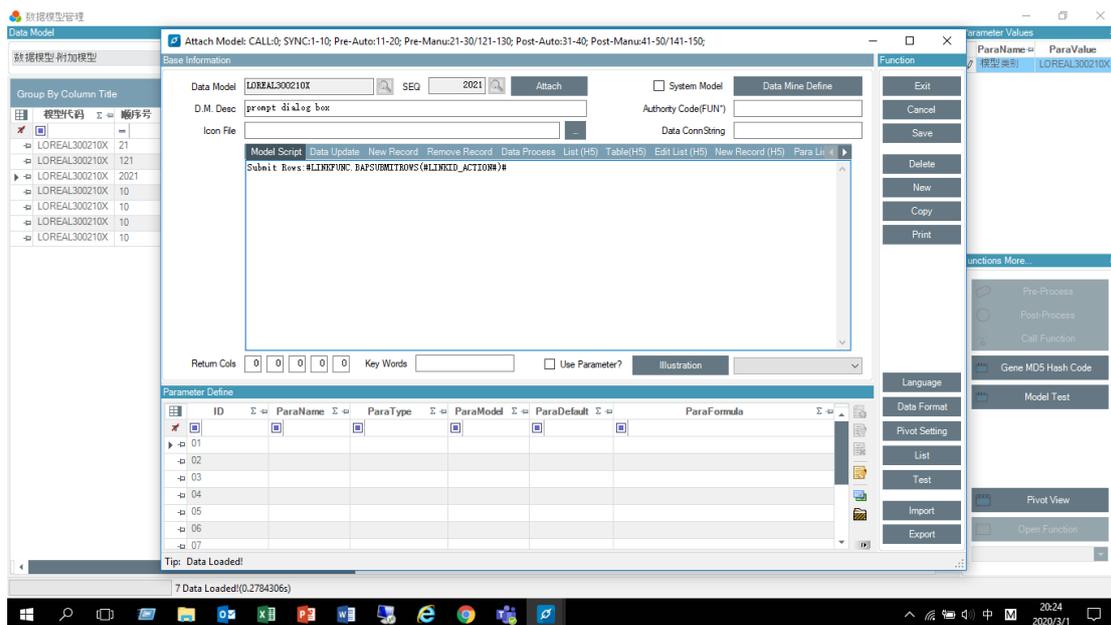


{"type":2,"message":"Are you sure you want to submit records?"}

- 121 模型，用#LINKID_ACTION#标记提交的记录



● 附加 2021 基本模型代码返回提交的行数



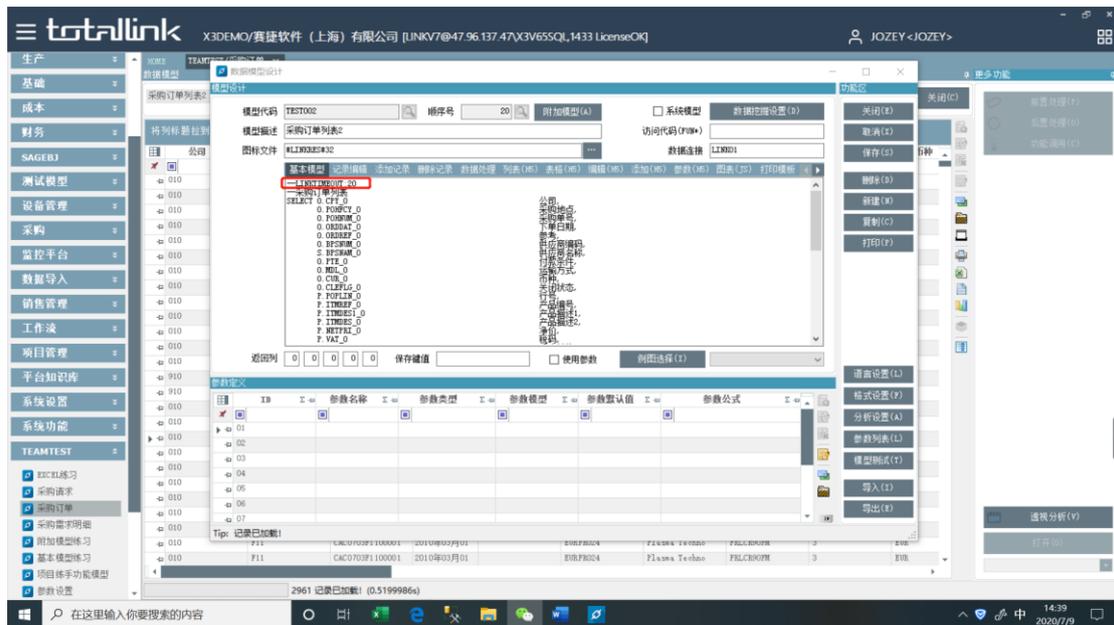
Submit Rows:#LINKFUNC.BAPSUBMITROWS(#LINKID_ACTION)#

// #LINKFUNC.BAPSUBMITROWS#在 LINKFUNC 参数列表的定义为:

select COUNT(BATCH_ID) from BAP_INVOICE WHERE BATCH_ID='{0}'

2.12 Timeout 设置--LINKTIMEOUT:秒数

系统模型内的 sql 处理默认 timeout 时间设置为 10 秒,执行系统外部的语句默认 timeout 时间是 5 分钟。可通过再在 SQL 代码最上方添加 “—LINKTIMEOUT:秒数”,自定义 timeout 的时间

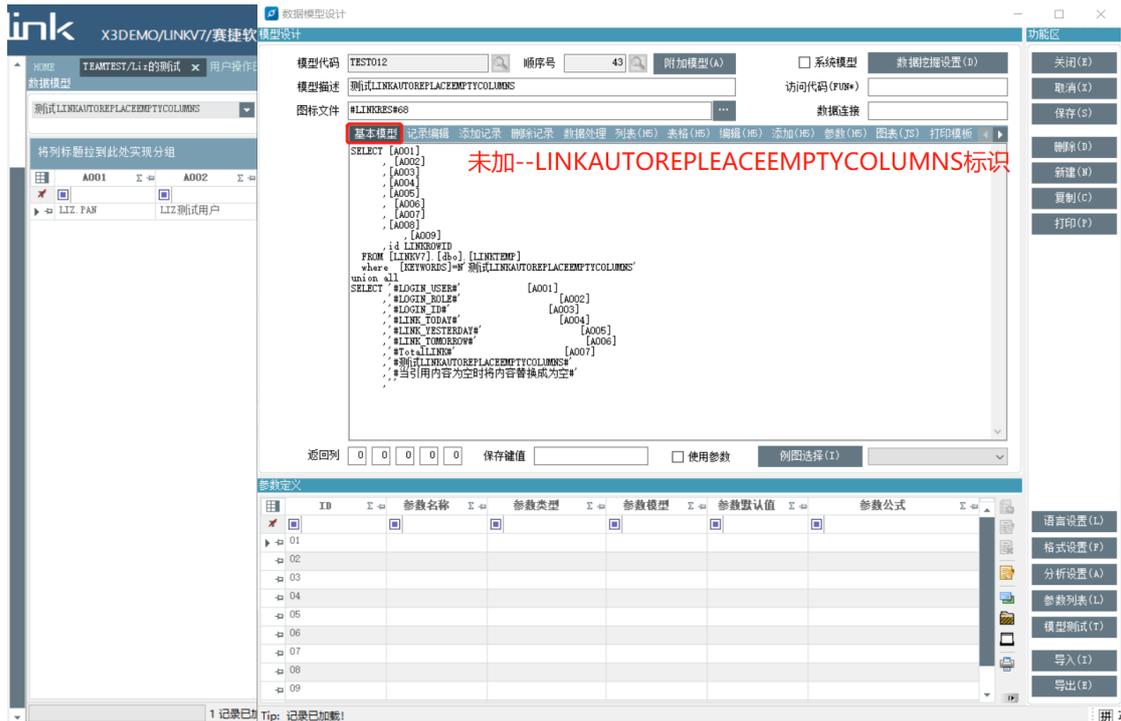


2.13 将#内容#替换为空—

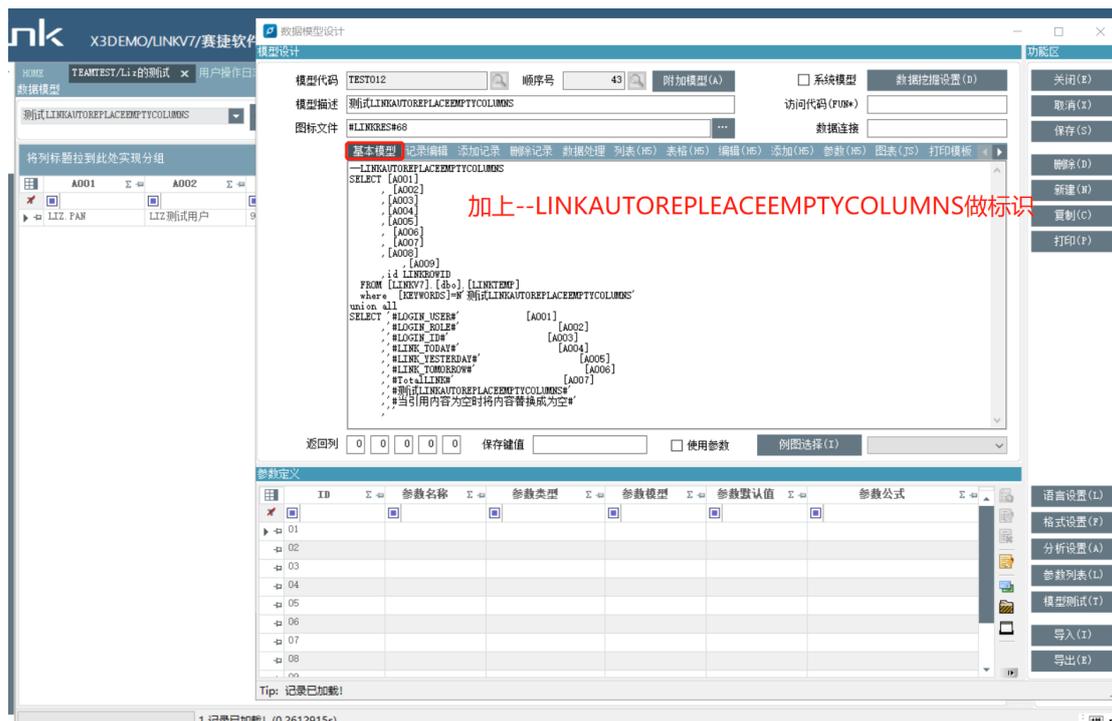
LINKAUTOREPLACEEMPTYCOLUMNS

增加—LINKAUTOREPLACEEMPTYCOLUMNS标识则##里面的内容会被替换为空显示和保存,不增加此标识则以#内容#显示保存;根据实际情况,在对应的地方加上该标识,例如,在记录编辑脚本中使用##则在记录编辑页签中加上该标识,系统替换脚本带有##不起作用,具体使用效果如下。

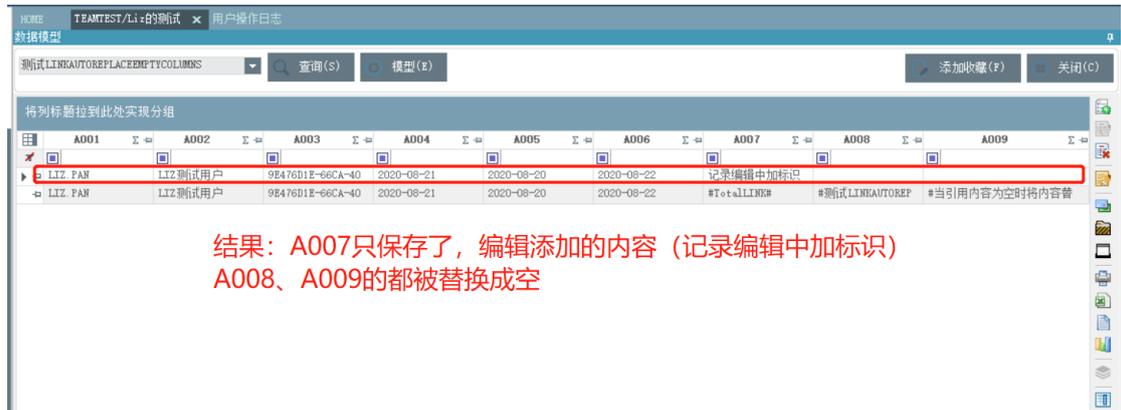
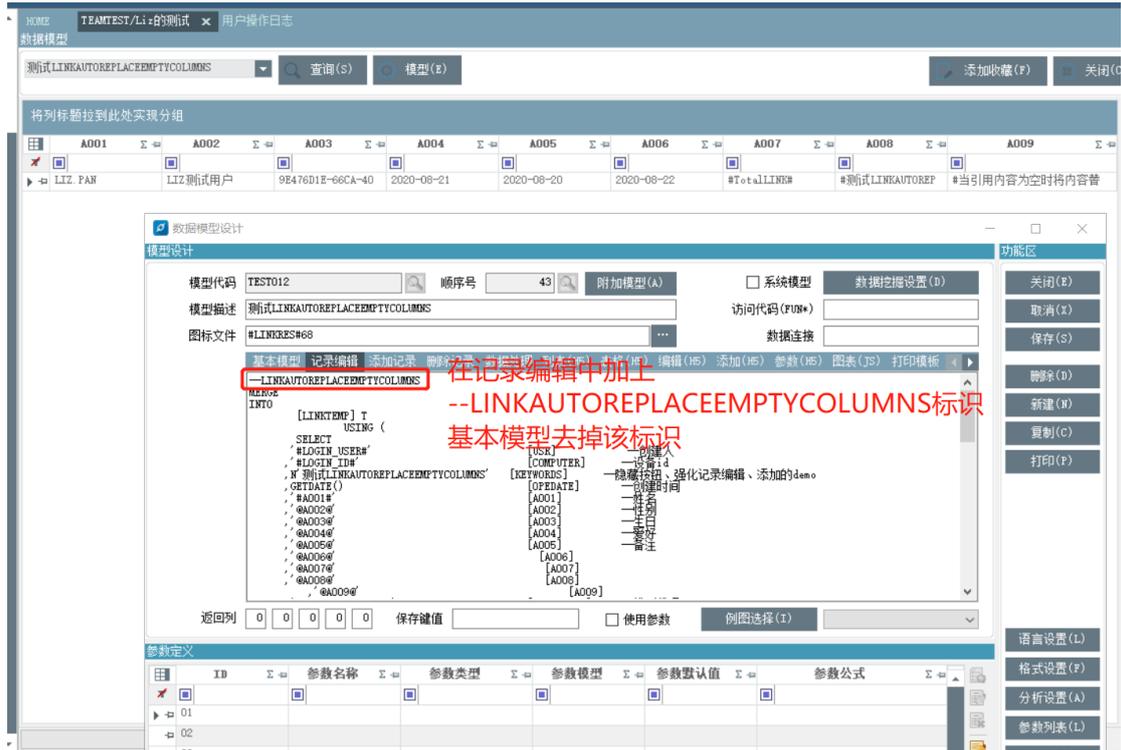
- 未添加—LINKAUTOREPLACEEMPTYCOLUMNS标识



- 基本模型加上--LINKAUTOREPLACEEMPTYCOLUMNS标识



- 在记录编辑中添加--LINKAUTOREPLACEEMPTYCOLUMNS标识



2.14 用户数据权限—LINKMYROWDATA

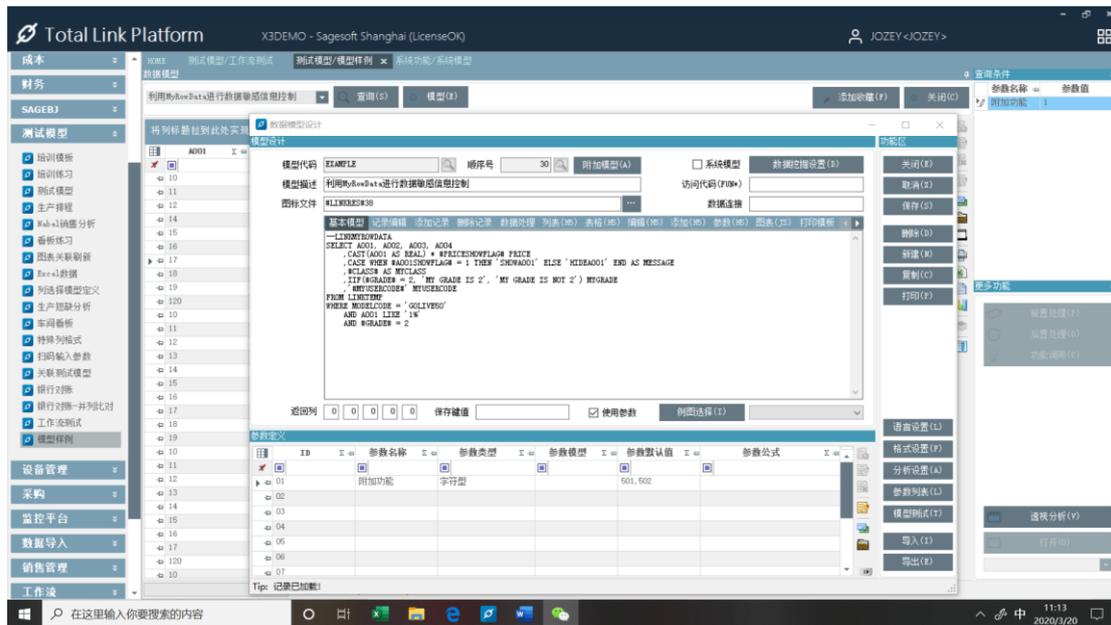
附加模型 301，用于在基本模型的查询及数据处理执行前，获取预置的数据，控制敏感信息。使用方式有两种：

- 基本模型背后的附加模型 301（优先），对当前基本模型起作用

- 全局系统模型_LINKSYS_#301，对系统中所有基本模型起作用
- 对主模型的数据处理脚本，该模式有效，可以对脚本进行动态改造
- 对于动作执行的模型，比如 21、41、501 等，也可以用这种模型对执行脚本进行动态改造

2.14.1 控制用户看到的数据

- 在基本模型写好相关代码，代码前面加 **--LINKMYROWDATA** 关键字



- 基本模型代码参考：

```
--LINKMYROWDATA
SELECT A001, A002, A003, A004 ,
CAST(A001 AS REAL) * #PRICESHOWFLAG# PRICE ,
CASE WHEN #A001SHOWFLAG# = 1 THEN 'SHOWA001' ELSE 'HIDEA001' END AS MESSAGE ,
#CLASS# AS MYCLASS ,
IIF(#GRADE# = 2, 'MY GRADE IS 2', 'MY GRADE IS NOT 2') MYGRADE ,
'#MYUSERCODE#' MYUSERCODE
FROM LINKTEMP WHERE MODELCODE = 'GOLIVE50' AND A001 LIKE '1%' AND #GRADE# = 2
```

- 在 301 模型定义 **PRICESHOWFLAG**、**A001SHOWFLAG**、**CLASS**、**GRADE** 字段（可以用基本模型背后的附加 301，也可以用系统模型_LINKSYS_#301）



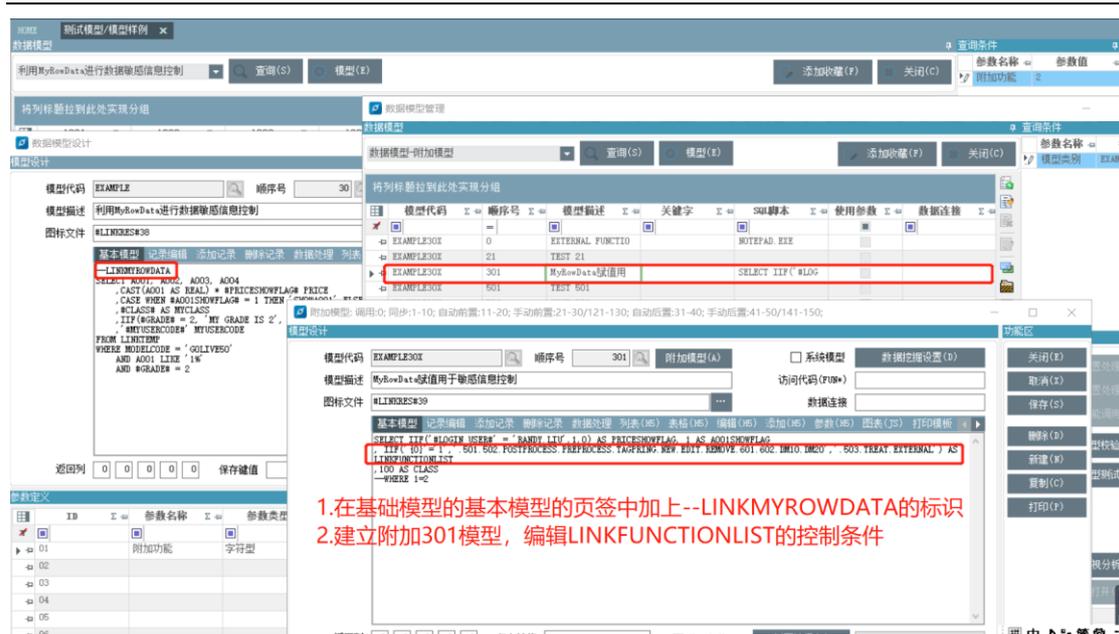
附加模型 301 代码参考:

```
SELECT
IIF('#LOGIN_USER#' = 'RANDY.LIU', 1, 0) AS PRICESHOWFLAG,
1 AS A001SHOWFLAG,
100 AS CLASS
```

