Practical 1

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

Solution:

In Power BI:

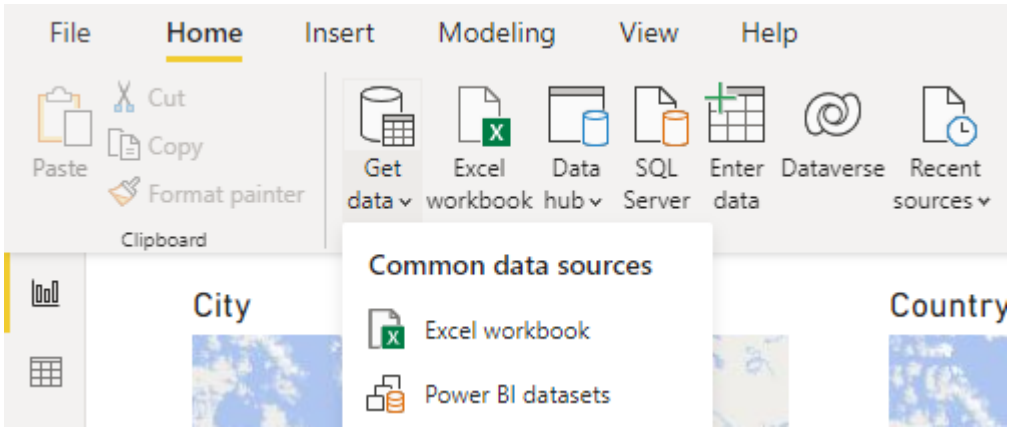
Steps:

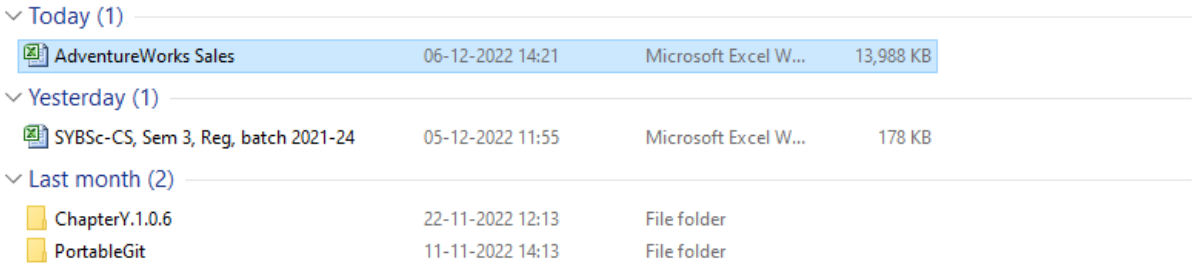
https://github.com/microsoft/powerbi-desktop-samples/blob/main/AdventureWorks%20Sales%20Sample/AdventureWorks%20Sales.xlsx

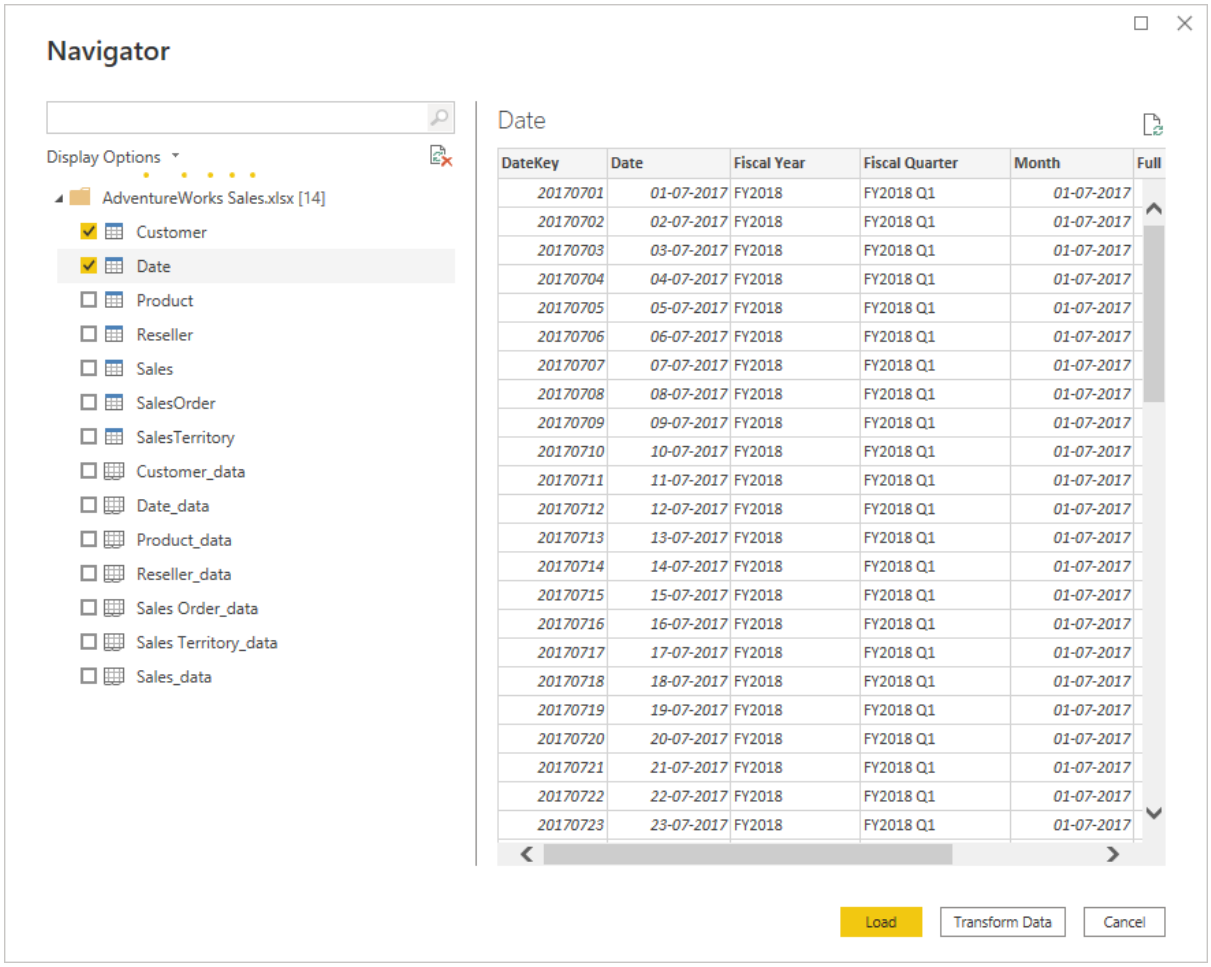
Download excel sheet

Import excel sheet in Power BI

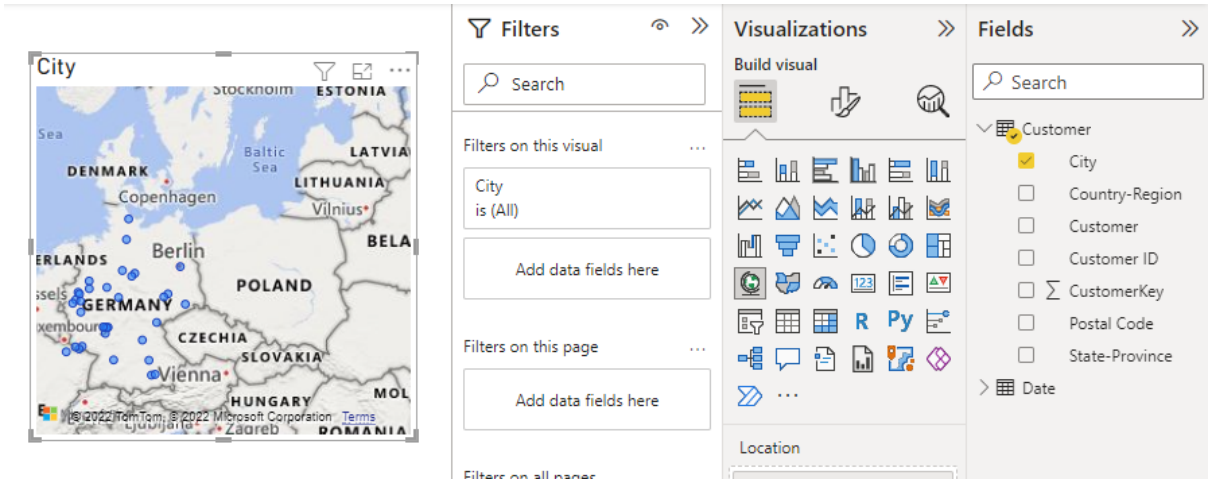
Select Get Data from home tab and select Excel workbook.

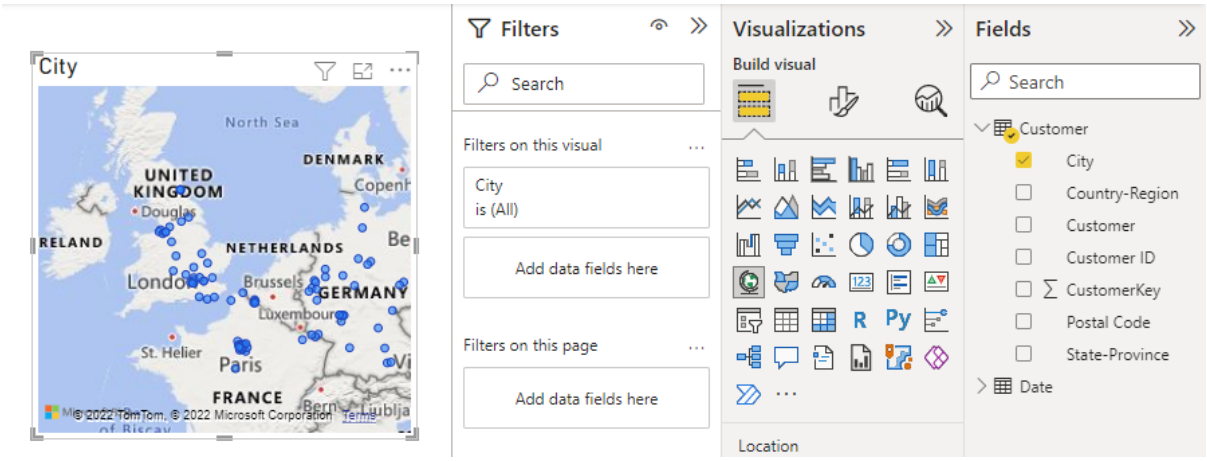


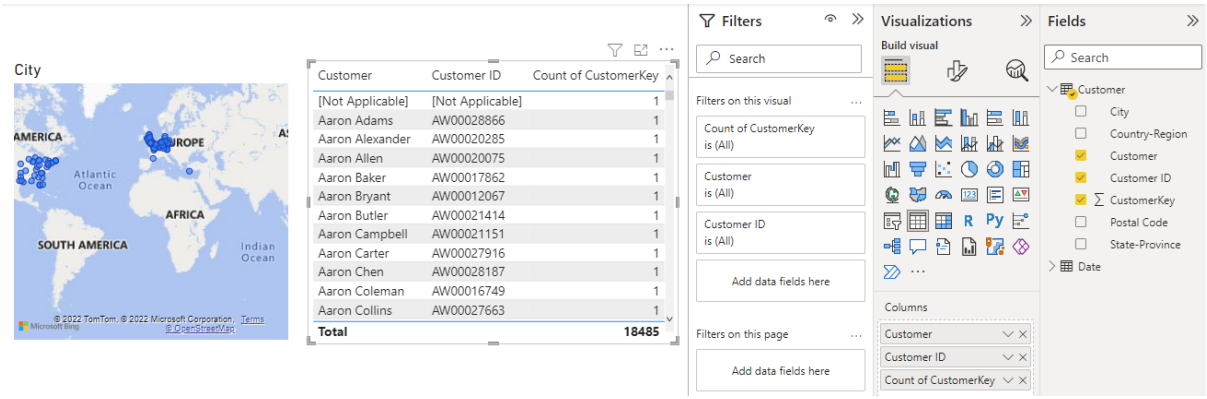




Now select the fields which you want to see.







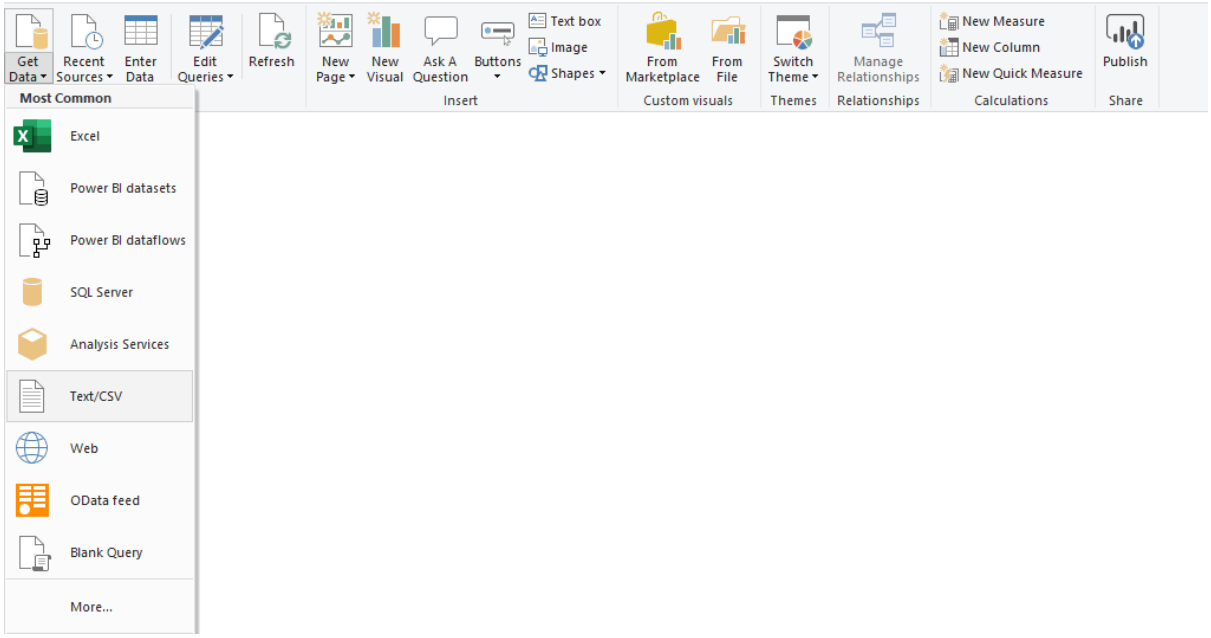
Importing through FlatFile/TextFile.

sampleDataTextBI - Notepad

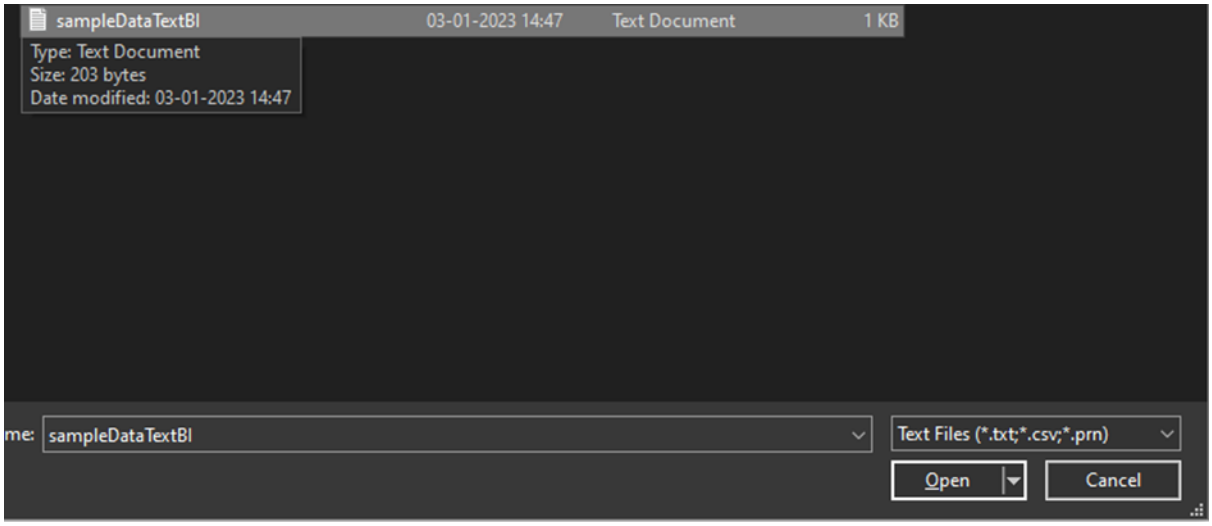
File Edit Format View Help

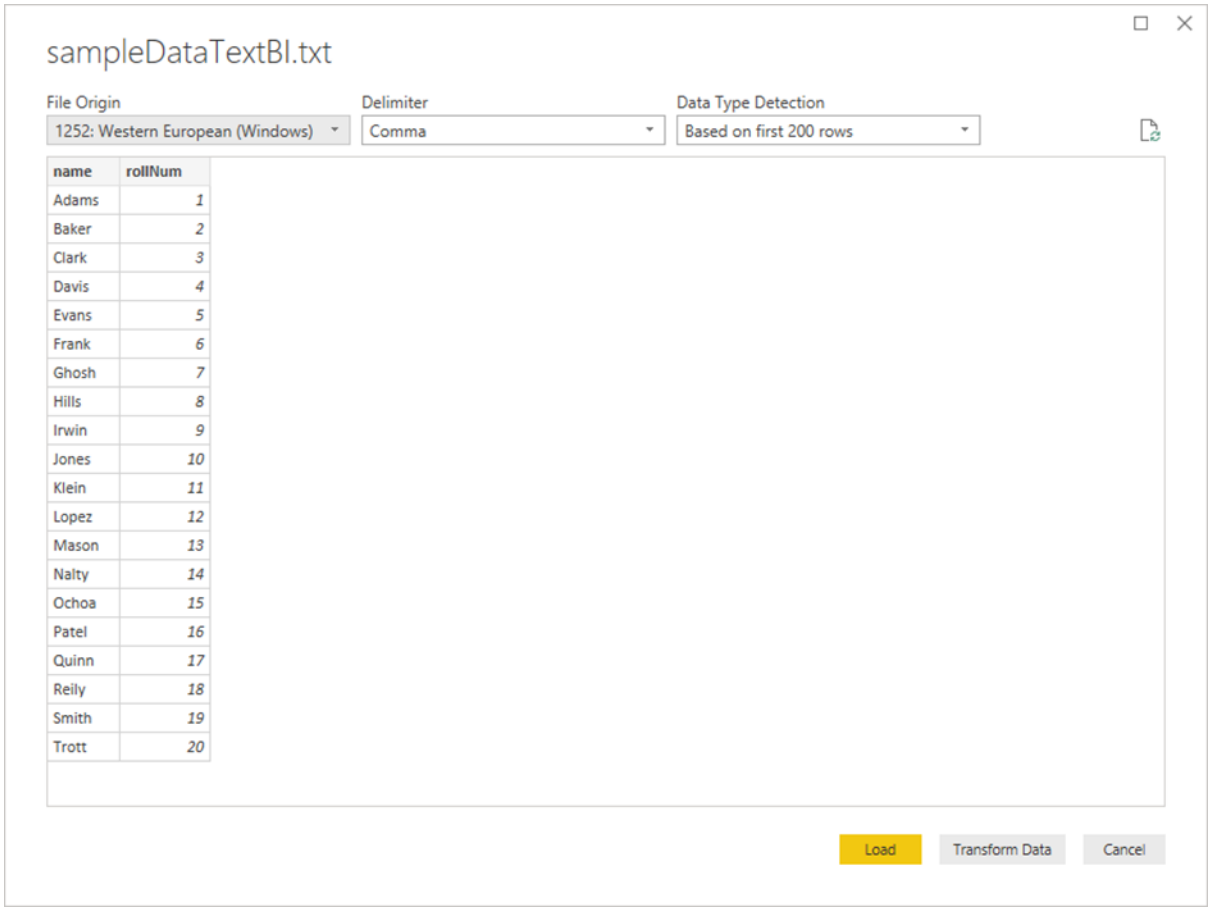
```
name,rollNum
Adams,1
Baker,2
Clark,3
Davis,4
Evans,5
Frank,6
Ghosh,7
Hills,8
Irwin,9
Jones,10
Klein,11
Lopez,12
Mason,13
Nalty,14
Ochoa,15
Patel,16
Quinn,17
Reily,18
Smith,19
Trott,20
```

Open Power BI and select Text/CSV option from Get Data present on Home tab.

