Custom Transformation - use cases with advanced SQL queries

Read more:

- Default automatic presets
- General syntax and SQL functions available

Using AI assistant

1. Switch the page to the edit mode
2. Insert the Table Transformer macro and paste the table/s or the macros outputting tables within the macro body
3. Select the macro and click **Edit**
4. Select **ChartGPT assistant** tab
5. Ask the assistant to write the SQL query according to your needs
6. Check the result via the Preview window
7. Modify the SQL query if needed
8. Save the macro

Find here how to enable/disable the AI assistant feature.

Using If / Then construct

Use case:

You have a table with the stationary orders. You need to output a new column in the table with the order priority according to the total sum of money: less than $1000 is Low, from $1000 to $2000 is Medium, and more than $2000 is High.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT
   *,
   CASE
      WHEN
         'Subtotal' < 1000
      THEN
         "LOW"
      WHEN
         'Subtotal' >= 1000
         AND 'Subtotal' < 2000
      THEN
         "MEDIUM"
      ELSE
         "HIGH"
   END
   AS 'Priority'
FROM
   T1
```

***CASE WHEN ... THEN ... ELSE ... END** goes through conditions and return a value when the first condition is met.*

***AS '...'** outputs a new 'Priority' column.*

6. Click **Next**.
7. Define the table settings and view options if needed.
8. **Save** the macro and the page.

If you want to replace the words Low, Medium and High by prominent statuses using the default Status macro or the Handy Status macro, just place a one-column table containing each status and the same column label in the macro body. Don't change anything in the SQL query.

You can use FORMATWIKI function to insert statuses into a table.

Comparing two dates using If / Then construct

Use case:

You have a table with two columns: Target date and Completion date. You need to compare and rate these dates:

- If the completion date is within 5 days, the status is ON TIME.
- If the completion date is more than 5 days late, but less than 10 days, the status is LATE.
- If the completion date is more than 10 days, the status is VERY LATE.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the tables or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT *,
   CASE
      WHEN
         (('Completion date' - 'Target
date') / "24h") < 5
      THEN "ON TIME"
      WHEN
         (('Completion date' - 'Target
date') / "24h") > 5
         AND
         (('Completion date' - 'Target
date') / "24h") <= 10
      THEN "LATE"
      ELSE "VERY LATE"
   END
```

```
   AS 'Rating'
FROM T1
```

6. Click **Next**.
7. Define the **date format** .
8. **Save** the macro and the page.

You can use FORMATWIKI function to insert statuses into a table.

Using formulas

Use case:

You have a table with the number of added lines of code and the number of defects. You need to calculate the quality of the code. The number of defects less than one defect per 10 lines indicates the good quality of the code. Otherwise the quality of the code is bad.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the tables or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT
   *,
   (
      'Number of defects' * 10 / 'Lines of
code added'
   )
   AS 'Defects per 10 lines',
   CASE
      WHEN
         (
            'Number of defects' * 10 /
'Lines of code added'
         )
         < 1
      THEN
         "Good"
      ELSE
         "Bad"
   END
   AS 'Code quality'
FROM
   T1
```

('...' * 10 / '...') AS '...' *calculates the number of defects per 10 lines of code and outputs the 'Defects per 10 lines' column.*

CASE WHEN ... THEN ... ELSE ... END AS '...' *goes through conditions and return a value when the first condition is met and outputs the 'Code quality' column.*

6. Click **Next**.
7. Define the table settings and view options if needed.
8. **Save** the macro and the page.

You can use FORMATWIKI function for the purposes of cell formatting.

You can merge two columns containing text with the help of formulas:

```
SELECT
   *,
   (
      'First name' + " " + 'Last name'
   )
   AS 'Full name'
FROM
   T1
```

Merging tables using filtration

Use case:

You have two tables where the columns differ.

You need to merge two tables where the entrys meet the conditions:

- transaction date in Table 1: October 2018
- customer type in Table 2: business

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the tables or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT
    *
FROM
    T1
```

```
   JOIN
      T2
      ON T1.'Transaction ID' =
T2.'Transaction ID'
WHERE
   (
      'Transaction Date' >= "10 / 1 / 2018"
      AND 'Transaction Date' < "11 / 1 /
2018"
   )
   AND
   (
      'Customer Type' = "Business"
   )
```

WHERE ('...' >= "10/1/2018" AND '...' < "11/1/2018") AND ('...' = "...") *extracts only those records that fulfill a specified condition.*

6. Click **Next**.
7. Define the table settings and view options if needed.
8. **Save** the macro and the page.

You can use FORMATWIKI function for the purposes of cell formatting.

