

尚硅谷大数据技术之 kettle

(作者：尚硅谷大数据研发部)

版本：V1.1

第 1 章 kettle 概述

1.1 什么是 kettle

Kettle 是一款开源的 ETL 工具，纯 java 编写，可以在 Window、Linux、Unix 上运行，绿色无需安装，数据抽取高效稳定。

1.2 Kettle 核心知识点

1.2.1 Kettle 工程存储方式

- 1) 以 XML 形式存储
- 2) 以资源库方式存储(数据库资源库和文件资源库)

1.2.2 Kettle 的两种设计



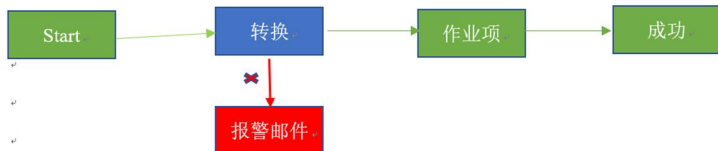
Kettle的两种设计



简述： Transformation（转换）：完成针对数据的基础转换。
Job（作业）：完成整个工作流的控制。

区别： (1) 作业是步骤流，转换是数据流。这是作业和转换最大的区别。
(2) 作业的每一个步骤，必须等到前面的步骤都跑完了，后面的步骤才会执行；
而转换会一次性把所有控件全部先启动（一个控件对应启动一个线程），
然后数据流会从第一个控件开始，一条记录、一条记录地流向最后的控件；

转换： 

作业： 

让天下没有难学的技术

1.2.3 Kettle 的组成



Kettle组成



1. 勺子(Spoon.bat/spoon.sh) : 是一个图形化的界面, 可以让我们用图形化的方式开发转换和作业。
windows选择Spoon.bat; Linux选择Spoon.sh
2. 煎锅 (Pan.bat/pan.sh) : 利用Pan可以用命令行的形式调用Trans
3. 厨房 (Kitchen.bat/kitchen.sh) : 利用Kitchen可以使用命令行调用Job
4. 菜单(Carte.bat/ Carte.sh): Carte是一个轻量级的Web容器, 用于建立专用、远程的ETL Server。



让天下没有难学的技术

1.3 kettle 特点



第 2 章 kettle 安装部署和使用

2.1 kettle 安装地址

官网地址

<https://community.hitachivantara.com/docs/DOC-1009855>

下载地址

更多 Java - 大数据 - 前端 - python 人工智能资料下载, 可百度访问: 尚硅谷官网

<https://sourceforge.net/projects/pentaho/files/Data%20Integration/>

2.2 Windows 下安装使用

2.2.1 概述

在实际企业开发中，都是在本地环境下进行 kettle 的 job 和 Transformation 开发的，可以在本地运行，也可以连接远程机器运行

2.2.2 安装

- 1) 安装 jdk
- 2) 下载 kettle 压缩包，因 kettle 为绿色软件，解压缩到任意本地路径即可
- 3) 双击 Spoon.bat，启动图形化界面工具，就可以直接使用了

2.2.3 案例

- 1) 案例一 把 stu1 的数据按 id 同步到 stu2，stu2 有相同 id 则更新数据

(1)在 mysql 中创建两张表

```
mysql> create database kettle;  
mysql> use kettle;  
mysql> create table stu1(id int,name varchar(20),age int);  
mysql> create table stu2(id int,name varchar(20));
```

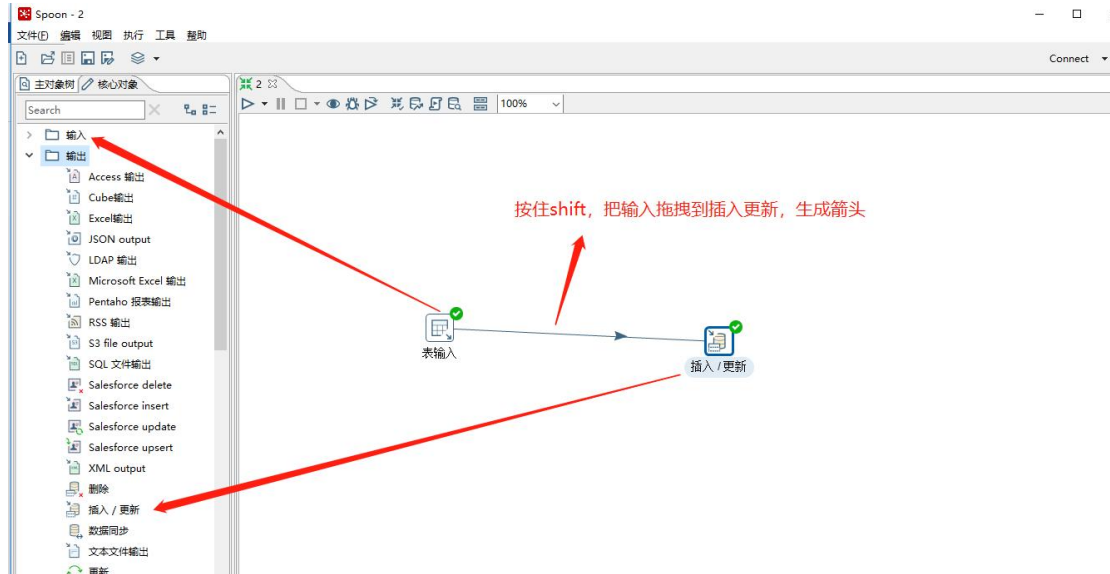
(2)往两张表中插入一些数据

```
mysql> insert into stu1 values(1001,'zhangsan',20),(1002,'lisi',18), (1003,'wangwu',23);  
mysql> insert into stu2 values(1001,'wukong');
```

