

**BUNTS SANGHA'S
S.M. SHETTY COLLEGE OF SCIENCE, COMMERCE &
MANAGEMENT STUDIES, POWAI-400076**



PRACTICAL JOURNAL

PROGRAMME: _____

SEMESTER: _____

COURSE: _____

SUBMITTED BY: _____

ROLL NUMBER: _____

**SUBMITTED TO
DEPARTMENT OF INFORMATION TECHNOLOGY**

Subject in charge

External Examiner

Coordinator

**For Partial Fulfillment for Degree of
Bachelor of Science (*Information Technology*) 2019 - 2020**

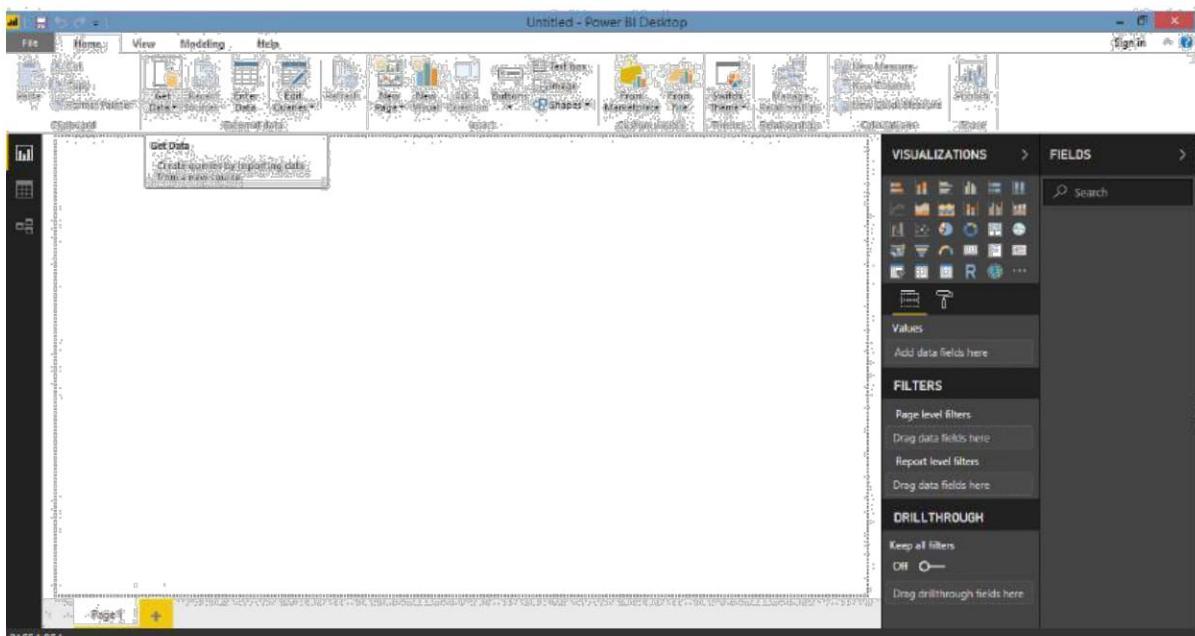
INDEX

PRACTICAL NO.	PRACTICAL NAME	DATE	SIGN
1	Import the legacy data from different sources such as (Excel , SqlServer, Oracle etc.) and load in the target system. (You can download sample database such as Adventureworks, Northwind, foodmart etc.)		
2	Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver.		
3	a. Create the Data staging area for the selected database. b. Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.		
4	a.Create the ETL map and setup the schedule for execution. b. Execute the MDX queries to extract the data from the datawarehouse.		
5	a. Import the datawarehouse data in Microsoft Excel and create the Pivot table and Pivot Chart. b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis.		
6	Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the datawarehouse data.		
7	Perform the data classification using classification algorithm.		
8	Perform the data clustering using clustering algorithm.		
9	Perform the Linear regression on the given datawarehouse data.		
10	Perform the logistic regression on the given datawarehouse data.		

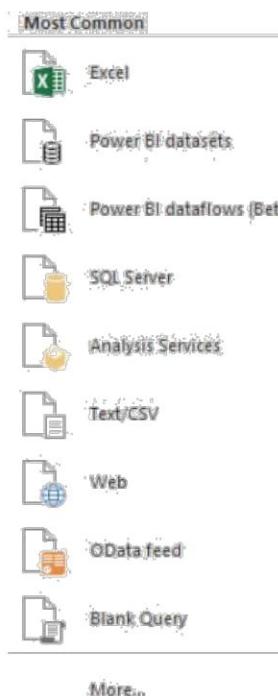
PRACTICAL 1

Import the legacy data from different sources such as (Excel , SqlServer, Oracle etc.) and load in the target system. (You can download sample database such as Adventureworks, Northwind, foodmart etc.)

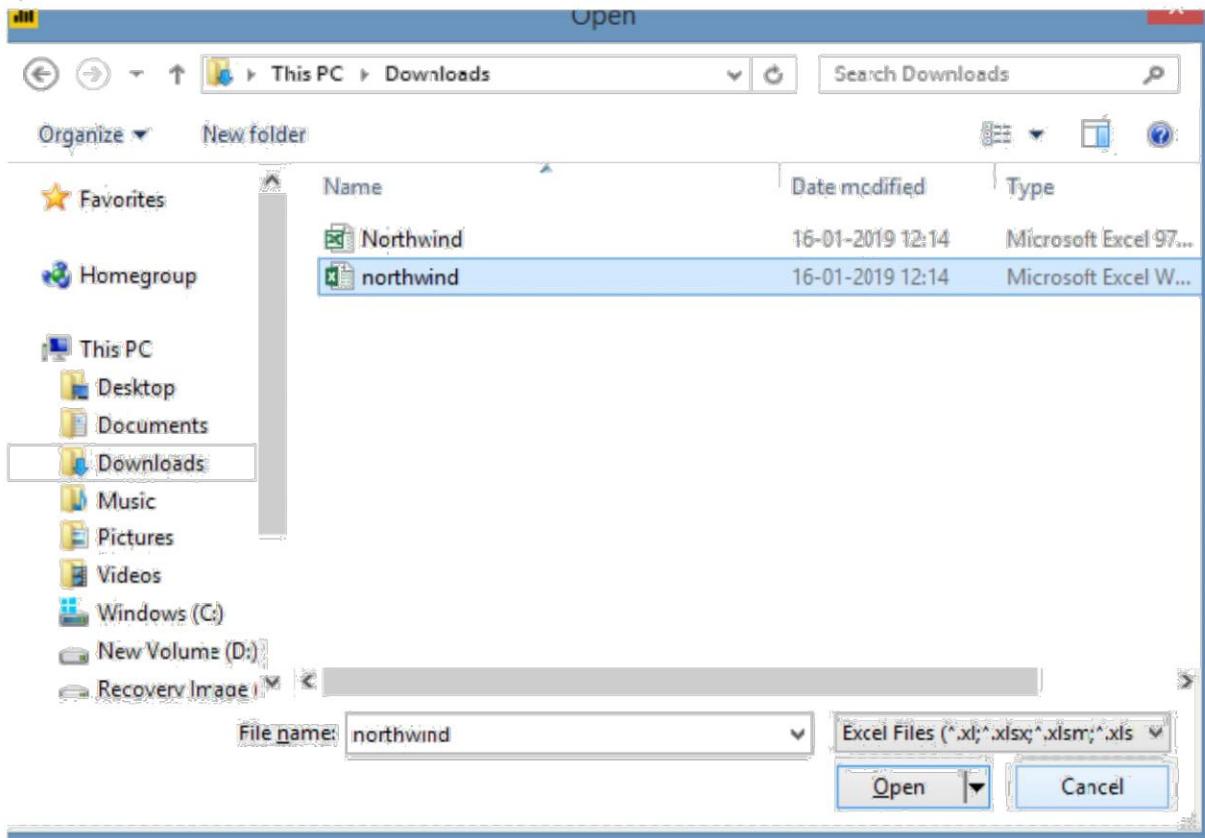
Step 1: Open Power BI



Step 2: Click on Get data following list will be displayed → select Excel



Step 3: Select required file and click on Open, Navigator screen appears



Step 4: Select



No items selected for preview

file and click on edit

Navigator

The screenshot shows the Power BI Navigator interface. On the left, there is a tree view of data sources: 'northwind.xlsx [2]' and 'Table_northwind.accdb'. The 'Table_northwind.accdb' node is expanded, showing its tables: 'Customer' and 'Sheet1'. On the right, a preview of the 'Customer' table from 'Table_northwind.accdb' is displayed. The table has columns: ID, Customers.Company, LastName, FirstName, and EmailAddress. The preview shows 22 rows of customer data. Below the preview, a note says: 'The data in the preview has been truncated due to size limits.' At the bottom are 'Load', 'Edit', and 'Cancel' buttons.

ID	Customers.Company	LastName	FirstName	EmailAddress
2	Customer A	Bedevere	Anna	
3	Customer B	Gracioso Solsona	Antonio	
4	Customer C	Axen	Thorhaf	
5	Customer D	Lee	Christina	
6	Customer E	O'Donnell	Martin	
7	Customer F	Pérez-Osseta	Francisco	
8	Customer G	Xie	Ming-Yang	
9	Customer H	Andersen	Elizabeth	
10	Customer I	Mortensen	Sven	
11	Customer J	Wacker	Roland	
12	Customer K	Kruecke	Peter	
13	Customer L	Edwards	John	
14	Customer M	Ludick	Andre	
15	Customer N	Grilo	Carlos	
16	Customer O	Kuprkova	Helena	
17	Customer P	Goldschmidt	Daniel	
18	Customer Q	Bagel	Jean Philippe	
19	Customer R	Autier Miconi	Catherine	
20	Customer S	Eggerer	Alexander	
21	Customer T	Li	George	
22	Customer U	Tham	Bernard	

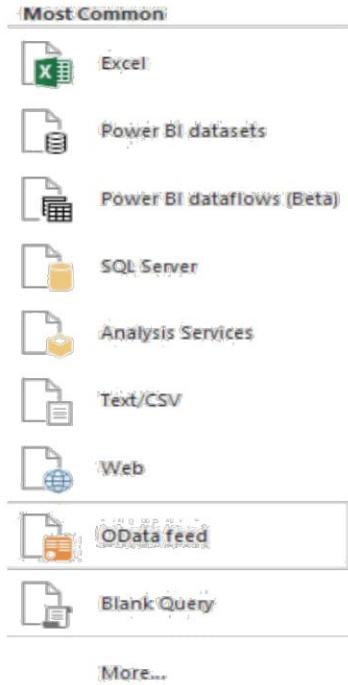
Step 5: Power query editor appears

The screenshot shows the Power Query Editor window titled 'Untitled - Power Query Editor'. The ribbon menu includes 'File', 'Home', 'Transform', 'Add Column', 'View', and 'Help'. The 'Transform' tab is selected. The main area displays the 'Customer' table from 'Table_northwind.accdb' with 26 rows. The 'QUERY SETTINGS' pane on the right shows the following details:

- PROPERTIES**: Name: Table_northwind.accdb
- APPLIED STEPS**:
 - Source
 - Navigation
 - > Changed Type

At the bottom of the editor, it says 'PREVIEW DOWNLOADED AT 12'.

Step 6: Again, go to Get Data and select OData feed



More...

Step 7: Paste url as <http://services.odata.org/V3/Northwind/Northwind.svc/> Click on ok



Step 8: Select orders table And click on edit Note: If you just want to see preview you can just click on table name without clicking on checkbox Click on edit to view table

Navigator

Display Options ▾

- <http://services.odata.org/V3/Northwind/No...>
- [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](#)
- [Orders](#)
- [Orders_Qries](#)
- [Product_Sales_for_1997](#)
- [Products](#)
- [Products_Above_Average_Prices](#)
- [Products_by_Categories](#)
- [Regions](#)

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	3	04-07-1996 00:00:00	01-08-1996 00:00:00
10249	TOMSP	6	05-07-1996 00:00:00	16-08-1996 00:00:00
10250	HANAR	4	05-07-1996 00:00:00	05-08-1996 00:00:00
10251	VICTE	3	05-07-1996 00:00:00	05-08-1996 00:00:00
10252	SUPRD	4	09-07-1996 00:00:00	06-08-1996 00:00:00
10253	HANAR	3	10-07-1996 00:00:00	24-07-1996 00:00:00
10254	CHOPS	3	11-07-1996 00:00:00	08-08-1996 00:00:00
10255	RICSU	9	12-07-1996 00:00:00	09-08-1996 00:00:00
10256	WILLI	3	15-07-1996 00:00:00	12-08-1996 00:00:00
10257	HILAA	4	15-07-1996 00:00:00	15-08-1996 00:00:00
10258	ERNSH	1	17-07-1996 00:00:00	14-08-1996 00:00:00
10259	CENTC	4	18-07-1996 00:00:00	15-08-1996 00:00:00
10260	OTTIK	4	19-07-1996 00:00:00	16-08-1996 00:00:00
10261	QUEDE	6	22-07-1996 00:00:00	16-08-1996 00:00:00
10262	RATTC	8	22-07-1996 00:00:00	19-08-1996 00:00:00
10263	ERNSH	9	23-07-1996 00:00:00	20-08-1996 00:00:00
10264	FOLKO	5	24-07-1996 00:00:00	21-08-1996 00:00:00
10265	BLONP	2	25-07-1996 00:00:00	22-08-1996 00:00:00
10266	WARTH	3	26-07-1996 00:00:00	06-09-1996 00:00:00
10267	FRANK	4	29-07-1996 00:00:00	26-08-1996 00:00:00
10268	GROSR	8	30-07-1996 00:00:00	27-08-1996 00:00:00
10269	WHITC	5	31-07-1996 00:00:00	14-08-1996 00:00:00
10270	WARTI	1	01-08-1996 00:00:00	29-08-1996 00:00:00

Select Related Tables Load Edit Cancel

Untitled - Power Query Editor

Transform Add Column View Help

Recent Data: accdb

Data source settings Manage Parameters Refresh Preview Advanced editor Properties Choose Columns Remove Columns Keep Rows Remove Rows Split Column Group By Replace values Data Type: Whole Number Use First Row as Headers

Query: Orders

QUERY SETTINGS

Properties Name: Orders All Properties

Applied Steps Source: Navigation

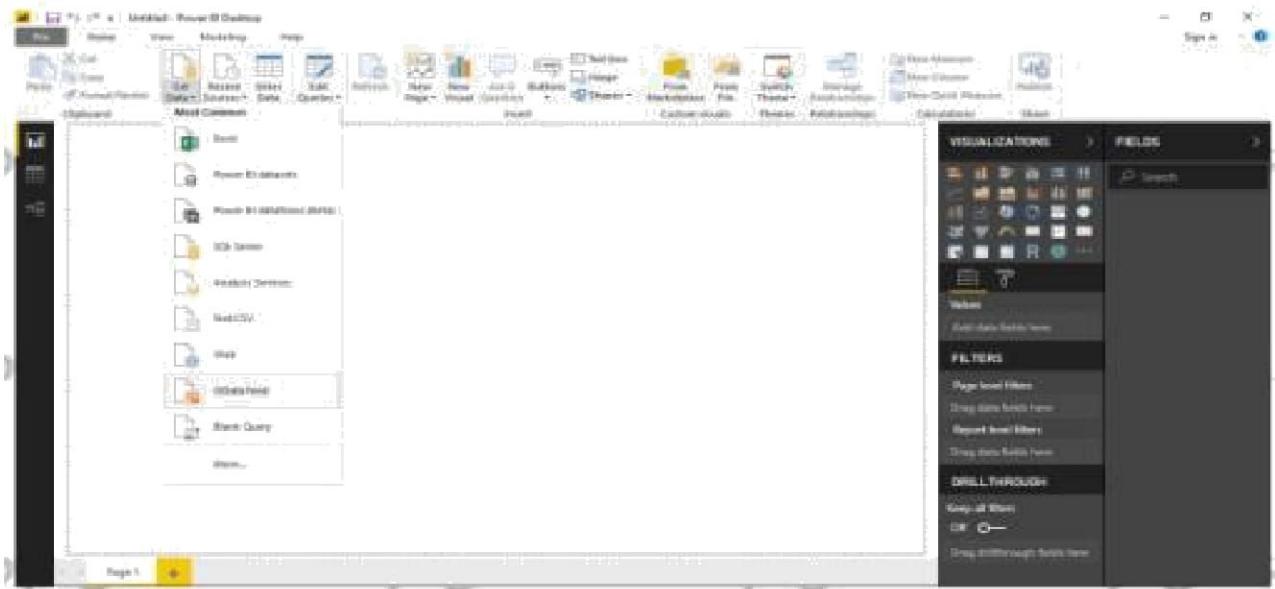
PREVIEW DOWNLOADED AT 12:00

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	Shipped
10248	VINET	3	04-07-1996 00:00:00	01-08-1996 00:00:00	16:00
10249	TOMSP	6	05-07-1996 00:00:00	16-08-1996 00:00:00	10:00
10250	HANAR	4	05-07-1996 00:00:00	05-08-1996 00:00:00	12:00
10251	VICTE	3	05-07-1996 00:00:00	05-08-1996 00:00:00	15:00
10252	SUPRD	4	09-07-1996 00:00:00	06-08-1996 00:00:00	21:00
10253	HANAR	3	10-07-1996 00:00:00	24-07-1996 00:00:00	15:00
10254	CHOPS	3	11-07-1996 00:00:00	08-08-1996 00:00:00	23:00
10255	RICSU	9	12-07-1996 00:00:00	09-08-1996 00:00:00	15:00
10256	WILLI	3	15-07-1996 00:00:00	12-08-1996 00:00:00	17:00
10257	HILAA	4	15-07-1996 00:00:00	13-08-1996 00:00:00	22:00
10258	ERNSH	1	17-07-1996 00:00:00	14-08-1996 00:00:00	23:00
10259	CENTC	4	18-07-1996 00:00:00	15-08-1996 00:00:00	25:00
10260	OTTIK	4	19-07-1996 00:00:00	16-08-1996 00:00:00	29:00
10261	QUEDE	6	22-07-1996 00:00:00	16-08-1996 00:00:00	20:00
10262	RATTC	8	22-07-1996 00:00:00	19-08-1996 00:00:00	25:00

Practical 2

2(A) Perform the Extraction Transformation and loading (ETL) process to construct the database in the Power BI

Step1: Open Power BI, Click on Get Data --> OData Feed



Paste Url: <http://services.odata.org/v3/Northwind/Northwind.svc/>

And Click OK



Step 2: Click on Check Box of Products table and then click on Edit

Navigator

Display Options: http://servicesodata.org/V3/Northwind/No...

- 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_Jubtotals
- Orders
- Orders_Gales
- Product_Sales_for_1997
- Products
- Products_Above_Average_Prices
- Products_By_Categories
- Regions

Select Related Tables

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit
1	Cogi	1	2	20 boxes x 20 bags
2	Chang	1	2	24 - 10 oz bottles
3	Aniseed Syrup	3	2	12 - 500 ml bottles
4	Chef Anton's Cajun Seasoning	2	2	48 - 8 oz jars
5	Chef Anton's Gumbo Mix	2	2	36 boxes
6	Grandma's Boysenberry Spread	3	2	12 - 1 lb jars
7	Uncle Bob's Organic Dried Pears	3	7	32 - 12 oz jars
8	Norwegian Blueberry Sauce	5	2	18 - 500 g pkgs
9	Mild Kobe Nuru	4	2	12 - 200 ml jars
10	Burn	4	2	3 kg pkg
11	Gusto Cabiilles	5	4	20 - 500 g pkgs
12	Queso Manchego La Pastora	5	4	2 kg box
13	Konbu	6	2	40 - 100 g pkgs
14	Tofu	6	2	30 - 250 ml bottles
15	Gehan Shoyu	6	2	

- 1) Remove other columns to only display columns of interest 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).

1 Unitled - Power Query Editor

File Home Insert Add Column View Help

Queries [1] Products

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit
1	Cogi	1	2	20 boxes x 20 bags
2	Chang	1	2	24 - 10 oz bottles
3	Aniseed Syrup	3	2	12 - 500 ml bottles
4	Chef Anton's Cajun Seasoning	2	2	48 - 8 oz jars
5	Chef Anton's Gumbo Mix	2	2	36 boxes
6	Grandma's Boysenberry Spread	3	2	12 - 1 lb jars
7	Uncle Bob's Organic Dried Pears	3	7	32 - 12 oz jars
8	Norwegian Blueberry Sauce	5	2	18 - 500 g pkgs
9	Mild Kobe Nuru	4	2	12 - 200 ml jars
10	Burn	4	2	3 kg pkg
11	Gusto Cabiilles	5	4	20 - 500 g pkgs
12	Queso Manchego La Pastora	5	4	2 kg box
13	Konbu	6	2	40 - 100 g pkgs
14	Tofu	6	2	30 - 250 ml bottles
15	Gehan Shoyu	6	2	

13 COLUMNS, 77 ROWS PREVIEW DOWNLOADED AT 00:53

QUERY SETTINGS

PROPERTIES Name Products

APPLIED STEPS Source

Select Remove Columns > Remove Other Columns from the ribbon, or rightclick on a column header and click Remove Other Columns

The screenshot shows the Power Query Editor interface with the 'Transform' tab selected. In the center, a table named 'Products' is displayed with columns: ProductID, ProductName, and QuantityPerUnit. A context menu is open over the 'QuantityPerUnit' column, with 'Remove Other Columns' highlighted. To the right, the 'QUERY SETTINGS' pane shows the query name 'Products' and the applied step 'Navigation'. The preview area at the bottom right shows the data has been reduced to four columns.

After selecting Remove Other Columns only selected four columns are displayed other columns are discarded

This screenshot shows the same Power Query Editor interface after the 'Remove Other Columns' step has been applied. The table now displays only three columns: ProductID, ProductName, and UnitsInStock. The 'QUERY SETTINGS' pane now includes an additional step: 'Removed Other Columns'. The preview area at the bottom right shows the data has been reduced to four columns.

2. Change the data type of the UnitsInStock column

- Select the UnitsInStock column.

a) Select the UnitsInStock column.

Check if the data type of selected column is a Whole number

The screenshot shows the Power Query Editor interface with the 'Products' query selected. The 'UnitsInStock' column is highlighted with a yellow selection bar. A red arrow points from the text above to the column header 'UnitsInStock'. The 'QUERY SETTINGS' pane on the right shows the query name 'Products' and the applied step 'Removed Other Columns'.

b) Select the Data Type drop-down button in the Home ribbon.

c) 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).

The screenshot shows the Power Query Editor interface with the 'Products' query selected. The 'UnitsInStock' column is highlighted with a yellow selection bar. The 'Data Type' dropdown menu is open, showing options like Decimal Number, Percentage, Date/Time, etc., with 'Whole Number' selected. The preview pane at the bottom right shows the data has been converted to whole numbers.

After clicking on Whole number, you can see the changed Datatype in column header of UnitsInStock.

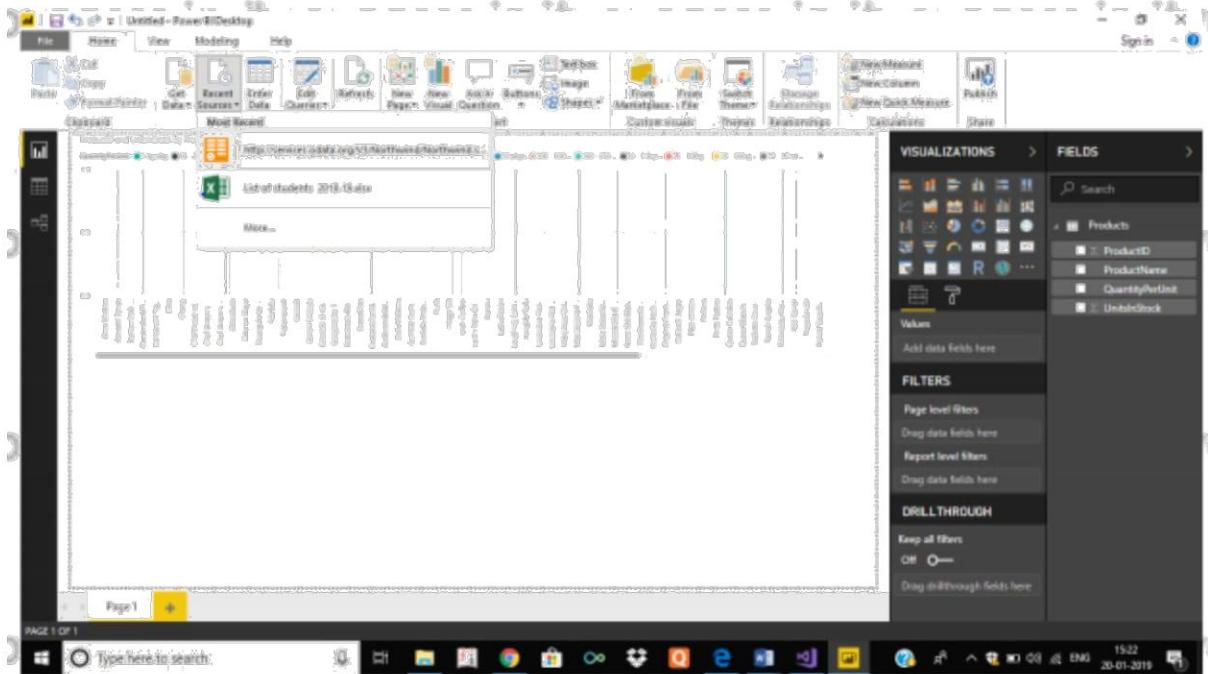
The screenshot shows the Power Query Editor interface. On the left, there's a 'Queries [1]' list with a single item named 'Products'. The main area displays a table with four columns: 'ProductID', 'QuantityPerUnit', 'ProductName', and 'Unordered'. The 'Unordered' column has a red circle around its header. To the right, the 'QUERY SETTINGS' pane is open, showing the 'Name' is set to 'Products' and the 'Applied Steps' list includes 'Source', 'Navigation', 'Removed Other Columns', and 'Changed Type'. At the bottom right, it says 'PREVIEW DOWNLOADED AT 00:53'.

After above step, close query editor and click on Yes to save changes.

Now you can view fields of Products table on right side, check all the fields of table to get representation in charts form.

The screenshot shows the Power BI desktop interface. On the right side, the 'FIELDS' pane is open, displaying the fields from the 'Products' table: ProductID, ProductName, QuantityPerUnit, and UnitStock. The rest of the screen shows the Power BI canvas with various visualizations and data sources.

3. Expand the Orders table Once You have loaded a data source, you can click on Recent Sources to select desired table (Orders).



After selecting the URL, Navigator window will appear from which you can select Orders table. Click on Edit.

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	04-07-1996 00:00:00	01-08-1996 00:00:00
10249	TOMSP	6	05-07-1996 00:00:00	16-08-1996 00:00:00
10250	HANAR	4	08-07-1996 00:00:00	05-08-1996 00:00:00
10251	VICTE	3	08-07-1996 00:00:00	03-08-1996 00:00:00
10252	SUPRD	4	09-07-1996 00:00:00	06-08-1996 00:00:00
10253	HANAR	5	10-07-1996 00:00:00	24-07-1996 00:00:00
10254	CHOPS	5	11-07-1996 00:00:00	08-08-1996 00:00:00
10255	RICSU	9	12-07-1996 00:00:00	09-08-1996 00:00:00
10256	WELLI	3	15-07-1996 00:00:00	12-08-1996 00:00:00
10257	HILAA	4	16-07-1996 00:00:00	13-08-1996 00:00:00
10258	ERNSH	1	17-07-1996 00:00:00	14-08-1996 00:00:00
10259	CENTC	4	18-07-1996 00:00:00	15-08-1996 00:00:00
10260	OTTIK	4	19-07-1996 00:00:00	16-08-1996 00:00:00
10261	QUEDE	4	19-07-1996 00:00:00	16-08-1996 00:00:00
10262	RATTC	8	22-07-1996 00:00:00	19-08-1996 00:00:00
10263	ERNSH	9	23-07-1996 00:00:00	20-08-1996 00:00:00
10264	POLKO	6	24-07-1996 00:00:00	21-08-1996 00:00:00
10265	BLOWP	2	25-07-1996 00:00:00	22-08-1996 00:00:00
10266	WARTH	3	26-07-1996 00:00:00	06-08-1996 00:00:00
10267	FRANK	4	29-07-1996 00:00:00	26-08-1996 00:00:00
10268	GROSR	8	30-07-1996 00:00:00	27-08-1996 00:00:00
10269	WHITC	5	31-07-1996 00:00:00	14-08-1996 00:00:00
10270	WARTH	1	01-08-1996 00:00:00	29-08-1996 00:00:00

Query Editor Window will appear

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.

The screenshot shows the Power Query Editor interface. A context menu is open over the 'Order_Details' column, specifically the header cell. The 'Expand' option is selected, and a sub-menu is displayed. In the 'Select All Columns' section, three specific columns are checked: 'ProductID', 'UnitPrice', and 'Quantity'. The 'OK' button at the bottom of the dialog is highlighted in yellow. The main query grid shows 18 columns and 830 rows. The 'Properties' pane on the right shows the query name is 'Orders'. The 'Applied Steps' pane shows a single step named 'Navigation'.

After clicking on OK following screen appears with combined columns

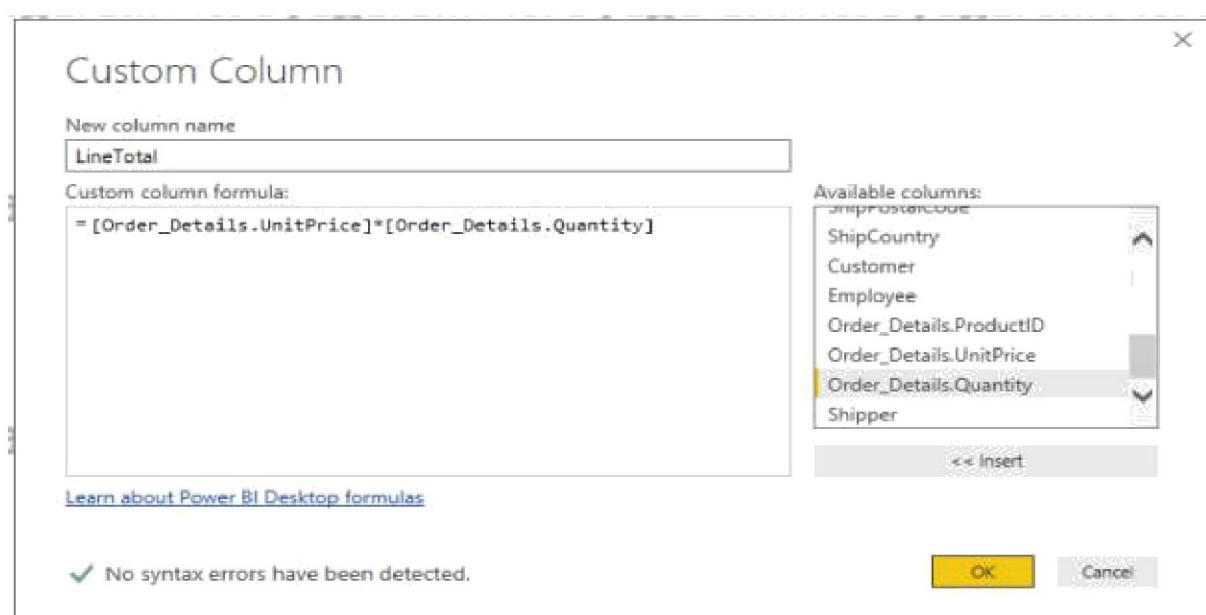
The screenshot shows the Power Query Editor after the expansion step has been applied. The 'Order_Details' column now contains three sub-columns: 'ProductID', 'UnitPrice', and 'Quantity', which are combined into a single column. The main query grid shows 20 columns and 199+ rows. The 'Properties' pane on the right shows the query name is 'Orders'. The 'Applied Steps' pane shows two steps: 'Navigation' and 'Expanded Order_Details'.

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:

- a) In the Add Column ribbon tab, click Add Custom Column.

The screenshot shows the Power Query Editor interface. The 'Queries [2]' pane on the left lists 'Products' and 'Orders'. The main area displays a preview of the 'Orders' query, which has been expanded. The preview table includes columns for OrderID, ProductID, UnitPrice, Quantity, and Shipper. The 'Applied Steps' pane on the right shows the step 'Expanded Order_Details'.

- b) In the Custom Column dialog box, in the Custom Column Formula textbox, enter [Order_Details.UnitPrice] * [Order_Details.Quantity] by selecting from available columns and click on insert for each column.
- c) In the New column name textbox, enter LineTotal.
- d) Click OK.



The screenshot shows the Power Query Editor interface with the 'Orders' query selected. The 'LineTotal' column is highlighted in yellow. The 'QUERY SETTINGS' pane on the right shows the 'Name' as 'Orders' and the 'Applied Steps' list containing 'Source', 'Navigation', 'Expanded Order_Details', and 'Added Custom'. The preview pane at the bottom right indicates it was downloaded at 9053.

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.

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

The screenshot shows the Power Query Editor interface with the 'Orders' query selected. The 'LineTotal' column is now positioned to the left of the 'ShipCountry' column. The 'QUERY SETTINGS' pane on the right shows the 'Name' as 'Orders' and the 'Applied Steps' list containing 'Source', 'Navigation', 'Expanded Order_Details', 'Added Custom', and 'Reordered Columns'. The preview pane at the bottom right indicates it was downloaded at 9053.

- 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.

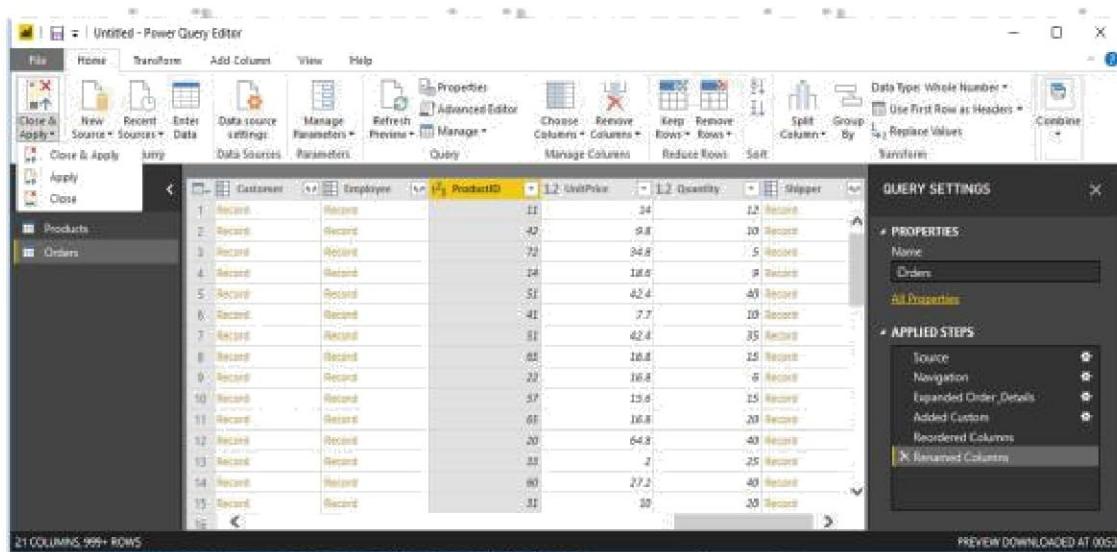
6. Combine the Products and Total Sales queries Power BI Desktop does not require you to combine queries to report on them.

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.

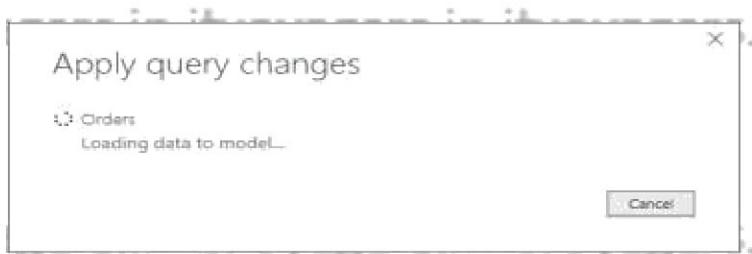
In this task, you confirm that a relationship is established in Power BI Desktop between the Products and Total Sales queries .

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 & Apply.

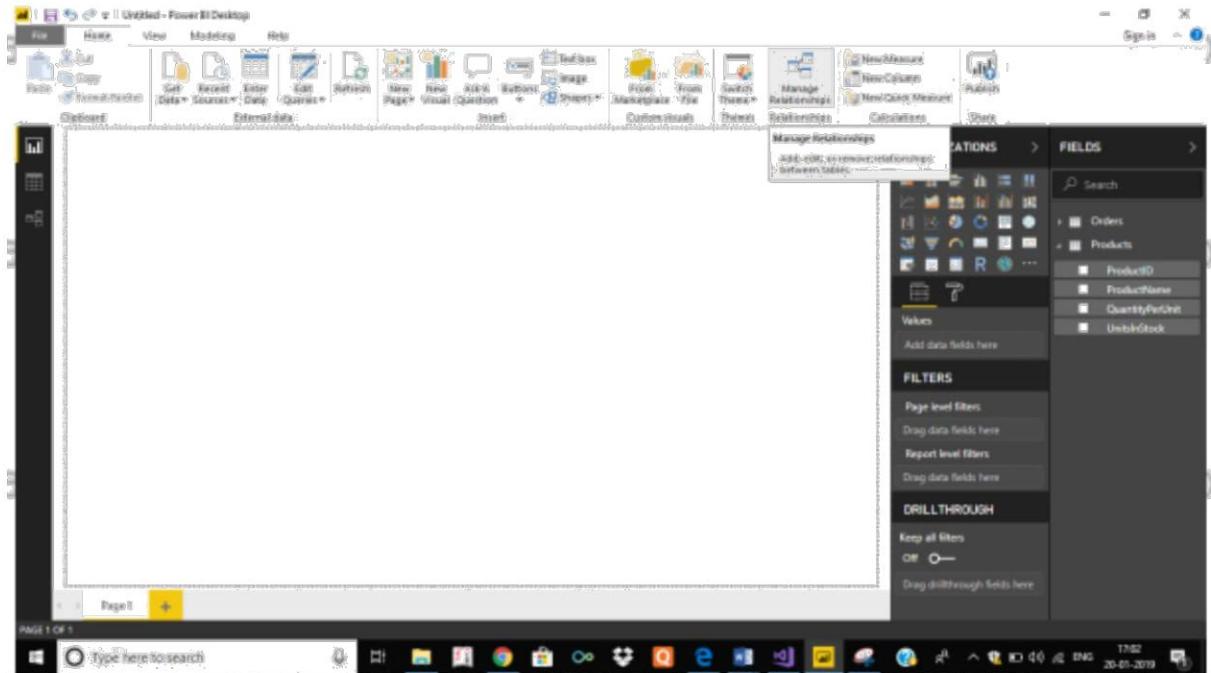


The screenshot shows the Power Query Editor interface with two queries loaded: 'Products' and 'Orders'. The 'Products' query is currently selected. The ribbon at the top has the 'Home' tab selected, with the 'Close & Apply' button highlighted. The 'Applied Steps' pane on the right shows the steps taken to process the query, including 'Source', 'Navigation', 'Expanded Order Details', 'Added Custom', and 'Reordered Columns'. A preview of the data is visible in the main grid, showing columns like ProductID, UnitPrice, Quantity, and Shipped.

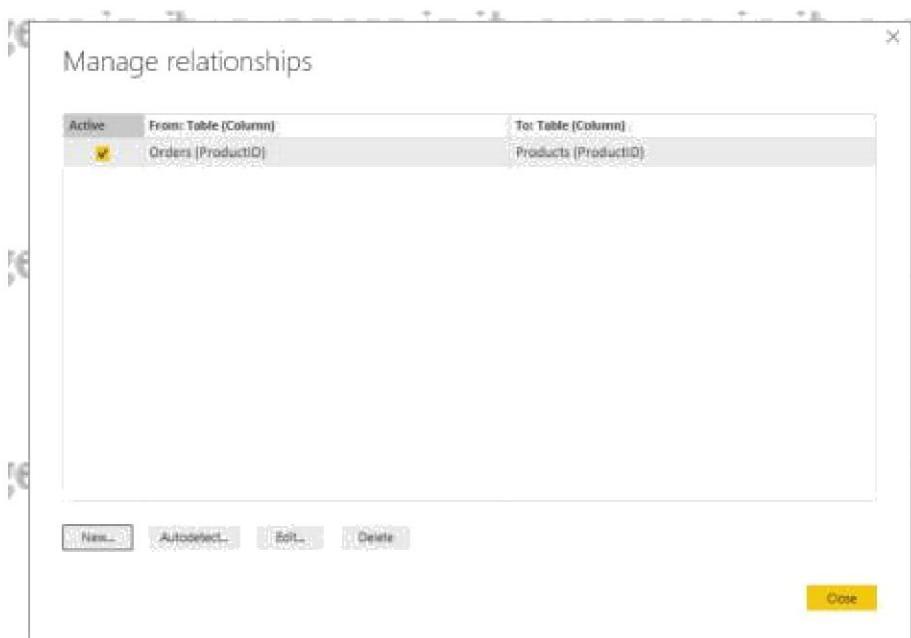
Step 2: Power BI Desktop loads the data from the two queries.



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



Step 4. Select the New... button



Step 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.

