

KNIME Analytics Platform

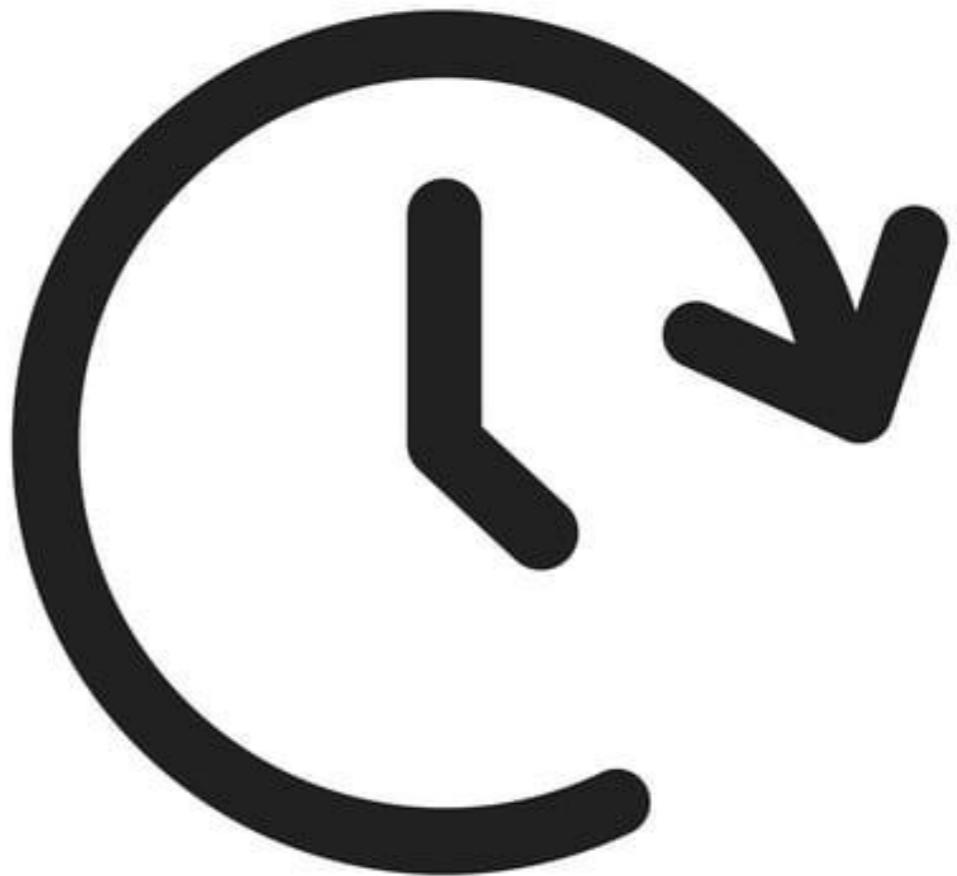
INTRODUCTION TO KNIME

HR needs your help



Requirements by HR

- Quick - the deadline is approaching!



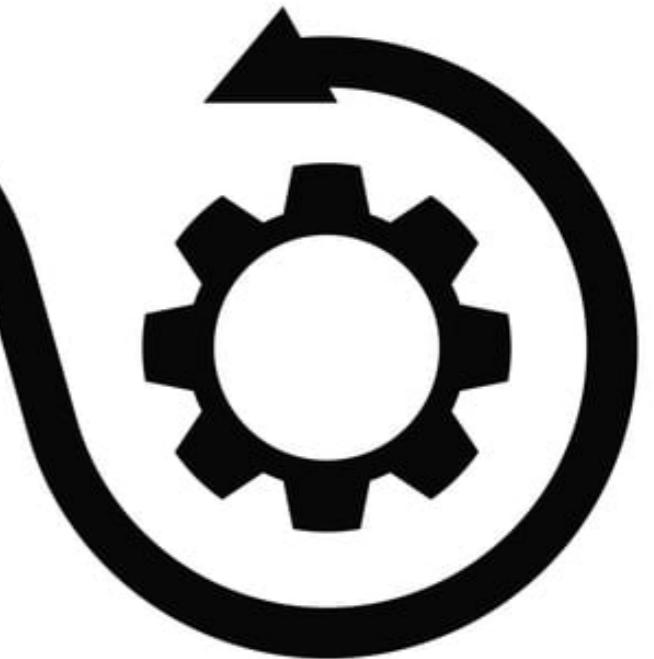
Requirements by HR

- Quick - the deadline is approaching!
- Understandable by non-technical users



Requirements by HR

- Quick - the deadline is approaching!
- Understandable by non-technical users
- Reusable and automated solution



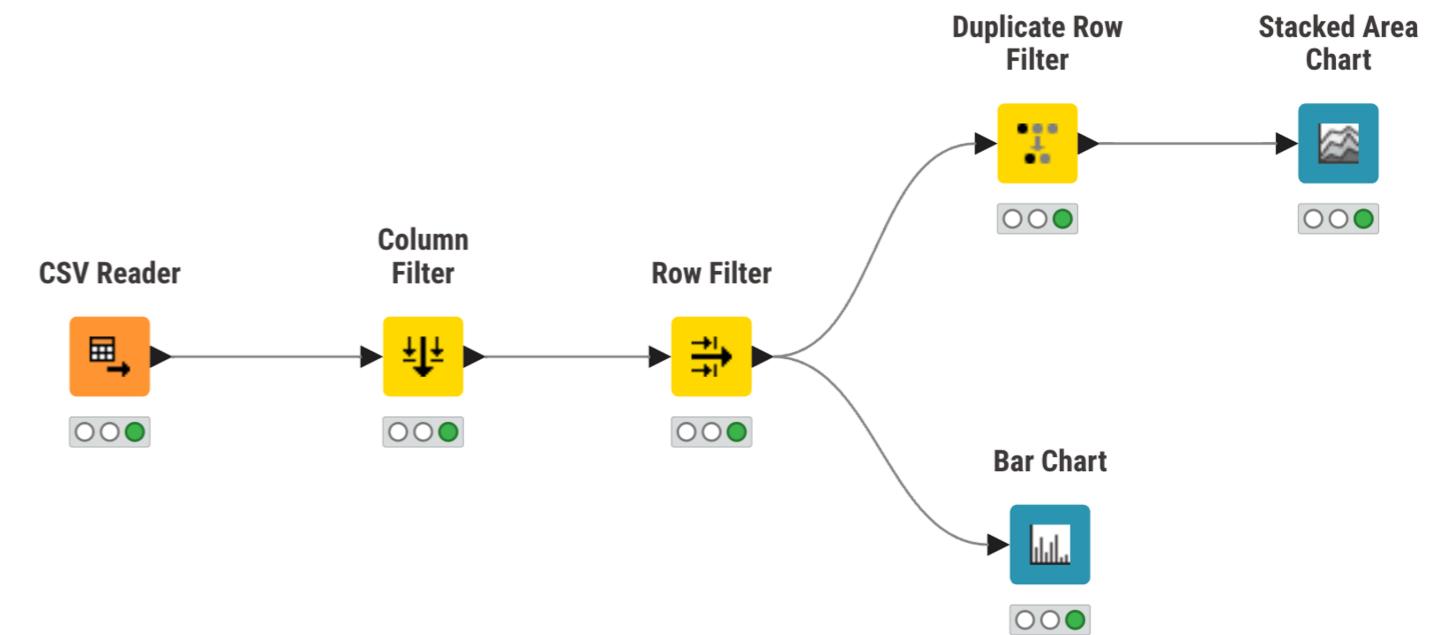
KNIME Analytics Platform

- No-code/low-code



KNIME Analytics Platform

- No-code/low-code
- Visual programming



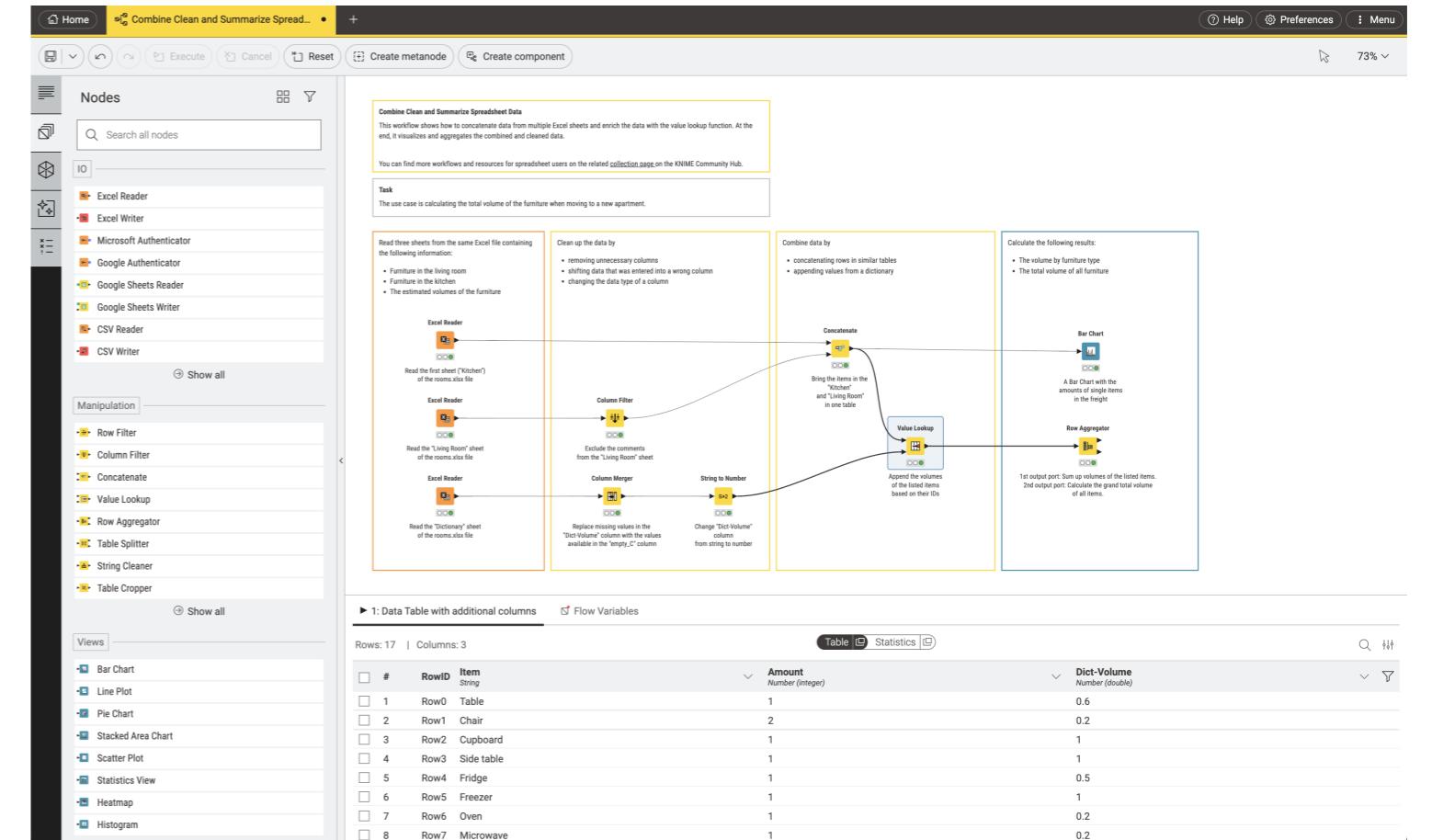
KNIME Analytics Platform

- No-code/low-code
- Visual programming
- Easy to use



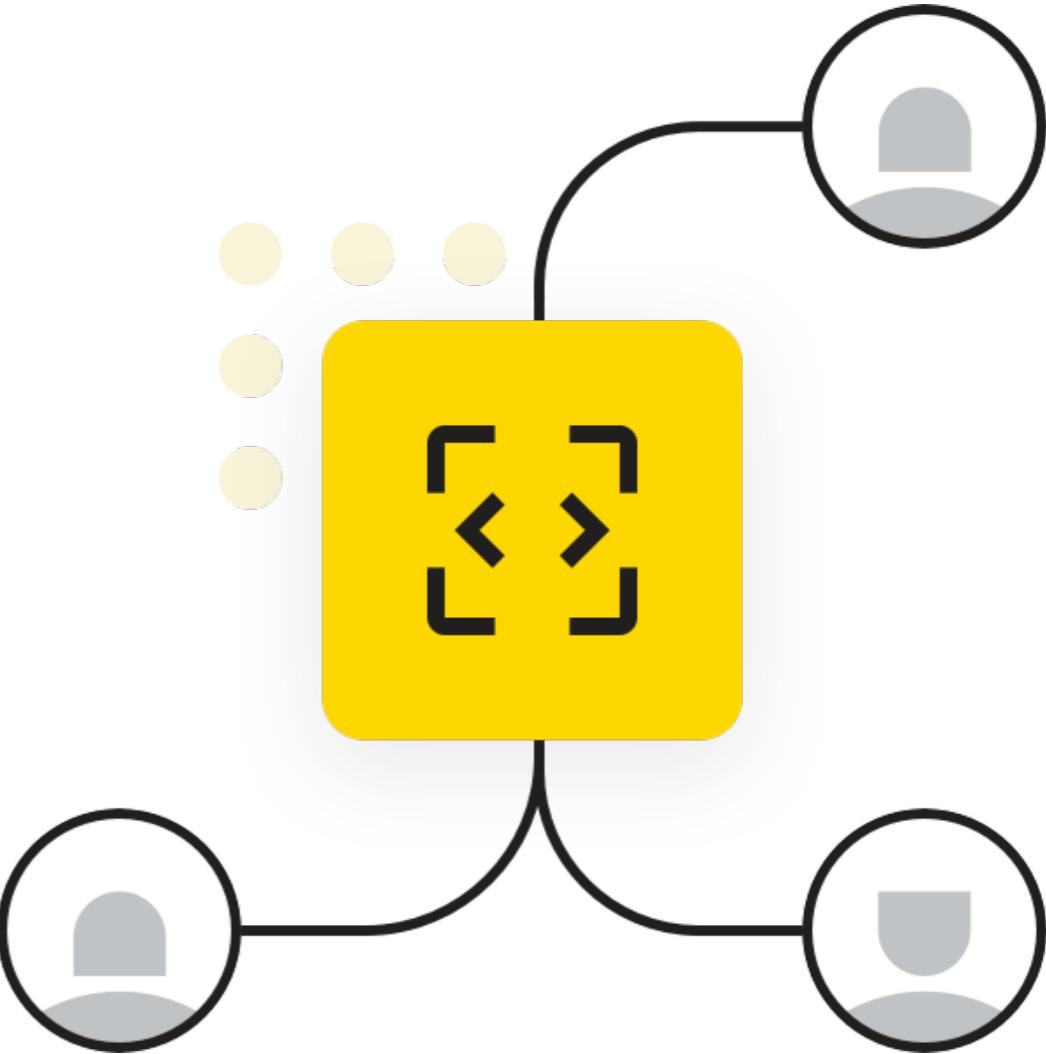
KNIME Analytics Platform

- No-code/low-code
- Visual programming
- Easy to use
- Process-oriented solution



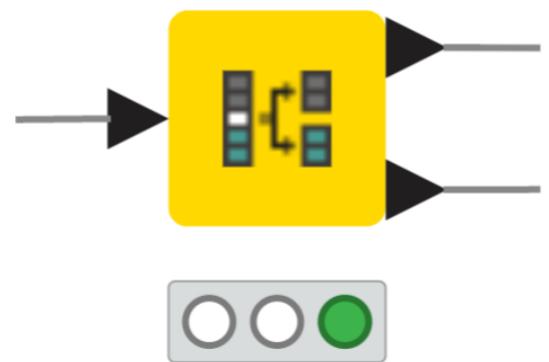
KNIME Analytics Platform

- No-code/low-code
- Visual programming
- Easy to use
- Process-oriented solution
- Free and open source

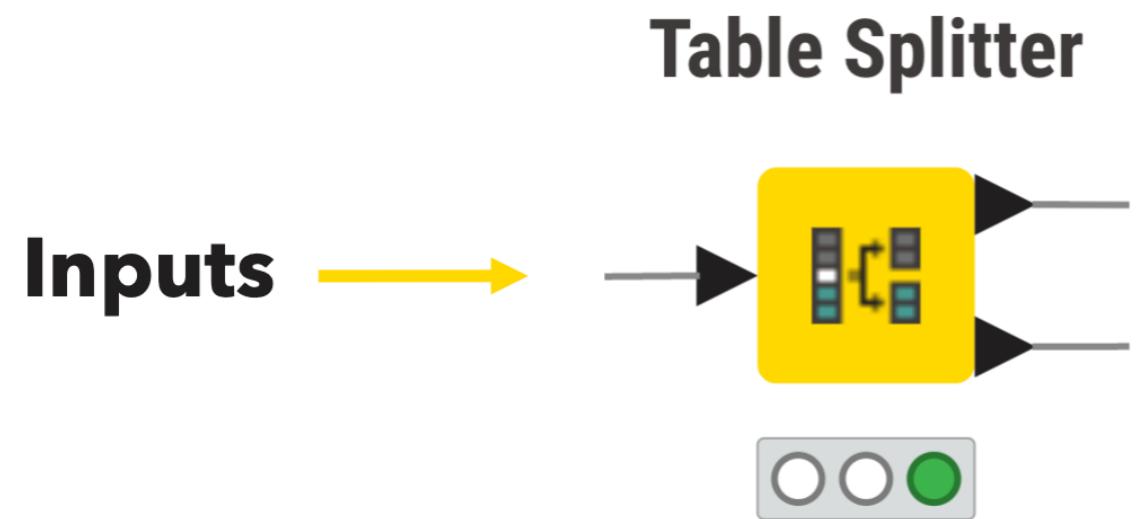


Nodes

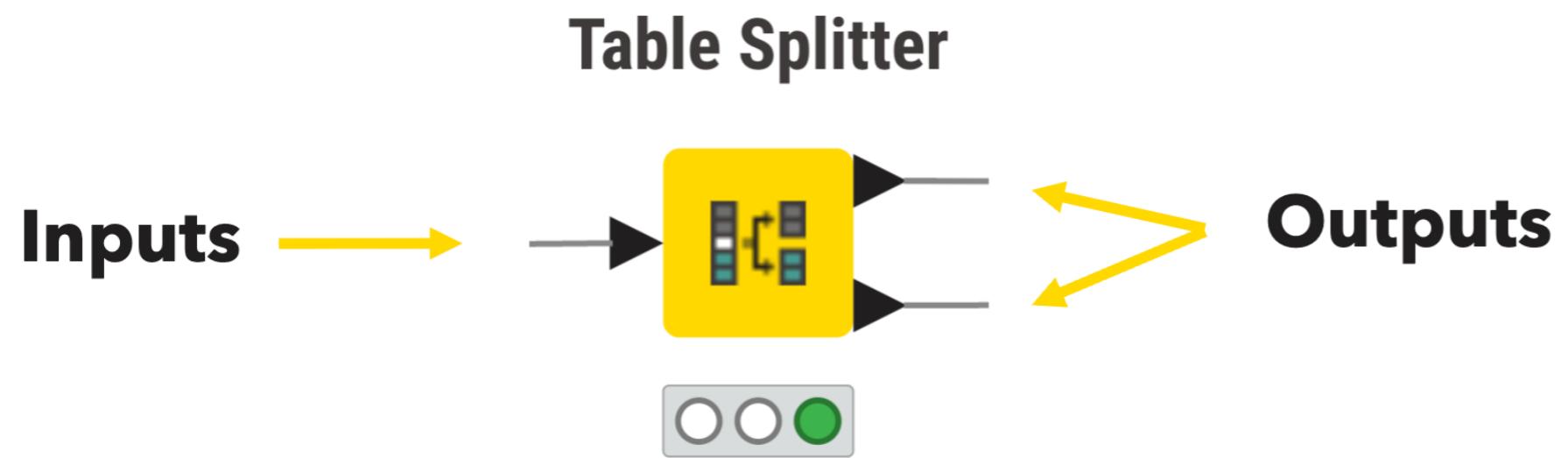
Table Splitter



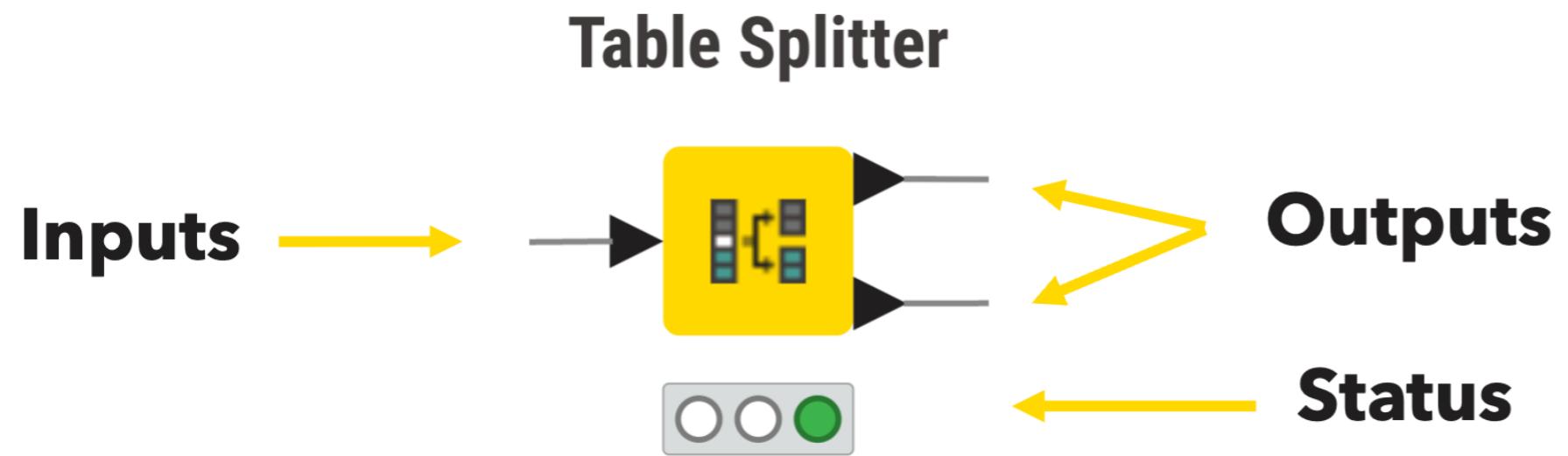
Nodes



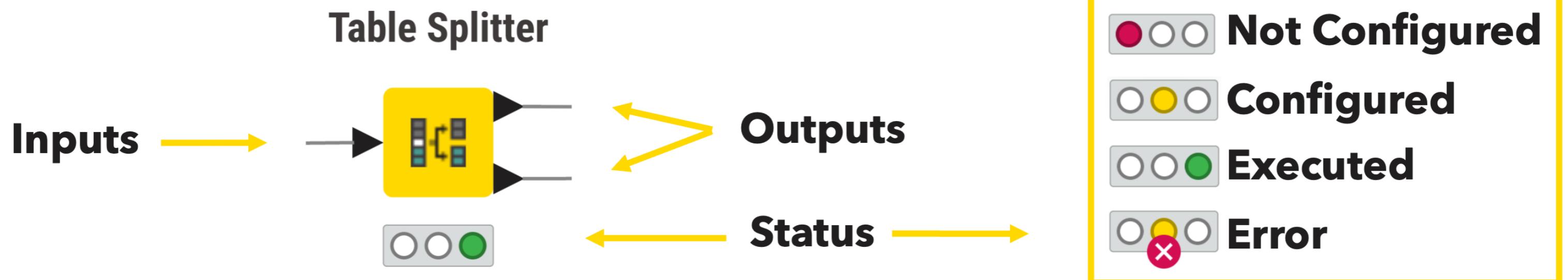
Nodes



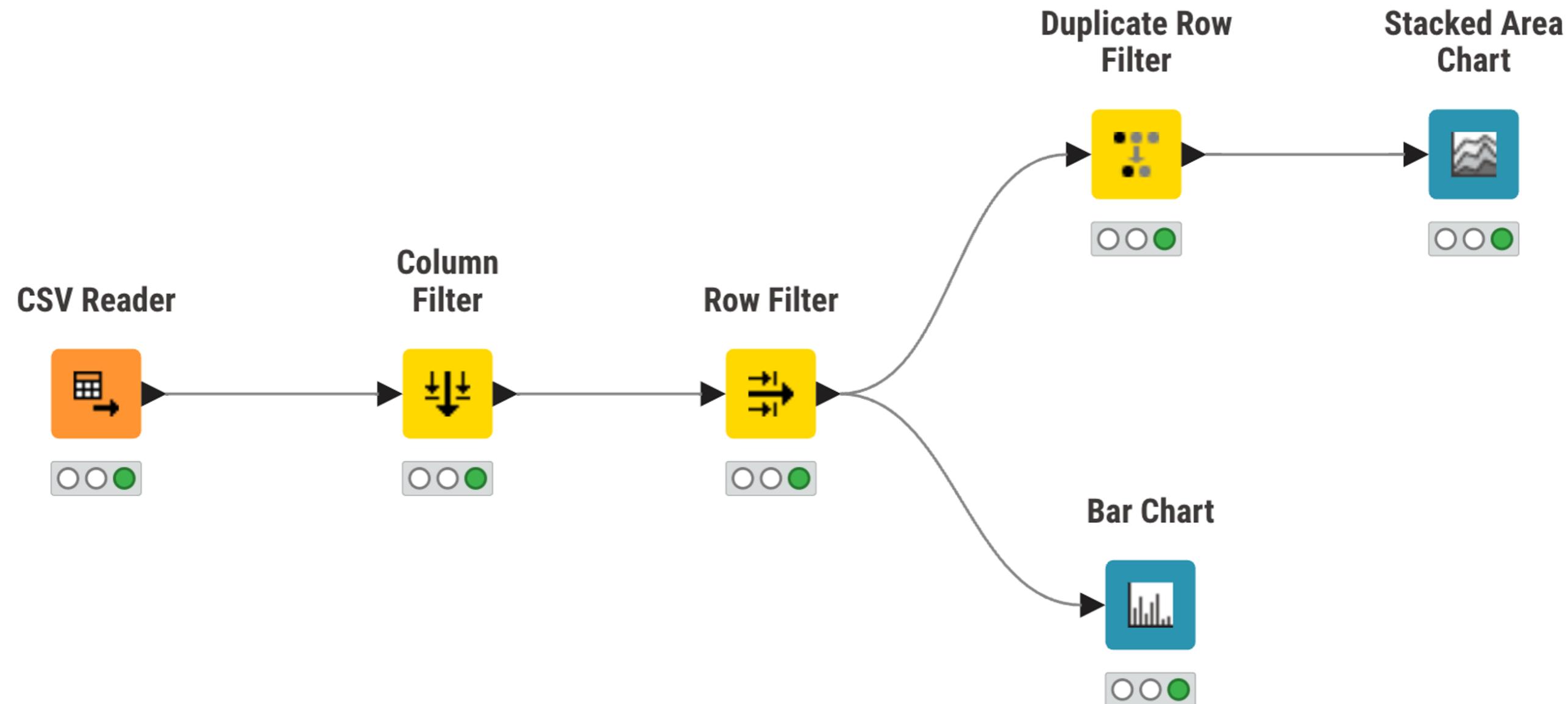
Nodes



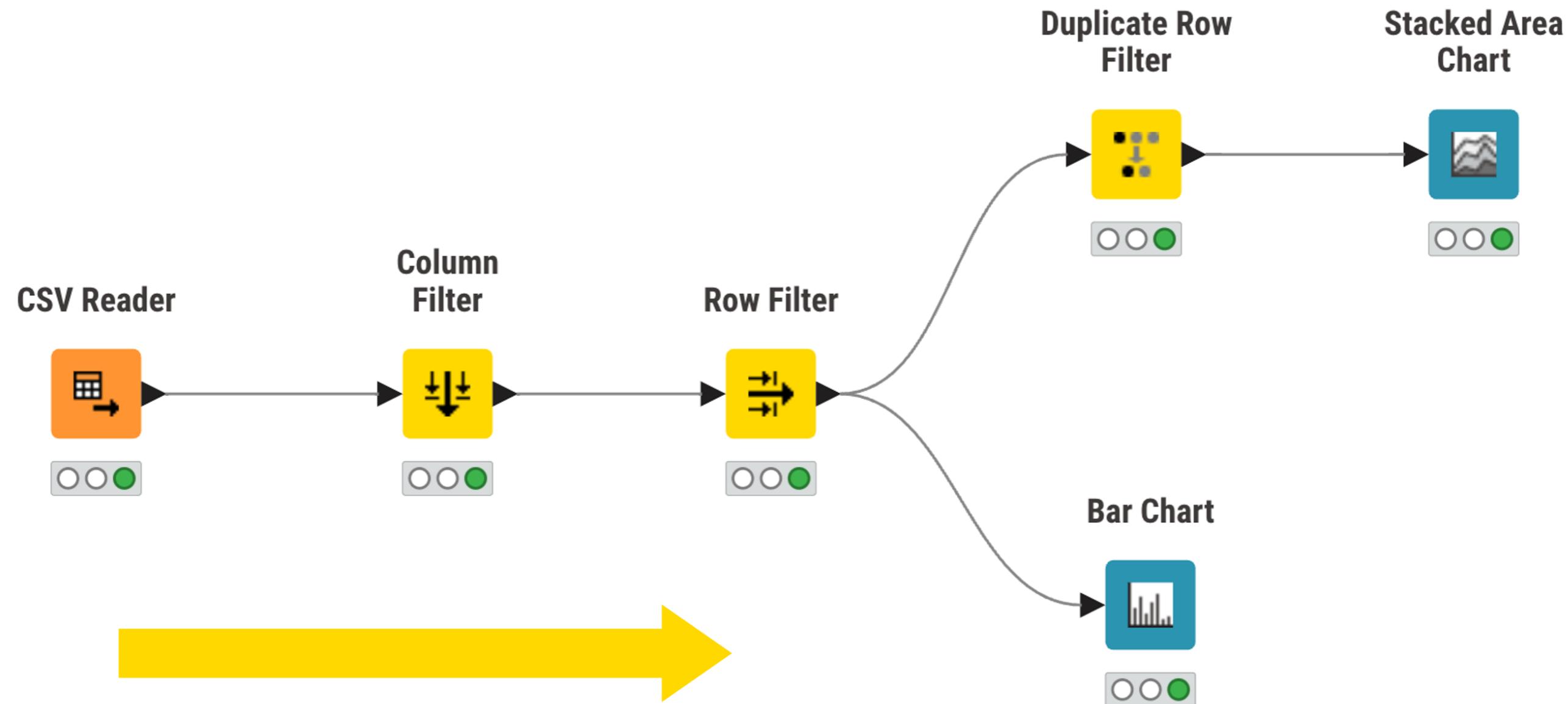
Nodes



Workflows



Workflows



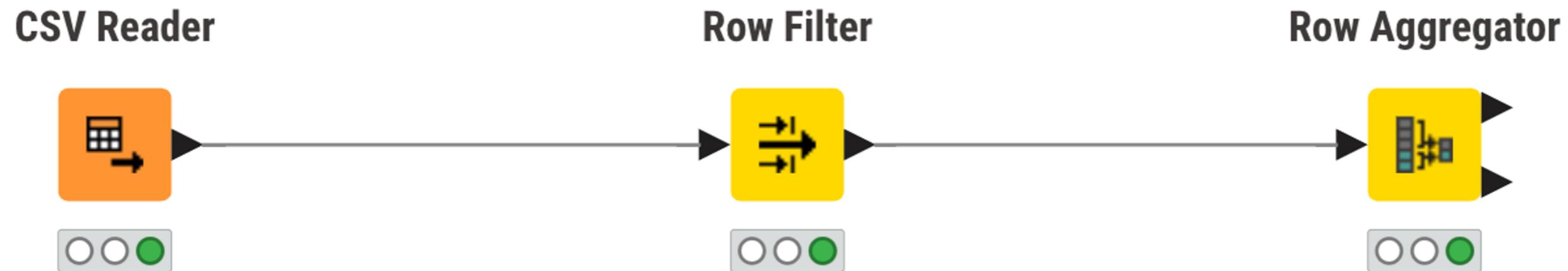
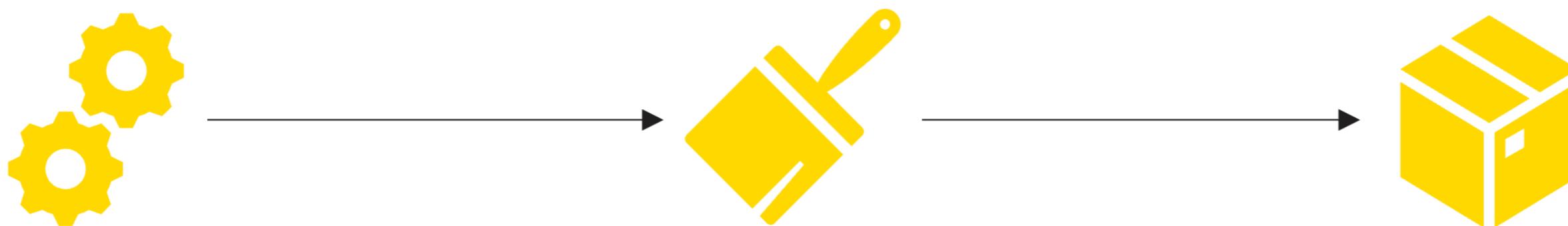
Production line



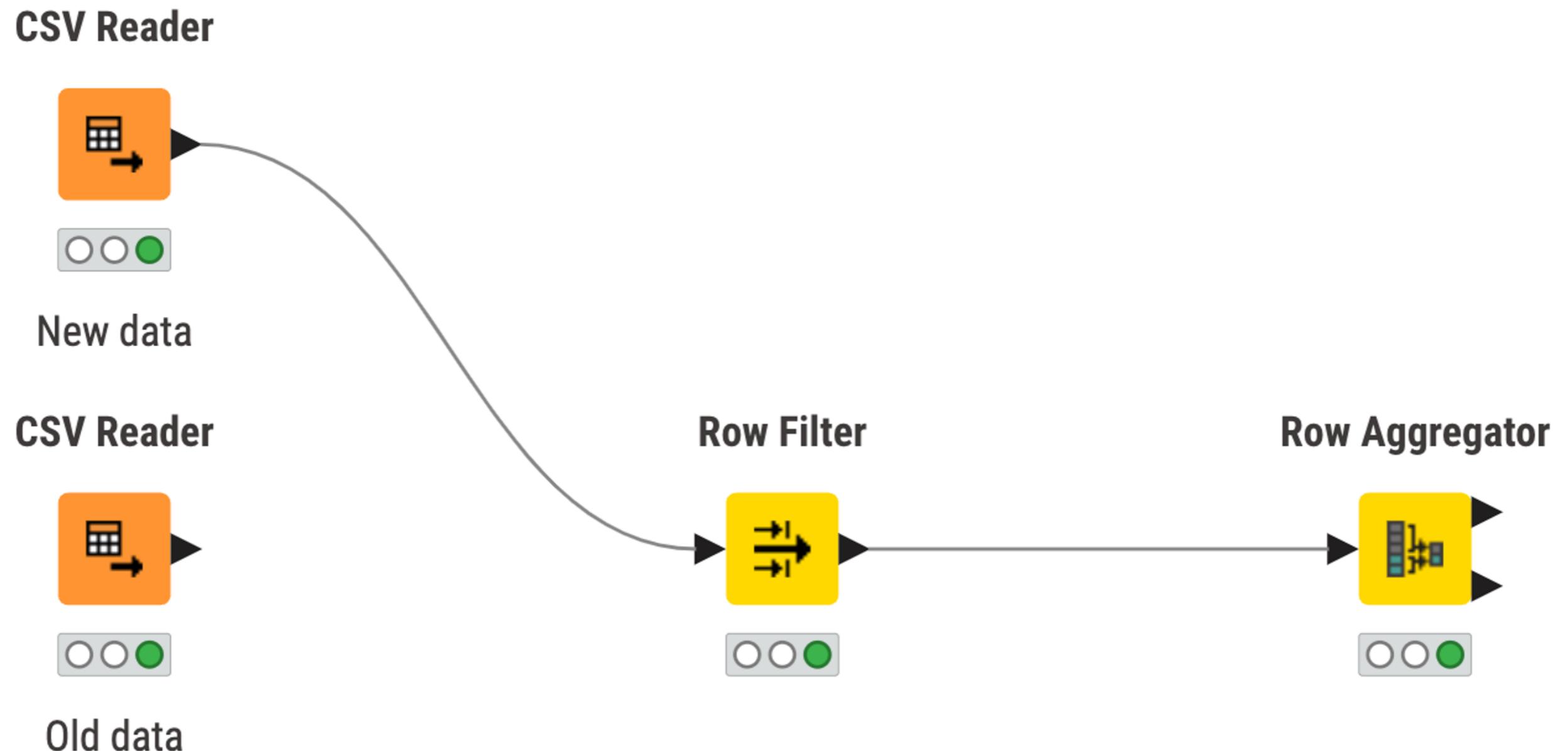
Production line



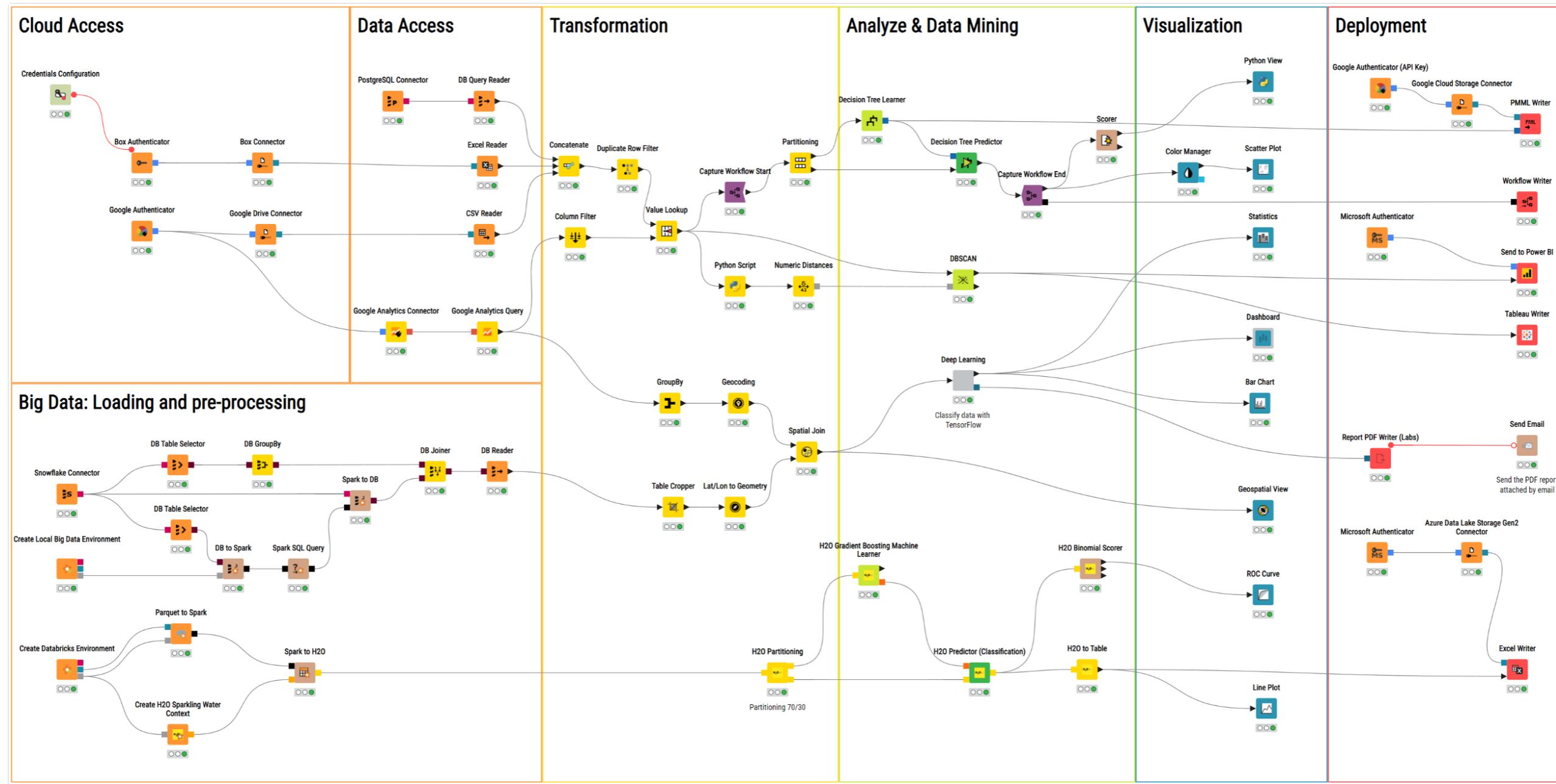
Production line



Data process



From data access to deployment



Let's practice!

INTRODUCTION TO KNIME

The user interface of **KNIME** Analytics Platform

INTRODUCTION TO KNIME



Let's practice!

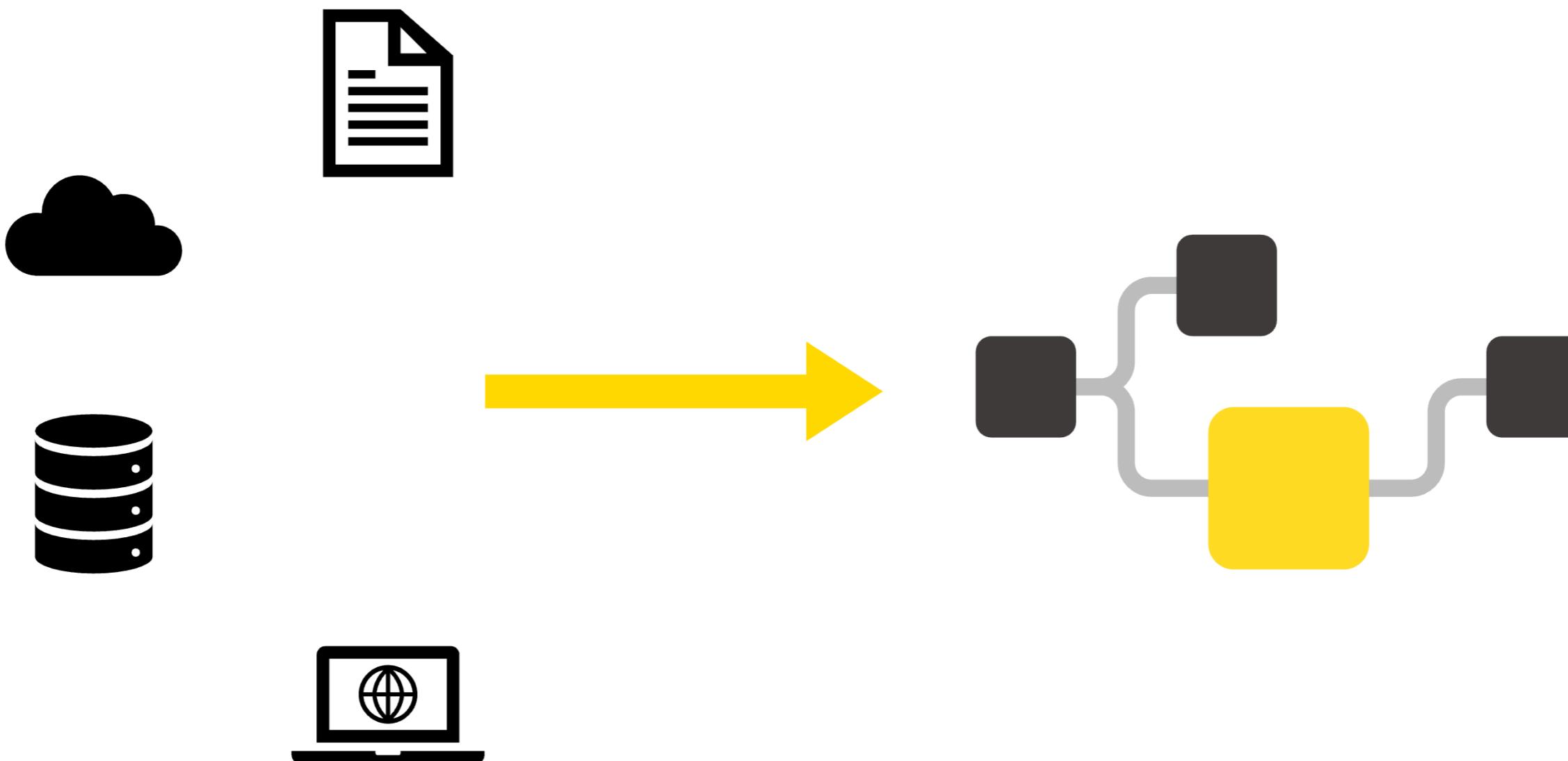
INTRODUCTION TO KNIME

Data access

INTRODUCTION TO KNIME



Data access



Data access

Excel Reader



File Table (Table)

Rows: 74 | Columns: 15

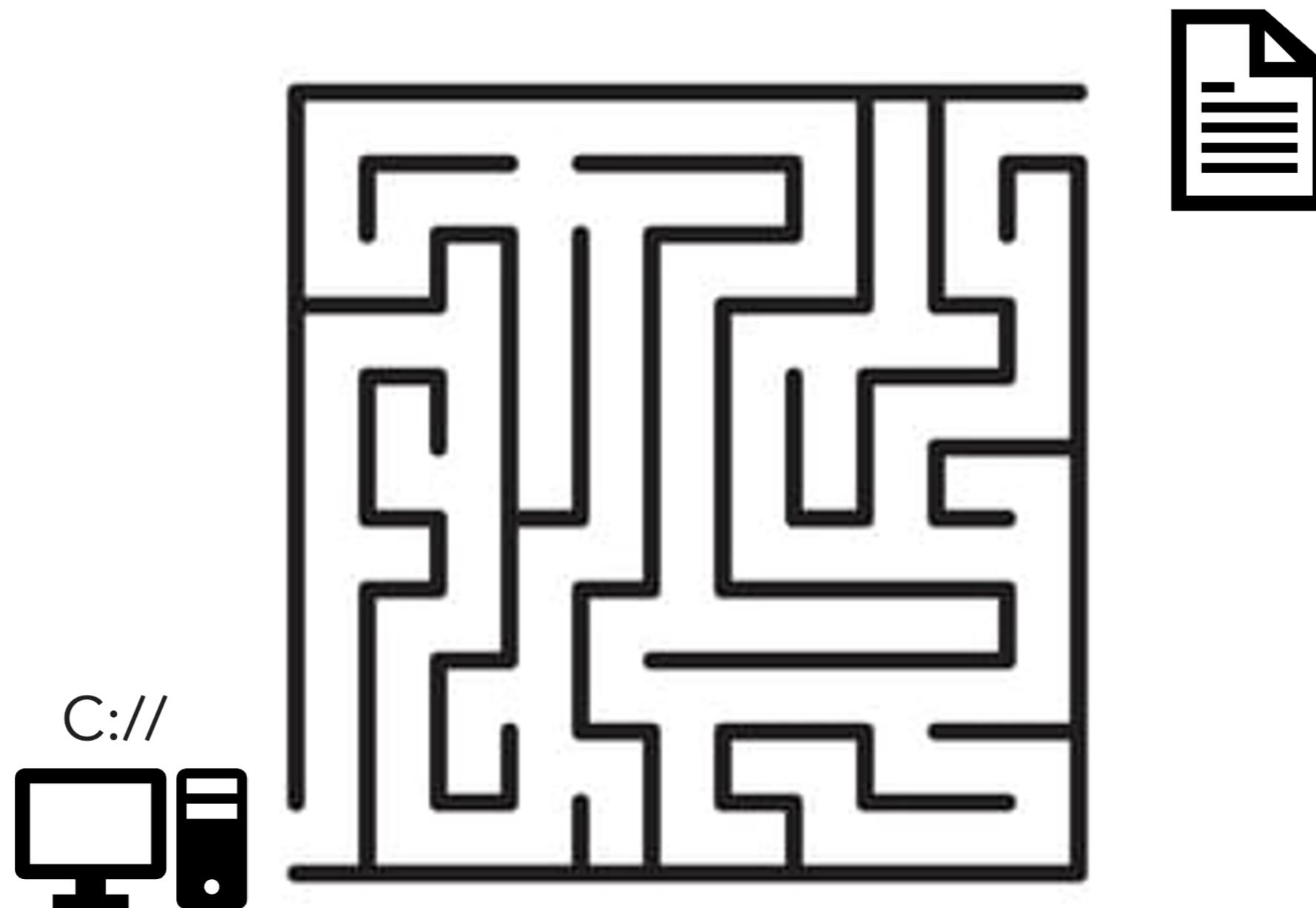
#	RowID	ID String	Age Number (integer)	Gender String	MaritalStatus String	Education Number (integer)	BusinessTravel String
1	Row0	ID0	36	Male	Single	3	Travel_Rarely
2	Row1	ID1	19	Male	Single	2	Travel_Rarely
3	Row2	ID2	26	Male	Single	3	Travel_Rarely
4	Row3	ID3	34	Female	Married	4	Travel_Frequent
5	Row4	ID4	48	Male	Single	4	Travel_Rarely
6	Row5	ID5	20	Male	Single	3	Travel_Rarely
7	Row6	ID6	33	Male	Divorced	3	Travel_Rarely
8	Row7	ID7	32	Male	Married	4	Travel_Rarely
9	Row8	ID8	29	Male	Married	1	Travel_Rarely
10	Row9	ID9	32	Female	Divorced	1	Non-Travel
11	Row...	ID10	35	Male	Married	4	Travel_Rarely
12	Row...	ID11	54	Female	Single	4	Travel_Frequent
13	Row...	ID12	45	Male	Married	2	Non-Travel
14	Row...	ID13	33	Male	Divorced	2	Non-Travel
15	Row...	ID14	34	Female	Divorced	3	Travel_Frequent
16	Row...	ID15	35	Male	Married	4	Travel_Rarely
17	Row...	ID16	25	Male	Married	3	Travel_Rarely
18	Row...	ID17	22	Female	Single	2	Travel_Daily

File types

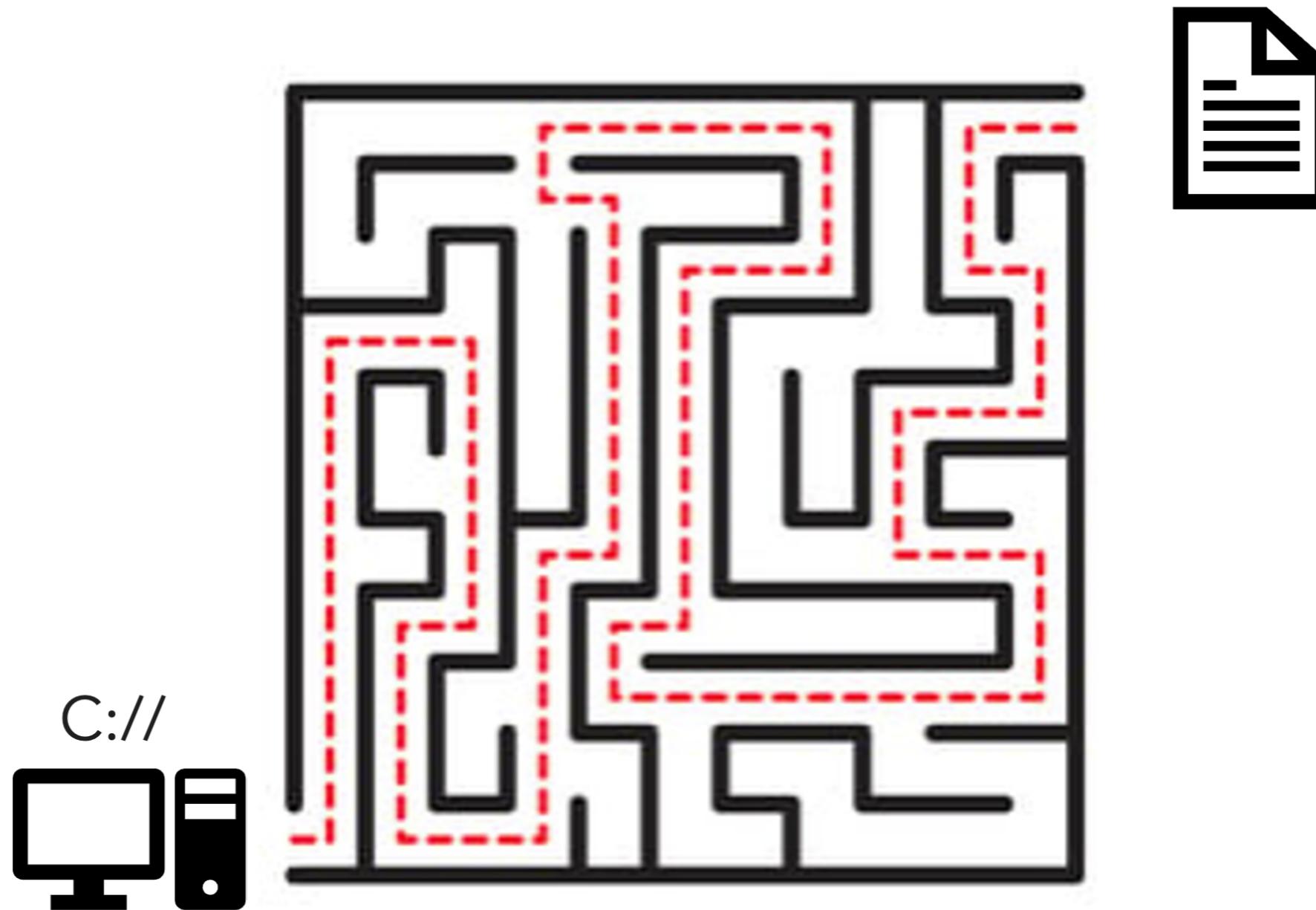
- .CSV
- .txt
- .xlsx
- .json
- .xml
- .table
- .png
- ...



