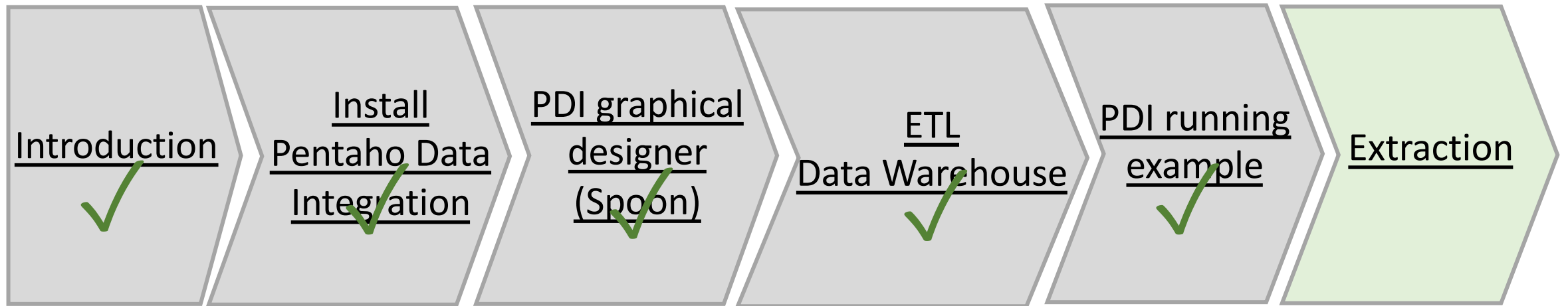


# DATA ANALYTICS (Data Warehouse) Pentaho Data Integration

Luca Cinelli, PhD  
[luca.cinelli@unical.it](mailto:luca.cinelli@unical.it)

# Outline

---



# Extraction

# Data sources



# Example: Datasets

1. Sales Data
2. Customer Data
3. Product Data

# Example: ETL problem scenario

## 1. Extraction

### Customer data

CSV  
TXT  
Excel  
Folder  
Zip file

### Product Data

XML  
JSON

### Sales Data

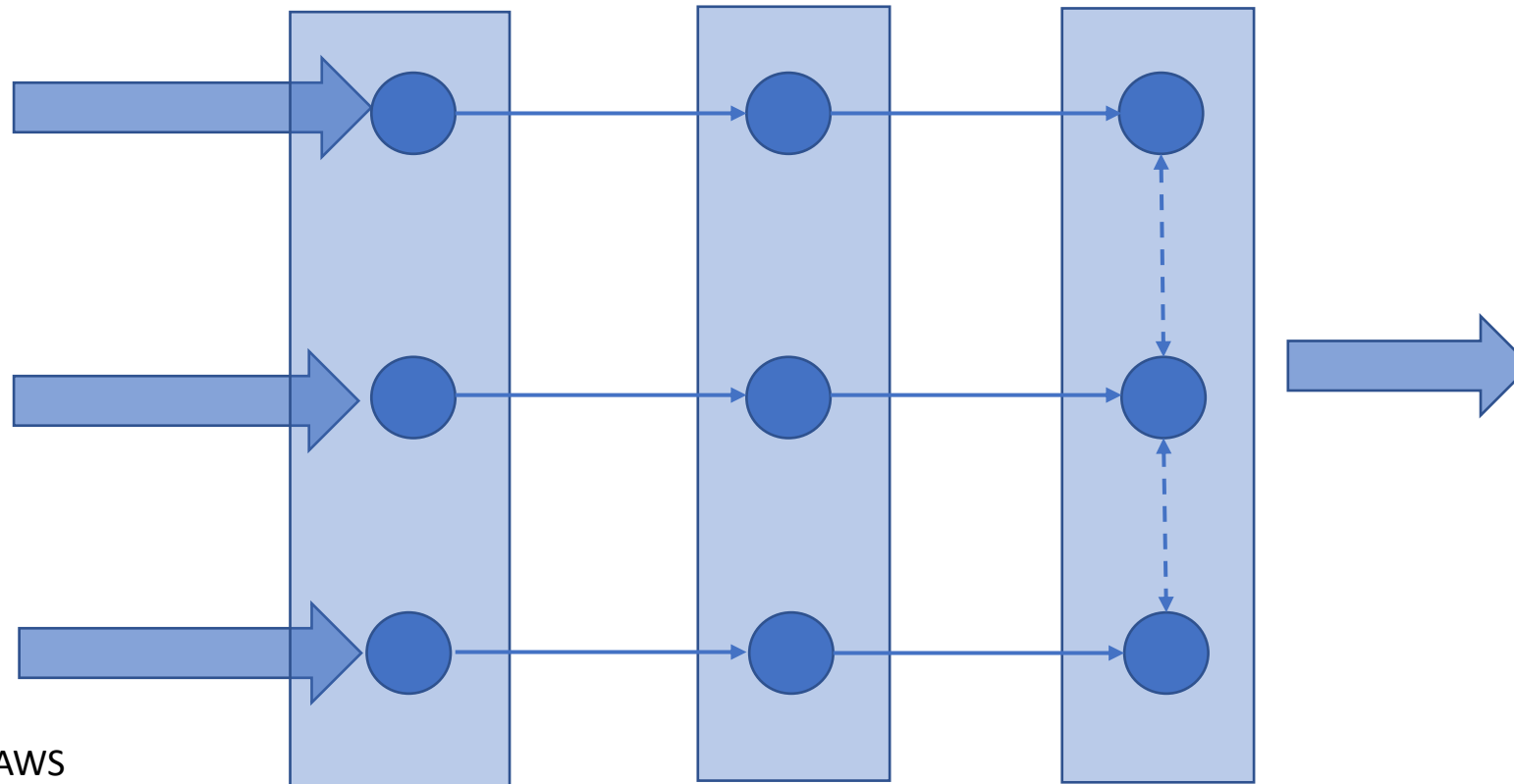
SQL based DB  
Cloud based Storage AWS

## 3. Cleaning and Validation

## 2. Merging

## 4. Transformation, aggregation and Joining

## 5. Loading to create a DataMart



# Extraction from tabular format

- Manually entering data into PDI (ManualInput.ktr)



Data grid

Step name: Manual Input

Meta Data

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Null if	Set empty string?
1	Customer ID	String		10						N
2	Customer Name	String		50						N
3	Segment	String		20						N
4	Age	String		10						N
5	Country	String		25						N
6	City	String		50						N
7	State	String		50						N
8	Postal Code	String		10						N
9	Region	String		10						N

Help OK Preview Cancel

# Extraction from tabular format

- Manually entering data into PDI
- Import data from text file (TxtInput.ktr, input: CustomerData\_Central.txt)



# Extraction from tabular format

- Manually entering data into PDI
- Import data from text file
- Import data from multiple CSV file (MultipleFiles.ktr, input: customer\_data\_multiple\_files)

# Extraction from tabular format

- Manually entering data into PDI
- Import data from text file
- Import data from multiple CSV file version 2  
(MultipleFiles\_withGetFileName.ktr, input:  
customer\_data\_multiple\_files)

# Extraction from tabular format

- Manually entering data into PDI
- Import data from text file
- Import data from multiple CSV file
- Import data from excel file (ExcelInput.ktr, input: CustomerData\_East.xlsx)

# Extraction from tabular format

- Manually entering data into PDI
- Import data from text file
- Import data from multiple CSV file
- Import data from excel file
- Extract data from zip file (ZipInput.ktr, input: CustomerData\_South.zip)

# Extraction from no-tabular format

- Extract data from XML file (XMLInput.ktr, input: ProductDataAsXML.xml)

# Extraction from no-tabular format

- Extract data from XML file
- Extract data from JSON file (JSONInput.ktr, input: ProductDataasJSON.js)

# Outline

---

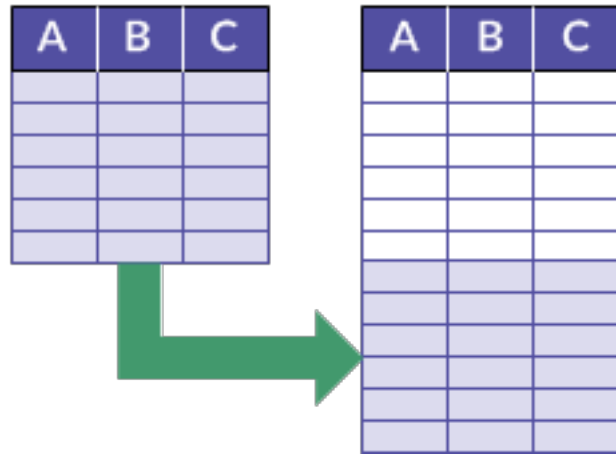


# Merging Data Streams

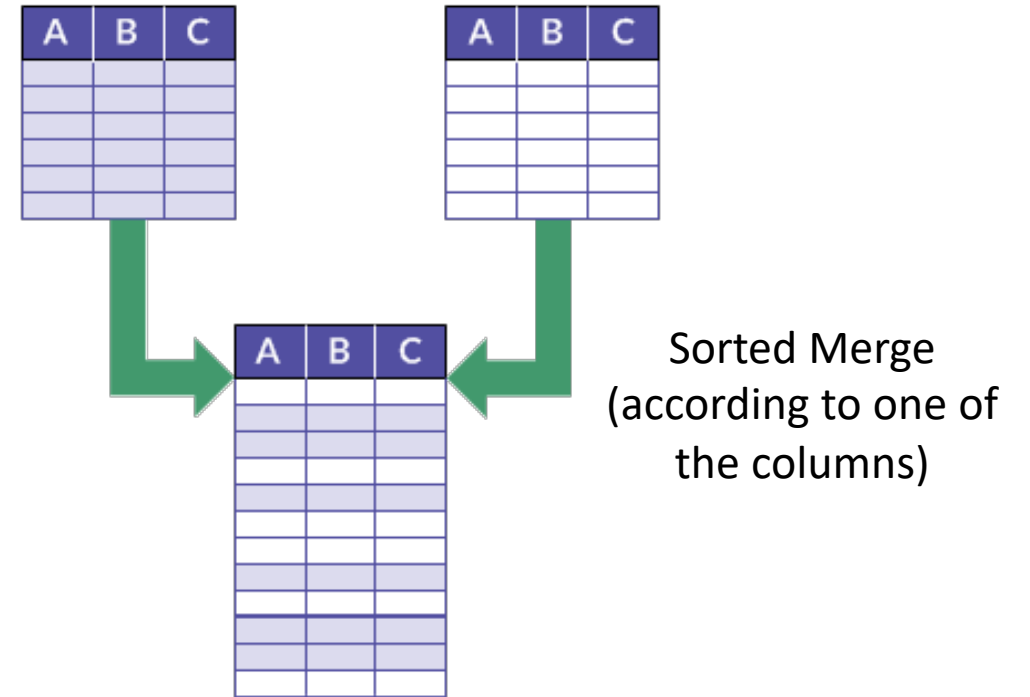


# Merging Streams of Data

- Prefer specialized merging steps such as Append Stream or Sorted Merge



Append  
(e.g. merging sales data)



