

Lesson 12:

Thiết kế kiến trúc Trích xuất – Chuyển đổi – Nạp dữ liệu hiện đại

- Xây dựng luồng ETL với Pentaho
- Phân biệt Relation, Job, Transform
- Thực hành xây dựng luồng ETL xử lý file

Giảng Viên: Trần Đăng Hòa

Trợ Giảng: Nguyễn Chí Thanh





Trần Đăng Hòa

Big Data Architecture & Data Governance

Chuyên gia. **Trần Đăng Hòa**

- Xây dựng, thiết kế Data warehouse
- Thiết kế kiến trúc Data Lake
- Phân tích và khoa học dữ liệu (DA - DS)
- Kiến trúc hạ tầng và mạng lưới hệ thống Big Data
- Quản trị dữ liệu
- Admin 'Cộng đồng Big Data Việt Nam'

Mục lục

1. ETL Overview

2. ETL tools

3. Transform, Job

4. Thực hành



1. ETL Overview



Back room

- Back room architecture, bao gồm ETL process, source system và ETL data store
- Chỉ ra cách chuyển dữ liệu từ source system thông qua những ETL process cung cấp bởi ETL service layer. Flow này được định hướng bởi metadata, nó sẽ mô tả vị trí, định nghĩa source và target, biến đổi dữ liệu, timing và các dependence.
- Back room không cung cấp query service cho business user, chỉ presentation server mới hỗ trợ query service bằng cách lưu trữ và thể hiện data trong các dimensional format

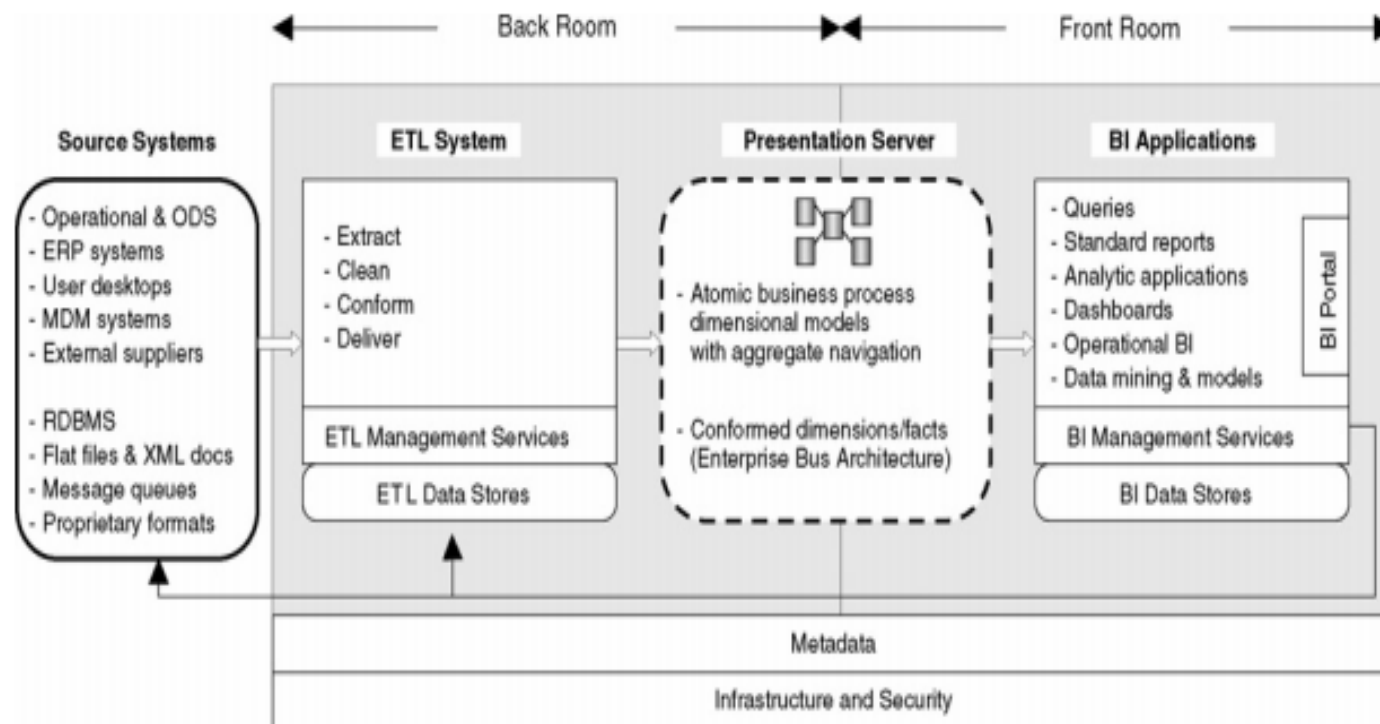


Figure 4-1: High level DW/BI system architecture model

ETL System

- Extract
- Clean and Conform
- Deliver
- Service quản lý ETL

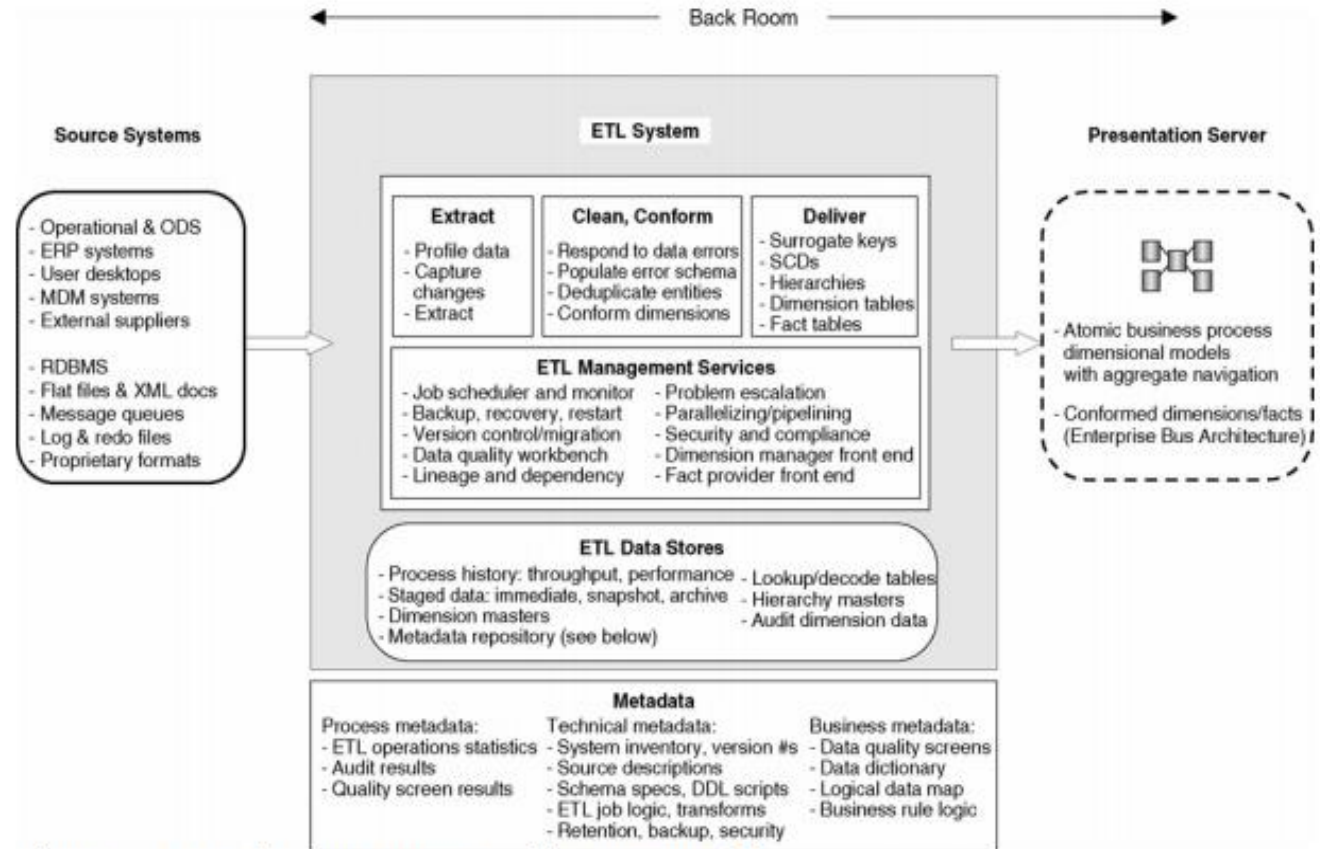


Figure 4-2: Back room system architecture model

Các yếu tố ảnh hưởng

1. Data size
2. Volatility (Sự biến động)
3. Số lượng business process
4. Bản chất sử dụng
5. Thỏa thuận Chất lượng Dịch vụ (SLA)
6. Technical readiness (Tính sẵn sàng về mặt kỹ thuật)
7. Tính khả dụng của phần mềm
8. Nguồn tài chính

2. ETL tools



Apache Oozie

- Apache Oozie is a Java Web application used to schedule Apache Hadoop jobs.
- Oozie combines multiple jobs sequentially into one logical unit of work.
- It is integrated with the Hadoop stack, with YARN as its architectural center, and supports Hadoop jobs for Apache MapReduce, Apache Pig, Apache Hive, and Apache Sqoop.
- Oozie can also schedule jobs specific to a system, like Java programs or shell scripts