分析: 通过附加模型 301 对 PRICESHOWFLAG、A001SHOWFLAG、CLASS 字段赋值, 在执行查询语句时, 系统根据语句算出这几个字段的值, 影响查询模型的结果

2.14.2 根据条件控制按钮的可见性 (LINKFUNCTIONLIST)

--LINKMYROWDATA 和系统变量 LINKFUNCTIONLIST 结合使用, 可控制按钮的可见性(APP 与客户端作用), 具体操作如下:



参考代码

```
SELECT IIF('#LOGIN_USER#' = 'RANDY.LIU',1,0) AS PRICESHOWFLAG, 1 AS A001SHOWFLAG,
IIF('{0}'='1','.501.502.POSTPROCESS.PREPROCESS.TAGPRINT.NEW.EDIT.REMOVE.601.602.D
M10.DM20','.503.TREAT.EXTERNAL') AS LINKFUNCTIONLIST
,100 AS CLASS
```

控制按钮的方式都是以.+指定名来规范,详情请参考如下:

按钮名称	LINKFUNCTIONLIST 引用
附加 501-520	.501-.520
附件 601-620	.601-.620
后处理模型	.POSTPROCESS
预处理模型	.PREPROCESS
标签打印	.TAGPRINT
添加	.NEW
编辑	.EDIT
删除	.REMOVE
数据处理	.TREAT
功能调用	.EXTERNAL
挖掘模型	.DM10.DM20.DM30

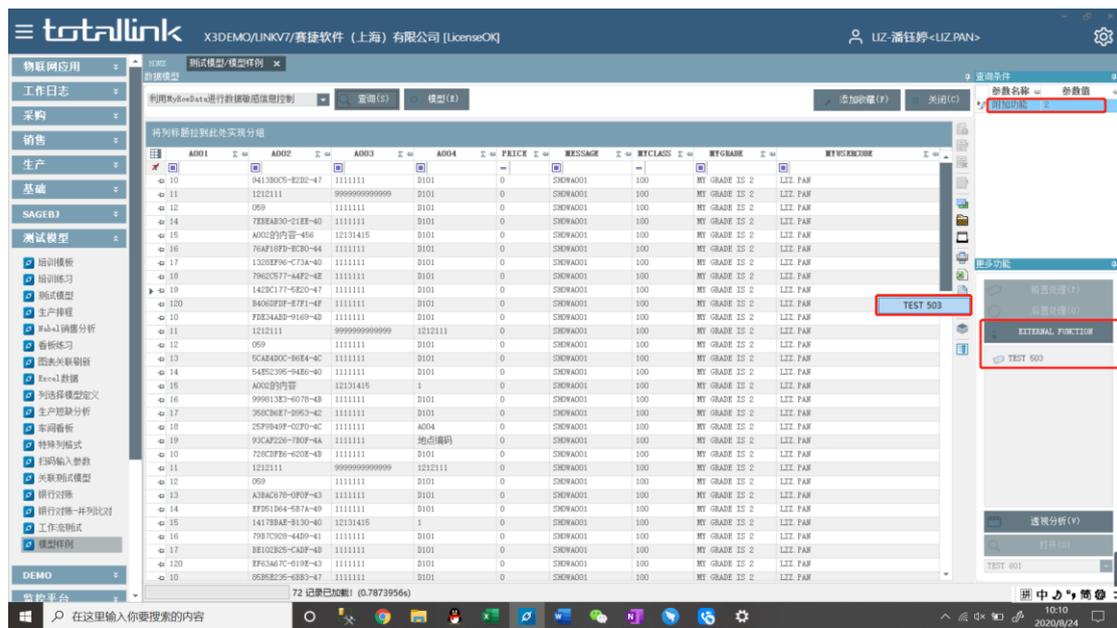
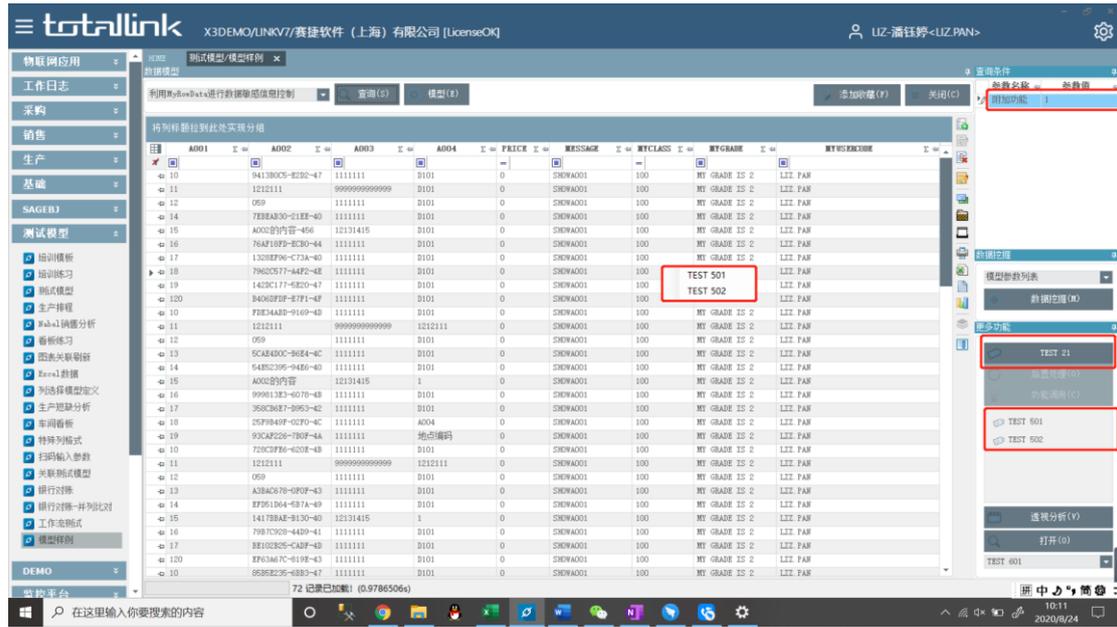
注意:

下面两个仅支持客户端

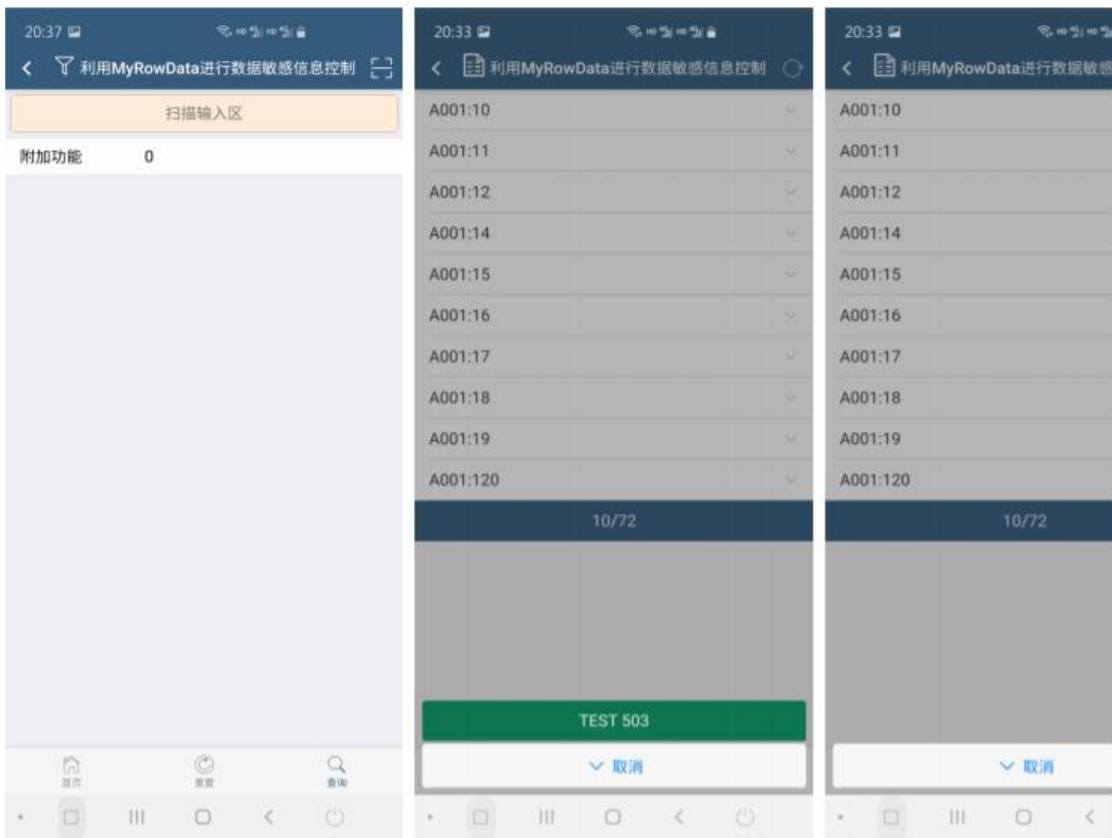
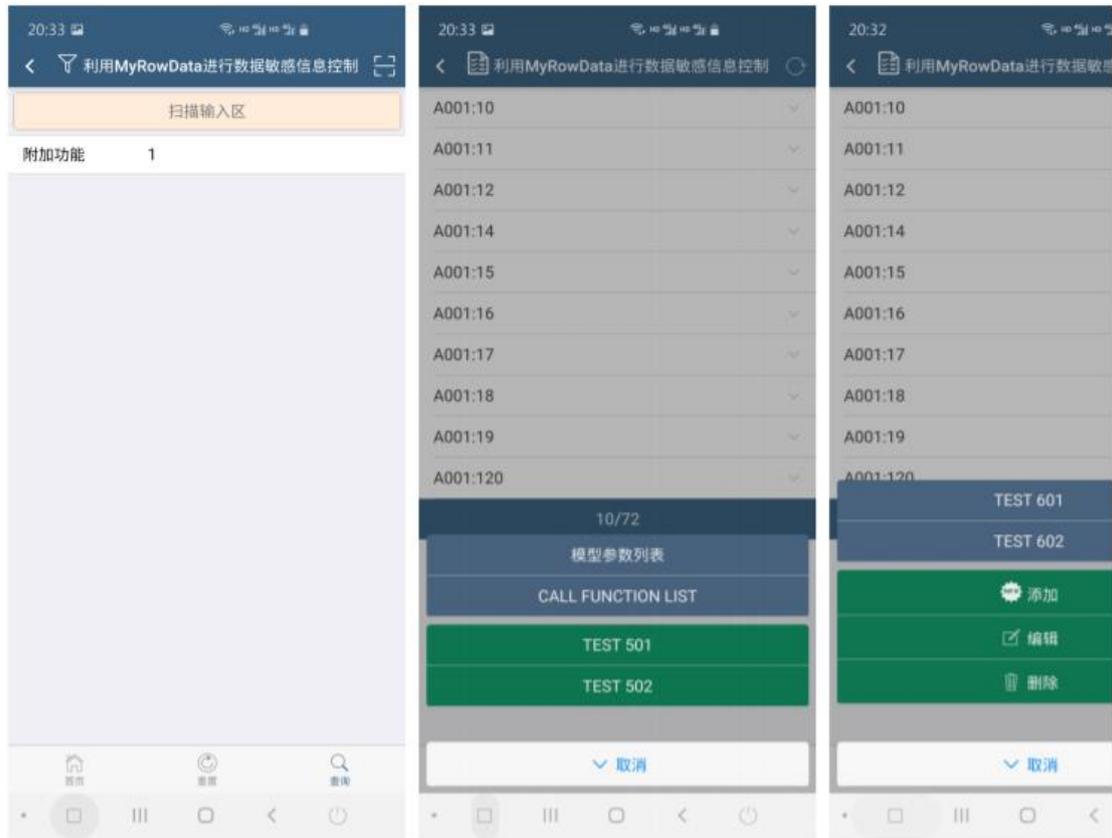
.TREAT—数据处理，.EXTERNAL—功能调用；

客户端 8.1.4.4 的版本附加 501-520 的按钮可在界面中显示。

客户端效果:



APP 效果:

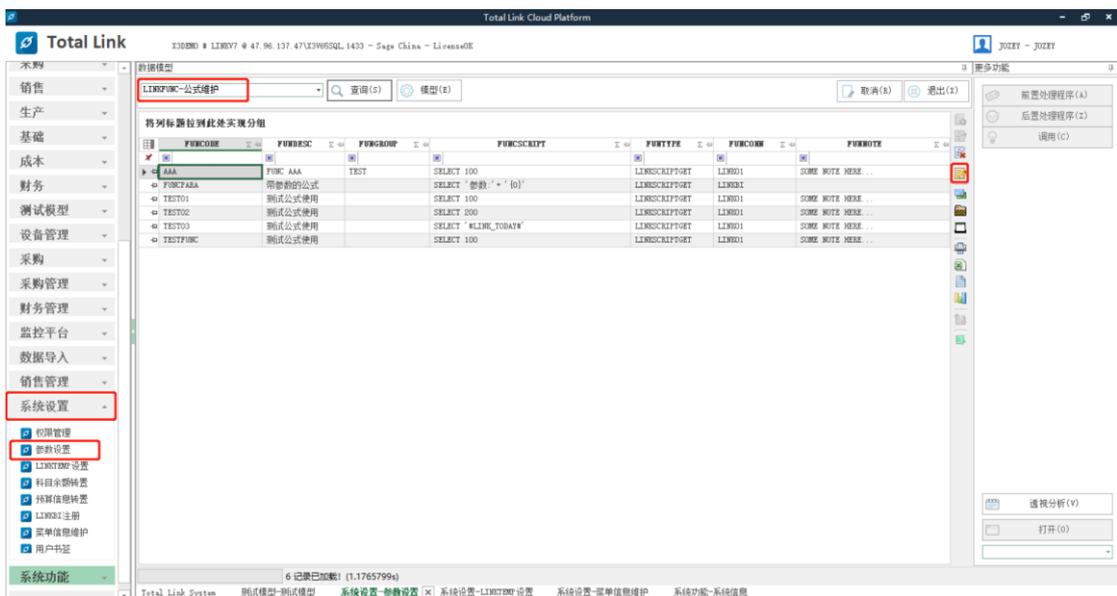


2.15 LINKFUNC 公式使用介绍

在一套业务系统中，往往某个数据需要在多个模型中用到，或者该数据处理过程比较复杂，可以通过 LINKFUNC 方法将常用的数据存到相应的关键字中，并对关键字进行处理，然后通过“LINKFUNC.FUNCODE”这种方式调用。

- 首先在系统中定义 FUNCODE，维护 FUNCODE 的内容，才能在其它模型中应用。

在菜单“系统设置—参数设置—LINKFUNC 公式维护”下点击编辑，将 FUNCODE 维护进系统



字段介绍:

FUNCODE:自定义功能代码

FUNDESC:描述

FUNGROUP:分组

FUNCSCRIPT:脚本代码，可以是 SQL 语句，也可以是公式

FUNCTYPE:公式类型，包括 LINKSCRIPTGET、LINKFORMULA、LINKSCRIPTRUN 等

FUNCCON:连接代码，需要用的模型来自哪个数据库，一般为“LINK01”（可通过系统设置查看）

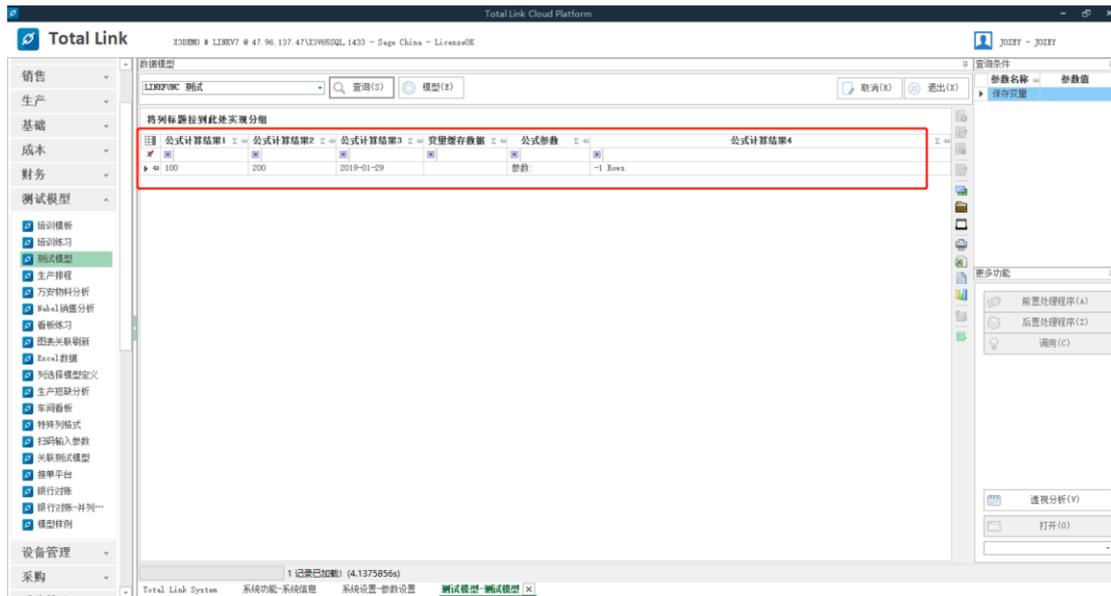
FUNNOTE:备注

- LINKFUNC 的 FUNCODE 有多种公式类型，下面是公式类型的介绍

类型	含义
LINKSCRIPTGET	返回执行脚本的结果
LINKFORMULA	返回公式计算的结果
LINKSCRIPTRUN	返回执行脚本影响的行数
LINKSETVALUE(CODE,Value)	定义变量并赋值，其中“CODE”为自定义的关键字，value 可以为数字、字符或参数等
LINKGETVALUE(CODE)	获取变量的值，和 LINKSETVALUE(CODE,Value)配套使用
FILEDIALOG	选择单个文件
GETFILELIST	选择多个文件

