http://www.disit.org

How to Create GTFS File Ingestion via ETL (Extract Transform Load): the case of Helsinki

Mirco Soderi

University of Florence, Department of Information Engineering,

DISIT Lab, https://www.snap4city.org

https://www.disit.org , https://www.sii-mobility.org, paolo.nesi@unifi.it, mirco.soderi@unifi.it

Get Started

Download a ready-to-use VM from:

http://www.disit.org/drupal/?q=node/6690

Unless you have a good reason to do otherwise,
I recommend you to pick one of these:

- **http://www.disit.org/vmsdetl/VMSDETL-2017-v0-8.rar**
- http://www.disit.org/vmsdetl/VMSDETL-2017-v0-8-ovf.rar

Decompress and run the VM with a tool of your choice. I use:

- VMware® Workstation 14 Player for Win10, version 14.1.7

If needed and possible, disable Windows Hyper-V Hypervisor in your host:

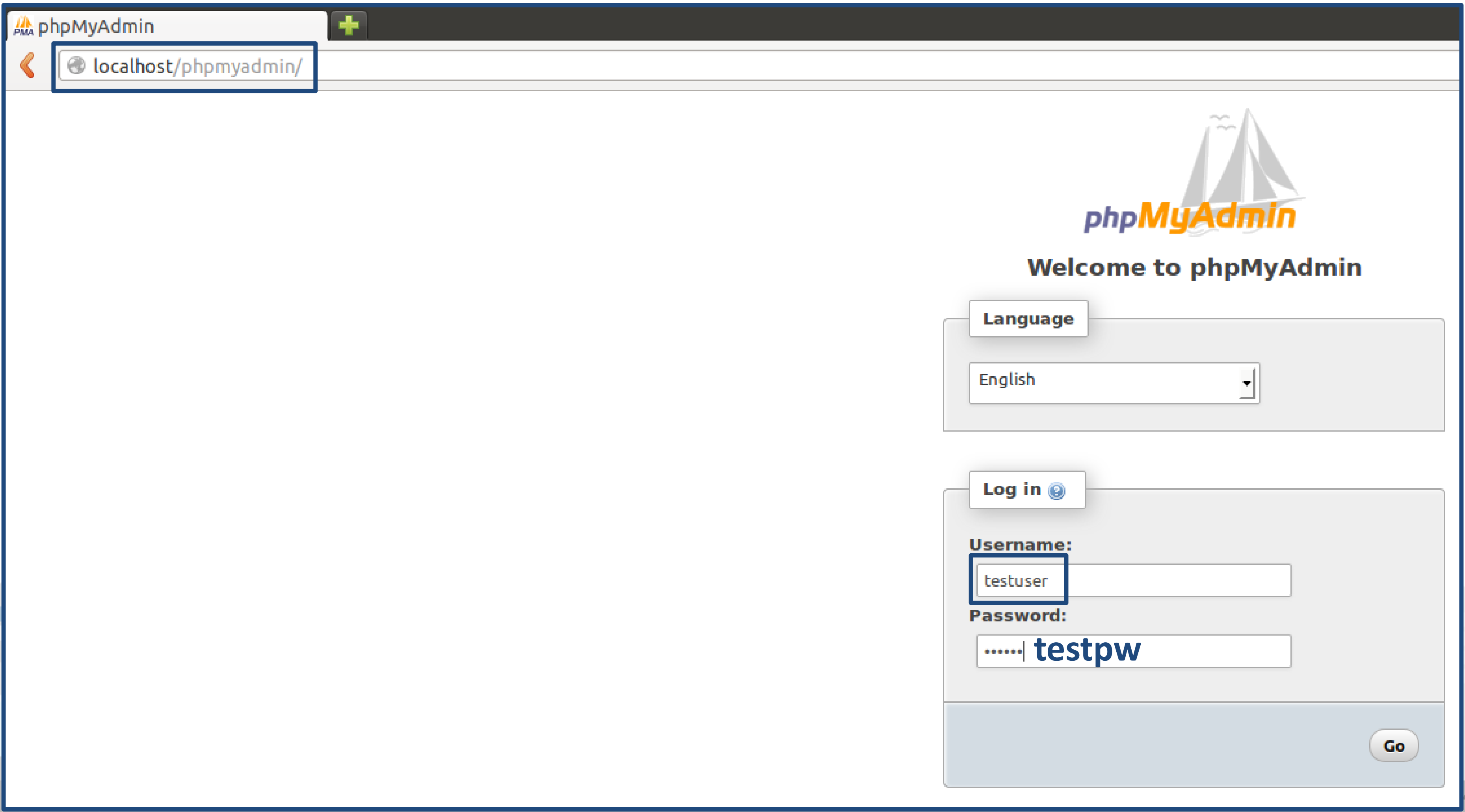
- https://stackoverflow.com/questions/39858200/vmware-workstation-and-device-credential-guard-are-not-compatible

Login with user **ubuntu** and password **ubuntu**

Create Folders

- Where you will put all of your jobs and transformations:
  - /home/ubuntu/Desktop/Trasformazioni/GTFS_Helsinki/Static/Ingestion
  - /home/ubuntu/Desktop/Trasformazioni/GTFS_Helsinki/Static/Triplification
- Where you will put files that you will download from the Internet:
  - /home/ubuntu/Desktop/Sources/TPLHelsinki/GTFS_Helsinki_zip_ST
- Where you will put your RDF triples:
  - /home/ubuntu/Desktop/Triples/TPLHelsinki/GTFS_Helsinki_zip_ST

Add MySql ETL Cfg Tbl row (1)



phpMyAdmin

localhost/phpmyadmin/

Welcome to phpMyAdmin

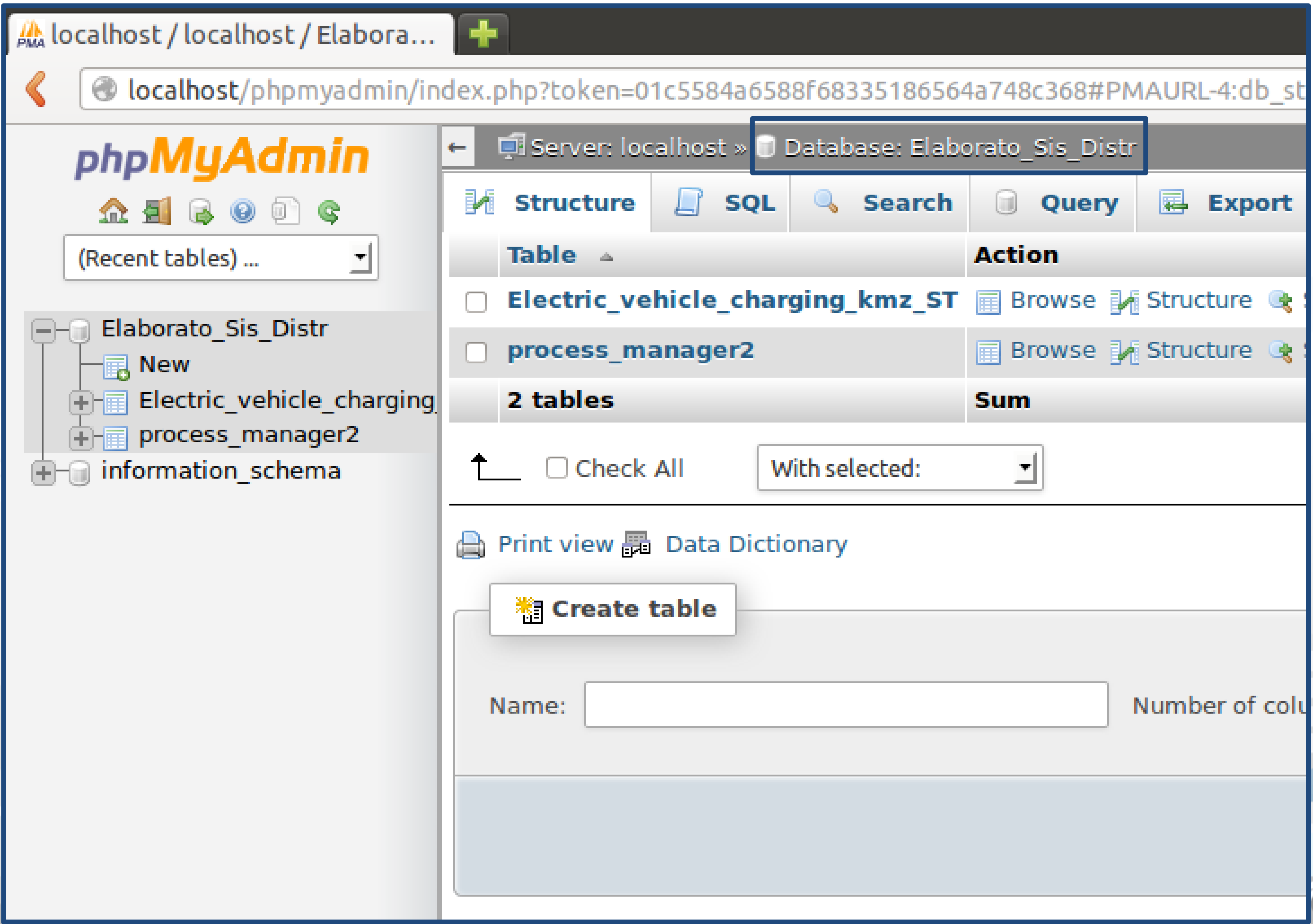
Language: English

Log in

Username: testuser

Password: •••••• testpw

Add MySql ETL Cfg Tbl row (2)


localhost / localhost / Elabora...

localhost/phpmyadmin/index.php?token=01c5584a6588f68335186564a748c368#PMAURL-4:db_st

Server: localhost » Database: Elaborato_Sis_Distr

Structure | SQL | Search | Query | Export

	Table	Action
☐	Electric_vehicle_charging_kmz_ST	Browse Structure
☐	process_manager2	Browse Structure
	2 tables	Sum

Check All | With selected:

Print view | Data Dictionary

Create table

Name: | Number of colu

Add MySql ETL Cfg Tbl row (3)

```
INSERT INTO `Elaborato_Sis_Distr`.`process_manager2` (`process`, `Resource`,
`Resource_Class`, `Category`, `Format`, `Automaticity`, `Process_type`, `Access`,
`Real_time`, `Source`, `A`, `B`, `C`, `D`, `E`, `status_A`, `status_B`, `status_C`, `status_D`,
`status_E`, `time_A`, `time_B`, `time_C`, `time_D`, `time_E`, `exec_A`, `exec_B`, `exec_C`,
`exec_D`, `exec_E`, `error_A`, `error_B`, `error_C`, `error_D`, `error_E`, `period`,
`overtime`, `param`, `last_update`, `last_triples`, `Triples_count`,
`Triples_countRepository`, `triples_insertDate`, `error`, `description`, `url_web_disit`,
`SecurityLevel`, `LicenseUrl`, `LicenseText`, `LicenseModel`, `startAt`) VALUES
('GTFS_Helsinki_zip_ST', 'TPLHelsinki', NULL, 'TPLHelsinki', 'zip', 'manual', 'ETL', 'HTTP',
'no', 'https://transitfeeds.com/p/helsinki-regional-transport/735', NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, 'no', NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
'https://dev.hsl.fi/gtfs/hsl.zip', NULL, NULL, NULL, NULL, NULL, NULL, 'GTFS di Helsinki',
NULL, NULL, NULL, NULL, NULL, NULL);
```

Create MySQL table of agencies

```
CREATE TABLE GTFS_Helsinki_zip_ST_agency
(
FinalKey text DEFAULT NULL,
agency_timezone text DEFAULT NULL,
process text DEFAULT NULL,
agency_name text DEFAULT NULL,
agency_url text DEFAULT NULL,
agency_phone text DEFAULT NULL,
actualDate text DEFAULT NULL,
agency_id text DEFAULT NULL,
AgencyTXTKey text DEFAULT NULL,
agency_lang text DEFAULT NULL,
timestamp text DEFAULT NULL
);
```

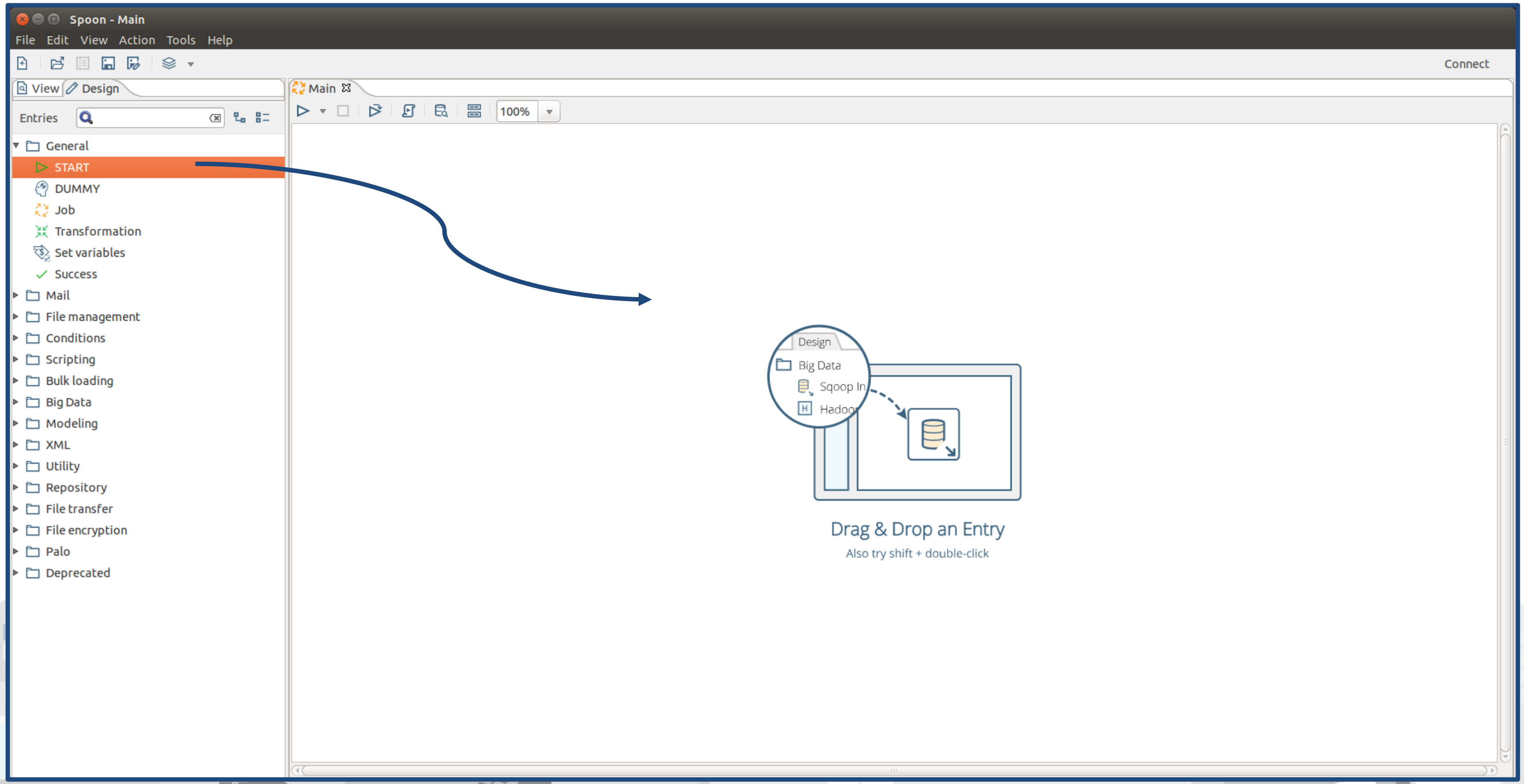
Static Ingestion

Create your first Job

- Open the terminal and issue the following:
  - **`spoon.sh`**
- Once Spoons has opened:
  - File → New → Job
  - File → Save
    - Move to /home/ubuntu/Desktop/Trasformazioni/GTFS_Helsinki/Static/Ingestion
    - Save your first job with name **Main.kjb**

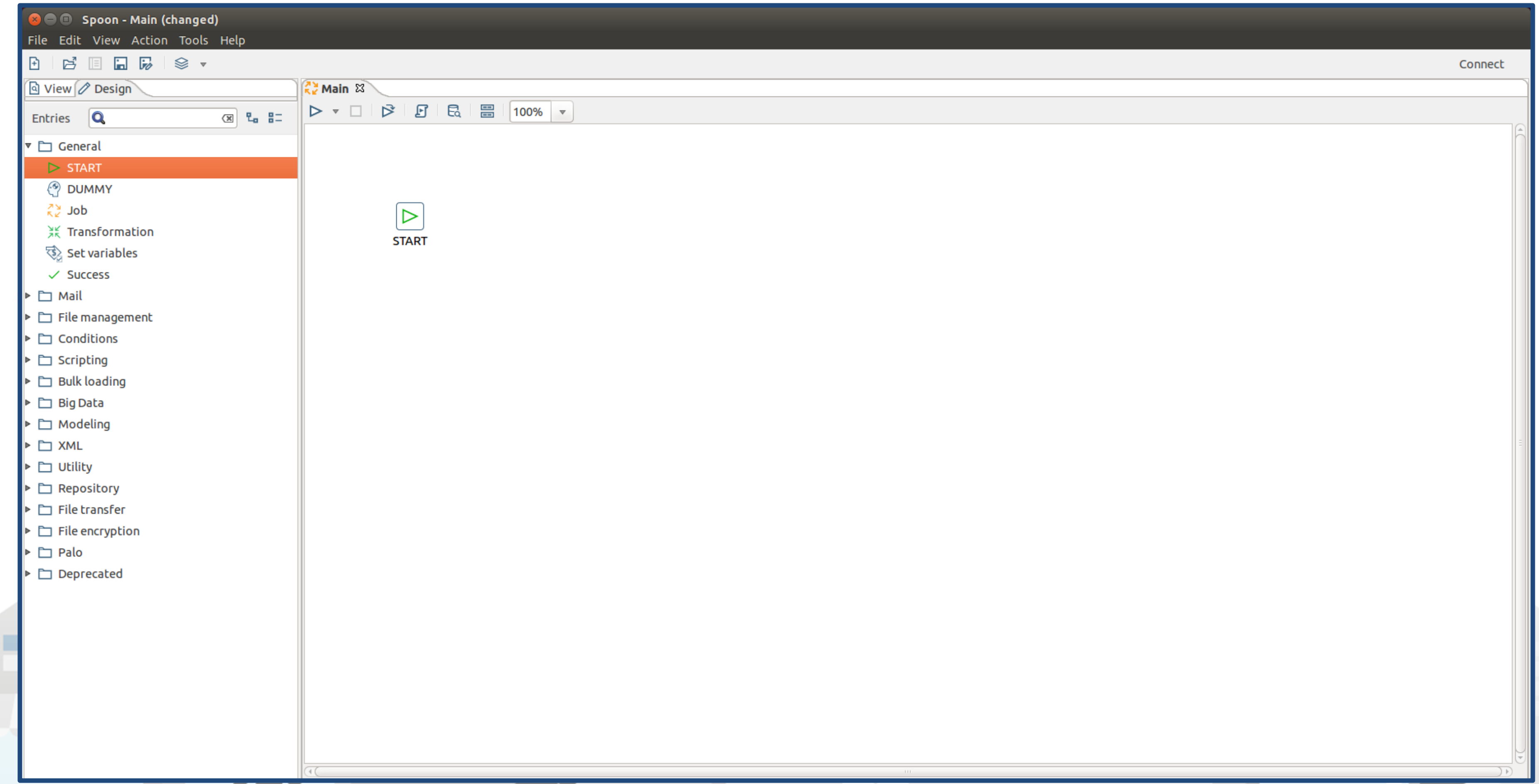
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org | SNAP4CITY

Add the START step (1)



http://www.disit.org

Add the START step (2)



Config Transformation Step (1)

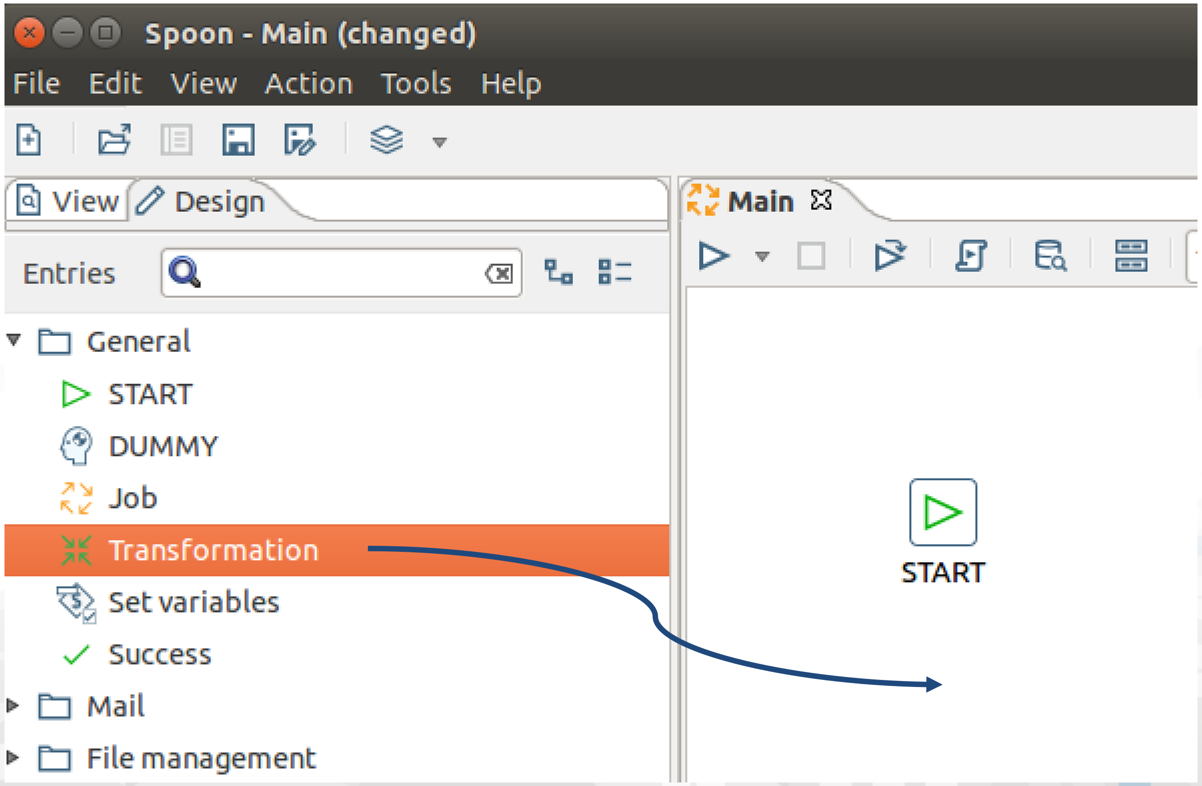
In your host, you have a ready-to-use transformation, located at /home/ubuntu/Desktop/Trasformazioni/getConfig.ktr, that we include as-is at the beginning of (nearly) all ETL jobs.

So, we now have to include this transformation in our newly created job, through the following steps:

1. **Add** an empty Transformation step
2. **Link** the Transformation step to the START step
3. **Configure** the Transformation step specifying the transformation to be performed

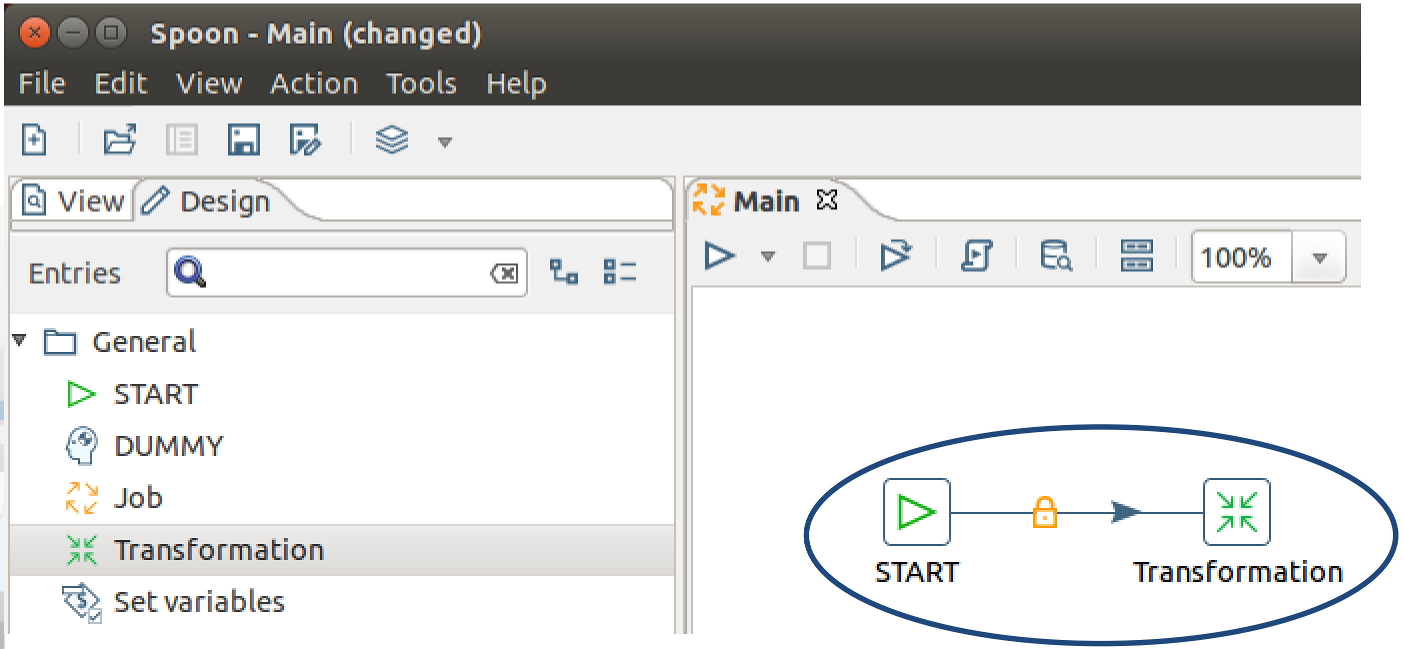
UNIVERSITÀ DEGLI STUDI FIRENZE DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org

Add a Transformation step



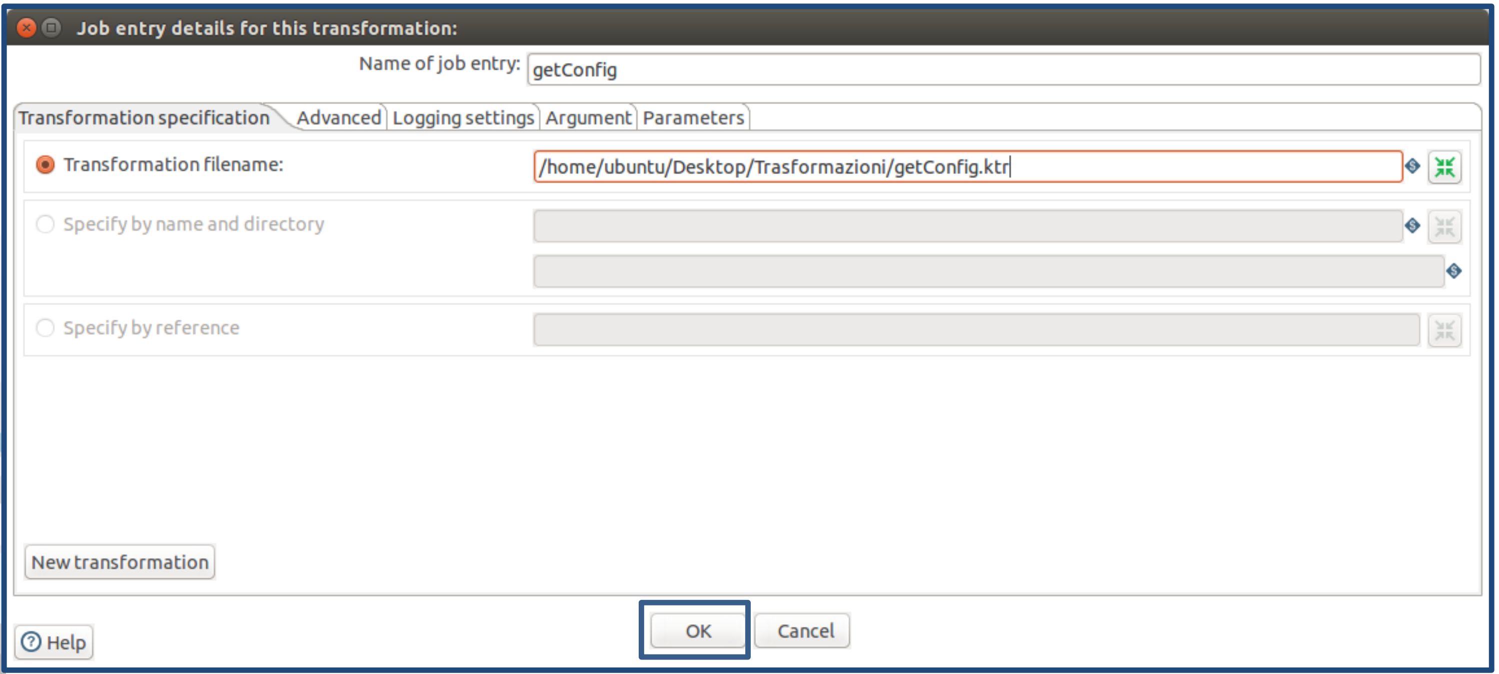
Link Transformation to START

- Press SHIFT and hold it down
- Left-click the **START** step and hold the mouse button down
- Move on the **Transformation** step
- Release both the mouse and the keyboard buttons



Configure Transformation step

- Double-click the **Transformation** step
- Rename the step from **Transformation** to **getConfig**
- Hit the button at the right of **Transformation filename** box, and pick /home/ubuntu/Desktop/Trasformazioni/getConfig.ktr



Job entry details for this transformation:

Name of job entry: getConfig

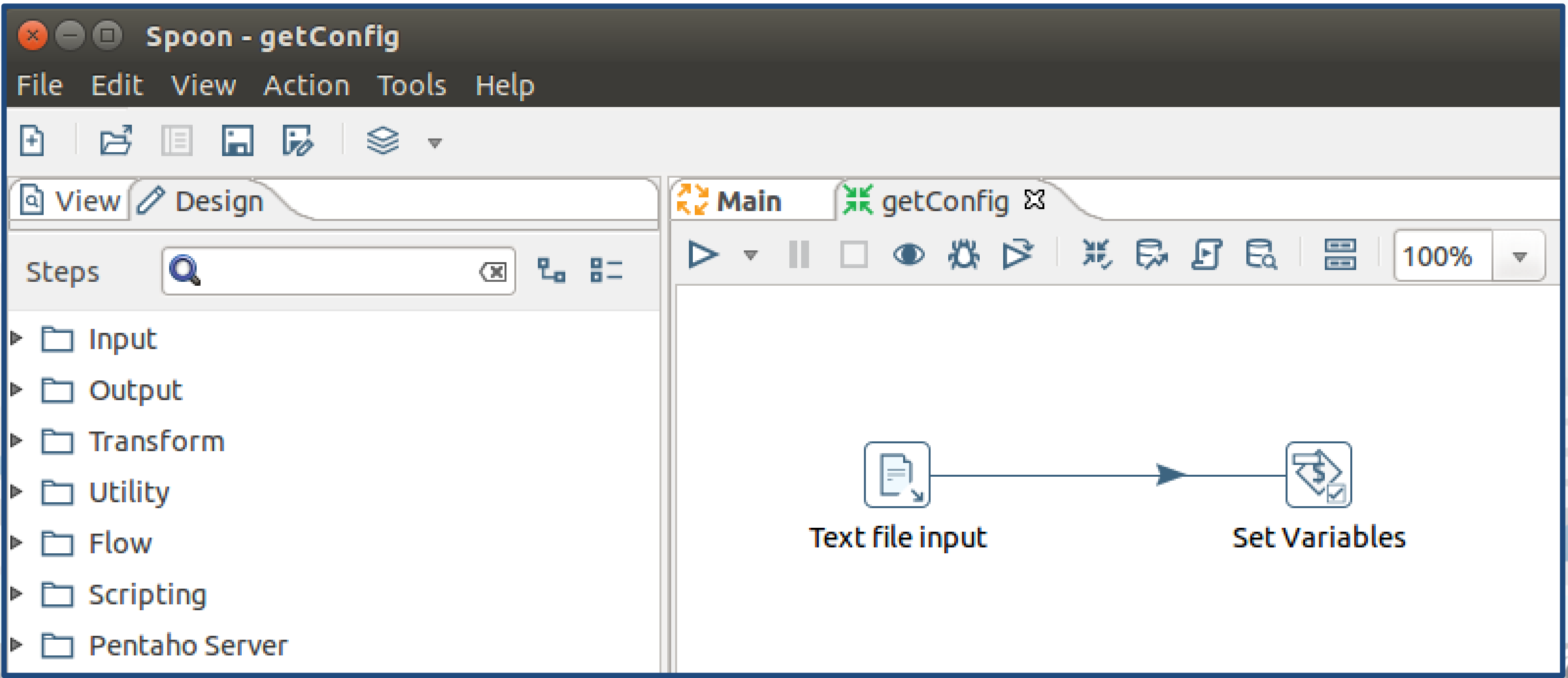
[✓ Transformation filename:] /home/ubuntu/Desktop/Trasformazioni/getConfig.ktr