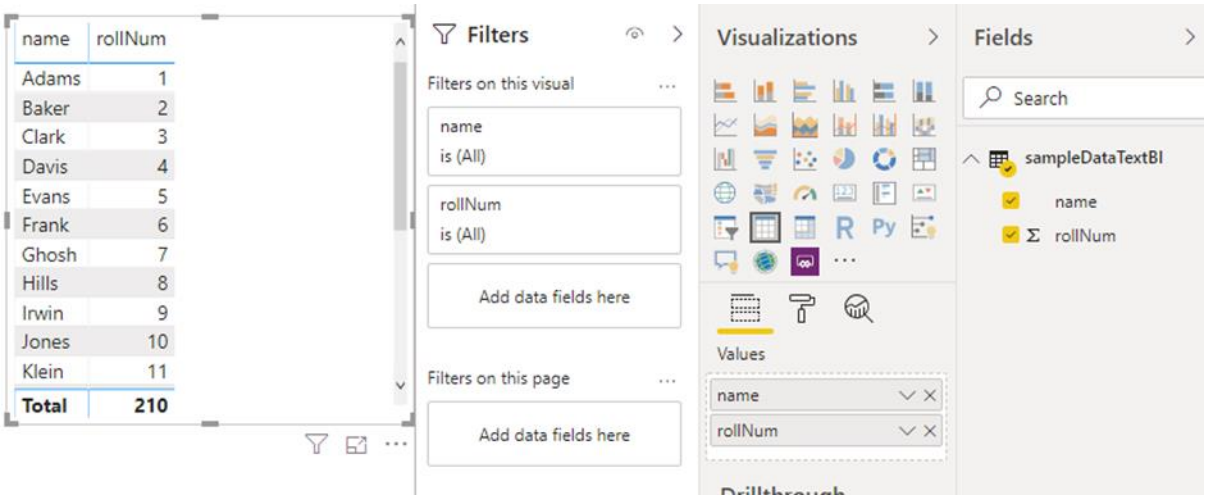


Select the saved TextFile





Select the fields.

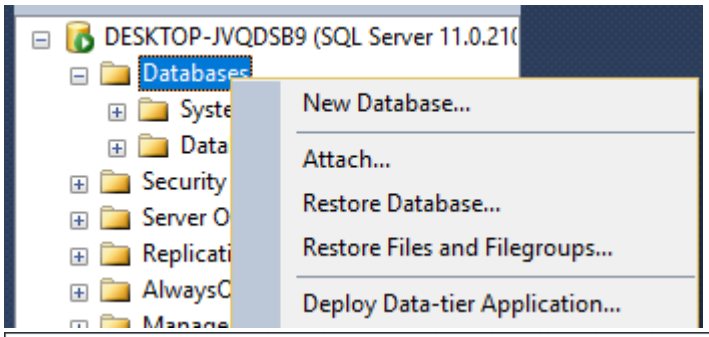


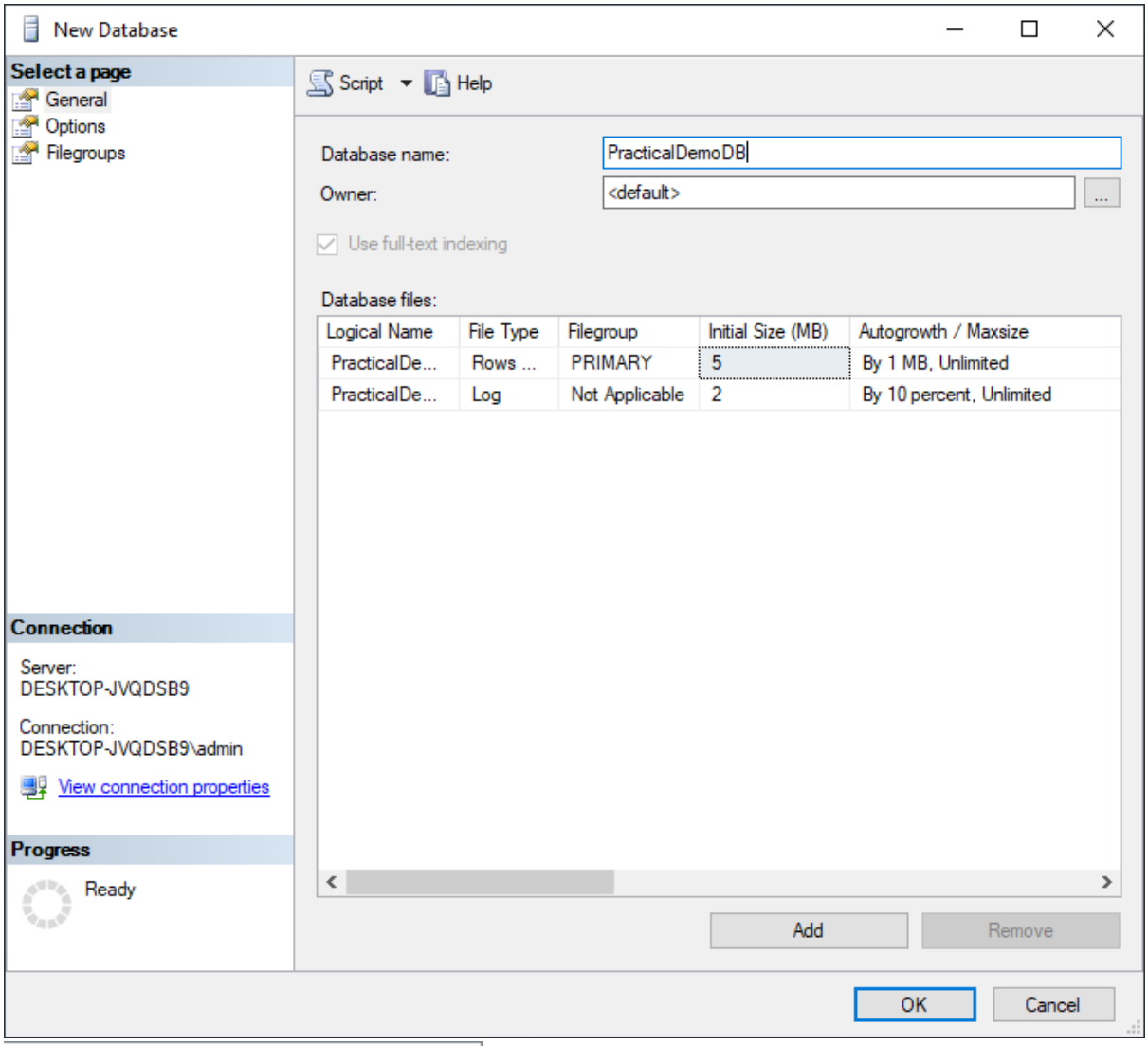
In SQL Server Management Studio and SQL Server Data Tools

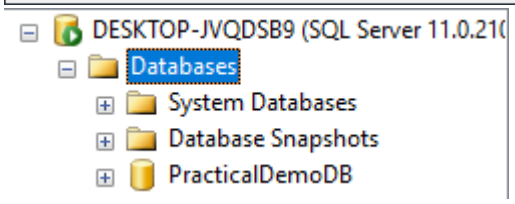
Importing data from Database

Steps:

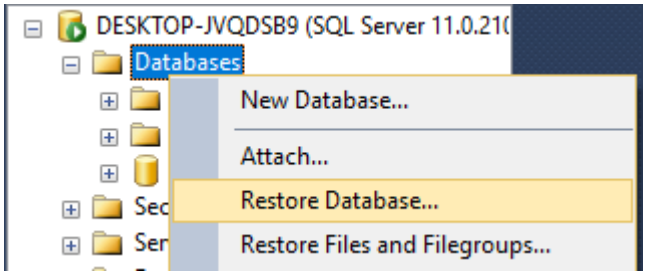
Launch the SQL server management studio and Create new database



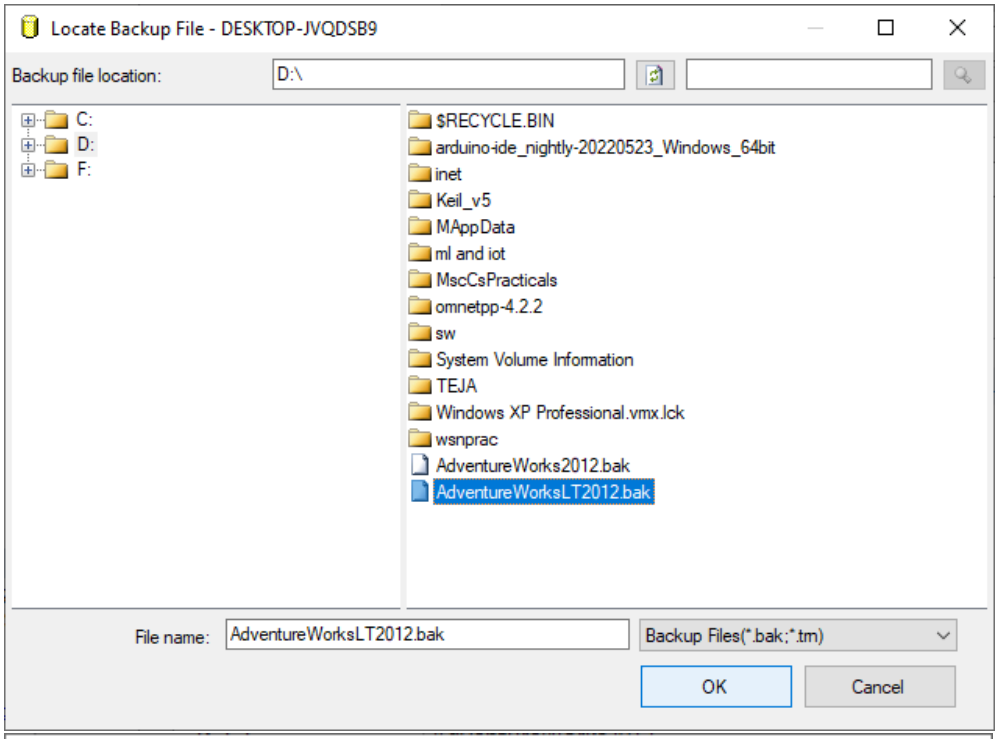


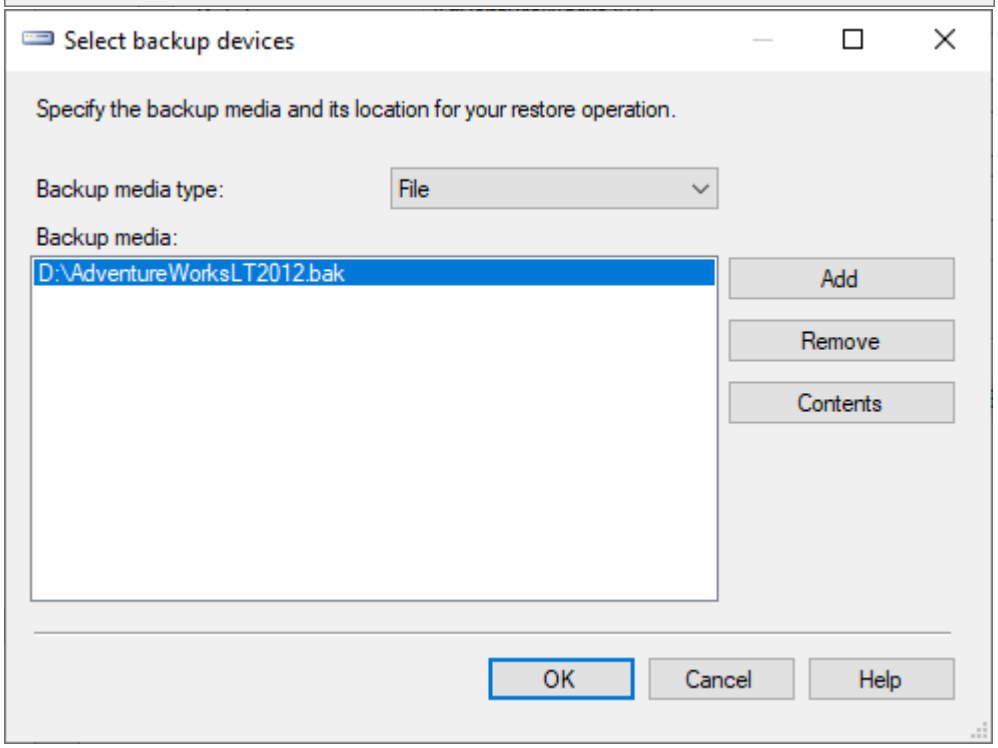


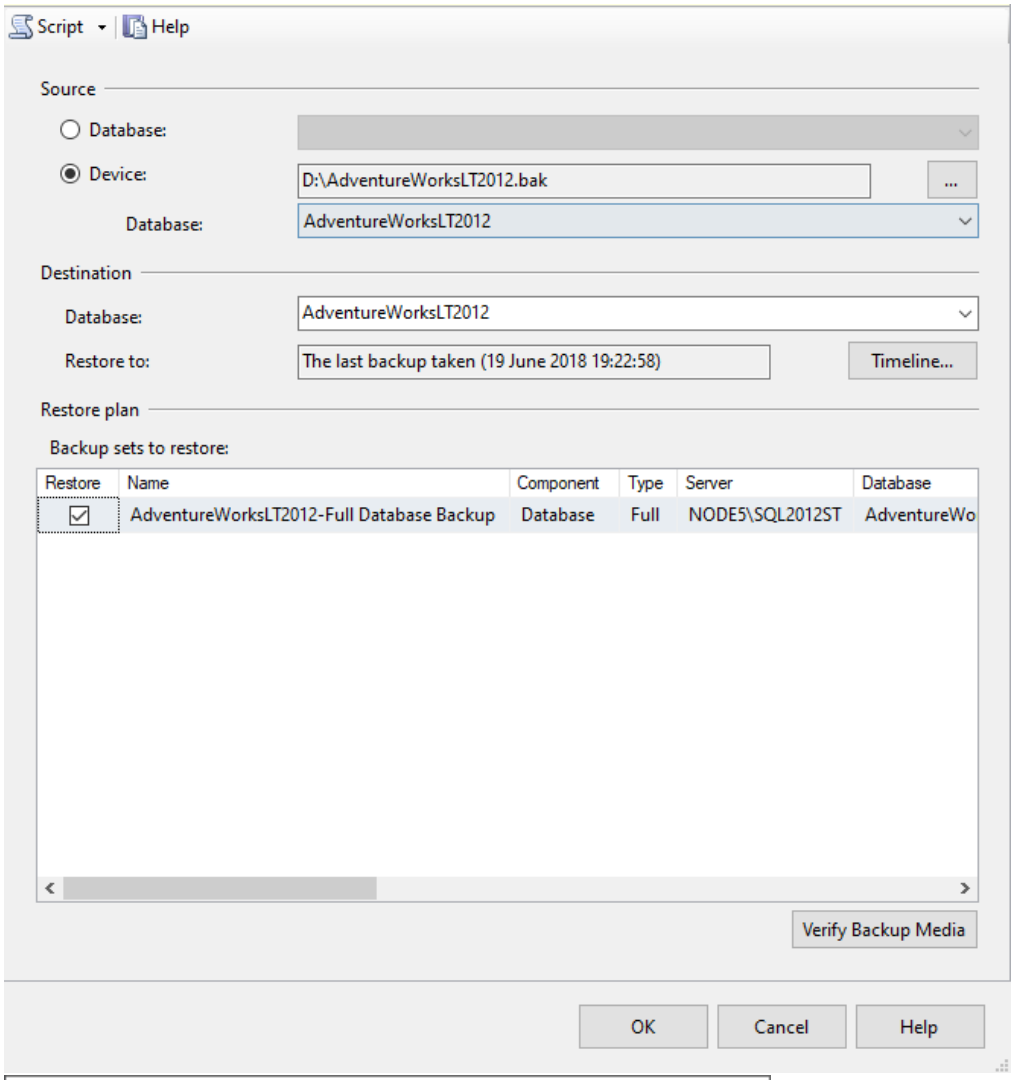
Now restore database

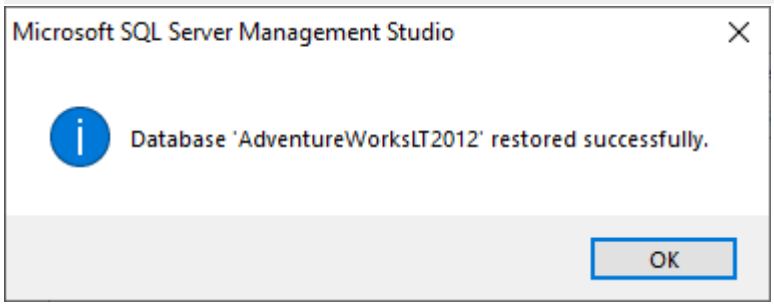


Select the downloaded database file.



