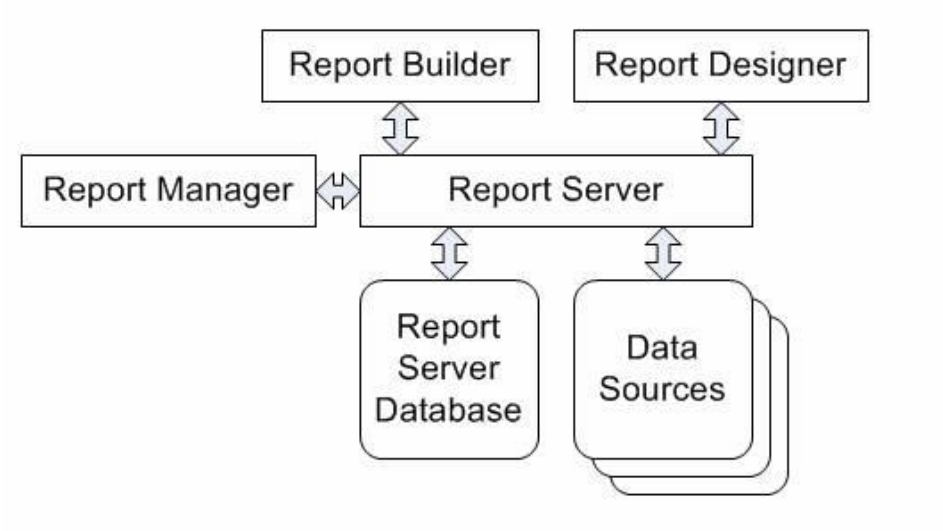
Hong Kong Baptist University
Department of Computer Science

COMP 7810/4096 Business Intelligence (2019-20)

SQL Server Reporting Services (SSRS) and Power BI

Introduction

Reporting Services has a quite complex architecture. The full Reporting Services architecture includes development tools, administration tools, and report viewers. The following shows a simplified diagram of the main Reporting Services components that will be used.



- **Report Server** is the core engine that drives Reporting Services.
- **Report Manager** is a Web-based administrative interface for Reporting Services.
- **Report Designer** is a developer tool for building complex reports.
- **Report Builder** is a simplified end-user tool for building reports.
- **Report Server database** stores report definitions. Reports themselves can make use of data from many different data sources.

Reporting Services includes two tools for creating reports:

- **Report Designer** can create reports of any complexity that Reporting Services supports, but requires you to understand the structure of your data and to be able to navigate the Visual Studio user interface.
- **Report Builder** provides a simpler user interface for creating ad hoc reports, directed primarily at business users rather than developers. Report Builder requires a developer or administrator to set up a data model before end users can create reports.

Learning Outcomes

By finishing this lab session, you should be able to

- create reports using SSRS for the OLAP cube
- create reports using SSRS for a relational database
- create report using PowerBI

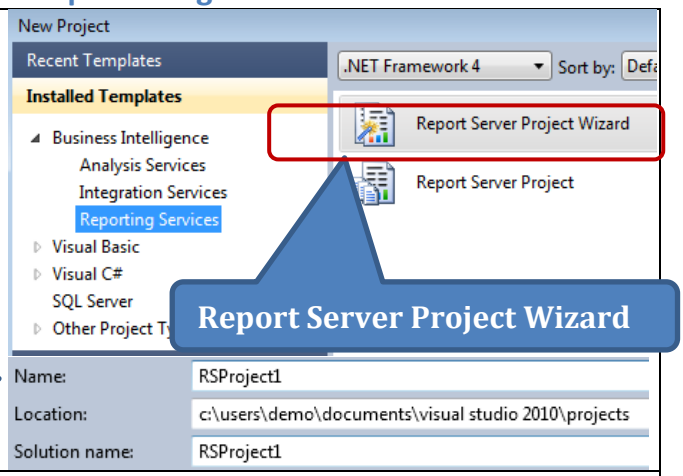
Tools

- Microsoft SQL Server Management Studio 2012
- Visual Studio 2010 with SQL Server Data Tools (SSDT)

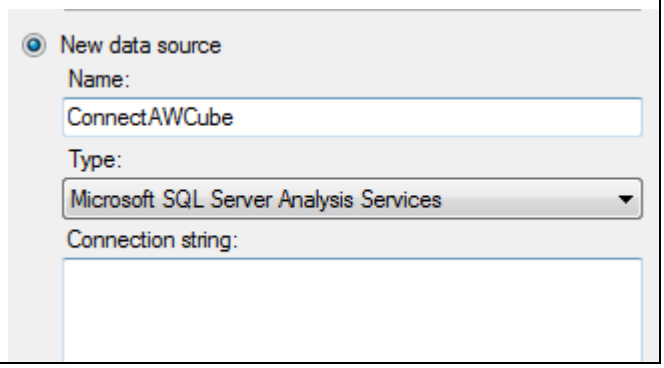
Part A: Generating reports using SSRS

I. Using Report Wizard to create the first report using cube data

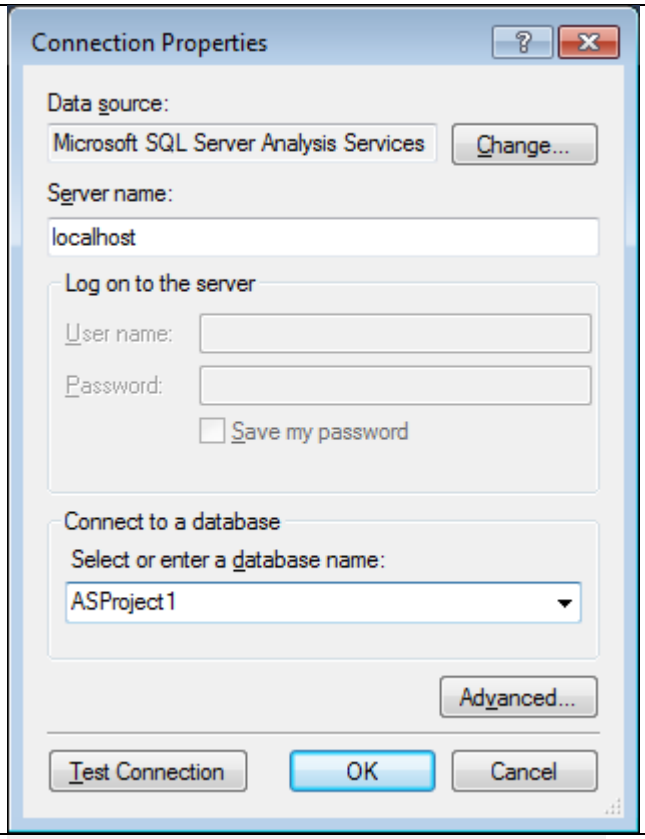
1. Open the **Visual Studio with SSDT.**
2. Select **File → New → Project.**
3. Expand **Business Intelligence → Reporting Services**, and then click **Report Server Project Wizard**.
4. Change the project name to **RSProject1**. Press **OK**.



5. Click **Next** in the welcome page.
6. In the Select the Data Source page, change the data source name to **ConnectAWCube**, then change the Type to **Microsoft SQL Server Analysis Services**
7. Click the **Edit** button.

