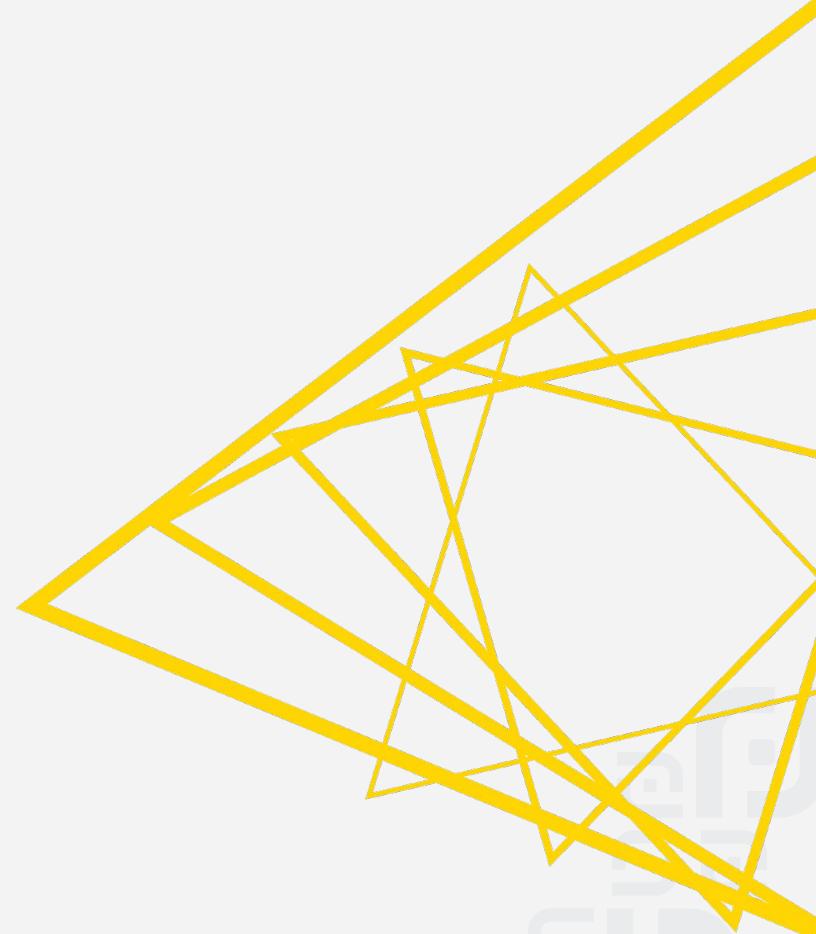


Data Manipulation



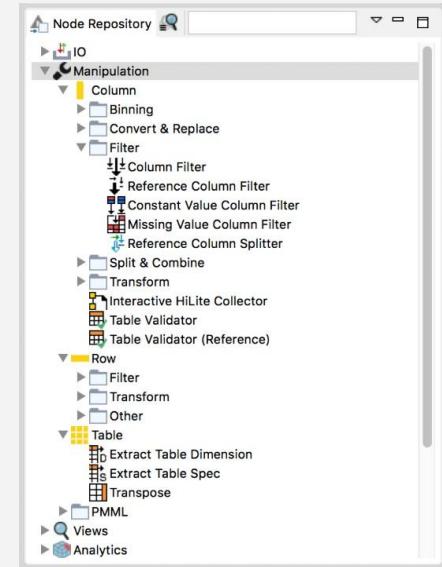
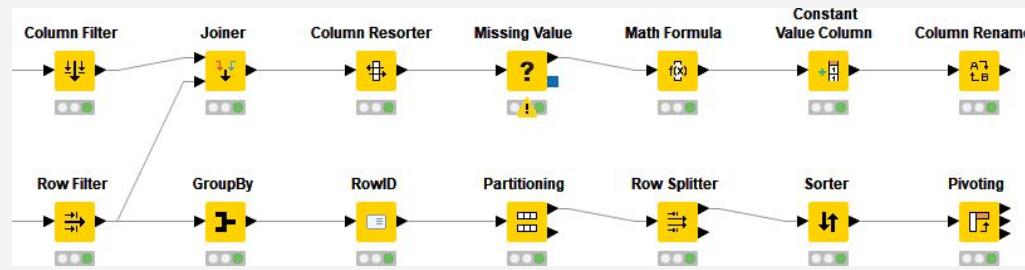
Data Manipulation

Clean, Join, Aggregate



Data Manipulation Nodes

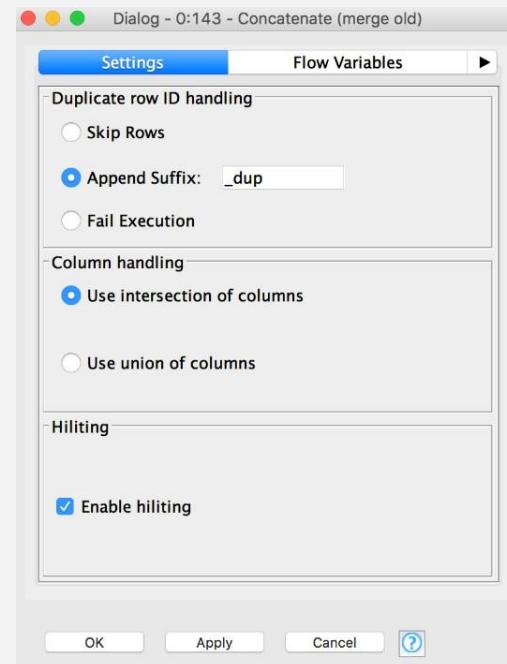
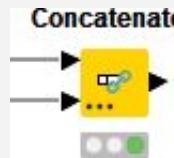
- Yellow color with a variety of input and output ports
- Apply a transformation to input data
- Many, many nodes!



New Node: Concatenate

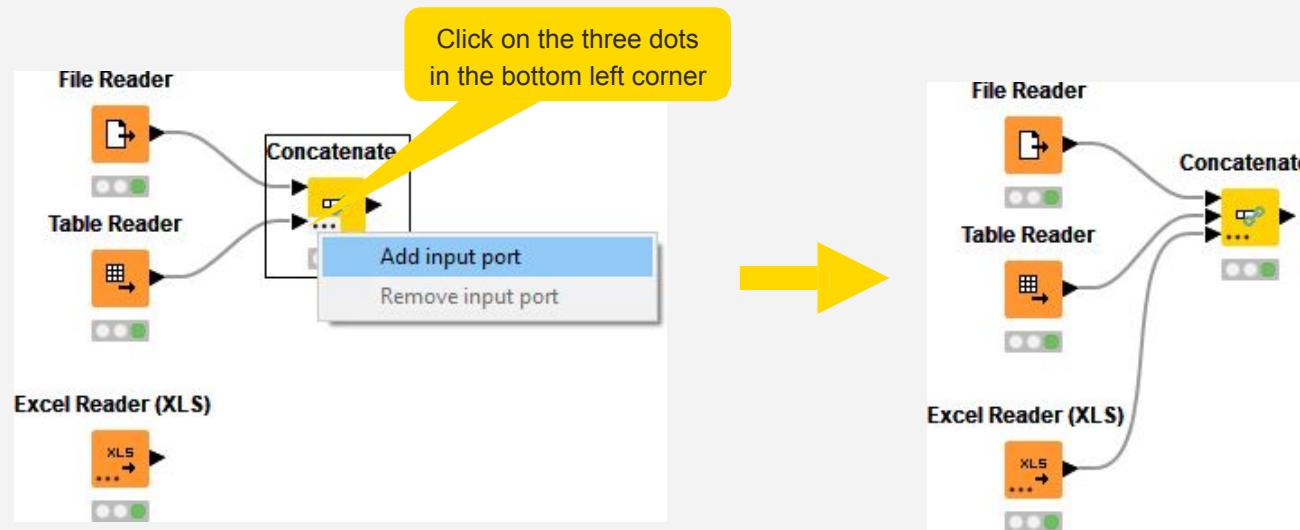
Combine rows from 2 or more tables with shared columns

- Handles duplicate row keys gracefully
- Take the union or intersection of columns



Dynamic Ports

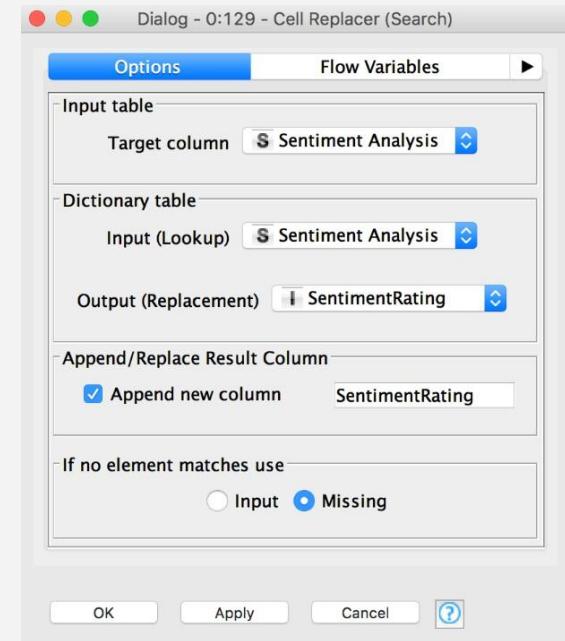
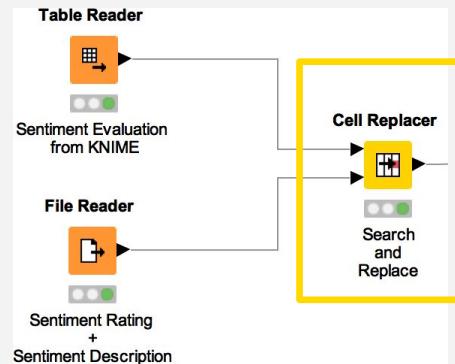
Add and remove node ports based on your needs, e.g. in order to concatenate three or more tables



New Node: Cell Replacer

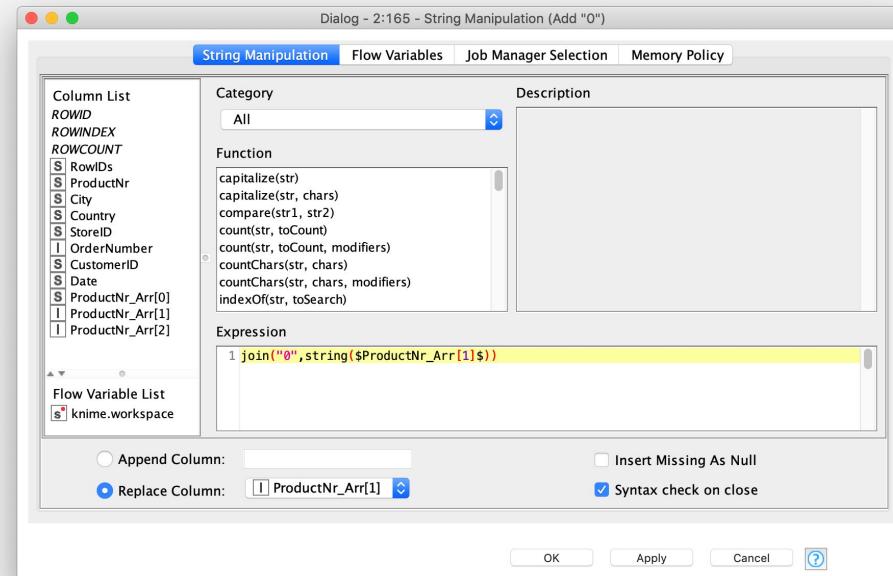
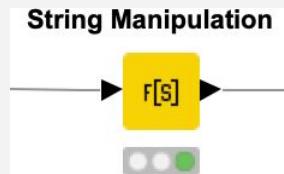
Replaces the content of a column based on a lookup

- Top port references the table to be searched
- Bottom port holds the lookup table (search keys and replacement values)



New Node: String Manipulation

- Create and edit values in a String Column
 - Cleans up capitalization
 - Joins string values
 - Pads strings, e.g. padLeft
 - Replaces string values



Data Manipulation Exercise, Activity I

Start with exercise: *Data Manipulation, Activity I*

- Concatenate web activity data from the old and new systems
- Replace the written sentiment values with the numeric sentiment scores
- Make sure that all product names in the product data spreadsheet are written in lower case letters

Joining Columns of Data

Left Table

CustomerKey	OrderDate	OrderID
22	2019-09-23	#23444
24	2019-09-30	#23457
15	2019-10-07	#28985
10	2091-10-13	#29999

Right Table

CustomerKey	DoB	City	Gender
17	1974-02-23	Berlin	F
65	2001-05-25	Stuttgart	F
35	1988-08-05	Cologne	M
15	1983-07-20	Hamburg	M
10	1993-01-13	Berlin	M

Join by CustomerKey

Inner Join

CustomerKey	OrderDate	OrderID	DoB	City	Gender
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M

Left Outer Join

Right Outer Join

CustomerKey	OrderDate	OrderID	DoB	City	Gender
22	2019-09-23	#23444	?	?	?
24	2019-09-30	#23457	?	?	?
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M

CustomerKey	OrderDate	OrderID	DoB	City	Gender
17	?	?	1974-02-23	Berlin	F
65	?	?	2001-05-25	Stuttgart	F
35	?	?	1988-08-05	Cologne	M
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M

Joining Columns of Data

Left Table

CustomerKey	OrderDate	OrderID
22	2019-09-23	#23444
24	2019-09-30	#23457
15	2019-10-07	#28985
10	2091-10-13	#29999

Join by CustomerKey

Full Outer Join

Right Table

CustomerKey	DoB	City	Gender
17	1974-02-23	Berlin	F
65	2001-05-25	Stuttgart	F
35	1988-08-05	Cologne	M
15	1983-07-20	Hamburg	M
10	1993-01-13	Berlin	M

Missing values in the left table

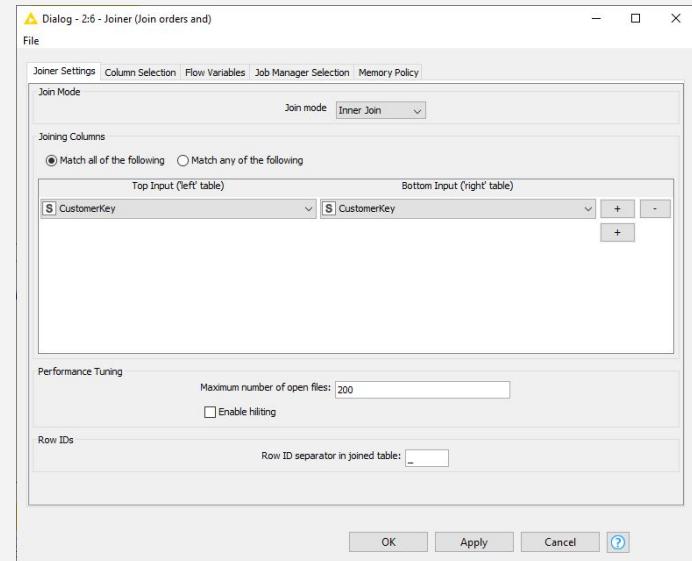
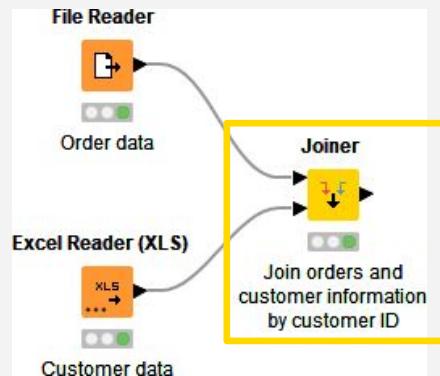
CustomerKey	OrderDate	OrderID	DoB	City	Gender
17	?	?	1974-02-23	Berlin	F
65	?	?	2001-05-25	Stuttgart	F
35	?	?	1988-08-05	Cologne	M
15	2019-10-07	#28985	1983-07-20	Hamburg	M
10	2091-10-13	#29999	1993-01-13	Berlin	M
22	2019-09-23	#23444	?	?	?
24	2019-09-30	#23457	?	?	?

