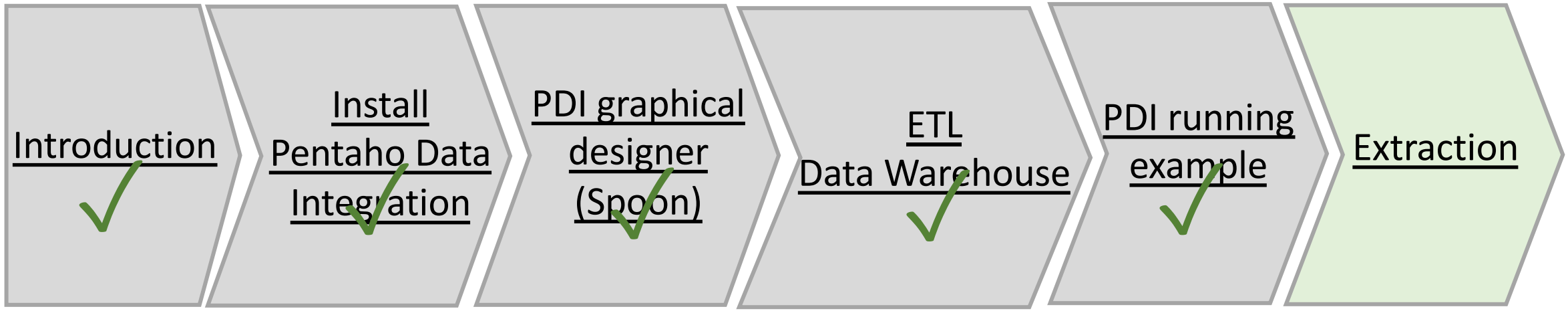
DATA ANALYTICS
(Data Warehouse)
Pentaho Data Integration

Luca Cinelli, PhD

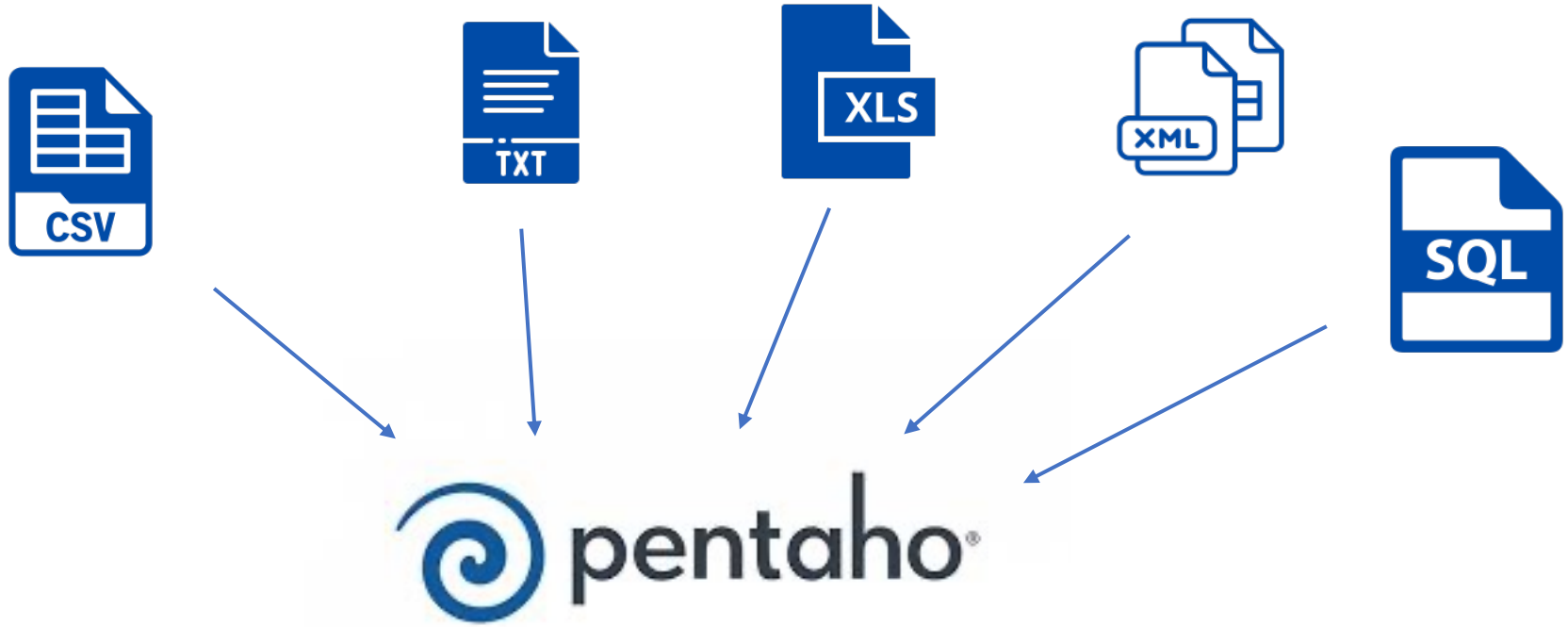
luca.cinelli@unical.it

Outline


Introduction
Install Pentaho Data Integration
PDI graphical designer (Spoon)
ETL Data Warehouse
PDI running example
Extraction