Products									
ProductID	QuantityPerUnit	ProductName	UnitsInStock	Discontinued	UnitPrice	UnitCost	ReorderLevel	SupplierID	CategoryID
1	10 boxes x 20 bags	Chai	39		18	5	15	1	1
2	24 - 12 oz bottles	Chang	17		15	4	10	3	1
3	12 - 550 ml bottles	Aniseed Syrup	13		15	4	10	1	2

Orders									
Name	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	ShipCountry	LineTotal	ProductID	UnitPrice	OrderID
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	595.2	10	18	10243
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	150	31	15	10244
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	40	33	10	10245

Cardinality

One to many (1:n)

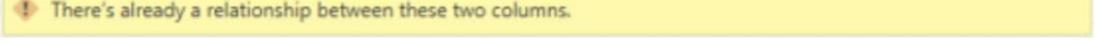
Make this relationship active

Assume referential integrity

Cross filter direction

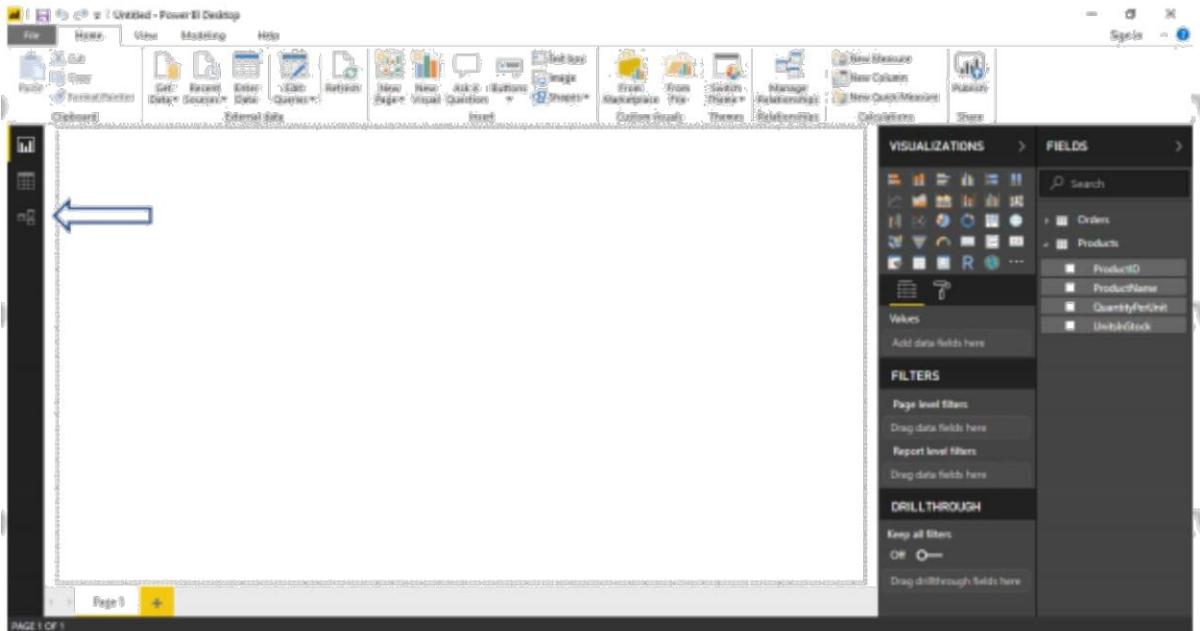
Single

Apply security filters in both directions

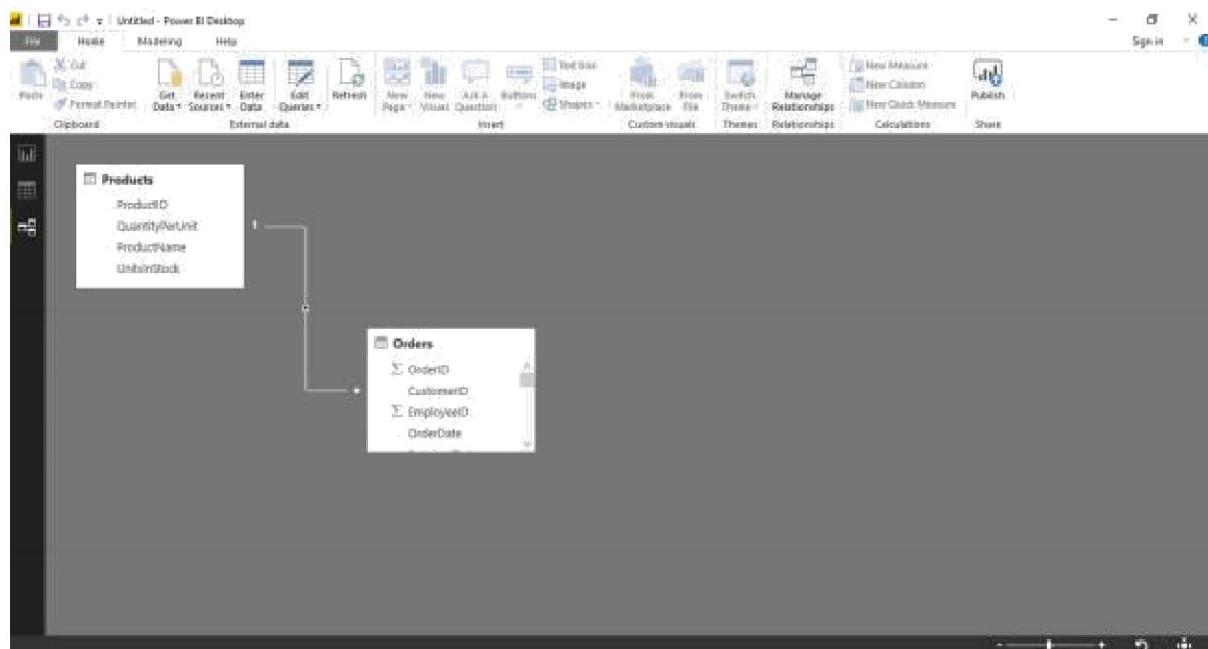
 There's already a relationship between these two columns.

OK Cancel

Step 6: Select Cancel, and then select Relationship view in Power BI Desktop.



Step 7: We see the following, which visualizes the relationship between the queries.



Create relationship

Select tables and columns that are related.

Products

ProductID	QuantityPerUnit	ProductName	UnitsInStock
1	10 boxes x 20 bags	Chai	39
2	24 - 12 oz bottles	Chang	17
3	12 - 550 ml bottles	Aniseed Syrup	13

Orders

Name	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	ShipCountry	LineTotal	ProductID	Ur
X-Stop	Taucherstraße 10	Cunewalde		null	Germany	595.2	10	
X-Stop	Taucherstraße 10	Cunewalde		null	Germany	150	31	
X-Stop	Taucherstraße 10	Cunewalde		null	Germany	40	33	

Cardinality

One to many (1:n)

Make this relationship active

Assume referential integrity

Cross filter direction

Single

Apply security filters in both directions

 There's already a relationship between these two columns.

OK

Cancel

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

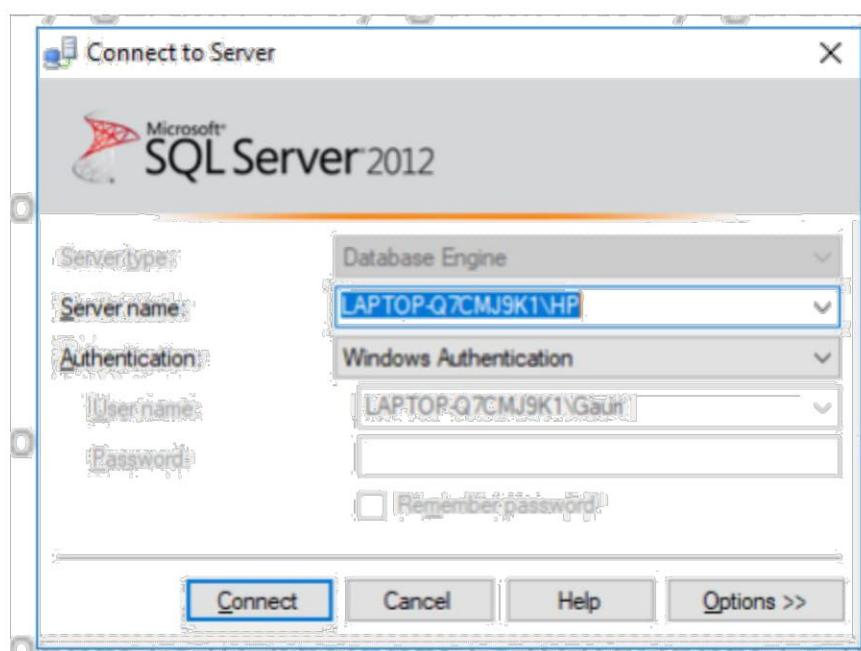
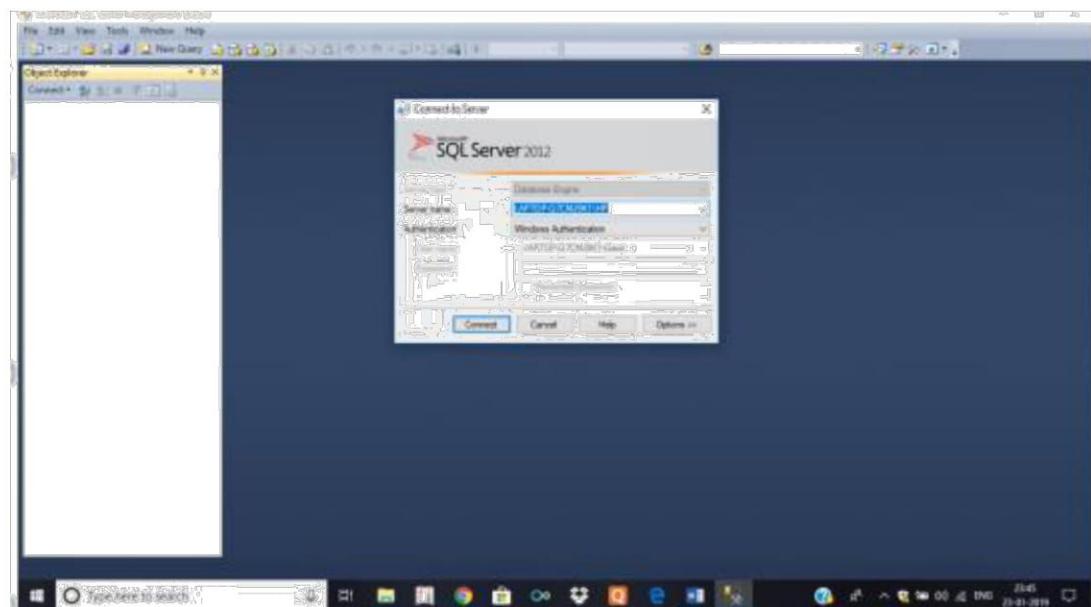
Practical 2 B

Perform the Extraction Transformation and Loading (ETL) process to construct the database in the SQL server.

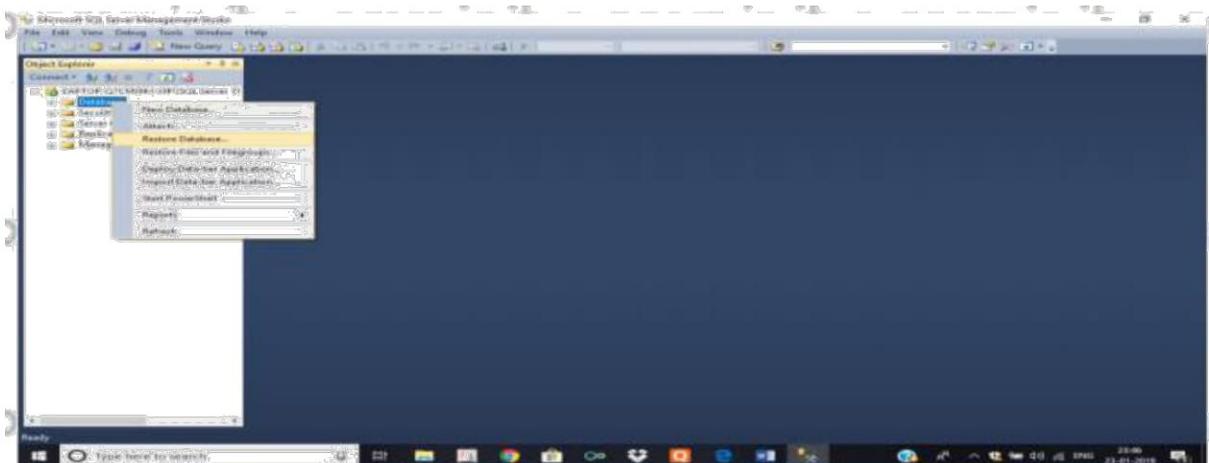
Software requirements: SQL SERVER 2012 FULL VERSION (SQLServer2012SP1-FullSlipstream-ENU-x86)

Steps to install SQL SERVER 2012 FULL VERSION (SQLServer2012SP1- FullSlipstream-ENU-x86) are given in my previous post.

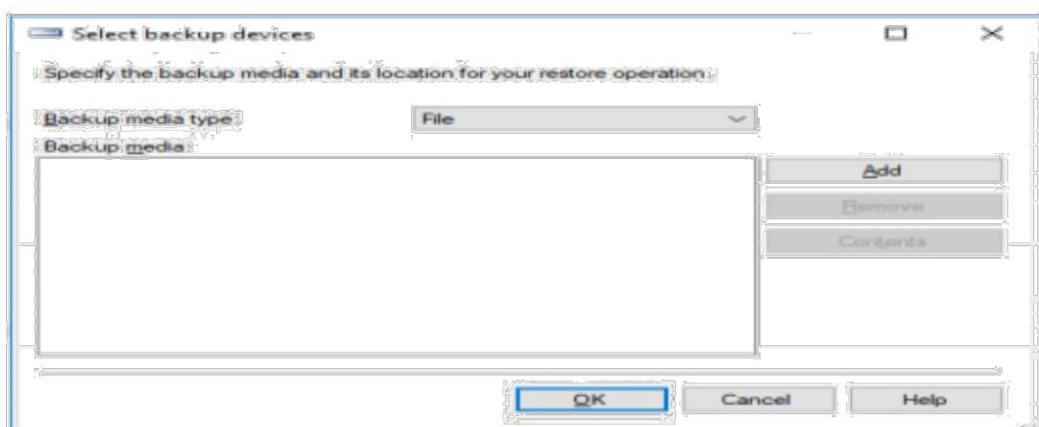
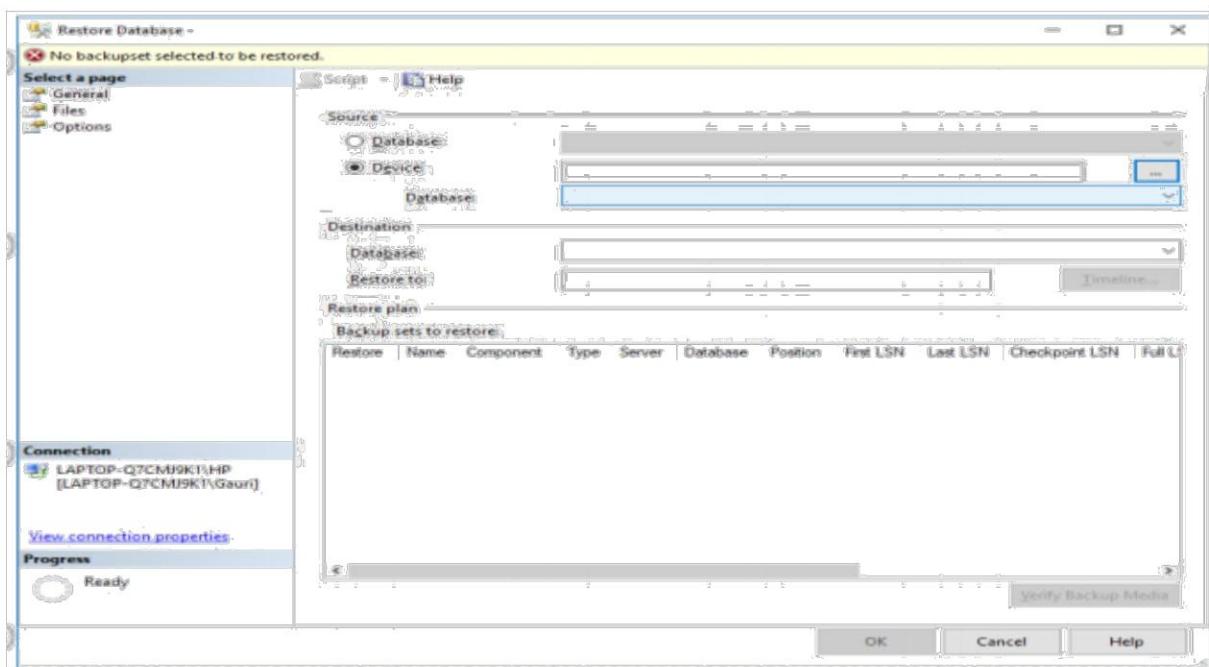
Step 1: Open SQL Server Management Studio to restore backup file



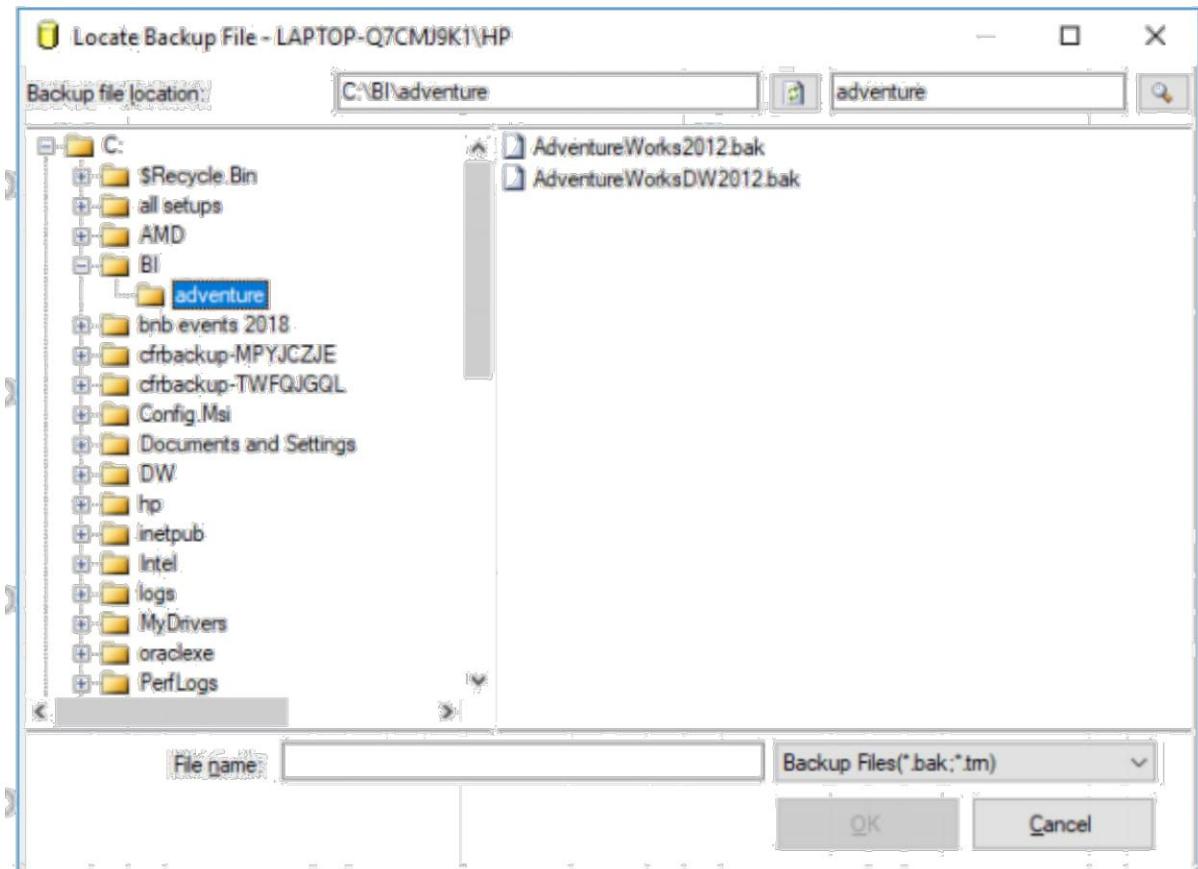
Step 2: Right click on Databases → Restore Database



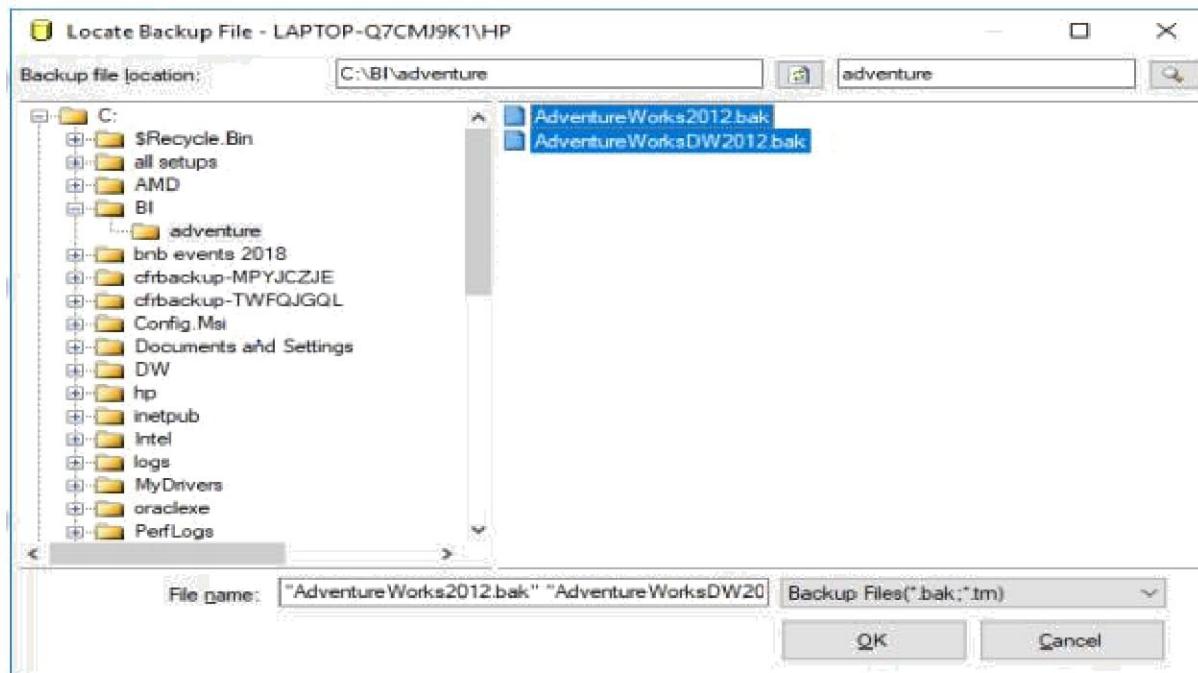
Step 3: Select Device → click on icon towards end of device box



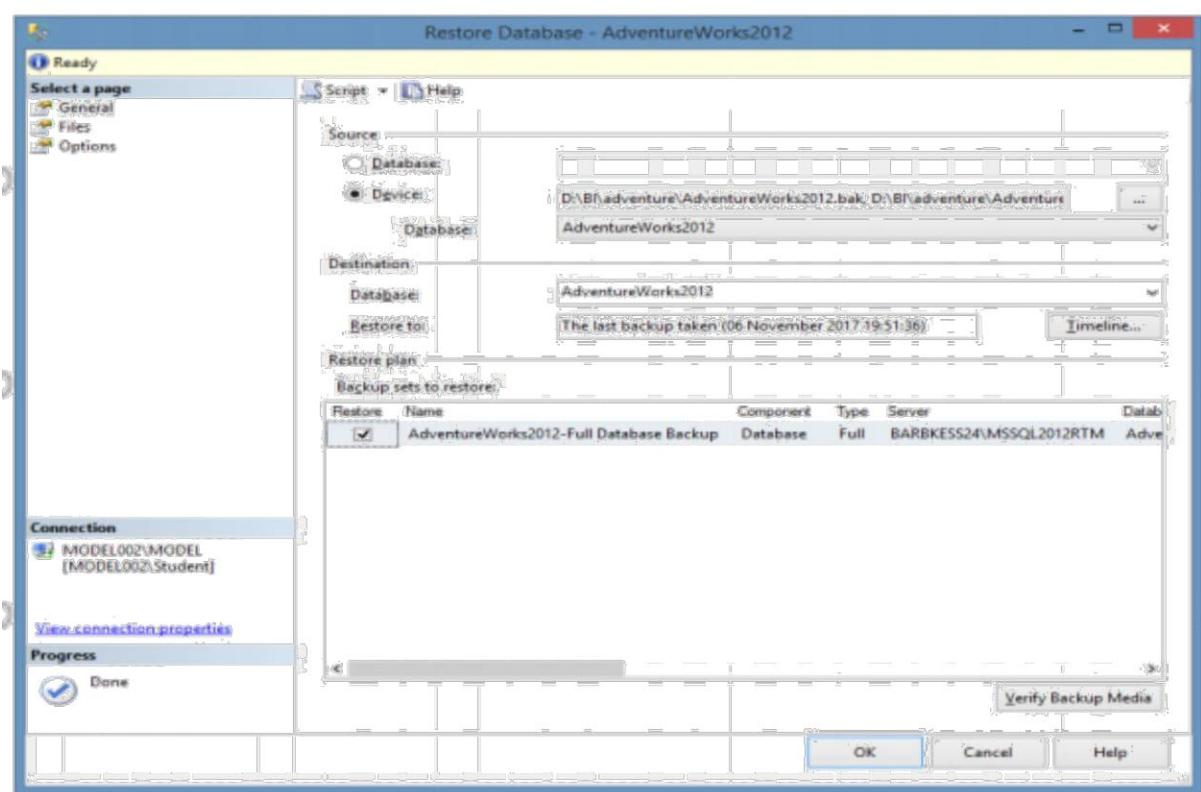
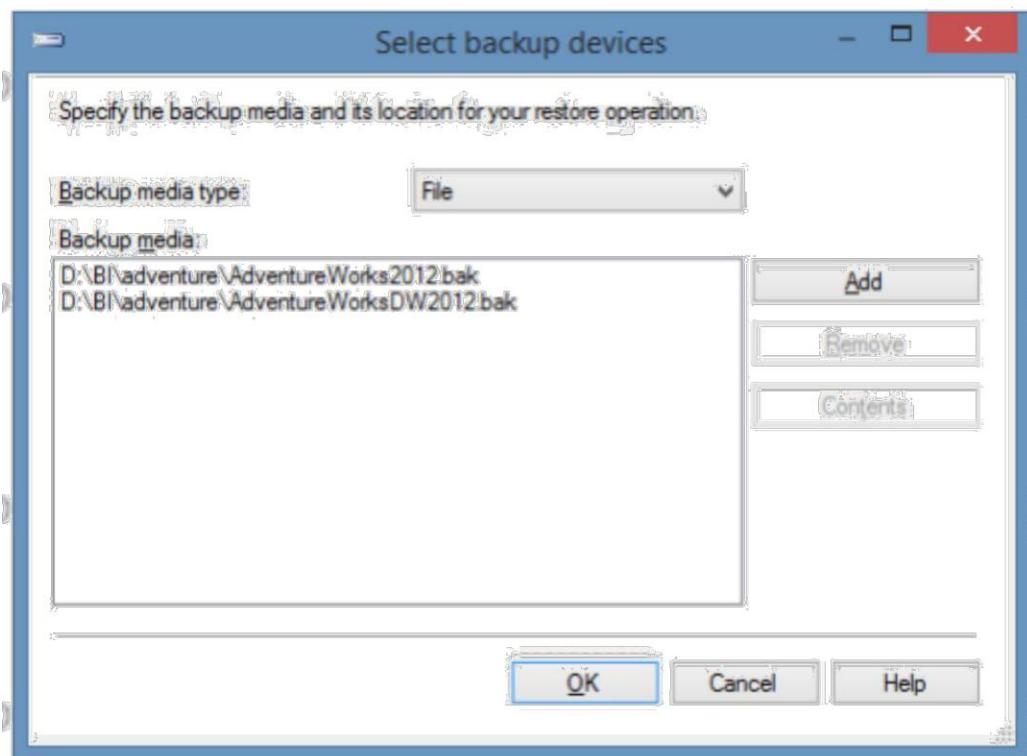
Step 4: Click on Add → Select path of backup files

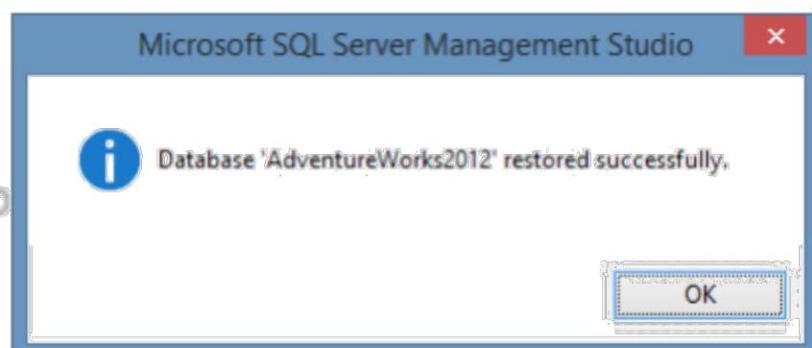
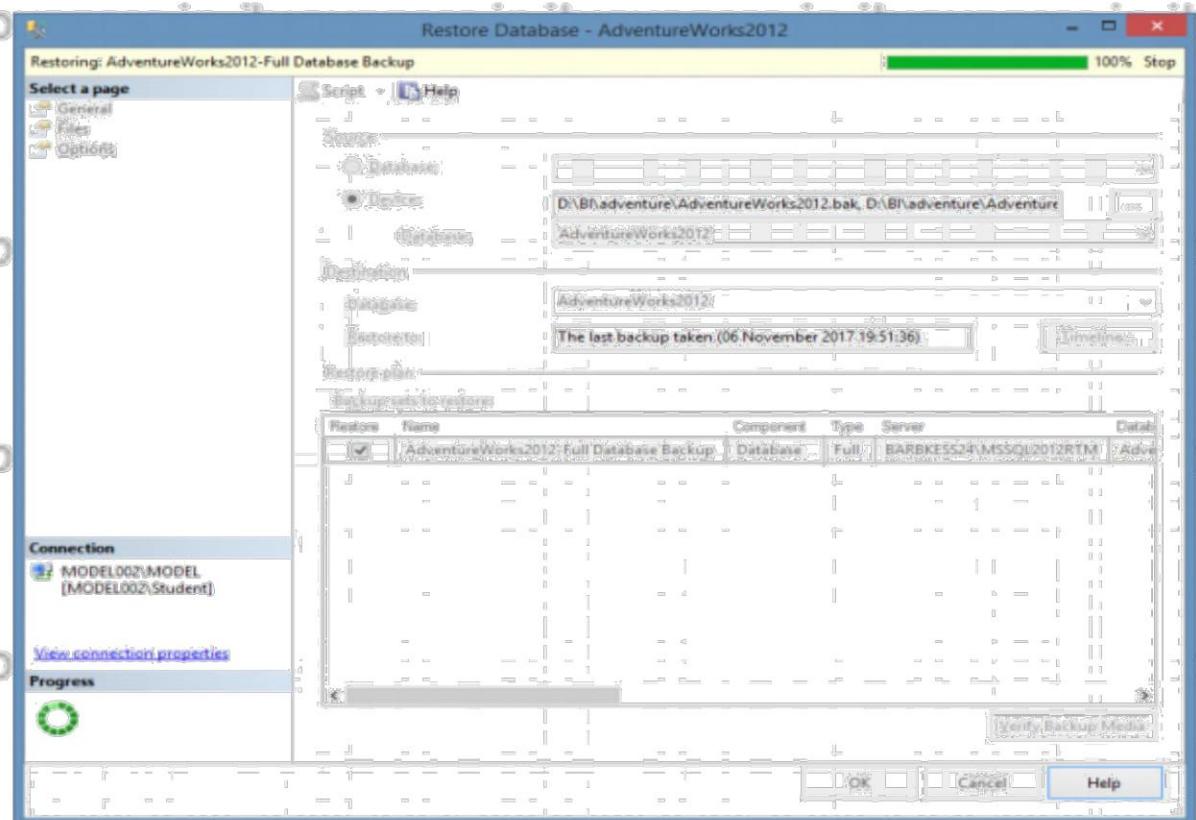


Step 5: Select both files at a time

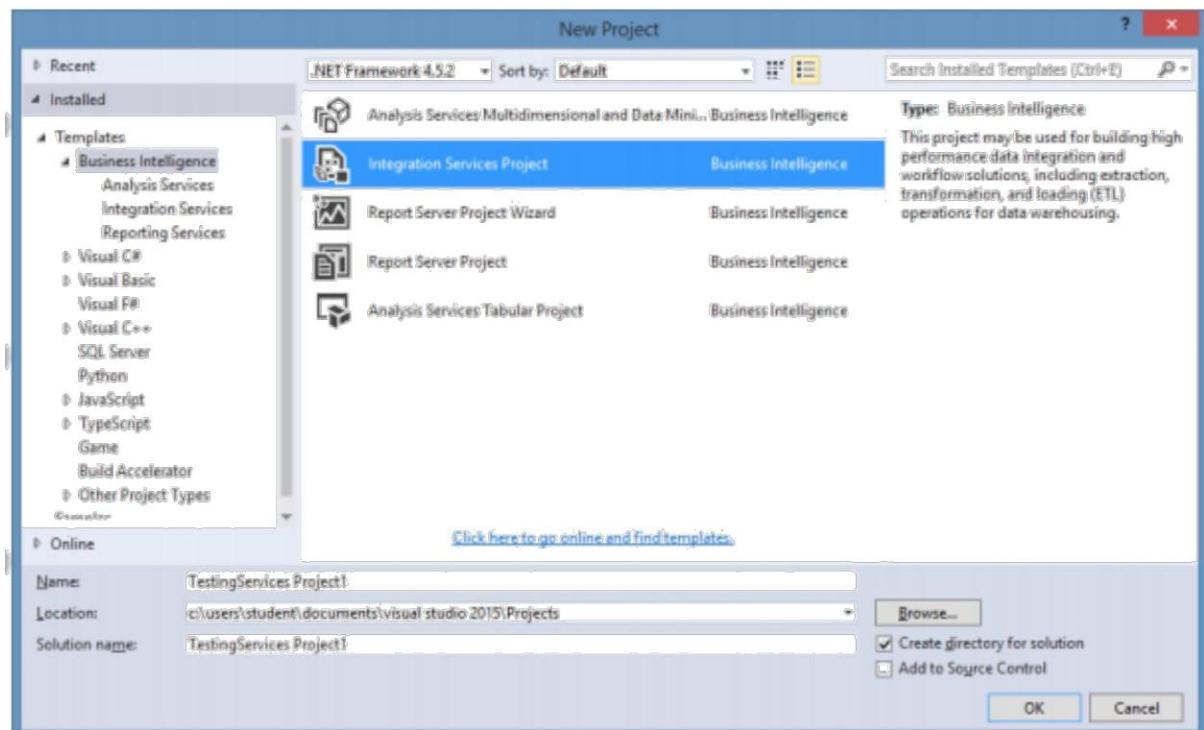
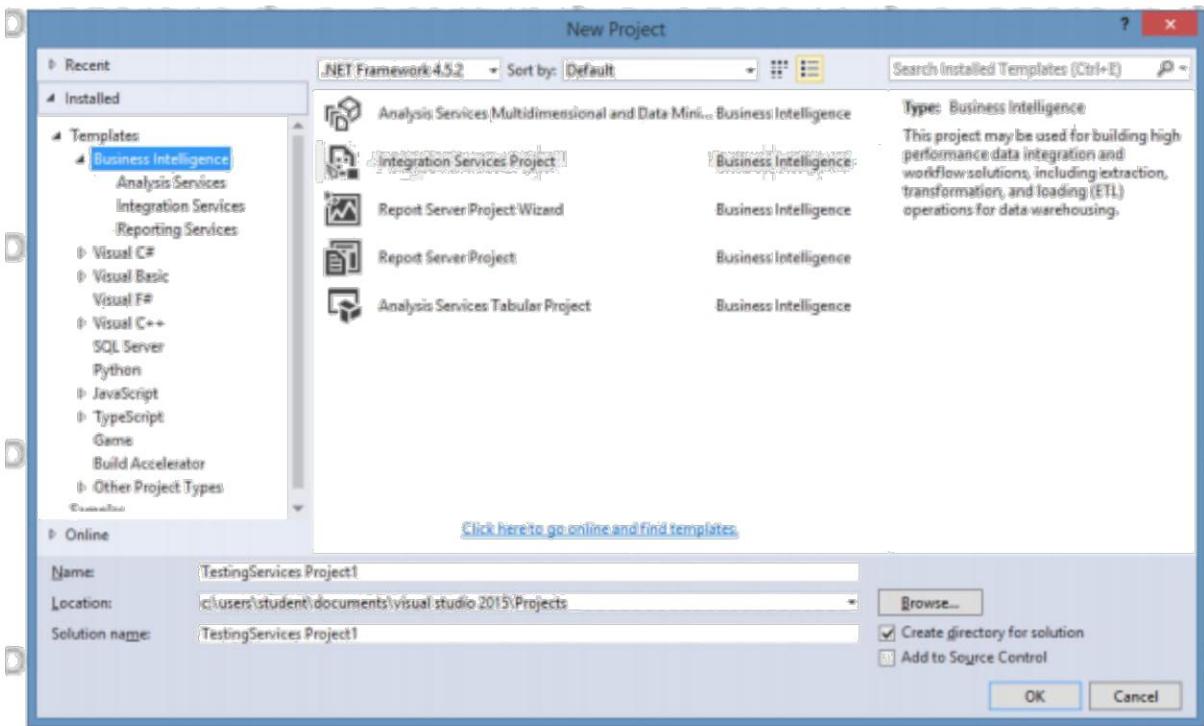


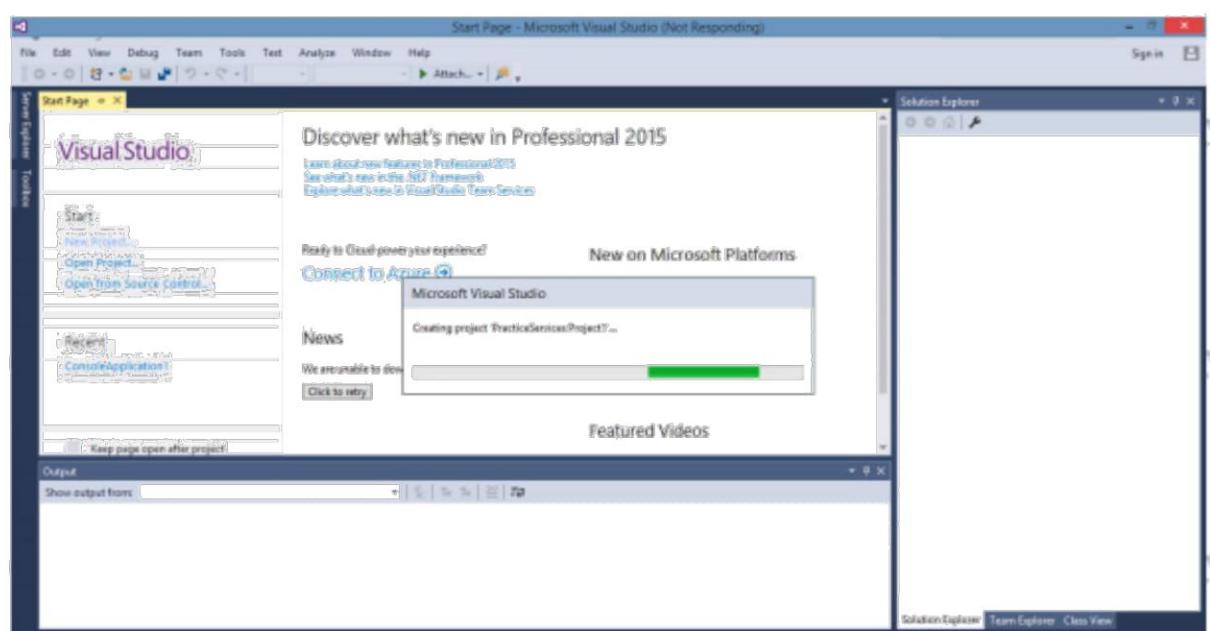
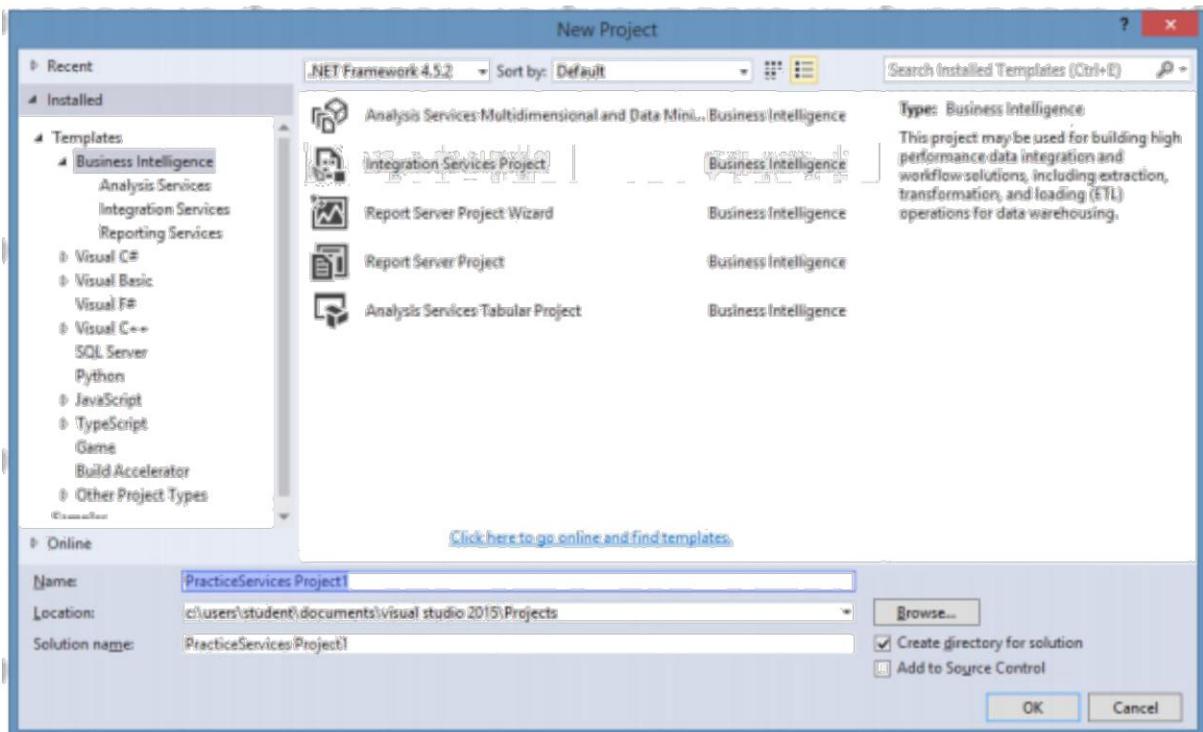
Step 6 : Click ok and in select backup devices window Add both files of AdventureWorks



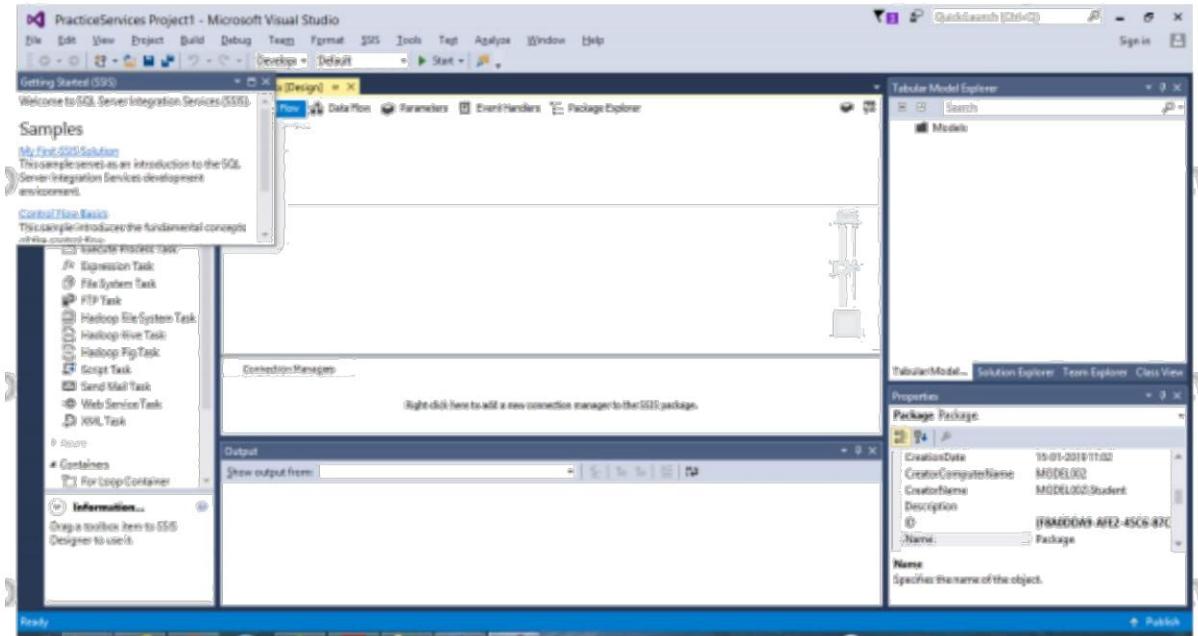


Step 7: Open SQL Server Data Tools Select File → New → Project → Business Intelligence → Integration Services Project & give appropriate project name.

