Pentaho Data Integration (PDI) - Kettle – guide

I. Creating new transformation

II. Adding operations (steps) into transformation.

a. Data input steps

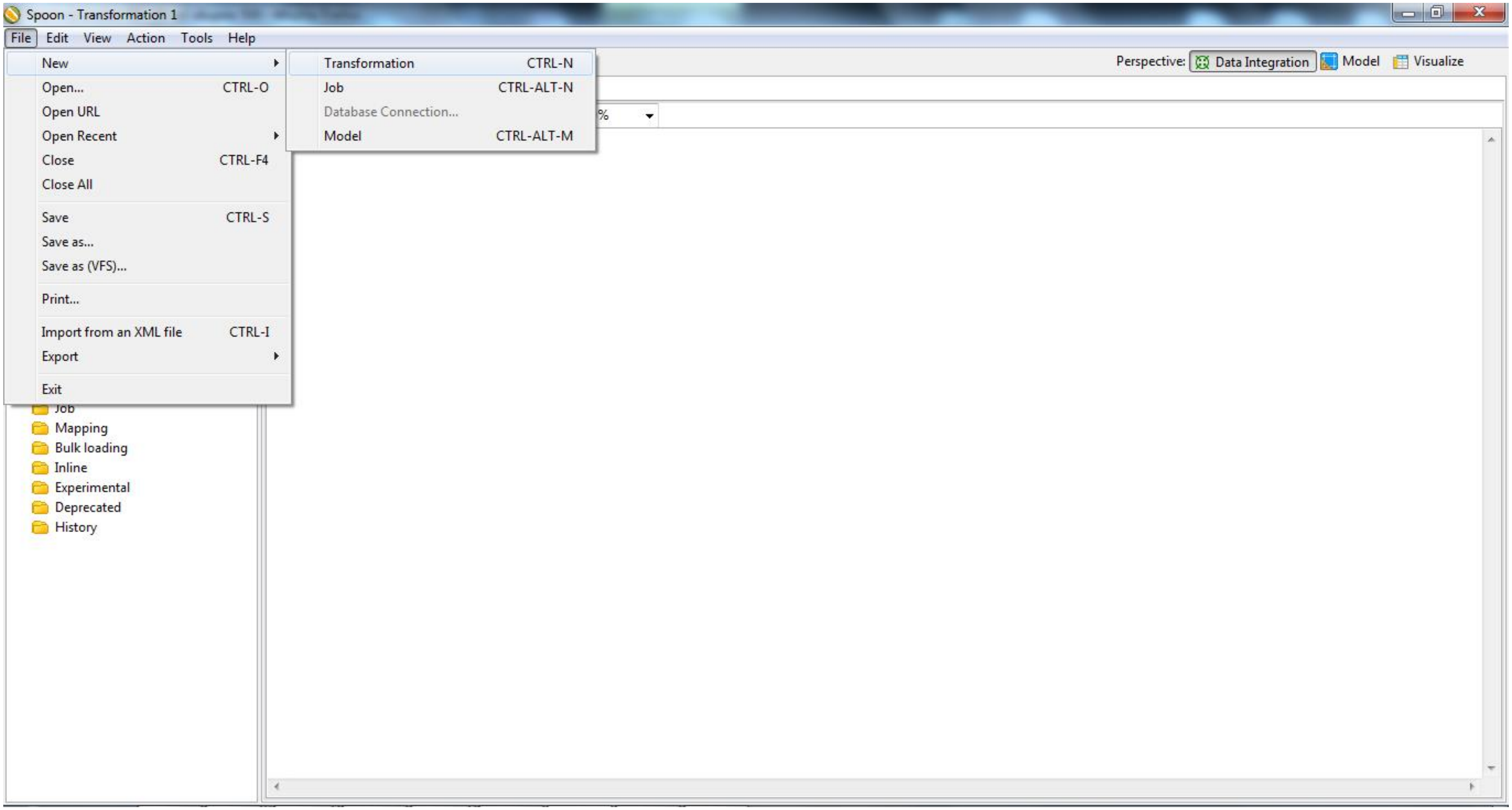
- *Table Input*
- *Get data from XML*
- *Microsoft Excel Input*

b. Transformation steps

- *Sort Rows*
- *Select values*
- *Filter rows*
- *Modified Java Script Value*
- *Join Rows*
- Group By

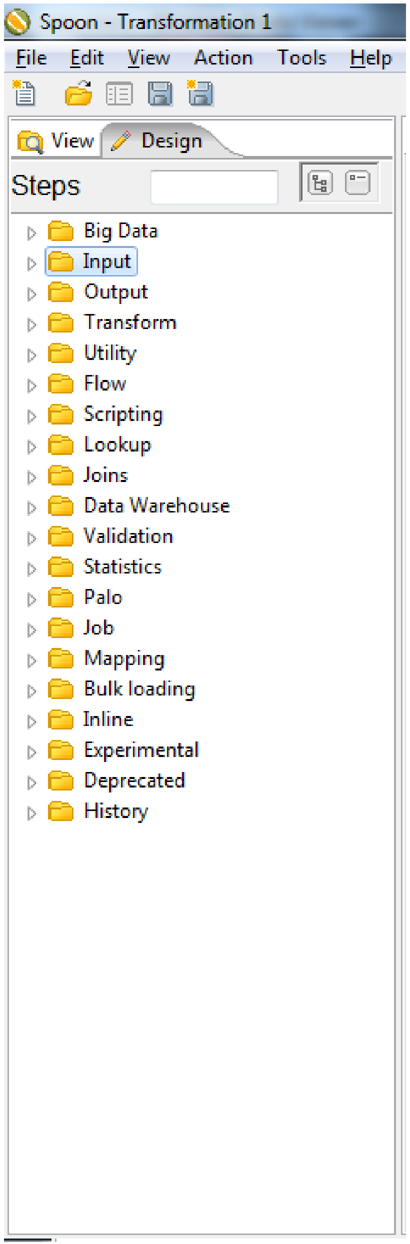
III. Creating edges between operations (hops)

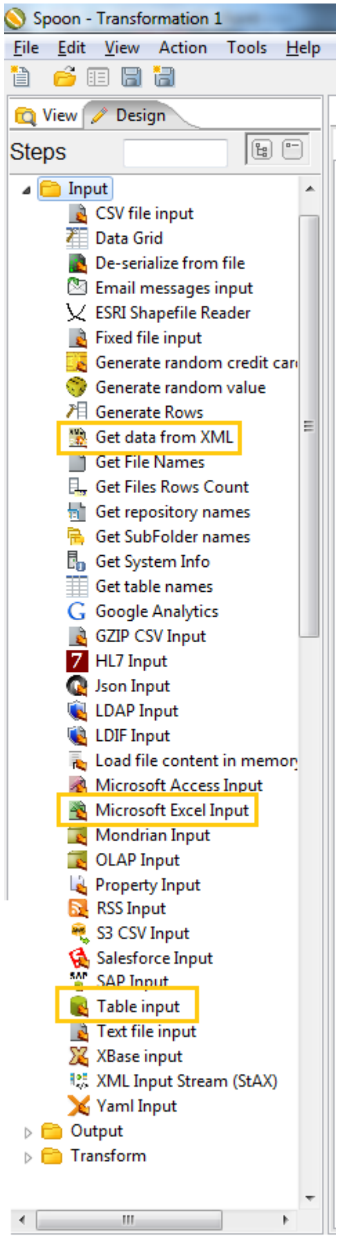
I. Creating new transformation

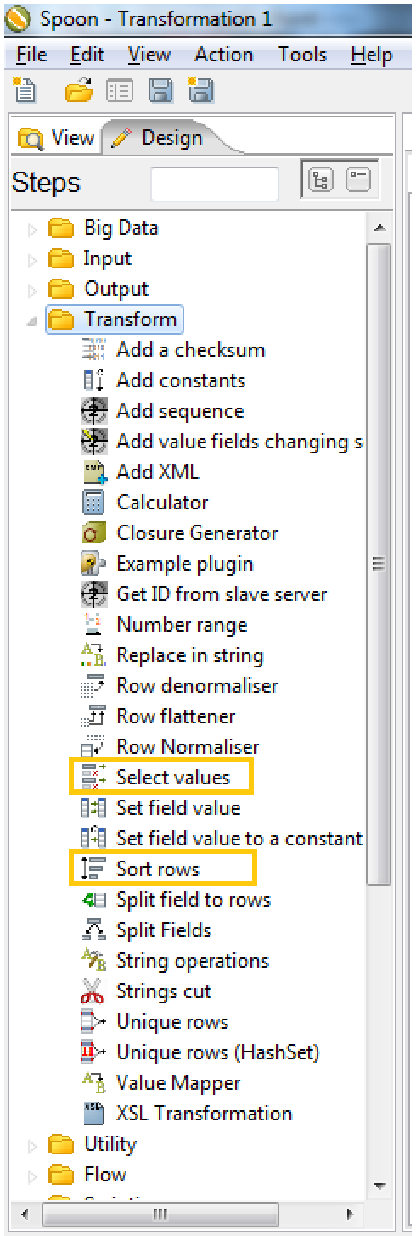


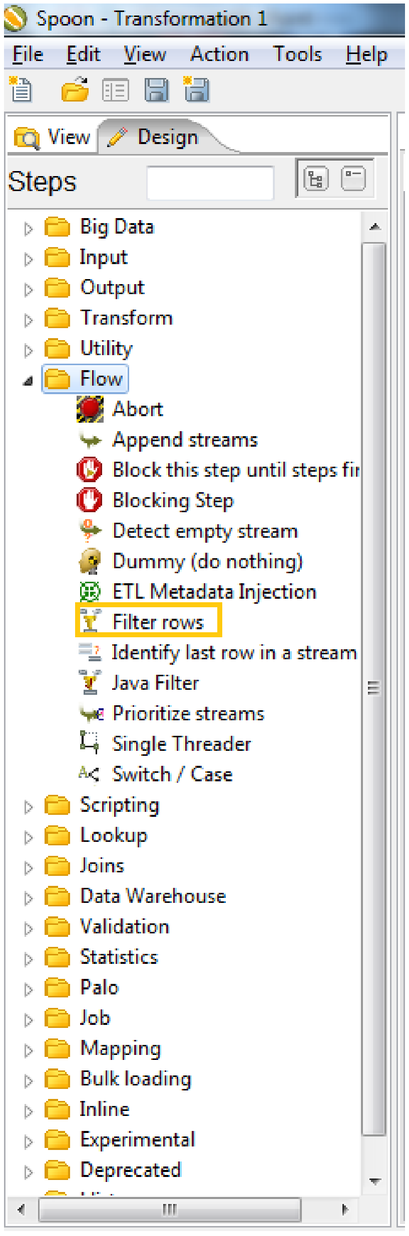
II. Adding operations (steps) into transformation.