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

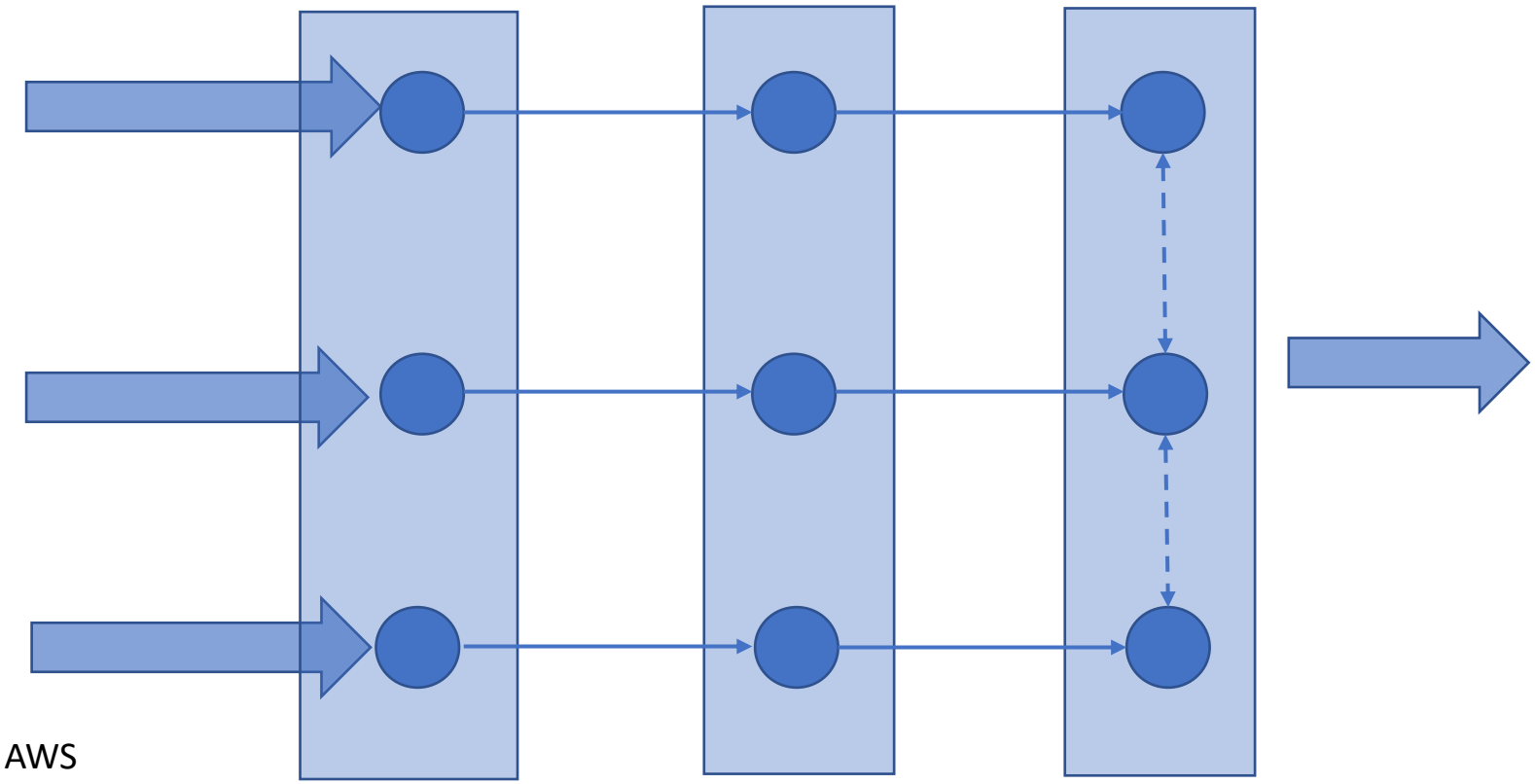
XML

JSON

Sales Data

SQL based DB

Cloud based Storage AWS



3. Cleaning and Validation

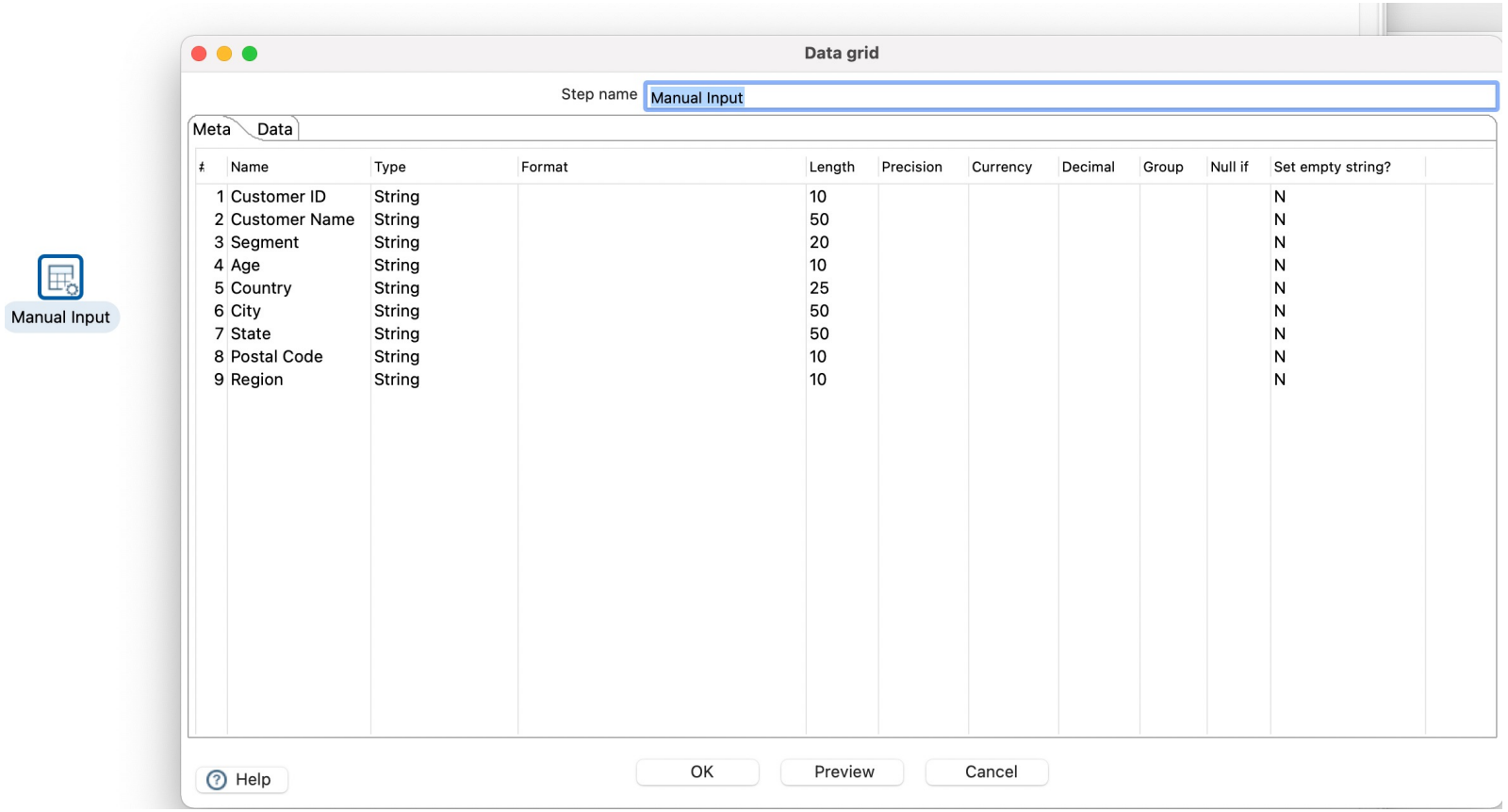
2. Merging

4. Transformation, aggregation and Joining

5. Loading to create a DataMart

Extraction from tabular format

- Manually entering data into PDI (ManualInput.ktr)



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)

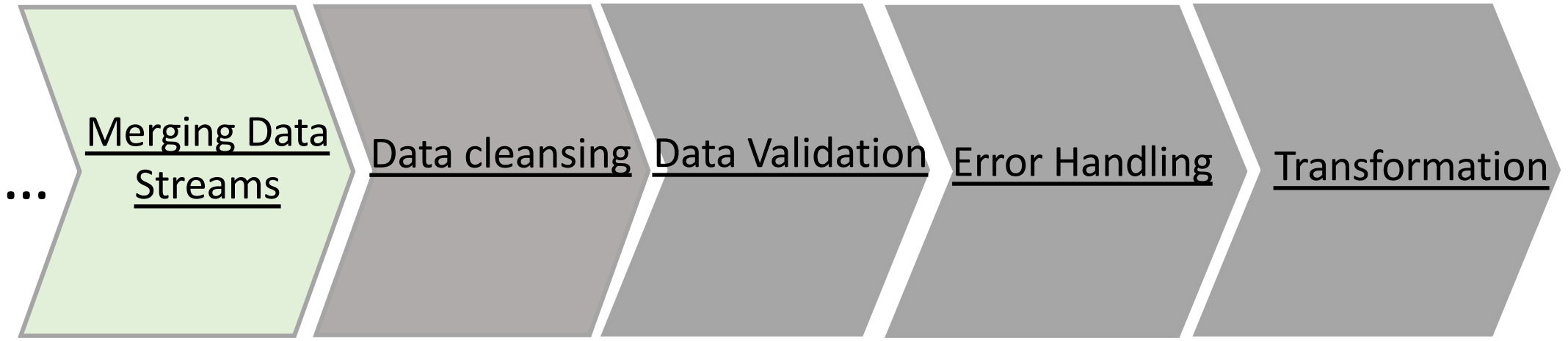
Extraction from no-tabular format

- 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)

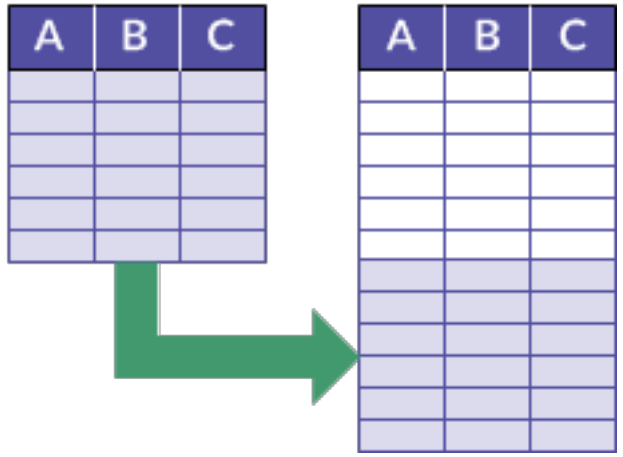
Outline


...
Merging Data Streams
Data cleansing
Data Validation
Error Handling
Transformation

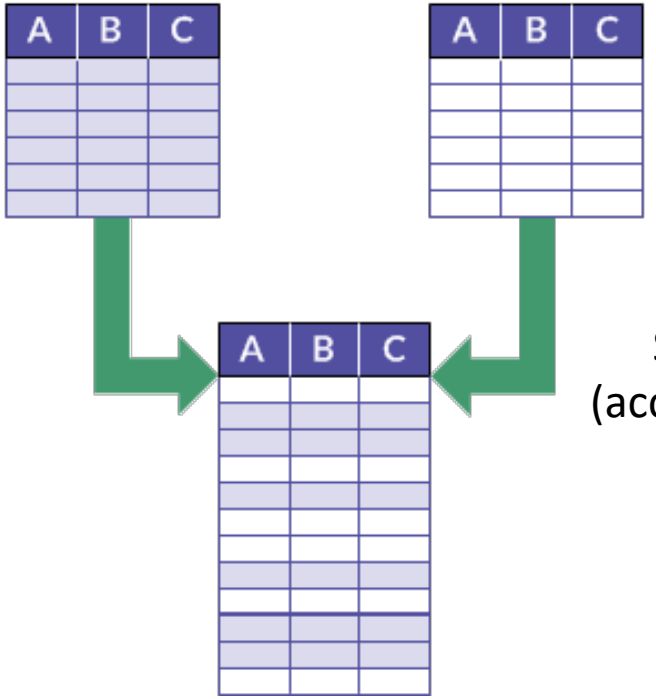
Merging Data Streams

Merging Streams of Data

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



Append
(e.g. merging sales data)

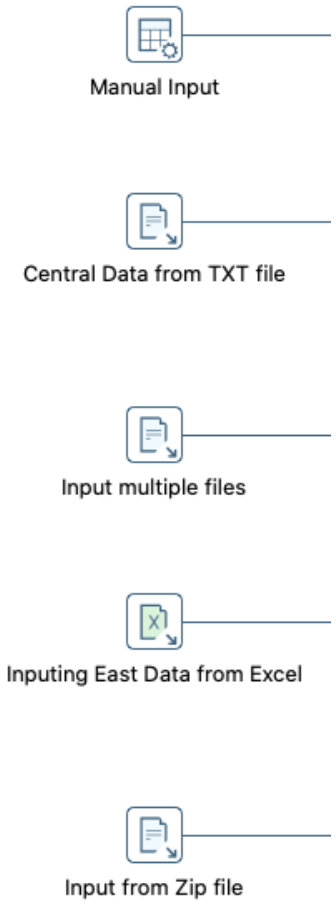


Sorted Merge
(according to one of the columns)

Merging Streams of Data

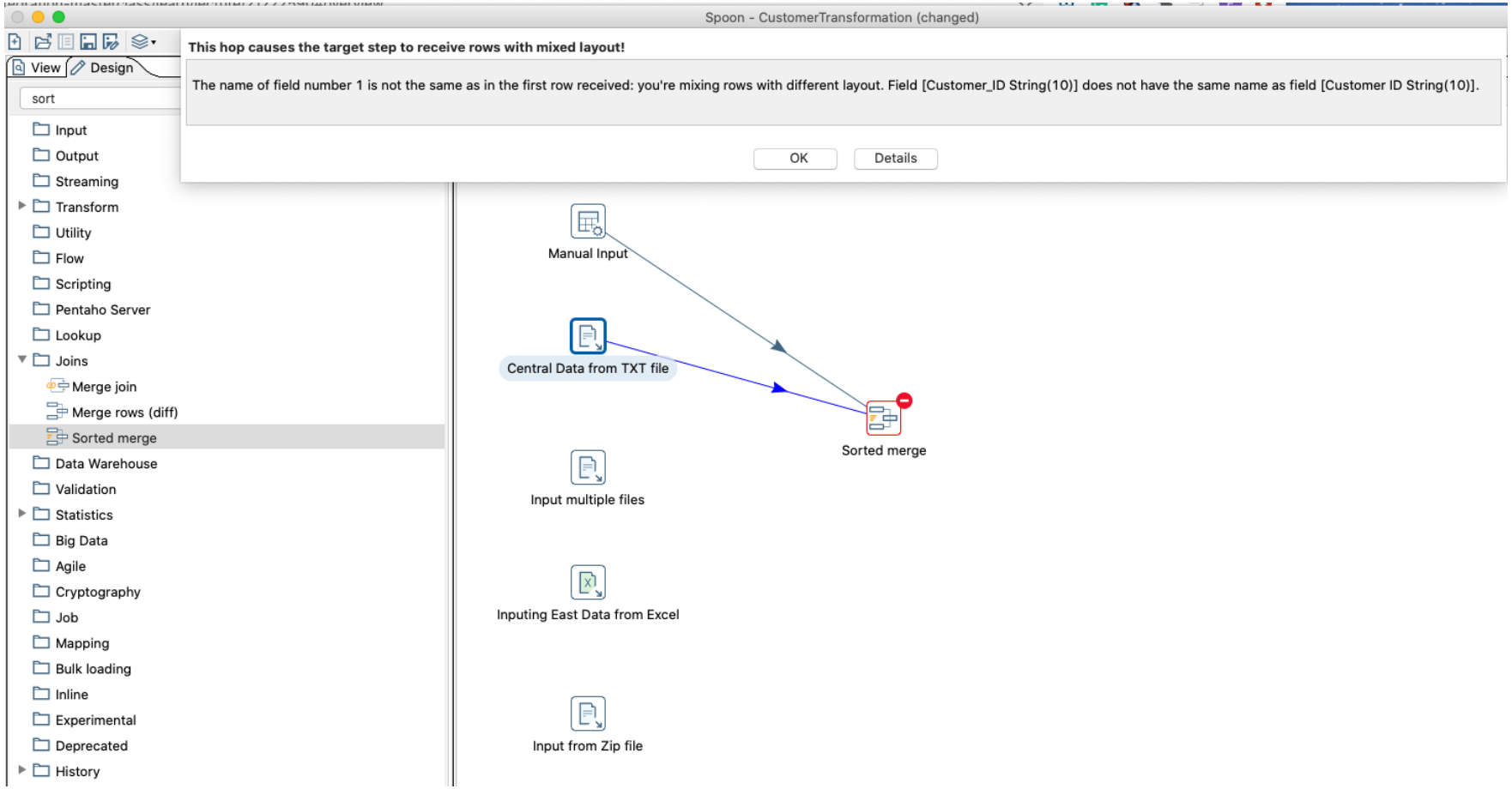
- Merged streams may have duplicates: do deduplication on the primary key column
- Ensure unique occurrence of primary key after merging
  - Same tuple => Delete one duplicated tuple
  - Same primary key but different tuples… Primary key should be unique => Error Handling
- Sort data before deduplicating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

