BUSINESS INTELLIGENCE & DATA ANALYSIS

PRACTICALS

TY BSc.IT

SEMESTER 6

NAME: ALISHA SHAIKH

ROLL NO: B20222507

COLLEGE: SASMIRA'S INSTITUTE OF COMMERCE & SCIENCE

YEAR: 2025

INDEX

Sr. No	Name	Date	Signature
1	Perform the analysis for the following:		
a	Import the data warehouse 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.		
2	Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data. Use Excel.		
3	Perform the data classification using classification algorithm using R/Python.		
4	Perform the data clustering using clustering algorithm using R/Python.		
5	Perform the Linear regression on the given data warehouse data using R/Python.		
6	Perform the logistic regression on the given data warehouse data using R/Python.		
7	Write a Python program to read data from a CSV file, perform simple data analysis, and generate basic insights. (Use Pandas is a Python library).		
8	Create the Data staging area for the selected database using SQL.		

Practical No.1

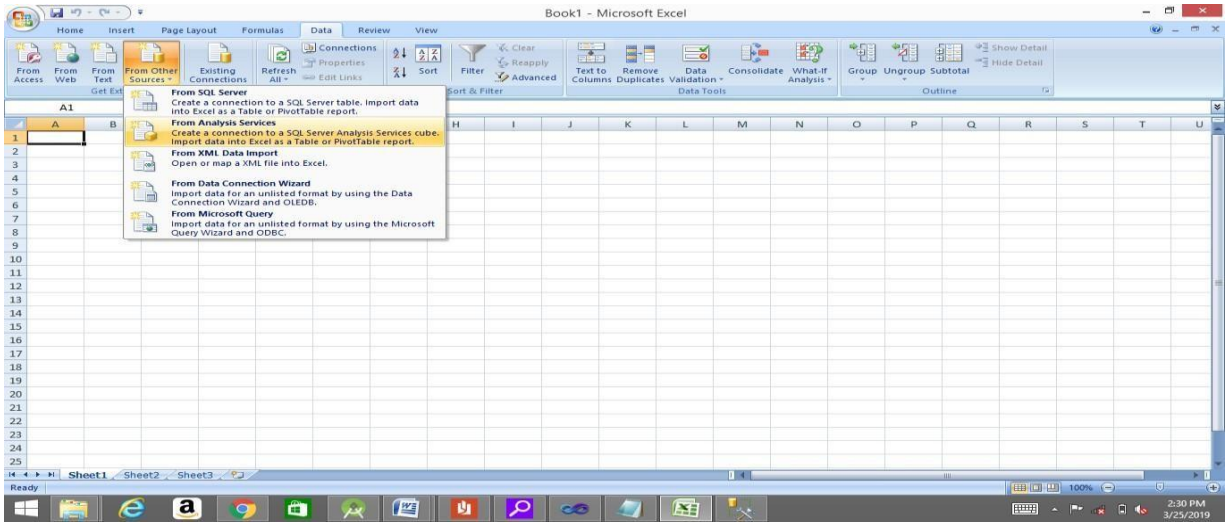
a. Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.
b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to Perform data analysis.

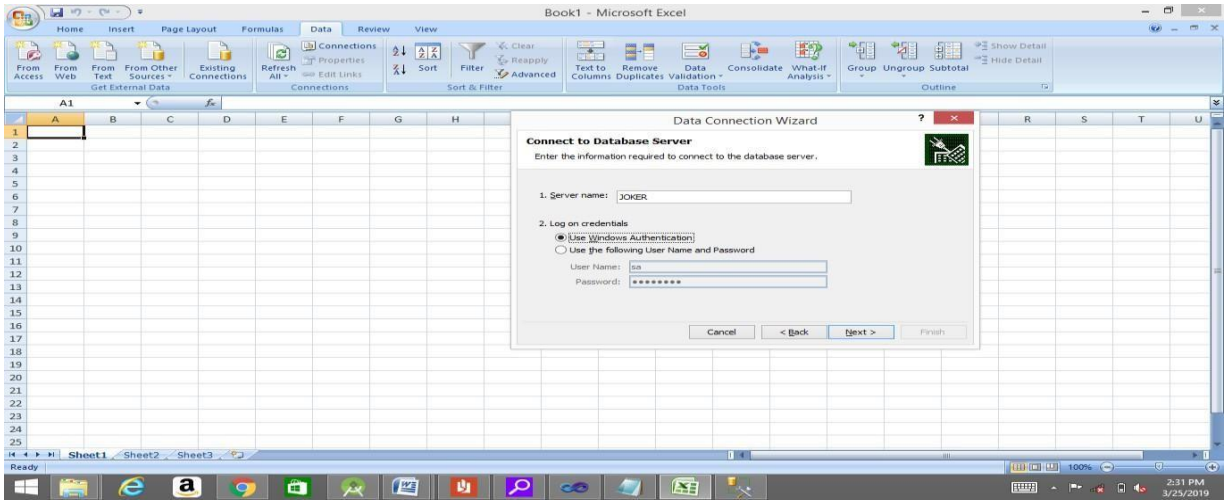
a. Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.

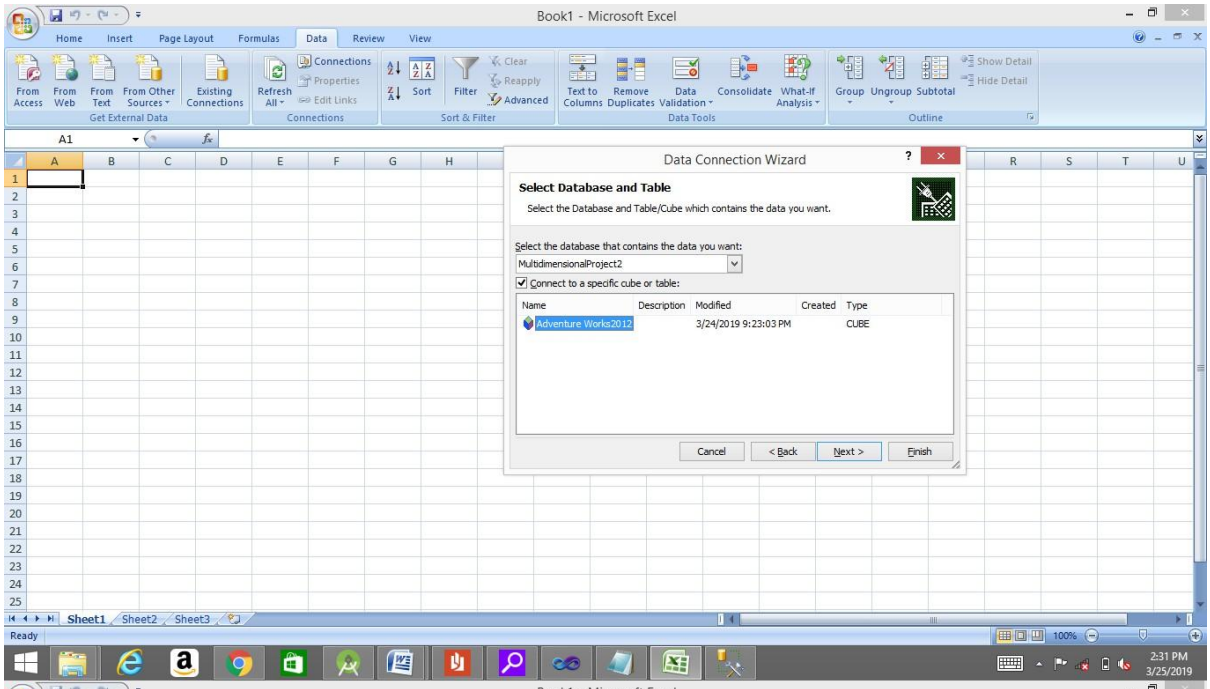
Pivot Tables

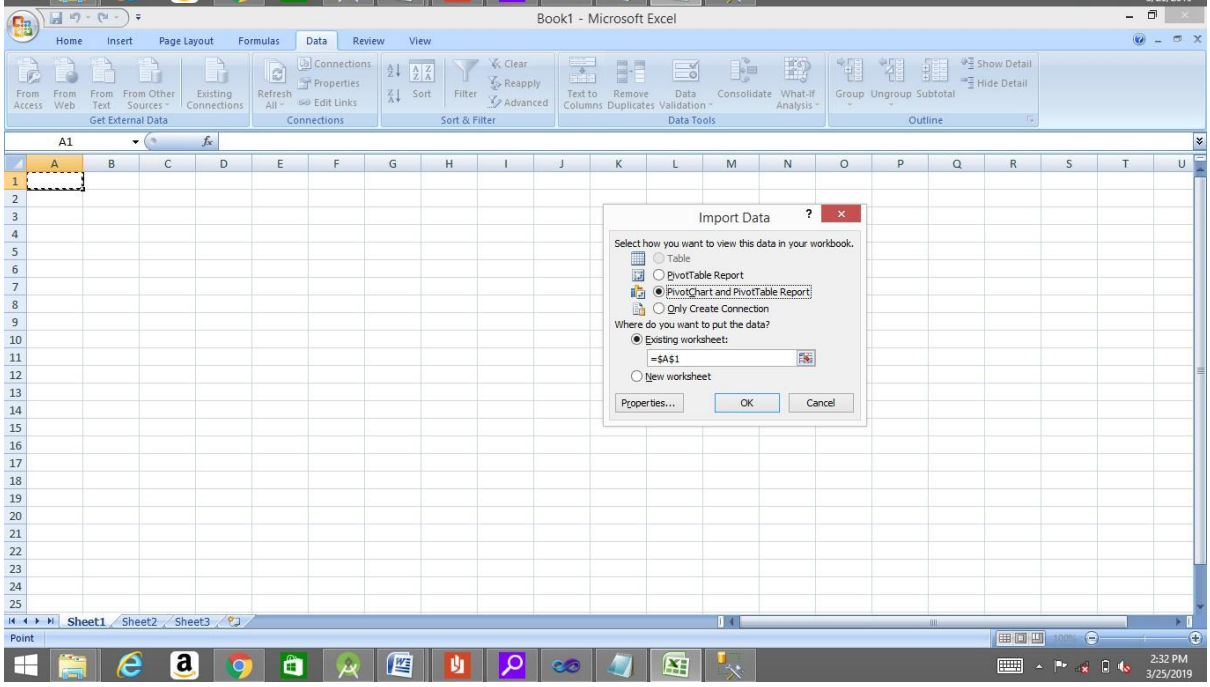
- **Pivot Tables** allow you to create a powerful view with data summarized **in a** grid, both **in** horizontal and vertical columns (also known as Matrix Views or Cross Tabs)
- A pivot chart is the visual representation of a pivot table in Excel. Pivot charts and pivot tables are connected with each other.

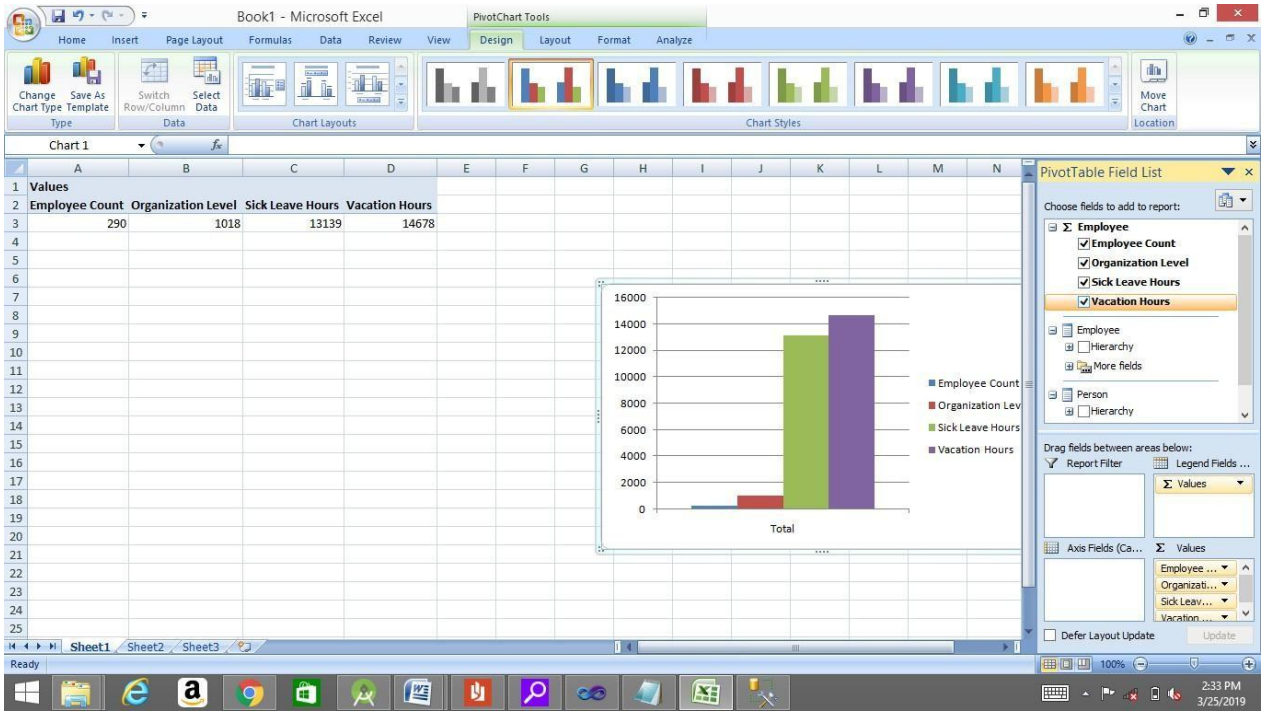
To create Pivot table and Pivot Chart Open Microsoft Excel




Book1 - Microsoft Excel
Data Connection Wizard
Connect to Database Server
Enter the information required to connect to the database server.
1. Server name:
JOKER
2. Log on credentials
Use Windows Authentication
Use the following User Name and Password
User Name:
sa
Password:
Cancel
< Back
Next >
Finish
Sheet1
Sheet2
Sheet3
Ready


Book1 - Microsoft Excel
Data Connection Wizard
Select Database and Table
Select the Database and Table/Cube which contains the data you want.
Select the database that contains the data you want:
MultidimensionalProject2
Connect to a specific cube or table:
Name
Description
Modified
Created
Type
Adventure Works2012
3/24/2019 9:23:03 PM
CUBE
Cancel
< Back
Next >
Finish
Sheet1
Sheet2
Sheet3
Ready
2:31 PM
3/25/2019


Book1 - Microsoft Excel
Import Data
Select how you want to view this data in your workbook.
Table
PivotTable Report
PivotChart and PivotTable Report
Only Create Connection
Where do you want to put the data?
Existing worksheet:
=$A$1
New worksheet
Properties...
OK
Cancel
Sheet1
Sheet2
Sheet3
Point
2:32 PM
3/25/2019



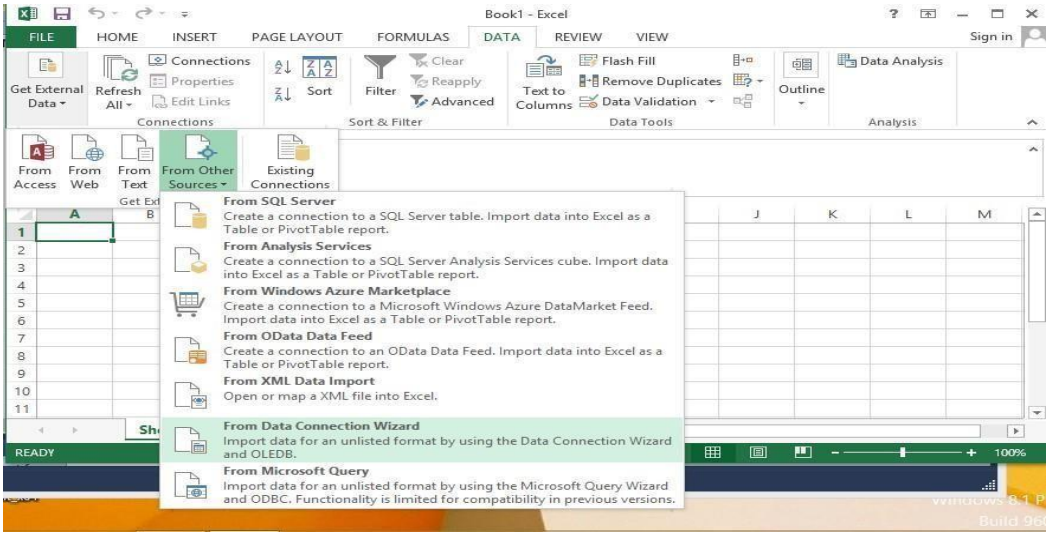
b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to Perform data analysis.

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

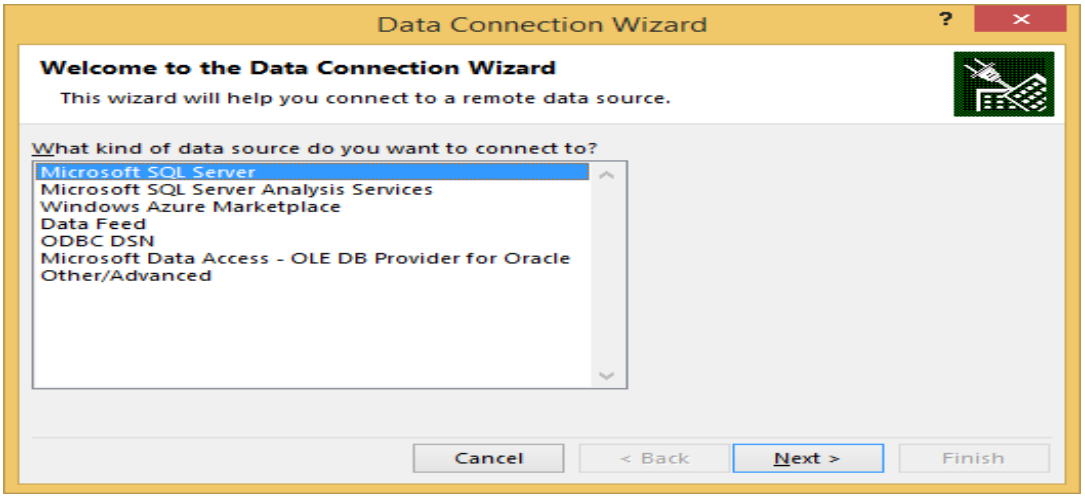
(Ms Office Professional is used to make sure Power View is enabled for visualization.)

Step 1: Open Excel 2013 (Professional)

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



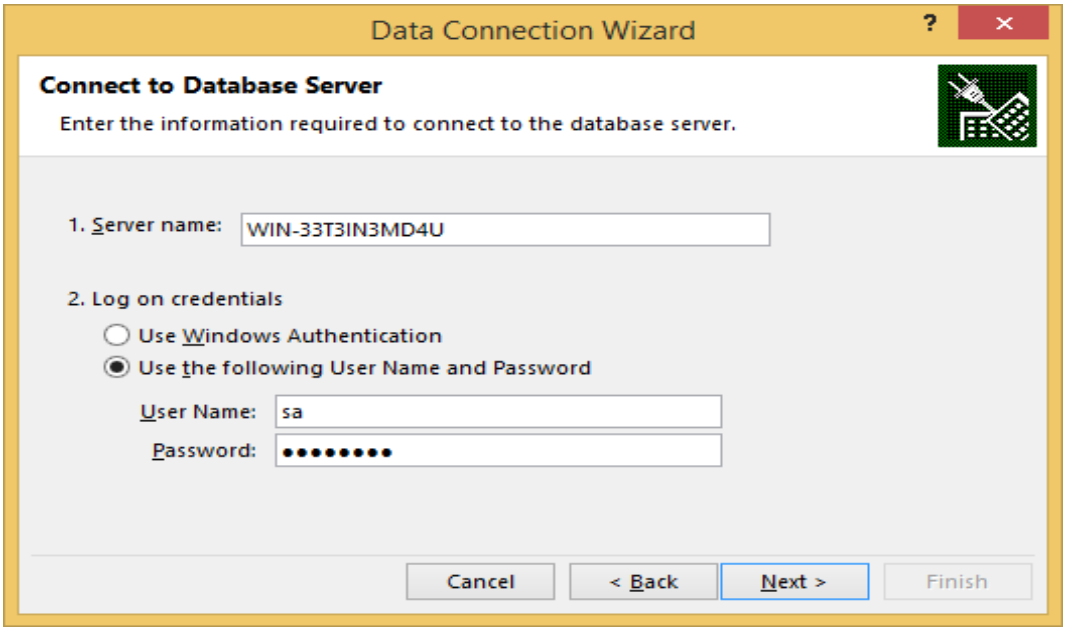
Step 2: In Data Connection Wizard → Select Microsoft SQL Server → Click on Next



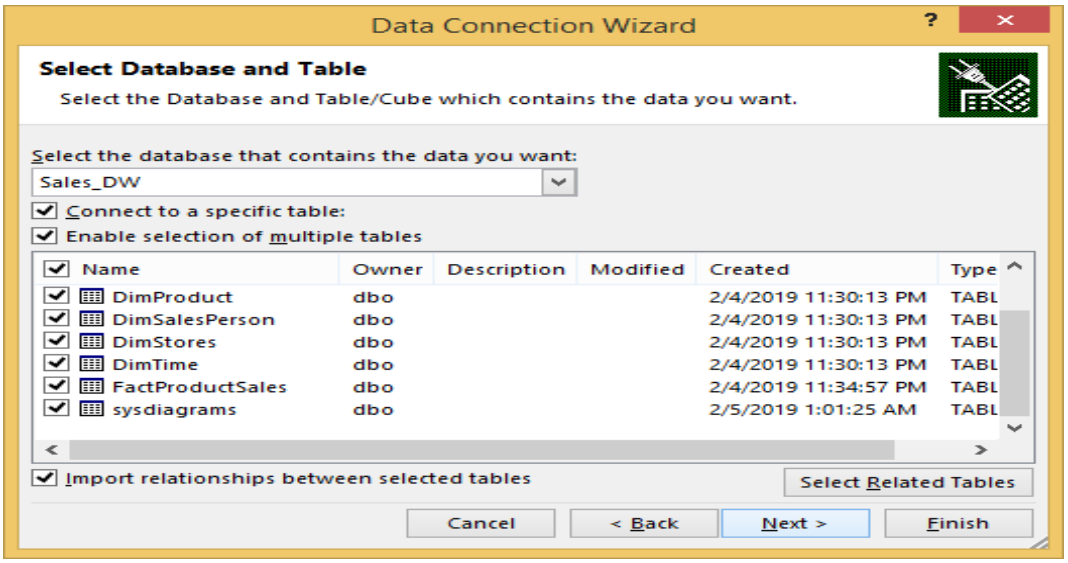
Step 3: In connect to Database Server provide Server name(Microsoft SQL Server Name)

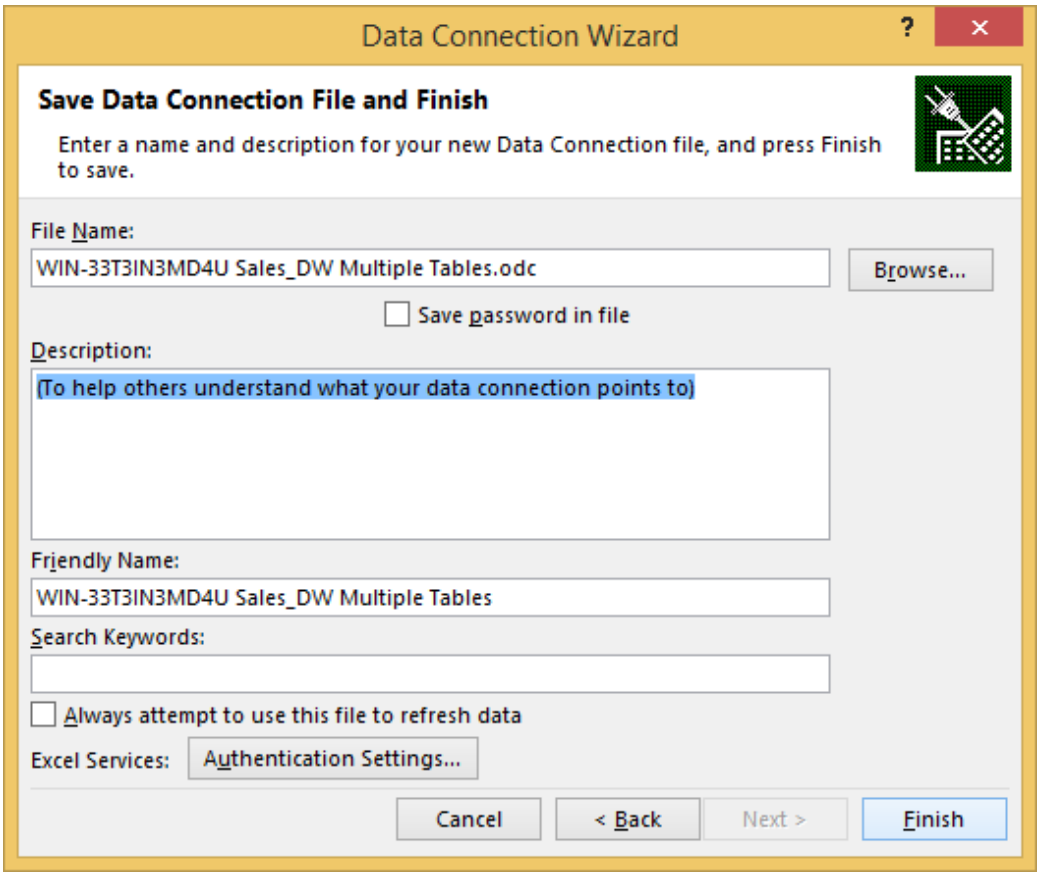
Provide password for sa account as given during installation of SQL Server 2012 full version)

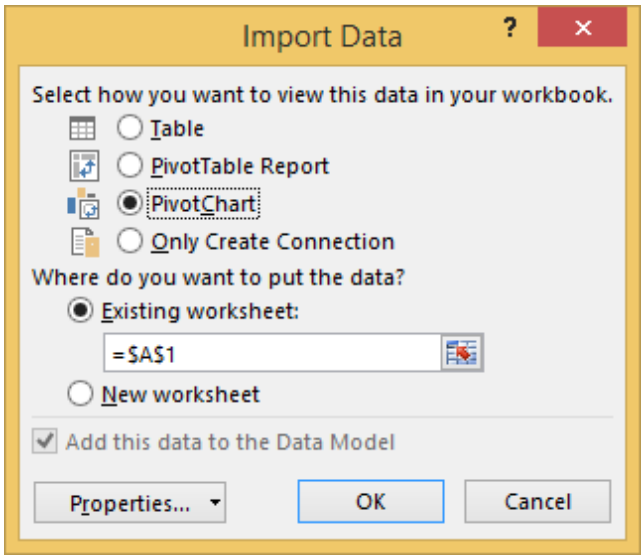
Password: admin123 Click on Next



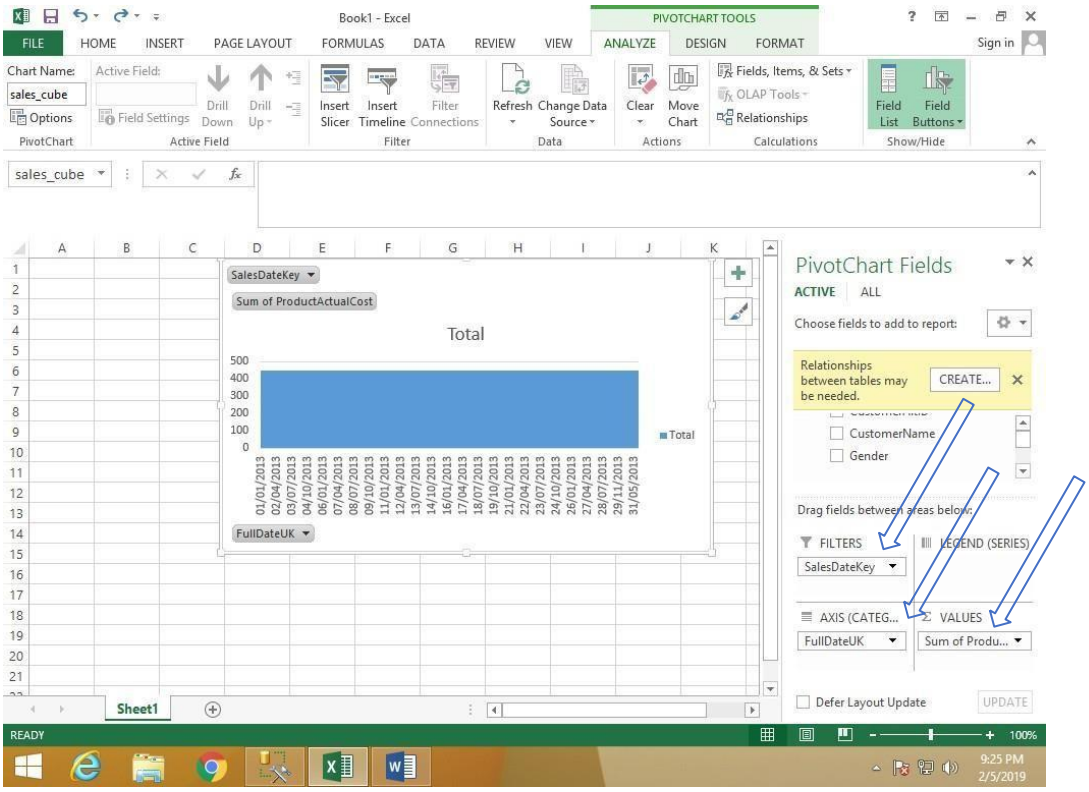
Step 4: In Select Database and Table→ Select Sales_DW (already created in SQL) → check all dimensions and import relationships between selected tables


Data Connection Wizard
Select Database and Table
Select the Database and Table/Cube which contains the data you want.
Select the database that contains the data you want:
Sales_DW
Connect to a specific table:
Enable selection of multiple tables
Name
Owner
Description
Modified
Created
Type
DimProduct dbo 2/4/2019 11:30:13 PM TABL
DimSalesPerson dbo 2/4/2019 11:30:13 PM TABL
DimStores dbo 2/4/2019 11:30:13 PM TABL
DimTime dbo 2/4/2019 11:30:13 PM TABL
FactProductSales dbo 2/4/2019 11:34:57 PM TABL
sysdiagrams dbo 2/5/2019 1:01:25 AM TABL
Import relationships between selected tables
Select Related Tables
Cancel
< Back
Next >
Finish

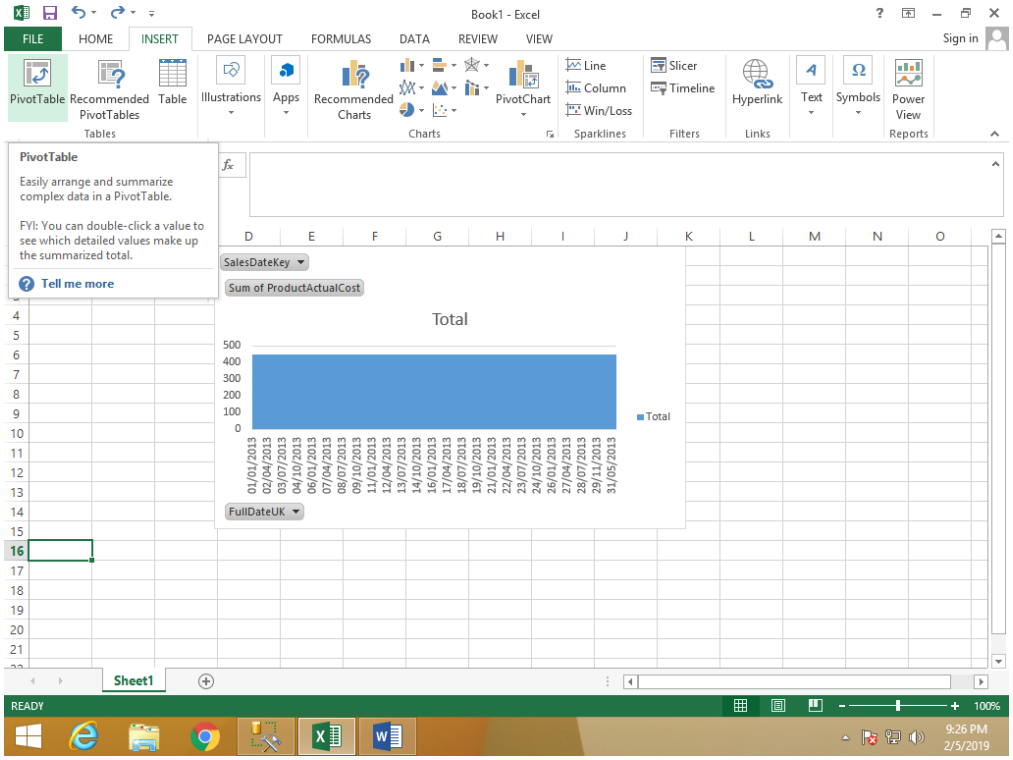

Data Connection Wizard
Save Data Connection File and Finish
Enter a name and description for your new Data Connection file, and press Finish to save.
File Name:
WIN-33T3IN3MD4U Sales_DW Multiple Tables.odc
Browse...
Save password in file
Description:
(To help others understand what your data connection points to)
Friendly Name:
WIN-33T3IN3MD4U Sales_DW Multiple Tables
Search Keywords:
Always attempt to use this file to refresh data
Excel Services:
Authentication Settings...
Cancel
< Back
Next >
Finish



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



Practical No.2

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

What-If Analysis

- **What-If Analysis** is the process of changing the values in cells to see how those changes will affect the outcome of formulas on the worksheet.
- What-If Analysis in Excel allows you to try out different values (scenarios) for formulas.
- Three kinds of **What-If Analysis** tools come with **Excel**:
- Scenarios, Goal Seek, and Data Tables.
- Scenarios and Data tables take sets of input values and determine possible results.

