Elixir Transform

Release 3.0.0

Elixir Technology Pte Ltd

Elixir Transform: Release 3.0.0

Elixir Technology Pte Ltd

Published 2014
Copyright © 2014 Elixir Technology Pte Ltd

Table of Contents

List of Figures

List of Tables

Chapter 1
Transform

Overview

Using a transform process, you can manipulate data by performing a sequence of operations according to your specific requirements. You receive exactly what you need from the database by creating processors in **Repertoire > Data Designer > Composite DataSource** or adding transforms in **Ambience > Designer > Universe Designer**. You can view the records after transformation to verify the output. The result of each transformation may be used to overwrite the original field value or add an additional field. The result can be saved as a new data source or forwarded to reports and dashboards for presentation.

The data operations impact only the results; the original data source remains unaffected, unless you choose to overwrite it. Transforms provide fast, convenient data manipulations using an intuitive user interface, without the need of writing any scripts.

When you create or edit transforms, the order of transforms is important as some operations may depend on fields created by previous steps. Where possible, the tool will prevent you from reordering items to produce an impossible sequence. For example, if Step 5 creates a field called X and Step 7 uses field X, you may be able to move Step 7 up one place to Step 6, but cannot move it above Step 5. Similarly, you cannot move Step 5 below Step 7. In addition, you cannot delete Step 5 because Step 7 depends on it. If you find that an ordering operation (move up or down) or delete operation is disabled, it means that making the change would yield a transform that could never succeed.

Launching the Transforms Editor

To create a transform using the Repository Designer, add a new composite data source as follows:

1. From the directory tree, right click the place where you want the data source to be created.

2. Select **Add** - **Datasource** as shown:

Figure 1.1. Add Datasource Option
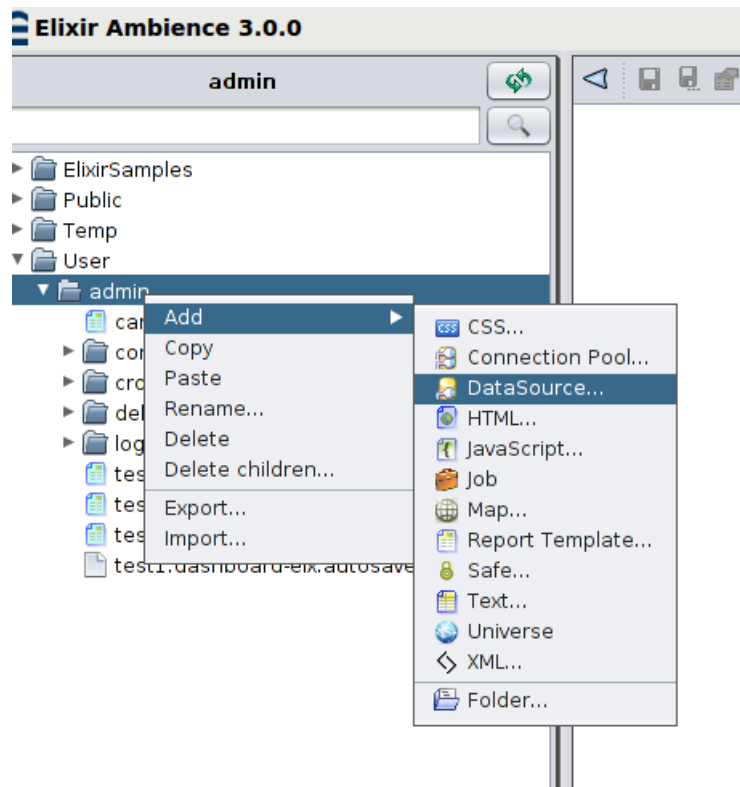
3. Select the `Composite` datasource, as shown. Then click **Next**.

Figure 1.2. Composite Datasource
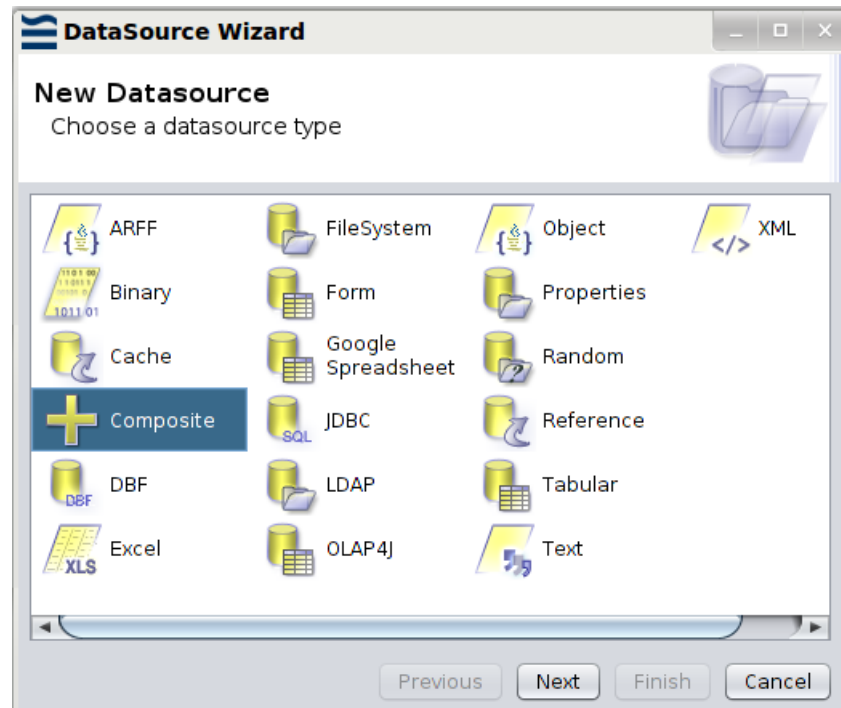
4. Enter a name and description for the datasource and click **Next**.

Figure 1.3. Naming the Datasource
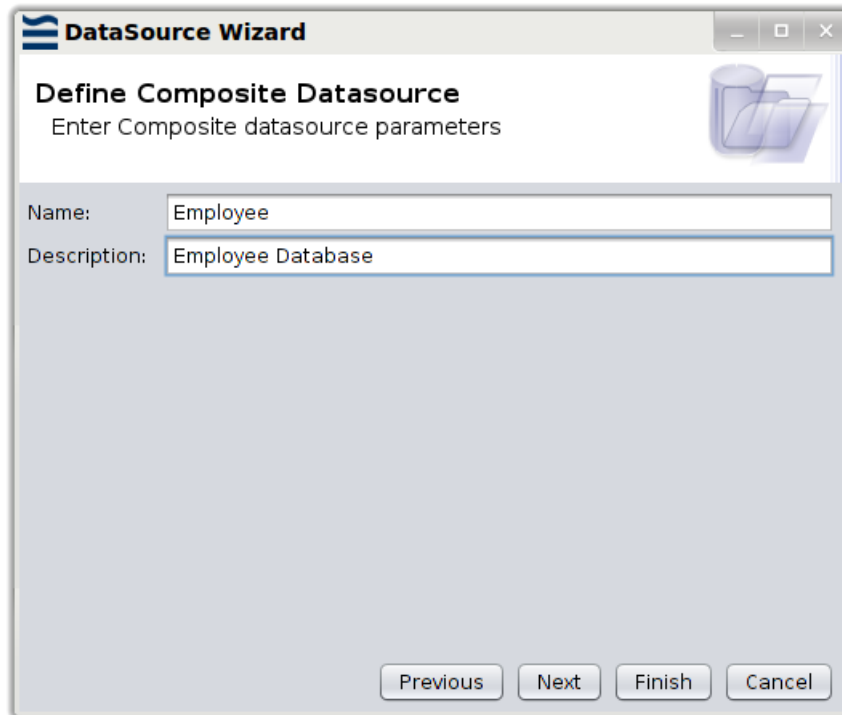
5. Optionally, set the security options as desired. The security options allow you to make the datasource read only, hide its details and encrypt it.

Figure 1.4. Securing the Datasource

6. Click **Finish** to create the datasource.

The transforms editor is displayed as follows:

Figure 1.5. Transforms Editor
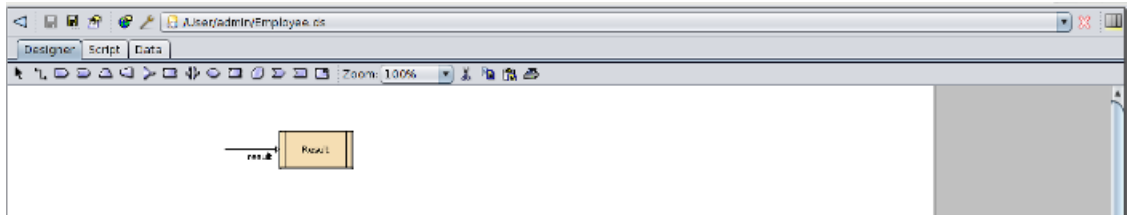
Static and Dynamic Transforms

Static transforms apply an operation to one single field at a time. Dynamic transforms work with several fields of a record simultaneously. In static transforms, you specify a constant value for data manipulation. In dynamic transforms, you use `${@Field_Name}` to call values from another field. To illustrate dynamic transforms, the following example calls values from Field 2 and adds them to values from Field 1. The operation is **Transform > Decimal > Add**:

Field 1 (Decimal)	Field 2 (Decimal)	Value	Output (Decimal)
50.6249	1.1	${@Field 2}	**51.7249**
0.721	5.2	${@Field 2}	**5.921**

Merge and Retain Operations

Many transforms include the merge or retain operation. Before using the merge or retain operation, make sure the fields have been sorted first, which minimizes memory use. Comparing each record with many other unsorted records requires a significantly large memory if there are huge volumes of data. However, if we only have to compare each record with the previous record, we can run through massive amounts of data without needing huge amounts of memory.

The merge operation ensures that no information is lost. When one field is being operated on, the merge will be between only these records, while all the other fields remain the same. This ensures that the record that is the result of the merge operation does not discard any information. If you need to merge across variations in non-essential fields, those fields will need to be discarded first.

The retain operation works similarly with merge. The only difference is that the retain operation does not discard duplicate records after the manipulation.

