

# Alteryx Designer Core Certification

## Table of Contents (Learning Modules):

- [Layout](#)
- [Benefits of Designer](#)
- [Input Data](#)
- [Text Input](#)
- [Datatypes](#)
- [Select](#)
- [Browse](#)
- [Filter](#)
- [Sort](#)
- [Sample](#)
- [Text to Columns](#)
- [Unique](#)
- [Union](#)
- [Join](#)
- [DateTime](#)
- [Rows Vs Columns](#)
- [Functions](#)
- [Expression Editor](#)
- [Formula](#)
- [Summarize](#)
- [Transpose & Cross Tab](#)
- [Find Replace](#)
- [Append Fields](#)
- [Output Data](#)
- [File types and extensions](#)
- [Data Quality Categories](#)

## Practice Exercises:

- [Practice Exercise 1](#)
- [Practice Exercise 2](#)
- [Practice Exercise 3](#)
- [Practice Exercise 4](#)

## Alteryx Layout

- Canvas
- Tool Palette
- Configuration Window
- Results Window

## Benefits of Designer

- Transparency
  - o Visibility into all the actions performed on the dataset. The workflow shows what steps were taken and the order in which they were performed
- Repeatability
  - o Use a workflow to transform manual steps into an automated process with a repeatable workflow, simply change the input data and click “run”
- Safety
  - o Designer offers a safe development environment for testing, exploring, and solving. You choose where and how to save results
- Scalability
  - o Designer accommodates multiple data sources at once, in the same workflow. There is now row limit, so you can work with your large data sets all at once. It also means that those sources can exist as varying formats in one workflow



## INPUT DATA

Bring data into your workflow by selecting a file or connecting to a database (optionally, using a query).

## Input Data Tool

- Connect to data sources using input data tools
- You can also drag a file onto the canvas in designer and an input data tool will be added and configured for you if the file type is supported
- Data will be imported as a table, with each cell value associated with a particular column (field) and row (record). In this tabular format, values are tied to the header of their column. They are also related to the values in the same record.

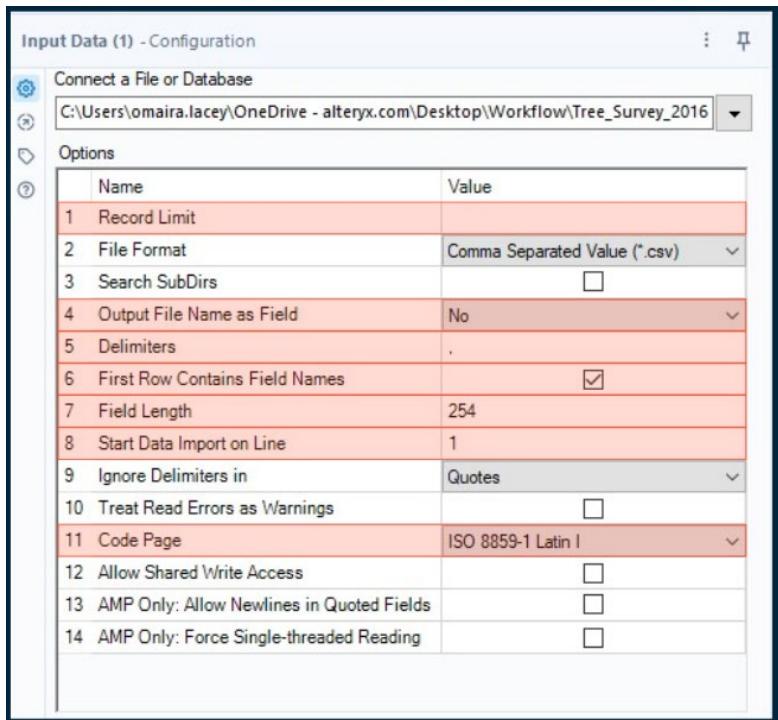
### *Interactive lesson one:*

- Nearly every workflow you build will start with an input data tool
- Input data tool can be found in the In/Out and Favorites tool categories

- Reads data into designer from a connected source
- Data at the connected source remains unchanged
- An input's data value and metadata are only changed to an overwritten with an output data tool
- Input data tool includes only one anchor, an Output anchor – makes data available to other connected tools
- Best practice: close all input files to avoid errors

## CSV Configuration

- Csv file contains string values that are usually comma delimited
- When a csv file is used as the input file type. The screen below is shown:



### Record limit

- Limit the number of records read in to improve run times
- Manually type record limit in

### Output File Name as Field

- Add a column showing source file name or full path
- Easy to identify and use source info in the workflow

### Delimiters

- May not always be a comma
- Can manually enter delimiters such as a pipe (|) or a space

### First Row Contains Field Names

- Deselect if columns do have names

## Field Length

- By default, field length is 254 characters
- If values are truncated, manually change length to avoid truncated values

## Start Data Import on Line

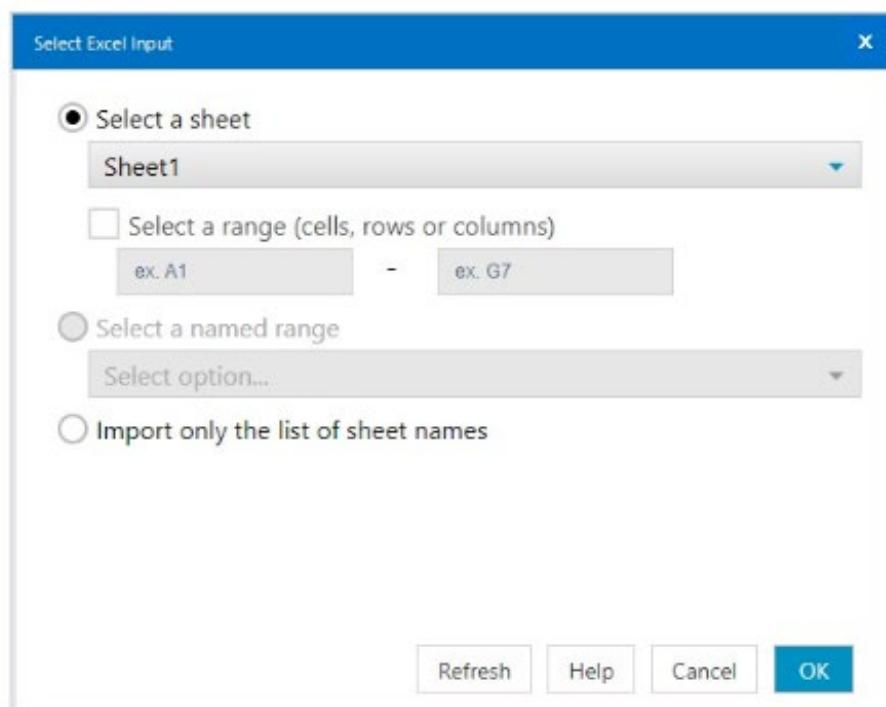
- Avoid importing extraneous information from data sources
- Enter line on which data values start by manually entering in line number

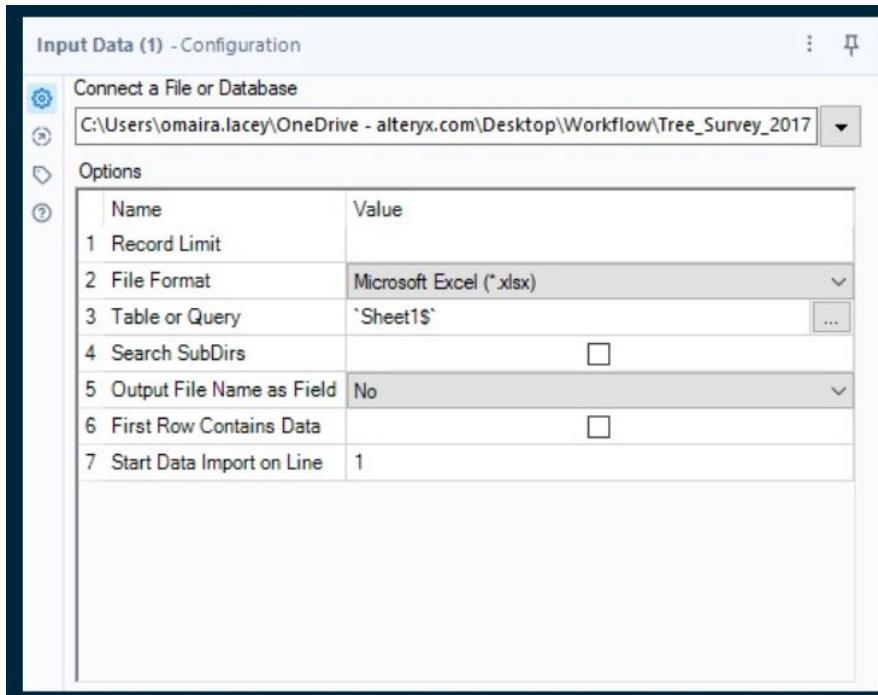
## Code Page

- Some data is uncoded with a particular format (e.g. UTF-8 or language)
- Select the correct code page for your data

## Excel Workbook Configuration

Excel workbook stores data in a tabular structure in sheets or a named range





#### Record Limit

- Manually type records to input in the configuration window

#### Table or Query

- Displays your configuration selection
- Click the ellipsis (...) to edit

#### Output File Name as Field

- Add a column showing source file name or full path
- Easy to identify and use source info in the workflow

#### First Row Contains Field Names

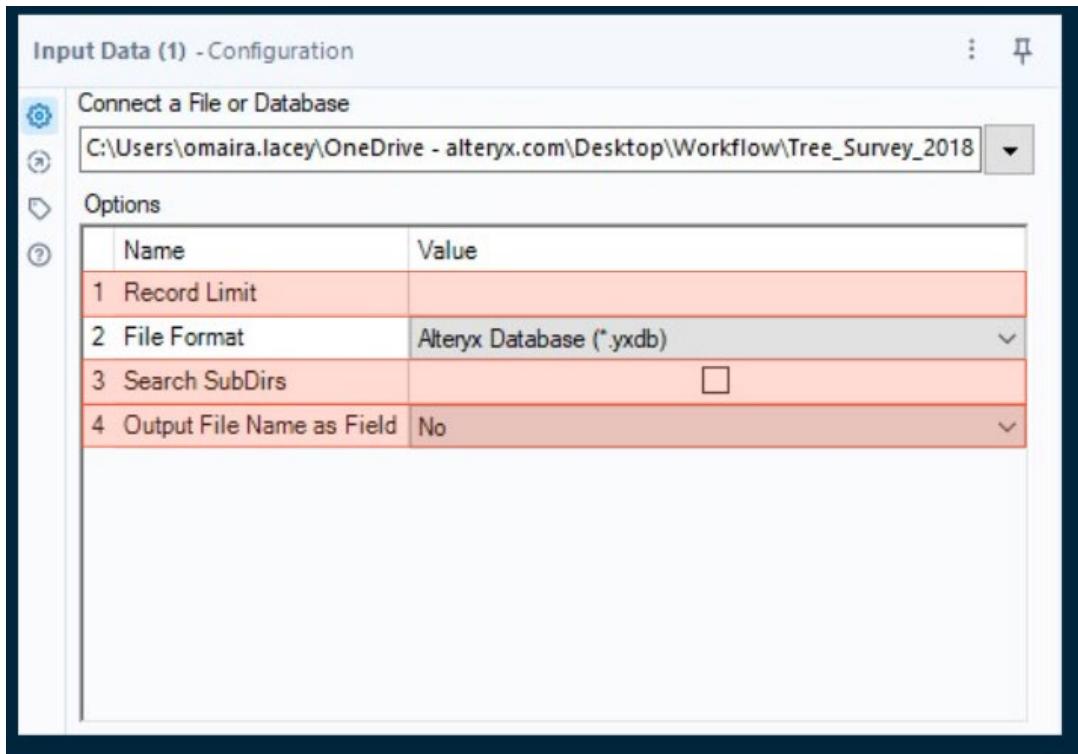
- Deselect if columns do have names

#### Start Data Import on Line

- Avoid importing extraneous information from data sources
- Enter line on which data values start by manually entering in line number

#### **YXDB Configuration**

An Alteryx database file or .YXDB is the most efficient filetype for reading data into Alteryx



#### Record Limit

- Manually type records to input in the configuration window

#### Search Subdirs

- Search Subdirectories to input other files with the same name, type or structure

#### Output File Name as Field

- Add a column showing source file name or full path
- Easy to identify and use source info in the workflow

#### Previewing Data

- The input data tools configuration window displays only the first 100 rows of data from the connected input as a preview
- To input the contents of the entire dataset, click run to initiate the flow of data from the source to designer

#### Questions:

1. True or False? The input data tool can only connect to files?

Answer: False

2. Which of the following are present in the configuration options for alteryx databases, Microsoft excel workbooks, and CSVs? Select all that apply.

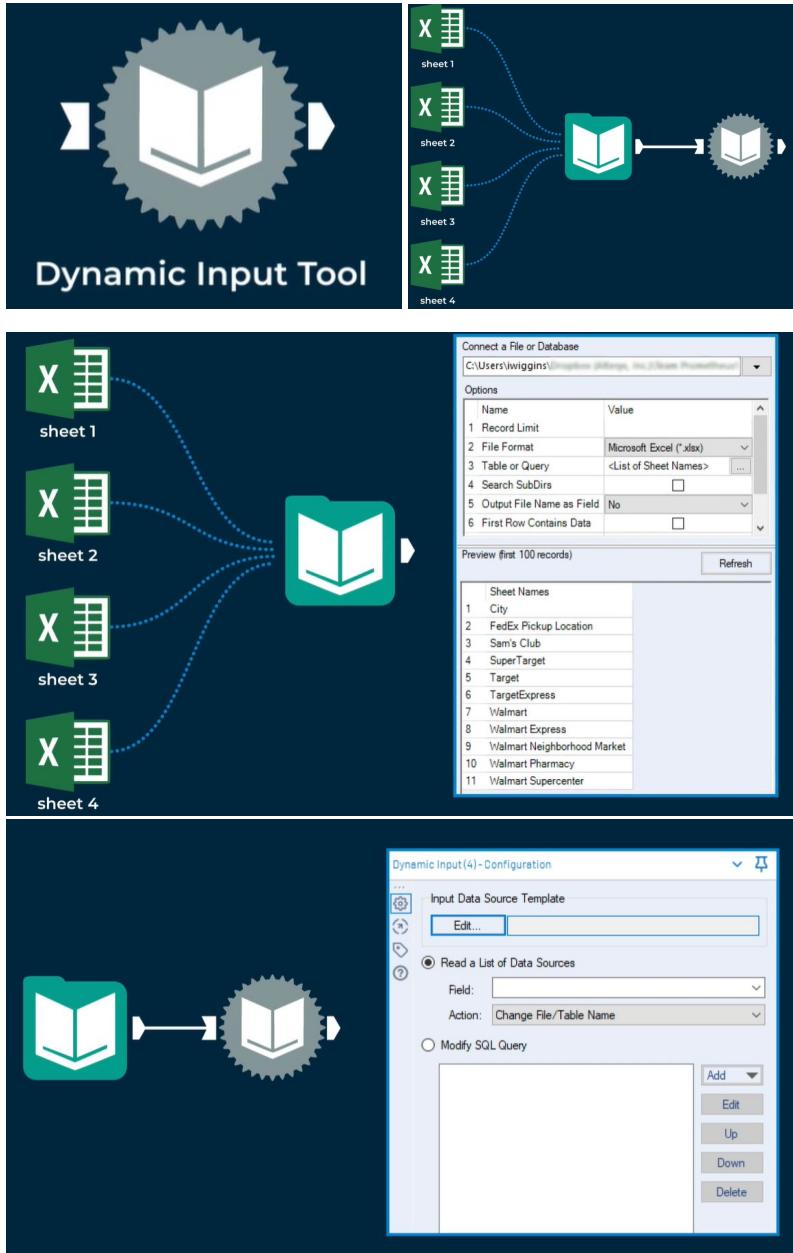
- a. Delimiters
- b. Record Limit

- c. First Row Contains Data
- d. Output File Name as a Field

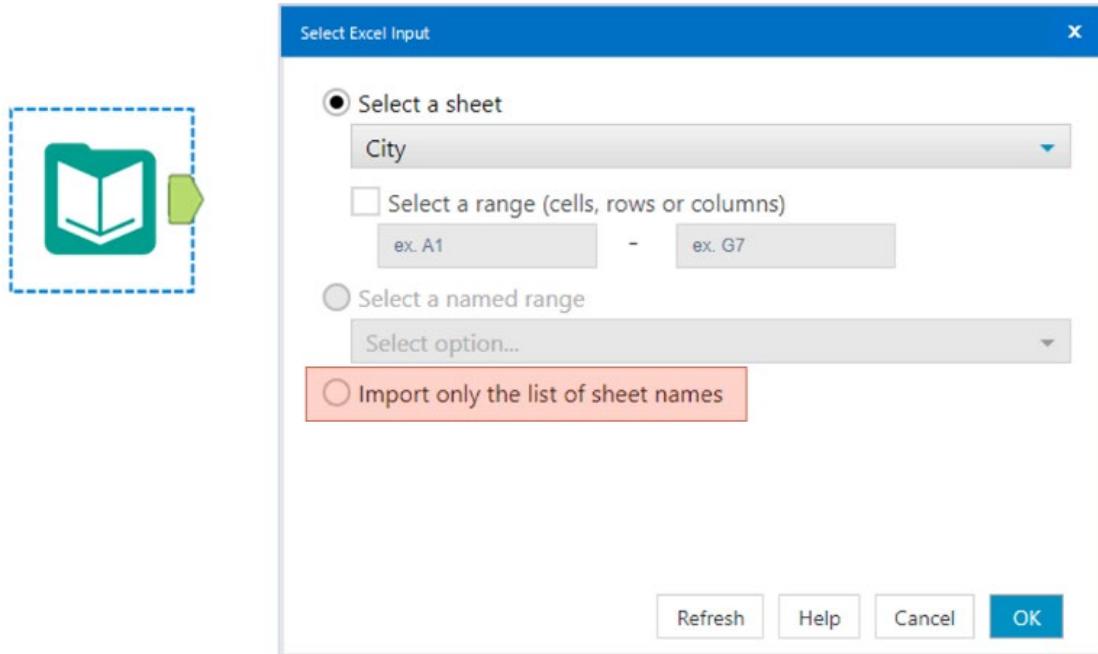
Answer: b, c, d

### Connecting to Multiple Sheets at Once

- Dynamic Input tool must be used
- Reading Multiple Sheets is only supported for reading .xlsx files



1. Import only the list of sheet names



## 2. Sheet names created

**Input Data (2) - Configuration**

**Connect a File or Database**  
C:\Users\syvia.pereira\alteryx.com\GRP\_Prometheus - Documents\Ge

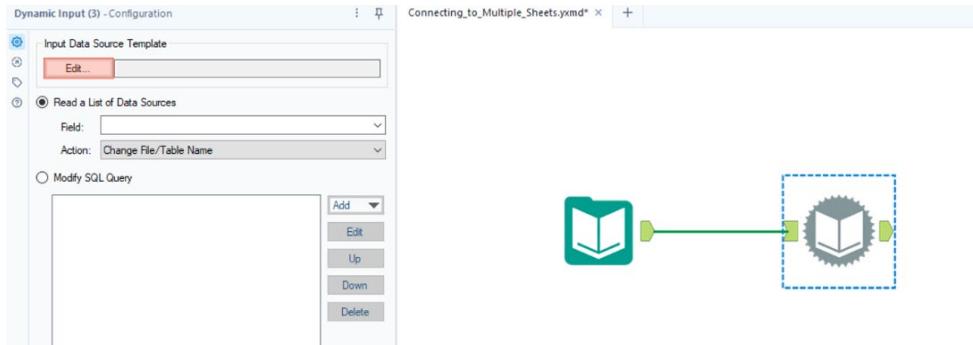
**Options**

Name	Value
1 Record Limit	
2 File Format	Microsoft Excel (*.xlsx)
3 Table or Query	<List of Sheet Names>
4 Search SubDirs	
5 Output File Name as Field	No
6 First Row Contains Data	

**Preview (first 100 records)**

Sheet Names
1 City
2 FedEx Pickup Location
3 Sam's Club
4 Walmart
5 Walmart Express
6 Walmart Neighborhood Market
7 Walmart Pharmacy
8 Walmart Supercenter
9 TargetExpress
10 SuperTarget
11 Target

## 3. Connect dynamic input tool and edit



4. Select Template
  - If a sheet does not have the same number of columns as the template it will not be read in
  - Template selection is critical to avoid misalignment of data
5. Click field sheet names and leave action as default
6. Run workflow

**Questions:**

1. Which of the following file types support reading in multiple sheets? Select all that apply.
  - a. .xls
  - b. .xlsx
  - c. .yxdb
  - d. .csv
  - e. All listed options

**Answer: b**

2. What happens when designer fails to read in a sheet?
  - a. An error message
  - b. This would never happen in designer
  - c. A warning message appears
  - d. Nothing happens

**Answer: c**

3. True or False? The Dynamic Input tool is the only tool required to import multiple sheets at once?

**Answer: False**

## Text Input



# TEXT INPUT

Manually add data values that will be stored in the workflow.