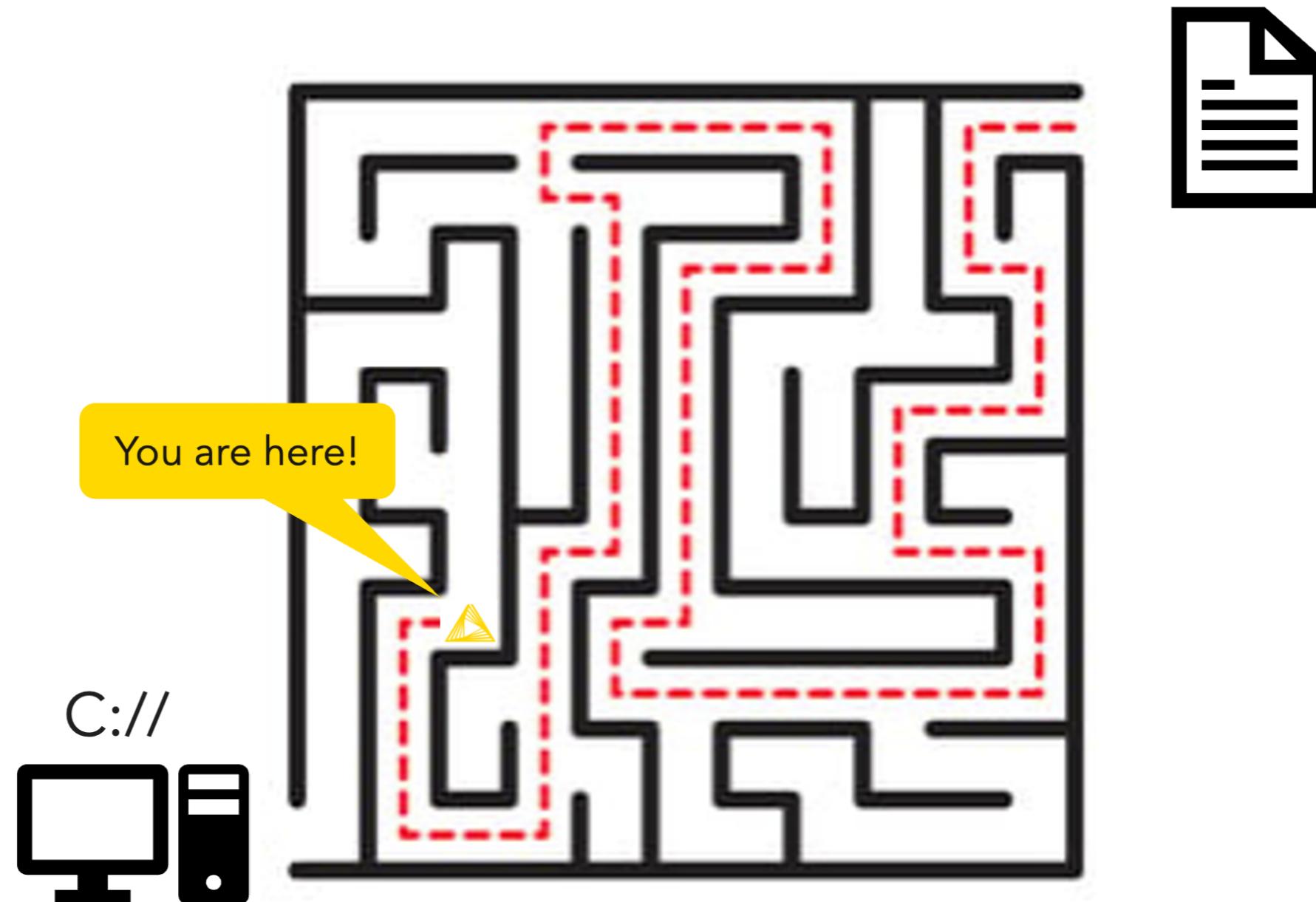
File path



Absolute file path



Relative file path



Connected file systems

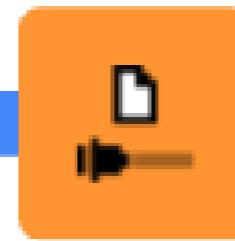


Connected file systems

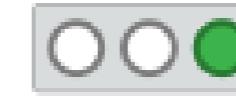
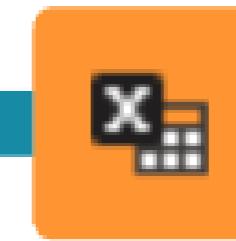
**Google
Authenticator**



**Google Drive
Connector**



**Excel
Reader**



Multiple files



Excel
Reader



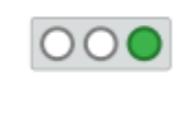
file 1

Excel
Reader



file 2

Excel
Reader



file 3

Excel
Reader



file 4

Excel
Reader



file 5

Excel
Reader



file 6

Excel
Reader



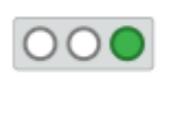
file 1

Excel
Reader



file 2

Excel
Reader



file 3

Excel
Reader



file 4

Excel
Reader



file 5

Excel
Reader

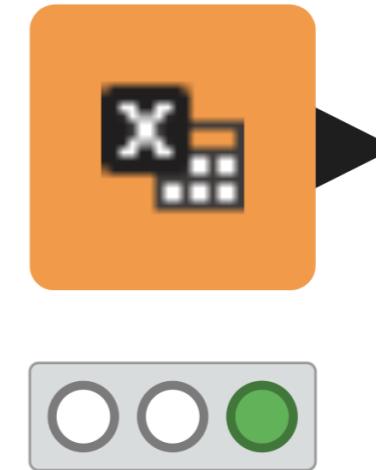


file 6

Reading folders

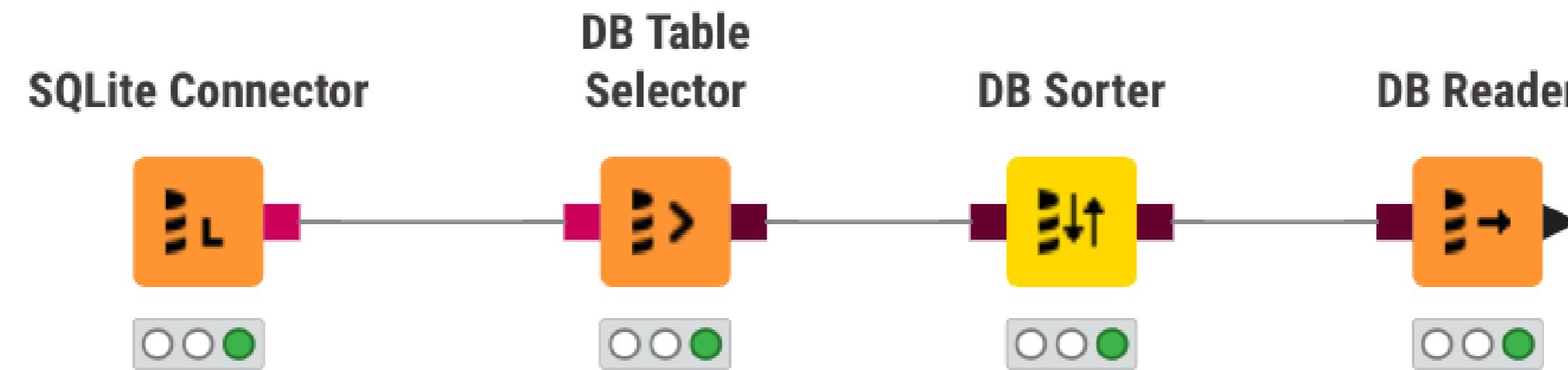


Excel Reader



Read all files in
employee folder

Accessing Databases



Let's practice!

INTRODUCTION TO KNIME

Accessing data in KNIME Analytics Platform

INTRODUCTION TO KNIME



Let's practice!

INTRODUCTION TO KNIME

Data cleaning

INTRODUCTION TO KNIME



Data cleaning



Filter columns

#	Name	Department	Salary	Favorite food
1	Jenny	Sales	?	Pizza
2	Alex	Finance	22000	Sushi
3	Taylor	R&D	28000	Tacos
4	Alex	Finance	22000	Sushi
5	Sam	R&D	23000	Salad

Filter columns

Filter out

#	Name	Department	Salary	Favorite food
1	Jenny	Sales	?	Pizza
2	Alex	Finance	22000	Sushi
3	Taylor	R&D	28000	Tacos
4	Alex	Finance	22000	Sushi
5	Sam	R&D	23000	Salad

Filter rows

#	Name	Department	Salary	Favorite food
1	Jenny	Sales	?	Pizza
2	Alex	Finance	22000	Sushi
3	Taylor	R&D	28000	Tacos
4	Alex	Finance	22000	Sushi
5	Sam	R&D	23000	Salad

Employees
from R&D

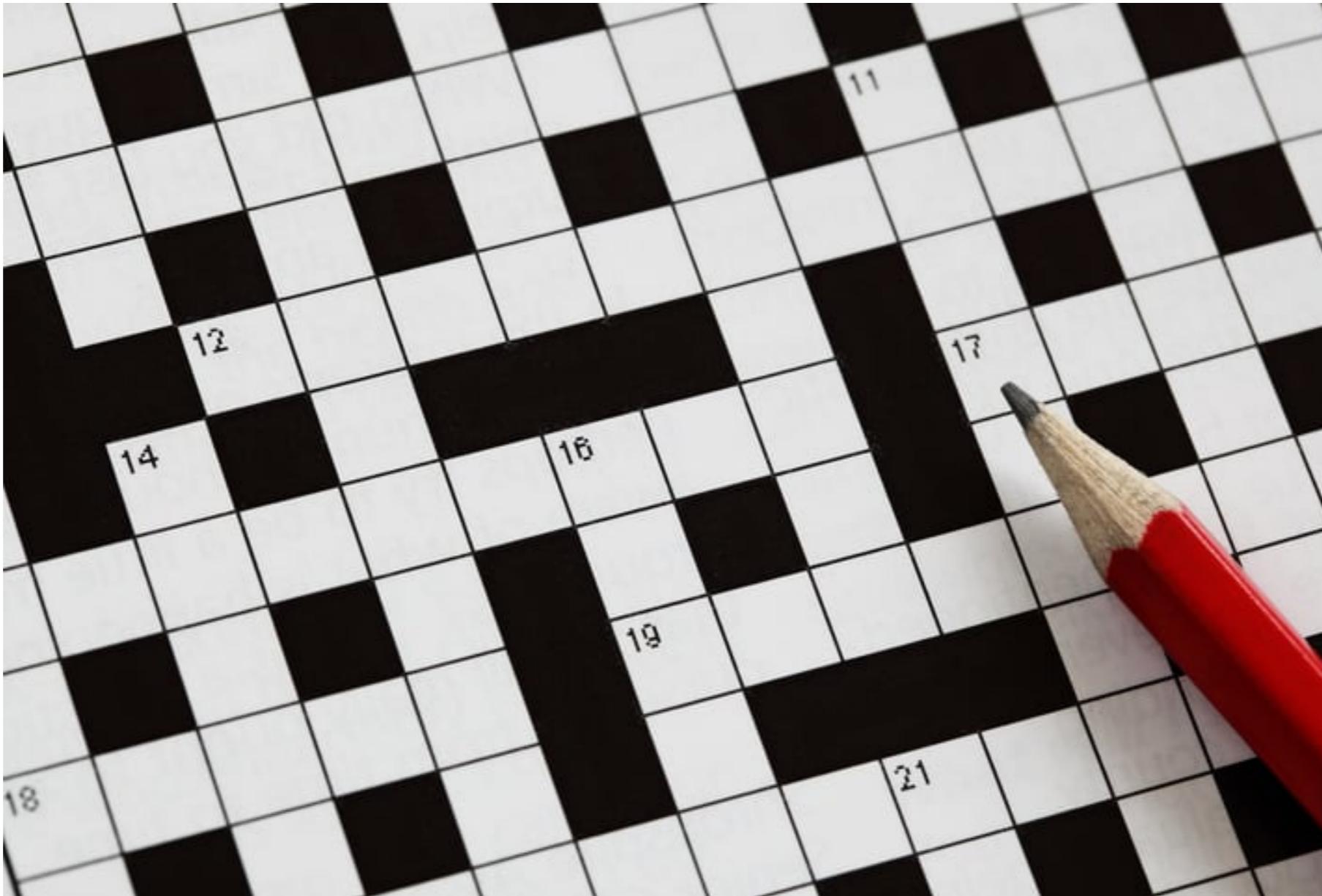
Filter duplicates

#	Name	Department	Salary	Favorite food	
1	Jenny	Sales	?	Pizza	Duplicate row
2	Alex	Finance	22000	Sushi	
3	Taylor	R&D	28000	Tacos	
4	Alex	Finance	22000	Sushi	
5	Sam	R&D	23000	Salad	

Handle missing values

#	Name	Department	Salary	Favorite food
1	Jenny	Sales	?	Pizza
2	Alex	Finance	22000	<div data-bbox="1960 736 2549 864">Missing value</div> Sushi
3	Taylor	R&D	28000	Tacos
4	Alex	Finance	22000	Sushi
5	Sam	R&D	23000	Salad

Handle missing values



Data types

#	Name	Salary	Birthday	Remote
1	Jenny	25000	12/12/1990	TRUE
2	Alex	22000	04/03/1998	FALSE
3	Taylor	28000	11/02/1987	FALSE
4	Alex	22000	04/03/1998	FALSE
5	Sam	23000	02/07/1989	TRUE

Data types

String

#	Name	Salary	Birthday	Remote
1	Jenny	25000	12/12/1990	TRUE
2	Alex	22000	04/03/1998	FALSE
3	Taylor	28000	11/02/1987	FALSE
4	Alex	22000	04/03/1998	FALSE
5	Sam	23000	02/07/1989	TRUE

Data types

```
graph TD; String --> Name; String --> Salary; Number --> Salary;
```

#	Name	Salary	Birthday	Remote
1	Jenny	25000	12/12/1990	TRUE
2	Alex	22000	04/03/1998	FALSE
3	Taylor	28000	11/02/1987	FALSE
4	Alex	22000	04/03/1998	FALSE
5	Sam	23000	02/07/1989	TRUE

Data types

The diagram illustrates the mapping of data types to columns in a table. At the top, three yellow rounded rectangles represent the data types: 'String' (left), 'Number' (middle), and 'Date&Time' (right). Below them, a table has its first row colored yellow, serving as the header. The header row contains five columns labeled '#', 'Name', 'Salary', 'Birthday', and 'Remote'. Arrows point from each data type to its corresponding column in the table: 'String' points to 'Name', 'Number' points to 'Salary', and 'Date&Time' points to 'Birthday'. The table body consists of five rows, each containing data for these five columns.

#	Name	Salary	Birthday	Remote
1	Jenny	25000	12/12/1990	TRUE
2	Alex	22000	04/03/1998	FALSE
3	Taylor	28000	11/02/1987	FALSE
4	Alex	22000	04/03/1998	FALSE
5	Sam	23000	02/07/1989	TRUE

Data types

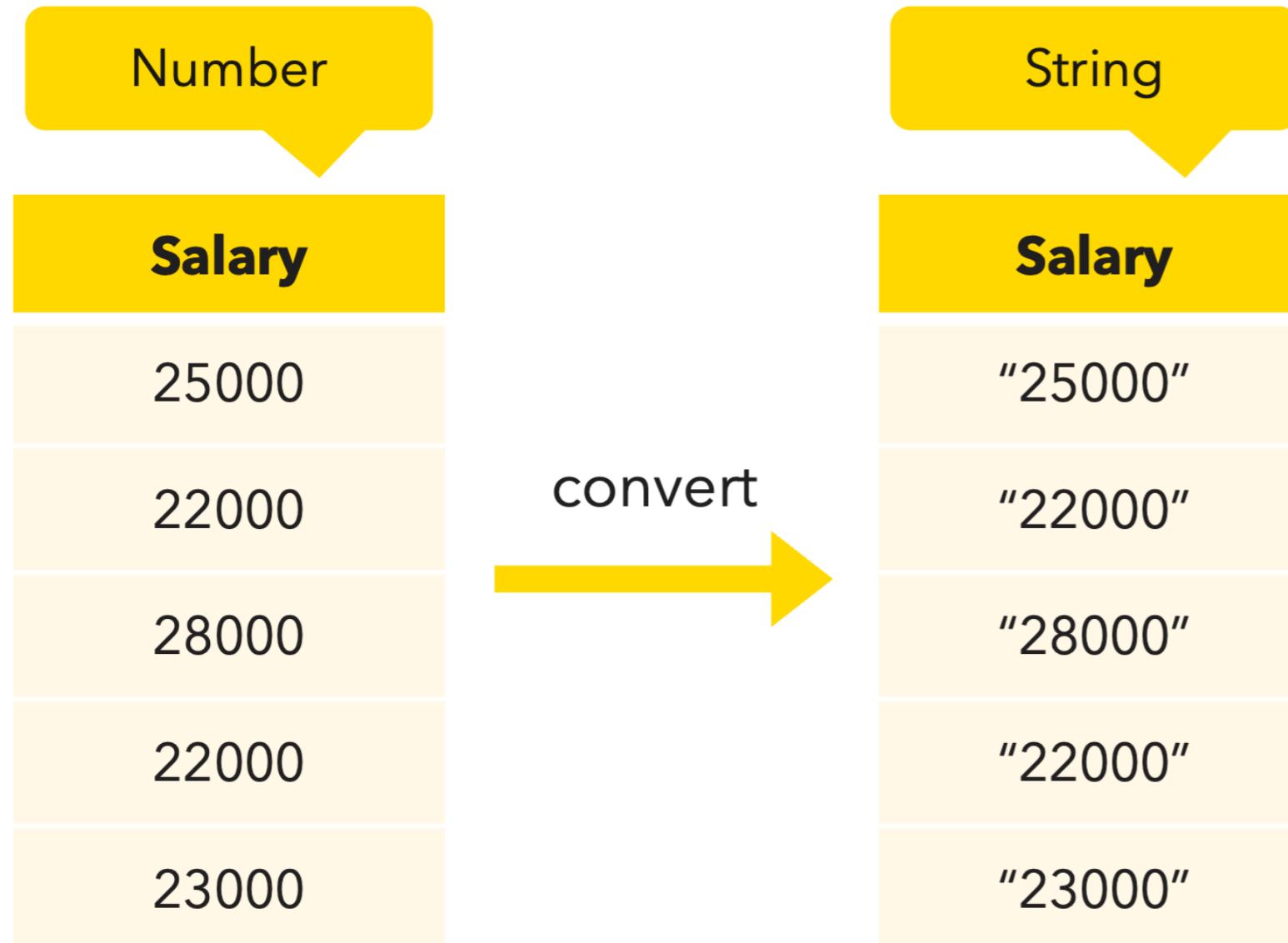
The diagram illustrates the mapping of data types to columns in a table. At the top, four yellow rounded rectangles represent the data types: String, Number, Date&Time, and Boolean. Below them, a table has columns labeled '#', Name, Salary, Birthday, and Remote. Arrows point from each data type to its corresponding column in the table.

	String	Number	Date&Time	Boolean
#	Name	Salary	Birthday	Remote
1	Jenny	25000	12/12/1990	TRUE
2	Alex	22000	04/03/1998	FALSE
3	Taylor	28000	11/02/1987	FALSE
4	Alex	22000	04/03/1998	FALSE
5	Sam	23000	02/07/1989	TRUE

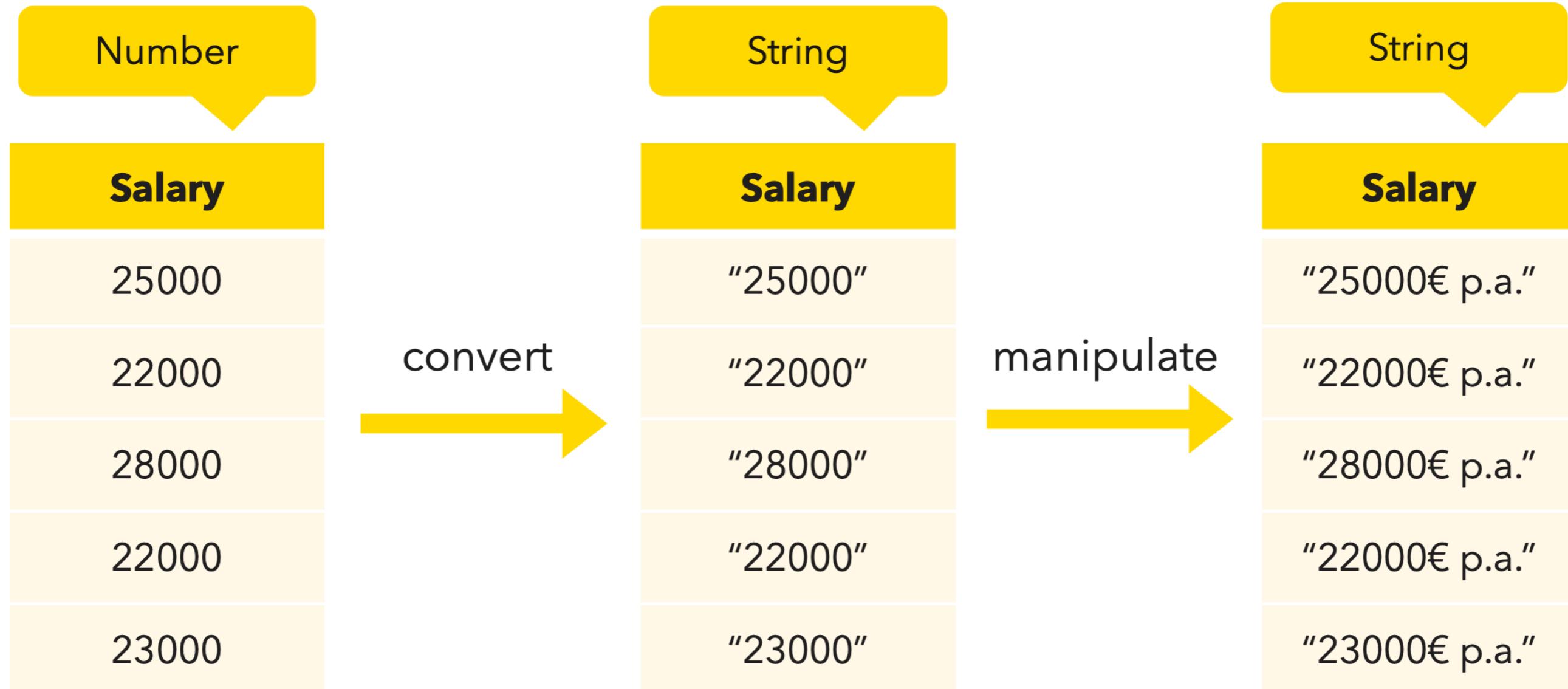
Convert Data Types

Number
Salary
25000
22000
28000
22000
23000

Convert Data Types



Convert Data Types



Clean strings

Email

jenny.brówñ @EXAMPLE.COM

alex.léé @EXAMPLE.COM

taylor.tóñs @EXAMPLE.COM

alex.léé @EXAMPLE.COM

sóphia.kim @EXAMPLE.COM

Clean strings



Let's practice!

INTRODUCTION TO KNIME

Cleaning data in KNIME Analytics Platform

INTRODUCTION TO KNIME



Let's practice!

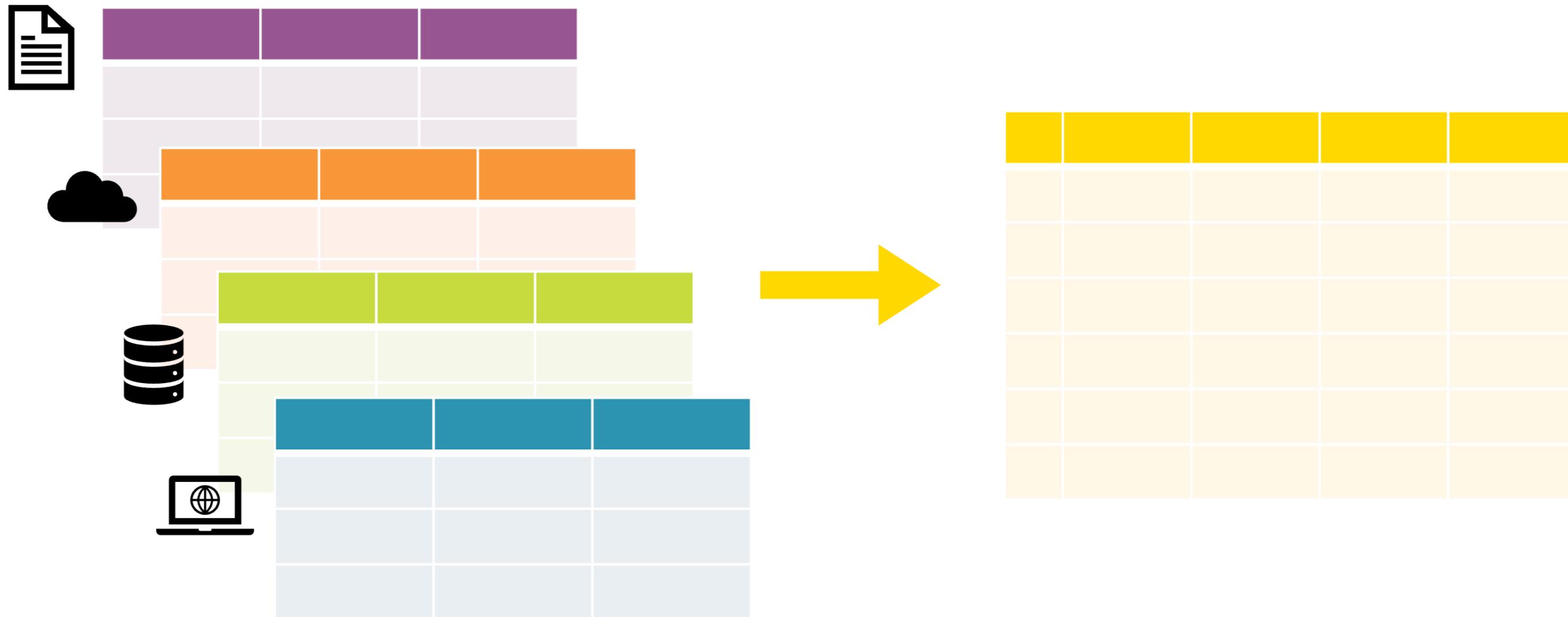
INTRODUCTION TO KNIME

Mix and squeeze data

INTRODUCTION TO KNIME



Data merging



Value lookup

#	EmployeeID	Name	Department	Favorite food
1	EMP-43	Jenny	Sales	Pizza
2	EMP-29	Alex	Finance	Sushi
3	EMP-11	Taylor	R&D	Tacos
4	EMP-29	Alex	Finance	Sushi
5	EMP-66	Sam	R&D	Salad

Value lookup

#	EmployeeID	Name	Department	Favorite food
1	EMP-43	Jenny	Sales	Pizza
2	EMP-29	Alex	Finance	Sushi
3	EMP-11	Taylor	R&D	Tacos
4	EMP-29	Alex	Finance	Sushi
5	EMP-66	Sam	R&D	Salad

#	ID	Salary
1	EMP-11	25000
2	EMP-29	22000
3	EMP-43	28000
4	EMP-88	22000

Value lookup

#	EmployeeID	Name	Department	Favorite food
1	EMP-43	Jenny	Sales	Pizza
2	EMP-29	Alex	Finance	Sushi
3	EMP-11	Taylor	R&D	Tacos
4	EMP-29	Alex	Finance	Sushi
5	EMP-66	Sam	R&D	Salad

#	ID	Salary
1	EMP-11	25000
2	EMP-29	22000
3	EMP-43	28000
4	EMP-88	22000

Value lookup

#	EmployeeID	Name	Department	Favorite food	Salary
1	EMP-43	Jenny	Sales	Pizza	
2	EMP-29	Alex	Finance	Sushi	
3	EMP-11	Taylor	R&D	Tacos	
4	EMP-29	Alex	Finance	Sushi	
5	EMP-66	Sam	R&D	Salad	

#	ID	Salary
1	EMP-11	25000
2	EMP-29	22000
3	EMP-43	28000
4	EMP-88	22000

Value lookup

#	EmployeeID	Name	Department	Favorite food	Salary
1	EMP-43	Jenny	Sales	Pizza	
2	EMP-29	Alex	Finance	Sushi	
3	EMP-11	Taylor	R&D	Tacos	
4	EMP-29	Alex	Finance	Sushi	
5	EMP-66	Sam	R&D	Salad	

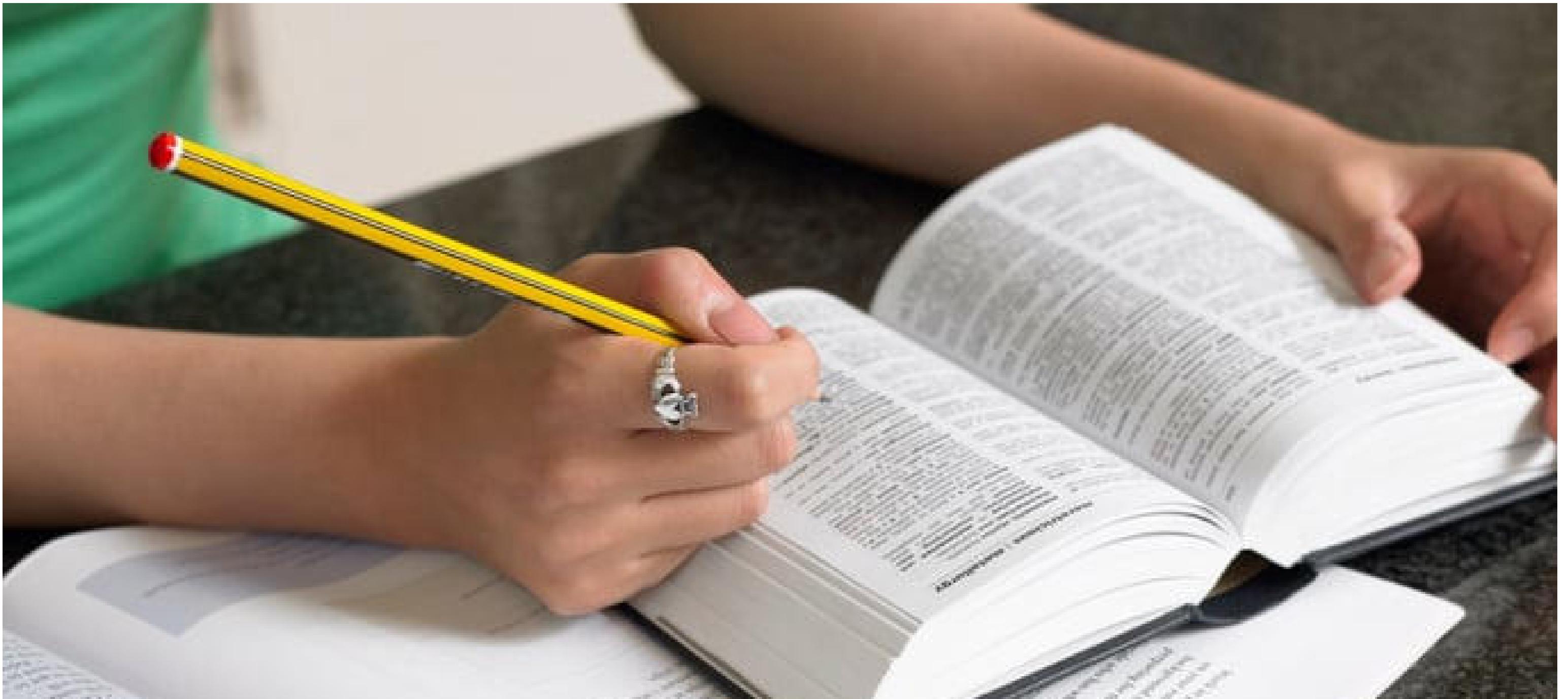
#	ID	Salary
1	EMP-11	25000
2	EMP-29	22000
3	EMP-43	28000
4	EMP-88	22000

Value lookup

#	EmployeeID	Name	Department	Favorite food	Salary
1	EMP-43	Jenny	Sales	Pizza	28000
2	EMP-29	Alex	Finance	Sushi	
3	EMP-11	Taylor	R&D	Tacos	
4	EMP-29	Alex	Finance	Sushi	
5	EMP-66	Sam	R&D	Salad	

#	ID	Salary
1	EMP-11	25000
2	EMP-29	22000
3	EMP-43	28000
4	EMP-88	22000

Value lookup



Value lookup

#	EmployeeID	Name	Department	Favorite food	Salary
1	EMP-43	Jenny	Sales	Pizza	28000
2	EMP-29	Alex	Finance	Sushi	22000
3	EMP-11	Taylor	R&D	Tacos	25000
4	EMP-29	Alex	Finance	Sushi	22000
5	EMP-66	Sam	R&D	Salad	

#	ID	Salary	Dictionary
1	EMP-11	25000	
2	EMP-29	22000	
3	EMP-43	28000	
4	EMP-88	22000	

Value lookup

#	EmployeeID	Name	Department	Favorite food	Salary
1	EMP-43	Jenny	Sales	Pizza	28000
2	EMP-29	Alex	Finance	Sushi	22000
3	EMP-11	Taylor	R&D	Tacos	25000
4	EMP-29	Alex	Finance	Sushi	22000
5	EMP-66	Sam	R&D	Salad	

Diagram illustrating Value lookup:

The diagram illustrates a value lookup operation. It consists of two tables: a main table and a dictionary table.

Main Table:

#	ID	Salary
1	EMP-11	25000
2	EMP-29	22000
3	EMP-43	28000
4	EMP-88	22000

Dictionary Table:

#	EmployeeID	Name	Department	Favorite food	Salary
1	EMP-43	Jenny	Sales	Pizza	28000
2	EMP-29	Alex	Finance	Sushi	22000
3	EMP-11	Taylor	R&D	Tacos	25000
4	EMP-29	Alex	Finance	Sushi	22000
5	EMP-66	Sam	R&D	Salad	

Value lookup

#	EmployeeID	Name	Department	Favorite food	Salary
1	EMP-43	Jenny	Sales	Pizza	28000
2	EMP-29	Alex	Finance	Sushi	22000
3	EMP-11	Taylor	R&D	Tacos	25000
4	EMP-29	Alex	Finance	Sushi	22000
5	EMP-66	Sam	R&D	Salad	?

Diagram illustrating Value lookup:

- Lookup column:** Points to the **EmployeeID** column in the main table.
- Key column:** Points to the **ID** column in the dictionary table.
- Dictionary:** Points to the secondary table containing a mapping between Employee IDs and their corresponding salaries.

#	ID	Salary
1	EMP-11	25000
2	EMP-29	22000
3	EMP-43	28000
4	EMP-88	22000

Data aggregation



Data aggregation



Data aggregation

#	Name	Department	Salary
1	Jenny	Sales	28000
2	Alex	Finance	22000
3	Taylor	R&D	25000
4	Andrea	Finance	22000
5	Sam	R&D	27000

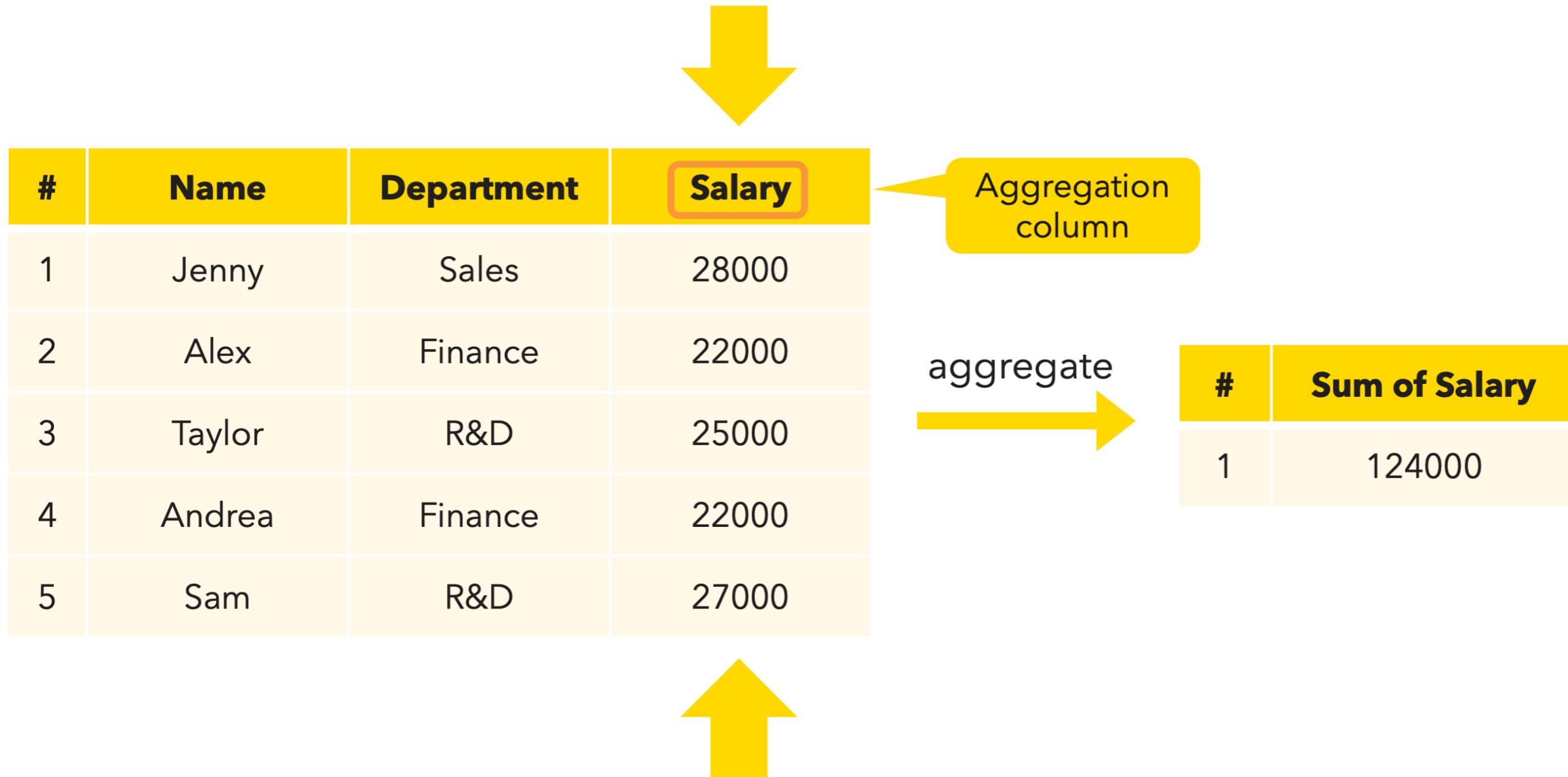


aggregate
→

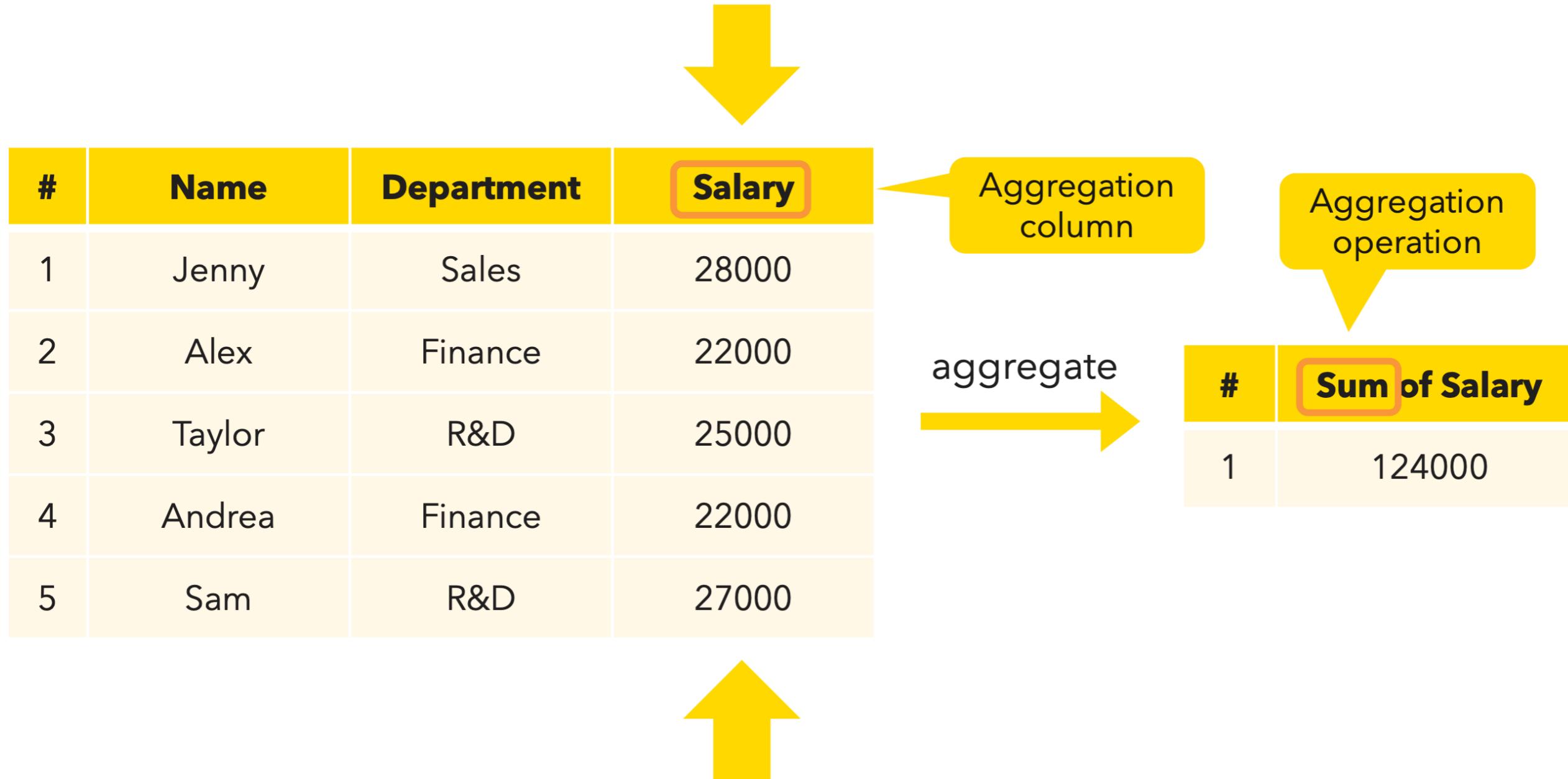
#	Sum of Salary
1	124000



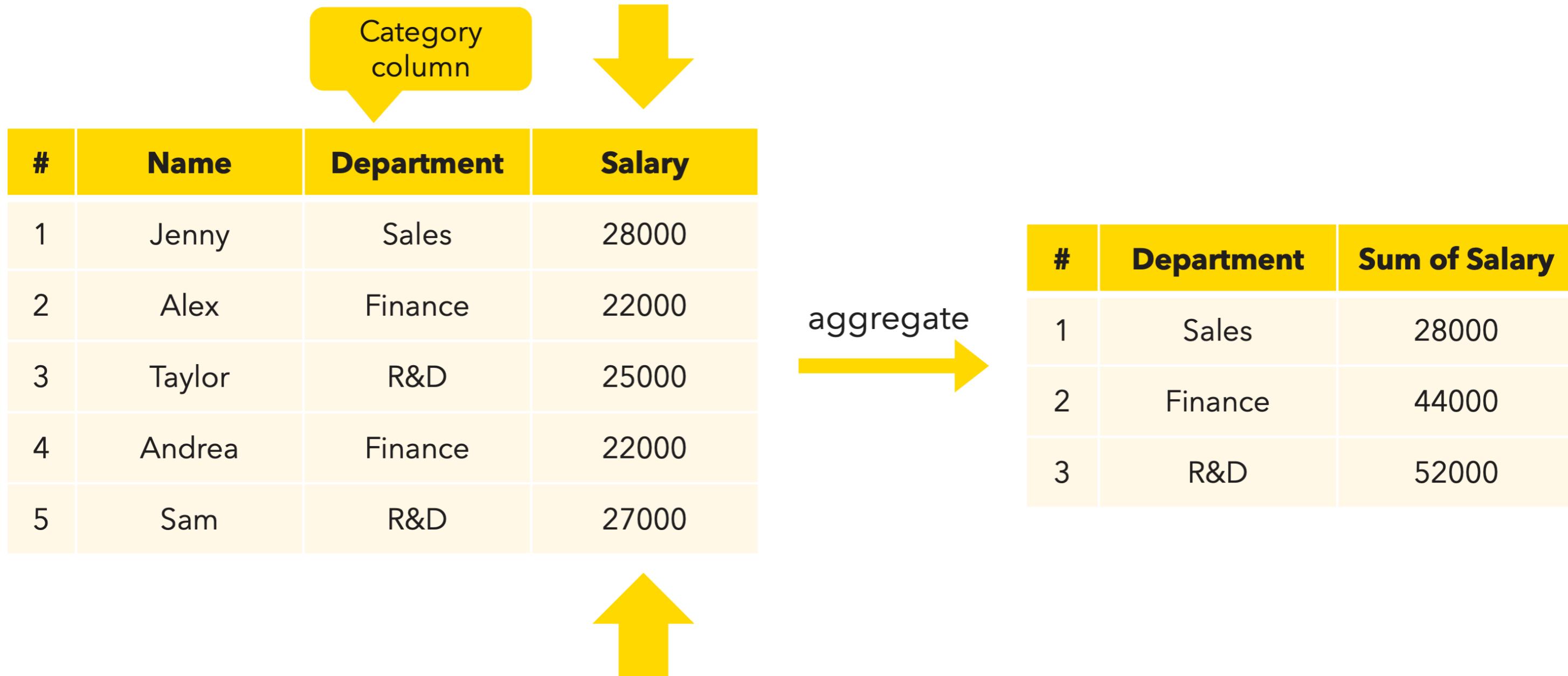
Data aggregation



Data aggregation



Data aggregation



Data aggregation

#	Name	Department	Salary
1	Jenny	Sales	28000
2	Alex	Finance	22000
3	Taylor	R&D	25000
4	Andrea	Finance	22000
5	Sam	R&D	27000

aggregate
→

#	Department	Number of employees
1	Sales	1
2	Finance	2
3	R&D	2

Data aggregation

#	Name	Department	Salary
1	Jenny	Sales	28000
2	Alex	Finance	22000
3	Taylor	R&D	25000
4	Andrea	Finance	22000
5	Sam	R&D	27000

aggregate
→

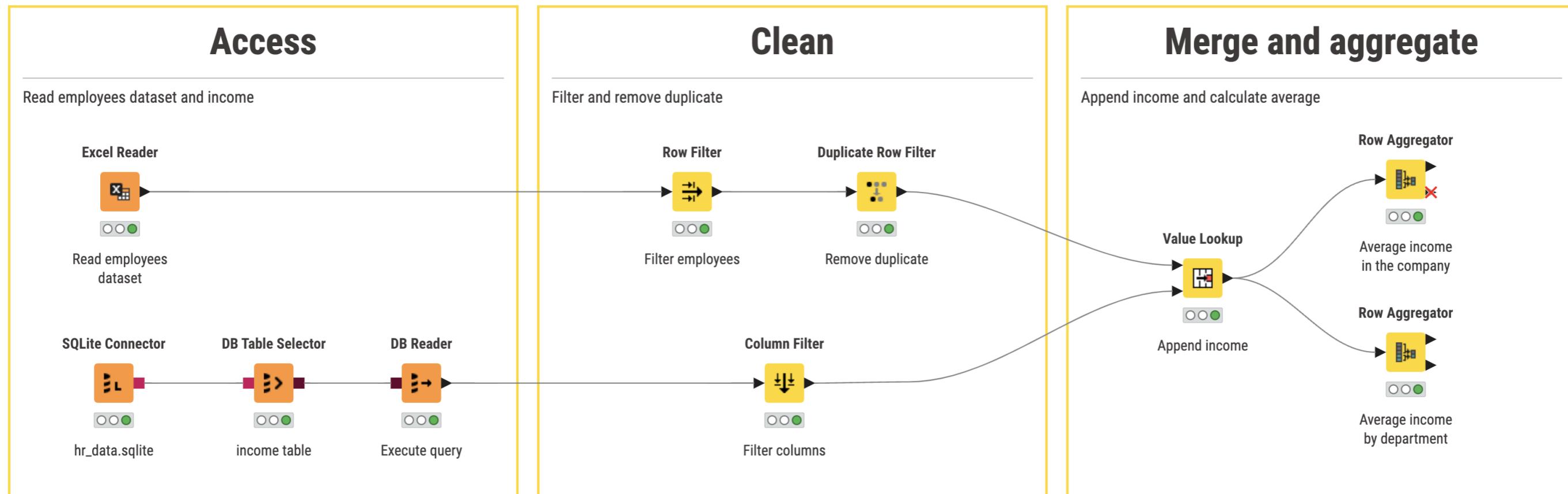
#	Department	Average Salary
1	Sales	28000
2	Finance	22000
3	R&D	26000

Documentation

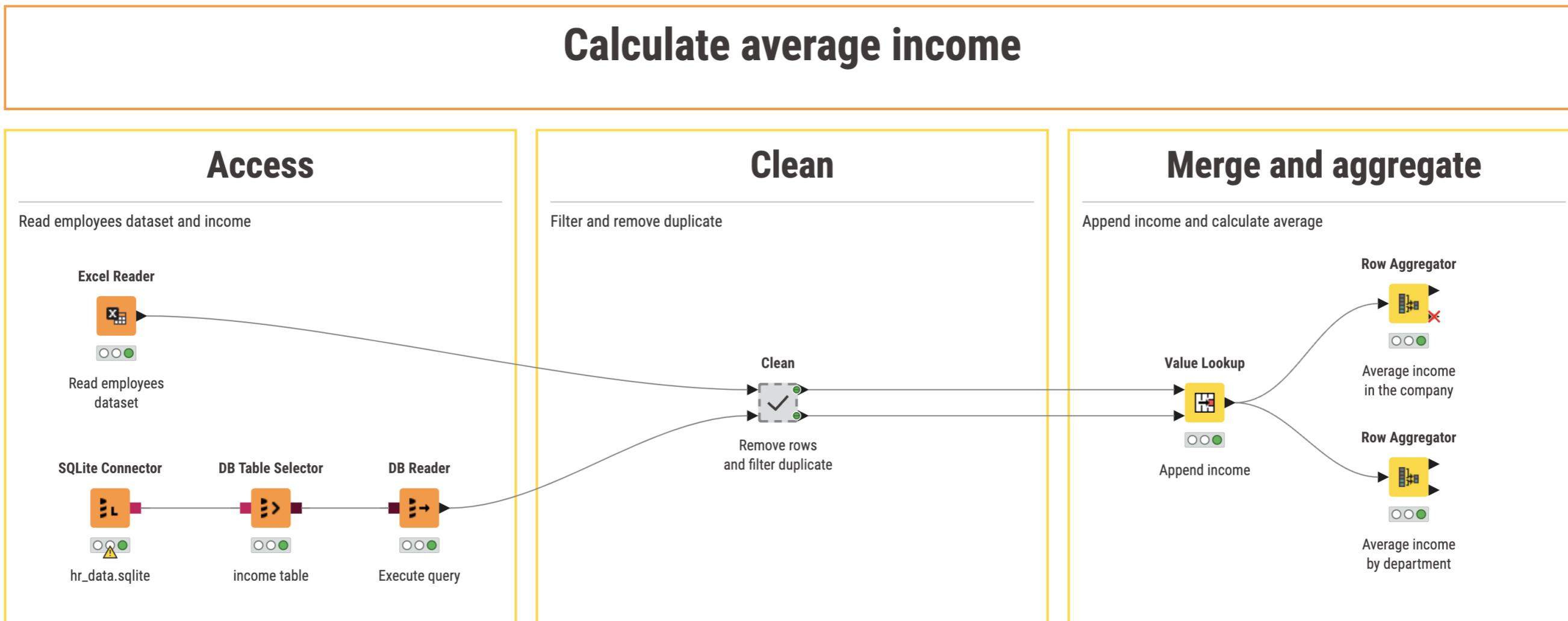


Documentation

Calculate average income



Metanodes



Let's practice!