Missing values in the right table

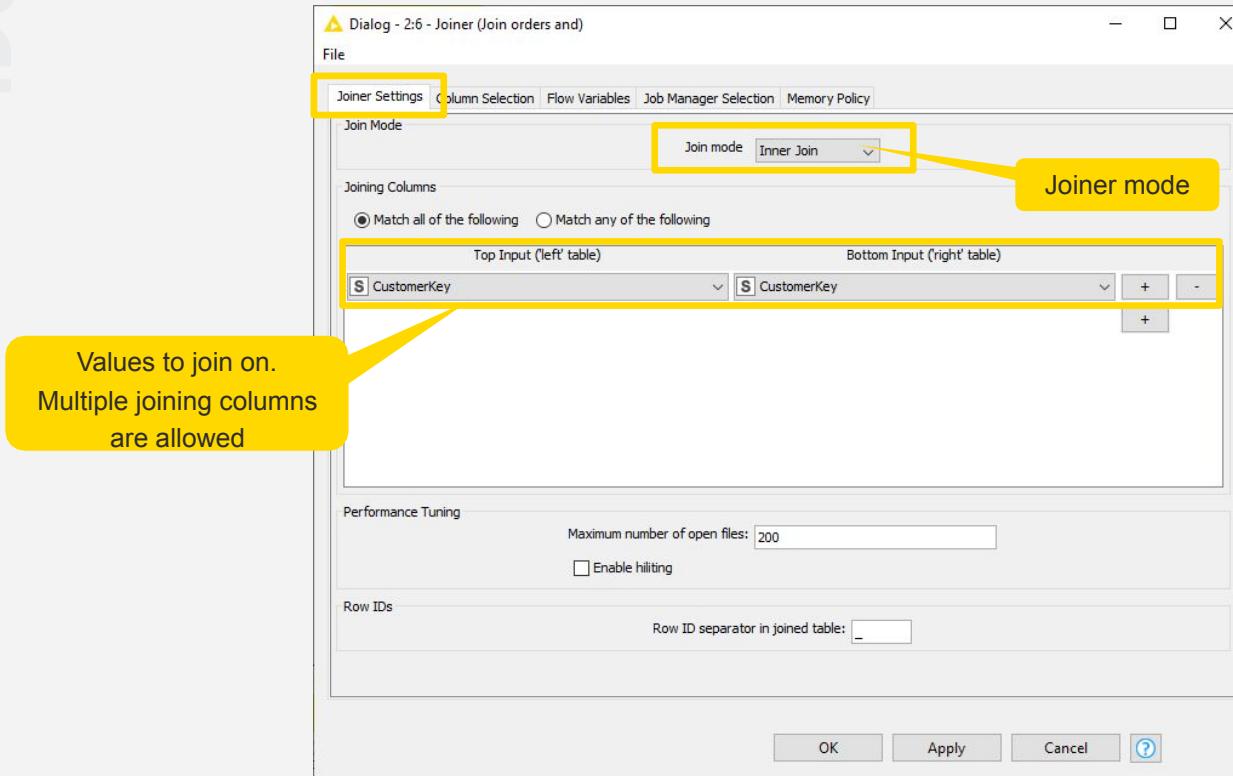
New Node: Joiner

Combines columns from 2 different tables

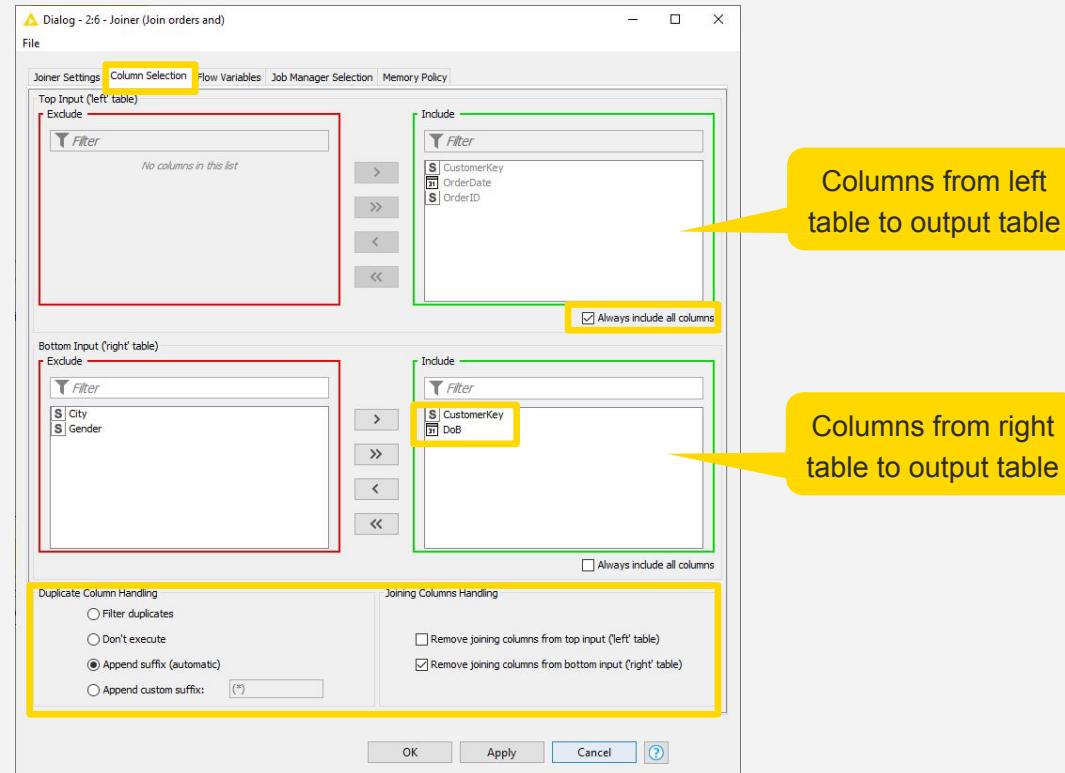
- Top port contains “Left” data table
- Bottom port contains “Right” data table



Joiner Configuration – Linking Rows



Joiner Configuration – Column Selection



Data Aggregation

Product ID	Category	# Ordered Items
P 1	Clothing	2
P 2	Home	3
P 3	Clothing	1
P 4	Clothing	5
P 5	Electronics	7
P 6	Electronics	5



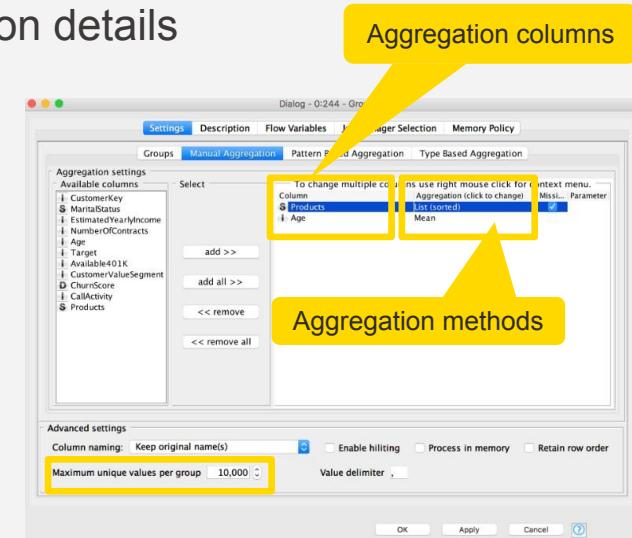
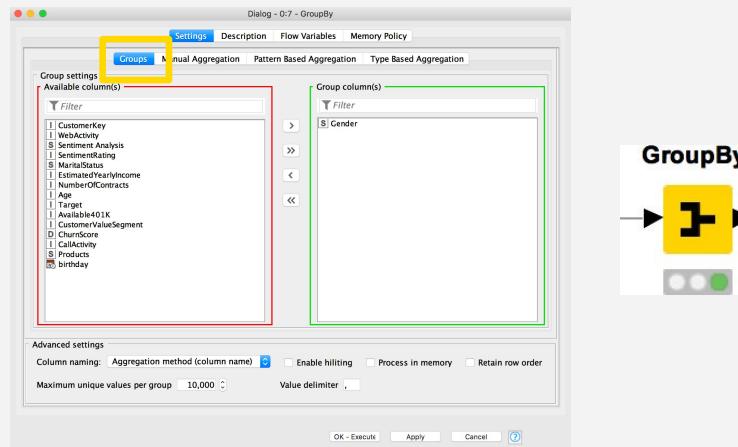
Group	Sum(# Ordered Items)
Clothing	8
Home	3
Electronics	12

Aggregated on Category (group) by Sum (aggregation method)

New Node: GroupBy

Aggregate rows to summarize data

- First tab provides grouping options
 - Second tab provides control over aggregation details

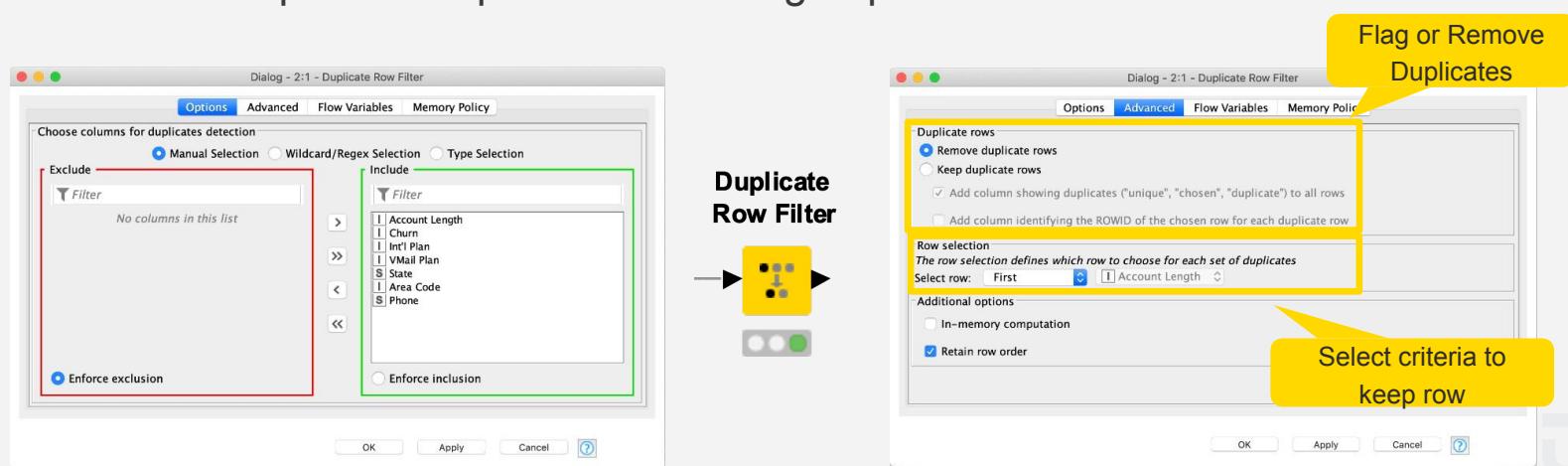


YouTube KNIME TV video:
<https://youtu.be/bDwF-TOMtWw>

New Node: Duplicate Row Filter

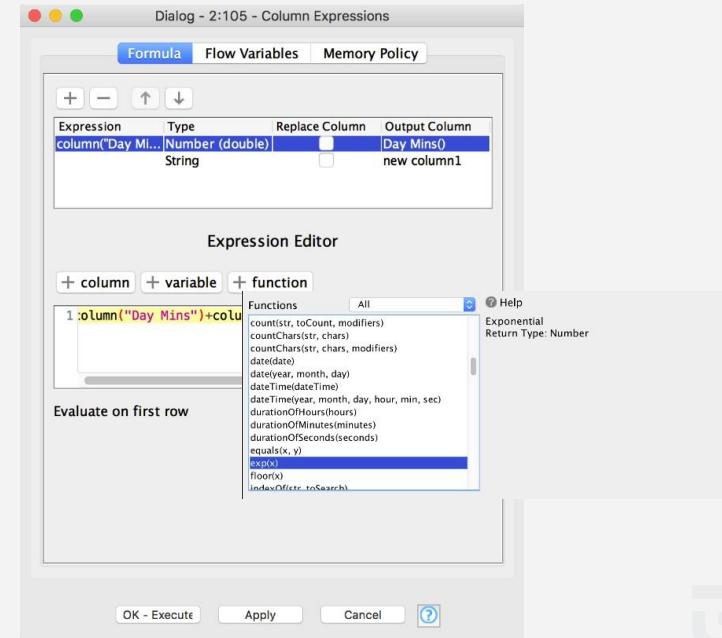
Detect duplicate rows and apply a selected treatment

- First tab provides the option to select columns
- Second tab provides options for treating duplicated values



New Node: Column Expression

- Append or modify an arbitrary number of columns using expressions
- Many different functions are available
- No restriction on number of lines per expression allow to write complex expressions
- Part of the KNIME Labs extension



Sorter

- Sorts a table!
- Choice of ascending or descending
- Sort by multiple columns

The image shows the KNIME interface with a 'Sorter' node in the workspace. To the right is the 'Dialog - 0:5 - Sorter' window, which contains the following settings:

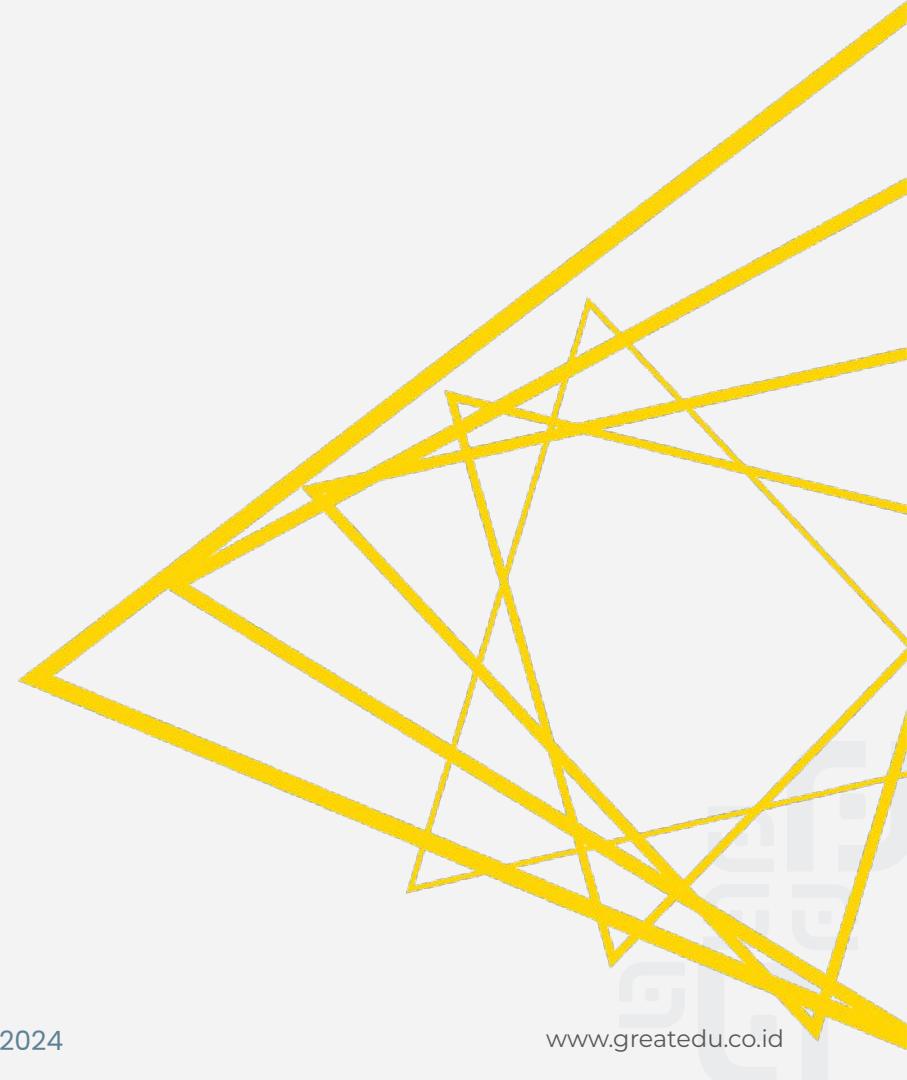
- Sort by:** Count*(Age)
- Order:** Descending (selected radio button)
- Add columns:** 1 column selected, 'new columns' button available.
- Checkboxes:** 'Sort in memory' and 'Move Missing Cells to end of sorted list' are unchecked.

