Preface

The notes in this document are not meant to be a word for word script. They are instead a list of notes which act as reminders to the various steps needed to meet an objective of a module.

While this set of notes is intended for use by the presenter, it is hoped you as a viewer of the course will find them useful as well.

1 – Introduction

Give standard welcome to the course.

Discuss the nature of this course being a Playbook, and how it differs from normal Pluralsight courses. Stress that this is not an in-depth course that explains ever last little option in SSRS. Instead, it will give you only the bare necessities to get you up and running in SSRS.

List the tools needed for this course, and where they can be downloaded. Explain that while we will be using SQL Server 2016, but the content is applicable to previous versions of SSRS.

- Visual Studio 2015
- SQL Server 2016 Developer Edition
- SQL Server Data Tools (SSDT) – BI
- SQL Server Management Studio (SSMS)
- Wide World Importers sample databases
- Wide World Importers SSAS Multidimensional

Demo – Show creating a new SSRS Solution

Make sure to point out that if the user does not see the Business Intelligence templates it means they haven't installed the SSDT BI components.

Show how to view the SSRS report items pane.

2 – Creating a Line Item Listing Report with SQL Server Data Tools

First Demo – Show creating using the wizard

1. Start with an empty project.
2. Right click on the Reports branch in Solution Explorer.
3. Click on Add New Report to launch the Report Wizard.
4. Explain the first screen is just the welcome screen. Point out the "Don't show this page again" checkbox at the bottom. Click Next to continue.
5. Explain the data source represents a connection to the source database. First, enter a good name, in this case since we're pulling from the Wide World Importers Data Warehouse, we'll use WideWorldImportersDW for the name.
6. For the type, we'll leave it at Microsoft SQL Server. Hit the drop down though to show the variety of sources SSRS supports.
7. Beside the connection string box click Edit…
8. For our server name, we'll enter localhost.
9. Point out the authentication area, for our purposes we'll stick with Windows Authentication.
10. Use the drop down under connect to a database to pick out the WideWorldImportersDW.
11. Click Test Connection to ensure it works.
12. Click OK to close the Connection Properties window and return to the Wizard.
13. Point out the "Make this a shared data source" at the bottom, give a brief overview of what it is but don't click.
14. Click next to go on to the next page in the wizard.
15. On the Design the Query page of the wizard, click the big Query Builder button at the top.
16. Click the + button on the upper right of the toolbar.
17. Find the Sale (Fact) table, click on it then click the Add button.
18. Find the Customer (Dimension) table and add it. Make sure to mention the schema is what is in parenthesis. Click Close to close the Add Table dialog.
19. Point out the query area at the bottom. Show how its already begun building the query, and was able to build the INNER JOIN based on the foreign keys.
20. Add columns by placing check boxes beside them, in this order:
    - 20.1. Sale WWI Invoice ID
    - 20.2. Sale Invoice Date Key
    - 20.3. Customer Customer
    - 20.4. Sale Description
    - 20.5. Sale Quantity
    - 20.6. Sale Total Excluding Tax
21. Click the red exclamation mark to run the query in order to test it.
22. Mention the red stop button can be used to interrupt the query, then show rows retrieved.
23. Now explain we have a problem, and it's important we understand the data. For this database, the customer represents the individual store, and the bill to customer represents the parent company. The way the designer built the query, only orders going to the parent store will be in the report. We want those but we also want the store orders too. In order to get them, we need to remove the AND clause from the query.

24. Remove the and clause, then use the red exclamation to run the query again. Show the new number of rows.
25. Click OK to close the designer.
26. Click next to go to the next page of the wizard.
27. We'll use Tabular for the design. Mention we'll show a Matrix report later in the course. Click Next.
28. On the Design the Table page, explain the various Page, Group, and Details.
29. Click the WWI_Invoice_ID then click the Details button to move it over.
30. Click on Invoice_Date_Key, then drag and drop in details, just to show you can do it that way too.
31. Click on Customer, but put it in the Group area.
32. Click on Description, then holding down shift click on Total_Excluding_Tax, to highlight them all. Click the Details button to place them all in, again just to show another way of doing it.
33. Mention you don't have to use all fields. Also mention the Remove button at the bottom in case you accidentally put in a field you don't want.
34. Mention the Finish button is enabled, but we want to specify a few more things rather than taking the defaults finish would give us. Click Next.
35. Show the difference between stepped and blocked, and we'll use stepped.
36. Check on Include subtotals, explaining its use.
37. Leave Enable drilldown unchecked, explain its use.
38. Click next.
39. On the last page, give the report a good name, in our case M02-Wizard.
40. Mention that the preview report option would generate the report and take you right to preview, but we'll do that on our own.
41. Click finish.
42. Show the report, and explain the various components.
43. Now click on Preview. Make sure to mention when you Preview SSDT automatically saves the report before it previews.
44. Show the report. Explain that it's not pretty, and we'll fix it up in the next module.
45. Show the buttons at the top to quickly explain their use.
46. Return to the Designer.
47. Show the Row Groups area.
48. Click the down arrow by details group, pick Group Properties.
49. While we'll take the default name, in a bigger report you may wish to give this a better name.
50. Show the Page Break, explain that in the details it probably doesn't make sense to use it.
51. Click on Sorting, then click Add.
52. In the drop down, pick WWI_Invoice_ID. This will now sort the details by the invoice ID, a more natural order for the report user.
53. Quickly show the other pages and explain, but don't use.
54. Click OK to close.
55. Now use the Group Properties for the table1_Customer grouping.
56. Jump right to Page Breaks. Explain here it would make sense to put in a page break between each customer. Check it on.
57. Go to sorting, just to show when we picked customer earlier in the table layout it knew we wanted to sort by Customer. Then click OK.

