# Index

| Sr No. | Practicals | Date | Sign |
|---|---|---|---|
| 1 | Import the legacy data from different sources such as (Excel, SqlServer, Oracle etc.) and load in the target system. | | [illegible] |
| 2 | Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Power BI. | | [illegible] |
| 3 | Data Visualization from ETL Process | | [illegible] |
| 4 | Creating a Cube in SQL server 2012 | | [illegible] |
| 5 | Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data. | | [illegible] |
| 6 | Implementation of Classification algorithm in R Programming. | | [illegible] |
| 7 | K-means clustering using R. | | [illegible] |
| 8 | Prediction Using Linear Regression. | | [illegible] |
| 9 | Data Analysis using Time Series Analysis. | | [illegible] |
| 10 | Data Modelling and Analytics with Pivot Table in Excel. | | [illegible] |
| 11 | Data Analysis and Visualization using Advanced Excel. | | [illegible] |

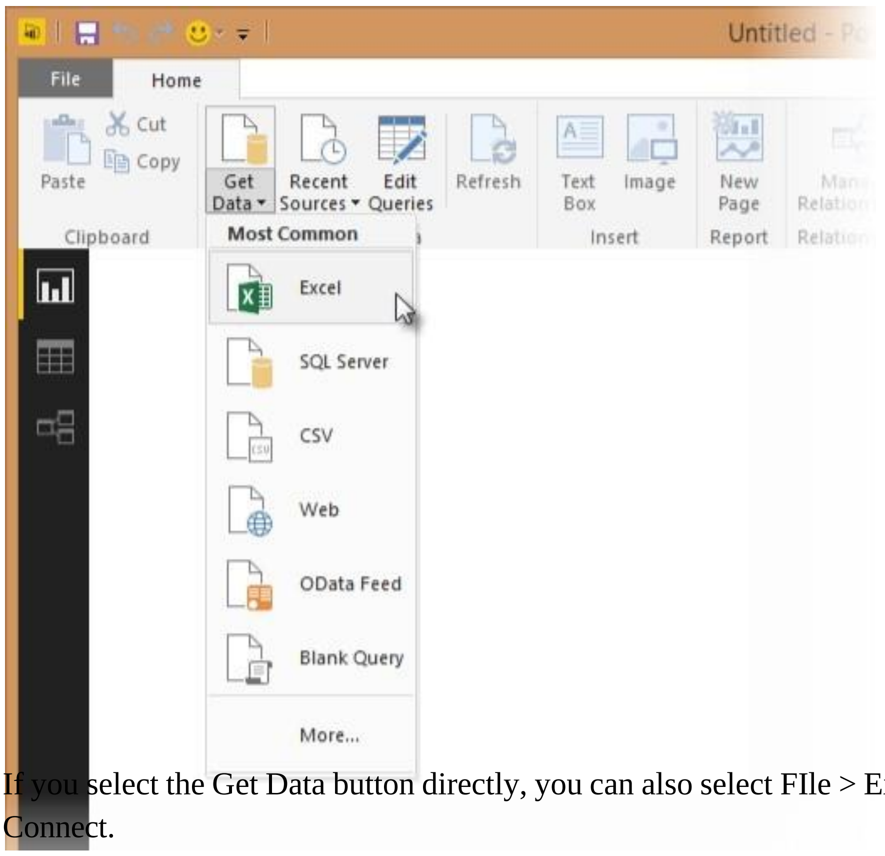
# Practical 1

**Import the legacy data from different sources such as (Excel, SqlServer, Oracle etc.) and load in the target system.**

**Importing Excel Data**

1) Launch Power BI Desktop.

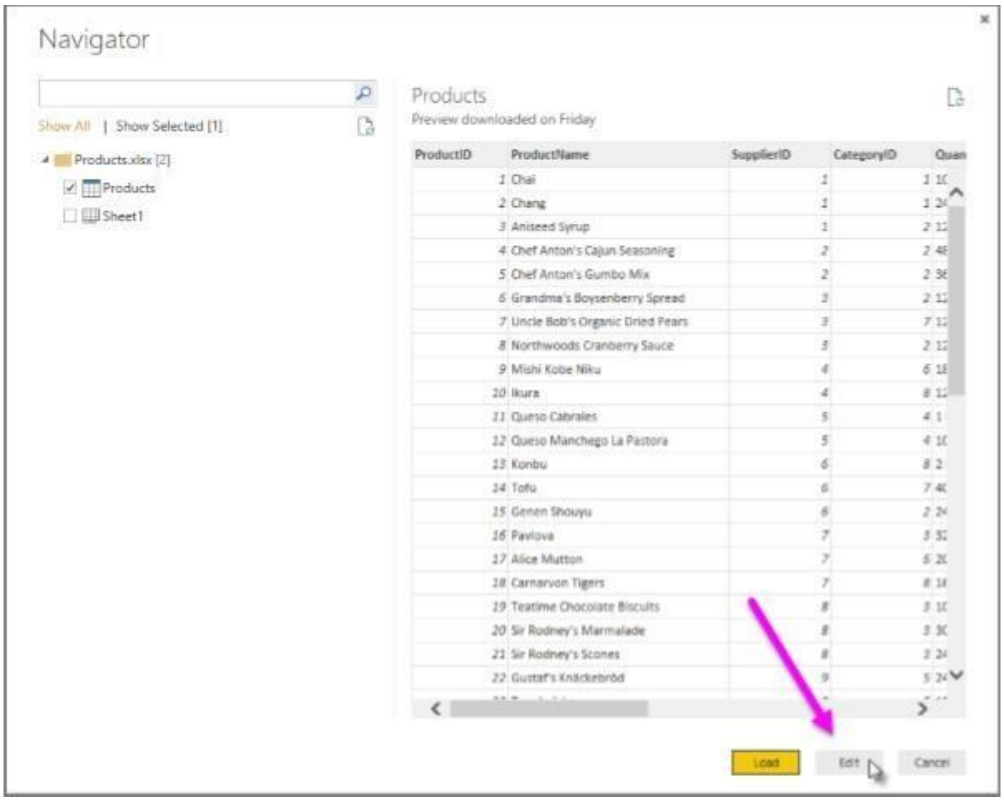
2) From the Home ribbon, select Get Data. Excel is one of the Most Common data connections, so you can select it directly from the Get Data menu.



3) If you select the Get Data button directly, you can also select FIle > Excel and select Connect.

4) In the Open File dialog box, select the Products.xlsx file.

5) In the Navigator pane, select the Products table and then select Edit.



Products
Preview downloaded on Friday

| ProductID | ProductName | SupplierID | CategoryID | Quan |
|---|---|---|---|---|
| 1 | Chai | 1 | 1 | 10 |
| 2 | Chang | 1 | 1 | 24 |
| 3 | Aniseed Syrup | 1 | 2 | 12 |
| 4 | Chef Anton's Cajun Seasoning | 2 | 2 | 48 |
| 5 | Chef Anton's Gumbo Mix | 2 | 2 | 36 |
| 6 | Grandma's Boysenberry Spread | 3 | 2 | 12 |
| 7 | Uncle Bob's Organic Dried Pears | 3 | 7 | 12 |
| 8 | Northwoods Cranberry Sauce | 3 | 2 | 12 |
| 9 | Mishi Kobe Niku | 4 | 6 | 18 |
| 10 | Ikura | 4 | 8 | 12 |
| 11 | Queso Cabrales | 5 | 4 | 1 |
| 12 | Queso Manchego La Pastora | 5 | 4 | 10 |
| 13 | Konbu | 6 | 8 | 2 |
| 14 | Tofu | 6 | 7 | 40 |
| 15 | Genen Shouyu | 6 | 2 | 24 |
| 16 | Pavlova | 7 | 3 | 32 |
| 17 | Alice Mutton | 7 | 6 | 20 |
| 18 | Carnarvon Tigers | 7 | 8 | 16 |
| 19 | Teatime Chocolate Biscuits | 8 | 3 | 10 |
| 20 | Sir Rodney's Marmalade | 8 | 3 | 30 |
| 21 | Sir Rodney's Scones | 8 | 3 | 24 |
| 22 | Gustaf's Knäckebröd | 9 | 5 | 24 |

**Importing Data from OData Feed**

In this task, you'll bring in order data. This step represents connecting to a sales system. You import data into Power BI Desktop from the sample Northwind OData feed at the following URL, which you can copy (and then paste) in the steps below: http://services.odata.org/V3/Northwind/Northwind.svc/

Connect to an OData feed:

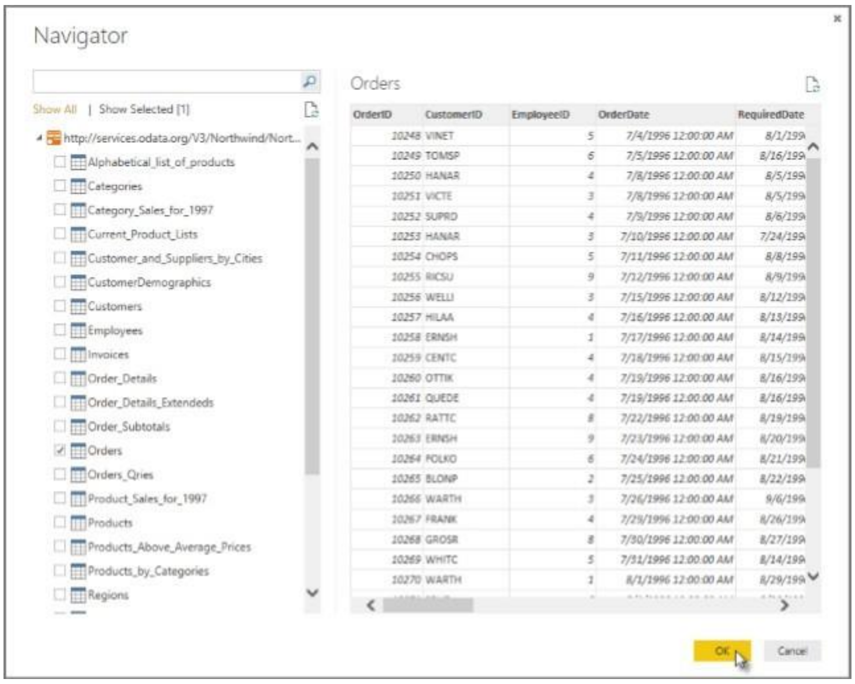
1) From the Home ribbon tab in Query Editor, select Get Data.

2) Browse to the OData Feed data source.

3) In the OData Feed dialog box, paste the URL for the Northwind OData feed.

4) Select OK.

5) In the Navigator pane, select the Orders table, and then select Edit.



Navigator

Show All | Show Selected [1]

http://services.odata.org/V3/Northwind/Nort...

- [ ] Alphabetical_list_of_products
- [ ] Categories
- [ ] Category_Sales_for_1997
- [ ] Current_Product_Lists
- [ ] Customer_and_Suppliers_by_Cities
- [ ] CustomerDemographics
- [ ] Customers
- [ ] Employees
- [ ] Invoices
- [ ] Order_Details
- [ ] Order_Details_Extendeds
- [ ] Order_Subtotals
- [x] Orders
- [ ] Orders_Qries
- [ ] Product_Sales_for_1997
- [ ] Products
- [ ] Products_Above_Average_Prices
- [ ] Products_by_Categories
- [ ] Regions

Orders

| OrderID | CustomerID | EmployeeID | OrderDate | RequiredDate |
|---|---|---|---|---|
| 10248 | VINET | 5 | 7/4/1996 12:00:00 AM | 8/1/199 |
| 10249 | TOMSP | 6 | 7/5/1996 12:00:00 AM | 8/16/199 |
| 10250 | HANAR | 4 | 7/8/1996 12:00:00 AM | 8/5/199 |
| 10251 | VICTE | 3 | 7/8/1996 12:00:00 AM | 8/5/199 |
| 10252 | SUPRD | 4 | 7/9/1996 12:00:00 AM | 8/6/199 |
| 10253 | HANAR | 3 | 7/10/1996 12:00:00 AM | 7/24/199 |
| 10254 | CHOPS | 5 | 7/11/1996 12:00:00 AM | 8/8/199 |
| 10255 | RICSU | 9 | 7/12/1996 12:00:00 AM | 8/9/199 |
| 10256 | WELLI | 3 | 7/15/1996 12:00:00 AM | 8/12/199 |
| 10257 | HILAA | 4 | 7/16/1996 12:00:00 AM | 8/13/199 |
| 10258 | ERNSH | 1 | 7/17/1996 12:00:00 AM | 8/14/199 |
| 10259 | CENTC | 4 | 7/18/1996 12:00:00 AM | 8/15/199 |
| 10260 | OTTIK | 4 | 7/19/1996 12:00:00 AM | 8/16/199 |
| 10261 | QUEDE | 4 | 7/19/1996 12:00:00 AM | 8/16/199 |
| 10262 | RATTC | 8 | 7/22/1996 12:00:00 AM | 8/19/199 |
| 10263 | ERNSH | 9 | 7/23/1996 12:00:00 AM | 8/20/199 |
| 10264 | FOLKO | 6 | 7/24/1996 12:00:00 AM | 8/21/199 |
| 10265 | BLONP | 2 | 7/25/1996 12:00:00 AM | 8/22/199 |
| 10266 | WARTH | 3 | 7/26/1996 12:00:00 AM | 9/6/199 |
| 10267 | FRANK | 4 | 7/29/1996 12:00:00 AM | 8/26/199 |
| 10268 | GROSR | 8 | 7/30/1996 12:00:00 AM | 8/27/199 |
| 10269 | WHITC | 5 | 7/31/1996 12:00:00 AM | 8/14/199 |
| 10270 | WARTH | 1 | 8/1/1996 12:00:00 AM | 8/29/199 |

**Note -** You can click a table name, without selecting the checkbox, to see a preview.

# Practical 2

## Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Power BI.

**Step 1 : Data Extraction :**

The data extraction is first step of ETL. There are 2 Types of Data Extraction

1. Full Extraction : All the data from source systems or operational systems gets extracted to staging area. (Initial Load)

2. Partial Extraction : Sometimes we get notification from the source system to update specific date. It is called as Delta load.

Source System Performance: The Extraction strategies should not affect source system performance.

**Step 2 : Data Transformation :**

The data transformation is second step.After extracting the data there is big need to do the transformation as per the target system.I would like to give you some bullet points of Data Transformation.

- Data Extracted from source system is in to Raw format. We need to transform it before loading in to target server.
- Data has to be cleaned, mapped and transformed.
- There are following important steps of Data Transformation :

1. **Selection :** Select data to load in target

2. **Matching :** Match the data with target system

3. **Data Transforming :** We need to change data as per target table structures

**Real life examples of Data Transformation :**

- Standardizing data : Data is fetched from multiple sources so it needs to be standardized as per the target system.
- Character set conversion : Need to transform the character sets as per the target systems. (Firstname and last name example)
- Calculated and derived values: In source system there is first val and second val and in target we need the calculation of first val and second val.
- Data Conversion in different formats : If in source system date in in DDMMYY format and in target the date is in DDMONYYYY format then this transformation needs to be

done at transformation phase.

## Step 3 : Data Loading

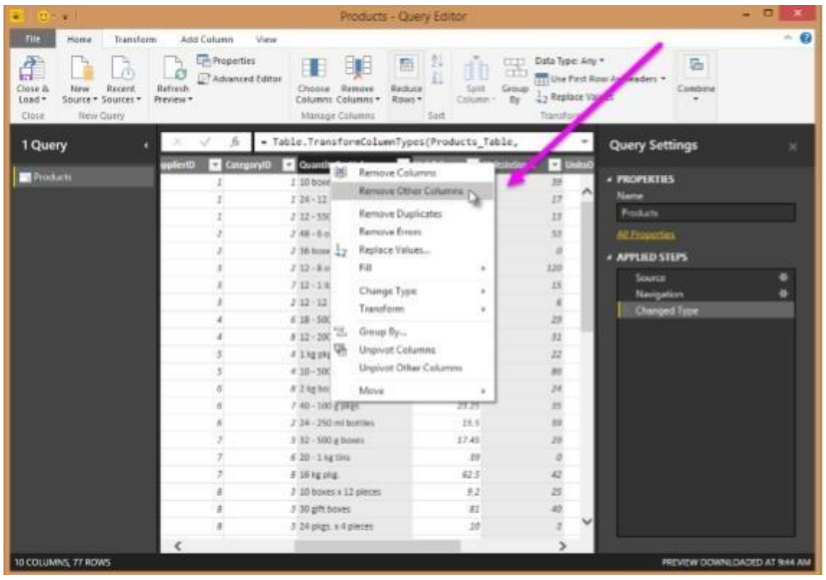
## ETL Process in Power BI

### 1) Remove other columns to only display columns of interest

In this step you remove all columns except **ProductID**, **ProductName**, **UnitsInStock**, and **QuantityPerUnit**

Power BI Desktop includes Query Editor, which is where you shape and transform your data connections. Query Editor opens automatically when you select **Edit** from Navigator. You can also open the Query Editor by selecting Edit Queries from the Home ribbon in Power BI Desktop. The following steps are performed in Query Editor.

1. In **Query Editor**, select the **ProductID, ProductName**, **QuantityPerUnit,** and **UnitsInStock** columns (use **Ctrl+Click** to select more than one column, or **Shift+Click** to select columns that are beside each other).
2. Select **Remove Columns > Remove** Other Columns from the ribbon, or right-click on a column header and click Remove Other Columns.



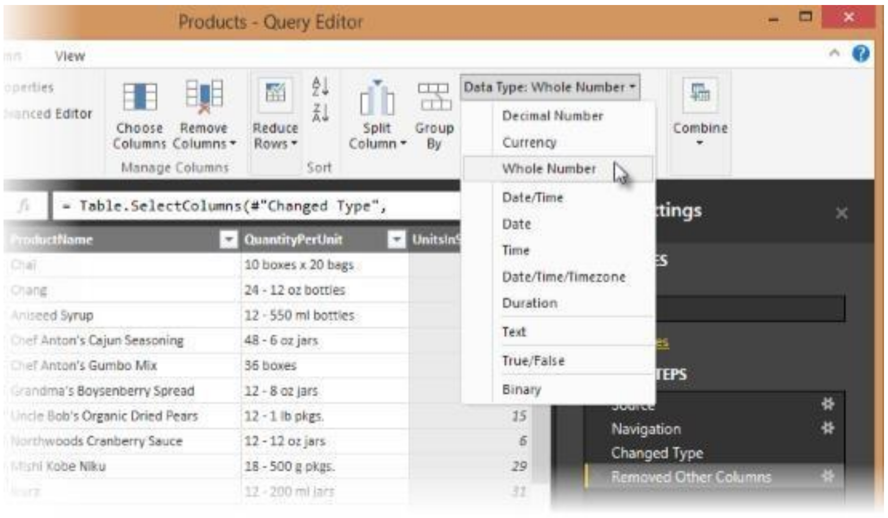
### 3. Change the data type of the UnitsInStock column

When Query Editor connects to data, it reviews each field and to determine the best data type. For the Excel workbook, products in stock will always be a whole number, so in this step you confirm the **UnitsInStock** column's datatype is Whole Number.

1. Select the **UnitsInStock** column.

2. Select the **Data Type drop-down button** in the **Home ribbon**.

3. If not already a Whole Number, select **Whole Number** for data type from the drop down (the Data Type: button also displays the data type for the current selection).



```
= Table.SelectColumns(#"Changed Type",
```

| ProductName | QuantityPerUnit | UnitsIn |
|---|---|---|
| Chai | 10 boxes x 20 bags | |
| Chang | 24 - 12 oz bottles | |
| Aniseed Syrup | 12 - 550 ml bottles | |
| Chef Anton's Cajun Seasoning | 48 - 6 oz jars | |
| Chef Anton's Gumbo Mix | 36 boxes | |
| Grandma's Boysenberry Spread | 12 - 8 oz jars | |
| Uncle Bob's Organic Dried Pears | 12 - 1 lb pkgs. | 15 |
| Northwoods Cranberry Sauce | 12 - 12 oz jars | 6 |
| Mishi Kobe Niku | 18 - 500 g pkgs. | 29 |
| Ikura | 12 - 200 ml jars | 31 |

3. **Expand the Order_Details table**

The Orders table contains a reference to a Details table, which contains the individual products that were included in each Order. When you connect to data sources with multiples tables (such as a relational database) you can use these references to build up your query

In this step, you expand the **Order_Details** table that is related to the Orders table, to combine the **ProductID**, **UnitPrice**, and **Quantity** columns from **Order_Details** into the **Orders table**. This is a representation of the data in these tables:

The Expand operation combines columns from a related table into a subject table. When the query runs, rows from the related table (**Order_Details**) are combined into rows from the subject table (**Orders**).

After you expand the Order_Details table, three new columns and additional rows are added to the Orders table, one for each row in the nested or related table.

1. In the Query View, scroll to the Order_Details column.

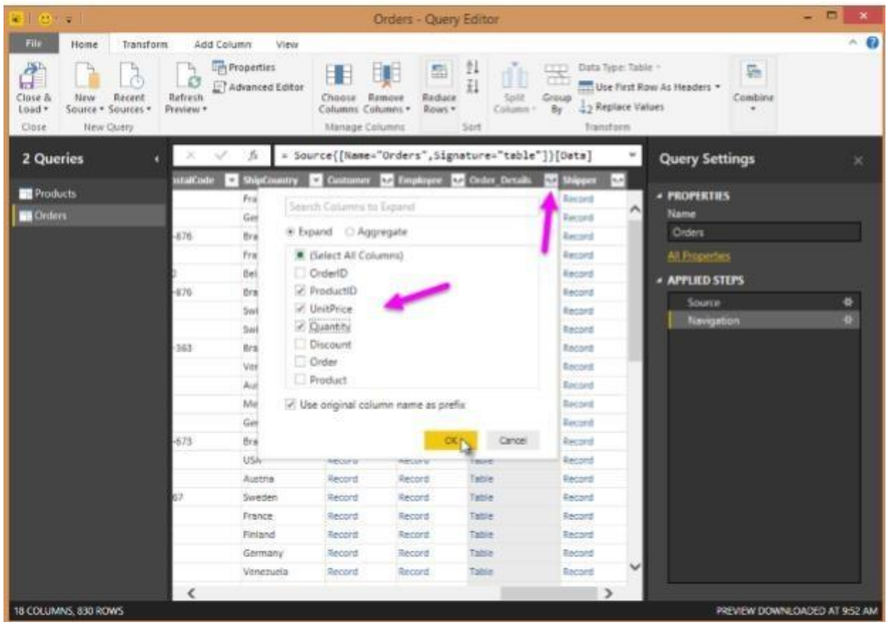
2. In the Order_Details column, select the expand icon ( ).

3. In the Expand drop-down:

   a. Select (Select All Columns) to clear all columns.

   b. Select ProductID, UnitPrice, and Quantity.

   c. Click OK.

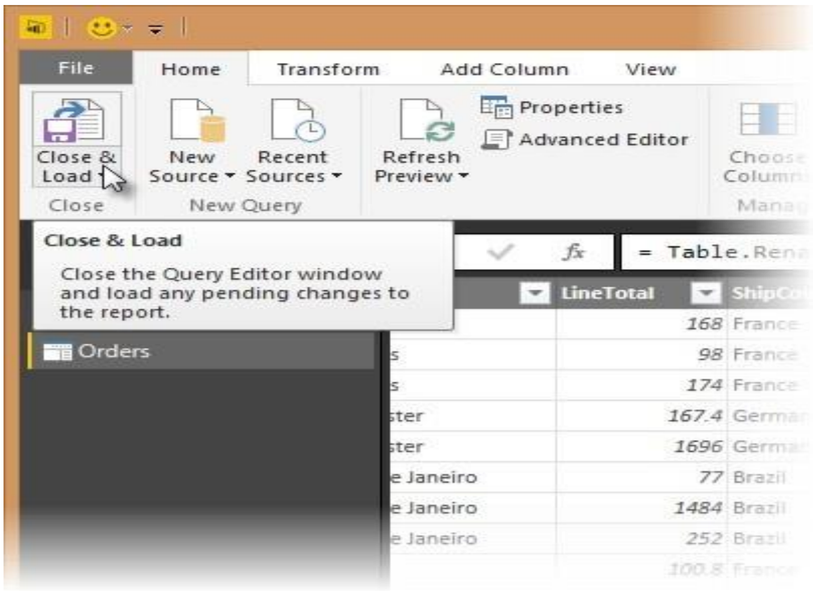


## 4. Calculate the line total for each Order_Details row

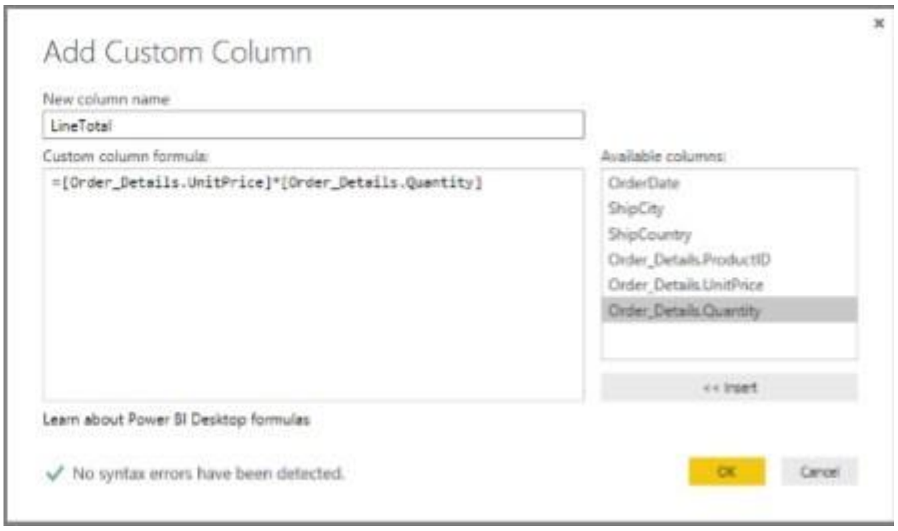
Power BI Desktop lets you to create calculations based on the columns you are importing, so you can enrich the data that you connect to. In this step, you create a Custom Column to calculate the line total for each Order_Details row.

Calculate the line total for each Order_Details row:

1. In the Add Column ribbon tab, click Add Custom Column.



2. In the Add Custom Column dialog box, in the Custom Column Formula textbox, enter [Order_Details.UnitPrice] * [Order_Details.Quantity].
3. In the New column name textbox, enter LineTotal.
4. Click OK.



Add Custom Column

New column name

LineTotal

Custom column formula:

```
=[Order_Details.UnitPrice]*[Order_Details.Quantity]
```

Available columns:

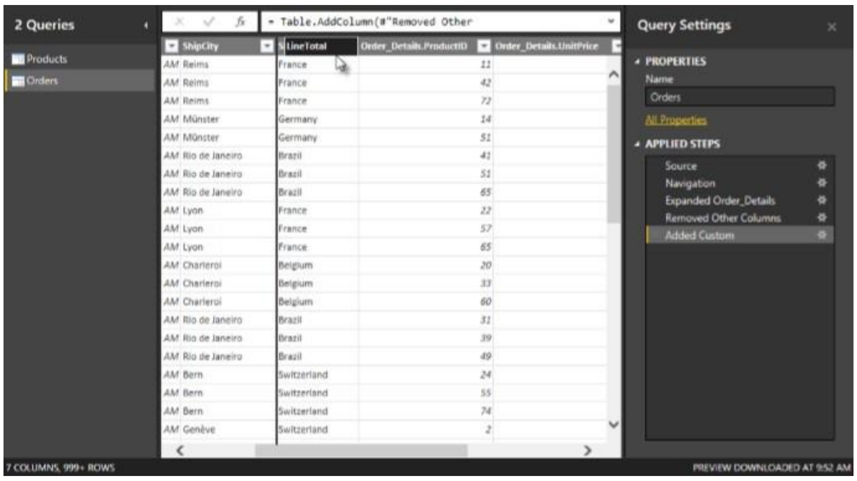
- OrderDate
- ShipCity
- ShipCountry
- Order_Details.ProductID
- Order_Details.UnitPrice
- Order_Details.Quantity

✓ No syntax errors have been detected.

**5. Rename and reorder columns in the query**

In this step you finish making the model easy to work with when creating reports, by renaming the final columns and changing their order.

1. In Query Editor, drag the LineTotal column to the left, after ShipCountry.



```
= Table.AddColumn(#"Removed Other
```

| ShipCity | LineTotal | Order_Details.ProductID | Order_Details.UnitPrice |
|---|---|---|---|
| AM Reims | France | 11 | |
| AM Reims | France | 42 | |
| AM Reims | France | 72 | |
| AM Münster | Germany | 14 | |
| AM Münster | Germany | 51 | |
| AM Rio de Janeiro | Brazil | 41 | |
| AM Rio de Janeiro | Brazil | 51 | |
| AM Rio de Janeiro | Brazil | 65 | |
| AM Lyon | France | 22 | |
| AM Lyon | France | 57 | |
| AM Lyon | France | 65 | |
| AM Charleroi | Belgium | 20 | |
| AM Charleroi | Belgium | 33 | |
| AM Charleroi | Belgium | 60 | |
| AM Rio de Janeiro | Brazil | 31 | |
| AM Rio de Janeiro | Brazil | 39 | |
| AM Rio de Janeiro | Brazil | 49 | |
| AM Bern | Switzerland | 24 | |
| AM Bern | Switzerland | 55 | |
| AM Bern | Switzerland | 74 | |
| AM Genève | Switzerland | 2 | |

Name: Orders

Applied steps:
- Source
- Navigation
- Expanded Order_Details
- Removed Other Columns
- Added Custom

7 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 9:52 AM

2. Remove the Order_Details. prefix from the Order_Details.ProductID, Order_Details.UnitPrice and Order_Details.Quantity columns, by double-clicking on each column header, and then deleting that text from the column name.