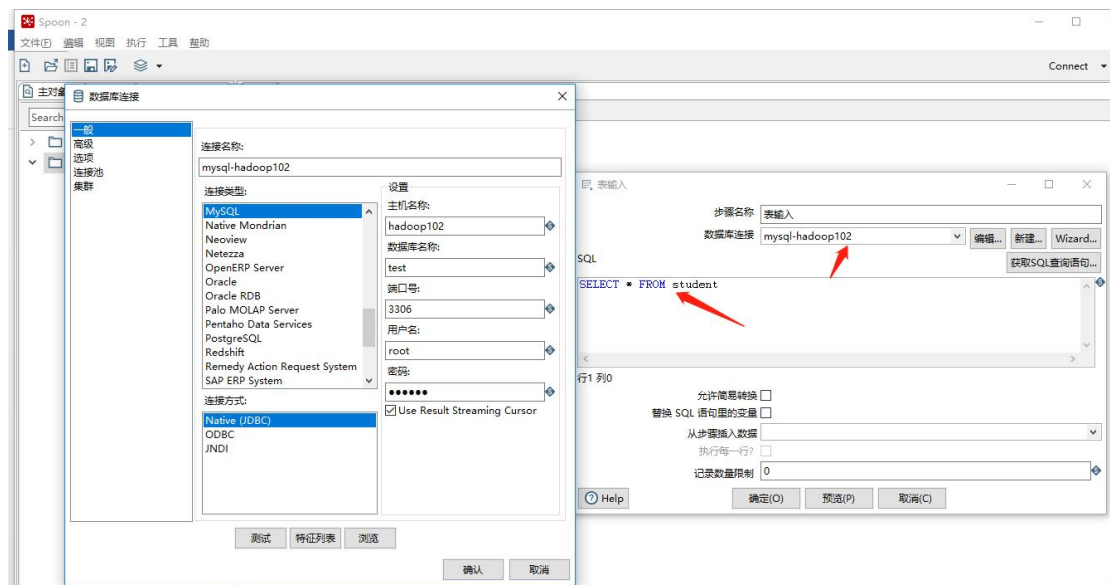
(3)在 kettle 中新建转换



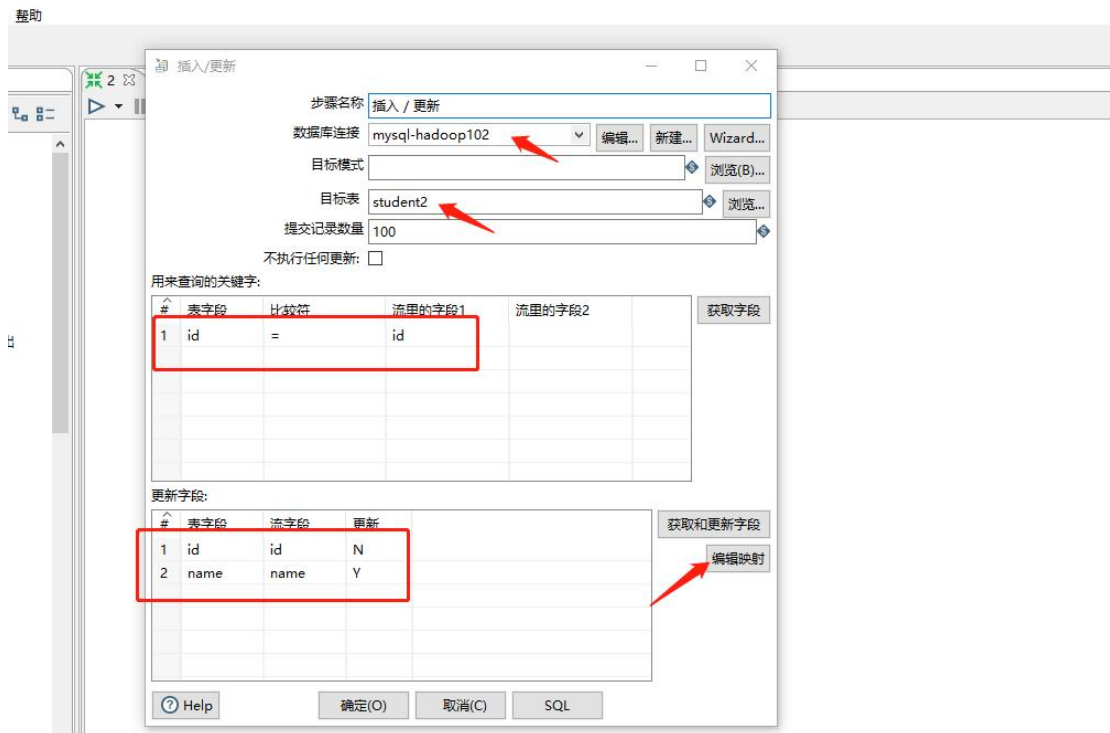
(4)分别在输入和输出中拉出表输入和插入/更新



(5)双击表输入对象，填写相关配置，测试是否成功

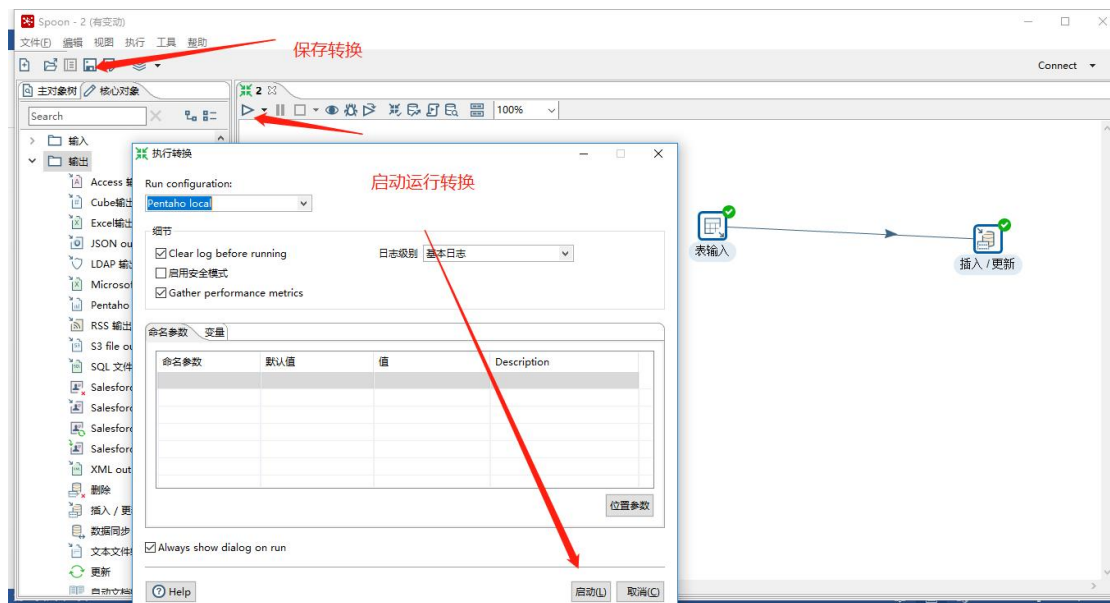


(6)双击 更新/插入对象，填写相关配置



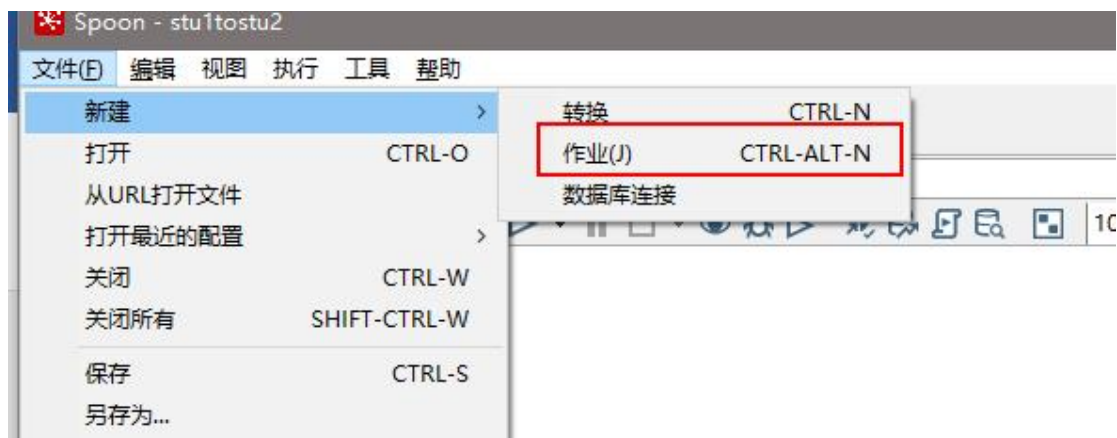
(7)保存转换，启动运行，去 mysql 表查看结果

注意：如果需要连接 mysql 数据库，需要要先将 mysql 的连接驱动包复制到 kettle 的根目录下的 lib 目录中，否则会报错找不到驱动。

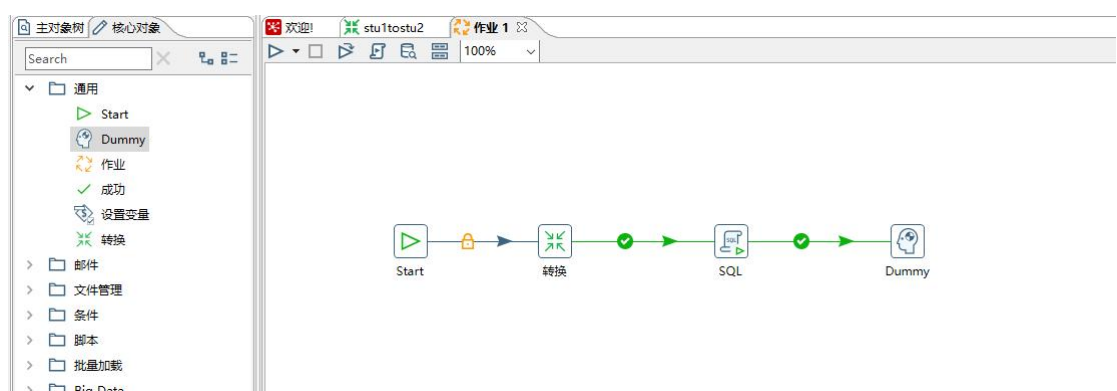


2) 案例 2：使用作业执行上述转换，并且额外在表 student2 中添加一条数据

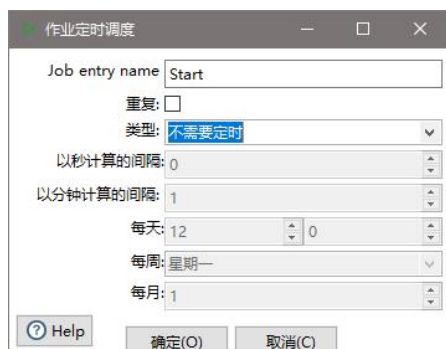
(1)新建一个作业



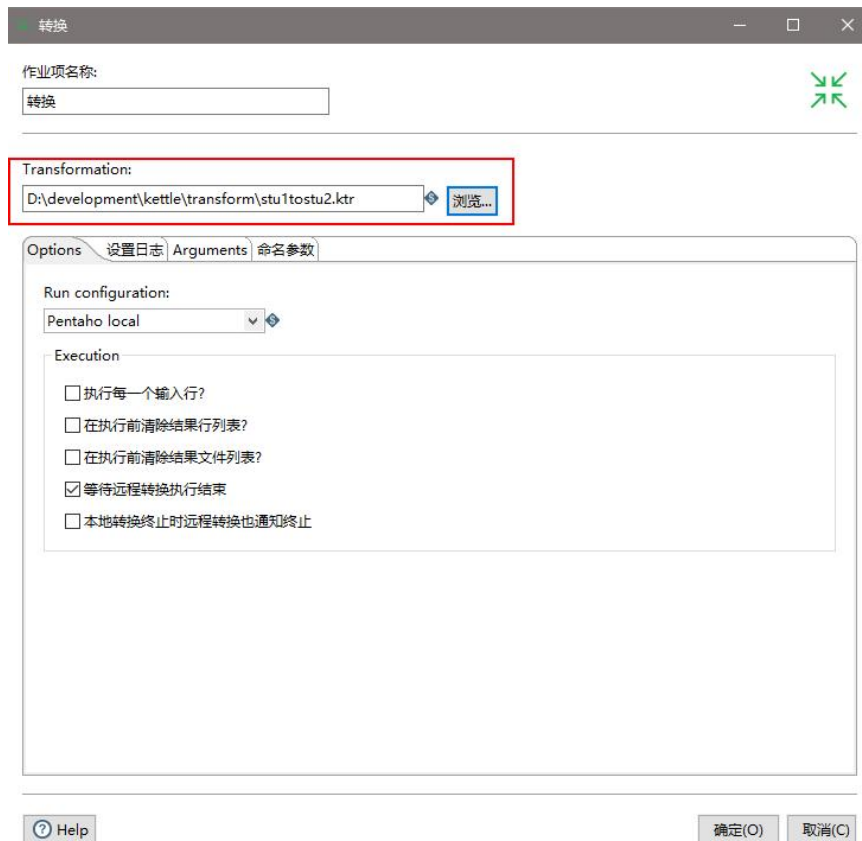
(2) 按图示拉取组件



(3) 双击 Start 编辑 Start



(4) 双击转换，选择案例 1 保存的文件



(5)双击 SQL，编辑 SQL 语句



(6)保存执行

3) 案例 3：将 hive 表的数据输出到 hdfs

(1)因为涉及到 hive 和 hbase 的读写，需要修改相关配置文件。

修改解压目录下的 data-integration\plugins\pentaho-big-data-plugin 下的 plugin.properties，设置 active.hadoop.configuration=hdp26，并将如下配置文件拷贝到 data-integration\plugins\pentaho-big-data-plugin\hadoop-configurations\hdp26 下

享 查看

« atguigu_work » beike » kettle » pdi-ce-8.2.0.0-342 » data-integration » plugins » pentaho-big-data-plugin » hadoop-configurations » hdp26

名称	修改日期	类型	大小
lib	2018/11/14 18:48	文件夹	
config.properties	2018/11/14 17:20	PROPERTIES 文件	1 KB
core-site.xml	2019/4/12 14:20	XML 文档	2 KB
hbase-site.xml	2019/4/13 23:08	XML 文档	2 KB
hdfs-site.xml	2019/4/12 14:19	XML 文档	2 KB
hive-site.xml	2019/4/12 14:20	XML 文档	2 KB
mapred-site.xml	2019/4/12 14:19	XML 文档	2 KB
PentahoHadoopShim_hdp26_OSS_Lic...	2018/11/14 17:20	Chrome HTML D...	1,363 KB
pentaho-hadoop-shims-hdp26-8.2.2...	2018/11/14 17:38	Executable Jar File	392 KB
pentaho-hadoop-shims-hdp26-hbas...	2018/11/14 17:38	Executable Jar File	9 KB
yarn-site.xml	2019/4/12 14:19	XML 文档	2 KB

(2)启动 hdfs, yarn, hbase 集群的所有进程, 启动 hiveserver2 服务

```
[atguigu@hadoop102 ~]$ /opt/module/hadoop-2.7.2/sbin/start-all.sh
开启 HBase 前启动 Zookeeper
[atguigu@hadoop102 ~]$ /opt/module/hbase-1.3.1/bin/start-hbase.sh
[atguigu@hadoop102 ~]$ /opt/module/hive/bin/hiveserver2
```

(3)进入 beeline, 查看 10000 端口开启情况

```
[atguigu@hadoop102 ~]$ /opt/module/hive/bin/beeline
Beeline version 1.2.1 by Apache Hive
beeline> !connect jdbc:hive2://hadoop102:10000 (回车)
Connecting to jdbc:hive2://hadoop102:10000
Enter username for jdbc:hive2://hadoop102:10000: atguigu (输入 atguigu)
Enter password for jdbc:hive2://hadoop102:10000: (直接回车)
Connected to: Apache Hive (version 1.2.1)
Driver: Hive JDBC (version 1.2.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://hadoop102:10000> (到了这里说明成功开启 10000 端口)
```