Specify by name and directory

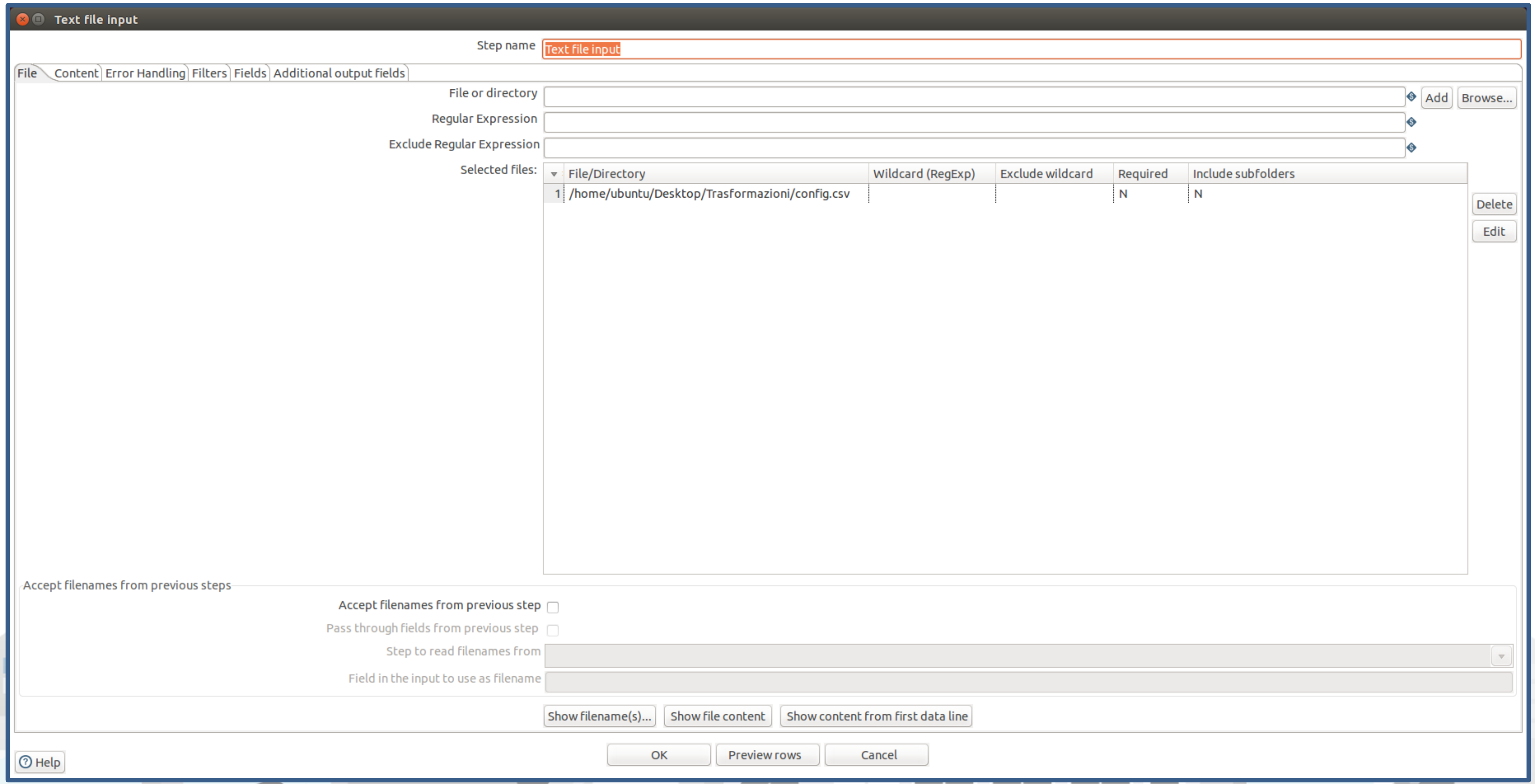
Specify by reference

What have we added? (1)

- Right-click the **getConfig** step
- Find **Open Referenced Object**
- Click **Transformation**



What have we added? (2.1)



Text file input

Step name: Text file input

File | Content | Error Handling | Filters | Fields | Additional output fields

File or directory:

Regular Expression:

Exclude Regular Expression:

Selected files:

	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1	/home/ubuntu/Desktop/Trasformazioni/config.csv			N	N

Accept filenames from previous steps

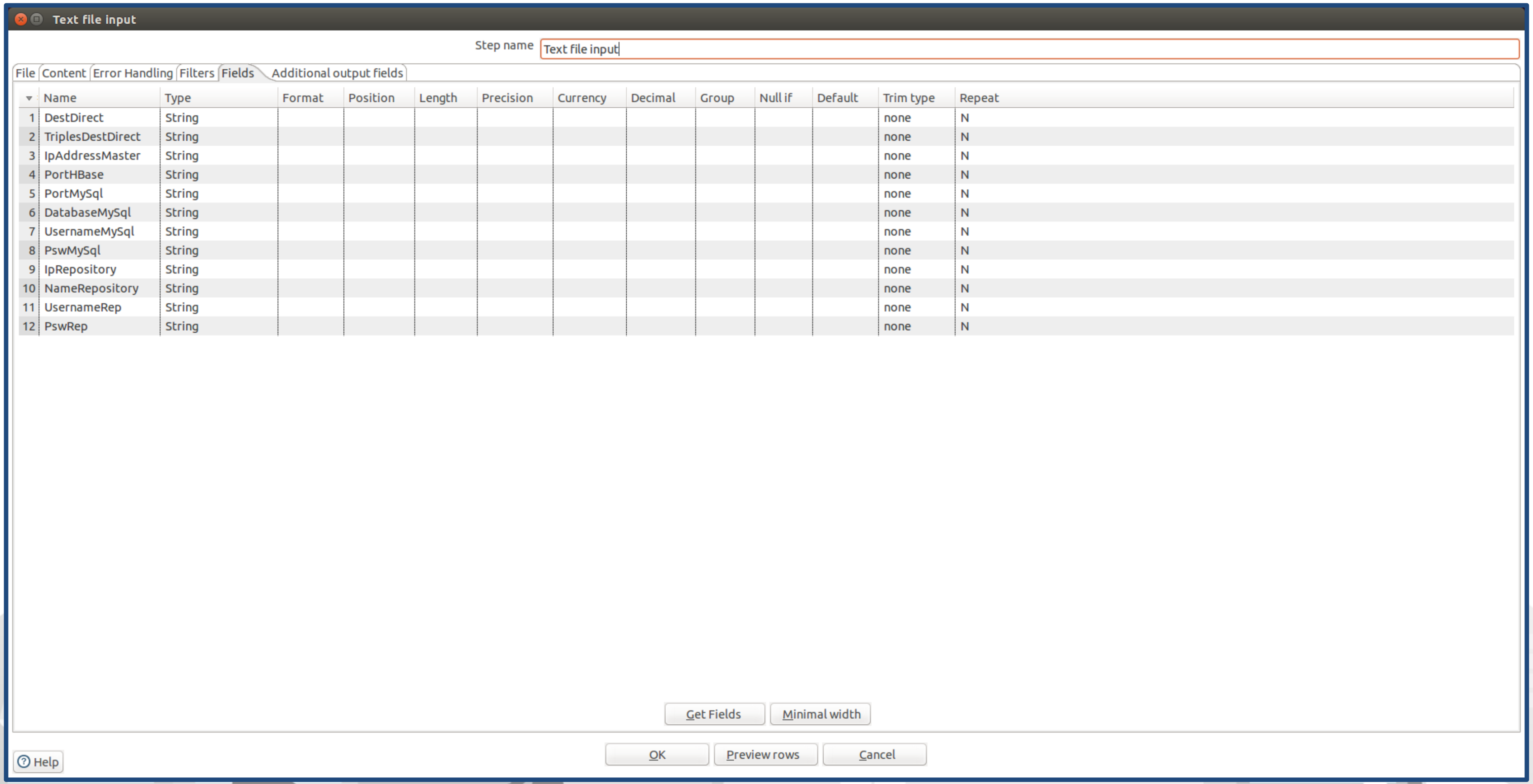
Accept filenames from previous step: ☐

Pass through fields from previous step: ☐

Step to read filenames from:

Field in the input to use as filename:

What have we added? (2.2)

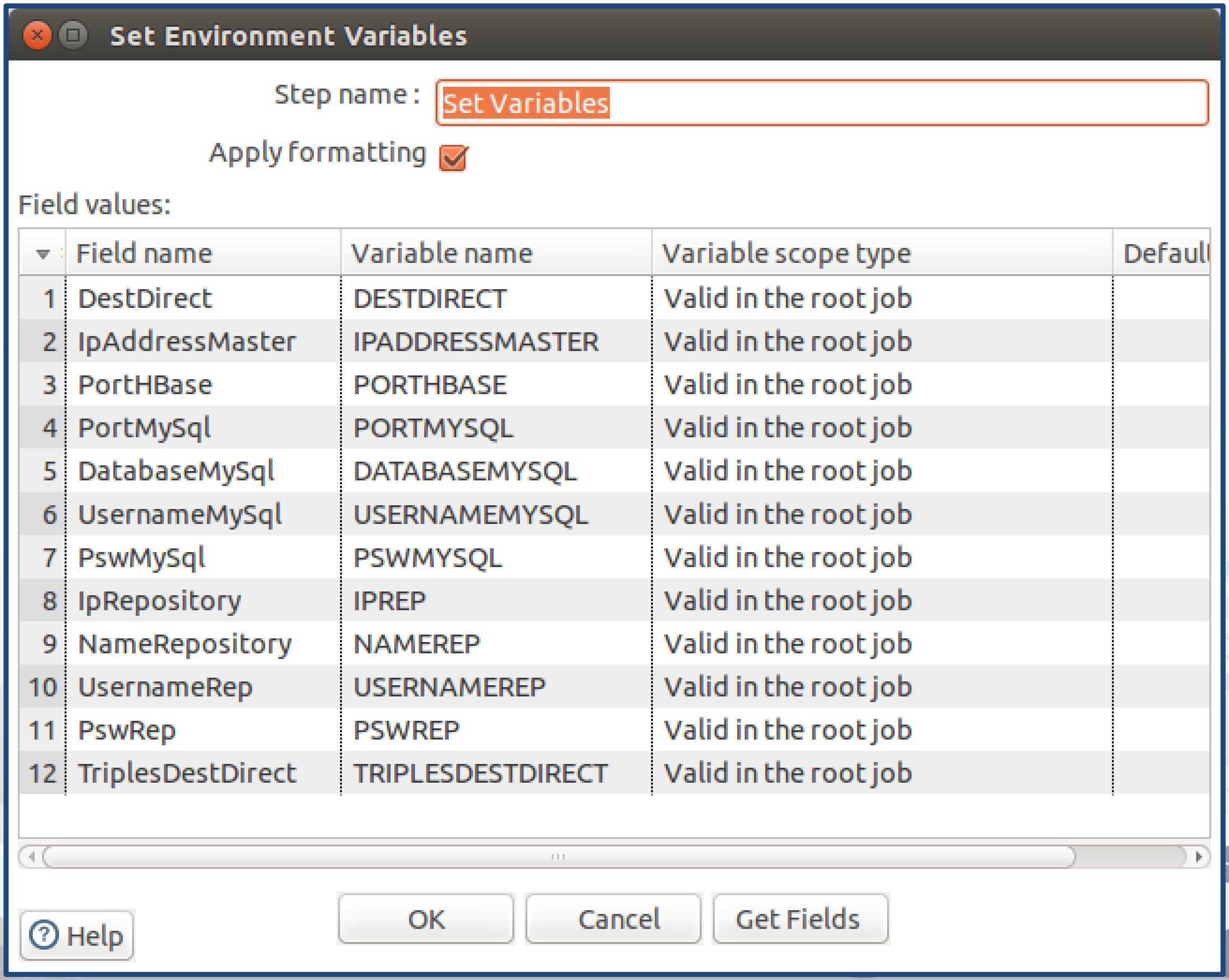


Step name: Text file input

	Name	Type	Format	Position	Length	Precision	Currency	Decimal	Group	Null if	Default	Trim type	Repeat
1	DestDirect	String										none	N
2	TriplesDestDirect	String										none	N
3	IpAddressMaster	String										none	N
4	PortHBase	String										none	N
5	PortMySql	String										none	N
6	DatabaseMySql	String										none	N
7	UsernameMySql	String										none	N
8	PswMySql	String										none	N
9	IpRepository	String										none	N
10	NameRepository	String										none	N
11	UsernameRep	String										none	N
12	PswRep	String										none	N

http://www.disit.org

What have we added? (3)



Step name : Set Variables

Apply formatting [✓]

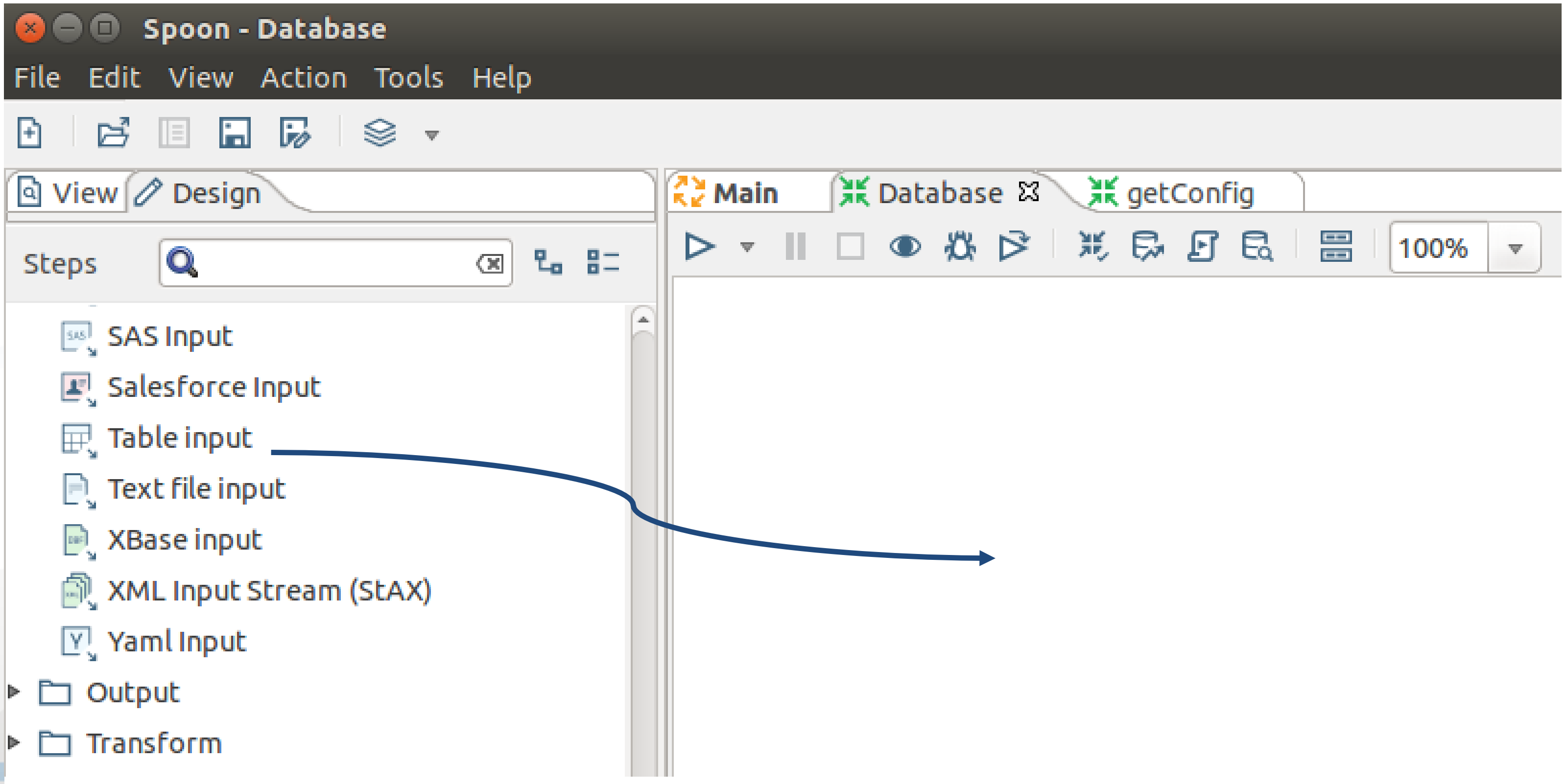
Field values:

	Field name	Variable name	Variable scope type	Defaul
1	DestDirect	DESTDIRECT	Valid in the root job	
2	IpAddressMaster	IPADDRESSMASTER	Valid in the root job	
3	PortHBase	PORTHBASE	Valid in the root job	
4	PortMySql	PORTMYSQL	Valid in the root job	
5	DatabaseMySql	DATABASEMYSQL	Valid in the root job	
6	UsernameMySql	USERNAMEMYSQL	Valid in the root job	
7	PswMySql	PSWMYSQL	Valid in the root job	
8	IpRepository	IPREP	Valid in the root job	
9	NameRepository	NAMEREP	Valid in the root job	
10	UsernameRep	USERNAMEREP	Valid in the root job	
11	PswRep	PSWREP	Valid in the root job	
12	TriplesDestDirect	TRIPLESDESTDIRECT	Valid in the root job	

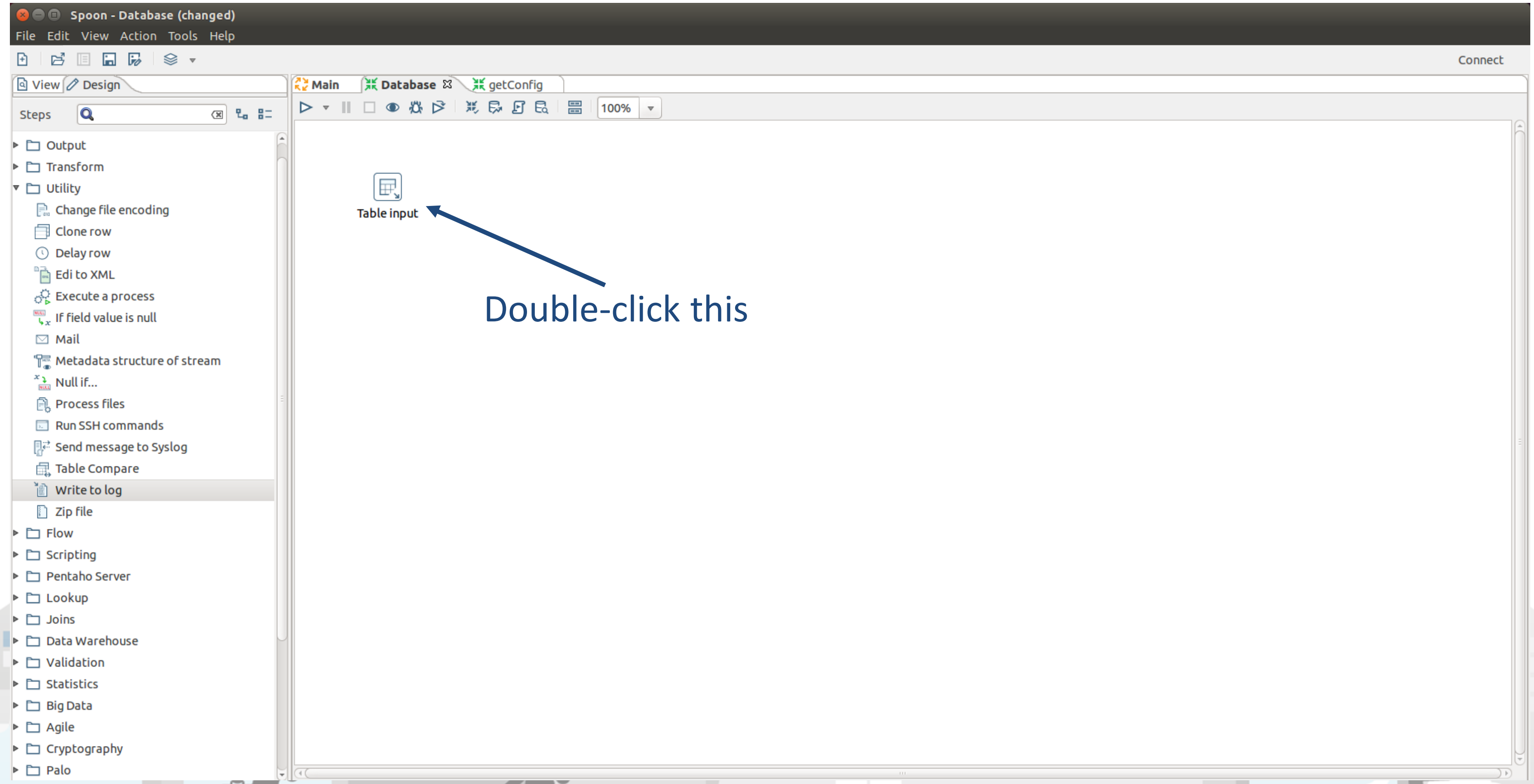
Create a new Transformation

- We will now create a new Transformation where we will read the MySQL database to retrieve configuration parameters that are specific of the ETL that we are now developing:
  - File → New → Transformation
  - File → Save
    - Move to /home/ubuntu/Desktop/Trasformazioni/GTFS_Helsinki/Static
    - Save your first transformation with name **Database.ktr**

Read from MySQL database (1)



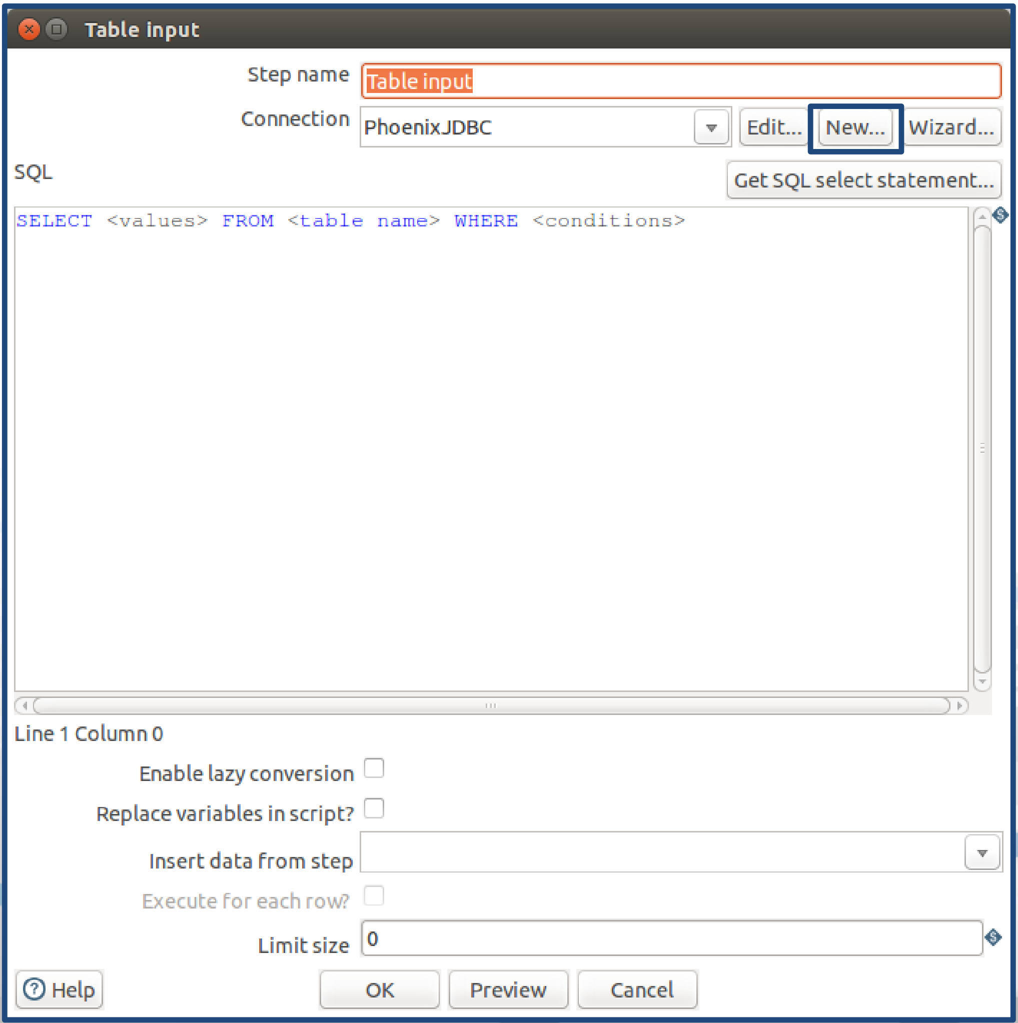
Read from MySQL database (2)



Double-click this

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org | SNAP4CITY | KM4CITY

Read from MySQL database (3)



Step name: Table input

Connection: PhoenixJDBC

SQL

```
SELECT <values> FROM <table name> WHERE <conditions>
```

Line 1 Column 0

Enable lazy conversion ☐

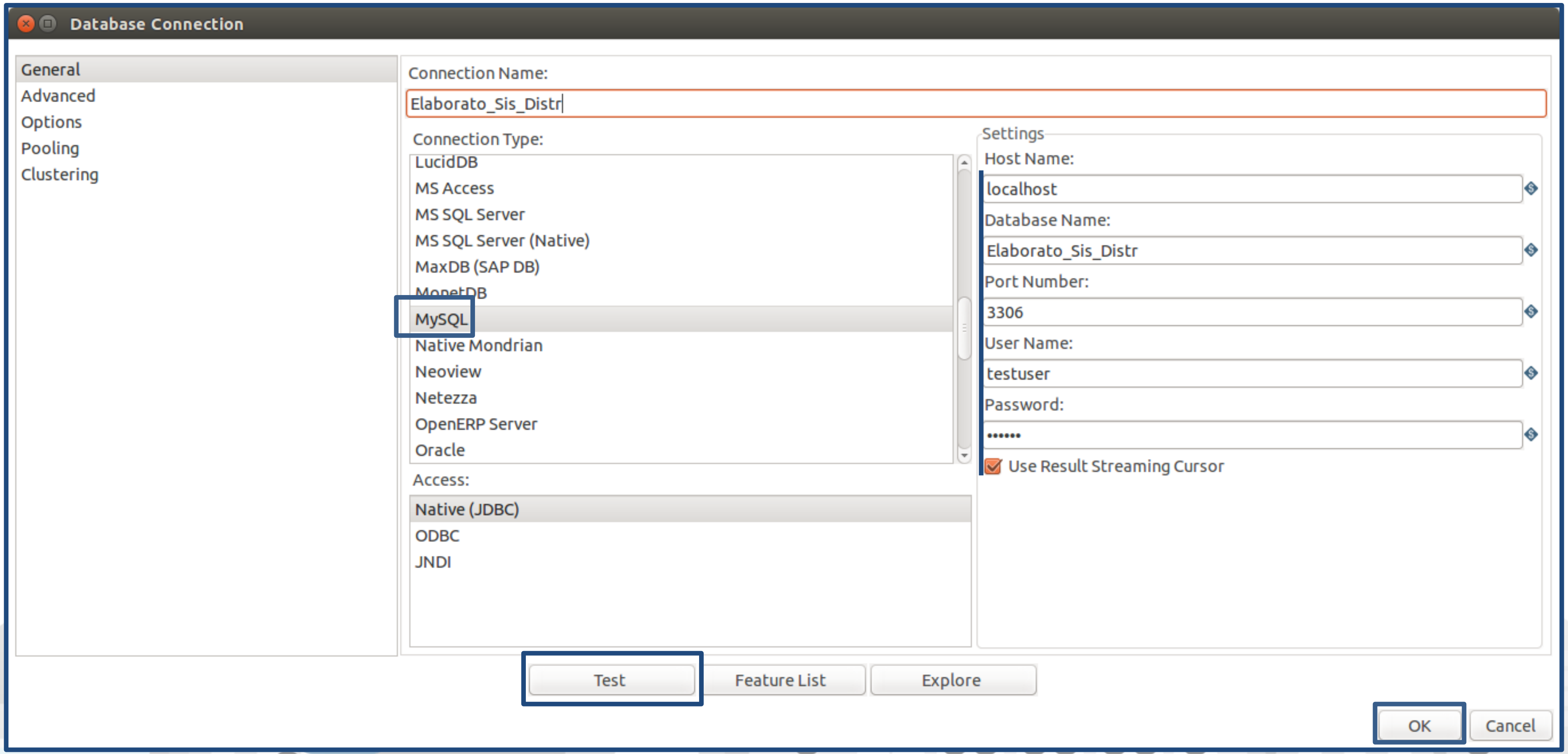
Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size 0

Read from MySQL database (4)


Database Connection

General
Advanced
Options
Pooling
Clustering

Connection Name: Elaborato_Sis_Distr

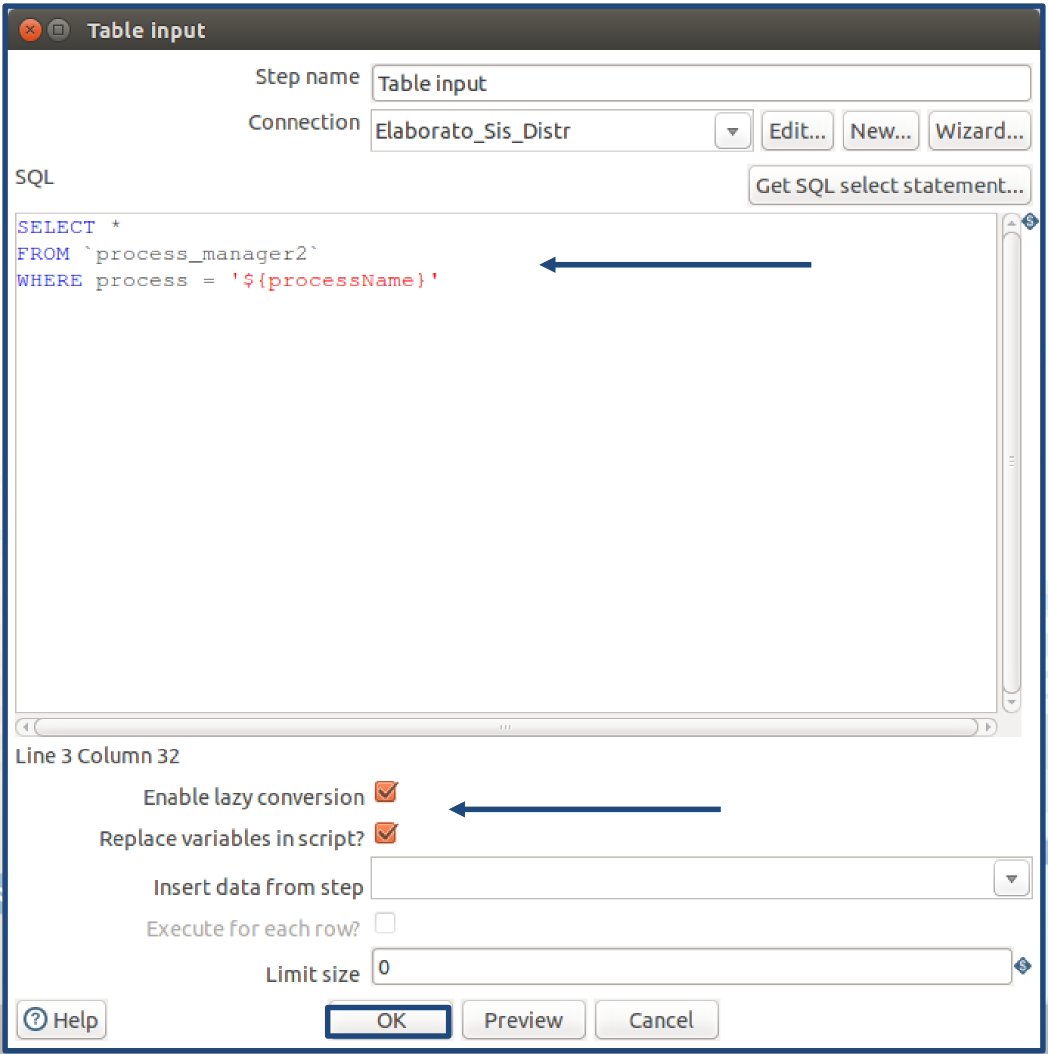
Connection Type:
LucidDB
MS Access
MS SQL Server
MS SQL Server (Native)
MaxDB (SAP DB)
MonetDB
MySQL
Native Mondrian
Neoview
Netezza
OpenERP Server
Oracle

Access:
Native (JDBC)
ODBC
JNDI

Settings
Host Name: localhost
Database Name: Elaborato_Sis_Distr
Port Number: 3306
User Name: testuser
Password: ••••••
[✓] Use Result Streaming Cursor

http://www.disit.org

Read from MySQL database (5)



Step name: Table input

Connection: Elaborato_Sis_Distr

```
SELECT *
FROM `process_manager2`
WHERE process = '${processName}'
```

Line 3 Column 32

Enable lazy conversion ✓

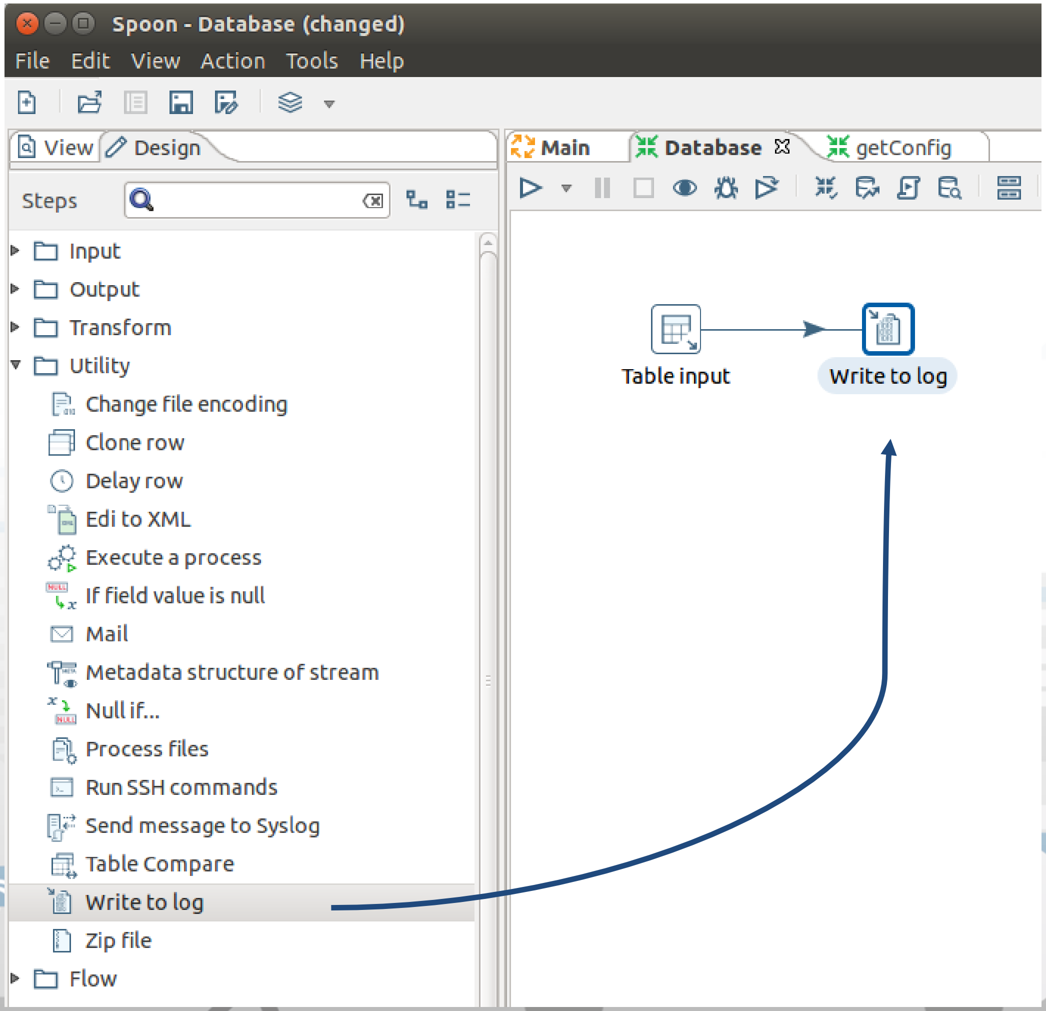
Replace variables in script? ✓

Insert data from step:

Execute for each row? ☐

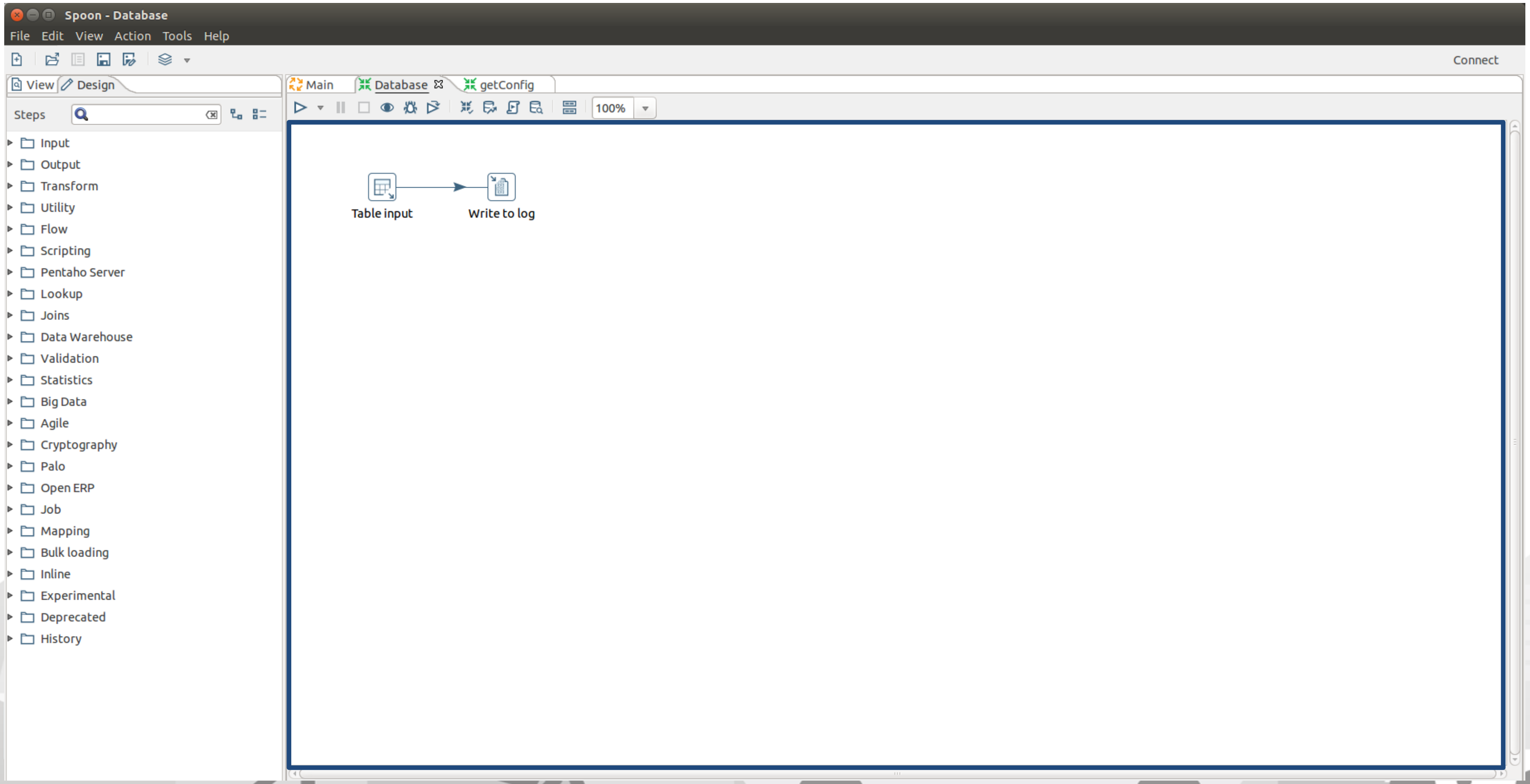
Limit size: 0

Add a Write to log step



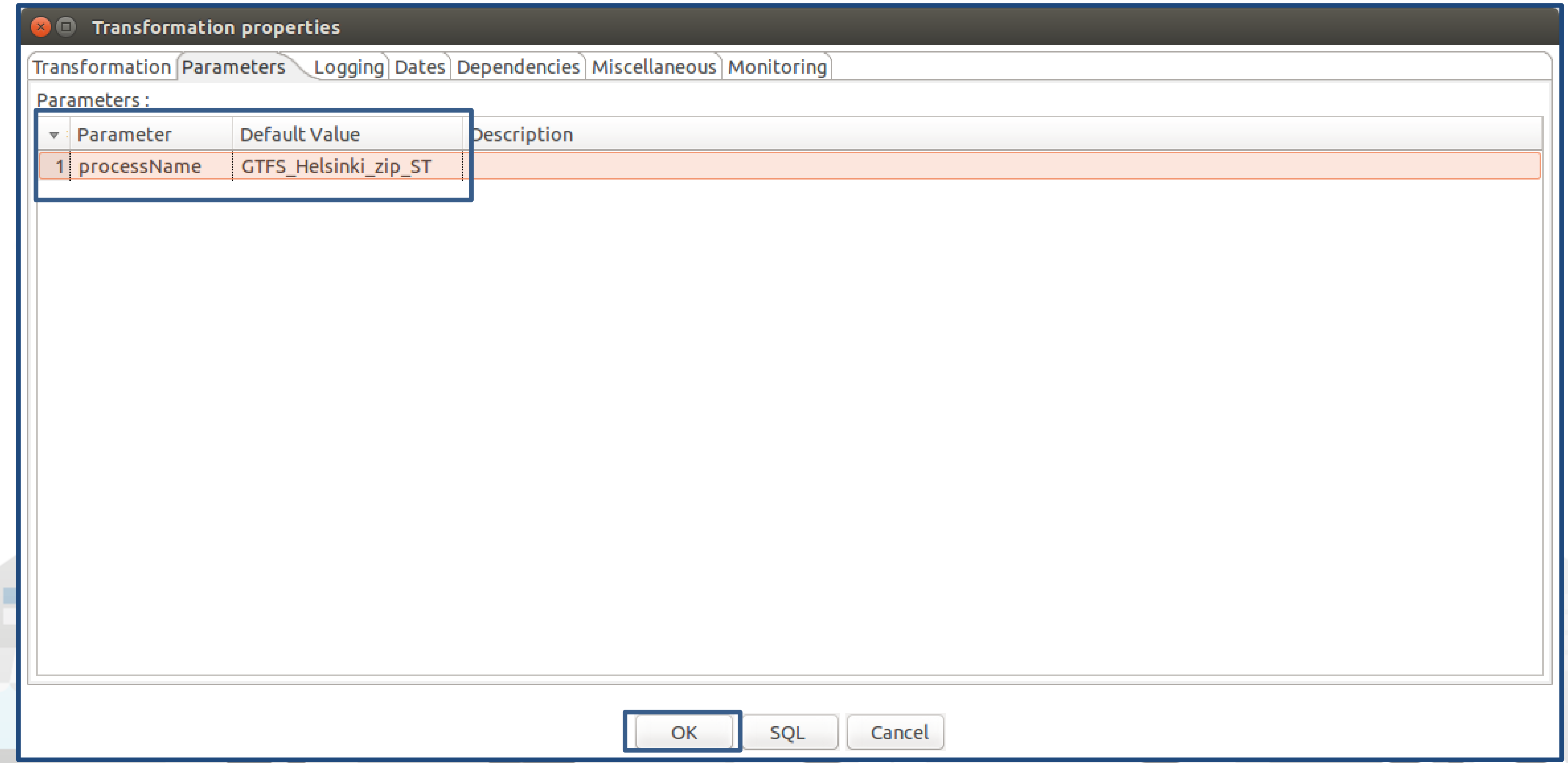
Configure Transformation (1)

- Double-click the Transformation workspace



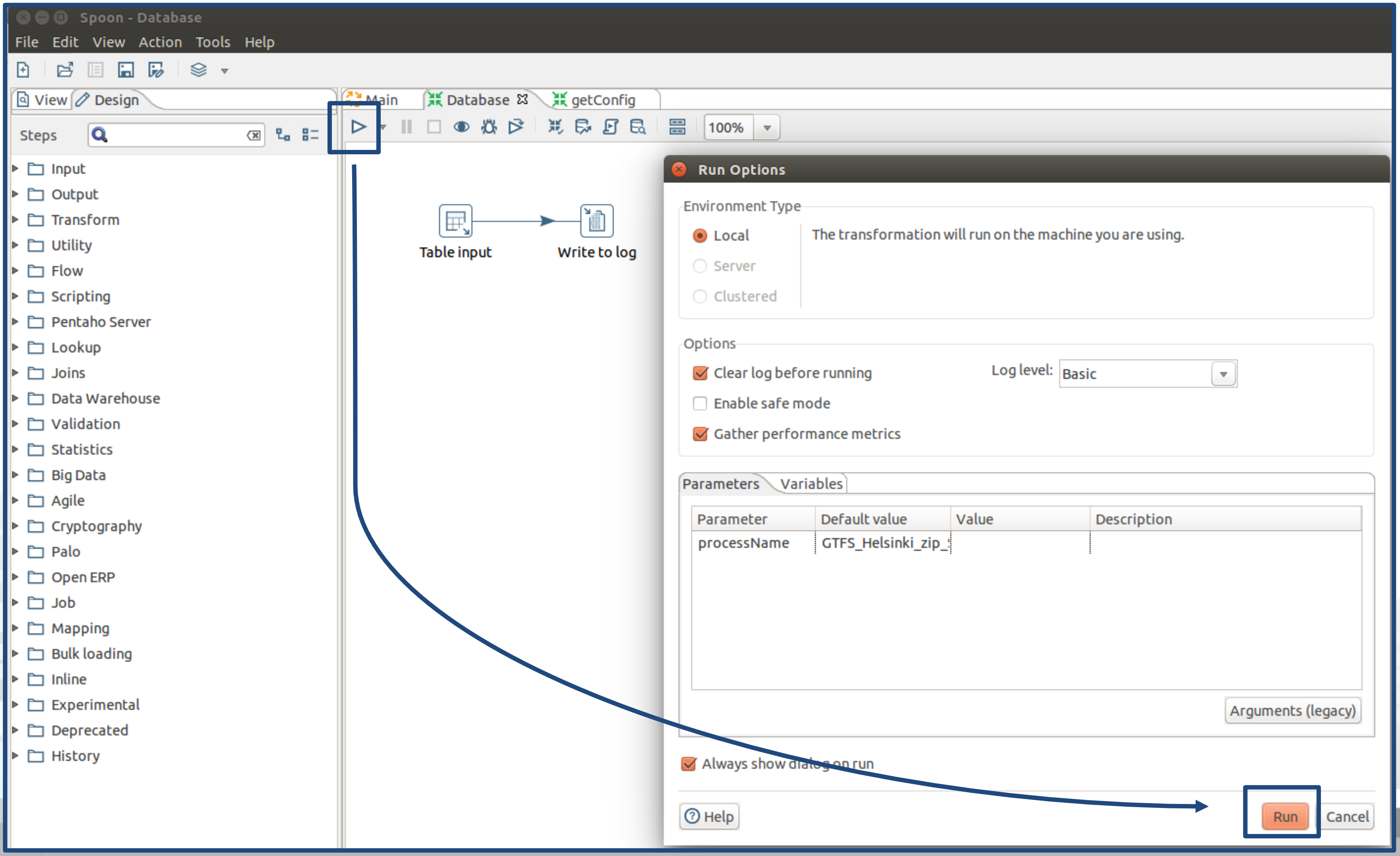
Run the Transformation (2)

- Add **processName** as an expected input parameter



	Parameter	Default Value	Description
1	processName	GTFS_Helsinki_zip_ST	

Run the Transformation



Run Options

Environment Type

- [✓ Local] The transformation will run on the machine you are using.
- Server
- Clustered

Options

- [✓] Clear log before running — Log level: Basic
- [ ] Enable safe mode
- [✓] Gather performance metrics

Parameters | Variables

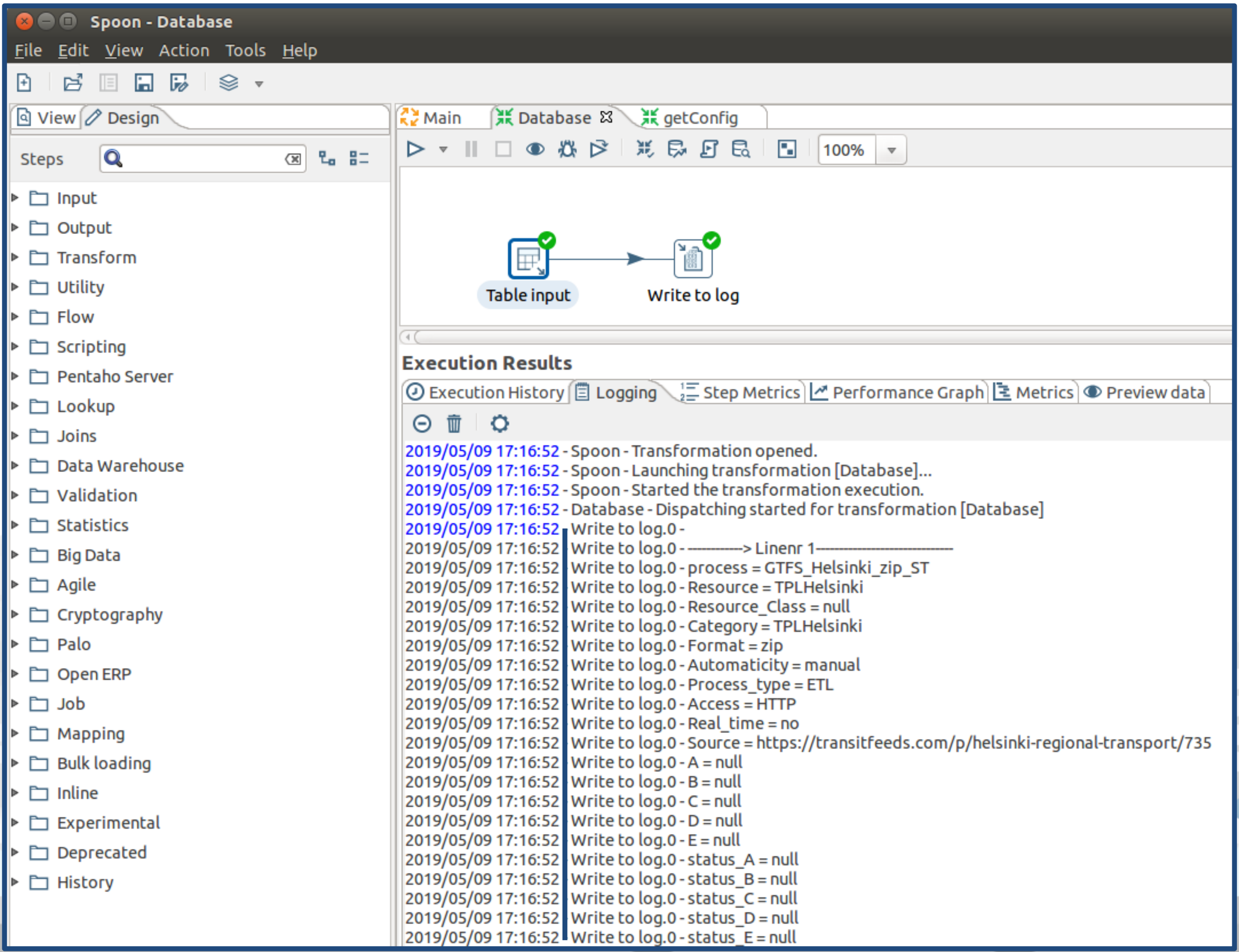
Parameter	Default value	Value	Description
processName	GTFS_Helsinki_zip_		

Arguments (legacy)

[✓] Always show dialog on run

Help — Run — Cancel

Inspect Execution Logs



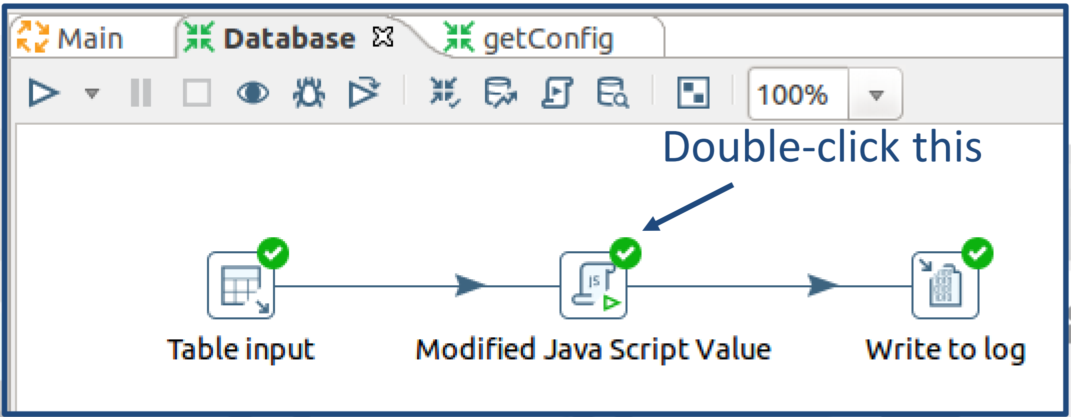
```
2019/05/09 17:16:52 - Spoon - Transformation opened.
2019/05/09 17:16:52 - Spoon - Launching transformation [Database]...
2019/05/09 17:16:52 - Spoon - Started the transformation execution.
2019/05/09 17:16:52 - Database - Dispatching started for transformation [Database]
2019/05/09 17:16:52 - Write to log.0 -
2019/05/09 17:16:52 - Write to log.0 - ------------> Linenr 1------------------------------
2019/05/09 17:16:52 - Write to log.0 - process = GTFS_Helsinki_zip_ST
2019/05/09 17:16:52 - Write to log.0 - Resource = TPLHelsinki
2019/05/09 17:16:52 - Write to log.0 - Resource_Class = null
2019/05/09 17:16:52 - Write to log.0 - Category = TPLHelsinki
2019/05/09 17:16:52 - Write to log.0 - Format = zip
2019/05/09 17:16:52 - Write to log.0 - Automaticity = manual
2019/05/09 17:16:52 - Write to log.0 - Process_type = ETL
2019/05/09 17:16:52 - Write to log.0 - Access = HTTP
2019/05/09 17:16:52 - Write to log.0 - Real_time = no
2019/05/09 17:16:52 - Write to log.0 - Source = https://transitfeeds.com/p/helsinki-regional-transport/735
2019/05/09 17:16:52 - Write to log.0 - A = null
2019/05/09 17:16:52 - Write to log.0 - B = null
2019/05/09 17:16:52 - Write to log.0 - C = null
2019/05/09 17:16:52 - Write to log.0 - D = null
2019/05/09 17:16:52 - Write to log.0 - E = null
2019/05/09 17:16:52 - Write to log.0 - status_A = null
2019/05/09 17:16:52 - Write to log.0 - status_B = null
2019/05/09 17:16:52 - Write to log.0 - status_C = null
2019/05/09 17:16:52 - Write to log.0 - status_D = null
2019/05/09 17:16:52 - Write to log.0 - status_E = null
```

Add some Javascript (1)

We are now going to add a **Modified Java Script Value** step, that belongs to the category of **Scripting** steps.

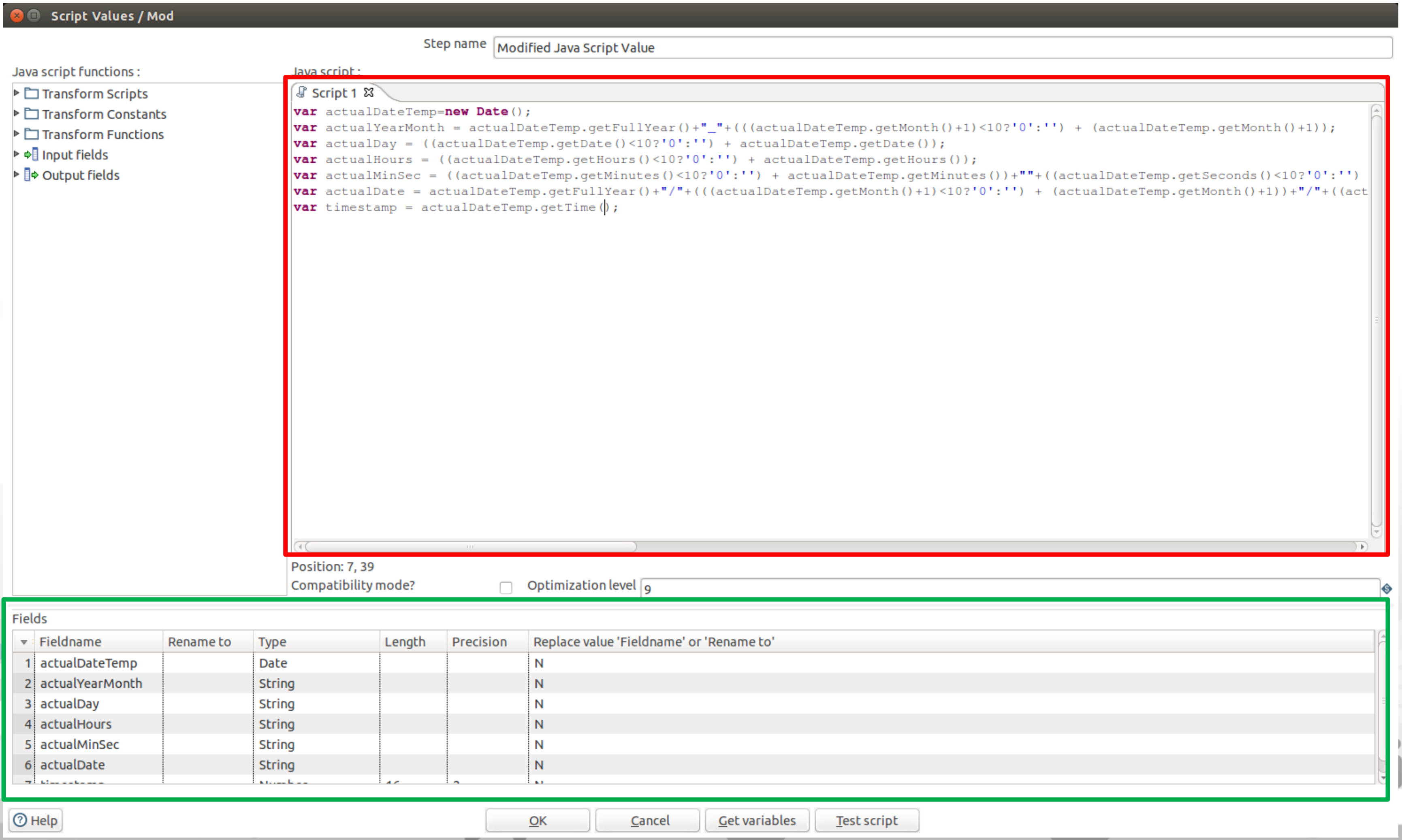
We will put it immediately after the reading from the MySQL database, with the purpose of generating some other fields (current date and time), to be sent forward together with the data that we have retrieved from the database.

After the javascript step, we will preserve the log step, so that we can run the transformation, and see the newly generated fields added to the list, each with its value.



UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org | SNAP4CITY | KM4CITY

Add some Javascript (2)



Step name: Modified Java Script Value

```
var actualDateTemp=new Date();
var actualYearMonth = actualDateTemp.getFullYear()+"_"+(((actualDateTemp.getMonth()+1)<10?'0':'') + (actualDateTemp.getMonth()+1));
var actualDay = ((actualDateTemp.getDate()<10?'0':'') + actualDateTemp.getDate());
var actualHours = ((actualDateTemp.getHours()<10?'0':'') + actualDateTemp.getHours());
var actualMinSec = ((actualDateTemp.getMinutes()<10?'0':'') + actualDateTemp.getMinutes())+""+((actualDateTemp.getSeconds()<10?'0':'')
var actualDate = actualDateTemp.getFullYear()+"/"+(((actualDateTemp.getMonth()+1)<10?'0':'') + (actualDateTemp.getMonth()+1))+"/"+((act
var timestamp = actualDateTemp.getTime();
```

Position: 7, 39
Compatibility mode? ☐ Optimization level 9

Fields

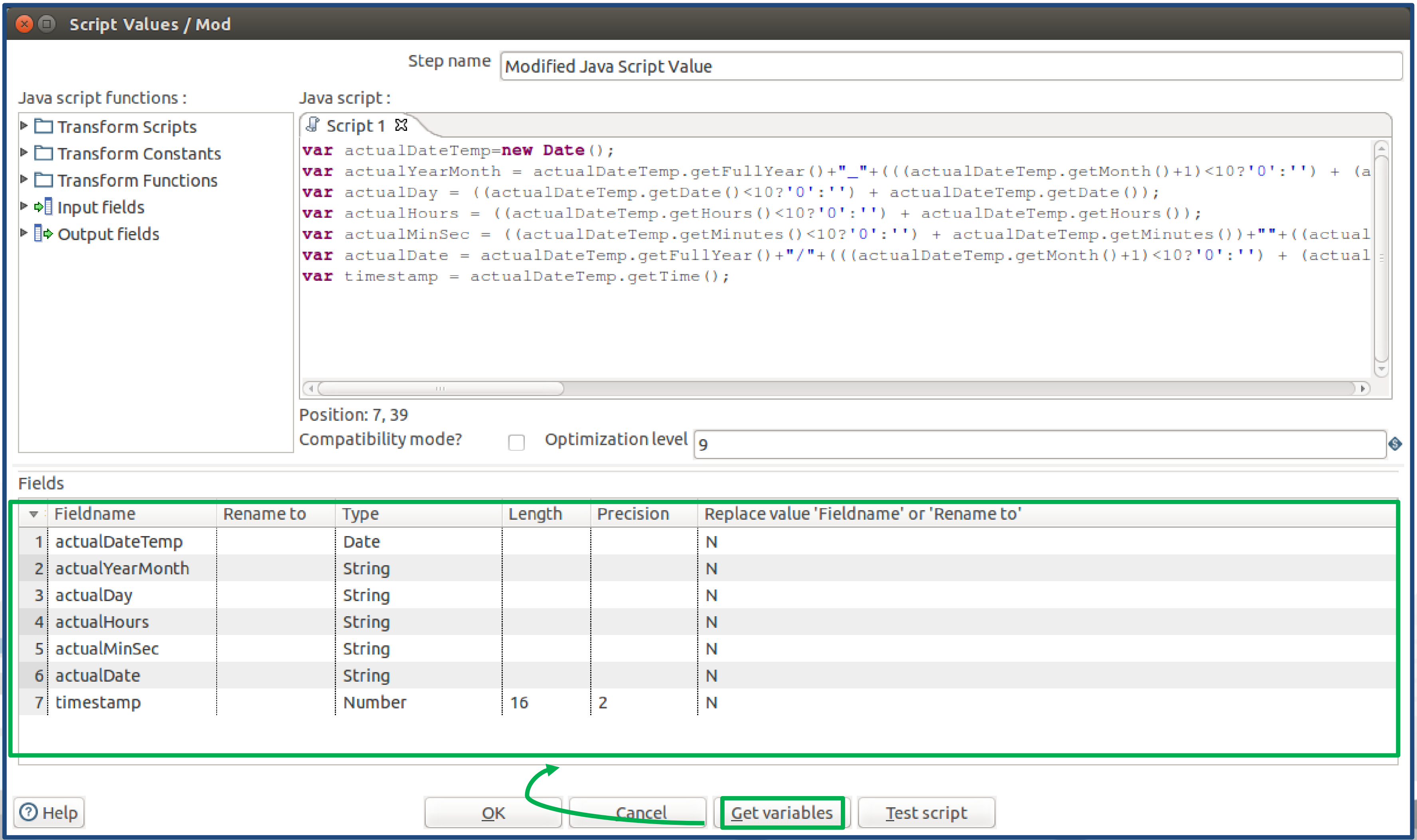
	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	actualDateTemp		Date			N
2	actualYearMonth		String			N
3	actualDay		String			N
4	actualHours		String			N
5	actualMinSec		String			N
6	actualDate		String			N

Add some Javascript (3)

```
var actualDateTemp=new Date();
var actualYearMonth =
actualDateTemp.getFullYear()+"_"+(((actualDateTemp.getMonth()+1)<10?'0':'') +
(actualDateTemp.getMonth()+1));
var actualDay = ((actualDateTemp.getDate()<10?'0':'') + actualDateTemp.getDate());
var actualHours = ((actualDateTemp.getHours()<10?'0':'') + actualDateTemp.getHours());
var actualMinSec = ((actualDateTemp.getMinutes()<10?'0':'') +
actualDateTemp.getMinutes())+""+((actualDateTemp.getSeconds()<10?'0':'') +
actualDateTemp.getSeconds());
var actualDate = actualDateTemp.getFullYear()+"/"+(((actualDateTemp.getMonth()+1)<10?'0':'') +
(actualDateTemp.getMonth()+1))+"/"+((actualDateTemp.getDate()<10?'0':'') +
actualDateTemp.getDate())+" "+((actualDateTemp.getHours()<10?'0':'') +
actualDateTemp.getHours())+":"+((actualDateTemp.getMinutes()<10?'0':'') +
actualDateTemp.getMinutes())+":"+((actualDateTemp.getSeconds()<10?'0':'') +
actualDateTemp.getSeconds()+".000");
var timestamp = actualDateTemp.getTime();
```

http://www.disit.org

Add some Javascript (4)



Step name: Modified Java Script Value

Java script :

```
var actualDateTemp=new Date();
var actualYearMonth = actualDateTemp.getFullYear()+"_"+(((actualDateTemp.getMonth()+1)<10?'0':'') + (a
var actualDay = ((actualDateTemp.getDate()<10?'0':'') + actualDateTemp.getDate());
var actualHours = ((actualDateTemp.getHours()<10?'0':'') + actualDateTemp.getHours());
var actualMinSec = ((actualDateTemp.getMinutes()<10?'0':'') + actualDateTemp.getMinutes())+""+((actual
var actualDate = actualDateTemp.getFullYear()+"/"+(((actualDateTemp.getMonth()+1)<10?'0':'') + (actual
var timestamp = actualDateTemp.getTime();
```

Position: 7, 39

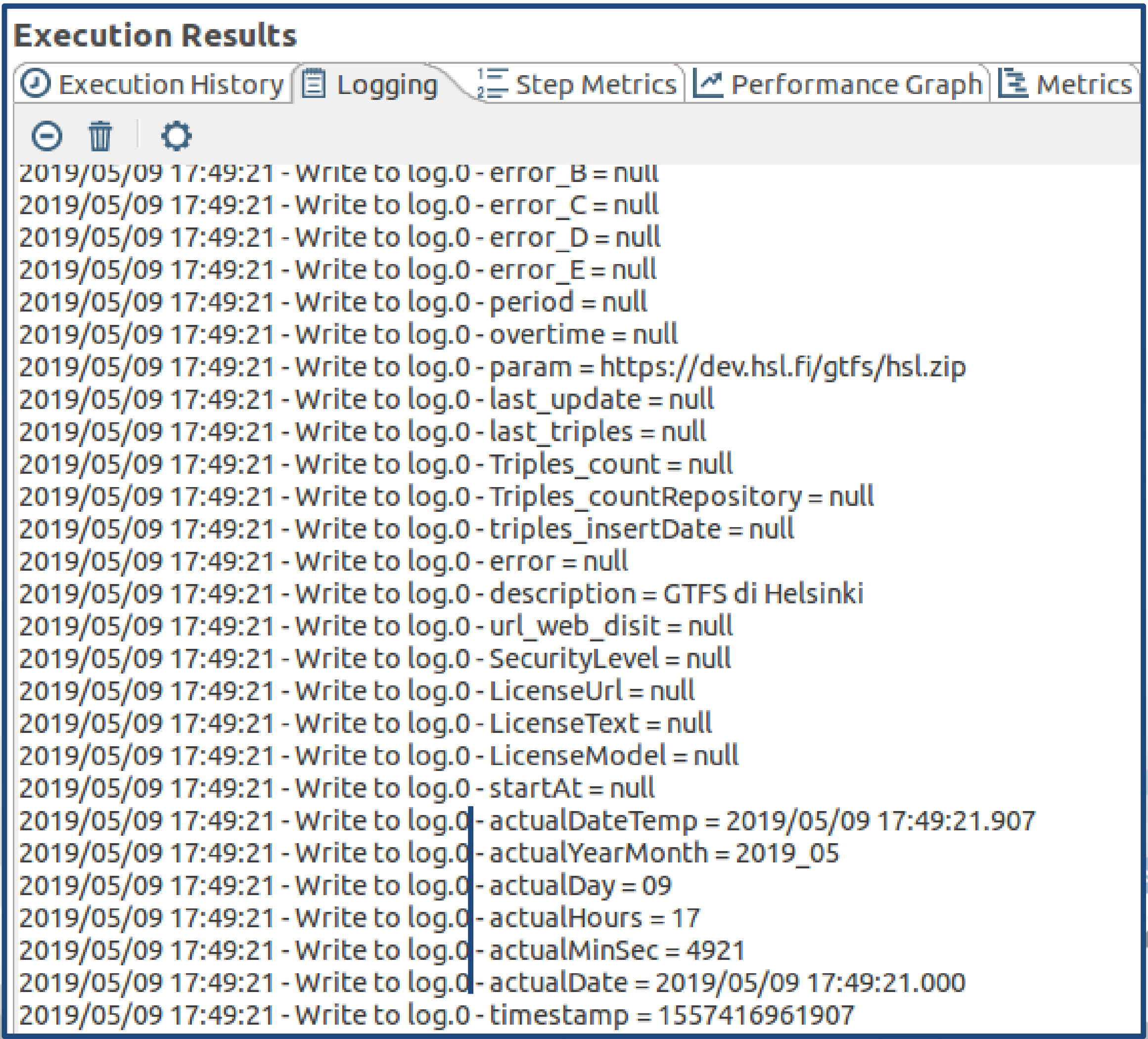
Compatibility mode? ☐ Optimization level 9