INTRODUCTION TO KNIME

Crunching data in KNIME Analytics Platform

INTRODUCTION TO KNIME



Let's practice!

INTRODUCTION TO KNIME

Congratulations!

INTRODUCTION TO KNIME



Recap



KNIME Analytics Platform

The screenshot shows the KNIME Analytics Platform interface. The top navigation bar includes 'Home', 'Combine Clean and Summarize Spread...', 'Execute', 'Cancel', 'Reset', 'Create metanode', and 'Create component'. The main area is titled 'Combine Clean and Summarize Spreadsheet Data' with a sub-instruction: 'This workflow shows how to concatenate data from multiple Excel sheets and enrich the data with the value lookup function. At the end, it visualizes and aggregates the combined and cleaned data.' Below this is a 'Task' section: 'The use case is calculating the total volume of the furniture when moving to a new apartment.'

The workflow diagram illustrates the process:

- Read three sheets from the same Excel file containing the following information:**
 - Furniture in the living room
 - Furniture in the kitchen
 - The estimated volumes of the furniture
- Clean up the data by:**
 - removing unnecessary columns
 - shifting data that was entered into a wrong column
 - changing the data type of a column
- Combine data by:**
 - concatenating rows in similar tables
 - appending values from a dictionary
- Calculate the following results:**
 - The volume by furniture type
 - The total volume of all furniture

The workflow consists of several nodes connected by arrows:

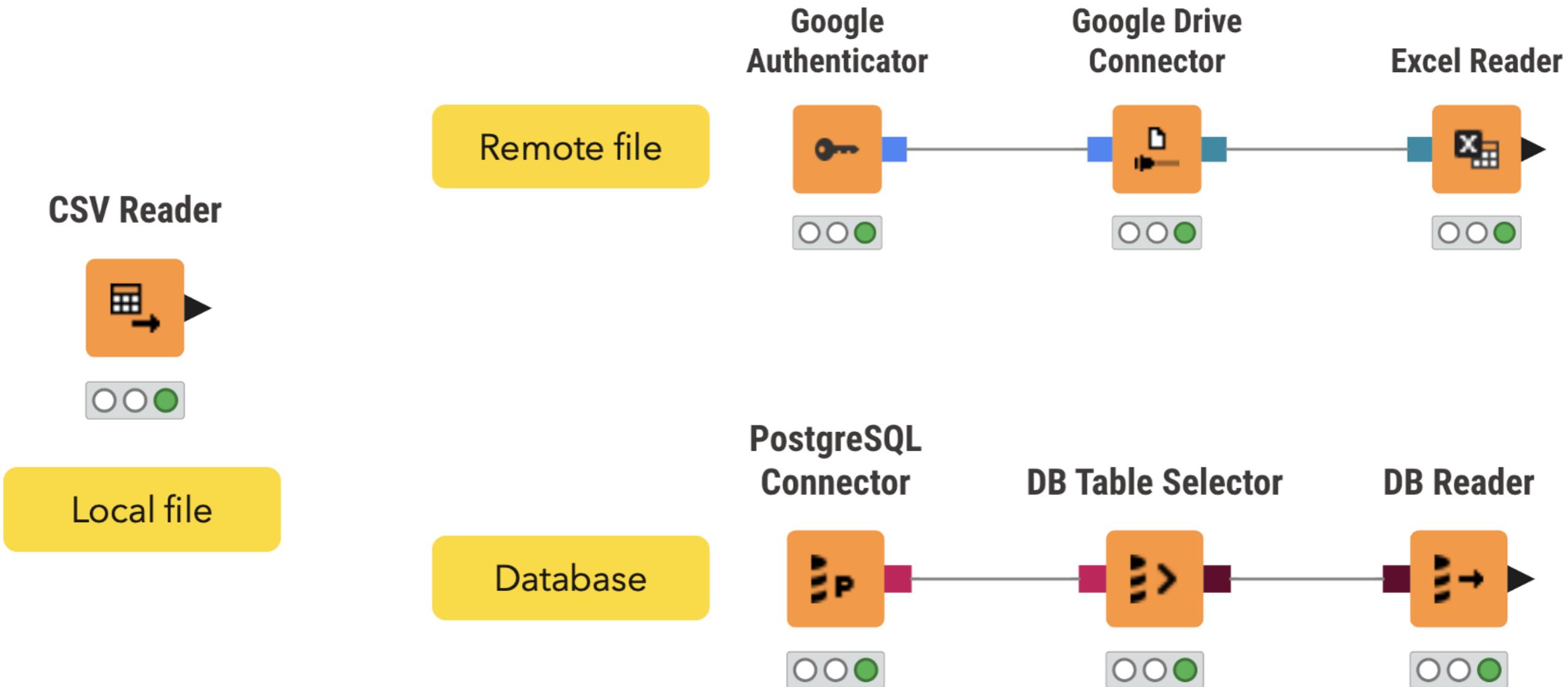
- Three **Excel Reader** nodes read the 'Kitchen', 'Living Room', and 'Dictionary' sheets of a 'rooms.xlsx' file.
- A **Column Filter** node filters the 'Living Room' sheet.
- A **Column Merger** node merges the 'Kitchen' and 'Living Room' data.
- A **String to Number** node converts the 'Dict-Volume' column to a number.
- A **Value Lookup** node appends volumes from the 'Dictionary' sheet based on item IDs.
- A **Concatenate** node combines the data.
- A **Row Aggregator** node calculates the total volume.
- A **Bar Chart** node visualizes the data.

At the bottom, a data table view shows the resulting data:

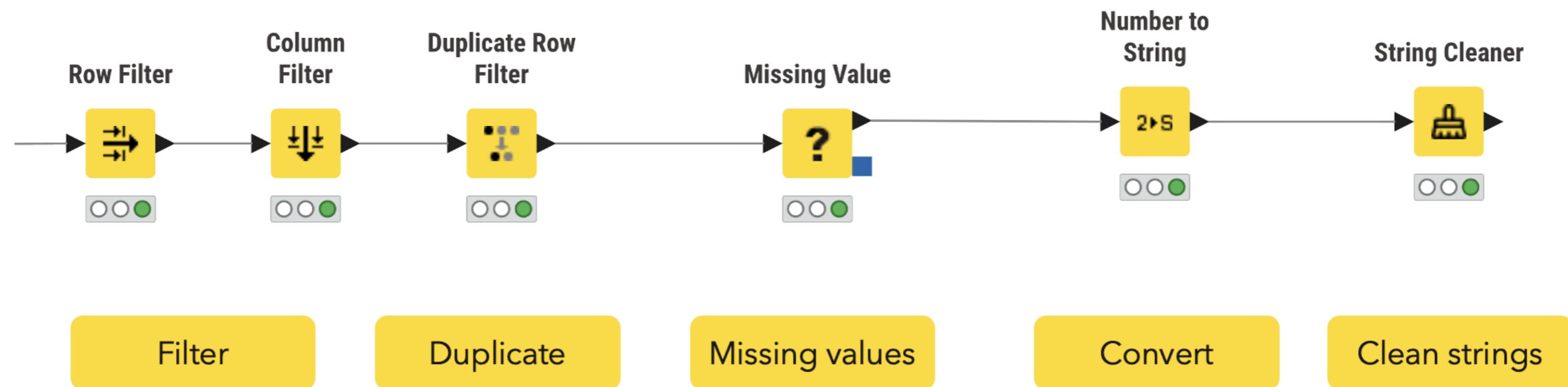
#	RowID	Item	Amount	Dict-Volume
1	Row0	Table	1	0.6
2	Row1	Chair	2	0.2
3	Row2	Cupboard	1	1
4	Row3	Side table	1	1
5	Row4	Fridge	1	0.5
6	Row5	Freezer	1	1
7	Row6	Oven	1	0.2
8	Row7	Microwave	1	0.2

INTRODUCTION TO KNIME

Data access



Data cleaning



Data merging and aggregation



More to come



Congratulations!

INTRODUCTION TO KNIME

Conditional statements and logical operators

DATA TRANSFORMATION IN KNIME



Conditional statements

- Allow you to assign values based on a condition
- If the condition is not met, assigns default value
- Example: IF Statement

IF statement

- Allows specifying one or more conditions
- Value 1 is assigned if condition one is met
- Assigns default value if all conditions fail

Syntax

```
IF(condition_1, value_1, additional_conditions, value_if_all_false)
```

Comparison operators

Operators are used to compare column values to

- Values of other columns
- Constant values

Operator	Expression	Meaning
=	<code>\$Column\$ = x</code>	Equal to x
>	<code>\$Column\$ > x</code>	Greater than x
<	<code>\$Column\$ < x</code>	Less than x
<code>>=</code>	<code>\$Column\$ >= x</code>	Greater than or Equal to x
<code><=</code>	<code>\$Column\$ <= x</code>	Less than or Equal to x

Comparison operators

Add "Age group" column

- "Adult" when the age is greater than 18
- "Non-Adult" if the age is lower than 18

Name	Age
Mark	27
Bob	17
John	18

Syntax:

```
IF($Age$>18,"Adult","Non-Adult")
```

Output:

Name	Age	Age Group
Mark	27	Adult
Bob	17	Non-Adult
John	18	Non-Adult

Logical operators

- Combines two or more conditional statements
- Enables using complex expressions
- Outputs a boolean value

AND

Returns True if all the conditions are met

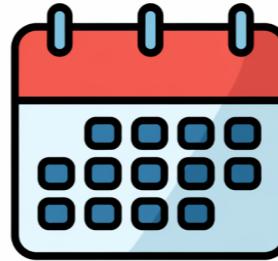
OR

Returns True if any of the conditions are met

NOT

Returns the negation of boolean expression

Logical operators



Assign the tag as a weekend

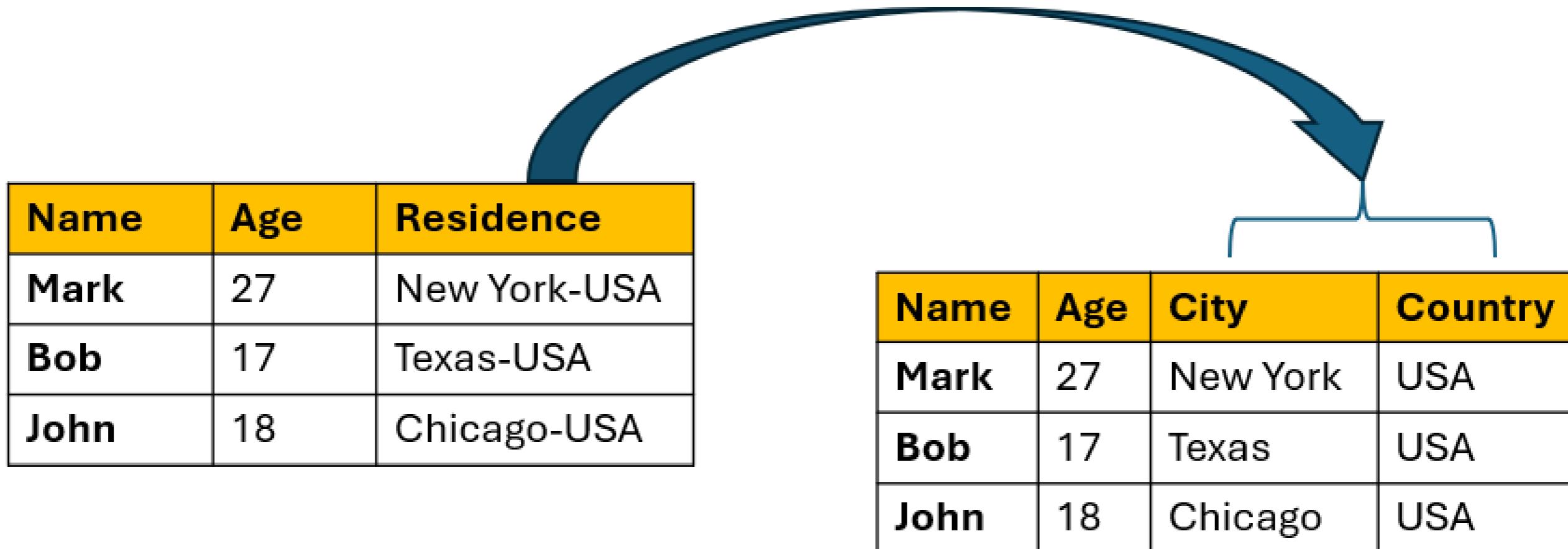
- **Condition 1:** If the day is "Sunday"
- **Condition 2:** If the day is "Saturday"

Syntax

```
IF($Day$="Saturday" OR $Day$="Sunday", "Weekend", "Weekday")
```

Cell splitting

- Split the cell values into two or more columns
- Specify a delimiter to split the cell values



Our dataset



Orders data from KIA - Online furniture shopping company

- Order ID
- Customer ID, Email ID
- Customer Location
- Furniture category and product name
- Cost of product
- Sales price of the product

Let's practice

DATA TRANSFORMATION IN KNIME

Enhance student data

DATA TRANSFORMATION IN KNIME



Let's practice!

DATA TRANSFORMATION IN KNIME

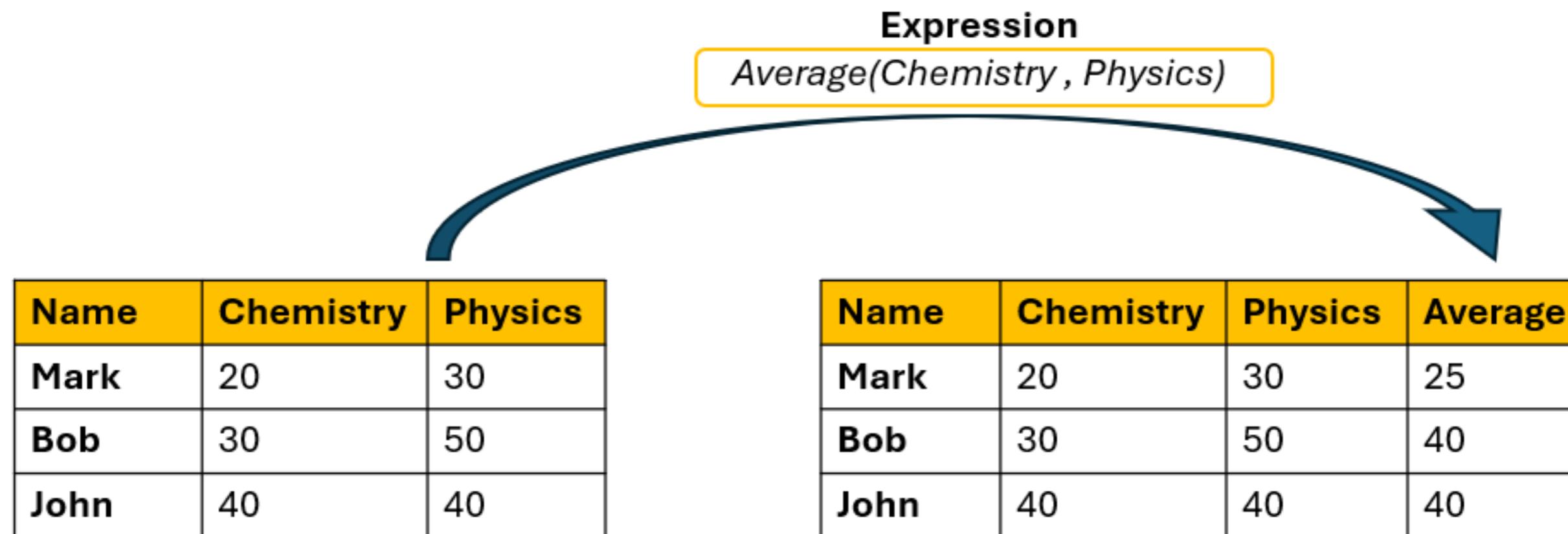
String and math expressions

DATA TRANSFORMATION IN KNIME



Expression-based transformation

- Allows applying custom logic
- Manipulate data with expressions
- Simple arithmetic to complex string manipulations



Formulating an expression

An expression can be formulated by

- Using operators

+ - = * /  **Operators**

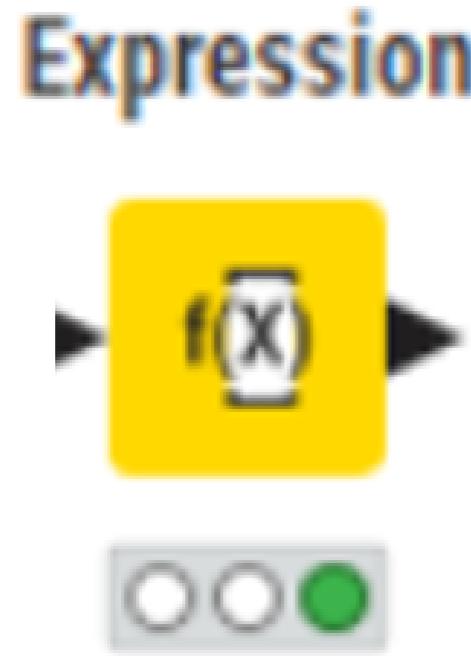
- Using functions

join(),mean(),max()  **Functions**

- Combination of operators and functions

mean() + max()  **Combination**

Expression Node



- Functions for string manipulation
- Math formulas
- Use a range of operators
- Create or replace a new column

String expressions

- Perform simple and complex string manipulations
- Use one or more string functions

Example

```
replace("I loved the m@vie", "@", "o")
```

Output

```
I loved the movie
```

Popular string functions

Name	Expression	Meaning
lowercase, uppercase	lowercase(x) uppercase(x)	Convert the entire string to lower/upper case
capitalize	capitalize(x)	Capitalize first letter of each word

Popular string functions

Name	Expression	Meaning
lowercase, uppercase	lowercase(x) uppercase(x)	Convert the entire string to lower/upper case
capitalize	capitalize(x)	Capitalize first letter of each word
join	join(separator, x1, x2)	Join 2 or more strings with separator

Popular string functions

Name	Expression	Meaning
lowercase, uppercase	lowercase(x) uppercase(x)	Convert the entire string to lower/upper case
capitalize	capitalize(x)	Capitalize first letter of each word
join	join(separator, x1, x2)	Join 2 or more strings with separator
replace	replace(x, pattern, replacement)	Replace pattern with replacement
find	find(x, string)	Find the string location index in x
length	length(x)	Length of input string

Math expressions

- Perform mathematical transformation
- Use math operators or functions and formulas

Example

```
max(20, 30, 70+5)
```

Output

```
75
```

Popular math functions

Name	Expression	Meaning
max, min,	max(x), min(x)	Maximum and minimum value

Popular math functions

Name	Expression	Meaning
max, min,	max(x), min(x)	Maximum and minimum value
stddev, average, median	stddev(x), average(x), median(x)	Standard deviation, average and median of value

Popular math functions

Name	Expression	Meaning
max, min,	max(x), min(x)	Maximum and minimum value
stddev, average, median	stddev(x), average(x), median(x)	Standard deviation, average and median of value
log, exp	log(x), exp(x)	Logarithm and exponential of a value
round	round(x, precision)	Round off the value up to precision

Mixed expressions

- Perform string and mathematical transformation.
- Use math and string functions together.

Example

Val1	Val2
20	50

Add a comment column

Val1	Val2	Comment
20	50	The minimum value is 20

Mixed expressions

Calculate minimum value

Syntax

```
min(column1, column2)
```

Expression

```
min($Val1$, $Val2$)
```

Convert it to string format

Syntax

```
string(input)
```

Expression

```
string(min($Val1$, $Val2$))
```

Mixed expressions

Syntax

```
join(separator, string1, string2)
```

Expression

```
join(" ", "The minimum value is", string(min($val1$, $val2$)))
```

Output Comment column

```
The minimum value is 20
```

Let's practice!

DATA TRANSFORMATION IN KNIME

Calculate student scores

DATA TRANSFORMATION IN KNIME



Let's practice!

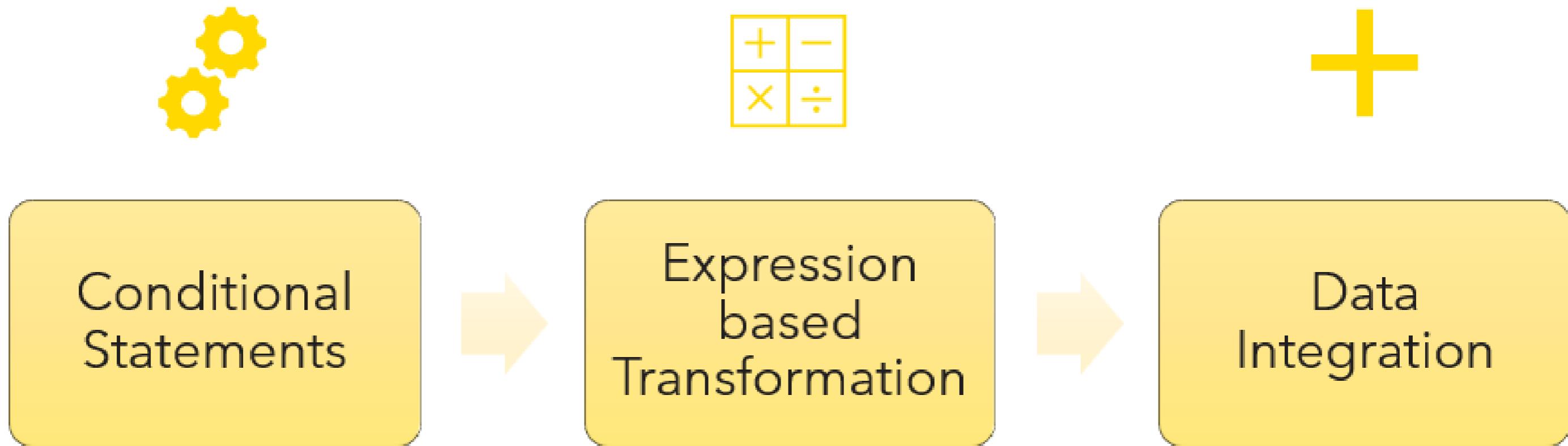
DATA TRANSFORMATION IN KNIME

Data integration

DATA TRANSFORMATION IN KNIME



Data integration

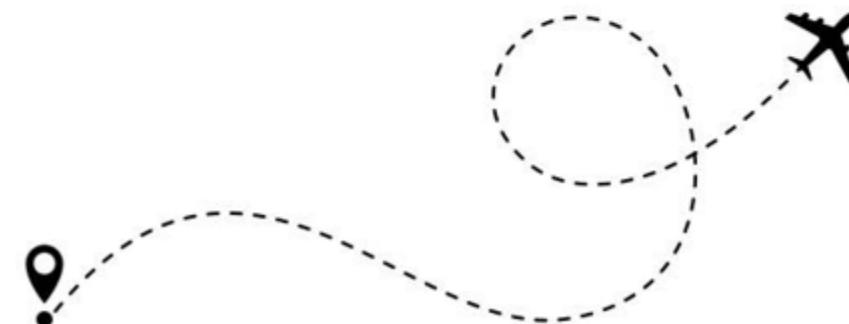


What is data integration?



Use case - Air travel

Flight Info			Meal Info		
ID	Departure	Price	ID	Meal1	Meal2
1A	Berlin	\$1000	1A	Sandwich	?
1B	Paris	\$500	1B	Juice	Salad
2A	Berlin	\$1500	2A	?	Salad
2C	Paris	\$2000	2C	Fruit Bowl	?



Append columns

Combined Info

ID	Departure	Price	Meal1	Meal2
1A	Berlin	\$1000	Sandwich	?
1B	Paris	\$500	Juice	Salad
2A	Berlin	\$1500	?	Salad
2C	Paris	\$2000	Fruit Bowl	?

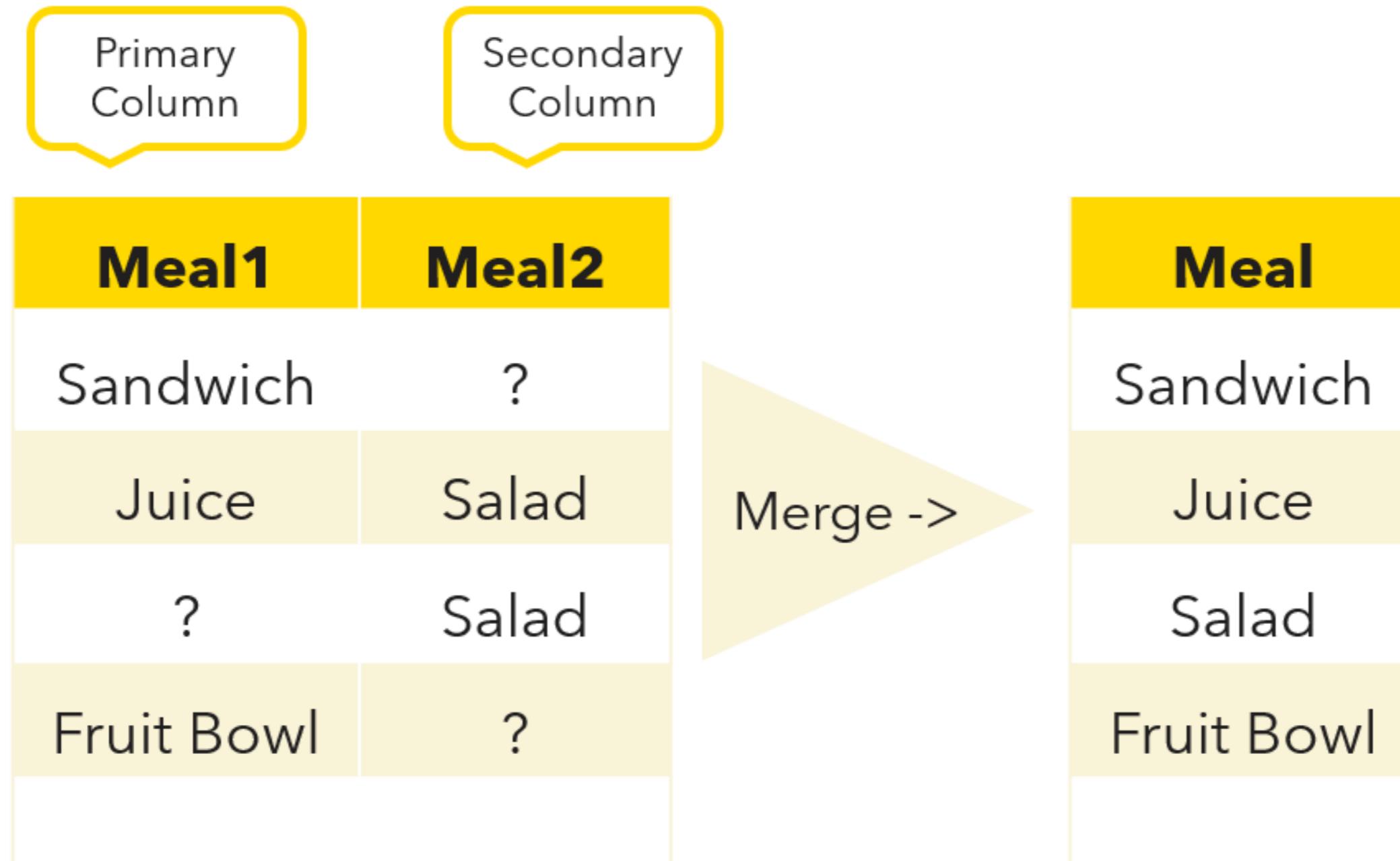
Meal preference

Meal1	Meal2
Sandwich	?
Juice	Salad
?	Salad
Fruit Bowl	?

Meal preference

		Primary Column	Secondary Column
		Meal1	Meal2
Sandwich			?
Juice		Salad	
	?		Salad
Fruit Bowl			?

Merge columns



Context

ID	Departure	Price	Meal
1A	Berlin	\$1000	Sandwich
1B	Paris	\$500	Juice
2A	Berlin	\$1500	Salad
2C	Paris	\$2000	Fruit Bowl

All flight
on 7th May

Constant value column



ID	Departure	Price	Meal	Date
1A	Berlin	\$1000	Sandwich	7 May
1B	Paris	\$500	Juice	7 May
2A	Berlin	\$1500	Salad	7 May
2C	Paris	\$2000	Fruit Bowl	7 May

Constant value column

ID	Departure	Price	Meal	Date
1A	Berlin	\$1000	Sandwich	7 May
1B	Paris	\$500	Juice	7 May
2A	Berlin	\$1500	Salad	7 May
2C	Paris	\$2000	Fruit Bowl	7 May

Reorder column sequence

Move to first position

ID	Departure	Price	Meal	Date
1A	Berlin	\$1000	Sandwich	7 May
1B	Paris	\$500	Juice	7 May
2A	Berlin	\$1500	Salad	7 May
2C	Paris	\$2000	Fruit Bowl	7 May

Reorder column sequence

Date	ID	Departure	Price	Meal
7 May	1A	Berlin	\$1000	Sandwich
7 May	1B	Paris	\$500	Juice
7 May	2A	Berlin	\$1500	Salad
7 May	2C	Paris	\$2000	Fruit Bowl

Reorder column sequence

Date	ID	Departure	Price	Meal
7 May	1A	Berlin	\$1000	Sandwich
7 May	1B	Paris	\$500	Juice
7 May	2A	Berlin	\$1500	Salad
7 May	2C	Paris	\$2000	Fruit Bowl



Reorder column sequence

Date	ID	Departure	Meal	Price
7 May	1A	Berlin	Sandwich	\$1000
7 May	1B	Paris	Juice	\$500
7 May	2A	Berlin	Salad	\$1500
7 May	2C	Paris	Fruit Bowl	\$2000

Sorting

First Criteria -Sort alphabetically

Date	ID	Departure	Meal	Price
7 May	1A	Berlin	Sandwich	\$1000
7 May	1B	Paris	Juice	\$500
7 May	2A	Berlin	Salad	\$1500
7 May	2C	Paris	Fruit Bowl	\$2000

Sorting

Sorted by Departure

Date	ID	Departure	Meal	Price
7 May	1A	Berlin	Sandwich	\$1000
7 May	2A	Berlin	Salad	\$1500
7 May	1B	Paris	Juice	\$500
7 May	2C	Paris	Fruit Bowl	\$2000

Sorting - add criteria

Second Criteria -Sort Descending

Date	ID	Departure	Meal	Price
7 May	1A	Berlin	Sandwich	\$1000
7 May	2A	Berlin	Salad	\$1500
7 May	1B	Paris	Juice	\$500
7 May	2C	Paris	Fruit Bowl	\$2000

Sorting - add criteria

Date	ID	Departure	Meal	Price
7 May	2A	Berlin	Salad	\$1500
7 May	1A	Berlin	Sandwich	\$1000
7 May	2C	Paris	Fruit Bowl	\$2000
7 May	1B	Paris	Juice	\$500

Expensive flight -2A

Let's practice!

DATA TRANSFORMATION IN KNIME

Integrate and analyze students data

DATA TRANSFORMATION IN KNIME



Let's practice!

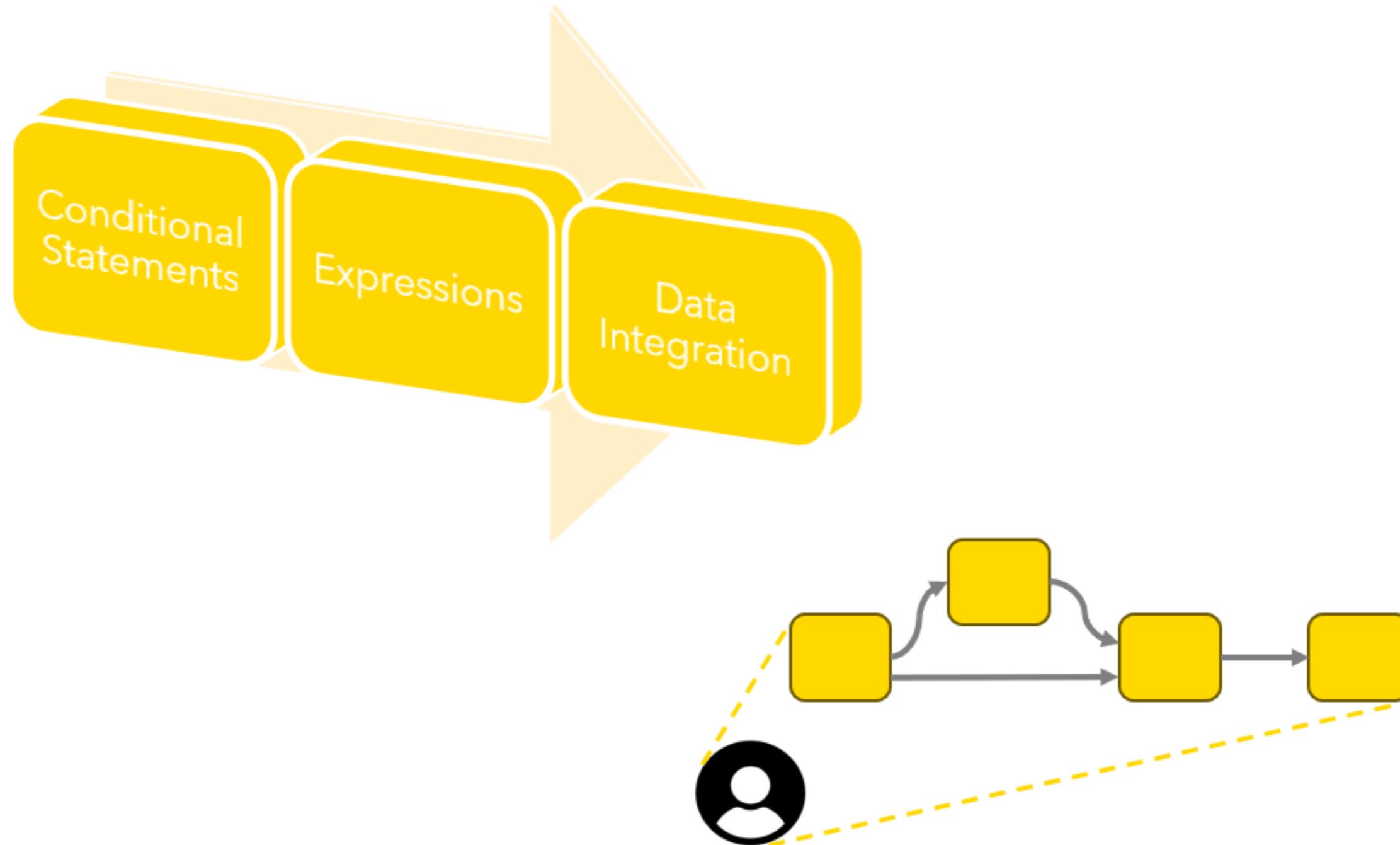
DATA TRANSFORMATION IN KNIME

Share workflows & collaborate

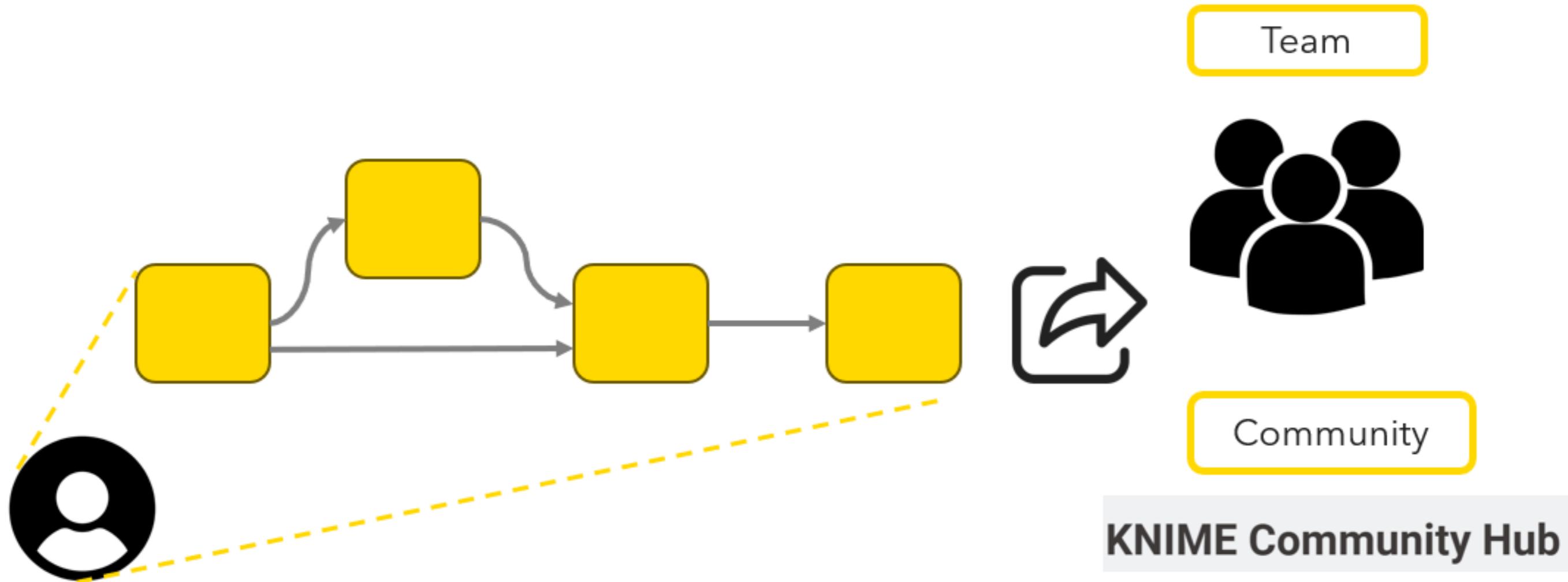
DATA TRANSFORMATION IN KNIME



Sharing & collaboration



Sharing & collaboration



KNIME Community Hub

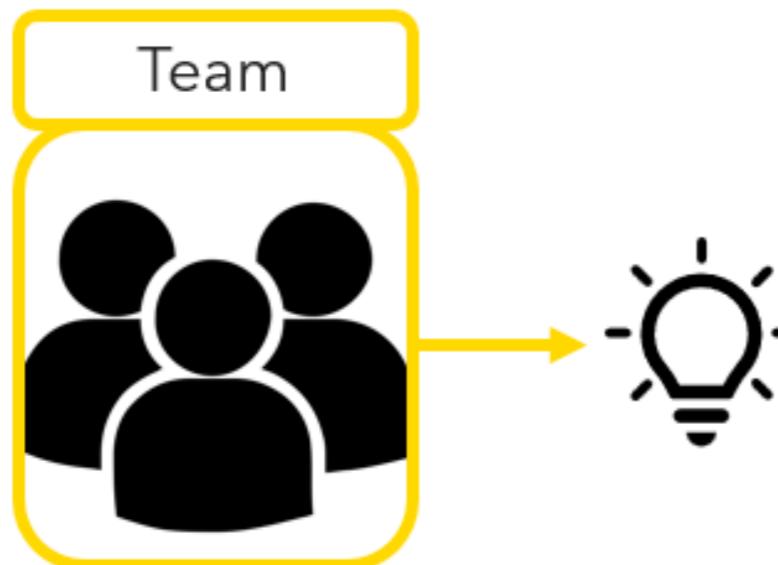
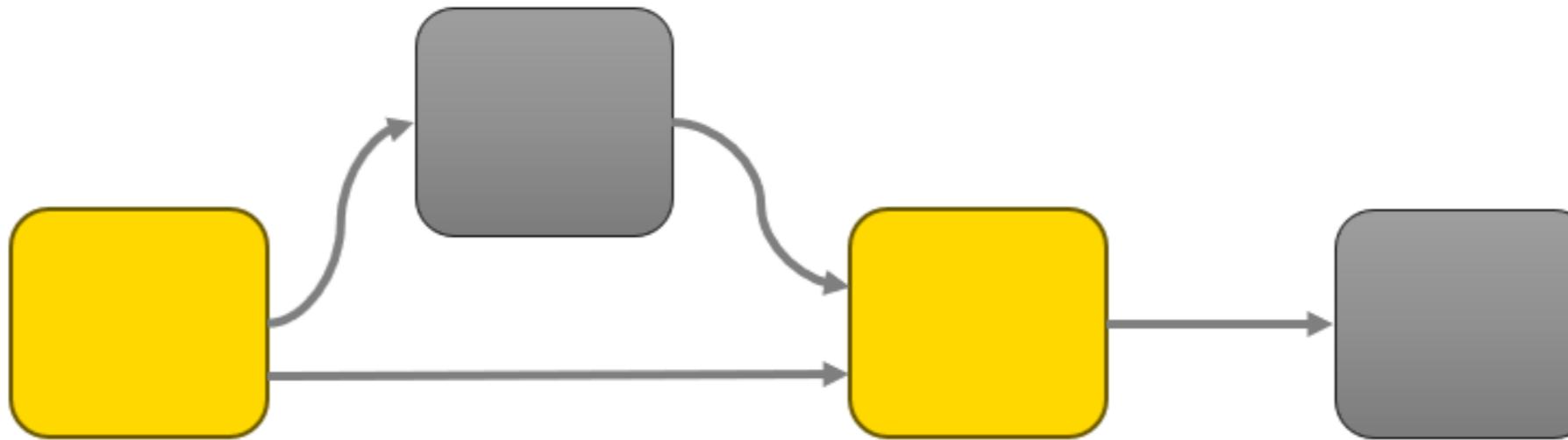
Welcome to KNIME Community Hub

Access [26839 workflows](#), [2337 components](#) and [252 extensions](#)
provided by KNIME and our community.