Ways to enter text input:

- Manually type
- Copy and Paste from another file
- Import from an existing file

Differences between Text input and Input Data:

Text Input:

- Data entered within the text input tool lives within the workflow
- Values are static
- Can be shared by sharing the workflow
- Data is automatically categorized to smallest type and size

Input Data:

- Data lives outside of the workflow
- Values update
- Must export workflow to share files
- Data is configured based on the file type being imported


## DATA TYPES

Data types may need to match that of other inputs, especially when that data will be blended with other data

Be mindful of the characteristics of each data type and how it may impact your workflow later on

Empty values are assigned null values

Results - Text Input (9) - Output

Record	Name	Type	Size	Source	Description
1	Borough Code	Byte	1	Textinput:	
2	Borough Name	String	13	Textinput:	

**Questions:**

1. Cells that are left blank in a Text Input tool are assigned what value?
  - a. Empty
  - b. Null
  - c. Zero (0)

Answer: b

2. How can data be added to a workflow with a Text Input tool? Choose all that apply.
  - a. Input data from a file or database
  - b. Manually enter values in cells
  - c. Copy and paste values from another source

Answer: a, b, c

3. True or False? Any value entered into the Text Input tool is characterized as a string data type.

Answer: False

**Datatypes**

## DATATYPES IN DESIGNER

After reading data into Designer, you'll need a thorough understanding of it to perform meaningful analysis. All incoming data is assigned one of [five main datatypes in Designer](#). Datatypes influence things like viewing, blending and preparing data in Designer.

**5 Data Types:****1. String**

A string represents alphanumeric data and can include letters, numbers, spaces, or other types of characters. You can also think of a string as plain text. All the characters in a string are processed as text even if the characters are digits.

Type	Description	Example
String	Fixed Length Latin-1 String. The length should be at least as large as the longest string you want to be contained in the field, or values are truncated. Limited to 8192 Latin-1 characters.	Any string whose length does not vary much from value to value, and only contains simple Latin-1 characters.
WString	Wide String accepts any character (Unicode.) Limited to 8192 characters.	Any string whose length does not vary much from value to value and contains any character.
V_String	Variable Length. The length of the field adjusts to accommodate the entire string within the field.	Any string whose length varies from value to value, and only contains simple Latin-1 characters.
V_WString	Variable Length Wide String. The length of the field adjusts to accommodate the entire string within the field and will accept any character.	Any string whose length varies from value to value and contains any character.

## 2. Numeric

There are several different numeric data types including integers, decimals, floats, and doubles. With the exception of the Fixed Decimal type, numeric data types are not adjustable in length.

Type	Description	Example
Byte	A unit of data that is 8 binary digits (bits) long. A byte field is a positive whole number that falls within the range 0 thru 255, or $2^8$	0, 1, 2, 3....253, 254, 255
Int16	A numeric value without a decimal equal to 2 bytes, or $-(2^{15})$ to $(2^{15})-1$	-32,768 to 32,767
Int32	A numeric value without a decimal equal to 4 bytes, or $-(2^{31})$ to $(2^{31})-1$	-2,147,483,648 to 2,147,483,647
Int64	A numeric value without a decimal equal to 8 bytes, or $-(2^{63})$ to $(2^{63})-1$	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Fixed Decimal	A numeric value with a decimal.  The length (precision) of a fixed decimal is equal to the width of the integer (left side of decimal) plus the decimal point plus the	A value of 1234.567 with a length of 7.2 results in 1234.57  A value of 1234.567 with a length of 7.3 results in a field conversion error and Null output, as the value does not fit within the specified precision.

	<p>width of the scale (right side of decimal). If a number is negative, the negative sign is also included in the length.</p> <p>Alteryx defaults a Fixed Decimal to 19.6. The maximum precision is 50, inclusive of the decimal point and negative sign (if applicable).</p> <p>A Fixed Decimal is the only numeric data type with an adjustable length.</p> <p>Be careful when using Fixed Decimal in the Formula tool and when converting Fixed Decimal to Float or Double. In Formula, Fixed Decimal is implicitly converted into Double. The maximum precision for the Double data type is 15 digits and for Float 7 digits. If you are converting Fixed Decimal, you need to expect to lose all the data that couldn't fit into the type you are converting to.</p>	<p>A value of 1234.567 with a length of 6.1 results in 1234.6</p> <p>A value of 1234.567 with a length of 8.3 results in 1234.567</p> <p>A value of -1234.567 with a length of 8.3 results in a field conversion error and Null output, as the value does not fit within the specified precision.</p> <p>A value of 1234.567 with a length of 11.6 results in 1234.567000</p> <p>A value of 12222222222222222222222222222222.00000 when converted to Double results in 1.222222222222e+34</p> <p>A value of 1.983274187638715245 when converted to Double results in 1.98327418763872</p>
Float	<p>A standard single-precision floating-point value. It uses 4 bytes, and can represent values from +/- 3.4 x 10<sup>-38</sup> to 3.4 x 10<sup>38</sup> with 7 digits of precision.</p> <p>A float uses a decimal that can be placed in any position and is mainly used to save memory in</p>	+/- 3.4 x 10 <sup>-38</sup> to 3.4 x 10 <sup>38</sup> with 7 digits precision

	large arrays of floating-point numbers.	
Double	<p>A standard double-precision floating-point value. It uses 8 bytes and can represent values from +/- 1.7 x 10<sup>-308</sup> to 1.7 x 10<sup>308</sup> with 15 digits precision.</p> <p>A double uses a decimal that can be placed in any position. A double uses twice as many bits as a float and is generally used as the default data type for decimal values.</p>	

### 3. Data-time

Type	Description	Example
Date	A 10 character String in "yyyy-mm-dd" format.	December 2, 2005 = 2005-12-02
Time	An 8 character String in "hh:mm:ss" format.	2:47 and 53 seconds a.m. = 02:47:53 2:47 and 53 seconds p.m. = 14:47:53
DateTime	A 19 character String in "yyyy-mm-dd hh:mm:ss" format.	

### 4. Boolean

Type	Description	Example
Bool	An expression with only two possible values: True or False.	The words 'True' and 'False' display in the results where 'False' = 0 and 'True' = non-zero.

### 5. Spatial objects

Type	Description	Example
SpatialObj	The spatial object associated with a data record. A table can contain multiple spatial object fields.	A spatial object can consist of a point, line, polyline, or polygon.

**Select**

# SELECT

Select, deselect, reorder, and rename fields, change field type or size, and assign a description.

Designer makes it very easy to change the datatype at any point in the workflow. The **Select** tool displays the columns in your dataset. Use the Select tool to change the datatype of those columns and reorder, rename, and drop columns from the datastream.

- Applying changes to columns in the data
- Select tool allows you to make changes to a columns metadata such as
  - o Size
  - o Type
  - o Description

**Questions:**

1. Choose all the following processes that can be performed with the select tool
  - a. Change a column size
  - b. Remove columns
  - c. Change a data type
  - d. Rename columns

**Answer: All**

2. True or False? The ability to reorder and rname columns, as well as change a column's data type, exists only in the Select tool.

**Answer: False**

3. Type directly into the column "Rename" to modify a column

## Browse



# BROWSE

View data from a connected tool and view data profile information, reporting snippets, maps and behavior analysis information in the data.

You need visibility into your data when solving problems. Designer provides previews of your data along the way, but if your dataset is large, that preview can be truncated. All data is still in the workflow and will be processed but displaying all the data at each step in the workflow is resource-intensive. It is far more efficient for you to use a Browse tool to view the data when and where you want. The Browse tool displays your entire dataset and offers visual aids to help you evaluate the state of your data. What you see when you open the Browse tool will vary depending on the column selected in the results window and the datatype of that column.

- Clicking the input data tool shows the data in the results window
- Larger data requires the browse tool
- By using a browse tool, all data after run is shown in the results window
- The browse tool offers interactive experience of data exploration: data profiling in the browse tools configuration window allows you to access the quality, distribution, and attributes of data in a single column

### Browse Tools Pros & Cons

#### Pros:

- In-depth views of data
- Can view the entire dataset

#### Cons:

- Memory intensive
- Increase time and processing requirements

#### Questions:

1. Select all the ways in which a Browse tool can be added to the canvas.
  - a. Drag a Browse tool on the Canvas from the Tool Palette
  - b. Right click a tool on the Canvas and select “Add Browse After”
  - c. Select a tool on the Canvas and hit CTRL + Shift + B

#### Answer: All

2. True or False? A Browse tool is required to view the entire dataset in the Results window?

#### Answer: True

**3. Click the icon in the Results window to view the data prior to changes applied by the Select tool, then click Submit.**

Record	Planting Date	Tree ID	Height (m)	Status	Health	Latin Name	Common Name	Borough Code
1	2016-06-28	606945	10	True	Good	Fraxinus pennsylvanica	green ash	4
2	2016-08-19	160321	9	True	Good	Gleditsia triacanthos var. inermis	honeylocust	4
3	2016-12-30	541347	7	True	Good	Pyrus calleryana	Callery pear	4
4	2016-07-05	613930	10	True	Good	Pyrus calleryana	Callery pear	3
5	2016-06-13	018353	4	True	Good	Prunus virginiana	'Schubert' chokecherry	4
6	2016-06-15	021173	8	True	Good	Gleditsia triacanthos var. inermis	honeylocust	1
7	2016-01-20	544698	2	True	Fair	Quercus rubra	northern red oak	3
8	2016-02-06	546240	2	True	Good	Tilia americana	American linden	3
9	2016-07-29	646348	4	True	Good	Quercus palustris	pin oak	4

**Answer: To view data prior to changes, user the input anchor tab**

## Filter



## FILTER

Query records based on an expression to split data into two streams: True (records that satisfy the expression) and False (those that do not).

As you start to read in more and more data into Designer, you will continue to increase the amount of data in your workflow. That's great because you have it all in one place, but it can get overwhelming when you are seeking specific information.

An extremely useful tool for dividing your datasets is the **Filter** tool. Using the Filter tool, you can create logical statements. The incoming dataset is evaluated against that criteria and output to either the True anchor or the False anchor.

The basic filter option helps you to construct your criteria, but you can also use the custom filter option to create more complex statements.

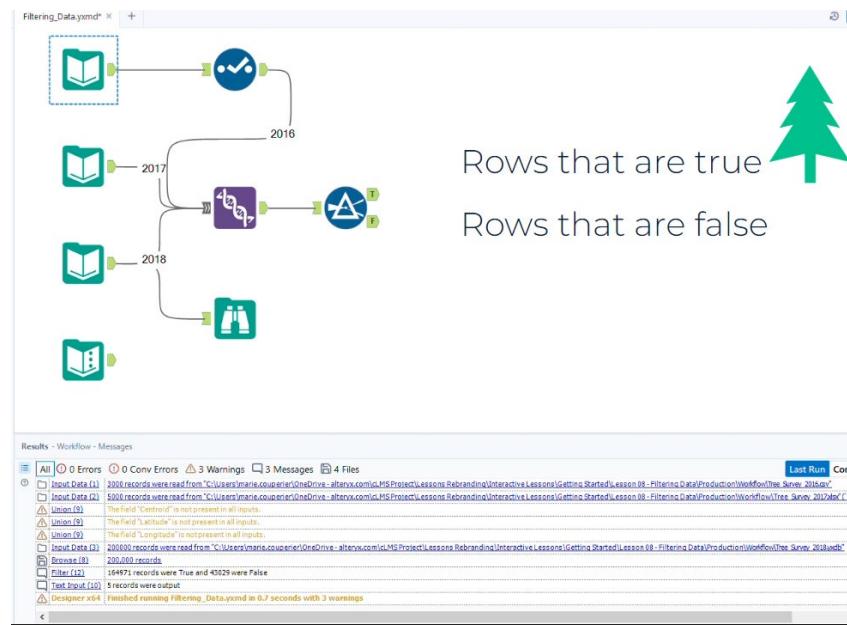
### Two types of Filters

- Basic
  - o Simple queries
  - o Evaluate single condition in one column of data
- Custom
  - o Complex queries
  - o More than one condition across multiple columns

## Basic Filter

The screenshot shows the 'Filter (11) - Configuration' screen. At the top, there are settings and a search bar. Below that, a title 'Select Basic or Custom Filter' is displayed. A radio button for 'Basic filter' is selected. Underneath are three dropdown menus: 'Column', 'Operator', and 'Condition to Test'. To the right of these are four small icons: a function icon (fx), a cancel icon (X), a document icon (File), and a print icon (Print).

After running a basic filter query, You will have two rows as a result of the filter, one for true and one for false



Output anchors: T & F

Click to view data for True and False to the Filter

## Custom Filter

The screenshot shows the 'Select Basic or Custom Filter' dialog. It has two options: 'Basic filter' and 'Custom filter'. The 'Custom filter' option is selected. Below it, there are two filter components: '[Health] = "Good"' and '[Height m] >= 10'. A blue callout box points to the first component with the text: 'String values in expressions must be quoted in Custom filters, while numeric values do not.'

**Custom filter**

- [fx] [Health] = "Good"
- [x] [Height m] >= 10

## Questions:

- True or False? The Filter tool includes two output anchors

Answer: True

- Drag the components of a Basic Filter into the appropriate places in the Configuration window.

The screenshot shows the 'Basic filter' configuration window. At the top, there are three green buttons: 'CONTAINS', '\"Oak\"', and '[Common Name]'. Below them, the 'Basic filter' configuration is shown with three input fields. The first field contains '[Common Name]' and has a dropdown arrow pointing down.

Answer:

The screenshot shows the 'Basic filter' configuration window with the components rearranged. The first field now contains '[Common Name]' and has a dropdown arrow pointing up. The second field contains 'CONTAINS' and has a dropdown arrow pointing down. The third field contains '\"Oak\"'.

3. Select the expression that was used in the Filter tool to create the True output below.

Borough Code	Health
1	Good
3	Good
4	Good
2	Good

- a. [Borough Code] > 0
- b. [Borough Code] = 1 OR [Health] = "Good"
- c. [Borough Code] = 1 AND [Health] = "Good"

Answer: b

## Sort



## SORT

Sort records based on the values in one or more fields.

Organize your large datasets by sorting information in ascending or descending order. The **Sort** tool's configuration window is simple but powerful. Select the column to be sorted, then the order for sorting.

You can select multiple columns for sorting in a single tool. The Sort tool will work on the first column listed in its configuration, then move onto the next in order.

Once dragging on the Sort tool:

Name and Order (ascending or descending)

Fields		
	Name	Order
▶*		▼

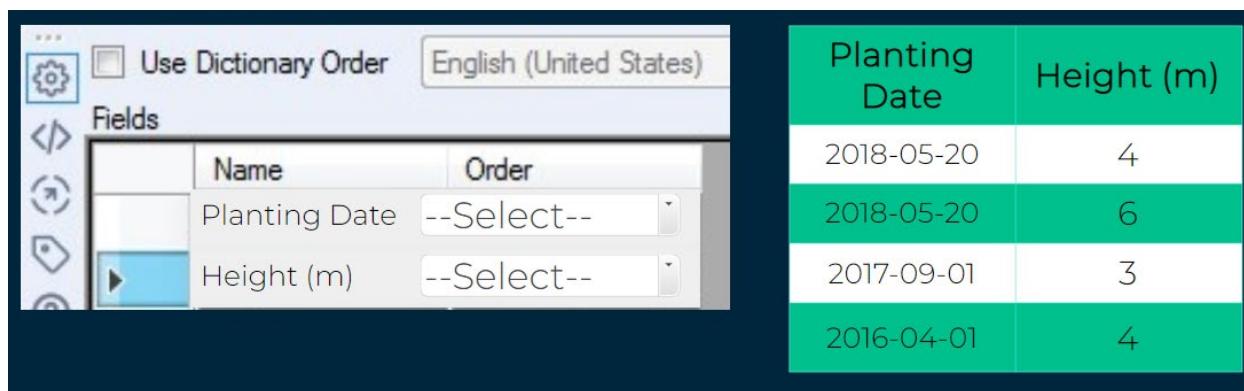
- The default sort order is Ascending
- You can sort multiple columns
- When Sorting a string ASCII values will be first then dictionary order A-Z
- Dictionary Order: Orders values how they would be in a dictionary, regardless of case

Questions:

1. True or False? The Sort tool can only sort one column of data?

Answer: False

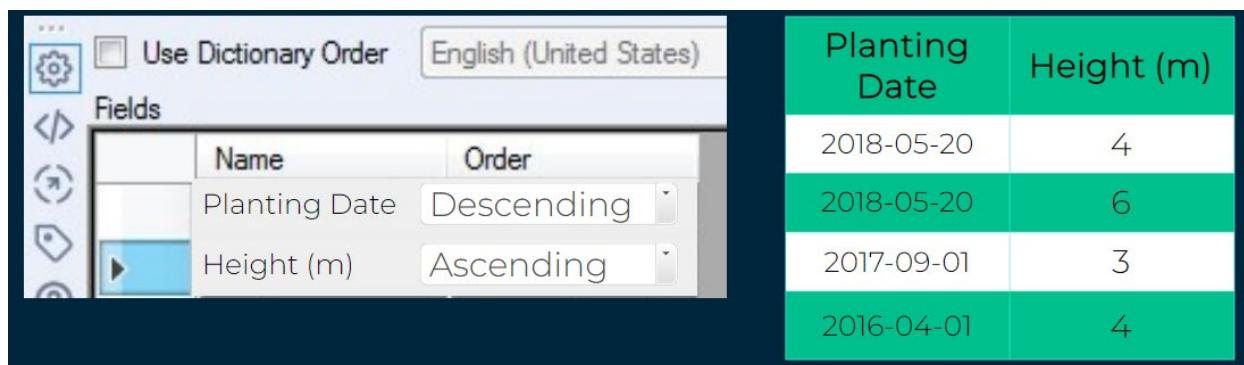
2. Configure the Sort tool to produce the results table using the Drop Down menus in the Sort tool's configuration window.