Example

Using transforms, how does a company calculate how much it receives from sales each month? In the following example, we will take sales values that occur throughout the month from a legacy system which for historical reasons may include spaces around the date values. Also, some date values are not available and are imported into the system as nulls. This is meant to illustrate real-world situations where the data needs to be cleansed before we can begin properly processing it.

- **Step 1:** Trim off the extra spaces, tabs and new lines in the string values. The operation is **Transform > String > Trim**. Select **In Place**
  .

Figure 1.6. Trim Option
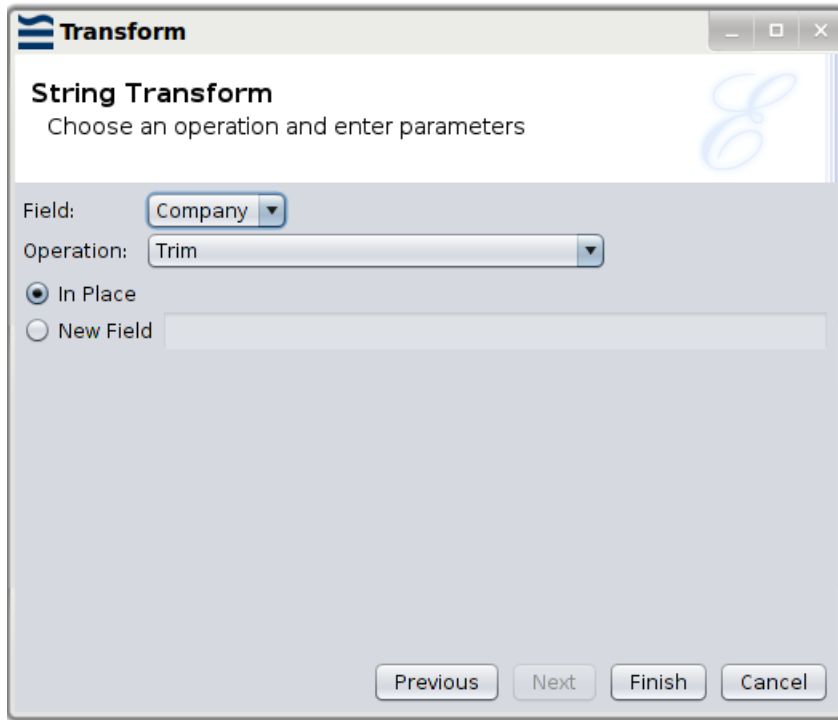
By doing this, we keep only the trimmed string values.

The input data is read from a server which may include extra spaces at the beginning and end. The date strings may look like the following. Therefore, we need to trim these spaces before parsing the date:

•   " 2011-07-31 "

•   " 2011-09-30 "

In the following table, the `When` field includes string representations of dates when a product was sold and the `Value` field shows the sale price. Later, we will introduce a `Commission` field, which describes how much of the sale price goes directly to the sales person as commission.

The When field displays the following output:

When (String)	Value (Decimal)
2011-03-28	2000.00
2011-03-07	3000.00
2011-3-19	1000.00
	2000.00
	1000.00
	3000.00
2011-2-16	2000.00
2011-2-27	1000.00
2011-02-03	2000.00
	4000.00
	1000.00

When (String)	Value (Decimal)
2011-05-12	2000.00
2011-05-16	1000.00
2011-05-07	1000.00
2011-05-24	3000.00
2011-6-07	2000.00
2011-6-15	1000.00
2011-6-21	1000.00
2011-07-09	2000.00
2011-07-14	3000.00
2011-07-26	1000.00
2011-07-29	1000.00
	1000.00
	2000.00
2011-09-17	1000.00
2011-09-05	2000.00
2011-09-11	1000.00
2011-09-23	3000.00
2011-10-16	2000.00
2011-10-04	3000.00
2011-10-27	1000.00

- **Step 2:** Convert the strings from the `When` field into dates. The operation is **Transform > String > To date**. Select **In Place**. The new values may look the same as the trimmed strings, but these are now Date objects, which means we can manipulate them as dates rather than just as sequences of characters.

In the following table, the When field shows the output:

When (Date)	Value (Decimal)
2011-03-28	2000.00
2011-03-07	3000.00
2011-03-19	1000.00
	2000.00
	1000.00
	3000.00
2011-02-16	2000.00
2011-02-27	1000.00
2011-02-03	2000.00
	4000.00
	1000.00
2011-05-12	2000.00
2011-05-16	1000.00
2011-05-07	1000.00

When (Date)	Value (Decimal)
2011-05-24	3000.00
2011-06-07	2000.00
2011-06-15	1000.00
2011-06-21	1000.00
2011-07-09	2000.00
2011-07-14	3000.00
2011-07-26	1000.00
2011-07-29	1000.00
	1000.00
	2000.00
2011-09-17	1000.00
2011-09-05	2000.00
2011-09-11	1000.00
2011-09-23	3000.00
2011-10-16	2000.00
2011-10-04	3000.00
2011-10-27	1000.00

- **Step 3:** Extract the month values from the `Date` field. The operation is **Transform > Date > Month**.

  In the following table, a new field named "Index" is created to show the output:

When (String)	Value (Decimal)	Index (Integer)
2011-03-28	2000.00	**3**
2011-03-07	3000.00	**3**
2011-03-19	1000.00	**3**
	2000.00	
	1000.00	
	3000.00	
2011-02-16	2000.00	**2**
2011-02-27	1000.00	**2**
2011-02-03	2000.00	**2**
	4000.00	
	1000.00	
2011-05-12	2000.00	**5**
2011-05-16	1000.00	**5**
2011-05-07	1000.00	**5**
2011-05-24	3000.00	**5**
2011-06-07	2000.00	**6**
2011-06-15	1000.00	**6**
2011-06-21	1000.00	**6**
2011-07-09	2000.00	**7**

When (String)	Value (Decimal)	Index (Integer)
2011-07-14	3000.00	**7**
2011-07-26	1000.00	**7**
2011-07-29	1000.00	**7**
	1000.00	
	2000.00	
2011-09-17	1000.00	**9**
2011-09-05	2000.00	**9**
2011-09-11	1000.00	**9**
2011-09-23	3000.00	**9**
2011-10-16	2000.00	**10**
2011-10-04	3000.00	**10**
2011-10-27	1000.00	**10**

- **Step 4:** Show the months in long names from the `When` field. The operation is **Transform > Date > Month name (long)**. Select **In Place**.

In the following table, the When field shows the output:

When (String)	Value (Decimal)	Index (Integer)
March	2000.00	3
March	3000.00	3
March	1000.00	3
	2000.00	
	1000.00	
	3000.00	
February	2000.00	2
February	1000.00	2
February	2000.00	2
	4000.00	
	1000.00	
May	2000.00	5
May	1000.00	5
May	1000.00	5
May	3000.00	5
June	2000.00	6
June	1000.00	6
June	1000.00	6
July	2000.00	7
July	3000.00	7
July	1000.00	7
July	1000.00	7
	1000.00	
	2000.00	

When (String)	Value (Decimal)	Index (Integer)
September	1000.00	9
September	2000.00	9
September	1000.00	9
September	3000.00	9
October	2000.00	10
October	3000.00	10
October	1000.00	10

You will notice that the nulls in the `When` field and the `Index` field are still nulls (represented by empty cells). Unless the transform specifically processes nulls, all transforms will just pass nulls through unchanged. In the next step, we will change those nulls into values we can work with.

- **Step 5:** Set the Null values from the `When` field to "Unknown". The operation is **Transform > String > Null to string**. Select **In Place**.

In the following table, the When field shows the output:

When (String)	Value (Decimal)	Index (Integer)
March	2000.00	3
March	3000.00	3
March	1000.00	3
Unknown	2000.00	
Unknown	1000.00	
Unknown	3000.00	
February	2000.00	2
February	1000.00	2
February	2000.00	2
Unknown	4000.00	
Unknown	1000.00	
May	2000.00	5
May	1000.00	5
May	1000.00	5
May	3000.00	5
June	2000.00	6
June	1000.00	6
June	1000.00	6
July	2000.00	7
July	3000.00	7
July	1000.00	7
July	1000.00	7
Unknown	1000.00	
Unknown	2000.00	
September	1000.00	9
September	2000.00	9
September	1000.00	9

When (String)	Value (Decimal)	Index (Integer)
September	3000.00	9
October	2000.00	10
October	3000.00	10
October	1000.00	10

- **Step 6:** Sort the table by month index. The operation is **Transform > Sort**. Choose the Index field to sort by. Choose the Ascending order. Leave the Max Memory (MB) field blank. By doing this, this operation will sort the sales values without a maximum memory limit.

The following table shows the output:

When (String)	Value (Decimal)	Index (Integer)
Unknown	2000.00	
Unknown	1000.00	
Unknown	3000.00	
Unknown	4000.00	
Unknown	1000.00	
Unknown	1000.00	
Unknown	2000.00	
February	2000.00	**2**
February	1000.00	**2**
February	2000.00	**2**
March	2000.00	**3**
March	3000.00	**3**
March	1000.00	**3**
May	2000.00	**5**
May	1000.00	**5**
May	1000.00	**5**
May	3000.00	**5**
June	2000.00	**6**
June	1000.00	**6**
June	1000.00	**6**
July	2000.00	**7**
July	3000.00	**7**
July	1000.00	**7**
July	1000.00	**7**
September	1000.00	**9**
September	2000.00	**9**
September	1000.00	**9**
September	3000.00	**9**
October	2000.00	**10**
October	3000.00	**10**
October	1000.00	**10**

- **Step 7:** Create a new field named `Commission` in the original data source. Sum merge the sales values and commissions, and display a single total for each month. The operation is **Transform > Reduce**. Select the **Sum** action on the Value field and the Commission field. Leave the action blank on the other fields. By doing this, you can add up the sales values and commissions and view the total for each month. The rows have been merged down to one row per group, where the month names and indexes are the groups.

  Alternatively, you can also sum retain on the sales values and commissions, and then perform a `Discard Duplicates Transform` operation to remove the redundant rows. You will receive the same output.

