可视化ETL工具Kettle

ETL，是英文Extract-Transform-Load的缩写，用来描述将数据从来源端经过抽取（extract）、转换（transform）、加载（load）至目的端的过程。**ETL**一词较常用在数据仓库，但其对象并不限于数据仓库。

Kettle介绍

对于企业或行业应用来说，经常会遇到各种数据的处理，转换，迁移，掌握一种etl工具的使用，必不可少，这里要学习的ETL工具是——Kettle，现在已经更名为**PDI**。

- Kettle是一款国外开源的ETL工具，纯java编写，可以在Window、Linux、Unix上运行，绿色无需安装
- Kettle 中文名称叫水壶，该项目的主程序员MATT 希望把各种数据放到一个壶里，然后以一种指定的格式流出
- Kettle允许管理来自不同数据库的数据，提供一个图形化的用户环境来描述想做什么，无需关心怎么做

大数据岗位需求

Kettle安装、配置

环境要求：

- 安装、配置好JDK

1、下载Kettle

- `资料\安装包\pdi-ce-8.2.0.0-342.zip`

2、解压Kettle

3、双击spoon.bat 启动spoon

Kettle入门案例

需求：

- 把数据从CSV文件（ketttle测试数据\用户数据源\user.csv）抽取到Excel文件

数据源：

大数据开发-QY1001 (职位编号：TAIJ... 1.2-2万/月

[redacted]计算机股份有限公司 查看所有职位

杭州-西湖区 | 3-4年经验 | 本科 | 招若干人 | 10-04发布

五险一金 补充医疗保险 绩效奖金 交通补贴 餐饮补贴 弹性工作 定期体检

职位信息

岗位职责:

岗位职责:

1、从项目需求出发，运用数据挖掘和机器学习方法和技术，完成相关功能开发

2、优化算法，提升模型准确度，满足实际系统应用需求

任职资格:

任职要求

1.计算机相关专业，3年及以上大数据开发工作经验。

2.熟练使用apache hadoop生态体系相关技术,包括zookeeper、Hadoop、Hbase、hive、Spark等工具，具有实际项目经验。

3.熟练使用flume、sqoop、kettle、kafka和Elasticsearch等工具进行数据的采集和全文检索等工具的使用，具有实际项目经验。

ETL开发工程师 0.8-1.5万/月

北京凯拉斯信息技术有限公司 查看所有职位

北京-西城区 | 无工作经验 | 招若干人 | 09-30发布

五险一金 通讯补贴 餐饮补贴 年终奖金

职位信息

1.熟练使用Oracle、Mysql等数据库；熟练使用Informatic、Kettle等ETL工具进行数据处理；

2.至少熟练掌握ERwin、Power Designer等其中一种建模工具，可基于对业务的理解搭建需求模型；

实现步骤：

1、在Kettle中新建转换

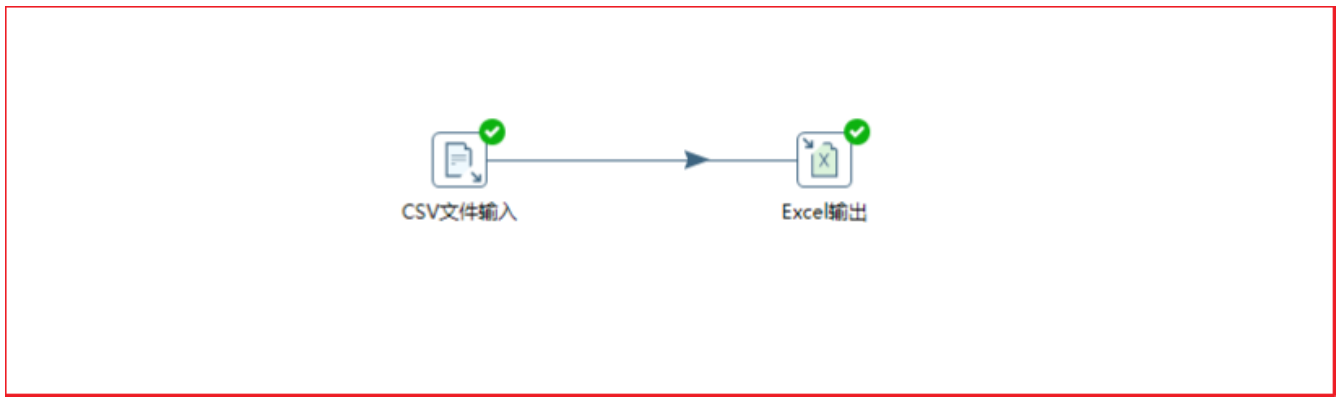
2、拖拽一个CSV输入组件、一个Excel输出组件、并按住Shift拖动鼠标连接两个组件

3、配置CSV输入组件、Excel输出组件

具体实现：

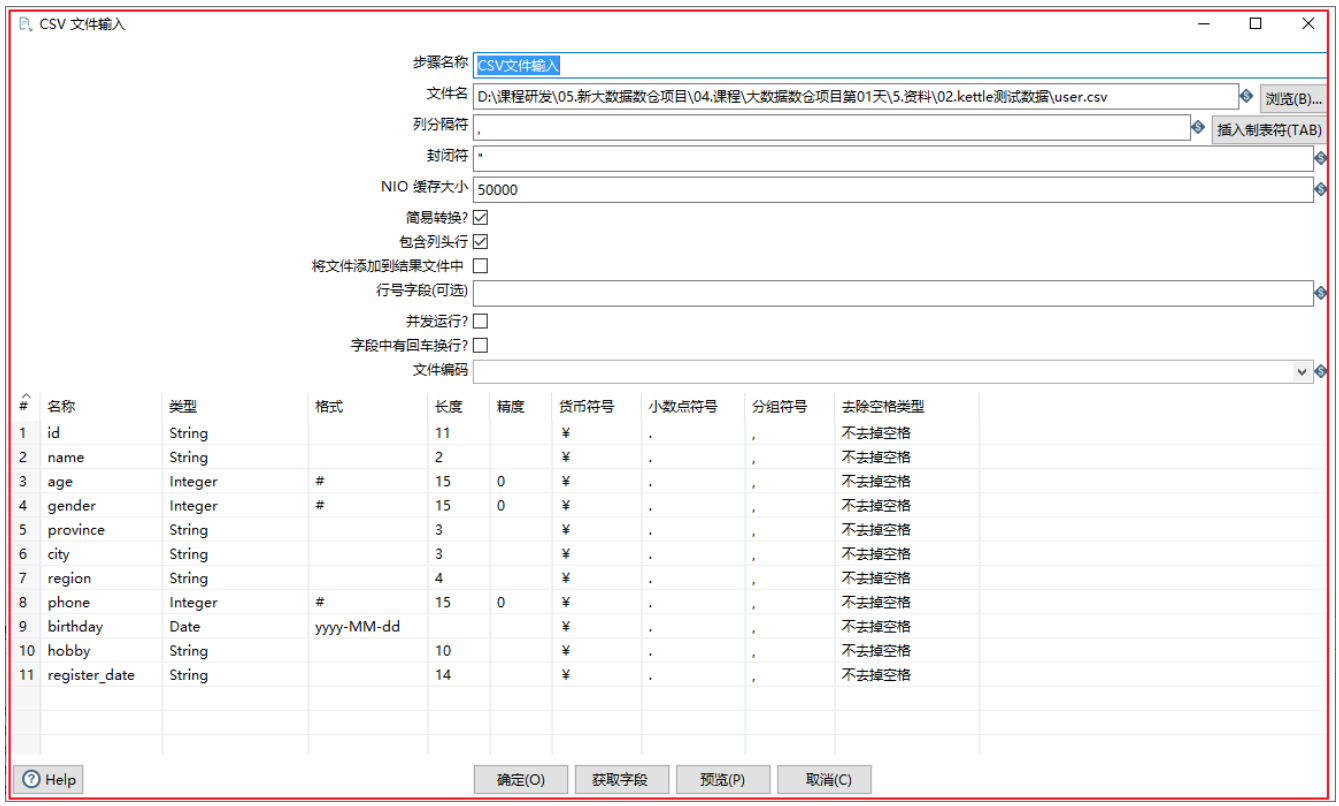
1、新建转换

2、拖拽一个CSV输入组件、一个Excel输出组件、并按住Shift拖动鼠标连接两个组件



3、配置CSV输入组件

- 选择要进行导入的CSV数据源
- 点击 「获取字段」，读取CSV中的列
- 点击 「预览」，浏览CSV中的数据

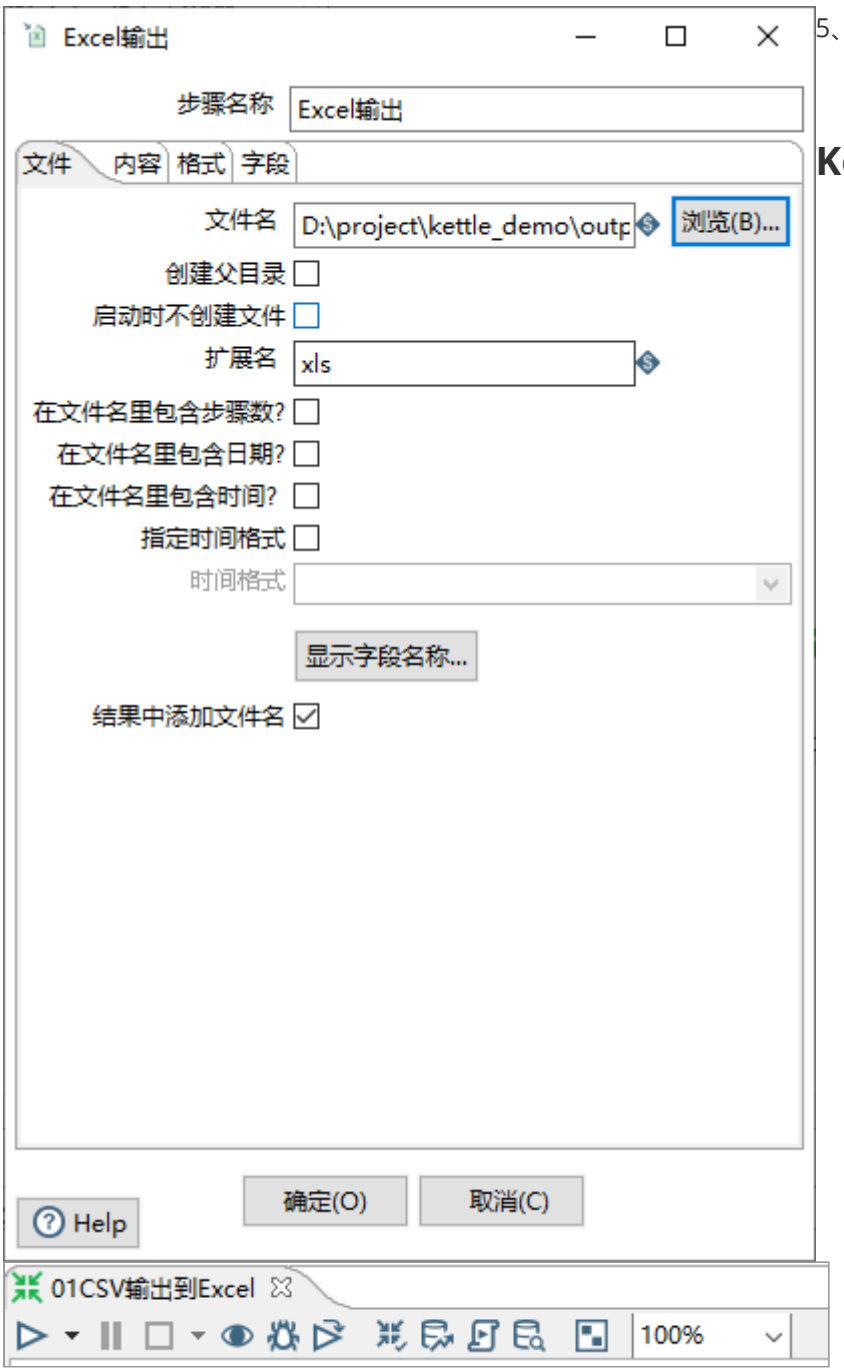


#	名称	类型	格式	长度	精度	货币符号	小数点符号	分组符号	去除空格类型
1	id	String		11		¥	.	,	不去掉空格
2	name	String		2		¥	.	,	不去掉空格
3	age	Integer	#	15	0	¥	.	,	不去掉空格
4	gender	Integer	#	15	0	¥	.	,	不去掉空格
5	province	String		3		¥	.	,	不去掉空格
6	city	String		3		¥	.	,	不去掉空格
7	region	String		4		¥	.	,	不去掉空格
8	phone	Integer	#	15	0	¥	.	,	不去掉空格
9	birthday	Date	yyyy-MM-dd			¥	.	,	不去掉空格
10	hobby	String		10		¥	.	,	不去掉空格
11	register_date	String		14		¥	.	,	不去掉空格

#	id	name	age	gender	province	city	region	phone	birthday	hobby	register_date
1	3.92456E+17	张三	20	0	北京市	昌平区	回龙观	18589407692	1970-08-19	美食;篮球;足球	2018-8-6 9:44
2	2.67456E+17	李四	25	1	河南省	郑州市	郑东新区	18681109672	1980-06-21	音乐;阅读;旅游	2017-4-7 9:14
3	8.92456E+17	王五	24	1	湖北省	武汉市	汉阳区	18798009102	1990-07-20	写代码;读代码;算法	2016-6-8 7:34
4	4.92456E+17	赵六	26	2	陕西省	西安市	莲湖区	18189189195	1987-12-19	购物;旅游	2016-1-9 19:15
5	3.92456E+17	张三	20	0	北京市	昌平区	回龙观	18589407692	1970-08-19	美食;篮球;足球	2018-8-6 9:44
6	3.92456E+17	张三	20	0	北京市	昌平区	回龙观	18589407692	1970-08-19	美食;篮球;足球	2018-8-6 9:44

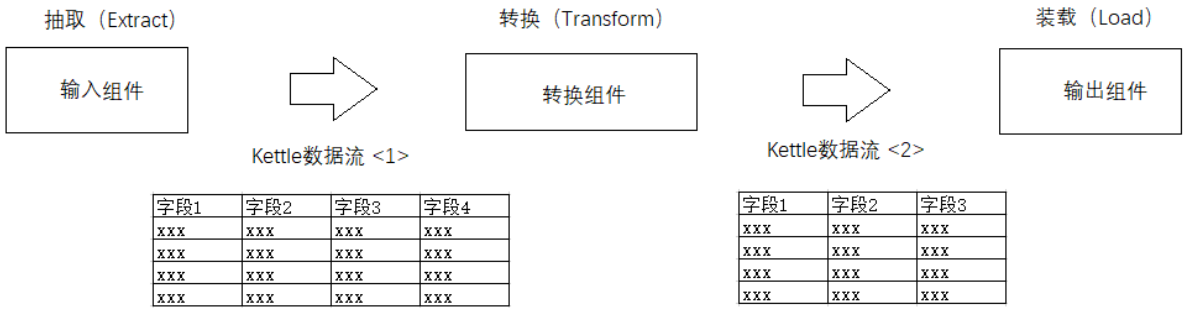
4、配置Excel组件

- 指定输出Excel文件的位置



5、点击 三角形 箭头执行

Kettle数据流结构图



字段1	字段2	字段3	字段4
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx

字段1	字段2	字段3
xxx	xxx	xxx
xxx	xxx	xxx
xxx	xxx	xxx
xxx	xxx	xxx

Kettle输入/输出组件

输入组件

JSON数据文件输入

需求：

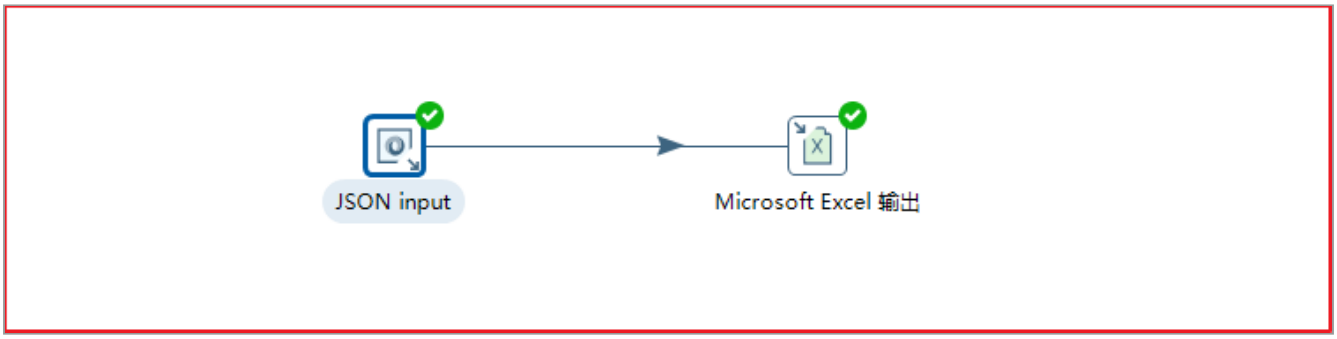
- 将 `资料\kettle测试数据\用户数据源\user.json` 数据文件，通过Kettle，抽取到Excel中


```
[
    {
        "id": "392456197008193000",
        "name": "张三",
        "age": 20,
        "gender": 0,
        "province": "北京市",
        "city": "昌平区",
        "region": "回龙观",
        "phone": "18589407692",
        "birthday": "1970-08-19",
        "hobby": "美食;篮球;足球",
        "register_date": "2018-08-06 09:44:43"
    },
    {
        "id": "267456198006210000",
        "name": "李四",
        "age": 25,
        "gender": 1,
        "province": "河南省",
        "city": "郑州市",
        "region": "郑东新区",
        "phone": "18681109672",
        "birthday": "1980-06-21",
        "hobby": "音乐;阅读;旅游",
        "register_date": "2017-04-07 09:14:13"
    },
```