The screenshot shows the Tableau interface with the Sort tool configuration window open. The 'Use Dictionary Order' checkbox is checked, and the language is set to 'English (United States)'. The 'Fields' section contains two rows: 'Planting Date' and 'Height (m)', each with a dropdown menu labeled '--Select--'. To the right is a results table with four rows:

Planting Date	Height (m)
2018-05-20	4
2018-05-20	6
2017-09-01	3
2016-04-01	4

Answer:



The screenshot shows the Tableau interface with the Sort tool configuration window open. The 'Use Dictionary Order' checkbox is checked, and the language is set to 'English (United States)'. The 'Fields' section contains two rows: 'Planting Date' and 'Height (m)'. The 'Planting Date' row has 'Descending' selected in the dropdown. The 'Height (m)' row has 'Ascending' selected in the dropdown. To the right is a results table with four rows, showing the data sorted by Planting Date in descending order and Height in ascending order:

Planting Date	Height (m)
2018-05-20	4
2018-05-20	6
2017-09-01	3
2016-04-01	4

3. Based on the results below select the method that was used to sort the string values

Latin Name
Acer Platanoides
Acer Rubrum
Quercus Palustris
Ulmus Americana
acer plantanoides
ulmus americana

- Default Sort Behavior
- Use Dictionary Order

**Answer: Default Sort Behavior**

### Sample



## SAMPLE

Limit the data stream to a number, percentage, or random set of records.

After sorting data, you may be interested in a subset of the data. The **Sample** tool provides options for isolating a subset of data.

Use the radio button to select one of the configuration options, then set the value for N. The Sample tool's output will only include the specified data and drop the rest.

### Why Sample?

- Subset of Information
- Easy way to remove empty rows from the top or bottom of a dataset
- Pairing down data based on a pattern (e.g. selecting the first record of a group)
- Creating Dynamic random samples of data

6 different methods to Sample data:

The screenshot shows a user interface for selecting a sample type. At the top, a title bar says "Select Sample Type". Below it is a list of six sampling methods, each with a radio button:

- First N rows
- Last N rows
- Skip 1st N rows
- 1 of every N rows
- 1 in N chance to include each row
- First N% of rows

Below the list is a section titled "Group by column (optional)" with a "Select all" button. It contains four checkboxes:

- Search Date (highlighted with a red box)
- Product
- Type
- Price

A numeric input field "N = 3" with up and down arrows is also present.

Only option to generate randomness in a sample:

The screenshot shows the same "Select Sample Type" interface. The "1 in N chance to include each row" option is highlighted with a blue circle.

#### Questions:

1. All the following sampling methods are available in the Sample tool except:
  - a. Last N
  - b. First N

- c. Every Row
- d. Skip 1<sup>st</sup> N
- e. First N%
- f. 1 of every N rows

**Answer: Every Row**

2. Which of the following sample tool configurations would produce the results from this input dataset run after run?
- a. 1 in N chance to include each row
  - b. Skip 1<sup>st</sup> N rows
  - c. 1 of every N rows
  - d. First N rows

The image displays four windows arranged in a 2x2 grid, illustrating the results of two runs of a sample tool.

**Run 1:** This row shows the results of running the tool on the full input dataset (5 records).

- Input Data (5) - Output:** Shows 165 records displayed. The table has columns: Record, Song ID, Artist, and Lifetime Estimate. The data is as follows:
 

Record	Song ID	Artist	Lifetime Estimate
1	7	Artist 8	885685
2	12	Artist 9	618387
3	13	Artist 13	584104
4	25	Artist 20	161751
5	25	Artist 20	161751
6	27	Artist 15	931193
7	28	Artist 14	715976
8	34	Artist 9	187292
9	35	Artist 18	814841
10	40	Artist 2	464383
11	48	Artist 16	961
- Sample (2) - Output:** Shows 31 records displayed. The table has columns: Record, Song ID, Artist, and Lifetime Estimate. The data is as follows:
 

Record	Song ID	Artist	Lifetime Estimate
1	62	Artist 8	46029
2	75	Artist 7	909764
3	109	Artist 13	655866
4	124	Artist 16	297318
5	133	Artist 13	54005
6	133	Artist 13	54005
7	137	Artist 11	798932
8	137	Artist 11	798932
9	156	Artist 18	978032
10	198	Artist 16	203927
11	209	Artist 10	35815

**Run 2:** This row shows the results of running the tool on a sample of the input dataset (2 records).

- Sample (2) - Output:** Shows 40 records displayed. The table has columns: Record, Song ID, Artist, and Lifetime Estimate. The data is as follows:
 

Record	Song ID	Artist	Lifetime Estimate
1	25	Artist 20	161751
2	49	Artist 1	111190
3	62	Artist 8	46029
4	75	Artist 7	909764
5	110	Artist 11	734718
6	119	Artist 9	923265
7	124	Artist 16	297318
8	133	Artist 13	54005
9	137	Artist 11	798932
10	141	Artist 3	658442
11	153	Artist 20	158366

**Answer: a**

3. True or False? “1 in N chance to include each row” means that the resulting number of records will be equivalent to  $1/N$  of the records in the original dataset.

**Answer: False** (“1 in N chance to include each row” does not represent an exact fraction of the incoming records. To include  $1/N$  of all rows, select “1 of every N rows.”)

## Practice Exercise 1

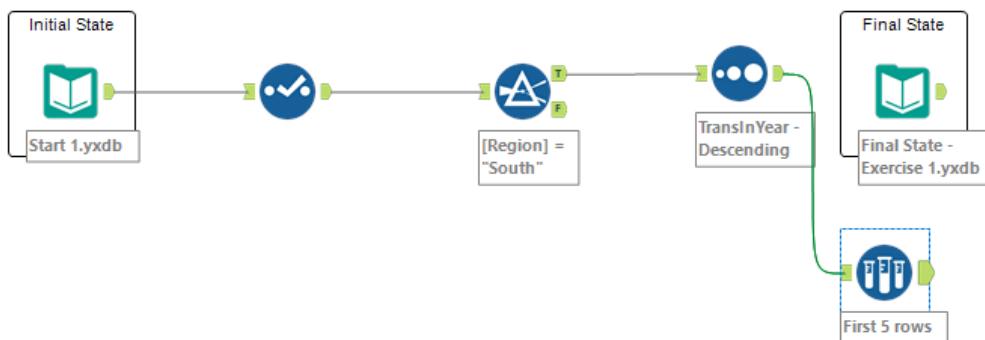
Practice Exercise 1.yxmd × +

# Practice Exercise 1

Find the five customers with the highest number of transactions in the Southern region and remove the "Weekly Sales" and "Store Volume" columns.

An Input Data tool contains 100 records, each representing a unique customer's transaction information. The Customer ID (a unique identifier for each customer) and Customer Segment provide basic information about the customer. The transactions column represents the average number of transactions that customer has made over the past year.

In this exercise, drop the columns "Weekly Sales" and "Store Volume". Then find the 5 customers with the highest average transactions in the "South" region. Your results window should match the Final State Input Data tool.



Results - Sample (16) - Output

5 of 5 Fields | Cell Viewer | 5 records displayed | ↑ ↓ |

	Record	Customer_ID	Store_Number	Customer_Segment	Region	TransInYear
1	1128	108	Consumer	South	150	
2	2911	107	Home Office	South	100	
3	787	104	Home Office	South	86	
4	483	105	Corporate	South	50	
5	466	104	Small Business	South	50	

Answer: Select → Filter → Sort → Sample

## Text to Columns



### TEXT TO COLUMNS

Split the text from one field into separate rows or columns.

Sometimes, information that should be separate is combined into a single cell, making it difficult to filter, sort, or sample. In those instances, it makes sense to split that information into columns or rows.

If the values are separated by a delimiter, the **Text to Columns** tool is a quick way to split data into columns or rows. A delimiter is a character that divides values. Commas, spaces, tabs, pipes, and many other characters can be used as a delimiter.

Single Cell with delimiters:

2018-03-15,2020-09-15,PE|Peru,45000000

Column 1

2018 03,15|PE@Peru|45000000

Result

2018	03	15	PE	Peru	45000000
------	----	----	----	------	----------

An underscore may also be used as a delimiter

, is Default delimiter

Manually Enter delimiters in the Delimiters field. In the below example both “,” and “|” are used

**Text To Columns (2) - Configuration**

Select Column to Split

Column to split: Loan Data Delimiters: |

Split to columns

Number of columns: 3  
Extra characters: Leave extra in last column  
Output root name: Loan Data

Split to rows

Advanced options:

- Ignore delimiters
- Ignore delimiters in single quotes
- Ignore delimiters in parentheses
- Ignore delimiters in brackets
- Skip empty columns

2018-02-15,2036-02-15,BR|Brazil,300000000

Next, Choose # of Output columns (e.g. 5)

**Text To Columns (2) - Configuration**

Select Column to Split

Column to split: Loan Data Delimiters: |

Split to columns

Number of columns: 5  
Extra characters: Leave extra in last column  
Output root name: Loan Data

Split to rows

Advanced options:

- Ignore delimiters in quotes
- Ignore delimiters in single quotes
- Ignore delimiters in parentheses
- Ignore delimiters in brackets

2018-02-15,2036-02-15,BR|Brazil,300000000

In the evenly leftover data does not fit in the specified number of columns, choose what to do with extra characters:

Select Column to Split

Column to split: Loan Data

Delimiters: ,

Split to columns (radio button selected)

Number of columns: 5

Extra characters: Leave extra in last column

Output root name:

Split to rows (radio button unselected)

Error (selected): Leave extra in last column

Advanced options:

- Ignore delimiters in quotes
- Ignore delimiters in single quotes
- Ignore delimiters in parentheses
- Ignore delimiters in brackets

By Default, Extra characters will be “Leave extra in last column”

After running workflow:

Record	Loan ID	Loan Data	Loan Data1	Loan Data2	Loan Data3	Loan Data4	Loan Data5
1	1	2018-02-15,2036-02-15,BR Brazil,30000000	2018-02-15	2036-02-15	BR	Brazil	30000000
2	2	2018-03-15,2030-09-15,PE Peru,15000000	2018-03-15	2030-09-15	PE	Peru	15000000
3	3	2018-03-15,2020-09-15,PE Peru,45000000	2018-03-15	2020-09-15	PE	Peru	45000000
4	4	2018-04-15,2033-04-15,JD Indonesia,20000000	2018-04-15	2033-04-15	ID	Indonesia	20000000
5	5	2018-04-15,2036-10-15,CO Colombia,636500000	2018-04-15	2036-10-15	CO	Colombia	636500000
6	6	2018-05-15,2029-05-15,AL Albania,21600000	2018-05-15	2029-05-15	AL	Albania	21600000
7	7	2018-05-15,2033-11-15,HR Croatia,103500000	2018-05-15	2033-11-15	HR	Croatia	103500000
8	8	2018-07-15,2031-01-15,RU Russian Federation,2...	2018-07-15	2031-01-15	RU	Russian Federation	20000000
9	9	2018-08-01,2033-08-01,JD Indonesia,50000000	2018-08-01	2033-08-01	ID	Indonesia	50000000
10	10	2018-08-15,2043-02-15,EC Ecuador,10000000	2018-08-15	2043-02-15	EC	Ecuador	10000000
11	11	2018-09-15,2037-03-15,CN China,100000000	2018-09-15	2037-03-15	CN	China	100000000
12	12	2018-09-15,2031-03-15,NI India,216000000	2018-09-15	2031-03-15	IN	India	216000000
13	13	2018-09-15,2029-03-15,BY Belarus,90000000	2018-09-15	2029-03-15	BY	Belarus	90000000
14	14	2018-10-01,2025-04-01,RO Romania,91800000	2018-10-01	2025-04-01	RO	Romania	91800000
15	15	2018-10-15,2025-04-15,LB Lebanon,15000000	2018-10-15	2025-04-15	LB	Lebanon	15000000
16	16	2018-11-01,2039-11-01,EG Egypt," Arab Republi...	2018-11-01	2039-11-01	EG	Egypt," Arab Republic of	200000000
17	17	2018-11-15,2043-05-15,CN China,80000000	2018-11-15	2043-05-15	CN	China	80000000

Advanced options to handle above issue

Select Column to Split

Column to split: Loan Data

Delimiters: ,

Split to columns

Number of columns: 5

Extra characters: Leave extra in last column

Output root name: Loan Data

Split to rows

Advanced options:

- Ignore delimiters in quotes
- Ignore delimiters in single quotes
- Ignore delimiters in parentheses
- Ignore delimiters in brackets
- Skip empty columns

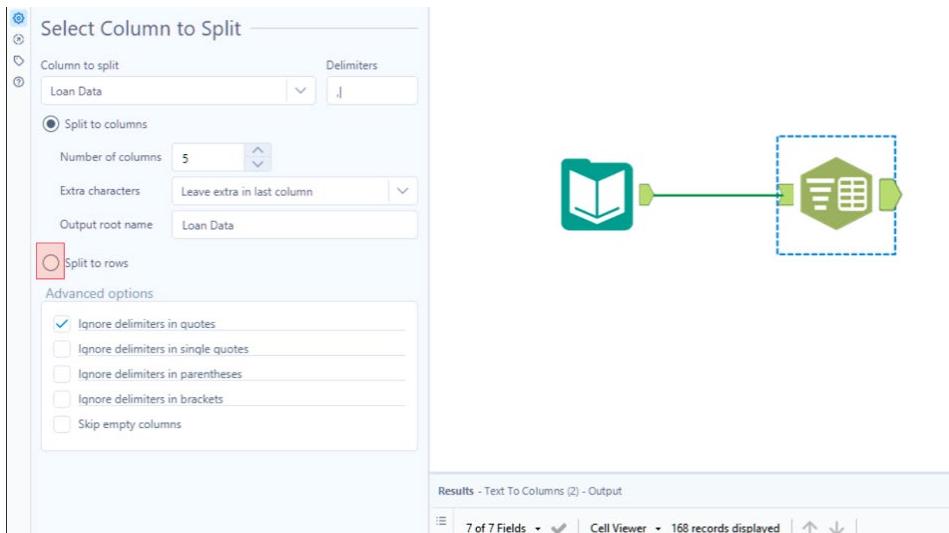


### Re-running the workflow:

Record	Loan ID	Loan Data	Loan Data1	Loan Data2	Loan Data3	Loan Data4	Loan Data5
12	12	2018-09-15,2031-03-15,IN India,216000000	2018-09-15	2031-03-15	IN	India	216000000
13	13	2018-09-15,2029-03-15,BY Belarus,90000000	2018-09-15	2029-03-15	BY	Belarus	90000000
14	14	2018-10-01,2025-04-01,RO Romania,91800000	2018-10-01	2025-04-01	RO	Romania	91800000
15	15	2018-10-15,2025-04-15,LB Lebanon,15000000	2018-10-15	2025-04-15	LB	Lebanon	15000000
16	16	2018-11-01,2039-11-01,EG Egypt," Arab Republi...	2018-11-01	2039-11-01	EG	Egypt," Arab Republic of	200000000
17	17	2018-11-15,2043-05-15,CN China,80000000	2018-11-15	2043-05-15	CN	China	80000000
18	18	2018-12-15,2036-12-15,AR Argentina,200000000	2018-12-15	2036-12-15	AR	Argentina	200000000
19	19	2019-02-01,2031-08-01,MK Macedonia," former...	2019-02-01	2031-08-01	MK	Macedonia," former Yugoslav Republic of	24000000
20	20	2019-03-15,2033-09-15,PH Philippines,10000000	2019-03-15	2033-09-15	PH	Philippines	10000000
21	21	2019-03-15,2037-09-15,GE Georgia,33500000	2019-03-15	2037-09-15	GE	Georgia	33500000
22	22	2019-03-15,2043-03-15,TR Turkey,67000000	2019-03-15	2043-03-15	TR	Turkey	67000000
23	23	2019-03-15,2044-03-15,BR Brazil,400000000	2019-03-15	2044-03-15	BR	Brazil	400000000
24	24	2019-04-01,2034-04-01,ID Indonesia,149980000	2019-04-01	2034-04-01	ID	Indonesia	149980000
25	25	2019-05-15,2041-05-15,MA Morocco,50000000	2019-05-15	2041-05-15	MA	Morocco	50000000
26	26	2019-05-15,2031-05-15,PL Poland,1307800000	2019-05-15	2031-05-15	PL	Poland	1307800000
27	27	2019-05-15,2038-11-15,CN China,100000000	2019-05-15	2038-11-15	CN	China	100000000
28	28	2019-06-15,2038-12-15,CN China,200000000	2019-06-15	2038-12-15	CN	China	200000000

Text to columns can also be split to rows

- When splitting into rows, it is important to have a unique identifier (e.g. Loan ID)



**Select Column to Split**

Column to split: Loan Data

Delimiters: ,

Split to columns

Number of columns: 5

Extra characters: Leave extra in last column

Output root name: Loan Data

Split to rows

Advanced options:

- Ignore delimiters in quotes
- Ignore delimiters in single quotes
- Ignore delimiters in parentheses
- Ignore delimiters in brackets
- Skip empty columns

Results - Text To Columns (2) - Output

7 of 7 Fields | Cell Viewer | 168 records displayed | ↑ ↓ |

Record	Loan ID	Loan Data
1	1	2018-02-15
2	1	2036-02-15
3	1	BR
4	1	Brazil
5	1	300000000
6	2	2018-03-15
7	2	2030-09-15
8	2	PE
9	2	Peru
10	2	150000000
11	3	2018-03-15
12	3	2020-09-15
13	3	PE
14	3	Peru
15	3	450000000
16	4	2018-04-15
17	4	2033-04-15

## Questions

1. Select the characters used to parse the data, as shown below. Select all that apply.
  - Semi-colon
  - Hyphen
  - Space
  - Comma

■ Semi-colon	BEFORE				
		Composer			
■ Hyphen	AFTER	van Beethoven,Ludwig;1770-1827			
■ Space		Composer_A	Composer_B	Composer_C	Composer_D
■ Comma		van Beethoven	Ludwig	1770	1827

Answer: a, b, d

2. Values in the data below are comma-delimited. How many columns should be specified for splitting the data if delimiters in quotes and brackets are ignored?
- a. 7
  - b. 4
  - c. 6
  - d. 5

Vehicle Information
1HGBH41JXMN109186, Mike's Auto"," LLC, [24,000], Mobile, AL

Answer: d

3. Based on the data before and after parsing, select the method in the Text to Columns tool that was used.
- a. Split to Columns
  - b. Split to Rows

Answer: b

## Unique



# UNIQUE

Separate data into two streams,  
duplicate and unique records, based  
on the fields of the user's choosing.

Another common need when analyzing data is finding the unique values within a dataset. The **Unique** tool will divide the dataset into unique values and duplicate values.

Selecting a single column in the Unique tool's configuration window will evaluate values in that column only. Selecting multiple columns will evaluate the combination of values and determine if the combination is unique.

### Removing Duplicate Values

U = Unique

D = Duplicate



Configuration Window shows columns to select

The screenshot shows the configuration window for the Unique tool. The top section asks "Select columns to find unique values from:". Below it, a "Column Names" section lists three options: "Date", "Machine Name", and "Session Length". Each option has a checkbox next to it. The "Date" checkbox is highlighted with a red box, indicating it is selected. There is also a "Select All" button at the bottom of the list.

- When multiple columns are selected (i.e. Date and Machine name), the values in each selected columns are combined and then compared across the dataset
- All columns in the data set can be selected
- Unique tool is case sensitive

Name	Type	Time
AA	1	00:50
AA	2	13:20
BC	3	17:35
AA	1	00:50

## Questions

1. How many unique rows would be identified if the Unique tool was configured as below?
  - a. 2
  - b. 3
  - c. 4

First	Last	MI
Jonnie	Good	B
Johnie	Good	B
Jon	goode	E
jonne	good	B
Jonnie	Good	b

Answer: c

2. Which of the following statements about the Unique tool are NOT true? Select all that apply.
  - a. You can select up to 10 columns in on Unique tool
  - b. It's best practice to join the results from the U & D outputs
  - c. The Unique tool is case sensitive

Answer: a, b

3. True or False? The Unique Tool is case sensitive and will display the last value from the dataset in the U output when duplicate values are found.
  - a. True
  - b. False

Answer: False

## Union



# UNION

Combine two or more data streams with similar structures based on field names or position. In the output, each column will contain the data from each input.