6. **Combine the Products and Total Sales queries**

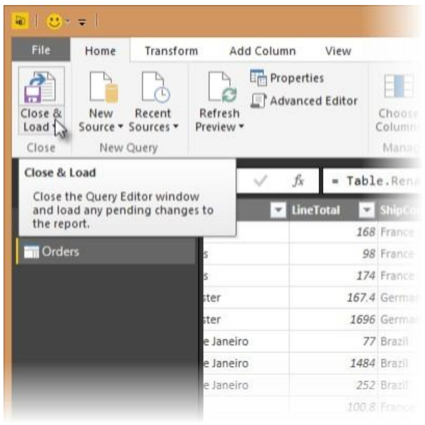
Power BI Desktop does not require you to combine queries to report on them. Instead, you can create Relationships between datasets. These relationships can be created on any column that is common to your datasets

we have Orders and Products data that share a common 'ProductID' field, so we need to ensure there's a relationship between them in the model we're using with Power BI Desktop. Simply specify in Power BI Desktop that the columns from each table are related (i.e. columns that have the same values). Power BI Desktop works out the direction and cardinality of the relationship for you. In some cases, it will even detect the relationships automatically.

In this task, you confirm that a relationship is established in Power BI Desktop between the Products and Total Sales queries

Step 1: Confirm the relationship between Products and Total Sales

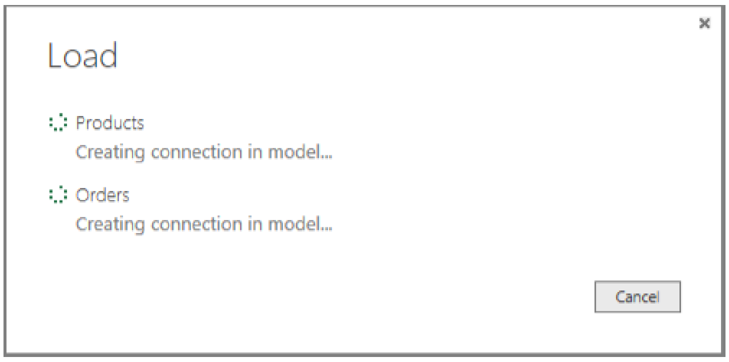
1. First, we need to load the model that we created in Query Editor into Power BI Desktop. From the Home ribbon of Query Editor, select Close & Load.



Close & Load

Close the Query Editor window and load any pending changes to the report.

2. Power BI Desktop loads the data from the two queries.



Load

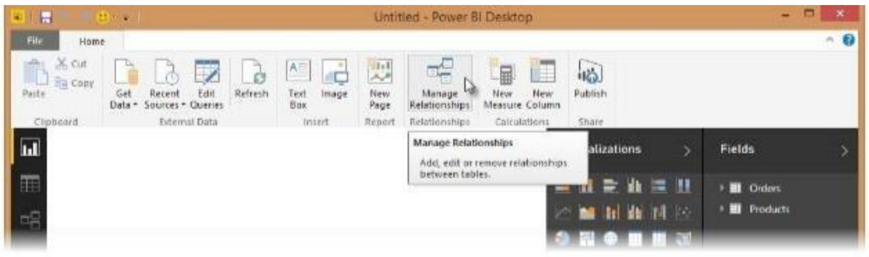
Products

Creating connection in model...

Orders

Creating connection in model...

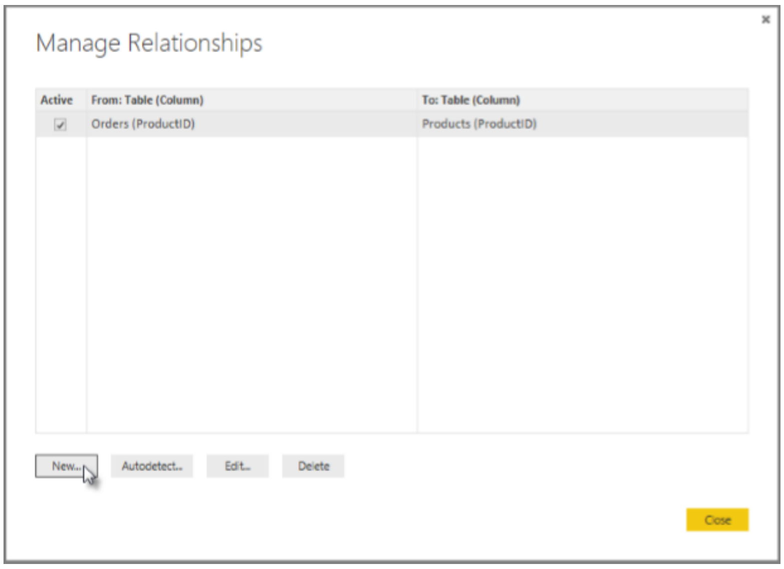
3. Once the data is loaded, select the Manage Relationships button Home ribbon.



Manage Relationships

Add, edit or remove relationships between tables.

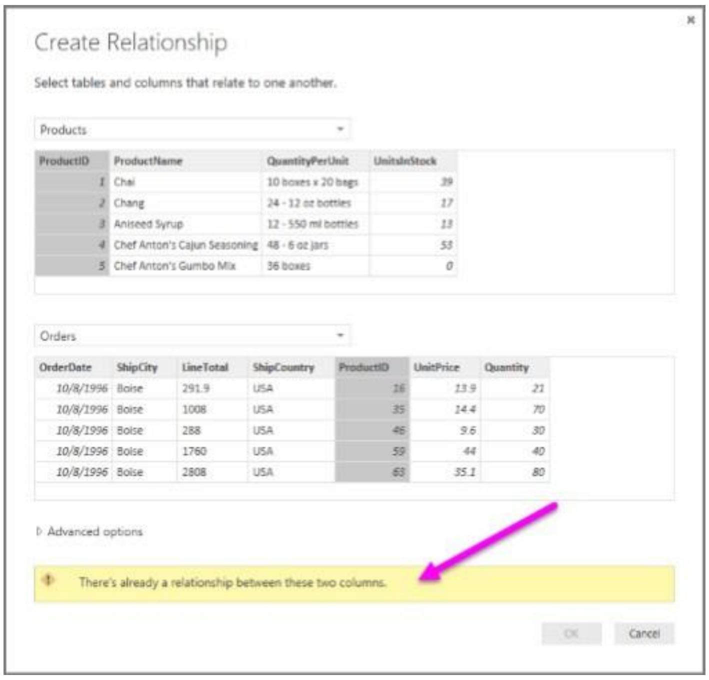
4. Select the New… button



Manage Relationships

| Active | From: Table (Column) | To: Table (Column) |
|---|---|---|
| ✓ | Orders (ProductID) | Products (ProductID) |

5. When we attempt to create the relationship, we see that one already exists! As shown in the Create Relationship dialog (by the shaded columns), the ProductsID fields in each query already have an established relationship.



Create Relationship

Select tables and columns that relate to one another.

Products

| ProductID | ProductName | QuantityPerUnit | UnitsInStock |
|---|---|---|---|
| 1 | Chai | 10 boxes x 20 bags | 39 |
| 2 | Chang | 24 - 12 oz bottles | 17 |
| 3 | Aniseed Syrup | 12 - 550 ml bottles | 13 |
| 4 | Chef Anton's Cajun Seasoning | 48 - 6 oz jars | 53 |
| 5 | Chef Anton's Gumbo Mix | 36 boxes | 0 |

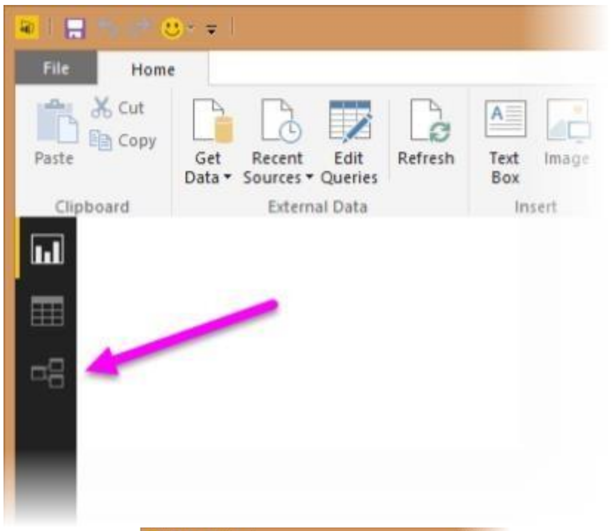
Orders

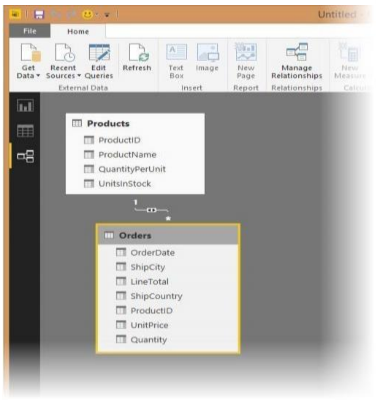
| OrderDate | ShipCity | LineTotal | ShipCountry | ProductID | UnitPrice | Quantity |
|---|---|---|---|---|---|---|
| 10/8/1996 | Boise | 291.9 | USA | 16 | 13.9 | 21 |
| 10/8/1996 | Boise | 1008 | USA | 35 | 14.4 | 70 |
| 10/8/1996 | Boise | 288 | USA | 46 | 9.6 | 30 |
| 10/8/1996 | Boise | 1760 | USA | 59 | 44 | 40 |
| 10/8/1996 | Boise | 2808 | USA | 63 | 35.1 | 80 |

There's already a relationship between these two columns.

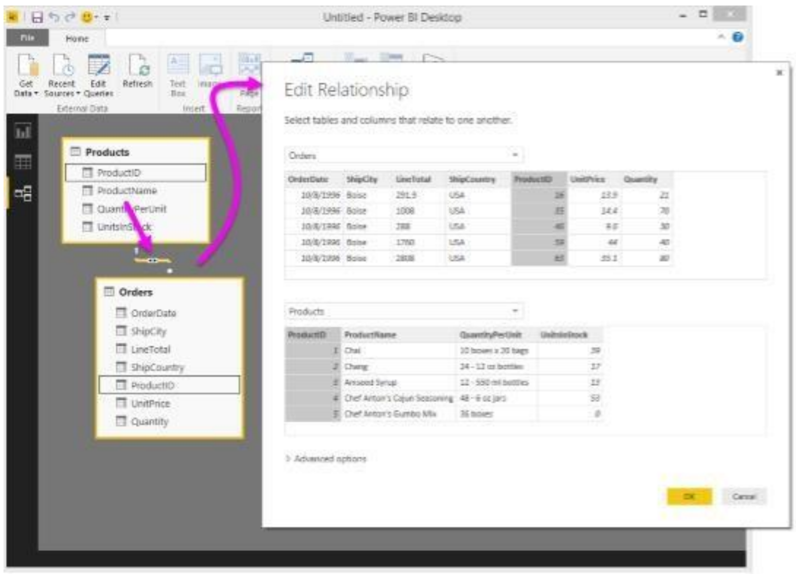
5. Select Cancel, and then select Relationship view in Power BI Desktop.

6. We see the following, which visualizes the relationship between the queries.





7. When you double-click the arrow on the line that connects the to queries, an Edit Relationship dialog appears.



Edit Relationship

Select tables and columns that relate to one another.

Orders

| OrderDate | ShipCity | LineTotal | ShipCountry | ProductID | UnitPrice | Quantity |
|---|---|---|---|---|---|---|
| 10/8/1996 | Boise | 291.9 | USA | 16 | 13.9 | 21 |
| 10/8/1996 | Boise | 1008 | USA | 35 | 14.4 | 70 |
| 10/8/1996 | Boise | 288 | USA | 46 | 9.6 | 30 |
| 10/8/1996 | Boise | 1760 | USA | 59 | 44 | 40 |
| 10/8/1996 | Boise | 2808 | USA | 65 | 35.1 | 80 |

Products

| ProductID | ProductName | QuantityPerUnit | UnitsInStock |
|---|---|---|---|
| 1 | Chai | 10 boxes x 20 bags | 39 |
| 2 | Chang | 24 - 12 oz bottles | 17 |
| 3 | Aniseed Syrup | 12 - 550 ml bottles | 13 |
| 4 | Chef Anton's Cajun Seasoning | 48 - 6 oz jars | 53 |
| 5 | Chef Anton's Gumbo Mix | 36 boxes | 0 |

Advanced options

OK Cancel

8. No need to make any changes, so we'll just select Cancel to close the Edit Relationship dialog.

# Practical 3

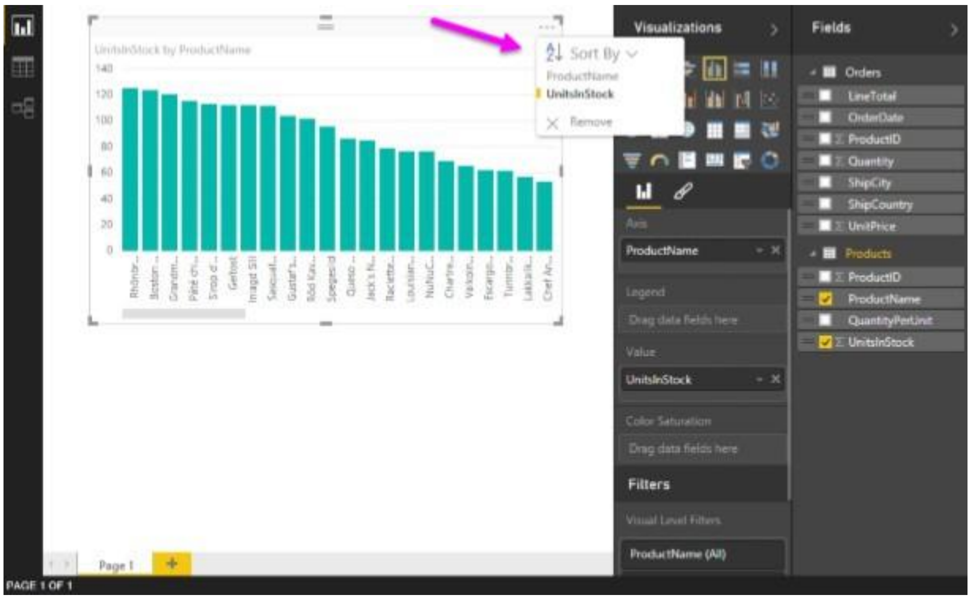
## Data Visualization from ETL Process

Power BI Desktop lets you create a variety of visualizations to gain insights from your data. You can build reports with multiple pages and each page can have multiple visuals. You can interact with your visualizations to help analyze and understand your data

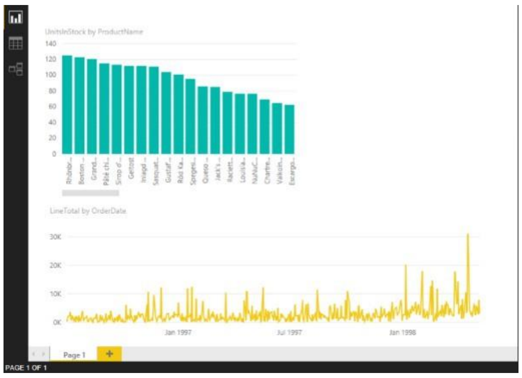
In this task, you create a report based on the data previously loaded. You use the Fields pane to select the columns from which you create the visualizations.

**Step 1: Create charts showing Units in Stock by Product and Total Sales by Year**

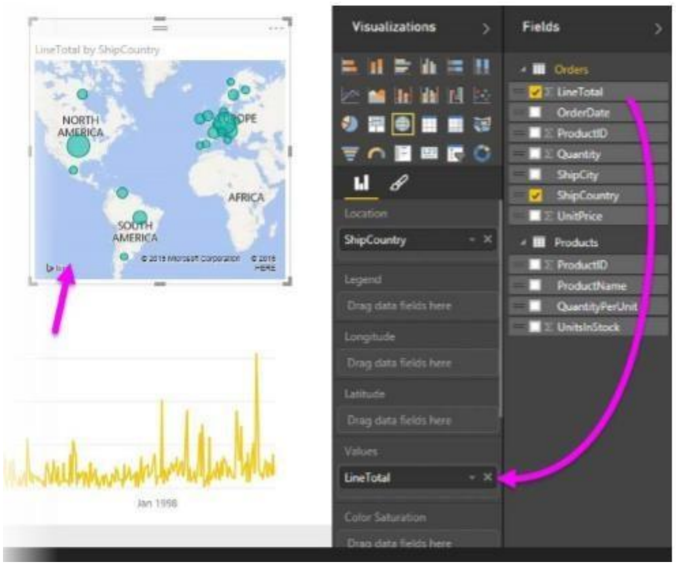
1. Drag UnitsInStock from the Field pane (the Fields pane is along the right of the screen) onto a blank space on the canvas. A Table visualization is created. Next, drag ProductName to the Axis box, found in the bottom half of the Visualizations pane. Then we then select Sort By > UnitsInStock using the skittles in the top right corer of the visualization.



2. Drag OrderDate to the canvas beneath the first chart, then drag LineTotal (again, from the Fields pane) onto the visual, then select Line Chart. The following visualization is created.



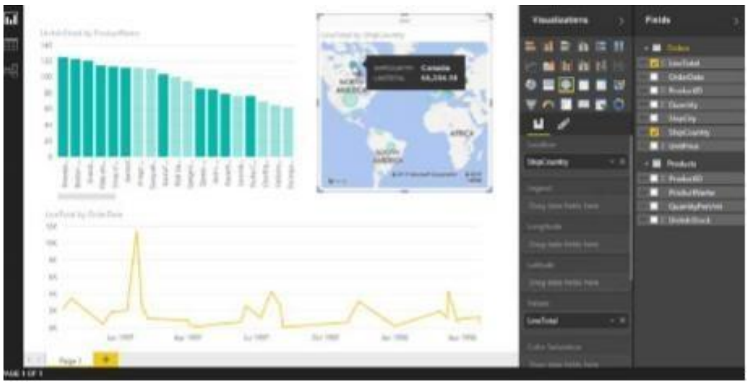
3. Next, drag ShipCountry to a space on the canvas in the top right. Because you selected a geographic field, a map was created automatically. Now drag LineTotal to the Values field; the circles on the map for each country are now relative in size to the LineTotal for orders shipped to that country.



## Step 2: Interact with your report visuals to analyze further

Power BI Desktop lets you interact with visuals that cross-highlight and filter each other to uncover further trends.

1. Click on the light blue circle centered in Canada. Note how the other visuals are filtered to show Stock (ShipCountry) and Total Orders (LineTotal) just for Canada.



# Practical 4

## Creating a Cube in SQL server 2012

### Creating Data Warehouse

Let us execute our T-SQL Script to create data warehouse with fact tables, dimensions and populate them with appropriate test values.

Download T-SQL script attached with this article for creation of Sales Data Warehouse or download from this article "**Create First Data Warehouse**" and run it in your SQL Server.

Follow the given steps to run the query in **SSMS** (SQL Server Management Studio).

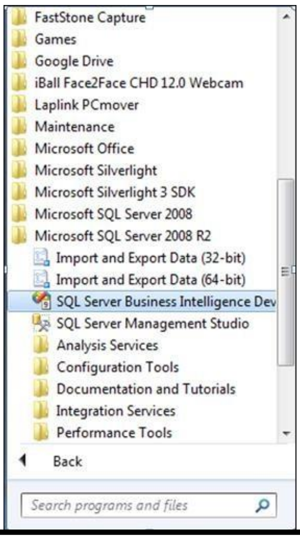
1. Open SQL Server Management Studio 2008
2. Connect Database Engine
3. Open **New Query** editor
4. Copy paste Scripts given below in various steps in new query editor window one by one
5. To run the given SQL Script, press **F5**
6. It will create and populate "Sales_DW" database on your SQL Server

### Developing an OLAP Cube

For creation of OLAP Cube in Microsoft BIDS Environment, follow the 10 easy steps given below.

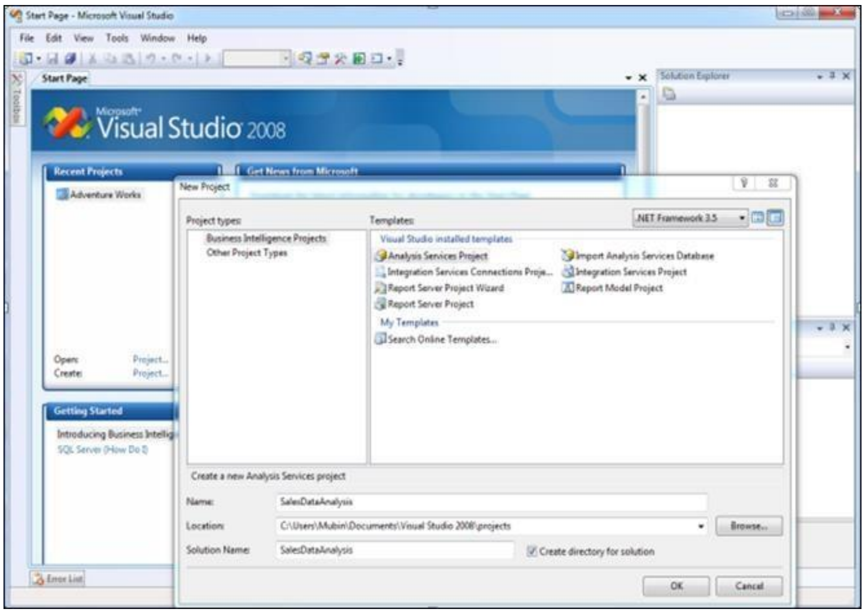
*Step 1: Start BIDS Environment*

Click on **Start Menu** -> Microsoft **SQL Server 2008 R2** -> Click **SQL Server Business Intelligence Development Studio.**


FastStone Capture
Games
Google Drive
iBall Face2Face CHD 12.0 Webcam
Laplink PCmover
Maintenance
Microsoft Office
Microsoft Silverlight
Microsoft Silverlight 3 SDK
Microsoft SQL Server 2008
Microsoft SQL Server 2008 R2
Import and Export Data (32-bit)
Import and Export Data (64-bit)
SQL Server Business Intelligence Dev
SQL Server Management Studio
Analysis Services
Configuration Tools
Documentation and Tutorials
Integration Services
Performance Tools
Back
Search programs and files

*Step 2: Start Analysis Services Project*

Click **File** -> **New** -> **Project** ->**Business Intelligence Projects** ->select **Analysis Services Project**-> Assign Project **Name** -> Click **OK**


Start Page - Microsoft Visual Studio
File Edit View Tools Window Help
Start Page
Solution Explorer
Microsoft Visual Studio 2008
Recent Projects
Adventure Works
Get News from Microsoft
Getting Started
Introducing Business Intellig
SQL Server (How Do I)
Error List
New Project
Project types:
Business Intelligence Projects
Other Project Types
Templates:
.NET Framework 3.5
Visual Studio installed templates
Analysis Services Project
Import Analysis Services Database
Integration Services Connections Proje...
Integration Services Project
Report Server Project Wizard
Report Model Project
Report Server Project
My Templates
Search Online Templates...
Browse...
OK
Cancel

Create a new Analysis Services project

Name: SalesDataAnalysis

Location: C:\Users\Mubin\Documents\Visual Studio 2008\projects

Solution Name: SalesDataAnalysis

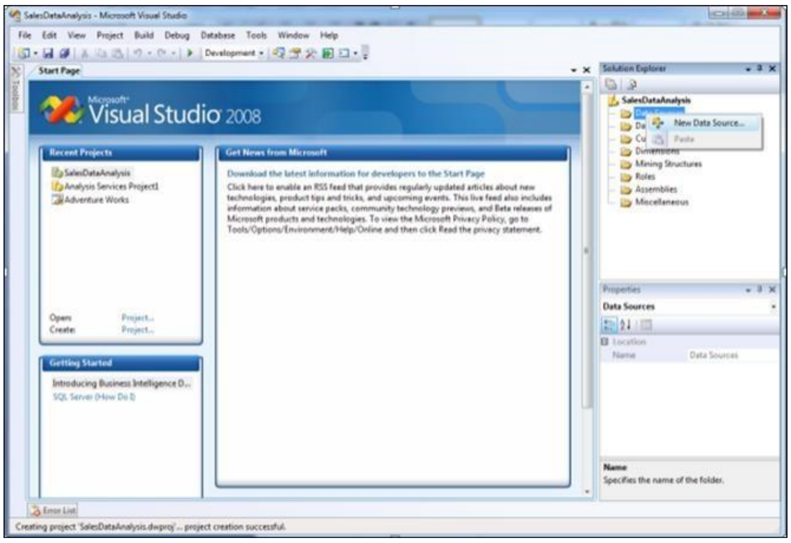
☑ Create directory for solution

*Step 3: Creating New Data Source*

3.1. In Solution Explorer, Right click on **Data Source** -> Click **New Data Source**

*Step 3: Creating New Data Source*

1. In Solution Explorer, Right click on **Data Source** -> Click **New Data Source**



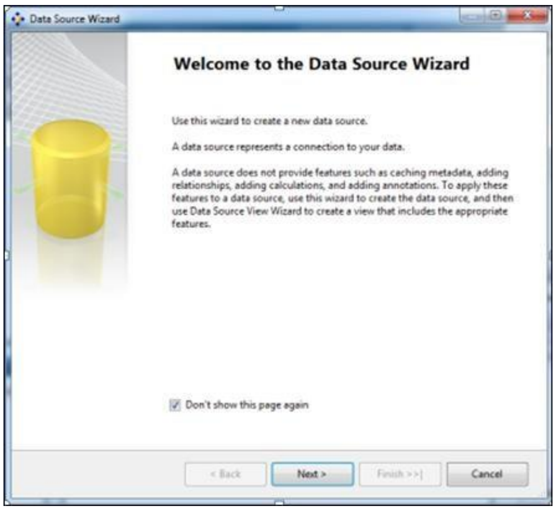
Download the latest information for developers to the Start Page

Click here to enable an RSS feed that provides regularly updated articles about new technologies, product tips and tricks, and upcoming events. This live feed also includes information about service packs, community technology previews, and Beta releases of Microsoft products and technologies. To view the Microsoft Privacy Policy, go to Tools/Options/Environment/Help/Online and then click Read the privacy statement.

Name
Specifies the name of the folder.

Creating project 'SalesDataAnalysis.dwproj'... project creation successful.

2. Click on **Next**



Data Source Wizard

Welcome to the Data Source Wizard

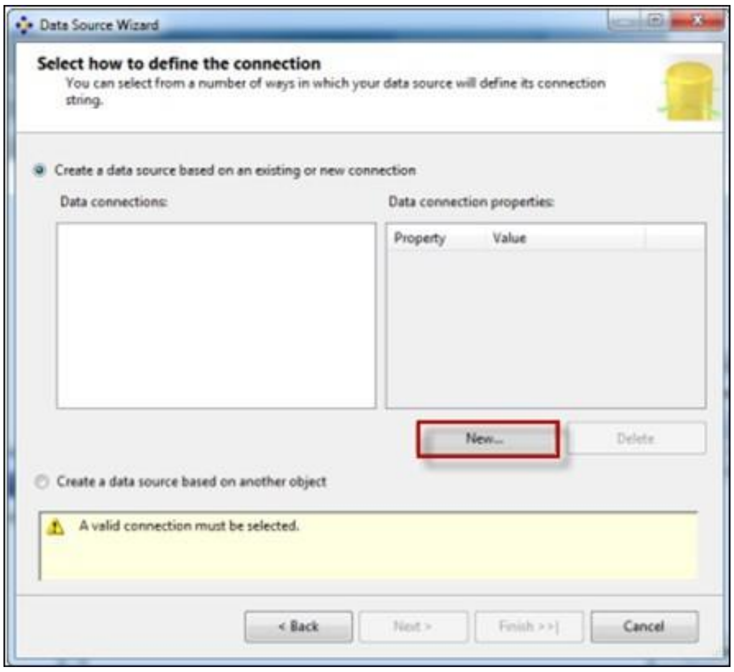
Use this wizard to create a new data source.

A data source represents a connection to your data.

A data source does not provide features such as caching metadata, adding relationships, adding calculations, and adding annotations. To apply these features to a data source, use this wizard to create the data source, and then use Data Source View Wizard to create a view that includes the appropriate features.