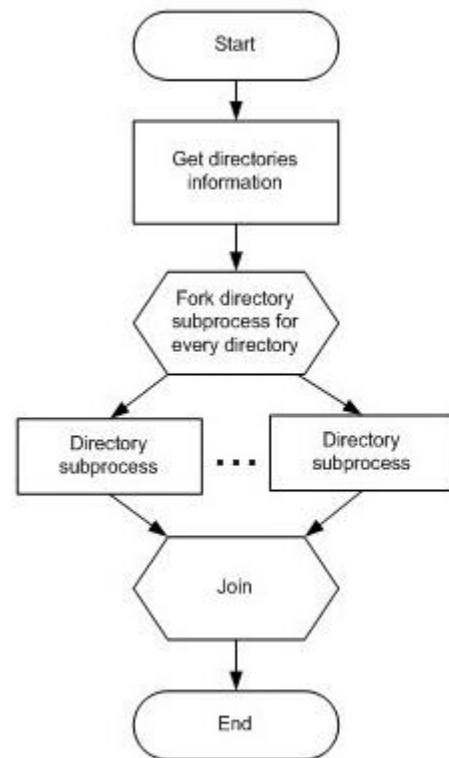


Oozie jobs

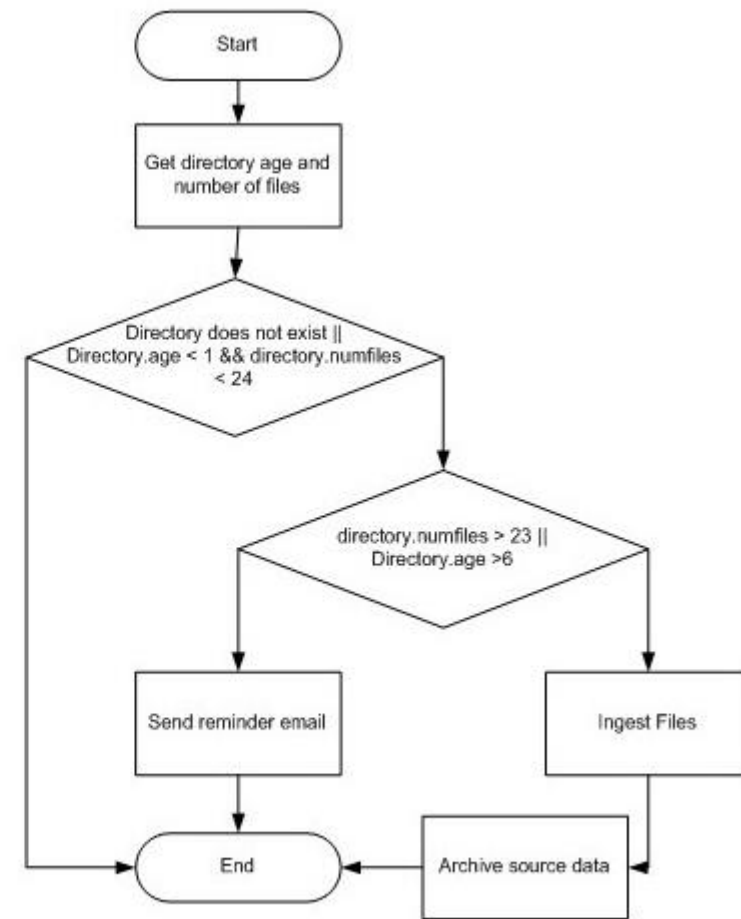
There are two basic types of Oozie jobs:

- Oozie Workflow jobs are Directed Acyclical Graphs (DAGs), specifying a sequence of actions to execute. The Workflow job has to wait.
- Oozie Coordinator jobs are recurrent Oozie Workflow jobs that are triggered by time and data availability.

Oozie Bundle provides a way to package multiple coordinator and workflow jobs and to manage the lifecycle of those jobs



Ingestion process



Directory subprocess

Oozie workflow for Hive

<https://www.youtube.com/watch?v=i1QW7NoAiwM>

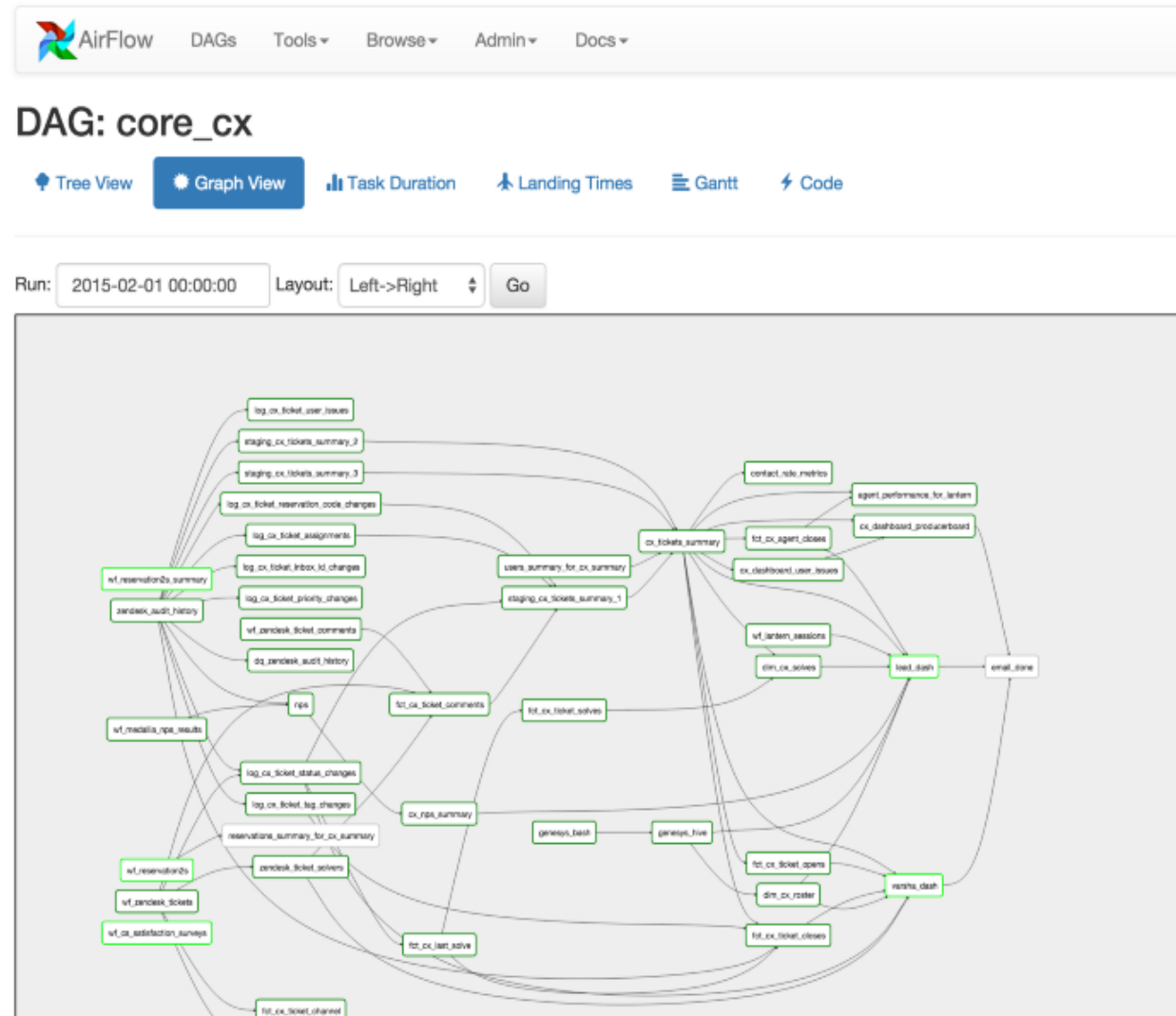
Apache Airflow

- Apache Airflow là một nền tảng lập lịch và monitor workflows.
- Sử dụng airflow tạo ra các workflows theo đồ thị chu kỳ có hướng của các task (DAGs-Directed acyclic graph).
- Airflow scheduler thực hiện các task của bạn trên một mảng các workers tuân theo các điều kiện chỉ định
- Airflow tạo một giao diện web cho phép người dùng dễ dàng hiển thị các pipeline đang chạy, theo dõi tiến trình và hỗ trợ khắc phục sự cố