When inputting more than one dataset into a single workflow, you will likely need to combine those datasets. The **Union** tool combines datasets vertically by name, by position, or manual configuration to align columns of data.

The Union tool's input anchor accepts multiple inputs and even includes an option to set a specific output order.

### Blending Data with Unions

- A Union stacks incoming data streams on top of each other based on matching column names, column position, or manual alignment

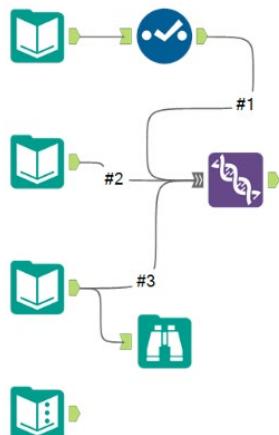
#### Benefits:

- Unified separate datasets
- Avoids duplicate processes
- Enables dynamic workflows

#### Input Anchor

- The input anchor contains 3 arrows showing that it can take in multiple data streams
- First input connected determines the output column names and data types

Connection strings are ordered by the number in which the inputs were connected to the union tool



- Double click the number to rename the connection string

### Vertical Alignment (1 of 3 ways)

- Column Name
- Column Position
- Manually

Aligning by column name is the default configuration of the Union tool

#### Column Name

2017 Tree Data			2018 Tree Data		
TREE ID	PLANTING DATE	LATIN NAME	LATIN NAME	TREE ID	PLANTING DATE
390375	2017-10-28	Pyrus Calleryana	Tilia Cordata	009232	2018-11-16
235695	2017-09-18	[Null]	Quercus Bicolor	575743	2018-10-28
175013	2017-08-25	Tilia Cordata	Ulmus Parvifolia	443336	2018-03-14
190756	2017-01-10	Prunus	Acer Platanoides	580367	2018-007-02
307086	2017-05-17	Acer Robrum	Prunus	988132	2018-09-07

Stacked vertically by name:

TREE ID	PLANTING DATE	LATIN NAME
390375	2017-10-28	Pyrus Calleryana
235695	2017-09-18	[Null]
175013	2017-08-25	Tilia Cordata
190756	2017-01-10	Prunus
307086	2017-05-17	Acer Robrum
009232	2018-11-16	Tilia Cordata
575743	2018-10-28	Quercus Bicolor
443336	2018-03-14	Ulmus Parvifolia
580367	2018-007-02	Acer Platanoides
988132	2018-09-07	Prunus

## Column Position

2017 Tree Data			2018 Tree Data		
TREE ID	PLANTING DATE	LATIN NAME	TREE ID	PLANTING DATE	LATIN NAME
390375	2017-10-28	Pyrus Calleryana	009232	2018-11-16	Tilia Cordata
235695	2017-09-18	[Null]	575743	2018-10-28	Quercus Bicolor
175013	2017-08-25	Tilia Cordata	443336	2018-03-14	Ulmus Parvifolia
190756	2017-01-10	Prunus	580367	2018-007-02	Acer Platanoides
307086	2017-05-17	Acer Robrum	988132	2018-09-07	Prunus

Stacked by order of position. It is important that the order of the columns match before entering the union tool

Tree Data UNION					
TREE ID	PLANTING DATE	LATIN NAME	TREE ID	PLANTING DATE	LATIN NAME
390375	2017-10-28	Pyrus Calleryana	009232	2018-11-16	Tilia Cordata
235695	2017-09-18	[Null]	575743	2018-10-28	Quercus Bicolor
175013	2017-08-25	Tilia Cordata	443336	2018-03-14	Ulmus Parvifolia
190756	2017-01-10	Prunus	580367	2018-007-02	Acer Platanoides
307086	2017-05-17	Acer Robrum	988132	2018-09-07	Prunus

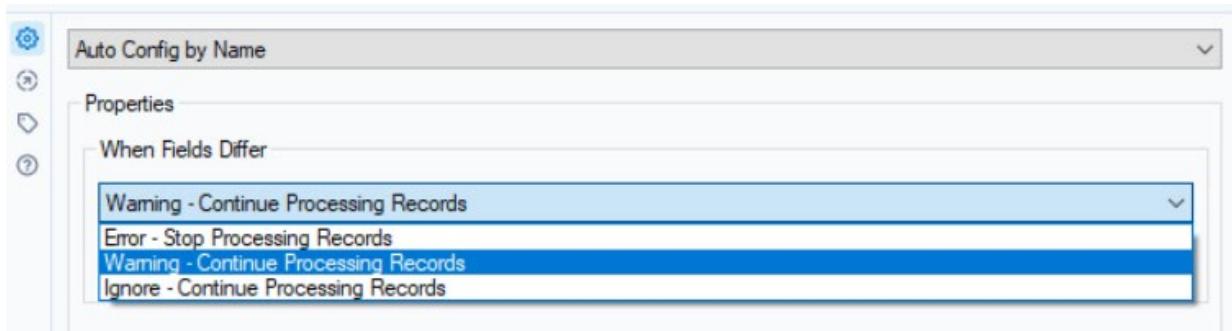
## Manually

The screenshot shows a 'Manually Configure Fields' dialog box. On the left, there's a sidebar with icons for properties, output columns, and help. The main area has tabs for 'Properties' and 'Output Columns'. Under 'Output Columns', there's a table with two rows. Row #2 contains columns: Col1, Col2, Col3, Col4, Col5, Col6, Col7, Col8, Col9, Col10, Col11, and Col12. Row #1 contains columns: Latitude, Longitude, Planting Date, Tree ID, Height (m), Status, Health, Latin Name, Common Name, Borough Code, and an empty column. A 'Reset' button and a set of navigation arrows are at the top right of the table.

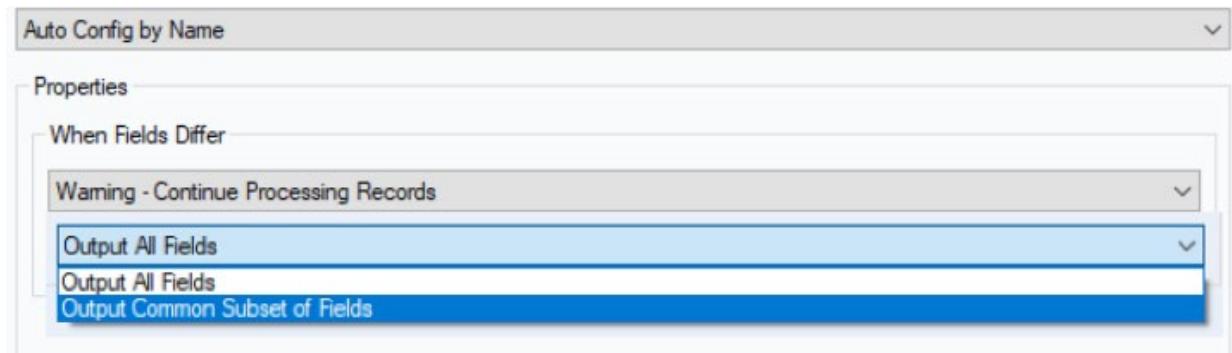
Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12
#2	Latitude	Longitude	Planting Date	Tree ID	Height (m)	Status	Health	Latin Name	Common Name	Borough Code	
#1	Latitude	Longitude	Planting Date	Tree ID	Height (m)	Status	Health	spc_latin	spc_common	Borough Code	

When the number of columns in the datasets differ, you can choose how they are handled in outputs

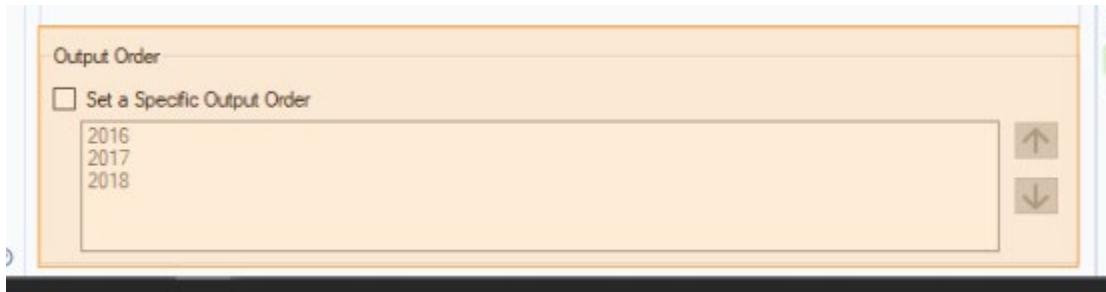
- By default, workflow processing will not stop but you will receive a warning in the messages window when running the workflow



Additionally, you can choose the output for all of the fields or only common fields



By default output values will be stacked in the order in which they were connected to the union tool



You can click "Set a Specific Order" and move the data to any order that you would like based off of your connection strings

## Questions

1. Which of the following is not an available option for combining data in the Union tool?
  - a. Data Type
  - b. Column Name
  - c. Column Position
  - d. Manual Configuration

**Answer: A**

2. Two inputs have been combined with the Union tool to produce an output. Based on the results, which configuration option was used to output the data?
- a. Output Common Subset of Fields
  - b. Output All Fields



Type	Height
Oak	9
Pine	7

Type	Health
Elm	Good
Maple	Fair

Type	Height	Health
Oak	9	[Null]
Pine	7	[Null]
Elm	[Null]	Good
Maple	[Null]	Fair

**Answer: b**

3. Based on the configuration below, how many rows will be output from the Union tool?
- a. 5
  - b. 3
  - c. 4
  - d. 0



» Auto Config by Position

Type	Height
Oak	9
Pine	7
Ash	4

Name	Health
Elm	Good
Maple	Fair

**Answer: 5**

## Join



# JOIN

Combine two data streams based on common fields (or record position). In the joined output, each row will contain the data from both inputs.

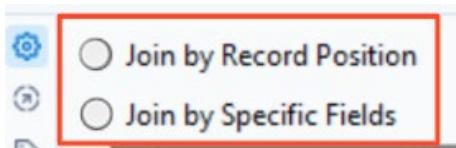
If you need to combine data horizontally, the **Join** tool can utilize a common field or combine by position. If both incoming datasets share a common column, joining on that column can be used to match rows of data. Alternatively, if you are confident that the row order of the datasets match, you can join by record position. This tool is very powerful and makes it easy to work with multiple data sources or combine disparate data streams.



L = Left

R = Right

- Join tool can associate rows in one data stream to rows in another data stream by:
  - o Record Position
  - o Specific Columns



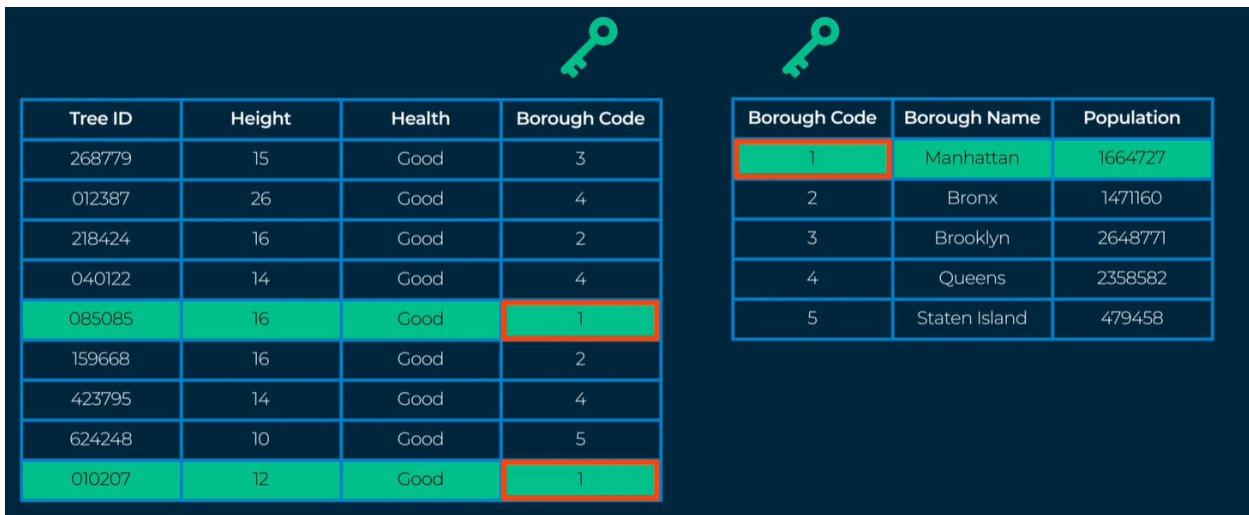
## Join By Position

JOIN BY POSITION		
Value A	Value B	Value C
wavy	wavy	wavy

Value A	Value D
wavy	wavy

## Join by specific fields



The diagram illustrates a join operation between two datasets. On the left, a table of tree data includes columns for Tree ID, Height, Health, and Borough Code. Two specific rows are highlighted: the 5th row (Tree ID 085085) and the 9th row (Tree ID 010207). The Borough Code column for both these rows is highlighted with a red box. On the right, a table of borough data includes columns for Borough Code, Borough Name, and Population. The first two rows (Borough Codes 1 and 2) are highlighted with a red box. The Borough Code column for the 5th row (Staten Island) is also highlighted with a red box. This visualizes how specific field values from one dataset are used to find matching records in another.

Tree ID	Height	Health	Borough Code
268779	15	Good	3
012387	26	Good	4
218424	16	Good	2
040122	14	Good	4
085085	16	Good	1
159668	16	Good	2
423795	14	Good	4
624248	10	Good	5
010207	12	Good	1

Borough Code	Borough Name	Population
1	Manhattan	1664727
2	Bronx	1471160
3	Brooklyn	2648771
4	Queens	2358582
5	Staten Island	479458

Join (10) - Configuration

Join by Record Position

Join by Specific Fields

Left Right

Left Right

L R R

- Select the corresponding column in the Right input to join data
- Matching column names automatically populate
- More than one column can be used to join data

Within the Join tool there is an embedded select window

- You can remove fields
- Rename fields
- Change order of outgoing data

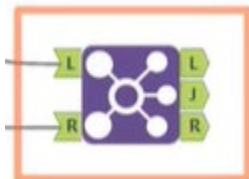
The screenshot shows the 'Join (10) - Configuration' dialog box. At the top, there are two radio button options: 'Join by Record Position' (unchecked) and 'Join by Specific Fields' (checked). Below this is a table with two columns, 'Left' and 'Right', showing a single join condition: 'Borough Code' from the left is joined with 'Borough Code' from the right. Below the table are three circular icons labeled L, J, and R, representing Left, Join, and Right data streams respectively. A red box highlights the 'Options' dropdown menu and the 'Description' column of the table below. The 'Description' column contains the names of the joined fields: 'Right\_Borough Code', 'Borough Name', 'Borough Population', and '\*Unknown'. A tip at the bottom of the table says: 'TIP: To reorder multiple rows: select, right-click and drag.'

	Input	Field	Type	Size	Rename	Description
<input checked="" type="checkbox"/>	Left	Planting Date	Date	10		
<input checked="" type="checkbox"/>	Left	Tree ID	V_String	255		
<input checked="" type="checkbox"/>	Left	Height (m)	Double	8		
<input checked="" type="checkbox"/>	Left	Status	Bool	1		
<input checked="" type="checkbox"/>	Left	Health	V_String	255		
<input checked="" type="checkbox"/>	Left	Latin Name	V_WString	255		
<input checked="" type="checkbox"/>	Left	Common Name	V_WString	255		
<input checked="" type="checkbox"/>	Left	Borough Code	V_WString	255		
<input checked="" type="checkbox"/>	Right	Borough Code	Int32	4	Right_Borough Code	
<input checked="" type="checkbox"/>	Right	Borough Name	V_String	214...		
<input checked="" type="checkbox"/>	Right	Borough Population	Int32	4		
<input checked="" type="checkbox"/>		*Unknown	Unknown	0		Dynamic or Unknown

- Only fields of the same data type can be used to join data together or you will receive an error message

Join tool has 3 output data streams

- J = Join (data joined between data sets)
- L = Left (Data that did not join from the left)
- R = Right (Data that did not join from the right)

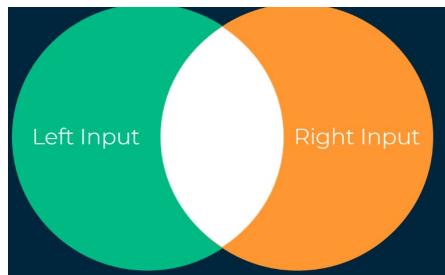


**Questions**

1. True or False? Columns classified as a numeric data type may be joined to columns classified as any data type when joining by specific fields?
  - a. True
  - b. False

Answer: b

2. Select the area in the diagram that represents the data from the Join tool's L output anchor.
  - a. Left Input
  - b. Right Input



Answer: a

3. Based on the results below, select the method that was used to join the two tables.
  - a. Join by Position
  - b. Join by Specific Fields

Left Input		Right Input	
Type	Height	Type	Health
Oak	9	Pine	Good
Pine	7	Oak	Fair

Results from J Output Anchor



Type	Height	Health
Oak	9	Good
Pine	7	Fair

Answer: a

## Practice Exercise 2

Practice Exercise 2

Combine both sets of Customer Transactions, then combine those entries with the User Details information.

Combine the Customer Transaction data in "Start 2.1" and "Start 2.2" and ensure there are no duplicate users. Then combine those results with the "User Details". The "User Details" information is delimited by pipes. | (ctrl + |).

```

graph LR
    Start1[Initial State  
Start 2.1.yxdb  
Start 2.2.yxdb] -- "#1" --> Union1(( ))
    Start2[User Details] --> Union1
    Union1 --> Join1(( ))
    Join1 --> Final[Final State -  
Exercise 2.yxdb]
    Join1 --> Union2(( ))
    Start1 -- "#2" --> Union2
    Union2 --> Join2(( ))
    Join2 --> Final
  
```

**Results - Join (28) - Out - Join**

Record	Customer_ID	Store_Number	Customer_Segment	Region	Annual Transactions	Name	Address	City	Zip
1	1128	108	Consumer	South	150	Bob	5847 Union St	Arvada	80004
2	1319	108	Consumer	South	150	Hudson	5595 S Kenton Way	Englewood	80111
3	2911	107	Home Office	South	100	Lou	2751 Wolff St	Denver	80212
4	466	104	Small Business	South	50	Nancy	2721 S Truckee St	Aurora	80013
5	483	105	Corporate	South	50	Drew	12801 E Kansas Pl	Aurora	80012
6	787	104	Home Office	South	86	Jack	915 S Lima St	Aurora	80012

**Answer:**

**Top: Union → Unique → Join**

**Bottom: Text to Columns → Select → Join**

**Join created off Customer\_ID**

## Date**Time**



# DATETIME

Transform date/time data to and from a variety of formats, including both expression-friendly and human readable formats.

Similar to the way some values need to be split in order to be as useful as possible, datetime values need to be formatted properly in order to be most useful. Designer contains functions which can calculate time intervals without needing extra conversion calculations between units. In order to use those functions, datetime values need to be formatted into a specific order.

The Select tool is a go-to for changing datatypes, but the requirement that the characters be arranged in the correct order is something the Select tool cannot achieve. The **DateTime** tool, however, can easily convert string data into properly formatted datetime data and vice versa. Simply select the direction of conversion and the format of the string value.