Sorted Merge  
(according to one of  
the columns)

# Merging Streams of Data

- Merged streams may have duplicates: do deduplication on the primary key column
- Ensure unique occurrence of primary key after merging
  - Same tuple => Delete one duplicated tuple
  - Same primary key but different tuples... Primary key should be unique => Error Handling
- Sort data before deduplicating
  - More efficient, much faster
- Metadata of merging streams must be same

# Merging Streams of Data: Customer data example



Manual Input



Central Data from TXT file



Input multiple files



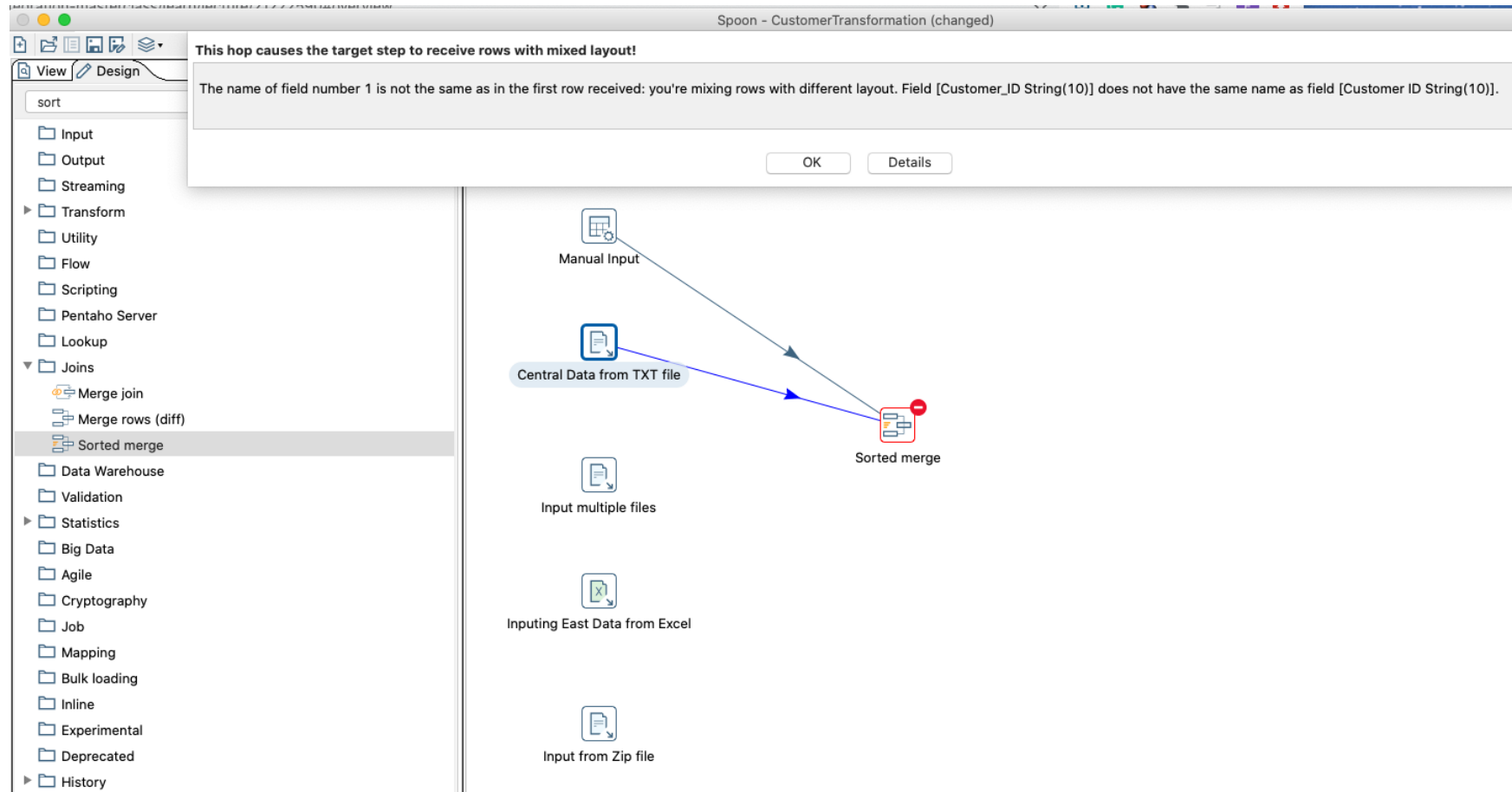
Inputting East Data from Excel



Input from Zip file

Search for “sort” in design Transformation search bar

# Merging Streams of Data: Customer data example



# Merging Streams of Data: Customer data example

This hop causes the target step to receive rows with mixed layout!

The name of field number 1 is not the same as in the first row received: you're mixing rows with different layout. Field [Customer\_ID String(10)] does not have the same name as field [Customer ID String(10)].

View Design

sort

- Input
- Output
- Streaming
- Transform
  - Utility
  - Flow
  - Scripting
  - Pentaho Server
  - Lookup
  - Joins
    - Merge join
    - Merge rows (diff)
    - Sorted merge
  - Data Warehouse
  - Validation
- Statistics
  - Big Data
  - Agile
  - Cryptography
  - Job
  - Mapping
  - Bulk loading
  - Inline
  - Experimental
  - Deprecated
- History

Meta			Data					
#	Name	Type	#	Name	Type	Format	Position	Leng
1	Customer_ID	String	1	Customer ID	String			10
2	Customer_Na...	String	2	Customer Name	String			50
3	Segment	String	3	Segment	String			20
4	Age	String	4	Age	String			10
5	Country	String	5	Country	String			25
6	City	String	6	City	String			50
7	State	String	7	State	String			50
8	Postal_Code	String	8	Postal Code	String			10
9	Region	String	9	Region	String			10

# Merging Streams of Data: Customer data example

Require to sort individual streams before merging

The screenshot displays the Pentaho Data Integration (Kettle) interface. On the left, the 'Transform' tab is active, showing a list of transformation steps. The 'Sort rows' step is highlighted. The main workspace shows a data flow diagram with several input streams (Manual Input, Central Data from TXT file, Input multiple files, Inputting East Data from, Input from Zip file) feeding into 'Sort rows' and 'Sort rows 2' steps. These sorted streams then feed into a 'Sorted merge' step. A configuration dialog for the 'Sort rows' step is open, showing the following settings:

- Step name: Sort rows
- Sort directory: %%java.io.tmpdir%%
- TMP-file prefix: out
- Sort size (rows in memory): 1000000
- Free memory threshold (in %):
- Compress TMP Files?: ☐
- Only pass unique rows? (verifies keys only): ☒

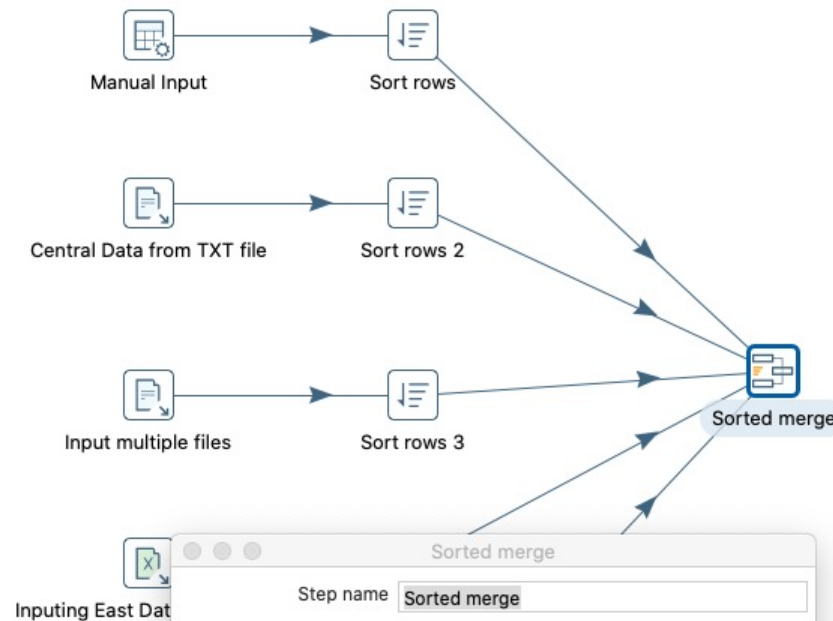
The 'Fields' table is also visible:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	Customer ID	Y	N	N	0	N

Sort size: the maximum number of rows sorted at a time:

- It will take up one million rows of data at a time.
- Those rows will be stored in the memory of Java Virtual Machine, in a temporary file.
- Then the next set of one million rows will be picked up.
- Those will be sorted.
- And these two sorted temp files will then be merged.

# Merging Streams of Data: Customer data example



Sorted merge

Step name:

Fields :

#	Fieldname	Ascending
1	Customer ID	Y

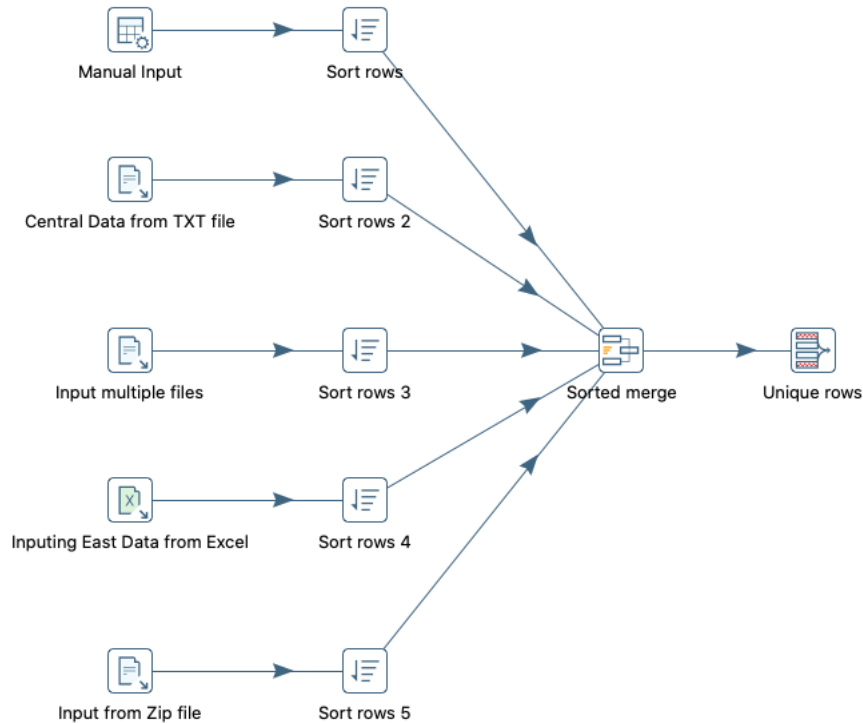
Notice

⚠ If the incoming data is not sorted on the specified keys, the output results may not be correct. We recommend sorting the incoming data within the transformation.