58. Click Preview again. (Point out the Save button in case we wanted to save the report without having to preview.
59. Show the details are now sorted by invoice id.
60. Point out one other issue, the report is trying to total the Invoice ID, which doesn't make sense.
61. Return to the designer.
62. Click in the invoice sum box, and delete it.
63. Preview one more time.
64. Return to designer.
65. That's it for the wizard, again remind we'll do clean up in next module.
66. Proceed to manual creation of the report.

Second Demo – Show creating the report manually

1. Add the report.
   1.1. Right click on Reports.
   1.2. Don't click on New Report. Instead, pick Add, New Item.
   1.3. Click on Report.
   1.4. Give the report a name. For this use M02-Manual. Click Add.
2. Add the data source.
   2.1. Show the Report Data pane at the left. If not visible, click on View, and pick Report Data at the bottom.
   2.2. Right click on Data Sources, and pick Add Data Source…
   2.3. Point out the options here are the same as in the Wizard.
   2.4. Give the data source a name of WideWorldImportersDW.
   2.5. Leave the type as Microsoft SQL Server.
   2.6. Click Edit beside connection string.
   2.7. Localhost for server, WideWorldImportersDW for the database. Test Connection, then click OK.
   2.8. Click OK to close the Data Source, you've now added it successfully.
3. Add the dataset.
   3.1. Right click on Datasets, and pick Add Dataset.
   3.2. Change the name to SalesData.
   3.3. Change the option to Dataset embedded in my report.
   3.4. Point to the Query Designer button at the bottom, and mention it will bring up the same dialog we've already seen (but don't do it).
   3.5. Jump to SSMS, and show the query below. Explain that experienced SQL developers will often find it easier to work out the query to get their data here, and then just paste it into the Query area.

```
SELECT fs.[WWI Invoice ID]
     , fs.[Invoice Date Key]
     , dc.[Customer]
     , fs.[Description]
     , fs.[Quantity]
     , fs.[Total Excluding Tax]
  FROM Fact.Sale fs
 INNER JOIN Dimension.Customer dc
```

```
ON fs.[Customer Key] = dc.[Customer Key]
```

3.6. Paste in the query. Mention that we'll go over some of the other items on the left in future modules, such as parameters.
3.7. Click OK. You've now added the dataset.

4. Add the table.
   4.1. Switch to the Toolbox pane. If not visible, make it so by going to the View menu and picking toolbox.
   4.2. Drag and drop a Table item to the design surface.
   4.3. Return to the Report Data pane.
   4.4. Drag and drop the WWI_Invoice_ID to the first column.
   4.5. Repeat with Invoice_Date_Key and Description.
   4.6. Drag the Quantity to the right edge, show the blue line as it will add the field as a new column.
   4.7. Click on Preview just to see what's been done thus far.
5. Add the Customer group.
   5.1. Now we need to add the group. Go to the Row Groups. Hit the drop down arrow beside details.
   5.2. Click Add Group, then Parent Group.
   5.3. Pick Customer.
   5.4. Check on group header. Header will hold the header which will be used to display the customer name.
   5.5. Click on the drop down beside Details.
   5.6. Hover over Add Total.
   5.7. Explain this adds totals for the detail section, i.e. each customer. If we use before we get the same as the previous report. This time let's pick After to place the totals at the bottom.
6. Add totals for entire report.
   6.1. Now we want to add totals for the entire report. This time click on the drop down beside Customer, pick Add Total, and After.
   6.2. Show how the table has updated. Explain the grouping arrows on the left of the table controls.
   6.3. Mention that you could update the wizard generated report in the same fashion.
7. Now explain we have the same issue with totaling the invoice number, so delete the group and grand totals for the invoice number.
8. Fix sorting of the details area
   8.1. Click the drop down beside the Details again, and pick group properties.
   8.2. Go to sorting, pick Add, and pick the WWI_Invoice_ID. Click OK.
9. Now click the drop down beside Customer. Go to Page Breaks, and check on Between each instance of a group.
10. Click Preview.
11. Scroll to roughly page 8 to show the subtotals.
12. Go to the last page to show the grand totals.
13. Return to the report.
14. Add a title in the report
    14.1. If the grid selectors (the gray bars around the table) aren't visible, click in the table.
    14.2. Click on the button in the upper right.
    14.3. Move the table down some.
    14.4. Return to the toolbox.

