

Class : MSC I

Sem : I

Subject : Data Warehousing & Data Mining(BI)

Paper : IV

Academic Year : 2022-23

Roll No : 522

INDEX

S.NO	Date	Topic	Page. No	Sign
1		Creation of Dimensions and Fact tables.		
2		Create Data Source using SSAS (SQL Server Analysis Services.)		
		Create Data Source View using SSAS (SQL Server Analysis Services.)		
4		Create cube using SSAS(SQL Server Analysis Services.) and process the cube.		
5		View cube data in multidimensional Format. (Excel Pivot Chart.)		
6		Working with measures in the cube.		
7		Creating an Excel Pivot Table and Pivot Chart by using the OLAP cube data.		
8		Firing Queries on Tables.		
9		Calculation & KPI		
10		Data pre-processing.		
11		Data discretization.		
12		Classification problems.		
13		Clustering Analysis.		
14		Association Rule Mining.		
15		Data visualization.		

Practical No 1

Aim : Creation of Dimensions and Fact tables.

Solution :

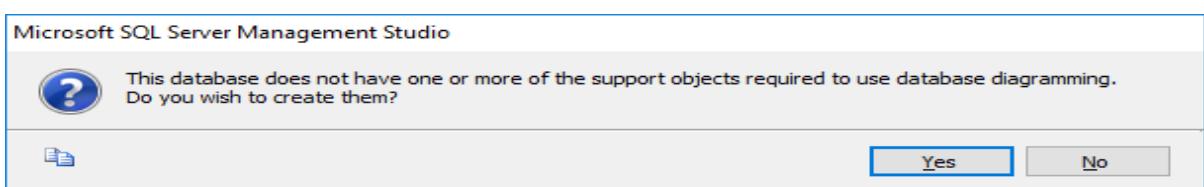
Open Application -> Microsoft SQL Server 2008 R2 -> SQL Server Management Studio

1. Select Connect Tab -> Database Engine -> Select Server Name(local)
2. Right Click the Database -> New Database
3. Types “SalesInformation” as the database name, click on OK to close the dialog box and to create the database.

Create a Database Diagrams

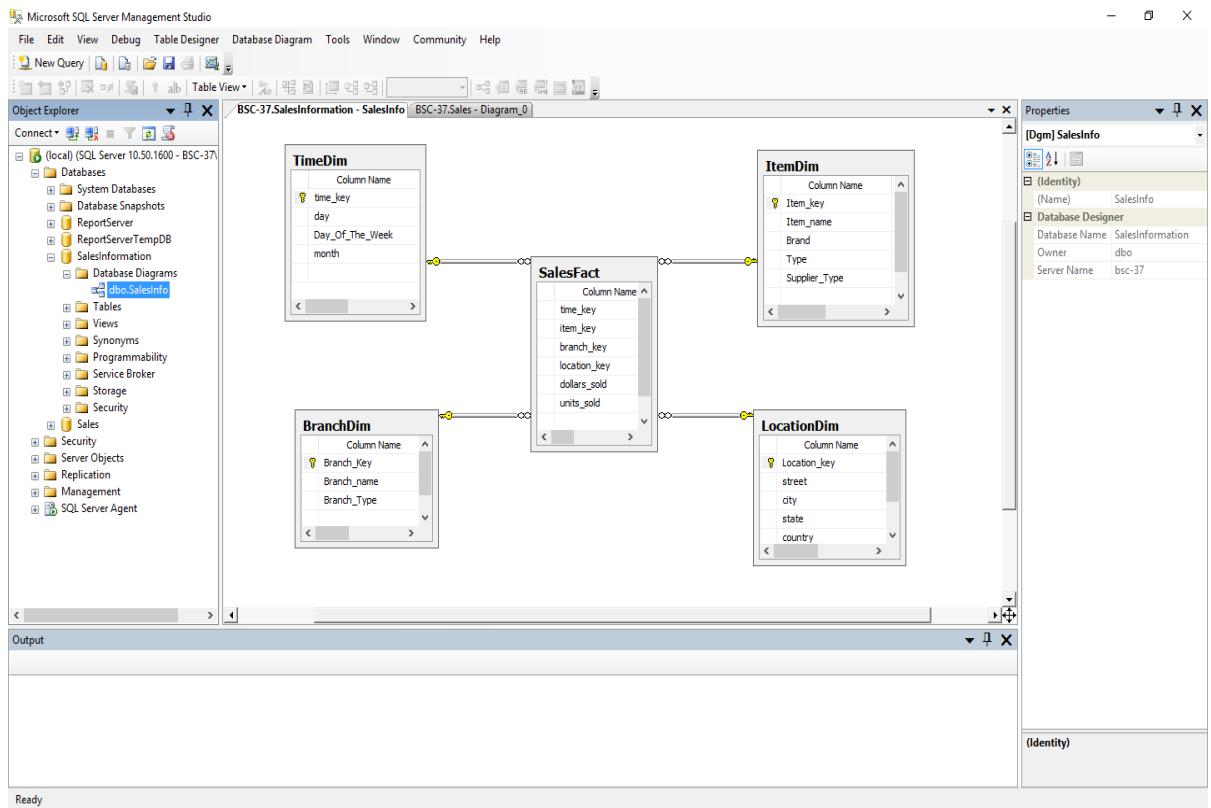
Expand the “SalesInformation” database folder.

1. Click on Database Diagrams to expand it



On click of it, above Dialog box appears, click on Yes to close it.

2. Right Click on Database Diagrams -> New Database Diagrams
3. Create fact and Dimension Tables. (Right click on surface, choose New Table to add tables on Database Diagrams.)



4. Establish relationship between fact and dimension tables.
5. Save Database Diagrams with name as “SalesInfo”. (After saving Database Diagrams fact and dimension tables are automatically placed in Table tab.)

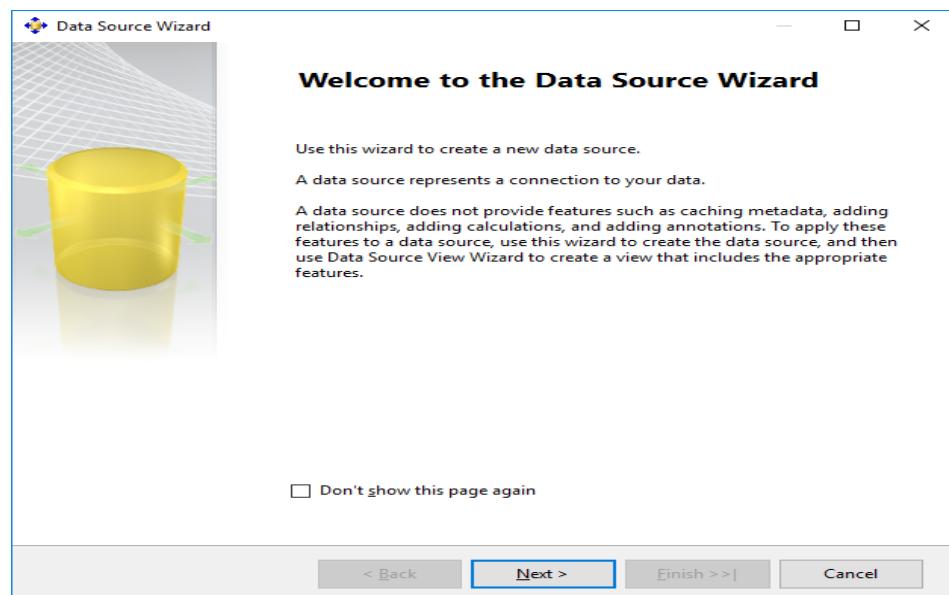
Practical No 2

Aim : Create Data Source using SSAS(SQL Server Analysis Services.)

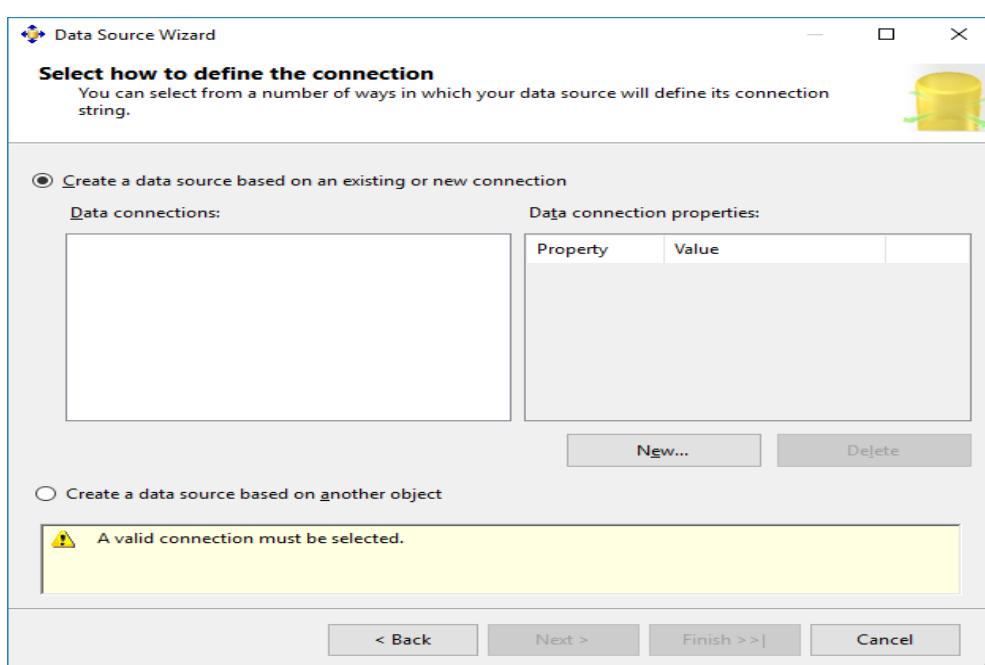
Solution :

Open Application -> Microsoft SQL Server 2008 R2 -> SQL Server Business Intelligence Development Studio

1. Select File -> New Project -> Choose Analysis Service Project -> Name it as "SalesInfo_BIPrj" and click on OK.
2. Right Click on Data Sources -> New Data Source

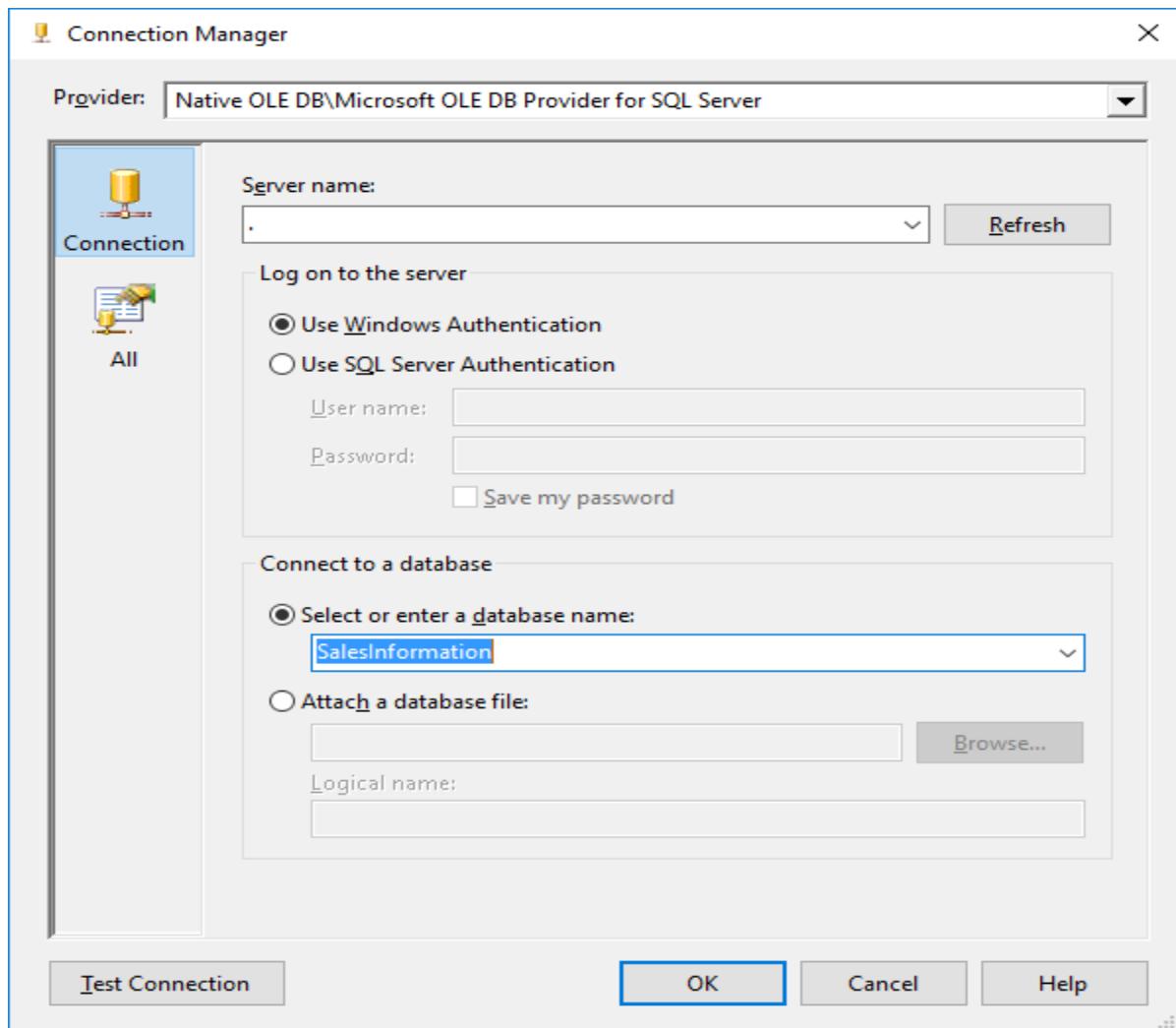


Click on Next.

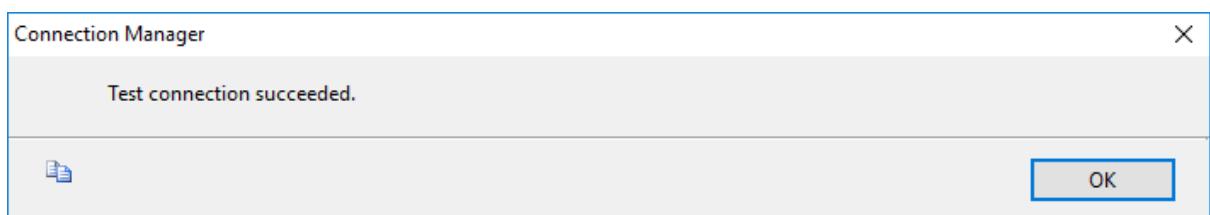


Click on New.

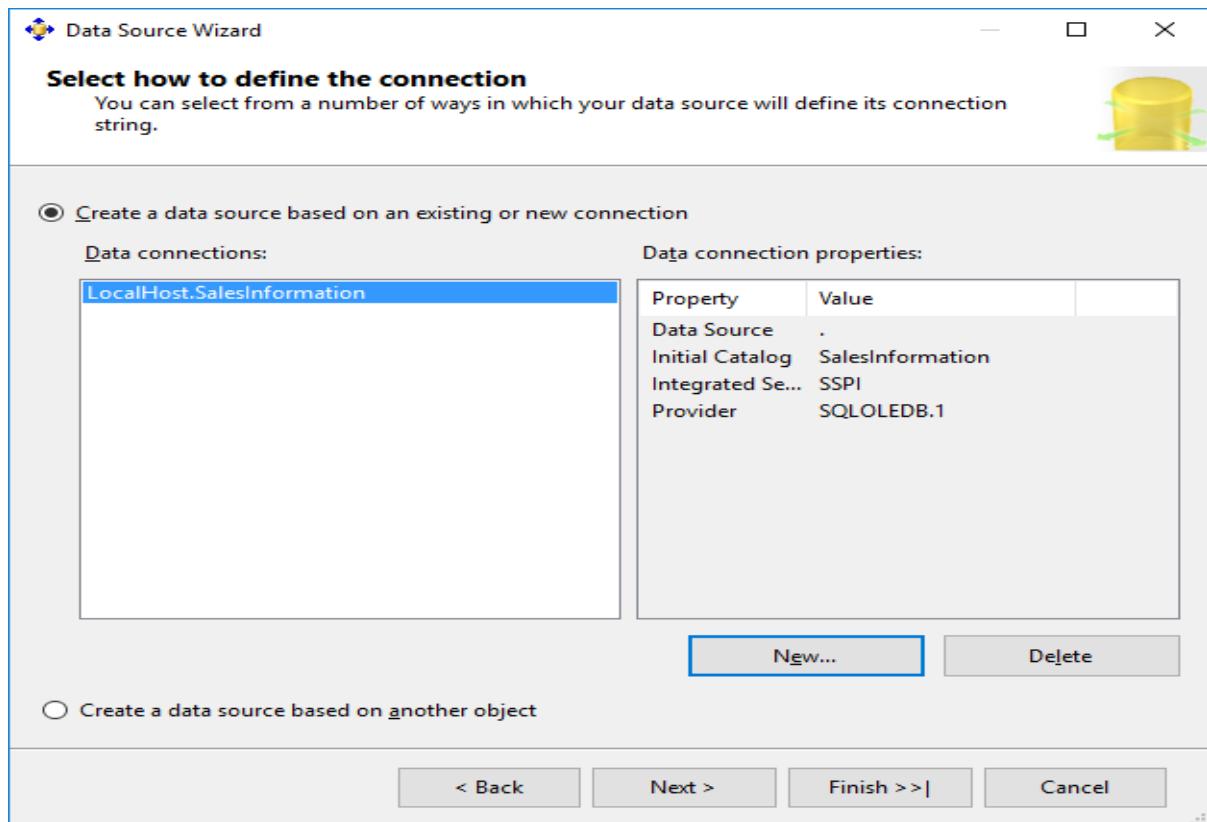
3. Choose Provider as “Microsoft OLEDB Provider for SQL Server”, Server Name as “.”, Select database name as “SalesInformation”.(Created in SQL Server Management studio).



4. Click on Test Connection.

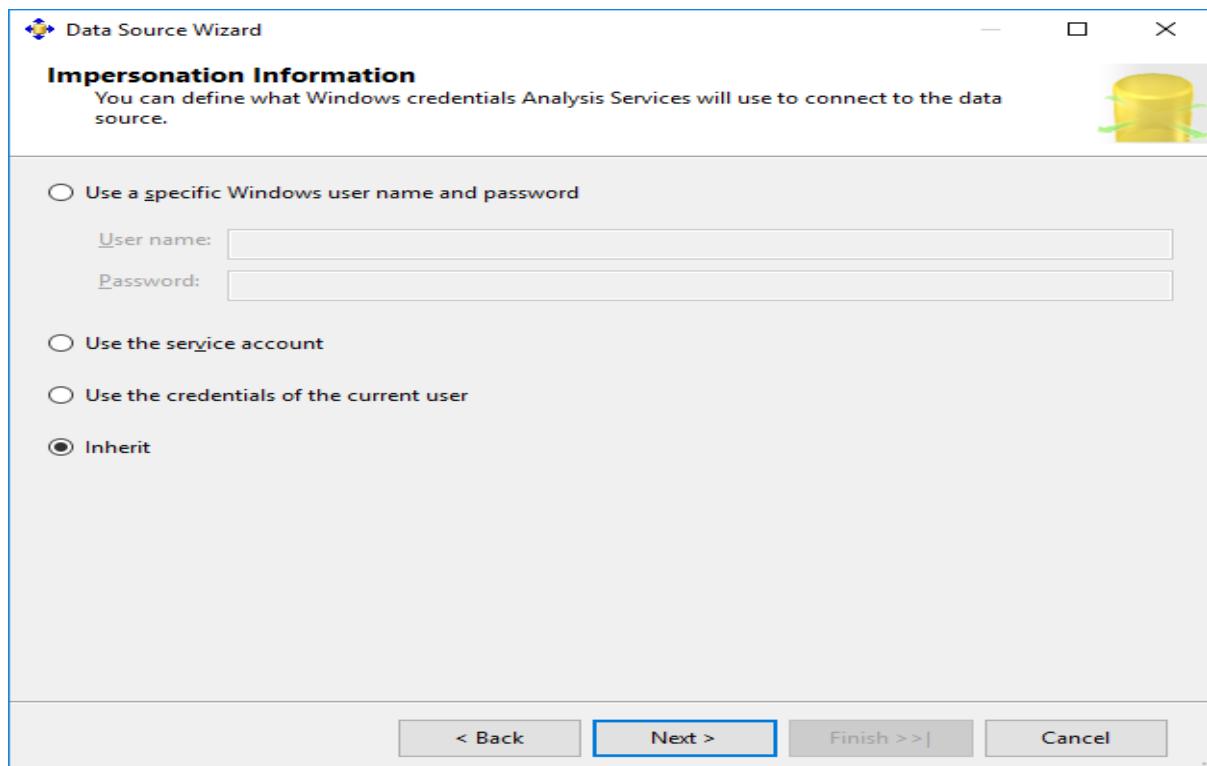


Click on OK.



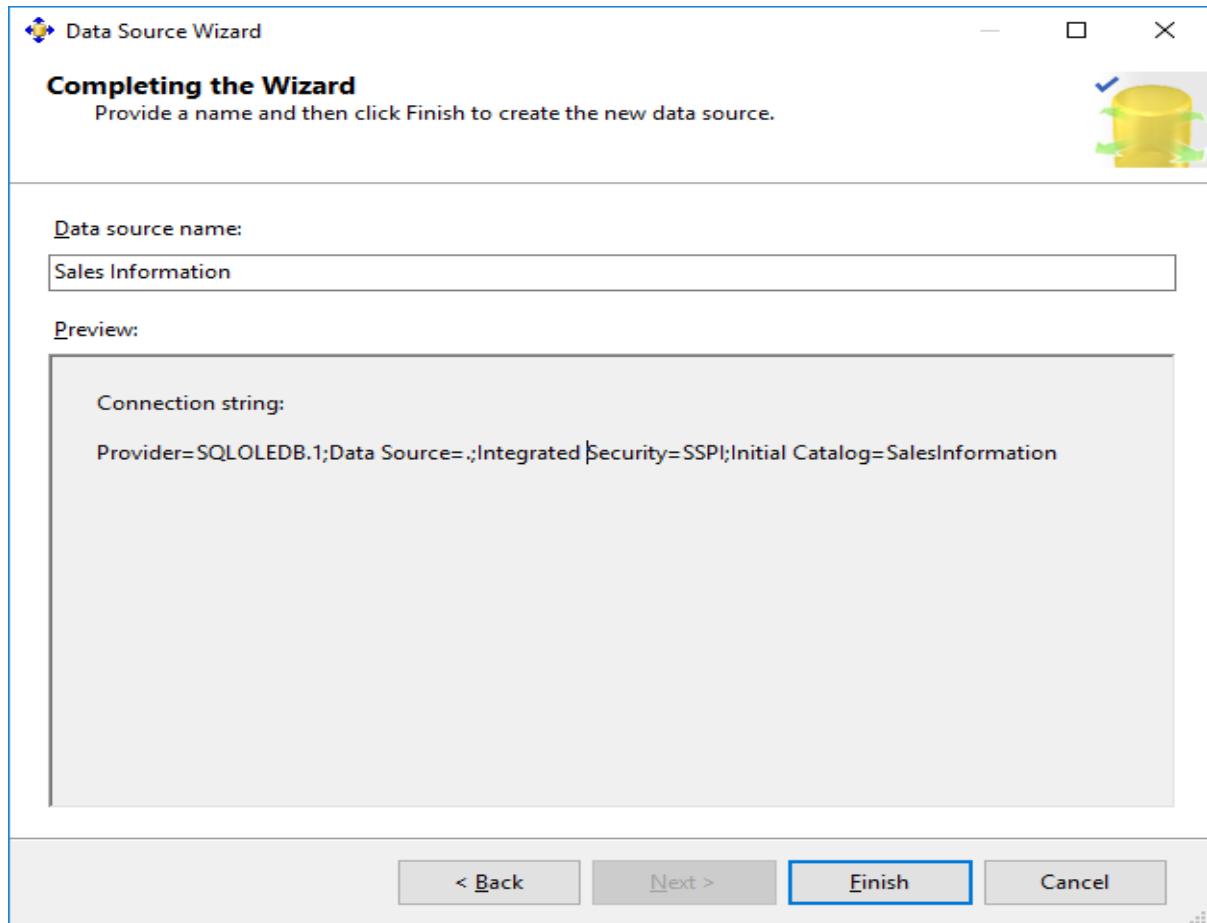
Click on Next

5. Choose “Inherit” option.



Click on Next.

6. Click on Finish.



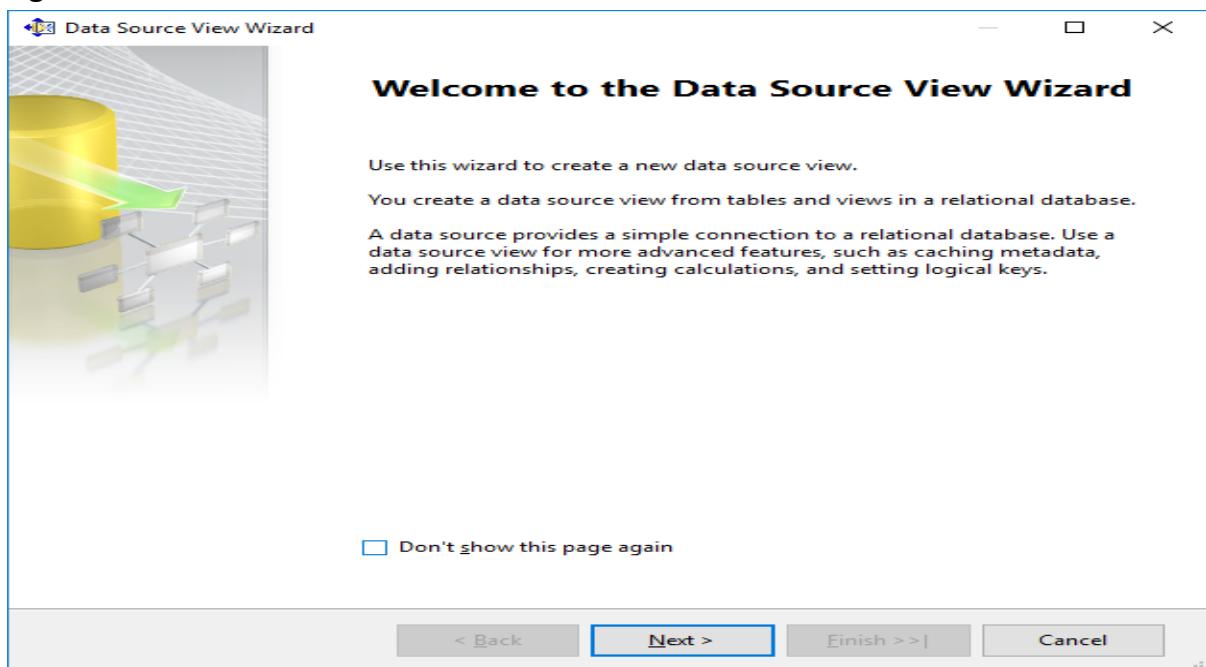
Name Data Source as “Sales Information”.

Practical No 3

Aim : Create Data Source View using SSAS(SQL Server Analysis Services.)