**DateTime (15) - Configuration**

Select the format to convert

Date/Time format to string

String to Date/Time format

Select the date/time field to convert

1

Select the new column name

3

Select the format for the new column

2

Specify a custom format for the new column

Example      Output

New Workflow2\* X

```

graph LR
    Start(( )) --> Convert[Convert Date To: yyyy-MM-dd]
    Convert --> End(( ))
  
```

Results - DateTime (15) - Output

Record #	Date	DateTime_String
1	1992-08-28	August 28, 1992
2	1991-10-12	October 12, 1991

Converting a date contained in a text field to a proper date format works similarly: you select the field to be formatted, the string that matches the current format and what the output field should be called.

The screenshot shows the Workbench interface with three main panels:

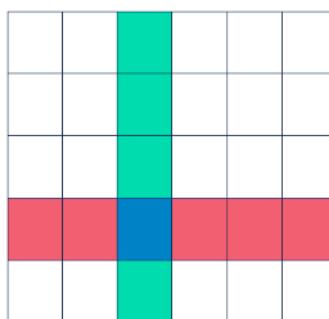
- Left Panel (DateTime (21)-Configuration):** A configuration dialog for a "Convert Date To" tool. It has three tabs: "Select the format to convert" (radio button selected for "String to Date/Time format"), "Select the string field to convert" (set to "DateTime\_String" with a circled "1"), and "Specify the new column name" (set to "DateTime\_Out" with a circled "3"). Below these are dropdowns for "Select the format that matches the incoming string field" (containing various date formats like "dd/MM/yyyy" and "Month dd, yyyy") and "Specify the format of the incoming string field" (containing "Month, yyyy"). A circled "2" highlights the dropdown for the incoming format.
- Middle Panel (New Workflow2\*):** A workflow diagram showing a sequence of steps: a record icon, a "Convert Date To" step (labeled "Convert Date To: Month dd, yyyy"), another "Convert Date To" step (labeled "Convert DateTime\_String From: Month dd, yyyy"), and a file icon.
- Right Panel (Results - DateTime (21)-Output):** A table showing the results of the workflow. It has columns: Record #, Date, DateTime\_String, and DateTime\_Out. Two records are displayed:
 

Record #	Date	DateTime_String	DateTime_Out
1	1992-08-28	August 28, 1992	1992-08-28
2	1991-10-12	October 12, 1991	1991-10-12

### Rows Vs Columns

## ROWS VS COLUMNS

An important concept to keep in mind when using Designer is that rows are not treated the same as columns. Unlike some spreadsheet programs where you can specify an array, values are tied to the headers above them in Designer. This is why you will see some tools which require data to be oriented in a particular way in order to function.



## Functions

# FUNCTIONS

Altering data is an integral part of Designer and one of the most powerful ways to alter your data is by applying [functions](#). Designer includes a Function Library which is categorized to help you find the one you need. Some functions will require that the data be in a specific datatype, but others are agnostic. Regardless of which function you need, you can use it in any tool that has an [expression editor](#). The Expression Editor is where you will construct your functions by selecting the function you want to use and properly formatting it into a statement. All expression editors have a tab with the full Function Library, as well as a tab containing the “columns and constants”. You can also choose which column to overwrite or create a new column with a name and datatype of your choosing.

	String	Numeric	Date/Time	Boolean	Spatial
Conditional	x	x	x	x	x
Conversion	x	x			
DateTime	x		x		
File	x				
Finance		x			
Math	x				
Math: Bitwise	x				
Min/Max	x				
Operators	x	x	x	x	x
Spatial	x				x
Specialized	x	x	x	x	x
String	x				
Test	x	x	x	x	x

## Types of Functions

- **Conditional** – Conditional functions allow you to perform an action or calculation using an IF statement.
- **Conversion** – Conversion functions convert numbers to strings or strings to numbers
- **DateTime** – DateTime functions allow you to perform an action or calculation on a date and time value.
- **File** – A file function builds file paths, checks to see if a file exists, or extracts a part of a file path.
- **Finance** – A finance function applies financial algorithms or mathematical calculations.
- **Math** – A math function performs mathematical calculations. Learn more in the [Math Functions](#) article.
- **Math: Bitwise** – A bitwise function operates on one or more bit patterns or binary numerals at the level of their individual bits. Use a bitwise function to manipulate values for comparisons and calculations.
- **Min/Max** – A minimum or maximum function finds the smallest and largest value of a set of values.
- **Operators** – An operator is a character that represents an action. Use an arithmetic operator to perform mathematical calculations or a Boolean operator to work with true/false values.
- **Spatial** – A spatial function builds spatial objects, analyzes spatial data, and returns metrics from spatial fields.
- **Specialized** – These functions perform a variety of specialized actions and can be used with all data types.
- **String** – A string function performs operations on text data. Use a string function to cleanse data, convert data to a different format or case, compute metrics about the data, or perform other manipulations.

- **Test** – A test function performs data comparisons. Use a test function to identify the data type of a value, or determine if a value exists.

## Expression Editor

You can build functions with the expression editor.

### Build an Expression

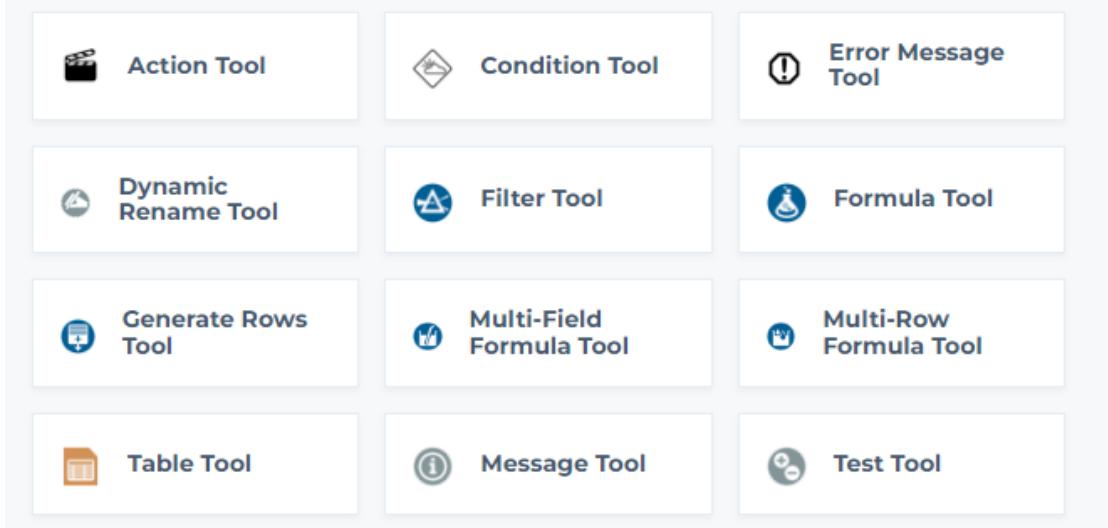
To build your expression, first, specify an Output Column. The Output Column will contain the results of your expression. You can overwrite an existing column or you can create a new column. Select the **Output Column** dropdown and choose an existing column or select **Add Column** and provide a name for your new column.

Once you specify an Output Column, use any of these methods to build an expression:

- Enter the function directly in the expression editor.
- Select the **Columns and Constants** button to browse through variables that can be used in the expression. Select the variable to add it to the expression editor.
- Enter a word or phrase to view a matching list of functions.
- Enter a [ (left bracket) to view a list of variables that you can use in the expression. Variables include...
  - **Columns:** Data from an incoming connection, or from a column created in a previous expression.
  - **Connections from Questions:** Values from an Interface tool connected to the Question anchor of a tool with an expression editor when the tool is used in an app or macro. Use the Question anchor when you want question values to be used as variables in the expression.
  - **Constants:** Global variables for a workflow that make it possible to change a value in a single location and have that change apply to the rest of the workflow.

## Tools with Expression Editor

These tools include an expression editor. Tools that use an expression editor have an additional Input anchor and Question anchor when the tool is used in an app or macro workflow. Use the Interface tools to connect to a Question anchor when you want question values to be used as variables in the expression. Go to [Interface Tools](#) for more information.



## Formula



### FORMULA

Create or update columns using one or more expressions to perform a broad variety of calculations and/or operations.

While there are many tools which support the use of functions, the most common is the **Formula** tool.

Using a Formula tool, you can: utilize values from other columns to perform calculations, categorize, convert datatypes, format values, and much more.

The only limitation is that values in the statement are limited to the current row being processed. One advantage of the Formula tool's configuration window is the unique ability to apply multiple expressions with a single tool.

Most commonly, Expressions are created in the Formula Tool



The formula Tool:

- Can create new columns of data
- Or modify existing columns of data

Create New Column :

This screenshot shows the 'Formula (14) - Configuration' screen. It has a header with tabs for 'Output Column' and 'Data Preview'. Under 'Output Column', there is a dropdown labeled 'Name' with the placeholder 'Enter Expression Here' and a 'Ctrl + Space for suggestions' keytip. Below the dropdown are icons for file operations. At the bottom, it shows 'Data type: V\_WString' and 'Size: 64'. A yellow circle highlights a small icon in the bottom-left corner of the main area.

By Default, New columns of data are given the Data type: V\_WString

Modify Existing Column

This screenshot shows the same 'Formula (14) - Configuration' screen, but the 'Output Column' dropdown is now highlighted in red and contains the text 'Select Column'. The rest of the interface is identical to the previous screenshot, including the 'Data Preview' tab, the 'Enter Expression Here' field, and the 'V\_WString' data type selection.

When a formula is inserted incorrectly there will be no color (i.e. below example)

The screenshot shows the SSIS Expression Editor interface. It displays two output columns. The first column, 'Name', has a data type of V\_String and a size of 64. Its expression is 'Uppercase([Name]) + „COUNTY' and its preview value is 'AUTauga COUNTY'. The second column, 'Density', has a data type of Double and a size of 8. Its expression is '[POP2010]/[AreaSqMi]' and its preview value is '86.1830885640015'.

Expressions are executed in which the order they are displayed

**Questions:**

1. True or False? The Formula tool is the only tool in designer in which expressions can be written
  - a. True
  - b. False

Answer: b

2. From the options listed below, how can functions be inserted into the Expression Editor?
  - a. Typing the name of the function directly into the Expression Editor and selecting the function from the auto-completed options
  - b. Clicking on the Function Library Icon to navigate and select a function
  - c. Right clicking in the Expression Editor and selecting a function

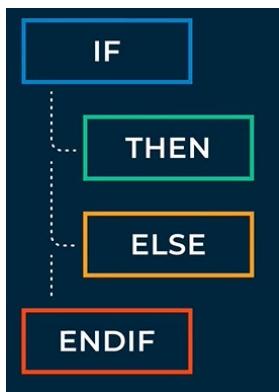
Answer: a, b

3. Select the statement that is false.
  - a. The Data Preview provides a view of the output in the first record of the incoming dataset
  - b. Only one expression can be written in a single Formula tool
  - c. Syntax highlighting distinguishes functions, columns, operators, and text

Answer: b

## Function Types

### Conditional Statements



The screenshot shows the Power BI formula editor interface. The 'Output Column' dropdown is set to 'Weekend'. The formula bar at the top contains the expression: `IF c THEN t ELSE f ENDIF`. Below the formula bar, the 'Conditional' category is selected in the navigation tree. Other categories listed include Conversion, DateTime, File, Finance, Math, Math: Bitwise, Min/Max, and Operators. The formula bar also displays other conditional functions like `IIF(bool, x, y)` and `Switch(Value,Default,Case1,Result1,...,CaseN,ResultN)`.

The screenshot shows the 'Formula (18) - Configuration' dialog. The 'Output Column' dropdown is set to 'Weekend' with a value of 1. The formula bar contains the expression: `IF [Day] = "Sat" OR [Day] = "Sun" THEN 1 ELSE 0 ENDIF`. The 'Data type:' dropdown is set to 'V\_WString' and the 'Size:' dropdown is set to 1073741823. The navigation tree on the left is collapsed.

**IF THEN STATEMENTS****IF** [Day] = "Sat" or [Day] = "Sun" **THEN 1 ELSE 0 ENDIF****INLINE IF STATEMENTS****IIF (** [Day] = "Sat" or [Day] = "Sun", **1 , 0 )**

Else IF for additional clauses

**IF** [X] >= 30  
**THEN** "Long"  
**ELSEIF** [X] < 30 AND [X] >= 5  
**THEN** "Medium"  
**ENDIF**

There is no limit to the number of Else if clauses to a multi-conditional statement

**MULTI-CONDITIONAL STATEMENT****IF** [Category] = "Long" **THEN 1**  
**ELSEIF** [Category] = "Medium" **THEN 2**  
**ELSEIF** [Category] = "Short" **THEN 3**  
**ELSE** Null()  
**ENDIF****SWITCH FUNCTION****Switch([Category], Null(), "Long", 1, "Medium", 2, "Short", 3)**

## Using Null & Empty Expressions

The screenshot shows the SSIS Expression Builder interface with two columns defined:

- Phone Num:** Data type V\_WString, Size 1073741823. Expression: `IF Length([Phone Num]) = 10 THEN [Phone Num] ELSE Null() ENDIF`
- Residence Name:** Data type V\_WString, Size 1073741823. Expression: `IF IsNull([Last Name]) THEN "Current Resident" ELSE [Last Name] + " Residence" ENDIF`

## Empty Example

The screenshot shows the SSIS Expression Builder interface with one column defined:

- Subscription Package:** Data type V\_String, Size 13. Expression: `IF IsEmpty([Subscription Package]) THEN "Standard" ELSE [Subscription Package] ENDIF`

## Questions

1. True or False? The functions Null() and IsNull() are found in the same category in the Function Library.
  - a. True
  - b. False

Answer: b

2. Which expression would convert Null values in the column [Status] to the values “Not available”?
  - a. IF IsNull([Status]) THEN “Not Available” ELSE [Status] END IF

- b. IF Null() THEN "Not Available" ELSE [Status] ENDIF
- c. IF [Status] = "Null" THEN "Not Available" ELSE [Status] ENDIF

Answer: a

3. How many cells in Column 1 would be evaluated as TRUE by the function isEmpty()?

- a. 0
- b. 2
- c. 1

Column 1
[Null]
Standard
Premium

Answer: b

## String Functions

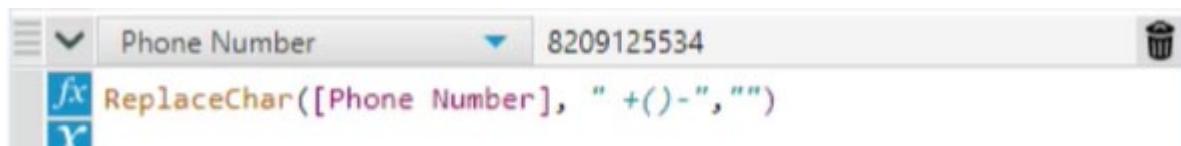
Removing whitespace



Deleting Whitespace:



Remove special characters:



**Case**

String
UpperCase()
LowerCase()
TitleCase()

**Questions**

1. Which function was used to produce the values in the column [Email]?
  - a. UpperCase()
  - b. TitleCase()
  - c. LowerCase()

Email
A_RODRIG@CS.COM
DGOLD@CS.CO.UK
RAVENJ@CS.COM
CA123@CS.CO.BR

**Answer: a**

2. Which of the following is the correct expression to remove leading zeros from the column [Name]?
  - a. Trim[Name], "")
  - b. Trim("0")
  - c. TrimLeft([Name]. "0")

Name
000Anna Rodriguez
000David Goldman
000Janet Raven

**Answer: c**

3. Which of the following functions replace every occurrence of a target string or character?  
Select all that apply.
  - a. ReplaceChar()
  - b. Replace()
  - c. ReplaceFirst()

**Answer: a, b**

## Numeric Functions

Monthly total car sales example:

Output Column	Data Preview
Monthly Total Car Sales	81033
[fx] [New]+[Used]	X
Data type: Int64	Size: 8

Output Column	Data Preview
Monthly Percent	7.65724264424481
[fx] [Monthly Total Car Sales]/[Annual Total Cars Sold]*100	X
Data type: V_WString	Size: 1073741823

Specialized rounding

**CEIL(x)** \_\_\_\_\_

Rounds UP to the nearest integer

**FLOOR(x)** \_\_\_\_\_

Rounds DOWN to the nearest integer

**SmartRound(x)** \_\_\_\_\_

Rounds to the nearest multiple,  
dynamically decided by Designer

**Round(x, mult)** \_\_\_\_\_

Rounds to the nearest multiple specified

Output Column	Data Preview
Monthly Percent	7.65724264424481
Rounded Monthly Percent	8
[fx] Round([Monthly Percent], 1)	X
Data type: Int16	Size: 2

## Questions

1. True or False? Math functions can only be used on Numeric data types

- a. True
- b. False

Answer: True

2. Which function would produce the result below?

- a. Ceiling()
- b. Round()
- c. Floor()

Percent	Result
45.224	45
78.611	79
20.906	21
94.854	95

Answer: b

3. Which function category would contain the function category "Addition"

- a. Math
- b. Operators
- c. Conversion
- d. Test

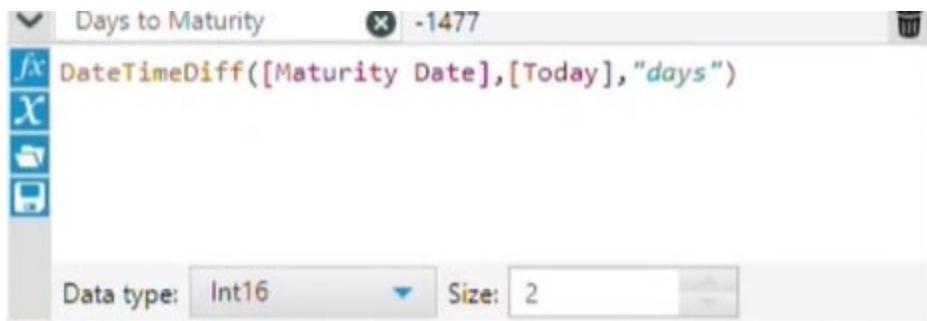
Answer: b

## DateTime Functions

The screenshot shows the Power BI Data Editor interface. A column named "Maturity Date" is selected. The formula bar displays the formula: `DateTimeAdd([Purchase Date], [Duration], "months")`. The "Data type" dropdown is set to "DateTime" and the "Size" is 19. The "Data Preview" shows the value `2018-04-10 17:56:18`.

Returns current date

The screenshot shows the Power BI Data Editor interface. A column named "Today" is selected. The formula bar displays the formula: `DateTimeNow()`. The "Data type" dropdown is set to "DateTime" and the "Size" is 19. The "Data Preview" shows the value `2022-04-27 10:32:08`.