  The following table shows the sorted table with the Commission field added:

When (String)	Value (Decimal)	Commission (Decimal)	Index (Integer)
Unknown	2000.00	**100.20**	
Unknown	1000.00	**50.10**	
Unknown	3000.00	**150.30**	
Unknown	4000.00	**200.40**	
Unknown	1000.00	**50.10**	
Unknown	1000.00	**50.10**	
Unknown	2000.00	**100.20**	
February	2000.00	**100.20**	2
February	1000.00	**50.10**	2
February	2000.00	**100.20**	2
March	2000.00	**100.20**	3
March	3000.00	**150.30**	3
March	1000.00	**50.10**	3
May	2000.00	**100.20**	5
May	1000.00	**50.10**	5
May	1000.00	**50.10**	5
May	3000.00	**150.30**	5
June	2000.00	**100.20**	6
June	1000.00	**50.10**	6
June	1000.00	**50.10**	6
July	2000.00	**100.20**	7
July	3000.00	**150.30**	7
July	1000.00	**50.10**	7
July	1000.00	**50.10**	7
September	1000.00	**50.10**	9
September	2000.00	**100.20**	9
September	1000.00	**50.10**	9
September	3000.00	**150.30**	9
October	2000.00	**100.20**	10
October	3000.00	**150.30**	10
October	1000.00	**50.10**	10

The following table shows the output of the Reduce Transform:

When (String)	Value (Decimal)	Commission (Decimal)	Index (Integer)
Unknown	**14000.00**	**701.40**	
February	**5000.00**	**250.50**	2
March	**6000.00**	**300.60**	3
May	**7000.00**	**350.70**	5
June	**4000.00**	**200.40**	4
July	**7000.00**	**350.70**	7
September	**7000.00**	**350.70**	9
October	**6000.00**	**300.60**	10

- **Step 8:** Subtract the commission from the sales value in each month. The operation is **Transform > Decimal > Subtract**. Specify `${@Commission}` as the value to be subtracted. In this case we are not subtracting a fixed amount (a static transform) but an amount read from another field - the Commission field. This is termed a dynamic transform, because the result requires input from more than one field.

In the following table, a new field named "Received" is created to show the output:

When (String)	Value (Decimal)	Commission (Decimal)	Index (Integer)	Received (Decimal)
Unknown	14000.00	701.40		**13298.60**
February	5000.00	250.50	2	**4749.50**
March	6000.00	300.60	3	**5699.40**
May	7000.00	350.70	5	**6649.30**
June	4000.00	200.40	4	**3799.60**
July	7000.00	350.70	7	**6649.30**
September	7000.00	350.70	9	**6649.30**
October	6000.00	300.60	10	**5699.40**

Now we can save the result of the transformations as a new data source for later processing. The transforms are repeatable, which means you can re-run the transforms, each month, to receive the most updated set of numbers, rather than create a sequence of transforms but run them only once.

Summary

The following is a summary of the transformations we used in the overview:

- **String trim:** remove whitespace at the beginning and end of date strings.

- **String to date:** build date objects.

- **Date to month:** extract month index from date objects.

- **Date to month name (long):** replace date objects with long month names.

- **Null to string:** set the null month names to "Unknown".

- **Sort by month index:** view the sales values throughout each month in the ascending order.

- **Reduce:** sum merge the sales values and commissions, reduce the redundant rows and display a single total for each month.

- **Subtract:** subtract the commissions from the sales values to calculate how much the company receives from sales each month.

You can find more details on these and all other supported transformations in the following chapters.

Chapter 2
Boolean and String Transforms

Boolean Transform

Data in Boolean type can be converted to String or Integer (1 for "true", and 0 for "false"). You can perform the following operations:

And

This operation applies to at least two selected fields. It follows the logical operation rules for the "And" operator. After selecting the field to operate on, specify other field names following the "A,B,C" syntax.

Field A	Field B	Output
true	true	true
false	true	false
true	false	false
false	false	false

Discard

This operation discards the rows that contain the "true" value in the selected field.

The following table shows an example of the input:

Field 1	Field 2
1	false
2	true
3	true
4	false

The following table shows the output:

Field 1	Field 2
1	false
4	false

Not

This operation applies to at least two selected fields. It follows the logical operation rules for the "Not" operator.

Input	Output
true	false

Input	Output
false	true

Or

This operation applies to at least two selected fields. It follows the logical operation rules for the "Or" operator. After selecting the field to operate on, specify other field names following the "A,B,C" syntax.

Field A	Field B	Output
true	true	true
false	true	true
true	false	true
false	false	false

Retain

This operation retains the rows that contain the "true" value in the selected field.

The following table shows an example of the input:

Field 1	Field 2
1	false
2	true
3	true
4	false

The following table shows the output:

Field 1	Field 2
2	true
3	true

To either

This operation enables you to specify new values (for example, "Yes" and "No", or "Success" and "Fail") for "true" and "false", and displays the result according to your input.

Input	Values	Output
true	true="Yes", false="No"	Yes
false	true="Yes", false="No"	No

String Transform

Data in String type can be converted to Date, Decimal, Float, Integer, Time or Timestamp type. You can also perform the following operations:

Add prefix

This operation enables you to specify a prefix and adds it in front of the strings.

Input	Prefix	Output
abc	XYZ	XYZabc
def	XYZ	XYZdef

Add suffix

This operation enables you to specify a suffix and adds it at the end of the strings.

Input	Suffix	Output
abc	XYZ	abcXYZ
def	XYZ	defXYZ

Ascii to zenkaku alphanumeric

This operation converts the string encoding from ascii to zenkaku alphanumeric.

Input	Output
0x6D	０ｘ６Ｄ

Capitalize

This operation capitalizes the first character in each string.

Input	Output
abc	Abc
def	Def

Count

This operation counts the number of characters in each string.

Input	Output
abc	3
defgh	5

Ends with

This operation enables you to specify a suffix, checks if the strings end with this suffix and returns Boolean values (true or false).

Input	Suffix	Output
abc	z	false
xyz	z	true

Enumeration to boolean

Before starting this operation, you must specify nominal or ordinal attributes for the specified field in a processor. These attributes should be values from the field. Create another processor for this operation, and link these 2 processors with the Flow connector.

This operation generates one or more new fields in Boolean type depending on the attributes. New field names will be the same with attribute names. If a row value equals the attribute value, it returns "true". Otherwise it returns "false".

The following table shows an example when the nominal or ordinal attributes are:

```
US
Mexico
```

Input	US	Mexico
US	true	false
Canada	false	false
Mexico	false	true

Enumeration to integer

Before starting this operation, you must specify nominal or ordinal attributes for the specified field in a processor. These attributes should be values from the field. Create another processor for this operation, and link these 2 processors with the Flow connector.

This operation generates one or more new fields in Integer type depending on the attributes. New field names will be the same with attribute names. If a row value equals the attribute value, it returns "1". Otherwise it returns "0".

The following table shows an example when the nominal or ordinal attributes are:

```
US
Mexico
```

Input	US	Mexico
US	1	0
Canada	0	0
Mexico	0	1

Extract

This operation enables you to specify a Regular Expression (RegExp) and extract characters from strings according to this expression.

Input	RegExp	Output
abcdef	..(.*)	cdef
uvwxyz	..(.*)	wxyz

First

This operation enables you to specify a length value and extract the specified number of characters from the beginning of each string.

Input	Length	Output
abcdef	2	ab
uvwxyz	2	uv

Hankaku katakana to zenkaku katakana

This operation converts Japanese characters from hankaku katakana to zenkaku katakana.

Input	Output
ｺﾝﾋﾟｭｰﾀ	コンピュータ

Hiragana to katakana

This operation converts Japanese characters from hiragana to katakana.

Input	Output
そふとうぇあ	ソフトウェア

Index of

This operation returns integers that indicate the first occurrence of the specified value in each string. The search proceeds from left to right. Returning "-1" means that the value cannot be found.

Input	Value	Output
Apple	r	-1
Berry	r	2
Orange	r	1
Strawberry	r	2

Katakana to hiragana

This operation converts Japanese characters from katakana to hiragana.

Input	Output
データベース	でーたべーす

Last

This operation enables you to specify a length value and extracts a certain length of characters from the end of each string.

Input	Length	Output
abcdef	2	ef
uvwxyz	2	yz

Last index of

This operation returns integers to find out where is the last occurrence of the specified value in each string. Returning "-1" means that the value cannot be found.

Input	Value	Output
Apple	r	-1
Berry	r	3
Orange	r	1
Strawberry	r	8

Left pad

This operation enables you to specify a prefix and a length value, and extends the length of strings by repeating the prefix from the left end. If the specified length is smaller than the original length, strings are unchanged.

Input	Prefix	Length	Output
abc	Z	6	ZZZabc
defghijk	Z	6	defghijk

Left trim

This operation trims off the extra spaces, tabs and new lines from the left end of strings. For example, you can left trim the string " 2011-10-31" and receive the output "2011-10-31".

Lower

This operation converts the uppercase letters into lowercase.

Input	Output
ABC	abc
Def	def

Matches

This operation enables you to specify a Regular Expression (RegExp), checks if the strings match this expression and returns Boolean values (true or false).

Input	RegExp	Output
abcd	...(.*)	true
ef	...(.*)	false

Merge

This operation merges strings from the specified field by connecting them with a separator, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	l
A	B	C	10	m
A	B	C	10	n

Field 1	Field 2	Field 3	Field 4	Field 5
D	E	F	0	p
D	E	F	0	q

The following table shows an example of the output (Separator: "."):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	l.m.n
D	E	F	0	p.q

Null to string

This operation displays the Null strings as a specified value, while the other values remain the same.

Input	Value	Output
	XYZ	XYZ
abc	XYZ	abc

Replace

This operation finds a specified keyword from strings and replaces it with a specified string value.

Input	Keyword	Value	Output
abc	a	Z	Zbc
def	a	Z	def

Right pad

This operation enables you to specify a suffix and a length value, and extend the length of strings by repeating the suffix from the right end. If the specified length is smaller than the original length, strings are unchanged.

Input	Suffix	Length	Output
abc	Z	6	abcZZZ
defghijk	Z	6	defghijk

Right trim

This operation trims off the extra spaces, tabs and new lines from the right of strings. For example, you can right trim the string "2011-10-31 " and receive the output "2011-10-31".