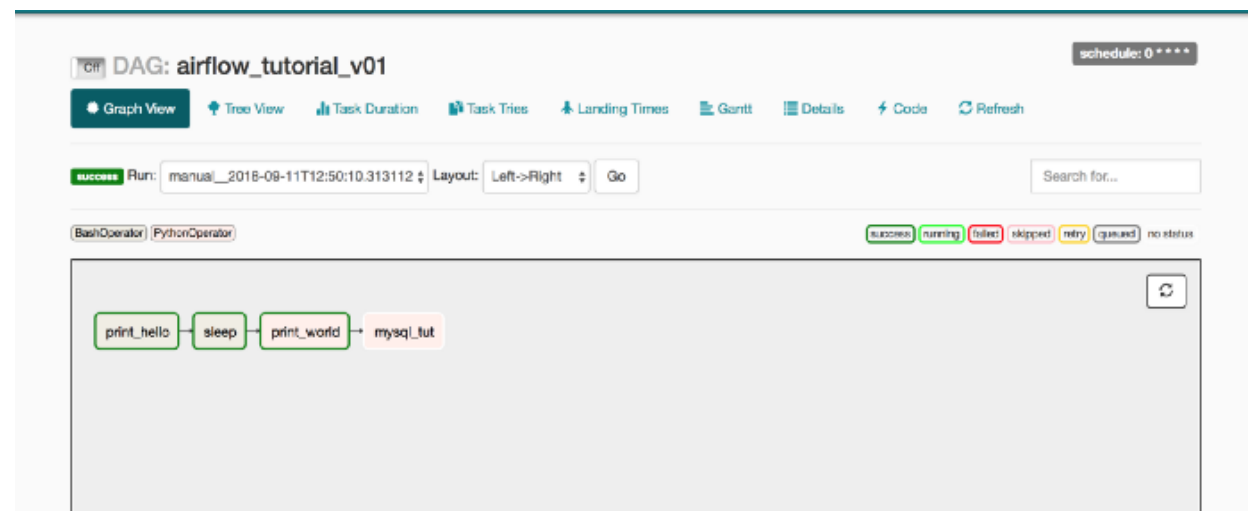
Airflow (DAGs)

- Shape của graph quyết định logic tổng thể của workflow. Một luồng workflow có thể bao gồm nhiều nhánh, bạn có thể quyết định nên theo dõi nhánh nào, và bỏ qua nhánh nào tại mỗi thời điểm thực hiện của workflow execution.
- Mỗi đồ thị chu kỳ có hướng trong Airflow (DAGs) bao gồm các thành phần chính sau:
 - BashOperator: thực hiện các bash command.
 - PythonOperator: gọi, sử dụng các hàm Python.
 - EmailOperator: gửi email.
 - SimpleHTTPOperator: thực hiện các HTTP Request.



Airflow (DAGs)

Viền màu của hình chữ nhật ứng với trạng thái của task như success, running,



Đánh giá Apache Airflow

Ưu điểm: Giao diện web dễ sử dụng, thân thiện với người dùng. Có thể xử lý logic phức tạp nhờ sự linh hoạt của Python, có tính năng phân quyền, lập lịch, trigger. Quản lý tất cả các DAG trên một giao diện web.

Nhược điểm: Mất nhiều thời gian để code. Không thể chỉnh sửa code ngay trên giao diện web mà cần phải vào Linux tìm file DAG để chỉnh sửa. Chỉ sử dụng duy nhất Python để tạo DAG.

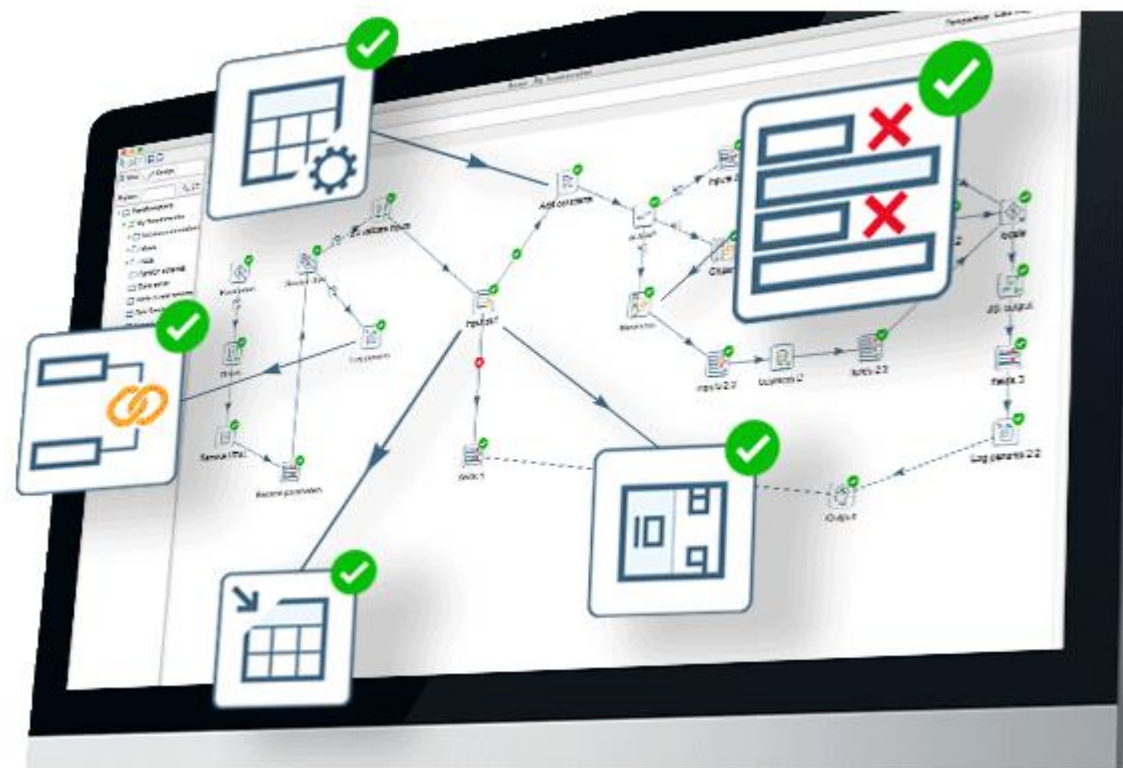
Ko kéo thả được, ko code được, đặt lịch chạy các file Python
Giao diện chỉ để Monitor và checklog

Apache Airflow demo

https://www.youtube.com/watch?v=iTg-a4icf_I

Pentaho Data Integration (Kettle)

- Là công cụ thiết kế các luồng tổng hợp dữ liệu (ETL) với các chức năng rất đa dạng như: thực thi câu lệnh SQL, ứng dụng Java, Spark, đẩy dữ liệu lên FTP, email báo cáo, ...
- Pentaho phiên bản mới nhất 9.0 hiện thuộc về Hitachi Vantara, hiện đang có bản community và tính phí, có thêm các tính năng mới như Pentaho BI Server hiển thị web UI các biểu đồ phân tích, phân quyền người dùng



Tính năng

Một số các tính năng của Pentaho như:

- **Data Integration:** Pentaho Data Integration (PDI) là công cụ GUI cho phép người dùng tương tác với các tiện ích thực hiện ETL mà không cần viết code
- **Big Data:** Pentaho giúp giảm thiểu thời gian và sự phức tạp khi thao tác với dữ liệu lớn, Pentaho hỗ trợ từ quá trình trích xuất dữ liệu, chuẩn bị dữ liệu cho đến việc xử lý phân tán sử dụng Hadoop, Spark
- **Multicloud Support:** Pentaho hỗ trợ việc tương tác với các nền tảng cloud, hybrid
- **Business Analytics:** Pentaho Business Analytics cung cấp giao diện web trực quan cho người dùng trực quan dữ liệu qua các phân tích, biểu đồ, báo cáo
- **Embedded Analytics:** Pentaho có khả năng nhúng vào các ứng dụng khác tùy thuộc vào yêu cầu của người dùng

Thành phần

Web-based Components

Pentaho sử dụng các thành phần web-based chia sẻ giải pháp business intelligent phân tích dữ liệu, thiết kế báo cáo, biểu đồ, các thành phần này bao gồm:

- **User Console:** Pentaho User Console là môi trường thiết kế để truy cập Analyzer, Interactive Reports và Dashboard Designer. Pentaho User Console cung cấp các tính năng quản trị Pentaho Server.
- **Analyzer:** cho phép hiển thị dữ liệu biểu đồ để đưa ra các quyết định kinh doanh.
- **Interactive Reports:** hỗ trợ việc sinh ra báo cáo bằng các tiện ích có sẵn.
- **Dashboard Designer:** dùng để chọn giao diện mẫu, nội dung cho việc thiết kế dashboard nhằm làm nổi bật dữ liệu cần tập trung. Dashboard Designer có thể gồm Interactive Reports, Analyzer và các nội dung liên quan.
- **CTools:** là một framework của cộng đồng sử dụng Javascripts, HTML, CSS để tạo dashboard động thông qua biểu đồ, bảng và nhiều thành phần khác.
- **Data Source Wizard:** Định nghĩa nguồn dữ liệu sinh ra các báo cáo, phân tích.
- **Data Source Model Editor:** Giúp tinh chỉnh và định nghĩa mô hình dữ liệu

Thành phần

Design Tools

Pentaho Design Tools hỗ trợ việc phát triển và lọc, định nghĩa cách dữ liệu được báo cáo, mô hình, biến đổi và lưu trữ. Các thành phần như:

- **Data Integration:** Pentaho Data Integration (PDI) hỗ trợ Extraction, Transformation và Loading (ETL) dữ liệu nhằm tạo ra dữ liệu có ích.
- **Report Designer:** Sinh ra báo cáo chi tiết sử dụng nhiều nguồn dữ liệu khác nhau.
- **Aggregation Designer:** cung cấp giao diện đơn giản cho phép tạo bảng tổng hợp từ các chiều xác định.
- **Metadata Editor:** Đơn giản hóa việc xây dựng báo cáo bằng cách sử dụng Pentaho metadata domains và model.
- **Schema Workbench:** cho phép tạo và chỉnh sửa multidimensional models.