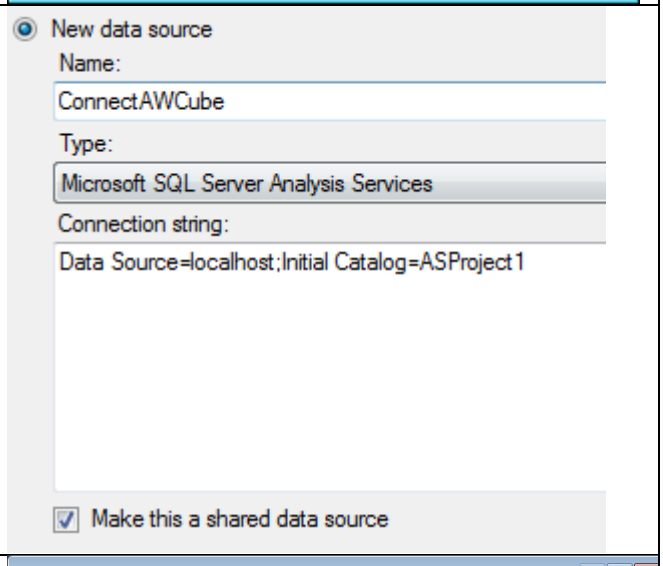


8. Connect to **localhost** and select the **ASProject1** project. And click **OK**.

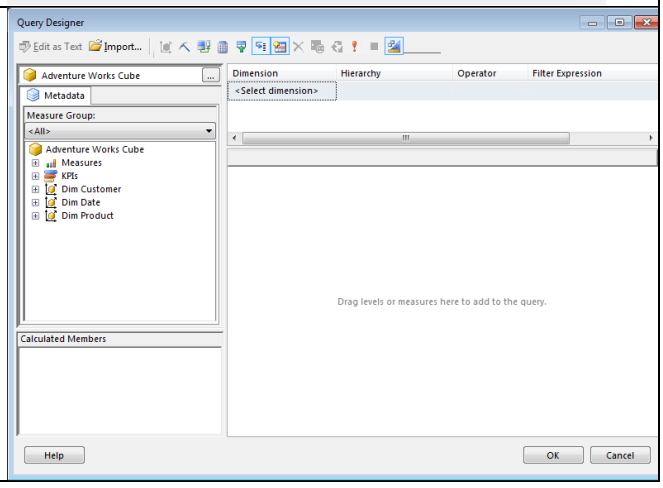


9. Check the box **Make this a shared data source**.

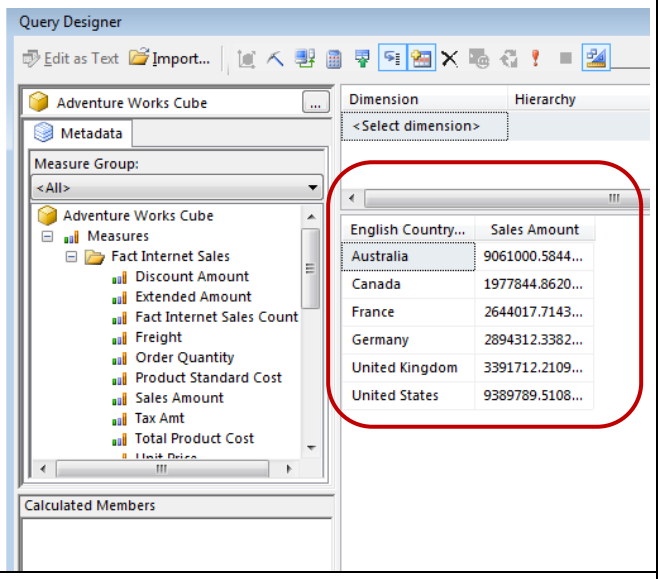
10. Click **Next**.



11. Click the **Query Builder** button. It shows a pane to build your queries from the cube **Adventure Works Cube.**



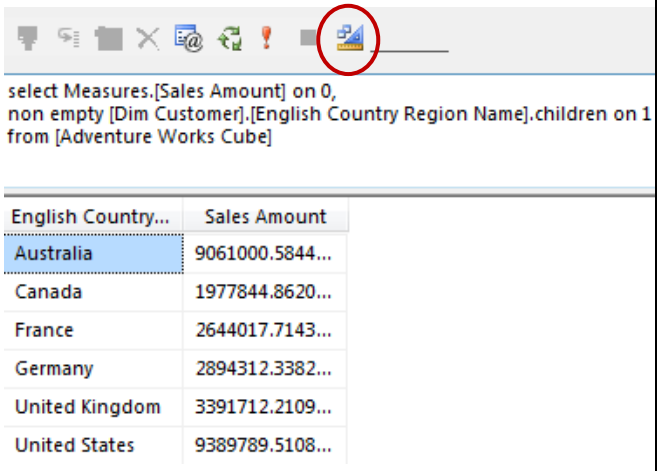
12. To show sales amount for different countries, drag the field **Sales Amount** and **English Country Region Name** to the right hand side.



13. Or you click **Design Mode** once, and type the MDX directly.

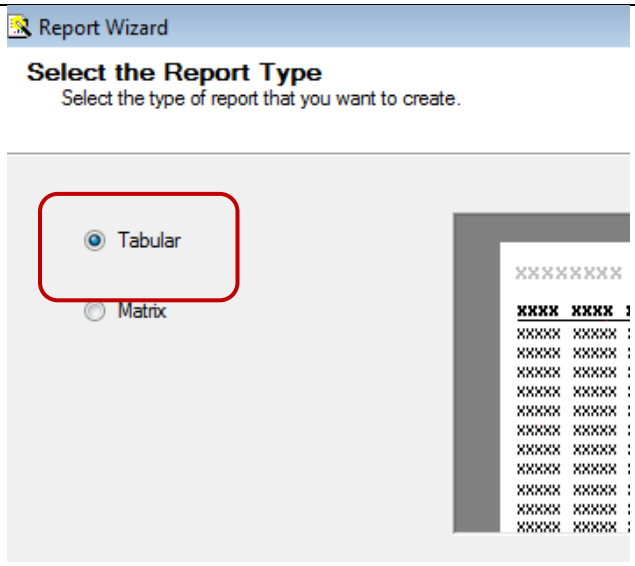
14. Execute the query by clicking button.

15. Press **OK** and then **Next** button to continue.

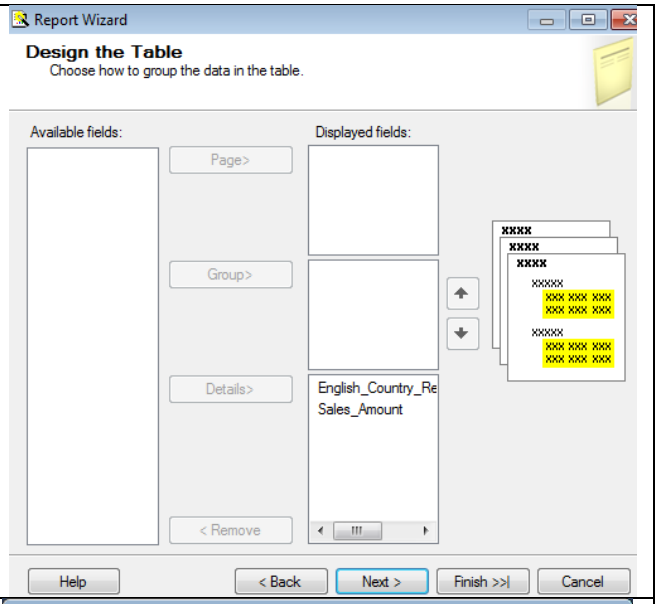



```
select Measures.[Sales Amount] on 0,
non empty [Dim Customer].[English Country Region Name].children on 1
from [Adventure Works Cube]
```

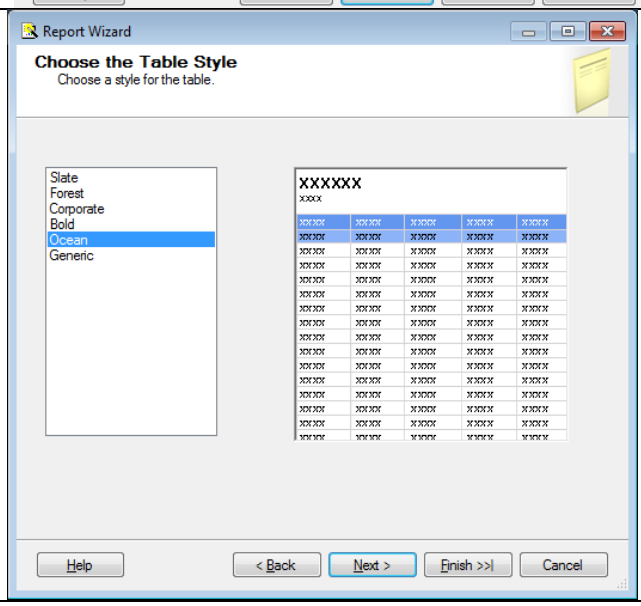
16. Select **Tabular** as the report type. And press **Next** button.



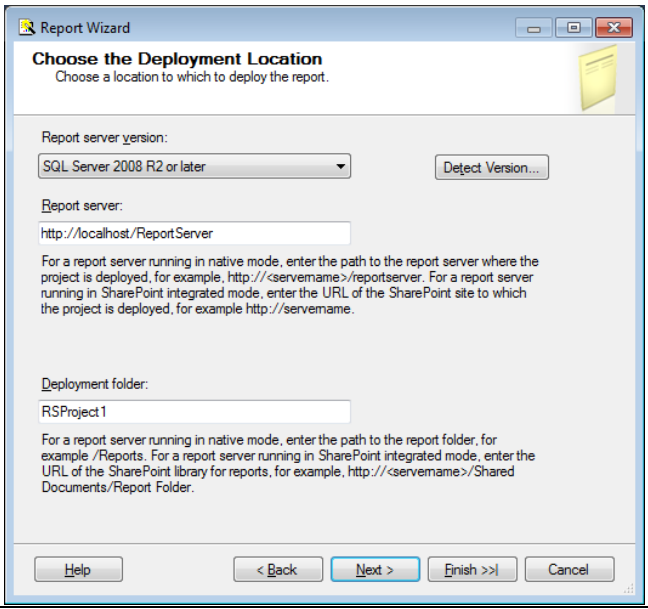
17. Move the columns **English_Country_Region_Name** and **Sales_Amount** to the **Details** area. Click **Next**.



18. Select **Ocean** as the table style and click **Next**.



19. Accept the default deployment settings and click **Next**.



20. Name the report **Sales Amount by Country**. And check the box **Preview Report**. Then click **Finish**.

21. The report showing the MDX result you typed before.

22. You may **adjust** the **column width** using **Design** tab. And format the sales amount to **Number** with **2 d.p.**

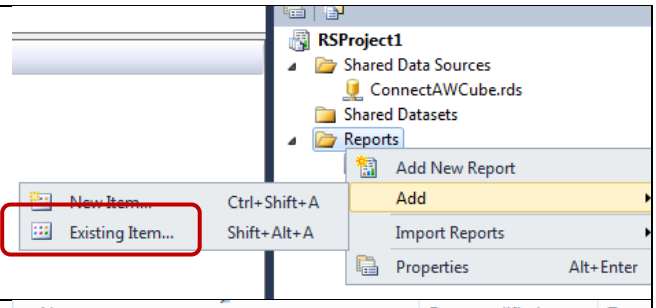
23. Select **File → Save All** to save the changes.

