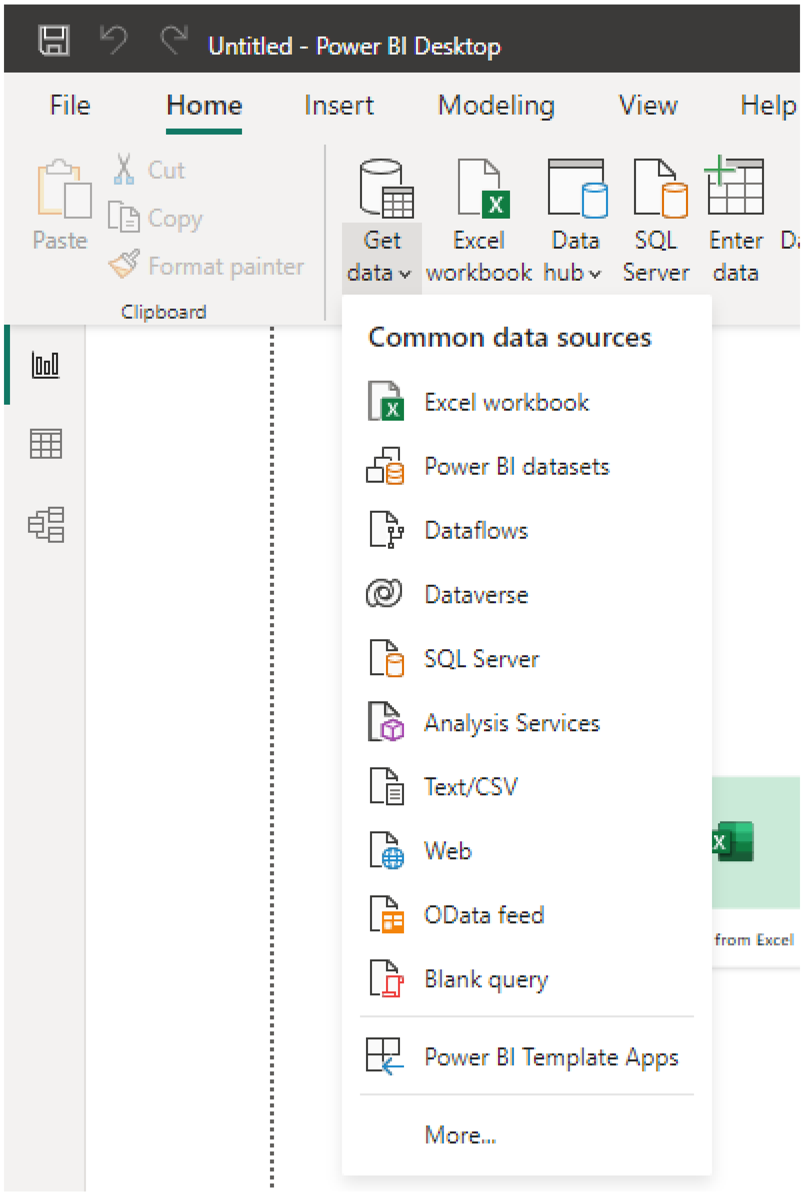
Practical 1: Import the legacy data from different sources such as (Excel, SqlServer, Oracle etc.) and load in the target system.

Importing Excel Data

1) Launch Power BI Desktop.

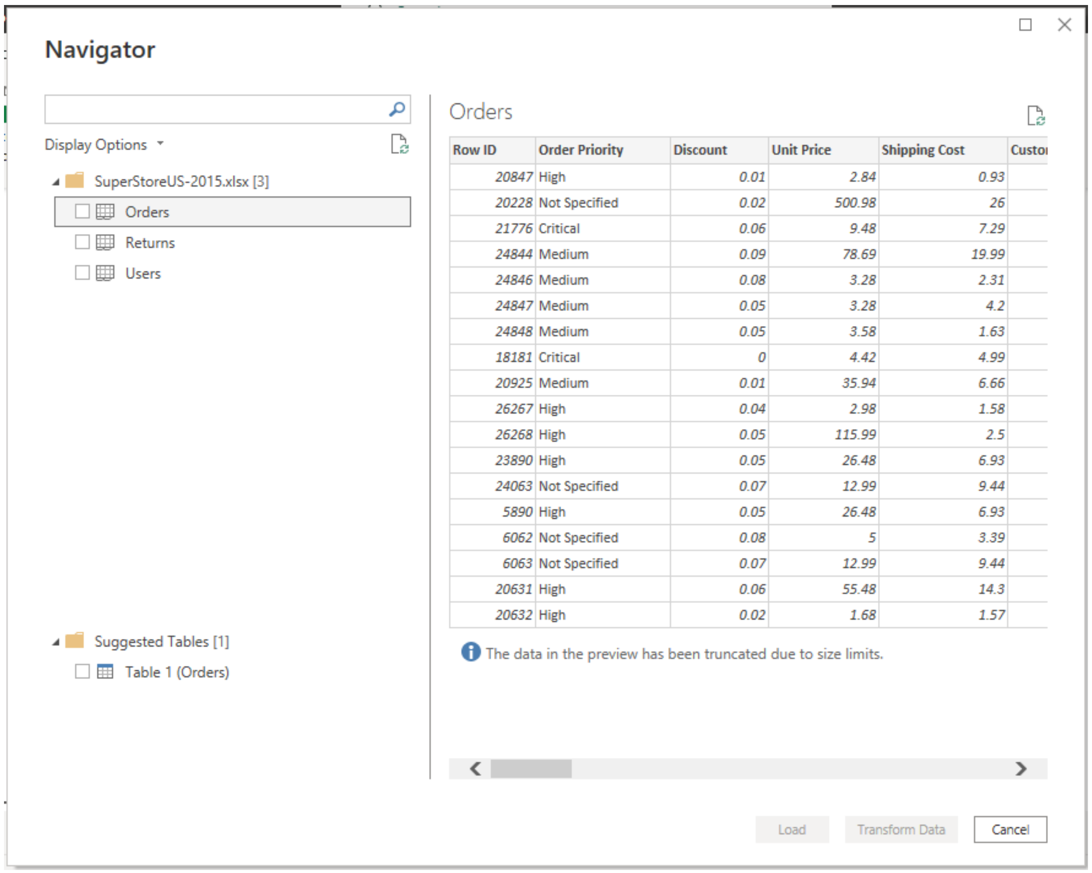
2) From the Home ribbon, select Get Data. Excel is one of the Most Common data connections, so you can select it directly from the Get Data menu



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.



Navigator

Display Options

- SuperStoreUS-2015.xlsx [3]
  - Orders
  - Returns
  - Users
- Suggested Tables [1]
  - Table 1 (Orders)

Orders

Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Custo
20847	High	0.01	2.84	0.93	
20228	Not Specified	0.02	500.98	26	
21776	Critical	0.06	9.48	7.29	
24844	Medium	0.09	78.69	19.99	
24846	Medium	0.08	3.28	2.31	
24847	Medium	0.05	3.28	4.2	
24848	Medium	0.05	3.58	1.63	
18181	Critical	0	4.42	4.99	
20925	Medium	0.01	35.94	6.66	
26267	High	0.04	2.98	1.58	
26268	High	0.05	115.99	2.5	
23890	High	0.05	26.48	6.93	
24063	Not Specified	0.07	12.99	9.44	
5890	High	0.05	26.48	6.93	
6062	Not Specified	0.08	5	3.39	
6063	Not Specified	0.07	12.99	9.44	
20631	High	0.06	55.48	14.3	
20632	High	0.02	1.68	1.57	

The data in the preview has been truncated due to size limits.

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:

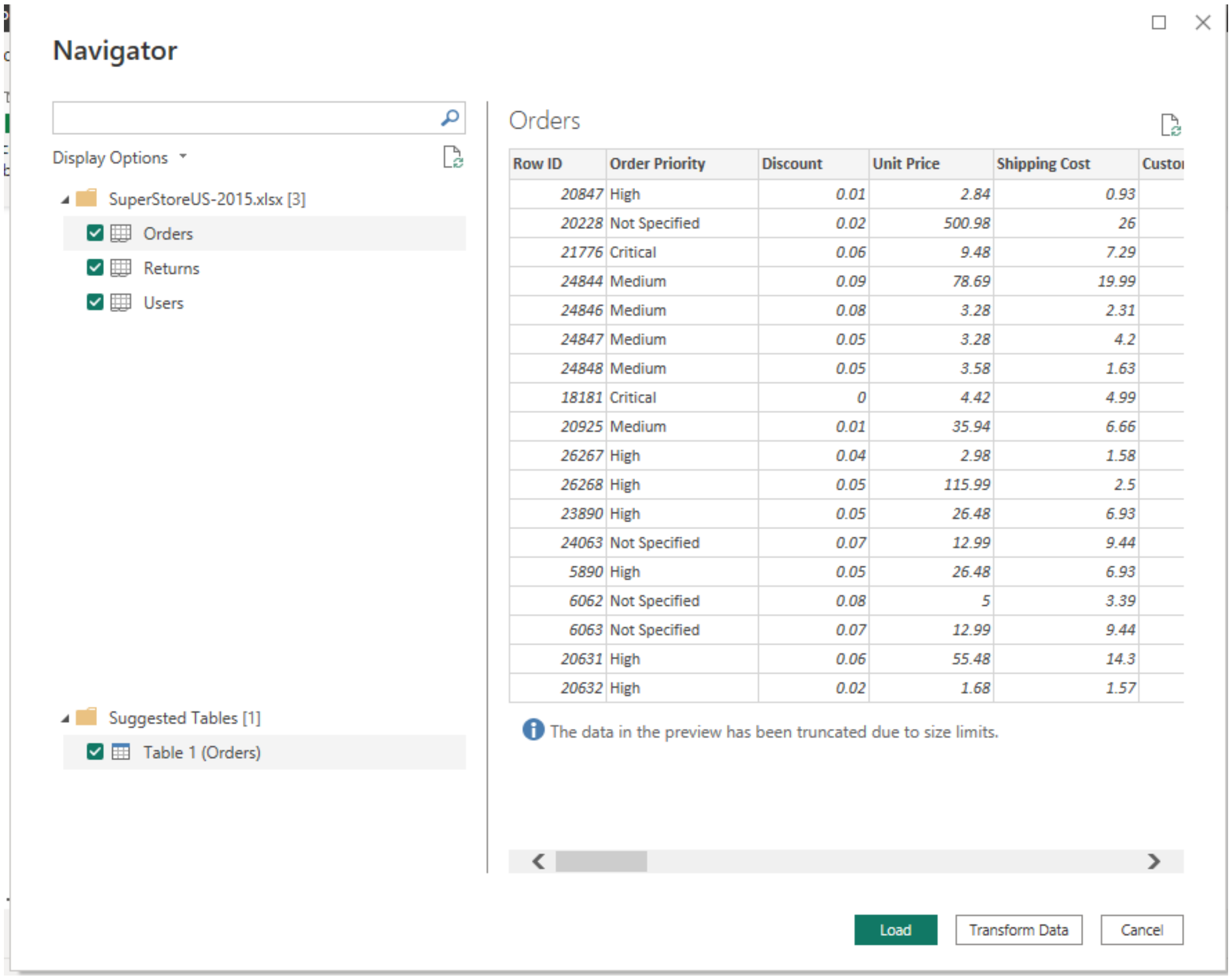
1) From the Home ribbon tab in Query Editor, select Get Data.

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



Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Custo
20847	High	0.01	2.84	0.93	
20228	Not Specified	0.02	500.98	26	
21776	Critical	0.06	9.48	7.29	
24844	Medium	0.09	78.69	19.99	
24846	Medium	0.08	3.28	2.31	
24847	Medium	0.05	3.28	4.2	
24848	Medium	0.05	3.58	1.63	
18181	Critical	0	4.42	4.99	
20925	Medium	0.01	35.94	6.66	
26267	High	0.04	2.98	1.58	
26268	High	0.05	115.99	2.5	
23890	High	0.05	26.48	6.93	
24063	Not Specified	0.07	12.99	9.44	
5890	High	0.05	26.48	6.93	
6062	Not Specified	0.08	5	3.39	
6063	Not Specified	0.07	12.99	9.44	
20631	High	0.06	55.48	14.3	
20632	High	0.02	1.68	1.57	

The data in the preview has been truncated due to size limits.

Practical 2: Perform the Extraction Transformation and Loading (ETL) Power BI.

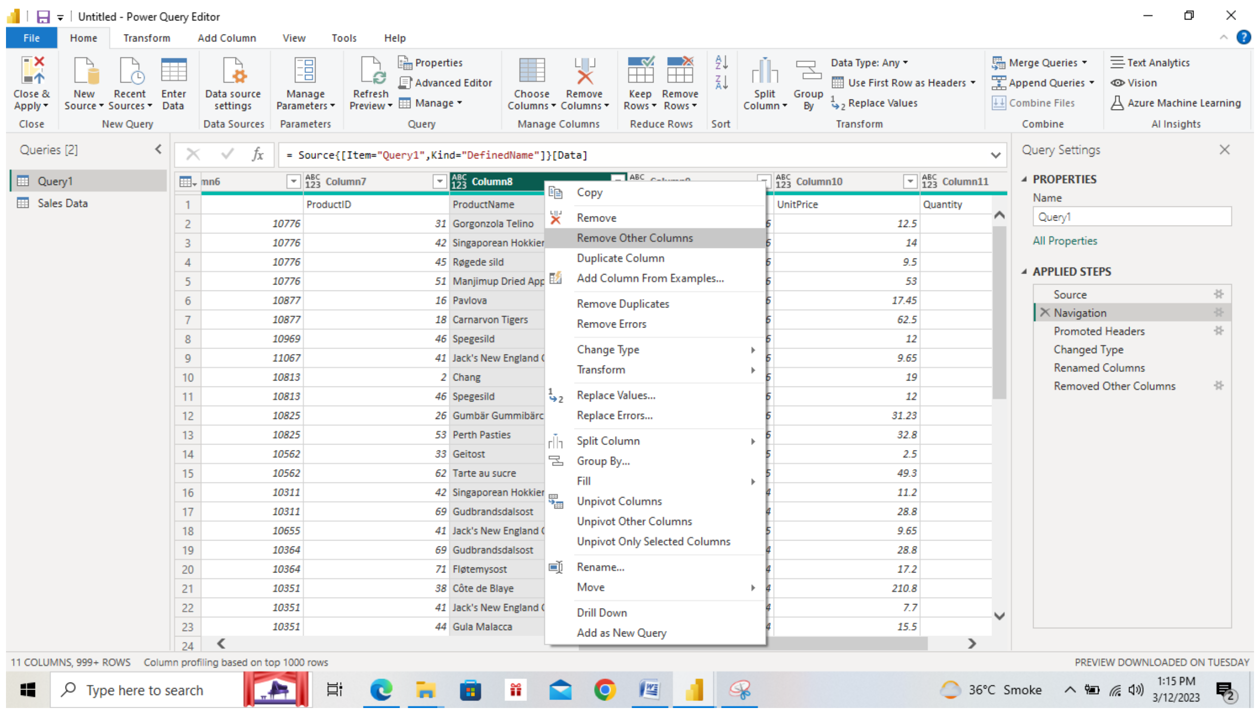
ETL Process in Power BI

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.



= Source{[Item="Query1",Kind="DefinedName"]}[Data]

	mn6	Column7	Column8	Column10	Column11
1		ProductID	ProductName	UnitPrice	Quantity
2	10776	31	Gorgonzola Telino	12.5	
3	10776	42	Singaporean Hokkier	14	
4	10776	45	Røgede sild	9.5	
5	10776	51	Manjimup Dried App	53	
6	10877	16	Pavlova	17.45	
7	10877	18	Carnarvon Tigers	62.5	
8	10969	46	Spegesild	12	
9	11067	41	Jack's New England (	9.65	
10	10813	2	Chang	19	
11	10813	46	Spegesild	12	
12	10825	26	Gumbär Gummibärc	31.23	
13	10825	53	Perth Pasties	32.8	
14	10562	33	Geitost	2.5	
15	10562	62	Tarte au sucre	49.3	
16	10311	42	Singaporean Hokkier	11.2	
17	10311	69	Gudbrandsdalsost	28.8	
18	10655	41	Jack's New England (	9.65	
19	10364	69	Gudbrandsdalsost	28.8	
20	10364	71	Fløtemysost	17.2	
21	10351	38	Côte de Blaye	210.8	
22	10351	41	Jack's New England (	7.7	
23	10351	44	Gula Malacca	15.5	
24					

11 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED ON TUESDAY

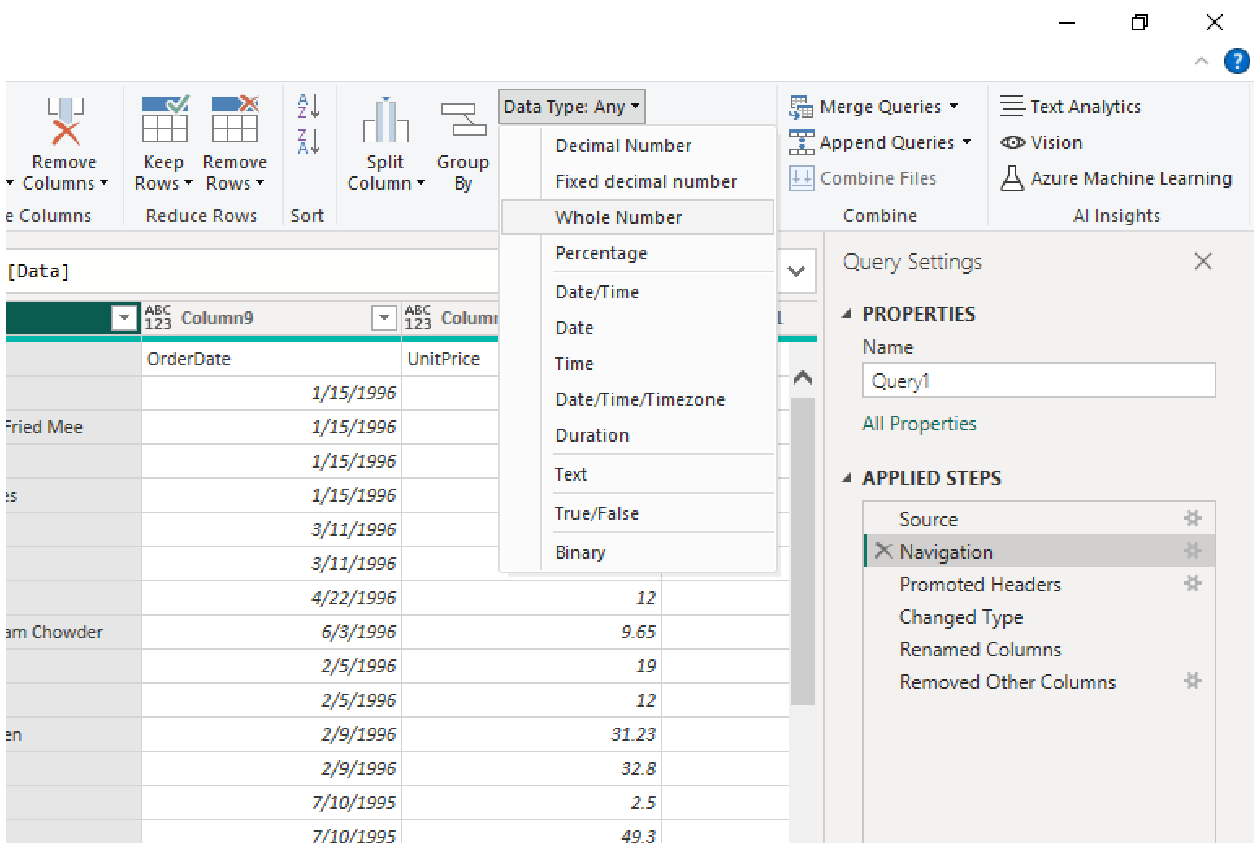
3. Change the data type of the UnitsInStock column When Query Editor connects to data, it reviews each field and to determine the best data type.