Slice

This operation enables you to specify two integer values, which indicate the start and end of the slice operation.

Input	From, To	Output
a/b/c	1, 3	/b
d.e.f	1, 3	.e

Split

This operation splits strings according to a specified separator, and displays the split characters in separate rows.

The following table shows an example of the input:

Input
a/b/c
X/Y

The following table shows an example of the output (Separator: "/"):

Output
a
b
c
X
Y

Starts with

This operation enables you to specify a prefix, check if the strings start with this prefix and return a Boolean value (true or false).

Input	Prefix	Output
abc	a	true
def	a	false

Trim

This operation trims off the extra spaces, tabs and new lines from both ends of strings. For example, you can trim the string " 2011-10-31 " and receive the output "2011-10-31".

Upper

This operation converts the lowercase letters into uppercase.

Input	Output
abc	ABC
Def	DEF

Zenkaku alphanumeric to ascii

This operation converts the string encoding from zenkaku alphanumeric to ascii.

Input	Output
０ｘ６Ｅ	0x6E

Zenkaku katakana to hankaku katakana

This operation converts Japanese characters from zenkaku katakana to hankaku katakana.

Input	Output
ハードウェア	ﾊｰﾄﾞ ｳｪｱ

Chapter 3
Decimal, Float and Integer Transforms

Decimal Transform

Data in Decimal type can be converted to Float, Integer or String type. You can perform the following operations:

Add

This operation adds a specified value to the decimal.

Input	Value	Output
50.6249	1.1	51.7249
0.721	1.1	1.821

Average merge

This operation calculates the values from the decimal field into the average and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one decimal field, this operation directly calculates the values into the average.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.1
D	E	F	0	2.2

Average retain

This operation is similar to the Average merge operation. The differences are in the following:

- It enables you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.1
A	B	C	10	7.1
A	B	C	10	7.1
D	E	F	0	2.2
D	E	F	0	2.2

Divide

This operation divides the decimal by a specified value.

Input	Value	Output
50.6249	2	25.31245
0.721	2	0.3605

Max

This operation changes each value in the selected field to a specified maximum value. If the original value is less than the specified maximum, the result becomes the maximum, otherwise it is unchanged.

Input	Maximum	Output
1.1051	10.1051	10.1051
11.2345	10.1051	11.2345

Max merge

This operation merges the values from the decimal field into the maximum and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one decimal field, this operation directly returns the maximum.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1

Field 1	Field 2	Field 3	Field 4	Field 5
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	9.1
D	E	F	0	3.2

Max retain

This operation is similar to the Max merge operation. The differences are in the following:

- It enables you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	9.1
A	B	C	10	9.1
A	B	C	10	9.1
D	E	F	0	3.2
D	E	F	0	3.2

Median merge

This operation merges the values from the decimal field into a median value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one decimal field, this operation directly returns the median. In most cases, the output should be an existing value most close to the average. If there are only two values in the decimal field, it returns the average directly.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.78
A	B	C	10	9.1

Field 1	Field 2	Field 3	Field 4	Field 5
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.78
D	E	F	0	2.2

Median retain

This operation is similar to the Median merge operation. The differences are in the following:

• It allows you to select a Group Field.

• It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.78
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.78
A	B	C	10	7.78
A	B	C	10	7.78
D	E	F	0	2.2
D	E	F	0	2.2

Min

This operation changes each value in the selected field to a specified minimum value. If the original value is greater than the specified minimum, the result becomes the minimum, otherwise it is unchanged.

Input	Maximum	Output
10.1051	1.1051	1.1051
0.2345	1.1051	0.2345

Min merge

This operation merges the values from the decimal field into the minimum and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information.

These rows will become one row as a result of merge. If there is only one decimal field, this operation directly returns the minimum.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
D	E	F	0	1.2

Min retain

This operation is similar to the Min merge operation. The differences are in the following:

• It allows you to select a Group Field.

• It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	5.1
A	B	C	10	5.1
D	E	F	0	1.2
D	E	F	0	1.2

Multiply

This operation multiplies a specified value with the decimal.

Input	Value	Output
50.6249	2	101.2498

Input	Value	Output
0.721	2	1.442

Percent

This operation calculates the percentage of values in the corresponding rows from the decimal field, when the corresponding rows from the Group Field include exactly the same information. Otherwise, it returns 1, which means 100 percent. The summary of the output from the corresponding rows equals to 1.

Field 1	Field 2	Output (Group Field: Field 1)
A	2.0001	0.20001
A	3.0001	0.30001
A	4.9998	0.49998
B	4.0001	1
C	4.9998	1

Product merge

This operation calculates the values from the decimal field into the product value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one decimal field, this operation directly calculates the values into the product.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	329.511
D	E	F	0	3.84

Product retain

This operation is similar to the Product merge operation. The differences are in the following:

• It allows you to select a Group Field.

• It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	329.511
A	B	C	10	329.511
A	B	C	10	329.511
D	E	F	0	3.84
D	E	F	0	3.84

Remainder

This operation returns the remainder when the values in the decimal field are divided by a specified value.

Field 1	Value	Output
3.0001	2	1.0001
4.9998	2	0.9998

Round down

This operation calculates the round-down value of the decimal.

Specify the **Decimals** value, which indicates how many decimal places to keep on the rounding choice. The default **Decimals** value is "0", which means it will not keep any decimal places. You can also use dynamic transform by setting this value to `${Parameter_Name}` or `${@Field_Name}`.

Field 1	Field 2	Decimals	Output
50.9999	1	${@Field 2}	50.9
2.163	2	${@Field 2}	2.16

Round half down

This operation calculates the round-half-down value of the decimal. For instructions on how to keep decimal places by setting the **Decimals** value, refer to the "Round down" section.

Field 1	Field 2	Decimals	Output
50.9999	1	${@Field 2}	51.0
2.163	2	${@Field 2}	2.16

Round half even

This operation calculates the round-half-even value of the decimal. For instructions on how to keep decimal places by setting the **Decimals** value, refer to the "Round down" section.

Field 1	Field 2	Decimals	Output
4.00012	3	${@Field 2}	4.000
4.99987	4	${@Field 2}	4.9999

Round half up

This operation calculates the round-half-up value of the decimal. For instructions on how to keep decimal places by setting the **Decimals** value, refer to the "Round down" section.

Field 1	Field 2	Decimals	Output
7.60853	3	${@Field 2}	7.609
3.57395	4	${@Field 2}	3.5740

Round up

This operation calculates the round-up value of the decimal. For instructions on how to keep decimal places by setting the **Decimals** value, refer to the "Round down" section.

Field 1	Field 2	Decimals	Output
2.163	2	${@Field 2}	2.17
4.00012	3	${@Field 2}	4.001

Standard deviation merge

This operation calculates the values from the decimal field into the standard deviation value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one decimal field, this operation directly calculates the values into the standard deviation.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	2
D	E	F	0	1.4142*

*This is an approximate value of the output for the convenience and is used for illustration.

Standard deviation retain

This operation is similar to the Standard deviation merge operation. The differences are in the following:

• It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	2
A	B	C	10	2
A	B	C	10	2
D	E	F	0	1.4142*
D	E	F	0	1.4142*

*This is an approximate value of the output for the convenience of illustration.

Subtract

This operation subtracts a specified value from the decimal.

Input	Value	Output
50.6249	1.1	49.5249
10.537	1.1	9.437

Sum merge

This operation calculates the values from the decimal field into the summary and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one decimal field, this operation directly calculates the values into the summary.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	21.3

Field 1	Field 2	Field 3	Field 4	Field 5
D	E	F	0	4.4

Sum retain

This operation is similar to the Sum merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	21.3
A	B	C	10	21.3
A	B	C	10	21.3
D	E	F	0	4.4
D	E	F	0	4.4

To ceiling

This operation calculates the ceiling value of the decimal.

Input	Output
50.9999	51.0
2.163	3.0

To floor

This operation calculates the floor value of the decimal.

Input	Output
50.9999	50.0
2.163	2.0

Variance merge

This operation calculates the values from the decimal field into the variance and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information.

These rows will become one row as a result of merge. If there is only one decimal field, this operation directly calculates the values into the variance.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	4.00
D	E	F	0	2.00

Variance retain

This operation is similar to the Variance merge operation. The differences are in the following:

• It allows you to select a Group Field.

• It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	4.00
A	B	C	10	4.00
A	B	C	10	4.00
D	E	F	0	2.00
D	E	F	0	2.00

Float Transform

Data in Float type can be converted to Decimal, Integer or String type. You can perform the following operations:

Add

This operation adds a specified value to the float value.

Input	Value	Output
50.6249	1.1	51.7249
0.721	1.1	1.821

Average merge

This operation calculates the values from the float field into the average and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly calculates the values into the average.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.1
D	E	F	0	2.2

Average retain

This operation is similar to the Average merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.1

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.1
A	B	C	10	7.1
D	E	F	0	2.2
D	E	F	0	2.2

Divide

This operation divides the float value by a specified value.

Input	Value	Output
50.6249	2	25.31245
0.721	2	0.3605

Max

This operation changes each value in the selected field to a specified maximum value. If the original value is less than the specified maximum, the result becomes the maximum, otherwise it is unchanged.

Input	Maximum	Output
1.1051	10.1051	10.1051
11.2345	10.1051	11.2345

Max merge

This operation merges the values from the float field into the maximum and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly returns the maximum.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	9.1
D	E	F	0	3.2

Max retain

This operation is similar to the Max merge operation. The differences are in the following:

• It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	9.1
A	B	C	10	9.1
A	B	C	10	9.1
D	E	F	0	3.2
D	E	F	0	3.2

Median merge

This operation merges the values from the float field into a median value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly returns the median. In most cases, the output should be an existing value most close to the average. If there are only two values in the float field, it returns the average directly.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.78
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.78
D	E	F	0	2.2

Median retain

This operation is similar to the Median merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.78
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.78
A	B	C	10	7.78
A	B	C	10	7.78
D	E	F	0	2.2
D	E	F	0	2.2

Min

This operation changes each value in the selected field to a specified minimum value. If the original value is greater than the specified minimum, the result becomes the minimum, otherwise it is unchanged.