操作步骤：

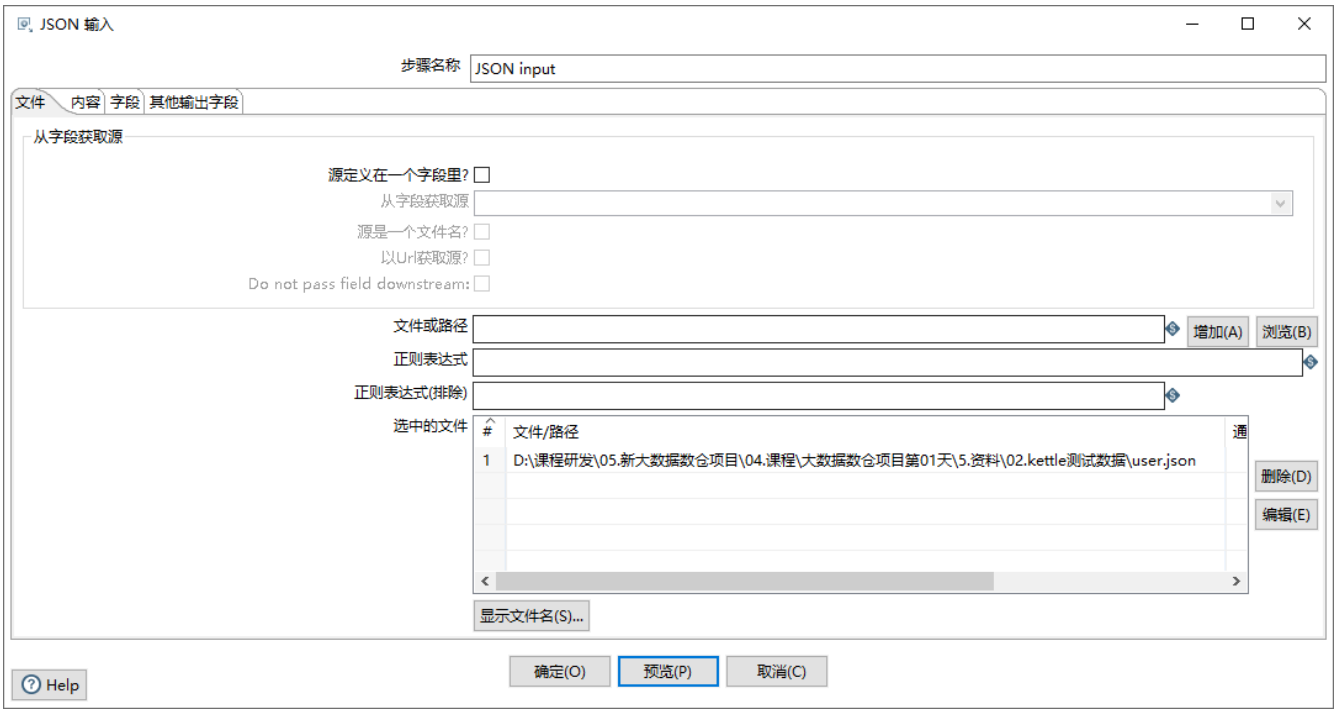
1、新建转换

2、拽入 JSON input组件、Microsoft Excel输出组件、并连接两个组件

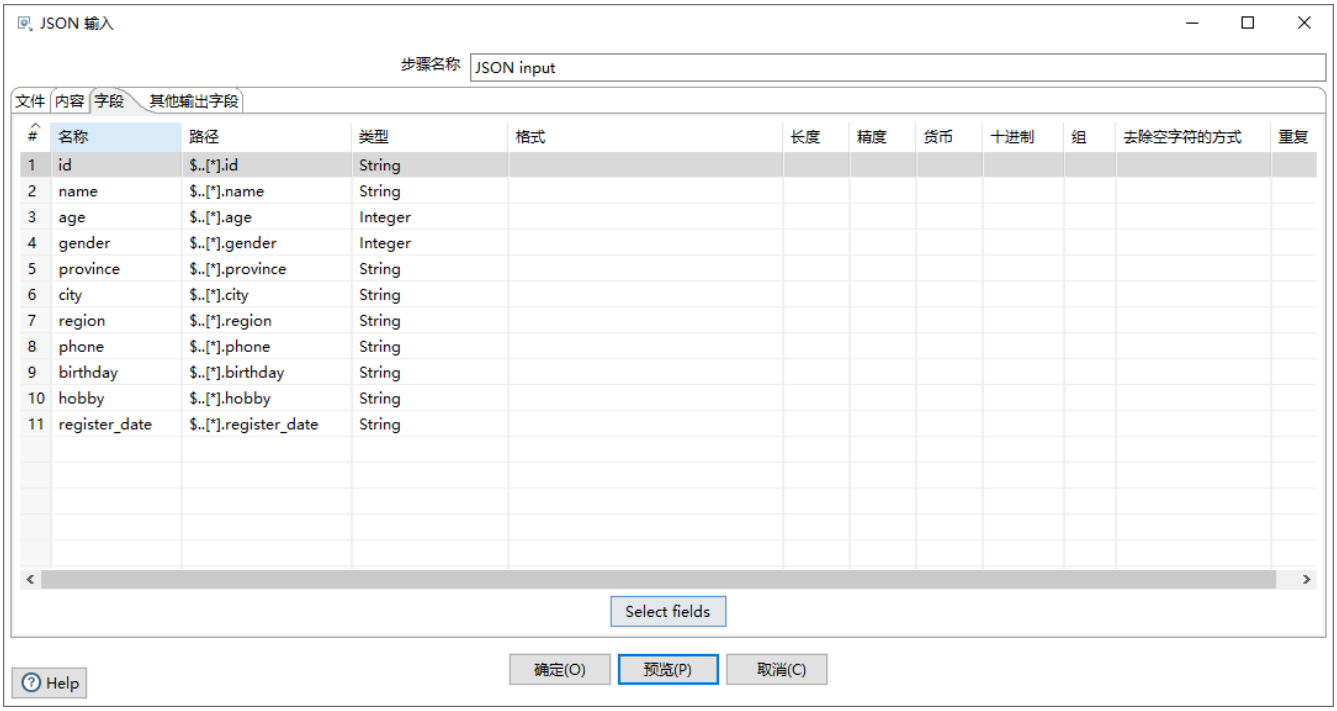


3、配置 JSON input 组件

① 指定JSON文件数据源

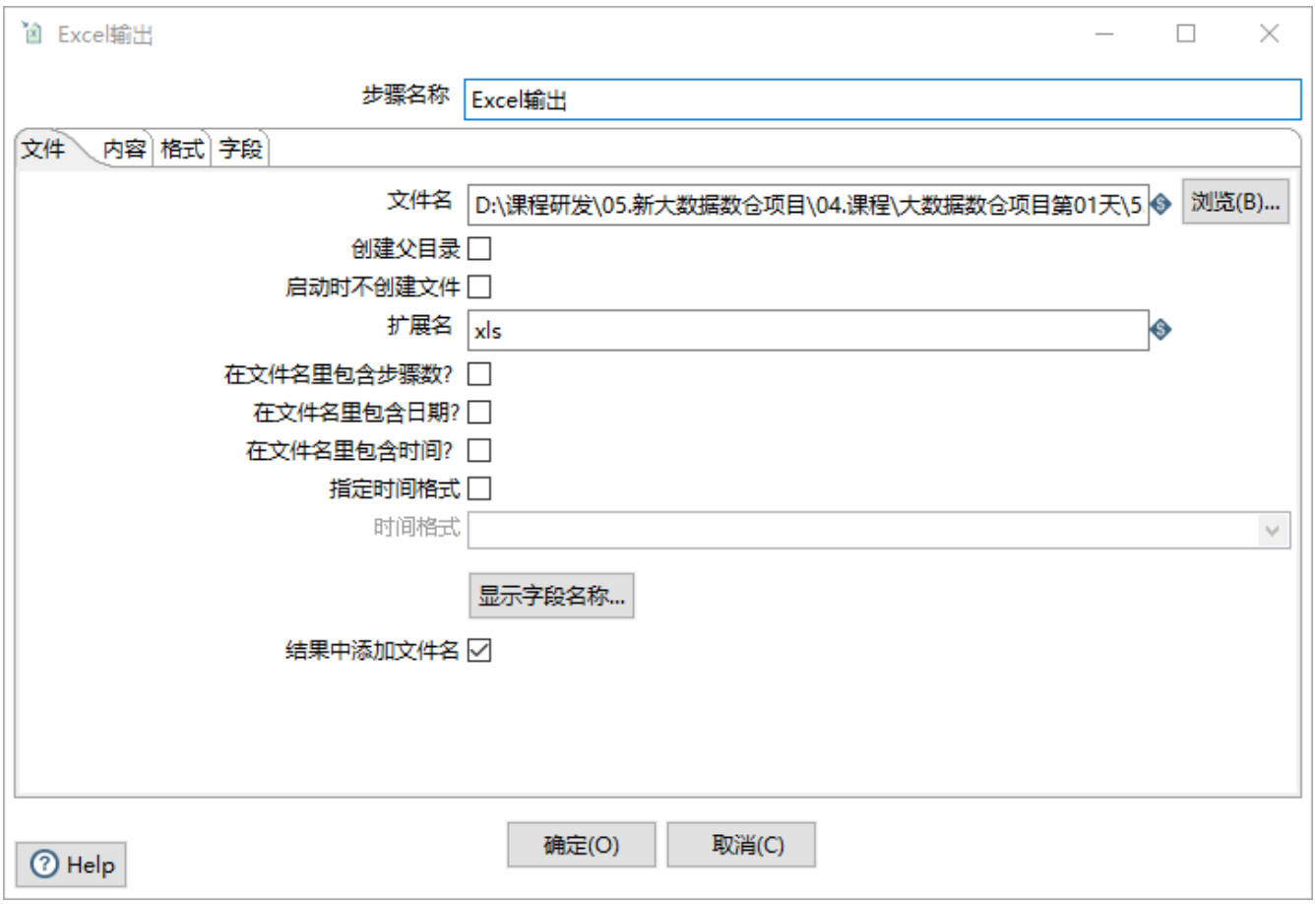


② 选择 JSON 字段



4、配置 Excel 输出 组件

- 指定Excel文件输出位置



5、启动执行

表输入

需求：

- 将MySQL数据库中的 user 表中的数据抽取到Excel文件中

环境准备：

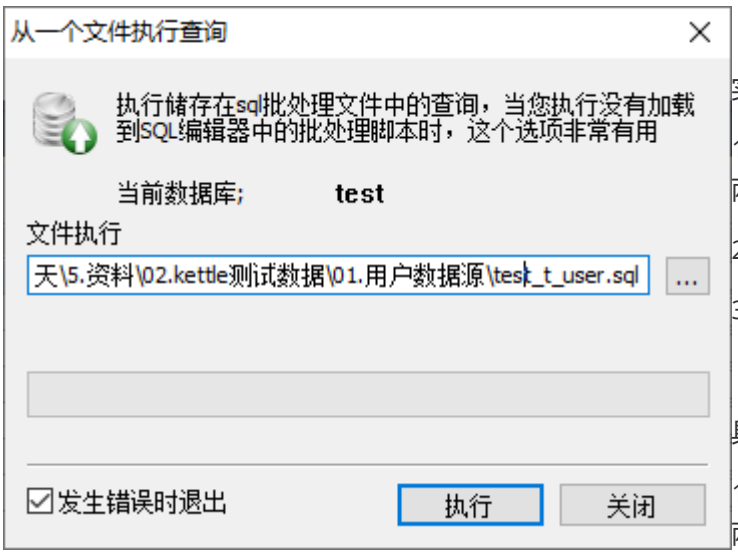
一、Kettle整合MySQL数据库

1、将资料中的 MySQL jdbc 驱动包导入到 pdi-ce-8.2.0.0-342\data-integration\lib 中

2、重启 Kettle

二、MySQL建库

1、导入 `资料\kettle测试数据\用户数据源\test_t_user.sql` 到 MySQL数据库中



实现步骤：

1、拉动 输入/表输入 、输出/Excel输出 组件、连接两个组件

2、配置表输入

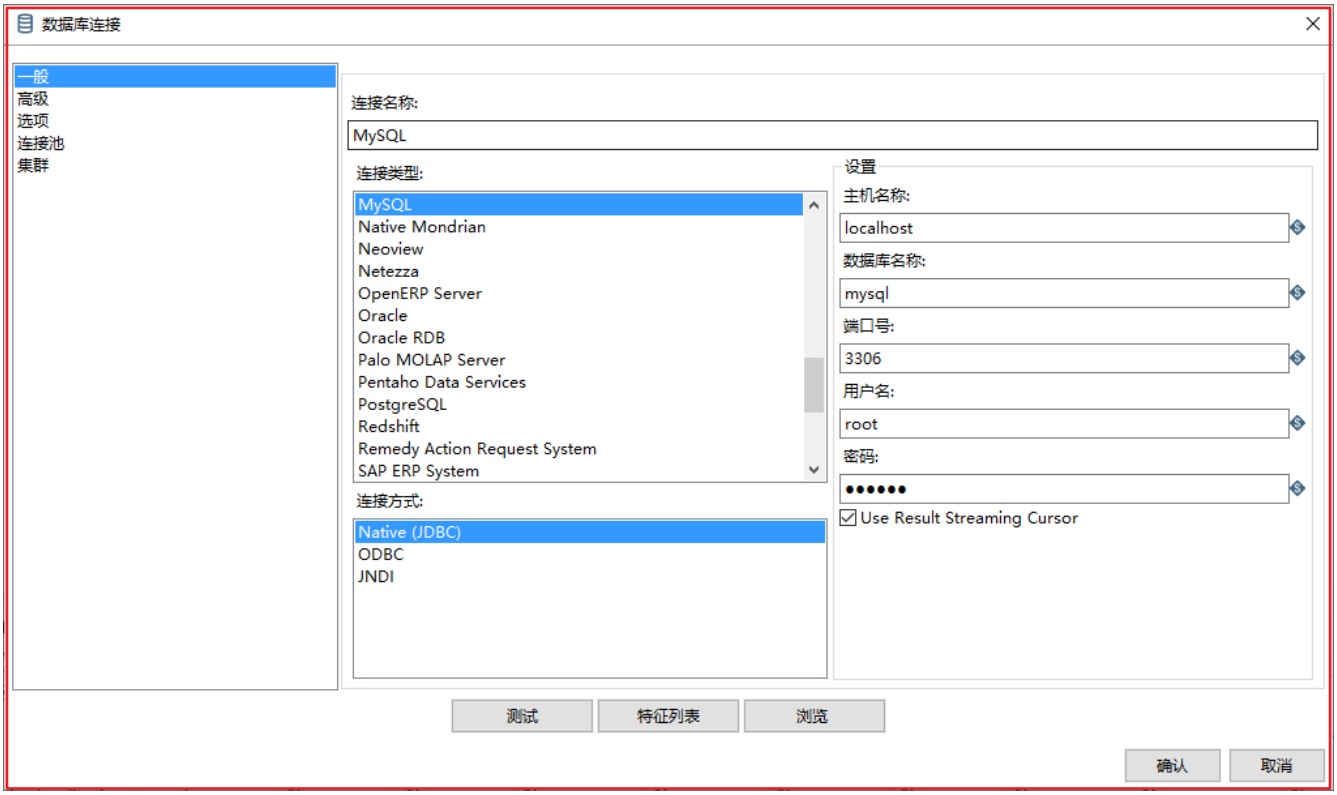
3、配置Excel输出组件

具体操作：

1、拉动 输入/表输入 、输出/Excel输出 组件、连接两个组件

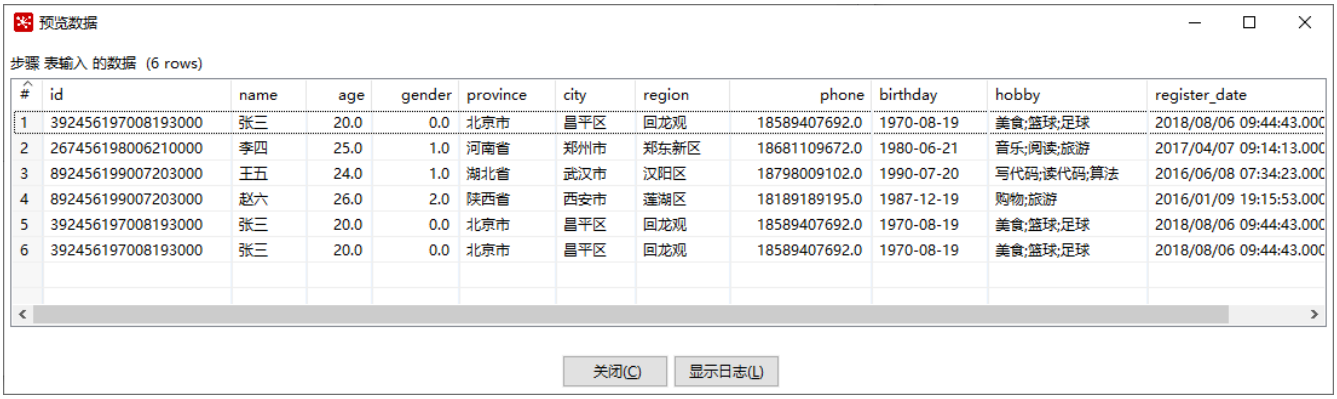
2、配置表输入

2.1 新建数据库连接



2.2 选择 t_user 表，并获取SQL查询语句

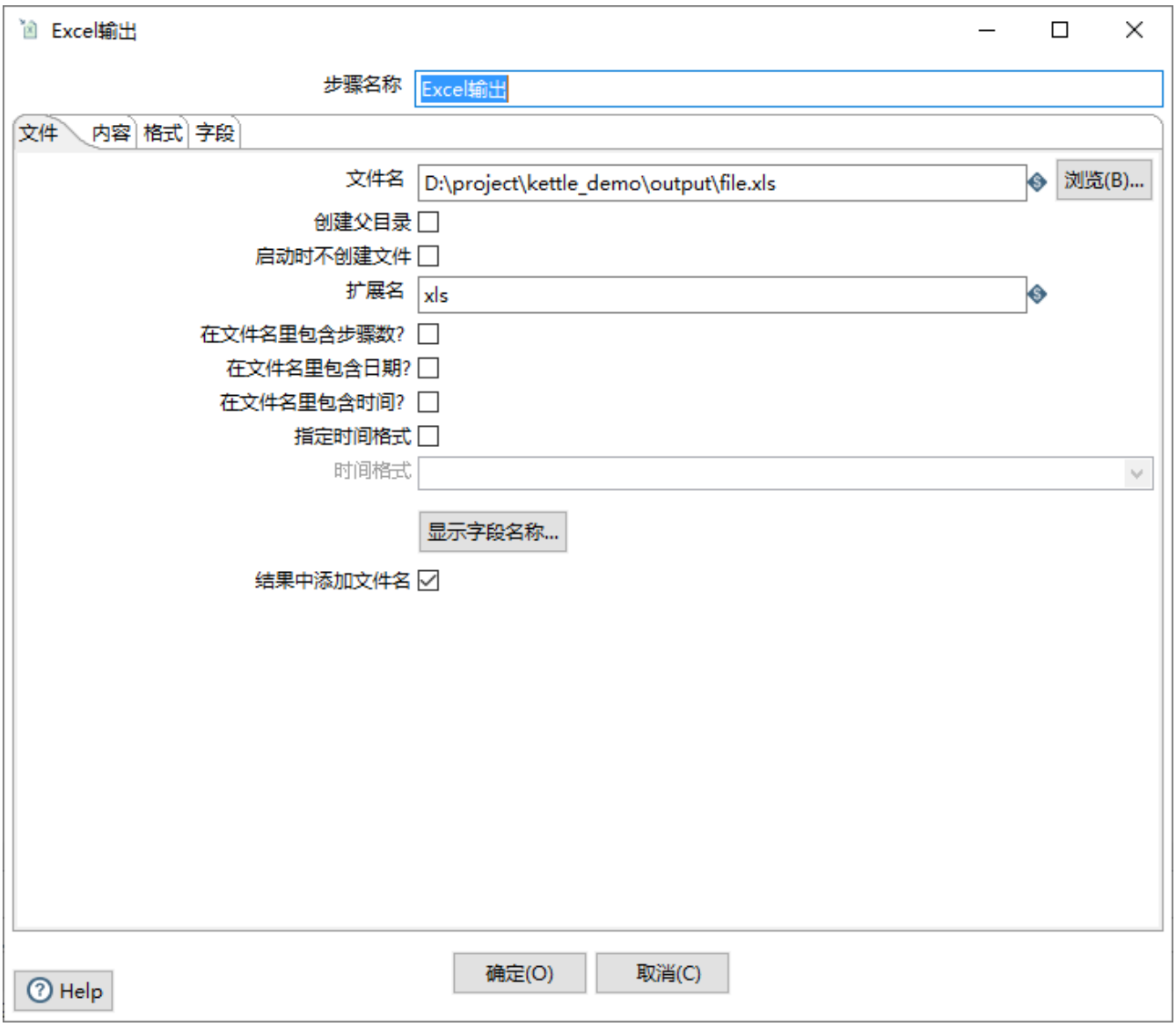
2.3 预览数据



#	id	name	age	gender	province	city	region	phone	birthday	hobby	register_date
1	3924561970081930000	张三	20.0	0.0	北京市	昌平区	回龙观	18589407692.0	1970-08-19	美食;篮球;足球	2018/08/06 09:44:43.000
2	267456198006210000	李四	25.0	1.0	河南省	郑州市	郑东新区	18681109672.0	1980-06-21	音乐;阅读;旅游	2017/04/07 09:14:13.000
3	892456199007203000	王五	24.0	1.0	湖北省	武汉市	汉阳区	18798009102.0	1990-07-20	写代码;读代码;算法	2016/06/08 07:34:23.000
4	892456199007203000	赵六	26.0	2.0	陕西省	西安市	莲湖区	18189189195.0	1987-12-19	购物;旅游	2016/01/09 19:15:53.000
5	3924561970081930000	张三	20.0	0.0	北京市	昌平区	回龙观	18589407692.0	1970-08-19	美食;篮球;足球	2018/08/06 09:44:43.000
6	3924561970081930000	张三	20.0	0.0	北京市	昌平区	回龙观	18589407692.0	1970-08-19	美食;篮球;足球	2018/08/06 09:44:43.000

3、配置Excel输出组件

- 指定Excel输出位置



生成记录

数据仓库中绝大多数的数据都是业务系统生成的动态数据，但是其中一部分维度数据不是动态的，比如：日期维度。静态维度数据就可以提前生成。

需求：

- 往 Excel 文件中插入1000条记录：id为1，name为itheima，age为18

操作步骤：

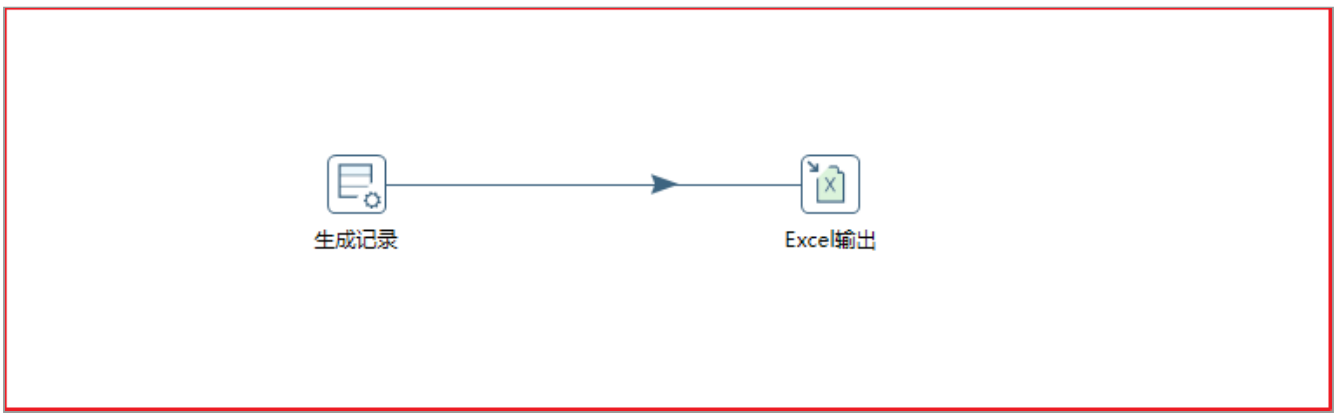
1、拖入 输入/生成记录 组件、输出/Excel输出 组件、连接两个组件

2、配置生成记录组件

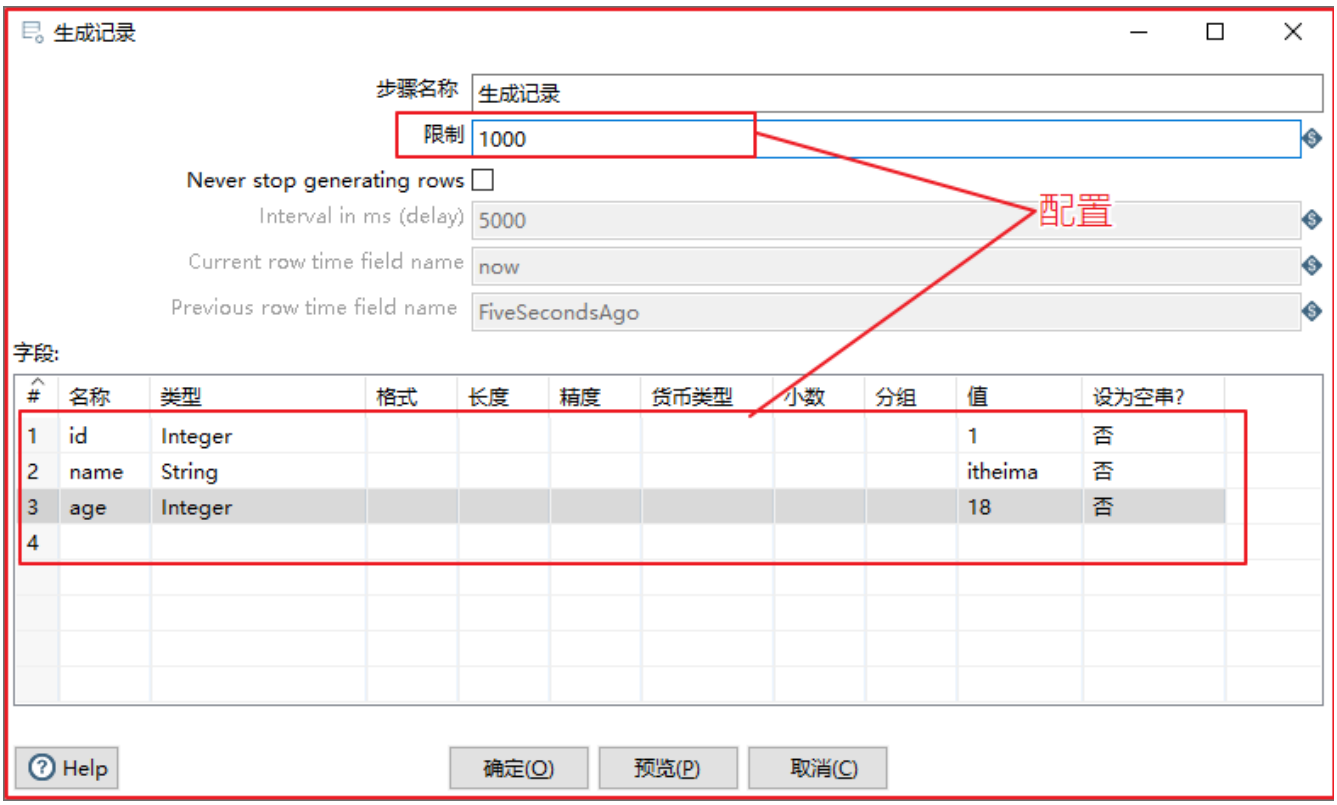
3、配置Excel输出

具体实现：

1、拖入 输入/生成记录 组件、输出/Excel输出 组件、连接两个组件



2、配置生成记录组件



输出组件

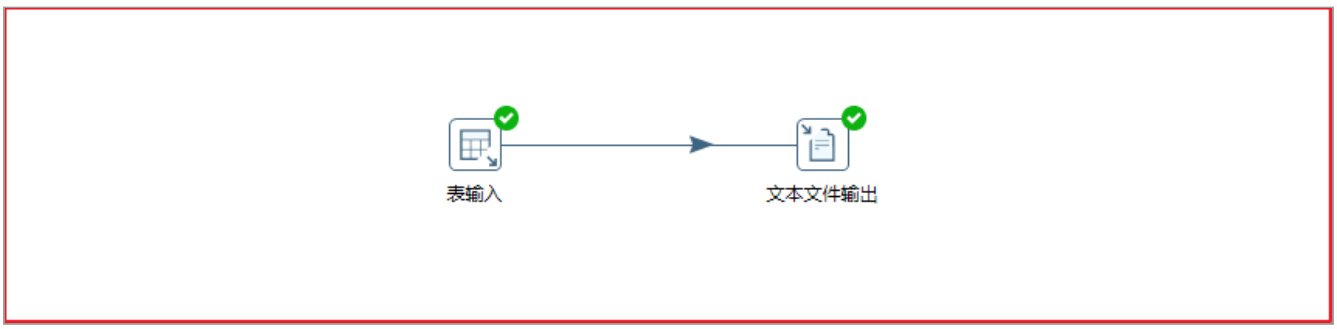
文本文件输出

需求：

- 从mysql数据库的test库的t_user表 抽取数据到文本文件中

步骤：

1、拖入 一个 输入/表输入、一个 输出/文本文件输出、并连接两个组件



2、指定 从哪个表中获取数据

3、指定表中的数据输出到哪个文件

表输出

- Json输出就是把数据写入指定的表

需求：

- 从 `资料\kettle测试数据\用户数据源\user.json` 中读取id, name, age字段的数据，
- 装载到mysql数据库的 t_user_1 表中