Fields

	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	actualDateTemp		Date			N
2	actualYearMonth		String			N
3	actualDay		String			N
4	actualHours		String			N
5	actualMinSec		String			N
6	actualDate		String			N
7	timestamp		Number	16	2	N

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org | SNAP4CITY | KM4CITY

Run & Inspect Execution Logs

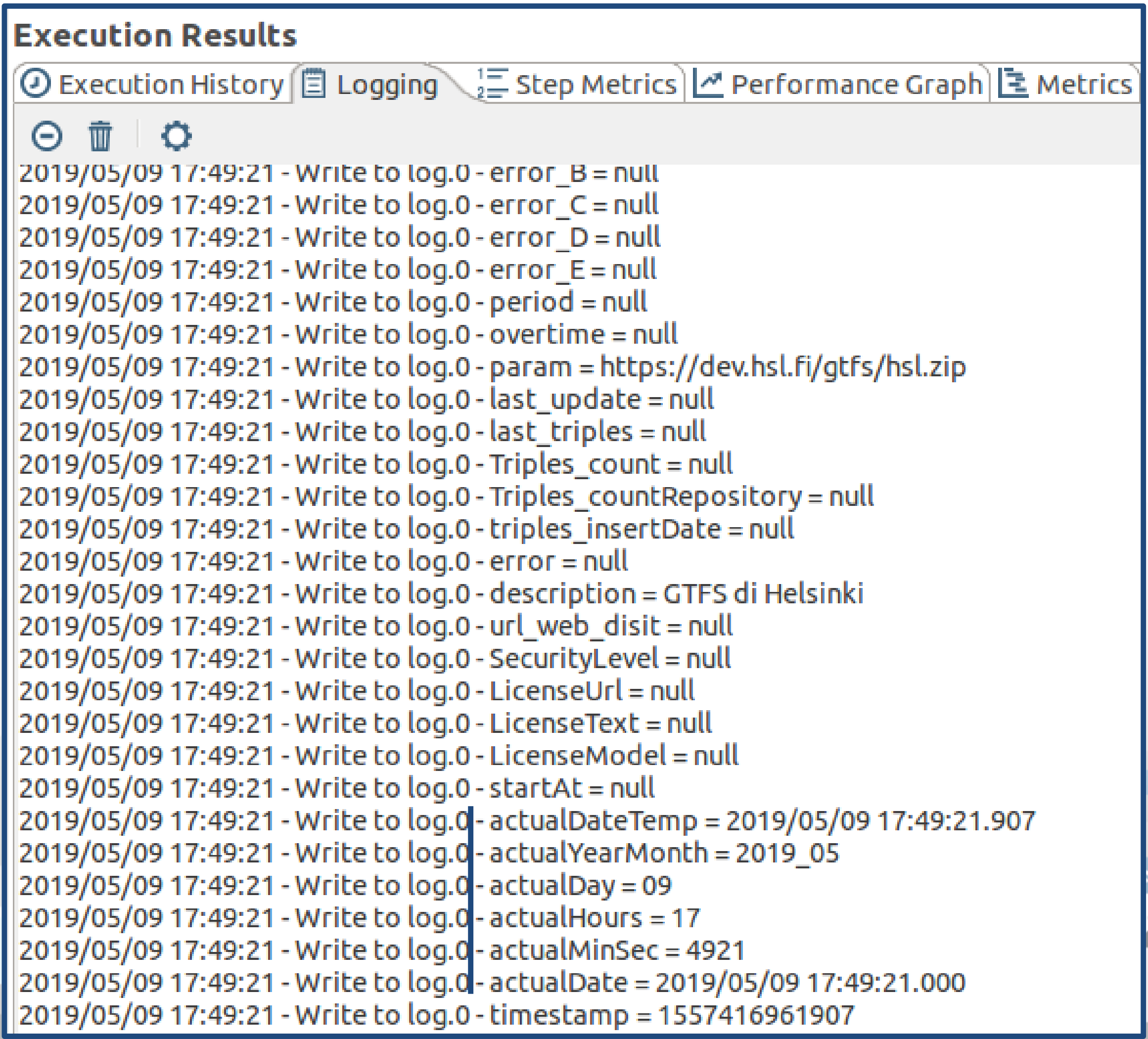


```
2019/05/09 17:49:21 - Write to log.0 - error_B = null
2019/05/09 17:49:21 - Write to log.0 - error_C = null
2019/05/09 17:49:21 - Write to log.0 - error_D = null
2019/05/09 17:49:21 - Write to log.0 - error_E = null
2019/05/09 17:49:21 - Write to log.0 - period = null
2019/05/09 17:49:21 - Write to log.0 - overtime = null
2019/05/09 17:49:21 - Write to log.0 - param = https://dev.hsl.fi/gtfs/hsl.zip
2019/05/09 17:49:21 - Write to log.0 - last_update = null
2019/05/09 17:49:21 - Write to log.0 - last_triples = null
2019/05/09 17:49:21 - Write to log.0 - Triples_count = null
2019/05/09 17:49:21 - Write to log.0 - Triples_countRepository = null
2019/05/09 17:49:21 - Write to log.0 - triples_insertDate = null
2019/05/09 17:49:21 - Write to log.0 - error = null
2019/05/09 17:49:21 - Write to log.0 - description = GTFS di Helsinki
2019/05/09 17:49:21 - Write to log.0 - url_web_disit = null
2019/05/09 17:49:21 - Write to log.0 - SecurityLevel = null
2019/05/09 17:49:21 - Write to log.0 - LicenseUrl = null
2019/05/09 17:49:21 - Write to log.0 - LicenseText = null
2019/05/09 17:49:21 - Write to log.0 - LicenseModel = null
2019/05/09 17:49:21 - Write to log.0 - startAt = null
2019/05/09 17:49:21 - Write to log.0 - actualDateTemp = 2019/05/09 17:49:21.907
2019/05/09 17:49:21 - Write to log.0 - actualYearMonth = 2019_05
2019/05/09 17:49:21 - Write to log.0 - actualDay = 09
2019/05/09 17:49:21 - Write to log.0 - actualHours = 17
2019/05/09 17:49:21 - Write to log.0 - actualMinSec = 4921
2019/05/09 17:49:21 - Write to log.0 - actualDate = 2019/05/09 17:49:21.000
2019/05/09 17:49:21 - Write to log.0 - timestamp = 1557416961907
```

Remark

Fields that you produce in a Javascript step are **added** to those that you already had in input.

Run & Inspect Execution Logs



```
2019/05/09 17:49:21 - Write to log.0 - error_B = null
2019/05/09 17:49:21 - Write to log.0 - error_C = null
2019/05/09 17:49:21 - Write to log.0 - error_D = null
2019/05/09 17:49:21 - Write to log.0 - error_E = null
2019/05/09 17:49:21 - Write to log.0 - period = null
2019/05/09 17:49:21 - Write to log.0 - overtime = null
2019/05/09 17:49:21 - Write to log.0 - param = https://dev.hsl.fi/gtfs/hsl.zip
2019/05/09 17:49:21 - Write to log.0 - last_update = null
2019/05/09 17:49:21 - Write to log.0 - last_triples = null
2019/05/09 17:49:21 - Write to log.0 - Triples_count = null
2019/05/09 17:49:21 - Write to log.0 - Triples_countRepository = null
2019/05/09 17:49:21 - Write to log.0 - triples_insertDate = null
2019/05/09 17:49:21 - Write to log.0 - error = null
2019/05/09 17:49:21 - Write to log.0 - description = GTFS di Helsinki
2019/05/09 17:49:21 - Write to log.0 - url_web_disit = null
2019/05/09 17:49:21 - Write to log.0 - SecurityLevel = null
2019/05/09 17:49:21 - Write to log.0 - LicenseUrl = null
2019/05/09 17:49:21 - Write to log.0 - LicenseText = null
2019/05/09 17:49:21 - Write to log.0 - LicenseModel = null
2019/05/09 17:49:21 - Write to log.0 - startAt = null
2019/05/09 17:49:21 - Write to log.0 - actualDateTemp = 2019/05/09 17:49:21.907
2019/05/09 17:49:21 - Write to log.0 - actualYearMonth = 2019_05
2019/05/09 17:49:21 - Write to log.0 - actualDay = 09
2019/05/09 17:49:21 - Write to log.0 - actualHours = 17
2019/05/09 17:49:21 - Write to log.0 - actualMinSec = 4921
2019/05/09 17:49:21 - Write to log.0 - actualDate = 2019/05/09 17:49:21.000
2019/05/09 17:49:21 - Write to log.0 - timestamp = 1557416961907
```

Remark

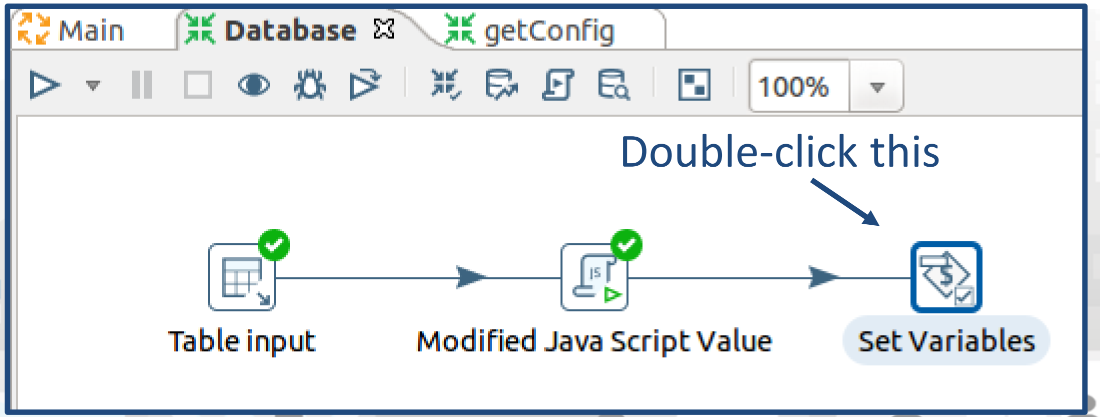
Fields that you produce in a Javascript step are **added** to those that you already had in input.

Set some variables (1)

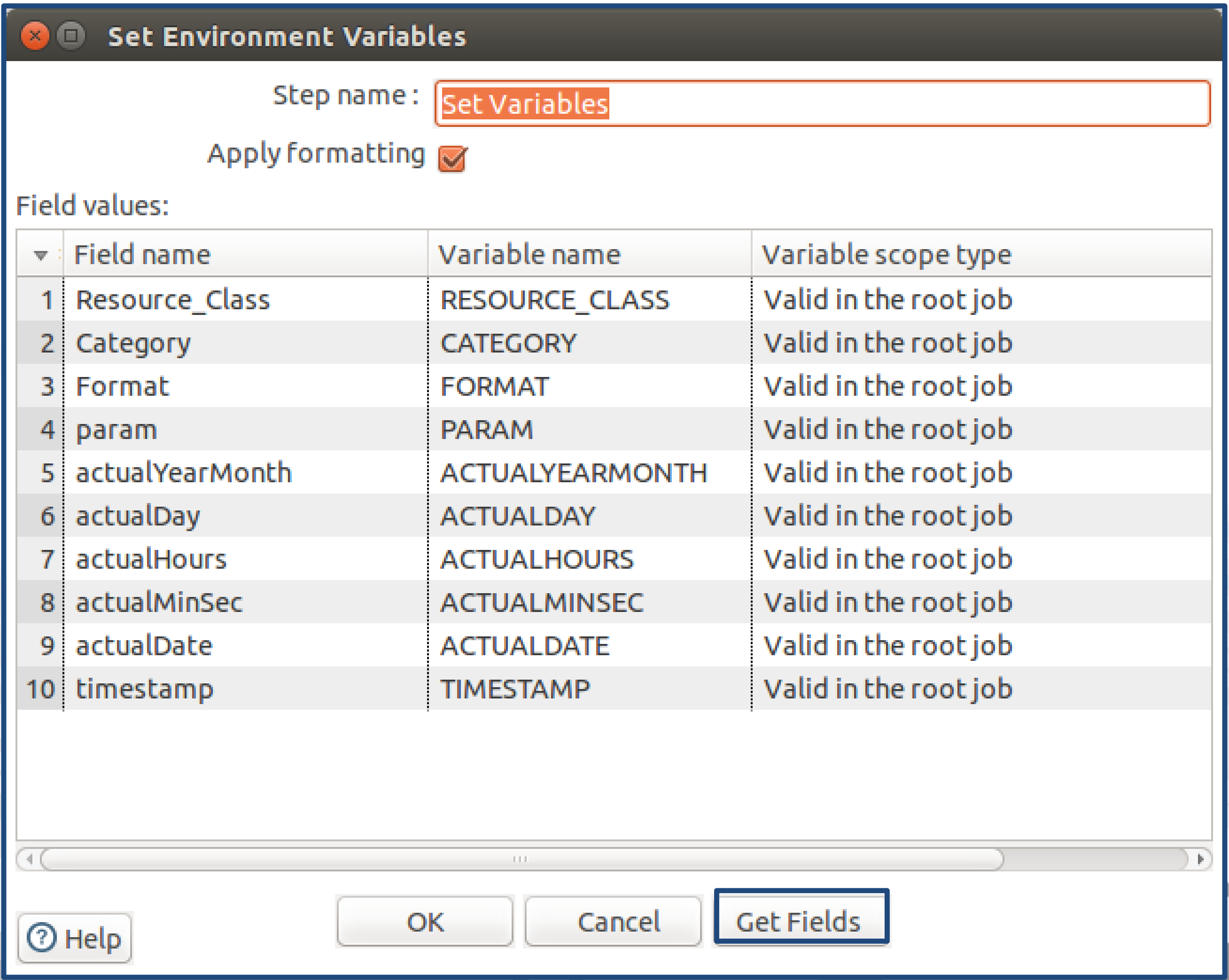
We are now going to use the data that we have retrieved from the database, and those that we have added through the Javascript, to set some variables that we will then access in the next along our ETL process.

We will do so, using a **Set Variables** step, that belongs to the category of **Job** steps.

We will put it immediately after the Javascript, in place of the **Write to log**.



Set some variables (2)



Step name : Set Variables

Apply formatting ✓

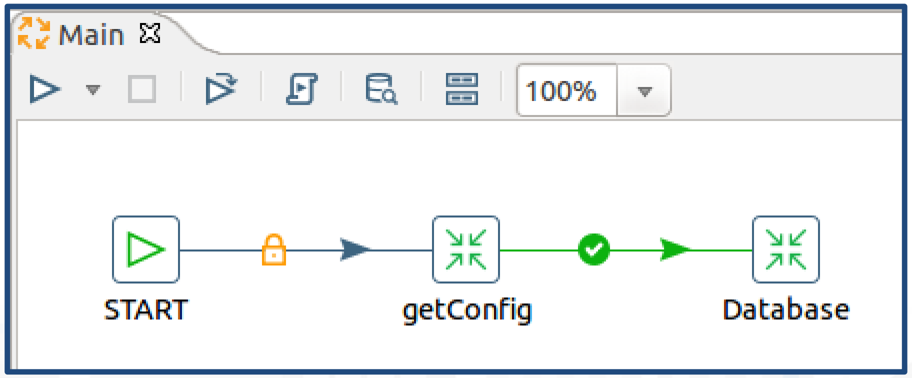
Field values:

	Field name	Variable name	Variable scope type
1	Resource_Class	RESOURCE_CLASS	Valid in the root job
2	Category	CATEGORY	Valid in the root job
3	Format	FORMAT	Valid in the root job
4	param	PARAM	Valid in the root job
5	actualYearMonth	ACTUALYEARMONTH	Valid in the root job
6	actualDay	ACTUALDAY	Valid in the root job
7	actualHours	ACTUALHOURS	Valid in the root job
8	actualMinSec	ACTUALMINSEC	Valid in the root job
9	actualDate	ACTUALDATE	Valid in the root job
10	timestamp	TIMESTAMP	Valid in the root job

Hit **Get Fields** in first, then select rows of unwanted fields and press **Canc**, obtaining the list that you can see in the picture.

Add transformation to job

Database transformation is complete. We now need to append it to the **Main** job.



Create Folders (1)

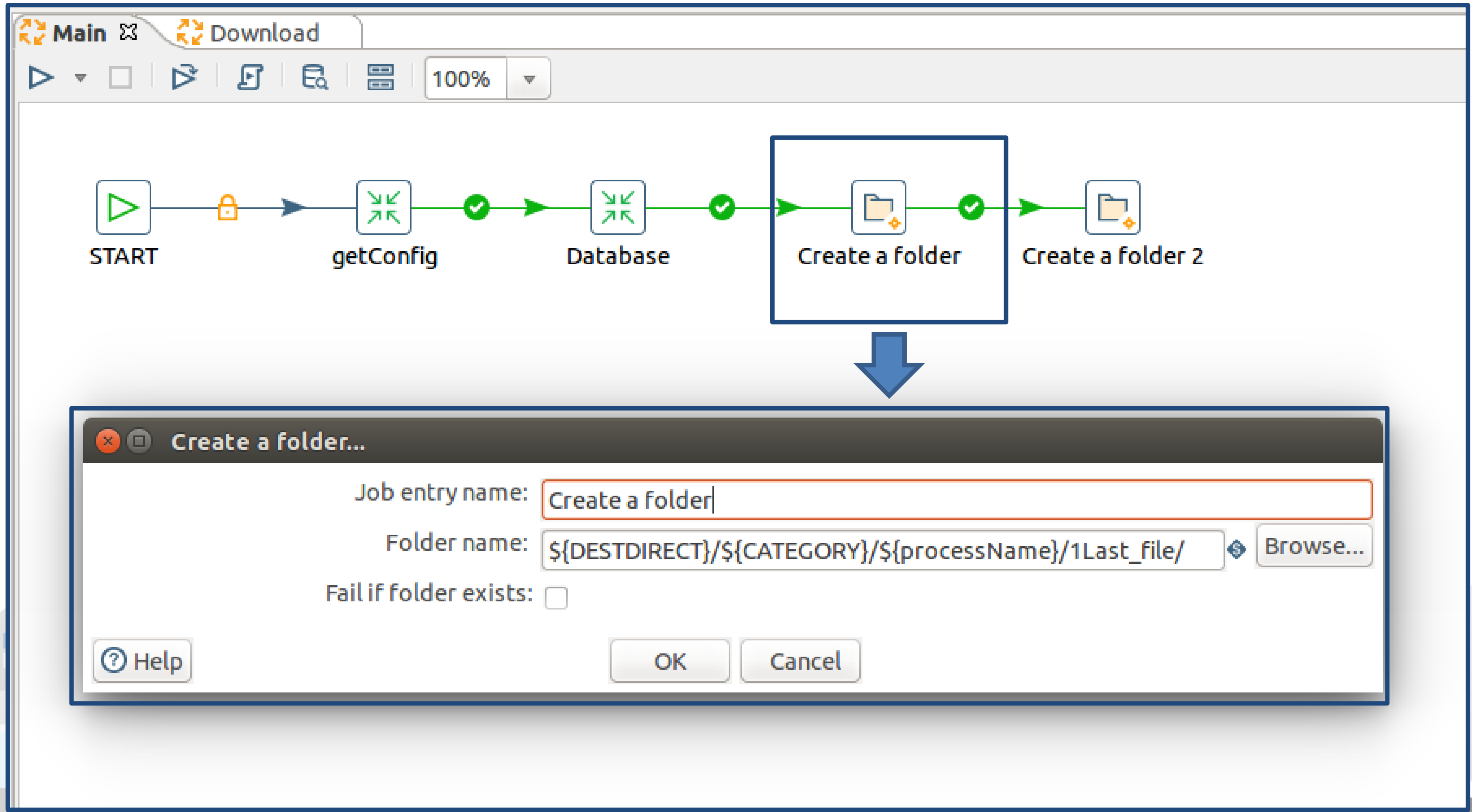
We now have to create two new folders:

- **${DESTDIRECT}/${CATEGORY}/${processName}**/1Last_file/
- **${DESTDIRECT}/${CATEGORY}/${processName}/${ACTUALYEARMONTH}/${ACTUALDAY}/${ACTUALHOURS}/${ACTUALMINSEC}**

Note the syntax **${*VARIABLE_NAME*}** to include values of variables and parameters that we have set earlier in our ETL process. ETL must not fail if folder already exists.

The step to be used is the **Create a folder**, that belongs to the category of **File management** steps. We append two of them at the end of the **Main** job.

Create Folders (2)



Job entry name: Create a folder

Folder name: ${DESTDIRECT}/${CATEGORY}/${processName}/1Last_file/

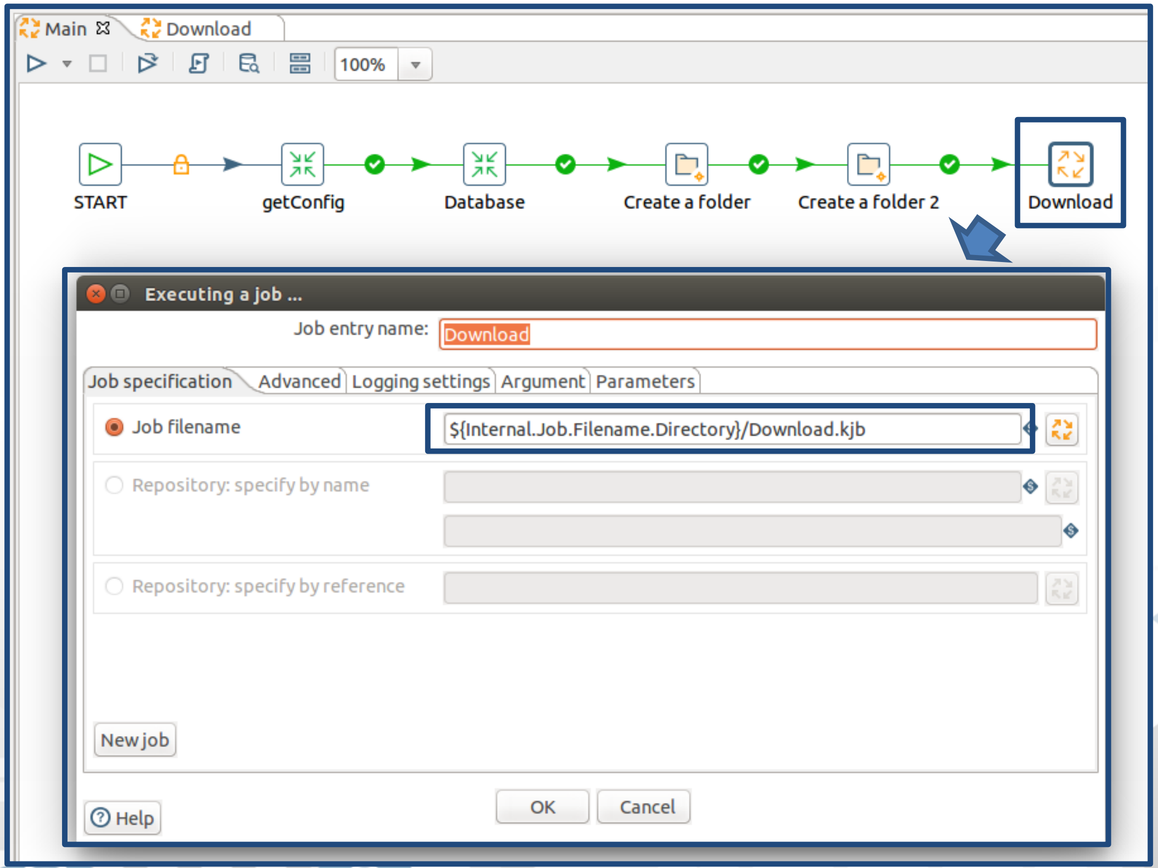
Fail if folder exists: ☐

Create the Download Job (1)

We now have to create a job to download the source GTFS file from the Internet, verify that it is not something that we already have processed, and decompress it to an appropriate folder, so:

- Create a new empty job, and save it with name **Download.kjb**, in the same folder where all of the other Spoon artifacts locate
  - File → New → Job
- Append an empty **Job** step at the end of the **Main.kjb** job. The Job step belongs to the General category.
- Configure the **Job** step that you have appended to the **Main.kjb** job, so that it points to the newly created **Download.kjb** job
  - It is very similar to what we have seen when configuring the **Transformation** step to add the **getConfig.ktr** transformation to our **Main.kjb** job.

Create the Download Job (2)

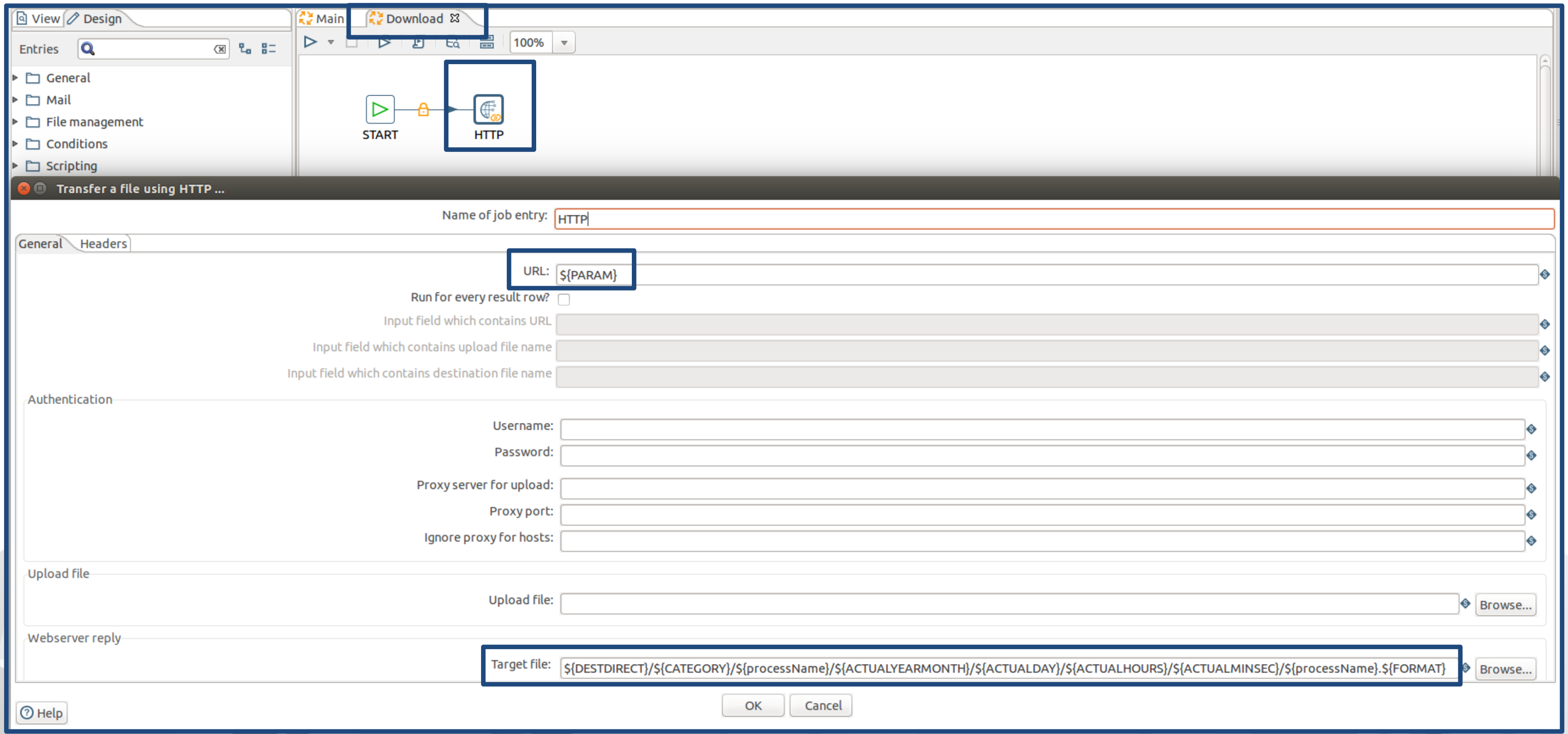


Job entry name: Download

Job filename: ${Internal.Job.Filename.Directory}/Download.kjb

Add the HTTP step

We now add the **HTTP** step from the **File management** category, and configure it as below.



Name of job entry: HTTP

URL: ${PARAM}

Run for every result row? ☐

Input field which contains URL

Input field which contains upload file name

Input field which contains destination file name

Authentication

Username:

Password:

Proxy server for upload:

Proxy port:

Ignore proxy for hosts:

Upload file

Upload file:

Webserver reply

Target file: ${DESTDIRECT}/${CATEGORY}/${processName}/${ACTUALYEARMONTH}/${ACTUALDAY}/${ACTUALHOURS}/${ACTUALMINSEC}/${processName}.${FORMAT}

Verify the HTTP step

For a number of reasons, the HTTP connection could fail, or the download could not to complete correctly anyway.

To verify if everything is ok, run the Main job, that implies running the Download job, and then verify if the zip file is present where expected, for example in:

/home/ubuntu/Desktop/Sources/TPLHelsinki/GTFS_Helsinki_zip_ST/2019_05/10/14/5756

Before running the Main job, add the **processName** as an expected input parameter of the job. The procedure is very similar to that seen for the **Database** transformation.

Verify & Copy the source file (1)

Once a new source file is downloaded, we have to verify if it is ***of actual interest***, that is:

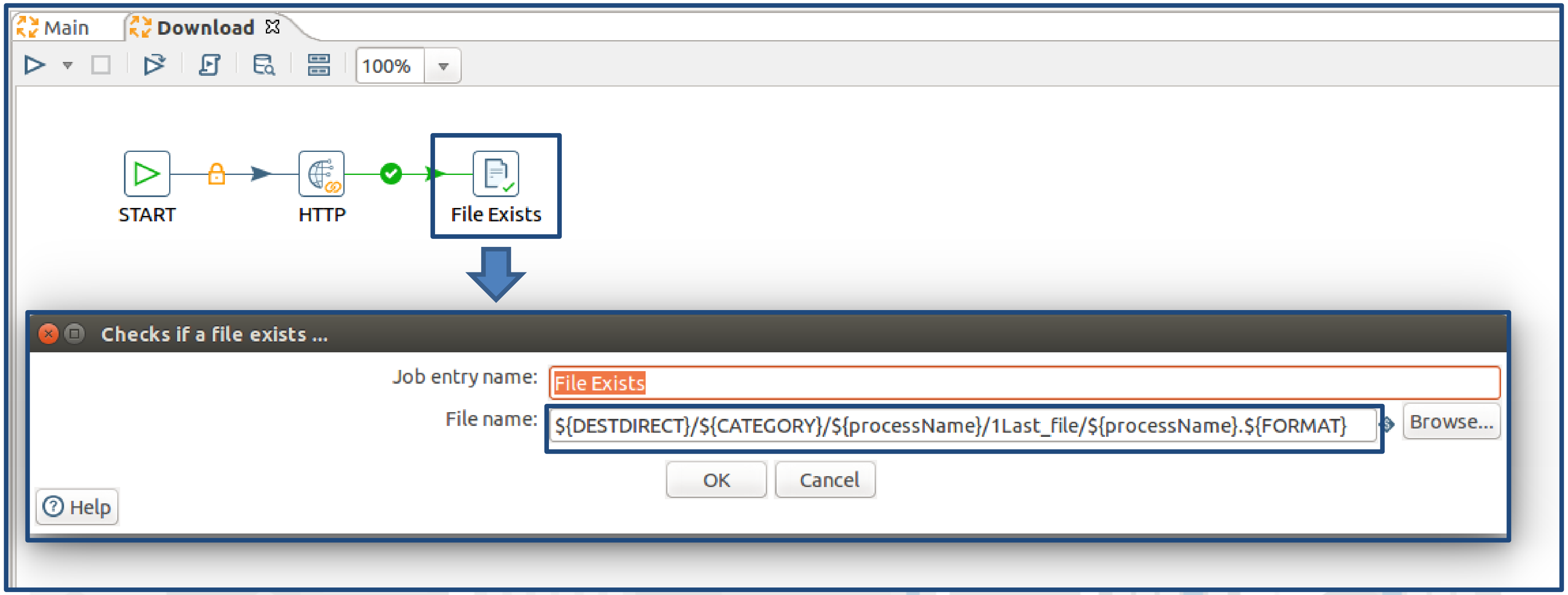
1. If the folder is empty, it means that the ETL is running for the first time, and therefore the newly downloaded source file is of interest for sure;
2. If the folder is not empty, a further check is needed to verify if the newly downloaded file is different from that already locating in the folder, in which case it is of interest.

If the newly downloaded file is of interest, it must be copied from its original position (e.g. /home/ubuntu/Desktop/Sources/TPLHelsinki/GTFS_Helsinki_zip_ST/2019_05/10/14/5756) into the 1Last_file folder, and processed. Otherwise, the ETL terminates.

For implementing this logic, we need:

- A **File Exists** step, in the **Condition** category, to check if a source file already exists in the 1Last_file folder or not;
- A **File Compare** step, in the **File management** category, to check if the newly downloaded source file is exactly the same of the one that already locates in the 1Last_file folder
- A **Copy Files** step, in the **File management** category, to copy the file into the 1Last_file folder, in those cases when it is necessary.

