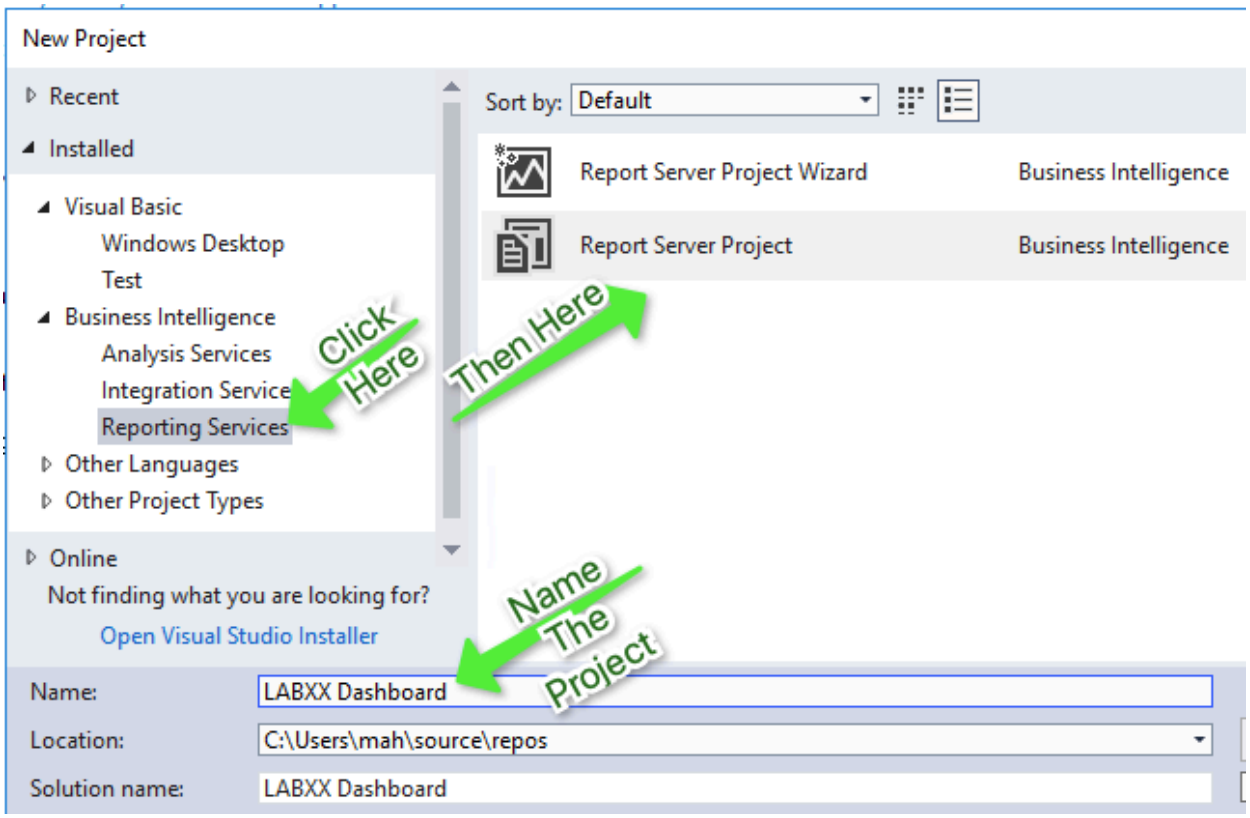
SSRS: Creating Dashboards and Widgets

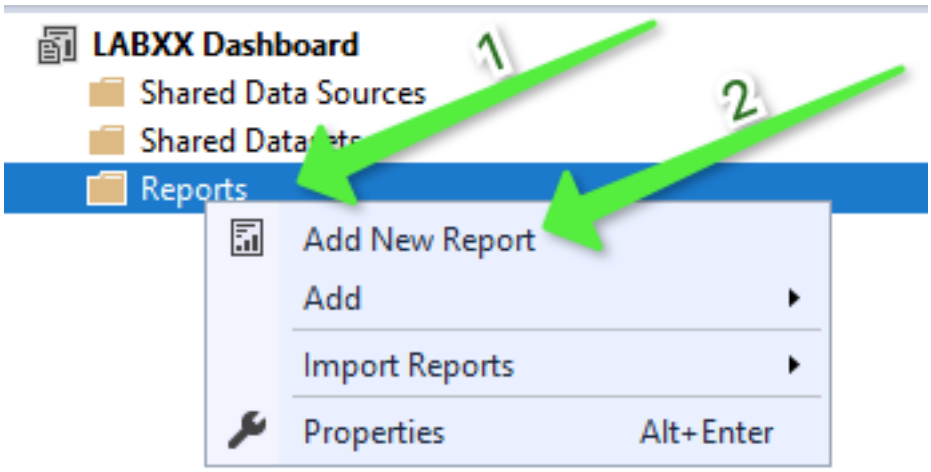
In this lab we will create a company dashboard made up of multiple datasets that paint a picture of the company's health from multiple measurements that we will define. We will also consume data from two different kinds of data sources: SQL server directly and also from SQL SSAS.

1) Start Visual Studio SSDT by double clocking on the purple desktop icon on your terminal server session.

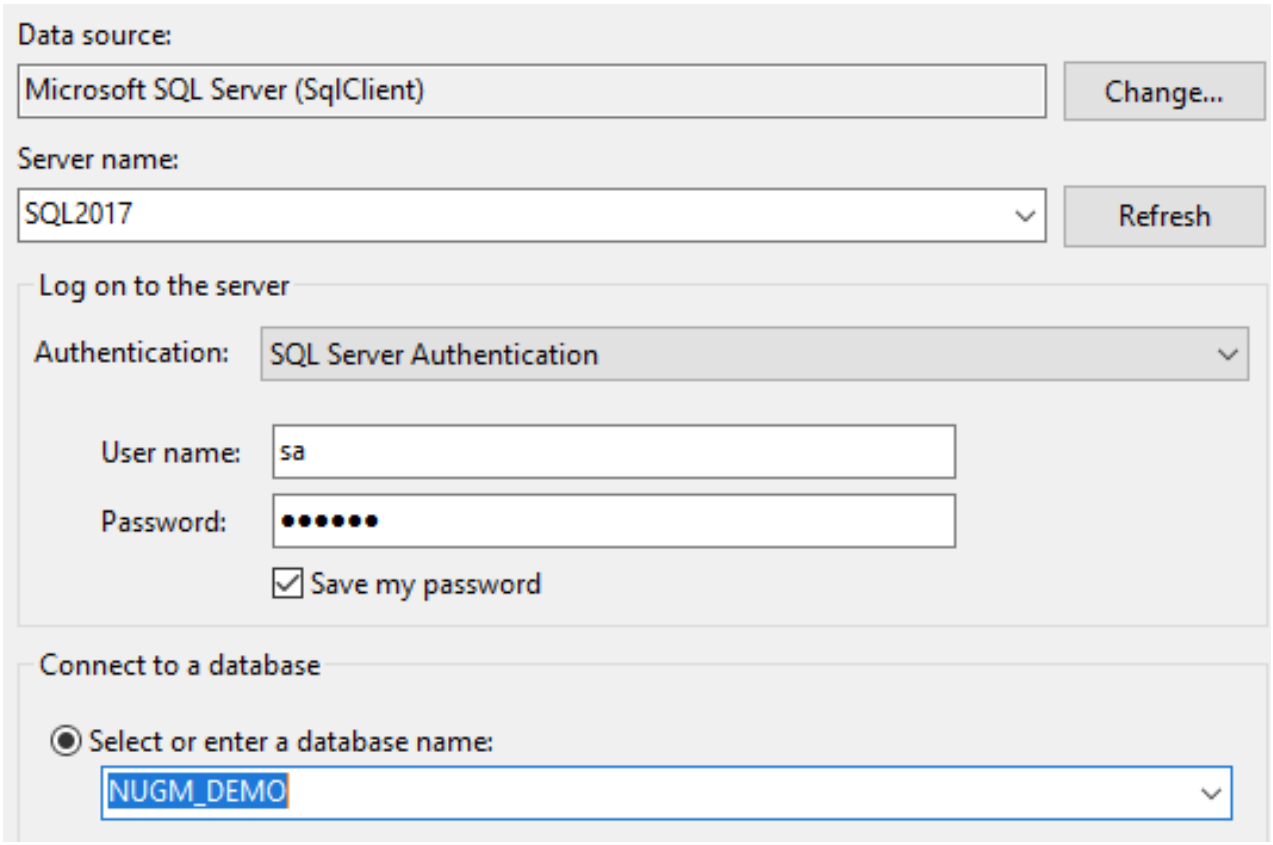
2) Select File/New Project.
3) Choose Business Intelligence / Reporting Services / Report Server Project.
4) Name your project **XXX Dashboard. (Where XXX is – replace this with your LAB user ID.)**



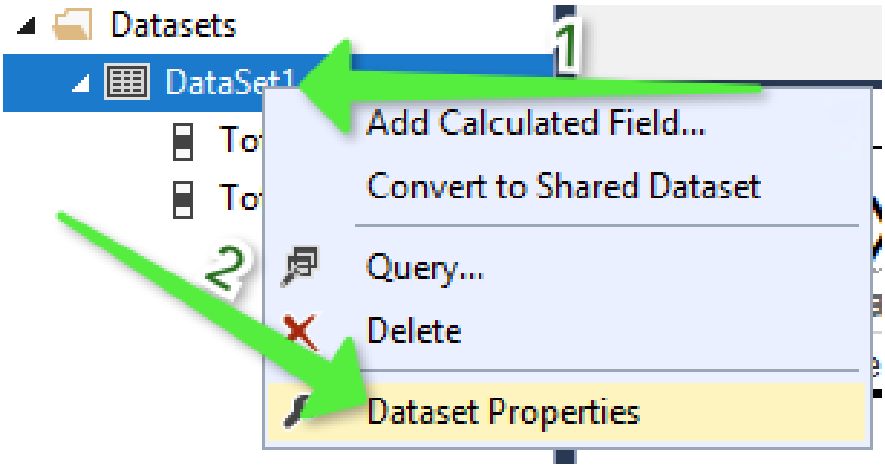
5) Right click on Reports under the LABXX Dashboard Solution, and choose **Add New Report**.



