SAS® Viya® Trial

Manage Data Guide

Data Engineering Tasks

Intro

Data and AI Life Cycle: Manage Data

A recent study by The Futurum Group showed that SAS Viya increases data and AI team productivity by 4.6x.

The analysts compared SAS Viya to alternatives in an end-to-end customer churn prediction analysis, a common use case relevant to many industries.

The first step in the data and AI life cycle is **Manage Data**. This was performed by a **Data Engineer** persona, who was tasked with evaluating and preparing the raw data into an analytical base table that could then be utilized by a Data Scientist persona.

This guide will walk you through the steps that a Data Engineer took to complete the Manage Data portion of the life cycle in SAS Viya.

Data

Data Sets

The data we use for this experiment is simulated data by using SAS code.

We'll use two data sets in CSV (comma delimited).

1. BANKING_ACCOUNT (10,088 rows & 19 columns).
2. BANKING_CUSTOMER (10,095 rows & 39 columns).

The primary key for these data sets is the “Id” column.

Our target variable (what we want to predict) is called “Churn.”

The purpose for the Data Engineer is to understand, clean and prepare the data and create a modeling-ready table (aka Analytical Base Table (ABT)), which will be passed to the Data Scientist to predict which customers have the highest probability of churn so the business can take the necessary actions to try to prevent that from happening.

Data Engineer

Clean, Prepare and Govern Data

Tasks

1. Data Profiling
2. Data Sensitivity & PII Checks
3. Data Quality
4. ETL Flow

Resources

Watch before starting

- Quick Start - Data & AI Life Cycle
- Quick Start - SAS Drive
- Quick Start - Manage Data
- Quick Start - Discover Information Assets
- Quick Start - Develop SAS Code in SAS Studio
- Quick Start - Develop Flows in SAS Studio
- Webinar - SAS Studio Flow & Steps

The Data Engineer

- As a Data Engineer, it is your job to upload a new data set for analysis and perform some basic data quality measures to ensure the data is ready for further use and modeling. It is also essential that the data doesn’t include any personally identifiable information (PII) since this project will be accessed by many different parties who shouldn't have access to sensitive information, and you want to mitigate any risks and be regulatory compliant.

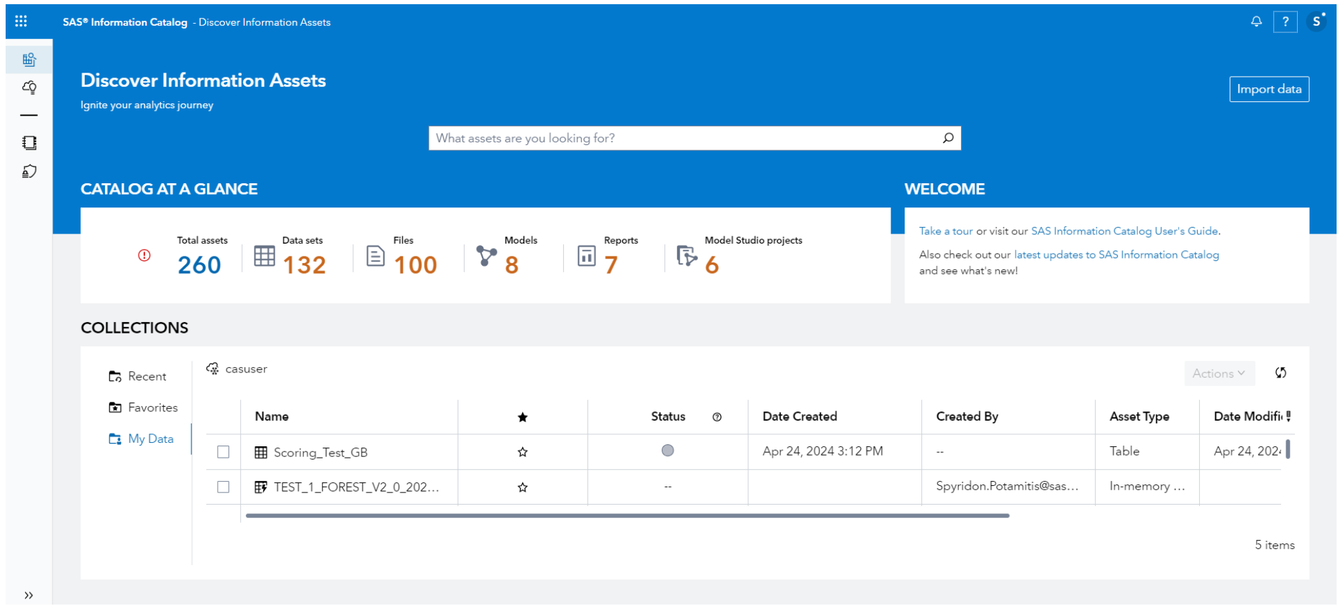
- You will work on uploading the data, profiling it to examine data quality and sensitivity and creating an ETL (Extract, Transform, Load) flow to prepare the data for modeling. The flow will create a unified data set that the Data Scientist needs to develop AI models.

Data Profiling

Information Catalog

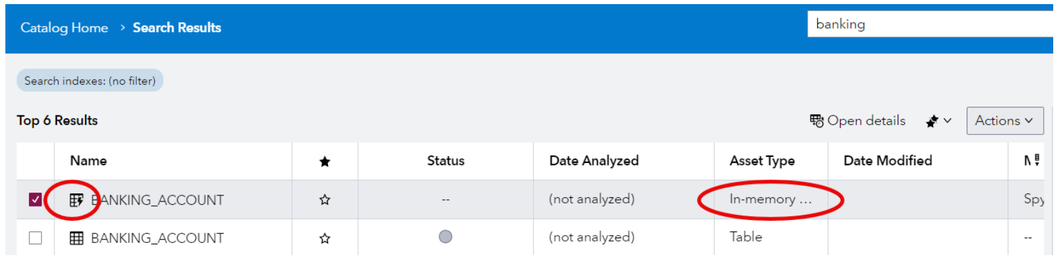
Intro

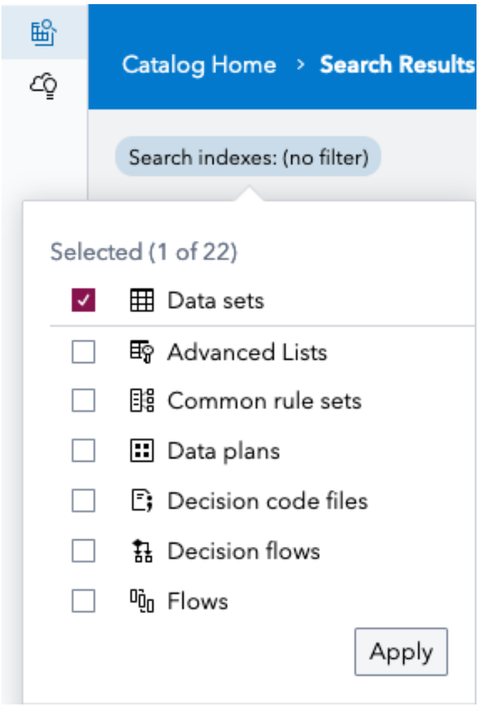
- To get more information on the data quality, we should navigate to Information Catalog and run a data profile. This will get us privacy information and descriptive statistics about the quality of our data.
- SAS Information Catalog lets you create and maintain an inventory of your information assets. Such a catalog gives you the ability to ingest, integrate and enrich metadata from the assets that are distributed across your enterprise. You can use this metadata to find and understand the relevant assets that you need to reach your business goals. An information catalog also enables data administrators to review data usage, such as when the data was created, who created the data, who modified it and when the data was modified for the last time, from a single point of access.



Data Profiling

- Select the Applications Menu (on the left of the screen) and go to “DISCOVER INFORMATION ASSETS.” From there, you can search for and select the data sets we’ve just uploaded. Filter for data sets or type “banking” in the search tab to make searching easier.
- You'll see the names of data sets appearing twice. The ones with the thunderbolt next to their table name are already loaded into memory. We'll only use in-memory data sets.

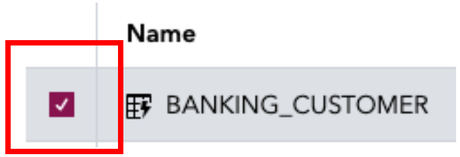




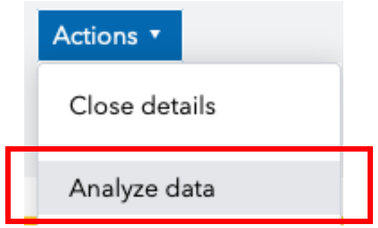
Data Profiling

- Let’s start by selecting the BANKING_CUSTOMER data set. Navigate to the ACTIONS dropdown from the right of the screen and select ANALYZE DATA to run a data profile on the data set.
- Once complete, click on “View analysis” near the top of the screen, and you will be able to view descriptive statistics about your data set. Let's dive deeper.

