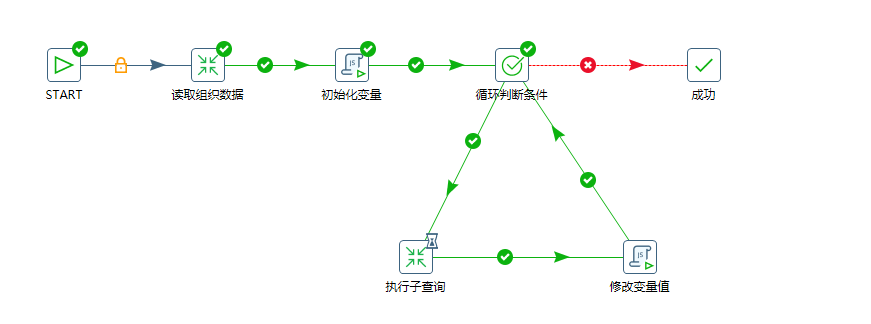
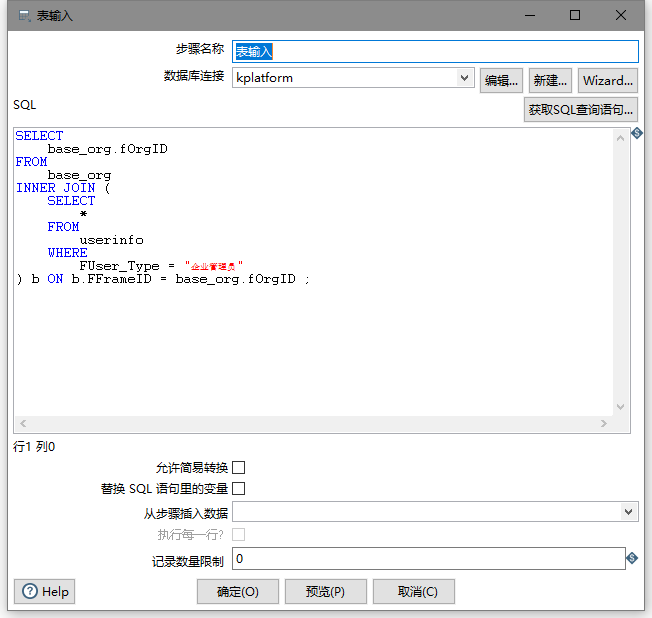
整体流程如下：



这是一个Kettle 的作业，第一个转换内容是查询的表数据，同时复制到记录上。



这是转换1(读取组织架构数据)



值得注意的是：在这里只是为了测试循环遍历的整个流程，整个转换在使用时使用的应该是配置文件，通用的配置文件。

第二个（初始化变量步骤） 遍历获取功能是一个JavaScript脚本验证，主要使用的是获取上一个转换传递过来的数据。具体代码如下：

var prevRow=previous\_result.getRows();//获取上一个传递的结果

if (prevRow == null &&(prevRow.size()=0))

{

false;

}else{

parent\_job.setVariable("tables", prevRow);//ArrayList存储表名变量，以数组形式保存入table1，table2

parent\_job.setVariable("size", prevRow.size());//存储执行表的总数量

parent\_job.setVariable("i", 0);//循环控制变量

/\*\*

TABLENAME 这里写死了，hbp\_record 代表了需要查询的表，

prevRow.get(0).getString("TABLES","")//这是通用版的

\*/

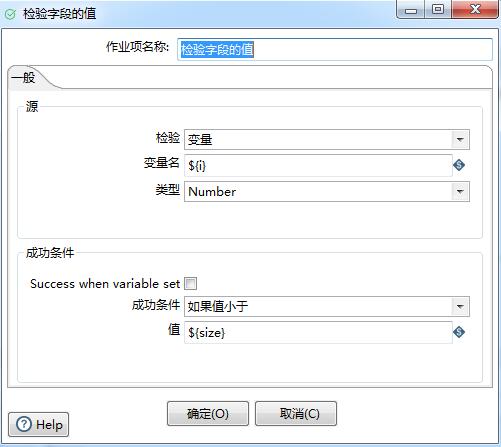
parent\_job.setVariable("TABLENAME","HBP\_RECORD");

true;