Below the dialog are three buttons: OK, Apply, Cancel, and Help.

To the right of the dialog is the 'Sorted Table - 0:5 - Sorter' window, displaying the sorted data:

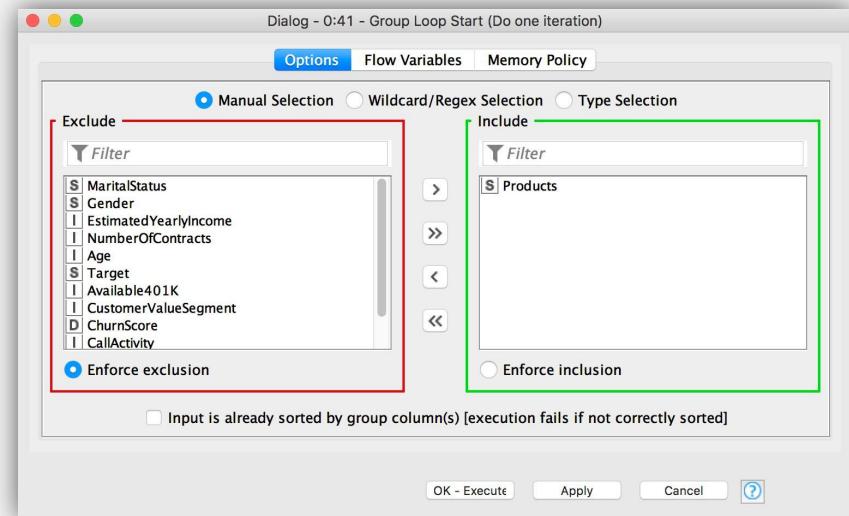
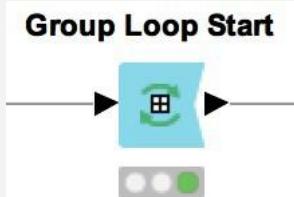
Row ID	Products	Count*(Age)
Row4	Private Investment	5296
Row3	P+B Investment	5018
Row2	Gold Investment	3258
Row1	Fund Manager+	3143
Row0	CO Investment	1549

Flow Variables



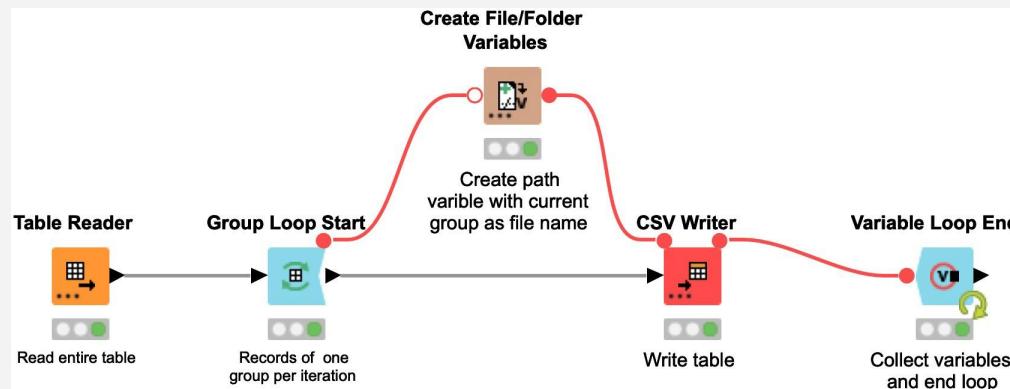
Group Loop Start

- Similar to GroupBy except without aggregation tab.
- Each iteration of the loop passes the next group of rows.
- You can implement an aggregation task. It can be anything from a complex calculation to updating a database.



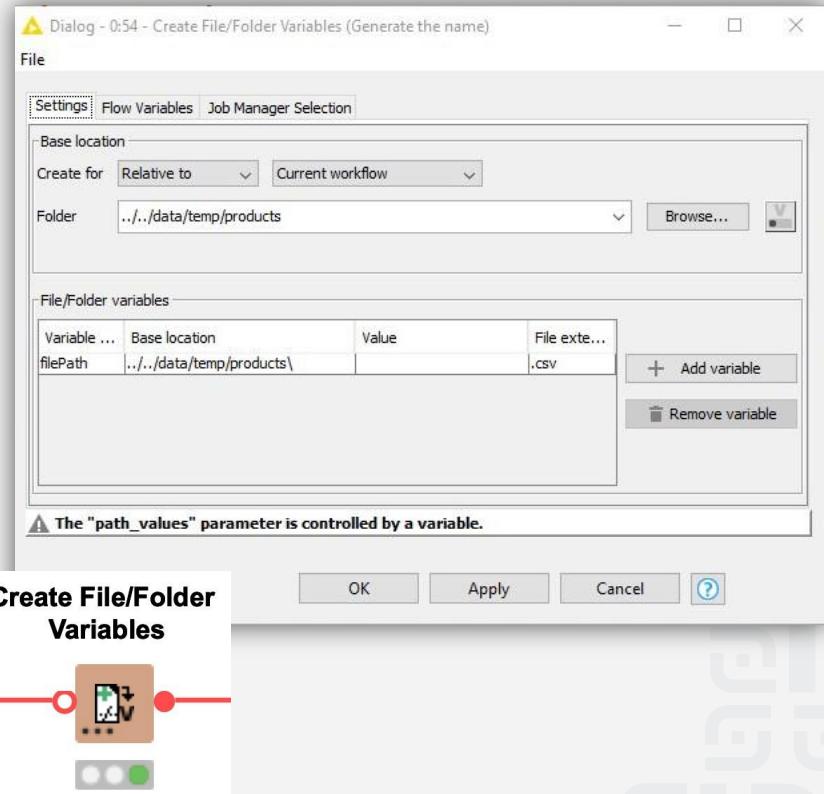
Example: Writing Aggregated Files

- Group Loop Start → Variable Loop End
- Group data by specific column values
- Iterate over all groups of data
- Create an appropriate file name
- Write grouped data to tables with new file name

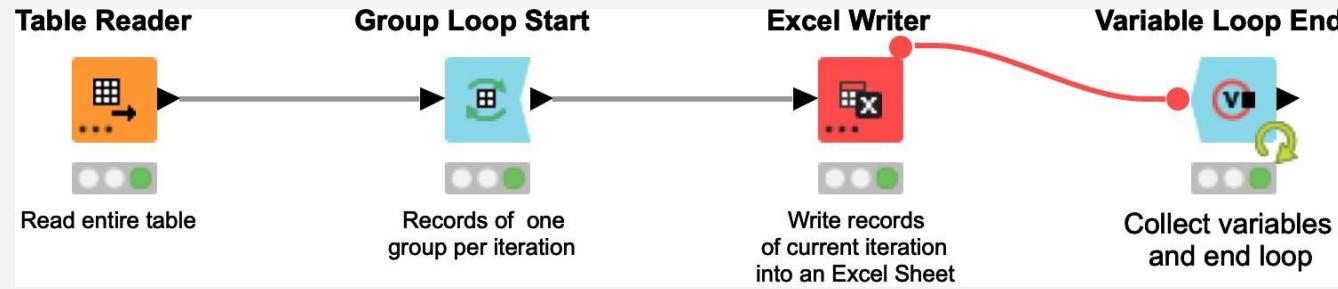


Create File/Folder Variables

- Creates one or multiple path flow variable(s) pointing to files / folders
- Inputs:
 - Base location
 - Flow variable name(s)
 - Value (file name or path relative to base location)
 - File extension (optional)
- Output variables can be used to control the output location in writer nodes.



Example: Writing Multiple Excel Sheets



Example: Reading Many Excel Sheets

- List all sheet names of an Excel file
- Convert sheet name into a flow variable (1 sheet name per iteration)
- In each iteration, read the spreadsheet with the current sheet name
- Close the loop and collect the results

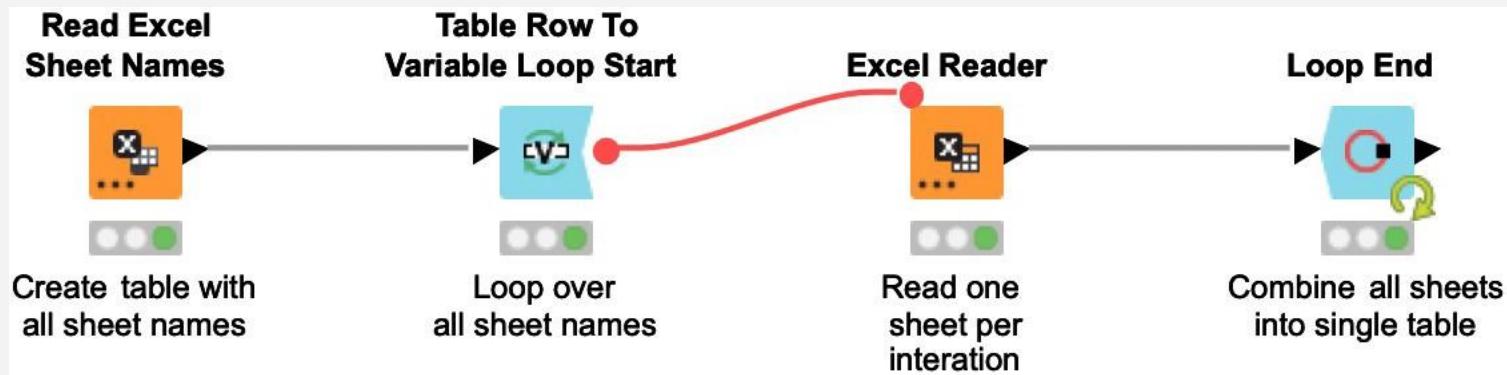
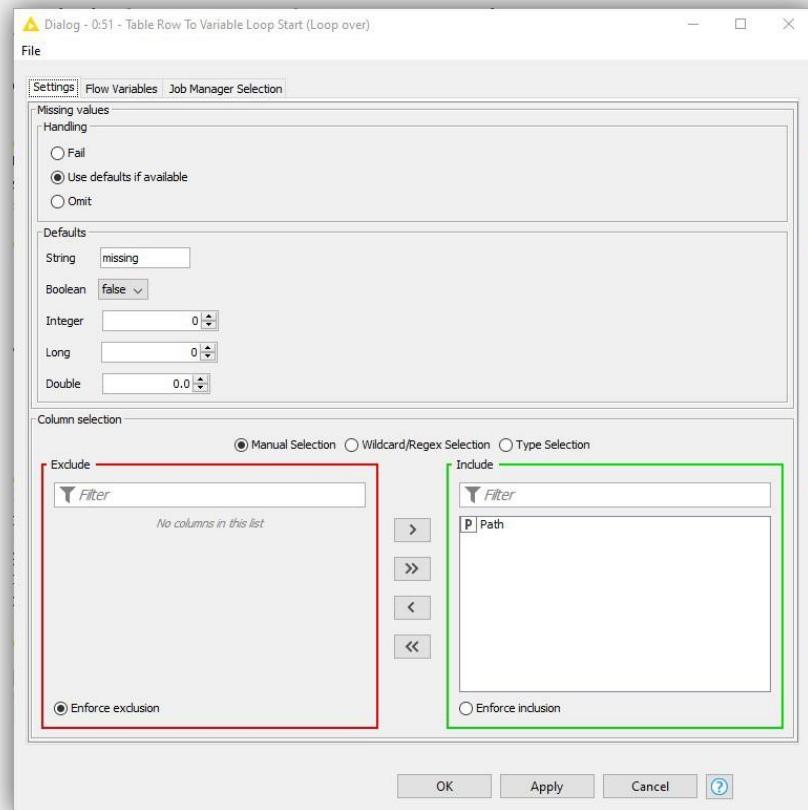
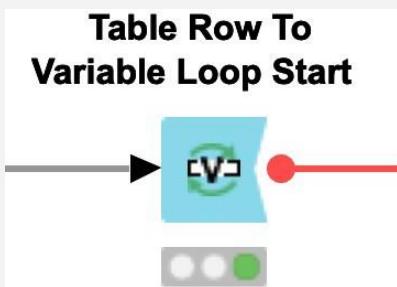


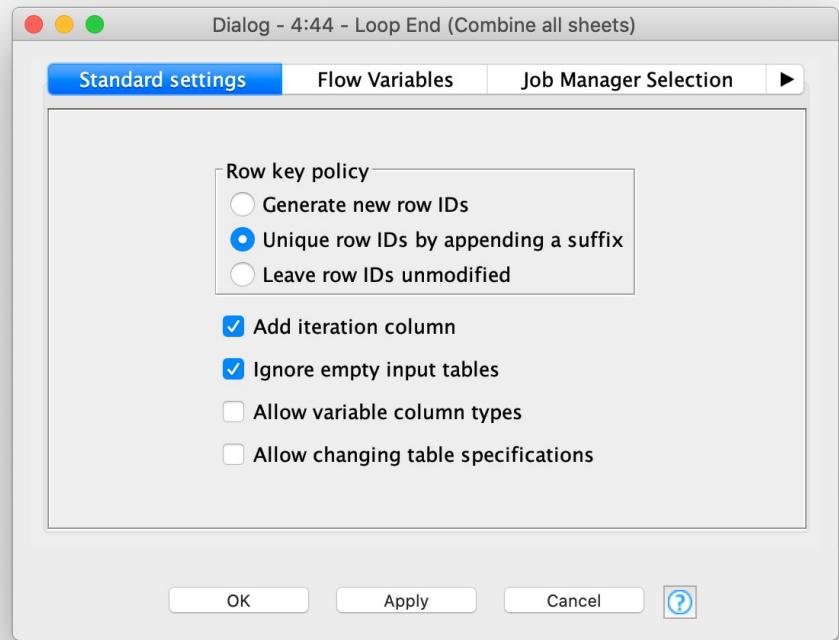
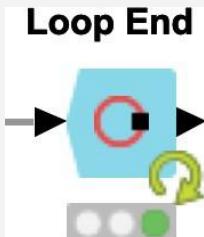
Table Row to Variable Loop Start

- Similar to the Table Row to Variable node
- Each iteration of the loop converts the next row of the input table into Flow Variables
- Injects variables into other nodes to re-execute subflows with a progression of settings



Loop End

- Can be used to end of a loop
- Collects the results of the different iterations by row-wise concatenation of the incoming tables
- Provides options to:
 - Add a column with the iteration number
 - Allow variable column types
 - Allow changing table specifications

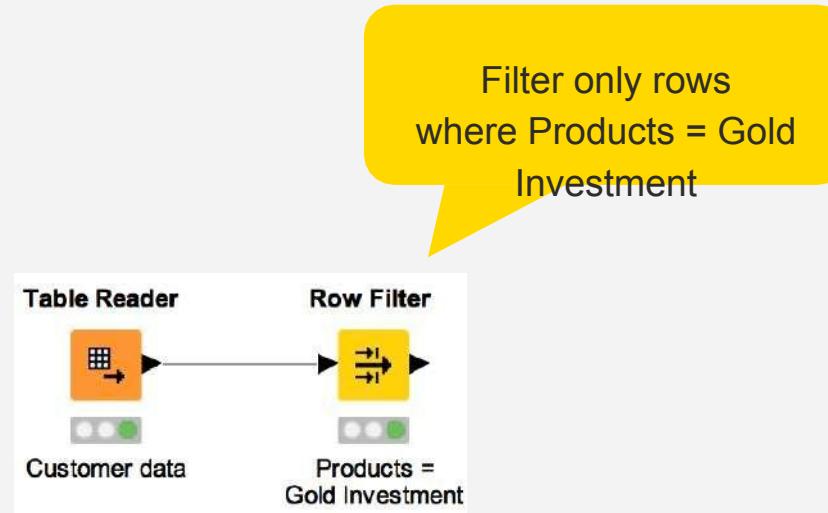


Goal of this Session

- What is a Flow Variable?
- Create a Flow Variable
- Use a Flow Variable as a parameter in the node settings
- Use a Configuration node to parameterize a Component
- Use a Widget node to enable interaction on a WebPortal page

Flow Variables: Usage Example

- Each month you need to produce a sales report for the most popular product



Flow Variables: Usage Example

- Each month I need to launch the Analytics Platform, aggregate the data to identify the most popular product, and update the Row Filter accordingly
- Or do I? Perhaps Flow Variables can help...