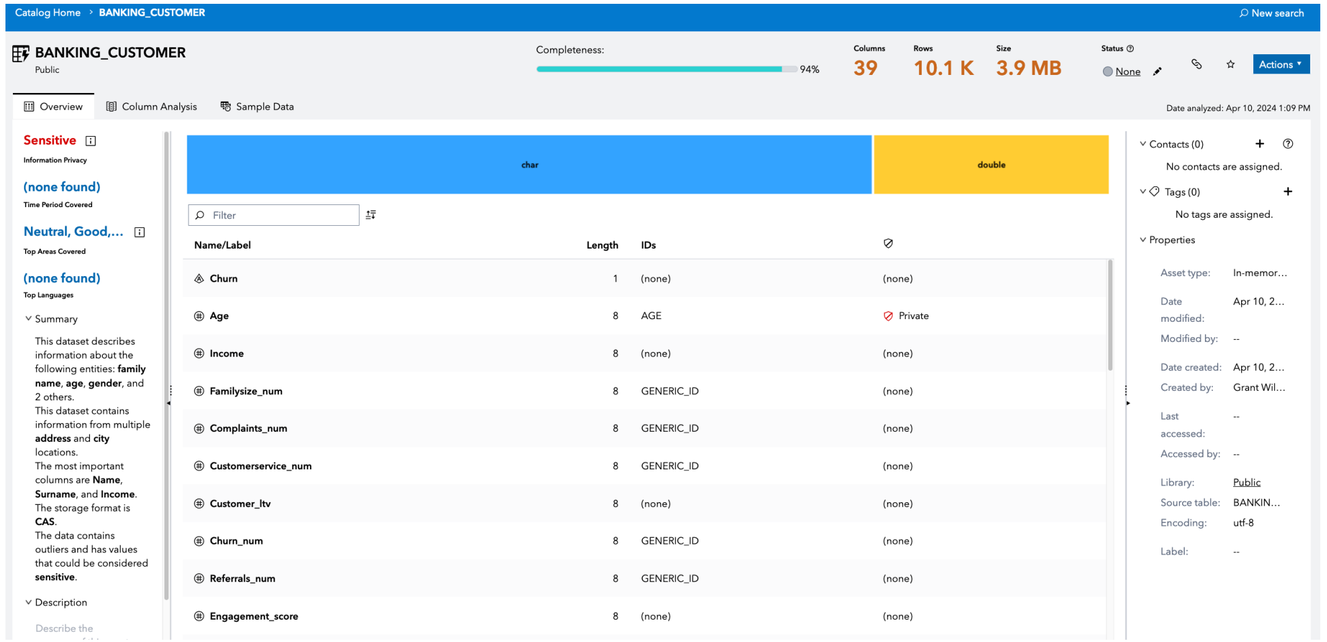
1



2



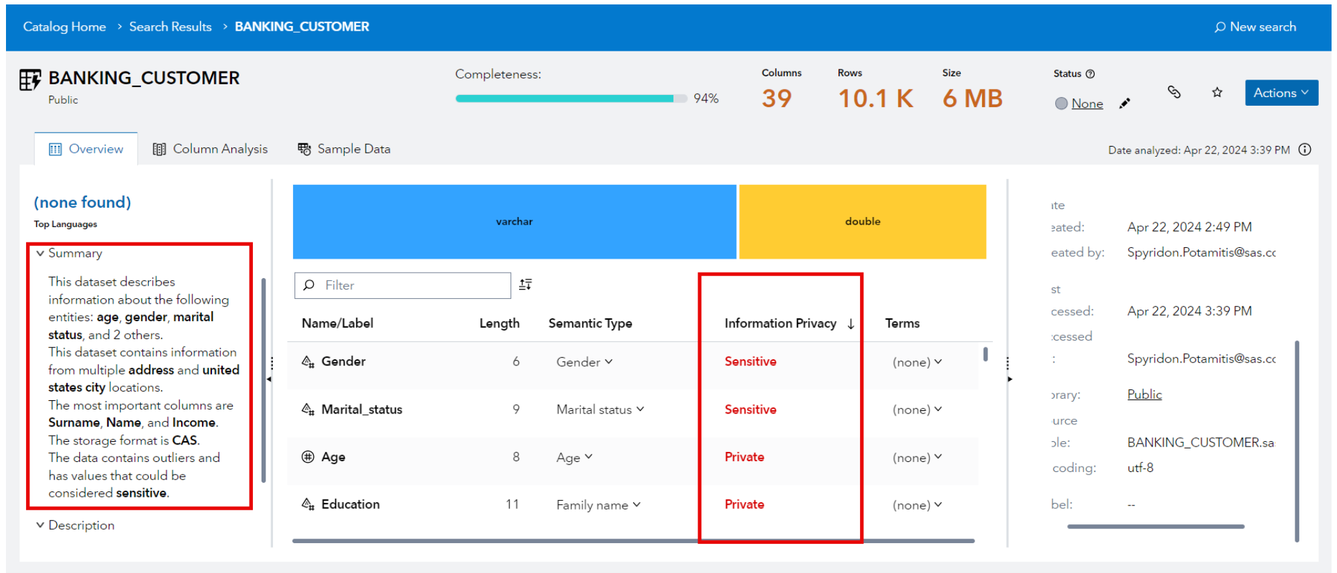
3



Data Sensitivity & PII Checks

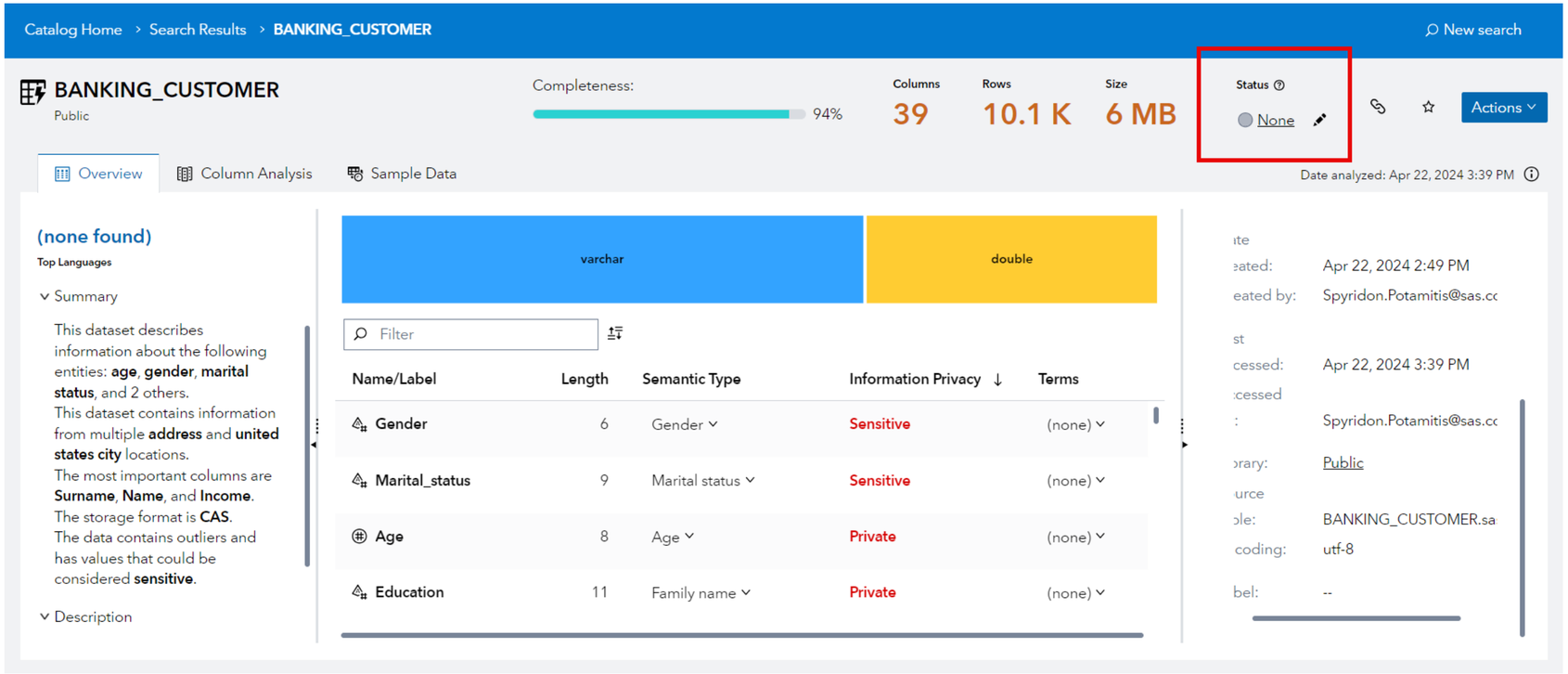
View Column-Level Data Sensitivity

- In the middle of the screen, we see descriptive information about our column-level data sensitivity. On the left of the screen, you should also see a summary of the findings in natural language where sensitive data and outliers are detected. Both descriptions are automatically generated by SAS Viya. The information privacy is broken into four categories (in this data set you should see all categories). Based on this information, the Data Engineer should take the necessary steps to safeguard customers' personal and sensitive data as well as investigate and treat outliers if necessary.
  - None
    - There is no PII in this column.
  - Candidate
    - There may be PII in this column.
    - Location, Market Condition.
  - Private
    - This column contains PII.
    - Name, Surname, Age, Gender.
  - Sensitive
    - There is sensitive PII in this column.
    - Marital Status.



Flag Data Set

- Now that we know more about the data set, we want to let our Data Engineering colleagues know that they shouldn't share this data set with other teams that shouldn't have access to customers' personal and sensitive data. To do that, we'll set a warning for this data set by clicking on the pencil icon under the “Status” tab. In there, we'll also write a comment that “personal and sensitive data was detected.” This flag will be visible to everyone who has access to the data set in all SAS Viya applications. For example, if another Data Engineer wants to use SAS Viya applications such as Manage Data, Discover Information Assets or even Explore Lineage, this important info will be available to ensure proper governance of our data assets throughout the organization.

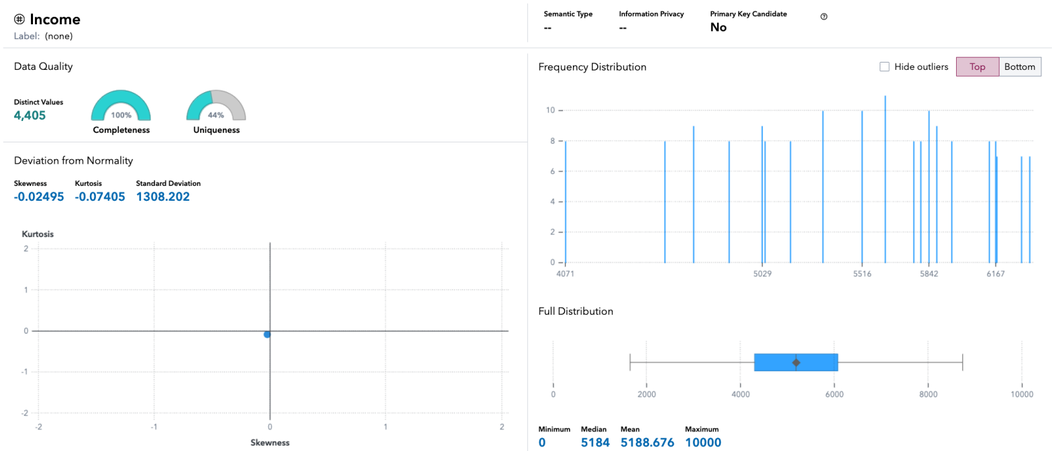


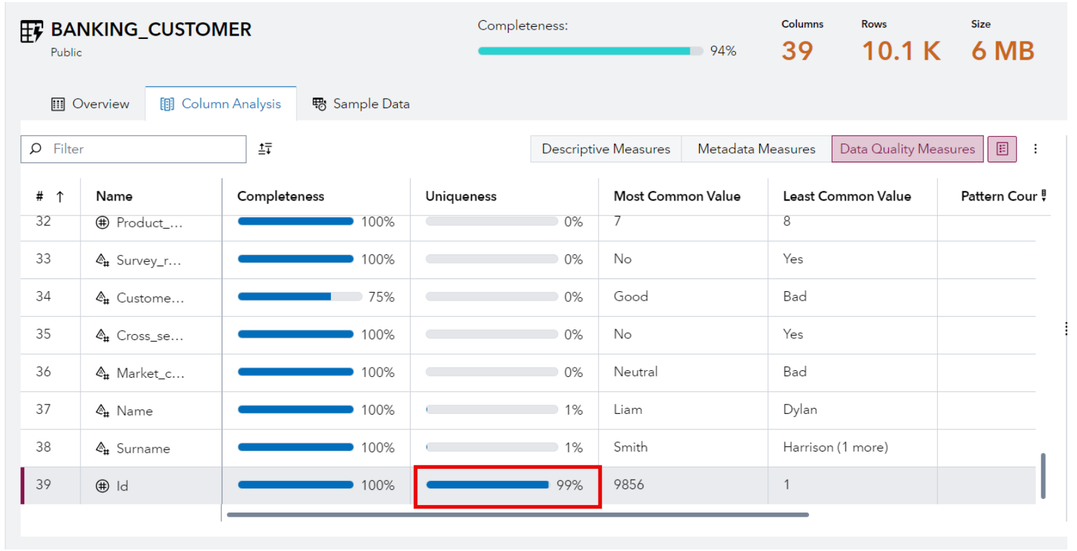
Data Quality

Get Data Quality Measures

Click a column name to view Data Quality Measures

- From the data profile, we can drill into the variables to get some metrics for data quality. Select the name of one of the numeric variables or click “Column Analysis.”
  - Skewness
  - Kurtosis
  - Completeness
  - Uniqueness
  - Distinct Values
- For numeric variables, you will see metrics. For character variables, you will see patterns.
- Very quickly we can identify that our primary key “ID” has duplicate values by examining the uniqueness indicator.

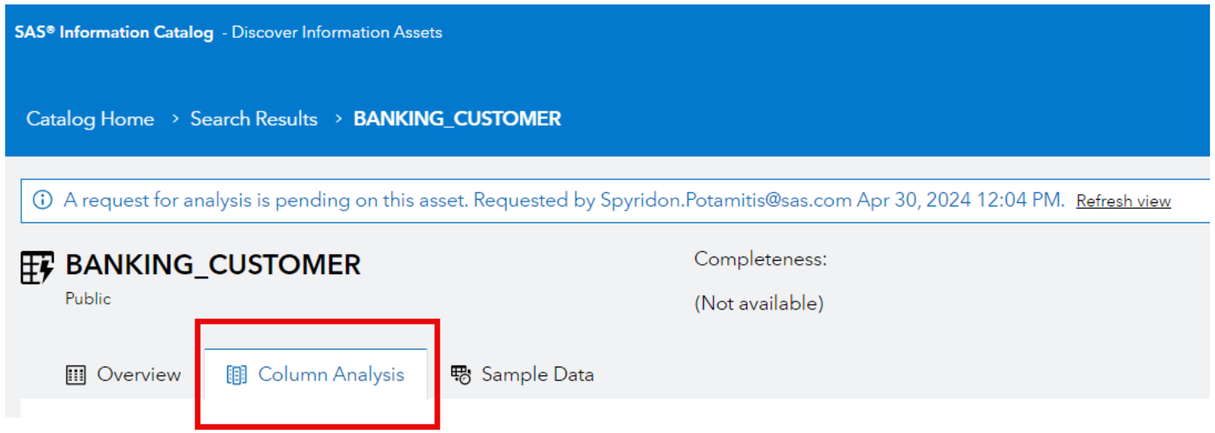


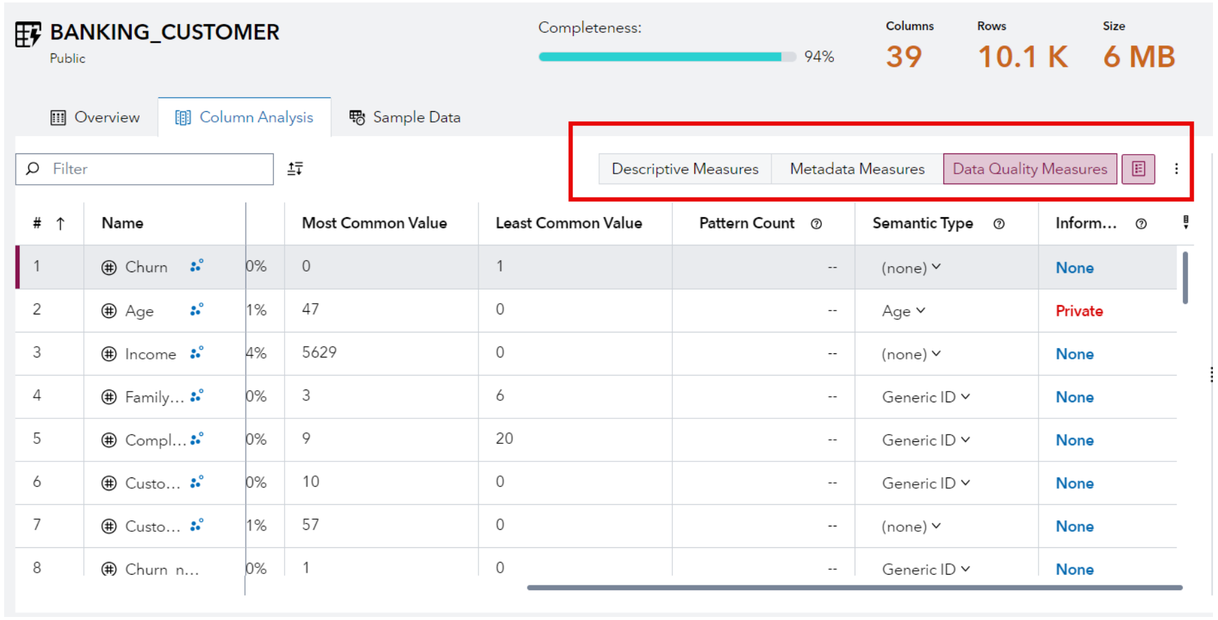


Column Analysis

Data Quality Measures

- On the Column Analysis page, we can view Descriptive Measures, Metadata Measures and Data Quality Measures.
- First click the Column Analysis and run it. You'll be able to see the analysis by clicking the "View the analysis" button when the run is complete.
- Here, we can view Descriptive Measures, Metadata Measures and Data Quality measures to get a comprehensive view of our data.
- Opening the Data Quality Measures page gives us the following information:
    - Completeness
    - Uniqueness
    - Most and Least Common Values
    - Pattern Count
    - Semantic Type
    - Information Privacy



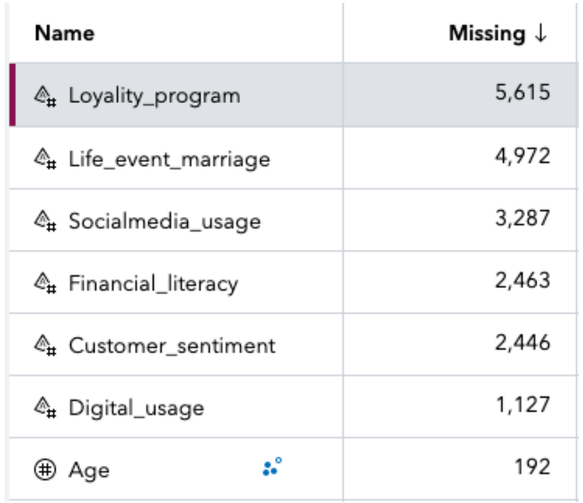


Column Analysis

Descriptive Measures

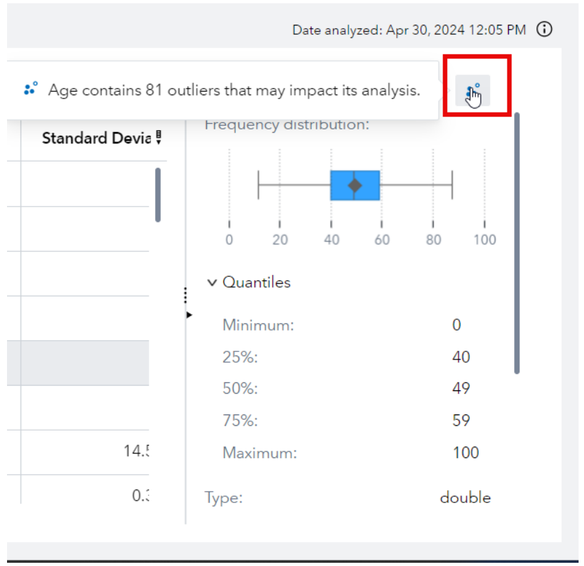
- Let's open the “Descriptive Measures” tab for our BANKING_CUSTOMER data set.
- If we sort by our “Missing” column (by clicking on “Missing” column name), we can identify which columns have missing values. We can quickly notice a few that we’ll need to assess in our data flow:
  - Loyalty Program
  - Life Event Marriage
  - Social Media Usage
  - Financial Literacy
  - Customer Sentiment
  - Digital Usage
  - Age
- When we click next to a variable name, we’ll see a distribution graph, and the quantiles appear as well. Selecting the “Age” variable, for example, will give us a box and whisker plot describing the values of our selected variable and measures of central tendency. We’ll also get a small icon on the top right that when we double-click on it will tell us how many outliers we have in this variable.

1

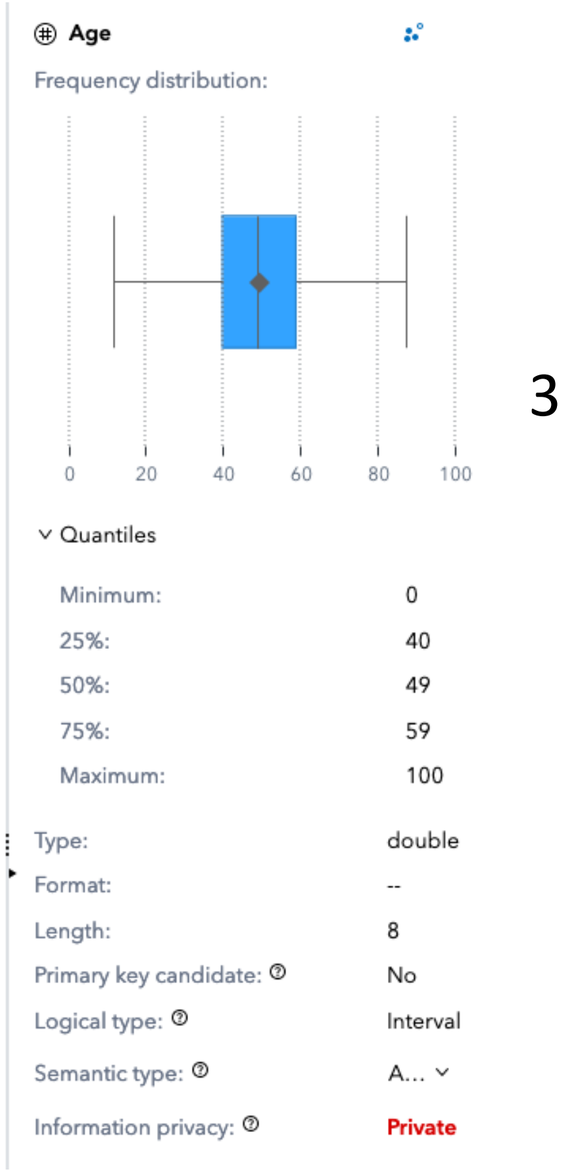


Name	Missing ↓
Loyality_program	5,615
Life_event_marriage	4,972
Socialmedia_usage	3,287
Financial_literacy	2,463
Customer_sentiment	2,446
Digital_usage	1,127
Age	192

2



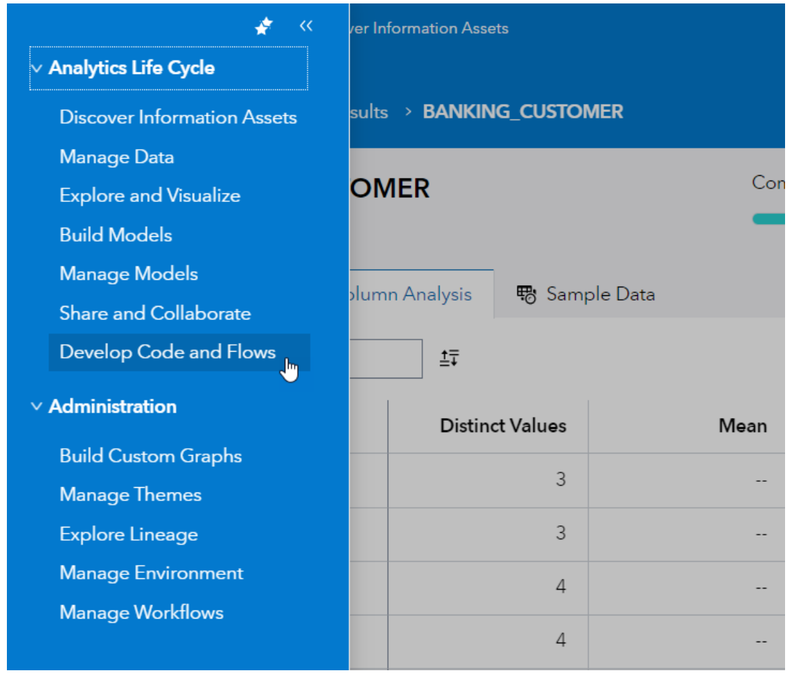
3

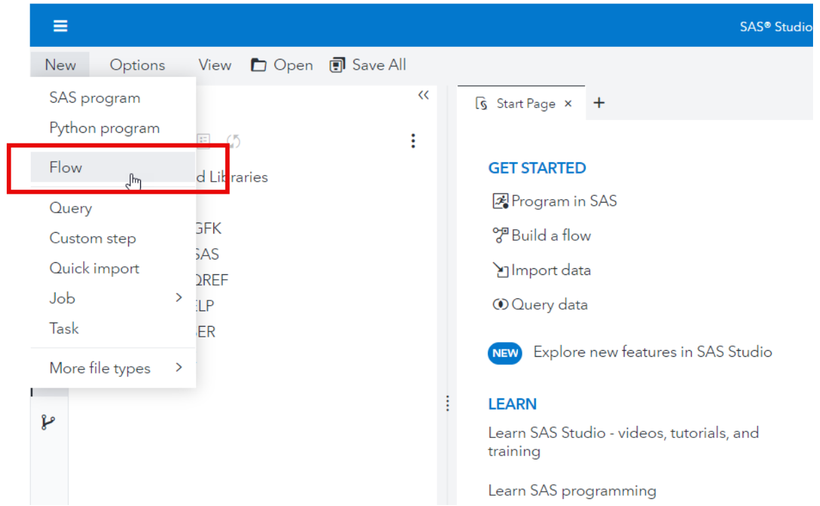


ETL Flow

Create a Flow

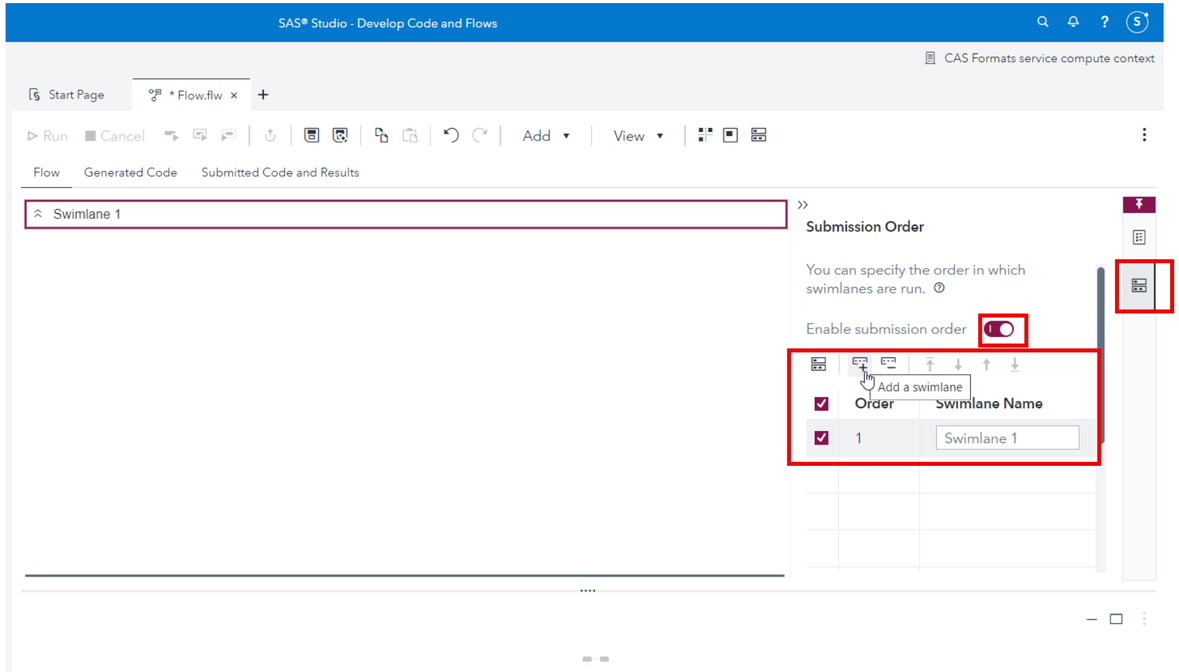
- Now that we've had a chance to explore our imported data sets, let's build a custom flow to create an ETL (Extract, Transform, Load) workflow that will be easily repeatable and easy to implement in a low-code/no-code environment to automate repetitive processes.
- The end goal of our ETL process is to develop an ABT table, which is ready for the Data Scientist to use to develop models.
- Open the Applications Menu (top left of the screen) and select "DEVELOP CODE AND FLOWS."
- Open the "New" dropdown and select "Flow." This will serve as the canvas for our ETL Flow.





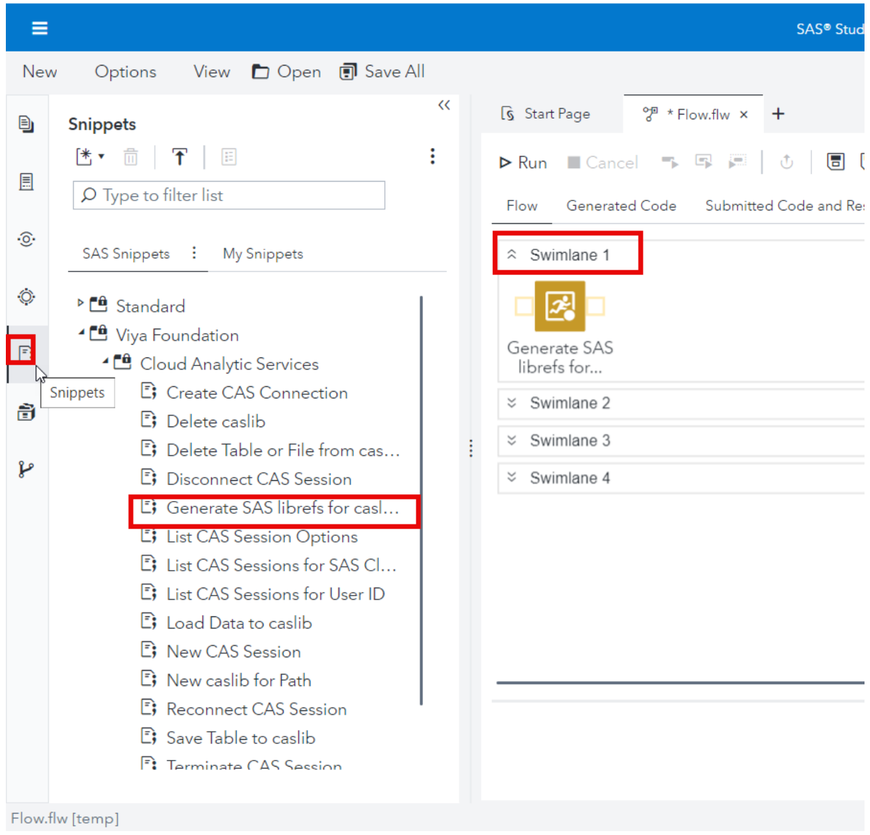
Create a Flow

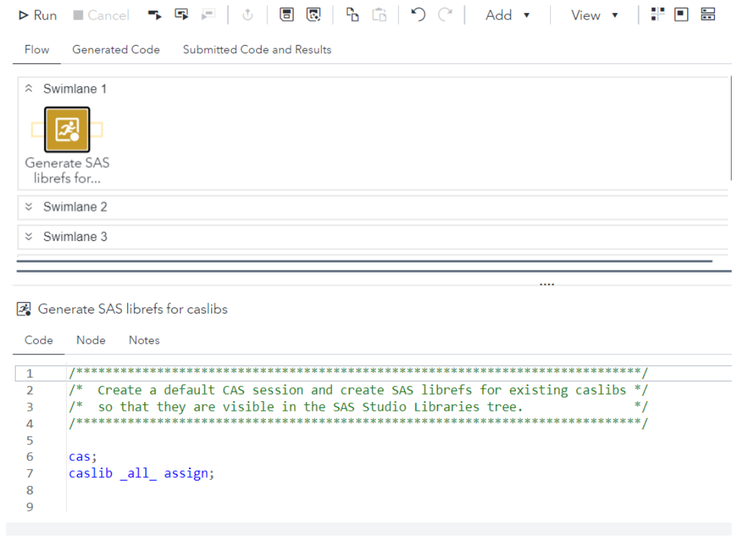
- In examining data quality, you've noticed duplicates in the data in addition to some data sensitivity issues. We will need to clean each table and join them together.
- Start by ensuring swimlanes are enabled by opening the pane on the right and clicking the relevant button. This will enable us to run parts of our code separately and ensure we're not rerunning code needlessly. This also guarantees that we get a logical sequence in the flow with the necessary dependencies enforced.
- Click the "Add a swimlane button" as you see on the right to add a swimlane in the flow. Now you should have two swimlanes in total.



Connect to CAS

- As a first step in our flow, we need to connect to Cloud Analytic Services (CAS), which is a SAS Viya in-memory engine, and start a session.
- Let’s begin by dragging a pre-built SAS Code node referred to as a “Snippet” onto our canvas. Code snippets are lines of commonly used code that you can use in an existing program or as the basis for a new program. Code snippets enable you to quickly insert SAS code into your program and customize it to meet your needs.
- Find the “Generate SAS librefs for caslibs” Snippet and drag it to Swimlane 1.
- If you click on the node you created in the flow, you will be able to see the code that is generated automatically for you.
- Run the node by right-clicking it and clicking “Run node.”

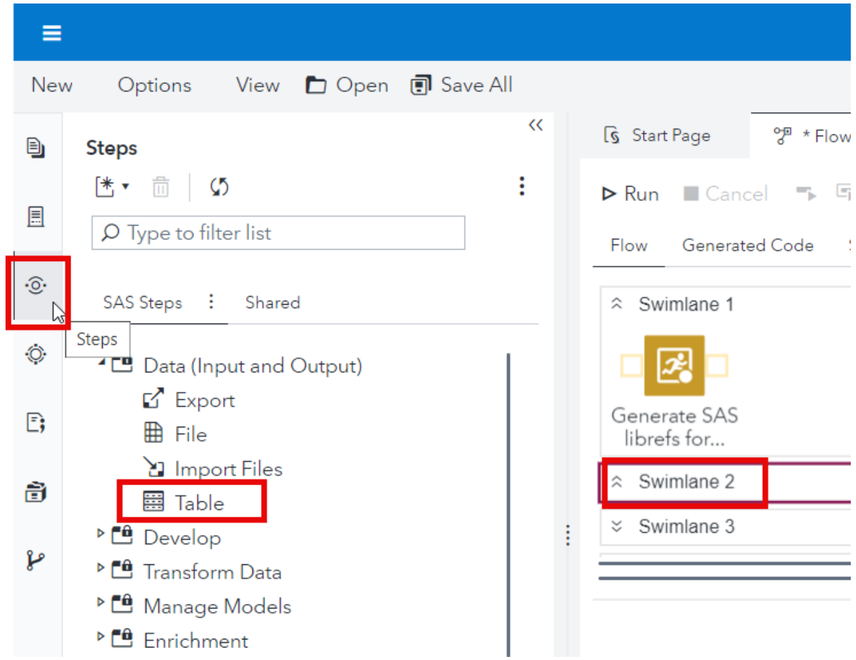


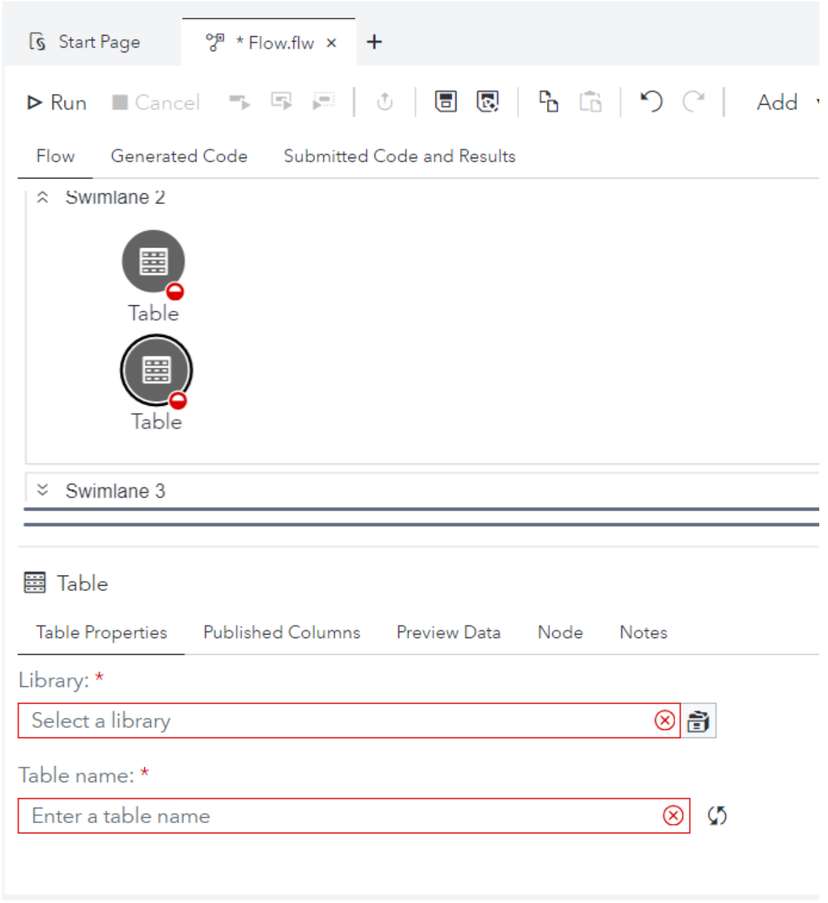


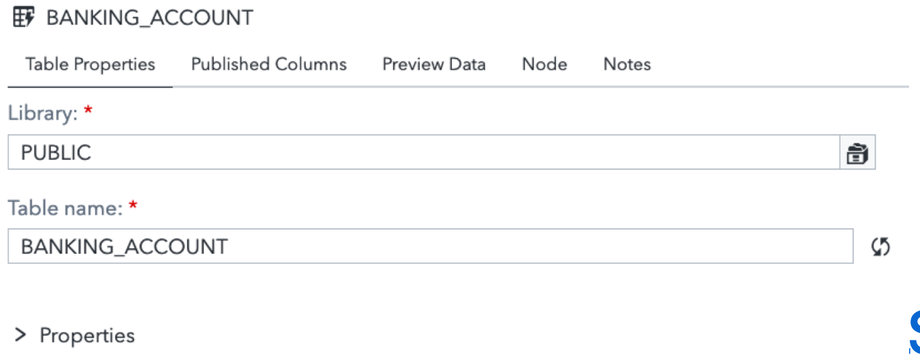
Deduplicate

We've seen from the uniqueness analysis that input data sets contain duplicates for both the BANKING_ACCOUNT and BANKING_CUSTOMER data sets.

1. Open the "Steps" pane from the left of the screen and drag two table Steps into Swimlane 2 of the flow and select our input data sets by using the options on the bottom of the page.
2. For the first table, select "Public" as Library and "BANKING_ACCOUNT" as a table name. For the second table, select "Public" as Library and "BANKING_CUSTOMER" as the table name.







Deduplicate

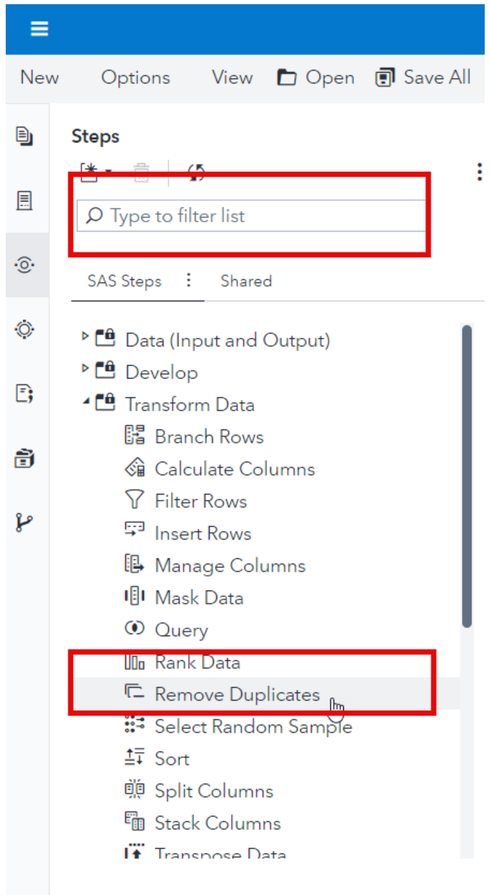
1. Now identify the “Remove Duplicates” step in the list under the “Transform Data” folder. You could also use the search on the top to find it quicker.

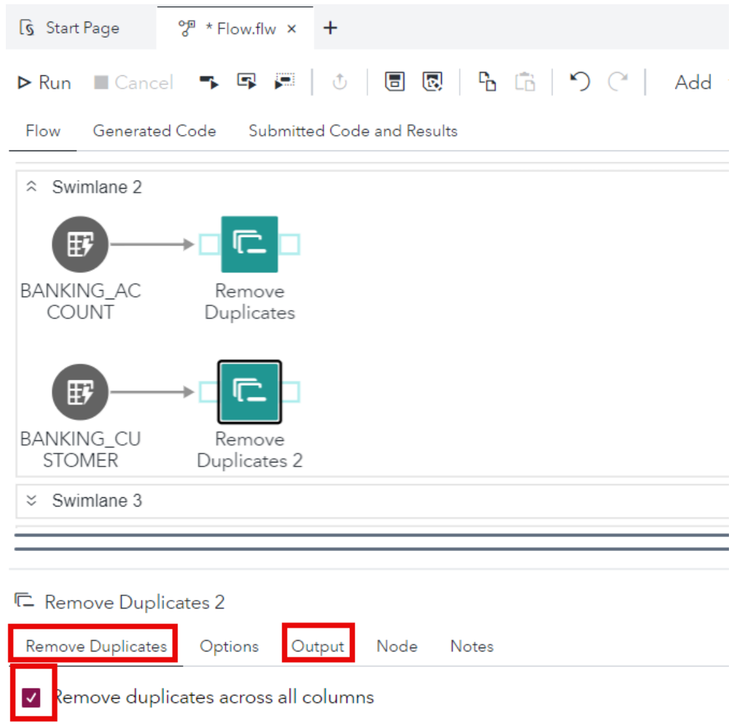
2. Drag two of these steps in the Swimlane 2.

3. Connect the table steps you see in the flow to “Remove Duplicates” steps by using your mouse. Hover over the table nodes and when you see a small hand (instead of the default arrow icon when using the mouse), hold your left click to create an arrow that will connect the two nodes.

4. Ensure you’re removing duplicates across all columns and replacing the output table with the same name (default options).

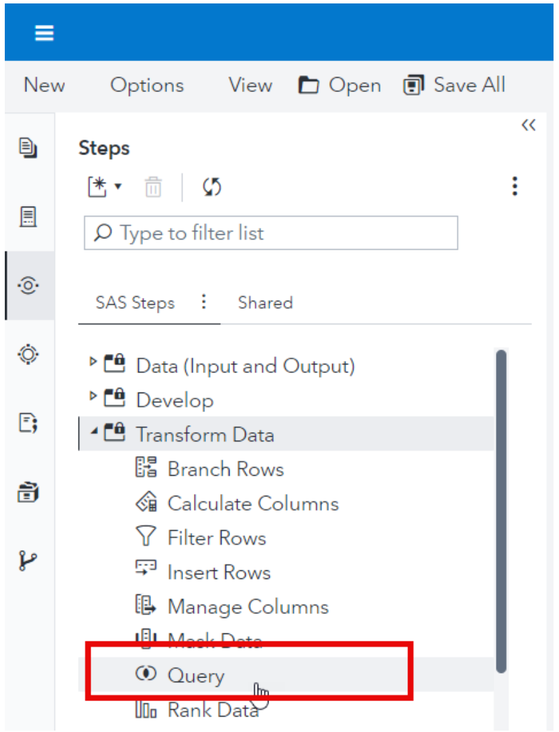
5. Right-click anywhere in the flow (Swimlane 2) and select “Run Swimlane.”

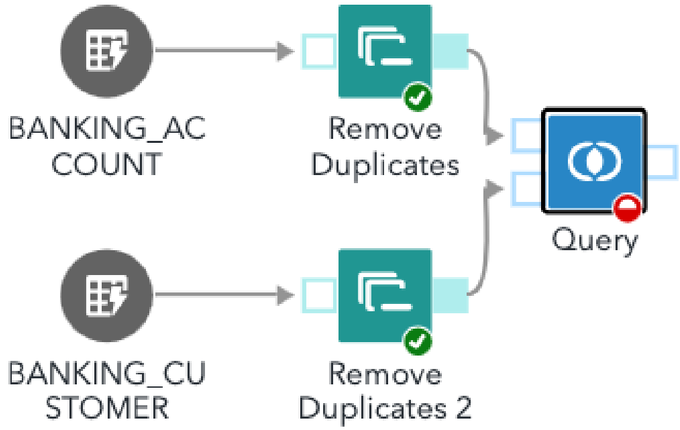




Join the Data

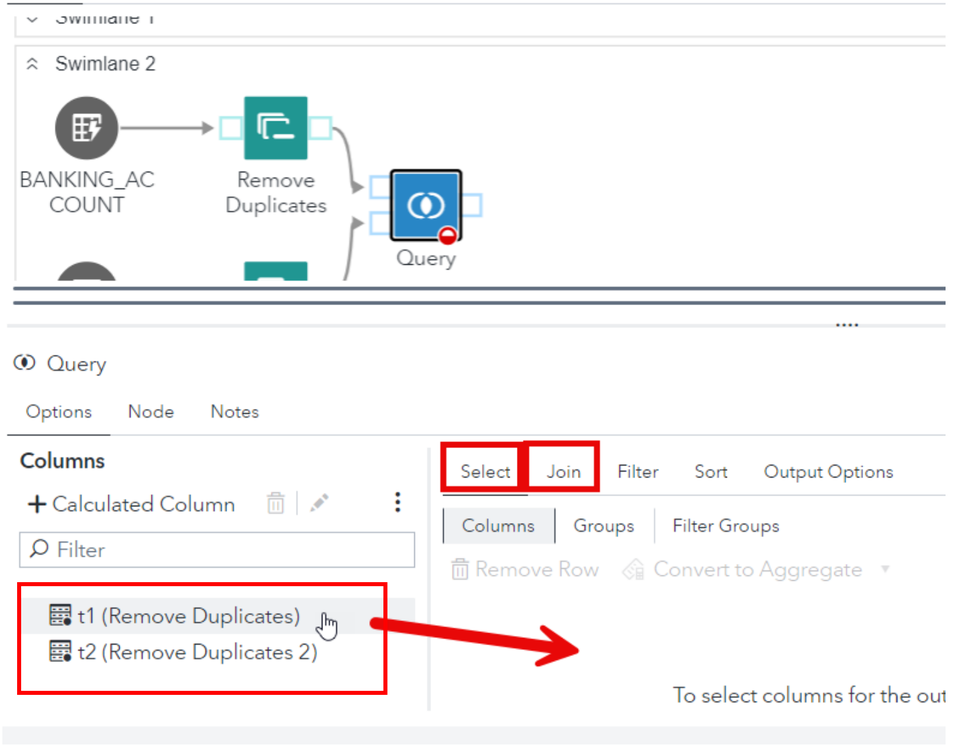
- Once we've deduplicated our data, we'll need to join it together.
- From the "Steps" tab, drag a Query step into the flow and connect the previous nodes ("Remove Duplicates") to the input ports of the Query node as you see on the right by using your mouse.
- Now, click on the Query step and check the options at the bottom of the page.

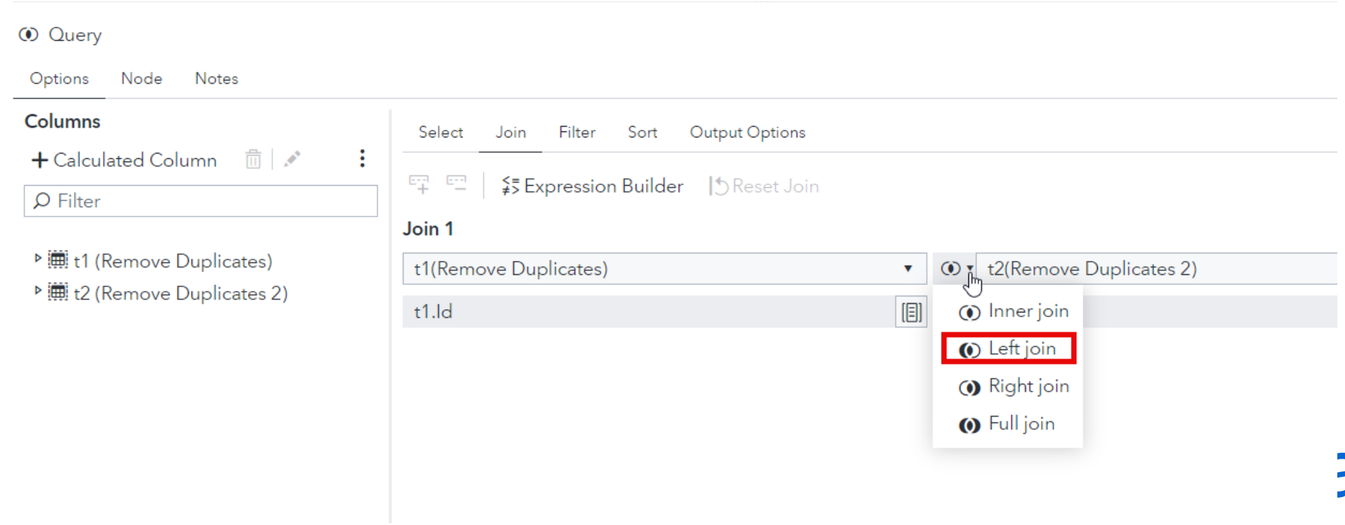




Join the Data

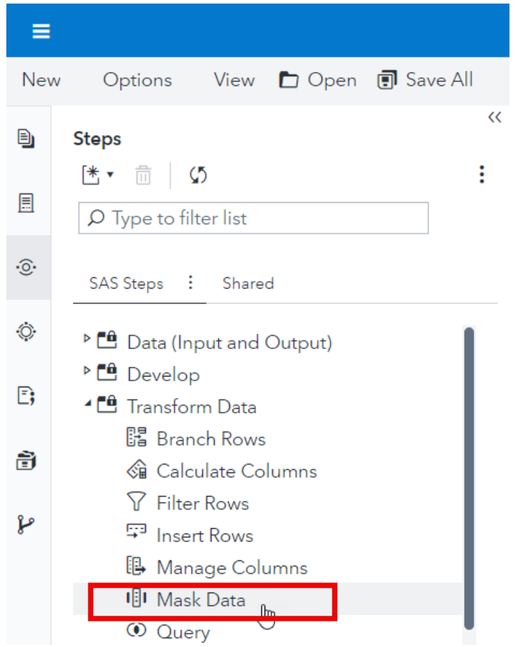
- Double-click or drag both tables (t1 and t2) onto the “Select” canvas options pane.
- Open the “Join” tab in the “Query” options and ensure that we are:
  - Joining on our ID variable. This should have been detected automatically.
  - Running a “Left Join.”
    - This won’t be the default. Select the “Venn Diagram” icon as you saw in the second graph to change to a “Left Join.”
- We conduct a “Left Join” since our customers may have multiple accounts, and we don’t want to ignore any accounts.
- Right-click on the “Query” node and select “Run node.”

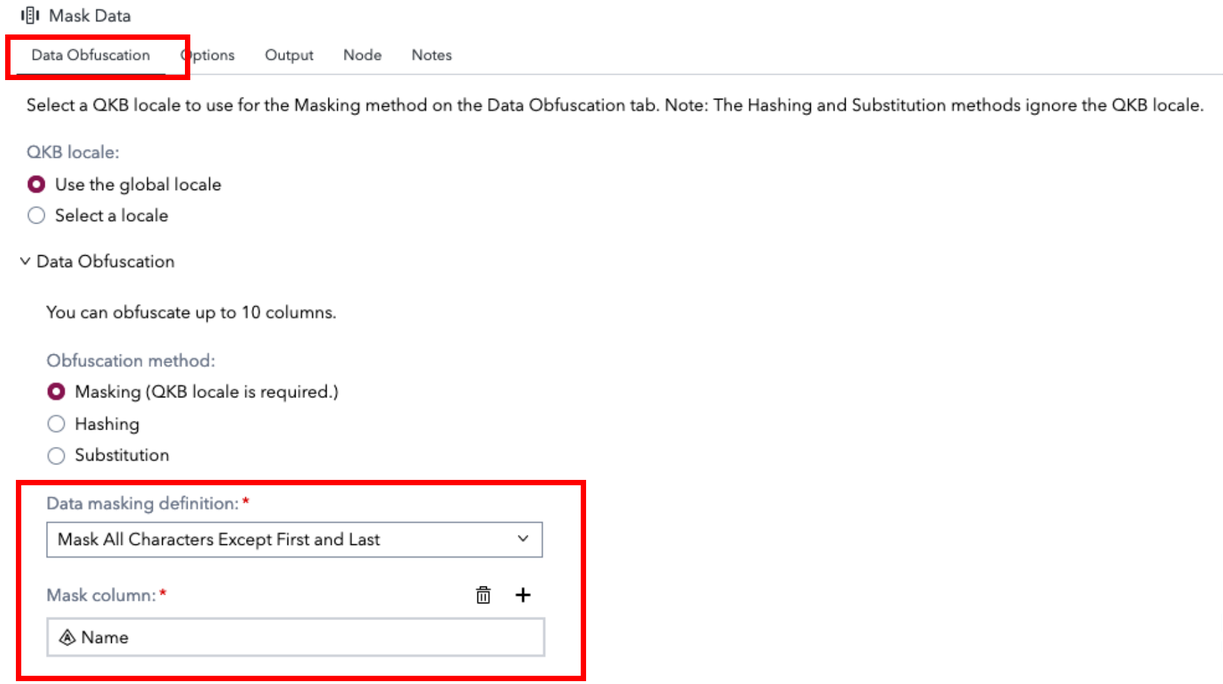




Mask Data

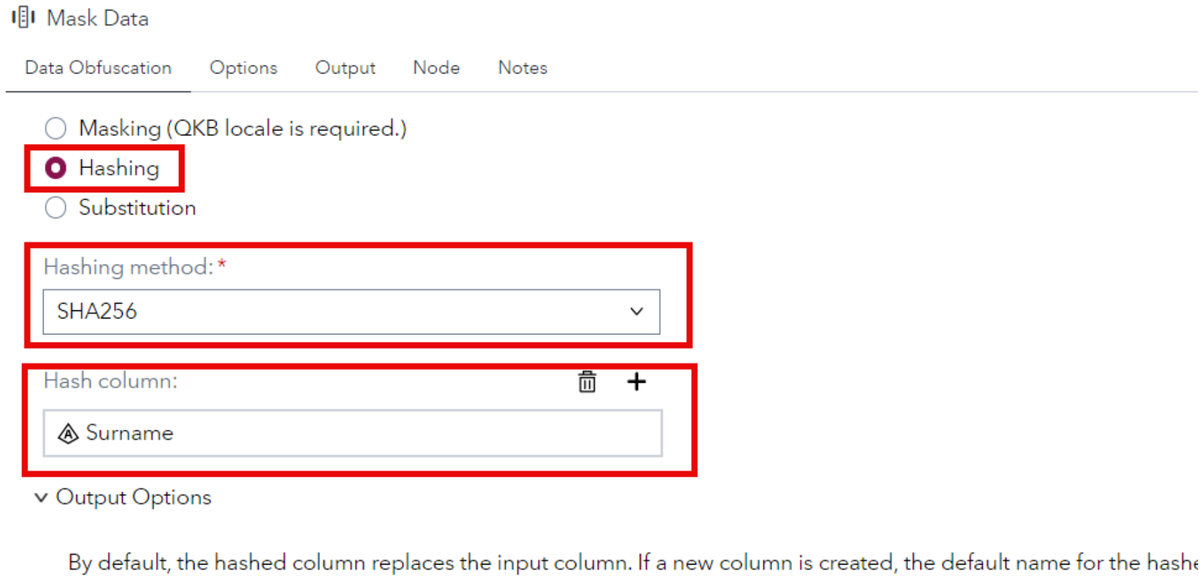
- In addition to joining our data, we'll need to "Mask" some sensitive and/or private information before moving the data to the Data Scientists based on our company's policies.
- Drag a "Mask Data" step into the flow and connect it to the "Query" node that you ran before.
- Recall that we have "Name" and "Surname" variables in our data that are private.
  - In the "Mask Data" options, under the "Data Obfuscation" pane in the "Mask Data" node, let's choose Masking for the "Name" column and mask all characters except the first and last and replace the column so it no longer contains the input names.





Mask Data

- Scroll down to see “Additional Data Obfuscation” and create a “Hash” for the “Surname” variable so it’s not human readable but could be decoded later.
- Make sure in the “Output” pane that we are replacing the existing table.
- Don’t run the node yet, as we want to save the output table in a temporary table to check our results so far and save our flow, as well.

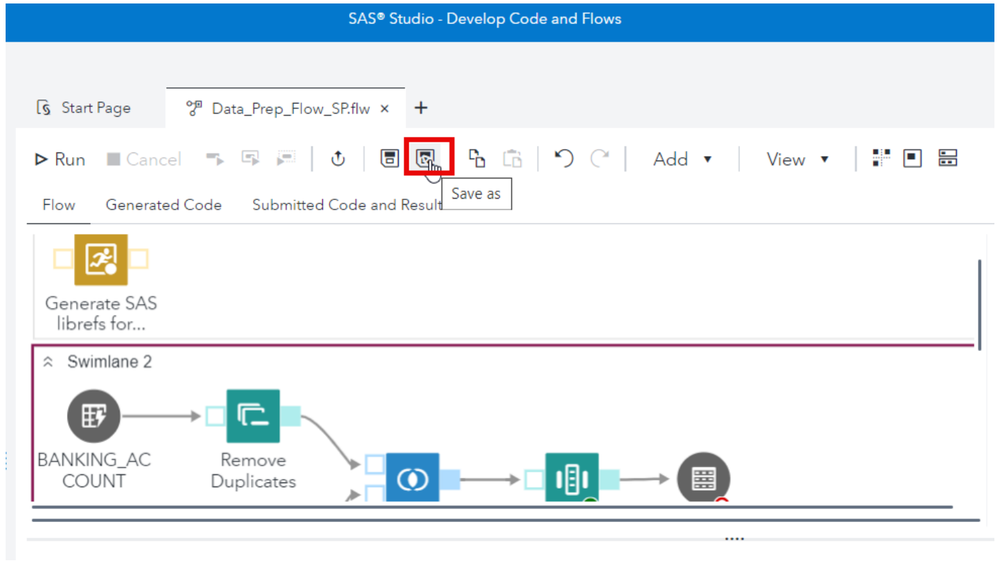


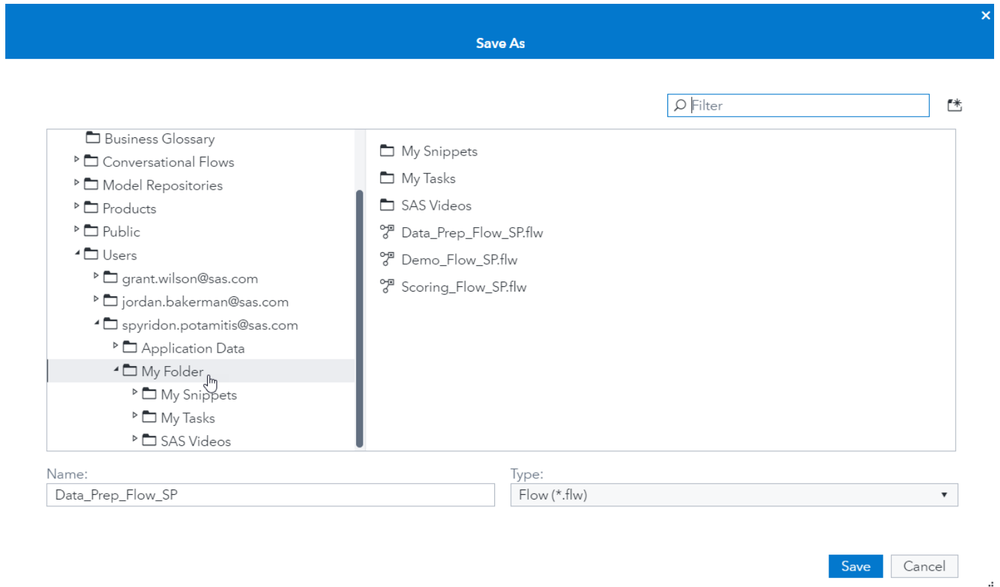
Save Data and Progress So Far

Since we have performed a lot of work already, we want to save the flow we created so far and the output of our flow till now.

First save the flow by clicking anywhere on the flow and then the “Save” icon. Navigate to “SAS Content”-> “Users”-> “My Folder,” and then give an appropriate name to your flow and click save.

The next thing we want to do is to save our transformed and masked data so far in a table.



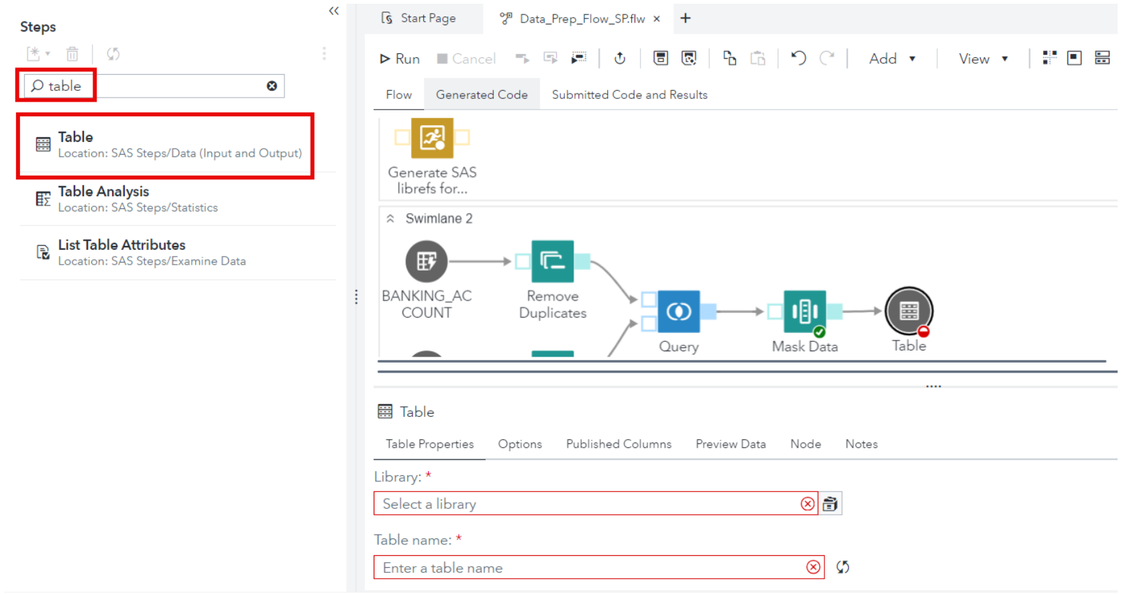


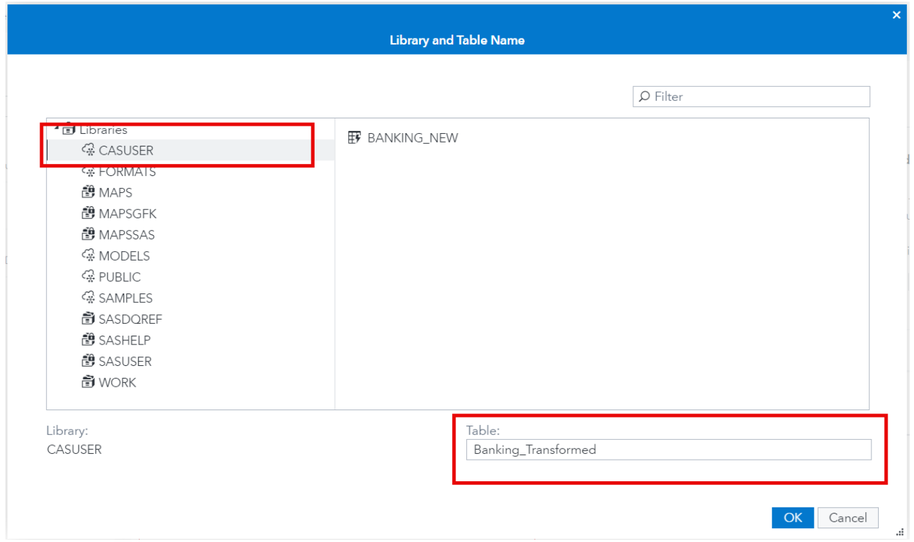
Save Data and Progress So Far

From the “Steps tab,” connect a “Table” node to the “Mask Data” node in Swimlane 2.

As this is not the final table that we want to deliver to the Data Scientist and works like a checkpoint, select in the “Table” node options as “Library” “Your User Library," “CASUSER” and as a “Table Name,” give the name “Banking Transformed” and click OK.

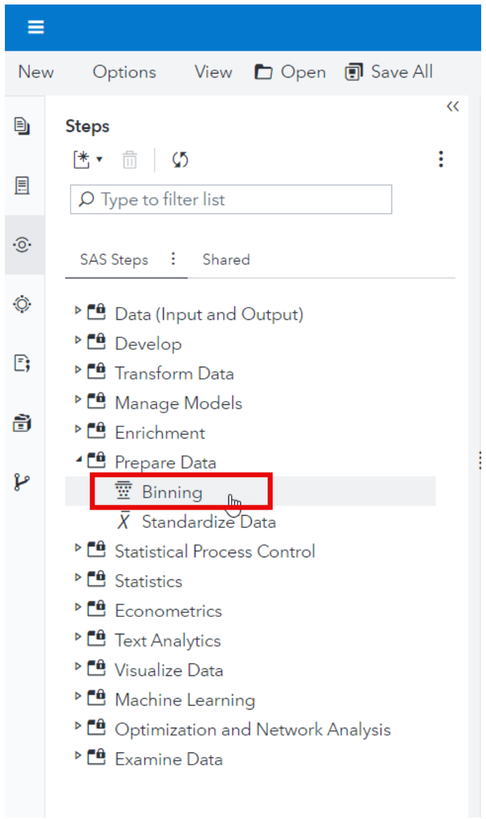
Right-click on the “Table” node you created and select “Run to Node.” After the node is run, click again on the “Table” node, and in the options below, select “Preview Data.” Check that all transformations were performed as expected.

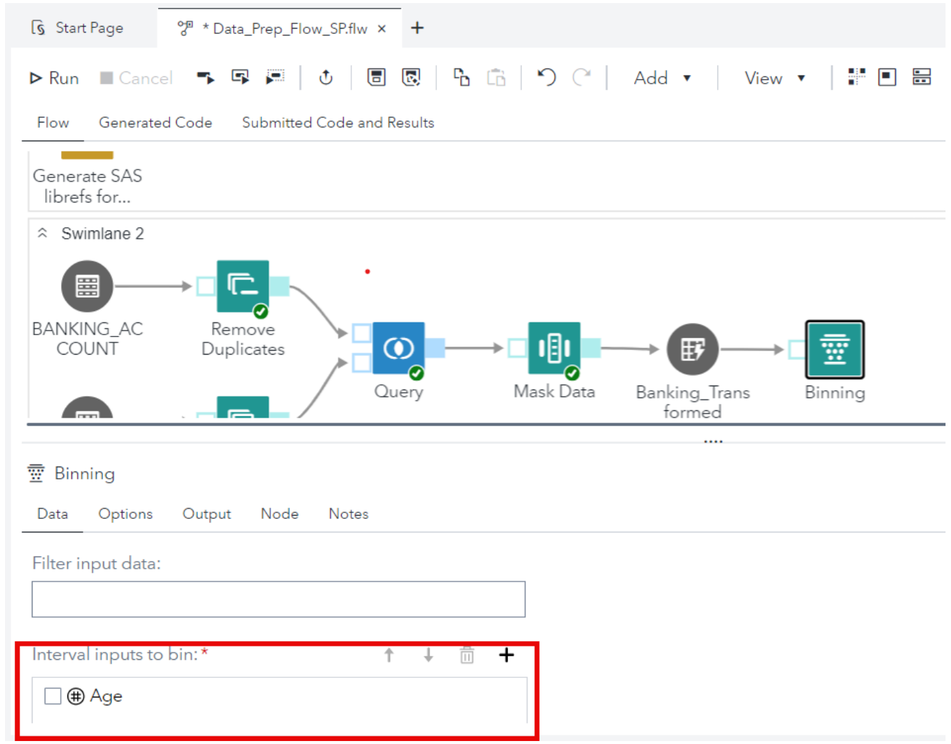


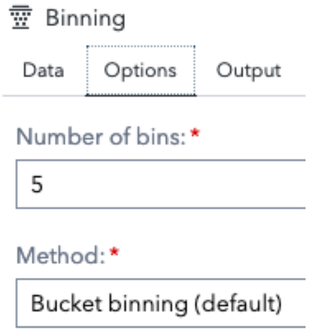


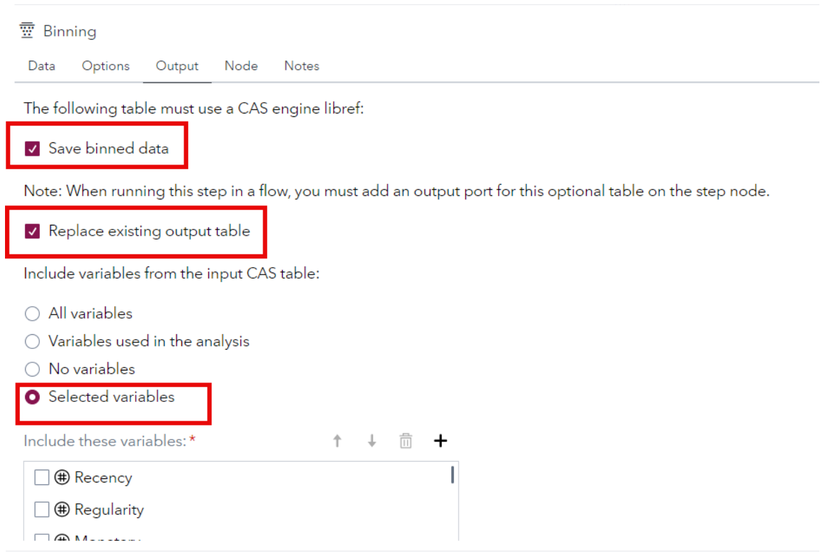
Bin Data

- Now that we've removed duplicates, joined our tables and masked private data, let's go one step further in the name of data privacy.
- From the "Steps" tab, drag a "Binning" node to our canvas and connect the "Banking Transformed" node with it as you see on the right.
- Add "Age" as a variable to be binned. This way, we de-identify the ages of our customers, but we can still use the binned info in modeling later if we wish.
- In the options, select "5" as the number of bins.
- In the "Output" tab, select "Save binned data," "Replace existing output table" and use the "Selected variables" option to get all variables in the output data except the "Age" variable, which will be replaced with its binned version.



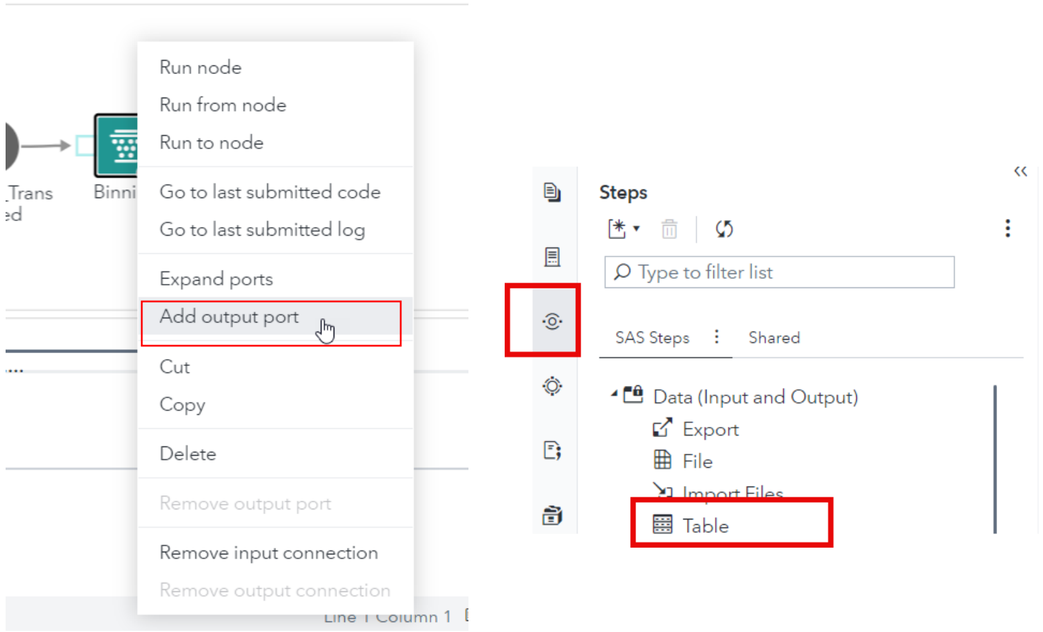


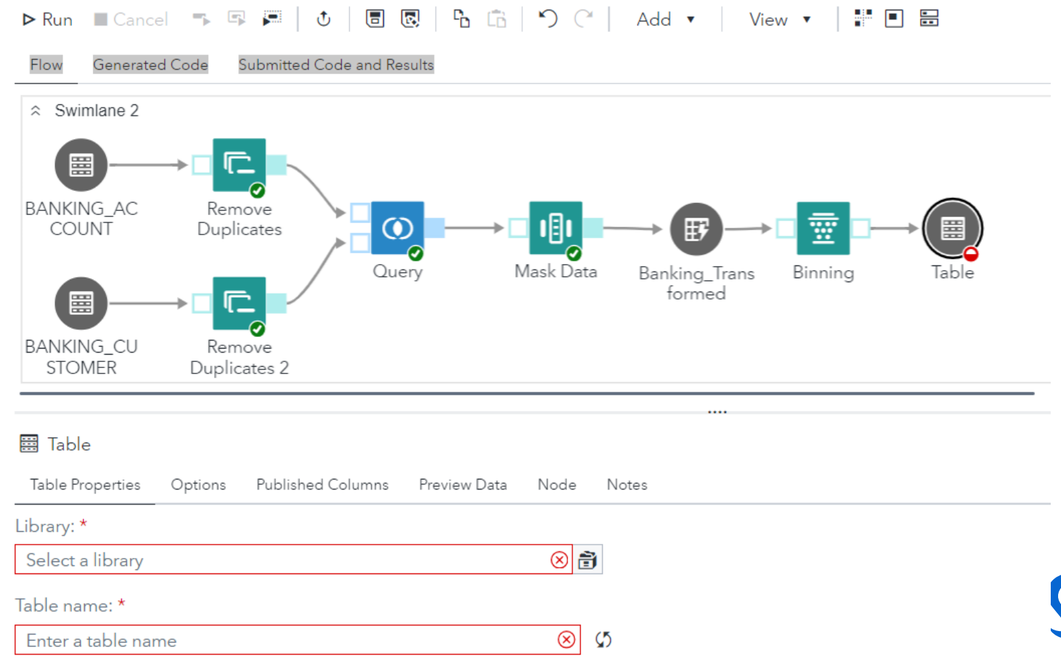




Bin Data

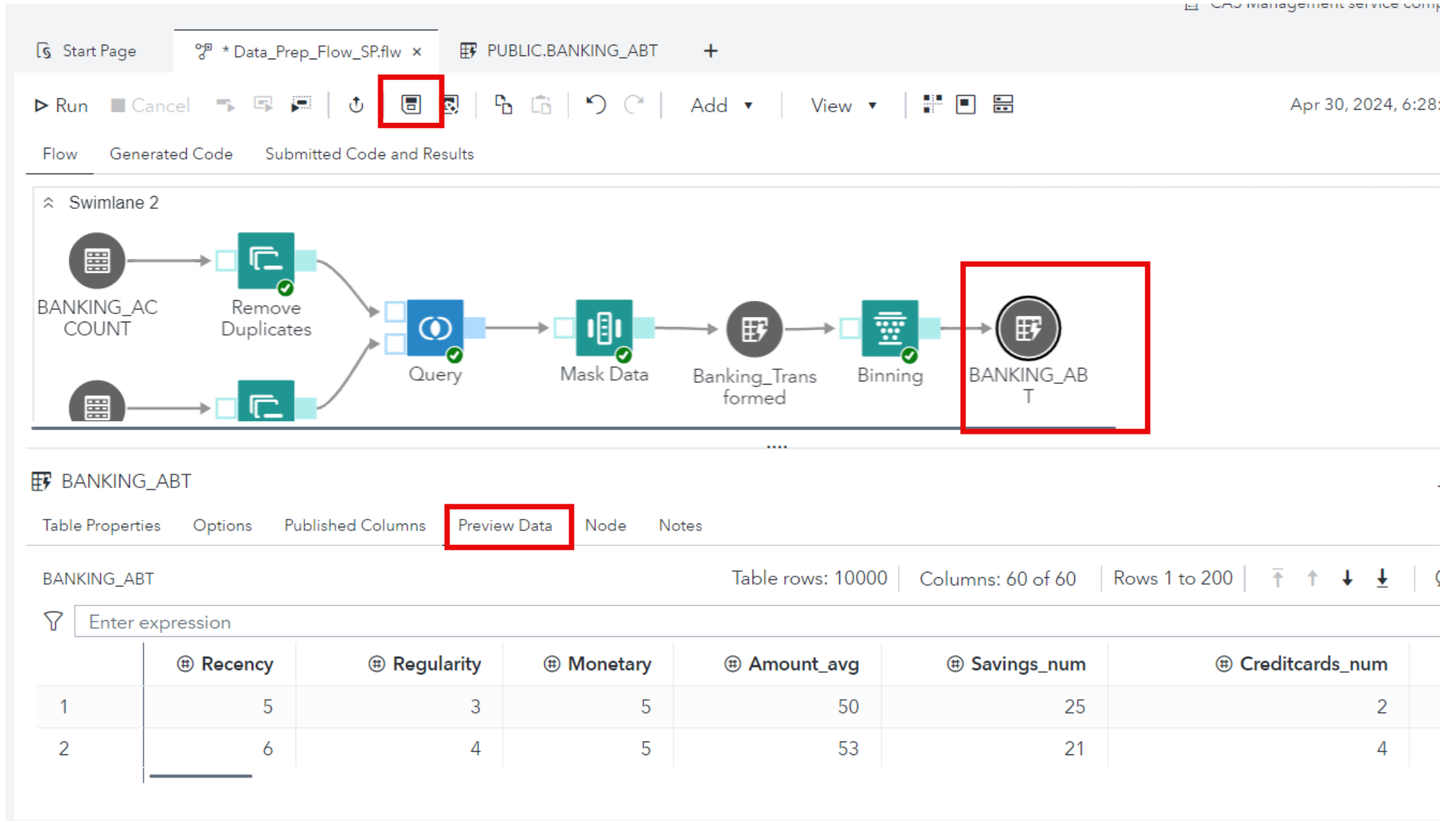
- Now we want to make sure that we save the final output to our ABT table, which is the table the Data Scientist will use.
- To do that, right-click on the “Binning” node and select “Add output port.”
- Then grab another “Table” node from the “Steps” menu.
- Connect the “Binning” node to the “Table” node as you see in the graph.
- Since this is the final table, we are going to use for modeling, use the “Table” node’s options to set the “Library” to “Public” and name the table “BANKING_ABT.” Click OK.





Bin Data

- Right-click on the “Table” node in your flow and select “Run to Node.” Now in the “Table” node you added last, “BANKING_ABT,” you can preview your data and make sure everything is OK.
- Save your flow by using the “Save” icon or use “Ctrl” and “S” to save as you would save a document.
- Your work as a Data Engineer is completed.



Thank you!