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).



	ABC 123 Column9	ABC 123 Columr
	OrderDate	UnitPrice
	1/15/1996	
Fried Mee	1/15/1996	
	1/15/1996	
es	1/15/1996	
	3/11/1996	
	3/11/1996	
	4/22/1996	12
am Chowder	6/3/1996	9.65
	2/5/1996	19
	2/5/1996	12
en	2/9/1996	31.23
	2/9/1996	32.8
	7/10/1995	2.5
	7/10/1995	49.3

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.

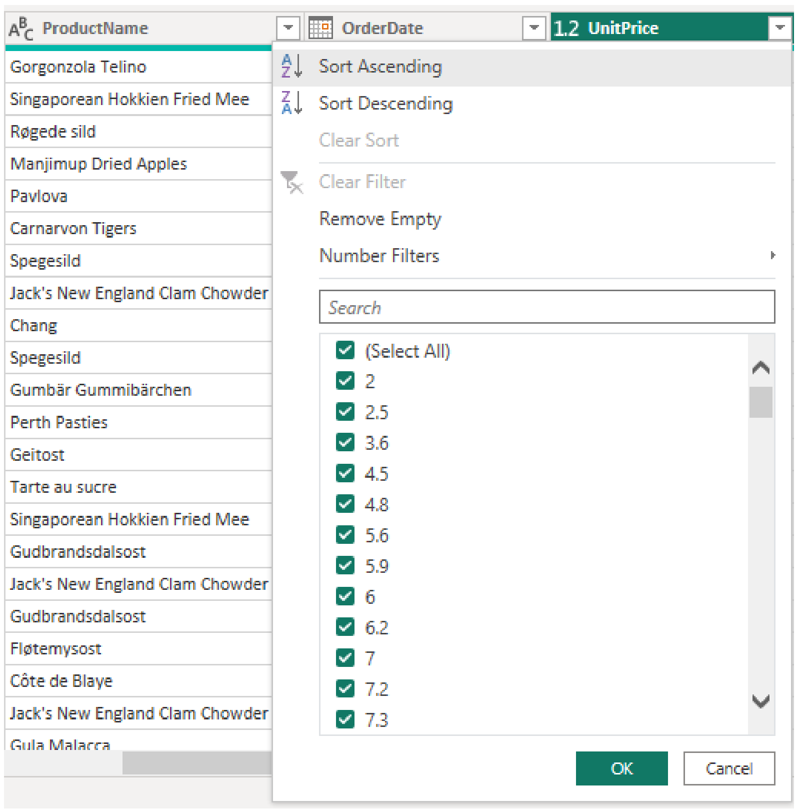
1. In the Query View, scroll to the Order_Details column.

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.

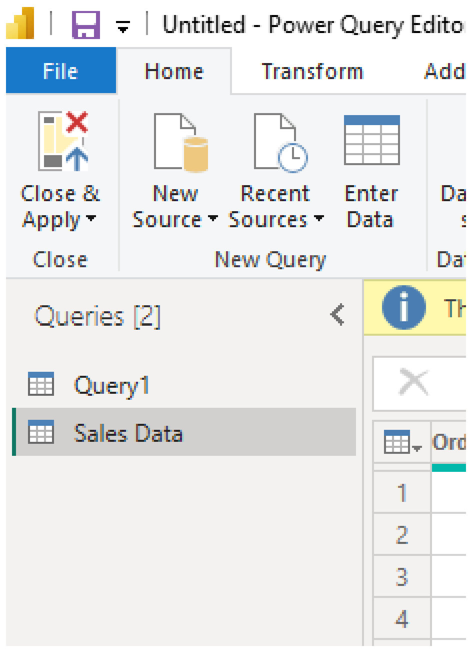


ProductName	OrderDate	1.2 UnitPrice
Gorgonzola Telino		
Singaporean Hokkien Fried Mee		
Røgede sild		
Manjimup Dried Apples		
Pavlova		
Carnarvon Tigers		
Spegesild		
Jack's New England Clam Chowder		
Chang		
Spegesild		
Gumbär Gummibärchen		
Perth Pasties		
Geitost		
Tarte au sucre		
Singaporean Hokkien Fried Mee		
Gudbrandsdalsost		
Jack's New England Clam Chowder		
Gudbrandsdalsost		
Fløtemysost		
Côte de Blaye		
Jack's New England Clam Chowder		
Gula Malacca		

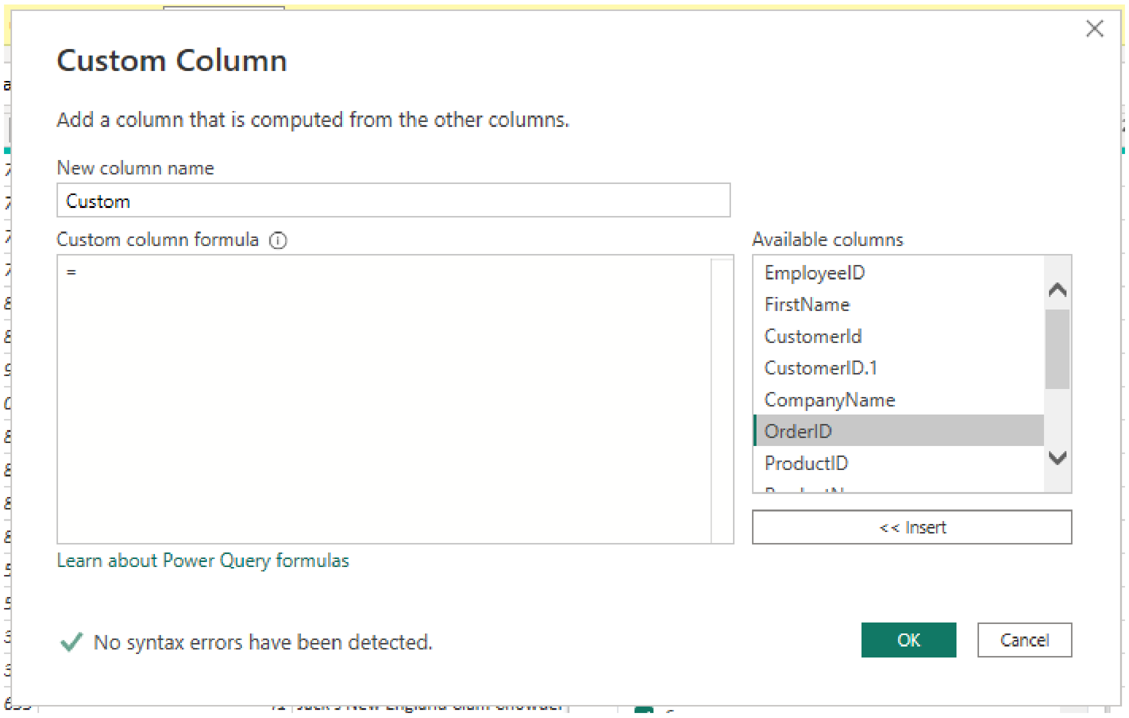
4. Calculate the line total for each Order_Details row Power BI Desktop lets you to create calculations based on the columns you are importing, so you can enrich the data that you connect to.

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.



Custom Column

Add a column that is computed from the other columns.

New column name: Custom

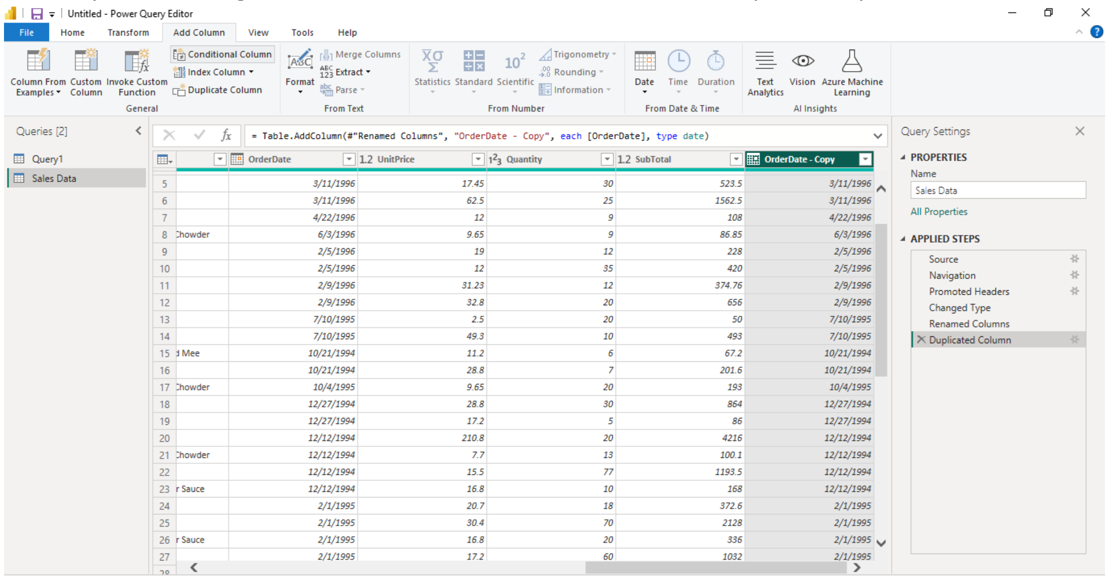
Custom column formula: =

Available columns: EmployeeID, FirstName, CustomerId, CustomerID.1, CompanyName, OrderID, ProductID

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 Line Total column to the left, after Ship Country.



```
= Table.AddColumn(#"Renamed Columns", "OrderDate - Copy", each [OrderDate], type date)
```

		OrderDate	UnitPrice	Quantity	SubTotal	OrderDate - Copy
5		3/11/1996	17.45	30	523.5	3/11/1996
6		3/11/1996	62.5	25	1562.5	3/11/1996
7		4/22/1996	12	9	108	4/22/1996
8	Chowder	6/3/1996	9.65	9	86.85	6/3/1996
9		2/5/1996	19	12	228	2/5/1996
10		2/5/1996	12	35	420	2/5/1996
11		2/9/1996	31.23	12	374.76	2/9/1996
12		2/9/1996	32.8	20	656	2/9/1996
13		7/10/1995	2.5	20	50	7/10/1995
14		7/10/1995	49.3	10	493	7/10/1995
15	d Mee	10/21/1994	11.2	6	67.2	10/21/1994
16		10/21/1994	28.8	7	201.6	10/21/1994
17	Chowder	10/4/1995	9.65	20	193	10/4/1995
18		12/27/1994	28.8	30	864	12/27/1994
19		12/27/1994	17.2	5	86	12/27/1994
20		12/12/1994	210.8	20	4216	12/12/1994
21	Chowder	12/12/1994	7.7	13	100.1	12/12/1994
22		12/12/1994	15.5	77	1193.5	12/12/1994
23	r Sauce	12/12/1994	16.8	10	168	12/12/1994
24		2/1/1995	20.7	18	372.6	2/1/1995
25		2/1/1995	30.4	70	2128	2/1/1995
26	r Sauce	2/1/1995	16.8	20	336	2/1/1995
27		2/1/1995	17.2	60	1032	2/1/1995

PROPERTIES

Name: Sales Data

All Properties

APPLIED STEPS
- Source
- Navigation
- Promoted Headers
- Changed Type
- Renamed Columns
- Duplicated Column

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.

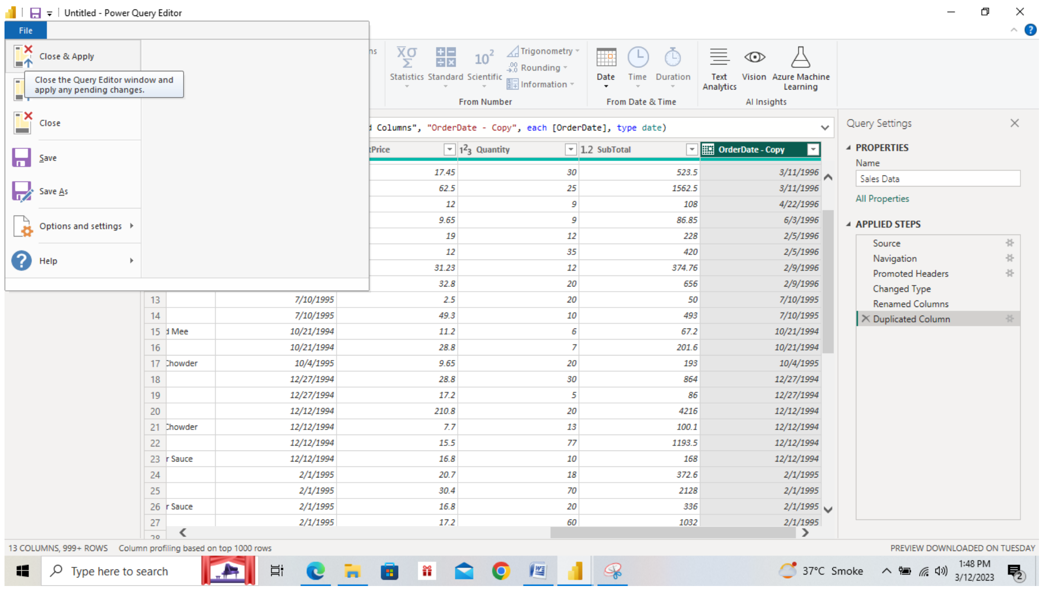
6. Combine the Products and Total Sales queries

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 the Query Editor window and apply any pending changes.

```
d Columns", "OrderDate - Copy", each [OrderDate], type date)
```

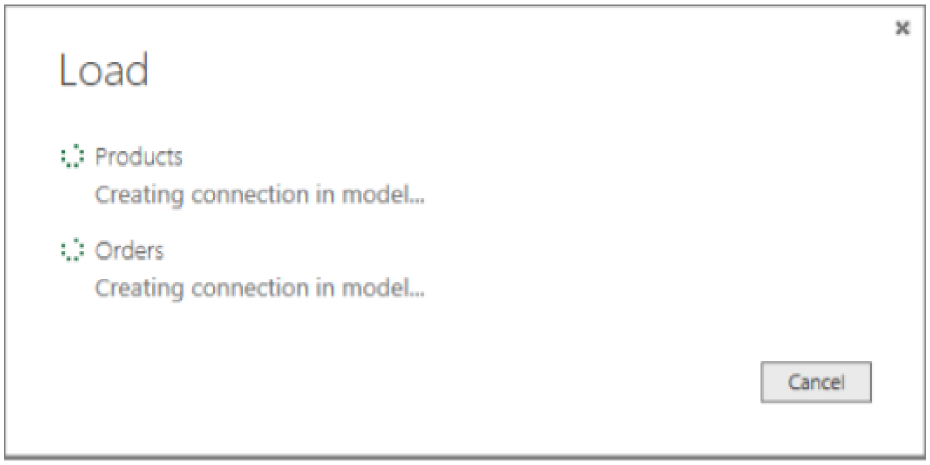
			tPrice	Quantity	SubTotal	OrderDate - Copy
			17.45	30	523.5	3/11/1996
			62.5	25	1562.5	3/11/1996
			12	9	108	4/22/1996
			9.65	9	86.85	6/3/1996
			19	12	228	2/5/1996
			12	35	420	2/5/1996
			31.23	12	374.76	2/9/1996
			32.8	20	656	2/9/1996
13		7/10/1995	2.5	20	50	7/10/1995
14		7/10/1995	49.3	10	493	7/10/1995
15	d Mee	10/21/1994	11.2	6	67.2	10/21/1994
16		10/21/1994	28.8	7	201.6	10/21/1994
17	Chowder	10/4/1995	9.65	20	193	10/4/1995
18		12/27/1994	28.8	30	864	12/27/1994
19		12/27/1994	17.2	5	86	12/27/1994
20		12/12/1994	210.8	20	4216	12/12/1994
21	Chowder	12/12/1994	7.7	13	100.1	12/12/1994
22		12/12/1994	15.5	77	1193.5	12/12/1994
23	r Sauce	12/12/1994	16.8	10	168	12/12/1994
24		2/1/1995	20.7	18	372.6	2/1/1995
25		2/1/1995	30.4	70	2128	2/1/1995
26	r Sauce	2/1/1995	16.8	20	336	2/1/1995
27		2/1/1995	17.2	60	1032	2/1/1995

Name: Sales Data

APPLIED STEPS: Source, Navigation, Promoted Headers, Changed Type, Renamed Columns, Duplicated Column

13 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON TUESDAY

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



Load

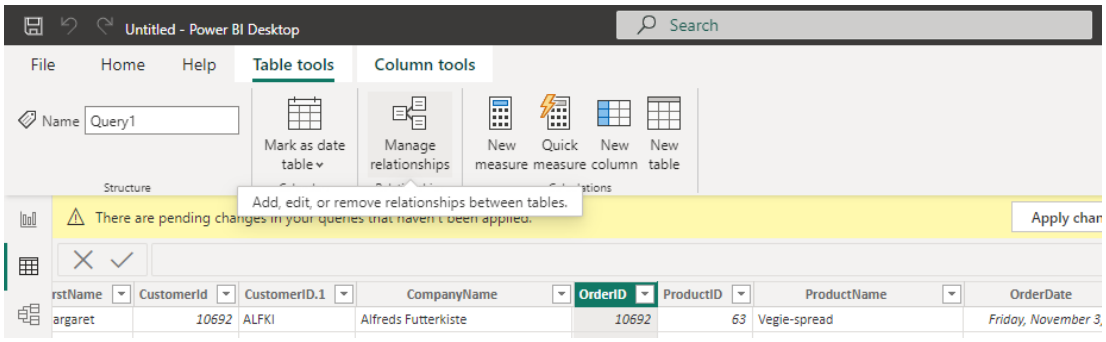
Products

Creating connection in model...

Orders

Creating connection in model...