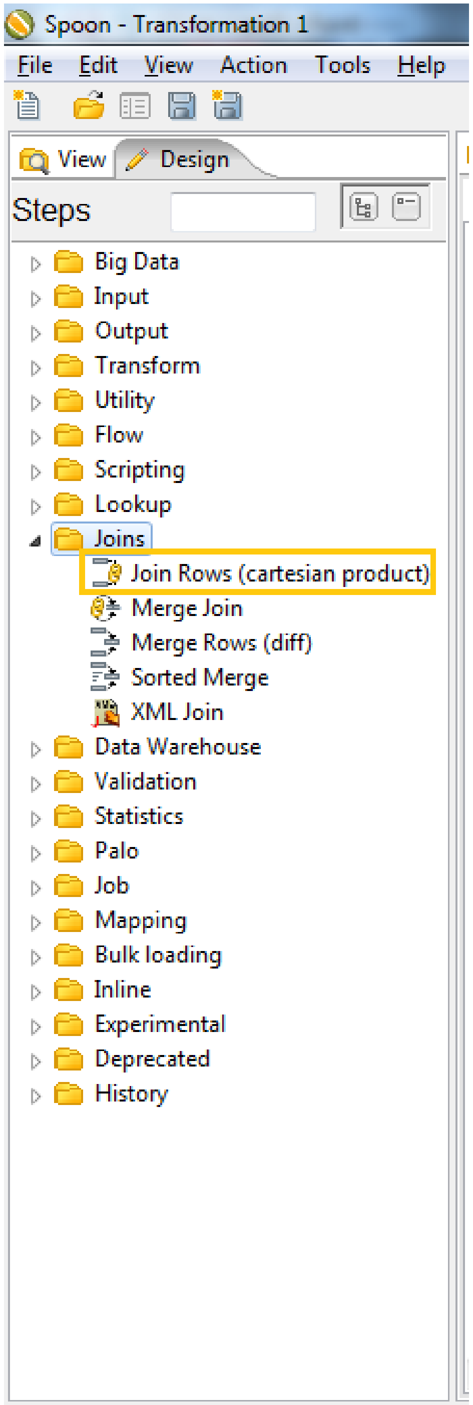
After you create new transformation or open the existing one, in the tab “Design” you may find the complete palette of the operations supported by Kettle. The operations are divided into different groups and they are easily added to an ETL design by dragging them from the palette into the transformation canvas. In the following figures, you may see different operations in the Spoon design palette. The ones that should be used for this round of the case study are marked with orange rectangle and further explained.

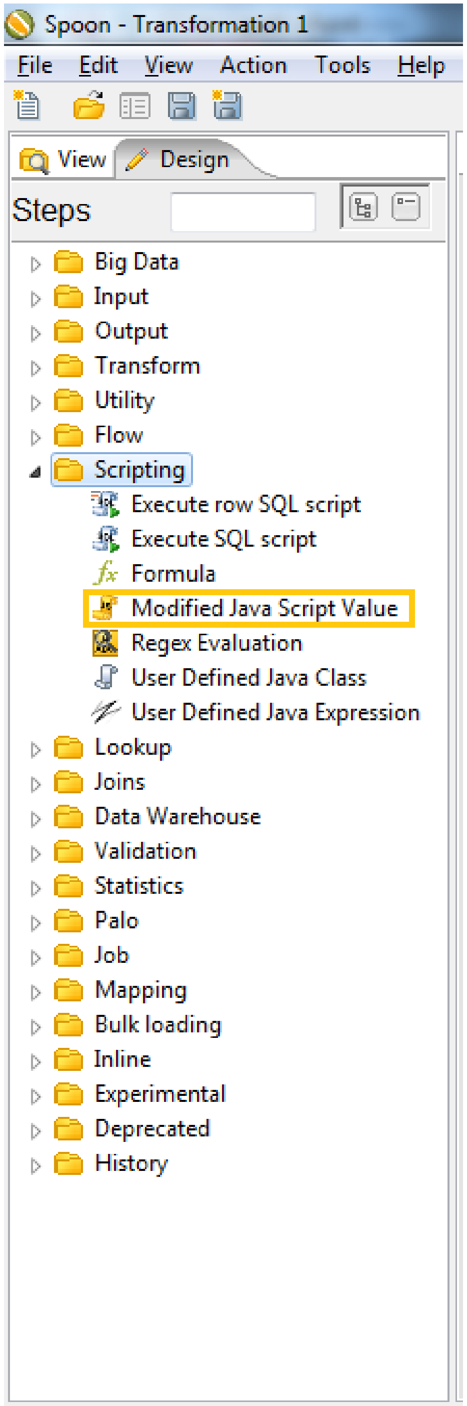


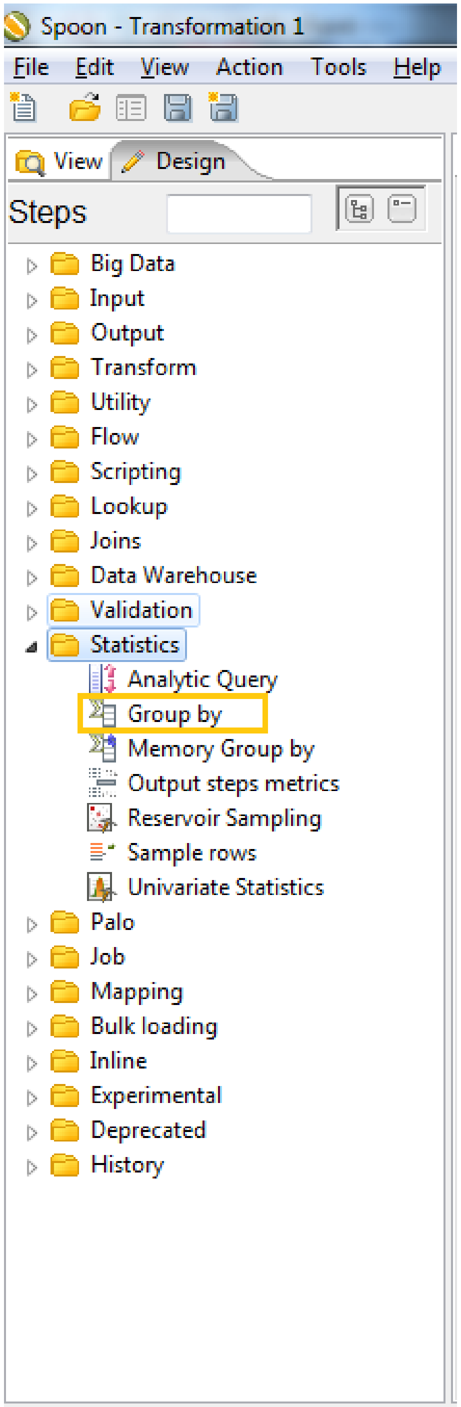







Spoon - Transformation 1
File Edit View Action Tools Help
View
Design
Steps
Big Data
Input
Output
Transform
Utility
Flow
Scripting
Lookup
Joins
Join Rows (cartesian product)
Merge Join
Merge Rows (diff)
Sorted Merge
XML Join
Data Warehouse
Validation
Statistics
Palo
Job
Mapping
Bulk loading
Inline
Experimental
Deprecated
History


Spoon - Transformation 1
File Edit View Action Tools Help
View
Design
Steps
Big Data
Input
Output
Transform
Utility
Flow
Scripting
Execute row SQL script
Execute SQL script
Formula
Modified Java Script Value
Regex Evaluation
User Defined Java Class
User Defined Java Expression
Lookup
Joins
Data Warehouse
Validation
Statistics
Palo
Job
Mapping
Bulk loading
Inline
Experimental
Deprecated
History


Spoon - Transformation 1
File Edit View Action Tools Help
View
Design
Steps
Big Data
Input
Output
Transform
Utility
Flow
Scripting
Lookup
Joins
Data Warehouse
Validation
Statistics
Analytic Query
Group by
Memory Group by
Output steps metrics
Reservoir Sampling
Sample rows
Univariate Statistics
Palo
Job
Mapping
Bulk loading
Inline
Experimental
Deprecated
History

a) Data input steps

Table Input 	*This step is used to read information from a database, using a connection and SQL. Basic SQL statements are generated automatically.*
Options	**Description**
Step name	Name of the step;the name has to be unique in a single transformation
Connection	The database connection from which to read data

SQL	The SQL statement used to read information from the database connection. You can also click Get SQL select statement... to browse tables and automatically generate a basic select statement.
	For example to read the data about assessment item:
	`SELECT distinct`
	`dificultat, course, id`
	`FROM`
	`t_questions`

Enable lazy conversion	When enables, lazy conversion avoids unnecessary data type conversions and can result in a significant performance improvements.
Replace variables in script?	Enable to replace variables in the script; this feature was provided to allow you to test with or without performing variable substitutions.
Insert data from step	Specify the input step name where Pentaho? an expect information to come from. This information can then be inserted into the SQL statement. The locators where Pentaho? inserts information is indicated by ? (question marks).
Execute for each row?	Enable to perform the data insert for each individual row.
Limit size	Sets the number of lines that is read from the database; zero (0) means read all lines.