Sales Amount by Country

English Country Region Name	Sales Amount
Australia	9061000.58
Canada	1977844.86
France	2644017.71
Germany	2894312.34
United Kingdom	3391712.21
United States	9389789.51

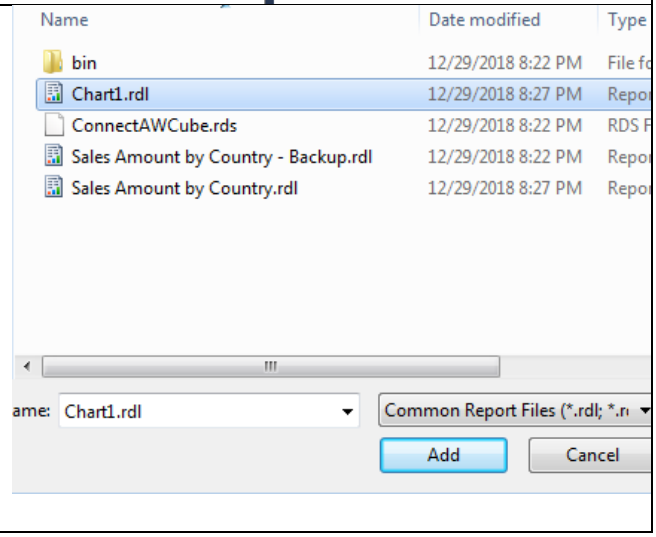
II. Create a chart to visualize the result

1. Right click **Reports** folder in **Solution Explorer**, select **Add → Existing Item**



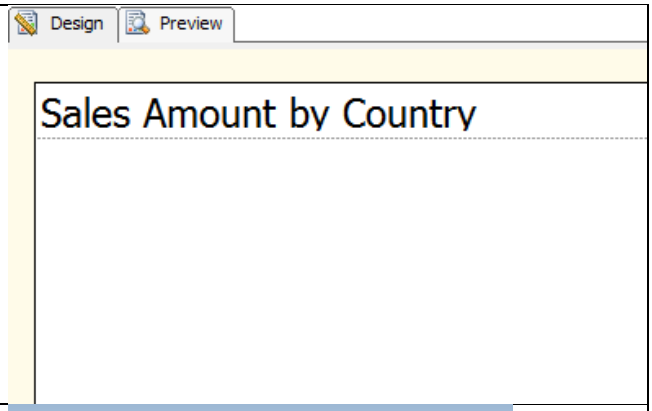
2. **Duplicate** the previous report (rdl file) and paste it in the same directory, rename the copy to **Chart1.rdl**

3. Select it and click **Add** to add the second report.

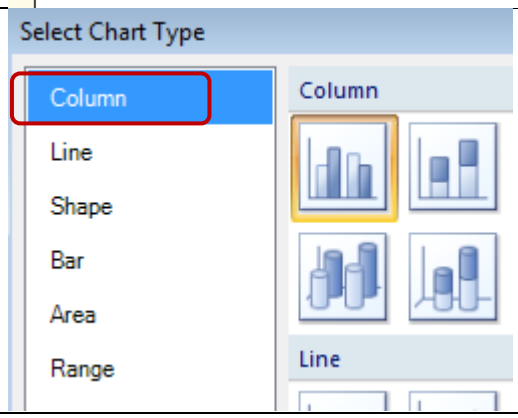


4. **Double click** to open the **Chart1.rdl** in *Solution Explorer* using **Design** tab. (You may close the previous report)

5. **Delete** the two fields. Adjust the area of the report for placing a chart.

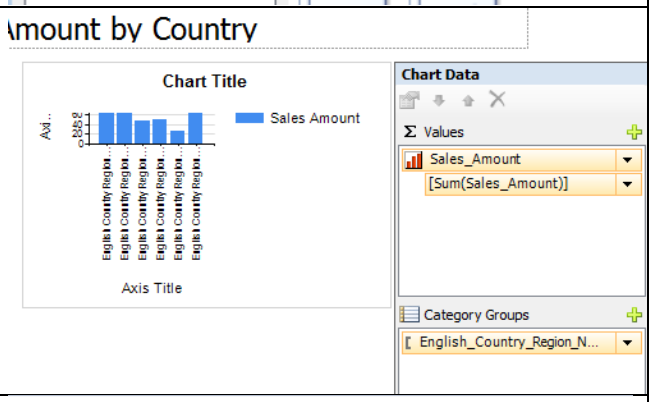


6. Select **View → Toolbox.** Select Chart option and click on the area under report title. Select **Column** chart and click **OK**.



7. **Double click** the chart and add **Sales_Amount** as **Values**. It changes to Sum(Sales_Amount).

8. Select **English_Country_Resgion_Name** as Category Groups.

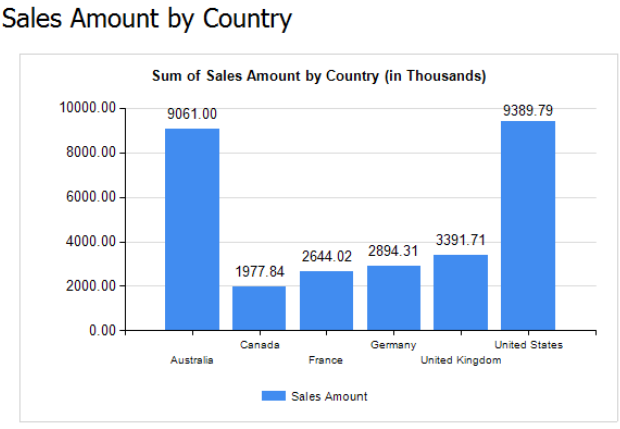


9. Modify the charts according to the following:

    1. Place the legend at the **bottom**
    2. Change chart title to **Sum of Sales Amount by Country (in Thousands)**
    3. **Delete** x and y axis titles
    4. Add **Data label** to each column, format it to **Number with 2 d.p.**
    5. Show values in **Thousands**

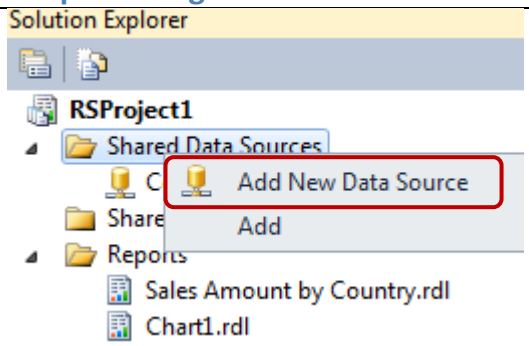
10. Preview the chart, it should be similar to this one.

11. Select **File → Save All** to save all the changes.



III. Using Report Wizard to create another report using DB data

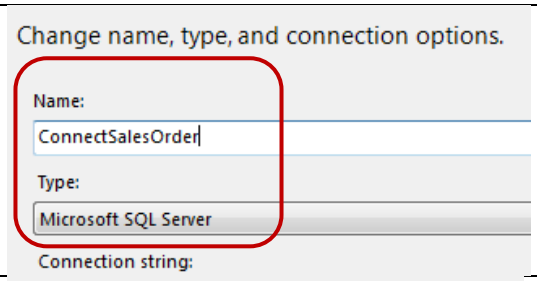
1. Right click **Shared Data Sources** in the *Solution Explorer.* Choose **Add New Data Source**



2. Add a new data source with the name **ConnectSalesOrder**

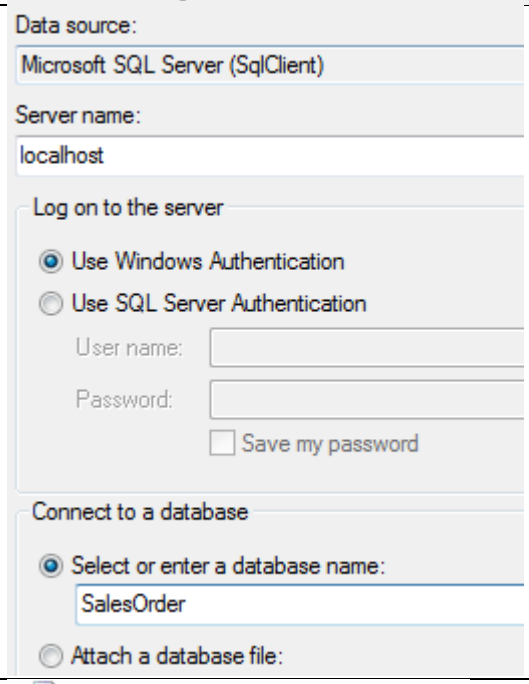
3. **Type** is **Microsoft SQL Server.** Click **Edit.**

   (a relational DB)

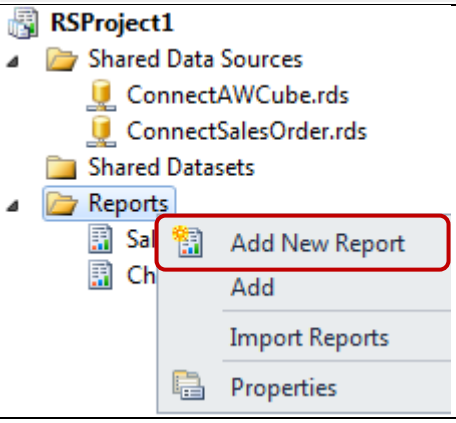


4. Set Server name as **localhost**, and Select **SalesOrder** DB.

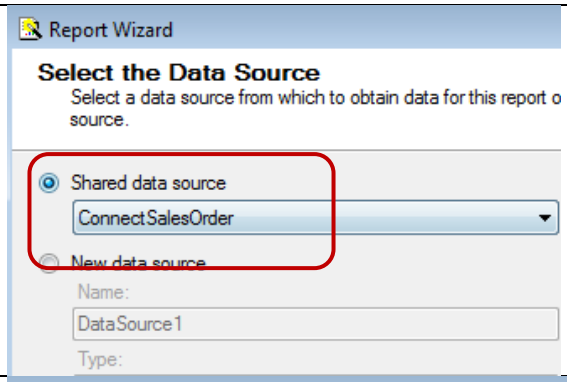
5. Press **OK** and **OK**.