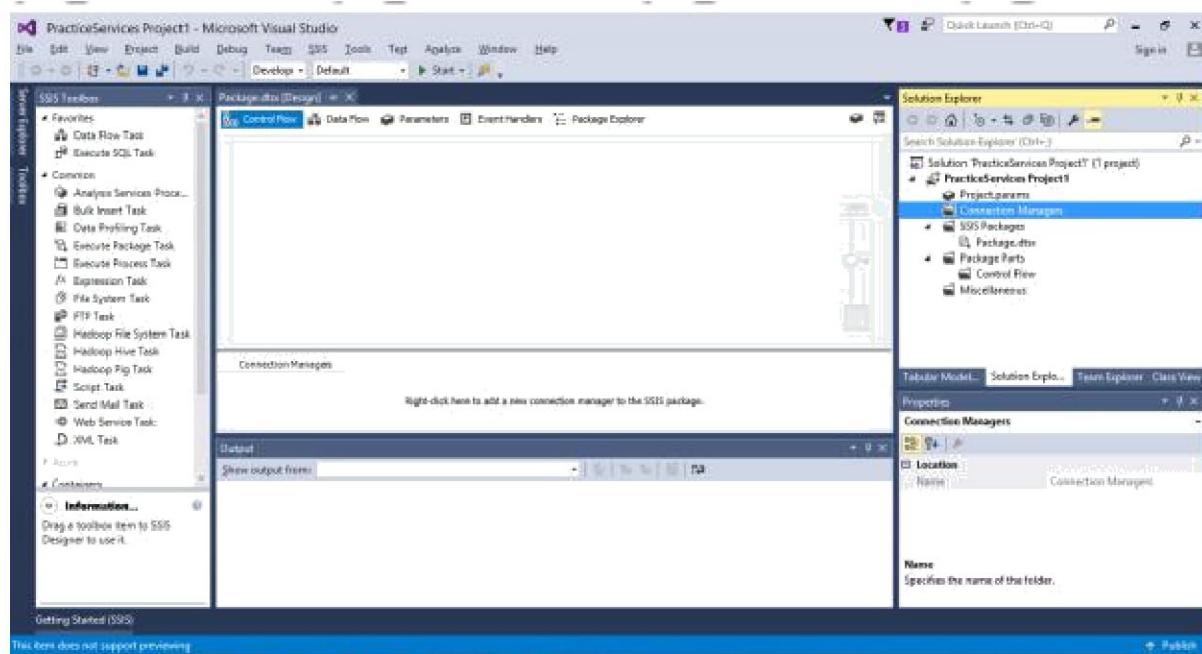




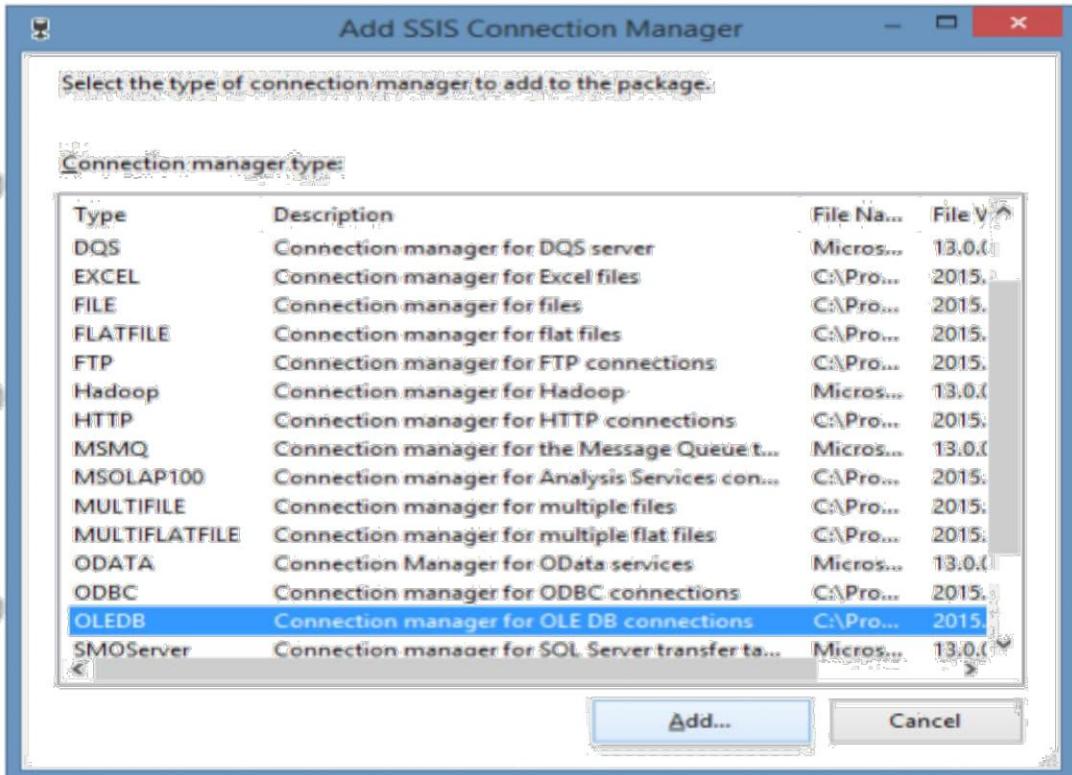
Environment consists of SQL Server Integration Services(SSIS)



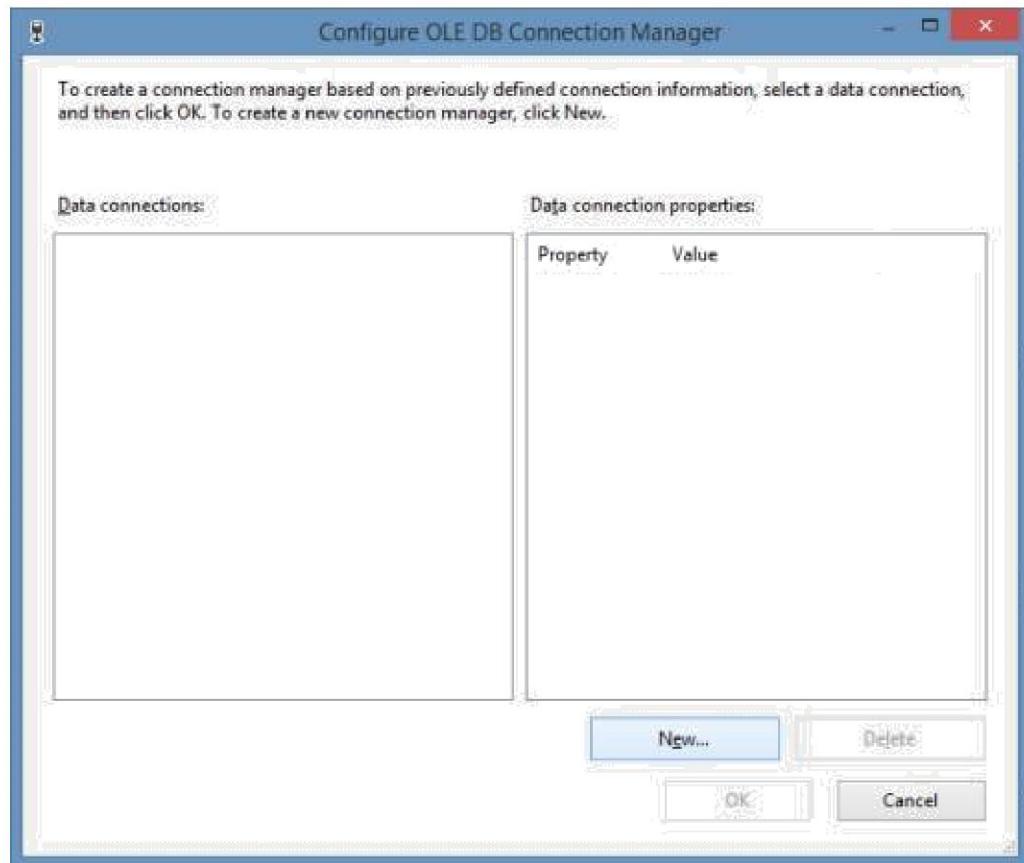
Step 8: Right click on Connection Managers in solution explorer and click on New Connection Manager. Add SSIS connection manager window appears.



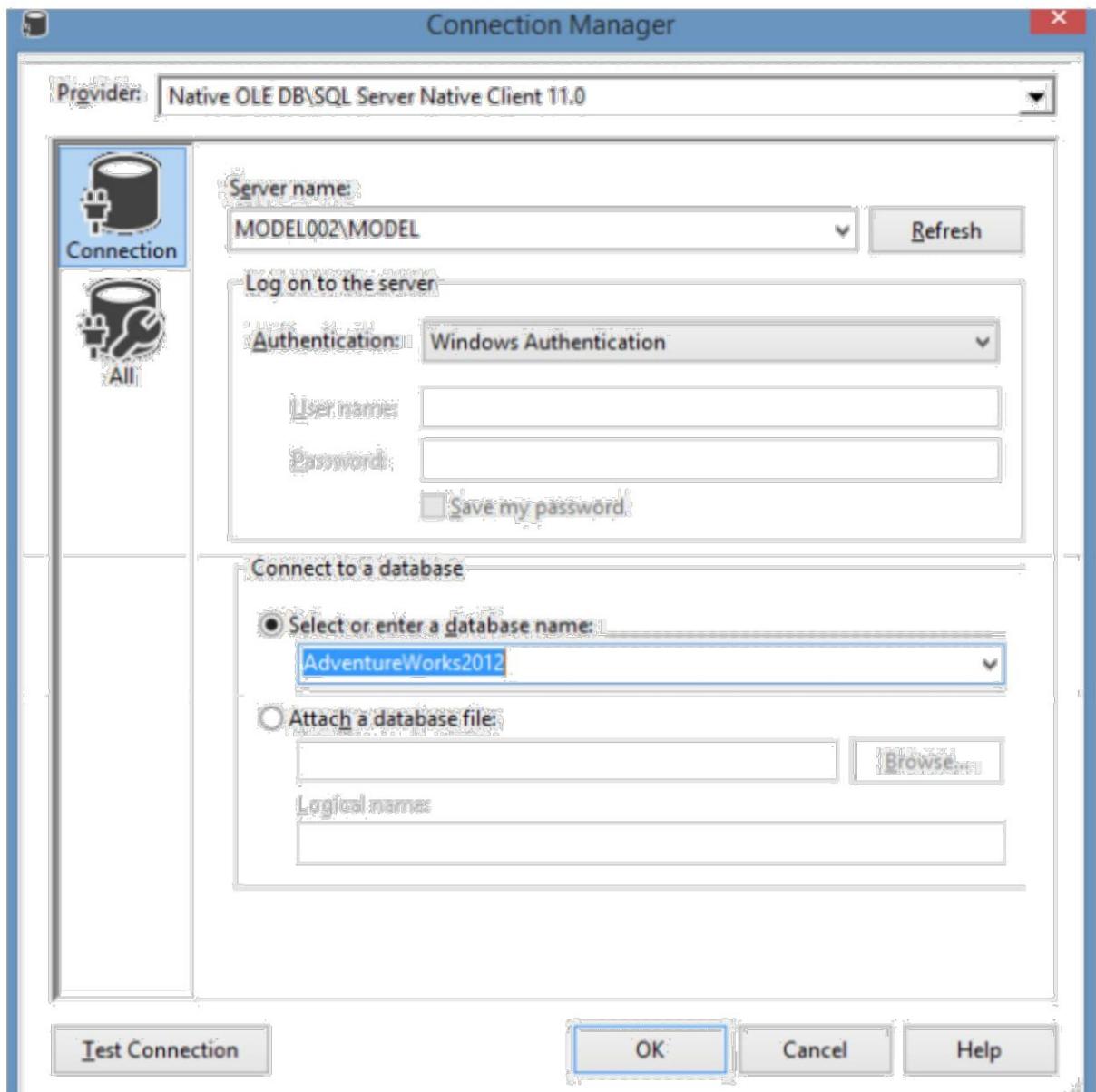
Step 9: Select OLEDB Connection Manager and Click on Add



Step 10: Configure OLE DB Connection Manager window appears → Click on New



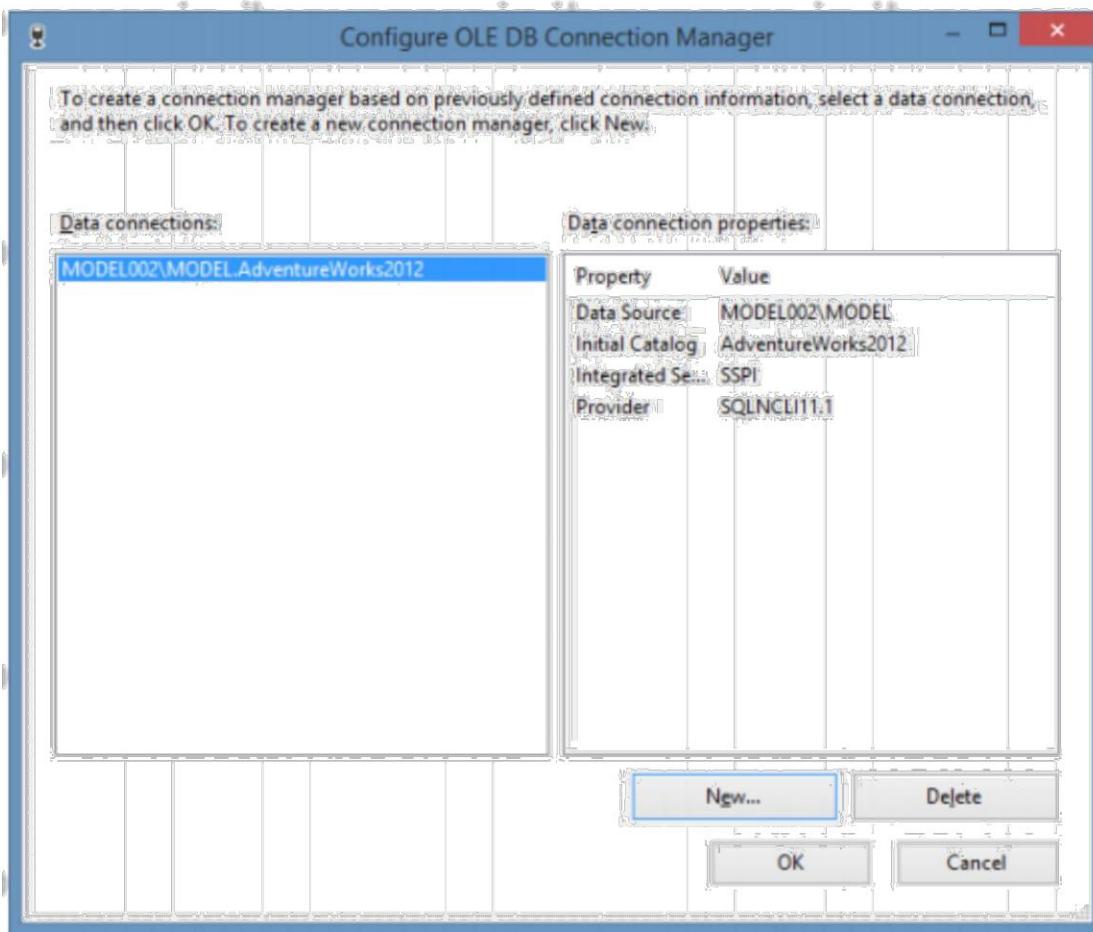
Step 11: Select Server name(as per your machine) from drop down and database name and click on Test connection.



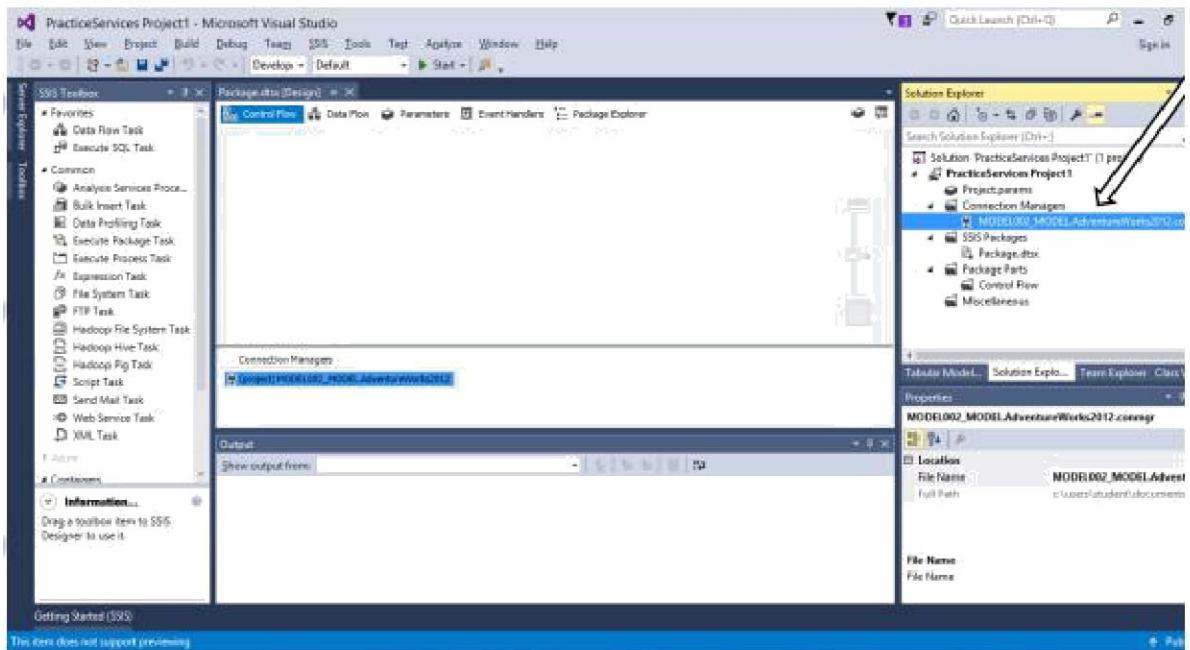
If test connection succeeded click on OK



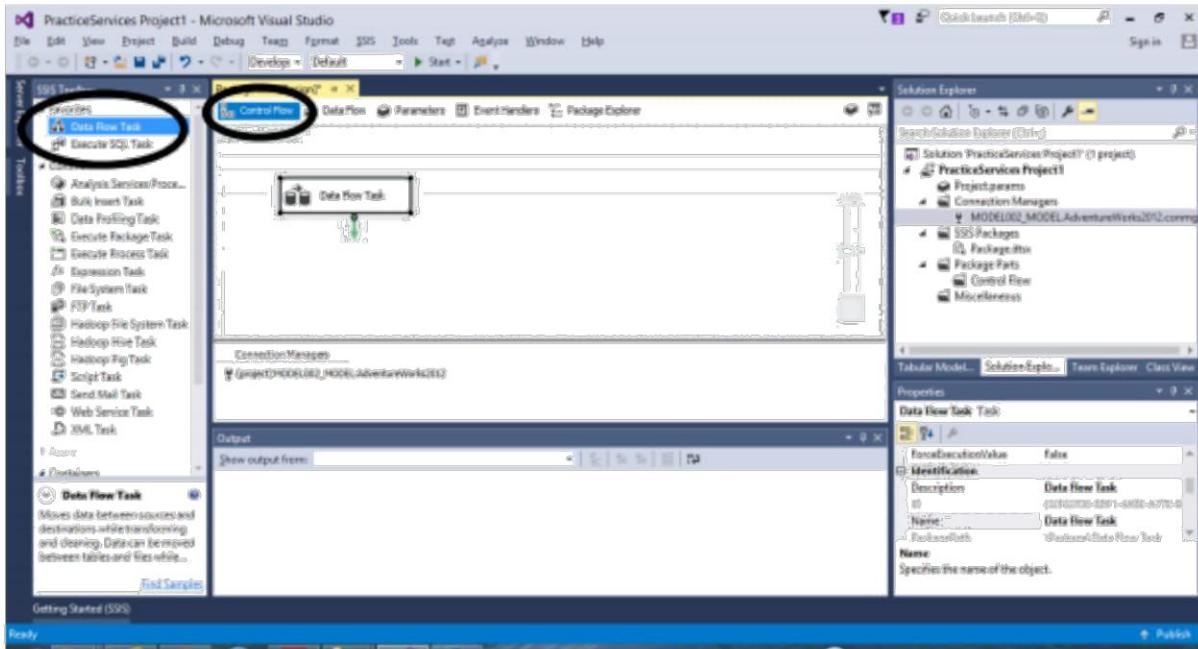
Step 12: Click on OK



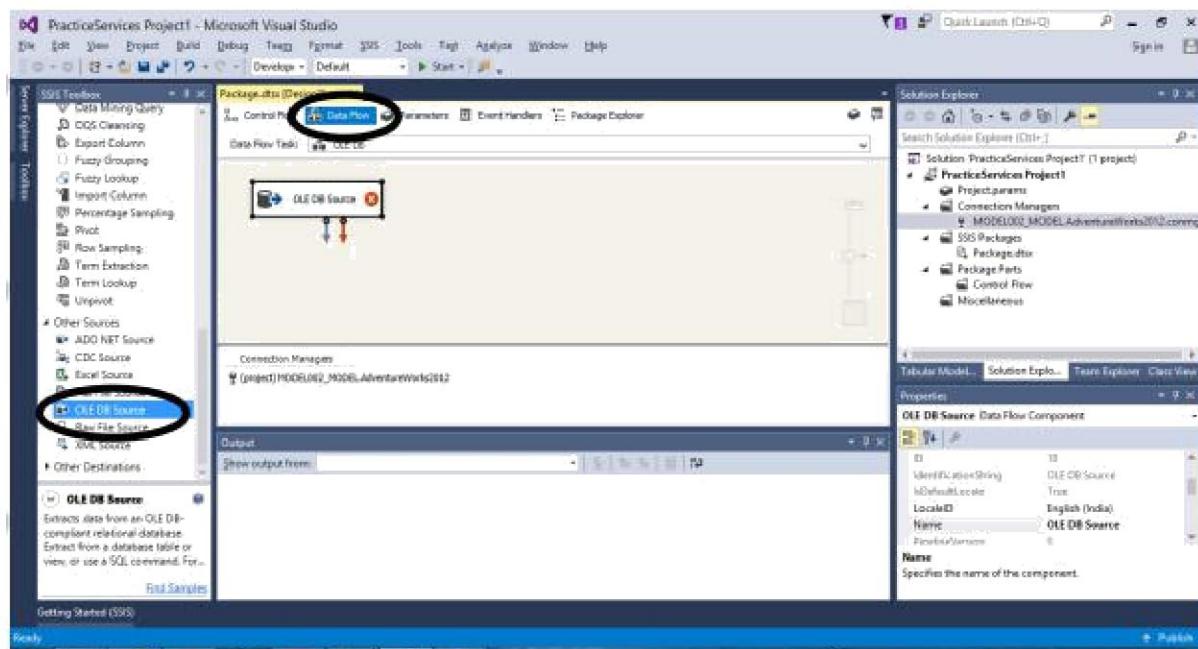
Connection is added to connection manager



Step 13: Drag and drop Data Flow Task in Control Flow tab

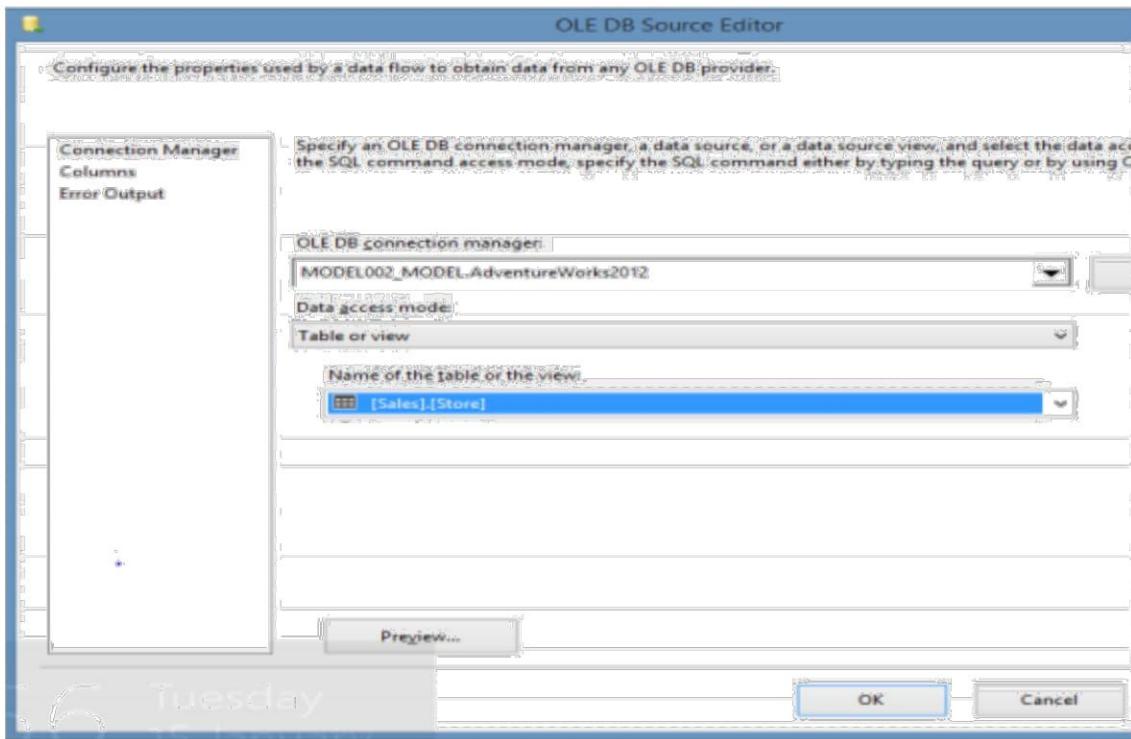


Step 14: Drag OLE DB Source from Other Sources and drop into Data Flow tab

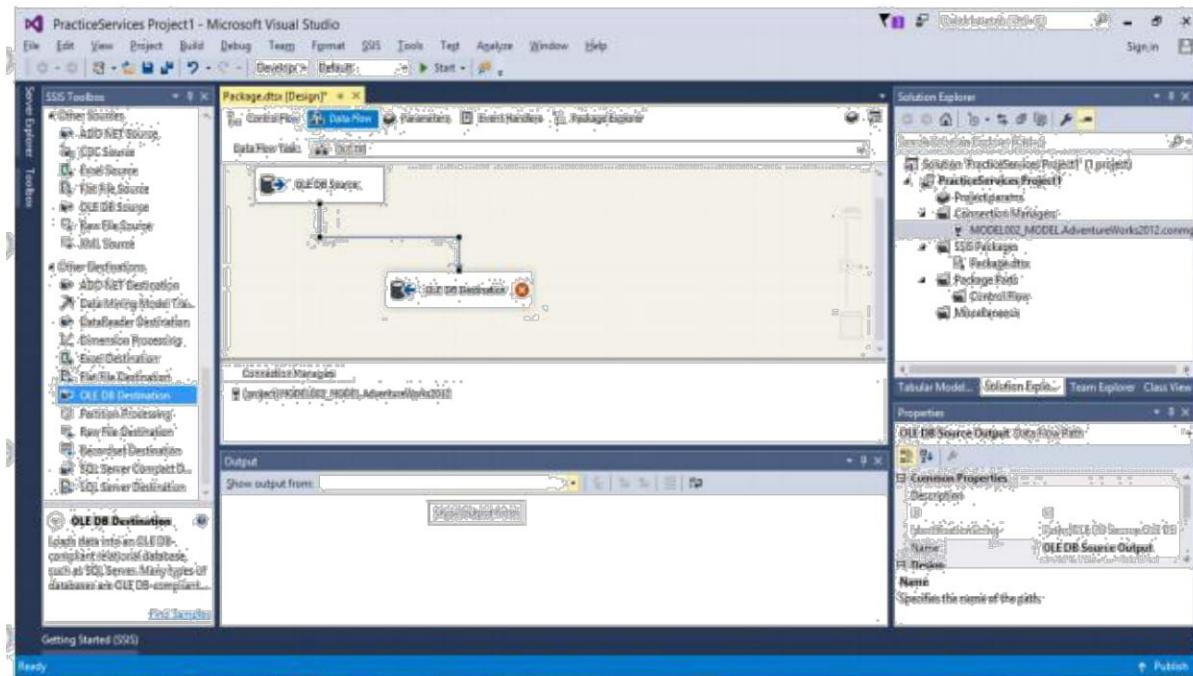


Step 15: Double click on OLE DB source → OLE DB Source Editor appears→ click on New to add connection manager.

Select [Sales].[Store] table from drop down → ok

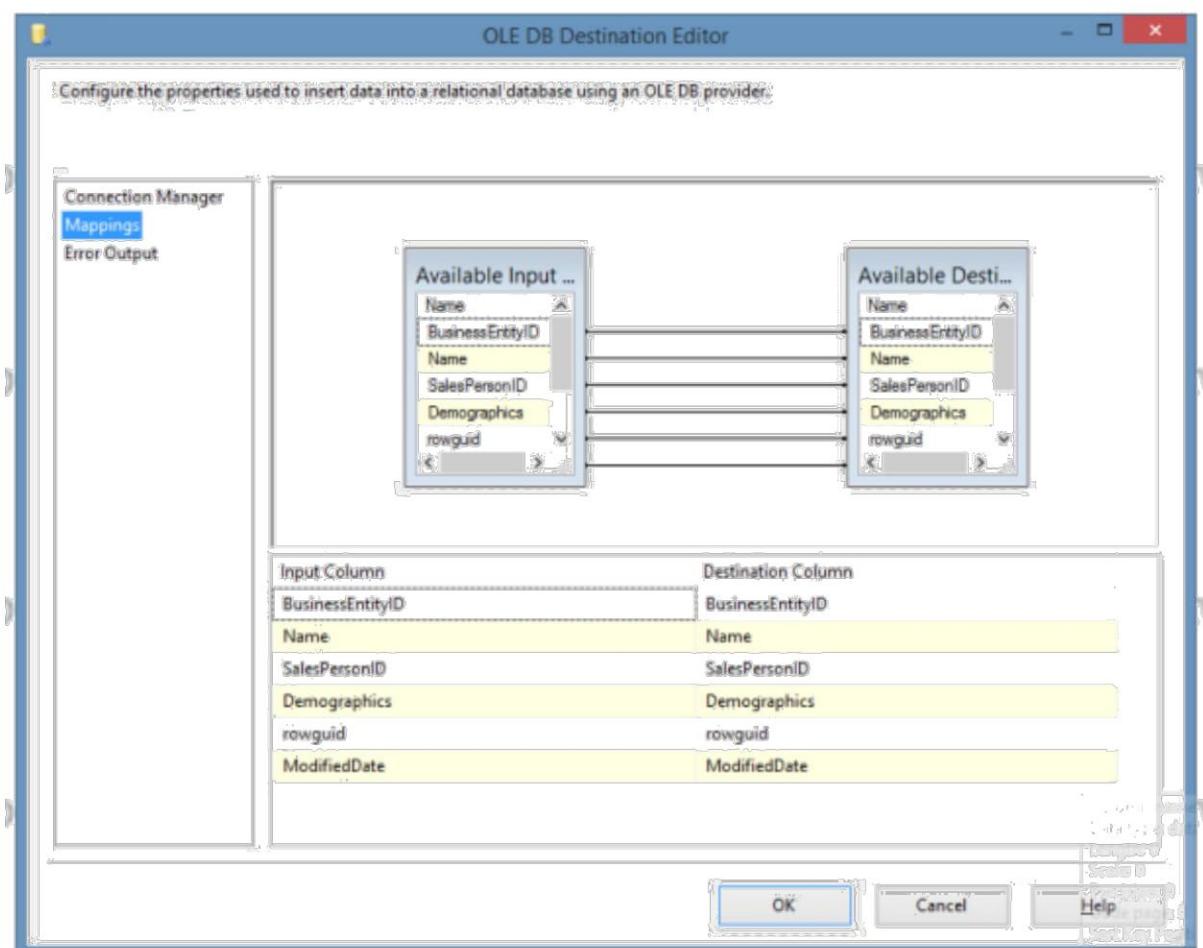
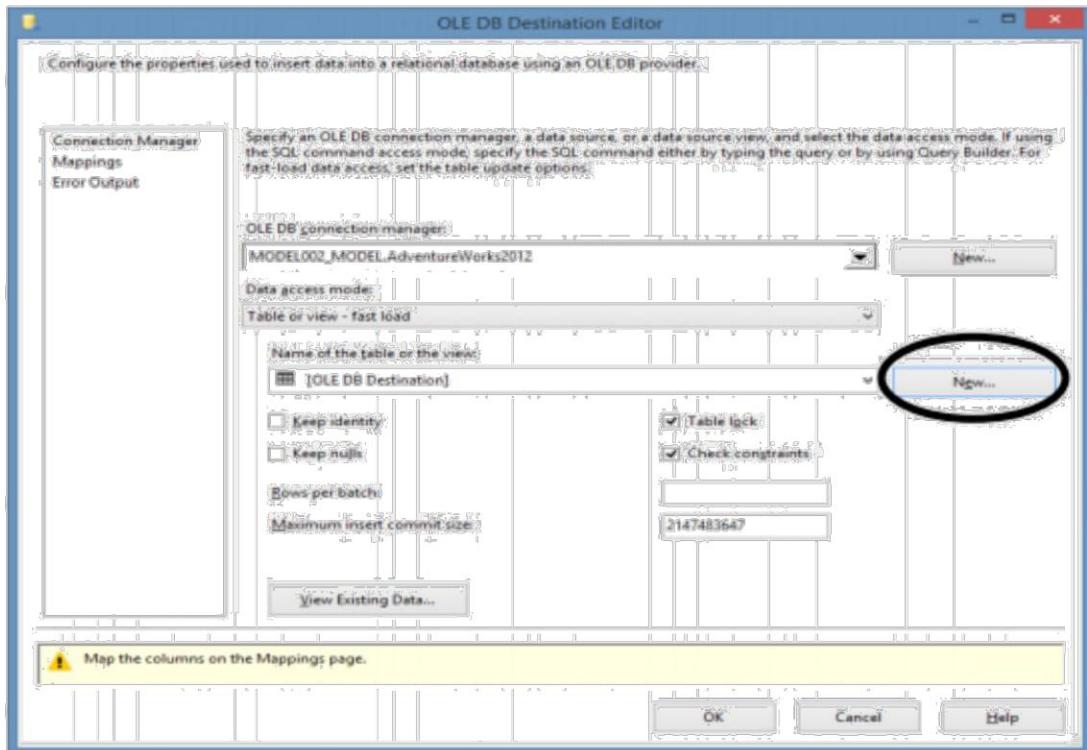


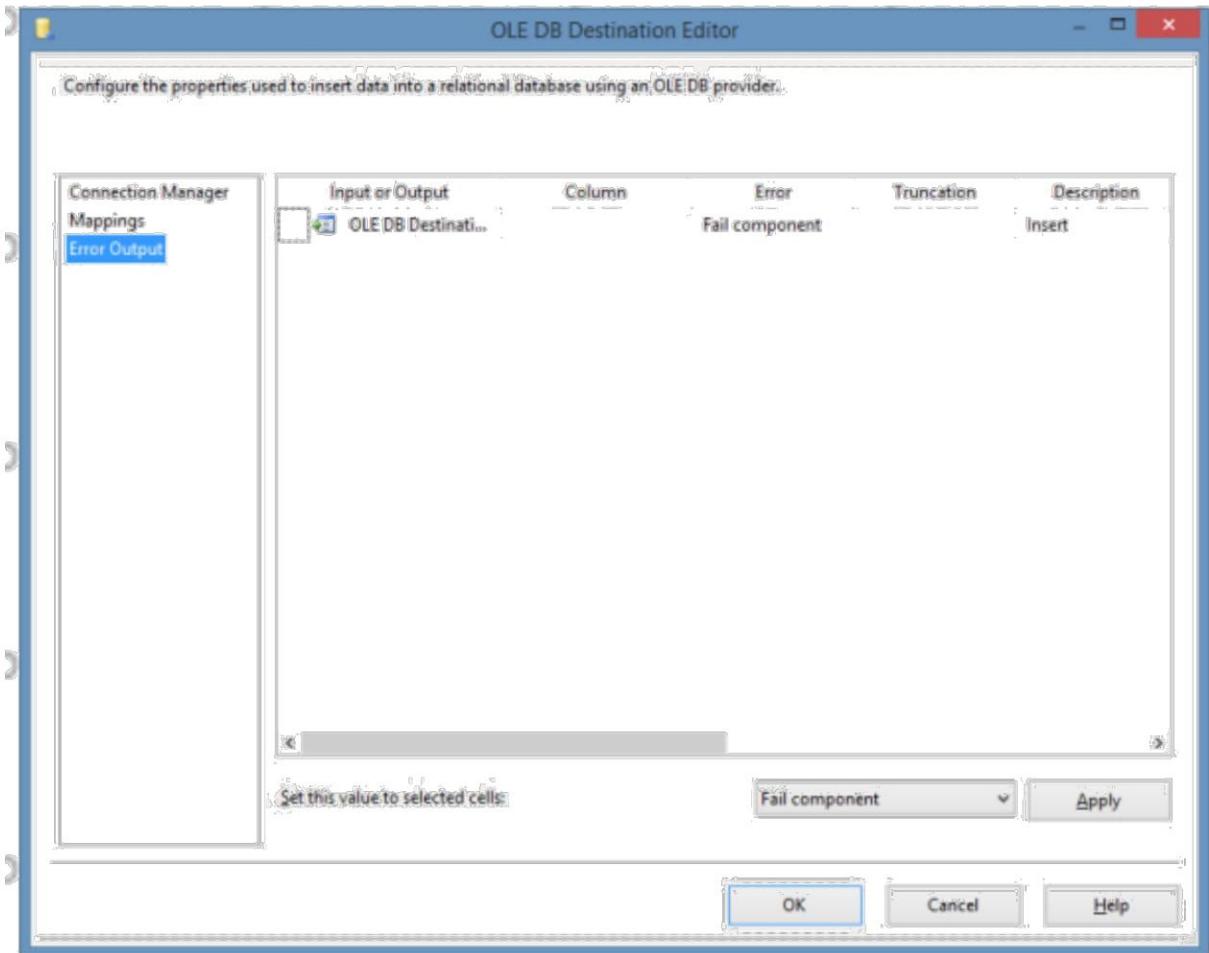
Step 16: Drag ole db destination in data flow tab and connect both



Step 17: Double click on OLE DB destination.