[✓] Don't show this page again

3. Click on New **Button**



Data Source Wizard

Select how to define the connection

You can select from a number of ways in which your data source will define its connection string.

● Create a data source based on an existing or new connection

Data connections:

Data connection properties:

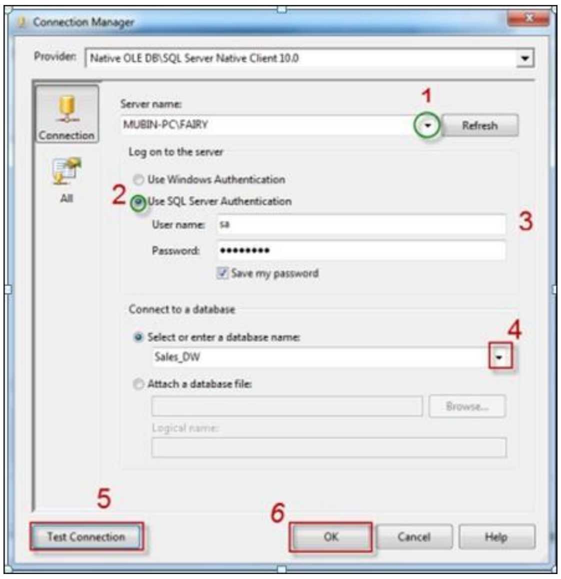
| Property | Value |
|---|---|

○ Create a data source based on another object

A valid connection must be selected.

4. Creating **New connection**

1. Specify Your SQL **Server Name** where your Data Warehouse was created
2. Select Radio Button according to your **SQL Server Authentication** mode
3. Specify your **Credentials** using which you can connect to your SQL Server
4. Select database Sales_DW.
5. Click on **Test Connection** and verify for its success
6. Click **OK**.



Provider: Native OLE DB\SQL Server Native Client 10.0

Server name: MUBIN-PC\FAIRY

Log on to the server

Use Windows Authentication [✓ Use SQL Server Authentication]

User name: sa

Password: ••••••••

[✓ Save my password]

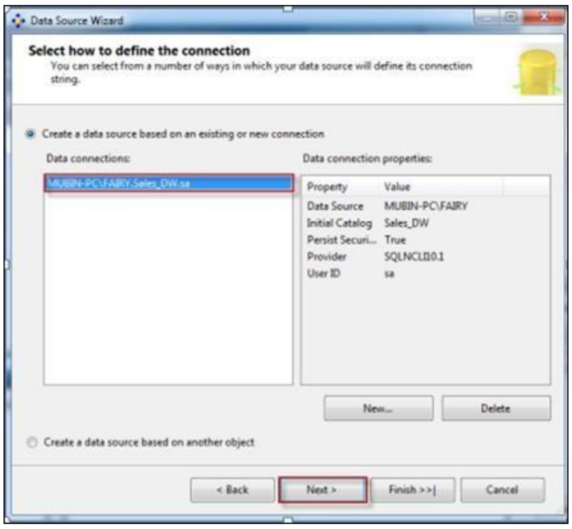
Connect to a database

[✓ Select or enter a database name:] Sales_DW

Attach a database file:

Logical name:

5. Select Connection created in **Data Connections**-> Click **Next**



**Select how to define the connection**

You can select from a number of ways in which your data source will define its connection string.

[✓ Create a data source based on an existing or new connection]

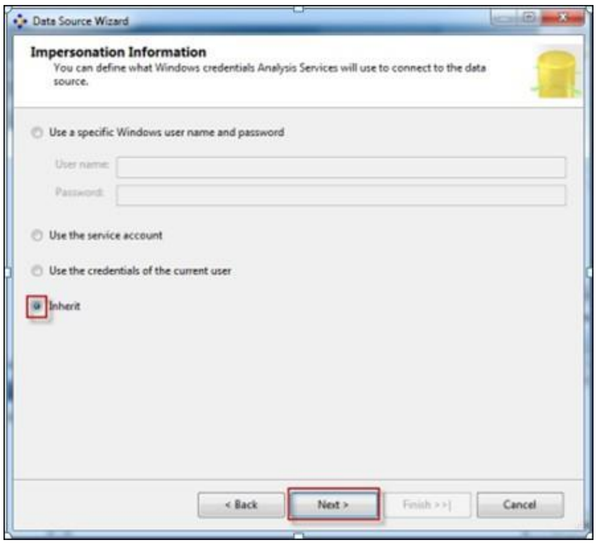
Data connections: MUBIN-PC\FAIRY.Sales_DW.sa

Data connection properties:

| Property | Value |
|---|---|
| Data Source | MUBIN-PC\FAIRY |
| Initial Catalog | Sales_DW |
| Persist Securi... | True |
| Provider | SQLNCLI10.1 |
| User ID | sa |

Create a data source based on another object

6. Select Option **Inherit**



**Impersonation Information**

You can define what Windows credentials Analysis Services will use to connect to the data source.

Use a specific Windows user name and password

User name:

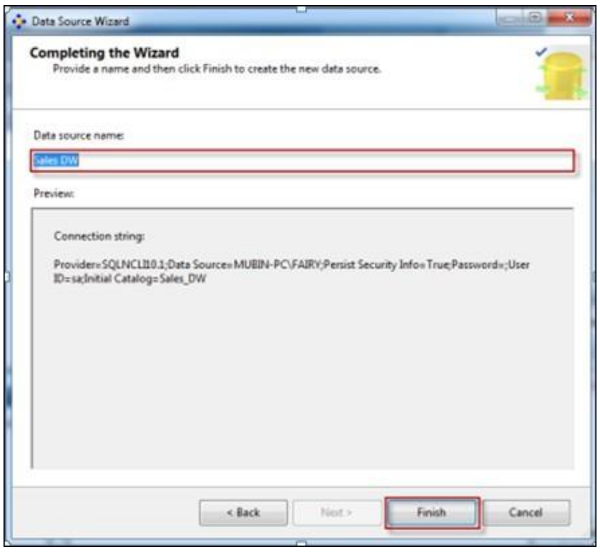
Password:

Use the service account

Use the credentials of the current user

[✓ Inherit]

7. Assign Data Source **Name** -> Click **Finish**



Completing the Wizard

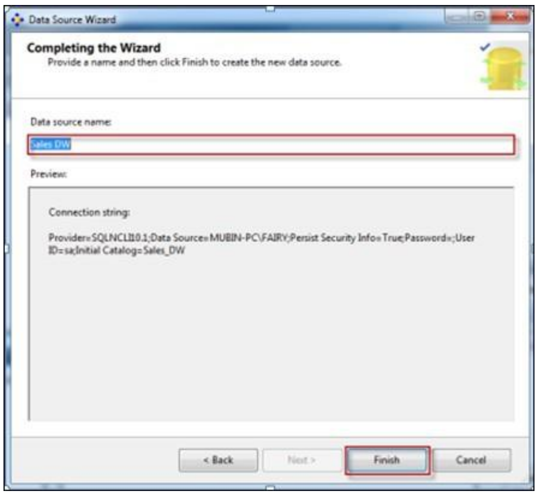
Provide a name and then click Finish to create the new data source.

Data source name: Sales DW

Preview:

Connection string:

Provider=SQLNCLI10.1;Data Source=MUBIN-PC\FAIRY;Persist Security Info=True;Password=;User ID=sa;Initial Catalog=Sales_DW



Completing the Wizard

Provide a name and then click Finish to create the new data source.

Data source name: Sales DW

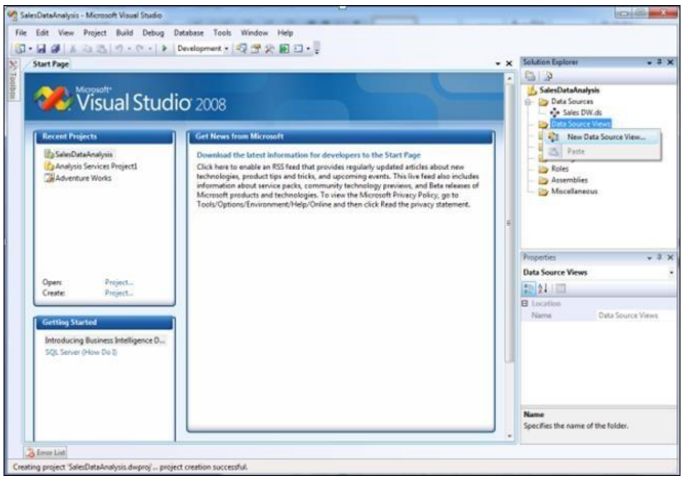
Preview:

Connection string:

Provider=SQLNCLI10.1;Data Source=MUBIN-PC\FAIRY;Persist Security Info=True;Password=;User ID=sa;Initial Catalog=Sales_DW

**Step 4: Creating New Data Source View**

1. In the Solution Explorer, Right Click on **Data Source View** -> Click on **New Data Source View**



Download the latest information for developers to the Start Page

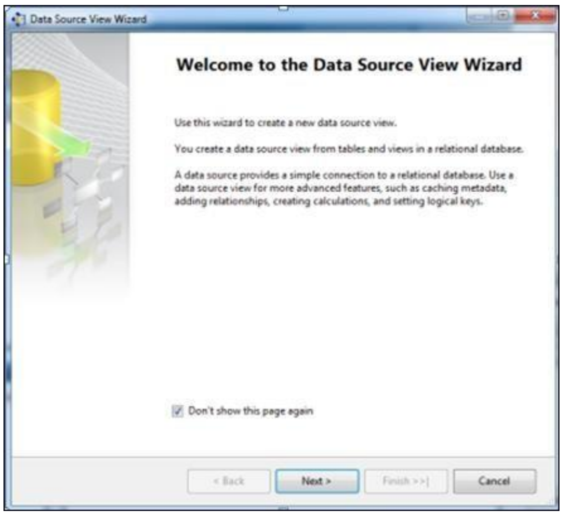
Click here to enable an RSS feed that provides regularly updated articles about new technologies, product tips and tricks, and upcoming events. This live feed also includes information about service packs, community technology previews, and Beta releases of Microsoft products and technologies. To view the Microsoft Privacy Policy, go to Tools/Options/Environment/Help/Online and then click Read the privacy statement.

Name

Specifies the name of the folder.

Creating project 'SalesDataAnalysis.dwproj'... project creation successful.

2. Click **Next**


Data Source View Wizard

Welcome to the Data Source View Wizard

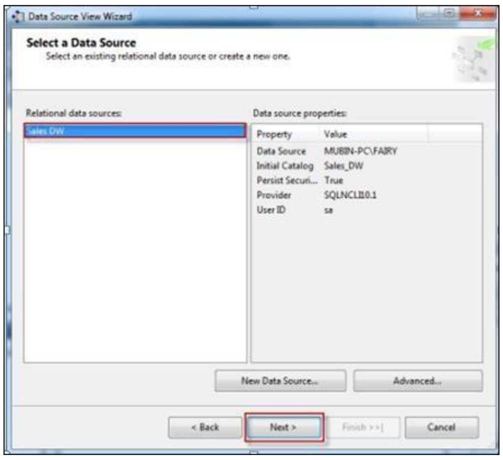
Use this wizard to create a new data source view.

You create a data source view from tables and views in a relational database.

A data source provides a simple connection to a relational database. Use a data source view for more advanced features, such as caching metadata, adding relationships, creating calculations, and setting logical keys.

[✓] Don't show this page again

3. Select **Relational Data Source** we have created previously (Sales_DW)-> Click **Next**


Data Source View Wizard

Select a Data Source

Select an existing relational data source or create a new one.

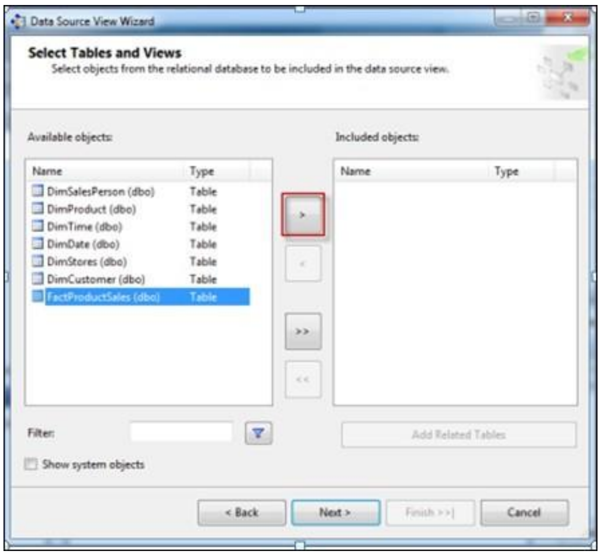
Relational data sources:

Sales DW

Data source properties:

| Property | Value |
|---|---|
| Data Source | MUBIN-PC\FAIRY |
| Initial Catalog | Sales_DW |
| Persist Securi... | True |
| Provider | SQLNCLI10.1 |
| User ID | sa |

4. First move your **Fact Table** to the right side to include in object list.


Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

| Name | Type |
|---|---|
| DimSalesPerson (dbo) | Table |
| DimProduct (dbo) | Table |
| DimTime (dbo) | Table |
| DimDate (dbo) | Table |
| DimStores (dbo) | Table |
| DimCustomer (dbo) | Table |
| FactProductSales (dbo) | Table |

Included objects:

| Name | Type |
|---|---|

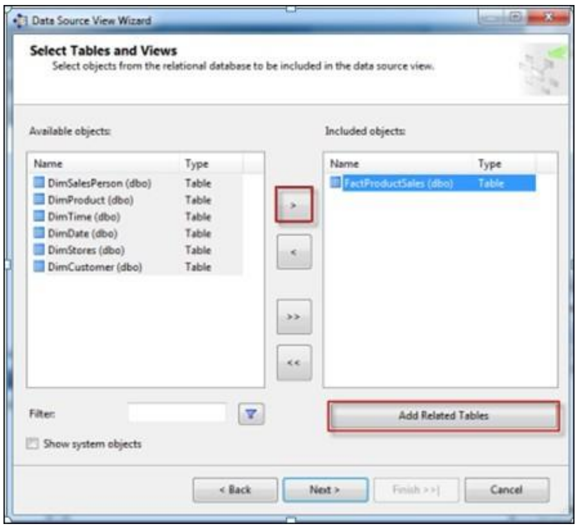
Filter:

[ ] Show system objects

Select FactProductSales Table -> Click on Arrow Button to move the selected object to Right Pane.

5. Now to **add dimensions** which are **related** to your **Fact Table**, follow the given steps:

Select **Fact Table** in Right Pane (Fact product Sales) -> Click On **Add Related Tables**



Select Tables and Views
Select objects from the relational database to be included in the data source view.

Available objects:

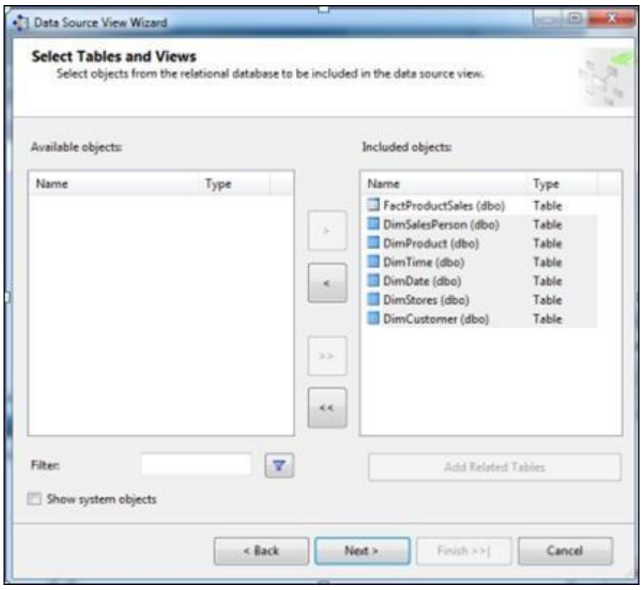
| Name | Type |
|---|---|
| DimSalesPerson (dbo) | Table |
| DimProduct (dbo) | Table |
| DimTime (dbo) | Table |
| DimDate (dbo) | Table |
| DimStores (dbo) | Table |
| DimCustomer (dbo) | Table |

Included objects:

| Name | Type |
|---|---|
| FactProductSales (dbo) | Table |

6. It will add all associated dimensions to your Fact table as per relationship specified in your SQL DW (Sales_DW).

Click **Next**.



Select Tables and Views
Select objects from the relational database to be included in the data source view.

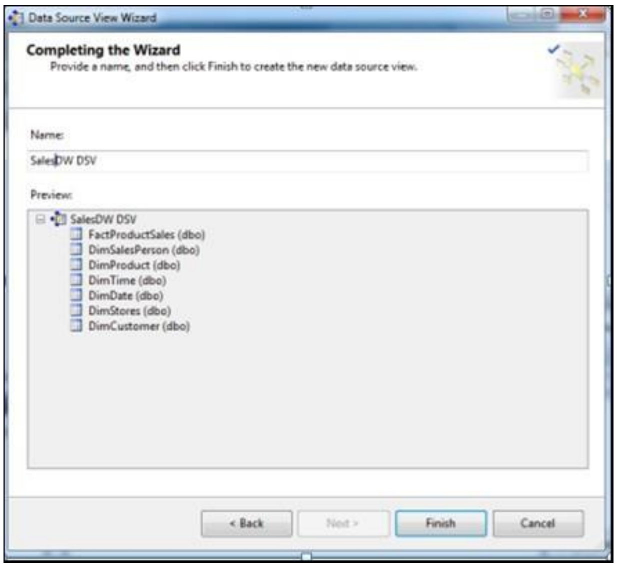
Available objects:

| Name | Type |
|---|---|

Included objects:

| Name | Type |
|---|---|
| FactProductSales (dbo) | Table |
| DimSalesPerson (dbo) | Table |
| DimProduct (dbo) | Table |
| DimTime (dbo) | Table |
| DimDate (dbo) | Table |
| DimStores (dbo) | Table |
| DimCustomer (dbo) | Table |

7. Assign **Name (SalesDW DSV)->** Click **Finish**


Data Source View Wizard

**Completing the Wizard**

Provide a name, and then click Finish to create the new data source view.

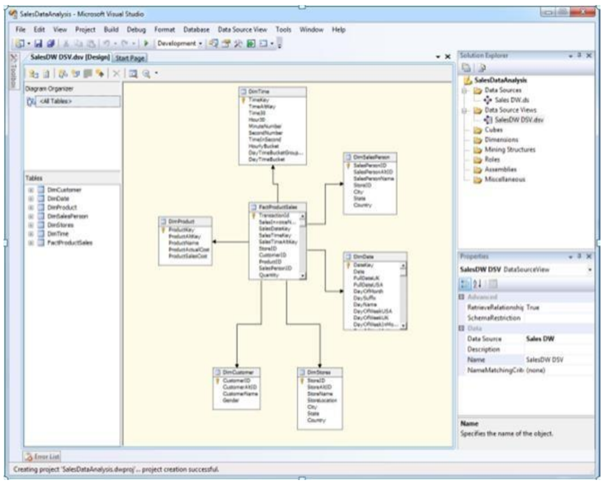
Name:

SalesDW DSV

Preview:

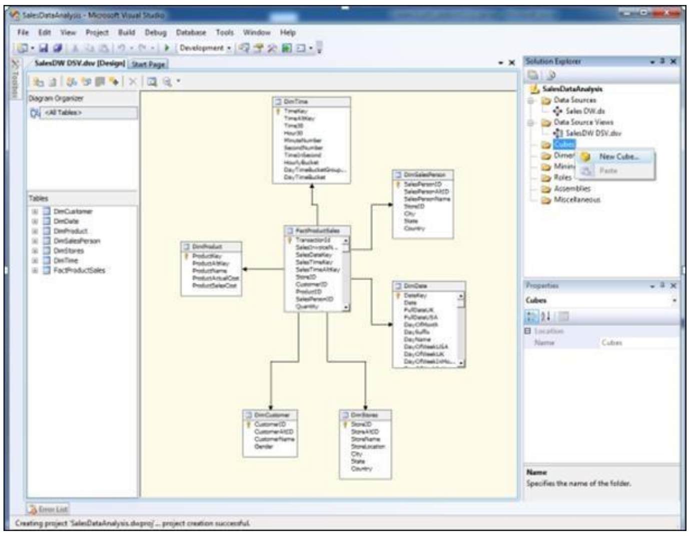
- SalesDW DSV
  - FactProductSales (dbo)
  - DimSalesPerson (dbo)
  - DimProduct (dbo)
  - DimTime (dbo)
  - DimDate (dbo)
  - DimStores (dbo)
  - DimCustomer (dbo)

8. **Now Data Source View is ready to use.**

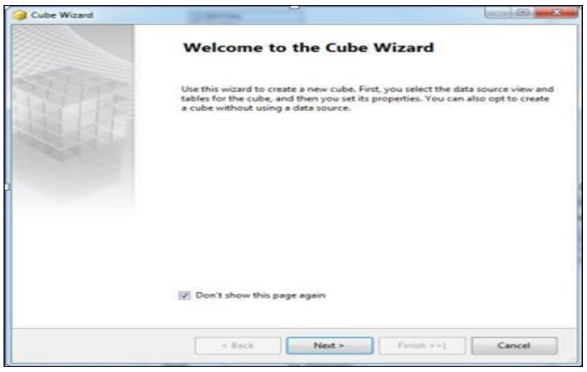


*Step 5: Creating New Cube*

1. In Solution Explorer -> Right Click on **Cube->** Click **New Cube**



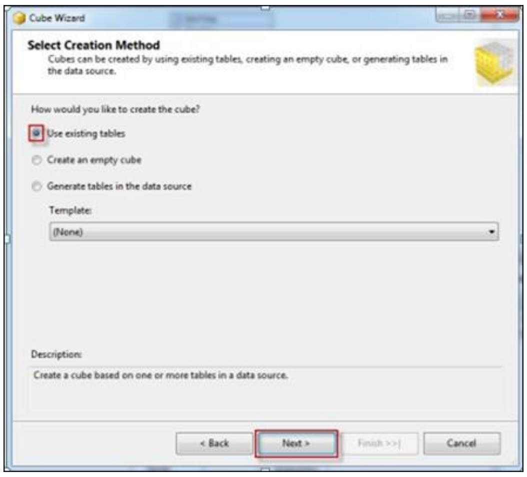
2. Click **Next**


Welcome to the Cube Wizard

Use this wizard to create a new cube. First, you select the data source view and tables for the cube, and then you set its properties. You can also opt to create a cube without using a data source.

☑ Don't show this page again

3. Select Option **Use existing Tables ->** Click **Next**


Select Creation Method

Cubes can be created by using existing tables, creating an empty cube, or generating tables in the data source.

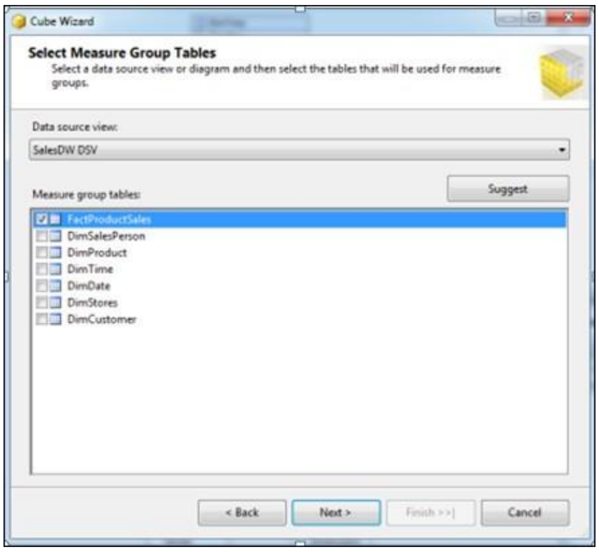
How would you like to create the cube?

[✓ Use existing tables]
Create an empty cube
Generate tables in the data source

Template: (None)

Description:
Create a cube based on one or more tables in a data source.

4. Select Fact Table Name from **Measure Group Tables (FactProductSales) ->** Click **Next**

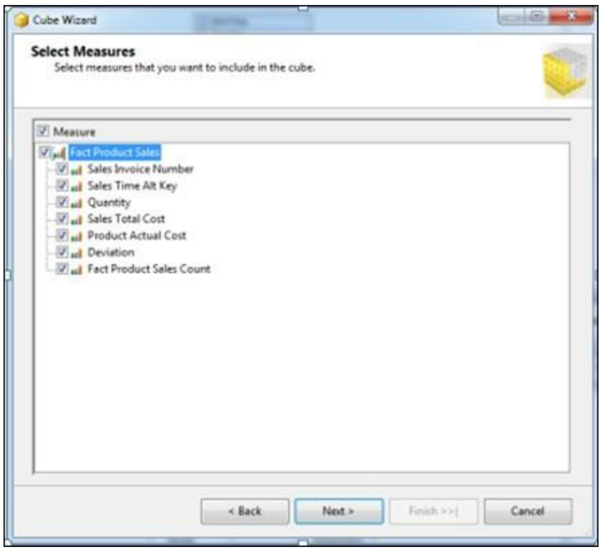

Select Measure Group Tables

Select a data source view or diagram and then select the tables that will be used for measure groups.

Data source view: SalesDW DSV

Measure group tables:
- [x] FactProductSales
- [ ] DimSalesPerson
- [ ] DimProduct
- [ ] DimTime
- [ ] DimDate
- [ ] DimStores
- [ ] DimCustomer

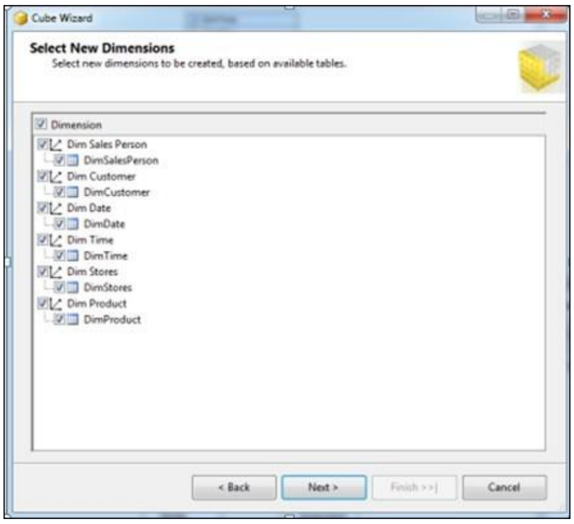
5. Choose **Measures** from the List which you want to place in your Cube --> Click **Next**



**Select Measures**
Select measures that you want to include in the cube.

- [✓] Measure
  - [✓] Fact Product Sales
    - [✓] Sales Invoice Number
    - [✓] Sales Time Alt Key
    - [✓] Quantity
    - [✓] Sales Total Cost
    - [✓] Product Actual Cost
    - [✓] Deviation
    - [✓] Fact Product Sales Count

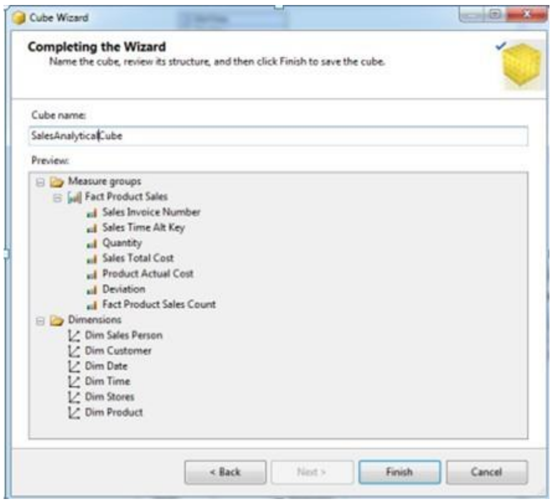
6. Select All **Dimensions** here which are associated with your Fact Table-> Click **Next**



**Select New Dimensions**
Select new dimensions to be created, based on available tables.

- [✓] Dimension
  - [✓] Dim Sales Person
    - [✓] DimSalesPerson
  - [✓] Dim Customer
    - [✓] DimCustomer
  - [✓] Dim Date
    - [✓] DimDate
  - [✓] Dim Time
    - [✓] DimTime
  - [✓] Dim Stores
    - [✓] DimStores
  - [✓] Dim Product
    - [✓] DimProduct

7. Assign **Cube Name** (SalesAnalyticalCube) -> Click **Finish**



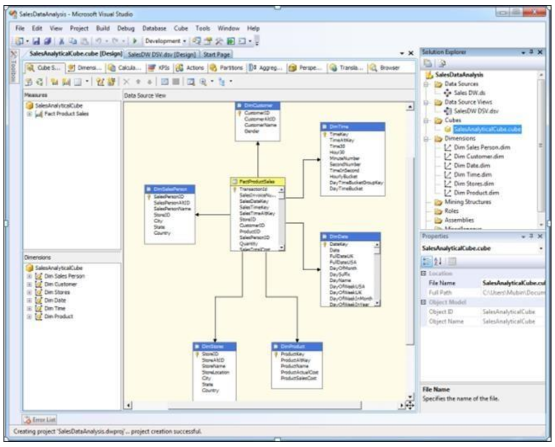
**Completing the Wizard**
Name the cube, review its structure, and then click Finish to save the cube.