☐ Don't show this message again.

Close

# Merging Streams of Data: Customer data example



Preview

Examine preview data

Rows of step: Unique rows (793 rows)

#	Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code
1	AA-10315	Alex Avila	Consumer	66	United States	Minneapolis	Minnesota	55407
2	AA-10375	Allen Arnold	Consumer	22	US	Mesa	Arizona	85204
3	AA-10480	Andrew Allen	Consumer	50	United States	Concord	North Carolina	28027
4	AA-10645	Anna Andreadi	Consumer	32	United States	Chester	Pennsylvania	19013
5	AB-10015	Aaron Bergman	Consumer	66	USA	Seattle	Washington	98103
6	AB-10060	Adam Bellavance	Home Office	25	United States	New York City	New York	10009
7	AB-10105	Adrian Barton	Consumer	63	US	Phoenix	Arizona	85023
8	AB-10150	Aimee Bixby	Consumer	65	United States	Long Beach	New York	11561
9	AB-10165	Alan Barnes	Consumer	22	United States	Los Angeles	California	90036
10	AB-10255	Alejandro Ballentine	Home Office	34	United States	Lorain	Ohio	44052
11	AB-10600	Ann Blume	Corporate	34	US	Tucson	Arizona	85705
12	AC-10420	Alyssa Crouse	Corporate	69	United States	San Francisco	California	94122
13	AC-10450	Amy Cox	Consumer	46	United States	Minneapolis	Minnesota	55407
14	AC-10615	Ann Chong	Corporate	61	United States	New York City	New York	10009
15	AC-10660	Anna Chung	Consumer	30	United States	Huntsville	Texas	77340
16	AD-10180	Alan Dominguez	Home Office	52	United States	Houston	Texas	77041
17	AF-10870	Art Ferguson	Consumer	63	United States	College Station	Texas	77840
18	AF-10885	Art Foster	Consumer	40	United States	Louisville	Kentucky	40214
19	AG-10270	Alejandro Grove	Consumer	18	United States	West Jordan	Utah	84084
20	AG-10300	Aleksandra Gannaway	Corporate	68	United States	Los Angeles	California	90049
21	AG-10330	Alex Grayson	Consumer	51	United States	Stockton	California	95207
22	AG-10390	Allen Goldenen	Consumer	47	United States	Cincinnati	Ohio	45231
23	AG-10495	Andrew Gjertsen	Corporate	24	United States	Philadelphia	Pennsylvania	19140
24	AG-10525	Andy Gerbode	Corporate	69	United States	Saint Petersburg	Florida	33710

Close

Logging

2020/11/14 11:09:00 - Spoon - Save file as...

2020/11/14 11:37:02 - Spoon - Save file as...

2020/11/14 11:37:17 - Spoon - Save file as...

2020/11/14 11:39:11 - CustomerTransformation\_check4\_ded - Dispatching started for transformation [CustomerTransformation\_check4\_ded]

2020/11/14 11:39:11 - Central Data from TXT file.0 - Opening file: file:///Users/linda/Documents/DWH\_PDI/L3/Data/Customer Data/CustomerData\_Central.txt

2020/11/14 11:39:11 - Input from Zip file.0 - Opening file: file:///Users/linda/Documents/DWH\_PDI/L3/Data/Customer Data/CustomerData\_South.zip

2020/11/14 11:39:11 - Input multiple files.0 - Opening file: file:///Users/linda/Documents/DWH\_PDI/L3/Data/Customer Data/CustomerData\_West/CustomerData...

2020/11/14 11:39:11 - Manual Input.0 - Finished processing (I=0, O=0, R=0, W=2, U=0, E=0)

2020/11/14 11:39:11 - Input multiple files.0 - Opening file: file:///Users/linda/Documents/DWH\_PDI/L3/Data/Customer Data/CustomerData\_West/CustomerData...

2020/11/14 11:39:11 - Input from Zip file.0 - Finished processing (I=134, O=0, R=0, W=133, U=1, E=0)

2020/11/14 11:39:11 - Central Data from TXT file.0 - Finished processing (I=185, O=0, R=0, W=184, U=1, E=0)

2020/11/14 11:39:11 - Sort rows.0 - Finished processing (I=0, O=0, R=2, W=2, U=0, E=0)

2020/11/14 11:39:11 - Input multiple files.0 - Opening file: file:///Users/linda/Documents/DWH\_PDI/L3/Data/Customer Data/CustomerData\_West/CustomerData...

2020/11/14 11:39:11 - Input multiple files.0 - Opening file: file:///Users/linda/Documents/DWH\_PDI/L3/Data/Customer Data/CustomerData\_West/CustomerData...

2020/11/14 11:39:11 - Input multiple files.0 - Opening file: file:///Users/linda/Documents/DWH\_PDI/L3/Data/Customer Data/CustomerData\_West/CustomerData...

2020/11/14 11:39:11 - Input multiple files.0 - Finished processing (I=259, O=0, R=0, W=254, U=5, E=0)

2020/11/14 11:39:11 - Sort rows 5.0 - Finished processing (I=0, O=0, R=133, W=133, U=0, E=0)

2020/11/14 11:39:11 - Sort rows 2.0 - Finished processing (I=0, O=0, R=184, W=184, U=0, E=0)

2020/11/14 11:39:11 - Sort rows 3.0 - Finished processing (I=0, O=0, R=254, W=254, U=0, E=0)

2020/11/14 11:39:11 - Inputting East Data from Excel.0 - Finished processing (I=220, O=0, R=0, W=220, U=0, E=0)

2020/11/14 11:39:11 - Sort rows 4.0 - Finished processing (I=0, O=0, R=220, W=220, U=0, E=0)

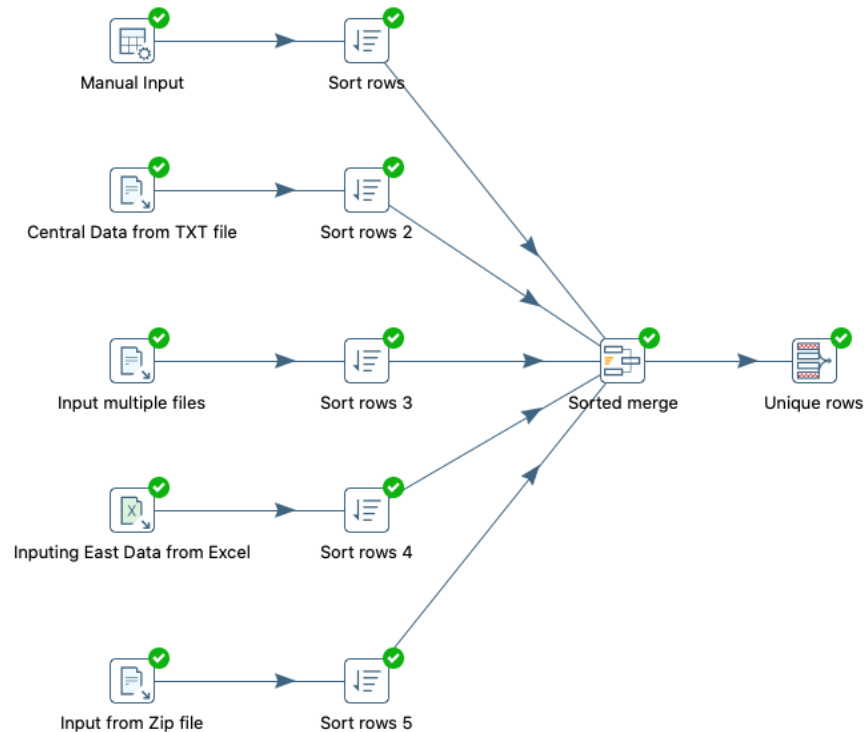
2020/11/14 11:39:12 - Sorted merge.0 - Finished processing (I=0, O=0, R=793, W=793, U=0, E=0)

2020/11/14 11:39:12 - Unique rows.0 - Finished processing (I=0, O=0, R=793, W=793, U=0, E=0)

2020/11/14 11:39:12 - Spoon - The transformation has finished!!



