使用自定义增量插件实现增量抽取

# 1 原理

通过数据库表来配置每个job的增量信息，转换执行前从配置表获取初始参数，设置为变量（checkColumn、start、end等），表输入控件中使用变量来控制查询条件，如果增量字段为时间戳，则条件为 checkColumn>=start and checkColumn<end（start处为>=条件，end处为<条件），如果增量字段为数值，则条件为checkColumn>start（start处为>条件）。

执行成功后，根据相关参数，修改增量配置表中的start和end参数值，以便下次执行使用。

# 2 配置文件修改

修改kettle配置文件 userHome/.kettle/kettle.properties文件，添加如下内容：

**driver=com.mysql.jdbc.Driver**

**url=jdbc:mysql://localhost:3306/orchestra**

**username=root**

**password=root**

**useLastValue=false**

**table=job\_schedule**

**updateField=incField**

其中，useLastValue=false 表示不使用记录集的最后一条记录作为更新end值，如果是数值类型的增量，则需要设置为true以便从记录集获取下一次的开始值，如果增量类型是时间戳则设置为false（该值在kettle.properties为全局变量，对所有kettle作业和转换都默认生效，如果对特定转换或者作业重新赋值，则需要在转换或者作业的命名参数或者变量里面重新设置即可）；table、updateField为增量配置表信息，不可修改。

# 3 数据库修改

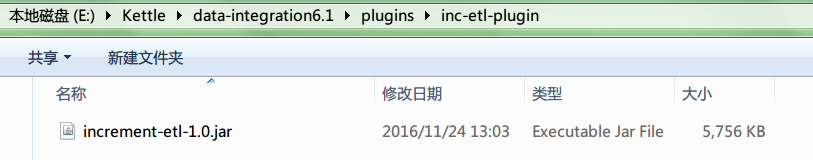
首先创建kettle.properties中指定的数据库，比如orchestra，设置编码为utf8，然后创建job\_schedule表，sql语句为：

CREATE TABLE `job\_schedule` (  
  `jobId` varchar(64) NOT NULL,  
  `jobName` varchar(300) DEFAULT NULL,  
  `incType` varchar(64) DEFAULT NULL COMMENT ‘增量方式：时间戳、数值、其他’,  
  `incField` varchar(64) DEFAULT NULL COMMENT ‘增量字段’,  
  `dataFormat` varchar(64) DEFAULT NULL COMMENT ‘增量字段的数据格式，主要针对时间’,  
  `lastStartValue` varchar(128) DEFAULT NULL COMMENT ‘上一次调度的开始值’,  
  `lastEndValue` varchar(128) DEFAULT NULL COMMENT ‘上一次调度的结束值’,  
  `lastRunTime` timestamp NULL DEFAULT NULL COMMENT ‘上一次执行时间戳’,  
  PRIMARY KEY (`jobId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8

# 4 kettle插件使用

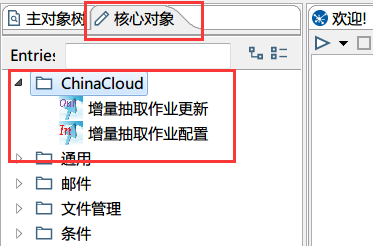
## 4.1 导入增量插件到kettle中

Kettle插件需要放到kettle程序的plugins目录下，比如 “E:\Kettle\data-integration6.1\plugins\inc-etl-plugin”：



## 4.1 创建job

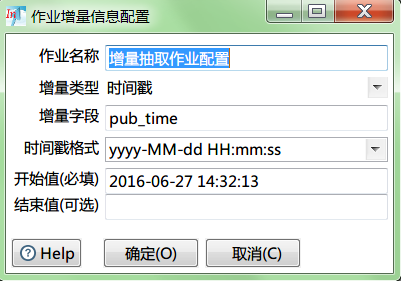
首先做一个job，在kettle的核心对象里面选择增量插件，在实际抽取流程（trans或者job）之前使用“**增量抽取作业配置**”插件，在抽取流程之后使用“**增量抽取作业更新**”插件，如下：



最后job的大致流程如下所示：



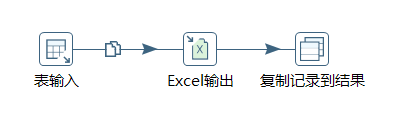
“**增量抽取作业更新**”插件无需配置，对于“**增量抽取作业配置**”插件的配置，见下图：



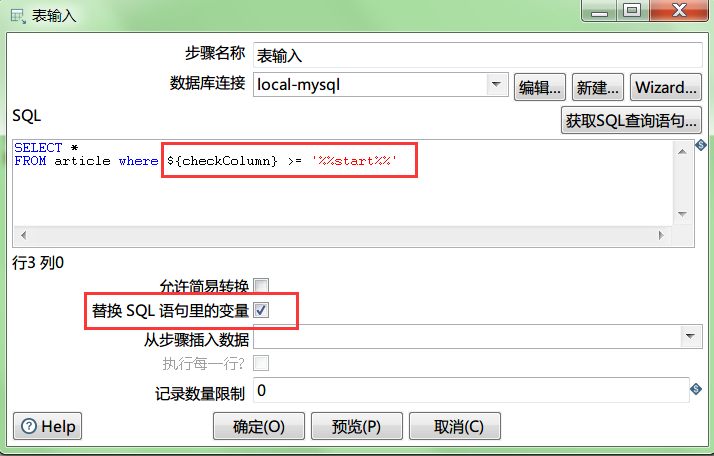
增量类型选择时间戳或者数值，增量字段则是表中用于增量查询的字段，对于时间戳则用时间字段，对于数值则用数字字段；如果是时间戳类型，则还需要设置时间戳格式；第一次抽取时，需要设置起始值，结束值可选。

## 4.2 创建抽取流程

抽取流程以转换为例，如下图所示：



在表输入中，使用checkColumn、start、end三个变量（kettle变量的使用可以使用${varName}或者%%varName%%两种形式，其中checkColumn变量可以不使用而直接使用数据库增量字段名）来控制增量数据来源，然后勾选下方的“替换SQL语句里的变量”，如下图所示：(要加ORDER BY )



然后是做与业务相关的转换清洗工作，最后将数据输出到指定地方。

需要注意的是，如果使用自定义的输出到ES的插件EDB和Hive的插件OrcParquet，则不需要“**复制记录到结果**”这个步骤，同时也不再需要前述job中的“**增量抽取作业更新**”插件，因为EDB和OrcParquet已经做了更新操作，否则必须加上这一步才能使job中的“**增量抽取作业更新**”插件生效。

# 5 运行Kettle作业

必须运行Kettle的作业才能实现增量功能，工具，命令行，调度系统均可。