Cube name: SalesAnalyticalCube

Preview:

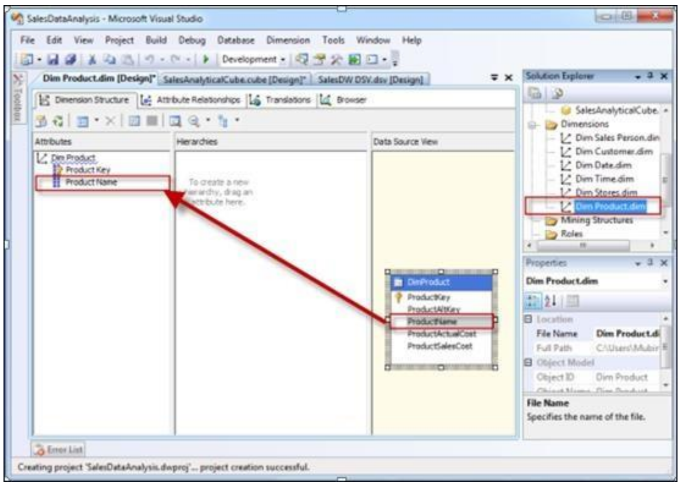
- Measure groups
  - Fact Product Sales
    - Sales Invoice Number
    - Sales Time Alt Key
    - Quantity
    - Sales Total Cost
    - Product Actual Cost
    - Deviation
    - Fact Product Sales Count
- Dimensions
  - Dim Sales Person
  - Dim Customer
  - Dim Date
  - Dim Time
  - Dim Stores
  - Dim Product

8. Now your Cube is ready, you can see the newly created cube and dimensions added in your solution explorer.



**Step 6: Dimension Modification**

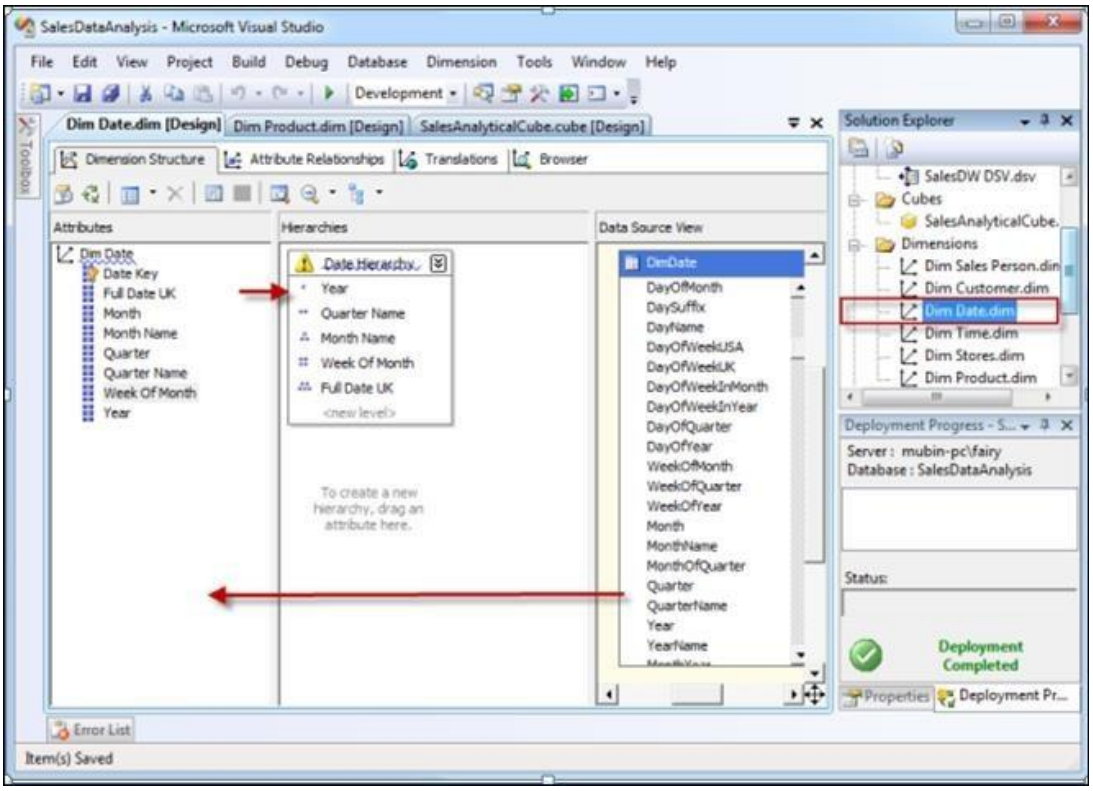
In Solution Explorer, double click on dimension **Dim Product ->** Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side.



*Step 7: Creating Attribute Hierarchy In Date Dimension*

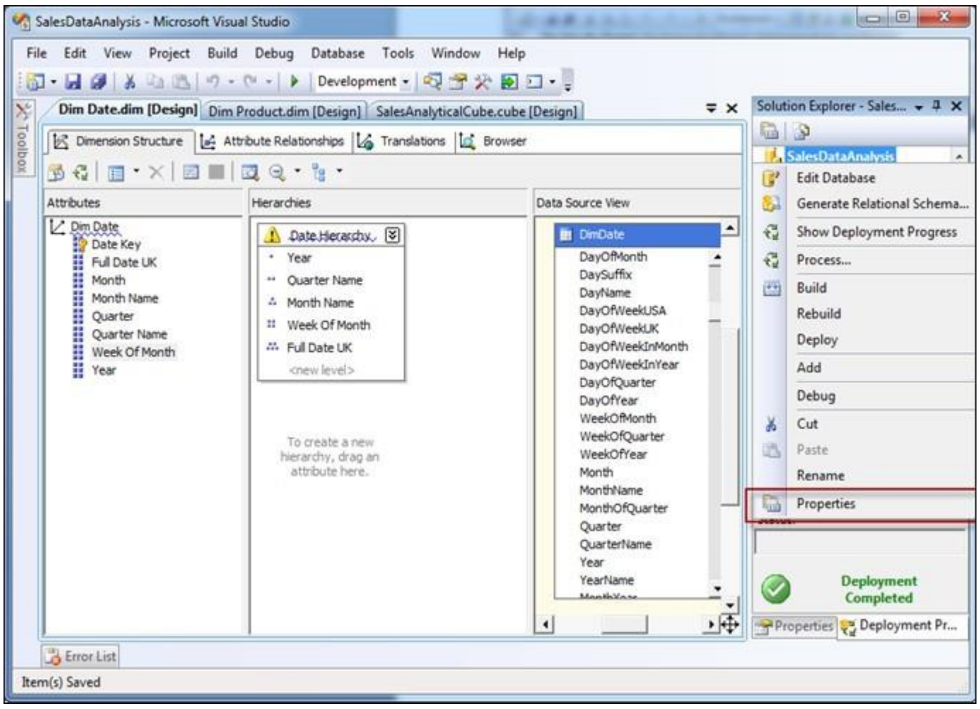
Double click On **Dim Date** dimension -> Drag and Drop Fields from Table shown in Data Source View to Attributes-> Drag and Drop attributes from leftmost pane of attributes to middle pane of Hierarchy.

Drag fields in sequence from Attributes to Hierarchy window (Year, Quarter Name, Month Name, Week of the Month, Full Date UK),



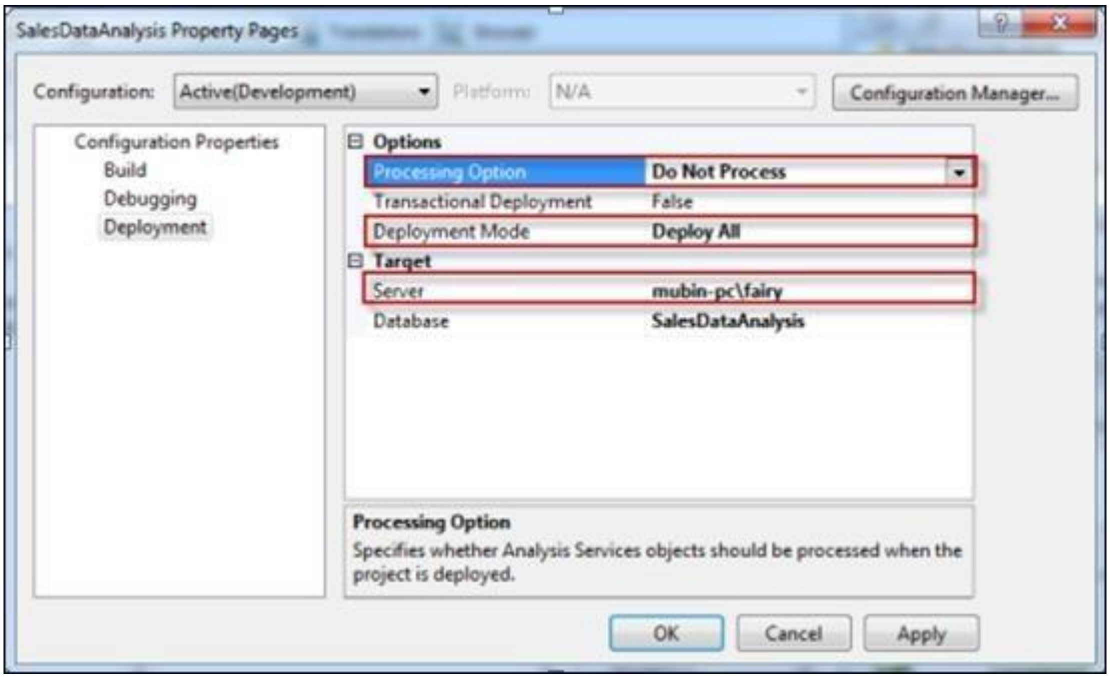
*Step 8: Deploy the Cube*

1. In Solution Explorer, right click on Project Name (SalesDataAnalysis) -- > Click **Properties**



2. Set **Deployment Properties** First

In Configuration Properties, Select Deployment-> Assign Your SQL Server Instance Name Where Analysis Services Is Installed (*mubin-pc\fairy*) (*Machine Name\Instance Name*) -> Choose Deployment Mode **Deploy All** as of now ->Select Processing Option **Do Not Process** -> Click **OK**



SalesDataAnalysis Property Pages

Configuration: Active(Development) Platform: N/A Configuration Manager...

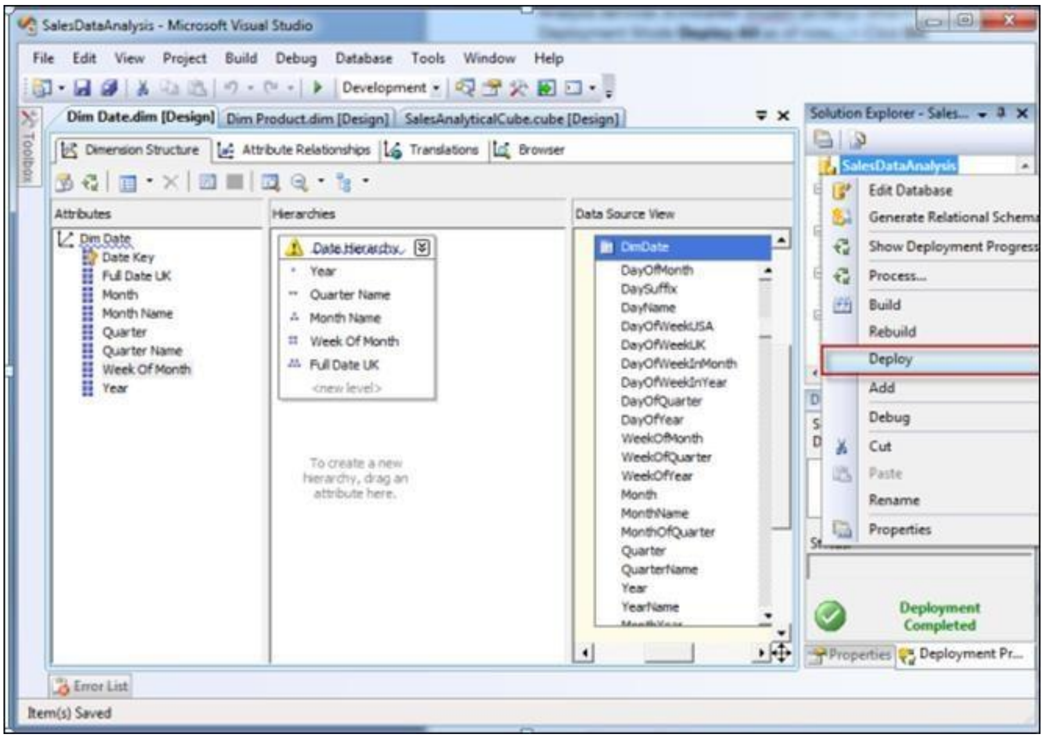
Configuration Properties
- Build
- Debugging
- Deployment

| Options | |
|---|---|
| Processing Option | Do Not Process |
| Transactional Deployment | False |
| Deployment Mode | Deploy All |
| **Target** | |
| Server | mubin-pc\fairy |
| Database | SalesDataAnalysis |

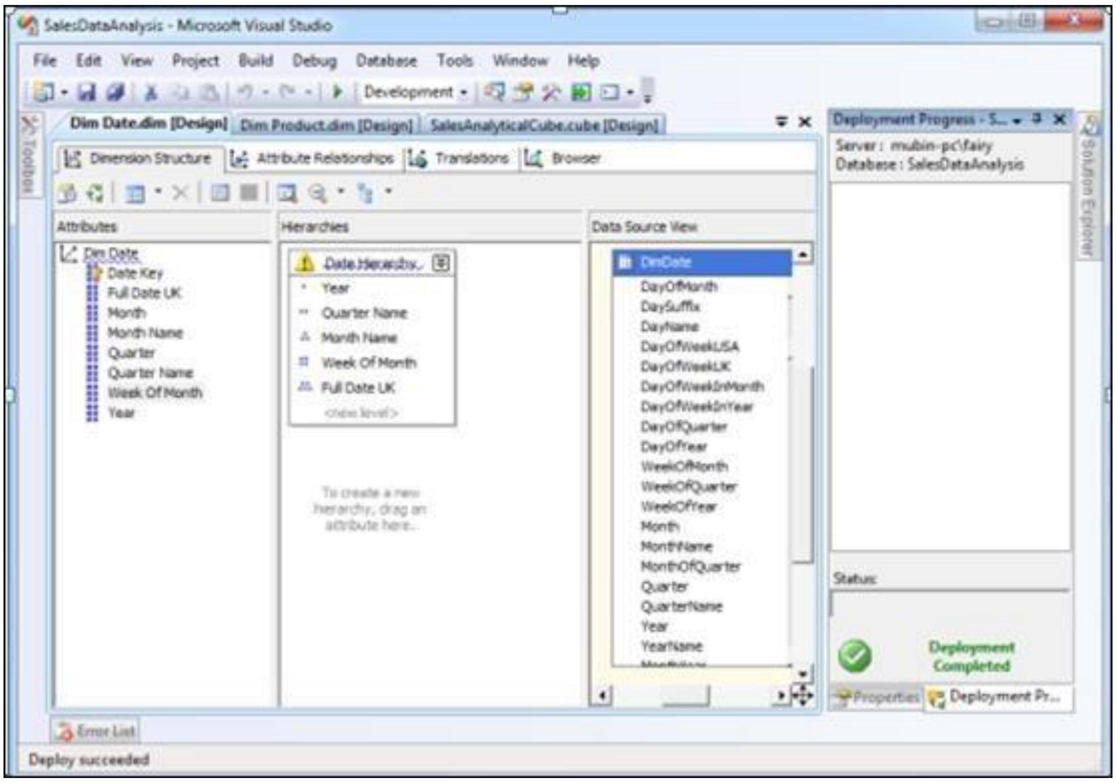
**Processing Option**
Specifies whether Analysis Services objects should be processed when the project is deployed.

OK Cancel Apply

3. In Solution Explorer, right click on **Project Name** (SalesDataAnalysis) -- > Click **Deploy**

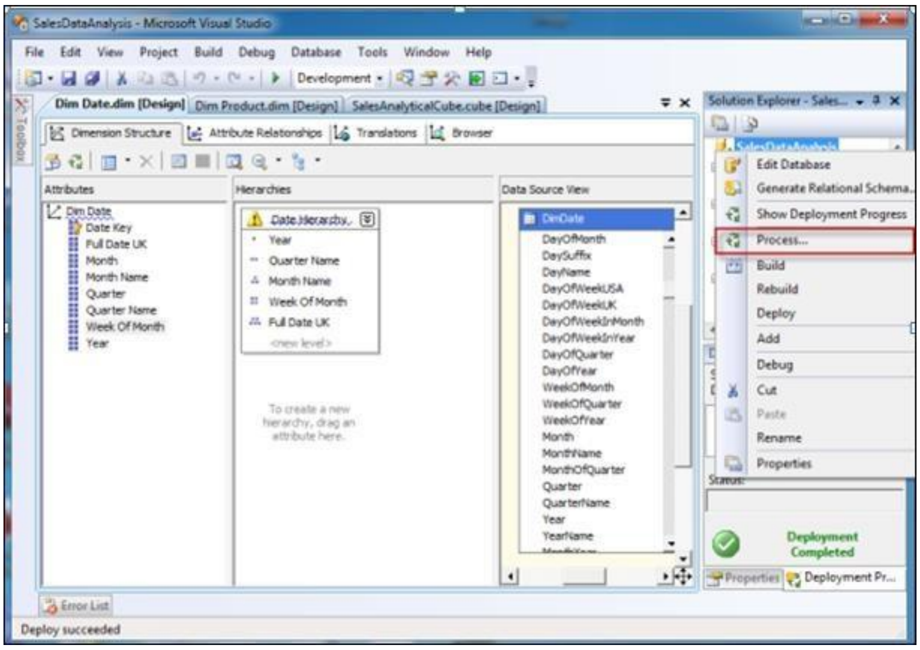


4. Once Deployment will finish, you can see the message **Deployment Completed** in deployment Properties.

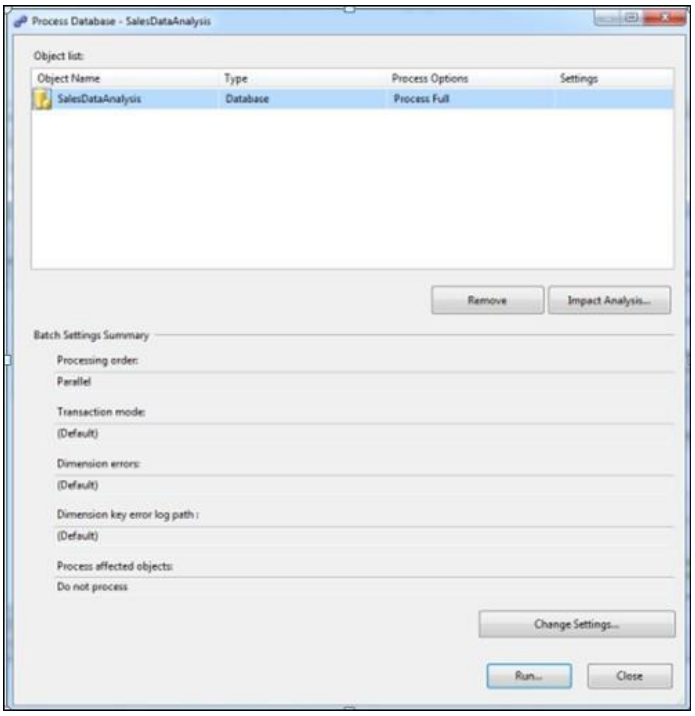


*Step 9: Process the Cube*

1. In Solution Explorer, right click on Project Name (SalesDataAnalysis) -- > Click **Process**



2. Click on **Run** button to process the Cube


Process Database - SalesDataAnalysis

Object list:

| Object Name | Type | Process Options | Settings |
|---|---|---|---|
| SalesDataAnalysis | Database | Process Full | |

Batch Settings Summary

Processing order:
Parallel

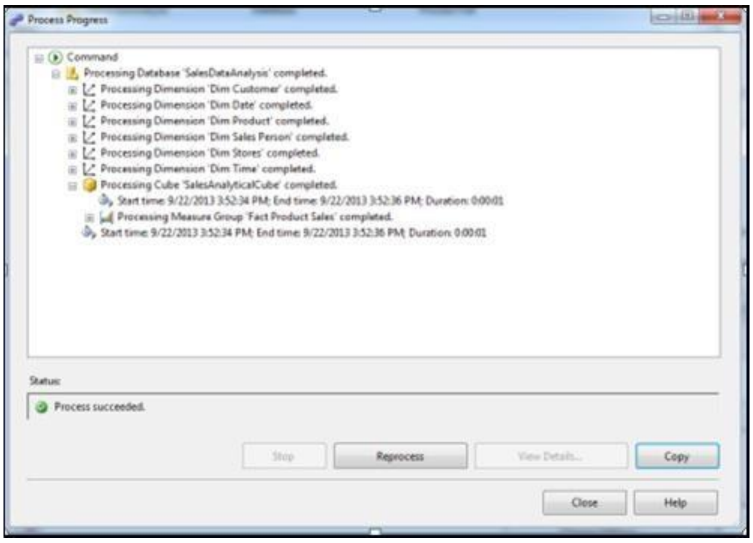
Transaction mode:
(Default)

Dimension errors:
(Default)

Dimension key error log path :
(Default)

Process affected objects:
Do not process

3. Once processing is complete, you can see **Status** as **Process Succeeded** -->Click **Close** to close both the open windows for processing one after the other.

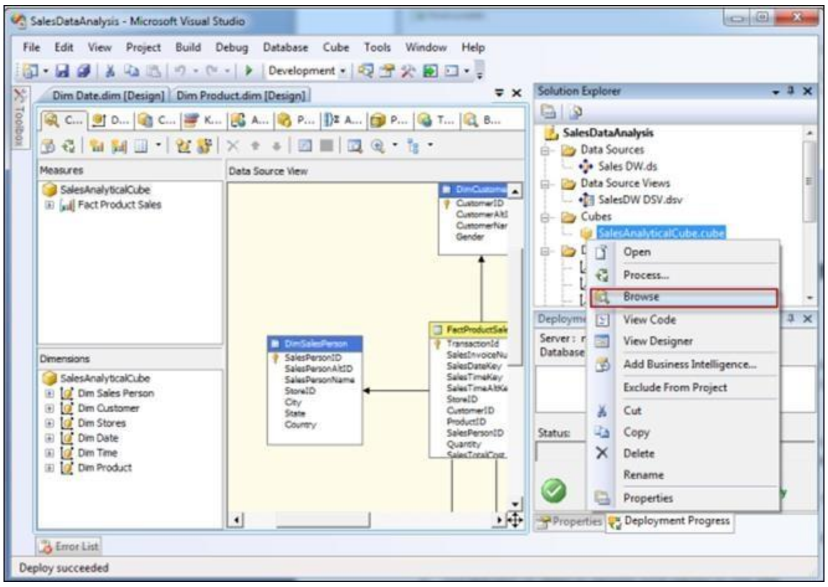

Process Progress

```
Command
  Processing Database 'SalesDataAnalysis' completed.
    Processing Dimension 'Dim Customer' completed.
    Processing Dimension 'Dim Date' completed.
    Processing Dimension 'Dim Product' completed.
    Processing Dimension 'Dim Sales Person' completed.
    Processing Dimension 'Dim Stores' completed.
    Processing Dimension 'Dim Time' completed.
    Processing Cube 'SalesAnalyticalCube' completed.
        Start time: 9/22/2013 3:52:34 PM; End time: 9/22/2013 3:52:36 PM; Duration: 0:00:01
      Processing Measure Group 'Fact Product Sales' completed.
    Start time: 9/22/2013 3:52:34 PM; End time: 9/22/2013 3:52:36 PM; Duration: 0:00:01
```

Status:
Process succeeded.

*Step 10: Browse the Cube for Analysis*

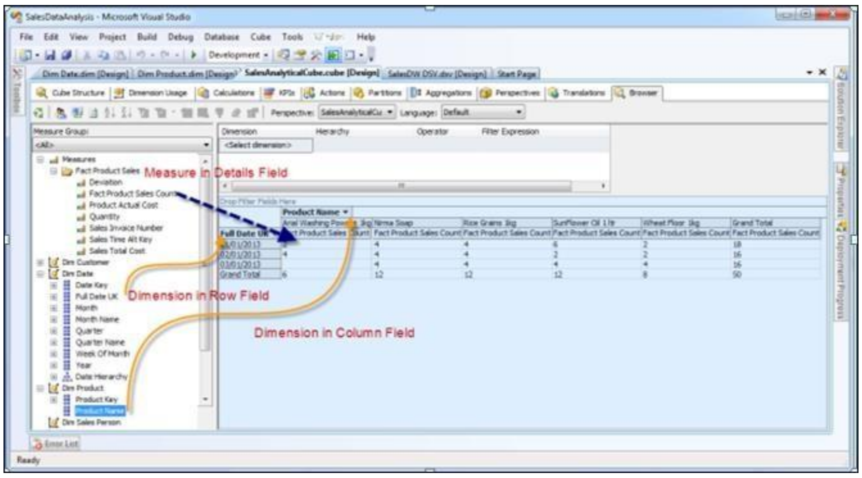
1. In Solution Explorer, right click on Cube Name (SalesDataAnalysisCube) -- > Click **Browse**



2. Drag and drop measures in to Detail fields, & Drag and Drop Dimension Attributes in Row Field or Column fields.

Now to **Browse Our Cube**

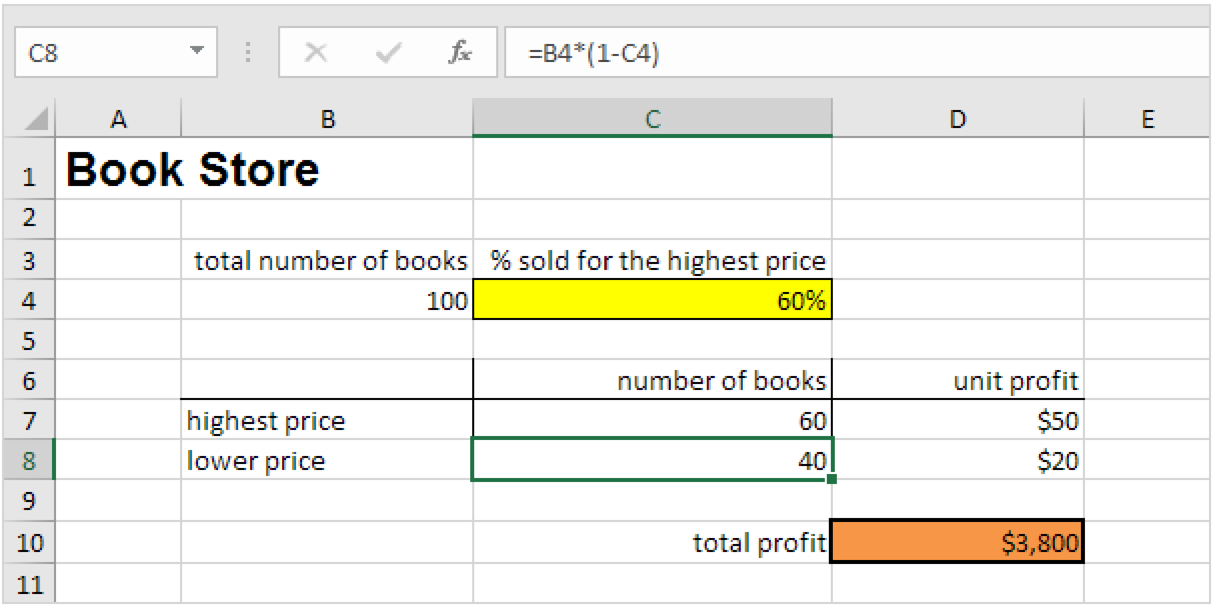
1. Product Name Drag & Drop into Column
2. Full Date UK Drag & Drop into Row Field
3. FactProductSalesCount Drop this measure in Detail area



# Practical 5

## Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data.

A book store and have 100 books in storage. You sell a certain % for the highest price of $50 and a certain % for the lower price of $20.



C8 =B4*(1-C4)

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Book Store | | | | |
| 2 | | | | | |
| 3 | | total number of books | % sold for the highest price | | |
| 4 | | 100 | 60% | | |
| 5 | | | | | |
| 6 | | | number of books | unit profit | |
| 7 | | highest price | 60 | $50 | |
| 8 | | lower price | 40 | $20 | |
| 9 | | | | | |
| 10 | | | total profit | $3,800 | |
| 11 | | | | | |

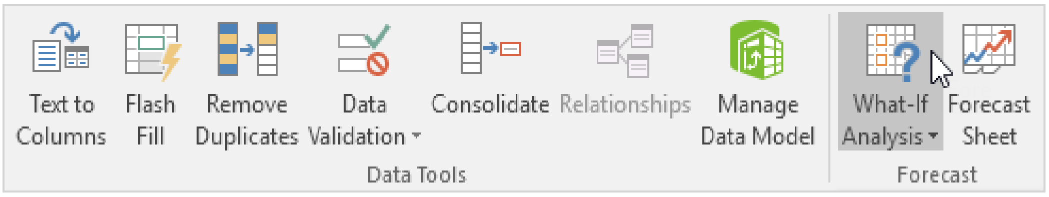
If you sell 60% for the highest price, cell D10 calculates a total profit of 60 * $50 + 40 * $20 = $3800.