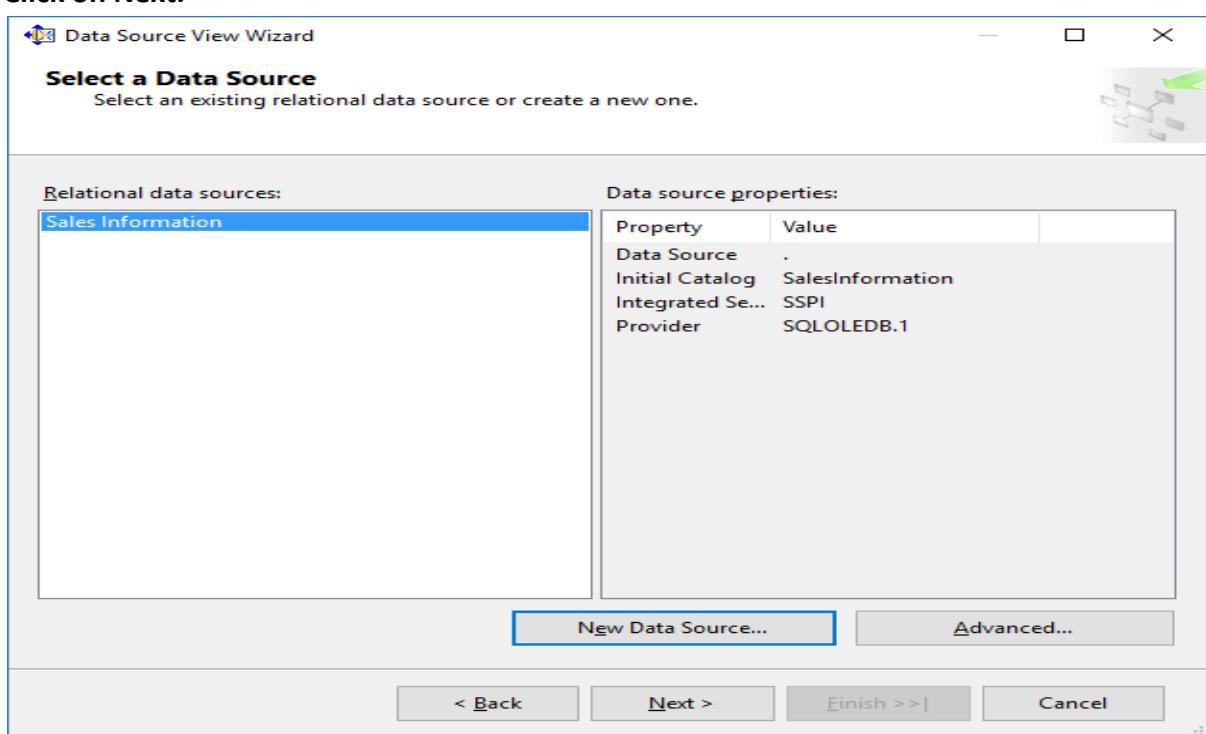
Solution :

1. Right click on Data Source View -> New Data Source View

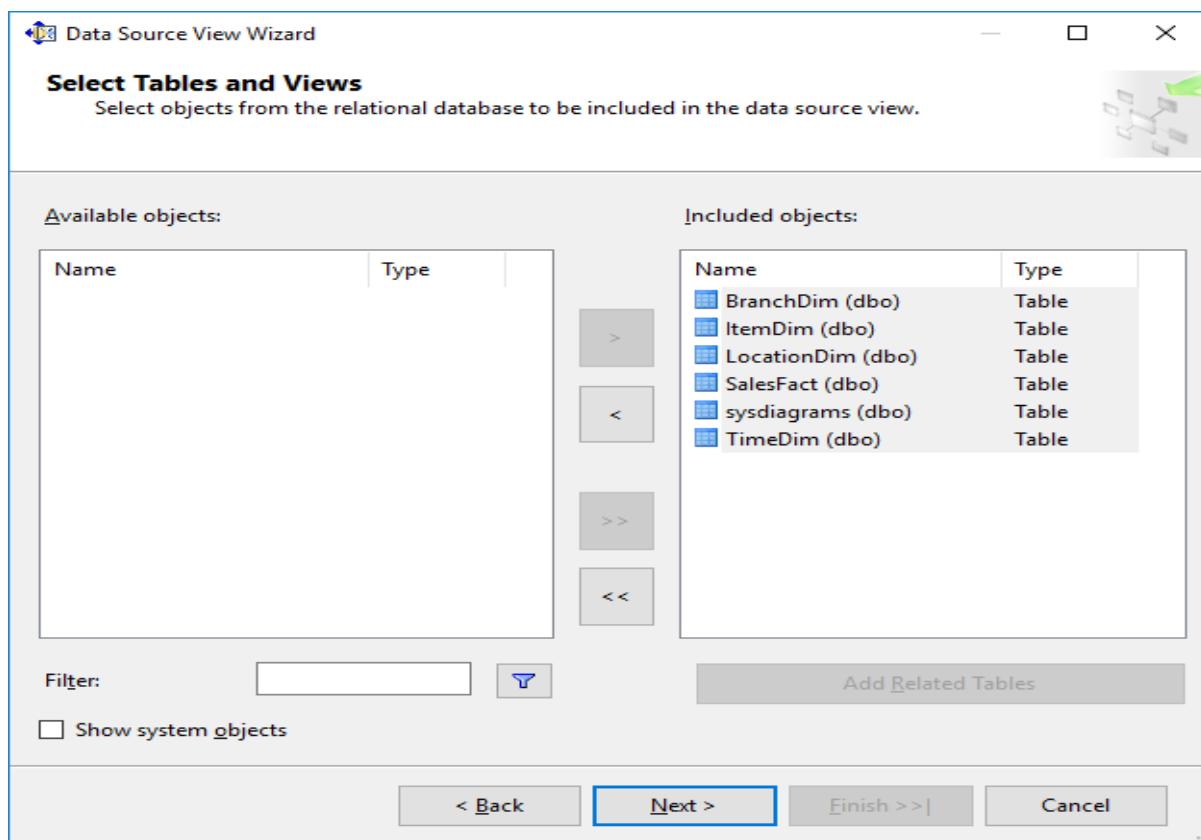
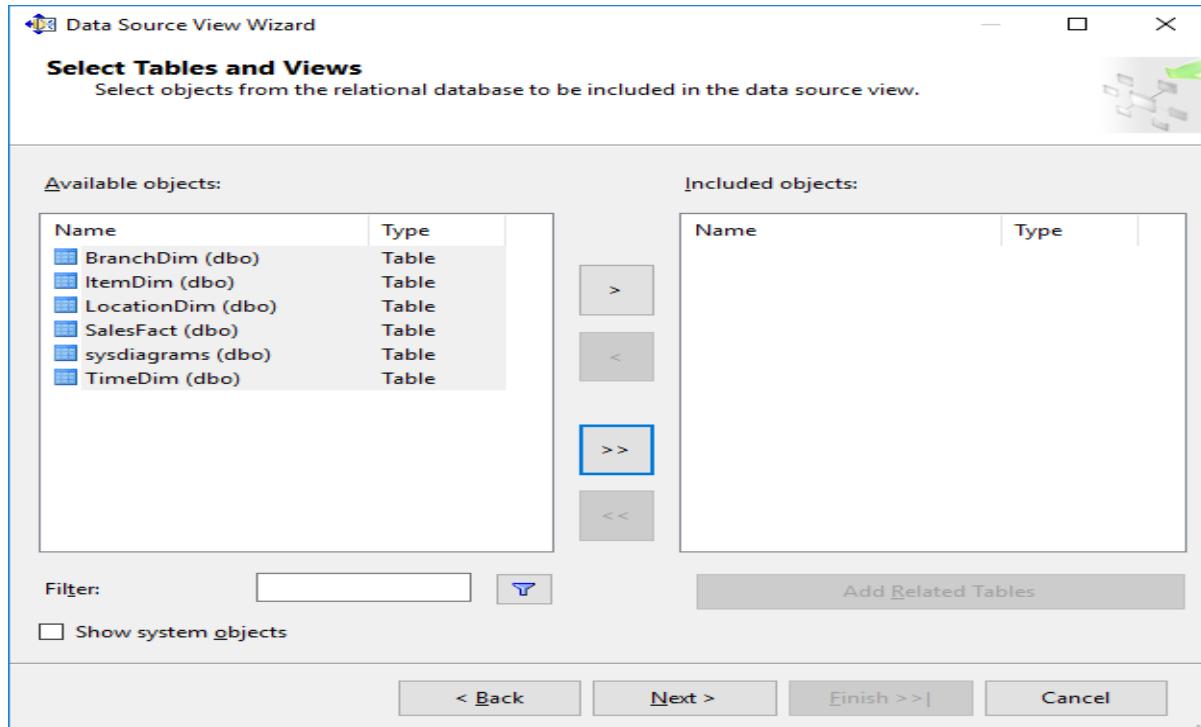


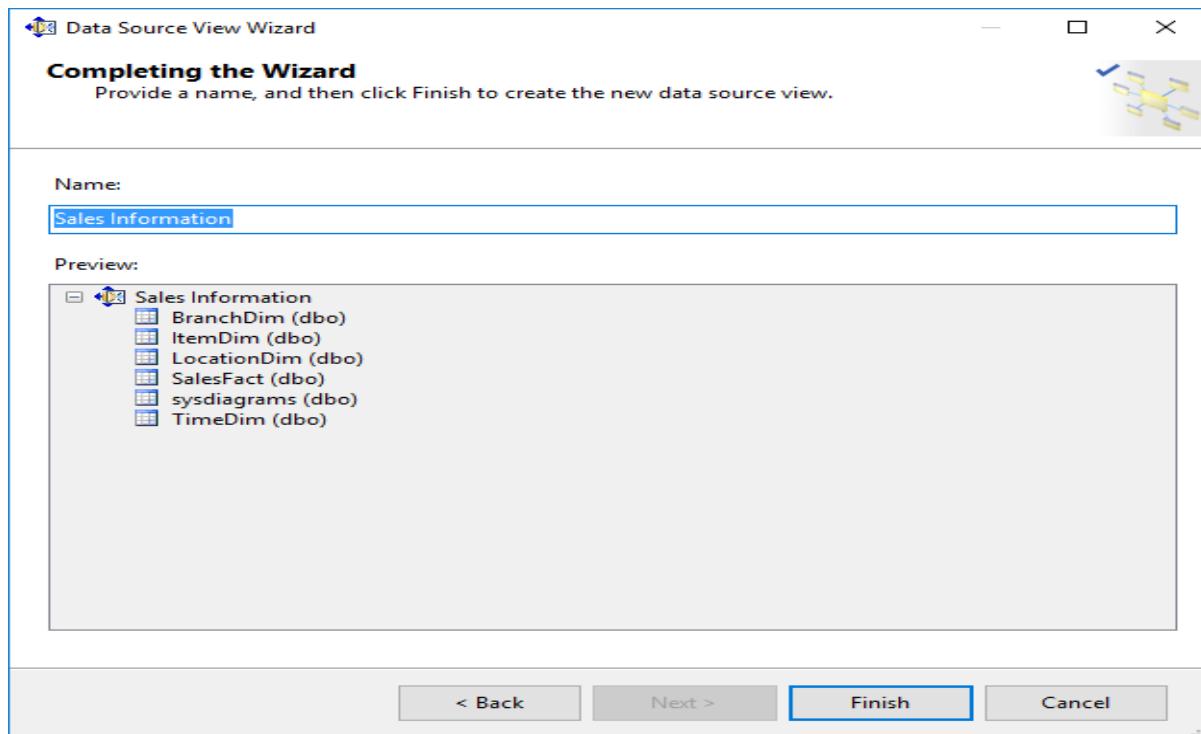
Click on Next.

2. Click on Next.



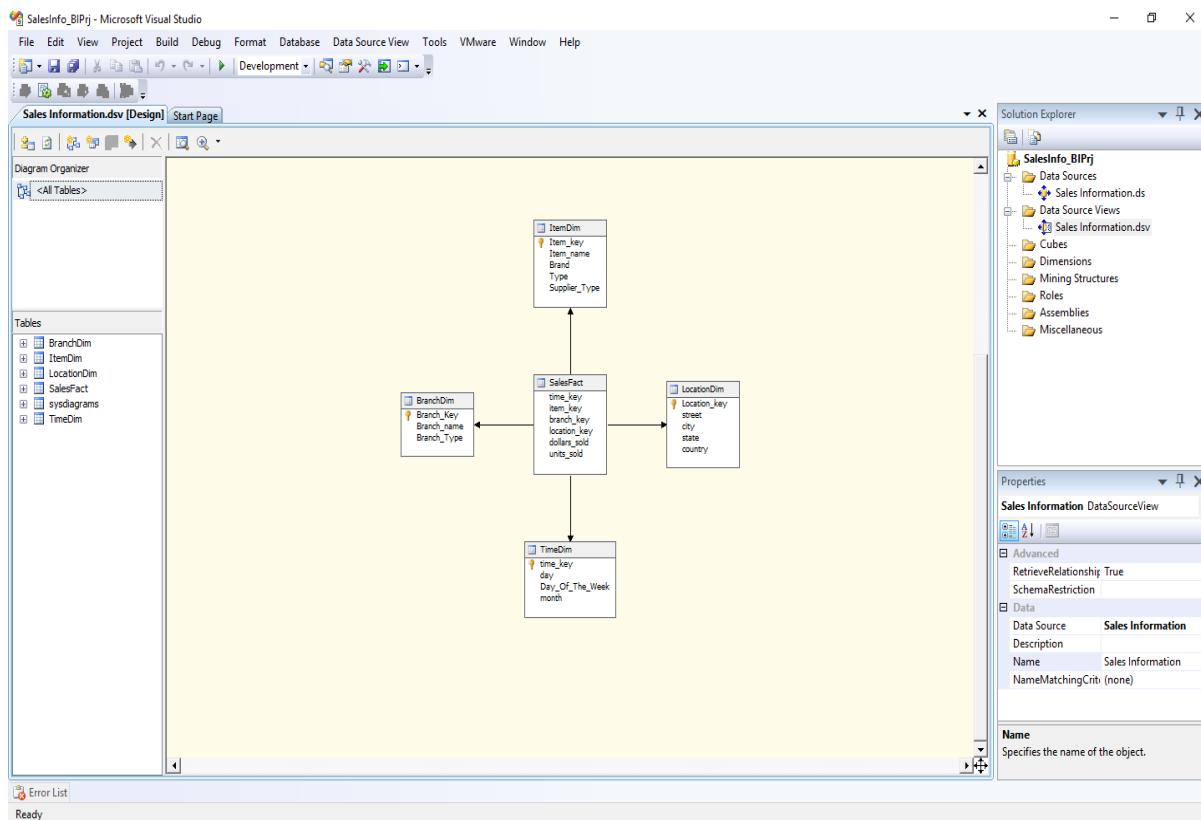
3. Select Tables and Views.





Click on Finish.

4. Finally, we will get the Data Source View like :

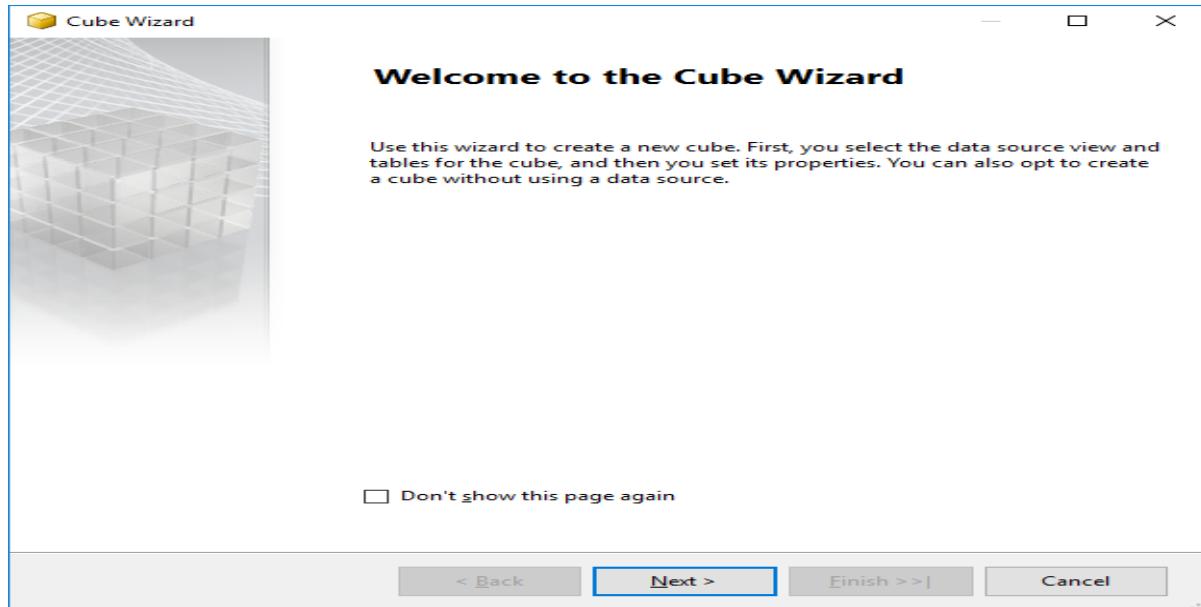


Practical No 4

Aim : Create cube using SSAS(SQL Server Analysis Services.) and process the cube.

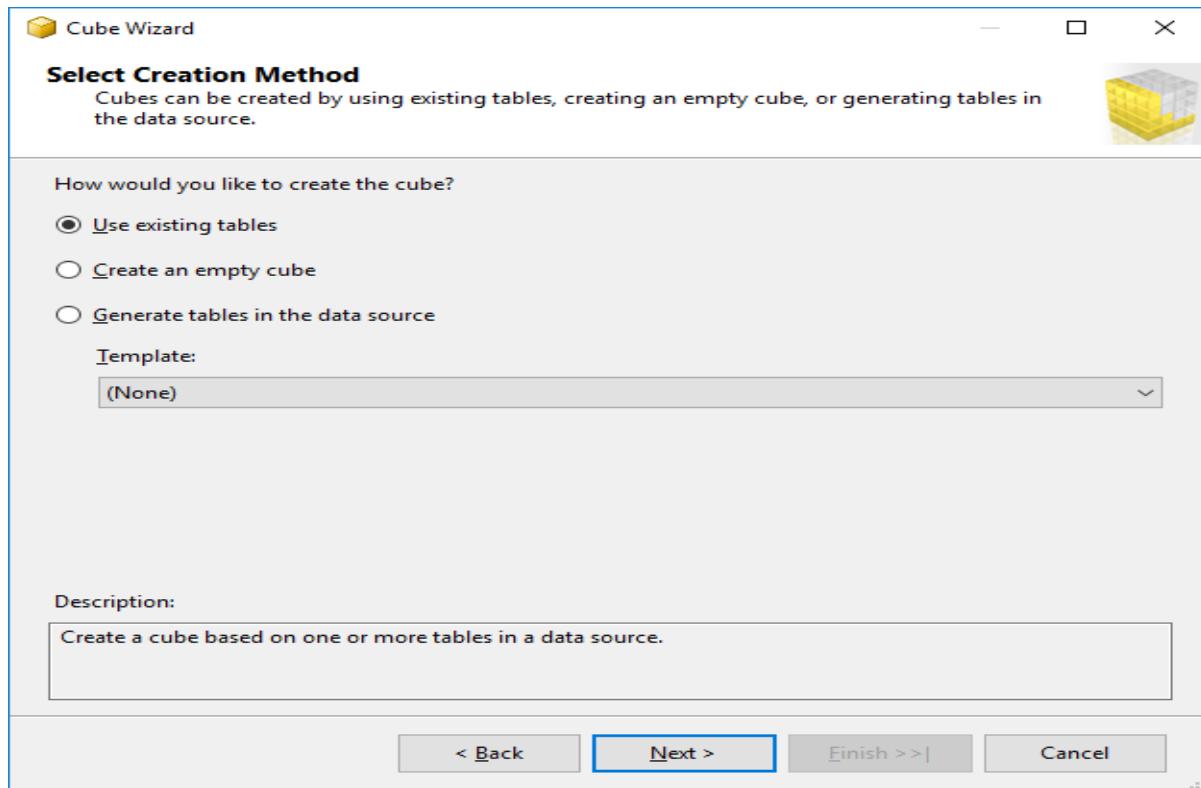
Solution :

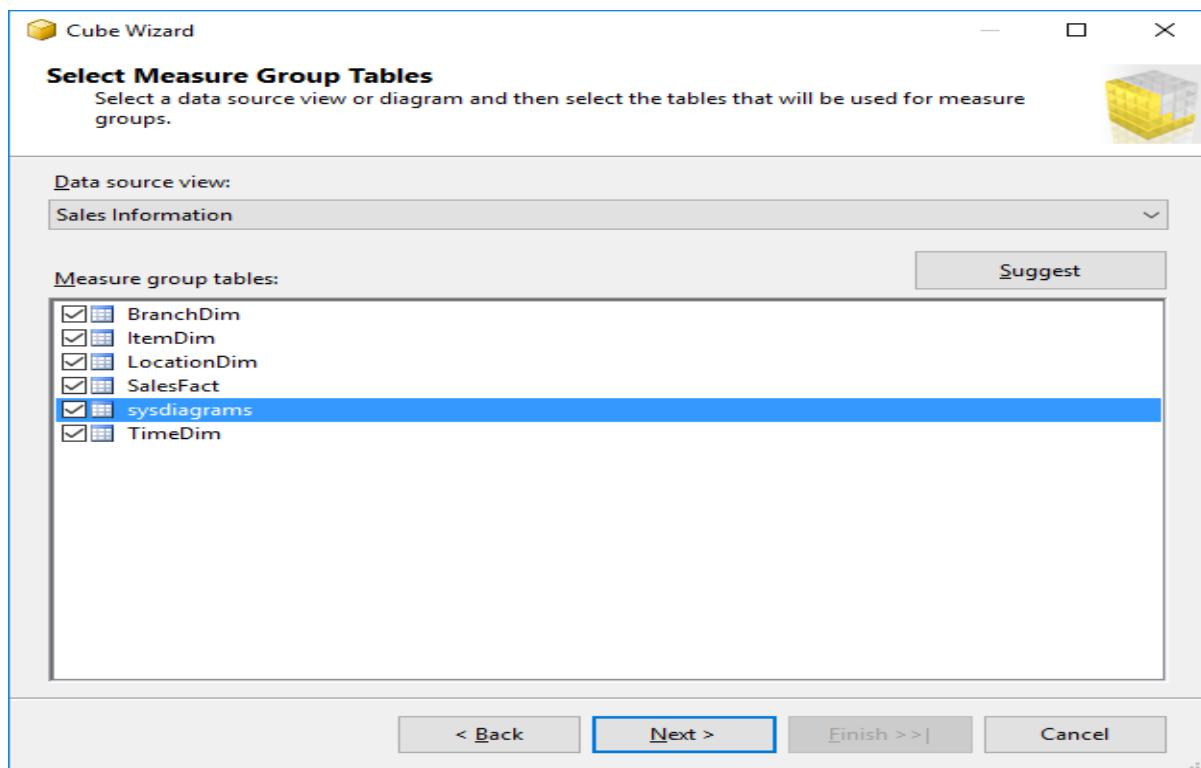
1. Right click on Cubes -> New Cube.



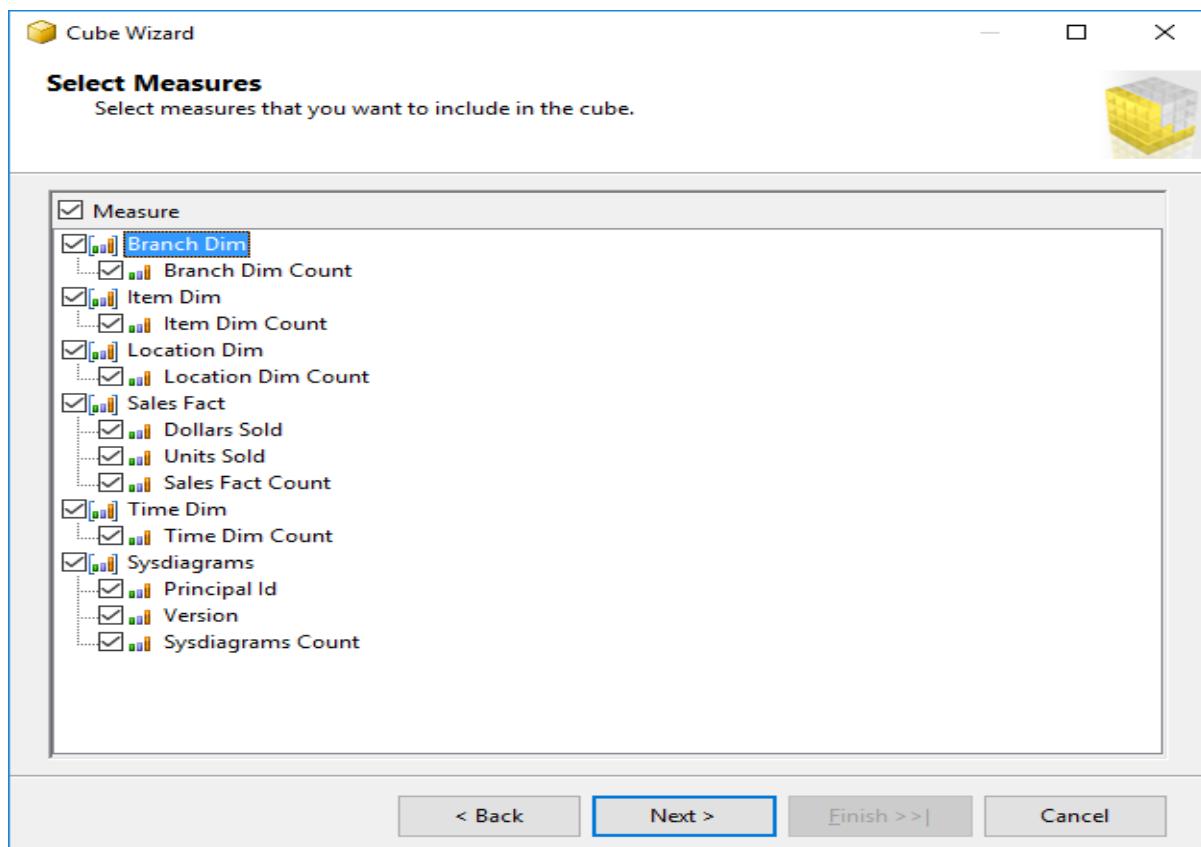
Click on Next.

2. Select First option “Use existing tables”. Click on Next.

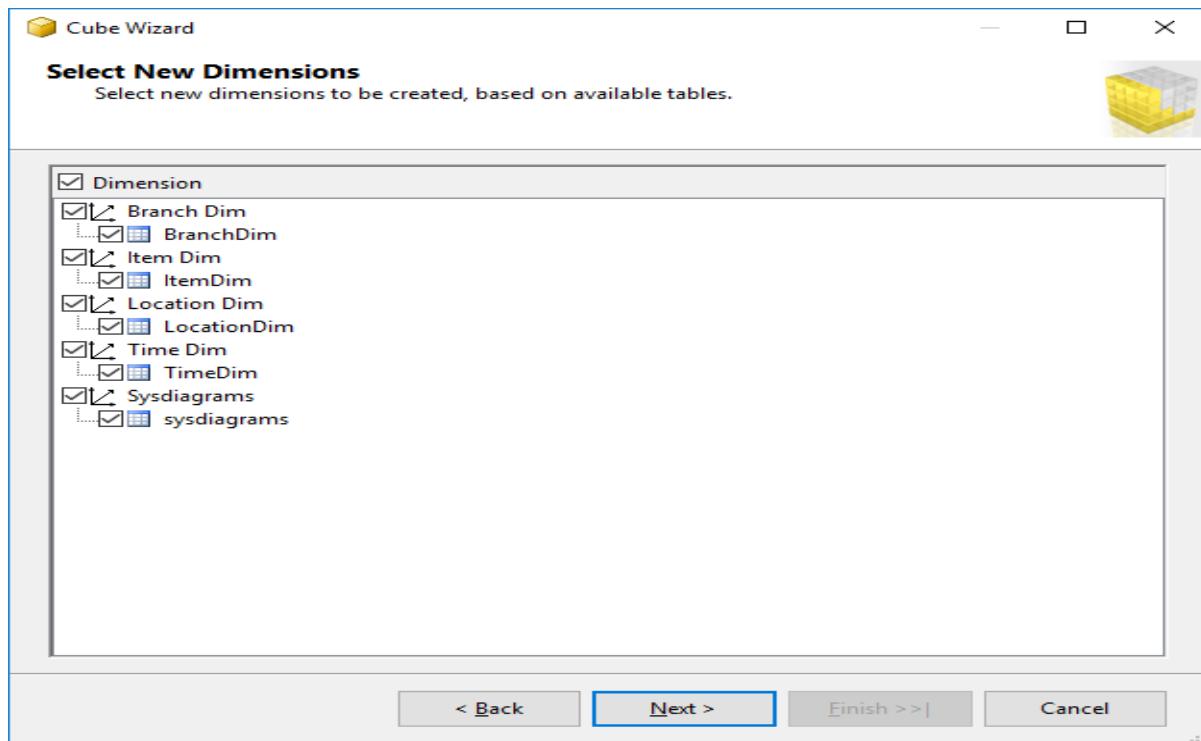


3. Select Data Source View as “Sales Information” and Select all the tables.

Click on Next.