操作步骤：

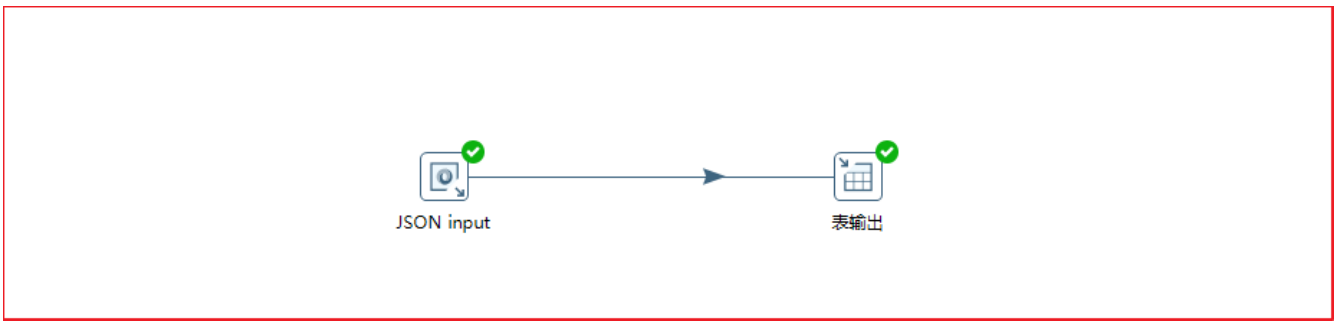
1、拖动 输入/JSON Input组件 ， 输出/表输出，连接两个组件

2、JSON输入配置

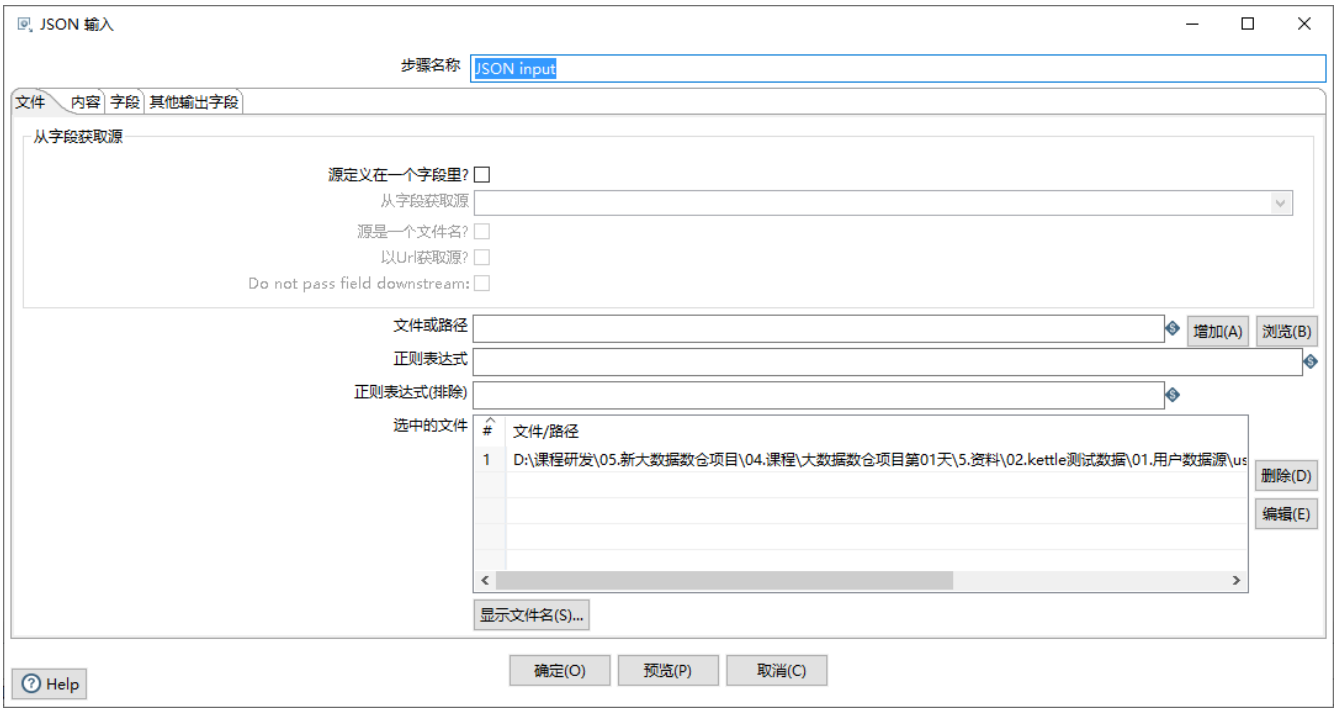
3、表输出配置

具体操作：

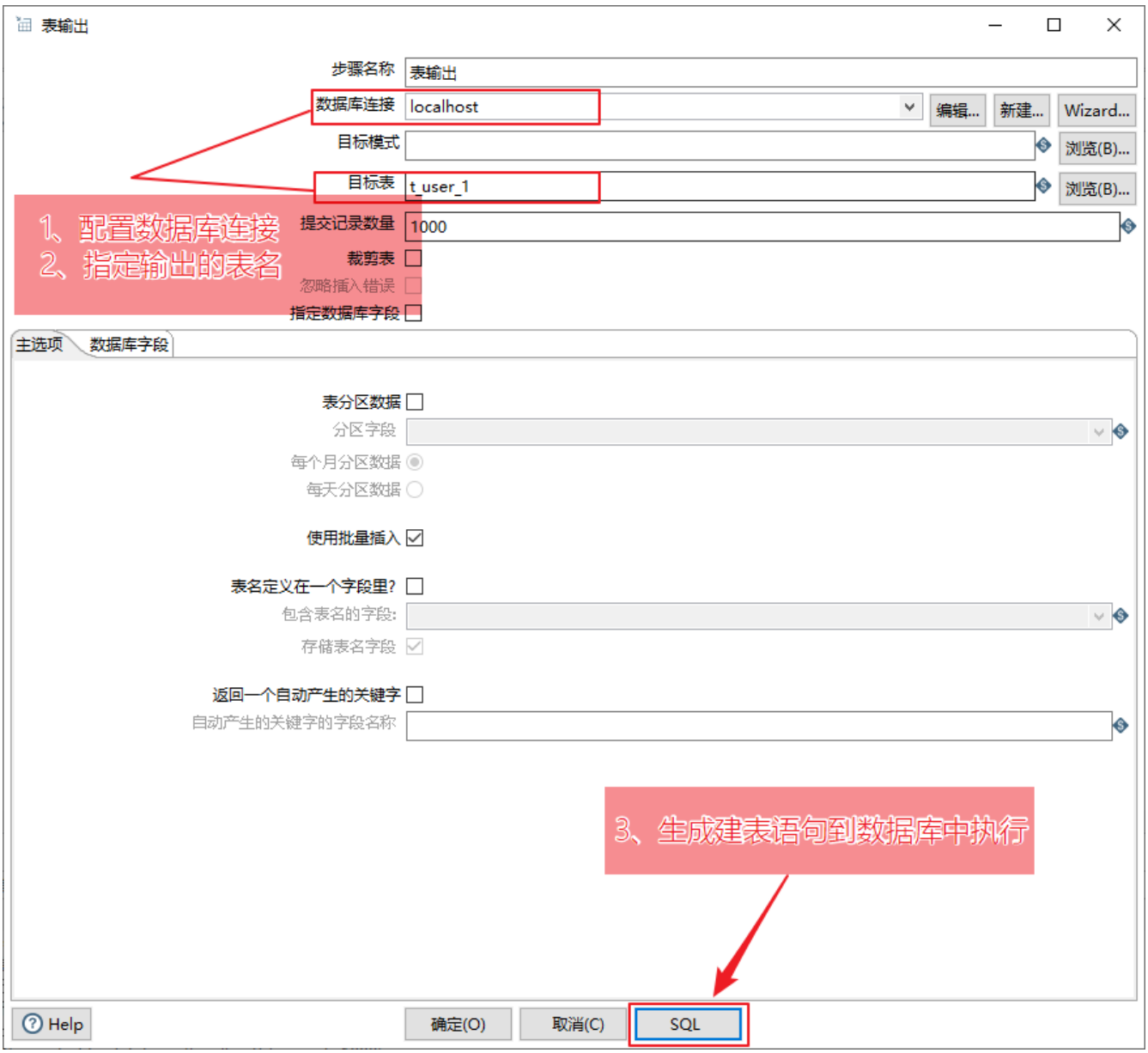
1、拖动 输入/JSON Input组件 ， 输出/表输出，连接两个组件



2、JSON输入配置



3、表输出配置



插入更新

- 插入更新就是把数据库已经存在的记录与数据流里面的记录进行比对
  - 如果不同就进行更新
  - 如果记录不存在，则会插入数据

需求：

- 从 `资料\kettle测试数据\user_new.json` 中读数据，并插入或更新到mysql数据库的 t_user_1 表中

操作步骤：

1、拖入一个 输入/JSON输入组件，一个 输出/插入更新组件、连接两个组件

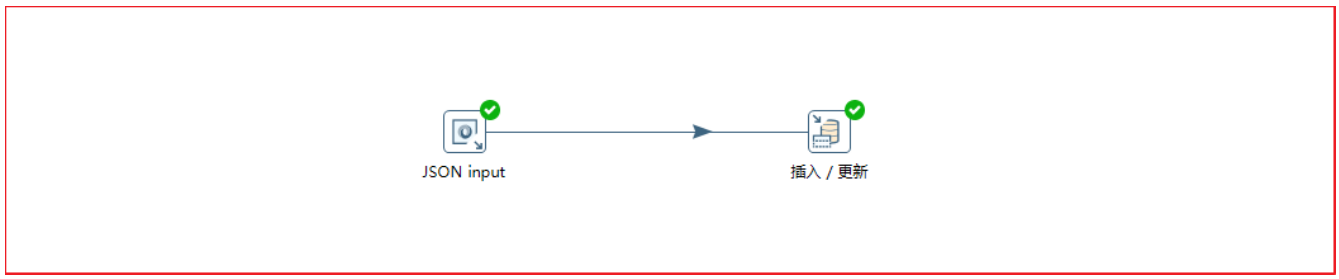
2、配置 JSON输入组件

3、配置 插入更新 组件

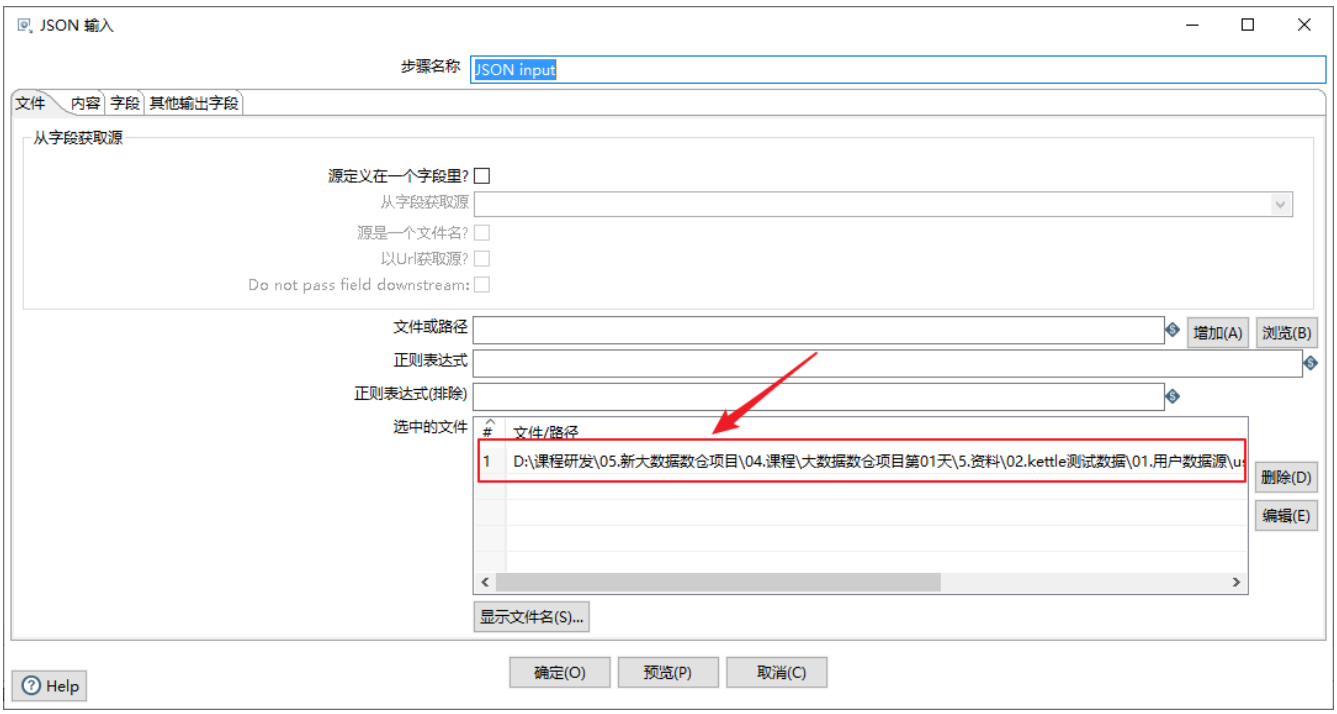
4、启动执行

具体实现：

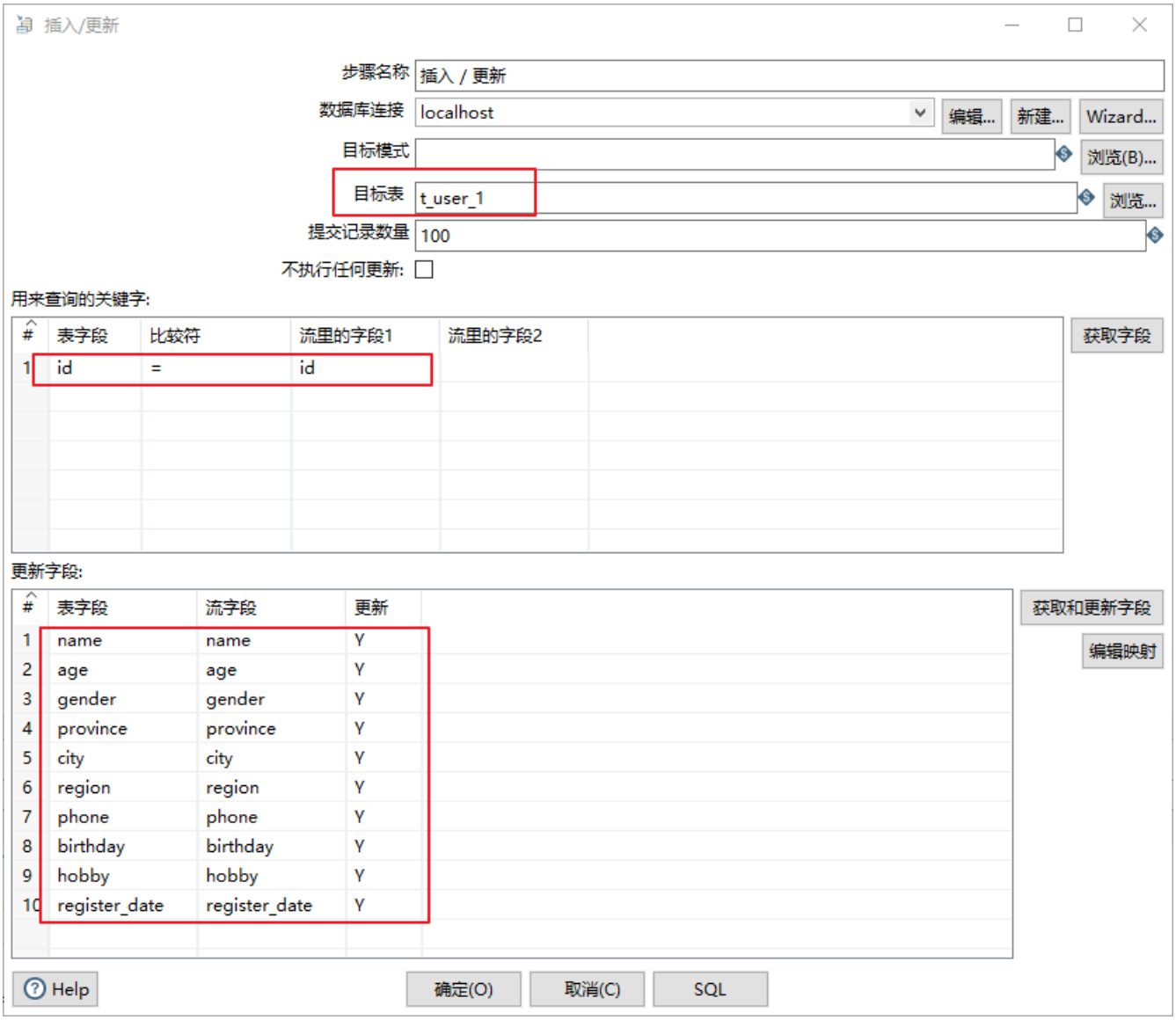
1、拖入一个 输入/JSON输入组件，一个 输出/插入更新组件、连接两个组件



2、配置 JSON输入组件



3、配置 插入更新 组件



4、启动执行

删除

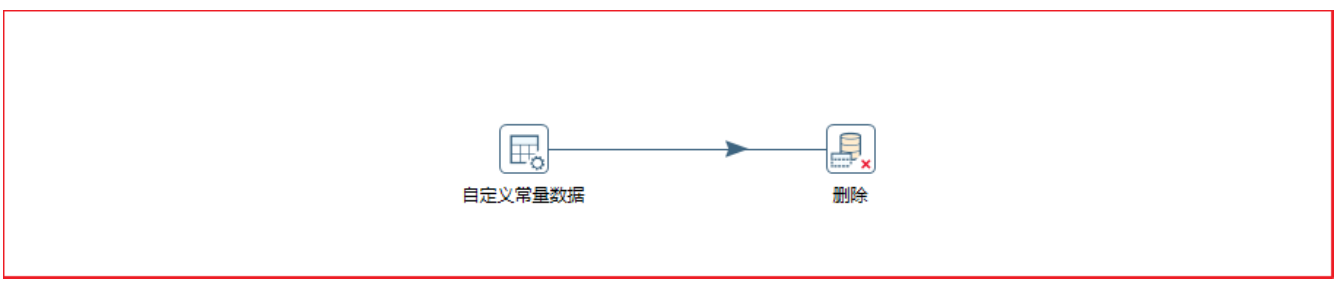
需求：

- 从mysql数据库 t_user_1 表中删除指定id为 492456198712198000 的数据

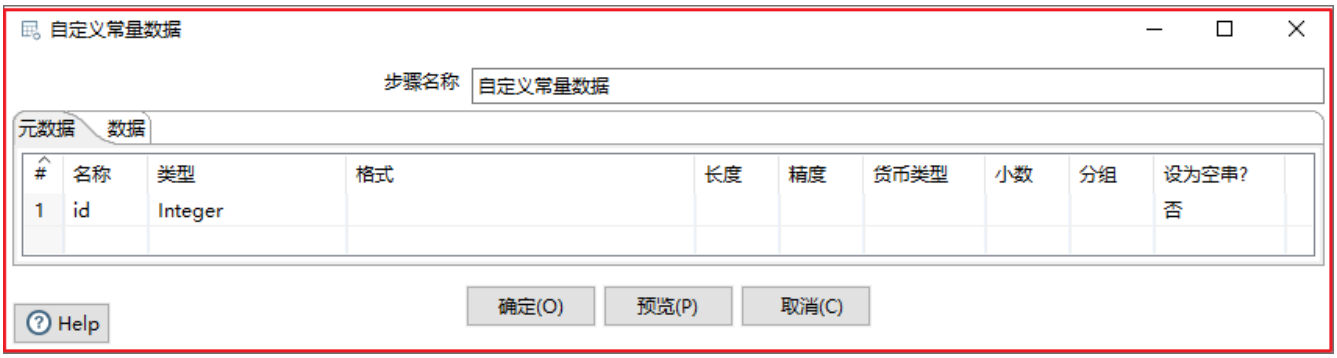
操作步骤：

1、拖入一个 输入/自定义常量数据、输出/删除 组件

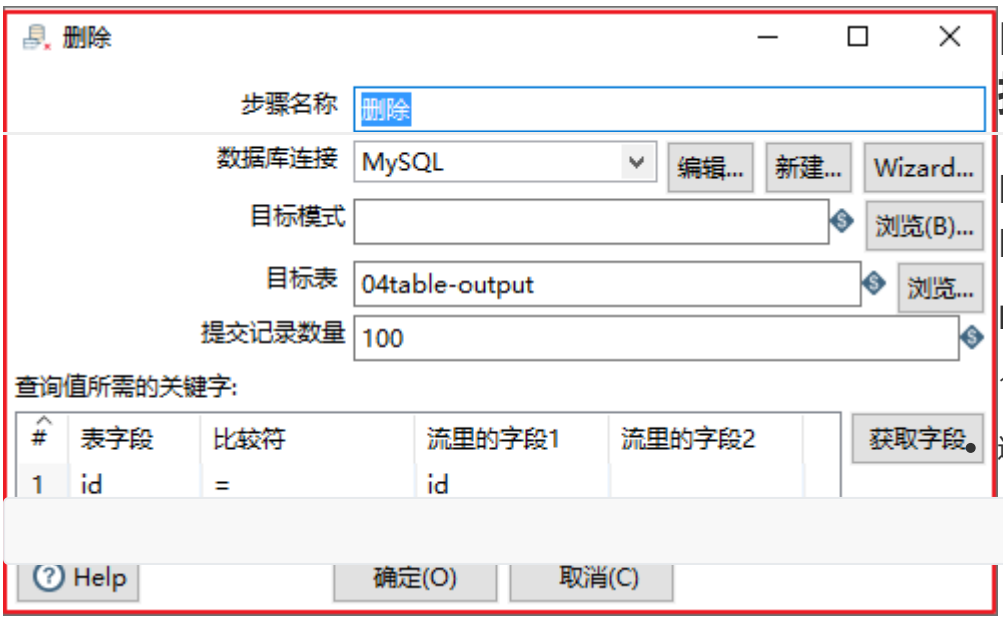
2、连接两个组件



3、配置自定义常量数据组件



4、配置删除组件



Kettle整合大数据平台

Kettle整合Hadoop

Hadoop环境准备

1、查看hadoop的文件系统

通过浏览器访问

```
http://node1:50070/
```

- 通过终端访问

```
hadoop fs -ls / # 查看文件
```

2、在hadoop文件系统中创建/hadoop/test目录

```
hadoop fs -mkdir -p /hadoop/test
```

3、在本地创建1.txt

- vim 1.txt

```
id,name
1,itheima
2,itcast
```

4、上传1.txt到hadoop文件系统的/hadoop/test目录

```
hadoop fs -put 1.txt /hadoop/test
```

kettle与hahoop环境整合

1、确保Hadoop的环境变量设置好HADOOP_USER_NAME为root

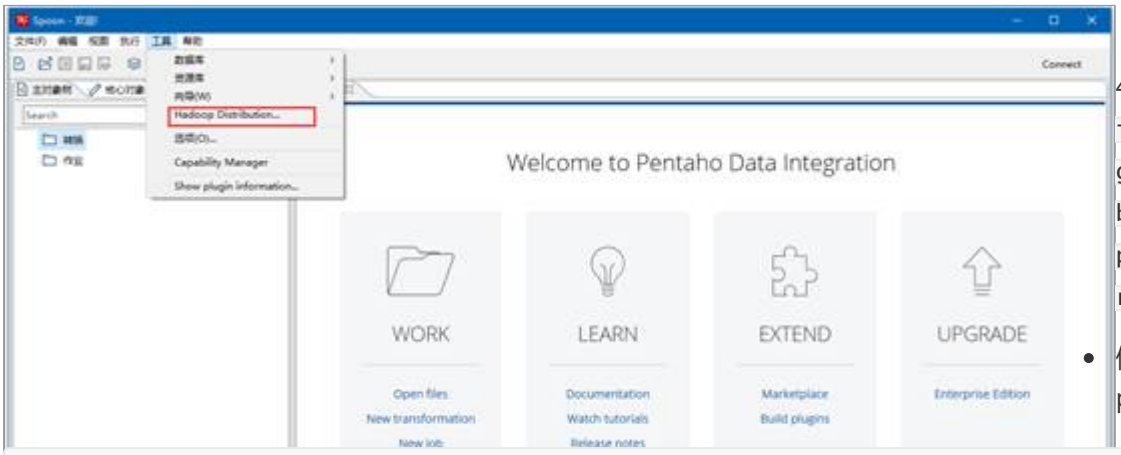
2、从hadoop下载核心配置文件

```
sz /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop/hdfs-site.xml
sz /export/servers/hadoop-2.6.0-cdh5.14.0/etc/hadoop/core-site.xml
```

文件会被下载到windows的下载目录

3、把hadoop核心配置文件放入kettle目录

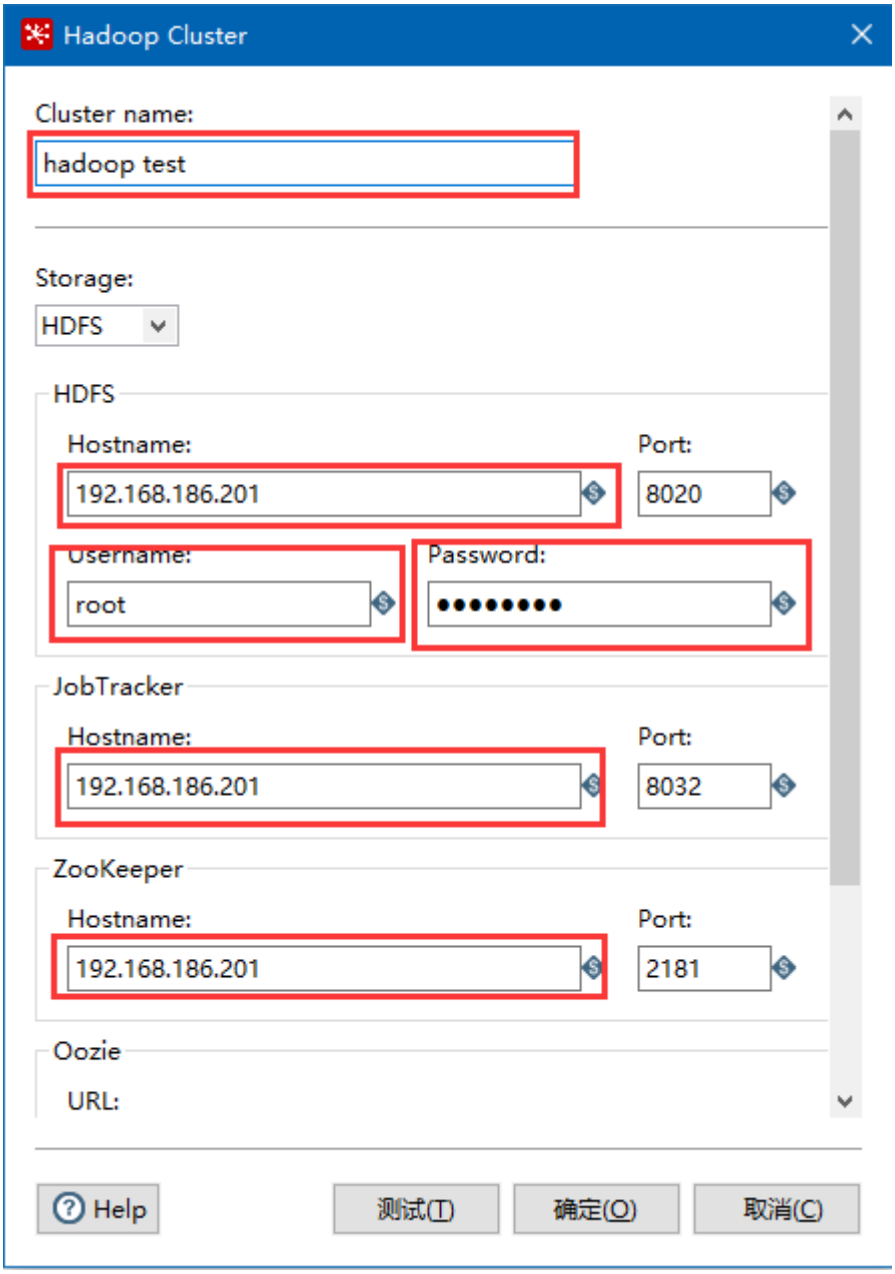
```
data-integration\plugins\pentaho-big-data-plugin\hadoop-configurations\cdh514
```

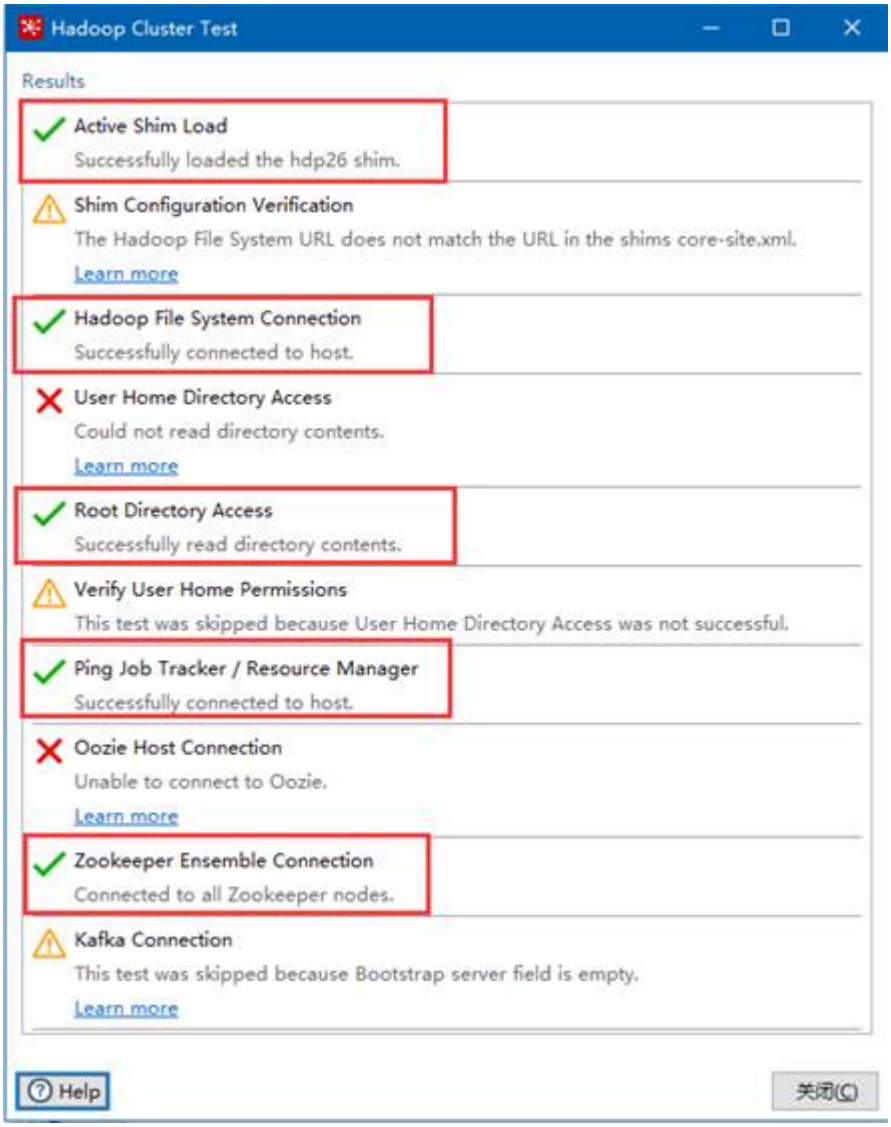


- 4、修改 `data-integration\plugins\pentaho-big-data-plugin\plugin.properties` 文件
- 修改 plugin.properties

```
active.hadoop.configuration=cdh514
```

5、 创建Hadoop clusters


Hadoop Cluster
Cluster name:
hadoop test
Storage:
HDFS
HDFS
Hostname:
192.168.186.201
Port:
8020
Username:
root
Password:
JobTracker
Hostname:
192.168.186.201
Port:
8032
ZooKeeper
Hostname:
192.168.186.201
Port:
2181
Oozie
URL:
Help
测试(T)
确定(O)
取消(C)



Hadoop file input组件

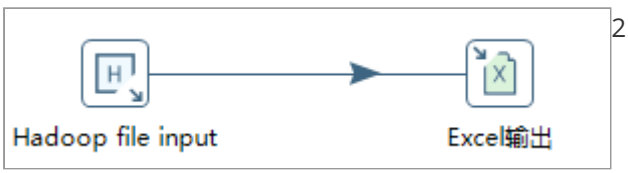
Kettle在Big data分类中提供了一个Hadoop file input 组件用来从hdfs文件系统中读取数据。

需求：

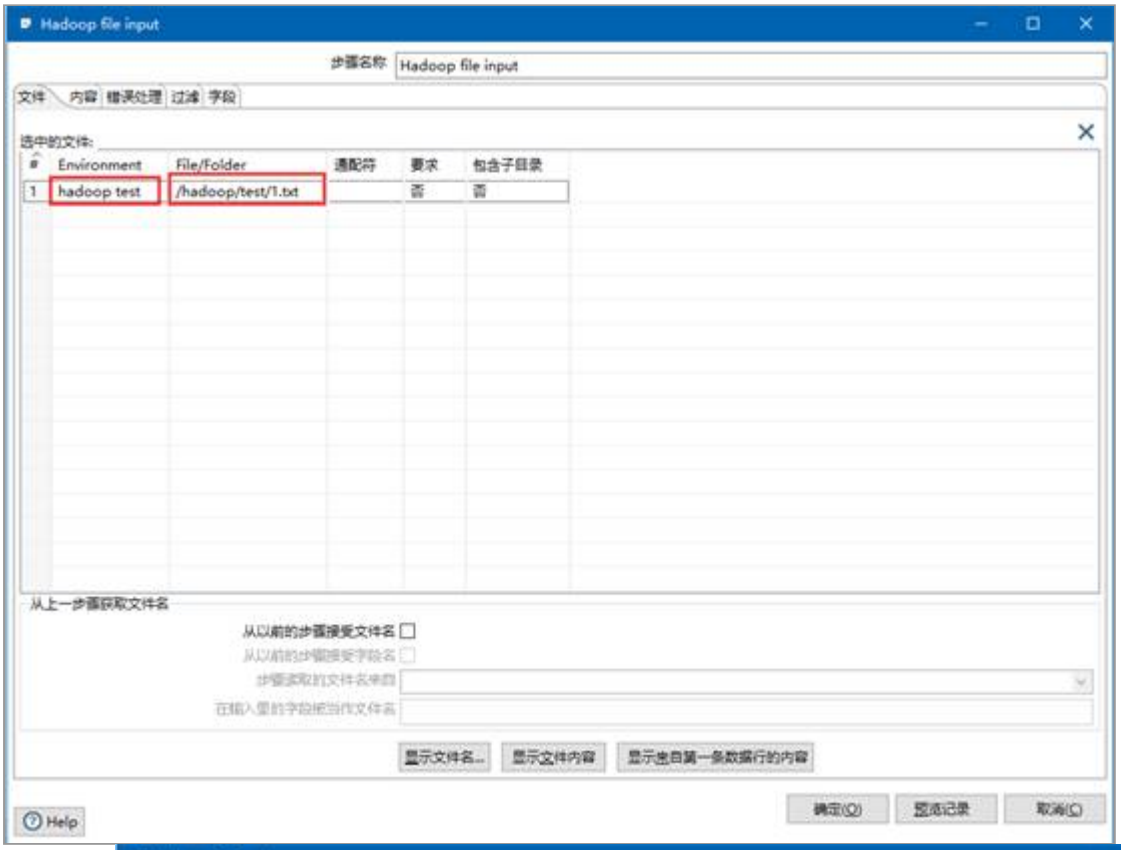
- 从Hadoop文件系统读取/hadoop/test/1.txt文件，把数据输入到Excel中。

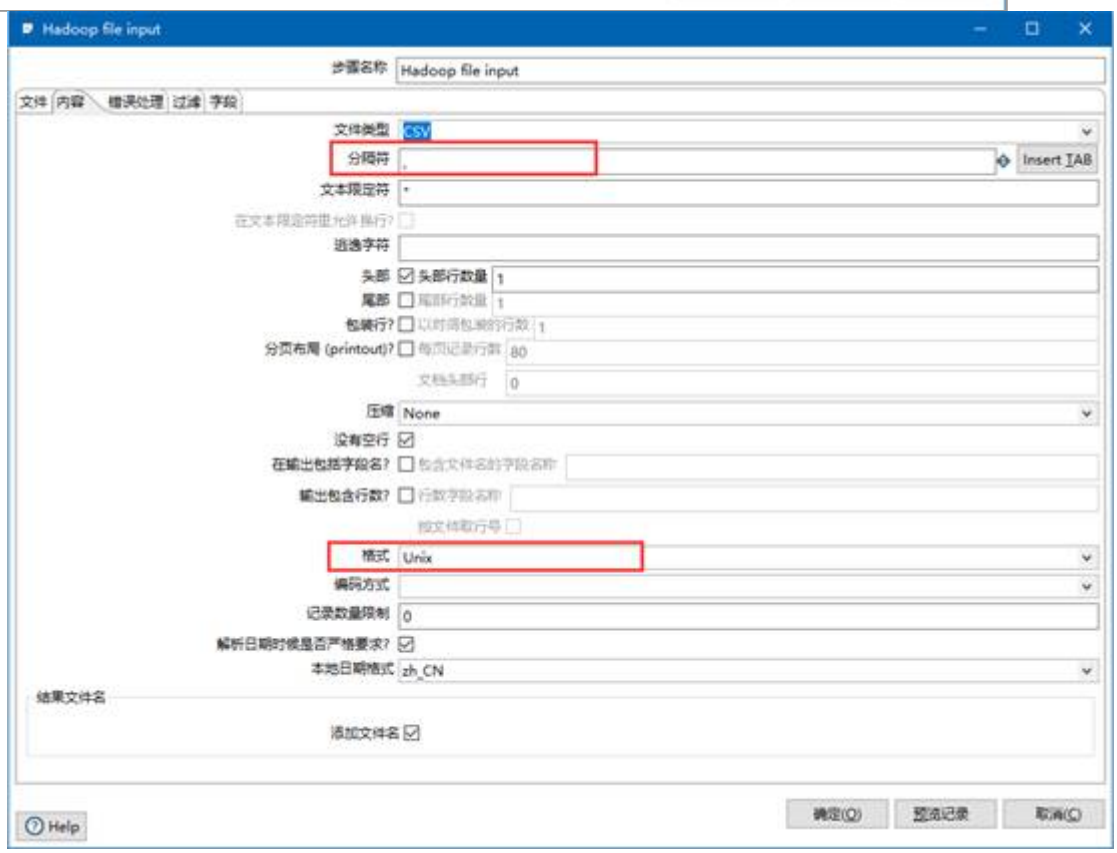
实习步骤：

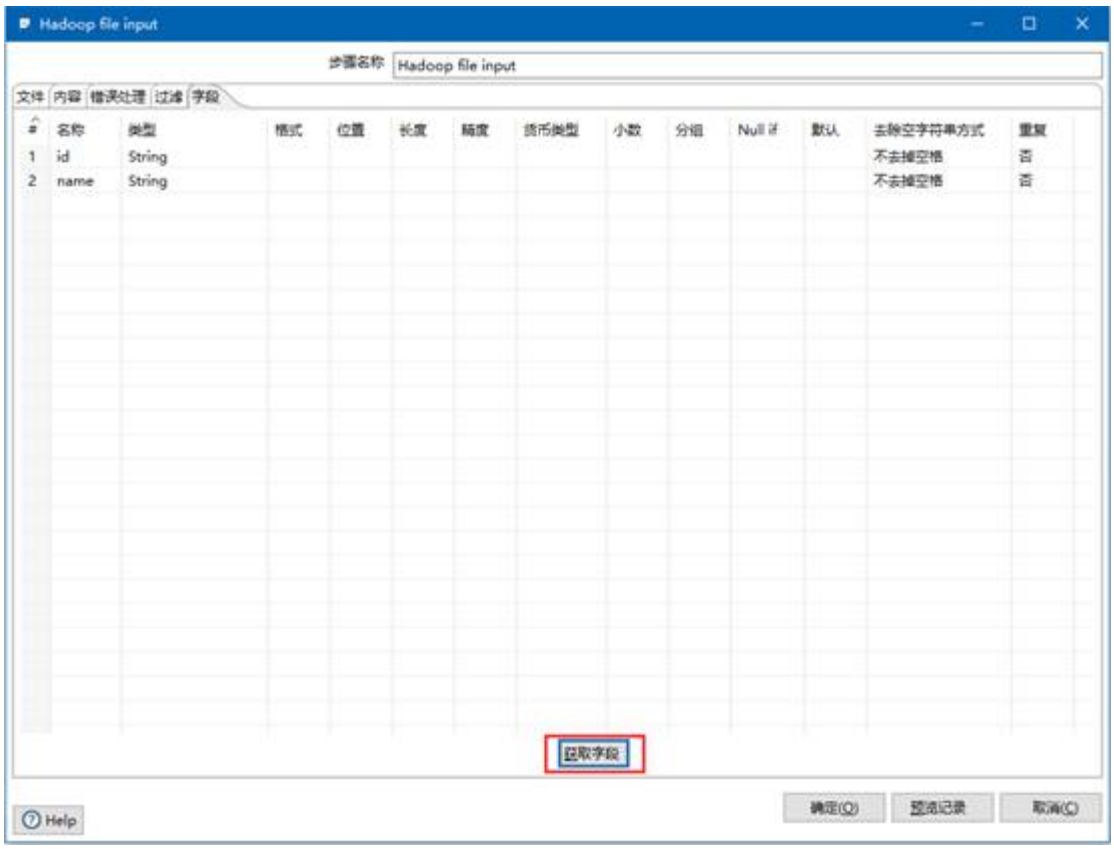
1、拖入以下组件

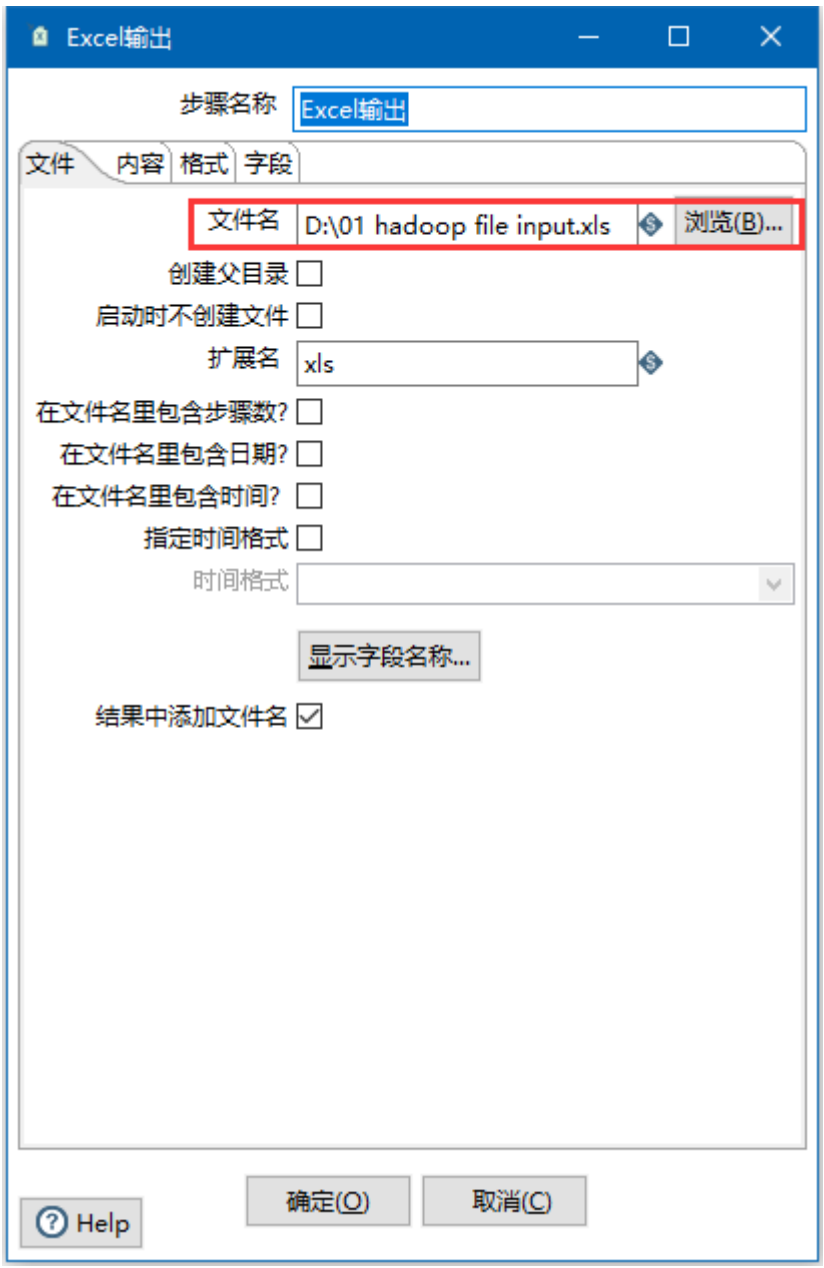


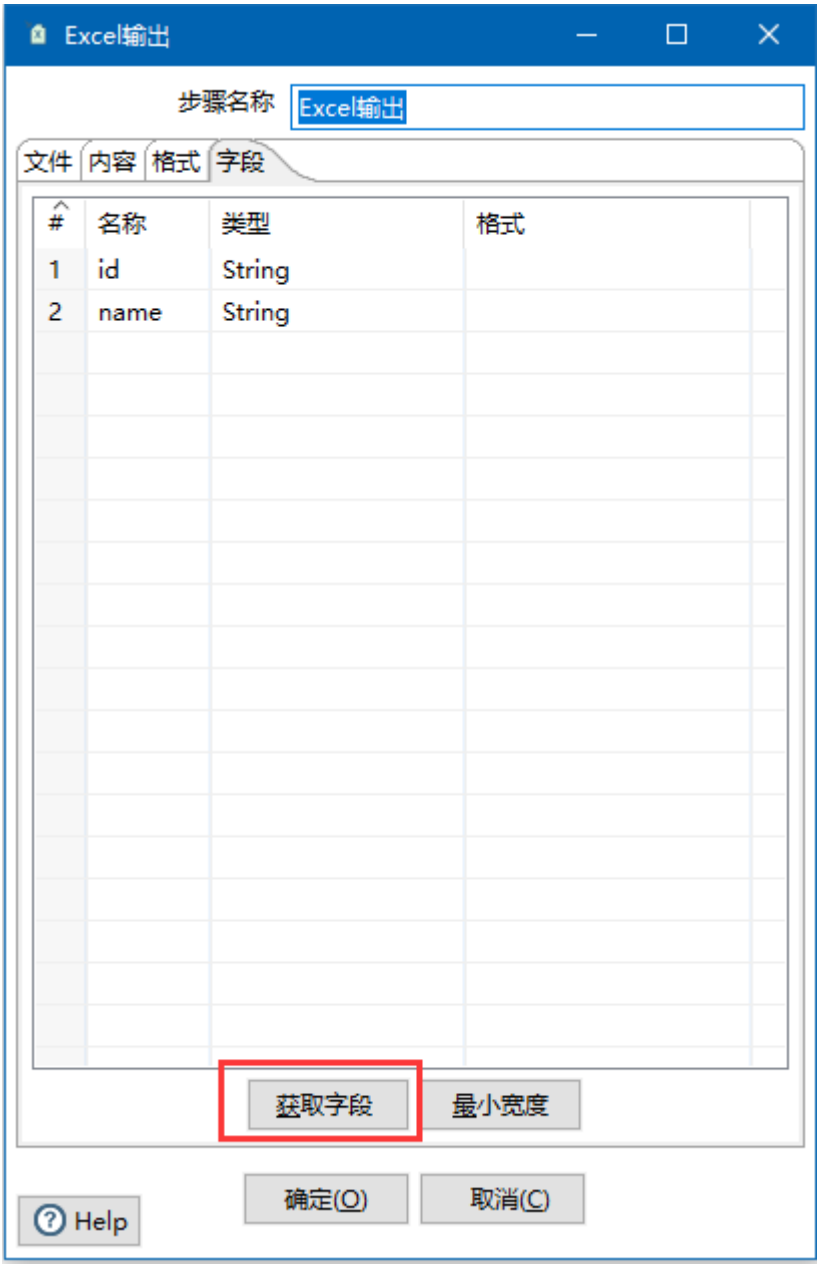
2、配置Hadoop File Input组件


Hadoop file input
步骤名称
Hadoop file input
文件
内容
错误处理
过滤
字段
选中的文件:
#
Environment
File/Folder
通配符
要求
包含子目录
1
hadoop test
/hadoop/test/1.txt
否
否
从上一步骤获取文件名
从以前的步骤接受文件名
显示文件名...
显示文件内容
显示前面第一条数据行的内容
Help
确定(O)
预览记录
取消(C)


Hadoop file input
步骤名称
Hadoop file input
文件
内容
错误处理
过滤
字段
文件类型
CSV
分隔符
,
Insert TAB
文本限定符
"
在文本限定符里允许换行?
逃逸字符
头部
头部行数量
1
尾部
尾部行数量
1
包装行?
以此数目包装的行数
1
分页布局 (printout)?
每页记录行数
80
文档头部行
0
压缩
None
没有空行
在输出包括字段名?
包含文件名的字段名称
输出包含行数?
行数字段名称
按文件取行号
格式
Unix
编码方式
记录数量限制
0
解析日期时候是否严格要求?
本地日期格式
zh_CN
结果文件名
添加文件名
Help
确定(O)
预览记录
取消(C)


Hadoop file input
步骤名称
Hadoop file input
文件
内容
错误处理
过滤
字段
名称 类型 格式 位置 长度 精度 货币类型 小数 分组 Null if 默认 去除空字符串方式 重复
1 id String 不去掉空格 否
2 name String 不去掉空格 否
获取字段
Help
确定(O)
预览记录
取消(C)