Create Different Scenarios

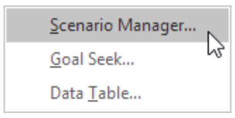
But what if you sell 70% for the highest price? And what if you sell 80% for the highest price? Or 90%, or even 100%? Each different percentage is a different **scenario**. You can use the Scenario Manager to create these scenarios.

Note: You can simply type in a different percentage into cell C4 to see the corresponding result of a scenario in cell D10. However, what-if analysis enables you to easily compare the results of different scenarios. Read on.

1. On the Data tab, in the Forecast group, click What-If Analysis.

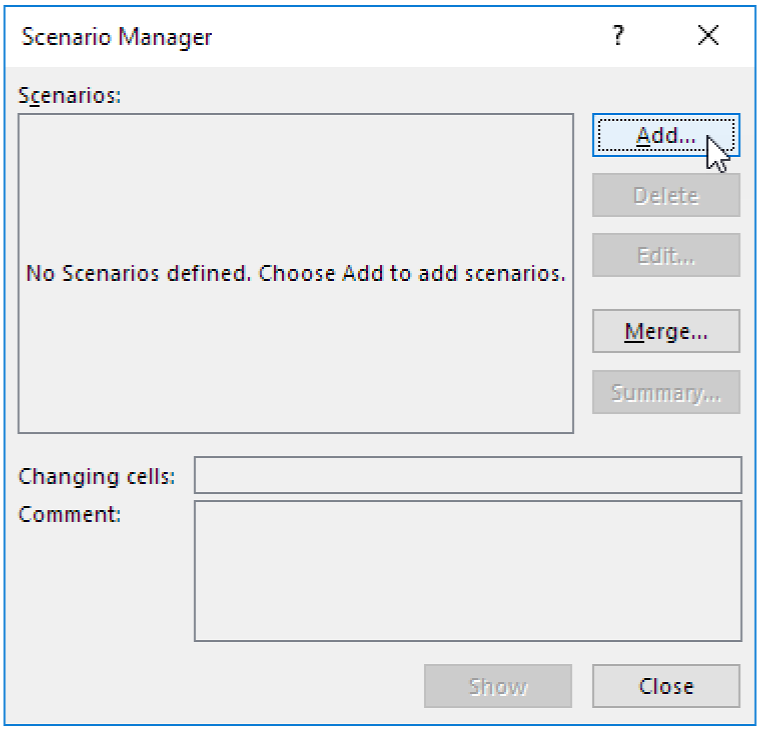


2. Click Scenario Manager.



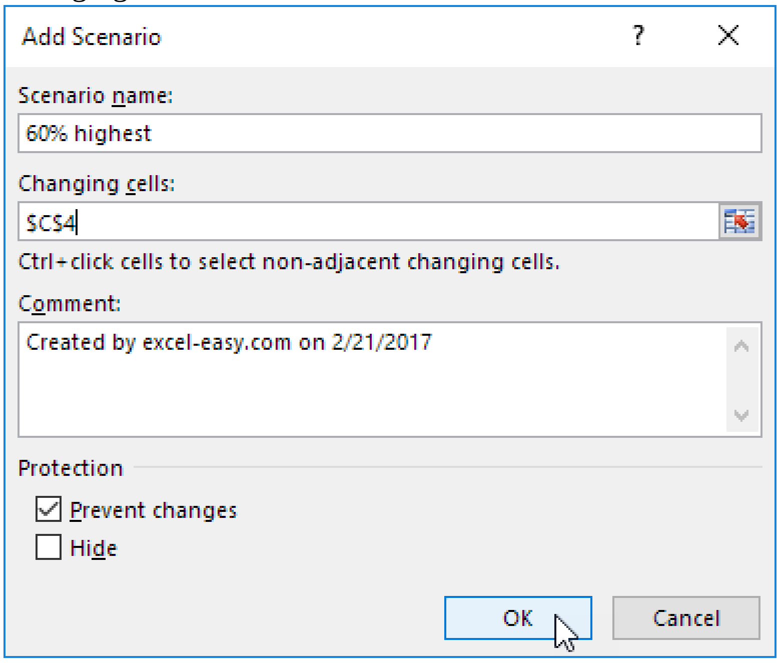
The Scenario Manager dialog box appears.

3. Add a scenario by clicking on Add.



No Scenarios defined. Choose Add to add scenarios.

4. Type a name (60% highest), select cell C4 (% sold for the highest price) for the Changing cells and click on OK.



Scenario name: 60% highest

Changing cells: $C$4

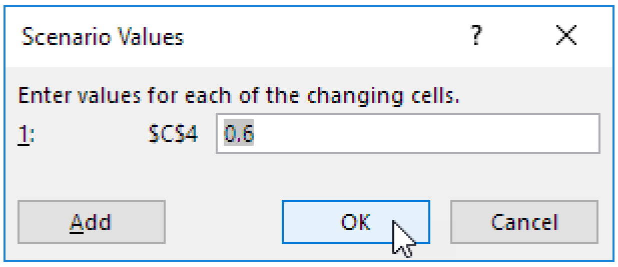
Ctrl+click cells to select non-adjacent changing cells.

Comment: Created by excel-easy.com on 2/21/2017

☑ Prevent changes

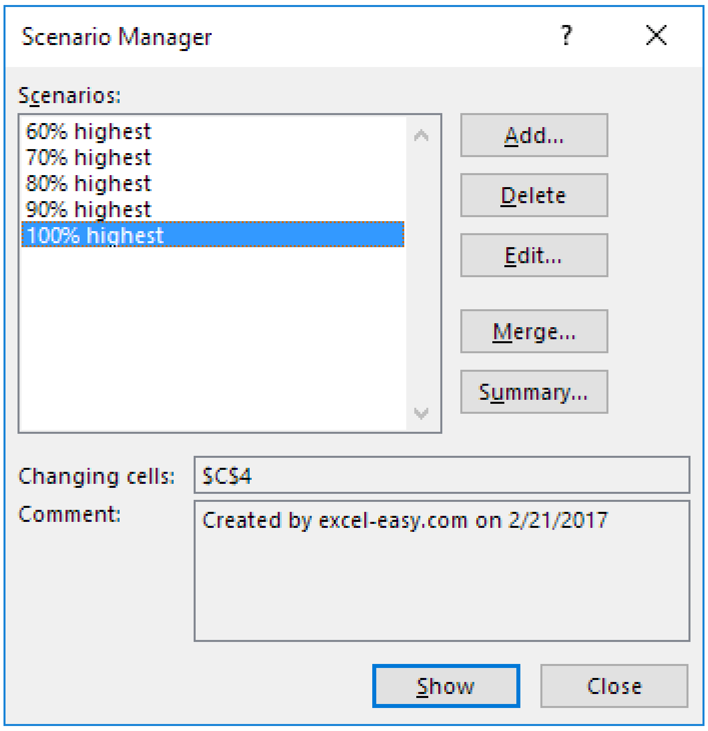
☐ Hide

5. Enter the corresponding value 0.6 and click on OK again.



6. Next, add 4 other scenarios (70%, 80%, 90% and 100%).

Finally, your Scenario Manager should be consistent with the picture below:



# Practical 6

## Implementation of Classification algorithm in R Programming.

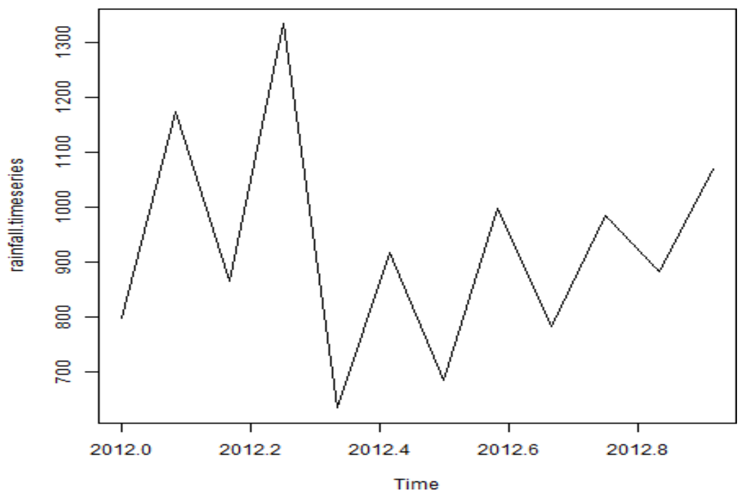
Consider the annual rainfall details at a place starting from January 2012. We create an R time series object for a period of 12 months and plot it.

```
# Get the data points in form of a R vector.
rainfall <-
c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)

# Convert it to a time series object.
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)

# Print the timeseries data.
print(rainfall.timeseries)

# Give the chart file a name.
png(file = "rainfall.png")

# Plot a graph of the time series.
plot(rainfall.timeseries)

# Save the file.
dev.off()
```

Output:

When we execute the above code, it produces the following result and chart −

```
  Jan     Feb     Mar     Apr     May      Jun     Jul     Aug     Sep

  201    799.    1174.8  865.1   1334.    635.    918.    685.    998.6
  2         0                    6        4       5       5       784.2

         Oct       Nov     Dec

  201    985.    882.8  1071.0
  2         0
```



## Practical Implementation of Decision Tree using R Tool

```
install.packages("party")
```

The package "party" has the function **ctree()** which is used to create and analyze decison tree.

### Syntax

The basic syntax for creating a decision tree in R is −

```
ctree(formula, data)
```

### Input Data

We will use the R in-built data set named **readingSkills** to create a decision tree. It describes the score of someone's readingSkills if we know the variables "age","shoesize","score" and whether the person is a native speaker or not.

Here is the sample data.

```
# Load the party package. It will automatically load other
# dependent packages.
library(party)

# Print some records from data set readingSkills.
print(head(readingSkills))
```

When we execute the above code, it produces the following result and chart −

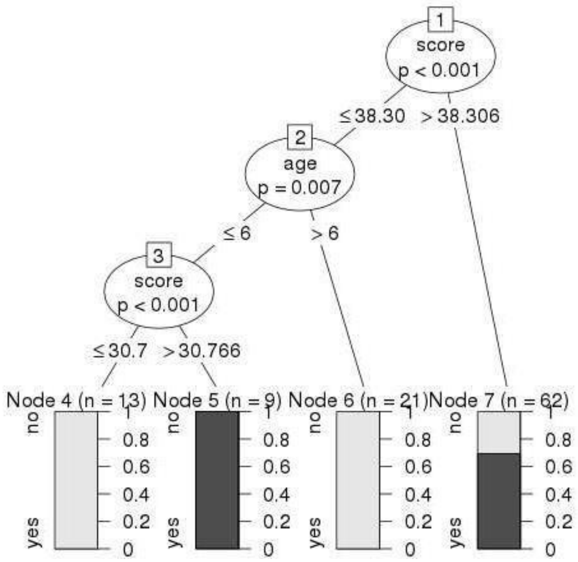
```
  nativeSpeaker   age   shoeSize     score

   1                5   24.83189  32.29385
           yes

   2                6   25.95238  36.63105
           yes

  3                11   30.42170  49.60593
            no

   4                7   28.66450  40.28456
           yes

   5               11   31.88207  55.46085
           yes

   6               10   30.07843  52.83124
           yes

Loading required package: methods
Loading required package: grid
...............................
...............................
```

We will use the **ctree()** function to create the decision tree and see its graph.

```
# Load the party package. It will automatically load other
# dependent packages.
library(party)

# Create the input data frame.
input.dat <- readingSkills[c(1:105),]

# Give the chart file a name.
png(file = "decision_tree.png")

# Create the tree.
  output.tree <- ctree(
  nativeSpeaker ~ age + shoeSize + score,
  data = input.dat)

# Plot the tree.
plot(output.tree)

# Save the file.
dev.off()
```

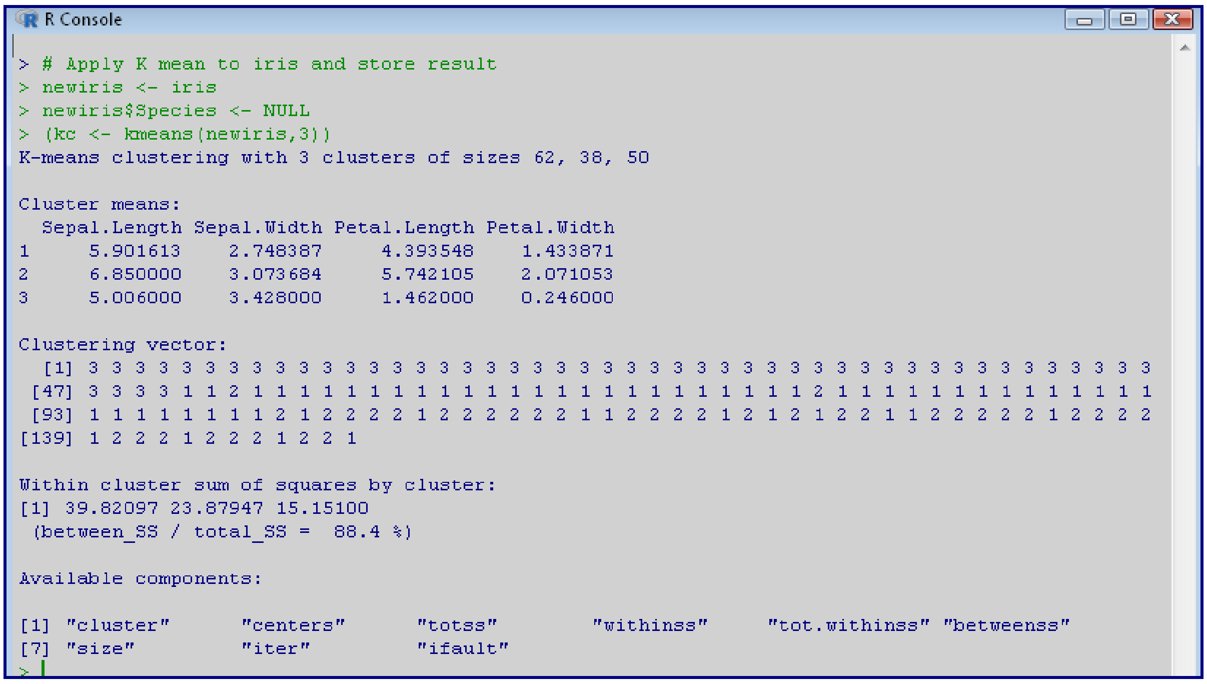
**Output:-**

```
 null device
          1
Loading required package: methods
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

as.Date, as.Date.numeric

Loading required package: sandwich
```



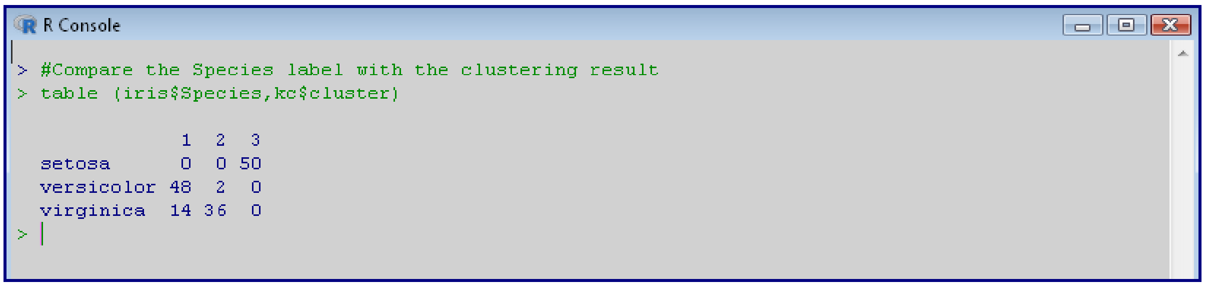
# Practical 7

## K-means clustering using R



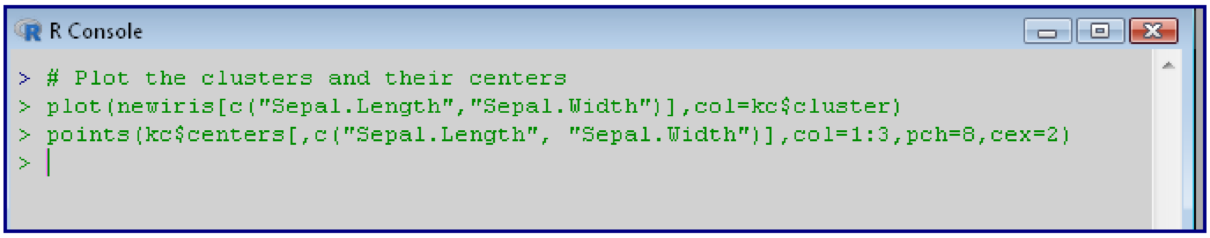
```
> # Apply K mean to iris and store result
> newiris <- iris
> newiris$Species <- NULL
> (kc <- kmeans(newiris,3))
K-means clustering with 3 clusters of sizes 62, 38, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.901613    2.748387     4.393548    1.433871
2     6.850000    3.073684     5.742105    2.071053
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [47] 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [93] 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2
[139] 1 2 2 2 1 2 2 2 1 2 2 1

Within cluster sum of squares by cluster:
[1] 39.82097 23.87947 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> 
```

Compare the Species label with the clustering result



```
> #Compare the Species label with the clustering result
> table (iris$Species,kc$cluster)

              1  2  3
  setosa      0  0 50
  versicolor 48  2  0
  virginica  14 36  0
> 
```

Plot the clusters and their centre



```
> # Plot the clusters and their centers
> plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)
> points(kc$centers[,c("Sepal.Length", "Sepal.Width")],col=1:3,pch=8,cex=2)
> 
```

# Practical 8

## Prediction Using Linear Regression

In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

$y = ax + b$ is an equation for linear regression.

Where, **y** is the response variable, **x** is the predictor variable and ***a*** and ***b*** are constants which are called the coefficients.

A simple example of regression is predicting weight of a person when his height is known. To do this we need to have the relationship between height and weight of a person.

The steps to create the relationship is −

- Carry out the experiment of gathering a sample of observed values of height and corresponding weight.
- Create a relationship model using the **lm()** functions in R.
- Find the coefficients from the model created and create the mathematical equation using these
- Get a summary of the relationship model to know the average error in prediction. Also called **residuals**.
- To predict the weight of new persons, use the **predict()** function in R.

### Input Data

Below is the sample data representing the observations −

```
# Values of height
151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# Values of weight.
63, 81, 56, 91, 47, 57, 76, 72, 62, 48
```

lm() Function

This function creates the relationship model between the predictor and the response variable.

### Syntax

The basic syntax for **lm()** function in linear regression is −

```
lm(formula,data)
```

Following is the description of the parameters used −

- **formula** is a symbol presenting the relation between x and y.
- **data** is the vector on which the formula will be applied.
-

## Create Relationship Model & get the Coefficients

```
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

print(relation)
```

When we execute the above code, it produces the following result –

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551          0.6746
```

## Get the Summary of the Relationship

```
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

print(summary(relation))
```

When we execute the above code, it produces the following result –

```
call:
lm(formula = y ~ x)

Residuals:
    Min      1Q     Median      3Q     Max
-6.3002    -1.6629  0.0412     1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

predict() Function

## Syntax

The basic syntax for predict() in linear regression is −

```
predict(object, newdata)
```

Following is the description of the parameters used −

- **object** is the formula which is already created using the lm() function.
- **newdata** is the vector containing the new value for predictor variable.

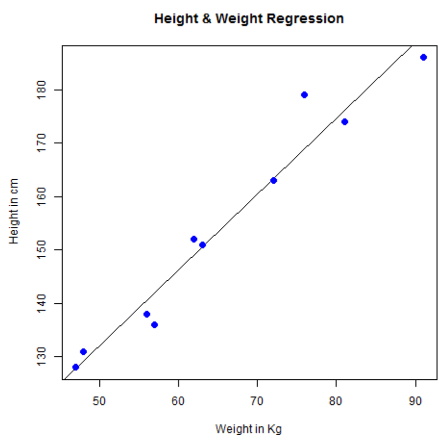
## Predict the weight of new persons

```
# The predictor vector.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# The resposne vector.
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

# Find weight of a person with height 170.
a <- data.frame(x = 170)
result <-  predict(relation,a)
print(result)

Result:
  1
76.22869
```

## Visualize the Regression Graphically

```
# Create the predictor and response variable.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation <- lm(y~x)

# Give the chart file a name.
png(file = "linearregression.png")

# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in
cm")

# Save the file.
dev.off()
```

output:



# Practical 9

## Data Analysis using Time Series Analysis

Time series is a series of data points in which each data point is associated with a timestamp. A simple example is the price of a stock in the stock market at different points of time on a given day. Another example is the amount of rainfall in a region at different months of the year. R language uses many functions to create, manipulate and plot the time series data. The data for the time series is stored in an R object called **time-series object**. It is also a R data object like a vector or data frame.

The time series object is created by using the **ts()** function.

### Syntax

The basic syntax for **ts()** function in time series analysis is −

```
timeseries.object.name <-  ts(data, start, end, frequency)
```

Following is the description of the parameters used −

- **data** is a vector or matrix containing the values used in the time series.
- **start** specifies the start time for the first observation in time series.
- **end** specifies the end time for the last observation in time series.
- **frequency** specifies the number of observations per unit time.

Except the parameter "data" all other parameters are optional.

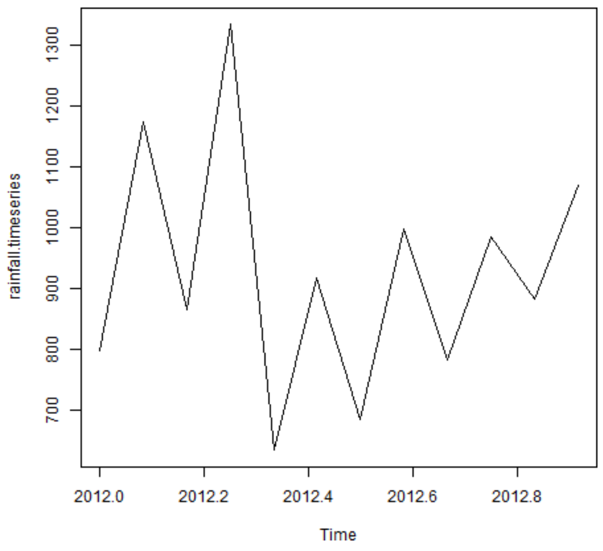
### Example

Consider the annual rainfall details at a place starting from January 2012. We create an R time series object for a period of 12 months and plot it.

```
# Get the data points in form of a R vector.
rainfall <-
c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)

# Convert it to a time series object.
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)

# Print the timeseries data.
print(rainfall.timeseries)

# Give the chart file a name.
png(file = "rainfall.png")

# Plot a graph of the time series.
plot(rainfall.timeseries)

# Save the file.
dev.off()
```

**Output:-**

```
Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep

201     799.     1174.8  865.1    1334.    635.     918.     685.     998.6
 2         0                      6        4        5        5        784.2
        Oct        Nov      Dec

201     985.     882.8 1071.0
 2         0
```



# Practical 10

## Data Modelling and Analytics with Pivot Table in Excel

**Data Model** is used for building a model where data from various sources can be combined by creating relationships among the data sources. A Data Model integrates the tables, enabling extensive analysis using PivotTables, Power Pivot, and Power View.

A **Data Model** is created automatically when you import two or more tables simultaneously from a database. The existing database relationships between those tables is used to create the Data Model in Excel.

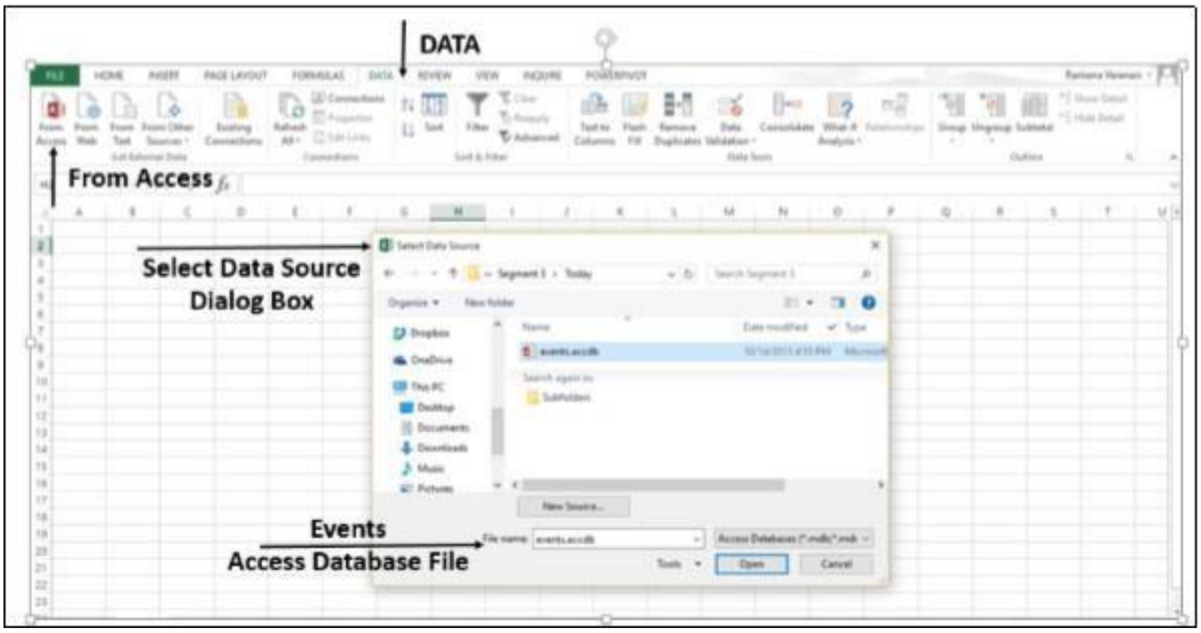
**Step 1** − Open a new blank Workbook in Excel.

**Step 2** − Click on the **DATA** tab.

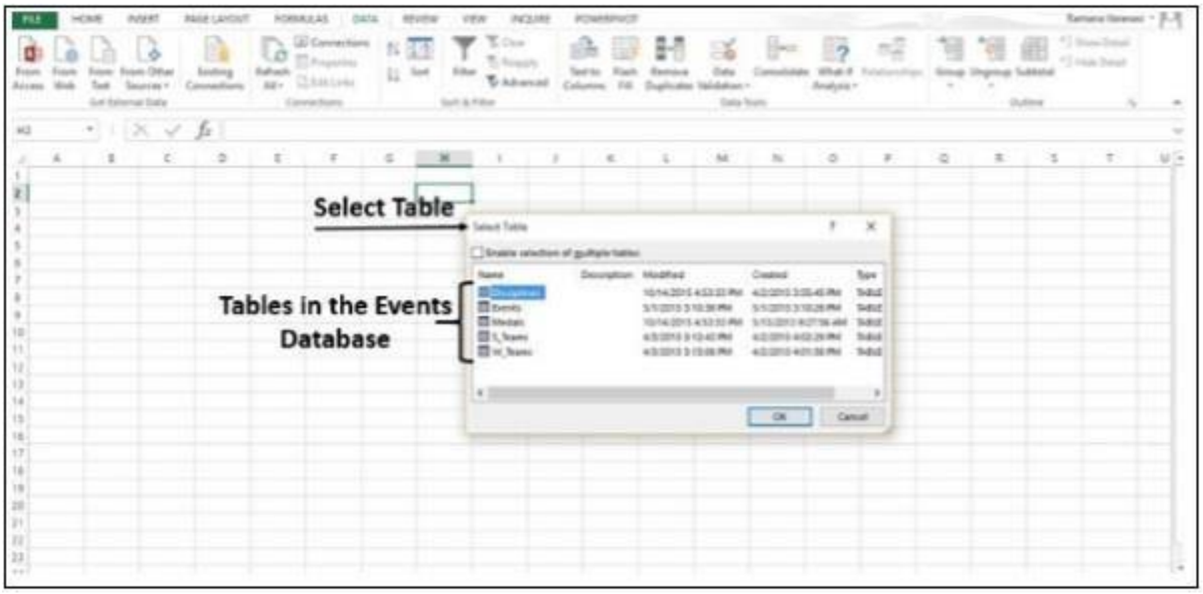
**Step 3** − In the **Get External Data** group, click on the option **From Access**. The **Select Data Source** dialog box opens.