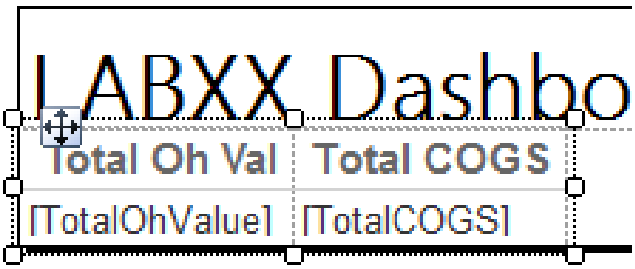
6) Click NEXT to start the wizard.
7) Create a new data source, name it **dsSqlServer**.
8) Click EDIT… to define the datasource as pictured here using the password "2getin".



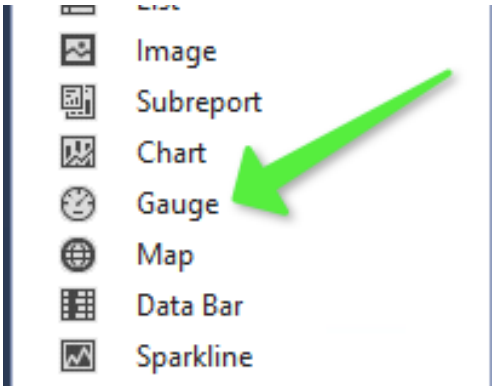
9) We will start with building an Inventory Turns measurement for the dashboard. Let's review the underlying SQL to define two things we need to construct inventory turns:
    a. One Year of COGS
    b. Current Inventory Value at Standard
10) Enter this select statement in the Query Text Box: SELECT * FROM View_Inventory_Turns_Summary
11) Click the Finish button at the bottom to jump to the end of the wizard. (We will manually do the layout in a moment.)
12) Name your report **LABXX Dashboard** (using your Login ID for LABXX) and click Finish again.
13) Rename **DataSet1** to **dsInventoryTurns** so that we will be able to identify it later on when we have multiple datasets.



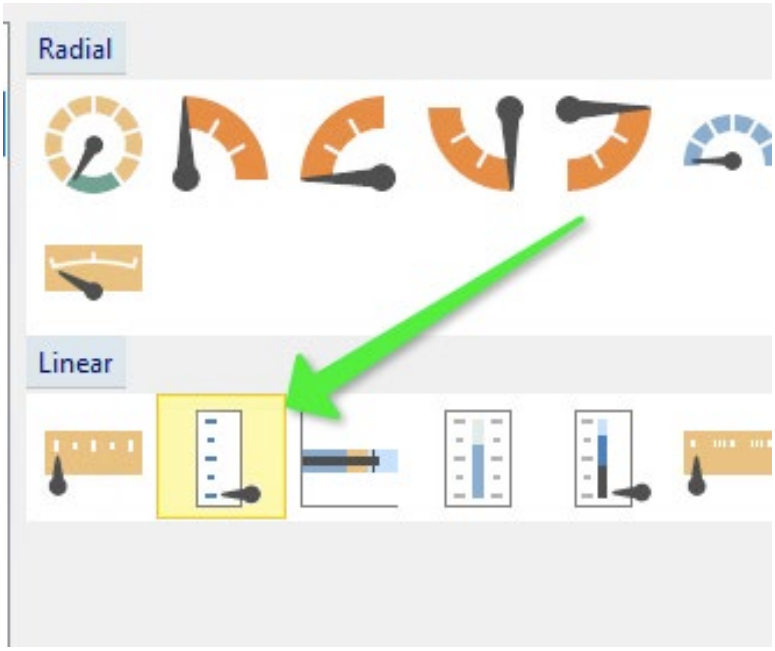
    a. Right click on DataSet1.
    b. Select Dataset Properties.
    c. Change the Name to **dsInventoryTurns**.
    d. Click OK.
14) Remove the grid that the wizard created. We will not need it. Highlight it, press your DEL key, or right click on it and choose DELETE.



15) From the toolbox on the left, drag the Gauge tool into the report design surface.

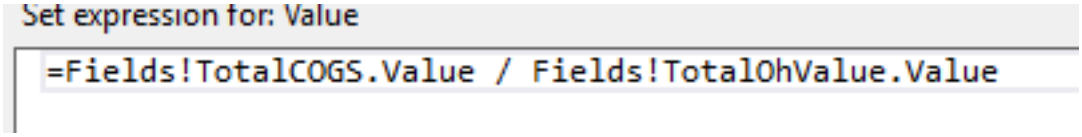


16) Choose the vertical scale under the Linear section and press OK.



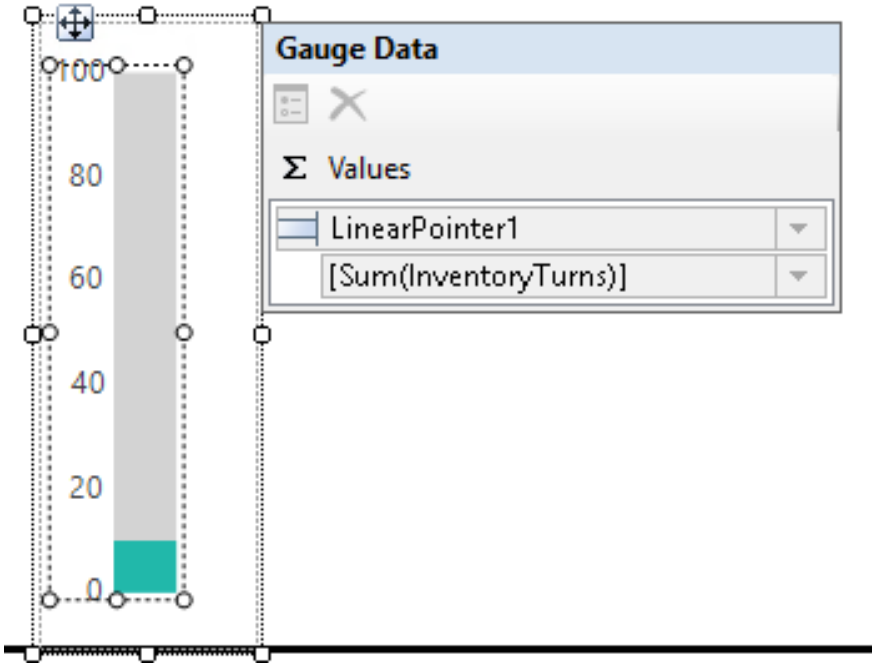
17) Add a calculated field to divide the Total Cost (sales) by the Onhand Value.

a. Right click on dsInventoryTurns and choose Add Calculated Field.
b. Name the field **InventoryTurns**.
c. Click the fx icon, and enter the formula pictured here:



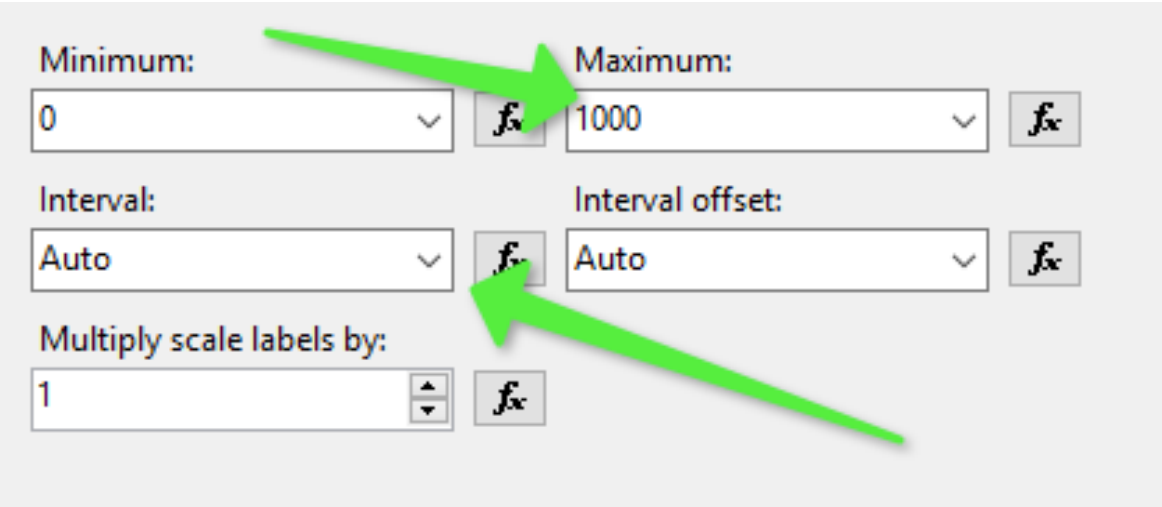
d. Click OK to accept the formula.
e. Click OK to accept the changes to the dataset.