For example, to read the data assessment items from the database you should parameterize the Table Input step like shown in the figure below. First you need to define the database connection providing the type (DBMS) and the parameters needed to enable reading data from that database. Then you can either manually write the SQL select statement to read the data from the defined database in the "SQL" text area, or you can automatically obtain it by choosing "Get SQL select statement" and exploring the database.

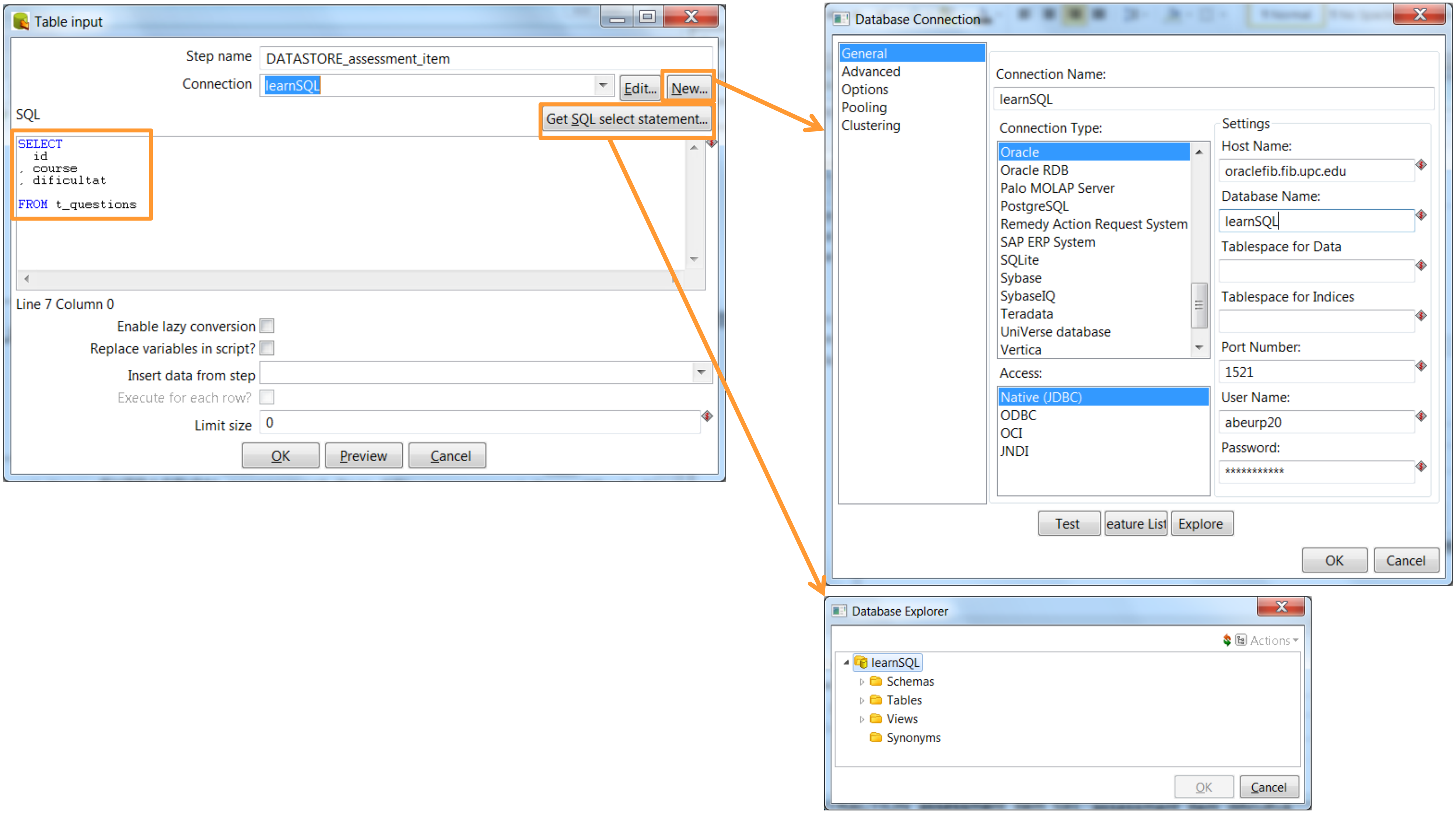


Table input

Step name: DATASTORE_assessment_item

Connection: learnSQL

SQL

```
SELECT
   id
,  course
,  dificultat

FROM t_questions
```

Line 7 Column 0

Enable lazy conversion □

Replace variables in script? □

Insert data from step:

Execute for each row? □

Limit size: 0

Database Connection

Connection Name: learnSQL

Connection Type:

- Oracle
- Oracle RDB
- Palo MOLAP Server
- PostgreSQL
- Remedy Action Request System
- SAP ERP System
- SQLite
- Sybase
- SybaseIQ
- Teradata
- UniVerse database
- Vertica

Access:

- Native (JDBC)
- ODBC
- OCI
- JNDI

Settings

Host Name: oraclefib.fib.upc.edu

Database Name: learnSQL

Tablespace for Data:

Tablespace for Indices:

Port Number: 1521

User Name: abeurp20

Password: ***********

Database Explorer

- learnSQL
  - Schemas
  - Tables
  - Views
  - Synonyms

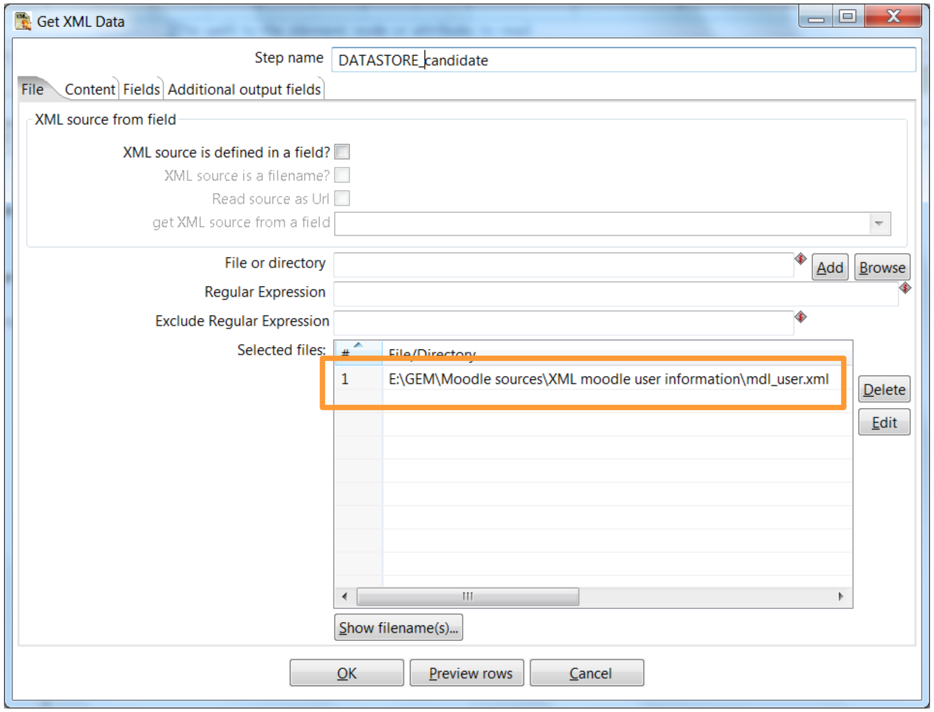
Get data from XML Get data from XML	*This step provides the ability to read data from any type of XML file using XPath specifications.*
FILES TAB	
Options	**Description**
Step Name	Name of the step; the name has to be unique in a single transformation.
XML Source from field	• XML source is defined in a field : the previous step is giving XML data in a certain field in the input stream. • XML source is a filename : the previous step is giving filenames in a certain field in the input stream. These are read. • Read source as URL : the previous step is giving URLs in a certain field in the input stream. These are read. • Get XML source from a field : specify the field to read XML, filename or URL from.
File or directory	Specifies the location and/or name of the input text file. ***Note***: Click Add to add the file/directory/wildcard combination to the list of selected files (grid) below.
Regular expression	Specifies the regular expression you want to use to select the files in the directory specified in the previous option.
Selected Files	Contains a list of selected files (or wildcard selections) and a property specifying if file is required or not. If a file is required and it is not found, an error is generated; otherwise, the file name is skipped.
Show filename(s)...	Displays a list of all files that will be loaded based on the current selected file definitions
Step Name	Name of the step; the name has to be unique in a single transformation.
XML Source from field	• XML source is defined in a field : the previous step is giving XML data in a certain field in the input stream. • XML source is a filename : the previous step is giving filenames in a certain field in the input stream. These are read. • Read source as URL : the previous step is giving URLs in a certain field in the input stream. These are read. • Get XML source from a field : specify the field to read XML, filename or URL from.
CONTENT TAB	
Option	**Description**
Settings	• Loop XPath : For every "Loop XPath" location we find in the XML file(s), we will output one row of data. This is the main specification we use to flatten the XML file(s). You can use the "Get XPath nodes" button to