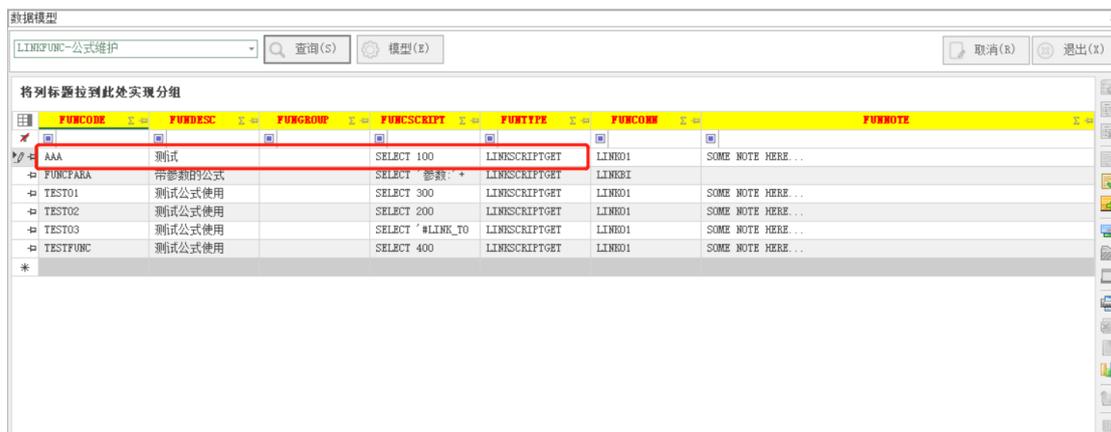
- 模型调用 FUNCODE

- 查询结果

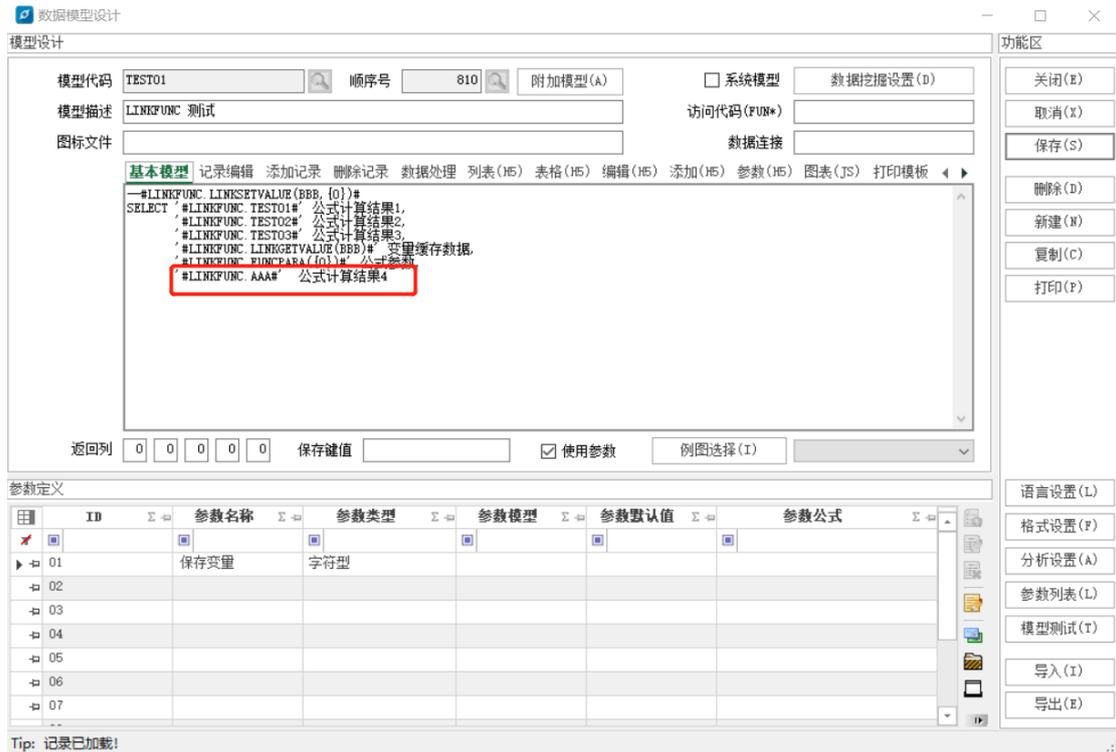


2.15.1 应用——LINKSCRIPTGET

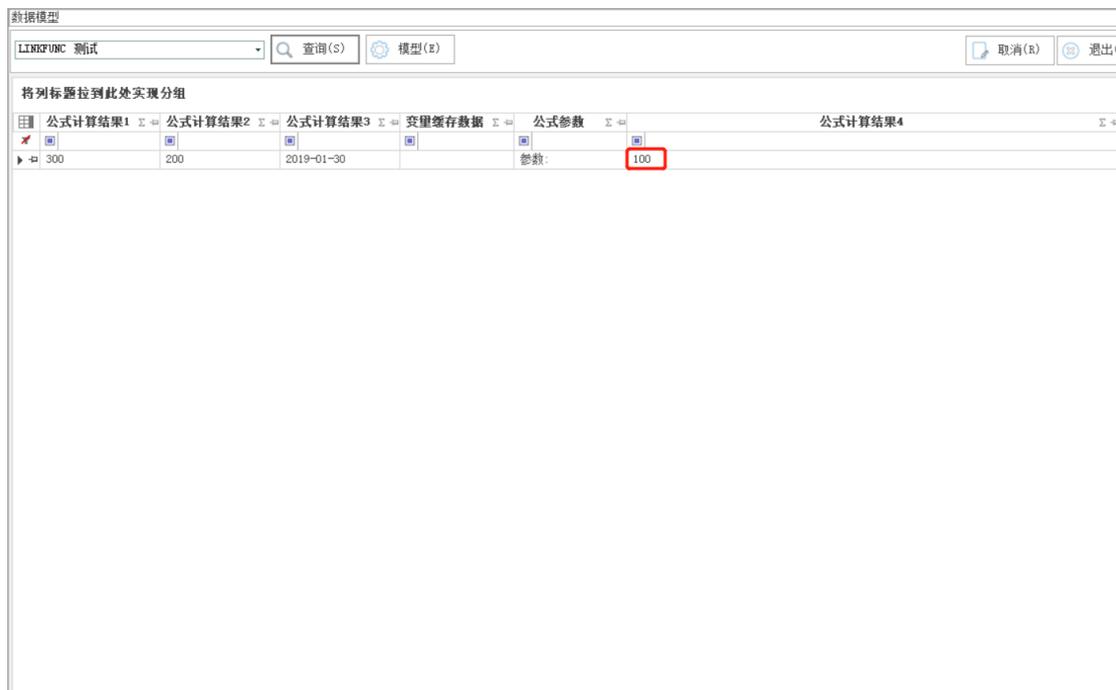
- 此小节以 LINKSCRIPTGET 类型举例，除了 LINKSETVALUE(CODE,Value) 和 LINKSETVALUE(CODE)这两个特殊的类型，其余上述提到的类型用法和 LINKSCRIPTGET 相同，包括：LINKFORMULA、LINKSCRIPTRUN 等
- 按照下图自定义一个 AAA FUNCODE，类型为“LINKSCRIPTGET”，脚本内容为：“SELECT 100”，点击保存
- 含义：给该 AAA FUNCODE 设置值为 100



- 在查询模型中输入：SELECT '#LINKFUNC.AAA#'



- 保存后点击查询，查询结果为 100

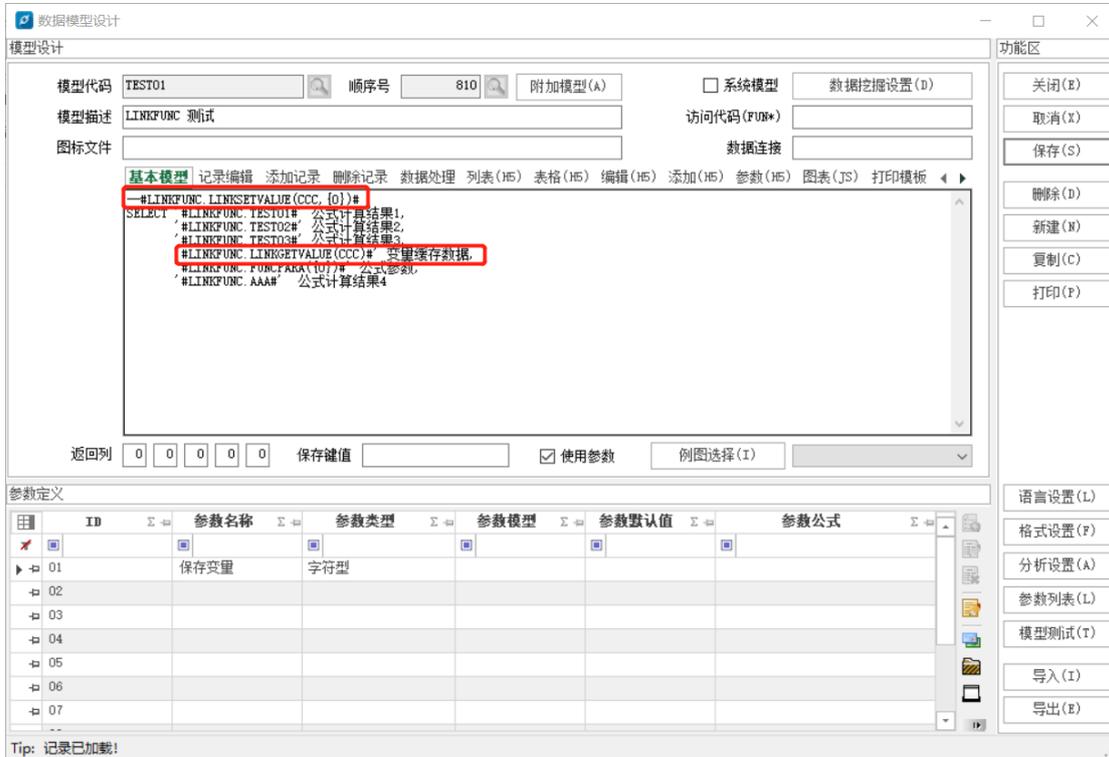


2.15.2 应用——LINKSETVALUE/LINKGETVALUE

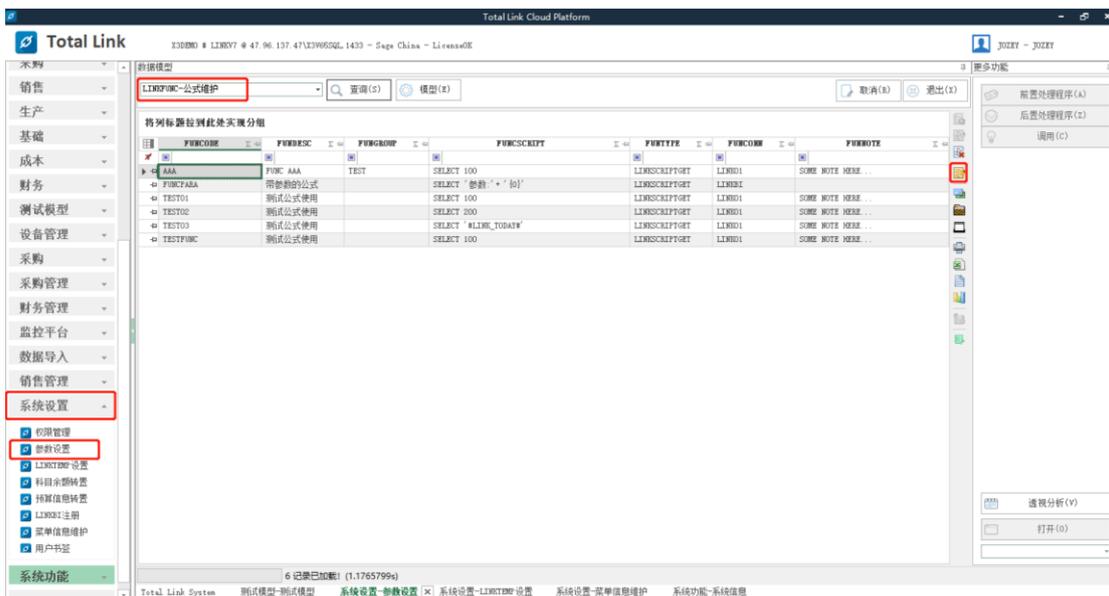
- LINKSETVALUE 是比较特殊的一个公式类型,它可以直接在模型中定义 FUNCODE 并赋值,然后查询出来, LINKSETVALUE(CODE,Value)和 LINKGETVALUE(CODE)搭配使用

- 本例中代码:

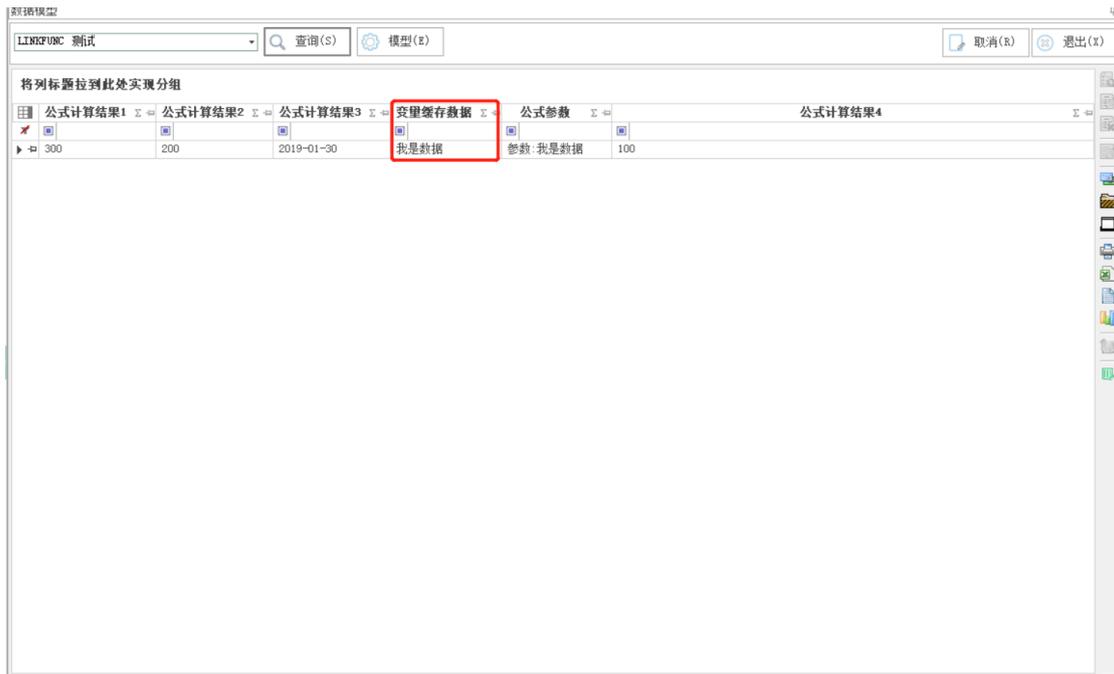
```
--#LINKFUNC.LINKSETVALUE(CCC,{0})# //定义 FUNCODE, 并将参数{0}的值传到 CCC 这个
FUNCODE 中
SELECT #LINKFUNC.LINKGETVALUE(CCC)# //获取 FUNCODE CCC 的值
```



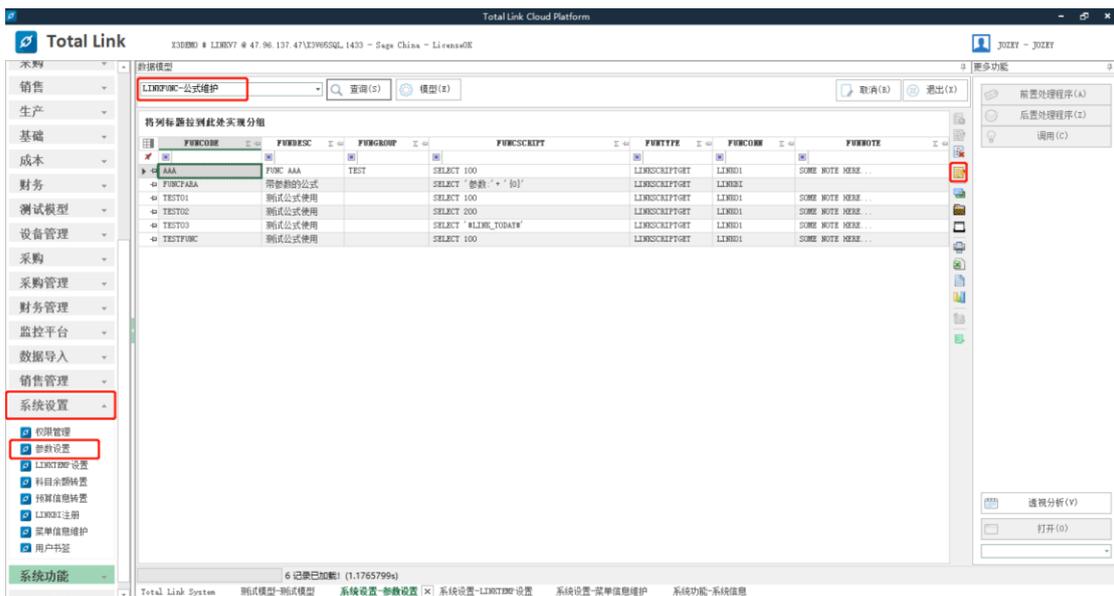
- 在参数设置添加该关键字



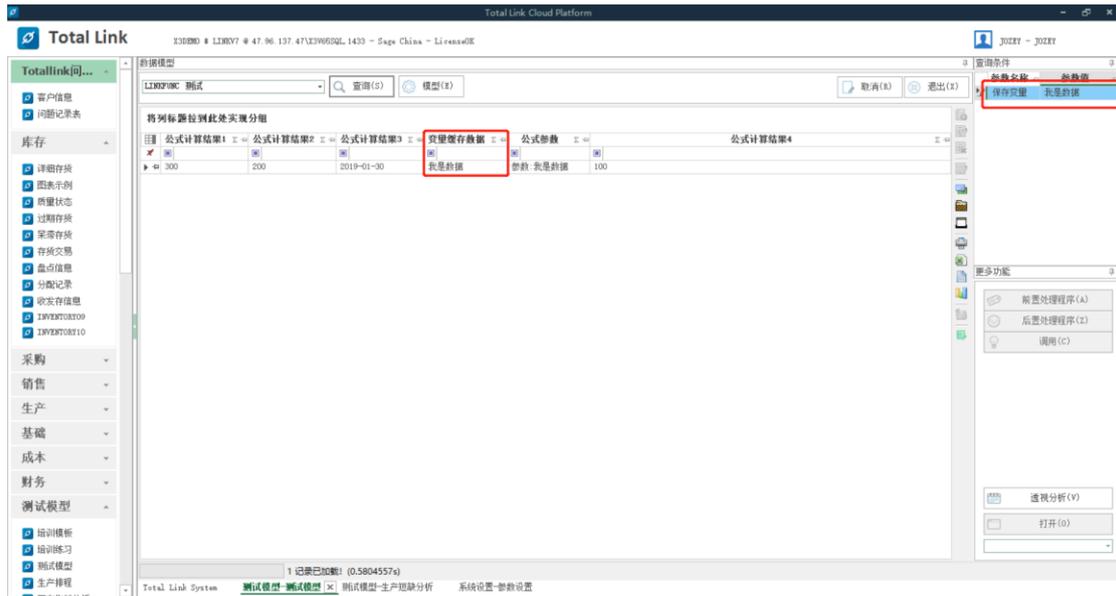
- 在右侧框输入参数值“我是数据”，点击查询，结果为“我是数据”



- 在参数设置添加该关键字



- 在右侧框输入参数值“我是数据”，点击查询，结果为“我是数据”



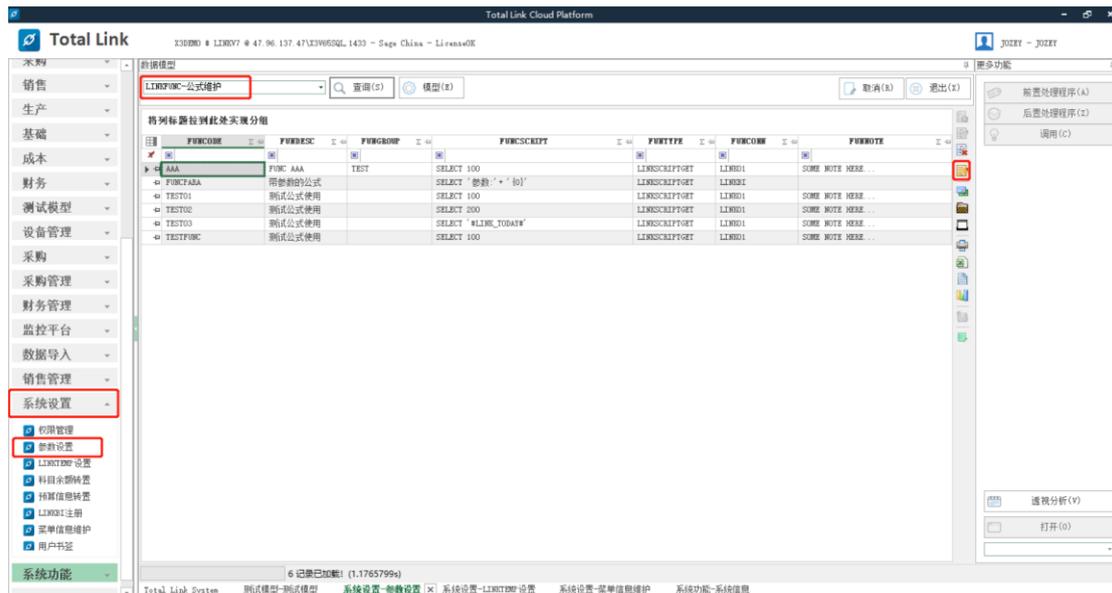
- 在上面的模型定义的 LINKSETVALUE,可以在其它模型调用
- 新建一个查询模型，输入下面的脚本

代码:

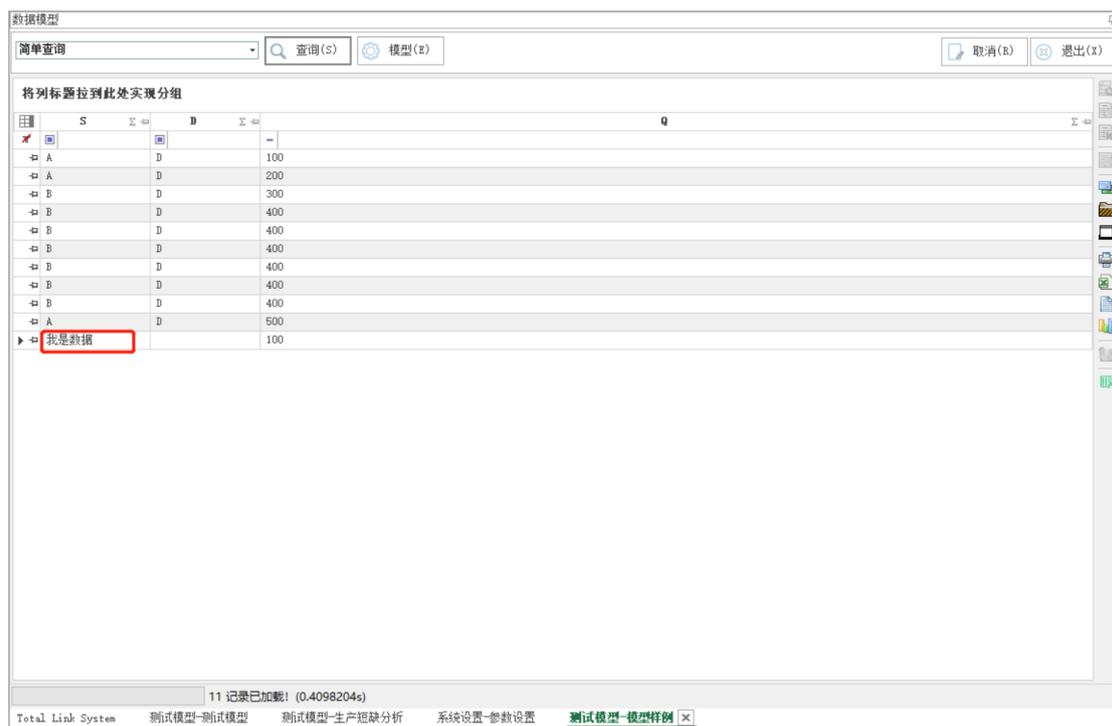
```
SELECT '#LINKFUNC.LINKGETVALUE(CCC) #' //获取 CCC FUNCODE 的值
```