(4)创建两张表 dept 和 emp

```
CREATE TABLE dept(deptno int, dname string,loc string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t';

CREATE TABLE emp(
empno int,
ename string,
job string,
mgr int,
hiredate string,
sal double,
comm int,
deptno int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t';
```

(5)插入数据

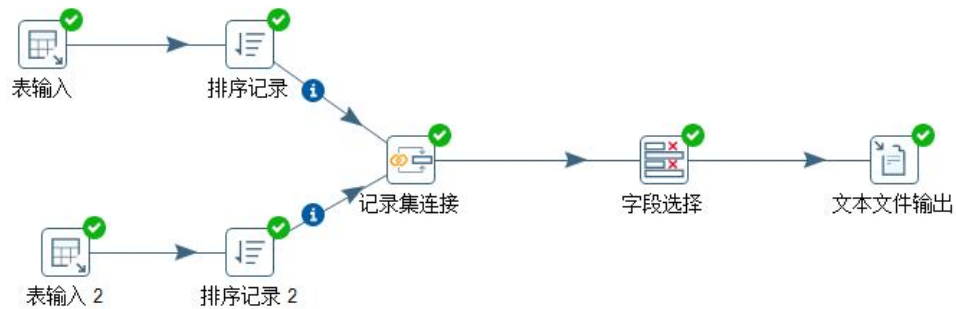
```
insert into dept values(10,'accounting','NEW YORK'),(20,'RESEARCH','DALLAS'),(30,'SALES','CHICAGO'),(40,'OPERATIONS','BOSTON');

insert into emp values
```

更多 Java - 大数据 - 前端 -python 人工智能资料下载, 可百度访问: 尚硅谷官网


```
(7369,'SMITH','CLERK',7902,'1980-12-17',800,NULL,20),
(7499,'ALLEN','SALESMAN',7698,'1980-12-17',1600,300,30),
(7521,'WARD','SALESMAN',7698,'1980-12-17',1250,500,30),
(7566,'JONES','MANAGER',7839,'1980-12-17',2975,NULL,20);
```

(6)按下图建立流程图



(7)设置表输入，连接 hive

表输入

步骤名称: 表输入

数据库连接: hive

SQL: `select * from dept`

行1 列0

允许简易转换 ☐

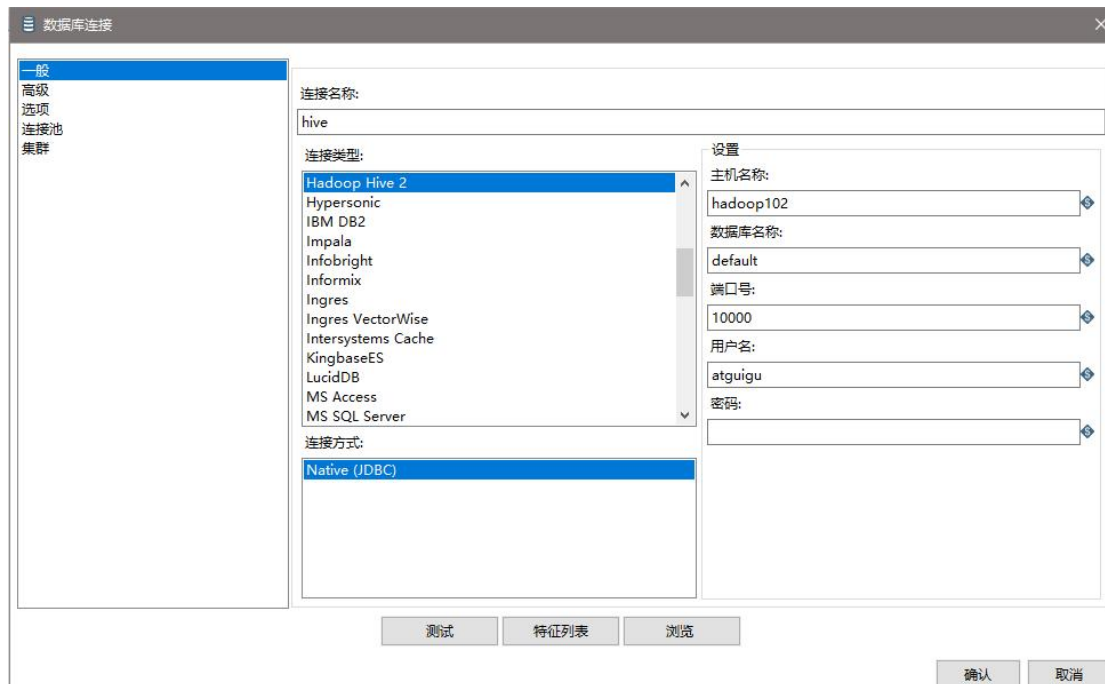
替换 SQL 语句里的变量 ☐

从步骤插入数据: [下拉菜单]

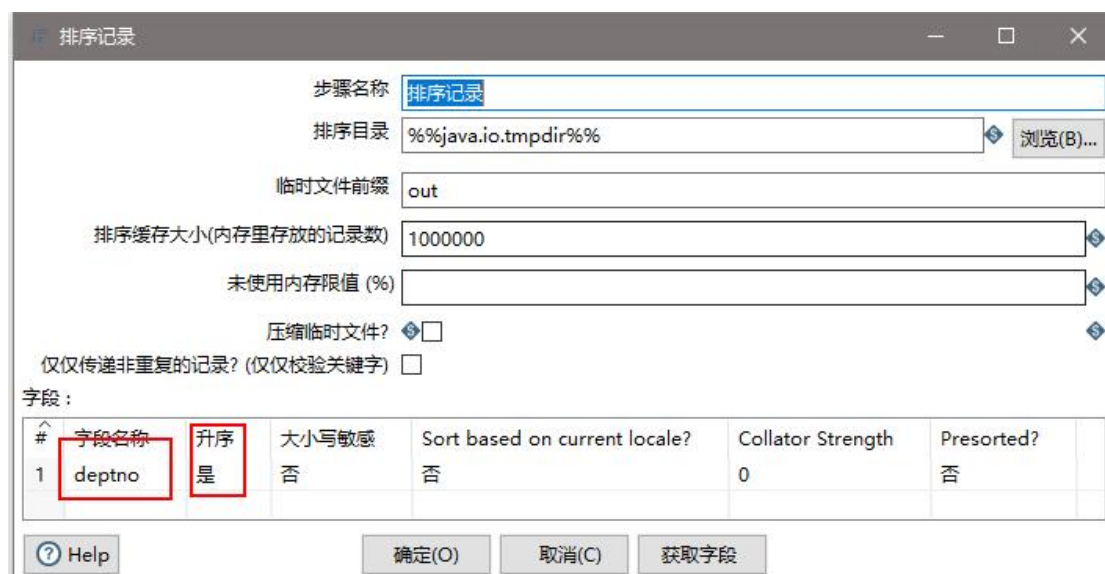
执行每一行? ☐

记录数量限制: 0

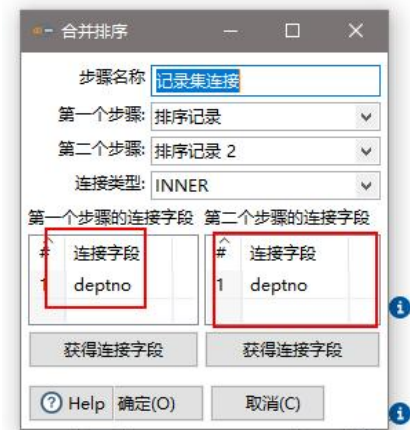
Help 确定(O) 预览(P) 取消(C)



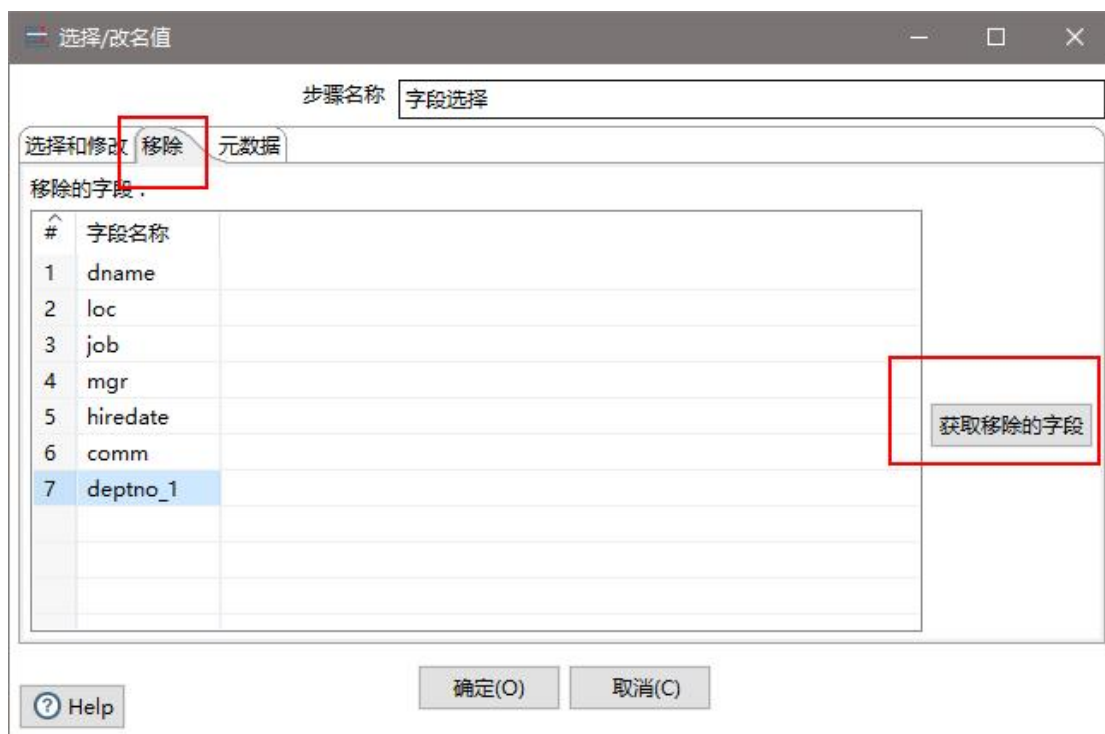
(8)设置排序属性



(9)设置连接属性



(10)设置字段选择



(11)设置文件输出

文本文件输出

步骤名称: 文本文件输出

文件 内容 字段

文件名称: hdfs://atguigu@hadoop102:9000/out 浏览(B)...

输出传递到servlet ☐

创建父目录 ☒

启动时不创建文件 ☒

从字段中获取文件名? ☐

文件名字段: ▼

扩展名: txt ▼

文件名里包含步骤数? ☐

文件名里包含数据分区号? ☐

文件名里包含日期? ☐

文件名里包含时间? ☐

指定日期时间格式: ▼

显示文件名...