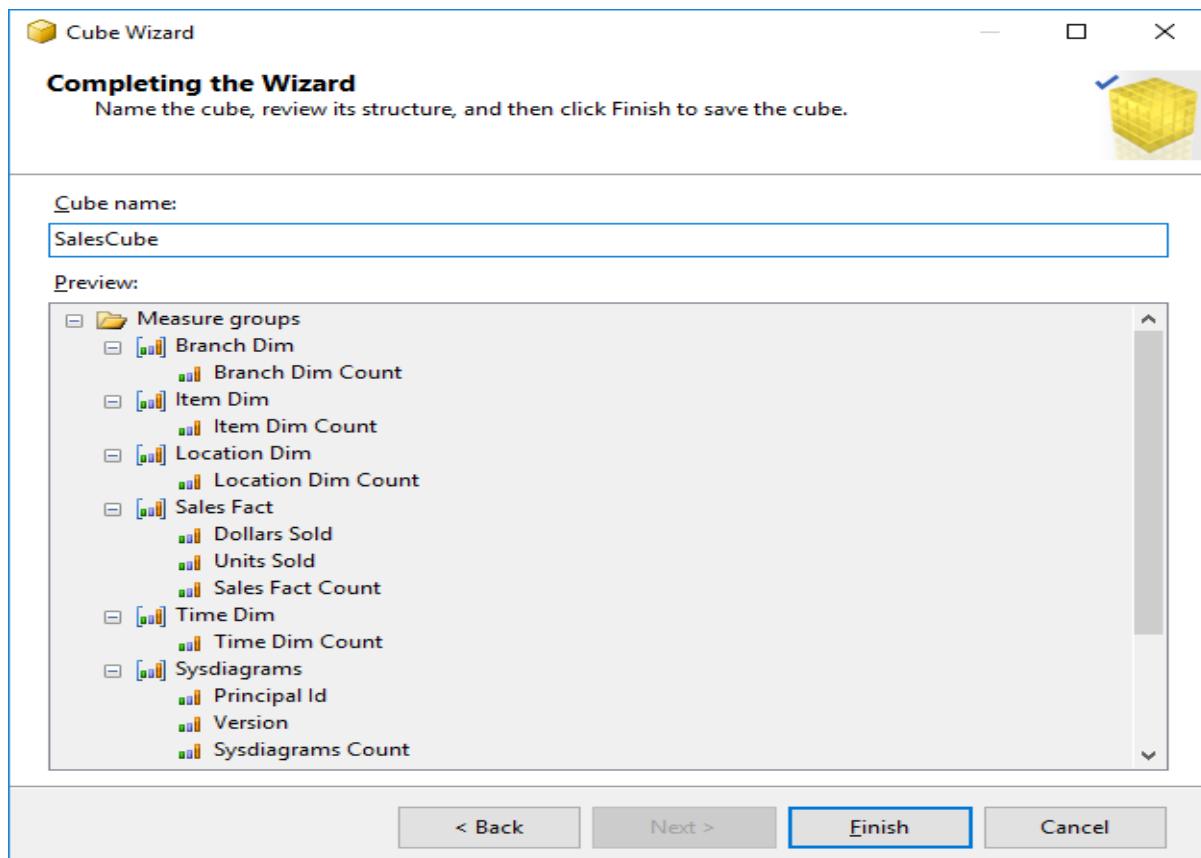


Click on Next.



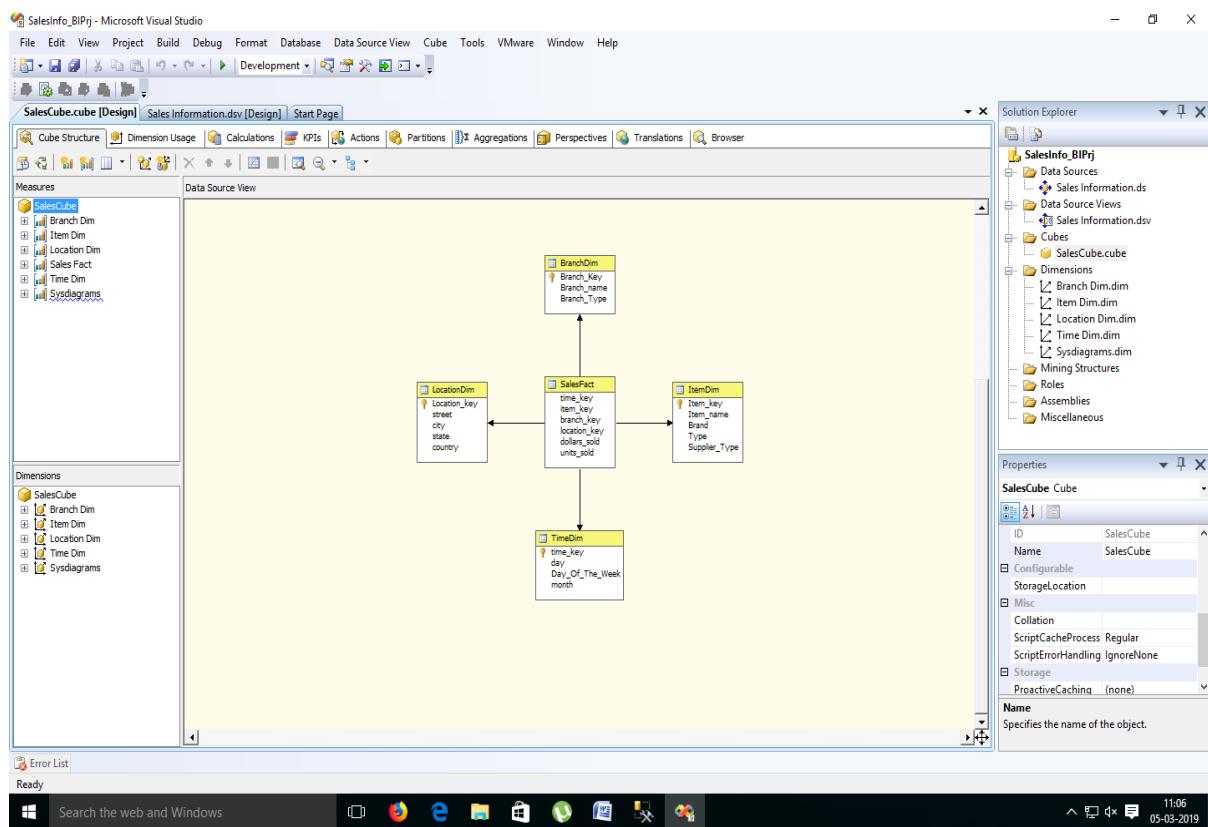
Click on Next.

4. Name Cube as “SalesCube”.

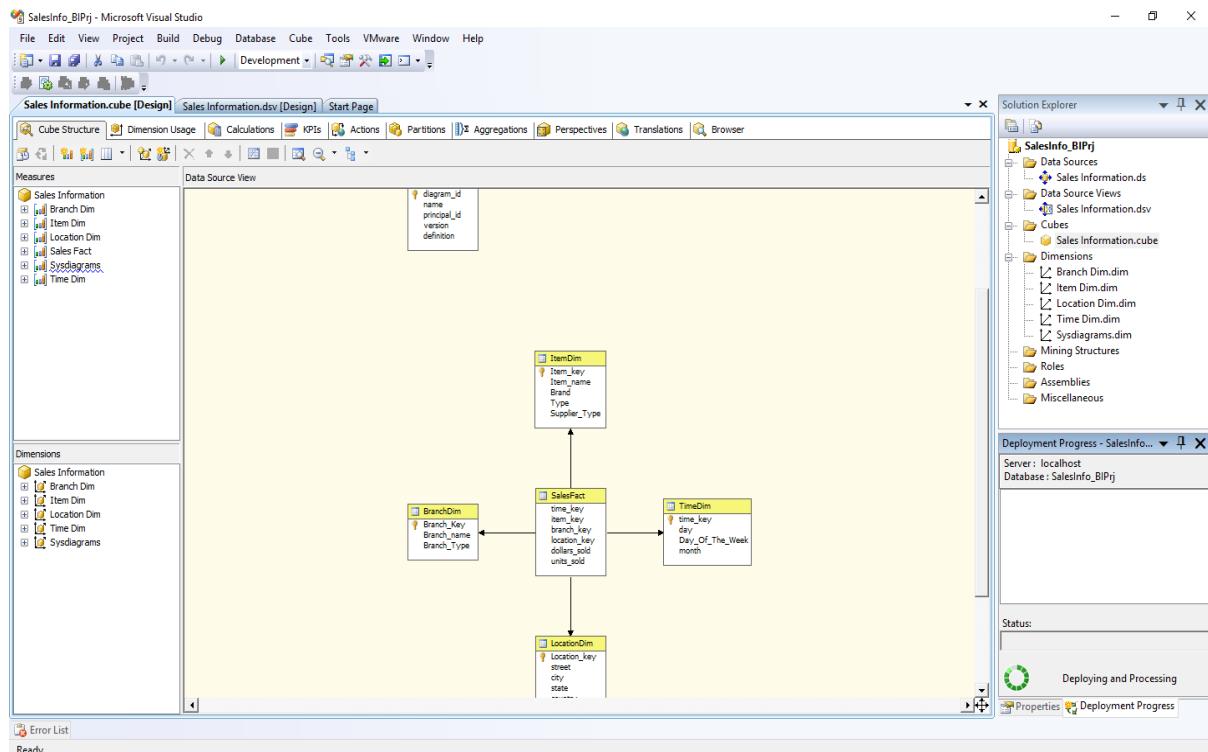


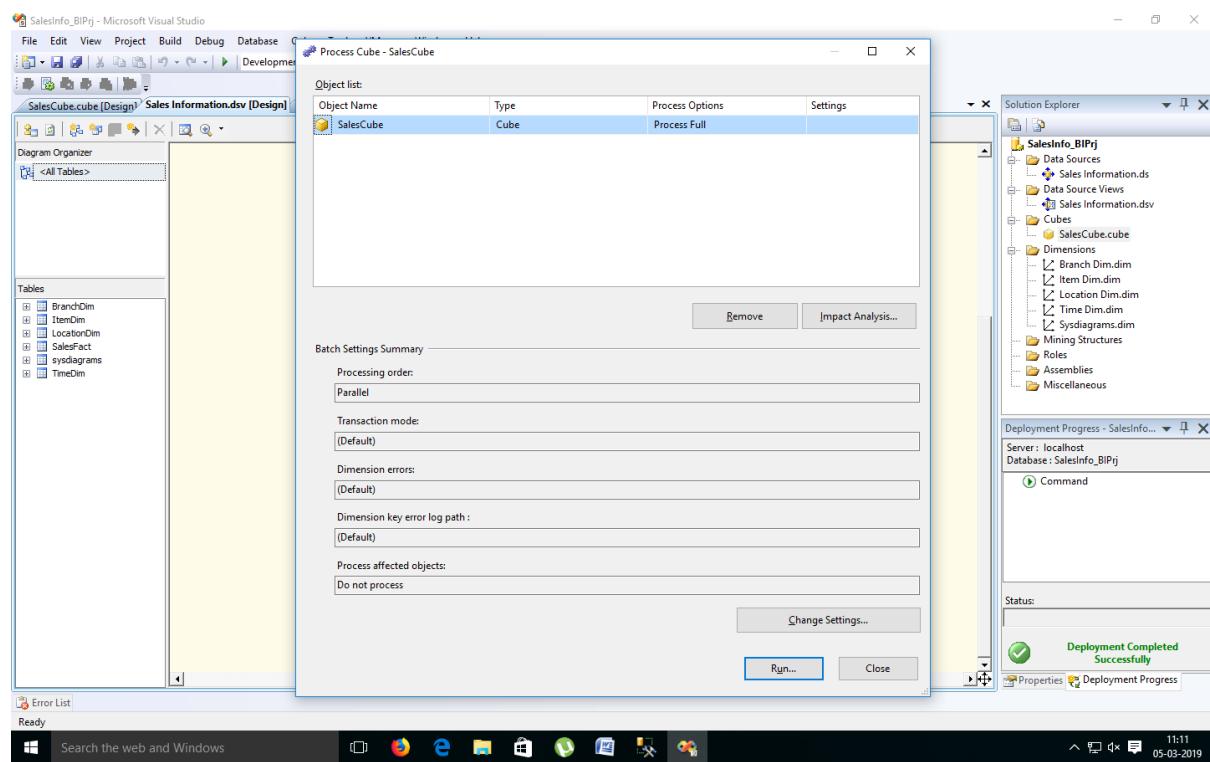
Click on Finish.

5. Finally, we will get the Cube View as well Dimensions View like :

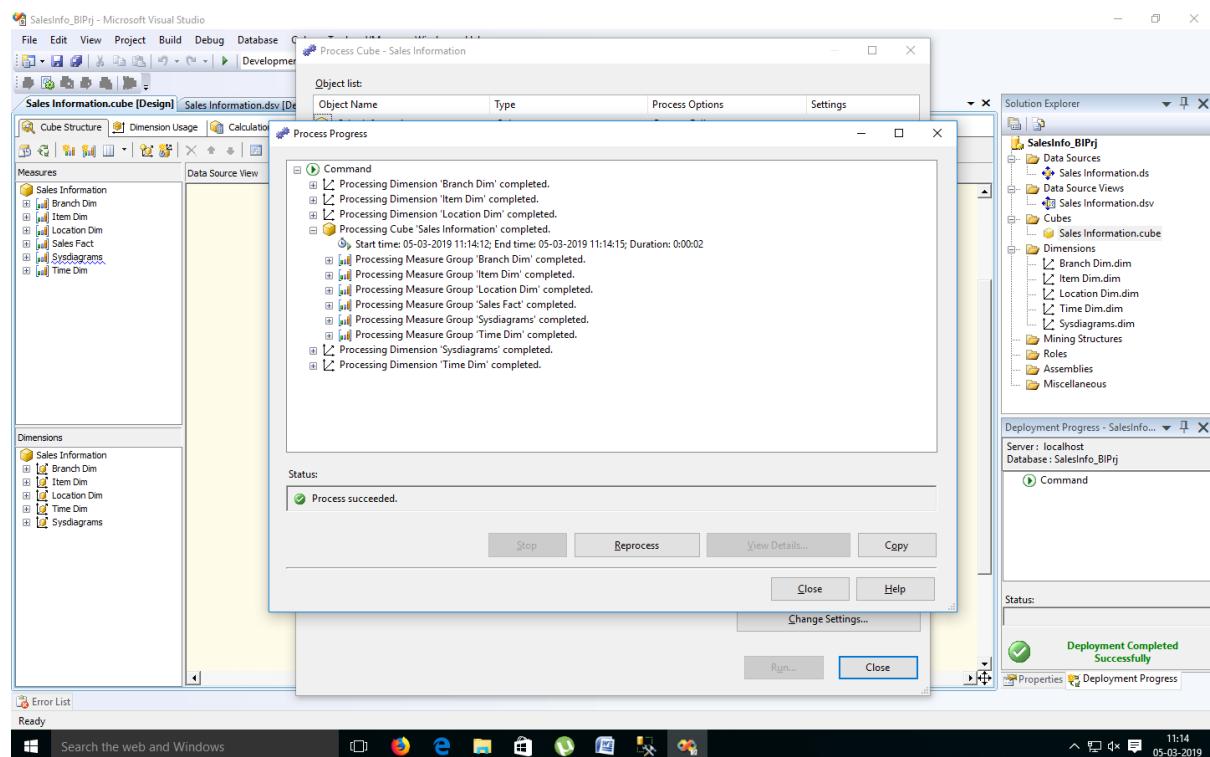


6. Finally, Process cube by Right click on SalesCube -> Process .





7. Click on Run.

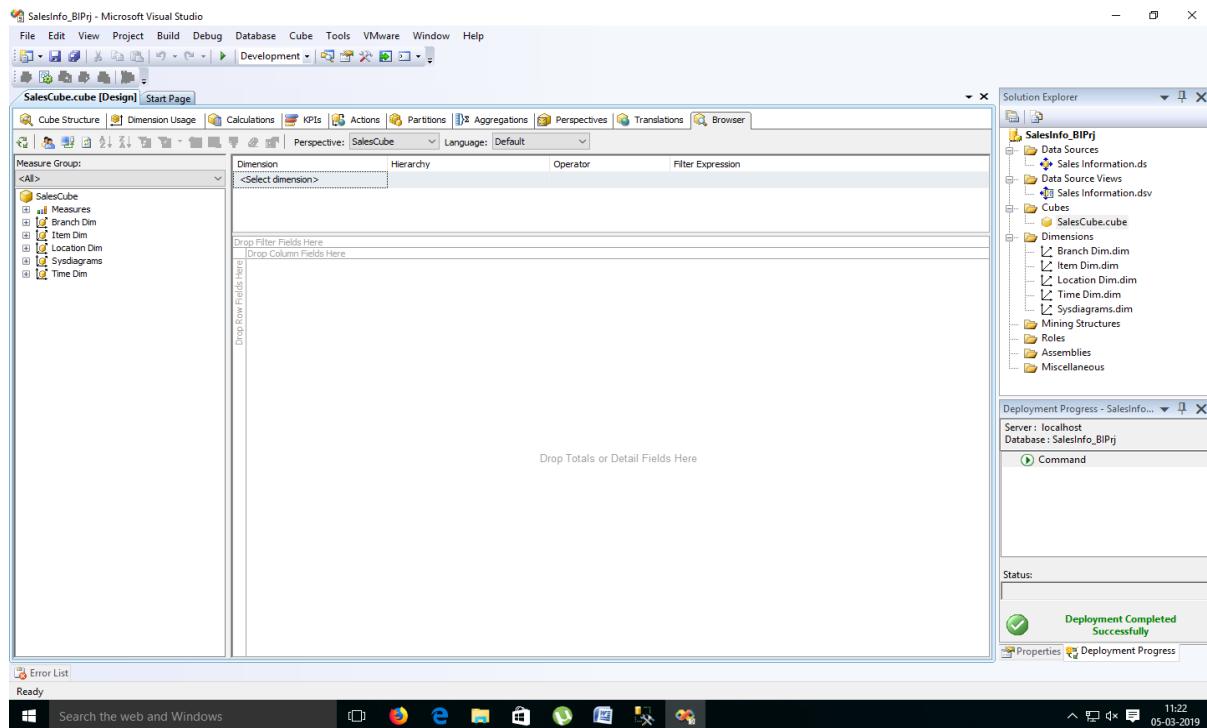


Practical No 5

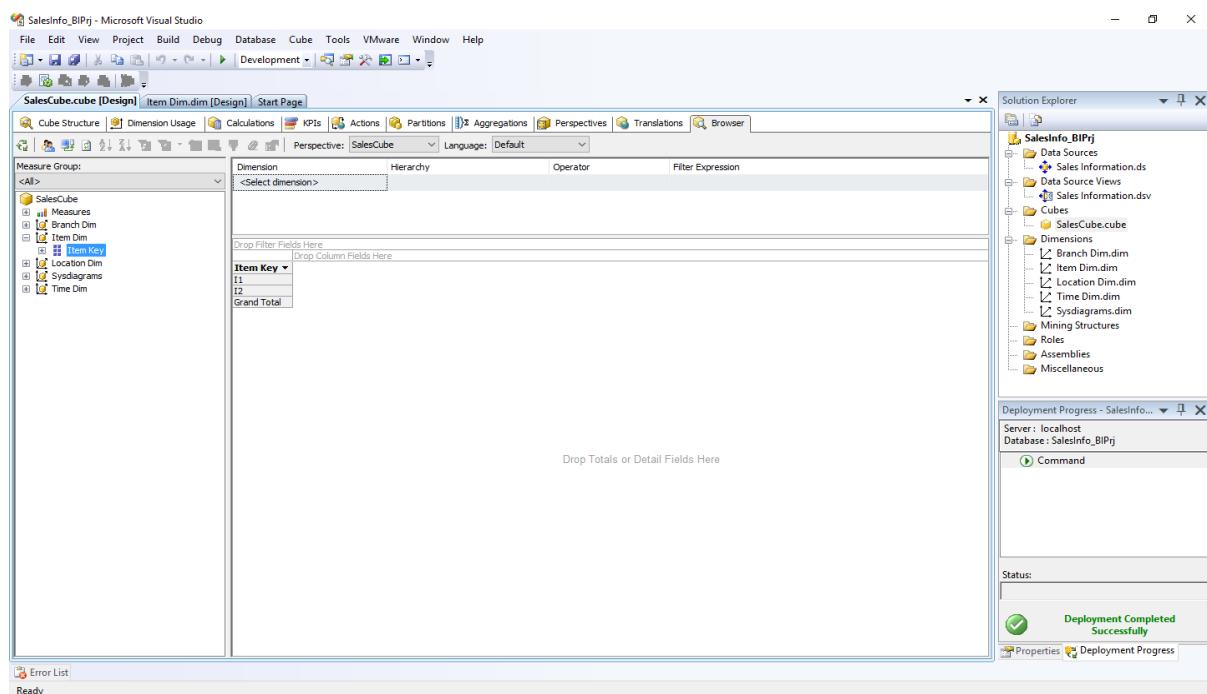
Aim : View cube data in multidimensional Format.

Solution :

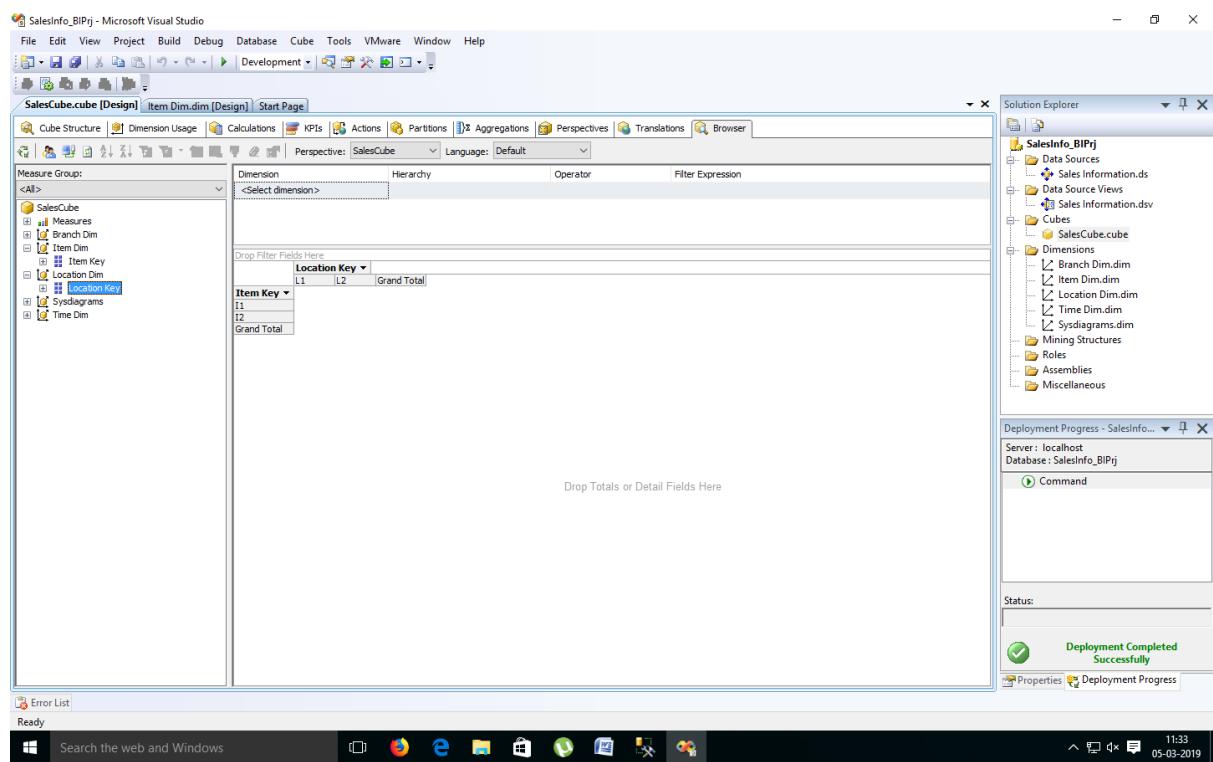
1. Double Click on “SalesCube”. Go to the “Browser” Tab.



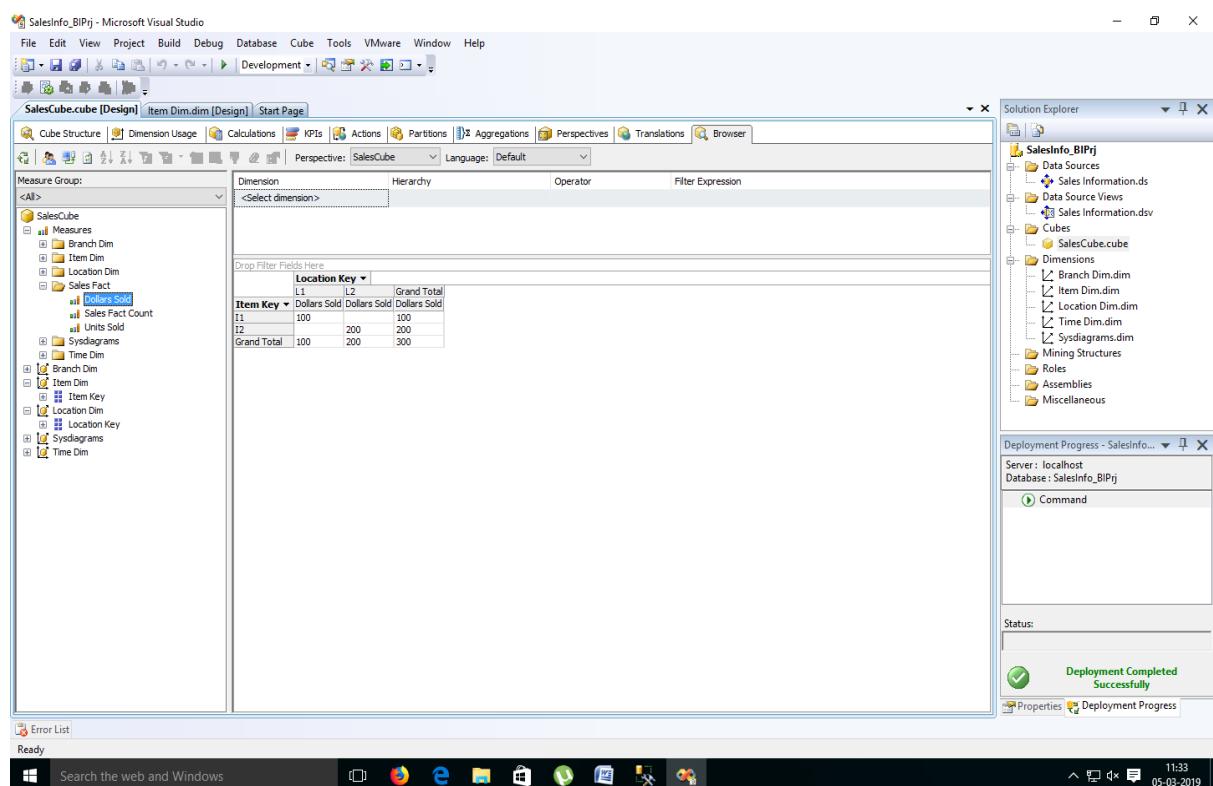
2. Go to the “Item Dimension”. Right Click on ‘Item Key’ -> Add to Row Area.



3. Go to the “Location Dimension”. Right Click on ‘Location Key’ -> Add to Column Area.



4. Go to ‘Measures’. Select ‘SalesFact’ -> Right Click on “Dollars Sold” -> Add to Data area.



5. Go to the “Branch Dimension”. Right Click on ‘Branch Key’ -> Add to Row Area.

The screenshot shows the Microsoft Visual Studio interface for Business Intelligence Development Studio (BIDS). The main window displays the 'SalesCube.cube [Design]' tab. On the left, the 'Measure Group' tree includes 'Measures' and 'Sales Fact' under 'SalesCube'. The 'Branch Dim' dimension is selected. In the center, a pivot table is shown with columns for 'Item Key', 'Branch Key', and 'Location Key'. The data grid contains the following values:

Item Key	Branch Key	Location Key		Grand Total
		L1	L2	
I1	B1	100	100	
	B2			
	Total	100	100	
I2	B1			
	B2	200	200	
	Total	200	200	
	Grand Total	100	200	300

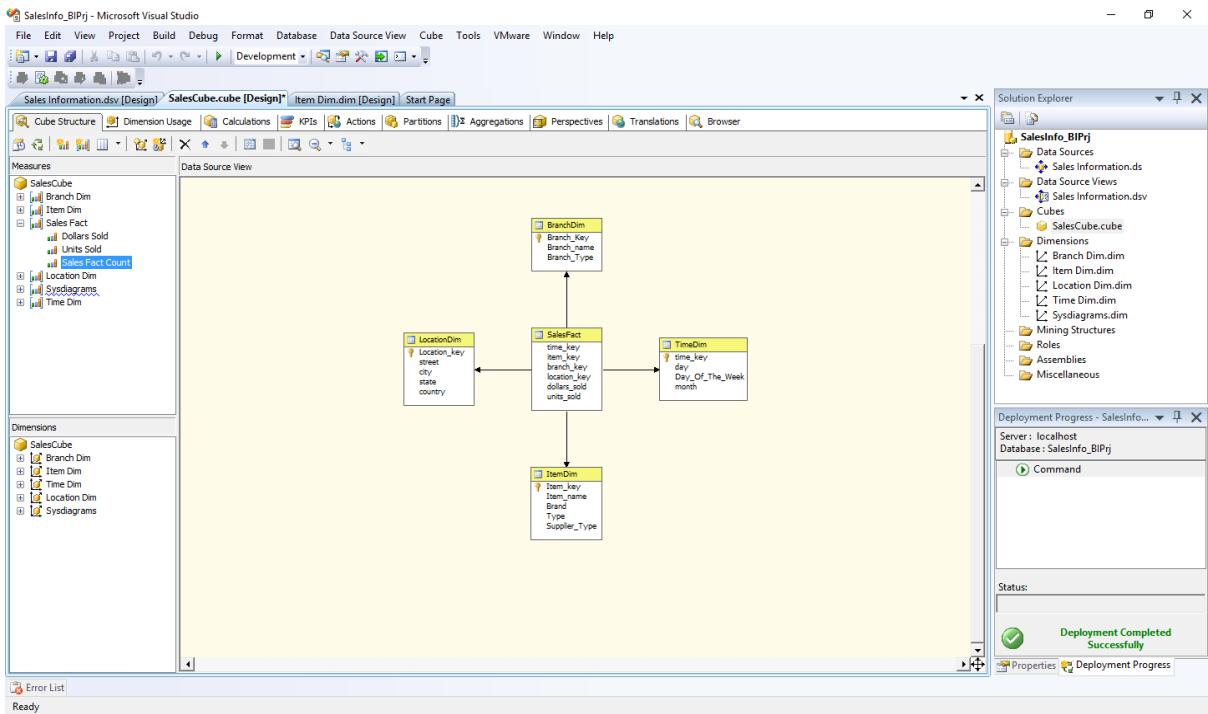
The right side of the screen shows the 'Solution Explorer' with the project structure for 'SalesInfo_BIPrj'. The 'Deployment Progress' window indicates a successful deployment to 'localhost' with the database 'SalesInfo_BIPrj'.