Input	Maximum	Output
10.1051	1.1051	1.1051
0.2345	1.1051	0.2345

Min merge

This operation merges the values from the float field into the minimum and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly returns the minimum.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
D	E	F	0	1.2

Min retain

This operation is similar to the Min merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	5.1
A	B	C	10	5.1
D	E	F	0	1.2
D	E	F	0	1.2

Multiply

This operation multiplies a specified value with the float value.

Input	Value	Output
50.6249	2	101.2498
0.721	2	1.442

Percent

This operation calculates the percentage of values in the corresponding rows from the float field, when the corresponding rows from the Group Field include exactly the same information. Otherwise, it returns 1, which means 100 percent. The summary of the output from the corresponding rows equals to 1.

Field 1	Field 2	Output (Group Field: Field 1)
A	2.0001	0.20001*
A	3.0001	0.30001
A	4.9998	0.49998
B	4.0001	1.0
C	4.9998	1.0

*This is an approximate value of the output for the convenience of illustration.

Product merge

This operation calculates the values from the float field into the product value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly calculates the values into the product.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	329.511
D	E	F	0	3.84

Product retain

This operation is similar to the Product merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	329.511
A	B	C	10	329.511
A	B	C	10	329.511
D	E	F	0	3.84
D	E	F	0	3.84

Remainder

This operation returns the remainder when the values in the float field are divided by a specified value.

Field 1	Value	Output
3.0001	2	1.0001
4.9998	2	0.9998

Round down

This operation calculates the round-down value of the float value.

Input	Output
50.9999	50
2.163	2

Round half down

This operation calculates the round-half-down value of the float value.

Input	Output
50.9999	51
2.163	2

Round half even

This operation calculates the round-half-even value of the float value.

Input	Output
50.9999	51
2.163	2

Round half up

This operation calculates the round-half-up value of the float value.

Input	Output
50.9999	51
2.163	2

Round up

This operation calculates the round-up value of the float value.

Input	Output
50.9999	51
2.163	3

Standard deviation merge

This operation calculates the values from the float field into the standard deviation value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly calculates the values into the standard deviation.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	2.0
D	E	F	0	1.4142*

*This is an approximate value of the output for the purposes of illustration.

Standard deviation retain

This operation is similar to the Standard deviation merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	2.0
A	B	C	10	2.0
A	B	C	10	2.0
D	E	F	0	1.4142*
D	E	F	0	1.4142*

*This is an approximate value of the output for the purpose of illustration.

Subtract

This operation subtracts a specified value from the float value.

Input	Value	Output
50.6249	1.1	49.5249
10.537	1.1	9.437

Sum merge

This operation calculates the values from the float field into the summary and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly calculates the values into the summary.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	21.3
D	E	F	0	4.4

Sum retain

This operation is similar to the Sum merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	21.3

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	21.3
A	B	C	10	21.3
D	E	F	0	4.4
D	E	F	0	4.4

To ceiling

This operation calculates the ceiling value of the float value.

Input	Output
50.9999	51.0
2.163	3.0

To floor

This operation calculates the floor value of the float value.

Input	Output
50.9999	50.0
2.163	2.0

Variance merge

This operation calculates the values from the float field into the variance and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one float field, this operation directly calculates the values into the variance.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	4.0
D	E	F	0	2.0*

*This is an approximate value of the output for the purpose of illustration.

Variance retain

This operation is similar to the Variance merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Group Field: Field 1):

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	4.0
A	B	C	10	4.0
A	B	C	10	4.0
D	E	F	0	2.0*
D	E	F	0	2.0*

*This is an approximate value of the output for the purpose of illustration.

Integer Transform

Data in Integer type can be converted to Date, Decimal, Float, Hex string, String, Time or Timestamp type. You can also perform the following operations:

Add

This operation adds a specified integer value to the integer.

Input	Value	Output
50	1	51
2	1	3

Average merge

This operation calculates the values from the integer field into the average and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly calculates the values into the average.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9

Field 1	Field 2	Field 3	Field 4	Field 5
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.0
D	E	F	0	2.0

Average retain

This operation is similar to the Average merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output (Group Field: Field 1), which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7.0
A	B	C	10	7.0
A	B	C	10	7.0
D	E	F	0	2.0
D	E	F	0	2.0

Divide

This operation divides the integer by a specified integer value.

Input	Value	Output
50	2	25
2	2	1

Max

This operation changes each value in the selected field to a specified maximum value. If the original value is less than the specified maximum, the result becomes the maximum, otherwise it is unchanged.

Input	Maximum	Output
1	10	10
11	10	11

Max merge

This operation merges the values from the integer field into the maximum and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly returns the maximum.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Integer type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	9
D	E	F	0	3

Max retain

This operation is similar to the Max merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output (Group Field: Field 1), which is in Integer type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	9
A	B	C	10	9
A	B	C	10	9
D	E	F	0	3

Field 1	Field 2	Field 3	Field 4	Field 5
D	E	F	0	3

Median merge

This operation merges the values from the integer field into a median value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly returns the median. In most cases, the output should be an existing value most close to the average. If there are only two values in the integer field, it returns the average directly.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	8
A	B	C	10	9
D	E	F	0	1
D	E	F	0	2

The following table shows an example of the output, which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	8.0
D	E	F	0	1.5

Median retain

This operation is similar to the Median merge operation. The differences are in the following:

• It allows you to select a Group Field.

• It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	8
A	B	C	10	9
D	E	F	0	1
D	E	F	0	2

The following table shows an example of the output, which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	8.0
A	B	C	10	8.0
A	B	C	10	8.0
D	E	F	0	1.5

Field 1	Field 2	Field 3	Field 4	Field 5
D	E	F	0	1.5

Min

This operation changes each value in the selected field to a specified minimum value. If the original value is greater than the specified minimum, the result becomes the minimum, otherwise it is unchanged.

Input	Maximum	Output
10	1	1
0	1	0

Min merge

This operation merges the values from the integer field into the minimum and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly returns the minimum.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Integer type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
D	E	F	0	1

Min retain

This operation is similar to the Min merge operation. The differences are in the following:

• It allows you to select a Group Field.

• It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Integer type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	5
A	B	C	10	5
D	E	F	0	1
D	E	F	0	1

Multiply

This operation multiplies a specified integer value with the integer.

Input	Value	Output
50	2	100
2	2	4

Percent

This operation calculates the percentage of values in the corresponding rows from the integer field, when the corresponding rows from the Group Field include exactly the same information. Otherwise, it returns 1, which means 100 percent. The summary of the output from the corresponding rows equals to 1. The output is in Float type.

Field 1	Field 2	Output (Group Field: Field 1)
A	2	0.2222*
A	3	0.3333*
A	4	0.4444*
B	4	1.0
C	4	1.0

*This is an approximate value of the output for the purpose of illustration.

Product merge

This operation calculates the values from the integer field into the product value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly calculates the values into the product.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Integer type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	315
D	E	F	0	3

Product retain

This operation is similar to the Product merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Integer type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	315
A	B	C	10	315
A	B	C	10	315
D	E	F	0	3
D	E	F	0	3

Remainder

This operation returns the remainder when the values in the integer field are divided by a specified integer value.

Field 1	Value	Output
50	3	2
2	3	2

Standard deviation merge

This operation calculates the values from the integer field into the standard deviation value and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly calculates the values into the standard deviation.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	2.0
D	E	F	0	1.4142*

*This is an approximate value of the output for the purpose of illustration.

Standard deviation retain

This operation is similar to the Standard deviation merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	2.0
A	B	C	10	2.0
A	B	C	10	2.0
D	E	F	0	1.4142*
D	E	F	0	1.4142*

*This is an approximate value of the output for the purpose of illustration.

Subtract

This operation subtracts a specified integer value from the integer.

Input	Value	Output
50	1	49
2	1	1

Sum merge

This operation calculates the values from the integer field into the summary and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly calculates the values into the summary.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	21
D	E	F	0	4

Sum retain

This operation is similar to the Sum merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	21
A	B	C	10	21
A	B	C	10	21
D	E	F	0	4
D	E	F	0	4

Variance merge

This operation calculates the values from the integer field into the variance and keeps the other fields unchanged, when the corresponding rows from the other fields include exactly the same information. These rows will become one row as a result of merge. If there is only one integer field, this operation directly calculates the values into the variance.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output, which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	4.0
D	E	F	0	2.0

Variance retain

This operation is similar to the Variance merge operation. The differences are in the following:

- It allows you to select a Group Field.

- It retains the number of rows, instead of merging the corresponding rows into one.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5
A	B	C	10	7
A	B	C	10	9
D	E	F	0	1
D	E	F	0	3

The following table shows an example of the output (Group Field: Field 1), which is in Float type:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	4.0
A	B	C	10	4.0
A	B	C	10	4.0
D	E	F	0	2.0
D	E	F	0	2.0

Chapter 4
Date, Time and Timestamp Transforms

Date Transform

Data in Date type can be converted to Decimal, Integer or String type. You can perform the following operations:

Add days

This operation adds an increment value to the day.

Input	Increment	Output
2009-12-27	5	2010-01-01
2009-12-14	5	2009-12-19

Add months

This operation adds an increment value to the month.

Input	Increment	Output
2009-05-14	2	2009-07-14
2009-12-14	2	2010-02-14

Add years

This operation adds an increment value to the year.

Input	Increment	Output
2009-12-14	1	2010-12-14
2011-12-01	1	2012-12-01

Date difference (in days)

This operation shows the date difference in days between the specified value and the value from the selected field.

Input	Value	Output
2011-10-08	2011-01-01	280
2011-11-20	2011-01-01	323

Day

This operation shows only the day value from the selected field.

Input	Output
2009-12-14	14
2011-06-23	23

First of month

This operation changes the day value to the first day of each month.

Input	Output
2009-12-14	2009-12-01
2011-06-23	2011-06-01

Last of month

This operation changes the day value to the last day of each month.

Input	Output
2011-02-03	2011-02-28
2008-02-03	2008-02-29

Max

This operation changes each value in the selected field to a specified maximum value. If the original value is less than the specified maximum, the result becomes the maximum, otherwise it is unchanged.