Goal Seek

- If you know the result that you want from a formula, but you are not sure what input value the formula requires to get that result, you can use the **Goal Seek** feature.

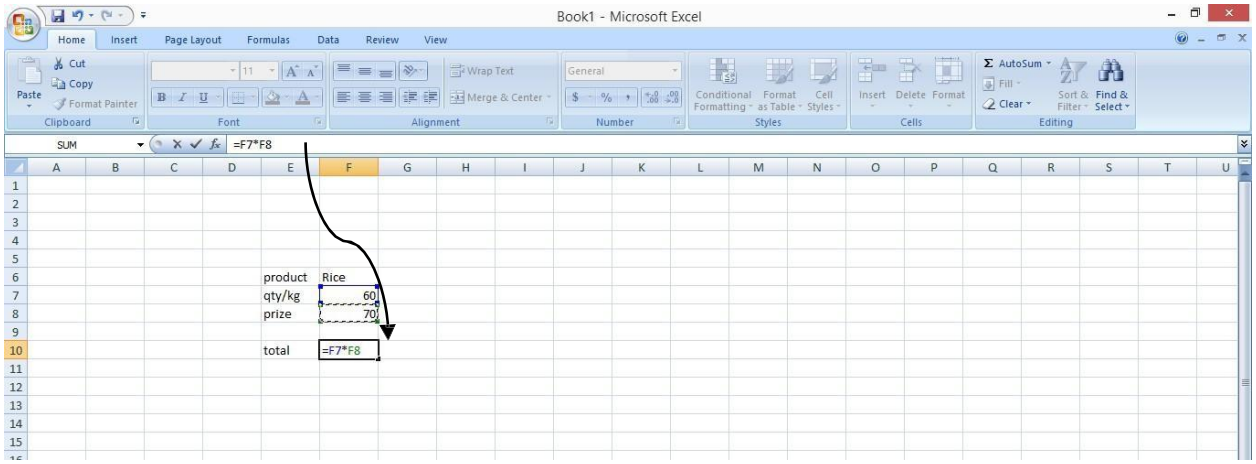
Data Table

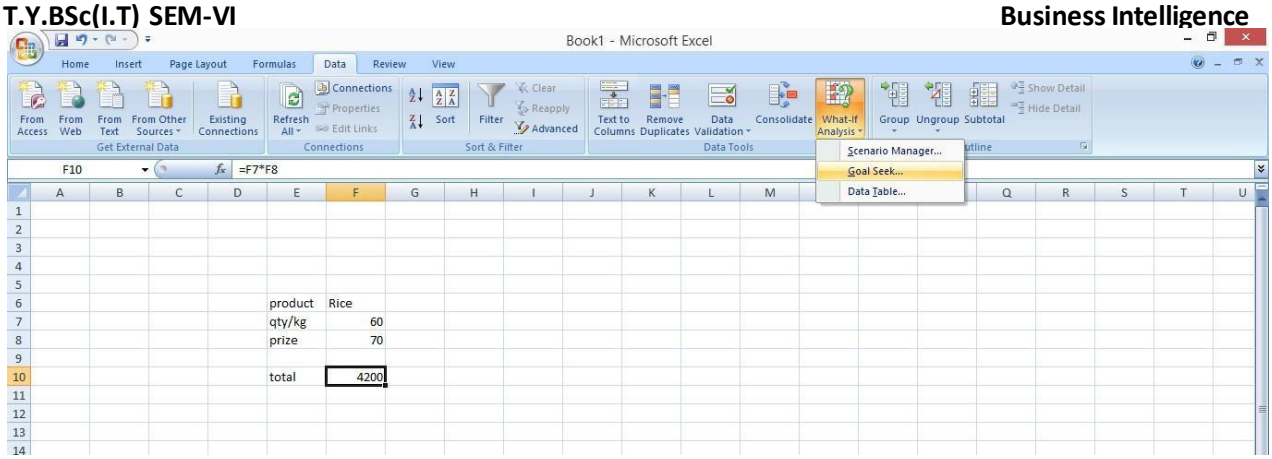
- If you have a formula that uses one or two variables, or multiple formulas that all use one common variable, you can use a **Data Table** to see all the outcomes in one place.

Scenario Manager

- A **Scenario Manager** is a set of values that Excel saves and can substitute automatically in cells on a worksheet. You can create and save different groups of values on a worksheet and then switch to any of these new scenarios to view different results.

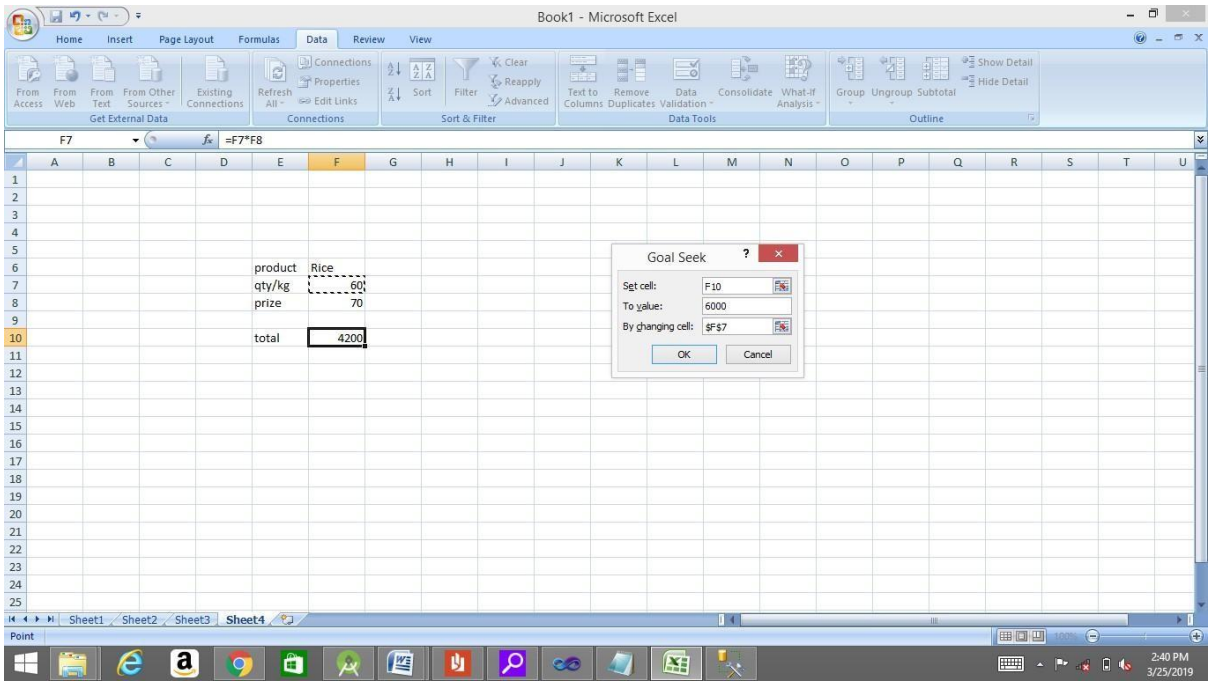
a) Understanding Goal Seek Option

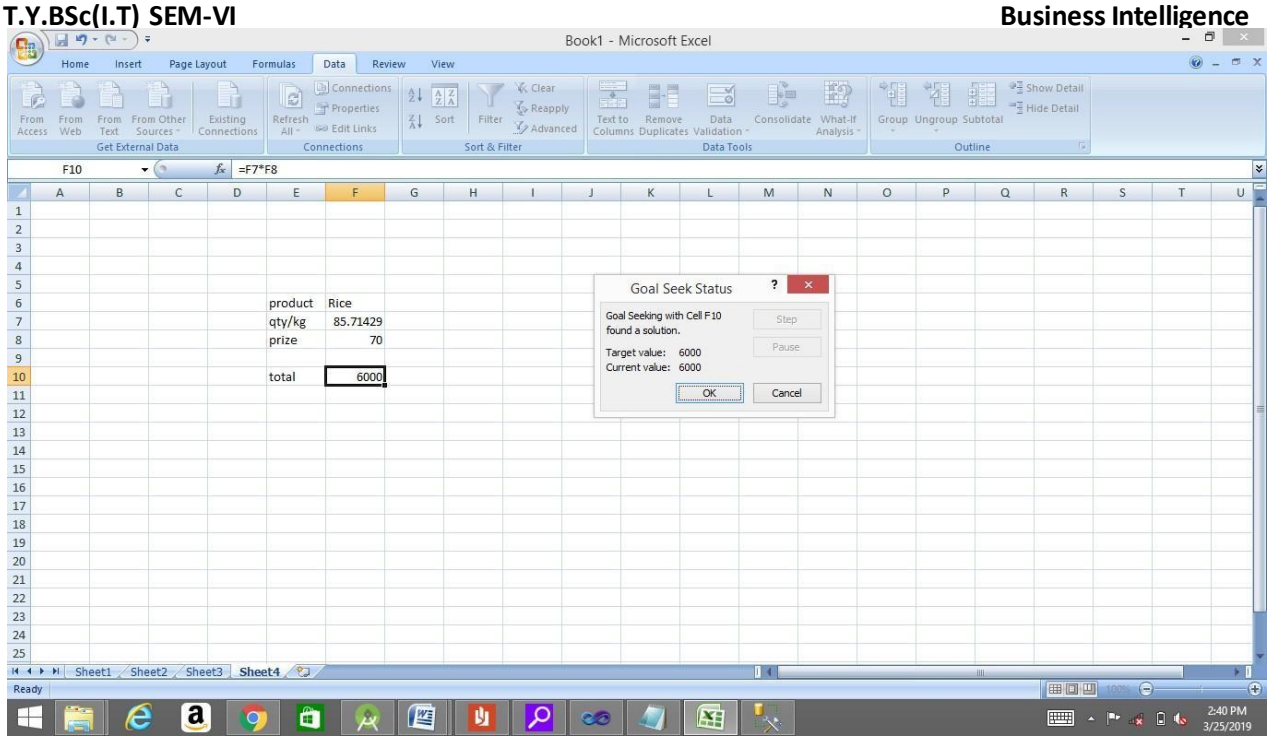




To Change qyt to achieve 6000/-. How much qty will be required.

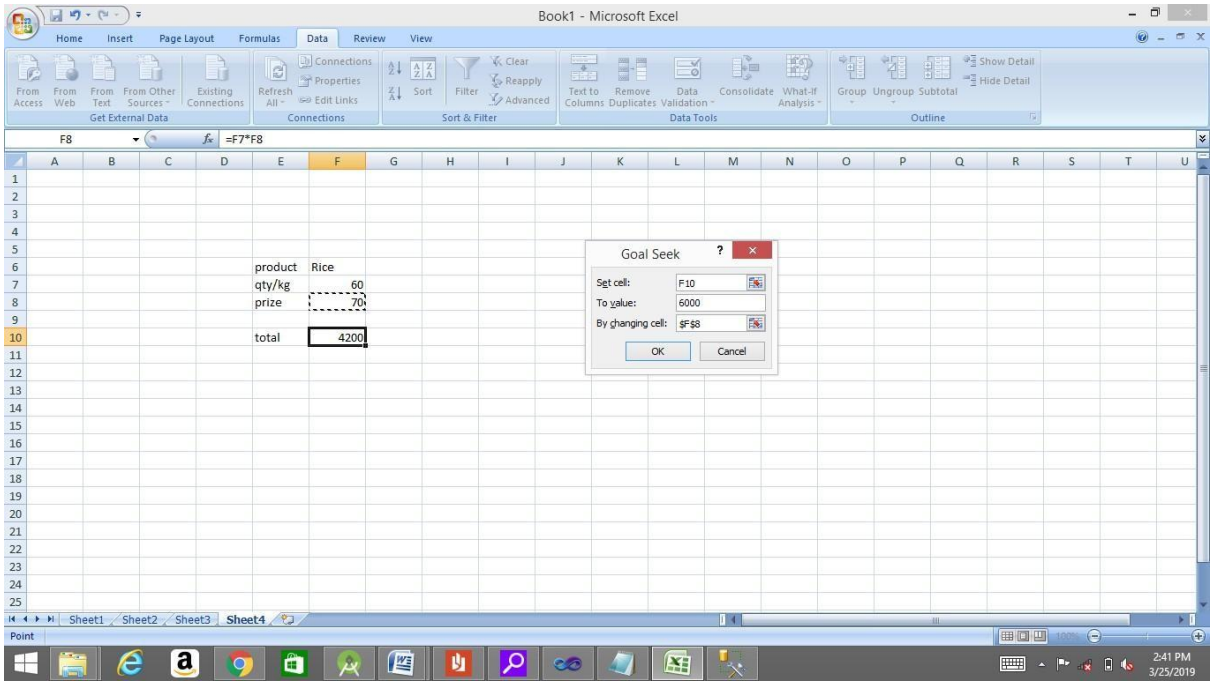
Go to What-If analysis and select Goal Seek Option

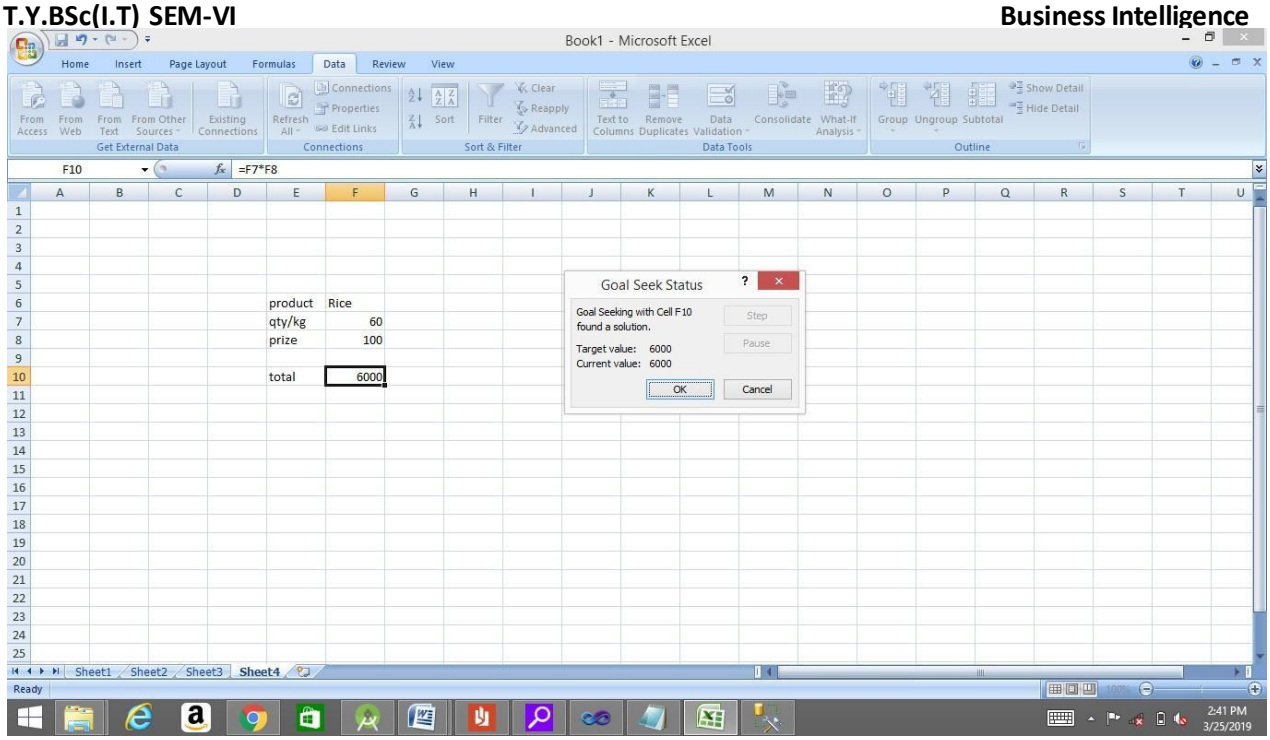




To Change prize to achieve 6000/-. How much prize will be required.

Go to What-If analysis and select Goal Seek Option

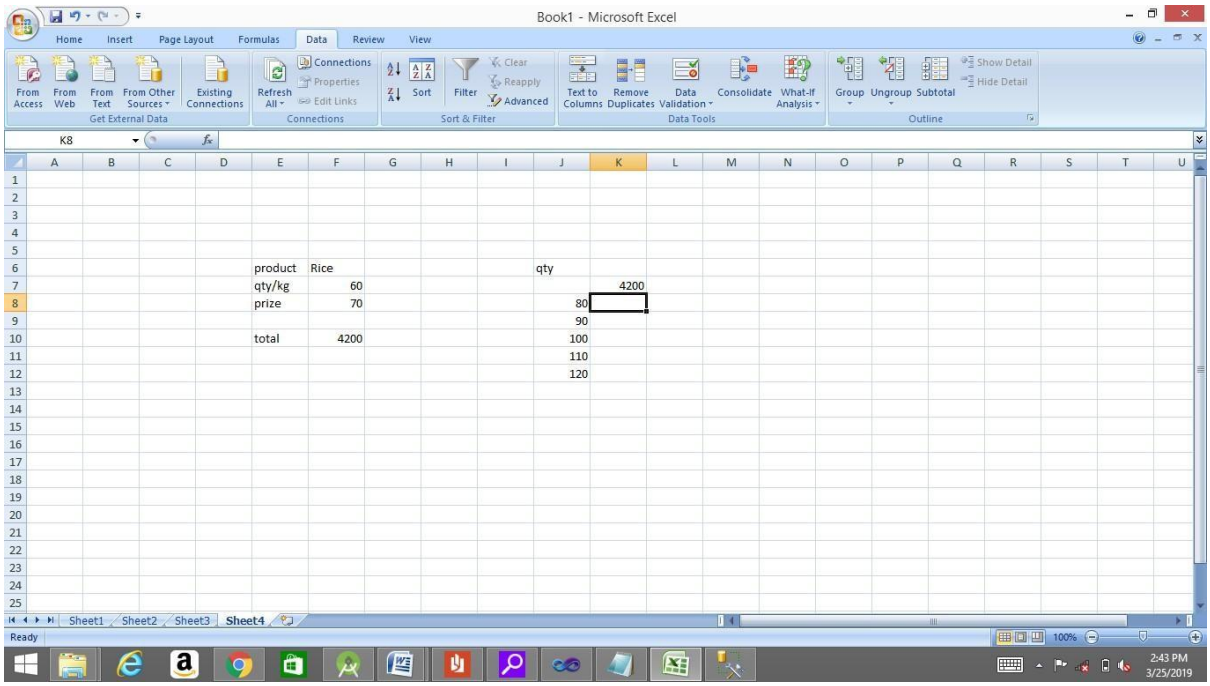


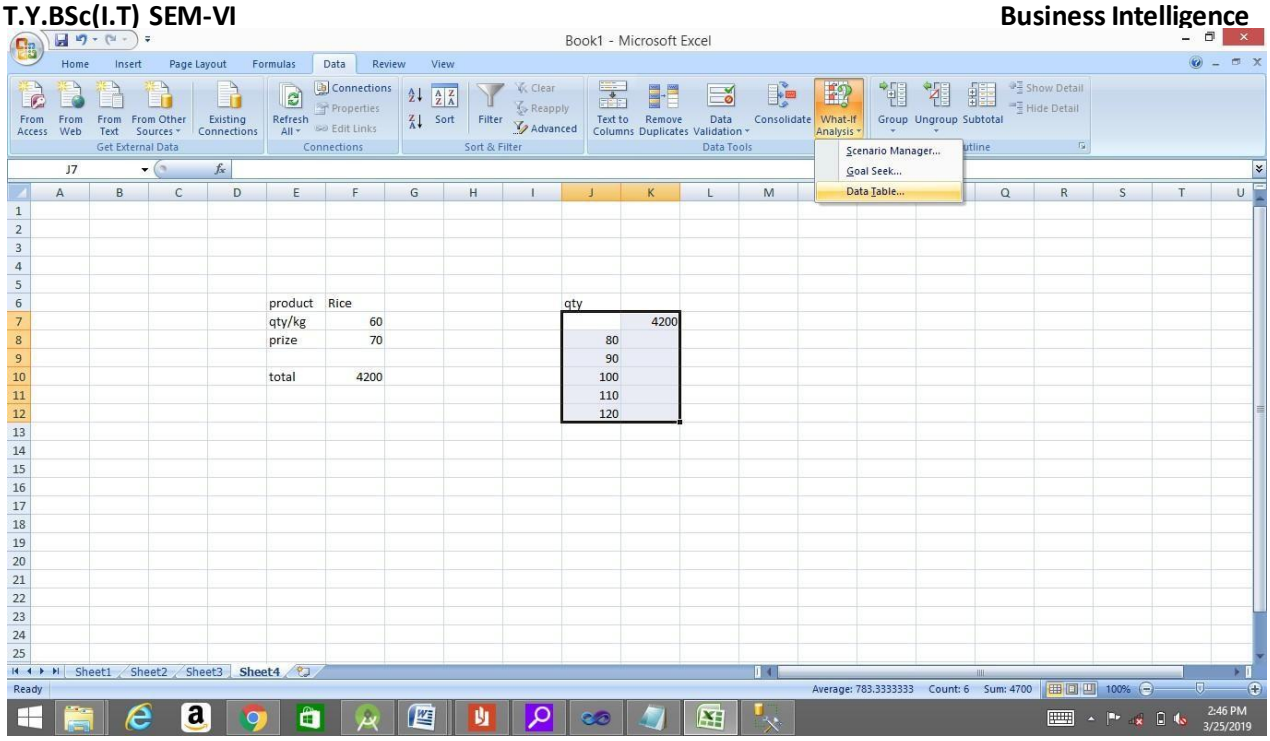


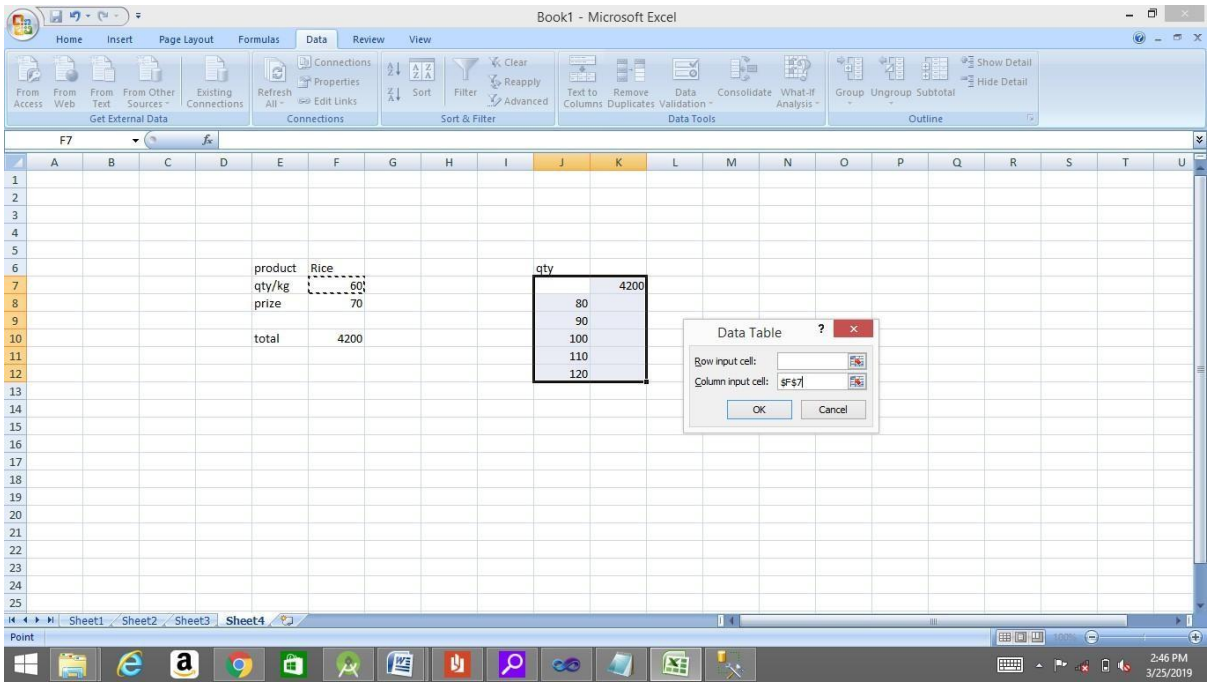
b) Understanding Data Table Option

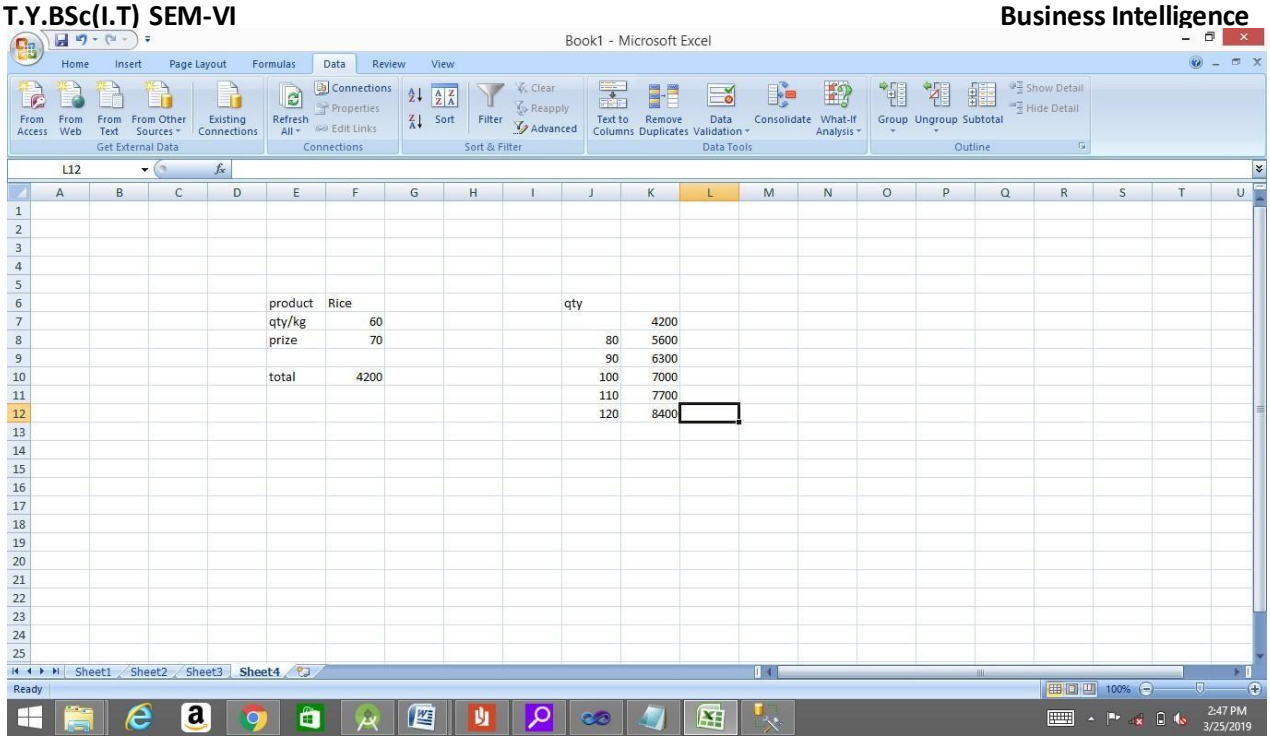
Generate Data table using qty.