	search for the possible repeating nodes in the XML document. Please note that if the XML document is large that this can take a while. • Encoding : the XML filename encoding in case none is specified in the XML documents. (yes, those still exist) • Namespace aware : check this to make the XML document namespace aware. • Ignore comments : Ignore all comments in the XML document while parsing. • Validate XML : Validate the XML prior to parsing. Use a token when you want to replace dynamically in a Xpath field value. A token is between @_ and - (@_fieldname-). Please see the Example 1 to see how it works. • Use token : a token is not related tro XML parsing but to PDI. • Igore empty file : an empty file is not a valid XML document. Check this if you want to ignore those altogether. • Do not raise an error if no file: Don't raise a stink if no files are found. • Limit : Limits the number of rows to this number (zero (0) means all rows). • Prune path to handle large files: almost the same value as the "Loop XPath" property with some exceptions, see Get Data from XML - Handling Large Files for more details. Note that you can use this parameter to avoid multiple HTTP URL requests.
Additional fields	• Include filename in output? : Allows you to specify a field name to include the file name (String) in the output of this step. • Rownum in output? : Allows you to specify a field name to include the row number (Integer) in the output of this step.
Add to result filename	• Add files to result filename : Adds the XML filenames read to the result of this transformation. A unique list is being kept in memory that can be used in the next job entry in a job, for example in another transformation.
FIELDS	

Option	Description
Name	The name of the output field

XPath	The path to the element node or attribute to read
Element	The element type to read: Node or Attribute
Type	The data type to convert to
Format	The format or conversion mask to use in the data type conversion
Length	The length of the output data type
Precision	The precision of the output data type
Currency	The currency symbol to use during data type conversion
Decimal	The numeric decimal symbol to use during data type conversion
Group	The numeric grouping symbol to use during data type conversion
Trim type	The type of trimming to use during data type conversion
Repeat	Repeat the column value of the previous row if the column value is empty (null)

For example, to read the data about the candidates from the provided xml files you should parameterize the Get data from XLM step like shown in the figures below. First you need to define the filepath to the XML file, then the XPath from where inside the XML file you want to read the data (or choose it from automaticallyobtained XPath nodes – "Get XPath nodes") and finally you can either define the fields to be read or you can automatically obtain them by choosing the option "Get fields".


Get XML Data

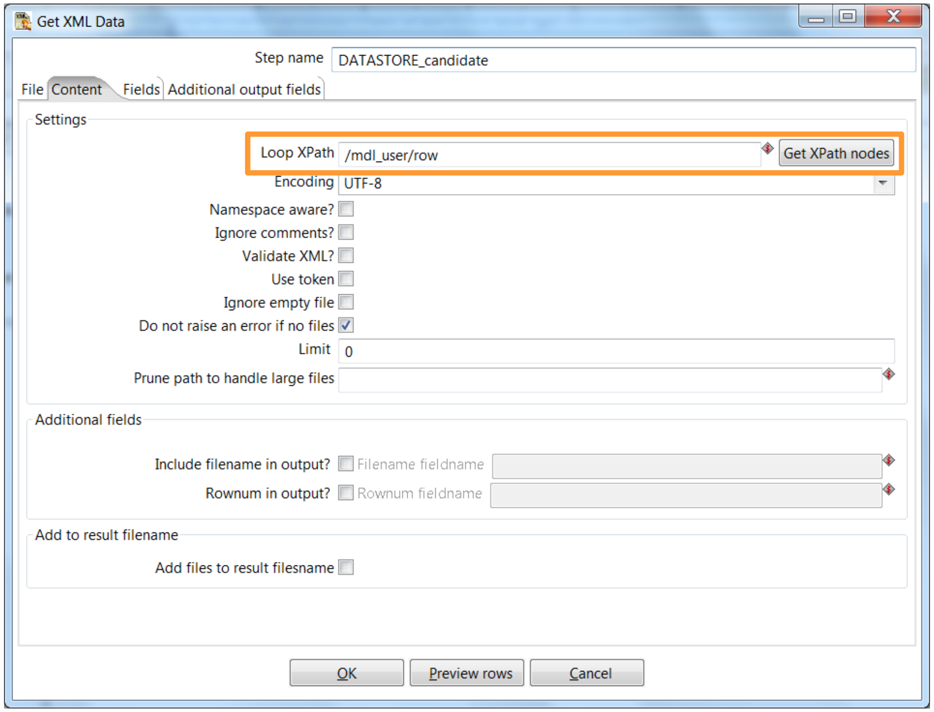
Step name: DATASTORE_candidate

XML source from field

XML source is defined in a field? □
XML source is a filename? □
Read source as Url □
get XML source from a field

File or directory
Regular Expression
Exclude Regular Expression
Selected files:

#	File/Directory
1	E:\GEM\Moodle sources\XML moodle user information\mdl_user.xml


Get XML Data

Step name: DATASTORE_candidate

Settings

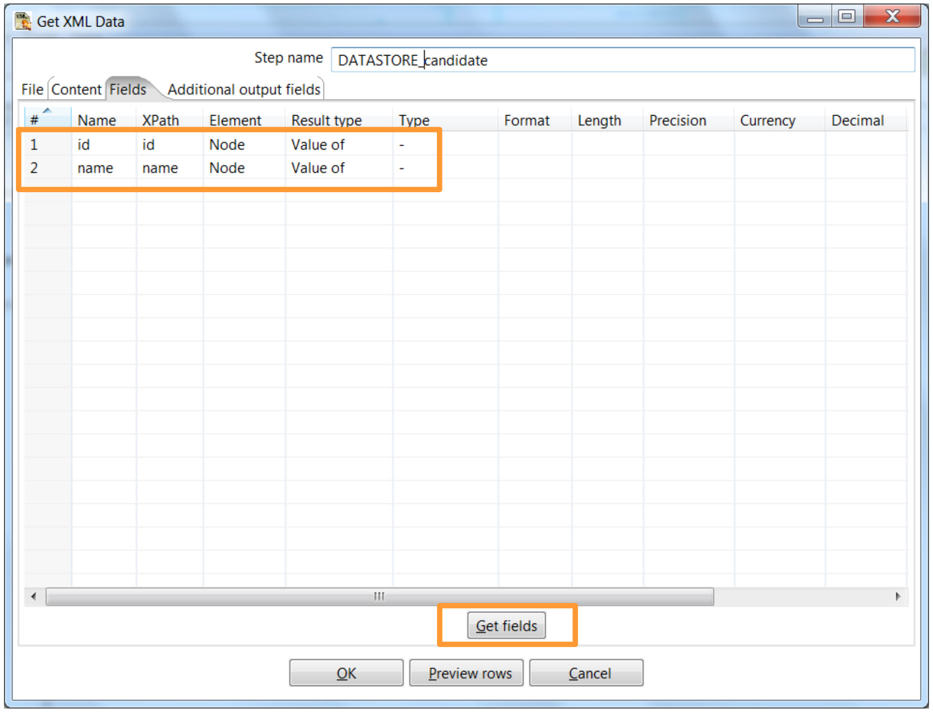
Loop XPath: /mdl_user/row
Encoding: UTF-8
Namespace aware? □
Ignore comments? □
Validate XML? □
Use token □
Ignore empty file □
Do not raise an error if no files ☑
Limit: 0
Prune path to handle large files

Additional fields

Include filename in output? □ Filename fieldname
Rownum in output? □ Rownum fieldname

Add to result filename

Add files to result filesname □



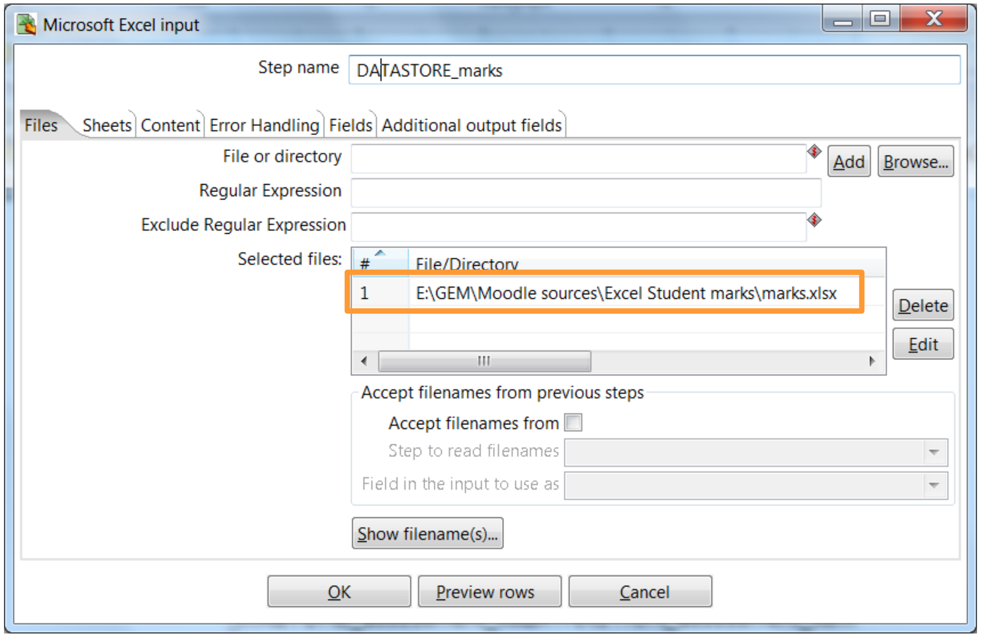
Step name DATASTORE_candidate

#	Name	XPath	Element	Result type	Type	Format	Length	Precision	Currency	Decimal
1	id	id	Node	Value of	-					
2	name	name	Node	Value of	-					