结果中添加文件名 ☒

Help 确定(O) 取消(C)

Open File

Location: HDFS ▼

Connection

Hadoop Cluster: ▼ Edit... New...

Open from Folder: hdfs:// ▼ +

Name

Filter: 文本文件 ▼

OK Cancel

Hadoop Cluster

Cluster name: hadoop

Storage: HDFS ▼

HDFS

Hostname: hadoop102 ▼ Port: 9000 ▼

Username: atguigu ▼ Password: ▼

JobTracker

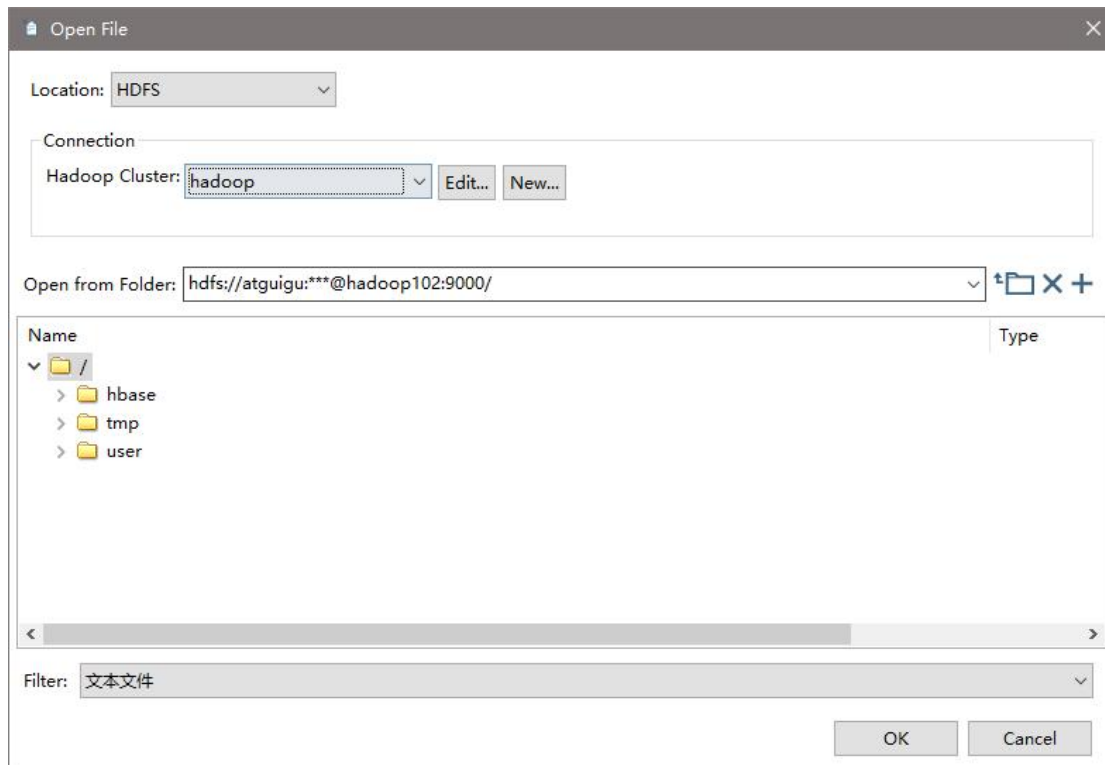
Hostname: hadoop103 ▼ Port: 8032 ▼

ZooKeeper

Hostname: hadoop102 ▼ Port: 2181 ▼

Oozie

Help 测试(T) 确定(O) 取消(C)



文件名称

输出传送到servlet ☐

创建父目录 ☒

启动时不创建文件 ☒

从字段中获取文件名? ☐

文件名字段

扩展名

文件名里包含步骤数? ☐

文件名里包含数据分区号? ☐

文件名里包含日期? ☐

文件名里包含时间? ☐

指定日期时间格式 ☐

日期时间格式

结果中添加文件名 ☒

The screenshot shows the 'Text File Output' dialog box in Oracle SQL Developer. The 'Field' tab is selected, showing a table of fields to be exported. The fields listed are:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Grouping	Remove String Padding
1	deptno	Integer	###0;-###0	9	0		.	,	不去掉空格
2	dname	String	###0;-###0	2147483647	0		.	,	不去掉空格
3	empno	Integer	###0;-###0	9	0		.	,	不去掉空格
4	ename	String	###0;-###0	2147483647	0		.	,	不去掉空格
5									

At the bottom of the dialog, there are two buttons: 'Fetch Field' (获取字段) and 'Minimum Width' (最小宽度). A tooltip for the 'Minimum Width' button indicates it 'Sets the output to non-padded width.'

(12)保存并运行查看 hdfs

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	atguigu	supergroup	0 B	Tue Apr 30 09:03:52 +0800 2019	0	0 B	hbase
-rw-r--r--	atguigu	supergroup	123 B	Fri May 10 16:34:58 +0800 2019	3	128 MB	out.txt
drwxrwx---	atguigu	supergroup	0 B	Fri Apr 26 14:45:09 +0800 2019	0	0 B	tmp
drwxr-xr-x	atguigu	supergroup	0 B	Fri Apr 26 14:43:55 +0800 2019	0	0 B	user

4)案例 4： 读取 hdfs 文件并将 sal 大于 1000 的数据保存到 hbase 中

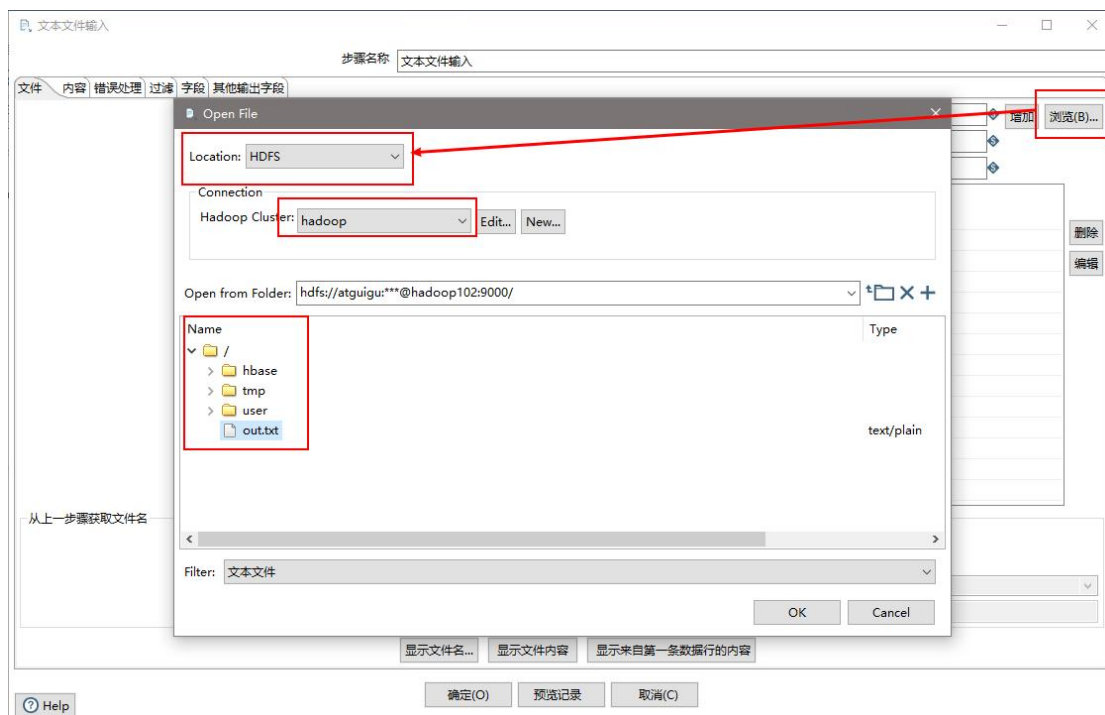
(1) 在 HBase 中创建一张表用于存放数据

```
[atguigu@hadoop102 ~]$ /opt/module/hbase-1.3.1/bin/hbase shell
hbase(main):004:0> create 'people','info'
```

(2)按下图建立流程图



(3)设置文件输入，连接 hdfs



文件或目录

hdfs://atguigu:123@hadoop102:9000/out.txt

增加

规则表达式

正则表达式(排除)

选中的文件:

#	文件/目录	通配符	通配符号(排除)	要求	包含子目录
1	hdfs://atguigu@hadoop102:9000/out.txt			否	否

从上一步骤获取文件名

从以前的步骤接受文件名

☐

从以前的步骤接受字段名

☐

步骤读取的文件名来自

在输入里的字段被当作文件名

显示文件名...