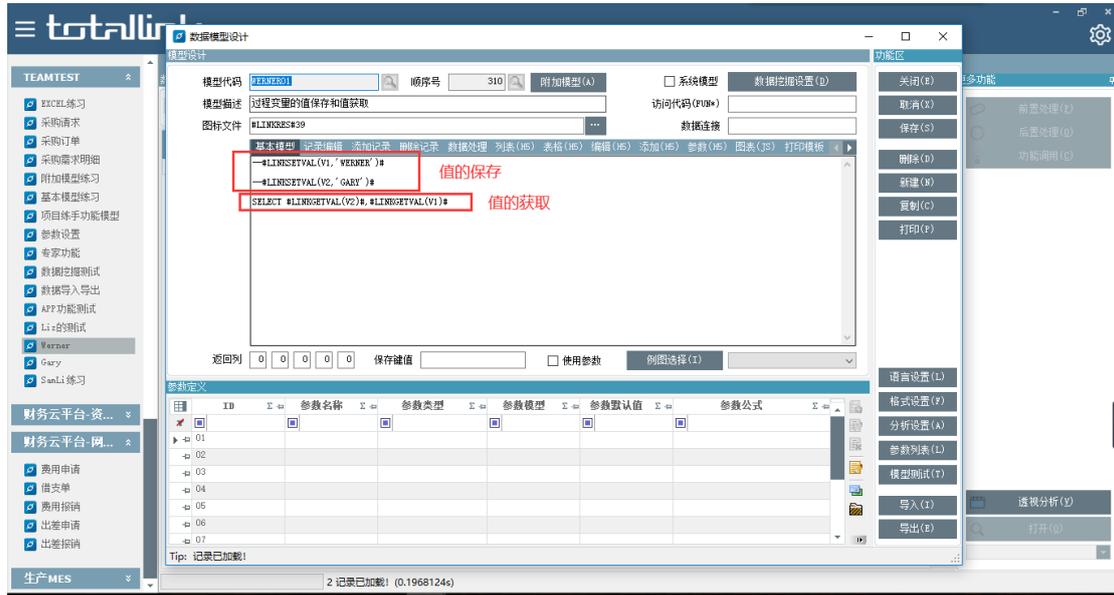
- 在参数设置添加该关键字



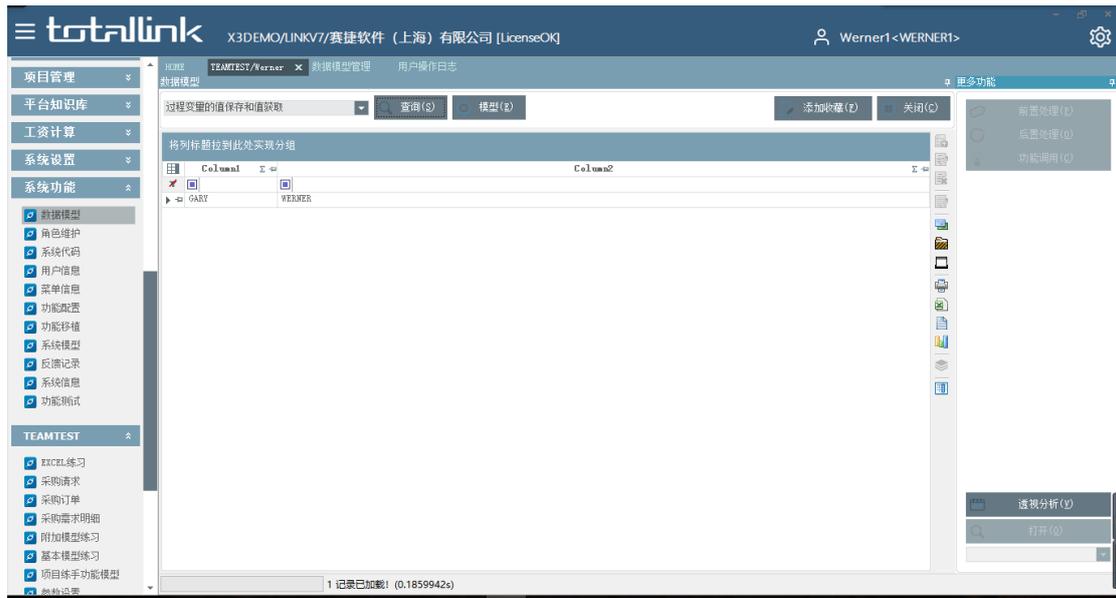
- 点击查询，该模型把上一个模型中的 CCC 内容传了过来，显示“我是数据”



2.15.3 过程变量值的保存和获取——LINKSETVAL/LINKGETVAL



- 可以处理不同用户自定义的变量代换，与 LINKSETVALUE/LINKGETVALUE 的区别是 LINKSETVAL\LINKGETVAL 先预处理“()”里面的内容，并允许允变量值中包含空格,用法与 LINKSETVALUE/LINKGETVALUE 一致
 - LINKSETVAL(V1,XXX) 值的保存
 - LINKGETVAL(V1) 值的获取
- 若客户端一直处于启动状态，则 LINKGETVAL 可以跨模型使用
- 若客户端重启之后，需先转到对应的模型执行 LINKSETVAL 之后，才可以继续使用 LINKGETVAL

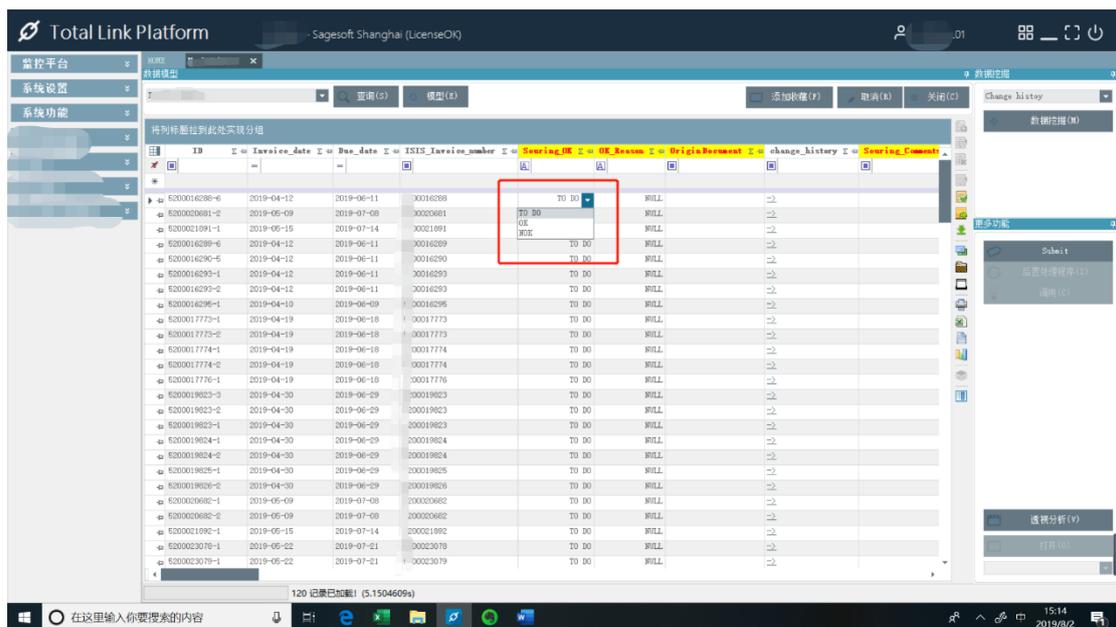


2.16 下拉选择框控制

在设计查询模型时，往往涉及到下拉框的使用。在系统中灵活选用 LINKLISTXXX、LINKDROPDOWNLISTXXX、LINKDROPVALIDDXXX(XXX 为字段名)模式的下拉框，可以简化模型代码，将业务数据灵活的转换到表中。

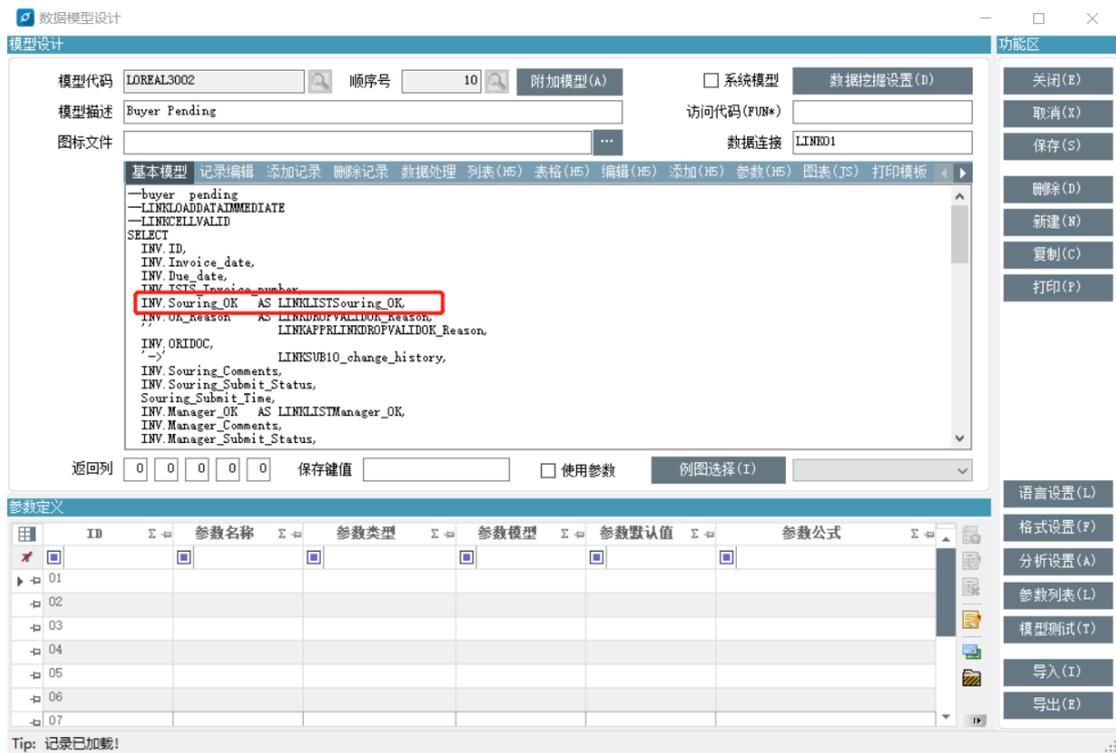
2.16.1 LINKLISTXXX

- LINKLISTXXX 模式的下拉框，适用于下拉列表字段固定且字段比较少的情況

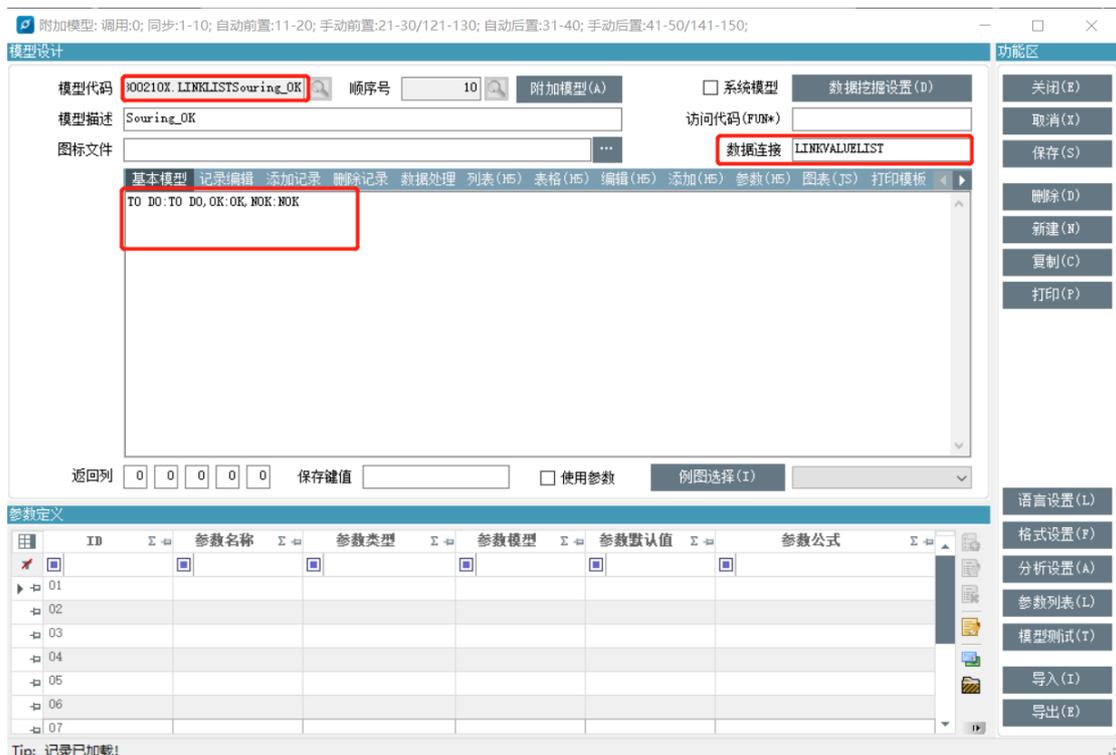


- 本例中，审批字段 Sourcing_OK 有 “TO DO”、“OK”、“NOK” 三种固定状态，下拉框设计如下

- 查询模型给“Sourcing_OK”字段别名为“LINKLISTSourcing_OK”



- 新建附加模型，模型代码后缀为字段的别名（此处为 LINK300210X.LINKLISTSourcing_OK），模型序号为 10（必须为 10 号模型），数据连接为 LINKVALUELIST

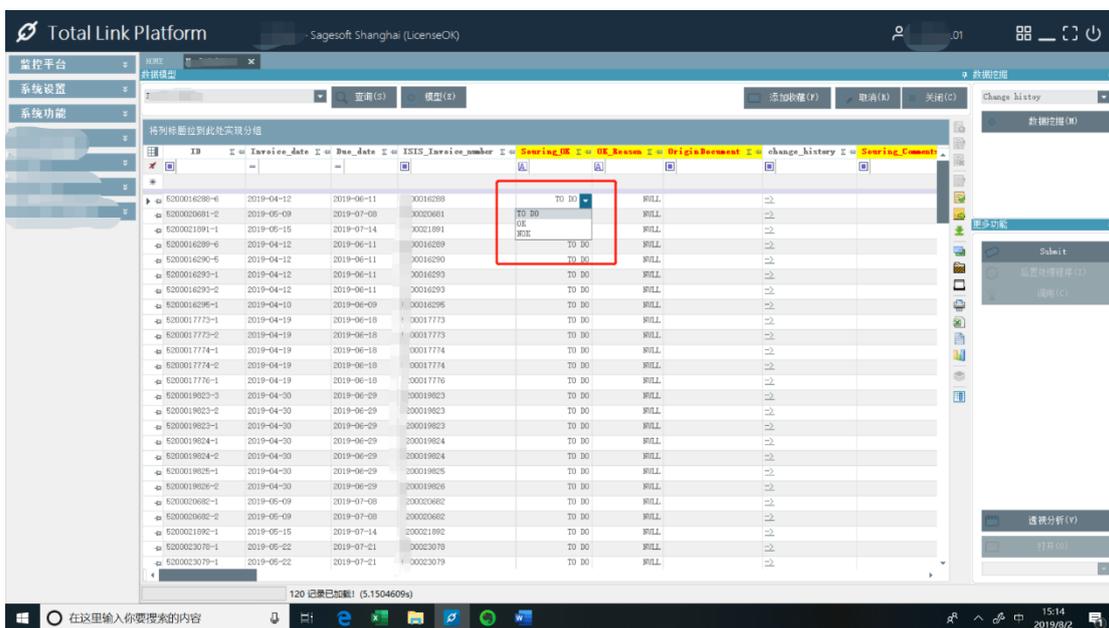


参考代码:

TO DO:TO DO,OK:OK,NOK:NOK

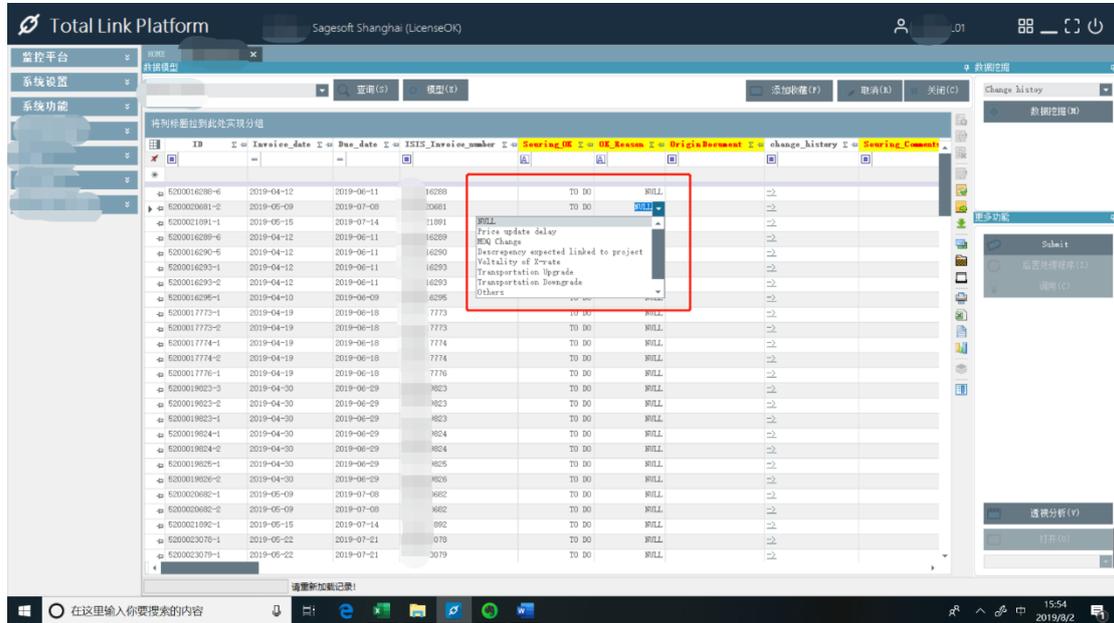
代码解释:

1. 数据以“键:值”的方式定义。冒号左边的值与数据库表的值一致，冒号右边的值为最终显示的值
2. 逗号“,”将下拉框的值分开
 - 实现效果如下，在进行编辑时，选择“OK”,系统传给数据库的值为“OK”，显示的值为“OK”