Automatically Filter by Most Popular Product

Count products, and put most popular at the top of the list

Create a Flow Variable containing the name and count of the most popular product

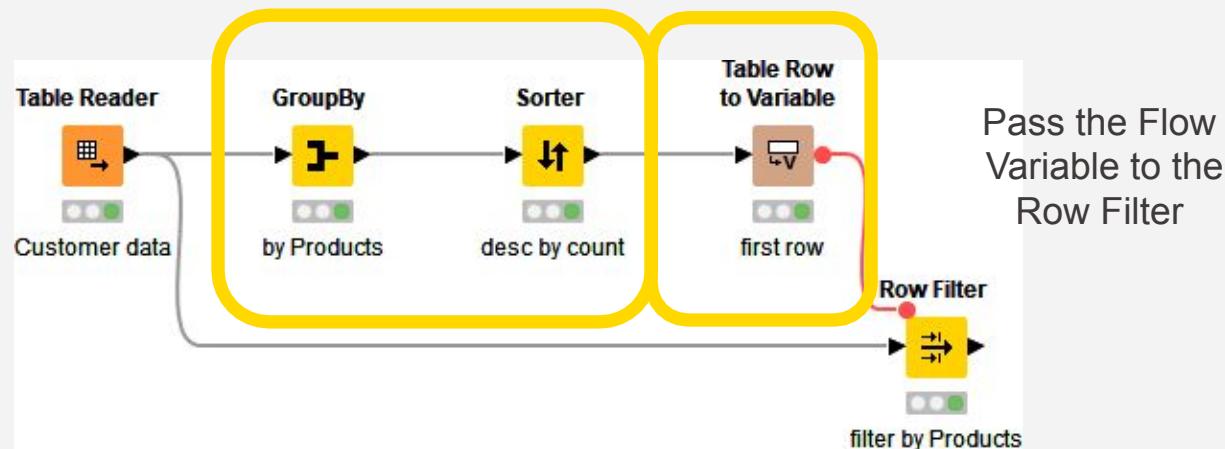


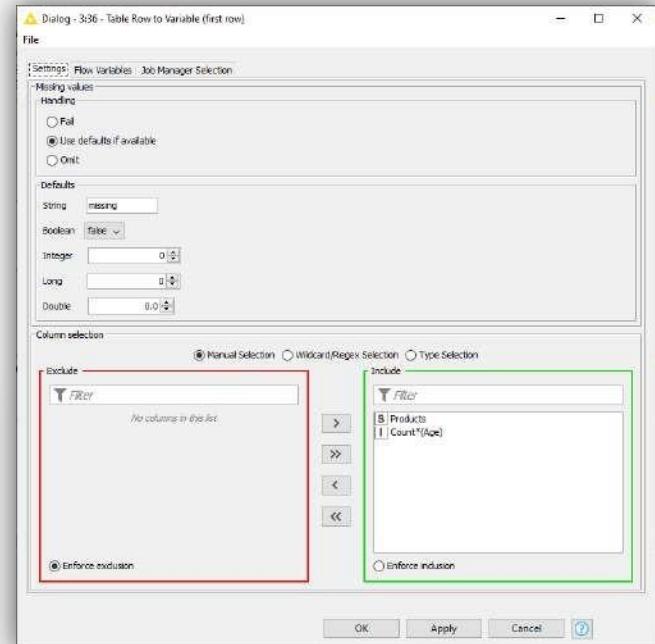
Table Row to Variable

- Takes a table as input and converts the first row to Flow Variables
 - Column names -> Flow Variable names
 - Column values -> Flow Variable values
- Only the first row is transformed, additional rows are discarded

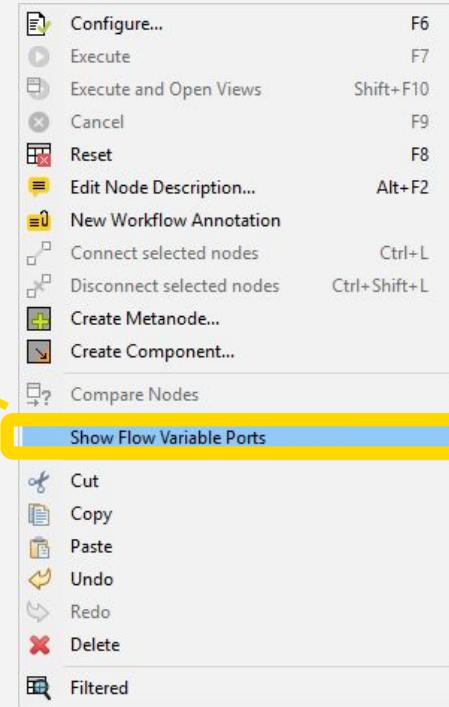
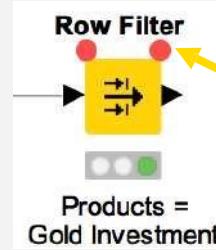
Sorted Table - 4:5 - Sorter (desc by count)		
File Hilite Navigation View		
Table "default" - Rows: 5 Spec - Columns: 2		
Row ID	Products	Count...
Row4	Private Investment	5296
Row3	P+B Investment	5018
Row2	Gold Investment	3258
Row1	Fund Manager+	3143
Row0	CO Investment	1549

Table Row to Variable

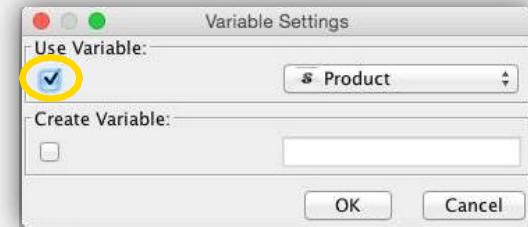
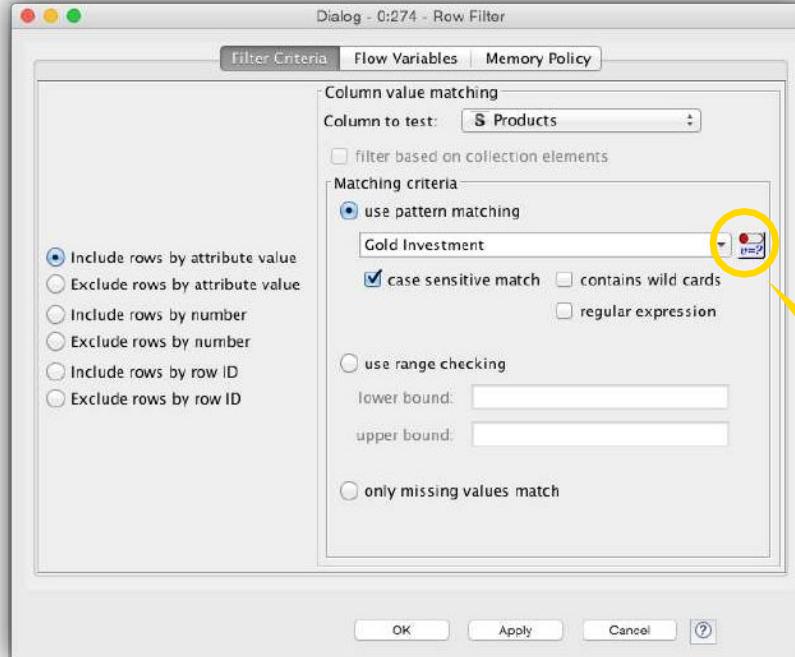
In...	Owner...	Name	Value
0 4:6	# Products	Private Investment	
0 4:6	# Count(Age)	5296	
0 4:6	# RowID	Row4	
0	knime.works...	/Users/kathrinmeicher/knime-workspa...	



Flow Variable Ports

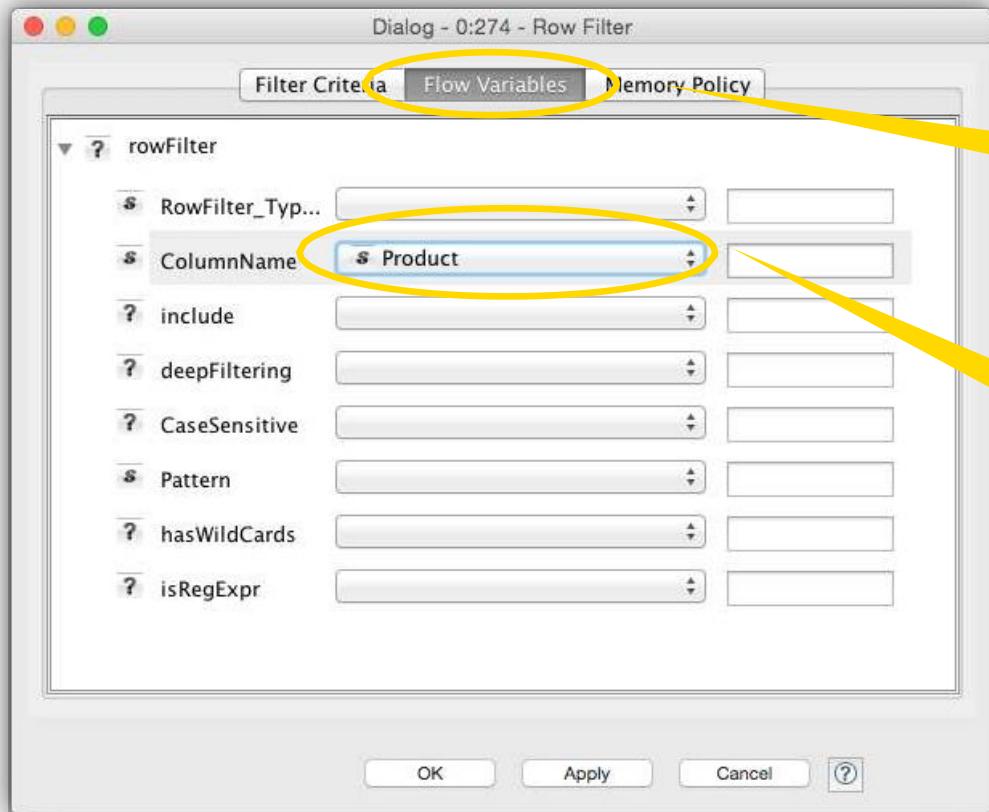


Apply a Flow Variable (Button)



The Flow Variable button

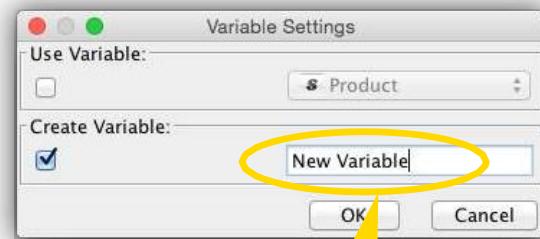
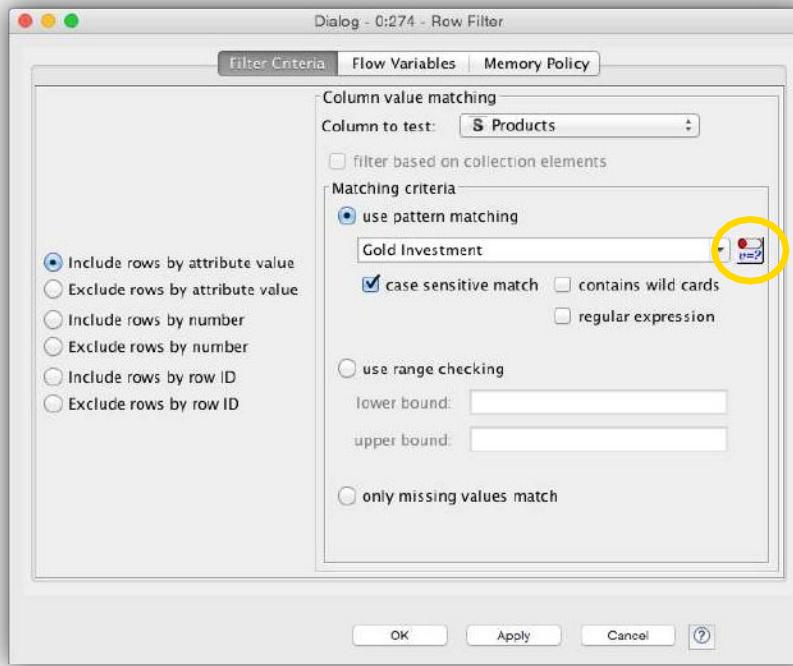
Apply a Flow Variable (Advanced)



The Flow
Variables
tab

List of available Flow
Variables

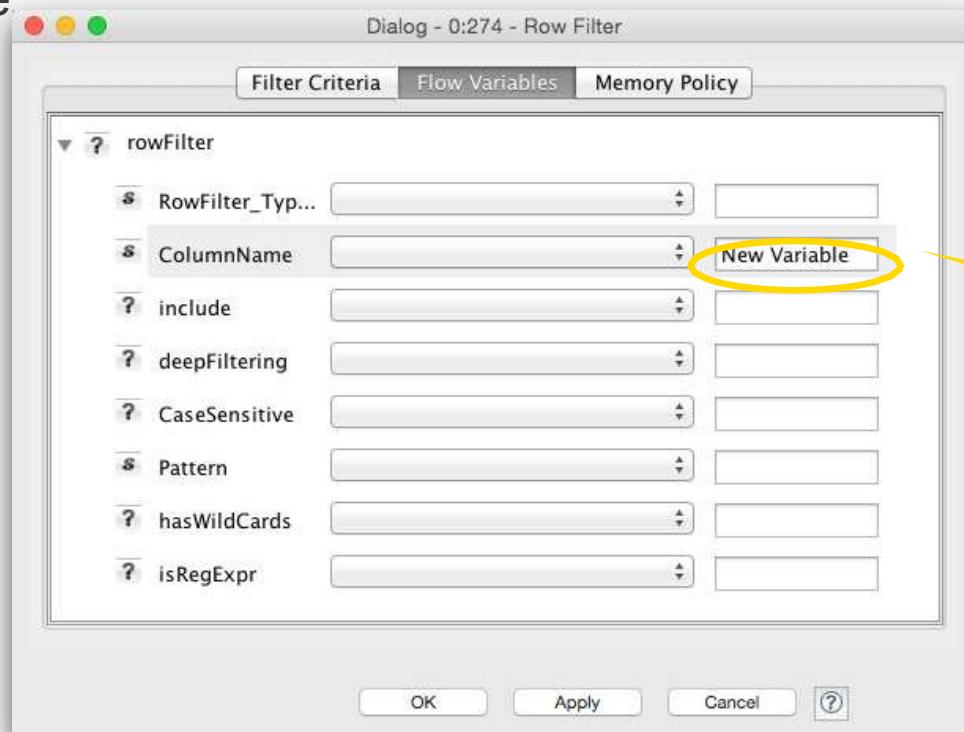
Create a Flow Variable (Button)



Name of the new
Flow Variable

Create a Flow Variable (Advanced)