Verify & Copy the source file (2)



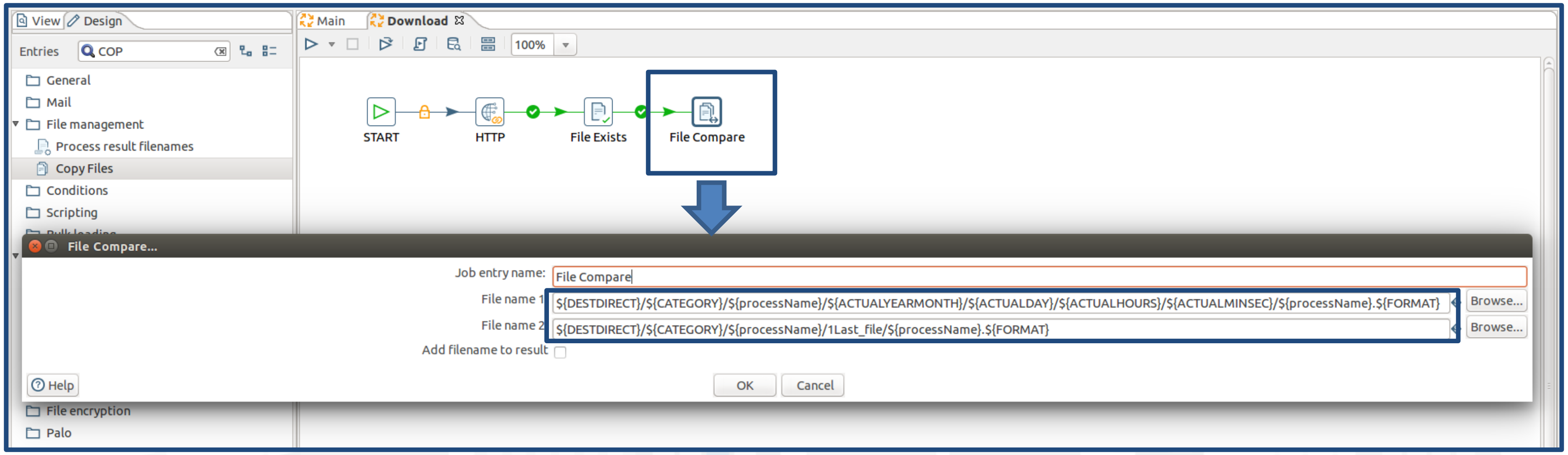
Job entry name: File Exists

File name: ${DESTDIRECT}/${CATEGORY}/${processName}/1Last_file/${processName}.${FORMAT}

The **File Exists** step returns an error if the file does not exist. We will manage the thing this way:

- If the step exit status is **OK**, we will go forward and check if the new and the old file are identical;
- If the step exit status is **ERROR**, we will copy the newly downloaded file into the 1Last_file folder

Verify & Copy the source file (3)



Job entry name: File Compare

File name 1: ${DESTDIRECT}/${CATEGORY}/${processName}/${ACTUALYEARMONTH}/${ACTUALDAY}/${ACTUALHOURS}/${ACTUALMINSEC}/${processName}.${FORMAT}

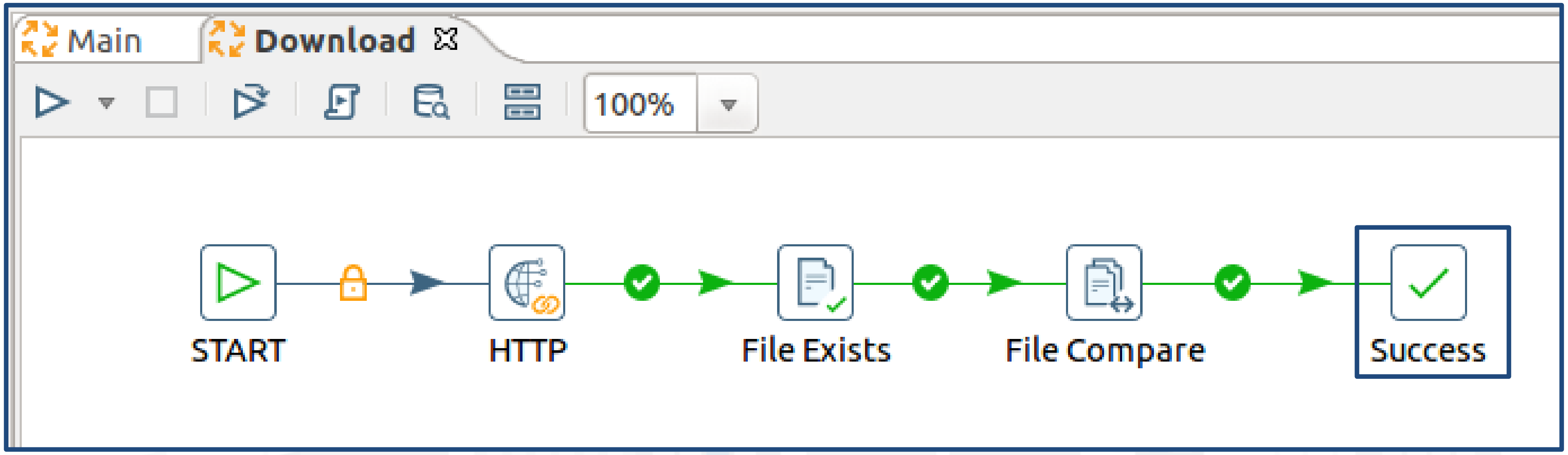
File name 2: ${DESTDIRECT}/${CATEGORY}/${processName}/1Last_file/${processName}.${FORMAT}

Add filename to result: ☐

The **File Compare** step returns an error if files are not identical. We will manage the thing this way:

- If the step exit status is **OK**, we will go forward and terminate without doing anything;
- If the step exit status is **ERROR**, we will copy the newly downloaded file into the 1Last_file folder.

Verify & Copy the source file (4)

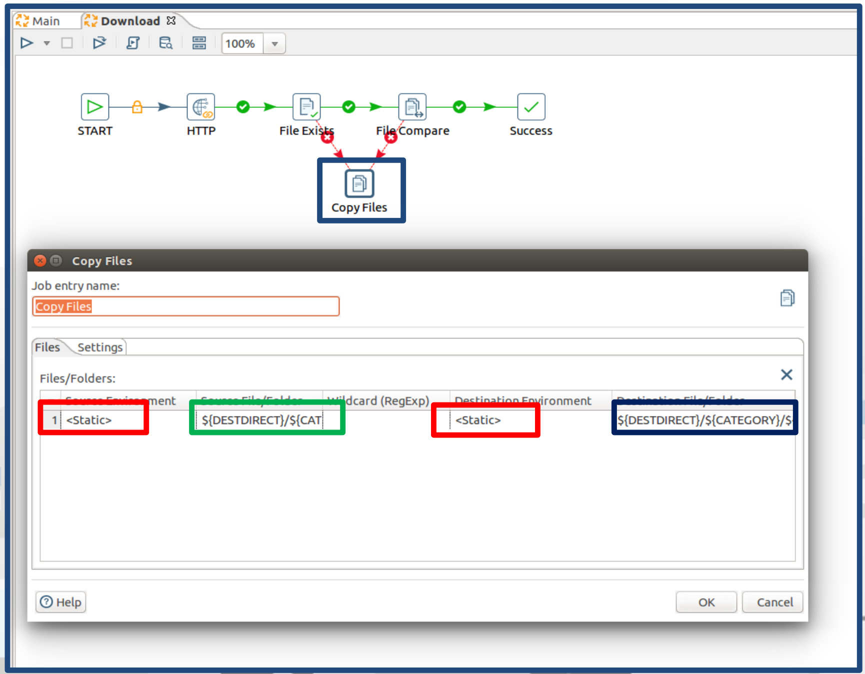


Verify & Copy the source file (5)

Set **Source Environment** and **Destination Environment** to **Static** since files do not locate on the same host where Spoon jobs run.
Indeed, all paths that can be found in the global configuration file **config.csv** start with **/media**

Set **Source File/Folder** to:
**${DESTDIRECT}/${CATEGORY}/${processName}/${ACTUALYEARMONTH}/${ACTUALDAY}/${ACTUALHOURS}/${ACTUALMINSEC}**/${processName}.${FORMAT}

Set **Destination File/Folder** to:
**${DESTDIRECT}/${CATEGORY}/${processName}**/1Last_file/${processName}.${FORMAT}



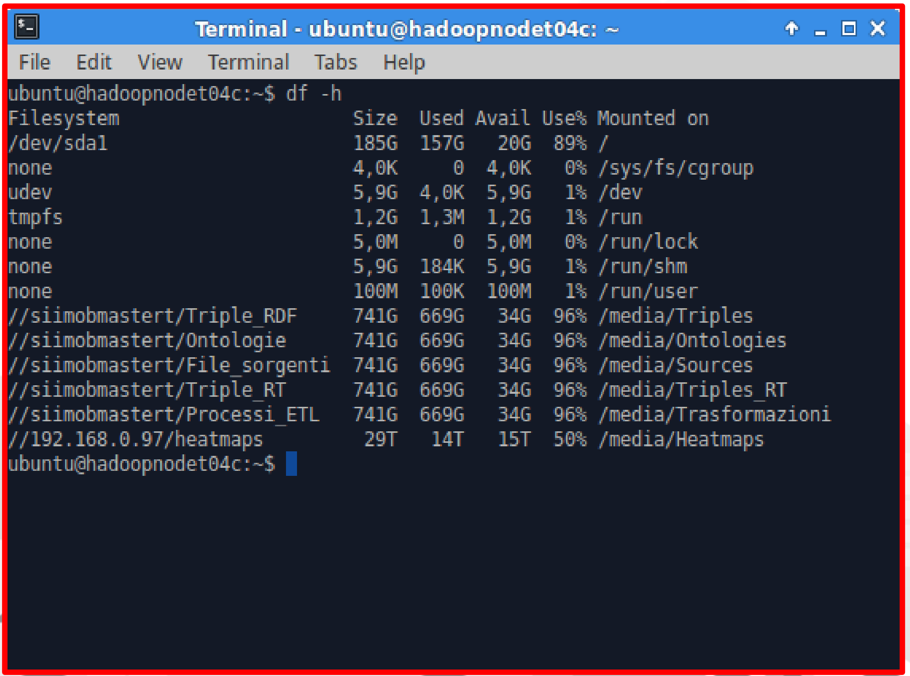
Copy Files

Job entry name: Copy Files

Files | Settings

Files/Folders:

	Source Environment	Source File/Folder	Wildcard (RegExp)	Destination Environment	Destination File/Folder
1	<Static>	${DESTDIRECT}/${CAT		<Static>	${DESTDIRECT}/${CATEGORY}/$


Terminal - ubuntu@hadoopnodet04c: ~
File Edit View Terminal Tabs Help

```
ubuntu@hadoopnodet04c:~$ df -h
Filesystem                     Size  Used Avail Use% Mounted on
/dev/sda1                      185G  157G   20G  89% /
none                           4,0K     0  4,0K   0% /sys/fs/cgroup
udev                           5,9G  4,0K  5,9G   1% /dev
tmpfs                          1,2G  1,3M  1,2G   1% /run
none                           5,0M     0  5,0M   0% /run/lock
none                           5,9G  184K  5,9G   1% /run/shm
none                           100M  100K  100M   1% /run/user
//siimobmastert/Triple_RDF     741G  669G   34G  96% /media/Triples
//siimobmastert/Ontologie      741G  669G   34G  96% /media/Ontologies
//siimobmastert/File_sorgenti  741G  669G   34G  96% /media/Sources
//siimobmastert/Triple_RT      741G  669G   34G  96% /media/Triples_RT
//siimobmastert/Processi_ETL   741G  669G   34G  96% /media/Trasformazioni
//192.168.0.97/heatmaps         29T   14T   15T  50% /media/Heatmaps
ubuntu@hadoopnodet04c:~$
```

Verify if source file is a ZIP (1)

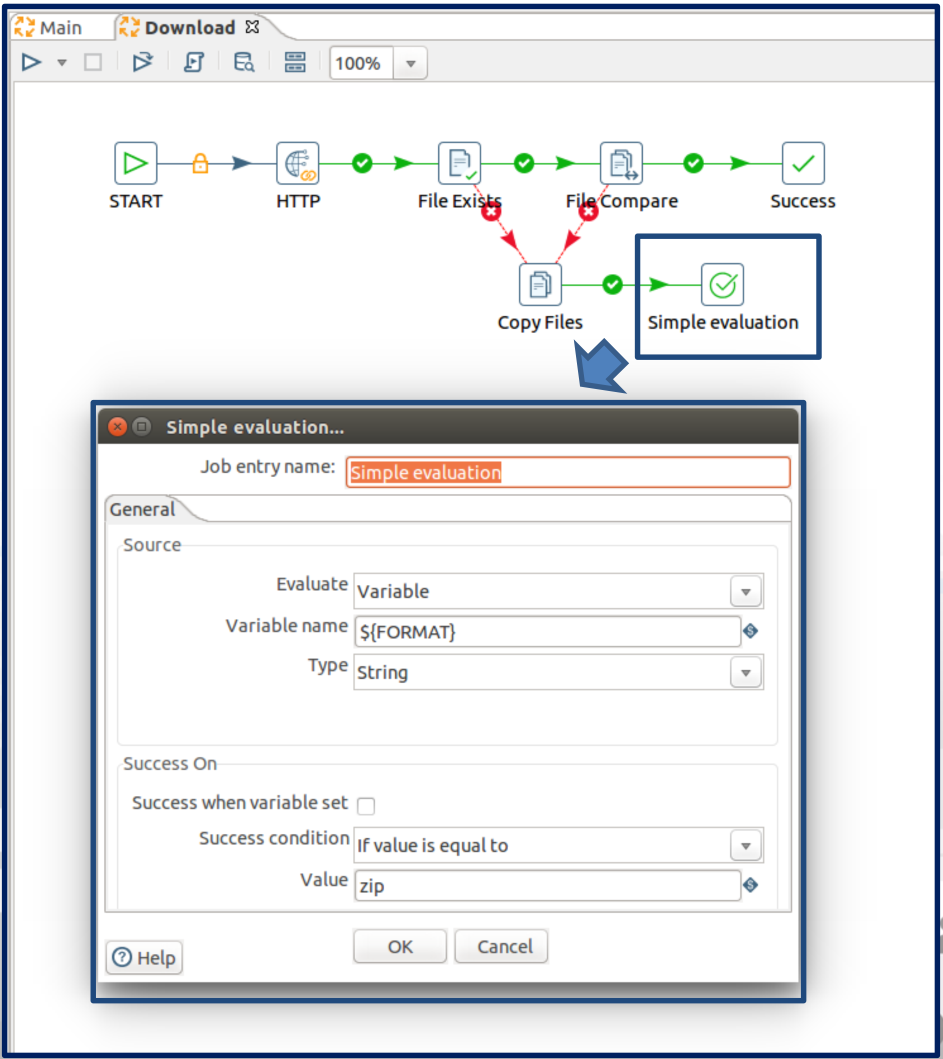
We need to verify if the source file that we have considered to be of interest and copied into the 1Last_file folder is an **archive**. Indeed, in this case an extra step is required, that consists in exploding the archive into a subfolder of the 1Last_file folder, that will have the same name of the archive file.

For the truth, since we are getting a GTFS, we know for sure that our source file is an archive, but adding the check allows us to eventually reuse this same job for downloading other source files that also could not to be archive files.

The check is simply performed looking at the **FORMAT**, that is one of the metadata associated to each ETL process and stored in the MySQL database (one of those data that we have retrieved through the **Database** transformation. The value of the FORMAT metadata simply is the expected extension of the source file of the ETL process.

The check is performed through the **Simple Evaluation** step, that belongs to the **Conditions** category.

Verify if source file is a ZIP (2)



Simple evaluation...

Job entry name: Simple evaluation

General

Source

Evaluate: Variable

Variable name: ${FORMAT}

Type: String

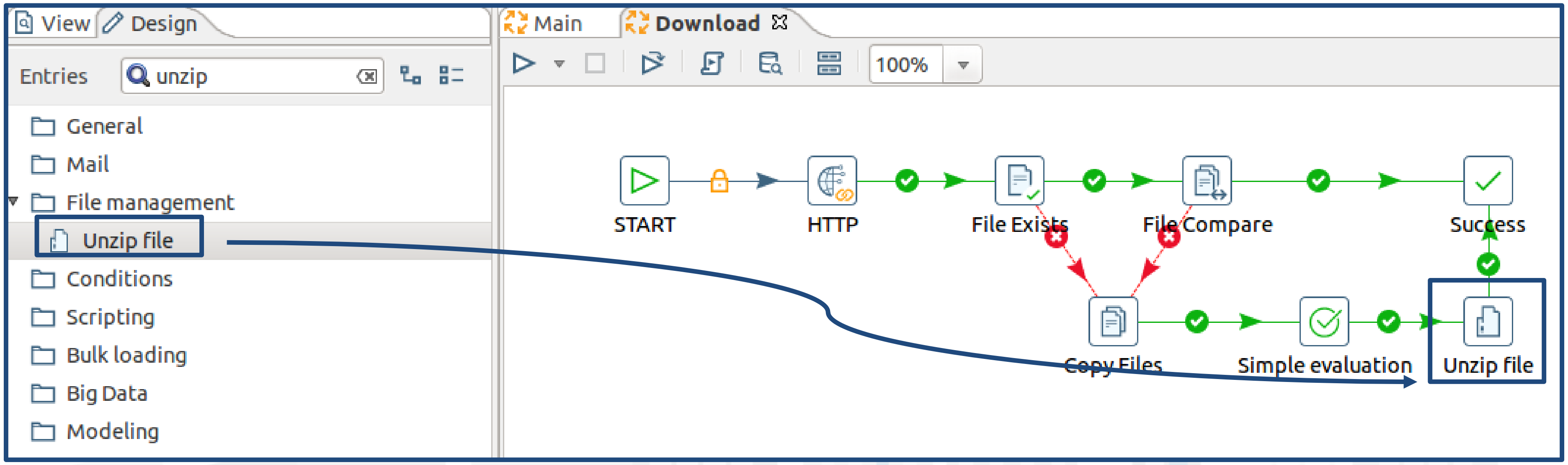
Success On

Success when variable set ☐

Success condition: If value is equal to

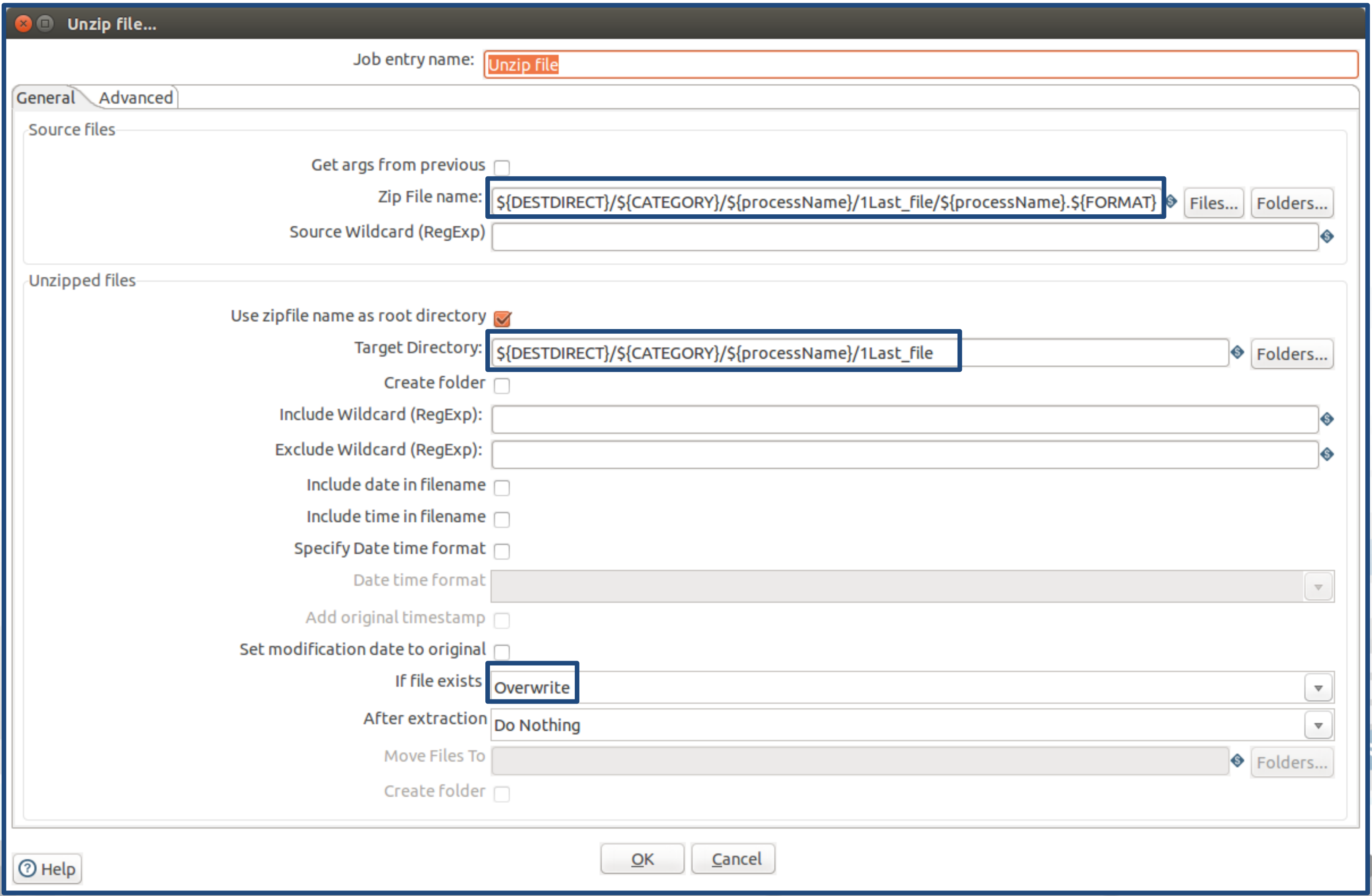
Value: zip

Unzip file (1)



View
Design
Entries
unzip
General
Mail
File management
Unzip file
Conditions
Scripting
Bulk loading
Big Data
Modeling
Main
Download
100%
START
HTTP
File Exists
File Compare
Success
Copy Files
Simple evaluation
Unzip file

Unzip file (2)



Unzip file...

Job entry name: Unzip file

General | Advanced

Source files

Get args from previous ☐

Zip File name: ${DESTDIRECT}/${CATEGORY}/${processName}/1Last_file/${processName}.${FORMAT}

Source Wildcard (RegExp):

Unzipped files

Use zipfile name as root directory ✓

Target Directory: ${DESTDIRECT}/${CATEGORY}/${processName}/1Last_file

Create folder ☐

Include Wildcard (RegExp):

Exclude Wildcard (RegExp):

Include date in filename ☐

Include time in filename ☐

Specify Date time format ☐

Date time format

Add original timestamp ☐

Set modification date to original ☐

If file exists: Overwrite

After extraction: Do Nothing

Move Files To

Create folder ☐

The Agency transformation (1)

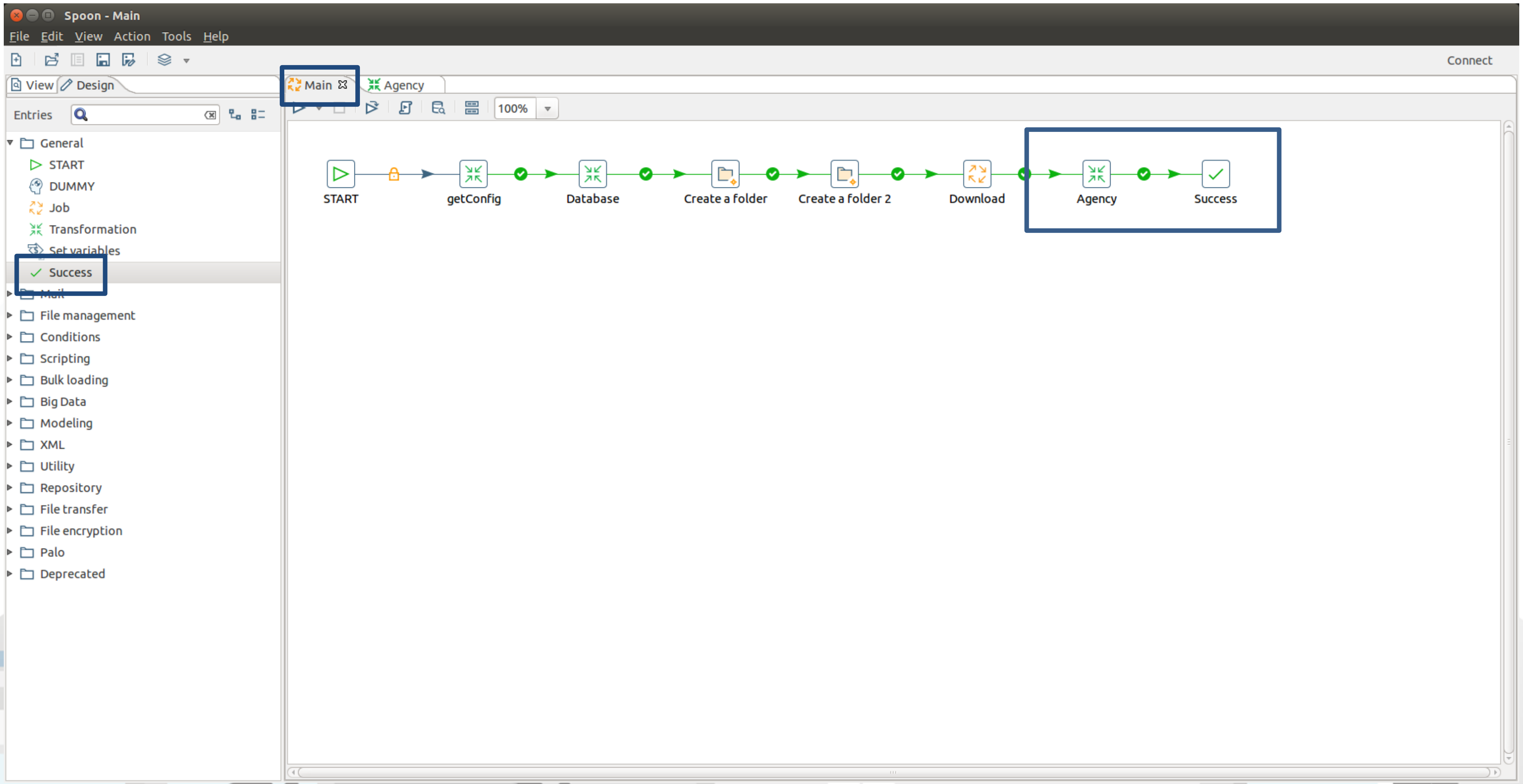
We are now ready to read data extracted from the archive, transform them, and put them into a HBase database.

In the today exercise, we will limit to the reading of metadata about public transport **agencies**. In first, we will put them in HBase through the Agency transformation that we are going to create and append to the Main job. Then, we will produce RDF triples from such data.

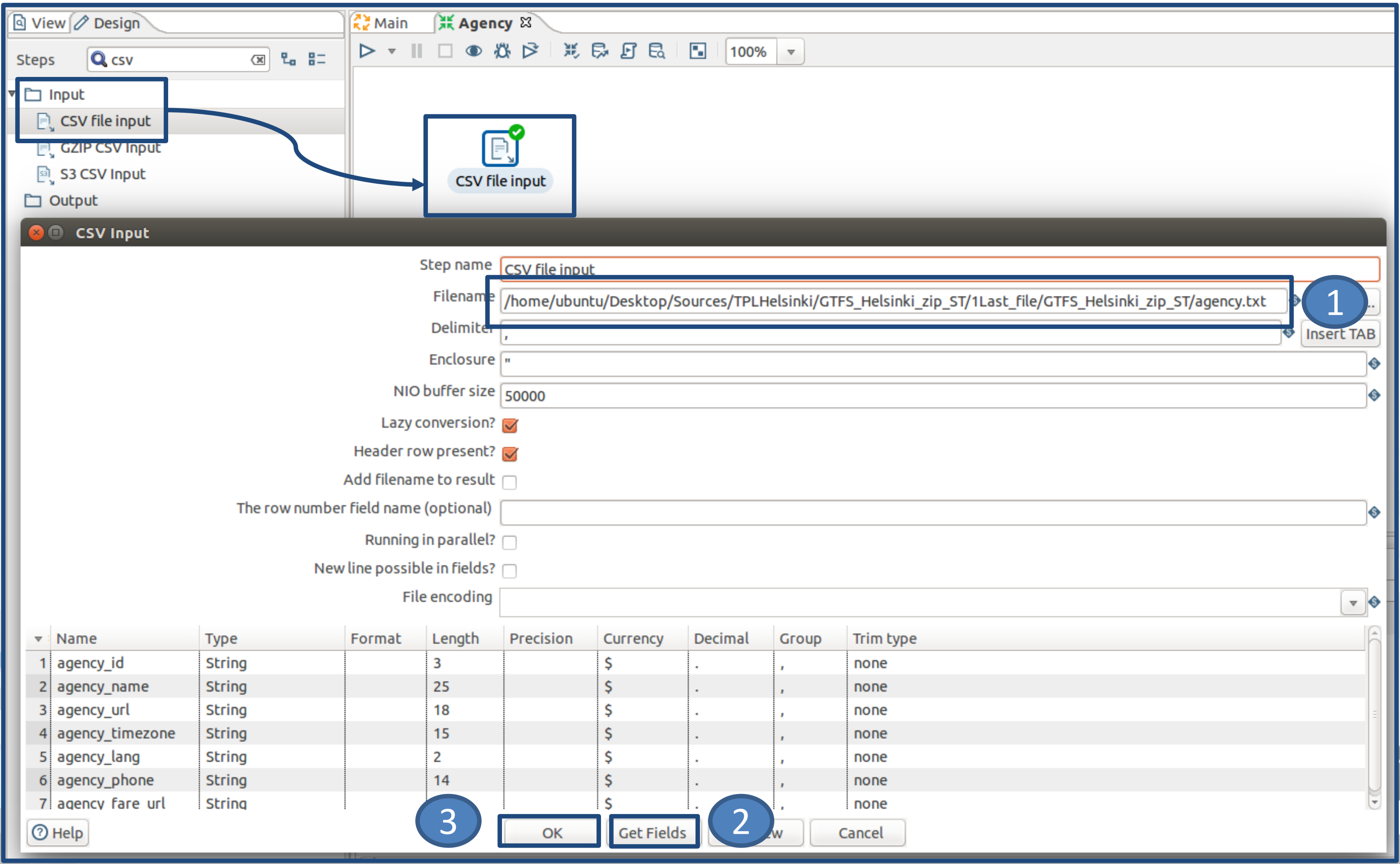
So let's start creating a new empty transformation named **Agency.ktr**, and appending a **Transformation** step at the end of the **Main** job, that points to the newly created transformation.

Storing data to HBase is the last task of the **Ingestion** phase. So, once you have added the **Transformation** step that points to **Agency.ktr**, you also can append a **Success** step, and finalize the **Main** job.