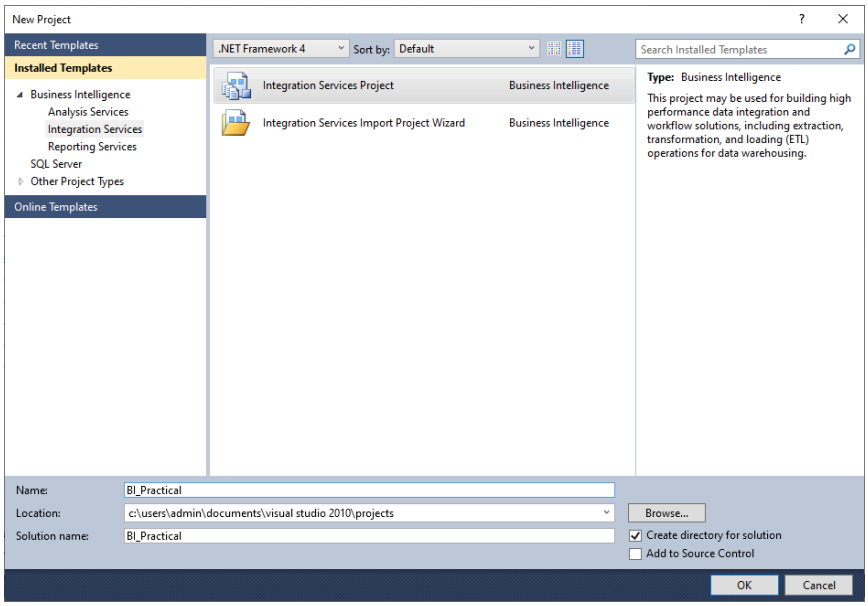



Script
Help
Source
Database:
Device:
D:\AdventureWorksLT2012.bak
...
Database:
AdventureWorksLT2012
Destination
Database:
AdventureWorksLT2012
Restore to:
The last backup taken (19 June 2018 19:22:58)
Timeline...
Restore plan
Backup sets to restore:
Restore
Name
Component
Type
Server
Database
AdventureWorksLT2012-Full Database Backup
Database
Full
NODE5\SQL2012ST
AdventureWo
Verify Backup Media
OK
Cancel
Help

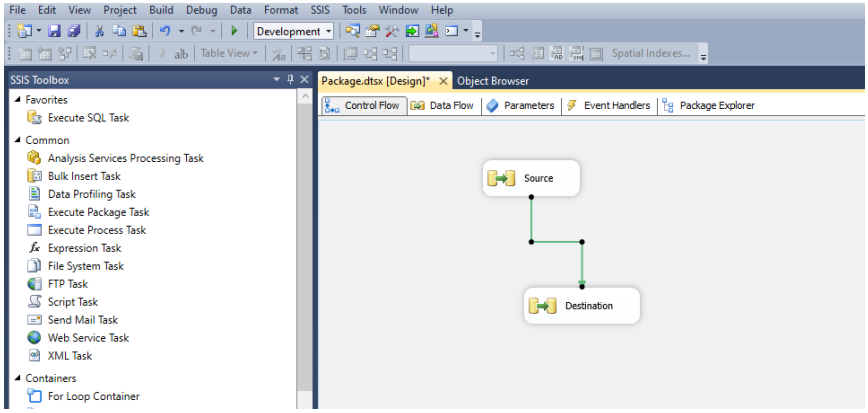

Microsoft SQL Server Management Studio
Database 'AdventureWorksLT2012' restored successfully.
OK

Now lauch SQL server data tools

Create new project



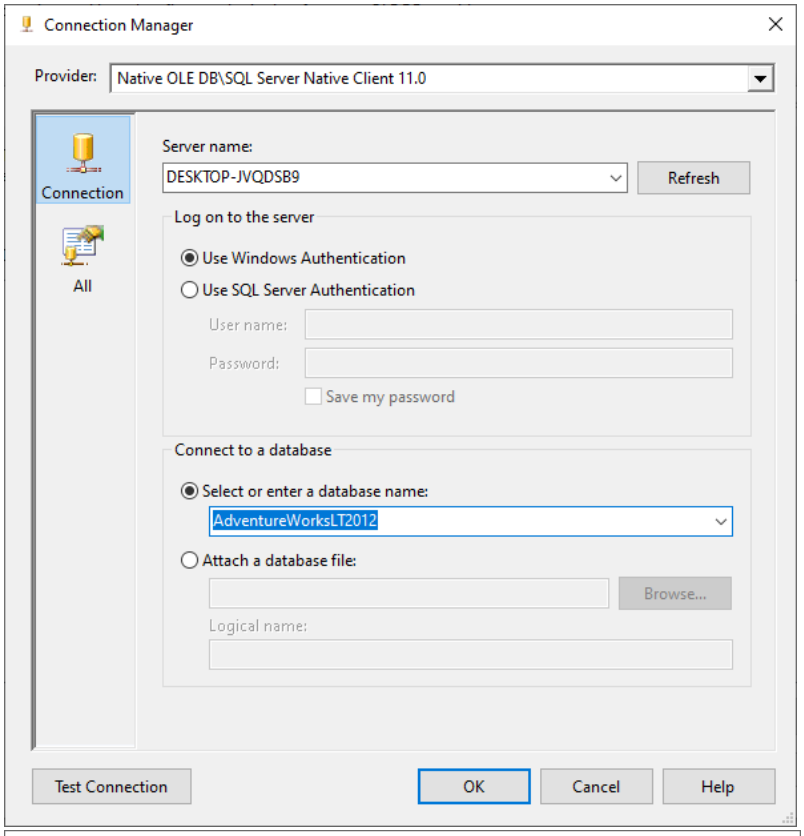
Now, place 2 Data Flow task

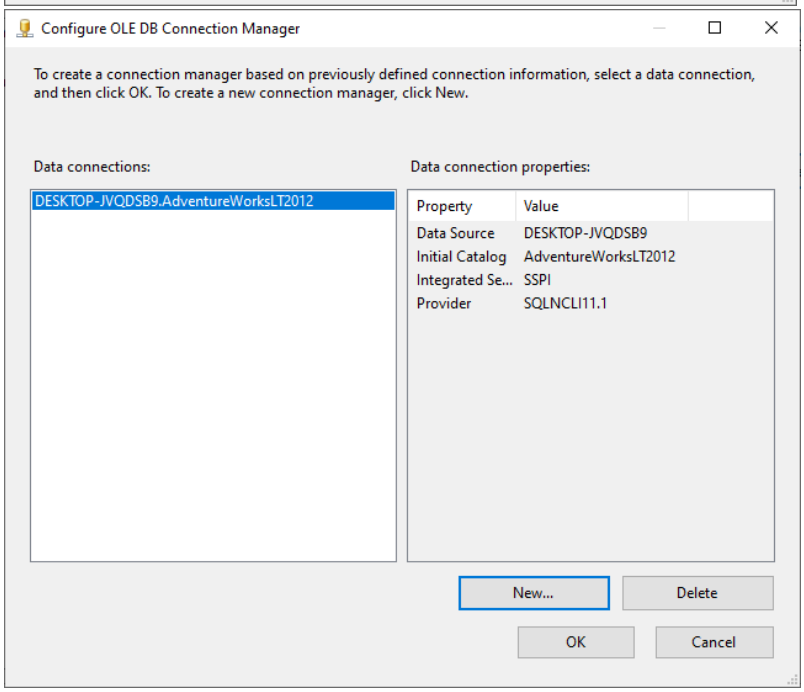


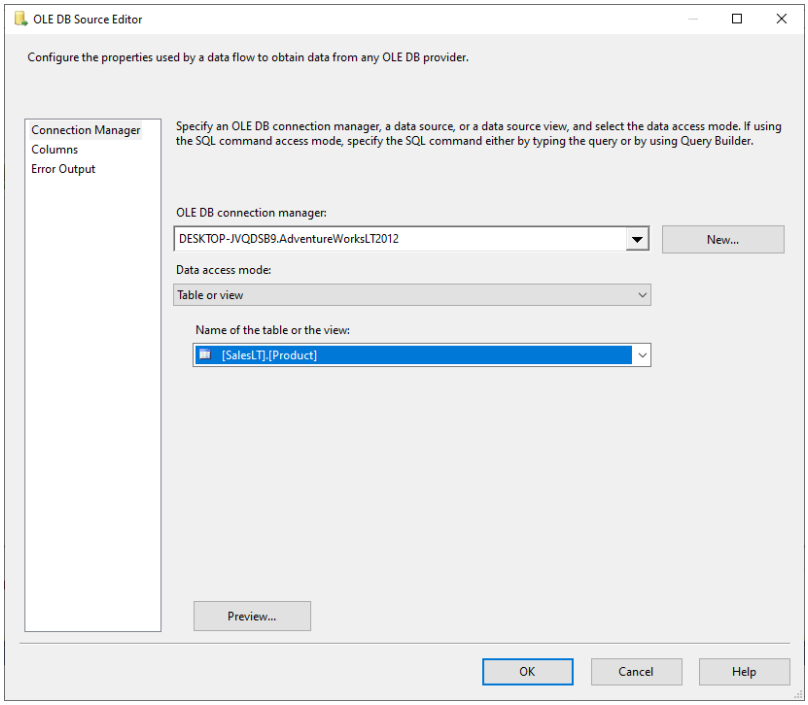
Go to source:

Add ole DB source

Double click on OLE DB source and add connection

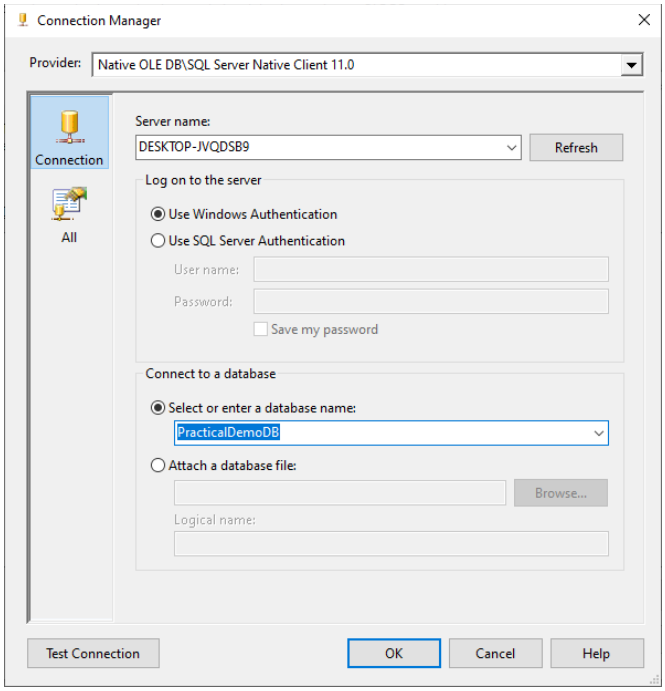


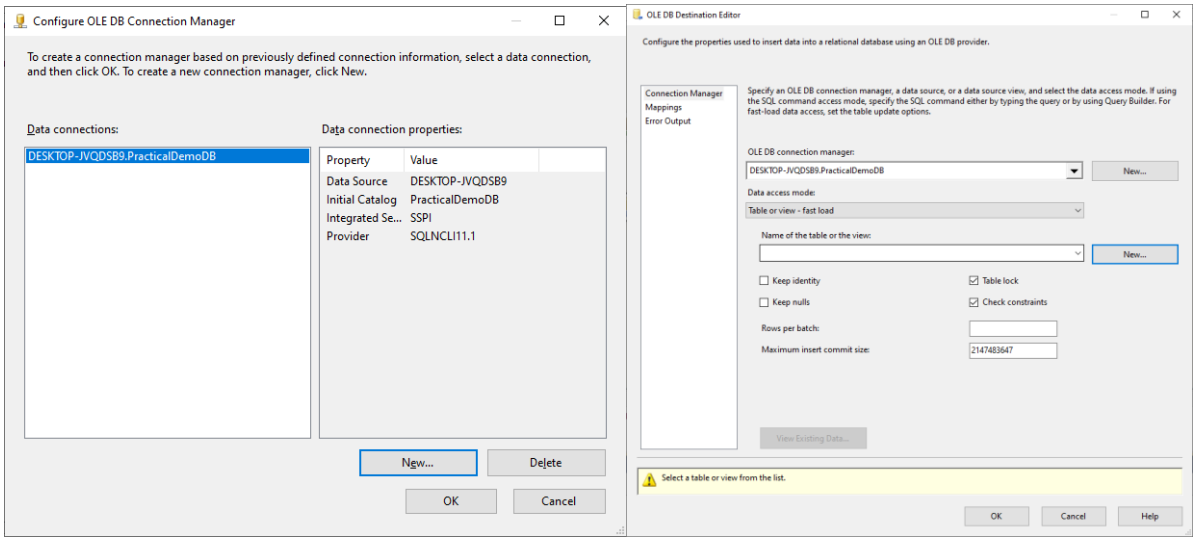




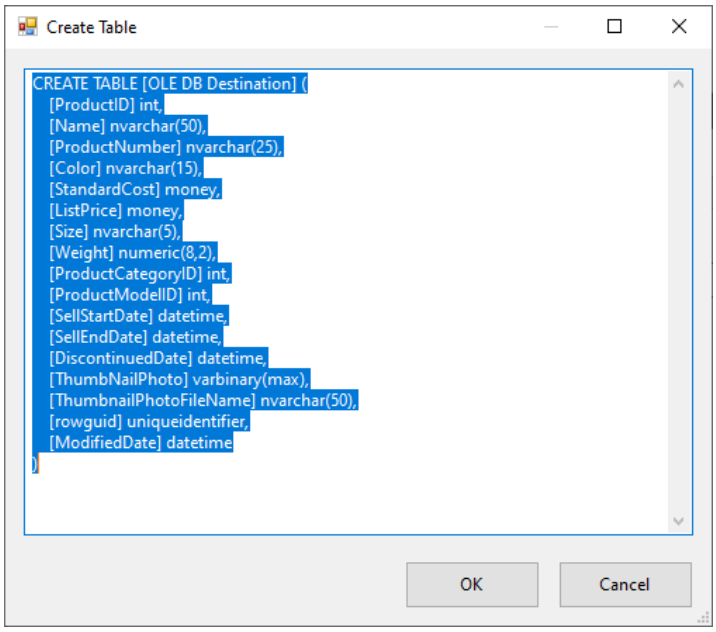
Now add one OLE DB destination

Then, add connection for created database (PracticalDemoDB)