Calculating the number of workdays in a period of time

Use case:

You have a table containing a period of time (start date and end date).

You need to count the number of workdays during this time period.

Advanced use case:

You have an additional table with national holidays which you also shouldn't consider as workdays.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table within the macro body.
3. Select the macro and click **Edit**.
4. Switch to the **SQL query** tab.
5. Enter the following SQL query:

```
SELECT *,
'Days' -
2 * (('Days' / 7)::integer) -
```

```
CASE
        WHEN 'Days' % 7 = 0 THEN 0
        WHEN 'Start date'::Date->getDay() =
0 THEN 1
        WHEN 'Start date'::Date->getDay() +
'Days' % 7 = 7 THEN 1
        WHEN 'Start date'::Date->getDay() +
'Days' % 7 > 7 THEN 2
        ELSE 0
END AS 'Work days'
FROM
(SELECT *,
ROUND((T1.'End date' - T1.'Start date') /
"1d") + 1 AS 'Days'
FROM T1)
```

6. Click **Next**.
7. Set the worklog settings.
8. **Save** the macro and the page.

> You can use FORMATWIKI function for the purposes of cell formatting.

Solution for the advanced use case:

Enter the following SQL query:

```
SELECT *,
'Days' - 2 * (('Days' / 7)::integer) -
CASE
        WHEN 'Days' % 7 = 0 THEN 0
        WHEN 'Start date'::Date->getDay() = 0 THEN
1
        WHEN 'Start date'::Date->getDay() + 'Days'
% 7 = 7 THEN 1
```

```
        WHEN 'Start date'::Date->getDay() + 'Days'
% 7 > 7 THEN 2
        ELSE 0
END
-
(SELECT COUNT(*) FROM
(SELECT *,'Holiday'::Date->getDay() AS 'Day' FROM
T2)
WHERE 'Holiday' >= TT.'Start date' AND 'Holiday'
<= TT.'End date'
AND 'Day' > 0 AND 'Day' < 6)
AS 'Work days'
FROM
(SELECT *, ROUND(('End date' - 'Start date') /
"1d") + 1 AS 'Days' FROM T1) AS TT
```

Calculating cumulative (running) totals

Use case:

You have a table (a pivot table) containing a sequence of years and some numeric values.

You need to calculate cumulative (running) totals when the value of each next year is added to the sum of the previous ones.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table within the macro body.
3. Select the macro and click **Edit**.
4. Switch to the **SQL query** tab.
5. Enter the following SQL query:

```
SELECT 'Year',
SUM (TT2.'Requests') AS 'Requests'
FROM T1 AS TT1
JOIN T1 AS TT2 on TT1.'Year' >= TT2.'Year'
GROUP BY TT1.'Year'
ORDER BY TT1.'Year'
```

6. Click **Next**.
7. Define the table settings and view options if needed.
8. **Save** the macro and the page.

As an alternative you can enable the **Cumulative count** option in the Pivot Table macro.

You can use FORMATWIKI function for the purposes of cell formatting.

Calculating standard deviation / variance

Use case:

You have a table (a macro outputting a table) containing a set of numeric values.

You need to calculate the standard deviation or variance.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table within the macro body.
3. Select the macro and click **Edit**.
4. Switch to the **SQL query** tab.
5. Enter the following SQL query (select the functions depending on what you'd like to calculate):

```
SELECT
MEDIAN(T1.'Value') AS 'Median',
STDDEV(T1.'Value') AS 'Standard deviation
population',
STDEV(T1.'Value') AS 'Standard deviation
sample',
VAR(T1.'Value') AS 'Variance sample',
VARP(T1.'Value') AS 'Variance population'
FROM T1
```

6. Click **Next**.
7. Define the table settings and view options if needed.
8. **Save** the macro and the page.

You can use FORMATWIKI function for the purposes of cell formatting.