Thành phần

Pentaho Repository

- Pentaho Repository lưu trữ và quản lý các jobs và transformations, cung cấp lịch sử sửa đổi để tracking, so sánh phiên bản và quay lại các phiên bản trước đó nếu cần thiết. PDI Client kết nối đến Pentaho Repository thông qua Pentaho Server.
- Ngoài ra Pentaho hỗ trợ sử dụng database repository (sử dụng cơ sở dữ liệu quan hệ để lưu trữ ETL metadata) và file repository (sử dụng hệ thống lưu trữ file cục bộ để lưu metadata).

Pentaho Server

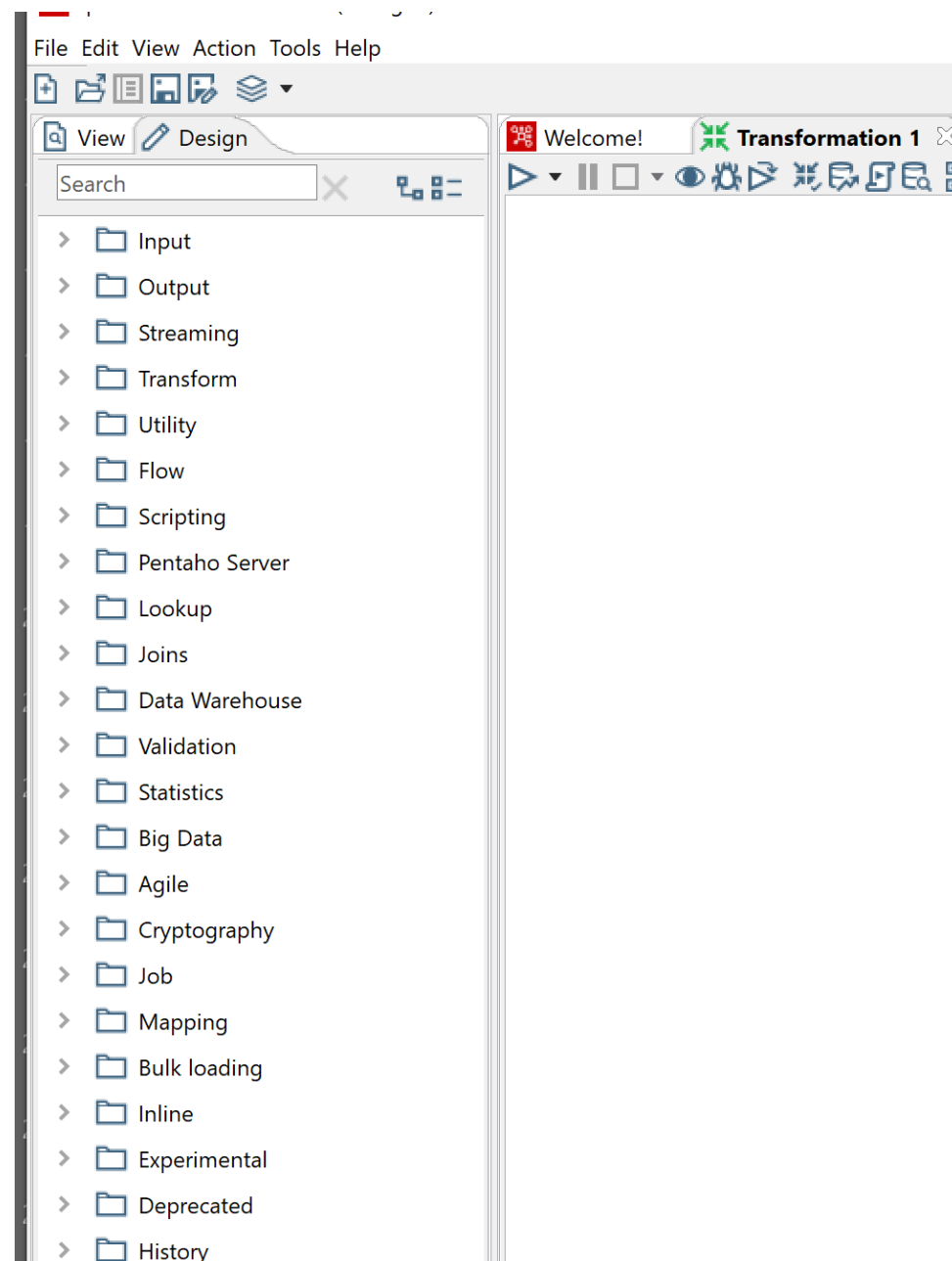
- Pentaho Server sử dụng postgresql docker để lưu trữ repository trên server

3. Transform, Job



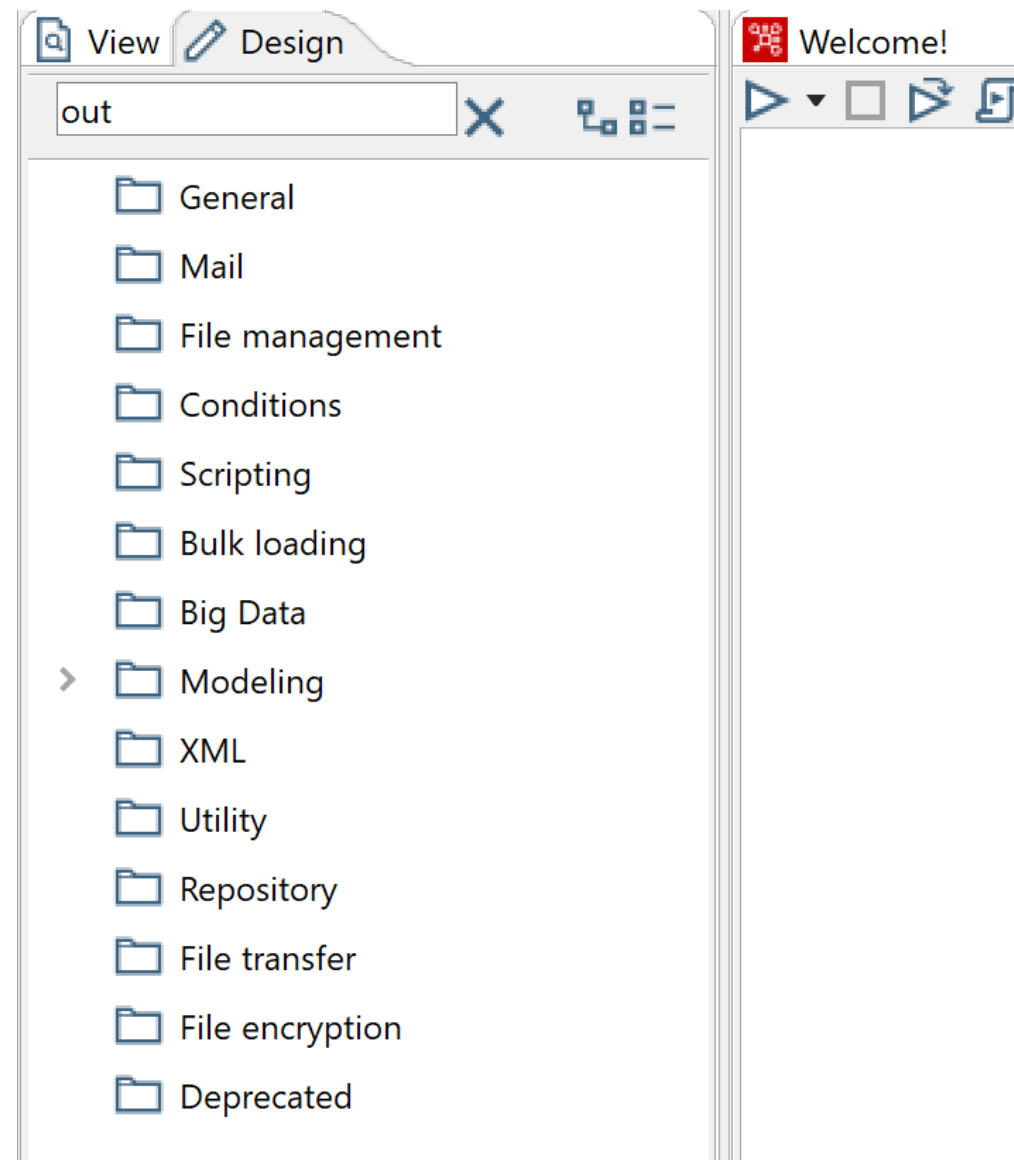
Transform

1. Input: Các step đọc dữ liệu
2. Output: Các step ghi dữ liệu
3. Streaming: Các step streaming
4. Transform: Các step biến đổi dữ liệu
5. ...



Job

1. General: Các Entry chung
2. Mail: Gửi nhận mail
3. File management: Gửi nhận file
4. Conditions: Check điều kiện
5. ...