For this, go to What-If analysis and select Data Table option.

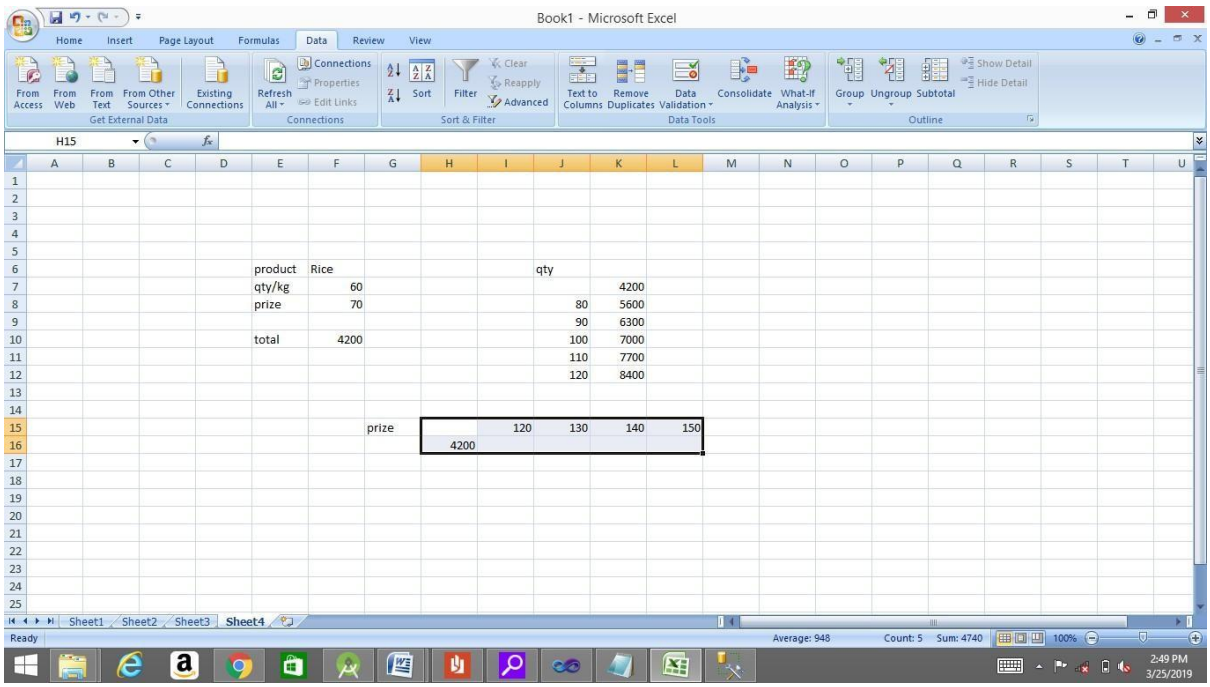


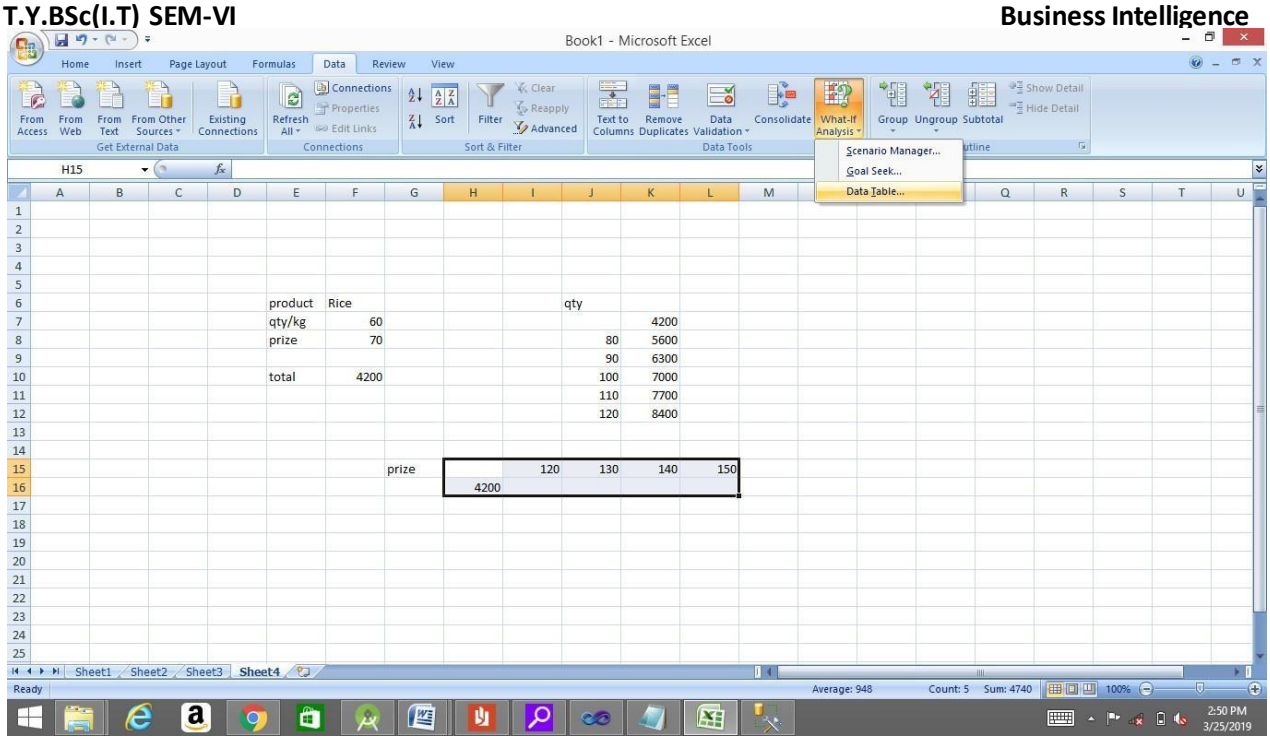

Book1 - Microsoft Excel
Home Insert Page Layout Formulas Data Review View
What-If Analysis
Scenario Manager...
Goal Seek...
Data Table...
product Rice
qty/kg 60
prize 70
total 4200
qty 4200
80
90
100
110
120
Sheet1 Sheet2 Sheet3 Sheet4
Ready
Average: 783.3333333 Count: 6 Sum: 4700

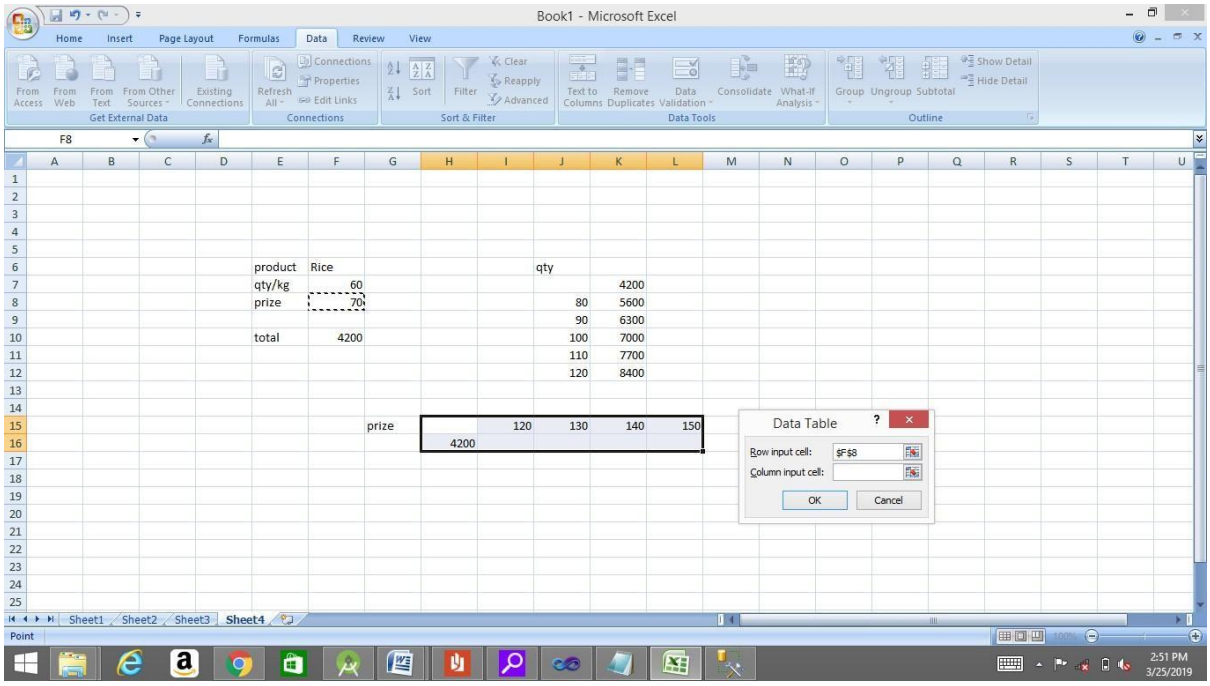

Book1 - Microsoft Excel
product Rice
qty/kg 60
prize 70
total 4200
qty 4200
80
90
100
110
120
Data Table
Row input cell:
Column input cell: $F$7
OK
Cancel
Sheet1 Sheet2 Sheet3 Sheet4
Point

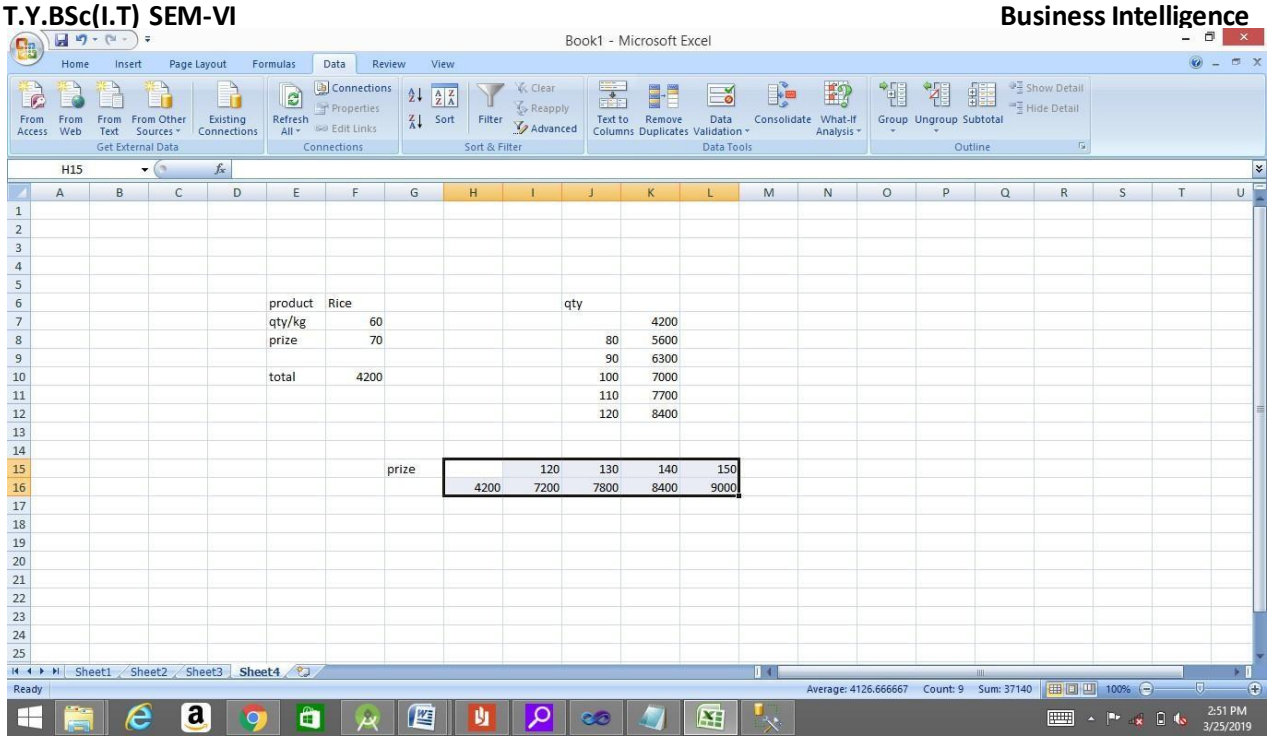


Generate Data table using prize.

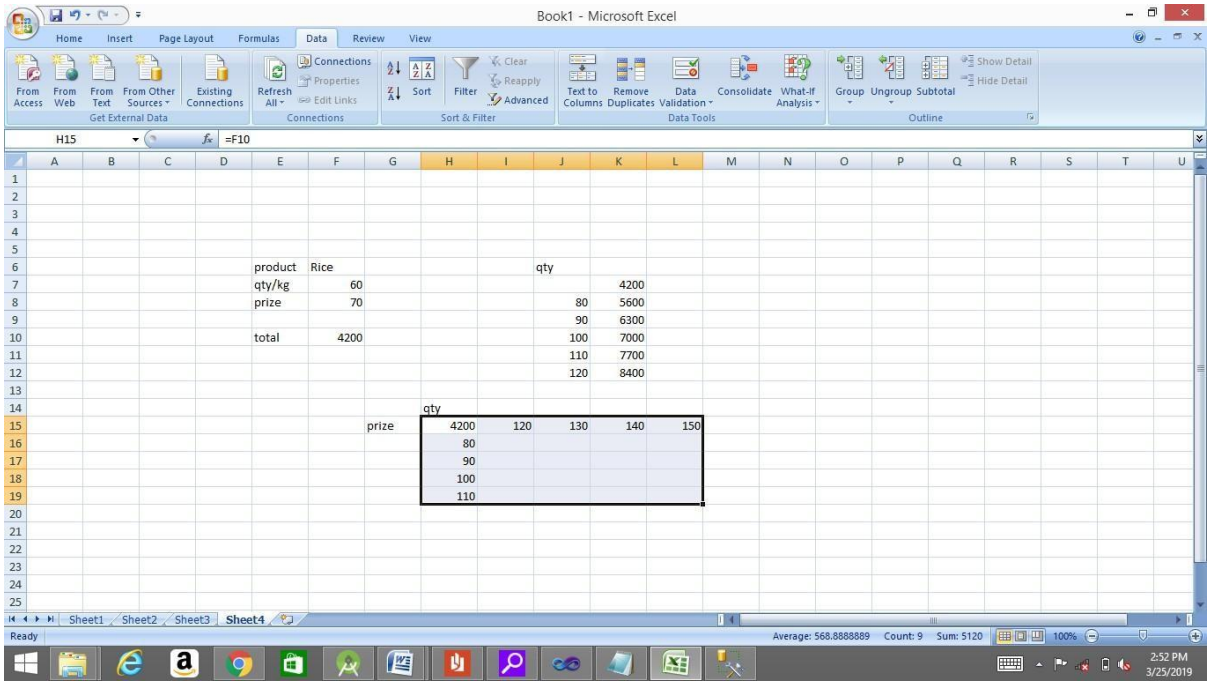


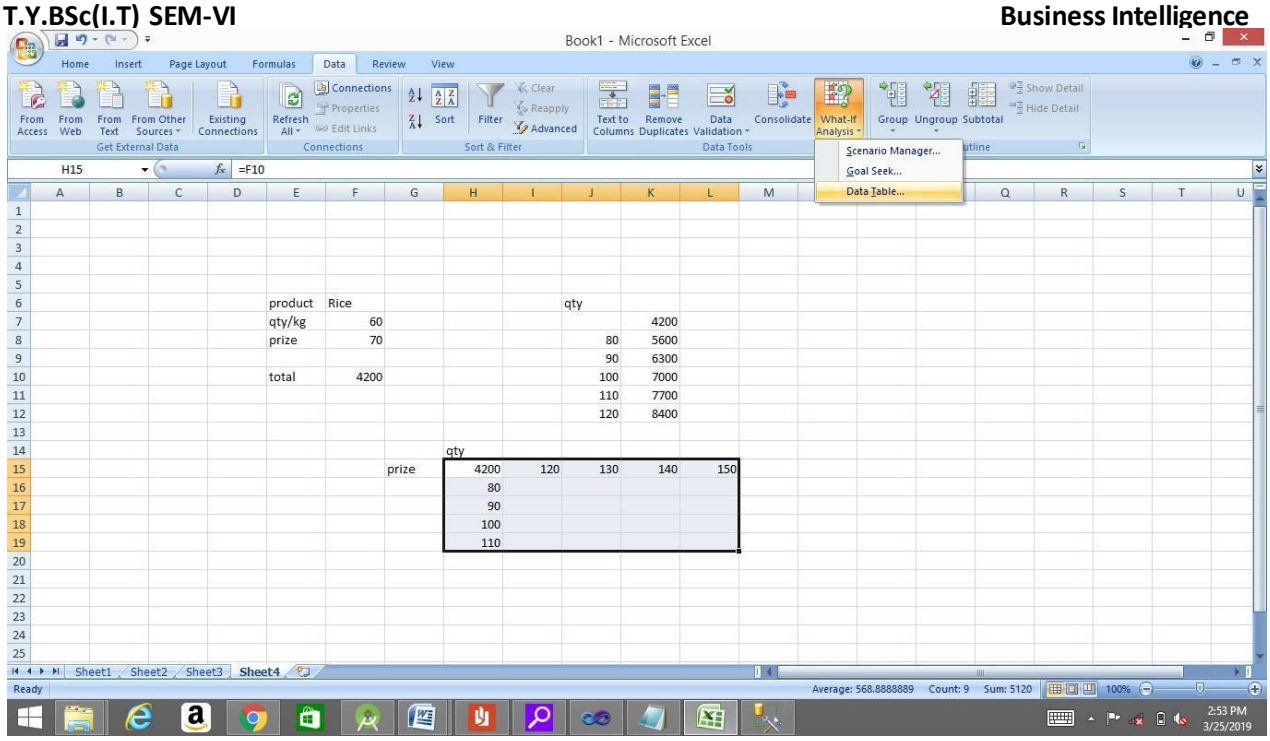

Book1 - Microsoft Excel
Scenario Manager...
Goal Seek...
Data Table...
product Rice
qty/kg 60
prize 70
total 4200
qty
4200
80 5600
90 6300
100 7000
110 7700
120 8400
prize 120 130 140 150
4200

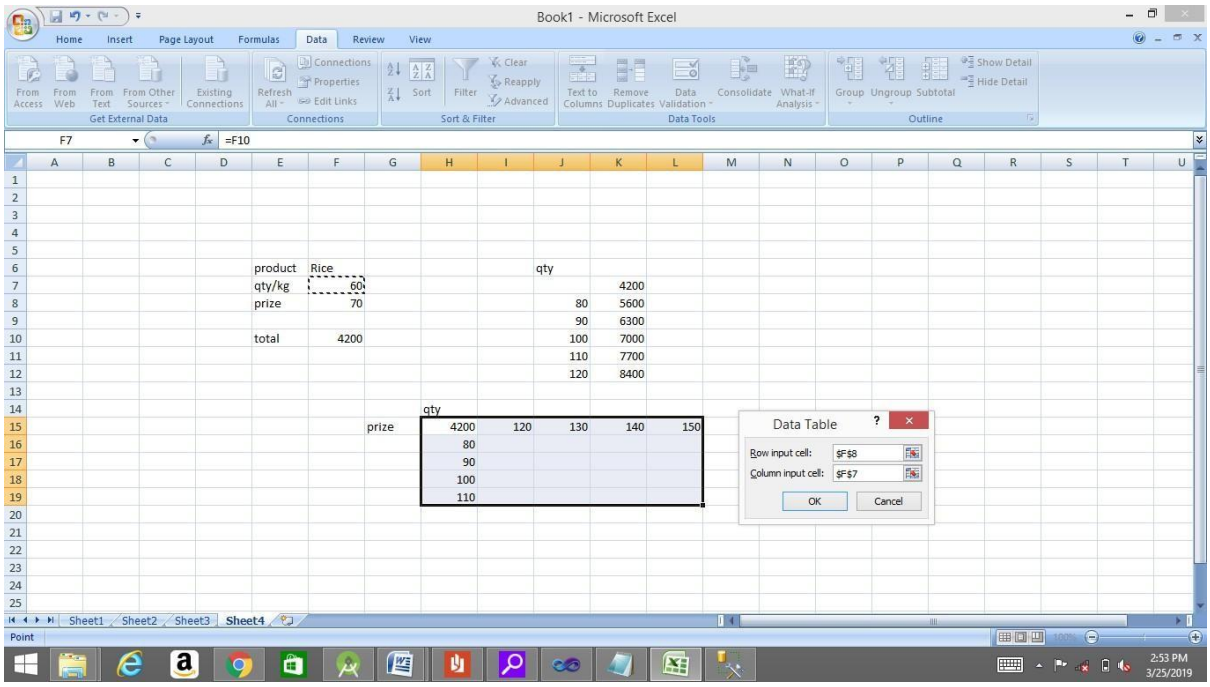

Book1 - Microsoft Excel
Data Table
Row input cell: $F$8
Column input cell:
OK
Cancel
product Rice
qty/kg 60
prize 70
total 4200
qty
4200
80 5600
90 6300
100 7000
110 7700
120 8400
prize 120 130 140 150
4200

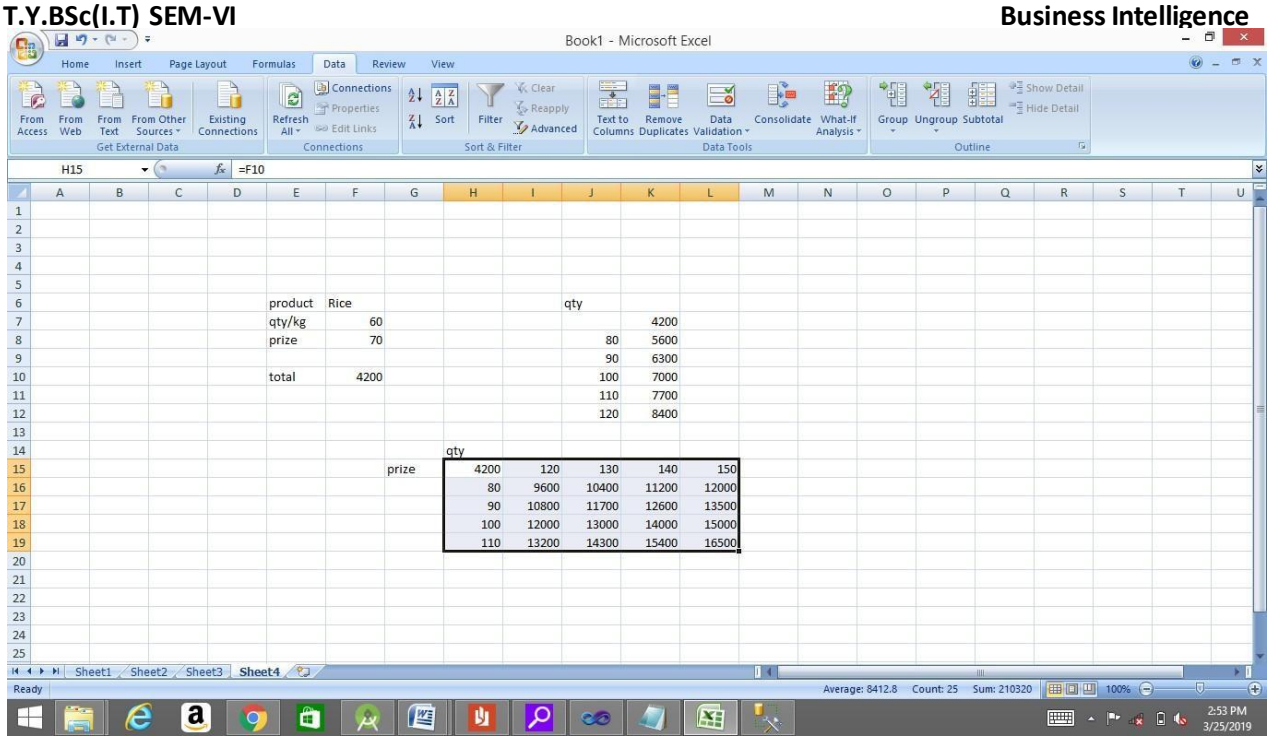


Generate Data table using qty and prize both.

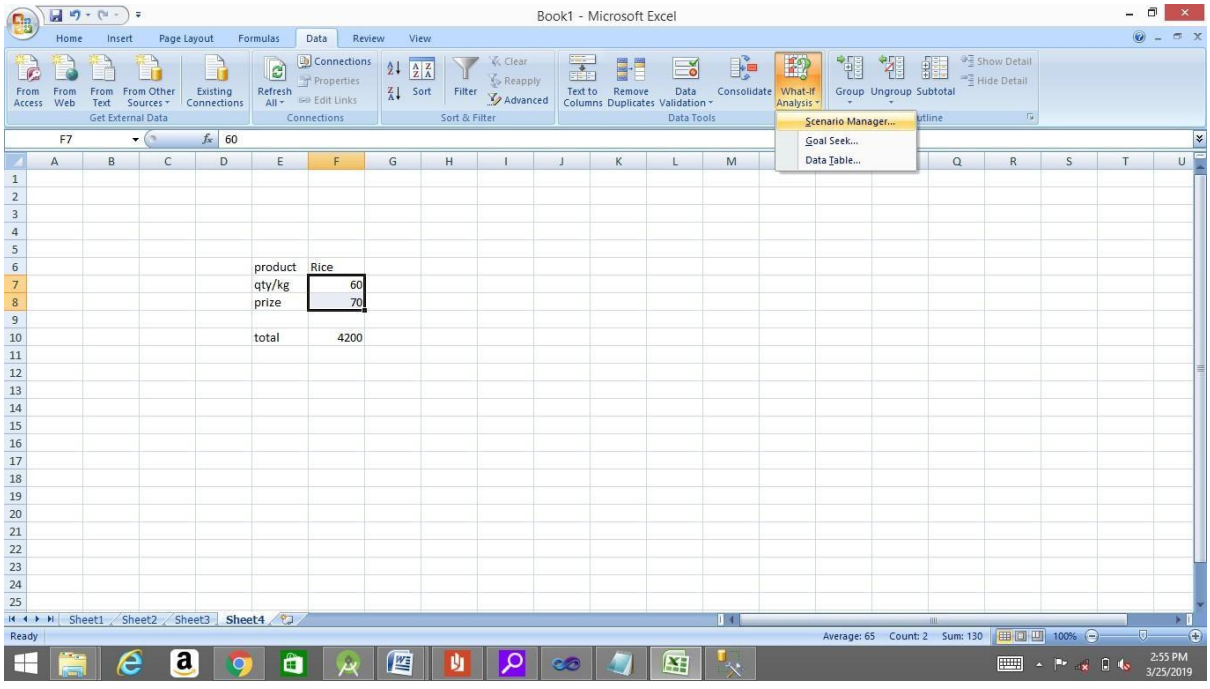


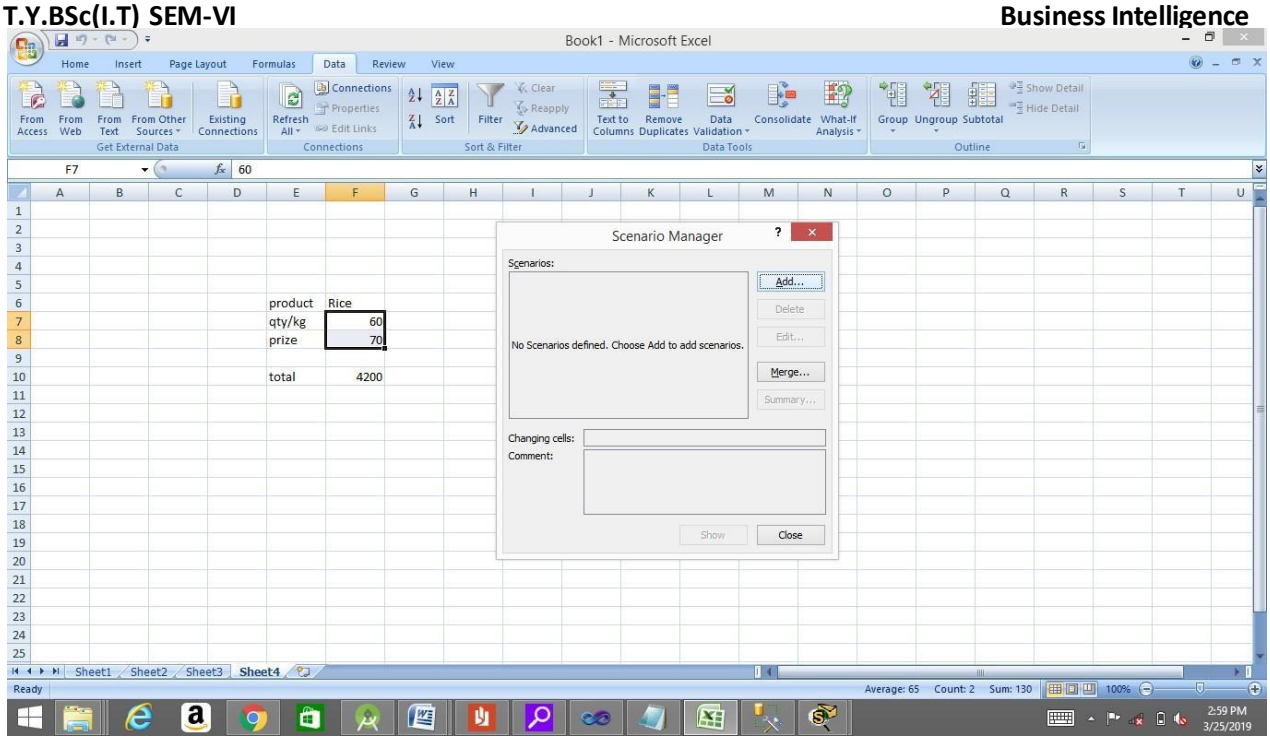

Book1 - Microsoft Excel
Home Insert Page Layout Formulas Data Review View
What-If Analysis
Scenario Manager...
Goal Seek...
Data Table...
H15 =F10
product Rice
qty/kg 60
prize 70
total 4200
qty
80 5600
90 6300
100 7000
110 7700
120 8400
qty
prize 4200 120 130 140 150
80
90
100
110
Sheet1 Sheet2 Sheet3 Sheet4
Ready
Average: 568.8888889 Count: 9 Sum: 5120
2:53 PM 3/25/2019