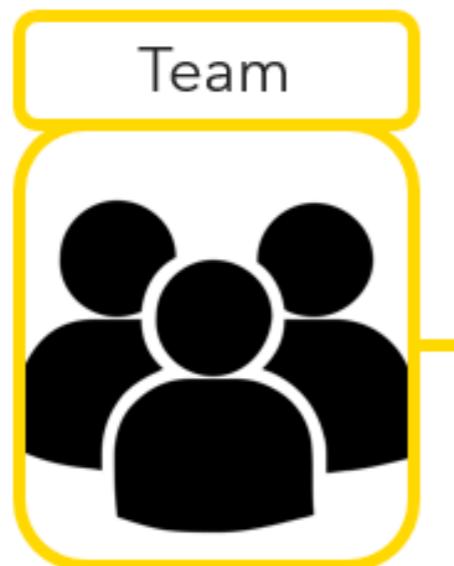
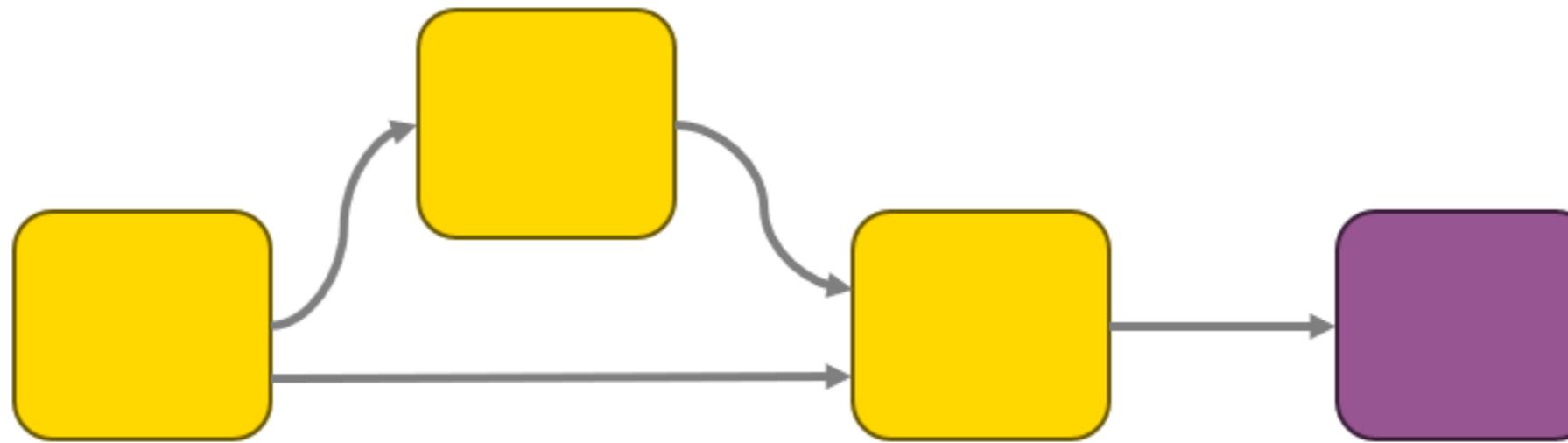
- Collections of workflows and resources
- Ability to search for workflows built by community
- Share workflows with community
- Collaborate with users from different disciplines

Sharing with team



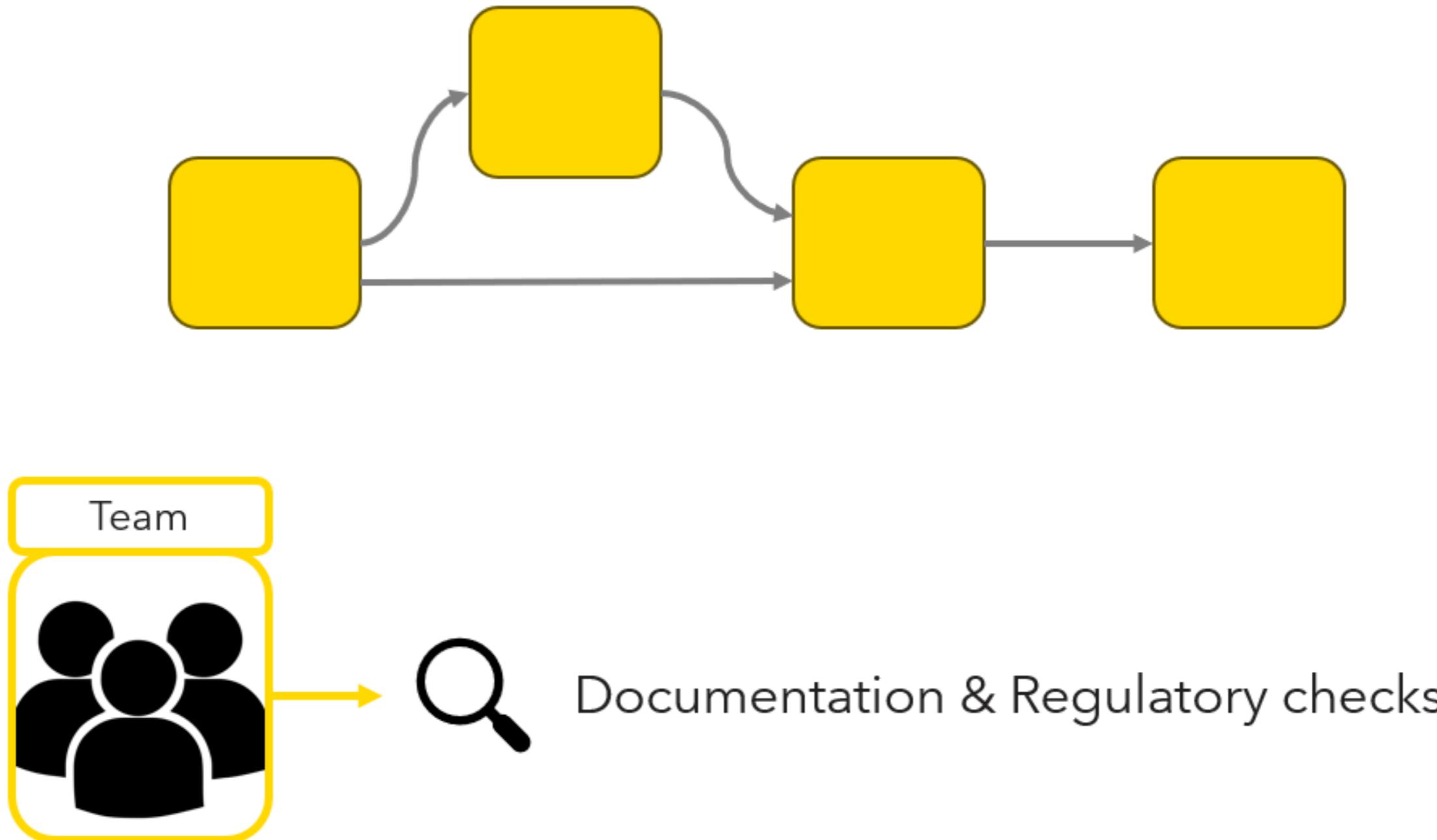
Use it as a reference, to create new workflows

Sharing with team

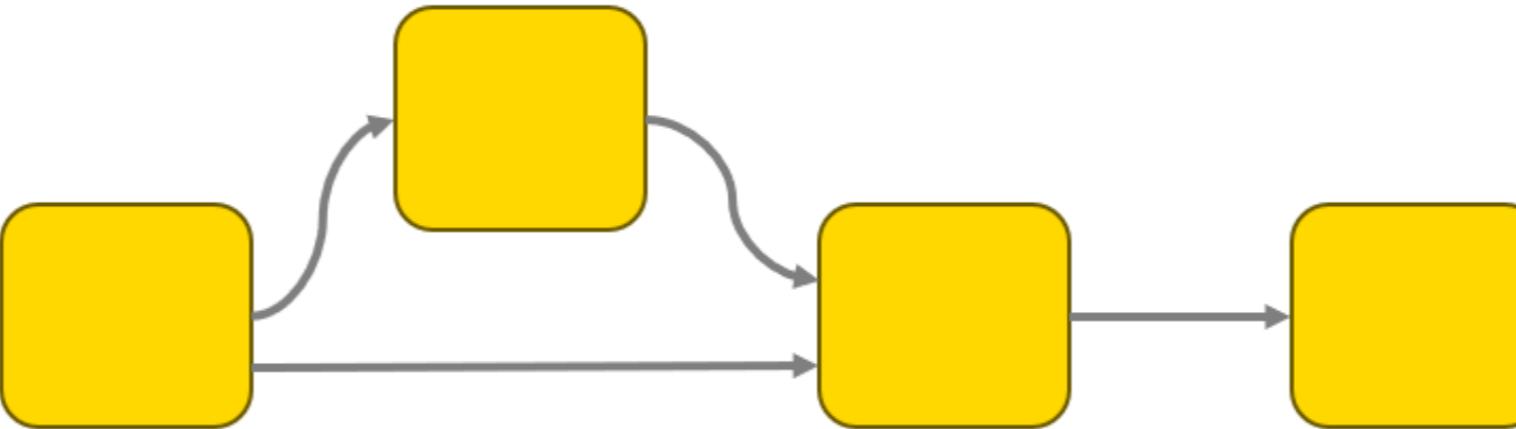


Modify and collaborate

Sharing with team



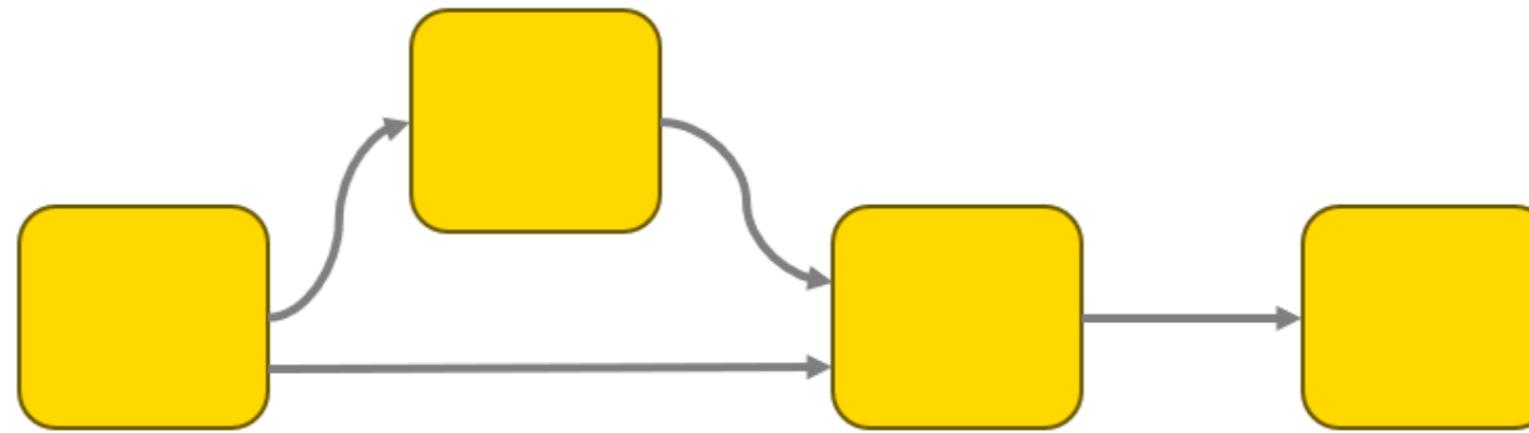
Workflow description



Workflow Description

- Objective
- Data used
- Key Transformations
 - Output

Workflow tags



Workflow Description

- Objective
- Data used
- Key Transformations
 - Output

Tags

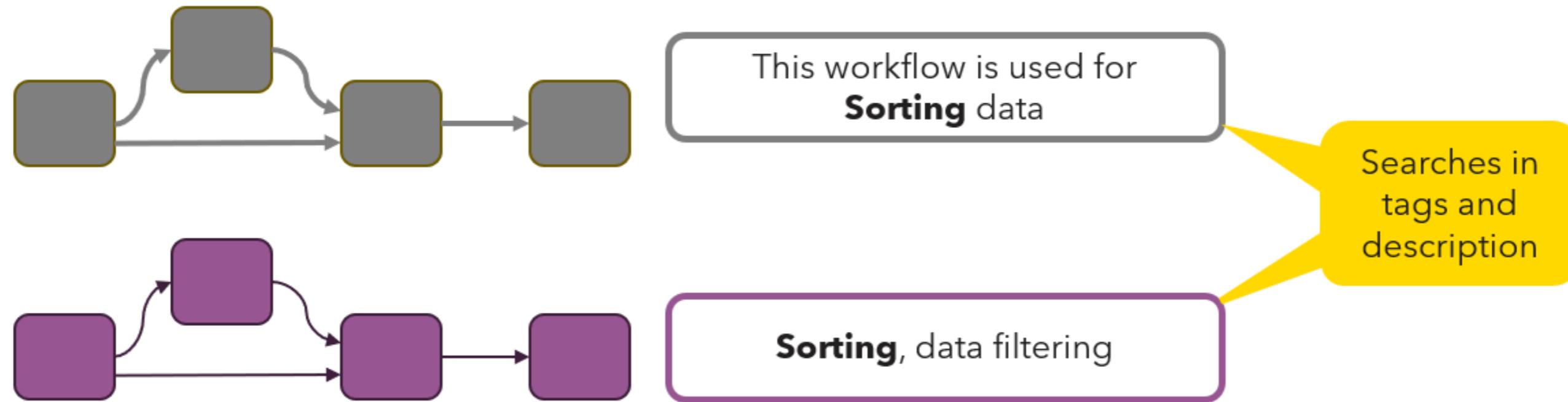
<sorting>
<furniture data>
<profit calculation>
<data filtering>

Search & retrieve

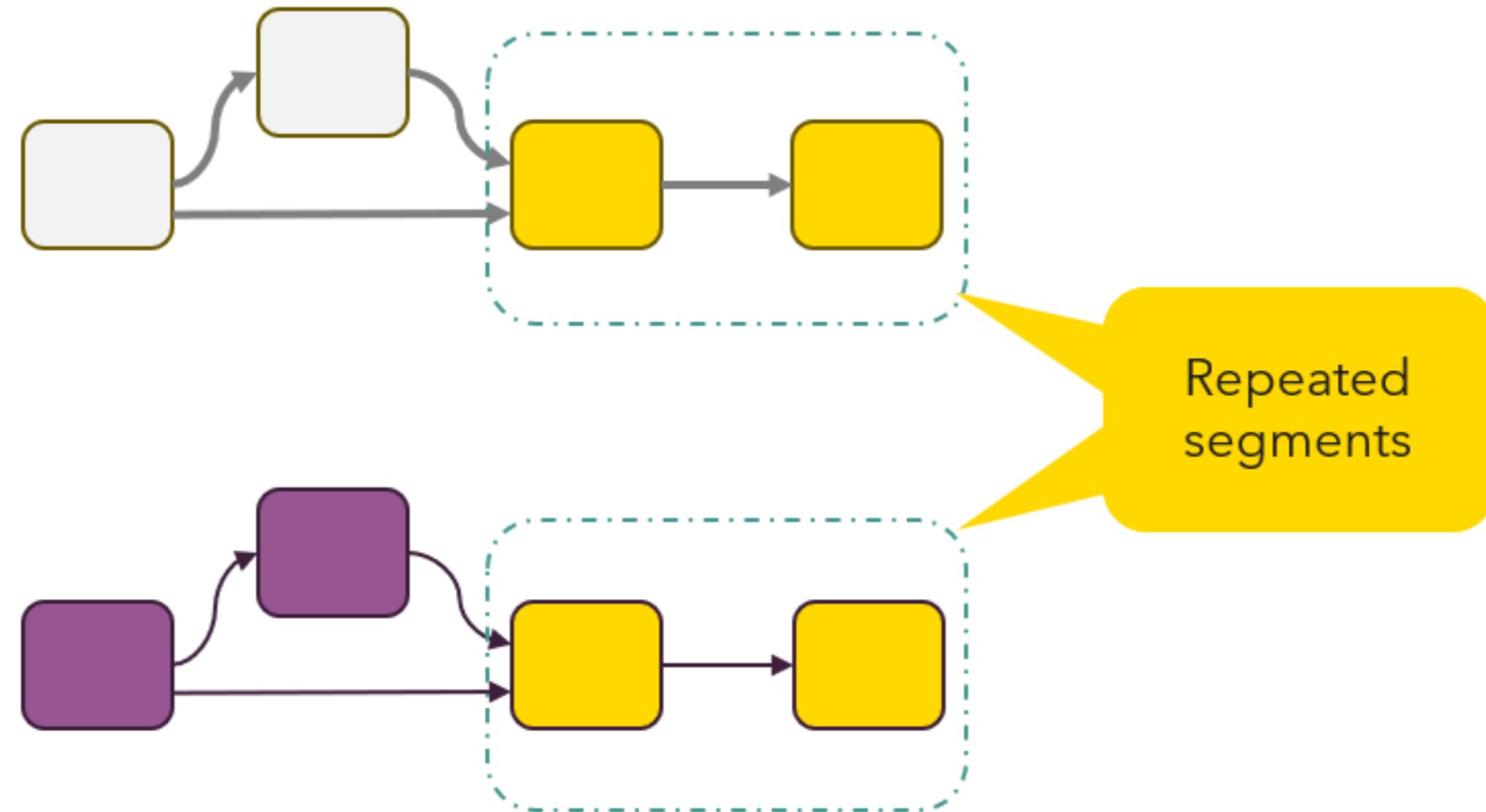
Welcome to KNIME Community Hub

Access [26839 workflows](#), [2337 components](#) and [252 extensions](#)
provided by KNIME and our community.

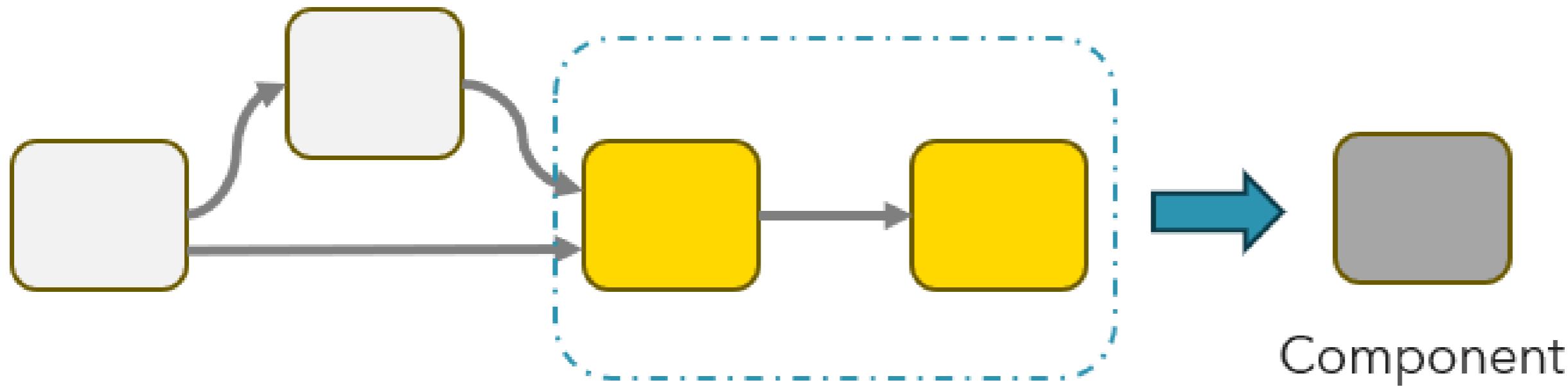
sorting ×



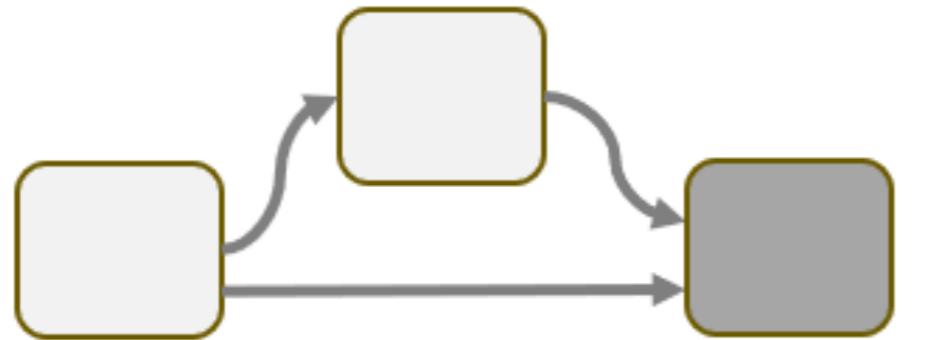
KNIME component



KNIME component

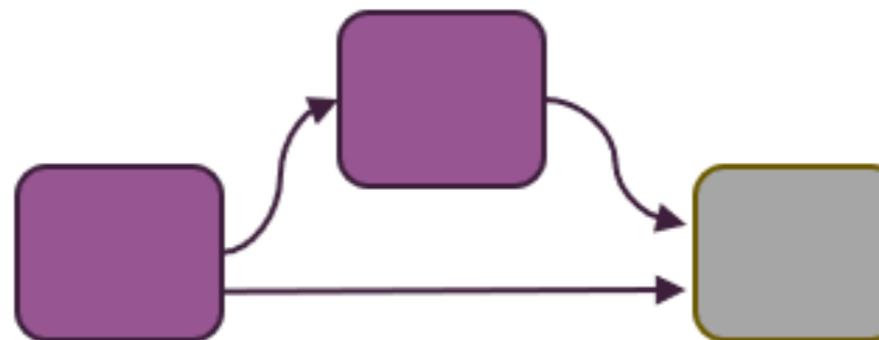


KNIME component



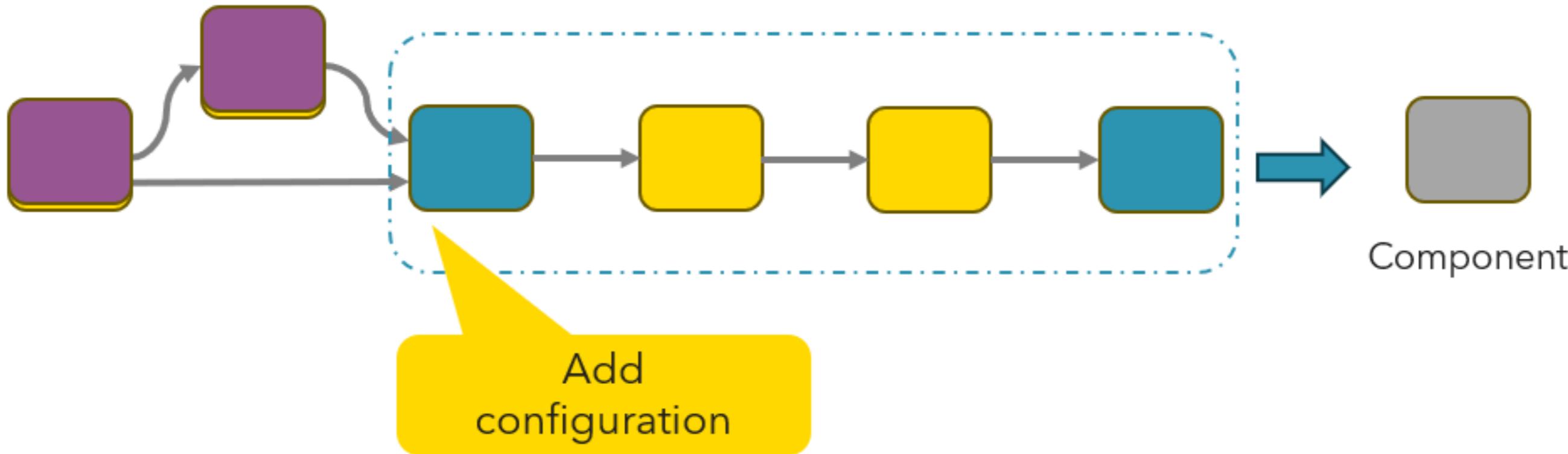
Component

Build once & reuse the
component in any workflow

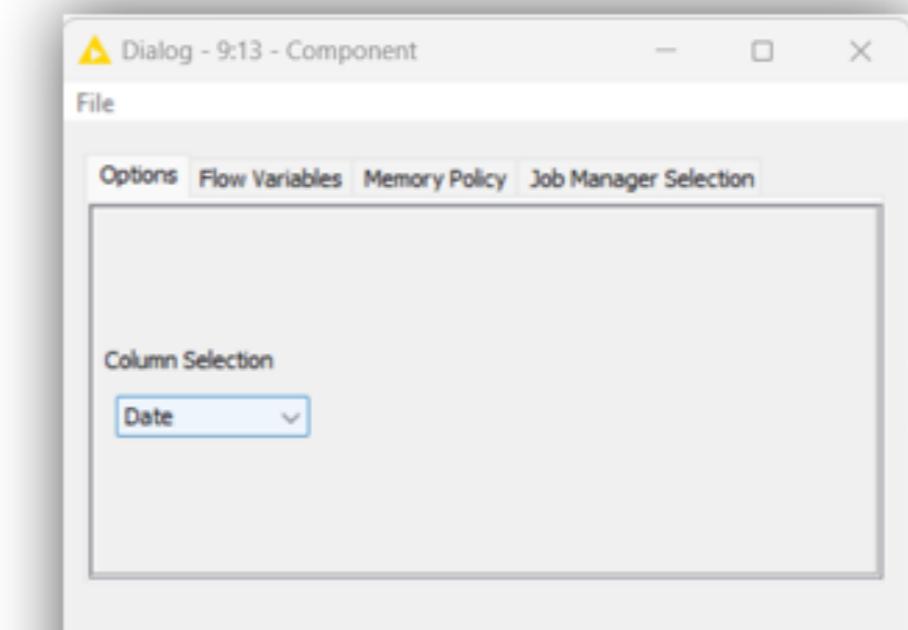
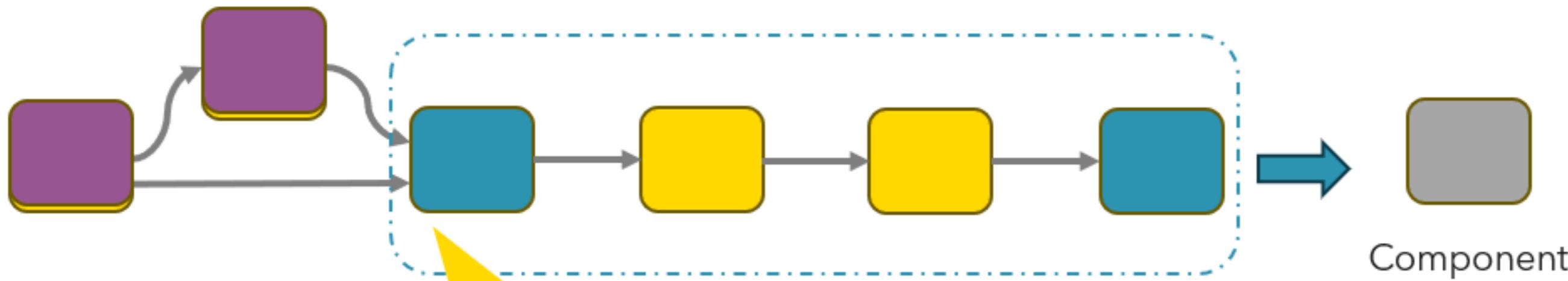


Component

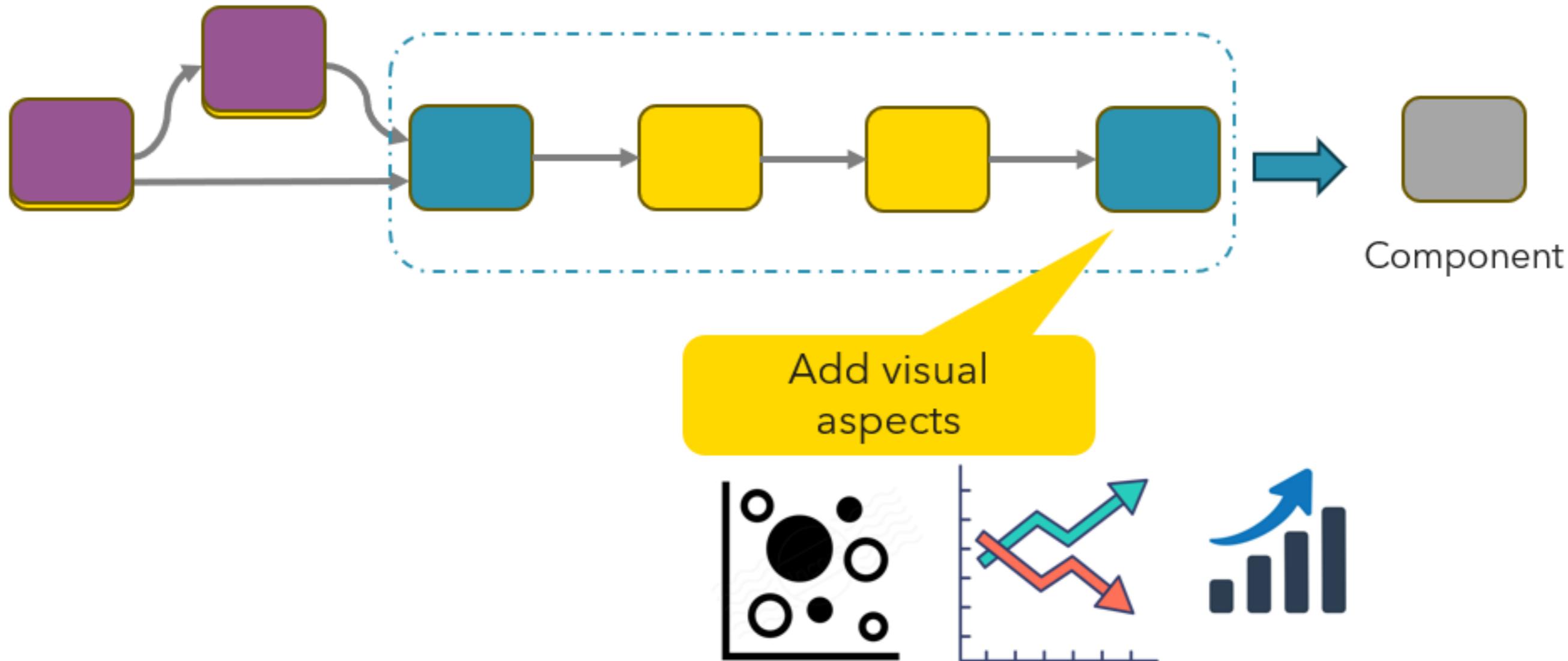
Adding configuration



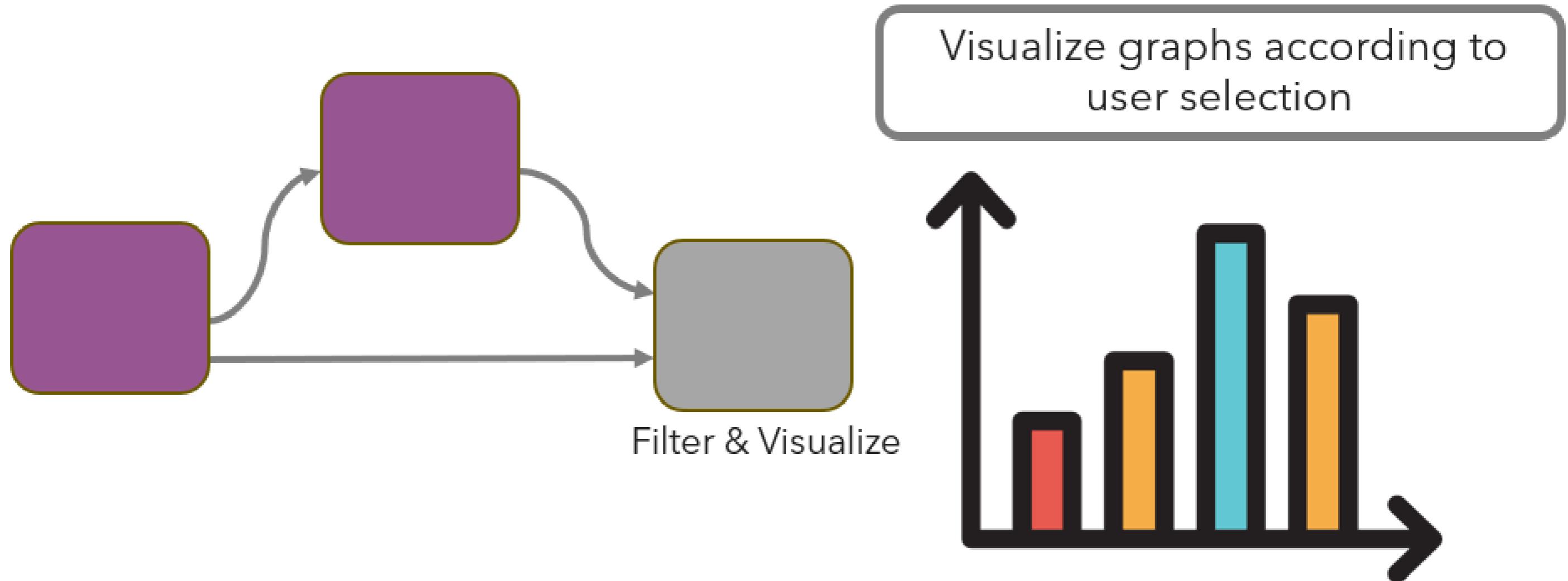
Adding configuration



Adding visualization



Component view



Component details



Filter & Visualize

Component Description

Tags

- Objective
- Configurations
 - Output

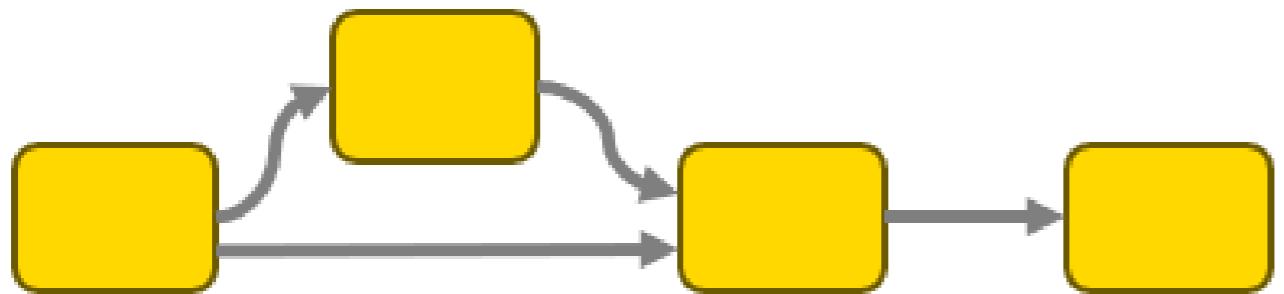
<bar graph>

<column selection>

<sorting>

<data filtering>

Ready to collaborate



Filter & Visualize



KNIME Community Hub

Let's practice!

DATA TRANSFORMATION IN KNIME

Share & collaborate student's workflow

DATA TRANSFORMATION IN KNIME



Let's practice!

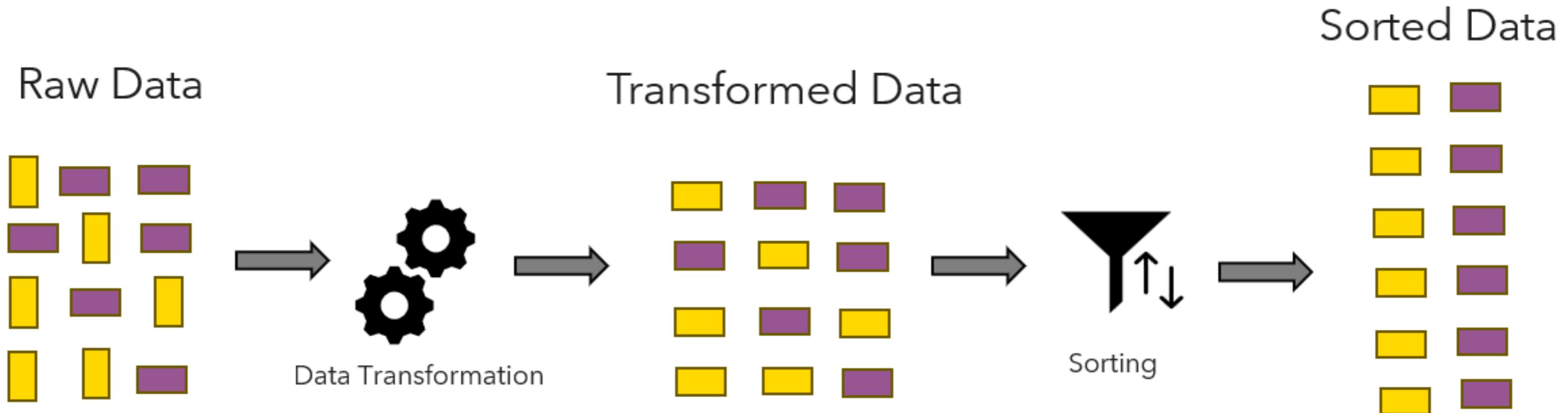
DATA TRANSFORMATION IN KNIME

Course wrap up

DATA TRANSFORMATION IN KNIME



Data transformation process



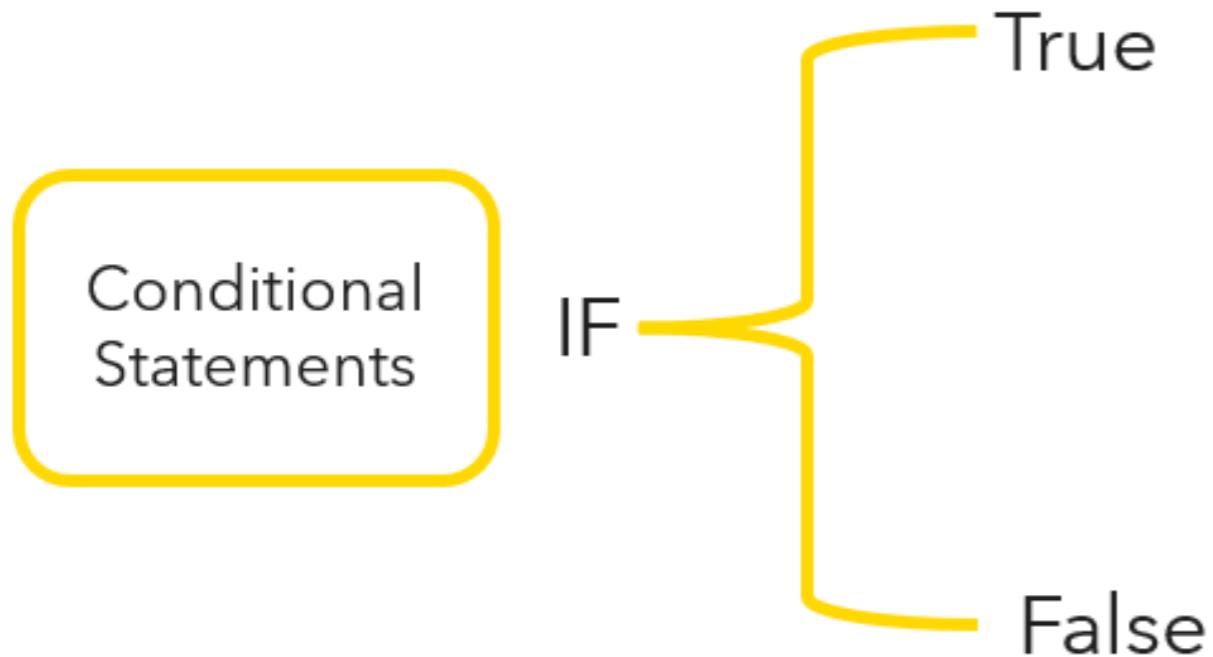
Online orders data



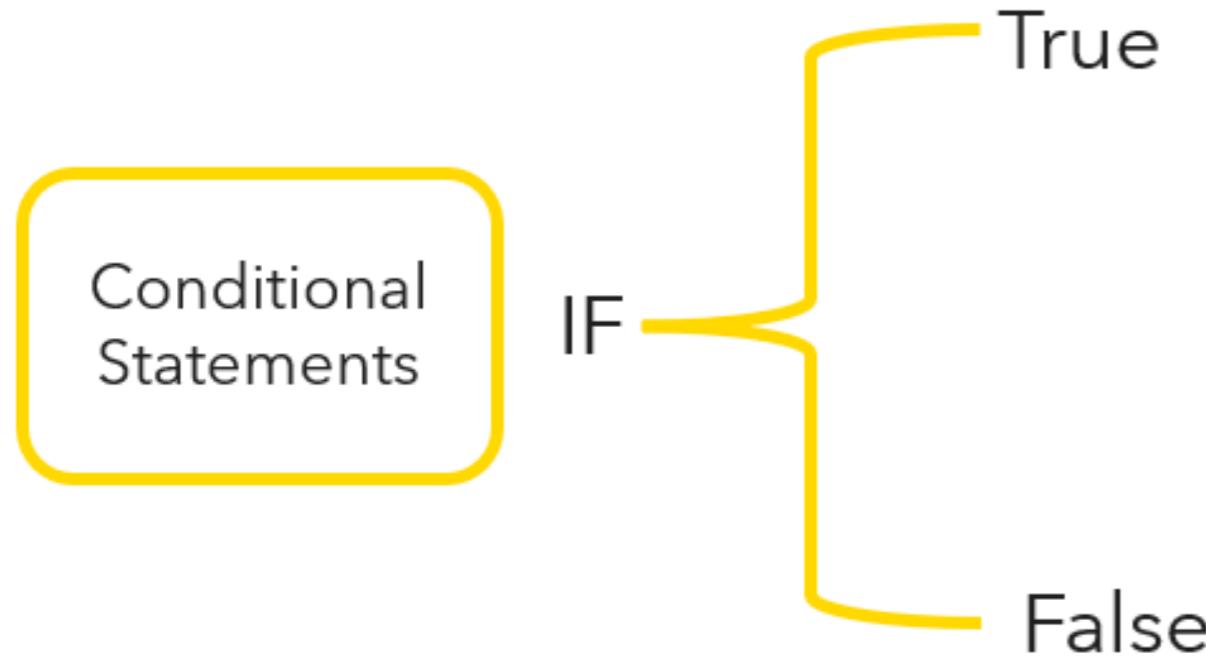
Order ID	Customer	Product	Sales Price	COGS
1	Bob	Chair	\$100	\$80
2	John	Wardrobe	\$200	\$150
.

DATA TRANSFORMATION IN KNIME

Conditional statements



Operators



Operator	Meaning
>	Greater than
<	Less than
=	Equal to
\geq	Greater than equal
\leq	Less than equal

Discount%

Created using Conditional Statement

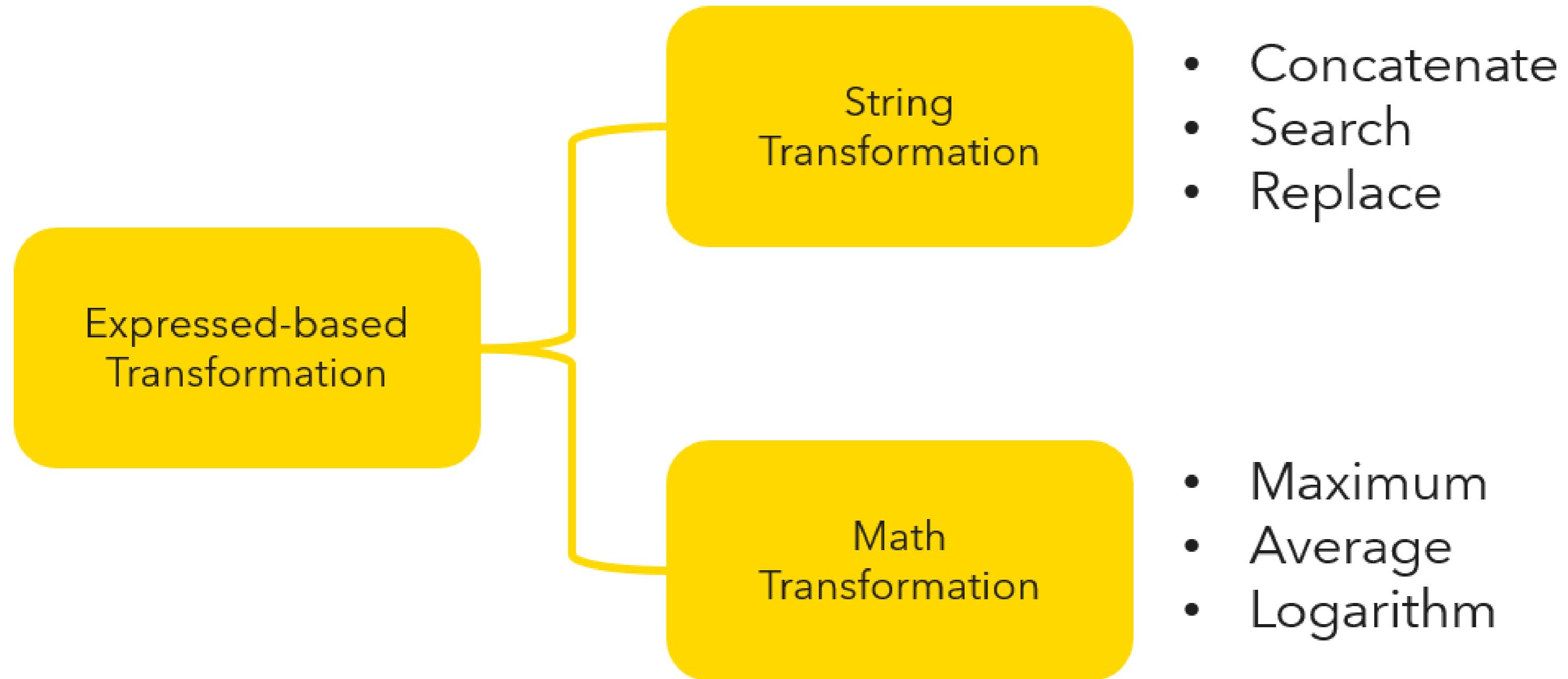
Order ID	Qty	Discount%
1	20	5
2	50	10

Expression

f(x)

...

Expression-based transformation



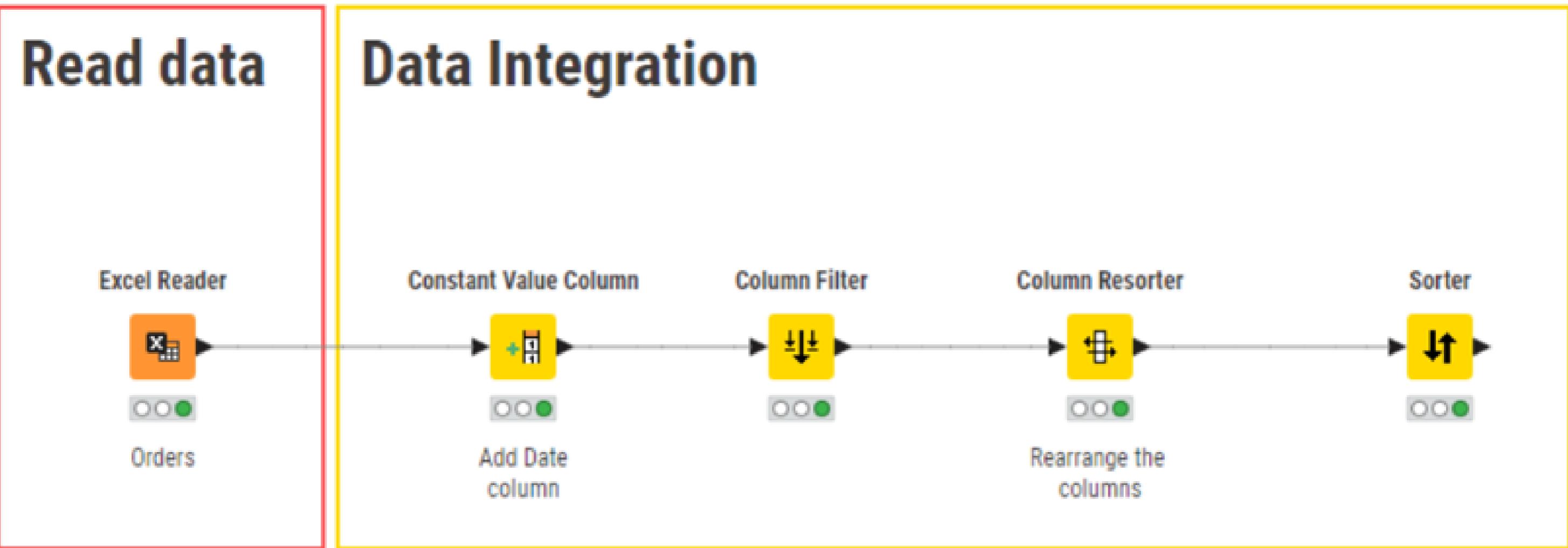
Compute the profit

Created using Math & String expression

Expression
f(x)

Order ID	Sales price	COGS	Profit
1	\$100	\$80	\$20
2	\$200	\$150	\$50
.	.	.	.
.	.	.	.