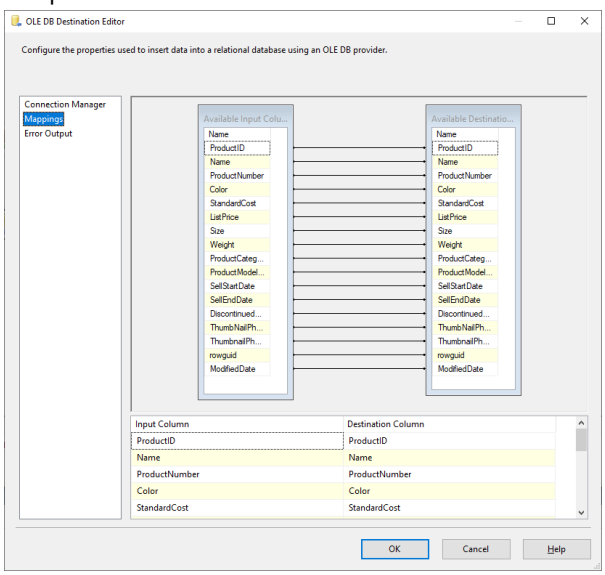


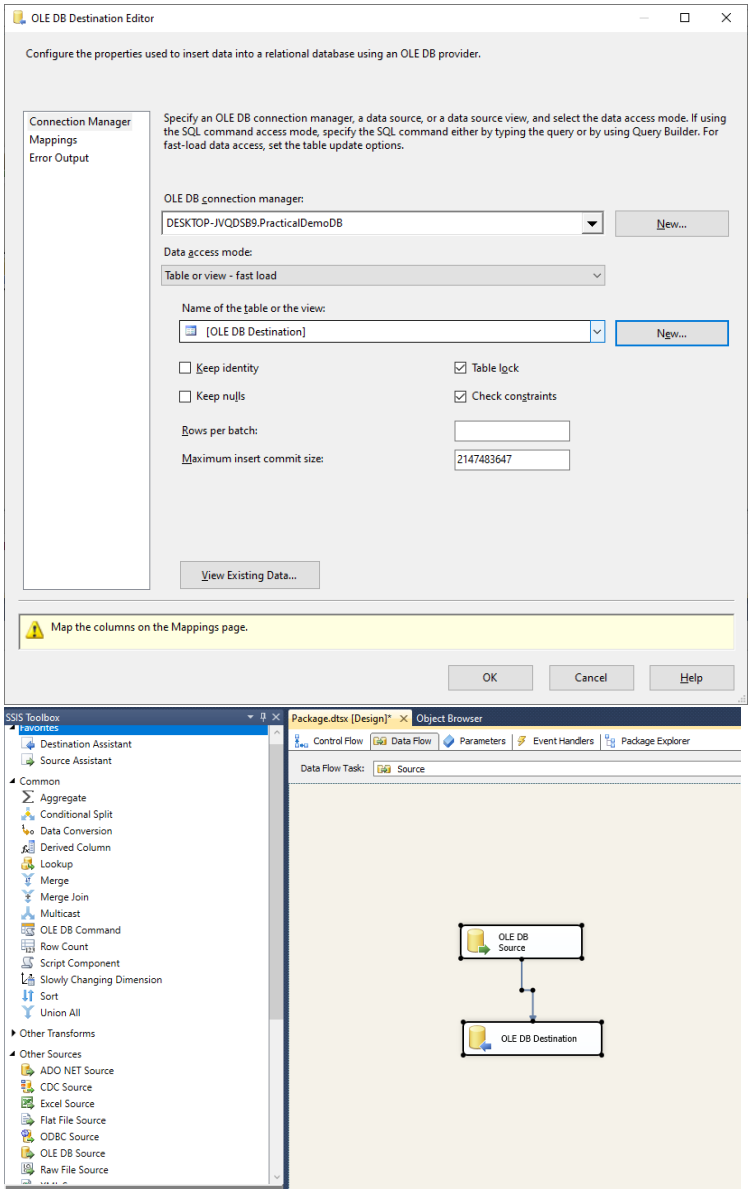


Click on “New...” to create table

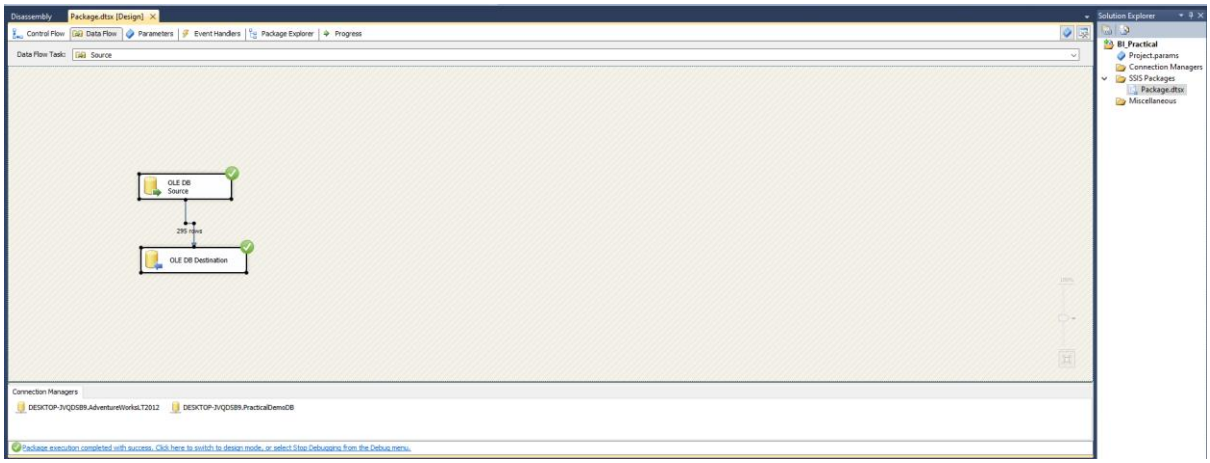


Map the columns

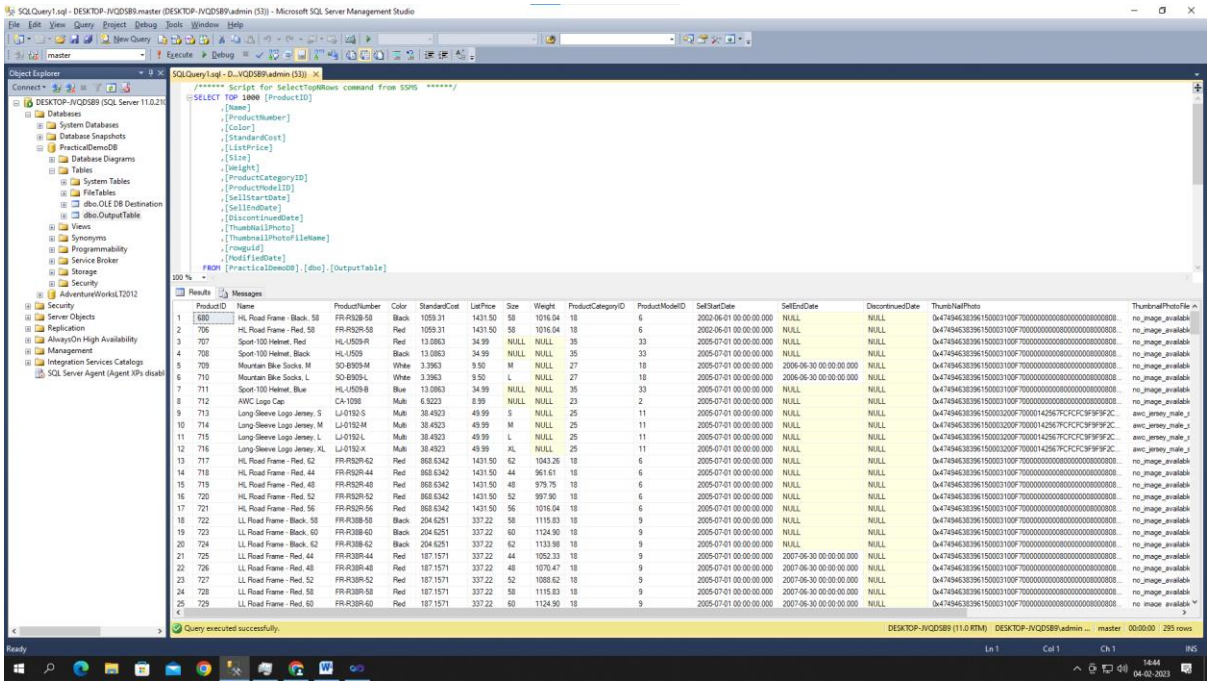




Then click debug,

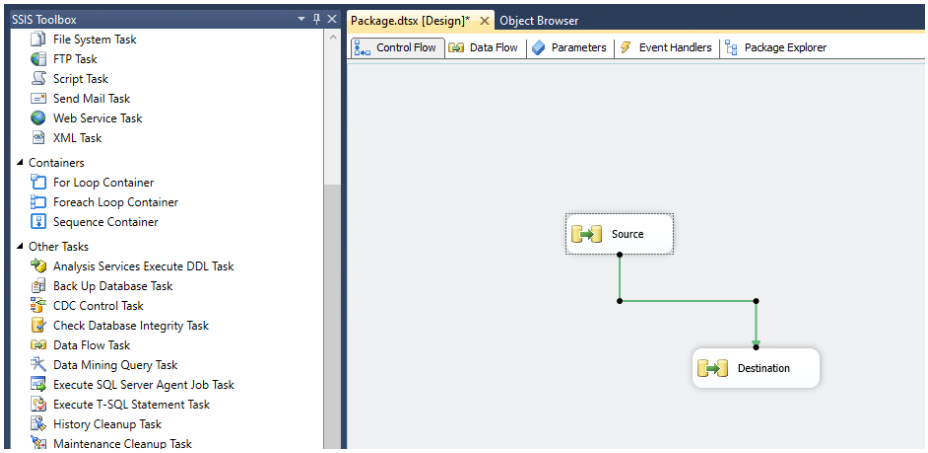


OUTPUT (Verifying it in SQL server management studio)

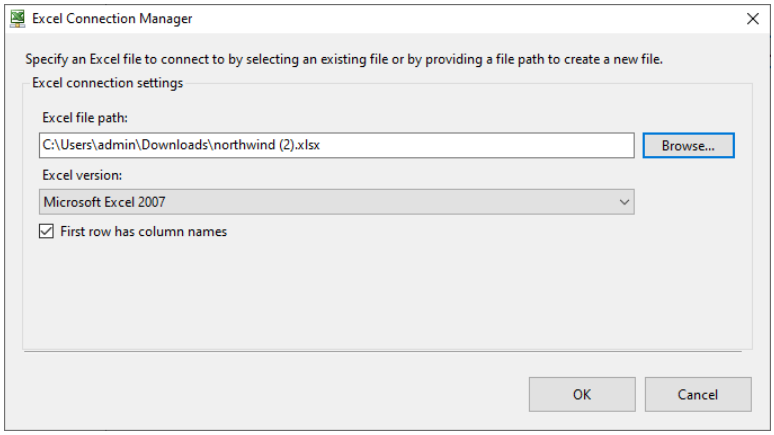


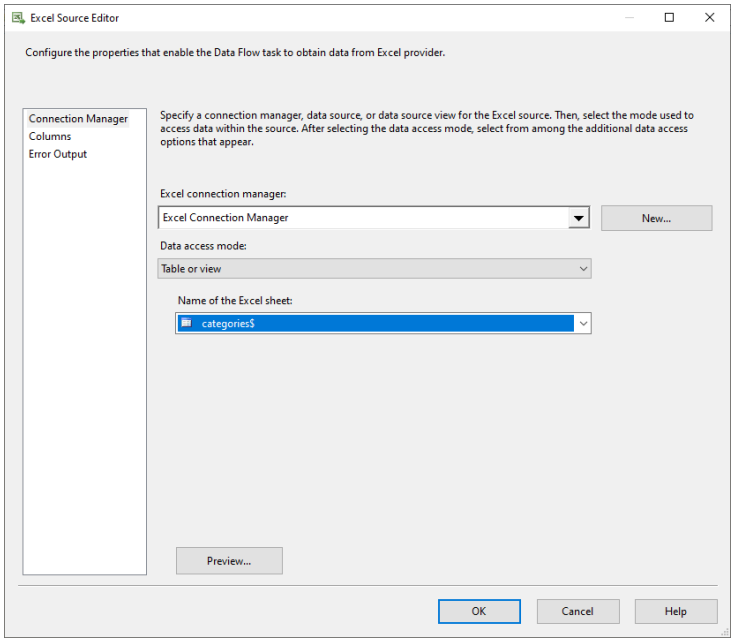
Importing Excel and loading it the system.

Place 2 Data Flow task.