**Step 4** − Select **Events.accdb**, Events Access Database file.

(url:/ https://fcschools.instructure.com/courses/373/files/10607)

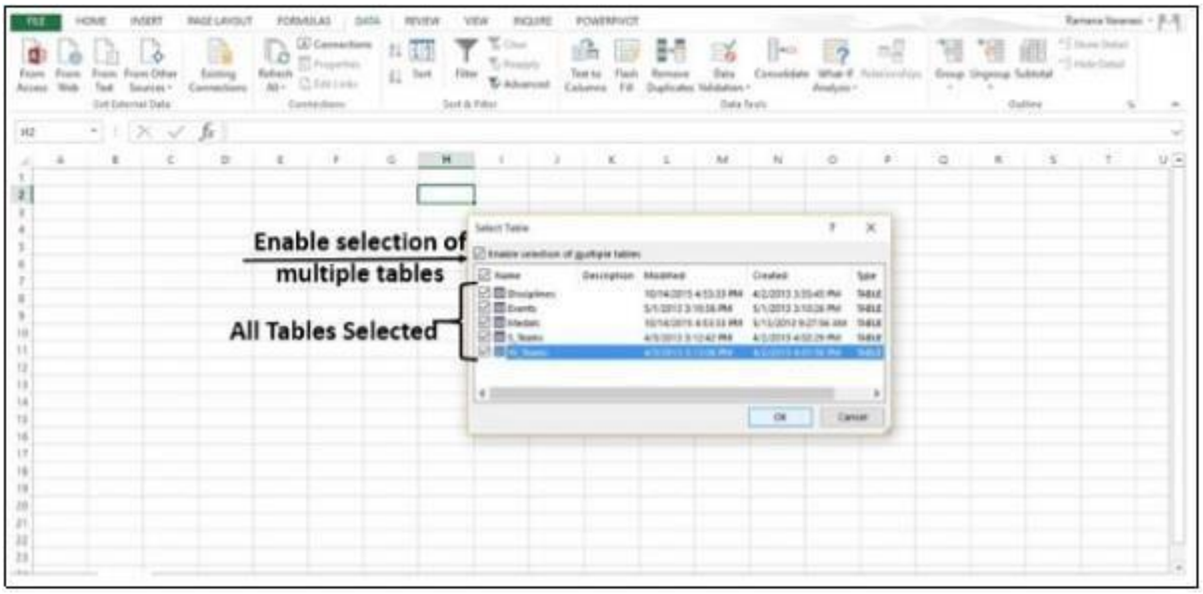


**Step 5** − The **Select Table** window, displaying all the **tables** found in the database, appears.

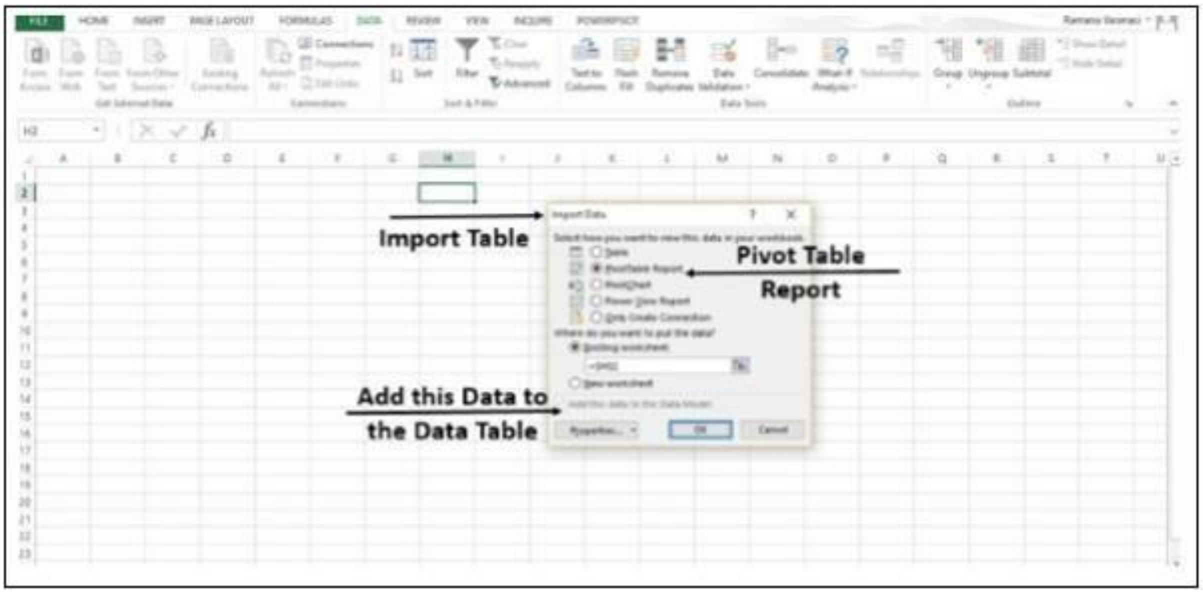


**Step 6** − Tables in a database are similar to the tables in Excel. Check the '**Enable selection of multiple tables**' box, and select all the tables. Then click **OK**.

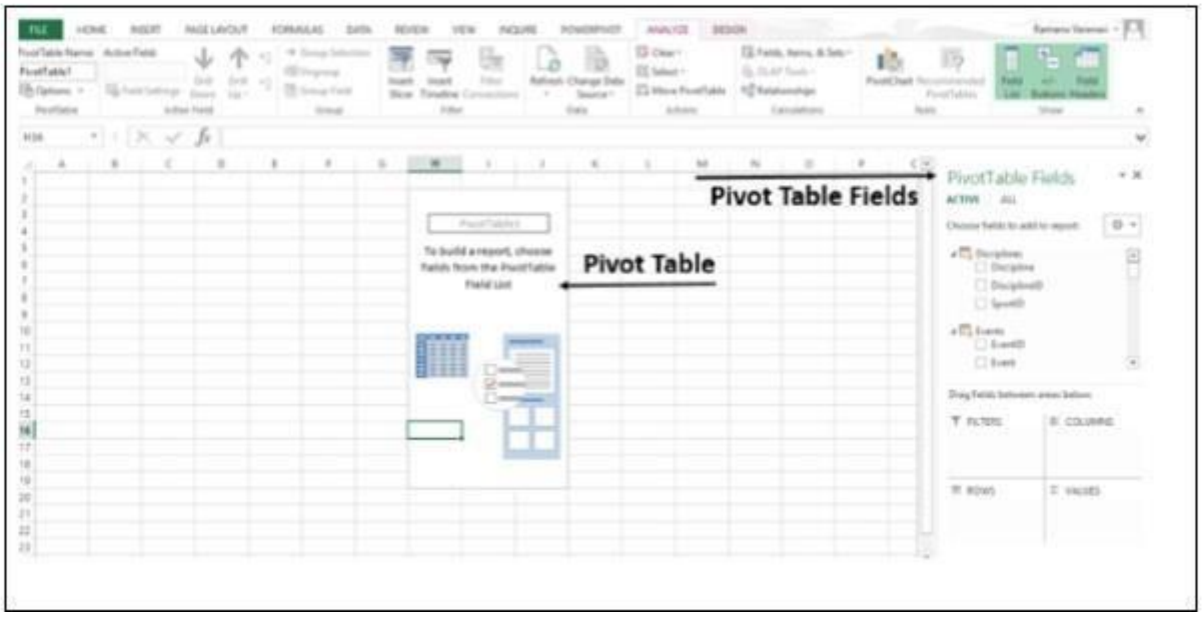
**Step 7** − The **Import Data** window appears. Select the **PivotTable Report** option. This option imports the tables into Excel and prepares a PivotTable for analyzing the imported tables.



Notice that the checkbox at the bottom of the window - '**Add this data to the Data Model**' is selected and disabled.



**Step 8** − The data is imported, and a **PivotTable** is created using the imported tables.



**Explore Data Using PivotTable**

**Step 1** − You know how to add fields to PivotTable and drag fields across areas. Even if you are not sure of the final report that you want, you can play with the data and choose the best-suited report.

In **PivotTable Fields**, click on the arrow beside the table - **Medals** to expand it to show the fields in that table. Drag the **NOC_CountryRegion** field in the **Medals** table to the **COLUMNS** area.

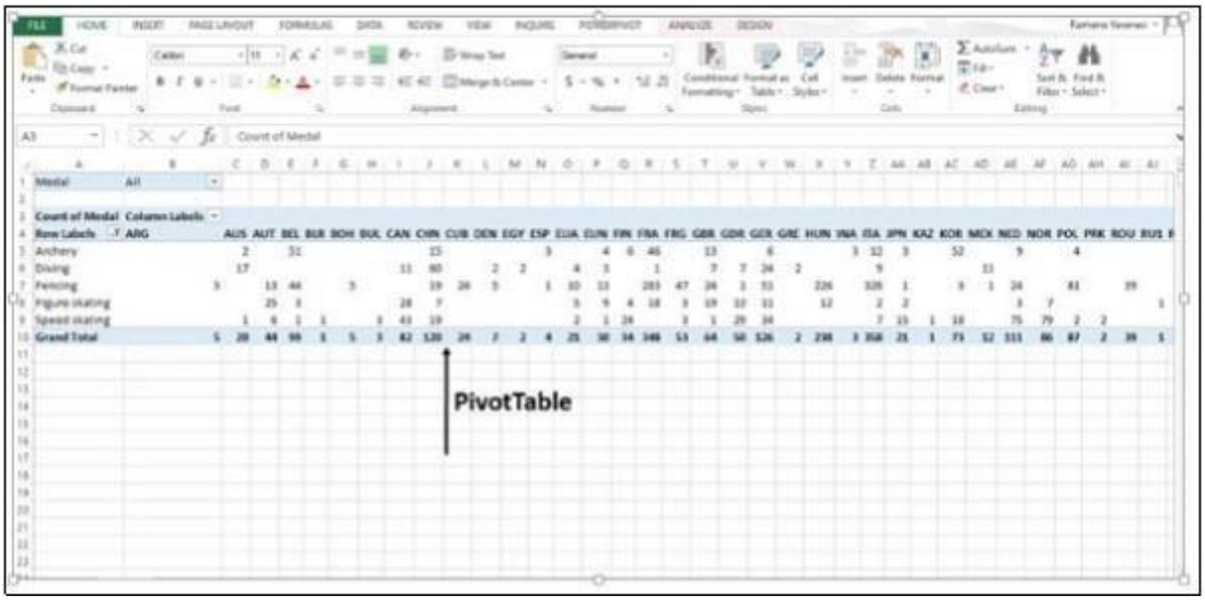
**Step 2** − Drag **Discipline** from the **Disciplines** table to the **ROWS** area.

**Step 3** − Filter **Discipline** to display only five sports: Archery, Diving, Fencing, Figure Skating, and Speed Skating. This can be done either in **PivotTable Fields** area, or from the

**Row Labels** filter in the PivotTable itself.

**Step 4** − In **PivotTable Fields**, from the **Medals** table, drag Medal to the **VALUES** area.

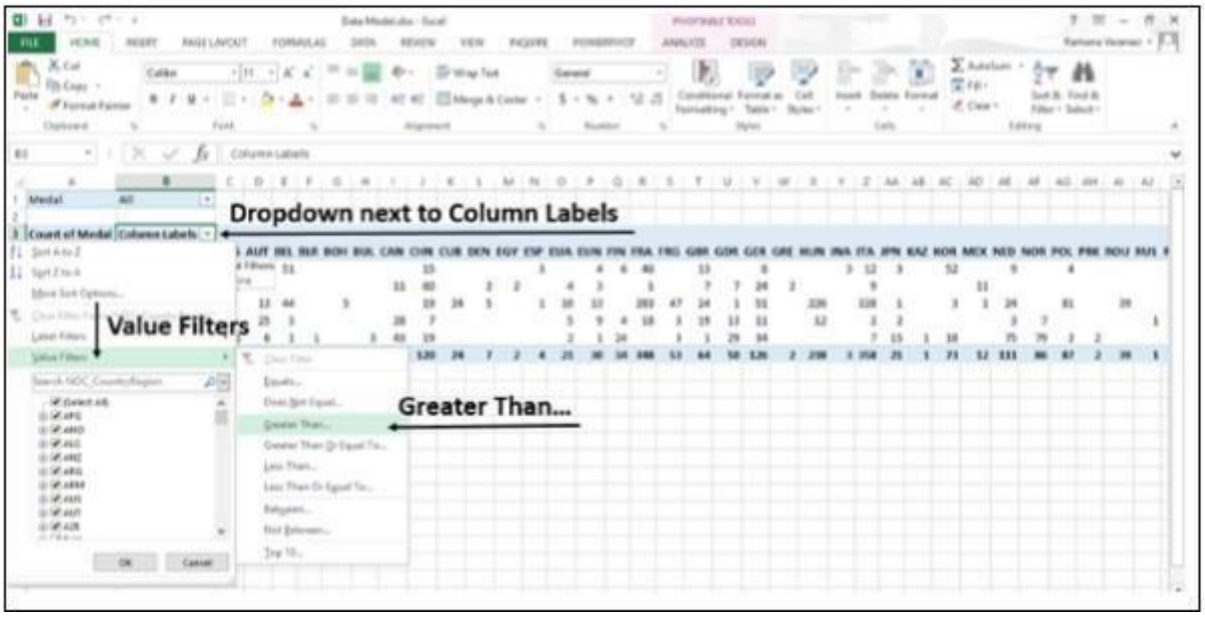
**Step 5** − From the **Medals** table, select **Medal** again and drag it into the **FILTERS** area.



**Step 6** − Click the dropdown list button to the right of the **Column** labels.

**Step 7** − Select **Value Filters** and then select **Greater Than**…

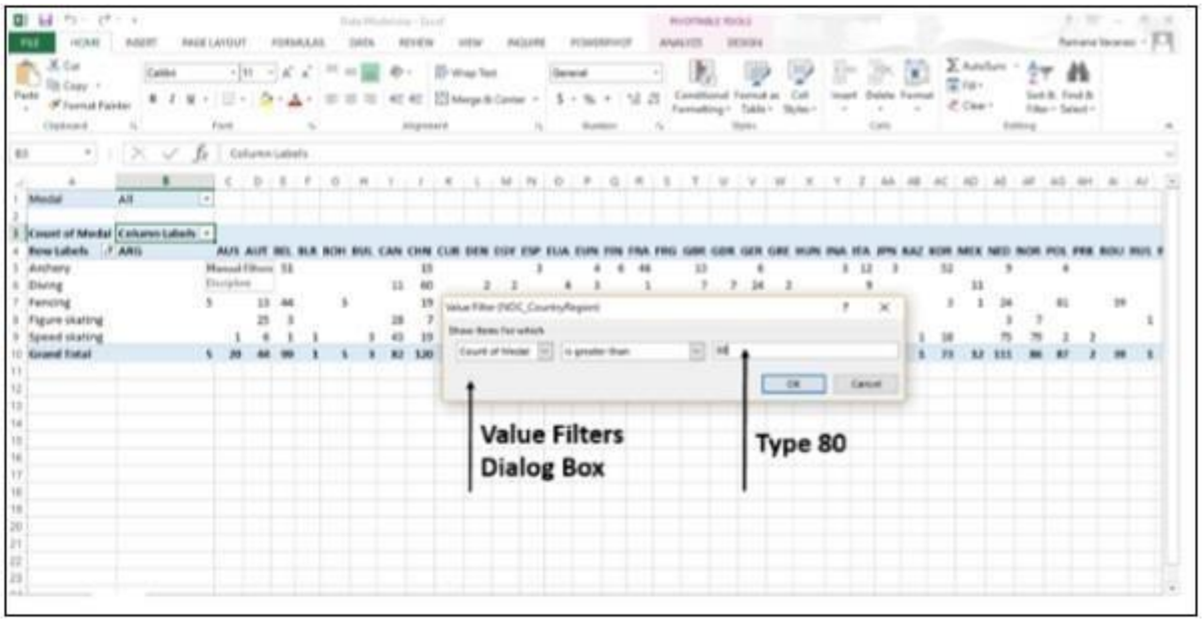
**Step 8** − Click **OK**.



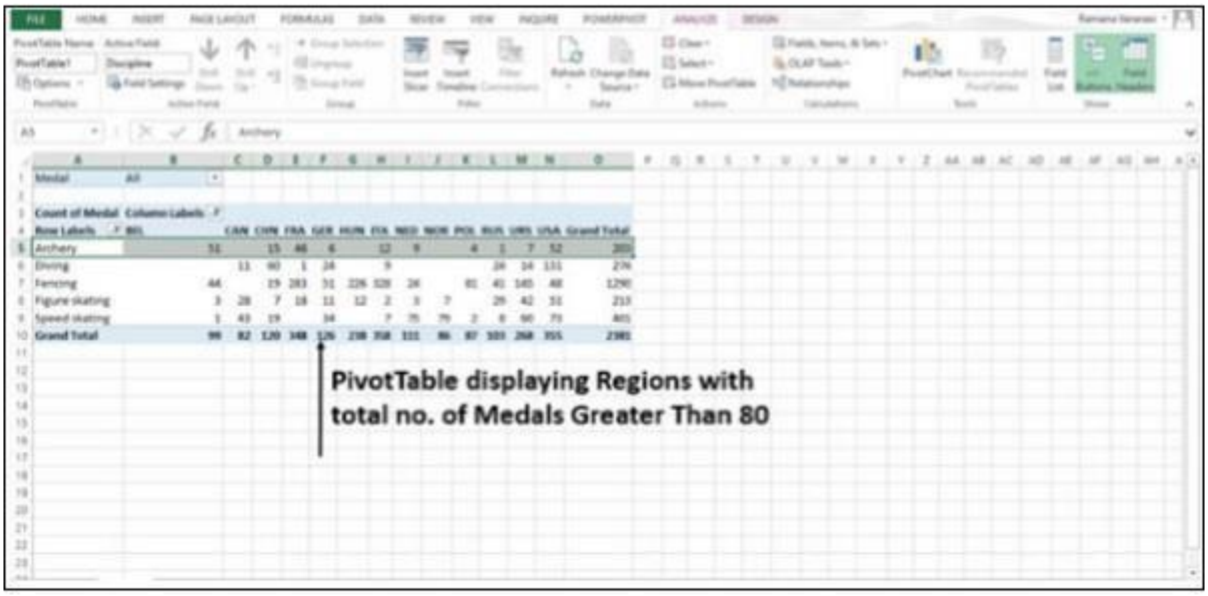
The Value **Filters dialog** box for the count of Medals **is greater than** appears.

**Step 9** − Type **80** in the **Right Field**.

**Step 10** − Click **OK**.



The PivotTable displays only those regions, which has more than total 80 medals.

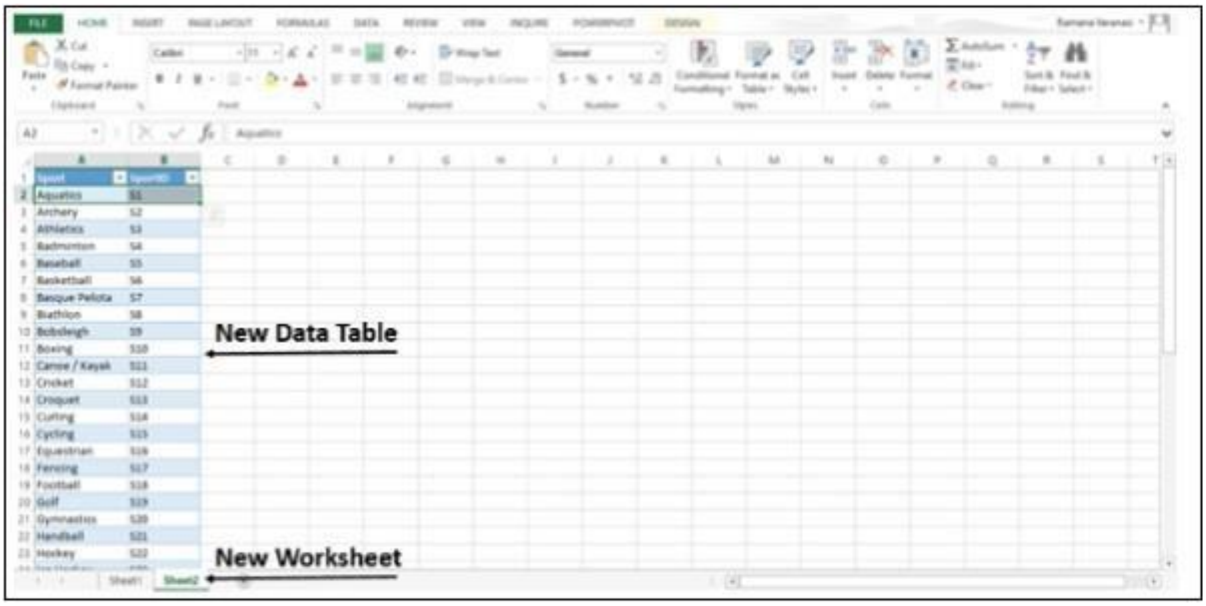

PivotTable displaying Regions with total no. of Medals Greater Than 80

## Create Relationship between Tables

Relationships let you analyze your collections of the data in Excel, and create interesting and aesthetic reports from the data you import.

**Step 1** − **Insert** a new Worksheet.

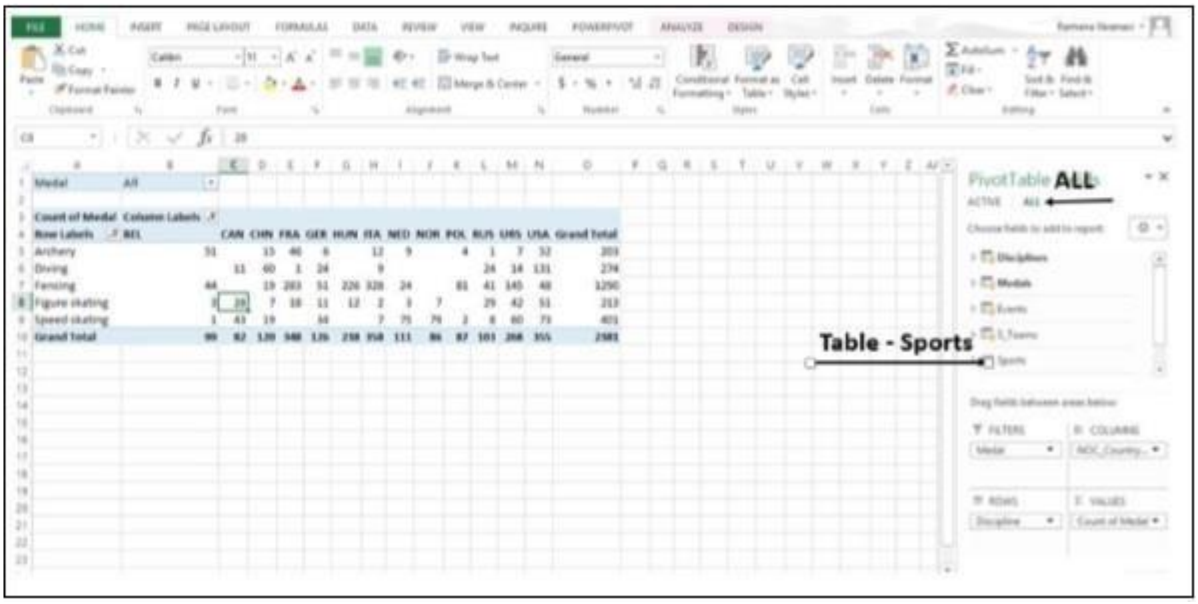
**Step 2** − Create a new table with new data. Name the new table as **Sports**.



| Sport | SportID |
|---|---|
| Aquatics | S1 |
| Archery | S2 |
| Athletics | S3 |
| Badminton | S4 |
| Baseball | S5 |
| Basketball | S6 |
| Basque Pelota | S7 |
| Biathlon | S8 |
| Bobsleigh | S9 |
| Boxing | S10 |
| Canoe / Kayak | S11 |
| Cricket | S12 |
| Croquet | S13 |
| Curling | S14 |
| Cycling | S15 |
| Equestrian | S16 |
| Fencing | S17 |
| Football | S18 |
| Golf | S19 |
| Gymnastics | S20 |
| Handball | S21 |
| Hockey | S22 |

**Step 3** − Now you can create relationship between this new table and the other tables that already exist in the **Data Model** in Excel. Rename the Sheet1 as **Medals** and Sheet2 as **Sports**.

On the **Medals sheet**, in the **PivotTable Fields List**, click **All**. A complete list of available tables will be displayed. The newly added table - **Sports** will also be displayed.



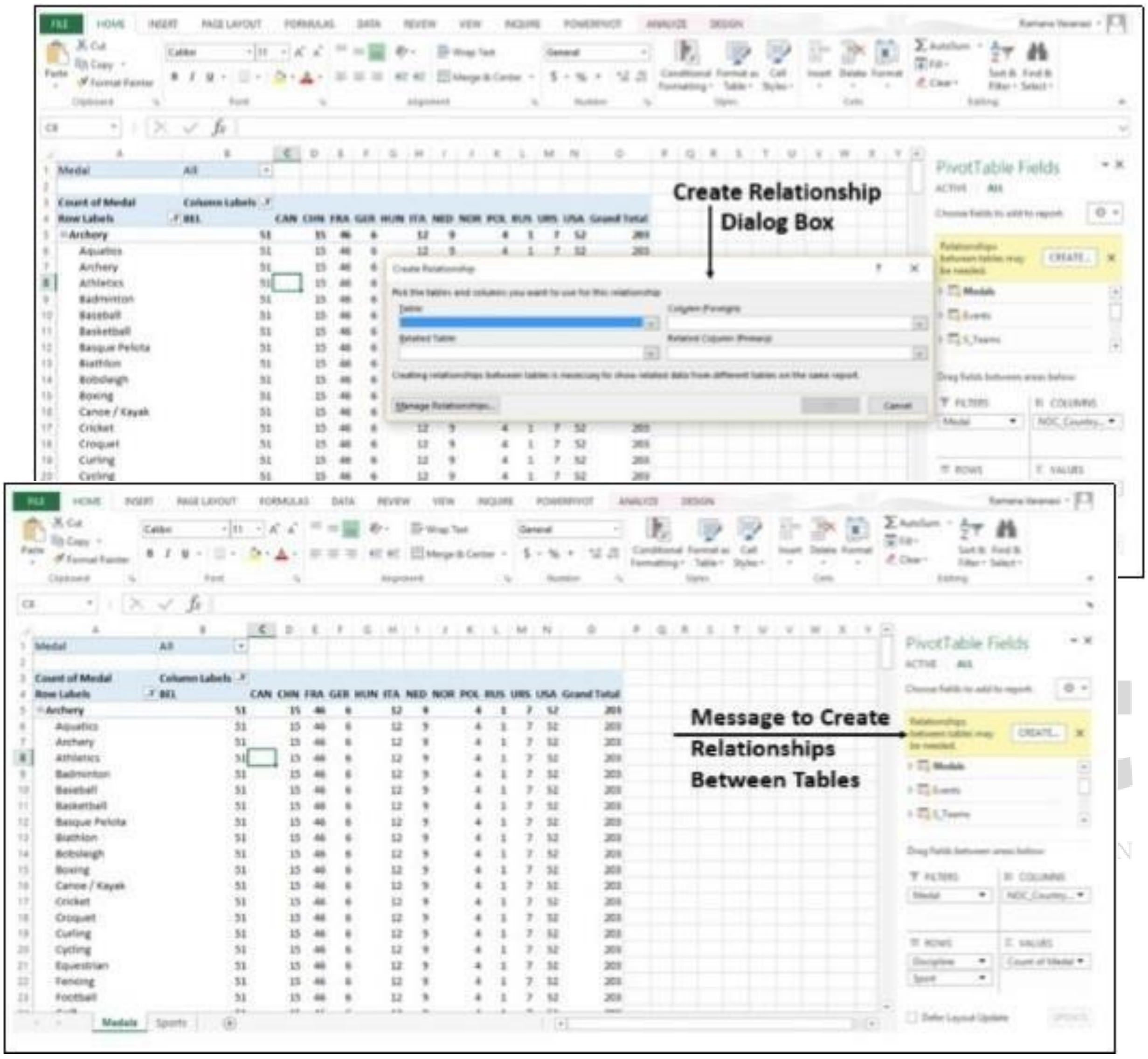
Medal: All

| Count of Medal | Column Labels | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Row Labels | BEL | CAN | CHN | FRA | GER | HUN | ITA | NED | NOR | POL | RUS | URS | USA | Grand Total |
| Archery | 51 | | 15 | 46 | 6 | | 12 | 9 | | 4 | 1 | 7 | 32 | 203 |
| Diving | | 11 | 60 | 1 | 24 | | 9 | | | | 24 | 14 | 131 | 274 |
| Fencing | 44 | | 19 | 283 | 51 | 226 | 328 | 24 | | 81 | 41 | 145 | 48 | 1290 |
| Figure skating | 3 | 28 | 7 | 18 | 11 | 12 | 2 | 3 | 7 | | 29 | 42 | 51 | 213 |
| Speed skating | 1 | 43 | 19 | | 44 | | 7 | 75 | 79 | 2 | 8 | 60 | 73 | 401 |
| Grand Total | 99 | 82 | 120 | 348 | 126 | 238 | 358 | 111 | 86 | 87 | 103 | 268 | 355 | 2583 |

FILTERS: Medal
COLUMNS: NOC_Country...
ROWS: Discipline
VALUES: Count of Medal

**Step 4** − Click on **Sports**. In the expanded list of fields, select **Sports**. Excel messages you to create a relationship between tables.