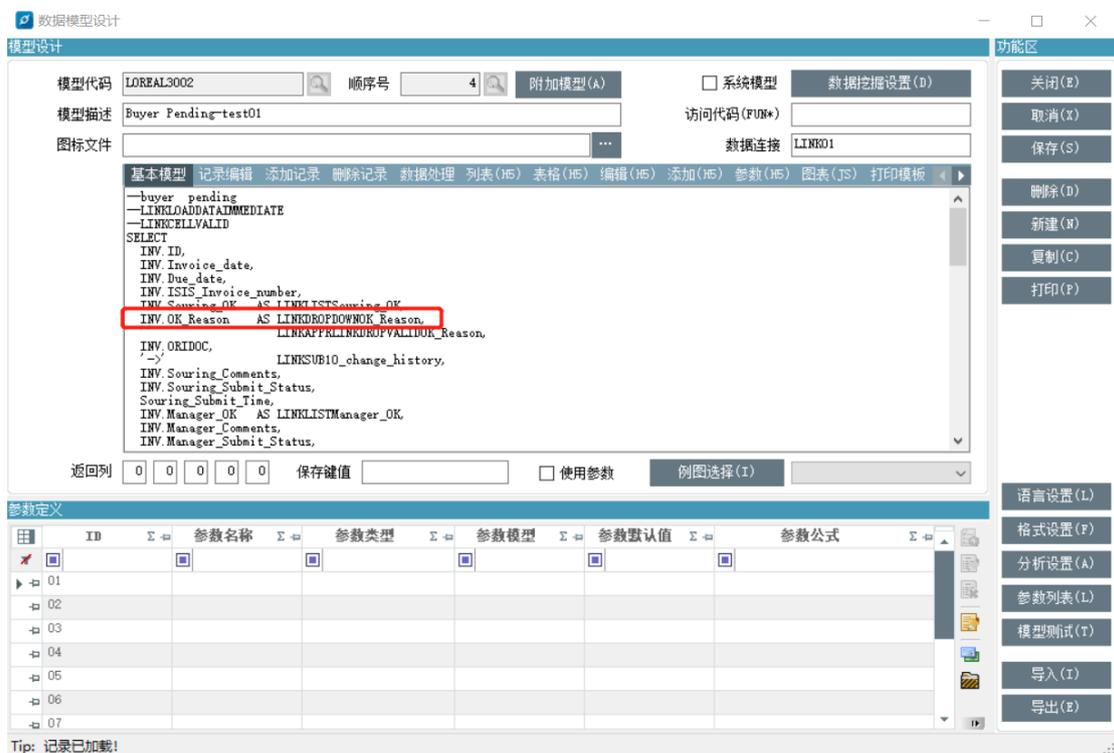


2.16.2 LINKDROPDOWNXXX

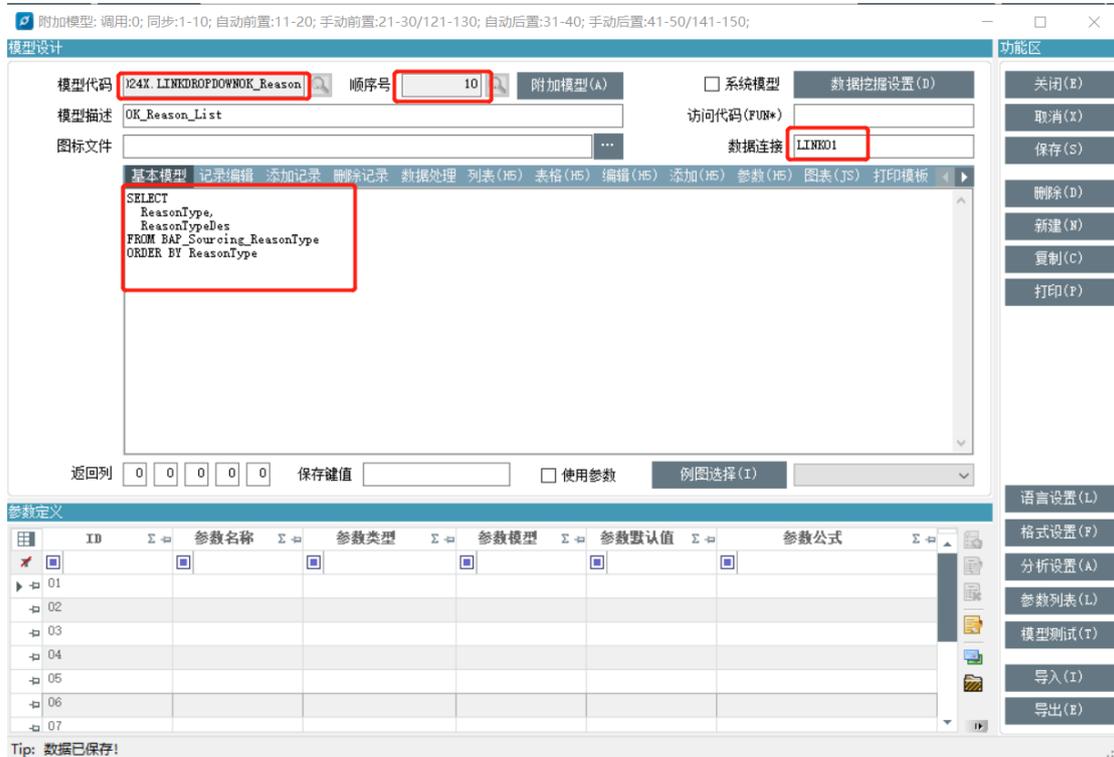
- LINKDROPDOWNXXX 模式的下拉框，适用于下拉列表数据比较多，数据来源于一张数据库表且输入时不需要验证的情况



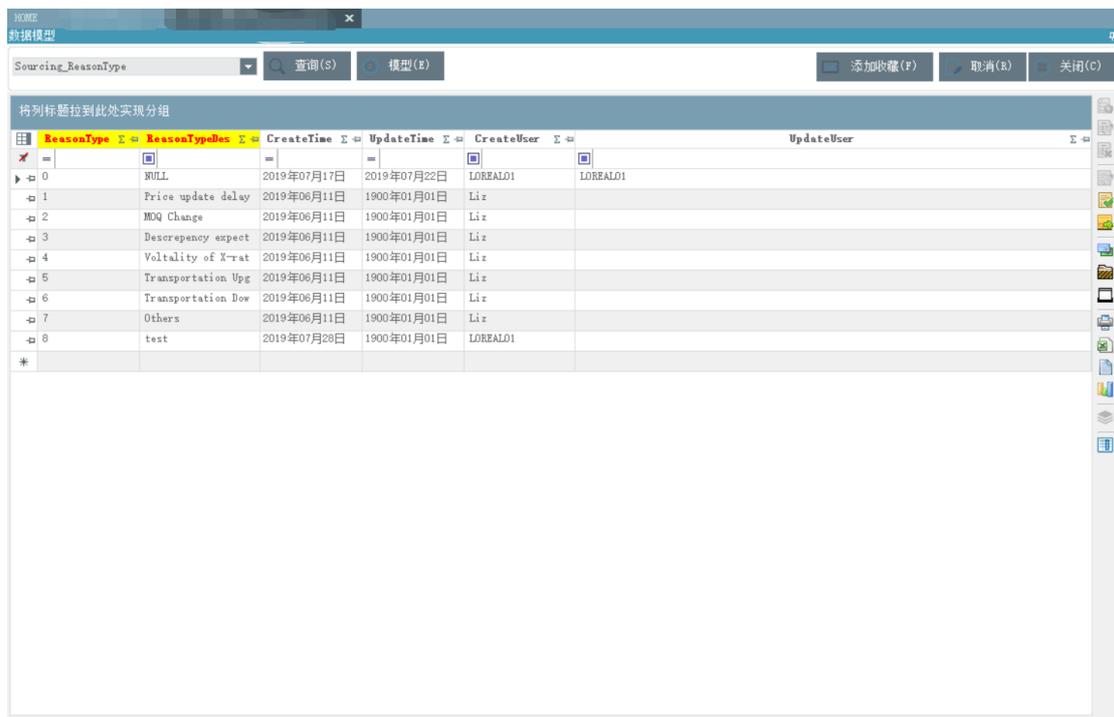
- 本例中，选择审批理由字段“OK_Reason”的值来源于另一张表，下拉框设计如下：
- 查询模型给“OK_Reason”字段别名为“LINKDROPDOWNOK_Reason”



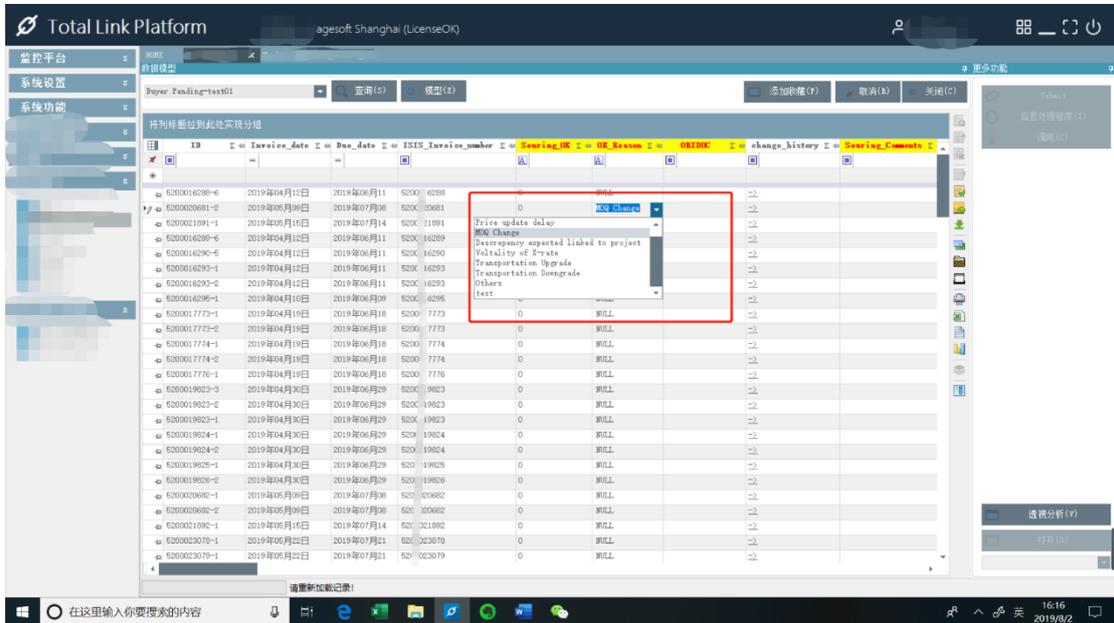
- 新建附加模型，模型代码后缀为字段的别名（此处为 LINK30024X.LINKDROPDOWNOK_Reason），模型序号为 10（必须为 10 号模型），数据连接为查询列表所在的数据源



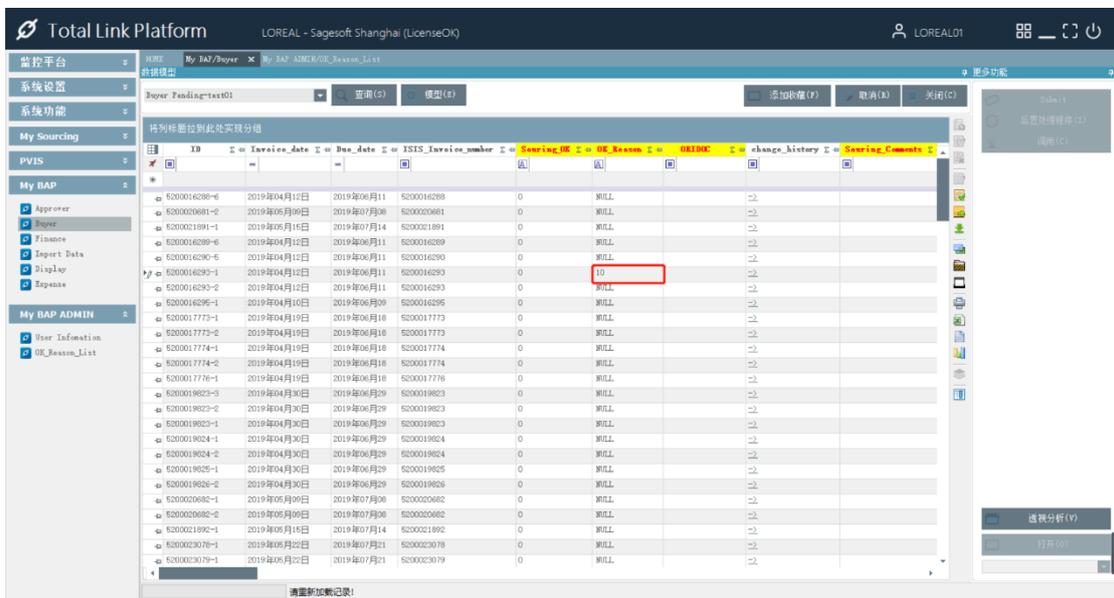
- OK_Reason 列表如下



- 实现效果如下，在进行编辑时，选择“NULL”时,系统传给数据库的值为“0”，保存后数据显示为“NULL”;选择“others”时，系统传给数据库的值为“7”，保存后数据显示为“others”

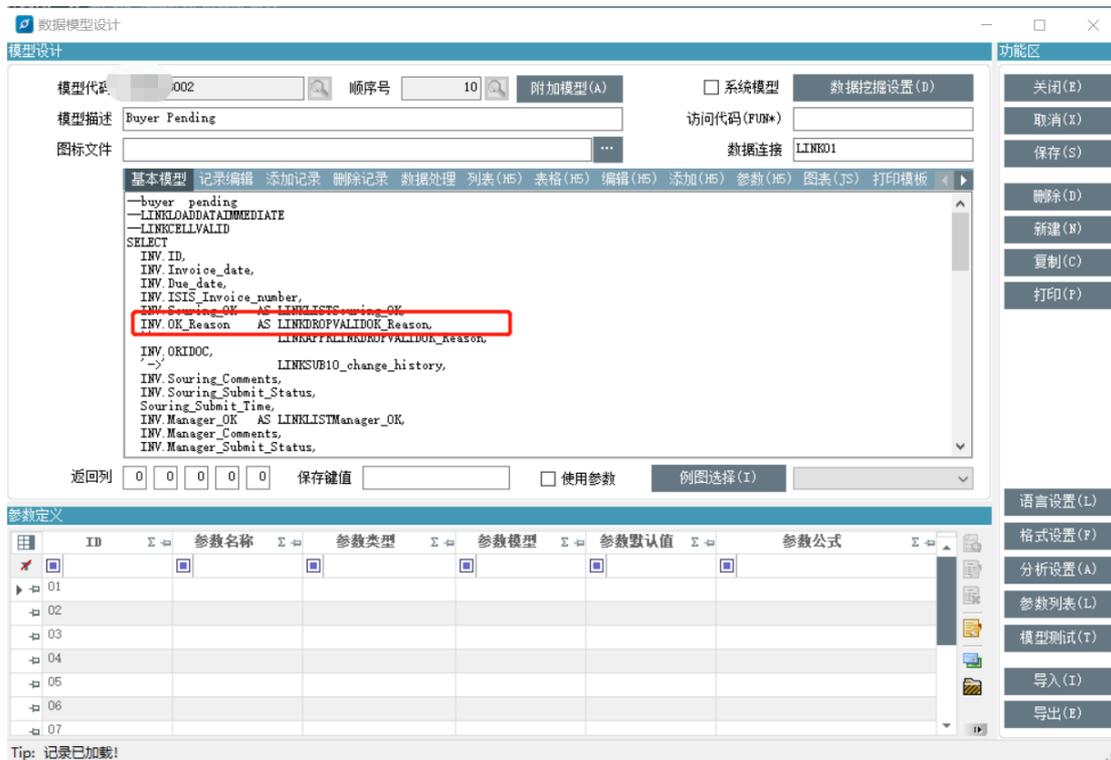


- 当输入“10”时（OK_Reason 表中不存在序号为 10 的显示值），系统不判断该序号在 OK_Reason 表中是否存在，保存进数据库的值为“10”，显示也为“10”

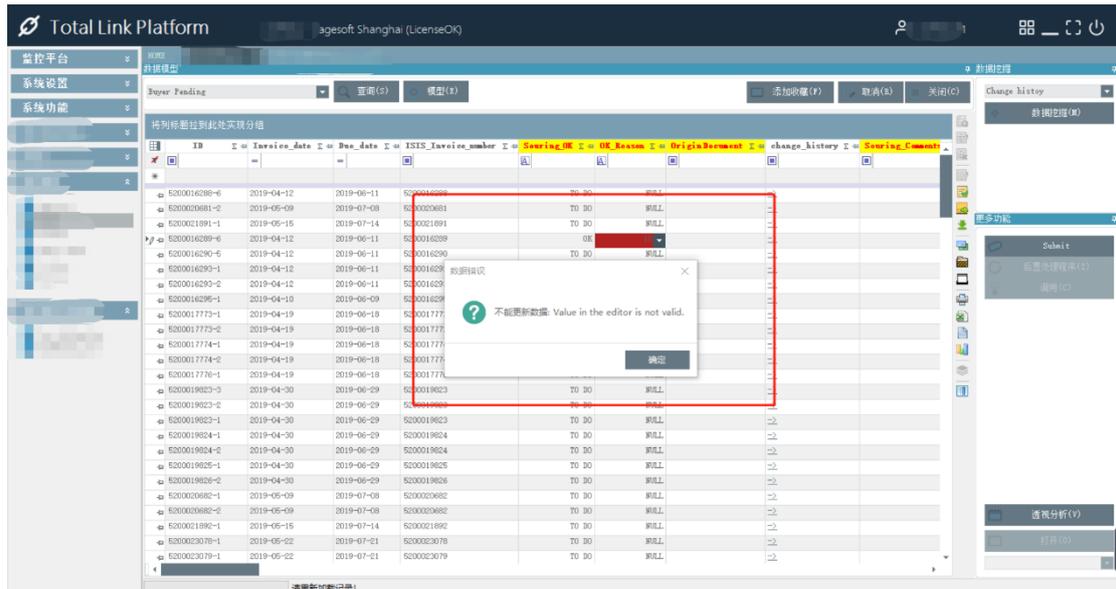


2.16.3 LINKDROPVALIDXXX

- LINKDROPVALIDXXX 下拉框为 LINKDROPDOWNXXX 下拉框的升级版，LINKDROPVALIDXXX 和 LINKDROPDOWNXXX 的区别在于：LINKDROPVALIDXXX 在输入下拉框的值时，不仅可以模糊搜索到值，而且在保存时，如果输入的值在来源表中不存在，则不允许保存
- 实现方式：将 LINKDROPDOWNXXX 的设计过程中所有的“LINKDROPDOWNXXX”换成“LINKDROPVALIDXXX”



- 实现效果如下，当输入 10 时，系统跳出提示编辑的数据无效



2.17 附加模型处理多行数据--LINKMULTIROWS

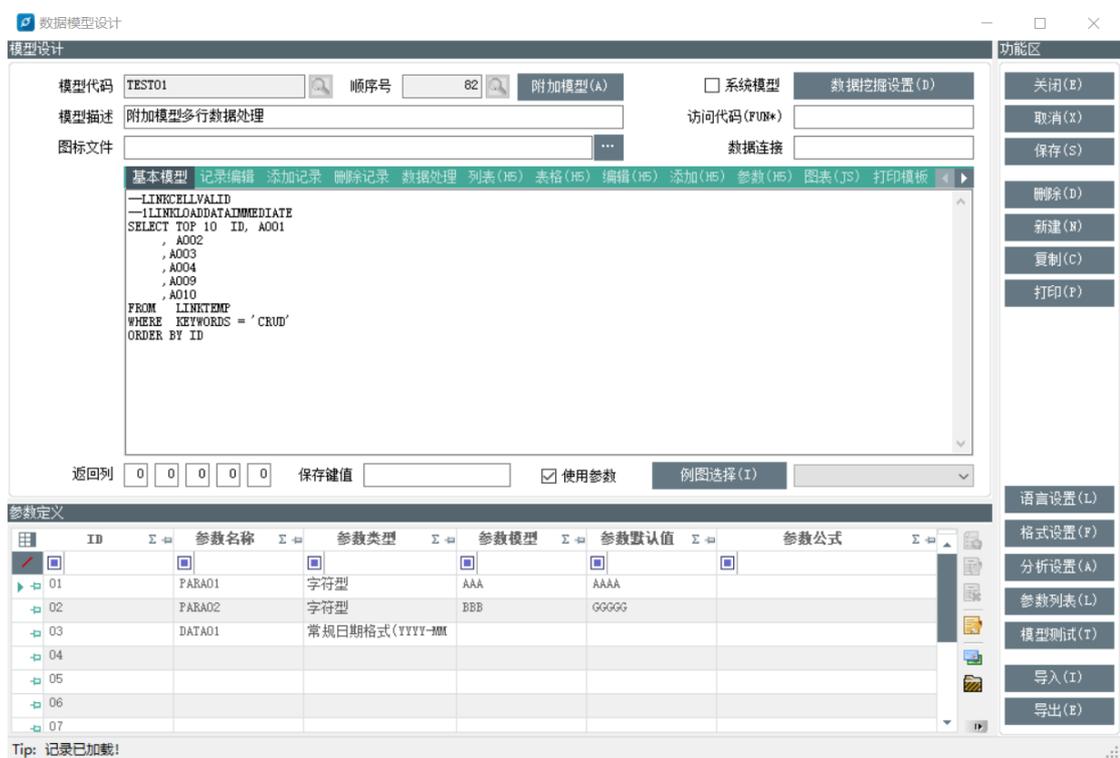
TotalLINK 在附加模型处理数据功能上做了升级，允许用户针对该附加模型查到的记录逐行处理，满足用户多样化的数据需求。应用场景有如发送邮件给多个客户，对不同区域的客户数据进行返利计算等。

使用方法：

- 在基本模型脚本中，增加--LINKMULTIROWS 标记
- 在数据处理脚本中，可以用原来的行记录，也可以用该附加模型查到的行记录
- CallFunction 的处理方式不变
- 不用这种模式时，原来的行为不变
- 使用这种模式时，会针对此处得到的记录，逐行处理（包括 CallFunction 的行为）
- 也可以用于自动任务时，对多条记录的处理
- 也可以用于处理一条记录，但需要通过脚本获取数据，在处理或 CallFunction 时使用

使用步骤介绍:

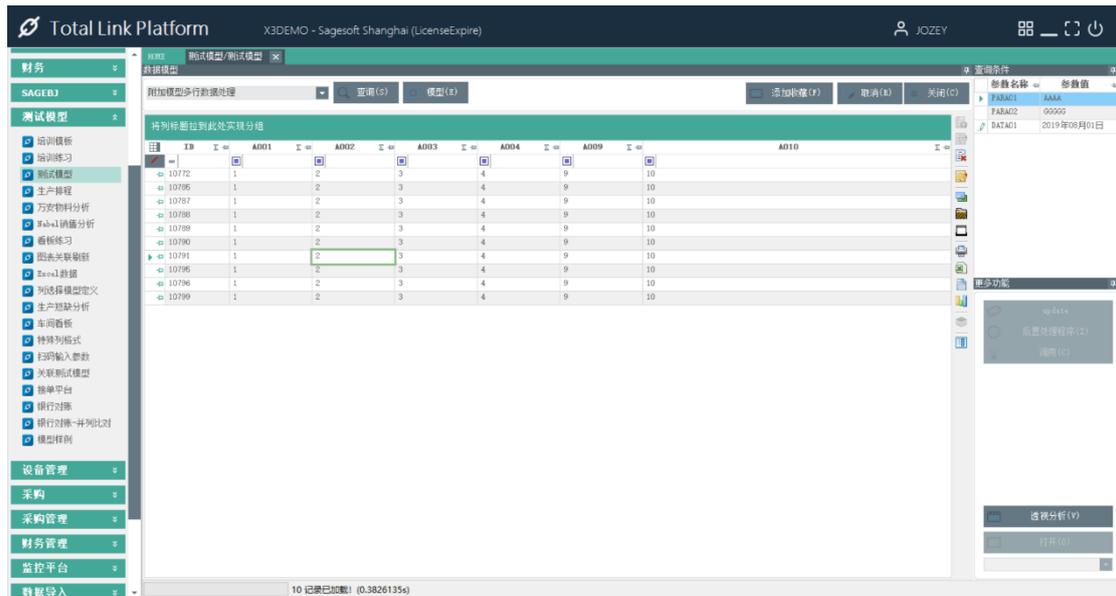
- 基本模型查询的代码如下，意为查询表 LINKTEMP 前 10 条数据:



参考代码:

```
SELECT TOP 10 ID, A001
    , A002
    , A003
    , A004
    , A009
    , A010
FROM LINKTEMP
WHERE KEYWORDS = 'CRUD'
ORDER BY ID
```

- 查询的数据如下:



- 新建附加模型 502，502 查询模型代码如下，意为查询表 LINKTEMP 前 5 条数据的 ID,A009,A10 以及引用基本模型中的 A001 列：



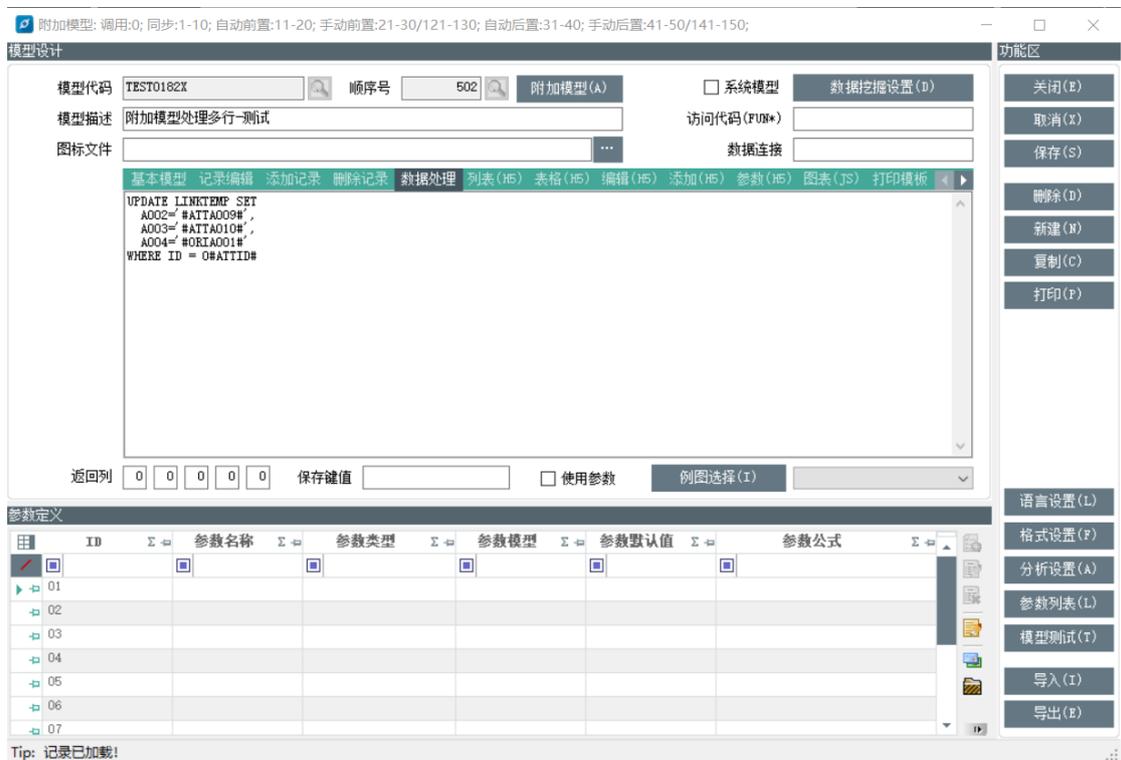
- 该代码数据库查询结果如下：

ATTID	ATTA009	ATTA010	ORIA001
10772	9	10	#A001#
10785	9	10	#A001#
10787	9	10	#A001#
10788	9	10	#A001#
10789	9	10	#A001#

参考代码:

```
--LINKMULTIROWS
SELECT TOP 5 ID ATTID, A009 ATTA009, A010 ATTA010, '#A001#' ORIA001 FROM LINKTEMP
WHERE KEYWORDS = 'CRUD'
ORDER BY ID
```

- 502 数据处理代码如下:



参考代码:

```
UPDATE LINKTEMP SET
A002='#ATTA009#',
A003='#ATTA010#',
A004='#ORIA001#'
WHERE ID = 0#ATTID#
```

- 执行 502 模型，数据变化情况如下表，LINKTEMP 表前 5 行数据做了批量更新:

Total Link Platform X3DEMO - Sagesoft Shanghai (LicenseExpire) JOZEY

财务 SAGEBJ 测试模型 附加模型多行数据处理 添加数据(F) 取消(R) 关闭(C)

特别标题拉到此处实现分组

ID	Σ	A001	Σ	A002	Σ	A003	Σ	A004	Σ	A009	Σ	A010
10772	1	9	10	1	9	10	9	10	9	10		
10785	1	9	10	1	9	10	9	10	9	10		
10787	1	9	10	1	9	10	9	10	9	10		
10788	1	9	10	1	9	10	9	10	9	10		
10789	1	9	10	1	9	10	9	10	9	10		
10790	1	2	3	4	9	10						
10791	1	2	3	4	9	10						
10795	1	2	3	4	9	10						
10796	1	2	3	4	9	10						
10799	1	2	3	4	9	10						

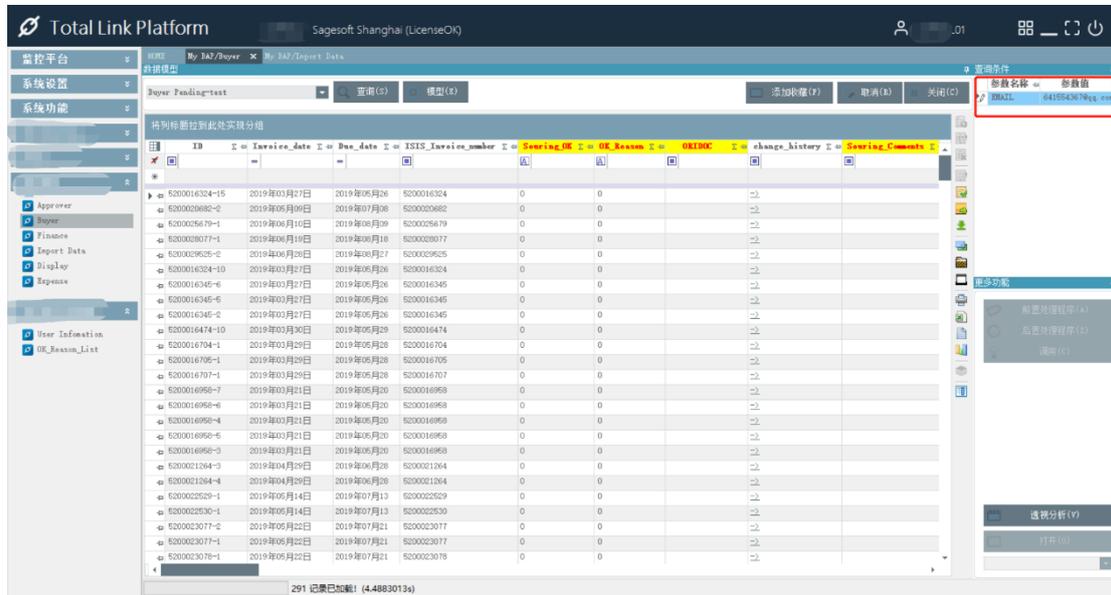
10 记录已加载! (0.5949842s)

参数名称 参数值
 PAR01 AAAA
 PAR02 00000
 DAT01 2019年08月01日

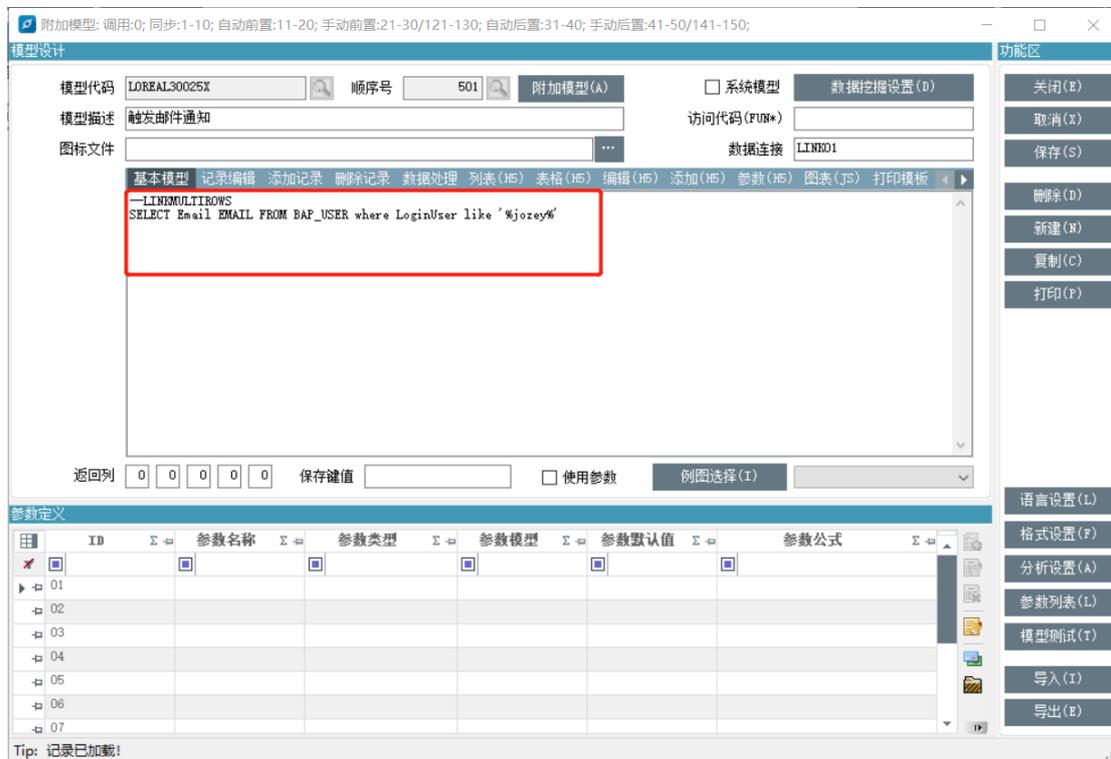
更多功能
 透视分析(V)
 打开(O)

2.17.1 应用一 发送邮件给多个用户

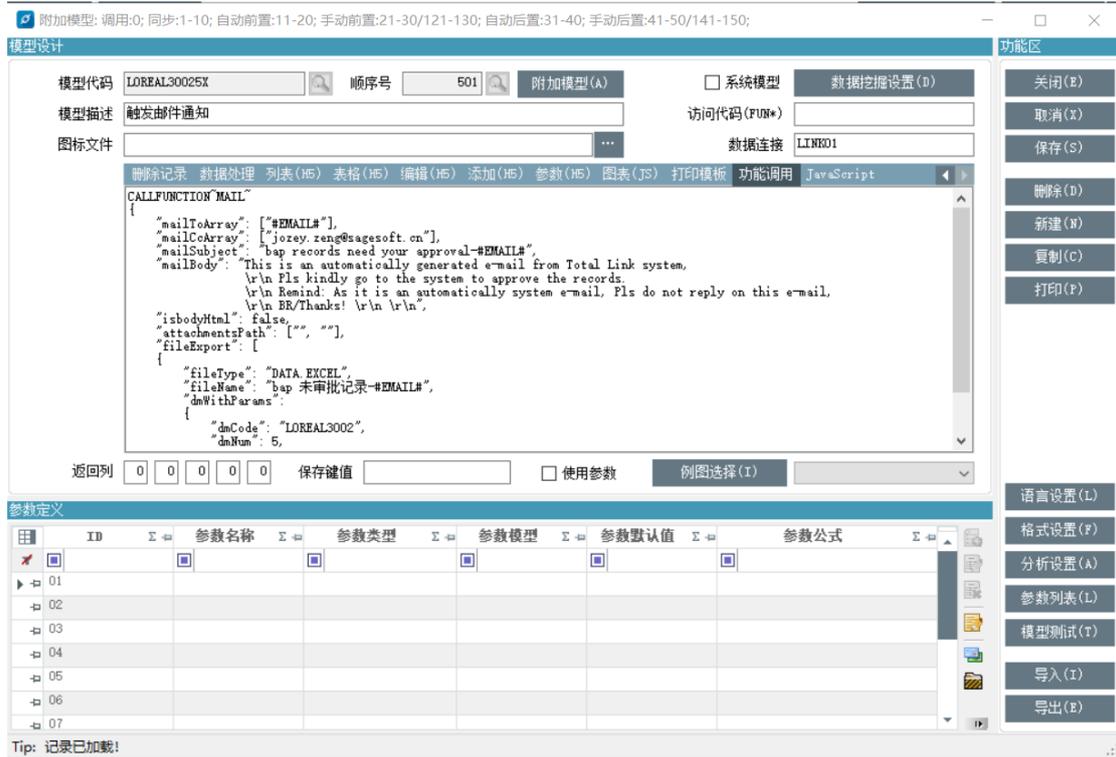
- 该案例背景：系统每周需要将用户未处理的发票数据以邮件方式自动发送到客户邮箱
- 在查询模型，加一个参数，根据用户邮箱查询用户未处理发票行



- 增加附加模型，在查询模型查出待发送邮件的用户邮箱



- 在 CallFunction 调用邮件发送功能



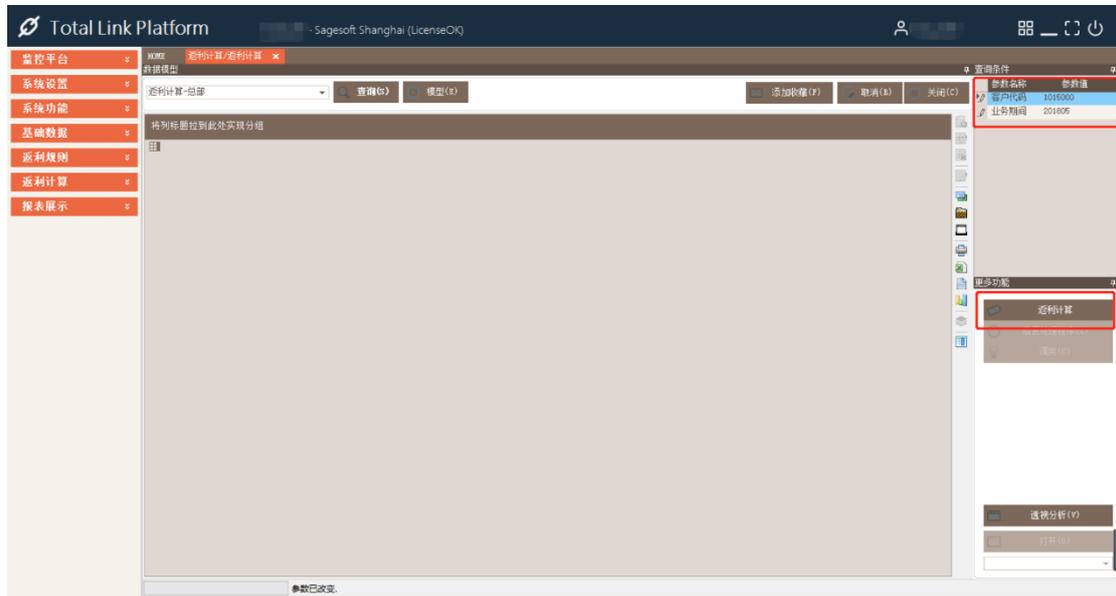
参考代码:

```
CALLFUNCTION~MAIL~
{
  "mailToArray": ["#EMAIL#"],
  "mailCcArray": ["jozey.zeng@sagesoft.cn"],
  "mailSubject": "bap records need your approval-#EMAIL#",
  "mailBody": "This is an automatically generated e-mail from Total Link system,
    \r\n Pls kindly go to the system to approve the records.
    \r\n Remind: As it is an automatically system e-mail, Pls do not
    reply on this e-mail,
    \r\n BR/Thanks! \r\n \r\n",
  "isbodyHtml": false,
  "attachmentsPath": [""],
  "fileExport": [
    {
      "fileType": "DATA.EXCEL",
      "fileName": "未审批记录-#EMAIL#",
      "dmWithParams":
      {
        "dmCode": "LOREAL3002",
        "dmNum": 5,
        "Para": ["#EMAIL#"]
      }
    }
  ]
}
```

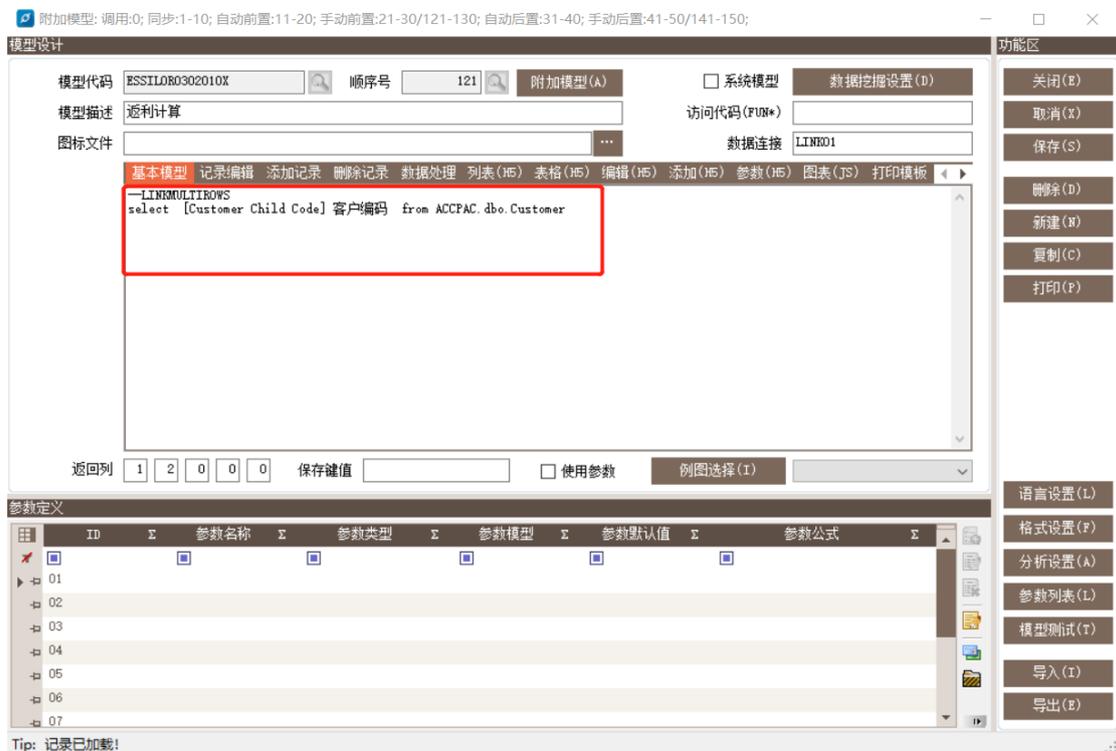
- 注：在有附件时，fileName 必须加上一个不重复的值，否则不同用户的附件内容将会被第一个用户的附件所覆盖
- 完成上面的设置，设定自动任务，可实现给多个用户自动发送邮件的功能。