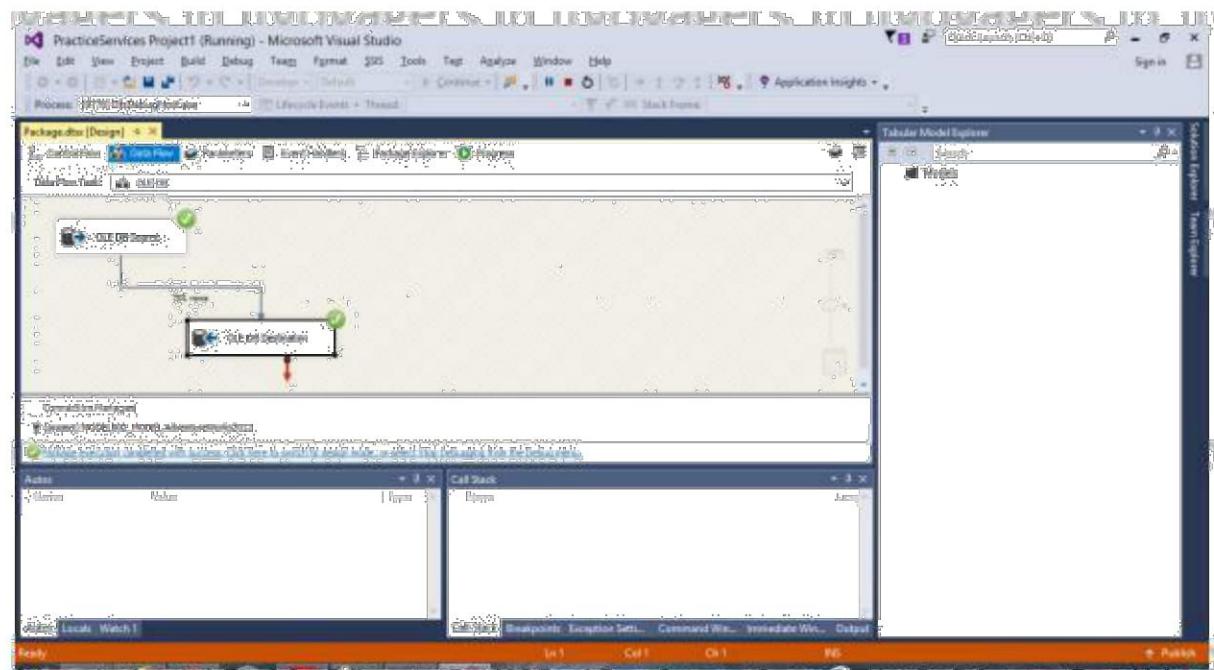
Click on New to run the query to get [OLE DB Destination] in Name of the table or the view.





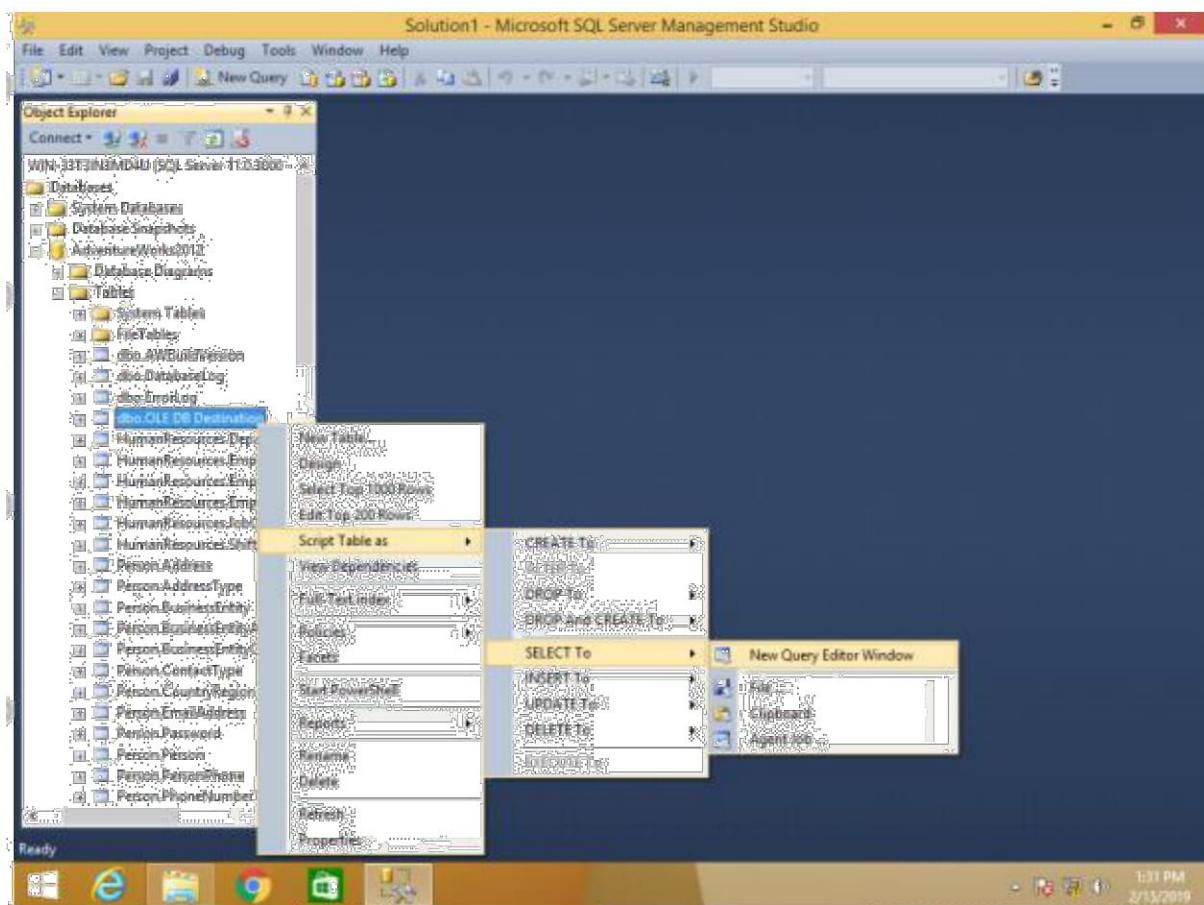
Click on ok

Step 18: Click on start.



Step 19: Go to SQL Server Management Studio.

In database tab → Adventureworks → Right click on [dbo].[OLE DB Destination] → Script Table as → SELECT To → New Query Editor Window



Step 20: Execute following query to get output.

```
USE [AdventureWorks2012]
```

```
GO
```

```
SELECT [BusinessEntityID]
```

```
,[Name]
```

```
,[SalesPersonID]
```

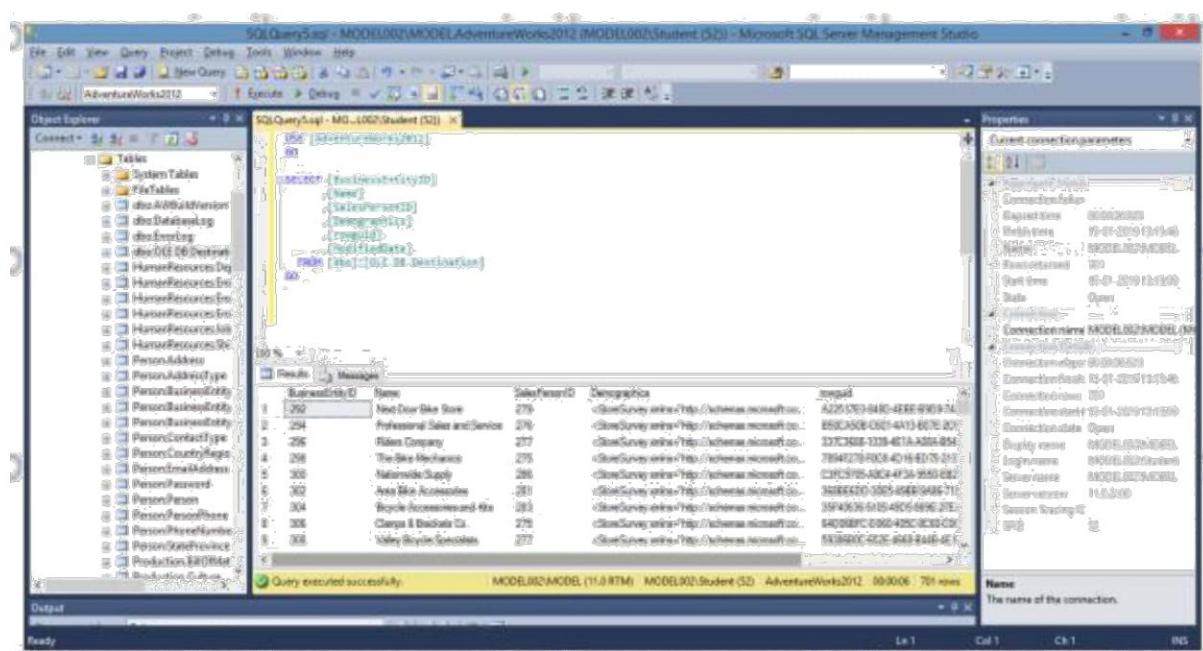
```
,[Demographics]
```

```
,[rowguid]
```

```
,[ModifiedDate]
```

```
FROM [dbo].[OLE DB Destination]
```

```
GO
```



PRACTICAL 3

Create the cube with suitable dimension and fact tables based on

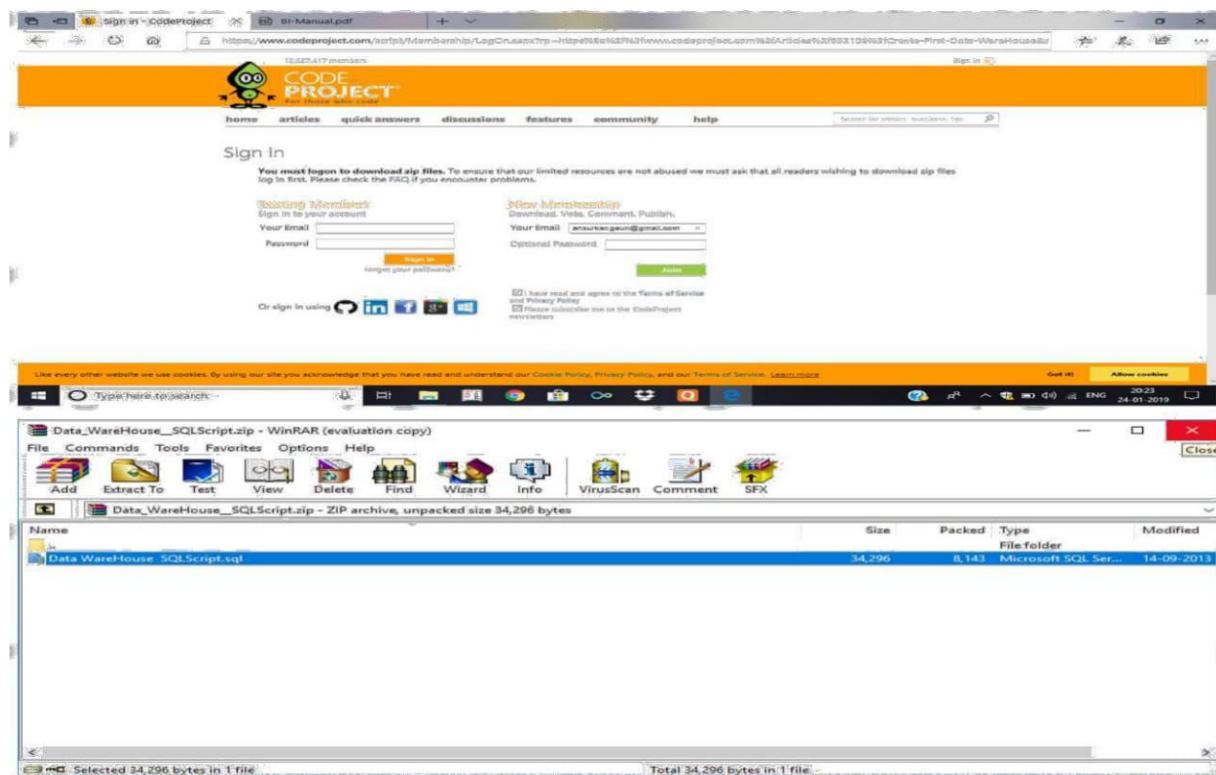
OLAP

Step 1: 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.

Downloading "Data_WareHouse_SQLScript.zip" from the article
<https://www.codeproject.com/Articles/652108/Create-First-Data-Warehouse>



After downloading extract file in folder.

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

1. Open SQL Server Management Studio 2012
2. Connect Database Engine



Password for sa : admin123 (as given during installation)

Click Connect.

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

OR

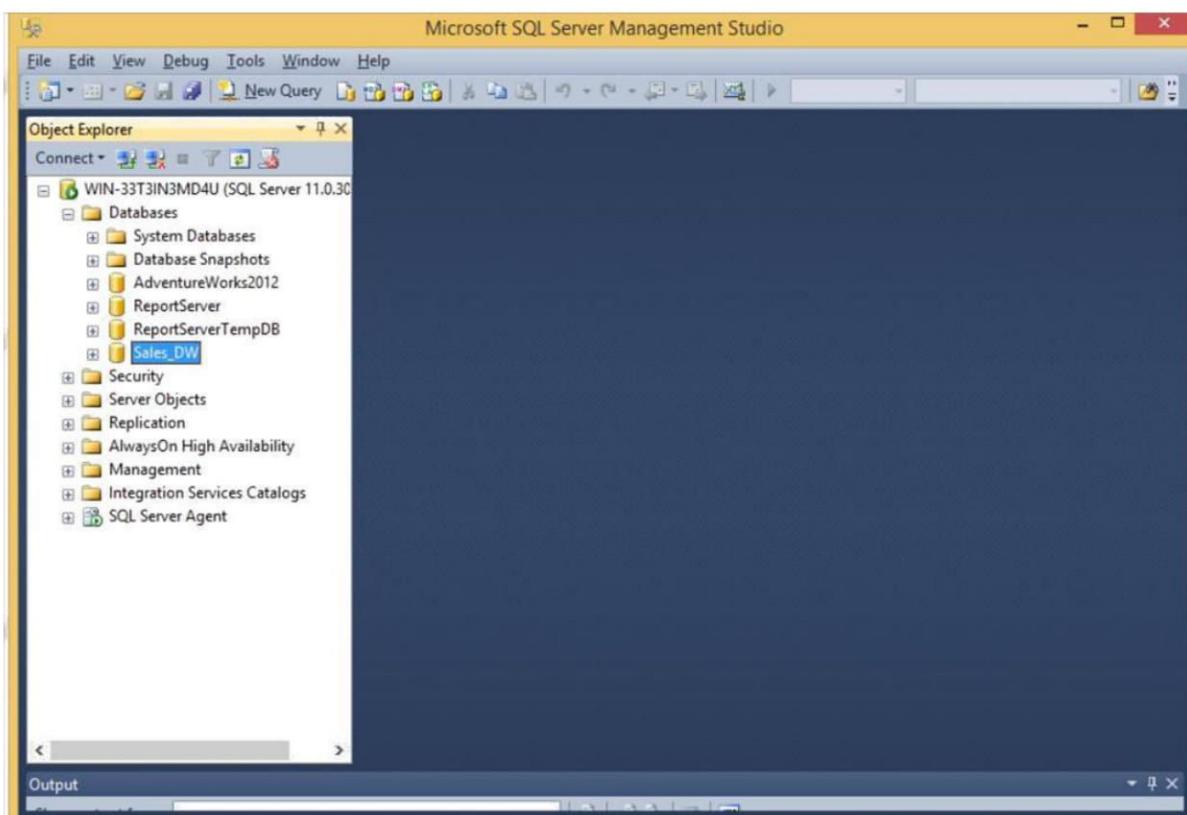
1. Go to the extracted sql file and double click on it.
2. New Sql Query Editor will be opened containing Sales_DW Database.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a connection to 'LAPTOP-Q7CM9K1\HP (SQL Server)' is selected, and the 'Databases' node is expanded, showing 'master', 'model', 'msdb', 'tempdb', and 'Sales_DW'. The 'Sales_DW' database is highlighted. In the center pane, a query window titled 'Data WareHouse S... (55) [Sales_DW]' displays the following T-SQL script:

```
--Create database Sales_DW
GO
--Create customer dimension table in Data Warehouse which will hold customer personal details.
CREATE TABLE DimCustomer
(
    CustomerID int primary key identity,
    CustomerName varchar(20) not null,
    CustomerEmail varchar(50),
    Gender varchar(20)
)
GO
--Insert 5 rows in customer dimension table with sample values
INSERT INTO DimCustomer(CustomerName,Gender)
VALUES
('John Doe', 'Male'),
('Jane Smith', 'Female'),
('Michael Johnson', 'Male'),
('Sarah Williams', 'Female'),
('David Wilson', 'Male')
GO
--Create basic level of Product dimension table without considering any Category or Subcategory.
CREATE TABLE DimProduct
```

The status bar at the bottom indicates 'Connected (1/1)', 'LAPTOP-Q7CM9K1\HP (11.0 RTM) LAPTOP-Q7CM9K1\Gauri ... Sales_DW 00:00:00 0 rows', and the system clock '20:49 24-01-2019'.

3. Click on execute or press F5 by selecting query one by one or directly click on Execute.
4. After completing execution save and close SQL Server Management studio & Reopen to see Sales_DW in Databases Tab.

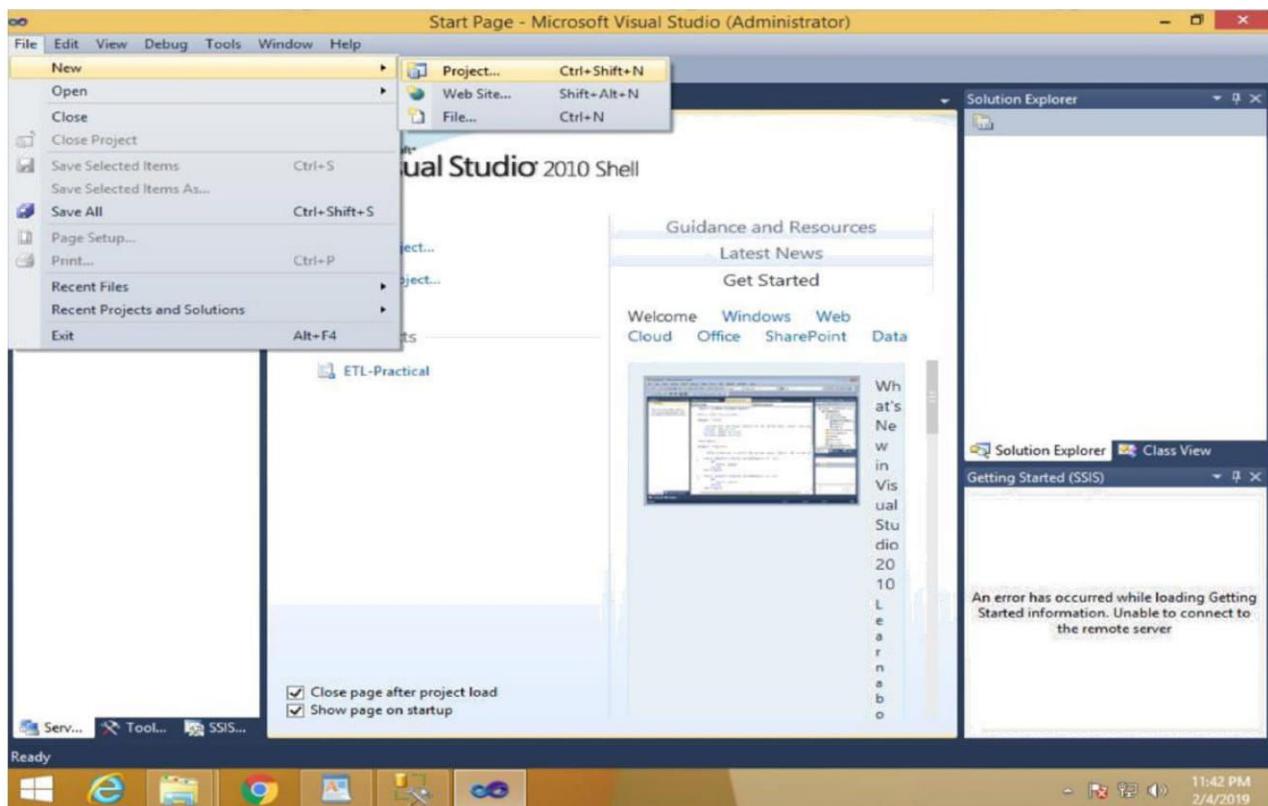


Step 2: Start SSDT environment and create New Data Source

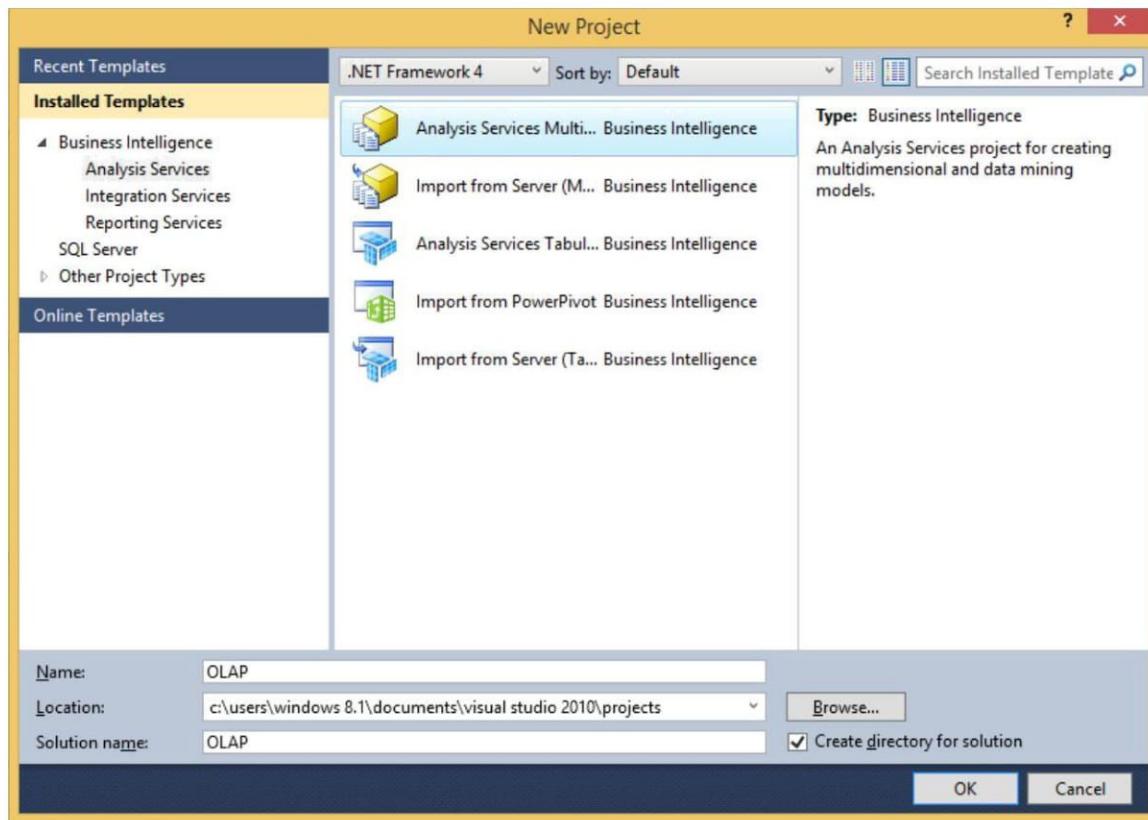
Go to Sql Server Data Tools --> Right click and run as administrator



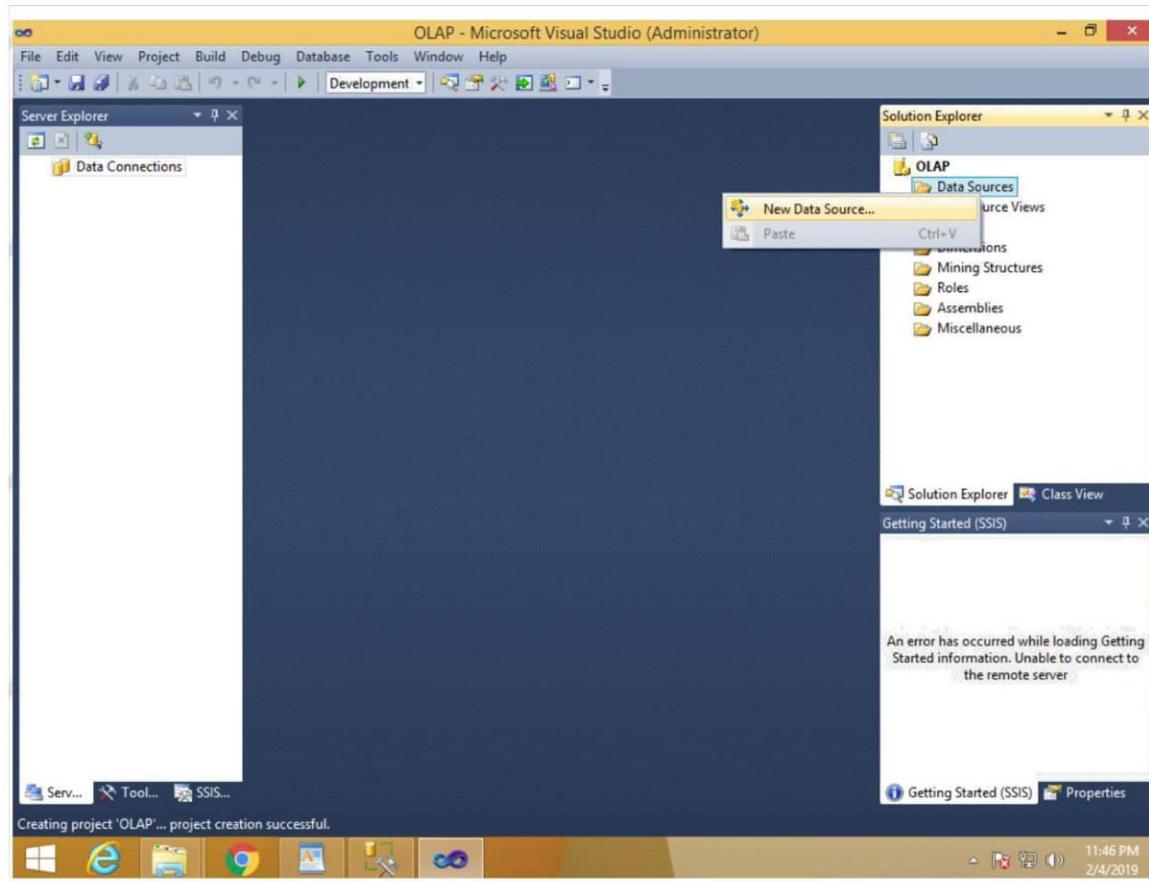
Click on File → New → Project



In Business Intelligence → Analysis Services Multidimensional and Data Mining models
→ appropriate project name → click OK



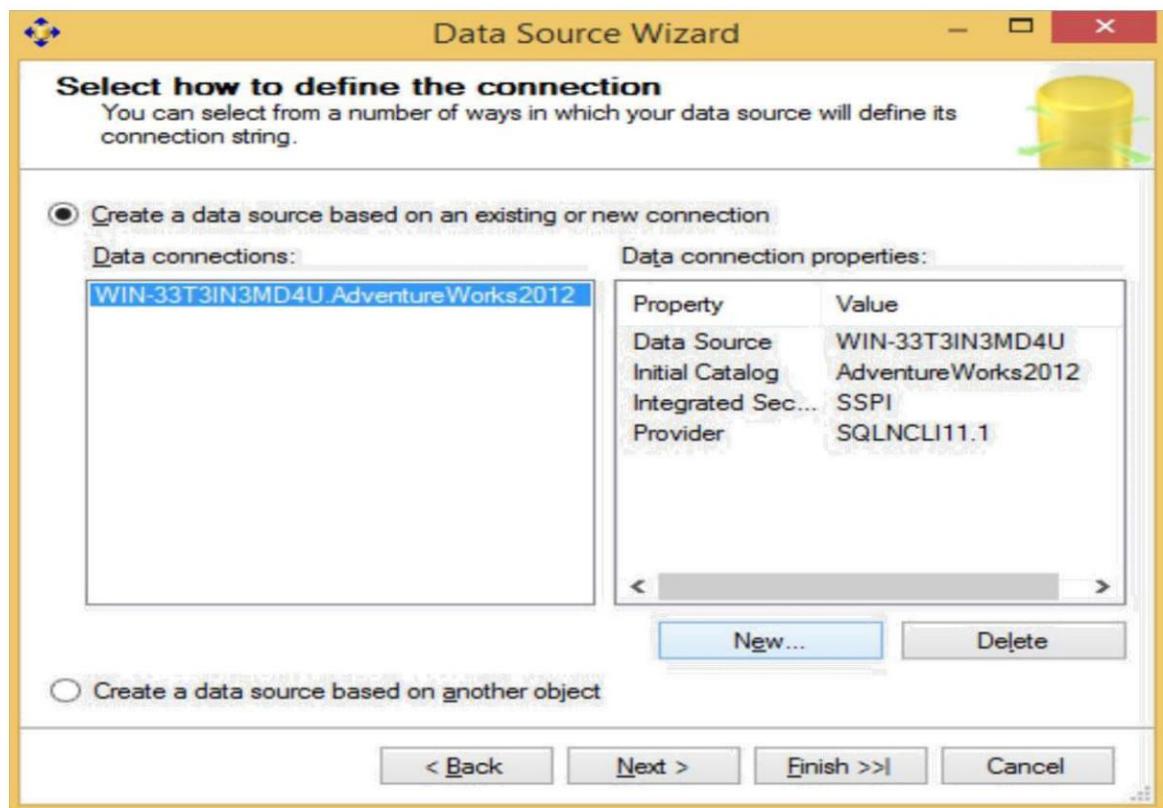
Right click on Data Sources in solution explorer → New Data Source



Data Source Wizard appears

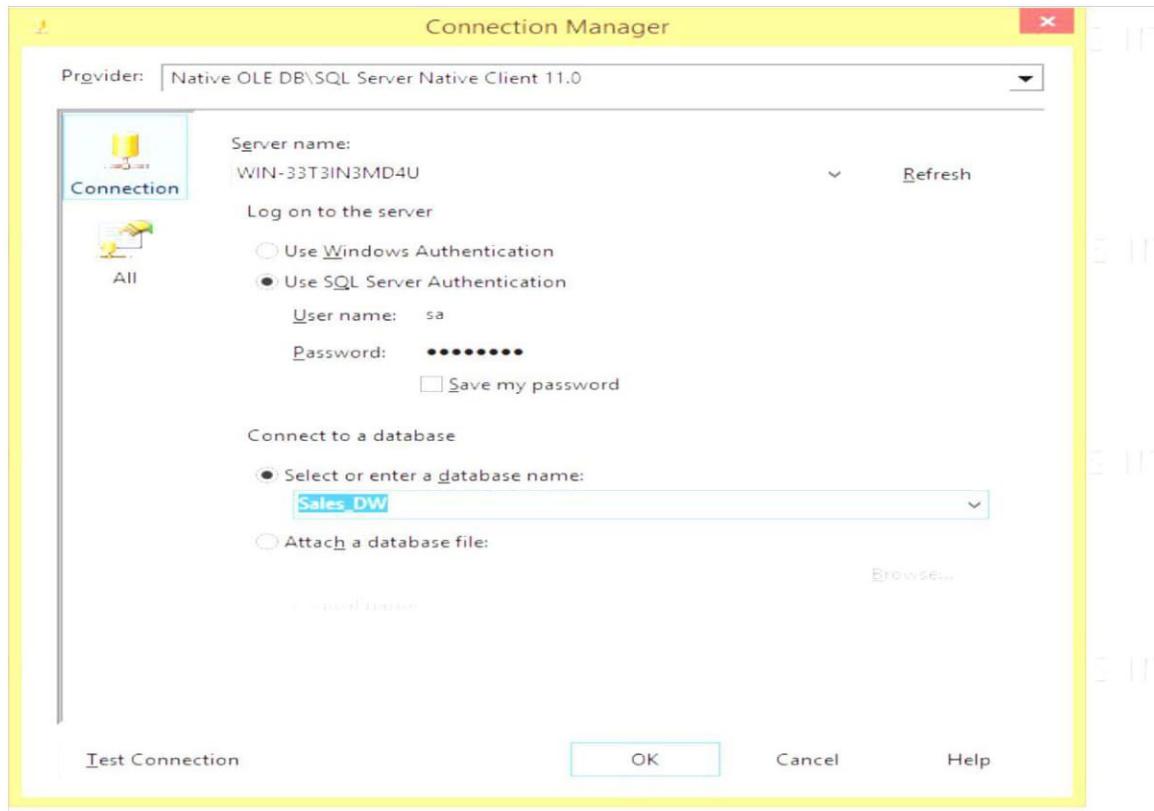


Click on New

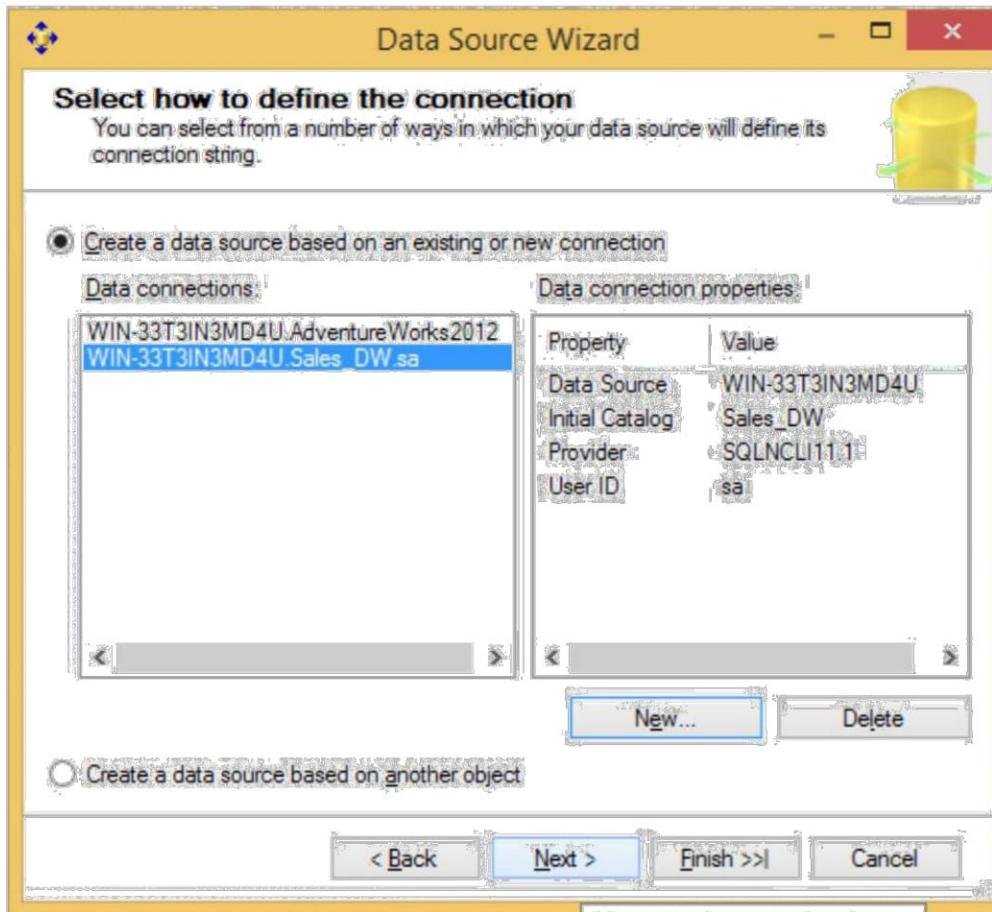


Select Server Name → select Use SQL Server Authentication → Select or enter a database name (Sales_DW)

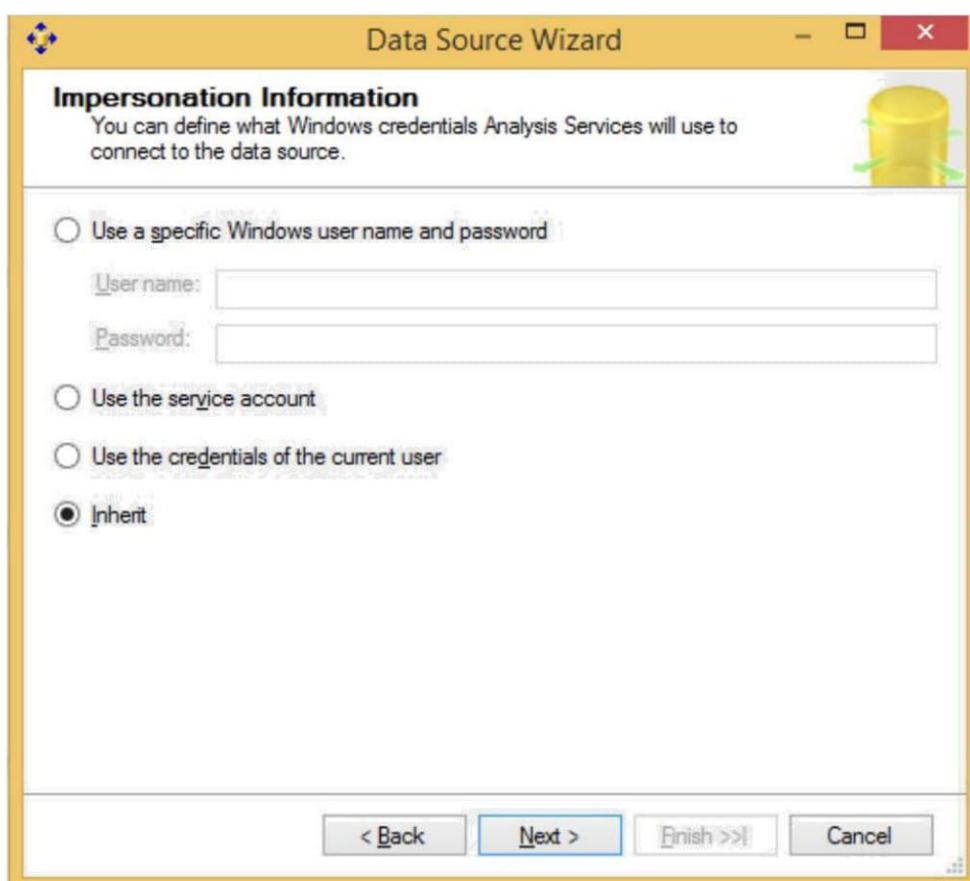
Note : Password for sa : admin123 (as given during installation of SQL 2012 full version)