Merging Streams of Data: Customer data example

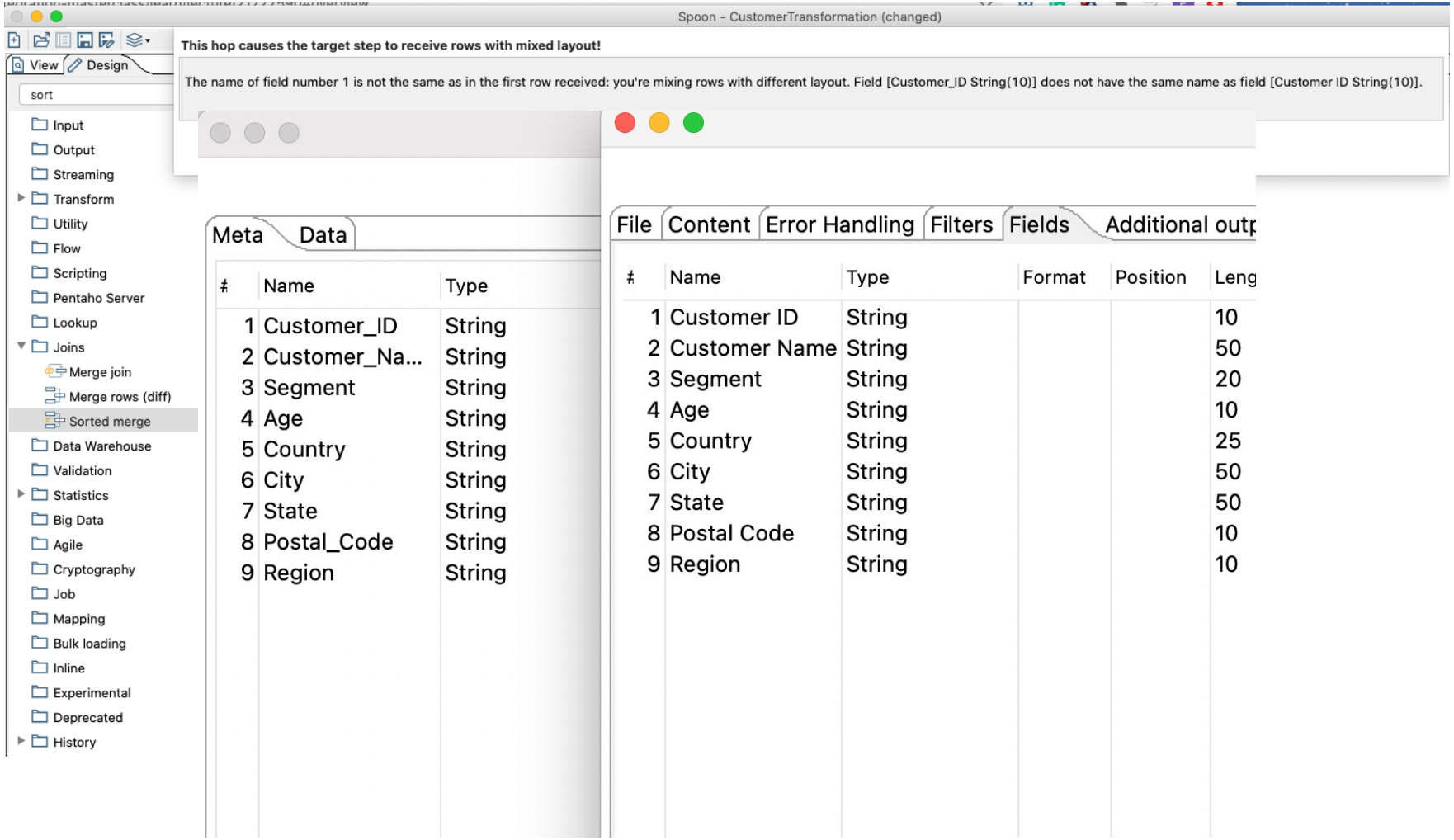


Search for “sort” in design Transformation search bar

Merging Streams of Data: Customer data example

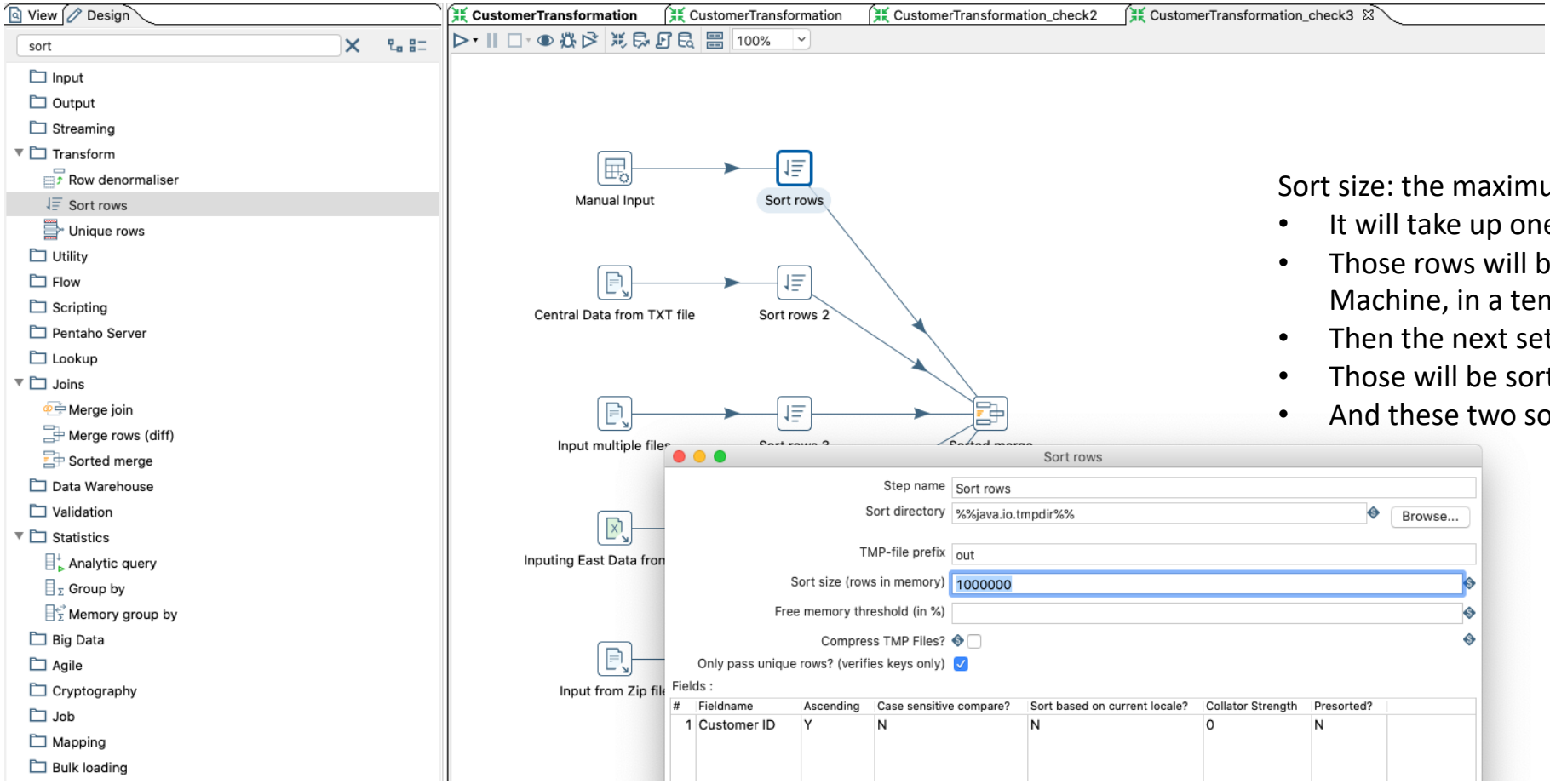


Merging Streams of Data: Customer data example



Merging Streams of Data: Customer data example

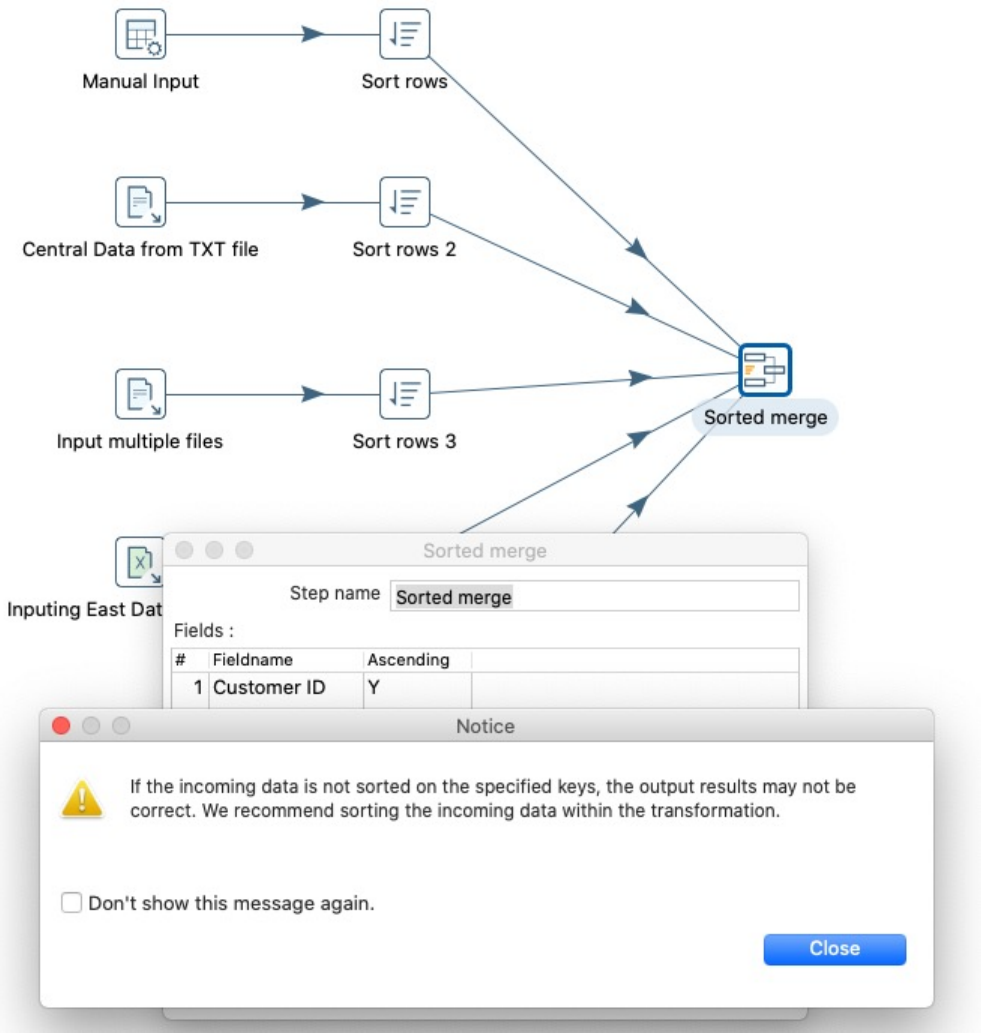
Require to sort individual streams before merging