2.17.2 应用二 自动处理不同区域客户数据

- 该案例背景：系统每月需要自动执行不同区域客户的返利计算



- 在附加模型基本模型查询客户列表



- 在数据处理执行时,直接调用附加模型查询模型里写的“#客户代码#”即可



- 完成上面的设置，设定自动任务，可实现系统每月自动计算所有客户的返利数据

2.18 网页集成说明

TotalLINK 平台为了更好给用户便捷的操作，提供认证模式支撑、统一单点登录和网页系统功能集成，实现一个门户管理多个系统的模式。



15:02 有望客户调查表

首页 / 销售报价合同审批 / 有望客户调查表

请输入搜索内容

编号	卷号	客户名称	工地地址	项目名称	项目经理	申请人	申请日期	流失状态	状态	操作
YWK191000202	789456	微信测试	可以吗			04526	2019-10-25	正常	通过	<input type="button" value="流失申请"/>
YWK191000201	JH19202402	测试客户-test	测试地址			04526	2019-10-24	已流失	通过	
YWK191000199	JH19102401	奥的斯	宁波北仑	奥的斯		04526	2019-10-24	正常	通过	<input type="button" value="流失申请"/>
YWK191000198	LL01	力隆测试	宁波	力隆项目	AA	04807	2019-10-24	正常	通过	<input type="button" value="流失申请"/>
YWK191000197	SYSY	赛义	杭州	AAAAAA	AA	04807	2019-10-23	正常	通过	<input type="button" value="流失申请"/>
YWK191000193	FYFY	冯一	hh	AAA	A	04807	2019-10-22	正常	通过	<input type="button" value="流失申请"/>
YWK191000192	TSETJH	测试客户	测试地址			04526	2019-10-22	正常	通过	<input type="button" value="流失申请"/>

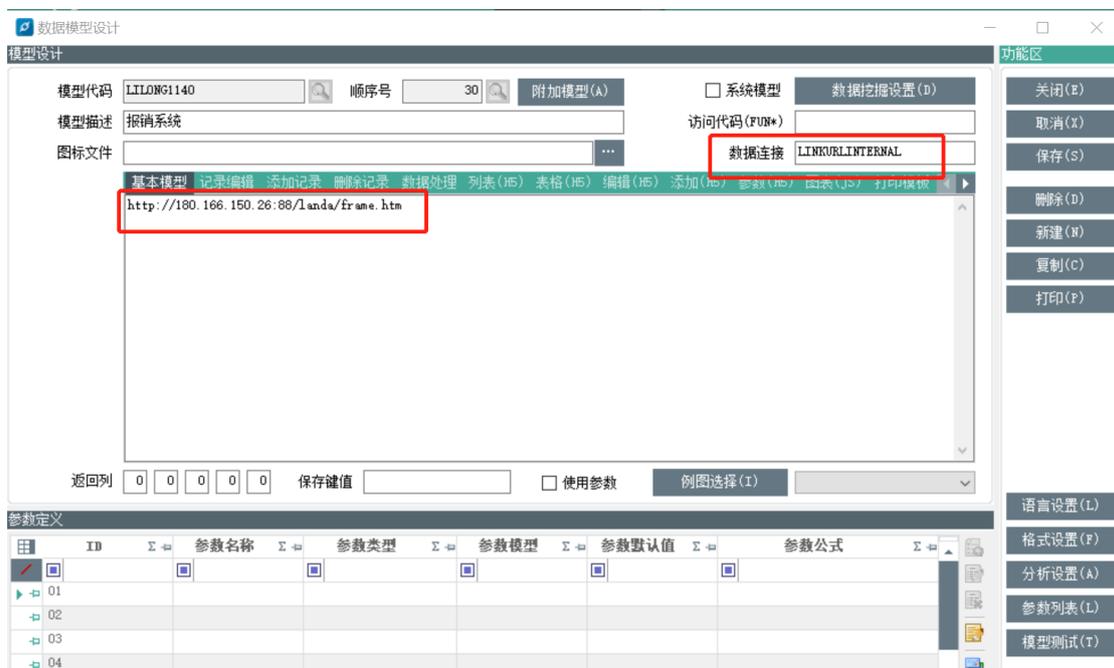
2.18.1 网页集成方式

TotalLINK 平台和网页的集成有两种方式，分别为内部浏览器集成和外部浏览器集成

2.18.1.1 内部集成 LINKURLINTERNAL

内部集成是对应的功能通过 Total Link 内嵌的浏览器直接打开，以客户端中页签的方式展现。模型中的数据连接为关键字：LINKURLINTERNAL

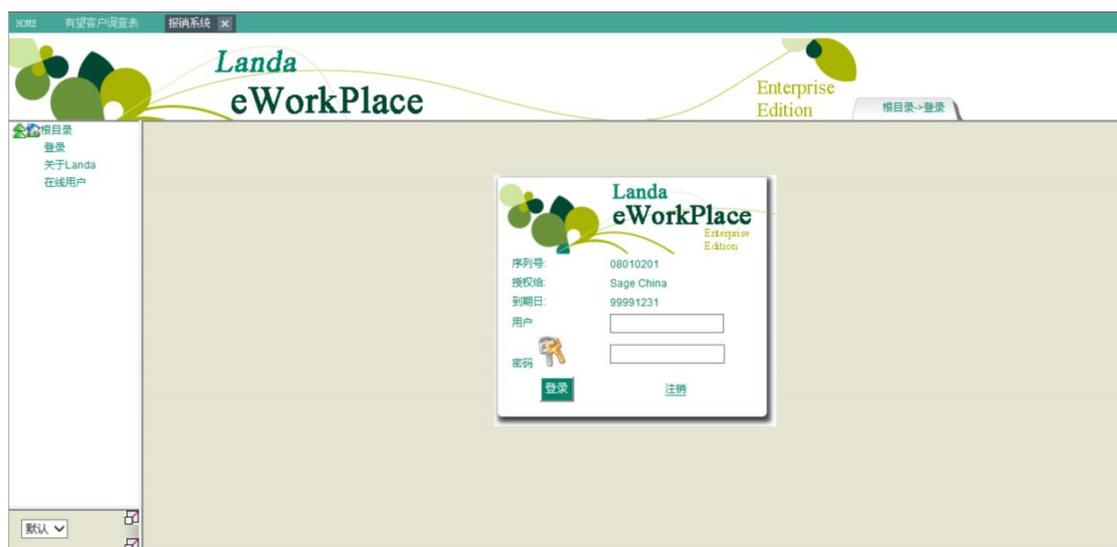
例如集成报销系统，只需要将对应的网址输入基本模型中。



保存后可以将功能收藏到 Home 页，通过图标模式直接打开即可：



多个系统打开后，会产生多个页签，通过页签切换可跳转到不同的系统中。



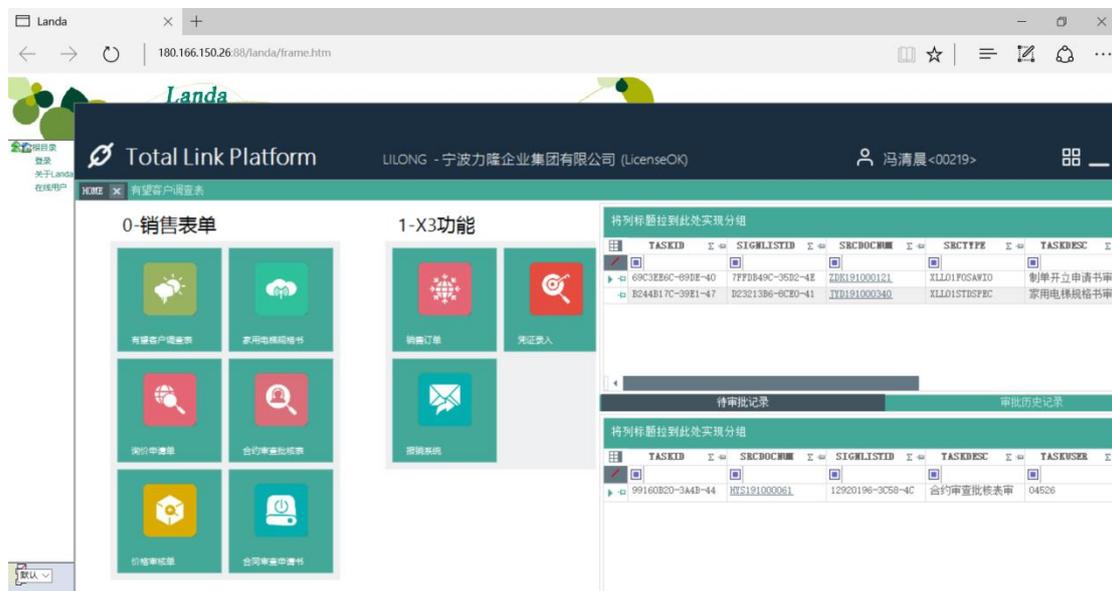
2.18.1.2 外部集成 LINKURL

外部集成是对应的功能通过 Total LINK 外部的浏览器打开，这个平台只是一个系统的入口，模型中的数据连接为关键字：LINKURL

例如集成报销系统，只需要将对应的网址输入基本模型中。

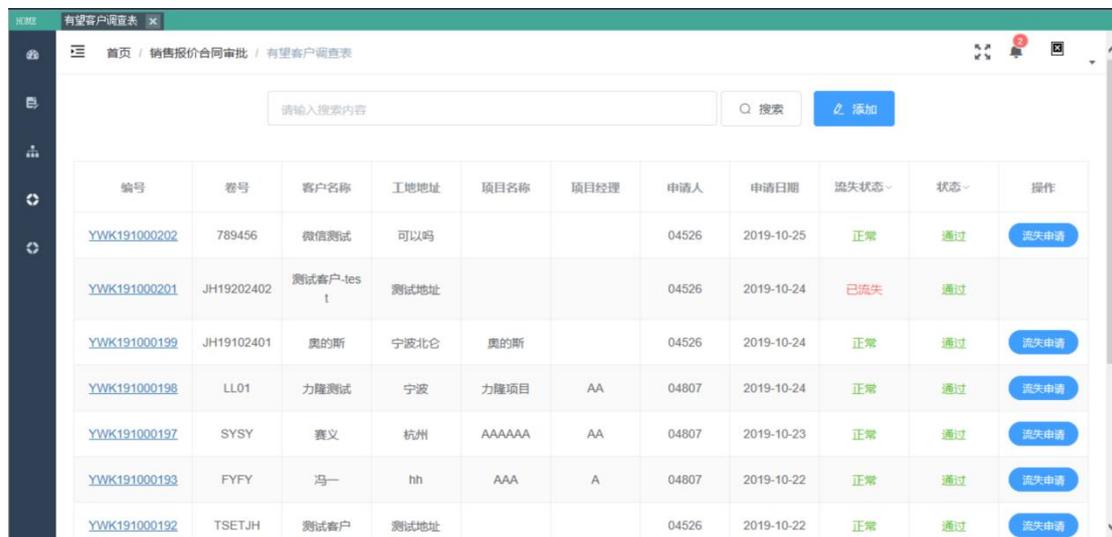
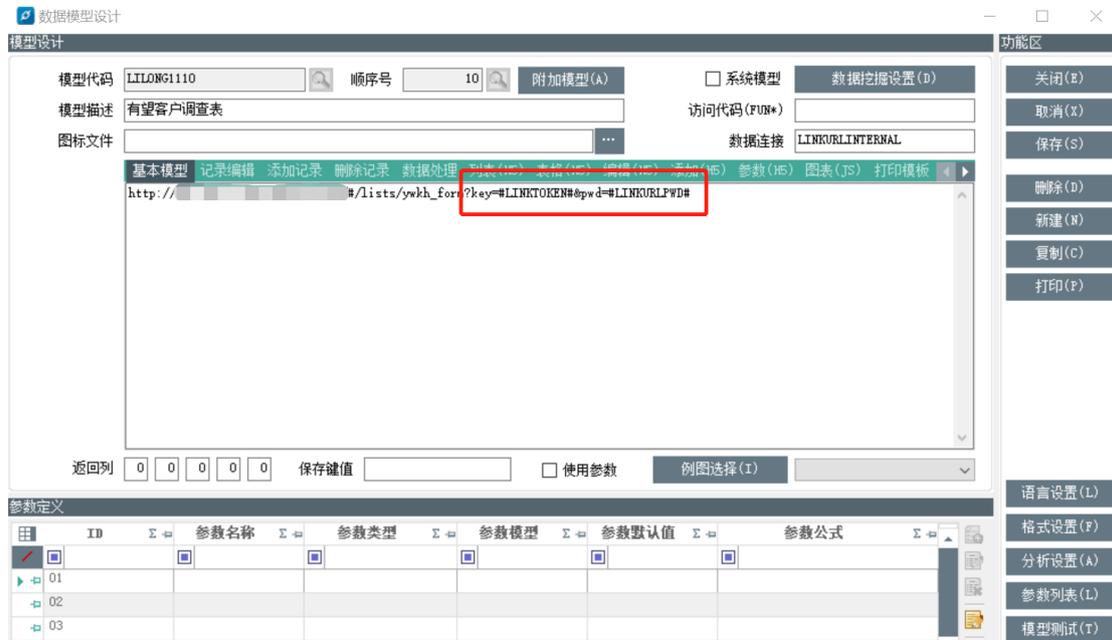


保存后可以将功能收藏到 Home 页，通过图标模式直接打开，会在外部的浏览器打开对应的网页。



2.18.2 用户认证

为了更好的实现无缝对接网页，在模型中加入用户认证参数信息，即可跳过登录直接进入功能界面。



2.18.3 链接到具体单据

实现效果：将待审批的记录挂在 HOME 页面，通过点击可挖掘的编号，直接进入具体的单据做审批。



实现步骤：

- 因为待审批的可能是不同类型的单据，所以需要设置为

动态数据挖掘，通过公式指定挖掘到多个不同的模型

- * 模型代码允许使用公式，可以引用当前行记录的数据
- * 模型编号必须统一，比如都是 10
- * 模型名称不需要填写，系统自动显示为 相关单据 (zh) / Related Document (en)

数据模型定义

查询(S) 模型(E) 添加收藏(F) 取消(R) 关闭(C)

将列标题拖到此处实现分组

ID	SEARCHCODE	SEARCHNUM	MODELDESC	PARA01	PARA02	PARA03	PARA04	PARA05
10	#DMCODE#	110		#LINKSUB10_SECDO	#SIGNLISTID#			
20	#DMCODE#	10		#LINKSUB10_SECDO				

数据模型设计

模型设计

模型代码: LILONG0107 序号: 110 附加模型(A) 系统模型 数据挖掘设置(D)

模型描述: 制单开立申请书-待审 访问代码(FUN*):

图标文件: 数据连接: LINKURLINTERNAL

基本模型 记录编辑 添加记录 删除记录 数据外理 列表(H5) 表格(H5) 编辑(H5) 添加(H5) 参数(H5) 图表(T5) 打印图标

`http://
@create=0 /lists/rdkl_form?key=#LINKTOKEN#&pwd=#LINKURLPWD#&type=0&id={1}&sign={2}`

返回列: 0 0 0 0 0 保存键值 使用参数 例图选择(I)

参数定义

ID	参数名称	参数类型	参数模型	参数默认值	参数公式
01					
02					
03					

type=0 是从待审批进入的 (需要传入 sign=SIGNLISTIID 或者 row={}, 当 row 没有值且 sign 有值则会自动查询获取对应的待审批记录)

type=1 是数据挖掘跳转

type=2 是添加编辑查看功能(需要传入 create=0/1/2, 对应的是查看/编辑/添加。当查看或者编辑的时候需要传入 id=VCRNUM_0 或者 row, 当 row 没有值且 id 有值则会自动查询获取对应的模型记录)

type=3 是从审批记录进入的与 type=0 一致

id=VCRNUM_0 是单据编号 传对应的单据编号

sign=SIGNLISTID 是待审批记录的 SIGNLISTID 传对应的单据编号

create=0 是查看

create=1 是编辑

create=2 是添加

row={} 是当前行的记录集

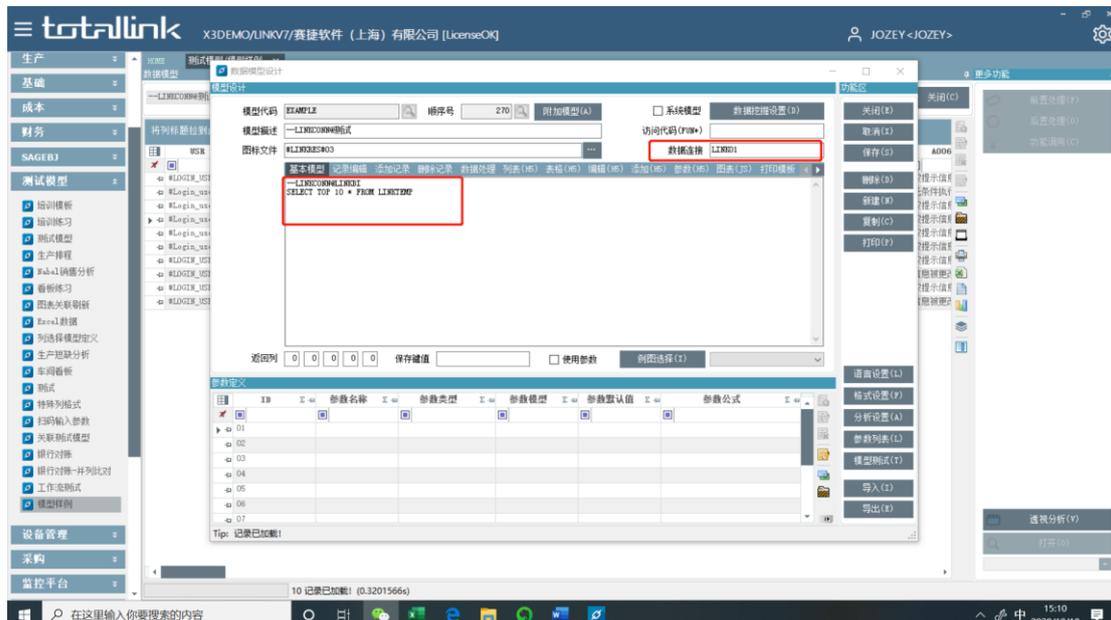
可以传入对象或者字符 '{}'

2.19 从指定的链接获取数据--LINKCONN@XXX

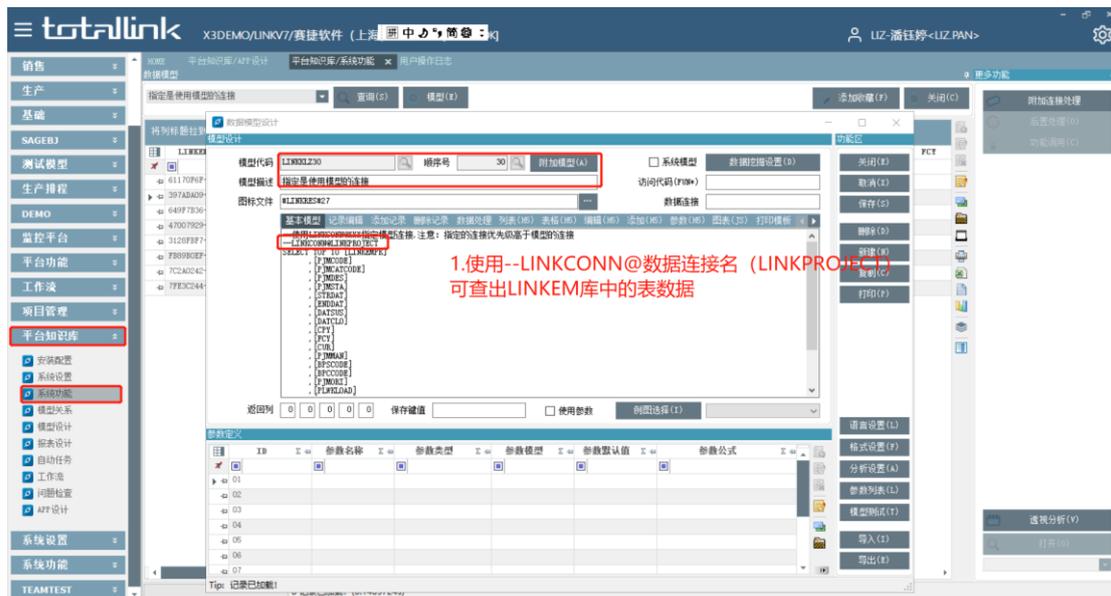
TotalLINK 支持用户在模型的数据链接处填写链接代码，也支持用户在模型模块代码中嵌入—LINKCONN@XXX 代码（版本 V8.1.5.7 及以上可使用）。