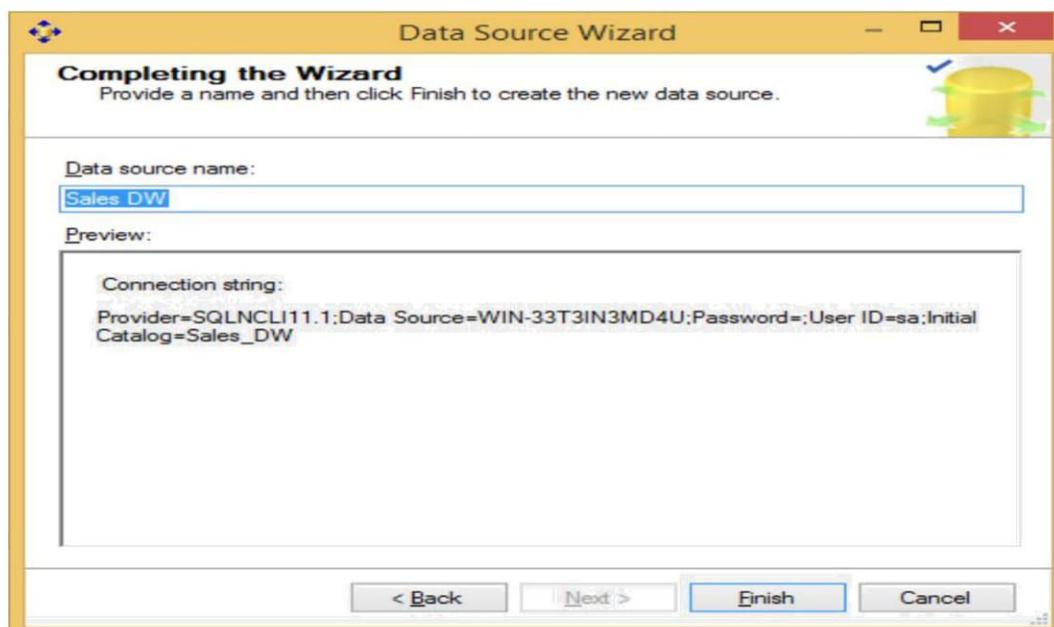
Click Next



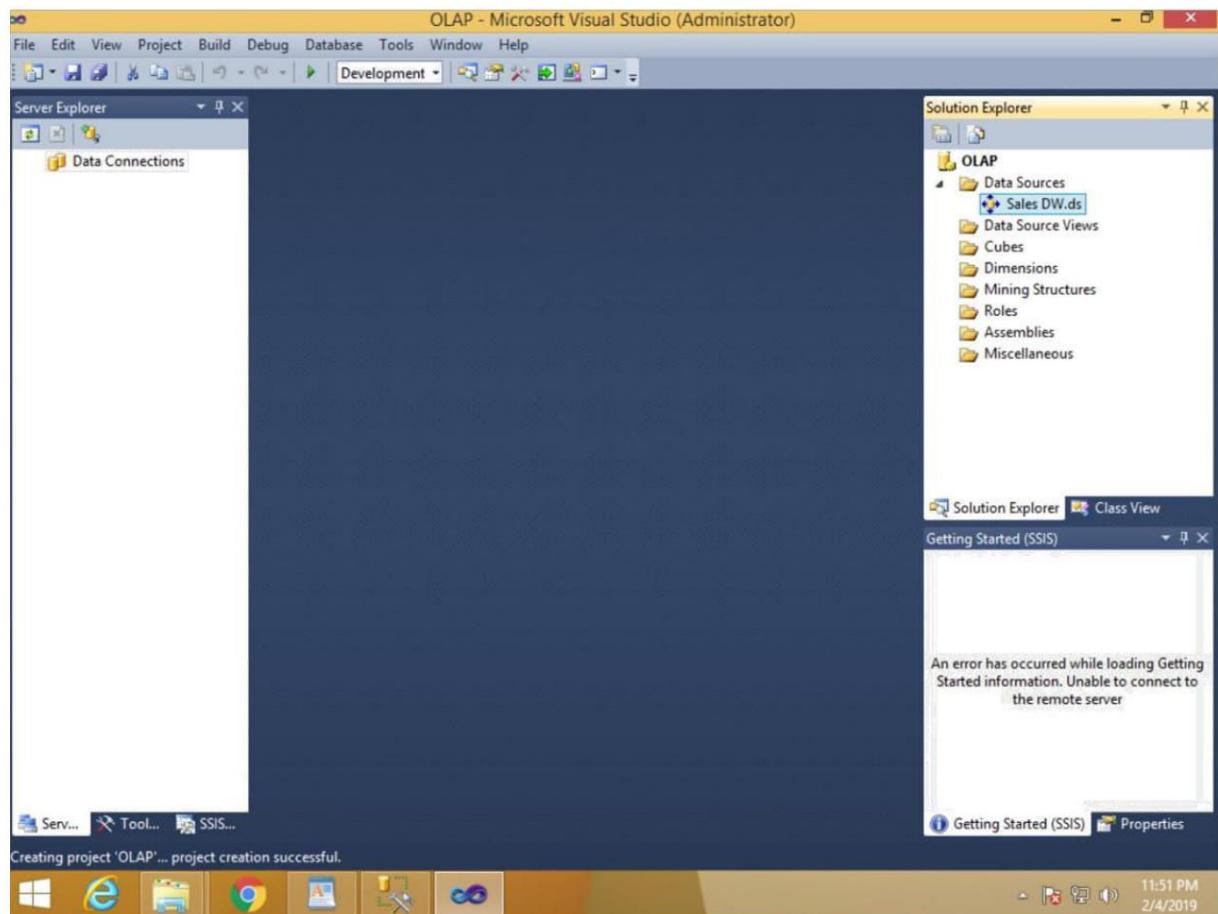
Select Inherit → Next



Click Finish

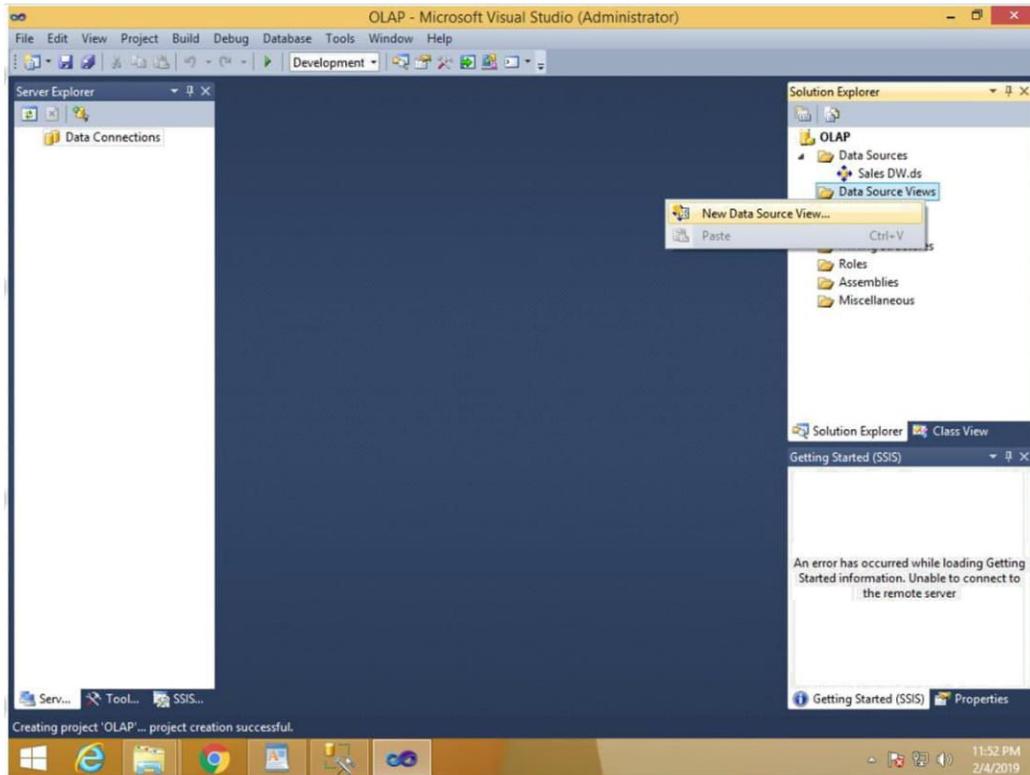


Sales_DW.ds gets created under Data Sources in Solution Explorer

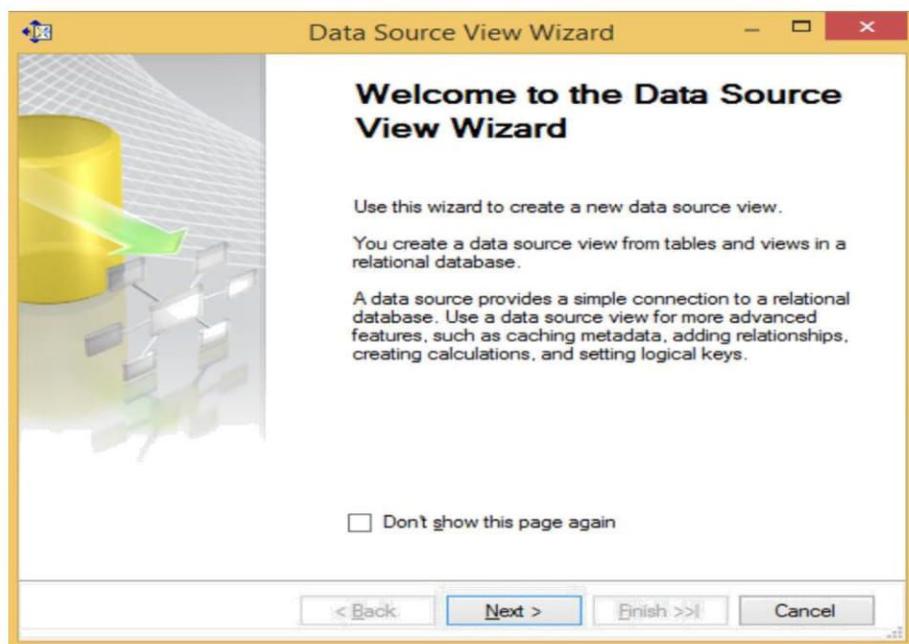


Step 3: Creating New Data Source View

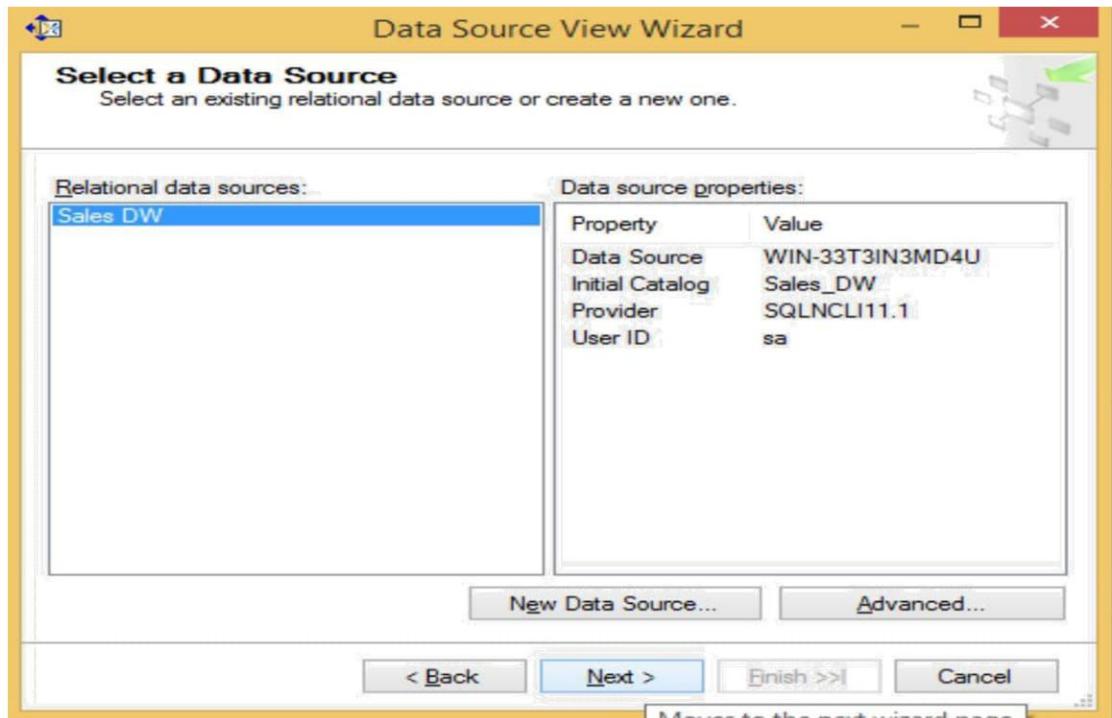
In Solution explorer right click on Data Source View → Select New Data Source View



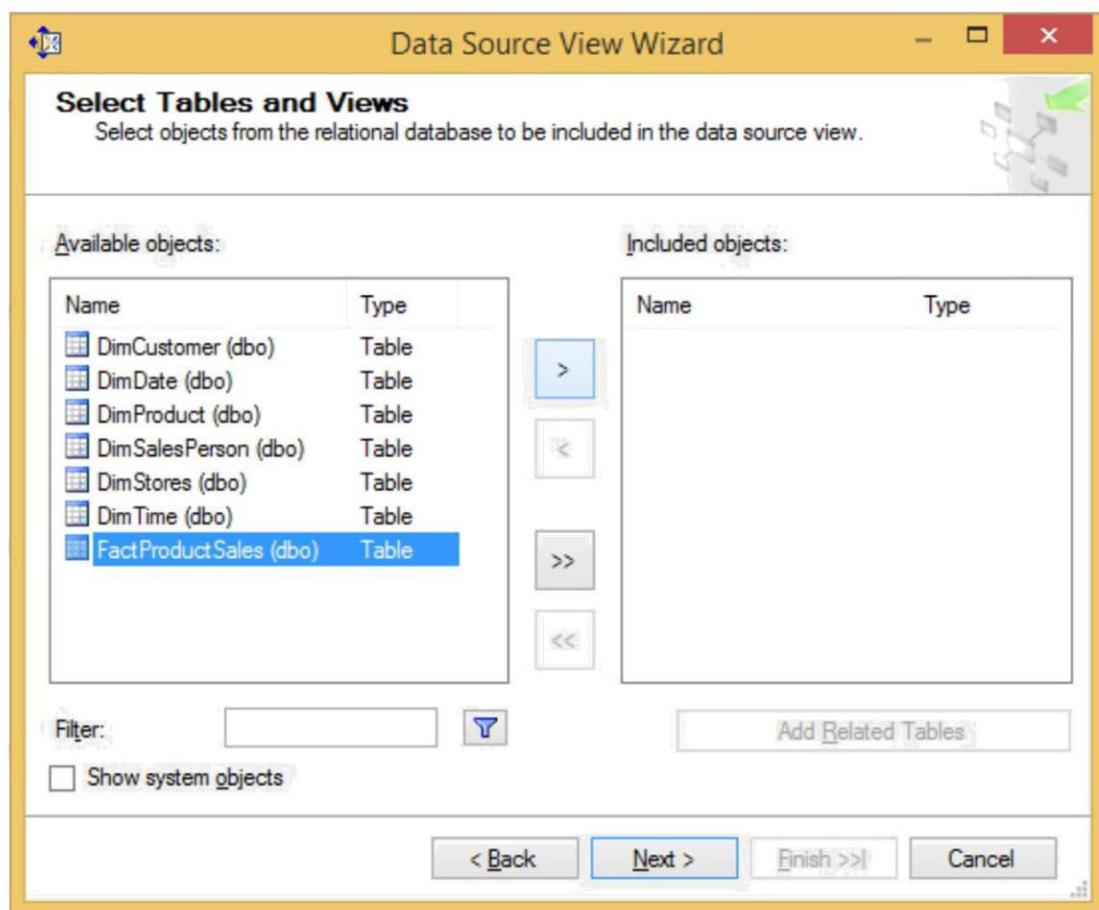
Click Next

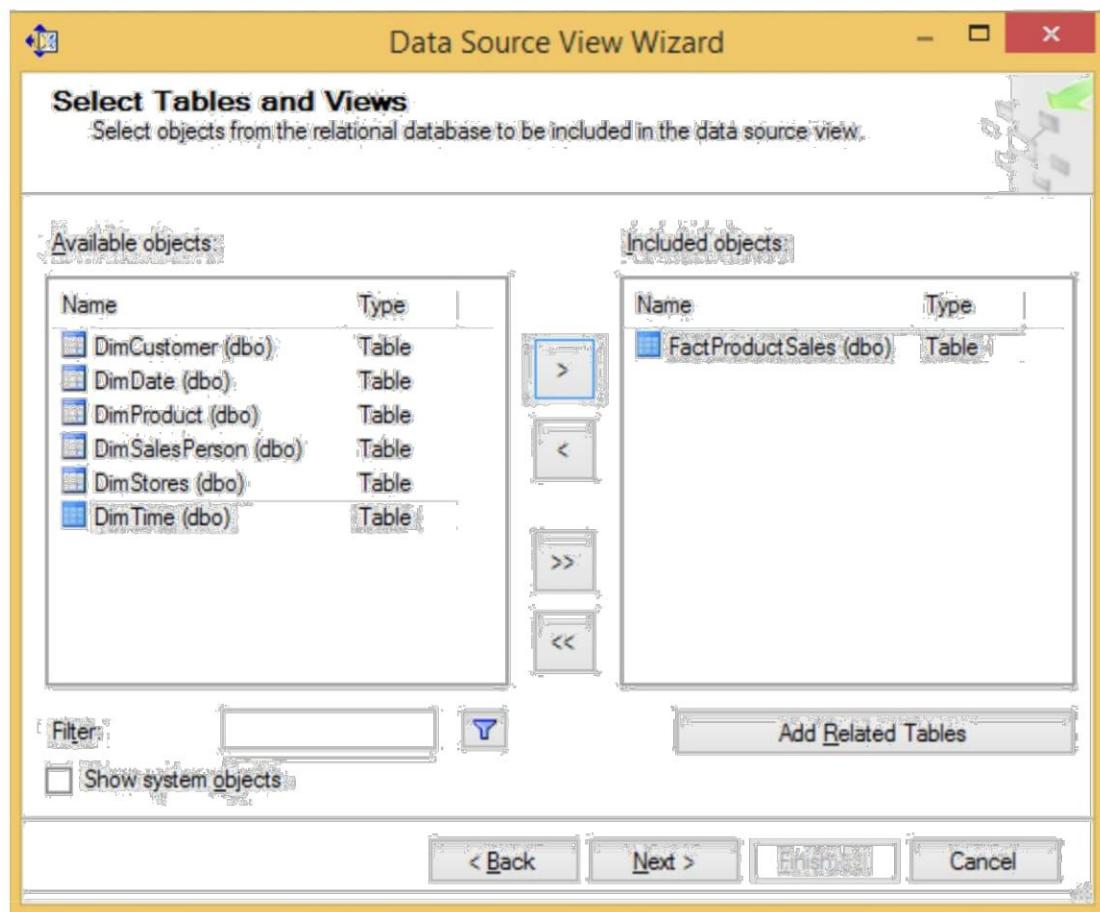


Click Next

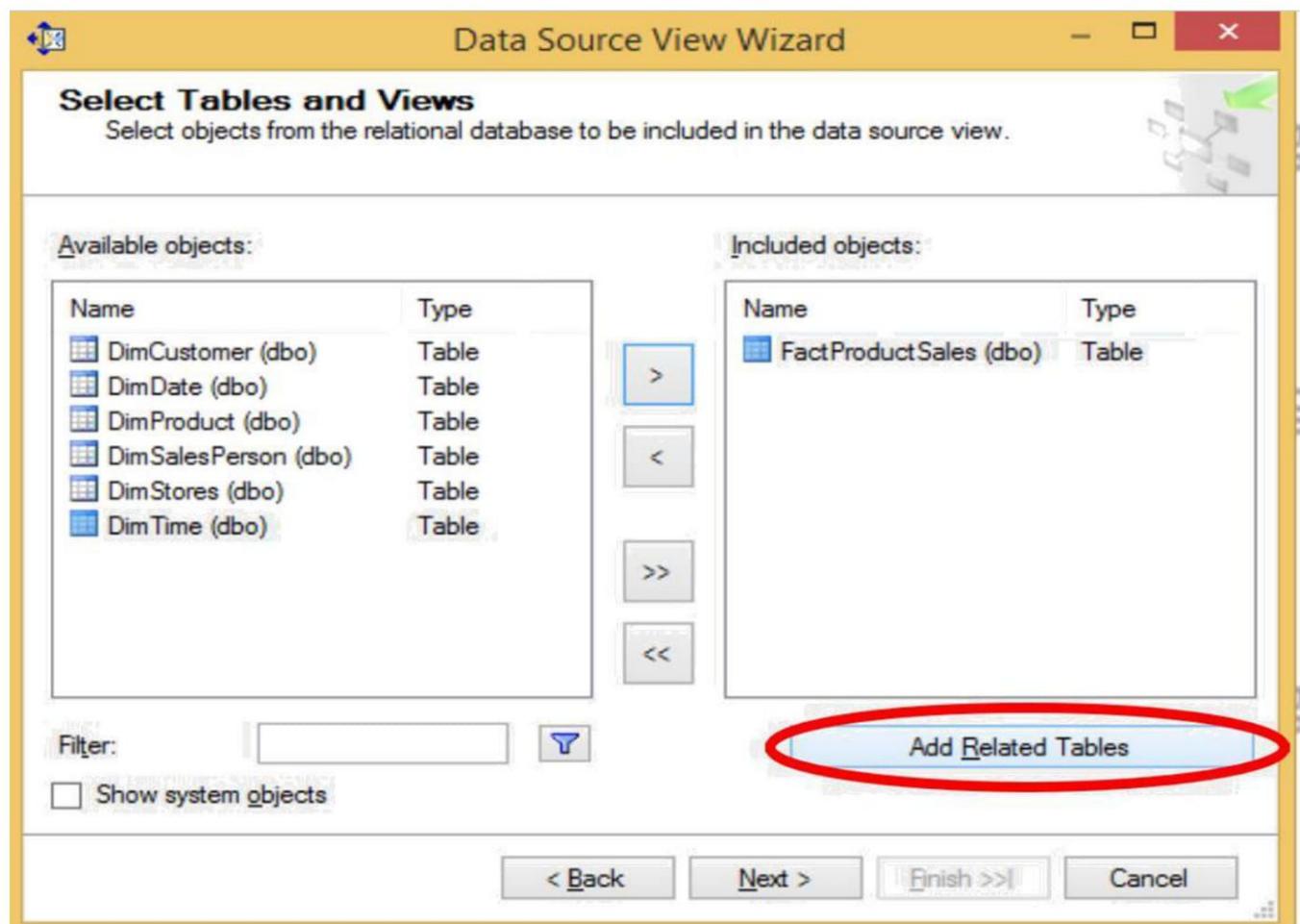


Select FactProductSales(dbo) from Available objects and put in Included Objects by clicking on .

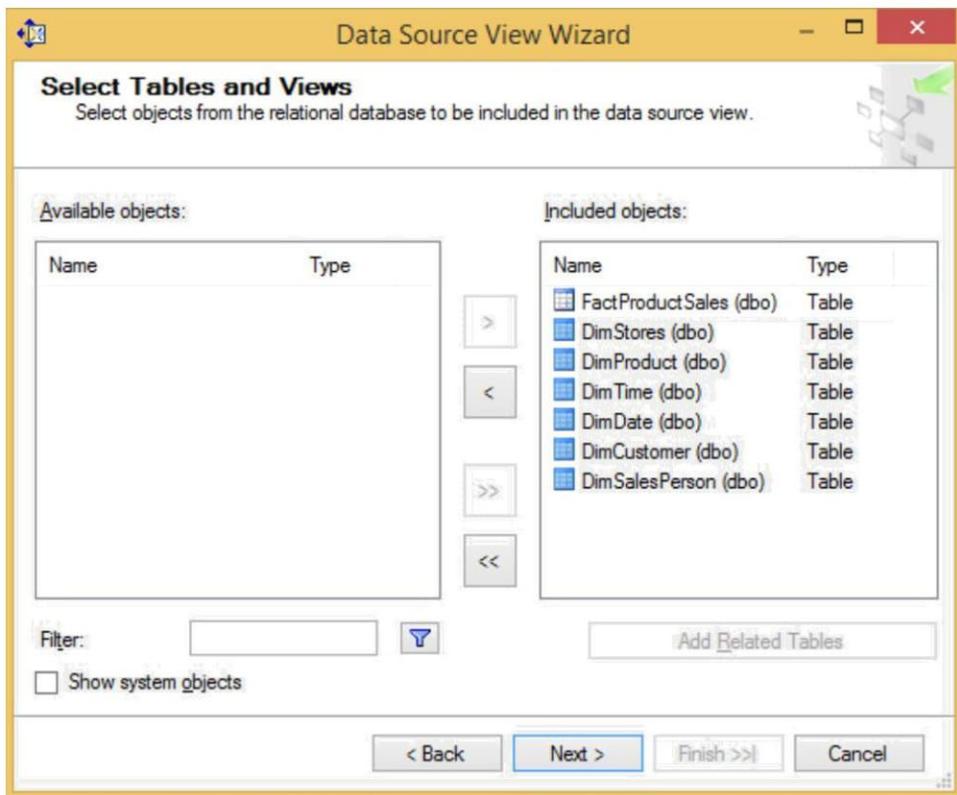




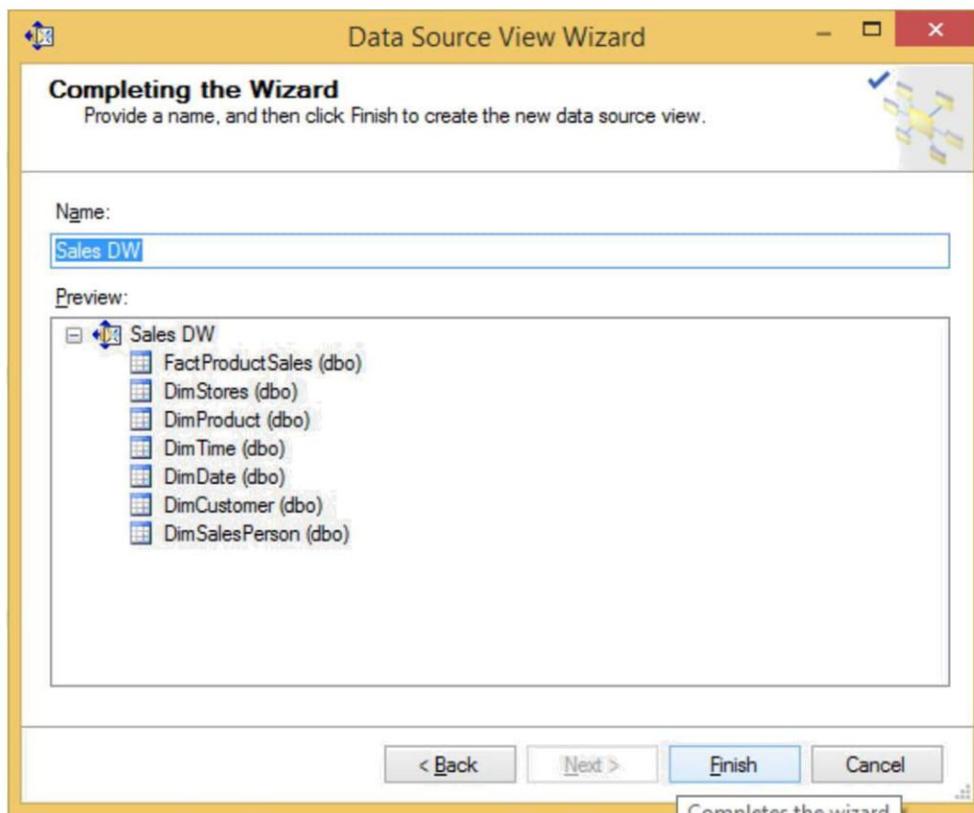
Click on Add Related Tables



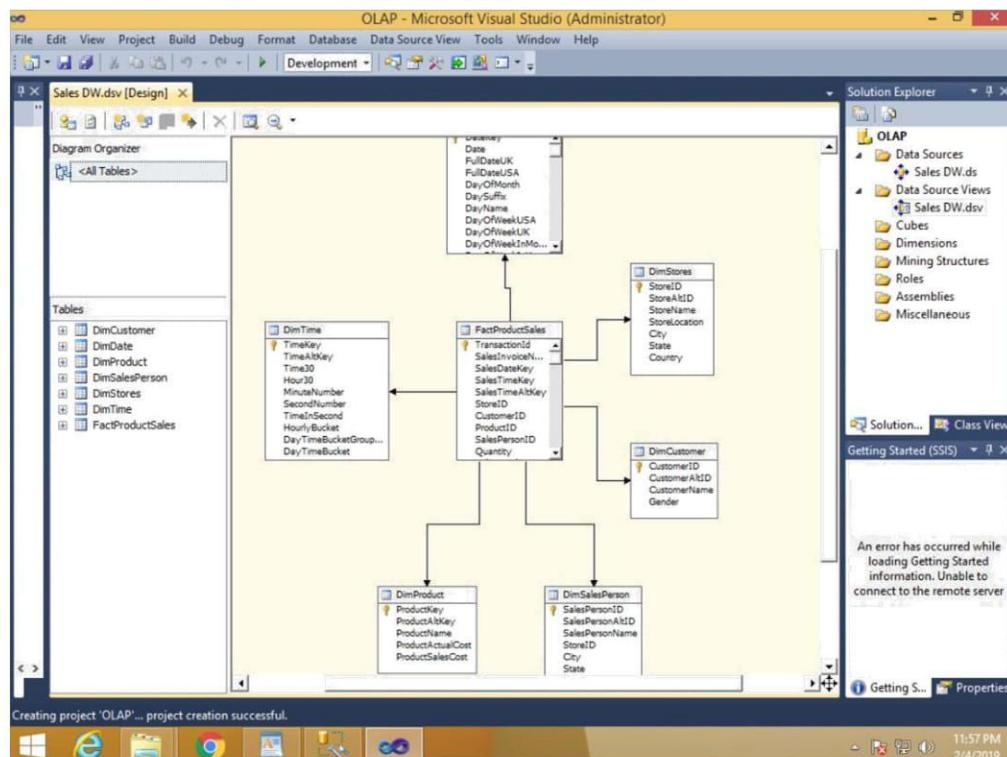
Click Next



Click Finish

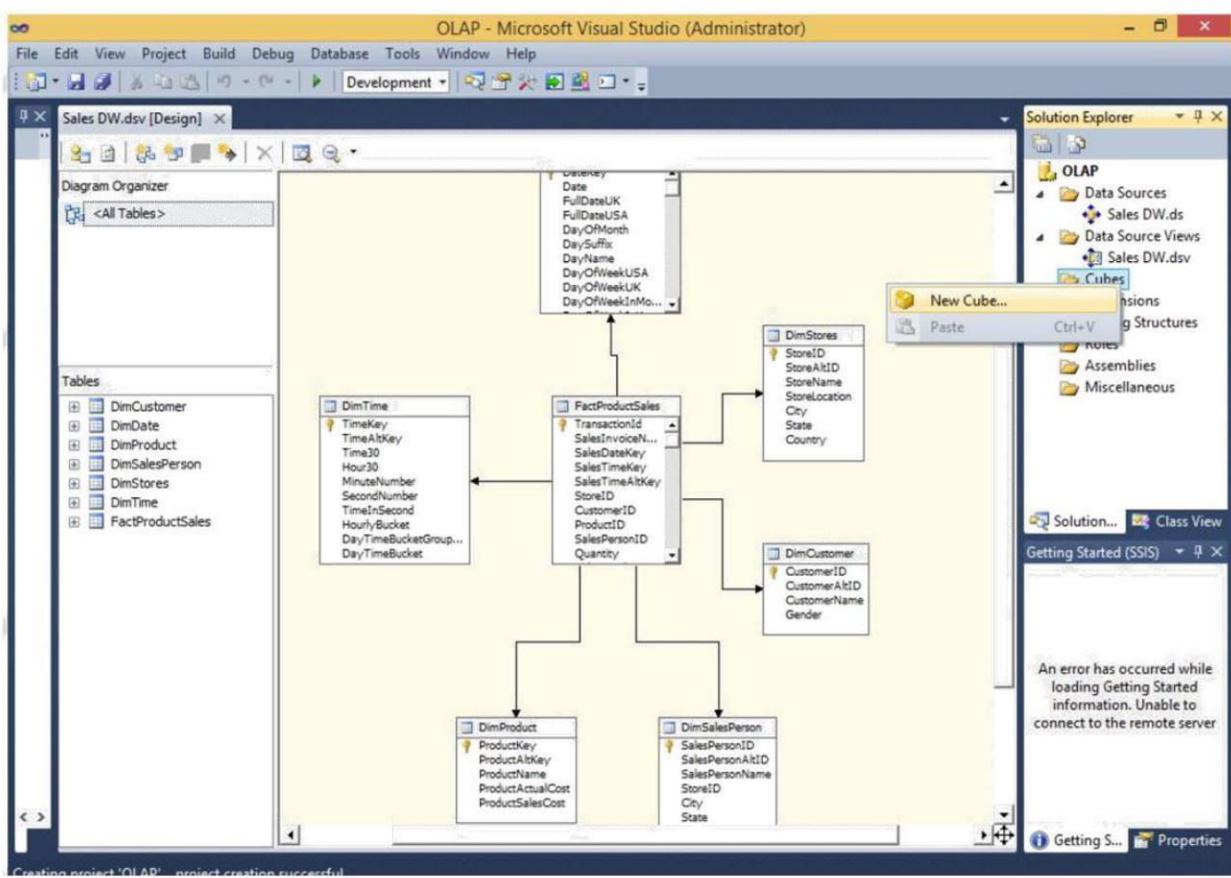


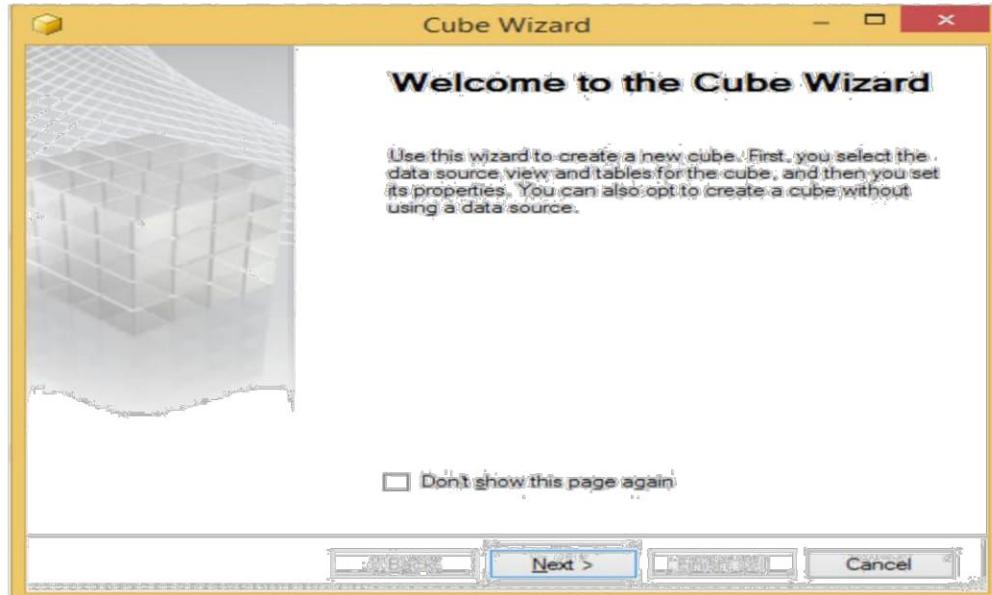
Sales DW.dsv appears in Data Source Views in Solution Explorer.



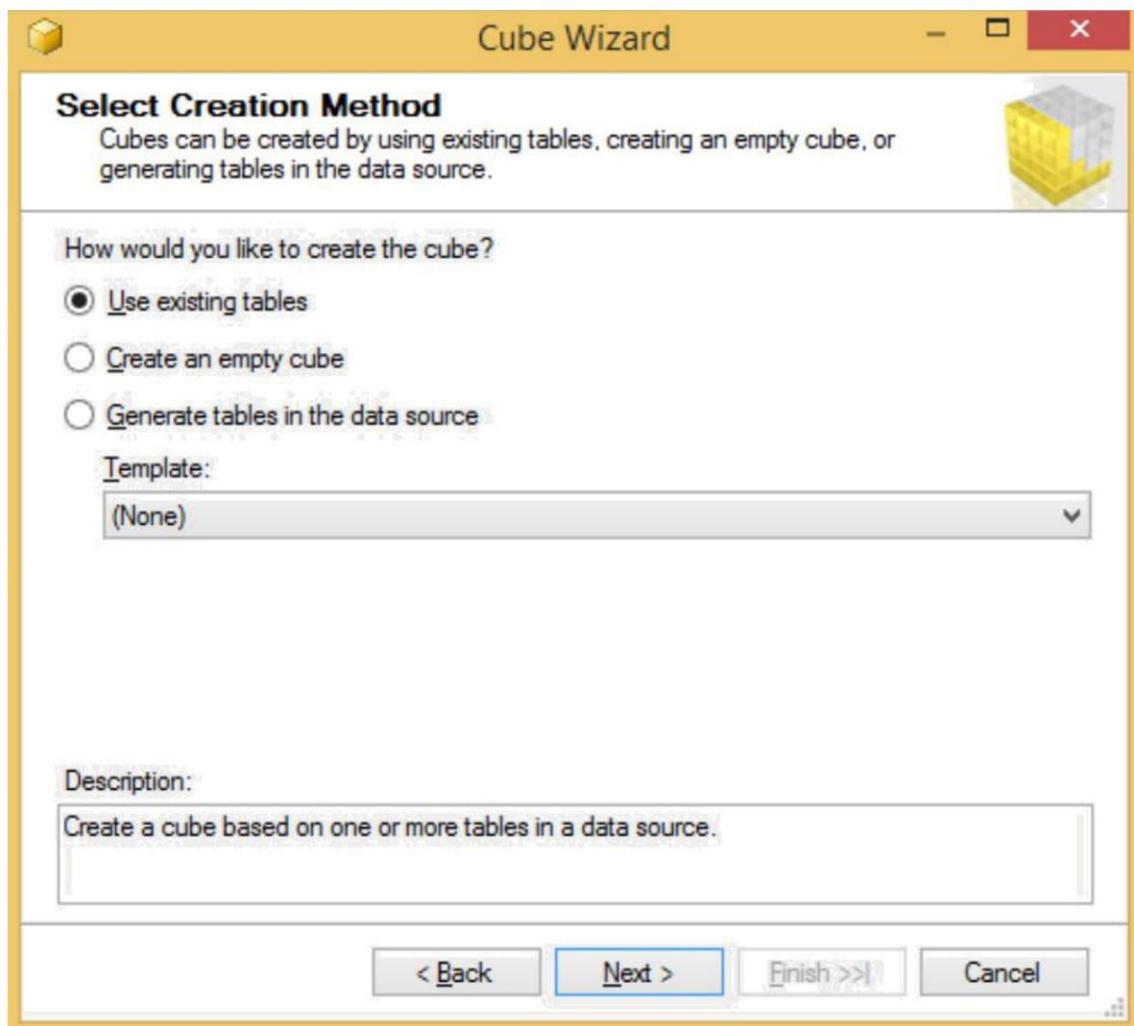
Step 4: Creating new cube

Right click on Cubes → New Cube

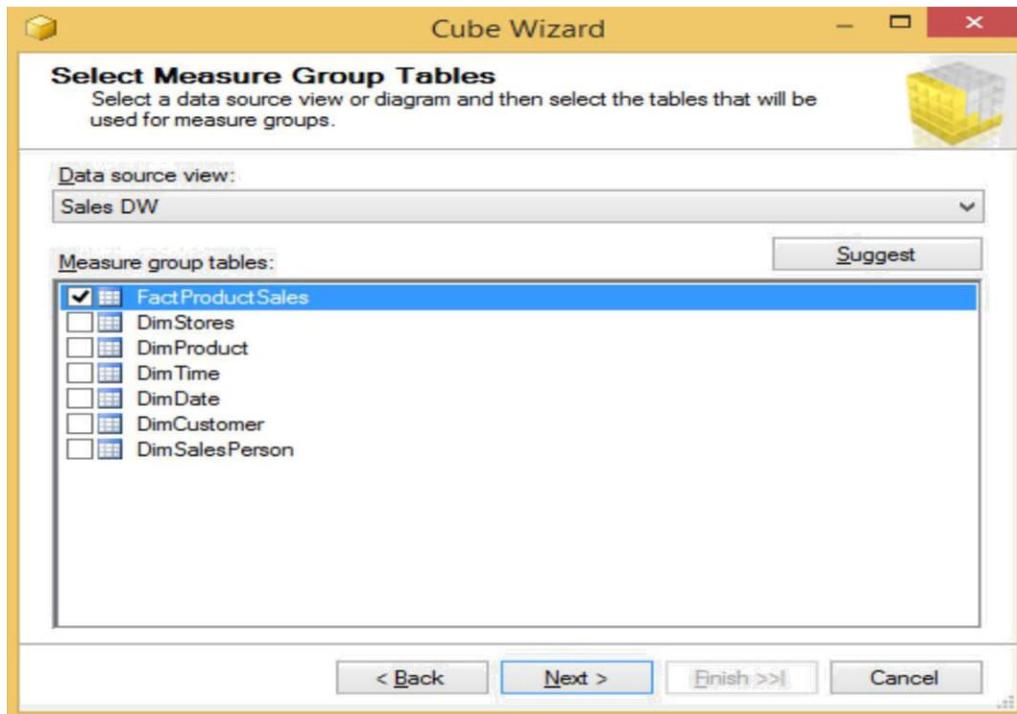




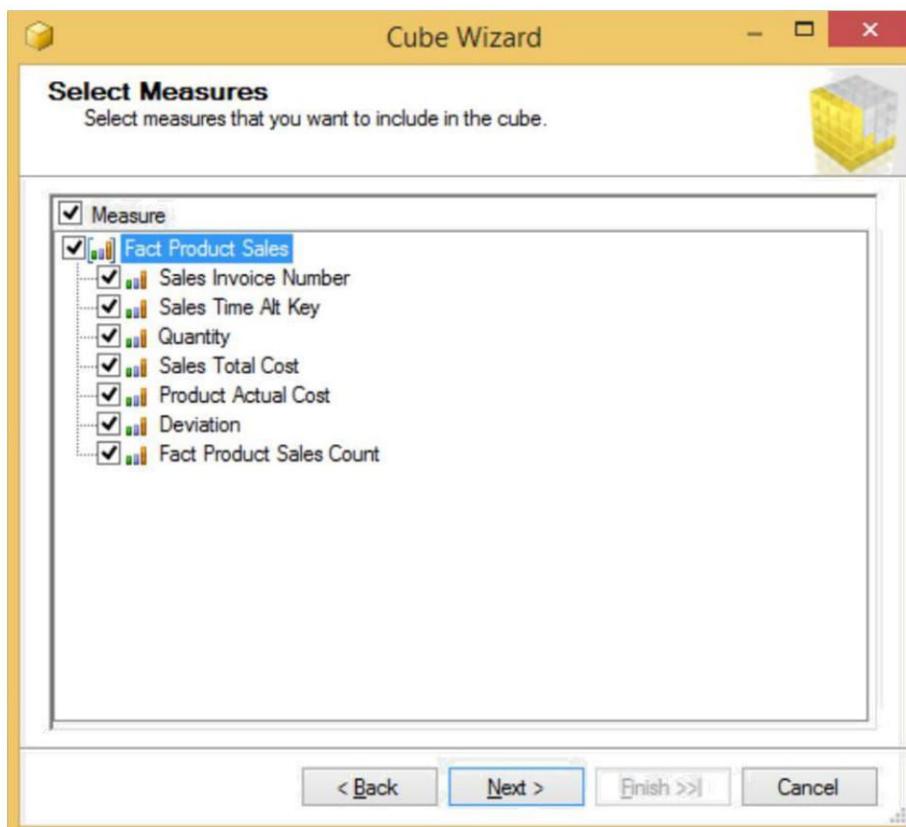
Select Use existing tables in Select Creation Method → Next