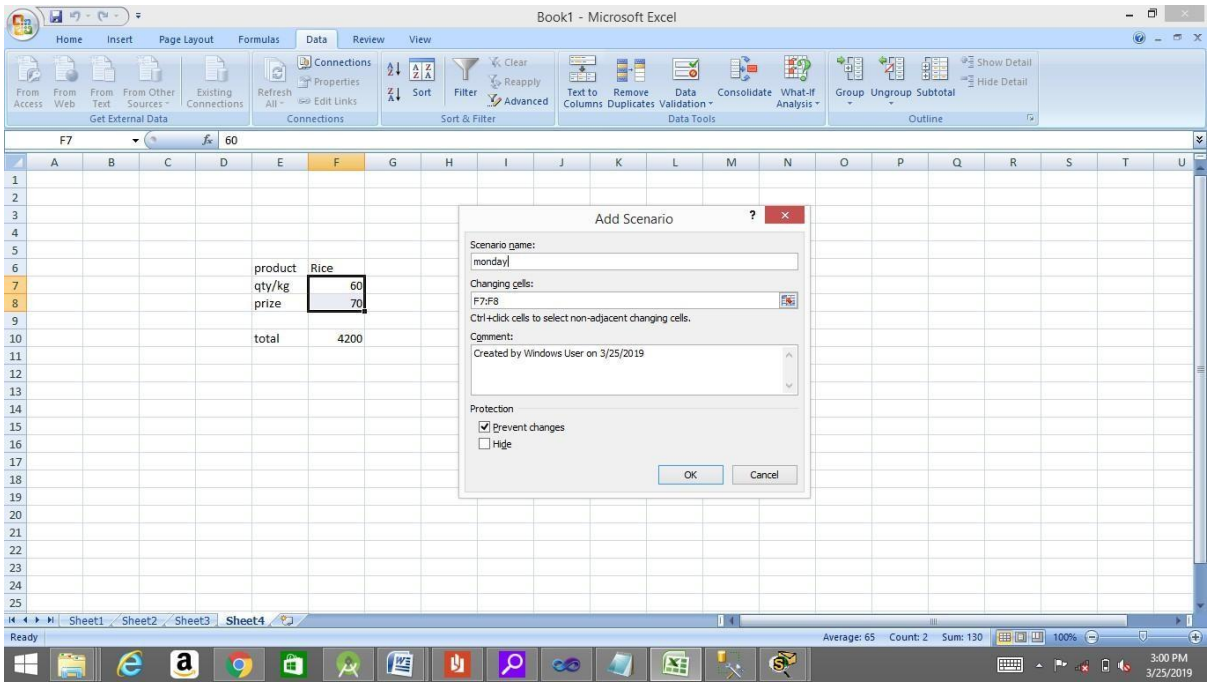

Book1 - Microsoft Excel
F7 =F10
product Rice
qty/kg 60
prize 70
total 4200
qty
80 5600
90 6300
100 7000
110 7700
120 8400
qty
prize 4200 120 130 140 150
80
90
100
110
Data Table
Row input cell: $F$8
Column input cell: $F$7
OK Cancel
Sheet1 Sheet2 Sheet3 Sheet4
Point
2:53 PM 3/25/2019

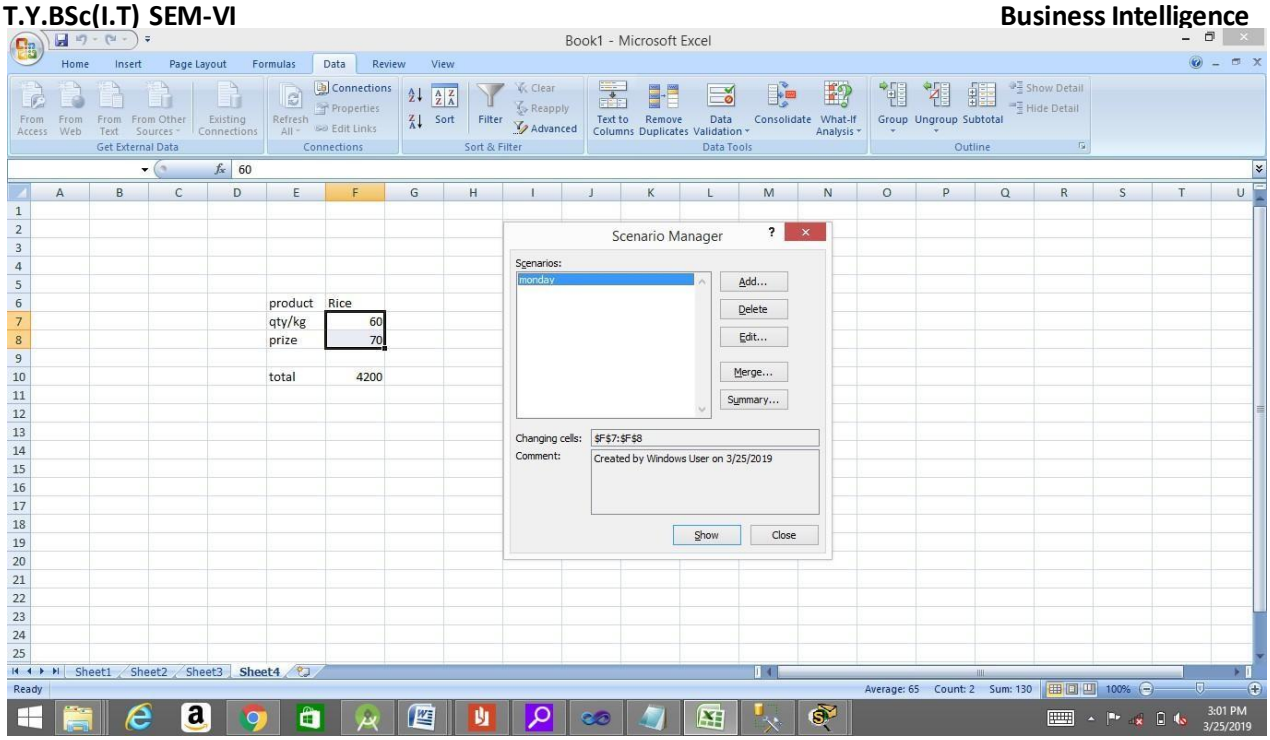


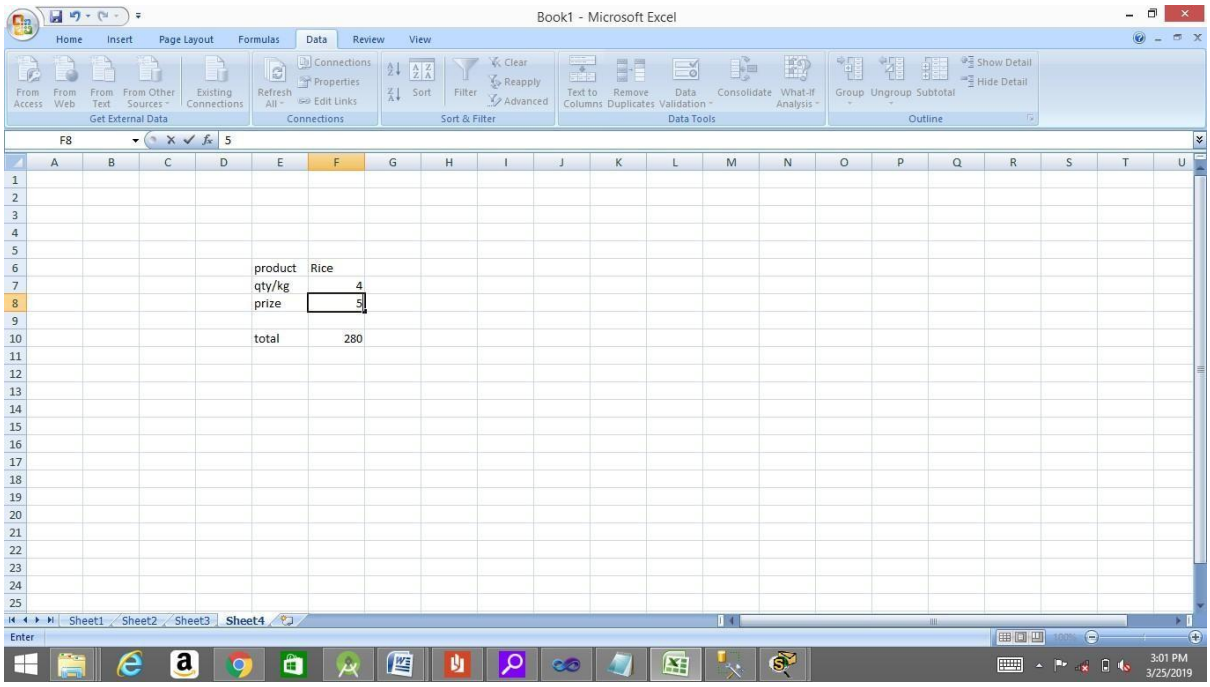
c) Understanding Scenario Manager Option

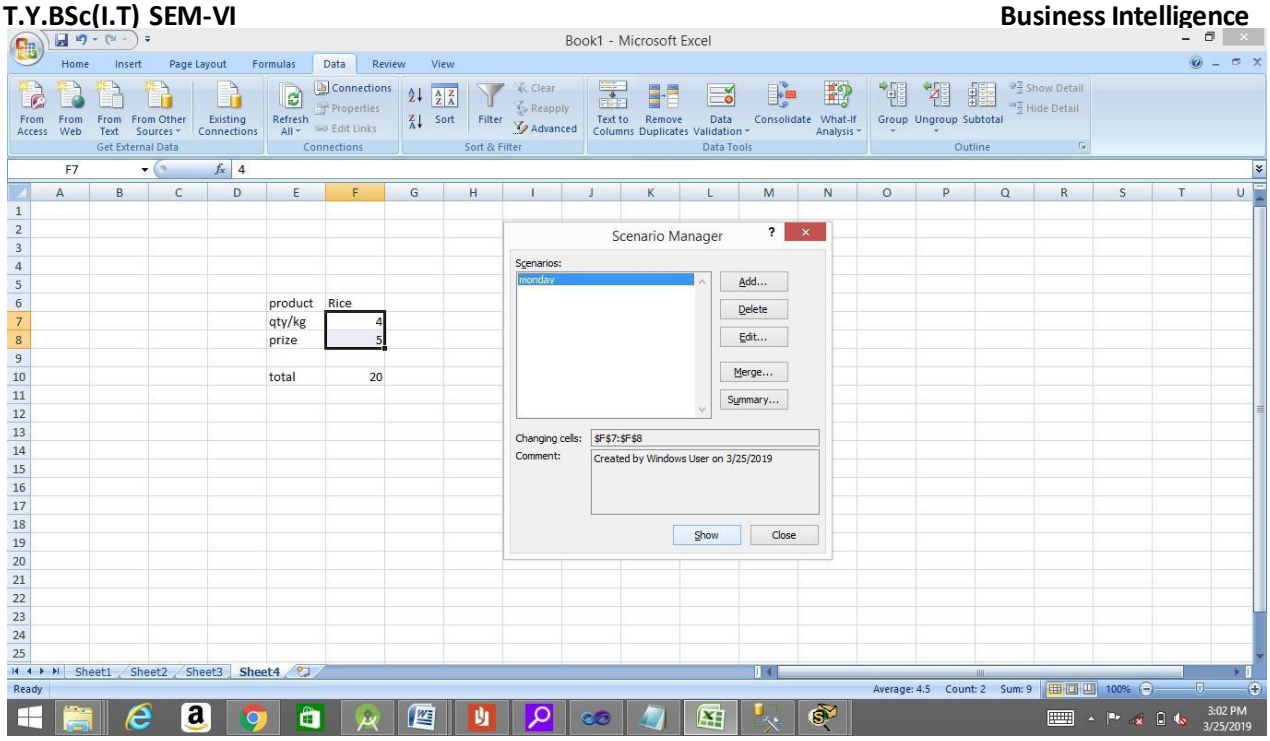


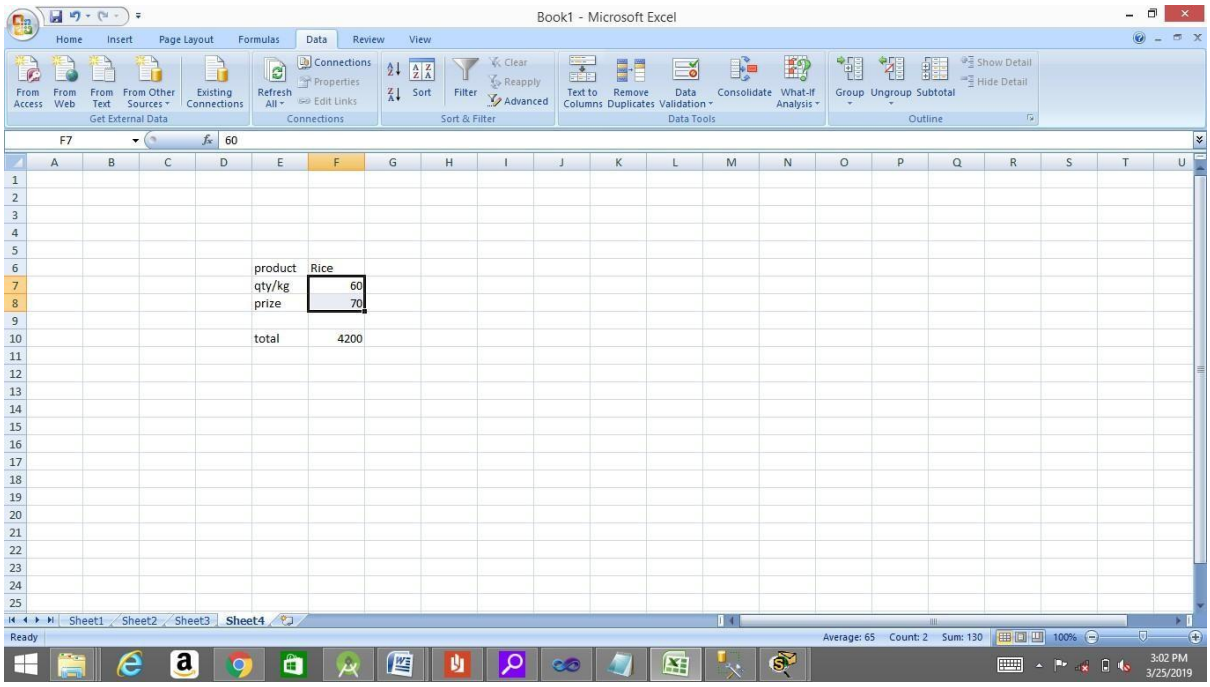

Book1 - Microsoft Excel
Scenario Manager
Scenarios:
No Scenarios defined. Choose Add to add scenarios.
Add...
Delete
Edit...
Merge...
Summary...
Changing cells:
Comment:
Show
Close
product Rice
qty/kg 60
prize 70
total 4200


Book1 - Microsoft Excel
Add Scenario
Scenario name:
monday
Changing cells:
F7:F8
Ctrl+click cells to select non-adjacent changing cells.
Comment:
Created by Windows User on 3/25/2019
Protection
Prevent changes
Hide
OK
Cancel
product Rice
qty/kg 60
prize 70
total 4200


Book1 - Microsoft Excel
Scenario Manager
Scenarios:
monday
Add...
Delete
Edit...
Merge...
Summary...
Changing cells: $F$7:$F$8
Comment: Created by Windows User on 3/25/2019
Show
Close
product Rice
qty/kg 60
prize 70
total 4200
Sheet1 Sheet2 Sheet3 Sheet4
Ready
Average: 65 Count: 2 Sum: 130
3:01 PM 3/25/2019


Book1 - Microsoft Excel
product Rice
qty/kg 4
prize 5
total 280
Sheet1 Sheet2 Sheet3 Sheet4
Enter
3:01 PM 3/25/2019


Book1 - Microsoft Excel
Scenario Manager
Scenarios:
monday
Add...
Delete
Edit...
Merge...
Summary...
Changing cells: $F$7:$F$8
Comment: Created by Windows User on 3/25/2019
Show
Close
product Rice
qty/kg 4
prize 5
total 20
Sheet1 Sheet2 Sheet3 Sheet4
Ready
Average: 4.5 Count: 2 Sum: 9
3:02 PM 3/25/2019


Book1 - Microsoft Excel
product Rice
qty/kg 60
prize 70
total 4200
Sheet1 Sheet2 Sheet3 Sheet4
Ready
Average: 65 Count: 2 Sum: 130
3:02 PM 3/25/2019

Practical No.3

Perform the data classification using classification algorithm using R/Python.

Classification

- **Classification** is a **data mining** function that assigns items in a collection to target categories or classes.
- The goal of **classification** is to accurately predict the target class for each case in the **data**.
- For example, a **classification** model could be used to identify loan applicants as risky and safe.
- Classifier
- Prediction

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

```
import pandas as pd
—----------------------------------------------------------------------
df= pd.read_csv("/content/shows.csv")
—----------------------------------------------------------------------
print(df)
```

	Age	Experience	Rank	Nationality	Go
0	36	10	9	UK	NO
1	42	12	4	USA	NO
2	23	4	6	N	NO
3	52	4	4	USA	NO
4	43	21	8	USA	YES
5	44	14	5	UK	NO
6	66	3	7	N	YES
7	35	14	9	UK	YES
8	52	13	7	N	YES
9	35	5	9	N	YES
10	24	3	5	USA	NO
11	18	3	7	UK	YES
12	45	9	9	UK	YES

```
—----------------------------------------------------------------------
from sklearn import tree
—----------------------------------------------------------------------
import pydotplus
—----------------------------------------------------------------------
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import matplotlib.image as pltimage
—----------------------------------------------------------------------
#change string values into numerical values:
d= {'UK':0, 'USA':1, 'N':2}
—----------------------------------------------------------------------
print (d)
```

```
{'UK': 0, 'USA': 1, 'N': 2}
```

—----------

df['Nationality']= df['Nationality'].map(d)

—----------

df

	Age	Experience	Rank	Nationality	Go
0	36	10	9	0	NO
1	42	12	4	1	NO
2	23	4	6	2	NO
3	52	4	4	1	NO
4	43	21	8	1	YES
5	44	14	5	0	NO
6	66	3	7	2	YES
7	35	14	9	0	YES
8	52	13	7	2	YES
9	35	5	9	2	YES
10	24	3	5	1	NO
11	18	3	7	0	YES
12	45	9	9	0	YES

—----------

```
d={'YES':1, 'NO': 0}
df['Go']=df['Go'].map(d)
```

—----------

print(df)

	Age	Experience	Rank	Nationality	Go
0	36	10	9	0	0
1	42	12	4	1	0
2	23	4	6	2	0
3	52	4	4	1	0
4	43	21	8	1	1
5	44	14	5	0	0
6	66	3	7	2	1
7	35	14	9	0	1
8	52	13	7	2	1
9	35	5	9	2	1
10	24	3	5	1	0
11	18	3	7	0	1
12	45	9	9	0	1

—----------

```
features =['Age','Experience','Rank','Nationality']
```

—----------

x= df[features]

—----------

y=df['Go']

—----------

print(x)

```
    Age  Experience  Rank  Nationality
0    36          10     9            0
1    42          12     4            1
2    23           4     6            2
3    52           4     4            1
4    43          21     8            1
5    44          14     5            0
6    66           3     7            2
7    35          14     9            0
8    52          13     7            2
9    35           5     9            2
10   24           3     5            1
11   18           3     7            0
12   45           9     9            0
```

—----------

print(y)

```
0     0
1     0
2     0
3     0
4     1
5     0
6     1
7     1
8     1
9     1
10    0
11    1
12    1
Name: Go, dtype: int64
```

—----------

df.shape

```
(13, 5)
```

—----------

dtree=DecisionTreeClassifier()

—----------

dtree=dtree.fit(x,y)

Decision Tree

```
import pandas as pd
—----------------------------------------------------------------------
from sklearn import tree
import pydotplus
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import matplotlib.image as pltimage
—----------------------------------------------------------------------
df= pd.read_csv("/content/shows.csv")
print(df)
```

	Age	Experience	Rank	Nationality	Go
0	36	10	9	UK	NO
1	42	12	4	USA	NO
2	23	4	6	N	NO
3	52	4	4	USA	NO
4	43	21	8	USA	YES
5	44	14	5	UK	NO
6	66	3	7	N	YES
7	35	14	9	UK	YES
8	52	13	7	N	YES
9	35	5	9	N	YES
10	24	3	5	USA	NO
11	18	3	7	UK	YES
12	45	9	9	UK	YES

```
—----------------------------------------------------------------------
# Change string values into numerical values:
d = {'UK': 0, 'USA': 1, 'N': 2}
df['Nationality'] = df['Nationality'].map(d)
d = {'YES': 1, 'NO': 0}
df['Go'] = df['Go'].map(d)
—----------------------------------------------------------------------
print(df)
```

	Age	Experience	Rank	Nationality	Go
0	36	10	9	0	0
1	42	12	4	1	0
2	23	4	6	2	0
3	52	4	4	1	0
4	43	21	8	1	1
5	44	14	5	0	0
6	66	3	7	2	1
7	35	14	9	0	1
8	52	13	7	2	1
9	35	5	9	2	1
10	24	3	5	1	0
11	18	3	7	0	1
12	45	9	9	0	1

```
—----------------------------------------------------------------------
print(x)
```

print(y)

df.shape

```
    Age  Experience  Rank  Nationality
0    36          10     9            0
1    42          12     4            1
2    23           4     6            2
3    52           4     4            1
4    43          21     8            1
5    44          14     5            0
6    66           3     7            2
7    35          14     9            0
8    52          13     7            2
9    35           5     9            2
10   24           3     5            1
11   18           3     7            0
12   45           9     9            0
0     0
1     0
2     0
3     0
4     1
5     0
6     1
7     1
8     1
9     1
10    0
11    1
12    1
Name: Go, dtype: int64
(13, 5)
```

—----------

features = ['Age','Experience','Rank','Nationality']

—----------

x=df[features]

y=df['Go']

print(x)

print(y)

```
    Age  Experience  Rank  Nationality
0    36          10     9            0
1    42          12     4            1
2    23           4     6            2
3    52           4     4            1
4    43          21     8            1
5    44          14     5            0
6    66           3     7            2
7    35          14     9            0
8    52          13     7            2
9    35           5     9            2
10   24           3     5            1
11   18           3     7            0
12   45           9     9            0
0     0
1     0
2     0
3     0
4     1
5     0
6     1
7     1
8     1
9     1
10    0
11    1
12    1
Name: Go, dtype: int64
```

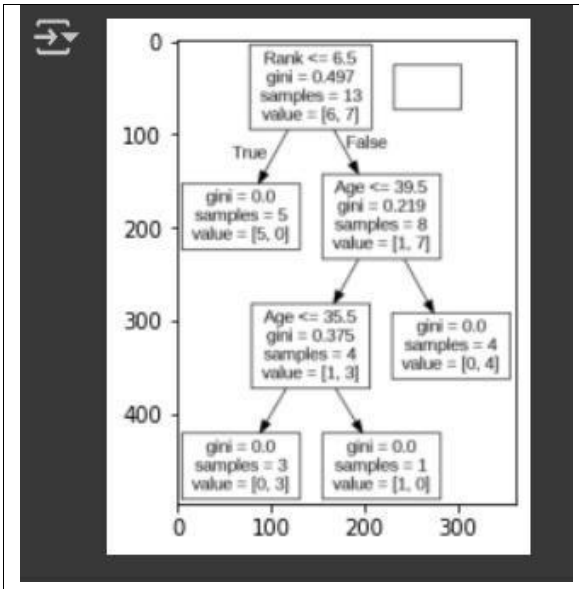
—----------

df.shape

```
(13, 5)
```

—----------

```
dtree=DecisionTreeClassifier()
dtree=dtree.fit(x.values,y.values)
data=tree.export_graphviz(dtree,out_file=None,feature_names=features)
graph=pydotplus.graph_from_dot_data(data)
graph.write_png('mydecisiontree.png')
img=pltimage.imread('mydecisiontree.png')
imgplot=plt.imshow(img)
plt.show()
```



—----------

```
print(dtree.predict([[40,10,7,1]]))
```

```
[1]
```

—----------

```
print(dtree.predict([[40,10,6,1]]))
```

```
[0]
```

—----------

```
features = ['Age','Experience','Rank','Nationality']
```

—----------

```
x=df[features]
```

—----------

```
print(x)
```

```
    Age  Experience  Rank  Nationality
0    36          10     9            0
1    42          12     4            1
2    23           4     6            2
3    52           4     4            1
4    43          21     8            1
5    44          14     5            0
6    66           3     7            2
7    35          14     9            0
8    52          13     7            2
9    35           5     9            2
10   24           3     5            1
11   18           3     7            0
12   45           9     9            0
```

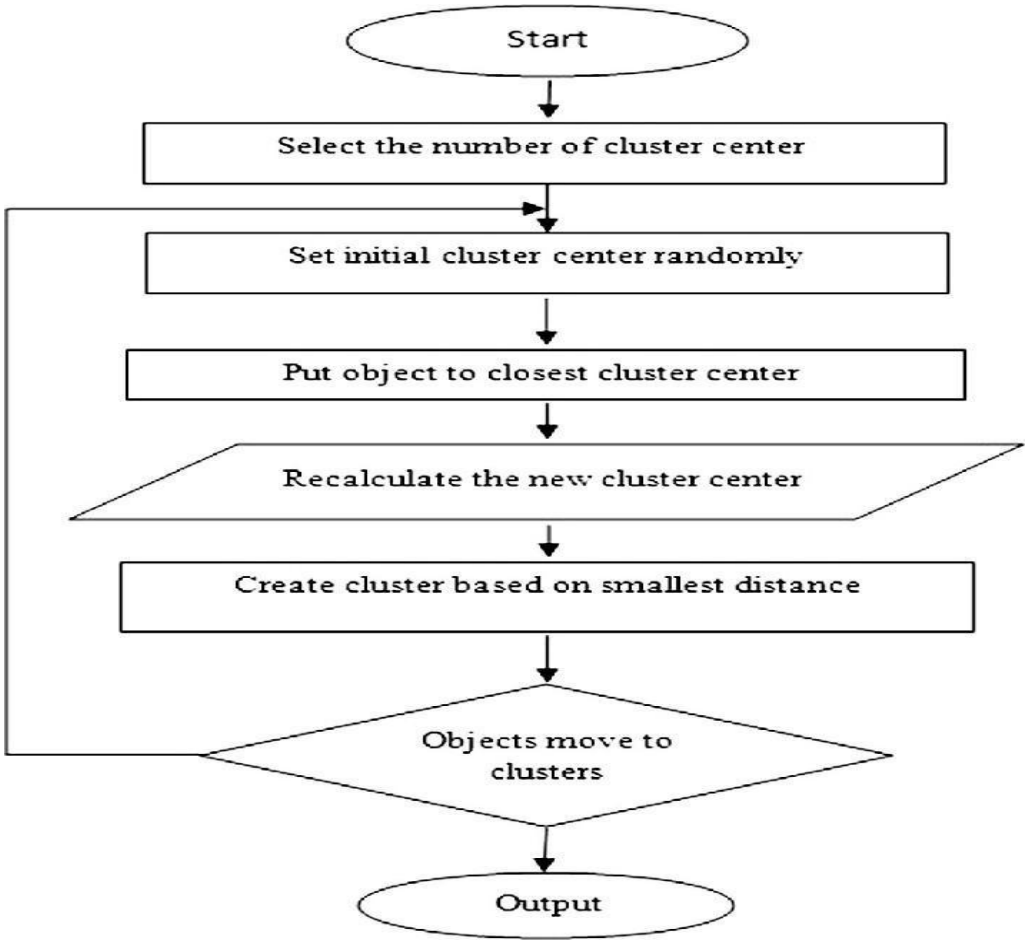
Practical No.4

Perform the data clustering using clustering algorithm using R/Python.