Excel输出
步骤名称
Excel输出
文件
内容
格式
字段
文件名
D:\01 hadoop file input.xls
浏览(B)...
创建父目录
启动时不创建文件
扩展名
xls
在文件名里包含步骤数?
在文件名里包含日期?
在文件名里包含时间?
指定时间格式
时间格式
显示字段名称...
结果中添加文件名
Help
确定(O)
取消(C)



Hadoop file output组件

Kettle在Big data分类中提供了一个Hadoop file output 组件用来向hdfs文件系统中保存数据

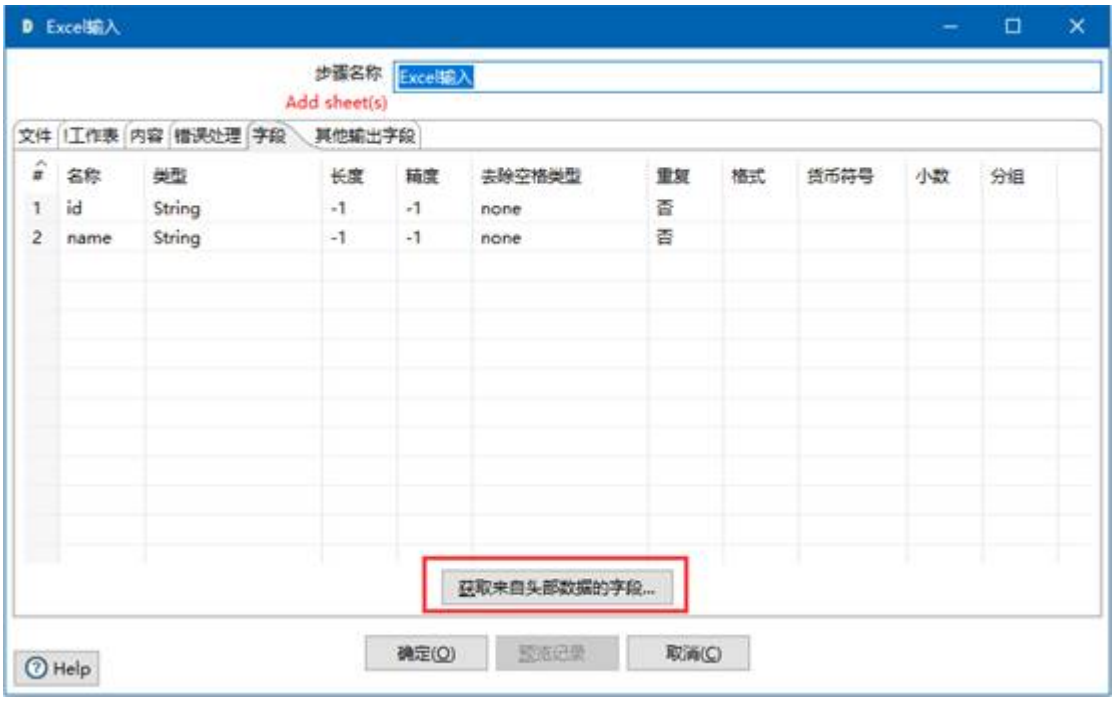
需求:

- 读取 user.json 把数据写入到hdfs文件系统的的/hadoop/test/2.txt中。

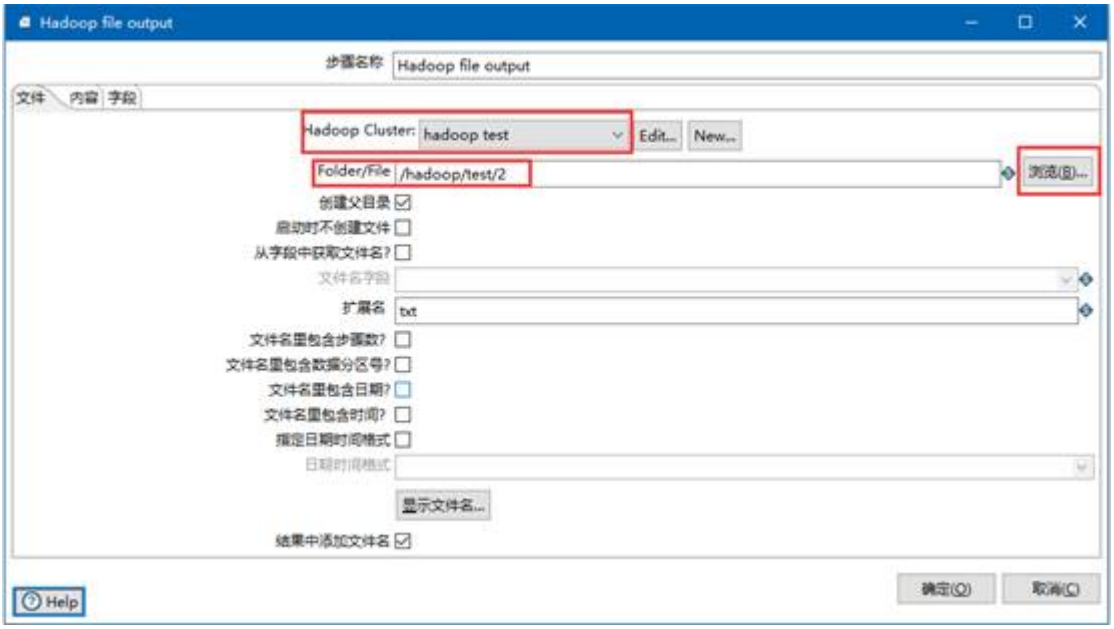
实现步骤:

1、拖入以下组件

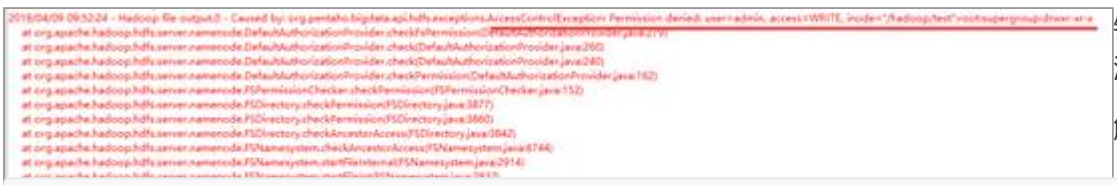
2、配置 JSON 输入组件



3、配置Hadoop file output组件



- 问题



错误：admin用户没有权限

解决：

```
# 添加用户
hadoop fs -mkdir /user/admin
# 权限分配
hadoop fs -chown admin:hdfs /user/admin
# 修改权限
sudo -u hdfs hadoop fs -chmod -R 777  /
```

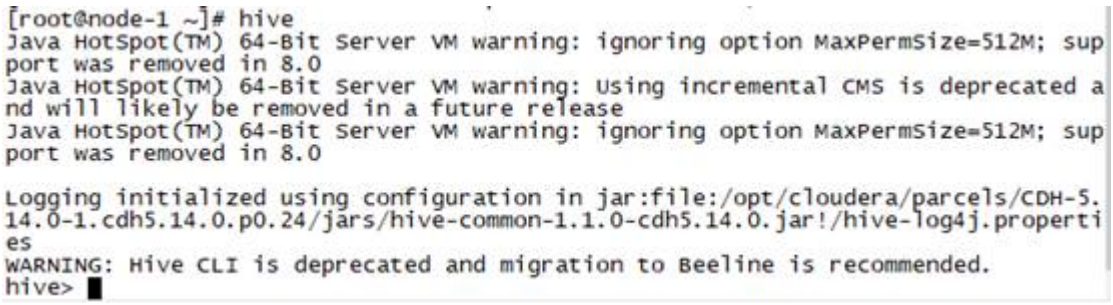
Kettle整合Hive

hive --service hiveserver2 &

hive --service metastore &

初始化数据

1、连接hive



2、创建并切换数据库

```
create database test;
use test;
```

3、创建表

```
create table a(
        a int,
    b int
)
row format delimited fields terminated by ',' stored as TEXTFILE;
show tables;
```

4、创建数据文件

```
vim a.txt
1,11
2,22
3,33
```

5、 从文件加载数据到表

```
load data local inpath '/root/a.txt' into table a;
```

6、查询表

```
select * from a;
```

kettle与Hive整合

1、从虚拟机下载Hadoop的jar包

```
sz /export/servers/hadoop-2.6.0-cdh5.14.0/share/hadoop/common/hadoop-common-2.6.0-
cdh5.14.0.jar
```

2、把jar包放置在\data-integration\lib目录下

3、重启kettle，重新加载生效

从hive中读取数据

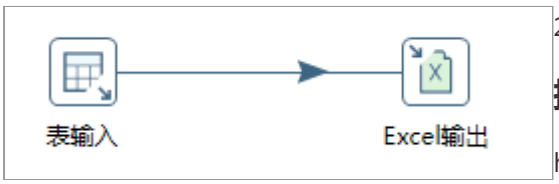
- hive数据库是通过jdbc来进行连接，可以通过表输入控件来获取数据。

需求：

- 从hive数据库的test库的a表中获取数据，并把数据保存到Excel中。

实现步骤：

1、设计一下kettle组件结构



2、配置表输入组件

把数据保存到hive数据库

hive数据库是通过jdbc来进行连接，可以通过表输出控件来保存数据。

需求：

- 从Excel中读取数据，把数据保存在hive数据库的test数据库的a表。

实现步骤：

1、设计如下kettle组件结构

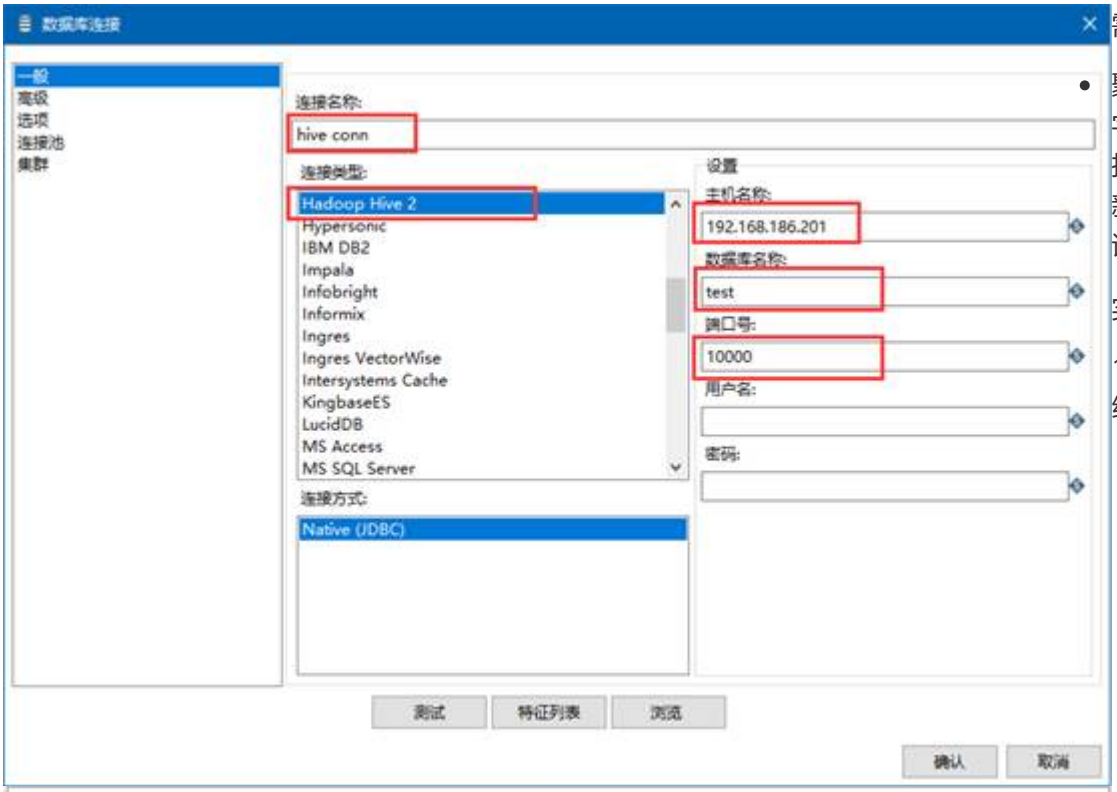
2、配置 Excel输入组件

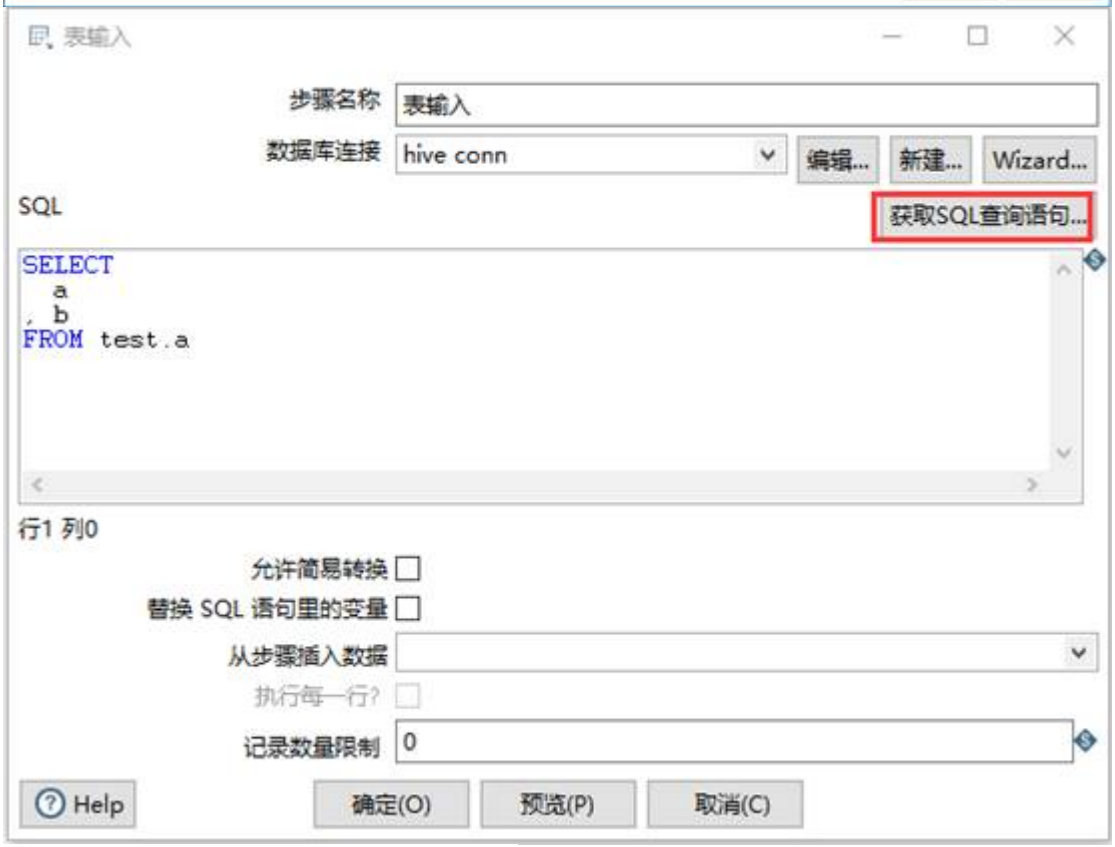
2、配置表输出组件

验证

执行Hive的HiveSQL语句

Kettle中可以执行Hive的HiveSQL语句，使用作业的SQL脚本。





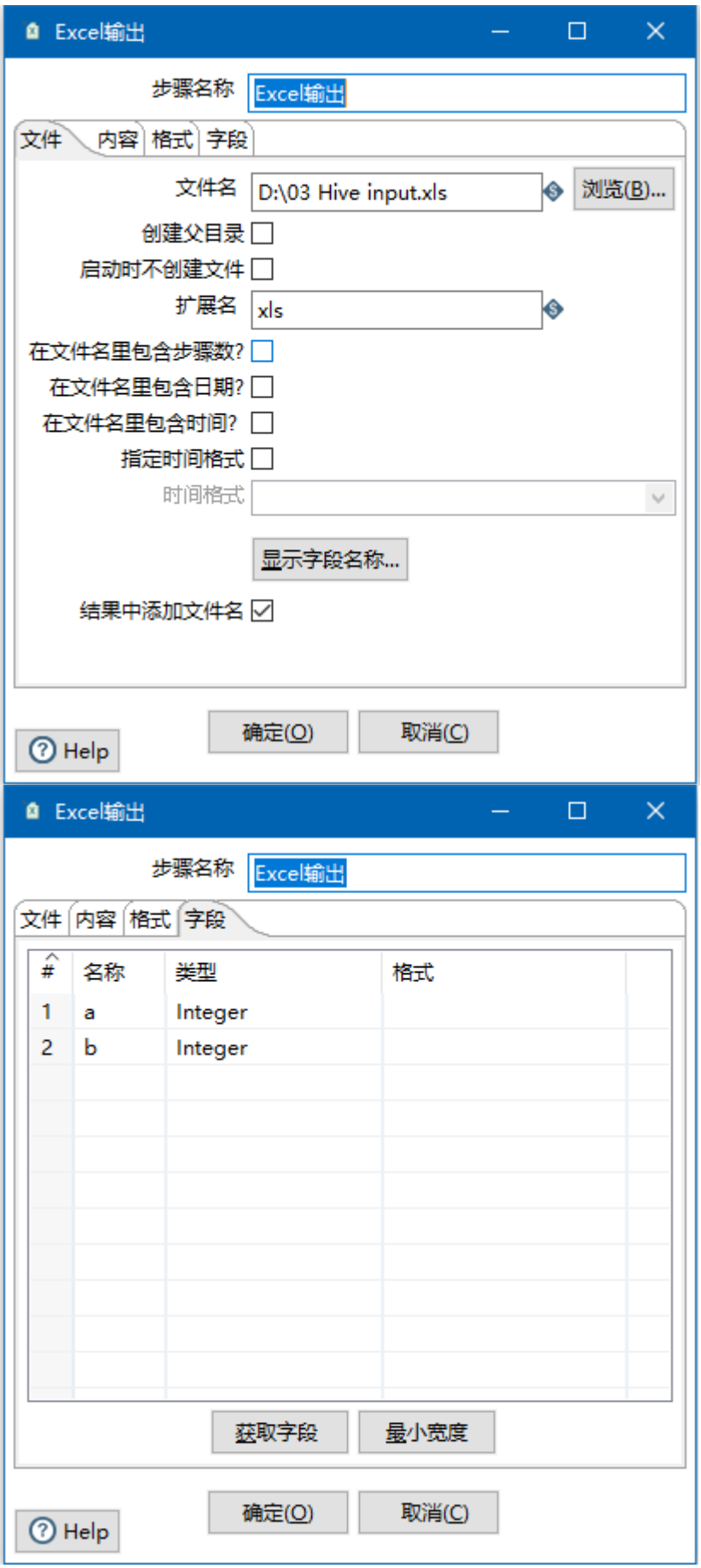
2、配置SQL组件

需求：

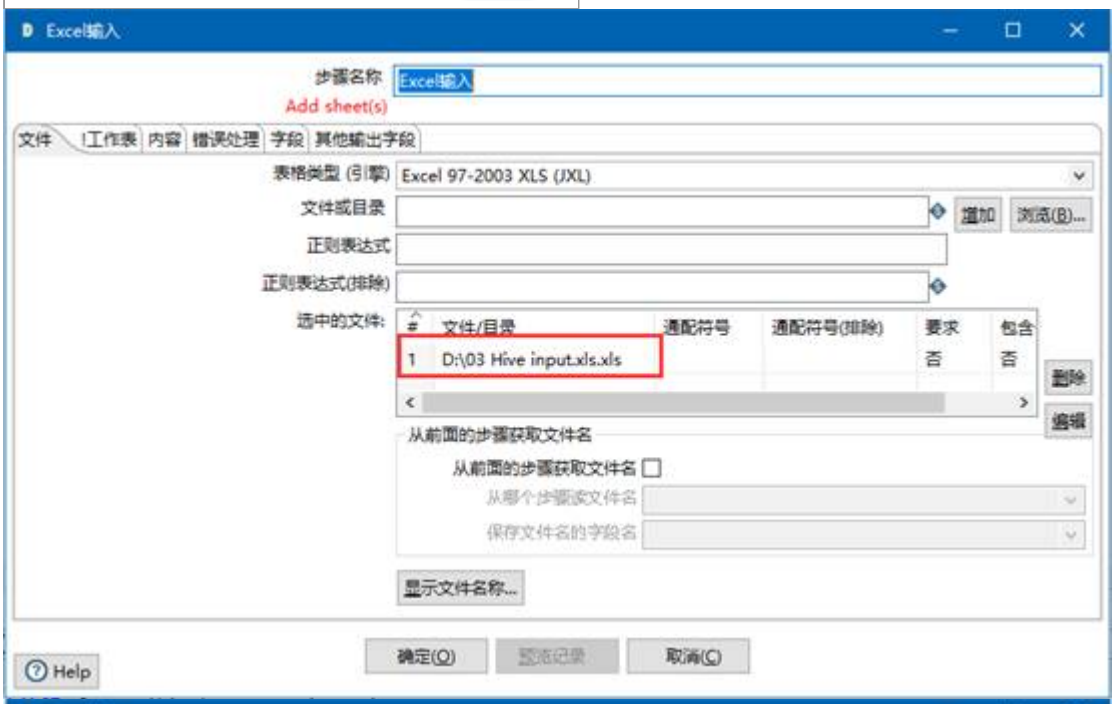
- 聚合查询a表表中a字段大于1的数据，同时建立一个新表new_a保存查询数据。

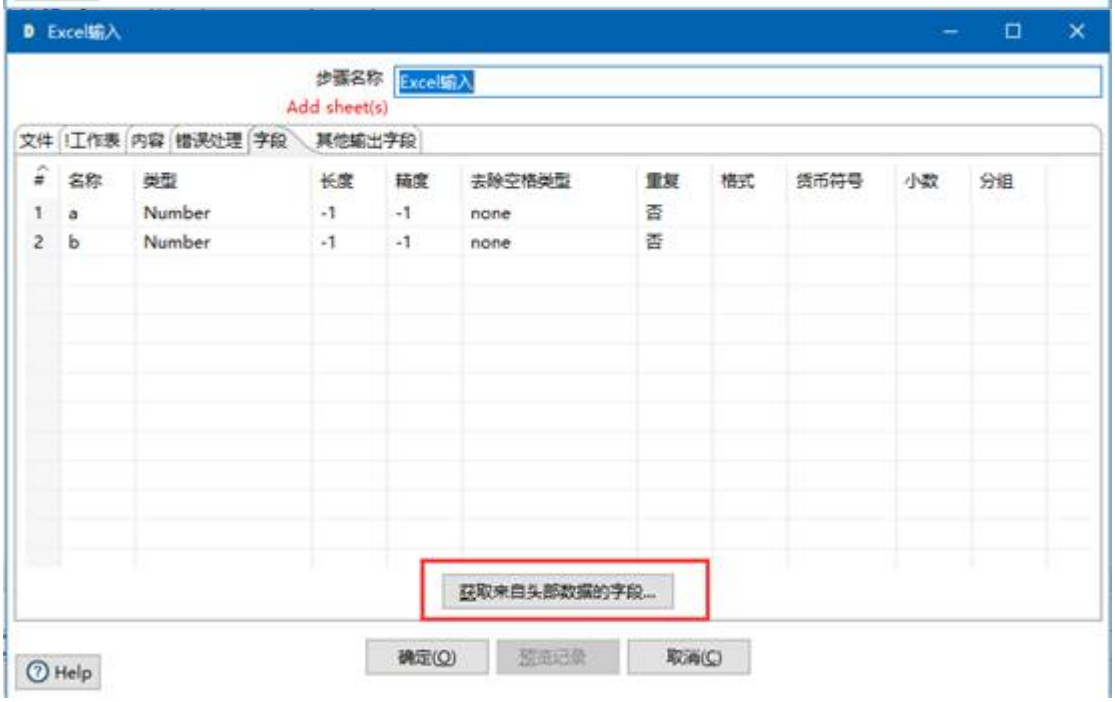
实现步骤：

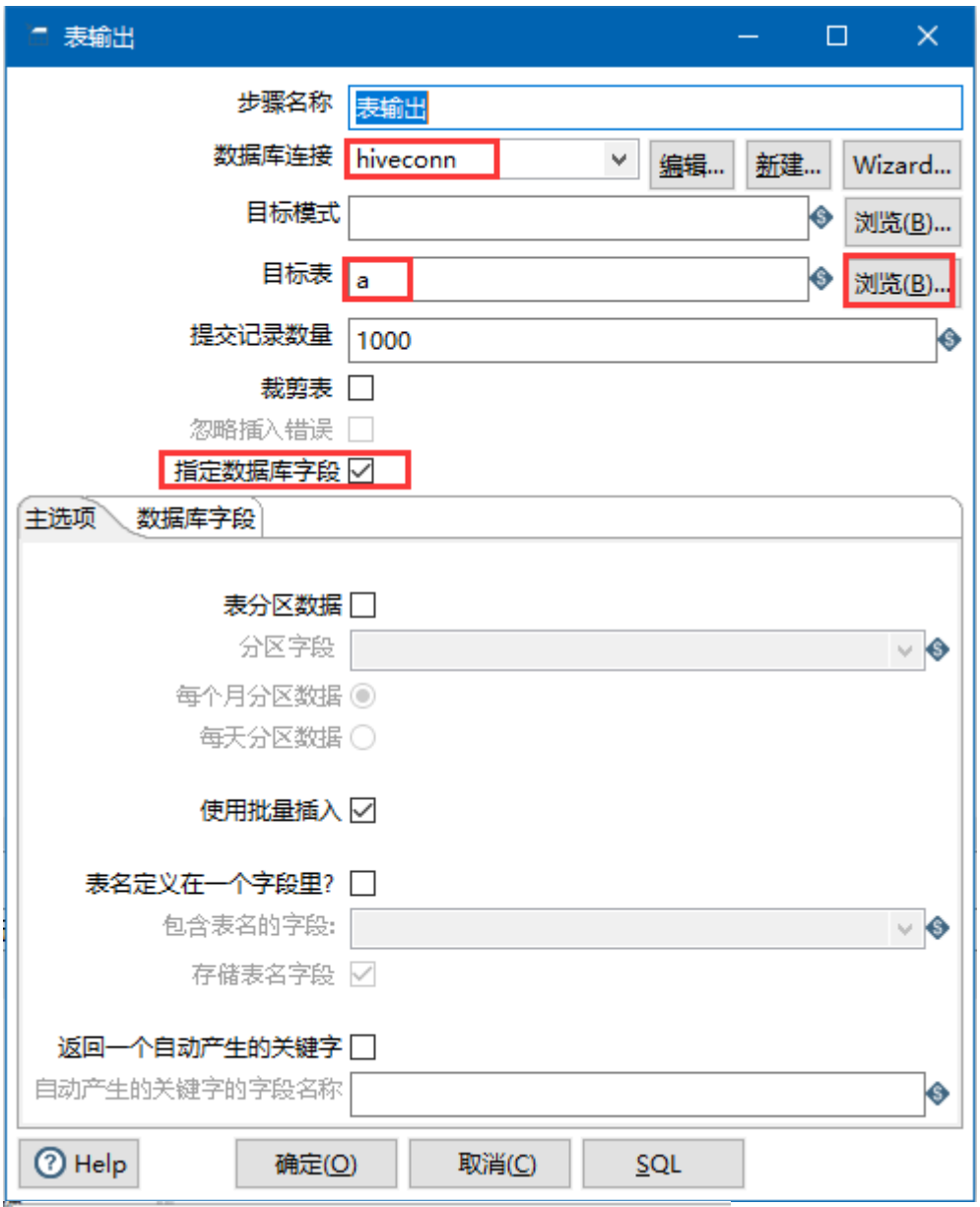
1、设计如下作业组件结构

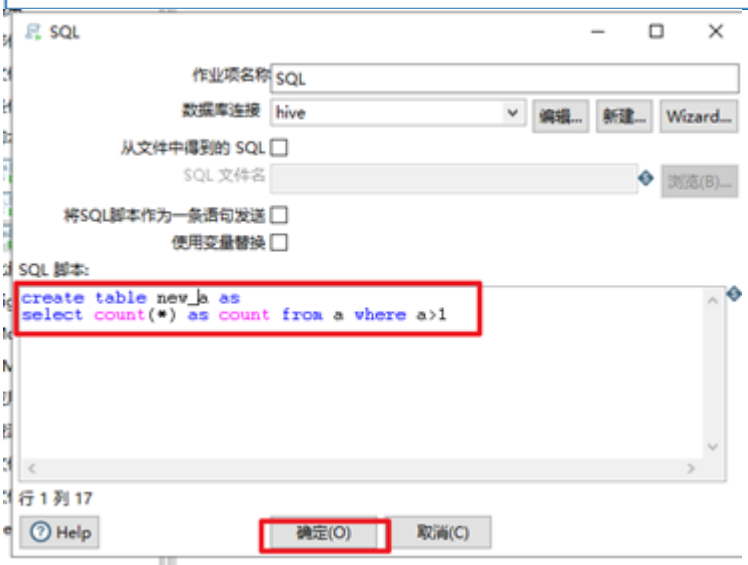

Excel输出
步骤名称
Excel输出
文件
内容
格式
字段
文件名
D:\03 Hive input.xls
浏览(B)...
创建父目录
启动时不创建文件
扩展名
xls
在文件名里包含步骤数?
在文件名里包含日期?
在文件名里包含时间?
指定时间格式
时间格式
显示字段名称...
结果中添加文件名
Help
确定(O)
取消(C)
Excel输出
步骤名称
Excel输出
文件
内容
格式
字段
#
名称
类型
格式
1
a
Integer
2
b
Integer
获取字段
最小宽度
Help
确定(O)
取消(C)


Excel输入
表输出

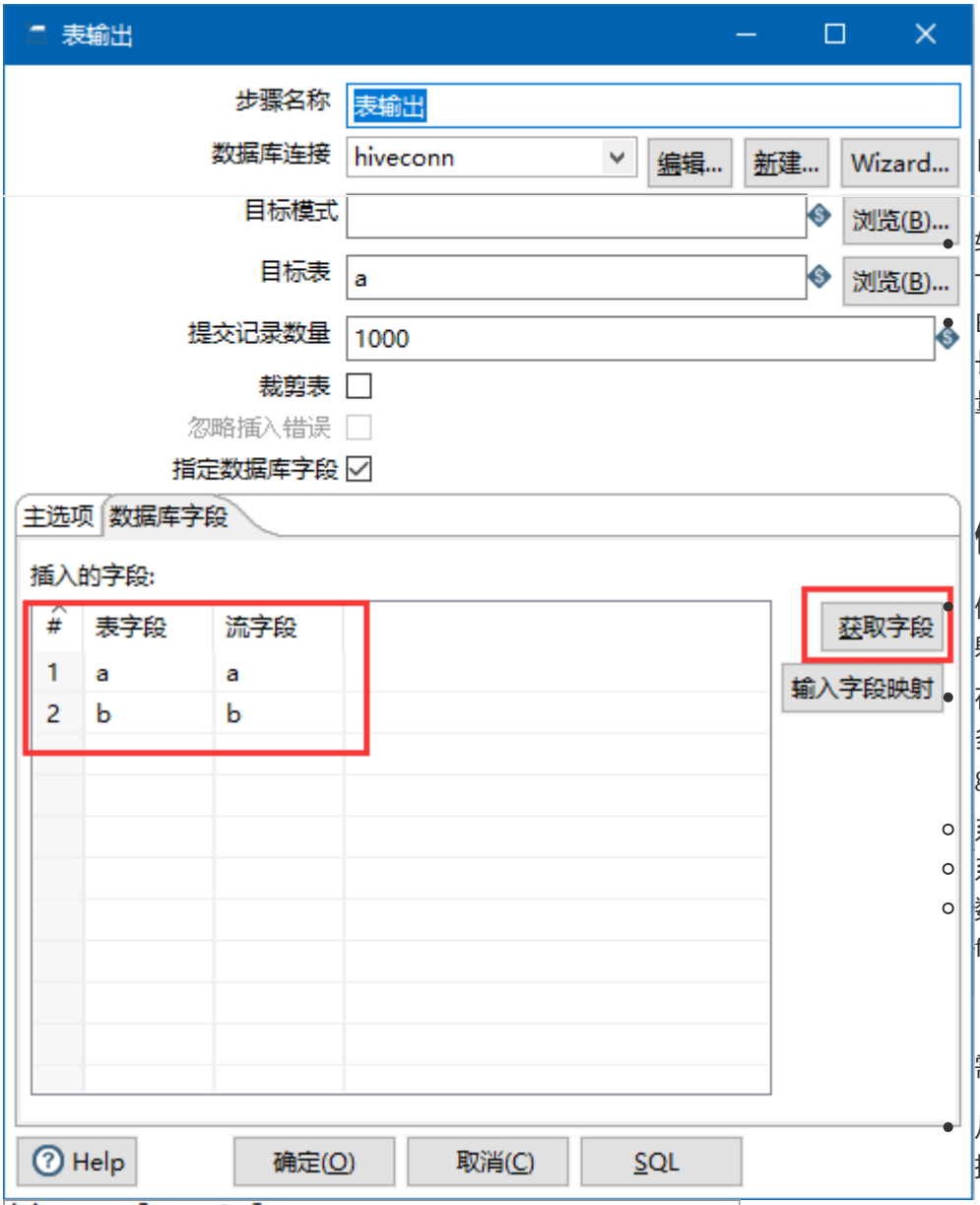

Excel输入
步骤名称
Excel输入
Add sheet(s)
文件
!工作表
内容
错误处理
字段
其他输出字段
表格类型 (引擎)
Excel 97-2003 XLS (JXL)
文件或目录
增加
浏览(B)...
正则表达式
正则表达式(排除)
选中的文件:
文件/目录 通配符号 通配符号(排除) 要求 包含
1 D:\03 Hive input.xls.xls 否 否
删除
编辑
从前面的步骤获取文件名
从前面的步骤获取文件名
从哪个步骤读文件名
保存文件名的字段名
显示文件名称...
Help
确定(O)
预览记录
取消(C)