18) Click on the new gauge, replace the static 35 data point with our data by dragging our calculated field from dsInventoryTurns onto the 35 entry. It should look like this:



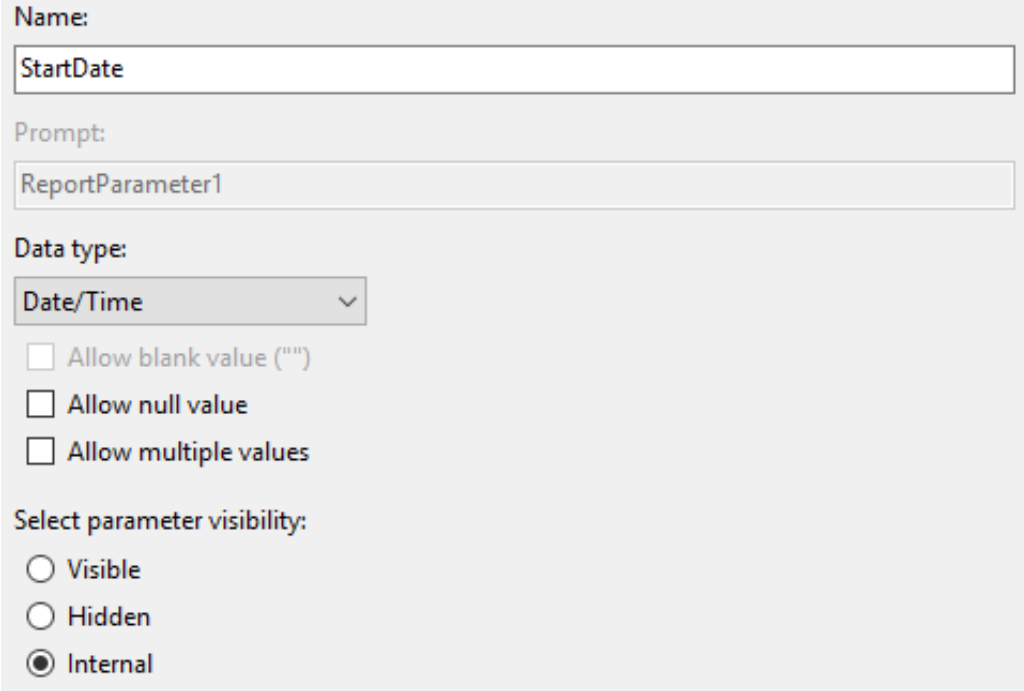
19) Click the Preview tab at the top to review the current chart. Note that the value is at 100% of the gauge. Change the gauge limit to 1000, and set the Interval to Auto.

a. Right click on the Scale and choose Properties.
b. Set the Maximum to 1000.
c. Set the Interval to Auto.
d. Click OK.

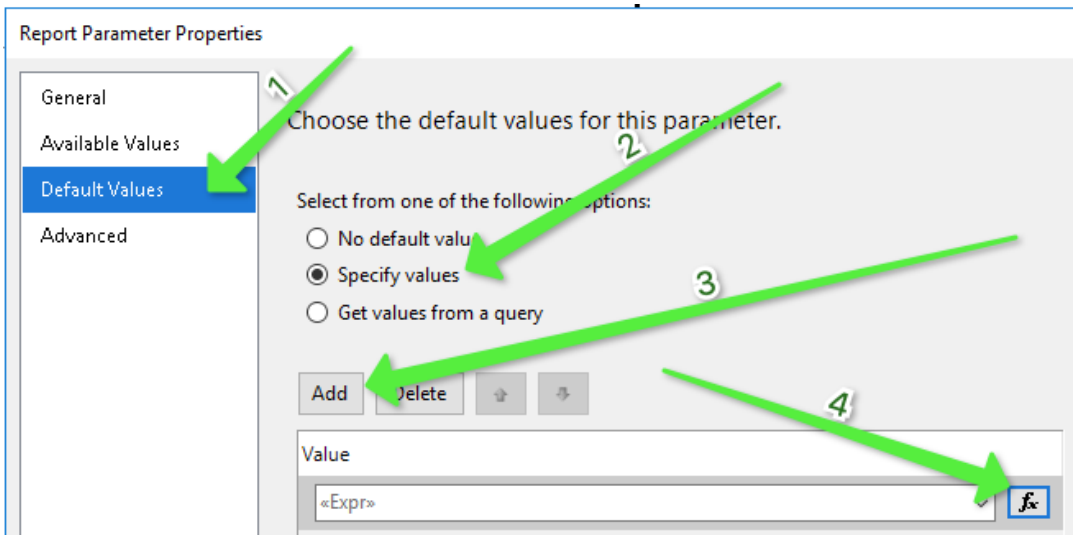


20) Preview it again. Make adjustments as needed.

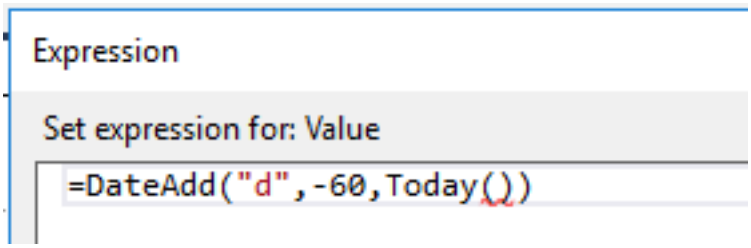
21) Next, create a line chart to represent 60 days of bookings.
22) Start by defining a parameter to represent 60 days ago.
    a. Right click on Parameters and Add a new parameter.
    b. Name it **StartDate**.
    c. Make it an Internal Parameter.
    d. Make its type Date/Time.



    e. Click on Default Values.
    f. Click on Specify values.
    g. Click Add.



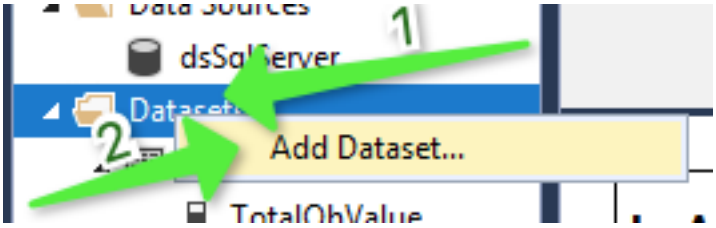
h. Click the fx icon next to the (NULL) value and enter this formula:



i. Click OK to accept the formula.

23) Next, add a dataset to represent 60 days' worth of bookings.

a. Right click on Datasets select Add Dataset.

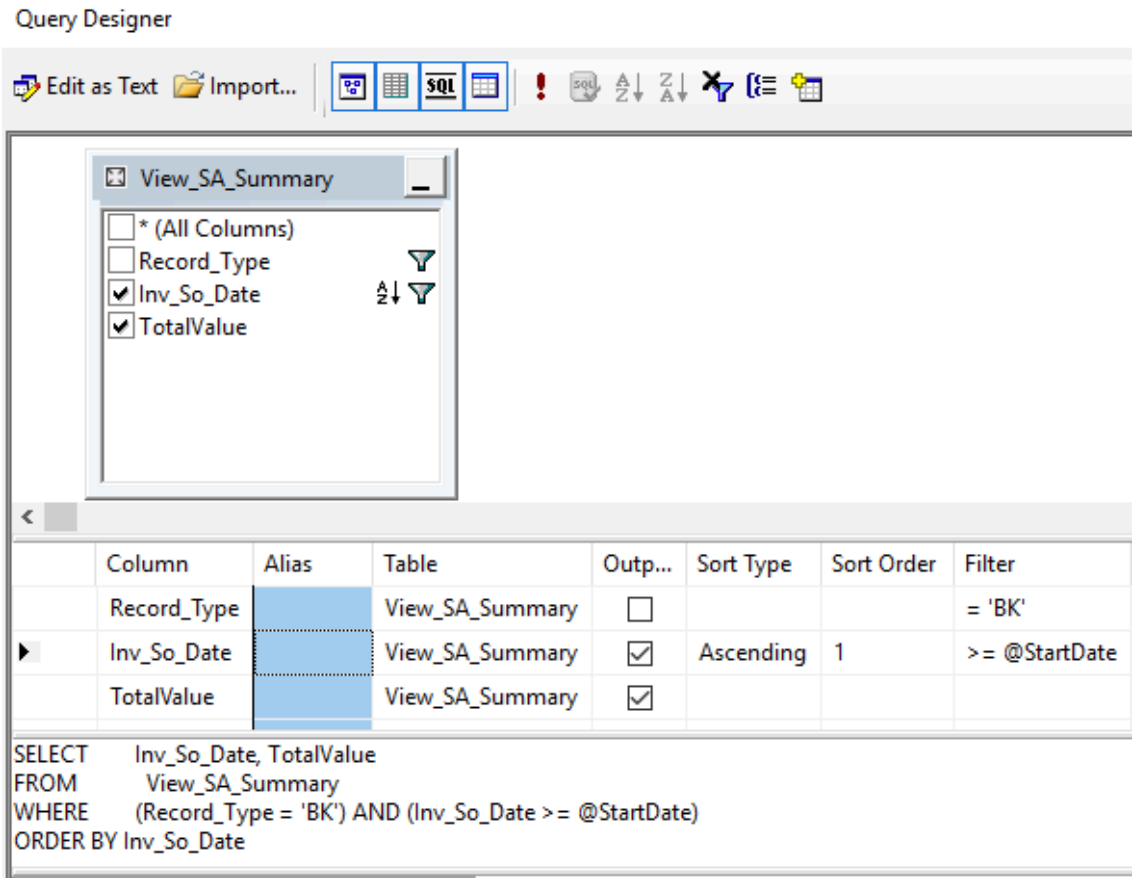


b. Name the new dataset **dsBookings60Days**.

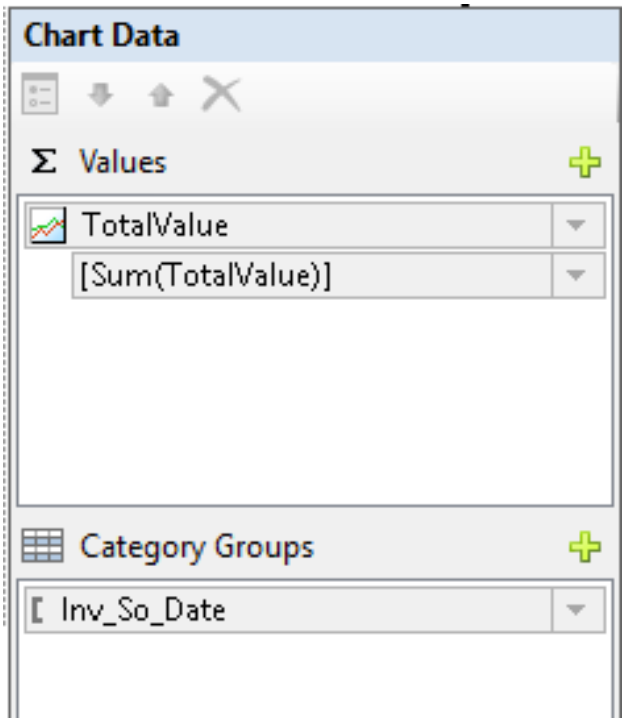
c. Click Use a dataset embedded in my report.

d. Select dsSqlServer for your datasource.