Practical No 6

Aim : Working with measures in the cube.

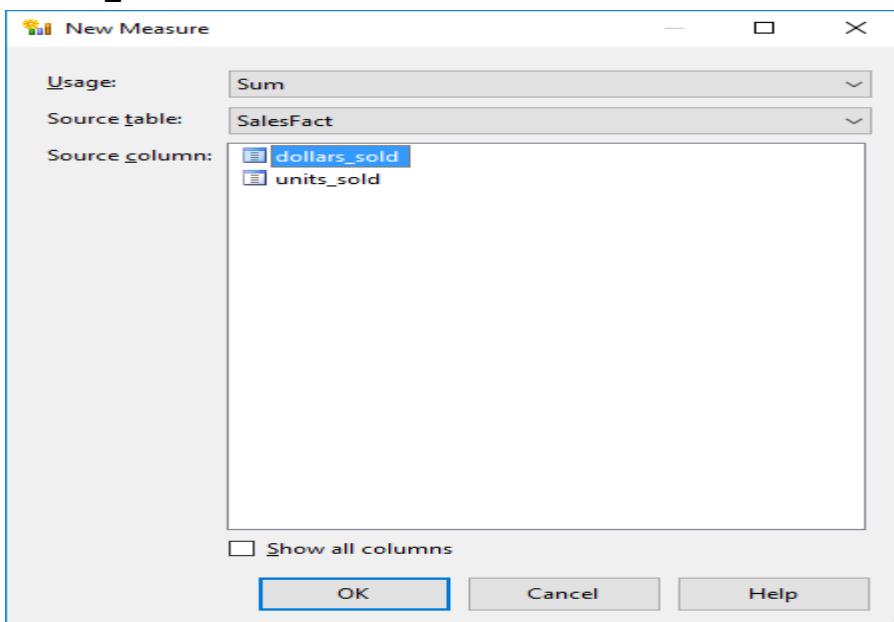
Solution :

1. Double click on 'SalesCube'. Go to cube structure.



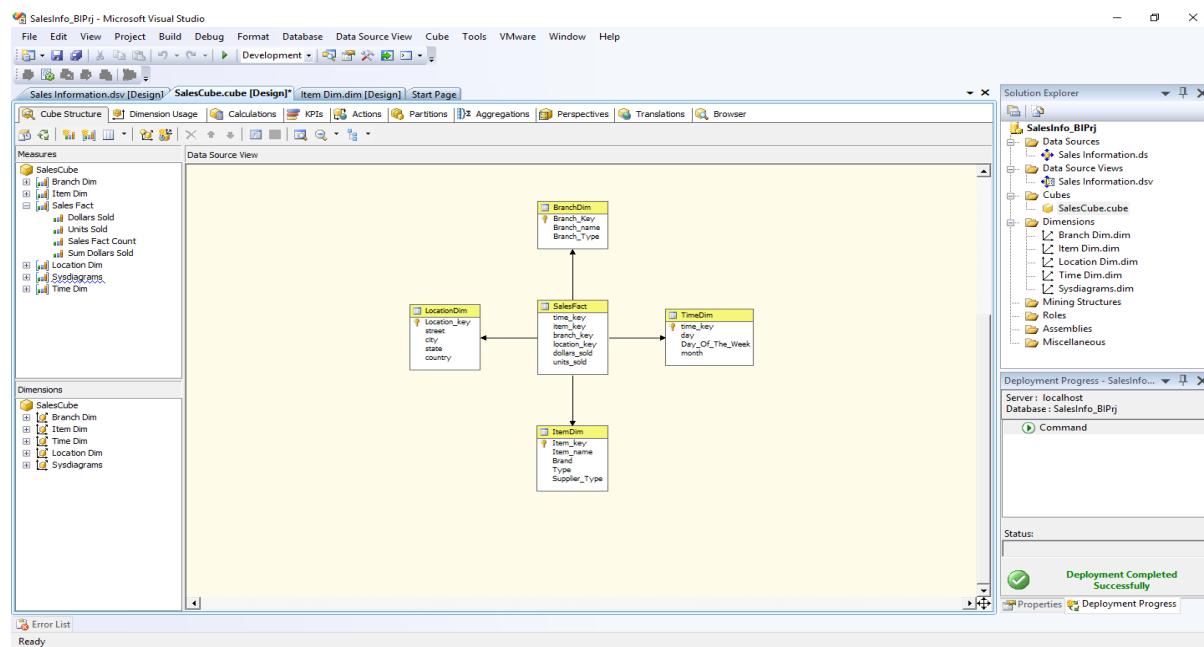
2. Right click on SalesCube -> New Measure.

Select Usage = "Sum" , Source table = "SalesFact" and Source Column = "dollars_sold".



Click on OK.

3. Rename Measure as “Sum Dollars sold”.



4. Process Cube and Go to Browser and Reconnect it. Right Click on “Sum Dollars Sold” -> Add to Data Area.

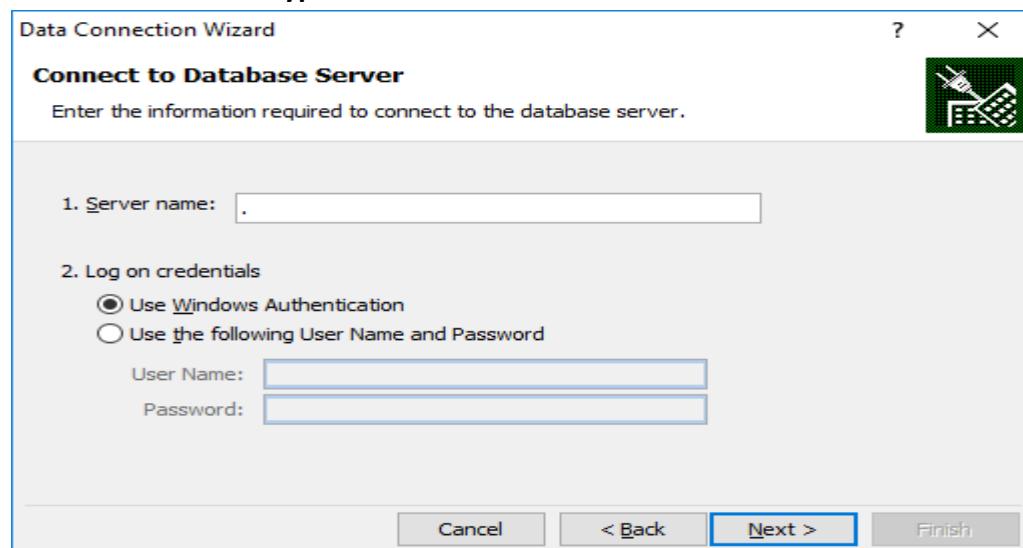
The screenshot shows the Microsoft Visual Studio interface for the SalesInfo_BIPrj project. The main window displays the Cube Structure designer for SalesCube.cube. The left sidebar shows the Measures pane, which now includes a renamed 'Sum Dollars Sold' measure under the SalesFact measure group. The cube structure is visualized in the center pane, showing the relationships between dimensions and facts. The Solution Explorer on the right shows the project structure, and the Deployment Progress panel indicates a successful deployment.

Practical No 7

Aim : Creating an Excel Pivot Table and Pivot Chart by using the OLAP cube data.

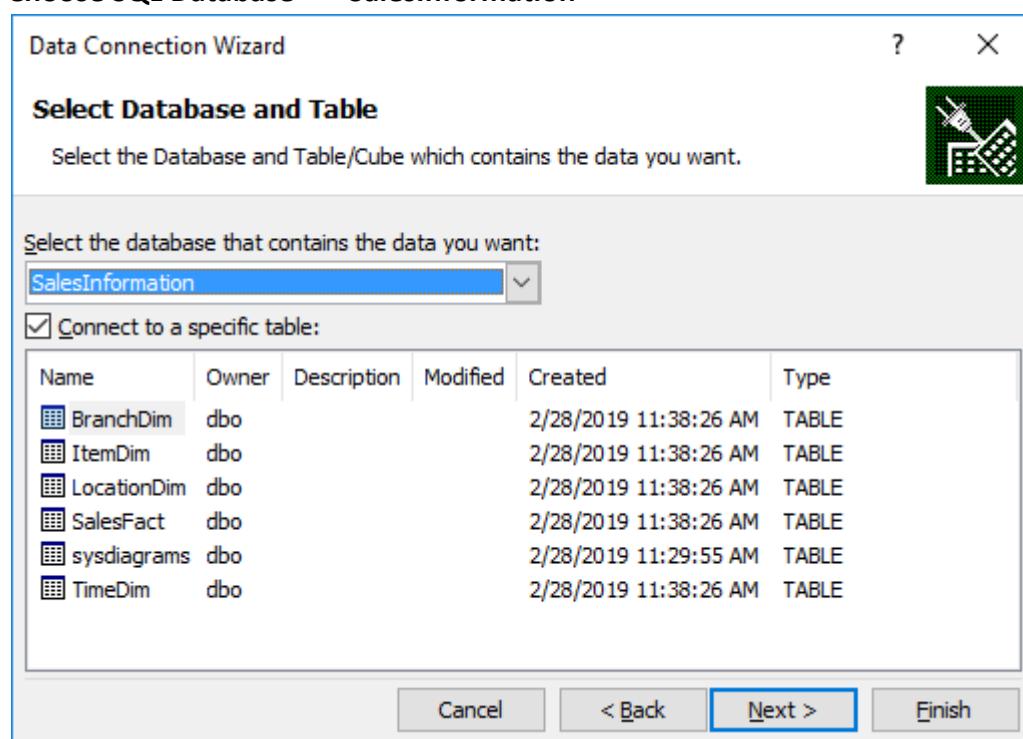
Solution :

1. Open MS-Excel. Click on Data Menu.
 2. Go to From Other Sources.
 - 2.1. From SQL Server -> Type Server name as “.”

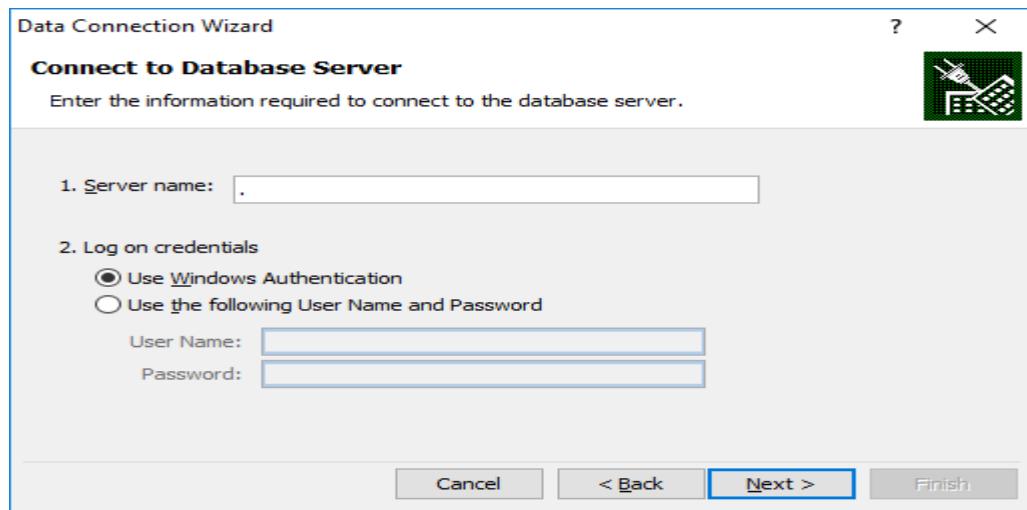


Click on Next.

Choose SQL Database -> “SalesInformation”

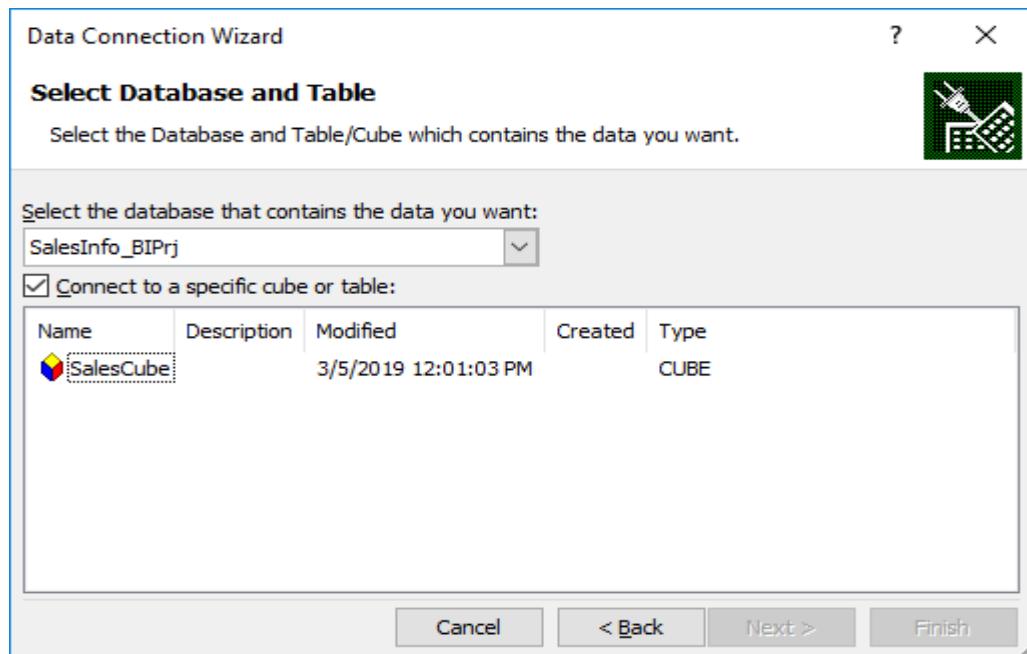


- ## 2.2. From Analysis Services -> Type Server name as “.”

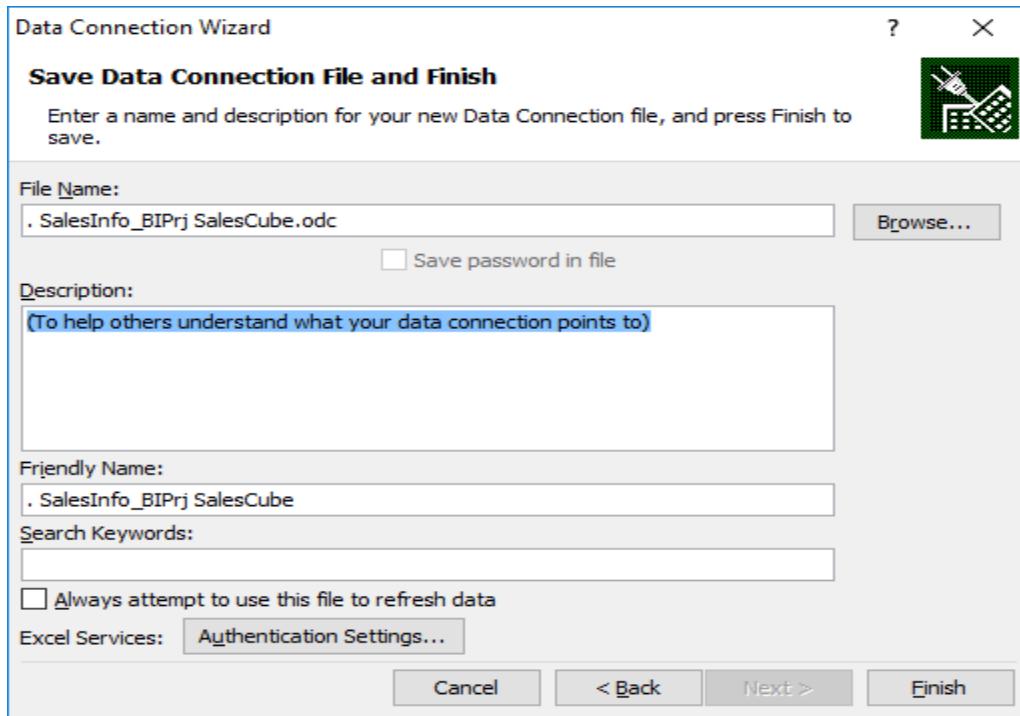


Click on Next.

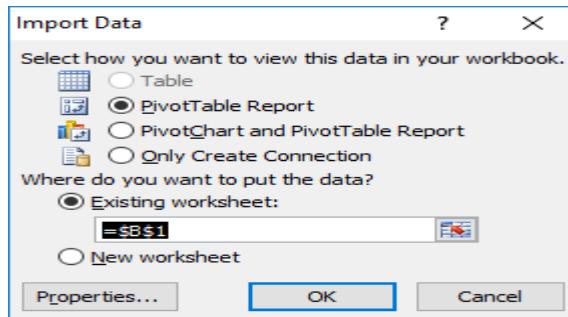
Choose Analysis Database as "SalesInfo_BIPrj". Click on Next.



Click on OK

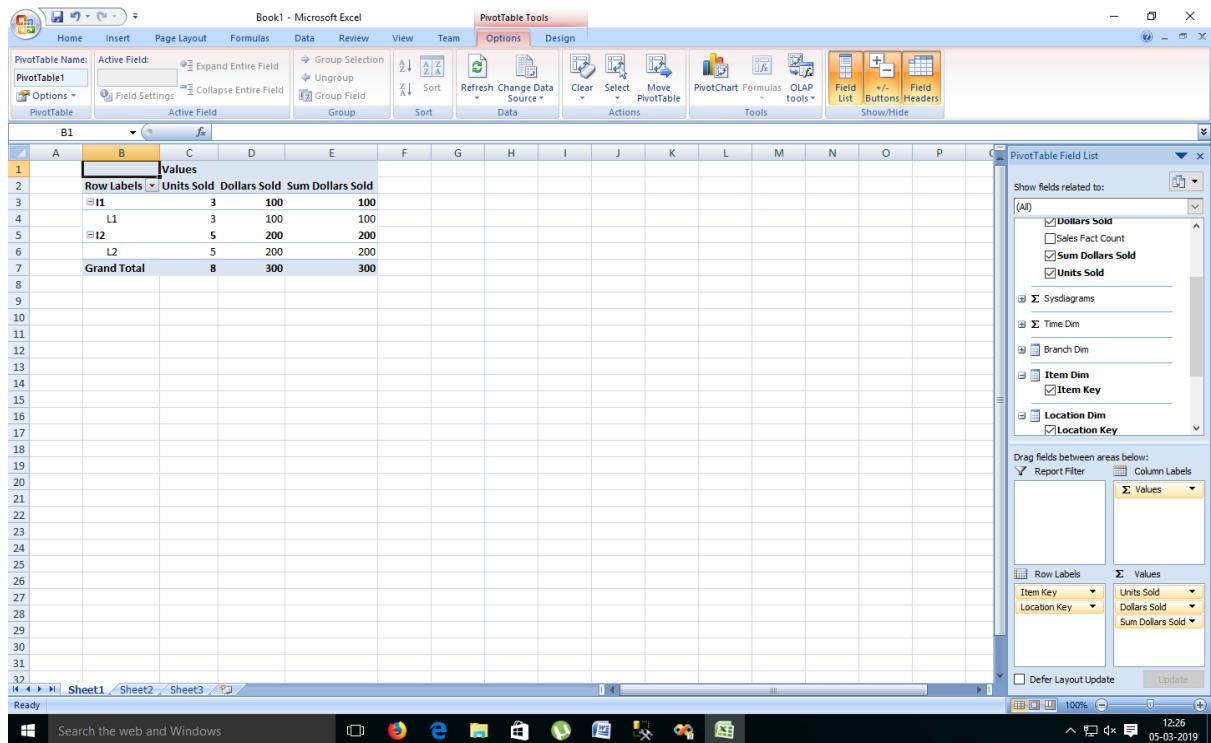


Click on Finish.

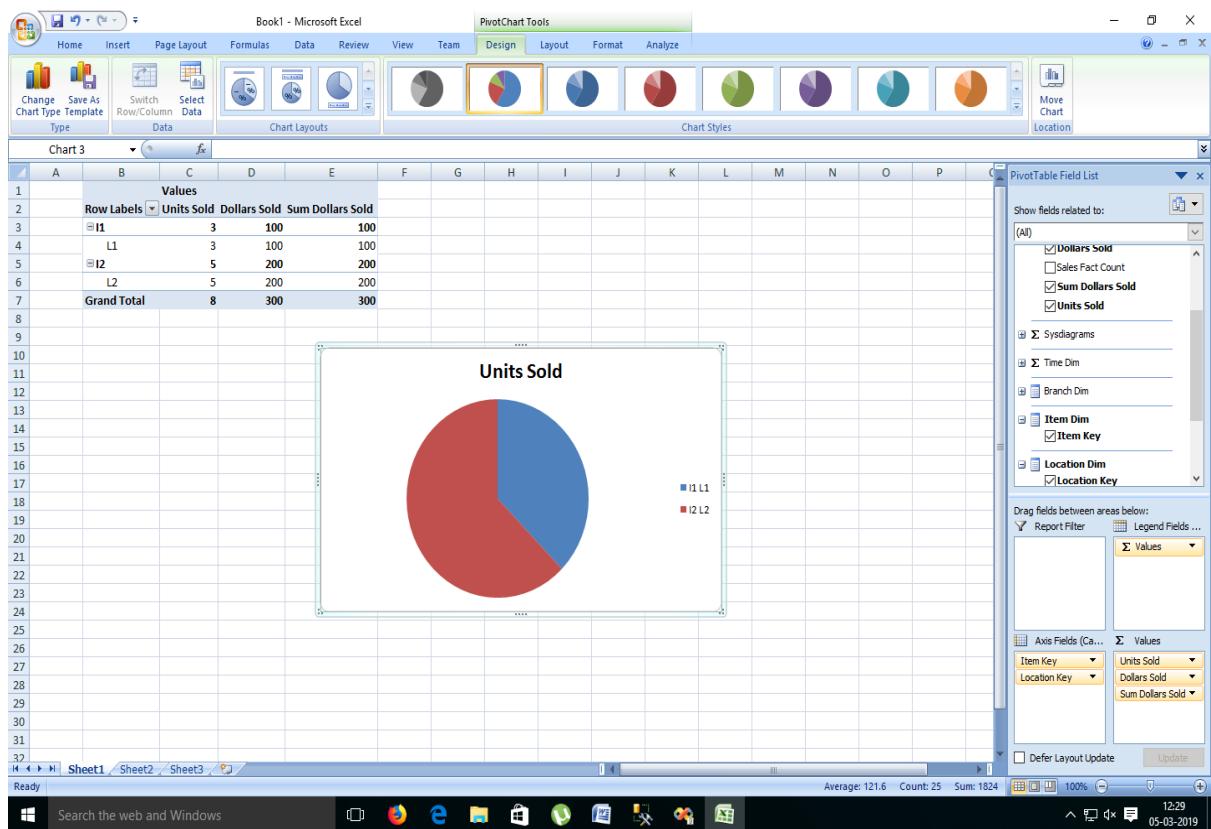


Click on OK.

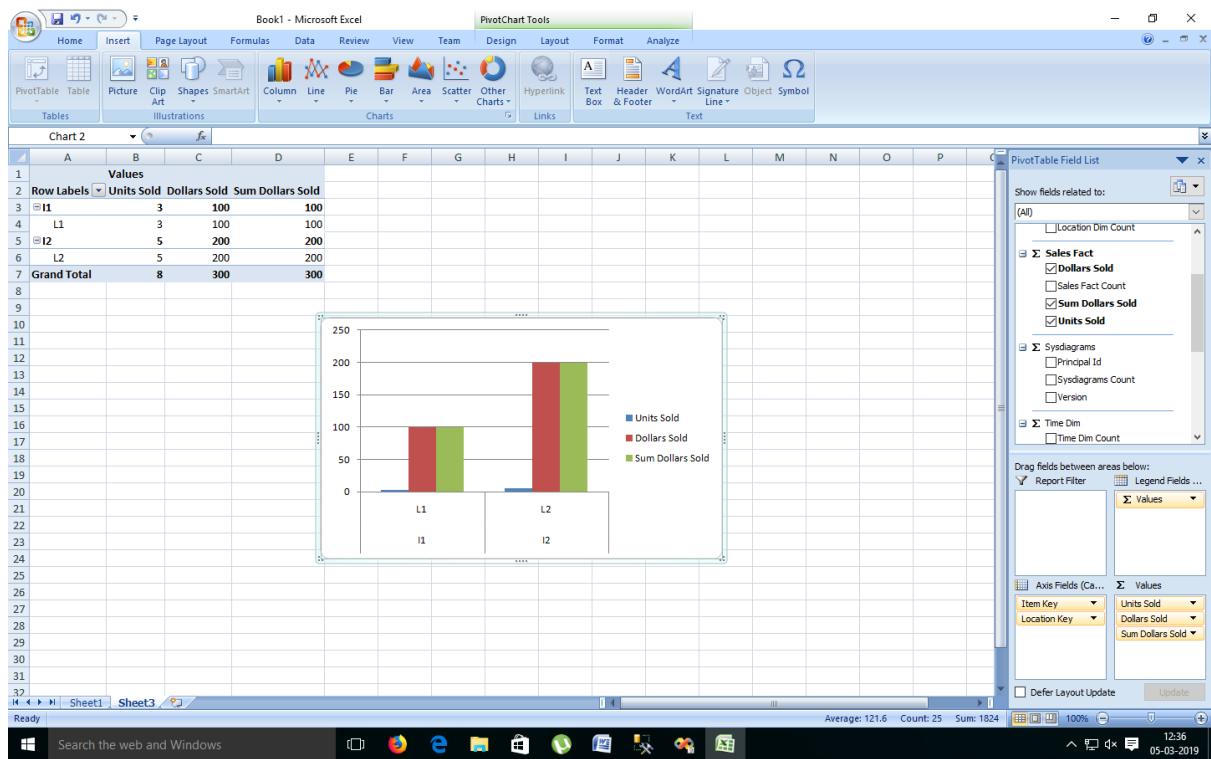
3. Select Item Key, Location Key and Measures as Dollars Sold, Units Sold and Sum Dollars Sold



4. Select Result Area. Go to Insert Menu. Select Pie Chart option.



5. Select Result Area. Go to Insert Menu. Select Column option.



Practical No 8

Aim : Firing Queries on Tables.