4. Thực hành



Practice

Xây dựng Job tổng hợp cho phép:

Bài 1:

- Xử lý danh sách sinh viên, danh sách điểm, danh sách môn
- Xuất ra danh sách sinh viên top 10 điểm trung bình cao nhất
- Xuất ra danh sách môn kèm sắp xếp sinh viên điểm cao nhất

Bài 2:

- Import danh sách sinh viên, điểm, môn vào Database My SQL
- Tạo ra bảng sinh viên sắp xếp theo điểm trung bình cao nhất
- Tạo ra bảng danh sách môn kèm sắp xếp sinh viên điểm cao nhất

Practice












Setup

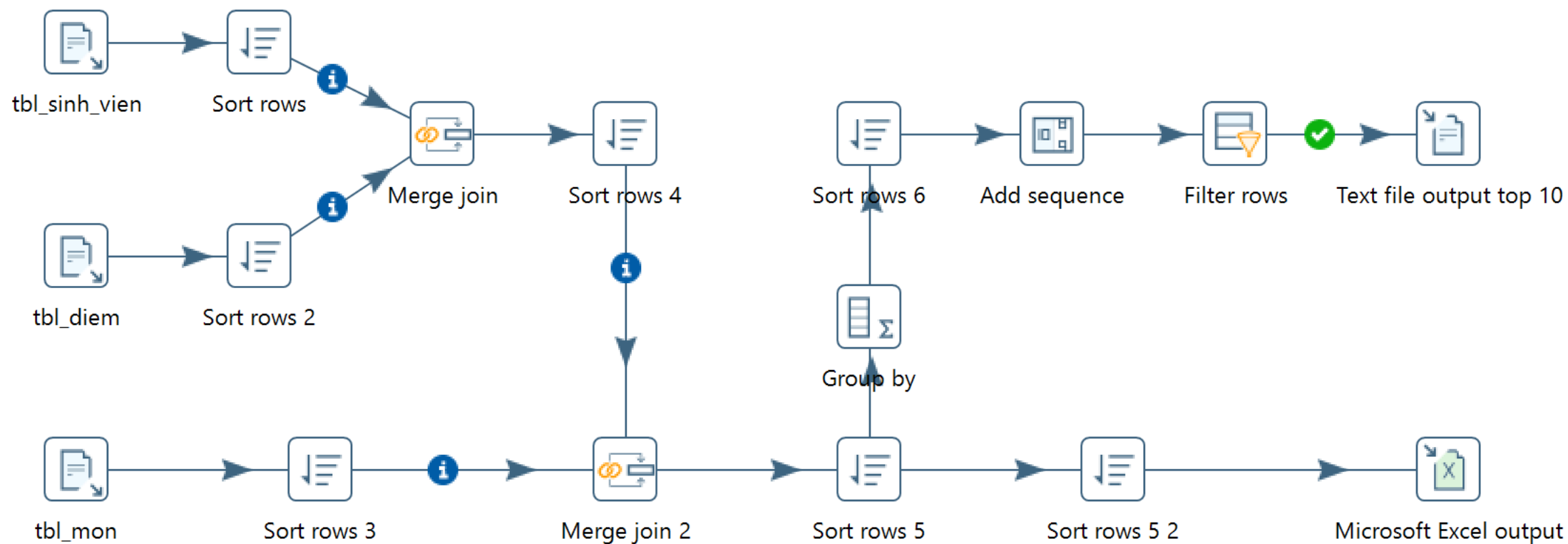
Download Pentaho:

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

Giải nén và chạy file Spoon.batch

	set-pentaho-env	9/7/2020 4:52 PM	Windows Batch File	6 KB
	set-pentaho-env	9/7/2020 4:52 PM	Shell Script	5 KB
	Spark-app-builder	9/7/2020 4:52 PM	Windows Batch File	2 KB
	spark-app-builder	9/7/2020 4:52 PM	Shell Script	2 KB
<input checked="" type="checkbox"/> 	Spoon	9/7/2020 4:52 PM	Windows Batch File	6 KB
	spoon.command	9/7/2020 4:52 PM	COMMAND File	2 KB
	spoon	9/7/2020 4:52 PM	Icon	204 KB
	spoon	9/7/2020 4:52 PM	PNG File	1 KB
	spoon	9/7/2020 4:52 PM	Shell Script	8 KB

Hướng dẫn



Setup DB

Khởi động máy ảo và cấu hình mạng:

🔴 Master - Settings

Network

Adapter 1 Adapter 2 Adapter 3 Adapter 4

☒ Enable Network Adapter

Attached to: Bridged Adapter ▼

Name: Marvell AVASTAR Wireless-AC Network Controller

▶ Advanced

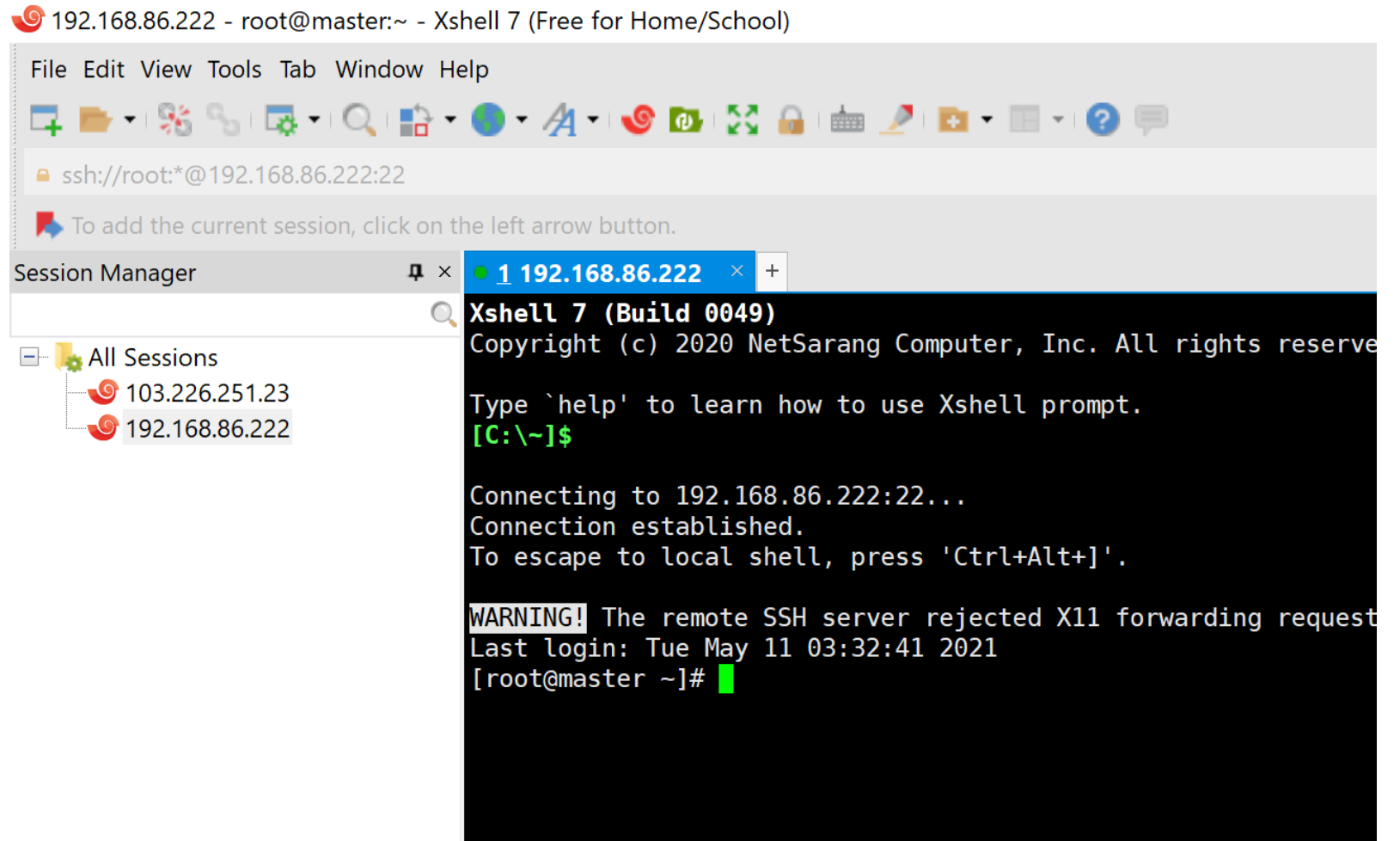
Setup

Lấy IP của máy ảo

```
master login: root
Password:
Last login: Mon May 10 09:14:25 from 192.168.86.171
[root@master ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defa
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
    group default qlen 1000
    link/ether 08:00:27:29:04:5b brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
    group default qlen 1000
    link/ether 08:00:27:b0:f2:0d brd ff:ff:ff:ff:ff:ff
    inet 192.168.86.222/24 brd 192.168.86.255 scope global noprefixroute dynam
    enp0s8
        valid_lft 3581sec preferred_lft 3581sec
    inet6 fe80::387b:9dff:cbe6:bc00/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@master ~]#
```

Setup

Login qua Xshell



Setup

Tắt firewall :

`systemctl stop firewalld`

Bổ sung tham số:

`nano /etc/my.cnf`

`skip-grant-tables`

Save: Ctrl +X

```
WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Tue May 11 03:32:41 2021
[root@master ~]# systemctl stop firewalld
[root@master ~]#
```

```
GNU nano 2.3.1 File: /etc/my.cnf

[mysqld]
skip-grant-tables
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d

#bind-address = 0.0.0.0
```

Setup

Start DB:

systemctl start mariadb

Vào Mysql

mysql

Tạo DB

Create database test;

```
[root@master ~]# systemctl start mariadb  
[root@master ~]#
```

Setup

Start DB:

systemctl start mariadb

```
[root@master ~]# systemctl start mariadb  
[root@master ~]#
```

Setup

Tải DBeaver

Cấu hình kết nối

Connection "default" configuration

Connection settings

MySQL connection settings

▼ Connection settings

- Initialization
- Shell Commands
- Client identification
- Transactions
- General
- Metadata
- Errors and timeouts
- Data editor
- SQL Editor

Main

Driver properties

SSH

Proxy

SSL

Server

Server Host: 192.168.86.222

Port: 3306

Database: test

Authentication (Database Native)

Username: root


Password: ●●●●●●

☒ Save password locally

Advanced

Server Time Zone: Auto-detect

Local Client: MySQL Binaries

 You can use variables in connection parameters.