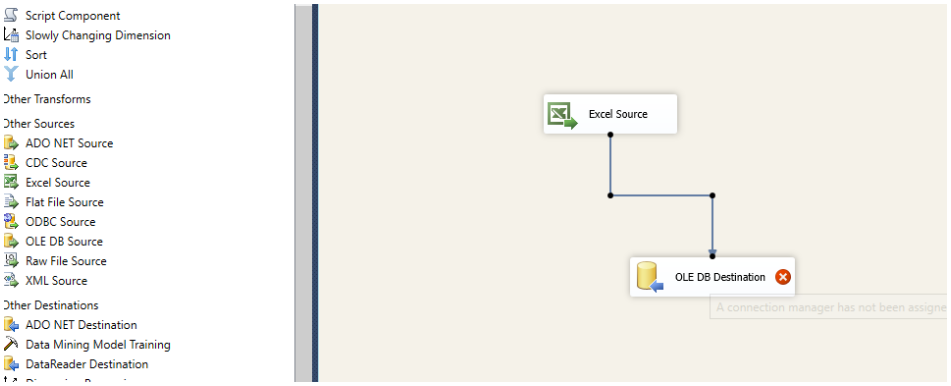


Open source and Place one Excel Source and choose your excel sheet

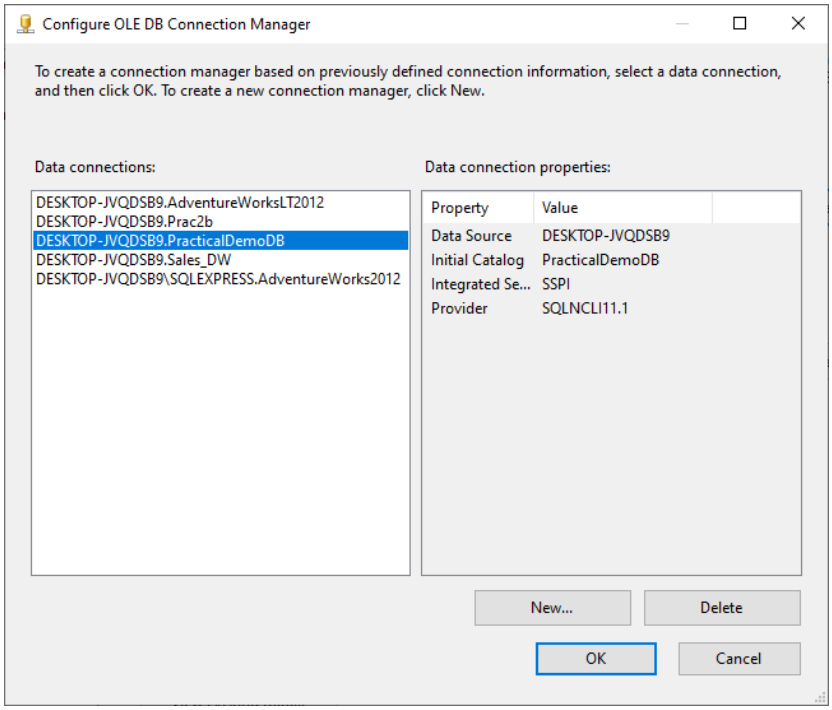


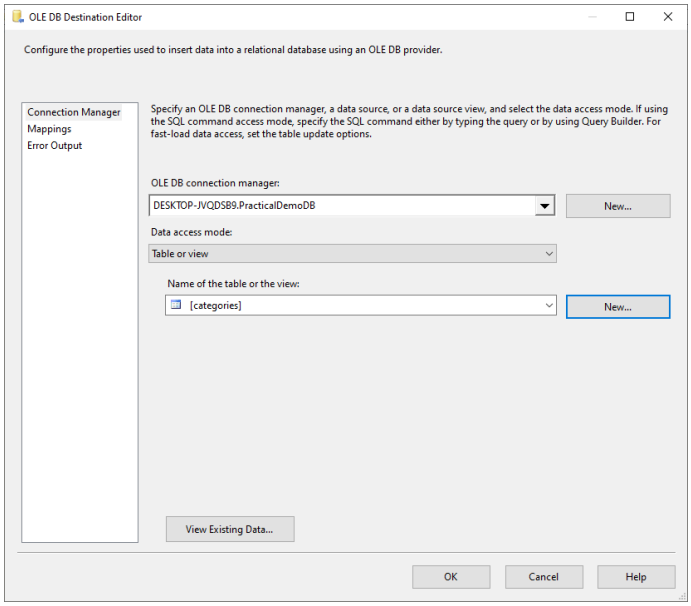


Now place, OLE DB Destination

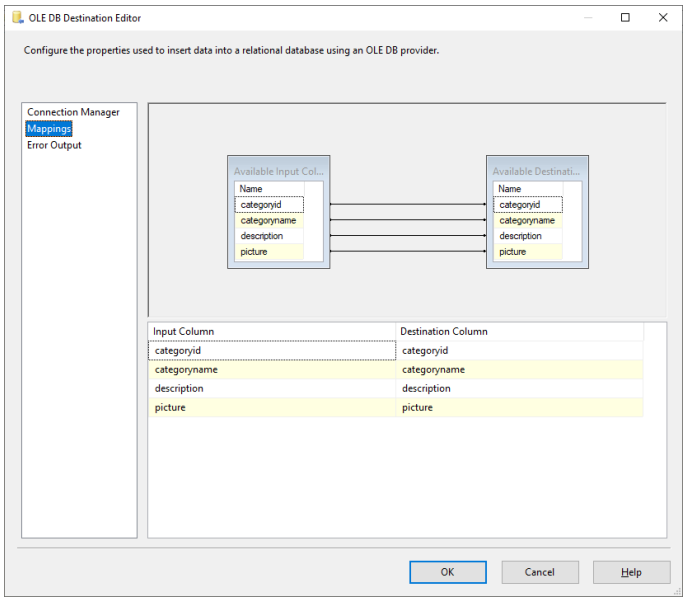


Open OLE DB Destination,

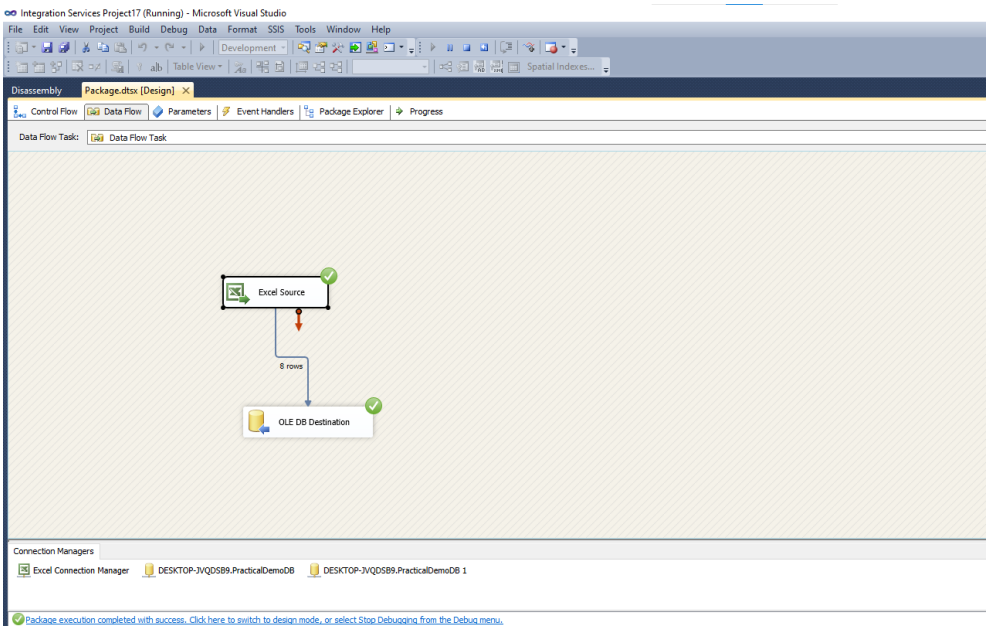




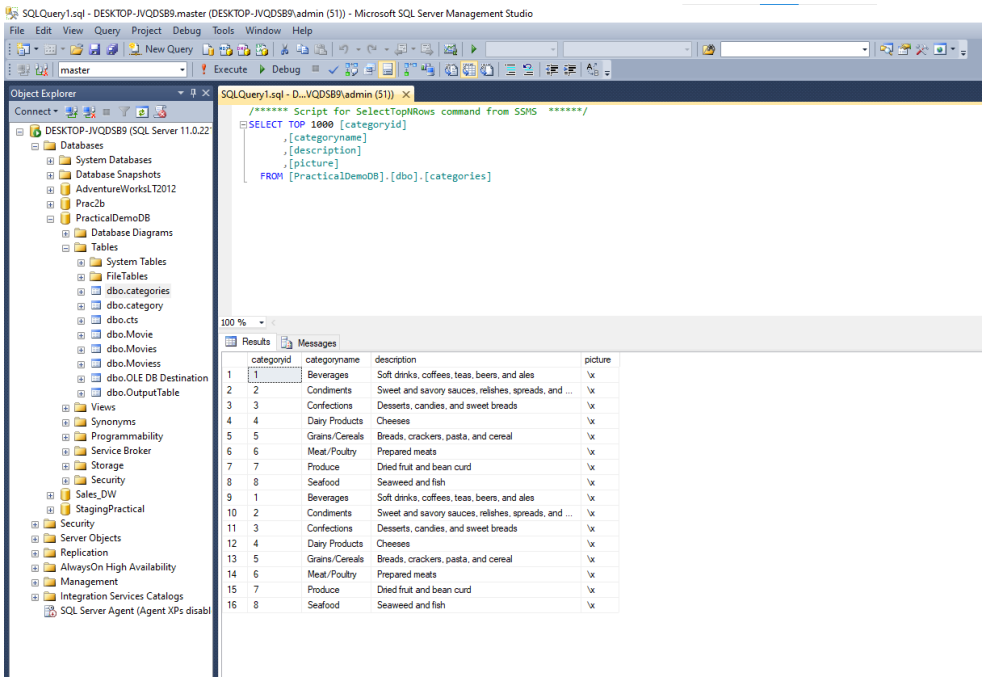
Map the columns



Now execute it,

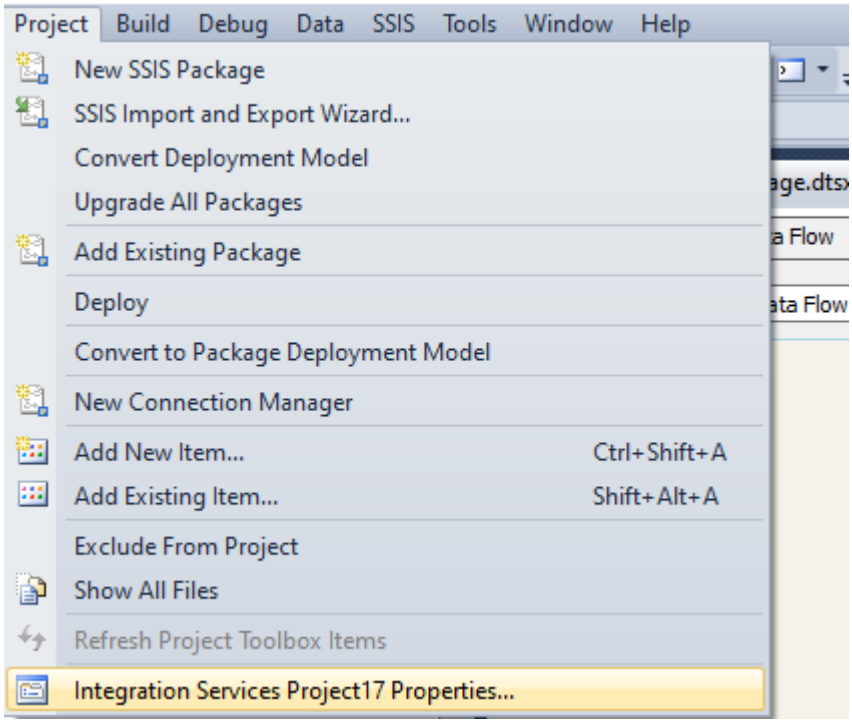


Verifying it in Sql management studio

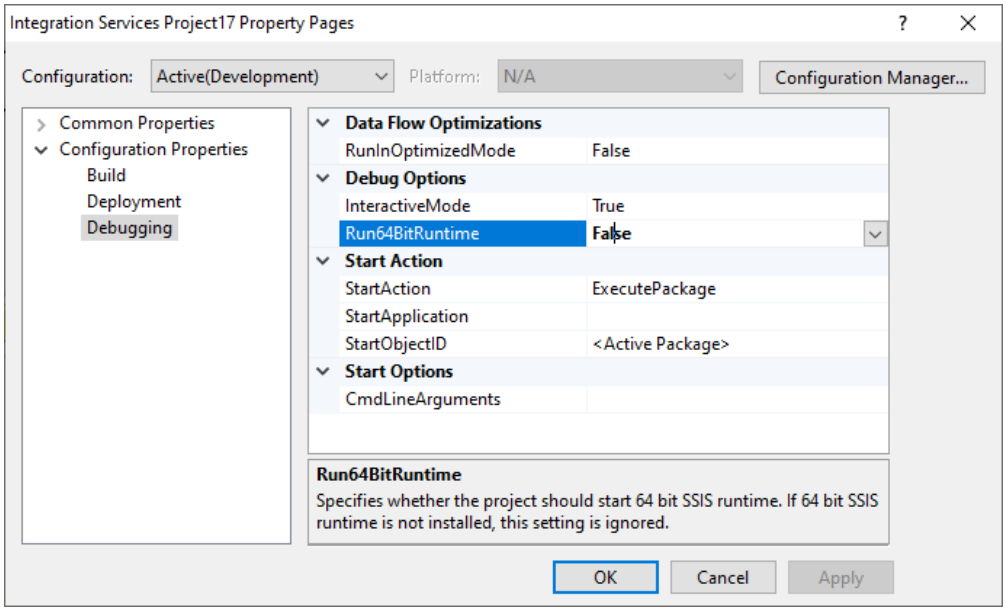


If error occurs while importing the excel data, follow below steps.

Go to Project properties.



Set Run64BitRuntime to false.



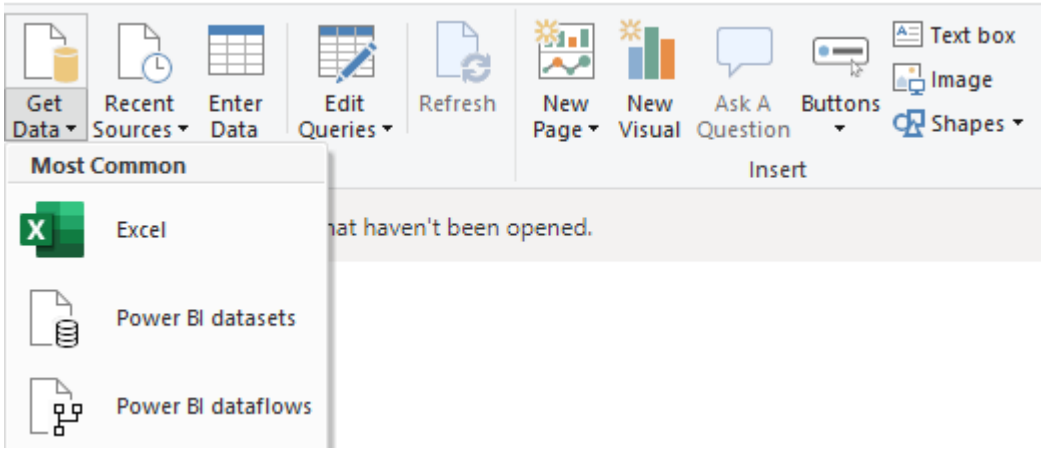
Practical 2

AIM: Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver.

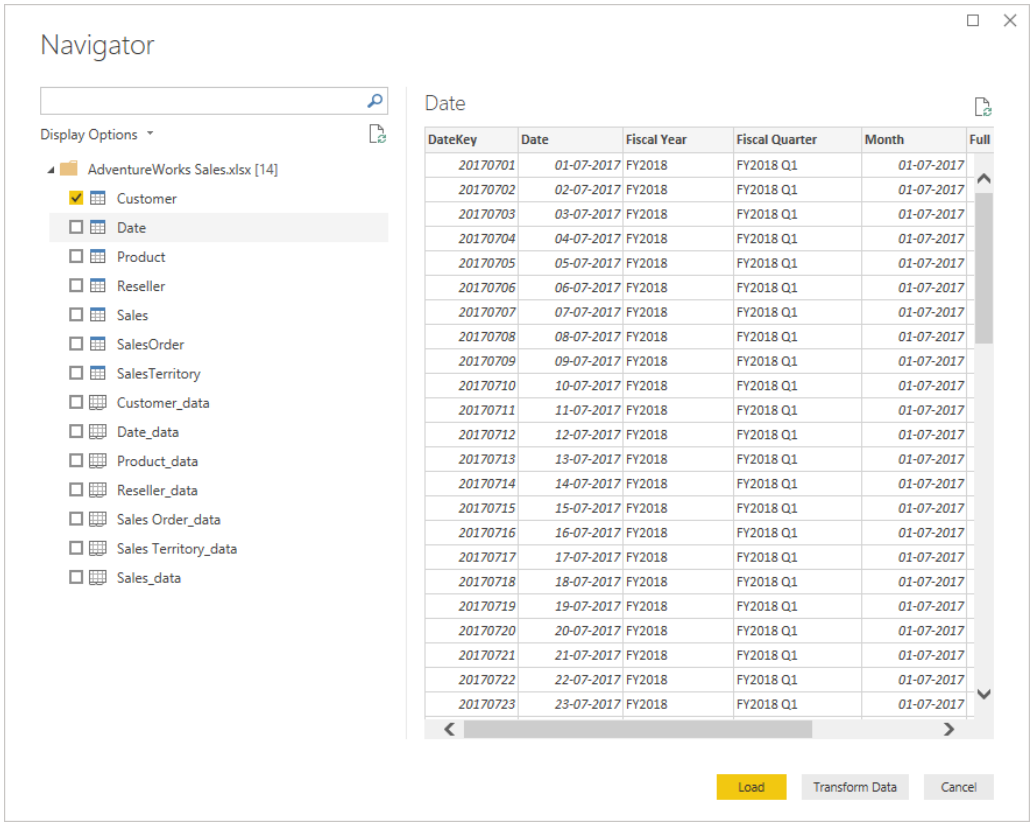
Solution:

In Power Bi,

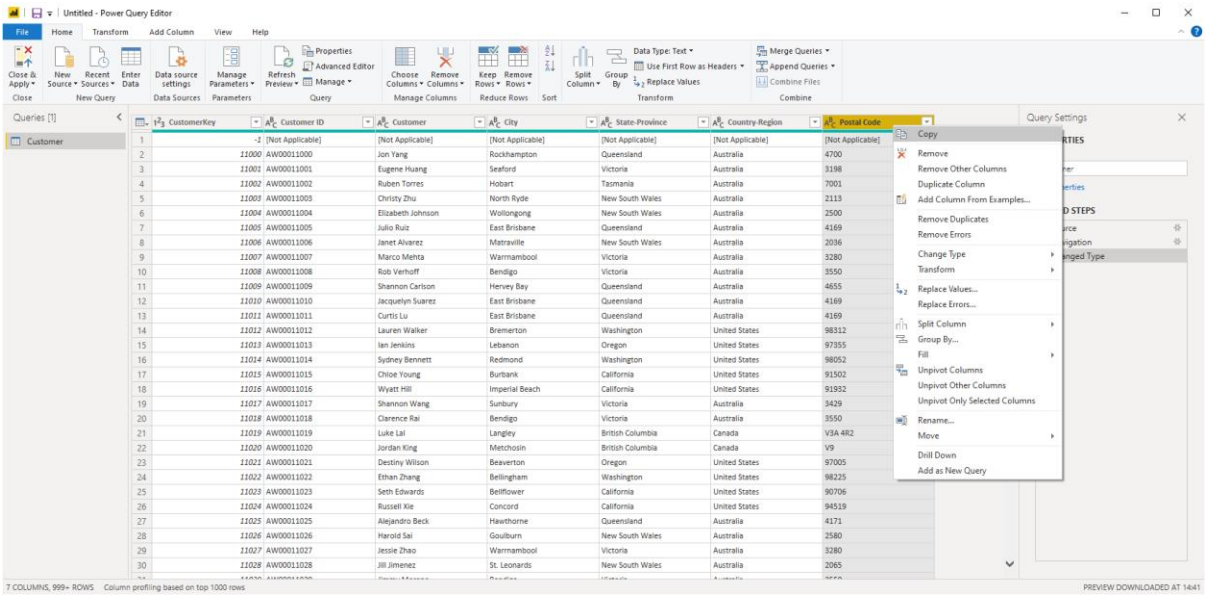
Import your desired data,



Select Tables.

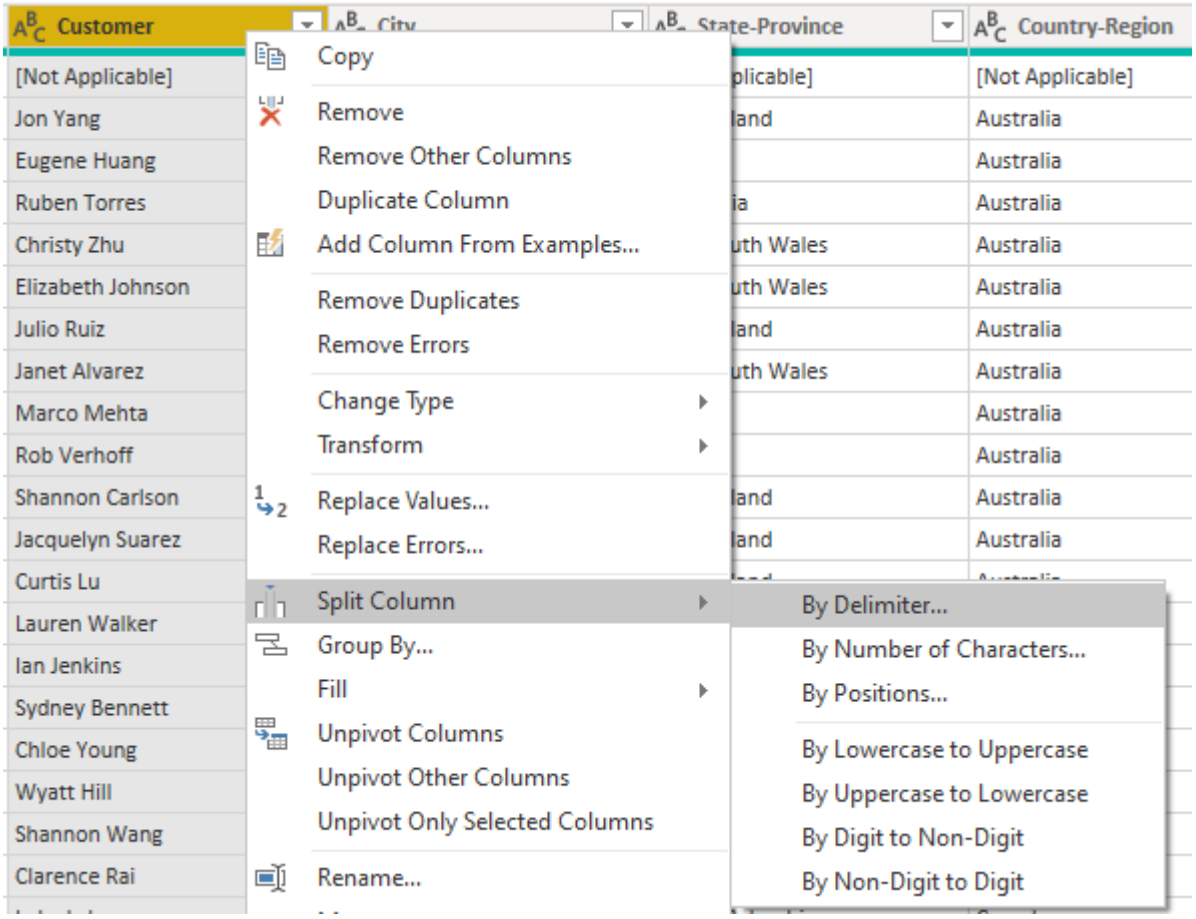


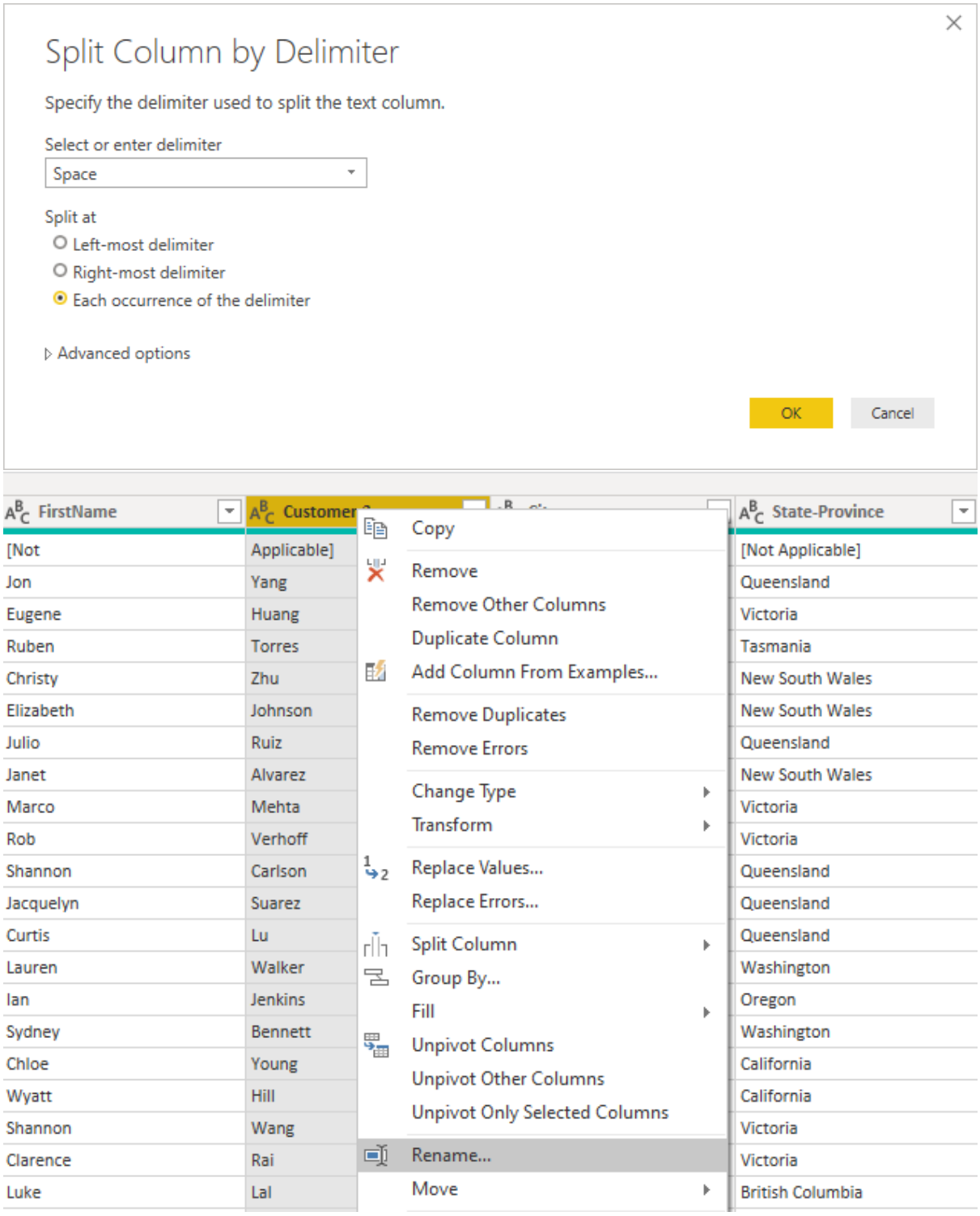
Click on Transform Data

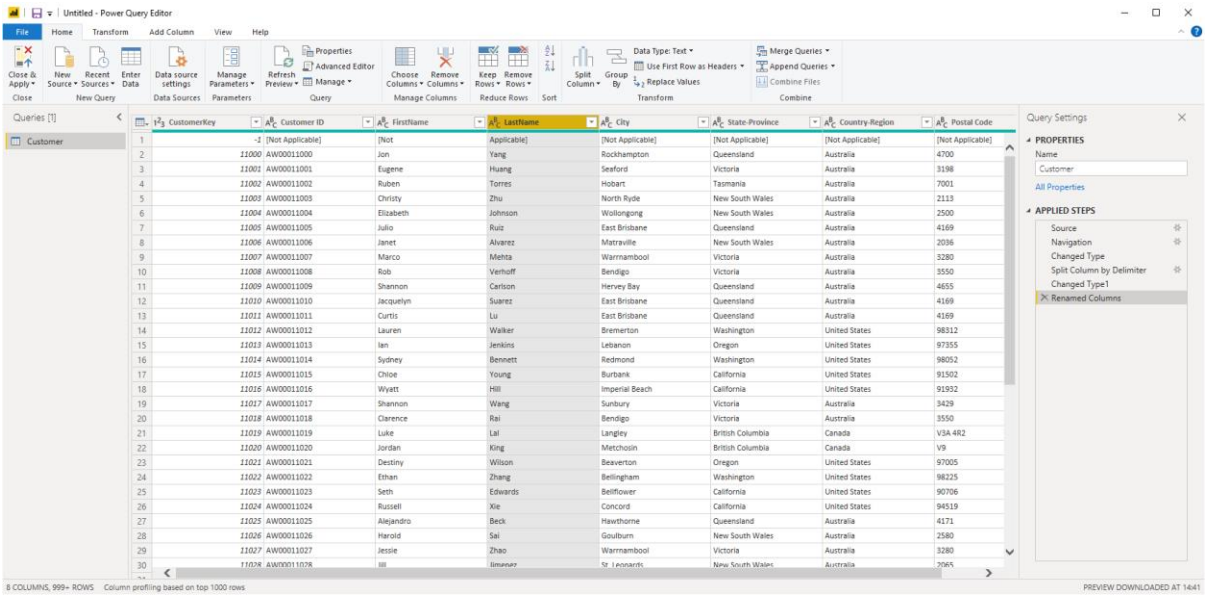


Transformation 1:

Splitting customer column into FirstName and LastName

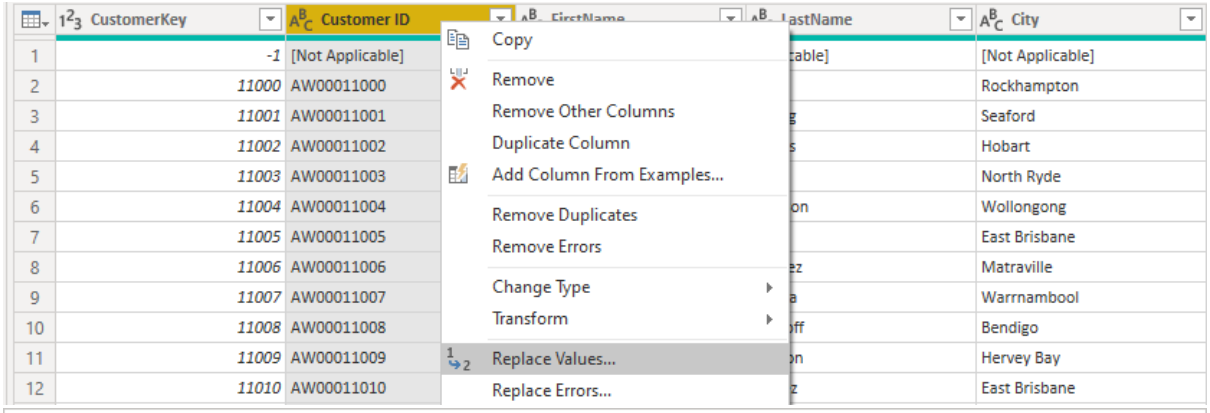


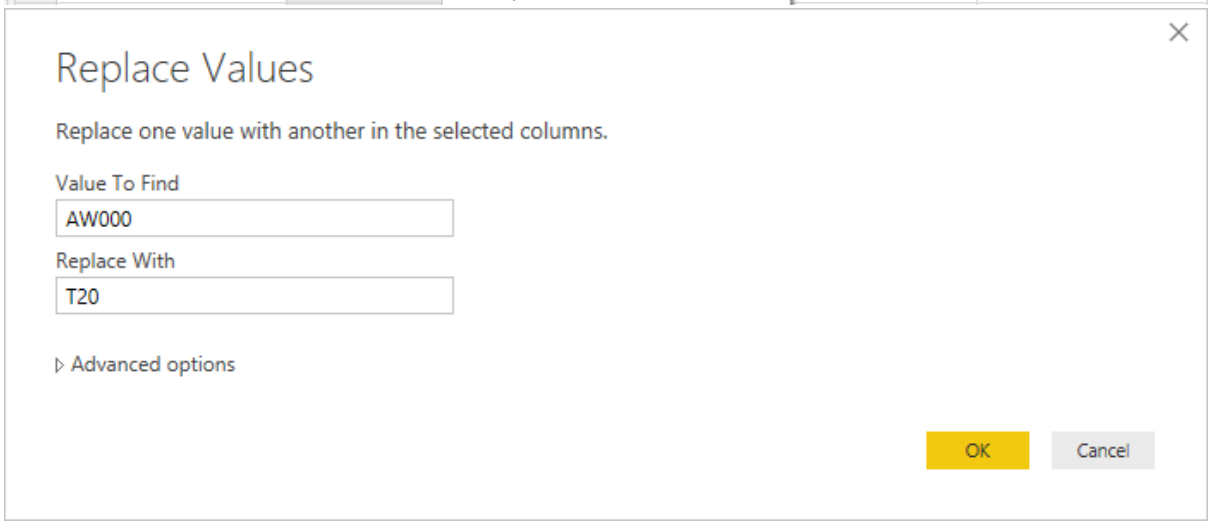

Split Column by Delimiter
Specify the delimiter used to split the text column.
Select or enter delimiter
Space
Split at
Left-most delimiter
Right-most delimiter
Each occurrence of the delimiter
Advanced options
OK
Cancel
FirstName
Customer
State-Province
[Not
Jon
Eugene
Ruben
Christy
Elizabeth
Julio
Janet
Marco
Rob
Shannon
Jacquelyn
Curtis
Lauren
Ian
Sydney
Chloe
Wyatt
Shannon
Clarence
Luke
Applicable]
Yang
Huang
Torres
Zhu
Johnson
Ruiz
Alvarez
Mehta
Verhoff
Carlson
Suarez
Lu
Walker
Jenkins
Bennett
Young
Hill
Wang
Rai
Lal
[Not Applicable]
Queensland
Victoria
Tasmania
New South Wales
New South Wales
Queensland
New South Wales
Victoria
Victoria
Queensland
Queensland
Queensland
Washington
Oregon
Washington
California
California
Victoria
Victoria
British Columbia
Copy
Remove
Remove Other Columns
Duplicate Column
Add Column From Examples...
Remove Duplicates
Remove Errors
Change Type
Transform
Replace Values...
Replace Errors...
Split Column
Group By...
Fill
Unpivot Columns
Unpivot Other Columns
Unpivot Only Selected Columns
Rename...
Move

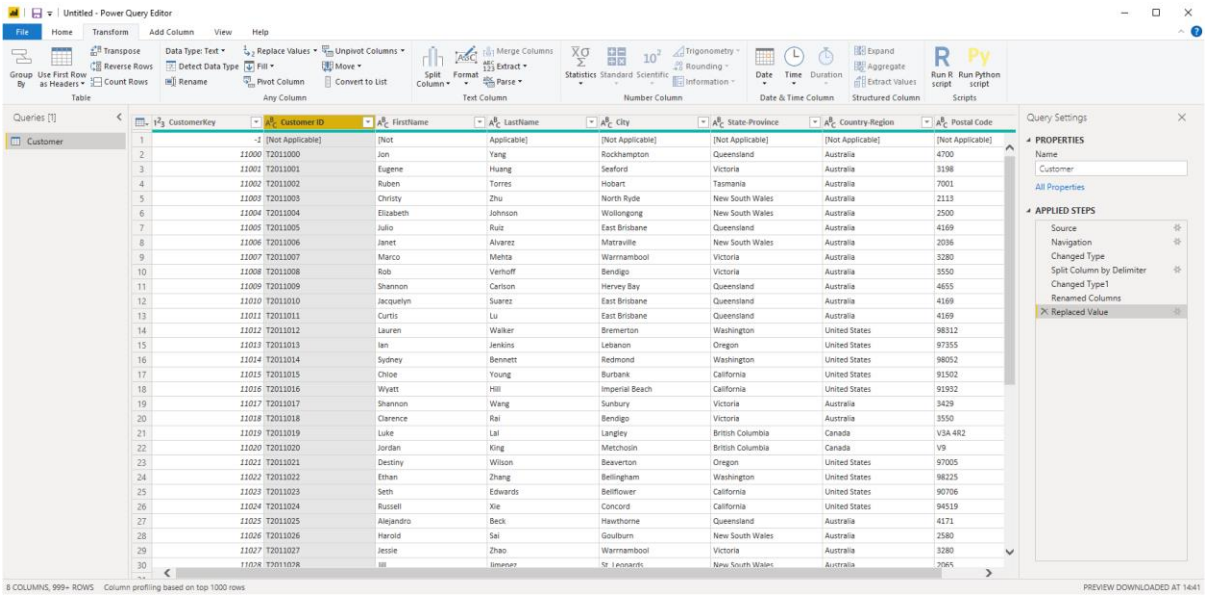


Transformation 2:

Replacing AW000 in Customer ID column with T20

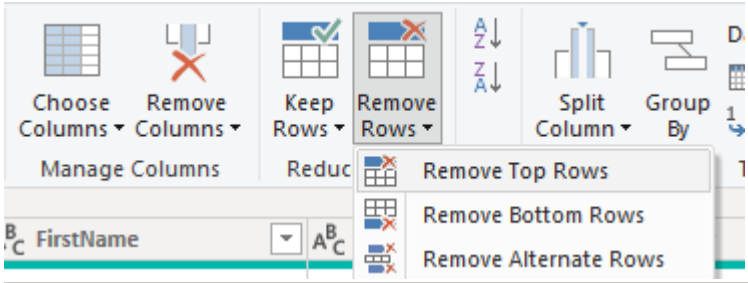


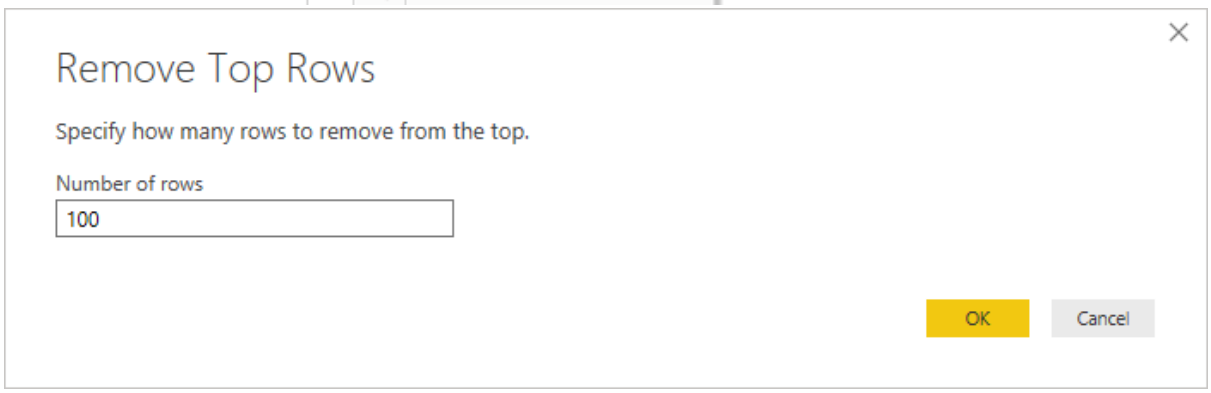


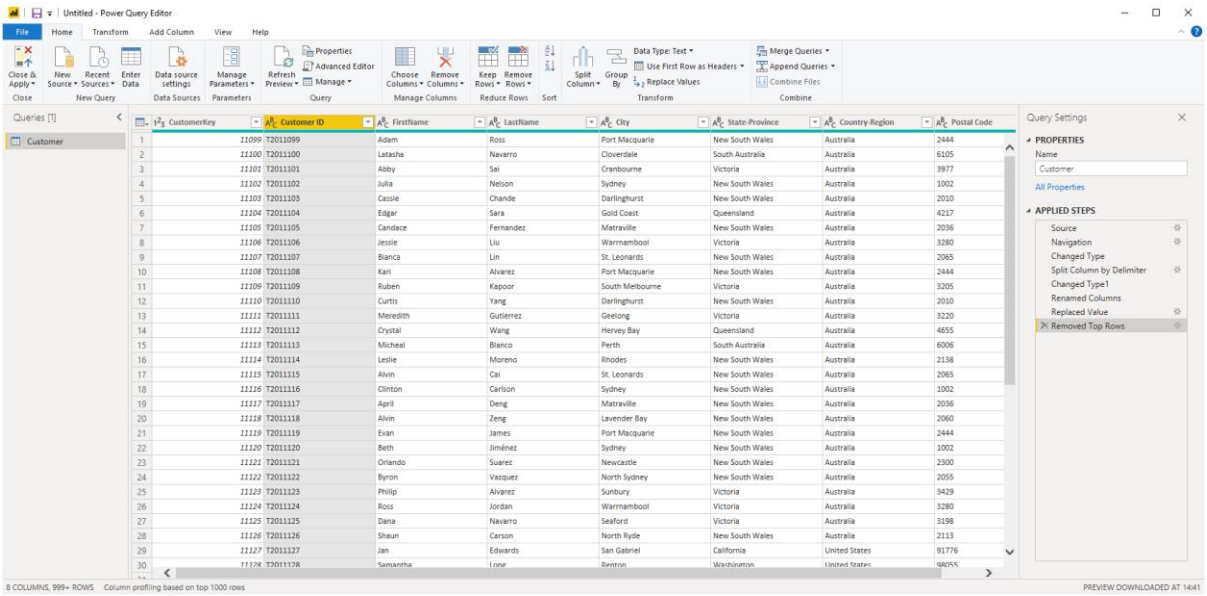


Transformation 3:

Deleting first 100 records of all columns

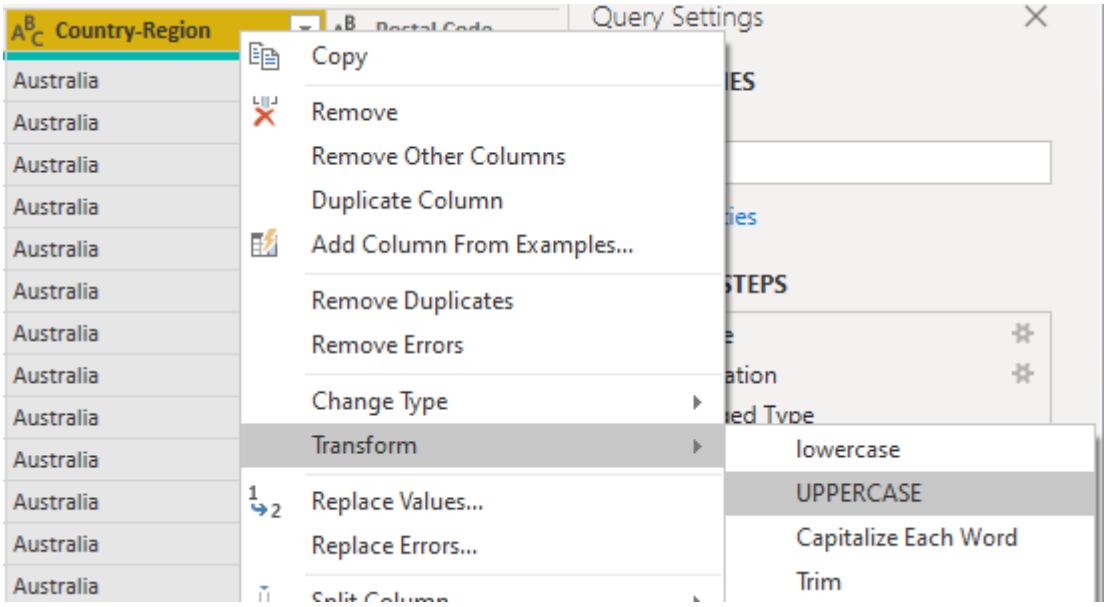


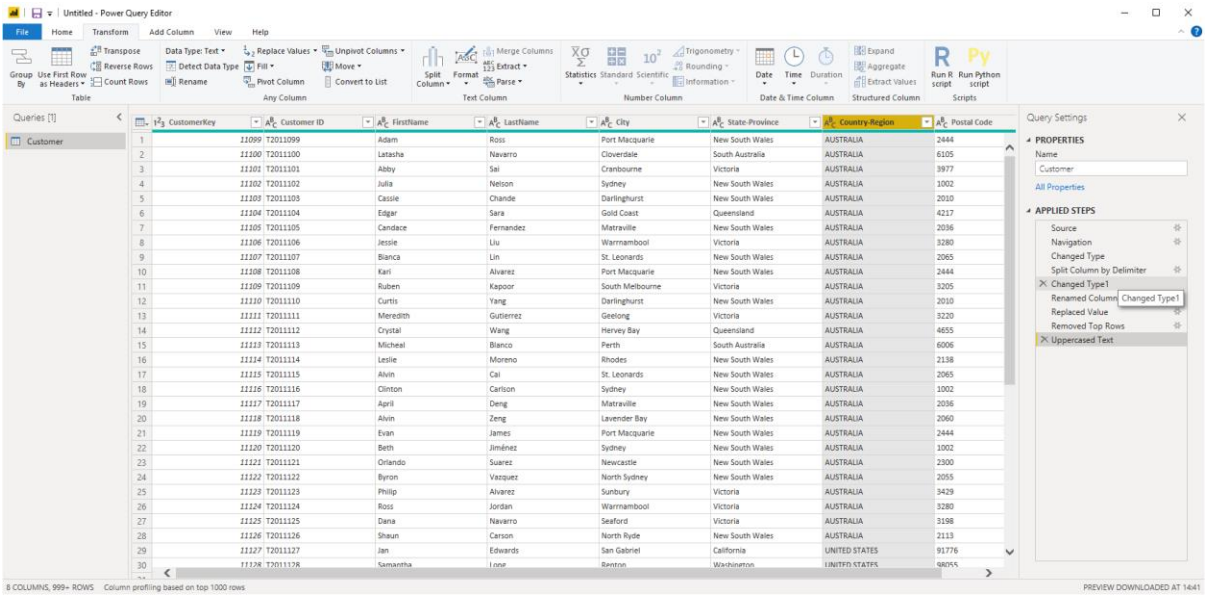




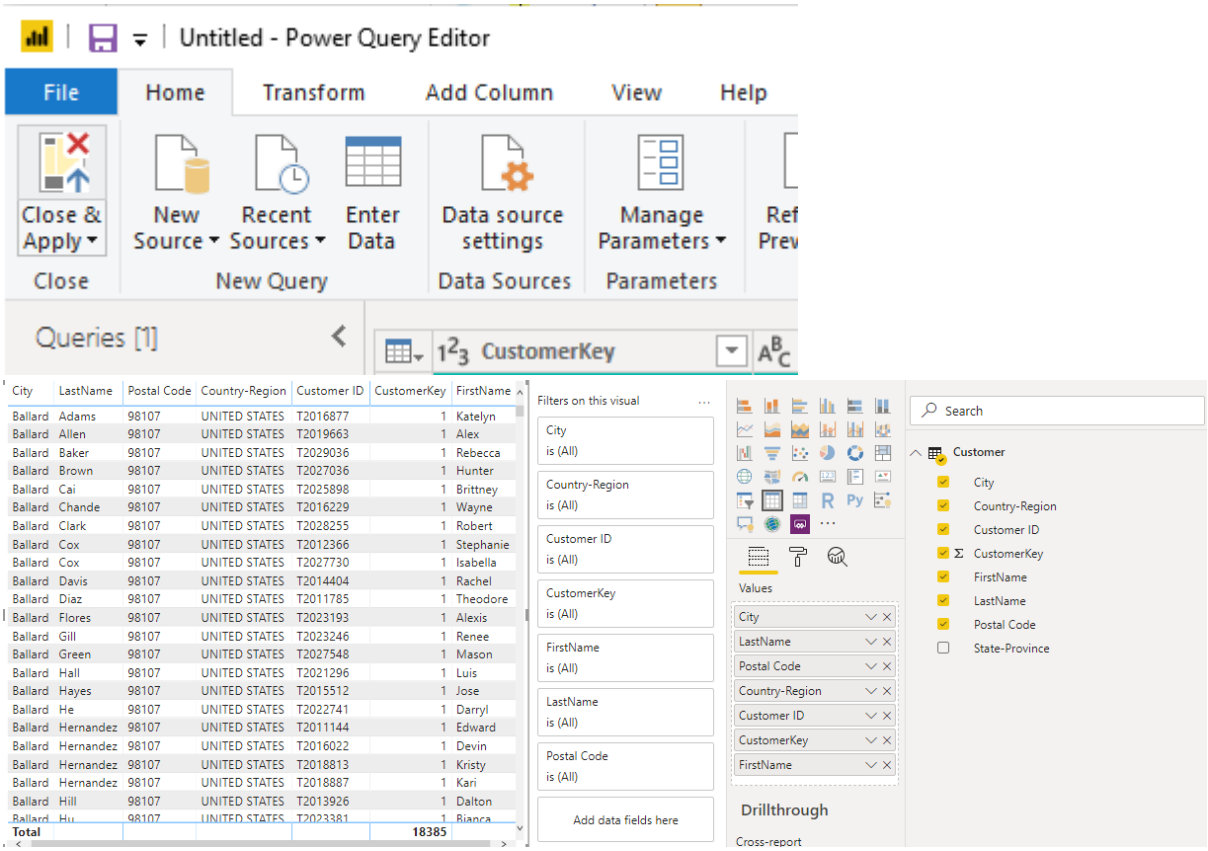
Transformation 4:

Converting Country-Region column values to uppercase





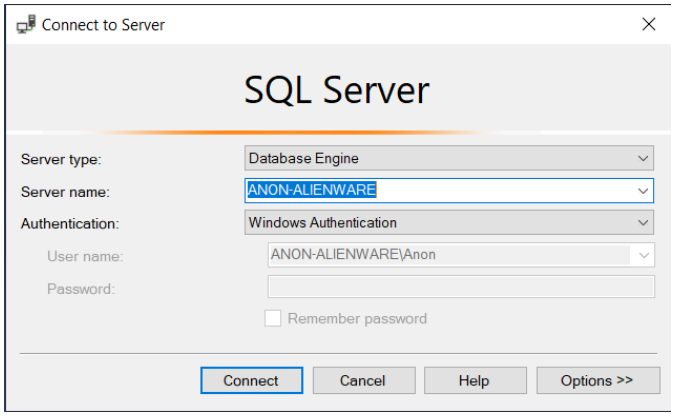
Finally click on “close and apply” to import



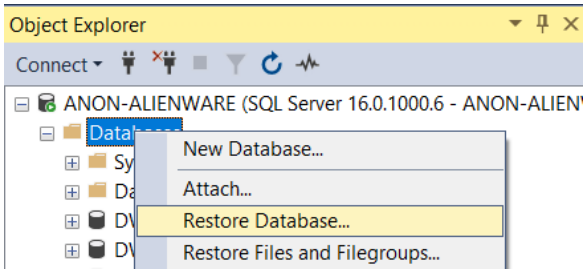
City	LastName	Postal Code	Country-Region	Customer ID	CustomerKey	FirstName
Ballard	Adams	98107	UNITED STATES	T2016877	1	Katelyn
Ballard	Allen	98107	UNITED STATES	T2019663	1	Alex
Ballard	Baker	98107	UNITED STATES	T2029036	1	Rebecca
Ballard	Brown	98107	UNITED STATES	T2027036	1	Hunter
Ballard	Cai	98107	UNITED STATES	T2025898	1	Brittney
Ballard	Chande	98107	UNITED STATES	T2016229	1	Wayne
Ballard	Clark	98107	UNITED STATES	T2028255	1	Robert
Ballard	Cox	98107	UNITED STATES	T2012366	1	Stephanie
Ballard	Cox	98107	UNITED STATES	T2027730	1	Isabella
Ballard	Davis	98107	UNITED STATES	T2014404	1	Rachel
Ballard	Diaz	98107	UNITED STATES	T2011785	1	Theodore
Ballard	Flores	98107	UNITED STATES	T2023193	1	Alexis
Ballard	Gill	98107	UNITED STATES	T2023246	1	Renee
Ballard	Green	98107	UNITED STATES	T2027548	1	Mason
Ballard	Hall	98107	UNITED STATES	T2021296	1	Luis
Ballard	Hayes	98107	UNITED STATES	T2015512	1	Jose
Ballard	He	98107	UNITED STATES	T2022741	1	Darryl
Ballard	Hernandez	98107	UNITED STATES	T2011144	1	Edward
Ballard	Hernandez	98107	UNITED STATES	T2016022	1	Devin
Ballard	Hernandez	98107	UNITED STATES	T2018813	1	Kristy
Ballard	Hernandez	98107	UNITED STATES	T2018887	1	Kari
Ballard	Hill	98107	UNITED STATES	T2013926	1	Dalton
Ballard	Hu	98107	UNITED STATES	T2023381	1	Bianca
Total					18385	

In SQL Server Data Tools.

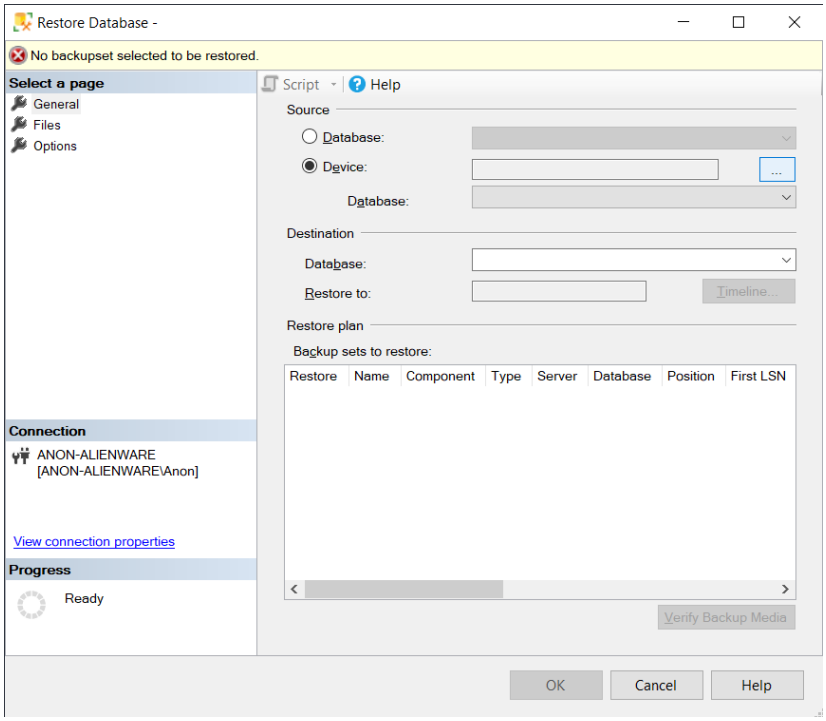
Step 1: Open SQL server management studio to restore backup file.

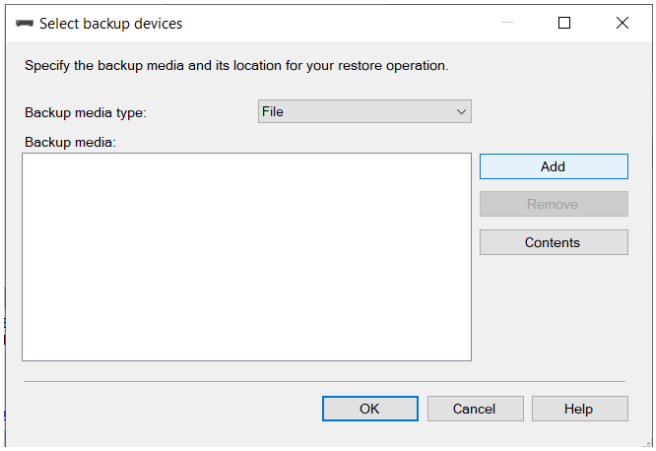


Step 2: Right click on the databases -> restore database

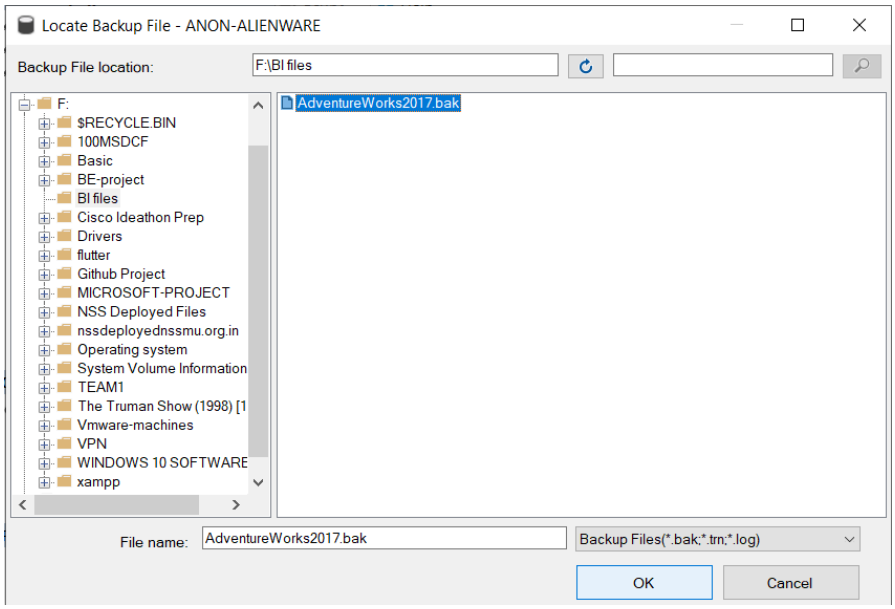


Step 3 : Select device -> click on 3 dots icon towards the end of device box

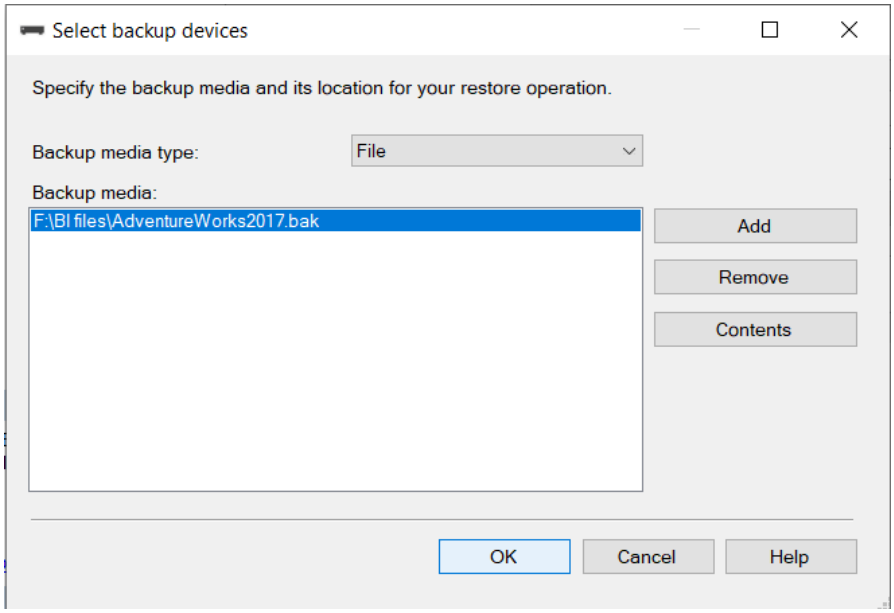




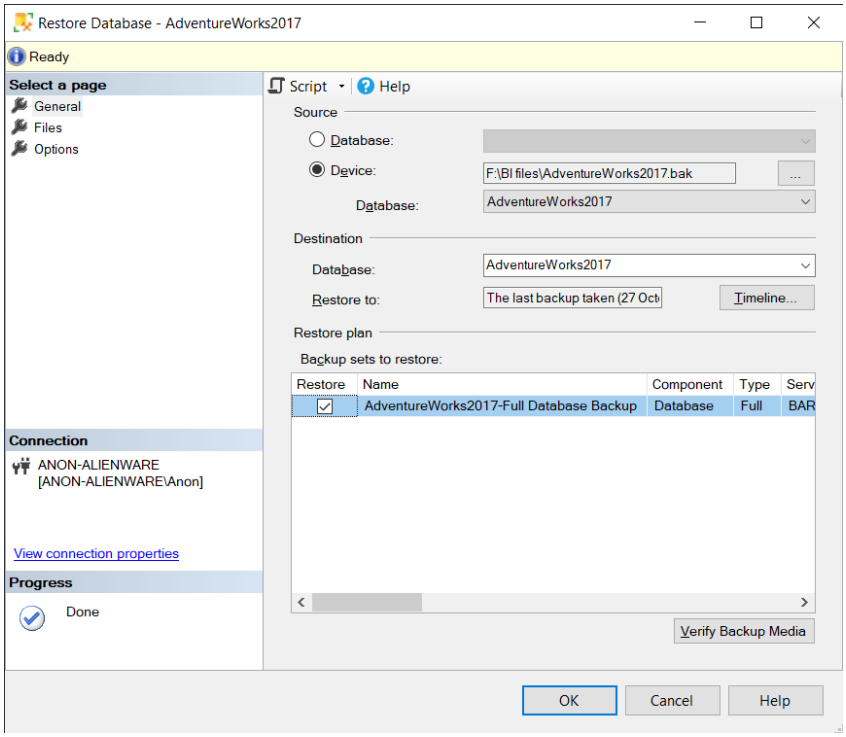
Step 4: Click on Add -> the path of backup file