显示文件内容

显示来自第一条数据行的内容

(4)设置过滤记录

过滤记录

步骤名称: 过滤记录

发送true数据给步骤: HBase output

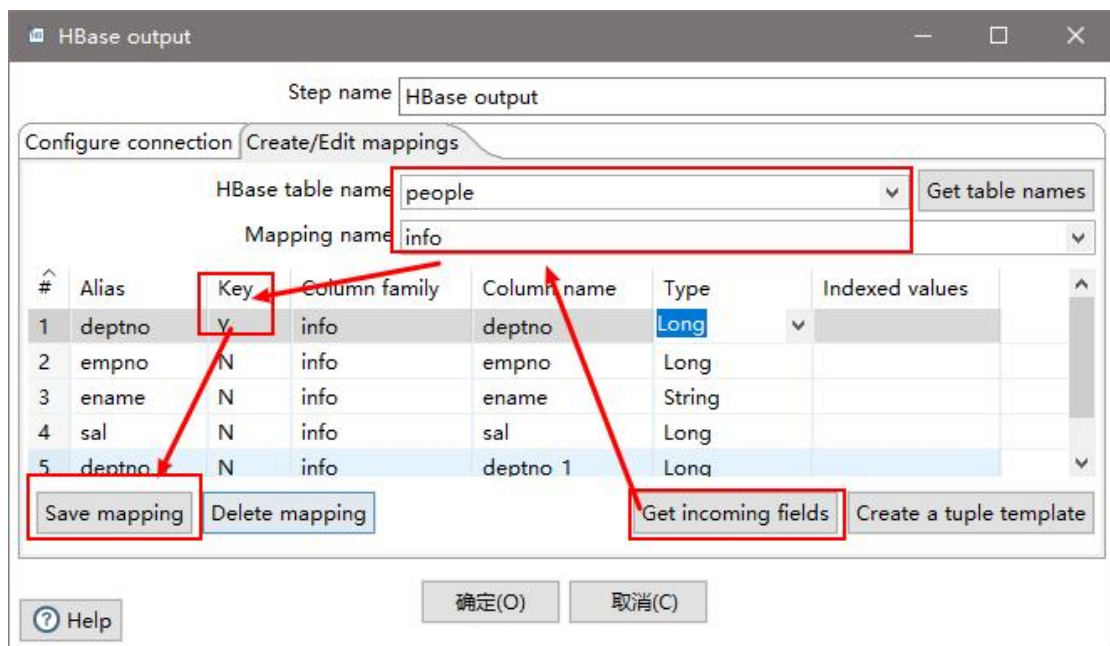
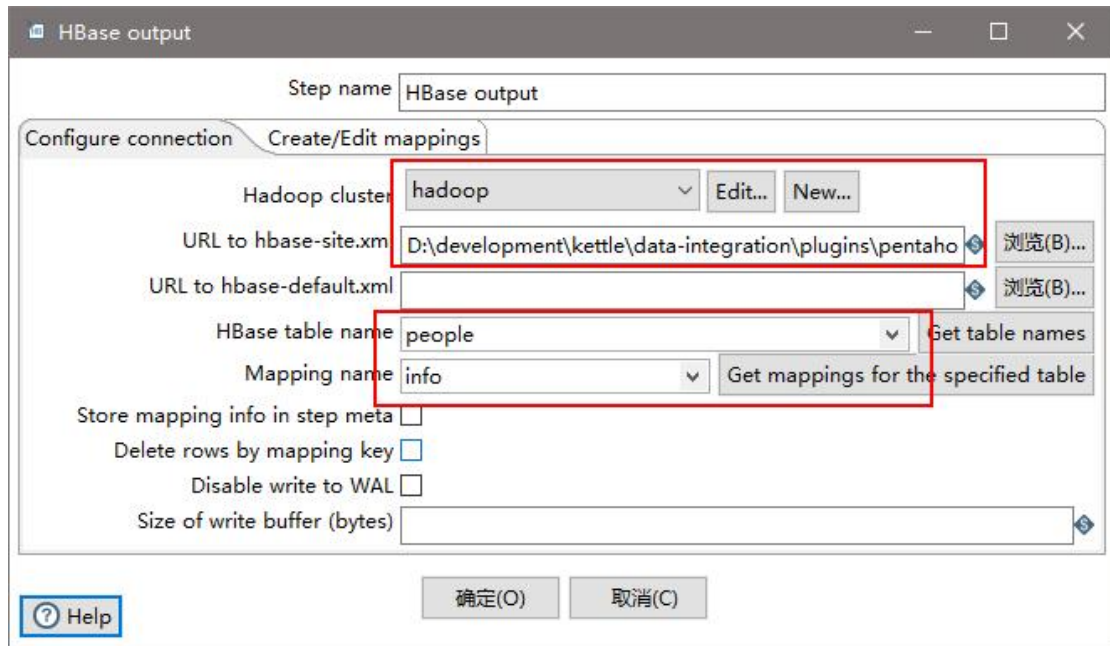
发送false数据给步骤: 空操作 (什么也不做)

条件:

sal > 1000 (Int)

Help 确定(O) 取消(C)

(5)设置 HBase output



注意：若报错没有权限往 hdfs 写文件，在 Spoon.bat 中第 119 行添加参数

"-DHADOOP_USER_NAME=atguigu" "-Dfile.encoding=UTF-8"

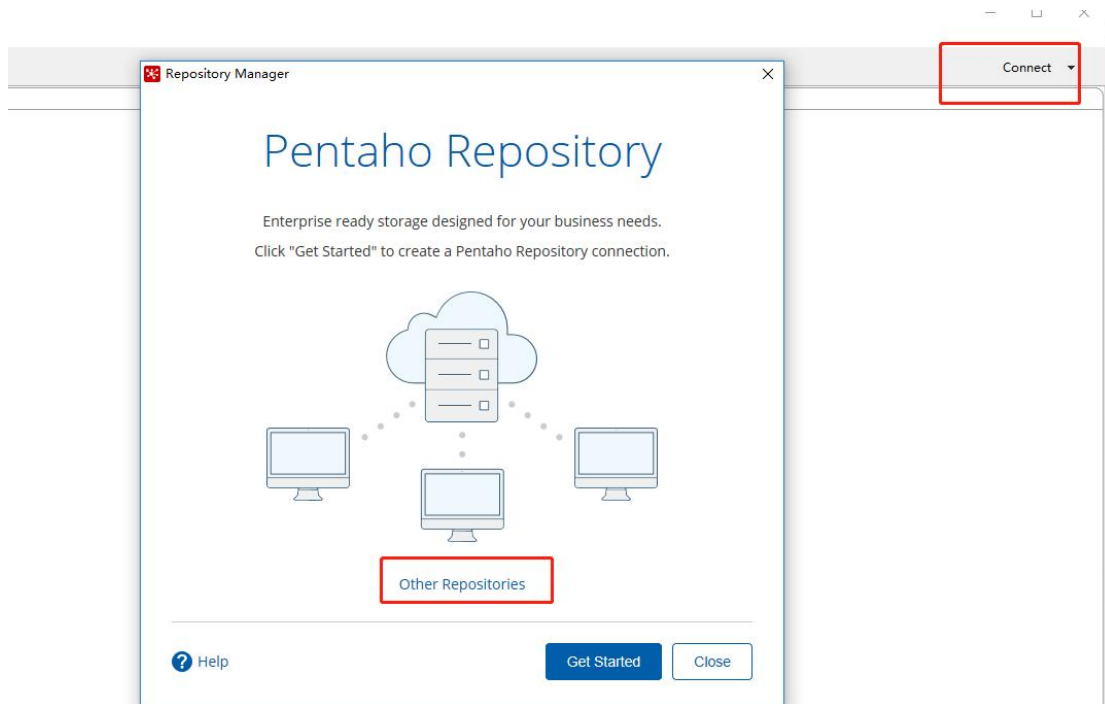
2.3 创建资源库

2.3.1 数据库资源库

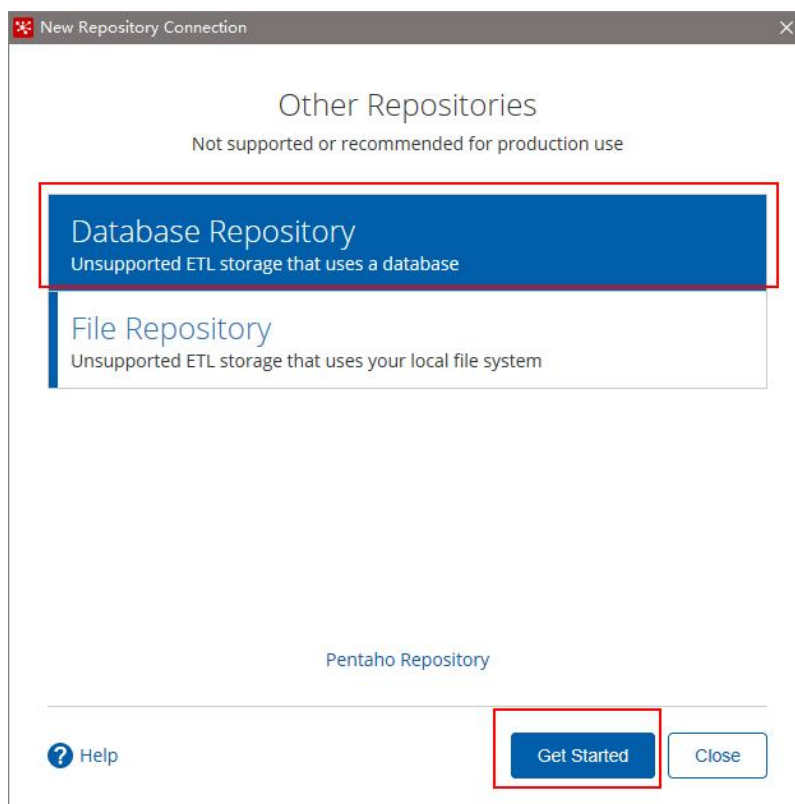
数据库资源库是将作业和转换相关的信息存储在数据库中，执行的时候直接去数据库读取信息，很容易跨平台使用

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

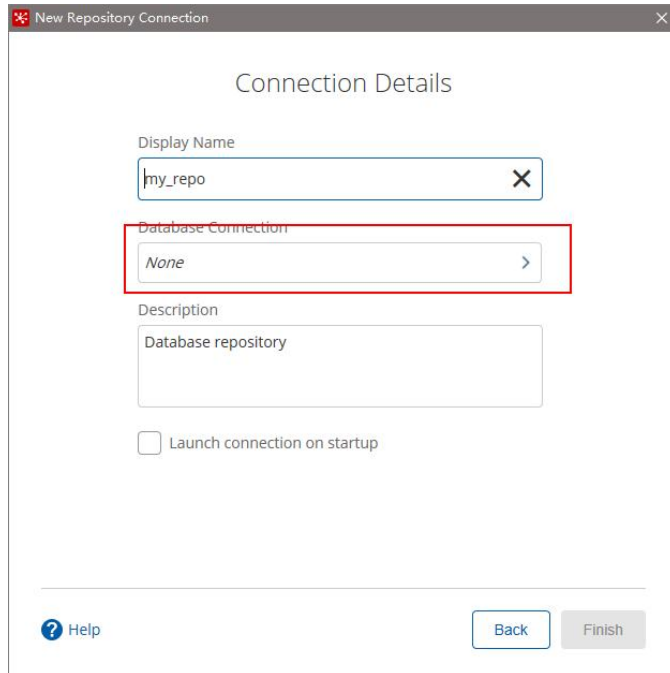
1) 点击右上角 connect，选择 Other Resporitory



2) 选择 Database Repository



3) 建立新连接



New Repository Connection

Connection Details

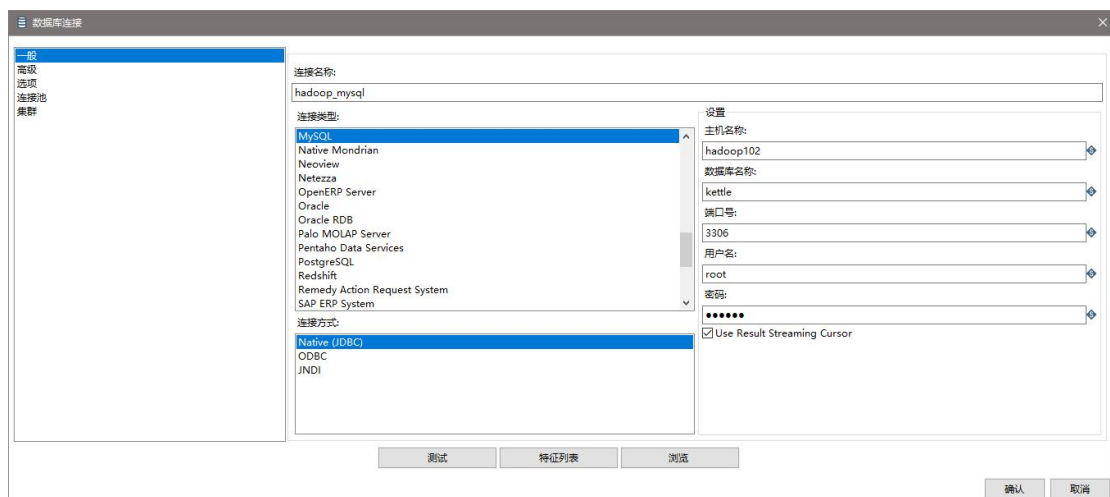
Display Name: my_repo

Database Connection: None

Description: Database repository

☐ Launch connection on startup

Buttons: ? Help, Back, Finish



数据库连接

连接名称: hadoop_mysql

连接类型: MySQL

连接方式: Native (JDBC)