Converting a setting value into a Flow Variable

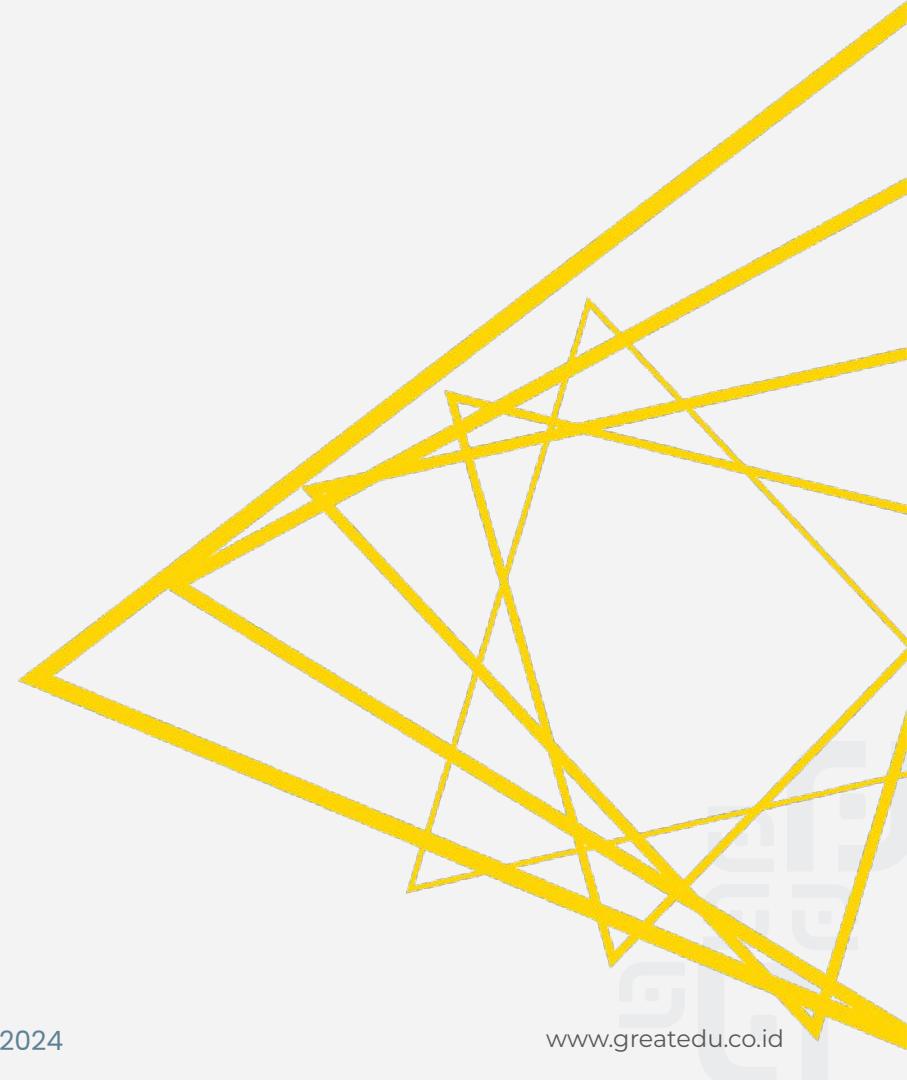


Name of the new
Flow Variable

Key Features: Flow Variables

- Flow Variables are workflow parameters used to overwrite existing node settings
- A Flow Variable is carried along workflow branches (parallel branches don't share local Flow Variables)
- Flow Variables can be of type String, Integer, Double, Boolean, Long and Array, Path
- Flow Variables can be created
 1. in the “Flow Variables” tab of any node
 2. using the “Table Row to Variable” node
 3. using Configuration and Widget nodes

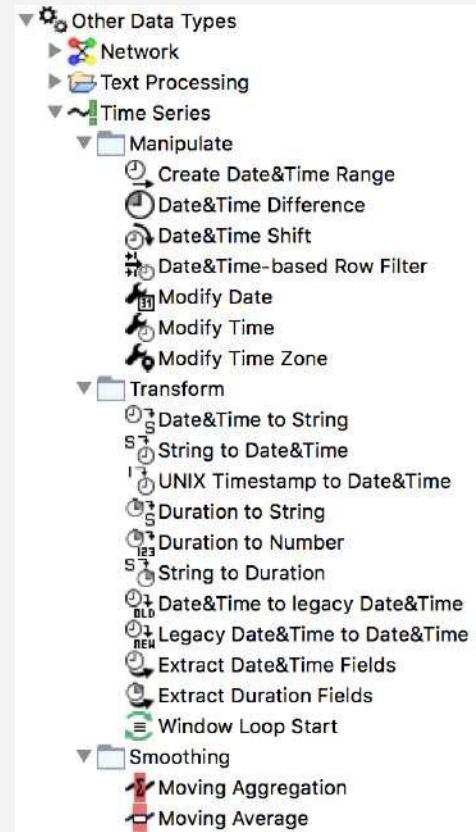
Date/Time Data



Date & Time Overview

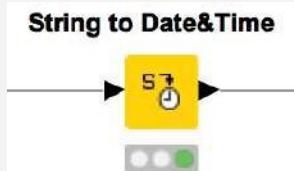
- Dedicated data type for date and time data
- Supported in Date&Time nodes
 - (and others: GroupBy, Pivot, Line Plot)
- Complete re-write in KNIME 3.4

Row ID	timestamp	Intensity
Row0	2007-01-01T00:01	10.4
Row1	2007-01-01T00:02	10.4
Row2	2007-01-01T00:03	10.4
Row3	2007-01-01T00:04	10.4
Row4	2007-01-01T00:05	10.4
Row5	2007-01-01T00:06	10.4
Row6	2007-01-01T00:07	10.2
Row7	2007-01-01T00:08	10.2
Row8	2007-01-01T00:09	10.2
Row9	2007-01-01T00:10	10.2
Row10	2007-01-01T00:11	10.2
Row11	2007-01-01T00:12	10.2
Row12	2007-01-01T00:13	10.2
Row13	2007-01-01T00:14	10.2
Row14	2007-01-01T00:15	10.2

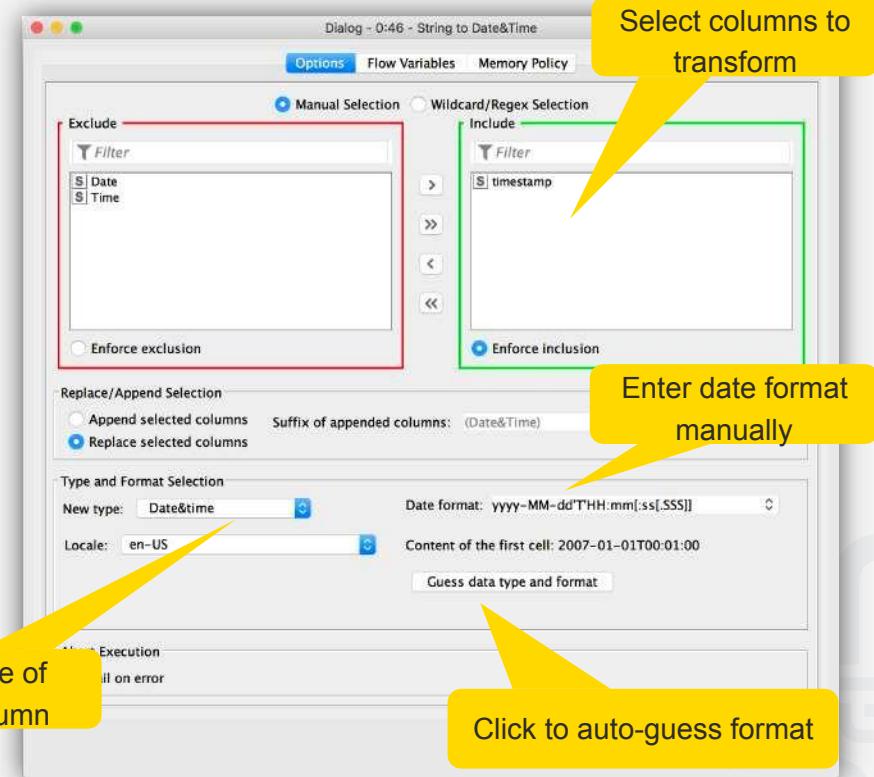


String to Date&Time

- Convert date/time data from String into a native Date&time cell
- Guesses correctly many date/time formats in String columns
 - Enter format manually if auto-guessing doesn't work
 - KNIME automatically adds custom formats to auto-guess list
 - Convert multiple columns that have the same date/time format by one node



Select type of output column



Date&Time – Data Types

Output table - 2:50 - Create Date&Time Range
File Hilit Navigation View
Table "default" - Rows: 1000
Row ID Date
Row0 2017-01-19
Row1 2017-01-19
Row2 2017-01-20
Row3 2017-01-20
Row4 2017-01-20
Row5 2017-01-21
Row6 2017-01-21
Row7 2017-01-22
Row8 2017-01-22
Row9 2017-01-22

Date

Dialog - 2:50 - Create Date&Time Range
File Hilit Navigation View
Options Flow Variables Job Manager Selection Memory Policy
Output Settings Date
Output type: Date&time
 Date&time with zone
New column name: Date&TIME
Mode Selection
Number of rows: Fixed: 1,000
 Variable
Starting Point
Start: Date: 2017-01-19 Time: 13:00:46
Time Zone: Europe/Berlin
 Use execution date&time
Ending Point
 Interval:
 End: Date: 2018-01-19 Time: 14:00:46
 Use execution date&time
OK Apply Cancel ?

Output table - 2:50 - Create Date&Time Range
File Hilit Navigation View
Table "default" - Rows: 1000
Row ID Date&Time
Row0 2017-01-19T13:00:46
Row1 2017-01-19T21:46:57
Row2 2017-01-20T06:33:08
Row3 2017-01-20T15:19:20
Row4 2017-01-21T00:05:31
Row5 2017-01-21T08:51:42
Row6 2017-01-21T17:37:53
Row7 2017-01-22T02:24:04
Row8 2017-01-22T11:10:15
Row9 2017-01-22T19:56:27
Row10 2017-01-23T04:42:38

Date & Time

Output table - 2:50 - Create Date&Time Range
File Hilit Navigation View
Table "default" - Rows: 1000
Row ID Time
Row0 14:02:31.155
Row1 14:02:31.155
Row2 14:02:31.155
Row3 14:02:31.155
Row4 14:02:31.155
Row5 14:02:31.155
Row6 14:02:31.155
Row7 14:02:31.155
Row8 14:02:31.155
Row9 14:02:31.155
Row10 14:02:31.155

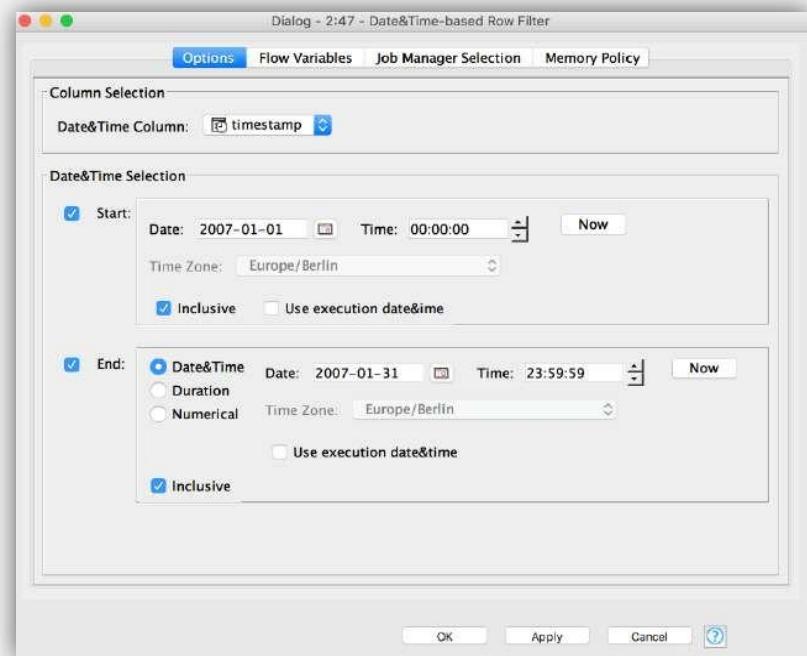
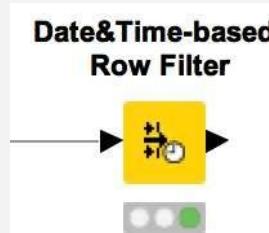
Time

Output table - 2:50 - Create Date&Time Range
File Hilit Navigation View
Table "default" - Rows: 1000 Spec - Column: 1
Row ID Date&Time
Row0 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row1 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row2 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row3 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row4 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row5 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row6 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row7 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row8 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row9 2018-01-19T14:02:31.155+01:00(Europe/Berlin)
Row10 2018-01-19T14:02:31.155+01:00(Europe/Berlin)

Date & Time +
zone

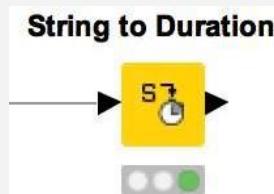
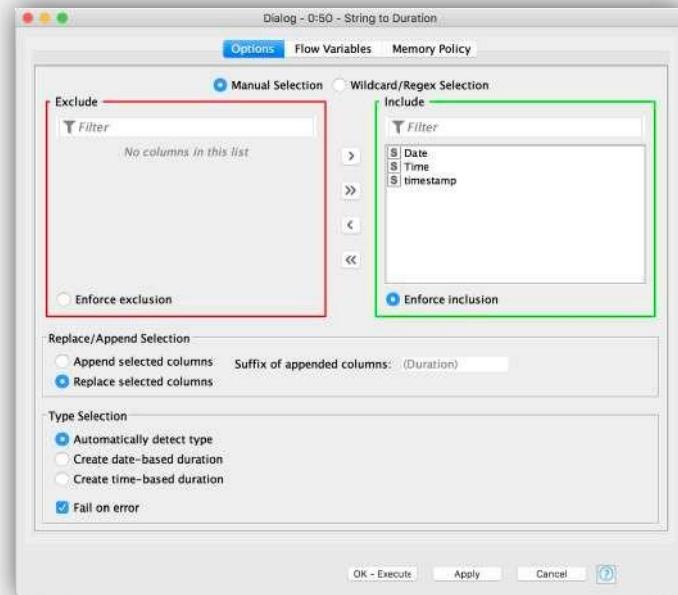
Date&Time-based Row Filter

- Filter rows from a specified time period
- Range can be limited on upper bound, lower bound, or both
- Options for end point:
 - Date&Time: Fixed date and time
 - Duration: Duration string (e.g. 2y 3M)
 - Numerical: Select granularity from a dropdown menu and enter a number



String to Duration

- Takes a String and converts it to a duration cell
- Three different options to format input Strings
- Example: Convert 1 year, 2 months, 3 weeks, and 4 days to duration cell
 - ISO-8601: “P1Y2M3W4D”
 - Short letter: “1y 2M 3w 4d”
 - Long word: “1 year 2 months 3 weeks 4 days”

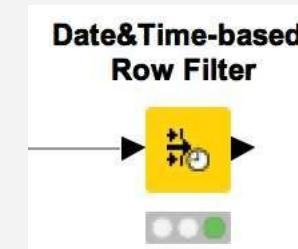


Output table - 0:51 - String to Duration

Row ID	iso	short	long	Spec - Columns: 6	Properties	Flow Variables
Row0	P1Y2M3W4D	1y 2M 3w 4d	1 year 2 months 3 weeks 4 days	1y 2M 25d	1y 2M 25d	1y 2M 25d

Duration-based Filtering

- Date&Time-based Row Filter allows to extract time periods
- From the start date, select all rows within the defined period
- Use one of the three options to define the duration, e.g.
 - ISO-8601: “P1Y2M3W4D”
 - Short letter: “1y 2M 3w 4d”
 - Long word: “1 year 2 months 3 weeks 4 days”



Output table - 0:50 - Create Date&Time Range

Row ID	Time
Row0	2018-01-19T14:02:31.155
Row1	2018-01-27T00:49:02.991
Row2	2018-02-03T11:35:34.828
Row3	2018-02-10T22:22:06.665
Row4	2018-02-18T09:08:38.502
Row5	2018-02-25T19:55:10.338
Row6	2018-03-05T06:41:42.175
Row7	2018-03-12T17:28:14.012
Row8	2018-03-20T04:14:45.849
Row9	2018-03-27T15:01:17.686
Row10	2018-04-04T01:47:49.522
Row11	2018-04-11T12:34:21.359
Row12	2018-04-18T23:20:53.196
Row13	2018-04-26T10:07:25.033
Row14	2018-05-03T20:53:56.870
Row15	2018-05-11T07:40:28.706

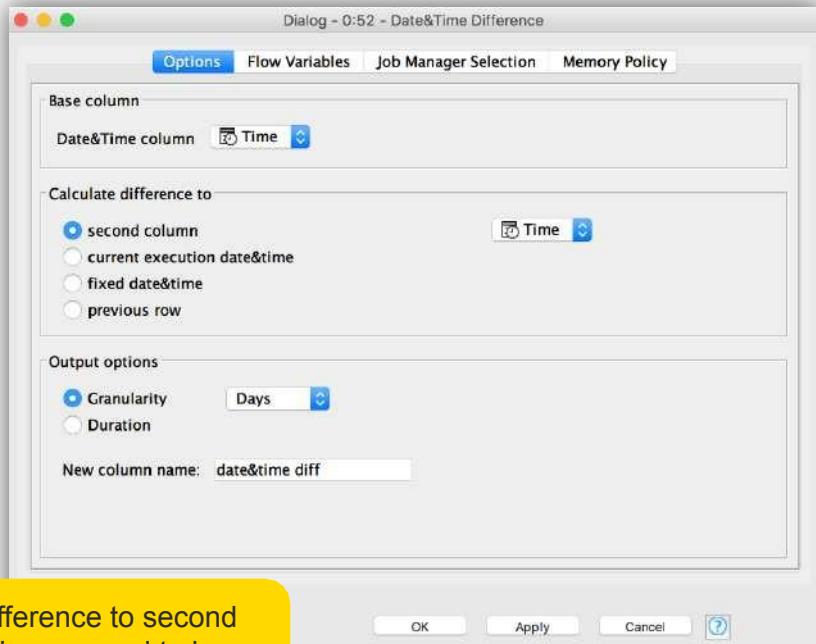


Output table - 0:60 - Date&Time-based Ro...