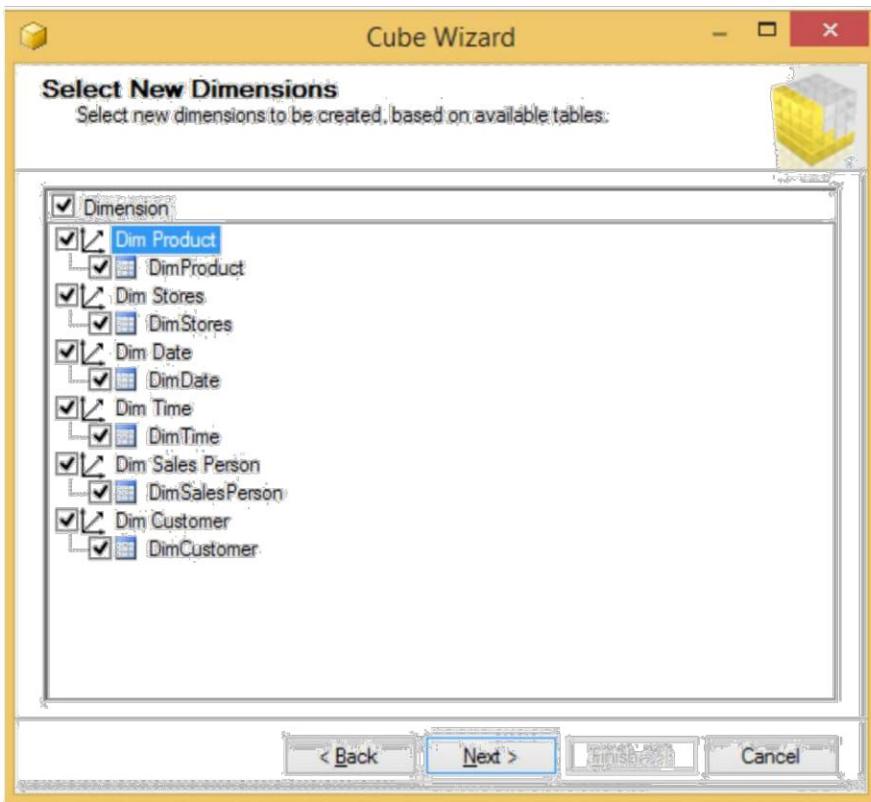
In Select Measure Group Tables → Select FactProductSales → Click Next



In Select Measures → check all measures → Next



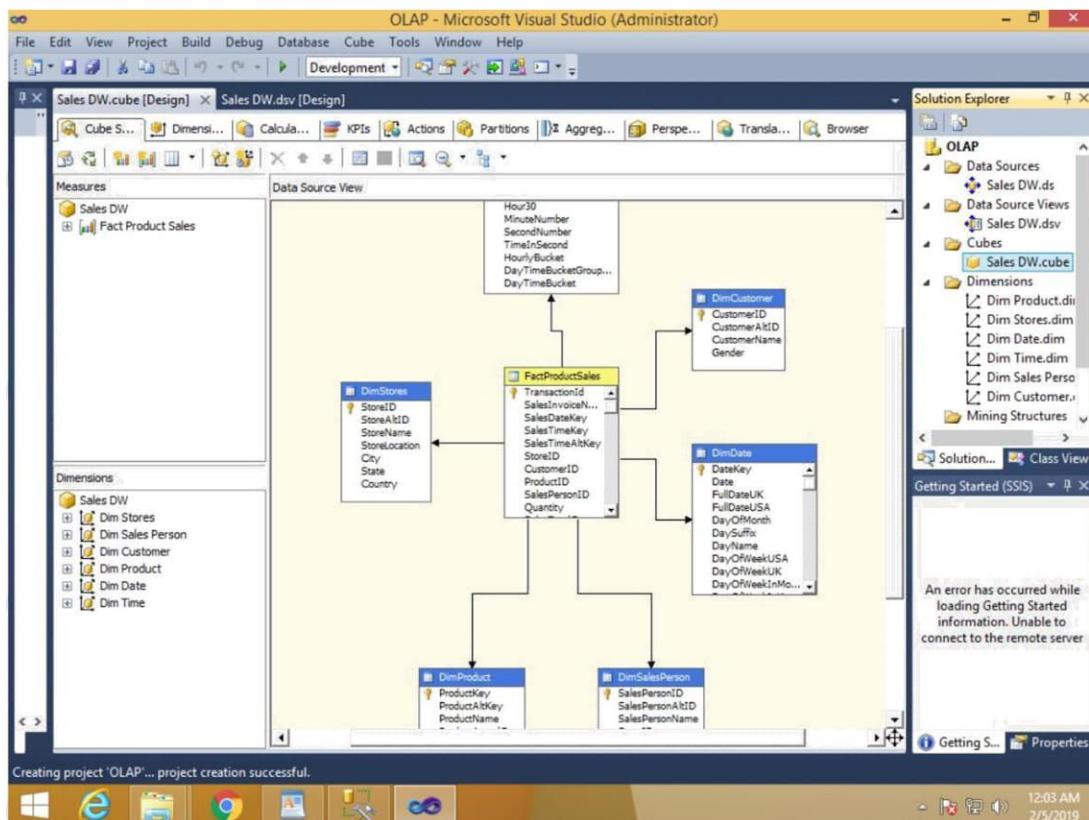
In Select New Dimensions → Check all Dimensions → Next



Click on Finish

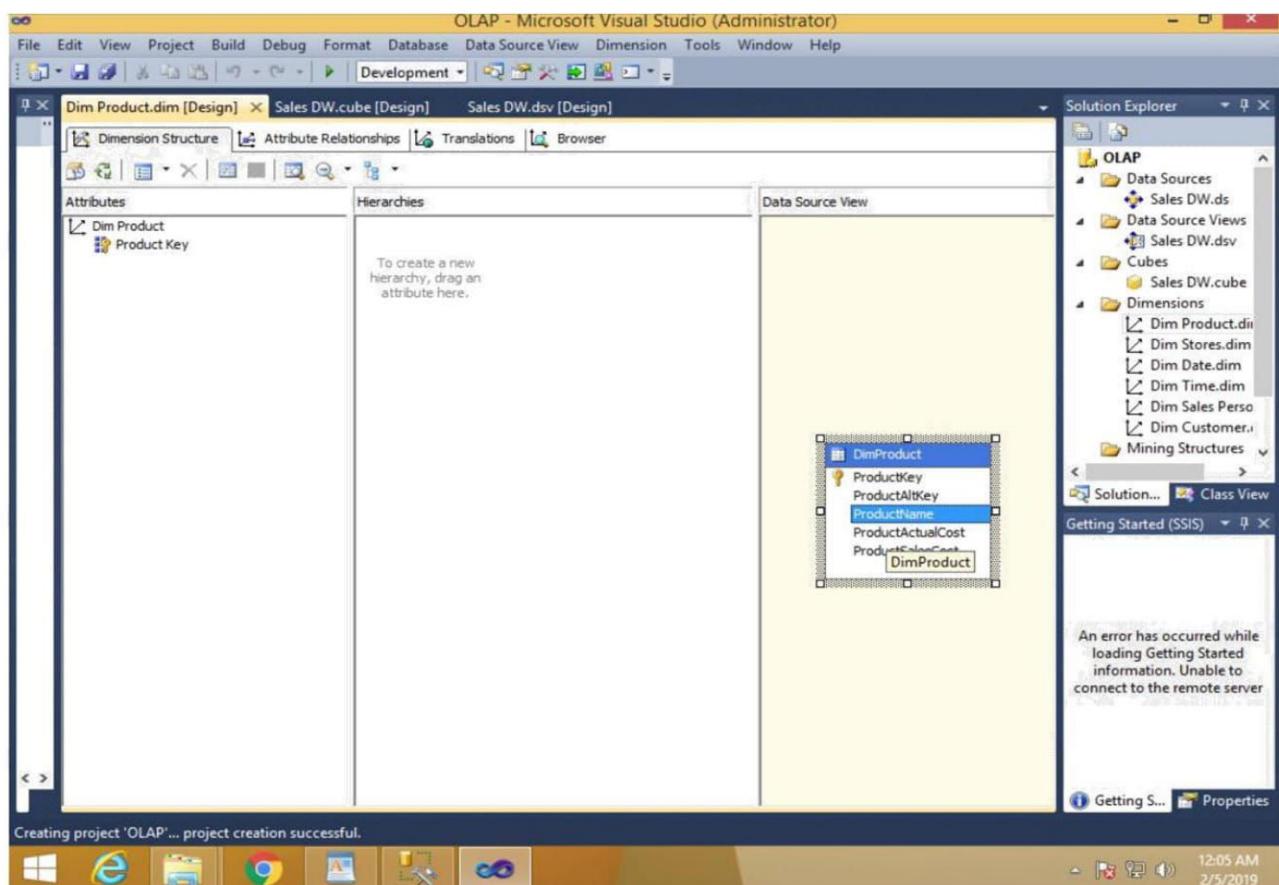


Sales_DW.cube is created

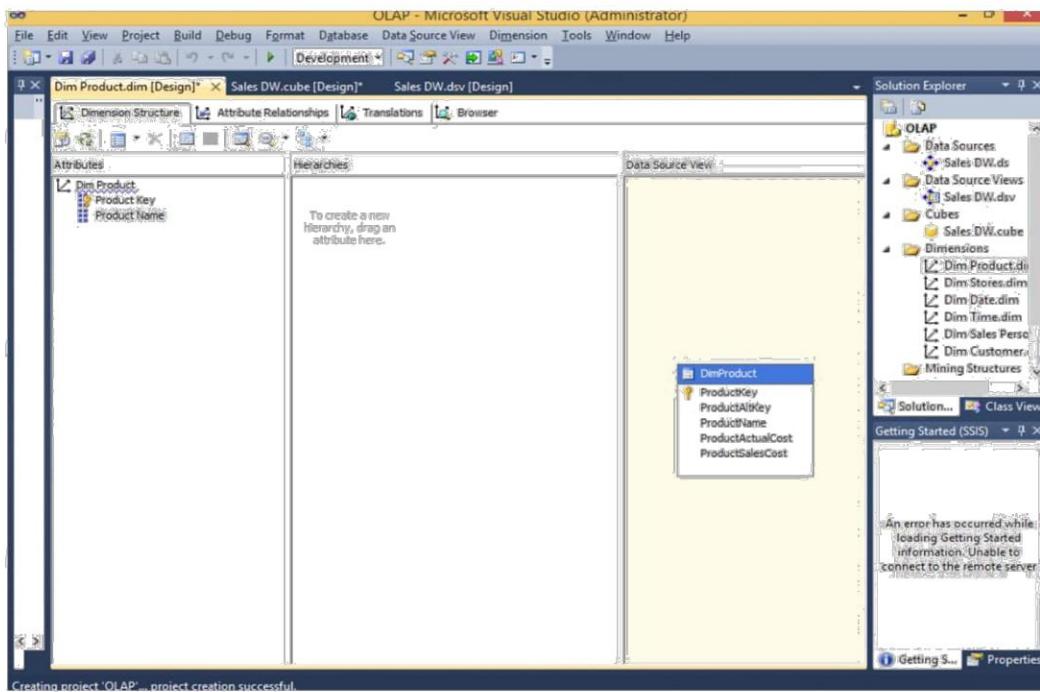


Step 5: Dimension Modification

In dimension tab → Double Click Dim Product.dim



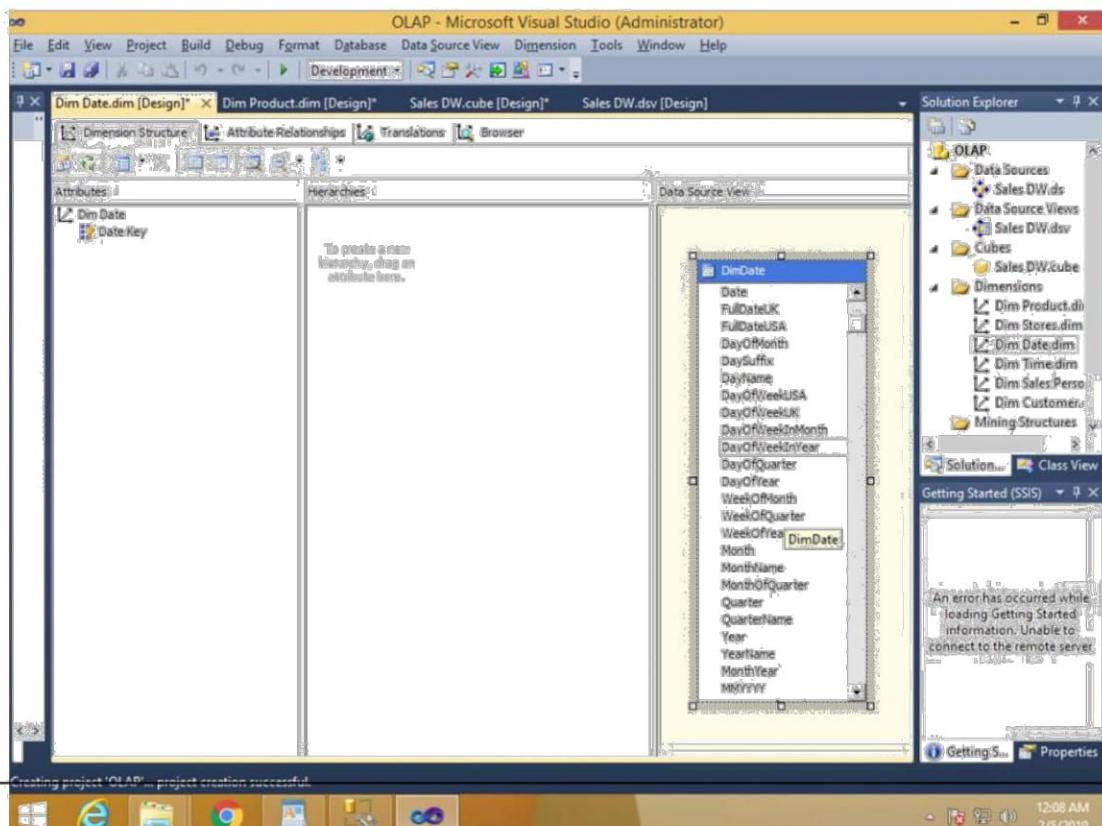
Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side

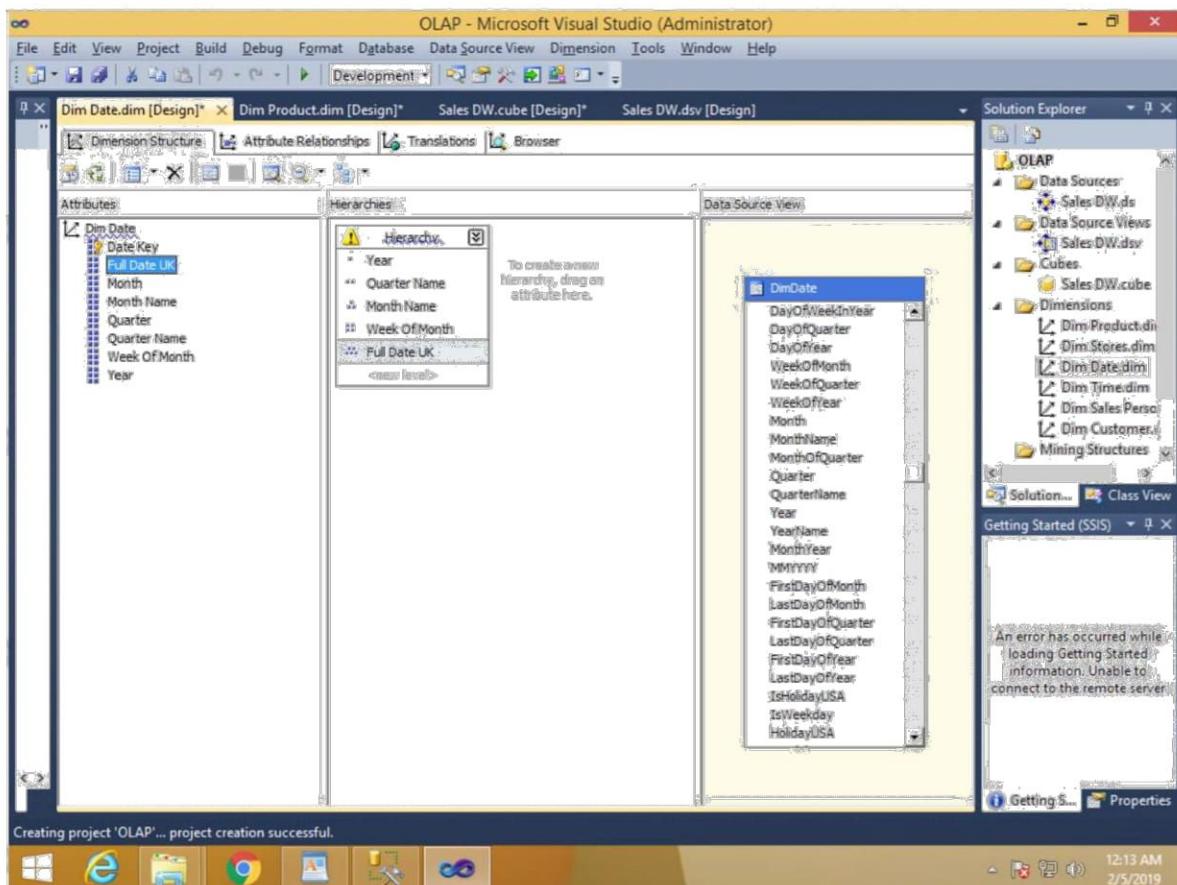


Step 6: Creating Attribute Hierarchy in Date Dimension

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.

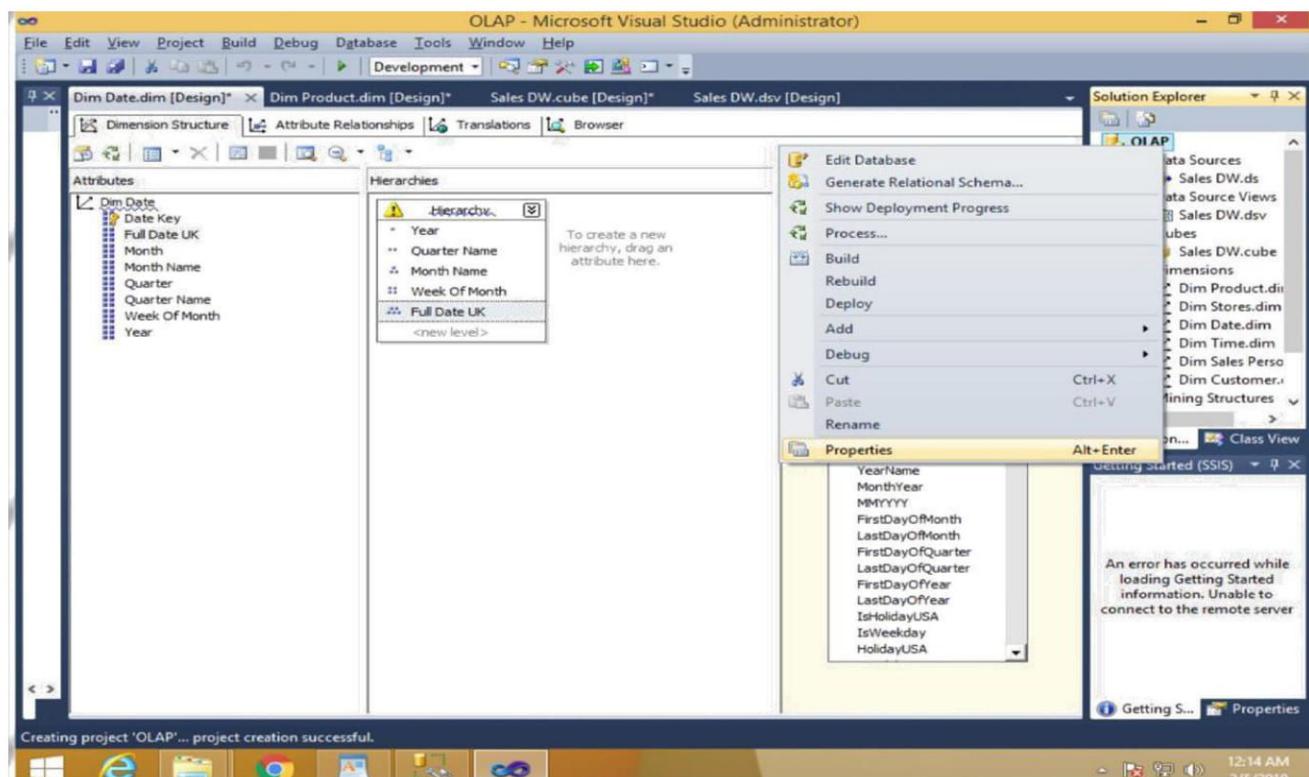
Drag fields in sequence from Attributes to Hierarchy window (Year, Quarter Name, Month Name, Week of the Month, Full Date UK)



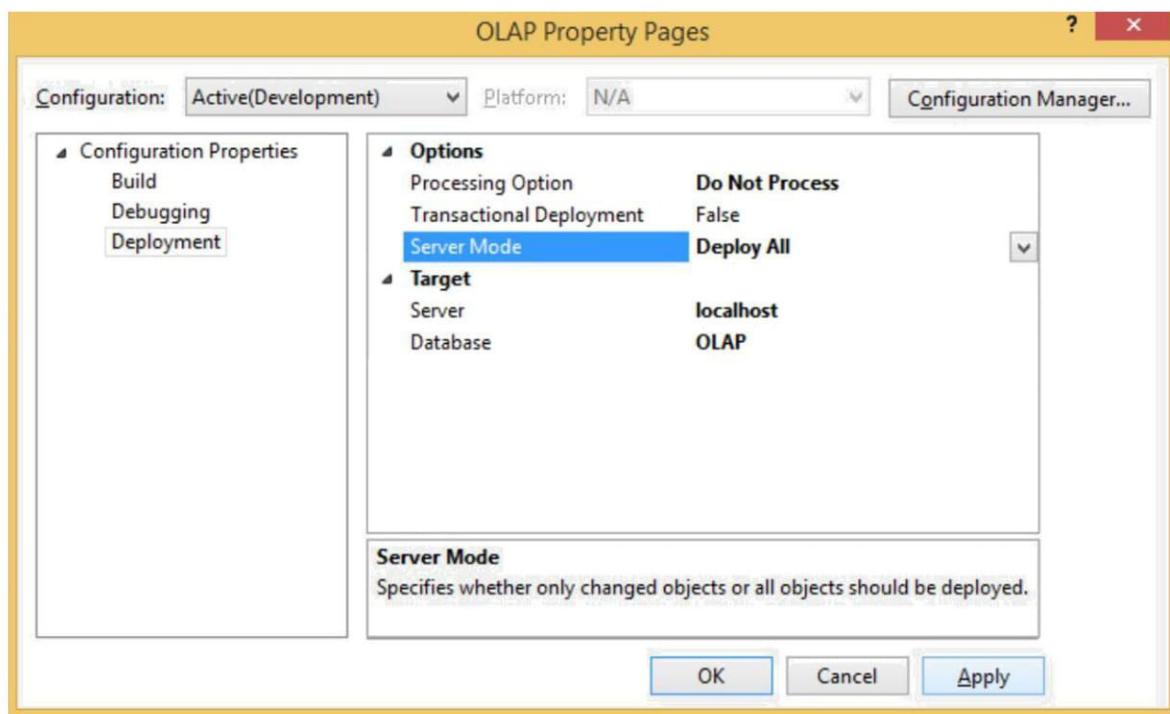
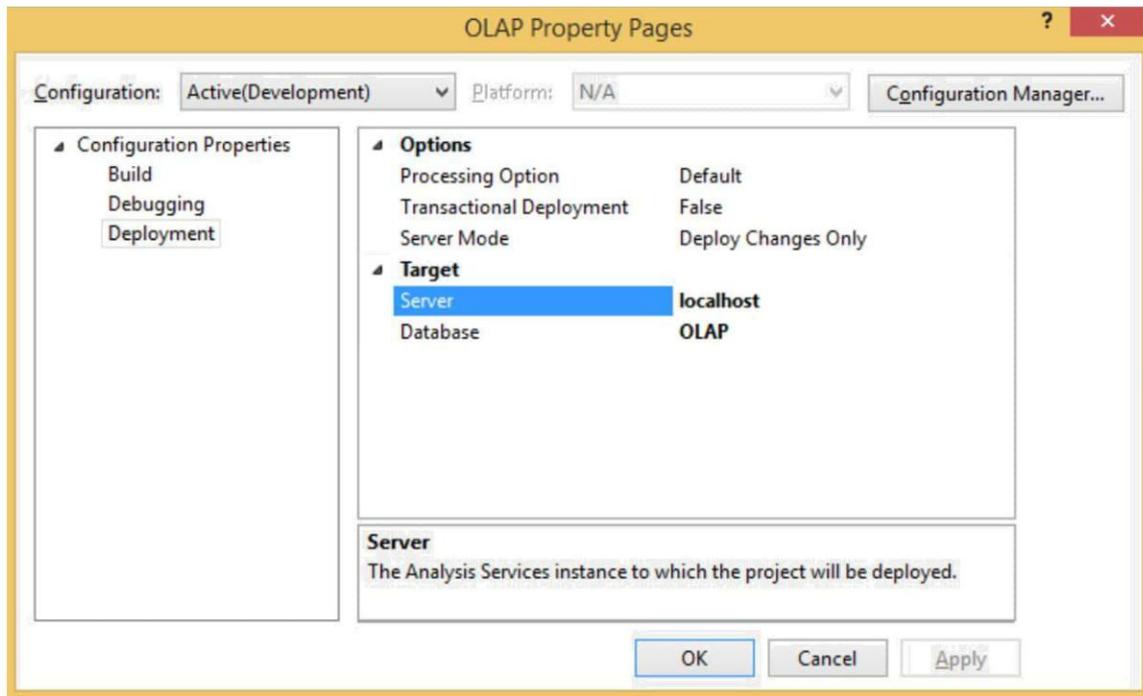


Step 7: Deploy Cube

Right click on Project name → Properties

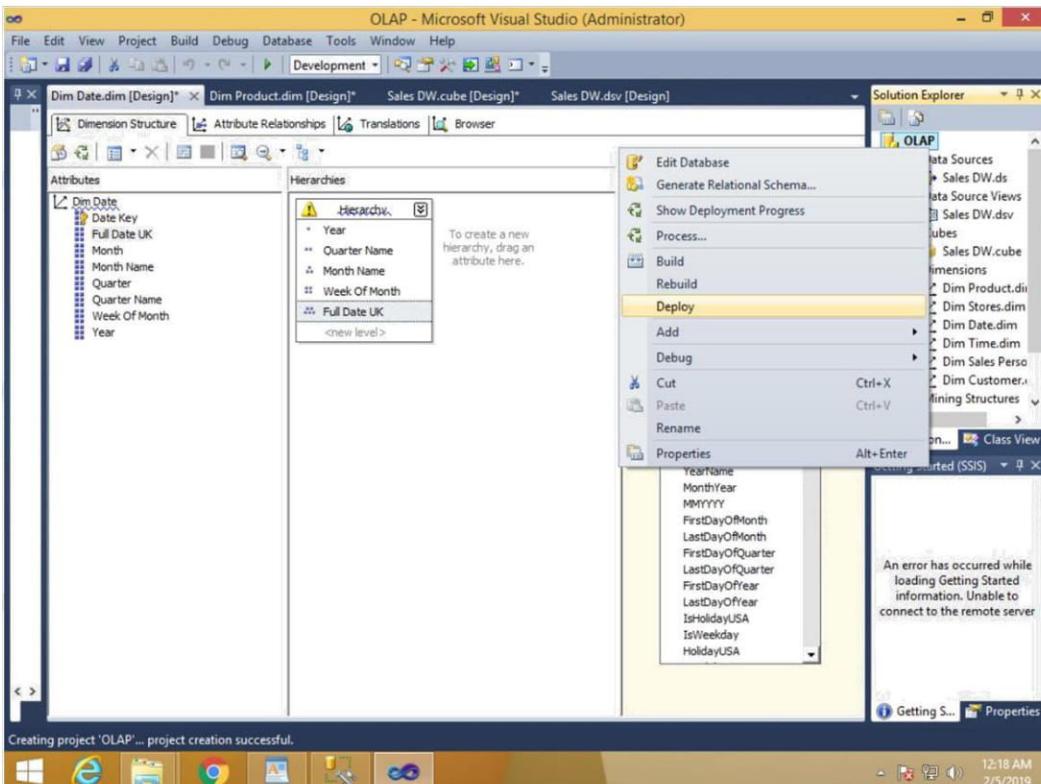


This window appears

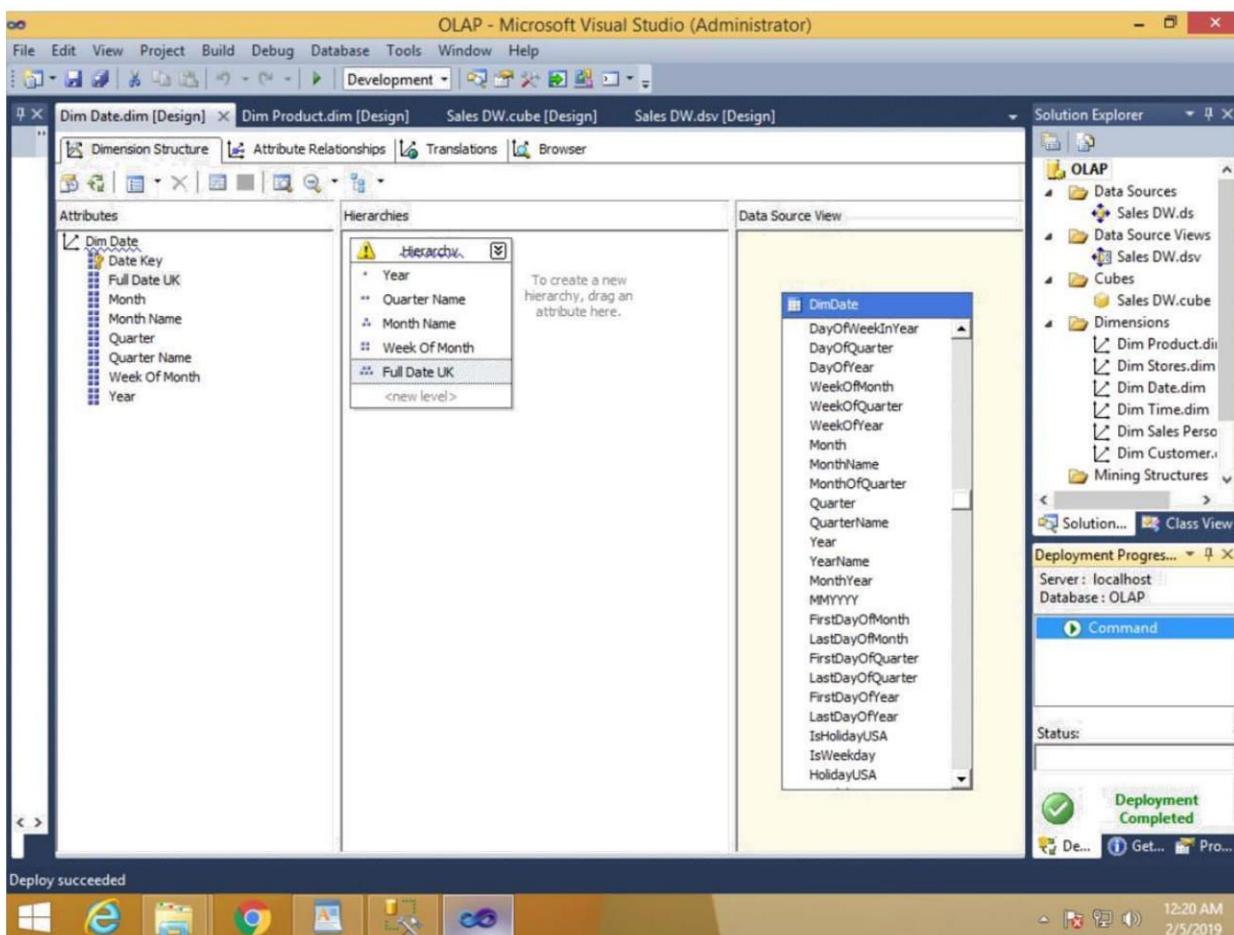


Do following changes and click on Apply & ok

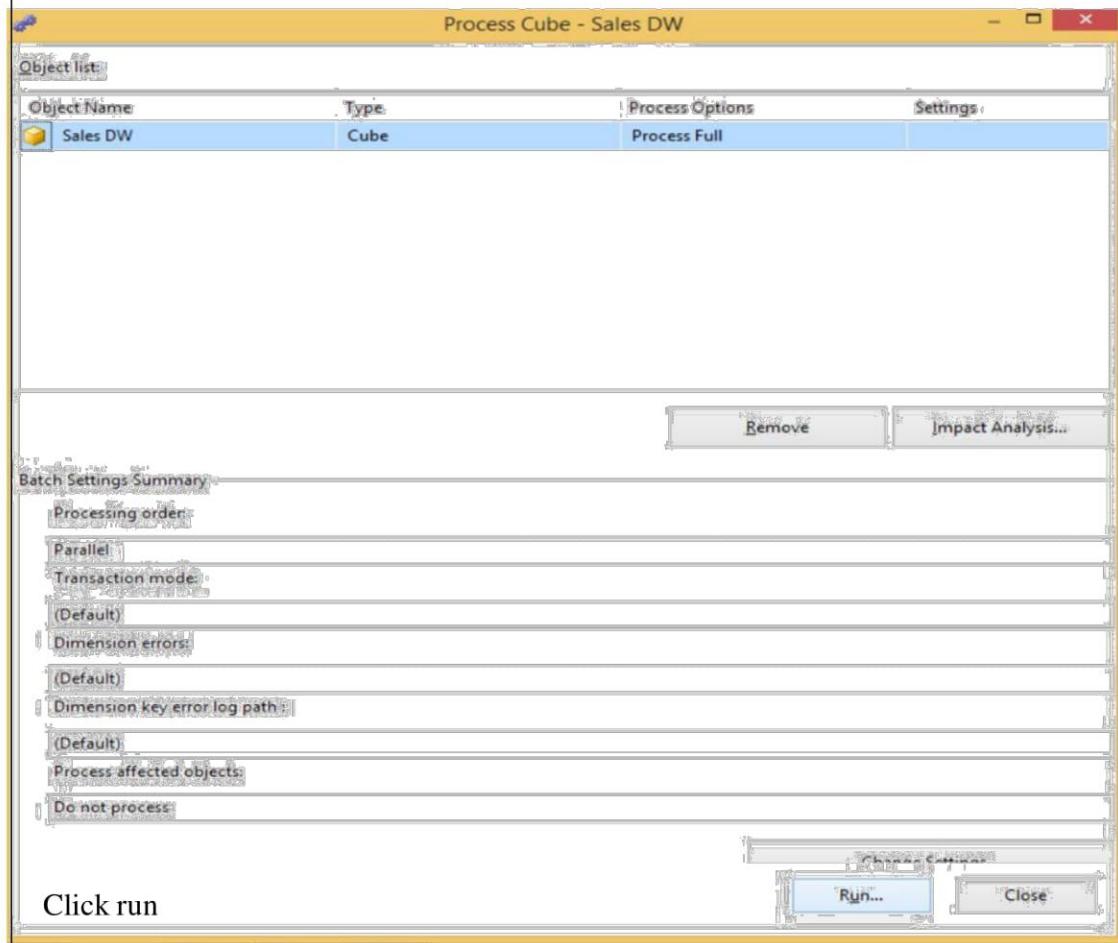
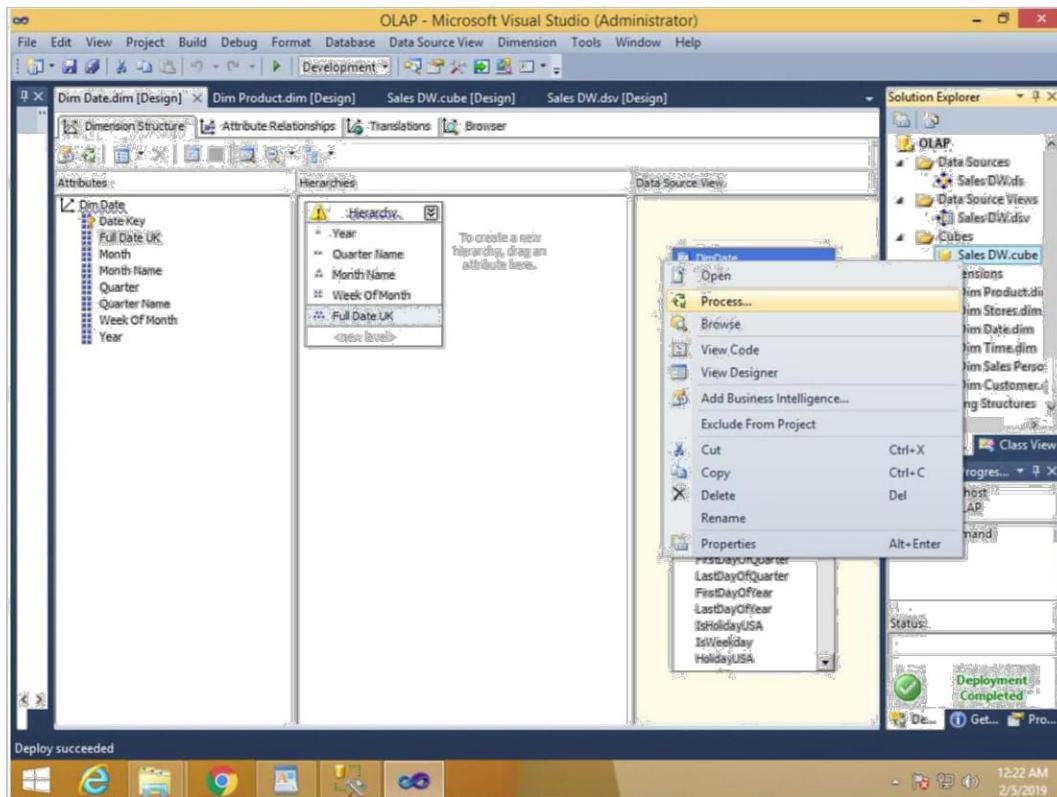
Right click on project name → Deploy

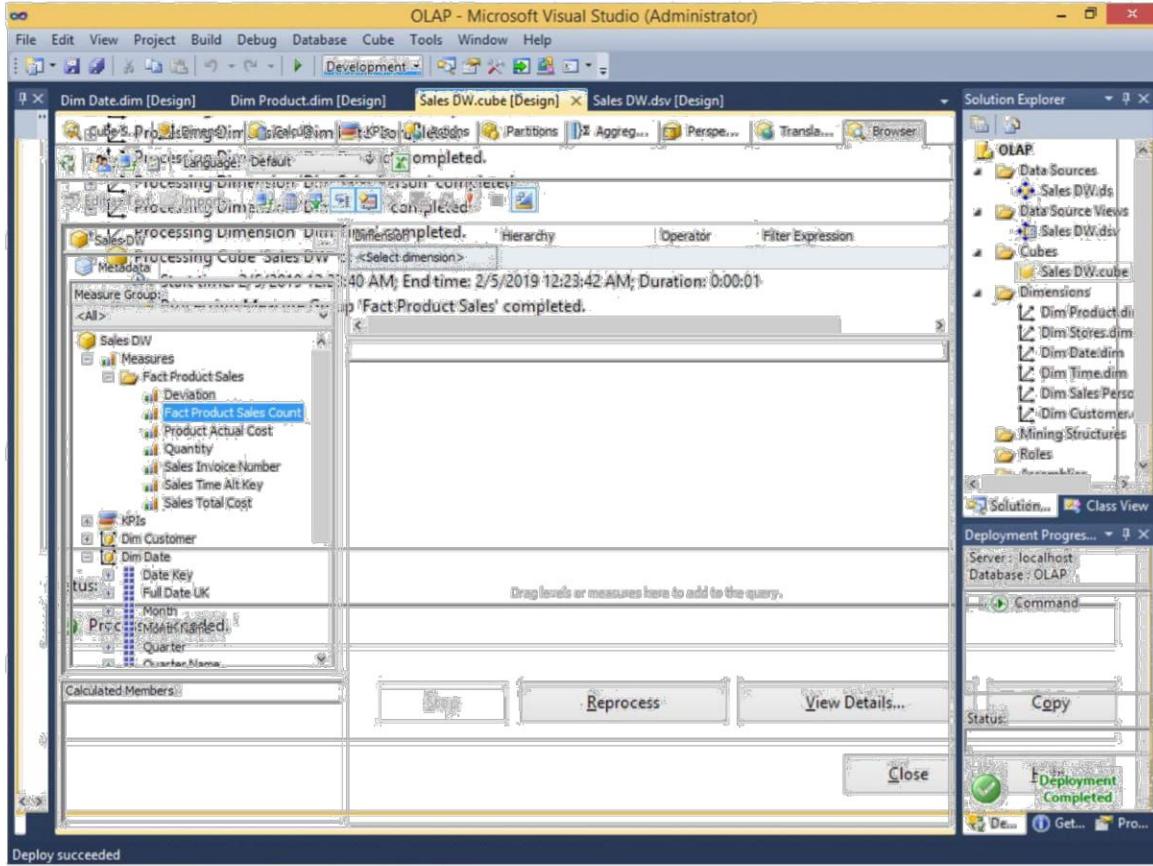


Deployment successful

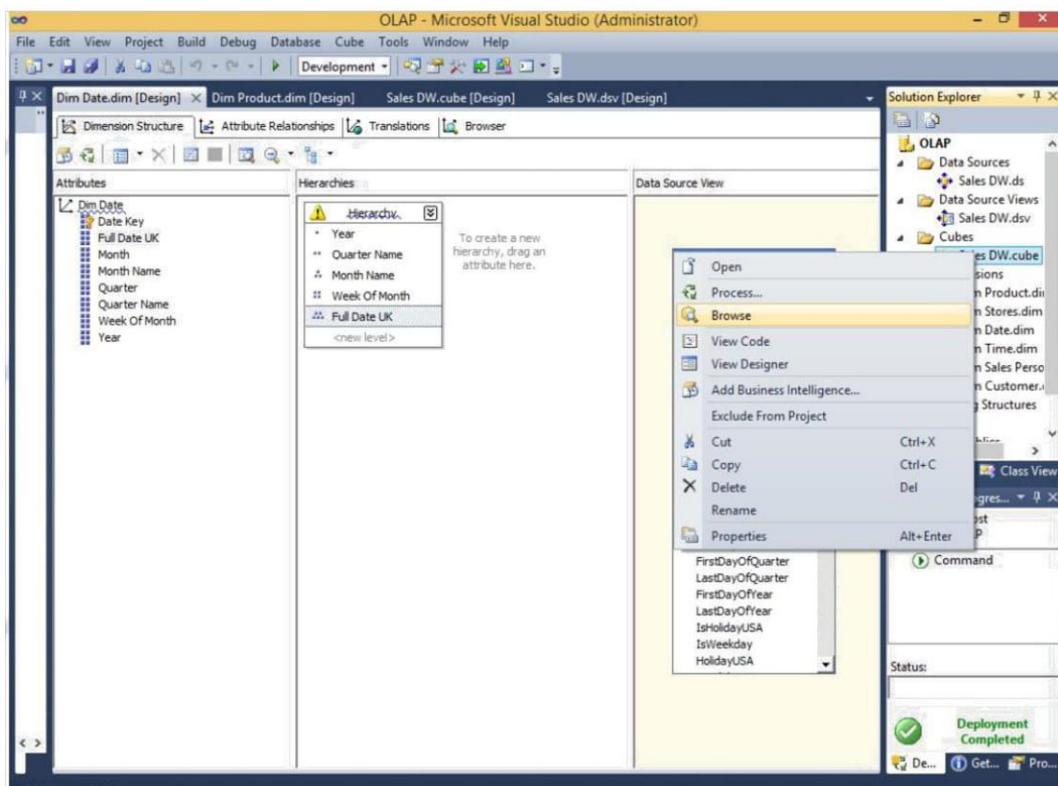


To process cube right click on Sales_DW.cube → Process





Browse the cube for analysis in solution explorer



PRACTICAL 4

Execute the MDX queries to extract the data from the datawarehouse.