设置:

- 主机名称: hadoop102
- 数据库名称: kettle
- 端口号: 3306
- 用户名: root
- 密码: *****
- ☒ Use Result Streaming Cursor

Buttons: 测试, 特征列表, 浏览, 确认, 取消

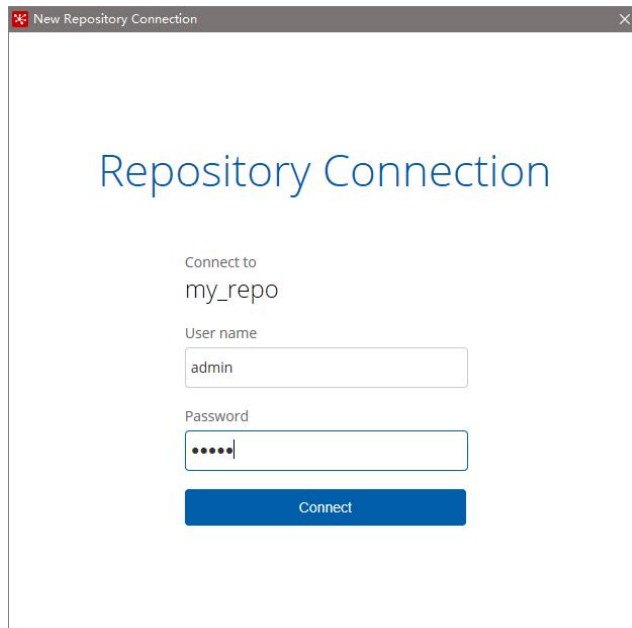
4) 填好之后，点击 finish，会在指定的库中创建很多表，至此数据库资源库创建完成

```
mysql> show tables;
+-----+
| Tables_in_kettle |
+-----+
| R_CLUSTER        |
| R_CLUSTER_SLAVE  |
| R_CONDITION       |
| R_DATABASE        |
| R_DATABASE_ATTRIBUTE |
| R_DATABASE_CONTYPE |
| R_DATABASE_TYPE   |
| R_DEPENDENCY      |
| R_DIRECTORY       |
| R_ELEMENT         |
| R_ELEMENT_ATTRIBUTE |
| R_ELEMENT_TYPE    |
| R_JOB             |
| R_JOBENTRY        |
| R_JOBENTRY_ATTRIBUTE |
| R_JOBENTRY_COPY   |
| R_JOBENTRY_DATABASE |
| R_JOBENTRY_TYPE    |
| R_JOB_ATTRIBUTE   |
| R_JOB_HOP         |
```

5) 连接资源库

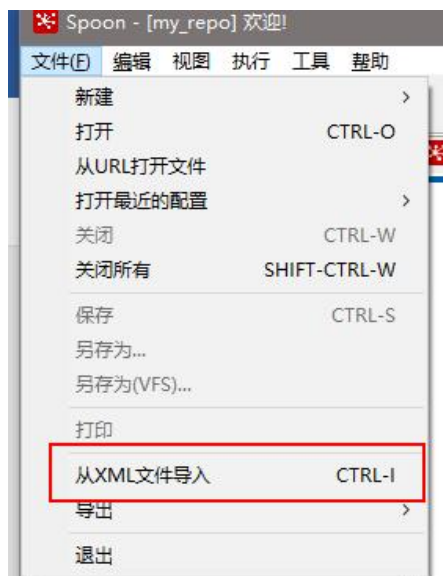
更多 [Java](#) - 大数据 - 前端 - python 人工智能资料下载，可百度访问：尚硅谷官网