Solution :

Open Application -> Microsoft SQL Server 2008 R2 -> SQL Server Management Studio

1. Select Connect Tab -> Database Engine -> Select Server Name(local)

2. Expand ‘Database’ -> Expand ‘SalesInformation’ -> Expand Tables.

3. Fire following queries :

**3.1. SELECT [Branch_Key], [Branch_name], [Branch_Type]
FROM [SalesInformation].[dbo].[BranchDim]**

```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
Object Explorer
Connect > (local) SQL Server 10.50.1600 - BSC->
  Databases
    System Databases
    Database Snapshots
    ReportServer
    ReportServerTempDB
    Sales
    SalesInformation
      Tables
        SalesInfo
        System Tables
        BranchDim
        ItemDim
        LocationDim
        SalesFact
        TimeDim
      Views
      Synonyms
      Programmability
      Service Broker
      Storage
      Security
      SalesProduct
      Security
      Server Objects
      Replication
  Synonyms
  Programmability
  Service Broker
  Storage
  Security
  SalesProduct
  Security
  Server Objects
  Replication
  Output
  Ready

SQLQuery16.sql - C:\Users\BSC-37\min (53)* | SQLQuery16.sql ... C:\Users\BSC-37\min (53)* | master | 00:00:00 | 2 rows
SELECT [Branch_Key]
      ,[Branch_name]
      ,[Branch_Type]
  FROM [SalesInformation].[dbo].[BranchDim]

Results Messages
Branch_Key Branch_name Branch_Type
1 B1 RU Vendor
2 B2 JR Vendor

Query executed successfully. | (local) (10.50 RTM) | BSC-37\admin (53) | master | 00:00:00 | 2 rows

Properties
Current connection parameters
Aggregate Status
Connection fails.
Elapsed time 00:00:00.031
Finish time 08-03-2019 09:52:20
Name (local)
Rows returned 2
Start time 08-03-2019 09:52:20
State Open
Connection
Connection name (local) (BSC-37\admin)
Connection Details
Connection elap 00:00:00.031
Connection finis 08-03-2019 09:52:20
Connection row 2
Connection start 08-03-2019 09:52:20
Connection stat Open
Display name (local)
Login name BSC-37\admin
Server name (local)
Server version 10.50.1600
Session Tracing I
SPID 53

Name The name of the connection.

Ln 5 Col 44 INS
```

**3.2. SELECT [Item_key], [Item_name], [Brand], [Type], [Supplier_Type]
FROM [SalesInformation].[dbo].[ItemDim]**

```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
Object Explorer
Connect > (local) SQL Server 10.50.1600 - BSC->
  Databases
    System Databases
    Database Snapshots
    ReportServer
    ReportServerTempDB
    Sales
    SalesInformation
      Tables
        SalesInfo
        System Tables
        BranchDim
        ItemDim
        LocationDim
        SalesFact
        TimeDim
      Views
      Synonyms
      Programmability
      Service Broker
      Storage
      Security
      SalesProduct
      Security
      Server Objects
      Replication
  Synonyms
  Programmability
  Service Broker
  Storage
  Security
  SalesProduct
  Security
  Server Objects
  Replication
  Output
  Ready

SQLQuery17.sql - C:\Users\BSC-37\min (55)* | SQLQuery16.sql ... C:\Users\BSC-37\min (55)* | master | 00:00:00 | 2 rows
SELECT [Item_key]
      ,[Item_name]
      ,[Brand]
      ,[Type]
      ,[Supplier_Type]
  FROM [SalesInformation].[dbo].[ItemDim]

Results Messages
Item_Key Item_name Brand Type Supplier_Type
1 I1 Laptop LG Accessories VT
2 I2 Mouse LG Accessories VT

Query executed successfully. | (local) (10.50 RTM) | BSC-37\admin (55) | master | 00:00:00 | 2 rows

Properties
Current connection parameters
Aggregate Status
Connection fails.
Elapsed time 00:00:00.037
Finish time 08-03-2019 09:54:15
Name (local)
Rows returned 2
Start time 08-03-2019 09:54:15
State Open
Connection
Connection name (local) (BSC-37\admin)
Connection Details
Connection elap 00:00:00.037
Connection finis 08-03-2019 09:54:15
Connection row 2
Connection start 08-03-2019 09:54:15
Connection stat Open
Display name (local)
Login name BSC-37\admin
Server name (local)
Server version 10.50.1600
Session Tracing I
SPID 55

Name The name of the connection.

Ln 7 Col 42 INS
```

3.3. `SELECT [Location_key], [street], [city], [state], [country]
FROM [SalesInformation].[dbo].[LocationDim]`

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the SalesInformation database and its tables. The Results pane displays the output of the query:

```
SELECT [Location_key]
      ,[street]
      ,[city]
      ,[state]
      ,[country]
  FROM [SalesInformation].[dbo].[LocationDim]
```

	Location_key	street	city	state	country
1	L1	LBS	Mumbai	Maha	India
2	L2	JM	Thane	Maha	India

The status bar at the bottom indicates "Query executed successfully." and shows the session details: (local) (10.50 RTM) | BSC-37\admin (57) | master | 00:00:00 | 2 rows.

3.4. `SELECT [time_key], [item_key], [branch_key], [location_key]
, [dollars_sold], [units_sold]
FROM [SalesInformation].[dbo].[SalesFact]`

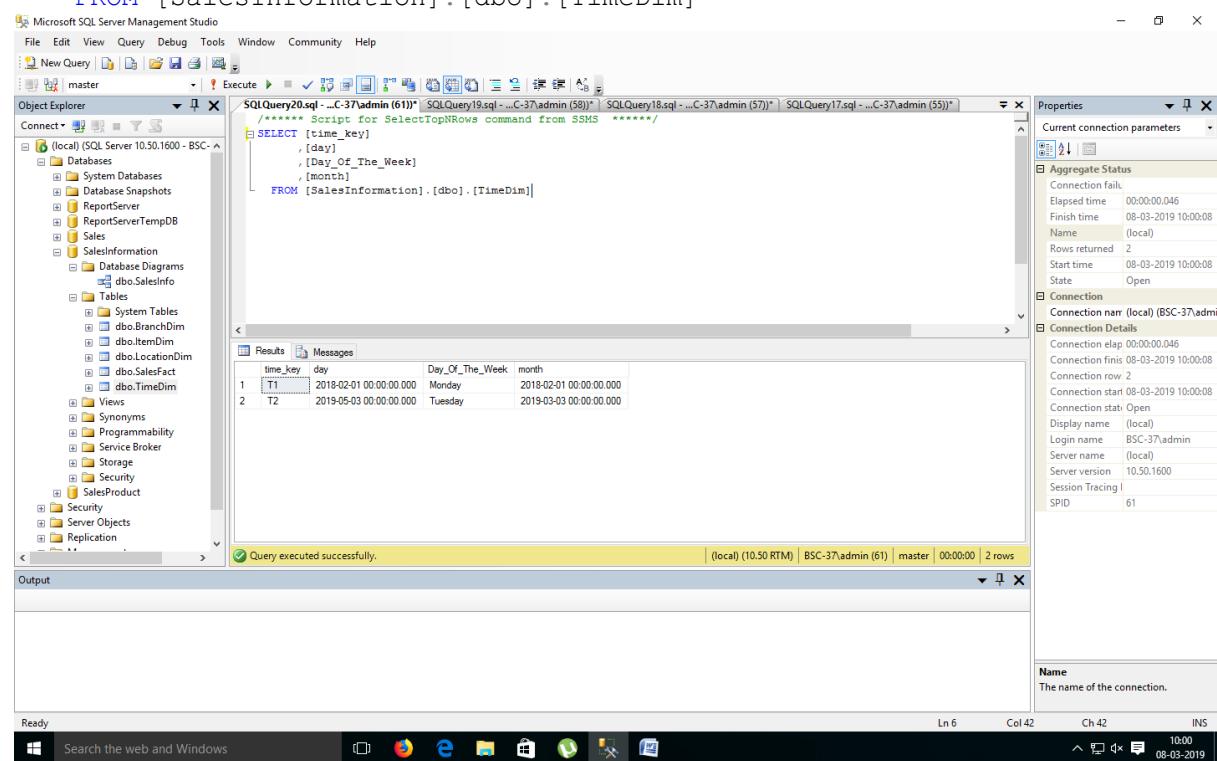
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the SalesInformation database and its tables. The Results pane displays the output of the query:

```
SELECT [time_key]
      ,[item_key]
      ,[branch_key]
      ,[location_key]
      ,[dollars_sold]
      ,[units_sold]
  FROM [SalesInformation].[dbo].[SalesFact]
```

	time_key	item_key	branch_key	location_key	dollars_sold	units_sold
1	T1	I1	B1	L1	100	3
2	T2	I2	B2	L2	200	5

The status bar at the bottom indicates "Query executed successfully." and shows the session details: (local) (10.50 RTM) | BSC-37\admin (58) | master | 00:00:00 | 2 rows.

3.5. `SELECT [time_key], [day], [Day_Of_The_Week], [month]
FROM [SalesInformation].[dbo].[TimeDim]`



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following code:

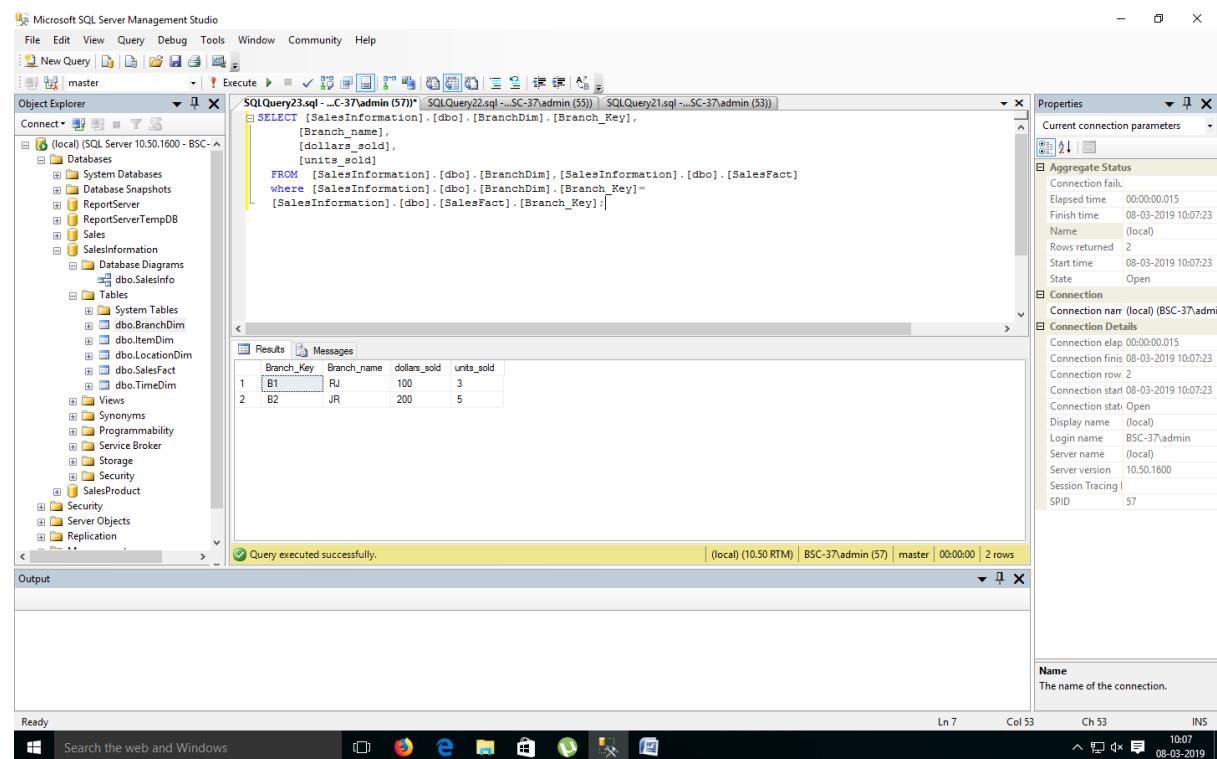
```
SQLQuery20.sql - C:\37\admin (61)* | SQLQuery19.sql - C:\37\admin (58)* | SQLQuery18.sql - C:\37\admin (57)* | SQLQuery17.sql - C:\37\admin (55)*
*****
-- Script for SelectTopNRows command from SSMS *****
SELECT [time_key]
      ,[day]
      ,[Day_Of_The_Week]
      ,[month]
  FROM [SalesInformation].[dbo].[TimeDim]
```

The results pane displays the following data:

time_key	day	Day_Of_The_Week	month
T1	2018-02-01 00:00:00.000	Monday	2018-02-01 00:00:00.000
T2	2019-05-03 00:00:00.000	Tuesday	2019-03-03 00:00:00.000

The status bar at the bottom right shows the date and time as 08-03-2019 10:00.

3.6. `SELECT [SalesInformation].[dbo].[BranchDim].[Branch_Key],
[Branch_name], [dollars_sold], [units_sold]
FROM [SalesInformation].[dbo].[BranchDim],
[SalesInformation].[dbo].[SalesFact]
where [SalesInformation].[dbo].[BranchDim].[Branch_Key]=
[SalesInformation].[dbo].[SalesFact].[Branch_Key];`



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following code:

```
SQLQuery23.sql - C:\37\admin (57)* | SQLQuery22.sql - C:\37\admin (55)* | SQLQuery21.sql - C:\37\admin (53)*
*****
-- Script for SelectTopNRows command from SSMS *****
SELECT [SalesInformation].[dbo].[BranchDim].[Branch_Key],
       [Branch_name],
       [dollars_sold],
       [units_sold]
  FROM [SalesInformation].[dbo].[BranchDim]
 WHERE [SalesInformation].[dbo].[BranchDim].[Branch_Key]=
       [SalesInformation].[dbo].[SalesFact].[Branch_Key];
```

The results pane displays the following data:

Branch_Key	Branch_name	dollars_sold	units_sold
B1	RJ	100	3
B2	JR	200	5

The status bar at the bottom right shows the date and time as 08-03-2019 10:07.

```
3.7. SELECT [SalesInformation].[dbo].[ItemDim].[Item_Key], [item_Name]
      , [Type], [dollars_sold], [units_sold]
   FROM [SalesInformation].[dbo].[ItemDim],
        [SalesInformation].[dbo].[SalesFact]
  Where [SalesInformation].[dbo].[ItemDim].[Item_key]=
    [SalesInformation].[dbo].[SalesFact].[item_key];
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window displays the following script:

```
***** Script for SelectTopNRows command from SSMS *****
SELECT [SalesInformation].[dbo].[ItemDim].[Item_Key]
      , [item_Name]
      , [Type]
      , [dollars_sold]
      , [units_sold]
   FROM [SalesInformation].[dbo].[ItemDim], [SalesInformation].[dbo].[SalesFact]
  where [SalesInformation].[dbo].[ItemDim].[Item_key]=[SalesInformation].[dbo].[SalesFact].[item_key];
```

The results pane shows the following data:

Item_Key	item_Name	Type	dollars_sold	units_sold
11	Laptop	Accessories	100	3
12	Mouse	Accessories	200	5

The status bar at the bottom indicates "Query executed successfully." and provides session details: (local) (10.50 RTM) | BSC-37\admin (56) | master | 00:00:00 | 2 rows.

```
3.8. SELECT [SalesInformation].[dbo].[LocationDim].[Location_key]
      , [city], [item_Key], [dollars_sold], [units_sold]
   FROM [SalesInformation].[dbo].[LocationDim],
        [SalesInformation].[dbo].[SalesFact]
  where [SalesInformation].[dbo].[LocationDim].[Location_key]=
    [SalesInformation].[dbo].[SalesFact].[location_key];
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window displays the following script:

```
***** Script for SelectTopNRows command from SSMS *****
SELECT [SalesInformation].[dbo].[LocationDim].[Location_key]
      , [city]
      , [item_Key]
      , [dollars_sold]
      , [units_sold]
   FROM [SalesInformation].[dbo].[LocationDim], [SalesInformation].[dbo].[SalesFact]
  where [SalesInformation].[dbo].[LocationDim].[Location_key]=
    [SalesInformation].[dbo].[SalesFact].[location_key];
```

The results pane shows the following data:

Location_Key	city	item_Key	dollars_sold	units_sold
L1	Mumbai	I1	100	3
L2	Thane	I2	200	5

The status bar at the bottom indicates "Query executed successfully." and provides session details: (local) (10.50 RTM) | BSC-37\admin (56) | master | 00:00:00 | 2 rows.

Below the results, a message box shows auto-attach information:

```
Auto-attach to process '1044' (SQL) '(local)' on machine '(local)' succeeded.
The thread '(local) (56)' (0x1680) has exited with code 0 (0x0).
The thread '(local) (56)' (0x1680) has exited with code 0 (0x0).
The program '1044' (SQL) '(local)' has exited with code 0 (0x0).
```

Practical No 9

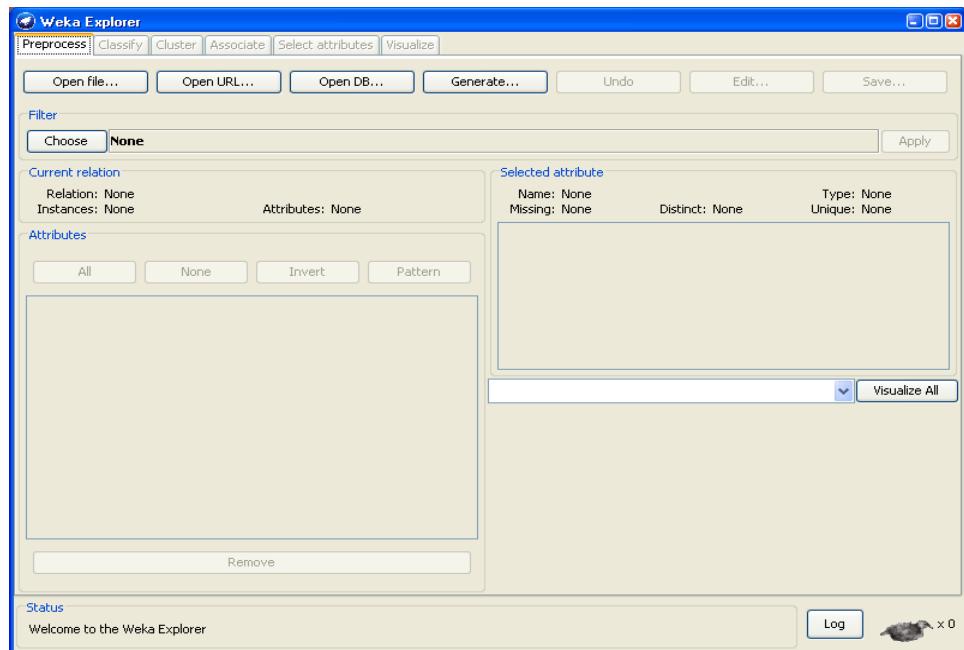
Aim : Calculation & KPI

Practical No - 10

Aim : Data PreProcessing

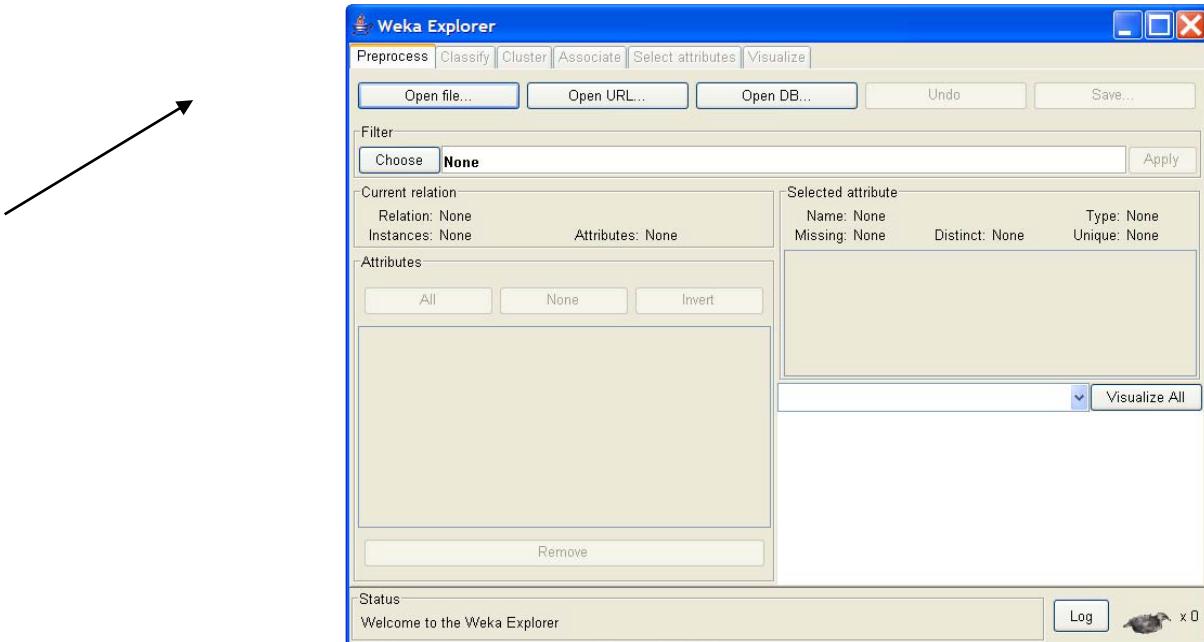
Solution :

Only the first tab, ‘Preprocess’, is active at the moment because there is no dataset open.

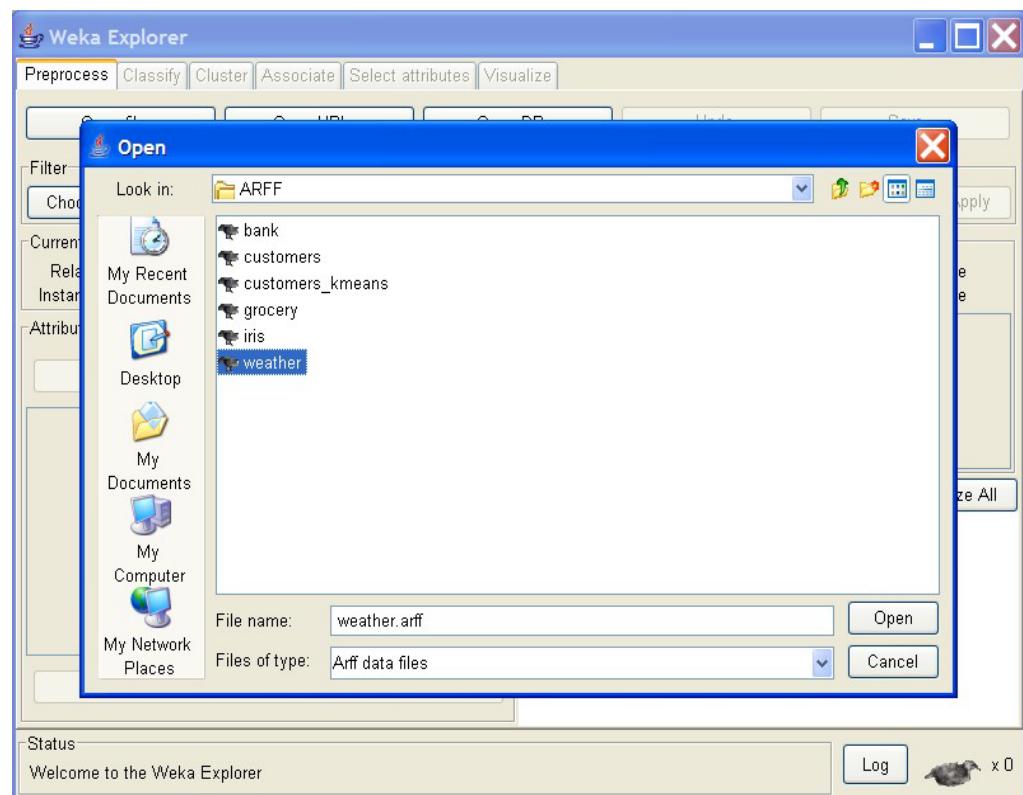


Opening file from a local file system

Click on ‘Open file...’ button

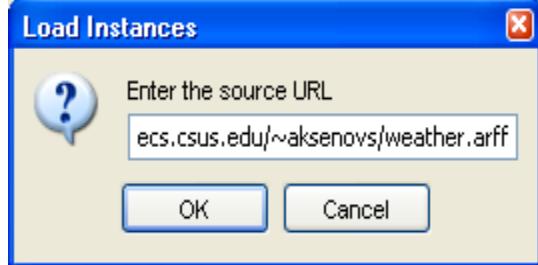


It brings up a dialog box allowing you to browse for the data file on the local file system, choose “weather.arff” file.

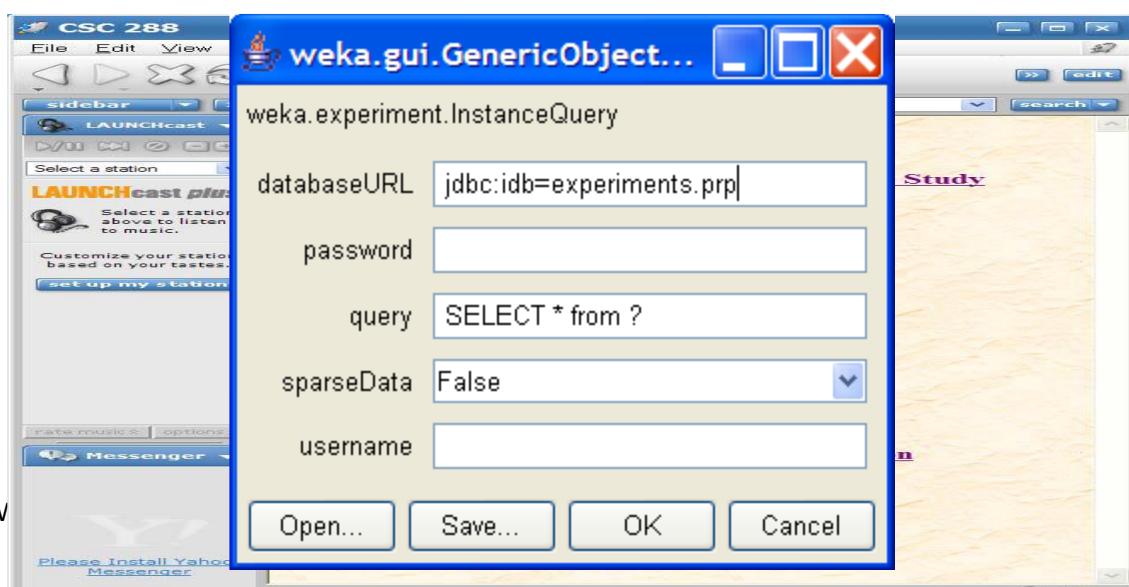


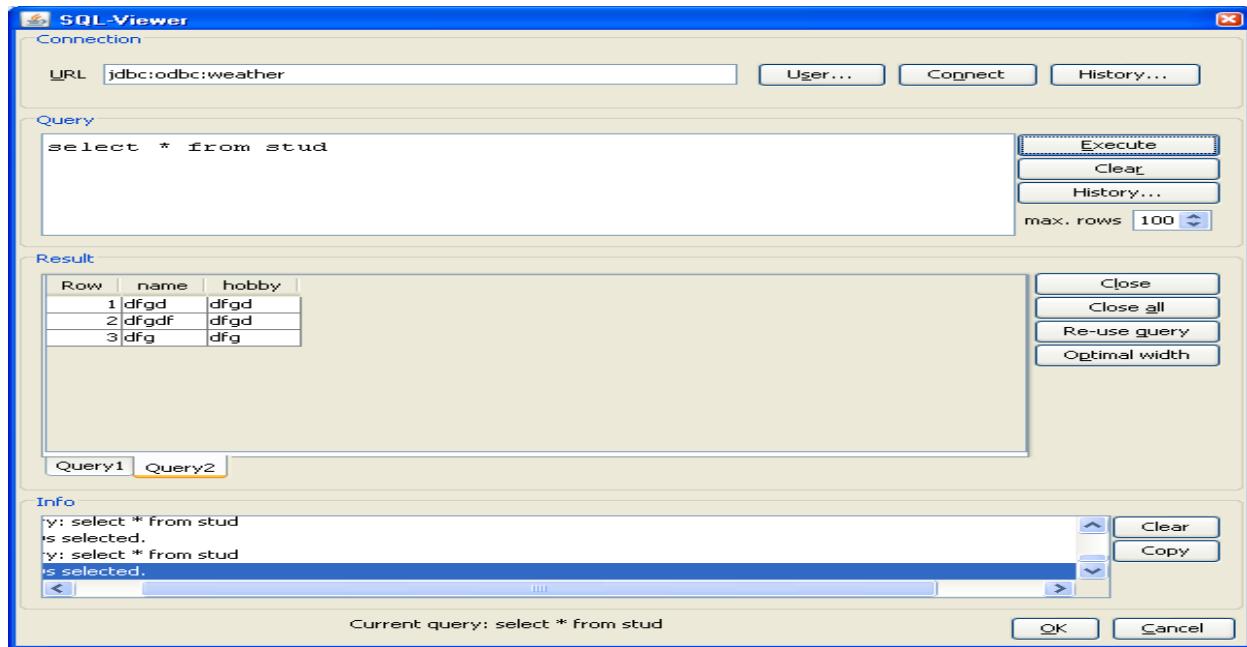
Opening file from a web site

A file can be opened from a website. Suppose, that “weather.arff” is on the following website:



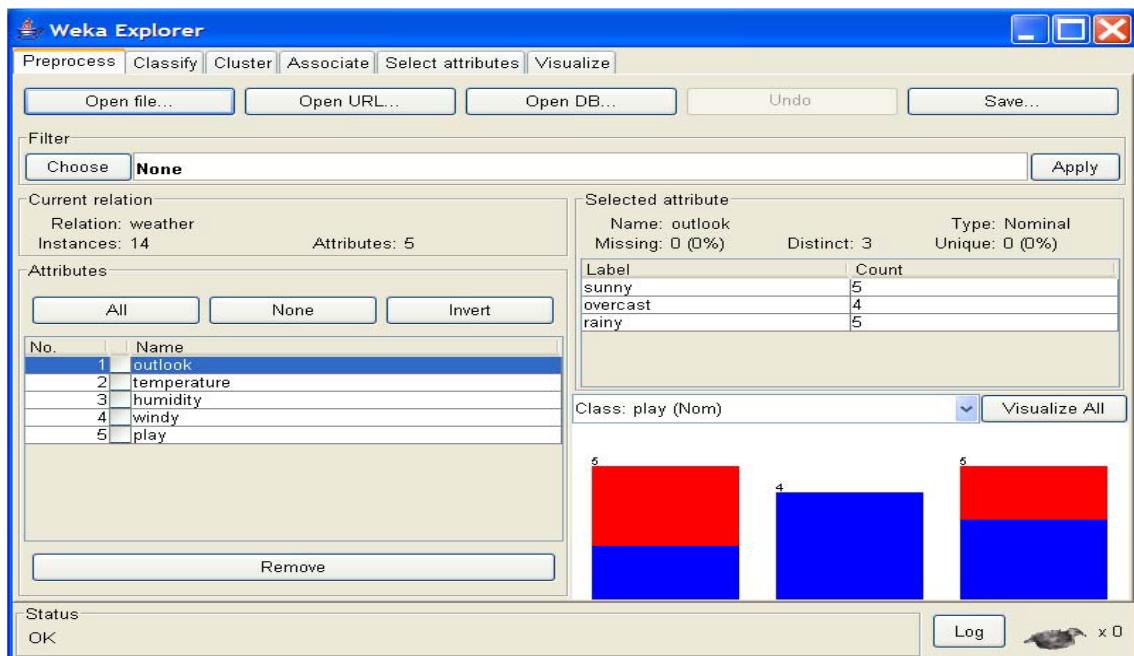
Reading data from a database:

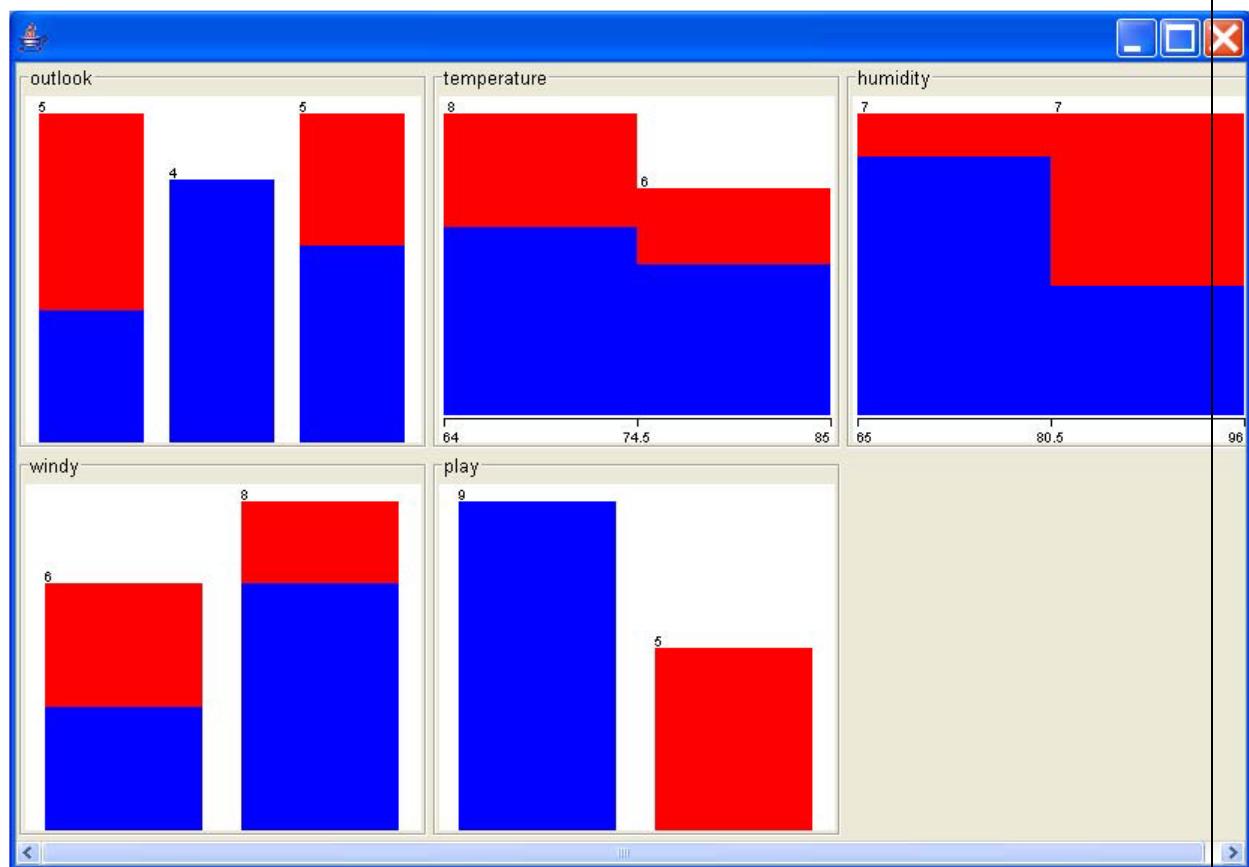




Loading data

The most common and easiest way of loading data into WEKA is from ARFF file, using Open File button.



Visualize Attributes:

visualize all attributes by clicking on 'Visualize All' button.,

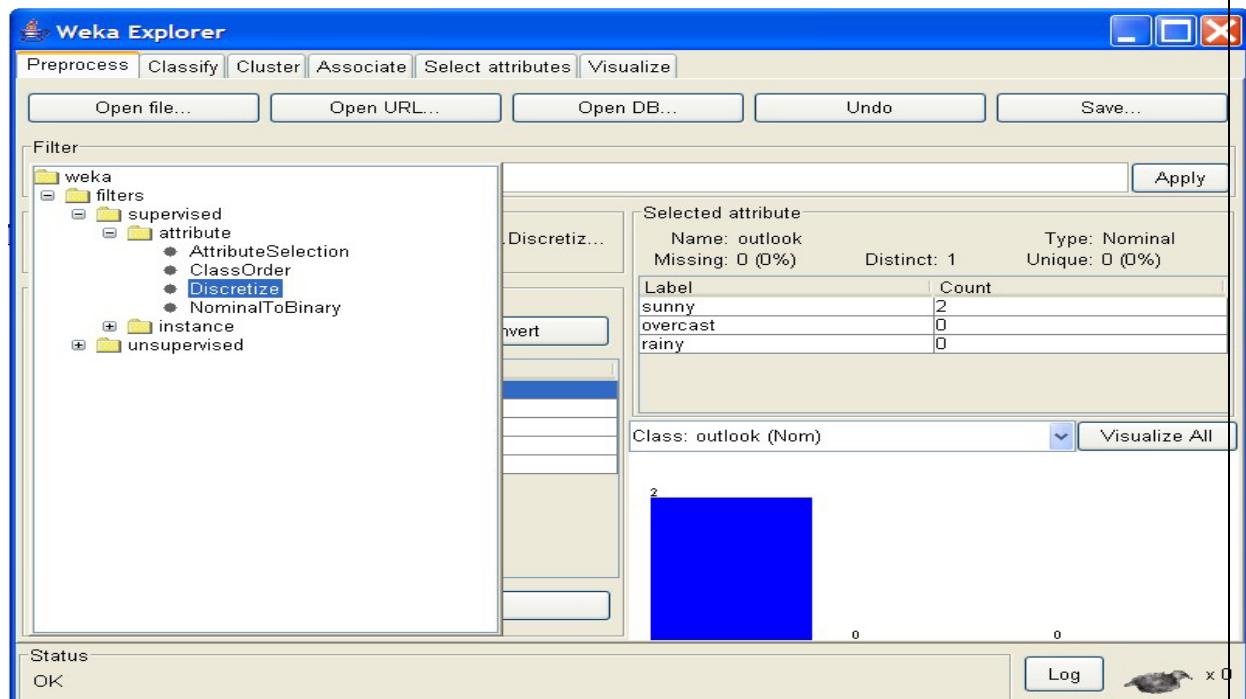
Practical No – 11

Aim : Data discretization.

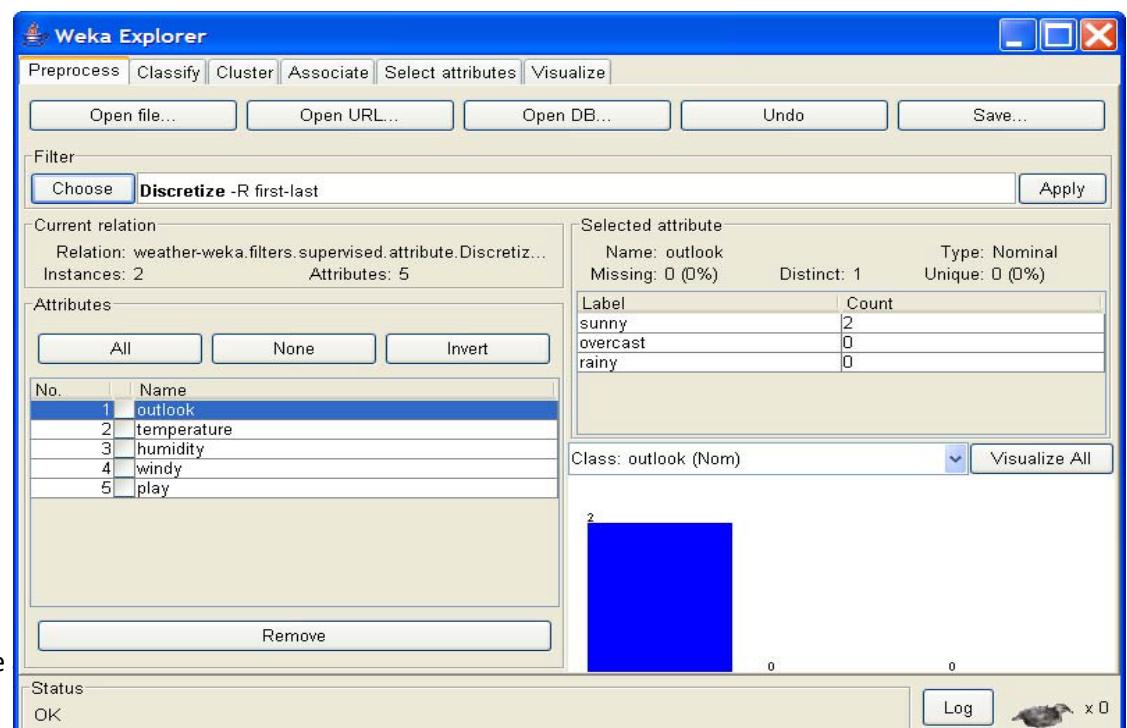
Solution :

In ‘Filters’ window, click on the ‘Choose’ button.

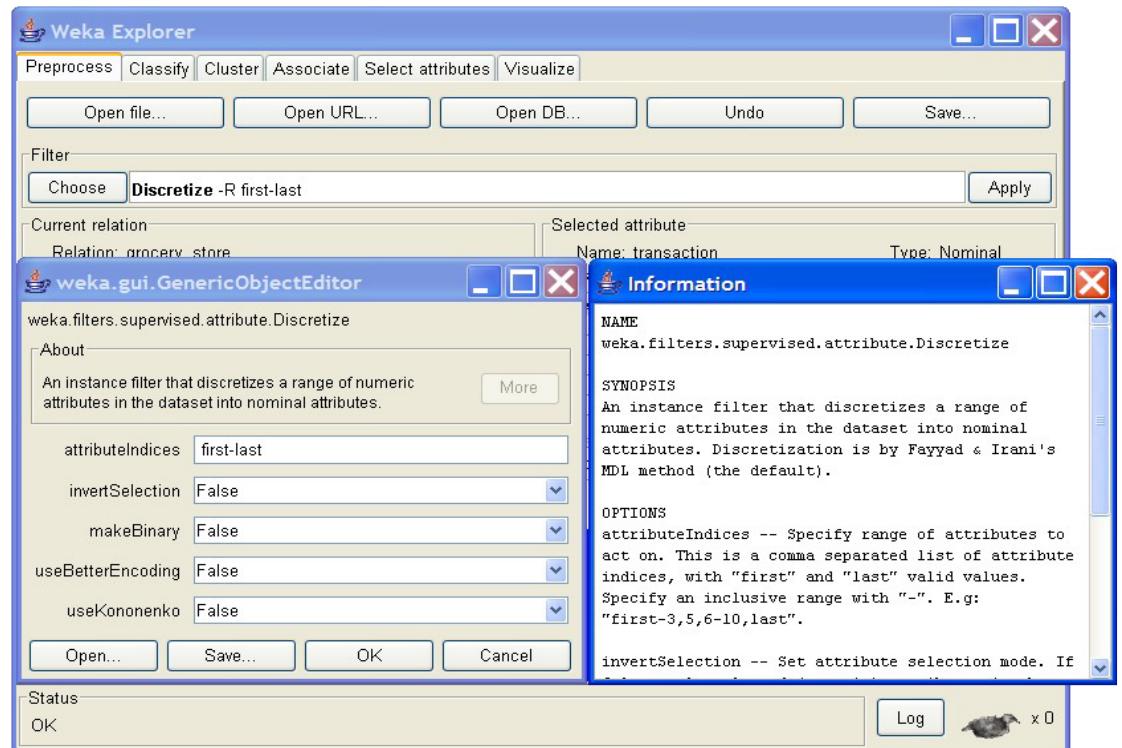
This will show pull-down menu with a list of available filters. Select Supervised → Attribute → Discretize and click on ‘Apply’ button.



The filter will convert Numeric values into Nominal.
the fields in the window changes to reflect available options.



a ‘GenericObjectEditor’ dialog box comes up on your screen.
The box lets you choose the filter configuration options.

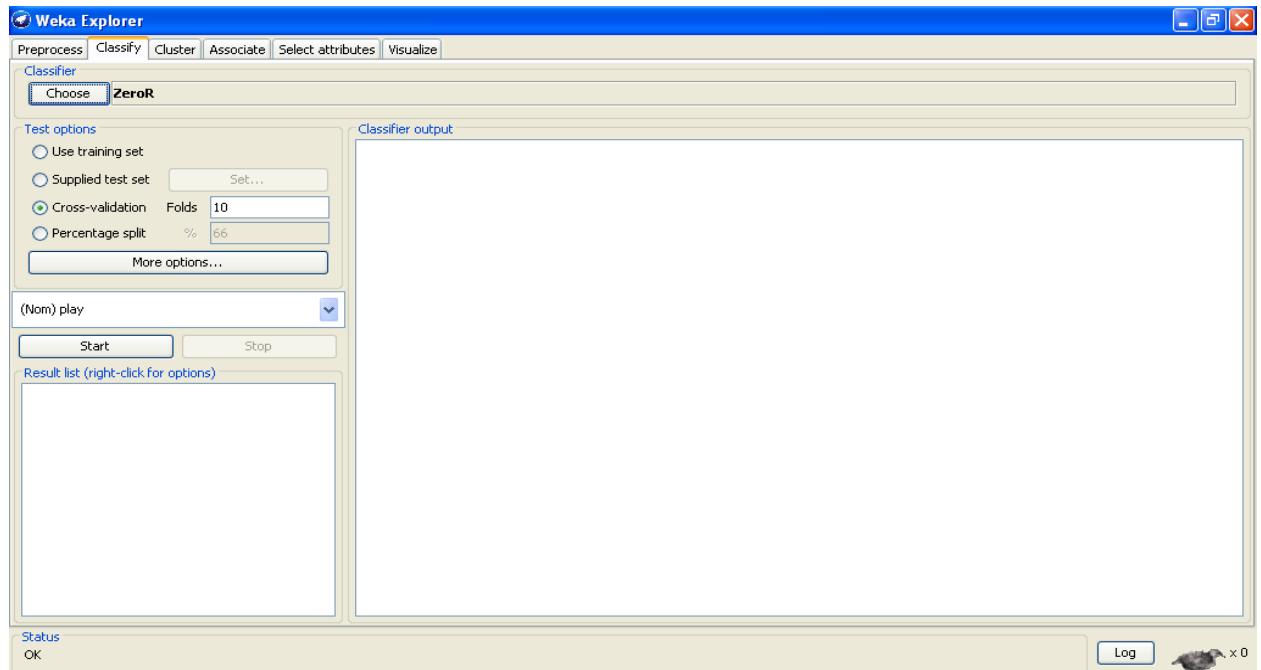


Practical No - 12

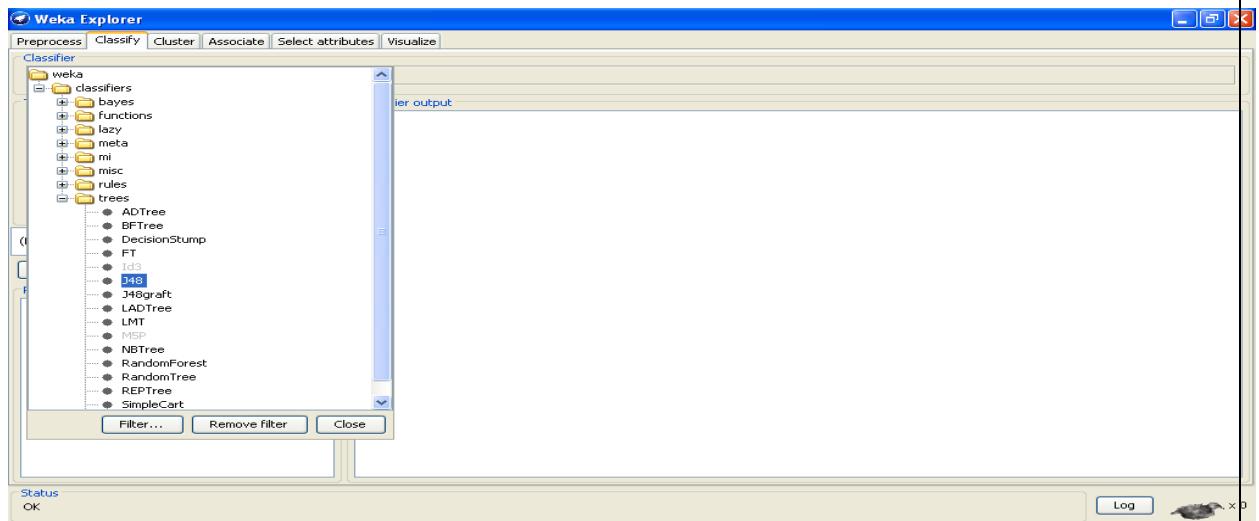
Aim : Classification problems.

Solution :

Once you have your data set loaded, all the tabs are available to you. Click on the ‘Classify’ tab.



Click on ‘Choose’ button in the ‘Classifier’ box just below the tabs and select C4.5 classifier WEKA □ Classifiers □ Trees □ J48.

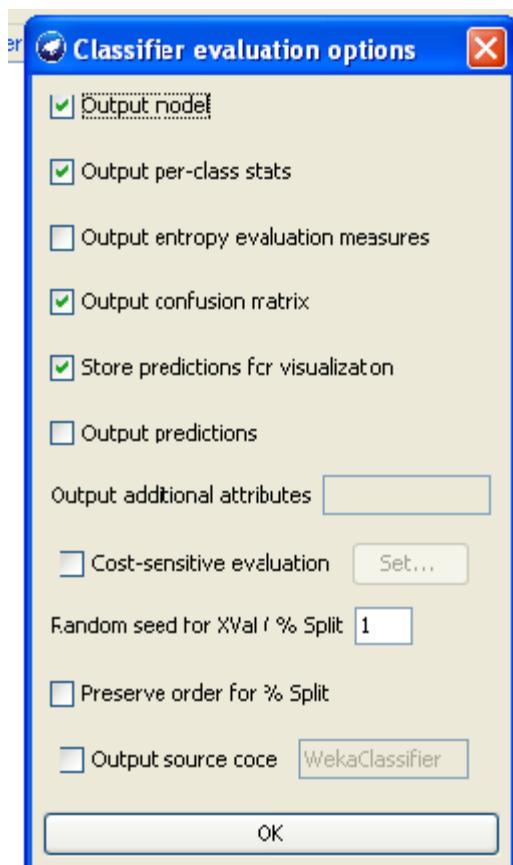


Check ‘Percentage split’ radio-button and keep it as default 66%. Click on ‘More options...’ button.

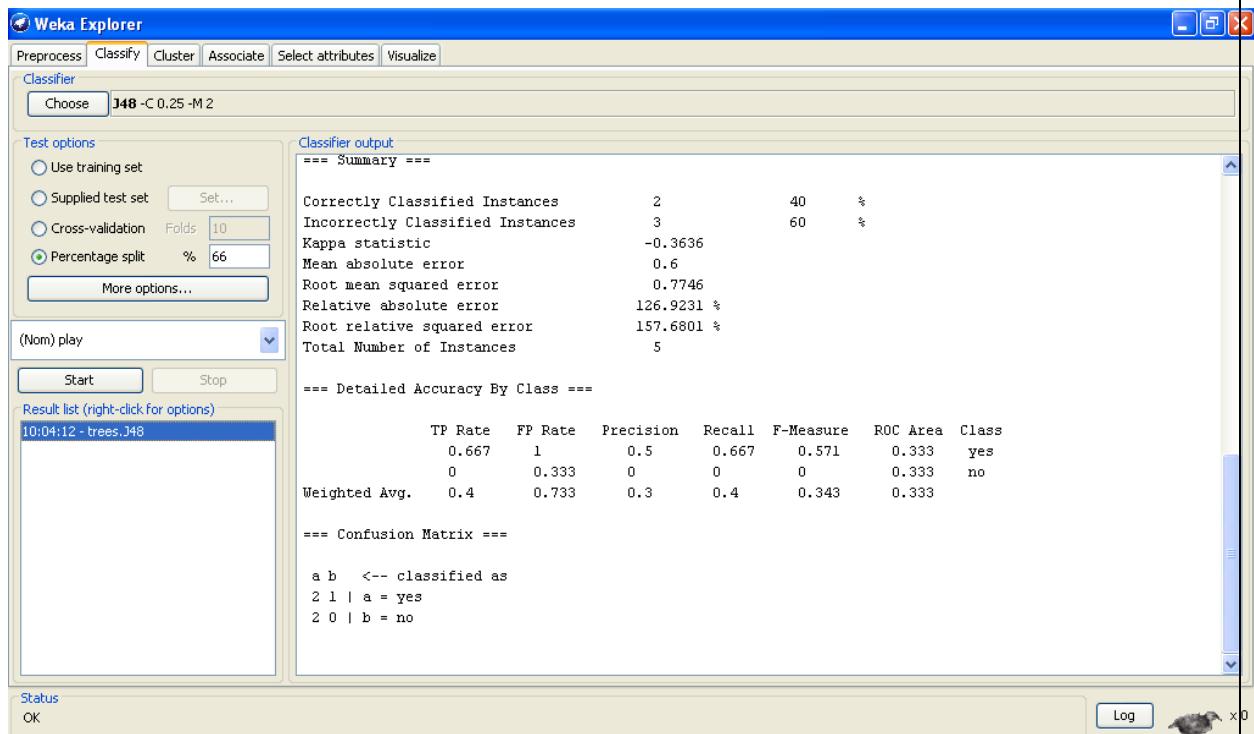
make sure that the Following options are checked :

1. Output model.

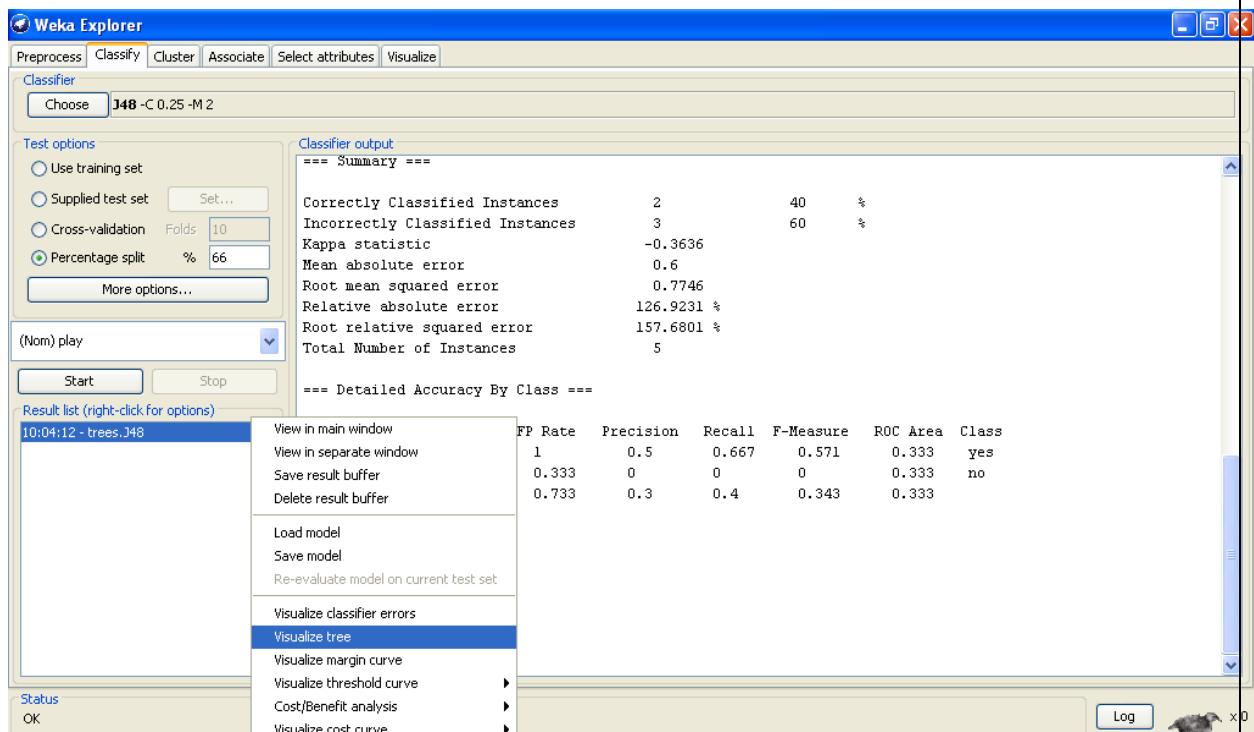
2. Output per-class stats.
3. Output confusion matrix
4. Store predictions for visualization.
5. Set ‘Random seed for Xval / % Split’ to 1.



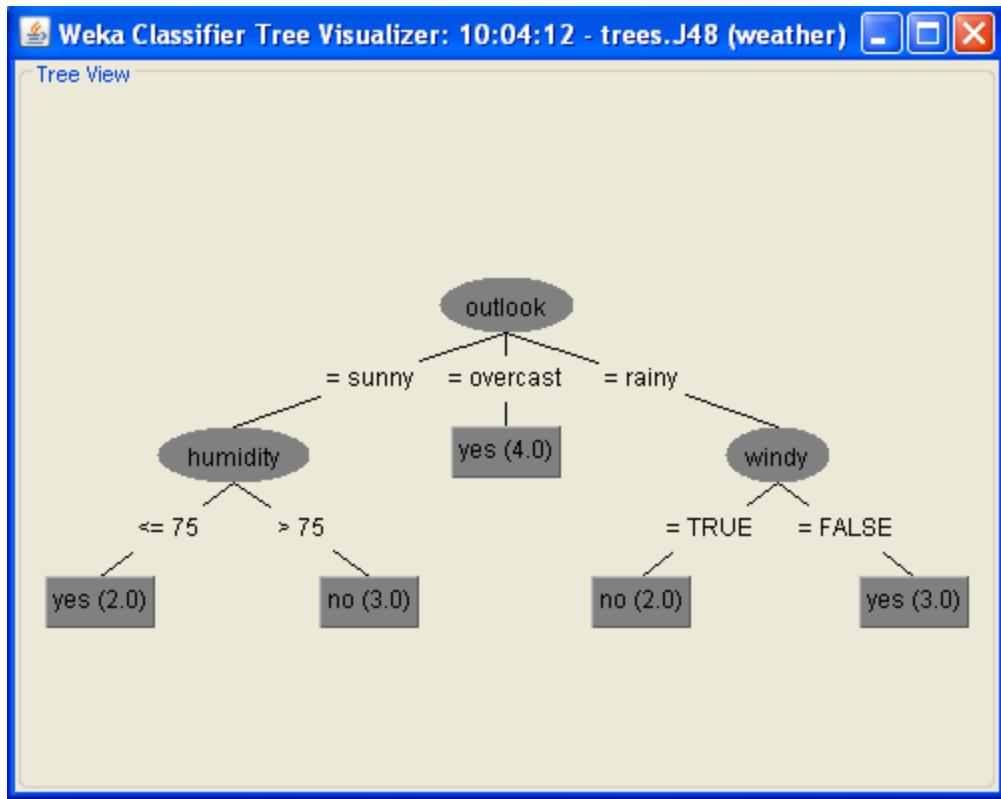
Once the options have been specified, you can run the classification algorithm. Click on ‘Start’ button



To see a graphical representation of the classification tree. Right-click on the entry in ‘Result list’ for which you would like to visualize a tree.