Microsoft Excel Input Microsoft Excel Input	*The Excel Input step provides you with the ability to read data from one or more Excel and OpenOffice files. The following sections describe each of the available features for configuring the Excel Input step.*
FILES TAB	
Option	**Description**
Step Name	Name of the step; the name has to be unique in a single transformation.
File or directory	Specifies the location and/or name of the input text file. ***Note***: Click Add to add the file/directory/wildcard combination to the list of selected files (grid) below.
Regular expression	Specifies the regular expression you want to use to select the files in the directory specified in the previous option.
Selected Files	Contains a list of selected files (or wildcard selections) and a property specifying if file is required or not. If a file is required and it is not found, an error is generated;otherwise, the file name is skipped.
Accept filenames from previous steps	Allows you to read in file names from a previous step in the transformation
Show filenames(s)...	Displays a list of all files that will be loaded based on the current selected file definitions
Preview rows	Click Preview to examine the contents of the specified Excel file
Option	Description
CONTENT	
Header	Enable if the sheets specified contain a header row to skip
No empty rows	Enable if you don't want empty rows in the output of this step
Stop on empty row	Makes the step stop reading the current sheet of a file when a empty line is encountered
Filename field	Specifies a field name to include the file name in the output of this step.
Sheetname field	Specifies a field name to include the sheet name in the output of this step.
Sheer row nr field	Specifies a field name to include the sheet row number in the output of the step. The sheet row number is the actual row number in the Excel sheet.
Row nrwritten field	Specifies a field name to include the row number in the output of the step. "Row number written" is the number of rows processed, starting at 1 and counting indefinitely

Limit	Limits the number of rows to this number (zero (0) means all rows).
Encoding	Specifies the character encoding (such as UTF-8, ASCII)
Spread sheet type (engine)	This field allows you to specify the spreadsheet type. (since version 4.1.0) Currently the following are supported: - Excel 97-2003 XLS: this is the default, backward compatible type provided for by the JXL software backend. - Excel 2007 XLSX: If you select this spread sheet type you can read all known Excel file types. Functionality provided by the Apache POI project. - Open Office ODS: By selecting this type you can read OpenOffice spreadsheet using the ODFDOM engine. The default Spread sheet type (engine) is set to Excel 97-2003 XLS. When you are reading other file types like OpenOffice, ODS, Excel 2007 and using special functions like protected worksheets, you need to change the Spread sheet type (engine) in the Content tab accordingly.

For example, to read the data about the candidates marks from the provided excel sheets you should parameterize the Microsoft Excel Input step like shown in the figures below. First you need to define the filepath to the Excel file, then the name of the sheet inside the excel file and the indices of row/column from where you want to start reading the data. In the case the headers for the tables are available in the sheet you can check the option "Header" and you can automatically obtain the fields from the sheet by choosing the option "Get fields from header row…", or you can manually define then fields and their names.



Step name: DATASTORE_marks

File or directory:

Regular Expression:

Exclude Regular Expression:

Selected files:

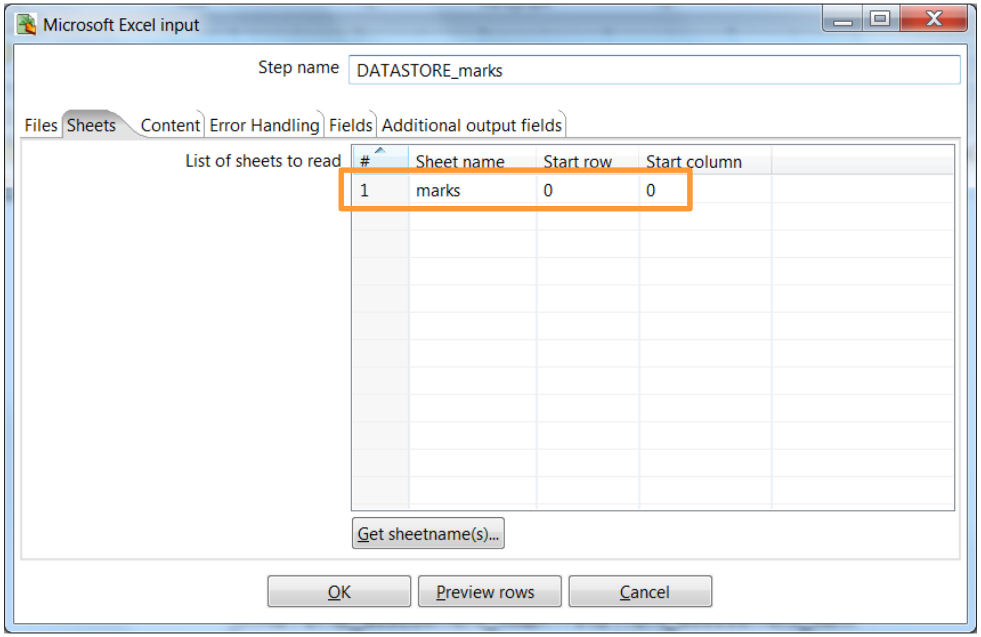
#	File/Directory
1	E:\GEM\Moodle sources\Excel Student marks\marks.xlsx

Accept filenames from previous steps

Accept filenames from ☐

Step to read filenames

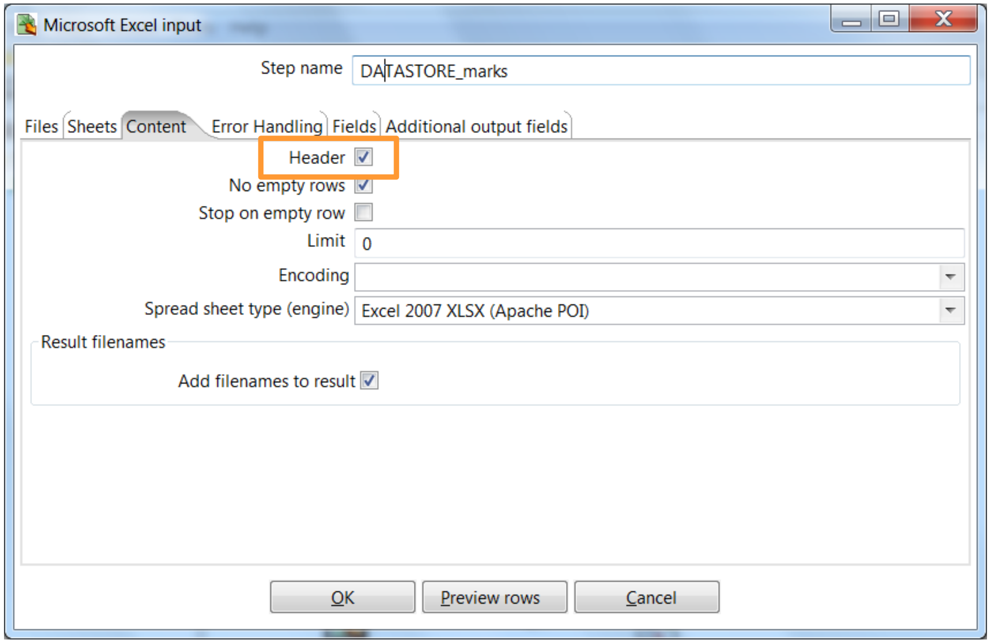
Field in the input to use as



Step name: DATASTORE_marks

List of sheets to read:

#	Sheet name	Start row	Start column
1	marks	0	0



Step name: DATASTORE_marks

Header ☑

No empty rows ☑

Stop on empty row □

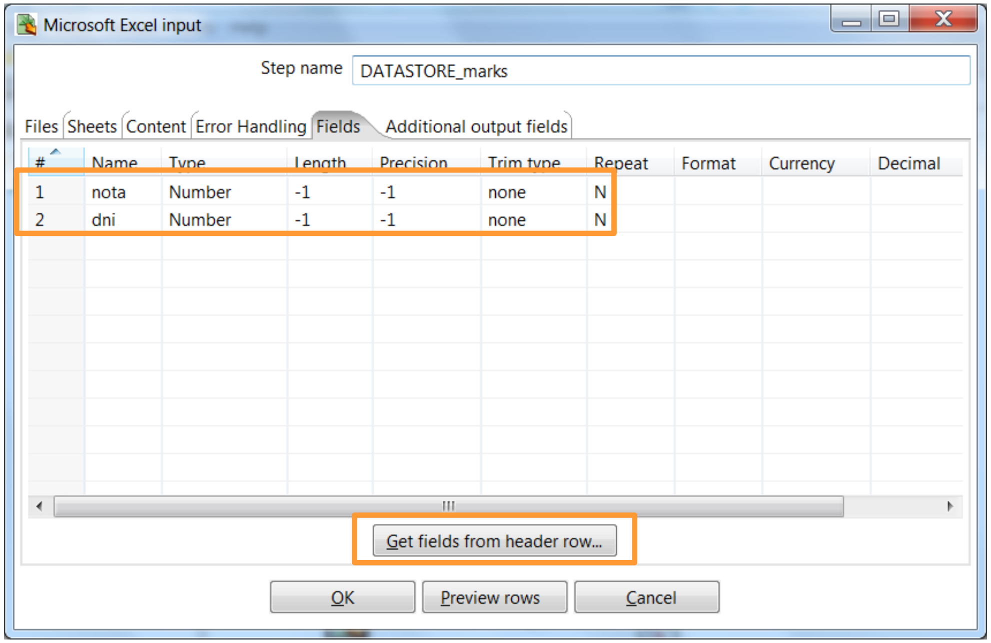
Limit: 0

Encoding:

Spread sheet type (engine): Excel 2007 XLSX (Apache POI)

Result filenames

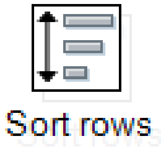
Add filenames to result ☑

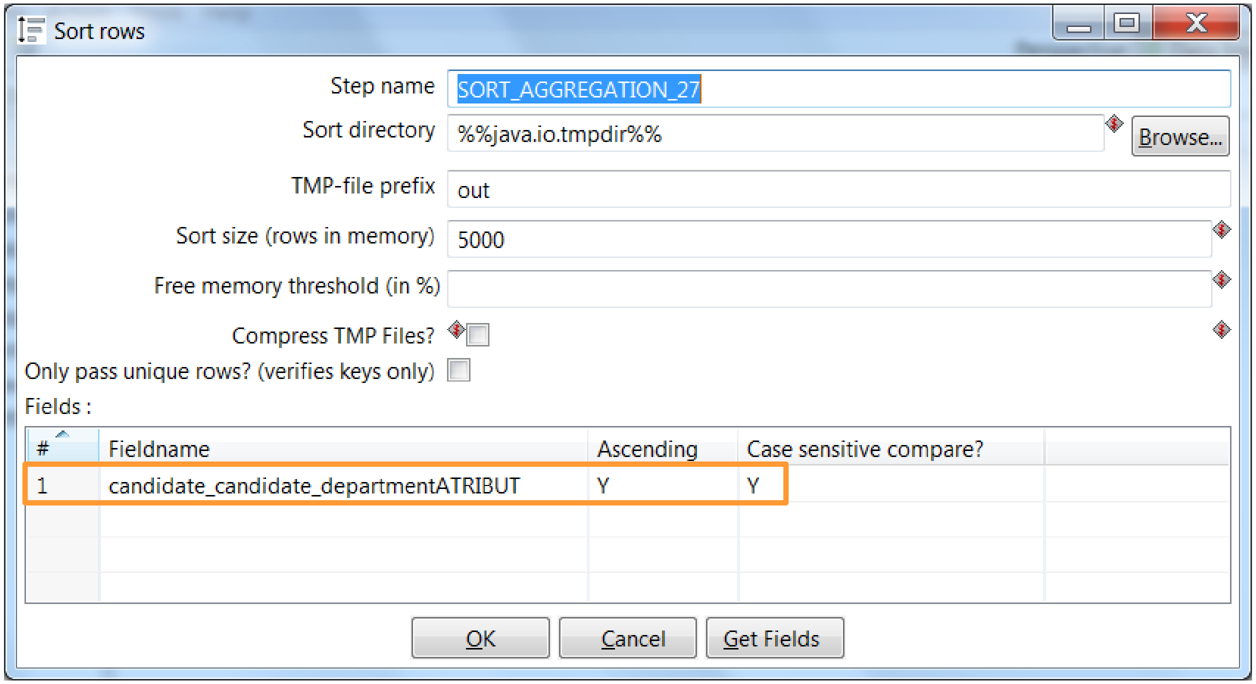


Step name: DATASTORE_marks

#	Name	Type	Length	Precision	Trim type	Repeat	Format	Currency	Decimal
1	nota	Number	-1	-1	none	N			
2	dni	Number	-1	-1	none	N			

b) Transformation steps

Sort Rows  	*The Sort rows step sorts rows based on the fields you specify and on whether they should be sorted in ascending or descending order.*
Option	**Description**
Step name	Name of the step;this name has to be unique in a single transformation.
Sort directory	The directory in which the temporary files are stored in case when needed; the default is the standard temporary directory for the system
TMP-file prefix	Choose an easily recognized prefix so you can identify the files when they show up in the temp directory.
Sort size	The more rows you store in memory, the faster the sorting process because fewer temporary files must be used and less I/O is generated.
Free memory threshold (in %)	If the sort algorithm finds that it has less available free memory than the indicated number, it will start to page data to disk. **Note:** This is not exact science, because: 1. This is checked every 1000 rows. Depending on the row size and other steps within complex transformations this could still lead to an OutOfMemoryError. 2. In a Java Virtual Machine it's not possible to know the exact amount of free memory. As such we recommend you don't use this for very complex transformations with other steps and processes that use up a lot of memory.
Compress TMP Files	Compresses temporary files when they are needed to complete the sort.
Only pass unique rows?	Enable if you want to pass unique rows only to the output stream(s).
Fields table	Specify the fields and direction (ascending/descending) to sort. You can specify whether to perform a case sensitive sort (optional)
Get Fields	Click to retrieve a list of all fields coming in on the stream(s).



Sort rows

Step name: SORT_AGGREGATION_27

Sort directory: %%java.io.tmpdir%%

TMP-file prefix: out

Sort size (rows in memory): 5000

Free memory threshold (in %):

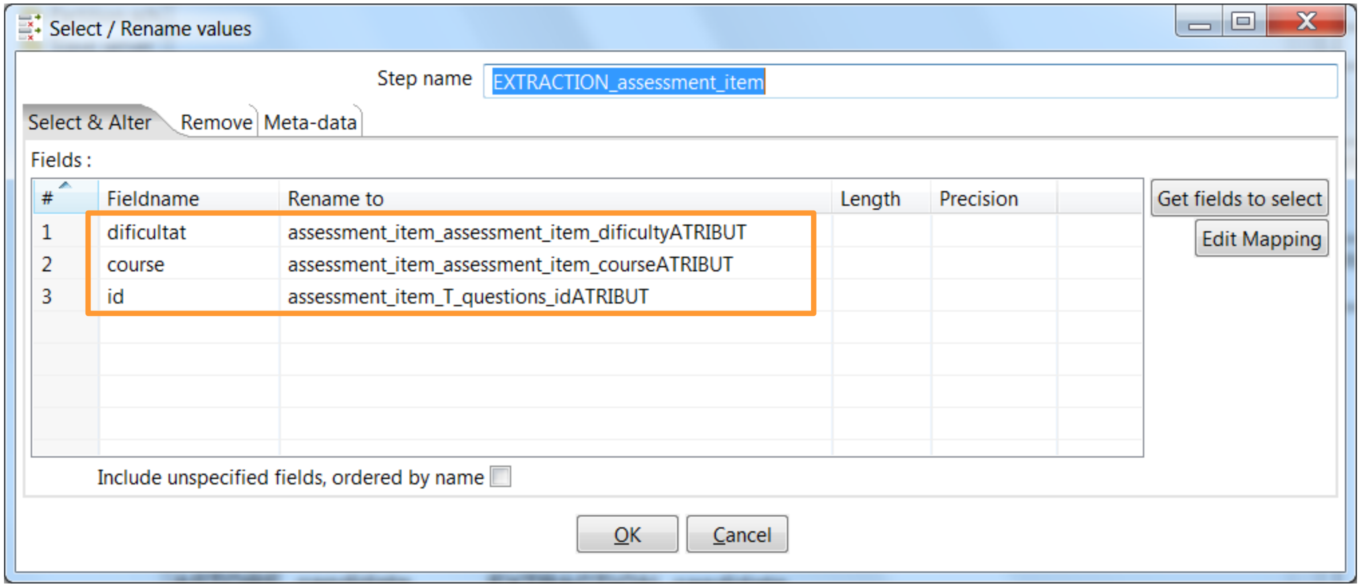
Compress TMP Files? □

Only pass unique rows? (verifies keys only) □

Fields :

#	Fieldname	Ascending	Case sensitive compare?
1	candidate_candidate_departmentATRIBUT	Y	Y

Select values Select values	*This Select values step is useful for selecting (projection), renaming, changing data types and configuring the length and precision of the fields on the stream. These operations are organized into different categories:* • *Select & Alter - Specify the exact order and name in which the fields have to be placed in the output rows* • *Remove - Specify the fields that have to be removed from the output rows* • *Meta-data - Change the name, type, length and precision (the meta-data) of one or more fields*
SELECT & ALTER	
Option	**Description**
Step name	Name of the step; this name has to be unique in a single transformation
Fields	Allows you to rename a field and specify the length and precision
Get fields to select	Click to insert fields from all input steams to the step
Edit Mapping	Click to open a mapping dialog to easily define multiple mappings between source and target fields. ***Note***: Works if there is only one target output step.
Include unspecified fields, ordered by name	Enable if you want to implicitly select all other fields from the input stream(s) that are not explicitly selected in the Fields section