Data integration

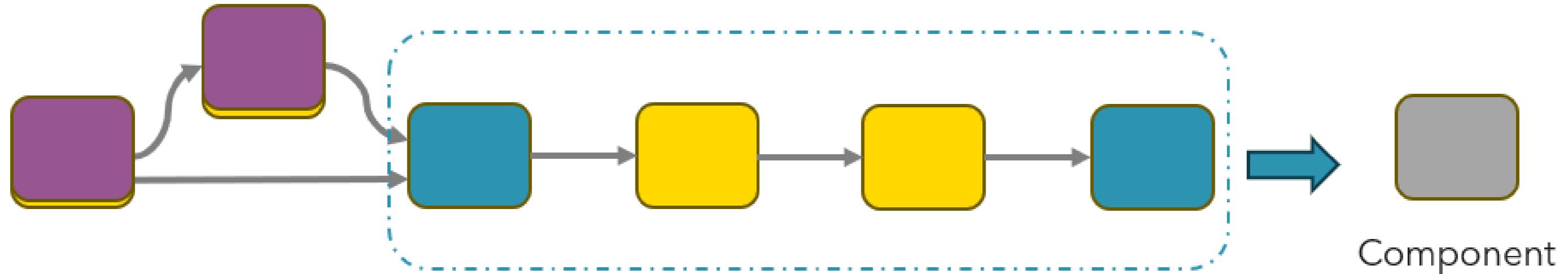


Most profitable orders

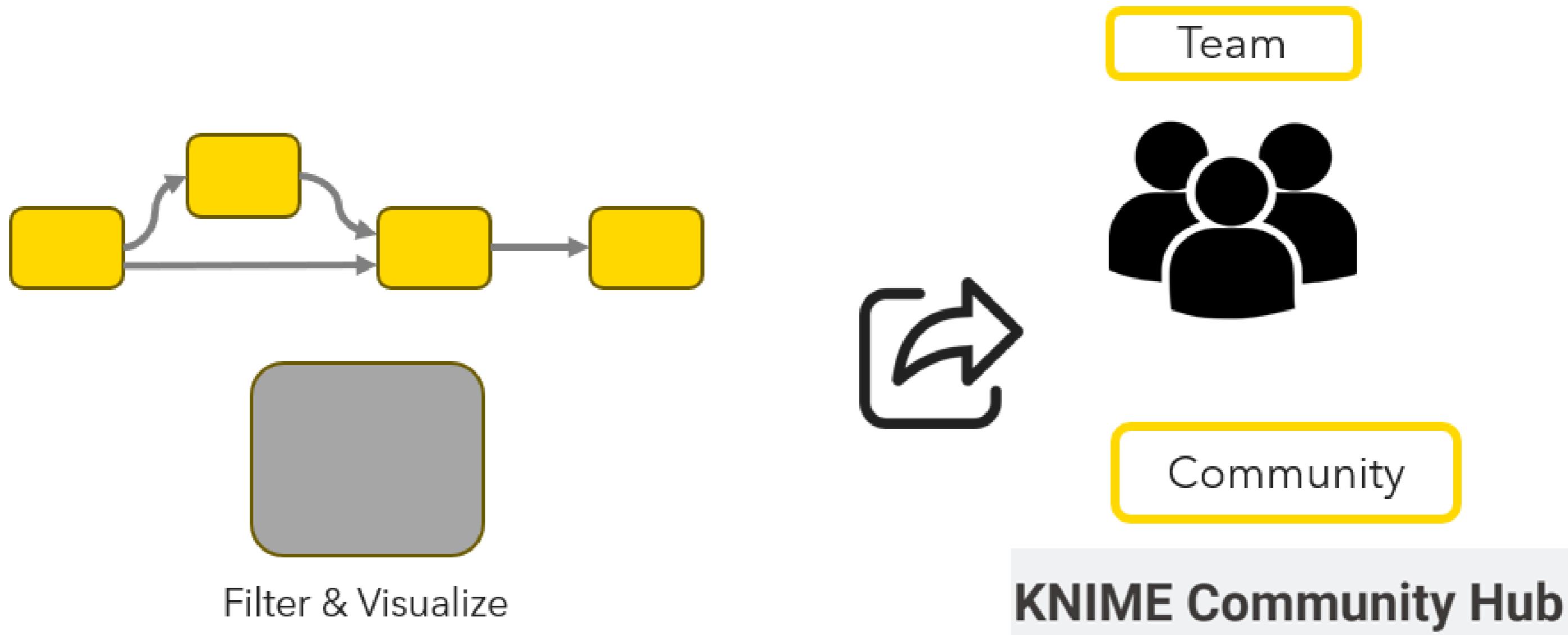
Sort by Profit

Order ID	Sales price	COGS	Profit
2	\$200	\$150	\$50
1	\$100	\$80	\$20
.	.	.	
.	.	.	

Component creation



Add details & share



Get started



DATA TRANSFORMATION IN KNIME

Get started



DATA TRANSFORMATION IN KNIME

Start sharing!

DATA TRANSFORMATION IN KNIME

Concatenation, Lookups, and Joins

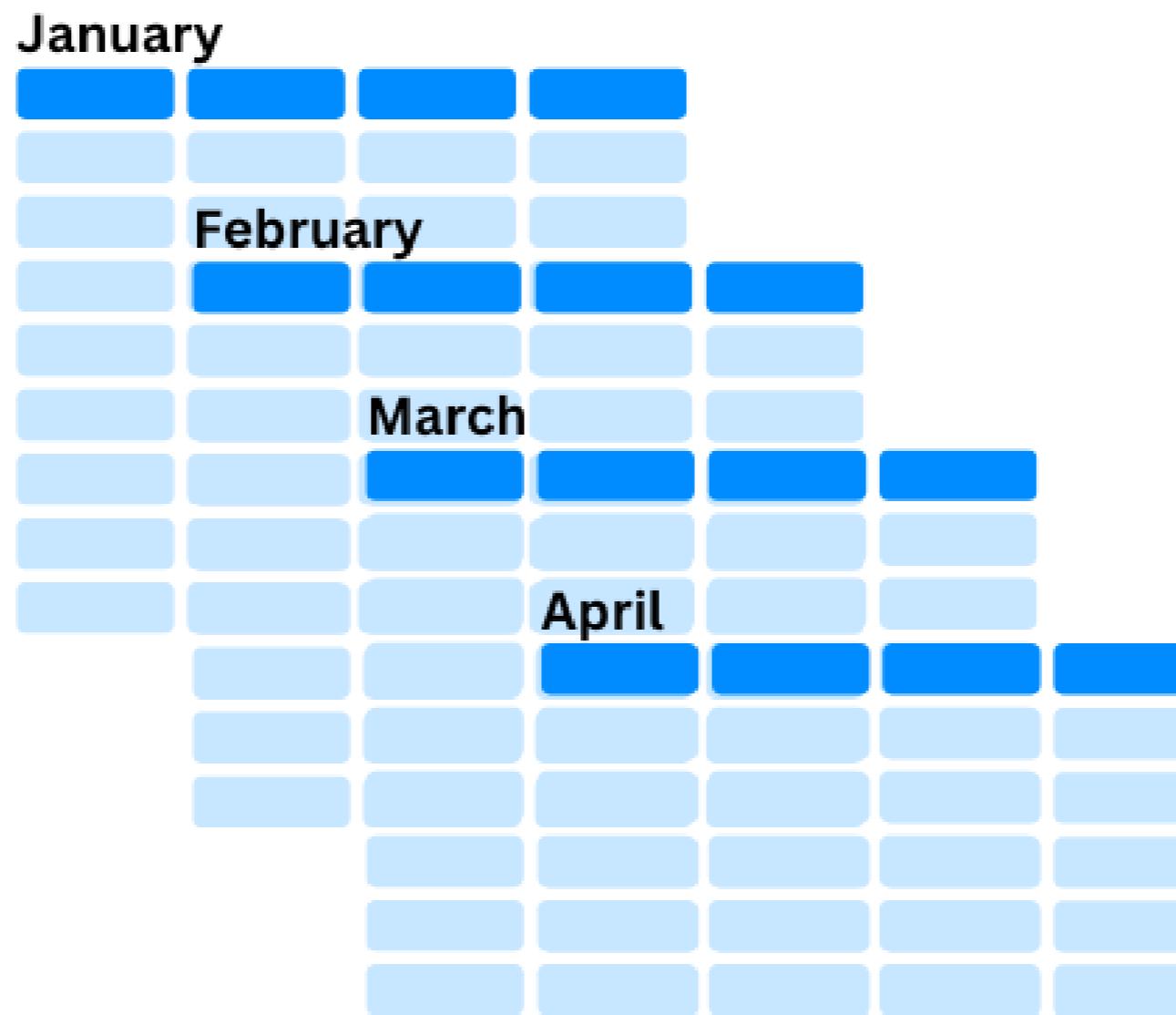
DATA MANIPULATION IN KNIME



The CFO needs clarity



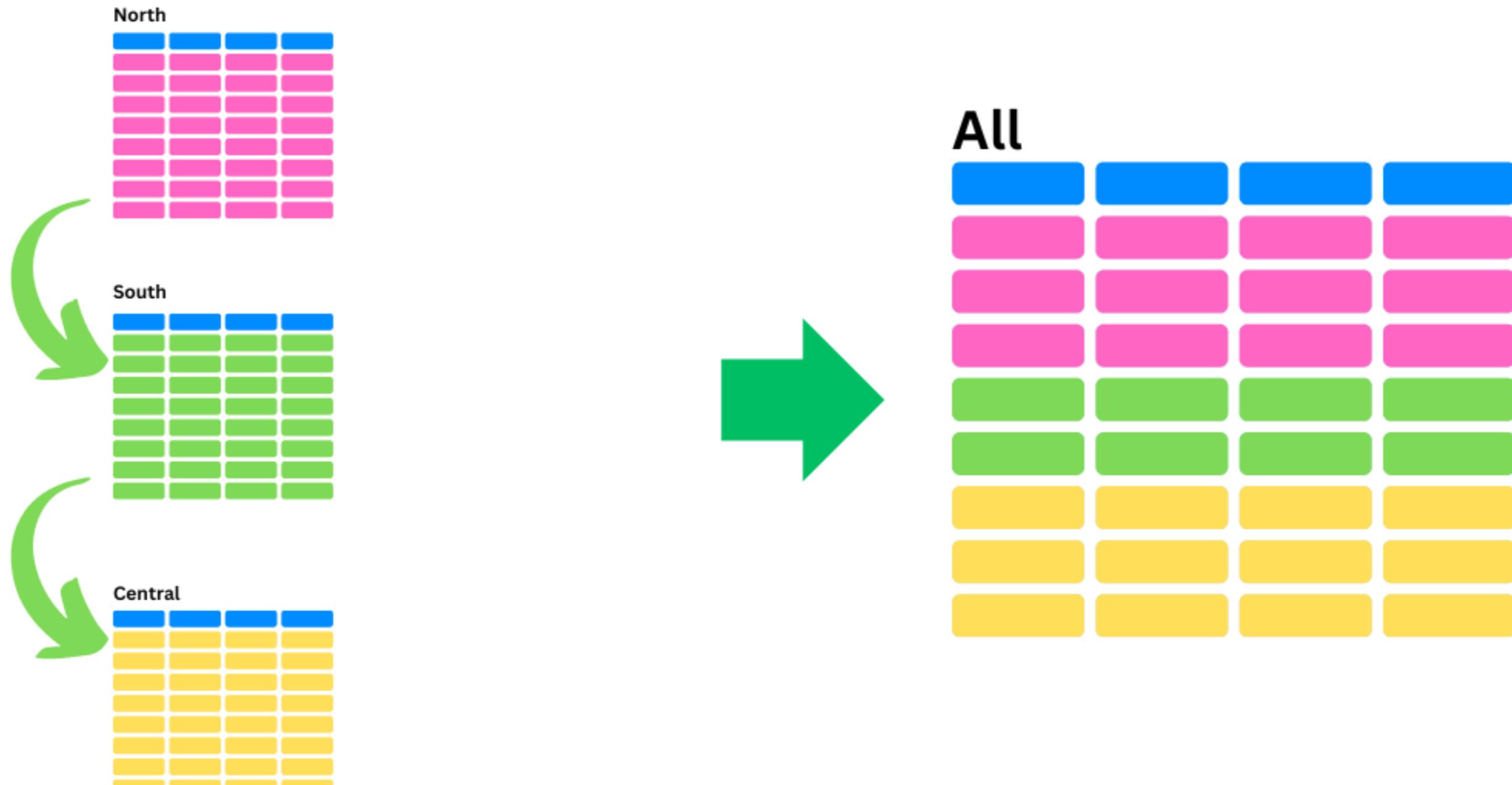
Concatenating tables



Concatenating tables

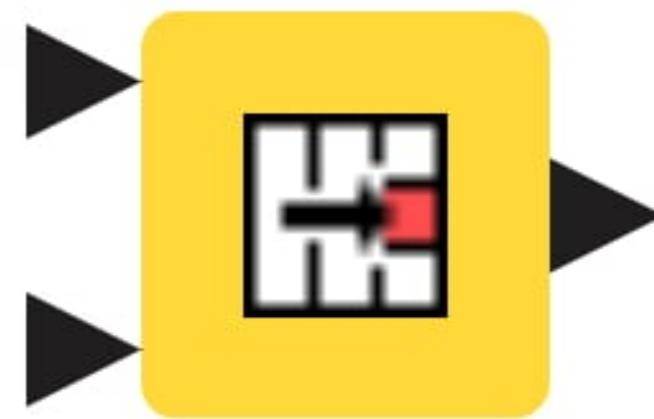


Concatenating tables



Simple lookups

Value Lookup



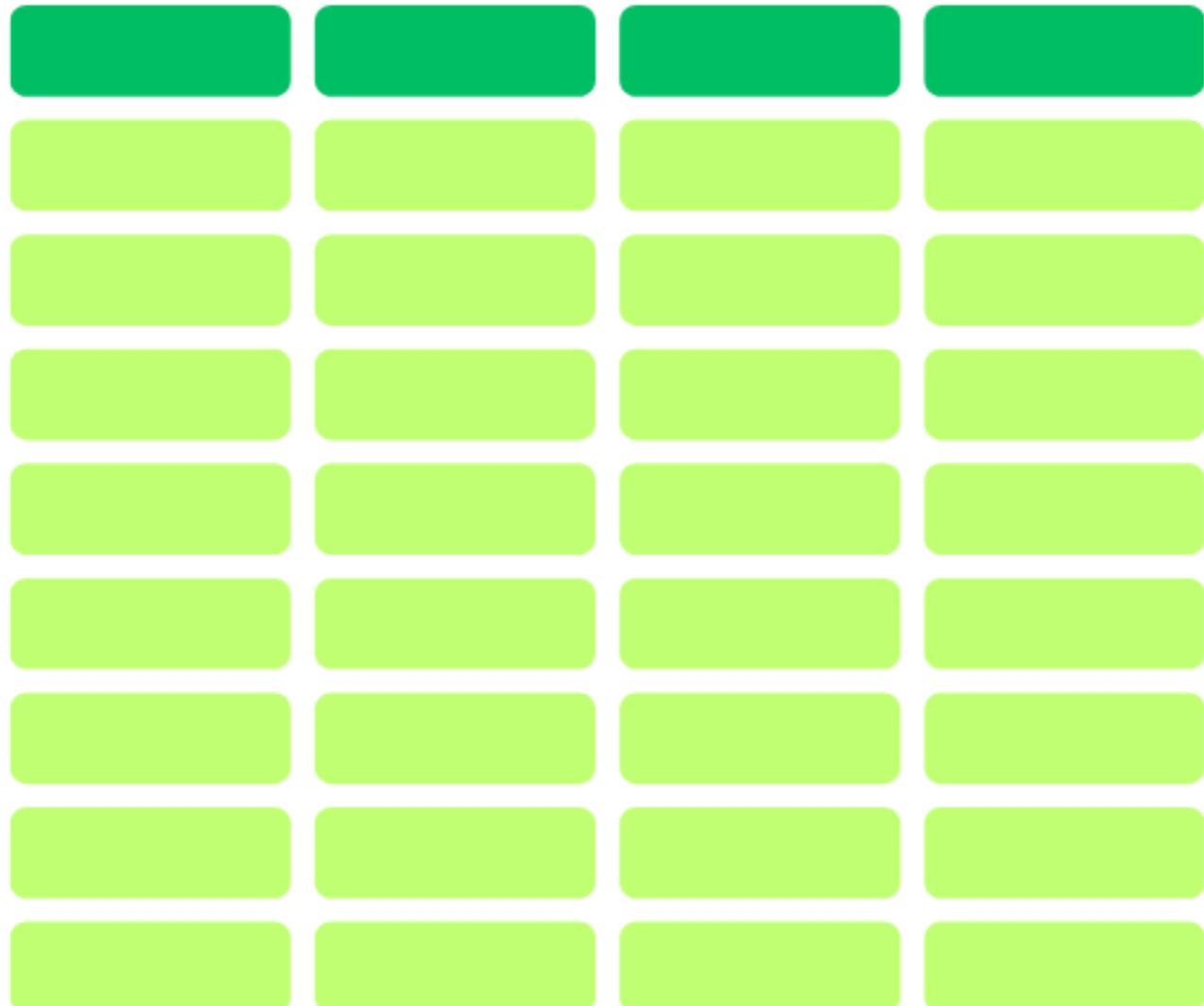
Simple lookups

- Enrich data
- Differing table structures



Simple lookups

Transactions



Products



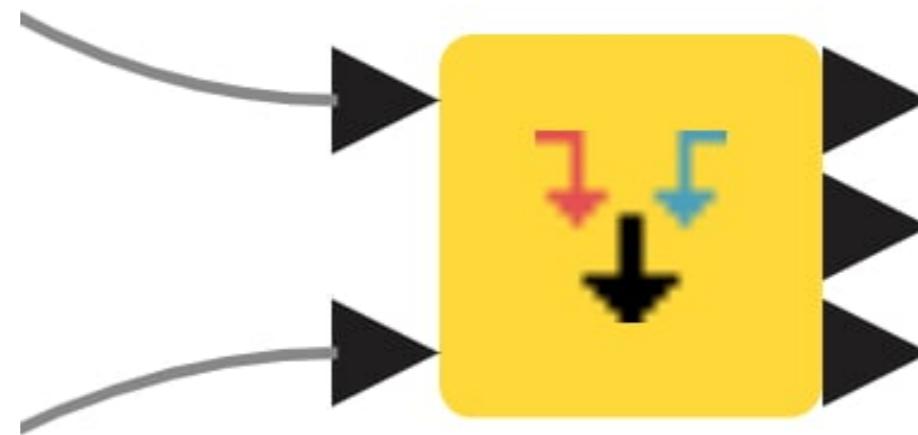
Simple lookups

Transactions

Products

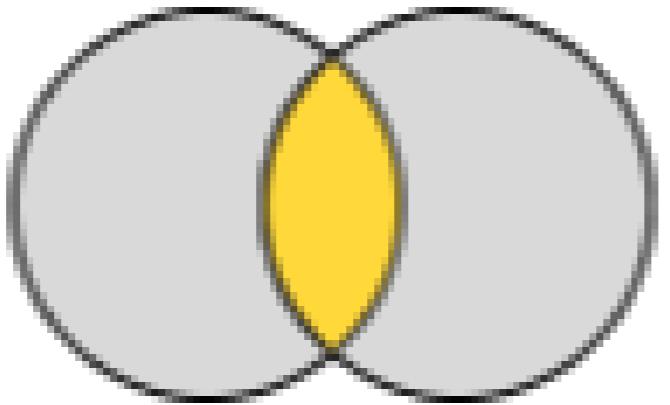
Merging with Joins

Joiner



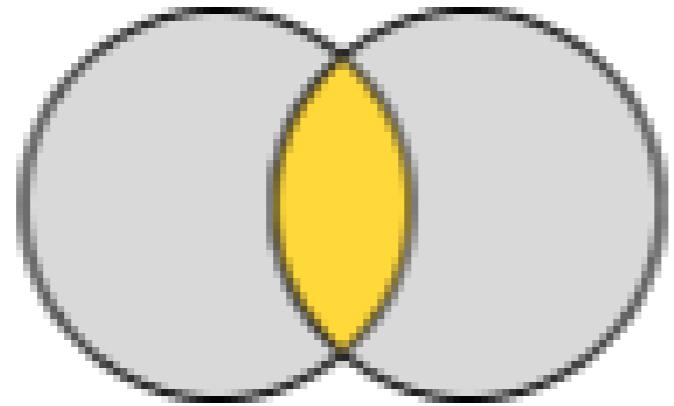
Merging with Joins

Inner Join

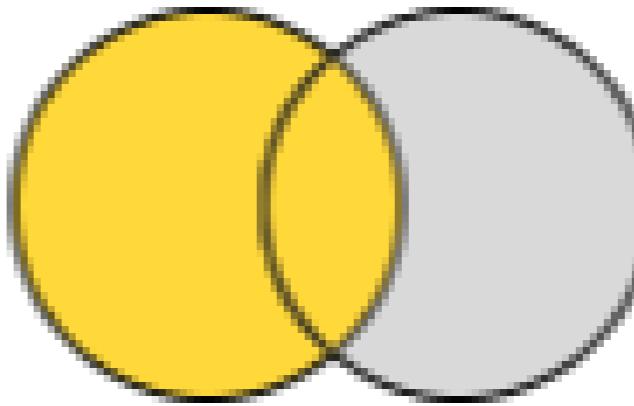


Merging with Joins

Inner Join

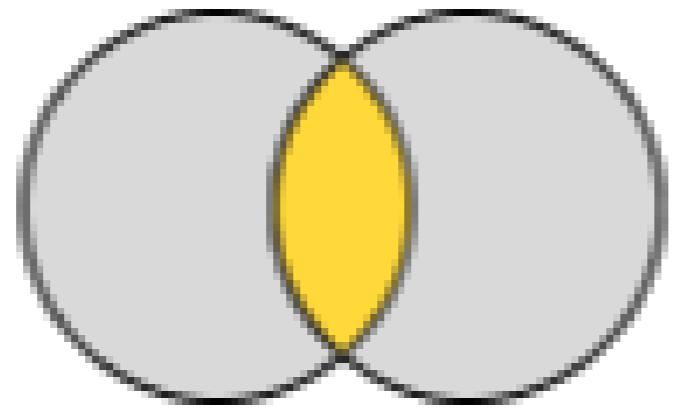


Left Outer Join

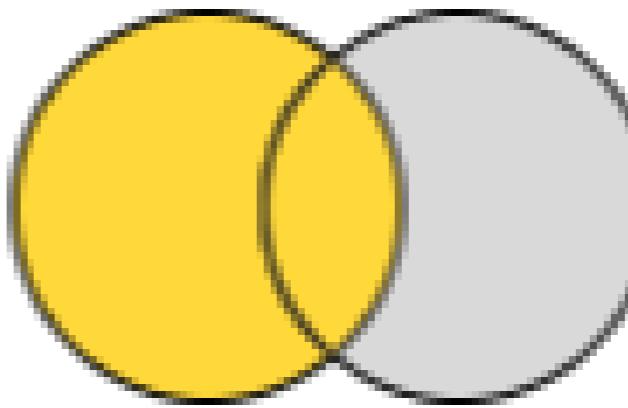


Merging with Joins

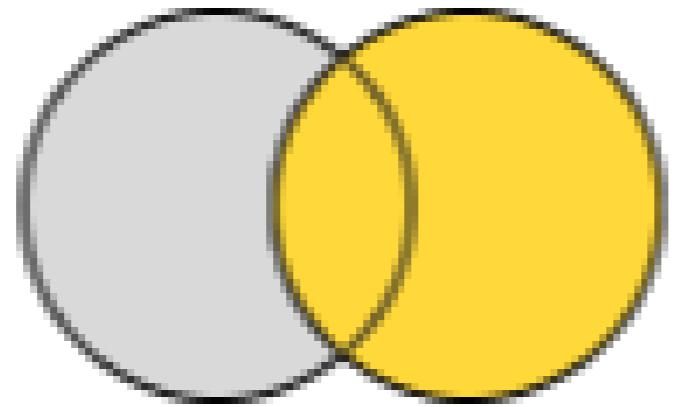
Inner Join



Left Outer Join

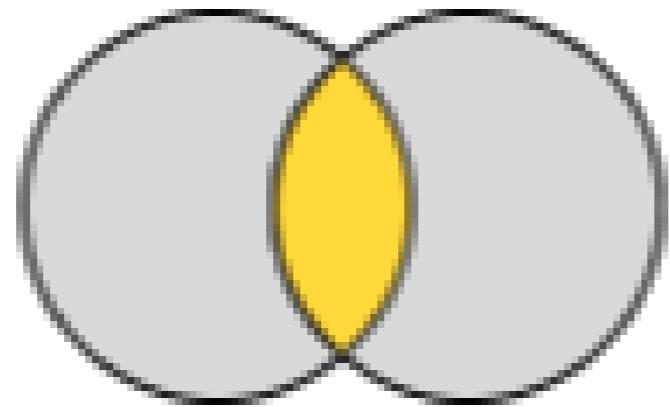


Right Outer Join

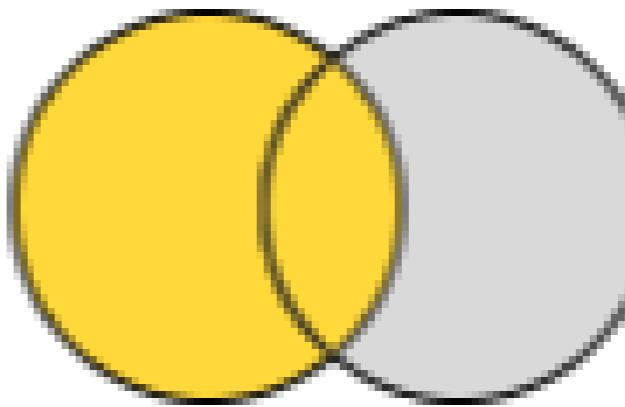


Merging with Joins

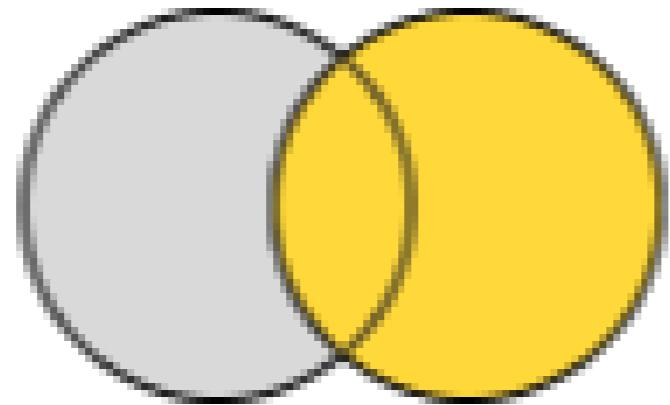
Inner Join



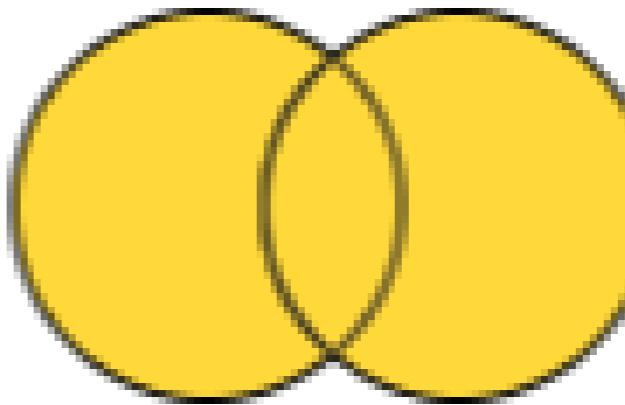
Left Outer Join



Right Outer Join



Full Outer Join



Let's practice

DATA MANIPULATION IN KNIME

Hands-on data merging in KNIME

DATA MANIPULATION IN KNIME



Let's practice!

DATA MANIPULATION IN KNIME

Introduction to Aggregating and Pivoting

DATA MANIPULATION IN KNIME



Introduction to Aggregation and Pivoting



Introduction to Aggregation and Pivoting



Introduction to Aggregation and Pivoting



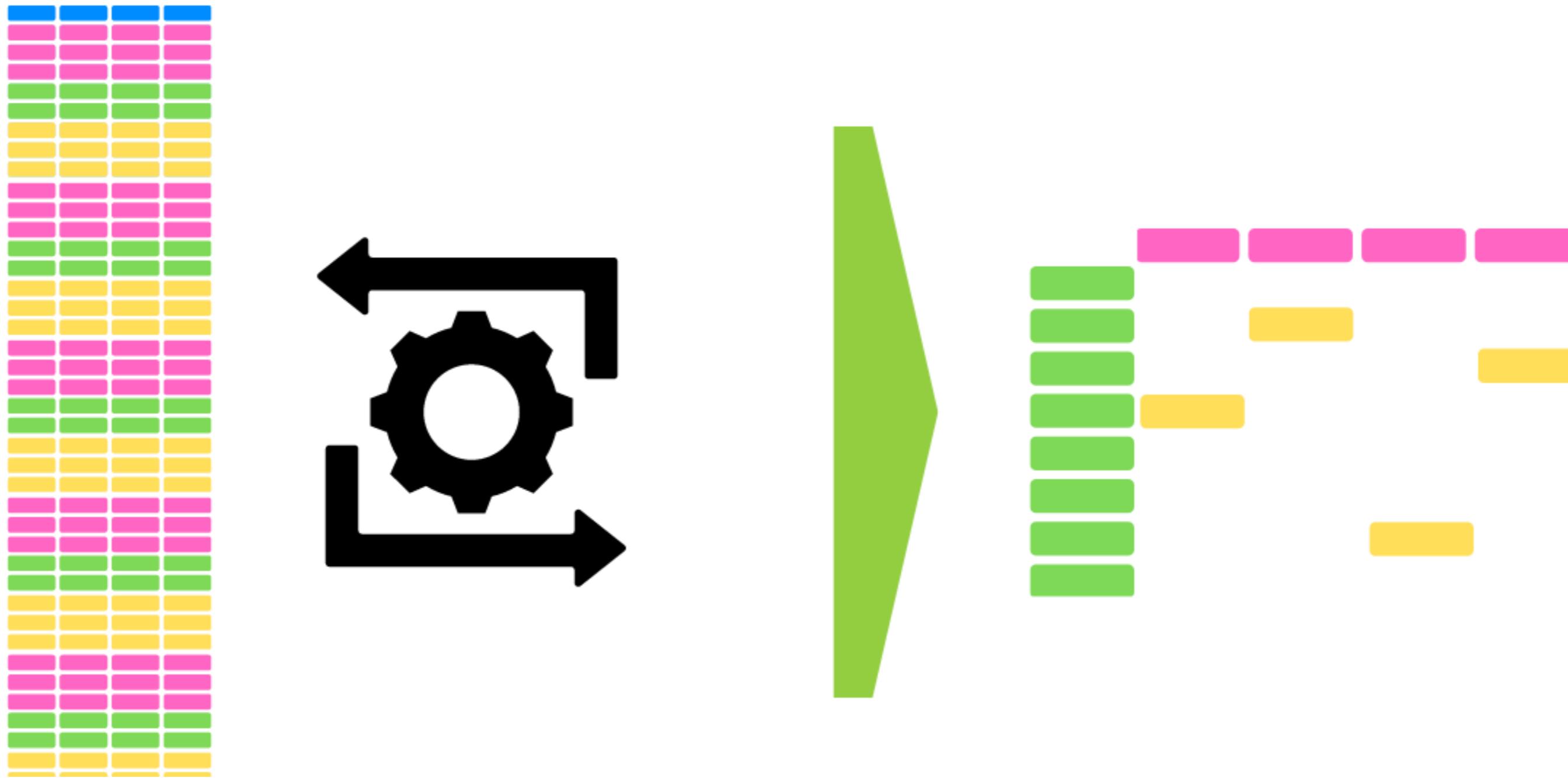
Introduction to Aggregation and Pivoting



Introduction to Aggregation and Pivoting

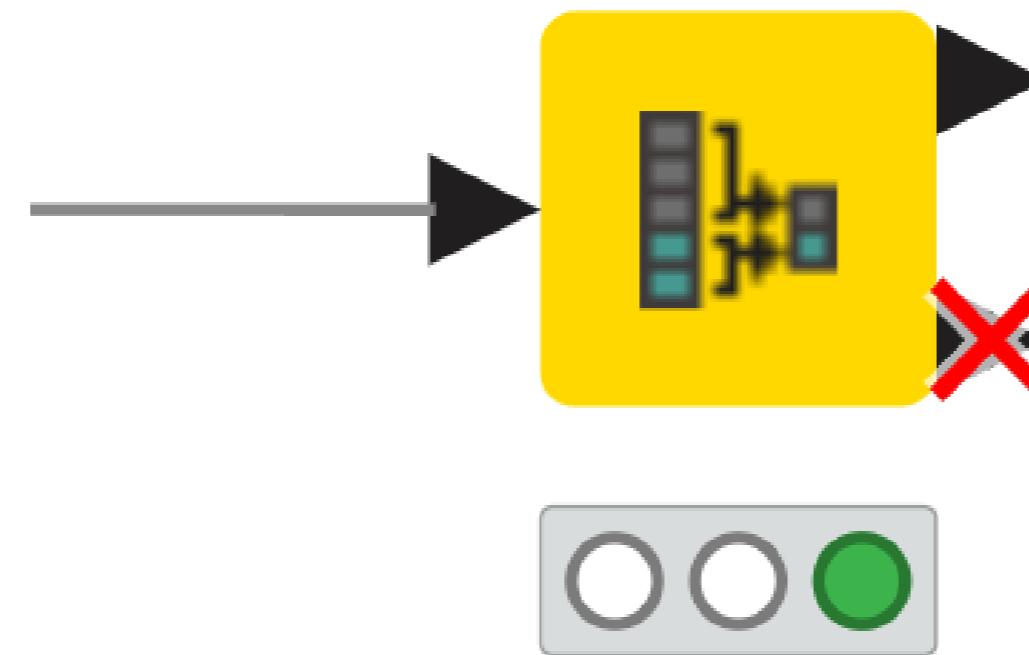


Introductions to Aggregation and Pivoting



Simple aggregations with the Row Aggregator

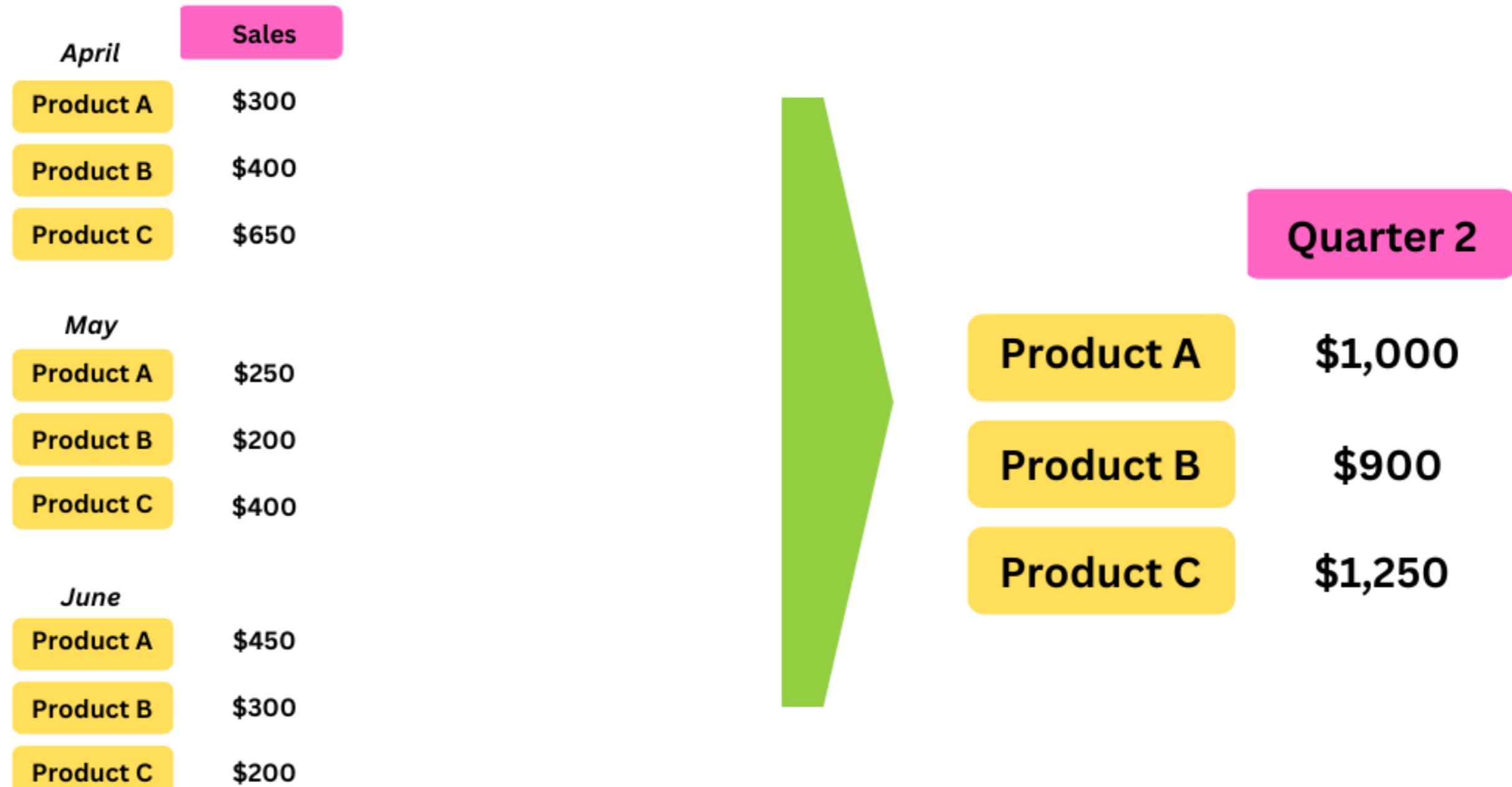
Row Aggregator



Simple aggregations with the Row Aggregator

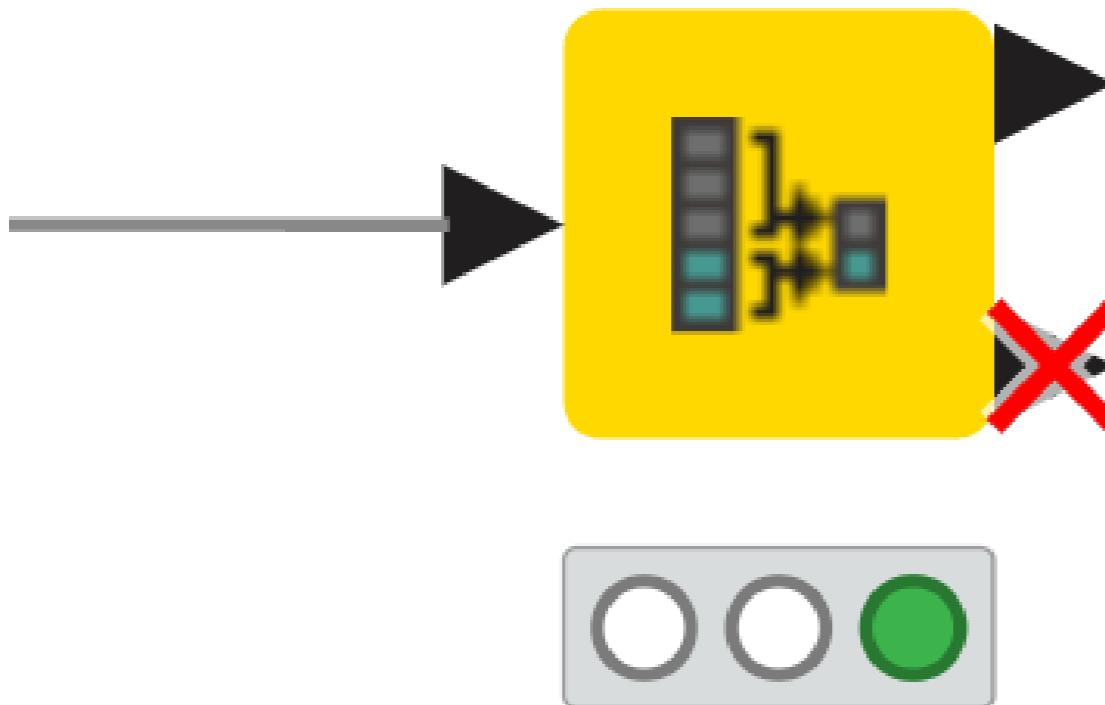
Sales	
<i>April</i>	
Product A	\$300
Product B	\$400
Product C	\$650
<i>May</i>	
Product A	\$250
Product B	\$200
Product C	\$400
<i>June</i>	
Product A	\$450
Product B	\$300
Product C	\$200

Simple aggregations with the Row Aggregator



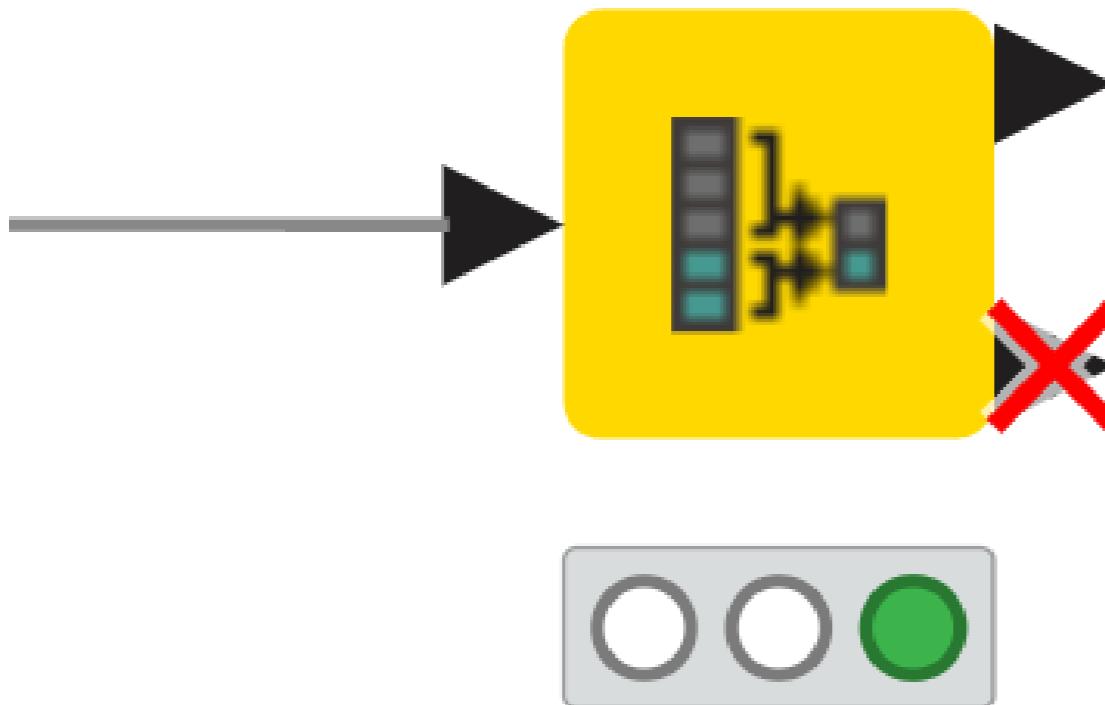
Simple aggregations with the Row Aggregator

Row Aggregator



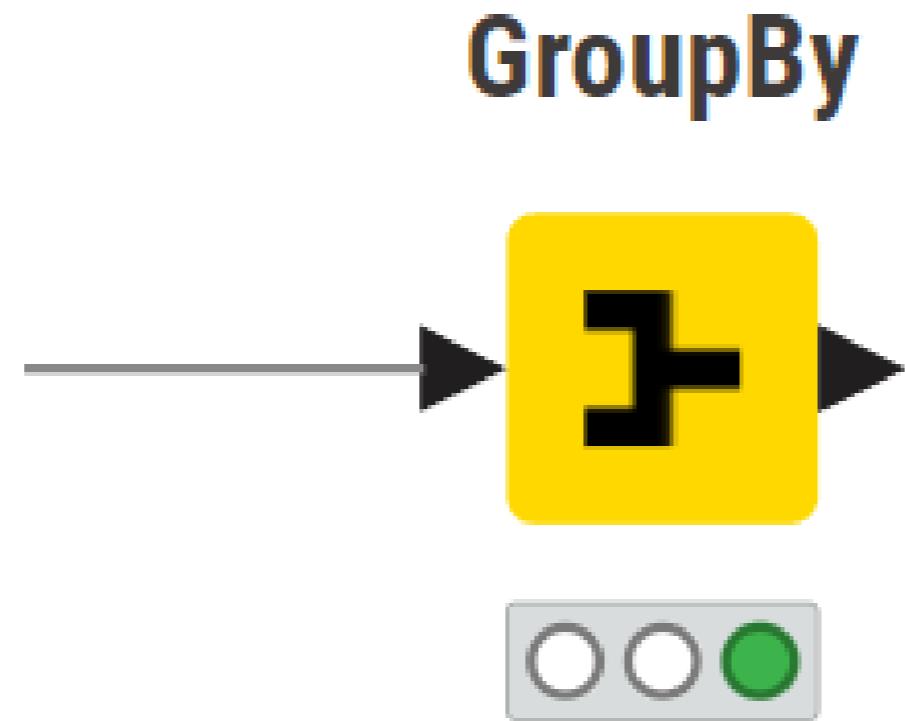
Simple aggregations with the Row Aggregator

Row Aggregator



- (Occurrence) Count
- Sum
- Average
- Minimum
- Maximum

Multi-level aggregations with GroupBy

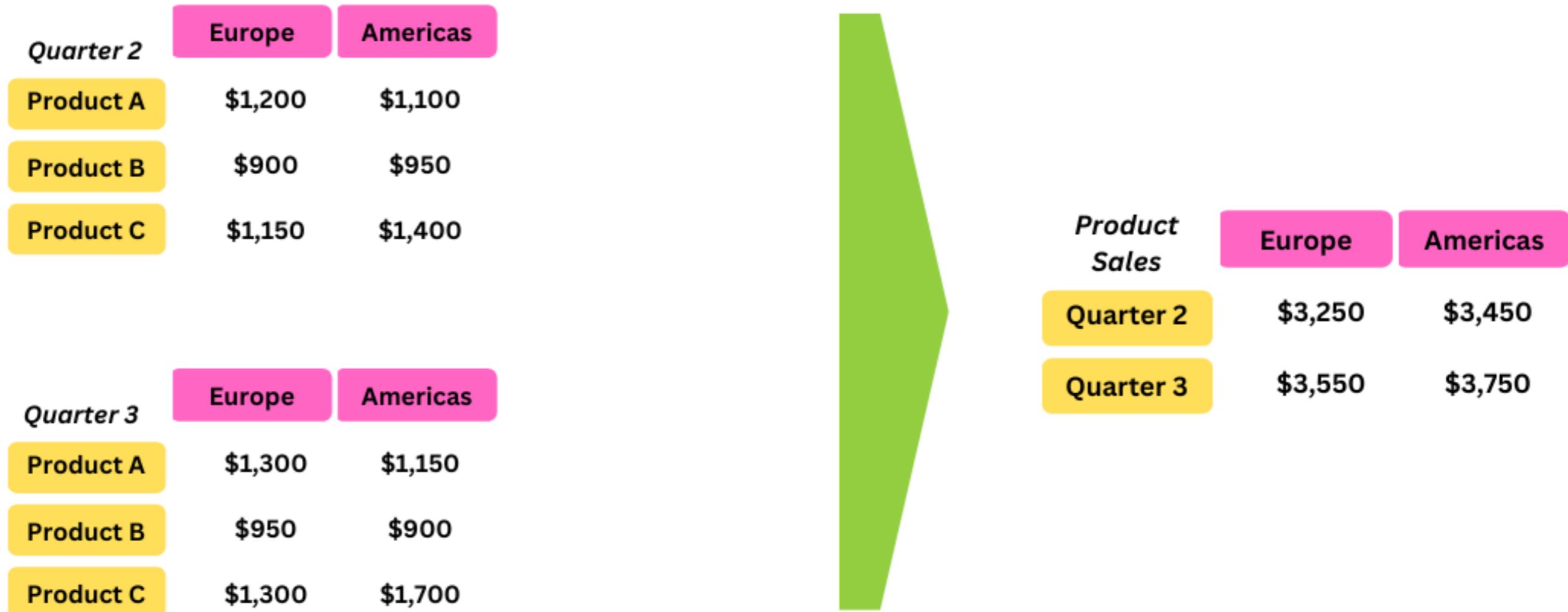


Multi-level aggregations with GroupBy

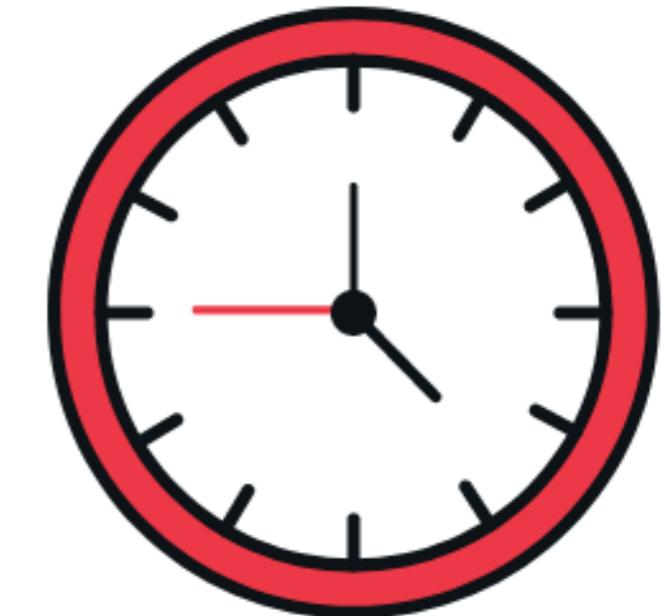
Quarter 2	Europe	Americas
	Product A	\$1,200
Product B	\$900	\$950
Product C	\$1,150	\$1,400

Quarter 3	Europe	Americas
	Product A	\$1,300
Product B	\$950	\$900
Product C	\$1,300	\$1,700

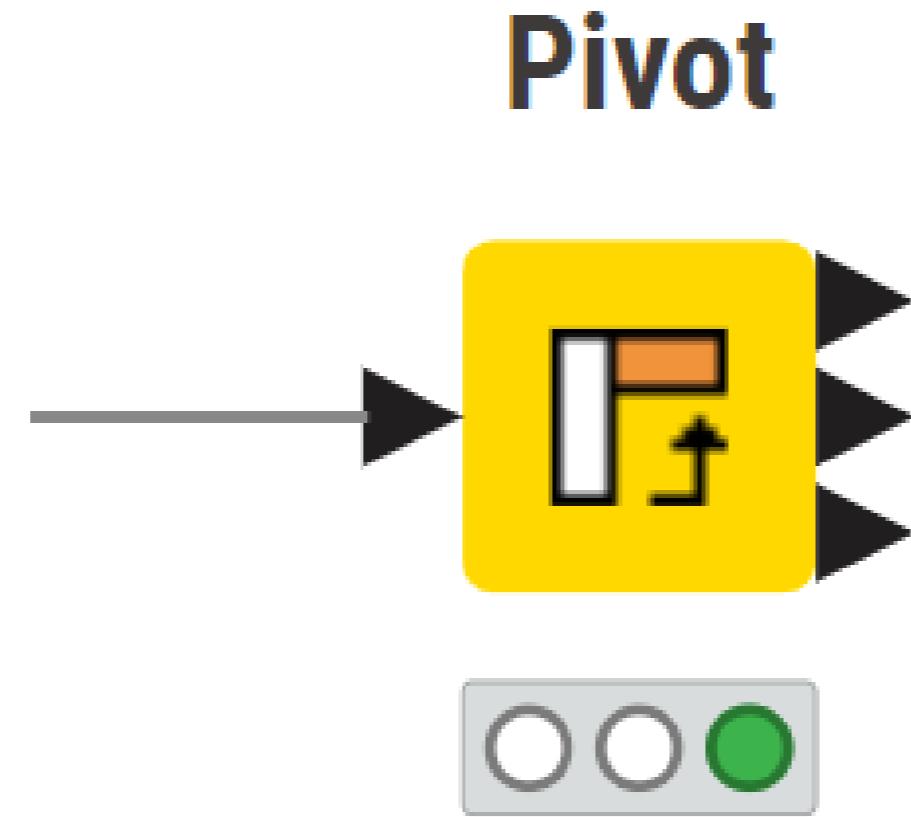
Multi-level aggregations with GroupBy



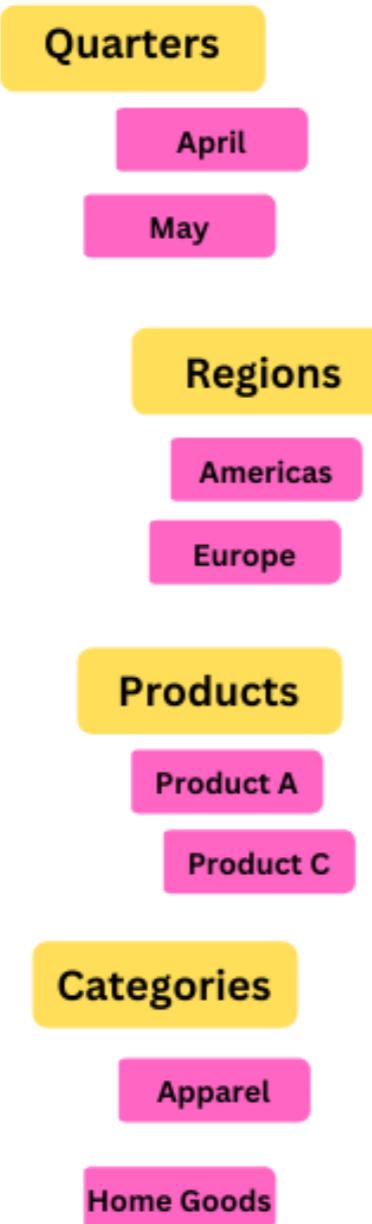
Multi-level aggregations with GroupBy



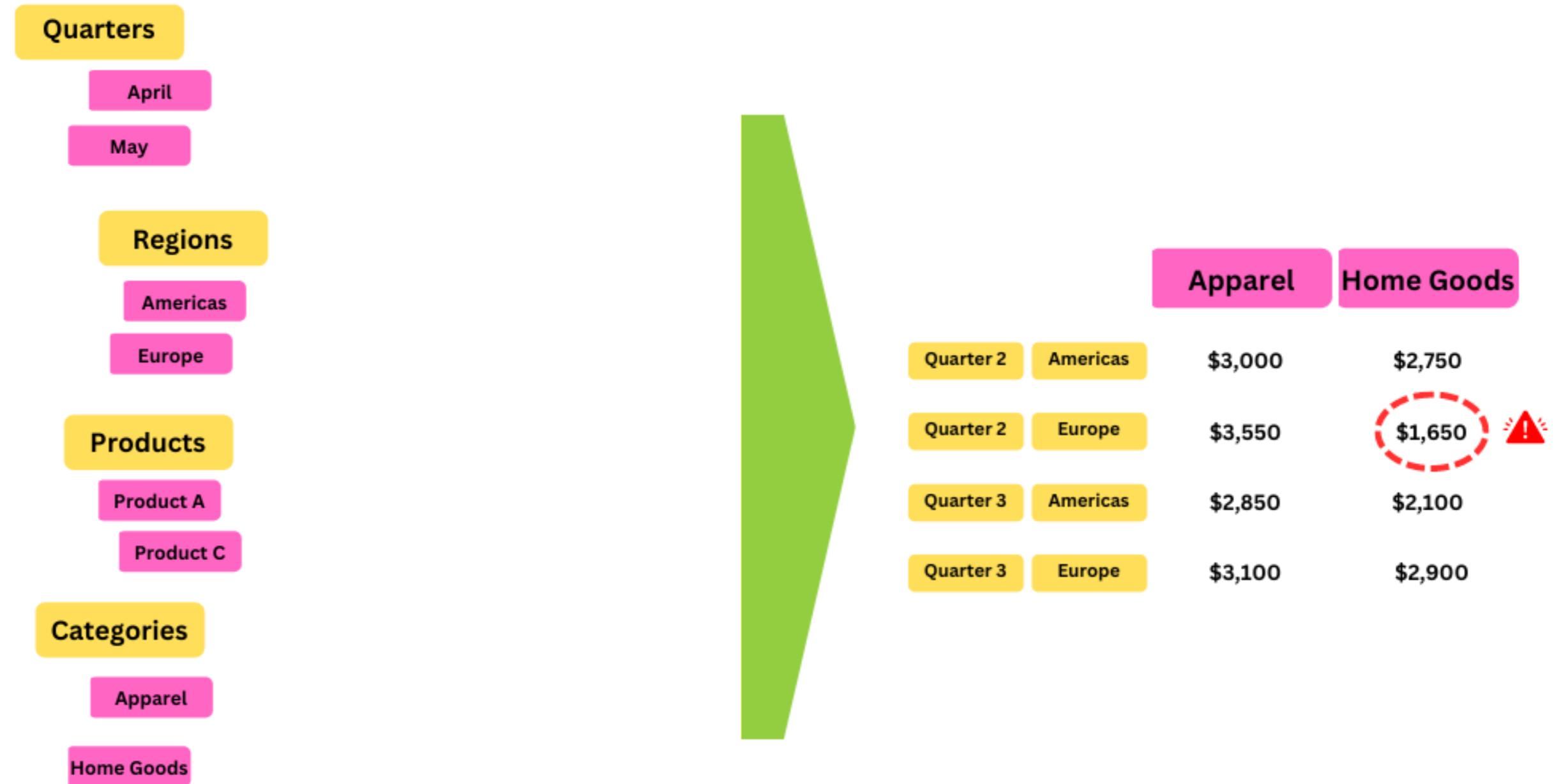
Complex aggregations with the Pivot node



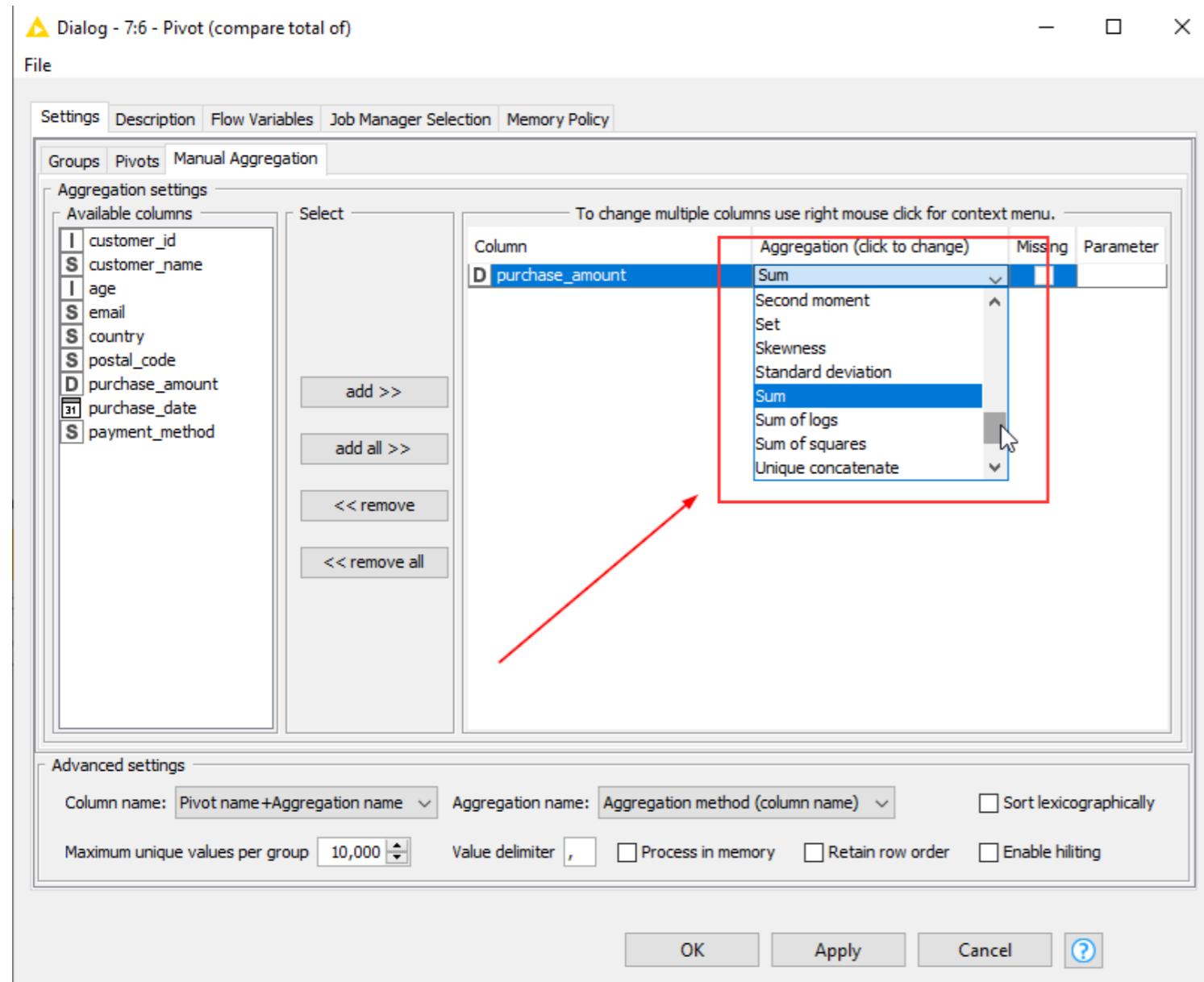
Complex aggregations with the Pivot node



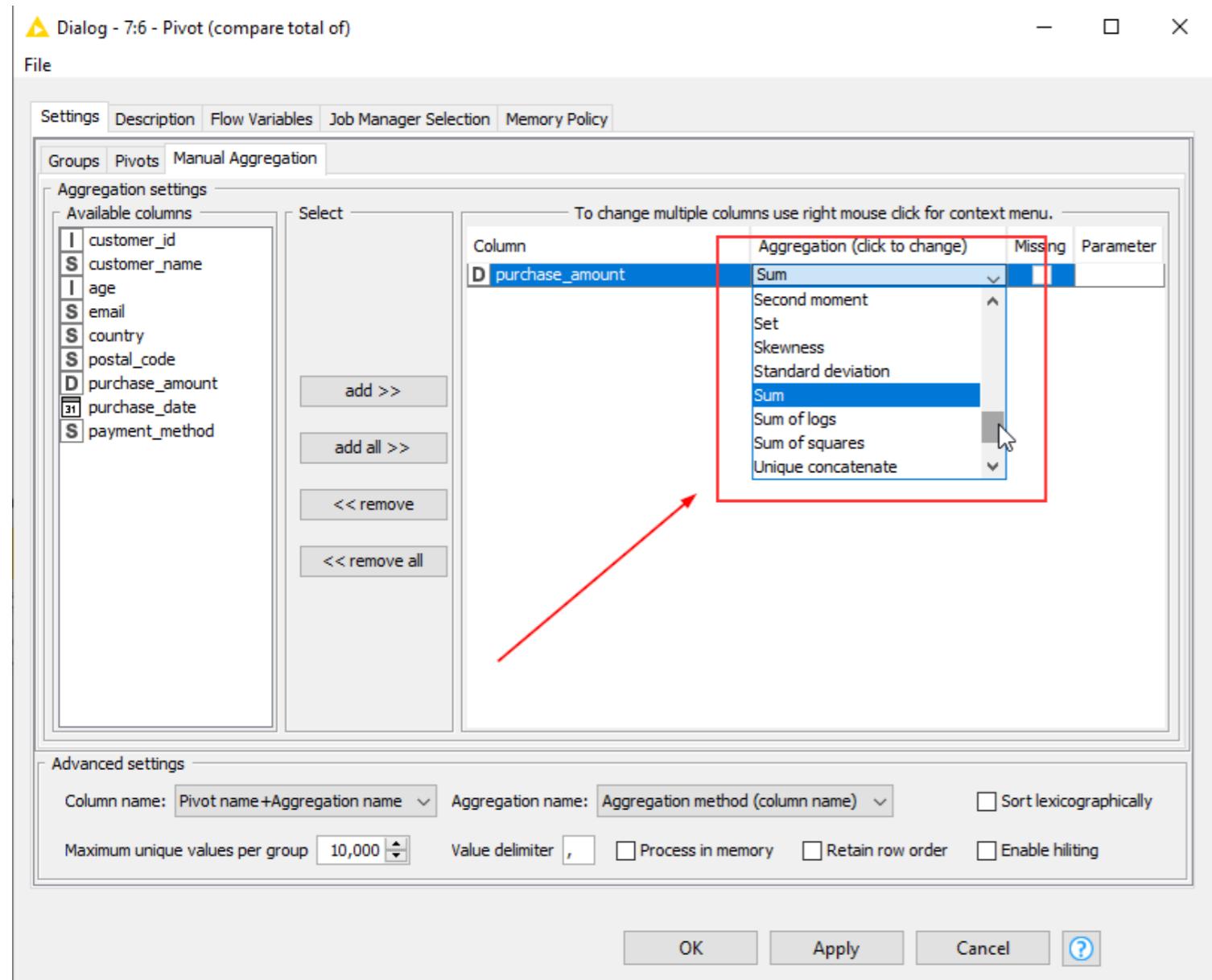
Complex aggregations with the Pivot node



Complex aggregations with the Pivot node



Complex aggregations with the Pivot node

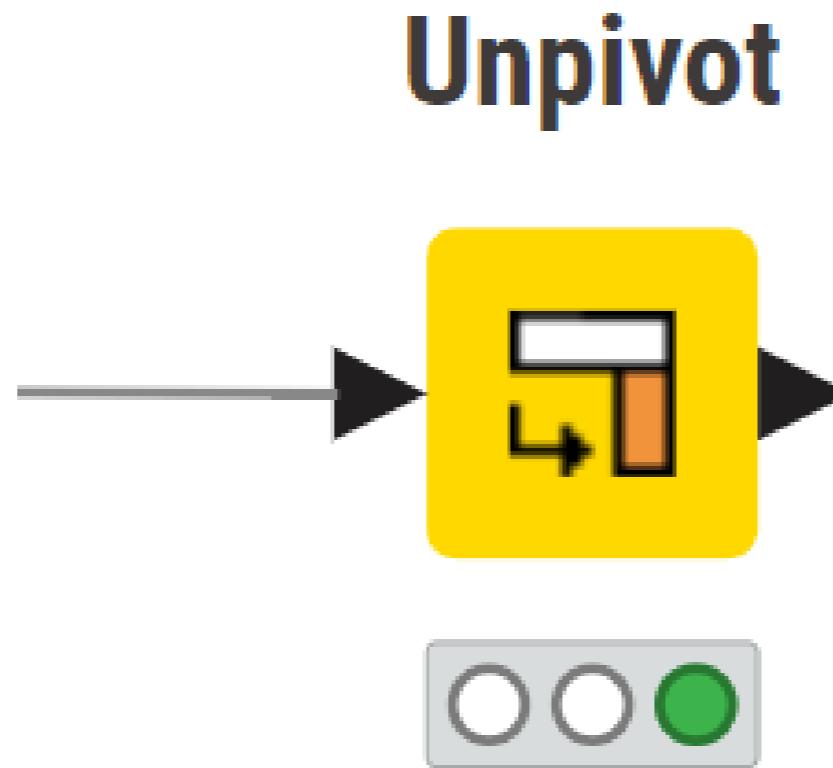


Pivot table (Table)

Rows: 12 | Columns: 5

#	RowID	Month (number) Number (integer)	Month (Name) String	clothing+Sum(purch... Number (double))	electronics+Sum(pur... Number (double))	home goods+Sum(p... Number (double))
1	Row0	1	January	15,056.61	11,530.9	12,694.35
2	Row1	2	February	14,152.12	13,255.38	9,814.33
3	Row2	3	March	13,921.61	12,824.92	18,216.98
4	Row3	4	April	16,525.78	21,793.2	15,475.6
5	Row4	5	May	11,157.18	8,532.4	17,259.38
6	Row5	6	June	18,965.8	10,650.66	18,665.12
7	Row6	7	July	13,720.78	16,602.19	9,844.59
8	Row7	8	August	12,935.55	14,563.27	11,741.15
9	Row8	9	September	14,185.79	13,779.06	14,199.38
10	Row9	10	October	15,408.02	12,660.83	14,614.42
11	Row...	11	November	10,713.58	14,392.62	14,636.52
12	Row...	12	December	13,187.37	15,329.57	14,057.04

Unpivoting data



Unpivoting data

		Apparel	Home Goods
Quarter	Region		
Quarter 2	Americas	\$3,000	\$2,750
Quarter 2	Europe	\$3,550	\$1,650 
Quarter 3	Americas	\$2,850	\$2,100
Quarter 3	Europe	\$3,100	\$2,900

Unpivoting data

		Apparel	Home Goods
Quarter	Region		
Quarter 2	Americas	\$3,000	\$2,750
Quarter 2	Europe	\$3,550	\$1,650 
Quarter 3	Americas	\$2,850	\$2,100
Quarter 3	Europe	\$3,100	\$2,900



Quarter	Region	Category	Sales
Quarter 2	Americas	Apparel	\$3,000
Quarter 3	Americas	Apparel	\$2,850
Quarter 2	Europe	Apparel	\$3,550
Quarter 3	Europe	Apparel	\$3,100
Quarter 2	Americas	Home Goods	\$2,750
Quarter 3	Americas	Home Goods	\$2,100
Quarter 2	Europe	Home Goods	\$1,650
Quarter 3	Europe	Home Goods	\$2,900

Let's practice!

DATA MANIPULATION IN KNIME

Aggregate data with GroupBy and Pivot nodes

DATA MANIPULATION IN KNIME



Let's practice!

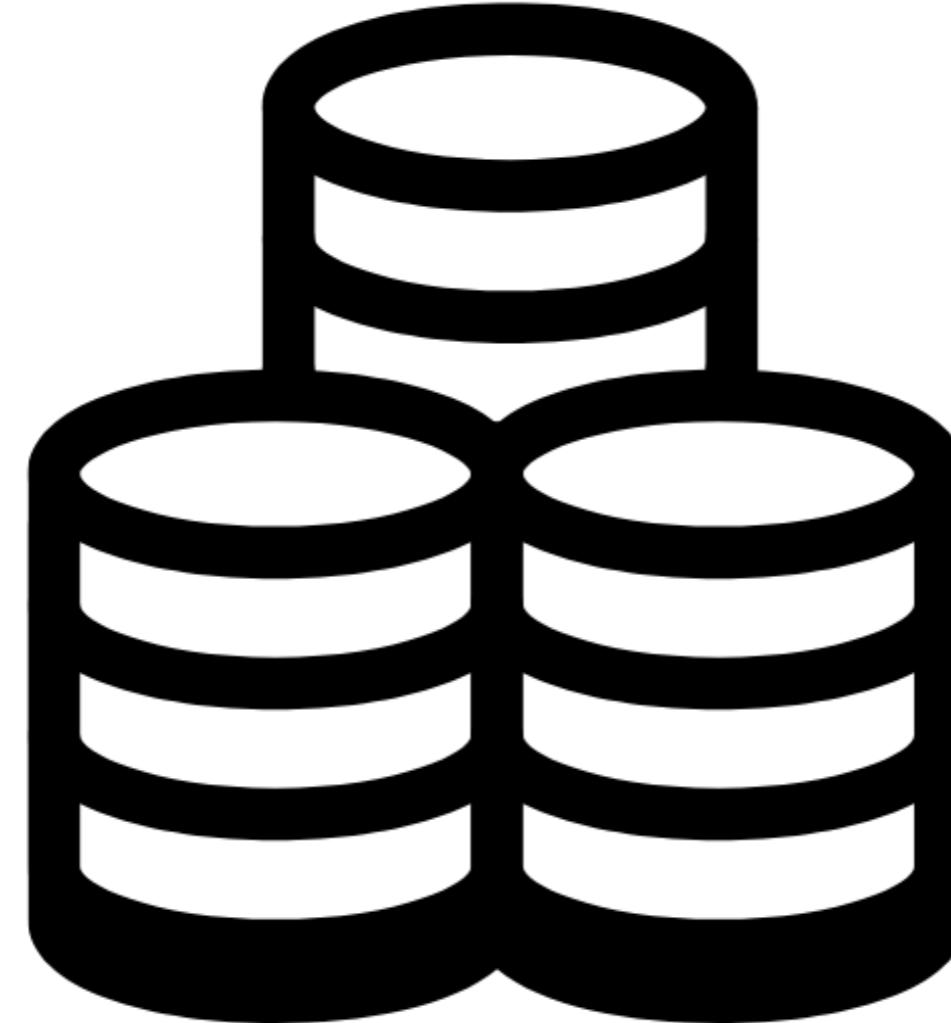
DATA MANIPULATION IN KNIME

Advanced data reading and writing

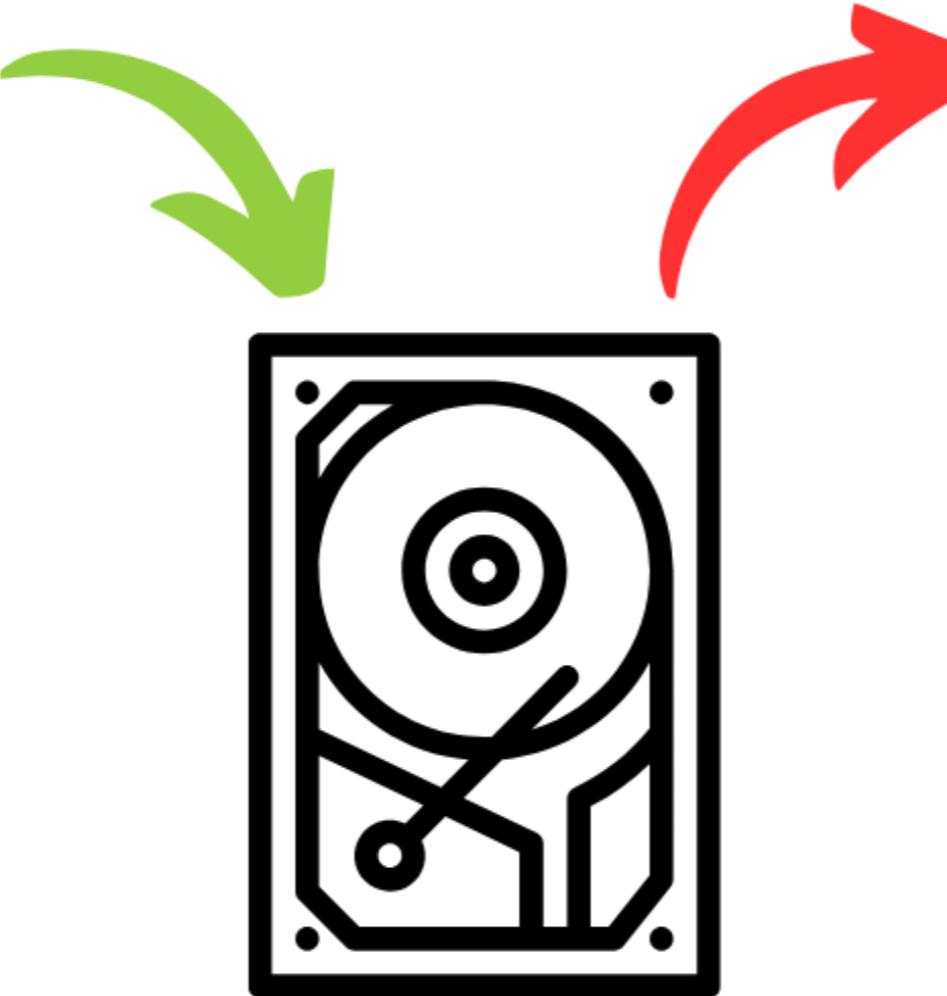
DATA MANIPULATION IN KNIME



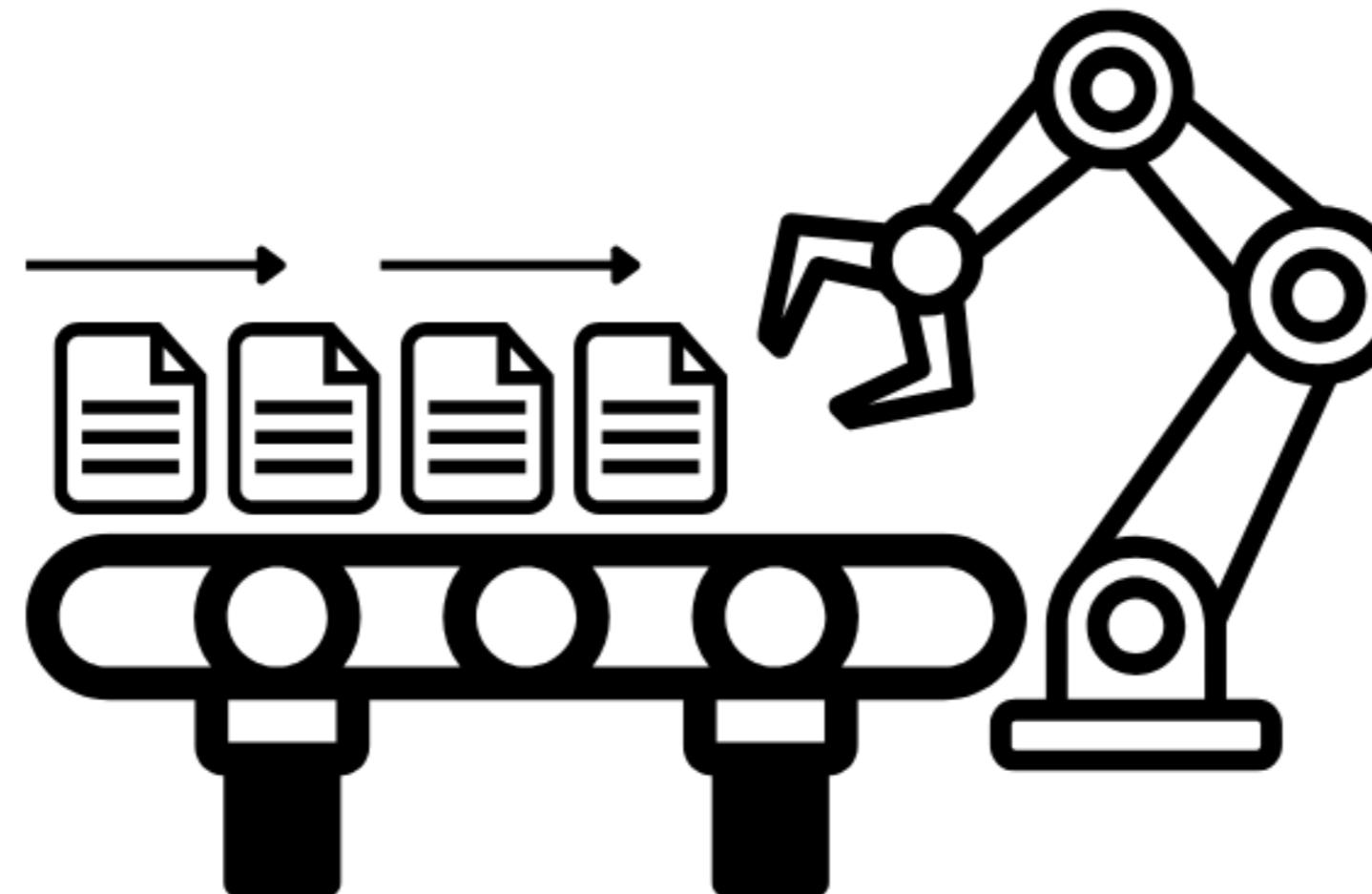
Automating file handling with KNIME



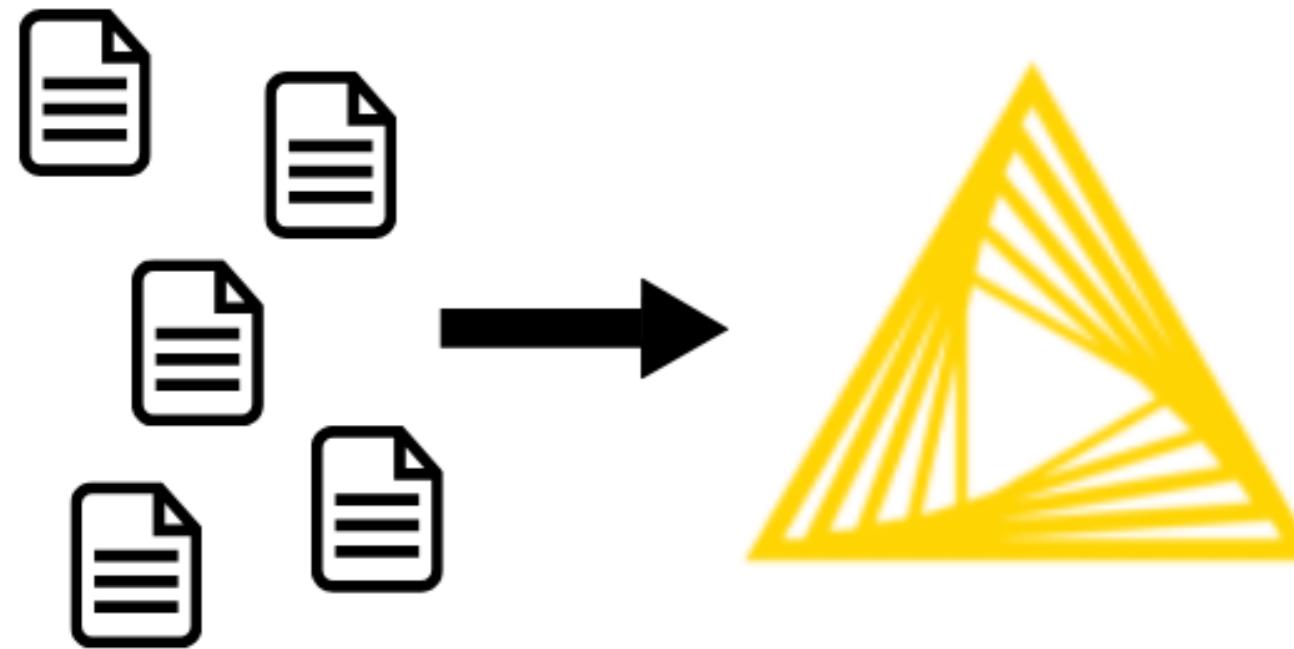
Automating file handling with KNIME



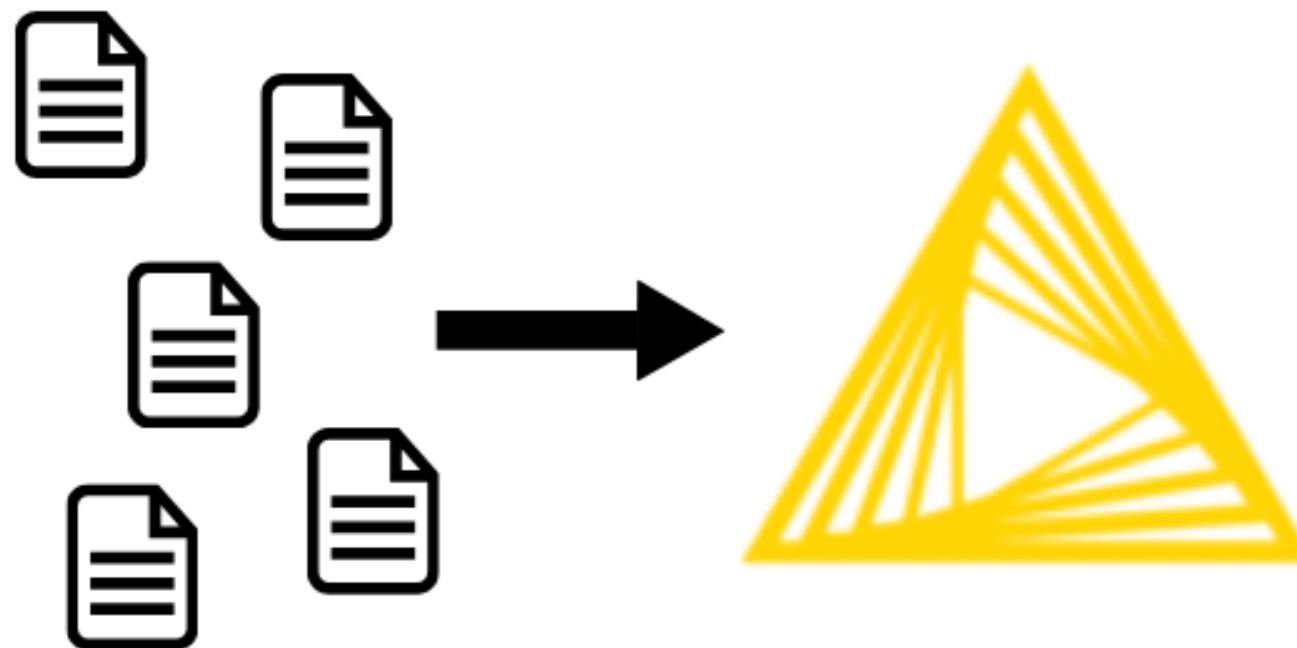
Automating file handling with KNIME



Working with multiple files



Working with multiple files

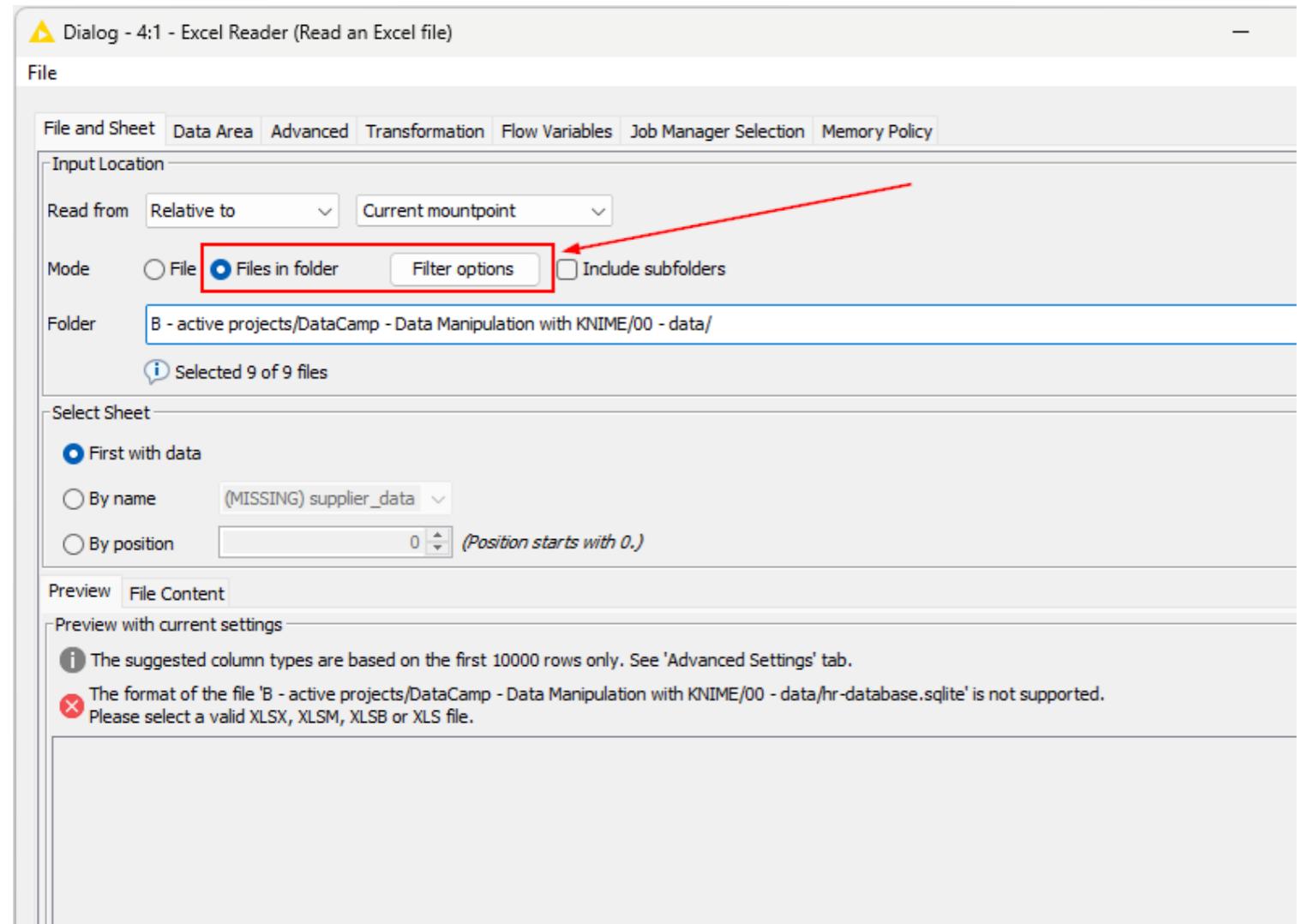


Excel Reader

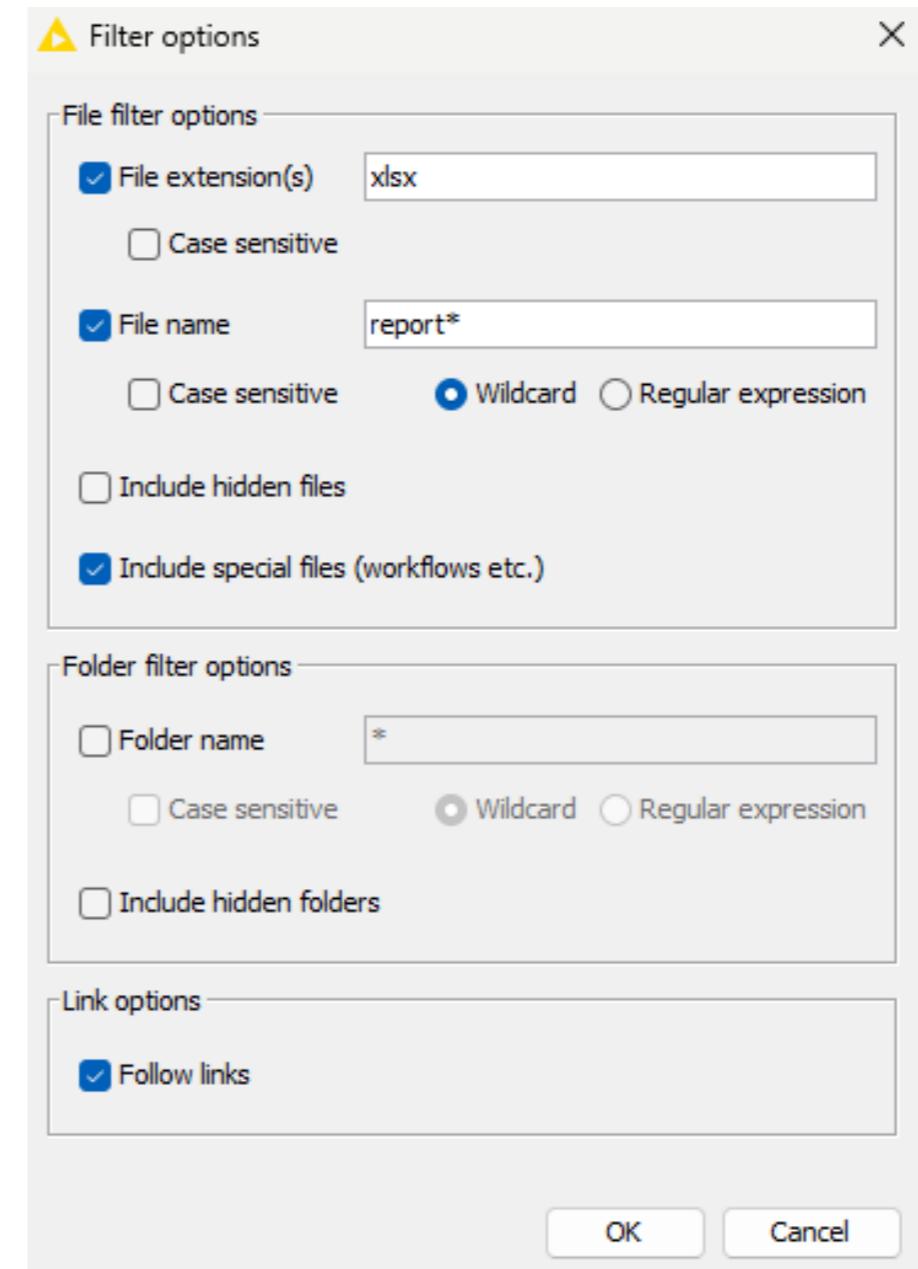
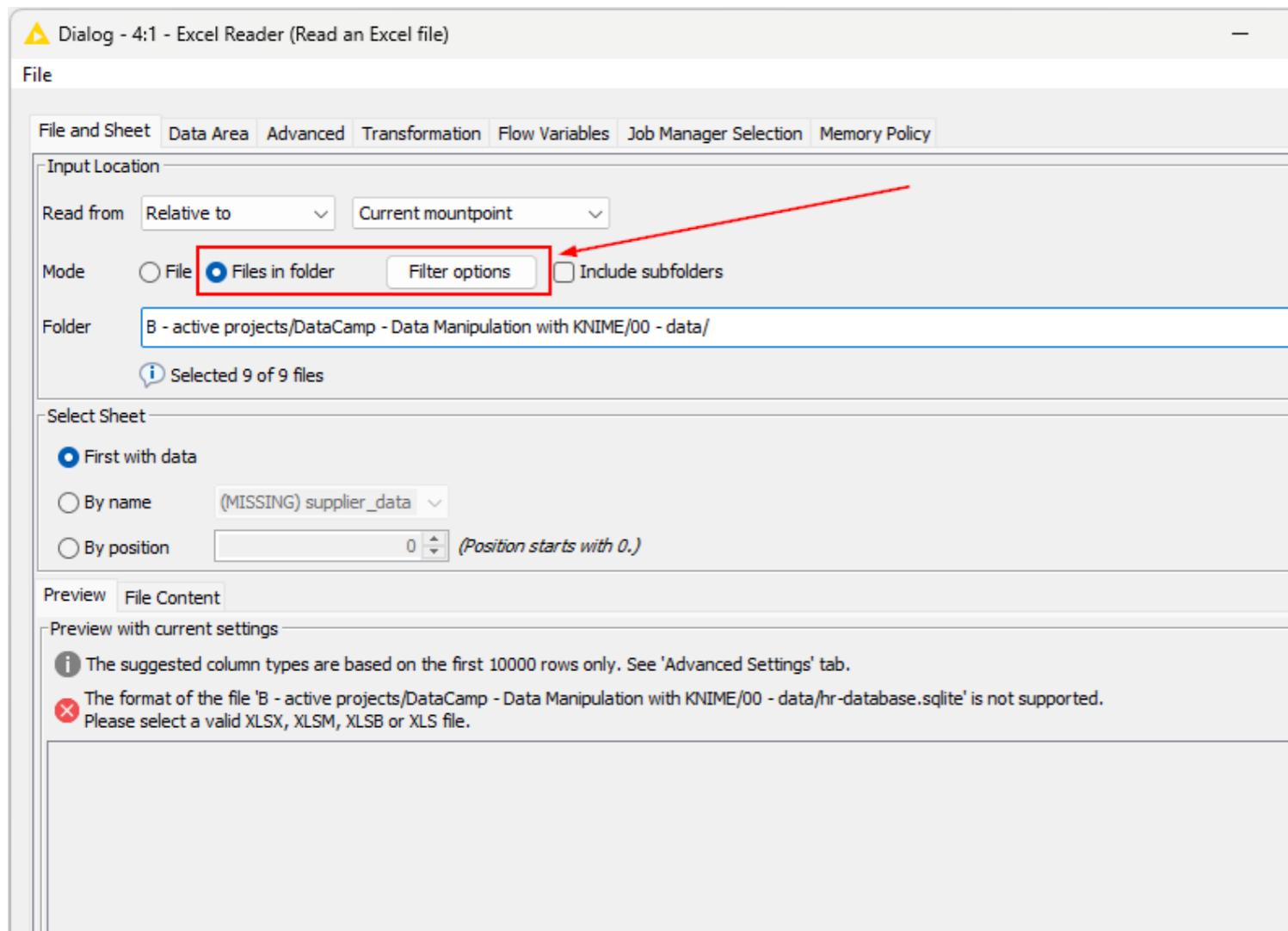


Read an Excel file

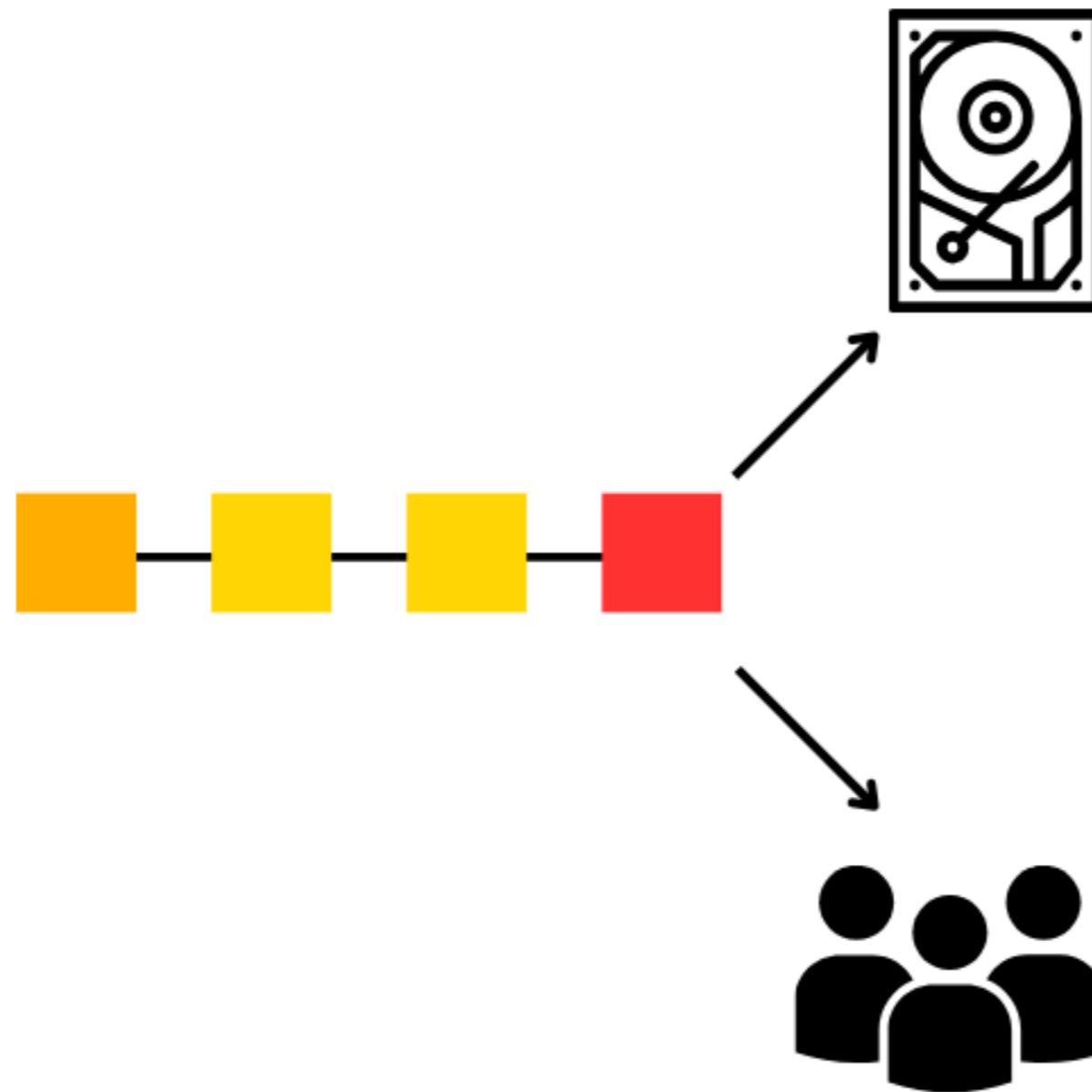
Working with multiple files



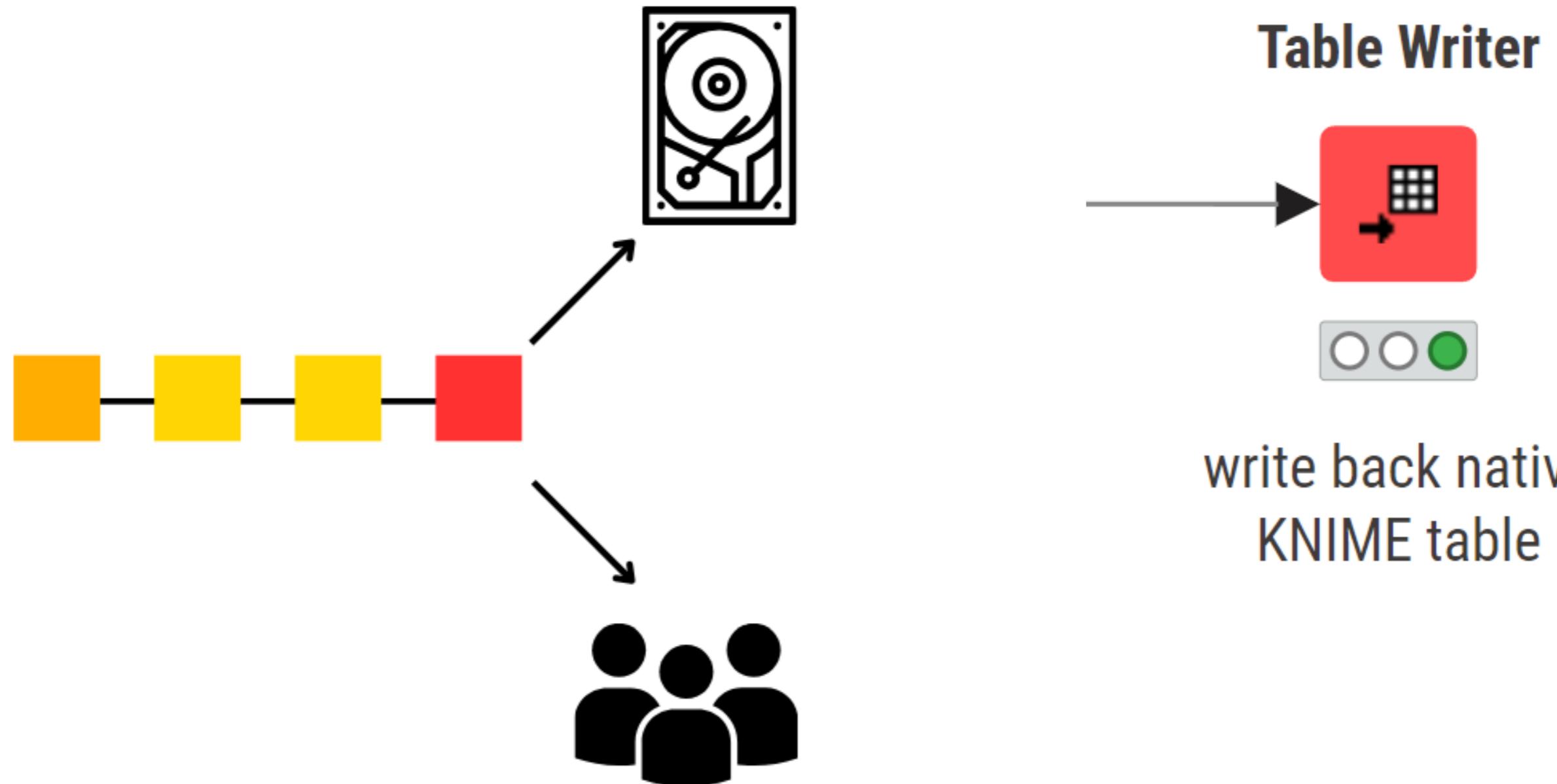
Working with multiple files



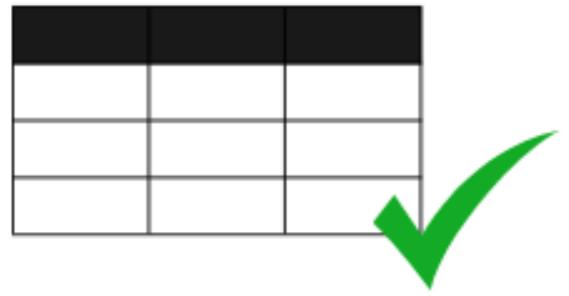
Writing data to external files



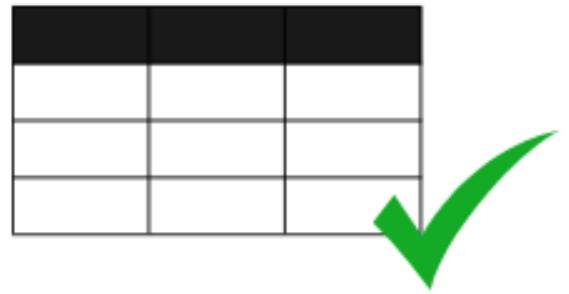
Writing data to external files



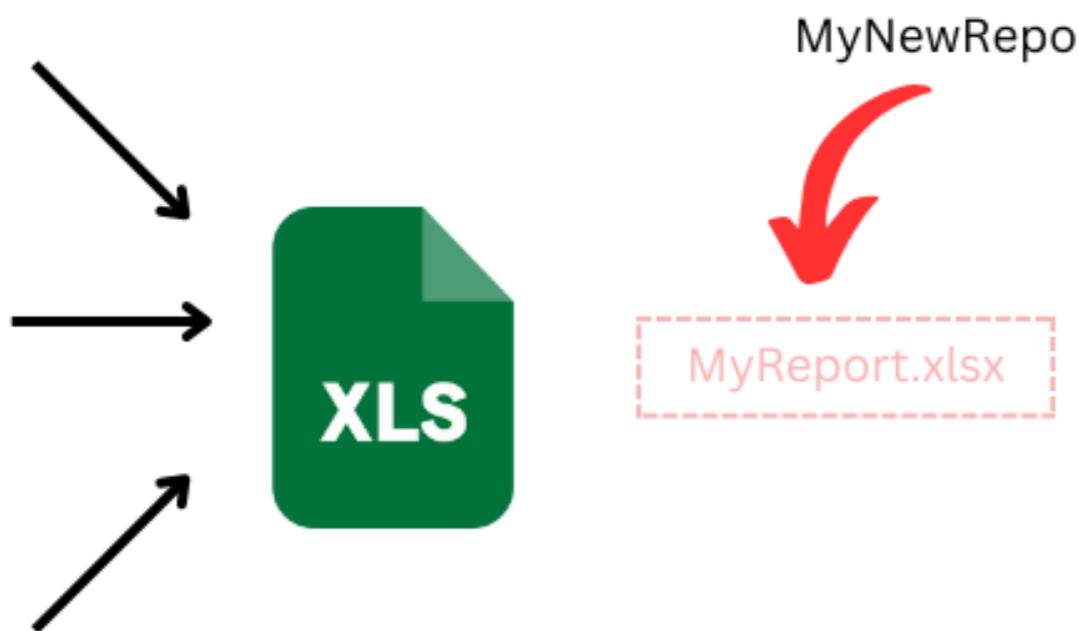
Writing data to external files



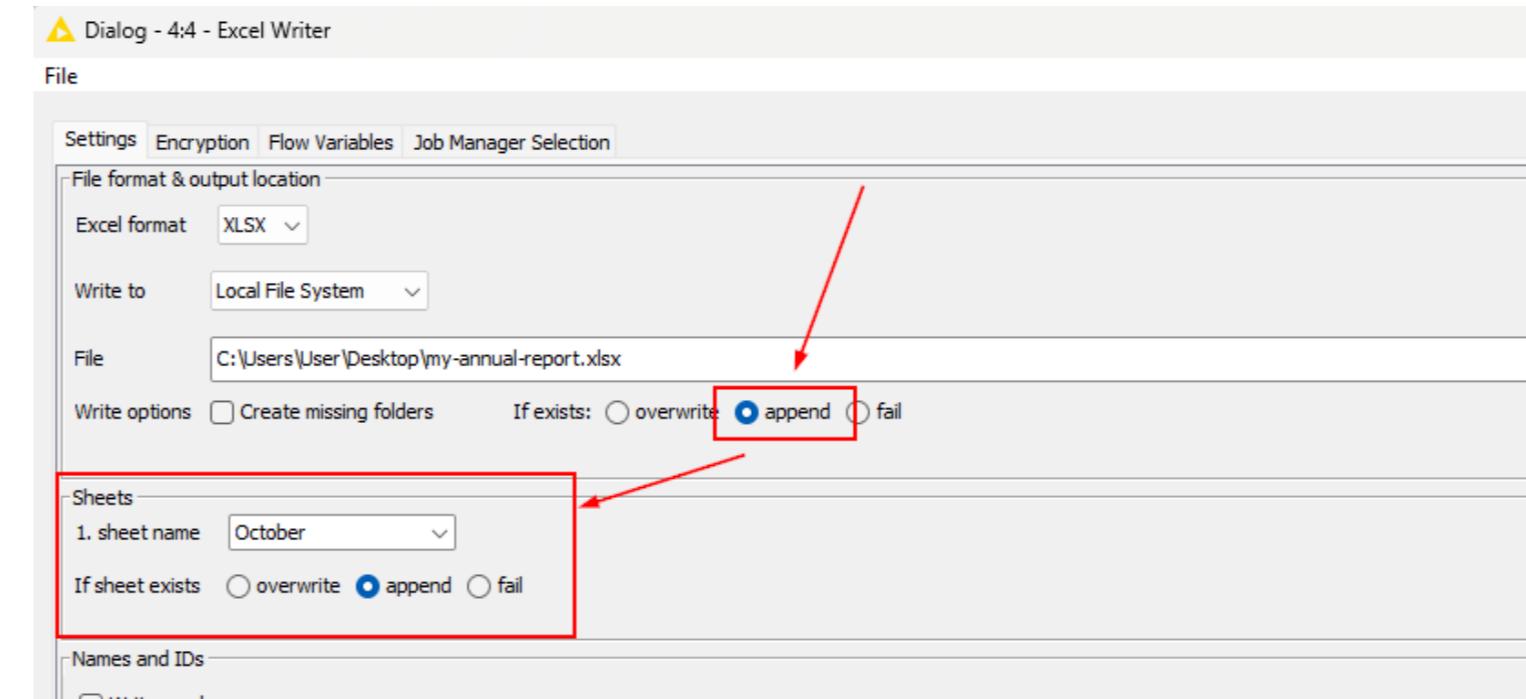
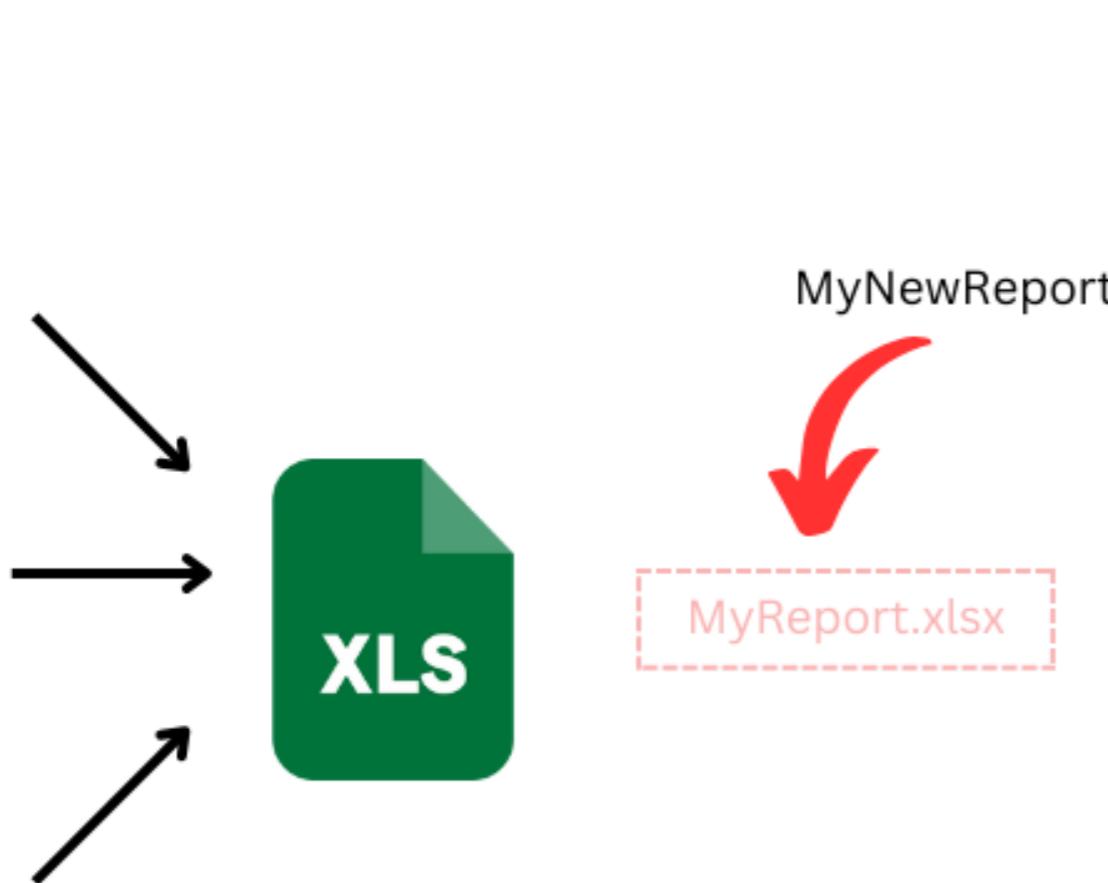
Writing data to external files



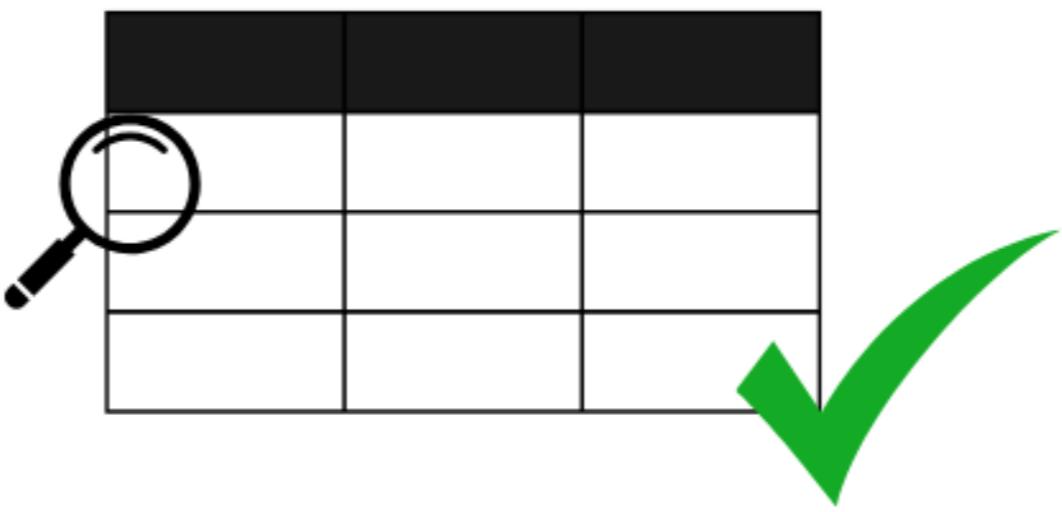
Writing data to external files



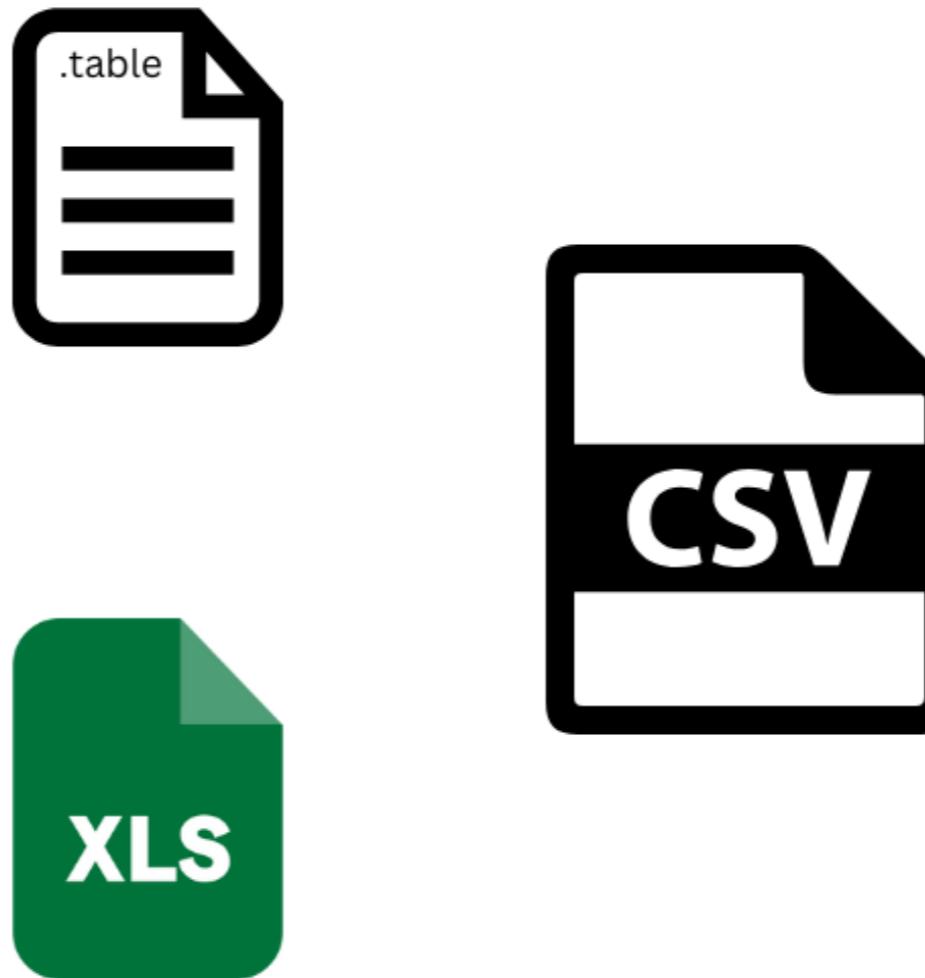
Writing data to external files



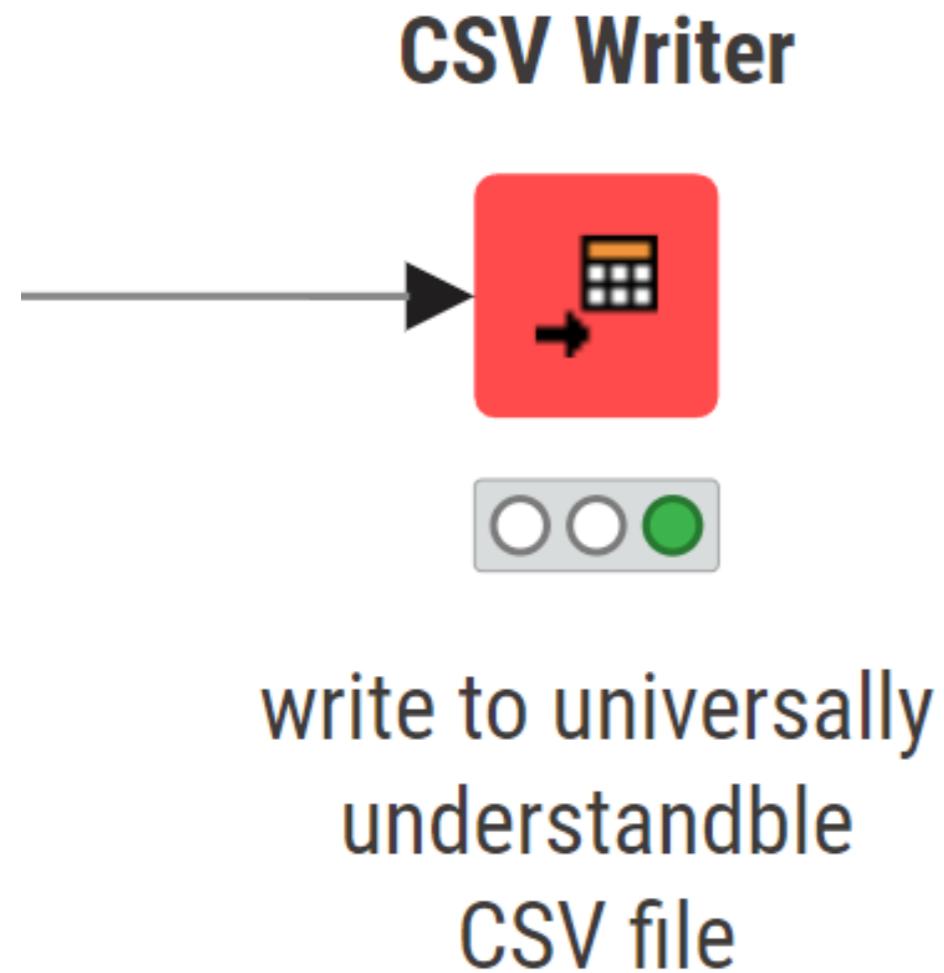
Writing data to external files



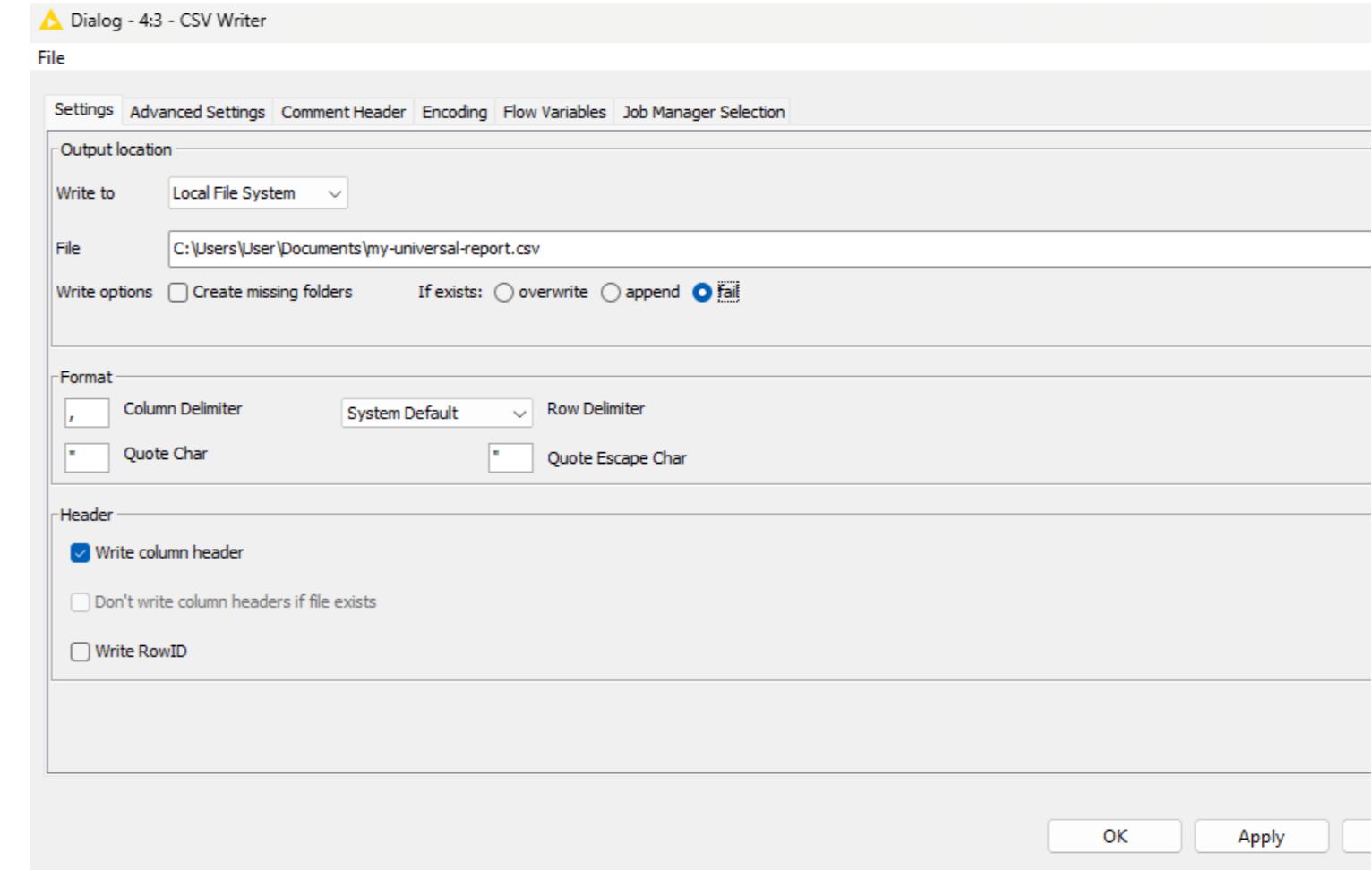
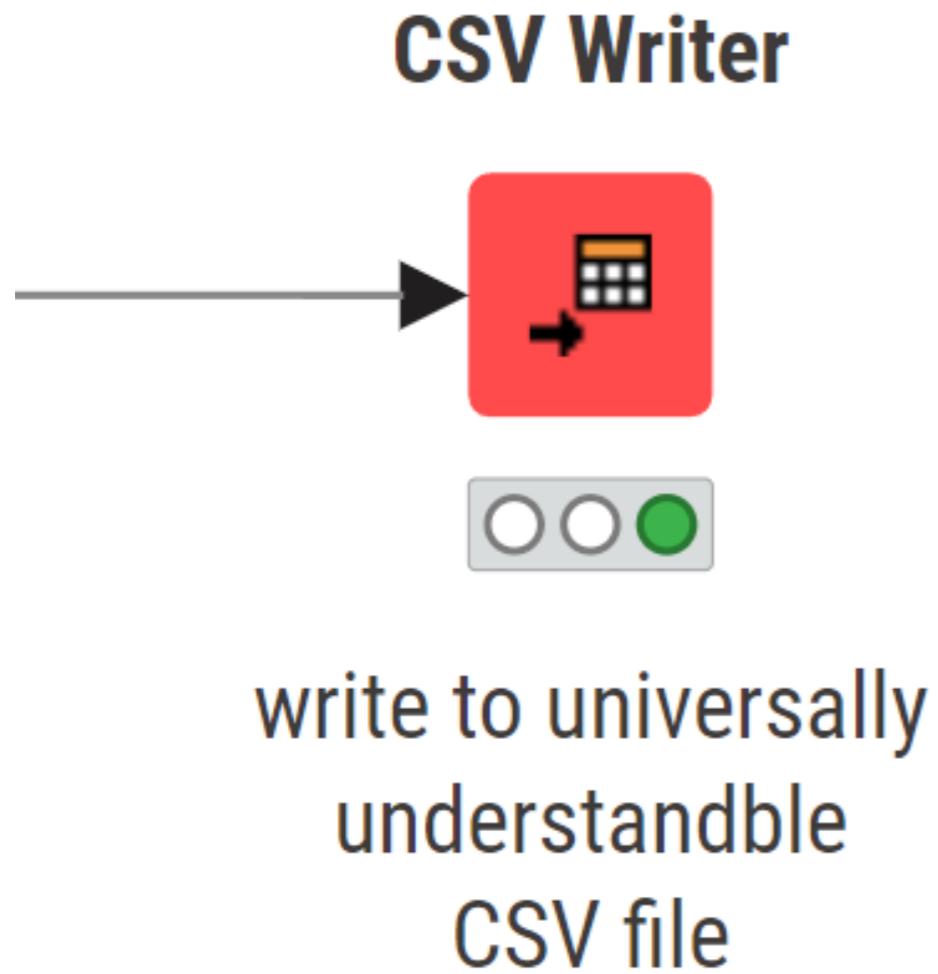
Exporting data to standard formats



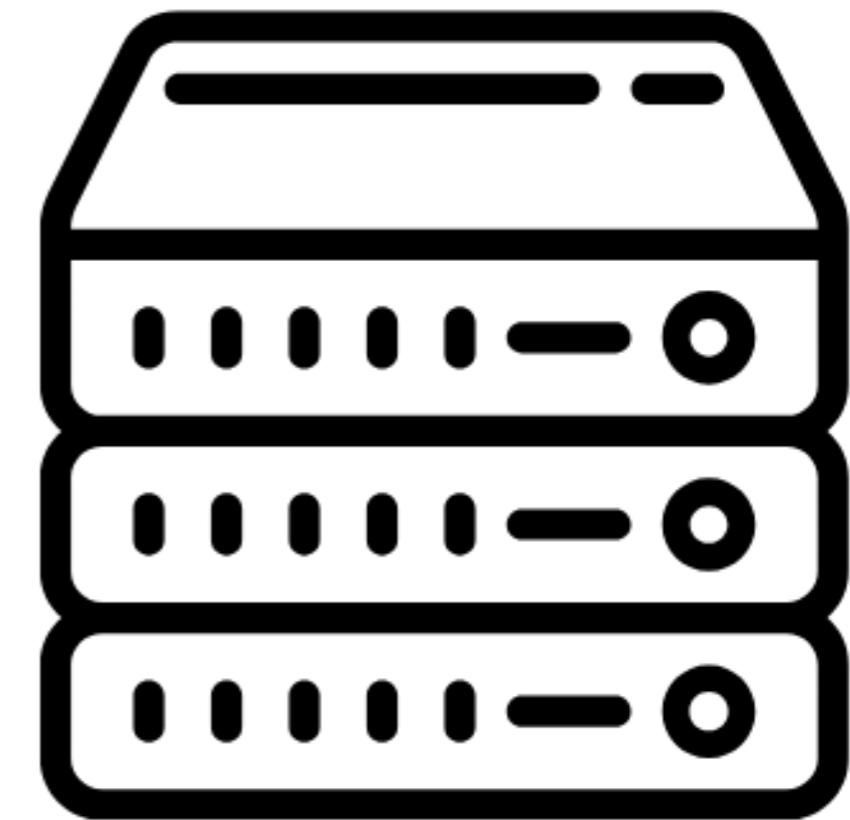
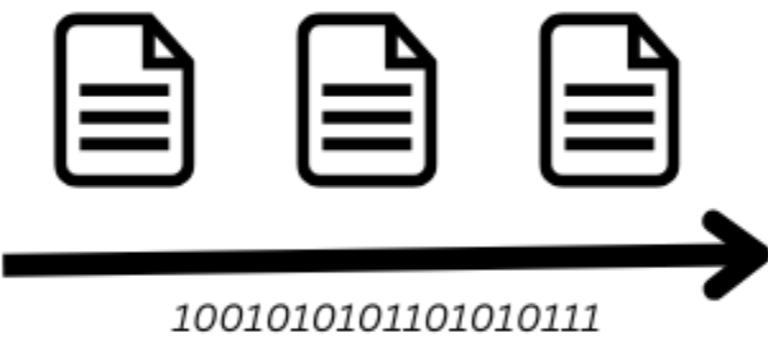
Exporting data to standard formats



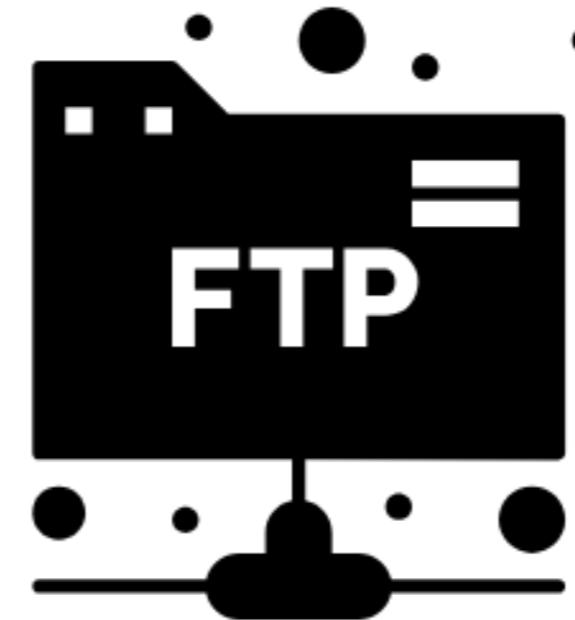
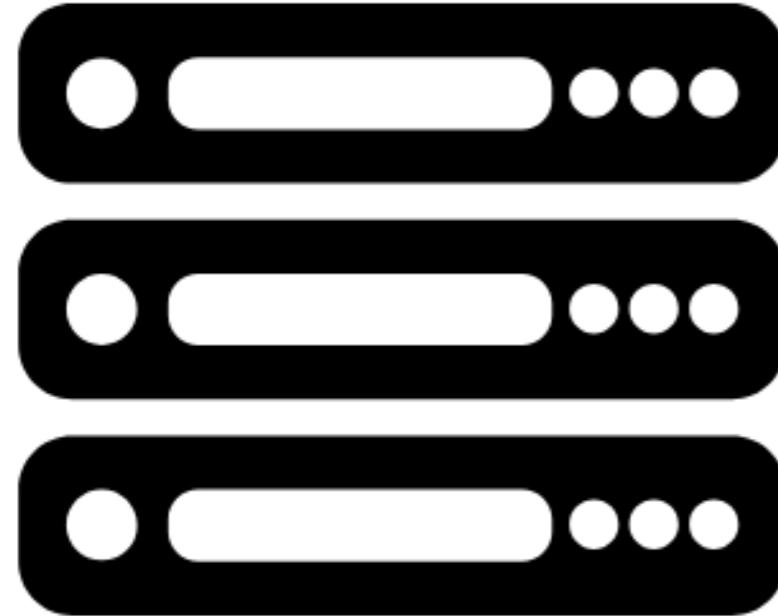
Exporting data to standard formats



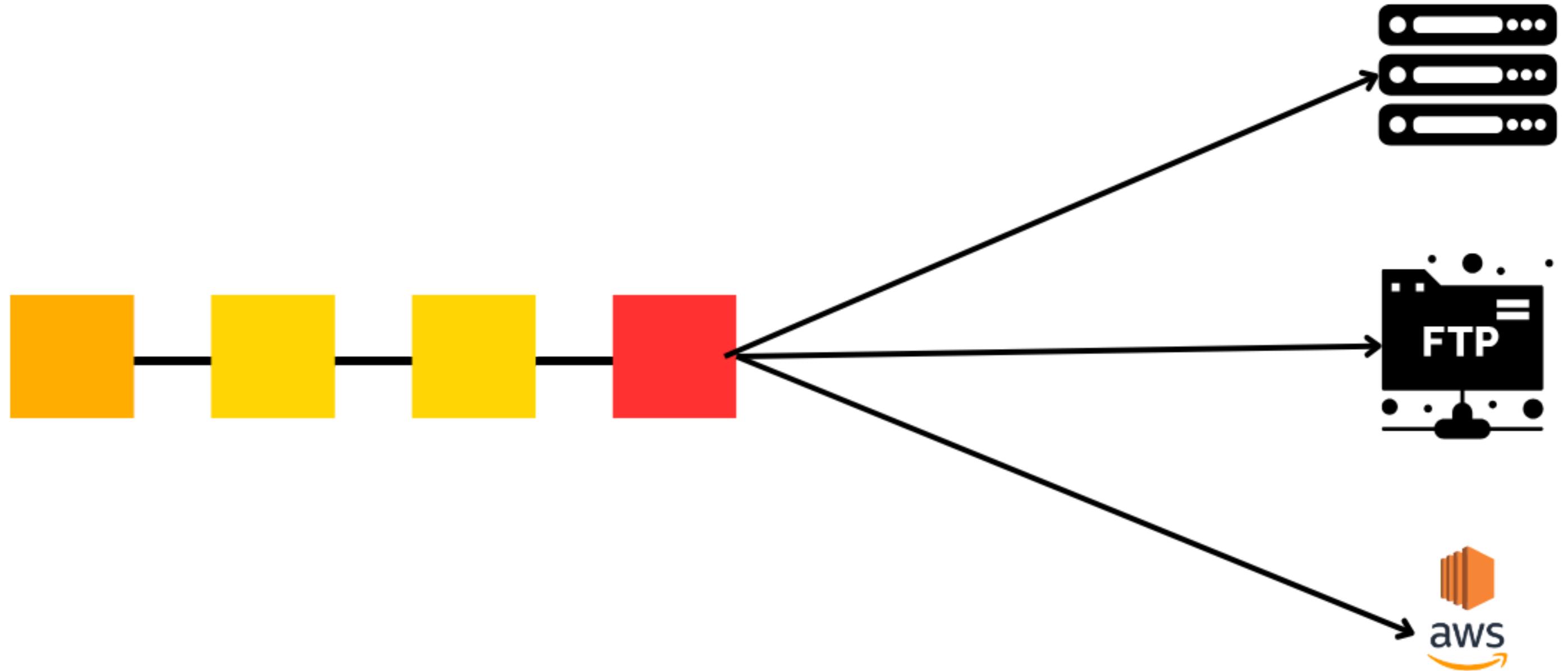
Remote file systems



Remote file systems

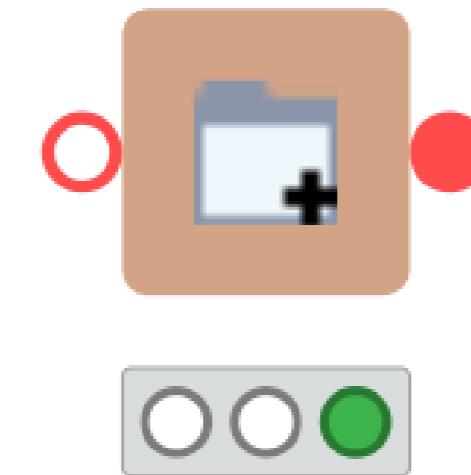


Remote file systems



Utility nodes for file automation

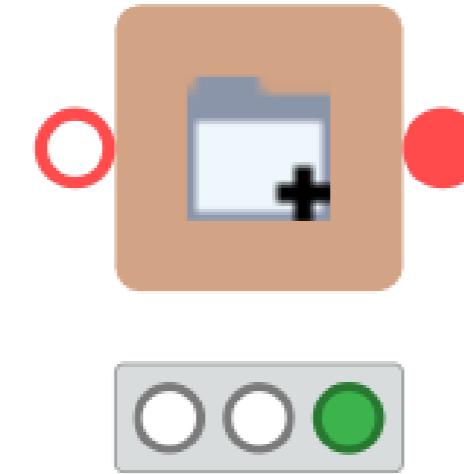
Create Folder



Create folder
on local machine

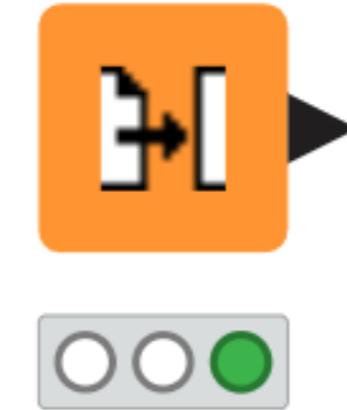
Utility nodes for file automation

Create Folder



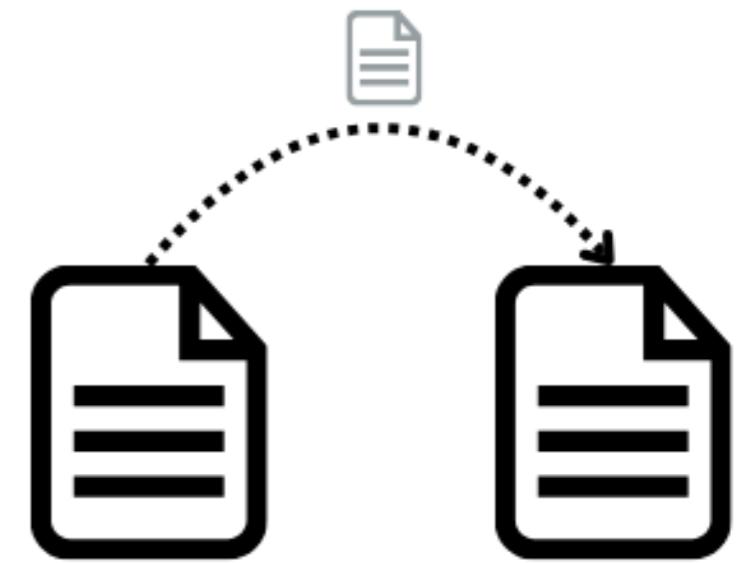
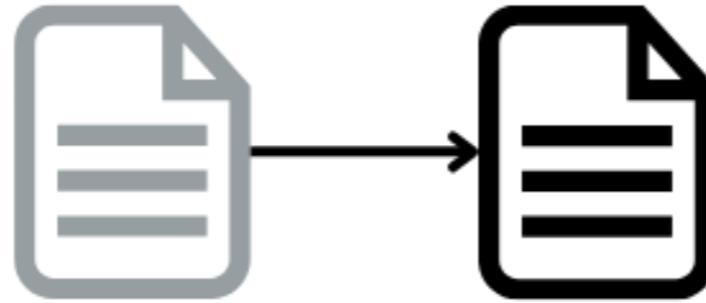
Create folder
on local machine

Transfer Files

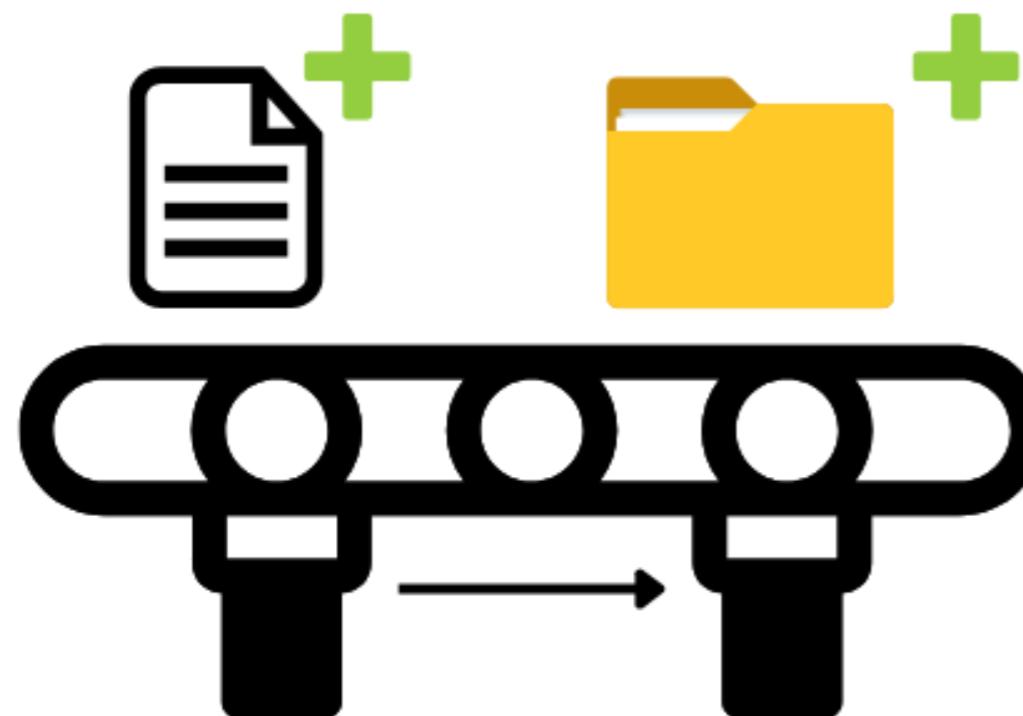


copy and move
files
automatically

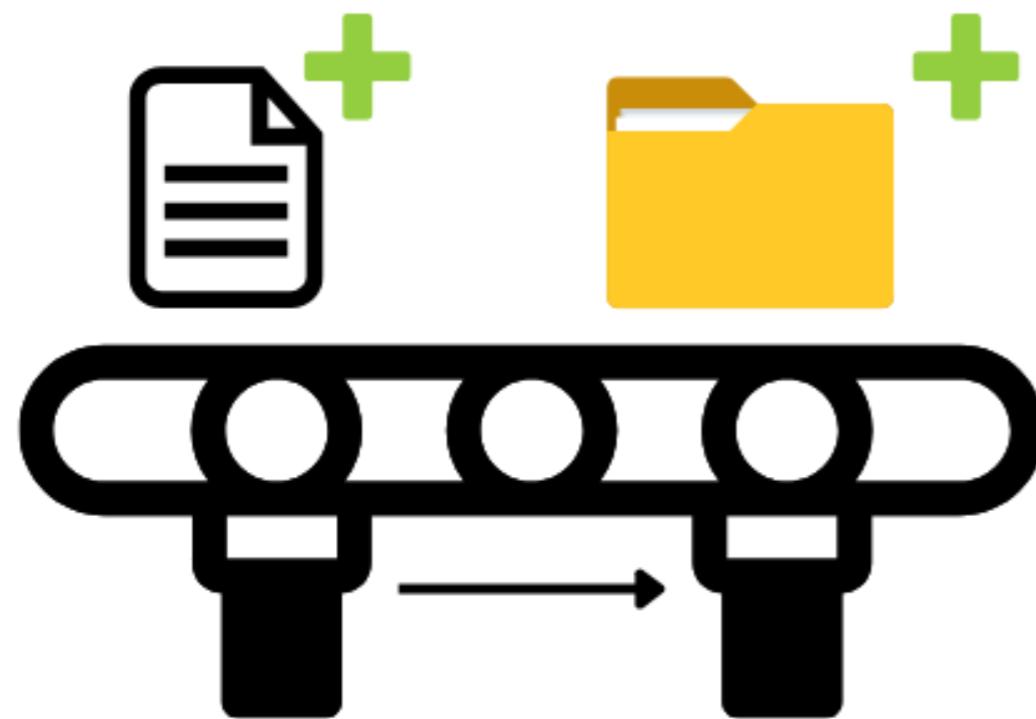
Utility nodes for file automation



Utility nodes for file automation



Utility nodes for file automation



Let's practice!

DATA MANIPULATION IN KNIME

Hands-on advanced data reading and writing

DATA MANIPULATION IN KNIME



Let's practice!