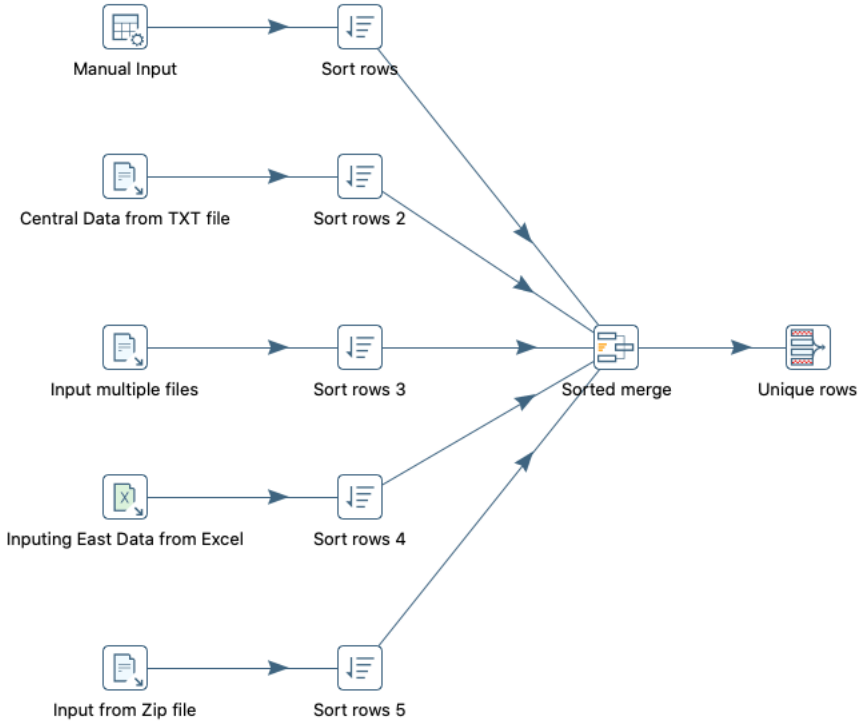
Sort size: the maximum number of rows sorted at s time:

- 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.

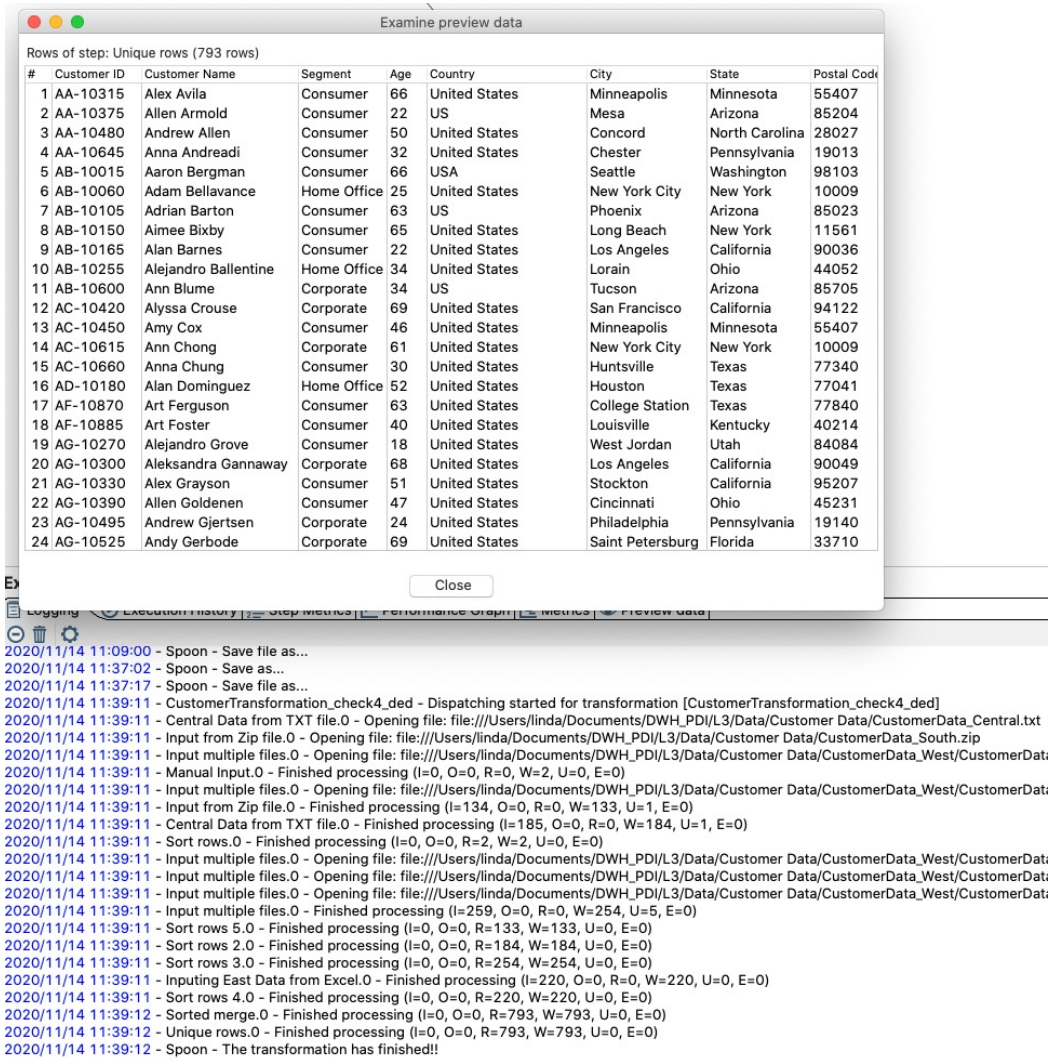
Merging Streams of Data: Customer data example



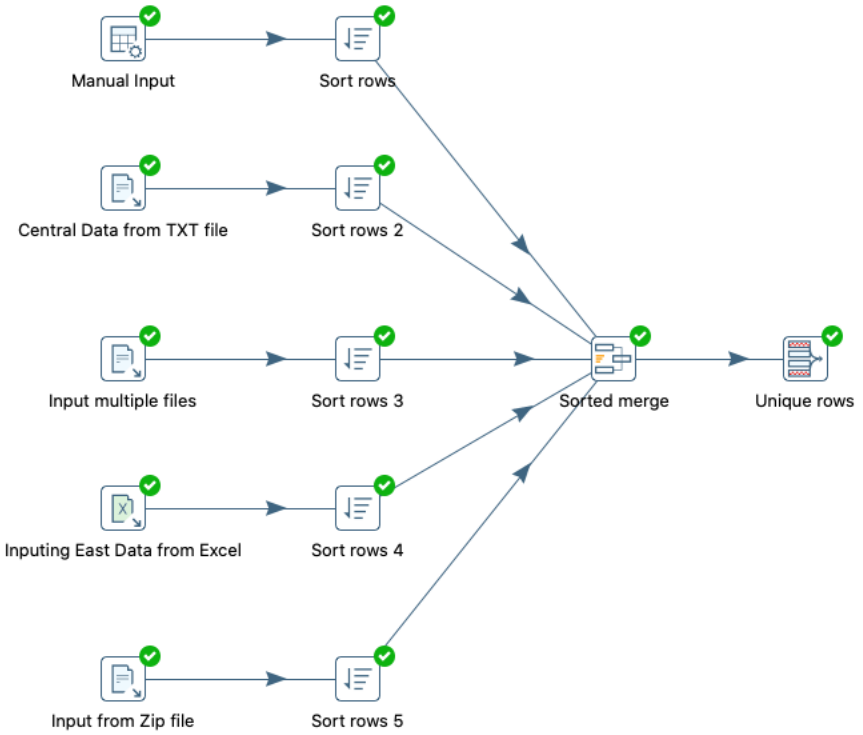
Merging Streams of Data: Customer data example



Preview



Merging Streams of Data: Customer data example



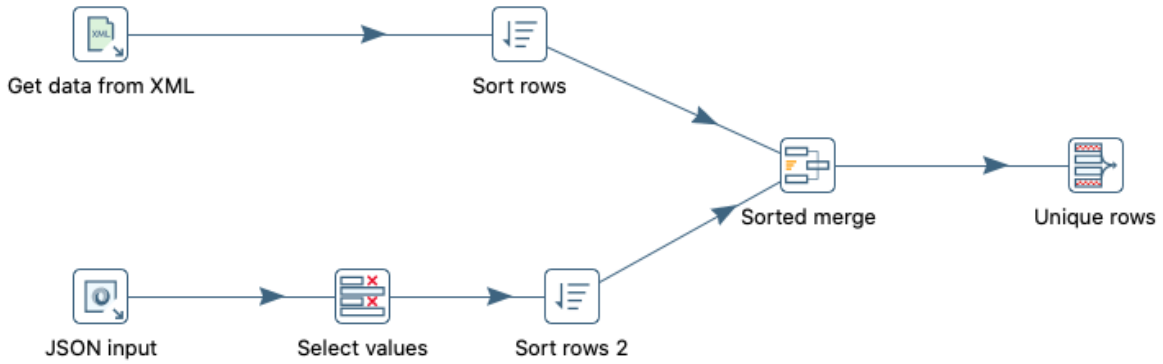
Execution Results

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	-

CustomerTransformation_check4_de.ktr

Merging Streams of Data: Product data example



Get data from XML

Step name: Get 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

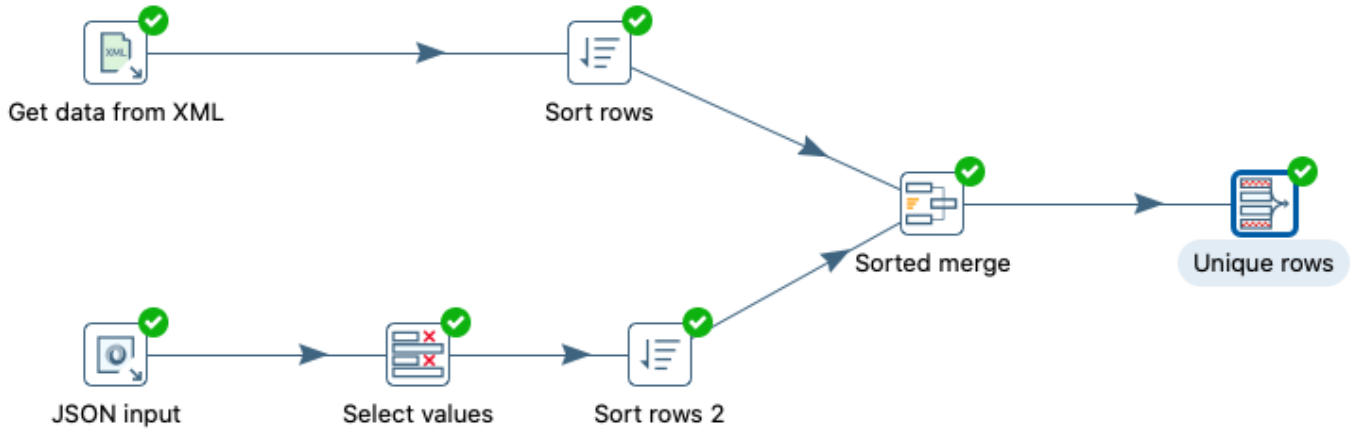
JSON input

Step name: JSON input

File | Content | Fields | Additional output fields

#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Rep
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