Row ID	Time
Row2	2018-02-03T11:35:34.828
Row3	2018-02-10T22:22:06.665
Row4	2018-02-18T09:08:38.502
Row5	2018-02-25T19:55:10.338

Date&Time Difference

- Choose desired granularity (days, hours, minutes, etc.)
- Check the difference between a time column and...
 - Another time column
 - Execution time
 - User-defined time
 - Previous row

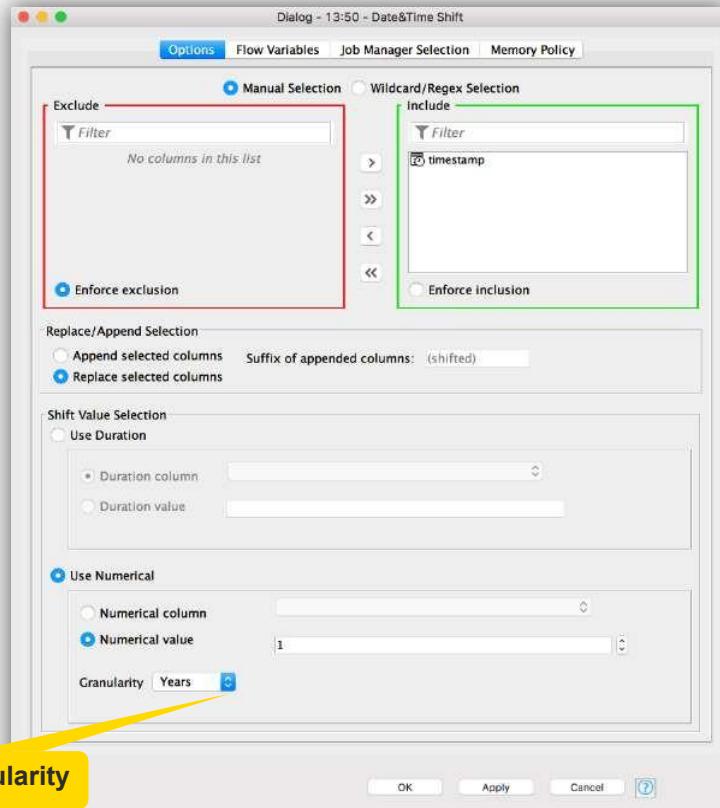


Date&Time Shift

- Shifts date or time by either a duration or a numerical value
- Use duration:
 - Use duration column
 - Or shift by user defined value
 - E.g. 1y, 2M, 5h, etc.
- Use numerical value in combination with selected granularity
 - Use numerical column
 - Or shift by user defined value

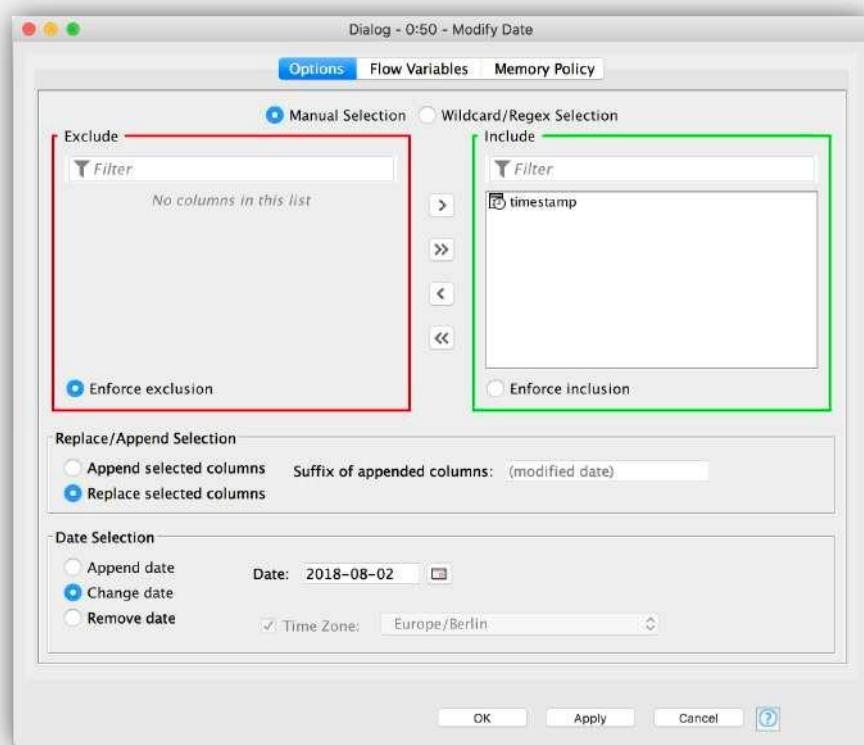
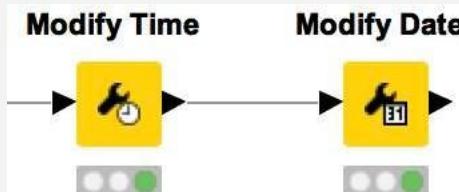


Select granularity



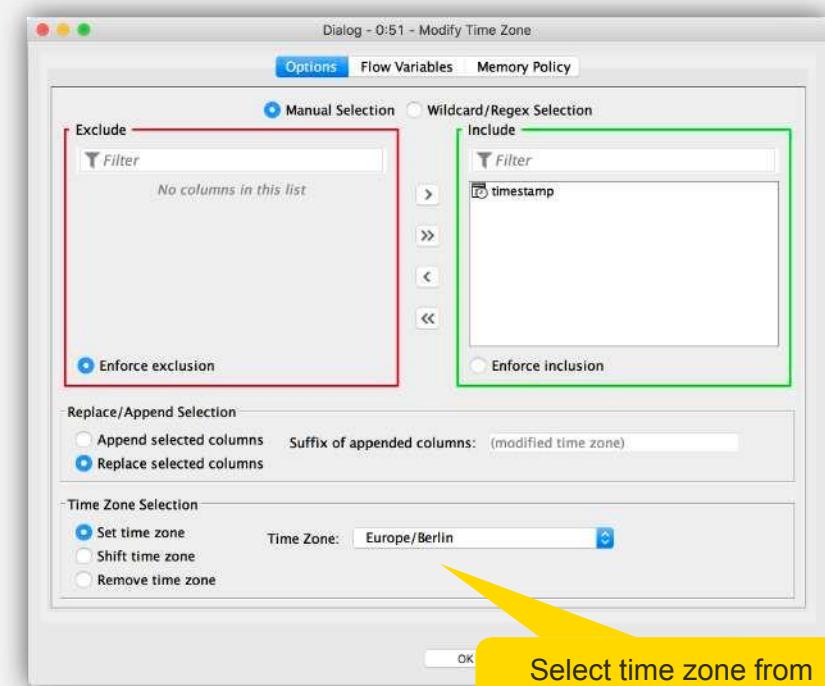
Modify Time / Modify Date

- Modify Date&Time columns
- Three options:
 - Append time (date) to a date (time) column
 - Change time (date) to a fixed value
 - Remove time (date) from a Date&Time column
- Column selection shows only columns that can be modified by the current configuration



Modify Time Zone

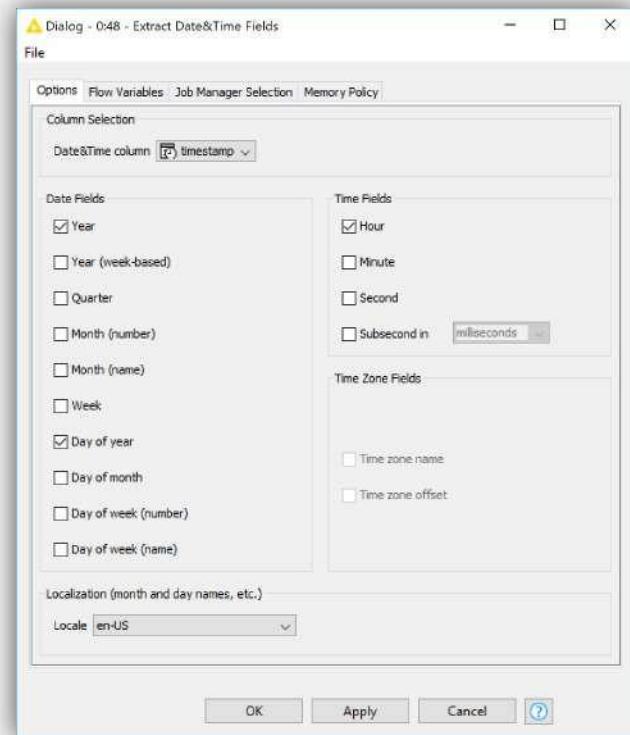
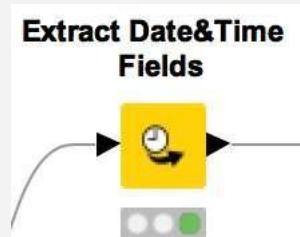
- Similar to Modify Time/Modify Date
- Input: Date&Time
 - Set time zone
- Input: Date&Time (Time zone)
 - Set time zone
 - Shift time zone
 - Remove time zone



Extract Date&Time

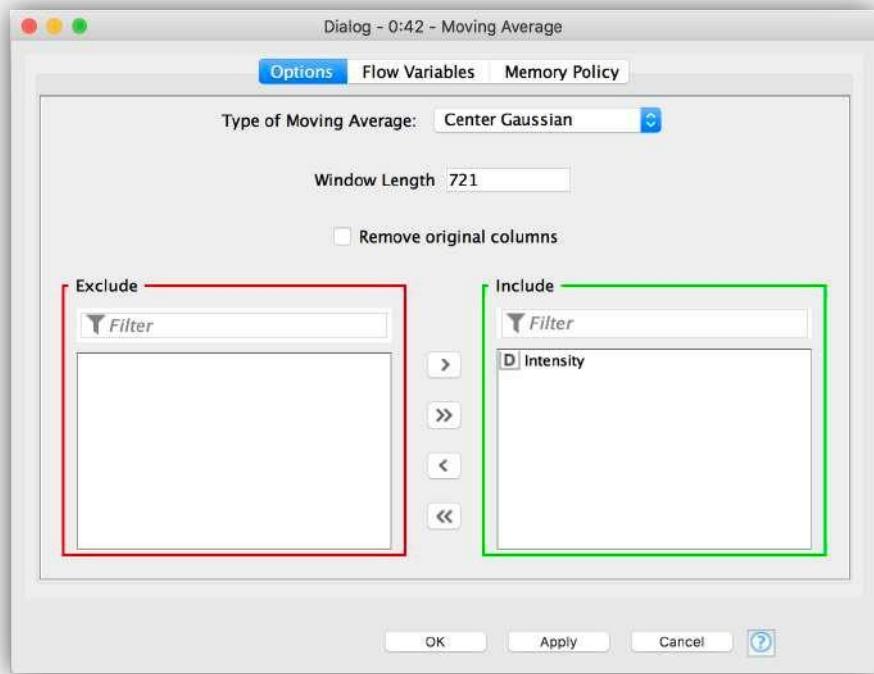
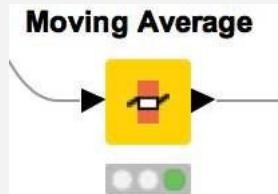
Fields

- Extract date fields (year, day, month,...) or time fields (hour, minute, second,...) from a Date&Time cell
- Pick and choose which fields to include
- Useful when used in combination with data aggregation nodes (GroupBy, Pivoting, etc.)



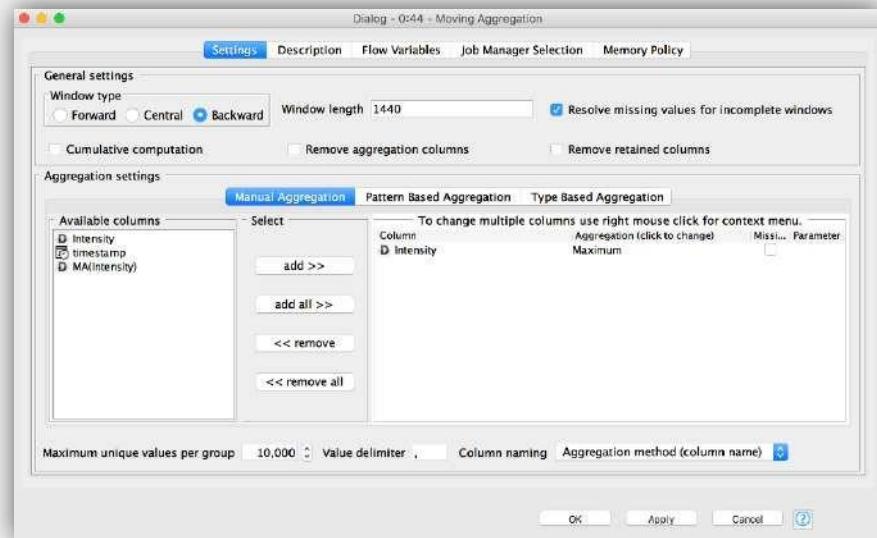
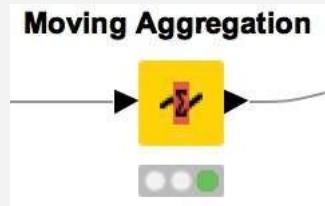
Moving Average

- Effective “smoothing” node
- Smoothing defined by
 - window type (centered, forward or backward)
 - window length
 - weighted or not
- Useful when plotting aggregated time series data to more easily see trends

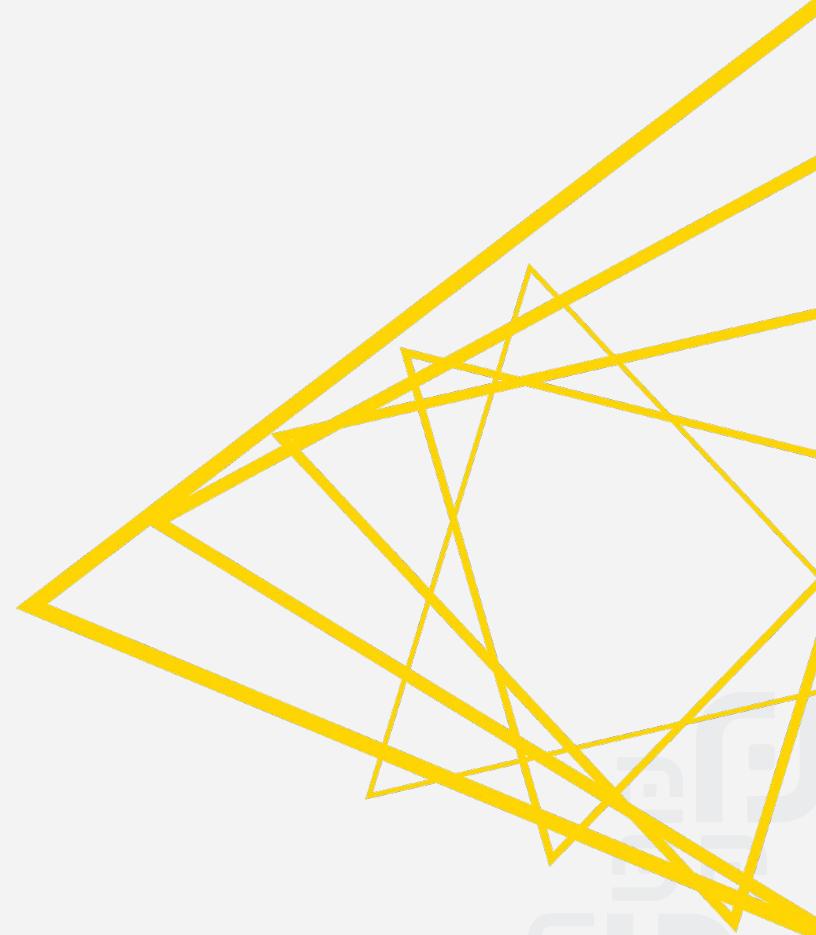


Moving Aggregation

- Blend of GroupBy + Moving Average Functionality
- Group by moving window
- Aggregate using standard KNIME methods

