

INDEX

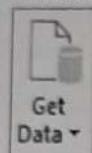
Practical No.	Practical Name	Page No.	Date	Signature
1.	Import the legacy data from different sources such as (Excel, SQLServer, Oracle etc.) and load in the target system. (You can download sample database such as AdventureWorks, Northwind, Foodmart etc.)	3	5/01/24	
2.	(A) Perform the Extraction Transformation and Loading (ETL) process to construct the database in Power BI (B) Perform the Extraction Transformation and Loading (ETL) process to construct the database in SQLServer	9 15	12/01/24	
3.	Create a cube with suitable dimensions and fact tables based on OLAP	23	19/01/24	
4.	Execute the MDX queries to extract the data from the data warehouse	31	02/02/24	<i>Sanjay</i>
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	33 35	09/02/24 16/02/24	
6.	Apply the What – if Analysis for data visualization. Design and generate necessary reports based on the datawarehouse data. (A) Using RStudio (B) Using Scenario Manager in Excel (C) Using Goal Seek Analysis in Excel	39 40 43	23/02/24 20/02/24 11/3/24	
7.	Perform the data classification using classification algorithm	47	15/3/24	
8.	Perform the data clustering using clustering algorithm	49	15/3/24	
9.	Perform the Linear Regression on the given data warehouse data	51	22/03/24	
10.	Perform Data Analysis using Timeseries.	53	22/3/24	

Practical No. 1

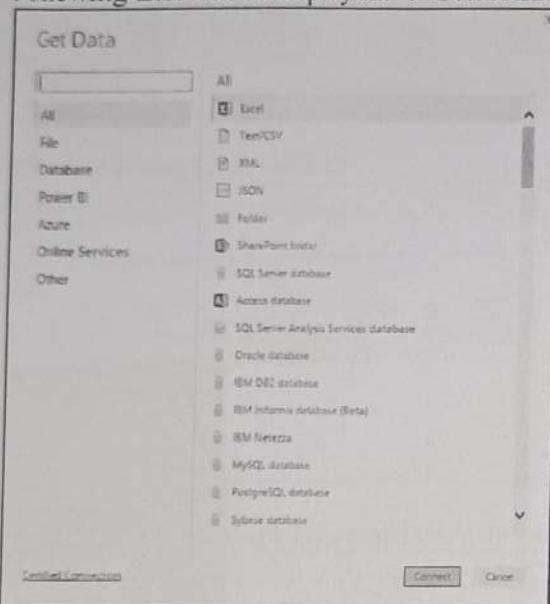
Import the legacy data from different sources such as (Excel, SQLServer, Oracle etc.) and load in the target system. (You can download sample database such as AdventureWorks, Northwind, Foodmart etc.)

Steps to follow :

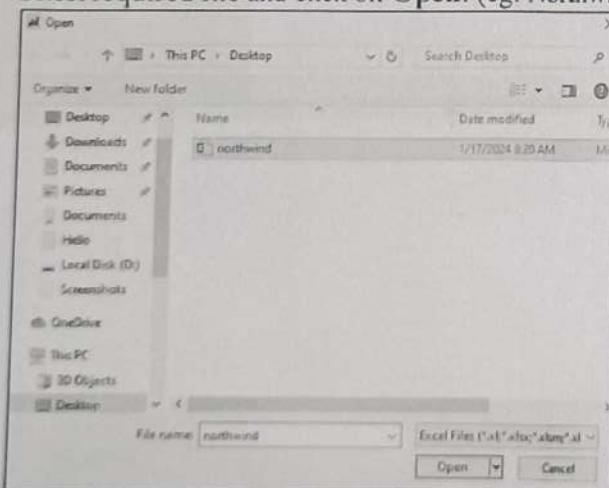
1. Open Power BI.
2. Click on **Get Data**.



3. Following List will be displayed. → Select **Excel** → Click on **Connect**.



4. Select required file and click on **Open**. (eg. Northwind)



Business Intelligence Practical Journal

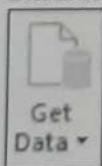
5. Navigator screen appears. → Select file and click on Load.

The screenshot shows the Power BI Navigator interface. On the left, there's a tree view with 'northwind.xlsx (2)' selected, under which 'Table_northwindCustomers' is expanded. The main area displays a preview of the 'Customers' table with columns: ID, CustomerName, LastName, FirstName, and TitleofAddress. The preview shows 22 rows of data. A message at the bottom states: 'The data in the preview has been truncated due to size limits.' At the bottom right are 'Load', 'Edit', and 'Cancel' buttons.

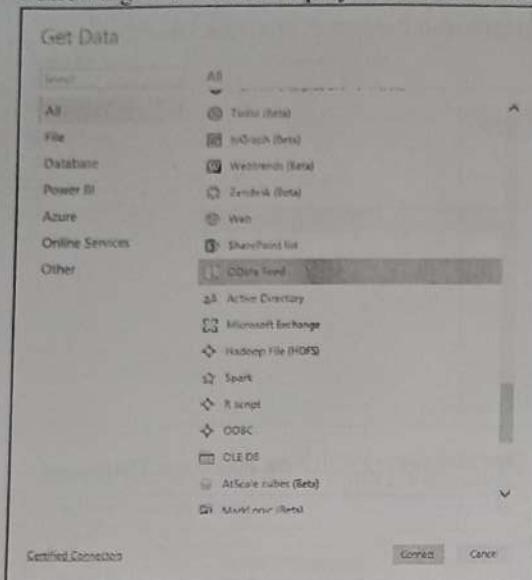
6. Power Query Editor appears.

The screenshot shows the Power Query Editor window. The ribbon menu includes Home, View, Modeling, and File. The main area is currently empty. On the right side, there are two panes: 'VISUALIZATIONS' and 'FIELDS'. The 'FIELDS' pane shows a list of columns from the 'northwindCustomers' table, including CustomerID, CustomerName, LastName, FirstName, and TitleofAddress. There are also sections for 'FILTERS' and 'DRILLTHROUGH'.

7. Click on Get Data.

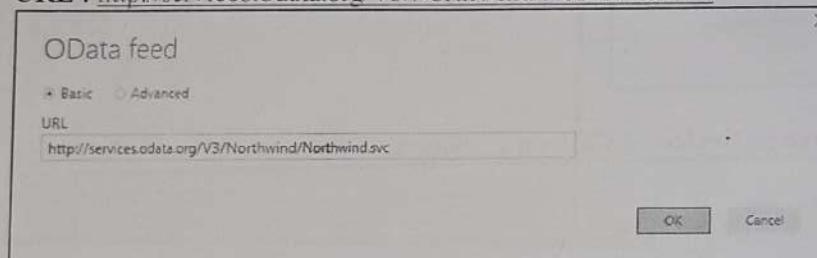


8. Following List will be displayed. → Select OData Feed.



9. Paste URL and Click on OK.

URL : <http://services.odata.org/V3/Northwind/Northwind.svc/>

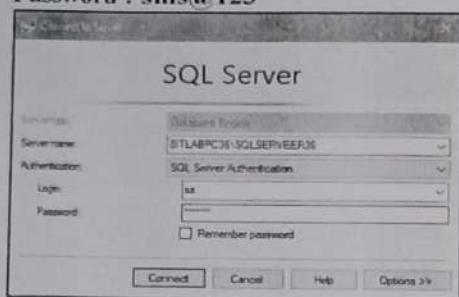


10. Select Orders, Product Tables and click on Load to view table.

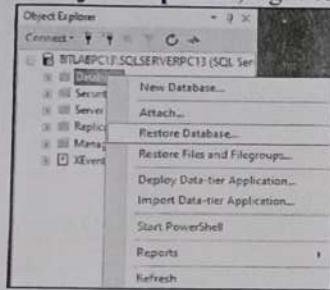
The screenshot shows the Power BI Navigator interface. On the left, the 'Navigator' pane lists various data sources and tables, with 'Products' selected. In the center, the 'Products' table is displayed with columns: ProductID, ProductName, SupplierID, CategoryID, and Quant. The data includes items like 'Aniseed Syrup', 'Chef Anton's Gumbo Mix', and 'Grandma's Organic Dried Plums'. On the right, the 'FIELDS' pane shows a search bar and a list of selected fields: 'Orders', 'Products', 'Sheet1', and 'Table_northwind_acesso'.

Business Intelligence Practical Journal

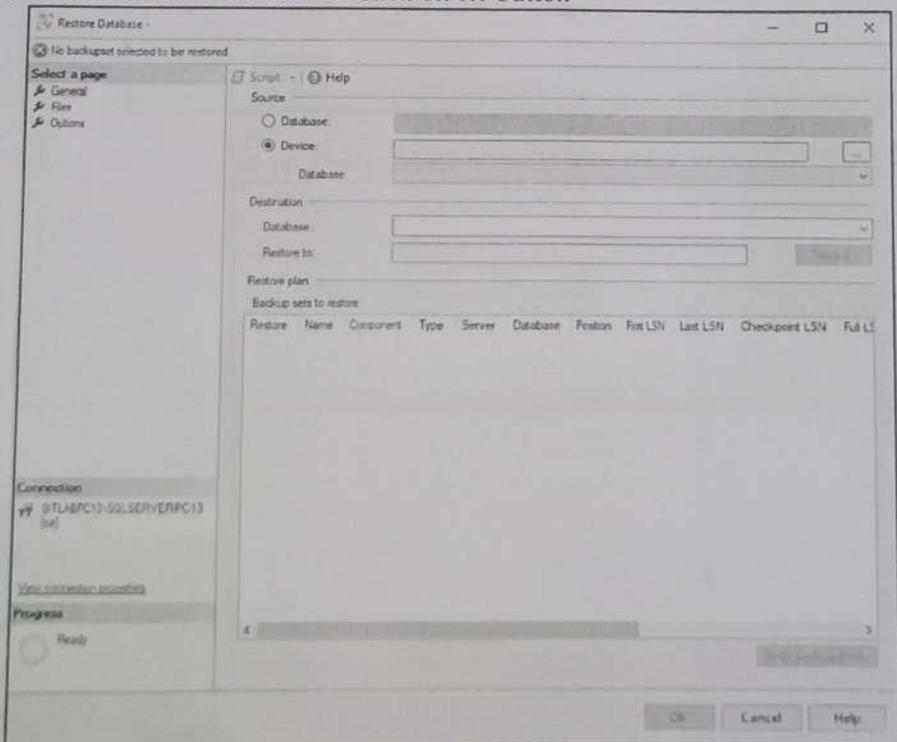
11. Open SQL Server Management Studio 2012.
12. Connect Database Engine. Enter Login and Password. → Click on Connect.
Login : sa
Password : sms@123



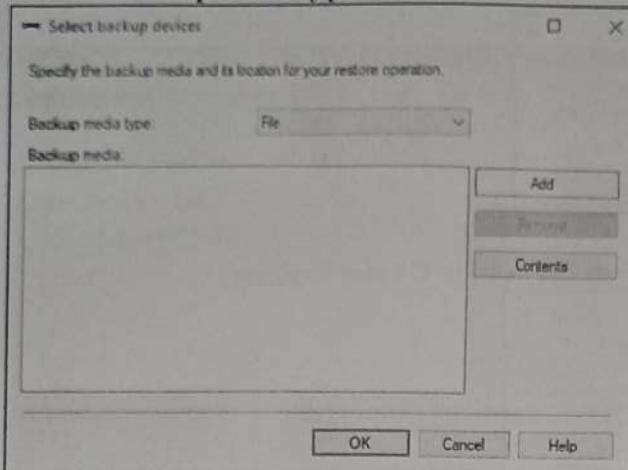
13. In **Object Explorer**, right click on **Databases**. → Click on **Restore Database...**



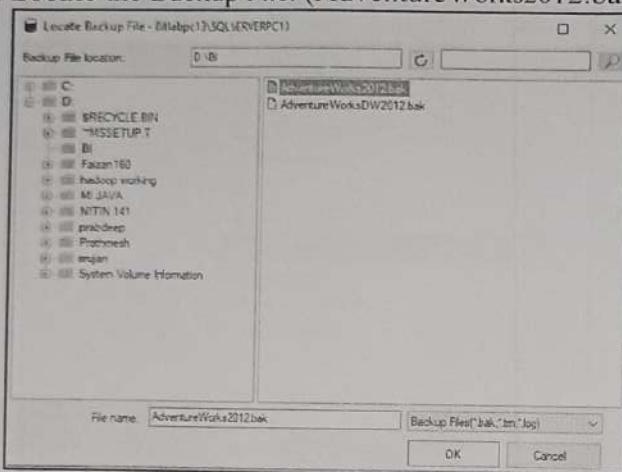
14. Select Source as Device. → Click on ... button



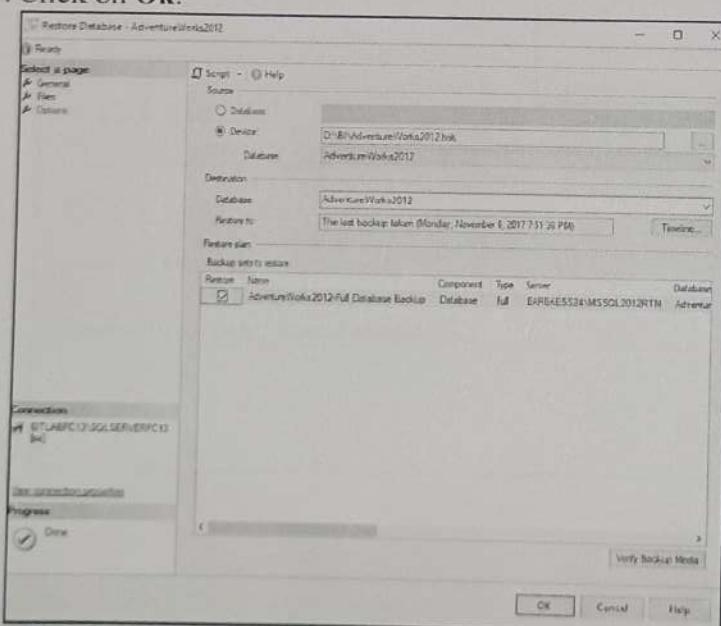
15. Choose Backup media type as File. → Click on Add.



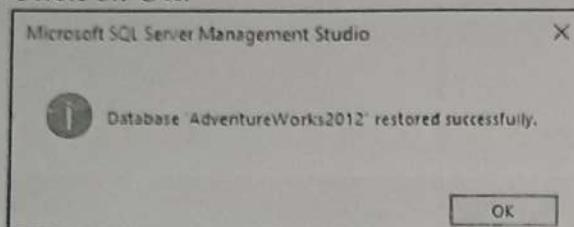
16. Locate the Backup File. (AdventureWorks2012.bak) → Click on Ok.



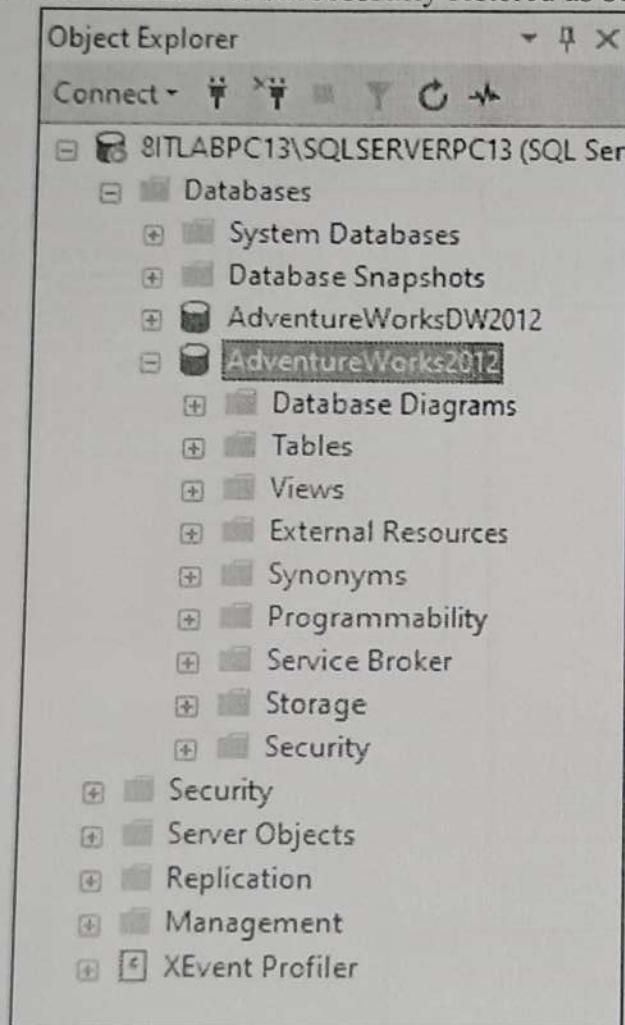
17. Click on Ok.