The Agency transformation (2)



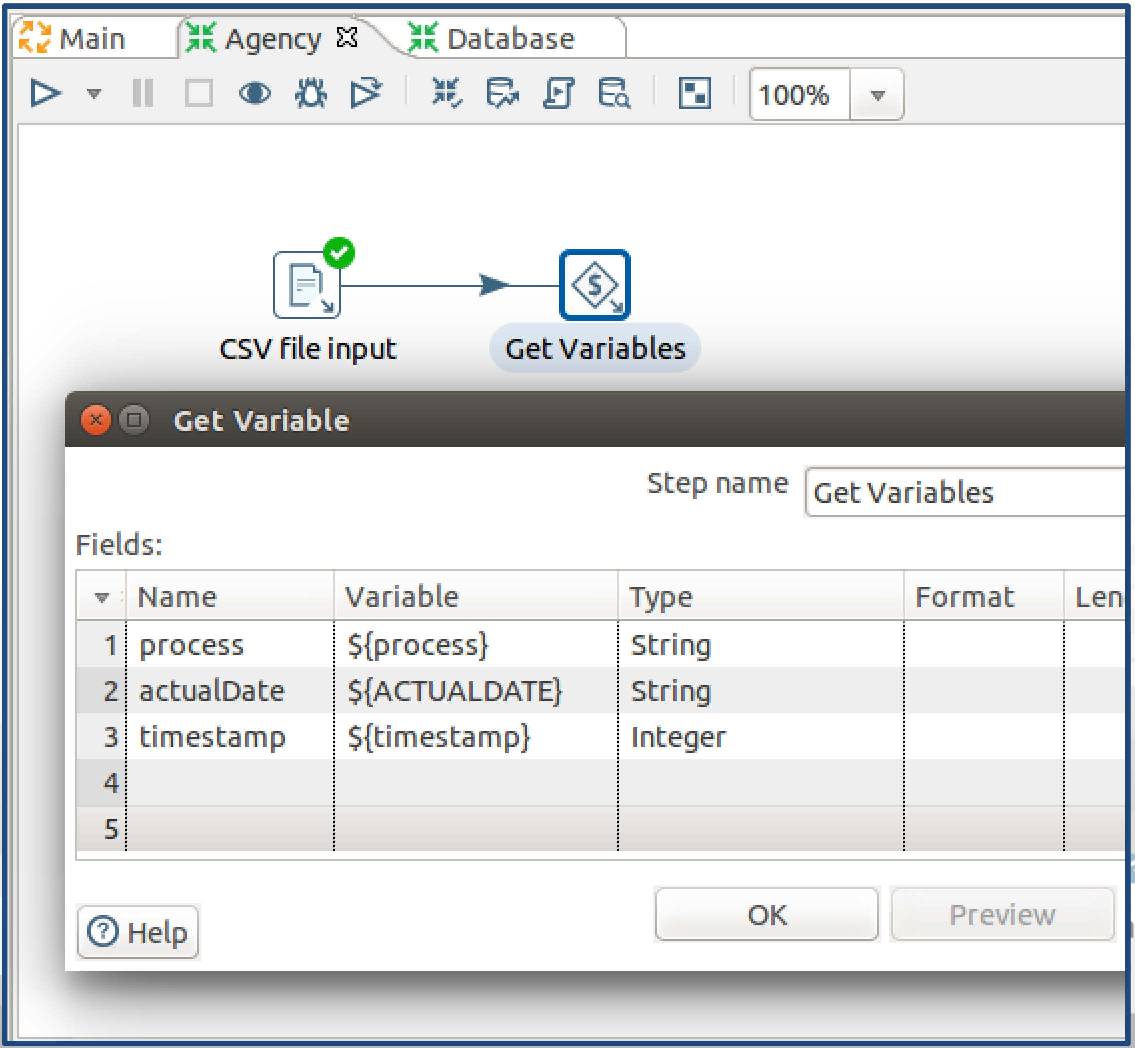
Agency: Read agency.txt



Step name: CSV file input
Filename: /home/ubuntu/Desktop/Sources/TPLHelsinki/GTFS_Helsinki_zip_ST/1Last_file/GTFS_Helsinki_zip_ST/agency.txt
Delimiter: ,
Enclosure: "
NIO buffer size: 50000
Lazy conversion? ✓
Header row present? ✓
Add filename to result ☐
The row number field name (optional):
Running in parallel? ☐
New line possible in fields? ☐
File encoding:

	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	agency_id	String		3		$	.	,	none
2	agency_name	String		25		$	.	,	none
3	agency_url	String		18		$	.	,	none
4	agency_timezone	String		15		$	.	,	none
5	agency_lang	String		2		$	.	,	none
6	agency_phone	String		14		$	.	,	none
7	agency_fare_url	String				$	.	,	none

Agency: Get Variables

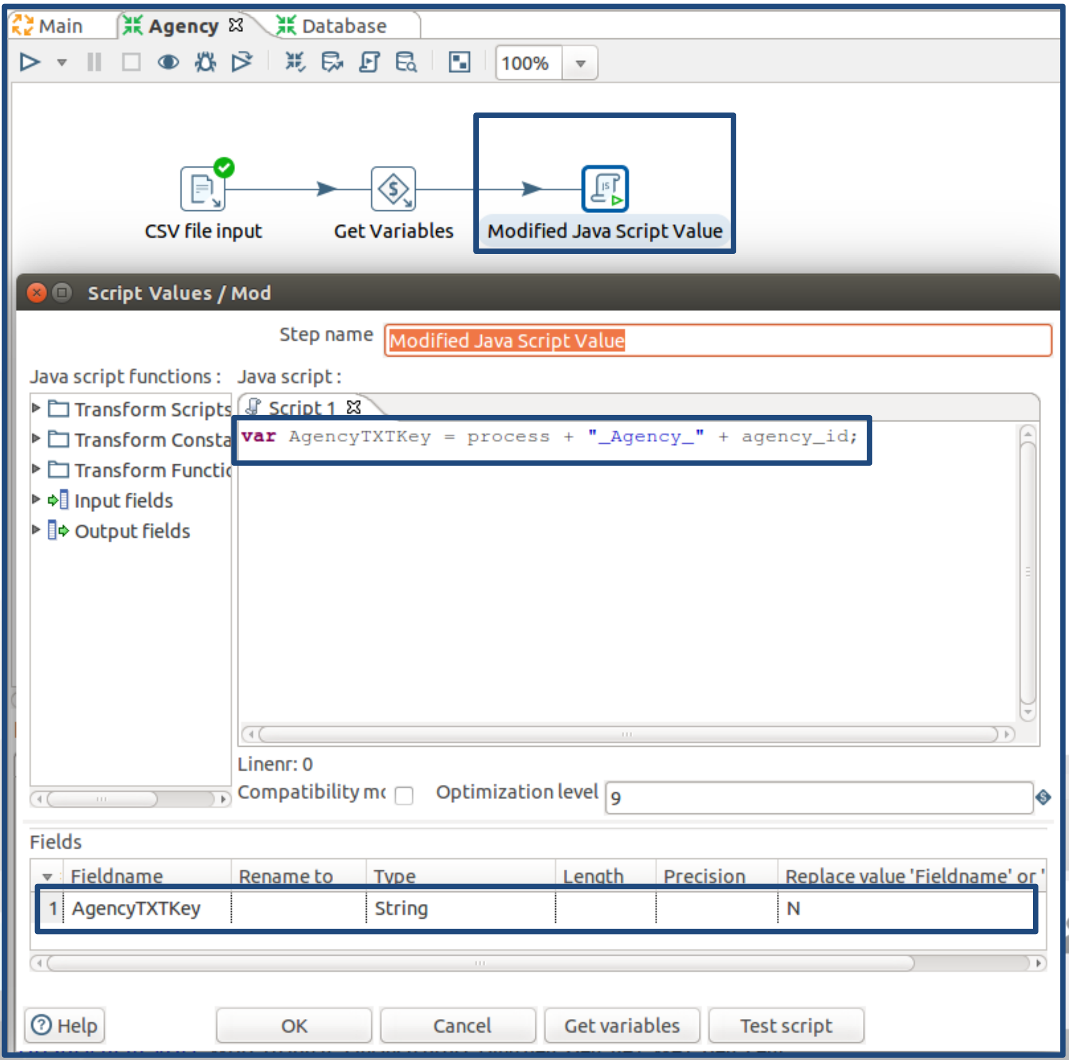


Step name: Get Variables

Fields:

	Name	Variable	Type	Format	Len
1	process	${process}	String		
2	actualDate	${ACTUALDATE}	String		
3	timestamp	${timestamp}	Integer		
4					
5					

Agency: Build Identifier



Step name: Modified Java Script Value

Java script :

Script 1

```
var AgencyTXTKey = process + "_Agency_" + agency_id;
```

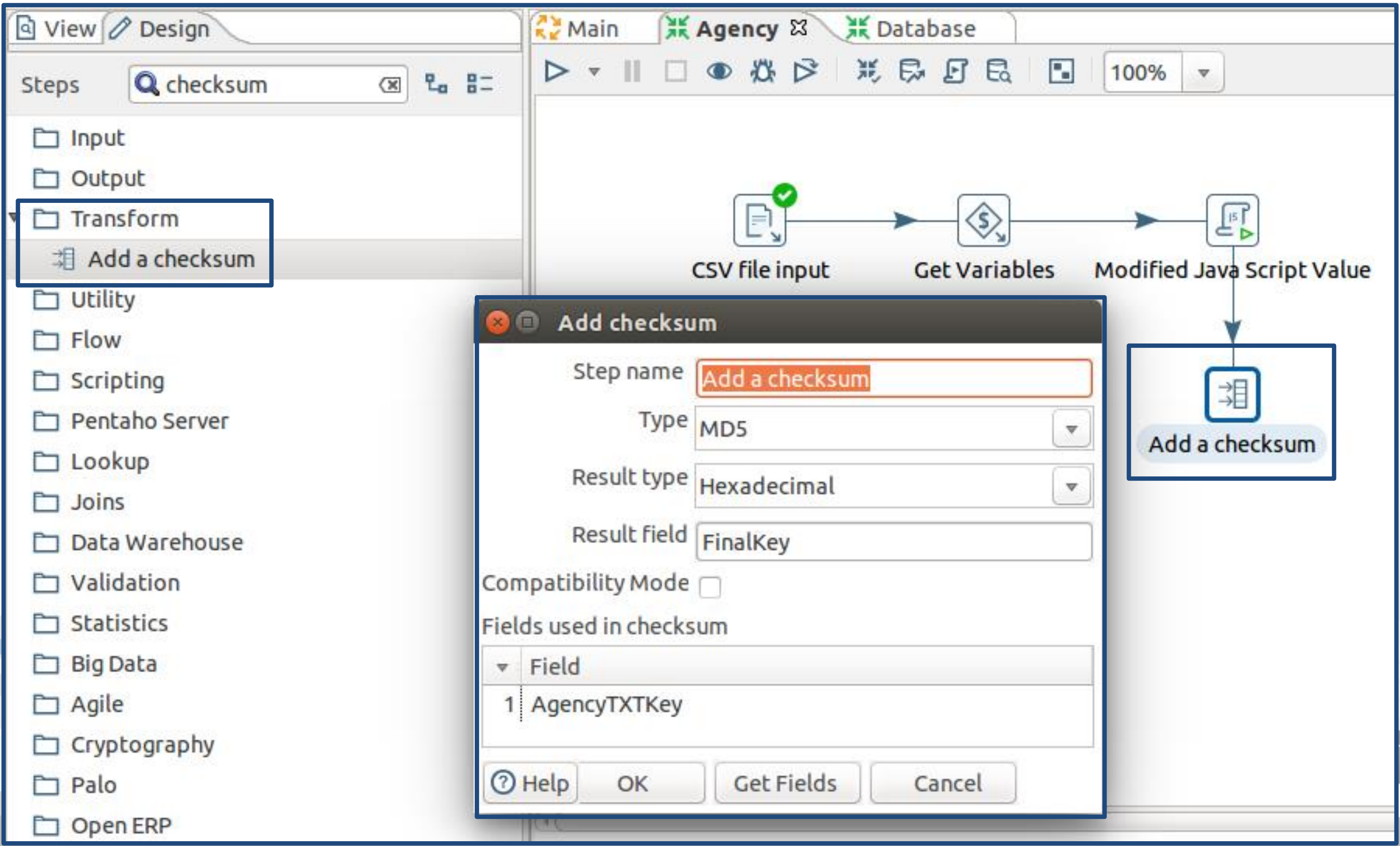
Linenr: 0

Compatibility mc ☐ Optimization level 9

Fields

	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or '
1	AgencyTXTKey		String			N

Agency: Add a checksum



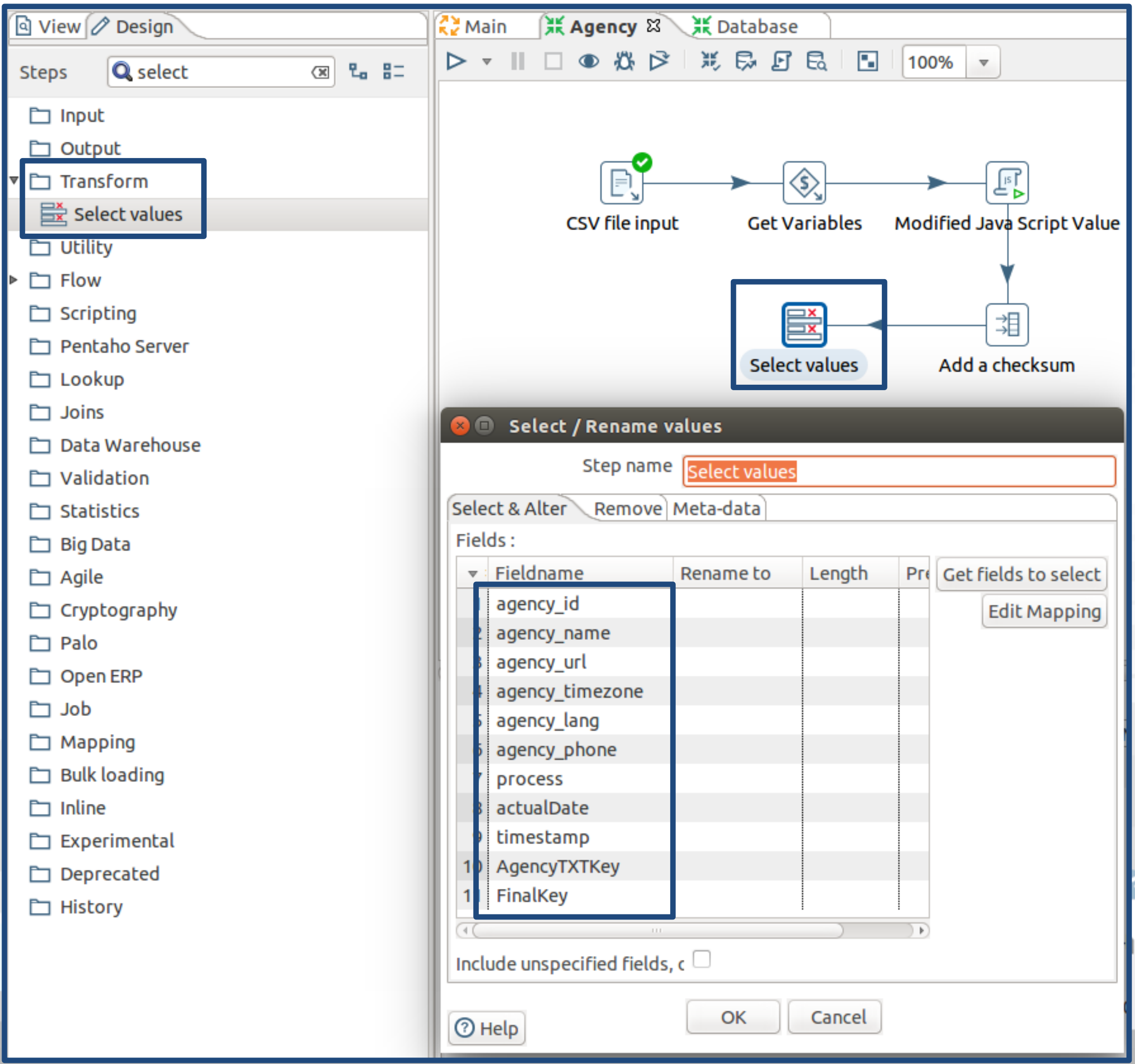
Add checksum

Step name: Add a checksum
Type: MD5
Result type: Hexadecimal
Result field: FinalKey
Compatibility Mode: ☐

Fields used in checksum

	Field
1	AgencyTXTKey

Agency: Select values



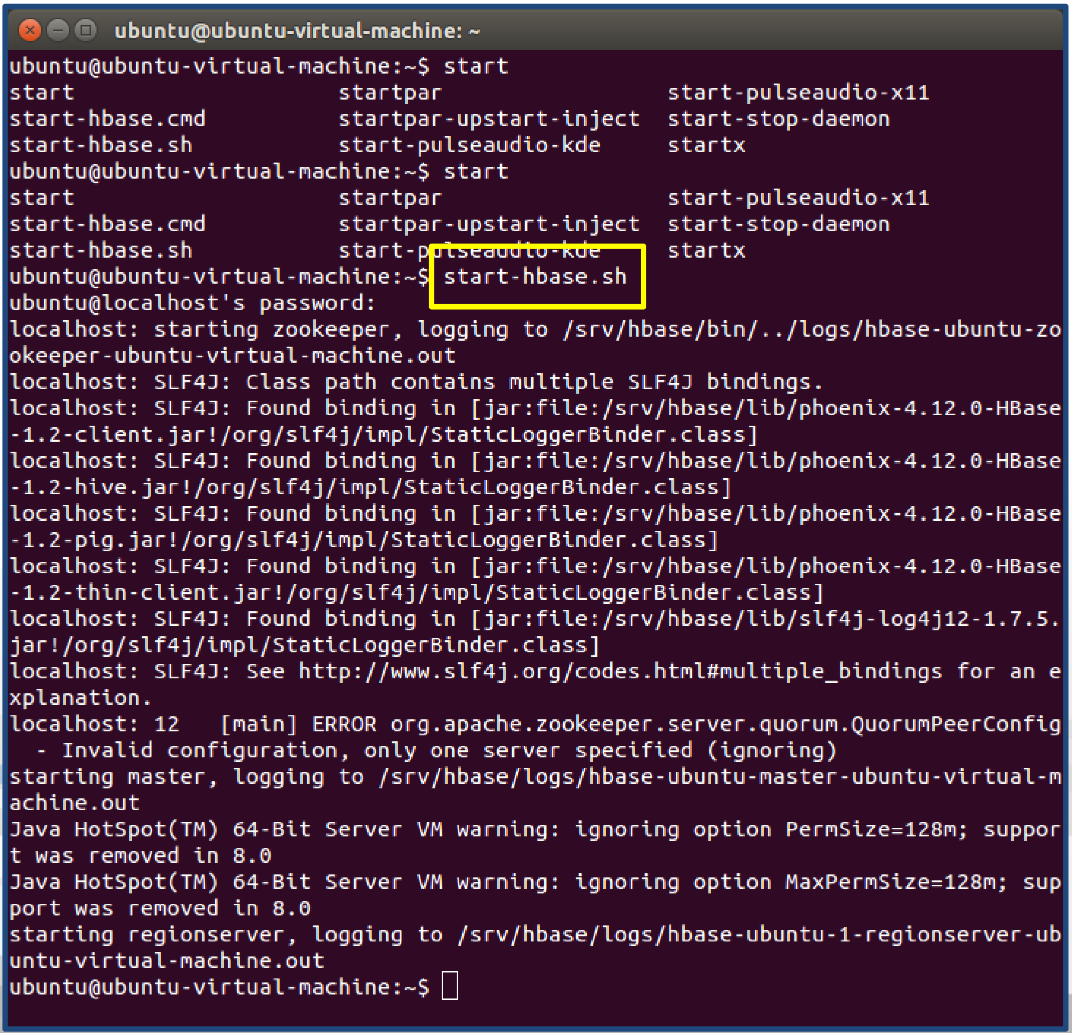
Step name: Select values

Fields :

#	Fieldname	Rename to	Length	Pre
1	agency_id			
2	agency_name			
3	agency_url			
4	agency_timezone			
5	agency_lang			
6	agency_phone			
7	process			
8	actualDate			
9	timestamp			
10	AgencyTXTKey			
11	FinalKey			

Include unspecified fields, c ☐

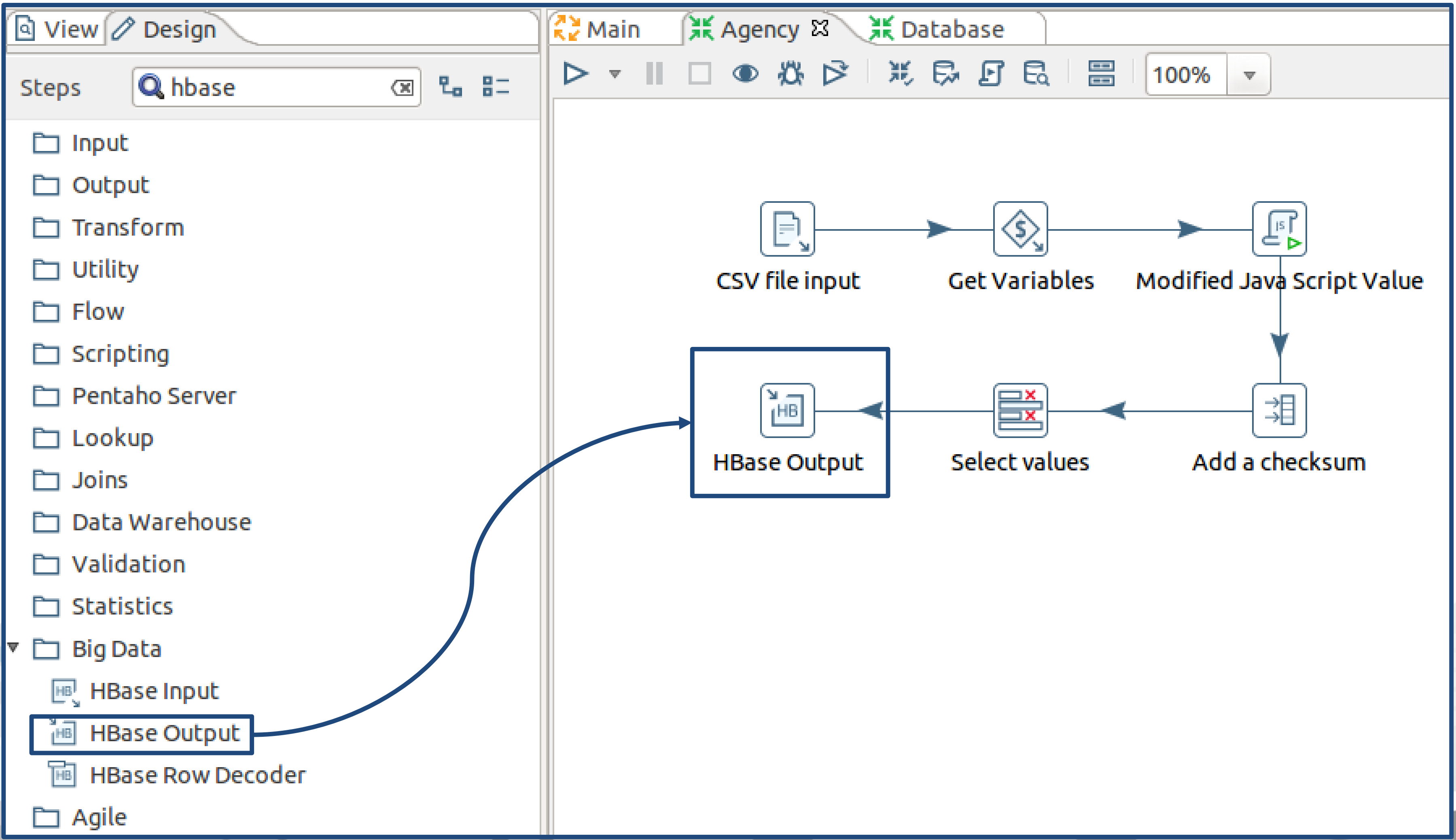
Agency: Write to HBase (1)



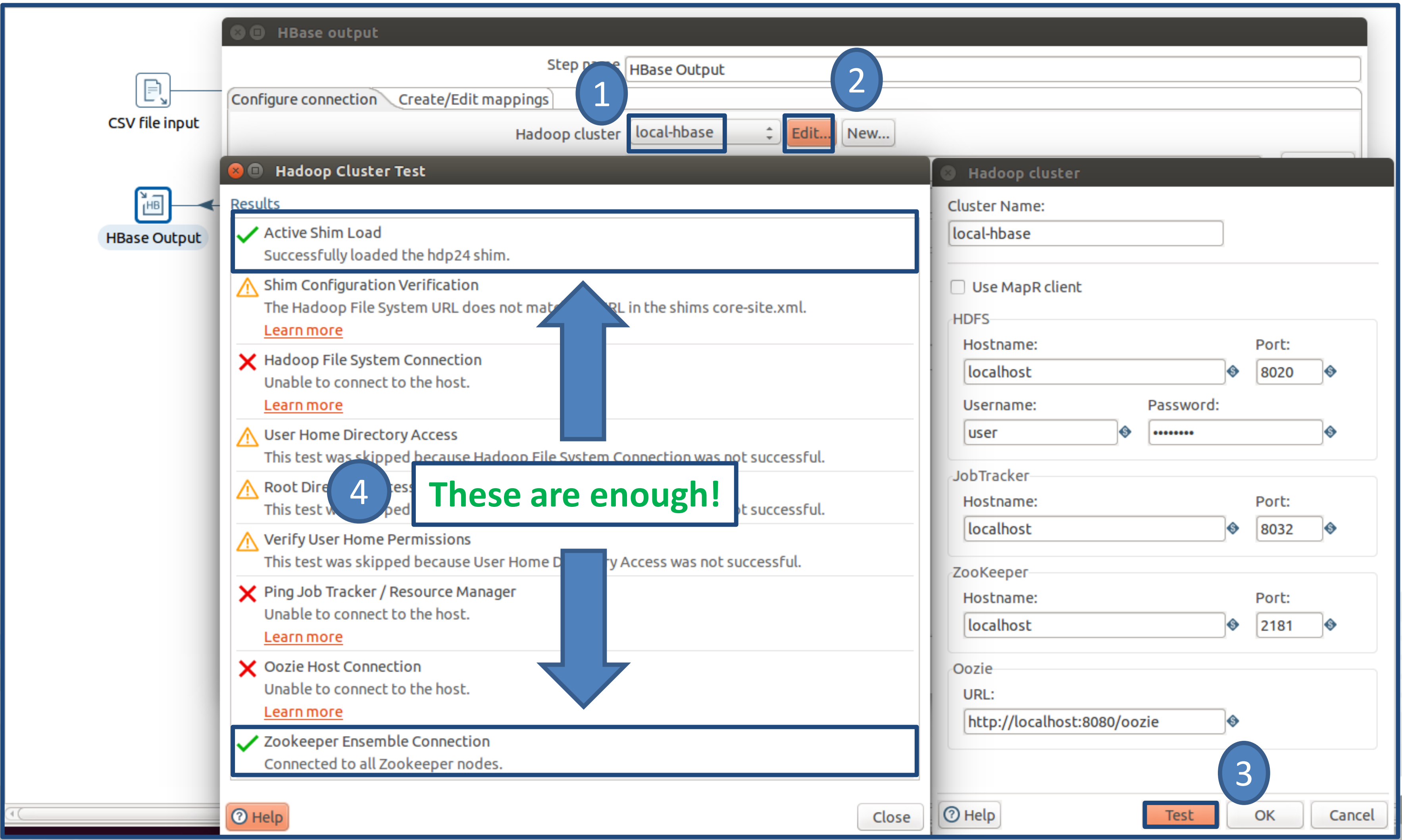
```
ubuntu@ubuntu-virtual-machine:~$ start
start                    startpar                 start-pulseaudio-x11
start-hbase.cmd          startpar-upstart-inject  start-stop-daemon
start-hbase.sh           start-pulseaudio-kde     startx
ubuntu@ubuntu-virtual-machine:~$ start
start                    startpar                 start-pulseaudio-x11
start-hbase.cmd          startpar-upstart-inject  start-stop-daemon
start-hbase.sh           start-pulseaudio-kde     startx
ubuntu@ubuntu-virtual-machine:~$ start-hbase.sh
ubuntu@localhost's password:
localhost: starting zookeeper, logging to /srv/hbase/bin/../logs/hbase-ubuntu-zo
okeeper-ubuntu-virtual-machine.out
localhost: SLF4J: Class path contains multiple SLF4J bindings.
localhost: SLF4J: Found binding in [jar:file:/srv/hbase/lib/phoenix-4.12.0-HBase
-1.2-client.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: Found binding in [jar:file:/srv/hbase/lib/phoenix-4.12.0-HBase
-1.2-hive.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: Found binding in [jar:file:/srv/hbase/lib/phoenix-4.12.0-HBase
-1.2-pig.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: Found binding in [jar:file:/srv/hbase/lib/phoenix-4.12.0-HBase
-1.2-thin-client.jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: Found binding in [jar:file:/srv/hbase/lib/slf4j-log4j12-1.7.5.
jar!/org/slf4j/impl/StaticLoggerBinder.class]
localhost: SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an e
xplanation.
localhost: 12   [main] ERROR org.apache.zookeeper.server.quorum.QuorumPeerConfig
  - Invalid configuration, only one server specified (ignoring)
starting master, logging to /srv/hbase/logs/hbase-ubuntu-master-ubuntu-virtual-m
achine.out
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; suppor
t was removed in 8.0
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=128m; sup
port was removed in 8.0
starting regionserver, logging to /srv/hbase/logs/hbase-ubuntu-1-regionserver-ub
untu-virtual-machine.out
ubuntu@ubuntu-virtual-machine:~$
```

Despite of errors that you can see in the terminal after the issuing of the **start-hbase.sh**, the server starts successfully for our purposes.

Agency: Write to HBase (2)



Agency: Write to HBase (3)



HBase output

Step name: HBase Output

Configure connection | Create/Edit mappings

Hadoop cluster: local-hbase

Hadoop Cluster Test

Results

✓ Active Shim Load
Successfully loaded the hdp24 shim.

⚠ Shim Configuration Verification
The Hadoop File System URL does not mat[...]RL in the shims core-site.xml.
Learn more

✗ Hadoop File System Connection
Unable to connect to the host.
Learn more

⚠ User Home Directory Access
This test was skipped because Hadoop File System Connection was not successful.

⚠ Root Dire[...]ess
This test w[...]ped[...]t successful.

⚠ Verify User Home Permissions
This test was skipped because User Home D[...]y Access was not successful.

✗ Ping Job Tracker / Resource Manager
Unable to connect to the host.
Learn more

✗ Oozie Host Connection
Unable to connect to the host.
Learn more

✓ Zookeeper Ensemble Connection
Connected to all Zookeeper nodes.

Hadoop cluster

Cluster Name: local-hbase

☐ Use MapR client

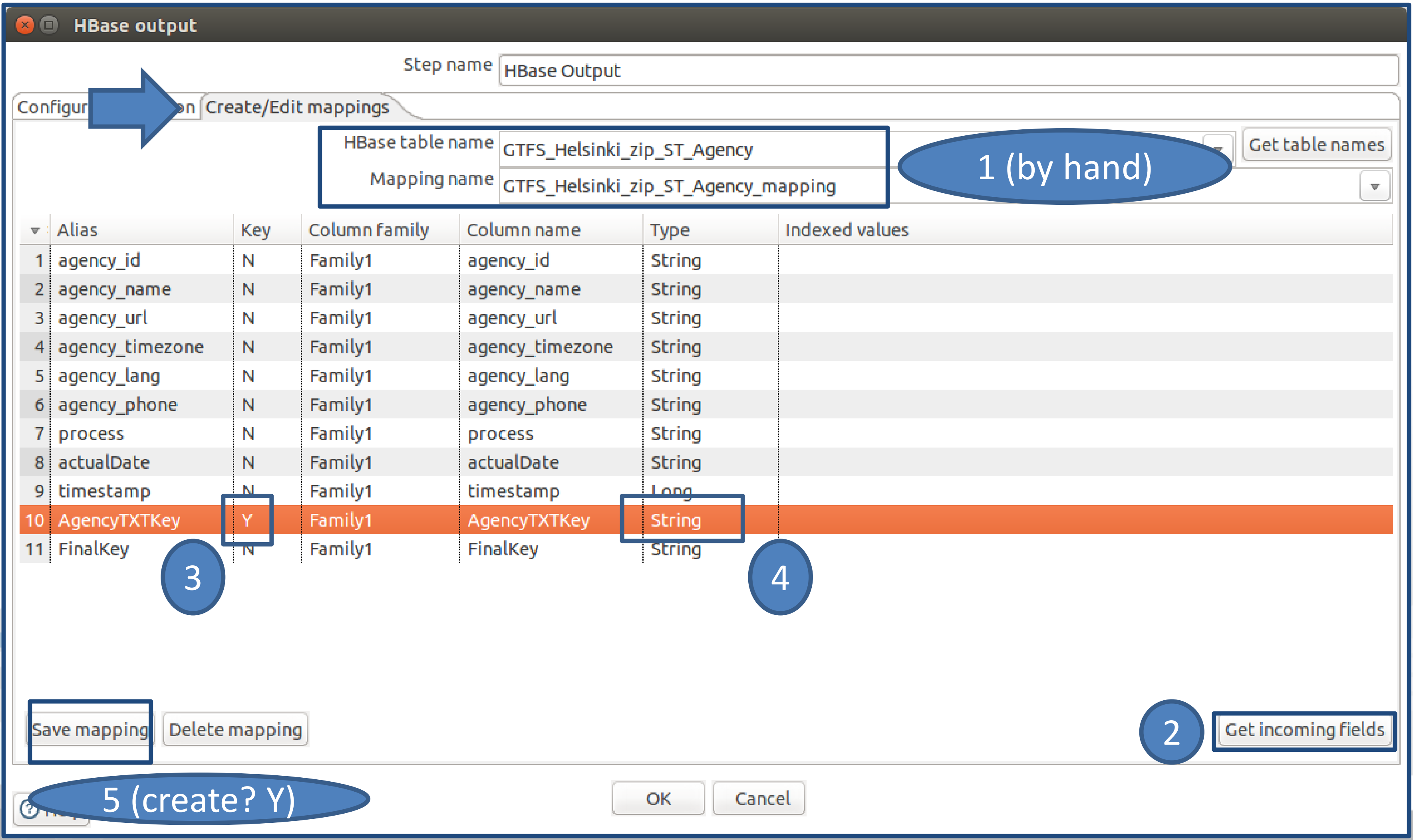
HDFS
Hostname: localhost Port: 8020
Username: user Password: ••••••••

JobTracker
Hostname: localhost Port: 8032

ZooKeeper
Hostname: localhost Port: 2181

Oozie
URL: http://localhost:8080/oozie

Agency: Write to HBase (4)



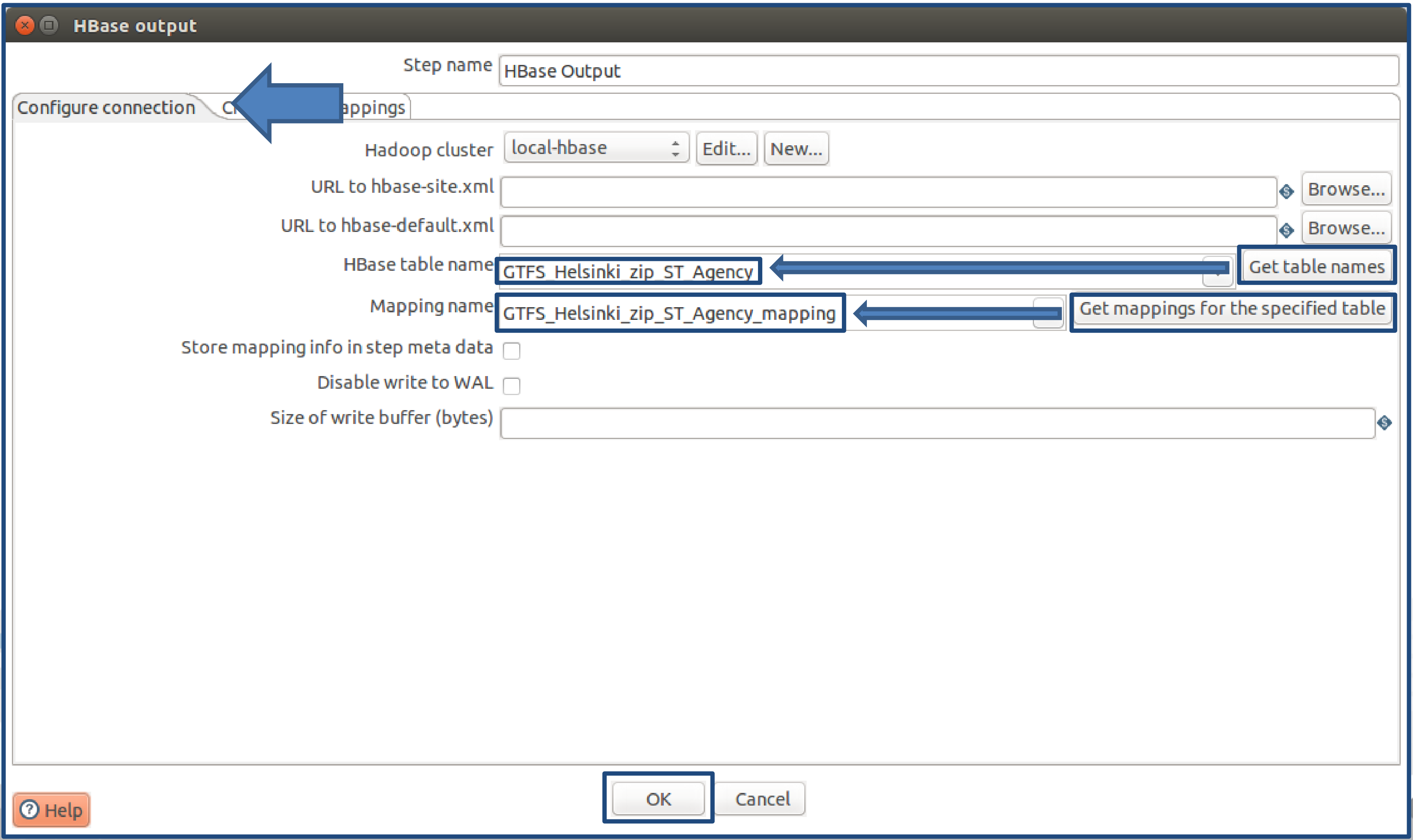
Step name: HBase Output

HBase table name: GTFS_Helsinki_zip_ST_Agency

Mapping name: GTFS_Helsinki_zip_ST_Agency_mapping

	Alias	Key	Column family	Column name	Type	Indexed values
1	agency_id	N	Family1	agency_id	String	
2	agency_name	N	Family1	agency_name	String	
3	agency_url	N	Family1	agency_url	String	
4	agency_timezone	N	Family1	agency_timezone	String	
5	agency_lang	N	Family1	agency_lang	String	
6	agency_phone	N	Family1	agency_phone	String	
7	process	N	Family1	process	String	
8	actualDate	N	Family1	actualDate	String	
9	timestamp	N	Family1	timestamp	Long	
10	AgencyTXTKey	Y	Family1	AgencyTXTKey	String	
11	FinalKey	N	Family1	FinalKey	String	

Agency: Write to HBase (5)


HBase output

Step name: HBase Output

Configure connection | C appings

Hadoop cluster: local-hbase | Edit... | New...