默认账号密码为 admin

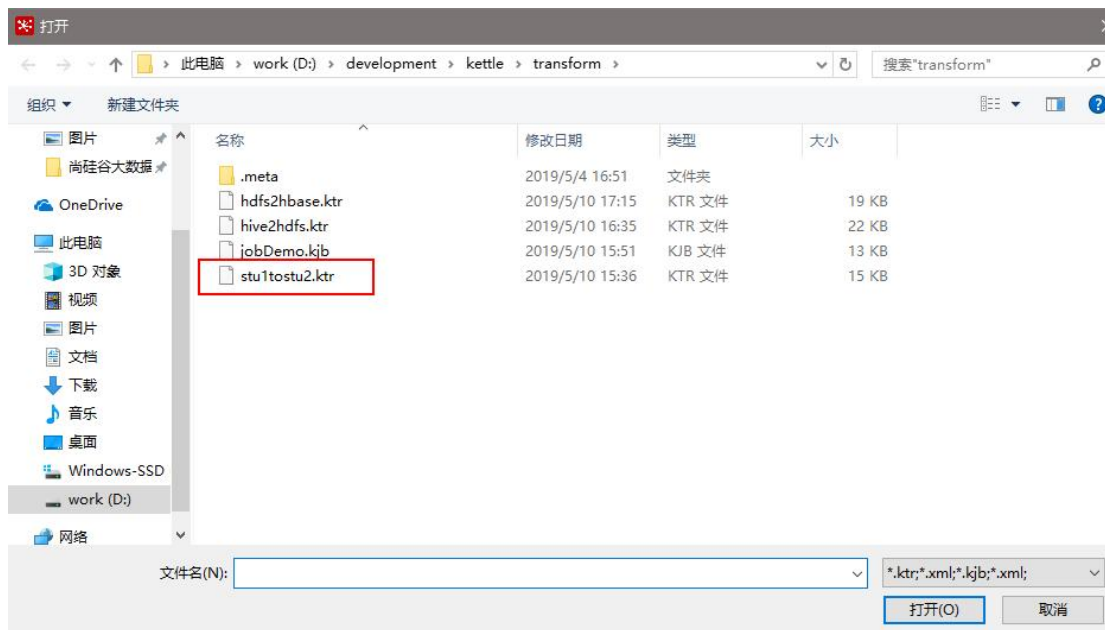


6) 将之前做过的转换导入资源库

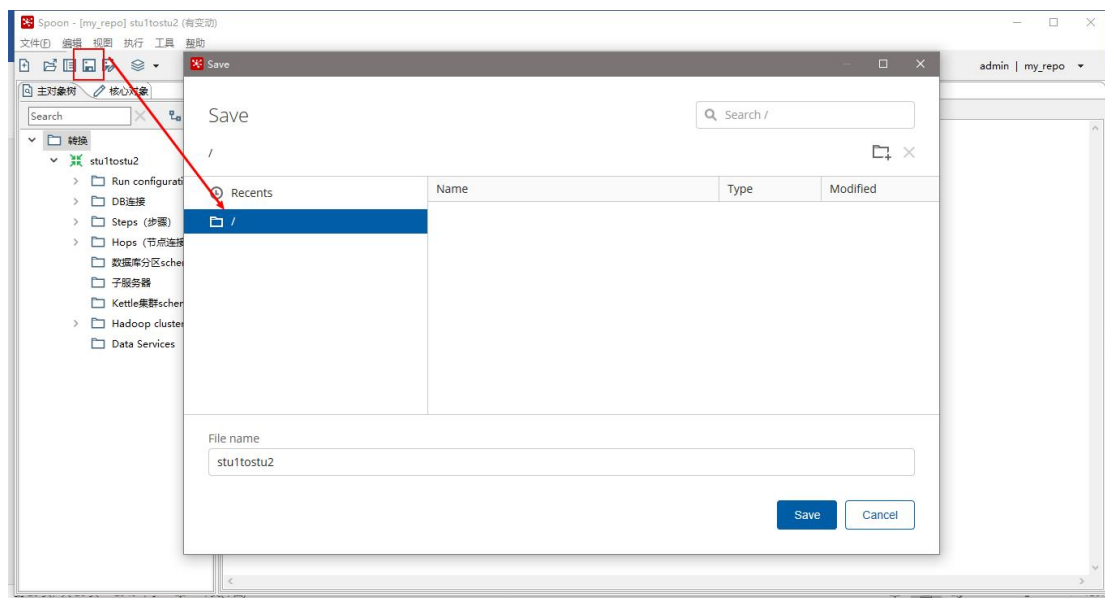
(1) 选择从 xml 文件导入



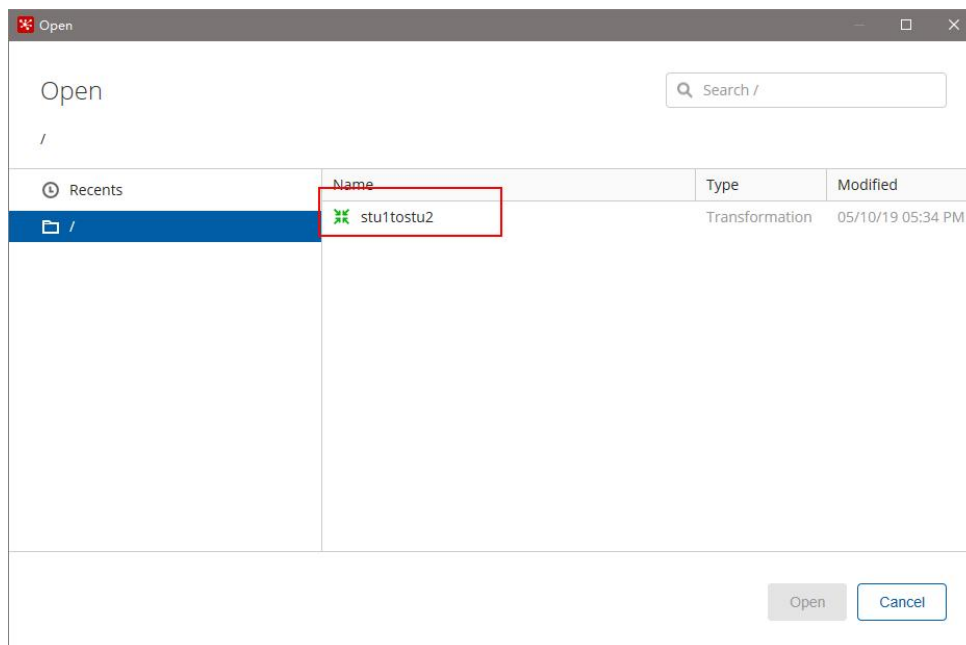
(2) 随便选择一个转换



(3) 点击保存，选择存储位置及文件名



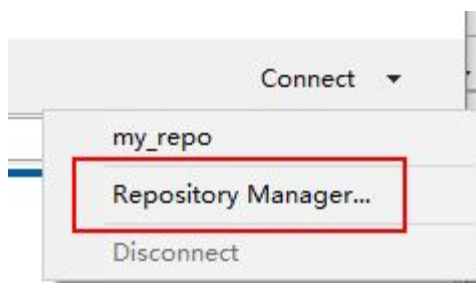
(4) 打开资源库查看保存结果



2.3.2 文件资源库

将作业和转换相关的信息存储在指定的目录中，其实和 XML 的方式一样创建方式跟创建数据库资源库步骤类似，只是不需要用户密码就可以访问，跨平台使用比较麻烦

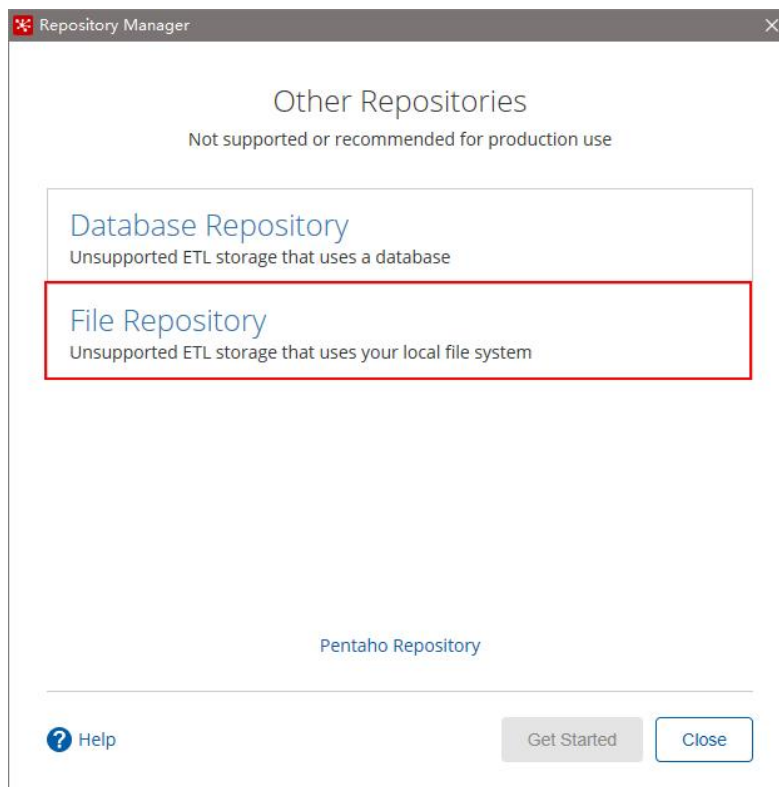
1)选择 connect



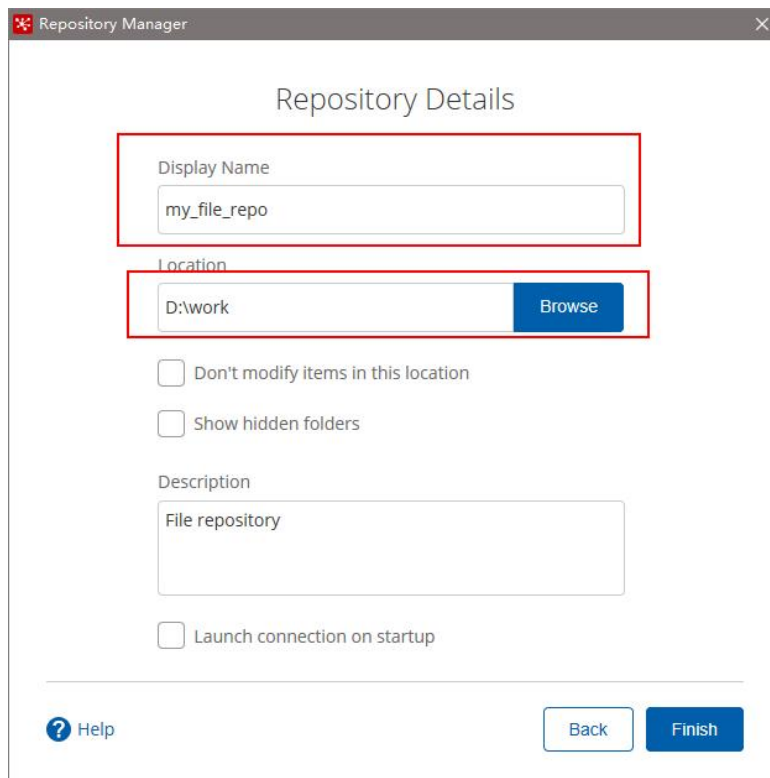
2)点击 add 后点击 Other Repositories



3)选择 File Repository



4)填写信息



2.4 Linux 下安装使用

2.4.1 单机

1)jdk 安装

2)安装包上传到服务器，解压

注意：1. 把 mysql 驱动拷贝到 lib 目录下

2. 将本地用户家目录下的隐藏目录 C:\Users\自己用户名\.kettle，整个上传到 linux 的家目录/home/atguigu/下

3)运行数据库资源库中的转换：

```
[atguigu@hadoop102 data-integration]$./pan.sh -rep=my_repo -user=admin -pass=admin  
-trans=stu1tostu2 -dir=/
```

参数说明：

-rep	资源库名称
-user	资源库用户名
-pass	资源库密码
-trans	要启动的转换名称
-dir	目录(不要忘了前缀 /)

```
icyLoader]
五月 10, 2019 5:57:45 下午 org.apache.cxf.bus.blueprint.NamespaceHandlerRegisterer register
信息: Registered blueprint namespace handler for http://cxf.apache.org/ws/rm/manager
五月 10, 2019 5:57:45 下午 org.apache.cxf.bus.blueprint.NamespaceHandlerRegisterer register
信息: Registered blueprint namespace handler for http://schemas.xmlsoap.org/ws/2005/02/rm/policy
五月 10, 2019 5:57:45 下午 org.apache.cxf.bus.osgi.CXFExtensionBundleListener addExtensions
信息: Adding the extensions from bundle org.apache.cxf.cxf-rt-javascript (242) [org.apache.cxf.javascript.JavascriptServerListener]
五月 10, 2019 5:57:45 下午 org.pentaho.caching.impl.PentahoCacheManagerFactory$RegistrationHandler$1 onSuccess
信息: New Caching Service registered
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/software/data-integration/launcher/./lib/slf4j-log4j12-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/software/data-integration/plugins/pentaho-big-data-plugin/lib/slf4j-log4j12-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2019/05/10 17:57:47 - Pan - 开始运行.
2019/05/10 17:57:47 - RepositoriesMeta - Reading repositories XML file: /home/atguigu/.kettle/repositories.xml
五月 10, 2019 5:57:48 下午 org.apache.cxf.endpoint.ServerImpl initDestination
信息: Setting the server's publish address to be /lineage
五月 10, 2019 5:57:48 下午 org.apache.cxf.endpoint.ServerImpl initDestination
信息: Setting the server's publish address to be /i18n [stultostu2]
2019/05/10 17:57:48 - stultostu2 - 为了转换解除补丁开始 [stultostu2]
2019/05/10 17:57:48 - ???0 - Finished reading query, closing connection.
2019/05/10 17:57:48 - ???0 - 完成处理 (I=5, O=0, R=0, W=5, U=0, E=0)
2019/05/10 17:57:48 - ?? / ??0 - 完成处理 (I=5, O=1, R=5, W=5, U=0, E=0)
2019/05/10 17:57:48 - Pan - 完成!
2019/05/10 17:57:48 - Pan - 开始=2019/05/10 17:57:47.399, 停止=2019/05/10 17:57:48.854
2019/05/10 17:57:48 - Pan - 1 秒后处理结束.
2019/05/10 17:57:48 - stultostu2 -
2019/05/10 17:57:48 - stultostu2 - 进程 ???0 成功结束, 处理了 5 行. ( 5 行/秒)
2019/05/10 17:57:48 - stultostu2 - 进程 ?? / ??0 成功结束, 处理了 5 行. ( 5 行/秒)
[atguigu@hadoop102 data-integration]$
```

4)运行资源库里的作业:

记得把作业里的转换变成资源库中的资源

```
[atguigu@hadoop102 data-integration]$.kitchen.sh -rep=repo1 -user=admin -pass=admin
-job=jobDemo1 -logfile=./logs/log.txt -dir=
```

参数说明:

-rep - 资源库名

-user - 资源库用户名

-pass - 资源库密码

-job - job 名

-dir - job 路径

-logfile - 日志目录

2.4.2 集群模式(了解)

1) 准备三台服务器, hadoop102 作为 Kettle 主服务器, 服务器端口号为 8080, hadoop103 和 hadoop104 作为两个子服务器, 端口号分别为 8081 和 8082。

2) 安装部署 jdk

3) hadoop 完全分布式环境搭建, 并启动进程(因为要使用 hdfs)

4) 上传解压 kettle 的安装包

5) 进到/opt/module/data-integration/pwd 目录, 修改配置文件

修改主服务器配置文件 carte-config-master-8080.xml

```
<slaveserver>
  <name>master</name>
  <hostname>hadoop102</hostname>
```

更多 Java - 大数据 - 前端 - python 人工智能资料下载, 可百度访问: 尚硅谷官网

```
<port>8080</port>
<master>Y</master>
<username>cluster</username>
<password>cluster</password>
</slaveserver>
```

修改从服务器配置文件 carte-config-8081.xml

```
<masters>
  <slaveserver>
    <name>master</name>
    <hostname>hadoop102</hostname>
    <port>8080</port>
    <username>cluster</username>
    <password>cluster</password>
    <master>Y</master>
  </slaveserver>
</masters>
<report to masters>Y</report to masters>
<slaveserver>
  <name>slave1</name>
  <hostname>hadoop103</hostname>
  <port>8081</port>
  <username>cluster</username>
  <password>cluster</password>
  <master>N</master>
</slaveserver>
```

修改从配置文件 carte-config-8082.xml

```
<masters>
  <slaveserver>
    <name>master</name>
    <hostname>hadoop102</hostname>
    <port>8080</port>
    <username>cluster</username>
    <password>cluster</password>
    <master>Y</master>
  </slaveserver>
</masters>
<report_to_masters>Y</report_to_masters>
<slaveserver>
  <name>slave2</name>
  <hostname>hadoop104</hostname>
  <port>8082</port>
  <username>cluster</username>
  <password>cluster</password>
  <master>N</master>
</slaveserver>
```

6) 分发整个 kettle 的安装目录，xsync data-integration

7) 启动相关进程，在 hadoop102,hadoop103,hadoop104 上执行

```
[atguigu@hadoop102 data-integration]$./carte.sh hadoop102 8080
[atguigu@hadoop103 data-integration]$./carte.sh hadoop103 8081
[atguigu@hadoop104 data-integration]$./carte.sh hadoop104 8082
```

8) 访问 web 页面

<http://hadoop102:8080>

9) 案例：读取 hive 中的 emp 表，根据 id 进行排序，并将结果输出到 hdfs 上

注意：因为涉及到 hive 和 hbase 的读写，需要修改相关配置文件。

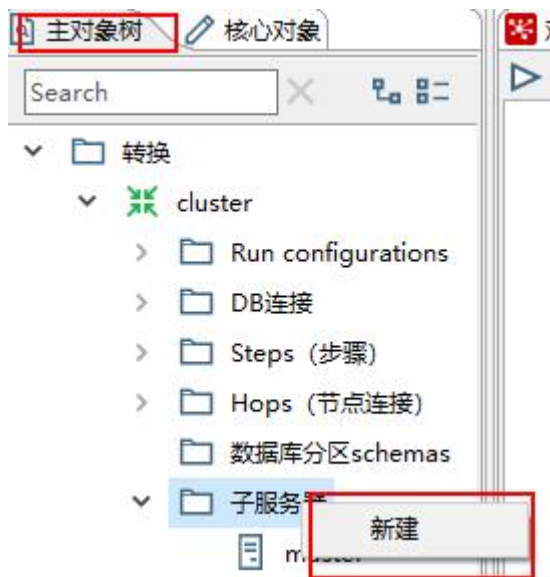
修改解压目录下的 data-integration\plugins\pentaho-big-data-plugin 下的 plugin.properties，设置 active.hadoop.configuration=hdp26，并将如下配置文件拷贝到 data-integration\plugins\pentaho-big-data-plugin\hadoop-configurations\hdp26 下



(1) 创建转换，编辑步骤，填好相关配置



(2) 创建子服务器，填写相关配置，跟集群上的配置相同

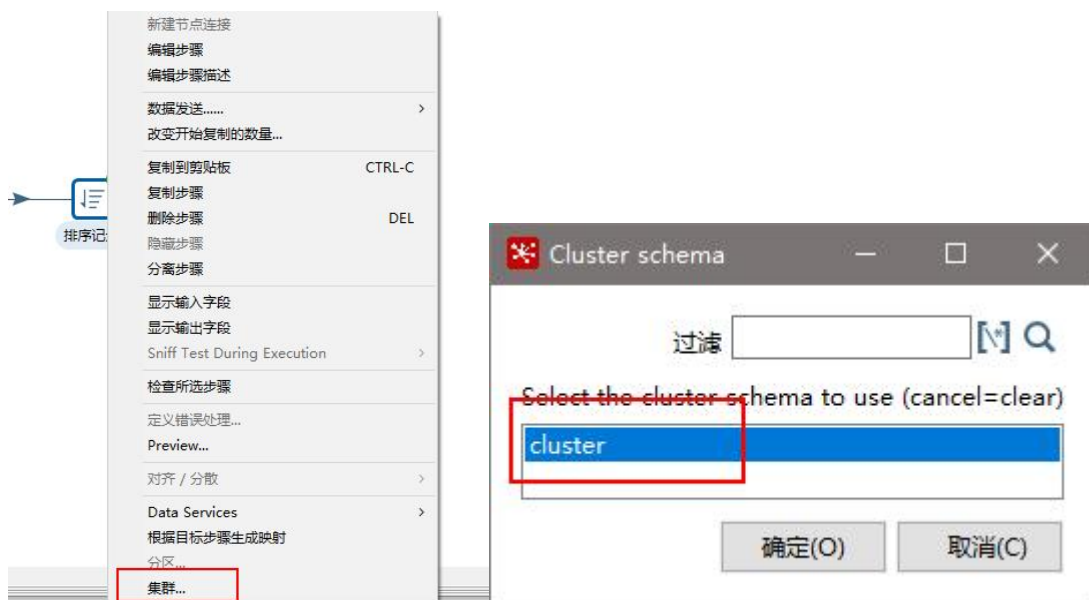




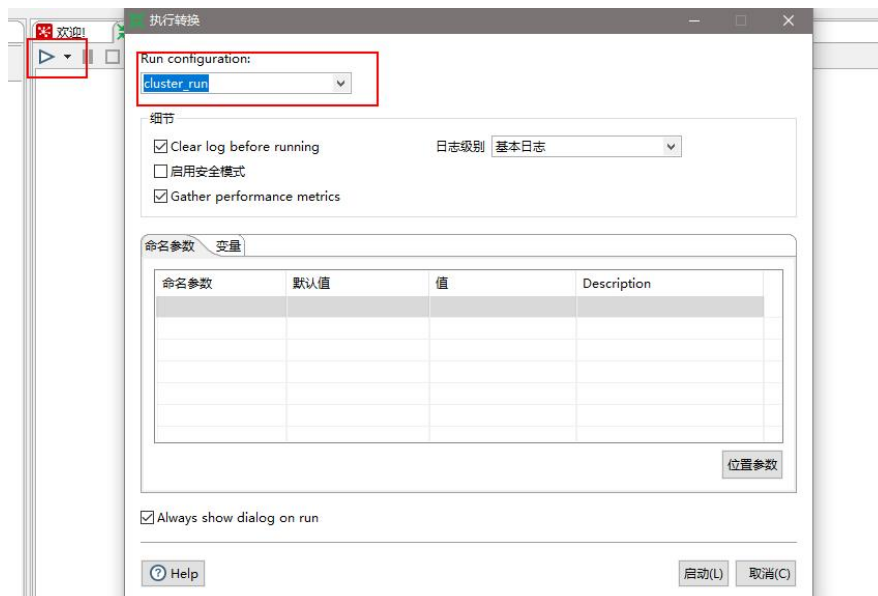
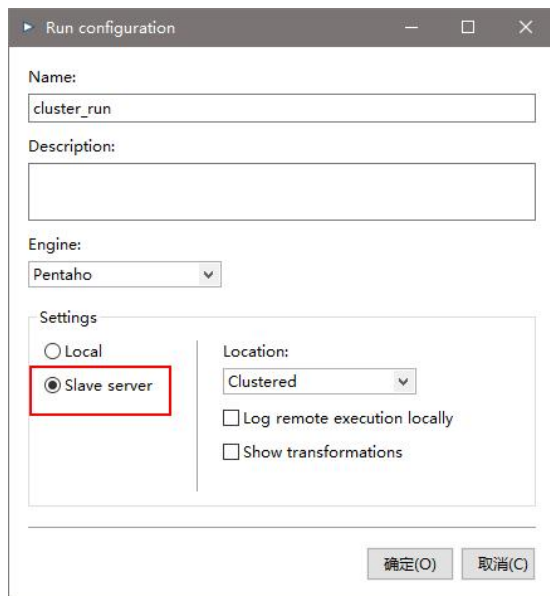
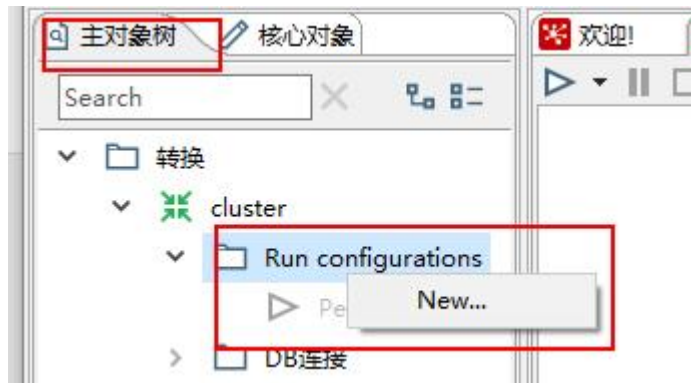
(3) 创建集群 schema，选中上一步的几个服务器



(4) 对于要在集群上执行的步骤，右键选择集群，选中上一步创建的集群 schema



(5) 创建 Run Configuration,选择集群模式，直接运行



第3章 调优

1、调整 JVM 大小进行性能优化，修改 Kettle 根目录下的 Spoon 脚本。

```
REM *****
REM ** Set java runtime options *****
REM ** Change 2048m to higher values in case you run out of memory **
REM ** or set the PENTAHO_DI_JAVA_OPTIONS environment variable *****
REM *****

if "%PENTAHO_DI_JAVA_OPTIONS%"=="%*" set PENTAHO_DI_JAVA_OPTIONS="-Xms1024m" "-Xmx2048m" "-XX:MaxPermSize=256m"

set OPT=%OPT% %PENTAHO_DI_JAVA_OPTIONS% "-Dhttps.protocols=TLSv1,TLSv1.1,TLSv1.2" "-Djava.library.path=%LIBSPATH%" "-DKETTLE_HOME"
```

参数参考：

-Xmx2048m：设置 JVM 最大可用内存为 2048M。

-Xms1024m：设置 JVM 促使内存为 1024m。此值可以设置与-Xmx 相同，以避免每次垃圾回收完成后 JVM 重新分配内存。

-Xmn2g：设置年轻代大小为 2G。整个 JVM 内存大小=年轻代大小 + 年老代大小 + 持久代大小。持久代一般固定大小为 64m，所以增大年轻代后，将会减小年老代大小。此值对系统性能影响较大，Sun 官方推荐配置为整个堆的 3/8。

-Xss128k：设置每个线程的堆栈大小。JDK5.0 以后每个线程堆栈大小为 1M，以前每个线程堆栈大小为 256K。更具应用的线程所需内存大小进行调整。在相同物理内存下，减小这个值能生成更多的线程。但是操作系统对一个进程内的线程数还是有限制的，不能无限生成，经验值在 3000~5000 左右。

2、调整提交（Commit）记录数大小进行优化，Kettle 默认 Commit 数量为：1000，可以根据数据量大小来设置 Commitsize：1000~50000

3、尽量使用数据库连接池；

4、尽量提高批处理的 commit size；

5、尽量使用缓存，缓存尽量大一些（主要是文本文件和数据流）；

6、Kettle 是 Java 做的，尽量用大一点的内存参数启动 Kettle；

7、可以使用 sql 来做的一些操作尽量用 sql；

Group , merge , stream lookup,split field 这些操作都是比较慢的，想办法避免他们，能用 sql 就用 sql；

8、插入大量数据的时候尽量把索引删掉；

9、尽量避免使用 update , delete 操作，尤其是 update,如果可以把 update 变成先 delete，后 insert；

更多 Java -大数据 -前端 -python 人工智能资料下载，可百度访问：[尚硅谷官网](http://www.shangguigu.com)

- 10、能使用 `truncate table` 的时候，就不要使用 `deleteall row` 这种类似 sql 合理的分区，如果删除操作是基于某一个分区的，就不要使用 `delete row` 这种方式（不管是 `deletesql` 还是 `delete` 步骤），直接把分区 `drop` 掉，再重新创建；
- 11、尽量缩小输入的数据集的大小（增量更新也是为了这个目的）；
- 12、尽量使用数据库原生的方式装载文本文件(Oracle 的 `sqlloader`, mysql 的 `bulk loader` 步骤)。