18. Click on Ok.



19. The database is successfully restored as seen in **Object Explorer**.

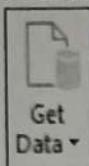


Practical No. 2

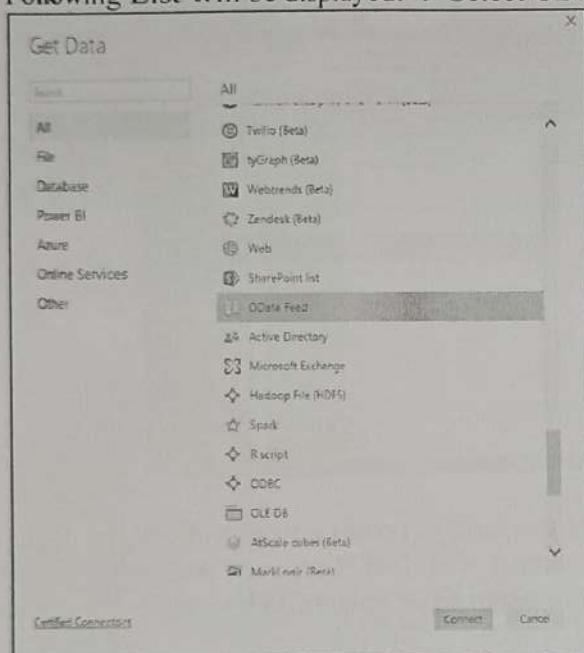
- (A) Perform the Extraction Transformation and Loading (ETL) process to construct the database in Power BI.**

Steps to follow:

1. Open Power BI.
2. Click on Get Data.

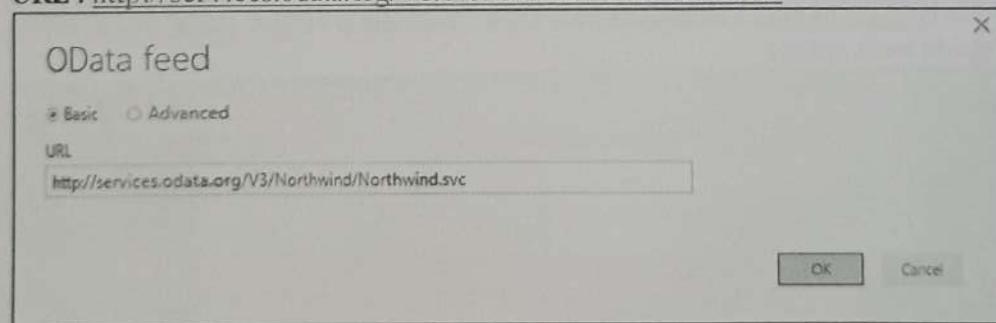


3. Following List will be displayed. → Select **OData Feed**.



4. Paste URL and Click on OK.

URL : <http://services.odata.org/V3/Northwind/Northwind.svc/>



5. Select Orders, Product Tables and click on Load to view table.

The screenshot shows the Microsoft Power BI Data View interface. At the top, there's a 'Navigator' pane on the left containing a tree view of various datasets and tables. The main area displays the 'Products' table with columns: ProductID, ProductName, SupplierID, Discontinued, and UnitPrice. Below the table is a 'Load' button. To the right of the table, there's a 'FIELDS' pane showing the structure of the table. Below the table, the 'Query Editor' is open, showing the raw data from the 'Products' table. The 'FIELDS' pane on the right of the Query Editor also lists the columns of the table.

6. In **Query Editor**, Select the columns you want to remove to only display columns of interest. → Right Click on selected columns. → Select **Remove Columns**.
 (Use Ctrl+Click to select more than one columns or Shift+Click to select columns that are beside each other.)

OR

In **Query Editor**, Select the columns of interest. → Right Click on selected columns. → Select **Remove Other Columns** to keep only interested columns.
 (Use Ctrl+Click to select more than one columns or Shift+Click to select columns that are beside each other.)

The screenshot shows the Microsoft Power BI Query Editor with the 'Products' table loaded. A context menu is open over the 'QuantityPerUnit' column, with the 'Remove Other Columns' option highlighted. Other options in the menu include 'Copy', 'Remove Columns', 'Remove All Columns', 'Add Column From Examples...', 'Remove Duplicates', 'Remove Errors', 'Replace Values...', 'Fill', 'Change Type', 'Merge Columns', 'Group By...', 'Unpivot Columns', 'Union Other Columns', 'Unpivot Only Selected Columns', and 'More...'. To the right of the editor, there's a 'QUERY SETTINGS' pane with sections for 'PROPERTIES' (Name: Products) and 'APPLIED STEPS' (Source).

7. Change the datatype of a column

Select UnitInPrice column → Select Whole Number from the Data Type drop-down in the Home ribbon → You can see the datatype changed in column header of the column.

8. Open Orders table.

9. In **Query Viewer**, scroll to **Order_Details** column → Select **Expand**. → Select **OrderID**, **ProductID**, **UnitPrice** and **Quantity**. → Click on **Ok**.

The screenshot shows the Power BI Data Editor interface. On the left, there's a navigation pane with 'General', 'Tables', 'Sheets', 'Orders', and 'Order Details'. The main area displays the 'Orders' table with columns: Order ID, Customer ID, Order Date, and Ship Date. A context menu is open over the 'Order Details' row, and the 'QUERY SETTINGS' dialog is displayed on the right. The 'APPLIED STEPS' section lists 'Source', 'Navigation', and 'Unloaded Order_Details', with 'Add Custom' being the most recent step.

10. Calculate the line total for each Order_Details row.

a. In the Add Column ribbon, click on Add Custom Column.

This screenshot shows the Power Query Editor ribbon. The 'Add Column' tab is highlighted. Other tabs include Home, Transform, View, and Help. The ribbon also features various data type conversion and manipulation tools like 'Merge Columns', 'Index Column', 'Format', 'Statistics', 'Rounding', and 'Information'.

b. In the Custom Column dialog box, enter the formula by selecting from Available Columns and click on << Insert to insert the selected column.

Formula:

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

This screenshot shows the 'Custom Column' dialog box. It has a 'Custom' section where the formula `= [Order_Details.UnitPrice] * [Order_Details.Quantity]` is entered. To the right is a list of 'Available columns' from the 'Order_Details' table, including Customer, Employee, Order_Details.OrderID, Order_Details.ProductID, Order_Details.UnitPrice, Order_Details.Quantity, and Shipped. At the bottom, there's a note about system errors and an 'OK' button.

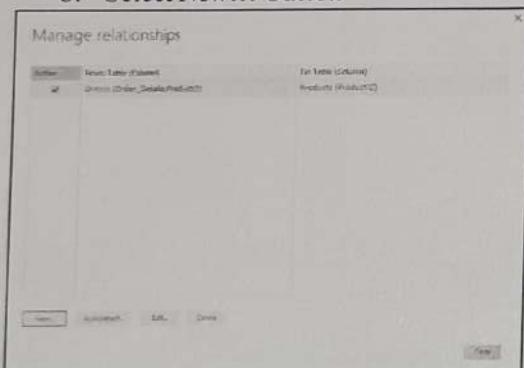
c. Click on Ok.

This screenshot shows the Power BI Data Editor after adding the custom column. The 'Orders' table now includes a new column 'LineTotal' at the end of the row. The data in 'LineTotal' is calculated as the product of 'UnitPrice' and 'Quantity' for each row. The 'QUERY SETTINGS' dialog is still visible on the right.

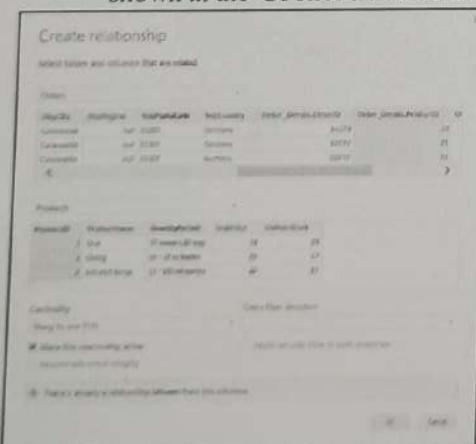
11. From the **Home** ribbon, select **Close & Apply** to load the model we created in **Query Editor**.
12. Power BI does not require you to combine queries to report on them. Instead, you can create relationships between datasets.

a. From **Home** ribbon, select **Manage Relationships**.

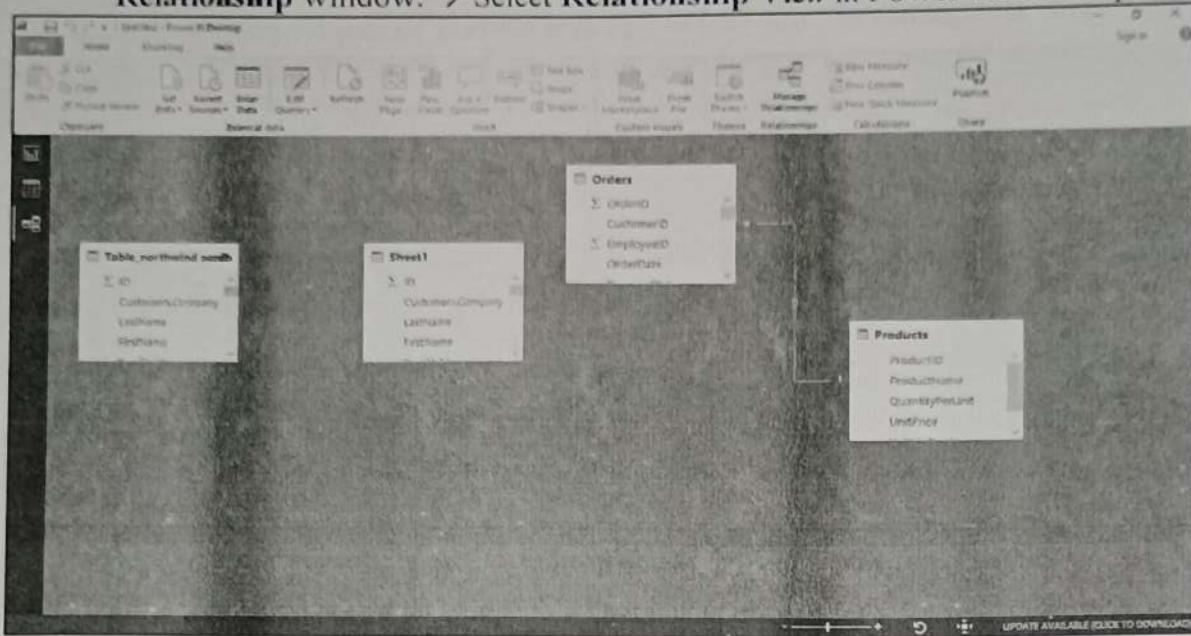
b. Select **New...** button



c. When we attempt to create the relationship, we see that one already exists! As shown in the **Create Relationship** dialog (Yellow Dialog Box).



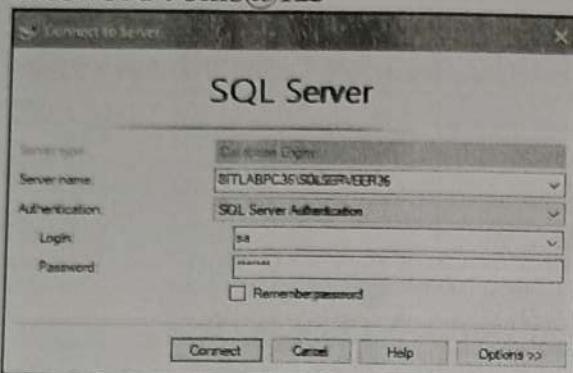
- d. No need to make any changes, so we'll just select **Cancel** and close the **Manage Relationship** window. → Select **Relationship View** in Power BI Desktop.



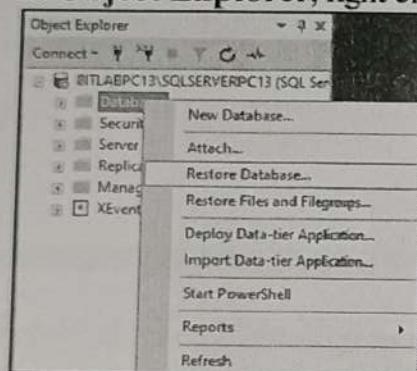
the database in SQLServer.

Steps to follow :

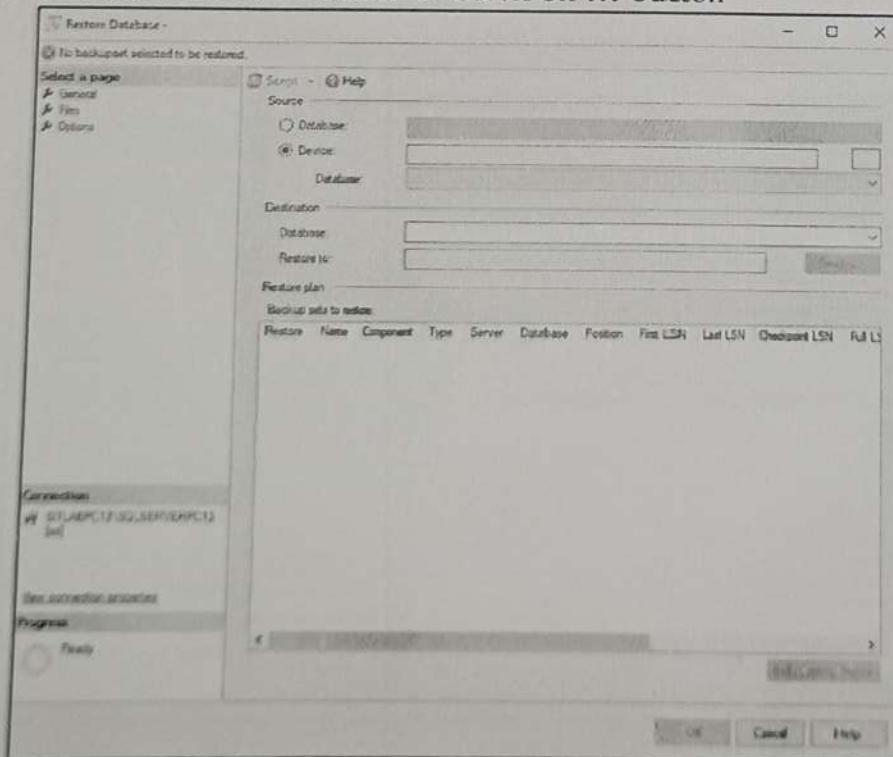
1. Open **SQL Server Management Studio 2012**.
2. Connect **Database Engine**. Enter Login and Password. → Click on **Connect**.
Login : sa
Password : sms@123



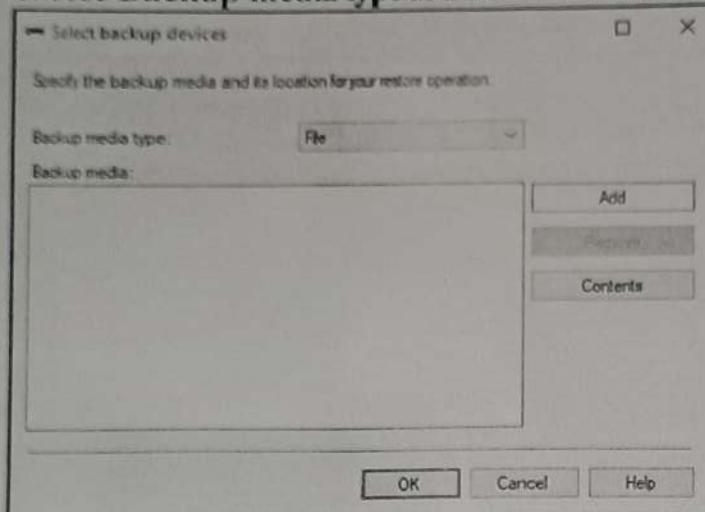
3. In **Object Explorer**, right click on **Databases**. → Click on **Restore Database...**



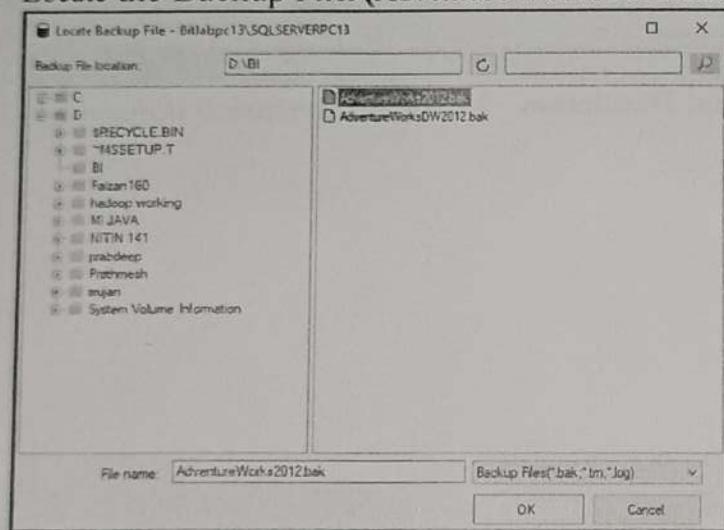
4. Select **Source as Device**. → Click on ... button



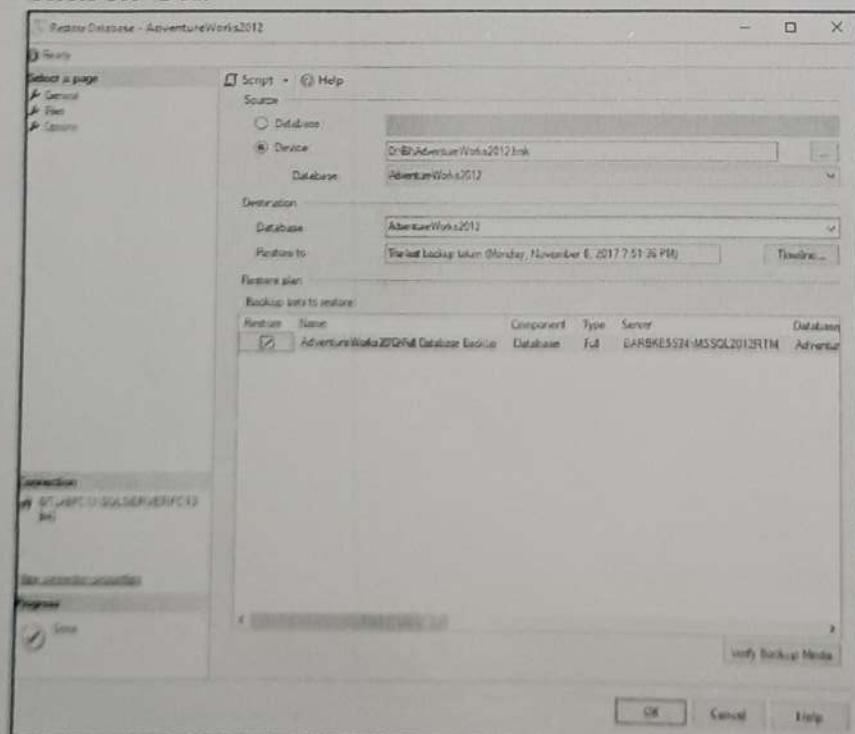
5. Choose Backup media type as File. → Click on Add.



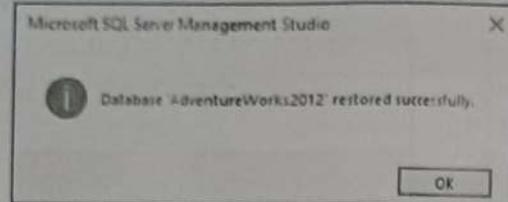
6. Locate the Backup File. (AdventureWorks2012.bak) → Click on Ok.