Excel输入
步骤名称
Excel输入
Add sheet(s)
文件
!工作表
内容
错误处理
字段
其他输出字段
名称 类型 长度 精度 去除空格类型 重复 格式 货币符号 小数 分组
1 a Number -1 -1 none 否
2 b Number -1 -1 none 否
获取来自头部数据的字段...
Help
确定(O)
预览记录
取消(C)


表输出
步骤名称
表输出
数据库连接
hiveconn
编辑...
新建...
Wizard...
目标模式
浏览(B)...
目标表
a
浏览(B)...
提交记录数量
1000
裁剪表
忽略插入错误
指定数据库字段
主选项
数据库字段
表分区数据
分区字段
每个月分区数据
每天分区数据
使用批量插入
表名定义在一个字段里?
包含表名的字段:
存储表名字段
返回一个自动产生的关键字
自动产生的关键字的字段名称
Help
确定(O)
取消(C)
SQL


SQL
作业项名称
SQL
数据库连接
hive
编辑...
新建...
Wizard...
从文件中得到的 SQL
SQL 文件名
浏览(B)...
将SQL脚本作为一条语句发送
使用变量替换
SQL 脚本:
create table new_a as
select count(*) as count from a where a>1
行 1 列 17
Help
确定(O)
取消(C)

3、测试数据是否生成




```
hive> select * from a;
OK
1       11
2       22
3       33
1       11
2       22
3       33
Time taken: 0.454 seconds, Fetched: 6 row(s)
hive>
```

Kettle转换组件

转换是ETL的T，T就是Transform清洗、转换

ETL三个部分中，T花费时间最长,是“一般情况下这部分工作量是整个ETL的2/3

值映射

值映射就是把字段的一个值映射成其他的值

在数据质量规范上使用非常多，比如很多系统对应性别gender字段的定义不同

- 系统1：1 男、2女
- 系统2：f 男、m 女
- 数据仓库统一为：male 男、female女

需求:

从user.json 中读取数据，并把gender列

0 -> 男

1 -> 女

2 -> 保密

写入到Excel文件

实现步骤:

1、拖入一个 JSON输入组件、一个值映射转换组件、一个Excel输出组件，连接三个组件

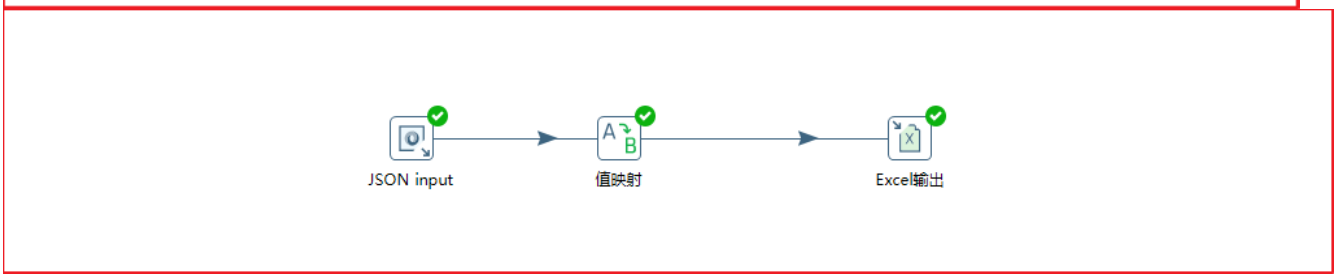
2、配置JSON输入组件

3、配置值映射转换组件

4、配置Excel输出组件

具体实现:

1、拖入一个 JSON输入组件、一个值映射转换组件、一个Excel输出组件，连接三个组件



2、配置JSON输入组件

3、配置值映射转换组件

4、配置Excel输出组件

增加序列

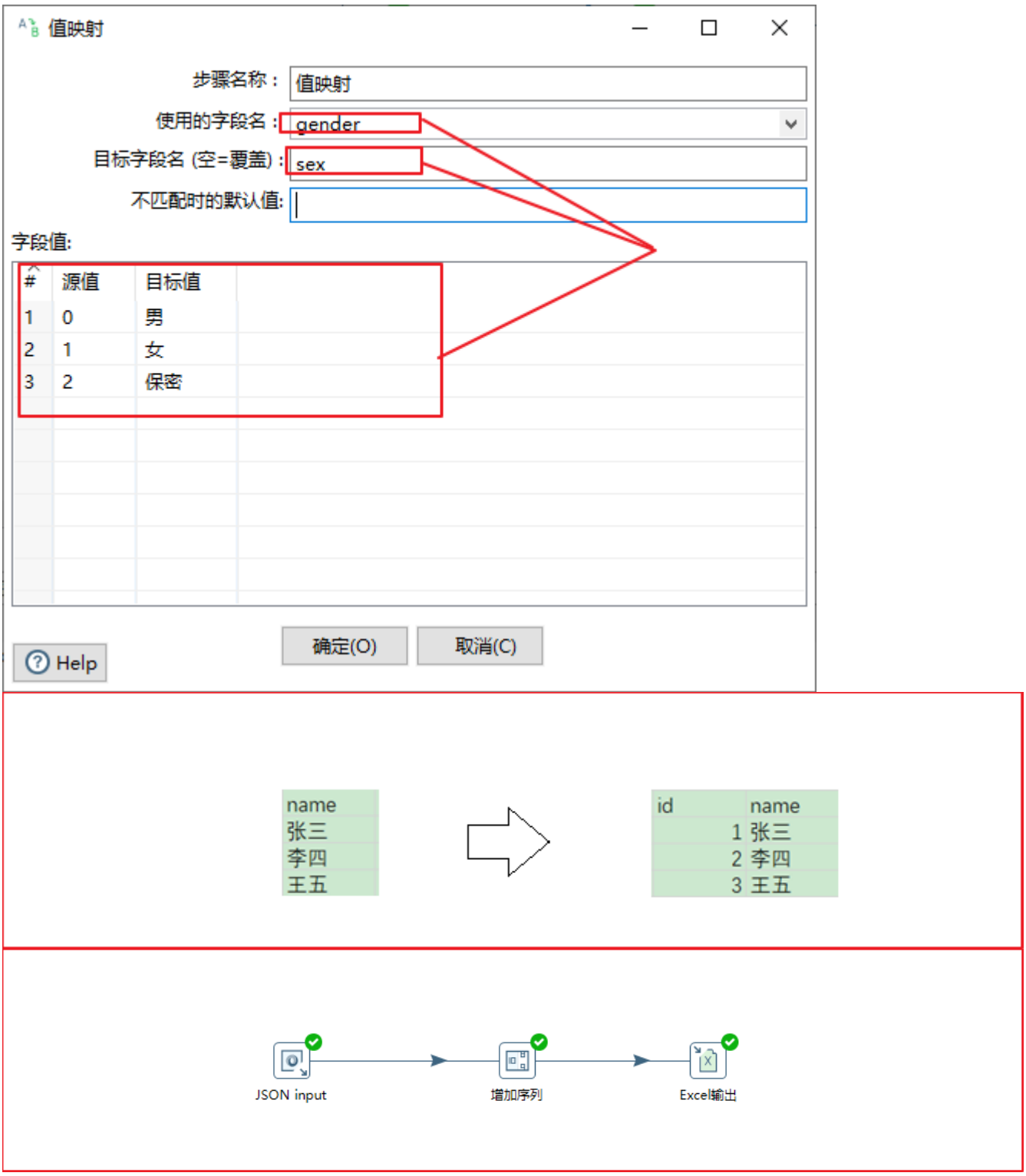
- 增加序列就是给数据流增加一个序列字段

需求：

- 从 user.json 读取数据，并添加序列，把数据保存到Excel

实现步骤：

1、拖入JSON输入组件、增加序列组件、Excel输出组件，并连接三个组件



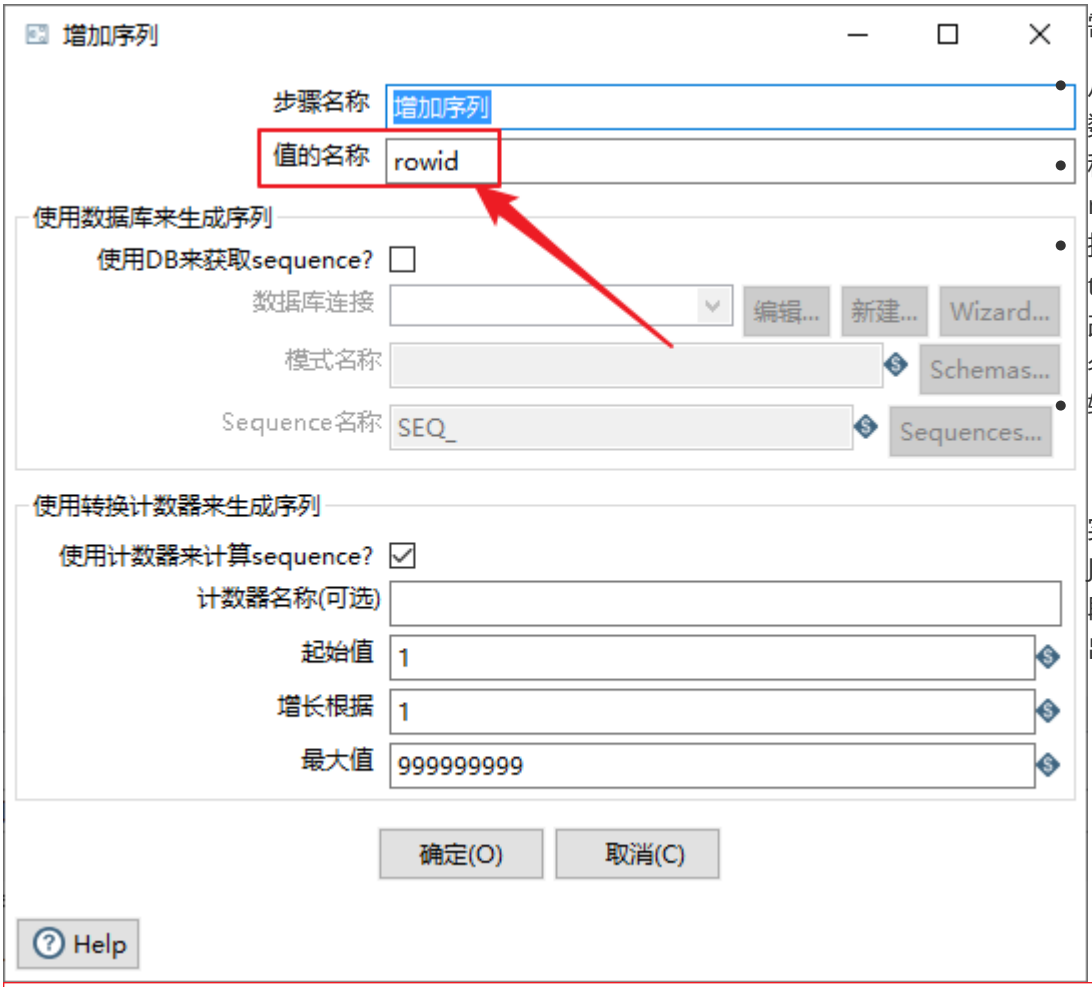
2、配置JSON Input组件

3、配置增加序列组件

4、配置Excel输出组件

字段选择

- 字段选择是从数据流中选择字段、改变名称、修改数据类型



需求：

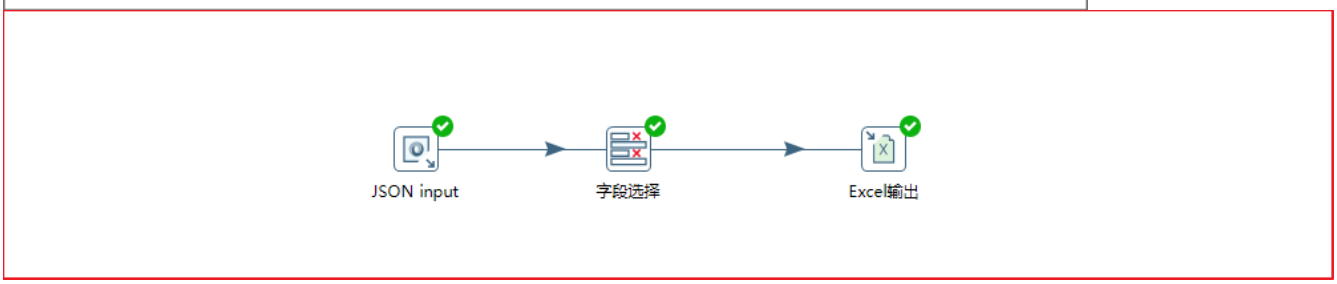
从 user.json 中读取数据

移除birthday和register_date

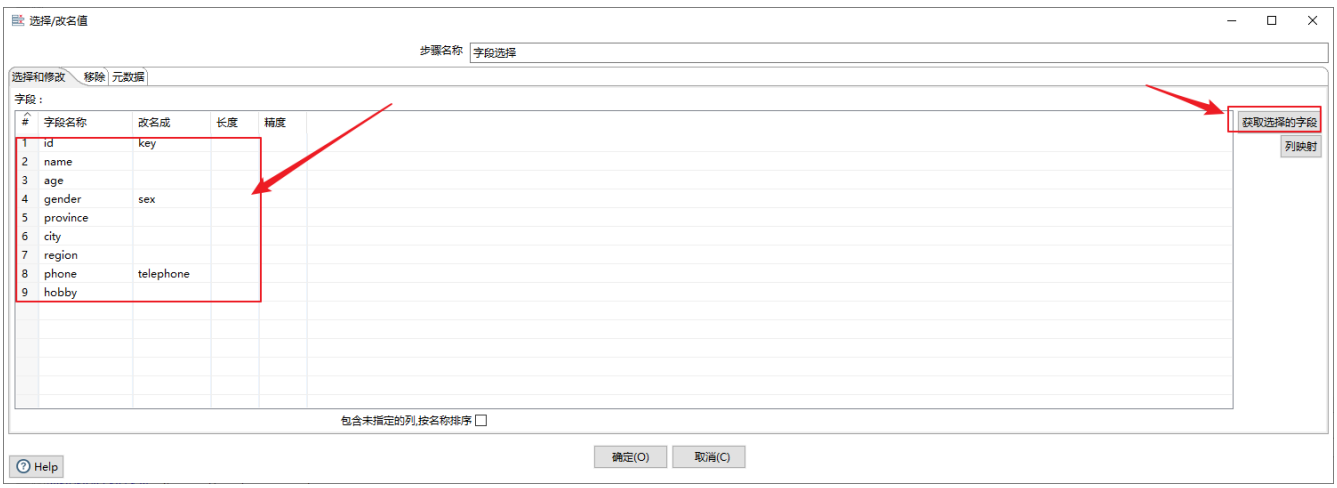
把phone列名改为telephone，id列名改为key，gender列名改为sex

输出到Excel文件中

实现步骤： 1、拖入JSON输入 组件、字段选择组件、Excel输出组件



2、配置输入、字段选择、输出组件

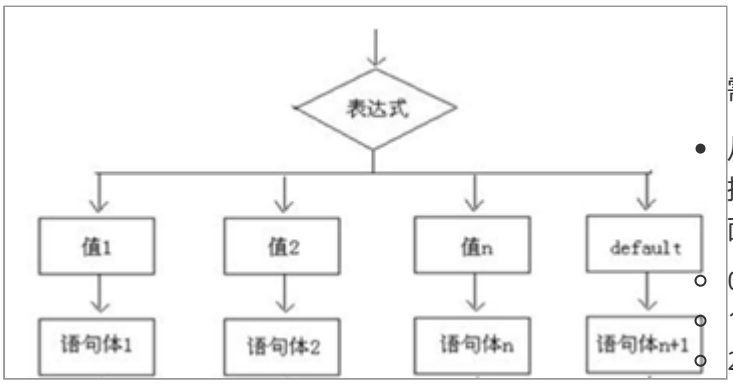


Kettle流程控件

- 流程主要用来控制数据流程和数据流向

switch case

- switch/case组件让数据流从一路到多路。

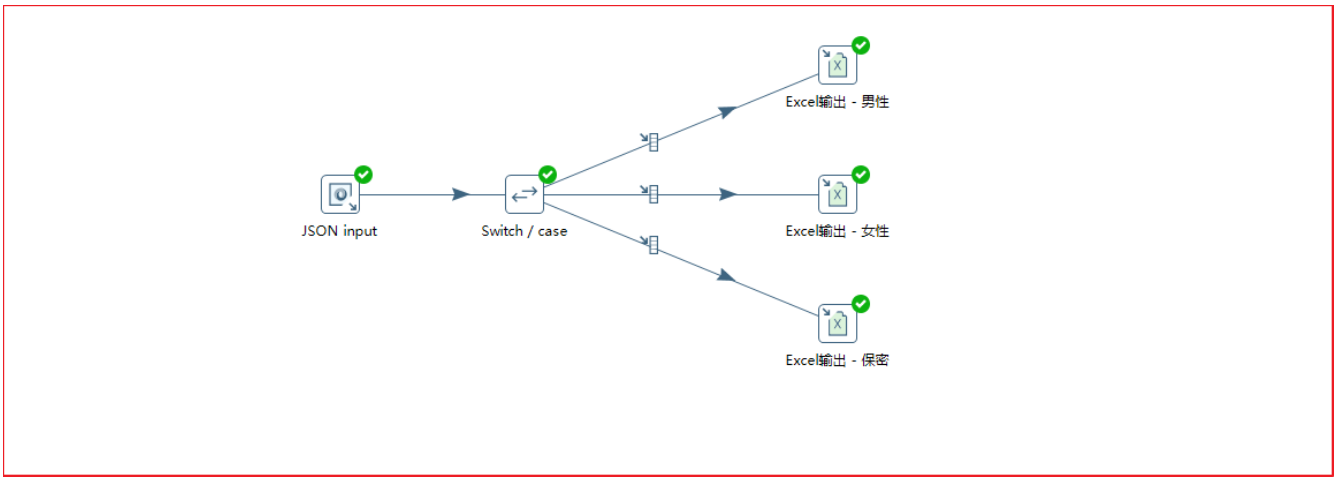


需求：

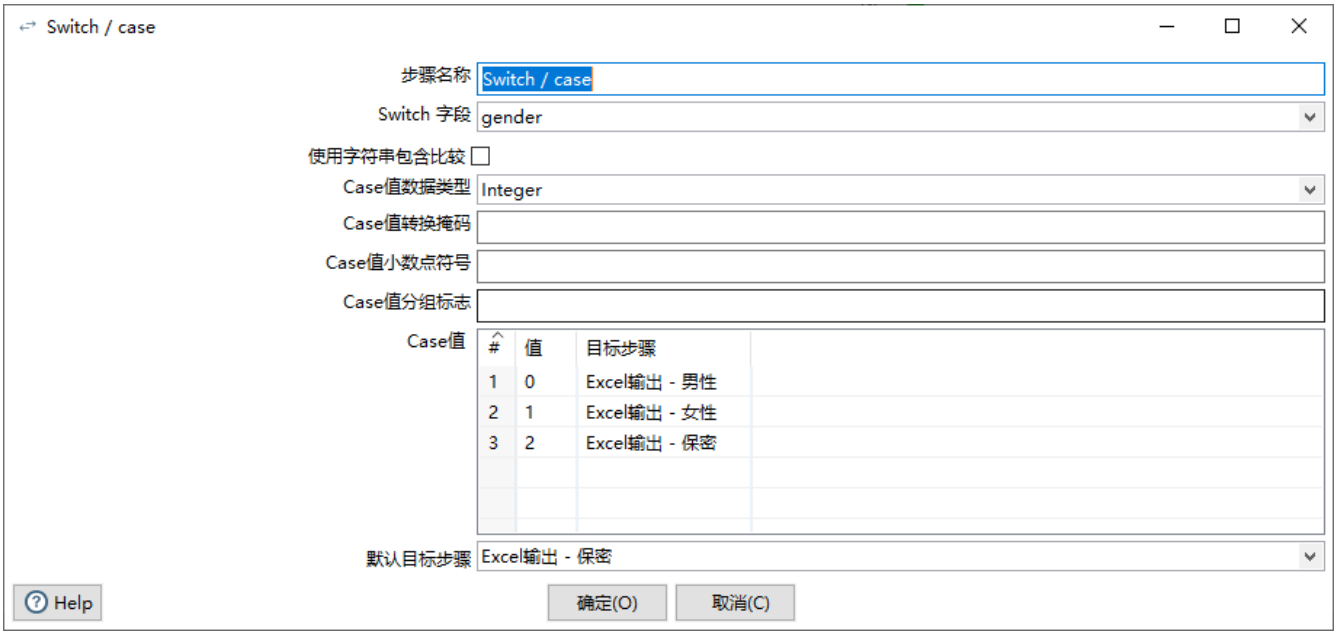
- 从 user.json 输入读取数据，按sex进行数据分类，把女性、男性、保密分别保存不同的Excel文件里面。
  - 0表示男性
  - 1表示女性
  - 2表示保密

实现步骤：

1、拖入 JSON输入组件，switch/case组件，三个Excel输出组件

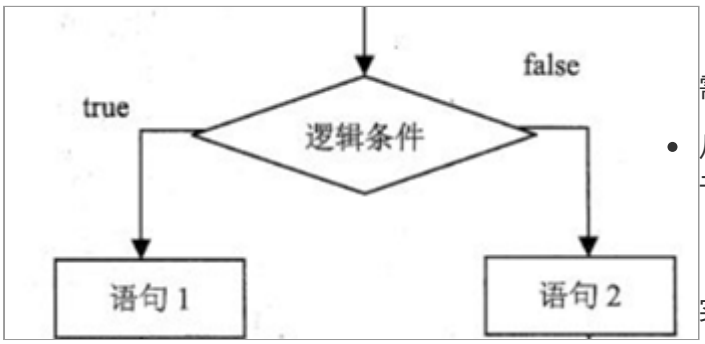


2、配置 switch/case 组件



过滤记录

过滤记录让数据流从一路到两路。

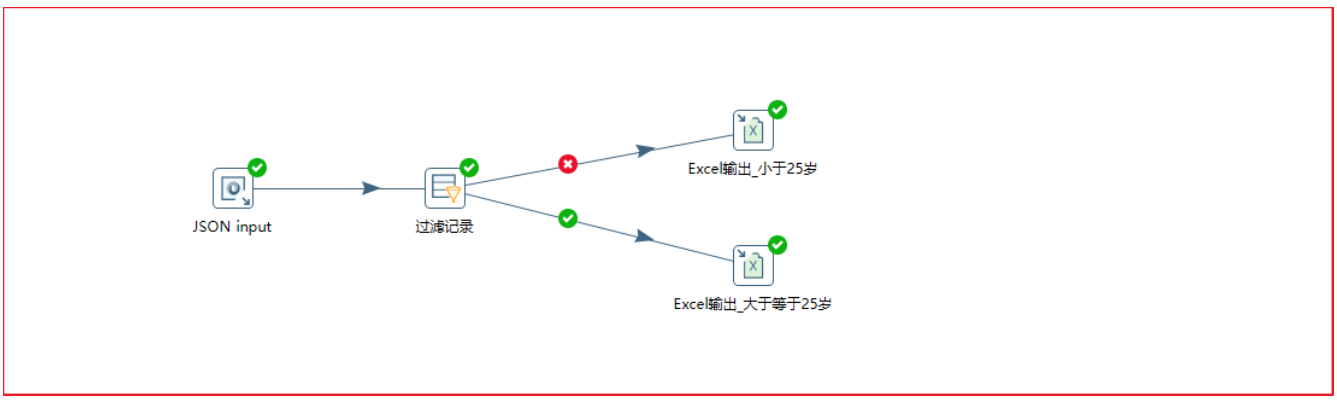


需求：

- 从 user.json 读取数据，分离出 年龄 大于等于25，小于25的数据，分别保存到不同的Excel文件

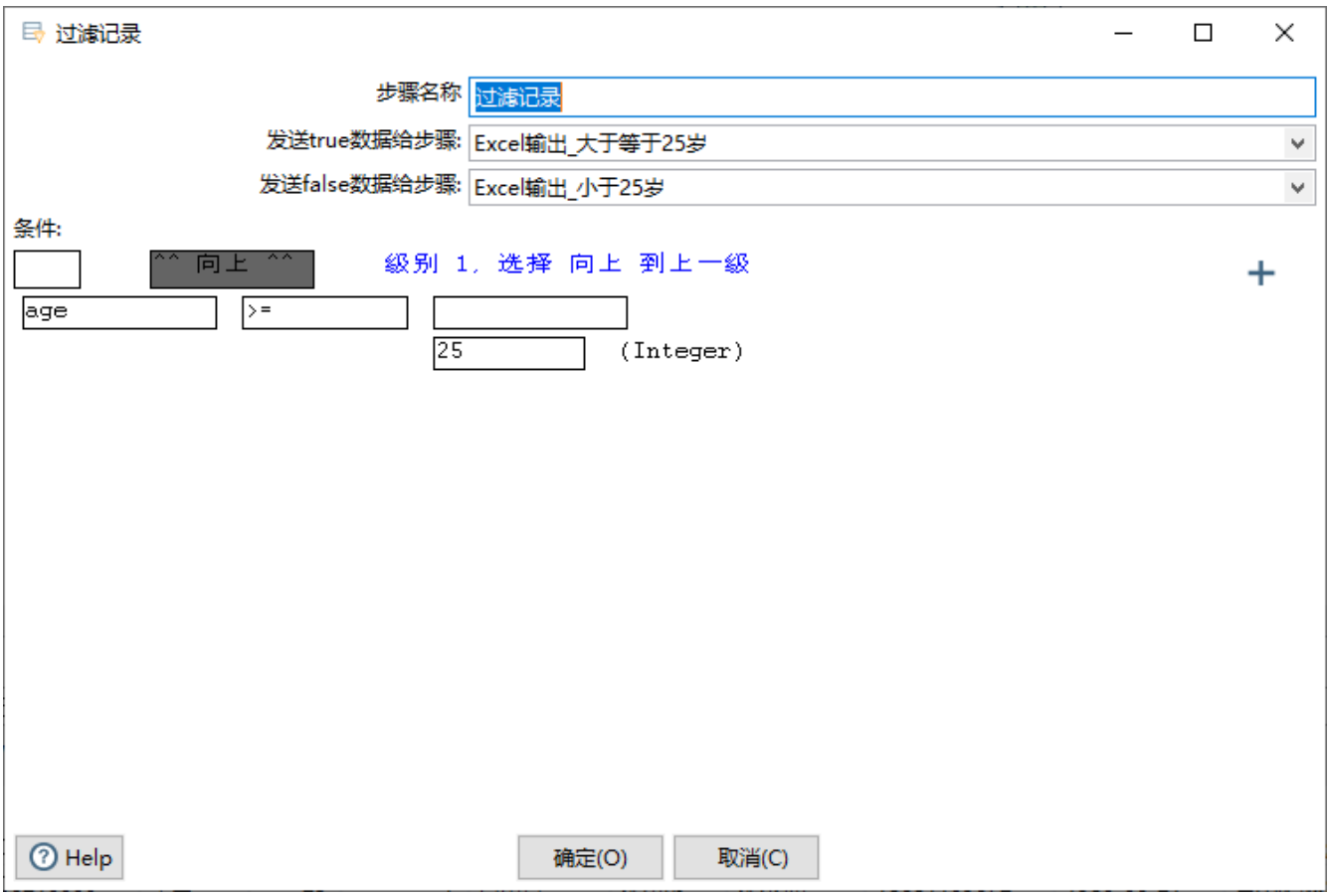
实现步骤：

1、拖入 JSON输入组件、过滤记录组件、两个Excel组件，并连接各个组件



2、配置过滤记录组件

Kettele连接控件



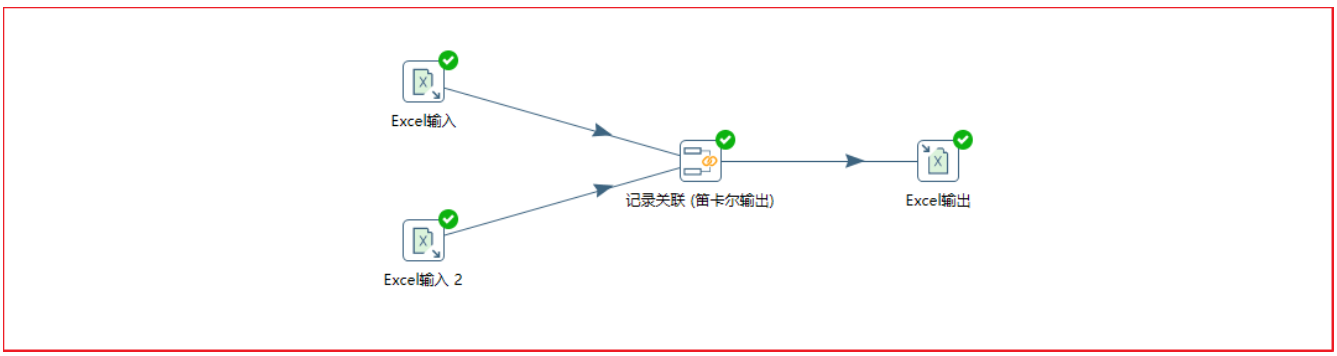
笛卡尔积

需求：

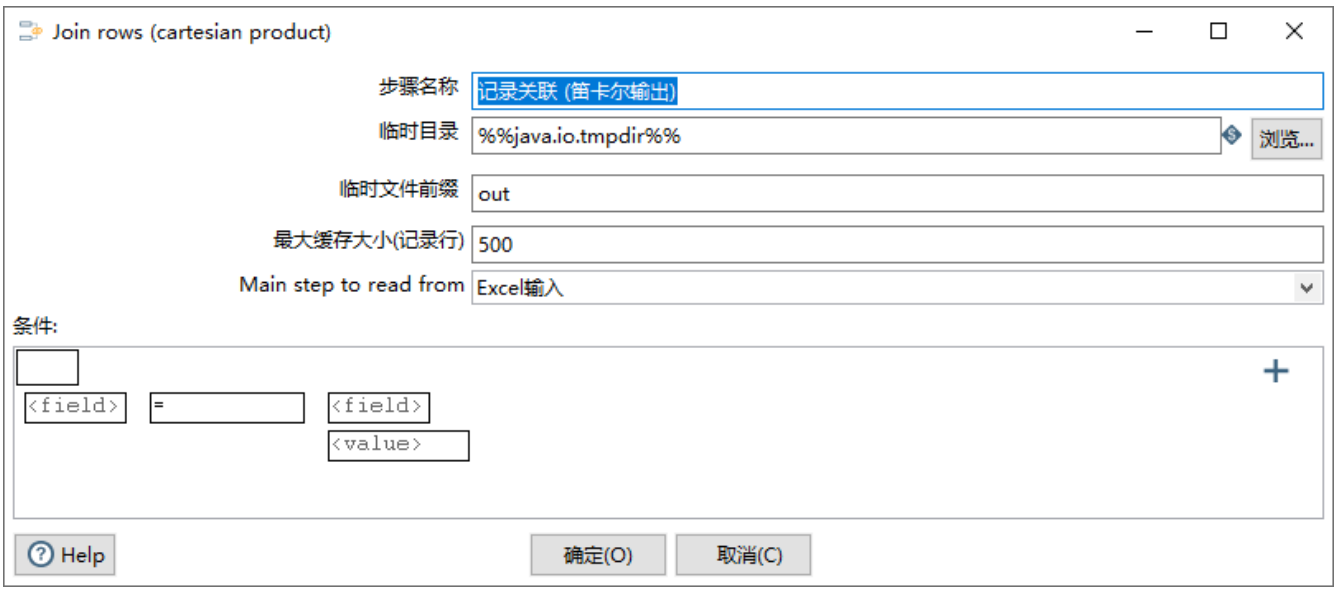
- 从Excel读取两位和三位数，完成两位数和三位数的组合（笛卡尔积）,把结果保存在Excel

实现步骤：

1、设计转换结构



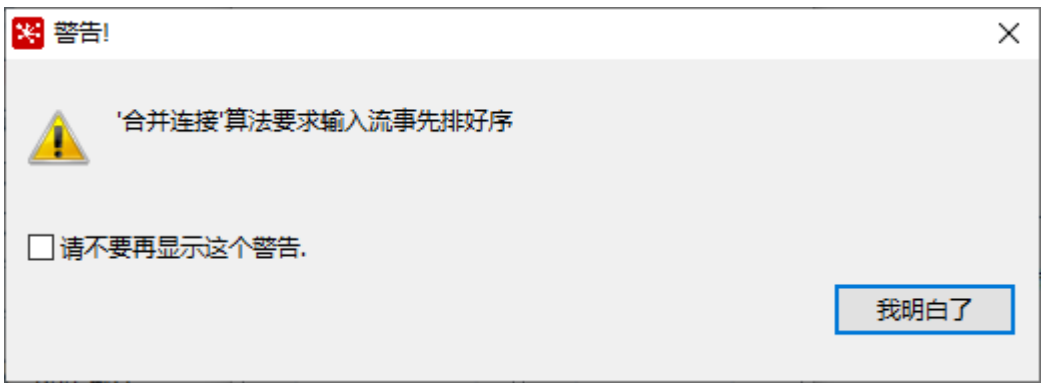
2、配置记录关联（笛卡尔积组件）【不要任何设置】



===

记录集连接

- 记录集连接类似数据库的左连接、右连接、内连接、外连接。
- 在进行记录集连接之前，应该要对记录集进行**排序。**

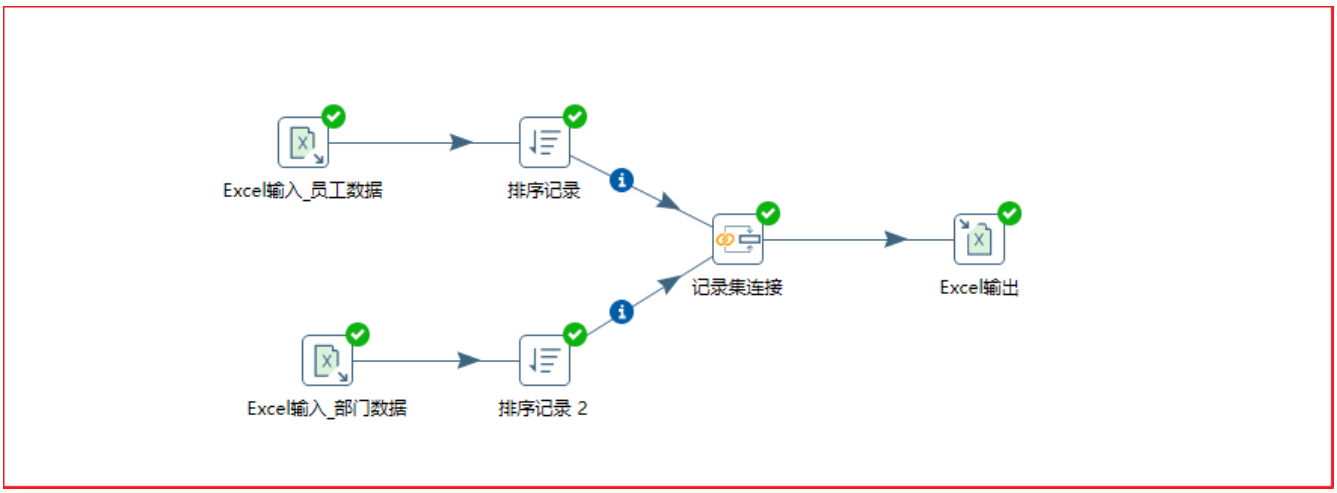


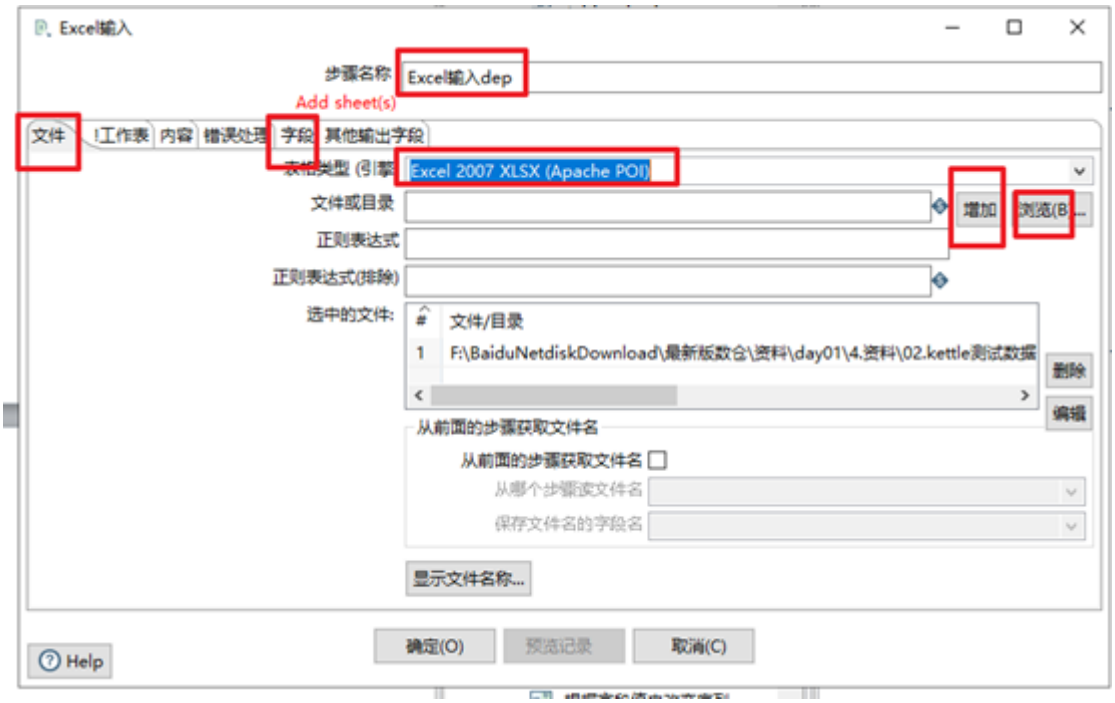
需求：

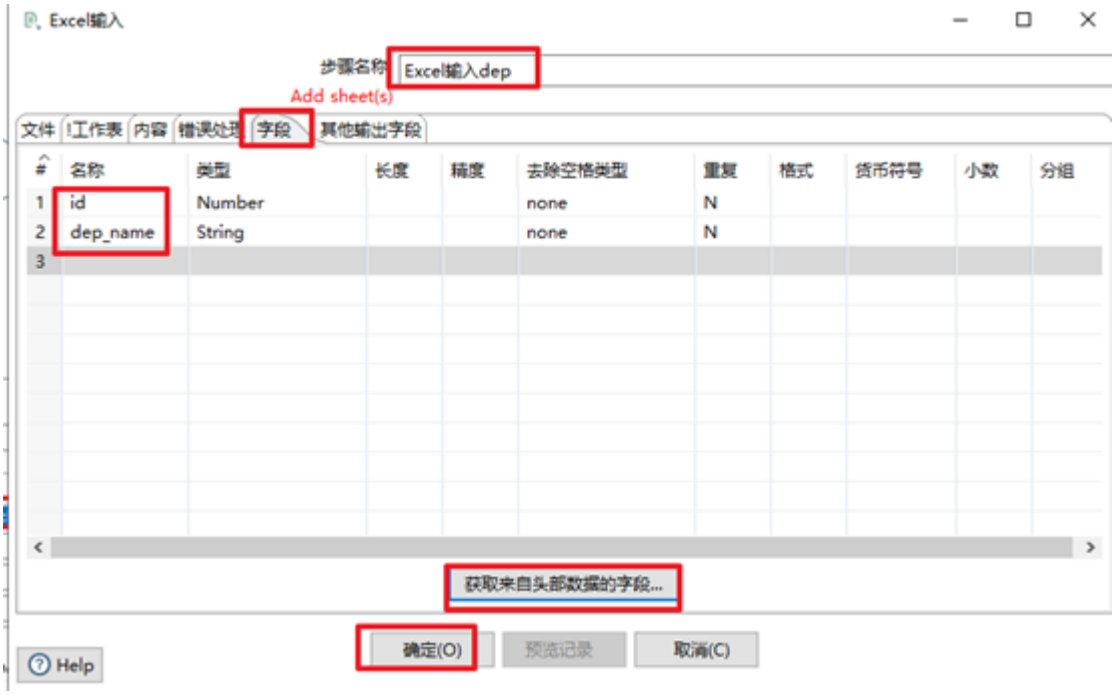
- 从Excel中读取employees和departments数据，进行内关联，左关联，右关联，全关联，把数据保存到Excel

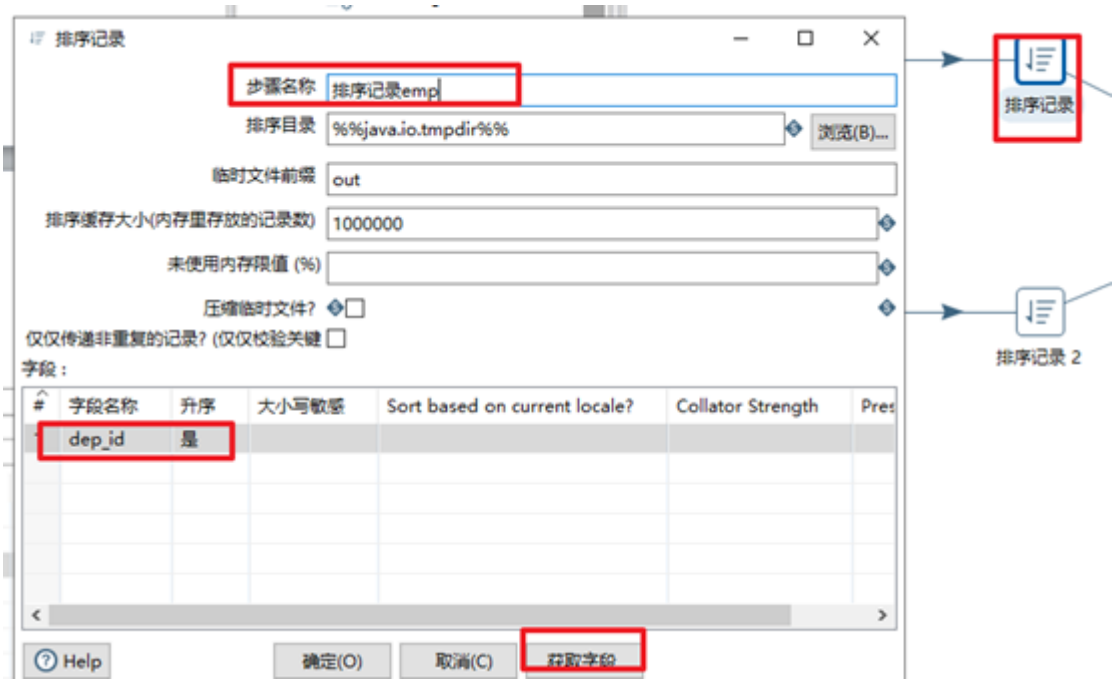
实现步骤：

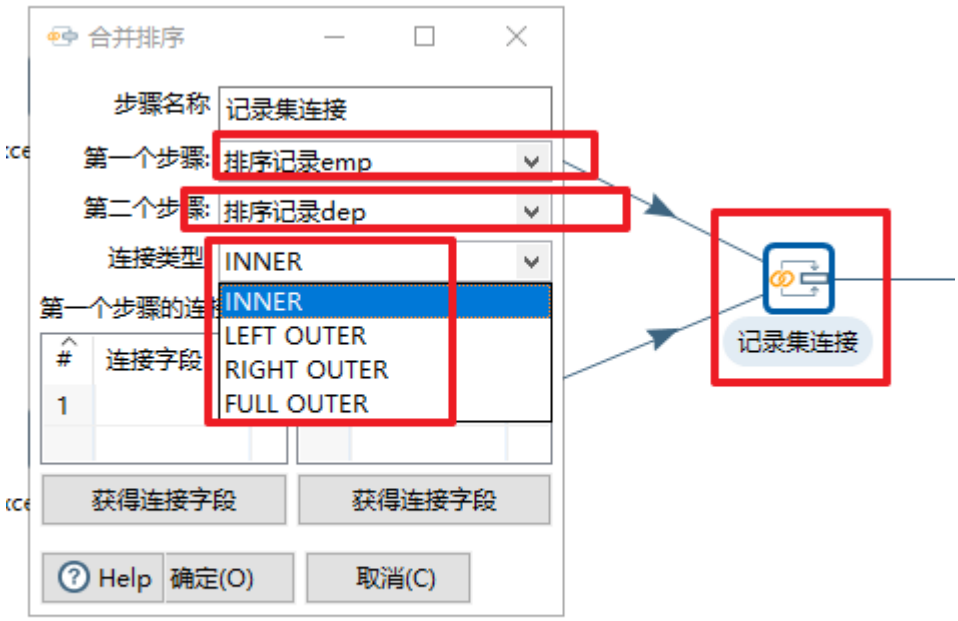
1、设计以下组件图

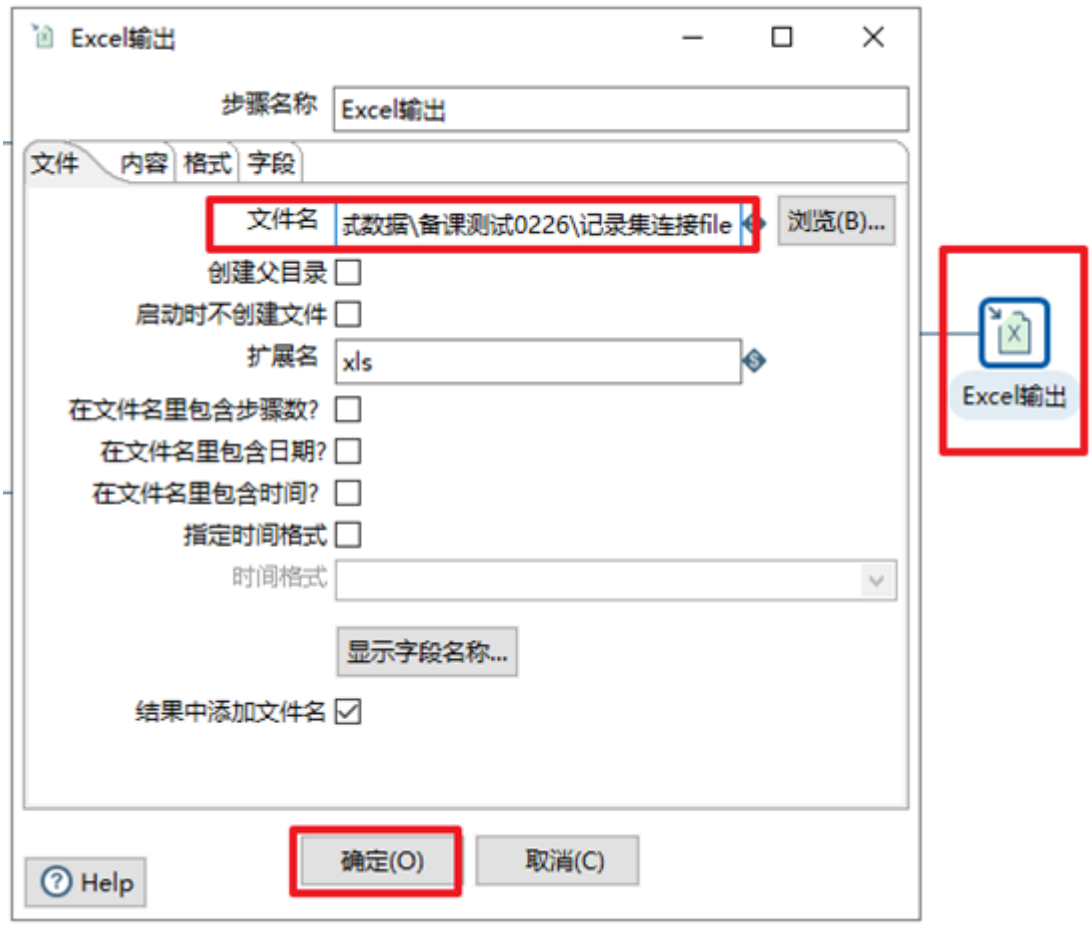

Excel输入_员工数据
排序记录
记录集连接
Excel输出
Excel输入_部门数据
排序记录 2

- 
Excel输入
步骤名称
Excel输入dep
Add sheet(s)
文件
工作表
内容
错误处理
字段
其他输出字段
Excel 2007 XLSX (Apache POI)
文件或目录
增加
正则表达式
正则表达式(排除)
选中的文件:
#
文件/目录
1
F:\BaiduNetdiskDownload\最新版数仓\资料\day01\4.资料\02.kettle测试数据
删除
编辑
从前面的步骤获取文件名
从前面的步骤获取文件名
从哪个步骤读文件名
保存文件名的字段名
显示文件名称...
Help
确定(O)
预览记录
取消(C)


Excel输入
步骤名称
Excel输入dep
Add sheet(s)
文件
工作表
内容
错误处理
字段
其他输出字段
#
名称
类型
长度
精度
去除空格类型
重复
格式
货币符号
小数
分组
1
id
Number
none
N
2
dep_name
String
none
N
3
获取来自头部数据的字段...
确定(O)
预览记录
取消(C)
Help

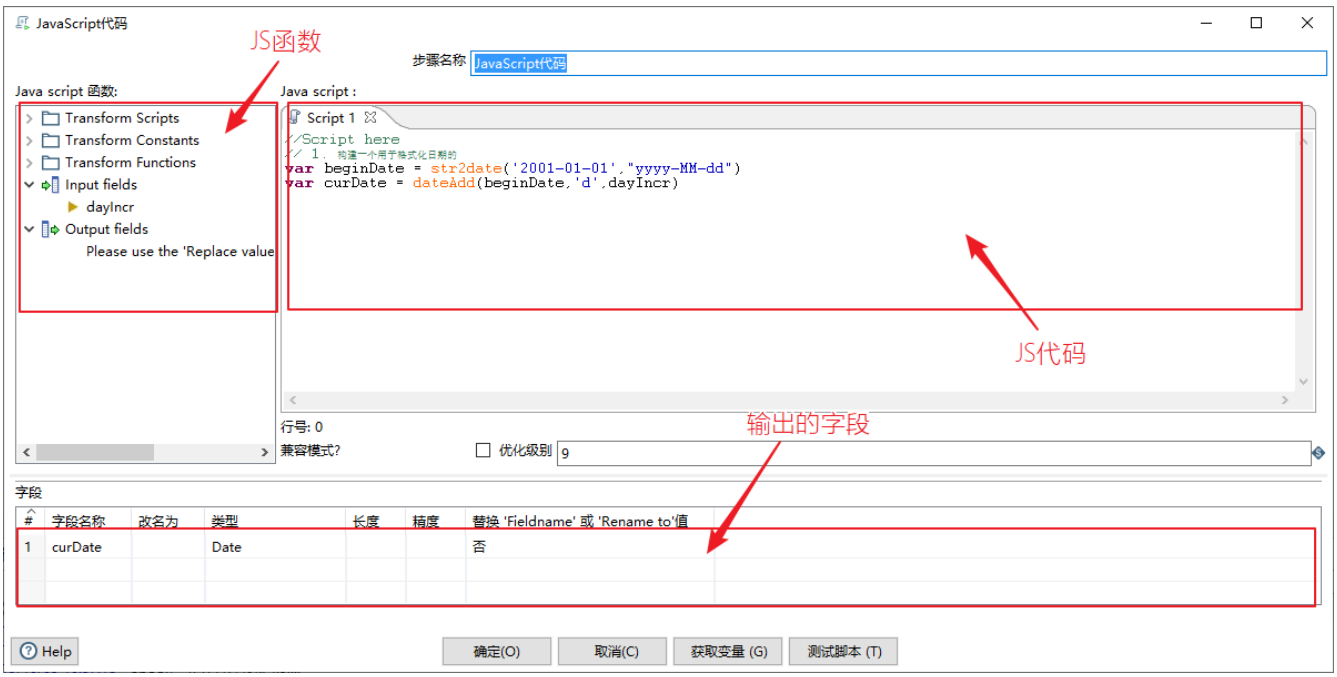

排序记录
步骤名称
排序记录emp
排序目录
%%java.io.tmpdir%%
浏览(B)...
临时文件前缀
out
排序缓存大小(内存里存放的记录数)
1000000
未使用内存限值 (%)
压缩临时文件?
仅仅传递非重复的记录? (仅仅校验关键
字段：
#
字段名称
升序
大小写敏感
Sort based on current locale?
Collator Strength
Pres
dep_id
是
Help
确定(O)
取消(C)
获取字段
排序记录
排序记录 2


合并排序
步骤名称
记录集连接
第一个步骤:
排序记录emp
第二个步骤:
排序记录dep
连接类型
INNER
INNER
LEFT OUTER
RIGHT OUTER
FULL OUTER
#
连接字段
1
获得连接字段
获得连接字段
Help
确定(O)
取消(C)
记录集连接


Excel输出
步骤名称
Excel输出
文件
内容
格式
字段
文件名
式数据\备课测试0226\记录集连接file
浏览(B)...
创建父目录
启动时不创建文件
扩展名
xls
在文件名里包含步骤数?
在文件名里包含日期?
在文件名里包含时间?
指定时间格式
时间格式
显示字段名称...
结果中添加文件名
Help
确定(O)
取消(C)
Excel输出

Kettle Javascript脚本组件

- Kettle中可以通过脚本完成一些复杂的操作
- javascript脚本就是使用javascript语言通过代码编程来完成对数据流的操作
- JS中有很多内置函数，可以在编写JS代码时查看
- 存在两种不同的模式不兼容模式和兼容模式
    - 不兼容模式：是默认的，也是推荐的
    - 兼容模式：兼容老版本的kettle



对比不兼容模式与兼容模式的差别：

操作	不兼容模式	兼容模式
获取字段	myVar = fieldName	myVar = 字段名称.getString() myVar = 字段名称.getNumber()
给字段赋值	字段名 = myVar	字段名.setValue(myVar)
在脚本中使用java类	var myVar = new java.lang.String("pentahochina.com")	var myVar = new Packages.java.lang.String("pentahochina.com")

需求：

- 生成日期维度数据日期，年，月，日，从2000年01月01日开始有1000条记录，保存到Excel

实现步骤：

1、设计转换结构图

2、编写Java代码，生成日期

- 输入变量
- 处理逻辑

- 打印输出变量

3、处理Java代码，使用完全限定类名

```
public class Class_1生成日期 {
    public static void main(String[] args) throws java.text.ParseException {
        // 1. 初始日期 2000-01-01，设置日期
        String initDate = "2000-01-01";
        // 累加序列
        Integer day = 2;
        // 字符串转换为日期
        java.text.SimpleDateFormat simpleDateFormat = new
java.text.SimpleDateFormat("yyyy-MM-dd");
        java.util.Date date = simpleDateFormat.parse(initDate);

        // 2. 生成新的日期
        java.util.Calendar calendar = java.util.Calendar.getInstance();
        calendar.setTime(date);
        calendar.add(java.util.Calendar.DAY_OF_MONTH, day);

        String newDate = simpleDateFormat.format(calendar.getTime());
        System.out.println(newDate);
    }
}
```

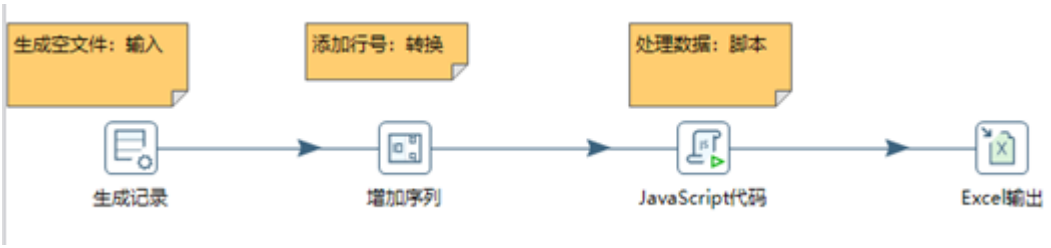
4、将Java代码转换为Kettle中能执行的JS代码

- 变量声明使用 var 代替

```
// 1. 初始日期 2000-01-01，设置日期
var initDate = "2000-01-01";
// 累加序列
 var day =1;

// 字符串转换为日期
var simpleDateFormat = new java.text.SimpleDateFormat("yyyy-MM-dd");
var date = simpleDateFormat.parse(initDate);

// 2. 生成新的日期
var calendar = java.util.Calendar.getInstance();
calendar.setTime(date);
calendar.add(java.util.Calendar.DAY_OF_MONTH, day);

var newDate = simpleDateFormat.format(calendar.getTime());
```

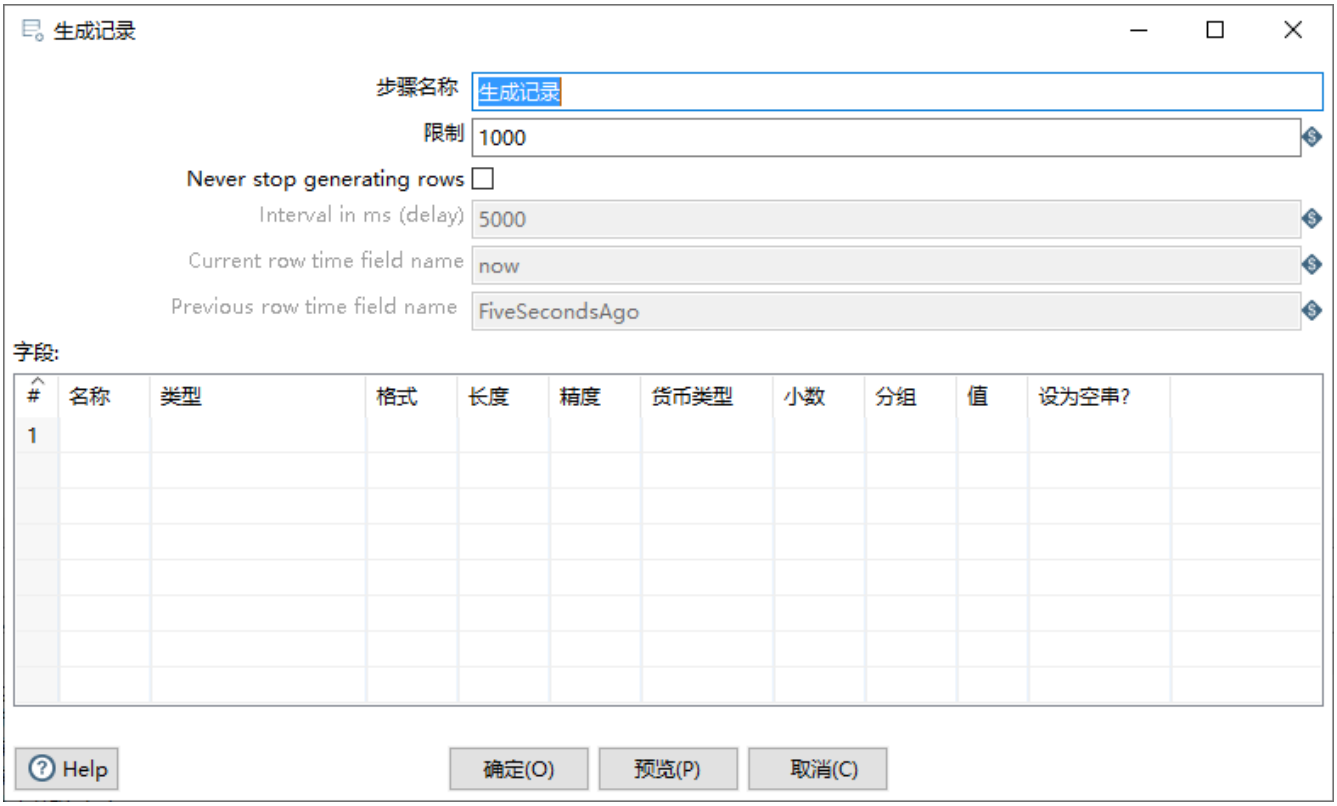
具体实现：

1、准备如下组件

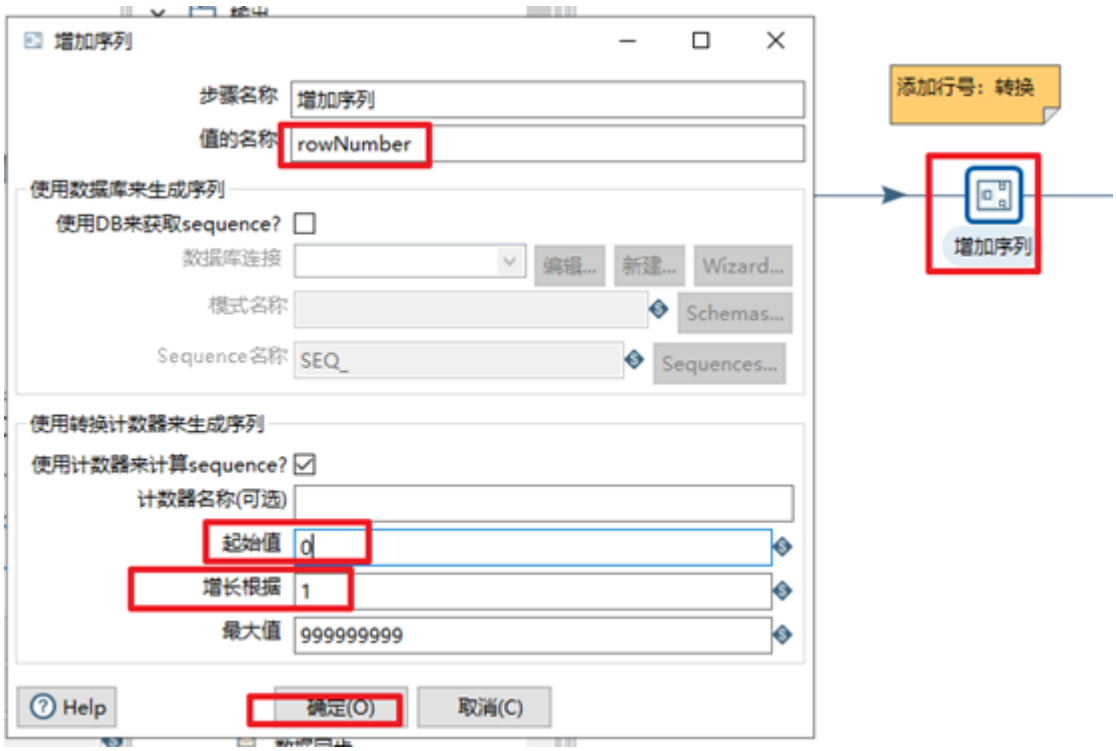


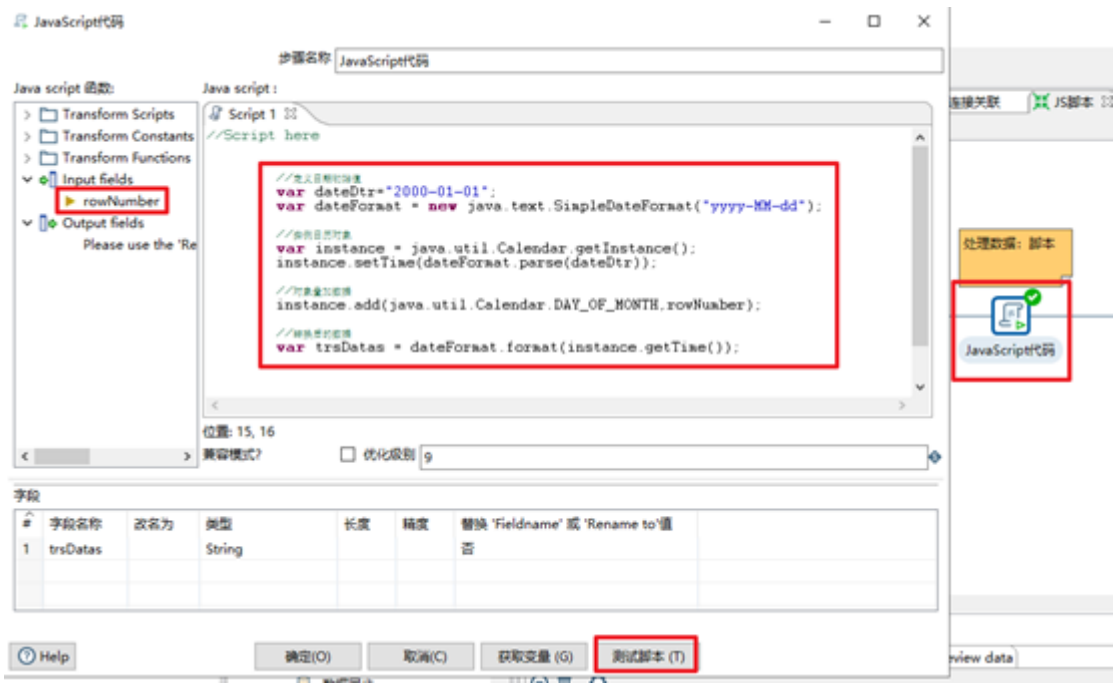
2、配置生成记录组件

- 定义生成多少条记录



3、配置增加序列组件





编写代码过程

第一步：编写java代码

```
public class test {
    public static void main(String[] args) throws java.text.ParseException {
        //定义一个尾部数据
        int newBumber=100;
```

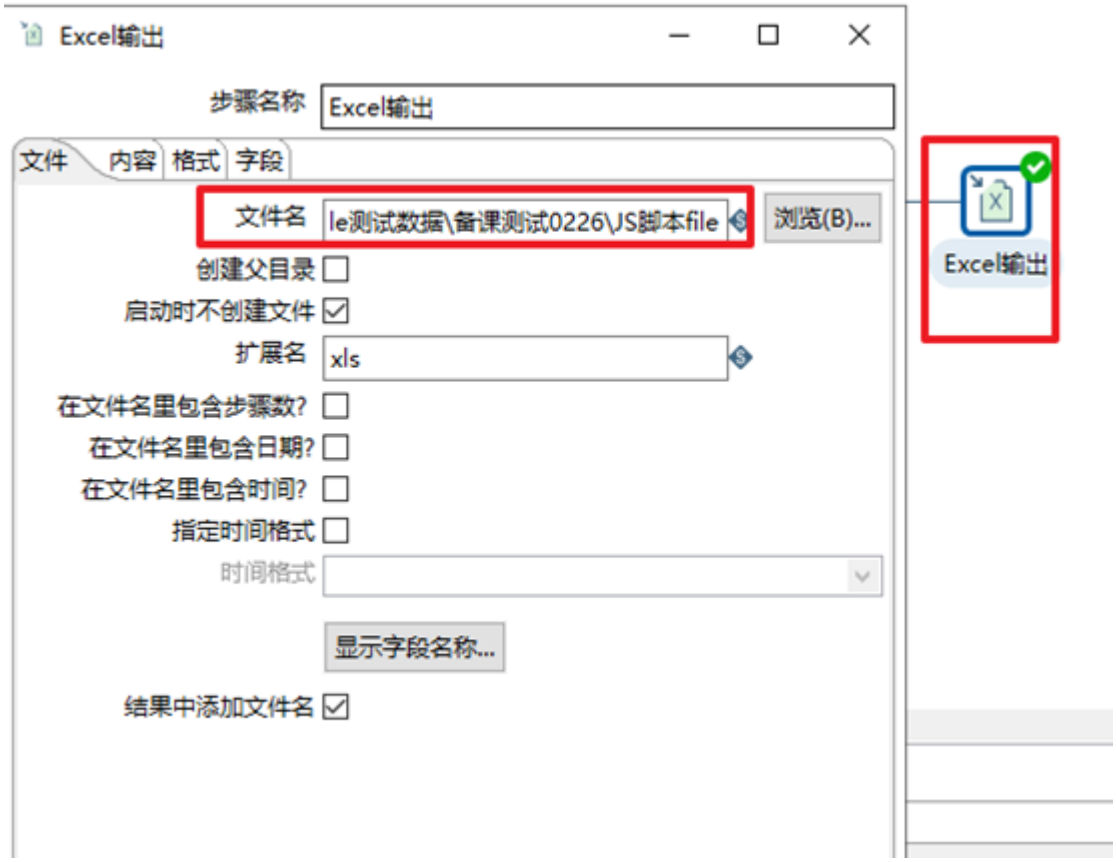
```
        //定义日期初始值
        String dateDtr="2000-01-01";
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        //实例日历对象
        Calendar instance = Calendar.getInstance();
        instance.setTime(dateFormat.parse(dateDtr));
        //对象叠加数据
        instance.add(Calendar.DAY_OF_MONTH,newBumber);
        //转换后的数据
        String trsDatas = dateFormat.format(instance.getTime());


        //输出
        System.out.println(dateFormat.format(instance.getTime()));

    }

}
```

public class test { **public static void** main(String[] args) **throws** java.text.ParseException { *//**定义一个尾部数据* **int** newBumber=100; *//**定义日期初始值* String dateDtr=**"2000-01-01"**; SimpleDateFormat dateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**); *//**实例日历对象* Calendar instance = Calendar.*getInstance*(); instance.setTime(dateFormat.parse(dateDtr)); *//**对象叠加数据* instance.add(Calendar.**DAY_OF_MONTH**,newBumber); *//**转换后的数据* String trsDatas = dateFormat.format(instance.getTime()); *//**输出* System.**out**.println(dateFormat.format(instance.getTime())); } }

第二步：将类名替换为使用完全限定类名，将变量使用“var” 声明

```
        //定义日期初始值
        var dateDtr="2000-01-01";
        var dateFormat = new java.text.SimpleDateFormat("yyyy-MM-dd");

        //实例日历对象
        var instance = java.util.Calendar.getInstance();
        instance.setTime(dateFormat.parse(dateDtr));

        //对象叠加数据
        instance.add(java.util.Calendar.DAY_OF_MONTH,rowNumber);

        //转换后的数据
        var trsDatas = dateFormat.format(instance.getTime());
```

//定义日期初始值 **var** dateDtr="2000-01-01"; **var** dateFormat = **new** java.text.SimpleDateFormat("yyyy-MM-dd"); //实例日历对象 **var** instance = java.util.Calendar.getInstance(); instance.setTime(dateFormat.parse(dateDtr)); //对象叠加数据 instance.add(java.util.Calendar.DAY_OF_MONTH,rowNumber); //转换后的数据 **var** trsDatas = dateFormat.format(instance.getTime());



#	rowNumber	trsDatas
1	0	2000-01-01
2	1	2000-01-02
3	2	2000-01-03
4	3	2000-01-04
5	4	2000-01-05
6	5	2000-01-06
7	6	2000-01-07
8	7	2000-01-08

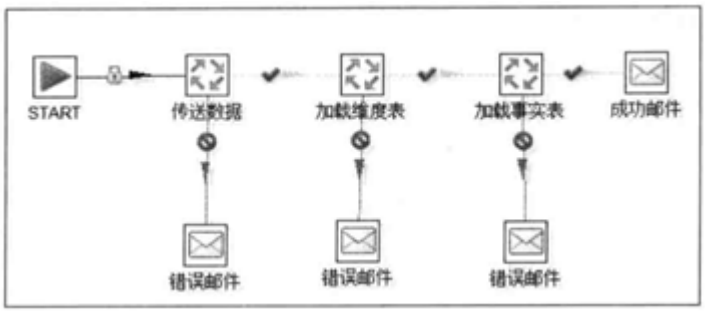
Kettle作业和参数

Job（作业）

大多数ETL项目都需要完成各种各样的操作，例如：

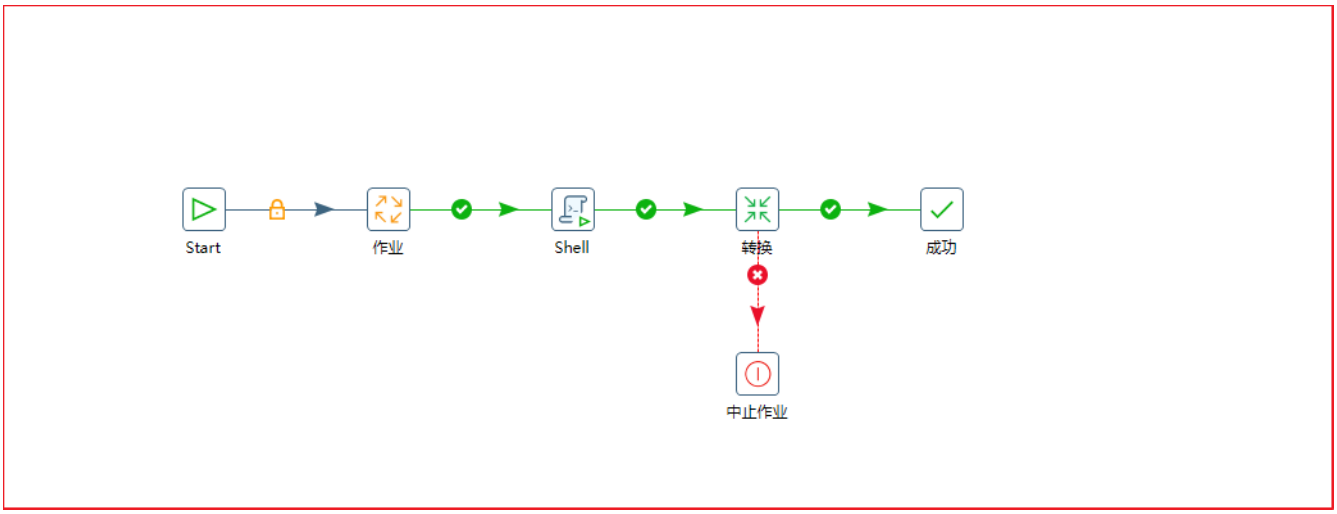
- 如何传送文件
- 验证数据库表是否存在，等等

而这些操作都是按照一定顺序完成，Kettle中的作业可以串行执行转换来处理这些操作。



Job Item（作业项）

作业项是作业的基本构成部分。如同转换的组件，作业项也可以用**图标的方式**展示。



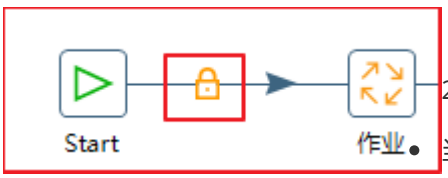
- 作业顺序执行作业项，必须定义一个**起点**
- 有一个「start」的作业项专门用来定义起点
- 一个作业只能定一个开始作业项

Job Hop（作业跳）

Job Hop是作业项之间的连接线，定义了作业的执行路径，作业里每个作业项的不同运行结果决定了作业的不同执行路径。以下为 Job Hop的几种执行方式：

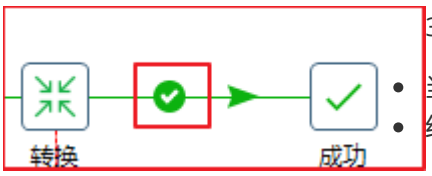
1、无条件执行

- 不论上一个作业项执行成功还是失败，下一个作业项都会执行
- 蓝色的连接线，上面有一个锁的图标



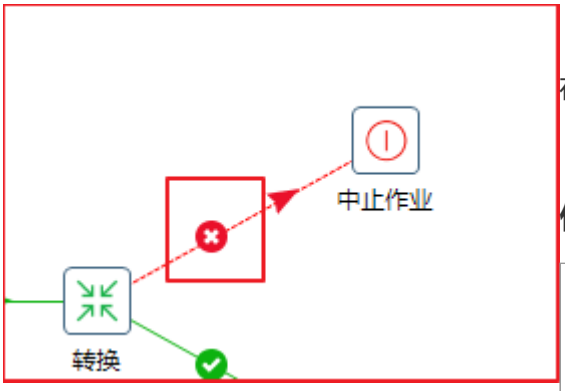
2、当运行结果为真时执行

- 当上一个作业项的执行结果为真时，执行下一个作业项
- 通常在需要无错误执行的情况下使用
- 绿色的连接线，上面有一个对钩号的图标。



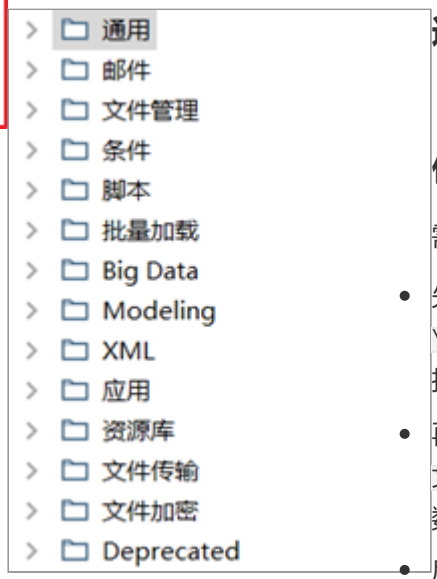
3、当运行结果为假时执行

- 当上一个作业项的执行结果为假或者没有成功执行时，执行下一个作业项
- 红色的连接线，上面有一个红色的停止图标



在图标上单击就可以对Hop进行设置

作业组件分类



通用组件

- 执行错误，显示执行错误消息框
- 执行成功，显示执行成功消息框

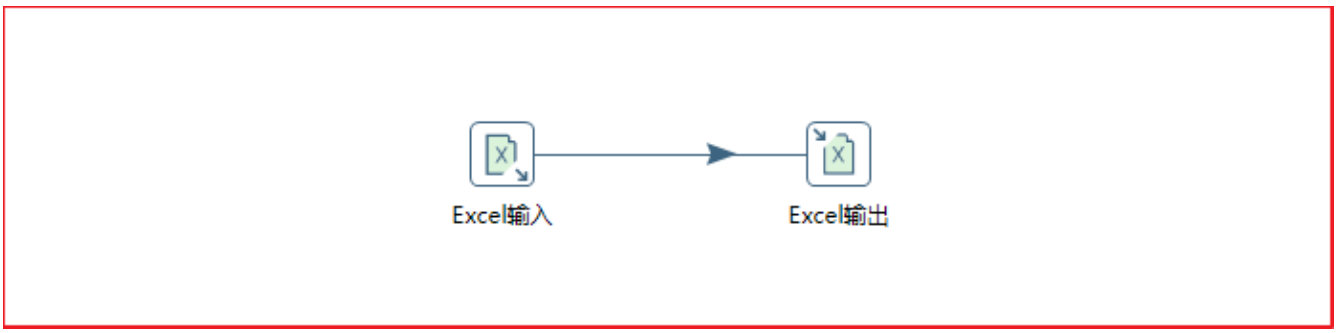
作业示例

需求：

- 先从 `资料\作业数据源\01Excel输入.xlsx` 读取数据，保存到Excel
- 再从 `资料\作业数据源\01文本文件输入.txt` 文本文件中读取数据，保存到Excel
- 启动作业执行

实现步骤：

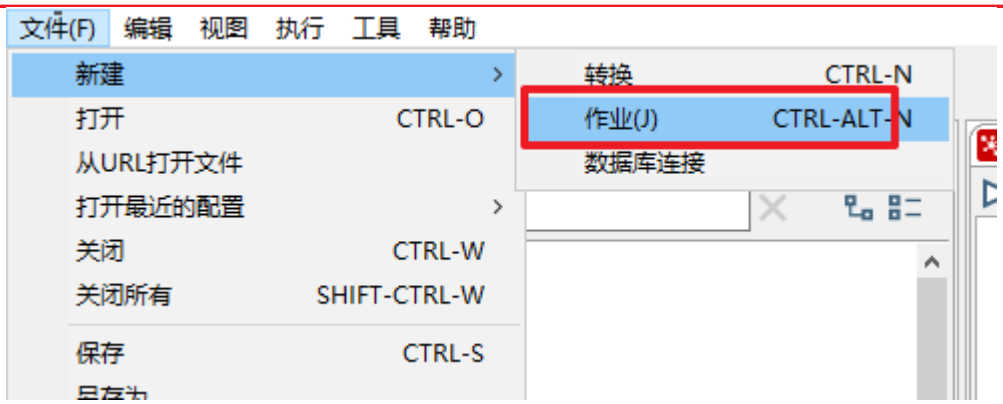
1、设计转换结构1（从Excel读取数据，保存到Excel）

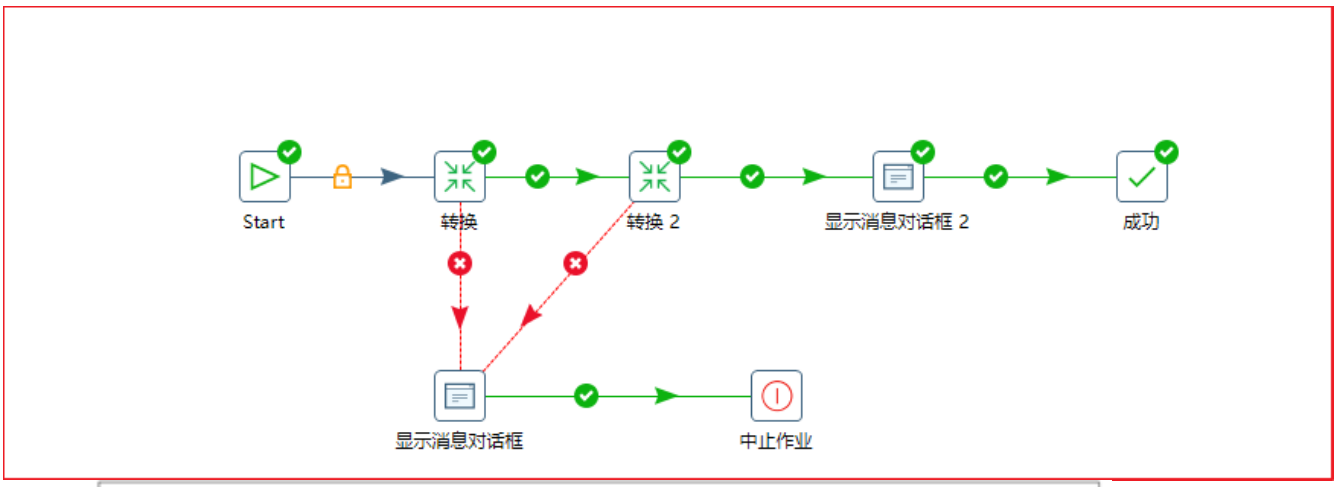


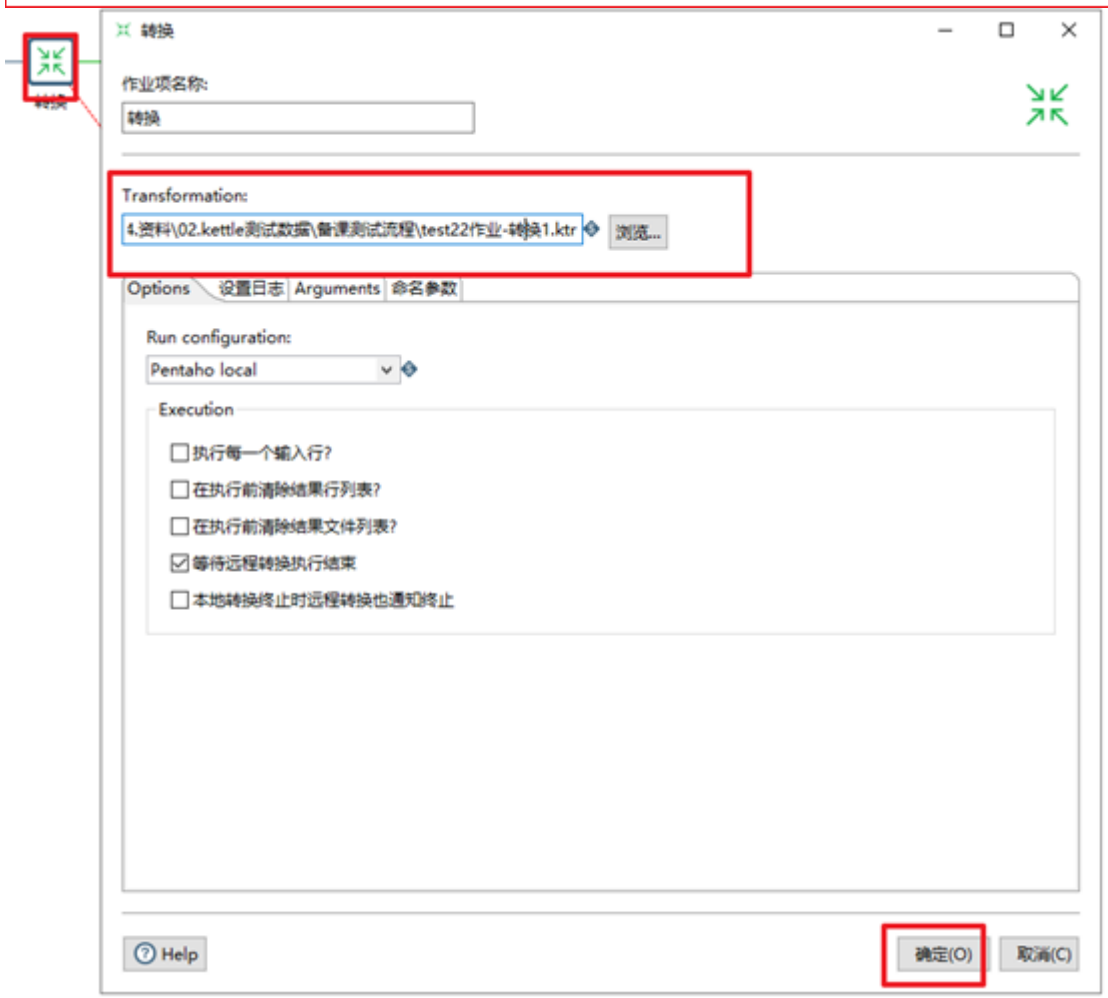
2、设计转换结构2（从文本文件中读取数据，保存到Excel）

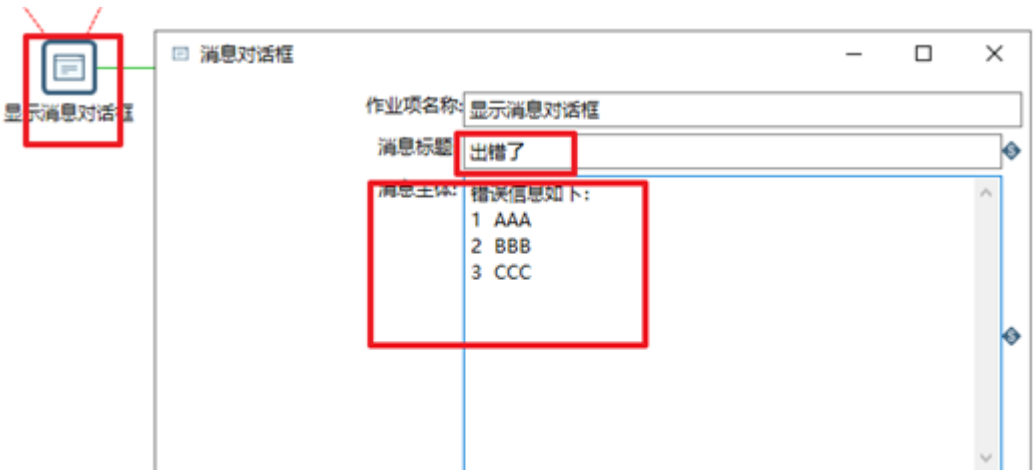
3、设计作业结构（先执行转换结构1、再执行转换结构2）

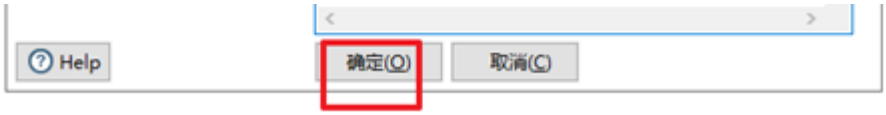

文本文件输入
Excel输出


文件(F)
编辑
视图
执行
工具
帮助
新建
打开 CTRL-O
从URL打开文件
打开最近的配置
关闭 CTRL-W
关闭所有 SHIFT-CTRL-W
保存 CTRL-S
转换 CTRL-N
作业(J) CTRL-ALT-N
数据库连接


Start
转换
转换 2
显示消息对话框 2
成功
显示消息对话框
中止作业


转换
作业项名称:
转换
Transformation:
4.资料\02.kettle测试数据\备课测试流程\test22作业-转换1.ktr
浏览...
Options
设置日志
Arguments
参名参数
Run configuration:
Pentaho local
Execution
执行每一个输入行?
在执行前清除结果行列表?
在执行前清除结果文件列表?
等待远程转换执行结束
本地转换终止时远程转换也通知终止
Help
确定(O)
取消(C)

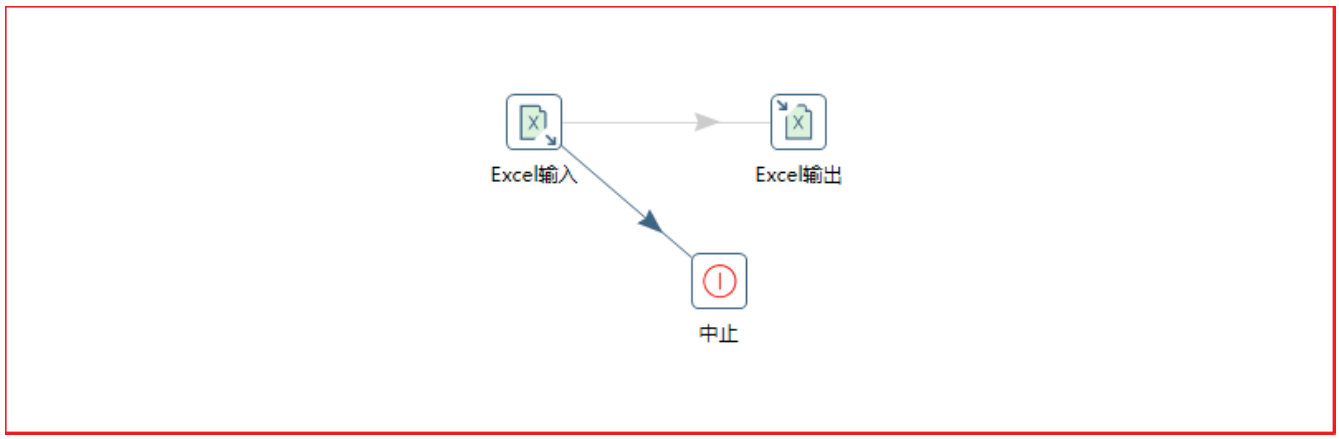

显示消息对话框
消息对话框
作业项名称:
显示消息对话框
消息标题:
出错了
消息主体:
错误信息如下:
1 AAA
2 BBB
3 CCC

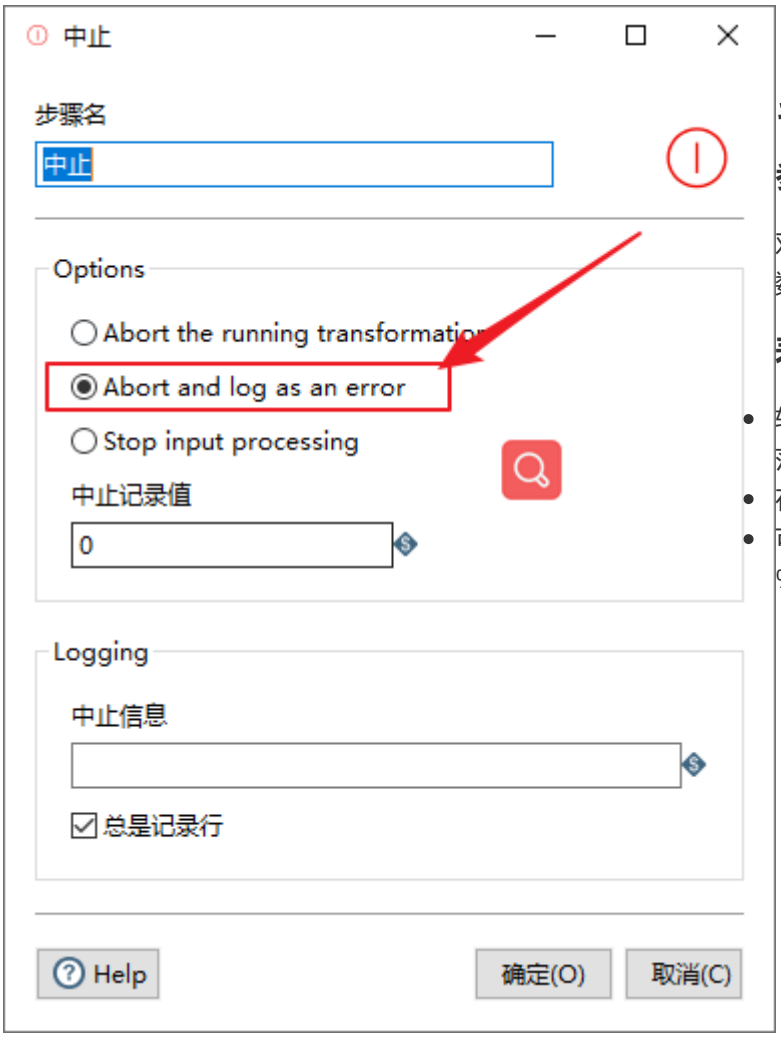


4、运行测试

5、错误测试

将第一个转换结构直接终止，并配置抛出一个错误





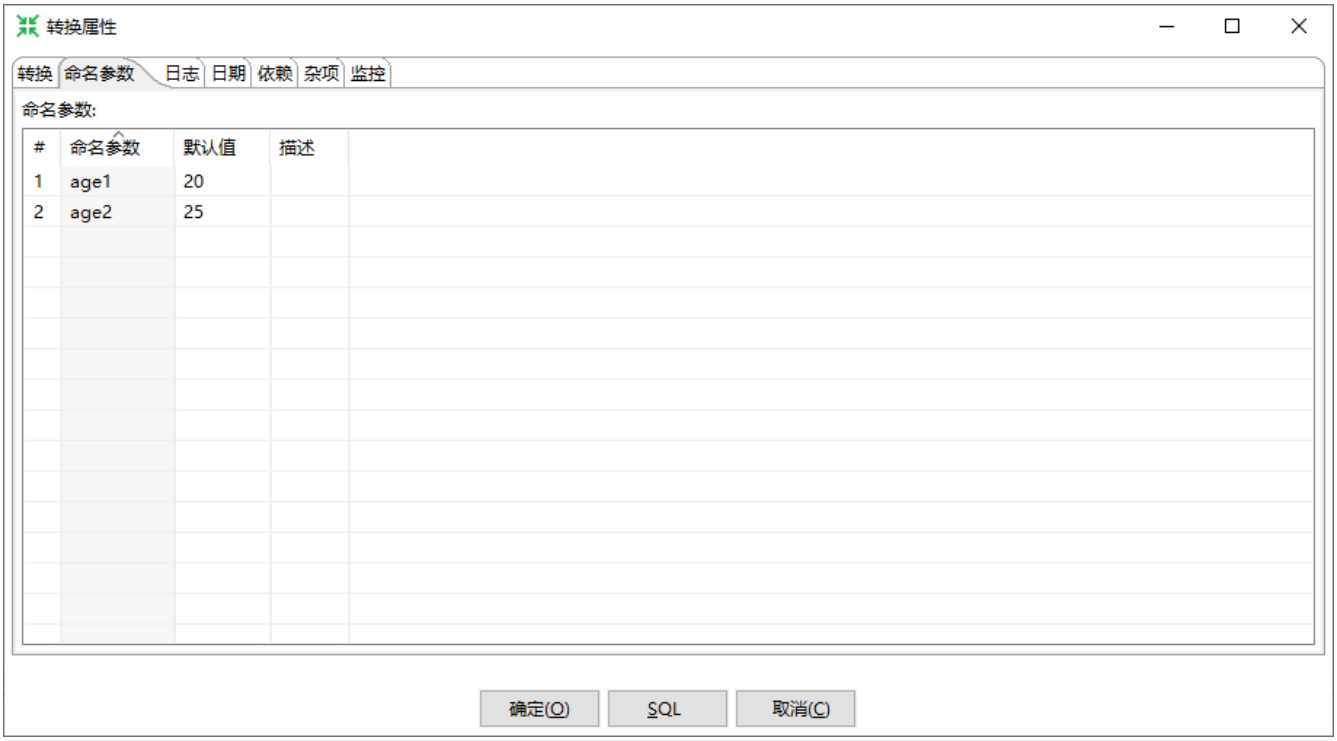
参数

参数的使用

对于ETL参数传递是一个很重要的环节，因为参数的传递会涉及到业务数据是如何抽取

表输入参数传递 - 转换命名参数

- 转换命名参数就是在转换内部定义的变量，作用范围是在转换内部
- 在转换的空白处双击左键，在转换属性中能看到
- 可以在表输入 SQL语句中使用 ${变量名} 或者 %%变量名%% 直接引用

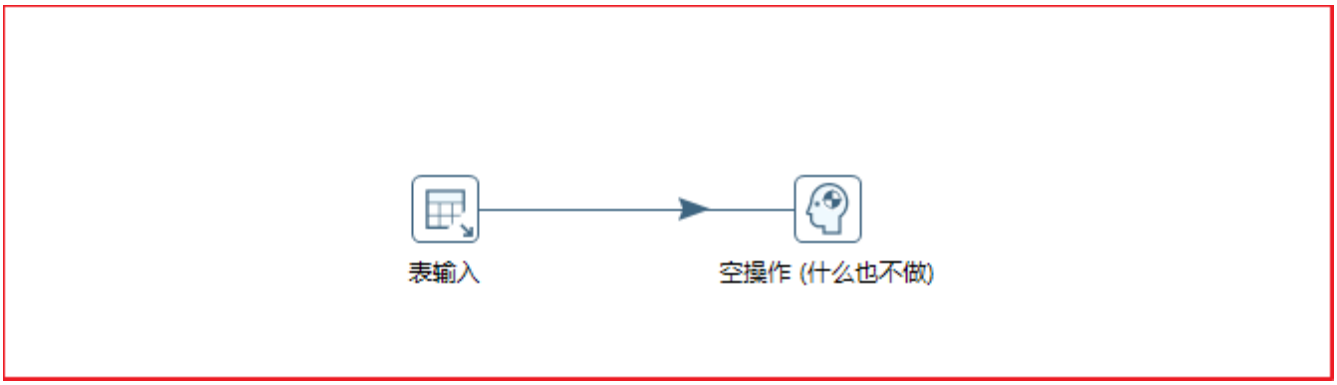


需求：

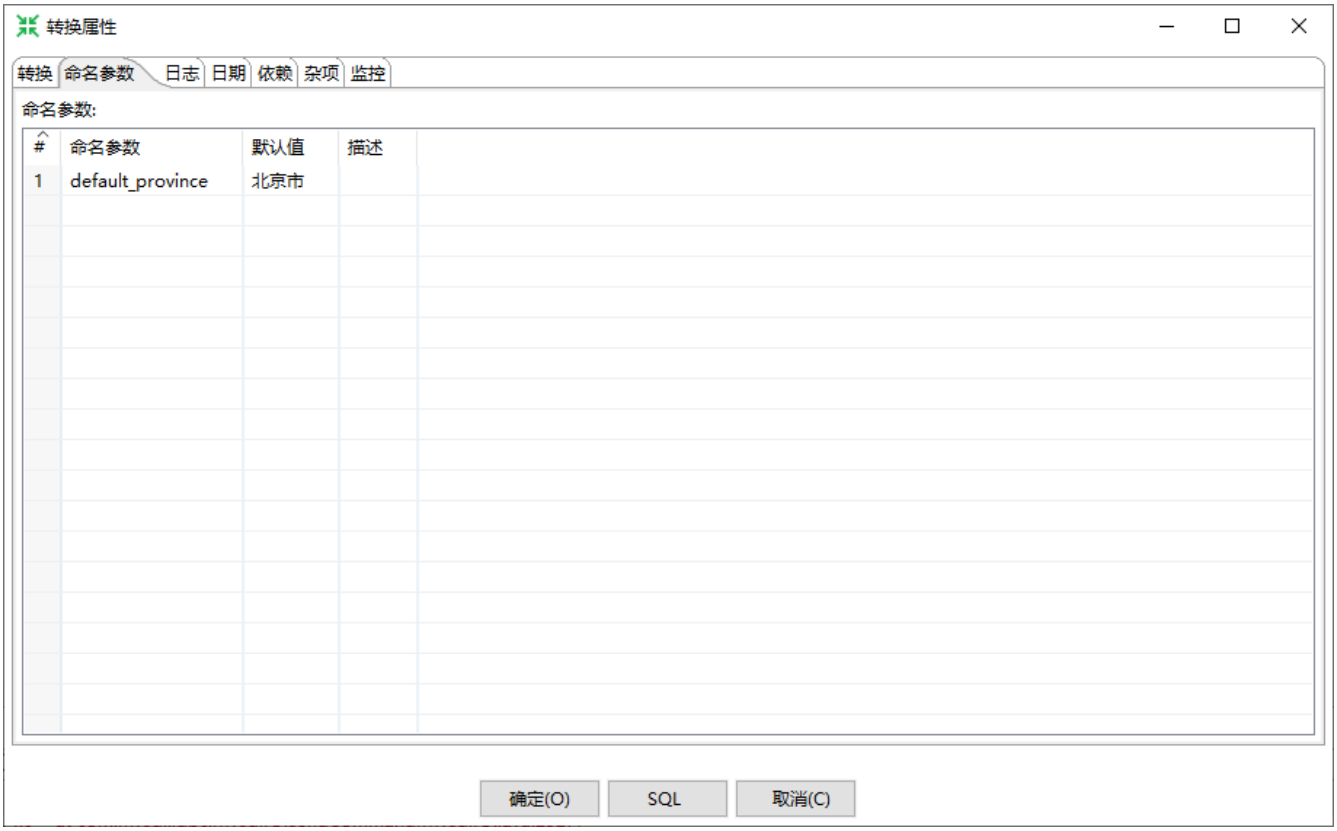
- 设置转换命名参数 default_province = 北京市
- 从t_user表中获取数据，满足条件 province=default_province，后续不要执行任何操作

实现步骤：

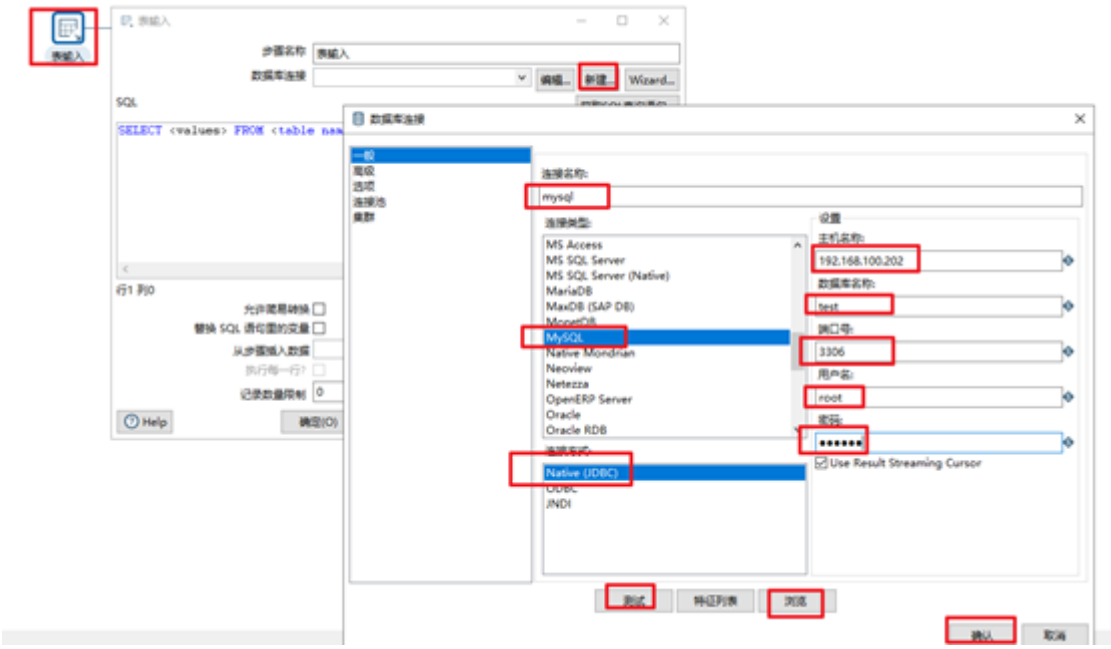
1、设计以下转换组件结构图

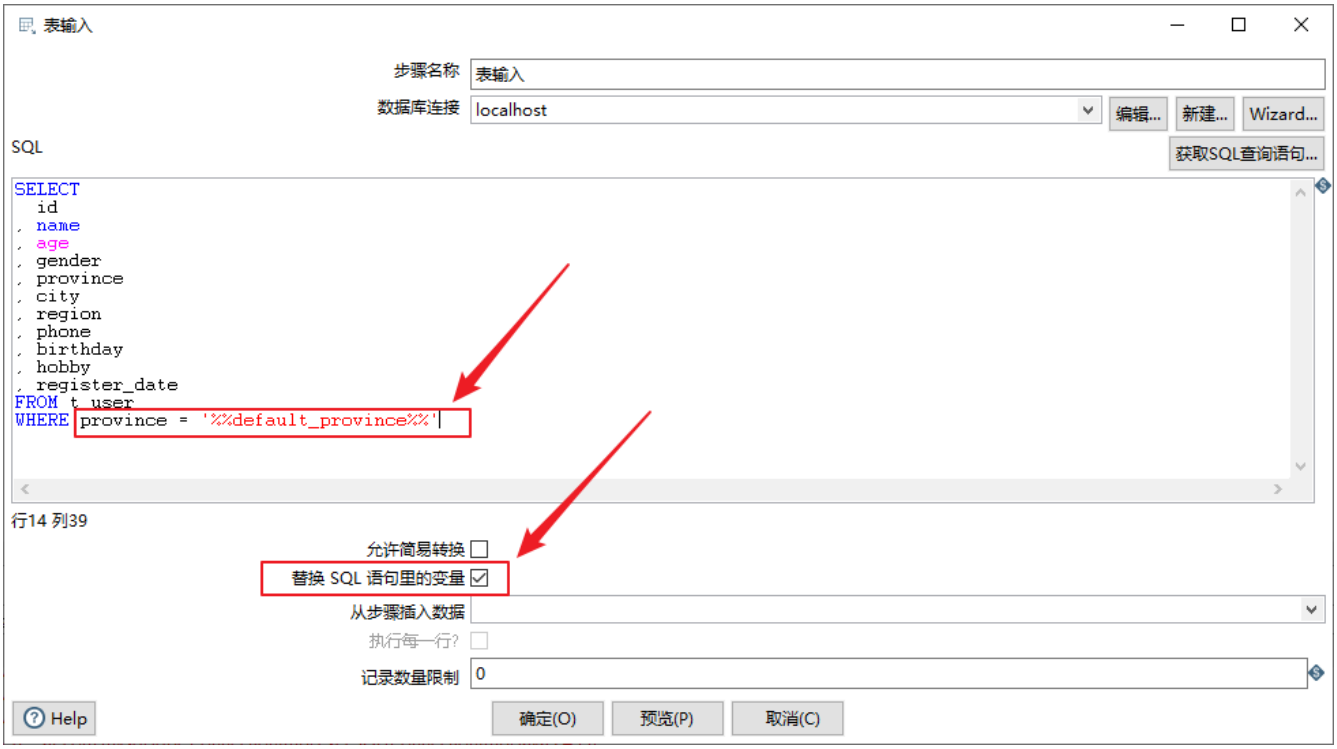


2、配置转换命名参数


转换属性
转换
命名参数
日志
日期
依赖
杂项
监控
命名参数:
#
命名参数
默认值
描述
1
default_province
北京市
确定(O)
SQL
取消(C)

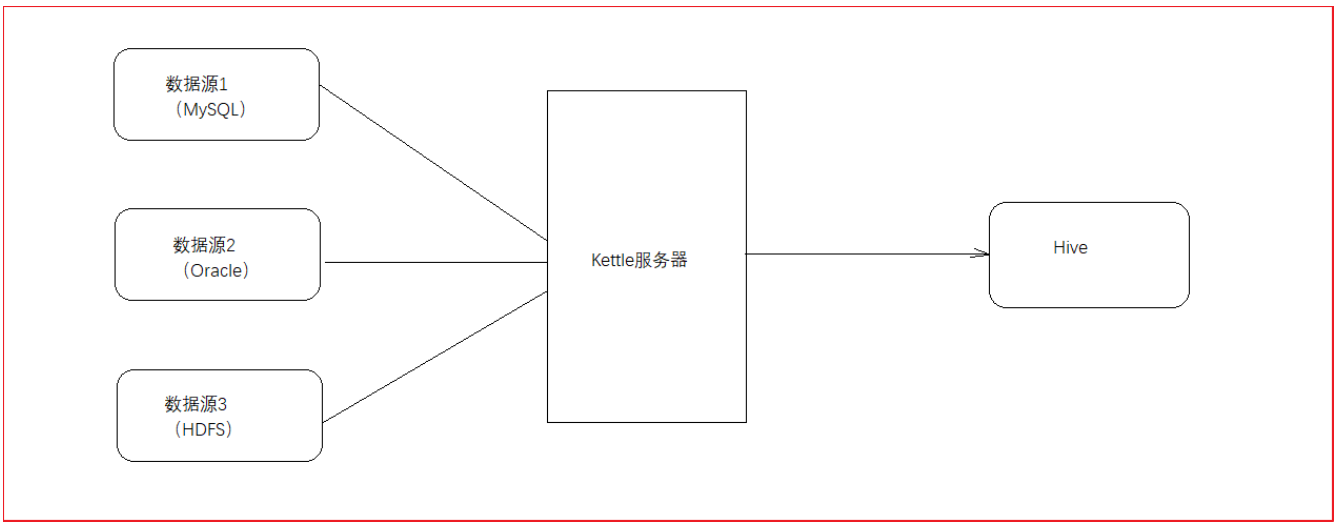
3、配置表输入组件


表输入
步骤名称
表输入
数据库连接
编辑...
新建...
Wizard...
SQL
SELECT <values> FROM <table nam
行1 列0
允许简易转换
替换 SQL 语句里的变量
从步骤插入数据
执行每一行?
记录数量限制
0
Help
确定(O)
数据库连接
一般
高级
选项
连接池
集群
连接名称:
mysql
连接类型:
MS Access
MS SQL Server
MS SQL Server (Native)
MariaDB
MaxDB (SAP DB)
MonetDB
MySQL
Native Mondrian
Neoview
Netezza
OpenERP Server
Oracle
Oracle RDB
连接方式:
Native (JDBC)
ODBC
JNDI
设置
主机名称:
192.168.100.202
数据库名称:
test
端口号:
3306
用户名:
root
密码:
Use Result Streaming Cursor
测试
特征列表
浏览
确认
取消



4、执行转换

Kettle Linux部署



Linux安装Kettle

1、用File Zilla将kettle上传到Linux服务器，并解压缩

2、在命令行执行

```
./pan.sh -version
./kitchen.sh -version
```

3、如果能够看到以下输出，表示kettle可以正确运行

```
2019/10/09 08:49:09 - Pan - Kettle version 8.2.0.0-342, build 8.2.0.0-342, build date :
2018-11-14 10.30.55
2019/10/09 08:49:09 - Pan - Start of run.
ERROR: No repository provided, can't load transformation.
```

```
2019/10/09 08:13:21 - Kitchen - Kettle version 8.2.0.0-342, build 8.2.0.0-342, build date
: 2018-11-14 10.30.55
2019/10/09 08:13:21 - Kitchen - Start of run.
ERROR: Kitchen can't continue because the job couldn't be loaded.
```

4、配置环境变量

```
# KETTLE
export KETTLE=/export/softwares/data-integration
export PATH=${KETTLE}:$PATH
```

Pan——转换执行引擎

pan.sh可以用来在服务器中执行一个转换

pan.sh的命令行参数:

```
-version：显示版本信息
-file：指定要运行的转换文件（XML文件）
-level：设置日志级别(Basic,Detailed,Debug,Rowlevel,Error,Nothing)
-log：指定日志文件
-param:key=value （该参数可以指定多个）覆盖之前指定的默认的命名参数
```

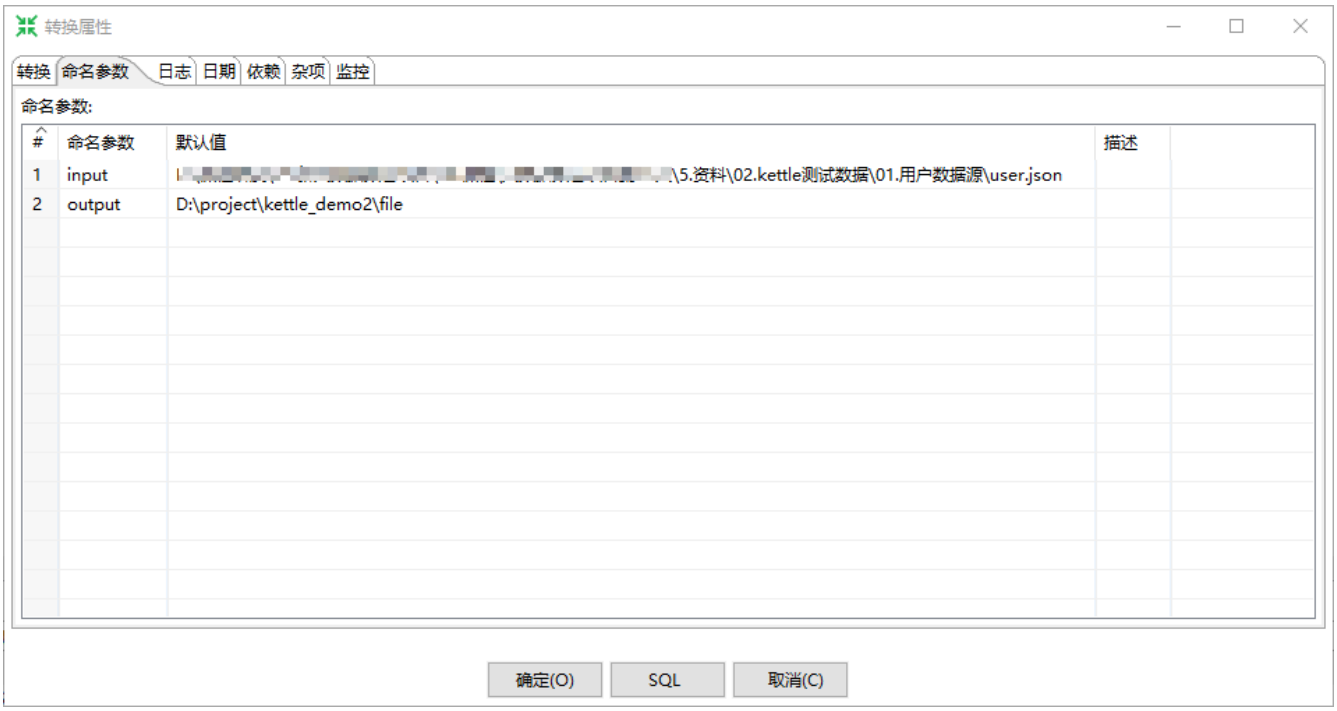
需求：

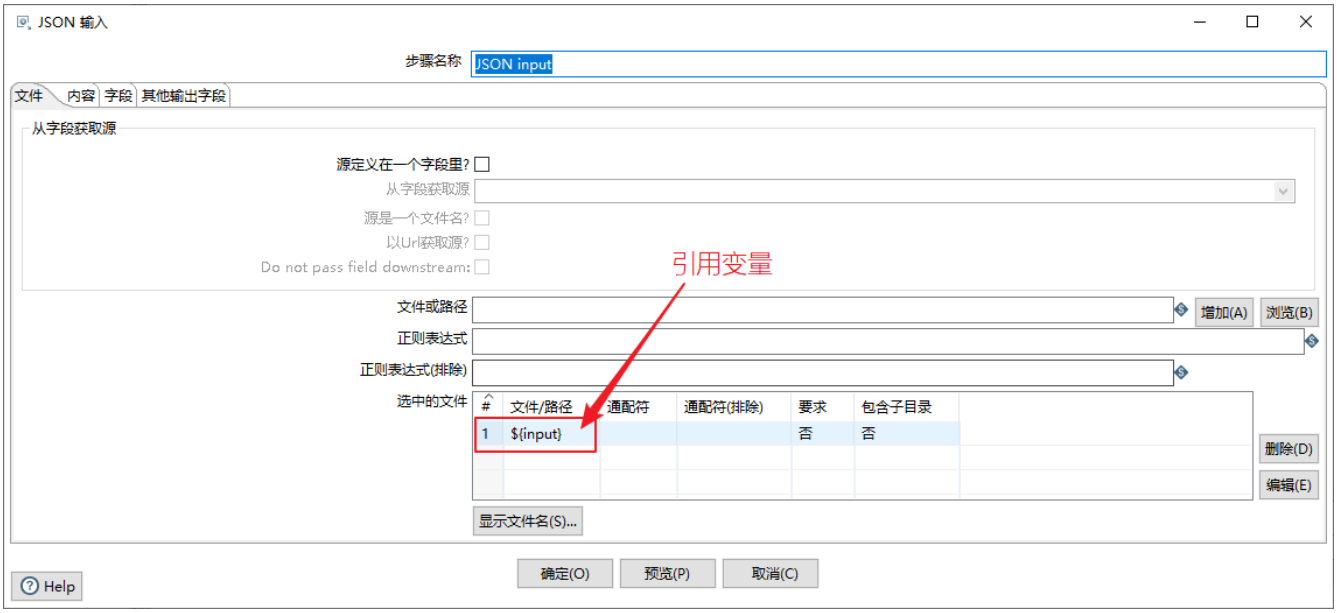
- 在Linux中，将 /root/kettle/user.json 数据抽取到 /root/kettle/user.xls 表格中

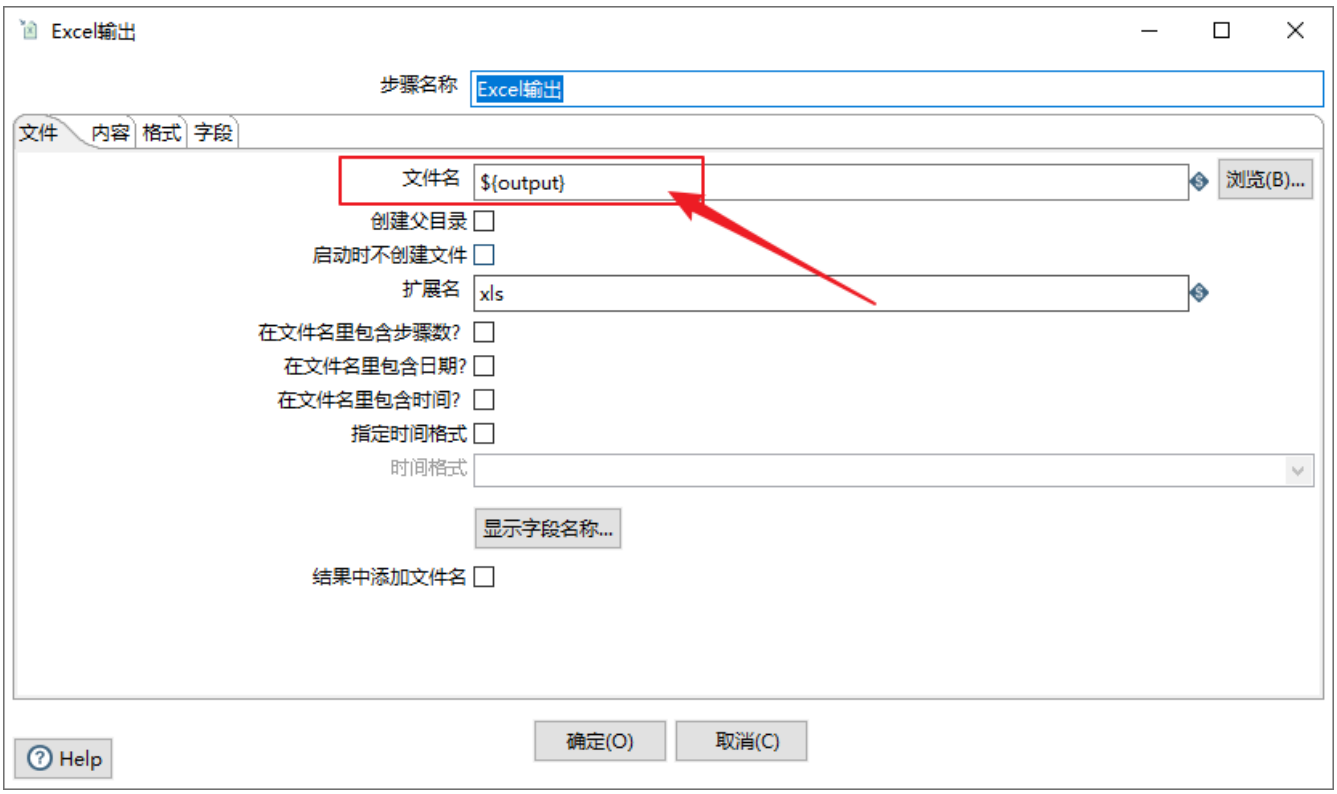
实现步骤：

1、在 windows 中开发转换，将 json数据抽取装载到 user.xls文件中

2、抽取路径参数，通过命令行指定 json数据文件路径，指定 user.xls 文件路径







3、将数据文件上传到 /root/kettle 目录

4、上传转换文件、json数据文件到Linux服务器

5、使用 pan.sh 执行转换

```
pan.sh -file 8.transform_param.ktr -level Basic -param:input=/root/kettle/user.json -
param:output=/root/kettle/output_user
```

Kitchen——作业执行引擎

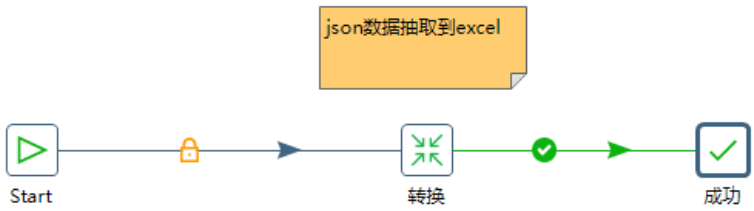
在Linux中，可以使用 kitchen.sh 来执行作业

需求：

- 执行JSON数据抽取到Excel中

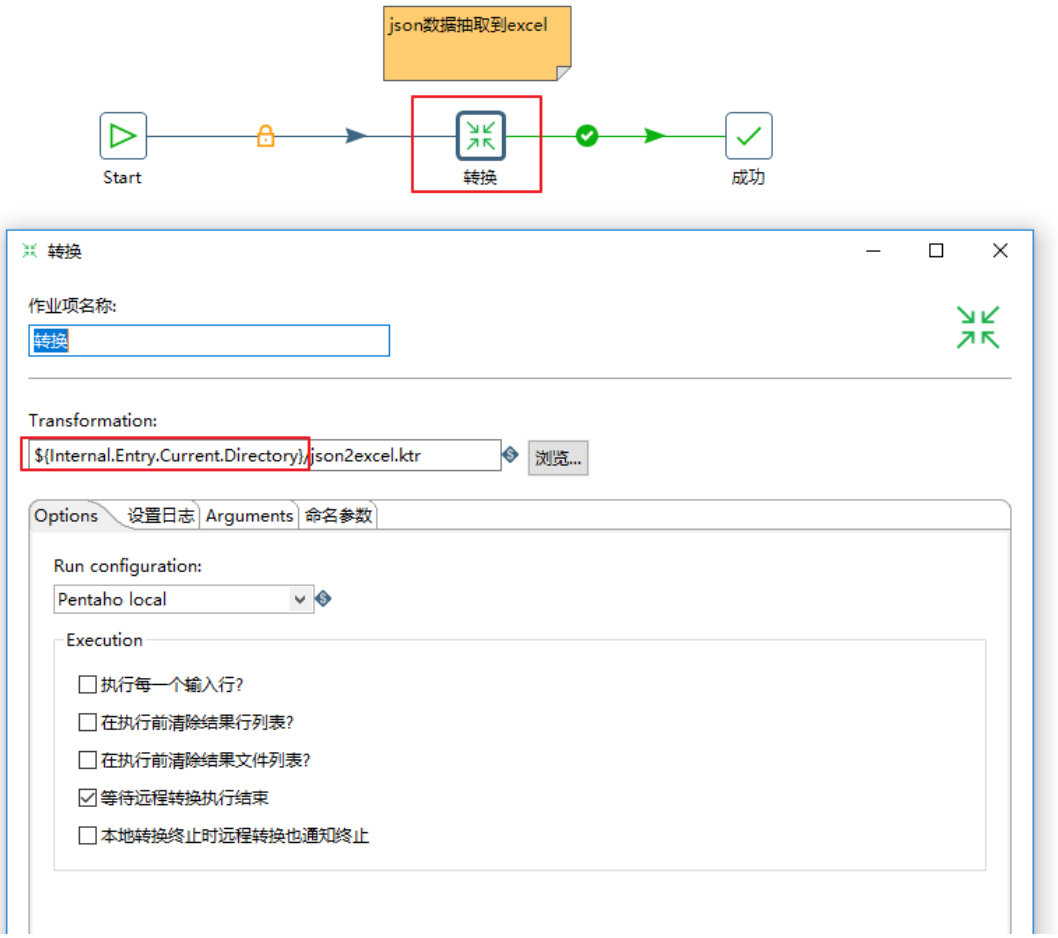
实现步骤：

1、在windows中开发作业



2、配置转换组件

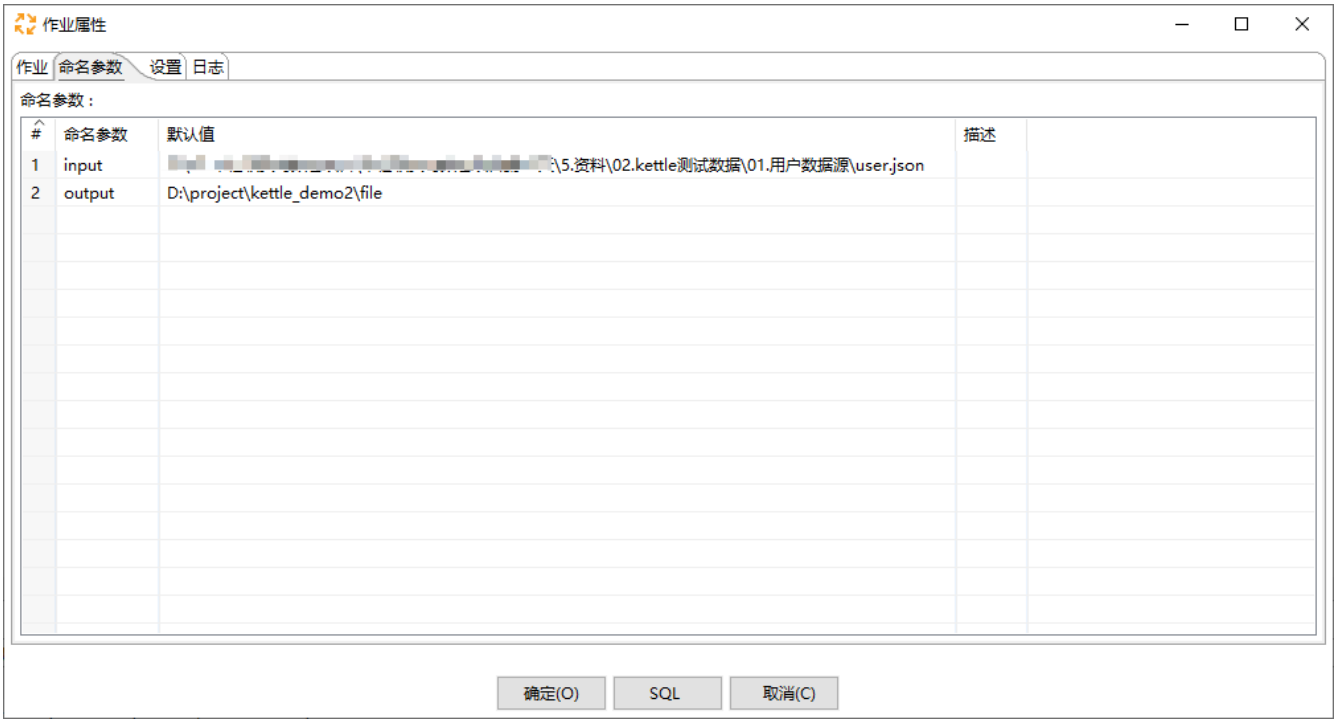
引入之前定义好的转换任务。



3 windows本地测试执行

4 、修改转换中的路径参数改为用变量来接收

5、配置作业命名参数



6、启动测试执行

6、上传JOB文件到Linux服务器的 `/root/kettle/` 目录

7、使用kitchen.sh执行作业

```
kitchen.sh -file job_transform.kjb -level Basic -param:input=/root/kettle/user.json -
param:output=/root/kettle/output_user
```