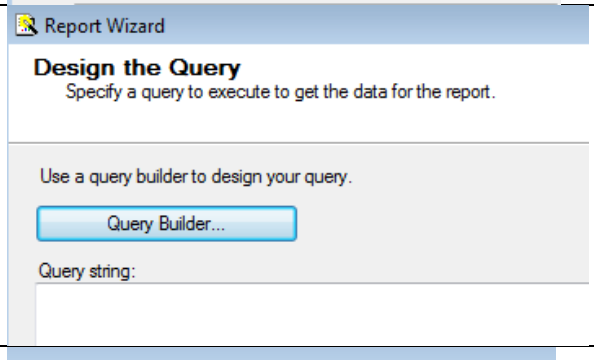
6. Right click **Reports** folder → choose **Add New Report**. Press **Next**.



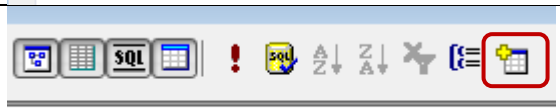
7. Select the new data source **ConnectSalesOrder**. Press **Next**.

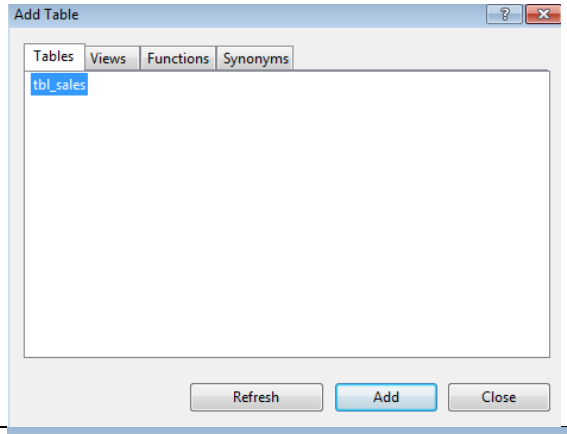


8. Click **Query Builder.** You can see the Query Designer is quite different.



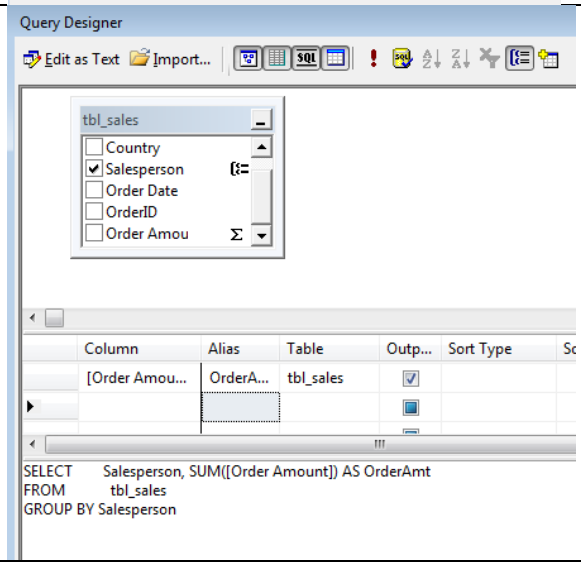
9. **Add** the table **tbl_sales** for generating reports



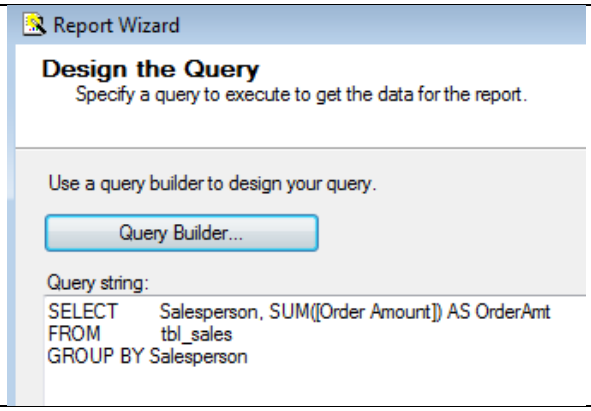


10. In the **Query Designer** you cannot type MDX but only SQL, or you can select the fields you want to display. Press **Execute** to preview the result.

```
SELECT Salesperson, SUM([Order Amount]) AS OrderAmt
FROM tbl_sales
GROUP BY Salesperson
```



11. Press **OK** then **Next**

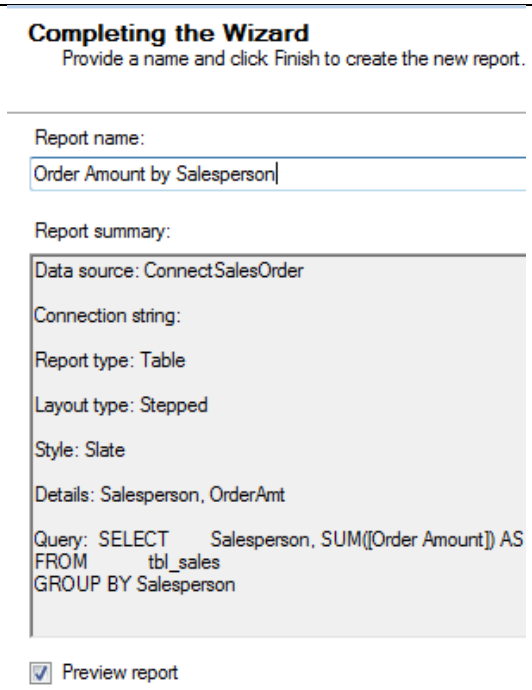


12. Select **Tabular** layout, press **Next**, move the two fields to **Details** section. Go to the **Next** step to select the table style you like. Then press **Next**

13. Name the report as **Order Amount by Salesperson**

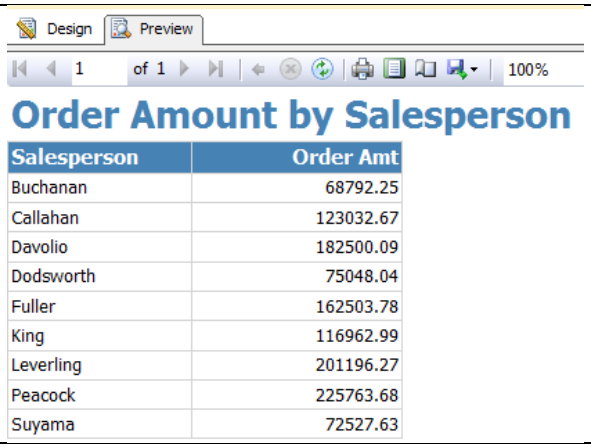
14. Check the box **Preview report**

15. Then press **Finish.**



16. **Adjust the field width** using **Design** tab.

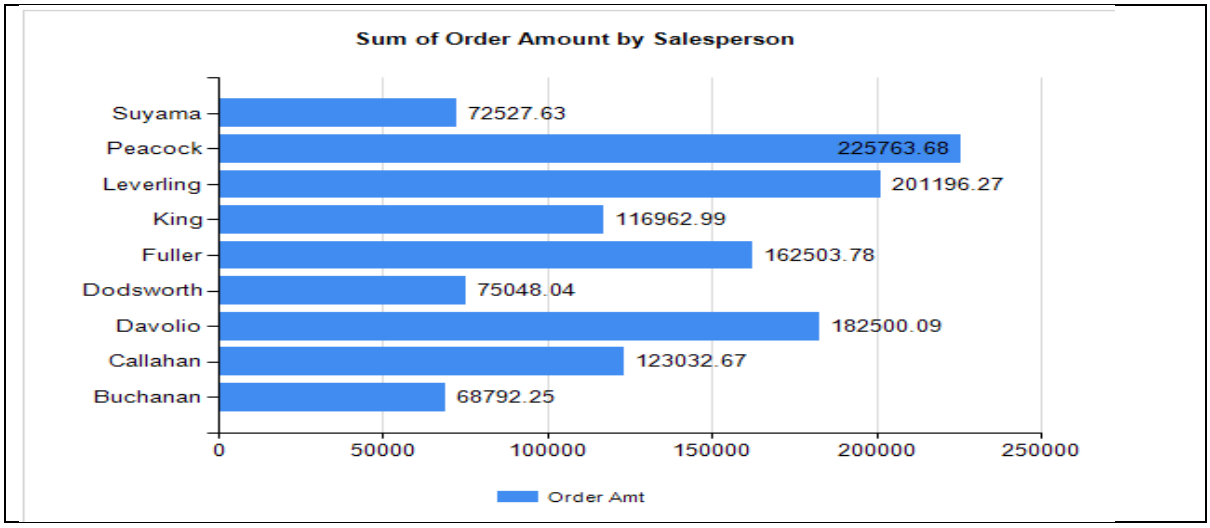
17. Select **File** ➔ **Save All** to save all the changes. And **close** the **Visual Studio** program



Salesperson	Order Amt
Buchanan	68792.25
Callahan	123032.67
Davolio	182500.09
Dodsworth	75048.04
Fuller	162503.78
King	116962.99
Leverling	201196.27
Peacock	225763.68
Suyama	72527.63

18. Create a 2D bar chart like this.
✓ Insert appropriate chart title
✓ Place the legend at the bottom
✓ Show data labels on the right

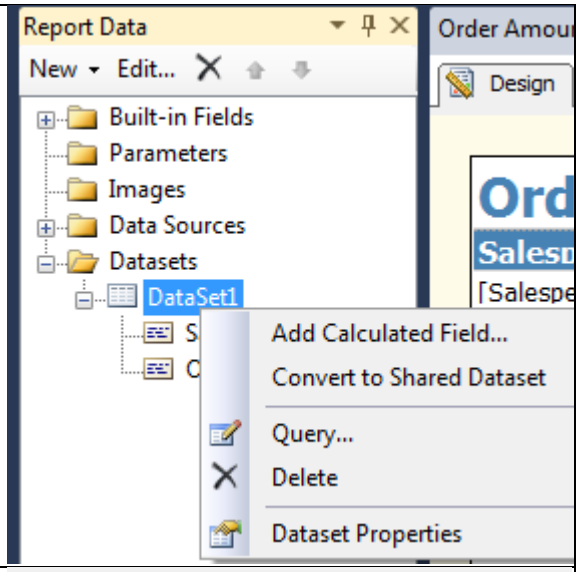
(see next page)



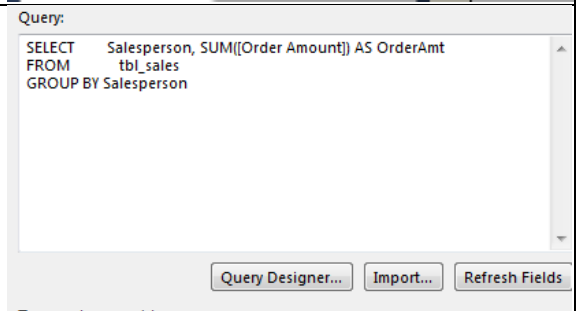
IV. Using Report Designer to modify the query

1. Click the **Design** tab to make the report editable.

2. In the **Report Data** window, expand **Datasets** field, **right click** on **DataSet1** and select **Dataset Properties.**

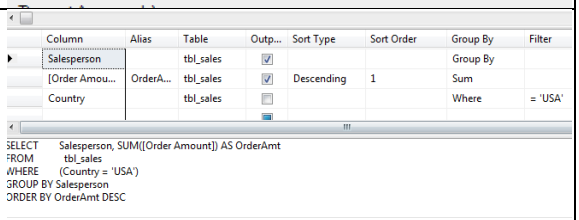


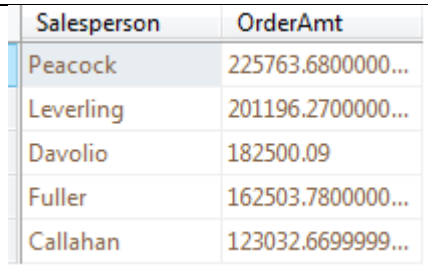
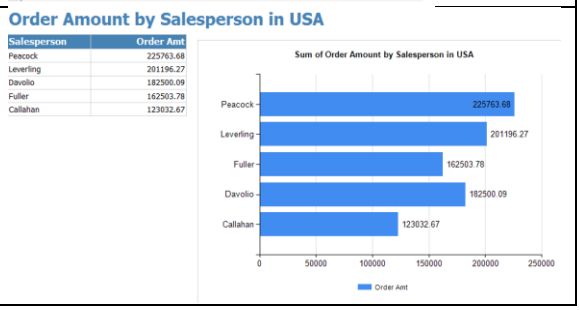
3. Press **Query Designer** button to edit the query



4. Change the SQL statements by adding **Order by** and **Where** clauses.

```
SELECT      Salesperson, SUM([Order Amount]) AS OrderAmt
FROM        tbl_sales
WHERE       (Country = 'USA')
GROUP BY Salesperson
ORDER BY OrderAmt DESC
```



5. Press **Execute** to preview the result. Press **OK** and **OK**	
6. **Preview** the report and chart. The report and chart are updated. 7. You may change the chart title. 8. Select **File** ➔ **Save All** to save all the changes.	

Part B: Generating reports using Power BI

Power BI is a data visualization and business intelligence tool that converts data from different data sources to interactive dashboards and BI reports.

Power BI Desktop is the Windows-desktop-based application, primarily for designing and creating reports. **Power BI Services** (a cloud service) is used to publish reports and data visualizations, while **Power BI mobile app** is used to view the reports and dashboards.

Some advantages of using Power BI:

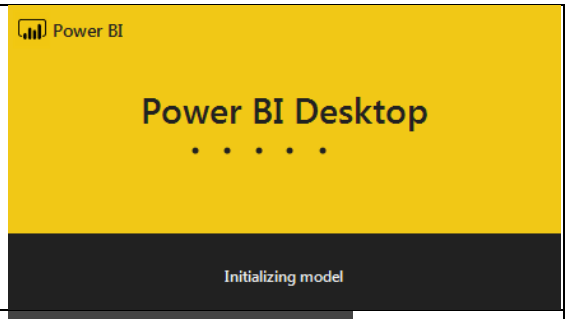
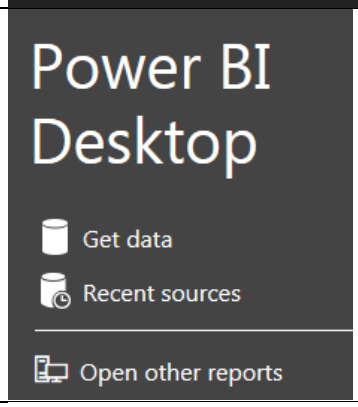
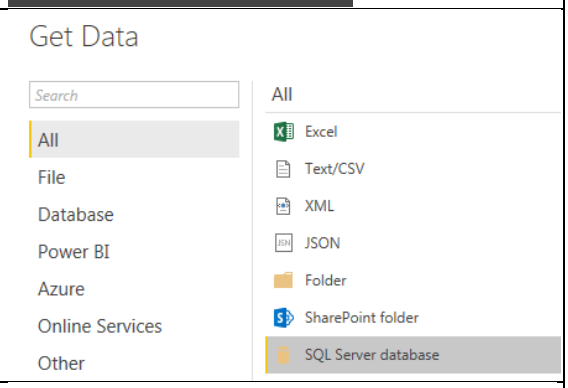
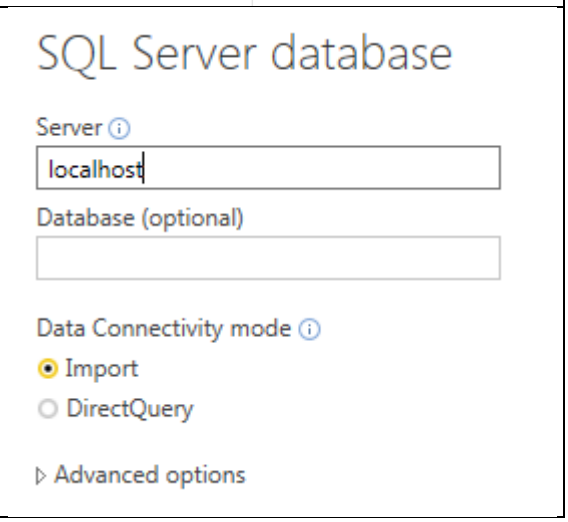
- Present visually, eye-catching
- Building Power BI reports is **faster** and **easier** than building SSRS reports
- Power BI **enable mobile reporting**
- Power BI has **more graphical component** as compared to SSRS

(Other comparison between SSRS and Power BI:
https://www.educba.com/power-bi-vs-ssrs/)

There are some important building blocks in Power BI:

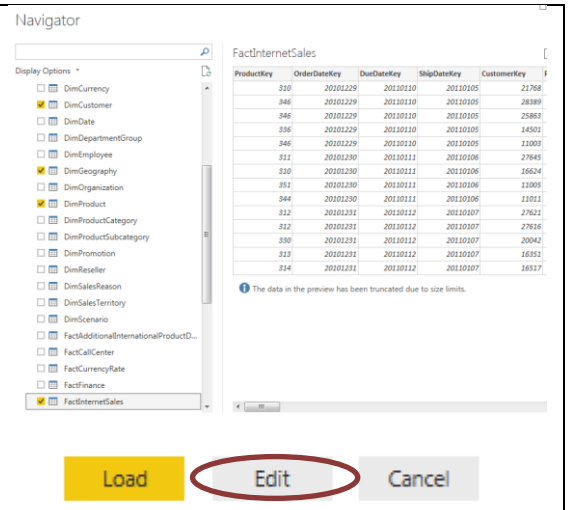
- **Visualizations** - is a type of chart built by Power BI designers
- **Reports** - is one or more pages of interactive visuals, text, and graphics that together make up a single report
- **Dashboards** - is a single screen with interactive visuals, text, and graphics

Don't use Power BI to replace Excel pivot tables and SSRS report, complement them.

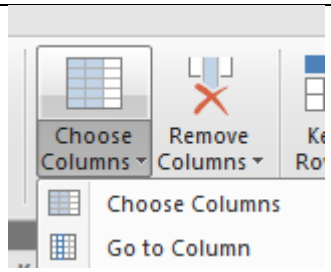
1. Open **Power BI Desktop**. You may skip the login.	
2. Select **Get data**	
3. Select **SQL Server database**. Press **Connect**.	
4. Use **localhost** as server. And press **OK.** (the sample Adventure Works DW database is hosted on SQL Server on the same machine)	

5. Expand **AdventureWorksDW2012**. Select **DimCustomer**, **DimGeography** and **DimProduct**, and **FactInternetSales**

6. Then press **Edit** to edit the data before loading to the Power BI.

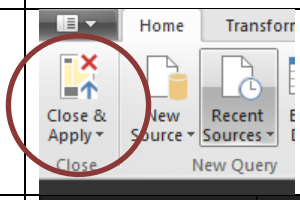


7. Remove some columns that are not necessary by using **Choose Columns.** Just remain the following in the tables as shown below:
   a) **FactInternetSales** (choose 5 columns):
      ProductKey,
      OrderDateKey,
      CustomerKey,
      OrderQuantity,
      SalesAmount
   b) **DimGeography** (choose 3 columns):
      GeographyKey,
      City,
      EnglishCountryRegionName,
   c) **DimCustomer** (choose 4 columns):
      CustomerKey,
      GeographyKey,
      MaritalStatus,
      Gender
   d) **DimProduct** (choose 3 columns):
      ProductKey,
      Color,
      ProductLine

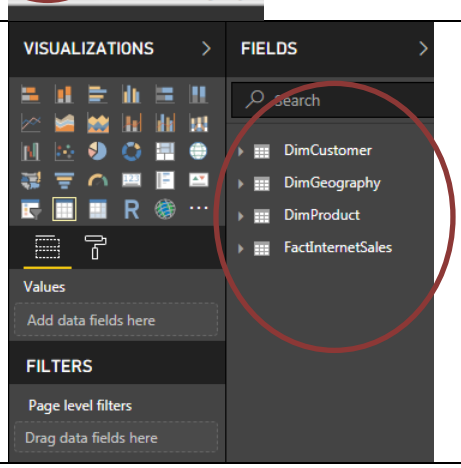


(The steps will be recored in **Query Settings**)

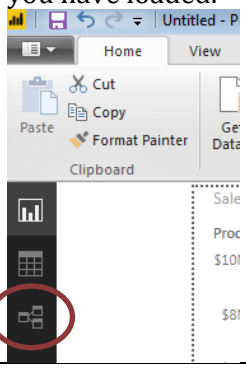
8. Select **Close & Apply** under **Home** tab.

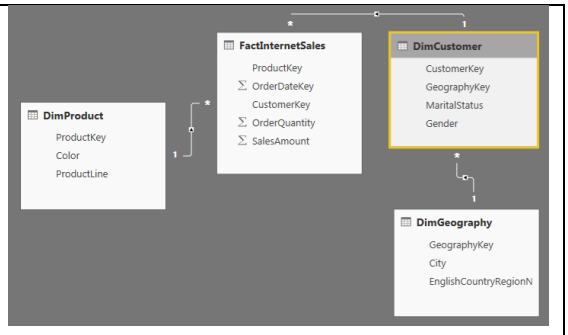


9. On the right hand side, you have got a **FIELDS** pane with 4 tables. You can drag and drop the elements from this pane onto the **Report** View.

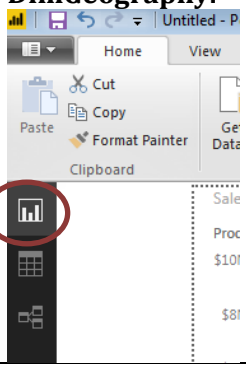


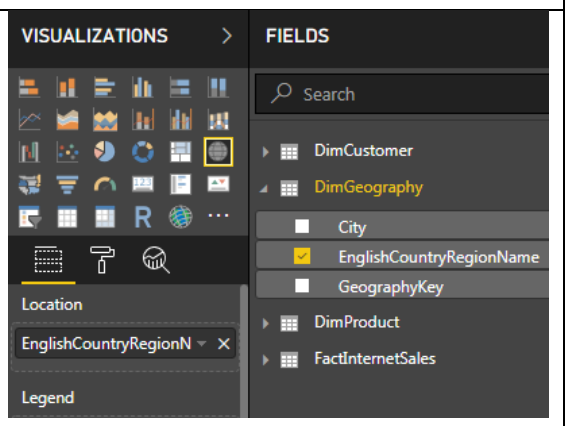
10. Select **Relationships** button on the left to display the relationships between the tables you have loaded.



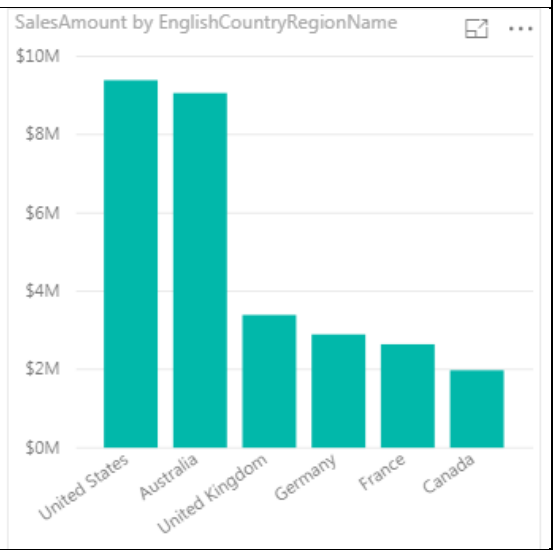


11. Go back to **Report** view. There are some tables in **FIELDS** pane. Select **SalesAmount** in **FactInternetSales**, then select **EnglishCountryRegionName** in **DimGeography**.

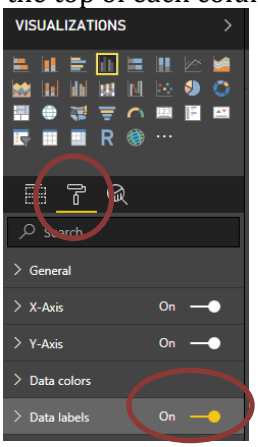


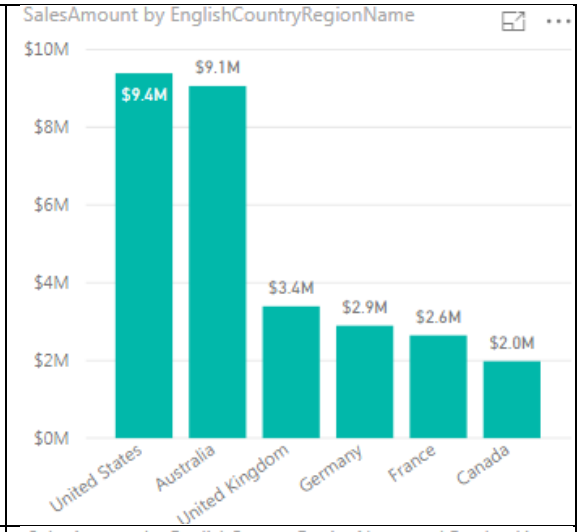


12. It shows a column chart in the canvas. The chart type is clustered column.

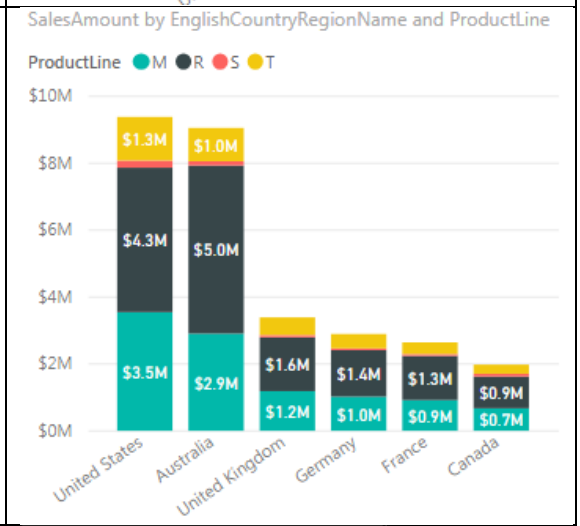


13. Select **Format** button to **show data labels** at the top of each column.

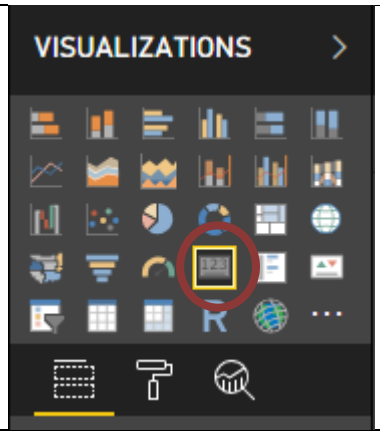




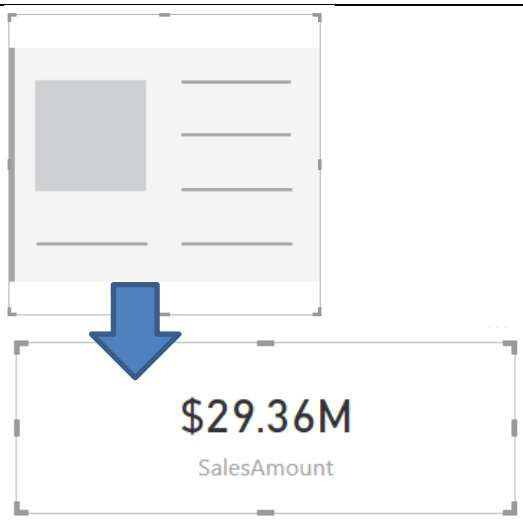
14. Add **ProductLine (**from **DimProduct)** to **Legend** and change the chart type to **stacked column chart**. (This visual is better as you create a columnar chart where each column has different colors for different product lines stacked on top of each other)



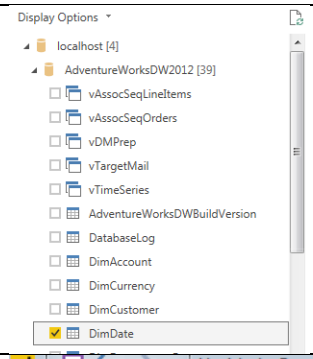
15. Click any blank area inside the canvas to create another visual - **Card**. Then select **SalesAmount (**from **FactInternetSales)** in **FIELDS** list. Now, it shows the total sum of sales.



16. The result is like this:

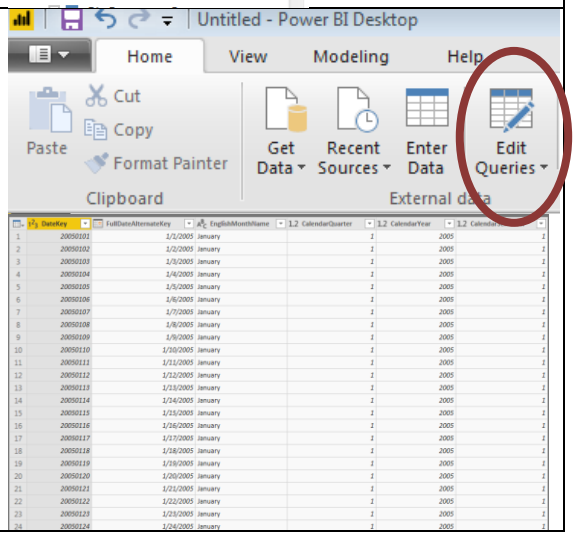


17. Add another table to the **FIELDS** pane by using **Get Data ➔ SQL server** (server: **localhost**), select **DimDate** and load the table directly to the model by pressing **Load** button.



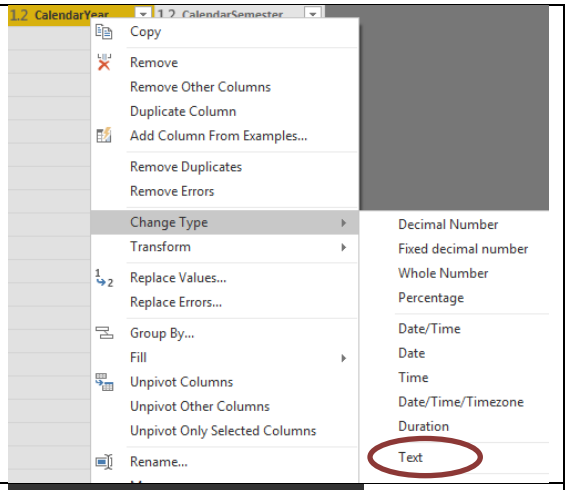
18. Select **Home ➔ Edit Queries**, select **Dim Date** to edit

19. Remove all the fields except the following:
    - DateKey
    - FullDateAlternateKey
    - EnglishMonthName
    - CalendarQuarter
    - CalendarSemester
    - CalendarYear

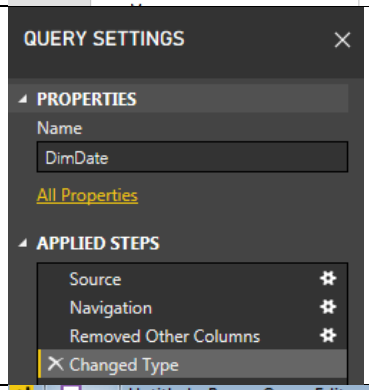


20. **Change** the **datatype** for **CalendarYear** to **Text** by right clicking the column **CalendarYear → Change Type → Text**

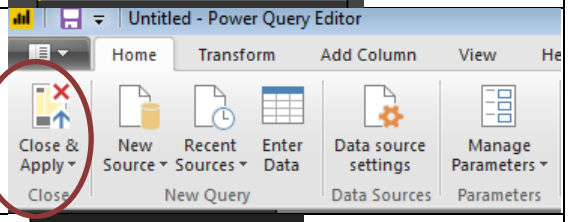
21. Repeat the previous step to change the datatype for **CalendarSemester** and **CalendarQuarter** to **Text**.



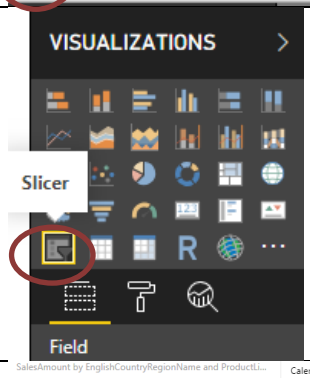
22. The procedures will be stored in QUERY SETTINGS (APPLIED STEPS) on the right.



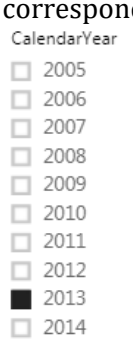
23. Select **Close & Apply** to apply the changes.

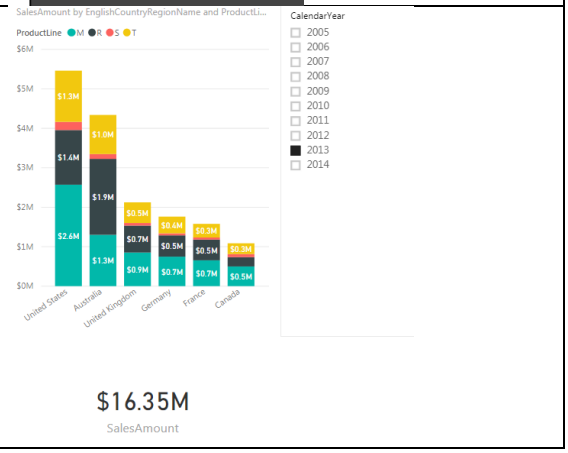


24. Click any blank area on canvas. Select the field **CalendarYear (**from **DimDate)** in the **FIELDS** list and change the visual type to **Slicer** in **VISUALIZATIONS** pane.

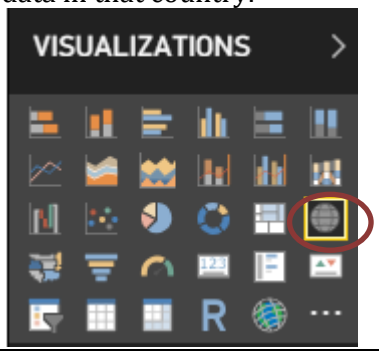


25. Select **2013** as slicer, the total sum and the stacked column chart will be changed correspondingly.

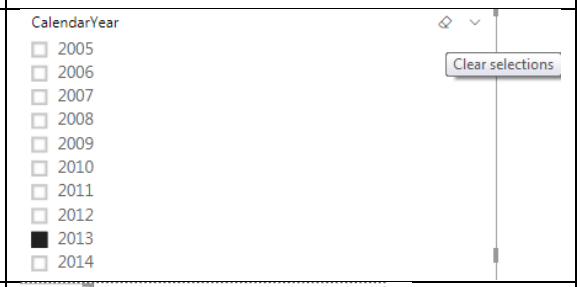




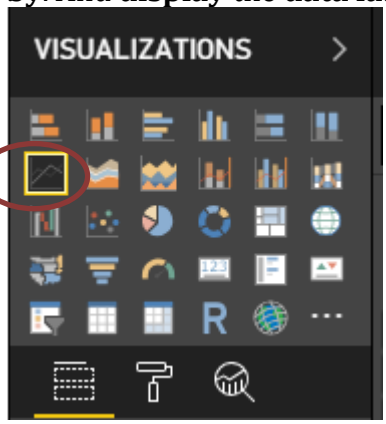
26. Click any blank area on the canvas to create another visual. Select **Map**. Select **EnglishCountryRegionName (**from **DimGeography)** in the **FIELDS** list. Select a certain country to display the corresponding data in that country.

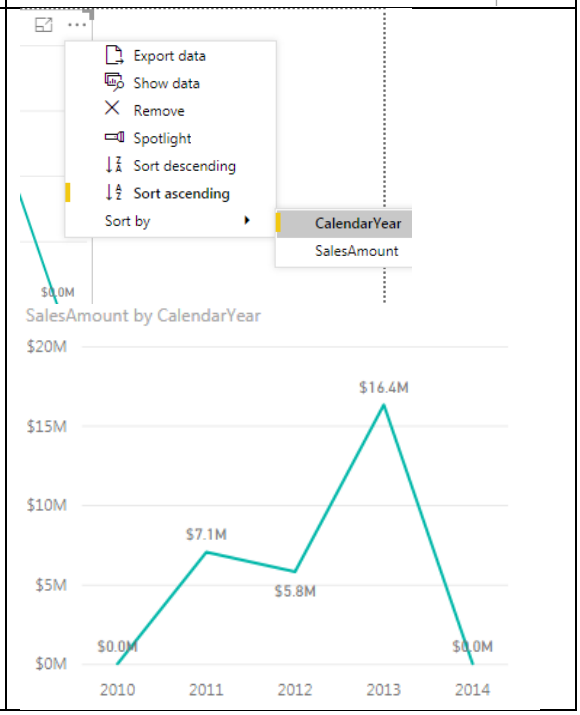


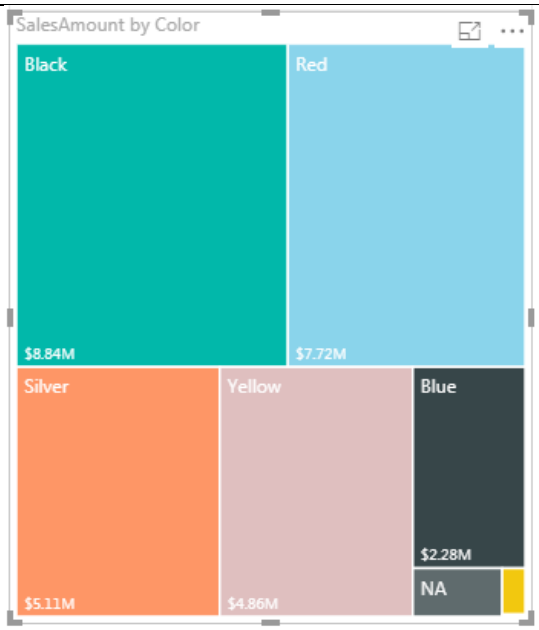
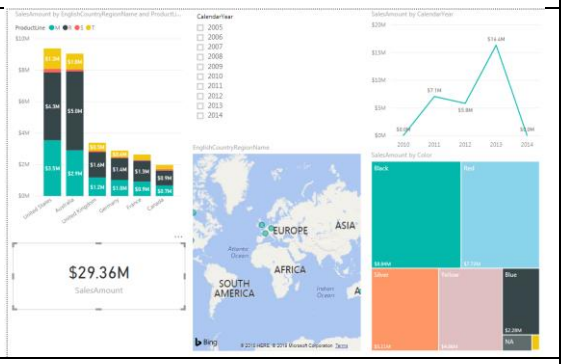
27. **Clear the filter** in **Map** and the **Slicer** (using **Clear selections**) to display all summarized data in different countries for all years.



28. Click any blank area on the canvas to create another visual. Select **Line chart** in the **VISUALIZATIONS** pane. Choose **CalendarYear** and **SalesAmount**. You can rearrange the x axis labels using **Sort by**. And **display** the **data labels**.





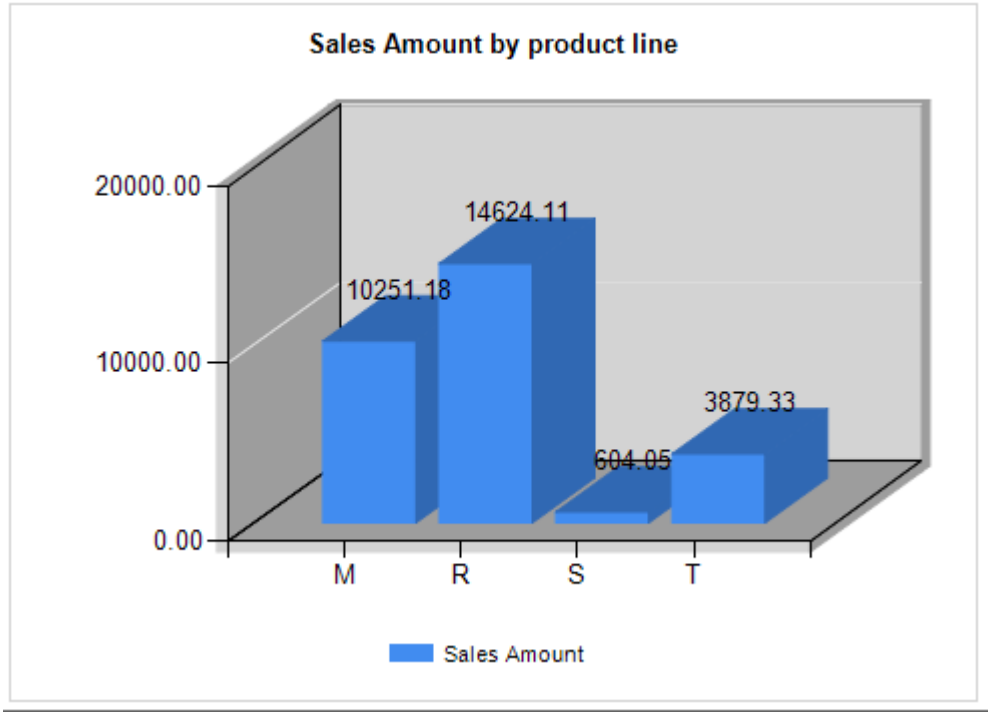
29. (Exercise) Create a **Treemap** to show the sales amount by product color, turn on the data labels. 30. Select **File → Save** to save the file named **lab4A-ans.pbix**	
31. The resulting visuals on Page 1 will be similar to this.	
32. If you have Microsoft account, you can publish your report using **Power BI service** and access it via http://app.powerbi.com	

V. Exercise

1) By using existing project **RSProject1** to add a new report by connecting to the cube (Adventure Works Cube) in ASProject1.Show total sales amount for each product line. Then create a 3-D column chart
   - ✓ Format the sales amount values in number with 2 d.p. in the report results
   - ✓ Insert appropriate chart title to the chart
   - ✓ Insert data labels to the chart (in number with 2 d.p. and show values as units of thousand)
   - ✓ Format the y-axis labels to number with unit of thousand
   - ✓ Place the legend at the bottom of the chart

Sales amount by product line

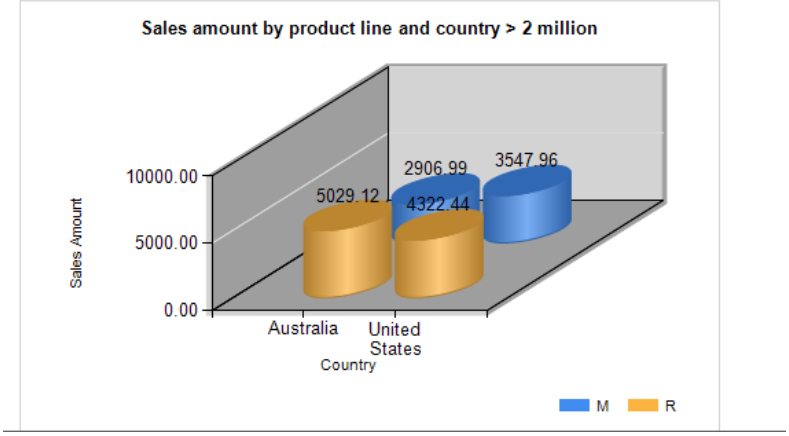
Product Line	Sales Amount
M	10251183.52
R	14624108.58
S	604053.30
T	3879331.82



2) By using existing project **RSProject1** to add a new report by connecting to the cube (Adventure Works Cube) in ASProject1. Show sales amount for each product line in each country, with sales amount greater than 2,000,000. (You may refer to Lab 3A). Create a 3-D cylinder chart like this. Save all the changes using **File ➔ Save All.**
   - Format the sales amount values in number with 2 d.p. in the report results
   - Insert appropriate chart title to the chart
   - Insert data labels to the chart (in number with 2 d.p. and show values as units of thousand)
   - Format the y-axis labels to number with unit of thousand
   - Place the legend at the bottom right of the chart

Sales amount by product line and country > 2 million

Product Line	English Country Region Name	Sales Amount
M	Australia	2906994.45
M	United States	3547956.78
R	Australia	5029120.41
R	United States	4322438.41



VI. Answer Submission

1. **Zip** your Report Services projects (**RSProject1 folder with RSProject1.sln**) that you created in C:\Users\demo\Documents\Visual Studio 2010\Projects. The default file name is **RSProject1.zip**

2. Submit the following files to the site **http://buelearning.hkbu.edu.hk/**
   - **lab4A-ans.pbix**
   - **RSProject1.zip**