**Step 5** – Click on **CREATE**. The **Create Relationship** dialog box opens.



**Step 6** − To create the relationship, one of the tables must have a column of unique, non-repeated, values. In the **Disciplines** table, **SportID** column has such values. The table **Sports** that we have created also has the **SportID** column. In **Table**, select **Disciplines**.

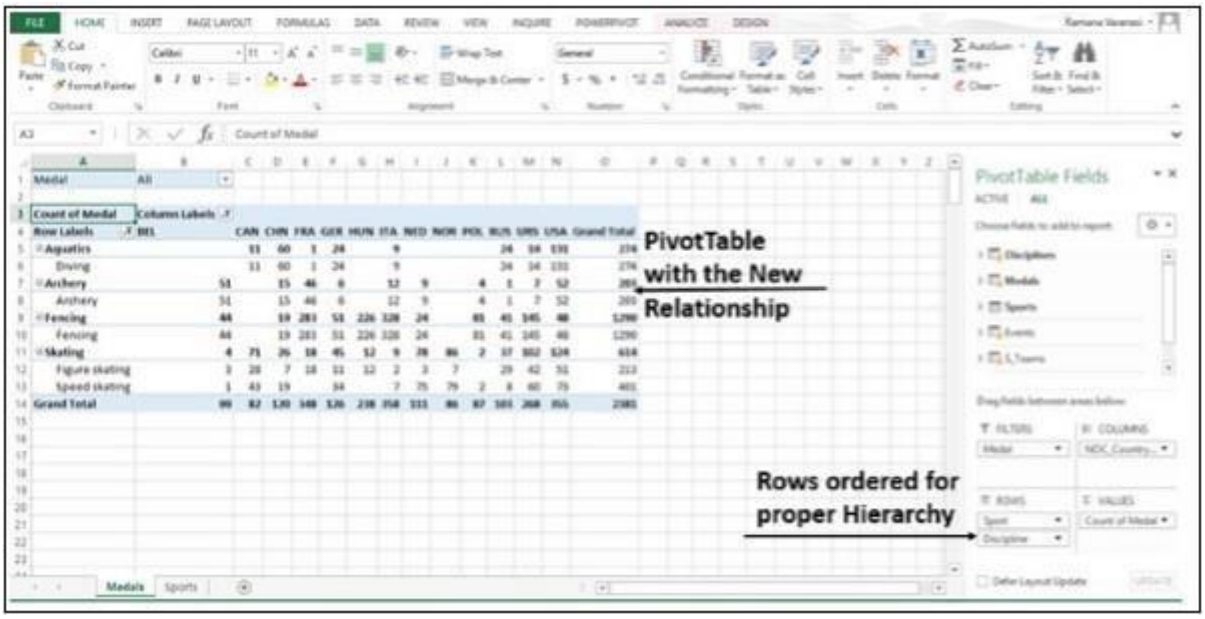
**Step 7** − In **Column (Foreign)**, select SportID.

**Step 8** − In **Related Table**, select **Sports**.

**Step 9** − In **Related Column (Primary)**, SportID gets selected automatically. Click **OK**.

**Step 10** − The **PivotTable** is modified to reflect the addition of the new **Data Field** Sport. Adjust the order of the fields in the Rows area to maintain the **Hierarchy**. In this case, **Sport**

should be first and **Discipline** should be the next, as **Discipline** will be nested in Sport as a sub-category.


PivotTable with the New Relationship
Rows ordered for proper Hierarchy
PivotTable Fields
FILTERS
COLUMNS
ROWS
VALUES

A3 Count of Medal

| Medal | All | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Count of Medal | Column Labels | | | | | | | | | | | | | |
| Row Labels | BEL | CAN | CHN | FRA | GER | HUN | ITA | NED | NOR | POL | RUS | URS | USA | Grand Total |
| **Aquatics** | | 11 | 60 | 1 | 24 | | 9 | | | | 24 | 34 | 131 | 274 |
| Diving | | 11 | 60 | 1 | 24 | | 9 | | | | 24 | 34 | 131 | 274 |
| **Archery** | 51 | | 15 | 46 | 6 | | 12 | 9 | | 4 | 1 | 7 | 52 | 203 |
| Archery | 51 | | 15 | 46 | 6 | | 12 | 9 | | 4 | 1 | 7 | 52 | 203 |
| **Fencing** | 44 | | 19 | 283 | 51 | 226 | 328 | 24 | | 81 | 41 | 145 | 48 | 1290 |
| Fencing | 44 | | 19 | 283 | 51 | 226 | 328 | 24 | | 81 | 41 | 145 | 48 | 1290 |
| **Skating** | 4 | 71 | 26 | 18 | 45 | 12 | 9 | 78 | 86 | 2 | 37 | 102 | 124 | 614 |
| Figure skating | 3 | 28 | 7 | 18 | 11 | 12 | 2 | 3 | 7 | | 29 | 42 | 51 | 213 |
| Speed skating | 1 | 43 | 19 | | 34 | | 7 | 75 | 79 | 2 | 8 | 60 | 73 | 401 |
| **Grand Total** | 99 | 82 | 120 | 348 | 126 | 238 | 358 | 111 | 86 | 87 | 101 | 268 | 355 | 2381 |

FILTERS: Medal
COLUMNS: NOC_CountryRegion
ROWS: Sport, Discipline
VALUES: Count of Medal

# Practical 11

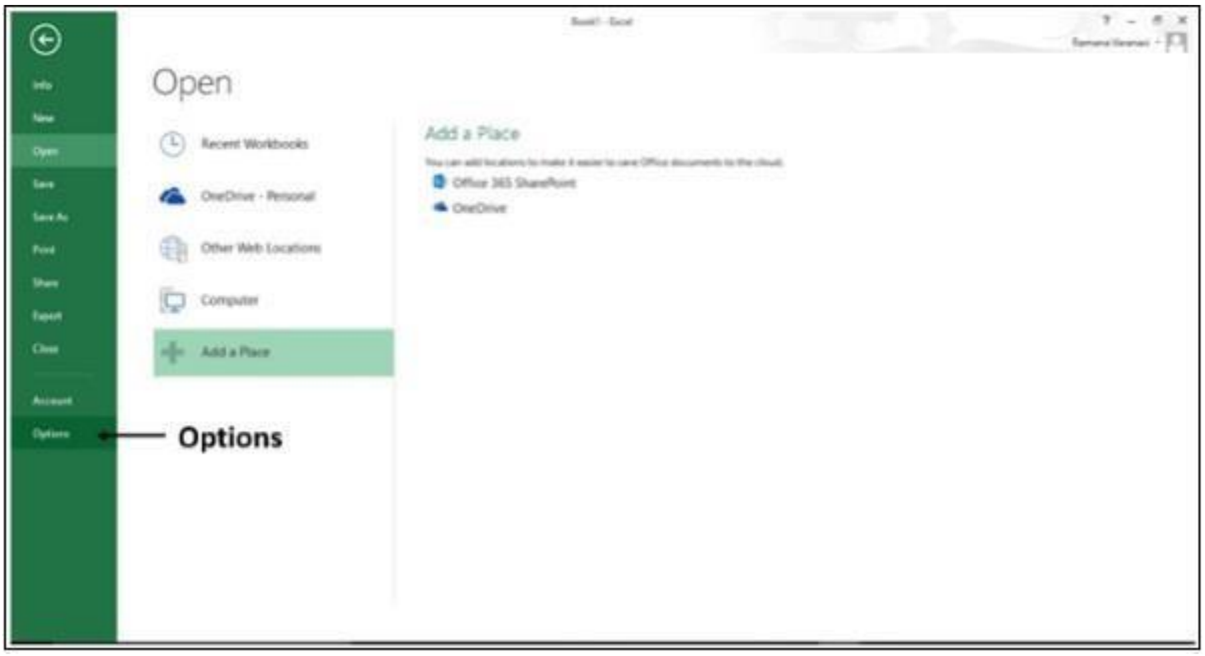
## Data Analysis and Visualization using Advanced Excel

Power View is a feature of Microsoft Excel 2013 that enables **interactive** data exploration, visualization, and presentation encouraging intuitive ad-hoc reporting.

### Create a Power View Sheet

Make sure **Power View** add-in is enabled in Excel 2013.

**Step 1** − Click on the **File** menu and then Click on **Options**.

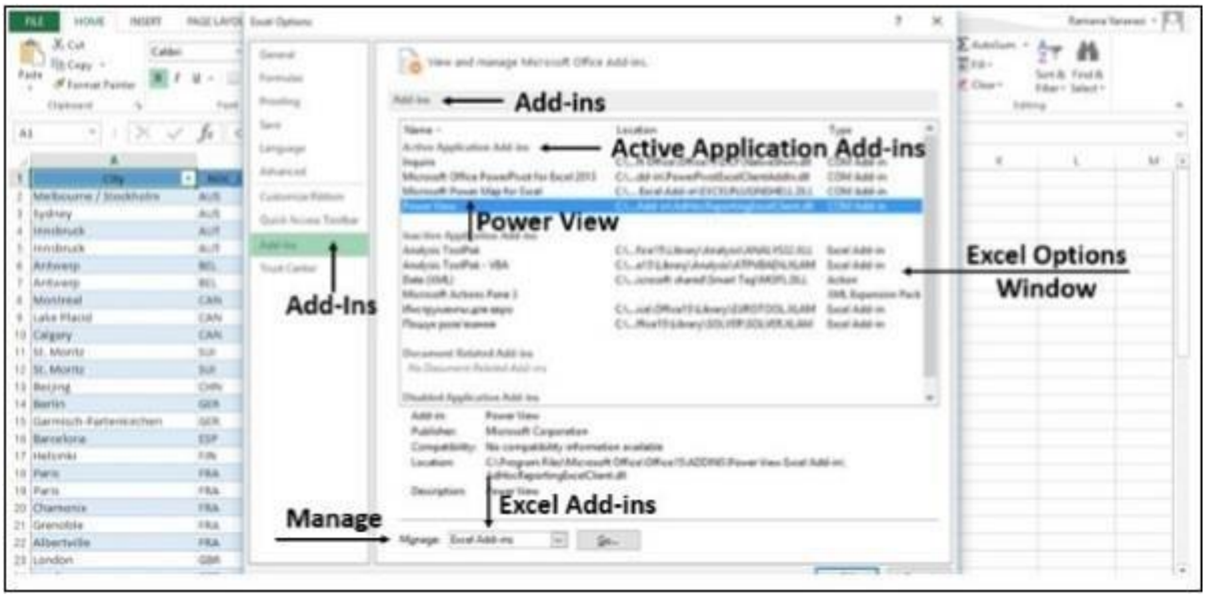


The **Excel Options** window appears.

**Step 2** − Click on **Add-Ins**.

**Step 3** − In the **Manage** box, click the drop-down arrow and select **Excel Add-ins**.

**Step 4** − All the available **Add-ins** will be displayed. If **Power View** Add-in is enabled, it appears in Active Application Add-ins.



If it does not appear, follow these steps –

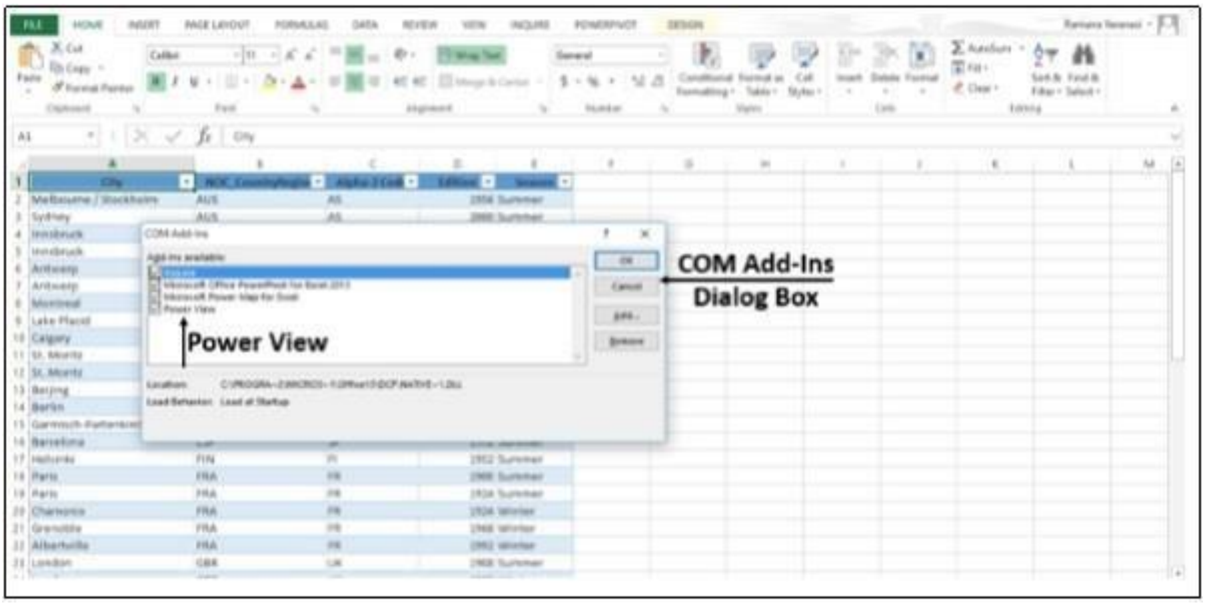
**Step 1** – In the **Excel Options** Window, Click on **Add-Ins**.

**Step 2** – In the **Manage** box, click the drop-down arrow and select **COM Add-ins**

**Step 3** – Click on the **Go** button. A **COM Add-Ins** Dialog Box appears.

**Step 4** – Check the **Power View** Check Box.

**Step 5** – Click **OK**.

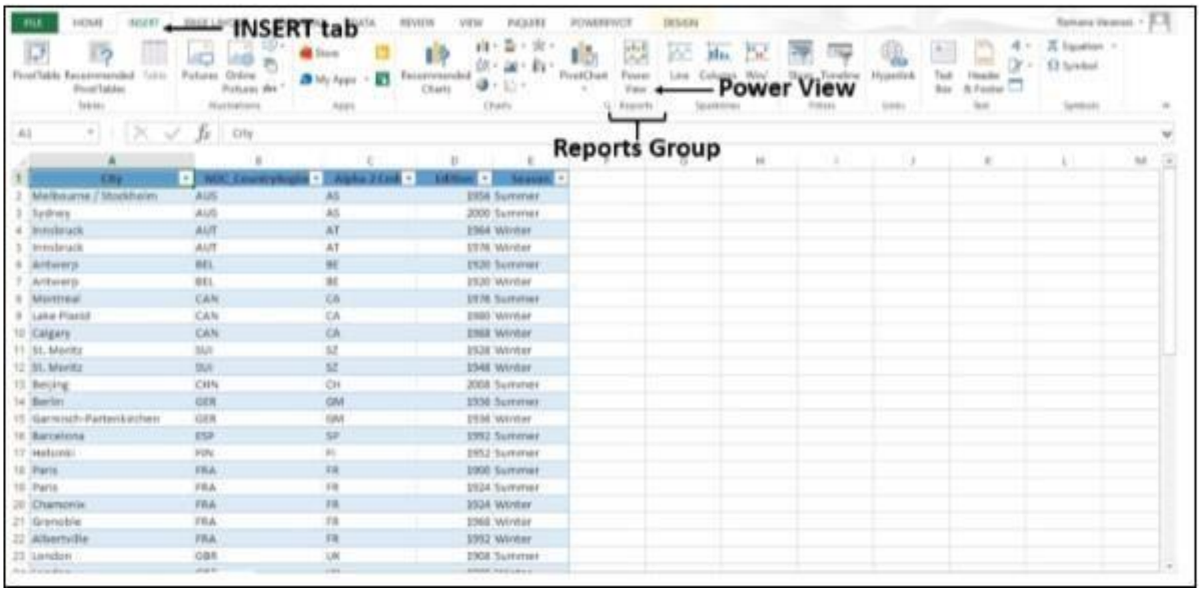


Now, you are ready to create the **Power View sheet**.

**Step 1** – Click on the **Data Table**.

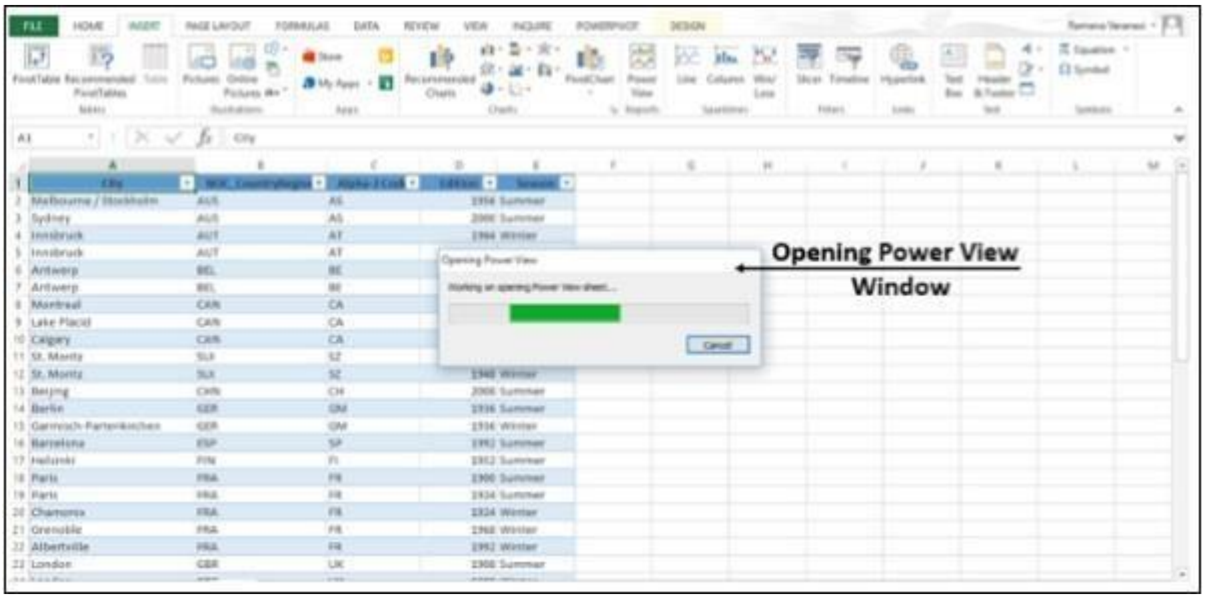
**Step 2** − Click on **Insert** tab.

**Step 3** − Click on **Power View** in **Reports** Group.



| | City | NOC_CountryRegion | Alpha 2 Code | Edition | Season |
|---|---|---|---|---|---|
| 2 | Melbourne / Stockholm | AUS | AS | 1956 | Summer |
| 3 | Sydney | AUS | AS | 2000 | Summer |
| 4 | Innsbruck | AUT | AT | 1964 | Winter |
| 5 | Innsbruck | AUT | AT | 1976 | Winter |
| 6 | Antwerp | BEL | BE | 1920 | Summer |
| 7 | Antwerp | BEL | BE | 1920 | Winter |
| 8 | Montreal | CAN | CA | 1976 | Summer |
| 9 | Lake Placid | CAN | CA | 1980 | Winter |
| 10 | Calgary | CAN | CA | 1988 | Winter |
| 11 | St. Moritz | SUI | SZ | 1928 | Winter |
| 12 | St. Moritz | SUI | SZ | 1948 | Winter |
| 13 | Beijing | CHN | CH | 2008 | Summer |
| 14 | Berlin | GER | GM | 1936 | Summer |
| 15 | Garmisch-Partenkirchen | GER | GM | 1936 | Winter |
| 16 | Barcelona | ESP | SP | 1992 | Summer |
| 17 | Helsinki | FIN | FI | 1952 | Summer |
| 18 | Paris | FRA | FR | 1900 | Summer |
| 19 | Paris | FRA | FR | 1924 | Summer |
| 20 | Chamonix | FRA | FR | 1924 | Winter |
| 21 | Grenoble | FRA | FR | 1968 | Winter |
| 22 | Albertville | FRA | FR | 1992 | Winter |
| 23 | London | GBR | UK | 1908 | Summer |

An **Opening Power View** window opens, showing the progress of Working on opening Power View sheet.

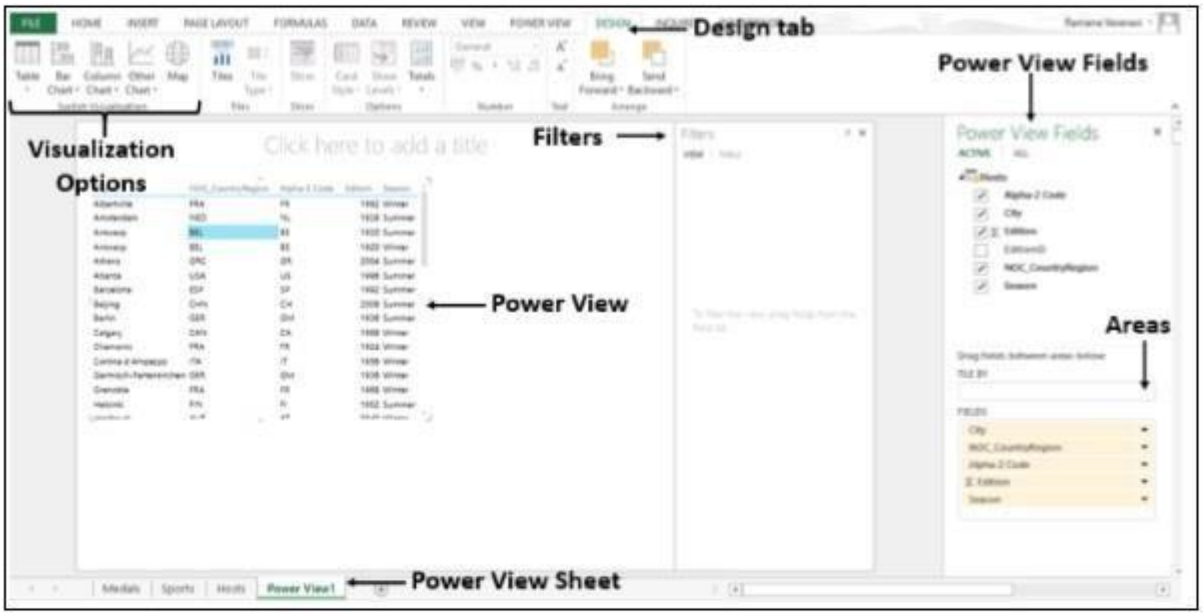


Opening Power View

Working on opening Power View sheet...

The **Power View sheet** is created for you and added to your Workbook with the **Power View**. On the Right-side of the **Power View**, you find the **Power View Fields**. Under the **Power View Fields** you will find **Areas**.

In the Ribbon, if you click on **Design tab**, you will find various **Visualization** options.



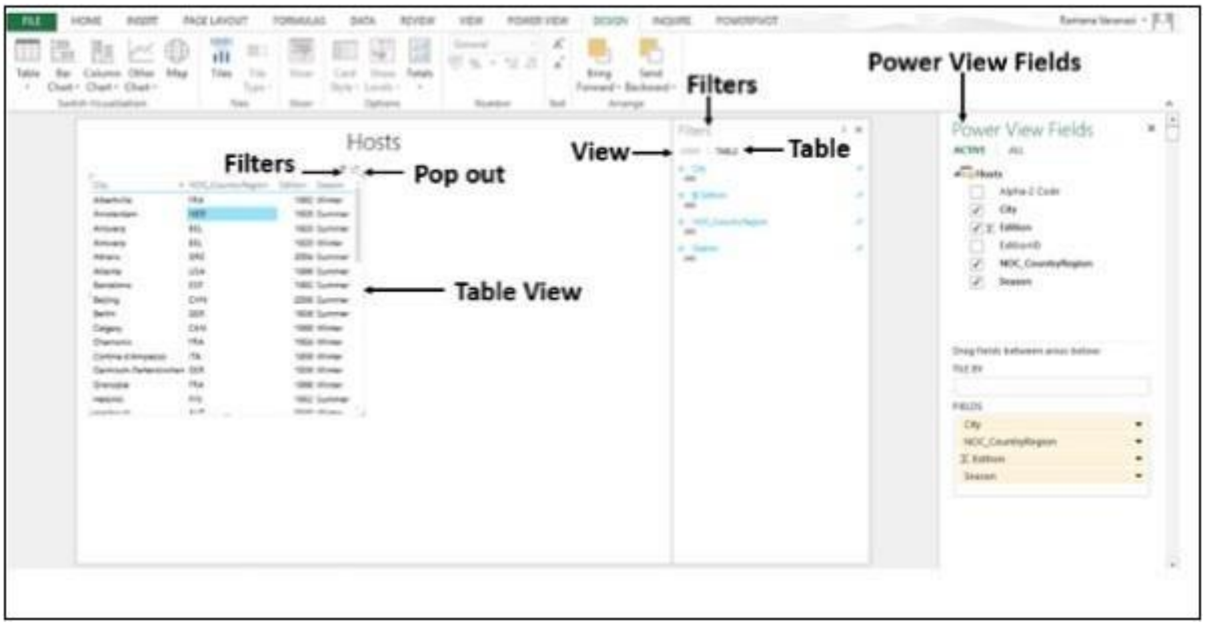
## Create Charts and other Visualizations

For every visualization you want to create, you start on a Power View sheet by creating a table, which you then easily convert to other visualizations, to find one that best illustrates your Data.

**Step 1** − Under the **Power View Fields**, select the fields you want to visualize.

**Step 2** − By default, the **Table** View will be displayed. As you move across the Table, on the top-right corner, you find two symbols – Filters and Pop out.

**Step 3** − Click on the **Filters** symbol. The filters will be displayed on the right side. **Filters** has two tabs. View tab to filter all visualizations in this **View** and **Table** tab to filter the specific values in this table only.



## Visualization – Matrix

A **Matrix** is made up of rows and columns like a **Table**. However, a Matrix has the following capabilities that a Table does not have −

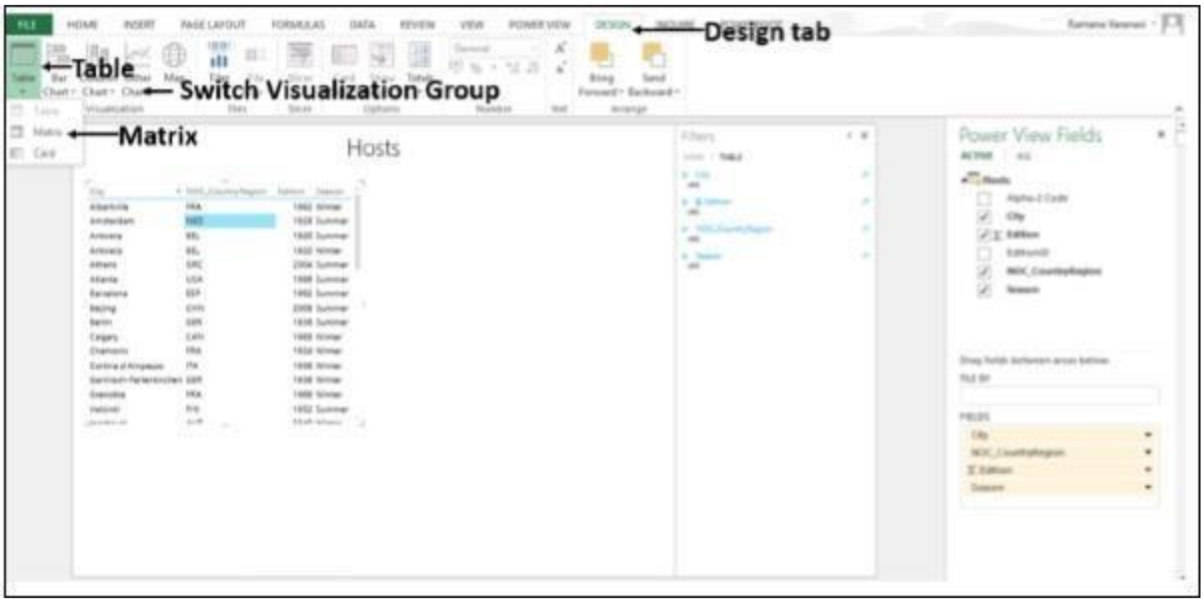
- Display data without repeating values.
- Display totals and subtotals by row and column.
- With a hierarchy, you can drill up/drill down.