Driver name: MySQL

[Edit Driver Settings](#)

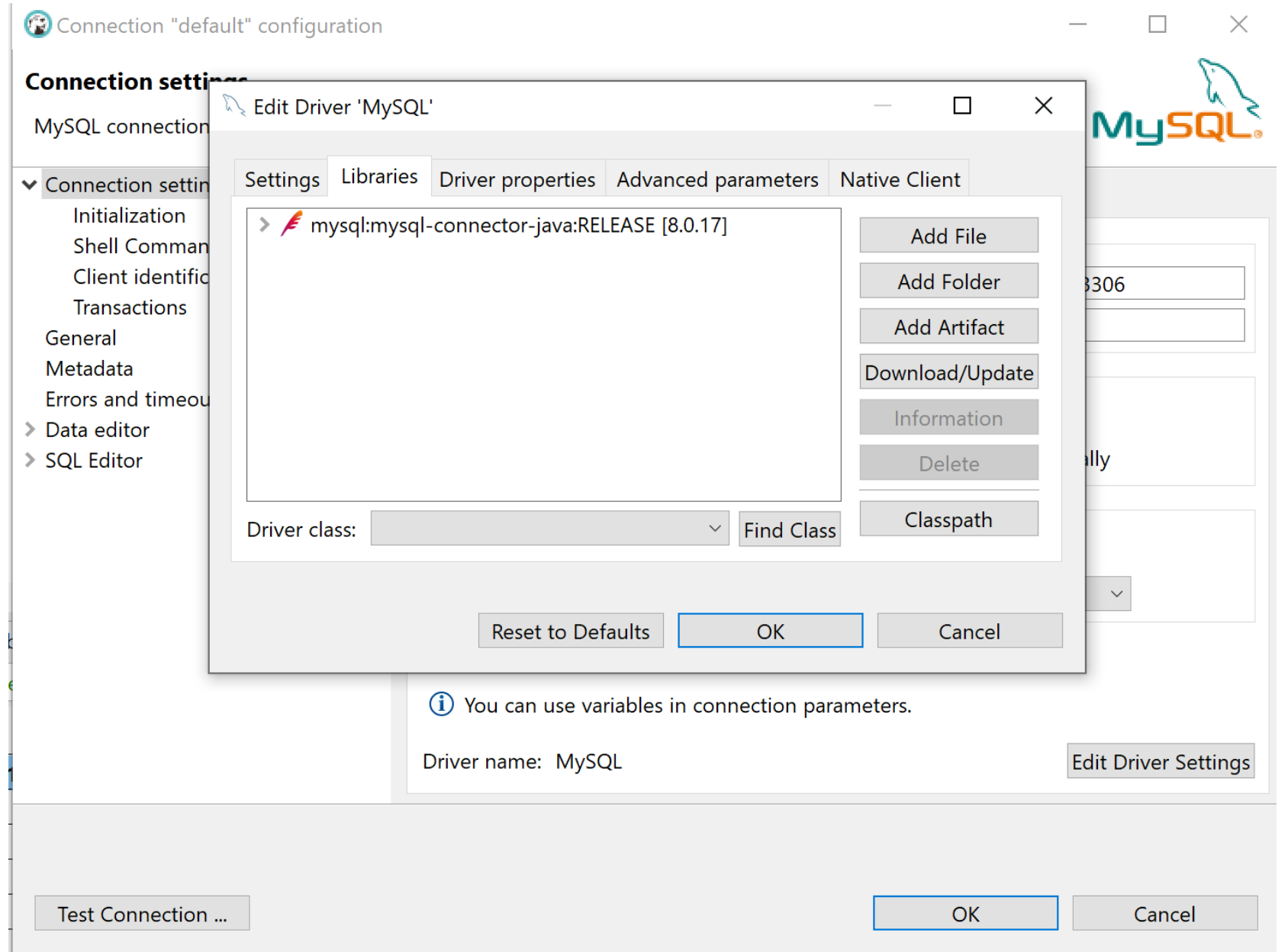
Test Connection ...

OK

Cancel

Setup

Add lib



Tạo job, tạo connection:

```
jdbc:mysql://192.168.86.222:3306/test?useUnicode=true&
useJDBCCompliantTimezoneShift=true&useLegacyDatetime
Code=false&serverTimezone=UTC
com.mysql.cj.jdbc.Driver
root/ai@acad
```

The screenshot shows a 'Connection name:' field with the value 'db_test'. Below it is a list of 'Connection type:' options, including 'Generic database', 'Google BigQuery', 'Greenplum', 'Gupta SQL Base', 'H2', 'Hadoop Hive', 'Hadoop Hive 2/3', 'Hive Warehouse Connector', 'Hypersonic', 'IBM DB2', 'Impala', 'Infobright', 'Informix', 'Ingres', 'Ingres VectorWise', 'Intersystems Cache', 'KingbaseES', and 'LucidDB'. The 'Access:' section lists 'Native (JDBC)', 'ODBC', and 'JNDI'. The 'Settings' section includes a 'Dialect:' dropdown set to 'Generic database', a 'Custom connection URL:' field with the value '.useLegacyDatetimeCode=false&serverTimezone=UTC', and a 'Custom driver class name:' field with the value 'com.mysql.cj.jdbc.Driver'. The 'Authentication' section has a 'Username:' field with the value 'root' and a 'Password:' field with masked characters. At the bottom, there are buttons for 'Test', 'Feature List', 'Explore', 'OK', and 'Cancel'.

Connection name:
db_test

Connection type:
Generic database
Google BigQuery
Greenplum
Gupta SQL Base
H2
Hadoop Hive
Hadoop Hive 2/3
Hive Warehouse Connector
Hypersonic
IBM DB2
Impala
Infobright
Informix
Ingres
Ingres VectorWise
Intersystems Cache
KingbaseES
LucidDB

Access:
Native (JDBC)
ODBC
JNDI

Settings
Dialect:
Generic database
Custom connection URL:
.useLegacyDatetimeCode=false&serverTimezone=UTC
Custom driver class name:
com.mysql.cj.jdbc.Driver

Authentication
Username:
root
Password:
.....

Test Feature List Explore OK Cancel

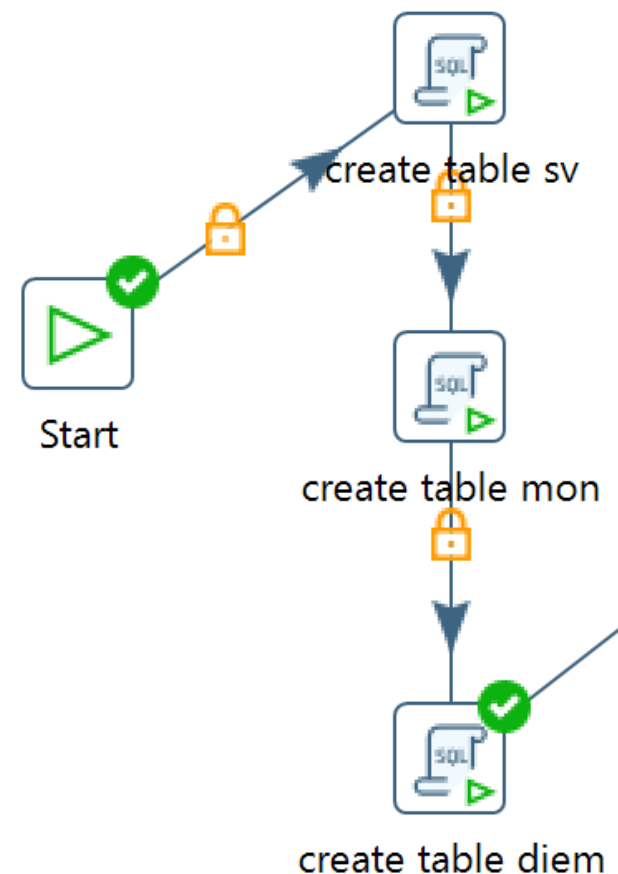
Job

Tạo entry SQL:

```
create table tbl_sinh_vien (  
ma_sv int,  
ho_ten varchar (100)  
);
```

```
create table tbl_mon (  
ma_mon int,  
ten_mon varchar (100)  
);
```