- LINKCONN@XXX 链接优先级高于模型的数据链接
- 系统执行完 LINKCONN@XXX 链接的语句后再执行模型的数据链接相关语句

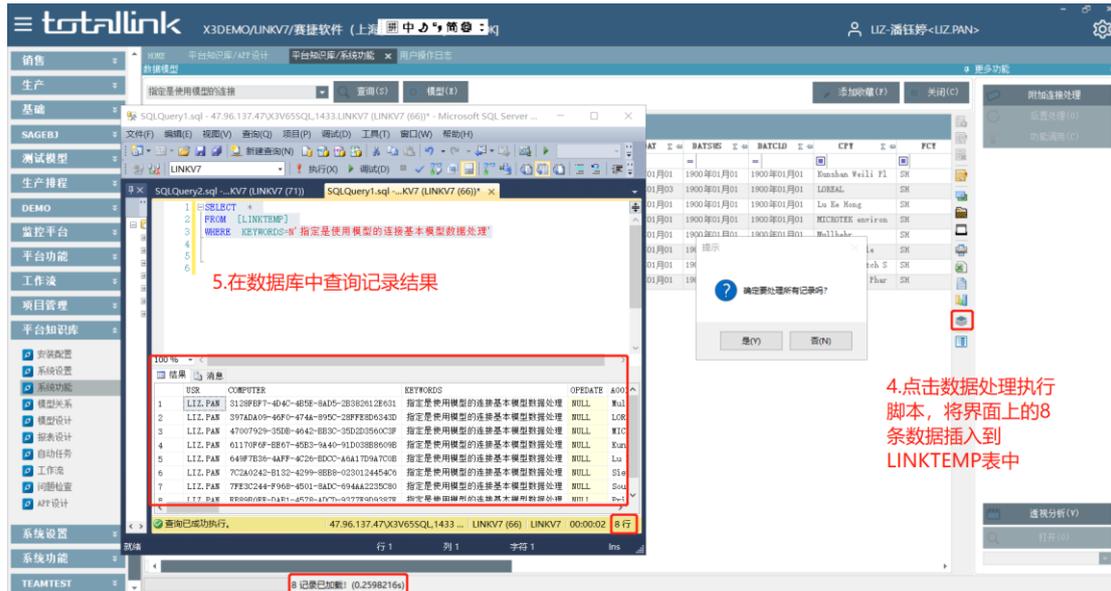
- 当数据链接所在数据库不能直接访问其它数据库数据时，可以用这种方式访问数据库



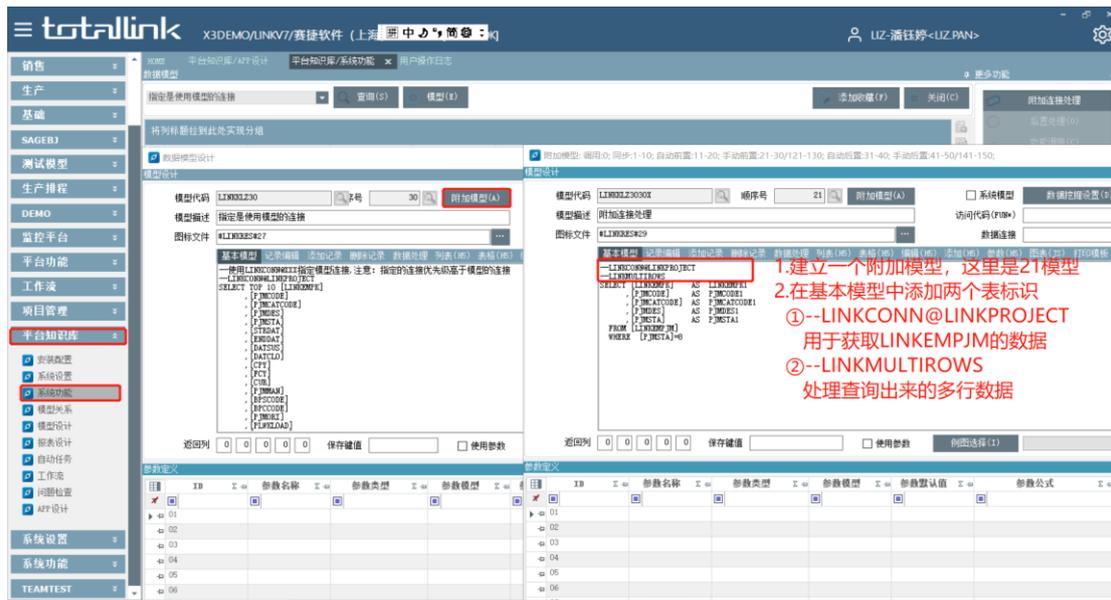
应用场景一：先处理其它数据库的数据，再处理模型链接的数据库代码。

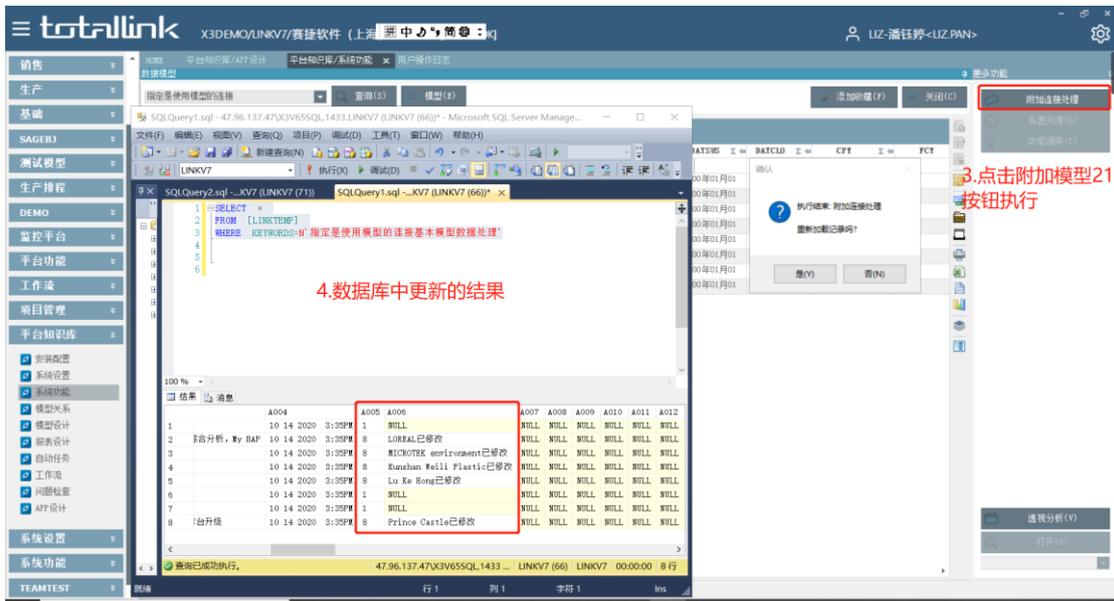
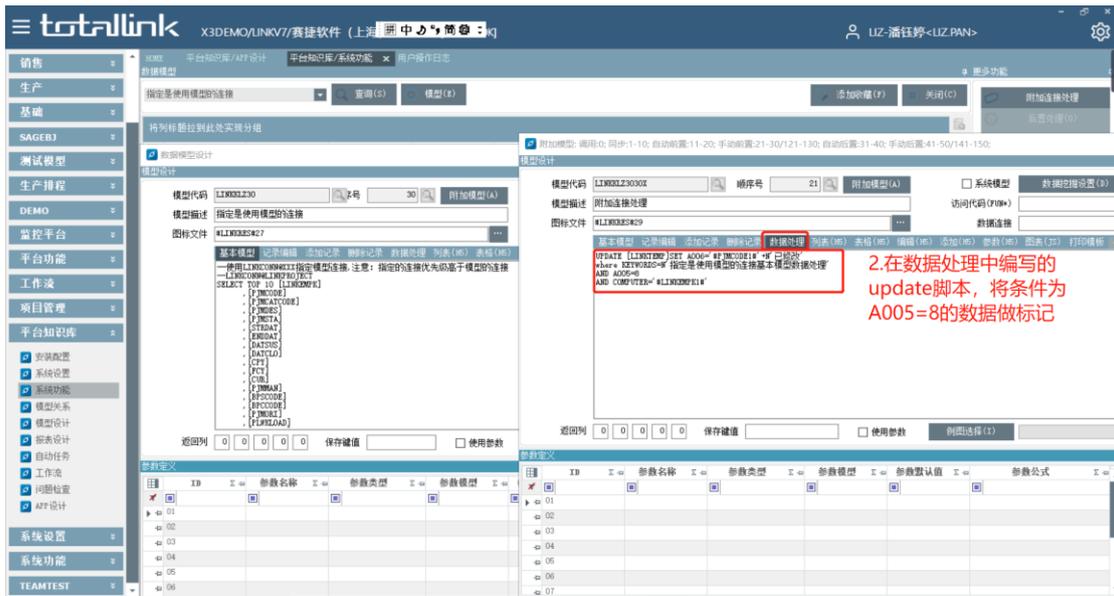






应用场景二：和--LINKMULTIROWS 关键字配合使用，先查出其他连接数据，修改本地数据。





应用场景参考平台知识库下“系统功能”菜单模型 LINKKLZ30，顺序号 30。

在 TotalLINK20 的 V8.2.3.15 版本中更新了一LINKCONN@XXX，允许在脚本中通过参数或字段值替换连接字符串代码。

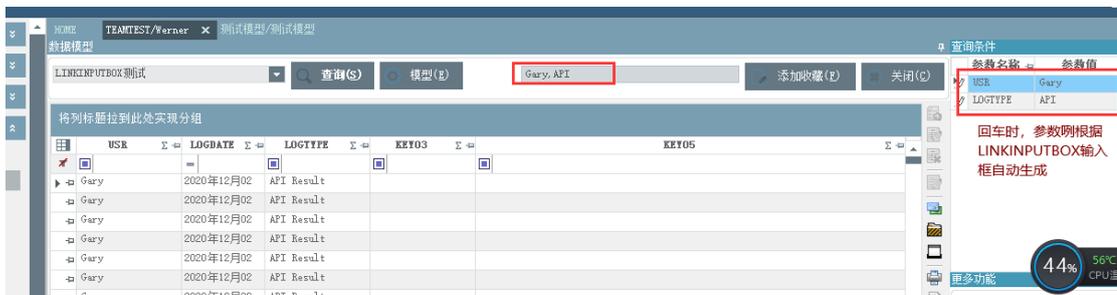


2.20 查询模型直接加载数据—LINKINPUTBOX

- 在查询模型中加入“—LINKINPUTBOX”，可以实现增加参数快捷输入

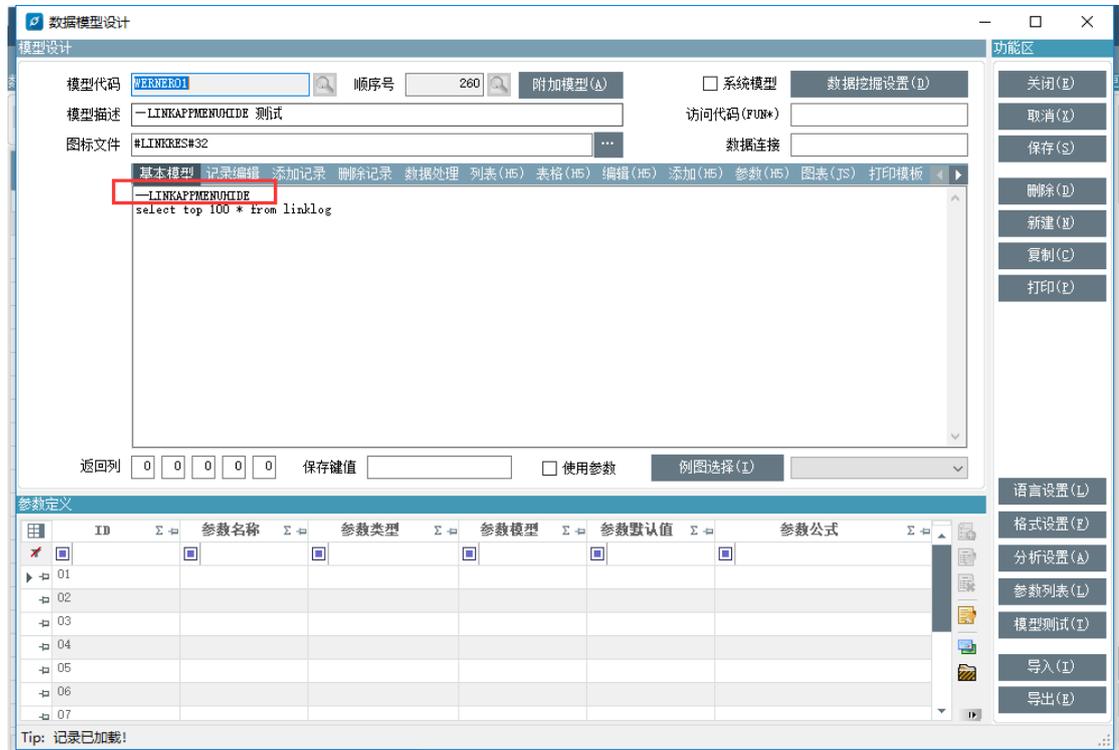


- 实现效果如下，允许用户在扫描解析直接处理（设置为带回车，或手工输入回车），输入时用“,”分隔参数



2.21 标记某些模型不出现在 APP 菜单中

在基本模型中输入—LINKAPPMENUHIDE



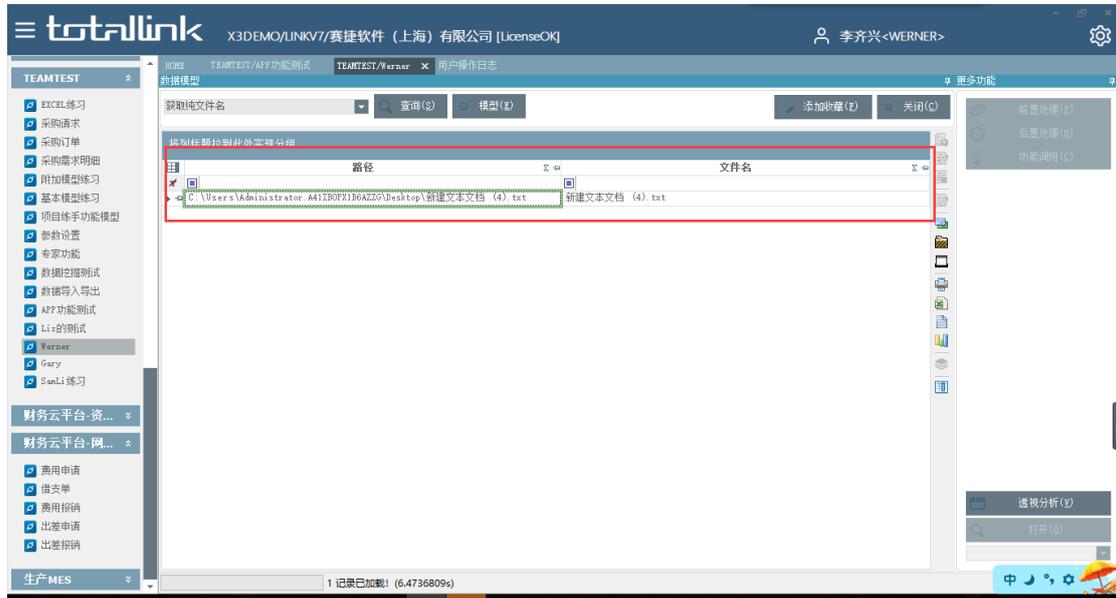
无 LINKAPPMENUHIDE 效果展示



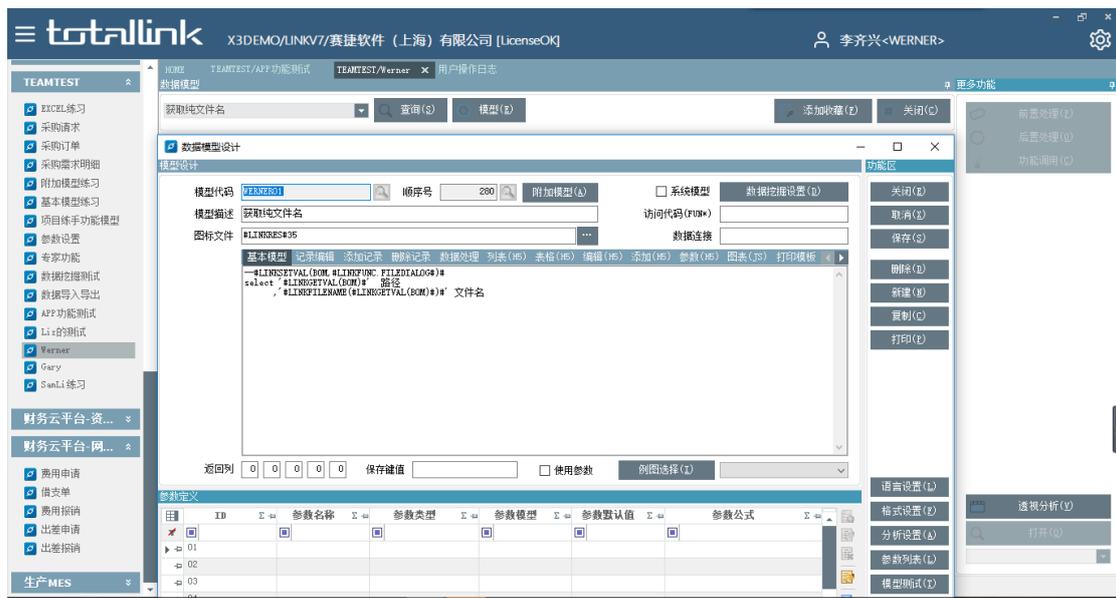
有 LINKAPPMENUHIDE 效果展示



2.22 获取纯文件名



具体实现操作如下：



主模型代码如下显示

```
--#LINKSETVAL(BOM,#LINKFUNC.FILEDIALOG#)#
select '#LINKGETVAL(BOM)#' 路径
, '#LINKFILENAME('#LINKGETVAL(BOM)#)#' 文件名
```

- 在这里采用的 LINKSETVAL 的方式（详细见 LINK10 过程变量的值保存和值获取），值的保存

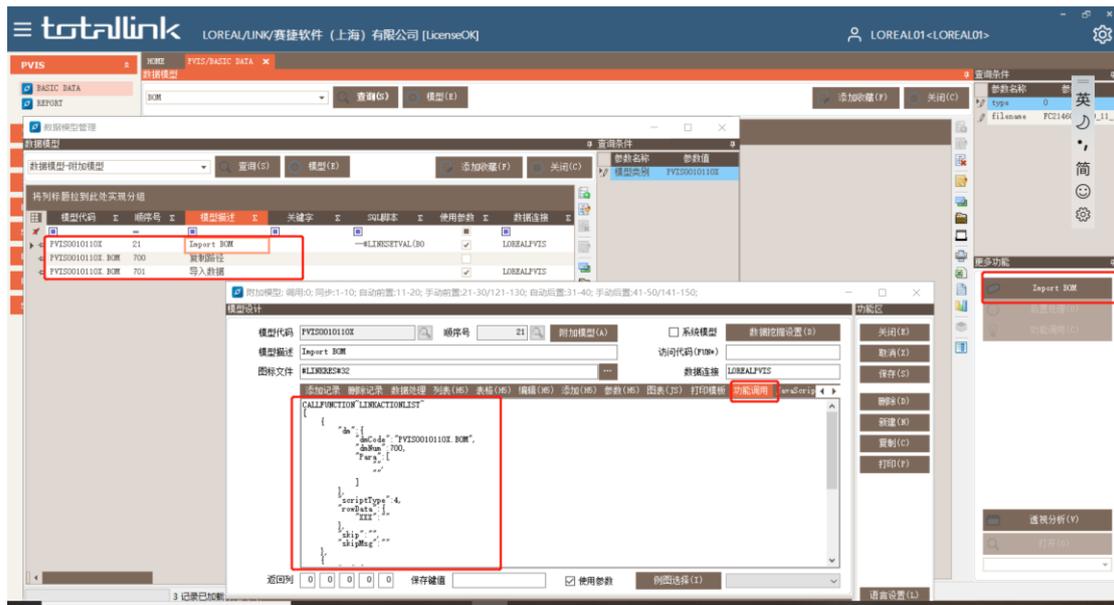
- #LINKFUNC.FILEDIALOG# 获取文件路径
- #LINKFILENAME(#LINKGETVAL(BOM)#)# 获取纯文件名

3 功能调用

3.1 LINKACTIONLIST 功能

LINKACTIONLIST 功能可以一次调用多个不同的模型。

案例：使用 Bulk Insert 导入“.TXT”文件数据，需将“.TXT”文件复制到服务器上，再进行导入，具体如下所示：



案例中，21 模型的功能调用编写 CALLFUNCTION~LINKACTIONLIST~，参考代码如下：

```
CALLFUNCTION~LINKACTIONLIST~
[
  {
    "dm":{
      "dmCode":"PVIS0010110X.BOM",//调用复制文件的模型
      "dmNum":700,
      "Para":["",""]
    },
    "scriptType":4,//执行复制文件的模型的功能调用
    "rowData":{
      "xxx":""
    },
    "skip":"","//如果 skip=1, 则跳过不执行对应的 Action
    "skipMsg":"","//skipMsg 有数据, 则提示信息
  },
  {
    "dm":{
```

```

        "dmCode": "PVIS0010110X.BOM", //调用导入数据模型
        "dmNum": 701,
        "Para": [ "", "" ]
    },
    "scriptType": 4,
    "rowData": {
        "XXX": ""
    },
    "skip": "#FLOG#", //该模型通过 LINKMULTIROWS，经判断传值
    "skipMsg": "提示"
}
]
    
```

特别说明：

- 21 模型的基本模型处理逻辑



- 复制文件到服务器上