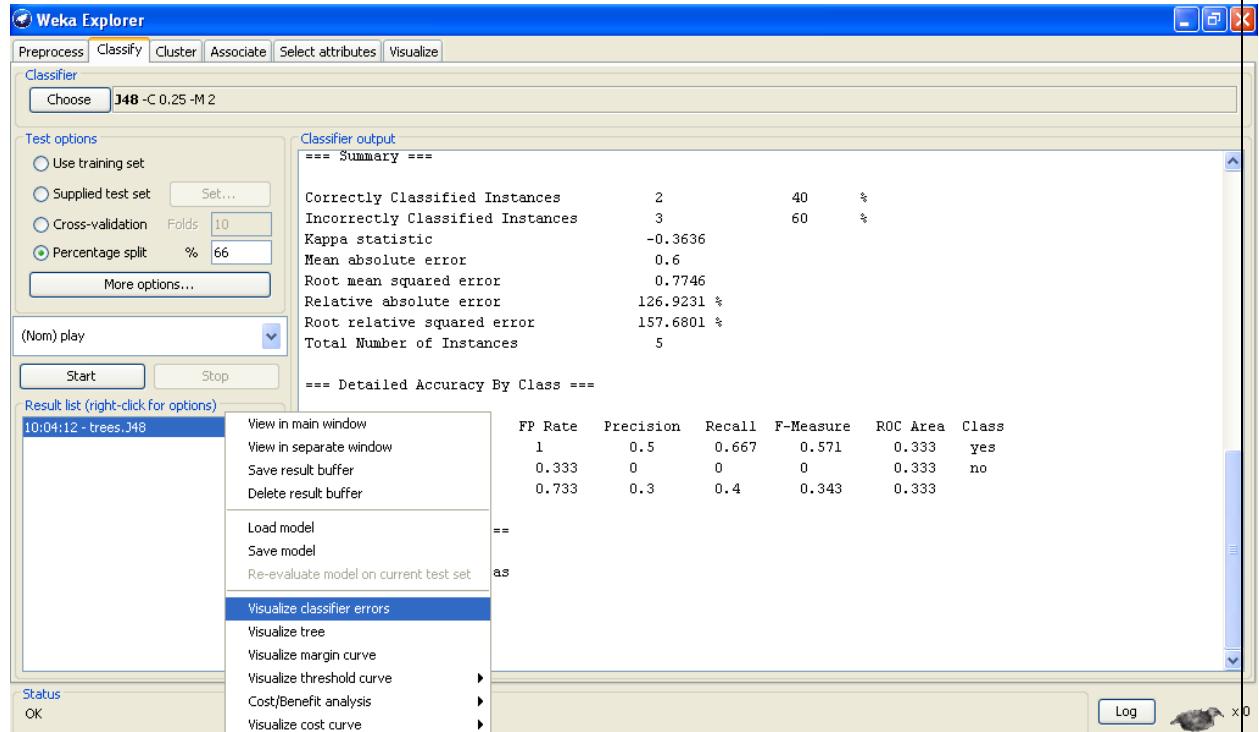


Select the item ‘Visualize tree’; a new window comes up to the screen displaying the tree.

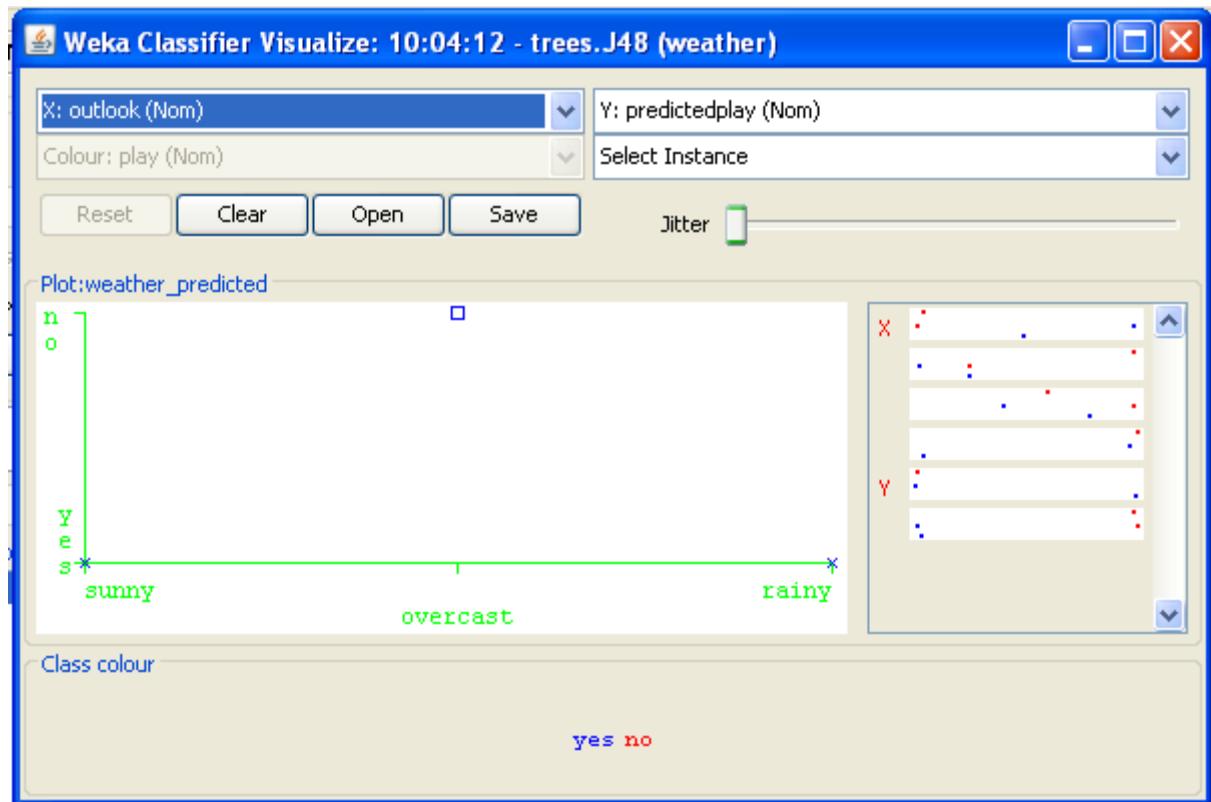


To visualize classification errors. Right-click on the entry in ‘Result list’ again

and select ‘Visualize classifier errors’ from the menu:



‘Visualize’ window displaying graph appears on the screen.



Practical N0 - 13

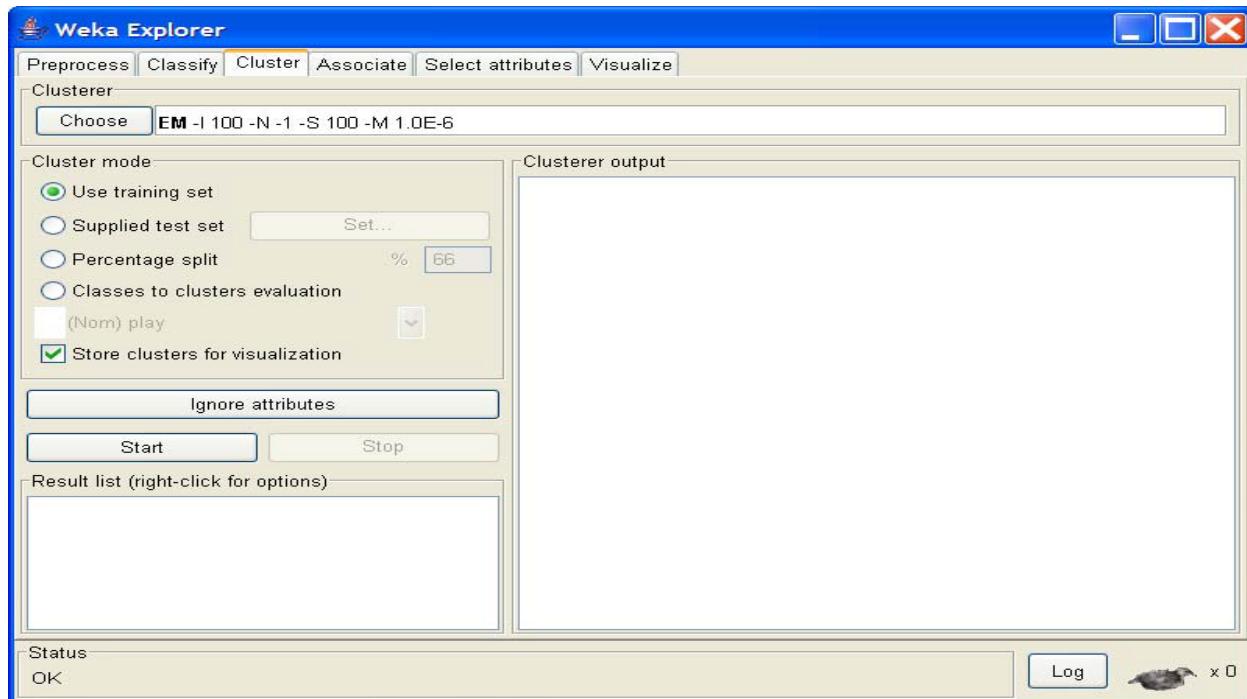
Aim : Clustering Analysis.

Solution :

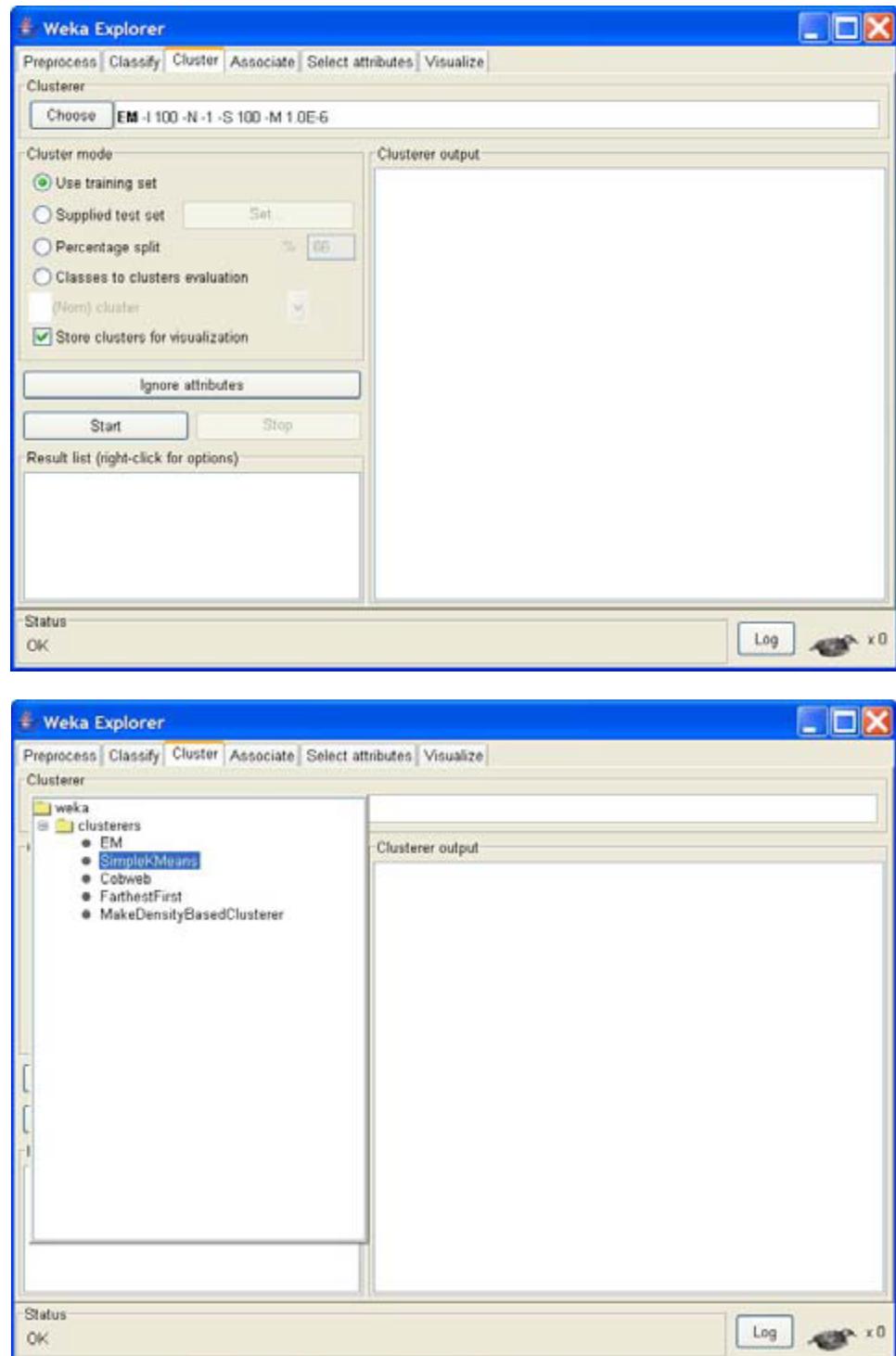
we will use customer data [6] that is contained in “customers.arff” file and analyze it with k-means clustering scheme.

	A	B	C	D	E
1	Income	Age	Children	Marital Status	Education
2					
3	25000	35	3	single	high school
4	15000	25	1	married	high school
5	20000	40	0	single	high school
6	30000	20	0	divorced	high school
7	20000	25	3	divorced	college
8	70000	60	0	married	college
9	90000	30	0	married	graduate school
10	200000	45	5	married	graduate school
11	100000	50	2	divorced	college
12					

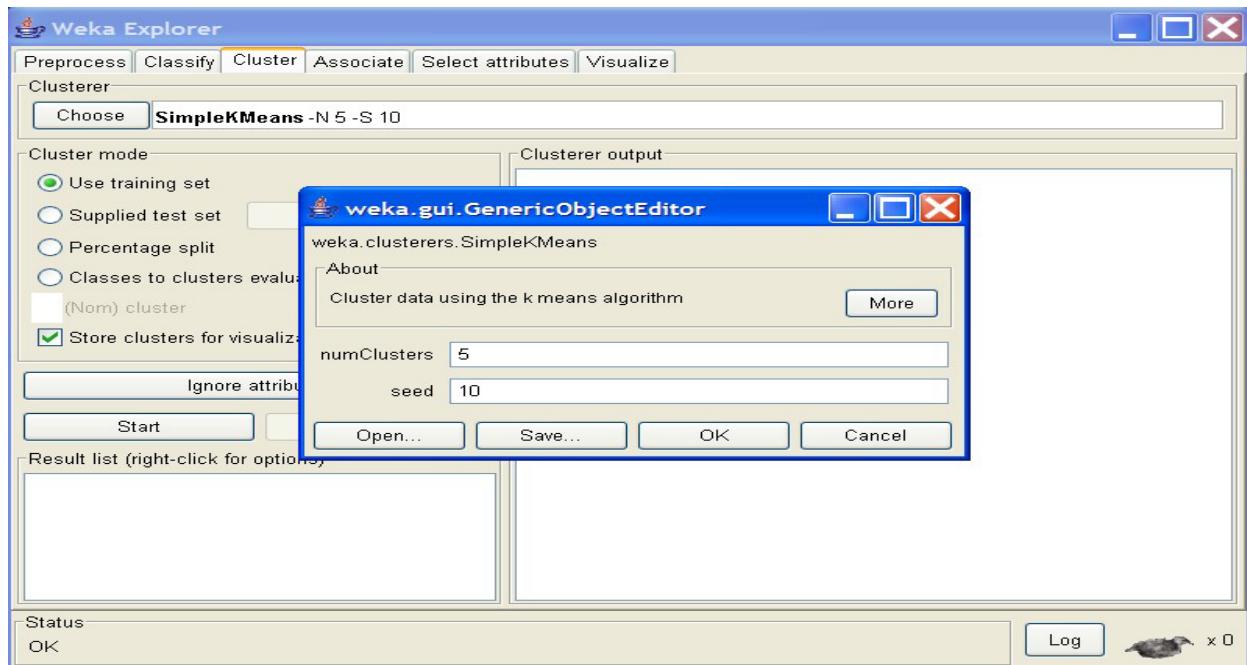
In ‘Preprocess’ window click on ‘Open file...’ button and select “customers.arff” file. Click ‘Cluster’ tab at the top of WEKA Explorer window.



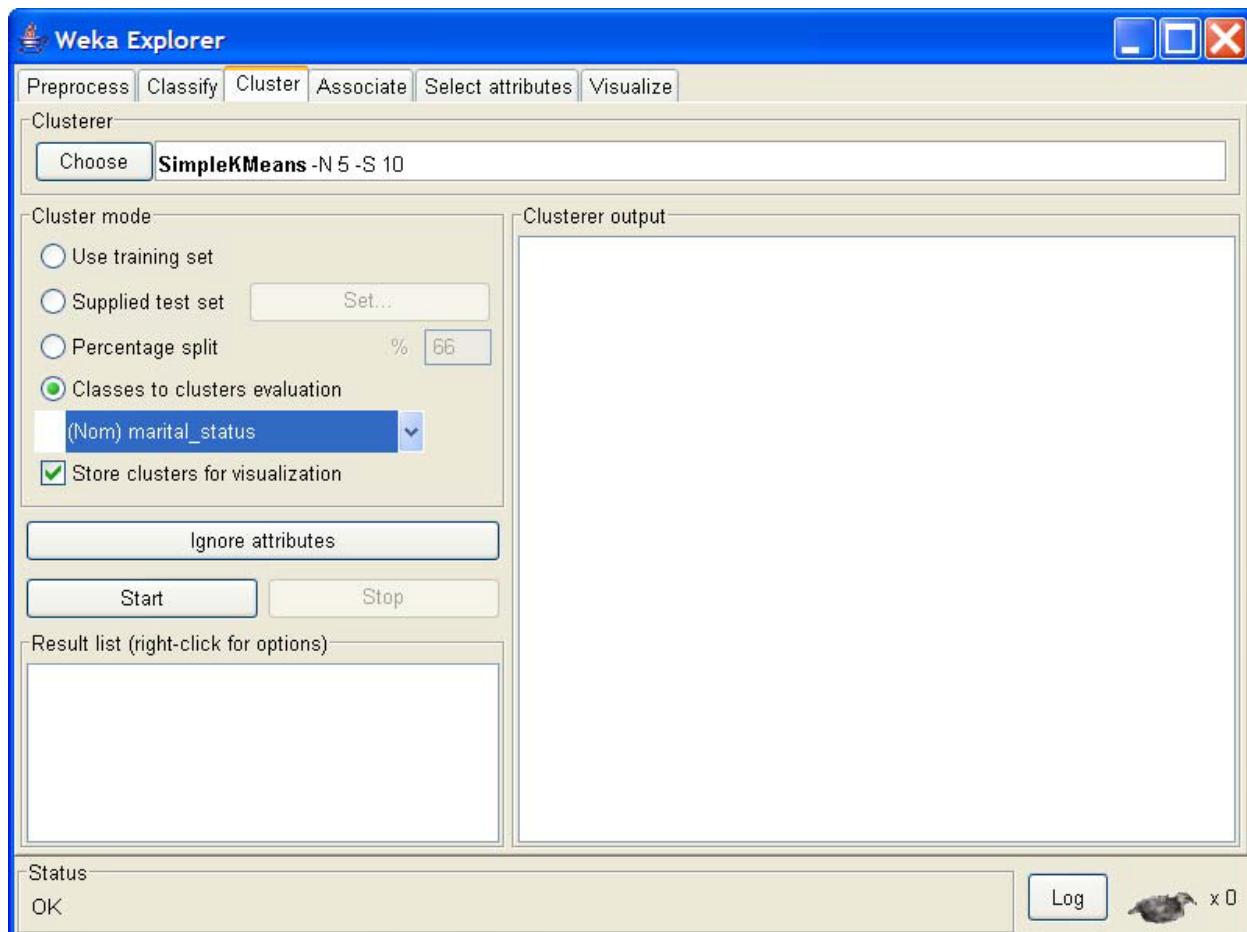
In the ‘Clusterer’ box click on ‘Choose’ button. In pull-down menu select WEKA □ Clusterers, and select the cluster scheme ‘SimpleKMeans’. Some implementations of K-means only allow numerical values for attributes.



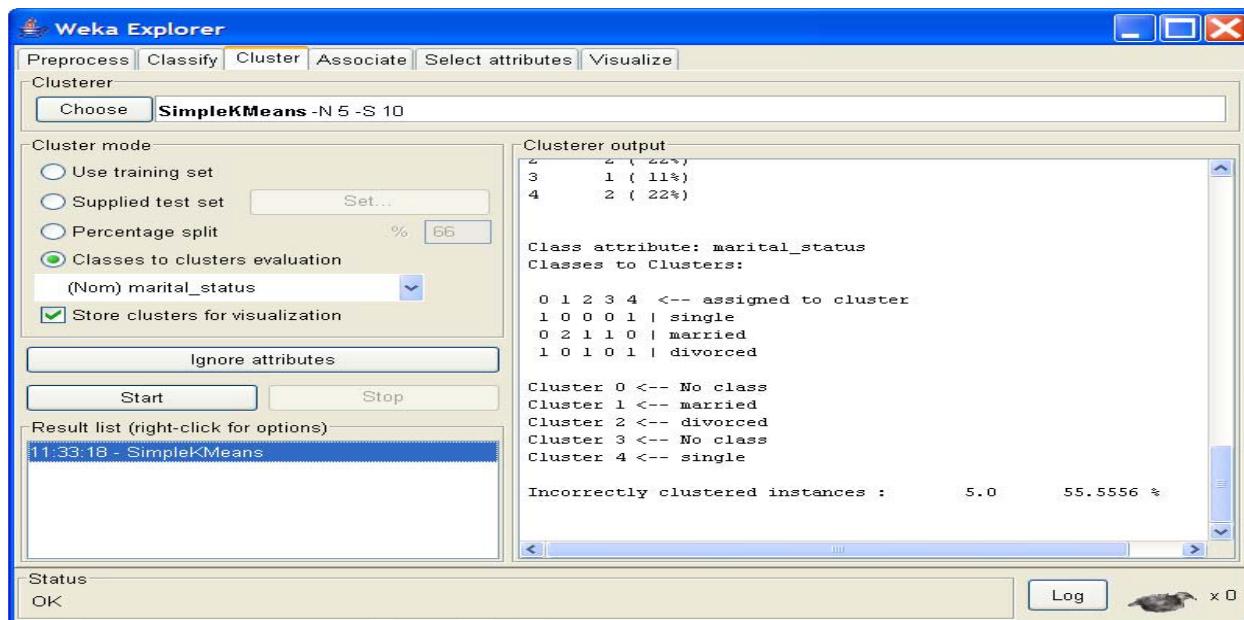
right-click on the algorithm “weak.gui.GenericObjectEditor” comes up to the screen. Set the value in “numClusters” box to 5(instead of default 2) because you have five clusters in your .arff file.



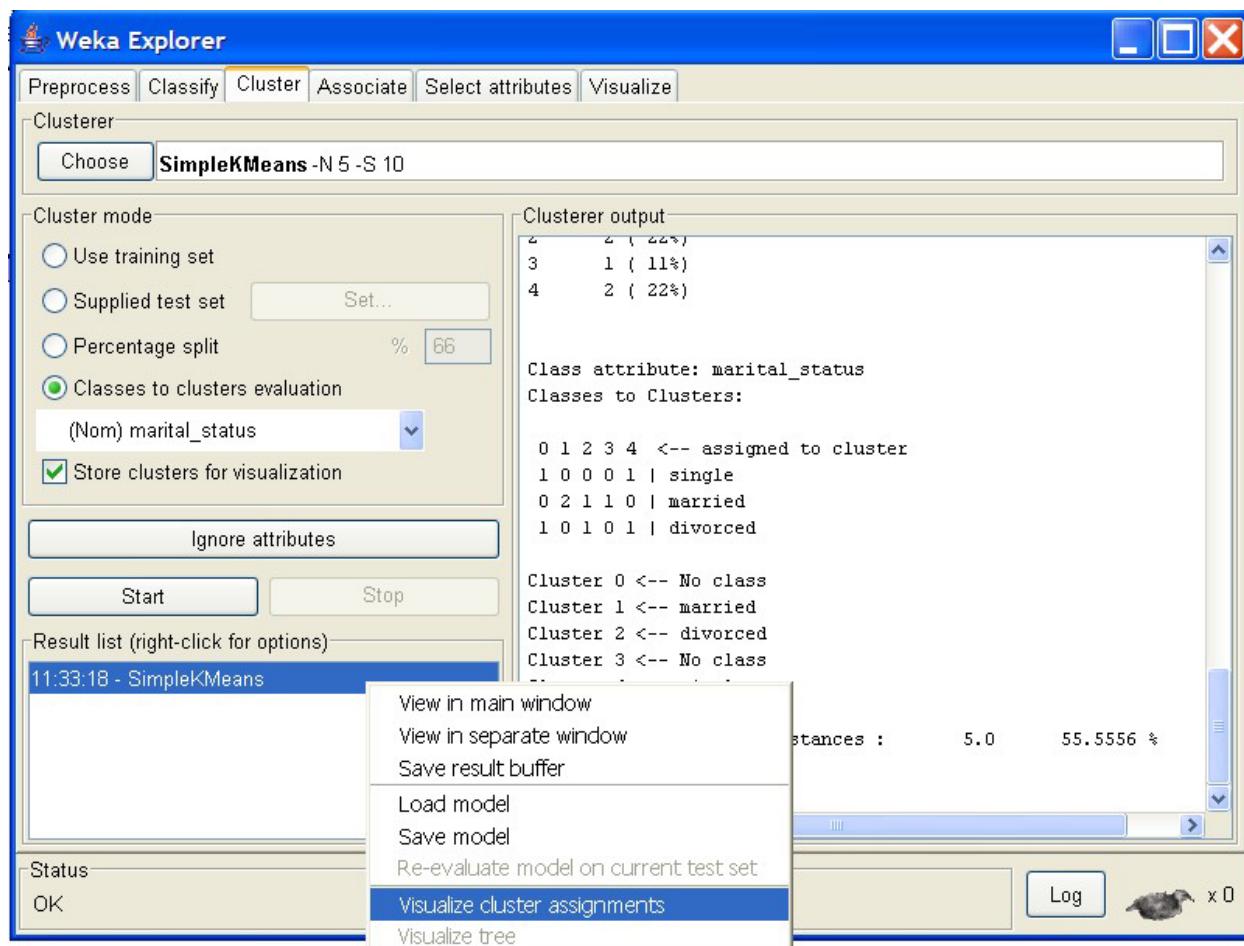
Click on ‘Classes to cluster evaluation’ radio-button in ‘Cluster mode’ box and select ‘marital_status’ in the pull-down box below.



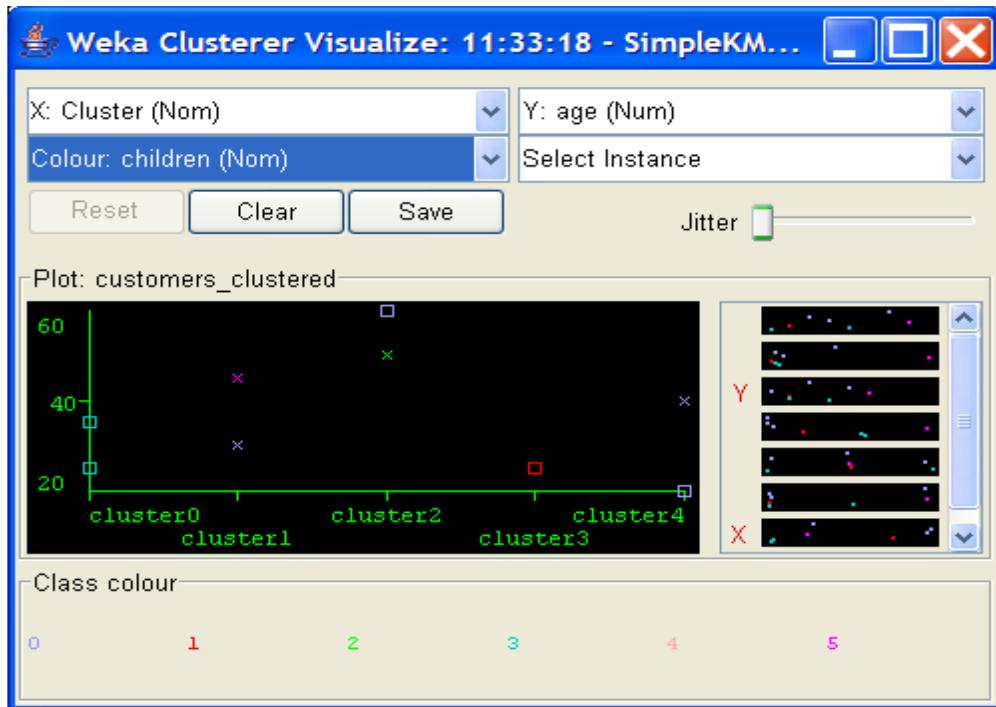
Click on the ‘Start’ button to execute the algorithm.



Right-click on the entry in the ‘Result list’ and select ‘Visualize cluster assignments’ in the pull-down window.



‘Weka Clusterer Visualize’ window.



there is a new attribute appeared in the file – ‘cluster’ that was added by WEKA. This attribute represents the clustering done by WEKA.

The screenshot shows a Microsoft Word document titled 'customers_kmeans - Microsoft Word'. The content of the document is an ARFF (Attribute-Relationship File Format) dataset. It starts with a header section defining attributes and their types, followed by a data section with 8 instances. The header includes attributes for 'Instance_number', 'income', 'age', 'children', 'marital_status', 'education', and 'Cluster'. The data section lists the following instances:

```

@relation customers_clustered

@attribute Instance_number numeric
@attribute income numeric
@attribute age numeric
@attribute children {0,1,2,3,4,5}
@attribute marital_status {single,married,divorced}
@attribute education {high_school,college,graduate_school}
@attribute Cluster {cluster0,cluster1,cluster2,cluster3,cluster4}

@data
0,25000,35,3,single,high_school,cluster0
1,15000,25,1,married,high_school,cluster3
2,20000,40,0,single,high_school,cluster4
3,30000,20,0,divorced,high_school,cluster4
4,20000,25,3,divorced,college,cluster0
5,70000,60,0,married,college,cluster2
6,90000,30,0,married,graduate_school,cluster1
7,200000,45,5,married,graduate_school,cluster1
8,100000,50,2,divorced,college,cluster2

```

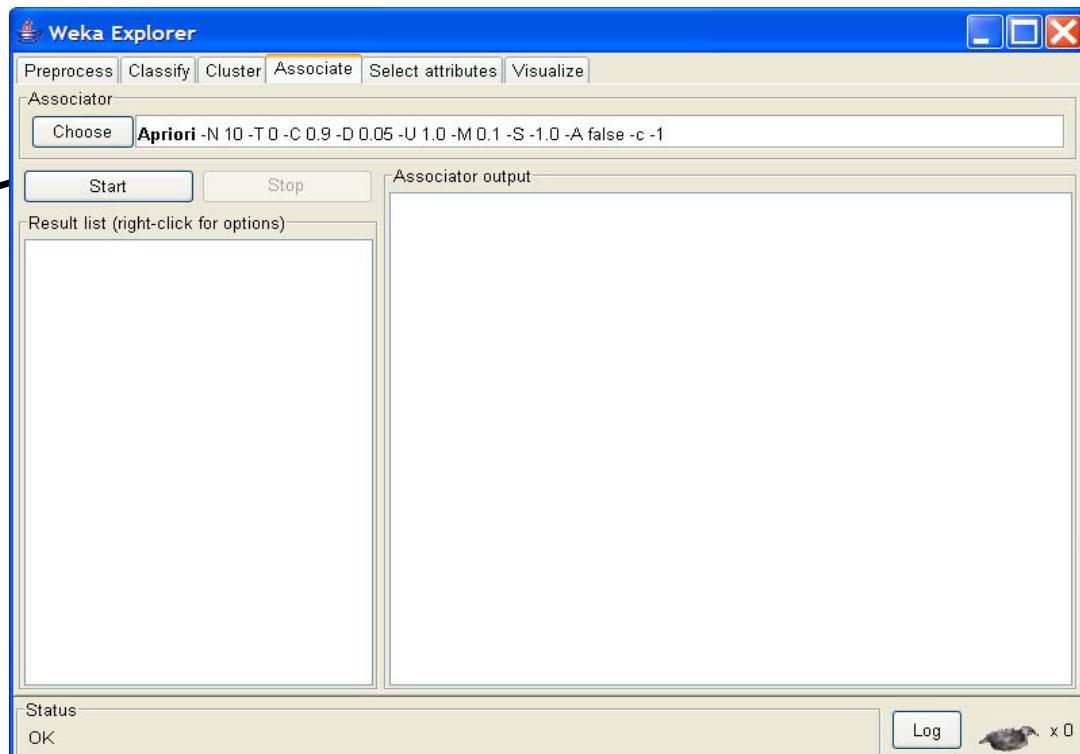
Practical No-14

Aim : Association Rule Mining.

Solution :

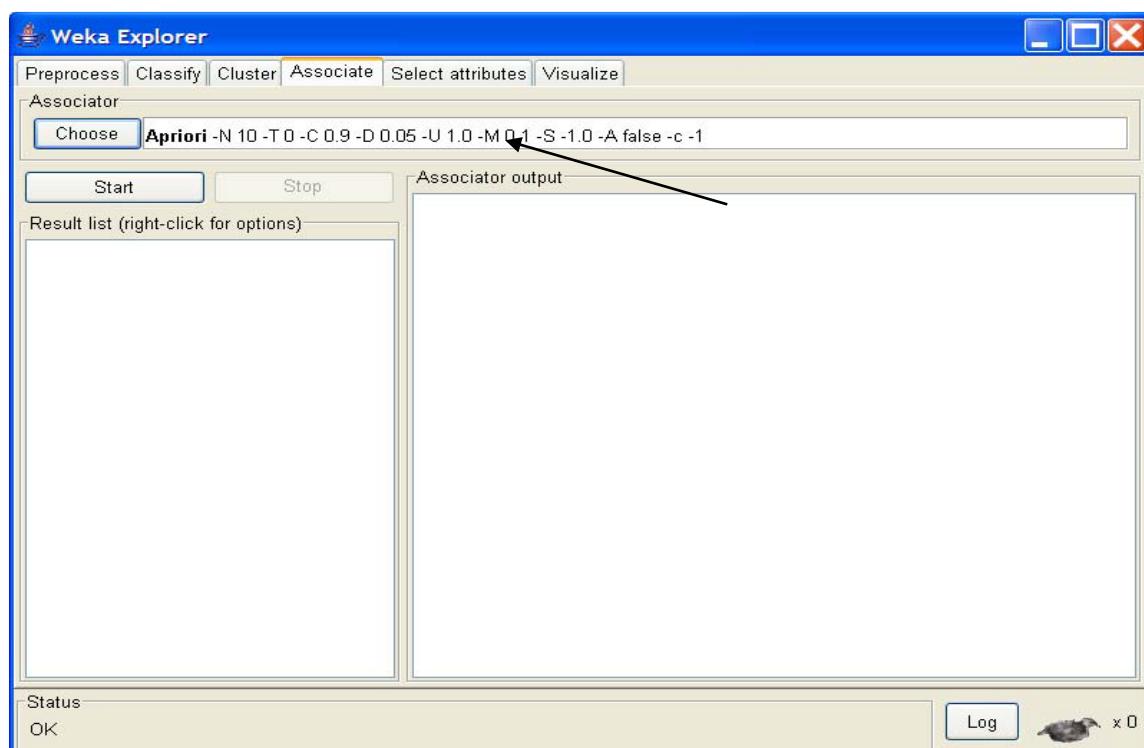
Choosing Association Scheme

Click ‘Associate’ tab at the top of ‘WEKA Explorer’ window. It brings up interface for the Apriori algorithm.

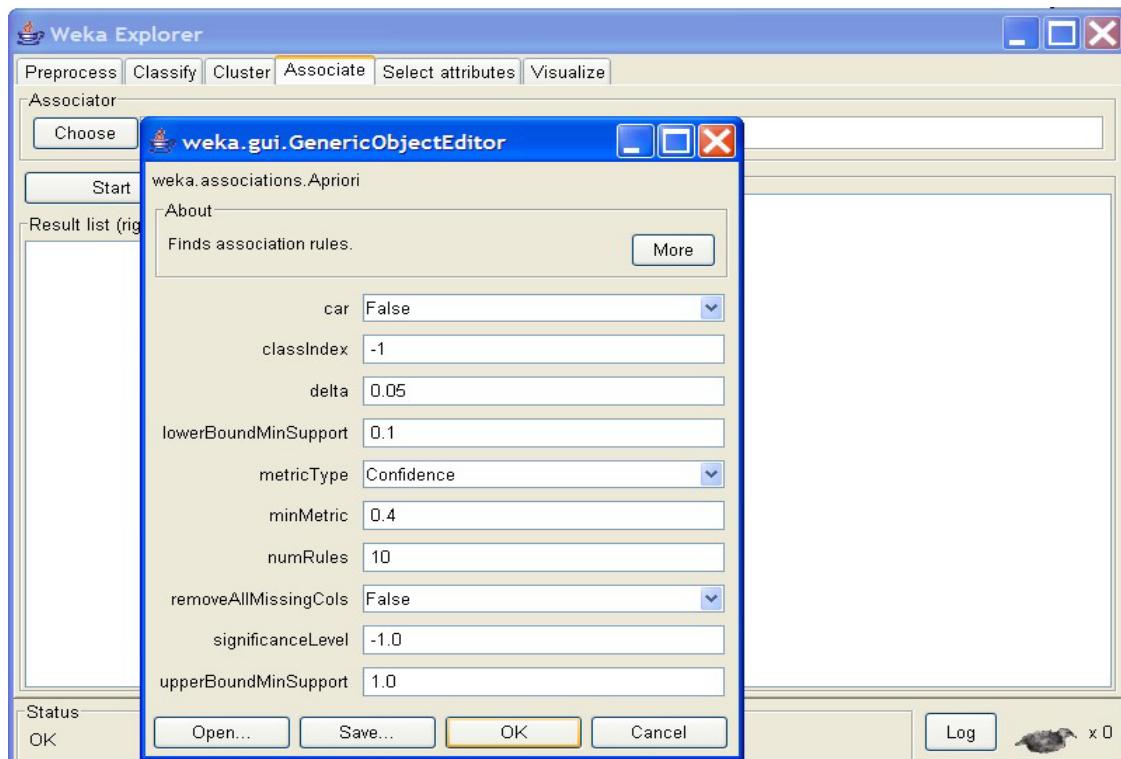


Setting Test Options

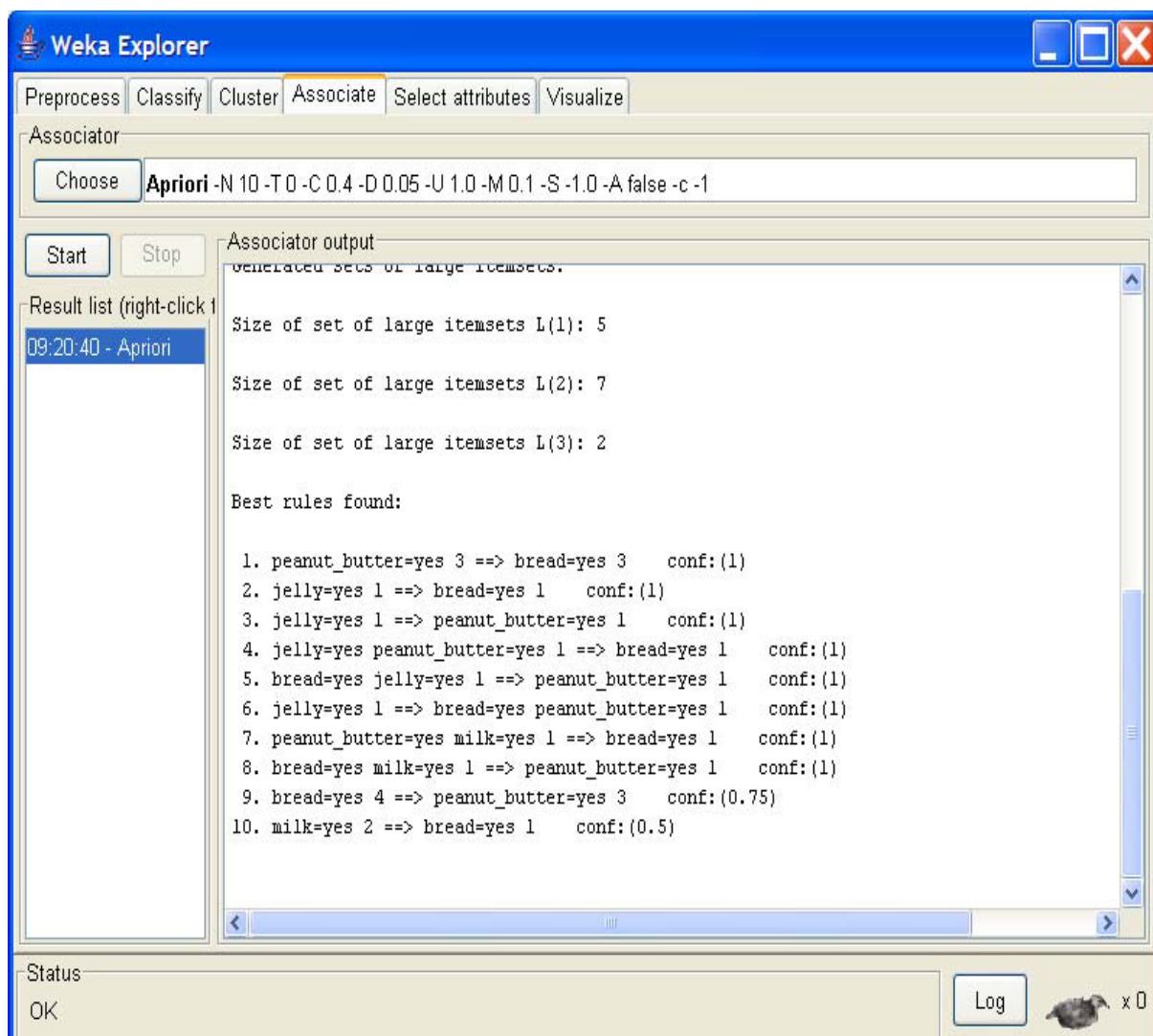
Check the text field in the ‘Associator’ box at the top of the window



Right-click on the ‘Associator’ box, ‘GenericObjectEditor’ appears on your screen



Click on the ‘Start’ button to execute the algorithm



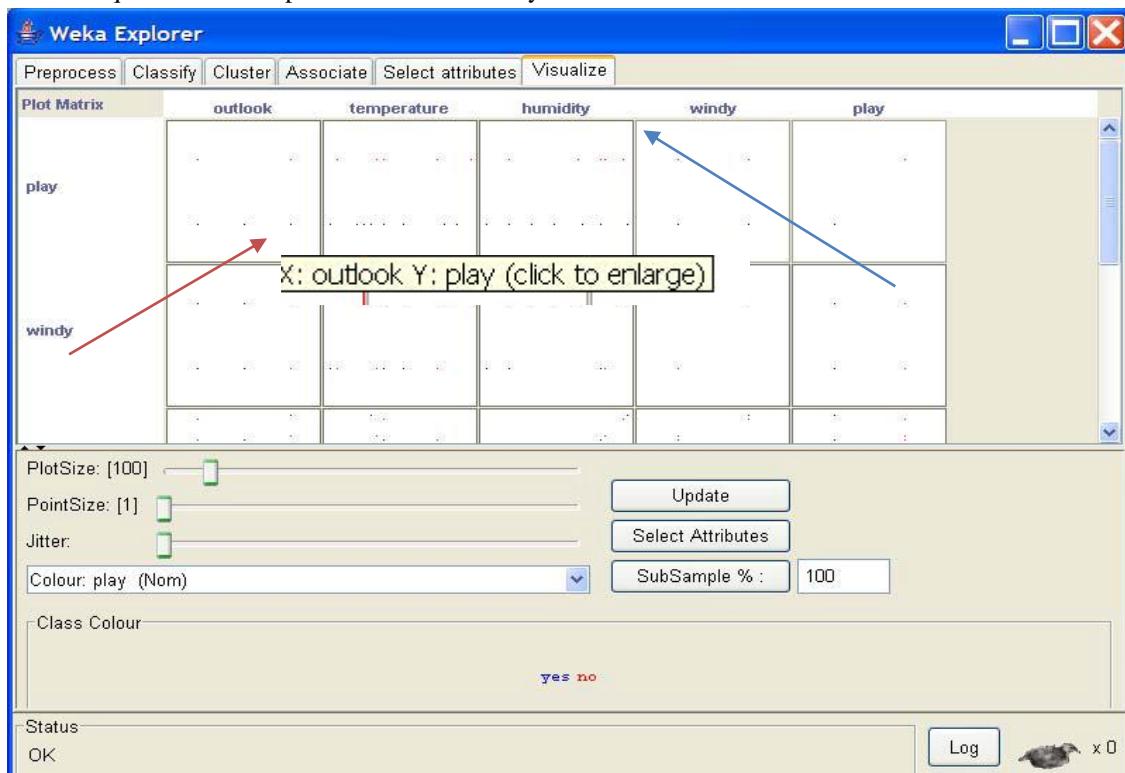
Practical No-15

Aim : Data Visualization

Solution :

To open Visualization screen, click ‘Visualize’ tab.

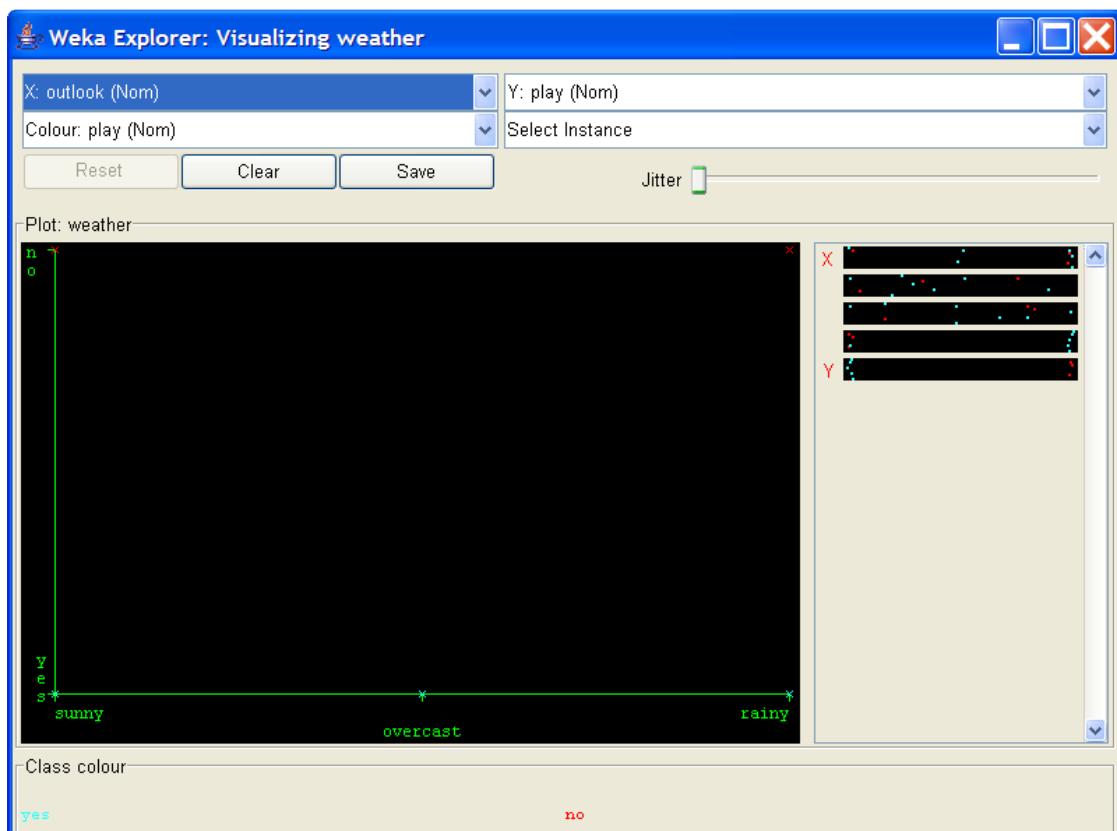
Select a square that corresponds to the attributes you would like to visualize.



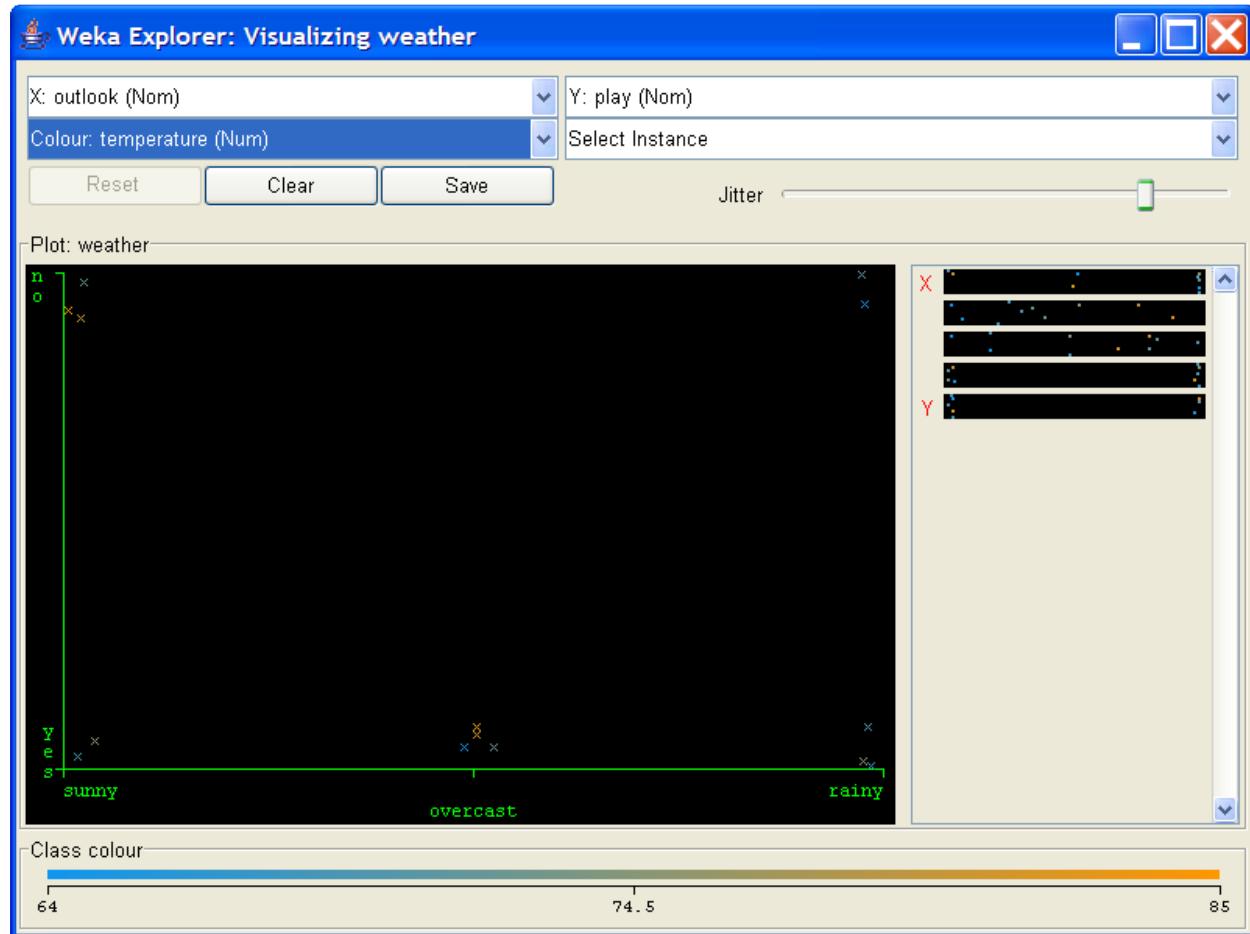
A ‘Visualizing weather’ window appears on the screen.

Changing the View

Keep sliding ‘Jitter’, a random



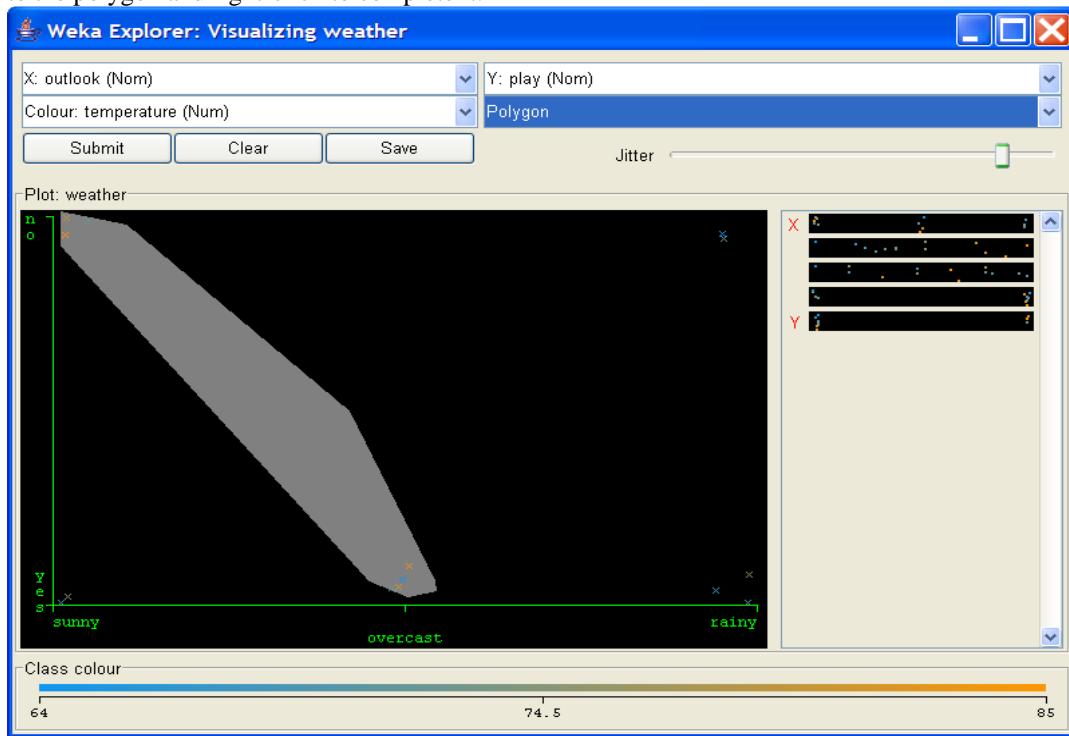
displacement given to all points in the plot, to the right, until you can spot concentration points



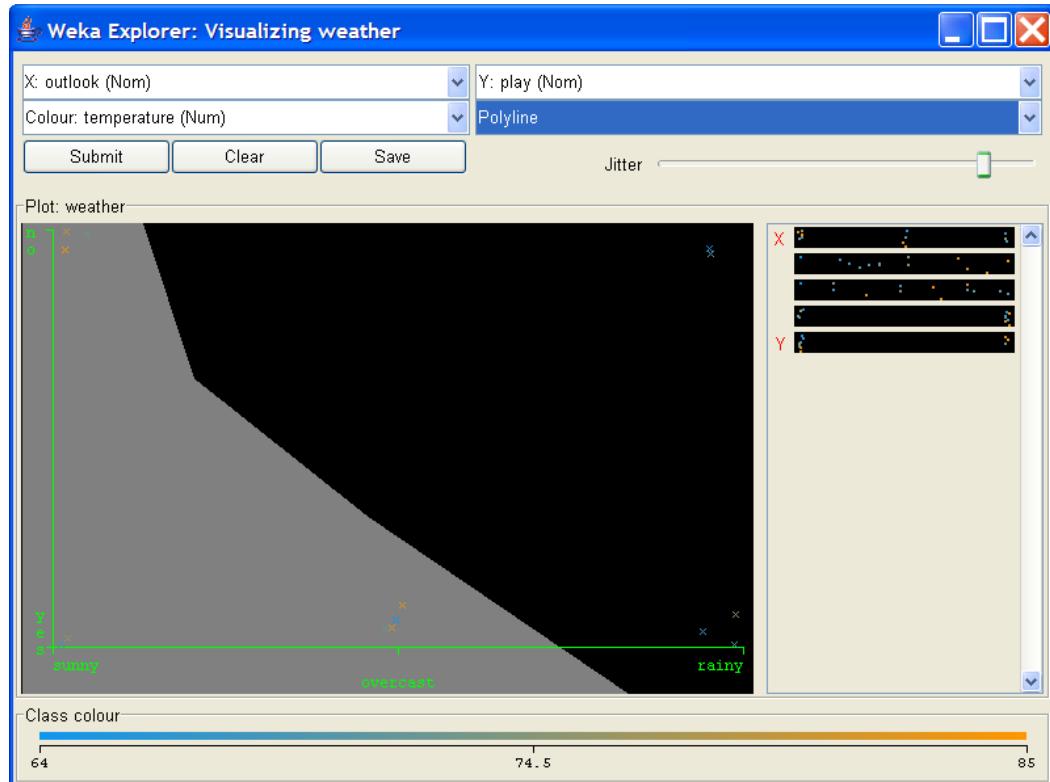
Selecting Instances: Click on an individual data point



3. **Polygon.** You can select several points by building a free-form polygon. Left-click on the graph to add vertices to the polygon and right-click to complete it.



4. **Polyline.** To distinguish the points on one side from the ones on another, you can build a polyline. Left-click on the graph to add vertices to the polyline and right-click to finish.



Rectangle. You can create a rectangle by dragging it around the points