Automatic filling blanks and updating values in tables

Use case:

You have two tables containing information about employees on different pages.

The first table is an Excel spreadsheet with data.

You need to fill in the blanks or update values in the second table with data from the first one by matching the 'Employee' columns.

Advanced use case:

You need to select values manually (without column matching) and add t hem into empty cells only.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the tables or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Ne xt**.

5. Enter the following SQL query:

```
SELECT T1.'Number', T1.'Employee',
T1.'Position',
 CASE
  WHEN T2.'City' IS NULL
  THEN T1.'Location'
  ELSE T2.'City'
 END
 AS 'Location'
FROM T1 LEFT JOIN T2 ON T1.'Employee'-
>toLowerCase() = T2.'Employee'toLowerCase()
```

6. Click **Next**
7. Define the table settings and view options if needed
8. **Save** the macro and the page.

You can use FORMATWIKI function for the purposes of cell formatting.

Solution for the advanced use case:

Enter the following SQL query:

```
SELECT T1.'Number',T1.'Employee',T1.'Position',
 CASE
  WHEN T1.'Employee' = "John Anderson" AND
T1.'Location' IS NULL
  THEN (SELECT T2.'City' FROM T2 WHERE
T2.'Employee' = "John Anderson")
  WHEN T1.'Employee' = "Alban Jacobs" AND
T1.'Location' IS NULL
  THEN (SELECT T2.'City' FROM T2 WHERE
T2.'Employee' = "Alban Jacobs")
  ELSE T1.'Location'
 END
 AS 'Location'
FROM T1
```

Calculating the number of distinct (unique) values by categories

Use case:

You have a table (a macro outputting a table) containing multiple columns with text data.

You need to calculate the number of distinct (unique) values by categories. For example, the number of unique customers in every segment.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table within the macro body.
3. Select the macro and click **Edit**.
4. Switch to the **SQL query** tab.
5. Enter the following SQL query (select the functions depending on what you'd like to calculate):

```
SELECT T1.'Segment',
COUNT (DISTINCT (T1.'Customer Name')) AS
'Number of customers'
FROM T*
GROUP BY T1.'Segment'
```

6. Click **Next**.
7. Define the table settings and view options if needed.
8. **Save** the macro and the page.

You can use FORMATWIKI function for the purposes of cell formatting.

Number formatting

Use case:

You have a table or a macro outputting a table.

You need to change the data format of the column containing numbers to add a dollar sign, a thousands separator into values and remove decimals.

Solution:

1. Switch the page to the edit mode.

2. Insert the Table Transformer macro and paste the table within the macro body.
3. Select the macro and click **Edit**.
4. Switch to the **SQL query** tab.
5. Enter the following SQL query:

```
SELECT 'Company',
"$ "+FORMATNUMBER('Profit per month') as
'Profit'
FROM T*
```

6. Click **Next**.
7. Define the table settings and view options.
8. **Save** the macro and the page.

FORMATNUMBER function allows you to format numbers according to the settings in the Table Transformer macro.

You can use FORMATWIKI function for the purposes of cell formatting.

Searching for a specified pattern in a column

Use case:

You need to find any values of the 'Customer Name' column that have "é" in any position.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table within the macro body.
3. Select the macro and click **Edit**.

4. Switch to the **SQL query** tab.
5. Enter the following SQL query:

```
SELECT *
FROM T1
WHERE 'Customer Name' LIKE "%é%"
```

6. Click **Next**.
7. **Save** the macro and the page.

LIKE "a%" - Finds any values that start with "a"
LIKE "%a" - Finds any values that end with "a"
LIKE "_r%" - Finds any values that have "r" in the second position
LIKE "a_%" - Finds any values that start with "a" and are at least 2 characters in length
LIKE "a__%" - Finds any values that start with "a" and are at least 3 characters in length
LIKE "B%s" - Finds any values that start with "B" and ends with "s"

You can use FORMATWIKI function for the purposes of cell formatting.

Using regular expression

Use case:

You need to find the first four values in the range from 0 to 8 in an 'ID' column.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table within the macro body.
3. Select the macro and click **Edit**.
4. Switch to the **SQL query** tab.
5. Enter the following SQL query:

```
SELECT
'ID'->match("^[0-8]{4}") AS 'Year', 'Brand'
FROM T1
```

6. Click **Next**.
7. **Save** the macro and the page.

The [0-9] expression is used to find any character between the brackets.
Use the [^0-9] expression to find any character that is NOT a digit.

[aeiou] matches any one of the charachers within the the square bracket.
[a-zA-Z] matches any uppercase or lowercase letters.

^ matches the beginning of the input string,
{4} means the number of characters to display

Here you can find more info about JavaScript regular expressions.

You can use FORMATWIKI function for the purposes of cell formatting.

Getting local time of different time zones

Use case:

You need to get time based on the time zones.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT *,
FORMATDATE(DATEADD(hour, T1.'GMT' +
(T1.'Created'::Date->getTimezoneOffset() /
60), T1.'Created'))
AS 'My Time',
FORMATDATE(DATEADD(hour, T1.'GMT' +
("today"::Date->getTimezoneOffset() / 60),
NOW()))
AS 'Current Time'
FROM T1
```

6. Click **Next**.

7. Define the **date format** .
8. **Save** the macro and the page.

The getTimezoneOffset() method returns the time difference between UTC time and local time, in minutes.

The NOW() function returns the current date and time.

The DATEADD() function adds a time/date interval to a date and then returns the date.

The FORMATDATE() function converts time/dates to the specified in the Table Transformer macro settings date format.

Changing a link text and preserving it clickable

Use case:

You need to change a text for links and preserve it clickable.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT T1.'Company',
FORMATWIKI( "[Click here|" +
 T1.'Link'->tfView->match("href=([^ >]*)")-
>1->slice(1, -1)
 + "]"
 ) AS 'Link'
FROM T*
```

6. Click **Next**.
7. **Save** the macro and the page.

Creating clickable links to the Jira issues

Use case:

You need to count the number of issues by status and make reference to JIRA.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT 'Status',
 FORMATWIKI("["+ COUNT(*) +"|" +
                  "https://jira.stiltsoft.
com/issues/?jql=issuekey%20in%20(" +
                  ARRAY(T1.'Key') + ")" +
"]")
AS 'COUNT'
FROM T1 GROUP BY T1.'Status'
```

```
UNION ALL CORRESPONDING
SELECT "Total" AS 'Status',
FORMATWIKI("["+ COUNT(*) +"|" +
                  "https://jira.stiltsoft.
com/issues/?jql=issuekey%20in%20(" +
                  ARRAY(T1.'Key') + ")" +
"]") AS 'COUNT'
FROM T1
```

6. Click **Next**.
7. Define the **date format** .
8. **Save** the macro and the page.

Grouping Jira issues

Use case:

You need to group Jira linked issues by types and preserve clickable links to Jira.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT
FIRST(T1.'Linked Issues'->split("-")->0) AS
'Type',
FORMATWIKI(SUM("[" + T1.'Linked Issues' +
"|https://jsd.stiltsoft.com/browse/" +
T1.'Linked Issues' + "]\\ ")) AS 'Issues'
FROM T1
GROUP BY
T1.'Linked Issues'->split("-")->0
```

6. Click **Next**.
7. Define the **date format** .
8. **Save** the macro and the page.

Merging Jira Issue tables by linked issues and keys

Use case:

You need to merge two Jira Issue tables by linked issues (a comma separated list) from one table and issue keys from another.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT T1.'T',T1.'Key',T1.'Summary',
CONCAT_VIEW_AGGR(FORMATWIKI(T2.'T' , " \n"))
AS 'Type - linked issues',
CONCAT_VIEW_AGGR(FORMATWIKI(T2.'Key', "
\n")) AS 'Key - linked issues',
CONCAT_VIEW_AGGR(FORMATWIKI(T2.'Summary', "
\n")) AS 'Summary - linked issues',
CONCAT_VIEW_AGGR(FORMATWIKI(T2.'Status', "
\n")) AS 'Status - linked issues'
FROM T1 LEFT JOIN T2 ON T1.'Linked Issues'-
>split(" , ")->indexOf(T2.'Key'::string) > -1
GROUP BY T1.'Key', T1.'T',T1.'Summary'
```

6. Click **Next**.
7. Define the **date format** .
8. **Save** the macro and the page.

A Javascript method in joing tables is used for splitting a comma-separated list of linked issues.

The CONCAT_VIEW_AGGR funtion is used for concatenation and aggregation table data preserving their original formatting.

The FORMATWIKI funtion is used for adding line breaks.

The GROUP BY statement groups rows that have the same values into summary rows.

Formatting a header of Jira Issue

Use case:

You need to apply custom formatting to a header of the Jira Issue table.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro, create a table containing only the Jira Issue table header formatted as you wish, and paste it along with the Jira Issue macro in the macro body.

3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Merge tables** and click **Next**.
5. **Save** the macro and the page.

Counting the definite value in a column

Use case:

You need to count the definite value in columns.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT "Red mark" AS '',
SUM(IF(T1.'Time' = "RED", 1, 0)) AS 'Time',
SUM(IF(T1.'Quality' = "RED", 1, 0)) AS
'Quality',
SUM(IF(T1.'Customer Satisfaction' = "RED",
1, 0)) AS 'Customer Satisfaction'
FROM T1
```

6. Click **Next**.
7. **Save** the macro and the page.

Calculating the completion ratio

Use case:

You need to calculate the completion ratio based on the start, end dates of project phases.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT *,
CASE WHEN 'Completion Ratio'>100
THEN "100%"
WHEN 'Completion Ratio'<0
THEN "0%"
ELSE ROUND('Completion Ratio')+"%"
END  AS 'Completion Ratio'
FROM
(SELECT *,
DATEDIFF(DAY,'Start Date',"today")/
DATEDIFF(DAY, 'Start Date', 'End Date')*100
AS 'Completion Ratio'
FROM T1)
```

6. Click **Next**.
7. **Save** the macro and the page.

Accounting for the monthly budget

Use case:

You need to calculate the difference between monthly budget and actual expenses by categories and display the appropriate status: "Under budget", "Over budget", "On budget".

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT *,
(T1.'Actual'- T1.'Budget') AS 'Difference,
$',
(T1.'Actual' / T1.'Budget' * 100) AS '%
Budget',
CASE
WHEN (T1.'Actual' / T1.'Budget' * 100) < 100
THEN FORMATWIKI("{status:
colour=Green|title=Under budget}")
WHEN (T1.'Actual' / T1.'Budget' * 100) > 100
THEN FORMATWIKI("{status:
colour=Red|title=Over budget}")
ELSE FORMATWIKI("{status:
colour=Yellow|title=On budget}")
END AS 'Indicator'
FROM T*
```

6. Click **Next**.
7. **Save** the macro and the page.

Merging tables by partial match

Use case:

You need to look up tables by partial match.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.

3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT *
FROM T1 LEFT JOIN T2 ON
T1.'Codes'->split("KR")->indexOf(REPLACE
(T2.'Code', "KR", ""))>-1
```

6. Click **Next**.
7. **Save** the macro and the page.

Pivoting tables

Use case:

You need to PIVOT table data in Table Transformer.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT * FROM T1
PIVOT (MAX('Value') for 'Month')
```

6. Click **Next**.
7. **Save** the macro and the page.

The aggregation function **MAX** is applied to an existing column *Value*. The *Month* column is the column the data are aggregated by.

The same output can be produced with the Pivot Table macro with versatile functions like **MAX**, **MIN**, **SUM**, etc., specifically designed for easy and accessible table data aggregation.