14.5. Drag and drop a text box at the top.
14.6. Enter M02-Manual for the title.
14.7. Preview.
14.8. Return to designer.

15. At this point we have pretty much have the same report as what the wizard generated. The exception being the fonts, which we'll clean up in the next module, and the placement of totals which we did just to be different.

3 – Formatting a Report

1. Copying a report as a starting point.
    1.1. Start with the reports from Module 2.
    1.2. Show the Solution at the top.
    1.3. If they don't see it, show them how to show it. Tools, Options. In the options, go to Projects and Solutions, General. Check on Always show solution. Click OK to save the options.
    1.4. Now explain why we want this. Right click on the solution, and pick Open Folder in File Explorer.
    1.5. In the folder right click on M02-Manual.rdl, Copy and Paste.
    1.6. Rename to M03-Format.rdl
    1.7. Go back to VS (SSDT).
    1.8. Right click on Reports branch. Add | Existing Item…
    1.9. Pick the M03-Format.rdl
    1.10. Preview the report to show the bad formatting. Return to design.
2. Formatting table values.
    2.1. Right click on Total Excluding Tax. Pick Text Box Properties.
    2.2. Go to the Number tab.
    2.3. Show the different categories, but pick Currency.
    2.4. Set Decimal Places to 0. Check on Use 1000 separator.
    2.5. Say normally you'd stop here, but do a quick tour of the other pages.
    2.6. Preview the report to show the formatted value.
    2.7. Go to the last page of the report.
    2.8. Point out the totals aren't formatted.
    2.9. Return to Design Mode.
    2.10. Explain we now need to format the totals. We could repeat the above steps, but there's a faster way.
    2.11. With the Total Excluding Tax box selected, go to the properties window.
    2.12. Find the Format property under Number.
    2.13. Copy to the clipboard.
    2.14. Click on the first sum and while holding the mouse button drag down to grab the grand total.
    2.15. In the properties window again find the Format property.
    2.16. Paste in what was copied.
    2.17. Preview the report, go to the last page. Show the formatted totals.
    2.18. Now to fix Quantity.
    2.19. Right click on Quantity. Pick Text Box Properties.
    2.20. Click on the Number tab. Pick no decimal, comma. Click OK to close.
    2.21. Go to properties and copy the format.
    2.22. Highlight the two total boxes, paste in the format into the properties window.
    2.23. Now fix the date. Right click in the Invoice Date. Pick Text Box Properties.
    2.24. Pick Number, then the Date category.
    2.25. Explain what the * is for, and take the first item.

2.26. Preview, go to last page and show.
2.27. Point out the Invoice ID won't be formatted, as it isn't meant to be a number but a single value.

3. Fixing the Page Formatting.
   3.1. On the main menu, pick Report, then Report Properties.
   3.2. Change the orientation to Landscape.
   3.3. Point out that you can change the Paper Size if need be, although we won't.
   3.4. Change the margins to 0.5 all around.
   3.5. Click in the Report area, and show the Report Body. Point out the new margins.
   3.6. Now click inside the Body area, and show the properties change.
   3.7. Point out the difference between the two.
   3.8. Show how you can change the body size with the mouse, but the properties window is much more accurate.
   3.9. Change the width of the body in the properties to 10 inches (11 – 0.5 – 0.5).
   3.10. Explain we don't have to adjust the height, as SSRS will auto expand it.
4. Expanding the table to take up the space.
   4.1. Click in the table.
   4.2. Expand the Total Excluding Tax so that the header is visible.
   4.3. Expand the Description to take up rest of space.
   4.4. Warn users not to drag too far, so it won't make the body bigger. Otherwise it spills to multiple pages and becomes unusable.
   4.5. Change Invoice Date Key header to just Invoice Date.
   4.6. In the tool bar, use the Right Align button.
   4.7. Select both Quantity and Total Excluding Tax headers by clicking and dragging. Again use the right align button. Mention you could also do this in the Properties pane, but this is faster.
   4.8. Preview the report, show the new formatting, then return to design mode.
5. Moving the Customer field.
   5.1. Now we can fix the Customer field to be more attractive, but we need to do a bit of trickery.
   5.2. Highlight the empty cells under the WWI Invoice ID to Total Excluding Tax headers.
   5.3. Right click and pick Merge Cells.
   5.4. Click on the Field Picker box on the very right of the new cell, and pick Customer.
   5.5. Right click on the Customer Header, and Delete Column.
   5.6. Merge the three cells to the left of the quantity subtotal.
   5.7. Merge the three cells to the left of the quantity grandtotal.
   5.8. Enter Subtotal and Grand Total into the boxes.
   5.9. Preview. Also go to last page to show subtotal/grand total.
   5.10. Return to design.
6. Formatting the text.
   6.1. Click in Customer, in the toolbar change to 11 pt, and Bold.
   6.2. Highlight the column headers, again use the toolbar, 12 pt Bold.
   6.3. In the toolbar pick background color button.
   6.4. Pick a light gray.
   6.5. Resize the column width / height for the new font sizes, as well as taking up the extra space given to us by the now absent Customer column.