URL to hbase-site.xml: | Browse...

URL to hbase-default.xml: | Browse...

HBase table name: GTFS_Helsinki_zip_ST_Agency | Get table names

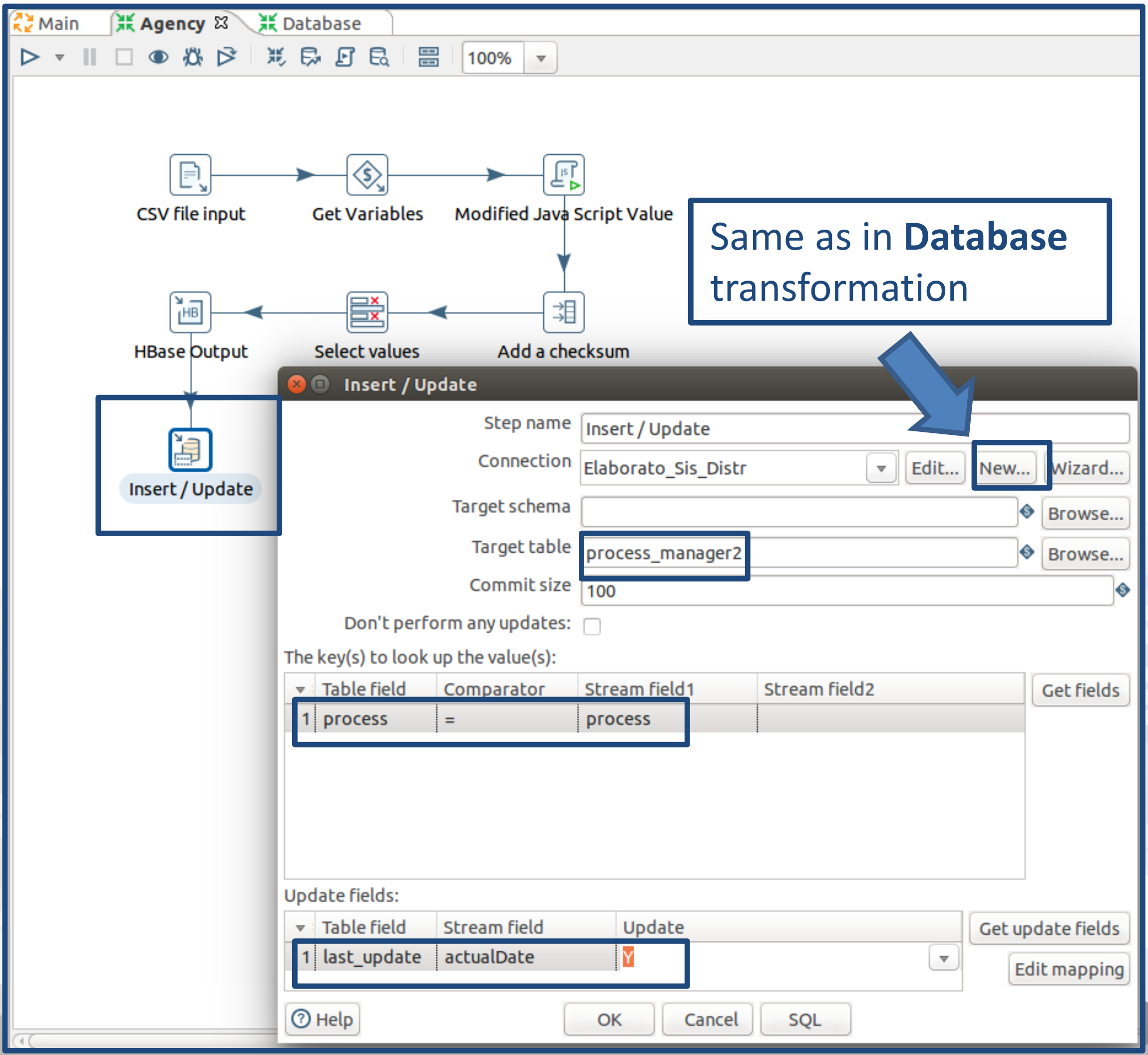
Mapping name: GTFS_Helsinki_zip_ST_Agency_mapping | Get mappings for the specified table

Store mapping info in step meta data ☐

Disable write to WAL ☐

Size of write buffer (bytes):

Agency: Update MySQL table



Same as in **Database** transformation

Insert / Update

Step name	Insert / Update
Connection	Elaborato_Sis_Distr
Target schema	
Target table	process_manager2
Commit size	100
Don't perform any updates:	☐

The key(s) to look up the value(s):

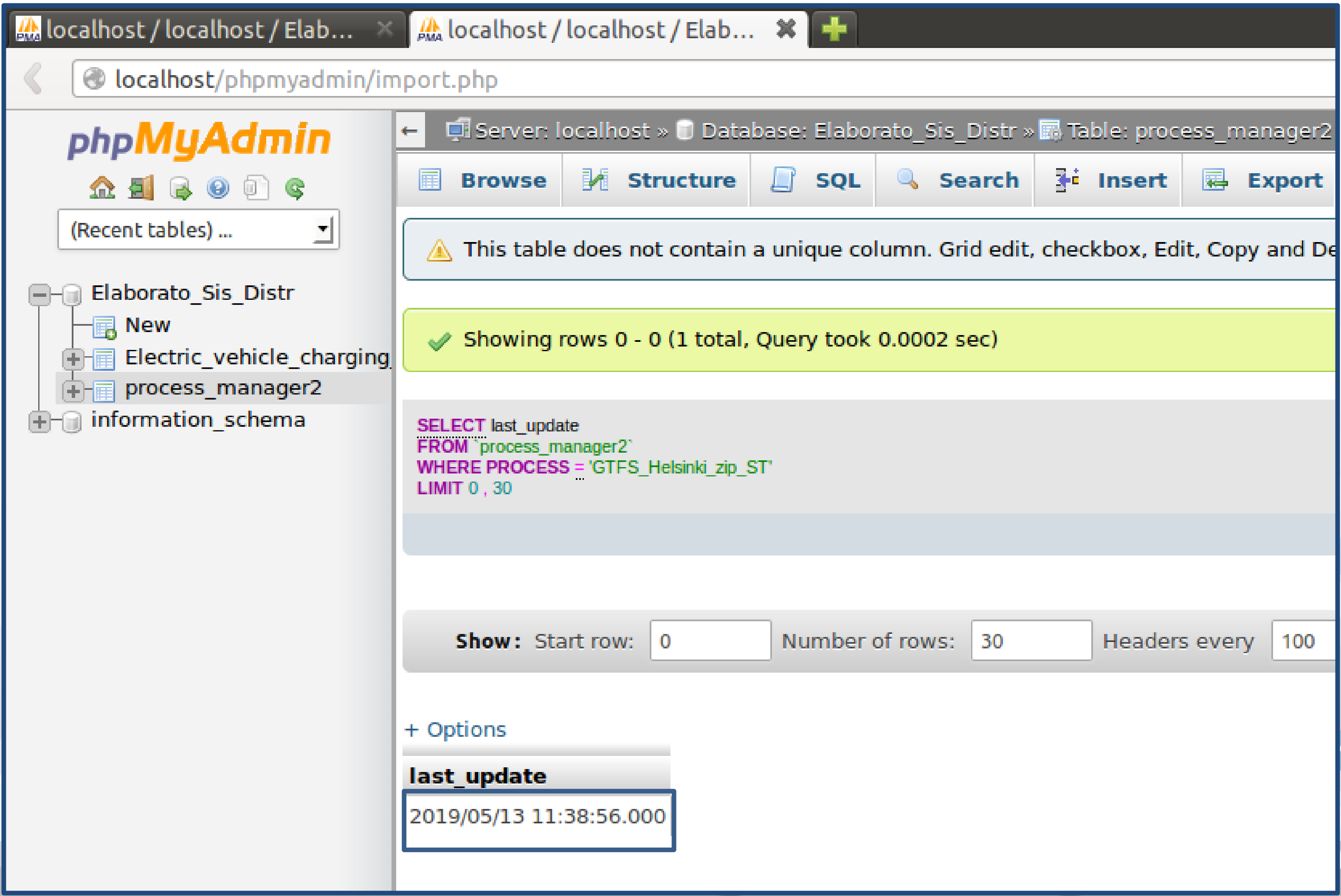
	Table field	Comparator	Stream field1	Stream field2
1	process	=	process	

Update fields:

	Table field	Stream field	Update
1	last_update	actualDate	Y

UNIVERSITÀ DEGLI STUDI FIRENZE DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org

Agency: Check MySQL table


localhost / localhost / Elab...
localhost / localhost / Elab...
localhost/phpmyadmin/import.php
phpMyAdmin
(Recent tables) ...
Elaborato_Sis_Distr
New
Electric_vehicle_charging
process_manager2
information_schema
Server: localhost » Database: Elaborato_Sis_Distr » Table: process_manager2
Browse
Structure
SQL
Search
Insert
Export

This table does not contain a unique column. Grid edit, checkbox, Edit, Copy and De

Showing rows 0 - 0 (1 total, Query took 0.0002 sec)

```
SELECT last_update
FROM `process_manager2`
WHERE PROCESS = 'GTFS_Helsinki_zip_ST'
LIMIT 0 , 30
```

Show : Start row: 0 Number of rows: 30 Headers every 100

+ Options

last_update
2019/05/13 11:38:56.000

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org

Homework (optional)

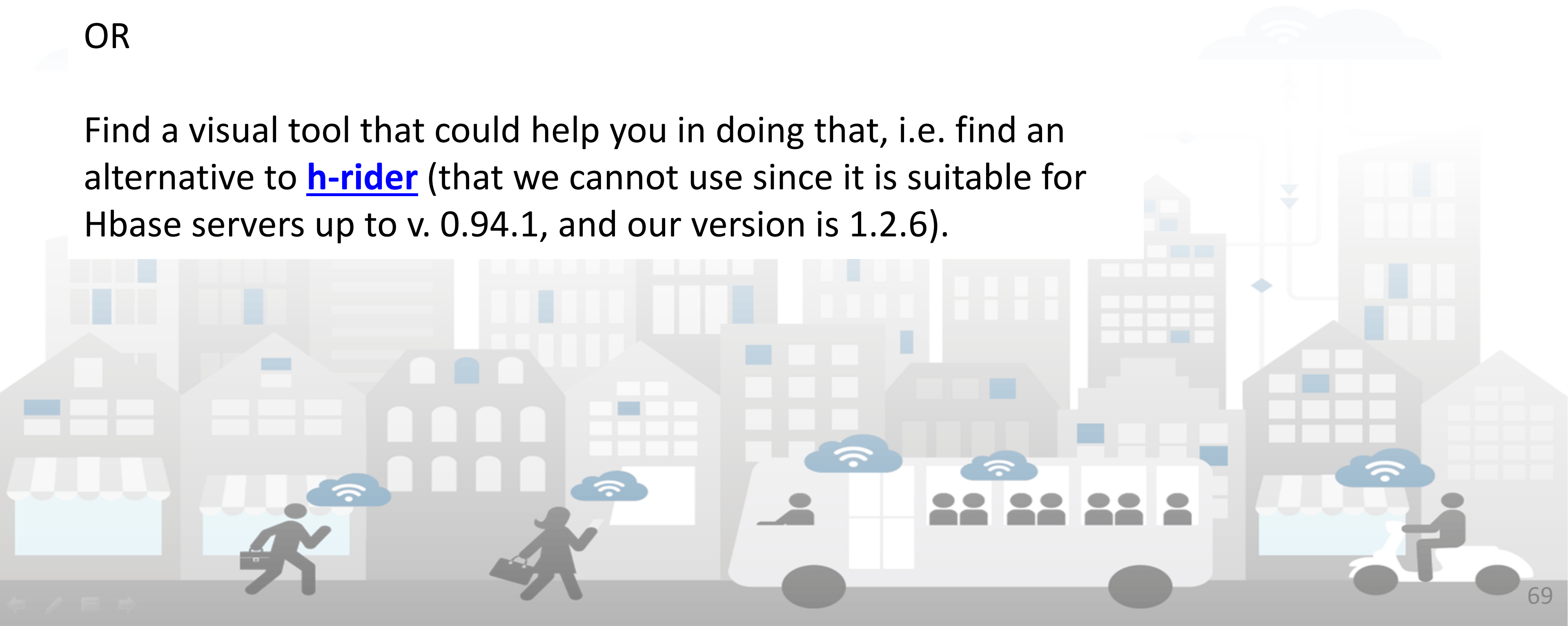
Exploit HBase REST APIs documented here:

- https://hbase.apache.org/1.2/book.html#_rest

to inspect data stored in the **GTFS_Helsinki_zip_ST_Agency** table

OR

Find a visual tool that could help you in doing that, i.e. find an alternative to **h-rider** (that we cannot use since it is suitable for Hbase servers up to v. 0.94.1, and our version is 1.2.6).



Karma Model

Run the Karma Server

Do the following to run the Karma server:

1. Open a shell
2. Move to ~/programs/Web-Karma-master/karma-web
3. Run mvn -Djetty.port=9999 jetty:run
4. Wait while the Jetty server comes up
5. Connect to localhost:9999 where you will find the Web application for building your model

Download Ontologies (1)

Download the following ontologies in a position of your choice in your VM

KM4City Ontology:
http://www.disit.org/drupal/?q=home&axoid=urn%3Aaxmedis%3A00000%3Aobj%3Aa863cca5-6dcc-492d-9afa-0c852aa34ae2

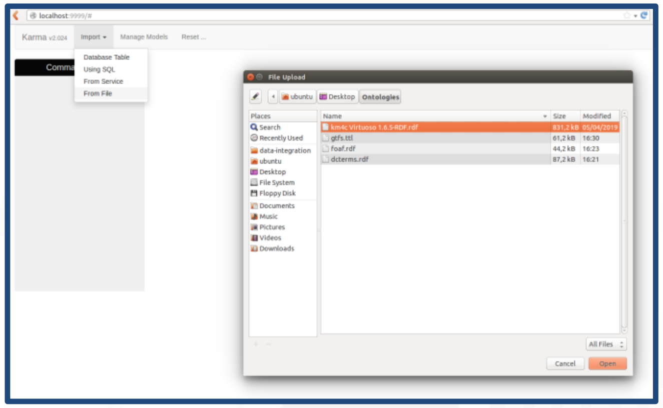
DCMI Metadata Terms:
http://dublincore.org/2012/06/14/dcterms.rdf

Friend of a Friend vocabulary:
http://xmlns.com/foaf/spec/index.rdf

General Transit Feed Specification:
https://raw.githubusercontent.com/OpenTransport/linked-gtfs/master/gtfs.ttl

Download Ontologies (2)

... and load them to Karma



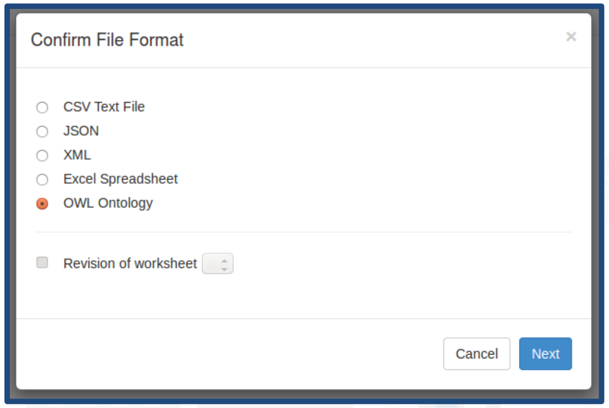
localhost:9999/#

Karma v2.024 | Import | Manage Models | Reset ...

File Upload

ubuntu Desktop **Ontologies**

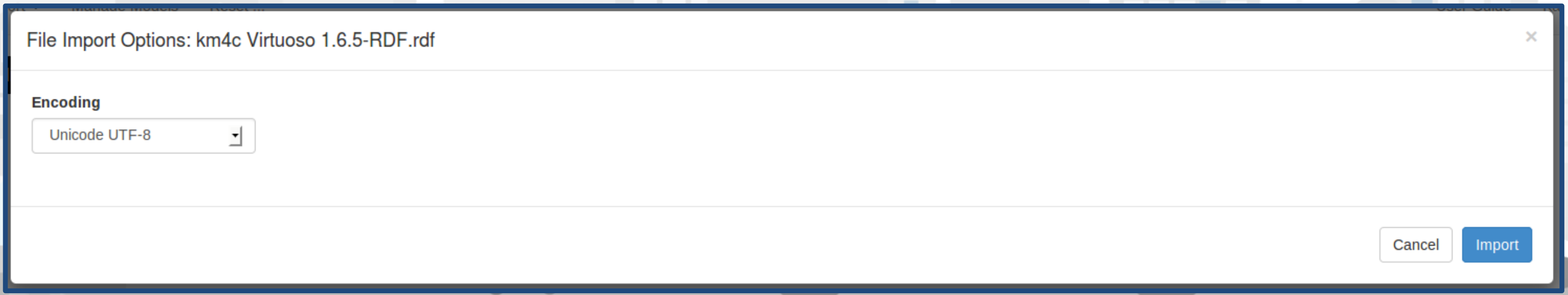
Name	Size	Modified
km4c Virtuoso 1.6.5-RDF.rdf	831,2 kB	05/04/2019
gtfs.ttl	61,2 kB	16:30
foaf.rdf	44,2 kB	16:23
dcterms.rdf	87,2 kB	16:21



Confirm File Format

- CSV Text File
- JSON
- XML
- Excel Spreadsheet
- [✓ OWL Ontology]

Revision of worksheet

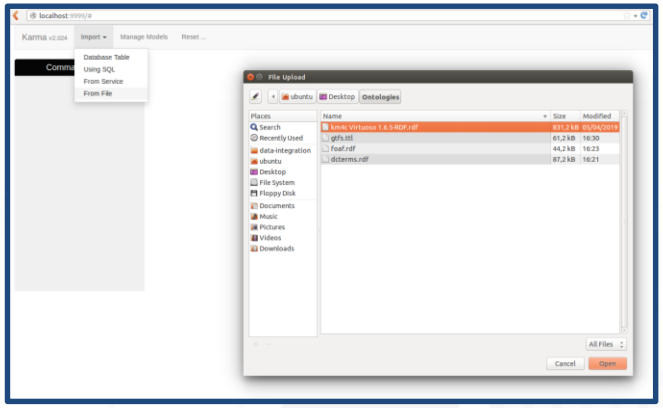


File Import Options: km4c Virtuoso 1.6.5-RDF.rdf

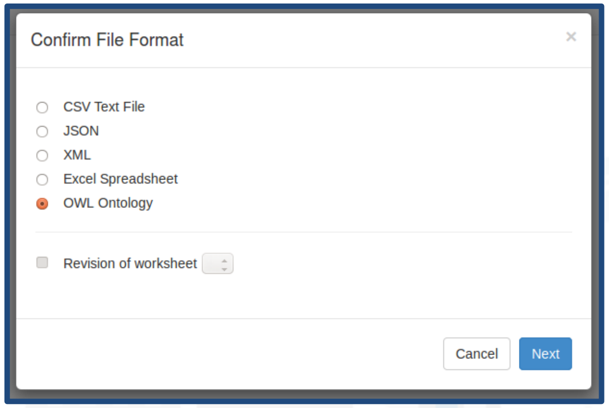
Encoding: Unicode UTF-8

Download Ontologies (2)

… and load them to Karma



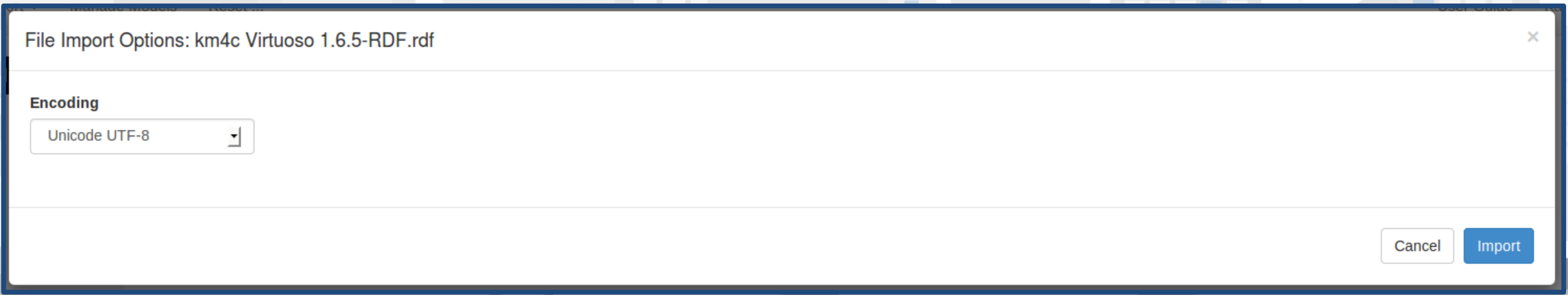
Name	Size	Modified
km4c Virtuoso 1.6.5-RDF.rdf	831,2 kB	05/04/2019
gtfs.ttl	61,2 kB	16:30
foaf.rdf	44,2 kB	16:23
dcterms.rdf	87,2 kB	16:21



Confirm File Format

- ○ CSV Text File
- ○ JSON
- ○ XML
- ○ Excel Spreadsheet
- ● OWL Ontology

☐ Revision of worksheet



File Import Options: km4c Virtuoso 1.6.5-RDF.rdf

Encoding

Unicode UTF-8

Load MySQL table

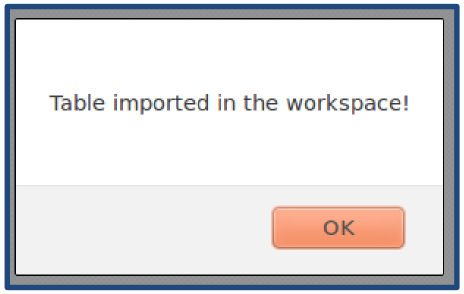
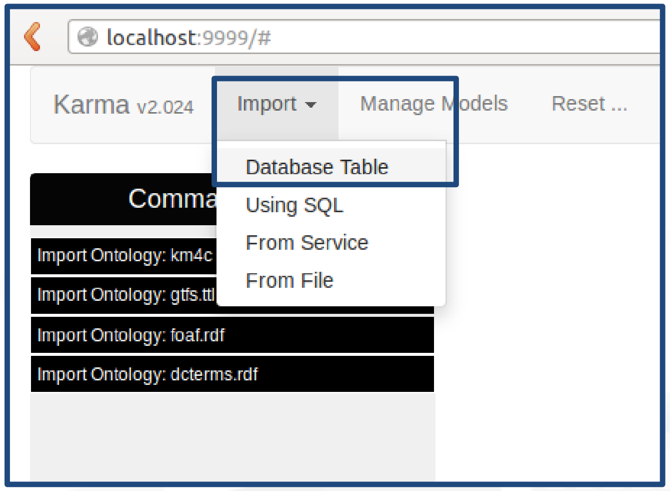
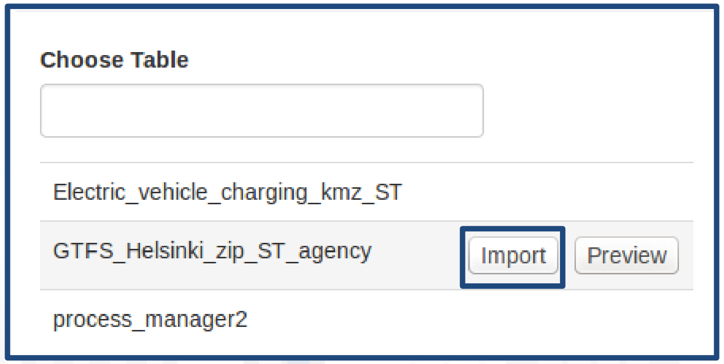


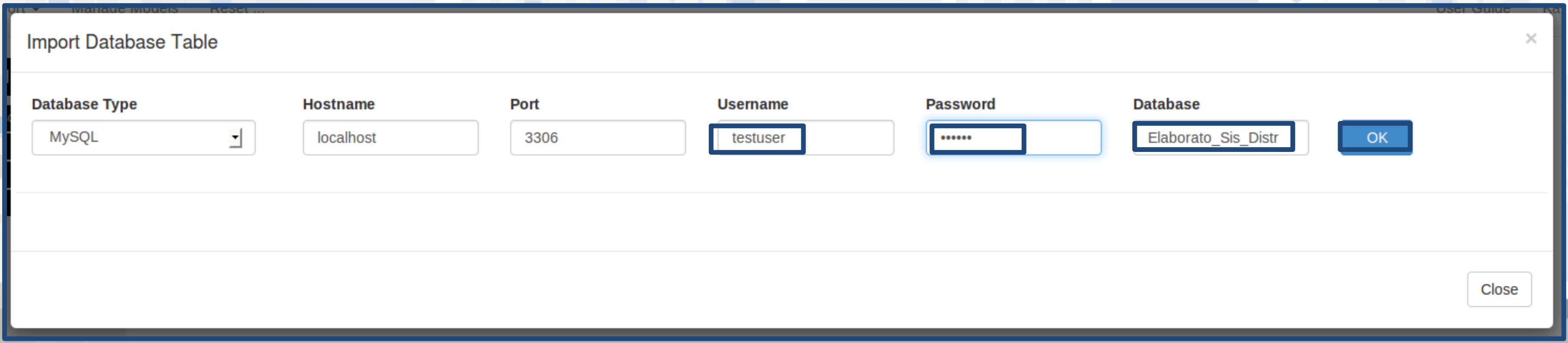
Table imported in the workspace!



Import Ontology: km4c
Import Ontology: gtfs.ttl
Import Ontology: foaf.rdf
Import Ontology: dcterms.rdf

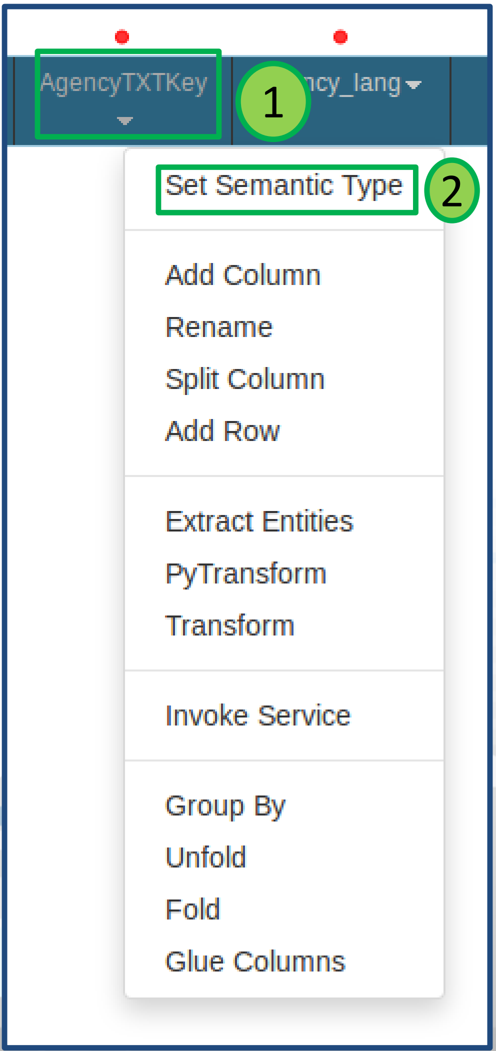

Choose Table

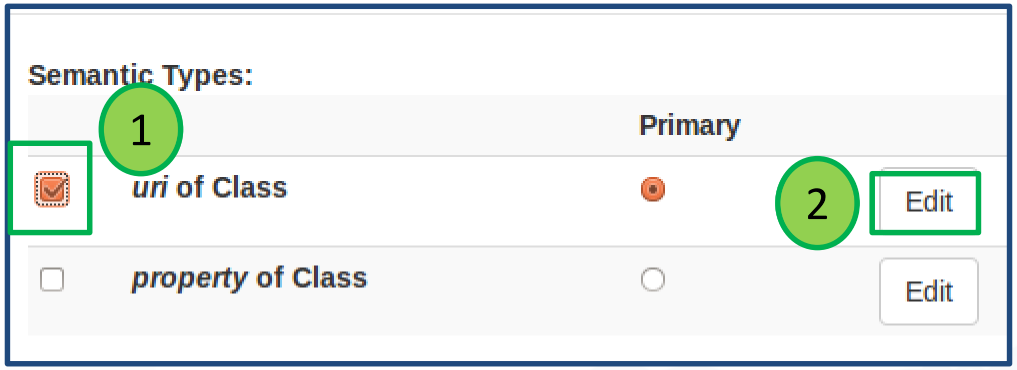
Electric_vehicle_charging_kmz_ST
GTFS_Helsinki_zip_ST_agency
process_manager2

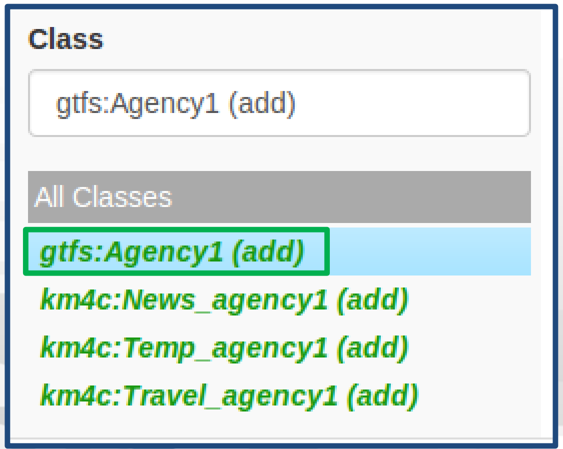

Import Database Table

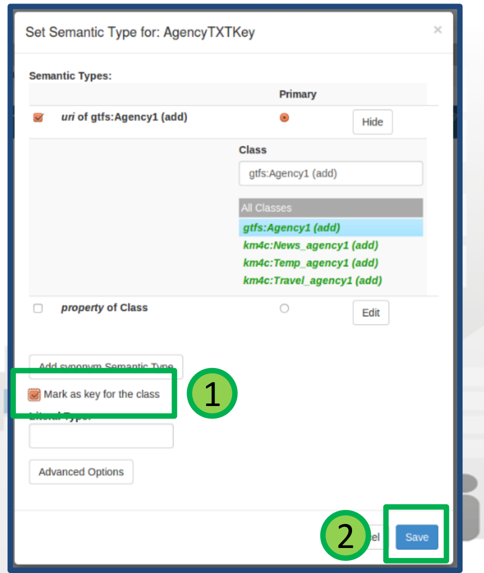
Database Type	Hostname	Port	Username	Password	Database
MySQL	localhost	3306	testuser	••••••	Elaborato_Sis_Distr

Map the identifier

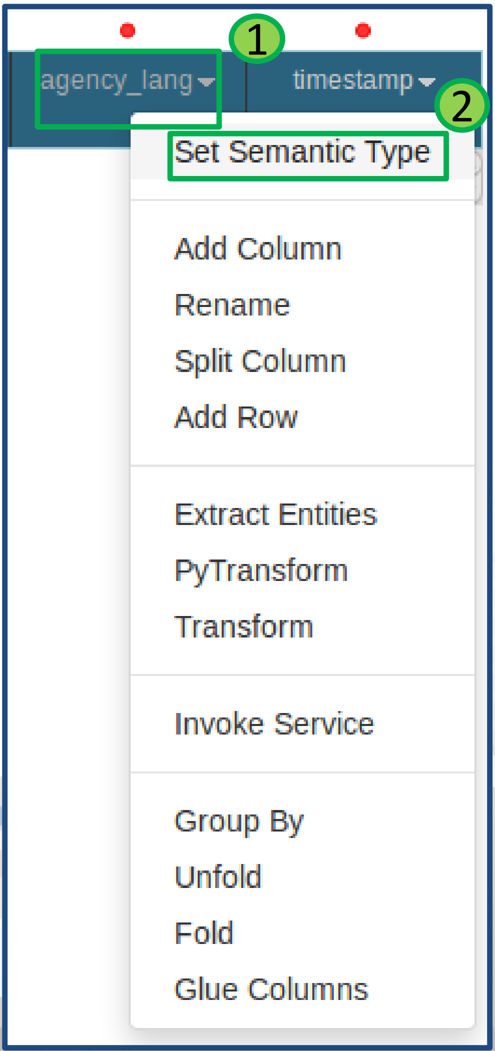


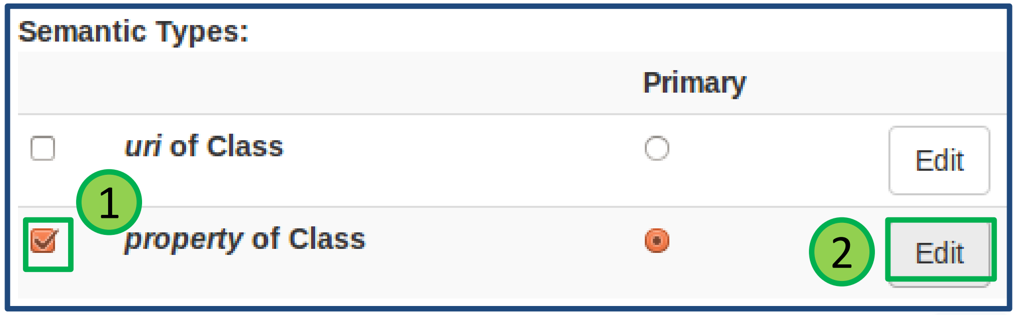


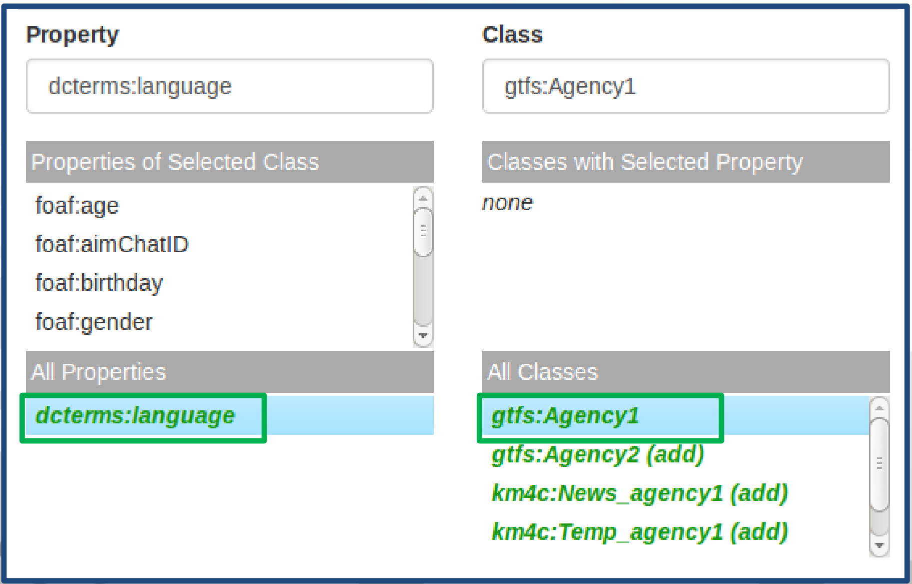




Map other properties (1)







UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org

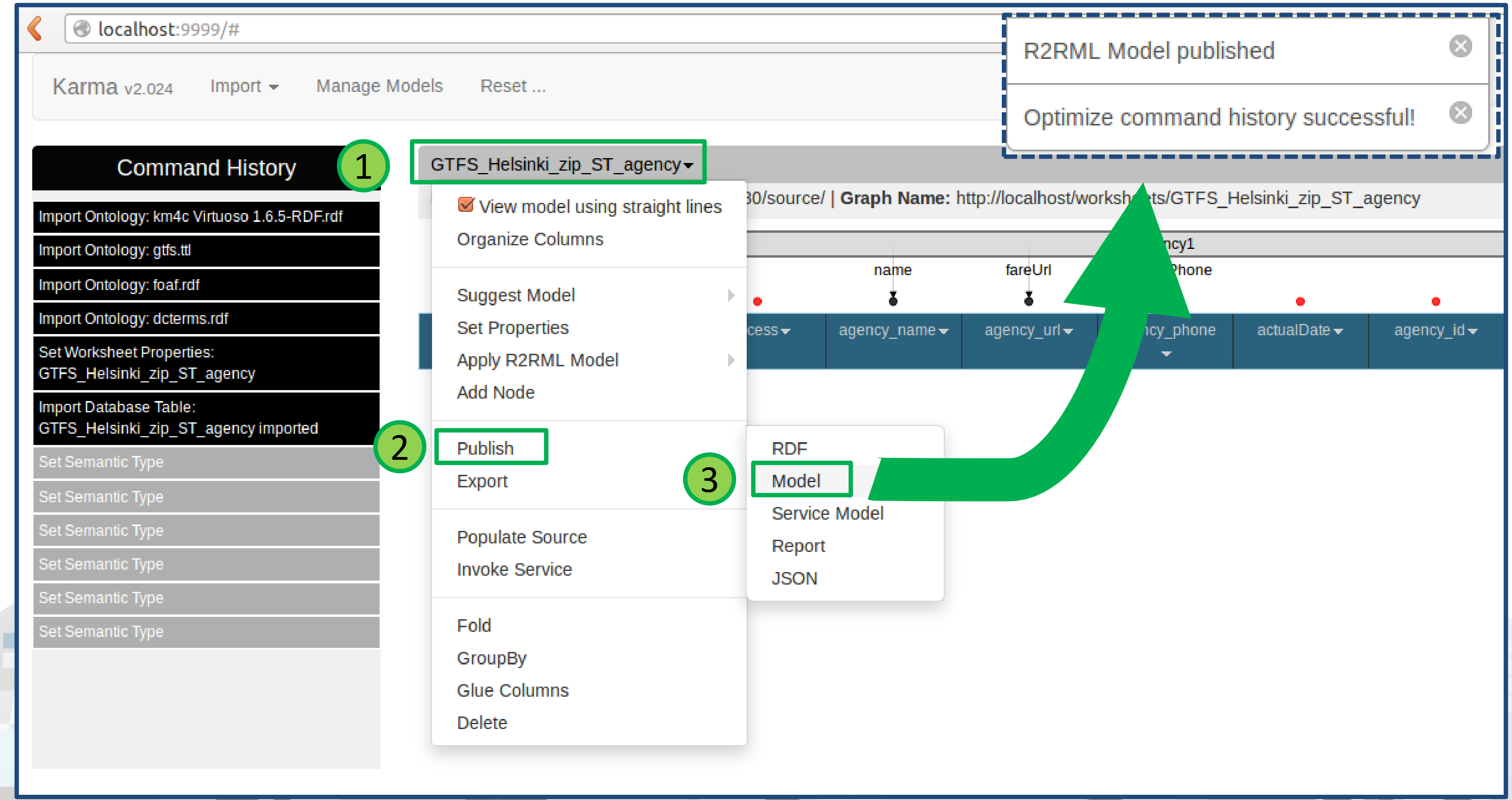
Map other properties (2)

Continue mapping:

- agency_timezone → gtfs:timeZone
- agency_name → foaf:name
- agency_url → gtfs:fareUrl
- agency_phone → km4c:agencyPhone
- agency_lang →dct:language

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org | SNAP4CITY

Export your Karma Model



localhost:9999/#

Karma v2.024 Import Manage Models Reset ...