Step 1: Open SQL Server Management Studio and connect to Analysis Services.

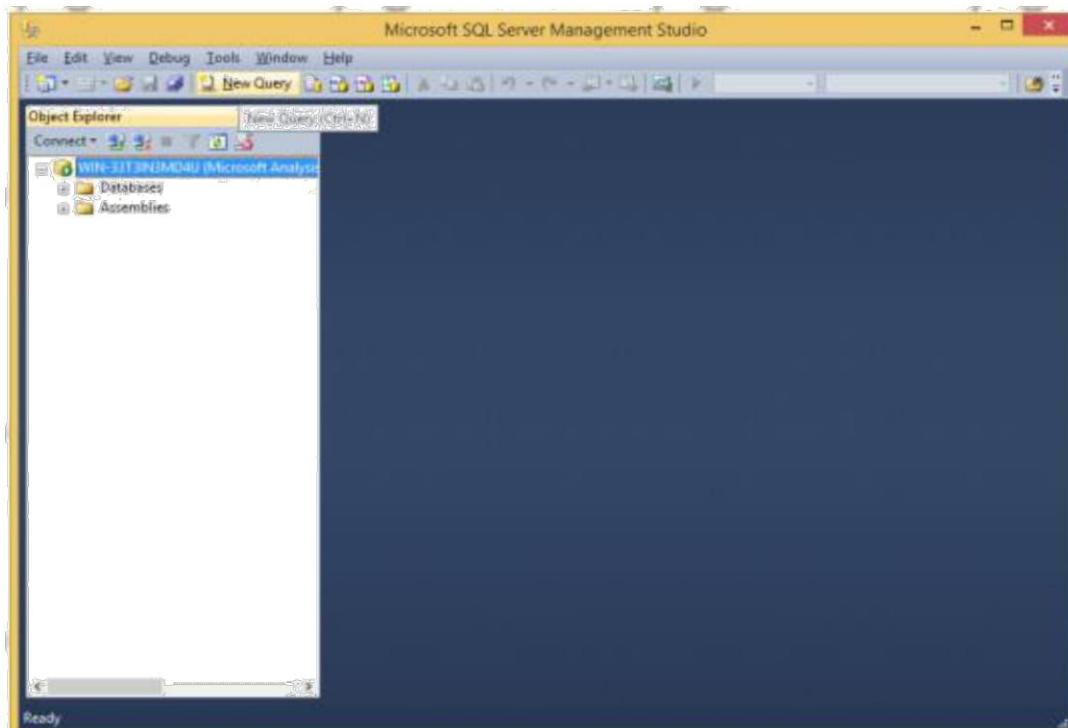
Server type: Analysis Services

Server Name: (according to base machine)

Click on connect

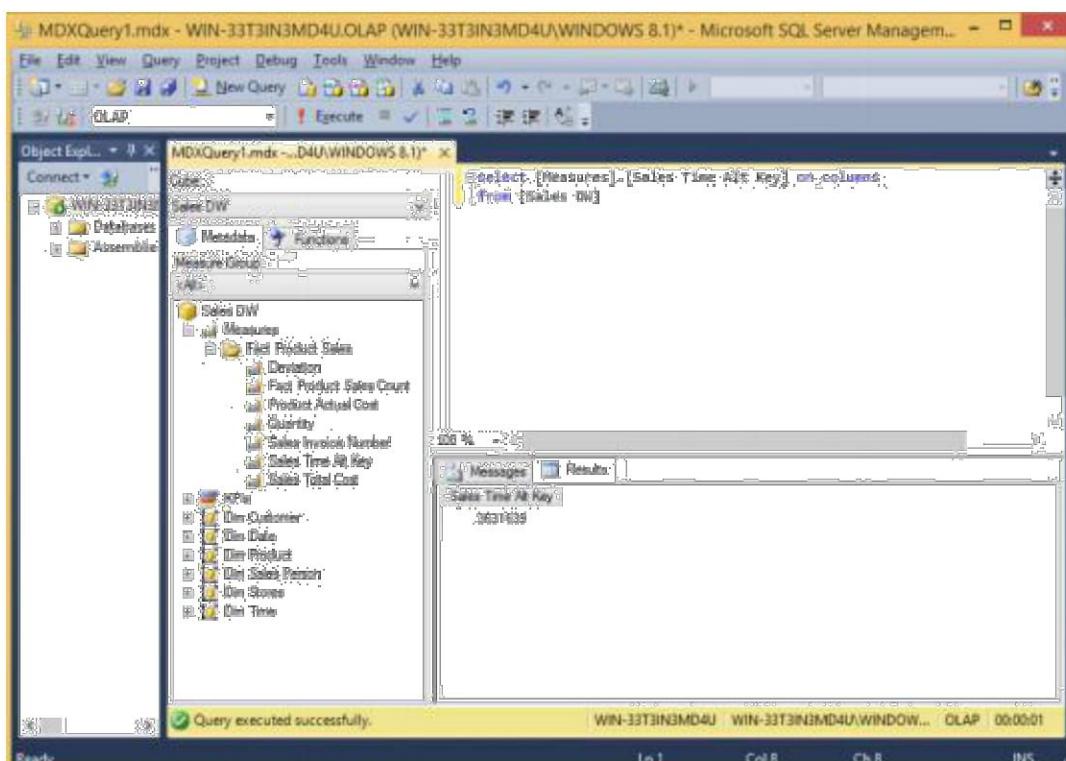


Step 2: Click on New Query & type following query based on Sales_DW



select [Measures].[Sales Time Alt Key] on columns
from [Sales DW]

Click on execute



select [Measures].[Quantity] on columns
from [Sales DW]

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "MDXQuery1.mdx - WIN-33T3IN3MD4U.OLAP (WIN-33T3IN3MD4U\WINDOWS 8.1) - Microsoft SQL Server Management..." The main window displays an MDX query:

```
select {[Measures].[Quantity]} on columns  
from [Sales DW]
```

The Object Explorer on the left shows a connection to "WIN-33T3IN3MD4U" containing "Databases" and "Assemblies". The "Sales DW" cube is selected, showing its "Metadata" and "Functions". The "Measure Group" node is expanded, revealing the "Sales DW" cube itself, which contains a "Measures" group. Under "Measures", there are several items including "Fact Product Sales", "Deviation", "Fact Product Sales Count", "Product Actual Cost", "Quantity", "Sales Invoice Number", "Sales Time All Key", and "Sales Total Cost". Below the cube, dimension nodes like "KPIs", "Dim Customer", "Dim Date", "Dim Product", "Dim Sales Person", "Dim Stores", and "Dim Time" are listed.

select [Measures].[Sales Invoice Number] on columns
from [Sales DW]

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "MDXQuery1.mdx - WIN-33T3IN3MD4U.OLAP (WIN-33T3IN3MD4U\WINDOWS 8.1) - Microsoft SQL Server Management..." The main window displays an MDX query:

```
select {[Measures].[Sales Invoice Number]} on columns  
from [Sales DW]
```

The Object Explorer on the left shows a connection to "WIN-33T3IN3MD4U" containing "Databases" and "Assemblies". The "Sales DW" cube is selected, showing its "Metadata" and "Functions". The "Measure Group" node is expanded, revealing the "Sales DW" cube itself, which contains a "Measures" group. Under "Measures", there are several items including "Fact Product Sales", "Deviation", "Fact Product Sales Count", "Product Actual Cost", "Quantity", "Sales Invoice Number", "Sales Time All Key", and "Sales Total Cost". Below the cube, dimension nodes like "KPIs", "Dim Customer", "Dim Date", "Dim Product", "Dim Sales Person", "Dim Stores", and "Dim Time" are listed.

select [Measures].[Sales Total Cost] on columns
from [Sales DW]

The screenshot shows the Microsoft SQL Server Management Studio interface with an MDX query window titled "MDXQuery1.mdx - WIN-33T3IN3MD4U.OLAP (WIN-33T3IN3MD4U\WINDOWS 8.1) - Microsoft SQL Server Management...".

The query pane contains the following MDX code:

```
select [Measures].[Sales Total Cost] on columns  
from [Sales DW]
```

The results pane displays the output of the query:

Sales Total Cost
1201.5

The status bar at the bottom indicates "Query executed successfully." and shows the session details: WIN-33T3IN3MD4U | WIN-33T3IN3MD4U\WINDOW... | OLAP | 00:00:01.

```
select [Measures].[Sales Total Cost] on columns  
, [Dim Date].[Year].[Year] on rows  
from [Sales DW]
```

The screenshot shows the Microsoft SQL Server Management Studio interface with an MDX query window titled "MDXQuery1.mdx - WIN-33T3IN3MD4U.OLAP (WIN-33T3IN3MD4U\WINDOWS 8.1) - Microsoft SQL Server Management...".

The query pane contains the following MDX code:

```
select [Measures].[Sales Total Cost] on columns  
, [Dim Date].[Year].[Year] on rows  
from [Sales DW]
```

The results pane displays the output of the query, showing three rows corresponding to the years 2003, 2004, and Unknown:

Sales Total Cost
1201.5
(null)
(null)

The status bar at the bottom indicates "Query executed successfully." and shows the session details: WIN-33T3IN3MD4U | WIN-33T3IN3MD4U\WINDOW... | OLAP | 00:00:01.

```
select [Measures].[Sales Total Cost] on columns  
, NONEMPTY({[Dim Date].[Year].[Year]}) on  
rows from [Sales DW]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "MDXQuery1.mdx - WIN-33T3IN3MD4U.OLAP (WIN-33T3IN3MD4U\WINDOWS 8.1) - Microsoft SQL Server Management..." The main window has two panes: the left pane is the Object Explorer showing a cube named "Sales DW" with various dimensions like "Dim Customer" and "Dim Date", and measures like "Sales Total Cost"; the right pane is the Results pane displaying the output of an MDX query. The query is:

```
select [Measures].[Sales Total Cost] on columns  
from [Sales DW]  
where [Dim Date].[Year].&[2013]
```

The results show a single row with "Sales Total Cost" in the column header and "2013" in the data cell, with a value of 1231.5.

select [Measures].[Sales Total Cost] on columns

from [Sales DW]

Where [Dim Date].[Year].&[2013]

This screenshot shows the same Microsoft SQL Server Management Studio environment. The title bar is identical. The Object Explorer on the left shows the "Sales DW" cube with its dimensions and measures. The Results pane on the right displays the same MDX query and its result:

```
select [Measures].[Sales Total Cost] on columns  
from [Sales DW]  
where [Dim Date].[Year].&[2013]
```

The results pane shows a single row with "Sales Total Cost" in the column header and "2013" in the data cell, with a value of 1231.5.

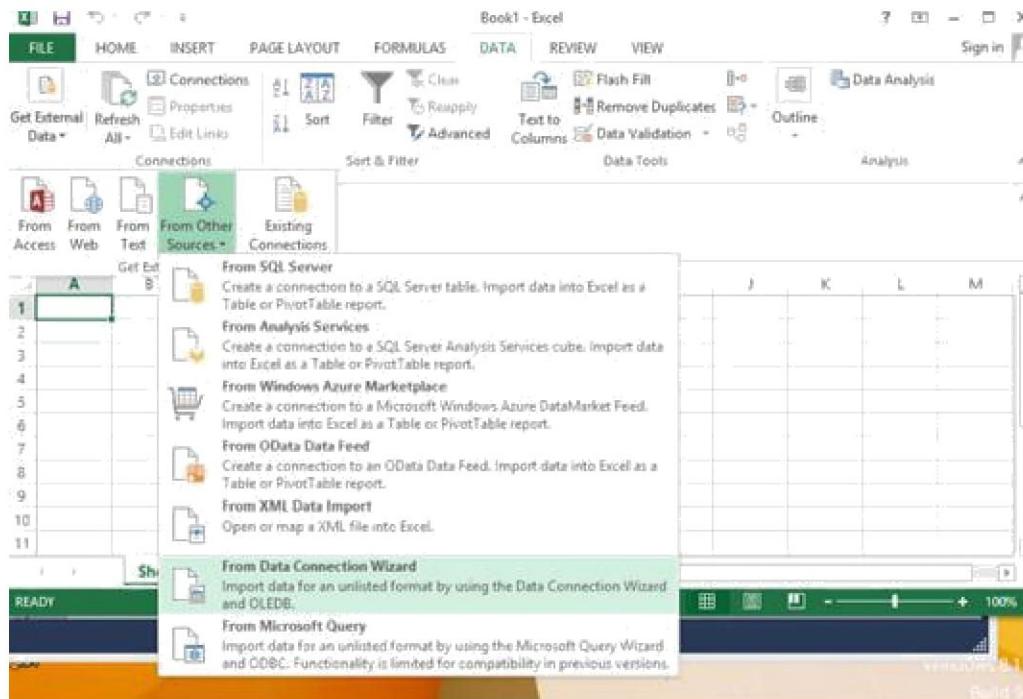
PRACTICAL 5 A

Import the datawarehouse data in Microsoft Excel and create the Pivot table and Pivot Chart

(Ms Office Professional is used to make sure Power View is enabled for visualization.)

Step 1: Open Excel 2013 (Professional)

Go to Data tab → Get External Data → From Other Sources → From Data Connection Wizard



Step 2: In Data Connection Wizard → Select Microsoft SQL Server → Click on Next



Step 3: In connect to Database Server provide Server name(Microsoft SQL Server

Name) Use Windows Authentication

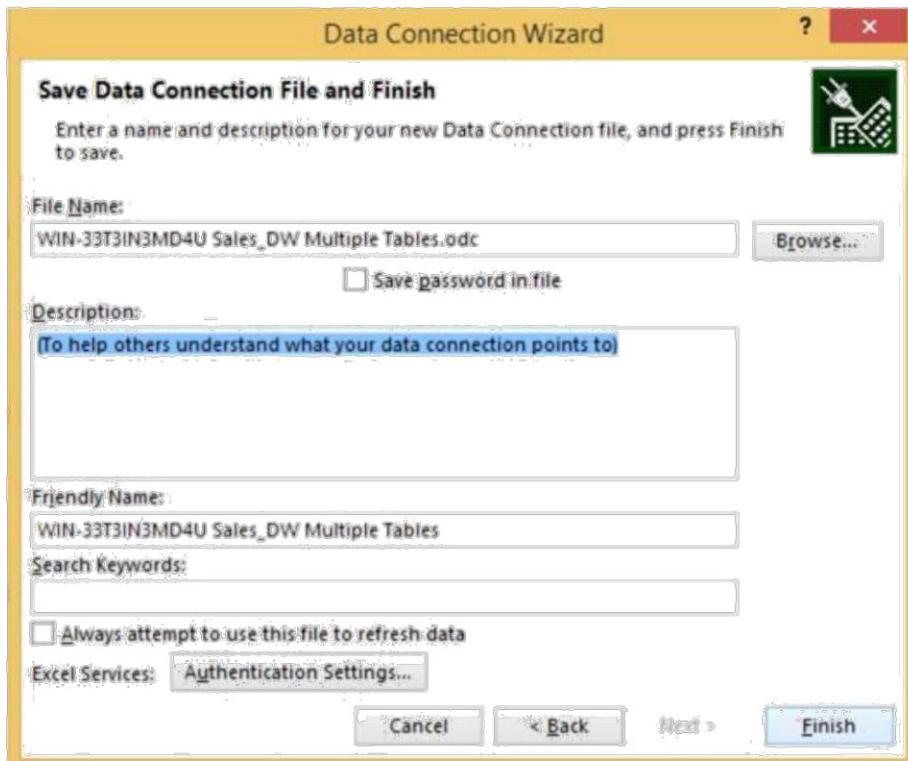
Click on Next



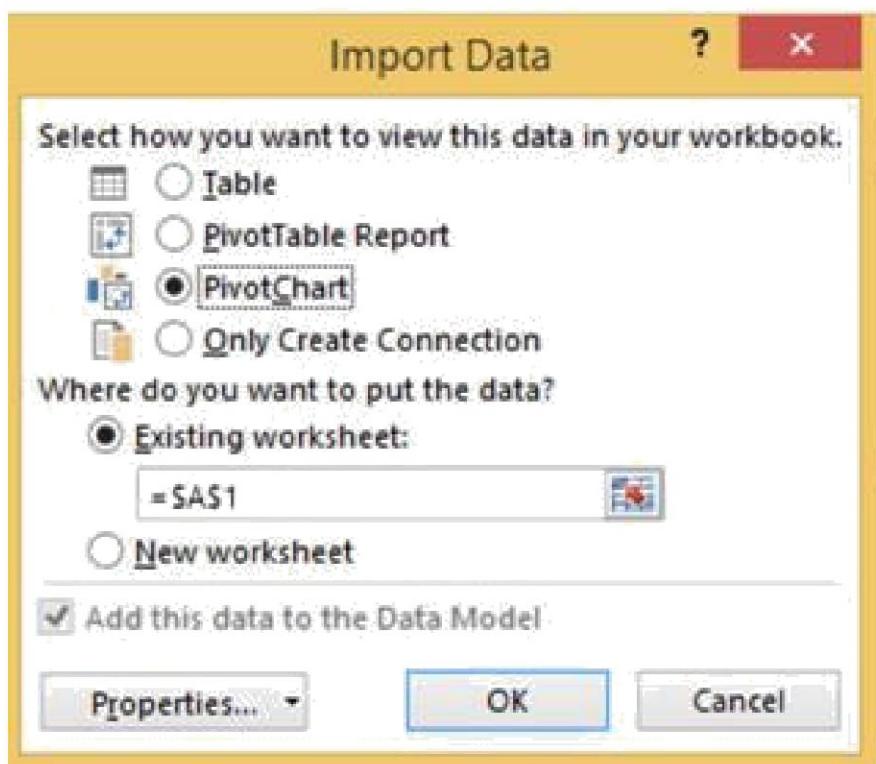
Step 4: In Select Database and Table → Select Sales_DW (already created in SQL) → check all dimensions and import relationships between selected tables



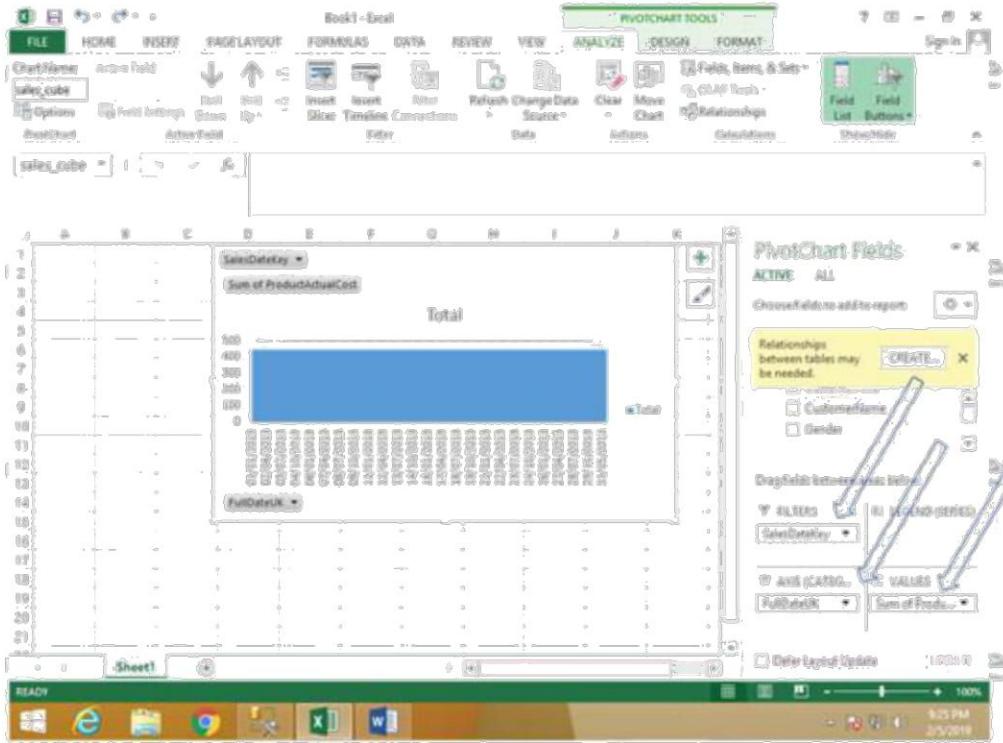
Step 5: In save data connection files browse path and click on Finish



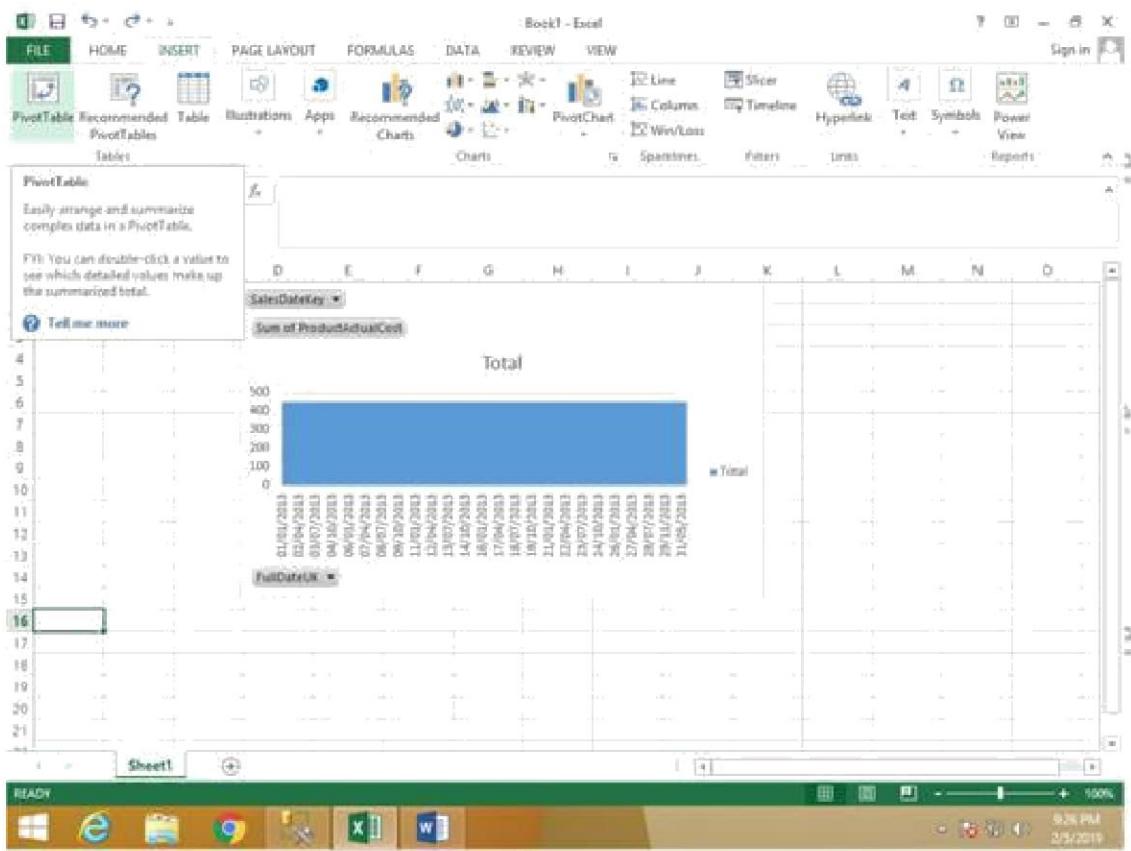
Step 6: In import data select Pivot Chart and click on OK



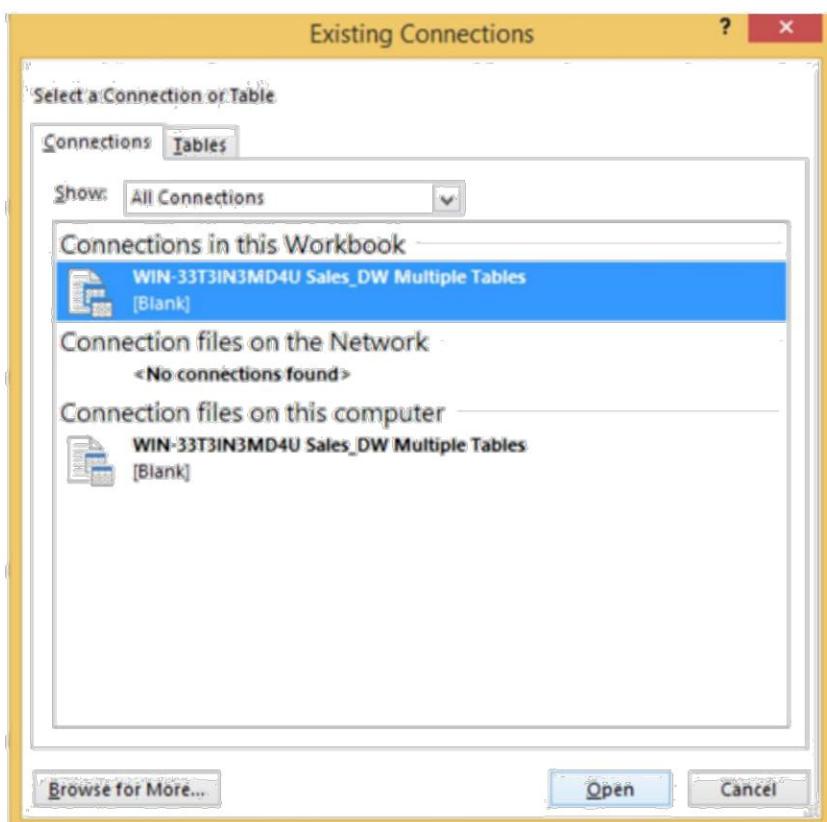
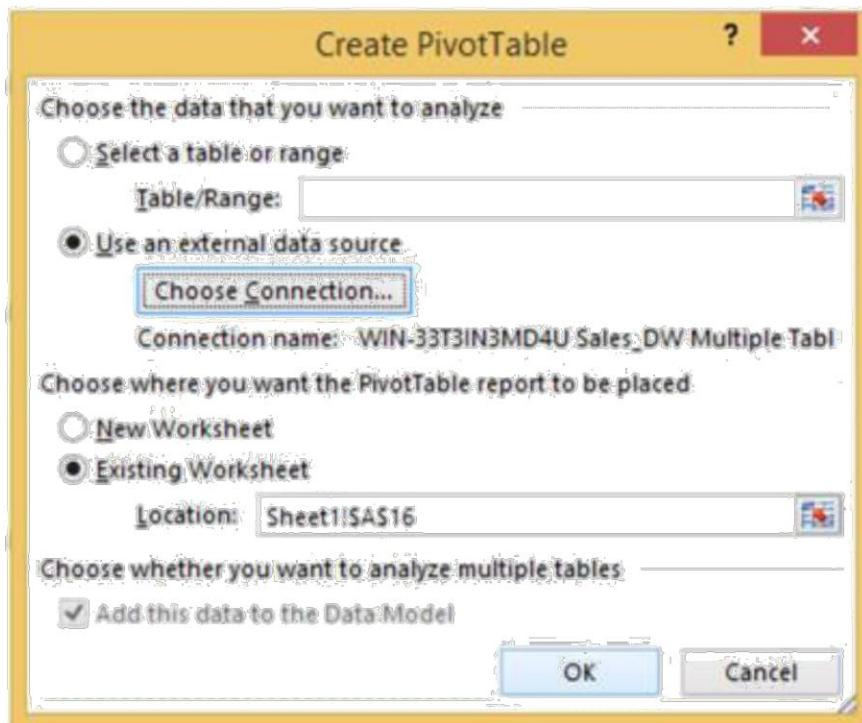
Step 7: In fields put SalesDateKey in filters, FullDateUK in axis and Sum of ProductActualCost in values



Step 8: In Insert Tab → go to Pivot Table



Step 9: Click on Choose Connection to select existing connection with Sales_DW and click on open



Pivot table and Pivot chart is created

Book1 - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW ANALYZE DESIGN

PivotTable Fields

ACTIVE: ALL

Choose fields to add to report

Relationships between tables may be needed.

CREATE...

Chart Area

ProductKey
ProductSubKey
ProductName

Drag fields between areas below:

FILTERS

COLUMNS

ROWS

VALUES

CustomerID
Gender
Sum of Sales
Sum of ProductKey

Defer Layout Update

UPDATE

9:28 PM
2/5/2019

100%

Row Labels

A16

FullDateUK

Sum of SalesPersonID Sum of ProductKey

	Sum of SalesPersonID	Sum of ProductKey
Bill Gates	78	55
M	78	55
Emma Watson	78	55
F	78	55
Henry Ford	78	55
M	78	55
Muskan Shaikh	78	55
F	78	55
Richard Thribin	78	55
M	78	55
Grand Total	78	55

Sheet1

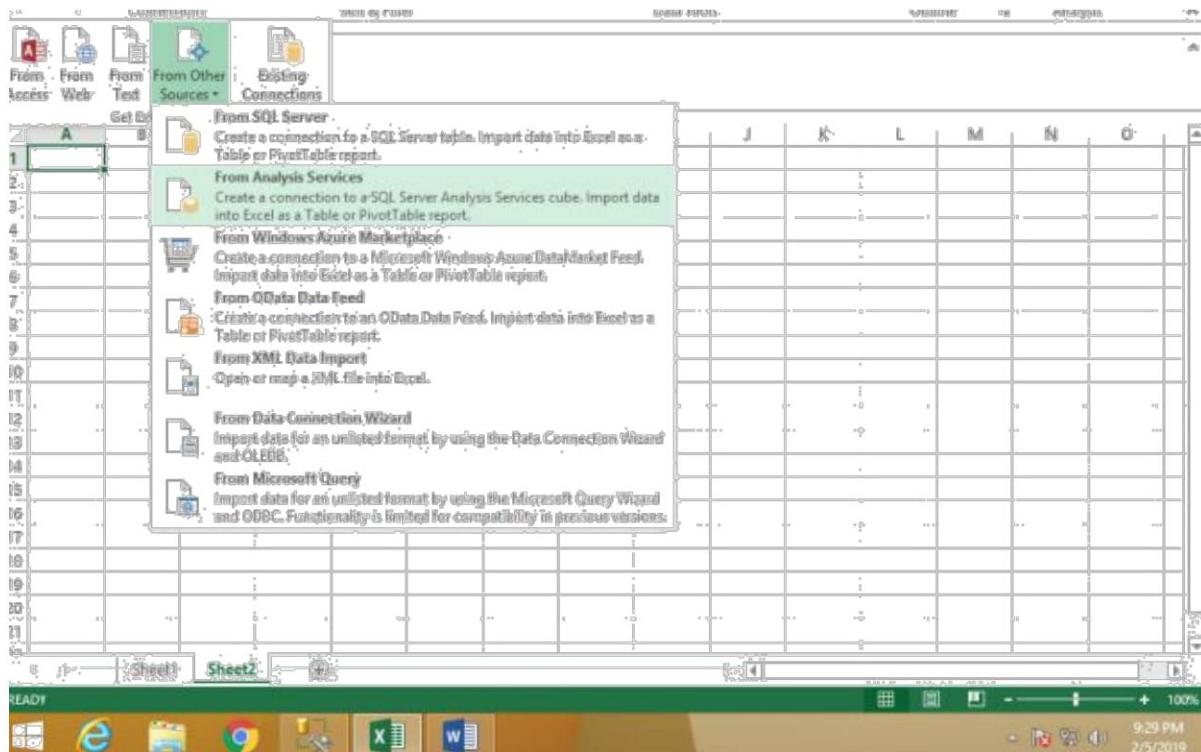
READY

This screenshot shows a Microsoft Excel spreadsheet titled 'Book1 - Excel'. The PivotTable Fields ribbon tab is selected. A 'PivotTable Fields' dialog box is open, showing the active fields: 'CustomerName', 'SalesPersonID', and 'ProductKey'. The 'Rows' section contains 'CustomerID' and 'Gender'. The 'Values' section contains 'Sum of Sales' and 'Sum of ProductKey'. The PivotTable itself displays data for various customers, grouped by gender, with a total row at the bottom. The PivotTable Fields ribbon tab includes options for 'Fields, Items, & Sets', 'OLAP Tools', 'Relationships', 'Calculated Fields', 'Calculated Items', and 'Tools'. The status bar at the bottom right shows the time as 9:28 PM and the date as 2/5/2019.

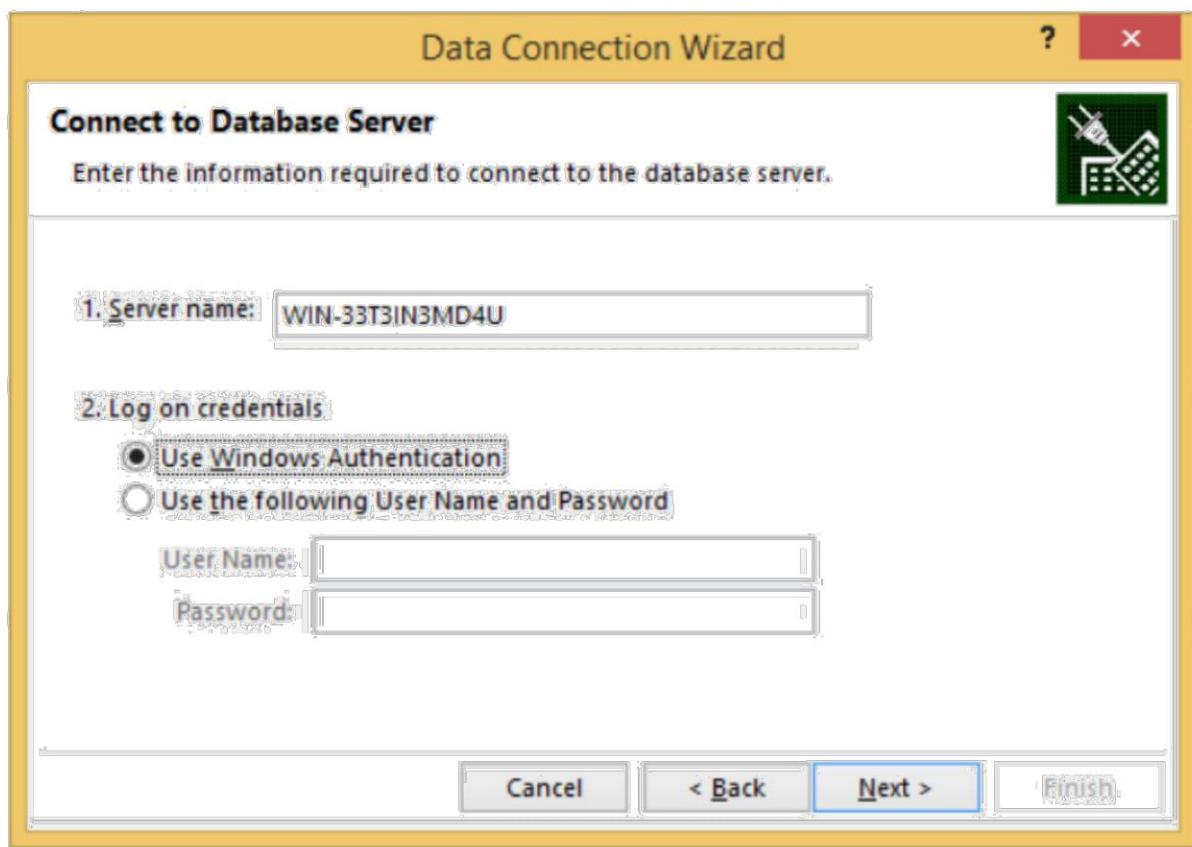
PRACTICAL 5 B

Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis.

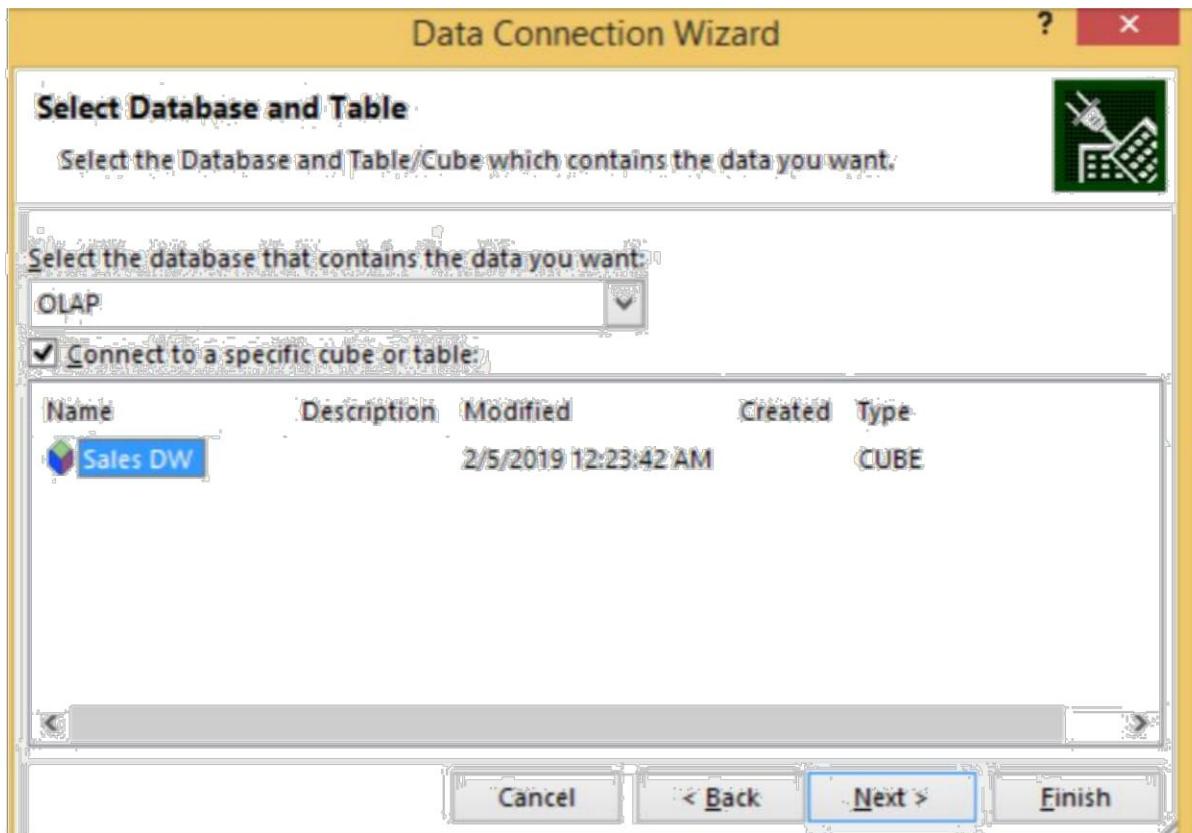
Step 1: Open Excel 2013 (Professional) Go to Data tab → Get External Data → From Other Sources → From Analysis Services



Step 2: Select Server name and Windows Authentication and click on Next



Step 3: Select OLAP(as per created before) click on Next



Step 4: Browse and select path name and click on Finish

Data Connection Wizard

?

X

Save Data Connection File and Finish

Enter a name and description for your new Data Connection file, and press Finish to save.



File Name:

WIN-33T3IN3MD4U OLAP Sales DW.odc

[Browse...](#)

Save password in file

Description:

(To help others understand what your data connection points to)

Friendly Name:

WIN-33T3IN3MD4U OLAP Sales DW

Search Keywords:

Always attempt to use this file to refresh data

Excel Services: [Authentication Settings...](#)

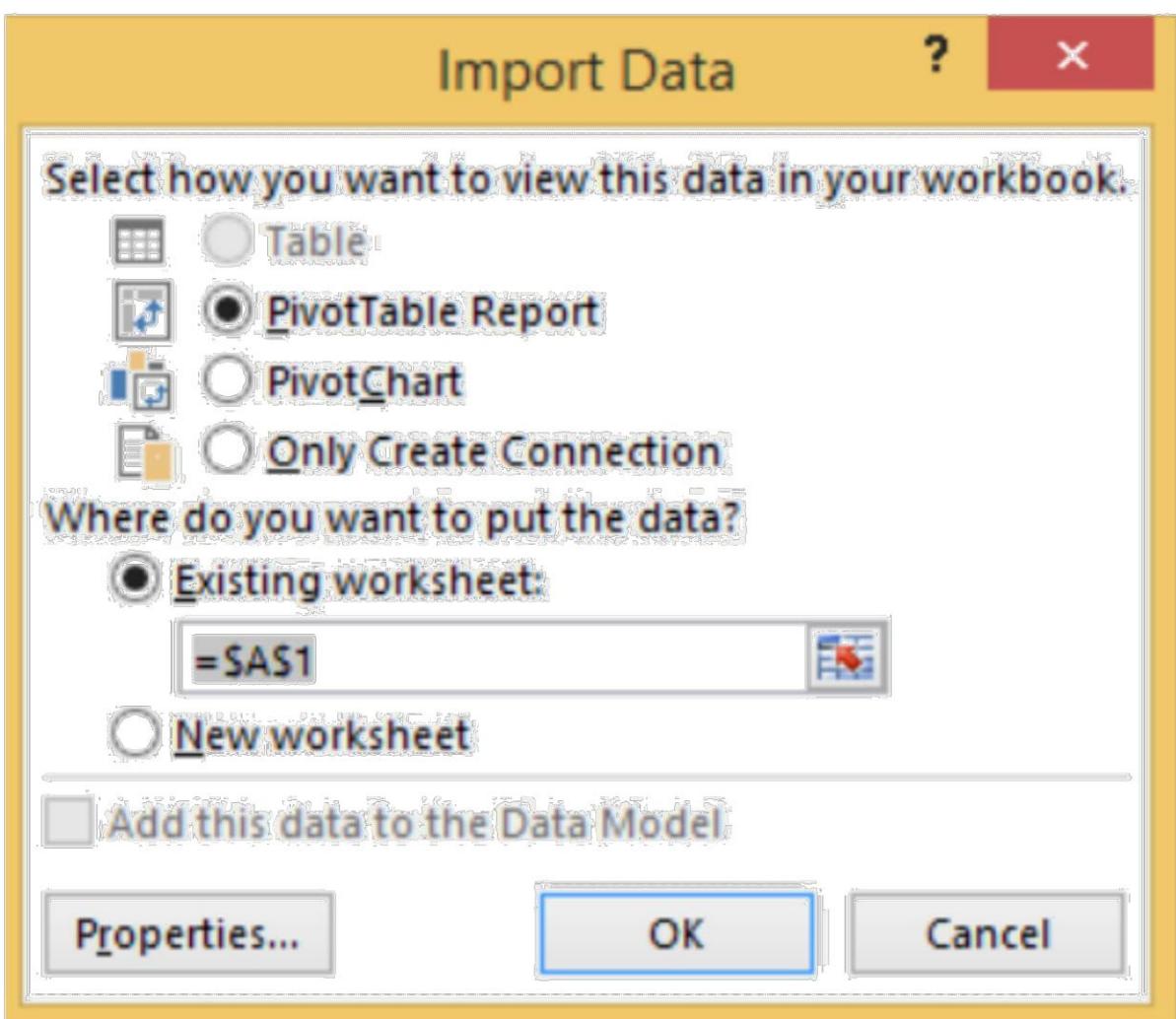
[Cancel](#)

[Back](#)

[Next](#)

[Finish](#)

Step 5: Select PivotTableReport → OK



Step 6 : Drag and Drop Fields in rows column and values

Step 7: Go to Insert tab → pivot chart and select Pivot Chart from drop down

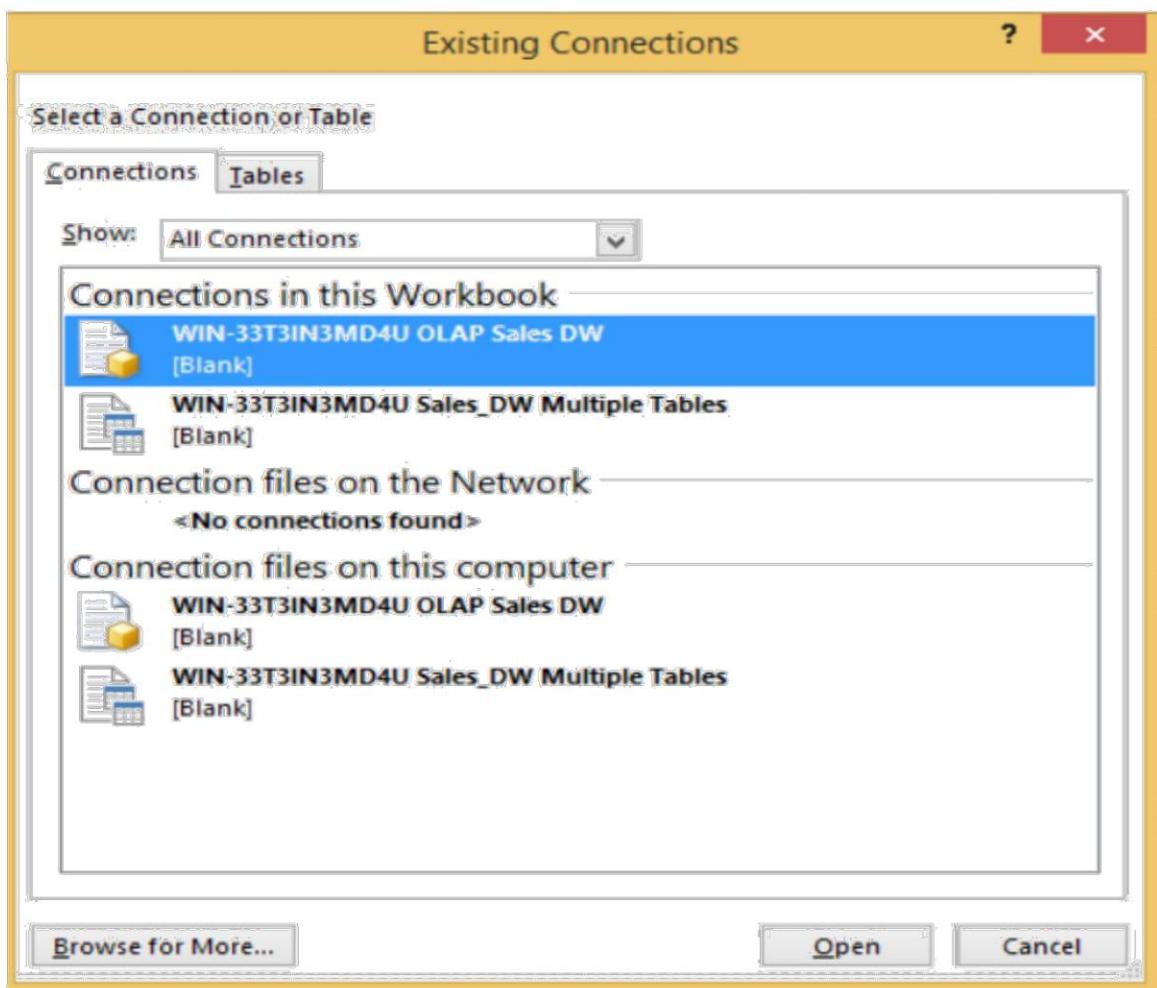
Screenshot of Microsoft Excel showing a PivotTable and PivotChart interface.

The PivotTable data is as follows:

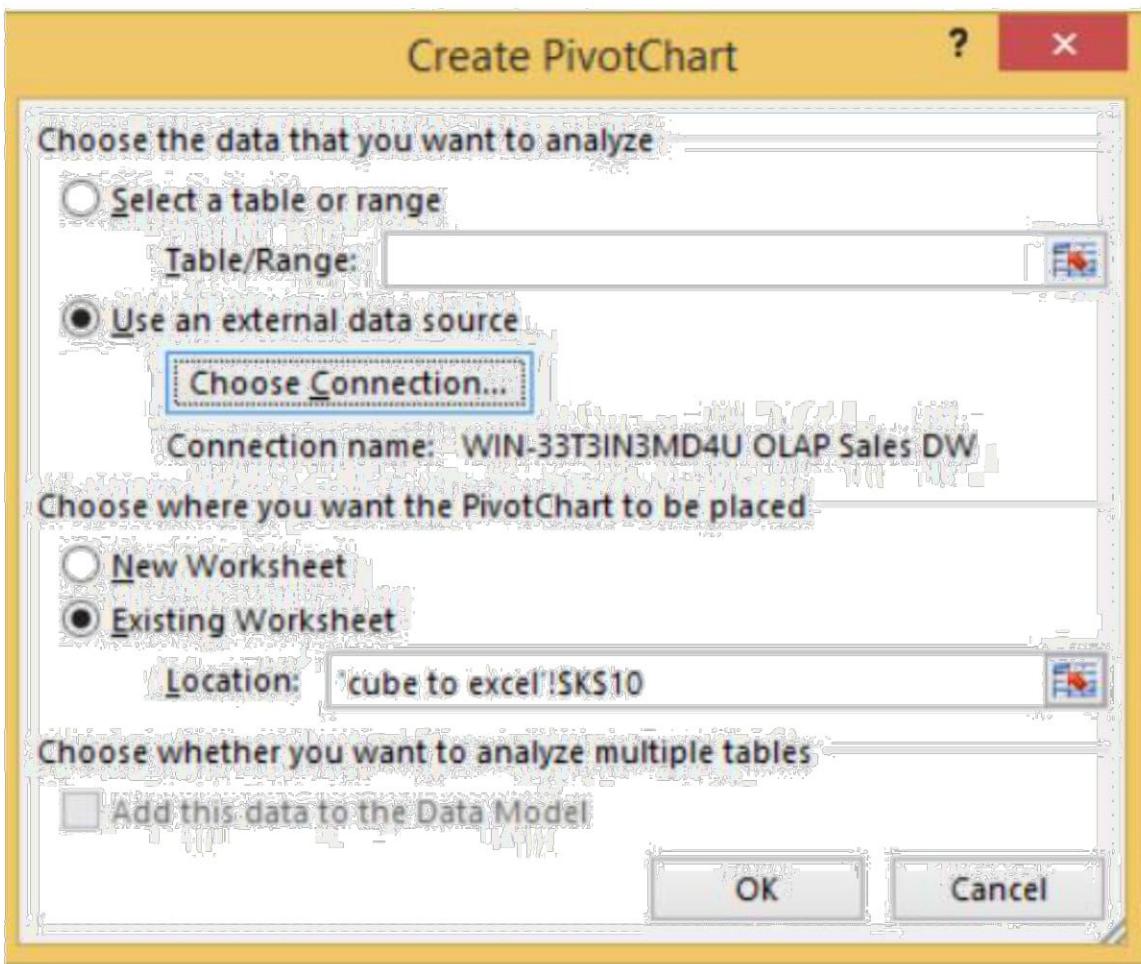
	A	B	C	D	E	F	G	Grand Total
Row Labels		44347	44519	52415	59326	59349	67390	74877
1	Arial Washing Powder 1kg	11		5		2		18
	Nirma Soap		3		3			6
	Rice Grains 1kg		2			1		3
	SunFlower Oil 1ltr		1		2		1	4
	Wheat Floor 1kg		4					4
2			8		2			10
10	Nirma Soap		6					6
	Rice Grains 1kg				1			1
	SunFlower Oil 1ltr		2					2
	Wheat Floor 1kg				1			1
14			10		5			15
15	Arial Washing Powder 1kg		2					2
16	Nirma Soap		3		3			6
17	Rice Grains 1kg		4					4
18	SunFlower Oil 1ltr		1					1
19	Wheat Floor 1kg				2			2
20	Grand Total		11	8	10	5	2	43

The PivotChart dropdown menu is open, showing options like PivotChart, PivotChart & PivotTable, PivotTable, Line, Column, Timeline, Win/Loss, Slicer, Hyperlink, Text, Symbols, Power View, and Reports.

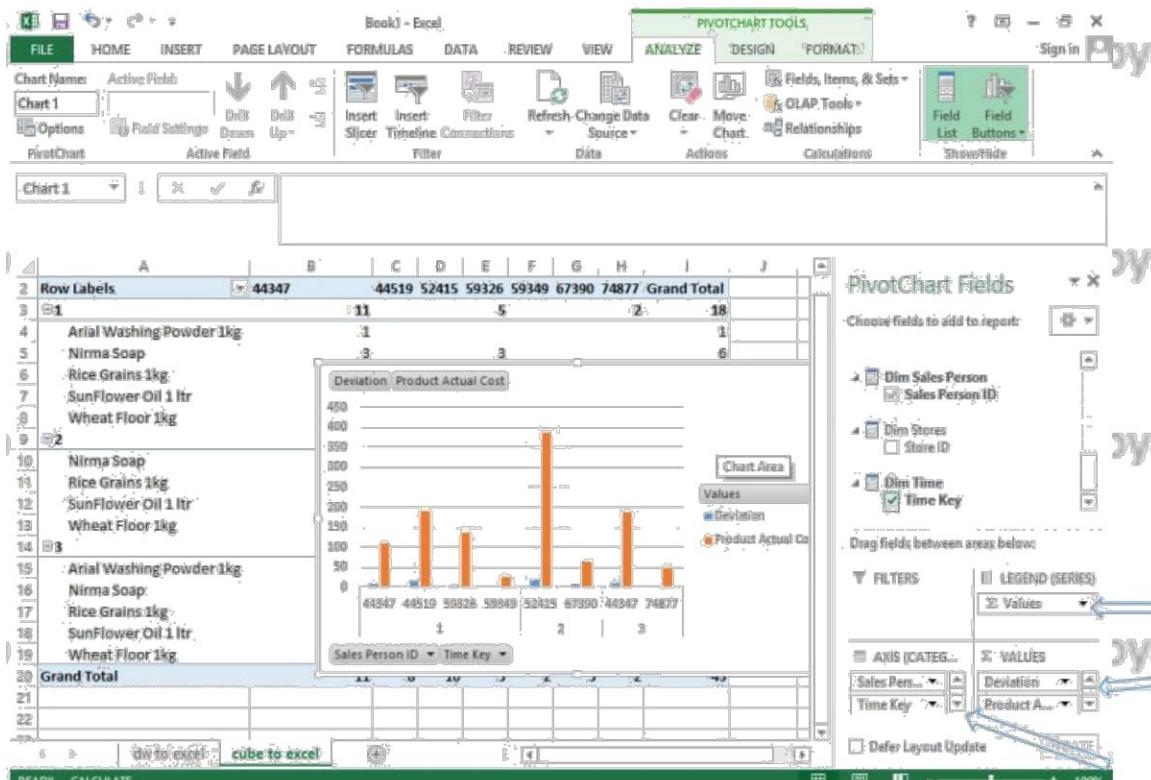
Step 8: Select existing connection OLAP Sales DW and click on Open



Step 9: Click on Choose connection to select path



Step 10: Click on OK



PRACTICAL-6

AIM: IMPLEMENTATION OF CLASSIFICATION ALGORITHM USING R TOOL

PROCEDURE:

Step 1: get the data points for 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)

Step 2:Convert to time series object

- rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)

Step 3:Print the data

- print(rainfall.timeseries)

Step 4:Give the file name

- png(file = "rainfall.png")

Step 5:Plot the graph

- plot(rainfall.timeseries)

Step 6 :Save the file

- dev.off()

Step 7 : plot again if it isn't visible

- plot(rainfall.timeseries)

The screenshot shows an R console window with the following text:

```
R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

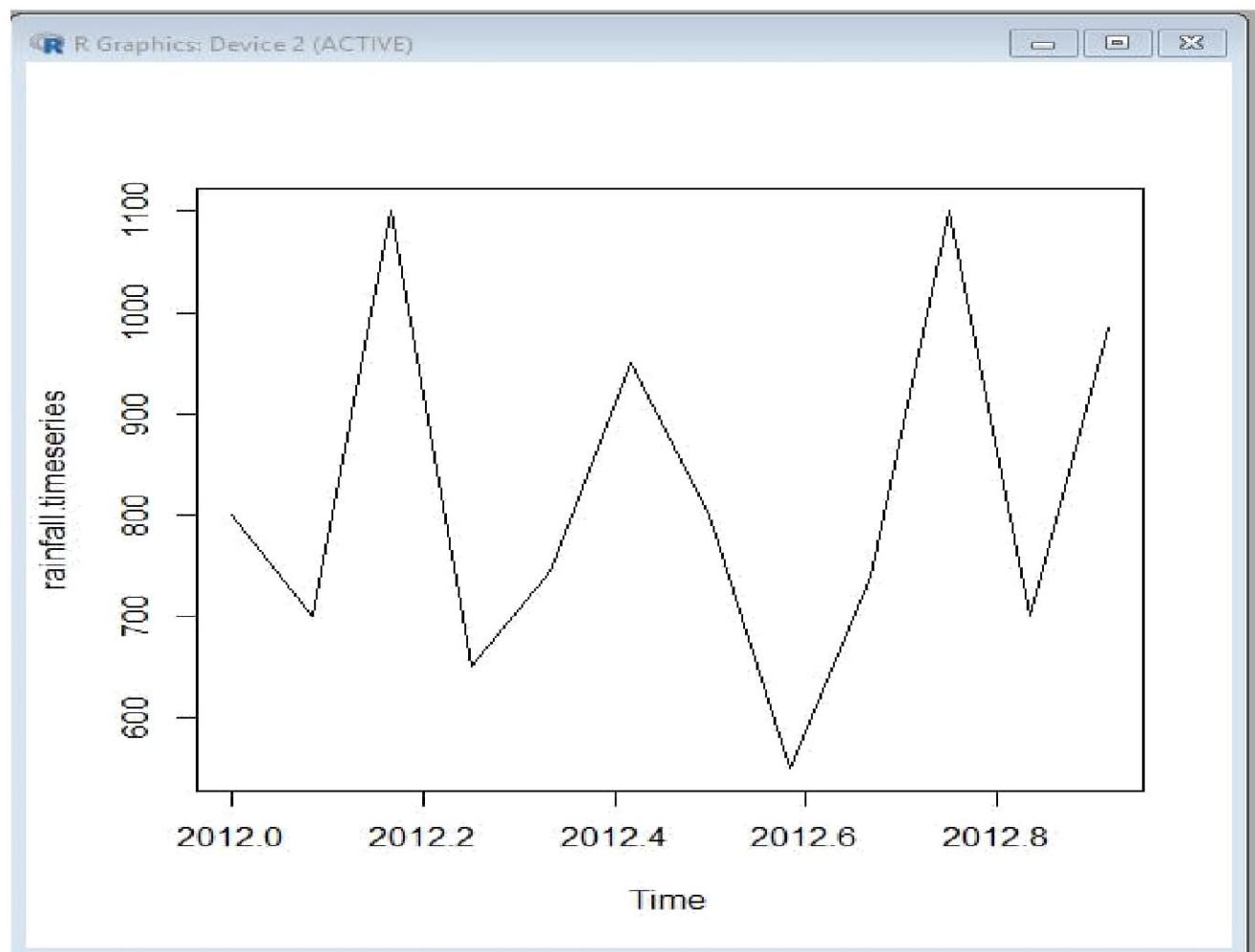
Natural language support but running in an English locale.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> rainfall <- c(800.44, 700, 1100.4, 650.6, 744.6, 950.55, 800.4, 650.44, 700.4, 1100.8, 700, 85
> rainfall.timeseries <- ts(rainfall,start=c(2012,1),frequency=12)
> print(rainfall.timeseries)
   Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep
2012 800.44 700.00 1100.40 650.60 744.60 950.55 800.40 550.44 700.40
          Oct    Nov    Dec
2012 1100.50 700.50 985.00
> plot(rainfall.timeseries)
>
```

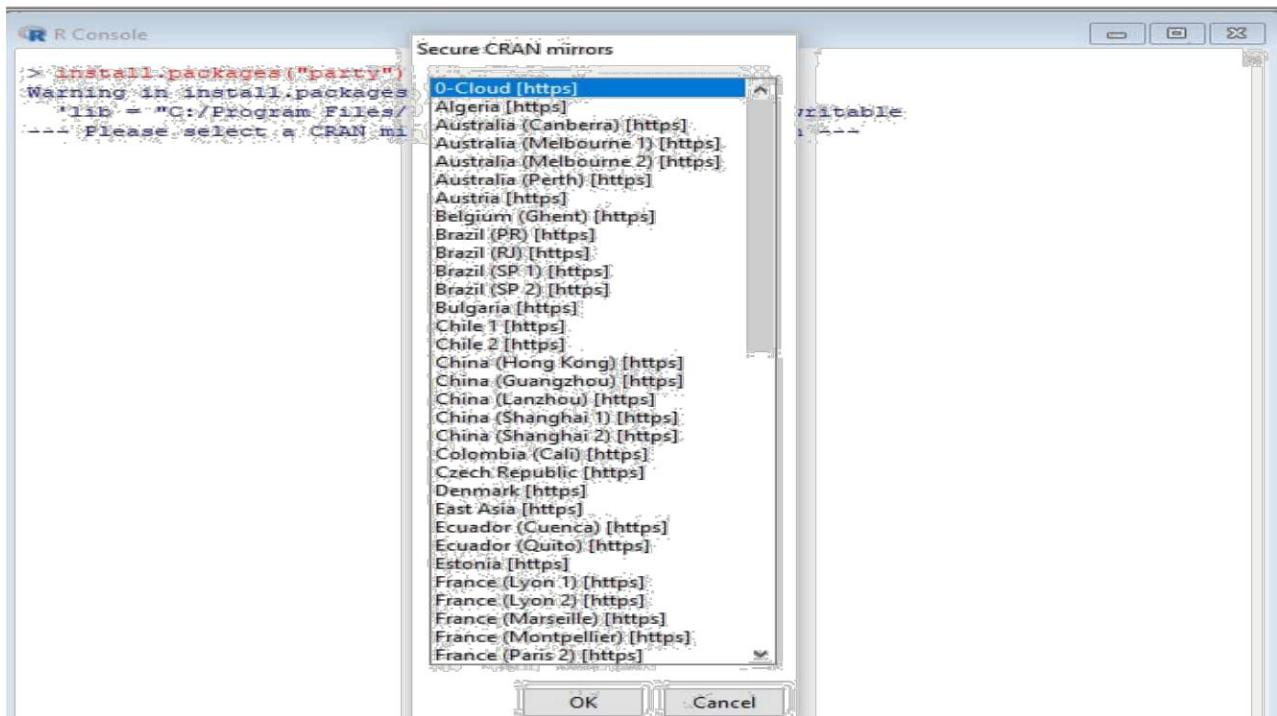
Output:



PRACTICAL-7

AIM: IMPLEMENTATION DECISION TREE USING R TOOL

PROCEDURE:



Step 1: Install the appropriate packages

- `install.packages("party")`

The downloaded binary packages are in

C:\Users\5itlab\AppData\Local\Temp\RtmpUNASzq\downloaded_packages

Step 2: Set the Repositories

- `setRepositories() -CRAN`

Step 3 :Choose the mirror

- `chooseCRANmirror()`

Step 4: Load the packages

- `library("party")`

Step 3:create an input data frame

- `input.dat <- readingSkills[c(1:105),]`

Step 4 :Give a file name

- `png(file="decision_tree.png")`

Step 5:Create the tree.

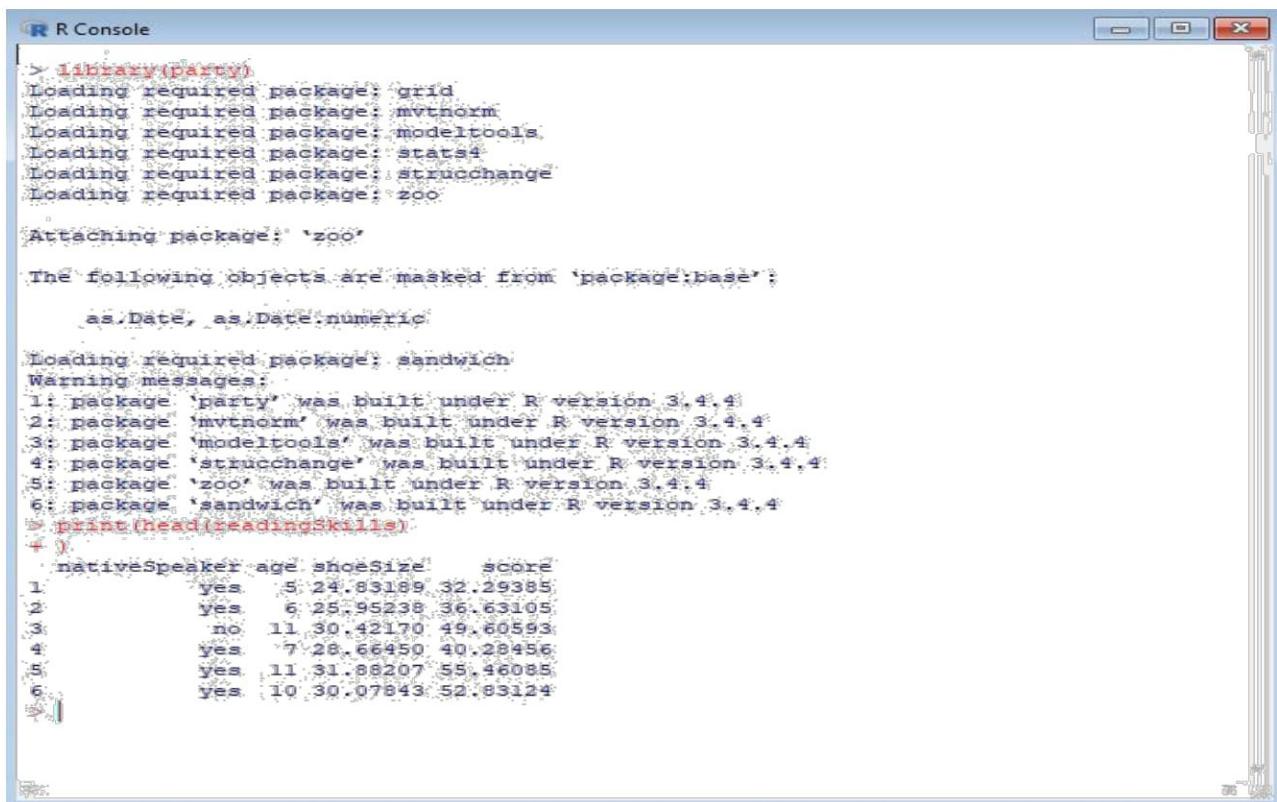
- output.tree<- ctree(nativeSpeaker~ age + shoeSize+ score,data=input.dat)

Step 6: Plot the tree

- plot(output.tree)

Step:7 Save the file :

- dev.off



R Console

```
> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

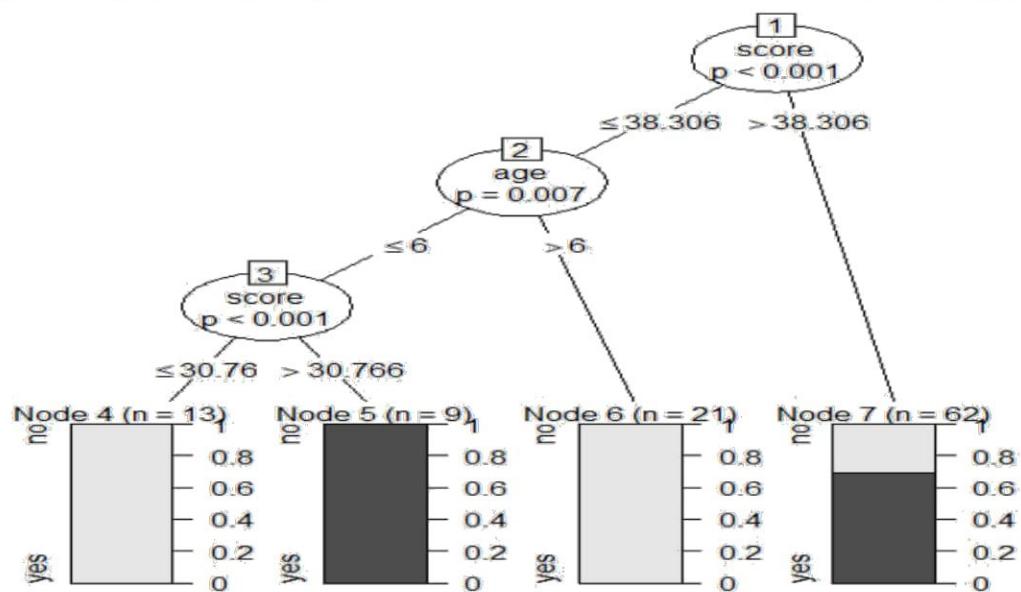
Attaching package: 'zoo'

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

  as.Date, as.Date.numeric

Loading required package: sandwich
Warning messages:
1: package 'party' was built under R version 3.4.4
2: package 'mvtnorm' was built under R version 3.4.4
3: package 'modeltools' was built under R version 3.4.4
4: package 'strucchange' was built under R version 3.4.4
5: package 'zoo' was built under R version 3.4.4
6: package 'sandwich' was built under R version 3.4.4
> print(head(readingSkills))
+ )
  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
```

OUTPUT



PRACTICAL NO-8

AIM :K-MEANSCLUSTERING USING R TOOL

PROCEDURE:

Step 1 : Extract the dataset of iris to perform the clustering.

- newiris <- iris

Step 2 : Assign the null value to the variable.

- newiris\$Species <- NULL

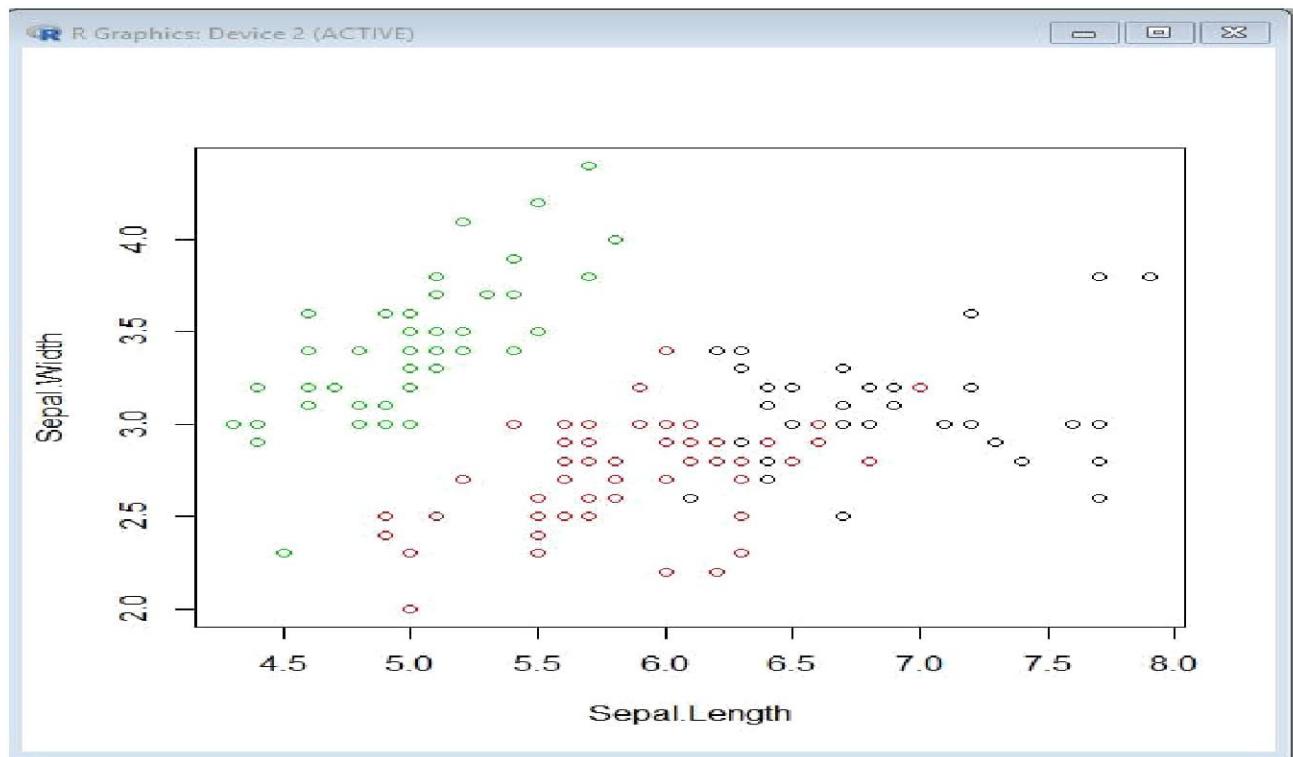
Step 3: perform the clustering and define the size for newiris cluster as 3

- (kc <- kmeans(newiris ,3))

Step 4 :plot the graph with following rows and columns

- `plot(newiris[c("Sepal.Length", "Sepal.Width")], col=kc$cluster)`

Output



PRACTICAL NO-9

AIM: PREDICTION USING LINEAR REGRESSION

A) Create relationship model

B) Get the summary of the relation

C) Predict the weight of new person

D) Visualize the relationship graphically

PROCEDURE:

Step 1 : Assign the variables to x and y variable

- `X <- c(151,174,138,186,128,136,179,162,151,132)`
- `Y <-c(63,81,56,91,47,57,76,72,62,48)`

Step 2 : Create linear regression variable

- `realtion <- lm(y~x)`

Step 3 : Print the variable

- `print(r)`

Call:

`lm(formula = y ~ x)`

Coefficients:

(Intercept)	x
27.3183	0.1144

Step 4 : Print the summary of the linear realtion variable.

- `print(summary(r))`

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

Residuals:

Min	1Q	Median	3Q	Max
-35.222	-9.336	-3.220	8.739	40.405

Coefficients:

Estimate	Std. Error	t value	Pr(> t)
(Intercept)	27.3183	51.9967	0.525
x	0.1144	0.3356	0.341

Residual standard error: 20.88 on 8 degrees of freedom

Multiple R-squared: 0.01432, Adjusted R-squared: -0.1089
F-statistic: 0.1162 on 1 and 8 DF, p-value: 0.742

Step 5 : create a data frame and find the y variable corresponding to x value.

- `a <- data.frame(x=179)`

Step 6 : predict the relation of the data frame and the linear relation variable

- result <- prediction(relation,a)

Step 7 : print the result

- print(result)

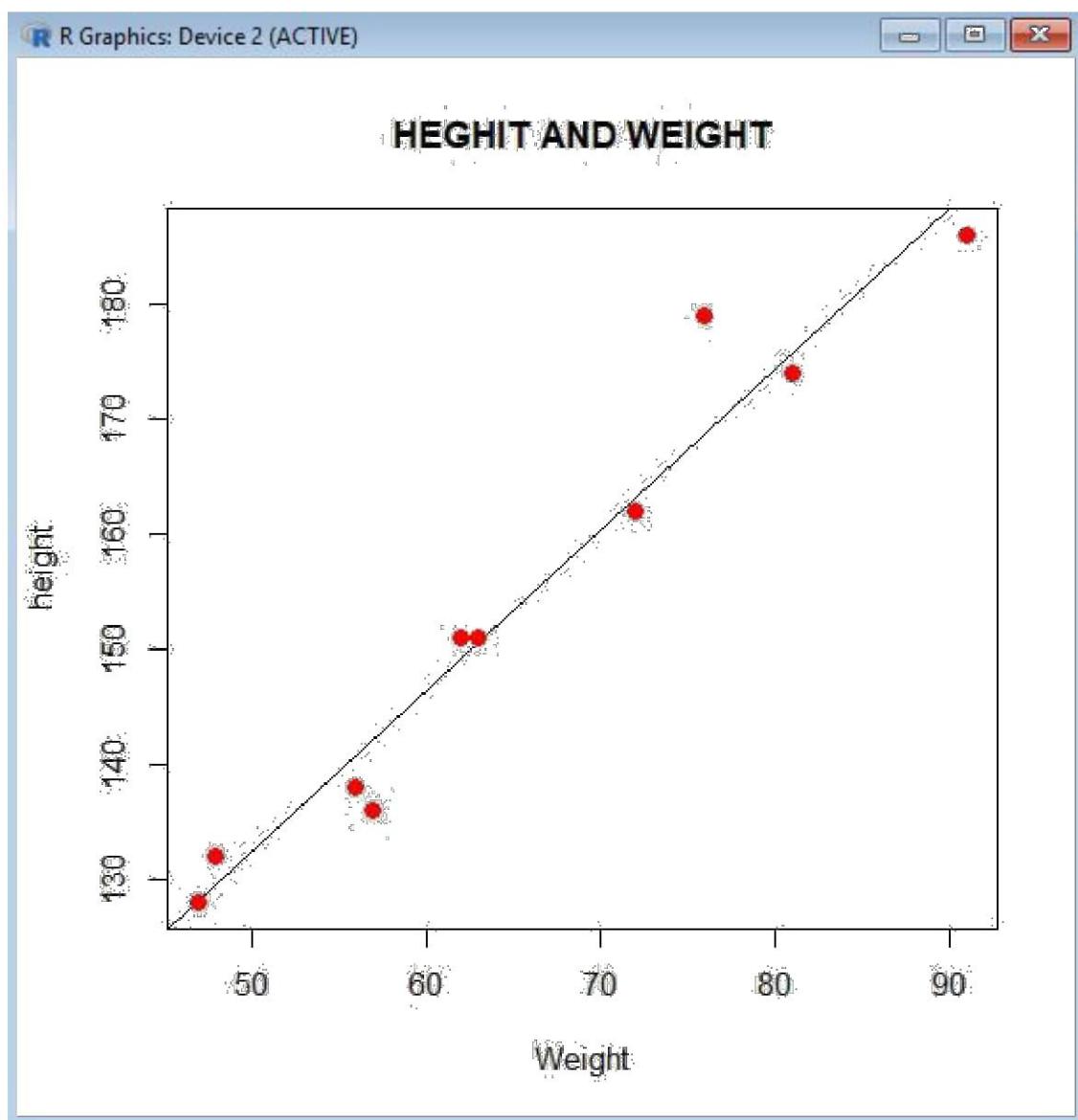
```
1  
46.76455
```

Step 8 : plot the graph for the following

- plot(y,x,col="red",main="HEIGHTANDWEIGHT",abline(lm(x~y)),cex=1.3,pch=16,xlab="Weight",ylab="height")

```
> x <- c(151,174,138,186,128,136,179,162,151,132)  
> y <- c(63,81,56,91,47,57,76,72,62,48)  
> relation <- lm(y~x)  
> print(summary(relation))  
  
Call:  
lm(formula = y ~ x)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-6.486 -1.310  0.298  1.774  3.759  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -39.1061    8.1822 -4.779  0.00139 **  
x             0.6793    0.0528 12.864 1.26e-06 ***  
---  
Signif. codes:  0 '*****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1  
  
Residual standard error: 3.285 on 8 degrees of freedom  
Multiple R-squared:  0.9539,   Adjusted R-squared:  0.9481  
F-statistic: 165.5 on 1 and 8 DF,  p-value: 1.26e-06  
  
> a <- data.frame(x=170)  
> result<- predict(relation,a)  
> print(result)  
1  
76.37234  
> plot(y,x,col="red",main="HEIGHT AND WEIGHT",abline(lm(x~y)),cex=1.3,pch=16,xlab="Weight",ylab="height")  
>
```

OUTPUT



PRACTICAL-10

Aim: DATA ANALYSIS USING TIME SERIES ANALYSIS

PROCEDURE:

Step 1: get the data points for 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)
```

Step 2:Convert to time series object

```
> rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
```

Step 3:Print the data

```
> print(rainfall.timeseries)
```

Step 4:Give the file name

```
> png(file = "rainfall.png")
```

Step 5:Plot the graph

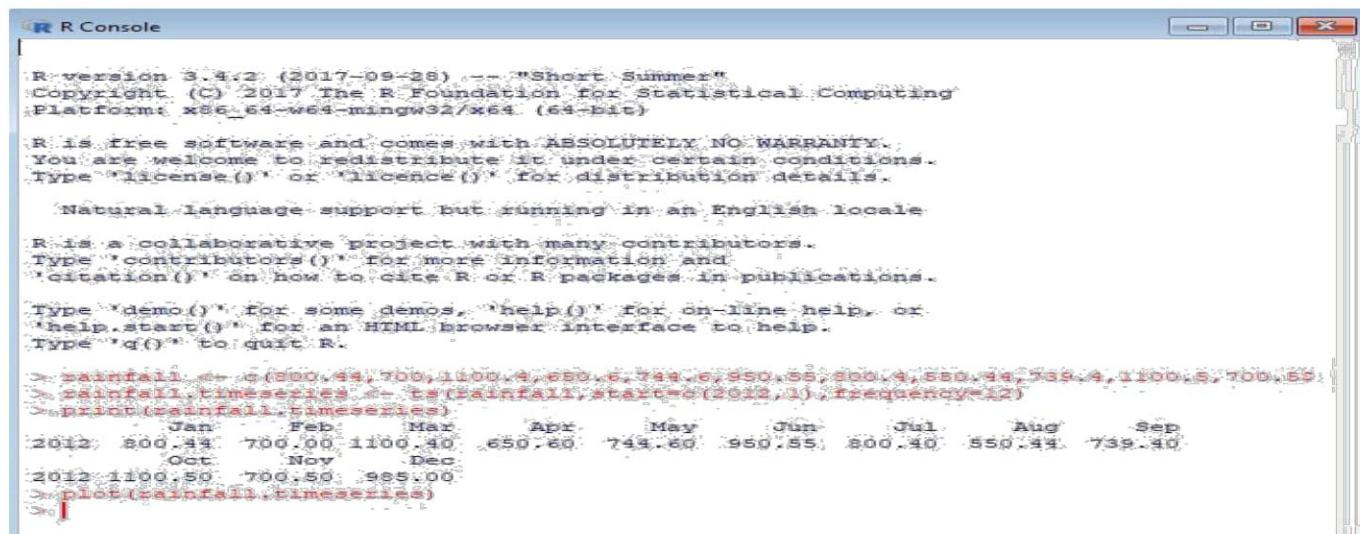
```
> plot(rainfall.timeseries)
```

Step 6 :Save the file

```
> dev.off()
```

Step 7 : plot again if it isn't visible

```
> plot(rainfall.timeseries)
```



The screenshot shows the R Console window with the following text:

```
R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> rainfall <- c(800,48,700,1100,4,680,6,744,6,950,58,800,4,680,44,739,4,1100,5,700,69,
> rainfall.timeseries <- ts(rainfall,start=c(2012,1),frequency=12)
> print(rainfall.timeseries)
Jan Feb Mar Apr May Jun Jul Aug Sep
2012 800.48 700.00 1100.40 680.60 744.60 950.58 800.44 680.44 739.40 1100.50 700.69
Oct Nov Dec
2012 1100.50 700.50 985.00
> plot(rainfall.timeseries)
>
```

Output:

