

# Create Branching Outputs

From a single collection of datasets, you may need to generate multiple outputs for downstream purposes. Examples:

- You want to preserve the ability to review and profile your source data. For more information, see [Profile Your Source Data](/dataprep/docs/html/Profile-Your-Source-Data_57344603) (/dataprep/docs/html/Profile-Your-Source-Data\_57344603).
- You need different pivot tables produced from the wrangled data.
- You need to filter down the set of rows or columns to deliver to one user community while delivering a different set of columns to another.

## Reshaping Transformations

If your next step is to add any of the following transformations and you wish to preserve the existing data for other uses, you should consider adding these steps in a separate dedicated recipe.

Transformation Name	Description
Union	A union appends one or more datasets to your current one. To preserve the original, you may need to create a branching output. See <a href="/dataprep/docs/html/Union-Page_57344884">Union Page</a> (/dataprep/docs/html/Union-Page_57344884).
Join	A join combines two datasets based on common values in specified columns in both datasets. These types of transformations can greatly change the shape of your data. See <a href="/dataprep/docs/html/Join-Window_57344880">Join Window</a> (/dataprep/docs/html/Join-Window_57344880).  Similarly, a lookup uses values from a column in your source data to pull in corresponding rows of data from a reference dataset. These transformations add columns to your dataset. See <a href="/dataprep/docs/html/Add-Lookup-Data_57344594">Add Lookup Data</a> (/dataprep/docs/html/Add-Lookup-Data_57344594).
Deduplicate	This transformation removes identical rows from your dataset. However, there may be a set of steps required to standardize values in various columns before applying the de-duplication. You may choose to manage this process in a branching recipe.
Delete columns	When a column is removed, it is no longer available for use in any downstream output. See <a href="/dataprep/docs/html/Remove-Data_57344583">Remove Data</a> (/dataprep/docs/html/Remove-Data_57344583).

Filter	Rows can be filtered from your dataset to render different perspectives. These changes may be best moved to a secondary, branching recipe. See <a href="/dataprep/docs/html/Filter-Data_90112050">Filter Data (/dataprep/docs/html/Filter-Data_90112050)</a> .
Pivot data	When you create a pivot table, all source data that is not explicitly specified in the pivot is dropped from the dataset. For more information, see <a href="/dataprep/docs/html/Pivot-Data_132067520">Pivot Data (/dataprep/docs/html/Pivot-Data_132067520)</a> .
Group by	You can perform aggregation calculations within a table, which may force column data to be dropped. See <a href="/dataprep/docs/html/Create-Aggregations_132067519">Create Aggregations (/dataprep/docs/html/Create-Aggregations_132067519)</a> .

## Basic Technique

Whenever you are applying a transformation that destroys data or otherwise reshapes your dataset and you wish to preserve the current state of the dataset, you should do the following:

1. In Flow View, select your current recipe. Click **Add new recipe**.
2. This recipe becomes the source for a branched output. Give the new recipe an appropriate name. For example, `Pivot-SalesPerProductPerStore`.
3. For this recipe, click the Output icon. Specify the appropriate output format and location that you'd like to generate for this branched output.
4. Select your current recipe again. Click **Add new recipe**.
5. This recipe becomes the extension of your current recipe. Give the new recipe an appropriate name. For example, `MyRecipe-Part2`.
6. Select the `Pivot-SalesPerProductPerStore` recipe. Click **Edit recipe**.
7. Build your pivot transformation in this recipe.
8. When ready, run the job. The output should be generated in the appropriate format and location.

The screenshot displays the Google Cloud Dataprep interface. On the left, a flow diagram shows a primary recipe node 'POS-r01-Part1' receiving input from 'POS-r01-source.txt'. From 'POS-r01-Part1', multiple lines branch out to other nodes: 'ValuesToCols', 'Pivot-SalesPerProductBySt...', 'POS-r01-Part2', and another 'POS-r01-Part1' node. The 'Pivot-SalesPerProductBySt...' node is highlighted with a blue circle and a play icon. On the right, a 'Details' panel for the 'Pivot-SalesPerProductByStore' recipe is open. It includes a 'Run Job' button, a 'Destinations' tab, and sections for 'Manual Destinations' and 'Scheduled Destinations'. The 'Manual Destinations' section shows a 'Create-CSV' destination with the path 's3://3fac-testing/trifacta/queryResults/admin@trifacta.local/Pivot-SalesPerProductByStore.csv', an environment of 'Trifacta Server', and profiling set to 'yes'.

**Figure: Multiple pivot tables sourced from output of a primary recipe for the flow. POS-r01-Part2 can be used for continued wrangling of primary recipe.**

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-07-13 UTC.