Input	Maximum	Output
2011-04-05	2011-01-01	2011-04-05
2010-01-01	2011-01-01	2011-01-01

Min

This operation changes each value in the selected field to a specified minimum value. If the original value is greater than the specified minimum, the result becomes the minimum, otherwise it is unchanged.

Input	Minimum	Output
2011-01-01	2011-04-05	2011-01-01
2011-05-01	2011-04-05	2011-04-05

Month

This operation shows only the month value from the selected field.

Input	Output
2009-12-14	12
2011-06-23	6

Month name (long)

This operation shows the month value from the selected field in a long name.

Input	Output
2009-12-14	December
2011-06-23	June

Month name (short)

This operation shows the month value from the selected field in a short name.

Input	Output
2009-12-14	Dec
2011-06-23	Jun

Null to date

This operation displays the Null dates as a specified value, while the other dates remain the same.

Input	Value	Output
	2000-01-01	2000-01-01
2008-06-23	2000-01-01	2008-06-23

Quarter

This operation returns an integer to show the quarter to which the specified date belongs.

Input	Output
2010-10-16	4
2011-02-07	1
2011-04-19	2
2011-07-20	3

Quarter name (short)

This operation returns a string to show the quarter to which the specified date belongs.

Input	Output
2010-10-16	Q4
2011-02-07	Q1
2011-04-19	Q2
2011-07-20	Q3

Set day

This operation enables you to set the day value in the selected field.

Input	Day	Output
2011-01-01	20	2011-01-20
2009-07-17	20	2009-07-20

Set month

This operation enables you to set the month value in the selected field.

Input	Month	Output
2011-01-01	1	2011-01-01
2009-07-17	1	2009-01-17

Set year

This operation enables you to set the year value in the selected field.

Input	Year	Output
2011-01-01	2011	2011-01-01
2009-07-17	2011	2011-07-17

Subtract days

This operation subtracts a decrement value from the day.

Input	Decrement	Output
2009-12-27	5	2009-12-22
2009-12-05	5	2009-11-30

Subtract months

This operation subtracts a decrement value from the month.

Input	Decrement	Output
2009-12-14	1	2009-11-14
2009-01-22	1	2008-12-22

Subtract years

This operation subtracts a decrement value from the year.

Input	Decrement	Output
2009-12-27	2	2007-12-27
2011-02-05	2	2009-02-05

To string format

This operation transforms dates into a specified String format. Use "dd" for day, "MM" for month, and "yy" or "yyyy" for year. Specify a punctuation to connect the string.

Input	String format	Output
2009-12-27	dd/MM/yyyy	27/12/2009
2011-01-05	dd/MM/yyyy	05/01/2011

Truncate

This operation changes the hour, minute and second values to zero. In most cases, the result looks the same with the original dates. However, manipulation has taken place to ensure that there are no hidden hours, minutes or seconds. Therefore, comparing and sorting will remain unaffected by those hidden values.

Input	Output
2009-12-27	2009-12-27
2010-12-05	2010-12-05

Year

This operation shows only the year value from the selected field.

Input	Output
2009-12-14	2009
2011-06-23	2011

Time Transform

Data in Time type can be converted to Decimal, Integer or String type. You can perform the following operations:

Add hours

This operation adds an increment value to the hour.

Input	Increment	Output
23:30:15	1	00:30:15
17:07:12	1	18:07:12

Add milliseconds

This operation adds an increment value to the millisecond.

Input	Increment	Output
23:30:15	10000	23:30:25
17:07:12	10000	17:07:22

Add minutes

This operation adds an increment value to the minute.

Input	Increment	Output
23:30:15	30	00:00:15

Input	Increment	Output
17:07:12	30	17:37:12

Add seconds

This operation adds an increment value to the second.

Input	Increment	Output
23:30:15	45	23:31:00
17:07:12	45	17:07:57

Am

This operation transforms the time into Boolean type. If the time is before noon, it is changed to "true", otherwise it is changed to "false".

Input	Output
16:51:02	false
11:29:46	true

Hours

This operation shows only the hours from the selected field.

Input	Output
09:28:07	9
17:56:39	17

Max

This operation changes each value in the selected field to a specified maximum value. If the original value is less than the specified maximum, the result becomes the maximum, otherwise it is unchanged.

Input	Maximum	Output
07:30:00	07:05:00	07:30:00
07:01:00	07:05:00	07:05:00

Milliseconds

This operation shows only the milliseconds from the selected field.

Input	Output
00:30:10	0
07:31:41	0

Min

This operation changes each value in the selected field to a specified minimum value. If the original value is greater than the specified minimum, the result becomes the minimum, otherwise it is unchanged.

Input	Minimum	Output
07:05:00	07:30:00	07:05:00
07:40:00	07:30:00	07:30:00

Minutes

This operation shows only the minutes from the selected field.

Input	Output
09:28:07	28
17:56:39	56

Null to time

This operation displays the Null time as a specified value, while the other values remain the same.

Input	Value	Output
	00:00:00	00:00:00
07:30:10	00:00:00	07:30:10

Pm

This operation transforms the time into Boolean type. If the time is between noon and midnight, it is changed to "true", otherwise changed to "false".

Input	Output
16:51:02	true
11:29:46	false

Seconds

This operation shows only the seconds from the selected field.

Input	Output
09:28:07	7
17:56:39	39

Set hour

This operation enables you to set the hour value in the selected field.

Input	Hour	Output
00:30:10	20	20:30:10
07:31:41	20	20:31:41

Set millisecond

This operation enables you to set the millisecond value in the selected field. If the specified value is not big enough to calculate into seconds, minutes or hours, the result may look the same with the original value. However, the hidden millisecond value can affect comparing and sorting.

Input	Value	Output
00:30:10	1	00:30:10
07:31:41	1	07:31:41

Set minute

This operation enables you to set the minute value in the selected field.

Input	Value	Output
00:30:10	1	00:01:10
07:31:41	1	07:01:41

Set second

This operation enables you to set the second value in the selected field.

Input	Value	Output
00:30:10	1	00:30:01
07:31:41	1	07:31:01

Subtract hours

This operation subtracts a decrement value from the hours field.

Input	Decrement	Output
23:30:15	5	18:30:15
17:07:12	5	12:07:12

Subtract milliseconds

This operation subtracts a decrement value from the milliseconds. If the decrement value is not big enough to affect seconds, minutes or hours, the result may look the same with the original value. However, the hidden millisecond value can affect comparing and sorting.

Input	Decrement	Output
23:30:15	30	23:30:15
17:07:12	30	17:07:12

Subtract minutes

This operation subtracts a decrement value from the minutes.

Input	Decrement	Output
23:30:15	30	23:00:15
17:07:12	30	16:37:12

Subtract seconds

This operation subtracts a decrement value from the seconds.

Input	Decrement	Output
23:30:15	15	23:30:00
17:07:12	15	17:06:57

Time difference (in ms)

This operation shows the time difference in milliseconds between the specified value and those from the selected field.

Input	Value	Output
21:05:17	12:00:00	32717000
08:35:40	12:00:00	-12260000

To milliseconds

This operation calculates the time value into milliseconds.

Input	Output
07:30:10	27010000
00:30:10	1810000

To seconds

This operation calculates the time value into seconds.

Input	Output
07:30:10	27010
00:30:10	1810

To string format

This operation transforms the time into a specified String format. Use "HH" for hours, "mm" for minutes, and "ss" for seconds. Specify a punctuation to connect the string.

Input	String format	Output
07:30:10	HH.mm.ss	07.30.10
00:31:41	HH.mm.ss	00.31.41

Timestamp Transform

Data in Timestamp type can be converted to Decimal, Integer or String type. You can perform the following operations:

Add days

This operation adds an increment value to the day.

Input	Increment	Output
2009-12-27 00:30:10	5	2010-01-01 00:30:10

Input	Increment	Output
2009-12-14 19:15:20	5	2009-12-19 19:15:20

Add hours

This operation adds an increment value to the hour.

Input	Increment	Output
2011-04-07 23:30:15	1	2011-04-08 00:30:15
2010-03-28 17:07:12	1	2010-03-28 18:07:12

Add milliseconds

This operation adds an increment value to the millisecond.

Input	Increment	Output
2009-04-07 23:30:15	10000	2009-04-07 23:30:25
2010-12-10 17:07:12	10000	2010-12-10 17:07:22

Add minutes

This operation adds an increment value to the minute.

Input	Increment	Output
2009-04-07 23:30:15	30	2009-04-08 00:00:15
2010-12-10 17:07:12	30	2010-12-10 17:37:12

Add months

This operation adds an increment value to the month.

Input	Increment	Output
2009-05-14 09:45:10	2	2009-07-14 09:45:10
2009-12-14 16:20:00	2	2010-02-14 16:20:00

Add seconds

This operation adds an increment value to the second.

Input	Increment	Output
2009-03-28 23:30:15	45	2009-03-28 23:31:00
2011-04-07 17:07:12	45	2011-04-07 17:07:57

Add years

This operation adds an increment value to the year.

Input	Increment	Output
2009-12-14 03:05:10	1	2010-12-14 03:05:10

Input	Increment	Output
2011-07-01 19:30:00	1	2012-07-01 19:30:00

Am

This operation transforms the time into Boolean type. If the timestamp value is before noon, it is changed to "true", otherwise it is changed to "false".

Input	Output
2010-12-30 16:51:02	false
2011-10-09 11:29:46	true

Day

This operation shows only the day from the selected field.

Input	Output
2009-12-14 16:30:00	14
2011-06-23 00:20:10	23

First of month

This operation changes the day value to the first day of each month.

Input	Output
2010-09-09 10:15:00	2010-09-01 10:15:00
2008-05-24 10:15:00	2008-05-01 10:15:00

Hours

This operation shows only the hours from the selected field.

Input	Output
2009-11-25 09:28:07	9
2011-01-01 17:56:39	17

Last of month

This operation changes the day value to the last day of each month.

Input	Output
2010-09-09 10:15:00	2010-09-30 10:15:00
2008-05-24 10:15:00	2008-05-31 10:15:00

Max

This operation changes each value in the selected field to a specified maximum value. If the original value is less than the specified maximum, the result becomes the maximum, otherwise it is unchanged.