Clustering

- **Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups.**
- **Let's understand this with an example.**
- **Suppose, you are the head of a rental store and wish to understand preferences of your Customers to scale up your business.**
- **Is it possible for you to look at details of each costumer and devise a unique business strategy for each one of them?**
- **Definitely not. But, what you can do is to cluster all of your Customers into say 10 groups based on their purchasing habits and use a separate strategy for costumers in each of these 10 groups. And this is what we call clustering.**

K-Means Clustering



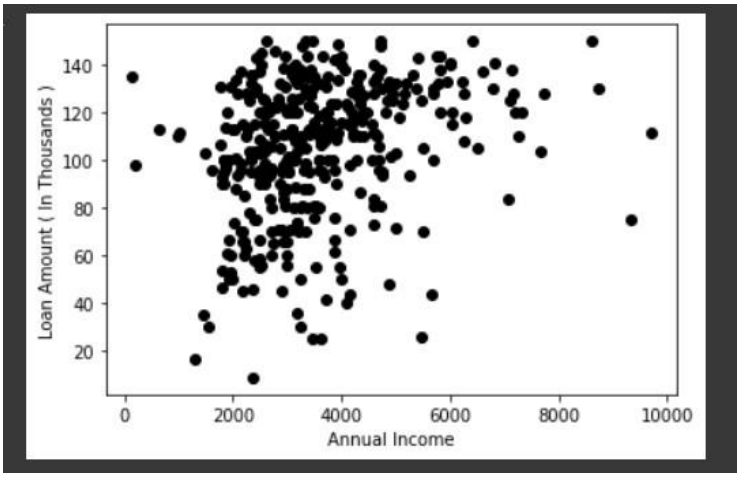
```
import pandas as pd
import numpy as np
import random as rd
import matplotlib.pyplot as plt
—----------------------------------------------------------------------
data = pd.read_csv("/content/clustering.csv")
—----------------------------------------------------------------------
```

data

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
1	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
2	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
3	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
4	LP001013	Male	Yes	0	Not Graduate	No	2333	1516.0	95.0	360.0	1.0	Urban	Y
...	...	...	...	...	...	...	...	...	...	...	...	...	...
376	LP002953	Male	Yes	3+	Graduate	No	5703	0.0	128.0	360.0	1.0	Urban	Y
377	LP002974	Male	Yes	0	Graduate	No	3232	1950.0	108.0	360.0	1.0	Rural	Y
378	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
379	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
380	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semiurban	N

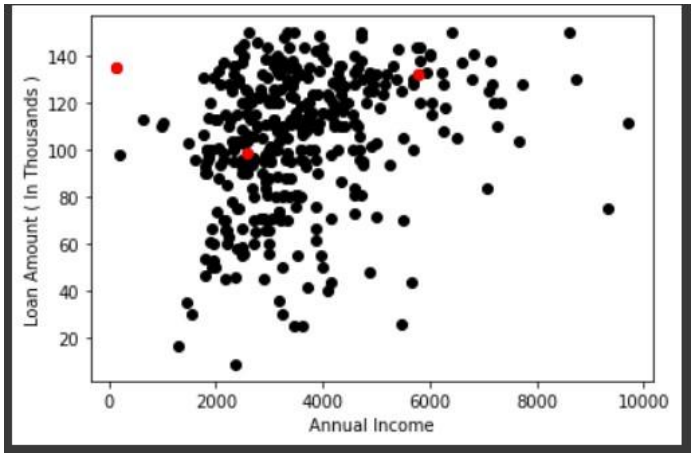
381 rows × 13 columns

```
—----------------------------------------------------------------------
X=data[["LoanAmount","ApplicantIncome"]]
#Visualize the data points (ap,la)
plt.scatter(X["ApplicantIncome"],X["LoanAmount"],c="black")
plt.xlabel("Annual Income")
plt.ylabel("Loan Amount ( In Thousands )")
plt.show()
```




```
—----------------------------------------------------------------------
# Step 1 and Step 2 Choose the number of clusters(K) and select random
centriod for each cluster
# number of clusters
K=3
# Select random observations as centriods
Centroids = (X.sample(n=K))
```

```
plt.scatter(X["ApplicantIncome"],X["LoanAmount"],c="black")
plt.scatter(Centroids["ApplicantIncome"],Centroids["LoanAmount"],c="re
d")
plt.xlabel("Annual Income")
plt.ylabel("Loan Amount ( In Thousands )")
plt.show()
```

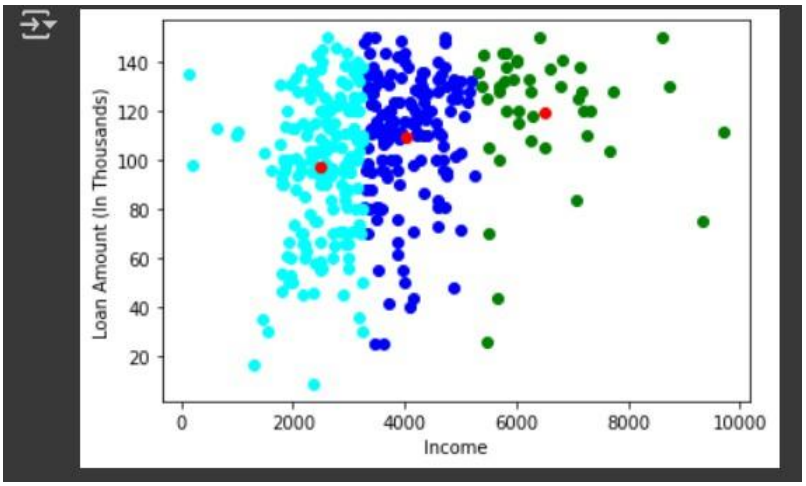



```
—-------------------------------------------------------------------------
diff = 1
j=0

while(diff!=0):
   XD=X
   i=1
   for index1,row_c in Centroids.iterrows():
      ED=[]
      for index2,row_d in XD.iterrows():
         d1=(row_c["ApplicantIncome"]-row_d["ApplicantIncome"])**2
         d2=(row_c["LoanAmount"]-row_d["LoanAmount"])**2
         d=np.sqrt(d1+d2)
         ED.append(d)
      X[i]=ED
      i=i+1

   C=[]
   for index,row in X.iterrows():
      min_dist=row[1]
      pos=1
      for i in range(K):
         if row[i+1] < min_dist:
            min_dist = row[i+1]
            pos=i+1
      C.append(pos)
   X["Cluster"]=C
```

```
   Centroids_new                                                    =
X.groupby(["Cluster"]).mean()[["LoanAmount","ApplicantIncome"]]
   if j == 0:
      diff=1
      j=j+1
   else:
      diff = (Centroids_new['LoanAmount'] -
Centroids['LoanAmount']).sum() + (Centroids_new['ApplicantIncome'] -
Centroids['ApplicantIncome']).sum()
      print(diff.sum())
   Centroids =
X.groupby(["Cluster"]).mean()[["LoanAmount","ApplicantIncome"]]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:26: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
760.9691465468836
570.3181122036852
400.02659764803184
405.26981705062997
186.27599218629547
227.7568592993333
255.37125148586168
244.66095351174067
229.06905235705375
218.24897861156342
107.07928213052429
52.84741626127729
98.54724443834282
90.64953219227577
18.274686272279013
9.21023994083339
18.345487493007468
46.27013250786139
0.0
```

```
—----------------------------------------------------------------------------
color=['blue','green','cyan']
for k in range(K):
   data=X[X["Cluster"]==k+1]
   plt.scatter(data["ApplicantIncome"],data["LoanAmount"],c=color[k])
plt.scatter(Centroids["ApplicantIncome"],Centroids["LoanAmount"],c='re
d')
plt.xlabel('Income')
plt.ylabel('Loan Amount (In Thousands)')
plt.show()
```


Loan Amount (In Thousands)
140
120
100
80
60
40
20
0
2000
4000
6000
8000
10000
Income

Practical No.5

Perform the Linear regression on the given data warehouse data using R/Python.

Regression

- **In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables**

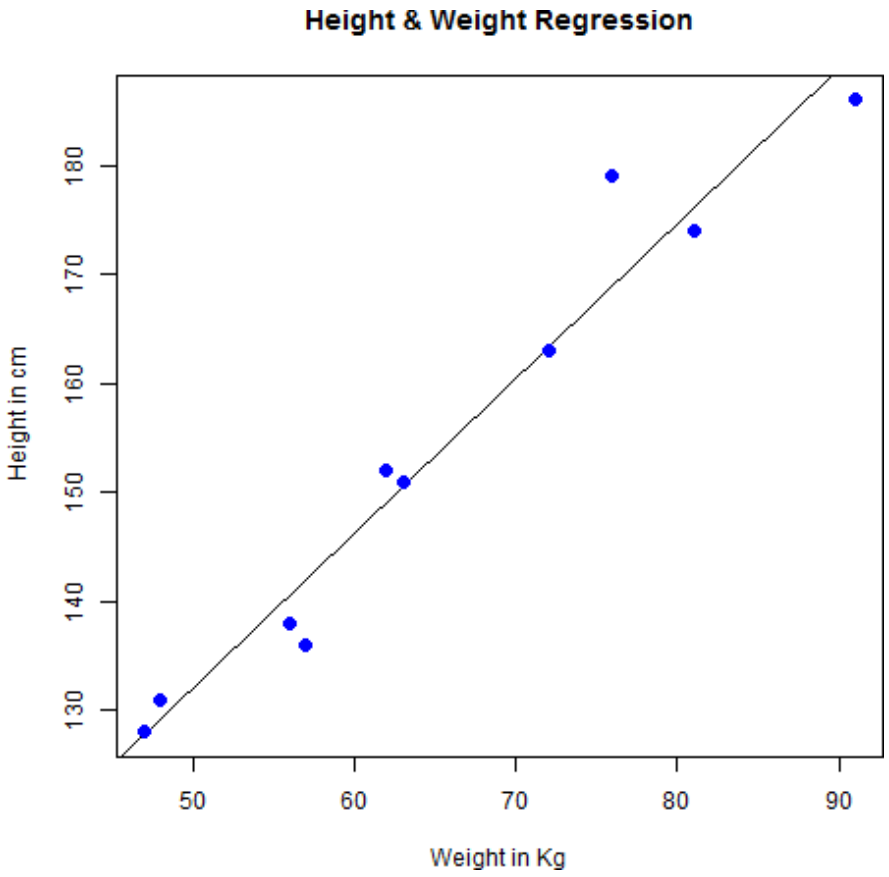
Linear Regression

- **In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1.**
- **y = ax + b is an equation for linear regression.**
- **Where, y is the response variable, x is the predictor variable and *a and b are constants which are called the coefficients.***

lm() Function

- **In R, the lm(), or "linear model," function can be used to create a simple regression model. The lm() function accepts a number of arguments ("Fitting Linear Models," n.d.).**

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

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

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

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

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

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


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

Practical No.6

Perform the logistic regression on the given data warehouse data using R/Python.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
—----------------------------------------------------------------------
dataset=pd.read_csv("/content/User_Data.csv")
—----------------------------------------------------------------------
x = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
—----------------------------------------------------------------------
x
```

```
array([[    19,  19000],
       [    35,  20000],
       [    26,  43000],
       [    27,  57000],
       [    19,  76000],
       [    27,  58000],
       [    27,  84000],
       [    32, 150000],
       [    25,  33000],
       [    35,  65000],
       [    26,  80000],
       [    26,  52000],
       [    20,  86000],
       [    32,  18000],
       [    18,  82000],
       [    29,  80000],
       [    47,  25000],
       [    45,  26000],
       [    46,  28000],
       [    48,  29000],
       [    45,  22000],
       [    47,  49000],
       [    48,  41000],
```

```
—----------------------------------------------------------------------
y
```

```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1])
```

```
—----------------------------------------------------------------------
```

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.25, random_state = 0)
—----------------------------------------------------------------------
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
xtrain = sc_x.fit_transform(xtrain)
xtest = sc_x.transform(xtest)
print (xtrain)
```

```
[[ 0.58164944 -0.88670699]
 [-0.60673761  1.46173768]
 [-0.01254409 -0.5677824 ]
 [-0.60673761  1.89663484]
 [ 1.37390747 -1.40858358]
 [ 1.47293972  0.99784738]
 [ 0.08648817 -0.79972756]
 [-0.01254409 -0.24885782]
 [-0.21060859 -0.5677824 ]
 [-0.21060859 -0.19087153]
 [-0.30964085 -1.29261101]
 [-0.30964085 -0.5677824 ]
 [ 0.38358493  0.09905991]
 [ 0.8787462  -0.59677555]
 [ 2.06713324 -1.17663843]
 [ 1.07681071 -0.13288524]
 [ 0.68068169  1.78066227]
 [-0.70576986  0.56295021]
 [ 0.77971394  0.35999821]
 [ 0.8787462  -0.53878926]
 [-1.20093113 -1.58254245]
```

```
—----------------------------------------------------------------------
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(xtrain, ytrain)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
—----------------------------------------------------------------------
y_pred = classifier.predict(xtest)
—----------------------------------------------------------------------
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(ytest, y_pred)
print ("Confusion Matrix : \n", cm)
```

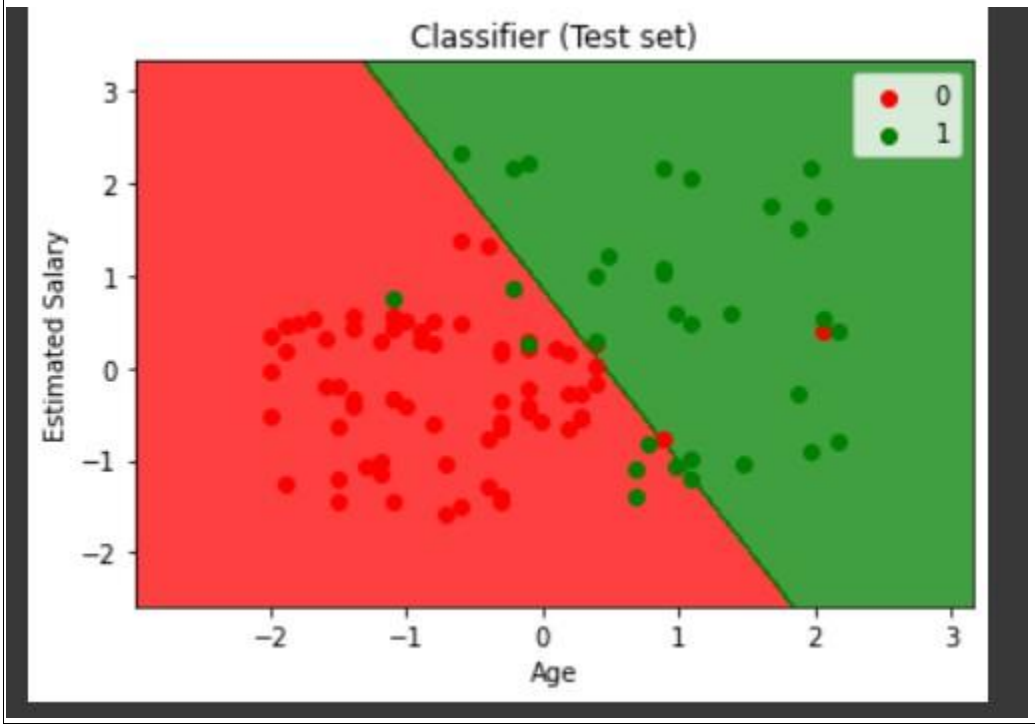
```
Confusion Matrix :
 [[65  3]
 [ 8 24]]
```

```
—----------------------------------------------------------------------
```

```
from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(ytest, y_pred))
```

```
Accuracy :  0.89
```

```
__----------------------------------------------------------------------
from matplotlib.colors import ListedColormap
X_set, y_set = xtest, ytest
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1,
                               stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1,
                               stop = X_set[:, 1].max() + 1, step = 0.01))

plt.contourf(X1, X2, classifier.predict(
             np.array([X1.ravel(), X2.ravel()]).T).reshape(
             X1.shape), alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Classifier (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```



Practical No.7

Write a Python program to read data from a CSV file, perform simple data analysis, and generate basic insights. (Use Pandas is a Python library).

```
import pandas as pd

# Step 1: Read the CSV file into a DataFrame
df = pd.read_csv("/content/data.csv")
—-------------------------------------------------------------------------
# Show the first 5 rows of the DataFrame
print("First 5 rows of the data:")
print(df.head())

# Step 3: Data summary and descriptive statistics
print("\nData Summary (info):")
print(df.info())

print("\nDescriptive Statistics:")
print(df.describe())

# Step 4: Count of unique values for categorical columns (Gender)
print("\nCount of unique values in 'Gender':")
print(df['Gender'].value_counts())

# Step 5: Average Salary by Gender
print("\nAverage Salary by Gender:")
print(df.groupby('Gender')['Salary'].mean())

# Step 6: Check for missing values
print("\nMissing values in each column:")
print(df.isnull().sum())

# Step 7: Find the highest and lowest salary
print("\nHighest Salary:")
print(df[df['Salary'] == df['Salary'].max()])

print("\nLowest Salary:")
print(df[df['Salary'] == df['Salary'].min()])

# Step 9: Filter data (e.g., Employees with Salary > 50000)
print("\nEmployees with Salary greater than 50000:")
print(df[df['Salary'] > 50000])
```

```
First 5 rows of the data:
   Name  Age  Gender  Salary
0  John   28    Male   50000
1  Jane   34  Female   60000
2   Doe   23    Male   45000
3  Anna   41  Female   70000
4   Tom   29    Male   52000

Data Summary (info):
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    5 non-null      object
 1   Age     5 non-null      int64
 2   Gender  5 non-null      object
 3   Salary  5 non-null      int64
dtypes: int64(2), object(2)
memory usage: 292.0+ bytes
None

Descriptive Statistics:
             Age        Salary
count   5.000000      5.000000
mean   31.000000  55400.000000
std     6.819091   9787.747443
min    23.000000  45000.000000
25%    28.000000  50000.000000
50%    29.000000  52000.000000
75%    34.000000  60000.000000
max    41.000000  70000.000000

Count of unique values in 'Gender':
Gender
Male      3
Female    2
Name: count, dtype: int64
```

```
Name: count, dtype: int64

Average Salary by Gender:
Gender
Female    65000.0
Male      49000.0
Name: Salary, dtype: float64

Missing values in each column:
Name      0
Age       0
Gender    0
Salary    0
dtype: int64

