Cognos Data Module

What we are going to build

CustomerLoyaltyProgram.csv

Customer Churn.xlsx”

Customer Claim Analysis.xlsx

For this exercise you wish to combine existing Customer Loyalty Program information with Customer Churn information and Customer Claim analysis.
You have this information in various files which you can upload directly into Cognos Analytics and blend, cleanse and model for use in your analysis.
For the first upload, you want to include your Customer Churn information for customers who had, or currently have, cell phone plans that are part of your Loyal customers program.

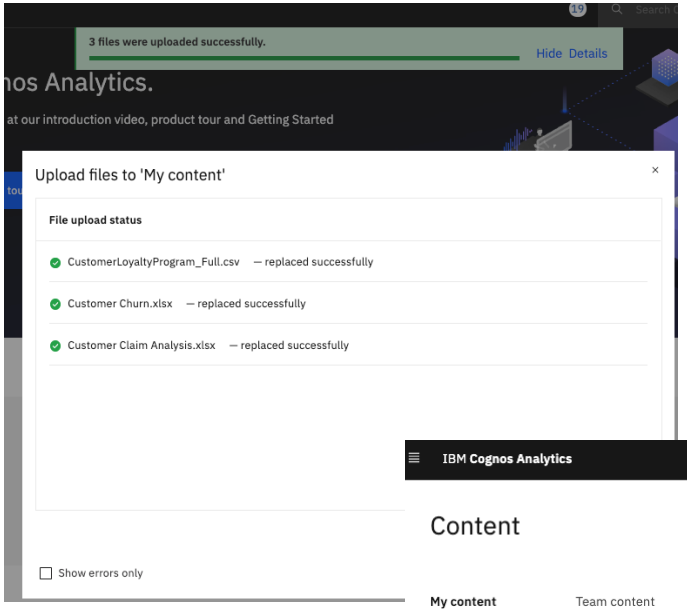
Load data

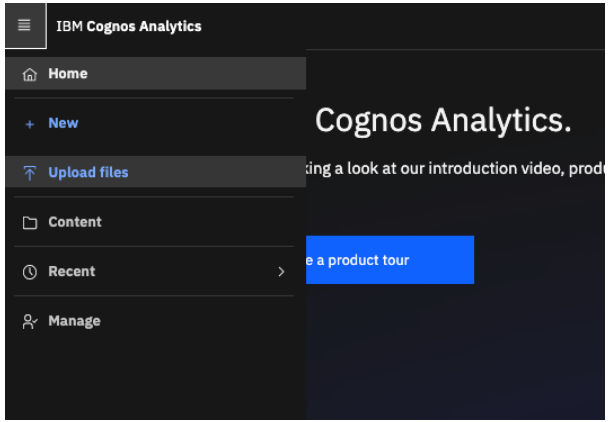
Load the 3 files

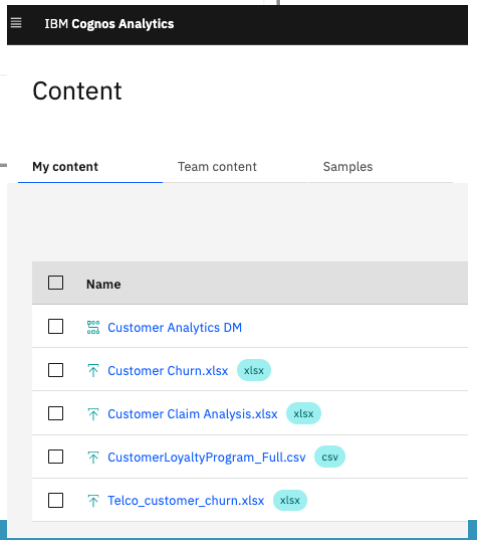
CustomerLoyaltyProgram_full.csv

Customer Churn.xlsx”

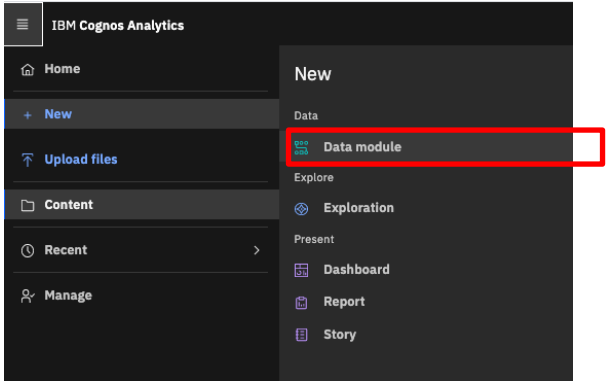
Customer Claim Analysis.xlsx

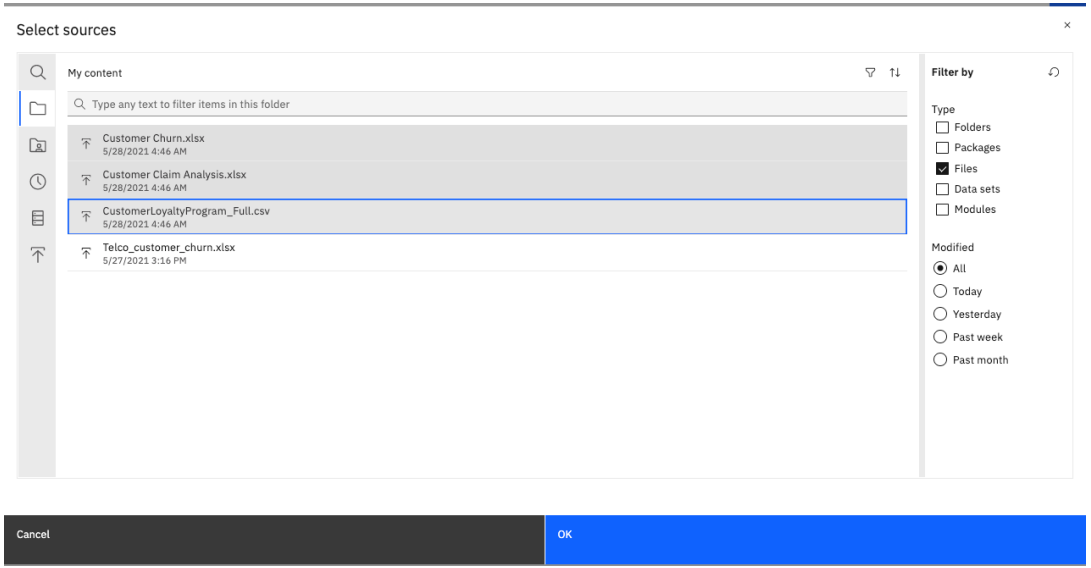






Design the data module



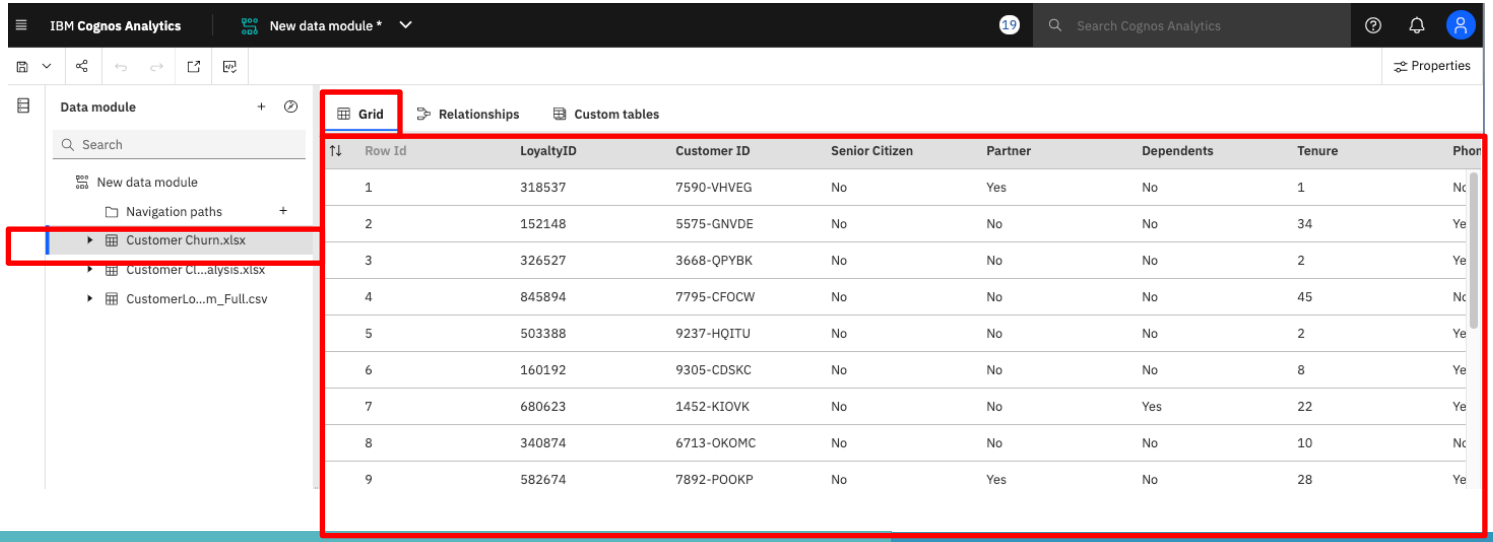


Select the 3 files uploaded

Design the data module: Tables

The data module will open with the Data Source panel on the left, and the canvas on the right. At the top left of the canvas, you will see that there are three views available with data modules:

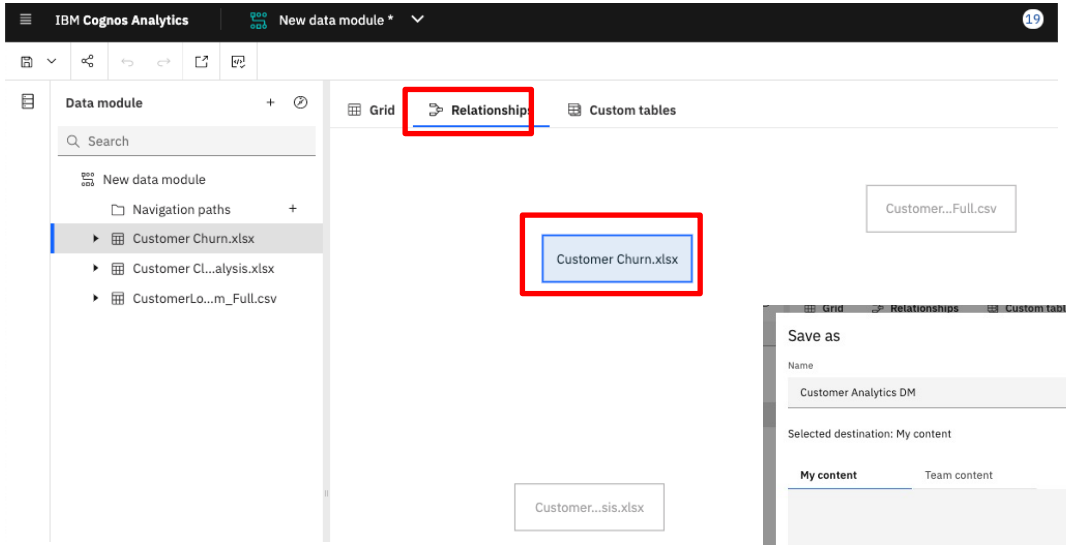
- **Grid** shows the data values.
- Relationships provides a visual representation of how the tables are related.
- Custom provides a visual breakdown of how the different views and tables were created and their individual components.



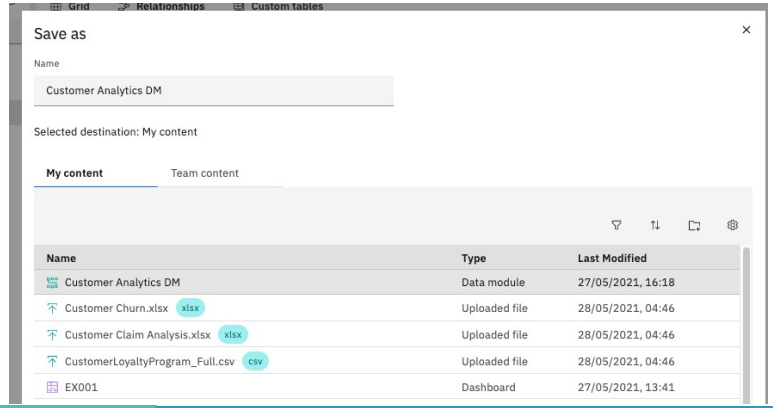
Row Id	LoyaltyID	Customer ID	Senior Citizen	Partner	Dependents	Tenure	Phor
1	318537	7590-VHVEG	No	Yes	No	1	Nc
2	152148	5575-GNVDE	No	No	No	34	Ye
3	326527	3668-QPYBK	No	No	No	2	Ye
4	845894	7795-CFOCW	No	No	No	45	Nc
5	503388	9237-HQITU	No	No	No	2	Ye
6	160192	9305-CDSKC	No	No	No	8	Ye
7	680623	1452-KIOVK	No	No	Yes	22	Ye
8	340874	6713-OKOMC	No	No	No	10	Nc
9	582674	7892-POOKP	No	Yes	No	28	Ye

Design the data module: Tables

Click the **Relationships** view. You will now see a diagram showing each of the three data files brought into the data module. Notice that Customer Churn is highlighted as it is the one that is currently in focus.

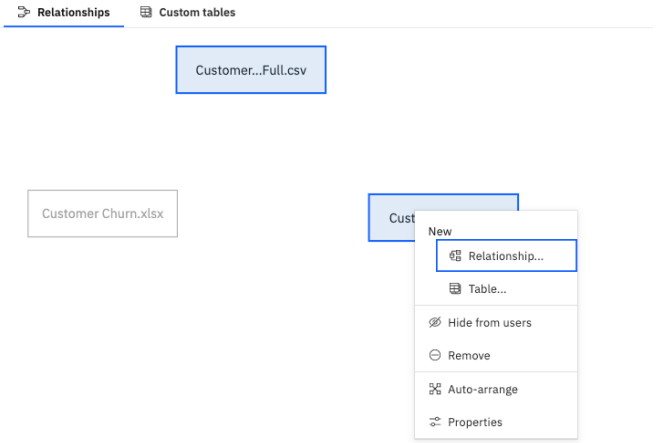


Save it with the name: Customer Analytics DM

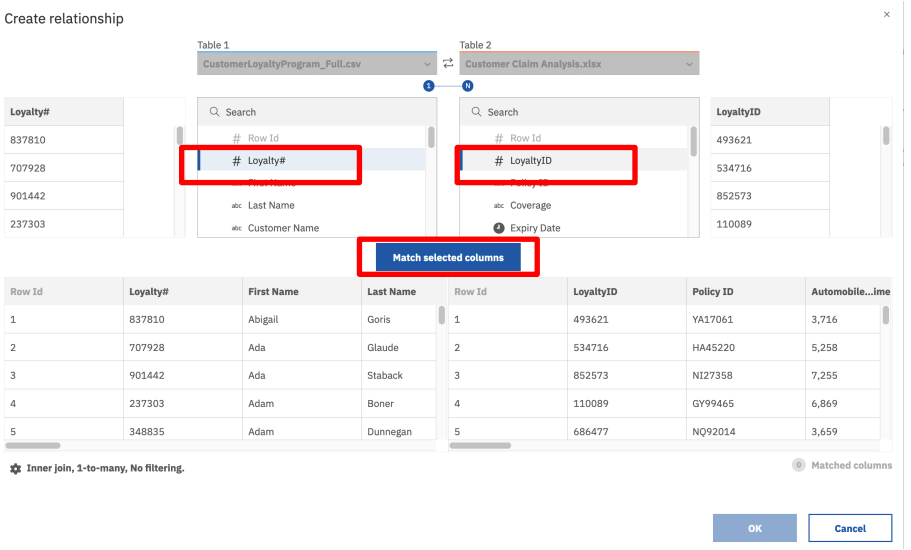


Design the data module: Joins

You will create two join relationships between the imported objects. The first relationship you will define is between “CustomerLoyaltyProgram” and “Customer Claim Analysis”. From the Relationships diagram, select “CustomerLoyaltyProgram” and then control click “Customer Claim Analysis”. Both tables will be highlighted to indicate they are in focus. Right click and select Relationship.