}

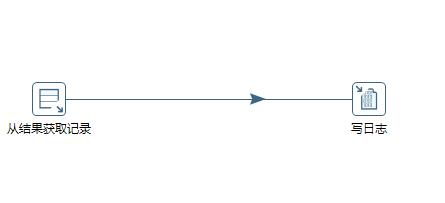
第三步：检验字段的值

这个步骤代表类似Java中的for循环语句，进行循环遍历数据使用的，配置信息如下：



检验字段的值

第四步：转换2具体的步骤如下：



转换2实现数据的输出

第五步：再次使用Javascript的脚步验证

主要功能是：进行数据的累加，遍历执行表的数据

代码如下：

var list\_Tables =parent\_job.getVariable("tables").replace("[","").replace("]","").split(",");

var size = new Number(parent\_job.getVariable("size"));

var i = new Number(parent\_job.getVariable("i"))+1;

if(i<size){

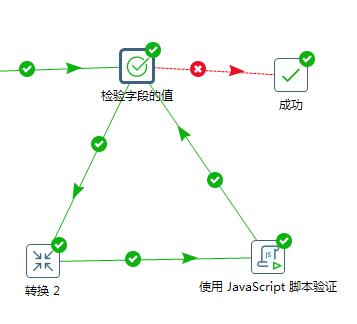
parent\_job.setVariable("TABLENAME", list\_Tables[i]);

}

parent\_job.setVariable("i",i);

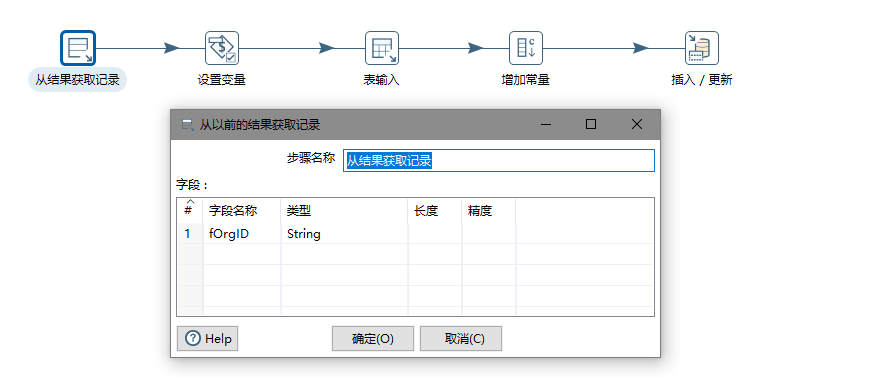
true;

备注：

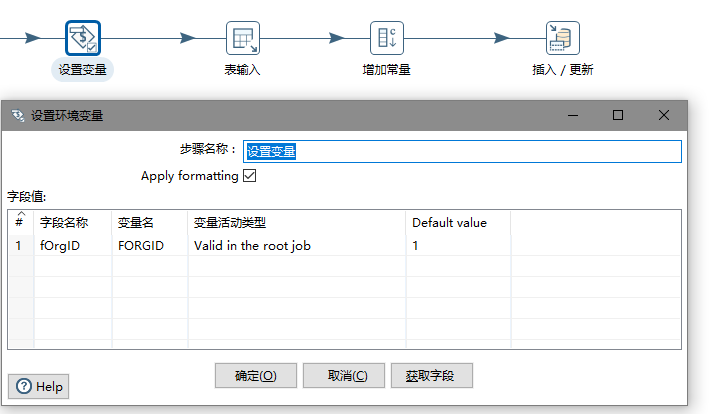


图表1

转换2（执行子查询）：我们需要在转换2中拿到每一个循环的数据



然后需要把拿到的结果映射为当前转换的上下文的变量：



后面就可以使用该变量了