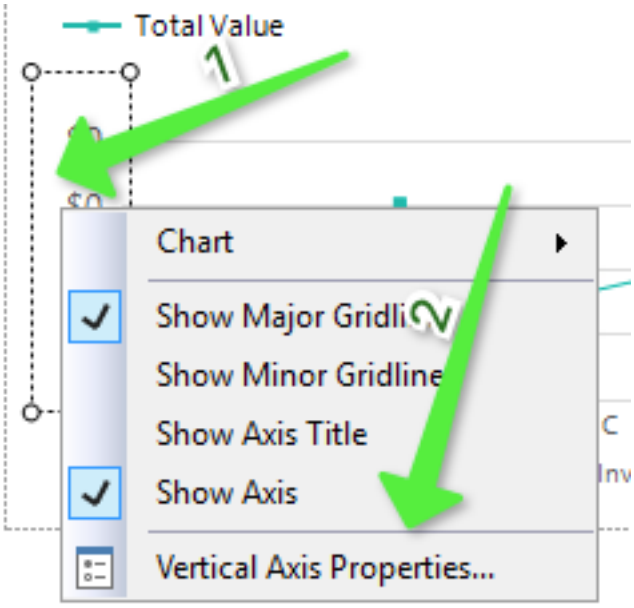
e. Build your Query as seen here:



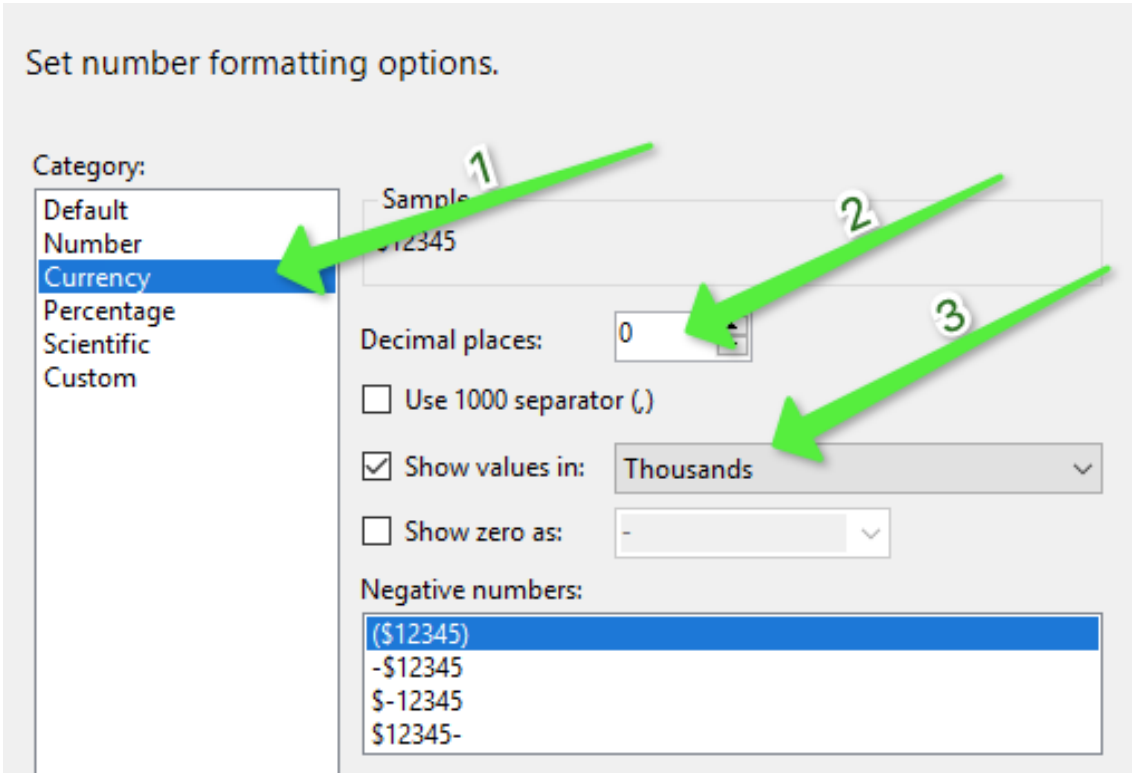
f. Save it by clicking OK until your back to the report designer.
g. Drag the Chart control from the toolbox onto the report design area. Choose a Line type chart.
h. Drag the Total Value field into the Values area of the chart.



i. Drag the Inv_So_Date field into the Category Groups area of the chart.
j. Change the Vertical Axis Properties to have no decimals and be expressed in Thousands:
    i. Right click the Vertical Axis.
    ii. Click Properties.

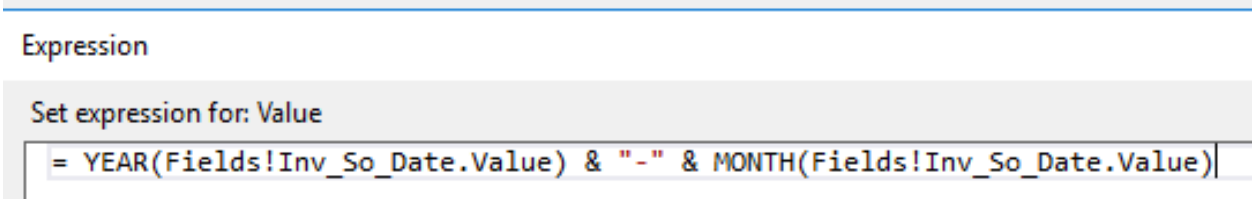


iii. Click Number.
iv. Click Currency.
v. Change Decimals to Zero.
vi. Enable show values in Thousands.



vii. Save and Return to the chart editor.

k. Format the Dates for only YY-MM.
   i. Create a calculated field in the Bookings60Days Dataset using this formula and call it SoYYMM.

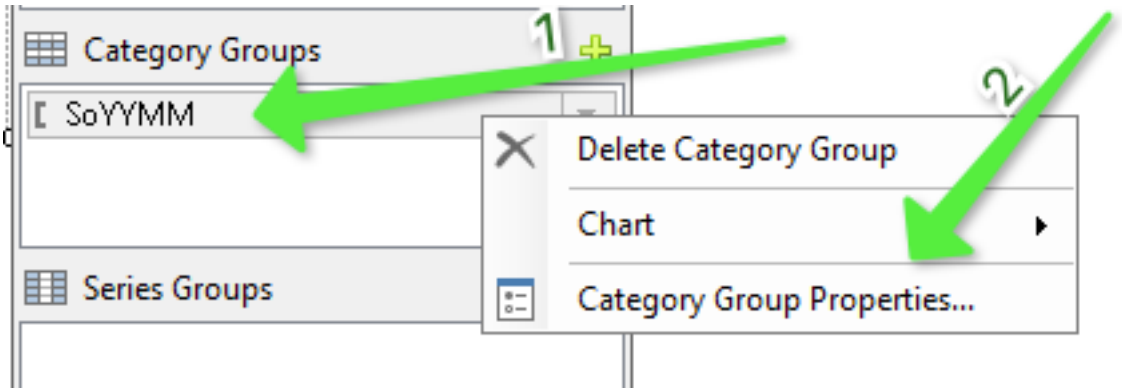


   ii. Change the category group to the new field by dragging it in and replace Inv_So_Date.

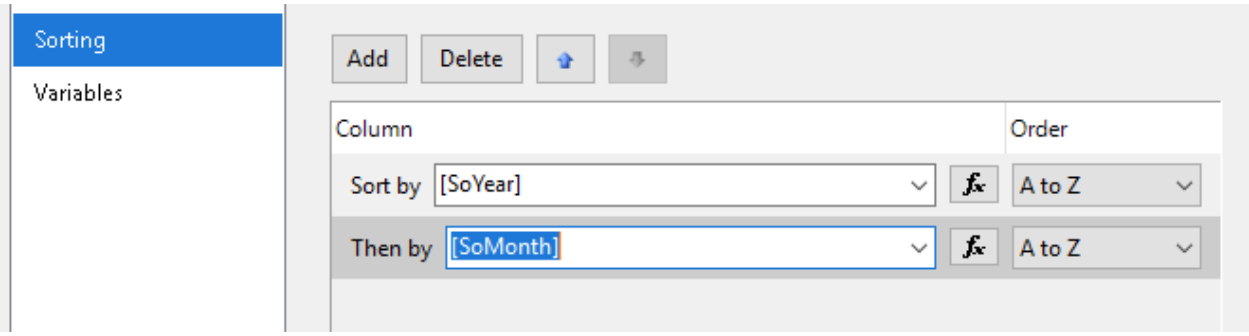
l. Preview the report – and see if you spot the defect…

m. Fix the defect by adding two more calculated fields:
   i. Create SoYear
   ii. Create SoMonth

n. Right click on the Category Group and get its Properties.