7. Click on Ok.

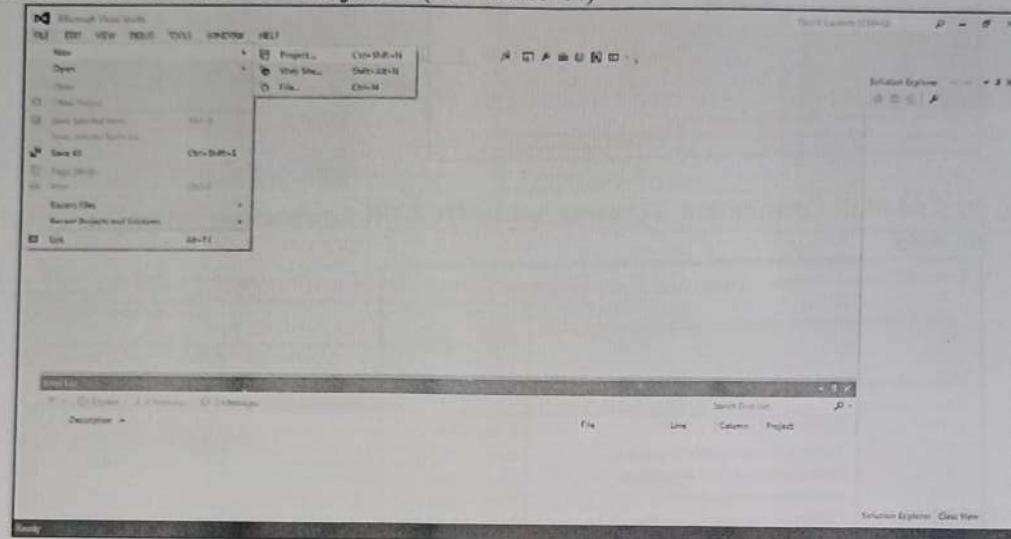


- Click on Ok.

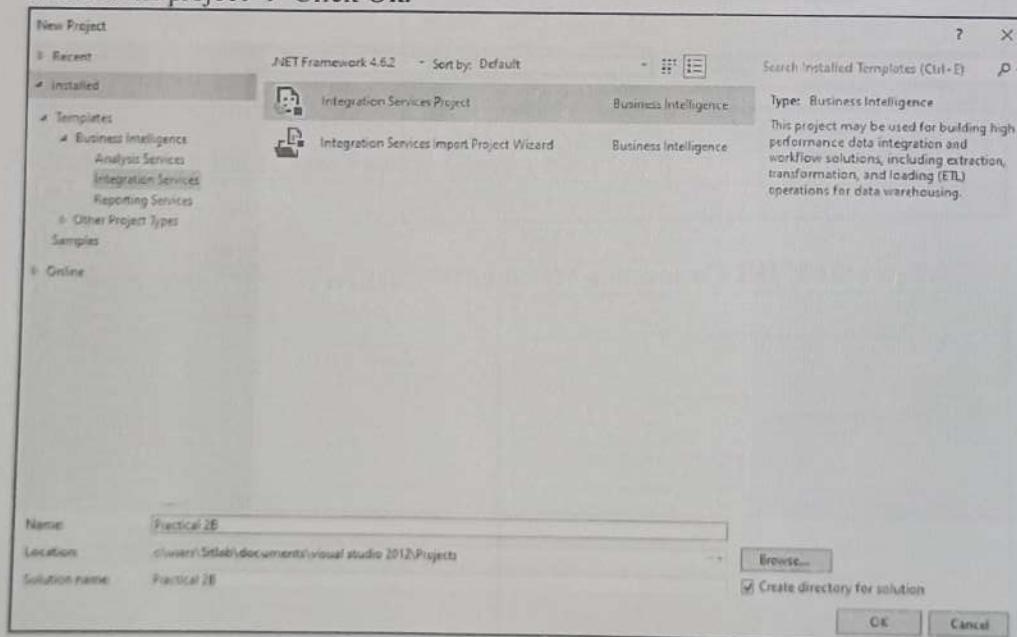


- Open SQL Server Data Tools.

- Select File → New → Project... (Ctrl+Shift+N)

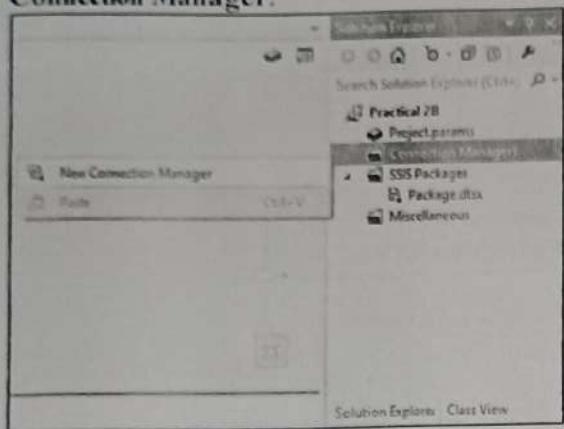


- Select Business Intelligence → Integration Services Project → Give appropriate name to the project → Click Ok.

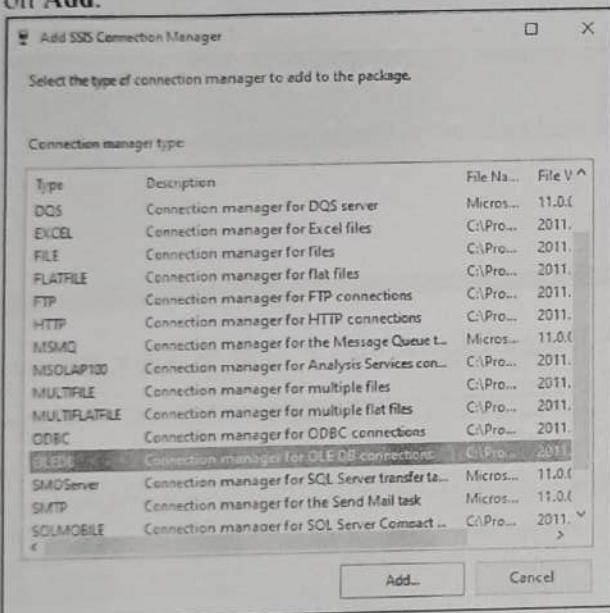


Business Intelligence Practical Journal

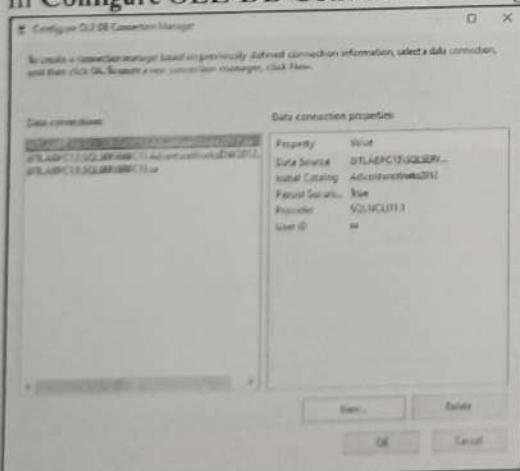
12. In Solution Explorer, right click on Connection Manager. → Click on New Connection Manager.



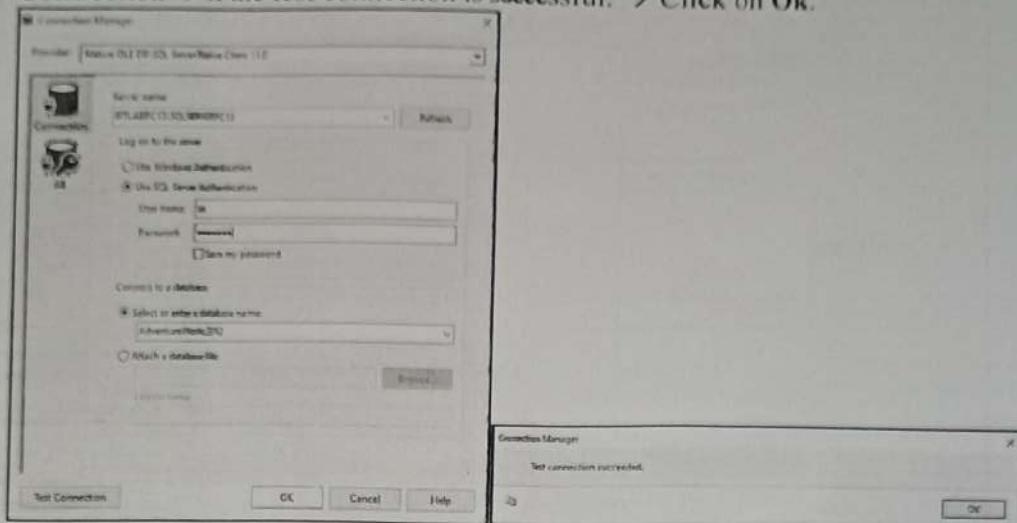
13. In Add SSIS Connection Manager, select OLEDB Connection Manager → Click on Add.



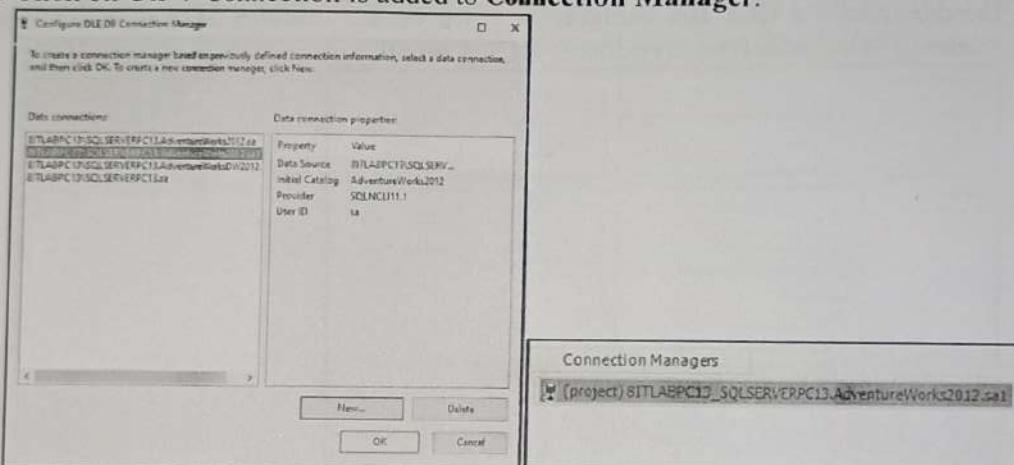
14. In Configure OLE DB Connection Manager → Click on New...



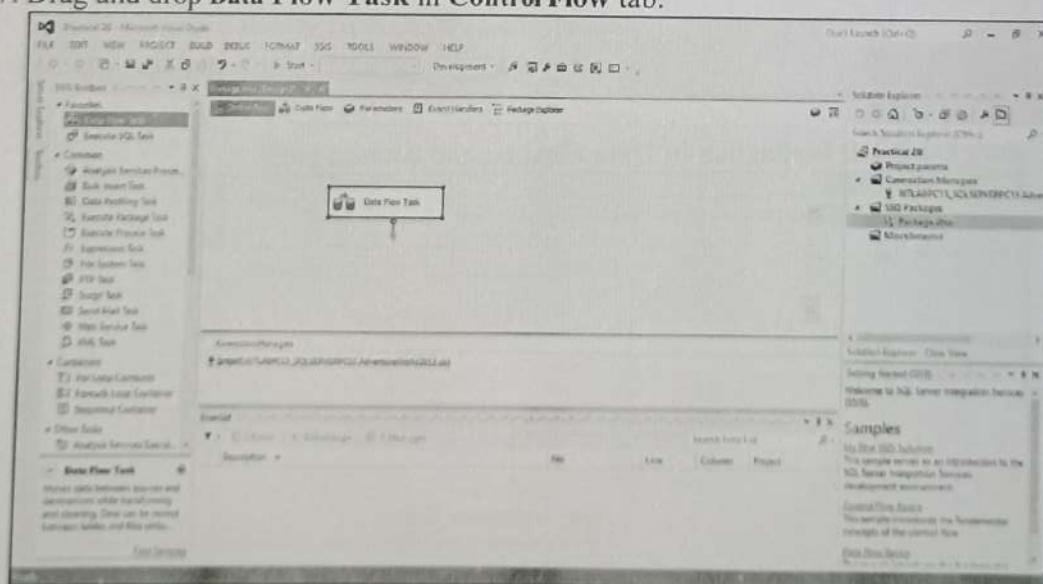
15. Select appropriate server from dropdown and log on to the server. → Click on **Test Connection** → If the test connection is successful. → Click on **Ok**.



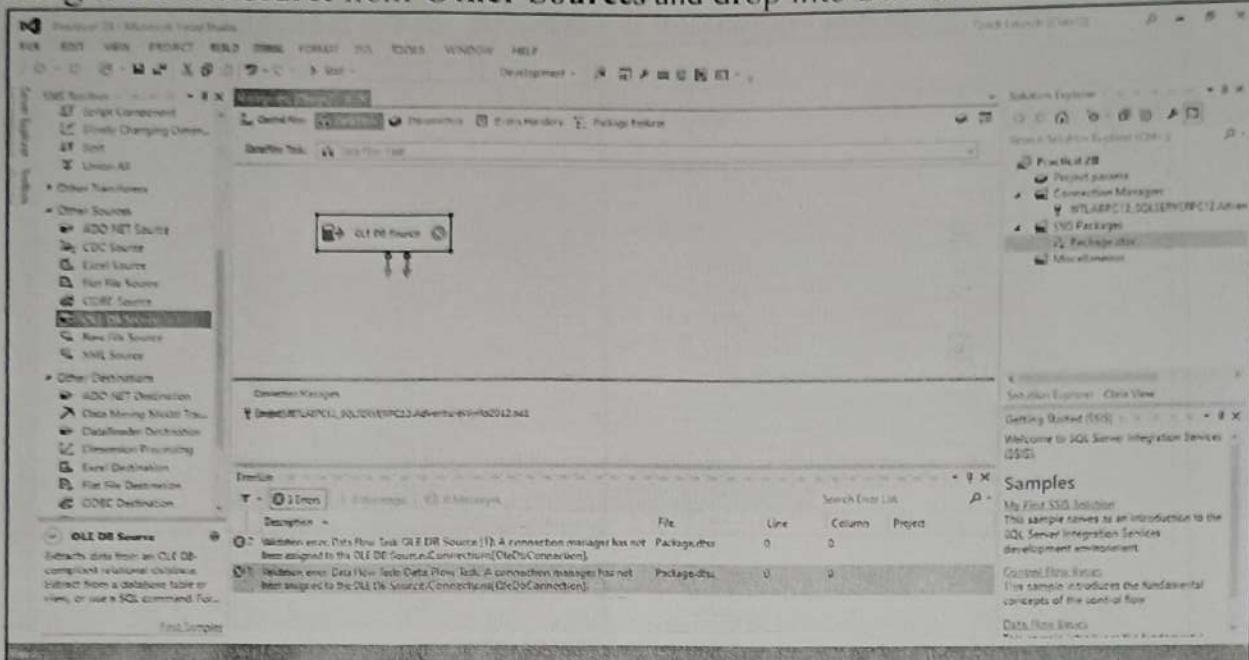
16. Click on **Ok** → Connection is added to **Connection Manager**.



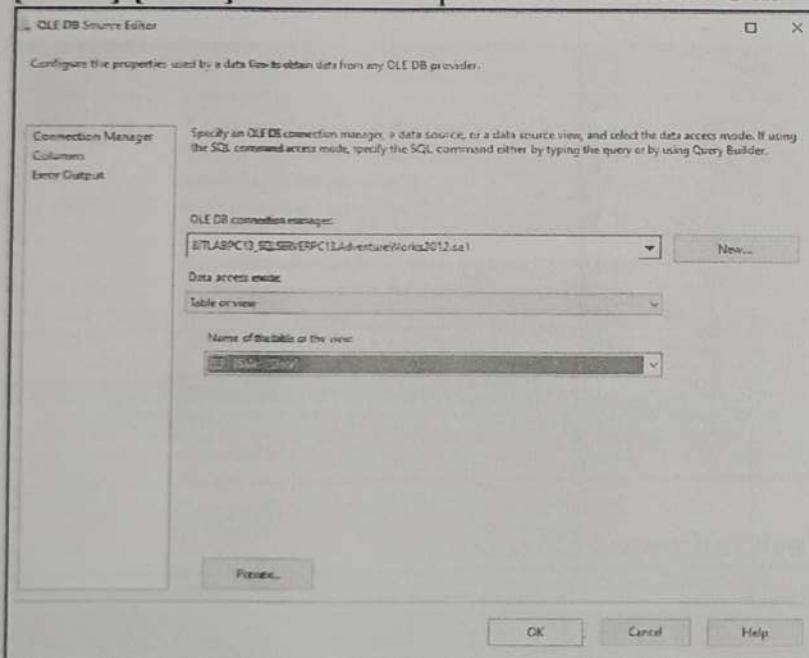
17. Drag and drop **Data Flow Task** in Control Flow tab.



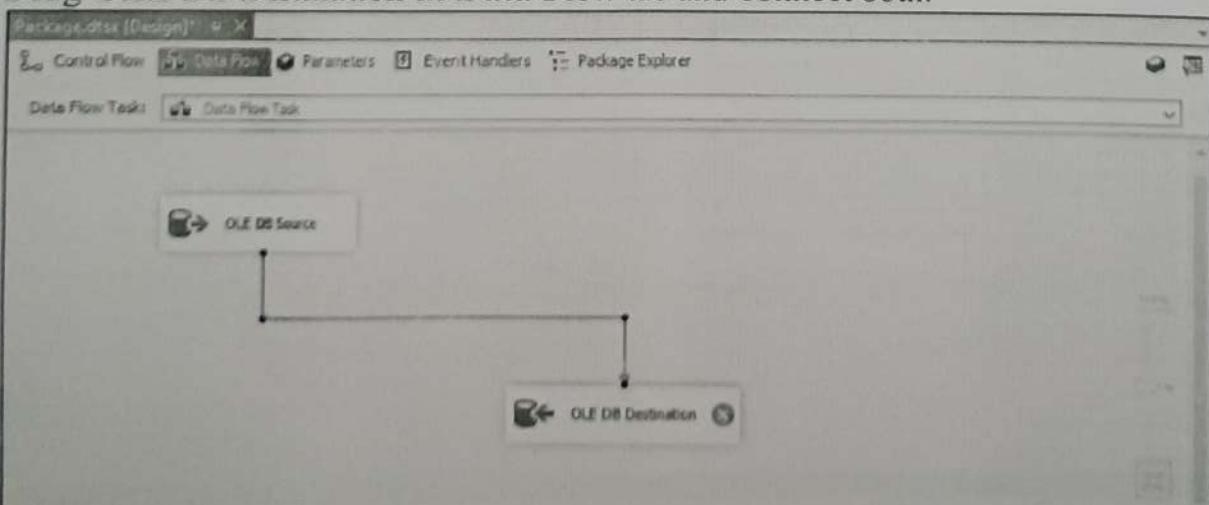
18. Drag OLE DB Source from Other Sources and drop into Data Flow tab.