Select / Rename values

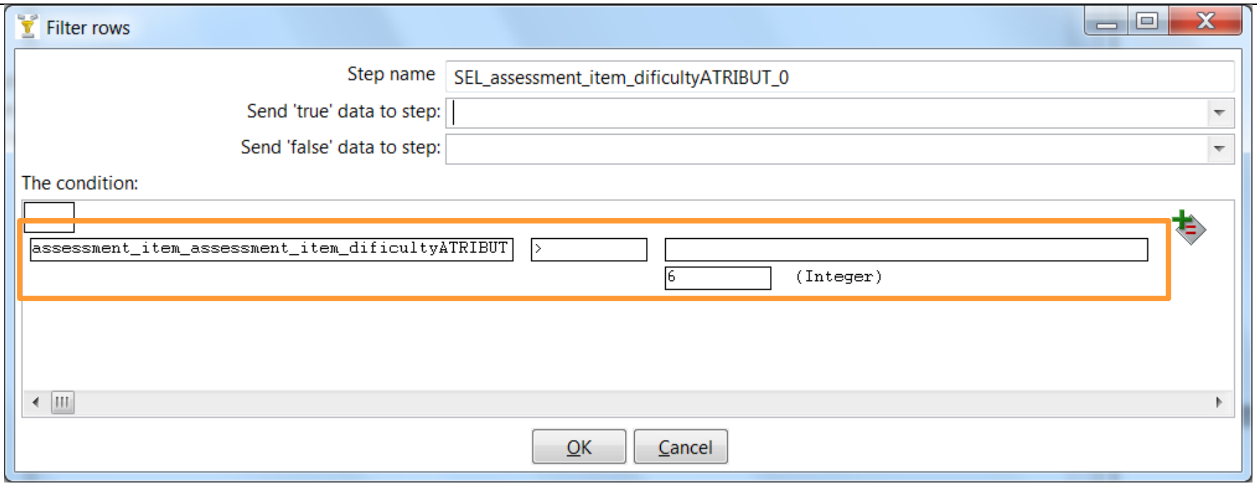
Step name: EXTRACTION_assessment_item

Fields :

#	Fieldname	Rename to	Length	Precision
1	dificultat	assessment_item_assessment_item_dificultyATRIBUT		
2	course	assessment_item_assessment_item_courseATRIBUT		
3	id	assessment_item_T_questions_idATRIBUT		

Include unspecified fields, ordered by name ☐

Filter rows Filter rows	*The Filter rows step allows you to filter rows based on conditions and comparisons. Once this step is connected to a previous step (one or more and receiving input), you can click on the "<field>", "=" and "<value>" areas to construct a condition. Click the Add condition icon to add conditions.* *Add condition converts the original condition to a sub-condition and adds a new condition. Click a sub-condition to edit it (go down one level into the condition tree).*
Option	**Description**
Step name	Name of the step; this name has to be unique in a single transformation.
Send 'true' data to step	The rows for which the condition specified is true are sent to this step
Send 'false' data to step	The rows for which the condition specified are false are sent to this step ***Note: If neither the step for ‘true’ nor for ‘false’ data is defined then only the data satisfying the condition are passed to the next step related by the hop to this step. If the step for either ‘true’ or ‘false’ is defined the other step must be defined too.***
The Condition	Click the 'NOT' button in the upper left to negate the condition. Click on the <Field> buttons to select from a list of fields from the input stream(s) to build your condition(s). Click on the <value> button to enter a specific value into your condition(s). To delete a condition, right-click and select Delete Condition
Add Condition button	Click to add a condition



Step name: SEL_assessment_item_dificultyATRIBUT_0

Send 'true' data to step:

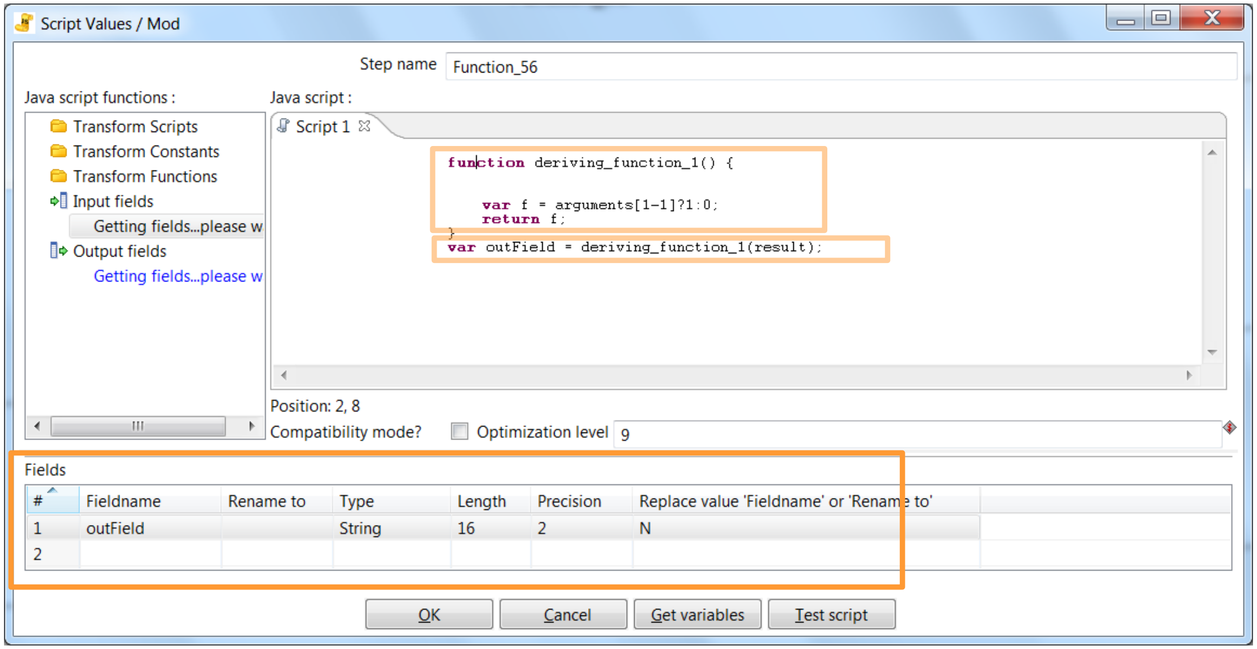
Send 'false' data to step:

The condition:

```
assessment_item_assessment_item_dificultyATRIBUT   >
                                                       6        (Integer)
```

Modified Java Script Value Modified Java Script Value	*Modified Java Script Value type allows you to perform complex calculations using JavaScript.*
Option	**Description**
Java Scripts	This section is where you edit the script for this step. You can insert functions, constants, input fields, etc. from the tree control on the left by double-clicking on the node you wish to insert or by dragging the object onto the Java Script panel.
Fields	The Fields table contains a list of variables from your script including the ability to add metadata like a descriptive name.

For example, you can define a java script function to derive a new value (e.g., derived measure). Function can be parameterized with the input fields of this step. The derived value (measure) need to be defined as the output field of the step in section "Fields". Provided java script needs to have the definition of the value for the output fields. However, you can define an arbitrary java script to calculate desired value from the input fields and assign it to the output field defined in section "Fields".



Step name: Function_56

```
function deriving_function_1() {


    var f = arguments[1-1]?1:0;
    return f;
}
var outField = deriving_function_1(result);
```

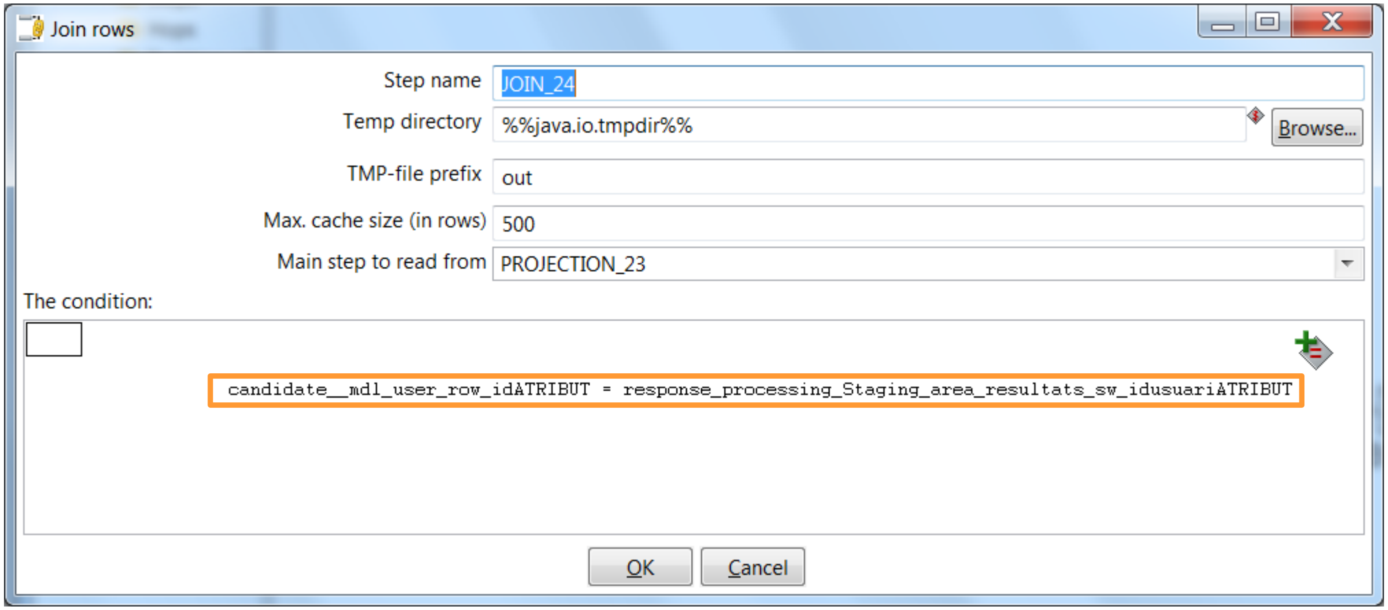
Position: 2, 8

Compatibility mode? □ Optimization level 9

Fields

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	outField		String	16	2	N
2						

Join Rows Join Rows (cartesian product)	*The Join rows step allows you to produce combinations (Cartesian product) of all rows in the input streams with additional condition to perform different kinds of joins.*
Option	**Description**
Step name	Name of the step; this name has to be unique in a single transformation
Temp directory	Specify the name of the directory where the system stores temporary files in case you want to combine more then the cached number of rows
TMP-file prefix	This is the prefix of the temporary files that will be generated
Max. cache size	The number of rows to cache before the system reads data from temporary files; required when you want to combine large row sets that do not fit into memory
Main step to read from	Specifies the step from which to read most of the data; while the data from other steps are cached or spooled to disk, the data from this step is not.
The Condition(s)	You can enter a complex condition to limit the number of output row. **Note**: The fields in the condition must have unique names in each of the streams.