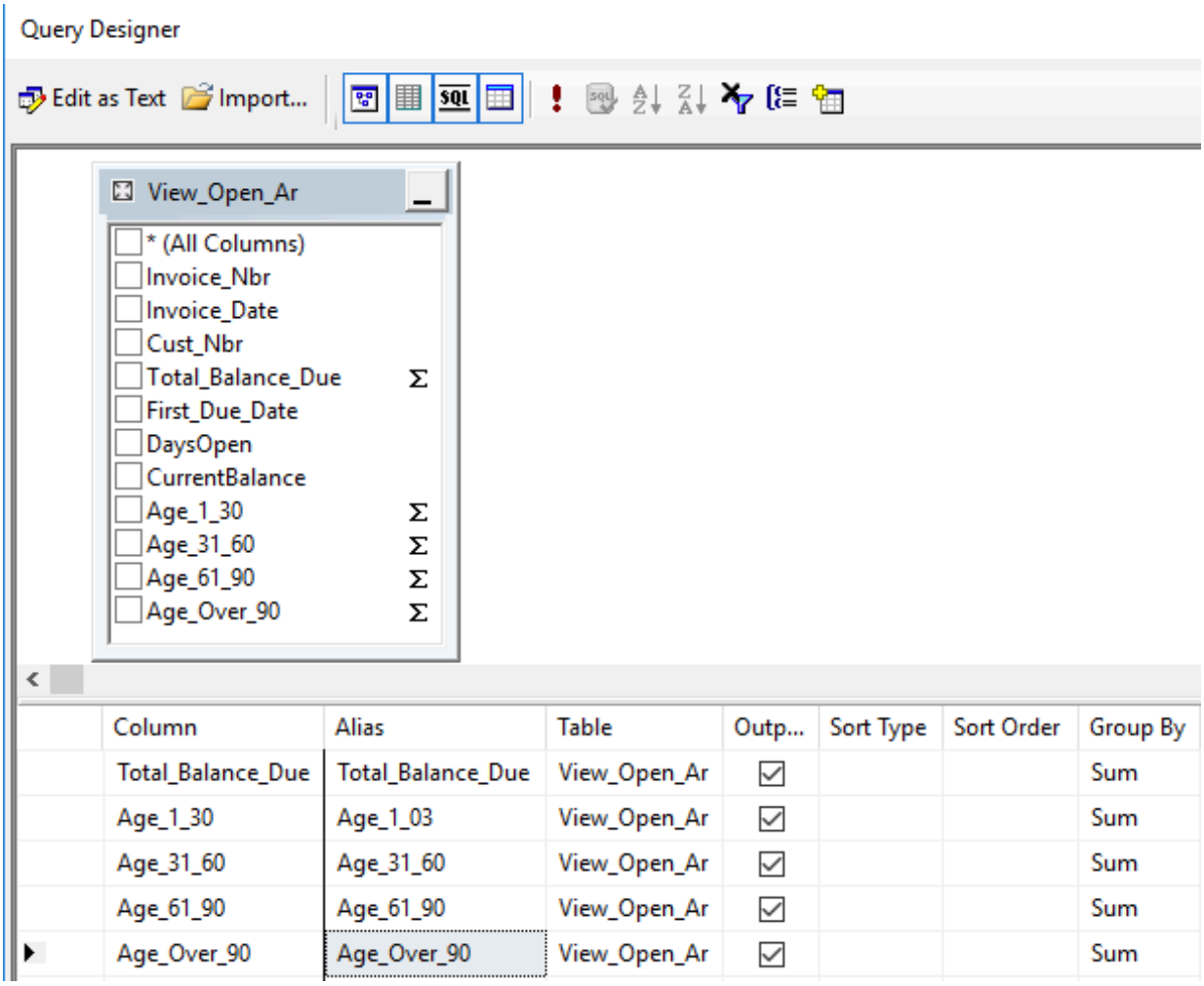
o. Click on Sorting.
p. Replace SoYYMM with SoYear and add SoMonth to the sort.



q. Save and preview to review changes.
r. Bonus: Add logic to remove future bookings.
s. Additional Bonus: Change logic to go back 365 days.
t. Additional Extra Bonus: Add another chart to show SALES for the same period of time.

24) Next, create a pie chart that represents all open AR.

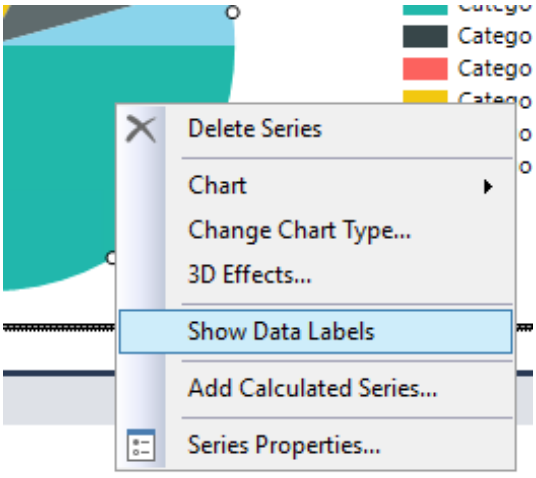
a. Create a new Dataset – Name it **dsOpenARSummary** - using this query:



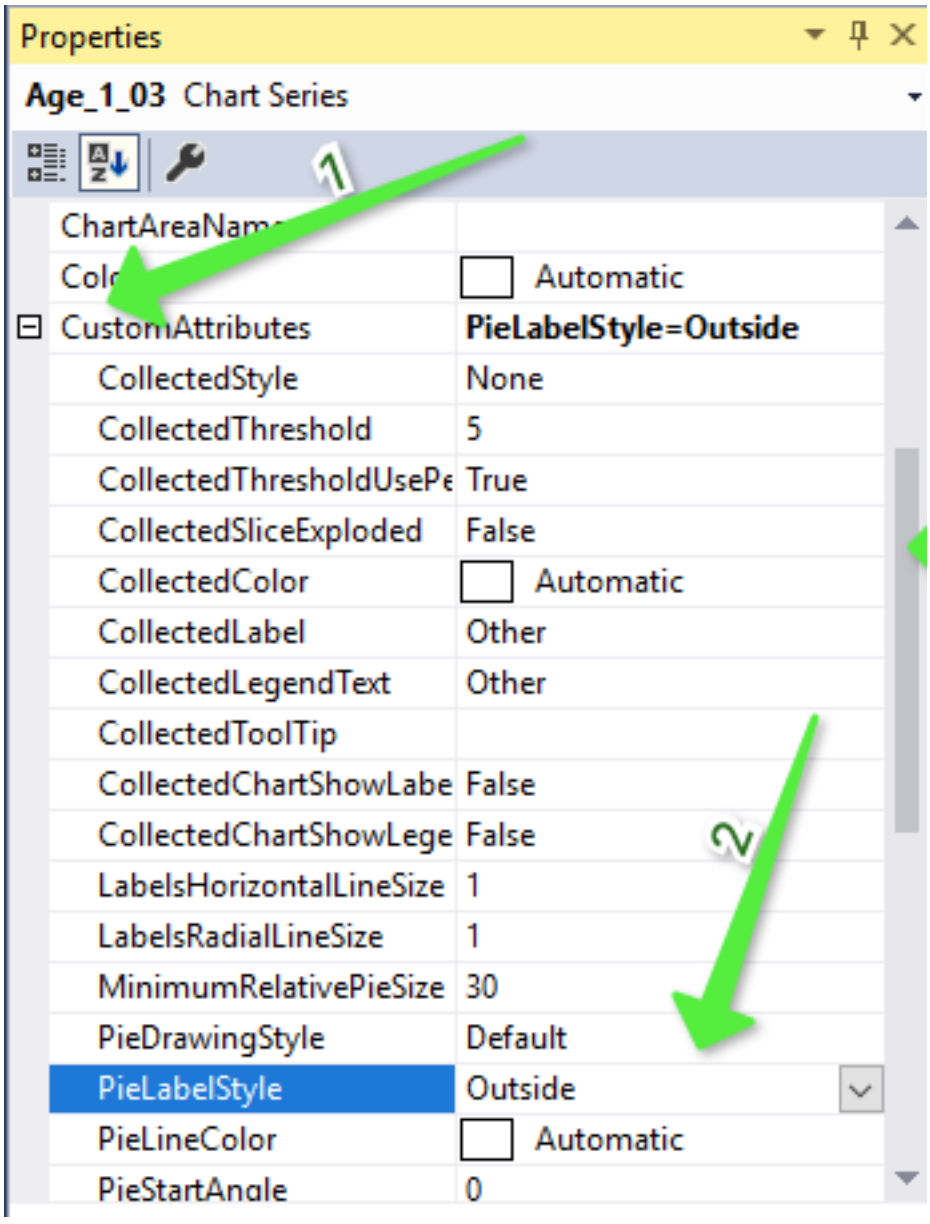
b. Drag the Chart tool into the report design area and choose a PIE type you like.

c. Drag the different Age calculated columns into the Values section of the pie chart as seen here.

d. Right click in the pie and select Show Data Labels:



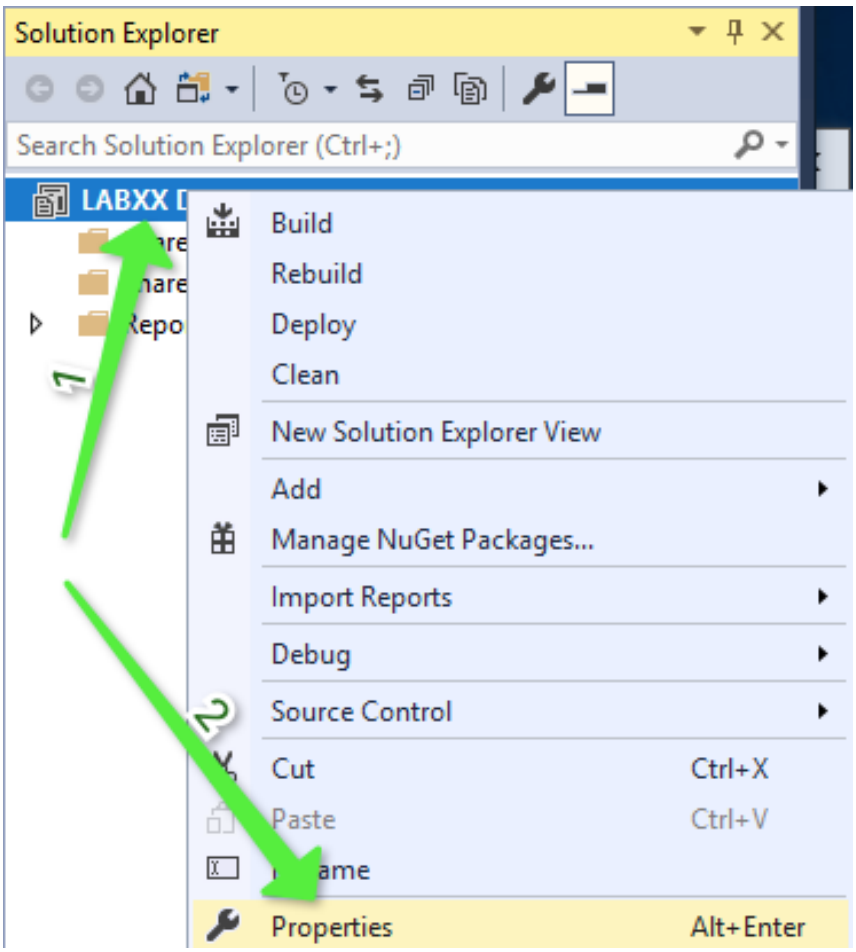
e. Right click the new labels and select Series Label Properties and change the number format to currency, thousands, no decimals.
f. Preview the report changes.
g. Next, move the labels to OUTSIDE the pie parts and draw a line. To do this highlight the pie chart and then go to the properties box:
    i. Expand CustomAttributes
    ii. Change PieLabelStyle to “Outside”