Input	Maximum	Output
2011-04-05 07:30:10	2011-01-01 12:00:00	2011-04-05 07:30:10
2010-01-01 16:00:00	2011-01-01 12:00:00	2011-01-01 12:00:00

Milliseconds

This operation shows only the milliseconds from the selected field.

Input	Output
2010-09-09 00:30:10	0
2008-05-31 07:31:41	0

Min

This operation changes each value in the selected field to a specified minimum value. If the original value is greater than the specified minimum, the result becomes the minimum, otherwise it is unchanged.

Input	Maximum	Output
2011-04-05 07:30:10	2011-01-01 12:00:00	2011-01-01 12:00:00
2010-01-01 16:00:00	2011-01-01 12:00:00	2010-01-01 16:00:00

Minutes

This operation shows only the minutes from the selected field.

Input	Output
2010-05-06 09:28:07	28
2009-07-16 17:56:39	56

Month

This operation shows only the month from the selected field.

Input	Output
2009-12-14 07:30:10	12
2011-06-23 21:01:07	6

Month name (long)

This operation shows the month value from the selected field in a long name.

Input	Output
2009-12-14 07:30:10	December
2011-06-23 00:15:05	June

Month name (short)

This operation shows the month value from the selected field in a short name.

Input	Output
2009-12-14 07:30:10	Dec
2011-06-23 00:15:05	Jun

Null to timestamp

This operation displays the Null timestamp as a specified value, while the other values remain the same.

Input	Value	Output
	2000-01-01 00:00:00	2000-01-01 00:00:00
2009-01-01 07:30:10	2000-01-01 00:00:00	2009-01-01 07:30:10

Pm

This operation transforms the timestamp into Boolean type. If the timestamp value is between noon and midnight, it is changed to "true", otherwise changed to "false".

Input	Output
2010-12-31 16:51:02	true
2011-09-20 11:29:46	false

Quarter

This operation returns an integer to show the quarter to which the specified timestamp value belongs.

Input	Output
2010-11-16 12:30:10	4
2011-03-28 21:00:05	1
2011-05-01 17:12:19	2
2011-08-20 00:30:00	3

Quarter name (short)

This operation returns a string to show the quarter to which the specified timestamp value belongs.

Input	Output
2010-11-16 12:30:10	Q4
2011-03-28 21:00:05	Q1
2011-05-01 17:12:19	Q2
2011-08-20 00:30:00	Q3

Seconds

This operation shows only the seconds from the selected field.

Input	Output
2011-03-28 09:28:07	7
2009-07-16 17:56:39	39

Set day

This operation enables you to set the day value in the selected field.

Input	Day	Output
2010-01-01 07:30:10	5	2010-01-05 07:30:10
2008-01-01 21:10:00	5	2008-01-05 21:10:00

Set hour

This operation enables you to set the hour value in the selected field.

Input	Hour	Output
2011-12-09 00:30:10	20	2011-12-09 20:30:10
2009-03-15 07:31:41	20	2009-03-15 20:31:41

Set millisecond

This operation enables you to set the millisecond value in the selected field. If the specified value is not big enough to calculate into seconds, minutes or hours, the result may look the same with the original value. However, the hidden millisecond value can affect comparing and sorting.

Input	Value	Output
2011-12-09 00:30:10	1	2011-12-09 00:30:10
2009-03-15 07:31:41	1	2009-03-15 07:31:41

Set minute

This operation enables you to set the minute value in the selected field.

Input	Value	Output
2011-12-09 00:30:10	1	2011-12-09 00:01:10
2009-03-15 07:31:41	1	2009-03-15 07:01:41

Set month

This operation enables you to set the month value in the selected field.

Input	Month	Output
2011-01-01 14:00:00	1	2011-01-01 14:00:00
2009-07-17 23:30:00	1	2009-01-17 23:30:00

Set second

This operation enables you to set the second value in the selected field.

Input	Value	Output
2011-12-09 00:30:10	1	2011-12-09 00:30:01
2009-03-15 07:31:41	1	2009-03-15 07:31:01

Set year

This operation enables you to set the year value in the selected field.

Input	Year	Output
2011-01-01 12:30:00	2011	2011-01-01 12:30:00
2008-07-17 19:00:00	2011	2011-07-17 19:00:00

Subtract days

This operation subtracts a decrement value from the day.

Input	Decrement	Output
2009-12-27 16:30:00	5	2009-12-22 16:30:00
2009-12-05 05:30:00	5	2009-11-30 05:30:00

Subtract hours

This operation subtracts a decrement value from the hour.

Input	Decrement	Output
2009-04-26 23:30:15	5	2009-04-26 18:30:15
2011-10-02 17:07:12	5	2011-10-02 12:07:12

Subtract milliseconds

This operation subtracts a decrement value from the milliseconds. If the decrement value is not big enough to affect seconds, minutes or hours, the result may look the same with the original value. However, the hidden millisecond value can affect comparing and sorting.

Input	Decrement	Output
2009-04-26 23:30:15	30	2009-04-26 23:30:15
2011-10-02 17:07:12	30	2011-10-02 17:07:12

Subtract minutes

This operation subtracts a decrement value from the minutes.

Input	Decrement	Output
2009-04-26 23:30:15	30	2009-04-26 23:00:15
2011-10-02 17:07:12	30	2011-10-02 16:37:12

Subtract months

This operation subtracts a decrement value from the month.

Input	Decrement	Output
2009-12-14 22:30:20	1	2009-11-14 22:30:20
2009-01-22 08:15:07	1	2008-12-22 08:15:07

Subtract seconds

This operation subtracts a decrement value from the seconds.

Input	Decrement	Output
2011-03-28 23:30:15	15	2011-03-28 23:30:00
2010-12-12 16:00:00	15	2010-12-12 15:59:45

Subtract years

This operation subtracts a decrement value from the year.

Input	Decrement	Output
2009-12-27 00:30:10	2	2007-12-27 00:30:10
2011-02-05 16:20:40	2	2009-02-05 16:20:40

Timestamp difference (in days)

This operation shows the timestamp difference in days between the specified value and those from the selected field.

When you use Timestamp difference (in days) on two timestamp values, the time will be ignored. Therefore, "2010-01-02 00:01:00" and "2010-01-01 23:59:00" are only two minutes apart, but the difference in days is 1.

Input	Value	Output
2011-01-01 18:00:00	2011-01-01 12:00:00	0
2011-01-02 14:00:00	2011-01-01 12:00:00	1

Timestamp difference (in ms)

This operation shows the timestamp difference in milliseconds between the specified value and those from the selected field.

Input	Value	Output
2011-01-01 18:00:00	2011-01-01 12:00:00	21600000
2011-01-02 14:00:00	2011-01-01 12:00:00	93600000

To milliseconds

This operation calculates the timestamp value into milliseconds since the start of the day.

Input	Output
2010-10-15 12:30:00	45000000
2008-05-24 10:15:00	36900000

To seconds

This operation calculates the timestamp value into seconds since the start of the day.

Input	Output
2010-10-15 12:30:00	45000
2008-05-24 10:15:00	36900

Truncate

This operation changes the hour, minute and second values to zero.

Input	Output
2009-12-27 01:10:55	2009-12-27 00:00:00
2009-01-05 17:59:30	2009-01-05 00:00:00

Year

This operation shows only the year value from the selected field.

Input	Output
2009-12-14 07:10:30	2009
2011-06-23 14:32:05	2011

Chapter 5
Compare, Sequence and Sort Transforms

Compare Transform

If you choose the Compare type in a transform process, it compares the values from a specified field with a specified value, and returns Boolean values ("true" or "false"). You can compare values by performing the following operations:

Equal

This operation returns "true" for values that are equal to the specified value, and returns "false" otherwise.

Input	Value	Output
A	A	true
D	A	false

Less than

This operation returns "true" for values that are less than the specified value, and returns "false" otherwise.

Input	Value	Output
A	B	true
D	B	false

Less than or equal

This operation returns "true" for values that are less than or equal to the specified value, and returns "false" otherwise.

Input	Value	Output
A	D	true
D	D	true

More than

This operation returns "true" for values that are more than the specified value, and returns "false" otherwise.

Input	Value	Output
A	B	false

Input	Value	Output
D	B	true

More than or equal

This operation returns "true" for values that are more than or equal to the specified value, and returns "false" otherwise.

Input	Value	Output
A	A	true
D	A	true

Next equal

This operation returns "true" for values whose next value is equal to them, and returns "false" otherwise.

Input	Output
A	false
D	true
D	false

Next less than

This operation returns "true" for values whose next value is less than them, and returns "false" otherwise.

Input	Output
1	false
4	false
4	true
3	true

Next less than or equal

This operation returns "true" for values whose next value is less than or equal to them, and returns "false" otherwise.

Input	Output
1	false
4	true
4	true
3	true

Next more than

This operation returns "true" for values whose next value is more than them, and returns "false" otherwise.

Input	Output
1	true

Input	Output
4	false
4	false
3	false

Next more than or equal

This operation returns "true" for values whose next value is more than or equal to them, and returns "false" otherwise.

Input	Output
1	true
4	true
4	false
3	false

Next not equal

This operation returns "true" for values whose next value is not equal to them, and returns "false" otherwise.

Input	Output
A	true
D	false
D	true

Not equal

This operation returns "true" for values that are not equal to the specified value, and returns "false" otherwise.

Input	Value	Output
A	D	true
D	D	false

Previous equal

This operation returns "true" for values whose previous value is equal to them, and returns "false" otherwise.

Input	Output
A	false
D	false
D	true

Previous less than

This operation returns "true" for values whose previous value is less than them, and returns "false" otherwise.

Input	Output
1	true
4	true
4	false
3	false

Previous less than or equal

This operation returns "true" for values whose previous value is less than or equal to them, and returns "false" otherwise.

Input	Output
1	true
4	true
4	true
3	false

Previous more than

This operation returns "true" for values whose previous value is more than them, and returns "false" otherwise.

Input	Output
1	false
4	false
4	false
3	true

Previous more than or equal

This operation returns "true" for values whose previous value is more than or equal to them, and returns "false" otherwise.