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

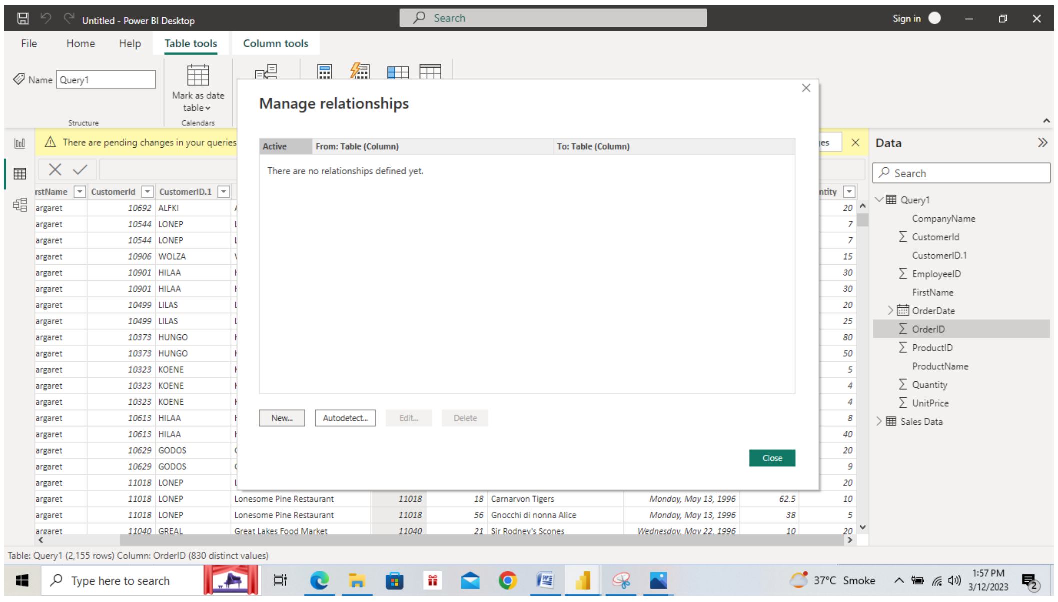


Add, edit, or remove relationships between tables.

There are pending changes in your queries that haven't been applied.

rstName	CustomerId	CustomerID.1	CompanyName	OrderID	ProductID	ProductName	OrderDate
argaret	10692	ALFKI	Alfreds Futterkiste	10692	63	Vegie-spread	Friday, November 3,

4. Select new button



Manage relationships

Active	From: Table (Column)	To: Table (Column)

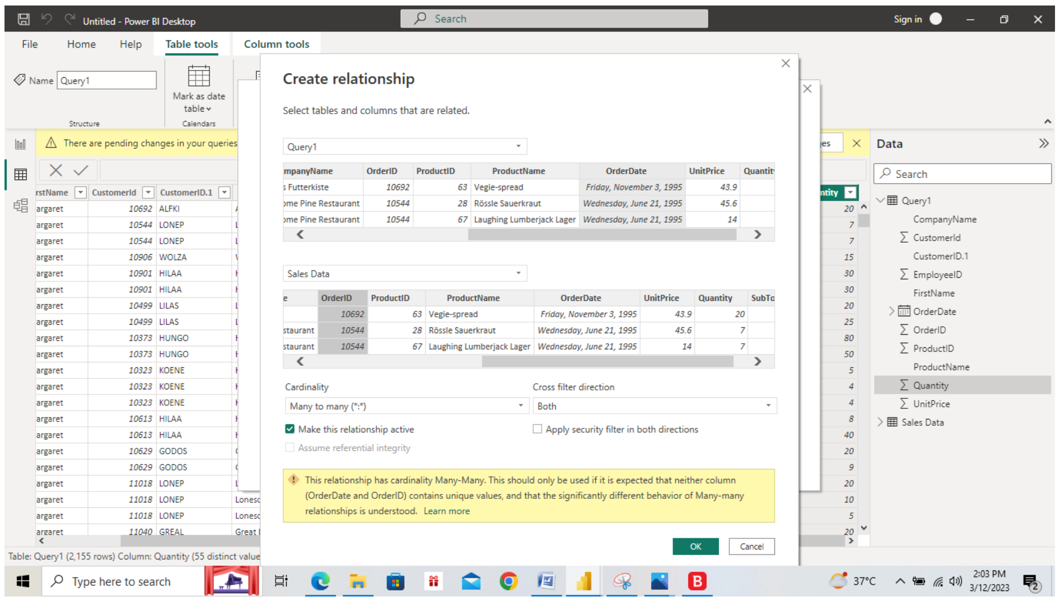
There are no relationships defined yet.

There are pending changes in your queries

rstName	CustomerId	CustomerID.1							ntity
argaret	10692	ALFKI							20
argaret	10544	LONEP							7
argaret	10544	LONEP							7
argaret	10906	WOLZA							15
argaret	10901	HILAA							30
argaret	10901	HILAA							30
argaret	10499	LILAS							20
argaret	10499	LILAS							25
argaret	10373	HUNGO							80
argaret	10373	HUNGO							50
argaret	10323	KOENE							5
argaret	10323	KOENE							4
argaret	10323	KOENE							4
argaret	10613	HILAA							8
argaret	10613	HILAA							40
argaret	10629	GODOS							20
argaret	10629	GODOS							9
argaret	11018	LONEP							20
argaret	11018	LONEP	Lonesome Pine Restaurant	11018	18	Carnarvon Tigers	Monday, May 13, 1996	62.5	10
argaret	11018	LONEP	Lonesome Pine Restaurant	11018	56	Gnocchi di nonna Alice	Monday, May 13, 1996	38	5
argaret	11040	GREAL	Great Lakes Food Market	11040	21	Sir Rodney's Scones	Wednesday, May 22, 1996	10	20

Table: Query1 (2,155 rows) Column: OrderID (830 distinct values)

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 are related.

Query1

mpanyName	OrderID	ProductID	ProductName	OrderDate	UnitPrice	Quantit
s Futterkiste	10692	63	Vegie-spread	Friday, November 3, 1995	43.9	
ome Pine Restaurant	10544	28	Rössle Sauerkraut	Wednesday, June 21, 1995	45.6	
ome Pine Restaurant	10544	67	Laughing Lumberjack Lager	Wednesday, June 21, 1995	14	

Sales Data

e	OrderID	ProductID	ProductName	OrderDate	UnitPrice	Quantity	SubTo
	10692	63	Vegie-spread	Friday, November 3, 1995	43.9	20	
staurant	10544	28	Rössle Sauerkraut	Wednesday, June 21, 1995	45.6	7	
staurant	10544	67	Laughing Lumberjack Lager	Wednesday, June 21, 1995	14	7	

Cardinality: Many to many (*:*)

Cross filter direction: Both

[✓] Make this relationship active

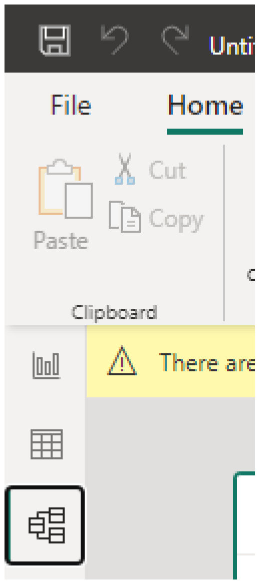
[ ] Apply security filter in both directions

[ ] Assume referential integrity

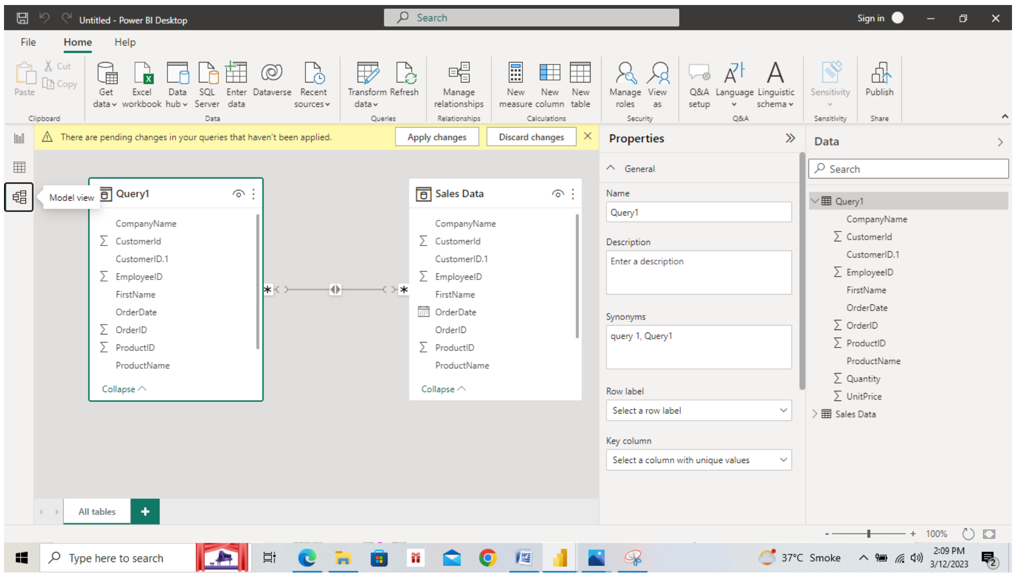
This relationship has cardinality Many-Many. This should only be used if it is expected that neither column (OrderDate and OrderID) contains unique values, and that the significantly different behavior of Many-many relationships is understood. Learn more

OK Cancel

5. Select cancel , and then select relationship view in power BI desktop



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



There are pending changes in your queries that haven't been applied. Apply changes Discard changes

Query1	Sales Data
CompanyName	CompanyName
CustomerId	CustomerId
CustomerID.1	CustomerID.1
EmployeeID	EmployeeID
FirstName	FirstName
OrderDate	OrderDate
OrderID	OrderID
ProductID	ProductID
ProductName	ProductName
Collapse	Collapse

Properties

General

Name: Query1

Description: Enter a description

Synonyms: query 1, Query1

Row label: Select a row label

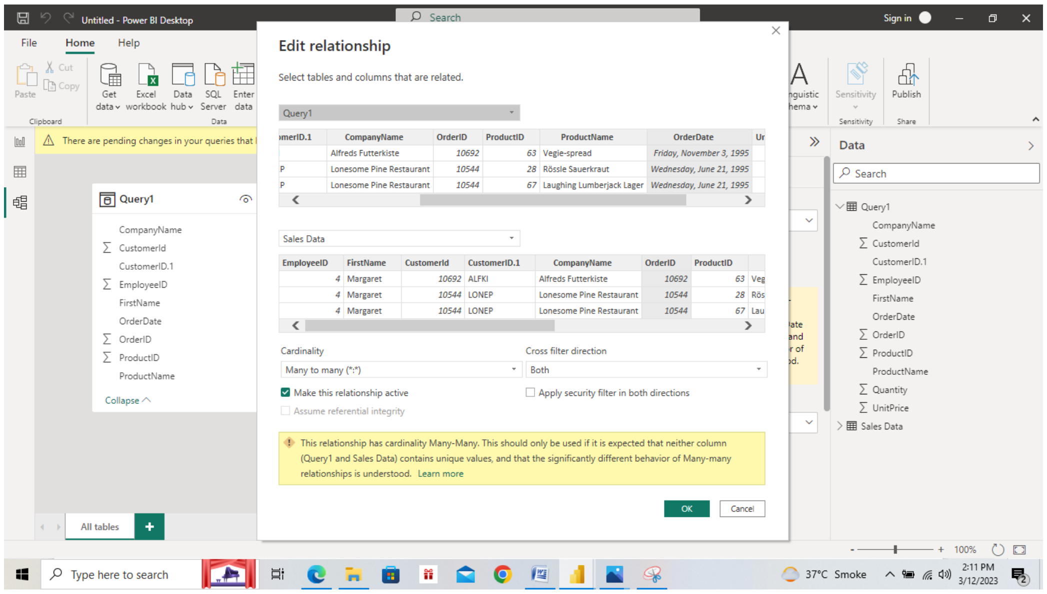
Key column: Select a column with unique values

Data

Search

- Query1
  - CompanyName
  - CustomerId
  - CustomerID.1
  - EmployeeID
  - FirstName
  - OrderDate
  - OrderID
  - ProductID
  - ProductName
  - Quantity
  - UnitPrice
- Sales Data

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 are related.

Query1

…merID.1	CompanyName	OrderID	ProductID	ProductName	OrderDate	Ur
	Alfreds Futterkiste	10692	63	Vegie-spread	Friday, November 3, 1995	
P	Lonesome Pine Restaurant	10544	28	Rössle Sauerkraut	Wednesday, June 21, 1995	
P	Lonesome Pine Restaurant	10544	67	Laughing Lumberjack Lager	Wednesday, June 21, 1995	

Sales Data

EmployeeID	FirstName	CustomerId	CustomerID.1	CompanyName	OrderID	ProductID	
4	Margaret	10692	ALFKI	Alfreds Futterkiste	10692	63	Veg
4	Margaret	10544	LONEP	Lonesome Pine Restaurant	10544	28	Rös
4	Margaret	10544	LONEP	Lonesome Pine Restaurant	10544	67	Lau

Cardinality: Many to many (*:*)

Cross filter direction: Both

[✓] Make this relationship active

[ ] Apply security filter in both directions

[ ] Assume referential integrity

This relationship has cardinality Many-Many. This should only be used if it is expected that neither column (Query1 and Sales Data) contains unique values, and that the significantly different behavior of Many-many relationships is understood. Learn more

OK Cancel

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

Practical 11: 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.

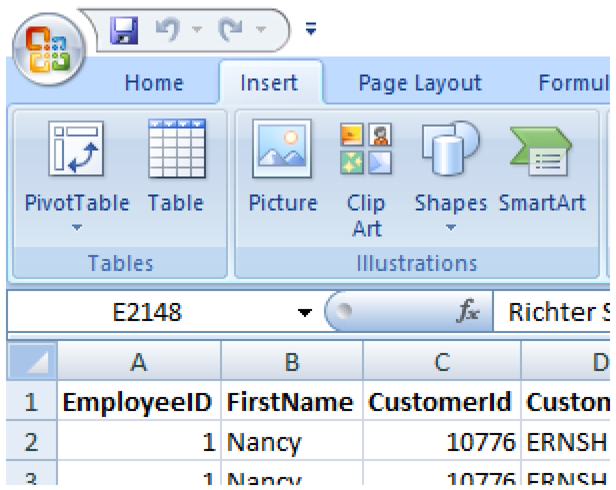
Step 4 – Select Events.accdb, Events Access Database file.

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

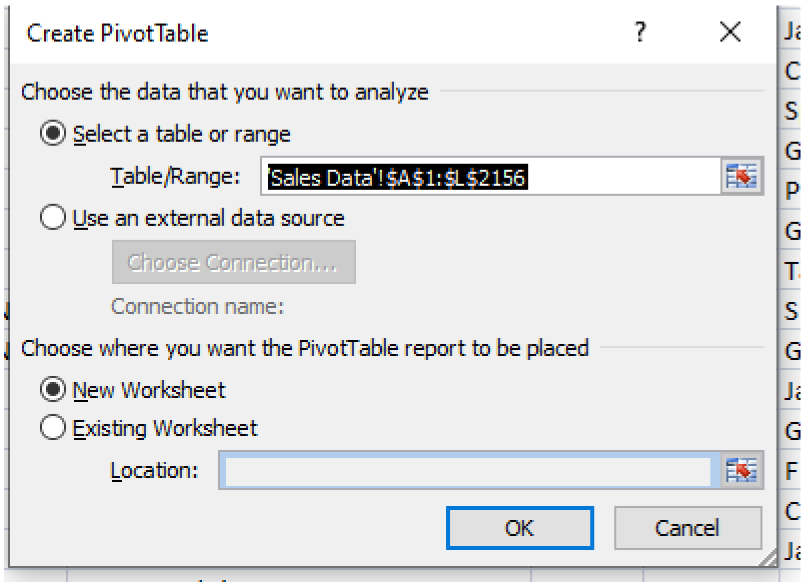
Step 6 – Tables in a database are similar to the tables in Excel. Check the 'Enable selection of multiple tables' box, and select all the tables. Then click OK.

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.



	A	B	C	D
1	EmployeeID	FirstName	CustomerId	Custon
2	1	Nancy	10776	ERNSH
3	1	Nancy	10776	ERNSH


Create PivotTable

Choose the data that you want to analyze

[✓ Select a table or range]

Table/Range: 'Sales Data'!$A$1:$L$2156

Use an external data source

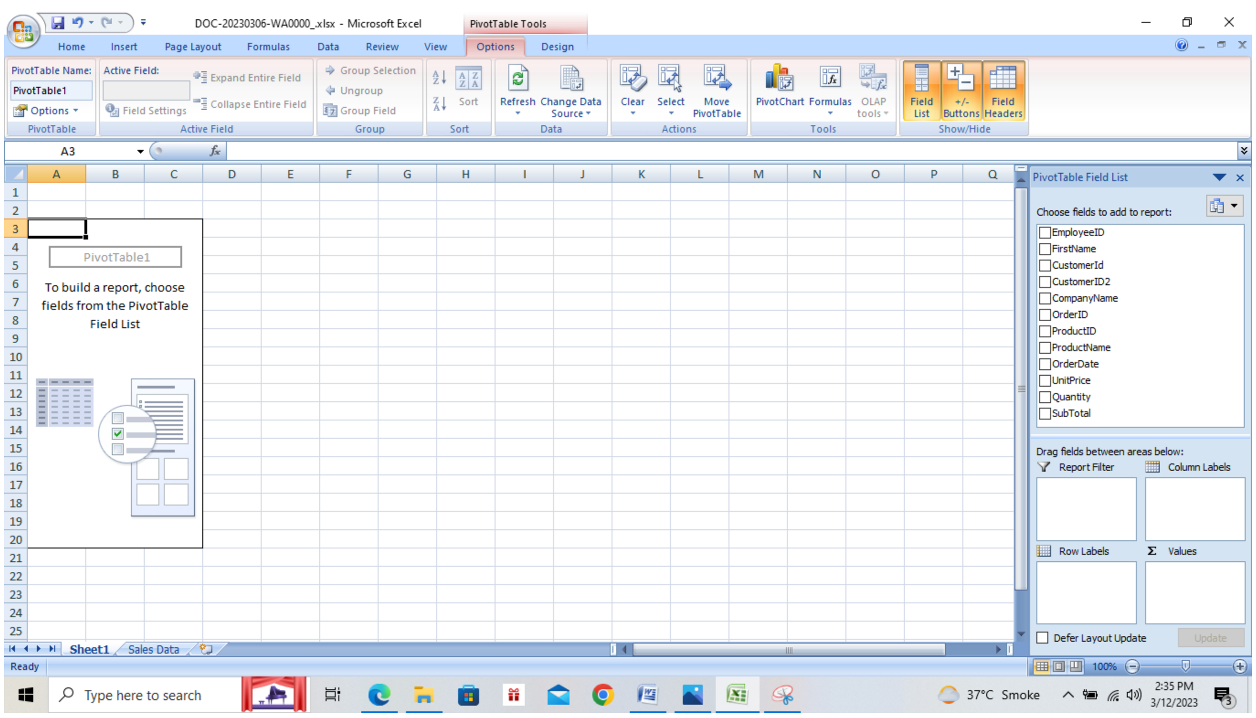
Connection name:

Choose where you want the PivotTable report to be placed

[✓ New Worksheet]

Existing Worksheet

Location:


DOC-20230306-WA0000_.xlsx - Microsoft Excel

PivotTable1

To build a report, choose fields from the PivotTable Field List

PivotTable Field List

Choose fields to add to report:

- EmployeeID
- FirstName
- CustomerId
- CustomerID2
- CompanyName
- OrderID
- ProductID
- ProductName
- OrderDate
- UnitPrice
- Quantity
- SubTotal

Drag fields between areas below:

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 bestsuited report.