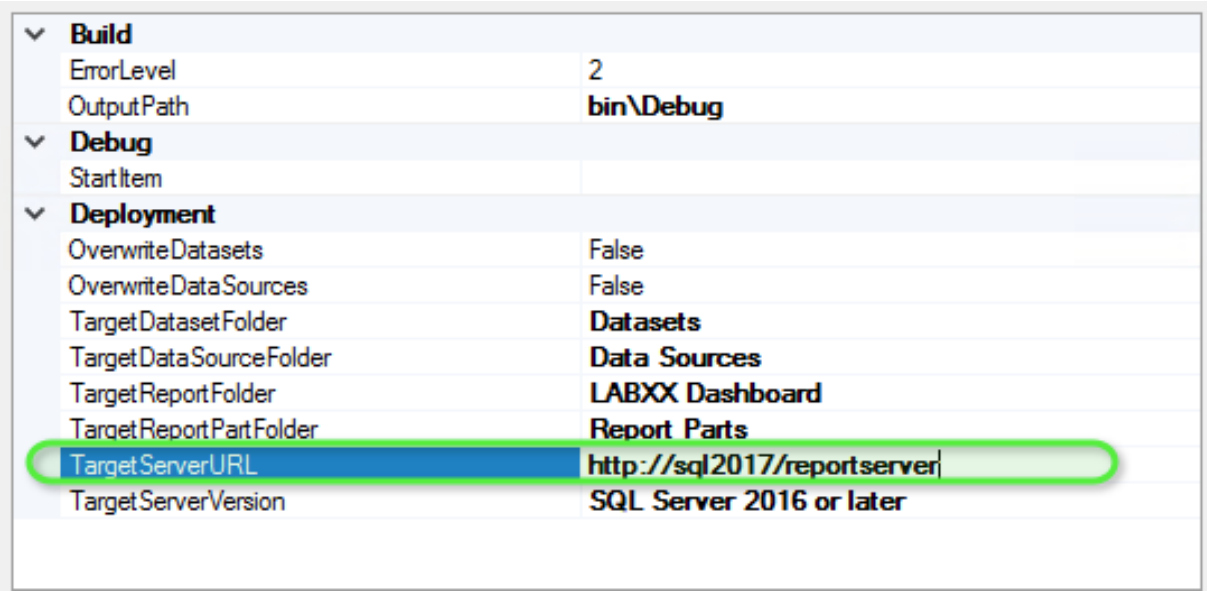
    iii. Below that property, set PieLineColor to Black to draw lines to the pie chunks.
    iv. Preview the changes.

25) Before we can proceed with the next part of our lab, we must publish our report to the report server.

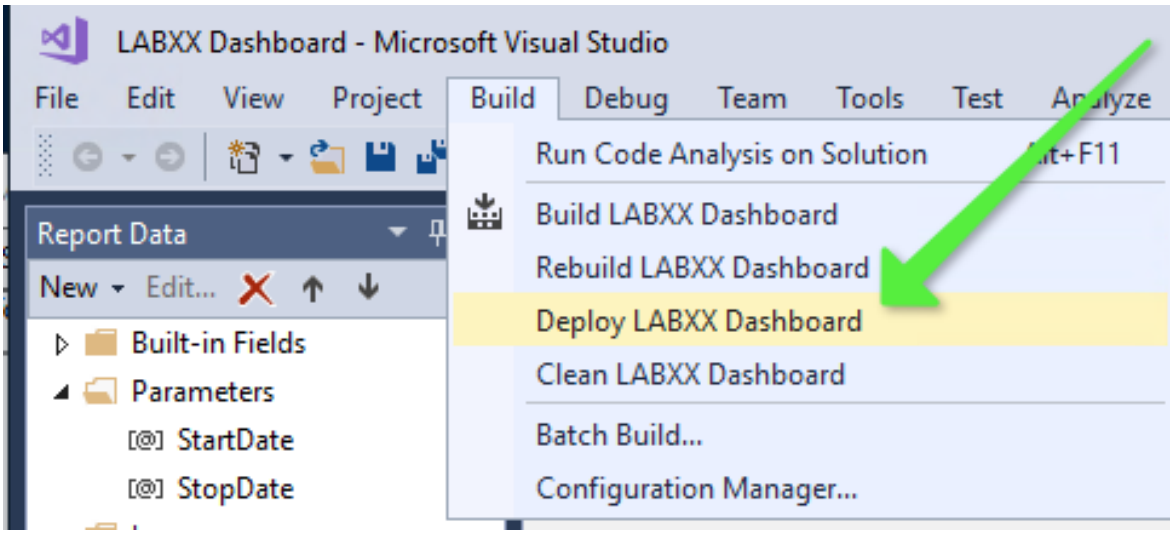
a. Right click the Lab Dashboard Solution and get its properties.



b. Change the TargetServerURL setting to target the remote SQL server at http://sql2017/reportserver as seen here:



c. Press Apply and then OK to accept the changes.
d. Choose Build/Deploy LABXX Dashboard to send the report to the web server.

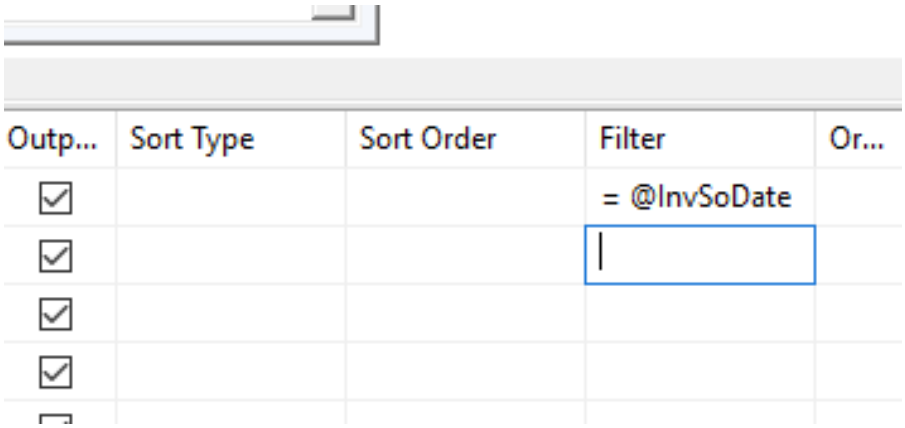


e. Use your web browser on the terminal server and navigate to the SSRS web page at http://sql2017/reports
f. Navigate to your dashboards folder and run your report.

26) Next, design supporting detail for two of our charts so that we can get additional details about a particular chart element when we click on it.

27) On the solution explorer, right click Reports and add another new report.

a. Since datasources are part of the report we will need to add the report source again in the wizard just like before.
b. In your SQL query, at a minimum make certain you include the Inv_So_Date and filter the results to only bookings so that we can match it to our line chart. Include whatever other detail you like from the SA in your query.
    i. Be sure to include @InvSoDate as a parameter for the input of an Inv_So_Date for selection criteria as seen here:

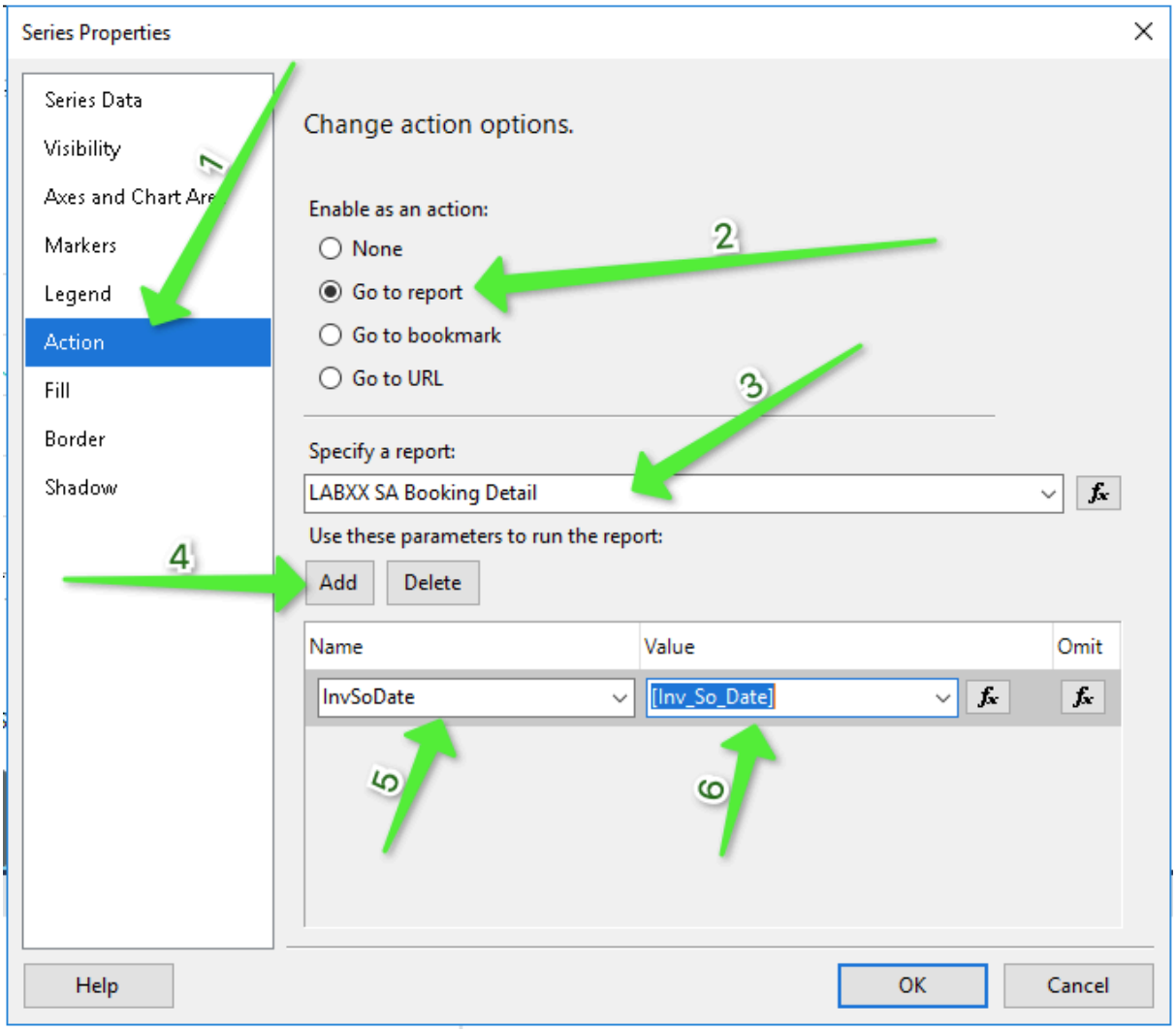


c. Use the default Tabular model and layout the report however you like.
d. Name the report **LABXX SA Booking Detail**.
e. Click FINISH.
f. Double click the InvSoDate parameter and change its type from Text to Date/Time. We will leave it visible and it will have no default.
g. Use the Build menu and Deploy the solution to the web server again.

28) Switch back to the Dashboard report, and click on the line in the chart series.

29) After selecting the line, right click on it and select Series Properties.
30) Under the Properties dialog box:

a. Click Action.
b. Select Go to report.
c. Choose the detail report from the drop down box.
d. Click Add to add a parameter link.
e. Choose the InvSoDate parameter in column 1.
f. Choose the Inv_So_Date column value in column 2.
g. Click OK.



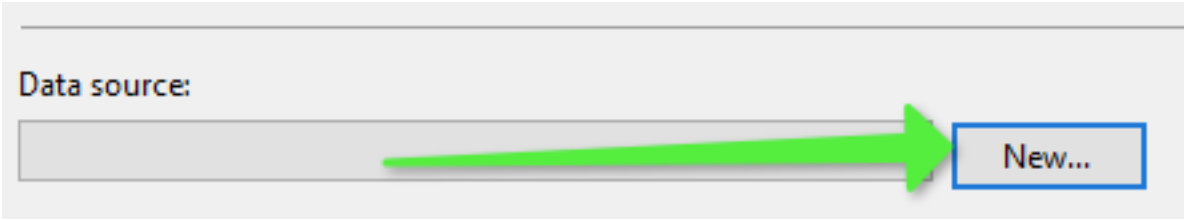
31) Build / Deploy the solution again.

32) Use your browser to test the linkage between the reports.

33) Now we will use a SSAS cube as a data source and add a chart for Sales by Customer.

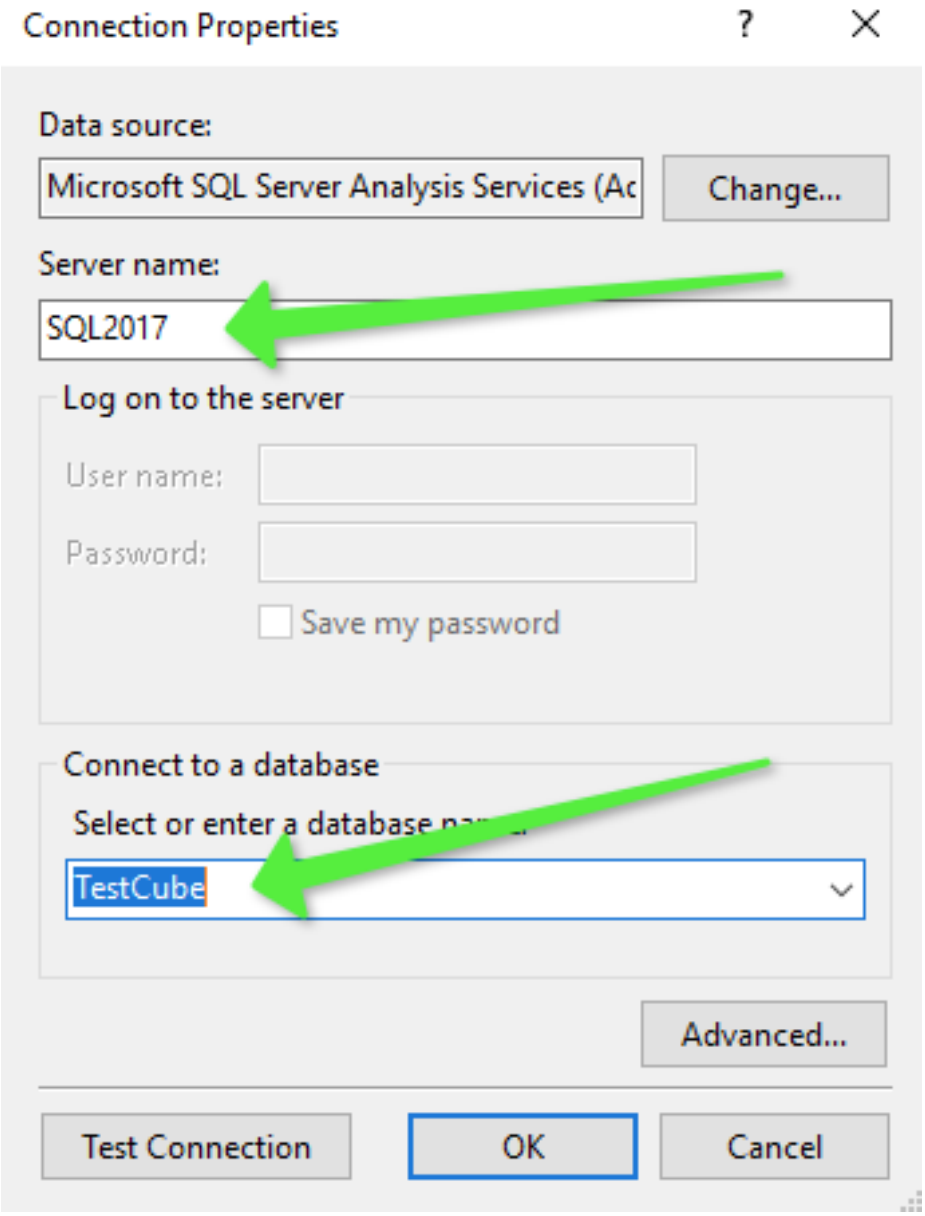
a. Add a new dataset and name it **dsSalesByCust**.

b. For the data source click the new button. We need a new source because SSAS is not the same as SQL server. It's a different source.

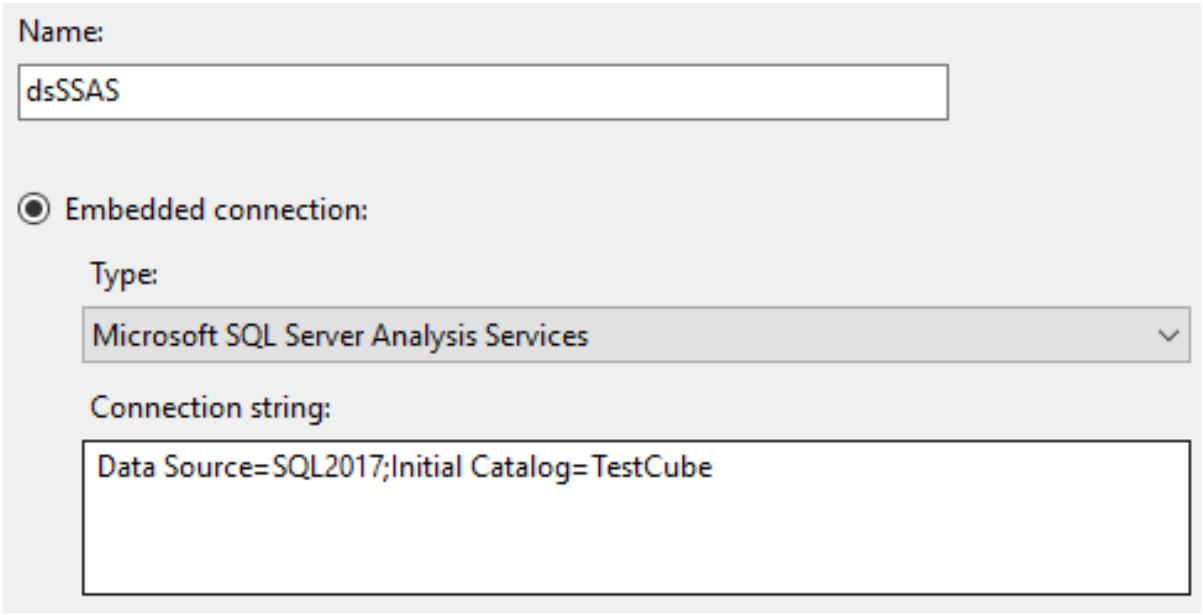


c. Name the source **dsSSAS**.