Input	Output
1	false
4	false
4	true
3	true

Previous not equal

This operation returns "true" for values whose previous value is not equal to them, and returns "false" otherwise.

Input	Output
A	true
D	true
D	false

Sequence Transform

If you choose the Sequence type in a transform process and specify a field to group on, a new field of data will display in the result. You can sequence the original data by performing the following operations:

Per group

This operation creates a sequence according to the groups, disregarding the records.

The following table shows an example of the input and output (Start: 1, Step: 1):

Input	Output
A	1
A	1
B	2
C	3
C	3

Per record

This operation creates a sequence according to the records, disregarding the groups.

The following table shows an example of the input and output (Start: 1, Step: 1):

Input	Output
A	1
A	2
B	3
C	4
C	5

Record per group

This operation creates a sequence according to the records from each group.

The following table shows an example of the input and output (Start: 1, Step: 1):

Input	Output
A	1
A	2
B	1
C	1
C	2

Round robin

This operation creates a round robin sequence according to the records, disregarding the groups.

The following table shows an example of the input and output (Start: 1, Count: 2):

Input	Output
A	1
A	2
B	1
C	2
C	1

Top

This operation enables you to specify a limit and creates a Boolean field as the sequence. It returns "true" from the top to the limited row, and returns "false" afterwards.

Input	Limit	Output
A	3	true
B	3	true
C	3	true
D	3	false
E	3	false

Sort Transform

If you choose the Sort type in a transform process, one or more selected fields of data will be sorted. The result may vary according to the order of your selection. The upper field has a priority over the lower field in the Sort Transform.

The following table shows an example of the input:

Field 1	Field 2
4	A
2	C
1	D
3	B

Select Field 2, and select the Ascending order. Alternatively, you can select Field 1, and select the Descending order. The following table shows the output:

Field 1	Field 2
4	A
3	B
2	C
1	D

Chapter 6

Constant, Discard, Discard Duplicates, Reduce, Retain and Security Transforms

Constant Transform

If you choose the Constant type in a transform process, a new field of the specified constant value will be created.

The following table shows an example of the input:

Product	Price before tax
A	20.30
B	15.99
C	50.29

The following table shows an example of the output (Field: Tax, Data Type: String, Value: 7%):

Product	Price before tax	Tax
A	20.30	7%
B	15.99	7%
C	50.29	7%

Constant Transform can be used to mask fields. Use the name of an existing field as the name of the new field, and specify a constant value, for example, "Confidential". Values in that field will be reset to "Confidential".

Discard Transform

If you choose the Discard type in a transform process, selected columns will not display in the result.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Selected columns: Field 2 and Field 3):

Field 1	Field 4	Field 5
A	10	5.1
A	10	7.1
A	10	9.1
D	0	1.2
D	0	3.2

Discard Duplicates Transform

The Discard Duplicates transform enables you to select one or more fields to compare, and discards duplicated rows depending on values in these fields. Before starting this transform, make sure the fields have been sorted first, to minimize memory use. This is because comparing each record with many other unsorted records requires a significantly large memory if there are huge volumes of data. However, if we only have to compare each record with the previous record, we can run through massive amounts of data without needing huge amounts of memory.

The following table shows an example of the input:

Field 1	Field 2
A	5
A	9
A	9
D	1
D	3

If you select Field 1 to compare, the following table shows the output:

Field 1	Field 2
A	5
D	1

If you select Field 2 to compare, the following table shows the output:

Field 1	Field 2
A	5
A	9
D	1
D	3

Reduce Transform

If you choose the Reduce type in a transform process, you can specify the following operations (or actions) on one or more fields of data. It reduces a set of records down to one summary record, which means corresponding rows will be merged into one row. If there are irrelevant columns which might affect the merge, use Retain Transform to keep the columns you need.

Before starting this transform, make sure the fields have been sorted first, which minimizes memory use. This is because comparing each record with many other unsorted records requires a significantly large memory if there are huge volumes of data. However, if we only have to compare each record

with the previous record, we can run through massive amounts of data without needing huge amounts of memory.

The following table shows an example of the input shared by all the operations under Reduce Transform:

Table 6.1. Shared input of Reduce Transform

Field 1	Field 2	Field 3	Field 4
A	B	2.0	2.9
A	B	1.0	1.9
A	B	3.0	3.9
D	E	5.0	5.9
D	E	4.0	4.9

Average

This operation calculates the values from the specified fields into the average.

If you choose the **Average** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	2.0	2.9
D	E	4.5	5.4

Comma Separated List

This operation merges the values from the specified fields into an unordered list separated by comma.

If you choose the **Comma Separated List** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	2.0,1.0,3.0	2.9,1.9,3.9
D	E	5.0,4.0	5.9,4.9

Comma Separated Set

This operation merges the values from the specified fields into an ordered list separated by comma.

If you choose the **Comma Separated Set** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	1.0,2.0,3.0	1.9,2.9,3.9
D	E	4.0,5.0	4.9,5.9

Count

This operation counts the number of values that are present in the specified fields.

If you choose the **Count** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	3	3
D	E	2	2

First

This operation extracts the first value from the specified fields.

If you choose the **First** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	2.0	2.9
D	E	5.0	5.9

Last

This operation extracts the last value from the specified fields.

If you choose the **Last** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	3.0	3.9
D	E	4.0	4.9

Max

This operation merges the values from the specified fields into the maximum.

If you choose the **Max** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	3.0	3.9
D	E	5.0	5.9

Median

This operation merges the values from the specified fields into the median value. If there are only two values in a field, it returns the average directly.

If you choose the **Median** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	2.0	2.9
D	E	4.5	5.4

Min

This operation merges the values from the specified fields into the minimum.

If you choose the **Min** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	1.0	1.9
D	E	4.0	4.9

Product

This operation calculates the values from the specified fields into the product.

If you choose the **Product** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	6.000	21.489
D	E	20.00	28.91

Standard Deviation

This operation calculates the values from the specified fields into the standard deviation.

If you choose the **Standard Deviation** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	1	1
D	E	0.7071*	0.7071*

*This is an approximate value of the output for the purpose of illustration.

Sum

This operation calculates the values from the specified fields into the summary.

If you choose the **Sum** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	6.0	8.7
D	E	9.0	10.8

Variance

This operation calculates the values from the specified fields into the variance.

If you choose the **Variance** action for Field 3 and Field 4, the following table shows the output of Table 6.1, "Shared input of Reduce Transform":

Field 1	Field 2	Field 3	Field 4
A	B	1.00	1.00
D	E	0.50	0.50

Retain Transform

If you choose the Retain type in a transform process, only the selected columns will be present in the result. This is important for Reduce Transform, which requires corresponding rows to have the same values in order to be grouped together. Use Retain Transform in conjunction with Reduce Transform to keep only the relevant columns and make sure the grouping can function.

The following table shows an example of the input:

Field 1	Field 2	Field 3	Field 4	Field 5
A	B	C	10	5.1
A	B	C	10	7.1
A	B	C	10	9.1
D	E	F	0	1.2
D	E	F	0	3.2

The following table shows an example of the output (Selected columns: Field 1 and Field 5):

Field 1	Field 5
A	5.1
A	7.1
A	9.1
D	1.2
D	3.2

Security Transform

If you choose the Security type in a transform process, you can specify which users and groups will be able to access data from the selected columns, the selected rows where the boolean column value is true, the rows where the field value matches user credential, or the entire table.

The following table shows an example of the input, which will be used in the following types: Column, Row, Row Credentials and Table.

Table 6.2. Shared input of Security Transform

Department	Employee	Salary
management	ABC	12000
sales	DEF	11000
sales	GHI	8000
marketing	JK	9000
marketing	LM	10500

Note

Users and groups should be created according to employee names and department names.

Column

This type enables you to choose which users and groups will be allowed to access the values in the column. For example, if you want to enable the management staff to view salaries, create a Security Transform and click **Next**. The Secure a table window opens. Select the **Column** type, the **Salary** column and the **management** group.

Users of the management group are able to view the Salary column. The output is the same with Table 6.2, "Shared input of Security Transform".

Users of the sales, marketing and other groups cannot view the Salary column. The following table shows the output:

Department	Employee	Salary
management	ABC	
sales	DEF	
sales	GHI	
marketing	JK	
marketing	LM	

Row

This type usually works with booleans, which you can build with any logic. Only authorized users and groups will be allowed to access the rows where the boolean column value is true. If you want to enable the sales staff to view salaries of only their own department, use Compare Transform first to create a new boolean column "CanSee". Select the **Department** field and the **Equal** operation. Type **CanSee** as the New Field and type **sales** as the Value.

In the Secure table window, select the **Row** type, the **CanSee** boolean column, and the **sales** group.

Users of the sales group can view the salaries of only their department. The following table shows the output of Table 6.2, "Shared input of Security Transform":

Department	Employee	Salary
sales	DEF	11000
sales	GHI	8000

Users of the other groups cannot view any contents, unless you have made further settings.

Row Credentials

This type compares the selected field values with user credentials. Only users and groups with a credential matching the field value will be able to access the row. If the field value is *****, it matches any credential, and any user can access the row. If the field value is a blank string, it does not match any credential, and no one can access the row.

Users and groups will not be distinguished in the credential list. There may be both a user and a group with the same name, for example, "sales". Users with other names can also have the "sales" credential if they are a member of the sales group. Each field value can contain at most one credential. If the value is "Elixir Sales", then it does not match any credential.

If you want to enable the employees to view salaries of only their own department, create a Security Transform and open the Secure a table window. Select the **Row Credentials** type and the **Department** column. Click **Finish**.

Each employee will be able to view salaries of their own department. For example, employees of the marketing department will see the following output of Table 6.2, "Shared input of Security Transform":

Department	Employee	Salary
marketing	JK	9000
marketing	LM	10500

Table

This type enables you to specify which users and groups will be able to access the records in the entire table.

If you want to enable the management group to access the table, create a Security Transform and open the Secure a table window. Select the **Table** type and select the **management** group.

Users in the management group will see all the records in the table, as shown in Table 6.2, "Shared input of Security Transform".

Users in other groups cannot see any contents, unless you have made further settings.