R2RML Model published

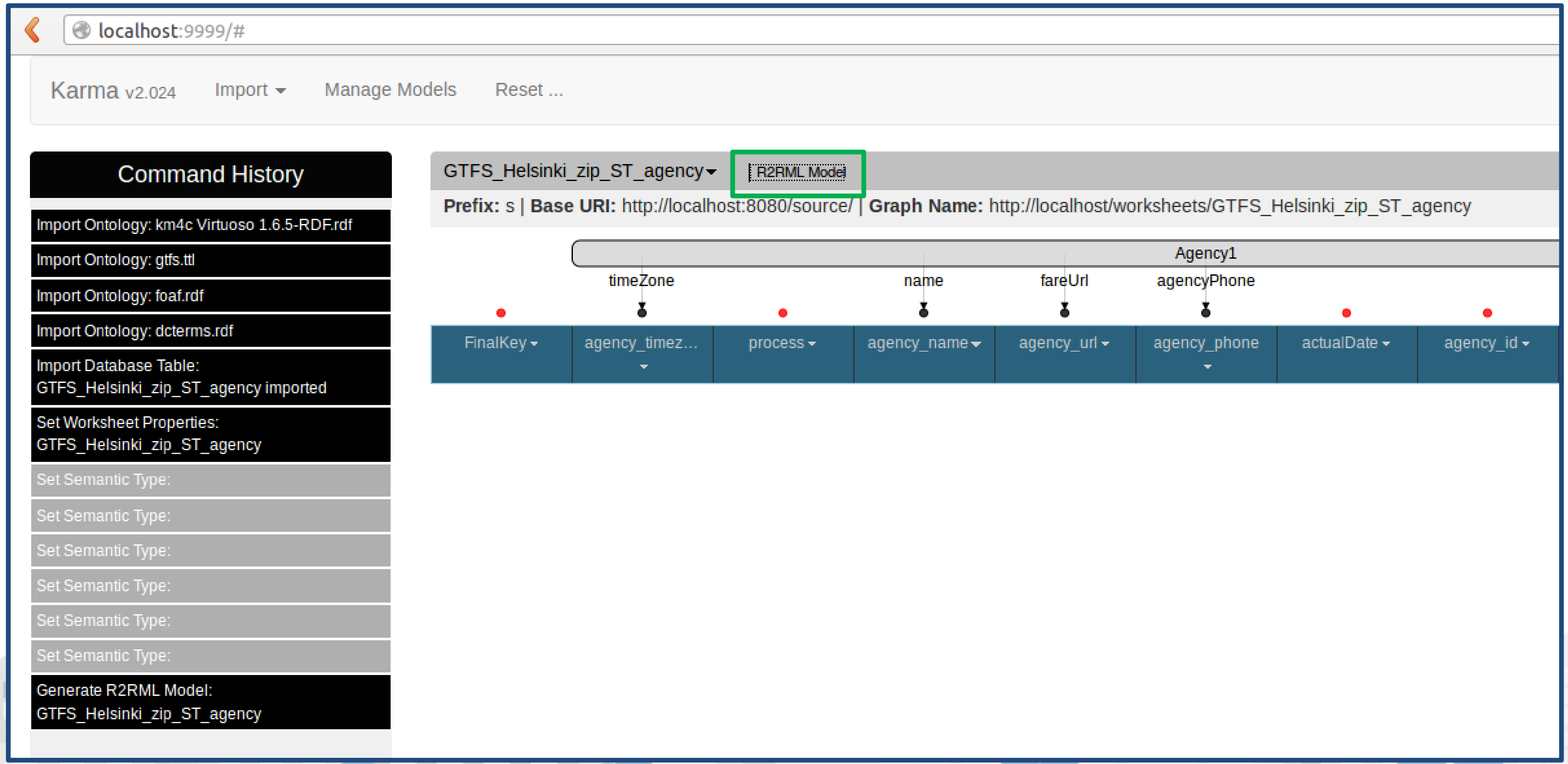
Optimize command history successful!

Command History

- Import Ontology: km4c Virtuoso 1.6.5-RDF.rdf
- Import Ontology: gtfs.ttl
- Import Ontology: foaf.rdf
- Import Ontology: dcterms.rdf
- Set Worksheet Properties: GTFS_Helsinki_zip_ST_agency
- Import Database Table: GTFS_Helsinki_zip_ST_agency imported
- Set Semantic Type
- Set Semantic Type
- Set Semantic Type
- Set Semantic Type
- Set Semantic Type
- Set Semantic Type

30/source/ | **Graph Name:** http://localhost/worksheets/GTFS_Helsinki_zip_ST_agency

Download your Karma Model



localhost:9999/#

Karma v2.024 | Import | Manage Models | Reset ...

Command History

- Import Ontology: km4c Virtuoso 1.6.5-RDF.rdf
- Import Ontology: gtfs.ttl
- Import Ontology: foaf.rdf
- Import Ontology: dcterms.rdf
- Import Database Table: GTFS_Helsinki_zip_ST_agency imported
- Set Worksheet Properties: GTFS_Helsinki_zip_ST_agency
- Set Semantic Type:
- Set Semantic Type:
- Set Semantic Type:
- Set Semantic Type:
- Set Semantic Type:
- Set Semantic Type:
- Generate R2RML Model: GTFS_Helsinki_zip_ST_agency

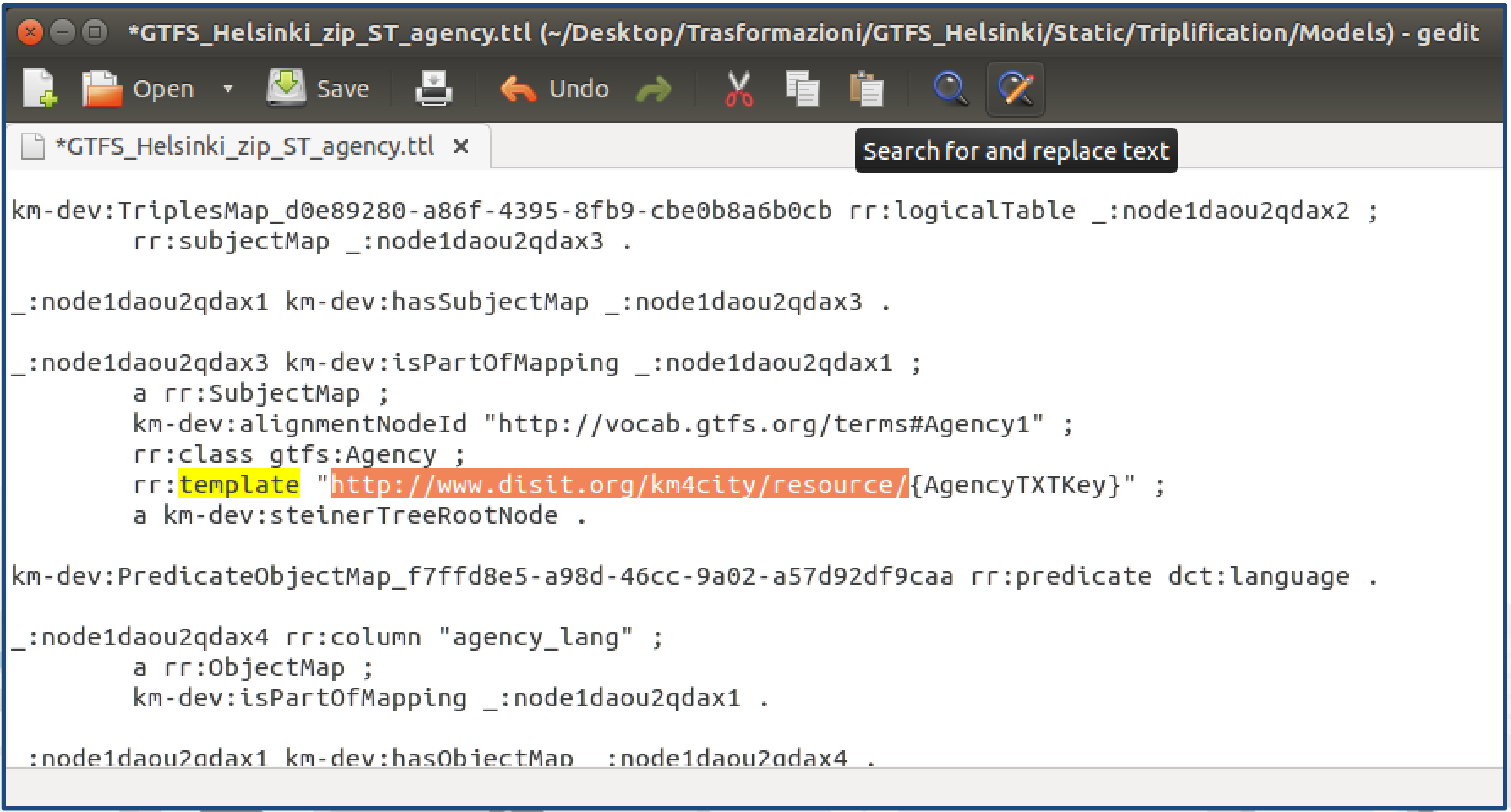
GTFS_Helsinki_zip_ST_agency | R2RML Model

Prefix: s | **Base URI:** http://localhost:8080/source/ | **Graph Name:** http://localhost/worksheets/GTFS_Helsinki_zip_ST_agency

Then **copy** the model to **/home/ubuntu/Desktop/Trasformazioni/GTFS_Helsinki/Static/Triplification/Models** and rename it as **GTFS_Helsinki_zip_ST_agency.ttl**

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org | SNAP4CITY

Manually Fix the URI template


*GTFS_Helsinki_zip_ST_agency.ttl (~/Desktop/Trasformazioni/GTFS_Helsinki/Static/Triplification/Models) - gedit
Open
Save
Undo
*GTFS_Helsinki_zip_ST_agency.ttl
Search for and replace text

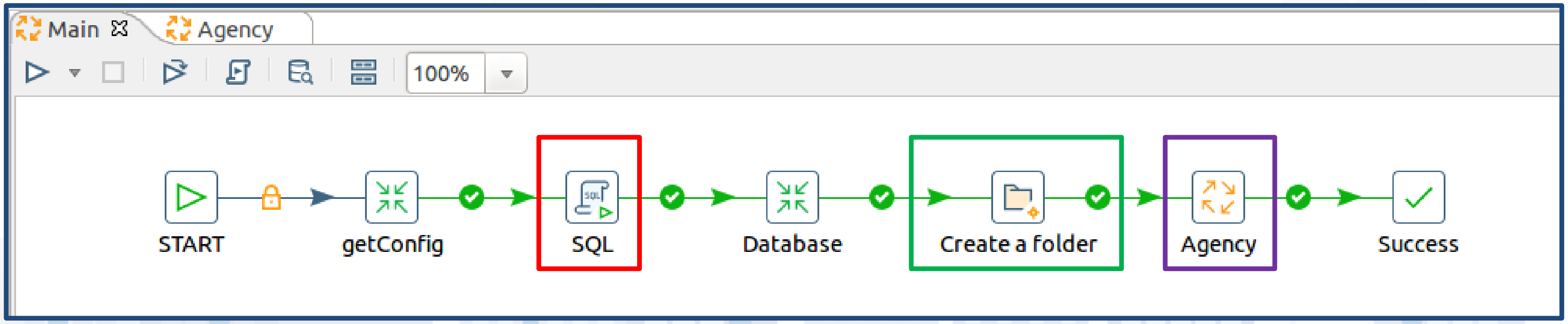
```
km-dev:TriplesMap_d0e89280-a86f-4395-8fb9-cbe0b8a6b0cb rr:logicalTable _:node1daou2qdax2 ;
        rr:subjectMap _:node1daou2qdax3 .

_:node1daou2qdax1 km-dev:hasSubjectMap _:node1daou2qdax3 .

_:node1daou2qdax3 km-dev:isPartOfMapping _:node1daou2qdax1 ;
        a rr:SubjectMap ;
        km-dev:alignmentNodeId "http://vocab.gtfs.org/terms#Agency1" ;
        rr:class gtfs:Agency ;
        rr:template "http://www.disit.org/km4city/resource/{AgencyTXTKey}" ;
        a km-dev:steinerTreeRootNode .

km-dev:PredicateObjectMap_f7ffd8e5-a98d-46cc-9a02-a57d92df9caa rr:predicate dct:language .

_:node1daou2qdax4 rr:column "agency_lang" ;
        a rr:ObjectMap ;
        km-dev:isPartOfMapping _:node1daou2qdax1 .

_:node1daou2qdax1 km-dev:hasObjectMap _:node1daou2qdax4 .
```

Static Triplification

Main job

Create new job named **Main.kjb** in folder:
/home/ubuntu/Desktop/Trasformazioni/GTFS_Helsinki/Static/Triplification

Jobs and transformations required by the job also have to be put there, apart from getConfig

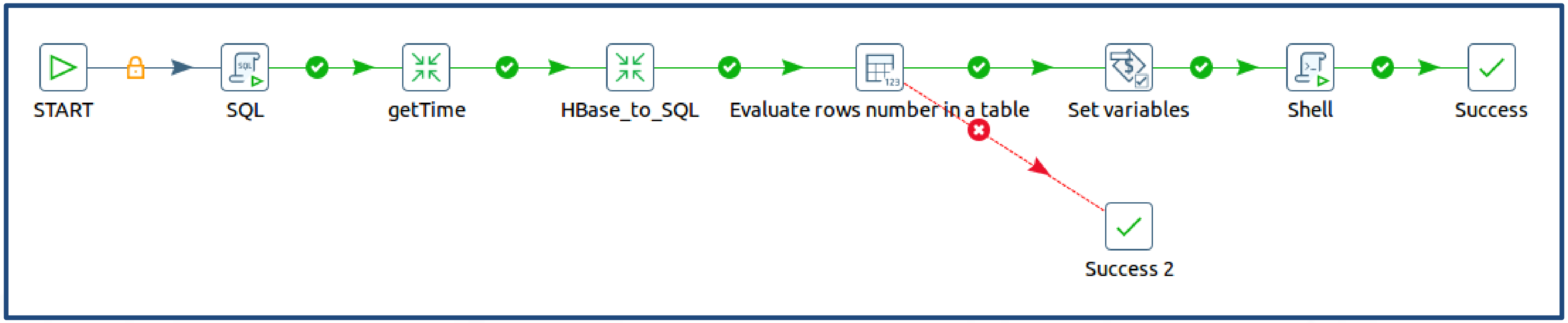


DROP TABLE IF EXISTS `${processName}_agency`;

${TRIPLESDESTDIRECT}/${CATEGORY}/${processName}/${ACTUALYEARMONTH}/${ACTUALDAY}/${ACTUALHOURS}/${ACTUALMINSEC}

New empty job

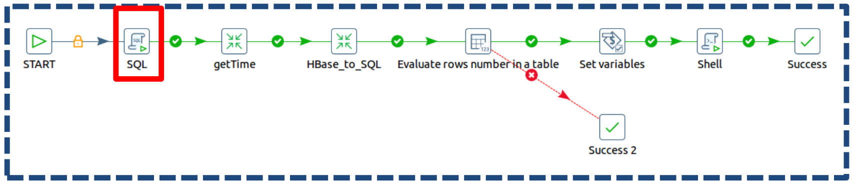
Agency: Outline



Populate the **Agency** job as in the above image.

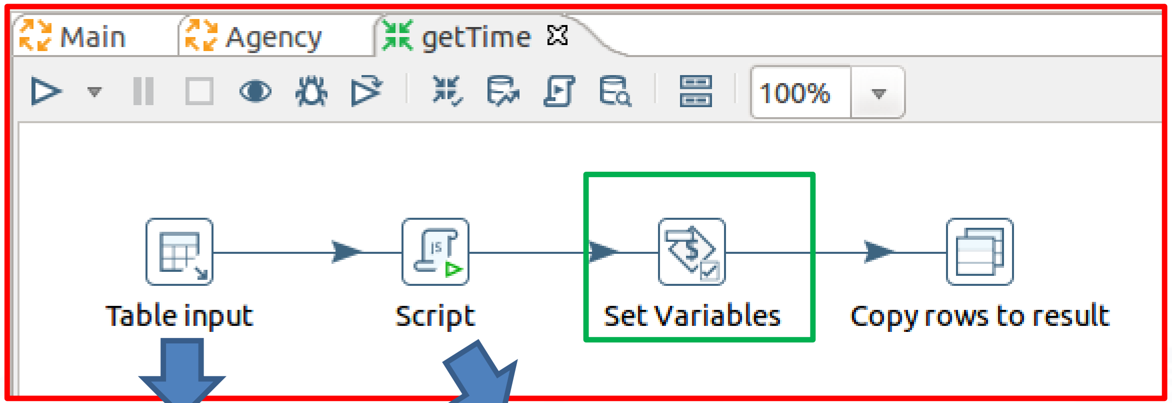
Leave steps unconfigured, we will see how to configure them later.

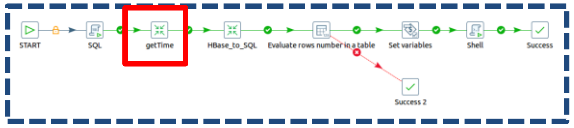
Agency: SQL



```
CREATE TABLE ${processName}_agency
(
FinalKey text DEFAULT NULL,
agency_timezone text DEFAULT NULL,
process text DEFAULT NULL,
agency_name text DEFAULT NULL,
agency_url text DEFAULT NULL,
agency_phone text DEFAULT NULL,
actualDate text DEFAULT NULL,
agency_id text DEFAULT NULL,
AgencyTXTKey text DEFAULT NULL,
agency_lang text DEFAULT NULL,
timestamp text DEFAULT NULL
);
```

Agency: getTime

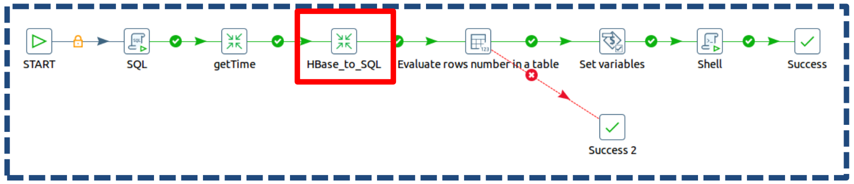


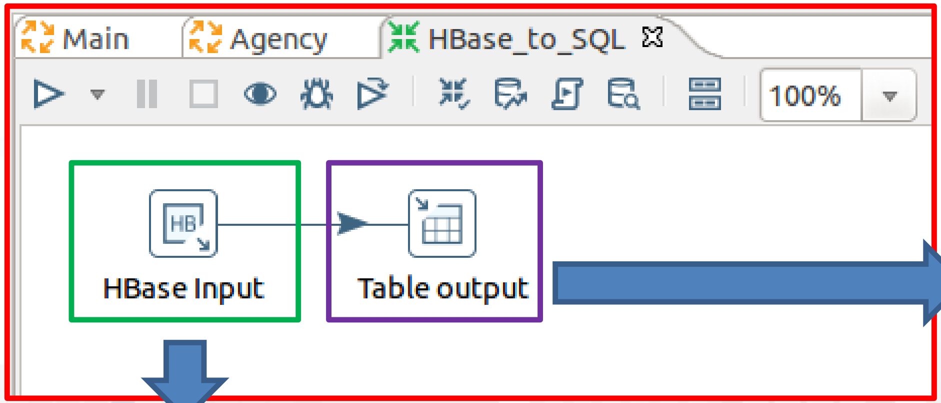


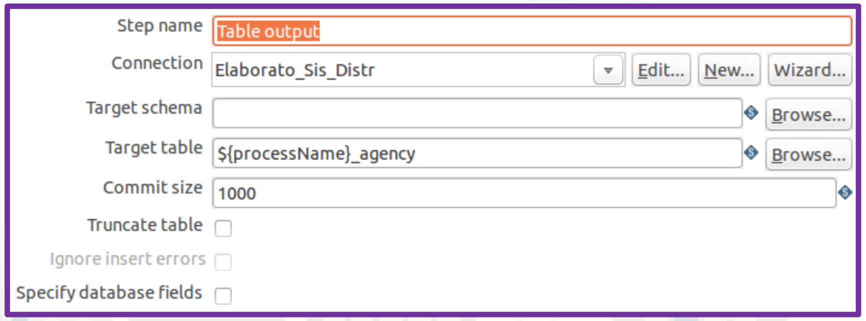
```
SELECT
last_triples
FROM
process_manager2
WHERE
process='${processName}'
```

```
var timestamp_LT = new Date(1970, 1, 1, 12, 0, 0, 0);
timestamp_LT = timestamp_LT.getTime();
var date = last_triples;
if(date != null) {
var anno = date.slice(0,4);
var mese = ((date.slice(5,7))-1);
var giorno = date.slice(8,10);
var ora = date.slice(11,13);
var minuti = date.slice(14,16);
var secondi = date.slice(17,19);
var millisec = date.slice(20,23);
var date_tmp = new Date(anno, mese, giorno, ora, minuti, secondi, millisec);
timestamp_LT = date_tmp.getTime();
}
```

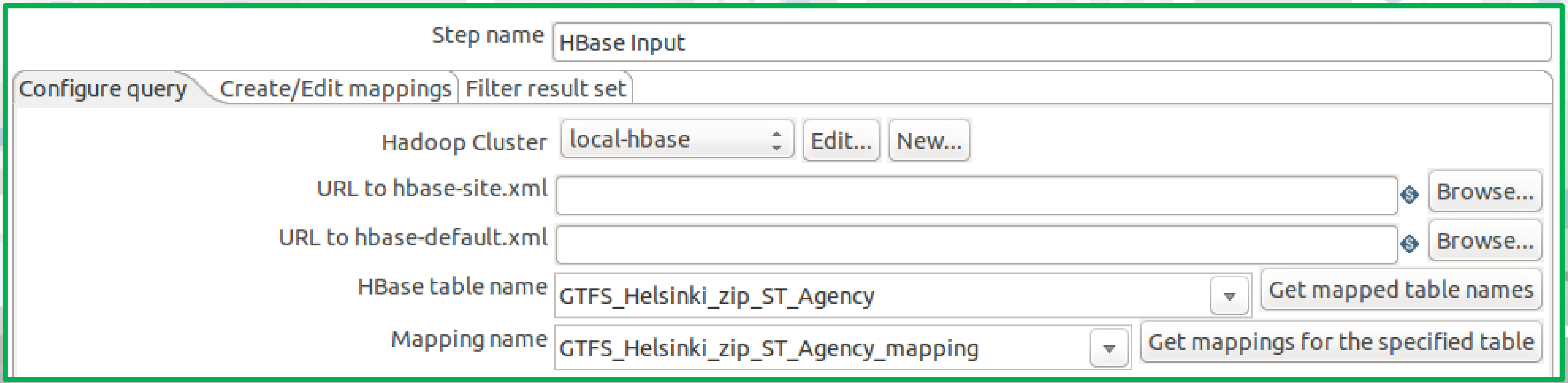
Agency: HBase_to_SQL







Step name	Table output
Connection	Elaborato_Sis_Distr
Target schema	
Target table	${processName}_agency
Commit size	1000
Truncate table	☐
Ignore insert errors	☐
Specify database fields	☐



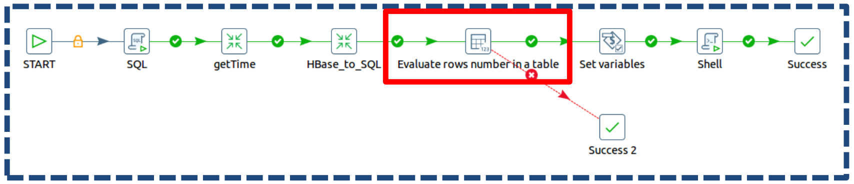
Step name: HBase Input

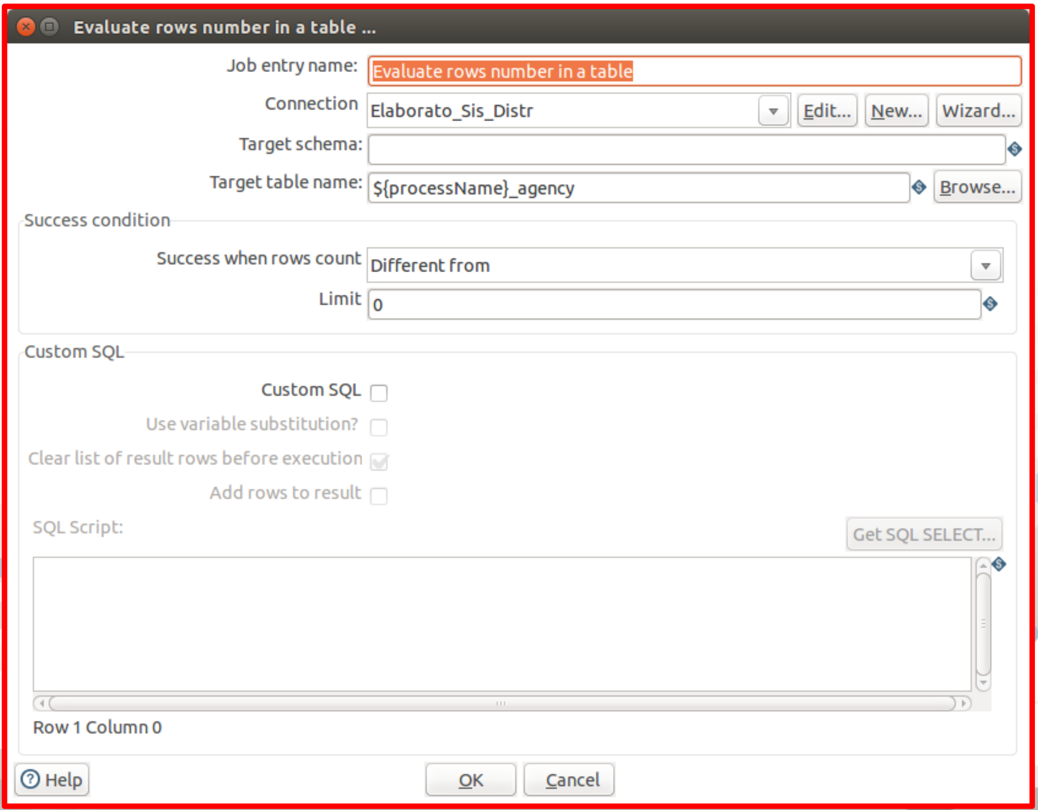
Configure query | Create/Edit mappings | Filter result set

Hadoop Cluster	local-hbase
URL to hbase-site.xml	
URL to hbase-default.xml	
HBase table name	GTFS_Helsinki_zip_ST_Agency
Mapping name	GTFS_Helsinki_zip_ST_Agency_mapping

UNIVERSITÀ DEGLI STUDI FIRENZE
DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB
http://www.disit.org

Agency: Evaluate rows




Evaluate rows number in a table ...

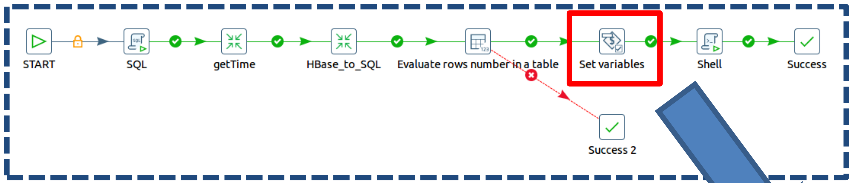
Job entry name: Evaluate rows number in a table
Connection: Elaborato_Sis_Distr
Target schema:
Target table name: ${processName}_agency

Success condition
Success when rows count: Different from
Limit: 0

Custom SQL
Custom SQL ☐
Use variable substitution? ☐
Clear list of result rows before execution ☑
Add rows to result ☐
SQL Script:

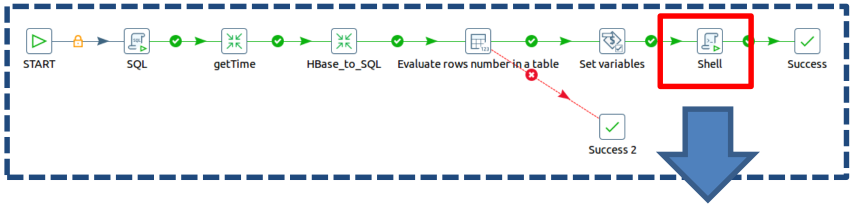
Row 1 Column 0

Agency: Set variables



Set **MODELPATH** = /home/ubuntu/Desktop/Trasformazioni/GTFS_Helsinki/Static/Triplification/Models

Agency: Shell



Working directory:
/home/ubuntu/programs/Web-Karma-master/karma-offline

Script:
mvn exec:java -Dexec.mainClass="edu.isi.karma.rdf.OfflineRdfGenerator" -Dexec.args=" --sourcetype DB --modelfilepath "${MODELPATH}/${processName}_agency.ttl" --outputfile ${TRIPLESDESTDIRECT}/${CATEGORY}/${processName}/${ACTUALYEARMONTH}/${ACTUALDAY}/${ACTUALHOURS}/${ACTUALMINSEC}/agency.n3 --dbtype MySQL --hostname ${IPADDRESSMASTER} --username ${USERNAMEMYSQL} --password ${PSWMYSQL} --portnumber ${PORTMYSQL} --dbname ${DATABASEMYSQL} --tablename ${processName}_agency" -Dexec.classpathScope=compile

Homework (optional)

What have I missed to do?

Small tips:

- There's something missing at the end of the Agency job
- There's something missing in the configuration of HBase input in Hbase_to_SQL

Bigger tip:

- The two missings are related to each other

Last tip:

- The two missings are such that it comes to be untidy to read the datetime of last triplification from the process_manager2 table in getTime transformation