6.6. Bold the subtotal / grand total areas.
6.7. Preview, go to end, then back to design.

7. Making headers appear on each page.
   7.1. Explain the easy way is to right click on the gray bar to the left of the Tablix and picking Tablix properties.
   7.2. On General you could check on the Repeat header columns on each page.
   7.3. However, SSRS doesn't always recognize which row we want to use for our column headers. In addition, we also want the Customer field to repeat on each page. So we'll have to use Advanced mode plus the properties pane to accomplish this.
   7.4. To the right of the Column Groups is a drop down arrow. Click it then pick Advanced Mode.
   7.5. The first row, Static, corresponds to our row holding the column headers. Click it.
   7.6. In the properties pane, set KeepWithGroup to After.
   7.7. In the RepeatOnNewPage, set to True.
   7.8. Now click on the Static row under the Customer group.
   7.9. In the properties pane, set KeepWithGroup to After.
   7.10. In the RepeatOnNewPage, set to True.
   7.11. Preview, then return to Design mode.
8. Headers
   8.1. In the main menu, Click on Report, then Add Page Header.
   8.2. From the Toolbox pane, drag and drop a text box.
   8.3. Resize the text box to take up the width of the top.
   8.4. Enter Sales Report into the text box.
   8.5. Make it bold, centered, 20 pt.
   8.6. Reduce the size of the header.
   8.7. Remove the text box with the name of the report, since we don't need it anymore.
   8.8. Click in the Tablix, then in the gray box in the upper left corner.
   8.9. Move the Tablix higher in the report.
   8.10. Shrink the report body.
   8.11. Preview, then return to Design Mode.
9. Footers
   9.1. Report menu, Add Page Footer
   9.2. Add three text boxes, left middle right.
   9.3. Show the blue lines you can use to align.
   9.4. Resize each to take more space. Make sure middle one is double high.
   9.5. Right click on left most box, pick Expression.
   9.6. After the = sign, enter *"Printed on" &* then under Built-in Fields double click Execution Time, add & vbCrLf & "By " & then double click UserId. The formula should now look like:
   9.7.
```
= "Printed on " & Globals!ExecutionTime & vbCrLf & "By " & User!UserID
```
   9.8. Click OK to close.
   9.9. Right click on middle textbox, and use this expression:
   9.10.
```
= "Copyright " & YEAR(NOW) & " Pluralsight" & vbcrlf & "All rights reserved"
```
   9.11. Click OK to close the expression. Using the toolbar set it to centered.
   9.12. In the right text box, bring up the Expression window.

9.13. After =, enter “Page “ & then double click OverallPageNumber, then & “ of “ & then OverallTotalPages. The expression should look like:

9.14. `= "Page " & Globals!OverallPageNumber & " of " & Globals!OverallTotalPages`

9.15. OK to close.

9.16. Use the toolbar to set the alignment to right.

9.17. Use the mouse to highlight / select all three boxes.

9.18. Use the Set Foreground Color button in the toolbar.

9.19. Set it to Dim Gray, explaining that this will deemphasize this info which isn’t as important as the main report.

9.20. Preview, then go back to design.

9.21. Explain why we picked the info to display. Also add that any of these could also have been done in the page header, we just chose to do so in the footer.

10. Summary

10.1. This gives us a nicely formatted report. We could have gone further in changing other fonts, colors, etc.

10.2. Suggest that your company come up with a set of standards for reports. Things like alignment, fonts, colors, what goes in the header/footer, etc. Then enforce that so all reports created in your company will have the same look and feel.