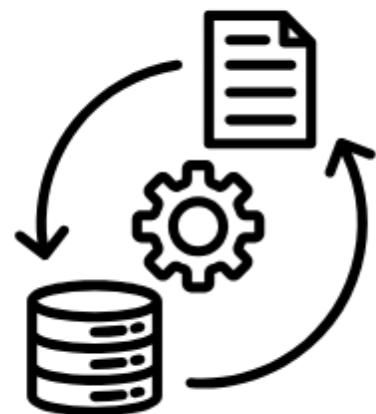
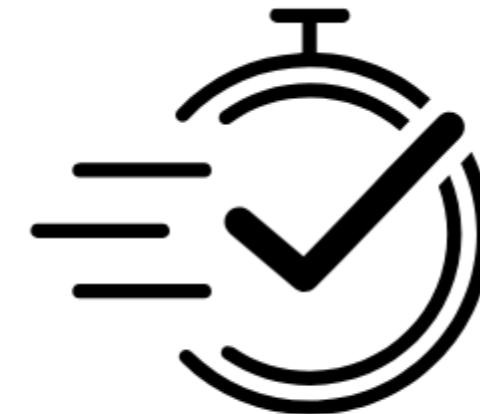
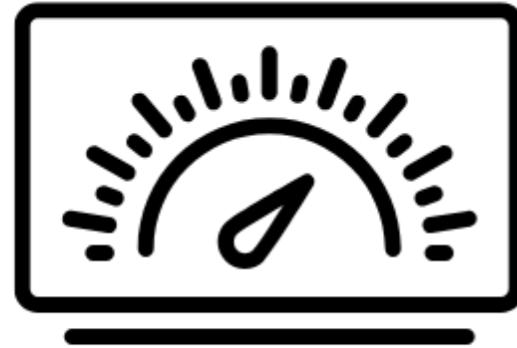
DATA MANIPULATION IN KNIME

Introduction to database operations in KNIME

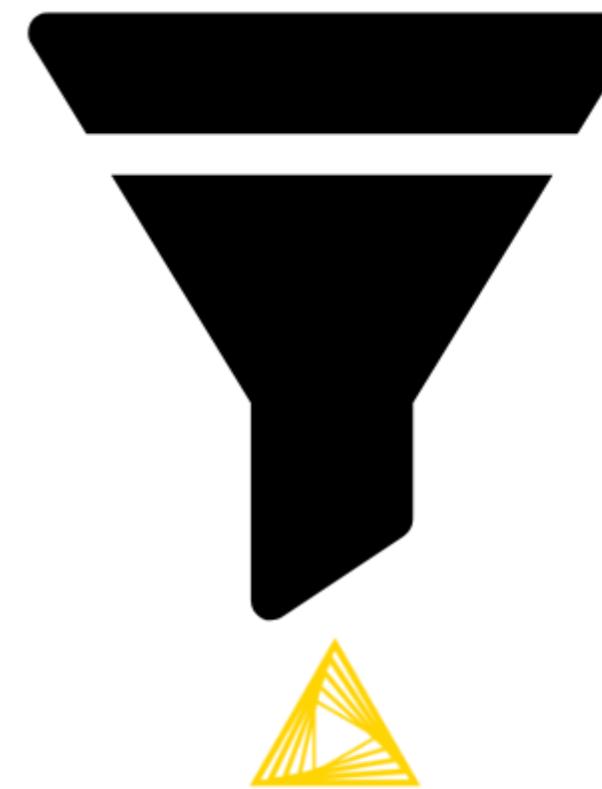
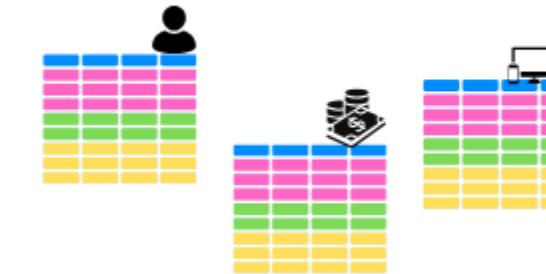
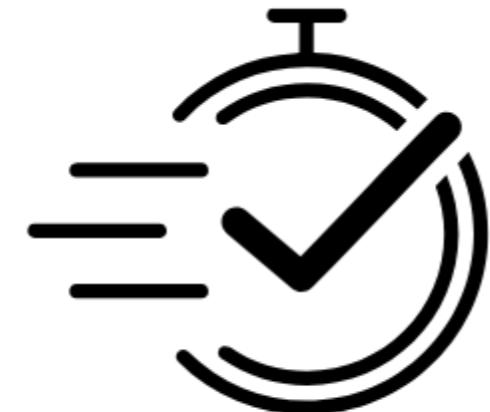
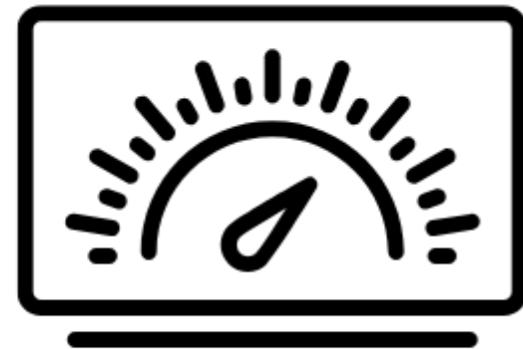
DATA MANIPULATION IN KNIME



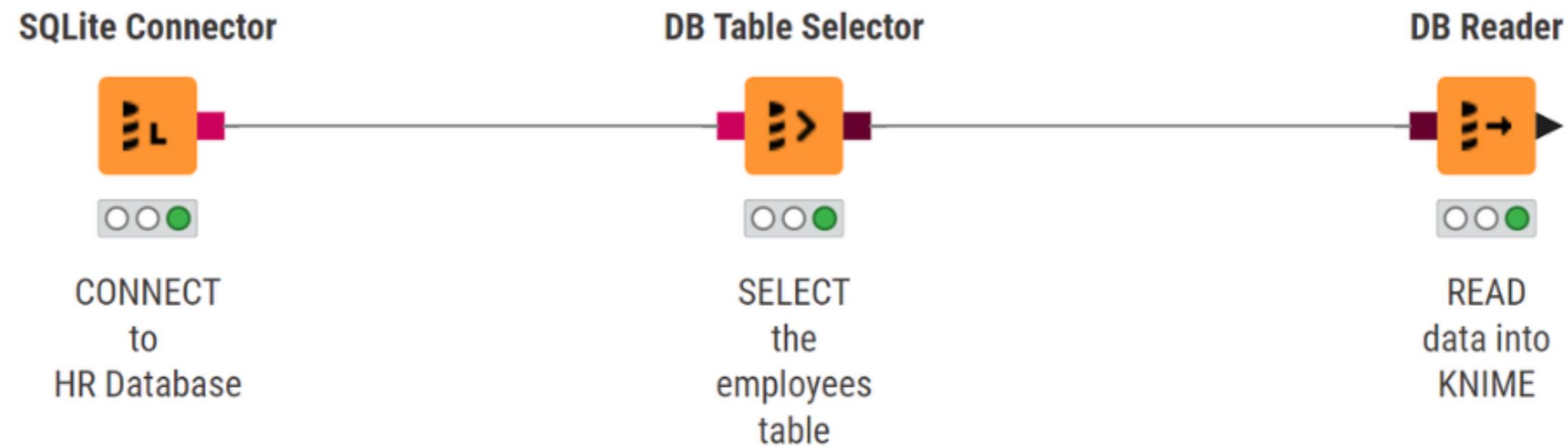
Benefits of working directly in a database



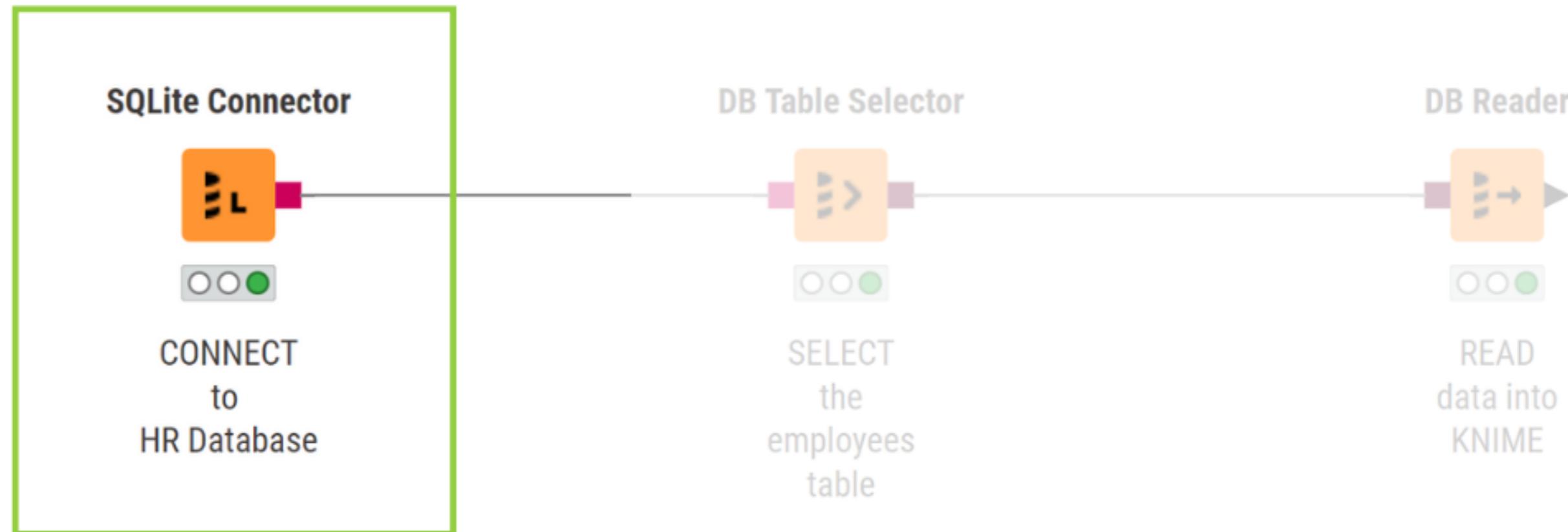
Benefits of working directly in a database



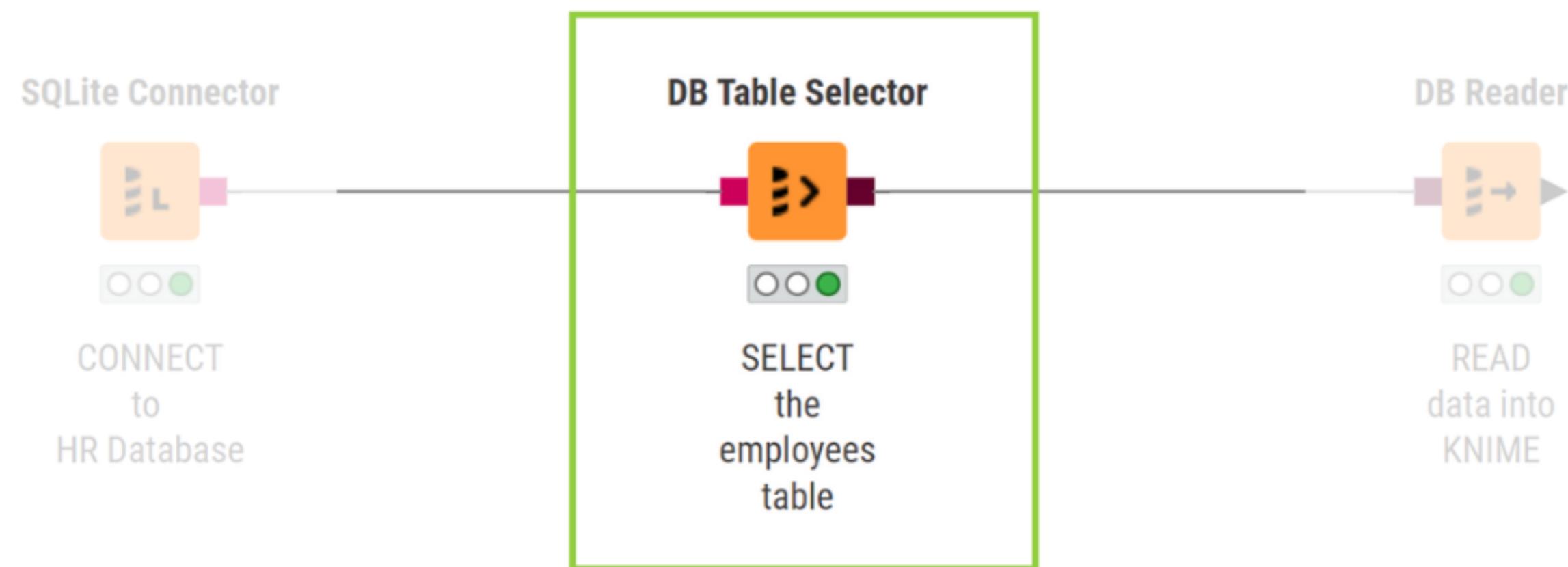
Connecting to databases



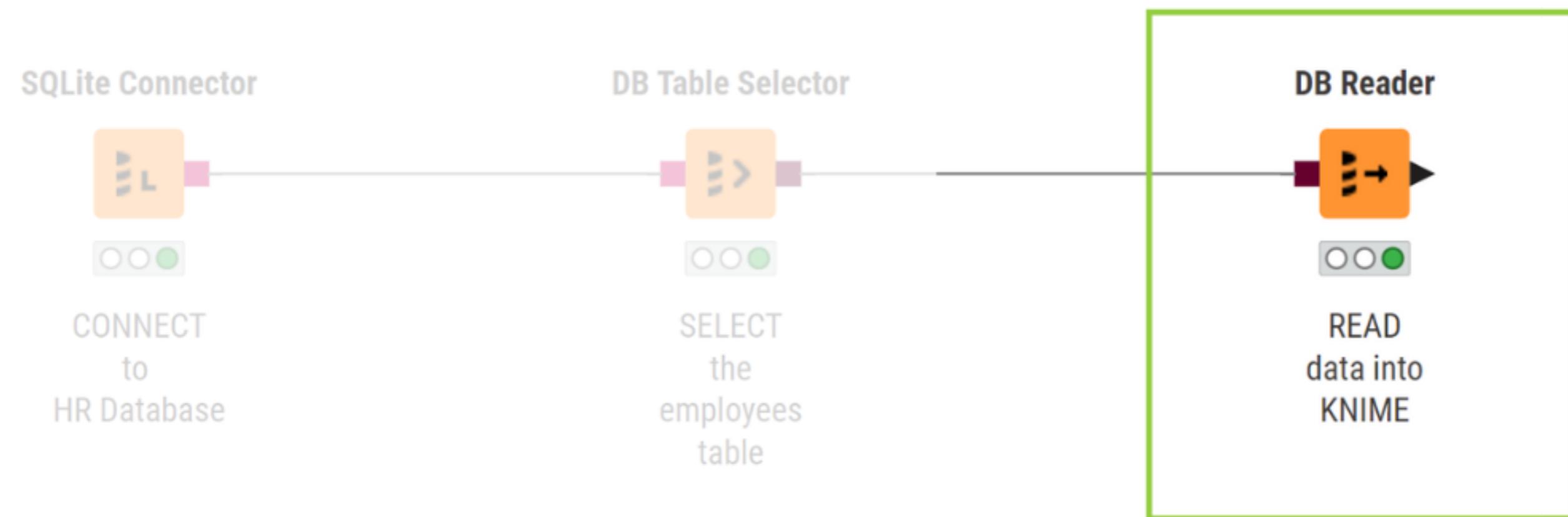
Connecting to databases



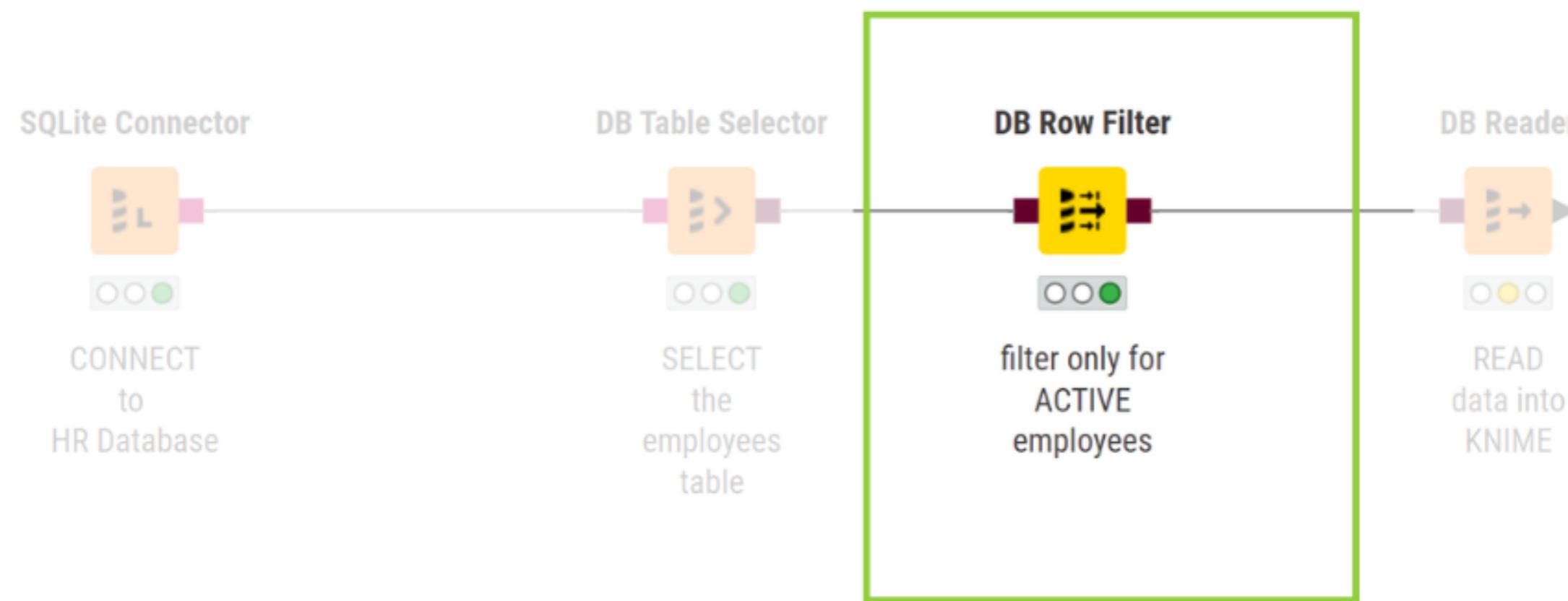
Connecting to databases



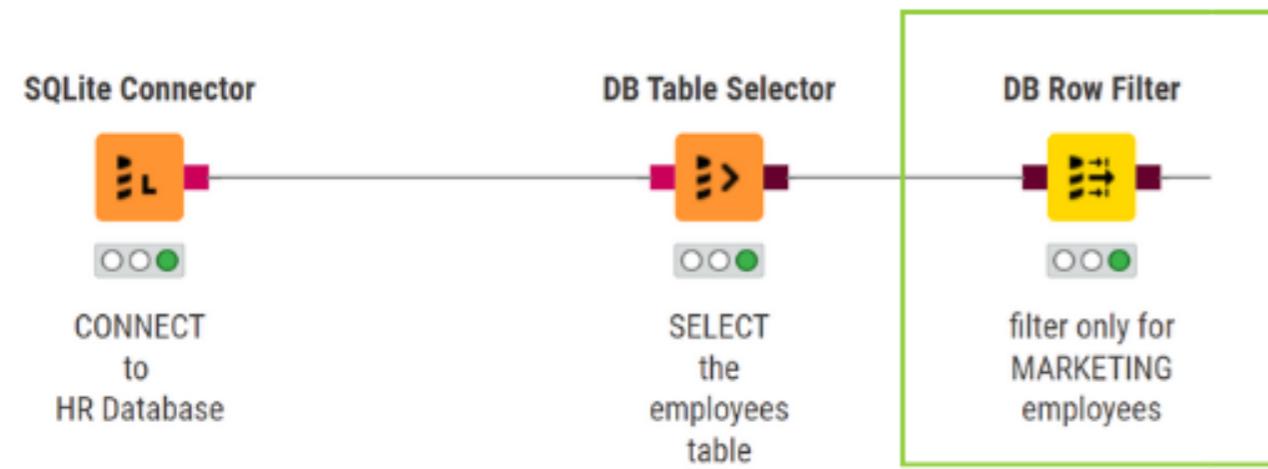
Connecting to databases



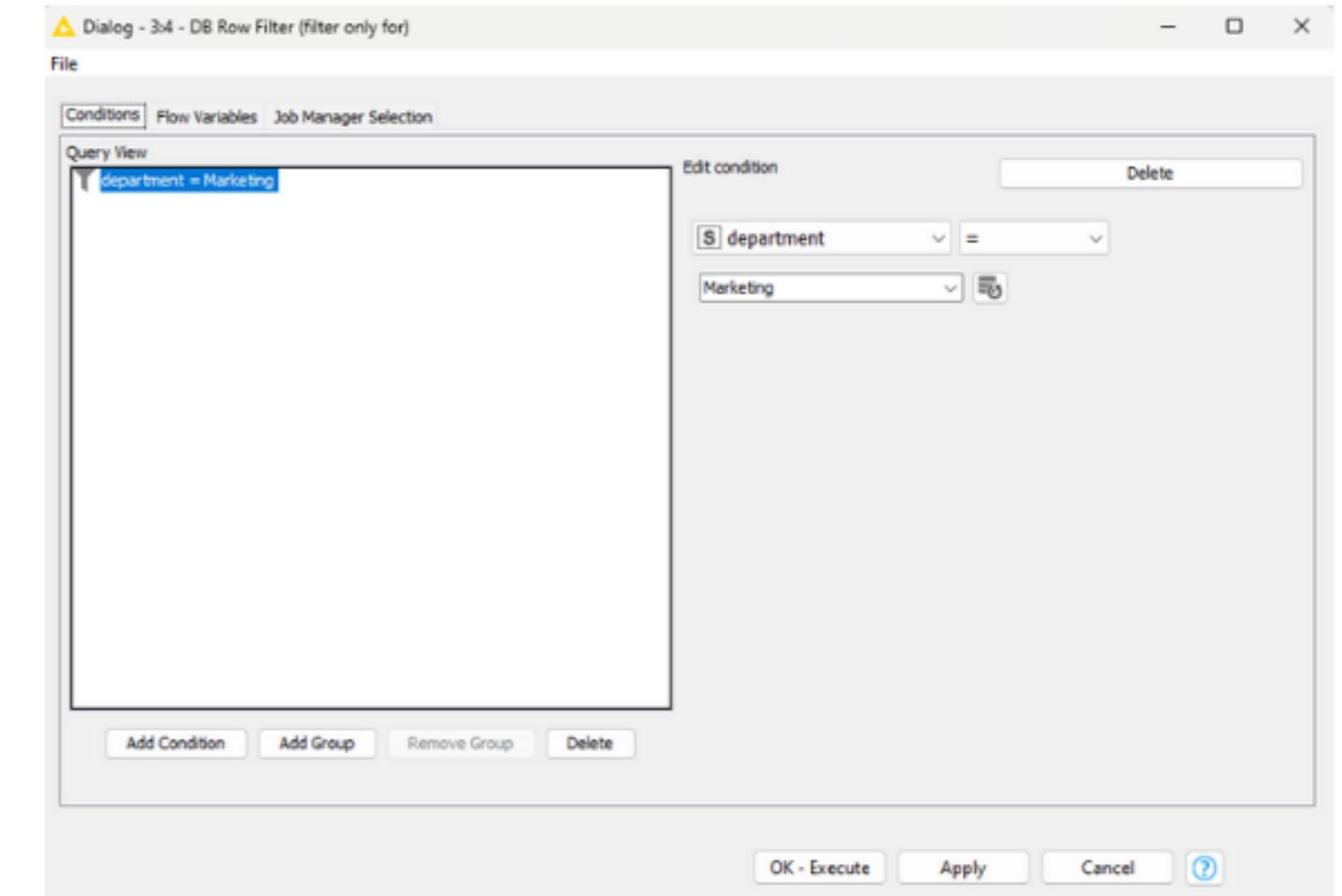
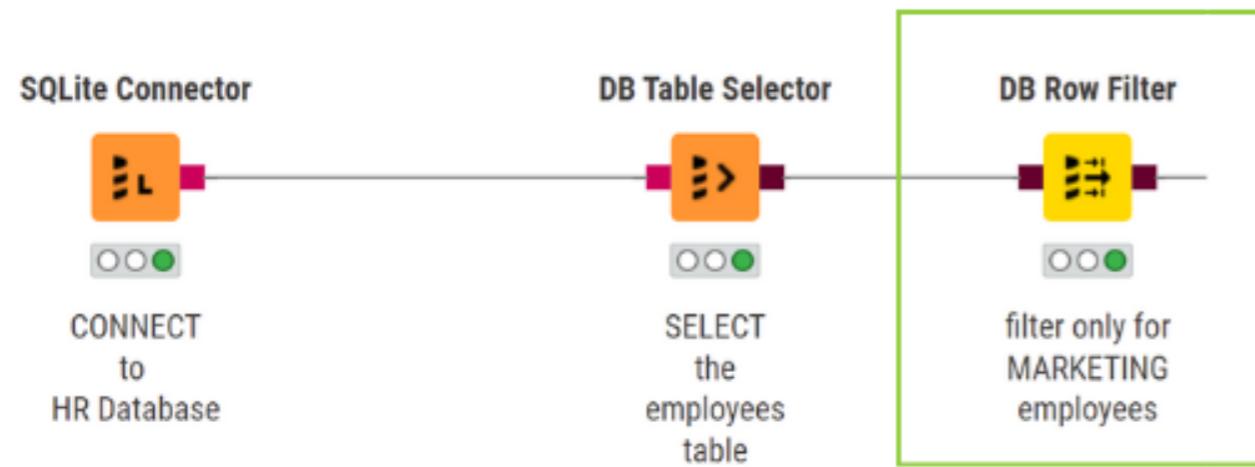
Filtering databases



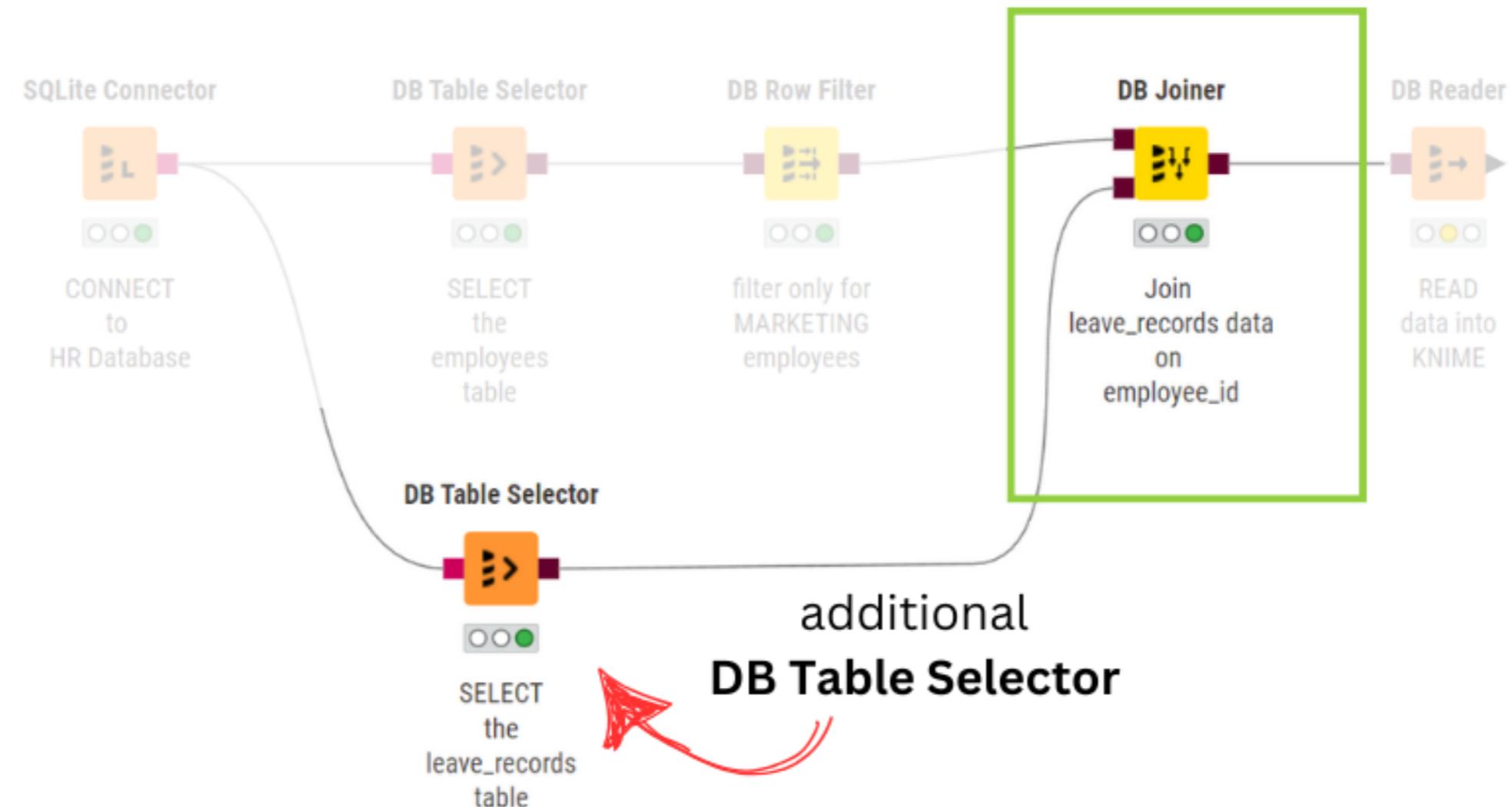
Filtering databases



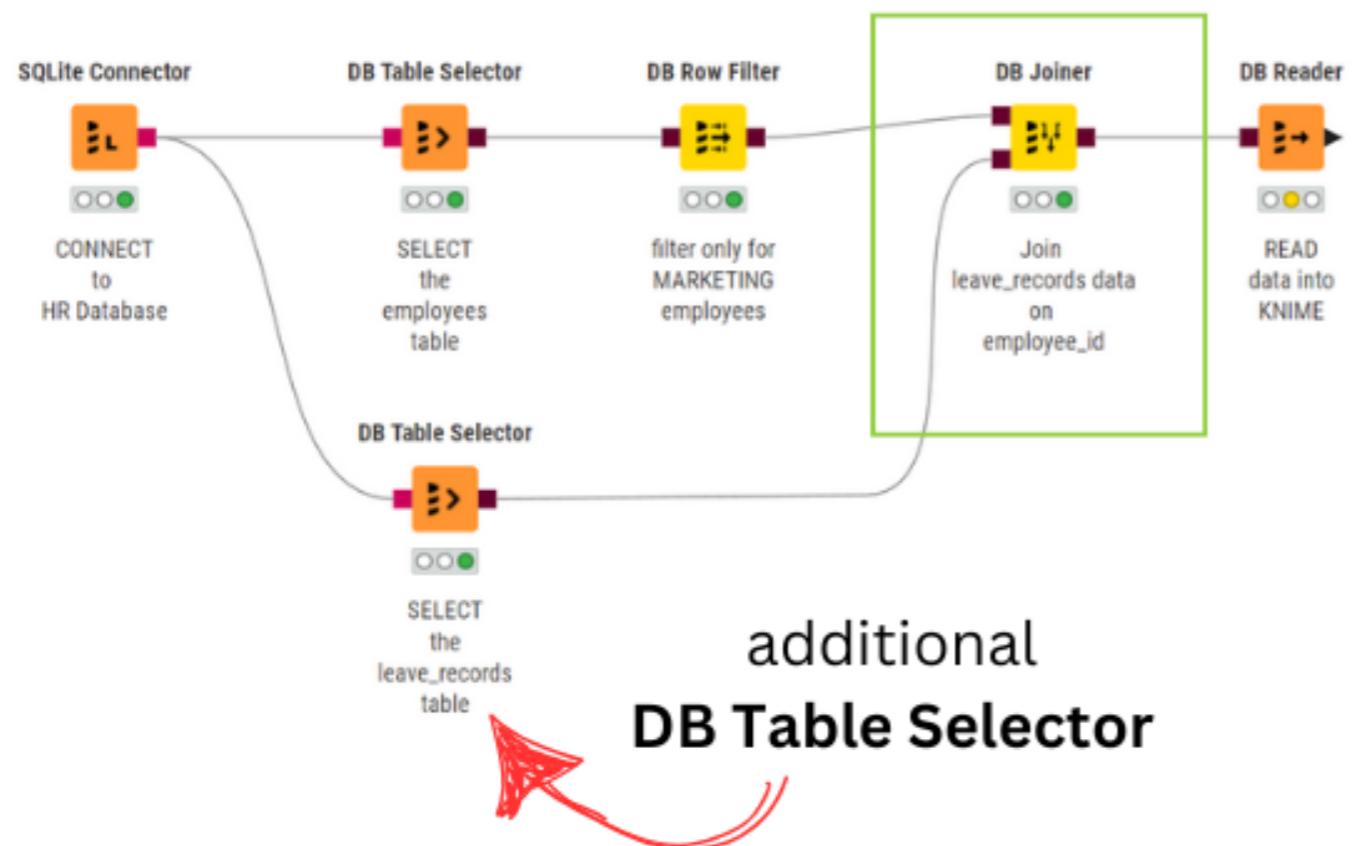
Filtering databases



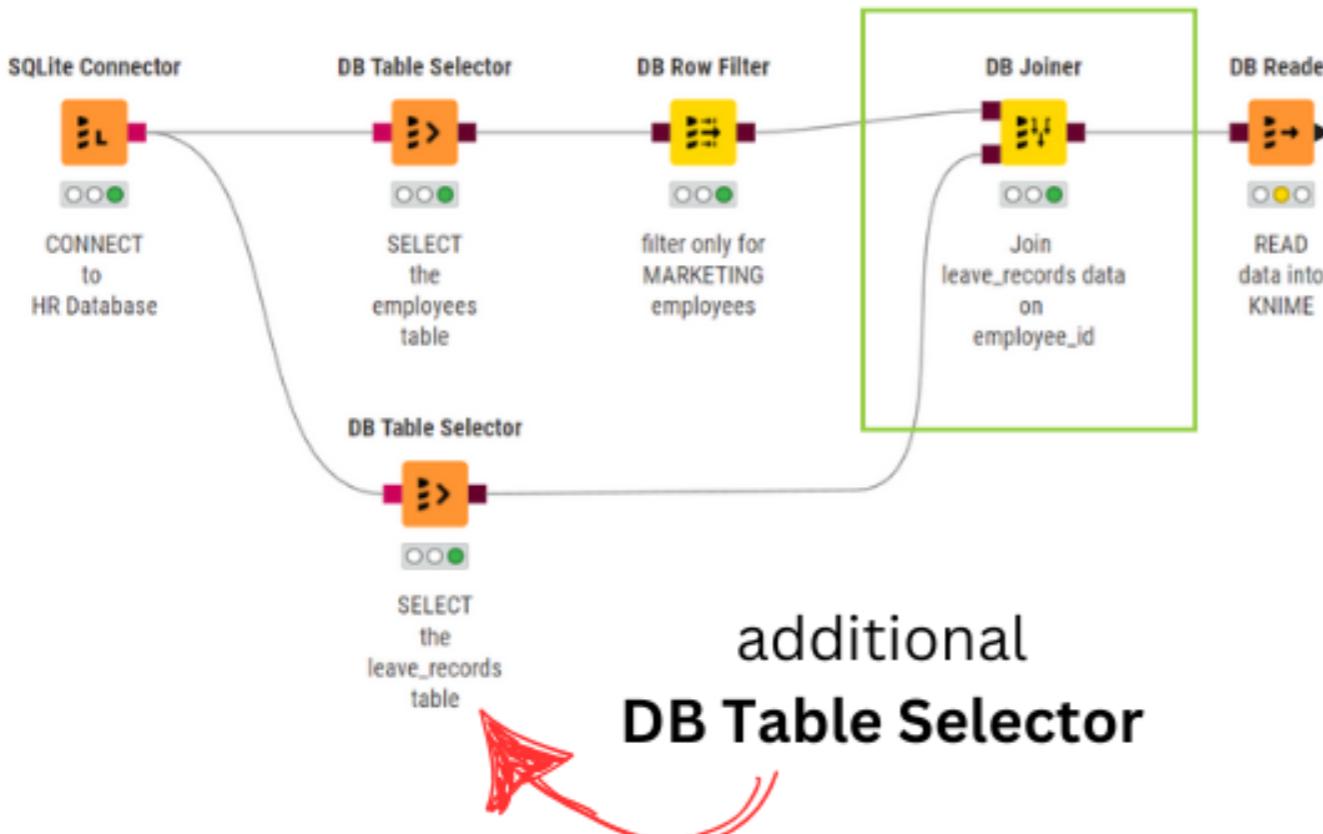
Merging database tables



Merging database tables



Merging database tables

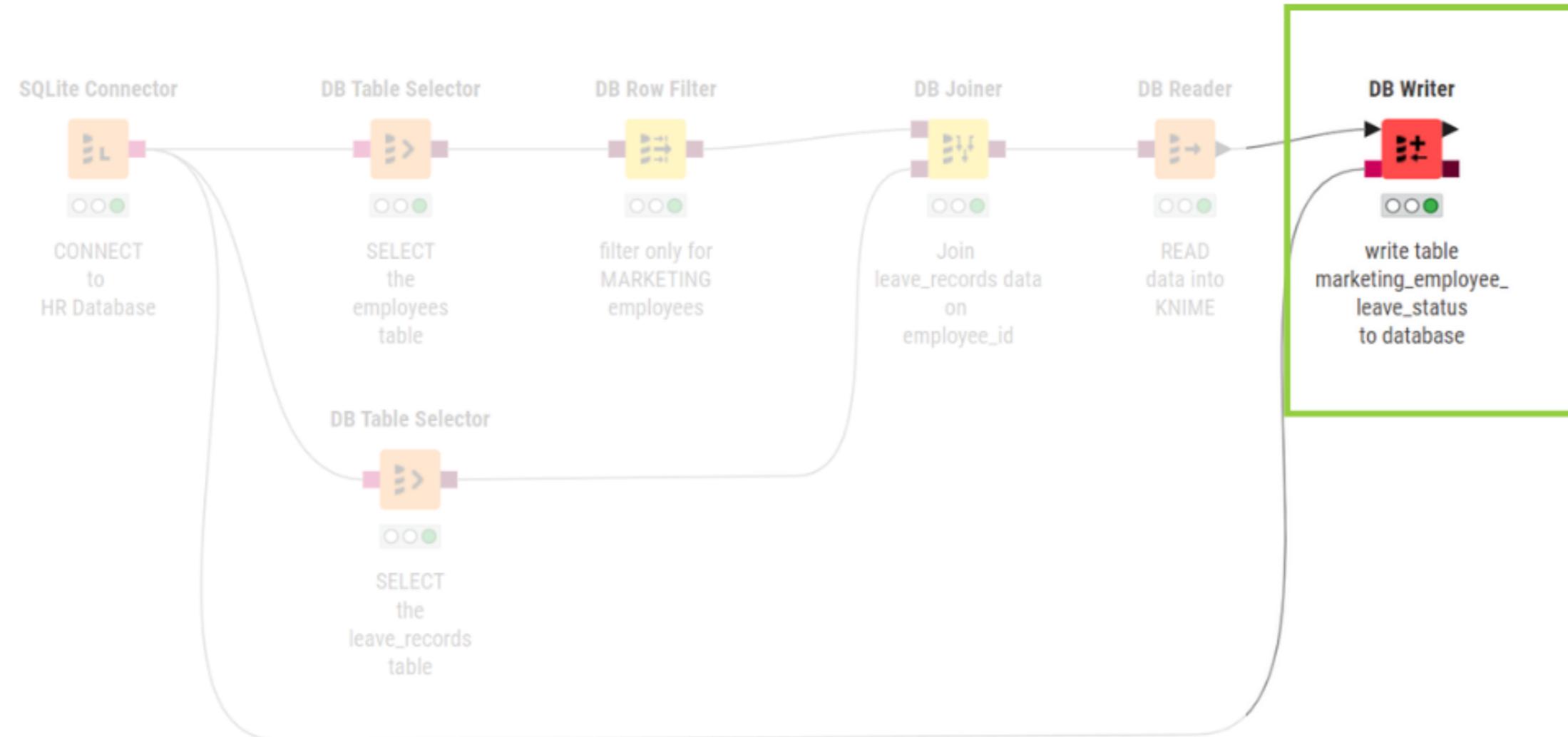


The screenshot shows the KNIME interface with three tables and their joined result:

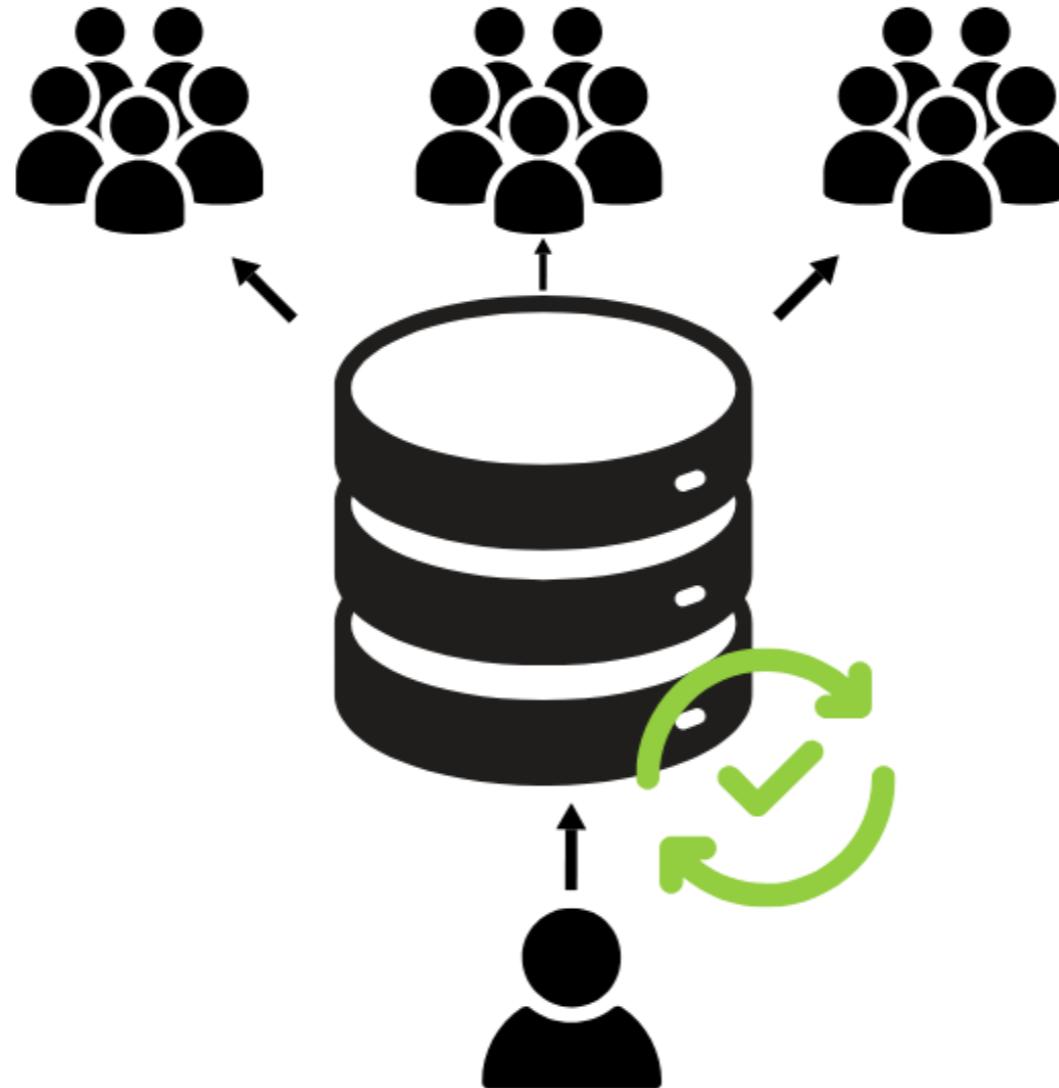
- Employees**: A table with columns: #, RowID, employee_id, first_name, last_name, department, hire_date, status, salary, leave_type, approval_status.
- Leave Records**: A table with columns: #, leave_id, leave_type, employee_id, start_date, end_date, leave_type, approved_status.
- Joined Table**: The resulting table after joining the two tables based on employee_id. It includes all columns from both tables.

#	RowID	employee_id	first_name	last_name	department	hire_date	status	salary	leave_type	approval_status
1	Row0_2	2	Salem	Ruford	Marketing	3/27/2008	Active	76481.52	sick leave	pending
2	Row1_5	5	Maisley	Ivanov	Marketing	2/25/2006	Inactive	71370.25	personal leave	pending
3	Row2_10	10	Wade	Winchester	Marketing	8/29/2009	On Leave	102777.07	sick leave	approved
4	Row3_12	12	Windham	JANIC	Marketing	7/18/2011	Inactive	106256.62	sick leave	rejected
5	Row4_20	20	Bukley	Gernse	Marketing	9/7/2002	Inactive	40952.27	vacation	pending
6	Row5_29	29	Zolene	Egle of Germany	Marketing	3/30/2011	Active	27663.07	personal leave	approved
7	Row6_34	34	Tamm	Lloyd	Marketing	8/6/2020	On Leave	128753.96	vacation	approved
8	Row7_38	38	Prudy	Domingo	Marketing	8/1/2020	On Leave	86272.11	vacation	pending
9	Row8_39	39	Hanni	Indyfultone	Marketing	8/31/2009	On Leave	137268.7	personal leave	approved
10	Row9_45	45	Lorenza	Clelland	Marketing	10/22/2002	Active	105886.46	vacation	approved
11	Row.._53	53	Allie	Simla	Marketing	3/4/2004	Inactive	97142.07	sick leave	pending
12	Row.._62	62	Hannet	Robertoni	Marketing	1/23/2020	Active	62310.47	sick leave	rejected
13	Row.._75	75	Egbert	Marisine	Marketing	10/17/2006	On Leave	29147.84	vacation	pending
14	Row.._80	80	Wain	Garlow	Marketing	12/31/2017	On Leave	32149.71	personal leave	rejected
15	Row.._82	82	Chester	Pavelka	Marketing	9/24/2013	On Leave	31675.91	personal leave	rejected
16	Row.._83	83	Josée	Fleckino	Marketing	6/28/2022	Active	33135.38	vacation	pending
17	Row.._86	86	Shea	Wisley	Marketing	9/6/2010	On Leave	59298.82	personal leave	rejected
18	Row.._90	90	Sheryln	O'Fallon	Marketing	6/23/2008	Inactive	121924.11	sick leave	rejected
19	Row.._92	92	Jenid	Qweste	Marketing	2/25/2007	Inactive	124653.83	personal leave	pending
20	Row.._97	97	Ernesta	Izon	Marketing	11/4/2003	Inactive	64991.12	vacation	pending
21	Row.._106	106	Merri	Feeley	Marketing	3/15/2001	Active	30723.42	vacation	rejected
22	Row.._109	109	Calhoun	Gonzalo	Marketing	8/27/2008	Active	59999.09	sick leave	approved
23	Row.._114	114	Banette	Mustard	Marketing	5/30/2001	Active	52124.42	vacation	approved
24	Row.._117	117	Yvonne	Cookley	Marketing	11/14/2009	Active	87032.68	vacation	approved
25	Row.._118	118	Mindy	O'Loughlin	Marketing	3/14/2010	Active	43025.37	personal leave	pending

Updating databases



Updating databases



Updating databases



Let's practice!

DATA MANIPULATION IN KNIME

Hands-on database tasks in KNIME

DATA MANIPULATION IN KNIME



Let's practice!

DATA MANIPULATION IN KNIME

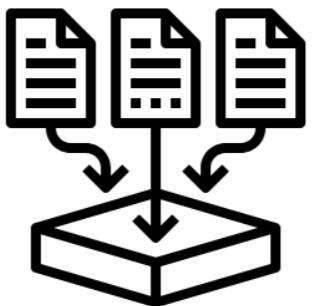
Congratulations!

DATA MANIPULATION IN KNIME



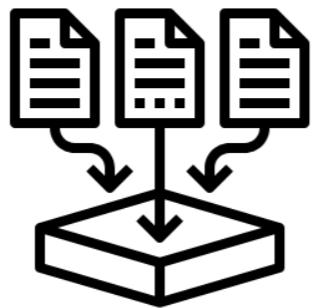
What you have learned

Merging data



What you have learned

Merging data

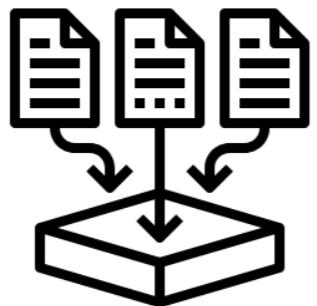


Aggregating data



What you have learned

Merging data



Aggregating data

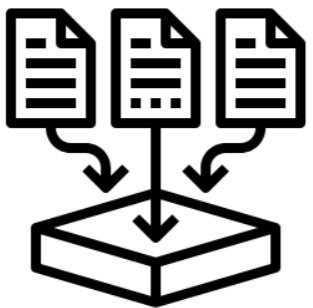


Advanced file handling



What you have learned

Merging data



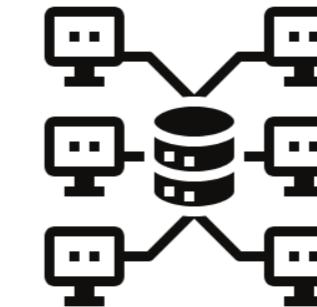
Aggregating data



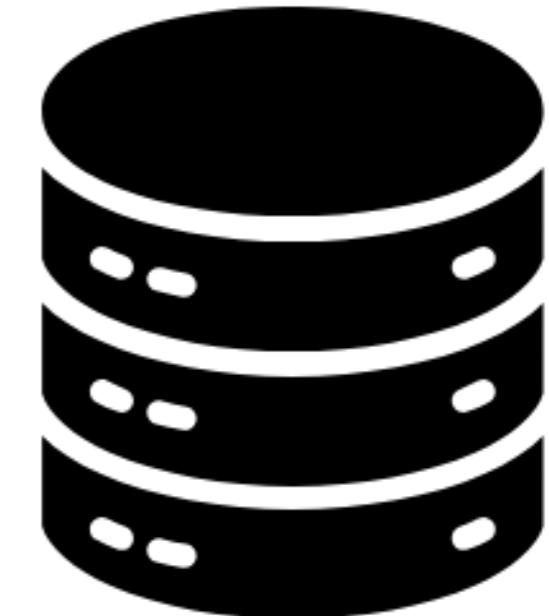
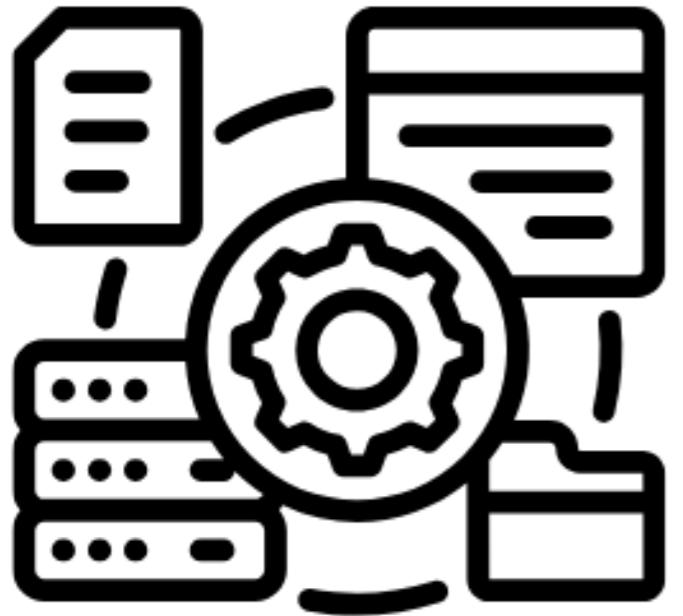
Advanced file handling



Database handling



Your new skills



DATA MANIPULATION IN KNIME

Congratulations!

DATA MANIPULATION IN KNIME