**Format DateTime Values:**

- %A** Full Weekday Name
- %B** Full month Name
- %d** Day of the Month (01-31)
- %j** Day of the Year (01-365)
- %W** Week Number (00-53)
- %X** 12 Hour Clock Time
- %Z** Full Timezone Name
- (many more)

<b>DATE</b>	2019-02-23
<b>/</b> slash <b>-</b> dash <b>,</b> comma <b>.</b> period space <b>:</b> colon <b>%n</b> new line <b>%t</b> tab	<b>%A</b> Full Weekday Name <b>%B</b> Full month Name <b>%d</b> Day of the Month (01-31) <b>%j</b> Day of the Year (01-365) <b>%W</b> Week Number (00-53) <b>%X</b> 12 Hour Clock Time <b>%Z</b> Full Timezone Name (many more)

### Questions

1. True or False? The `DateTimeDiff()` function will never return a negative value.

- a. True
- b. False

Answer: False

2. Which expression would output the value in the column [Output] in the table below?

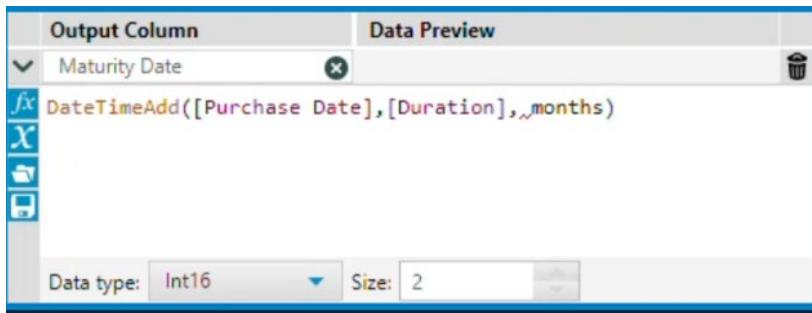
- a. `DateTimeYear([Start Date])`
- b. `DateTimeDiff([Start Date], [Output], "years")`
- c. `DateTimeAdd([Start Date], -1, "years")`

START DATE	OUTPUT
2017-01-01	2016-01-01

Answer: c

3. What about the configuration below is incorrect? Select all that apply.

- a. Data type
- b. Nothing is wrong
- c. Syntax error



**Answer:** a, c

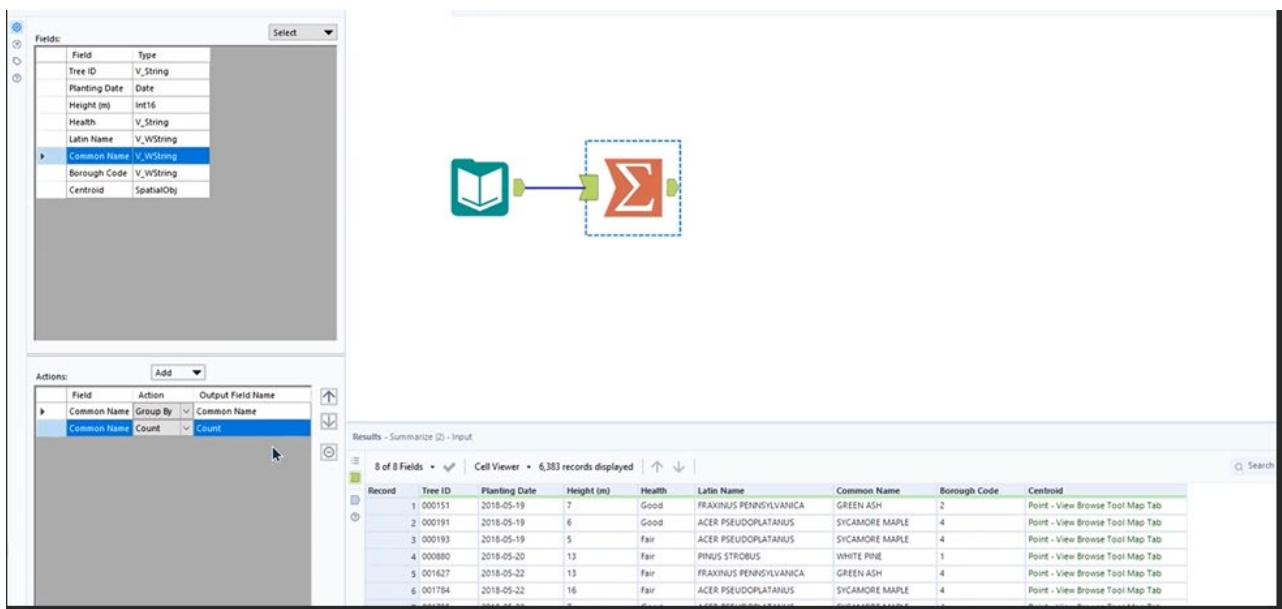
## Summarize



Functions are extremely versatile and powerful in Designer, but one limitation mentioned for the Formula tool was that values used in a statement must be on the row being processed. If your data is vertically oriented, you can also use the **Summarize** tool.

Using the **Summarize** tool, you can apply [Summary Processes](#) to data that is aligned vertically. Aggregations like Min, Max, Sum, Average, Count, Grouping, and many more are quick and simple with the **Summarize** tool.

### Group by and count on column name



Record	Tree ID	Planting Date	Height (m)	Health	Latin Name	Common Name	Borough Code	Centroid
1	000151	2018-05-19	7	Good	FRAXINUS PENNSYLVANICA	GREEN ASH	2	Point - View Browse Tool Map Tab
2	000191	2018-05-19	6	Good	ACER PSEUDOPLATANUS	SYCAMORE MAPLE	4	Point - View Browse Tool Map Tab
3	000193	2018-05-19	5	Fair	ACER PSEUDOPLATANUS	SYCAMORE MAPLE	4	Point - View Browse Tool Map Tab
4	000880	2018-05-20	13	Fair	PINUS STROBIS	WHITE PINE	1	Point - View Browse Tool Map Tab
5	001627	2018-05-22	13	Fair	FRAXINUS PENNSYLVANICA	GREEN ASH	4	Point - View Browse Tool Map Tab
6	001784	2018-05-22	16	Fair	ACER PSEUDOPLATANUS	SYCAMORE MAPLE	4	Point - View Browse Tool Map Tab

When using the summarize tool, only the columns chosen in the tool are shown in the output

The screenshot shows the 'Summarize' tool configuration and its resulting output.

**Actions:** Add ▾

Field	Action	Output Field Name
Common Name	Group By	Common Name
Common Name	Count	Count

**Results - Summarize (2) - Output**

2 of 2 Fields Cell Viewer 20 records displayed

Record	Common Name	Count
1	magnolia	8
2	CUCUMBER MAGNOLIA	43
3	cucumber magnolia	1
4	SYCAMORE MAPLE	833
5	black maple	1
6	AMERICAN BEECH	94
7	CATALPA	148
8	OSAGE-ORANGE	6
9	GREEN ASH	4746
10	SOUTHERN MAGNOLIA	40
11	green ash	105
12	american beech	1
13	amur cork tree	2
14	catalpa	2
15	MAGNOLIA	193

Action Properties  
This action has no properties

### Questions:

1. Given the output below, which is the correct configuration of the summarize tool?

The diagram illustrates the correct configuration of the Summarize tool by comparing three options against a reference table.

**Reference Table:**

FIELD 1	FIELD 2
▲	a
●	b
■	c
★	d
▲	e
●	f
■	g
★	h
▲	i
●	j
■	k
★	l
▲	m
●	n

**Options:**

- **Option A:** Field 1: GroupBy, Action: Field, Output Field Name: Field  
Field 2: Concat, Action: Concat, Output Field Name: Field
- **Option B:** Field 1: Count, Action: Field, Output Field Name: Field  
Field 2: Concat, Action: Concat, Output Field Name: Field
- **Option C:** Field 1: GroupBy, Action: Field, Output Field Name: Field  
Field 2: Mode, Action: Concat, Output Field Name: Field

**SUMMARIZE OUTPUT:**

FIELD	CONCAT
▲	a, e, i, m
●	b, f, j, n
■	c, g, k
★	d, h, l

Answer: a

2. True or False? The Summarize tool can apply summary processes to Spatial, String, and Numeric datatypes.

- a. True
- b. False

Answer: True

3. Which of the following function categories have Action Properties? Select all that apply.
- a. None

- b. Numeric
- c. String
- d. Reporting
- e. Finance

**Answer: c, e**

### Practice Exercise 3 Answer:

## Practice Exercise 3

Combine the customer information from the last exercise with "Order Details" to find the Total and Total Saved for each order.

Combine the customer information in "Start 3.1" with the "Order Details". Ensure both the "Order Date" and "Ship Date" columns are formatted as "date" datatype.

Find the number of days to ship after the order was placed for each line item. Also calculate the subtotal for each line item (discount and shipping are flat costs, not per quantity). Finally, find the Total cost for each order, the "Total Saved" for each order, and the minimum days to ship any line item within each order.

**Initial State**



Start 3.1.yxdb



Order Details

Order Details.yxdb

Convert Ship Date From: MM/dd/yyyy

Days to Ship = DateTimeDiff ([Ship Date], [Order Date], "days")

Total = ([Unit\_Price] \* [Quantity\_ordered\_new]) + ([Shipping\_Cost] - [Discount])...

Final State



Final State - Exercise 3.yxdb

Results - Summarize (27) - Output

Record	Order_ID	Total	Total Saved	Days to Ship
1	87222	53.58	3.28	5
2	87399	163.38	12.47	1
3	88060	1383.84	119.06	0
4	90353	74.65	2.15	1
5	90354	5896	441.39	1
6	90355	105.1	8.36	2
7	91515	491.61	42.23	0
8	91516	14.87	0.38	1
9	91519	1429.36	121.18	5
10	91560	235.46	7.13	2

## Transpose & Cross Tab



## TRANSPOSE

Pivot the orientation of the data in a table by moving horizontal data on the vertical axis.



## CROSS TAB

Pivot the orientation of data in a table by moving vertical data onto a horizontal axis and summarizing selected data.

Changing the layout of your data, or *pivoting* the data, is another key skill. Turning columns into rows and rows to columns can be very useful for sparking new ideas, or a necessary step in order to use a particular tool. You already know that Designer makes a distinction between rows and columns. For that reason, there is a separate tool for each action. The **Transpose** tool turns rows into columns, while the **Cross Tab** tool turns columns into rows.

### Pivoting data

- Unlike excel, you cannot specify an array in designer when applying a function
- It is applied to the entire dataset

**Transpose** – Pivots horizontal data to vertical orientation

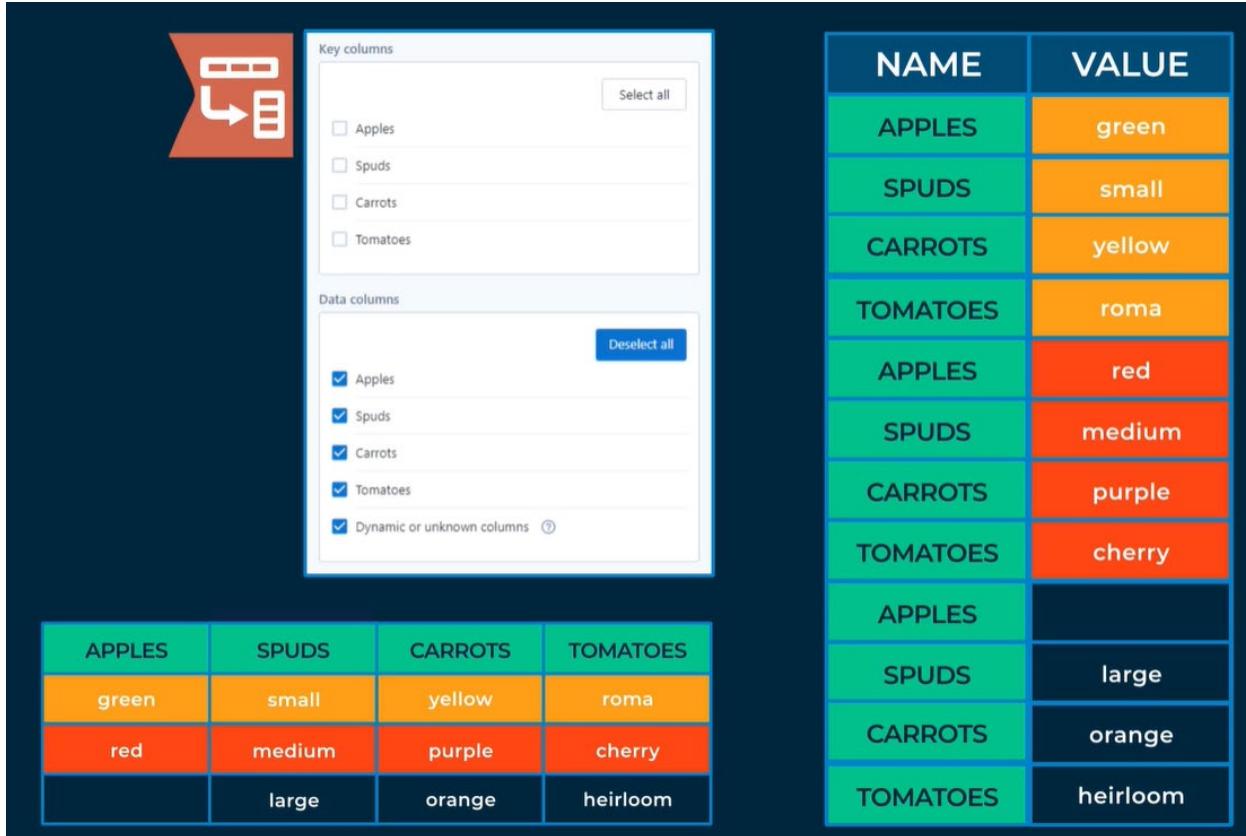
**Cross Tab** – Pivots vertical data to horizontal orientation

### Transpose

APPLES	SPUDS	CARROTS	TOMATOES
green	small	yellow	roma
red	medium	purple	cherry
	large	orange	heirloom

OUTPUT RECORDS = INCOMING COLUMNS X INCOMING ROWS

NAME	VALUE
APPLES	green
SPUDS	small
CARROTS	yellow
TOMATOES	roma
APPLES	red
SPUDS	medium
CARROTS	purple
TOMATOES	cherry
APPLES	
SPUDS	large
CARROTS	orange
TOMATOES	heirloom



The screenshot shows the Power BI Data Flow interface. On the left, there's a large orange arrow icon pointing right. Below it, two sections are visible: "Key columns" and "Data columns". In the "Key columns" section, four items are listed: Apples, Spuds, Carrots, and Tomatoes, each with a checkbox. A "Select all" button is at the top right. In the "Data columns" section, the same four items are listed with checked checkboxes, and a "Deselect all" button is at the top right. Below these sections is a preview of the data in a grid format. The grid has four columns labeled APPLES, SPUDS, CARROTS, and TOMATOES. The rows show the following data:

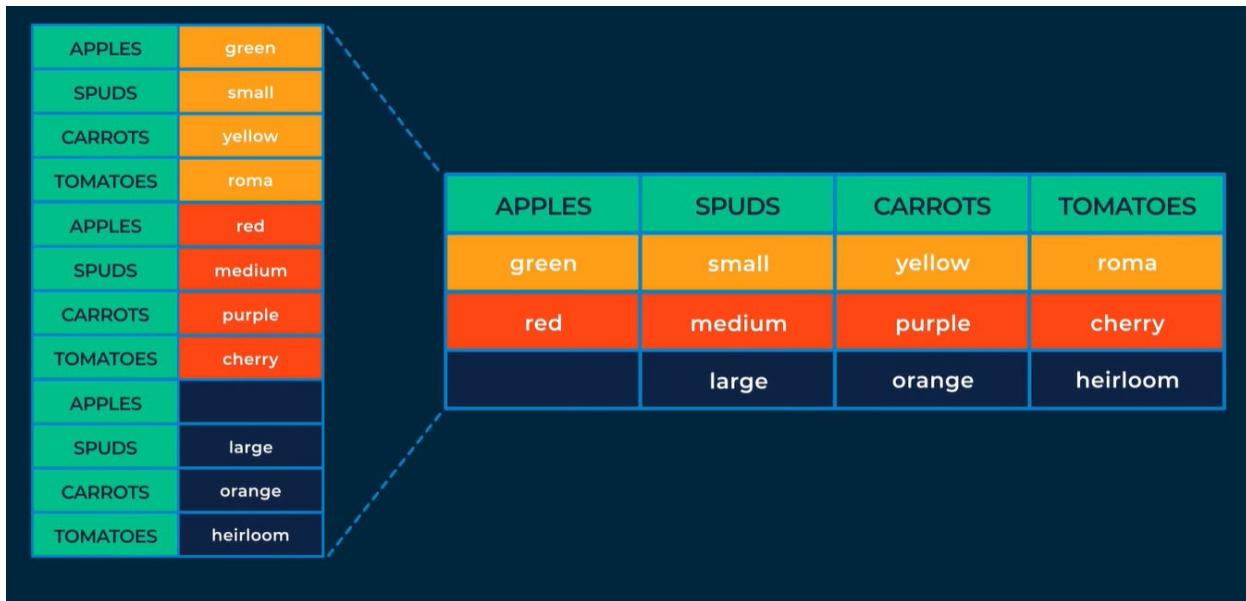
APPLES	SPUDS	CARROTS	TOMATOES
green	small	yellow	roma
red	medium	purple	cherry
	large	orange	heirloom

On the right side of the interface, there's a table titled "NAME" and "VALUE" with the following data:

NAME	VALUE
APPLES	green
SPUDS	small
CARROTS	yellow
TOMATOES	roma
APPLES	red
SPUDS	medium
CARROTS	purple
TOMATOES	cherry
APPLES	
SPUDS	large
CARROTS	orange
TOMATOES	heirloom

## Cross Tab

- Requires a pattern to pivot
- At least 2 columns must be present in the incoming dataset



FIELD 1	FIELD 2
▲	a
●	b
■	c
★	d
▲	e
●	f
■	g
★	h
▲	i
●	j
■	k
★	l
▲	m
●	n

 Cross Tab

Group data by these values

 Field1  
 Field2  

Select all

Change column headers

Field1

Values for new columns

Field2

Field1

Field2

Sum  
 Average  
 Count (without Nulls)  
 Count (with Nulls)  
 Percent Row  
 Percent Column

### Aggregating Values

**STRING**

Values for new columns

Value

Method for aggregating values

Select all

Concatenate  
 First  
 Last

Separator : ,

Field Size : 2048

**NUMERIC**

Values for new columns

Field2

Method for aggregating values

Select all

Sum  
 Average  
 Count (without Nulls)  
 Count (with Nulls)  
 Percent Row  
 Percent Column  
 Total Column  
 Total Row  
 First  
 Last

 Cross Tab

**String**

**Aggregation options for String Data**



FIELD 1	FIELD 2
▲	a
●	b
■	c
★	d
▲	e
●	f
■	g
★	h
▲	i
●	j
■	k
★	l
▲	m
●	n

**FIRST**

▲	●	■	★
a	b	c	d

**LAST**

▲	●	■	★
m	n	k	l

**CONCATENATE**

▲	●	■	★
a, e, i, m	f, j, b, n	k, c, g	d, h, l

**Numeric**

**Aggregation options for Numeric Data**



FIELD 1	FIELD 2
▲	1
●	11
■	21
★	7
▲	9
●	19
■	5
★	15
▲	17
●	3
■	13
★	23
▲	2
●	31

**SUM**

▲	●	■	★
29	54	39	45

**AVG**

▲	●	■	★
7.25	13.5	13	15

**FIRST**

▲	●	■	★
1	11	21	7

**LAST**

▲	●	■	★
2	31	13	23

### Grouping in Cross Tab

FIELD 1	FIELD 2	FIELD 3
▲	a	
●	b	
■	c	
★	d	
▲	e	
●	f	
■	g	
★	h	
▲	i	
●	j	
■	k	
★	l	

	▲	●	■	★
a		j	g	d
e		b	k	h
i		f	c	l

Cross Tab

If the incoming data set has multiple data types, it will automatically transform to the most robust data type

**Keep in mind when using  
TRANSPOSE & CROSS TAB**

NUMERIC  
STRING  
DATETIME

V\_STRING

Questions:

- Given the transpose configuration below, how many columns will appear in the output?
  - 1
  - 2
  - 3
  - 4

The screenshot shows the 'Key columns' and 'Data columns' sections of the Cross Tab tool. In the 'Key columns' section, 'Spuds' is selected. In the 'Data columns' section, 'Apples', 'Carrots', 'Tomatoes', and 'Dynamic or unknown columns' are selected.