Highest Salary:
   Name  Age  Gender  Salary
3  Anna   41  Female   70000

Lowest Salary:
  Name  Age Gender  Salary
2  Doe   23   Male   45000

Employees with Salary greater than 50000:
   Name  Age  Gender  Salary
1  Jane   34  Female   60000
3  Anna   41  Female   70000
4   Tom   29    Male   52000
```

Practical No.9

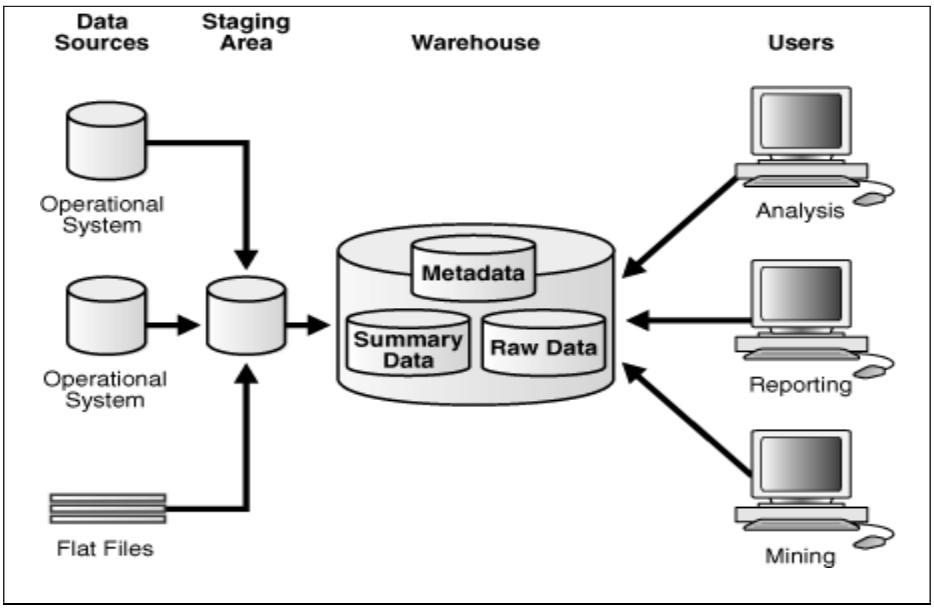
a. Create the Data staging area for the selected database.

b. Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.

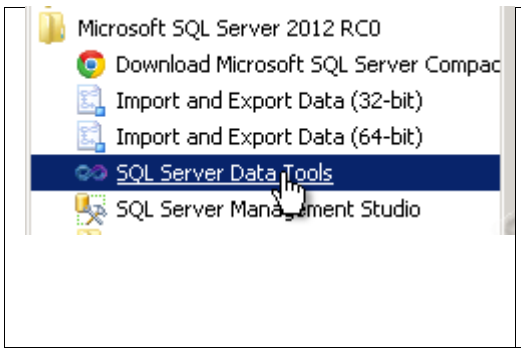
Practical No.3(a). Create the Data staging area for the selected database.

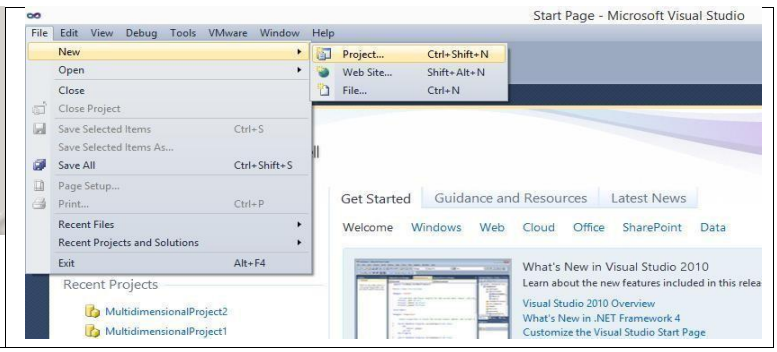
Staging Area:

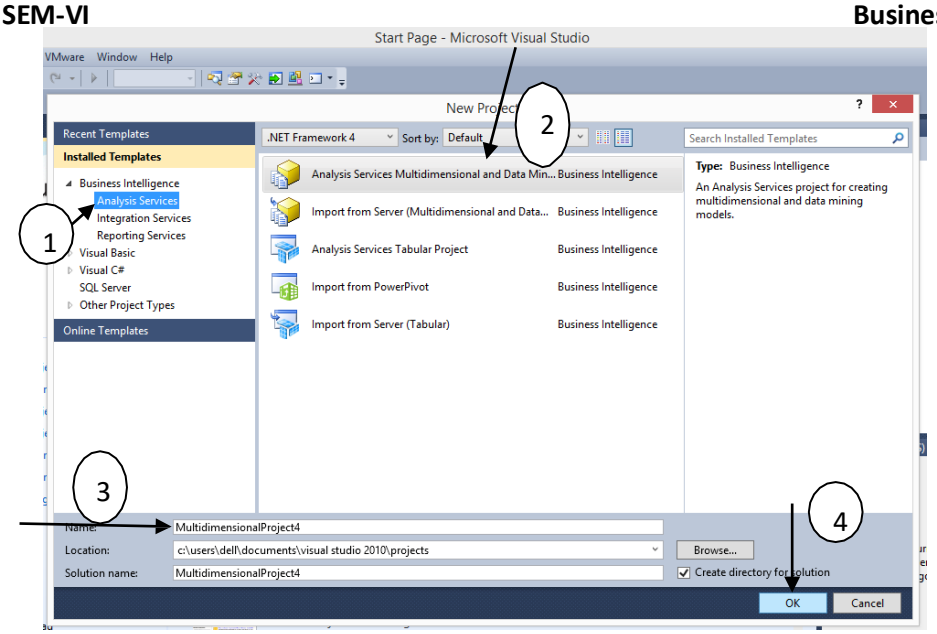
A staging area, or landing zone, is an intermediate storage area used for data processing during the extract, transform and load (ETL) process. The data staging area sits between the data source(s) and the data target(s), which are often data warehouses, data marts, or other data repositories



To create a staging area, we have to use SSDT tool and create a new project as

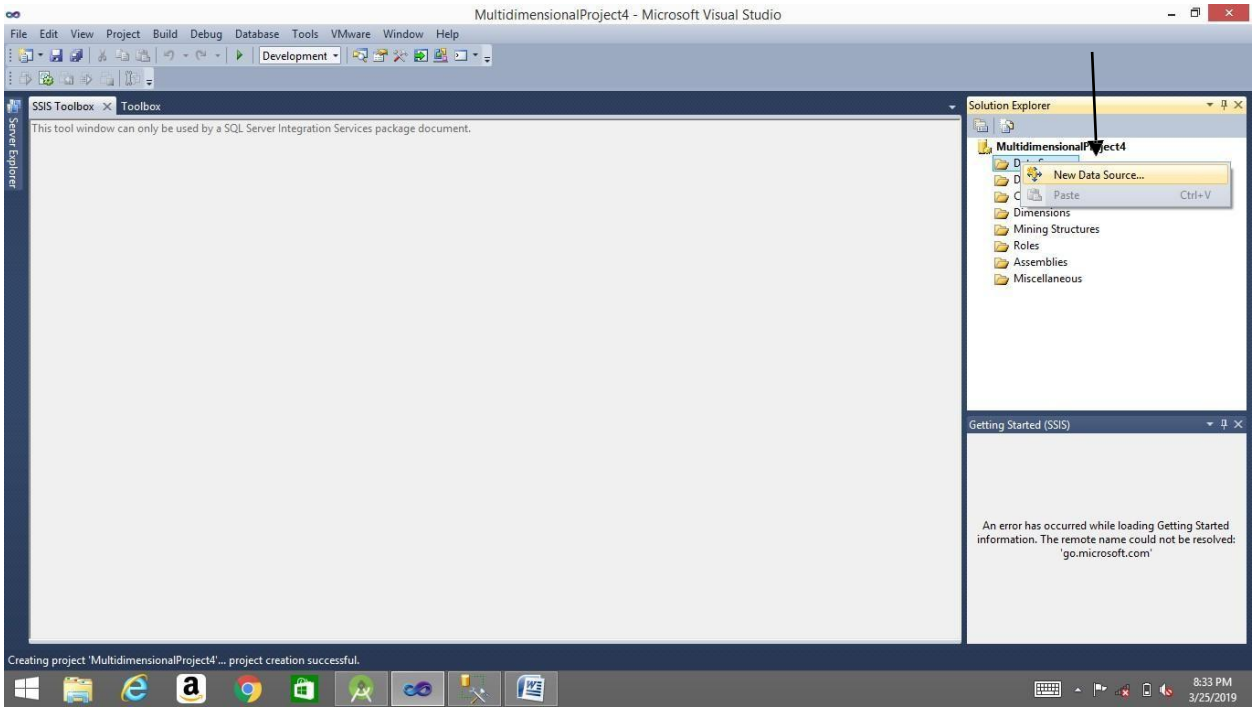


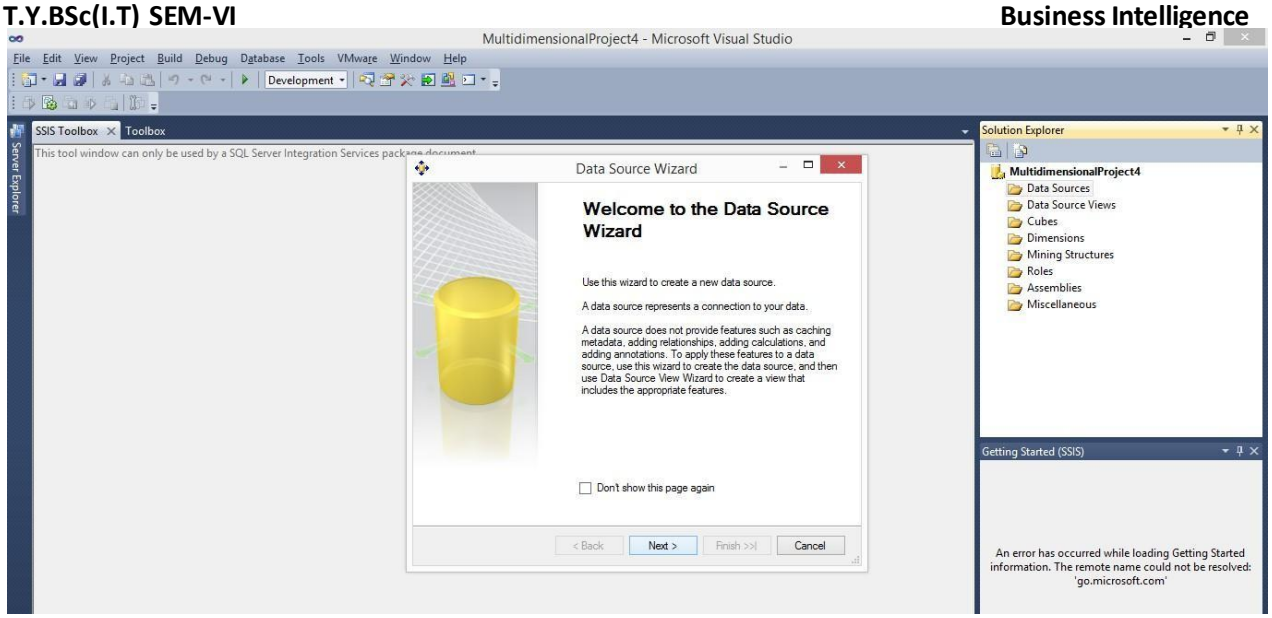


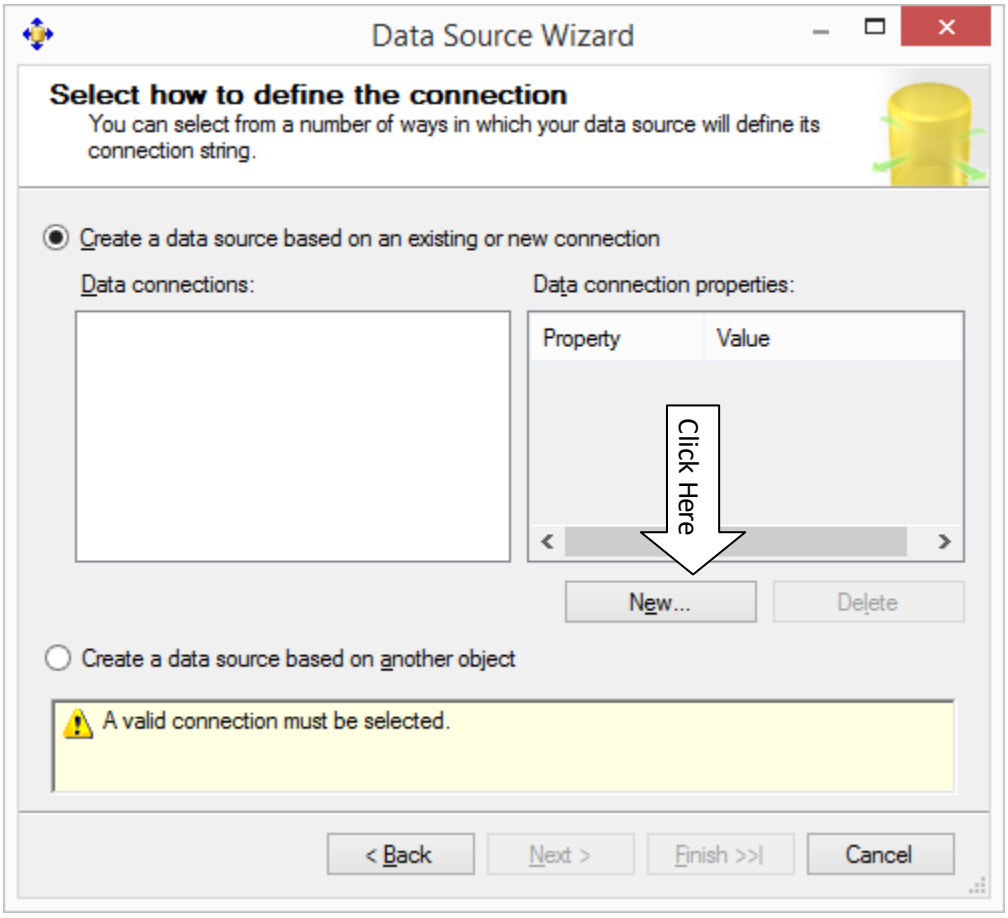


Step1 : Create a Data Source For Staging Area.

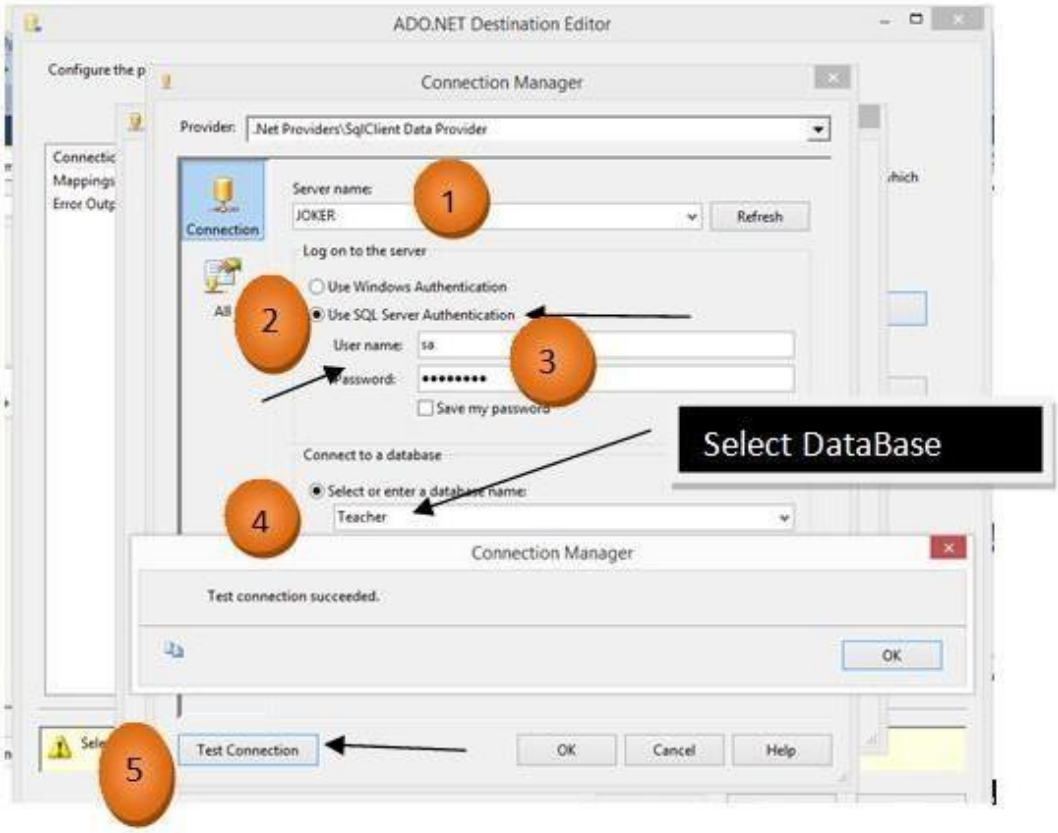
Right Click on Data Source Option and Select New Data Source from Solution Explorer as



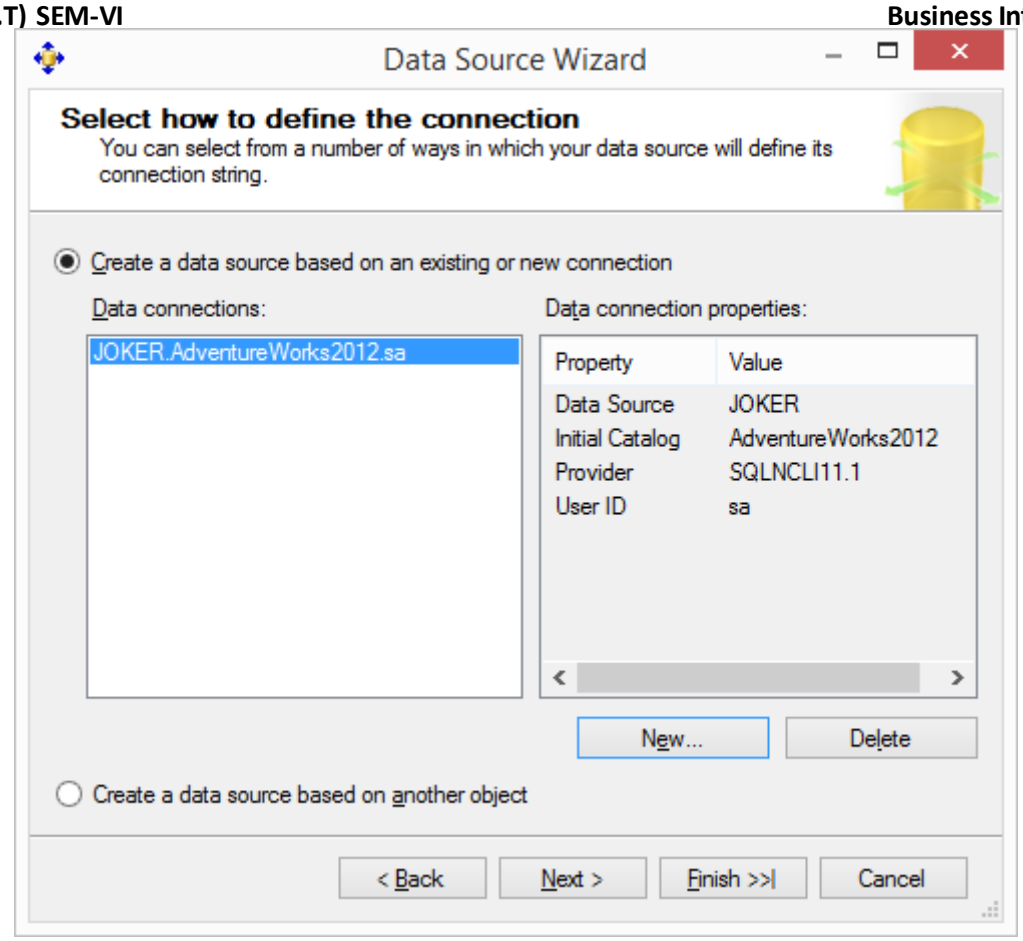

MultidimensionalProject4 - Microsoft Visual Studio
File Edit View Project Build Debug Database Tools VMware Window Help
Development
SSIS Toolbox
Toolbox
This tool window can only be used by a SQL Server Integration Services package document.
Data Source Wizard
Welcome to the Data Source Wizard
Use this wizard to create a new data source.
A data source represents a connection to your data.
A data source does not provide features such as caching metadata, adding relationships, adding calculations, and adding annotations. To apply these features to a data source, use this wizard to create the data source, and then use Data Source View Wizard to create a view that includes the appropriate features.
Don't show this page again
< Back
Next >
Finish >>|
Cancel
Solution Explorer
MultidimensionalProject4
Data Sources
Data Source Views
Cubes
Dimensions
Mining Structures
Roles
Assemblies
Miscellaneous
Getting Started (SSIS)
An error has occurred while loading Getting Started information. The remote name could not be resolved: 'go.microsoft.com'


Data Source Wizard
Select how to define the connection
You can select from a number of ways in which your data source will define its connection string.
Create a data source based on an existing or new connection
Data connections:
Data connection properties:
Property
Value
Click Here
New...
Delete
Create a data source based on another object
A valid connection must be selected.
< Back
Next >
Finish >>|
Cancel

Enter The SQL Server instance name = JOKER in this case

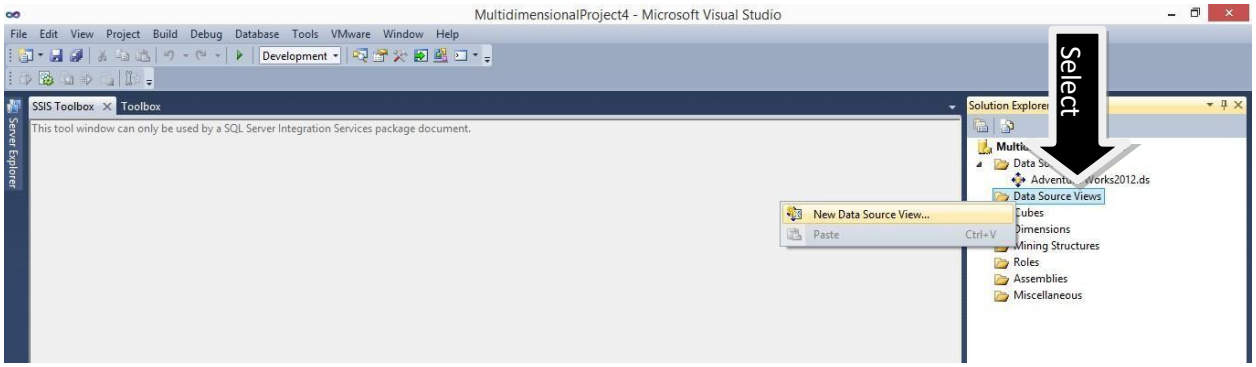


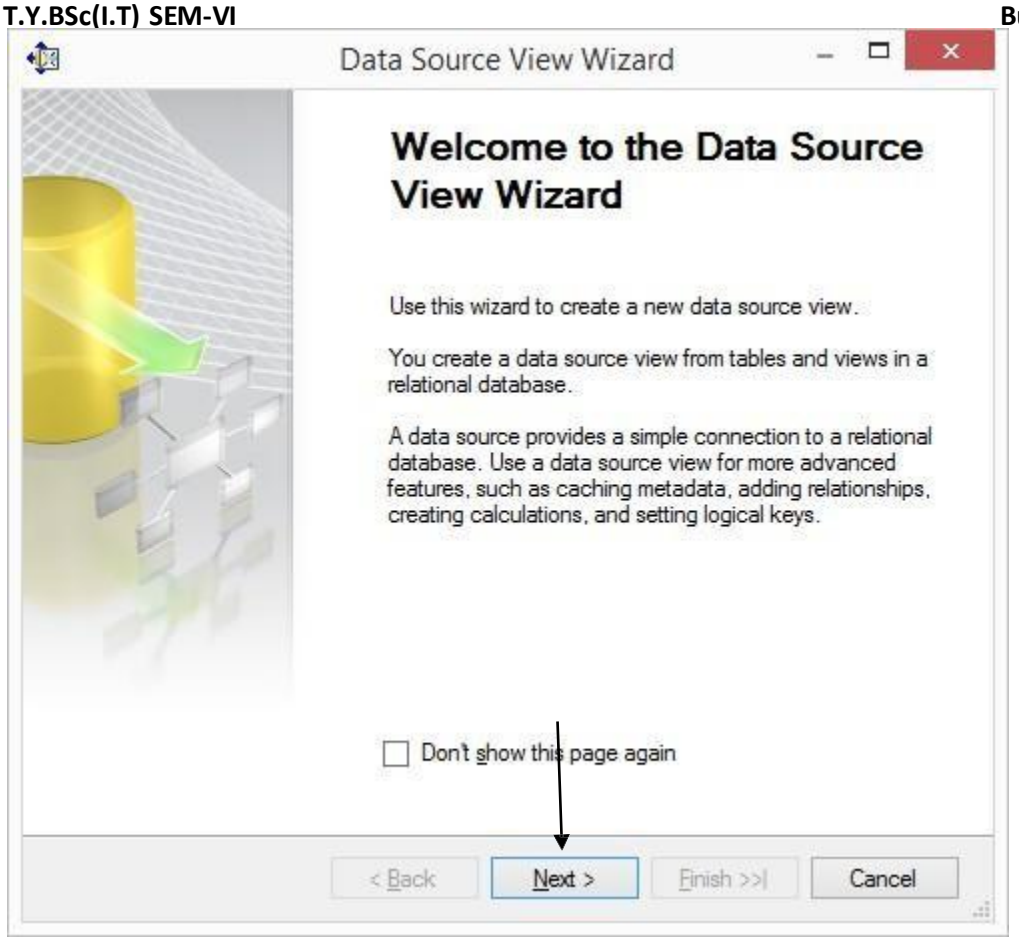
Click Ok

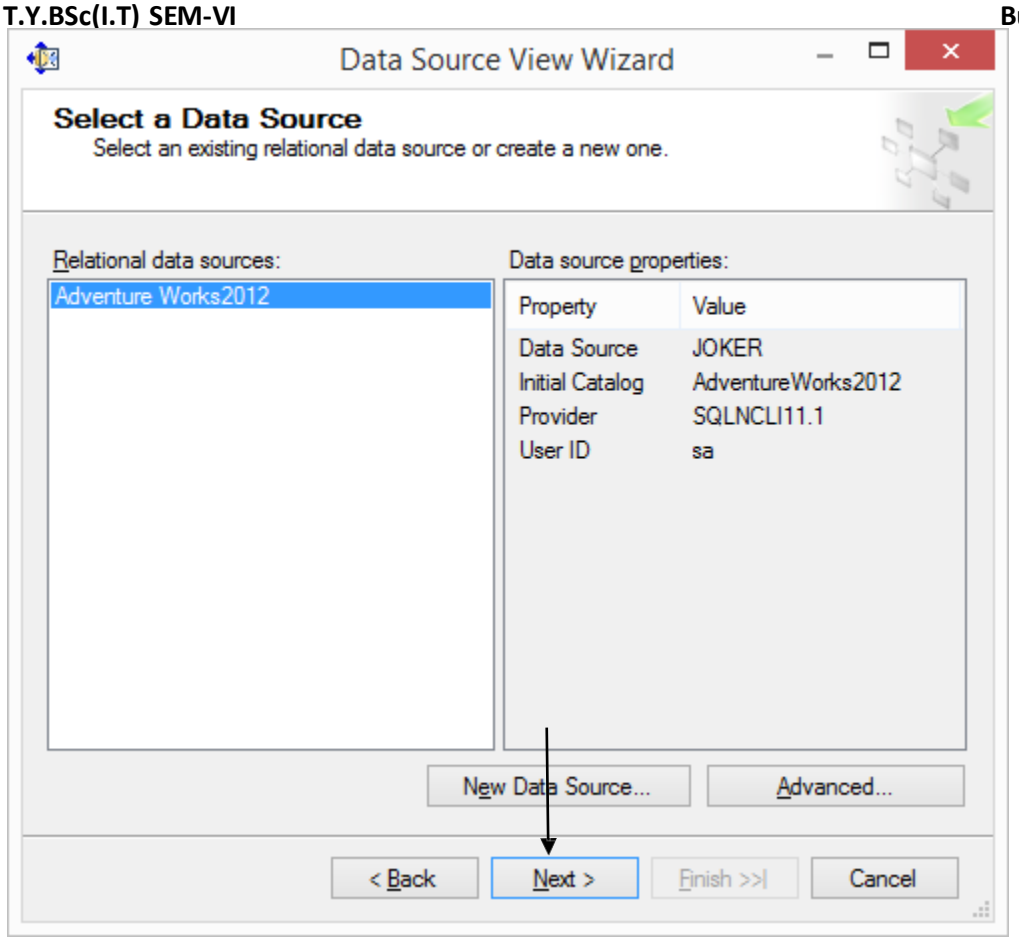


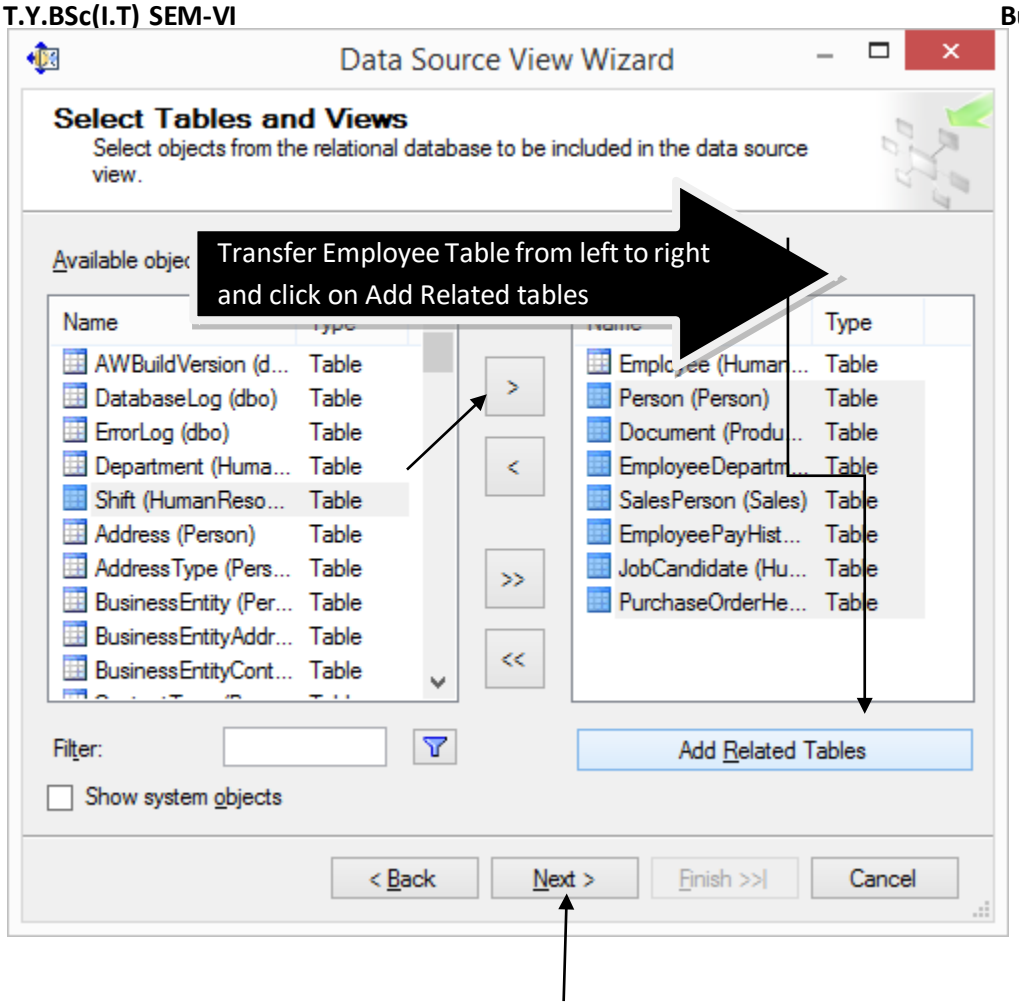
Step2: Now we will create data source View to visualize the Data Source.

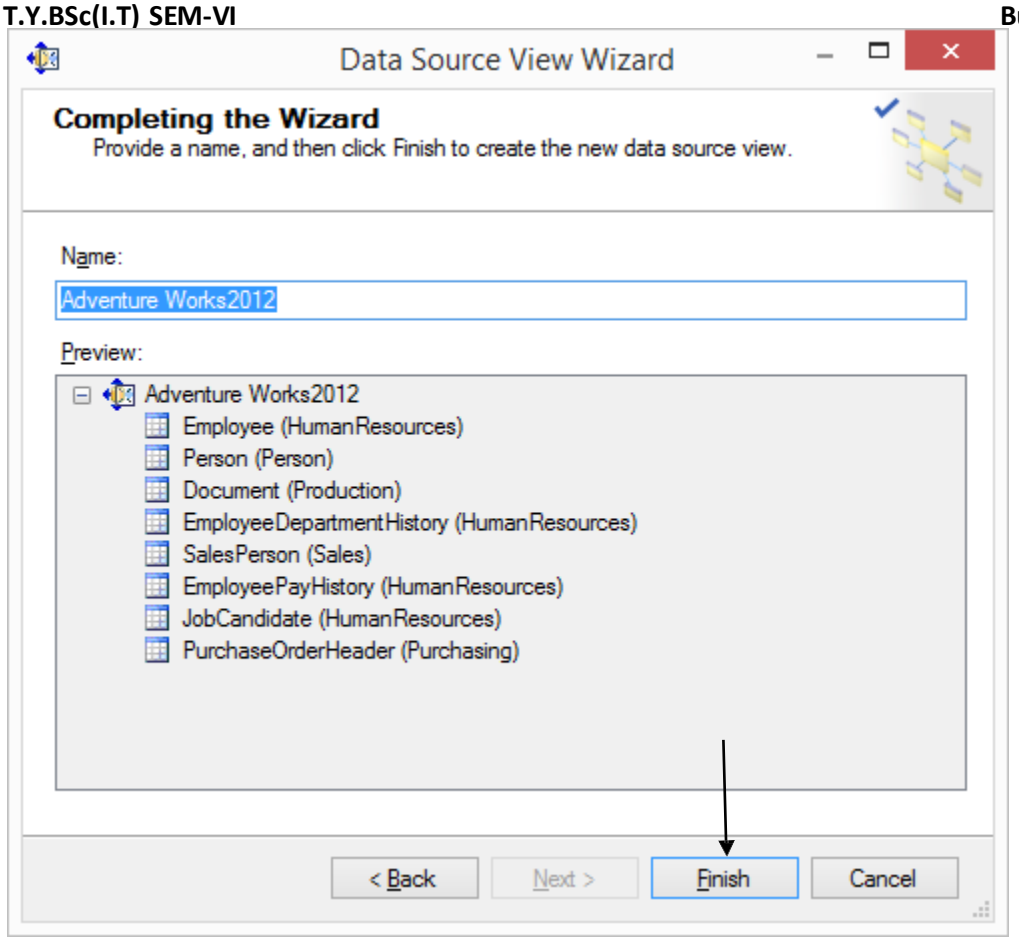
For this right click on Data Source View Option and select New Data Source View

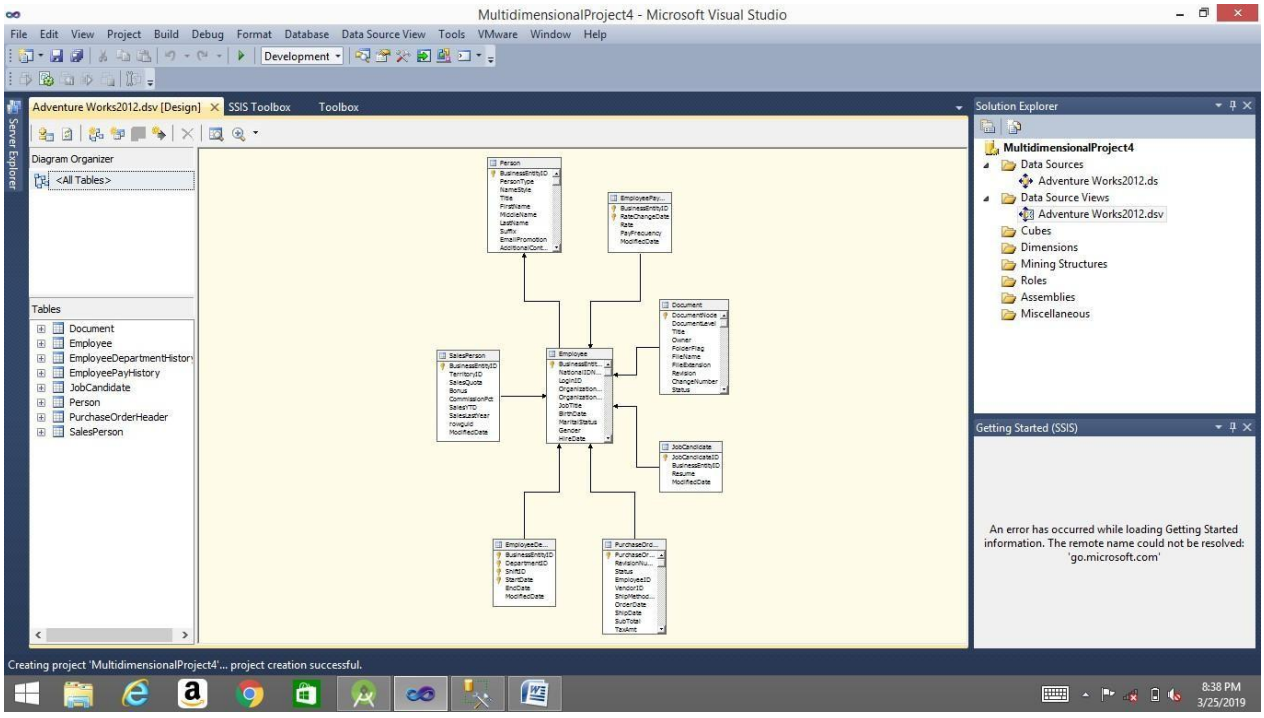



Data Source View Wizard
Welcome to the Data Source View Wizard
Use this wizard to create a new data source view.
You create a data source view from tables and views in a relational database.
A data source provides a simple connection to a relational database. Use a data source view for more advanced features, such as caching metadata, adding relationships, creating calculations, and setting logical keys.
Don't show this page again
< Back
Next >
Finish >>|
Cancel


Data Source View Wizard
Select a Data Source
Select an existing relational data source or create a new one.
Relational data sources:
Adventure Works2012
Data source properties:
Property
Value
Data Source
JOKER
Initial Catalog
AdventureWorks2012
Provider
SQLNCLI11.1
User ID
sa
New Data Source...
Advanced...
< Back
Next >
Finish >>|
Cancel


Data Source View Wizard
Select Tables and Views
Select objects from the relational database to be included in the data source view.
Transfer Employee Table from left to right and click on Add Related tables
Available objec
Name
Type
AWBuildVersion (d... Table
DatabaseLog (dbo) Table
ErrorLog (dbo) Table
Department (Huma... Table
Shift (HumanReso... Table
Address (Person) Table
AddressType (Pers... Table
BusinessEntity (Per... Table
BusinessEntityAddr... Table
BusinessEntityCont... Table
Person (Person) Table
Document (Produ... Table
EmployeeDepartm... Table
SalesPerson (Sales) Table
EmployeePayHist... Table
JobCandidate (Hu... Table
PurchaseOrderHe... Table
>
<
>>
<<
Filter:
Add Related Tables
Show system objects
< Back
Next >
Finish >>|
Cancel





This is a Staging area showing in START SCHEMA.

Practical No.3(a) done.

Practical No.3(b). Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.

Dimension Table

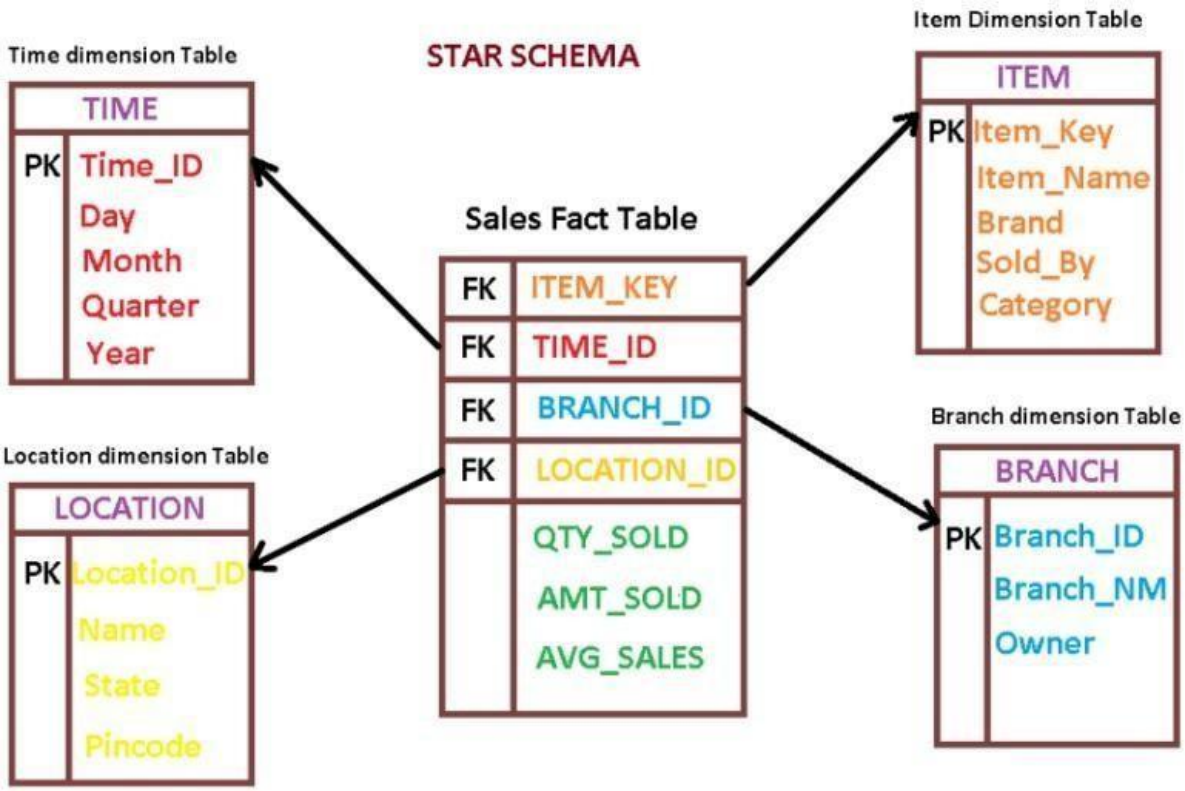
A Dimension Table is a table in a star schema of a data warehouse. Data warehouses are built using dimensional data models which consist of fact and dimension tables. Dimension tables are used to describe dimensions; they contain dimension keys, values and attributes.

Fact Table

A fact table is found at the center of a star schema or snowflake schema surrounded by dimension tables. A fact table consists of facts of a particular business process e.g., sales revenue by month by product. Facts are also known as measurements or metrics. A fact table record captures a measurement or a metric.

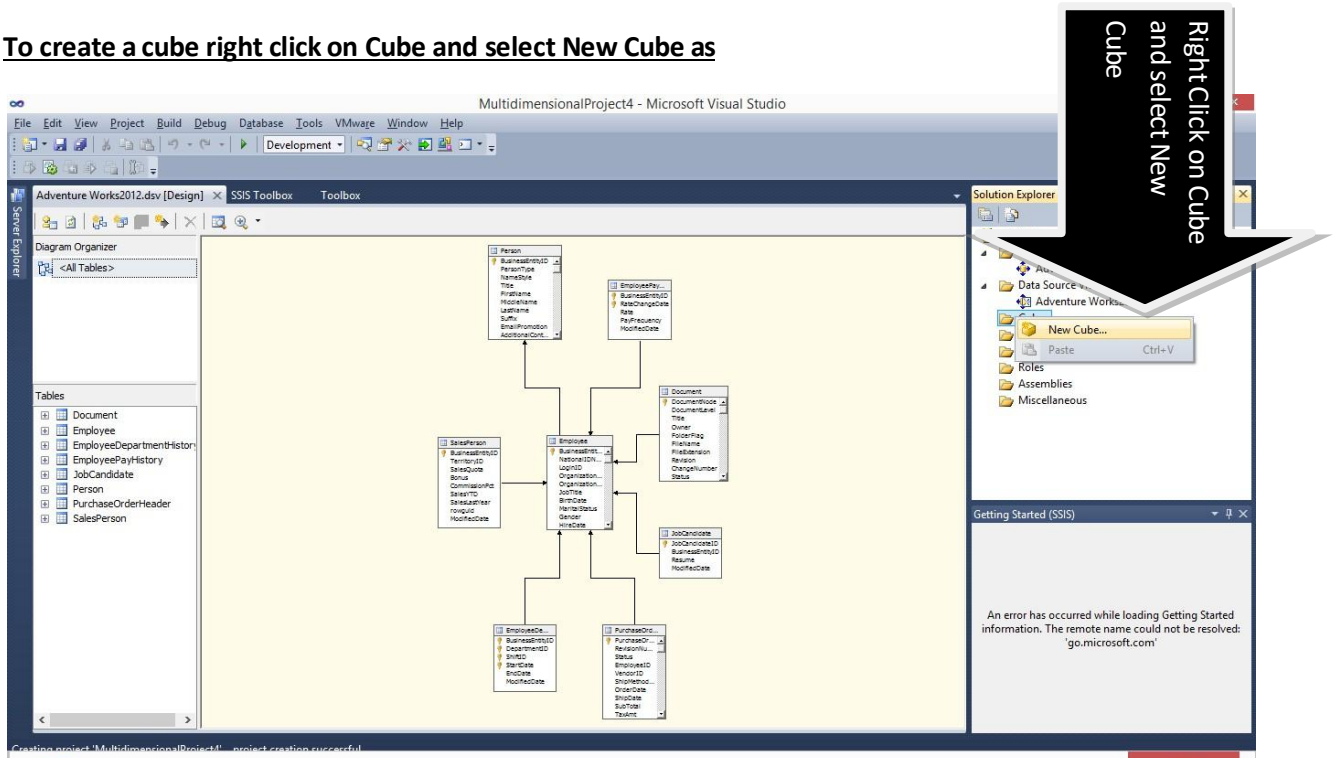
Cube

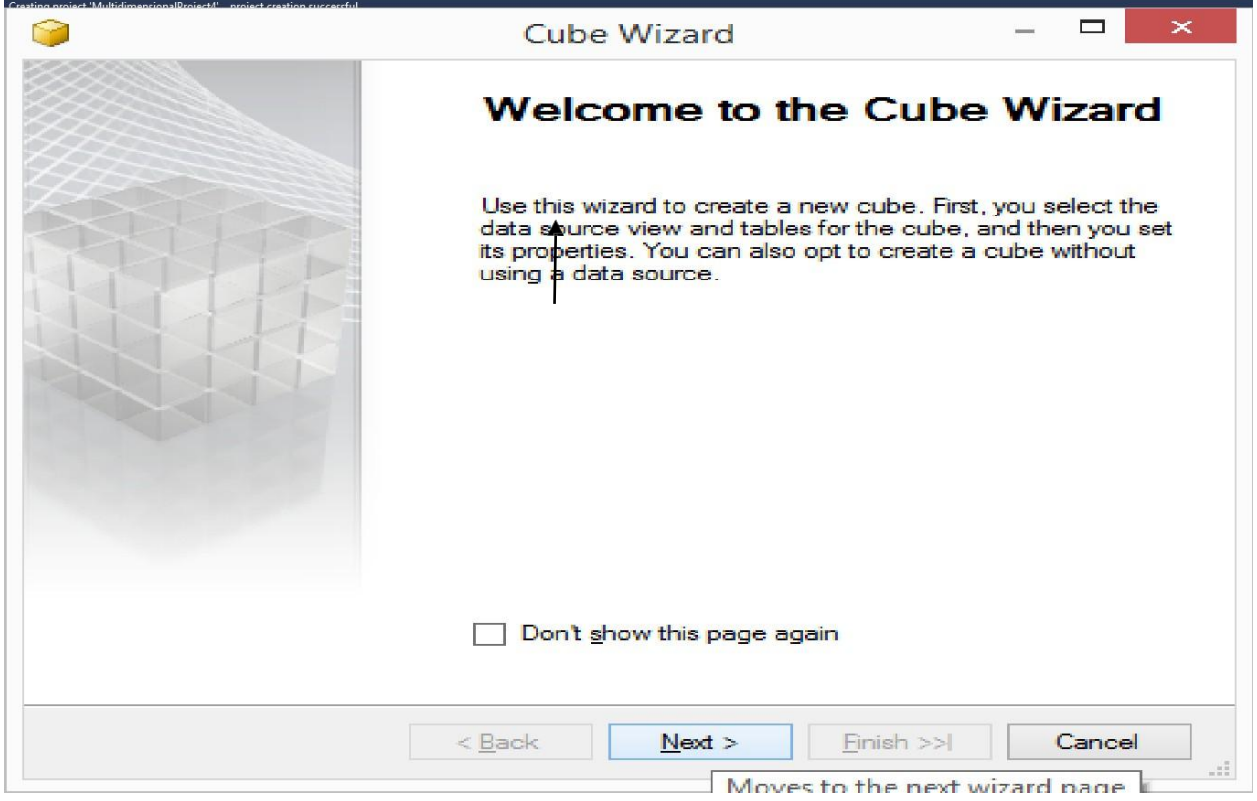