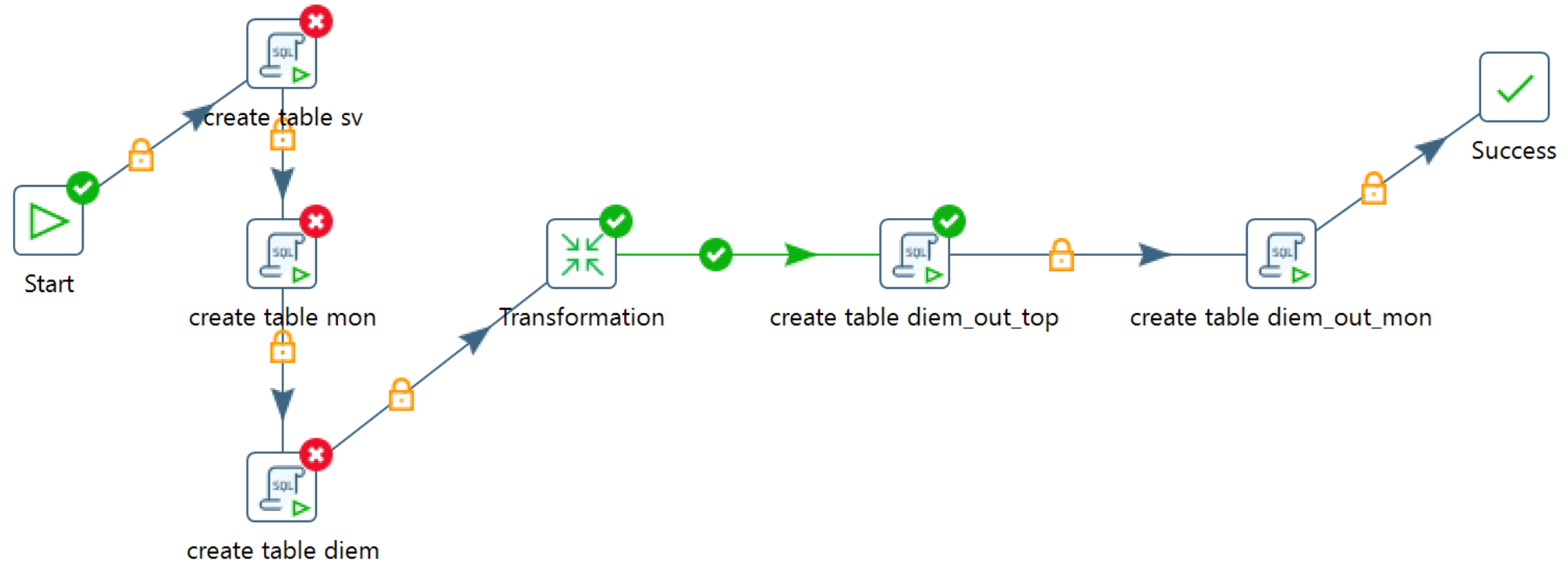
```
create table tbl_diem (  
ma_sv int,  
ma_mon int,  
diem double  
);
```



Trans



JOB



Hướng dẫn

1. Hướng dẫn dùng wildcard

Tính năng wildcard của Pentaho cho phép dùng regular expression để truy vấn một danh sách file hoặc folder trong một thư mục. Cách dùng như sau:

Mẫu	Thay thế cho	Ví dụ
<code>.*</code>	0 hoặc nhiều ký tự bất thể	<code>abc.*xyz</code> tìm các file bắt đầu bằng "abc" và kết thúc bằng "xyz"
<code>\.</code>	<u>dấu chấm</u> (.)	<code>abc\.</code> tìm file tên là "abc.doc"

2. Hướng dẫn dùng parameter và variable:

- Variable là biến môi trường mà ứng dụng ETL tự gán cho nó khi chạy job (hoặc transform). Parameter là biến môi trường do ứng dụng bên ngoài gán cho ETL trước khi chạy job (hoặc transform).
- Như vậy variable có ý nghĩa là biến dùng chung cho các job, các transform trong một ETL Slave. Parameter là biến riêng của một job hoặc transform nào đó, không liên quan đến job, transform khác.

Hướng dẫn

1. Hướng dẫn dùng wildcard

Tính năng wildcard của Pentaho cho phép dùng regular expression để truy vấn một danh sách file hoặc folder trong một thư mục. Cách dùng như sau:

Mẫu	Thay thế cho	Ví dụ
<code>.*</code>	0 hoặc nhiều ký tự bất thể	<code>abc.*xyz</code> tìm các file bắt đầu bằng "abc" và kết thúc bằng "xyz"
<code>\.</code>	<u>dấu chấm</u> (.)	<code>abc\.</code> tìm file tên là "abc.doc"

2. Hướng dẫn dùng parameter và variable:

- Variable là biến môi trường mà ứng dụng ETL tự gán cho nó khi chạy job (hoặc transform). Parameter là biến môi trường do ứng dụng bên ngoài gán cho ETL trước khi chạy job (hoặc transform).
- Như vậy variable có ý nghĩa là biến dùng chung cho các job, các transform trong một ETL Slave. Parameter là biến riêng của một job hoặc transform nào đó, không liên quan đến job, transform khác.

Hướng dẫn

3. Khai báo biến

Bước 1: Trong màn hình cấu hình job (hoặc transform), click vào nút “View” ở góc trên bên trái màn hình (1), sau đó click đúp trái vào tên job (hoặc transform) đang cấu hình (2)

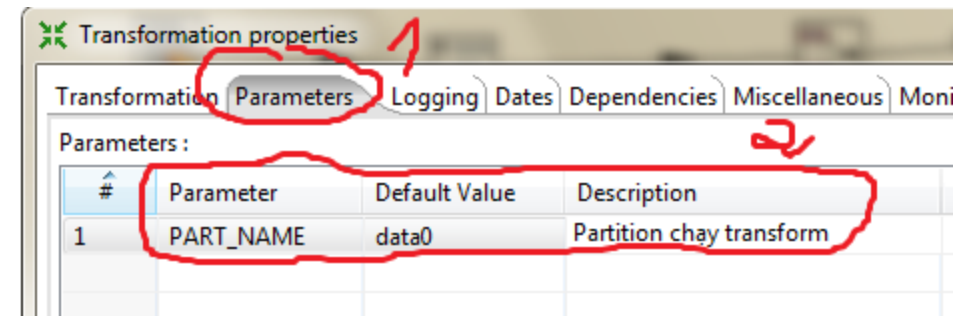
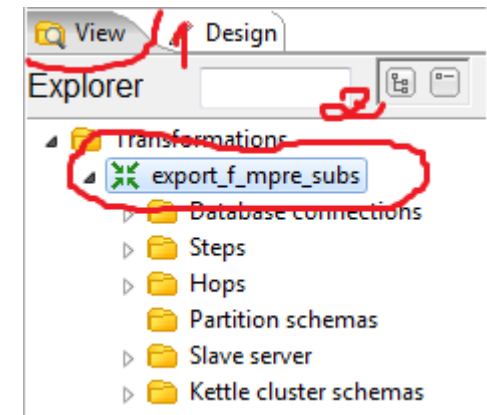
Bước 2: Trong màn hình “Transformation properties” mới hiện ra, chọn tab “Parameters”, sau đó điền thông tin parameter vào bảng

Parameters bên dưới theo thông tin sau:

Parameter: tên parameter (yêu cầu viết liền, có phân biệt viết hoa/viết thường, không có ký tự đặc biệt trừ dấu “_”)

Default value: giá trị mặc định gán cho biến nếu biến không được gán giá trị

Description: thông tin mô tả biến, chỉ có ý nghĩa với người cấu hình, không có ý nghĩa với hệ thống



Hướng dẫn

Thiết kế của Pentaho không cho phép chạy transformation độc lập mà transform phải nằm trong một job nào đó. Để truyền biến từ job sang transform làm như sau: Trong màn hình cấu hình job, click đúp vào transform muốn truyền biến, trong cửa sổ mới sinh ra chọn tab “Parameters”, sau đó điền giá trị vào bảng bên dưới:

Parameter: tên biến của transformation muốn gán giá trị

Stream column name: N/A

Value: giá trị biến truyền cho transformation

Job entry details for this transformation:

Name of job entry: export_f_mpre_subs

Transformation specification | Advanced | Logging settings | Argument | **Parameters**

Pass all parameter values down to the sub-transformation ☒

#	Parameter	Stream column name	Value
1	PART_NAME		\${PART_NAME}

Hướng dẫn

4. Join 2 luồng dữ liệu

Bước 1: sắp xếp luồng dữ liệu theo cột điều kiện join:

Sort rows

Step name: sort_fact

Sort directory: D:\test_data\huong_dan\sort_temp

TMP-file prefix: sort_fact

Sort size (rows in memory): 100

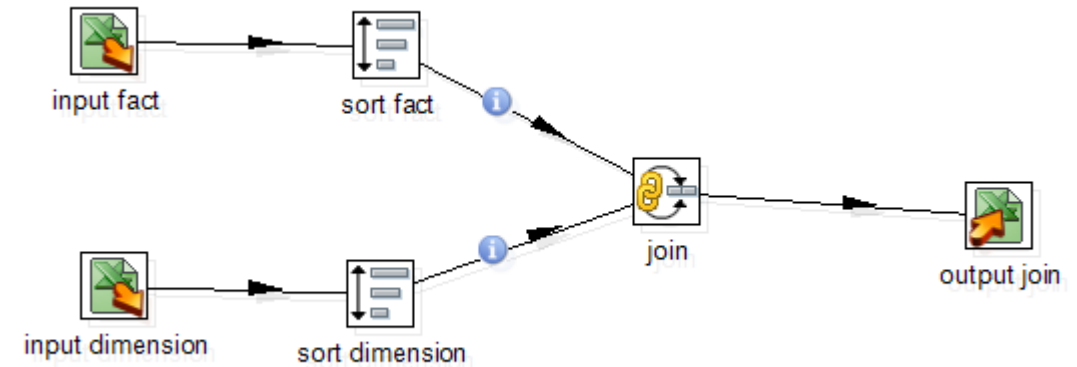
Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields :

#	Fieldname	Ascending	Case sensitive compare?
1	dimension_id	N	Y



Hướng dẫn

4. Join 2 luồng dữ liệu

Bước 2: Kéo step “Merge Join” vào transform, sau đó nối 2 luồng muốn join vào step. Cấu hình như sau:

Step name: tên step, chú ý đặt tên có tính gợi nhớ

First Step: tên luồng 1

Second Step: tên luồng 2

Step name: join

First Step: sort fact

Second Step: sort dimension

Join Type: LEFT OUTER

Keys for 1st step:

#	Key field	
1	dimension_id	

Keys for 2nd step:

#	Key field	
1	id	

Hướng dẫn

5. Tổng hợp số liệu

- Bước 1: Trong màn hình cấu hình transformation kéo step “Sort rows” vào, sau đó nối luồng dữ liệu cần join vào step sort

Step name: tên step, nên đặt tên có tính gợi nhớ để tiện tra log sau này

Sort directory: vị trí lưu file tạm sinh ra trong quá trình sort (sau khi sort xong file tạm bị xoá hết), mặc định là thư mục chứa file tạm của JVM (thể hiện bằng biến %%java.io.tmpdir%%)

TMP-file prefix: chuỗi ký tự thêm vào đầu file tạm (chú ý là step sort tự nó ghi nhớ đã sinh ra file tạm nào để xoá nên dù trong transform có nhiều step sort mà đặt biến TMP-file prefix giống nhau cũng không bị sai nghiệp vụ)

Sort size: số bản ghi lưu trong bộ nhớ, nếu vượt quá số lượng này thì step sẽ ghi vào file tạm, để giá trị 0 là không ghi ra file (chú ý cân đối số lượng bản ghi này, nếu để ít sẽ dùng ít memory nhưng tốc độ chậm, nếu để nhiều sẽ nhanh hơn nhưng tốn memory, nếu để nhiều quá có thể gặp lỗi Out of memory)

Free memory threshold: N/A

Compress TMP Files: N/A

Only pass unique rows: N/A

Bảng “Field”



Sort rows

Step name: sort fact

Sort directory: D:\test_data\huong_dan\sort_temp

TMP-file prefix: sort_fact

Sort size (rows in memory): 100

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields:

#	Fieldname	Ascending	Case sensitive compare?
1	dimension_id	N	Y

Hướng dẫn

5. Tổng hợp số liệu

Bước 2: Kéo step “Group by” vào, sau đó nối luồng từ step sort ở trên vào step “Group by” mới tạo. Cấu hình như sau:

Step name: tên step, chú ý đặt tên có tính gợi nhớ

Include all rows: N/A

Temporary files directory: N/A

TMP-file prefix: N/A

Add line number, restart in each group: N/A

Line number field name: N/A

Always give back a result row: N/A

Bảng “The fields that make up the group”

Step name: tong hop

Include all rows? ☐

Temporary files directory: %%java.io.tmpdir%%

TMP-file prefix: grp

Add line number, restart in each group ☐

Line number field name:

Always give back a result row ☐

The fields that make up the group:

#	Group field
1	dimension_id

Aggregates :

#	Name	Subject	Type
1	count	dimension_id	Number of Values (N)
2	sum	value	Sum
3	count_distinct	value	Number of Distinct Values (N)

Hướng dẫn

6. Sinh file theo partition

Bước 1: Kéo step “Modified Java Script Value” vào transform, nối luồng sau đó config như sau:

Step name: tên step, chú ý đặt tên có tính gợi nhớ

Java script: viết code javascript tạo ra cột dữ liệu làm cờ phân biệt partition. Trong ví dụ bên dưới cột dữ liệu tên là partition_column, tạo ra bằng cách cộng chuỗi “data” với giá trị cột “dimension_id”.

Bảng “field”:

Fieldname: điền tên cột làm cờ partition đã khai báo ở trên

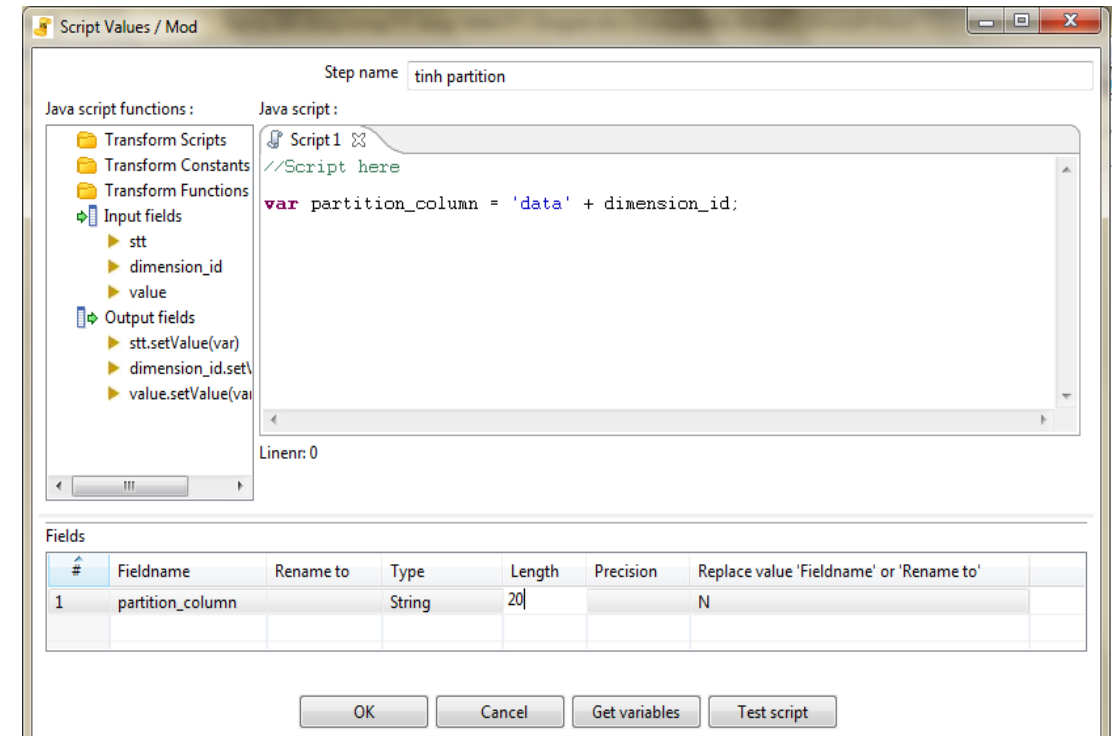
Rename to: để trống

Type: chọn String

Length: đặt theo độ dài trường dữ liệu

Precision: để trống

Replace value ‘Fieldname’ or ‘Rename to’: chọn N



Hướng dẫn

6. Sinh file theo partition

Bước 2: Hiện việc sinh file theo partition chỉ làm việc với step “Text file output” . Cấu hình như sau:

Tab “File”:

Filename: tên file

Create Parent folder: có tạo thư mục đích không

Extension: đuôi file (hiện chỉ hỗ trợ excel 2003, việc đổi đuôi file không làm thay đổi định dạng)

Specify datetime format: thêm ngày tháng vào tên file mới tạo không

Date time format: format tên file

Add filenames to result: thêm file mới tạo vào biến result

Tab “Content”:

Separator: ký tự ngăn trường dữ liệu

Enclosure: ký tự bao trường dữ liệu

Force the enclosure around fields: N/A

Disable the enclosure fix: N/A

Header: có thêm header cho file dữ liệu không (là dòng chứa tên trường dữ liệu)

Footer: N/A

Format: format file

Compression: có nén file không

Encoding: mã encode file sinh ra

Right pad field: N/A

Fast data dump: không format

Split every... rows: Tách thành nhiều file theo số bản ghi

Add ending line of file: Chèn thêm chuỗi vào dòng cuối cùng

Write to Partition Files: có ghi file theo partition không, chọn

Partitioning Field: tên cột dữ liệu làm cờ partition: chọn “partitioning column” (đã khai báo ở step trước)

Hướng dẫn

6. Sinh file theo partition

Tab “Fields”:

Name: tên cột dữ liệu

Type: loại dữ liệu

Format: format hiển thị dữ liệu trên file

Length: độ dài trường dữ liệu

Precision: N/A

Currency: N/A

Decimal: N/A

Group: N/A

Trim type: có cắt đầu/đuôi chuỗi ký tự không

Null: N/A

Step này sẽ ghi dữ liệu ra file text, ứng với mỗi giá trị trong cột “partitioning_column” sinh ra một file

Ví dụ: file transform 04_trans_input_partition.ktr

Hướng dẫn

7. Thay đổi giá trị trên cột dữ liệu có sẵn

Cách làm: Kéo step “Modified Java Script Value” vào transform, nối luồng sau đó config như sau:

Step name: tên step, chú ý đặt tên có tính gợi nhớ

Java script: viết code javascript thao tác trên dữ liệu muốn sửa giá trị

Bảng “field”:

Fieldname: điền tên cột dữ liệu đã sửa giá trị

Rename to: để trống

Type: chọn kiểu dữ liệu tương ứng với cột sửa giá trị

Length: đặt theo độ dài cột dữ liệu

Precision: đặt tương ứng độ dài cột dữ liệu muốn sửa

Replace value ‘Fieldname’ or ‘Rename to’: chọn Y

Sau step “Modified Java Script Value”, giá trị các cột dữ liệu được thay

Ví dụ: file transform 05_trans_sua_du_lieu_input.ktr