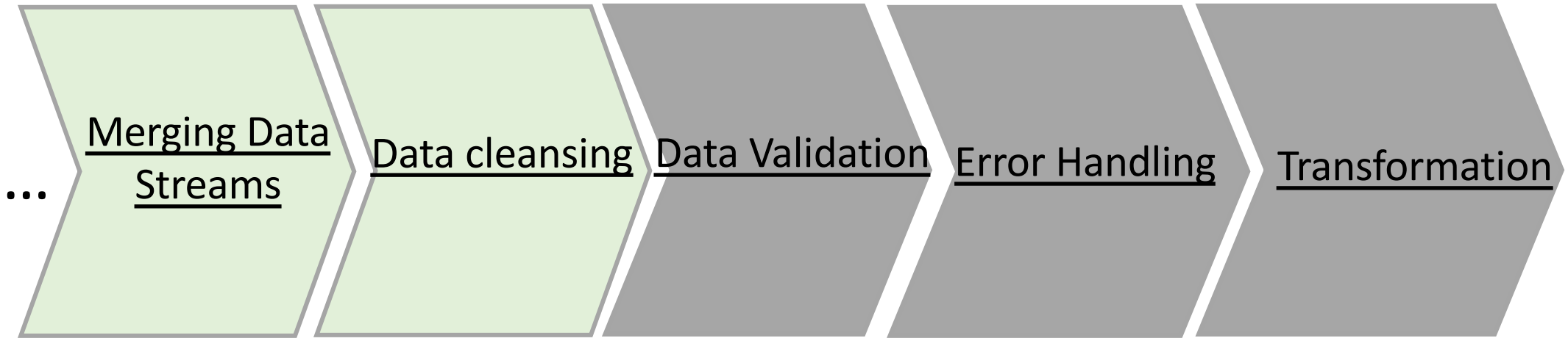
Execution Results

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	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.

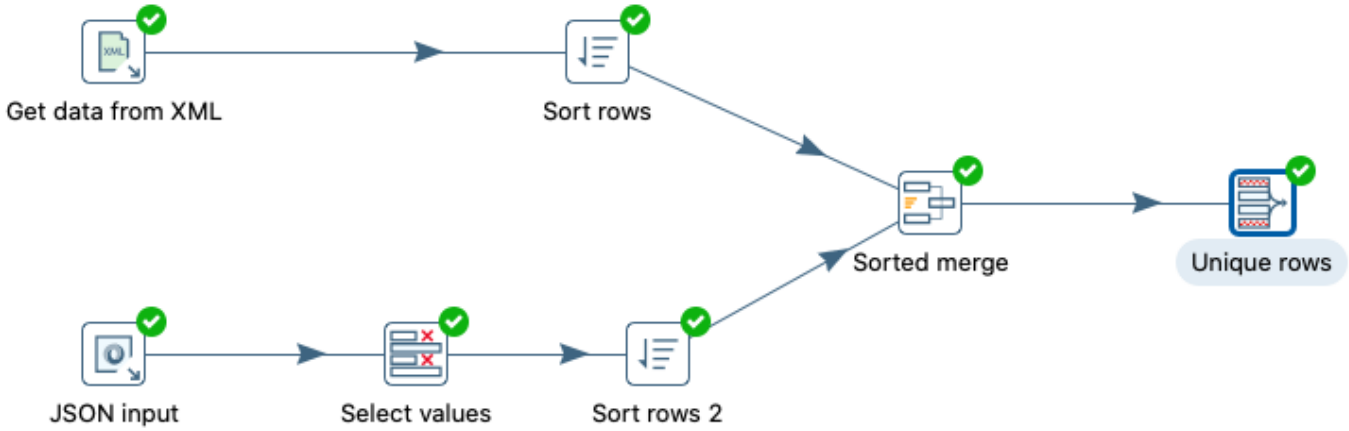
Outline


...
Merging Data Streams
Data cleansing
Data Validation
Error Handling
Transformation

Data Cleansing

Data Cleansing: Remove duplicates example

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



Data Cleansing

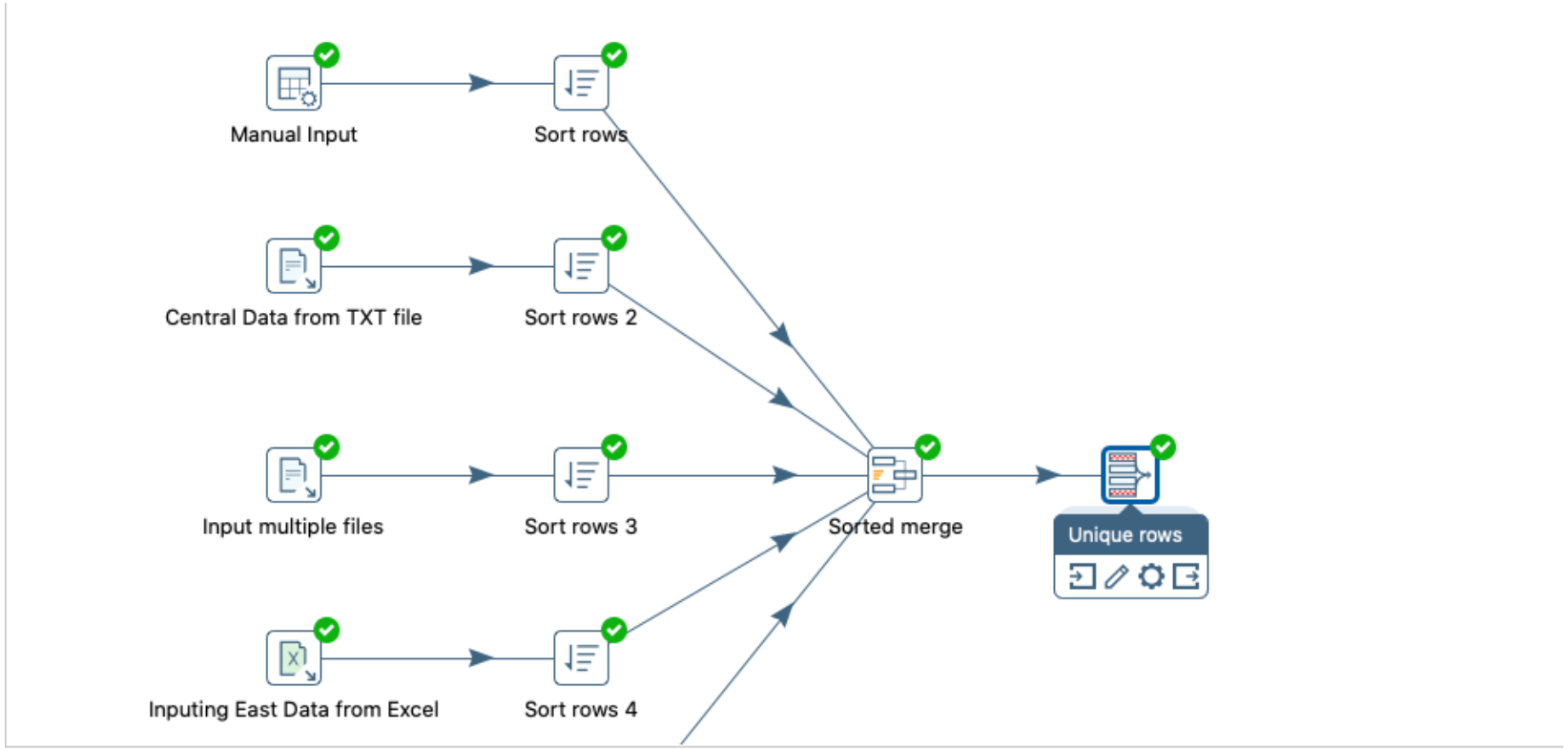
- Correcting small mistakes such as typing mistakes or data format related issues.
  - Examples: 5 vs 5.0, 16 Nov vs 16/11/2020, duplicate entries, etc.

- 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



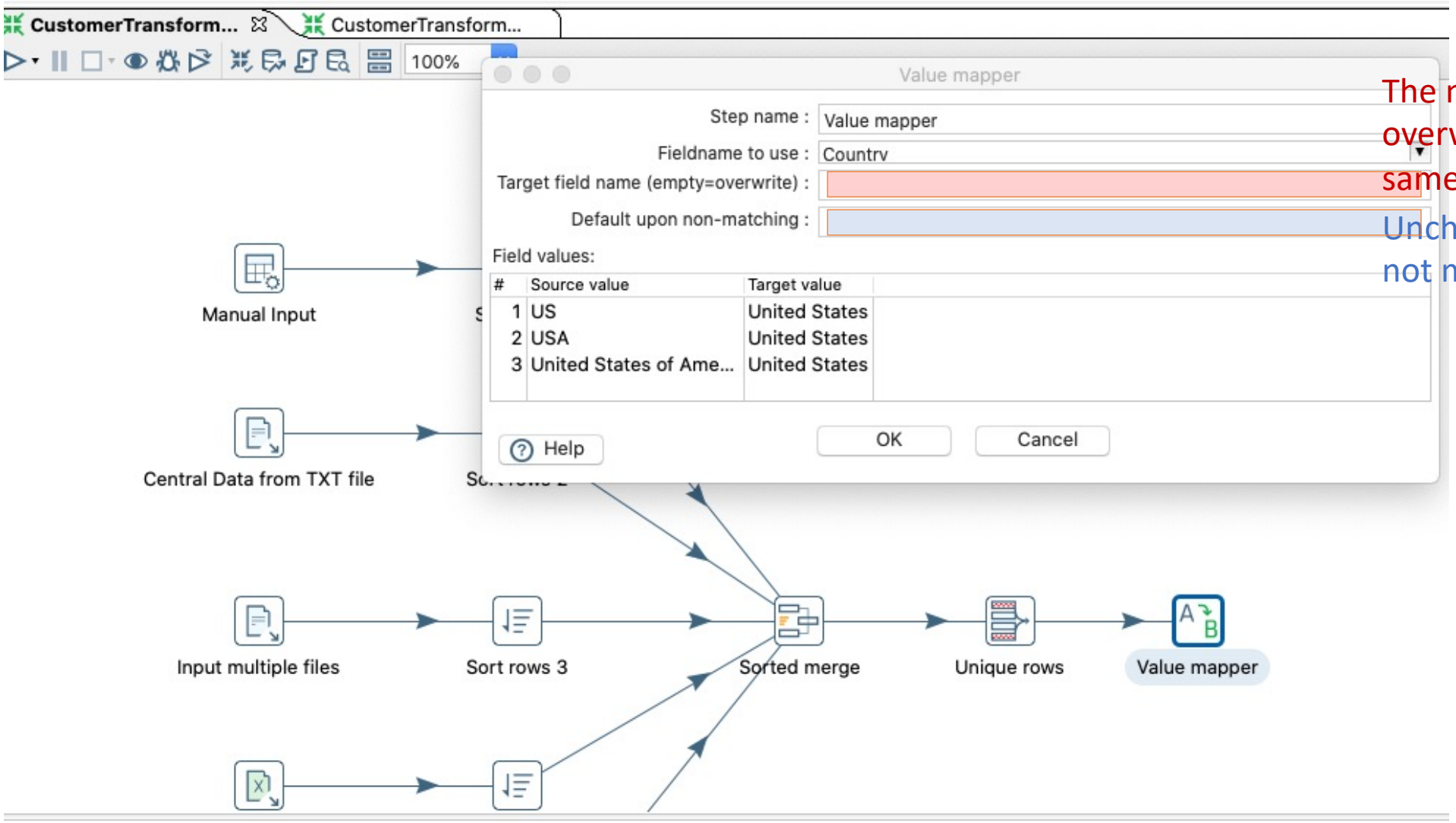
Execution Results

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 Armold	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	25	United States	New York City	New York	10009	East