4 – Creating a Matrix Report with SQL Server Data Tools

1. Right click on Reports, Add New Report.
2. Next to go past welcome page.
3. Set the data source.
   3.1. Enter WideWorldImporters_Cube for the Name.
   3.2. Under type, set it to Microsoft SQL Server Analysis Services.
   3.3. Click Edit by Connection String.
   3.4. Use localhost for Server Name.
   3.5. Pick the WWI-SSASMD for the database name.
   3.6. Test Connection, assuming it passes click OK to close, then OK to go to next page of the wizard.
4. Create the query.
   4.1. Click the Query Builder button.
   4.2. For this, we'll display sales by state and year.
   4.3. Expand Measures, then drag Sales Total Excluding Tax to the middle.
   4.4. Scroll down to Delivery Date and expand.
   4.5. Drag and drop Delivery Date.Calendar Abbreviation to the middle.
   4.6. Explain we won't drill down to the individual day, so grab the Date column and remove it.
   4.7. Drag the Month Year Key to the middle. Explain we'll need that later to do some sorting.
   4.8. Scroll up to City.
   4.9. Drag and drop Geography to the middle.
   4.10. Remind viewers that this is not the way the report will look on the screen, this is just the data we'll be getting.
   4.11. Click OK.
   4.12. Click Next to go to next page of the wizard.
5. Pick Matrix, then Next.
6. Set the location of the fields in the matrix.
   6.1. Put Year, Quarter, Month_Abbreviation in the Columns.
   6.2. Put Continent, Country, State_Provice, and City in Rows.
   6.3. Put Sales_Total_Excluding_Tax in the details.
   6.4. Important, check on Enable drilldown.
   6.5. Click Next.
7. Name the report M04-Matrix and click Finish.
8. Preview the report.
9. Show the drill up/down method.
10. Fix the ordering of months.
    10.1. Drill down to months, showing they are out of order.
    10.2. Go back to Design.
    10.3. In the Column Groups, click the drop down by matrix1_Month_Abbreviation and pick Group properties.
    10.4. Go to Sorting.
    10.5. Change the column to Month_Year_Key.
    10.6. Click OK.

10.7. Preview the report again.
10.8. Drill down in the calendar to show months now sorted correctly.

11. Explain users can now apply the same formatting techniques used in the module on Formatting.
12. Remind users that matrixes can also come from other data sources besides SSAS.

5 – Creating a Chart Report with SQL Server Data Tools

1. In this module, we'll create a report with a Chart that compares the sales for each customer buying group on a month by month basis.
2. Often data for charts comes from an aggregated source such as Analysis Services, so we'll be using that for this demo. However, know the source data can come from any valid data source.
3. The Wizard, while great for line item style reports, does not support charts. Thus, we'll have to manually add the data source and data set.
4. Mention that while we'll show the steps we won't go in depth into explanations. For more information refer to the earlier module, Creating a Line Item Listing Report.
5. Add the data source.
    5.1. Right click on Reports, pick Add | New Item
    5.2. In the dialog, select Report, then name it M05-Chart.
    5.3. Right click on Data Sources, pick Add Data Source.
    5.4. Name the data source WideWorldImporters_Cube.
    5.5. Change the type to Microsoft SQL Server Analysis Services.
    5.6. Click Edit by Connection String.
    5.7. Server name should be localhost.
    5.8. Connect to database WWI-SSASMD.
    5.9. Test, OK to close connection dialog, OK again to close data source properties.
6. Add the dataset.
    6.1. Right click on Datasets, pick Add Dataset.
    6.2. Name it BuyingGroupSalesByMonth (note that no spaces are allowed).
    6.3. Change to Use a dataset embedded in my report.
    6.4. In the Data source drop down, pick WideWorldImporters_Cube.
    6.5. Click on the Query Designer button.
    6.6. Expand Measures, Sale, move Sales Total Excluding Tax to middle.
    6.7. Expand Delivery Date, move Month Year and Month Year Key to middle. Mention we need the key later for sorting.
    6.8. Expand Customer, drag Buying Group to middle.
    6.9. Click OK to close the Query Designer.
    6.10. Click OK again to close the dataset properties.
7. Add the chart.
    7.1. Bring up the toolbox.
    7.2. Drag and drop a Chart onto the report.
    7.3. Under chart type pick Line as it is the most common.
    7.4. Resize to take up a good amount of space.
    7.5. Click inside the chart line area to bring up the chart data dialog.
    7.6. Click + beside values.
    7.7. Select Sales_Total_Excluding_Tax.
    7.8. Click + beside Category Groups. These are the values plotted across the bottom.
    7.9. Pick Month_Year.
    7.10. Click + beside Series Groups.