Cubes are data processing units composed of fact tables and dimensions from the data warehouse. They provide multidimensional views of data, querying and analytical capabilities to clients.

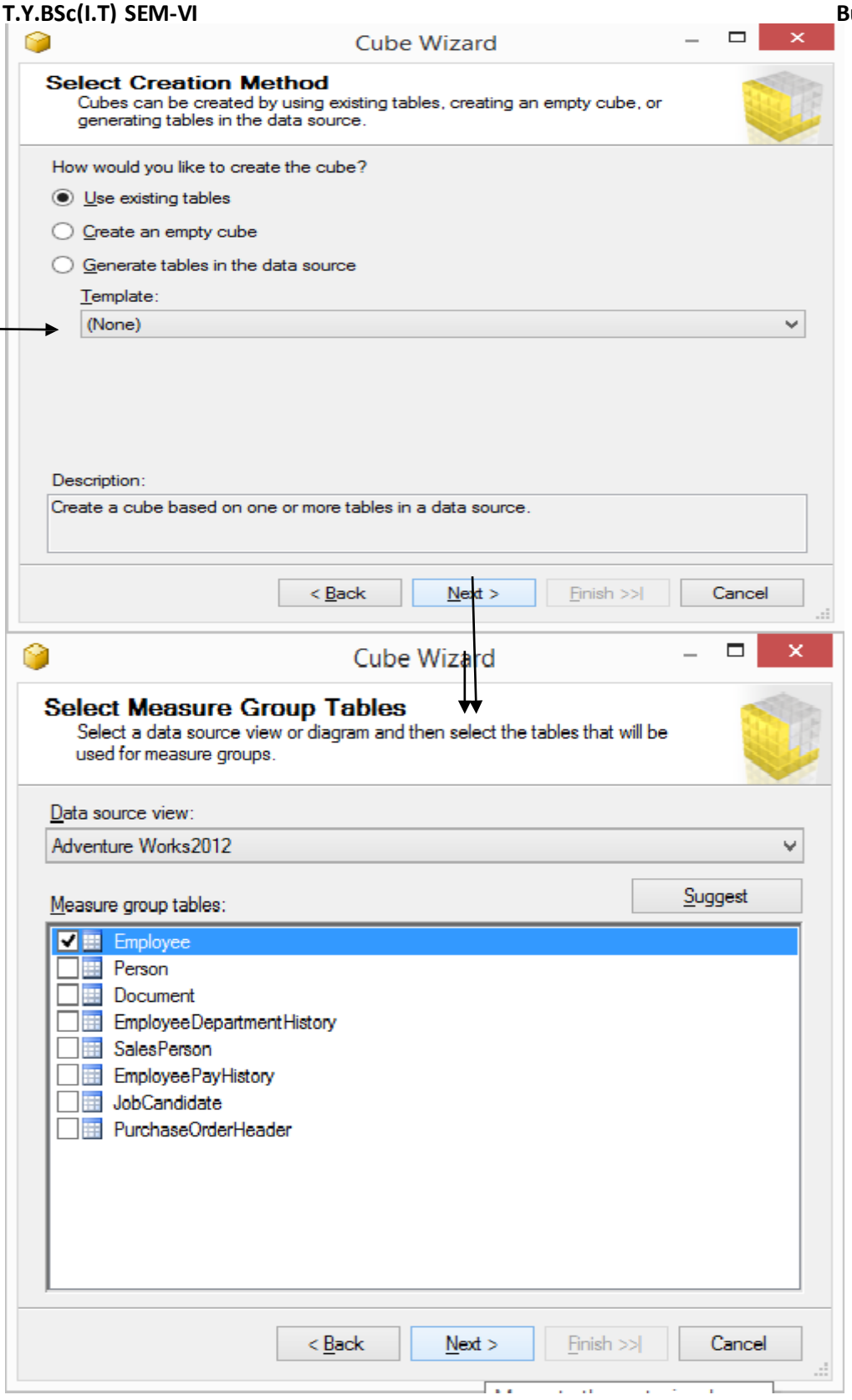


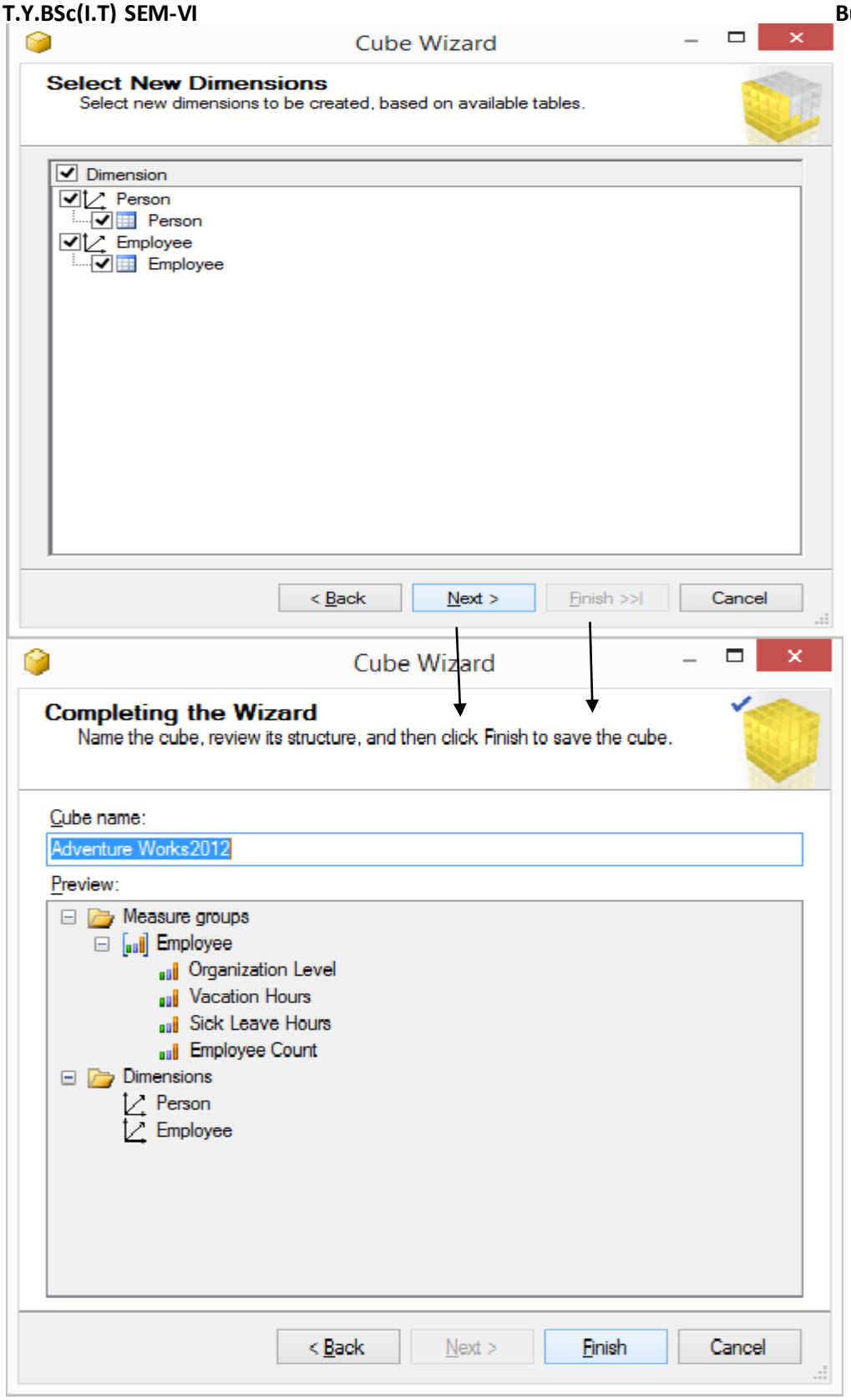
Creating a Cube in SSDT

To create a cube right click on Cube and select New Cube as

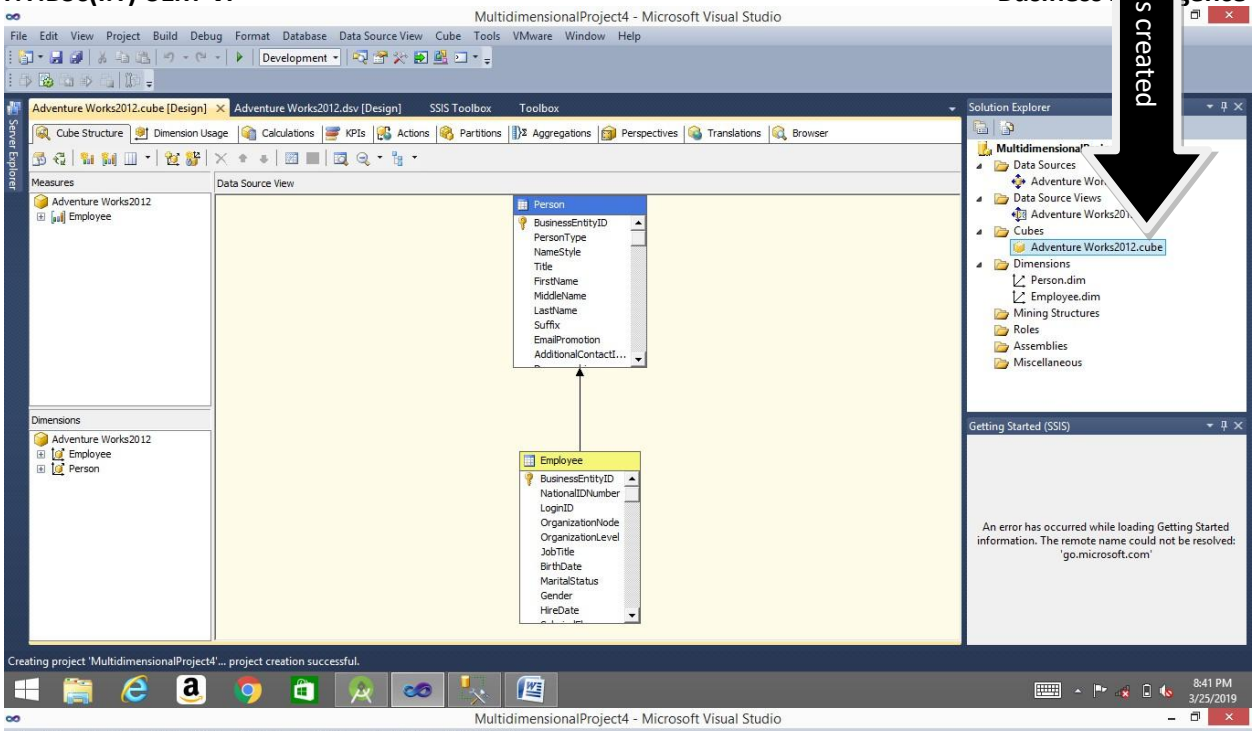


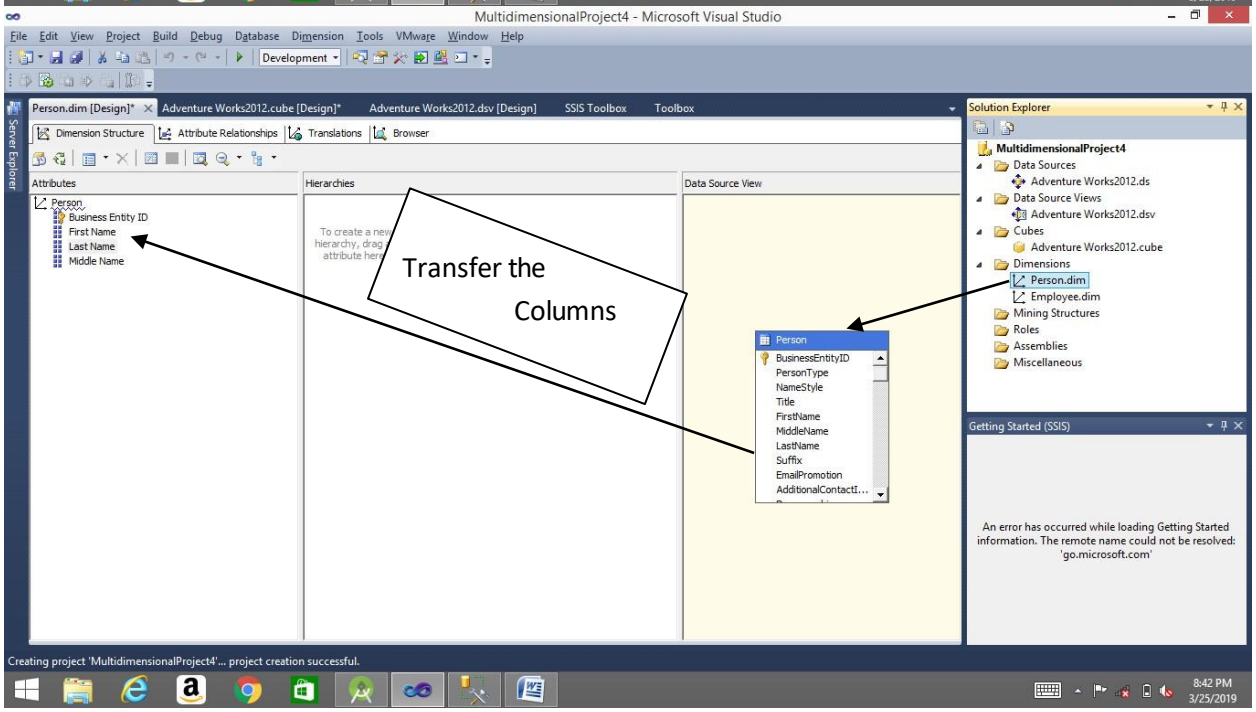


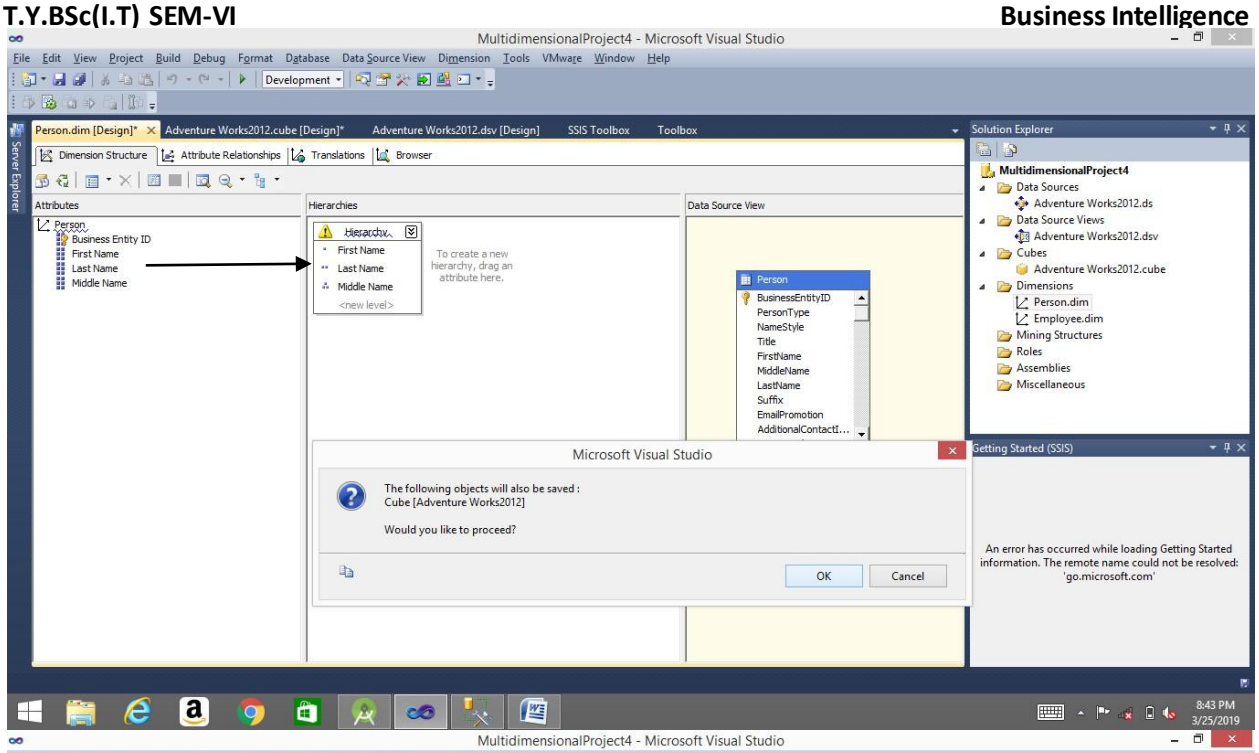

Cube Wizard
Select Creation Method
Cubes can be created by using existing tables, creating an empty cube, or generating tables in the data source.
How would you like to create the cube?
Use existing tables
Create an empty cube
Generate tables in the data source
Template:
(None)
Description:
Create a cube based on one or more tables in a data source.
< Back
Next >
Finish >>|
Cancel
Cube Wizard
Select Measure Group Tables
Select a data source view or diagram and then select the tables that will be used for measure groups.
Data source view:
Adventure Works2012
Suggest
Measure group tables:
Employee
Person
Document
EmployeeDepartmentHistory
SalesPerson
EmployeePayHistory
JobCandidate
PurchaseOrderHeader
< Back
Next >
Finish >>|
Cancel

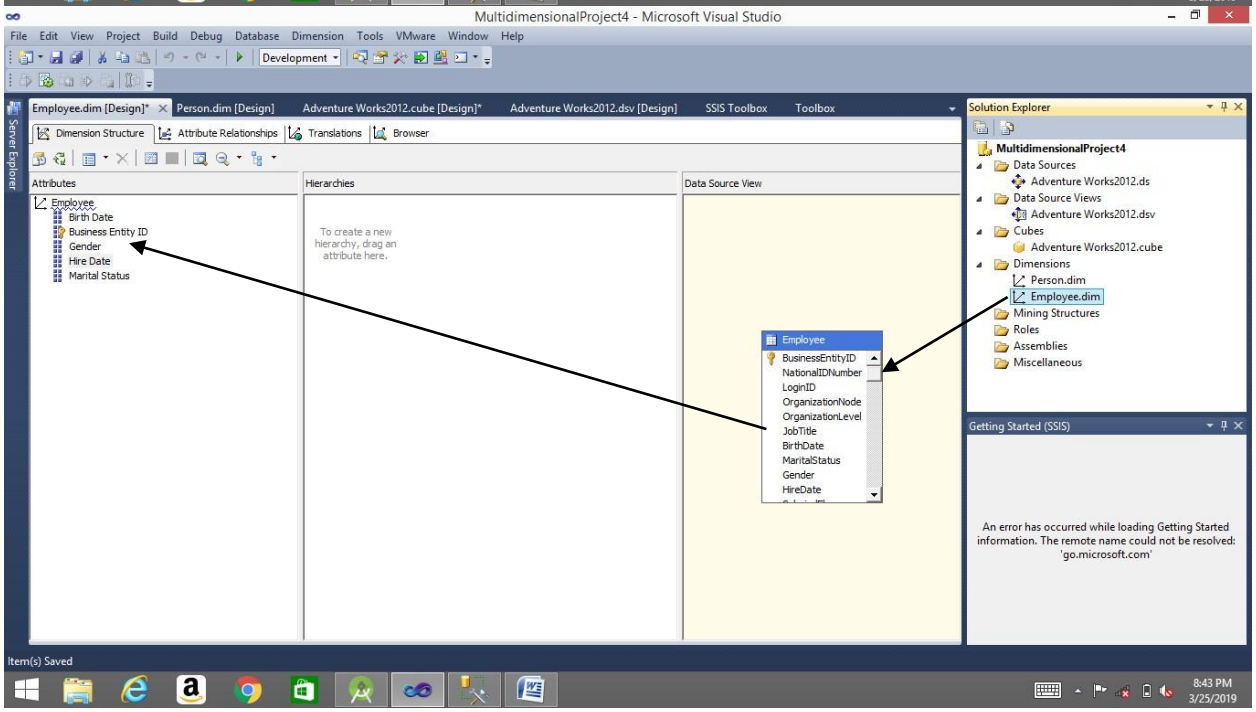

Cube Wizard
Select New Dimensions
Select new dimensions to be created, based on available tables.
Dimension
Person
Person
Employee
Employee
< Back
Next >
Finish >>|
Cancel
Cube Wizard
Completing the Wizard
Name the cube, review its structure, and then click Finish to save the cube.
Cube name:
Adventure Works2012
Preview:
Measure groups
Employee
Organization Level
Vacation Hours
Sick Leave Hours
Employee Count
Dimensions
Person
Employee
< Back
Next >
Finish
Cancel

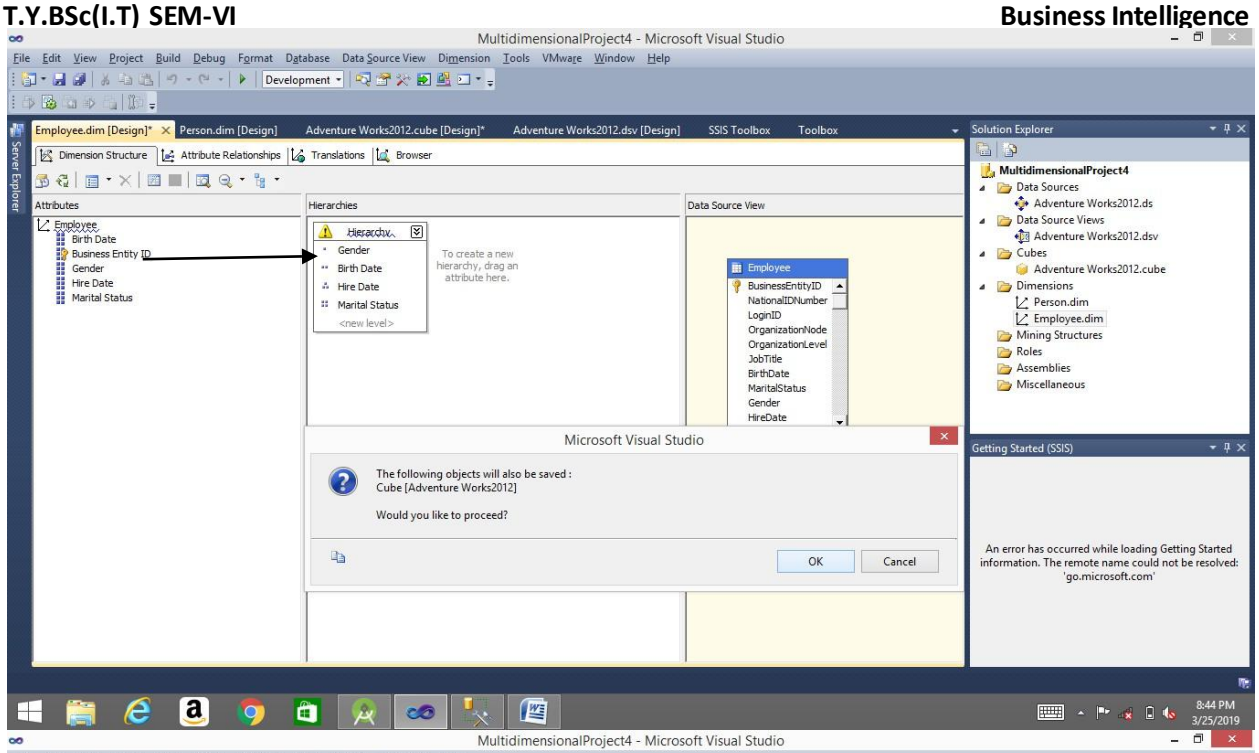
Cube is created

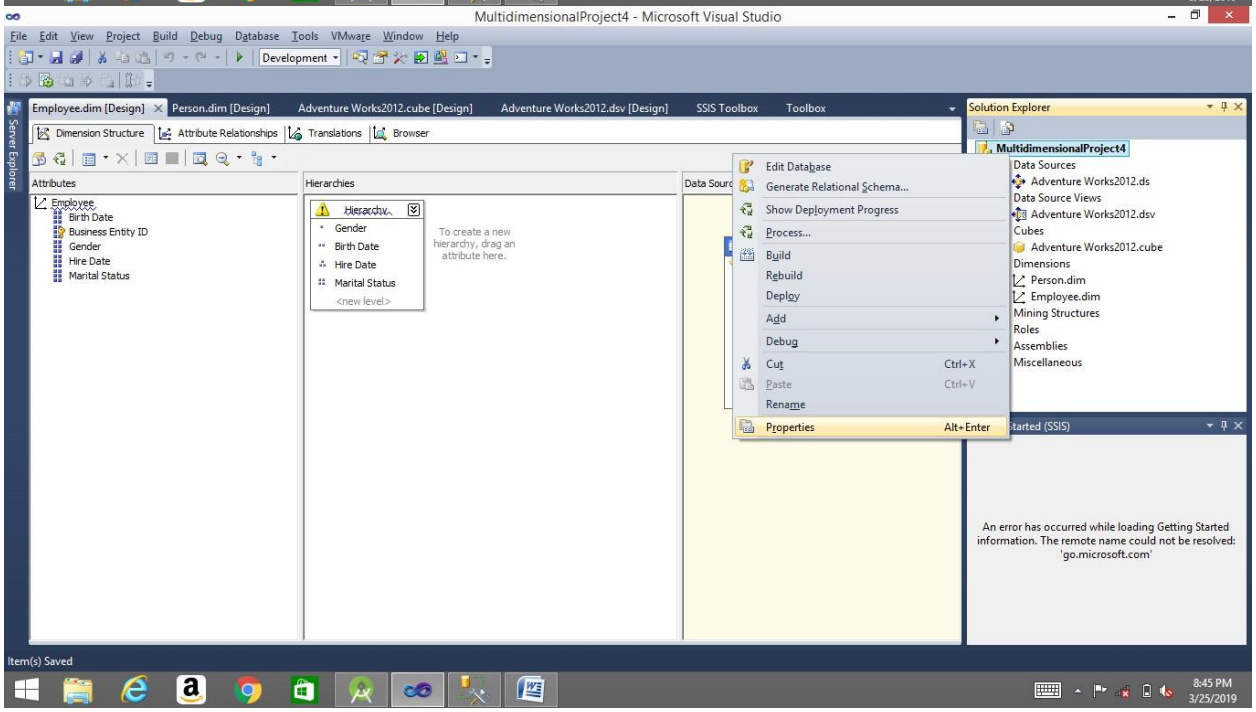


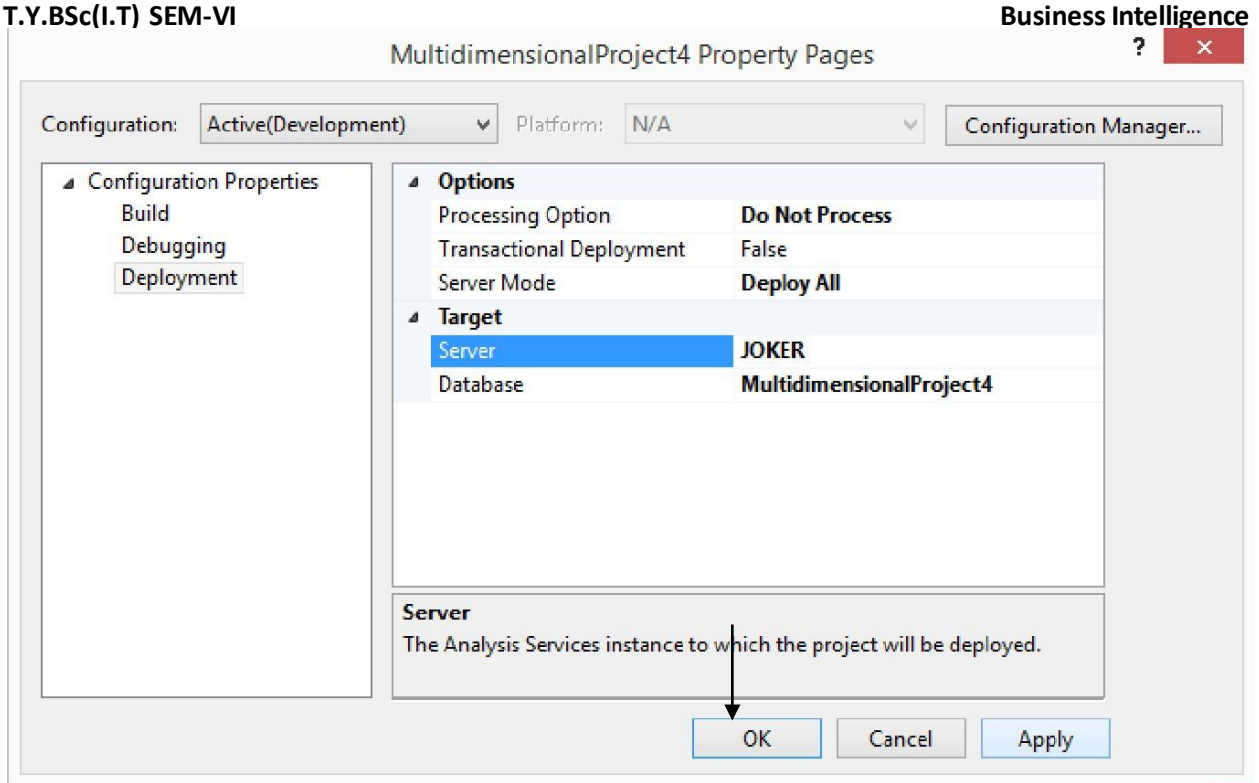


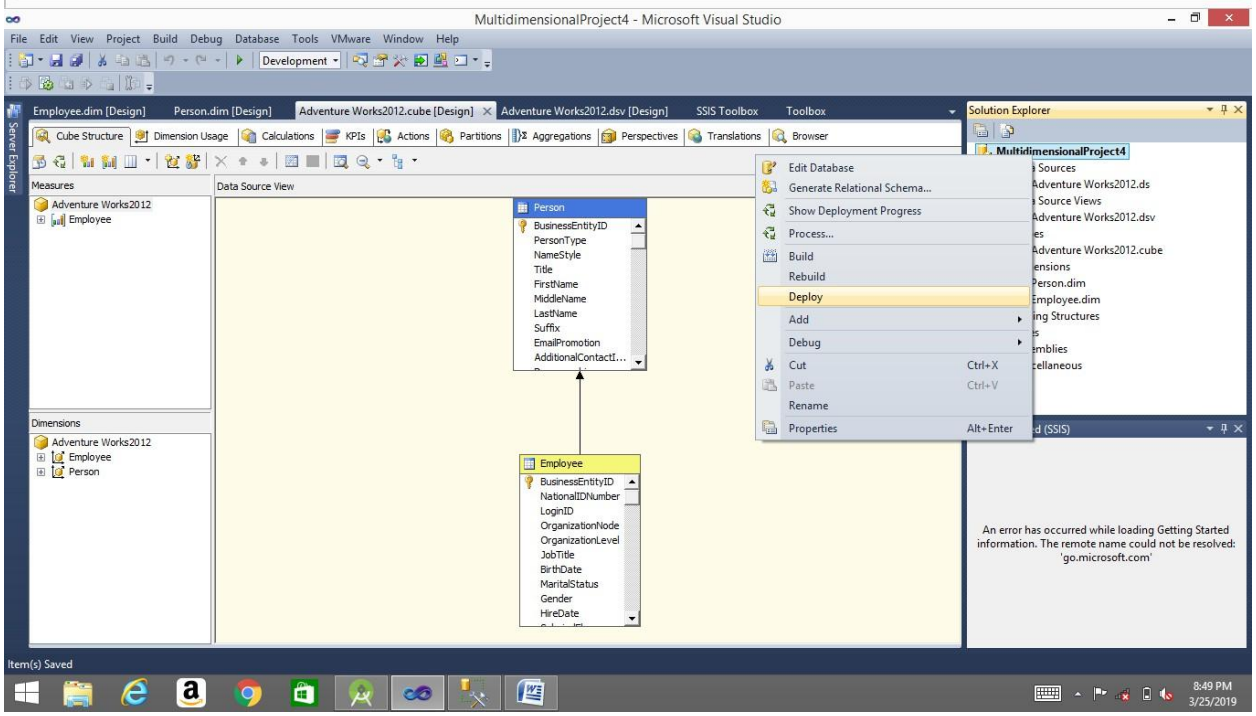

MultidimensionalProject4 - Microsoft Visual Studio
File Edit View Project Build Debug Format Database Data Source View Dimension Tools VMware Window Help
Development
Person.dim [Design]* Adventure Works2012.cube [Design]* Adventure Works2012.dsv [Design] SSIS Toolbox Toolbox
Dimension Structure Attribute Relationships Translations Browser
Attributes
Person
Business Entity ID
First Name
Last Name
Middle Name
Hierarchies
Hierarchy
First Name
Last Name
Middle Name
<new level>
To create a new hierarchy, drag an attribute here.
Data Source View
Person
BusinessEntityID
PersonType
NameStyle
Title
FirstName
MiddleName
LastName
Suffix
EmailPromotion
AdditionalContactI...
Solution Explorer
MultidimensionalProject4
Data Sources
Adventure Works2012.ds
Data Source Views
Adventure Works2012.dsv
Cubes
Adventure Works2012.cube
Dimensions
Person.dim
Employee.dim
Mining Structures
Roles
Assemblies
Miscellaneous
Getting Started (SSIS)
An error has occurred while loading Getting Started information. The remote name could not be resolved: 'go.microsoft.com'
Microsoft Visual Studio
The following objects will also be saved :
Cube [Adventure Works2012]
Would you like to proceed?
OK
Cancel
8:43 PM
3/25/2019

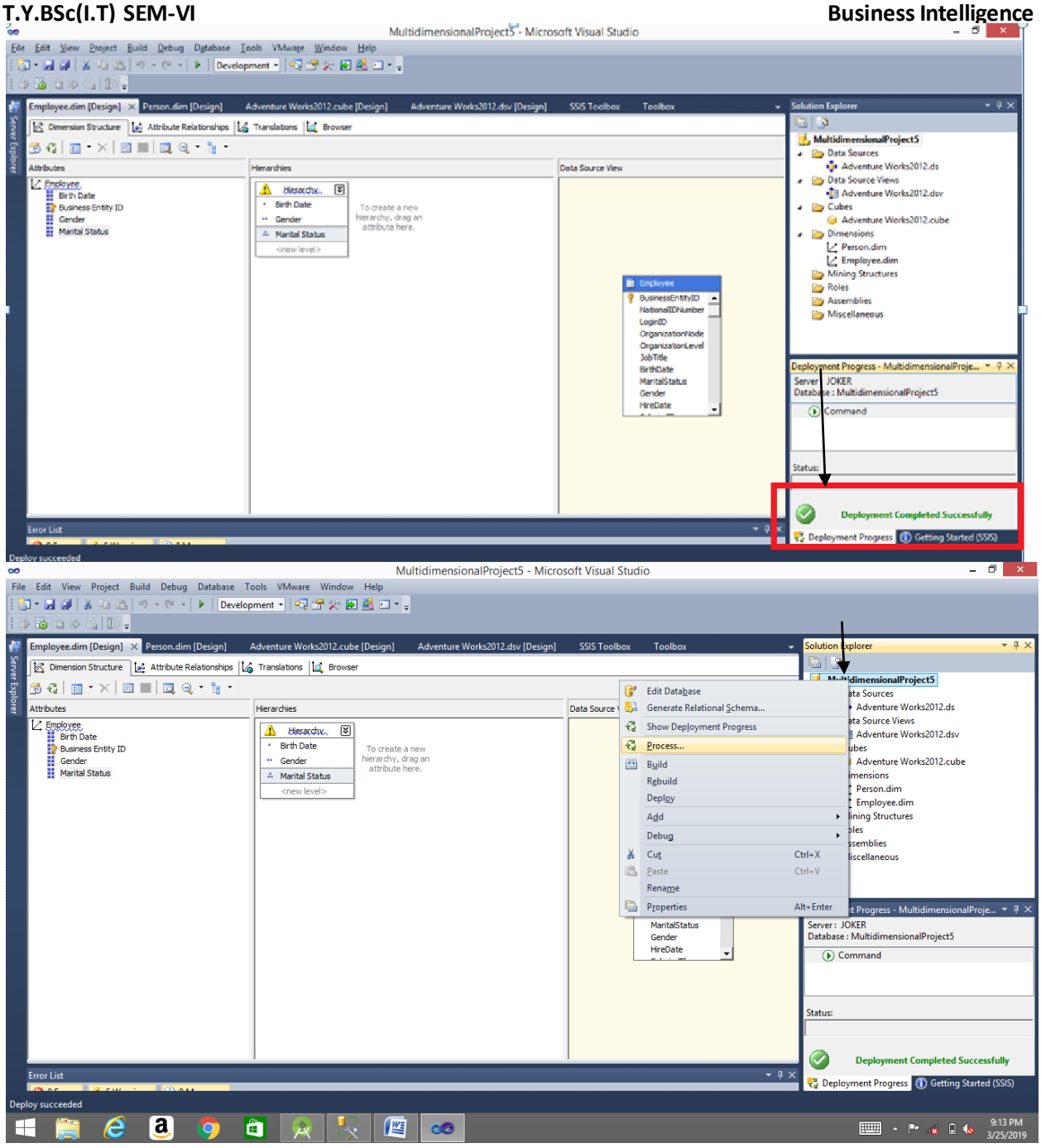

MultidimensionalProject4 - Microsoft Visual Studio
File Edit View Project Build Debug Database Dimension Tools VMware Window Help
Development
Employee.dim [Design]* Person.dim [Design] Adventure Works2012.cube [Design]* Adventure Works2012.dsv [Design] SSIS Toolbox Toolbox
Dimension Structure Attribute Relationships Translations Browser
Attributes
Employee
Birth Date
Business Entity ID
Gender
Hire Date
Marital Status
Hierarchies
To create a new hierarchy, drag an attribute here.
Data Source View
Employee
BusinessEntityID
NationalIDNumber
LoginID
OrganizationNode
OrganizationLevel
JobTitle
BirthDate
MaritalStatus
Gender
HireDate
Solution Explorer
MultidimensionalProject4
Data Sources
Adventure Works2012.ds
Data Source Views
Adventure Works2012.dsv
Cubes
Adventure Works2012.cube
Dimensions
Person.dim
Employee.dim
Mining Structures
Roles
Assemblies
Miscellaneous
Getting Started (SSIS)
An error has occurred while loading Getting Started information. The remote name could not be resolved: 'go.microsoft.com'
Item(s) Saved
8:43 PM
3/25/2019

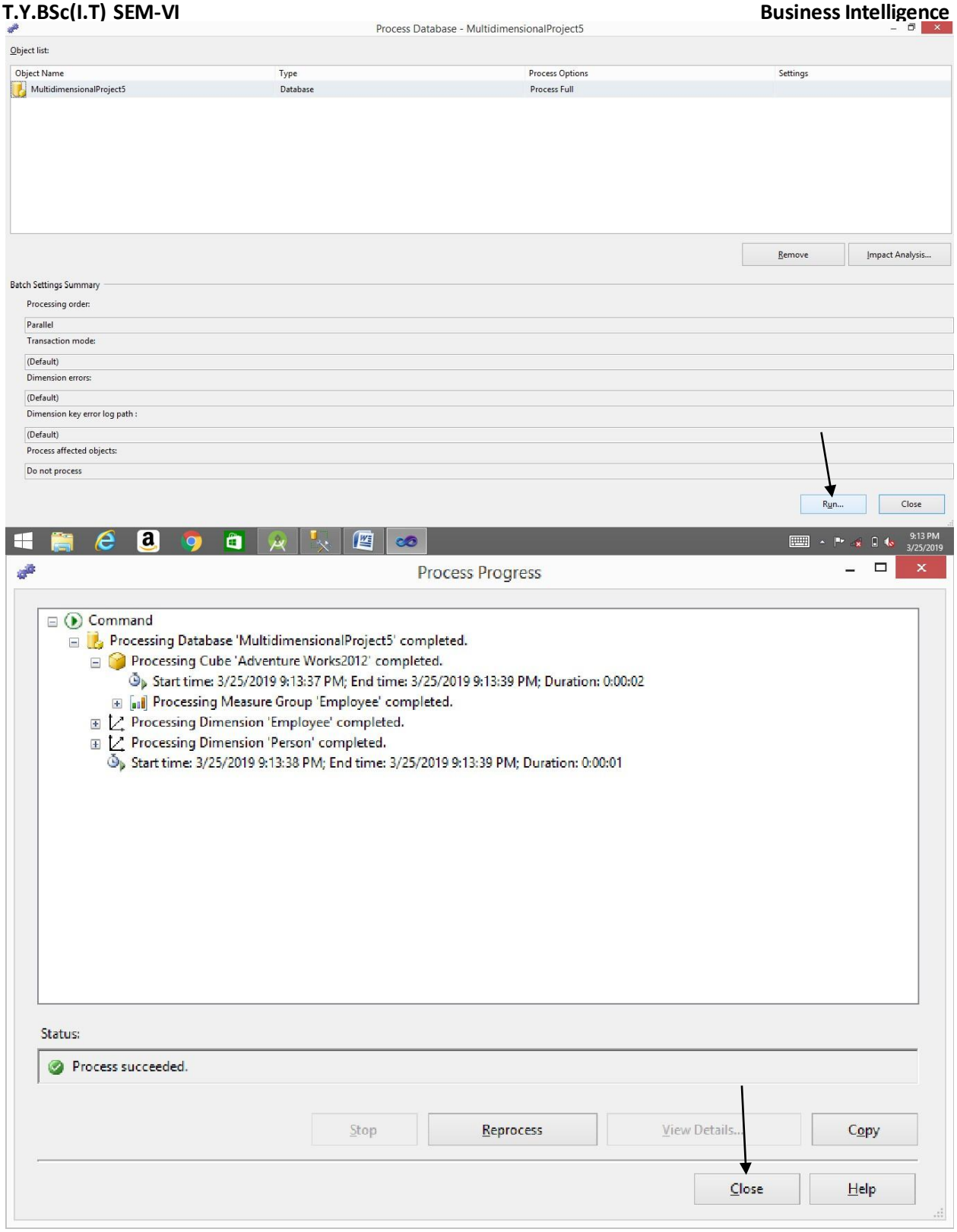

MultidimensionalProject4 - Microsoft Visual Studio
File Edit View Project Build Debug Format Database Data Source View Dimension Tools VMware Window Help
Development
Employee.dim [Design]* Person.dim [Design] Adventure Works2012.cube [Design]* Adventure Works2012.dsv [Design] SSIS Toolbox Toolbox
Dimension Structure Attribute Relationships Translations Browser
Attributes
Hierarchies
Data Source View
Employee
Birth Date
Business Entity ID
Gender
Hire Date
Marital Status
Hierarchy
<new level>
To create a new hierarchy, drag an attribute here.
BusinessEntityID
NationalIDNumber
LoginID
OrganizationNode
OrganizationLevel
JobTitle
BirthDate
MaritalStatus
Gender
HireDate
Solution Explorer
MultidimensionalProject4
Data Sources
Adventure Works2012.ds
Data Source Views
Adventure Works2012.dsv
Cubes
Adventure Works2012.cube
Dimensions
Person.dim
Employee.dim
Mining Structures
Roles
Assemblies
Miscellaneous
Getting Started (SSIS)
An error has occurred while loading Getting Started information. The remote name could not be resolved: 'go.microsoft.com'
Microsoft Visual Studio
The following objects will also be saved :
Cube [Adventure Works2012]
Would you like to proceed?
OK
Cancel
8:44 PM
3/25/2019

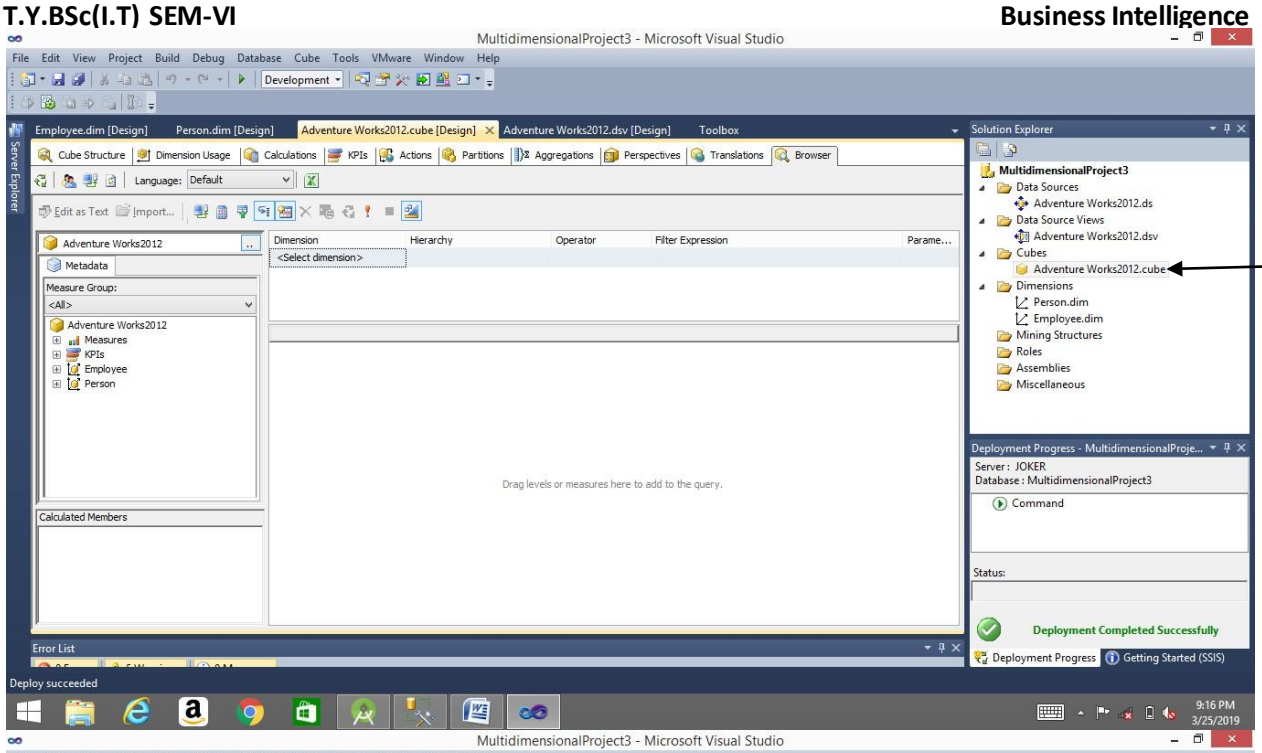

MultidimensionalProject4 - Microsoft Visual Studio
File Edit View Project Build Debug Database Tools VMware Window Help
Development
Employee.dim [Design] Person.dim [Design] Adventure Works2012.cube [Design] Adventure Works2012.dsv [Design] SSIS Toolbox Toolbox