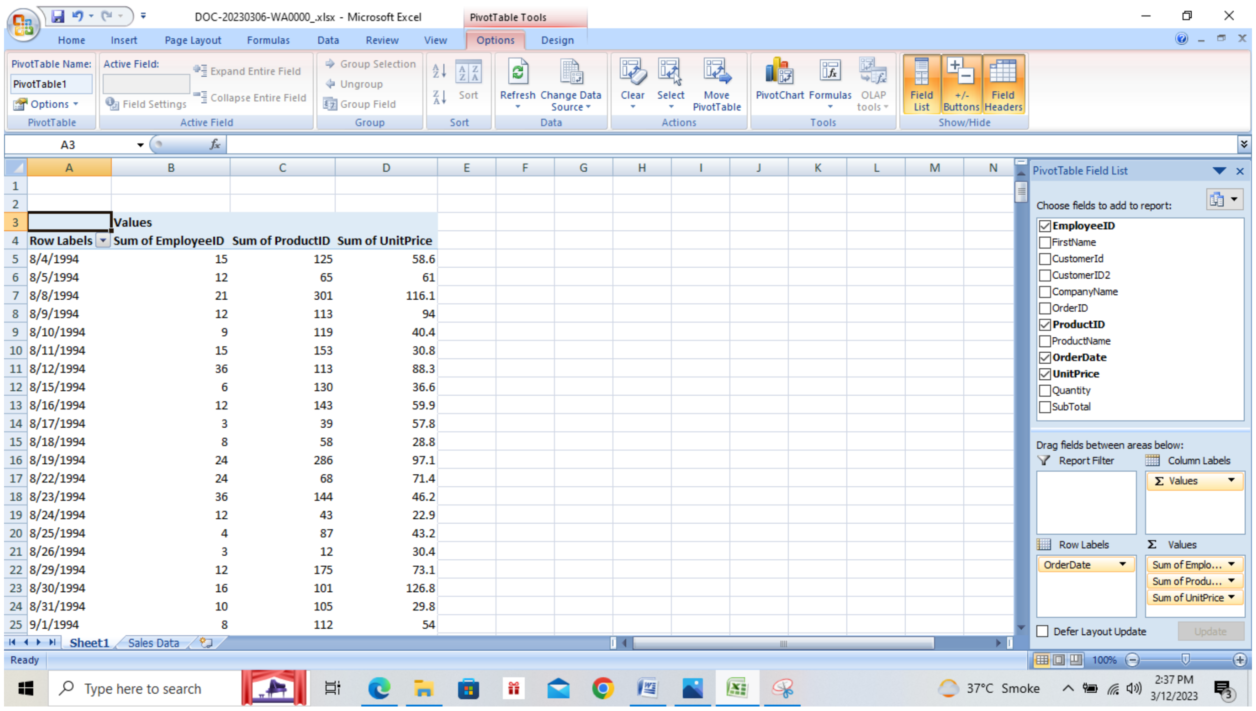
In PivotTable Fields, click on the arrow beside the table - Medals to expand it to show the fields in that table. Drag the NOC_CountryRegion field in the Medals table to the COLUMNS area.

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.



	Values		
Row Labels	Sum of EmployeeID	Sum of ProductID	Sum of UnitPrice
8/4/1994	15	125	58.6
8/5/1994	12	65	61
8/8/1994	21	301	116.1
8/9/1994	12	113	94
8/10/1994	9	119	40.4
8/11/1994	15	153	30.8
8/12/1994	36	113	88.3
8/15/1994	6	130	36.6
8/16/1994	12	143	59.9
8/17/1994	3	39	57.8
8/18/1994	8	58	28.8
8/19/1994	24	286	97.1
8/22/1994	24	68	71.4
8/23/1994	36	144	46.2
8/24/1994	12	43	22.9
8/25/1994	4	87	43.2
8/26/1994	3	12	30.4
8/29/1994	12	175	73.1
8/30/1994	16	101	126.8
8/31/1994	10	105	29.8
9/1/1994	8	112	54

PivotTable Field List

Choose fields to add to report:

- ☑ EmployeeID
- ☐ FirstName
- ☐ CustomerId
- ☐ CustomerID2
- ☐ CompanyName
- ☐ OrderID
- ☑ ProductID
- ☐ ProductName
- ☑ OrderDate
- ☑ UnitPrice
- ☐ Quantity
- ☐ SubTotal

Drag fields between areas below:

Report Filter | Column Labels: Σ Values

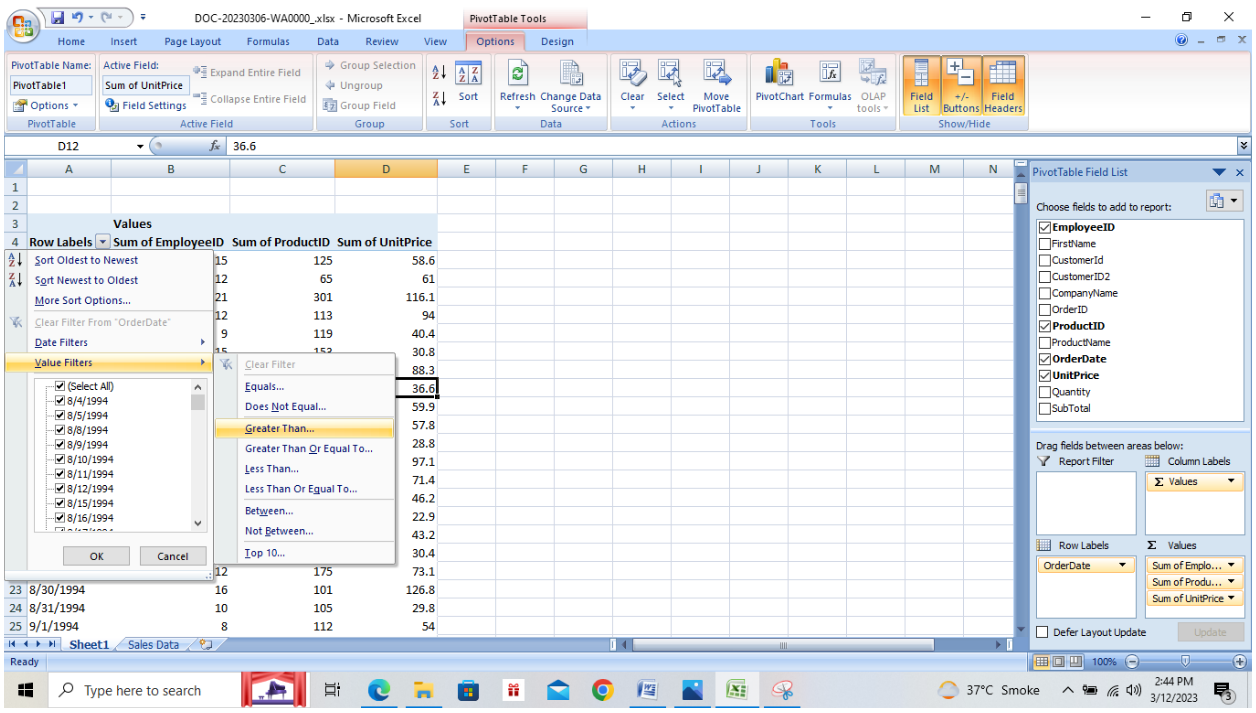
Row Labels: OrderDate | Values: Sum of Emplo…, Sum of Produ…, Sum of UnitPrice

☐ Defer Layout Update — Update

Step 6 – click dropdown list to the right of the column lables

Step 7 – select value filter and then select greater than

Step 8 – click ok



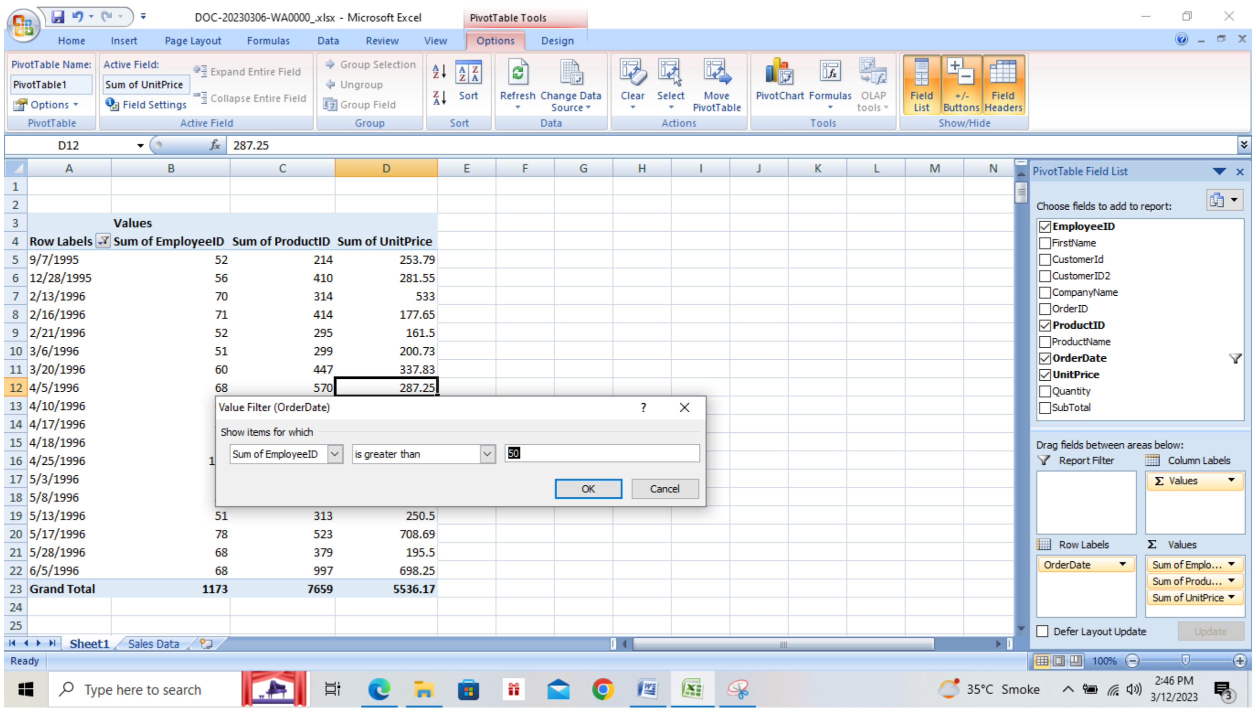
D12 36.6

	Values		
Row Labels	Sum of EmployeeID	Sum of ProductID	Sum of UnitPrice
	15	125	58.6
	12	65	61
	21	301	116.1
	12	113	94
	9	119	40.4
	15	152	30.8
			88.3
			36.6
			59.9
			57.8
			28.8
			97.1
			71.4
			46.2
			22.9
			43.2
			30.4
	12	175	73.1
8/30/1994	16	101	126.8
8/31/1994	10	105	29.8
9/1/1994	8	112	54

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.



D12 287.25

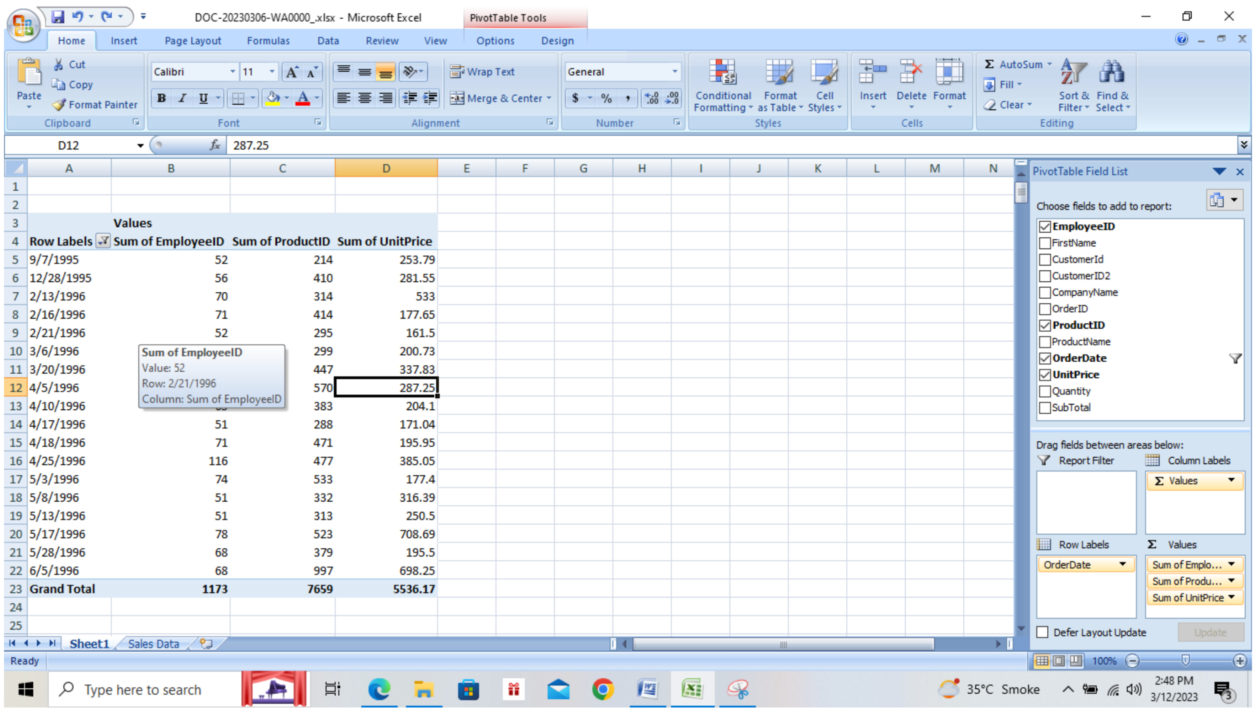
	Values		
Row Labels	Sum of EmployeeID	Sum of ProductID	Sum of UnitPrice
9/7/1995	52	214	253.79
12/28/1995	56	410	281.55
2/13/1996	70	314	533
2/16/1996	71	414	177.65
2/21/1996	52	295	161.5
3/6/1996	51	299	200.73
3/20/1996	60	447	337.83
4/5/1996	68	570	287.25
4/10/1996			
4/17/1996			
4/18/1996			
4/25/1996	1		
5/3/1996			
5/8/1996			
5/13/1996	51	313	250.5
5/17/1996	78	523	708.69
5/28/1996	68	379	195.5
6/5/1996	68	997	698.25
Grand Total	1173	7659	5536.17

Value Filter (OrderDate)

Show items for which

Sum of EmployeeID | is greater than | 50

OK Cancel



D12 287.25

	Values		
Row Labels	Sum of EmployeeID	Sum of ProductID	Sum of UnitPrice
9/7/1995	52	214	253.79
12/28/1995	56	410	281.55
2/13/1996	70	314	533
2/16/1996	71	414	177.65
2/21/1996	52	295	161.5
3/6/1996		299	200.73
3/20/1996		447	337.83
4/5/1996		570	287.25
4/10/1996		383	204.1
4/17/1996	51	288	171.04
4/18/1996	71	471	195.95
4/25/1996	116	477	385.05
5/3/1996	74	533	177.4
5/8/1996	51	332	316.39
5/13/1996	51	313	250.5
5/17/1996	78	523	708.69
5/28/1996	68	379	195.5
6/5/1996	68	997	698.25
Grand Total	**1173**	**7659**	**5536.17**

Sum of EmployeeID
Value: 52
Row: 2/21/1996
Column: Sum of EmployeeID

PivotTable Field List

Choose fields to add to report:

- [✓] **EmployeeID**
- [ ] FirstName
- [ ] CustomerId
- [ ] CustomerID2
- [ ] CompanyName
- [ ] OrderID
- [✓] **ProductID**
- [ ] ProductName
- [✓] **OrderDate**
- [✓] **UnitPrice**
- [ ] Quantity
- [ ] SubTotal

Drag fields between areas below:

Report Filter:

Column Labels: Σ Values

Row Labels: OrderDate

Values: Sum of Emplo…, Sum of Produ…, Sum of UnitPrice

[ ] Defer Layout Update

Practical 5: Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data.

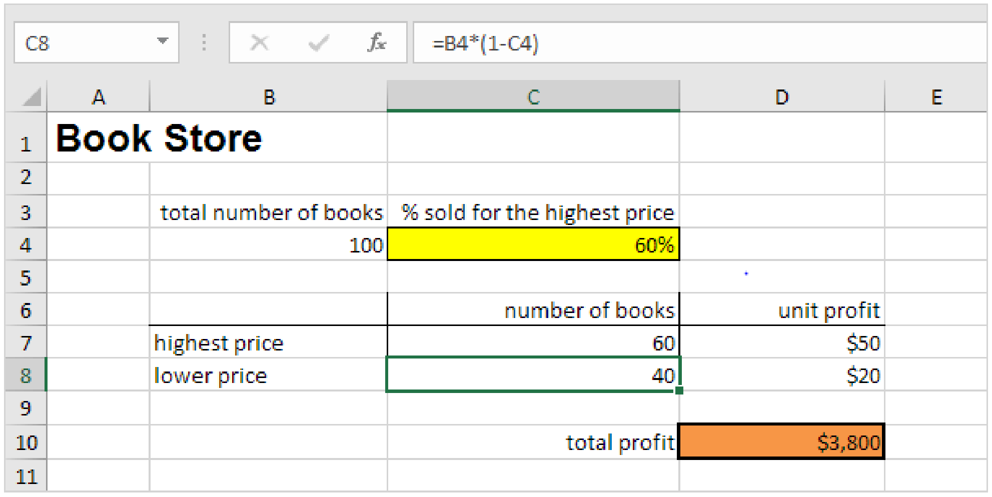
A book store and have 100 books in storage.

You sell a certain % for the highest price of $50 and a certain % for the lower price of $20.

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.

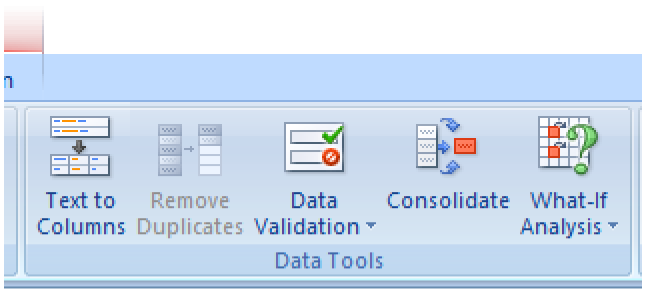


C8 =B4*(1-C4)

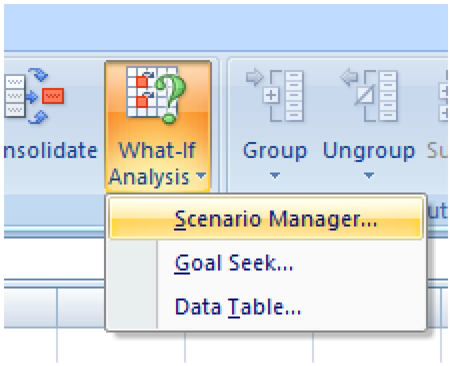
	A	B	C	D	E
1	**Book Store**				
2					
3		total number of books	% sold for the highest price		
4		100	60%		
5					
6			number of books	unit profit	
7		highest price	60	$50	
8		lower price	40	$20	
9					
10			total profit	$3,800	
11					

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.

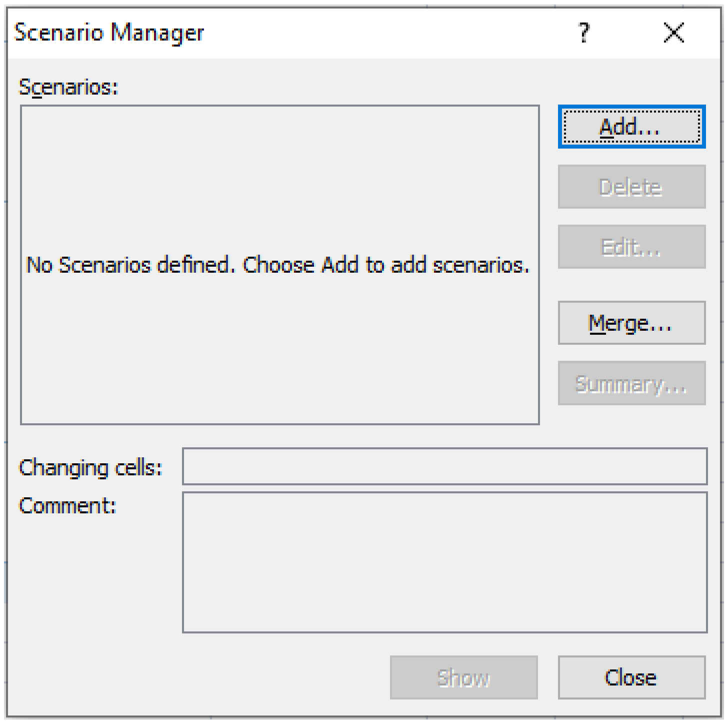


2. Click Scenario Manager.

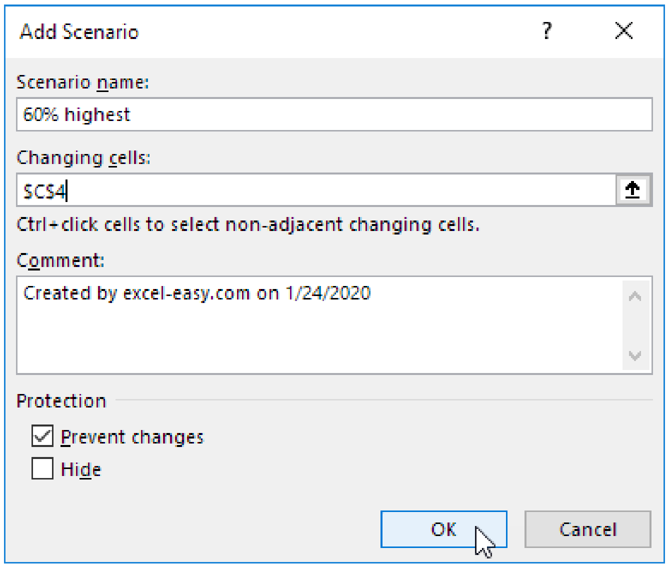


The Scenario Manager dialog box appears.

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



No Scenarios defined. Choose Add to add scenarios.


Add Scenario

Scenario name: 60% highest

Changing cells: $C$4

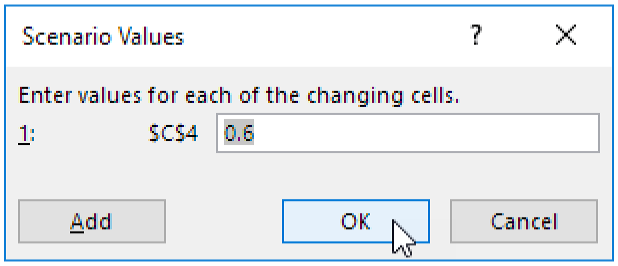
Ctrl+click cells to select non-adjacent changing cells.

Comment: Created by excel-easy.com on 1/24/2020

Protection

- [x] Prevent changes
- [ ] Hide

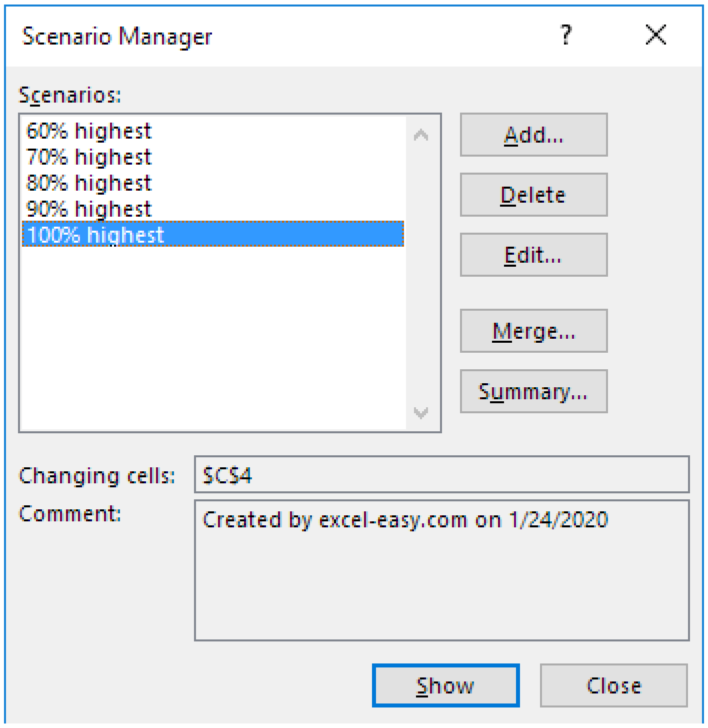
5. Enter the corresponding value 0.6 and click on OK again.


Scenario Values

Enter values for each of the changing cells.

1: $C$4 0.6

6. Next, add 4 other scenarios (70%, 80%, 90% and 100%). Finally, your Scenario Manager should be consistent with the picture below:


Scenario Manager

Scenarios:
60% highest
70% highest
80% highest
90% highest
100% highest

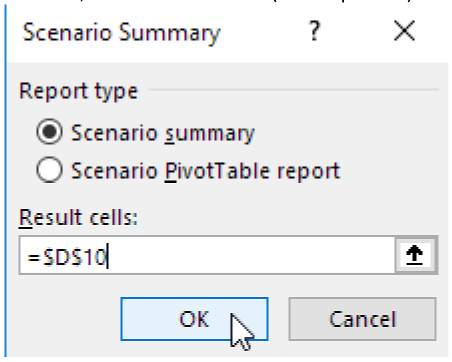
Changing cells: $C$4

Comment: Created by excel-easy.com on 1/24/2020

Scenario Summary

To easily compare the results of these scenarios, execute the following steps.

1. Click the Summary button in the Scenario Manager.
2. Next, select cell D10 (total profit) for the result cell and click on OK



Scenario Summary

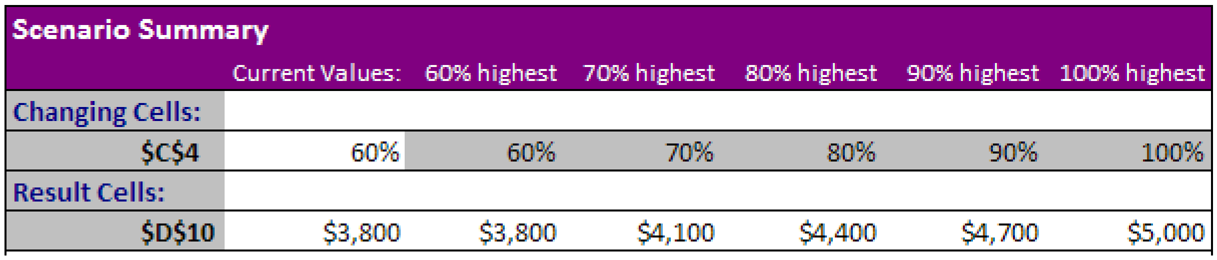
Report type

● Scenario summary

○ Scenario PivotTable report

Result cells:

=$D$10



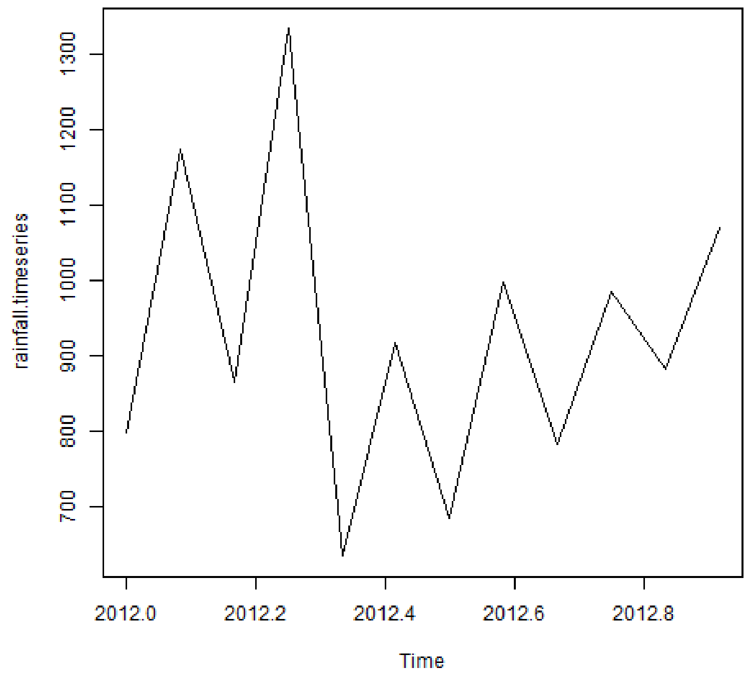
Scenario Summary

	Current Values:	60% highest	70% highest	80% highest	90% highest	100% highest
Changing Cells:						
$C$4	60%	60%	70%	80%	90%	100%
Result Cells:						
$D$10	$3,800	$3,800	$4,100	$4,400	$4,700	$5,000

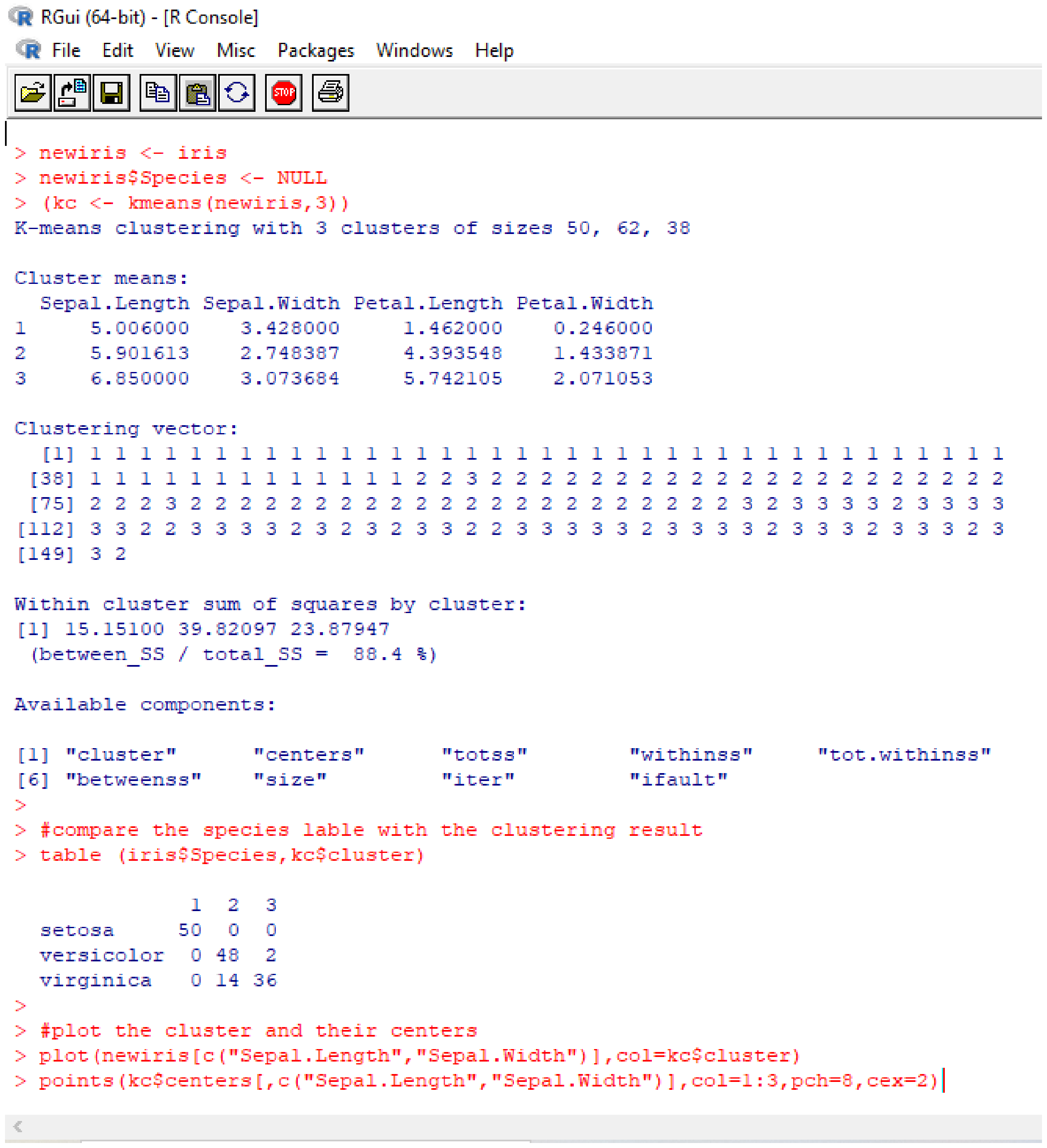
Practical 6: Implementation of Classification algorithm in R Programming.

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.

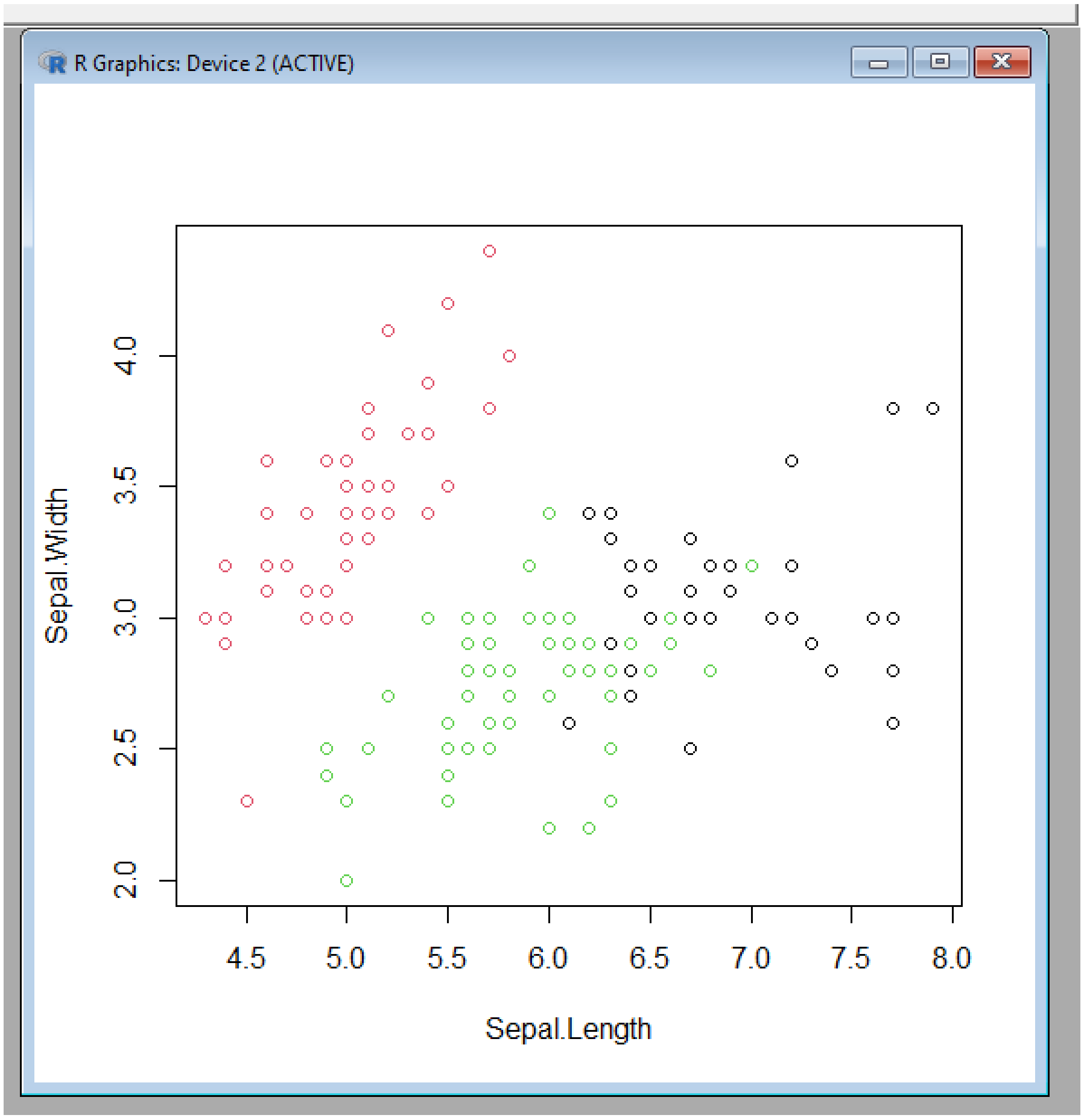
```
# Get the data points in form of a R vector.
rainfall <-
c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)

# 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()
```



Practical 8: k-means clustering using R


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
> newiris <- iris
> newiris$Species <- NULL
> (kc <- kmeans(newiris,3))
K-means clustering with 3 clusters of sizes 50, 62, 38

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

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3
[112] 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3
[149] 3 2

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

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
>
> #compare the species lable with the clustering result
> table (iris$Species,kc$cluster)

              1  2  3
  setosa     50  0  0
  versicolor  0 48  2
  virginica   0 14 36
>
> #plot the cluster 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)
```


R Graphics: Device 2 (ACTIVE)
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

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.

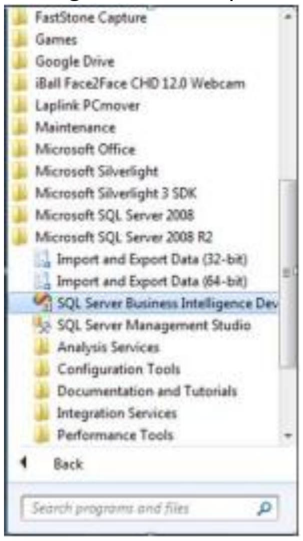
Follow the given steps to run the query in SSMS (SQL Server Management Studio).

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

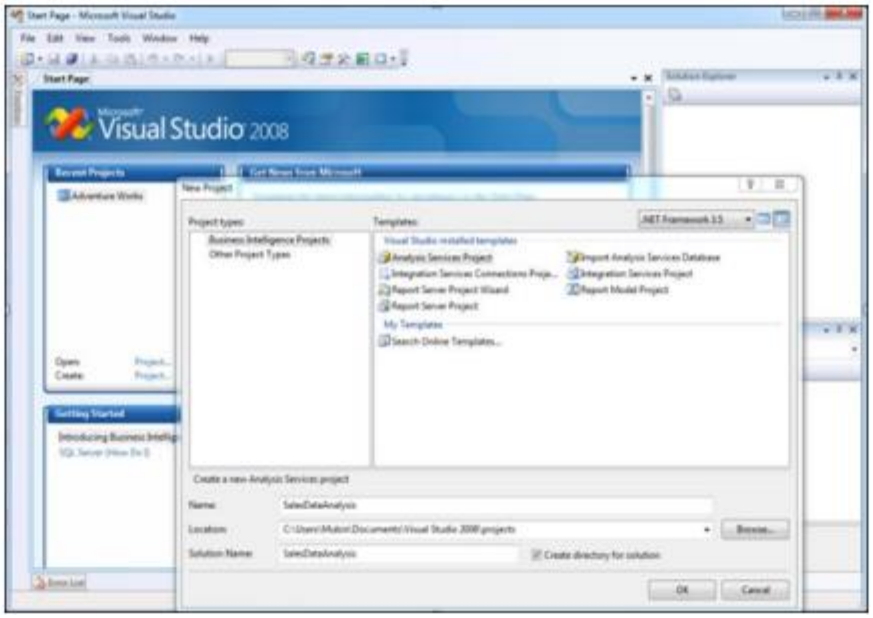
Step 1: Start BIDS Environment

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
Visual Studio 2008
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

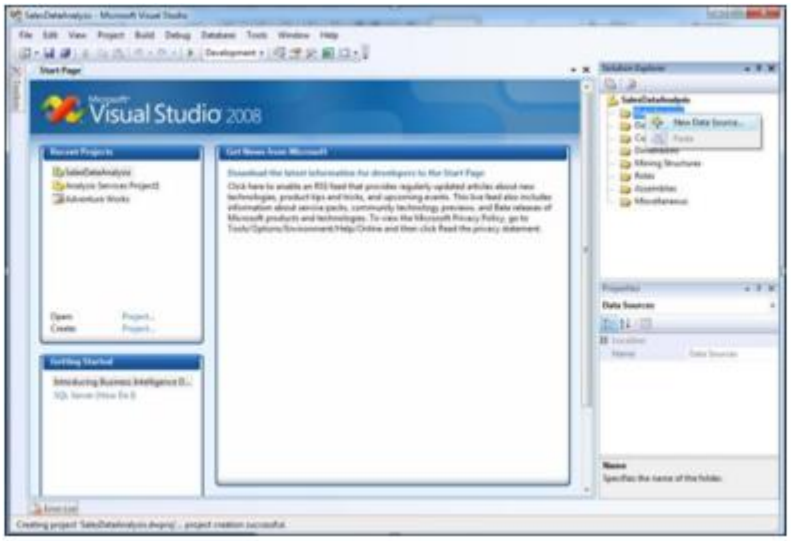
Name: SalesDataAnalysis

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

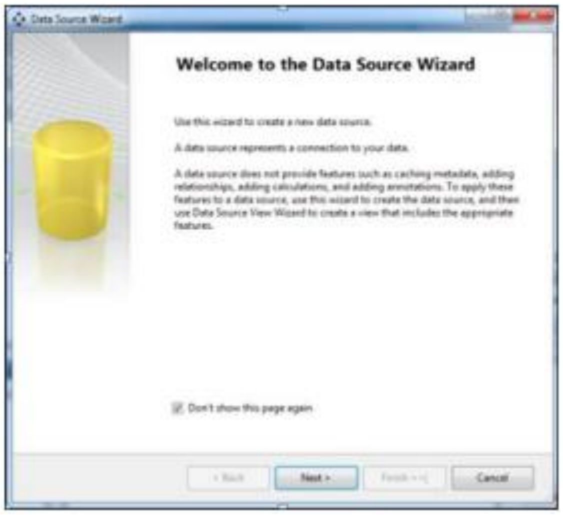
Solution Name: SalesDataAnalysis

[✓] Create directory for solution

Step 3: Creating New Data Source
3.1 In Solution Explorer, Right click on Data Source -> Click New Data Source


SalesDataAnalysis - Microsoft Visual Studio
File Edit View Project Build Debug Database Tools Window Help
Start Page
Visual Studio 2008
Solution Explorer
SalesDataAnalysis
New Data Source...
Paste
Mining Structures
Roles
Assemblies
Miscellaneous
Properties
Data Sources
Name
Specifies the name of the folder.
Creating project 'SalesDataAnalysis.dwproj'... project creation successful.

3.2 Click on Next



Data Source Wizard

Welcome to the Data Source Wizard

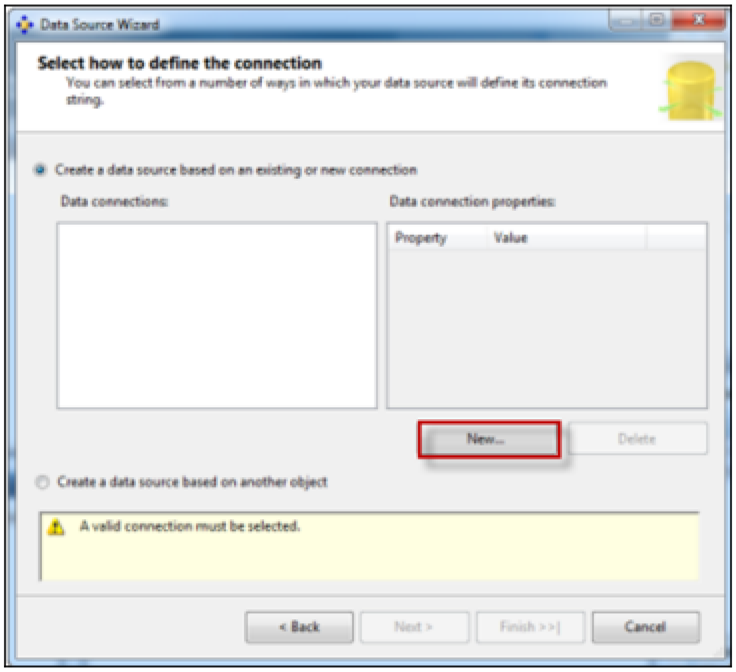
Use this wizard to create a new data source.

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

3.3 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:

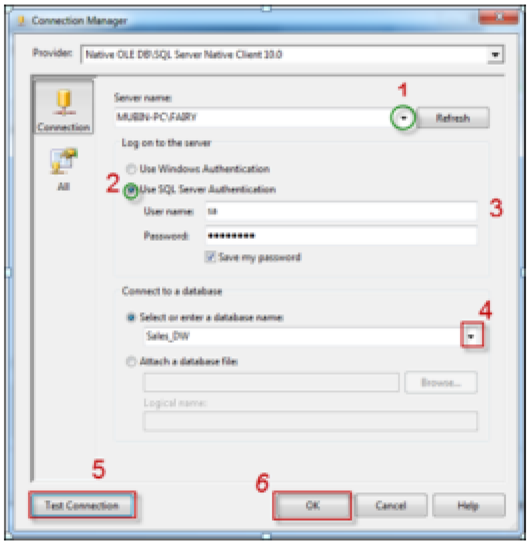
Data connection properties:

Property	Value

○ Create a data source based on another object

A valid connection must be selected.

3.4 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


Connection Manager

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

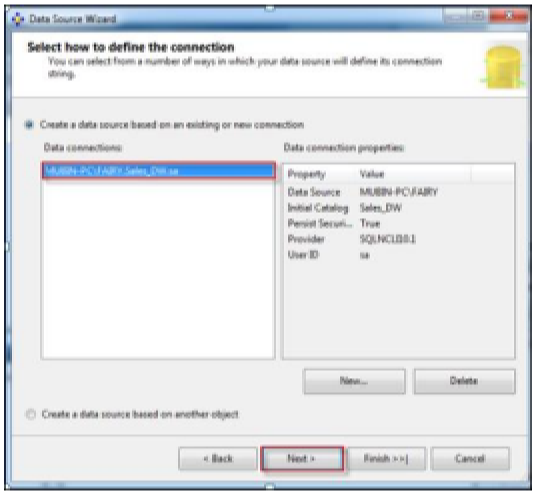
Connect to a database

[✓ Select or enter a database name:] Sales_DW

Attach a database file:

Logical name:

3.5 Select Connection created in **Data Connections**-> Click **Next**


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:

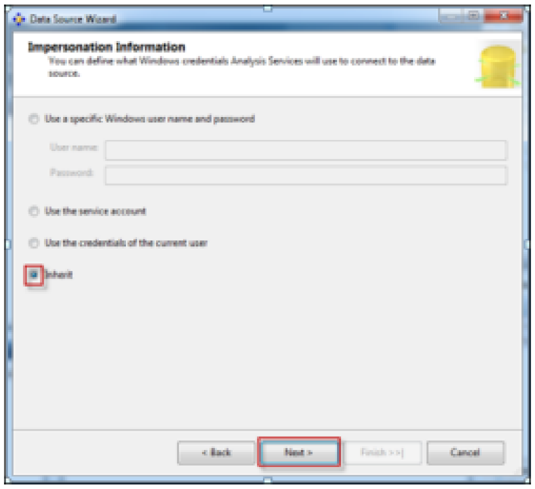
MUBIN-PC\FAIRY.Sales_DW.sa

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

3.6 Select Option **Inherit**


Data Source Wizard

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:

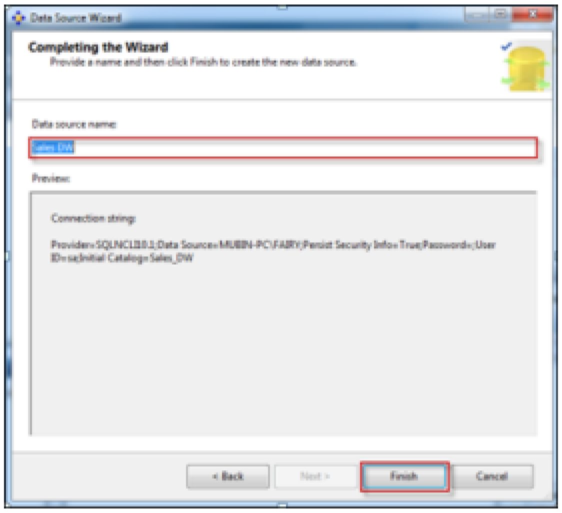
Password:

Use the service account

Use the credentials of the current user

[✓ Inherit]

3.7 Assign Data Source **Name** -> Click **Finish**



Completing the Wizard

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

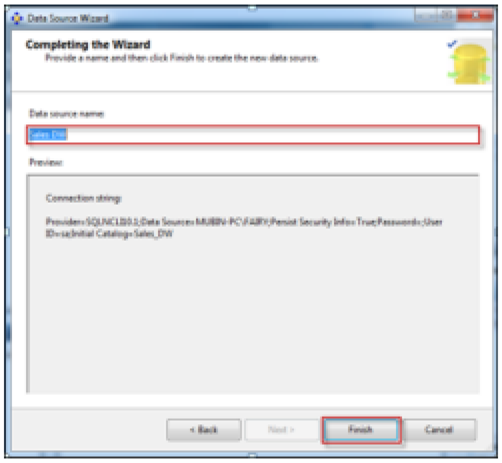
Data source name:

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

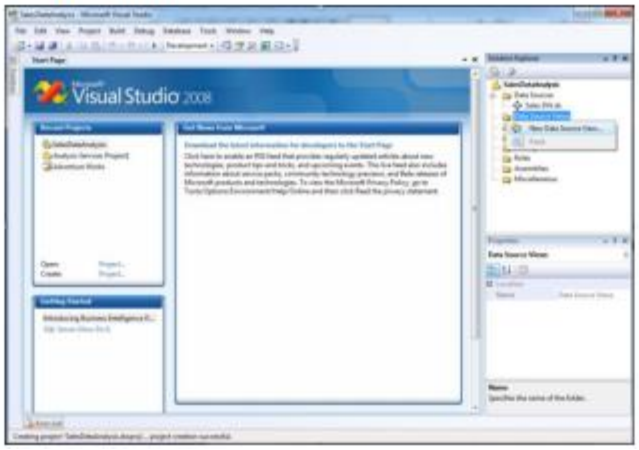
Preview:

Connection string:

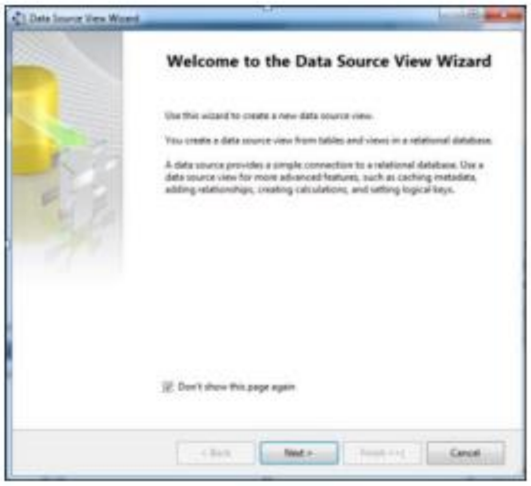
Provider=SQLNCLI10.1;Data Source=MUBIN-PC\FAIRY;Persist Security Info=True;Password=;User ID=sa;Initial Catalog=Sales_DW

Step 4: Creating New Data Source View

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



4.2 Click **Next**



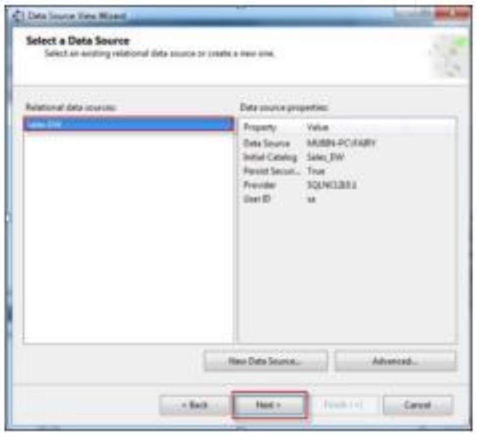
Data Source View Wizard
Welcome to the Data Source View Wizard
Don't show this page again
< Back
Next >
Finish >>|
Cancel

Use this wizard to create a new data source view.

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

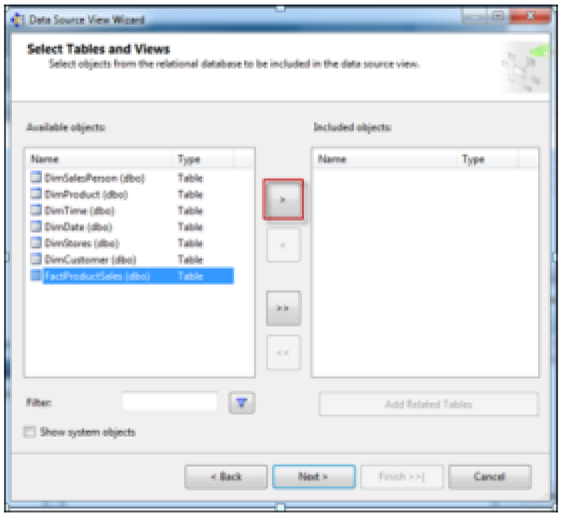
A data source provides a simple connection to a relational database. Use a data source view for more advanced features, such as caching metadata, adding relationships, creating calculations, and setting logical keys.

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



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

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

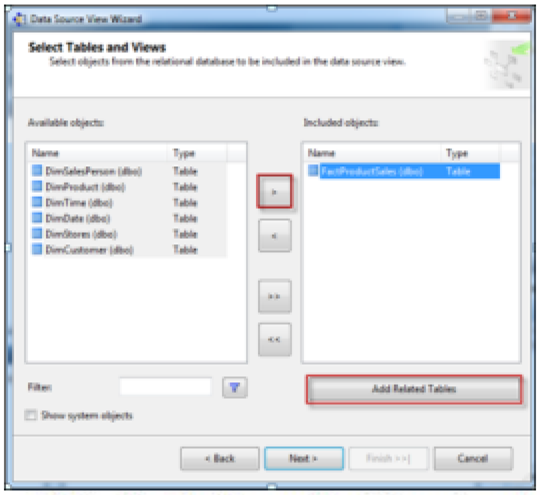


Name	Type
DimSalesPerson (dbo)	Table
DimProduct (dbo)	Table
DimTime (dbo)	Table
DimDate (dbo)	Table
DimStores (dbo)	Table
DimCustomer (dbo)	Table
FactProductSales (dbo)	Table

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

4.5 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:

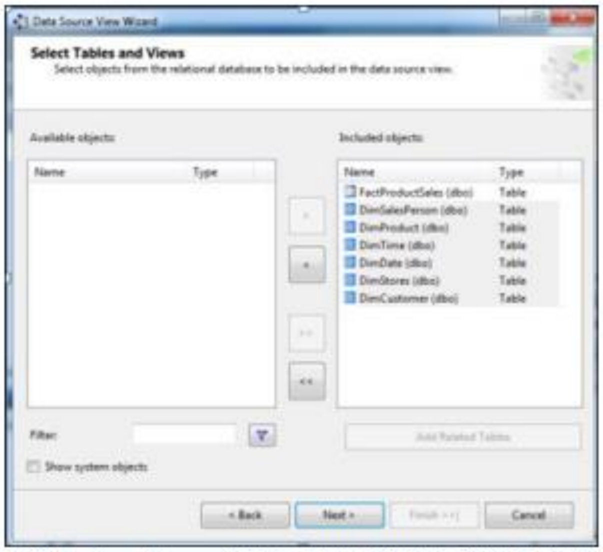
Name	Type
DimSalesPerson (dbo)	Table
DimProduct (dbo)	Table
DimTime (dbo)	Table
DimDate (dbo)	Table
DimStores (dbo)	Table
DimCustomer (dbo)	Table

Included objects:

Name	Type
FactProductSales (dbo)	Table

4.6 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.

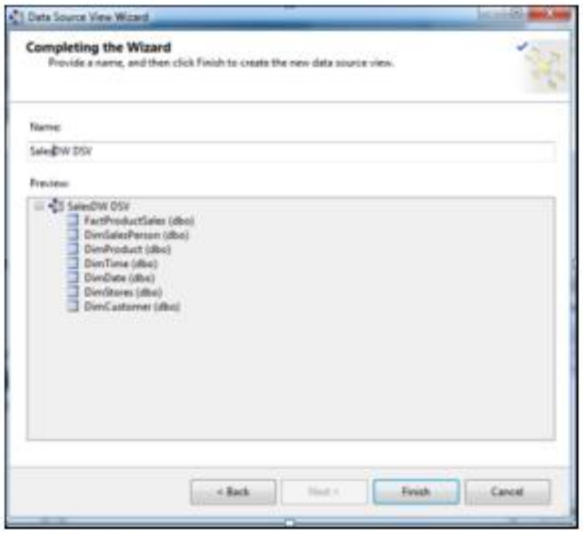
Available objects:

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

4.7 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.

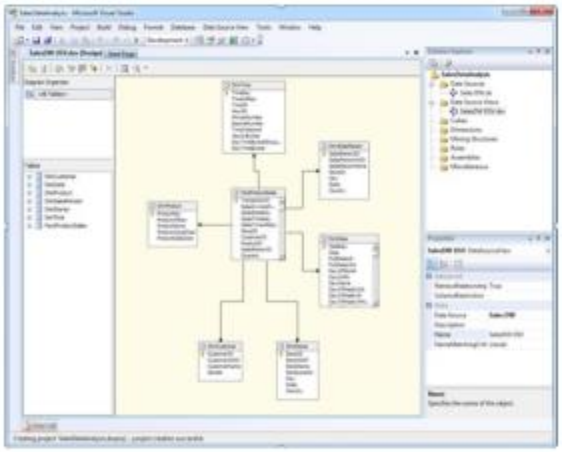
Name:

SalesDW DSV

Preview:

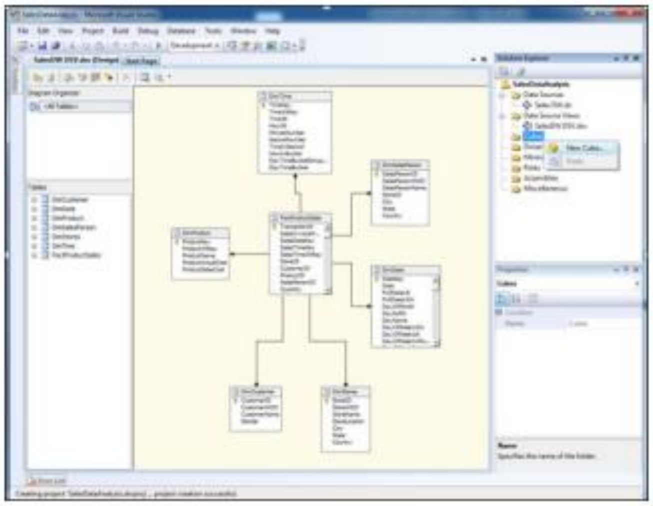
- SalesDW DSV
  - FactProductSales (dbo)
  - DimSalesPerson (dbo)
  - DimProduct (dbo)
  - DimTime (dbo)
  - DimDate (dbo)
  - DimStores (dbo)
  - DimCustomer (dbo)

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

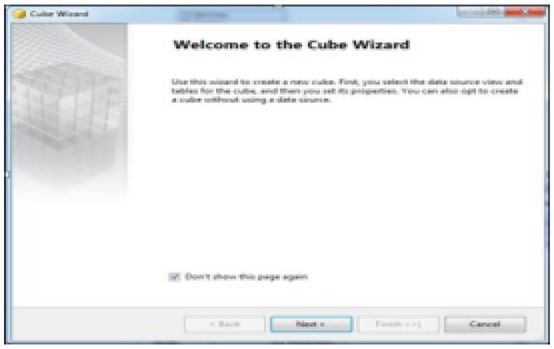


Step 5: Creating New Cube

5.1 In Solution Explorer -> Right Click on **Cube->** Click **New Cube**



5.2 Click **Next**

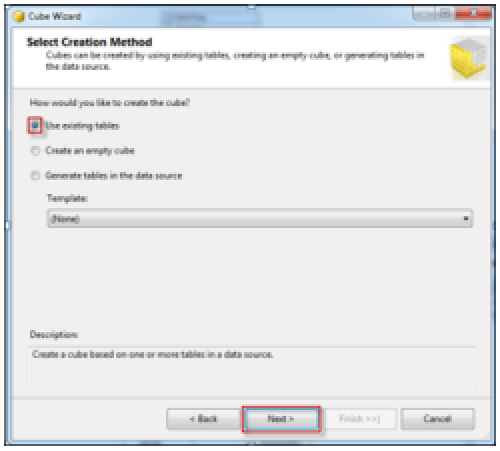

Cube Wizard

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

5.3 Select Option **Use existing Tables ->** Click **Next**


Cube Wizard

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?

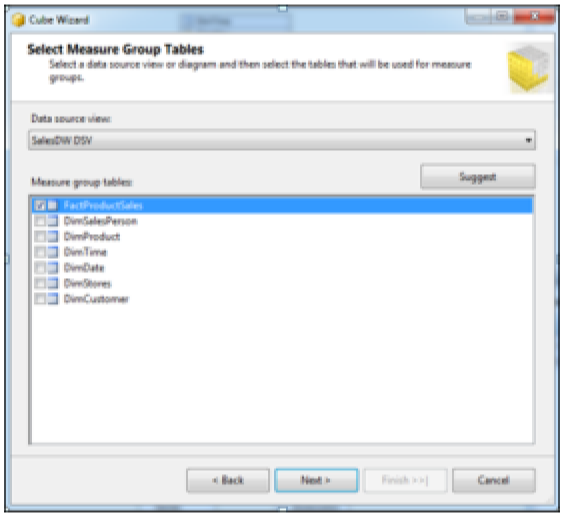
[● Use existing tables]
○ Create an empty cube
○ Generate tables in the data source

Template: (None)

Description:

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

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


Cube Wizard

Select Measure Group Tables

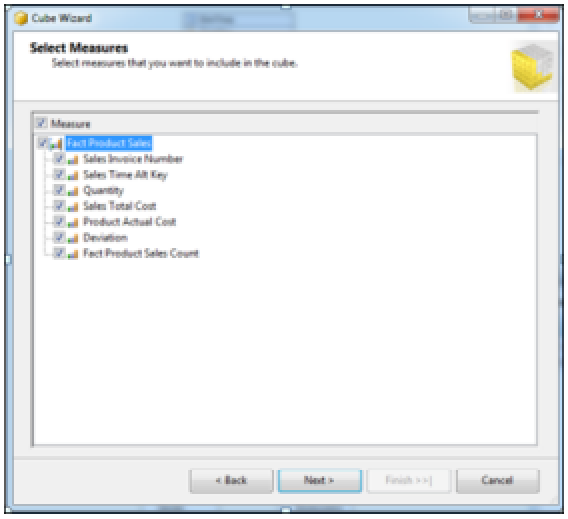
Select a data source view or diagram and then select the tables that will be used for measure groups.

Data source view: SalesDW DSV

Measure group tables:

[✓] FactProductSales
[ ] DimSalesPerson
[ ] DimProduct
[ ] DimTime
[ ] DimDate
[ ] DimStores
[ ] DimCustomer

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

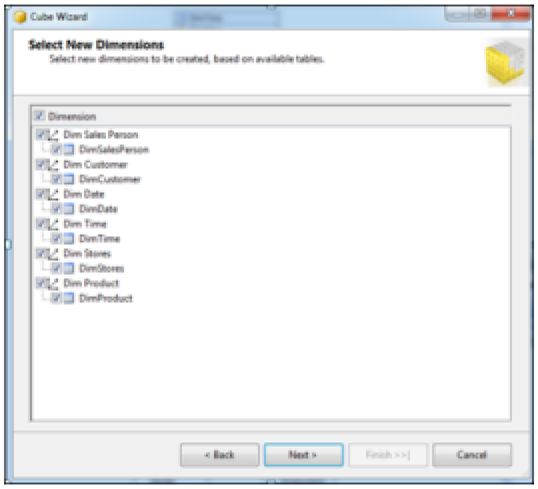


Select Measures

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

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

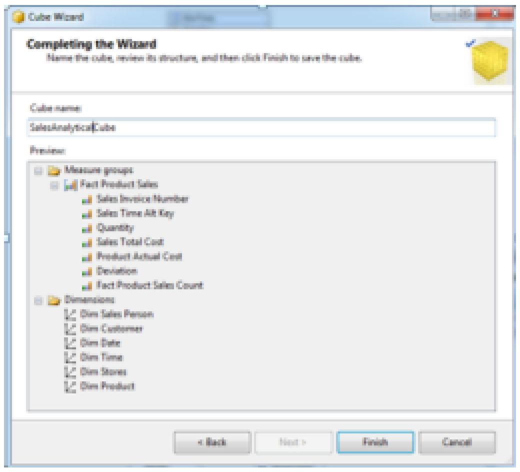


Select New Dimensions

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

5.7 Assign **Cube Name** (SalesAnalyticalCube) -> Click **Finish**



Completing the Wizard

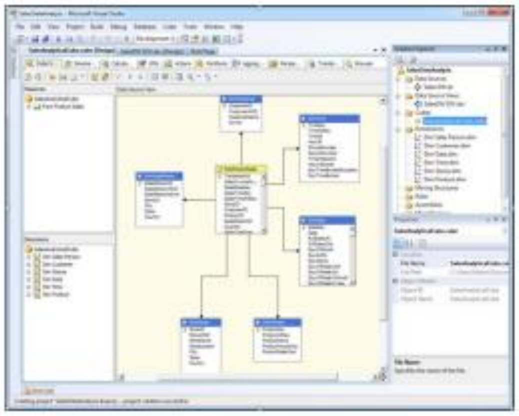
Name the cube, review its structure, and then click Finish to save the cube.

Cube name: SalesAnalyticalCube

Preview:

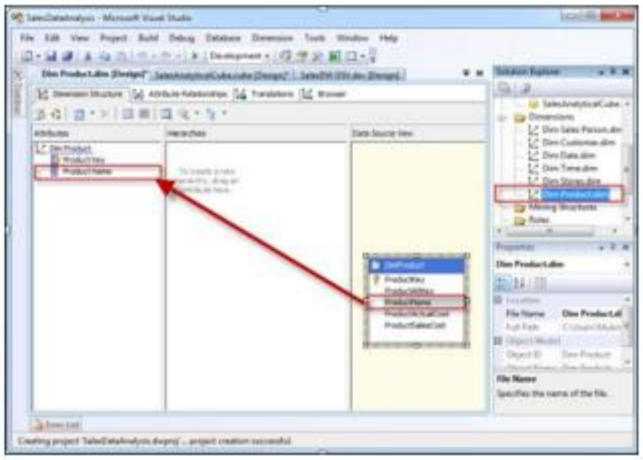
- Measure groups
  - Fact Product Sales
    - Sales Invoice Number
    - Sales Time Alt Key
    - Quantity
    - Sales Total Cost
    - Product Actual Cost
    - Deviation
    - Fact Product Sales Count
- Dimensions
  - Dim Sales Person
  - Dim Customer
  - Dim Date
  - Dim Time
  - Dim Stores
  - Dim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

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



Step 6: Dimension Modification

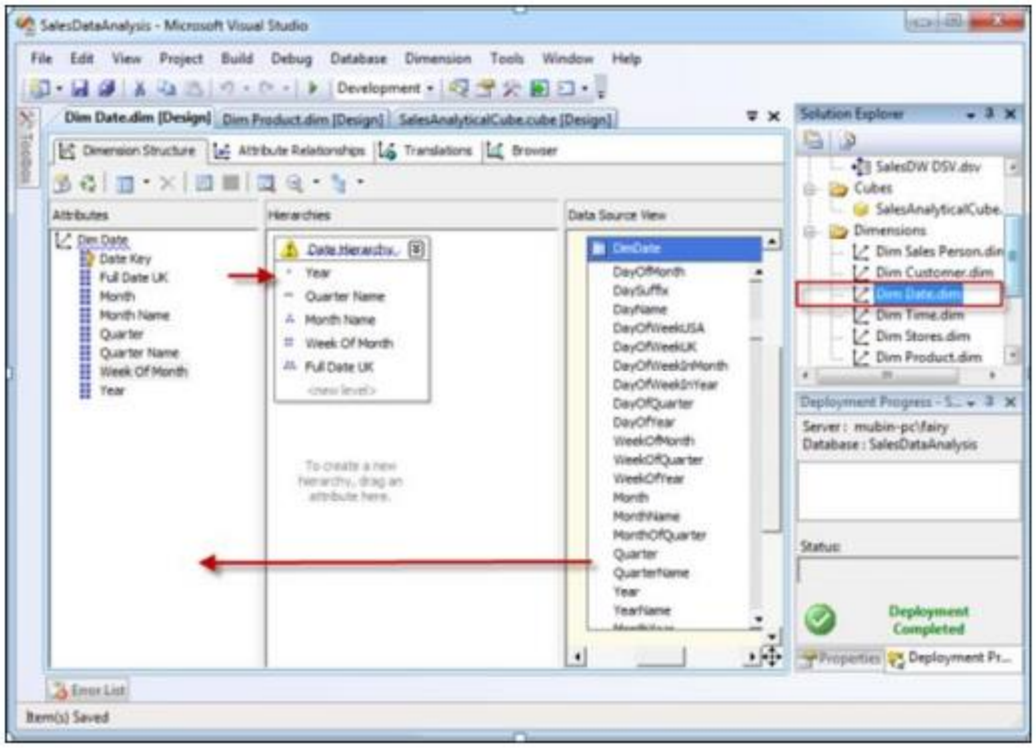
In Solution Explorer, double click on dimension **Dim Product ->** Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side.



Step 7: Creating Attribute Hierarchy In Date Dimension

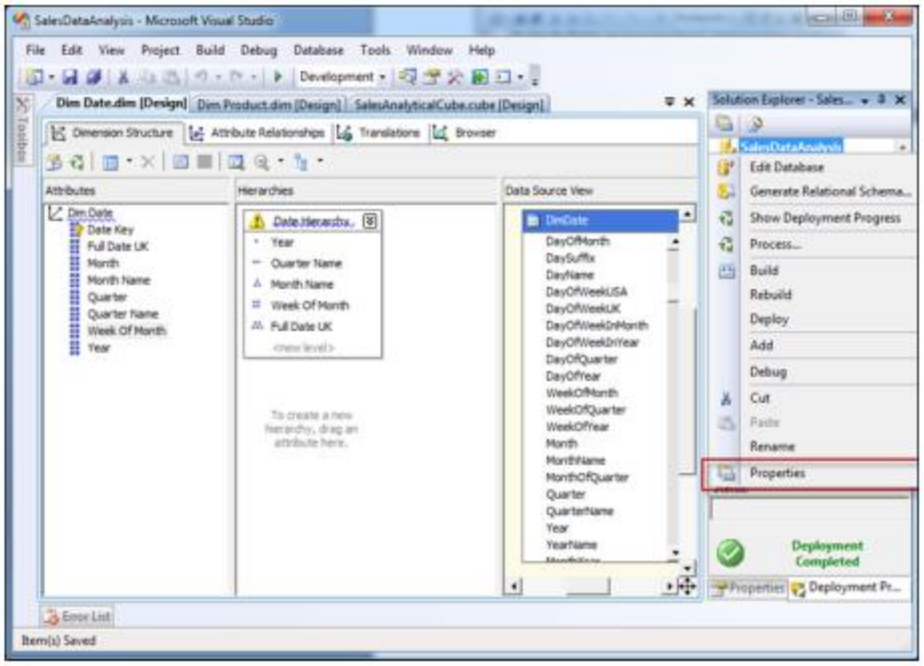
Double click On **Dim Date** dimension -> Drag and Drop Fields from Table shown in Data Source View to Attributes-> Drag and Drop attributes from leftmost pane of attributes to middle pane of Hierarchy.

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



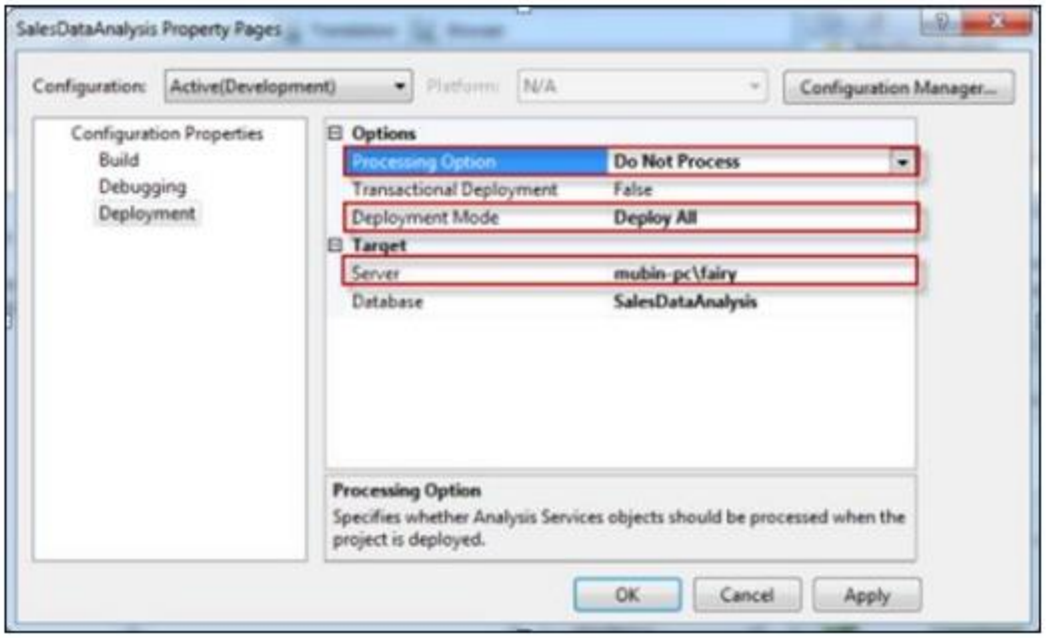
Step 8: Deploy the Cube

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



8.2 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

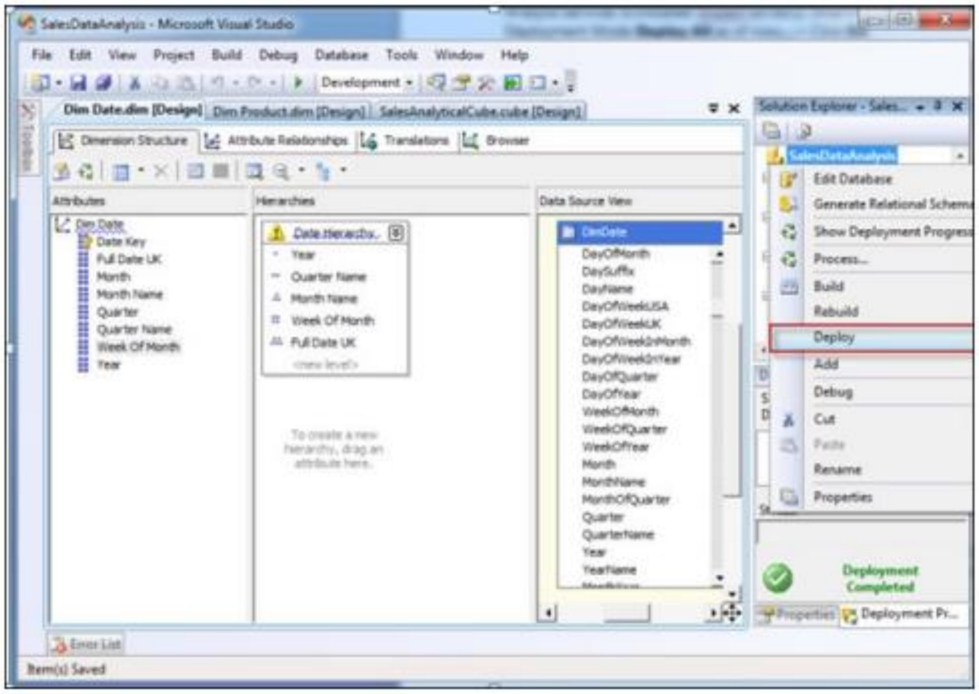
Processing Option	Do Not Process
Transactional Deployment	False
Deployment Mode	Deploy All

Target

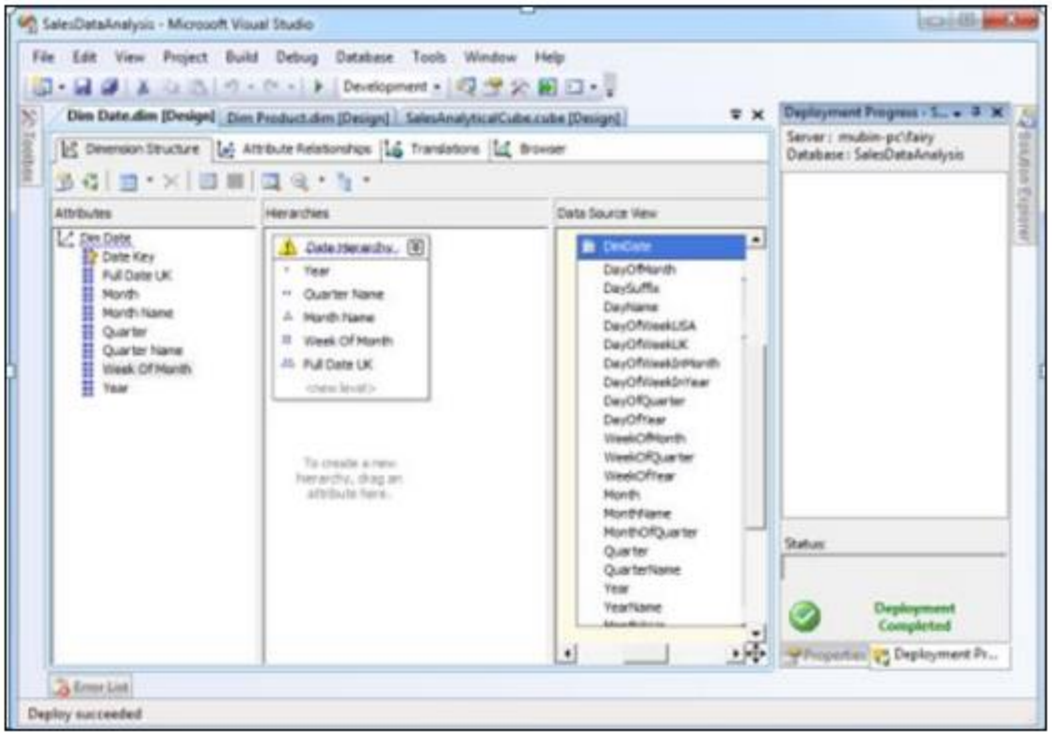
Server	mubin-pc\fairy
Database	SalesDataAnalysis

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

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

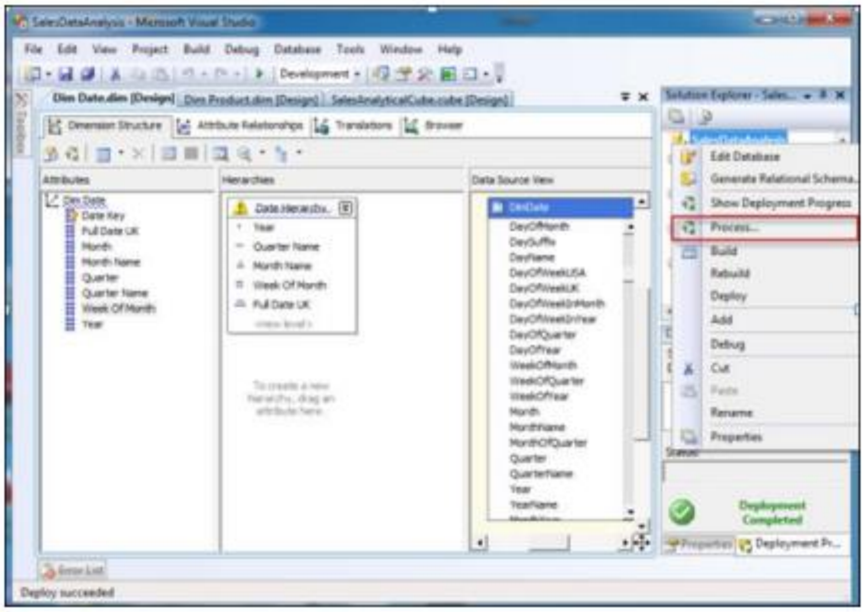


8.4 Once Deployment will finish, you can see the message **Deployment Completed** in deployment Properties.

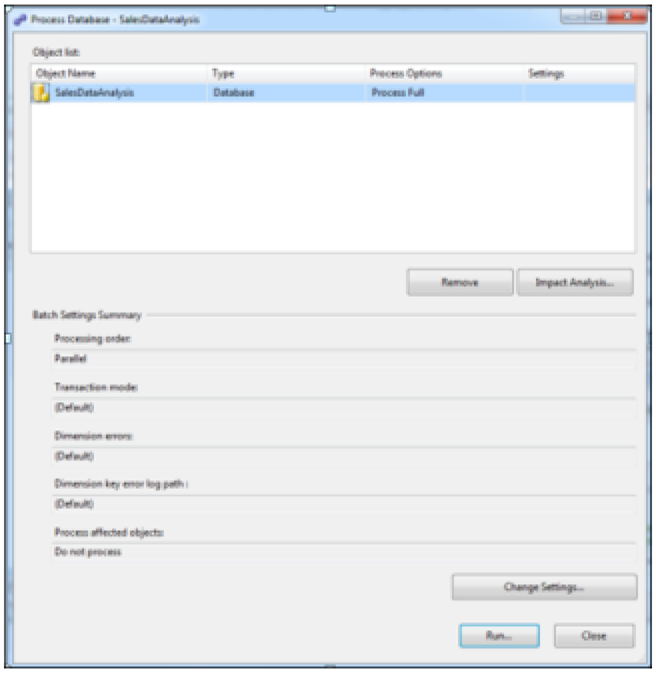

SalesDataAnalysis - Microsoft Visual Studio
File Edit View Project Build Debug Database Tools Window Help
Dim Date.dim [Design] Dim Product.dim [Design] SalesAnalyticalCube.cube [Design]
Dimension Structure Attribute Relationships Translations Browser
Attributes
Hierarchies
Data Source View
Deployment Progress
Server : mubin-pc\fairy
Database : SalesDataAnalysis
Status:
Deployment Completed
Deploy succeeded

Step 9: Process the Cube

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



9.2 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

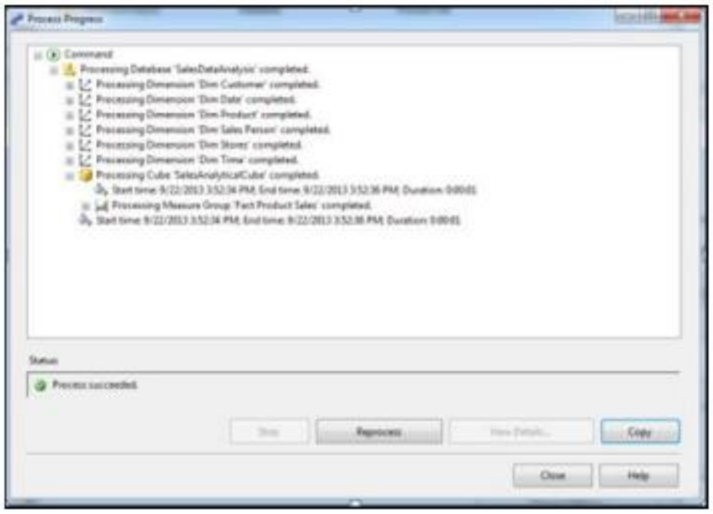
Transaction mode:
(Default)

Dimension errors:
(Default)

Dimension key error log path :
(Default)

Process affected objects:
Do not process

9.3 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.



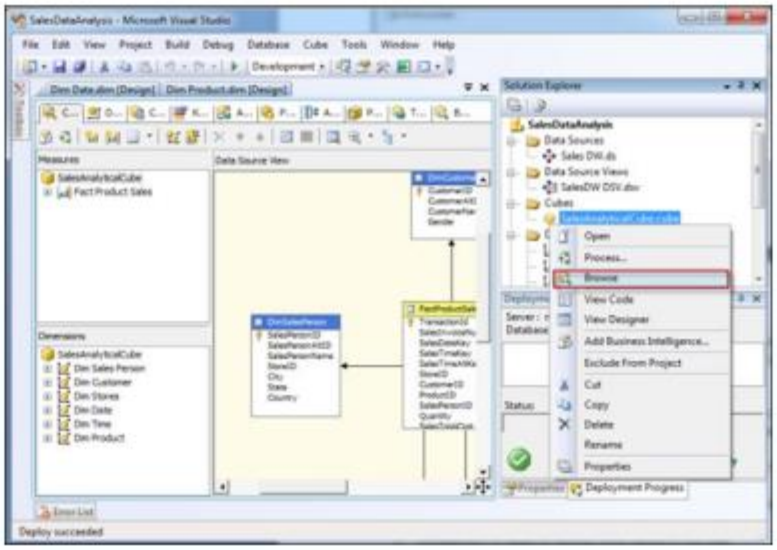
Process Progress

```
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 3:52:38 PM; Duration: 0:00:01
```

Status:
Process succeeded.

Step 10: Browse the Cube for Analysis

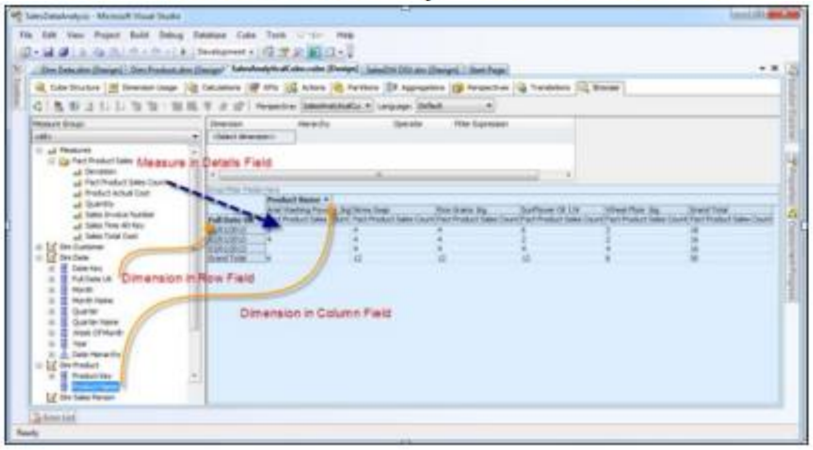
10.1 In Solution Explorer, right click on Cube Name (SalesDataAnalysisCube) -- > Click **Browse**



10.2 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 7: Practical Implementation of Decision Tree using R Tool

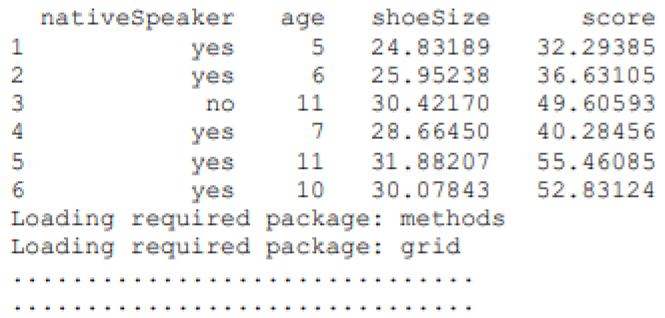
install.packages("party")

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 –

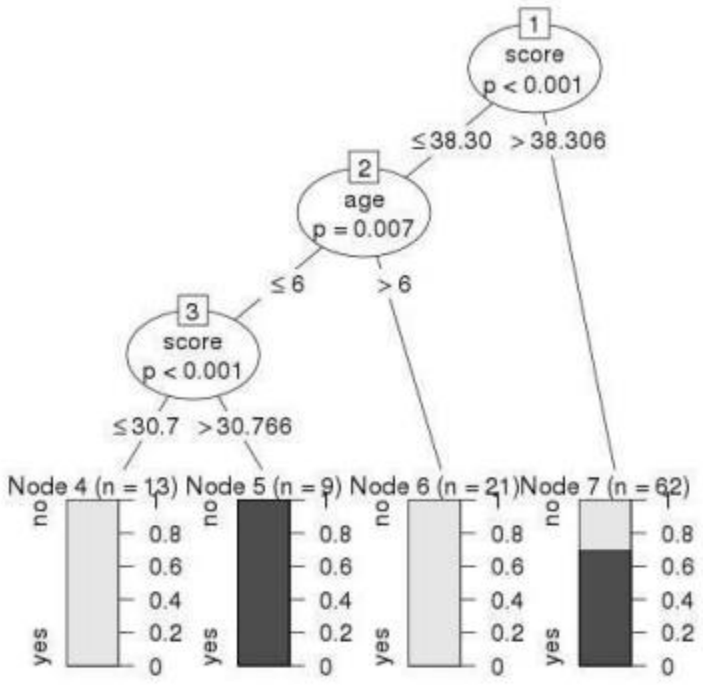