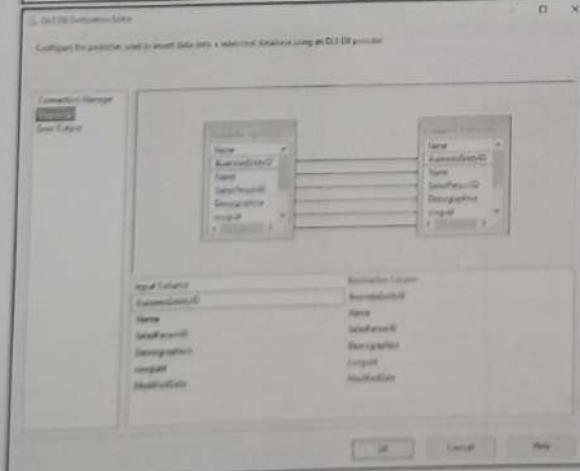
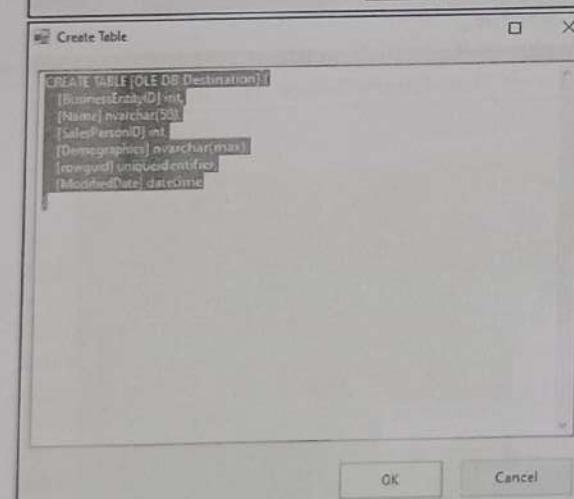
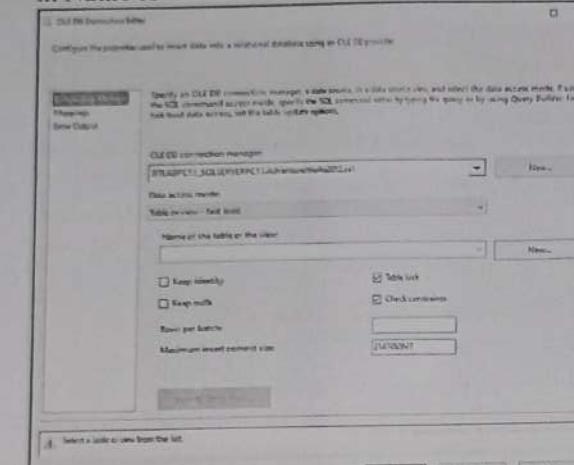
19. Double click on OLE DB Source. → Click on Connection Manager. → Select [Sales].[Sales] table from dropdown → Click on Ok.



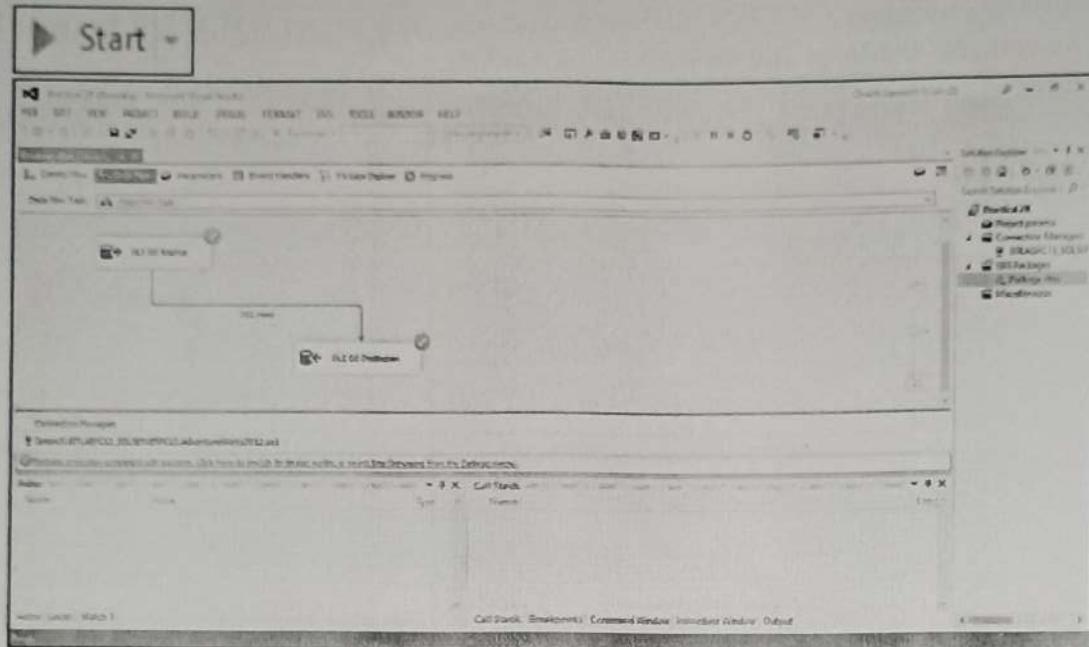
20. Drag OLE DB Destination in Data Flow tab and connect both.



21. Double click on **OLE DB Destination**. → Select [**OLE DB Destination**] in Name of the table or the view OR Click on New to run the query to get [**OLE DB Destination**] in Name of the table or the view → Click on Ok.

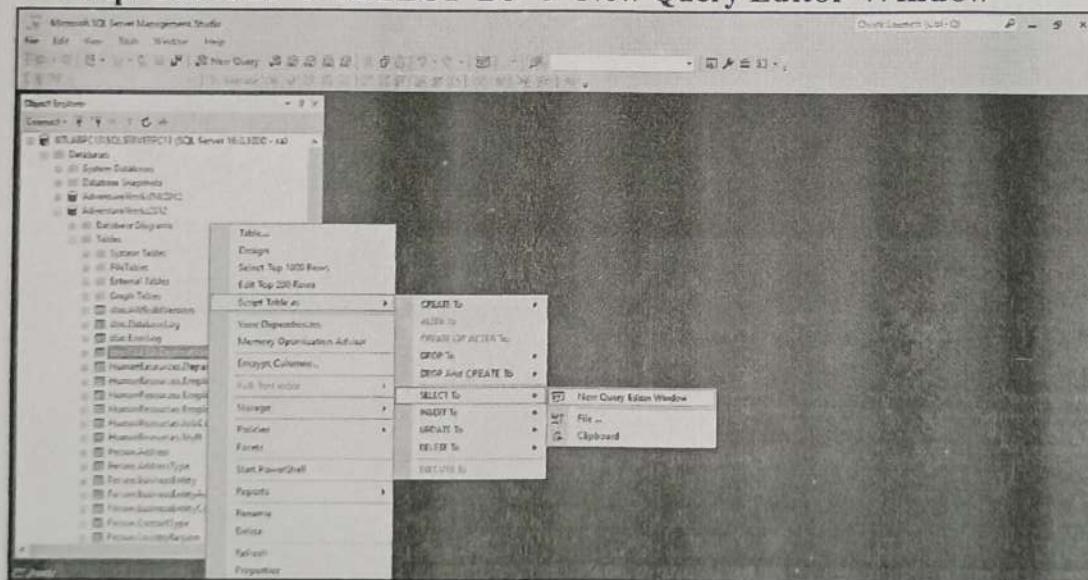


22. Click on Start.



23. Go to **SQL Server Management Studio**.

24. In Database tab → AdventureWorks → Right click on **[dbo].[OLE DB Destination]** → Script Table as → SELECT To → New Query Editor Window



25. Execute following query to get output.

```
SQLQuery1.sql - 8IT...Works2012 (sa (61))  ↗ X
USE [Adventureworks2012]
GO

SELECT [BusinessEntityID]
      ,[Name]
      ,[SalesPersonID]
      ,[Demographics]
      ,[rowguid]
      ,[ModifiedDate]
  FROM [dbo].[OLE DB Destination]

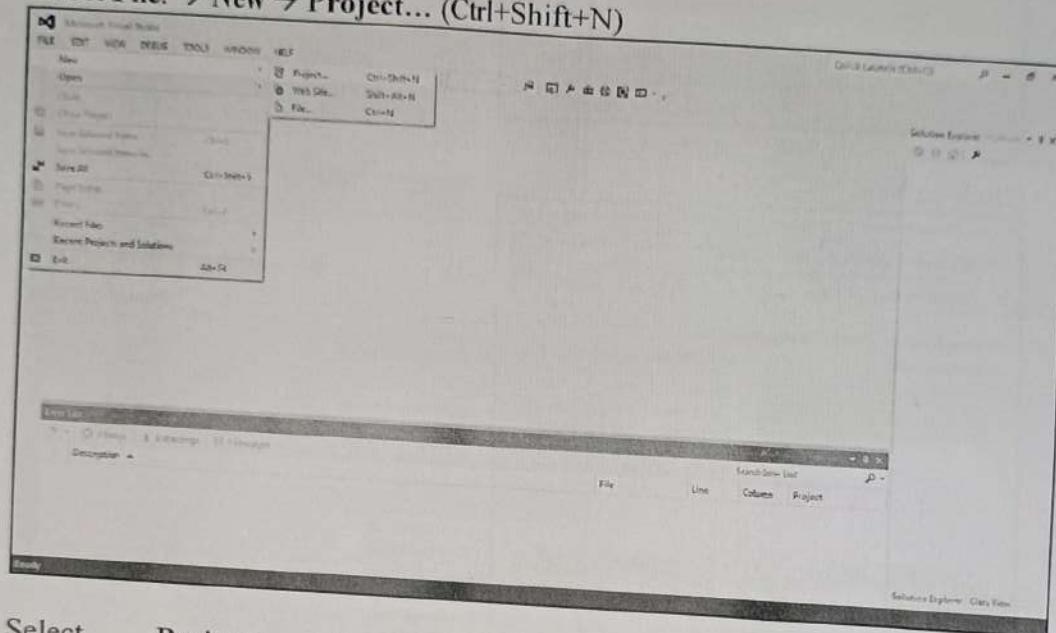
GO
```

Practical No. 3

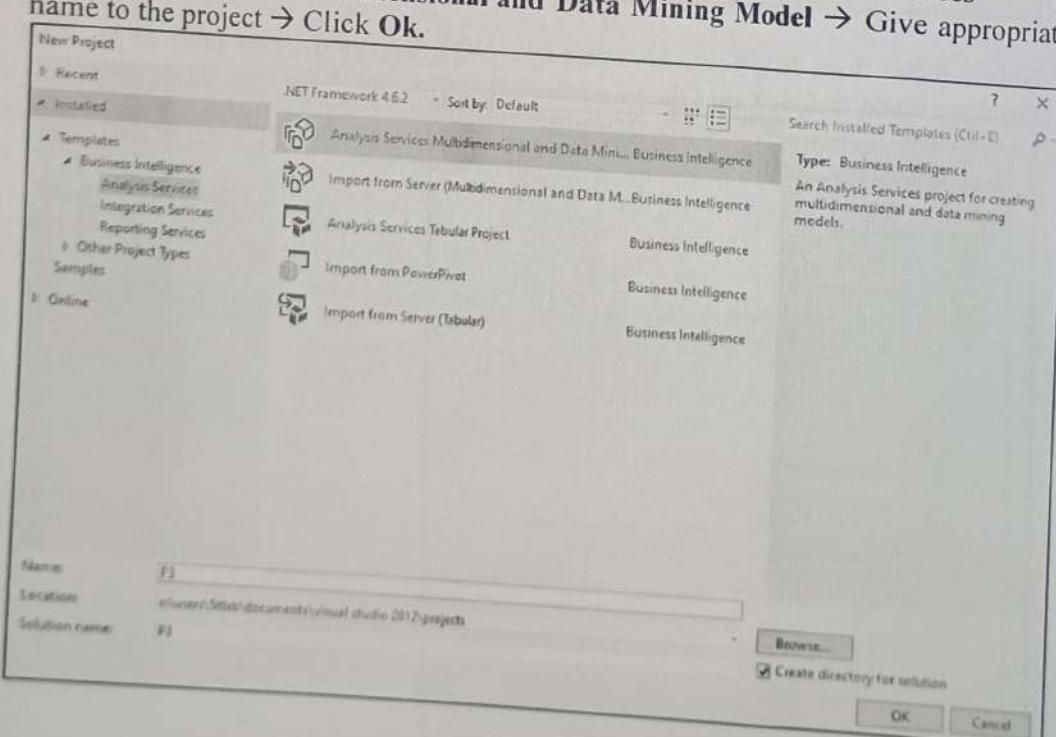
Create a cube with suitable dimensions and fact tables based on OLAP.

Steps to follow :

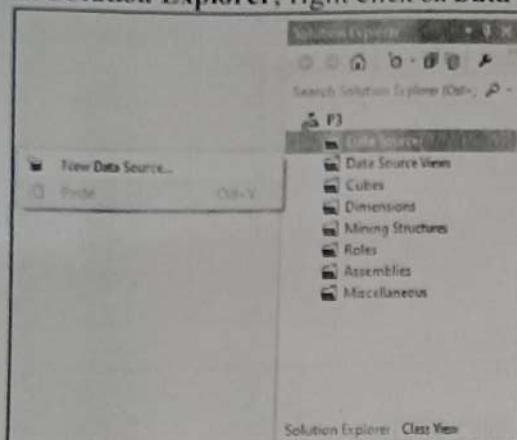
1. Open SQL Server Management Studio 2012.
2. Restore AdventureWorks2012 database.
3. Open SQL Server Data Tools.
4. Select File. → New → Project... (Ctrl+Shift+N)



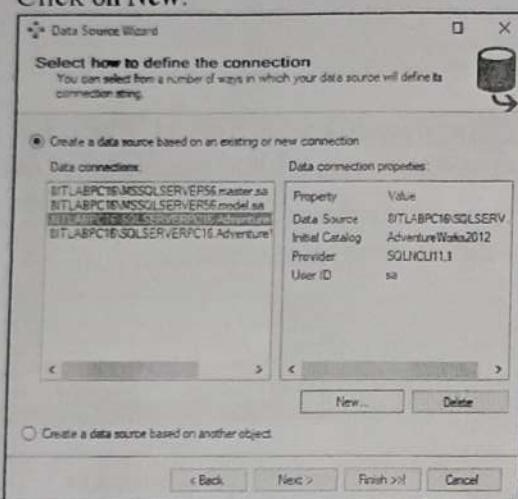
5. Select Business Intelligence → Analysis Services → Analysis Services Multidimensional and Data Mining Model → Give appropriate name to the project → Click Ok.



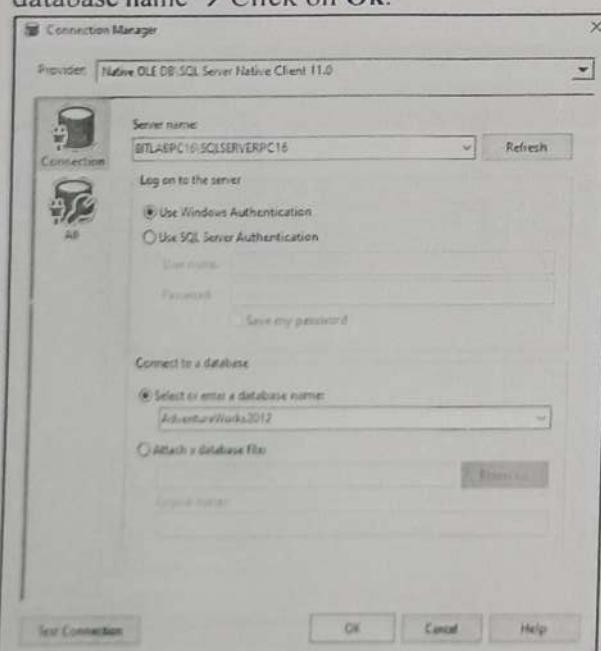
6. In Solution Explorer, right click on **Data Source** → New Data Source.



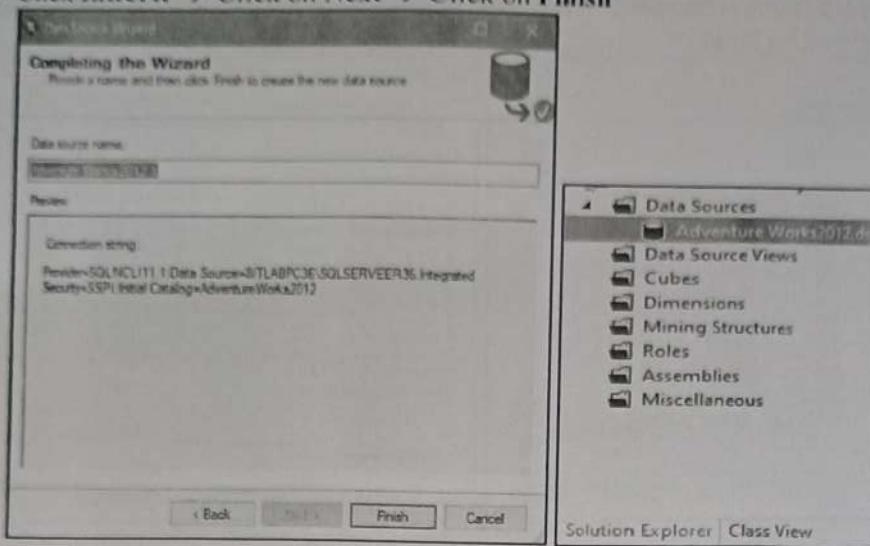
7. Click on **New**.



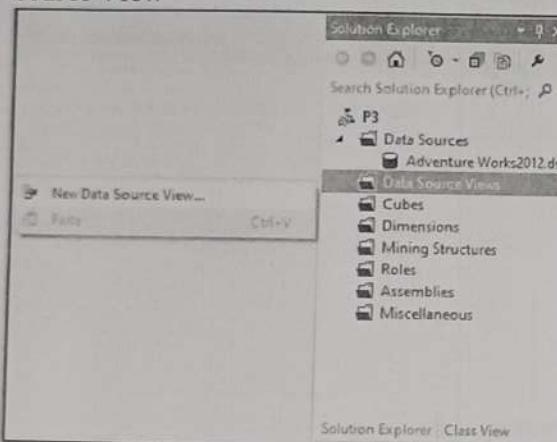
8. Select server from dropdown → Select **Use SQL Server Authentication** → Select a database name → Click on **Ok**.



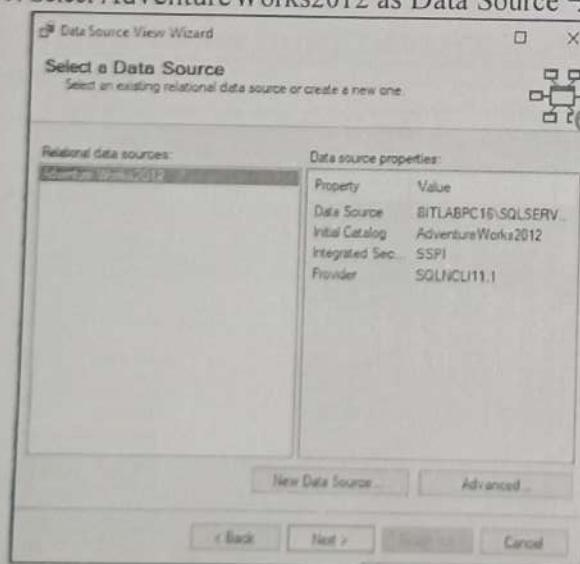
9. Click Inherit → Click on Next → Click on Finish



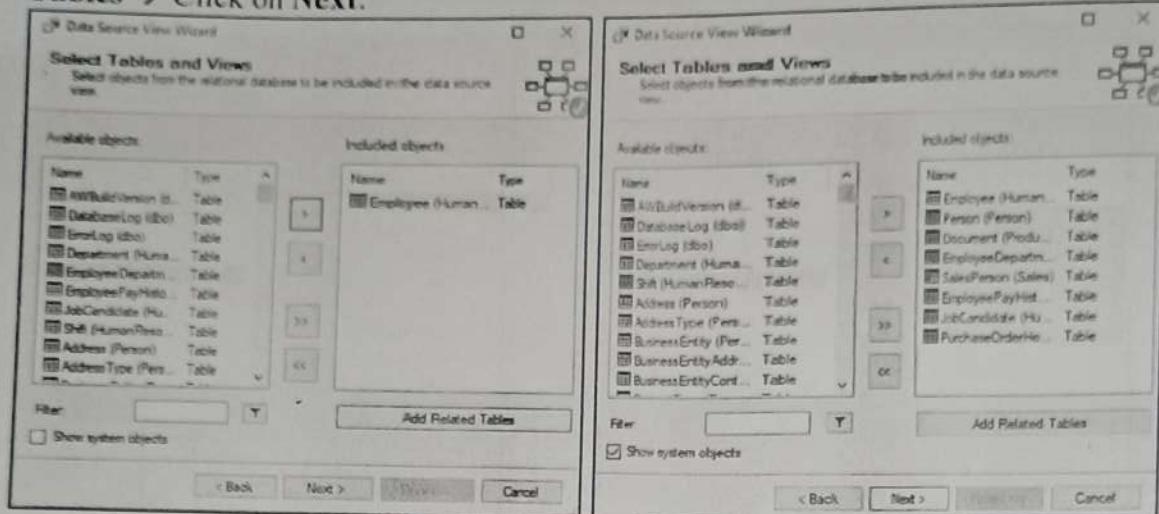
10. In Solution Explorer → Right Click on Data Source View → Select New Data Source View



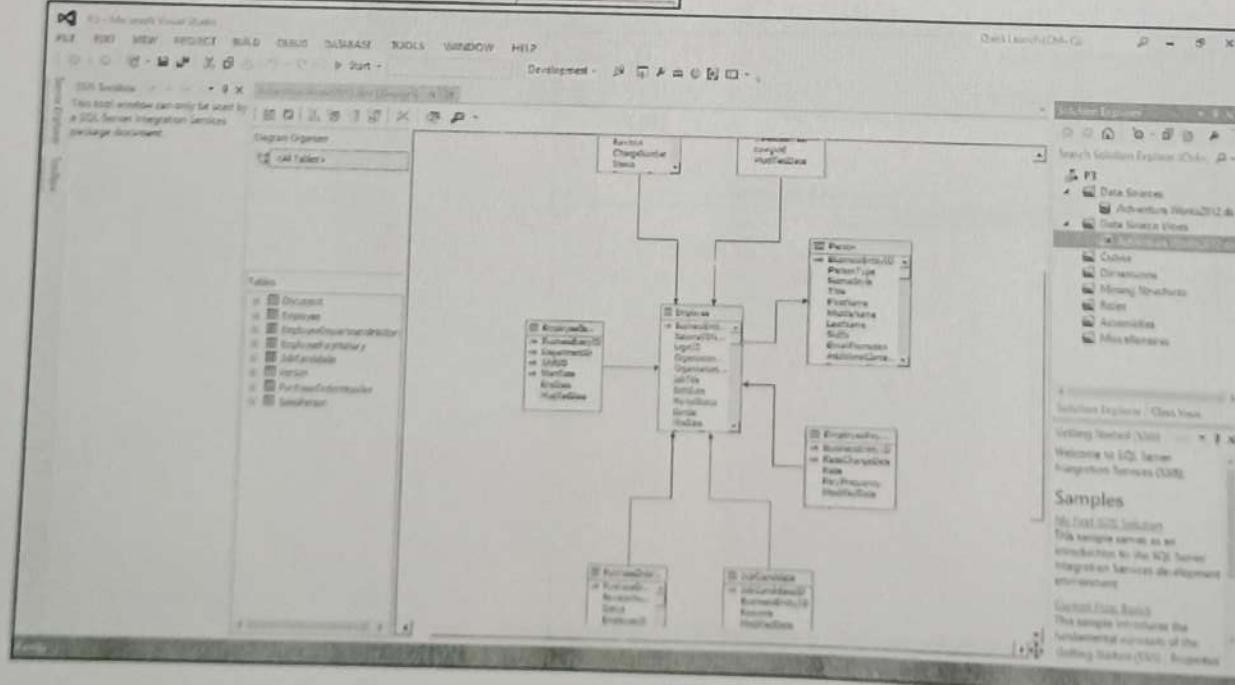
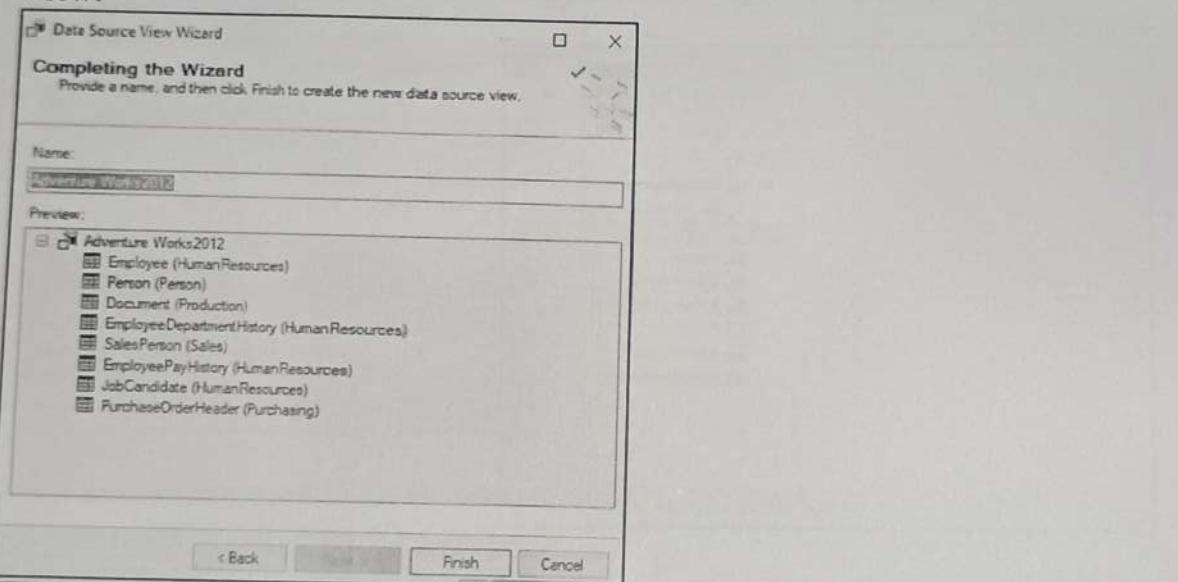
11. Select AdventureWorks2012 as Data Source → Click on Next.



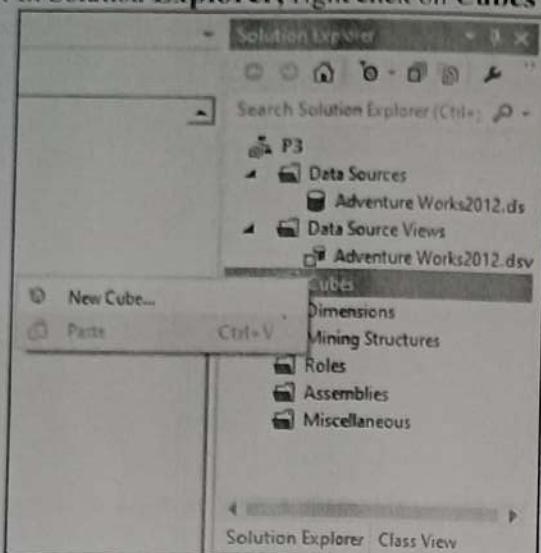
12. Select EmployeeDepartment [dbo] from Available objects. → Click on Add Related Tables → Click on Next.



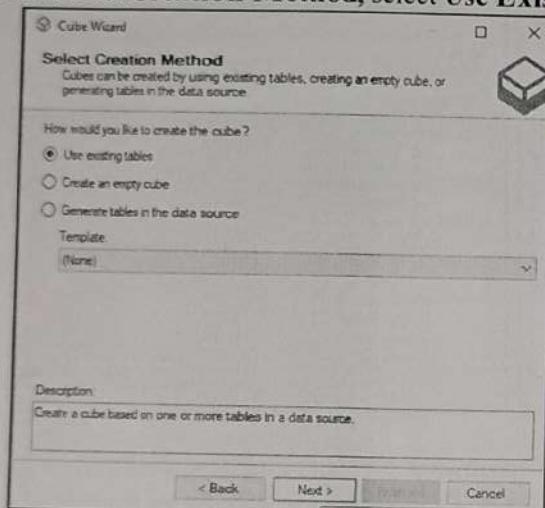
13. Click on Finish → In Solution Explorer, AdventureWorks2012.dsv in Data Source View.



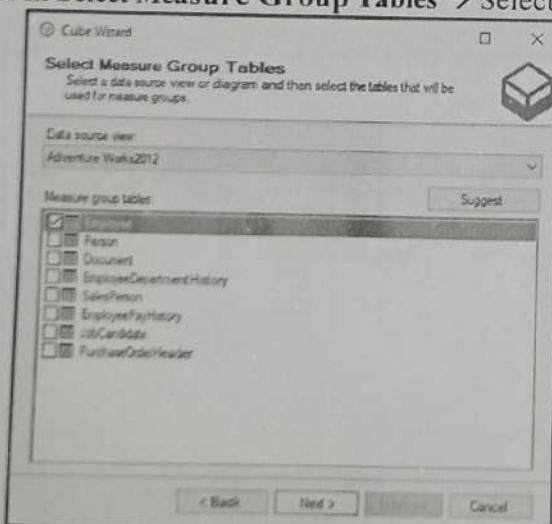
14. In Solution Explorer, right click on Cubes → Click on New Cube.



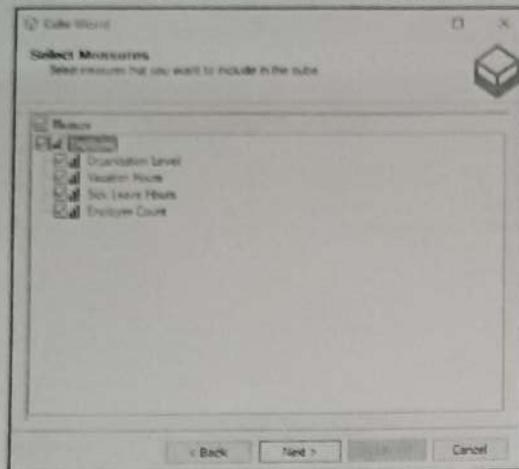
15. In Select Creation Method, select Use Existing Tables → Click on Next.



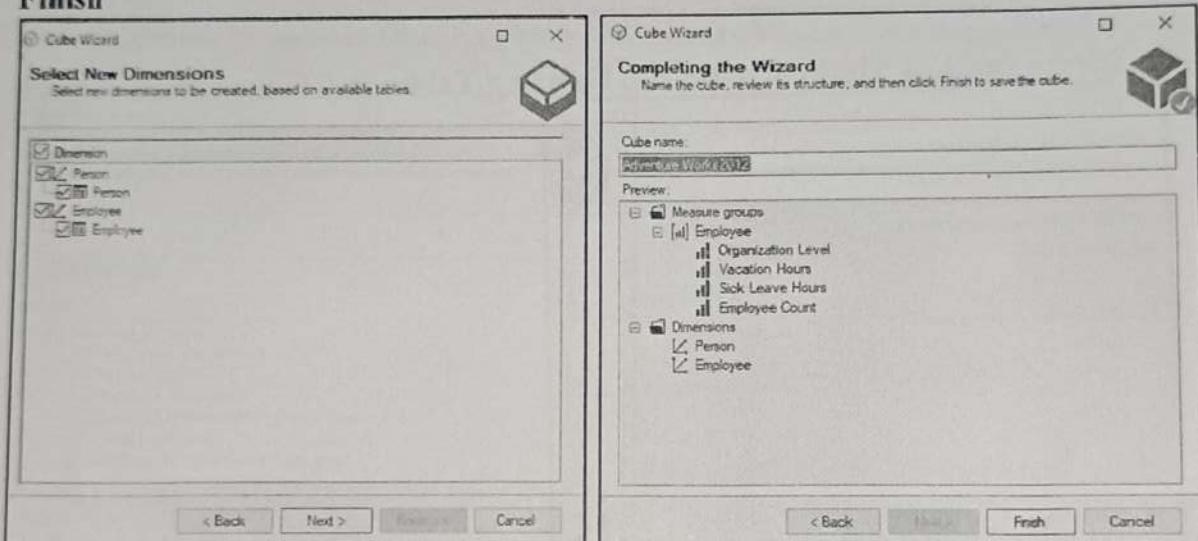
16. In Select Measure Group Tables → Select Employee → Click on Next



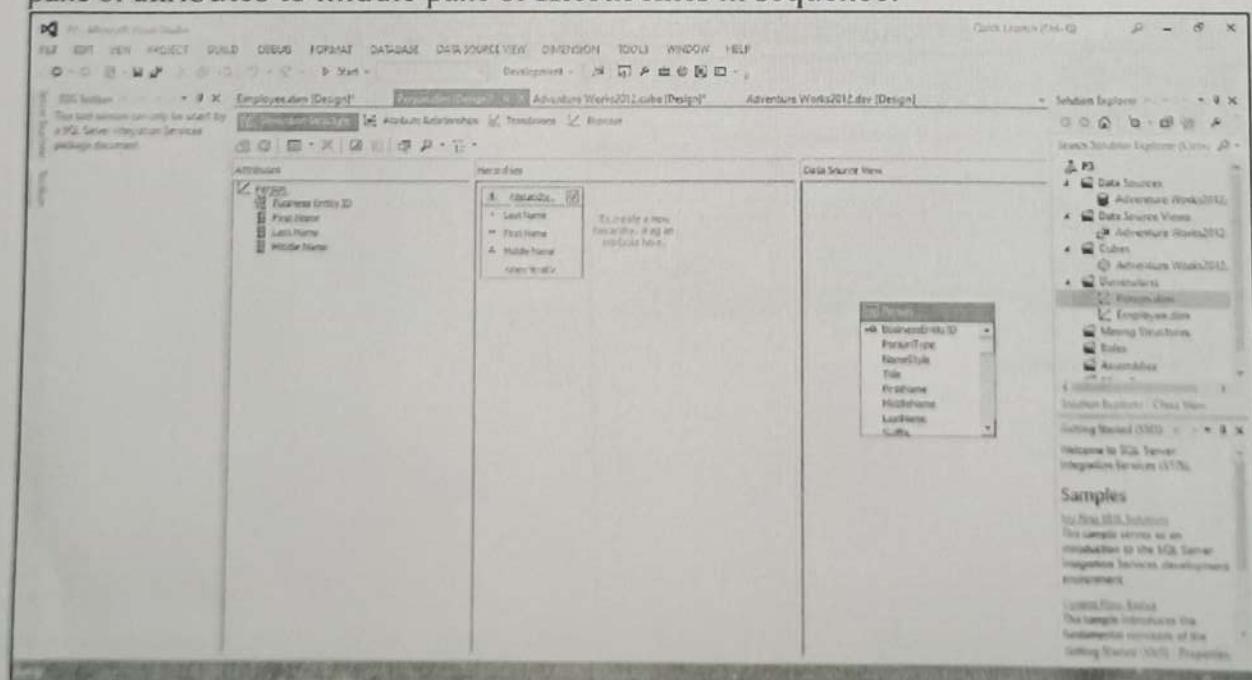
17. In Select Measures → Check all measures → Click on Next



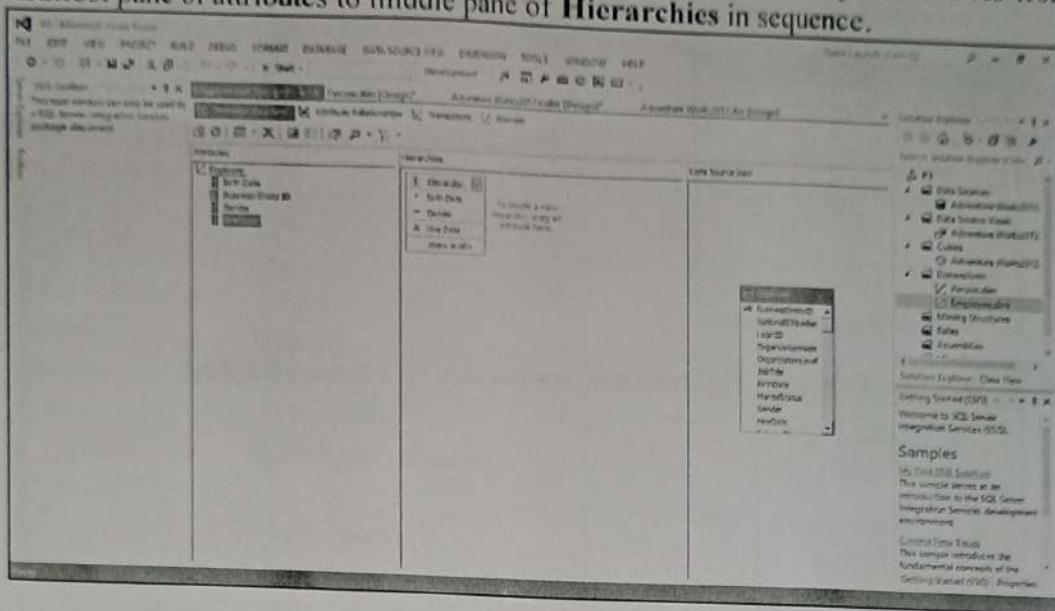
18. In Select New Dimension → Check All Dimensions → Click on Next → Click on Finish



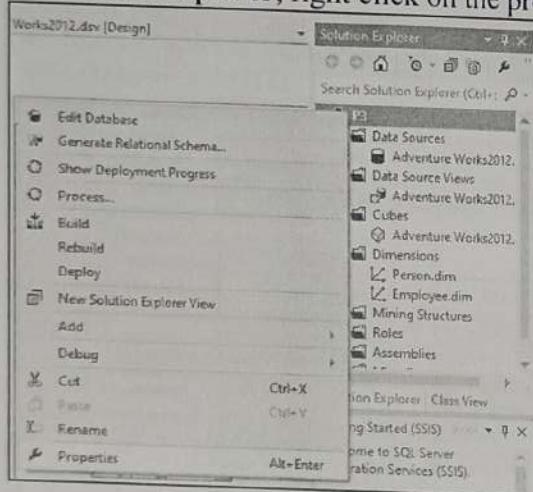
19. In Dimensions tab → Double click on Person.dim → Drag and drop fields from table shown in Data Source View to Attributes → Drag and drop Attributes from leftmost pane of attributes to middle pane of Hierarchies in sequence.



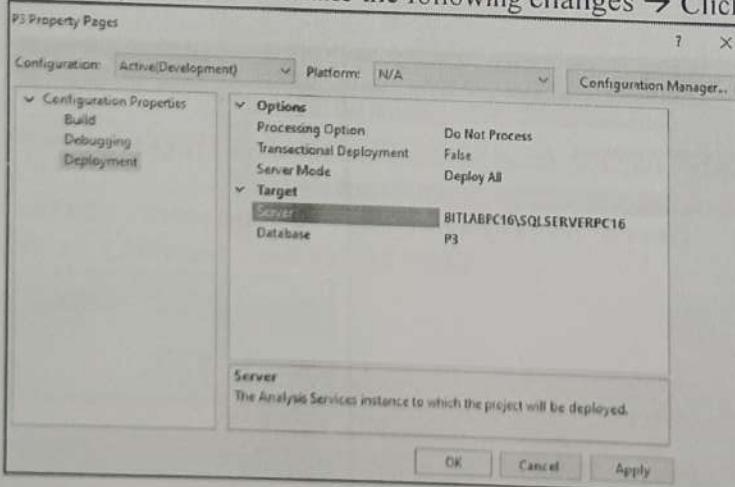
20. In Dimensions tab → Double click on Employee.dim → Drag and drop fields from table shown in Data Source View to Attributes → Drag and drop Attributes from leftmost pane of attributes to middle pane of Hierarchies in sequence.



21. In Solution Explorer, right click on the project → Click on Properties.



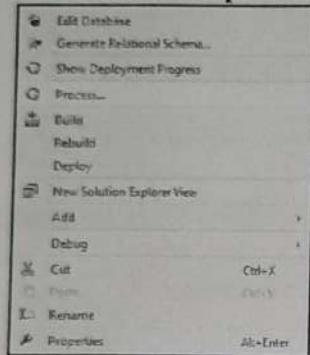
22. In Deployment tab → Make the following changes → Click on Apply → Click on Ok.



23. In Solution Explorer, right click on the project → Click on Deploy.



24. In Solution Explorer, right click on the project → Click on Deploy.



25. Click on Run

The image contains two overlapping windows. The top window is the 'Project Properties' dialog for 'MultidimensionalProject8'. It has tabs for 'General', 'Advanced', 'Project Options', 'Process...', and 'Settings'. The 'Process...' tab is active. The bottom window is the 'Deployment Progress' window titled 'Deployment Progress - MultidimensionalProject8'. It shows a tree view of processing tasks: 'Command' (Processing Database 'multidimensionalproject8' completed), 'Cube' (Processing Cube 'Adventure WorksDW' completed, duration 0:00:05), 'Dimension Group' (Processing Dimension Group 'Employee' completed), 'Dimension' (Processing Dimension 'Employee' completed, duration 0:00:04), and 'Dimension Range' (Processing Dimension Range 'Employee' completed, duration 0:00:04). The status bar at the bottom of the deployment progress window says 'Process succeeded.'

Practical No. 4

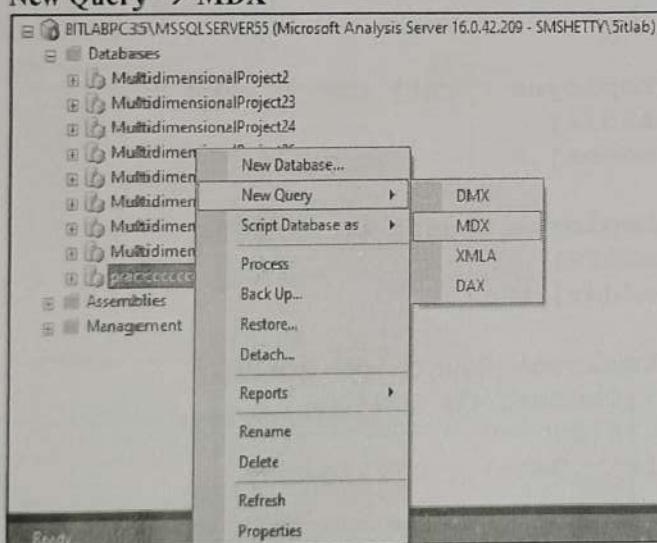
Execute the MDX queries to extract the data from the data warehouse.

Steps to follow :

1. Open SQL Server Management Studio.
2. Connect **Analysis Services**. Enter Server Name according to base machine. → Click on **Connect**.

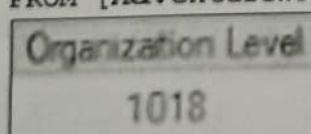


3. In Object Explorer, right click on project name where you have created the cube. → **New Query** → **MDX**



4. Now in the **Queries Section**. Write the query → Click on **Execute** to execute the query.

```
SELECT [Measures].[Organization Level] ON COLUMNS  
FROM [AdventureWorks2012]
```



Business Intelligence Practical Journal

```
SELECT [Measures].[Employee Count] ON COLUMNS  
FROM [AdventureWorks2012]
```

Employee Count	
	290

```
SELECT [Measures].[Employee Count] ON COLUMNS,  
[Employee].[Gender].[Gender] ON ROWS  
FROM [AdventureWorks2012]
```

	Employee Count
F	84
M	206
Unknown	(null)

```
SELECT [Measures].[Vacation Hours] ON COLUMNS  
FROM [AdventureWorks2012]
```

Vacation Hours	
	14678

```
SELECT [Measures].[Employee Count] ON COLUMNS  
FROM [AdventureWorks2012]  
WHERE [Employee].[Gender].F
```

```
SELECT [Measures].[Employee Count] ON COLUMNS  
FROM [AdventureWorks2012]  
WHERE [Employee].[Gender].M
```

```
SELECT [Measures].[Employee Count] ON COLUMNS,  
[Employee].[Gender].[Gender] ON ROWS  
FROM [AdventureWorks2012]  
WHERE [Employee].[Birth Date].[1951-10-17]
```

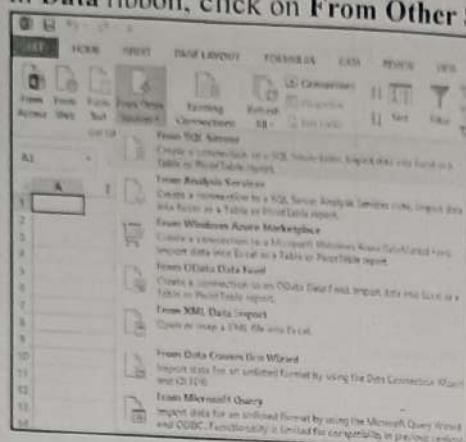
```
SELECT [Measures].[Vacation Hours] ON COLUMNS  
FROM [AdventureWorks2012]  
WHERE [Person].[First Name].[Stephen]
```

Practical No. 5

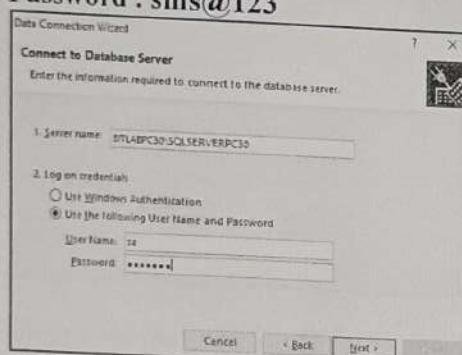
- (A) Import the datawarehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.

Steps to follow :

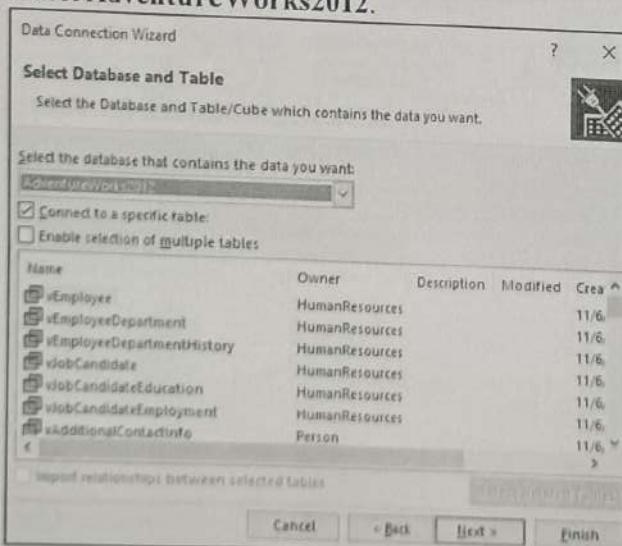
1. Open Excel.
2. In Data ribbon, click on From Other Sources. → Select From SQL Server.



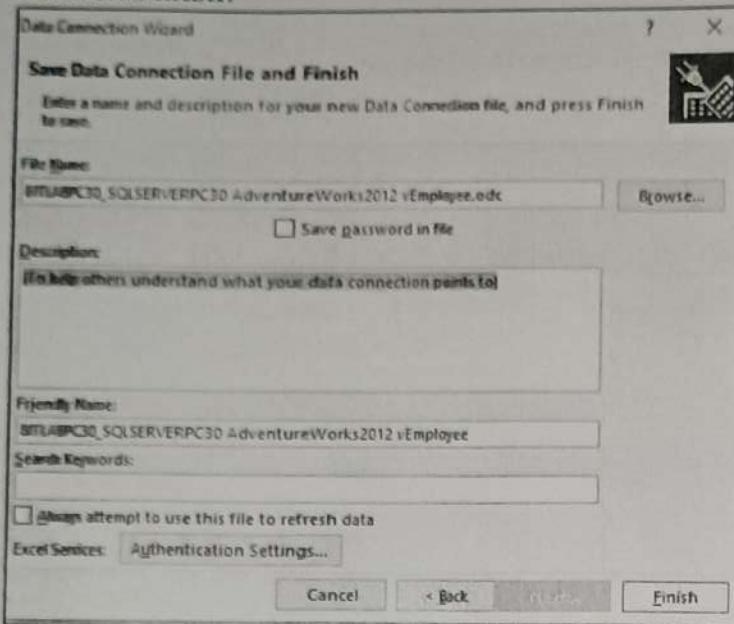
3. Connect to Database Server with AdventureWorks2012 already restored.
Login : sa
Password : sms@123



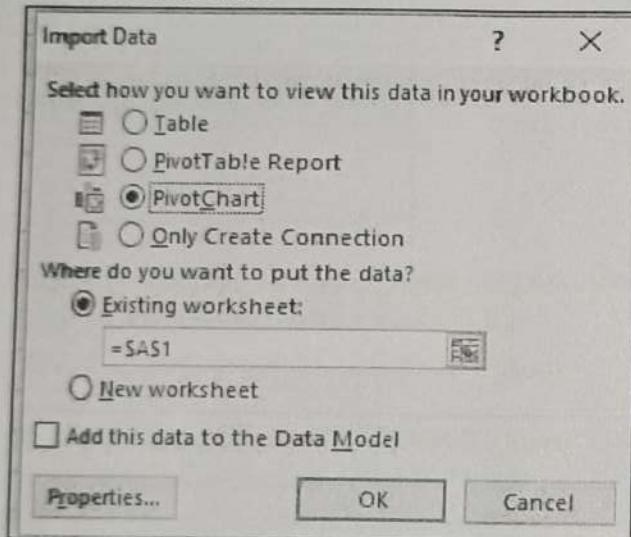
4. Select AdventureWorks2012.



5. Click on Finish.



6. Select PivotChart



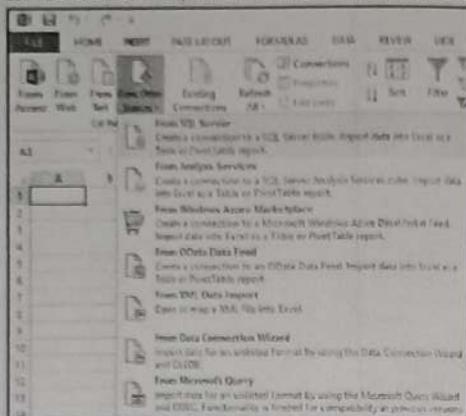
7. Drag and drop FirstName in Category and Sum of Emails in Values.

The screenshot shows a Microsoft Excel spreadsheet titled 'PivotTable - Excel'. The data is from the 'vEmployee' table. A PivotTable is being created with 'FirstName' in the 'Category' field and 'Sum of Email' in the 'Values' field. The PivotTable Fields pane on the right lists various fields: 'AddressLine1', 'BusinessEntityID', 'City', 'CountryRegionCode', 'DivisionID', 'EmailPromotion', 'FirstName', 'Gender', 'HomePhone', 'JobTitle', 'LastName', 'MiddleName', 'NationalIDNumber', 'Phone', 'Picture', 'StateProvinceID', 'Title', and 'TitleOfCourtesy'. The 'Gender' field is currently selected. The PivotTable is located at the bottom left of the sheet area.

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

Steps to follow :

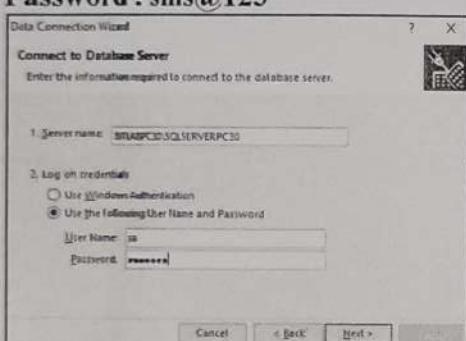
1. Open Excel.
2. In Data ribbon, click on From Other Sources. → Select From SQL Server.



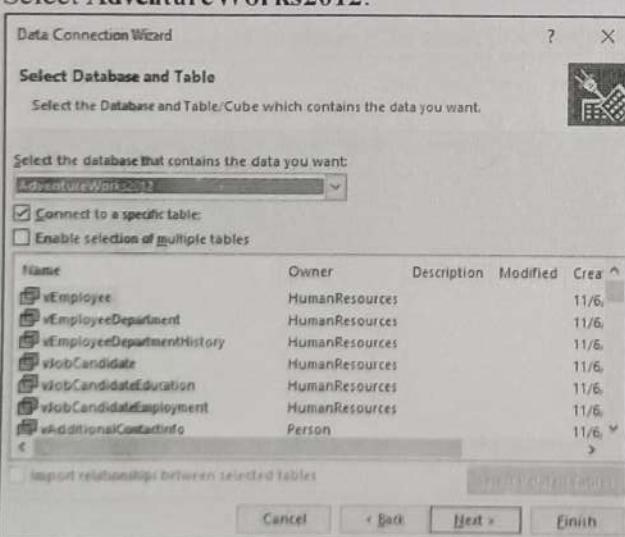
3. Connect to Database Server with AdventureWorks2012 already restored.

Login : sa

Password : sms@123

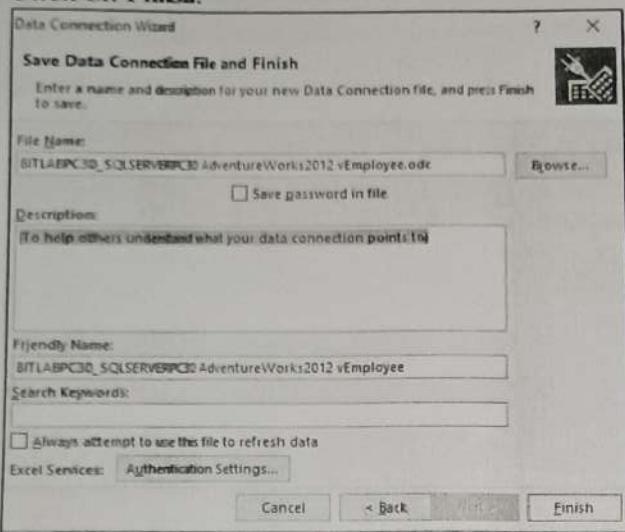


4. Select AdventureWorks2012.

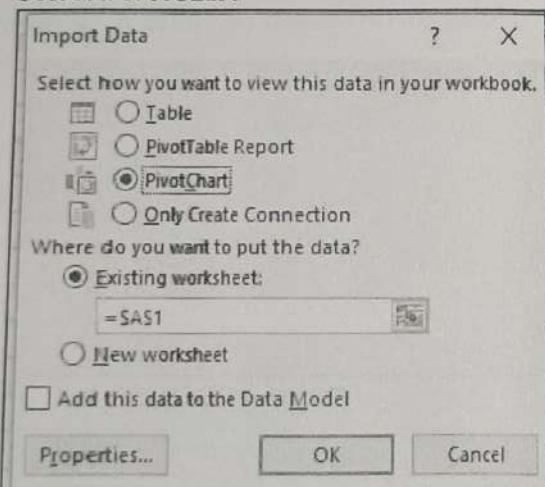


Business Intelligence Practical Journal

5. Click on Finish.



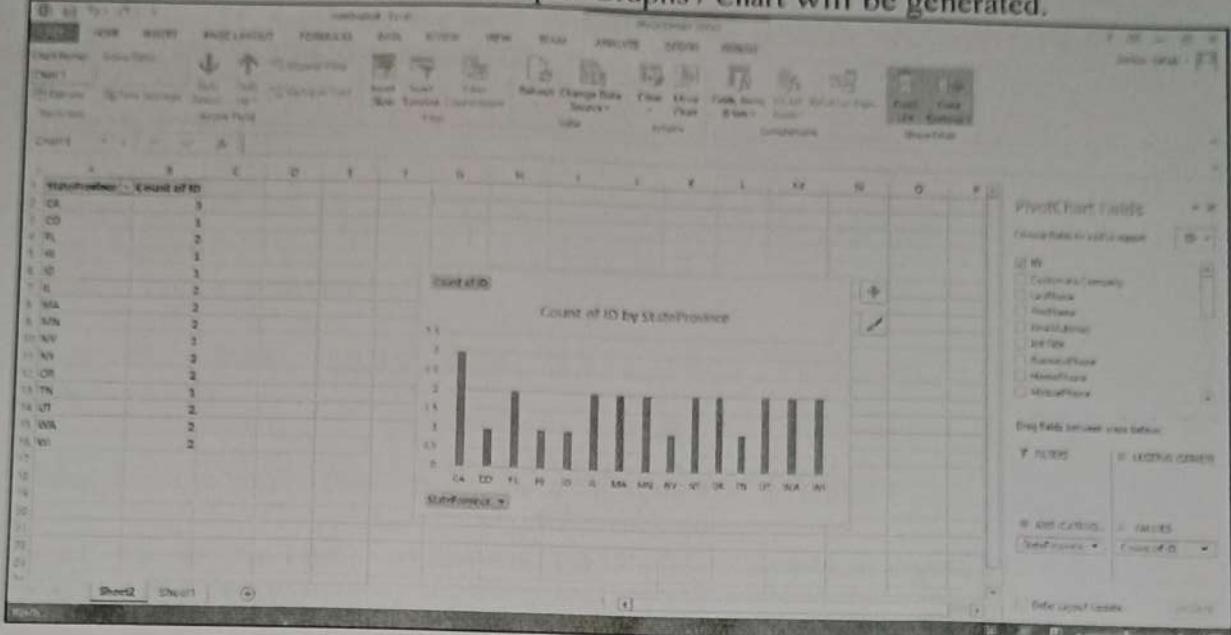
6. Select PivotChart



7. Drag and drop FirstName in Category and Sum of Emails in Values.



8. Select **Clustered Column**. Two Output Graphs / Chart will be generated.



Practical No. 6

Apply the What – if Analysis for data visualization. Design and generate necessary reports based on the datawarehouse data.

(A) USING R STUDIO

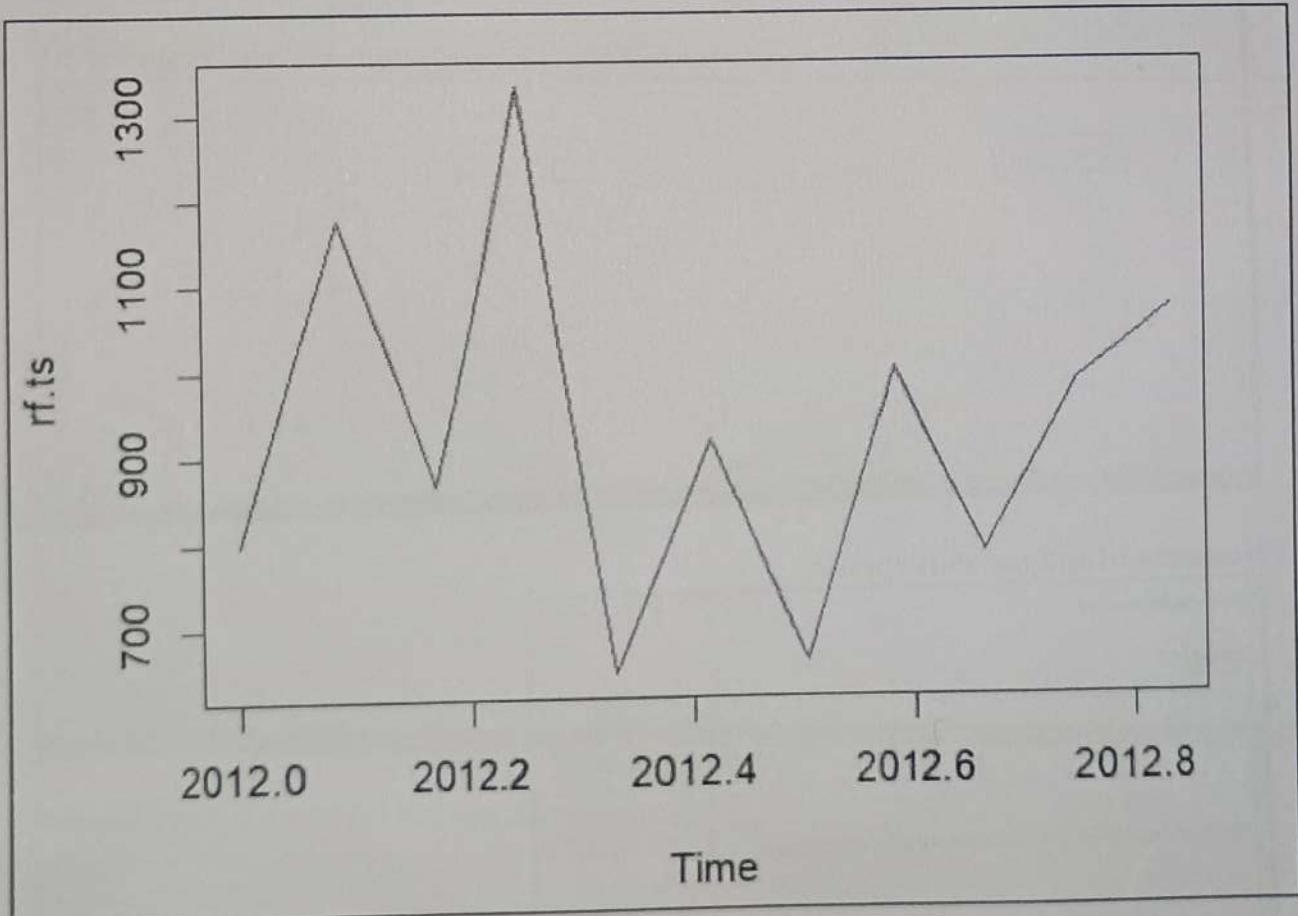
Steps to follow :

1. Open R Studio.
2. Create a vector named rf

```
> rf <- c(799, 1176, 865, 1334, 643, 917, 658, 998, 784,  
985, 1071)
```

3. Using the time series function combine the data and then plot.

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



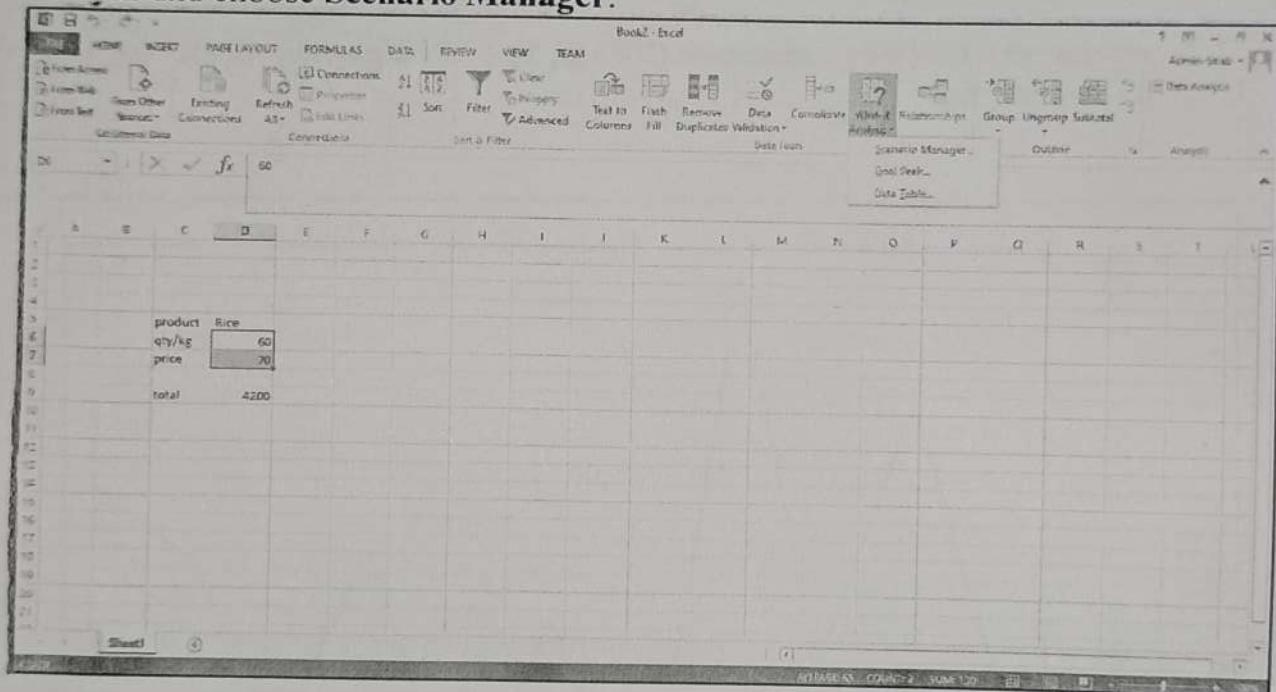
(B) USING SCENARIO MANAGER IN EXCEL

Steps to follow :

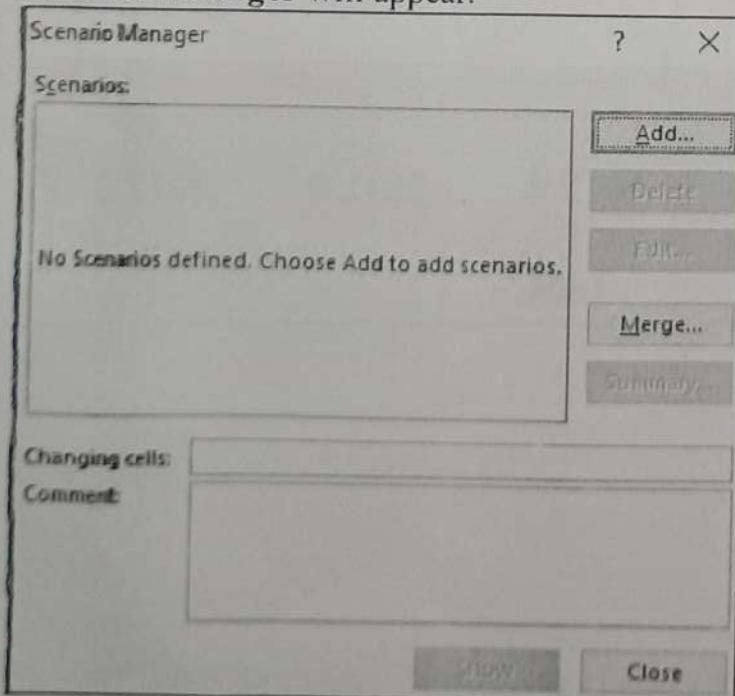
1. Open Excel.
2. Create the data in excel sheet as follows

product	Rice
qty/kg	60
price	70
total	4200

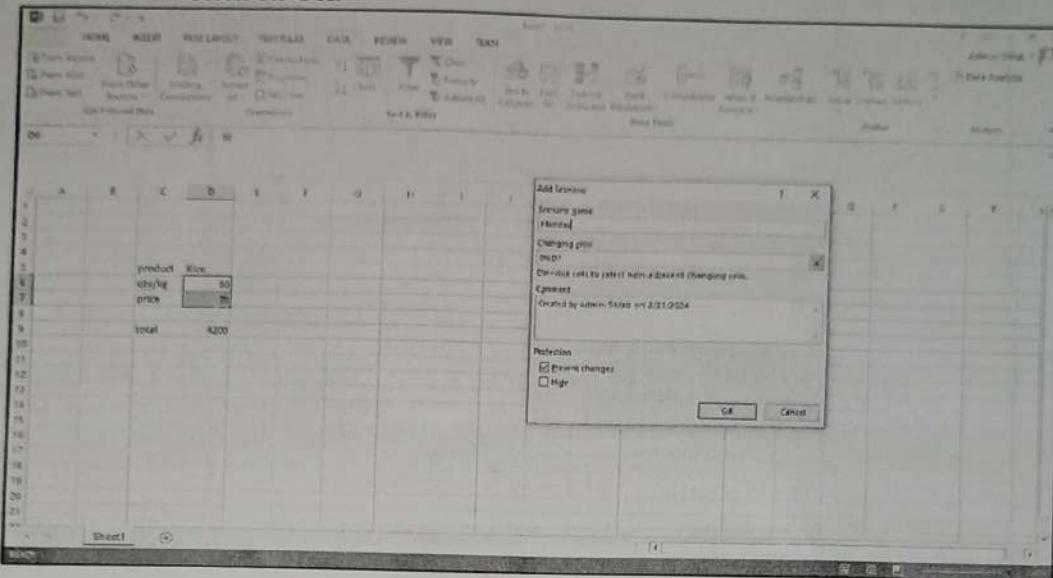
3. Choose the cells of **Quantity** and **Price**. Then in the **Data** ribbon, go to **What-if Analysis** and choose **Scenario Manager**.



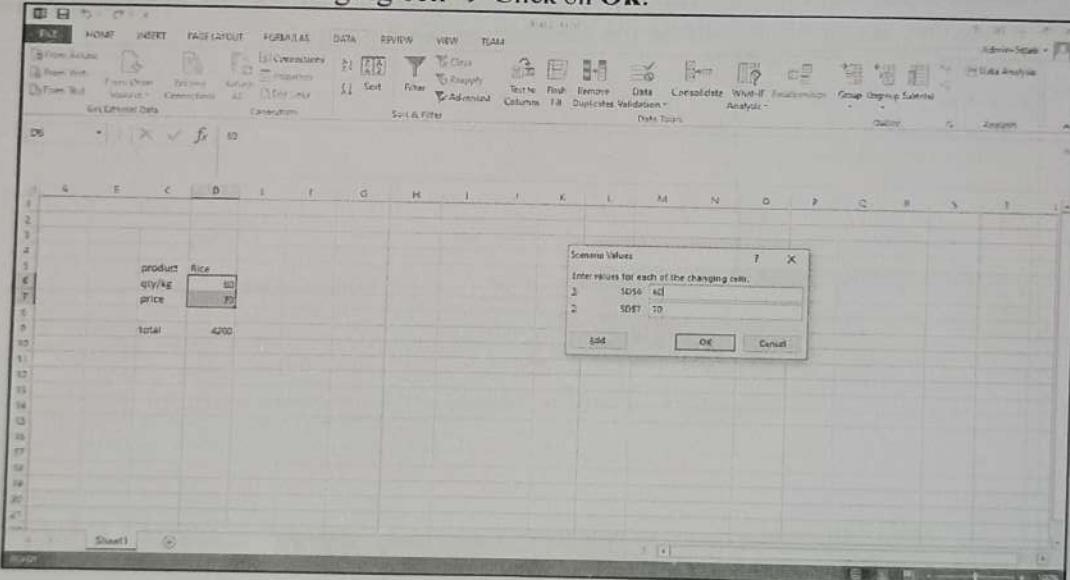
4. Scenario Manager will appear.



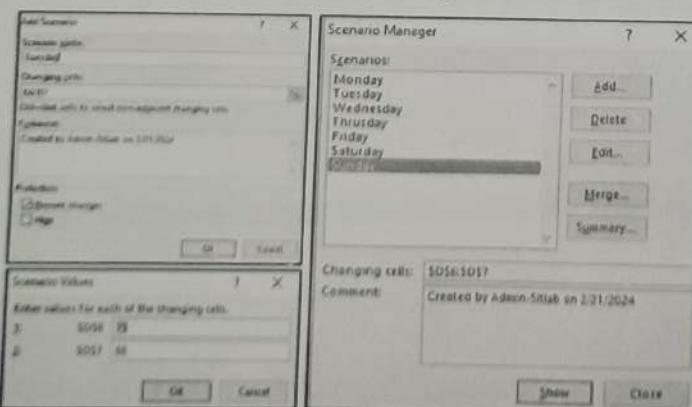
- Click on Add Button → Add Scenario window will appear → In the Scenario name add a unique name such as 'Monday'. Ensure the changing cells are correctly chosen as D6:D7 → Click on Ok.



- Now enter value for changing cell → Click on Ok.

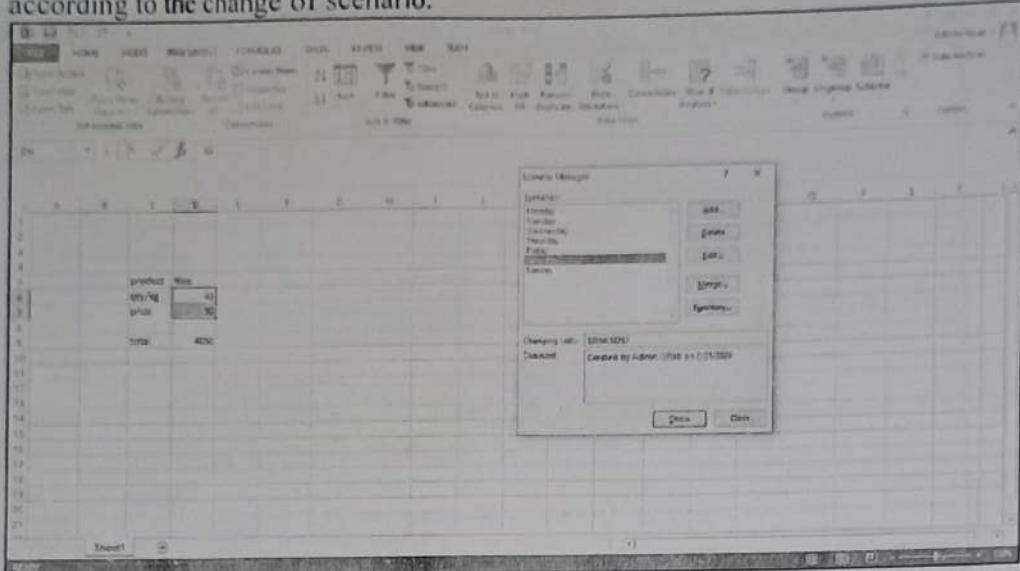


- Repeat steps 5 and 6 for all the days of week.

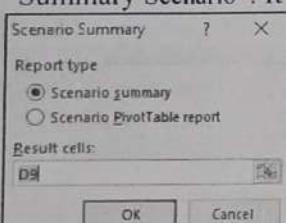


Business Intelligence Practical Journal

8. Click on any day of the week → Click on Show. Values of chosen cell will change according to the change of scenario.



9. Click on **Summary** Button. → Summary Scenario window will appear. → Choose the correct result cell for the summary. On clicking **Ok** a file will be produced named "Summary Scenario". It will be the summary of the data given.



	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Changing Cells	42	60	75	82	95	108	120
Result Cells	4056	4200	5100	4664	4225	5203	4050

Notes: Current Value column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in grey.

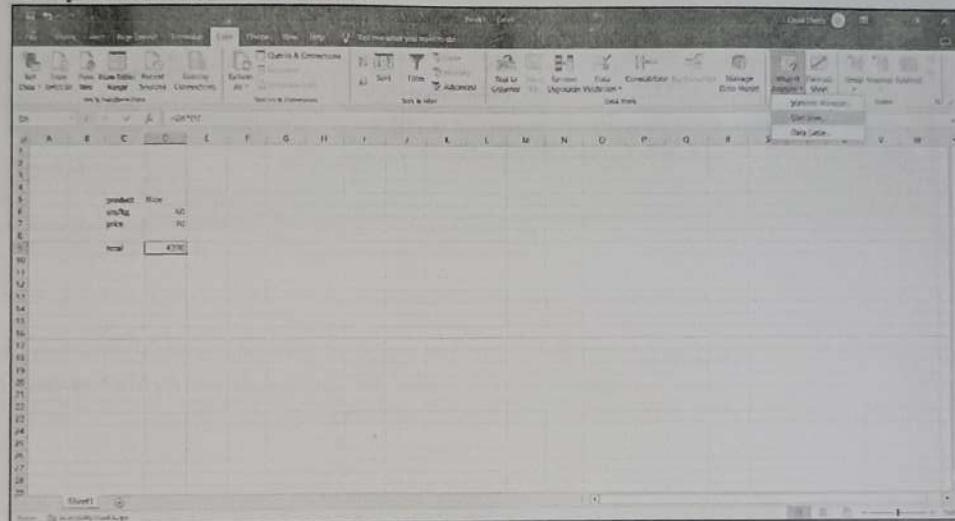
(C) USING GOAL SEEK ANALYSIS IN EXCEL

Steps to follow :

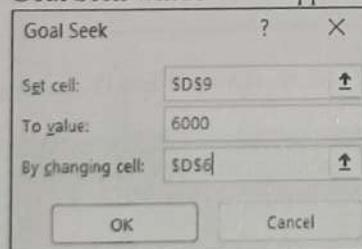
1. Open Excel.
2. Create the data in excel sheet as follows

product	Rice
qty/kg	60
price	70
total	4200

3. Choose the cells of **Quantity** and **Price**. Then in the **Data** ribbon, go to **What-if Analysis** and choose **Goal Seek**.



4. Goal Seek window will appear. → Update values accordingly



5. Click on Ok.

Goal Seek Status

Goal Seeking with Cell D9 found a solution.

Target value: 6000
Current value: 6000

Step Pause

OK Cancel

product	Rice
qty/kg	85.71429
price	70
total	6000

6. Repeat Step 4 with according to the following values.

The first screenshot shows the 'Goal Seek' dialog with the following settings:

- Set cell: \$D\$9
- To value: 6000
- By changing cell: \$D\$7

The second screenshot shows the 'Goal Seek Status' dialog indicating a solution found:

- Goal Seeking with Cell D9 found a solution.
- Target value: 6000
- Current value: 6000

The third screenshot is a summary table:

product	Rice
qty/kg	60
price	100
total	6000

7. Now update the following values while using the equal symbol (=) to create relation.

A	B	C	D	E	F	G	H	I
1								
2								
3								
4								
5		product	Rice			qty/kg		
6		qty/kg		60			=d9	
7		price		70				
8								
9		total		4200				
10								
11								

A	B	C	D	E	F	G	H	I	J	K
1										
2										
3										
4										
5		product	Rice			qty/kg				
6		qty/kg		60			4200			
7		price		70		80				
8						90				
9		total		4200		100				
10						110				
11						120				
12						130				
13						140				
14										

8. Now select the newly created data. → Go to Data ribbon → Go to What-If Analysis → Choose Data Table.

The screenshot shows the Microsoft Excel ribbon with the 'Data' tab selected. In the 'Data Tools' group, the 'What-If Analysis' button is highlighted. The main worksheet area displays the data from the previous tables, with the range A5:D9 selected.

9. Enter the column value as D6 → Click on Ok → Data will be generated in the newly formed table.

The screenshot shows a Microsoft Excel spreadsheet with a Data Table dialog box open. The dialog box has 'Data Table' selected, 'Row input cell' set to '\$D\$6', and 'Column input cell' set to '\$B\$3'. The main spreadsheet contains data for Rice products with columns for product, quantity/kg, and price. A generated table is shown on the right with columns labeled 'qty/kg' and 'price' (4200, 5600, 6300, 7000, 7700, 8400, 9100, 9800).

10. Repeat Step 9 but choose D7 in row input.

The screenshot shows a Microsoft Excel spreadsheet with a Data Table dialog box open. The dialog box has 'Data Table' selected, 'Row input cell' set to '\$D\$7', and 'Column input cell' set to '\$B\$3'. The main spreadsheet contains data for Rice products with columns for product, quantity/kg, and price. A generated table is shown on the right with columns labeled 'price' (100, 110, 120, 130, 140) and 'qty/kg' (4200, 6000, 6600, 7200, 7800, 8400).

11. Create new data according to below snippet. → Repeat Step 9 but choose D7 as row input and D6 as column input. → Data will be generated

The screenshot shows a Microsoft Excel spreadsheet with a Data Table dialog box open. The dialog box has 'Data Table' selected, 'Row input cell' set to '\$D\$7', and 'Column input cell' set to '\$B\$3:\$J\$3'. The main spreadsheet contains data for Rice products with columns for product, quantity/kg, and price. A generated table is shown on the right with columns labeled 'qty/kg' (4200, 60, 90, 100, 110, 120, 130, 140, 150) and 'price' (4200, 6000, 6600, 7200, 7800, 8400, 9100, 9800, 10500).

Practical No. 7

Perform the data classification using classification algorithm

Steps to follow :

1. Open RStudio.
2. Install appropriate packages –

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

3. Extract the installed package.

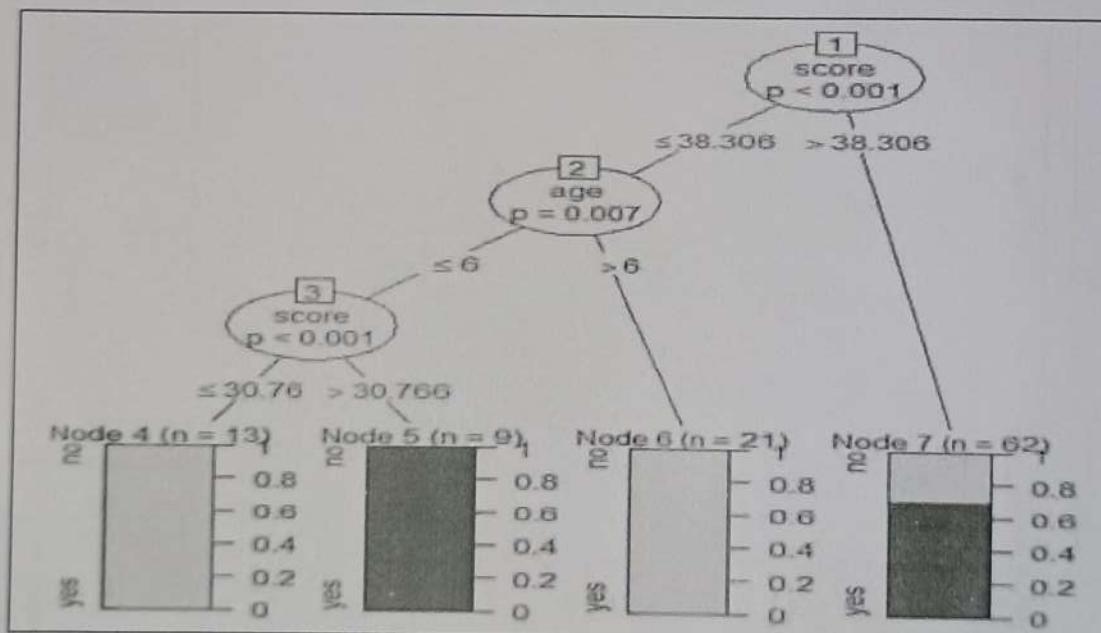
```
> library("party")
```

4. Store the data and then display it in output.

```
> input.data <- readingskills[c(1:105),]
> output.tree <-
ctree(nativeSpeaker~age+shoeSize+score,data=input.data)
```

5. Plot the graph.

```
> plot(output.tree)
```



Practical No. 8

Perform the **data clustering** using clustering algorithm

Steps to follow :

1. Open RStudio.
2. Extract the dataset of iris to perform clustering.

```
> newiris <- iris
```

3. Assign null value to the variable.

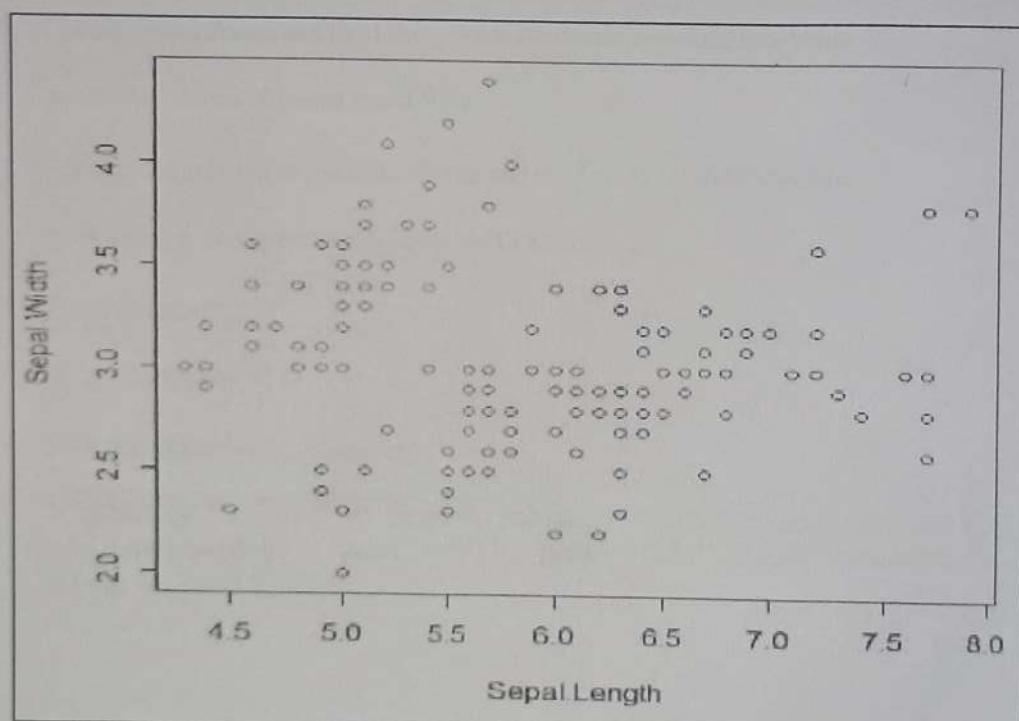
```
> newiris$Species <- NULL
```

4. Perform clustering and define the size of newiris cluster as 3

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

5. Plot the graph with the following rows and columns

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



Practical No. 9

Perform the Linear Regression on the given data warehouse data.

Steps to follow :

1. Open RStudio.
2. 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)
```

3. Create linear regression variable.

```
> relation <- lm(y~x)
```

4. Print the variable.

```
> print(relation)
```

5. Print the summary of the linear relation variable.

```
> print(summary)
```

6. Create a data frame and find the y variable corresponding to x value.

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

7. Predict the relation of the data frame and the linear relation variable.

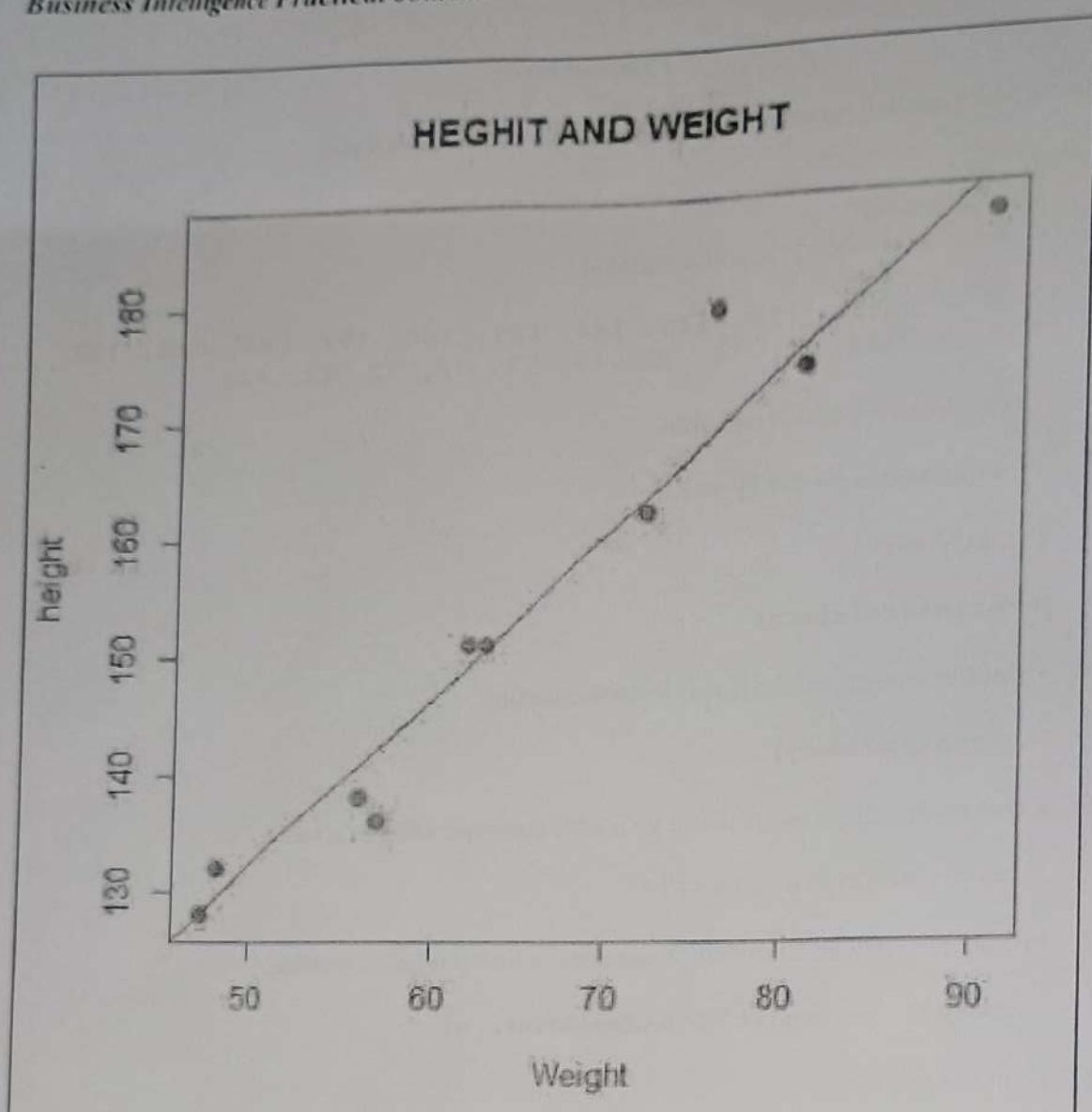
```
> result <- prediction(relation, a)
```

8. Print the result.

```
> print(result)
```

9. Plot the graph for the following.

```
> plot(y, x, col = "red", main = "HEIGHT and WEIGHT",
       abline(lm(x~y)), cex = 1.3, pch = 16, xlab= "Weight",
       ylab = "height")
```



Practical No. 10

Perform Data Analysis using Timeseries.

Steps to follow :

1. Open RStudio.
2. Get the data points for R vector

```
>rainfall <- c(799, 1174.8, 865.1, 1334.6, 635.6, 918.6,  
998.6, 784.2, 985.7, 882.8, 1071)
```

3. Convert to time series object.

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

4. Print the data

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

5. Give file name

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

6. Plot the graph

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

7. Save the file

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

8. Plot the file again if the graph is not visible

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