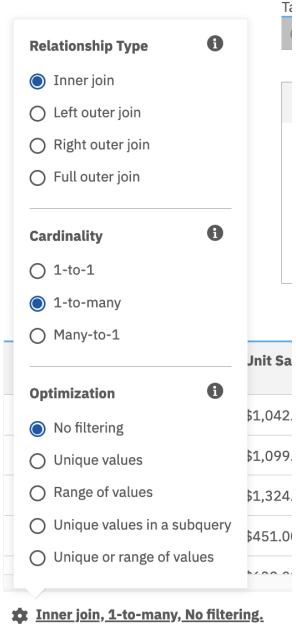


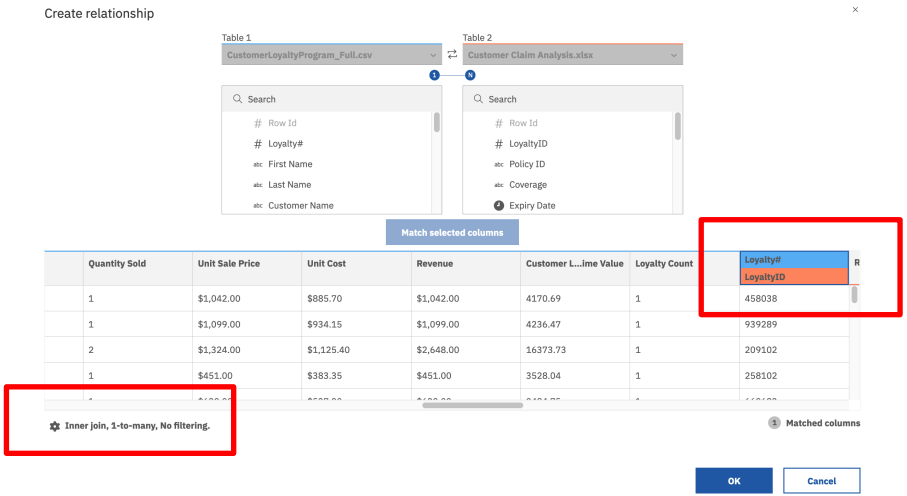
From the Create Relationship window, you can see that the common data between the files is the Loyalty #/Loyalty Id. Using the pick lists, from “CustomerLoyaltyProgram”, select Loyalty# and from “Customer Claim Analysis”, select LoyaltyID



Design the data module: Joins

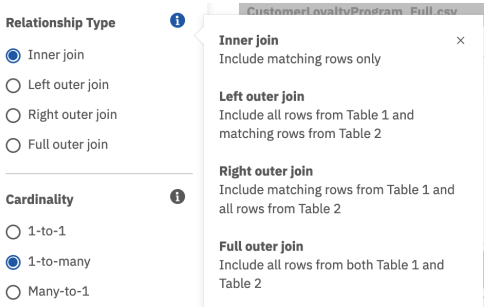
Click the Match Selected Columns button, then Refresh button to update the viewer to show the joined tables. Use the scroll bar to see the common field used for the join is now highlighted.
The Relationship Definition Join settings in the lower left corner show the rules (logic) setup for the join behavior. Click on the Join settings to open the join definitions

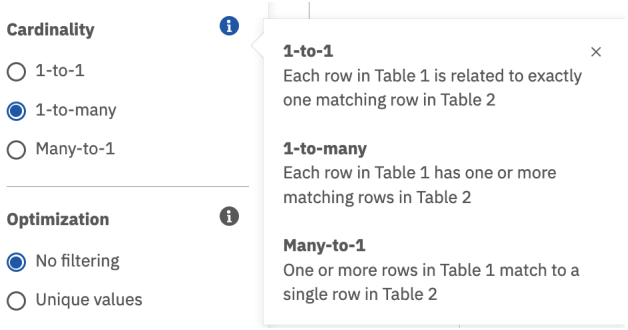




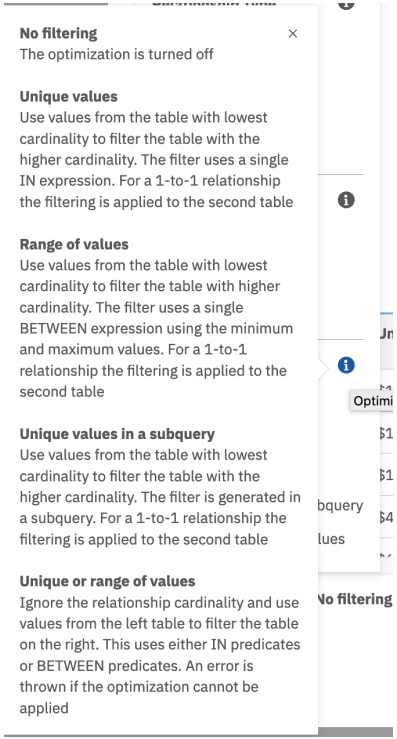
Design the data module: Joins

There are many ways to join data between files. The goal of the join is to ensure that relationship between the two files creates a unique record. For Relationship, this join is set to “Inner Join”. This will Include “matching rows only” between the two files, meaning it will only return records that exist in both files. Make no changes to the Relationship Type.



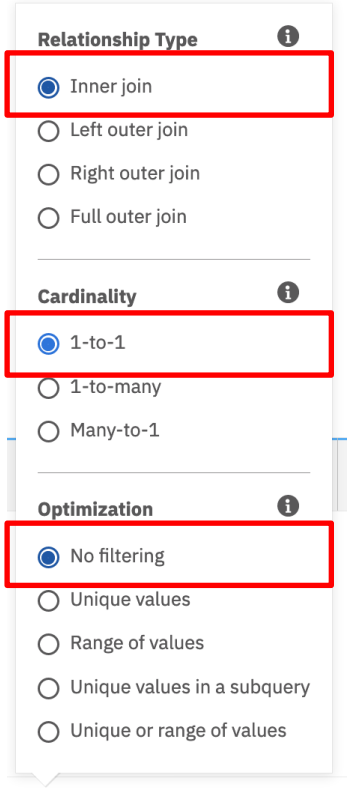


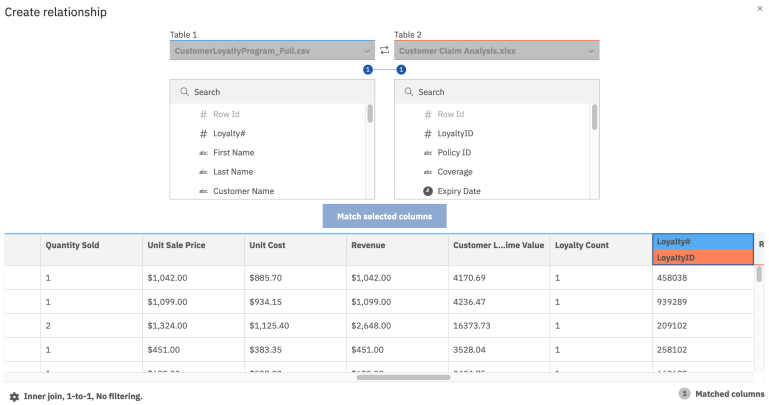
Cardinality refers to number of occurrences of the data item in each of the data sets. For this join, each of these tables is a master list of products and names respectively, so each record is unique. Click on the radial button next to “1-to-1” to change the cardinality.

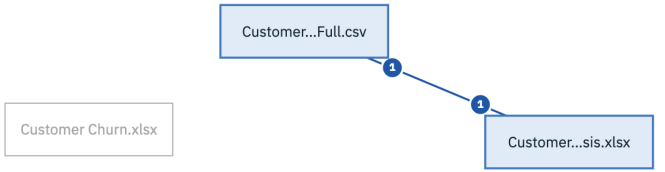


Optimization provides automated filtering options based on the values between the tables. Make no changes to the Optimization.

Design the data module: Joins





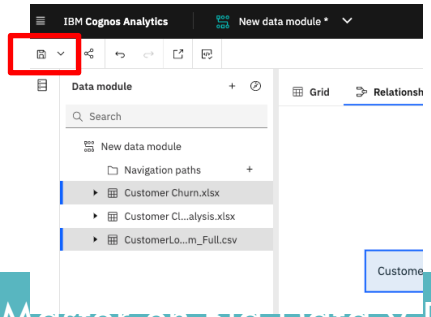


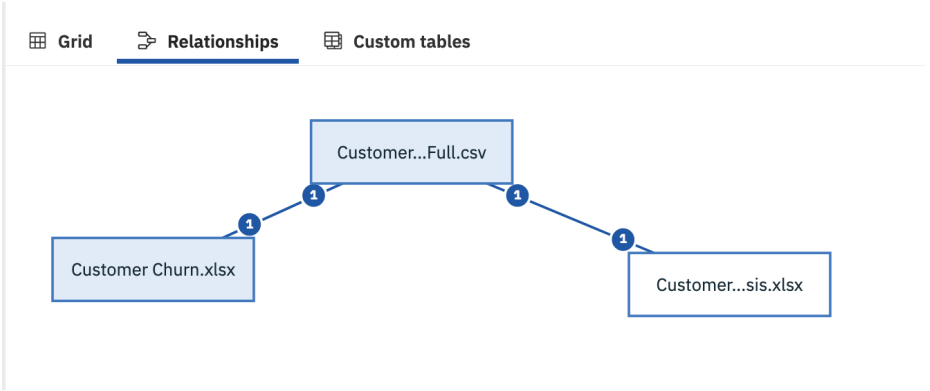
Design the data module: Joins

The next relationship you will define is between “CustomerLoyaltyProgram” and “Customer Churn Analysis”. Select “CustomerLoyaltyProgram” and then control click “Customer Churn”. Both tables will be highlighted to indicate they are in focus. Right click and select Create Relationship.

Repeat the previous steps to create a join. With the Customer Loyalty Program table on the left and Customer Churn table on the right, set the relationship settings as follows

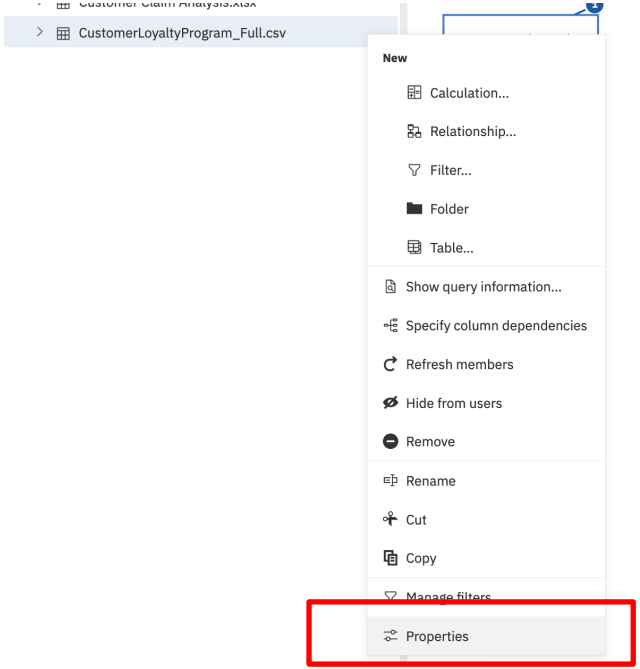
- Create a join relationship using the Customer Loyalty Program “Loyalty #” and the Customer Churn “Loyalty Id”
- Set Relationship type to “Inner Join”.
- Set Cardinality to “1-to-1”.
- Make no changes to Optimization



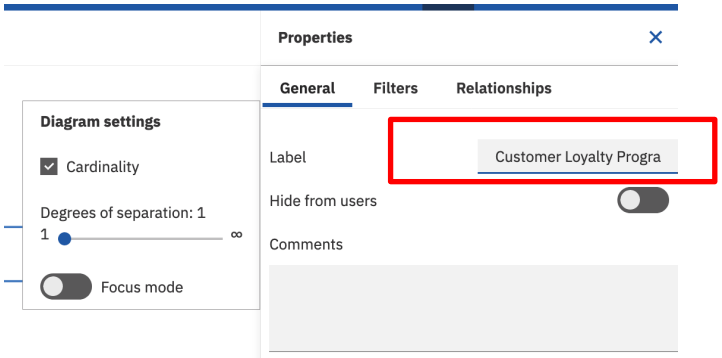


Design the data module: table properties

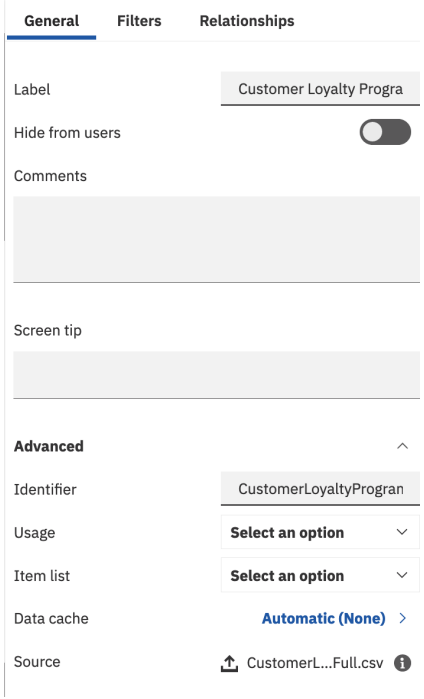
You can modify the Properties of the data tables for better usability for your Business Users working with the data module; such as table names, adding comments and screen tips and data usage type, etc.



The Properties panel opens to the General tab. From here you can change the Table Name (label), choose to show/hide the table to Users, add Comments and Screen tips for the table. Change the Label to "Customer Loyalty Program". Notice that as you change the Label, the name shown in the Data Module diagram to the left is dynamically updating to reflect the new label.

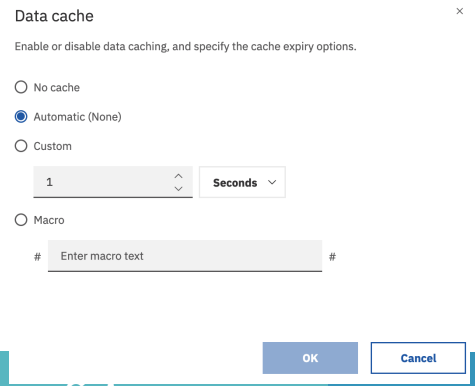


Design the data module: table properties

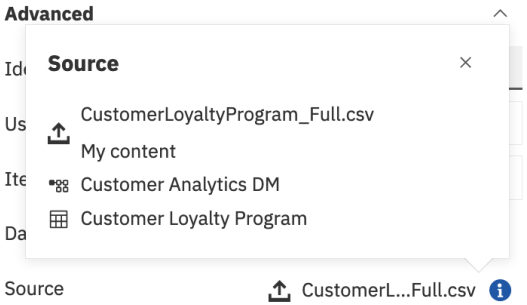


Usage: Allows you to set the Usage for the table:
- Automatic. Specifies that the query engine will detect if the query subject is a dimension or a fact.
- Bridge. Specifies that the query subject refers to a bridge table.
- Summary. Specifies that the query subject will be treated as a summary query

Data Cache. You may use the Data Cache rules to enable or disable data caching and specify the cache expiry options.

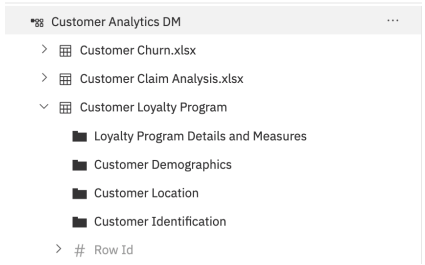


Source. The Source provides the User with lineage information

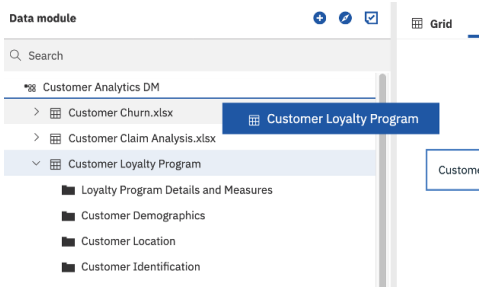


Design the data module: table Folders

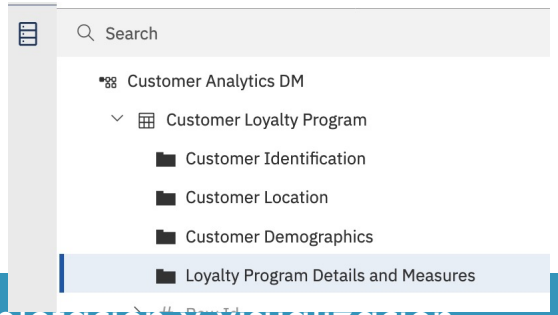
Right click on “Customer Loyalty Program”. Under New, select “Folder, Rename to “Customer Identification”
Repeat these steps three more times to create the following folders:
- "Customer Location"
- "Customer Demographics"
- "Loyalty Program Details and Measures"



Reorder the tables, where “Customer Loyalty Program” is above “Customer Churn”. Click on “Customer Loyalty Program” and drag and drop it above “Customer Churn”. A blue bar will indicate the drop zone.

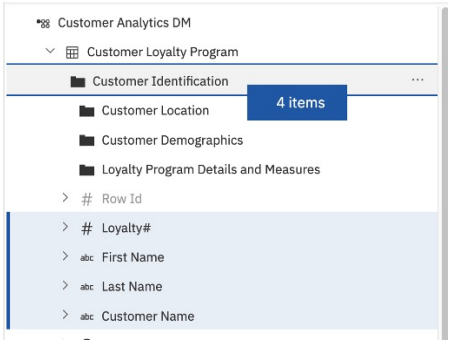


Reorder the newly created folders by dragging and dropping them into the following order:
- Customer Identification
- Customer Location
- Customer Demographics
- Loyalty Program Details and Measures.

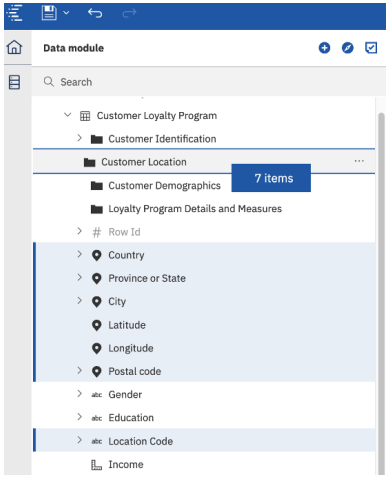


Design the data module: table Folders

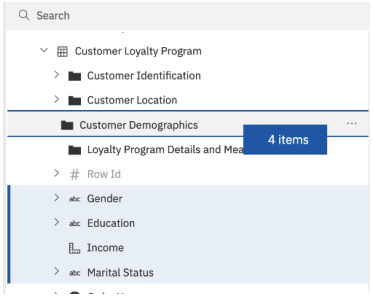
From the Customer Loyalty Program table, use control + click to multi-select columns "Loyalty#", "First Name", "Last Name", and "Customer Name". Drag them into the “Customer Identification” folder. Blue bars above and below the table name will indicate the drop zone.



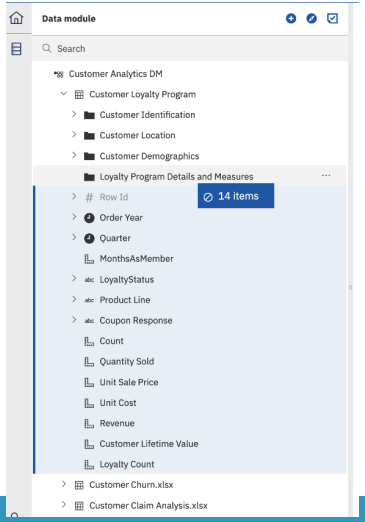
From the Customer Loyalty Program table, multi-select columns "Country", "Province or State", "City", "Latitude", "Longitude", "Postal code" and "Location Code". Drag them into the “Customer Location” folder



From the Customer Loyalty+ Program table, multi-select columns "Gender", "Education", "Income" and "Marital Status". Drag them into the “Customer Demographics” folder.

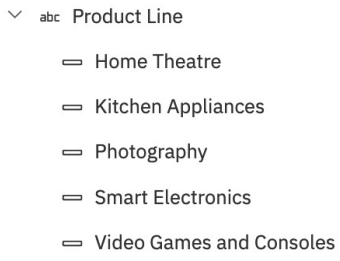


Notice that as the data items were organized into the folders, they no longer appeared in the original table list. From the Customer Loyalty table, multi-select all remaining data items. Drag them into the “Loyalty Program Details and Measures” folder

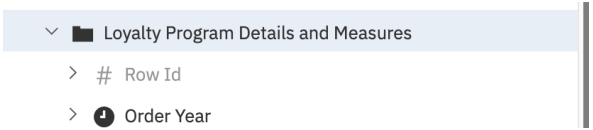


Design the data module: Data properties

As you look at the data items listed, you will notice many of the data items have Expand arrows next to them. This allows the User to fields in their metadata tree to see individual values or members. From the Customer Loyalty Program table, click on the Expand arrow next to Product line. Click again to collapse the member tree.



Hidden fields. Notice Row id is greyed out. This indicates the item is hidden from the report authors view.

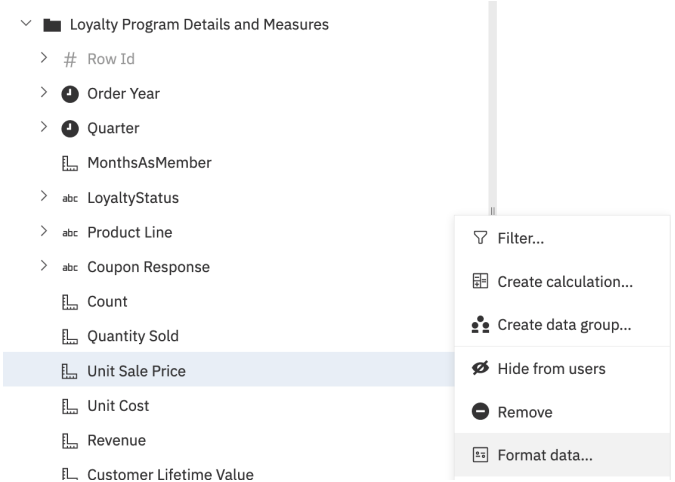


Usage Type. The icons next to the field names indicate the data type setting.

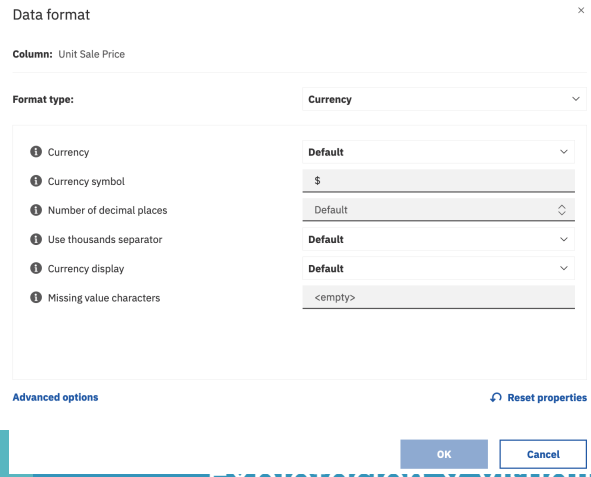
- Numeric fields. The '#' sign next to an item indicates it's a Numeric field used as an identifier.
- Alpha-numeric fields. The 'abc' sign next an item indicates it's an Alpha numeric field used as an identifier or attribute.
- Location. The pin indicates the field is a location dimension which may be used for geospatial mapping.
- Measures. The 'ruler' next to an item indicates it's a measure.
- Time. The clock indicates the field is a time dimension.

Design the data module: Data properties

Similar to the options you saw earlier for tables, data items also have many options to modify their Properties. Expand the Loyalty Program Details and Measures folder if needed. Click on Unit Sale Price, then the ellipse (...) and select Format data.



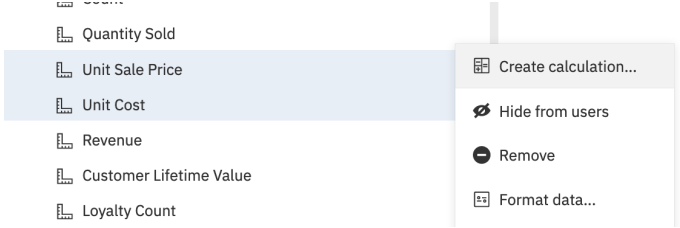
The Data format dialog box opens. Click the down arrow next to Format Type. Verify that the Format Type shows Currency. The Format type setting is based on Cognos Analytics having recognized a data format previously set in the uploaded file. A list of additional formatting settings for Currency are presented. Users are also able to assign a value for Missing value characters.



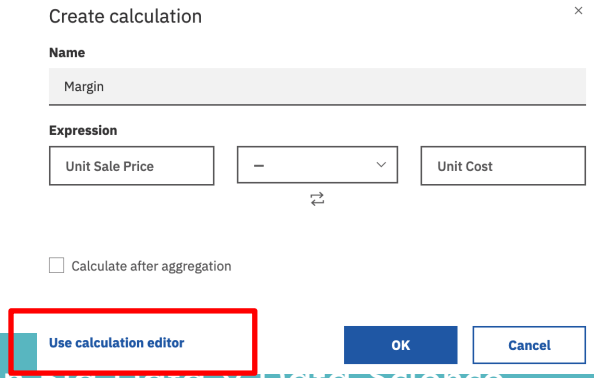
Verify: Unit Cost and Revenue is set to Currency

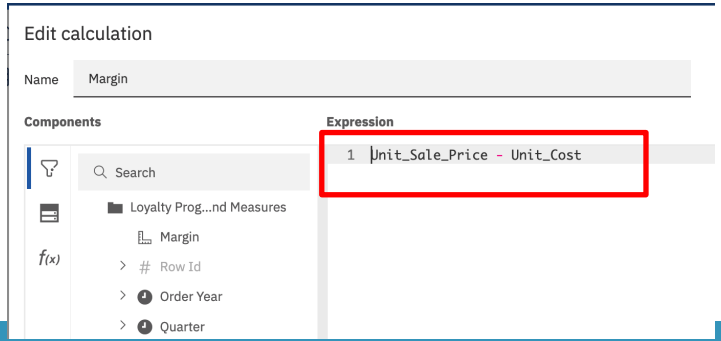
Design the data module: Calculations

Your data has Unit Sales Price and Unit Cost, but it would be helpful to have the Margin calculated so it is available for analysis. By adding this calculation to the data module, it will be reusable anywhere in the analysis. Users will not need to rebuild the calculation each time they need them in their dashboards, reports, etc., they simply use the calculated measure



The create Calculation dialog box appears. This dialog box allows the User to create a simple calculation. Users can select two fields and select from the basic operators for addition, subtraction, multiplication, division, percentage and percentage change calculations. Rename the calculation Column name to Margin. Use the pull-down arrow to select the Subtraction operator.





Design the data module: Calculations

The Expression editor provides many capabilities to make it easy for you to create, document, evaluate and test your expressions

- **Information**. Provides the User with additional context and assistance.

- **Preview**. Provides a sample rendering of the calculation results along with theExecution time to run. Click Preview. Results appear in the lower right window.

- **Validate**. Validation allows Users to test their calculation to ensure no errors occur.Click Validation. Results appear in the lower right window.

- **Comment**. Allows Users to insert comments within the calculation itself; to make notations throughout the expression. Click Comment. Notice that the calculation is greyed out and preceded with the familiar "//" used in SQL. (Queue cheers from the SQL Writers!!) Click Comments again to uncomment calculation.

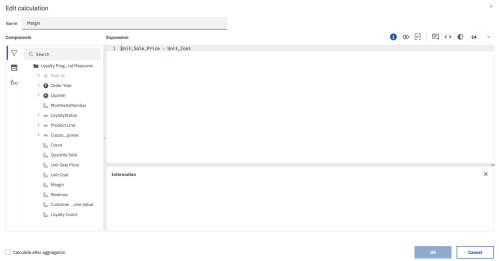
- **Prettify**. Restructures calculation expression into a more readable format for Users.E.g. When a CASE WHEN expression is in one long block of text that wraps across lines, Prettify will clean it up and break it out into shorter segments with each condition clause on a separate line.

- **High Contrast Mode**. This mode reverses the color scheme in the expression editor. This provides black background with light text some Users prefer. Click High Contrast. Click again if you wish to restore the original background.

- **Font**. The default font size for the expression editor is 14. Users may increase or decrease the font size used.

Design the data module: Calculations

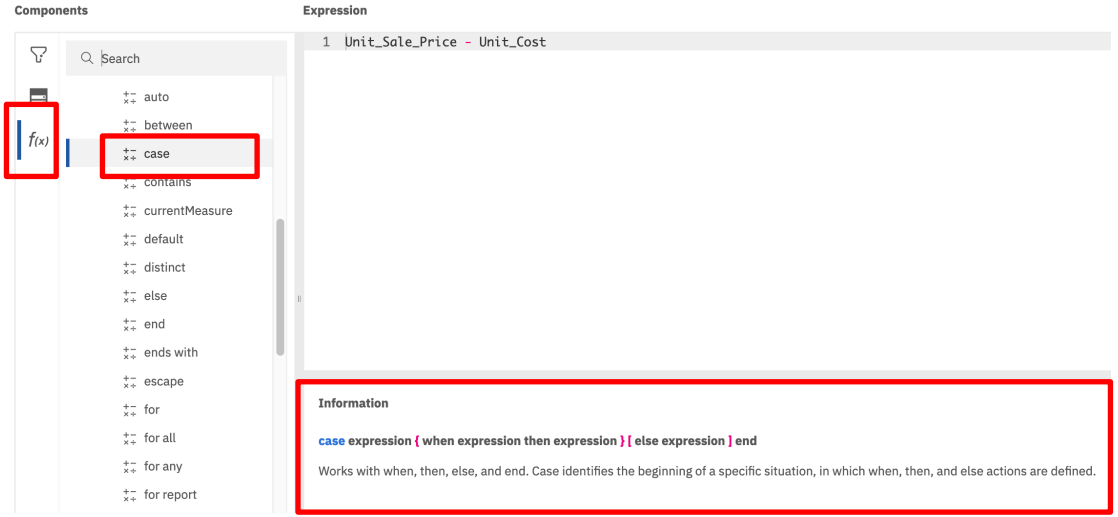
The Expression editor provides many capabilities to make it easy for you to create, document,
evaluate and test your expressions



- **Information**. Provides the User with additional context and assistance.
- **Preview**. Provides a sample rendering of the calculation results along with theExecution time to run. Click Preview. Results appear in the lower right window.
- **Validate**. Validation allows Users to test their calculation to ensure no errors occur.Click Validation. Results appear in the lower right window.
- **Comment**. Allows Users to insert comments within the calculation itself; to make notations throughout the expression. Click Comment. Notice that the calculation is greyed out and preceded with the familiar "//" used in SQL. (Queue cheers from the SQL Writers!!) Click Comments again to uncomment calculation.
- **Prettify**. Restructures calculation expression into a more readable format for Users.E.g. When a CASE WHEN expression is in one long block of text that wraps across lines, Prettify will clean it up and break it out into shorter segments with each condition clause on a separate line.
- **High Contrast Mode**. This mode reverses the color scheme in the expression editor. This provides black background with light text some Users prefer. Click High Contrast. Click again if you wish to restore the original background.
- **Font**. The default font size for the expression editor is 14. Users may increase or decrease the font size used.

Design the data module: Calculations, Functions

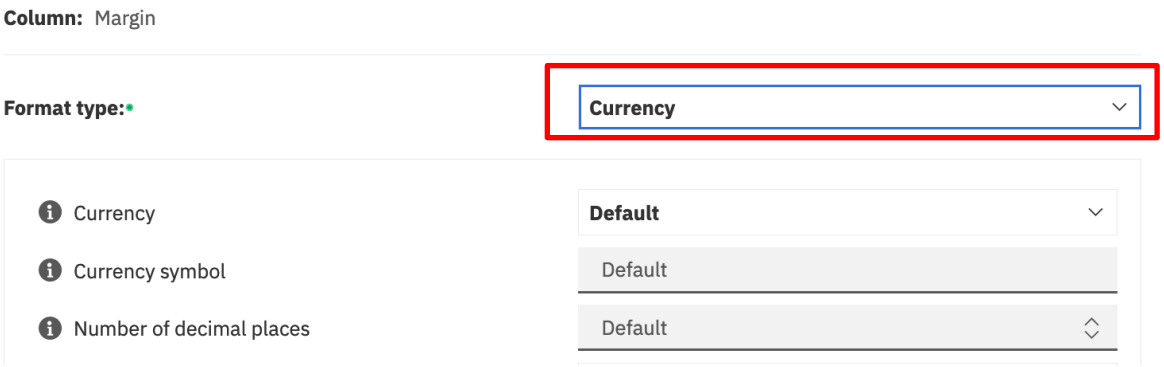
Scroll through the list and select various Operators. Click Information to see the expression syntax for the operator in the bottom right window. See examples below



Comments

Design the data module: Calculations, Functions

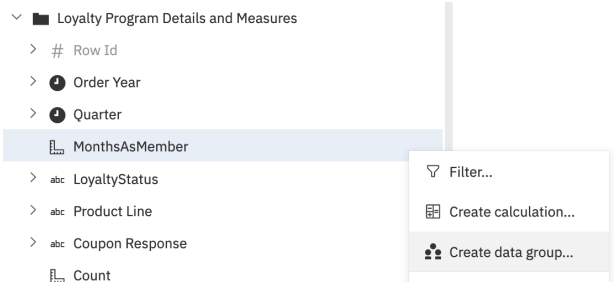
Now you want to Apply the same Currency format as you used for Revenue. From the Customer Loyalty Program Folder select Margin, right-click to open the options menu and select Format data.

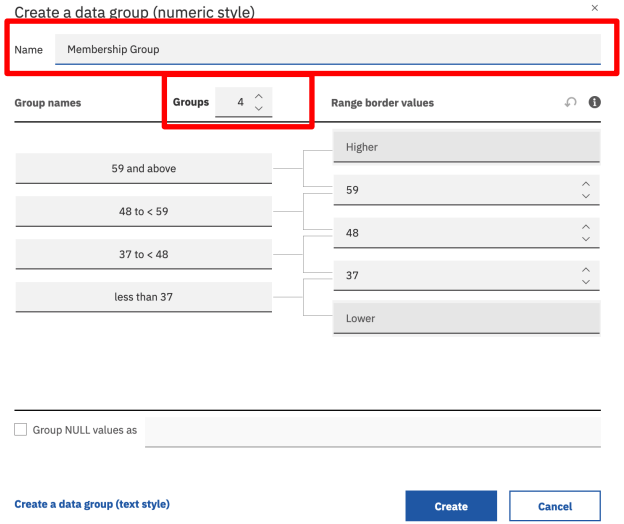


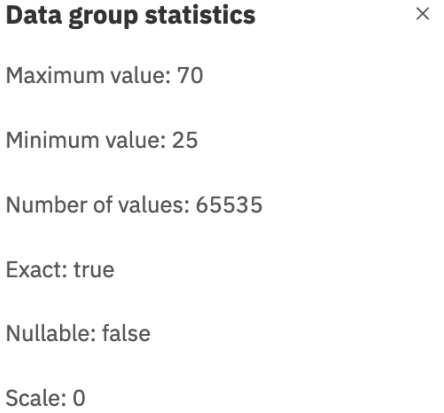
Design the data module: Custom Data Group

Often, Users need to organize data into groups for analysis, sometimes referred to as “binning” or “buckets”. In looking at “MonthsAsMember”, you feel that analyzing membership on each membership duration may not provide much insight. For instance, there may not be a great deal of difference between someone who has been a member for 15 versus 16 months. Rather, you would be more interested in the behaviors of members with similar membership durations. Therefore, you will create custom groupings to “bin/bucket” the individual membership duration into “Months as Member” category groups.

Click column “MonthsAsMember”. Click the ellipse (...), then select “Create data group”.

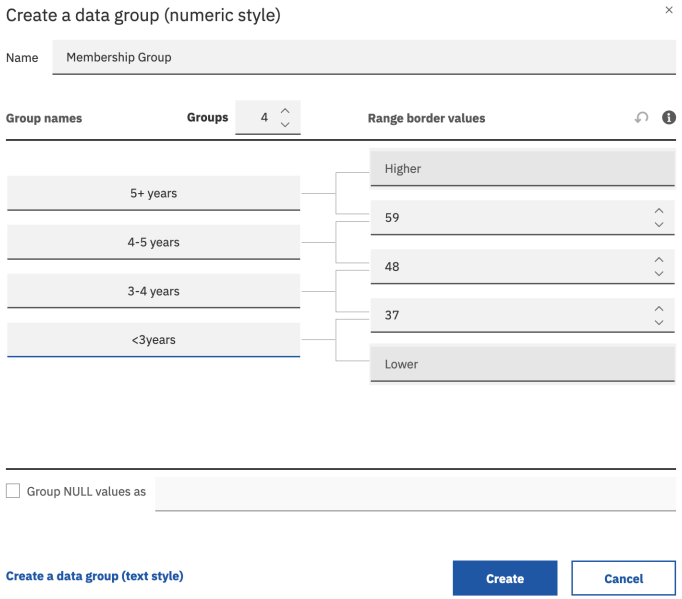




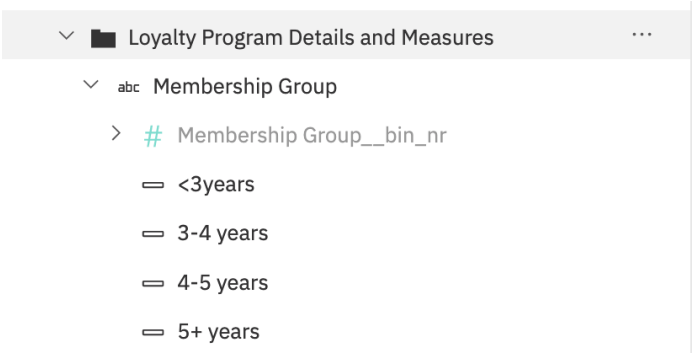


Design the data module: Custom Data Group

You can override the equal distribution to manually define your month groups and customize the group names. The equal distribution is interesting and shows that the distribution is very close to full years. You want to update the groups to monthly cutoffs that correspond to full years for every group. Click in each of the Range border values field and change the Range border values to those shown below. Then, click in the Group names fields to change Group names as follows:

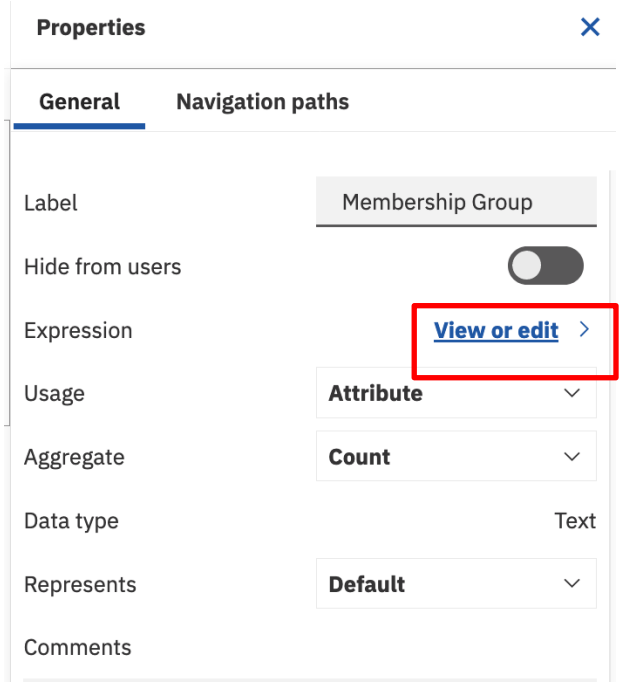


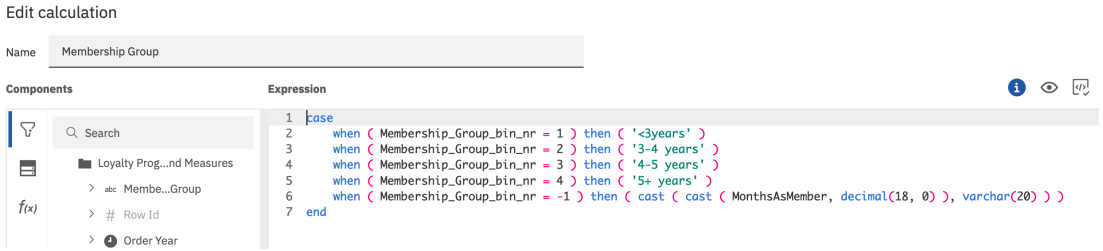
The new Membership Group column now appears in the Loyalty Program Details and Measures table. Click the arrow next to Membership Group. You will now see the Members you created, and a new column named “Membership Group__bin_nr”. Membership Group__bin_nr is an identifier field which may be used for sorting. By default, it is set to hidden so will be greyed out.



Design the data module: Custom Data Group

Click on Membership Group__bin_nr and use the ellipses "..." to open the Properties

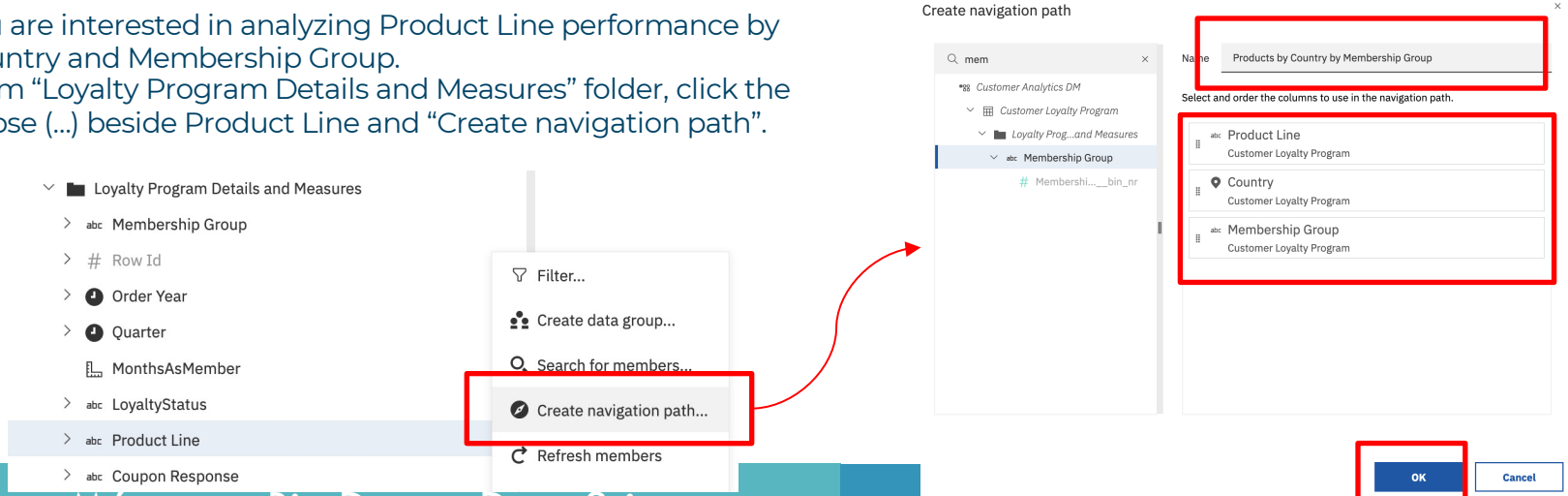




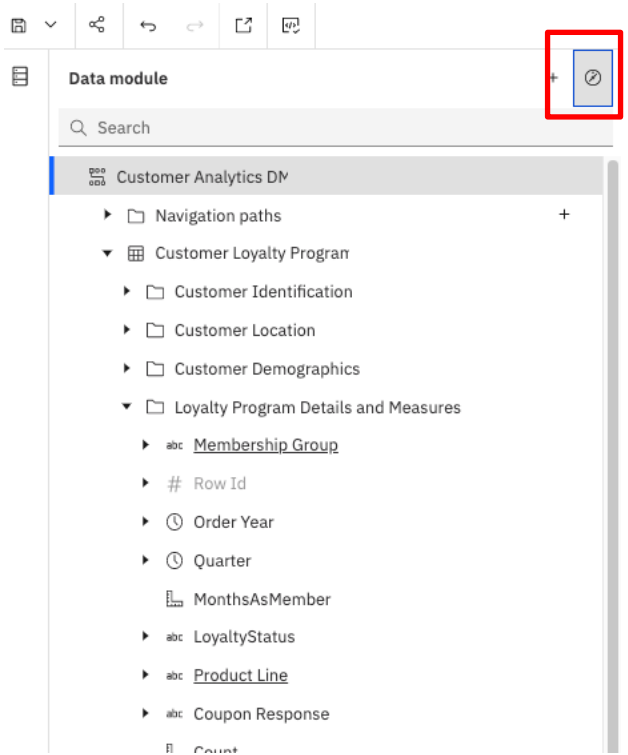
Design the data module: Navigation Path

A navigation path is a collection of non-measure columns that business users leverage for data exploration. Navigation paths can now be defined in a data module or dashboard to help users easily explore and drill down to see their underlying data. These can be “natural” navigation paths that follow a defined hierarchy, or they can be defined to allow users to navigate and drill down in any order that makes sense for their analysis.In traditional BI and OLAP technologies, a drill down action requires a pre-defined hierarchical data structure so that you could navigate the drill down i.e. Year to Month to Day. Navigation paths are much more flexible and can accommodate a drill down path that aligns with the thought process users go through to analyze their business. For your analysis, you need to analyze Product Line performance by Country and Months As Member Categories, so you will create a navigation path that allows you to drill down in your data on that path.

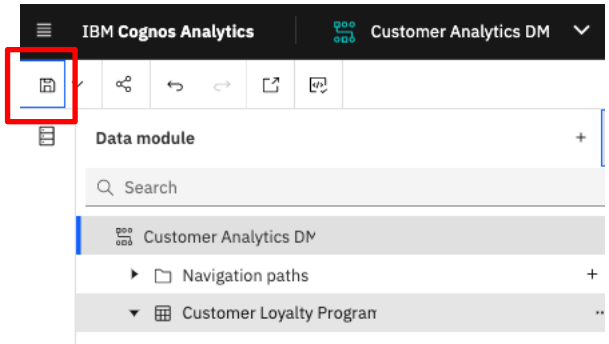
You are interested in analyzing Product Line performance by Country and Membership Group.
From “Loyalty Program Details and Measures” folder, click the ellipse (...) beside Product Line and “Create navigation path”.



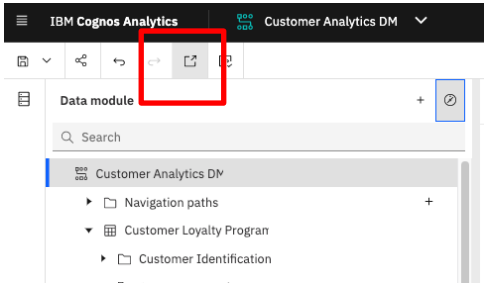
Design the data module: Navigation Path



To identify your Navigation path members in the Data Modules panel, click the Identify navigation path members icon at the top of the Data Module panel.
Each data item used in a navigation group is now underlined to identify its membership.

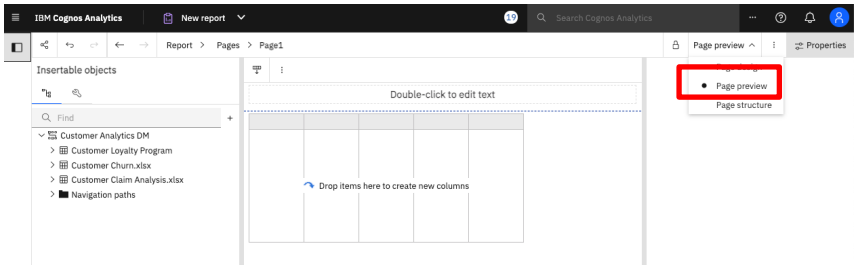


Design the data module: Testing

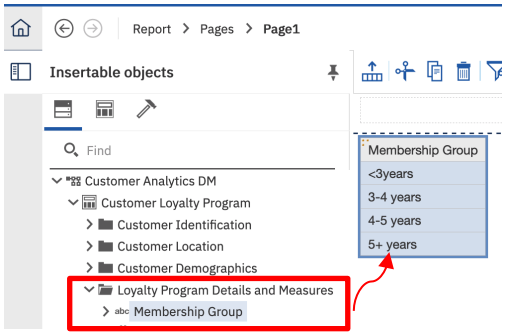


Cognos Analytics will open a new browser tab that will allow for the creation of a list report to validate the new Data Module

To begin, you'll put the report in Preview mode so that you can see the data update as you test the data module. Click the down arrow

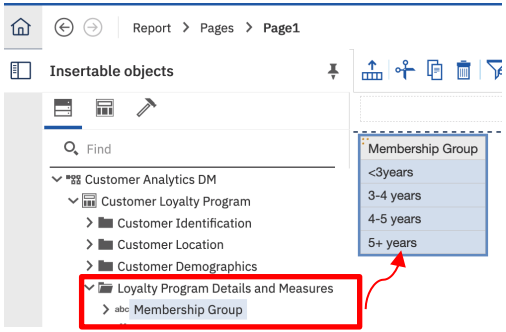


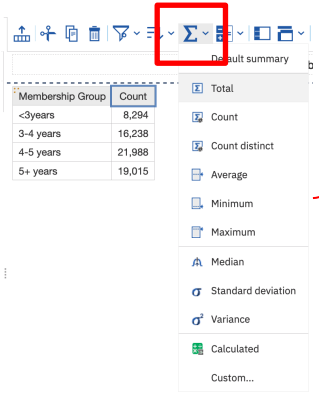
Click on Membership Group and drag it over to the list report, over where it says "Drop items
here to create new columns".

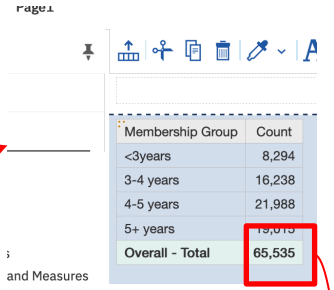


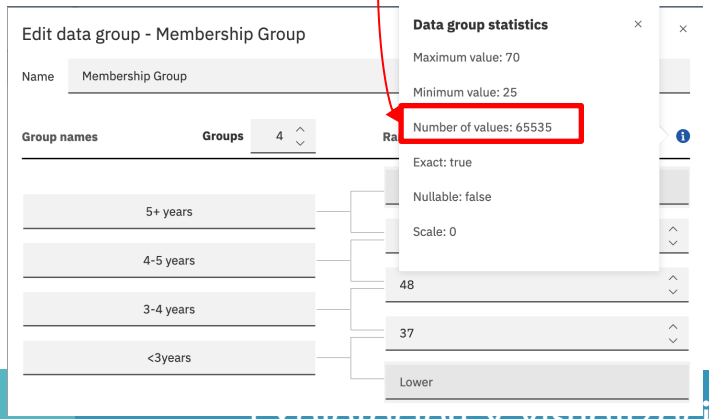
Design the data module: Testing

Drag and drop Count to the right of Membership Group, and Summarize the column Count using Total.









Notice the Overall-Total count matches what you saw earlier in the workshop as the "Number of Values" under Data group statistics when you created the custom data group.
You'll use this count to verify your data module is consistent for reporting as you look at other ways to "slice and dice" your data.