Dimension Structure Attribute Relationships Translations Browser
Attributes
Hierarchies
Employee
Birth Date
Business Entity ID
Gender
Hire Date
Marital Status
Hierarchy
<new level>
To create a new hierarchy, drag an attribute here.
Edit Database
Generate Relational Schema...
Show Deployment Progress
Process...
Build
Rebuild
Deploy
Add
Debug
Cut Ctrl+X
Paste Ctrl+V
Rename
Properties Alt+Enter
Solution Explorer
MultidimensionalProject4
Data Sources
Adventure Works2012.ds
Data Source Views
Adventure Works2012.dsv
Cubes
Adventure Works2012.cube
Dimensions
Person.dim
Employee.dim
Mining Structures
Roles
Assemblies
Miscellaneous
An error has occurred while loading Getting Started information. The remote name could not be resolved: 'go.microsoft.com'
Item(s) Saved
8:45 PM
3/25/2019

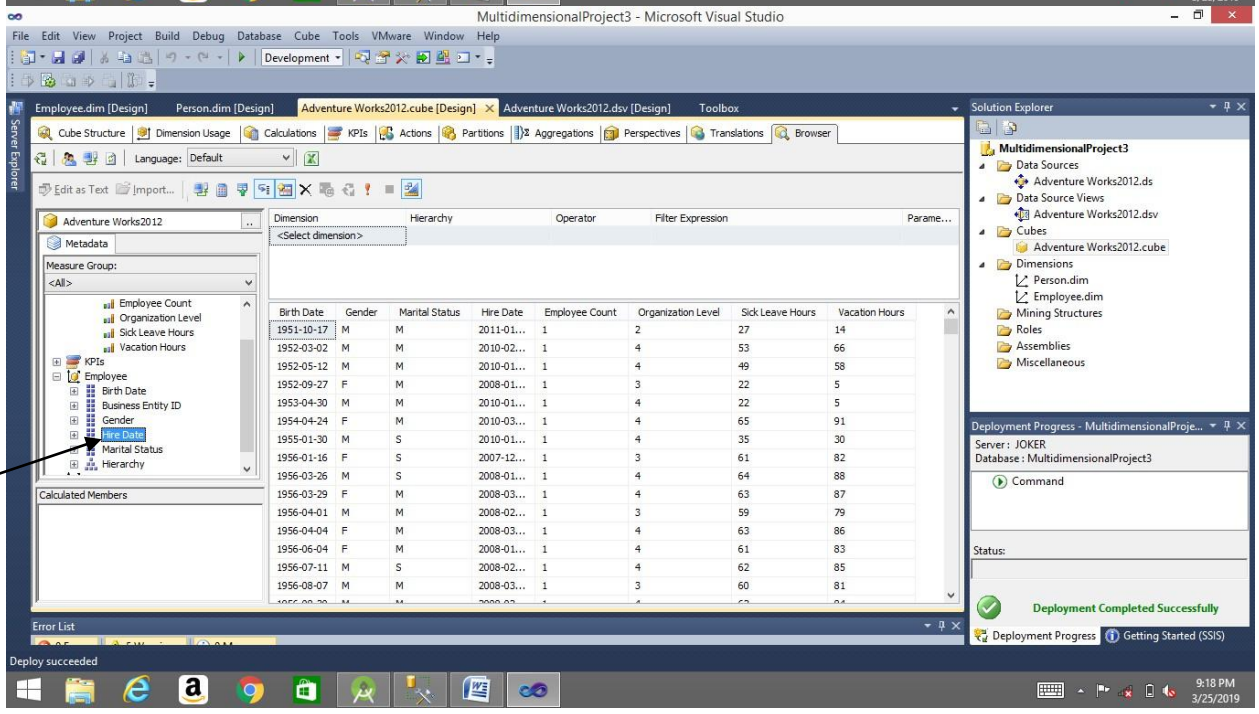

MultidimensionalProject4 Property Pages
Configuration: Active(Development)
Platform: N/A
Configuration Manager...
Configuration Properties
Build
Debugging
Deployment
Options
Processing Option
Do Not Process
Transactional Deployment
False
Server Mode
Deploy All
Target
Server
JOKER
Database
MultidimensionalProject4
Server
The Analysis Services instance to which the project will be deployed.
OK
Cancel
Apply


MultidimensionalProject4 - Microsoft Visual Studio
Edit Database
Generate Relational Schema...
Show Deployment Progress
Process...
Build
Rebuild
Deploy
Add
Debug
Cut
Paste
Rename
Properties
Person
Employee
Solution Explorer
An error has occurred while loading Getting Started information. The remote name could not be resolved: 'go.microsoft.com'
Item(s) Saved


MultidimensionalProject5 - Microsoft Visual Studio
Deployment Completed Successfully
Deployment Progress
Getting Started (SSIS)
Deploy succeeded
Edit Database
Generate Relational Schema...
Show Deployment Progress
Process...
Build
Rebuild
Deploy
Add
Debug
Cut Ctrl+X
Paste Ctrl+V
Rename
Properties Alt+Enter
Server : JOKER
Database : MultidimensionalProject5
Command
Status:
9:13 PM
3/25/2019


Process Database - MultidimensionalProject5
Object list:
Object Name
Type
Process Options
Settings
MultidimensionalProject5
Database
Process Full
Remove
Impact Analysis...
Batch Settings Summary
Processing order:
Parallel
Transaction mode:
(Default)
Dimension errors:
(Default)
Dimension key error log path :
(Default)
Process affected objects:
Do not process
Run...
Close
9:13 PM
3/25/2019
Process Progress
Command
Processing Database 'MultidimensionalProject5' completed.
Processing Cube 'Adventure Works2012' completed.
Start time: 3/25/2019 9:13:37 PM; End time: 3/25/2019 9:13:39 PM; Duration: 0:00:02
Processing Measure Group 'Employee' completed.
Processing Dimension 'Employee' completed.
Processing Dimension 'Person' completed.
Start time: 3/25/2019 9:13:38 PM; End time: 3/25/2019 9:13:39 PM; Duration: 0:00:01
Status:
Process succeeded.
Stop
Reprocess
View Details...
Copy
Close
Help


MultidimensionalProject3 - Microsoft Visual Studio
File Edit View Project Build Debug Database Cube Tools VMware Window Help
Development
Employee.dim [Design]
Person.dim [Design]
Adventure Works2012.cube [Design]
Adventure Works2012.dsv [Design]
Toolbox
Cube Structure
Dimension Usage
Calculations
KPIs
Actions
Partitions
Aggregations
Perspectives
Translations
Browser
Language: Default
Edit as Text
Import...
Adventure Works2012
Metadata
Measure Group:
<All>
Measures
KPIs
Employee
Person
Dimension
Hierarchy
Operator
Filter Expression
Parame...
<Select dimension>
Drag levels or measures here to add to the query.
Calculated Members
Solution Explorer
MultidimensionalProject3
Data Sources
Adventure Works2012.ds
Data Source Views
Adventure Works2012.dsv
Cubes
Adventure Works2012.cube
Dimensions
Person.dim
Employee.dim
Mining Structures
Roles
Assemblies
Miscellaneous
Deployment Progress - MultidimensionalProje...
Server : JOKER
Database : MultidimensionalProject3
Command
Status:
Deployment Completed Successfully
Deployment Progress
Getting Started (SSIS)
Error List
Deploy succeeded
9:16 PM
3/25/2019


MultidimensionalProject3 - Microsoft Visual Studio
File Edit View Project Build Debug Database Cube Tools VMware Window Help
Development
Employee.dim [Design]
Person.dim [Design]
Adventure Works2012.cube [Design]
Adventure Works2012.dsv [Design]
Toolbox
Cube Structure
Dimension Usage
Calculations
KPIs
Actions
Partitions
Aggregations
Perspectives
Translations
Browser
Language: Default
Edit as Text
Import...
Adventure Works2012
Metadata
Measure Group:
<All>
Employee Count
Organization Level
Sick Leave Hours
Vacation Hours
KPIs
Employee
Birth Date
Business Entity ID
Gender
Hire Date
Marital Status
Hierarchy
Dimension
Hierarchy
Operator
Filter Expression
Parame...
<Select dimension>
Birth Date Gender Marital Status Hire Date Employee Count Organization Level Sick Leave Hours Vacation Hours
1951-10-17 M M 2011-01... 1 2 27 14
1952-03-02 M M 2010-02... 1 4 53 66
1952-05-12 M M 2010-01... 1 4 49 58
1952-09-27 F M 2008-01... 1 3 22 5
1953-04-30 M M 2010-01... 1 4 22 5
1954-04-24 F M 2010-03... 1 4 65 91
1955-01-30 M S 2010-01... 1 4 35 30
1956-01-16 F S 2007-12... 1 3 61 82
1956-03-26 M S 2008-01... 1 4 64 88
1956-03-29 F M 2008-03... 1 4 63 87
1956-04-01 M M 2008-02... 1 3 59 79
1956-04-04 F M 2008-03... 1 4 63 86
1956-06-04 F M 2008-01... 1 4 61 83
1956-07-11 M S 2008-02... 1 4 62 85
1956-08-07 M M 2008-03... 1 3 60 81
Calculated Members
Solution Explorer
MultidimensionalProject3
Data Sources
Adventure Works2012.ds
Data Source Views
Adventure Works2012.dsv
Cubes
Adventure Works2012.cube
Dimensions
Person.dim
Employee.dim
Mining Structures
Roles
Assemblies
Miscellaneous
Deployment Progress - MultidimensionalProje...
Server : JOKER
Database : MultidimensionalProject3
Command
Status:
Deployment Completed Successfully
Deployment Progress
Getting Started (SSIS)
Error List
Deploy succeeded
9:18 PM
3/25/2019