```
  nativeSpeaker   age   shoeSize      score
1           yes     5   24.83189   32.29385
2           yes     6   25.95238   36.63105
3            no    11   30.42170   49.60593
4           yes     7   28.66450   40.28456
5           yes    11   31.88207   55.46085
6           yes    10   30.07843   52.83124
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 9: 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.

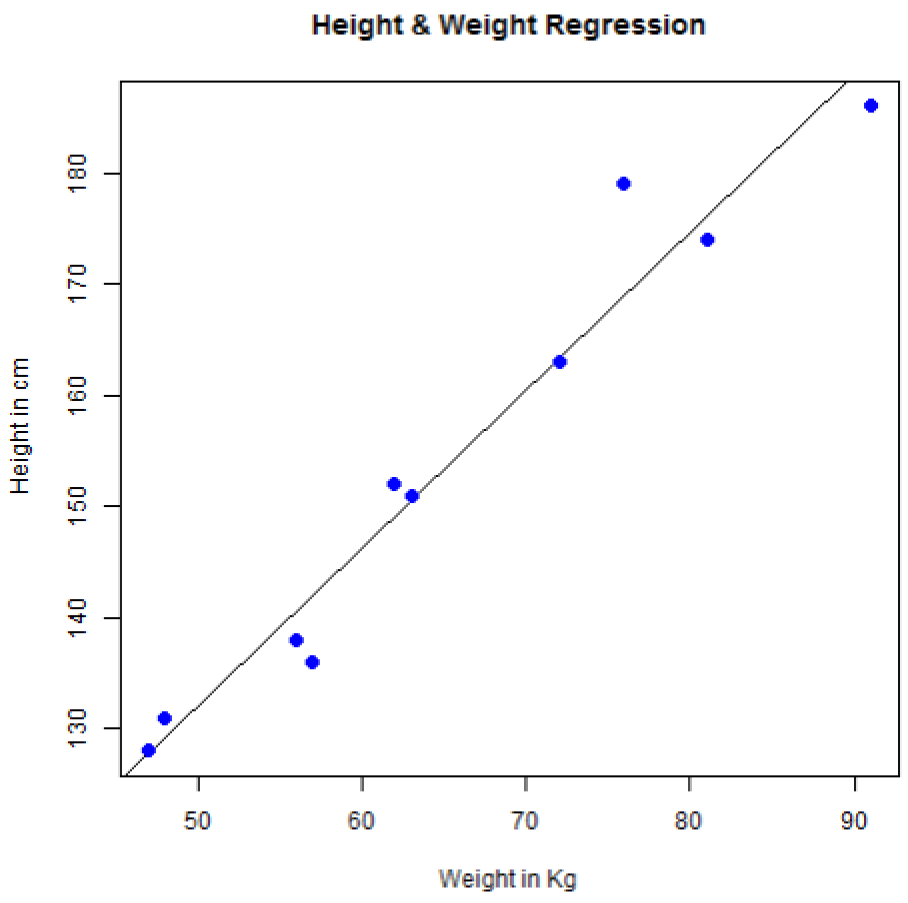
Predict the weight of new persons

```
# 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


Height & Weight Regression
Height in cm
Weight in Kg
130
140
150
160
170
180
50
60
70
80
90

Practical 10: 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.

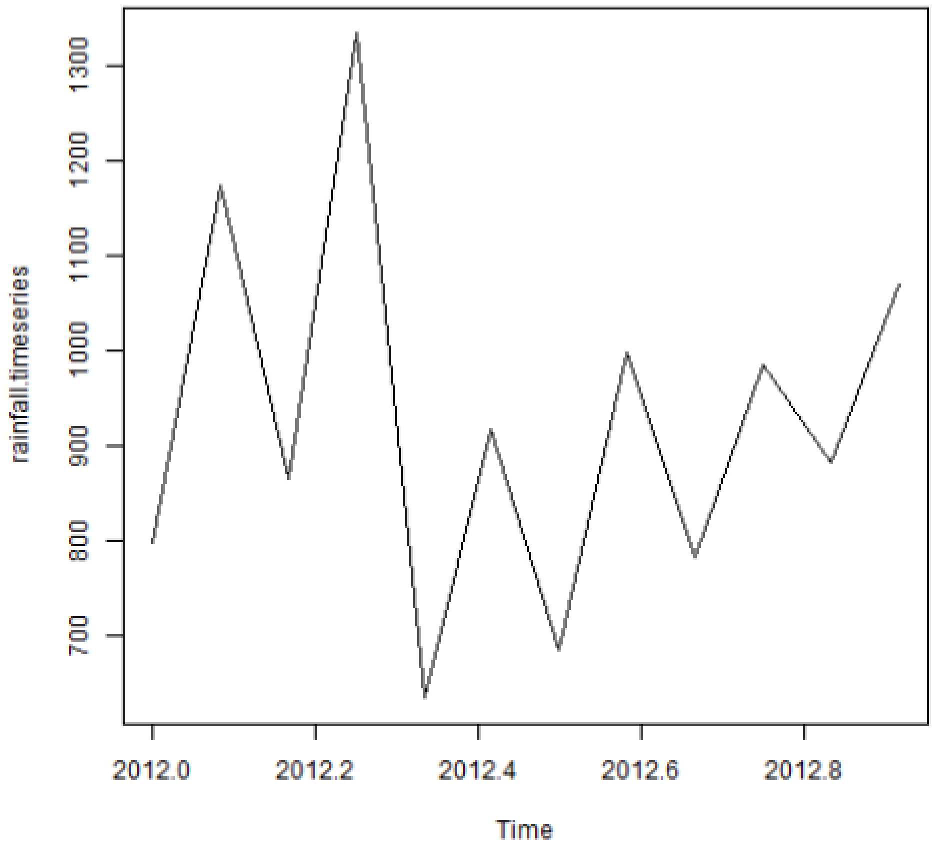
Except the parameter "data" all other parameters are optional.

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
 2012   799.0  1174.8  865.1  1334.6   635.4   918.5   685.5   998.6   784.2
          Oct     Nov     Dec
 2012   985.0   882.8 1071.0
```


rainfall.timeseries
Time