Unpivoting tables

Use case:

You need to UNPIVOT table data in Table Transformer.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT * FROM T1
UNPIVOT ('Value' for 'Month' in (T1.'01-2021', T1.'02-2021', T1.'03-2021'))
```

6. Click **Next**.
7. **Save** the macro and the page.

The **UNPIVOT** function is applied to the selection above and works with the table generated from it. In the function itself new custom columns - *Value* and *Month* - are created (column names can be adjusted). The *Value* column is filled with the data from the columns listed after **in** ('01-2021', '02-2021', '03-2021'). The *Month* column displays the relevant column names listed.

Aggregating by headers

Use case:

You are to aggregate headers from the Dev-stage 1 to Dev-stage 5 columns by their values and products.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT 'Product', 'Value', ARRAY('Dev-
stage') AS 'Dev-stage'
 FROM ( SELECT * FROM T1 UNPIVOT ('Value'
for 'Dev-stage'
in (T1.'Dev-stage 1',T1.'Dev-stage 2',
T1.'Dev-stage 3',T1.'Dev-stage 4',T1.'Dev-
stage 5')))
 GROUP BY 'Product', 'Value'
```

6. Click **Next** and **Save** the macro.
7. Insert another Table Transformer and wrap the previously created layout with the macro.
8. Select the macro and click **Edit.**
9. In the **Presets** tab select **Custom transformation** and click **Next**.
10. Enter the following SQL query:

```
SELECT * FROM T1
PIVOT (FIRST ('Dev-stage') for 'Value')
```

11. Click **Next**.
12. **Save** the macro and the page.

Splitting cell values in a column to different rows

Use case:

You need to split cell values in a column for further rows data aggregation.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SEARCH / AS @a EX('Skills'->split(",")) /
        RETURN(@a->'Name' AS 'Name', _ AS
'Skills') FROM T1
```

6. Click **Next**.
7. Define the table settings and view options if needed.
8. **Save** the macro and the page.

In this case the **underscore** in the query (i.e. *_ AS 'Skills'*) is utilized as a beacon to return the transformed data in the *Skills* column, while **@a ->** returns the data corresponding to their row values, hence multipling them as the *Skills* column data obtain additional rows after being split.

The resulting table can be further utilized with the Pivot Table macro for distinct value aggregation, counting and so on.

Using @currentUser and @pageTitle variables

Use case:

You have a table with the list of people responsible for publishing documentation pages.

You need to do the following:

- find the current Confluence username (i.e. who is logged in) with **@currentUser** variable in the Responsible column
- color it in blue and the others in green
- add the current page name (i.e. where the Confluence logged in user is now) with the **@pageTitle** variable to all the existing page names in the table.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table or the macros outputting tables within the macro body.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SELECT
FORMATWIKI("{color:" + IF(T1.'Responsible'=
@CURRENTUSER, "green", "blue") +"}" +
T1.'Responsible' + "{color}")
```

```
AS 'Responsible',
"@pageTitle - " + T1.'Page'
AS '@pageTitle'
FROM T1
```

6. Click **Next**.
7. **Save** the macro and the page.

@currentUser, @pageTitle variables are case-insensitive

Using multiple SQL statements. Inserting new rows

Use case:

You have a table with the date and the amount and need to show the amount increase by 4% from month to month.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table with the start date and the start amount.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
SET @cnt = 0;
SET @date = (SELECT T1.'Date' FROM T1);
SET @amount = (SELECT T1.'Amount' FROM T1);

WHILE @cnt < 11
BEGIN
SET @cnt = @cnt + 1;
SET @date = DATEADD(month, 1, @date);
SET @amount = @amount * 1.04;
```

```
INSERT INTO T1 (T1.'Date', T1.'Amount')
VALUES (@date, @amount);
END;

SELECT * FROM T1
```

6. Click **Next**.
7. Define the **date format** .
8. **Save** the macro and the page.

Using multiple SQL statements. Updating existing rows

Use case:

You have a table with dates and prices and need to move the dates two weeks forward and the prices 20% up.

Solution:

1. Switch the page to the edit mode.
2. Insert the Table Transformer macro and paste the table with the dates and the prices.
3. Select the macro and click **Edit**.
4. In the **Presets** tab select **Custom transformation** and click **Next**.
5. Enter the following SQL query:

```
UPDATE T1
SET T1.'Date' = DATEADD(week, 2, 'Date'),
T1.'Price' = ('Price' * 1.2);

SELECT * FROM T1
```

6. Click **Next**.
7. Define the **date format** .
8. **Save** the macro and the page.