7.11. Pick Buying Group.
7.12. Preview the report.

8. Fix the horizontal axis.
   8.1. Show the labels on the bottom are not very usable. Return to design.
   8.2. Right click on the Horizontal area at the bottom. Pick Horizontal Axis Properties.
   8.3. The interval indicates how many labels are displayed. A value of 1 displays every label, 2 is ever other, 3 is every third, etc. Set to 2.
   8.4. While on the Axis options, quickly point out some of the other pages like Labels and Label Fonts.
   8.5. Preview the report again.
   8.6. Now that we can see more labels, point out the sorting of months is incorrect.
   8.7. Return to Design mode.
   8.8. Click back in the main chart area.
   8.9. In Category Groups, click drop down by Month_Year.
   8.10. Click Sorting.
   8.11. Change to Month_Year_Key. Click OK to close.
   8.12. Preview, show good sorting, return to design.
9. Fix the vertical axis.
   9.1. Right click on the vertical axis.
   9.2. Pick Vertical Axis Properties.
   9.3. On the Axis Options page, uncheck Always Show Zero, but explain the pros and cons.
   9.4. Click Number.
   9.5. Set Decimal Places to 0, check on Use 1000 Separator, check on Show Values in Thousands.
   9.6. Right click again on the Vertical Axis, check on Show Axis Title.
   9.7. Click in the Axis Title.
   9.8. Enter Sales Excluding Tax in Thousands and hit enter.
   9.9. Point out the Horizontal Axis can have a title too, but in this case the labels of Month – Year make it clear what this axis represents, and thus becomes redundant.
10. Update Chart Title.
   10.1. Click in the Chart Title area. Rename to Sales per Buying Group.
   10.2. Right click to show you can remove the title if you want, as might be done when the chart is embedded in another report or web page.
   10.3. Also, point out the chart properties menu item, but don't open it.
11. Fix the Legend.
   11.1. Right click on the legend, and pick Legend Properties.
   11.2. Under general, point out Legend Position. Change to upper right.
   11.3. Close, and preview. Point out this now shrinks the chart area, so it is something of a tradeoff.
   11.4. DO NOT return to design
12. Summary
   12.1. Explain we now have a useful chart that conveys information in an easy to read format.

6 – Create a Report with Both a Chart and Line Item Listing in SSDT

This query will be used for this demo:

```
SELECT dc.[Buying Group]
     , dc.[Customer]
     , fs.[Delivery Date Key]
     , dd.[Calendar Year]
     , fs.[WWI Invoice ID]
     , fs.[Description]
     , fs.[Total Excluding Tax]
  FROM [Fact].[Sale] fs
 INNER JOIN [Dimension].[Date] dd
    ON fs.[Delivery Date Key] = dd.[Date]
 INNER JOIN [Dimension].[Customer] dc
    ON fs.[Customer Key] = dc.[Customer Key]
 INNER JOIN [Dimension].[Employee]  de
    ON fs.[Salesperson Key] = de.[Employee Key]
```

1. In this demo, we will create a report with both a chart and a table. Both areas will be sourced from the same dataset. It is possible to source them from totally different datasets, but it is more common to want to source them from the same spot.
2. Add the new empty report.
   2.1. Right click on Reports, Add, New Item.
   2.2. Pick Report, name it M06-ChartAndTable.rdl and click Add.
3. Add the data source.
   3.1. Right click on Data Sources, Add Data Source.
   3.2. Name it WideWorldImportersDW.
   3.3. Leave type at Microsoft SQL Server.
   3.4. Click Edit beside Credentitals.
   3.5. Server name localhost, database name of WideWorldImportersDW. Test, then OK to close.
   3.6. OK to close the data source.
4. Add the data set.
   4.1. Right click on Datasets, Add Dataset.
   4.2. Name it SalesData.
   4.3. Change to Use a dataset embedded in my report.
   4.4. Select WideWorldImportersDW as the data source.
   4.5. Mention that while the query designer seen in a previous module is available, many times report developers comfortable with SQL Server have already used SSMS to work out the query to retrieve the data. That's what we did here, so we'll just paste the query (found above) into the Query box.
   4.6. Click OK to close.
5. Add the Chart.
   5.1. Go to the Toolbox pane.
   5.2. Drag and drop a chart at the top. Select Column as the type. Click OK.

5.3. Mention that for a normal report we'd use the techniques shown in the Formatting a Report module to properly size the report. For this demo we'll just size it to something the viewer can see on the screen.
5.4. Click inside the chart to make the Chart Data dialog appear to the right of the chart.
5.5. Click + beside Values, select the Total_Excluding_Tax.
5.6. Click + beside Category Groups, select Calendar_Year.
5.7. Click + beside Series Groups, select Buying_Group.
5.8. Click Preview to validate the chart is correct. After doing so return to Design mode.

6. Add the table.
   6.1. Go back to the Toolbox pane, drag and drop a Table onto the report.
   6.2. Switch back to the Dataset pane.
   6.3. Drag the fields from the dataset onto the report in this order:
      6.3.1. WWI Invoice ID
      6.3.2. Delivery Date
      6.3.3. Buying Group
      6.3.4. Customer
      6.3.5. Description
      6.3.6. Total Excluding Tax.
7. Preview the report.
8. Explain we now have a report with both a table and chart sourced from the same dataset.
9. This technique could be used to place multiple charts, tables, etc. on a single report.