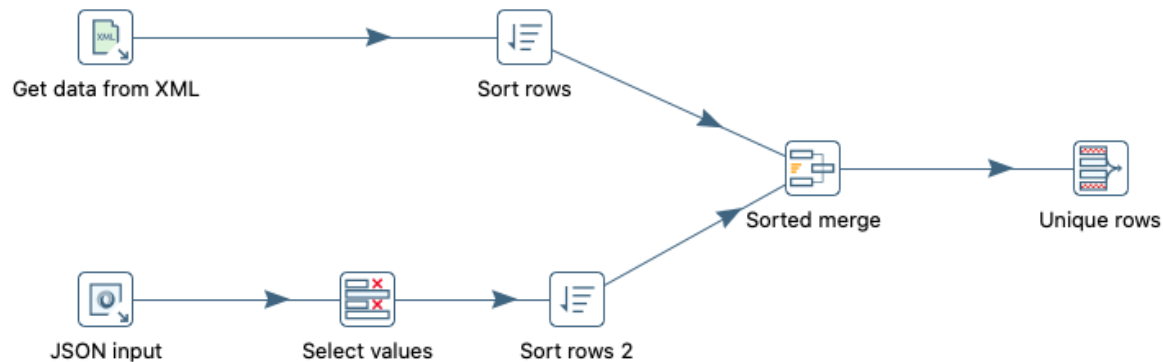
# Merging Streams of Data: Customer data example



## Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data													
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Central Data from TXT file	0	0	184	185	0	1	0	0	Finished	0.0s	18,500	-
2	Sort rows 2	0	184	184	0	0	0	0	0	Finished	0.0s	5,576	-
3	Input from Zip file	0	0	133	134	0	1	0	0	Finished	0.0s	14,889	-
4	Sort rows 5	0	133	133	0	0	0	0	0	Finished	0.0s	7,824	-
5	Input multiple files	0	0	254	259	0	5	0	0	Finished	0.0s	17,267	-
6	Sort rows 3	0	254	254	0	0	0	0	0	Finished	0.0s	6,195	-
7	Inputing East Data from E...	0	0	220	220	0	0	0	0	Finished	0.1s	2,973	-
8	Sort rows 4	0	220	220	0	0	0	0	0	Finished	0.1s	2,785	-
9	Manual Input	0	0	2	0	0	0	0	0	Finished	0.0s	286	-
10	Sort rows	0	2	2	0	0	0	0	0	Finished	0.0s	182	-
11	Sorted merge	0	793	793	0	0	0	0	0	Finished	0.9s	902	-
12	Unique rows	0	793	793	0	0	0	0	0	Finished	0.9s	899	-

# Merging Streams of Data: Product data example



Get data from XML

Step nameGet data from XML

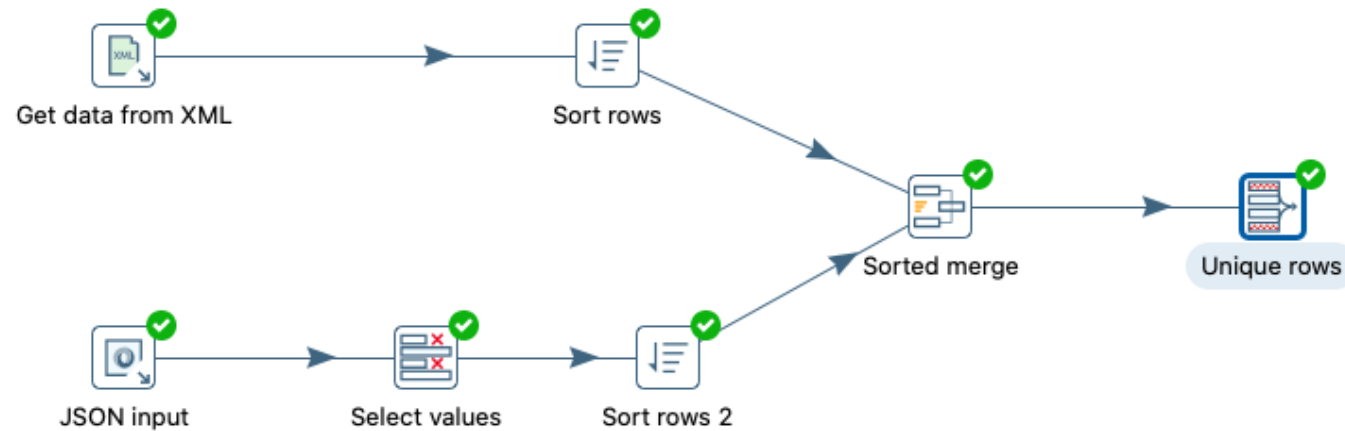
File	Content	Fields	Additional output fields										
#	Name	XPath	Element	Result type	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	Product_ID	Product_ID	Node	Value of	String		20					none	N
2	Category	Category	Node	Value of	String		50					none	N
3	Sub_Category	Sub_Category	Node	Value of	String		50					none	N
4	Product_Name	Product_Name	Node	Value of	String		200					none	N

JSON input

Step nameJSON input

File	Content	Fields	Additional output fields										
#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat		
1	Category	\$.[*].Category	String		50					none	N		
2	Product_ID	\$.[*].Product_ID	String		20					none	N		
3	Product_Name	\$.[*].Product_Name	String		50					none	N		
4	Sub_Category	\$.[*].Sub_Category	String		200					none	N		

# Merging Streams of Data: Product data example



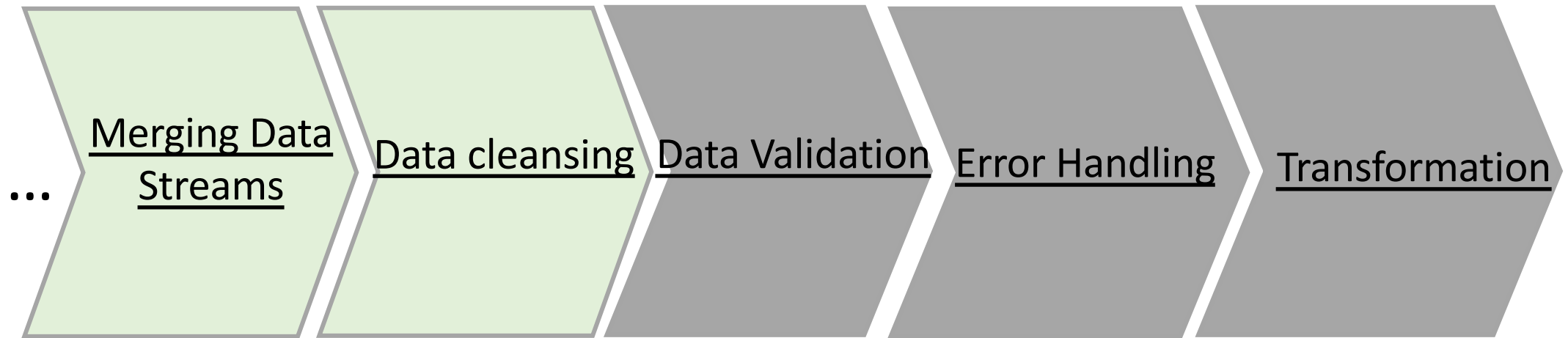
## Execution Results

Execution History													
Step Metrics													
Performance Graph													
Metrics													
Preview data													
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	JSON input	0	0	375	375	0	0	0	0	Finished	0.0s	28,846	-
2	Get data from XML	0	0	1496	1496	0	0	0	0	Finished	0.1s	24,933	-
3	Select values	0	375	375	0	0	0	0	0	Finished	0.0s	22,059	-
4	Sort rows	0	1496	1487	0	0	0	0	0	Finished	0.1s	22,328	-
5	Sort rows 2	0	375	375	0	0	0	0	0	Finished	0.0s	17,857	-
6	Sorted merge	0	1862	1862	0	0	0	0	0	Finished	0.4s	4,738	-
7	Unique rows	0	1862	1862	0	0	0	0	0	Finished	0.4s	4,714	-

There were nine duplicates which were removed by the Sort step.

# Outline

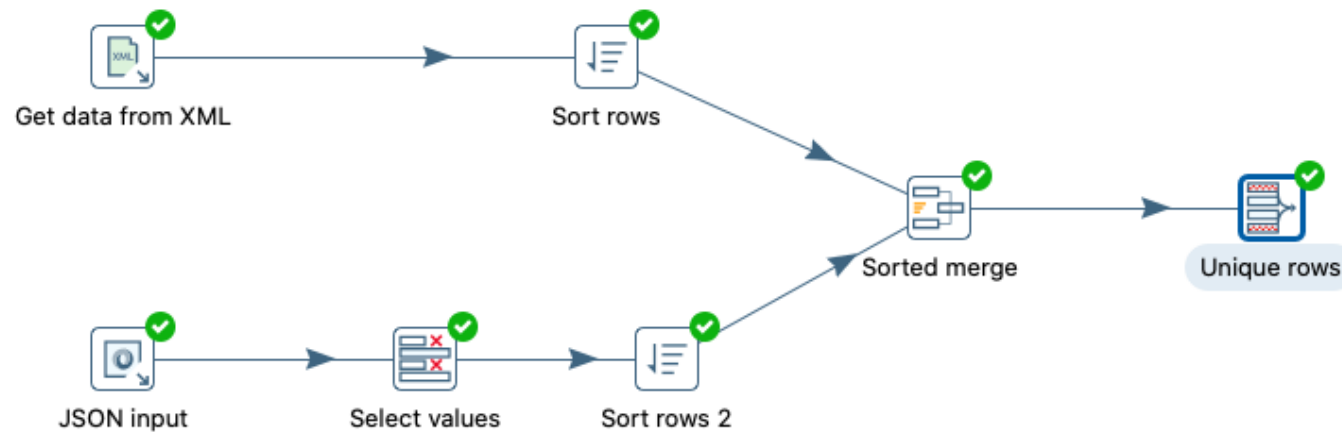
---



# Data Cleansing

# Data Cleansing: Remove duplicates example

- Remove duplicates is also part of the data cleansing process (productTransformation)



# Data Cleansing

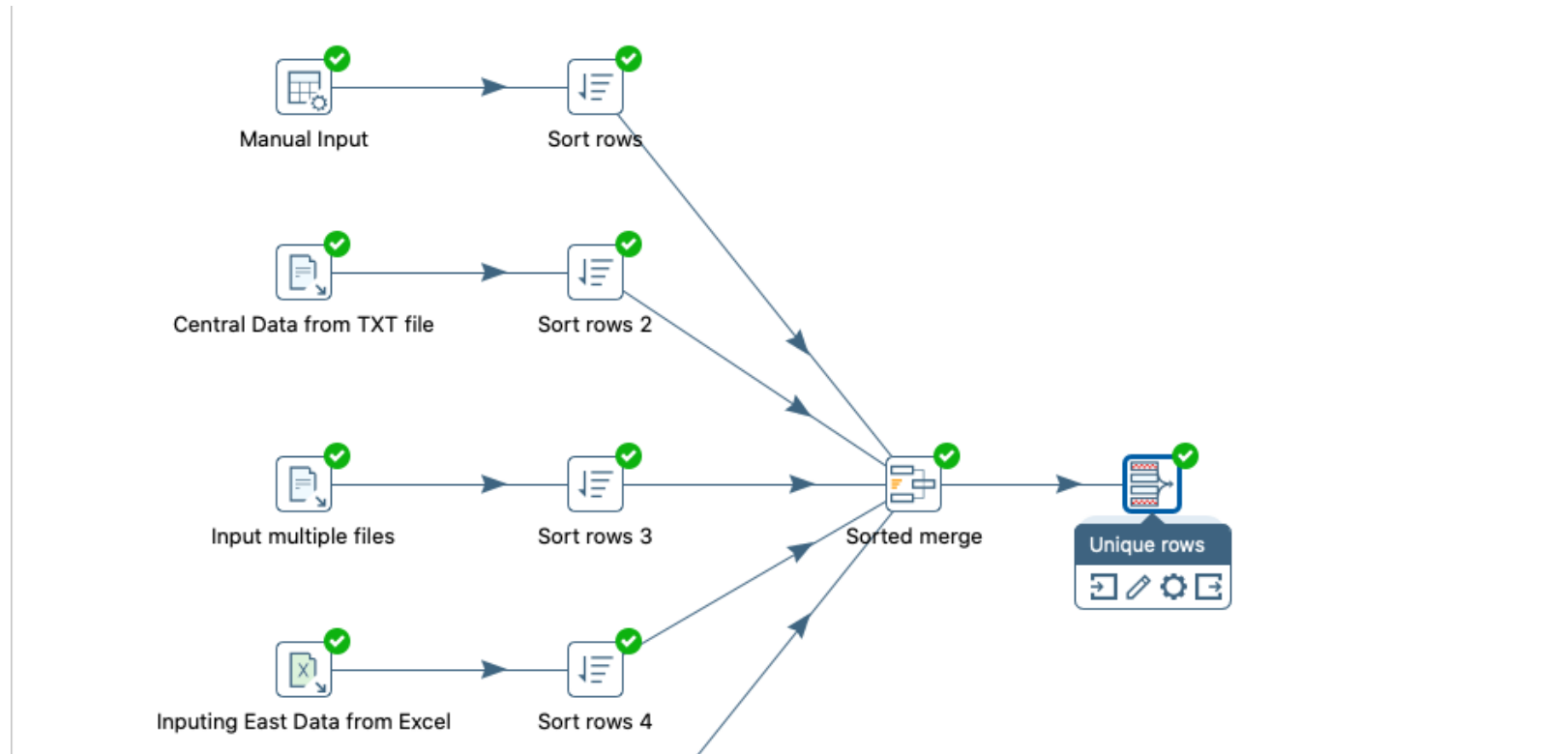
- Correcting small mistakes such as typing mistakes or data format related issues.
  - Examples: 5 vs 5.0, 16 Nov vs 16/11/2020, duplicate entries, etc.
- We cleaned data while extracting:
  - setting format of dates, sales and profit value in sales data, removing duplicates from product data etc.
- We perform data clean after data extraction, in transformation of ETL

# Data Cleansing: Examples

- Multiple names for the same entity:
  - Country field contains US, USA, United States and United States of America to represent one country
- Special symbols to remove:
  - # is present in City column
- Manual entry mistakes:
  - In state column California is sometimes written as Cakifornia and californis etc.
  - Exact Match and Fuzzy match
- Format changes:
  - Changing discount value in sales table to percentage value
  - Change date format from mm/dd/yyyy to dd-mm-yyyy for order and ship dates



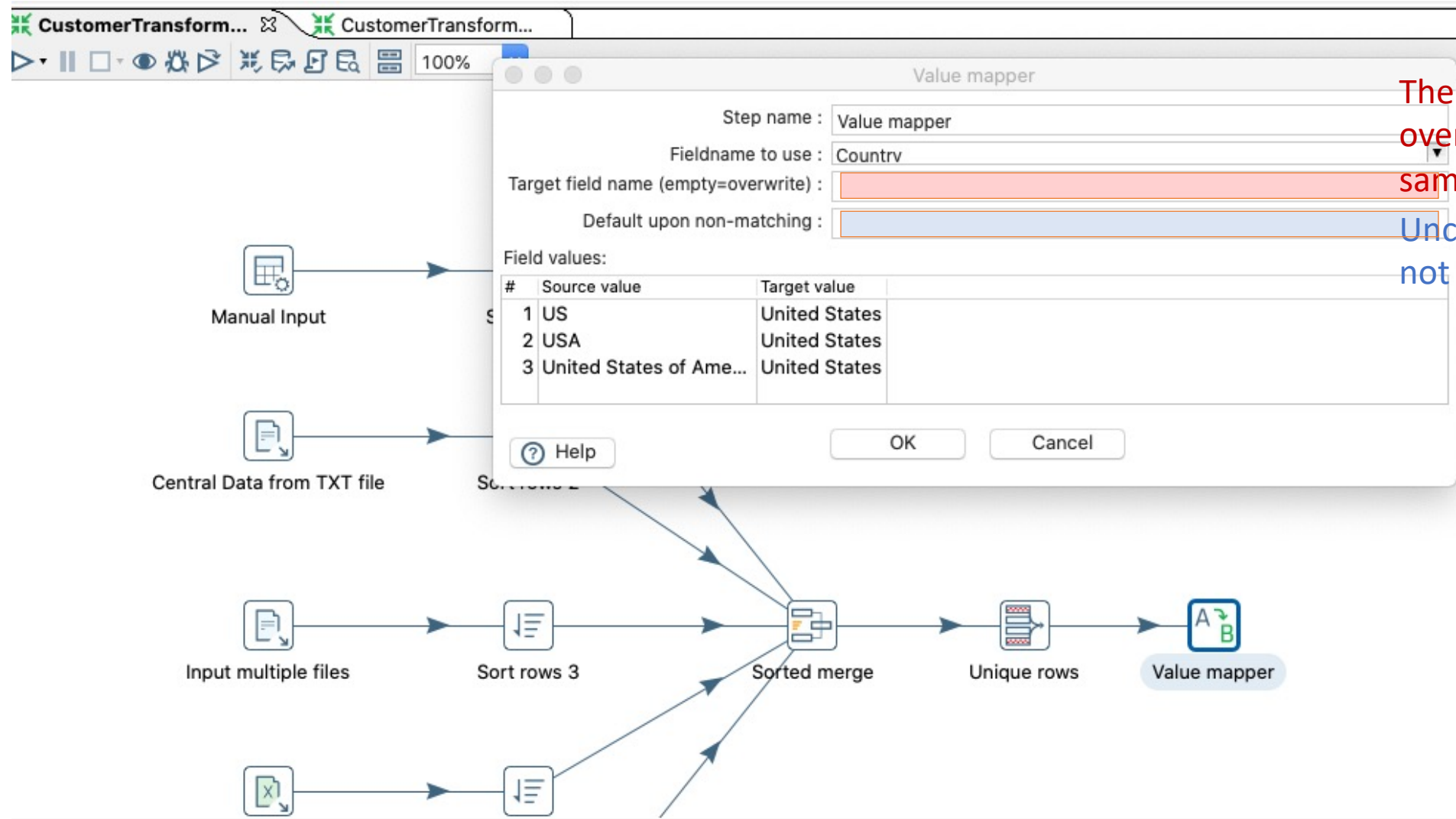
# Data Cleansing: Multiple names example



## Execution Results

Execution Results										
Logging Execution History Step Metrics Performance Graph Metrics Preview data										
First rows Last rows Off										
#	Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
1	AA-10315	Alex Avila	Consumer	66	United States	Minneapolis	Minnesota	55407	Central	
2	AA-10375	Allen Arnold	Consumer	22	US	Mesa	Arizona	85204	West	
3	AA-10480	Andrew Allen	Consumer	50	United States	Concord	North Carolina	28027	South	
4	AA-10645	Anna Andreadi	Consumer	32	United States	Chester	Pennsylvania	19013	East	
5	AB-10015	Aaron Bergman	Consumer	66	USA	Seattle	Washington	98103	West	
6	AB-10060	Adam Bellavance	Home Office	35	United States	New York City	New York	10009	East	

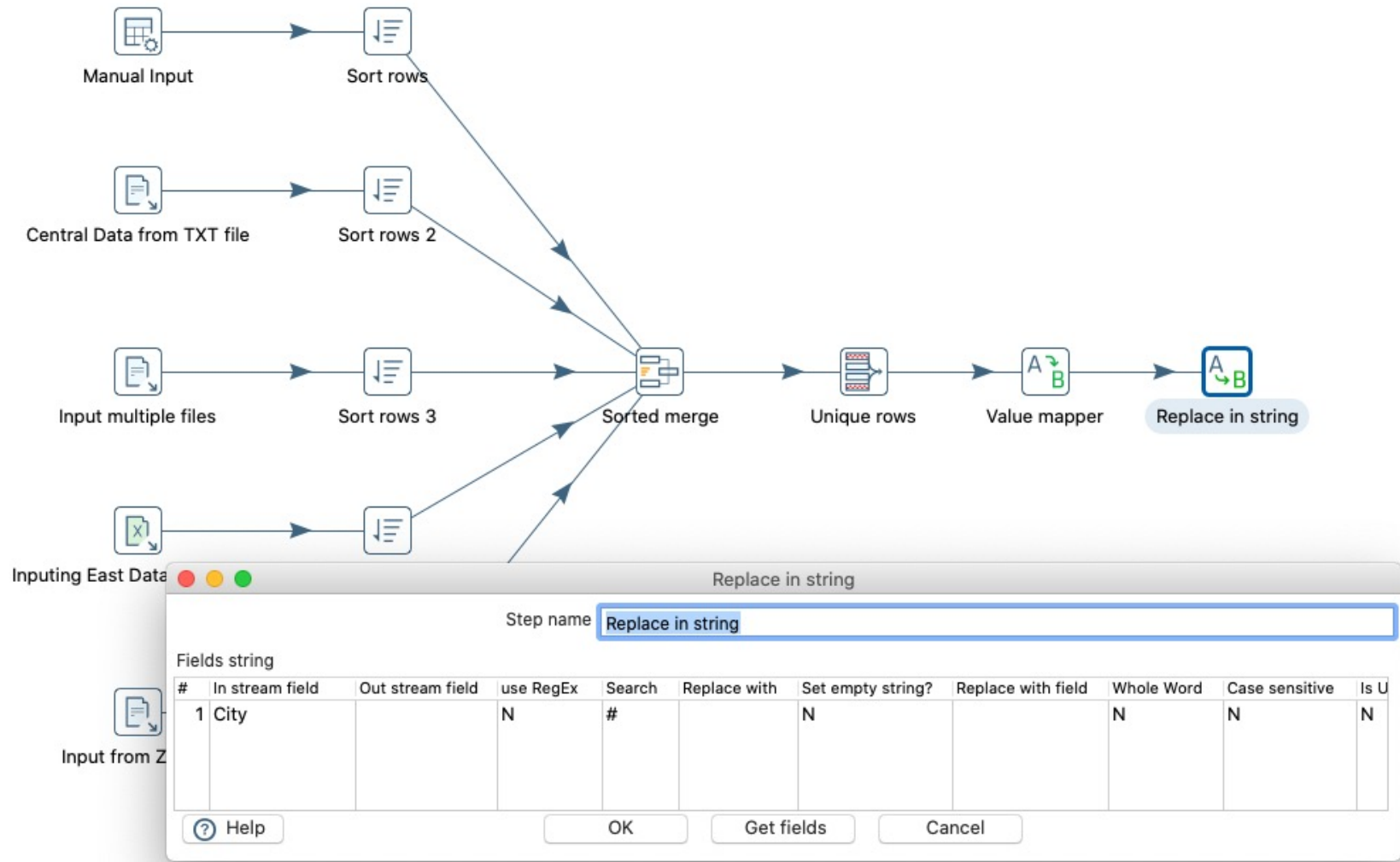
# Data Cleansing: Multiple names example



# Data Cleansing: Examples

- Multiple names for the same entity:
  - Country field contains US, USA, United States and United States of America to represent one country
- **Special symbols to remove:**
  - # is present in City column
- Manual entry mistakes:
  - In state column California is sometimes written as Cakifornia and californis etc.
  - Exact Match and Fuzzy match
- Format changes:
  - Changing discount value in sales table to percentage value
  - Change date format from mm/dd/yyyy to dd-mm-yyyy for order and ship dates

# Data Cleansing: Special symbols to remove example

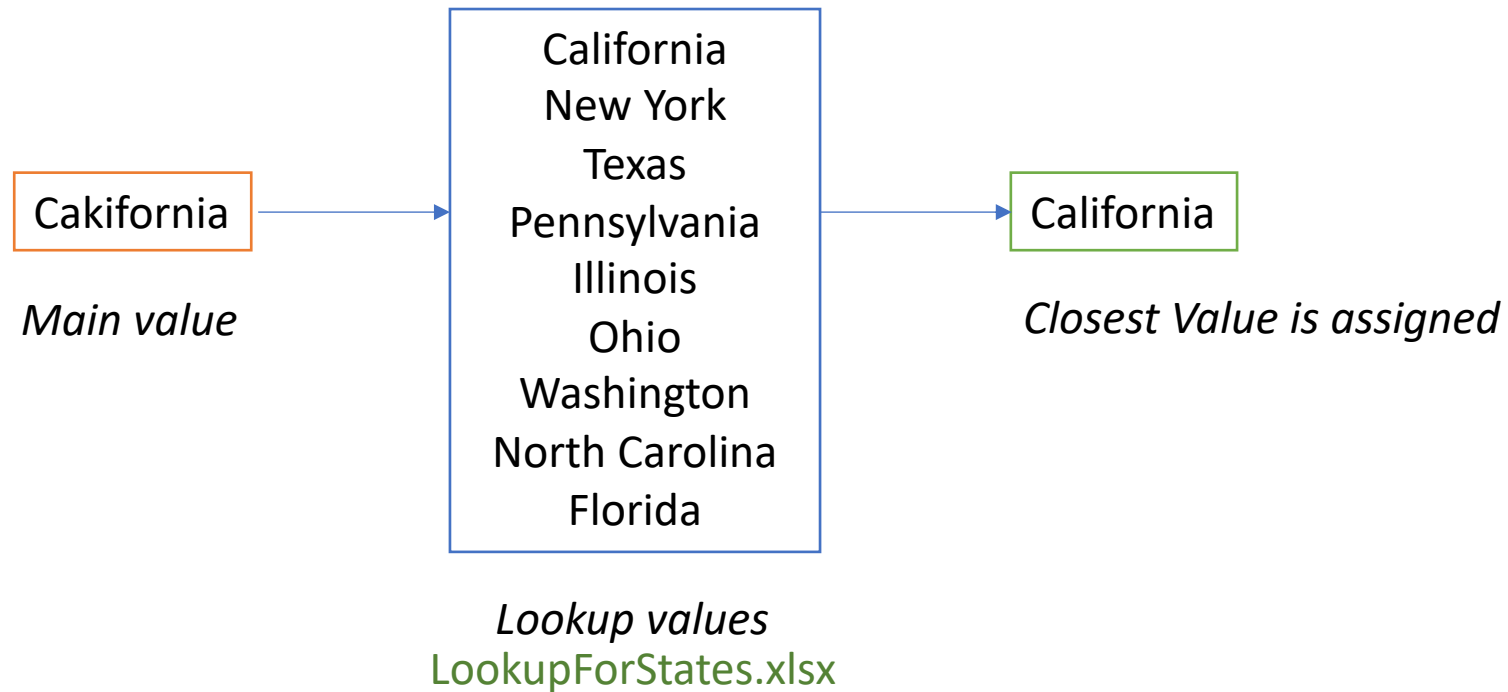


# Data Cleansing: Examples

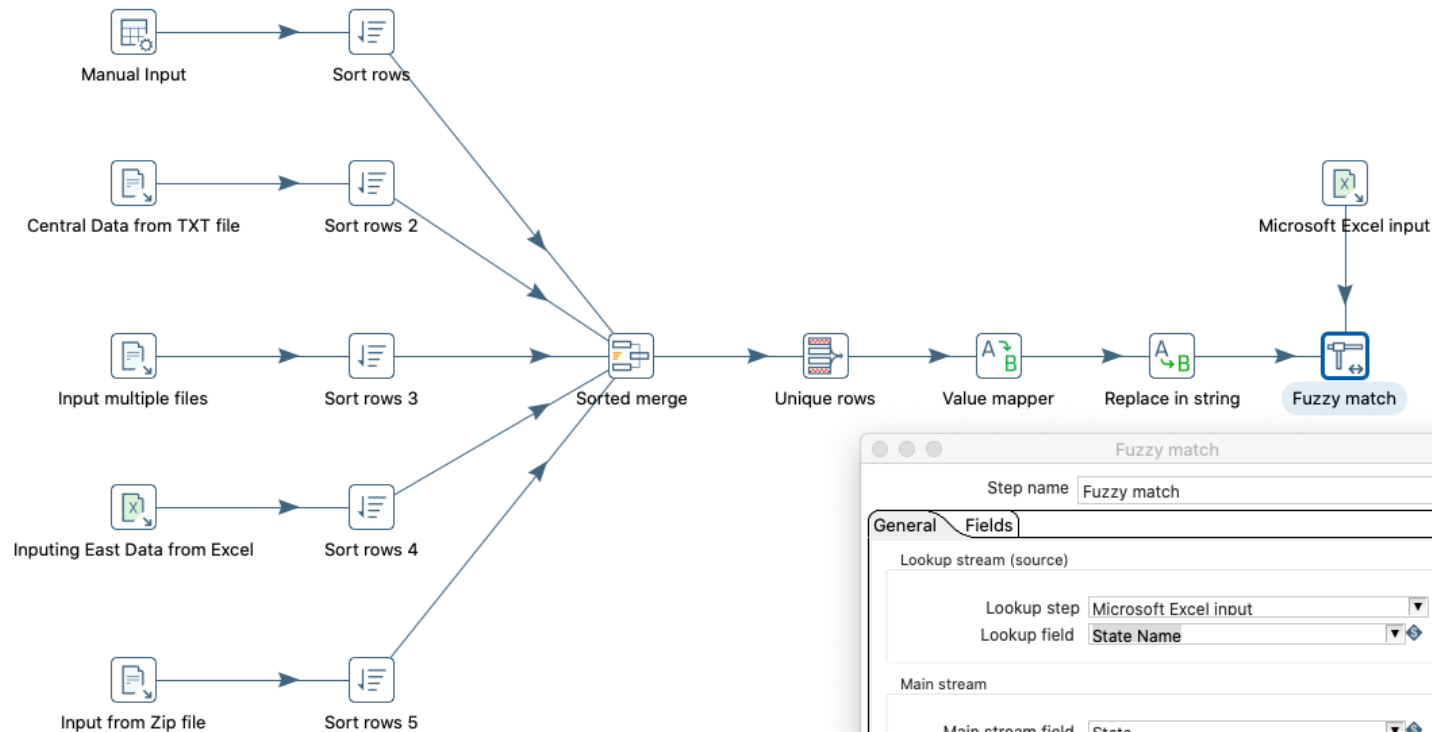
- Multiple names for the same entity:
  - Country field contains US, USA, United States and United States of America to represent one country
- Special symbols to remove:
  - # is present in City column
- Manual entry mistakes:
  - In state column California is sometimes written as Cakifornia and californis etc.
  - Exact Match and Fuzzy match
- Format changes:
  - Changing discount value in sales table to percentage value
  - Change date format from mm/dd/yyyy to dd-mm-yyyy for order and ship dates

# Data Cleansing: Fuzzy match

1. Measures closeness with Lookup values
2. Assign value which is most similar



# Data Cleansing: Fuzzy match example



## Results

Execution History Step Metrics Performance Graph Metrics Preview data

Fuzzy match

Step name Fuzzy match

General Fields

Lookup stream (source)