**Answer: b**

2. Which of the header values below would have characters converted to underscores when output by the Cross Tab tool? Select all that apply.
- A1
  - b&b
  - 1 a
  - a\_a
  - AyX

**Answer: b, c, d**

3. Which method of aggregation was used?
- Concatenate
  - First
  - Last
  - Sum
  - None of These

The screenshot shows a 'CROSS TAB OUTPUT' icon above two tables. The source table on the left has columns 'FIELD 1' and 'FIELD 2'. The result table on the right has columns labeled with icons: triangle, circle, square, and star.

FIELD 1	FIELD 2
▲	a
●	b
■	c
★	d
▲	e
●	f
■	g
★	h
▲	i
●	j
■	k
★	l
▲	m
●	n

	▲	●	■	★
m				
n				
k				
l				

**Answer: c**

### Find Replace

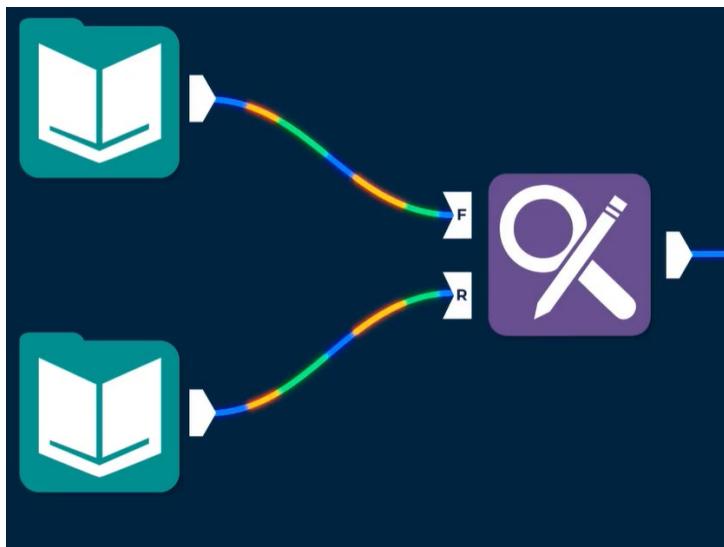


Another common task that hasn't been covered yet is updating values from a lookup table. With the tools covered so far, you would need to manually enter functions for each column containing the values to replace.

If you use the **Find Replace** tool, it will be much easier, not to mention faster. The Find Replace tool has two input anchors, one for a dataset (F for Find) and the other for the lookup table (R for Replace).

You can specify what constitutes a match, the column in which to search, and the column containing the replacement value. There is also an option to append values when a match is found, rather than replace the matched value.

### VLookups with Designer



Main Dataset		Lookup Table	
STORE ID	LOCATION	LOCATION	REGION
A	1	1	Coast
B	2	2	Inland
C	2	3	Mountain
D	1		
E	3		

Determine Parameters for a match between values:

**Find**

Beginning of Field  
 Any Part of Field  
 Entire Field

Find Within Field:  
 Zoning Code

Find Value:  
 Zoning Code

Case Insensitive Find  
 Match Whole Word Only

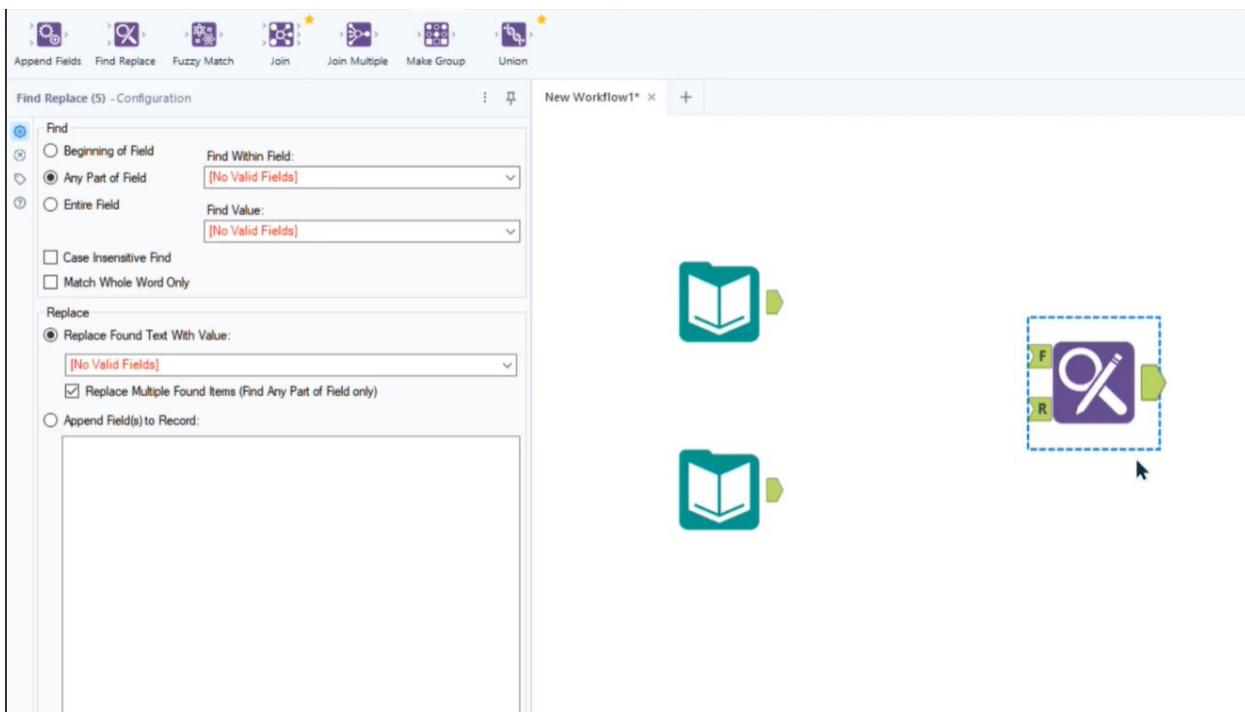
**Replace**

Replace Found Text With Value:  
 Zoning Name

Replace Multiple Found Items (Find Any Part of Field only)

Append Field(s) to Record:  
 Zoning Code  
 Zoning Name  
 Property Tax Code  
 Zone Land Category

## Find Replace Configuration



**F Anchor = Find**

**R Anchor = Replace**

- Note: Second input data above contains a lookup table

3 Find options for matching the values in the dataset

- Beginning of Field
- Any Part of Field
- Entire Field

**Beginning of Field****REPLACE DATASET**

Region	Region Abv.
West	W

**FIND DATASET**

Record #	Site ID	Zoning Code	Region
1	10000	A-G	Southwest
3	10002	U-RH	Front Range
8	10007	A-H	Wern Slope
21	10020	s-MU	W

Searches the column for the lookup string at the very beginning of the cell.

Does not match if another character is before or within.

**Any Part of Field****REPLACE DATASET**

Region	Region Abv.
West	W

**FIND DATASET**

Record #	Site ID	Zoning Code	Region
1	10000	A-G	SouthW
3	10002	U-RH	Front Range
8	10007	A-H	Wern Slope
21	10020	s-MU	W

Searches the column for the lookup string anywhere in the cell.

Does not match if another character is within.

**Entire Field**

**REPLACE DATASET**

Region	Region Abv.
West	W

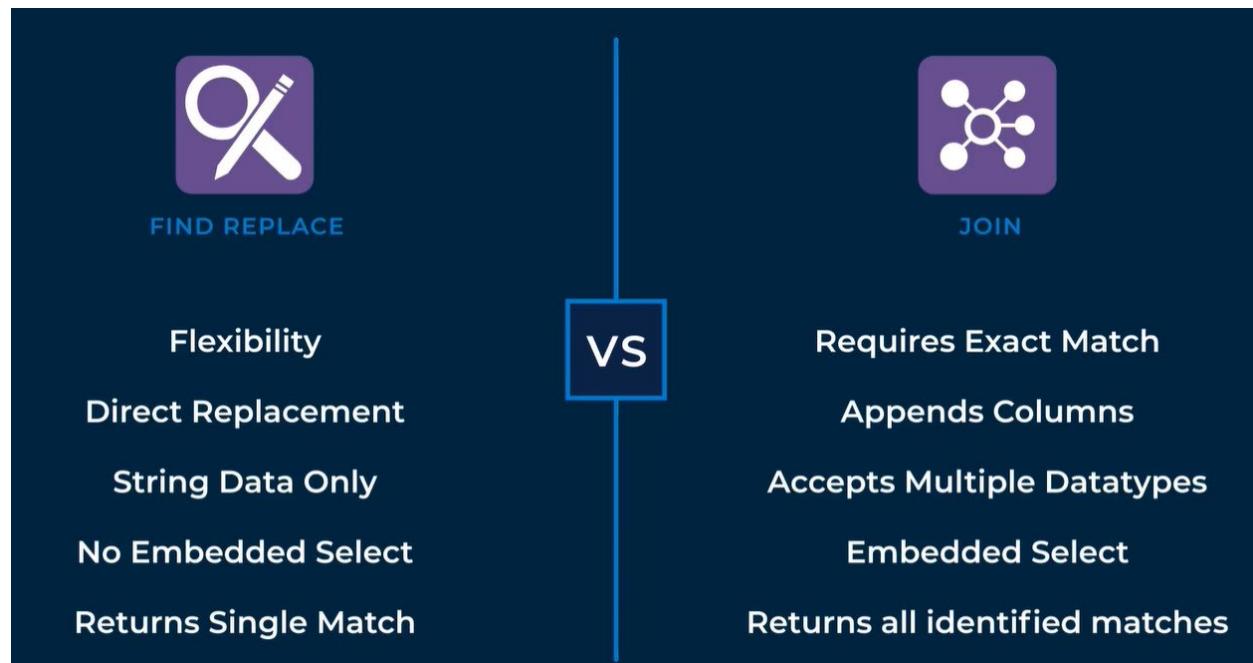
**FIND DATASET**

Record #	Site ID	Zoning Code	Region
1	10000	A-G	Southwest
3	10002	U-RH	Front Range
8	10007	A-H	Western Slope
21	10020	s-MU	W

Searches the column for the lookup string. Will only match if there are no other characters in the cell.

**Replace Options**

- Find value searches the dataset in the replace value anchor
- Find Replace tool is case sensitive by default

**Find Replace vs Join:**

**Questions:**

1. Which of the following data types are acceptable for identifying matching values when using the find replace tool? Select all that apply.
  - a. Numeric
  - b. String
  - c. Spatial
  - d. DateTime

Answer: b

2. True or False? The dataset that is connected to the “F” input of the Find Replace tool is the input in which data values are replaced or appended.
  - a. True
  - b. False

Answer: True

3. Select ALL features which are NOT associated with the Find Replace tool.
  - a. Appending a column
  - b. Partial field matching
  - c. Appending more than one column
  - d. Embedded select window
  - e. Works with numeric values
  - f. Case insensitivity

Answer: d, e

**Append Fields**

## APPEND FIELDS

Append the fields from a source input to every record of a target input. Each record of the target input will be duplicated for every record in the source input.

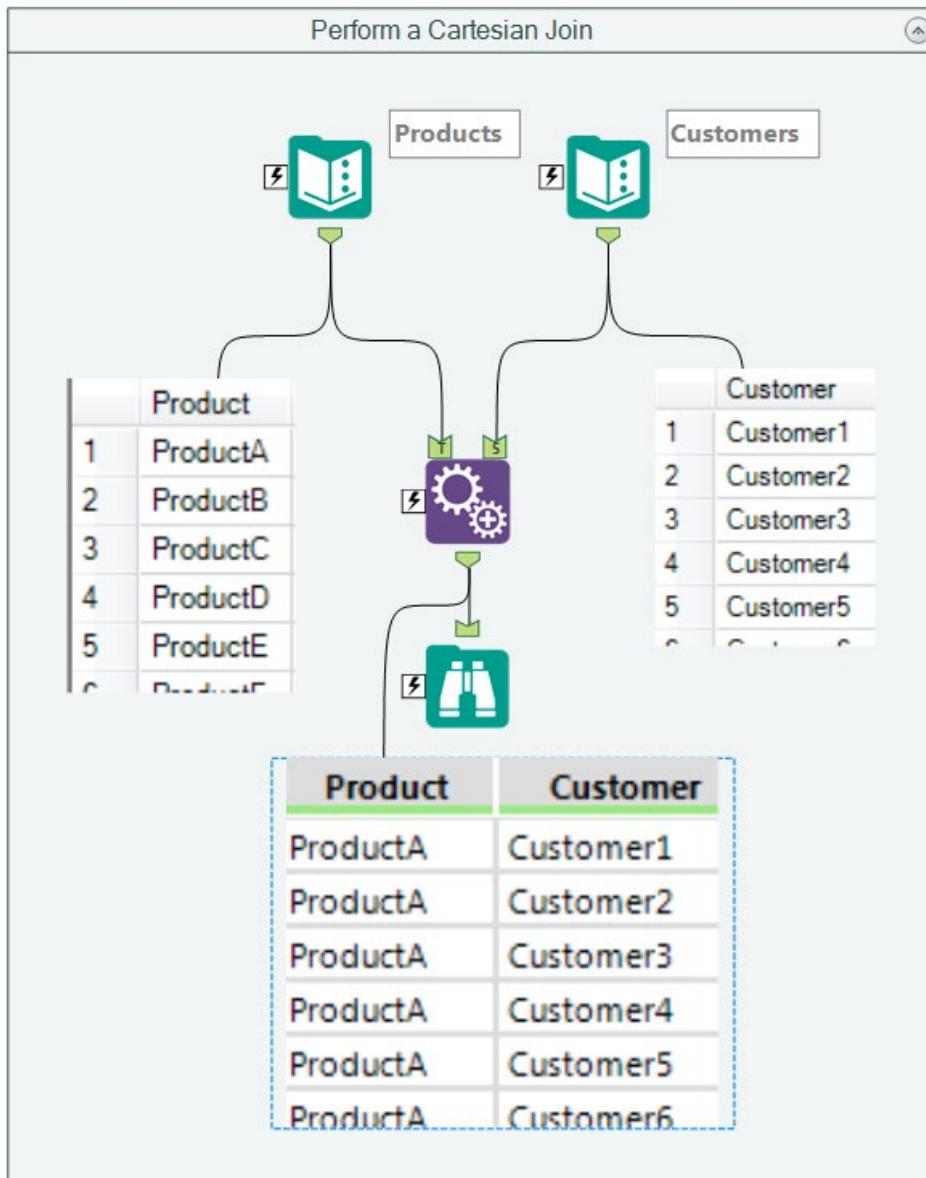
If you want to append values to all rows, the Append Fields tool is a great option.

This tool performs a Cartesian Join, meaning that each row of your dataset will be duplicated enough times to accommodate the new values. If you are appending three rows of data, each existing row is duplicated twice to accommodate the three values.

The Append Fields tool adds every row of the source input to every row of the target input as new columns.

This is known as a Cartesian or cross join. If your target input has A rows with B columns and your source input has C rows with D columns, the Append Fields tool with output A times C rows with B plus D columns.

There are times when you want to join every row from one table to every row of another table. Let's suppose you have a list of possible customers and your product inventory. You could use an Append Fields tool to join this data:



## Output Data



Finally the **Output Data** tool outputs your workflow to a file format of your choice.

Think of this tool as the reverse function of the Input Data tool. Instead of telling the tool where to find the data to read in, you tell it where to write a file. To overwrite an existing file, simply navigate to that file and select it.

In the same way that the Input Data tool can read in multiple files with a single tool, the Output Data tool can write several files with a single tool.

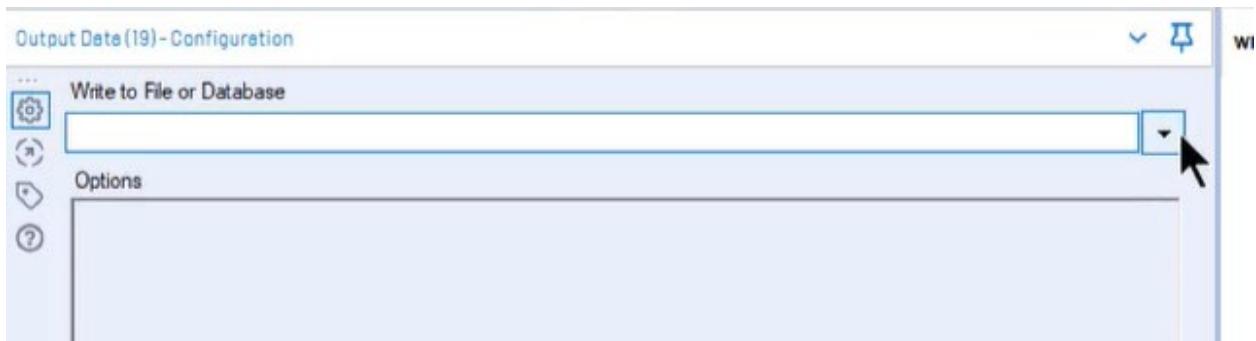
One noteworthy detail when you are just starting with the Output Data tool is that nothing is written until you configure the tool and click Run. Clicking "Save" will only save the workflow. After a successful run, you can click the canvas and the results window will display a hyperlink to the newly created file(s).