7 – Using Parameters in a Report

1. In this section, we'll see how to use parameters in a report.
2. First, we copied the report using the technique shown in the Formatting a report module. It was renamed to M07-Parameters and added to the project.
3. We then used techniques shown in that same module to format the table.
4. The chart was then formatted using techniques found in the Create a Chart Report with SQL Server Data Tools module.
5. First, we need to add two datasets to supply data to the parameters.
6. Add Buying Group dataset.
   6.1. Right click on Datasets, Add Dataset.
   6.2. Use BuyingGroup for the name.
   6.3. Change to Use a dataset embedded in my report.
   6.4. Use WideWorldImportersDW as the data source.
   6.5. Paste in the following query:

```
SELECT DISTINCT [Buying Group]
  FROM [Dimension].[Customer]
 ORDER BY [Buying Group]
```

   6.6. Click OK to close.
7. Add the BuyingGroup parameter.
   7.1. Right click on Parameters. Pick Add Parameter.
   7.2. Name it BuyingGroup.
   7.3. Prompt is Buying Group.
   7.4. Check on Allow multiple values.
   7.5. Click on Available Values.
   7.6. Select Get values from a query.
   7.7. Select BuyingGroup for the dataset.
   7.8. Pick Buying_Group for both the value and label.
   7.9. Explain the use of Default Values, but we won't sent one here.
8. Tie BuyingGroup to the Query.
   8.1. Right click on the SalesData dataset. Pick Dataset Properties.
   8.2. Click on the Parameters page.
   8.3. Click Add.
   8.4. Enter @BuyingGroup for the ParameterName.
   8.5. Pick BuyingGroup in the Parameter Value drop down.
   8.6. Click on the Query page.
   8.7. At the end of the query, append
   8.8. WHERE dc.[Buying Group] IN (@BuyingGroup)
   8.9. Click OK.
   8.10. Preview the report. Show the effect of the parameter.
9. Add Year dataset.
   9.1. Right click on Datasets, Add Dataset

9.2. Use CalendarYear as the name.
9.3. Change to Use a dataset embedded in my report.
9.4. Use WideWorldImportersDW as the data source.
9.5. Paste in the following query:

```
SELECT DISTINCT [Calendar Year] FROM [Dimension].[Date] ORDER BY [Calendar Year]
```

9.6. Click OK to close.

10. Add the Year Parameter.
    10.1. Right Click on Parameters, Add Parameter.
    10.2. Name DeliveryYear
    10.3. Prompt Delivery Year
    10.4. Check on Allow multiple values
    10.5. Go to the Available Values page.
    10.6. Set to Get values from a query.
    10.7. Dataset is CalendarYear
    10.8. Value and label is Calendar_Year
    10.9. Click Default Values
    10.10. Select Get values from a query
    10.11. Dataset is CalendarYear
    10.12. Value is Calendar_Year.
    10.13. Click OK.
11. Add the Year to the Query.
    11.1. Right click on the SalesData dataset. Pick Dataset Properties.
    11.2. This time show we're doing it by adding to the query first.
    11.3. At the bottom of the query, enter
    11.4. AND dd.[Calendar Year] IN (@DeliveryYear)
    11.5. Click on Parameters.
    11.6. Point out that DeliveryYear was added for us. In the drop down, pick @DeliveryYear.
    11.7. Click OK.
12. Preview the report. Show years has default. Pick various items to show the effect on the report.
13. Return to Design.
14. Arrange the parameters, explaining this is a 2016 only feature, and rerun.
15. Now lets see what happens if all parameters have default values.
    15.1. Right click on the BuyingGroup parameter, pick Parameter Properties.
    15.2. Go to Default Values.
    15.3. Get values from a query.
    15.4. BuyingGroup for Dataset, Buying_Group for Value.
    15.5. Click OK.
    15.6. Rerun the report to show it now runs automatically.
16. Add a final parameter, minimum total excluding tax to include.
    16.1. Right click on Parameters, Add Parameter.
    16.2. Name MinTotalExcludingTax
    16.3. Prompt Minimum Total Excluding Tax
    16.4. Set Data Type to Integer.
    16.5. Go to Default Values.

16.6. Pick Specify Values.
16.7. Click Add.
16.8. Enter a value of 0.
16.9. Click OK.

17. Add the parameter to the query.
    17.1. Right click on the SalesData. Pick Dataset Properties.
    17.2. At the bottom of the query, add:
    AND fs.[Total Excluding Tax] >= @MinTotalExcludingTax
    17.3. Click on the Parameters page.
    17.4. Pick the @MinTotalExcludingTax to assign to the new parameter.
    17.5. Click OK.
    17.6. Preview the report.
    17.7. Enter a value in the new parameter of 100, and View Report. Display Results.
18. Summary
19. Go over some pros and cons.
20. Only set defaults if you are reasonably sure these are the values that will be used the majority of the time to execute a report.
21. Be aware very large lists of values to pick from can cause issues.