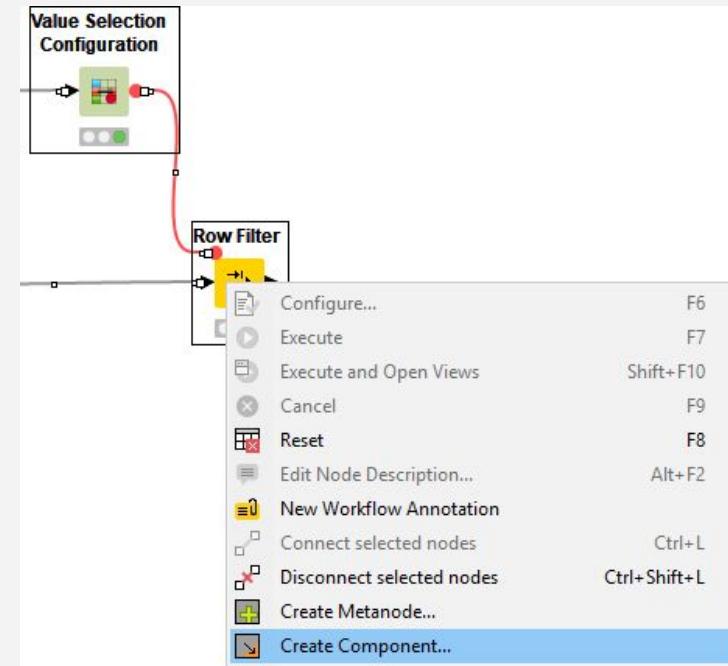


Workflow Organization and Documentation



Create a Component

- Select nodes to encapsulate into a Component
- Right click a node
- Select “Create Component...”



Component Description

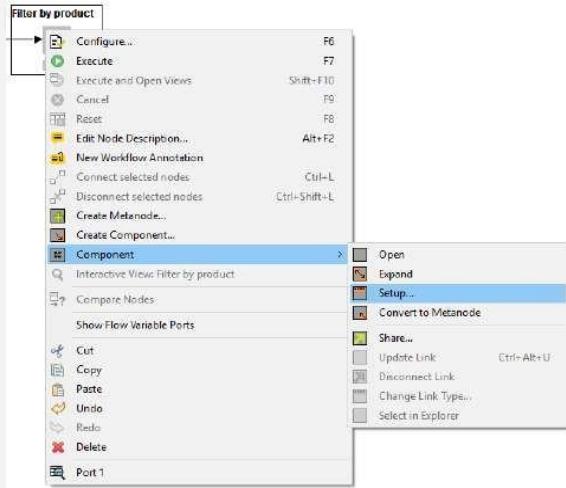
Component look like a KNIME node

The image shows the KNIME interface with a component configuration dialog open. The configuration dialog has three tabs: 'Description', 'Component Icon', and 'Ports'.

- Description:** Contains the text: "This component filters the data by the selected product name". A yellow callout points to this text with the instruction: "Add description of the component".
- Component Icon:** A placeholder for a square image file, with instructions: "Drag and Drop a square image file" and "PNG image of size 16x16 or larger". A yellow callout points to this area with the instruction: "Add description of the input and output ports".
- Ports:** Shows two ports:
 - In Port #1:** Name: "Filtered data", Description: "Data containing records for all products".
 - Out Port #1:** Name: "Filtered data", Description: "Data containing records for the selected product".A yellow callout points to this area with the instruction: "Add background color or icon".

Below the configuration dialog, the KNIME workflow interface shows a 'Table Reader' node connected to a 'Filter by Product' node. The 'Filter by Product' node has a configuration dialog open, showing the 'Dialog Options' tab with the text: "Select product: The product name to use for filtering". The 'Ports' tab lists the input port "Customer data" and the output port "Data containing records for the selected product".

Configure Component Ports



- Add input and output ports to Metanodes/Components
- Remove ports to adapt to changes after creation of Metanode/Component

Setup Component Wizard

Specify the name of the node and define the number and type of the desired in and out ports.

Component Name: Filter by product

In Ports:

- in_1 (Data)

Add... Remove Up Down

Out Ports:

- out_1 (Data)

Add... Remove Up Down

Finish Cancel

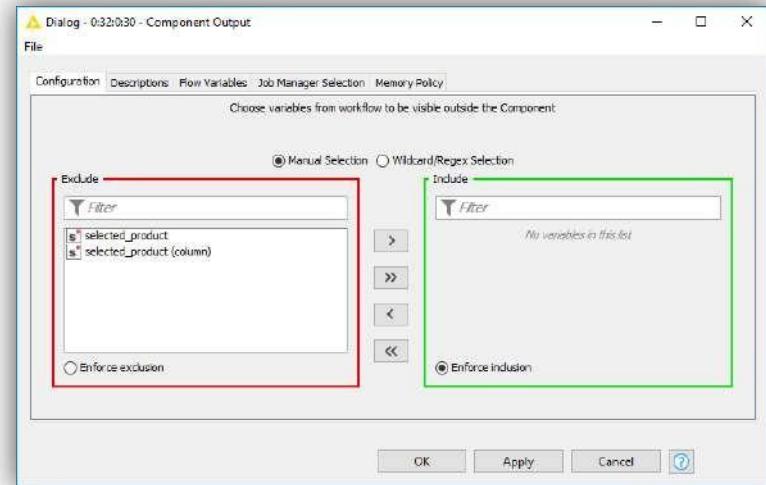
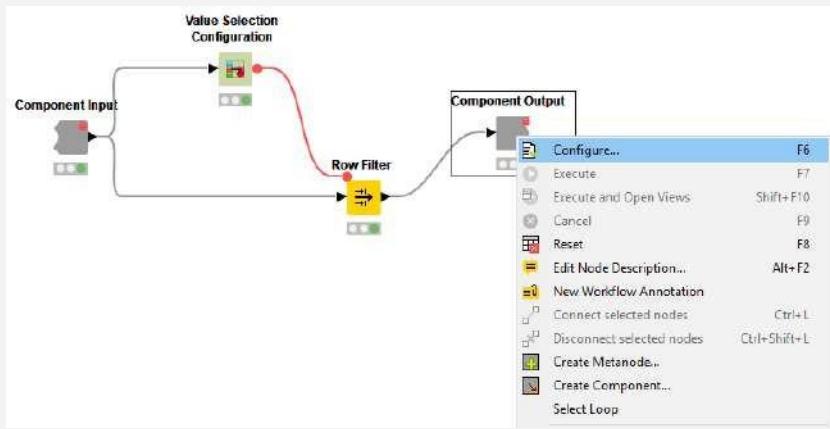
Add Meta Port

Port Type: Data

- Data
- Flowvariable
- PMML
- Database Connection
- Database Query
- AWS Comprehend Connection
- AWSConnection
- DB Data
- DB Session
- Distance Measure
- FilterDefinition
- Gradient Boosting Model
- H2O Context
- H2O Frame
- H2O Model
- Image
- KnimeConnection
- MOJO
- Outlier
- Python
- Python
- Regression Tree
- Regression Tree

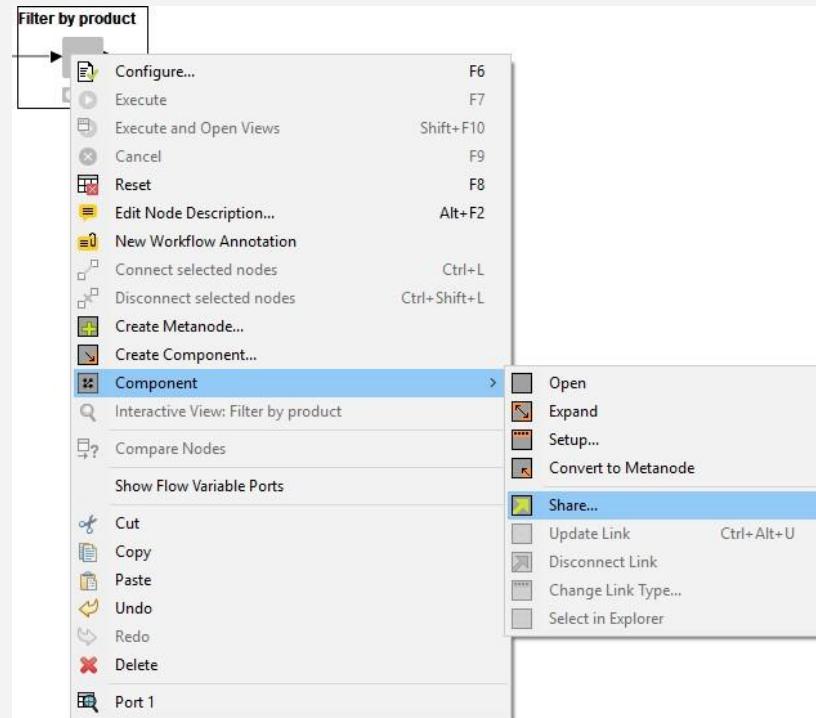
Passing Variables from Components

- Flow Variables by default only available locally inside Component
- Configure Component Input/Output to pass Flow Variables from/to outside Component



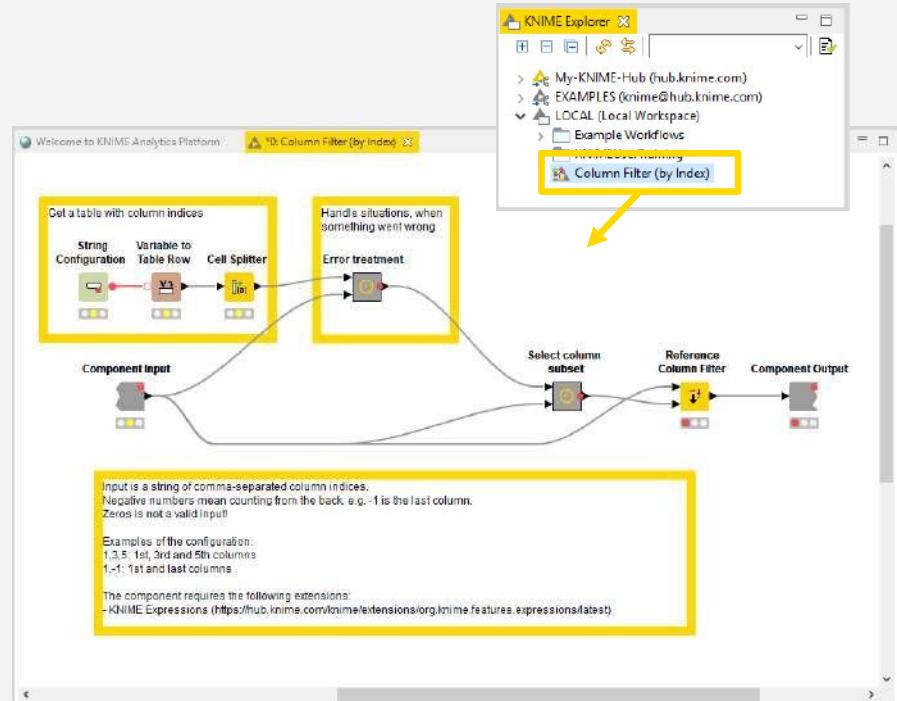
What is a Shared Component?

- Components can be saved in your KNIME workspace for later reuse
- To do this, simply right-click any Component and select “Share...”
- Shared Components are read-only instances of a Component
- Public Shared Components are available on EXAMPLES Server and on KNIME Hub



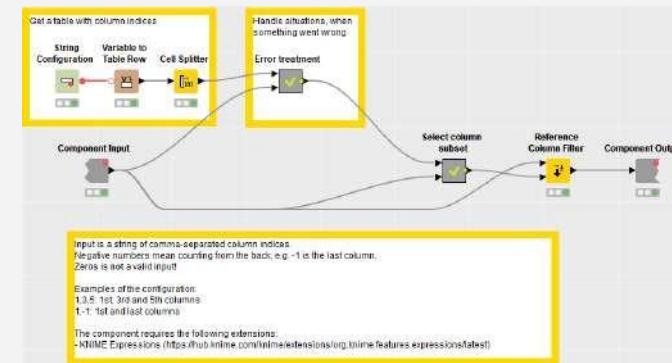
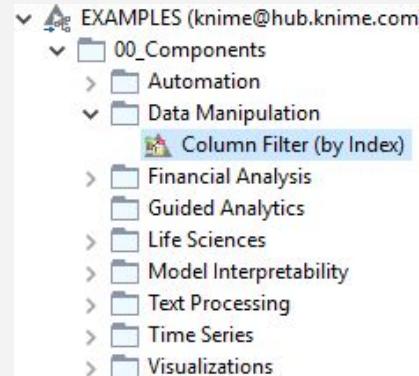
How can you Edit a Shared Component?

- Components can be edited using the Component Editor, similar to workflows
- To edit a Component using the Component Editor, double-click the Component in its location in the KNIME Explorer
- To ensure components are executable when opened in the Component Editor, chose the option to “Include input data with component” when sharing it



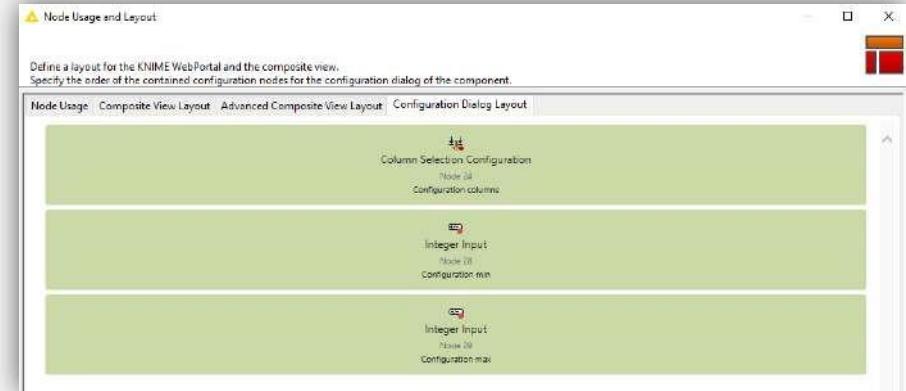
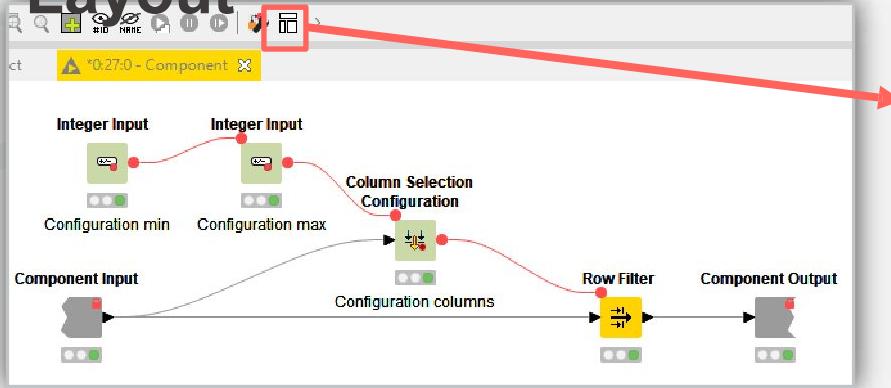
How can you Use a Shared Component?