Data Cleansing: Multiple names example



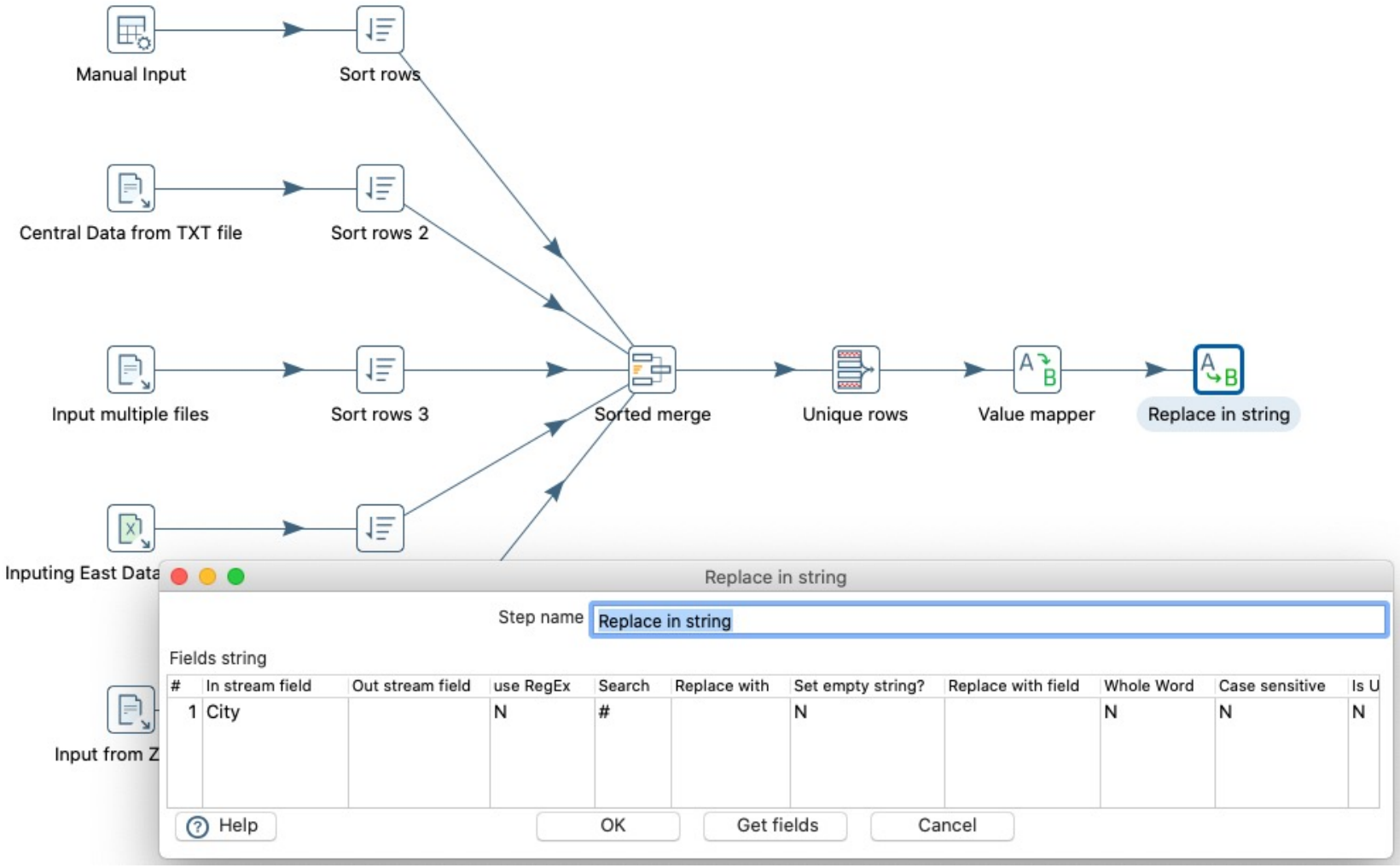
The new value will be overwritten in the same columns

Unchanged if we do not match

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

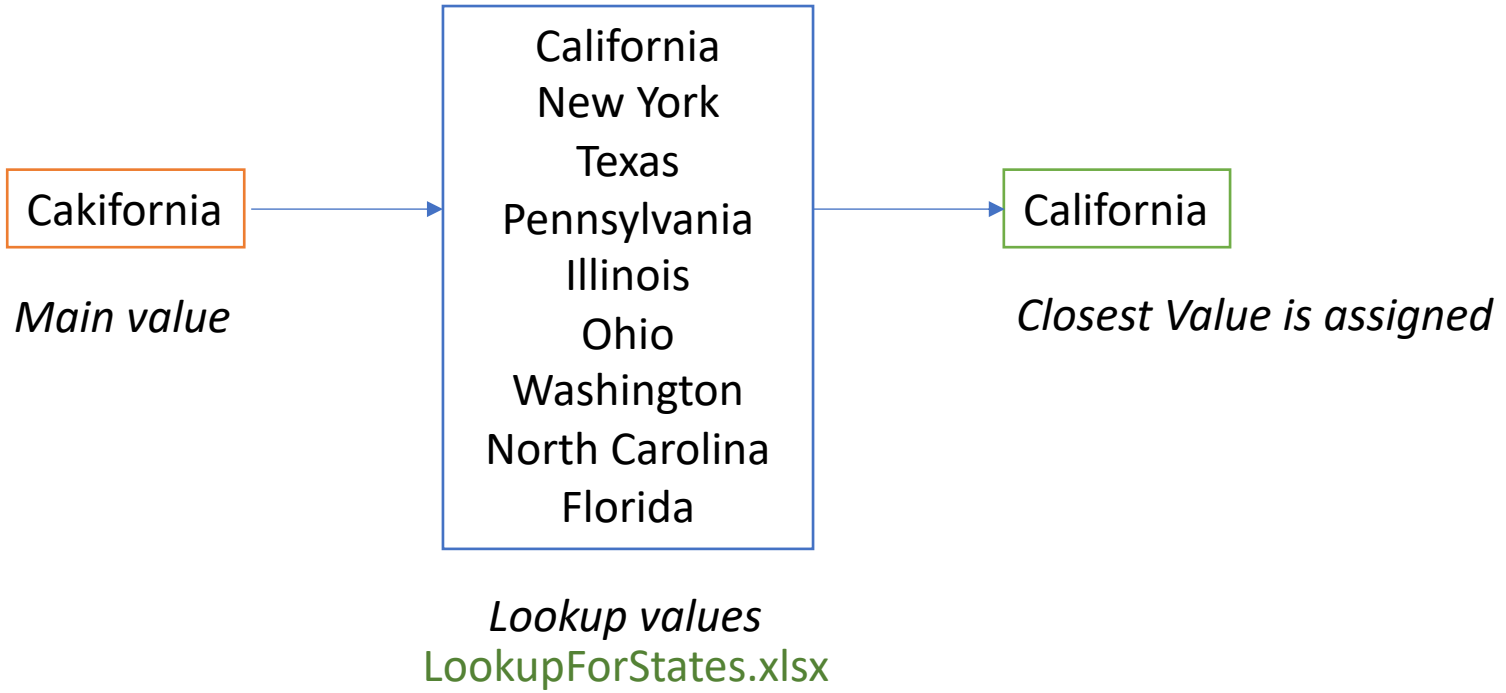


Data Cleansing: Examples

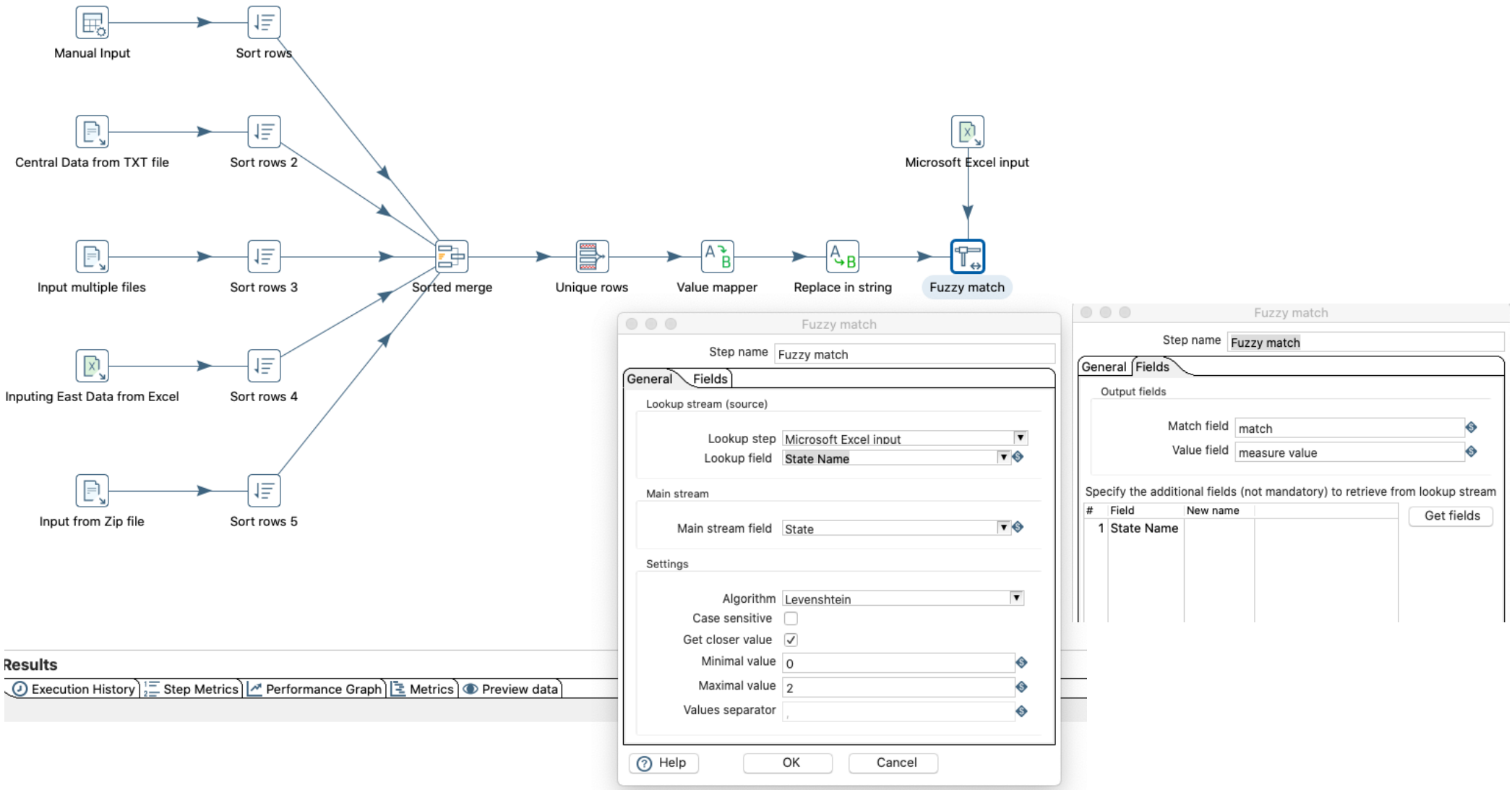
- Multiple names for the same entity:
  - Country field contains US, USA, United States and United States of America to represent one country
- Special symbols to remove:
  - # is present in City column
- Manual entry mistakes:
  - In state column California is sometimes written as Cakifornia and californis 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