## Writing Data

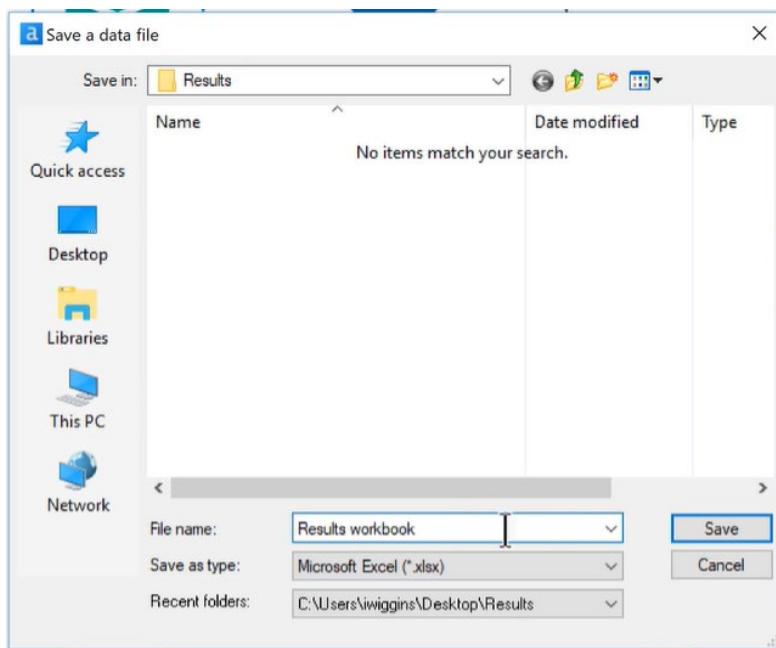


- When using the Output data tool in a workflow, it creates tangible results outside of designer (DB, cloud DB, file)
- Commonly used file types for output
  - o Excel Workbook
  - o CSV
  - o Alteryx Database

## Excel Workbook:

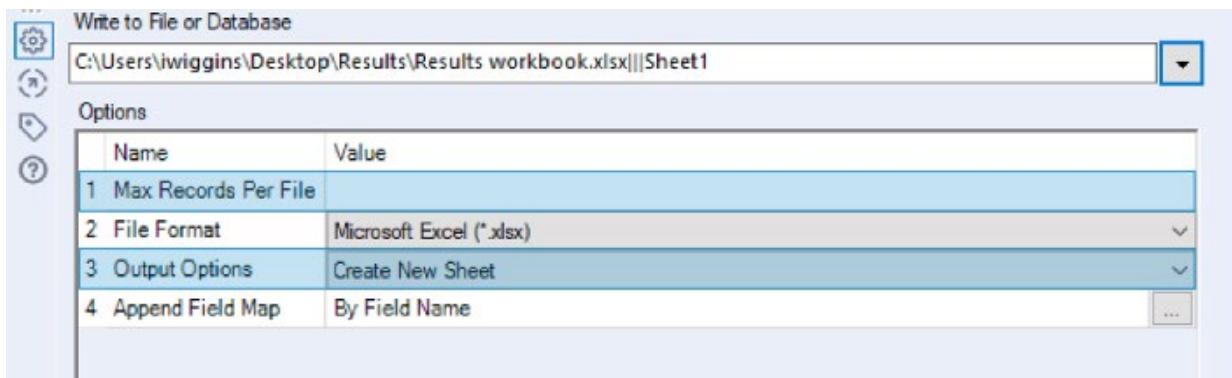


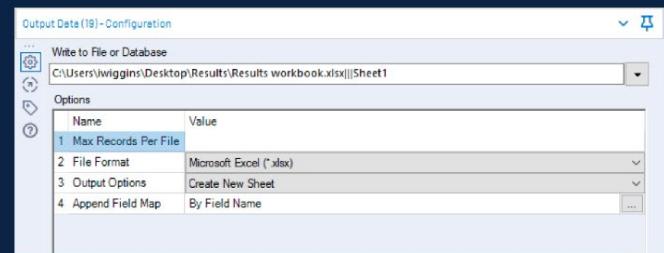
Choose file type and save to workspace



- Excel workbooks are limited in number of rows and types of data
- Special objects are not supported

Options when writing to an excel workbook:

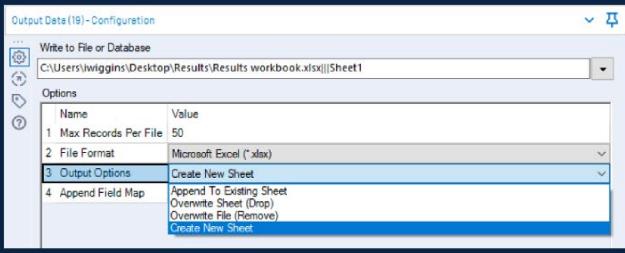




**MAX RECORDS PER FILE**

Limit the number of rows exported in a single file

Creates multiple files when there are more rows than you specify



**OUTPUT OPTIONS**

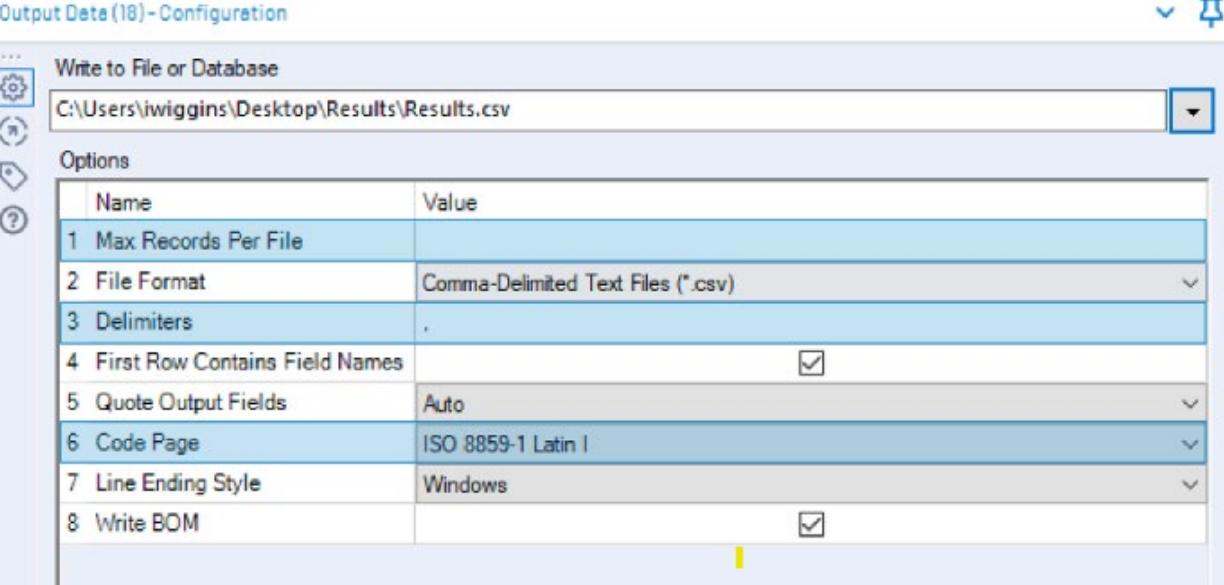
“Create New Sheet” when writing data for the first time

Change your selection to overwrite or modify existing files on subsequent runs

## CSV

- CSV will store all data values as string data types
- Any unsupported data types will be dropped in the output

### CSV Configuration Window

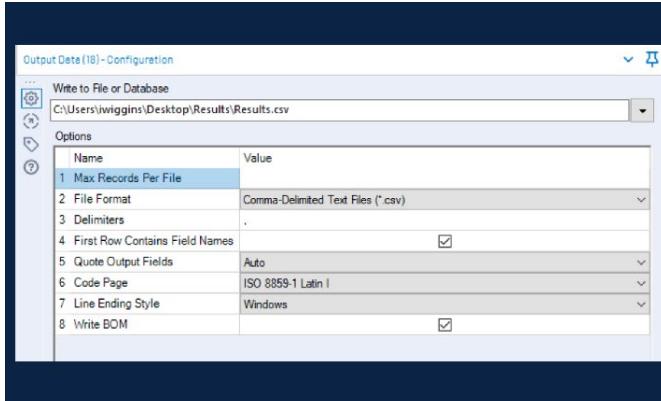


**Output Data (18)- Configuration**

**Write to File or Database**  
C:\Users\iwiggins\Desktop\Results\Results.csv

**Options**

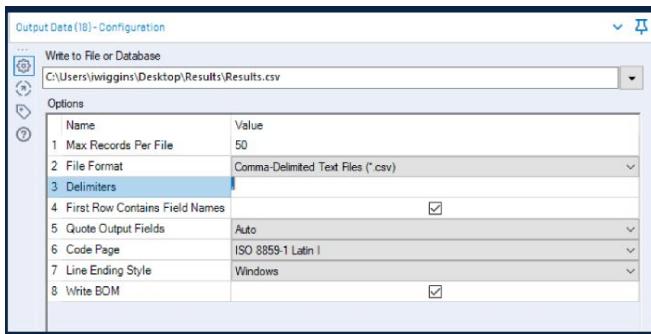
Name	Value
1 Max Records Per File	
2 File Format	Comma-Delimited Text Files (*.csv)
3 Delimiters	,
4 First Row Contains Field Names	<input checked="" type="checkbox"/>
5 Quote Output Fields	Auto
6 Code Page	ISO 8859-1 Latin I
7 Line Ending Style	Windows
8 Write BOM	<input checked="" type="checkbox"/>



## MAX RECORDS PER FILE

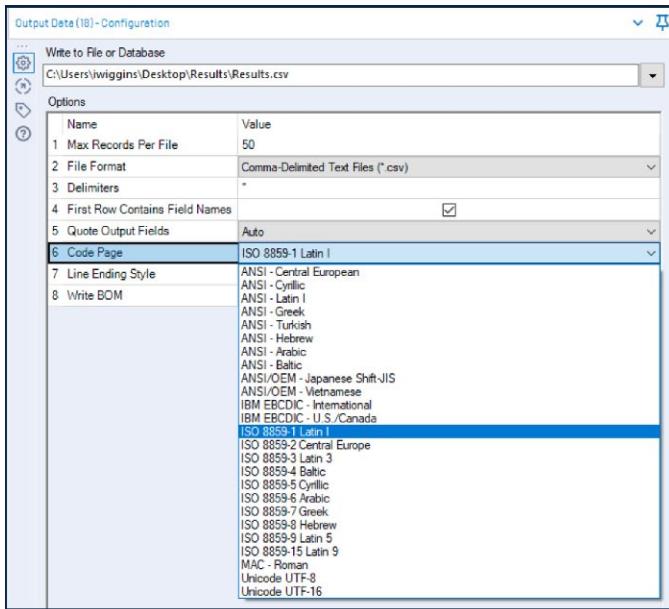
Limit the number of rows exported in a single file

Creates multiple files when there are more rows than you specify



## DELIMITERS

Specify the delimiter used to separate values



## CODE PAGE

Select a code page to convert text upon output

**Alteryx Database**

# ALTERYX DATABASE

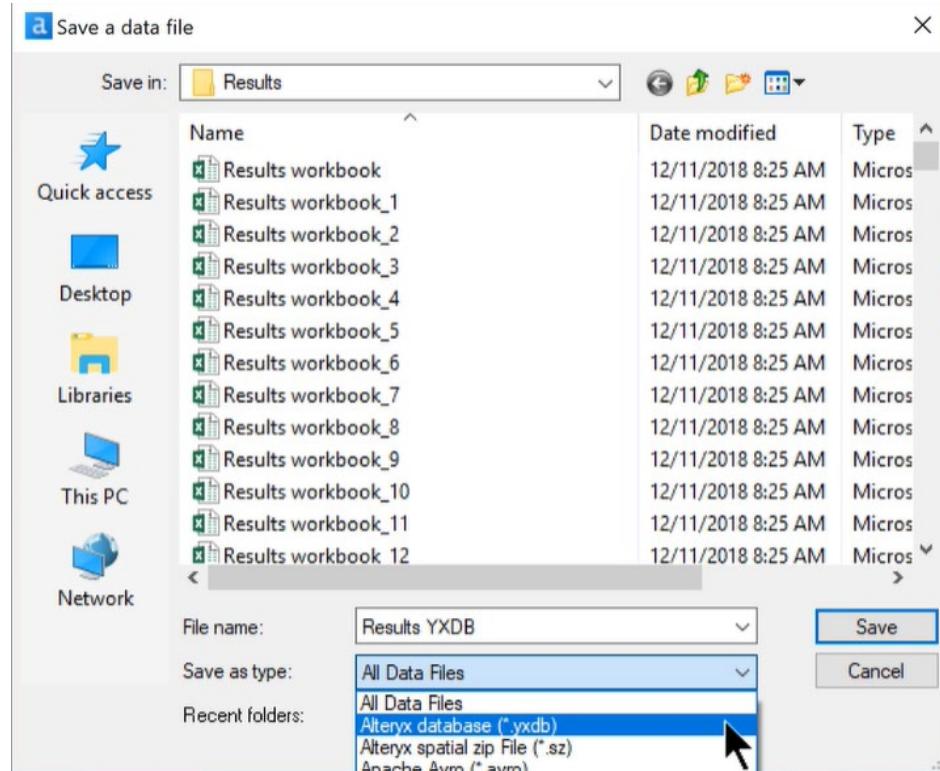
File extension = .YXDB

Most efficient output format

Compressed for maximum speed

Uses the same structure as Designer

No limit on rows



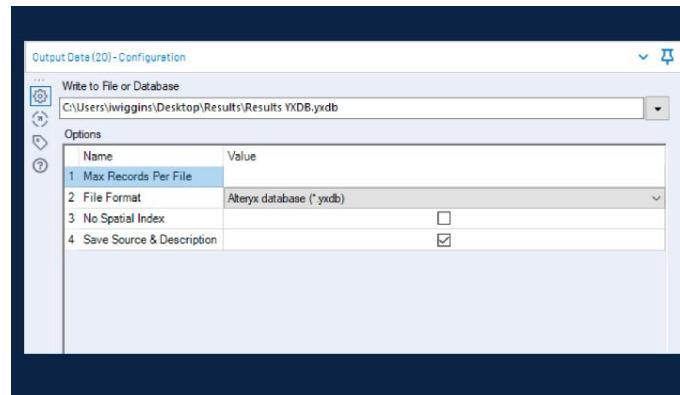
### Alteryx Output Configuration Window:

**Output Data (20)-Configuration**

Write to File or Database  
C:\Users\iwiggins\Desktop\Results\Results YXDB.yxdb

Options

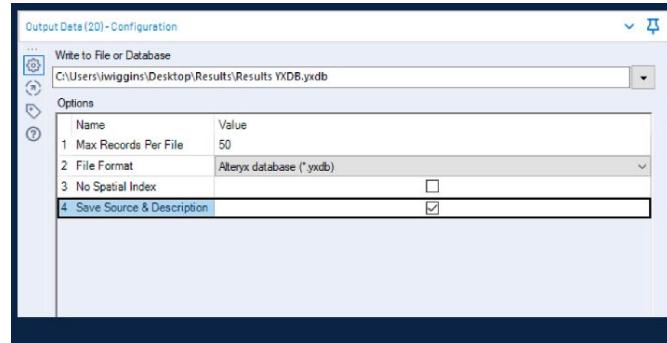
Name	Value
1 Max Records Per File	
2 File Format	Alteryx database (*.yxdb)
3 No Spatial Index	<input type="checkbox"/>
4 Save Source & Description	<input checked="" type="checkbox"/>



## MAX RECORDS PER FILE

Limit the number of rows exported in a single file

Creates multiple files when there are more rows than you specify



## SAVE SOURCE & DESCRIPTION

Default setting includes source and description data

Useful for understanding the origin of data

Running the workflow creates a new data source or modifies existing data

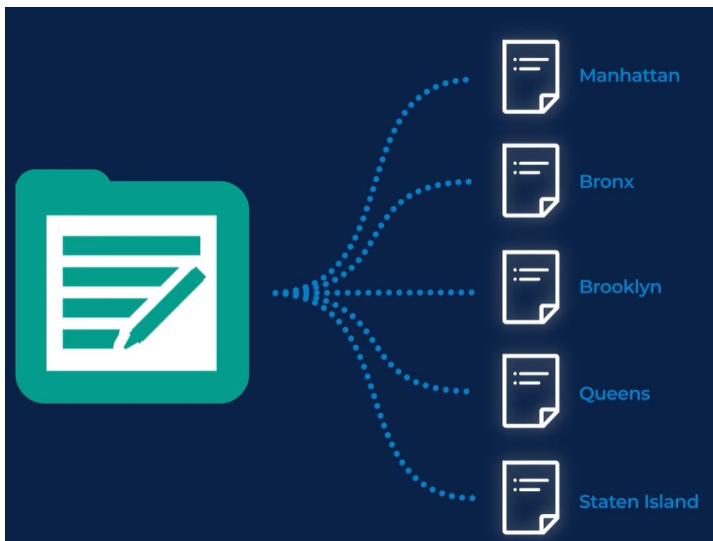
Output Window after running workflow:

Results - Workflow - Messages

0 Errors 0 Conv Errors 3 Warnings 3 Messages 5 Files All

- Union (9) The field "Centroid" is not present in all inputs.
- Union (9) The field "Latitude" is not present in all inputs.
- Union (9) The field "Longitude" is not present in all inputs.
- Input Data (3) 200000 records were read from "C:\Users\cbonthius\Desktop\manually\_entering\_data\workflow\Tree Survey\_2018.yxdb"
- Browse (8) 200000 records
- Filter (10) 86479 records were True and 121521 were False
- Text Input (6) 5 records were output
- Join (11) 86479 records were joined with 0 un-joined left records and 0 un-joined right records
- Output Data (20) 86479 records were written to "C:\Users\cbonthius\Desktop\manually\_entering\_data\workflow\Results.yxdb"
- Designer x64 Finished running writing\_data.ycmd in 2.1 seconds with 3 warnings

You can also output to multiple files



#### Questions:

1. True or False? Data is only created or modified when a workflow with a configured output data tool is run.
  - a. True
  - b. False

Answer: True

2. Based on the table below, how many output files will be produced when the Output Data tool is configured as shown?
  - a. 1
  - b. 2
  - c. 5

Borough Name	Common Name
Manhattan	Oak
Brooklyn	Pine
Queens	Oak
Staten Island	Oak
Bronx	Pine

● One (1)  
● Two (2)  
● Five (5)

**Answer: b**

**3. Select the statements that are true**

- a. The Output Data tool's configuration options change based on the output file type.
- b. The Output Data tool can only write to files.
- c. The Output Data tool can only output one file at a time.
- d. The Output Data tool does not include an output anchor

**Answer: a, d**

## Practice Exercise 4

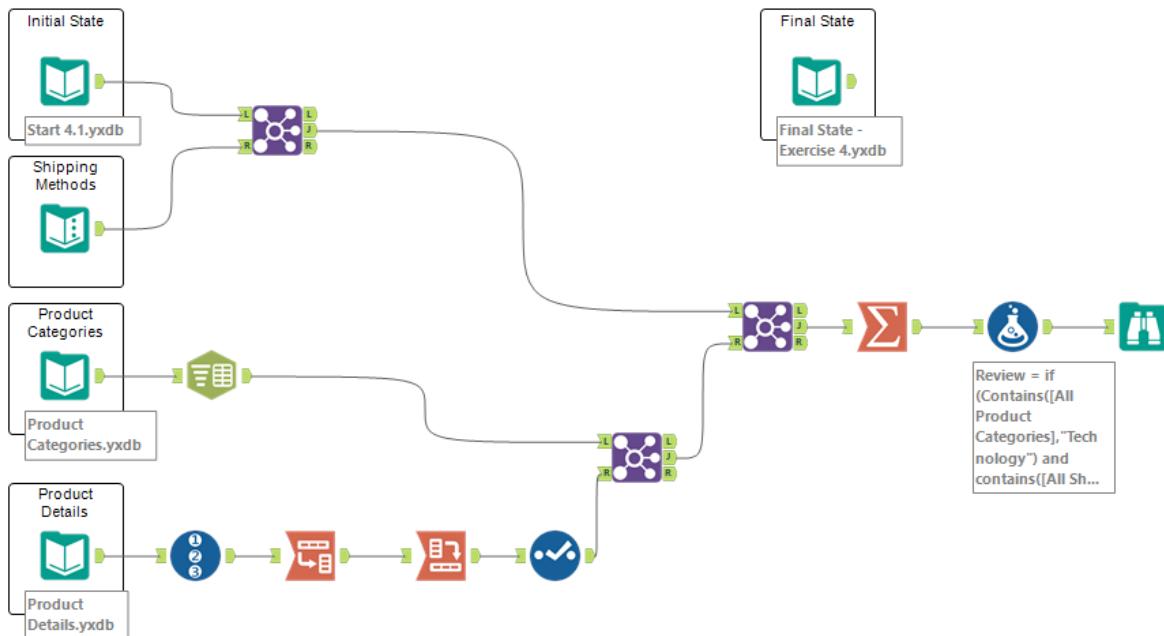
### Practice Exercise 4

Use the Shipping Methods table to add Shipping Methods as a column to "Start 4.1". Combine the "Product Details" and "Product Categories" information. Finally flag all orders containing "Technology" items that were shipped "Freight".

Associate the appropriate "Shipping Method" with each order based on its "Order\_Priority".

Then combine the "Product Categories" information with the "Product Details" info (the headers in Product Details are product SKUs).

Finally, combine both datastreams to flag all orders which contained "Technology" items and were sent via "Freight". Mark those orders with "REVIEW" and all others with null.



## File types and extensions

### File Types and Extensions:

Alteryx Workflow - .yxmd

Alteryx Analytic App - .yxwz

Alteryx Macro - .yxmc

Alteryx Field Types - .yxft

Alteryx Database - .yxdb

Alteryx Packaged Workflow - .yxzp.

## **Data Quality Categories**

### Data quality categories:

- NOT OK (Red): values contain leading or trailing whitespace
- NULL (Yellow): no values were provided
- EMPTY (Gray): strings with no values
- OK (Green): values without leading or trailing whitespace, NULL, or empty values.