## Collapse and Expand the Display

**Step 1** − Click on the **DESIGN** tab.

**Step 2** − Click on **Table** in the **Switch Visualization** Group.

**Step 3** − Click on **Matrix**.



The **Matrix Visualization** appear

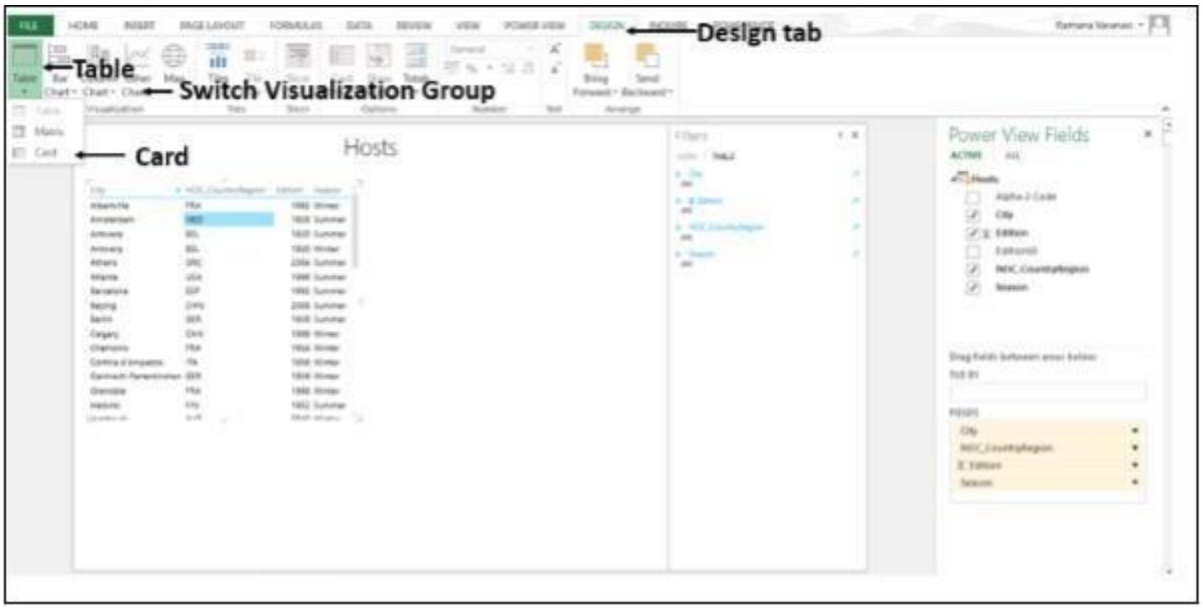
## Visualization – Card

You can convert a **Table** to a series of **Cards** that display the data from **each row** in the table laid out in a **Card format**, like an **index Card**.

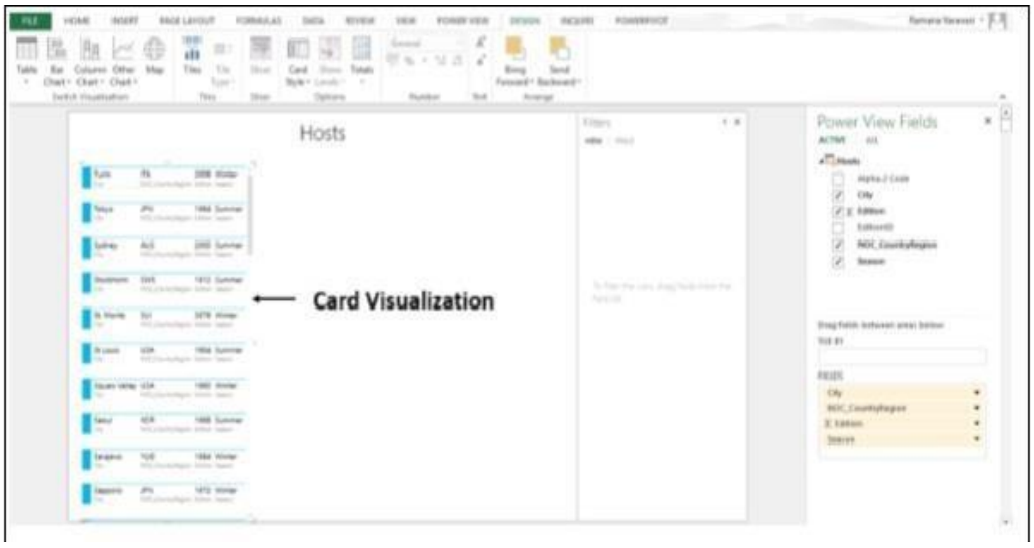
**Step 1** − Click on the **DESIGN** tab.

**Step 2** − Click on **Table** in the **Switch Visualization** Group.

**Step 3** − Click on **Card**.



The **Card Visualization** appears.



## Visualization – Charts

In **Power View**, you have a number of Chart options: Pie, Column, Bar, Line, Scatter, and Bubble. You can use several design options in a chart such as showing and hiding labels, legends, and titles.

Charts are interactive. If you click on a Value in one Chart −

- the **Value** in that chart is highlighted.
- All the Tables, Matrices, and Tiles in the report are filtered to that Value.
- That **Value** in all the other Charts in the report is highlighted.

The charts are interactive in a presentation setting also.

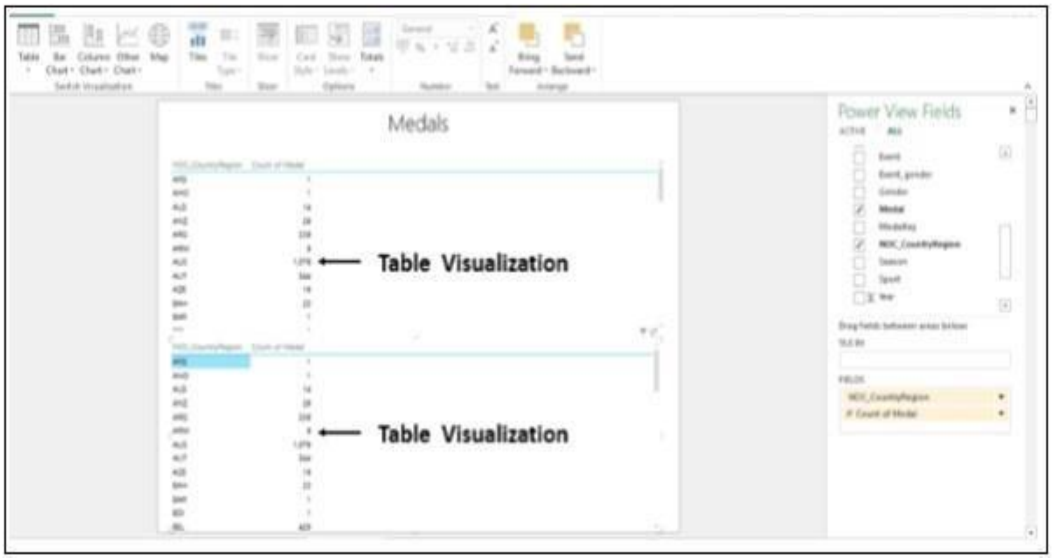
**Step 1** − Create a **Table Visualization** from **Medals** data.

You can use Line, Bar and Column Charts for comparing data points in one or more data series. In these Charts, the x-axis displays one field and the y-axis displays another, making it easy to see the relationship between the two values for all the items in the Chart.

Line Charts distribute category data evenly along a horizontal (category) axis, and all numerical value data along a vertical (value) axis.

**Step 2** − Create a Table Visualization for two Columns, **NOC_CountryRegion** and **Count of Medal**.

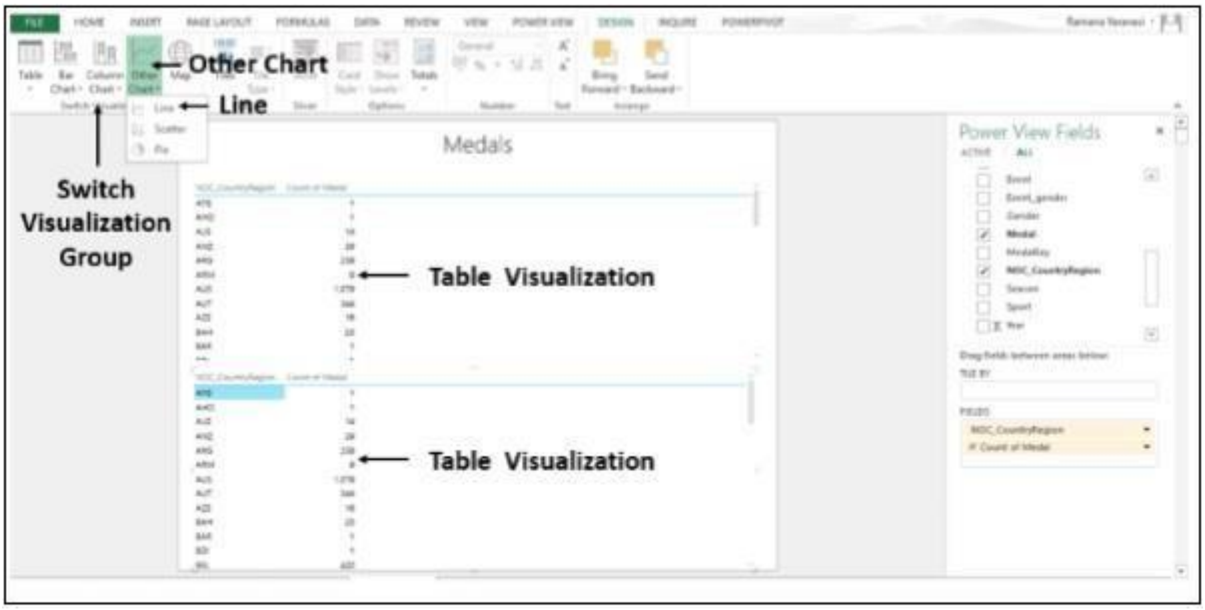
**Step 3** − Create the same Table Visualization below.



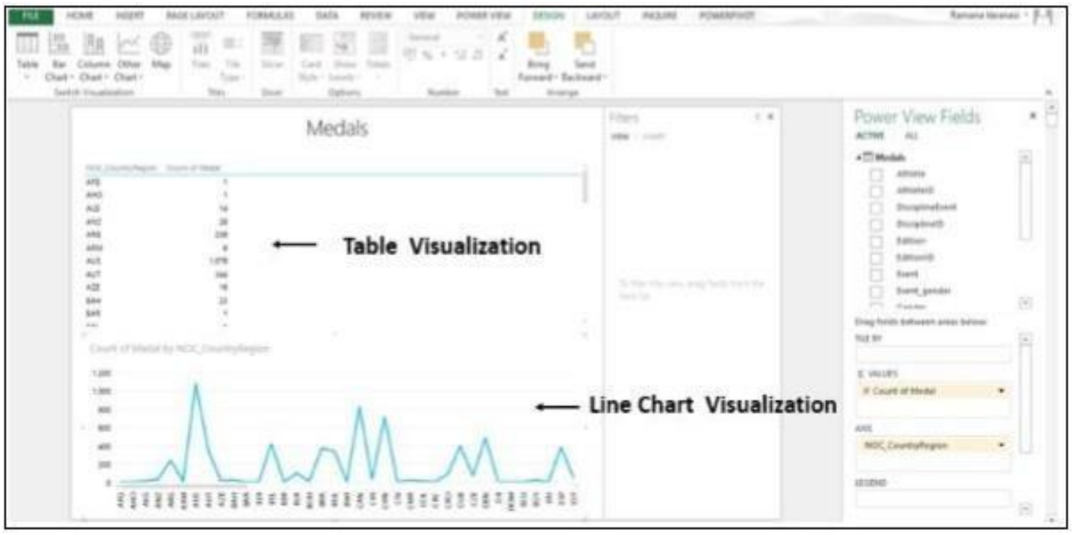
**Step 4** − Click on the **Table Visualization** below.

**Step 5** − Click on **Other Chart** in the **Switch Visualization** group.

**Step 6** − Click on **Line**.



The **Table Visualization** converts into **Line Chart Visualization**.

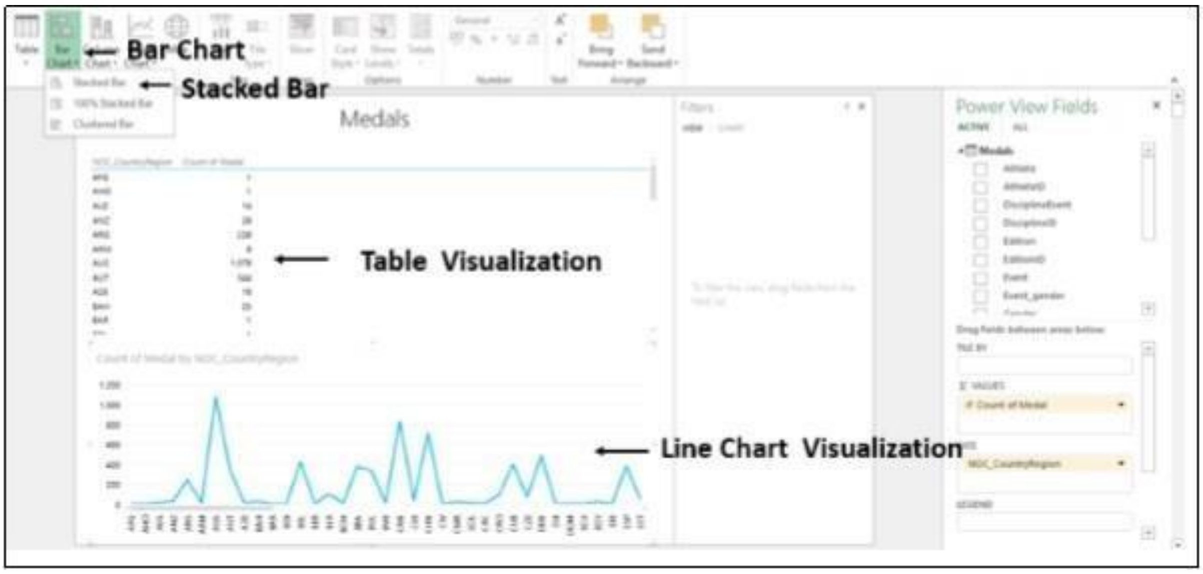


In a **Bar Chart**, categories are organized along the vertical axis and values along the horizontal axis. In **Power View**, there are three subtypes of the **Bar Chart: Stacked, 100% stacked**, and **Clustered**.

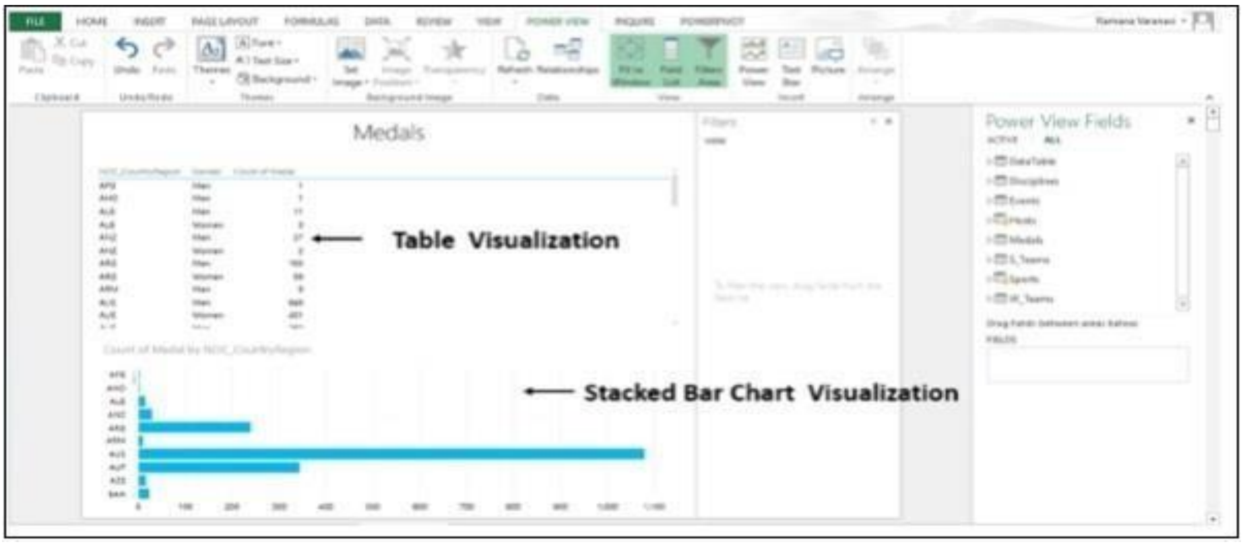
**Step 7** − Click on the **Line Chart Visualization**.

**Step 8** − Click on **Bar Chart** in the **Switch Visualization** Group.

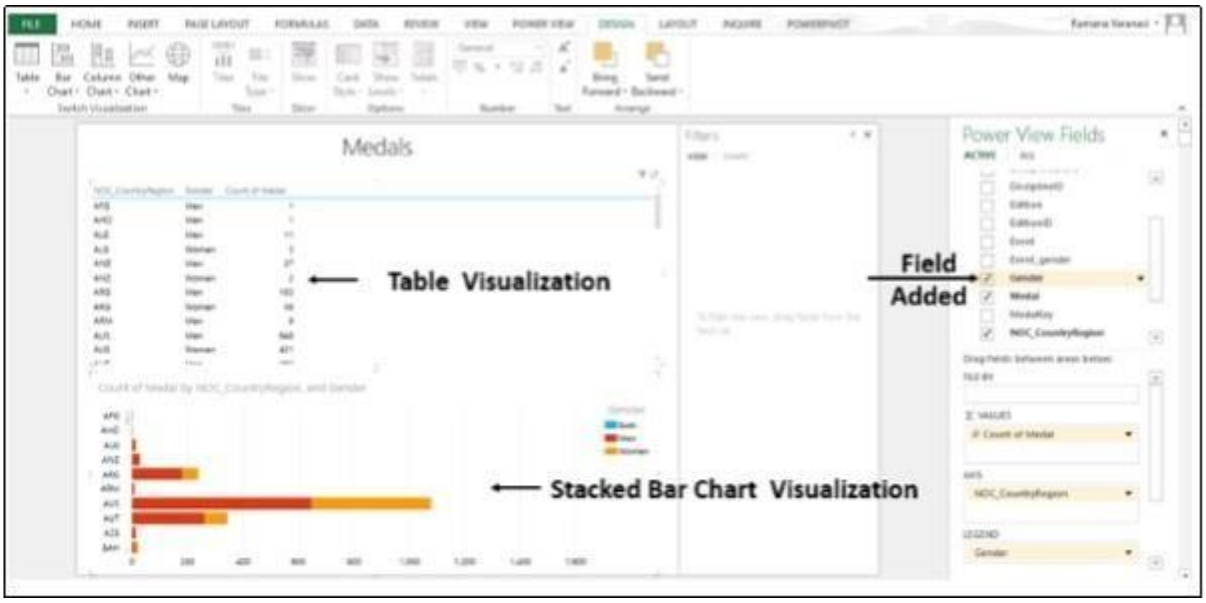
**Step 9** − Click on the **Stacked Bar** option.



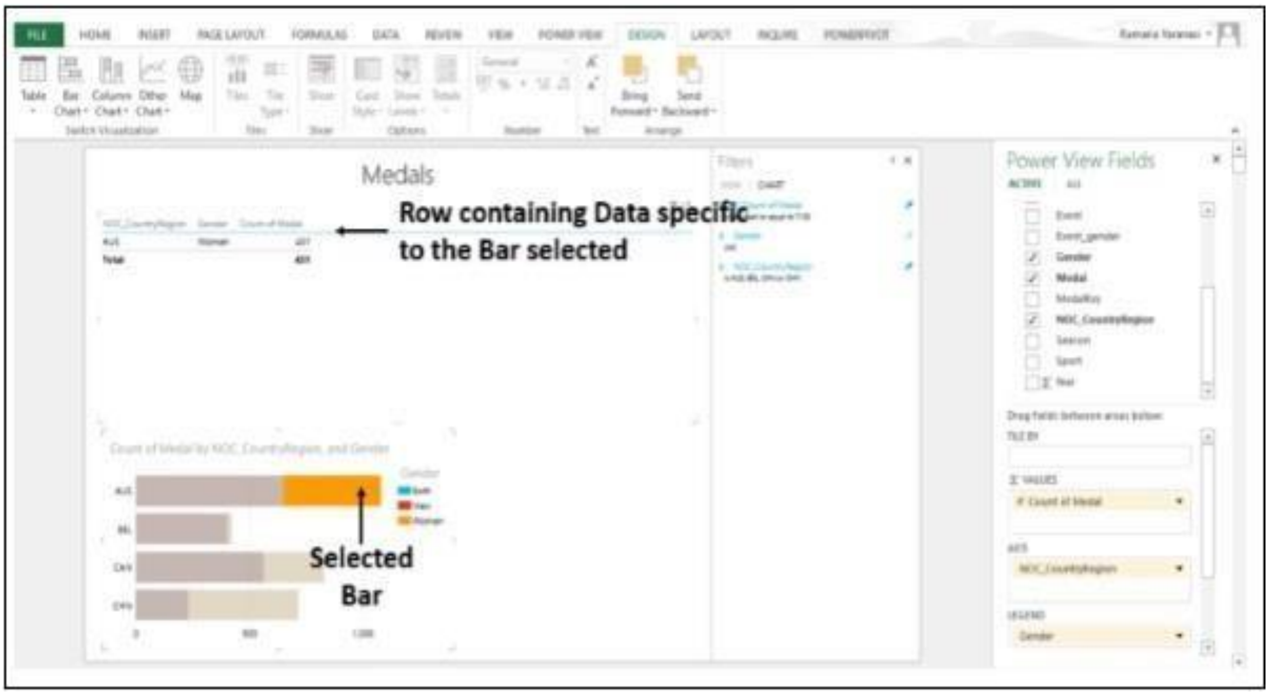
The **Line Chart Visualization** converts into **Stacked Bar Chart Visualization**.



**Step 10** – In the **Power View Fields**, in the **Medals** Table, select the **Field Gender** also.



**Step 11** – Click on one of the bars. That portion of the bar is highlighted. Only the row containing the Data specific to the selected bar is displayed in the table above.



You can use the column charts for showing data changes over a period of time or for illustrating comparison among different items. In a Column Chart, the categories are along the horizontal axis and values are along the vertical axis.

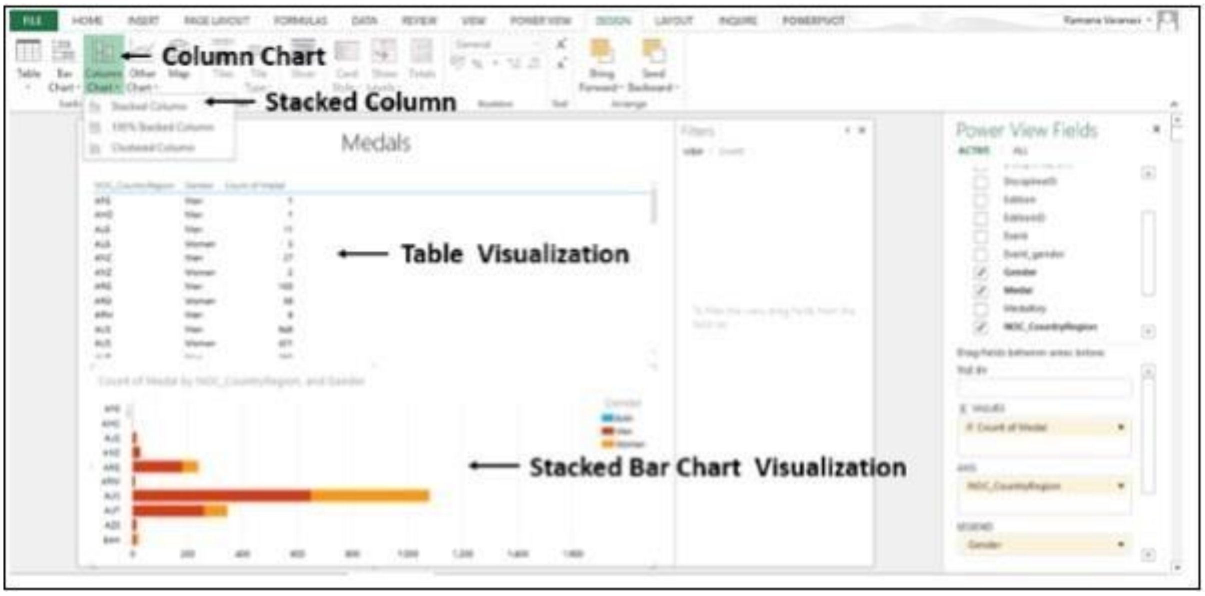
In Power View, there are three **Column Chart** subtypes: **Stacked, 100% stacked**, and

**Clustered**.

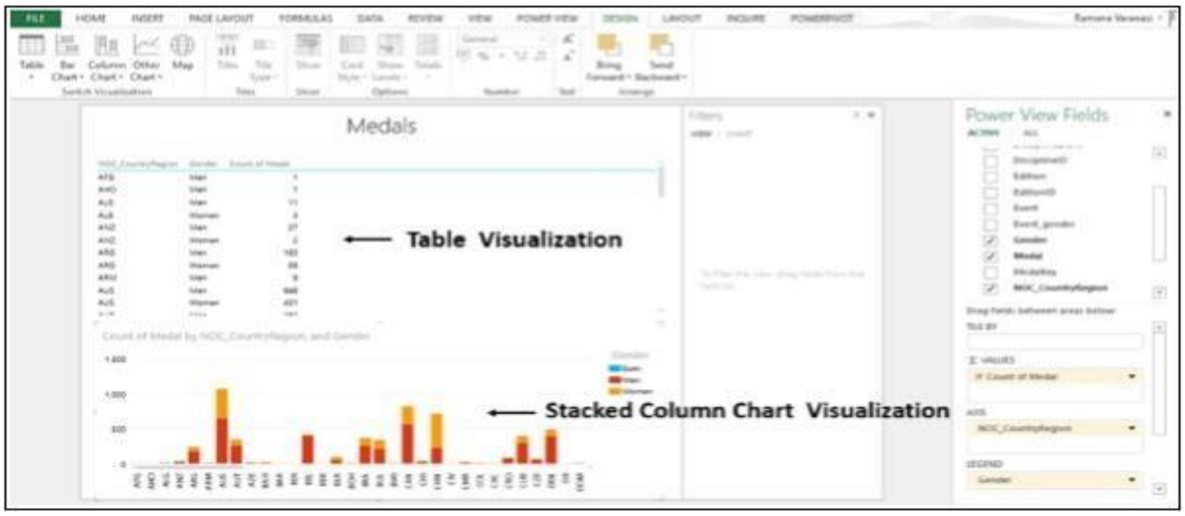
**Step 12** − Click on the **Stacked Bar Chart Visualization**.

**Step 13** − Click on Column Chart in the **Switch Visualization** group.

**Step 14** − Click on **Stacked Column**.



The **Stacked Bar Chart** Visualization converts into **Stacked Column Chart** Visualization.

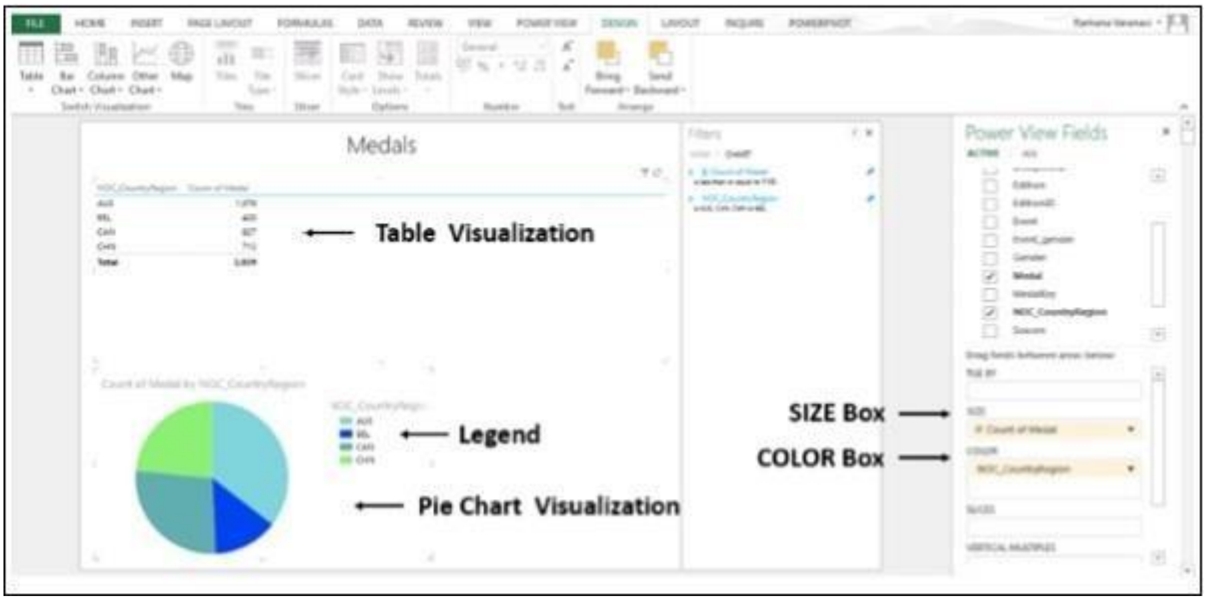


You can have simple **Pie Chart Visualizations** in Power View.

**Step 1** − Click on the **Table Visualization** as shown below.

**Step 2** − Click on **Other Chart** in the **Switch Visualization** group.

**Step 3** − Click on **Pie** as shown in the image given below.


Medals
Table Visualization
Legend
Pie Chart Visualization
SIZE Box
COLOR Box
Power View Fields
Filters