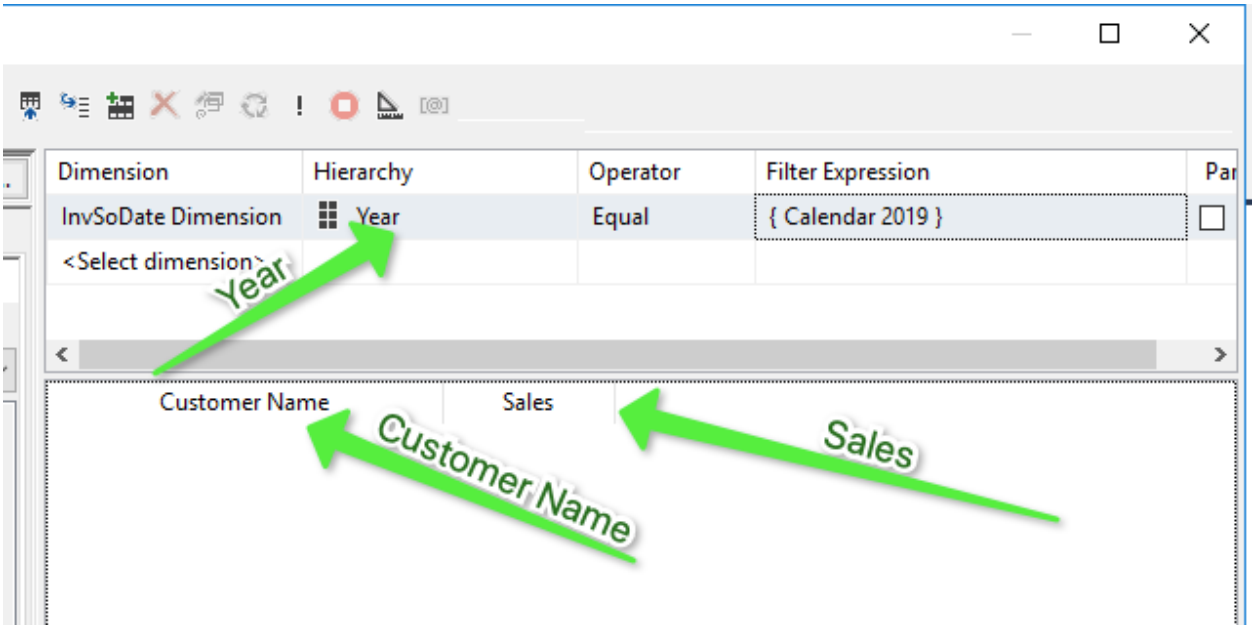
d. Choose SQL Server Analysis Services from the drop down box and click Edit… to edit the datasource.
    i. Enter **SQL2017** for the server name, choose TestCube from the drop down box.
    ii. Press OK.



e. Press OK to save the data source.



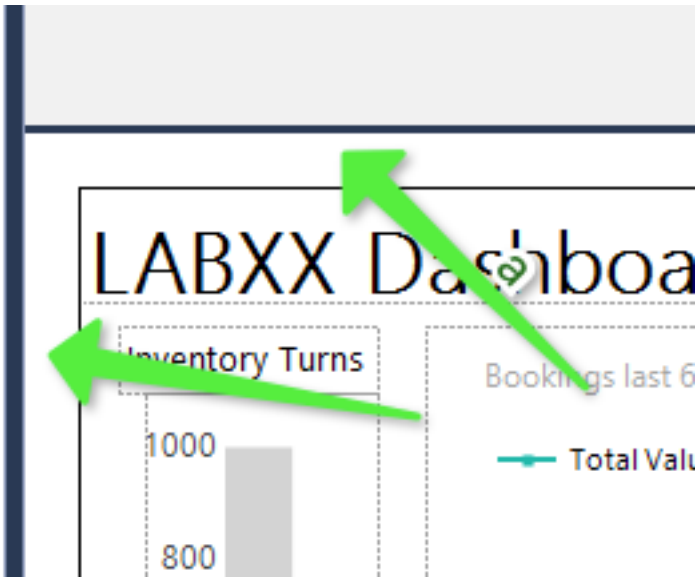
f. Click the Query Designer button to build a query.
g. Drag the customer name dimension and the Sales measure into the query window.
h. Drag the Year dimension into the top section. Your query should look like this:



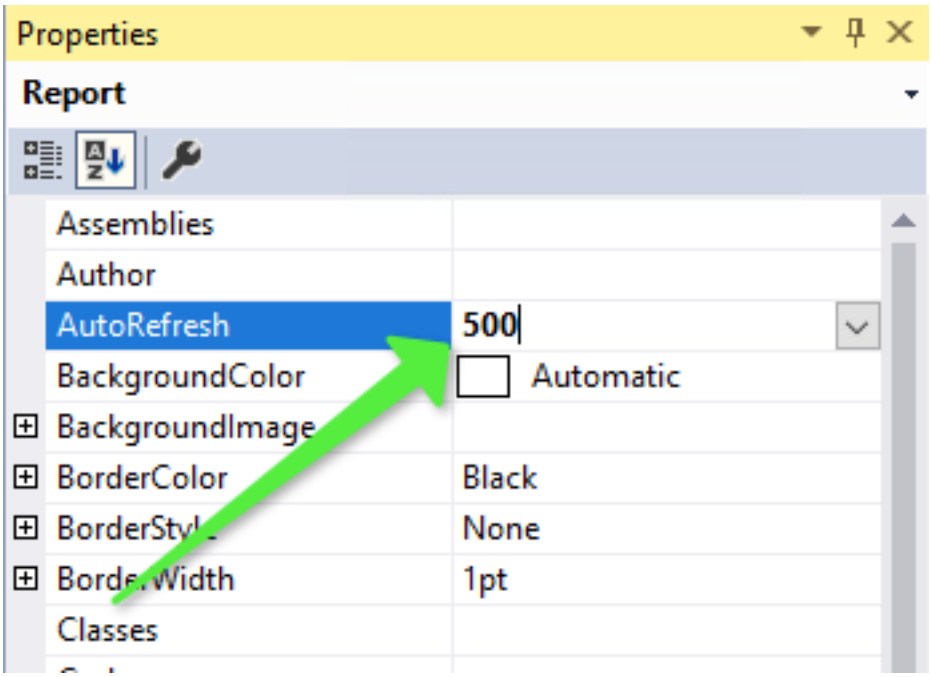
i. Press the "Click to run query" hyperlink to test the query.
j. Click OK and then OK again.
k. Use a chart or table from the toolbox to consume the Sales by Customer dataset in the dashboard.

34) Now that we have the report defined and published, we may want to run it on a display somewhere in the company and some of the measures we may want to "self refresh." Here is how:

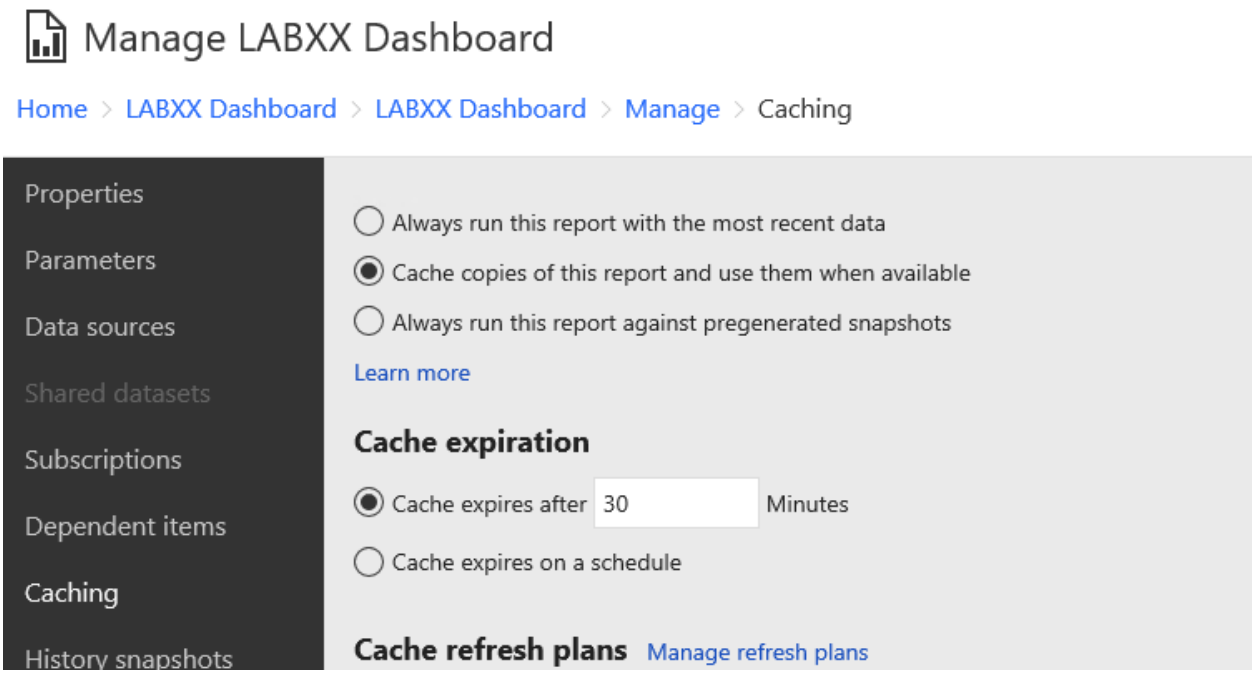
35) Click in the white space in the undefined area of the report.



36) Next look at the properties and note the AutoRefresh property. Populate it with a value in Seconds for how often to refresh the report.



37) If your report is very tough on the SQL server, and multiple people are running the dashboard, you may consider enabling the SSRS cache option on the report in the SSRS web portal as seen here:



At this point we will now do interactive Q&A and explore other options.