- To use a Shared Component, drag and drop it to the workflow editor
- Instances of Shared Components can be updated either manually or when workflow is opened
- Shared Component can also be unlinked from its original location, which makes it editable in the workflow directly
- Update Shared Components by overwriting them

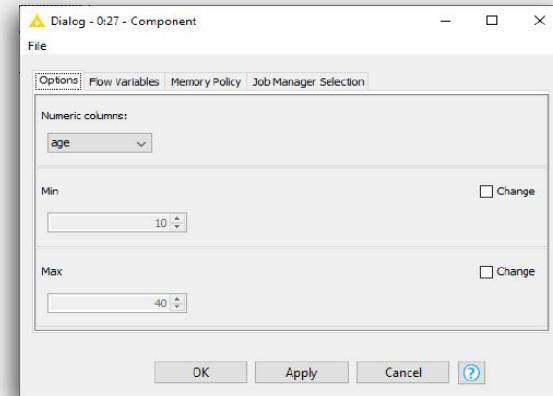


Configuration Dialog

Layout

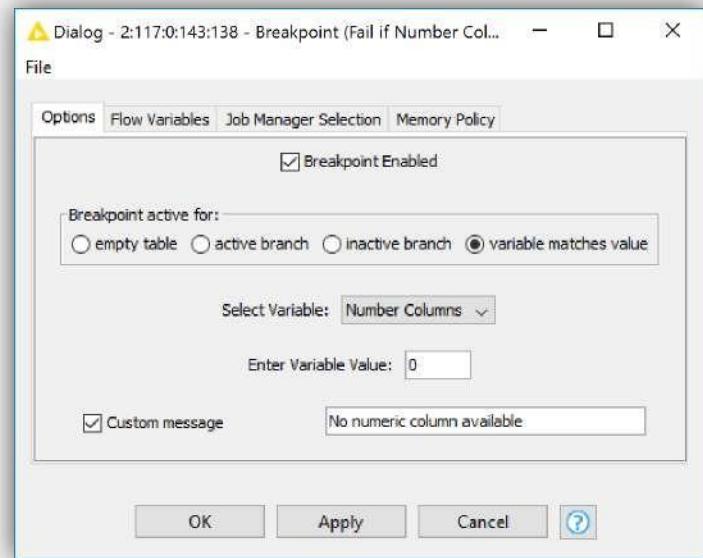
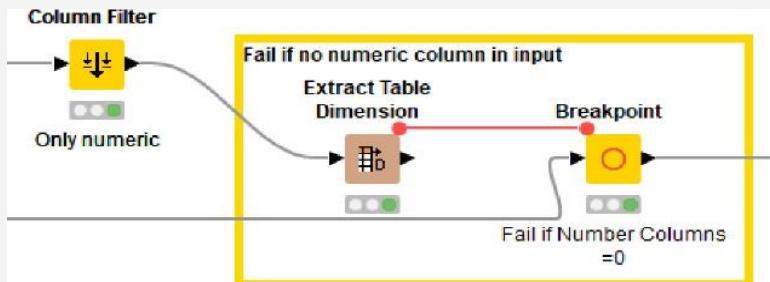


- Click layout button when inside component to modify the order of the setting options in configuration window of the component

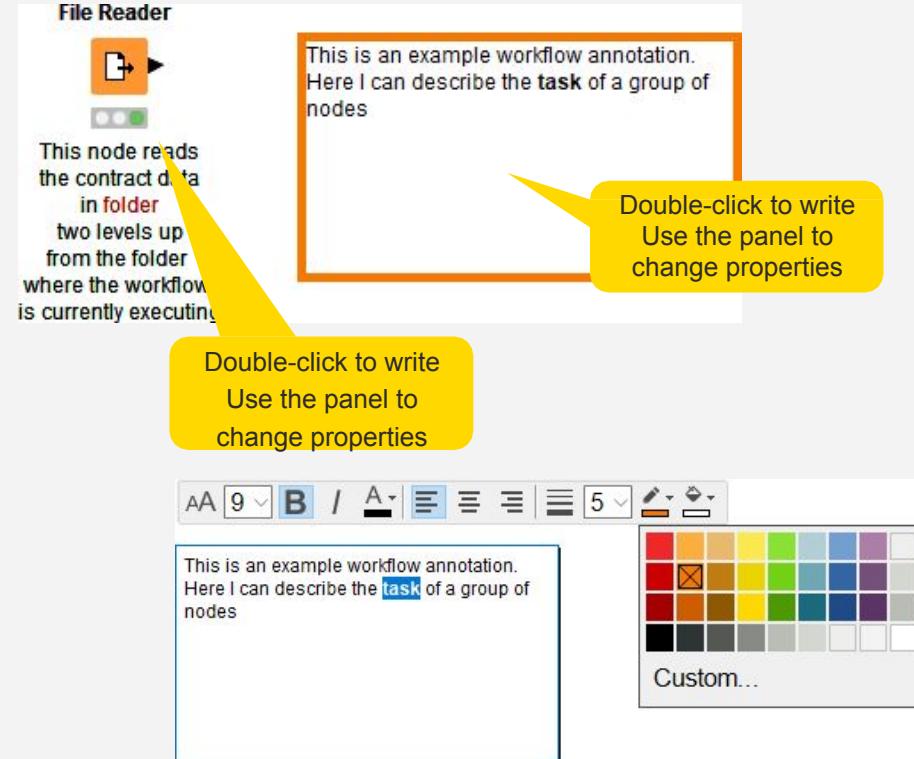
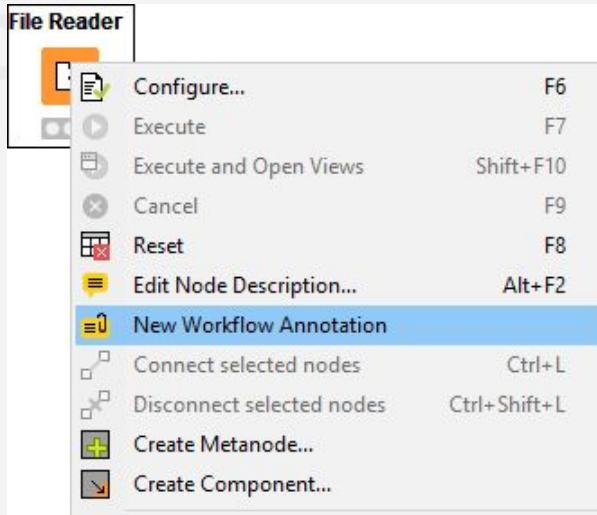


Breakpoint

- t Stops execution of a workflow branch
- Useful to stop the execution of a component and provide a custom error message
- Execution stops based on the selected condition:
 - Empty table
 - Active/Inactive branch
 - Flow Variable value



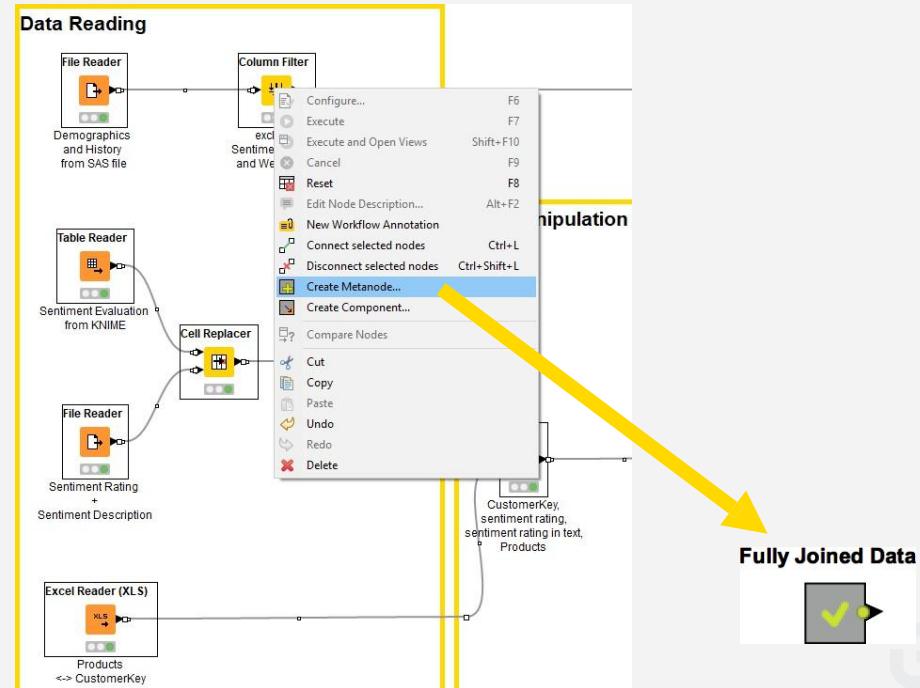
Comments & Annotations



YouTube KNIME TV Channel:
https://youtu.be/AHURYB_O8sA

Workflow Organisation – Good Practices

- Workflow annotations
- Node labels
- Metanodes
 - Right click -> Create Metanode...
 - Organize workflow by task
 - Hide complexity & improve readability



Workflow Organisation – Components

- Component encapsulates a reusable functionality as a KNIME workflow
- Components can be configured as any KNIME nodes
- Access and share components on the KNIME Hub

The screenshot shows the KNIME Hub interface with a search bar containing "column filter". Below the search bar, there are tabs: All, Nodes, Components (which is highlighted with a yellow circle), Workflows, and Extensions. The search results show 36 results. One result, "Interactive Column Filter", is displayed with its description and a "Component" button. Another result, "Column Filter (by Index)", is shown with its description and a "Component" button. A large yellow callout bubble points from the text "Drag and drop from the KNIME Hub to your workflow" towards the "Column Filter (by Index)" component. Below the callout, a detailed diagram of the "Column Filter (by Index)" node is shown, illustrating its internal structure and how it connects to other nodes like "String Configuration" and "Error treatment".

KNIME Hub - Search

36 results

All Nodes Components Workflows Extensions

Interactive Column Filter

This Component creates an interactive view to filter and select columns for your model.
paolotamag > Public > Interactive Column Filter

Component

Column Filter (by Index)

This component allows to filter columns by their index (i.e. position). An example can be: select the second and the last columns (e.g. with "2,-1"). Configuration takes a string of column positions: ...
knime > Examples > 00_Components > Data Manipulation > Column Filter (by Index)

Component

Drag and drop from the KNIME Hub to your workflow

Column Filter (by Index)

Get a table with column indices
String Configuration Variable to Table Row Cell Splitter

Handle situations, when something went wrong
Error treatment

Component Input
Component Output

Select column subset Reference Column Filter

Input is a string of comma-separated column indices.
Negative numbers mean counting from the back, e.g. -1 is the last column.
Example of the configuration:
1,3,5: 1st, 3rd and 5th columns
1:-1: 1st and last columns

The component requires the following extensions:
- KNIME Expressions (<https://hub.knime.com/knime/extensions/org.knime.features.expressions@latest>)

KNIME Workflow Diff

- Automates identification and comparison of nodes in a workflow, metanodes, and two different workflows
- Identifies insertions, deletions, substitutions, and parameter changes

The screenshot shows the KNIME Node Comparison interface. At the top, there are two 'Column Filter' nodes: 'Old' and 'New'. Below them is a comparison table divided into two sections: 'Column Filter 0:16' and 'Column Filter 0:15'. The 'Old' section contains three rows under 'included_names': array-size (int, value 3), 0 (string, value petal length), and 1 (string, value petal width). The 'New' section also has three rows under 'included_names': array-size (int, value 3), 0 (string, value sepal length), and 1 (string, value sepal width). A red box highlights the 'petal length' and 'sepal length' entries. The 'Excluded Names' section is collapsed. The 'System Node Settings' section is also collapsed.

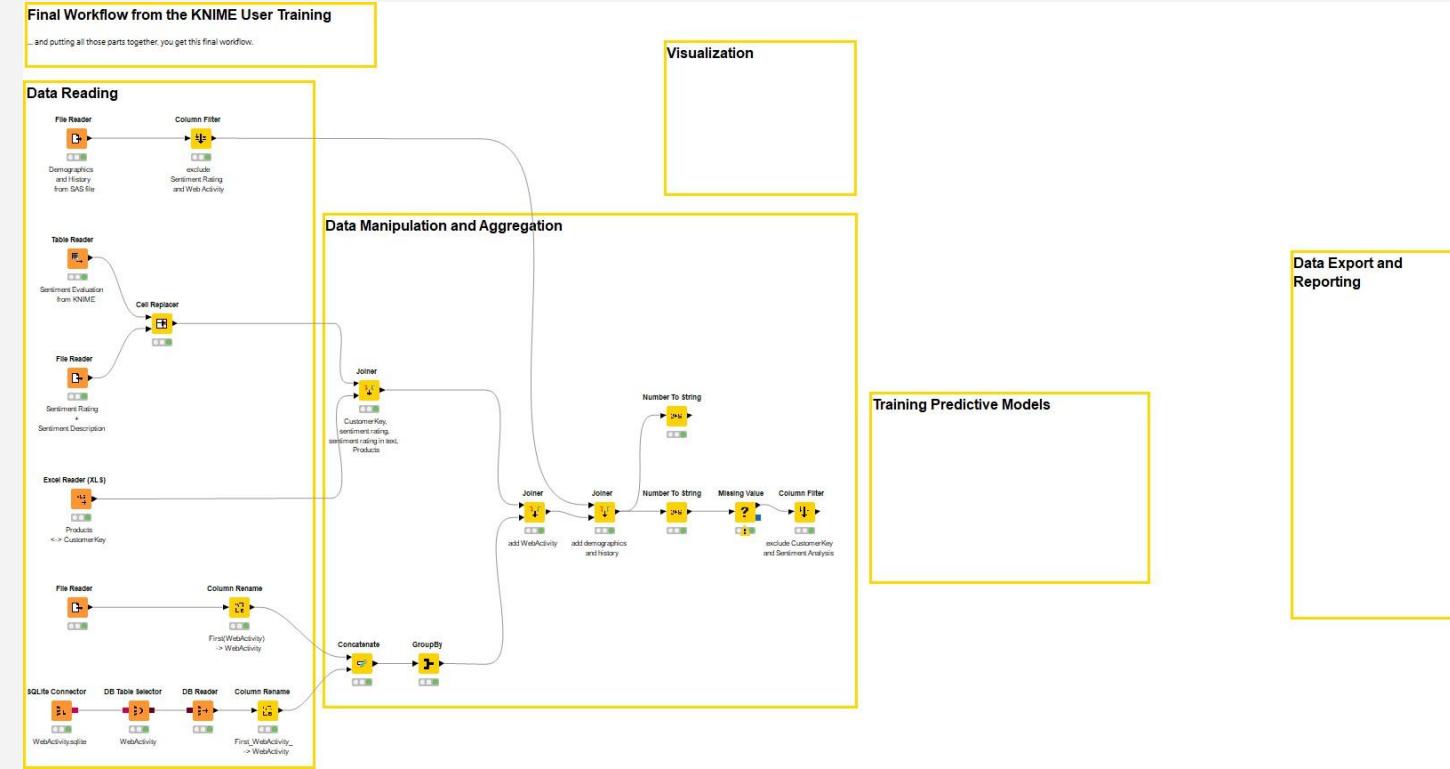
The screenshot shows the KNIME Workflow Comparison interface. It displays two workflows: 'LOCAL/03_Sentiment_Classification' and 'LOCAL/03_Sentiment_Classification_v2'. The left pane lists nodes from both workflows, while the right pane shows a detailed comparison between specific nodes. A blue box highlights the 'Snowball Stemmer' node in both workflows. The bottom pane shows a 'Node Settings Comparison' table for the 'Snowball Stemmer' node, where the 'Porter' setting is highlighted in red.

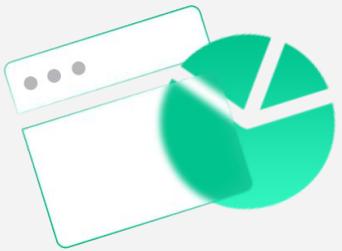
Data Manipulation Exercise

Start with exercise *Data Manipulation*

- Join all data into one table using a series of joiner nodes (use "Customer Key" as the joining column)
- Filter out duplicate rows
- Clean up and document your workflow using annotations, node labels, and metanodes

Today's Example





Terima Kasih

SIB Cycle 6 | 2024



www.greatedu.co.id