Join rows

Step name: JOIN_24

Temp directory: %%java.io.tmpdir%%

TMP-file prefix: out

Max. cache size (in rows): 500

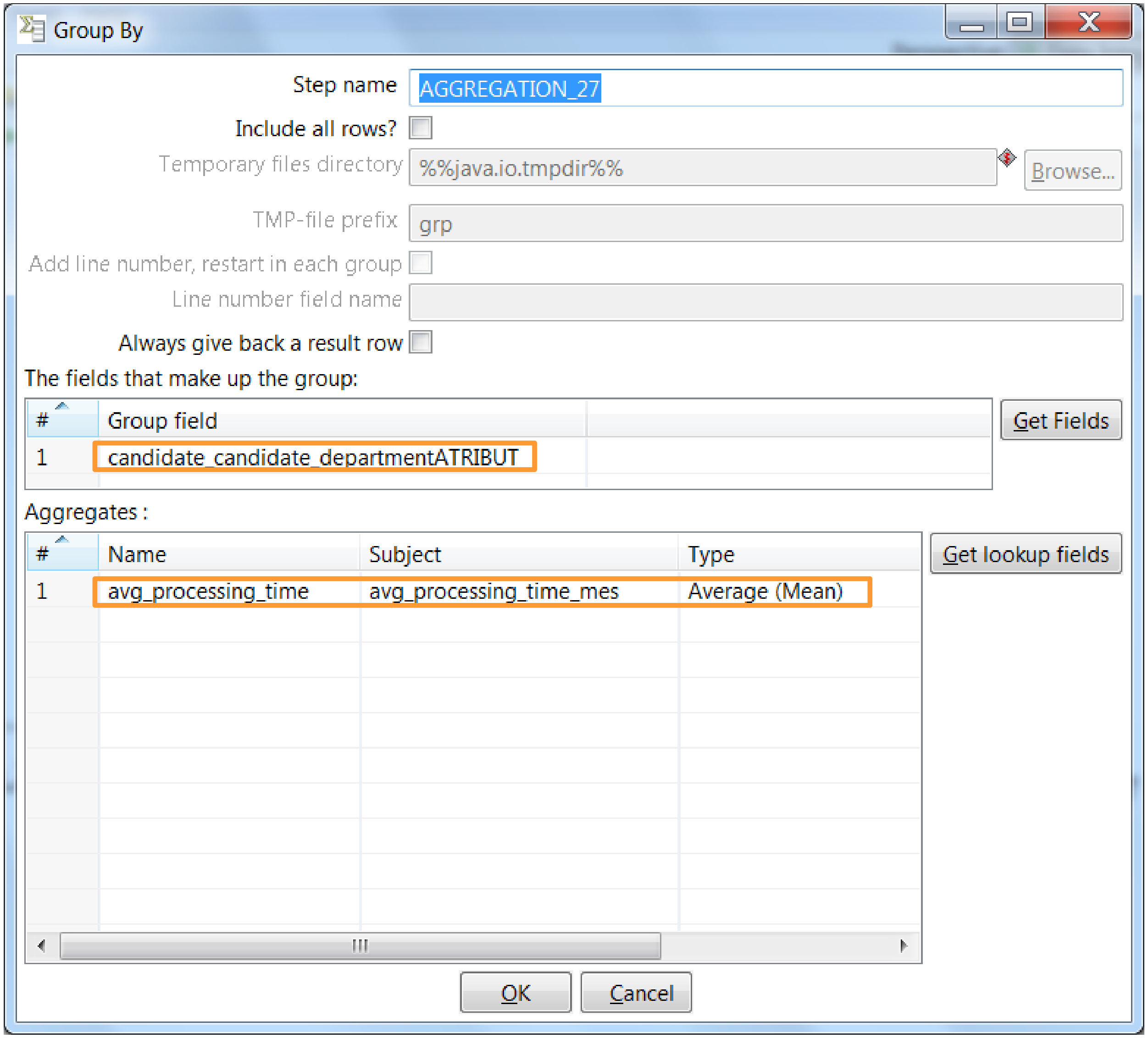
Main step to read from: PROJECTION_23

The condition:

```
candidate__mdl_user_row_idATRIBUT = response_processing_Staging_area_resultats_sw_idusuariATRIBUT
```

Group By Group by	*This step allows you to calculate values over a defined group of fields.* ***Note:*** *This step requires inputs to be ordered by the group by fields (attributes)*
Option	**Description**
Step name	Name of the step; this name has to be unique in a single transformation
Include all rows?	Enable if you want all rows in the output, not just the aggregation; to differentiate between the two types of rows in the output, a flag is required in the output. You must specify the name of the flag field in that case (the type is boolean).
Temporary files directory	The directory in which the temporary files are stored (needed when the *Include all rows* option is enabled and the number or grouped rows exceed 5000 rows); the default is the standard temporary directory for the system
TMP-file prefix	Specify the file prefix used when naming temporary files
Add line number, restart in each group	Enable to add a line number that restarts at 1 in each group
Line number field name	Enable to add a line number that restarts at 1 in each group
Always give back a row	If you enable this option, the Group By step will always give back a result row, even if there is no input row. This can be useful if you want to count the number of rows. Without this option you would never get a count of zero (0).
Group fields table	Specify the fields over which you want to group. Click Get Fields to add all fields from the input stream(s).
Aggregates table	Specify the fields that must be aggregated, the method and the name of the resulting new field. Here are the available aggregation methods : • Sum • Average (Mean) • Minimum • Maximum • Number of values (N) • Concatenate strings separated by , (comma) • First non-null value • Last non-null value • First value (including null) • Last value (including null) • Cumulative sum (all rows option only!) • Cumulative average (all rows option only!)

	• Standard deviation • Concatenate strings separated by <Value>: specify the separator in the Value column • Number of distinct values The metadata injection values for the aggregation type are (in respective order): SUM, AVERAGE, MIN, MAX, COUNT_ALL, CONCAT_COMMA, FIRST, LAST, FIRST_INCL_NULL, LAST_INCL_NULL, CUM_SUM, CUM_AVG, STD_DEV, CONCAT_STRING, COUNT_DISTINCT



Group By

Step name: AGGREGATION_27

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:

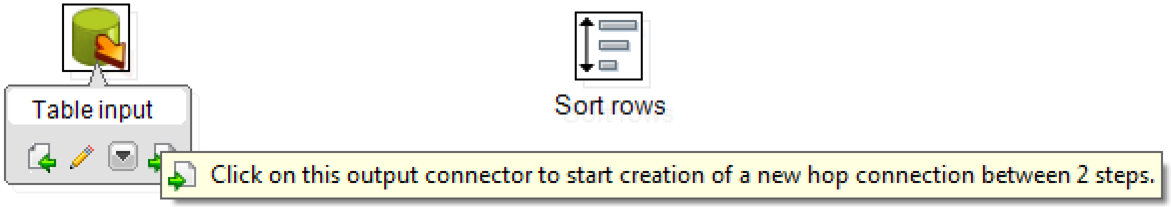
#	Group field
1	candidate_candidate_departmentATRIBUT

Aggregates :

#	Name	Subject	Type
1	avg_processing_time	avg_processing_time_mes	Average (Mean)

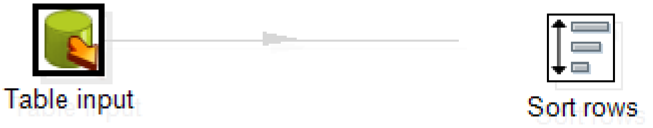
III. Creating edges between operations (hops)

1) Hover the mouse pointer over the source operation node. You should be able to see the options like in figure (1) below.
2) Press&hold the option pointing to the right (see Figure below (1)) to create the output edge from the given operation node.
3) Drag the mouse pointer to the operation node you want to be the destination of the edge.

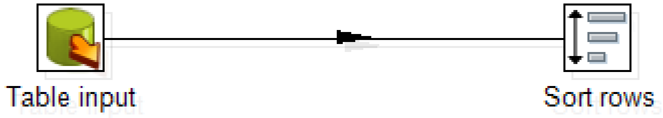


Click on this output connector to start creation of a new hop connection between 2 steps.

(1)



(2)



(3)