8 – Creating a Shared Dataset in SSDT

We will use this query:

```
SELECT fs.[WWI Invoice ID]
     , fs.[Invoice Date Key]
     , dc.[Customer]
     , fs.[Description]
     , fs.[Quantity]
     , fs.[Total Excluding Tax]
  FROM Fact.Sale fs
 INNER JOIN Dimension.Customer dc
    ON fs.[Customer Key] = dc.[Customer Key]
```

1. In this section, we'll create a shared dataset, as well as a shared data source. Shared datasets can be useful for business users to base reports on, and as a way to create datasets for frequently used parameters.
2. Shared datasets need a shared data source, so we'll create that first.
   2.1. Right click on Shared Data Sources in the project, pick Add New Data Source.
   2.2. Enter WideWorldImportersDW as the name.
   2.3. Leave the type as Microsoft SQL Server.
   2.4. Click Edit by connection string.
   2.5. localhost for the server name, WideWorldImportersDW for the database.
   2.6. Test, then OK.
   2.7. OK again to close the data source.
   2.8. Show the data source in the project.
3. Now we will add the dataset.
   3.1. Right click on Shared Datasets. Pick Add New Dataset.
   3.2. Name the dataset SalesSourceData.
   3.3. Point out the Data source comes from the set of Shared Data Sources.
   3.4. Paste in the query at the top as the query.
   3.5. Click OK to close.
   3.6. Show it in the solution.
4. Summary
   4.1. Explain we'll show how to use it in the next module.
   4.2. Also, mention more info will be used in the module Deploying Reports to the Report Portal from SSDT.

9 – Creating a Report using a Shared Dataset

1. In this quick module, we'll create a report based on the Shared Dataset in the previous module.
2. Right click on Reports, and pick Add, New Item.
3. Pick Report, name it M09-Shared, and Add.
4. Add the shared data source.
   4.1. Right click on Data Sources, pick Add Data Source.
   4.2. Name the data source WideWordImportersDW.
   4.3. Click on Use shared data source reference.
   4.4. Pick WideWorldImportersDW and click OK.
   4.5. Point out that you can use a shared data source, but your data set can be embedded. There can be good reasons for this, mostly the ability to change a data source when a report is deployed.
   4.6. Click OK.
5. Add the shared data set.
   5.1. Right click on Datasets, Add New Dataset.
   5.2. Name it SalesSourceData.
   5.3. Note that Use a shared dataset is defaulted.
   5.4. Point out the shared dataset is in the list.
   5.5. Click it, then OK.
6. From here it's like any other report. Add a table.
   6.1. In the Toolbox pane, add a table.
   6.2. Drag fields over, but leave one or two out just to show you don't have to use them all.
   6.3. Preview the report.

10 – Deploying Reports to Report Portal from SSDT

1. Once you have completed your reports, you'll want to deploy them to the Report Portal so your users can have access to them. In SQL Server 2016 this is done in the report portal, in previous versions this would be the Report Manager. While the look has been updated for 2016, the steps to deploy are the same regardless of the version.
2. By default, you'll find the Report Manager/Portal at the server name / reports. To find the exact address, open the Reporting Services Configuration Manager, located in the Windows menus for SQL Server menus under the sub menu Configuration Tools. (Switch to RSCM).
3. The address of the users portal can be found in the Web Portal URL area. As you can see, it is the name of my server followed by Reports. This is the address users will go to in order to use your reports.
4. We'll need one other web address though for SSDT to deploy reports to. This is the Web Service URL (click on it). As you can see, it is the name of the server followed by ReportServer.
5. (Switch to it in IE). Right now we don't see anything, as we've not deployed anything to our server yet. Let's fix that now. Before leaving, copy the URL into the clipboard.
6. (Switch to SSDT with the Project open.)
7. Right click on the project, and pick Properties. Make sure it's the project, and not the solution.
8. Go over each item under the Deployment area. Cover TargetServer items first, then the overwrite fields.
9. Click OK to close.
10. First, make sure the output window is visible.
11. Now right click on the project again, and pick Deploy.
12. Go to the site. Show the new folders, run a few reports.
13. Go to the Data Sources folder, show it and explain why it is good to have reports sourced from a shared data source.
14. Show the Dataset, explain how it works.

11 – Deploying Reports to Report Portal from the File System

1. As a precursor to this exercise, copy the report M03-Format.rdl as M11-Upload.rdl.
2. Navigate to the folder where you want the report.
3. Click Upload on the menu.
4. Select the file M11-Upload.rdl
5. Click the … beside the report.
6. Select Manage on the menu.
7. Go to Data Sources.
8. Validate the data source is accurate. Use Test Connection to validate.
9. Go back and execute the report.