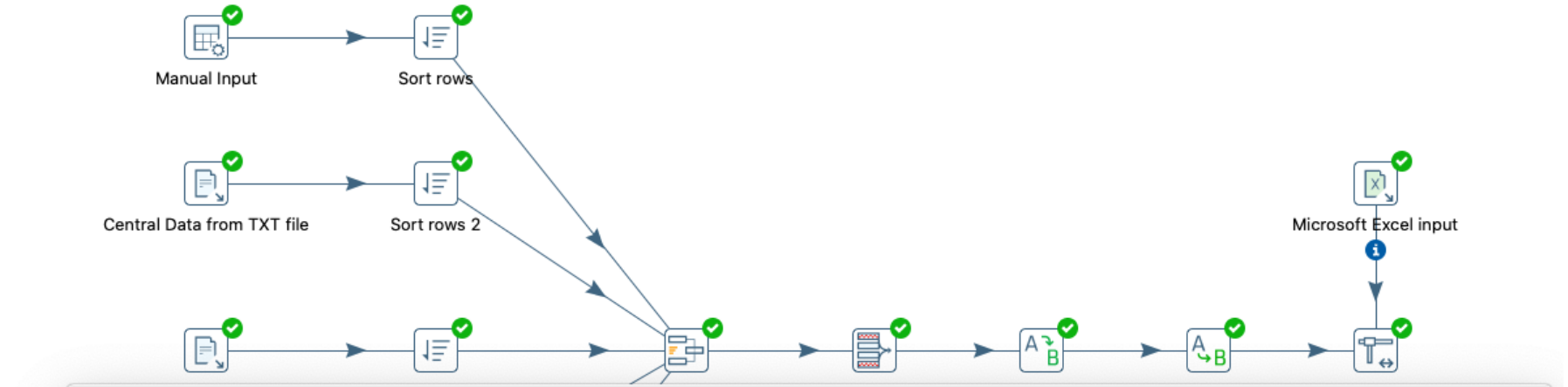
1. Measures closeness with Lookup values
2. Assign value which is most similar



Data Cleansing: Fuzzy match example



Data Cleansing: Fuzzy match example

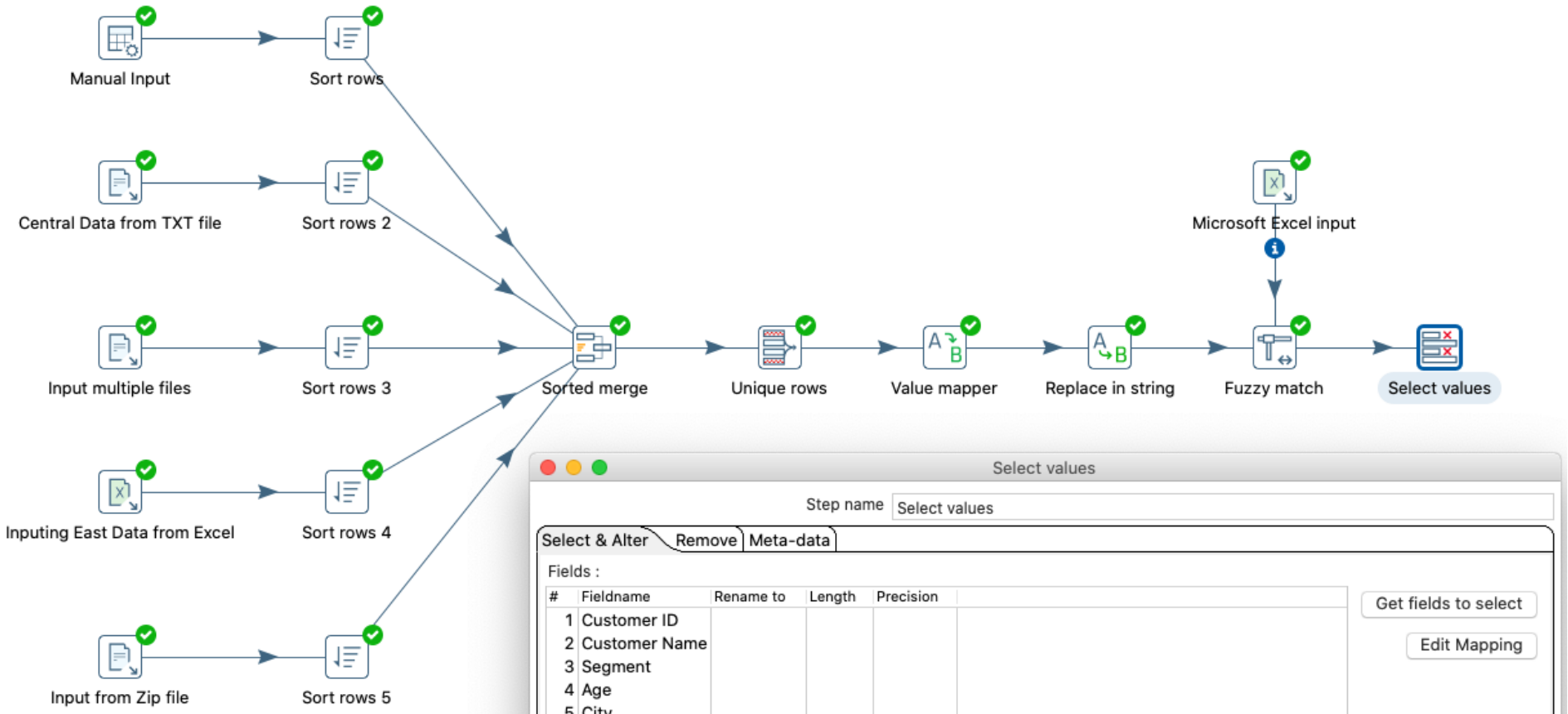


Examine preview data

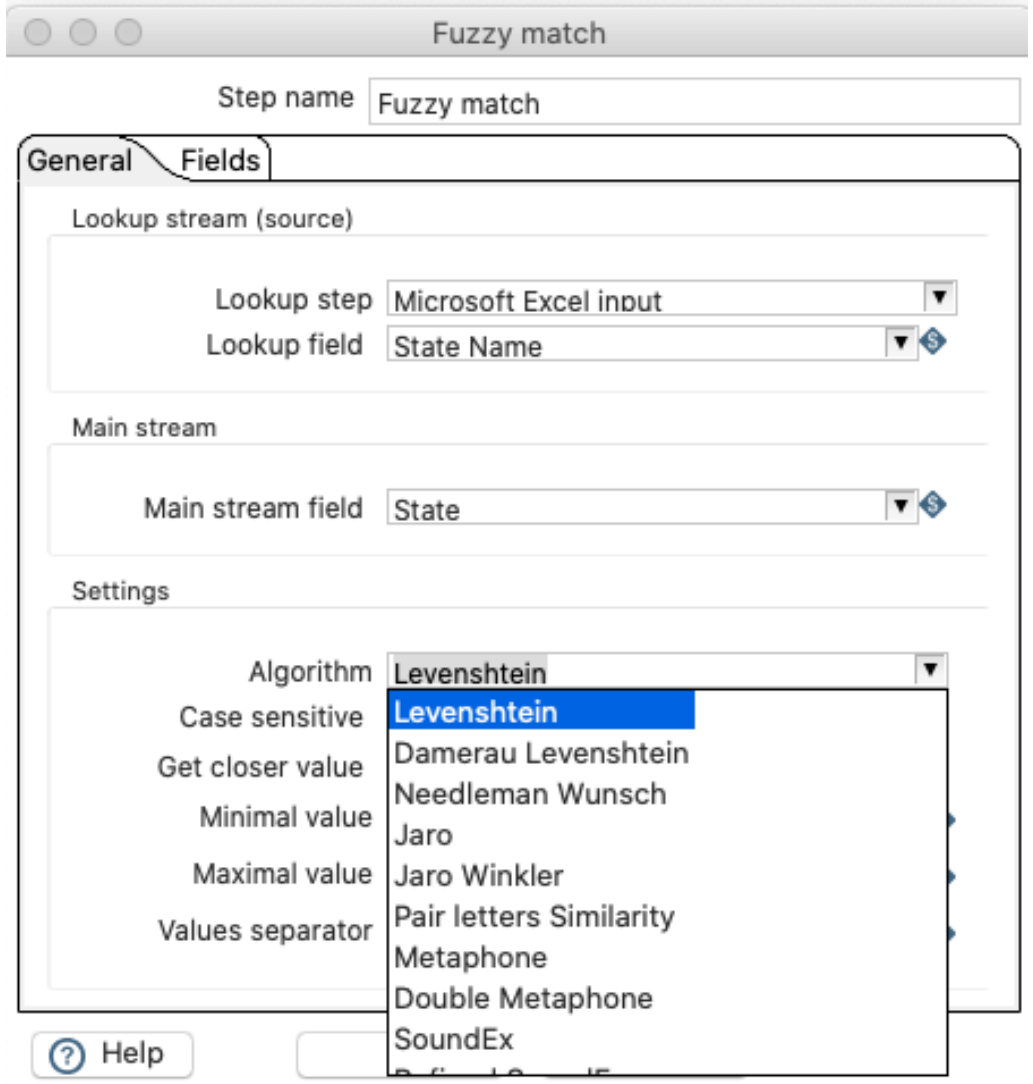
Rows of step: Fuzzy match (793 rows)

#	Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	match	measure value	State Name
49	AR-10540	Andy Reiter	Consumer	48	United States	New York City	New York	10024	East	New York	0	New York
50	AR-10570	Anemone Ratner	Consumer	36	United States	Columbus	Georgia	31907	South	Georgia	0	Georgia
51	AR-10825	Anthony Rawles	Corporate	39	United States	Vancouver	Washington	98661	West	Washington	0	Washington
52	AS-10045	Aaron Smayling	Corporate	42	United States	Jacksonville	North Carolina	28540	South	North Carolina	0	North Carolina
53	AS-10090	Adam Shillingsburg	Consumer	46	United States	Charlottesville	Virginia	22901	South	Virginia	0	Virginia
54	AS-10135	Adrian Shami	Home Office	65	United States	New York City	New York	10035	East	New York	0	New York
55	AS-10225	Alan Schoenberger	Corporate	31	United States	New York City	New York	10024	East	New York	0	New York
56	AS-10240	Alan Shonely	Consumer	68	United States	Tampa	Florida	33614	South	Florida	0	Florida
57	AS-10285	Alejandro Savely	Corporate	53	United States	San Francisco	California	94109	West	California	0	California
58	AS-10630	Ann Steele	Home Office	54	United States	Pasadena	Rwxas	77506	Central	Texas	2	Texas
59	AT-10435	Alyssa Tate	Home Office	19	United States	Los Angeles	California	90004	West	California	0	California
60	AT-10735	Annie Thurman	Consumer	30	United States	Seattle	Washington	98115	West	Washington	0	Washington
61	AW-10840	Anthony Witt	Consumer	37	United States	Allen	Texas	75002	Central	Texas	0	Texas
62	AW-10930	Arthur Wiediger	Home Office	58	United States	Aurora	Illinois	60505	Central	Illinois	0	Illinois
63	AY-10555	Andy Yotov	Corporate	61	United States	Oceanside	California	92054	West	California	0	California
64	AZ-10750	Annie Zypern	Consumer	59	United States	New York City	New York	10009	East	New York	0	New York
65	BB-10990	Barry Blumstein	Corporate	41	United States	Inglewood	California	90301	West	California	0	California
66	BB-11545	Brenda Bowman	Corporate	60	United States	Houston	Texas	77070	Central	Texas	0	Texas
67	BC-11125	Becky Castell	Home Office	35	United States	Peoria	Arizona	85345	West	Arizona	0	Arizona
68	BD-11320	Bill Donatelli	Consumer	68	United States	San Francisco	California	94110	West	California	0	California