Lookup step Microsoft Excel input

Lookup field State Name

Main stream

Main stream field State

Settings

Algorithm Levenshtein

Case sensitive ☐

Get closer value ☒

Minimal value 0

Maximal value 2

Values separator ,

Help OK Cancel

Fuzzy match

Step name Fuzzy match

General Fields

Output fields

Match field match

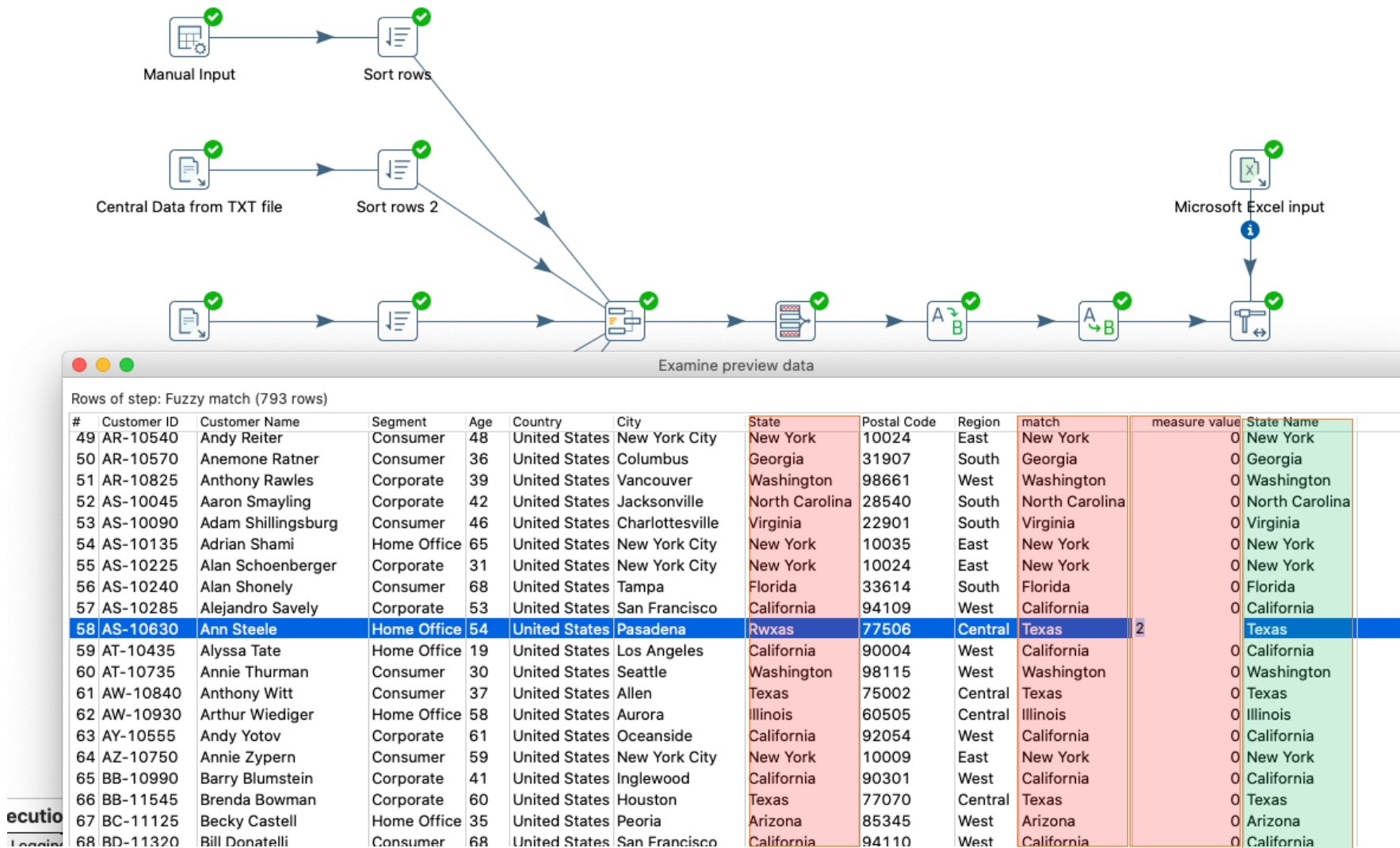
Value field measure value

Specify the additional fields (not mandatory) to retrieve from lookup stream

#	Field	New name
1	State Name	

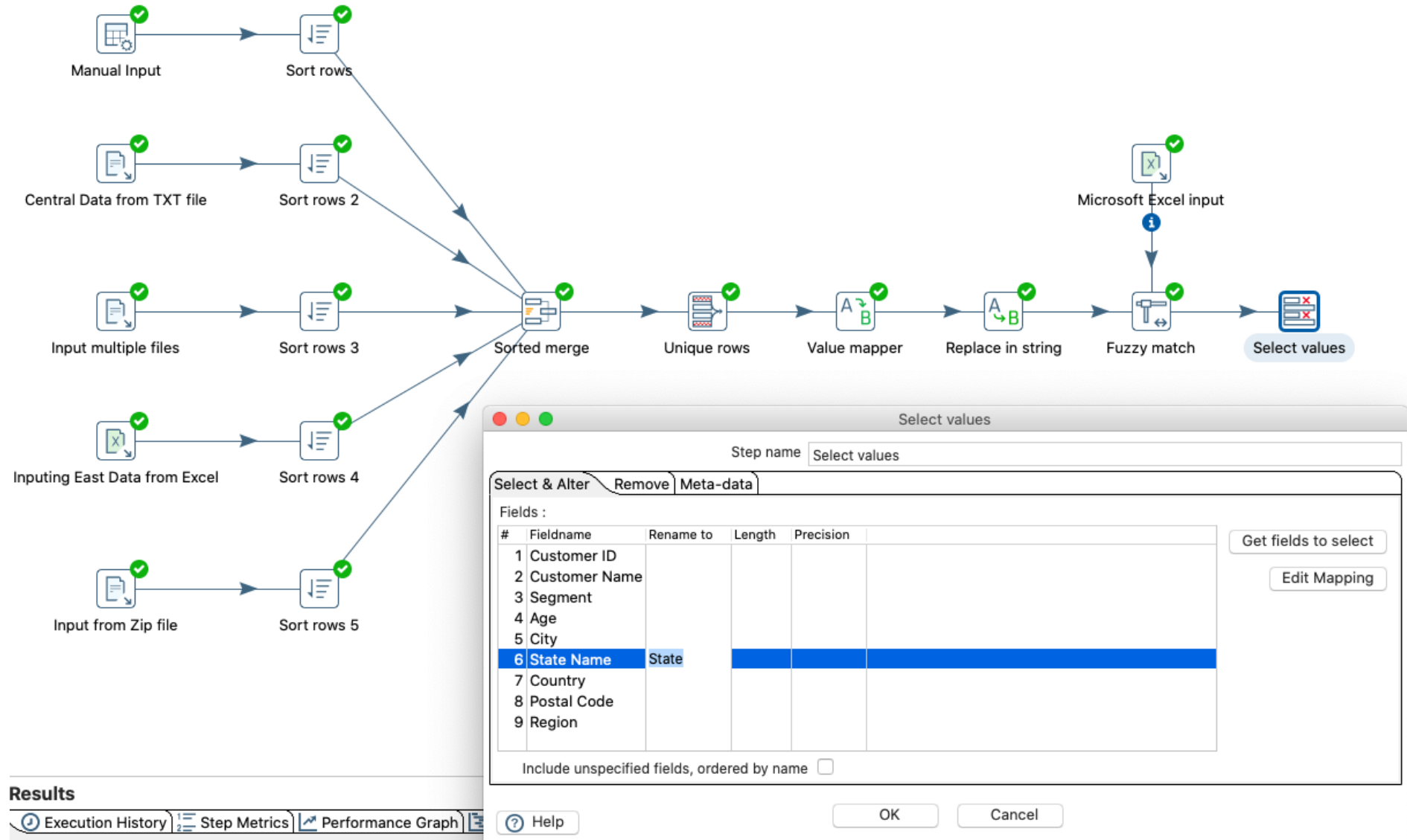
Get fields

# Data Cleansing: Fuzzy match example





# Data Cleansing: Fuzzy match example



# Data Cleansing: Fuzzy match Algorithms

The screenshot shows a software window titled "Fuzzy match". It has a "Step name" field set to "Fuzzy match". Below this are two tabs: "General" and "Fields". The "General" tab is active and contains three sections: "Lookup stream (source)", "Main stream", and "Settings".

In the "Lookup stream (source)" section, the "Lookup step" is set to "Microsoft Excel input" and the "Lookup field" is set to "State Name".

In the "Main stream" section, the "Main stream field" is set to "State".

In the "Settings" section, the "Algorithm" dropdown menu is open, showing a list of options: Levenshtein (highlighted), Damerau Levenshtein, Needleman Wunsch, Jaro, Jaro Winkler, Pair letters Similarity, Metaphone, Double Metaphone, and SoundEx. To the left of the dropdown, there are labels for "Case sensitive", "Get closer value", "Minimal value", "Maximal value", and "Values separator", but they are not currently selected or have their values obscured by the dropdown.

At the bottom left of the window is a "Help" button with a question mark icon.

# Data Cleansing: Fuzzy match Algorithms

## Levenshtein and Damerau-Levenshtein

- Calculates distance by calculating the edit steps
- Steps –Insert, Delete, Replace (**Transpose for Damerau-Levenshtein**)
  - Cakifornia-> California only one step, replace 'k' with 'l'
  - Akiforina-> California needs two steps, add 'c' and replace 'k' with 'l'
  - Cailifornia-> California needs two replace steps as per Levenshtein or one transpose step (il-> li) as per Damerau-Levenshtein

## Needleman-Wunsch

- Score is calculated as penalty
  - Cakifornia-> California will have a score of -1
- Different mismatches can have different weights

# Data Cleansing: Fuzzy match Algorithms

## Jaro and Jaro-Winkler

- Calculates similarity index between 0 and 1
- 0 –no similarity and 1 –completely similar

*How similar are CALIFORNIA and FLORIDA?*

Levenshtein distance –7  
Jaro similarity score of 0

## Pair letters similarity

- Example: find similarity between FLORIDA and FLOTISA
  1. Make two character pairs from both strings
    - FL, LO, OR, RI, ID, DA
    - FL, LO, OT, TI, IS, SA
  2. Calculate score using the formula
  3. Score = Total Pairs Matched/ Total Pairs = 4/ 12 = 0.33

# Data Cleansing: Fuzzy match Algorithms

## **Metaphone, Double Metaphone, Soundex, and RefinedSoundEx**

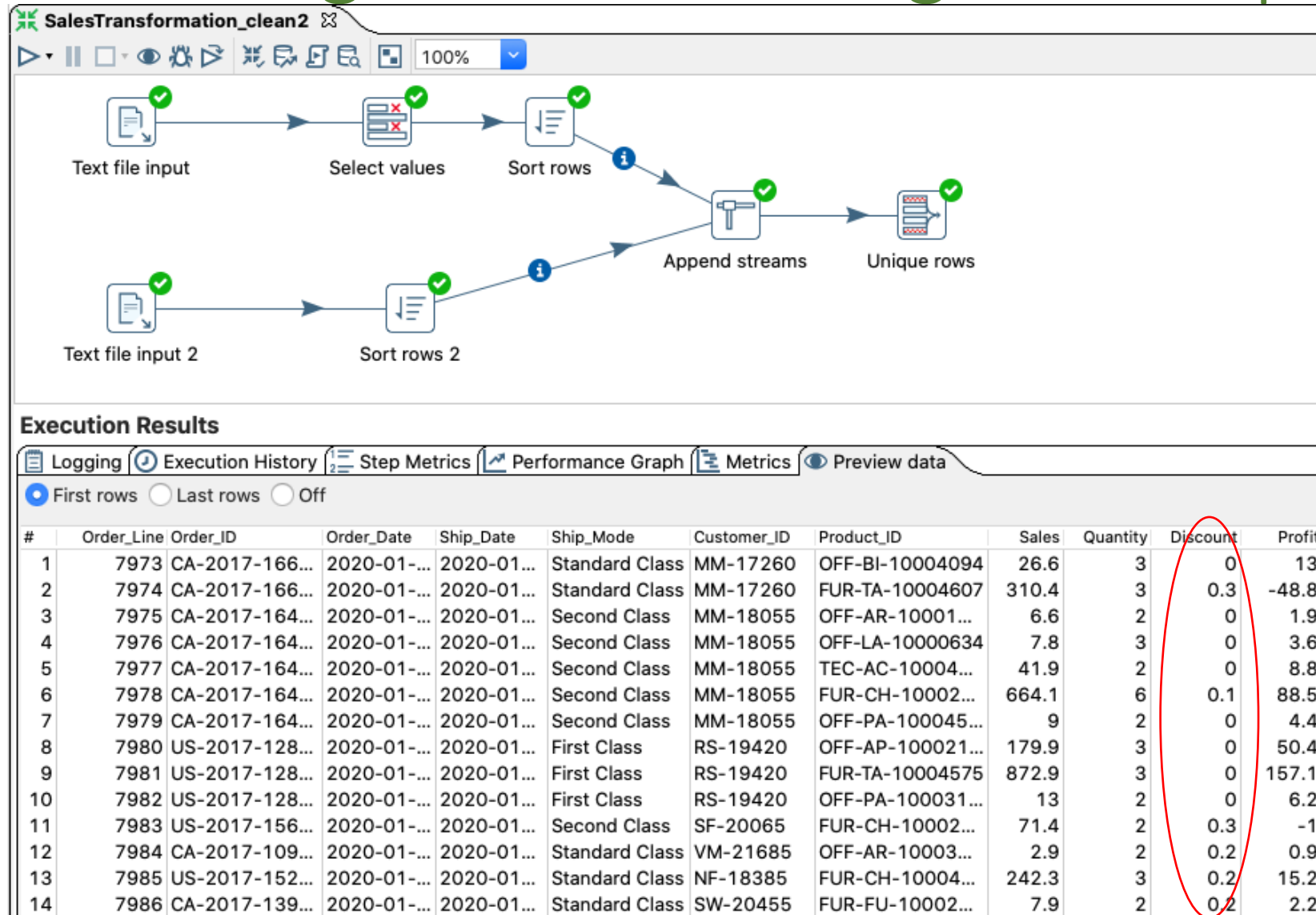
- 'Phonetic' Algorithms, try to match the sound of words
- Commonly used for deduplication
- Only applicable on English language
- Useful when the errors are due to persons not knowing exact spellings and inputting strings based on their sounds

# Data Cleansing: Examples

- Multiple names for the same entity:
  - Country field contains US, USA, United States and United States of America to represent one country
- Special symbols to remove:
  - # is present in City column
- Manual entry mistakes:
  - In state column California is sometimes written as Cakifornia and californis etc.
  - Exact Match and Fuzzy match
- Format changes:
  - Changing discount value in sales table to percentage value
  - Change date format from mm/dd/yyyy to dd-mm-yyyy for order and ship dates

# Data Cleansing: Format changes example

Sales  
Transformation



Discount  
→ in percentage

# Data Cleansing: Format changes example

Sales  
Transformation

The screenshot displays a data transformation tool interface. The top section shows a workflow diagram with the following steps: Text file input, Select values, Sort rows, Append streams, Unique rows, and Formula. A second path starts with Text file input 2, followed by Sort rows 2, which also feeds into the Append streams step. The bottom section, titled 'Execution Results', shows a table of data with columns: #, Order\_Line, Order\_ID, Order\_Date, Ship\_Date, Ship\_Mode, Customer\_ID, Product\_ID, Sales, Quantity, Discount, and Profit. The first 11 rows of data are visible. Two red circles are drawn around the 'Order\_Date' and 'Ship\_Date' columns, highlighting the date format change from 'mm/dd/yyyy' to 'dd-mm-yyyy'.

**Execution Results**

Logging Execution History Step Metrics Performance Graph Metrics Preview data

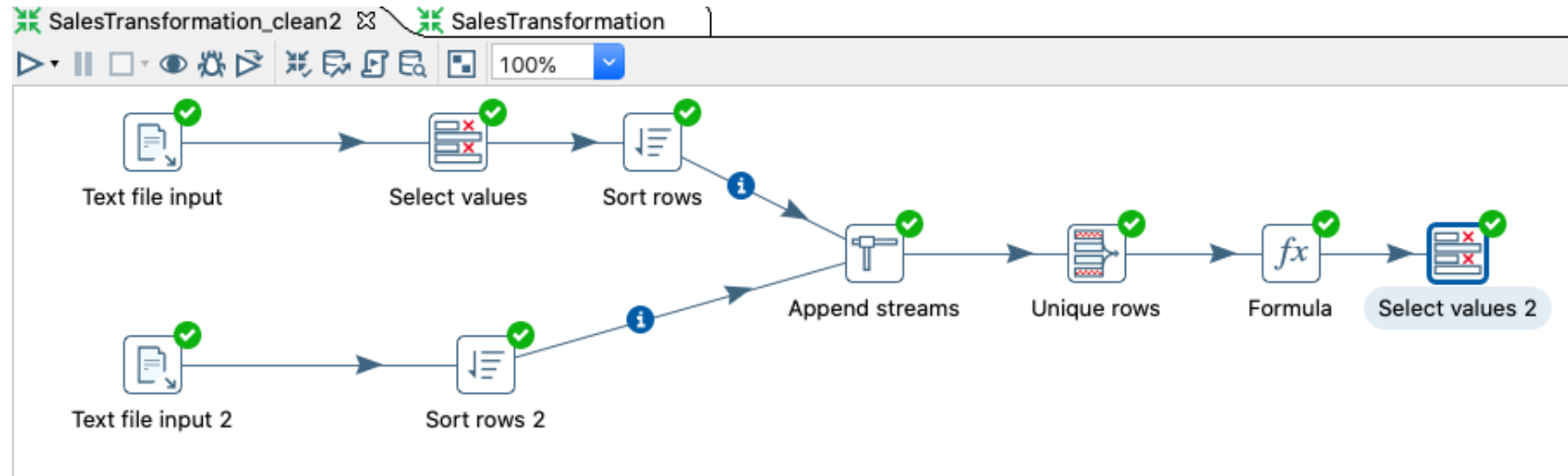
☒ First rows ☐ Last rows ☐ Off

#	Order_Line	Order_ID	Order_Date	Ship_Date	Ship_Mode	Customer_ID	Product_ID	Sales	Quantity	Discount	Profit
1	1	CA-2014-103...	06/21/2016	06/25/20...	Standard Class	DP-13000	OFF-PA-100001...	16.4	2	20	5.5
2	2	CA-2014-112...	06/22/2016	06/26/20...	Standard Class	PO-19195	OFF-LA-10003223	11.8	3	20	4.3
3	3	CA-2014-112...	06/22/2016	06/26/20...	Standard Class	PO-19195	OFF-ST-10002743	272.7	3	20	-64.8
4	4	CA-2014-112...	06/22/2016	06/26/20...	Standard Class	PO-19195	OFF-BI-10004094	3.5	2	80	-5.5
5	5	CA-2014-141...	06/23/2016	06/30/20...	Standard Class	MB-18085	OFF-AR-10003...	19.5	3	20	4.9
6	6	CA-2014-130...	06/24/2016	06/26/20...	Second Class	LS-17230	OFF-PA-100020...	19.4	3	0	9.3
7	7	CA-2014-106...	06/24/2016	06/25/20...	First Class	JO-15145	OFF-AR-10002...	12.8	3	0	5.2
8	8	CA-2014-167...	06/24/2016	06/28/20...	Standard Class	ME-17320	FUR-CH-10004...	2573.8	9	0	746.4
9	9	CA-2014-167...	06/24/2016	06/28/20...	Standard Class	ME-17320	OFF-BI-10004632	610	2	0	274.5
10	10	CA-2014-167...	06/24/2016	06/28/20...	Standard Class	ME-17320	OFF-AR-10001...	5.5	2	0	1.5
11	11	CA-2014-167...	06/24/2016	06/28/20...	Standard Class	ME-17320	OFF-AR-10001...	5.5	2	0	1.5

date format  
mm/dd/yyyy →  
dd-mm-yyyy



# Data Cleansing: Format changes example



## Execution Results

Execution Results											
Logging Execution History Step Metrics Performance Graph Metrics Preview data											
First rows Last rows Off											
#	Order_Line	Order_ID	Order_Date	Ship_Date	Ship_Mode	Customer_ID	Product_ID	Sales	Quantity	Discount	Profit
1		1 CA-2014-103...	21-06-20...	25-06-2...	Standard Class	DP-13000	OFF-PA-100001...	16.4	2	20	5.5
2		2 CA-2014-112...	22-06-20...	26-06-2...	Standard Class	PO-19195	OFF-LA-10003223	11.8	3	20	4.3
3		3 CA-2014-112...	22-06-20...	26-06-2...	Standard Class	PO-19195	OFF-ST-10002743	272.7	3	20	-64.8
4		4 CA-2014-112...	22-06-20...	26-06-2...	Standard Class	PO-19195	OFF-BI-10004094	3.5	2	80	-5.5
5		5 CA-2014-141...	23-06-20...	30-06-2...	Standard Class	MB-18085	OFF-AR-10003...	19.5	3	20	4.9
6		6 CA-2014-130...	24-06-20...	26-06-2...	Second Class	LS-17230	OFF-PA-100020...	19.4	3	0	9.3
7		7 CA-2014-106...	24-06-20...	25-06-2...	First Class	JO-15145	OFF-AR-10002...	12.8	3	0	5.2
8		8 CA-2014-167...	24-06-20...	28-06-2...	Standard Class	ME-17320	FUR-CH-10004...	2573.8	9	0	746.4
9		9 CA-2014-167...	24-06-20...	28-06-2...	Standard Class	ME-17320	OFF-BI-10004632	610	2	0	274.5
10		10 CA-2014-167...	24-06-20...	28-06-2...	Standard Class	ME-17320	OFF-AR-10001...	5.5	2	0	1.5
11		11 CA-2014-167...	24-06-20...	28-06-2...	Standard Class	ME-17320	TEC-PH-10004...	392	2	0	113.7
12		12 CA-2014-167...	24-06-20...	28-06-2...	Standard Class	ME-17320	TEC-PH-10004...	756	4	0	204.1

# Data Cleansing: Common steps

Scenario	Step
Value must have a particular <b>format</b>	Select values
Values in multiple columns are to be <b>combined</b> into a single column	Concat fields
Assign new value basis the value of a field containing number	Number range
Assign new value basis the value of a field containing a string	Value mapper
Remove <b>duplicates</b>	Unique rows
Remove/ change special characters or part of <b>strings</b>	Replace in string