Data Cleansing: Fuzzy match example



Data Cleansing: Fuzzy match Algorithms



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’
  - Cailfornia-> 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 C A L I **F O R** N **I A** and **F** L **O R I** D **A**?*

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

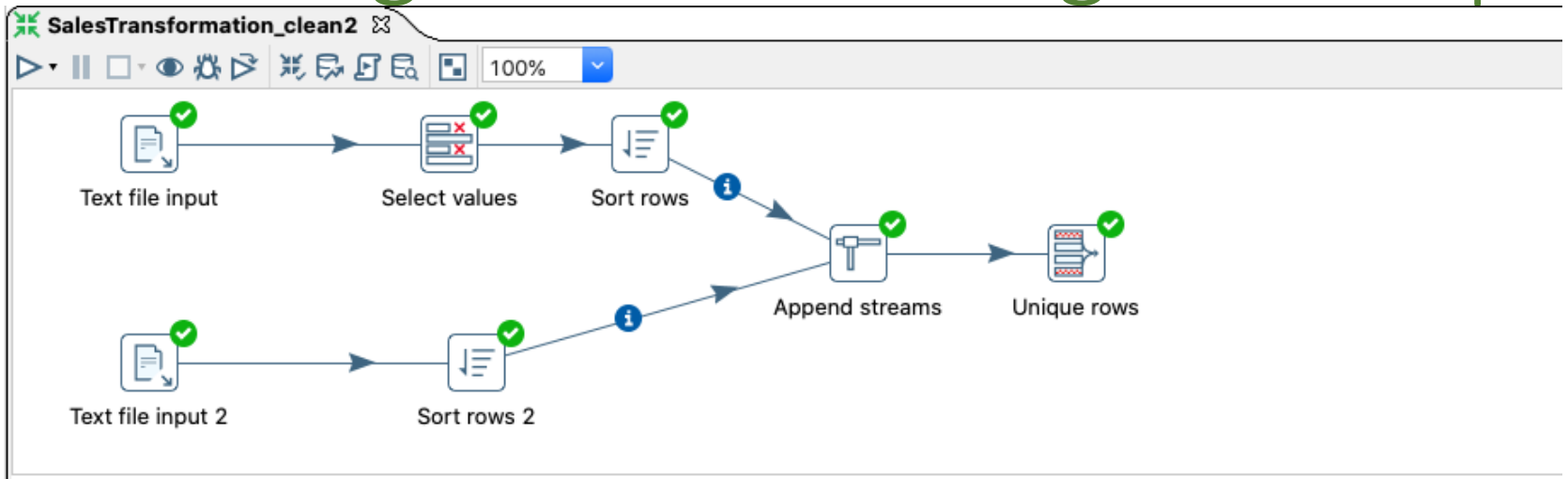
- ‘Phonetic’ Algorithms, try to match the sound of words
- Commonly used for deduplication
- Only applicable on English language
- Useful when the errors are due to persons not knowing exact spellings and inputting strings based on their sounds

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



Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

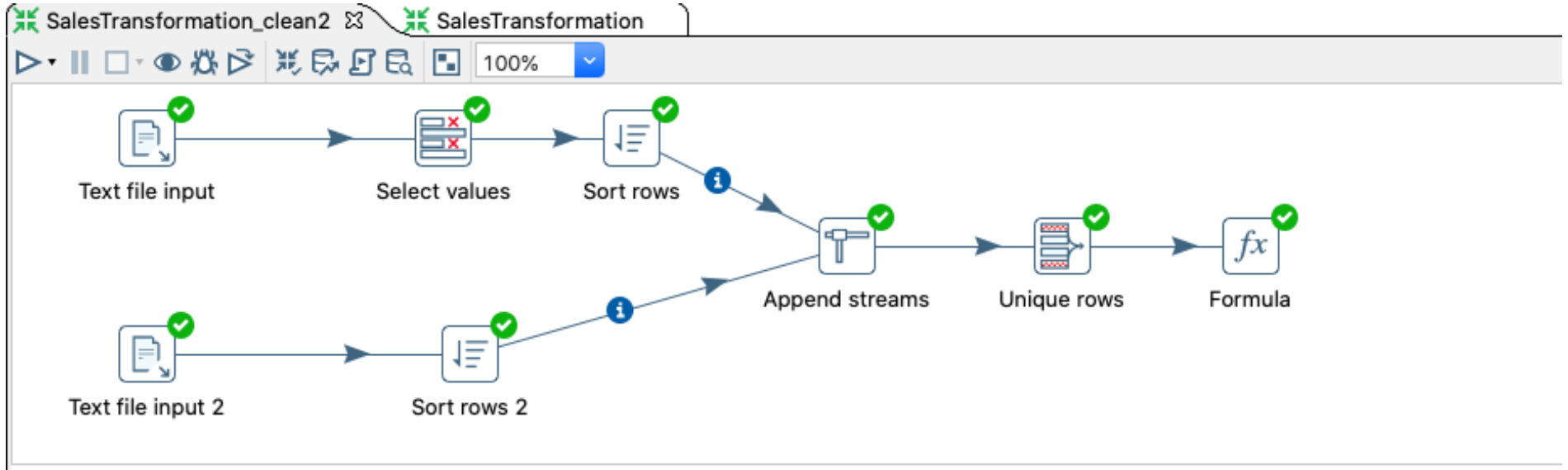
First rows | Last rows | Off

#	Order_Line	Order_ID	Order_Date	Ship_Date	Ship_Mode	Customer_ID	Product_ID	Sales	Quantity	Discount	Profit
1	7973	CA-2017-166...	2020-01-...	2020-01...	Standard Class	MM-17260	OFF-BI-10004094	26.6	3	0	13
2	7974	CA-2017-166...	2020-01-...	2020-01...	Standard Class	MM-17260	FUR-TA-10004607	310.4	3	0.3	-48.8
3	7975	CA-2017-164...	2020-01-...	2020-01...	Second Class	MM-18055	OFF-AR-10001...	6.6	2	0	1.9
4	7976	CA-2017-164...	2020-01-...	2020-01...	Second Class	MM-18055	OFF-LA-10000634	7.8	3	0	3.6
5	7977	CA-2017-164...	2020-01-...	2020-01...	Second Class	MM-18055	TEC-AC-10004...	41.9	2	0	8.8
6	7978	CA-2017-164...	2020-01-...	2020-01...	Second Class	MM-18055	FUR-CH-10002...	664.1	6	0.1	88.5
7	7979	CA-2017-164...	2020-01-...	2020-01...	Second Class	MM-18055	OFF-PA-100045...	9	2	0	4.4
8	7980	US-2017-128...	2020-01-...	2020-01...	First Class	RS-19420	OFF-AP-100021...	179.9	3	0	50.4
9	7981	US-2017-128...	2020-01-...	2020-01...	First Class	RS-19420	FUR-TA-10004575	872.9	3	0	157.1
10	7982	US-2017-128...	2020-01-...	2020-01...	First Class	RS-19420	OFF-PA-100031...	13	2	0	6.2
11	7983	US-2017-156...	2020-01-...	2020-01...	Second Class	SF-20065	FUR-CH-10002...	71.4	2	0.3	-1
12	7984	CA-2017-109...	2020-01-...	2020-01...	Standard Class	VM-21685	OFF-AR-10003...	2.9	2	0.2	0.9
13	7985	US-2017-152...	2020-01-...	2020-01...	Standard Class	NF-18385	FUR-CH-10004...	242.3	3	0.2	15.2
14	7986	CA-2017-139...	2020-01-...	2020-01...	Standard Class	SW-20455	FUR-FU-10002...	7.9	2	0.2	2.2

Discount
→ in percentage

Data Cleansing: Format changes example

Sales Transformation



date format mm/dd/yyyy → dd-mm-yyyy

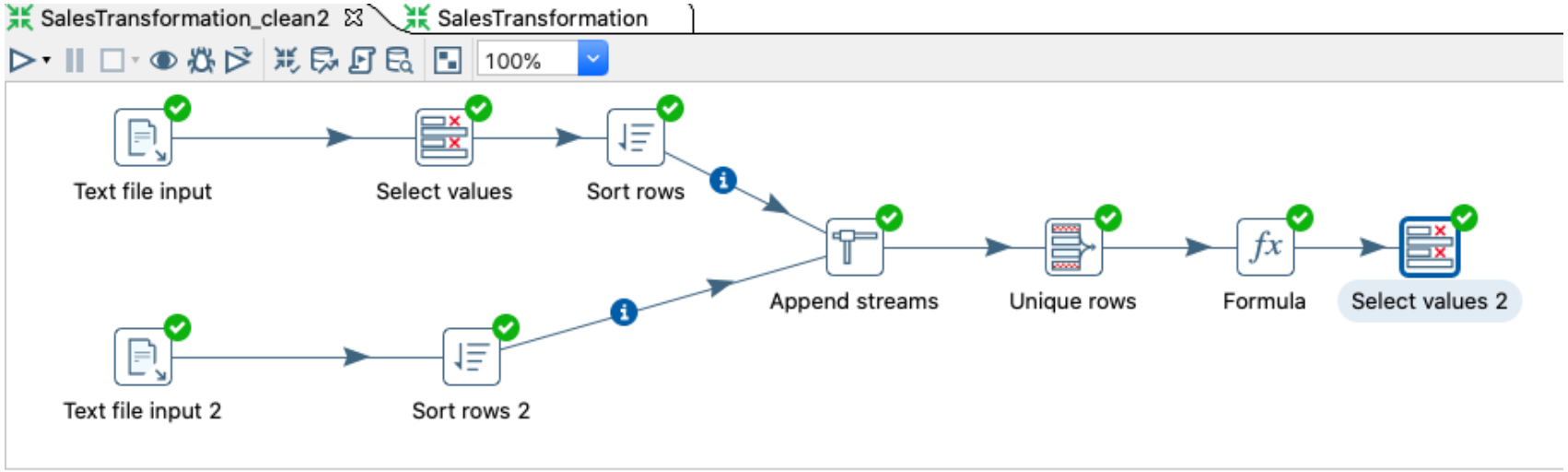
Execution Results

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

Data Cleansing: Format changes example



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 **format**	Select values
Values in multiple columns are to be **combined** 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 **duplicates**	Unique rows
Remove/ change special characters or part of **strings**	Replace in string