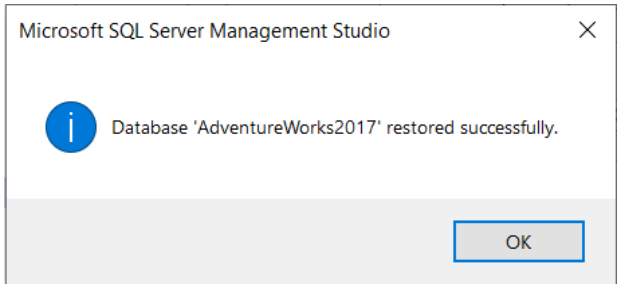


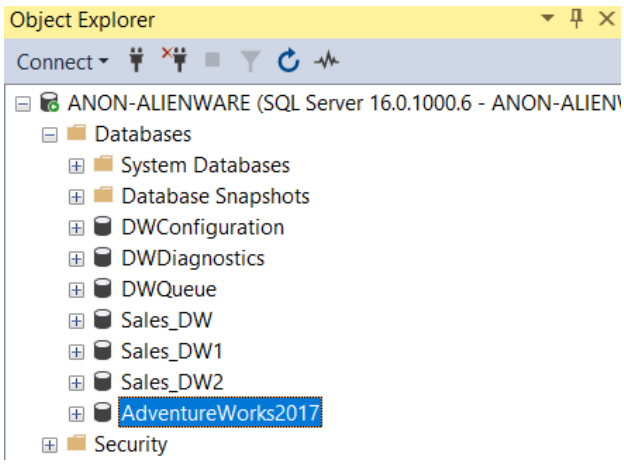
Step 5: Select the file



Step 6: Click on OK



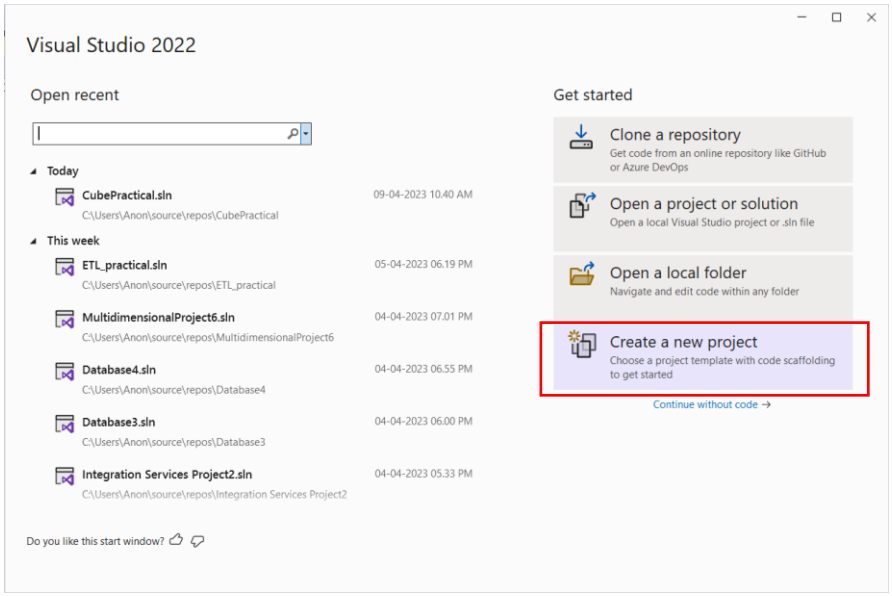


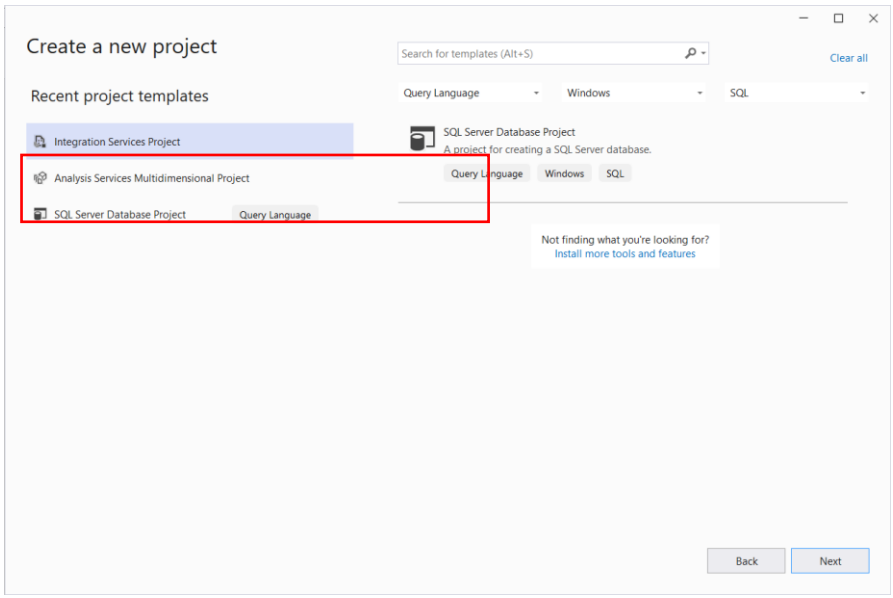


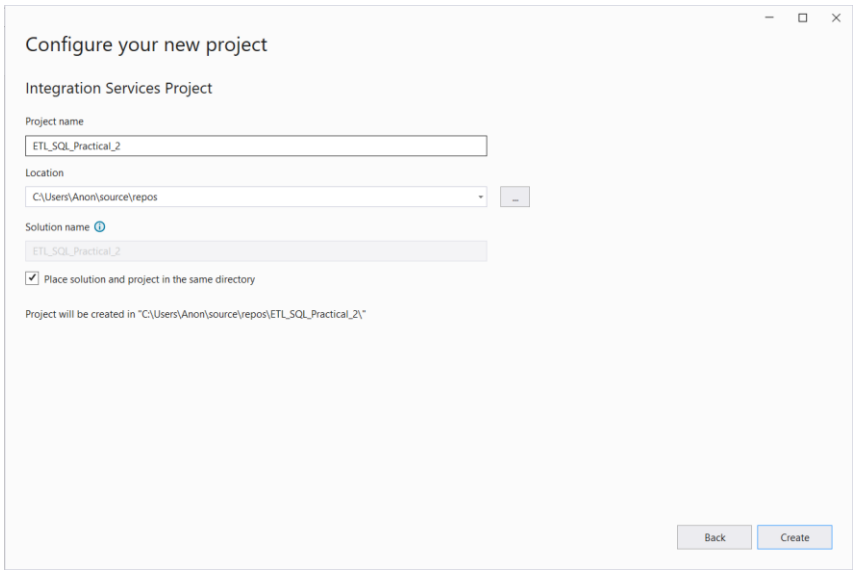
The database is restored successfully.

Step 7: Open SQL server data tools

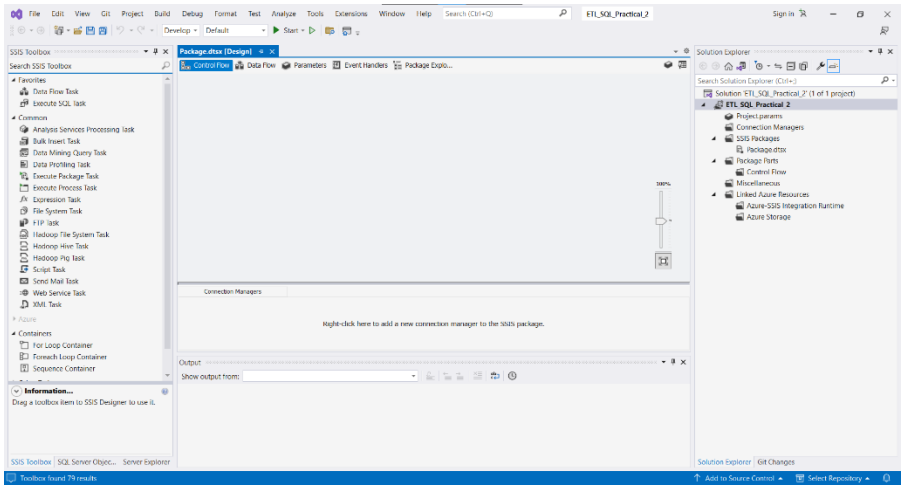
Select File -> new -> project -> Business Intelligence -> **Integration services project** & give project name



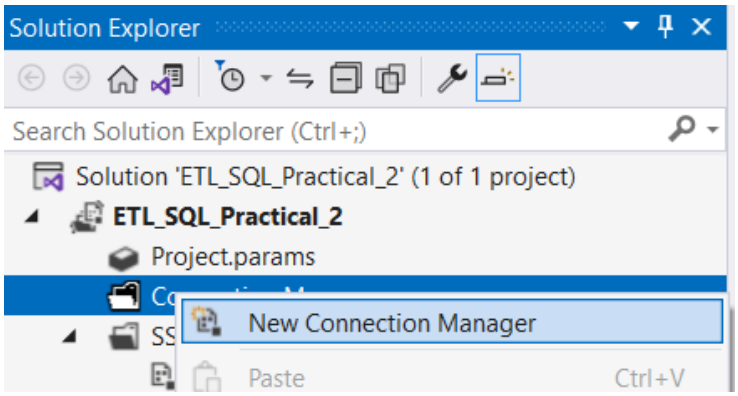




Environment consist of SQL Server Integration Services (SSIS)

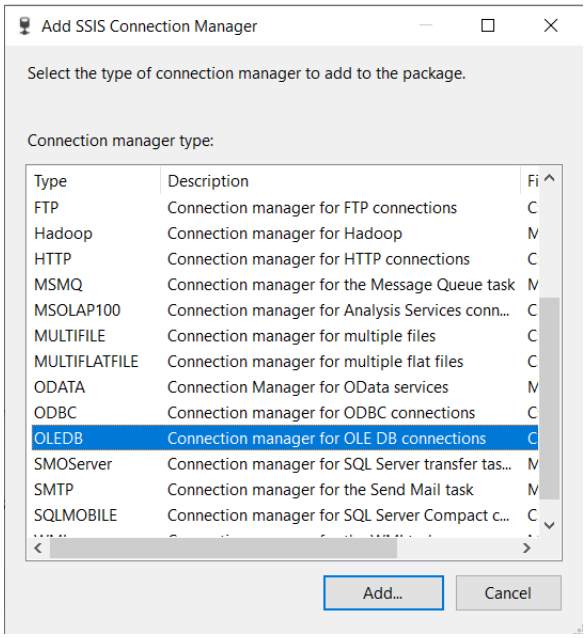


Step 8: Right click on connection manager in solution explorer and click on new connection manager

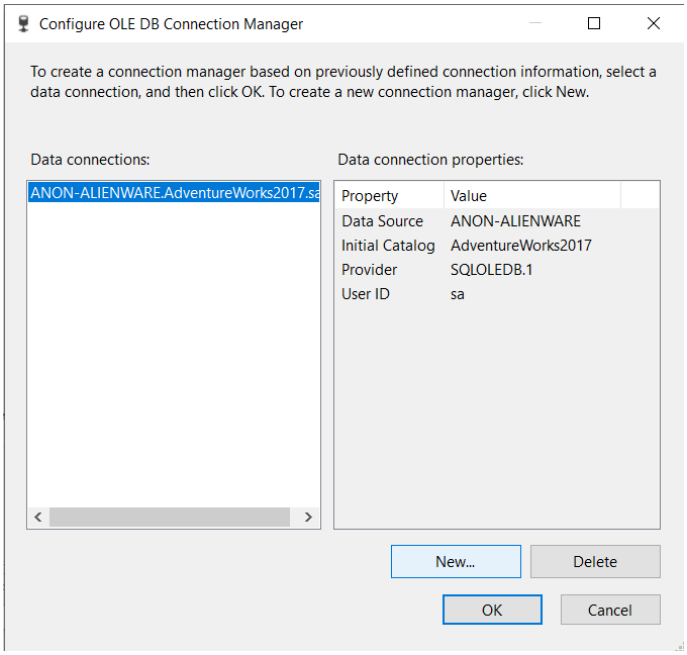


Add SSIS connection manager window appears

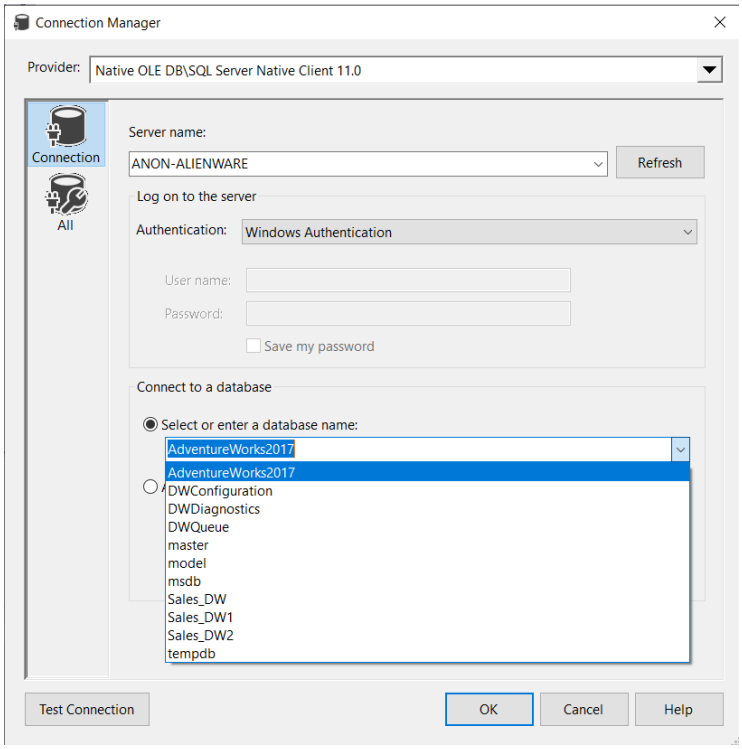
Step 9: Select OLEDB Connection Manager and Click on Add

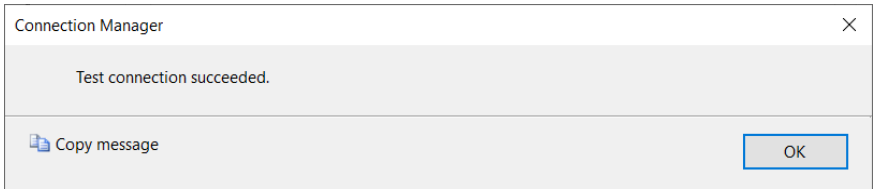


Step 10: Configure OLE DB Connection Manager window appears -> click on new

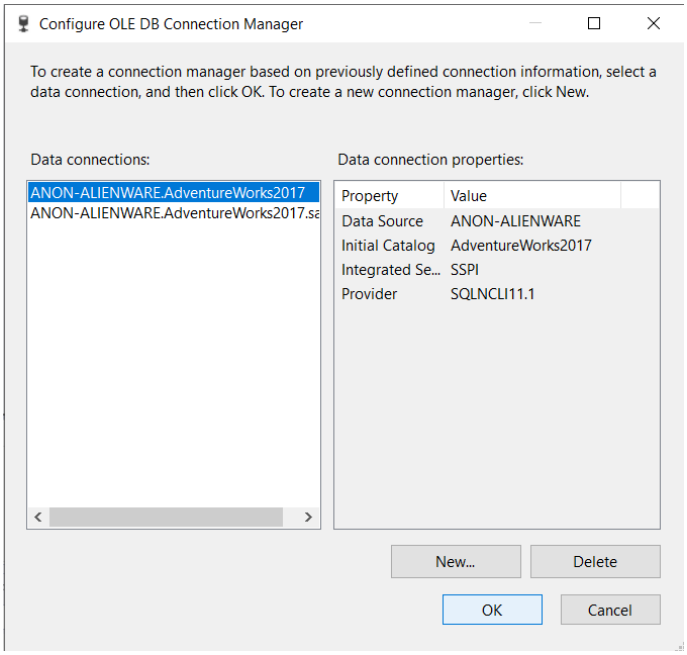


Step 11: Select Server name from drop down and database name and click on test connection

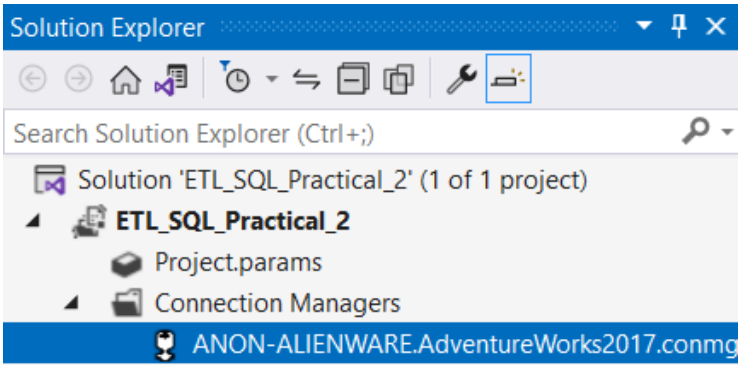




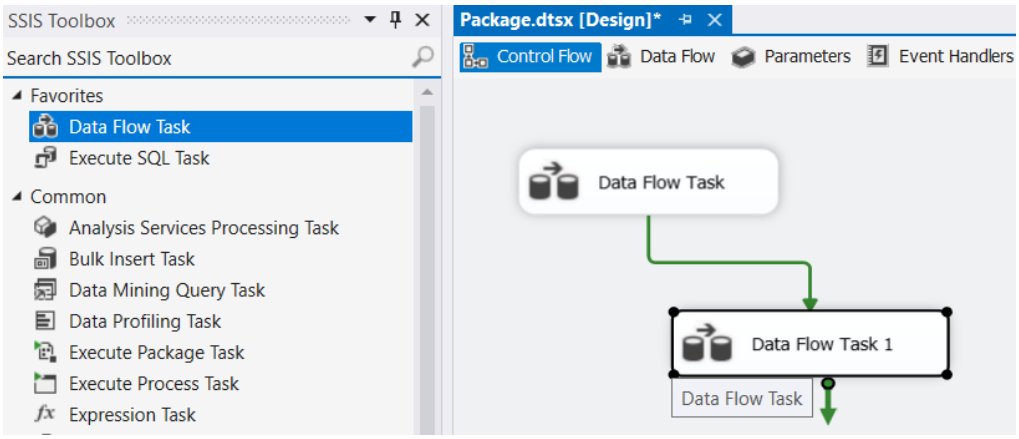
Step 12: Click on OK



Connection is added in connection manager

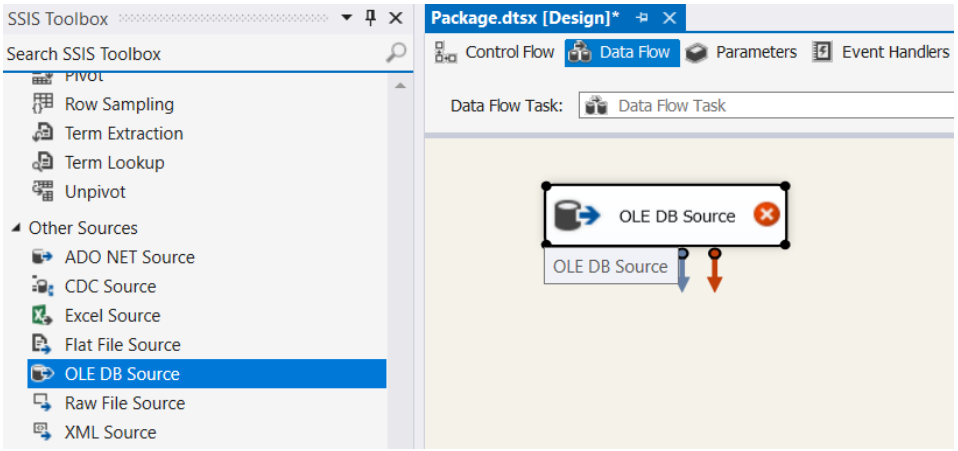


Step 13: Drag and drop **Data Flow Task** in **Control Flow Tab**

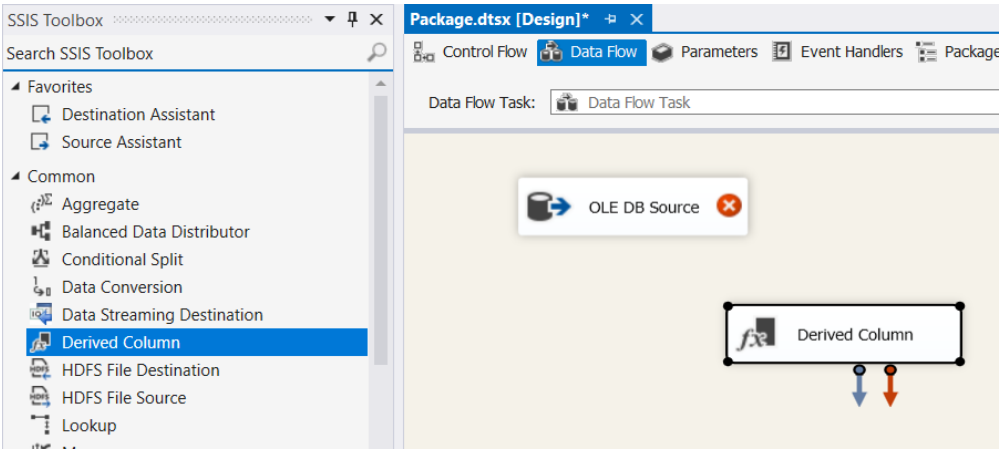


Double click on data flow task

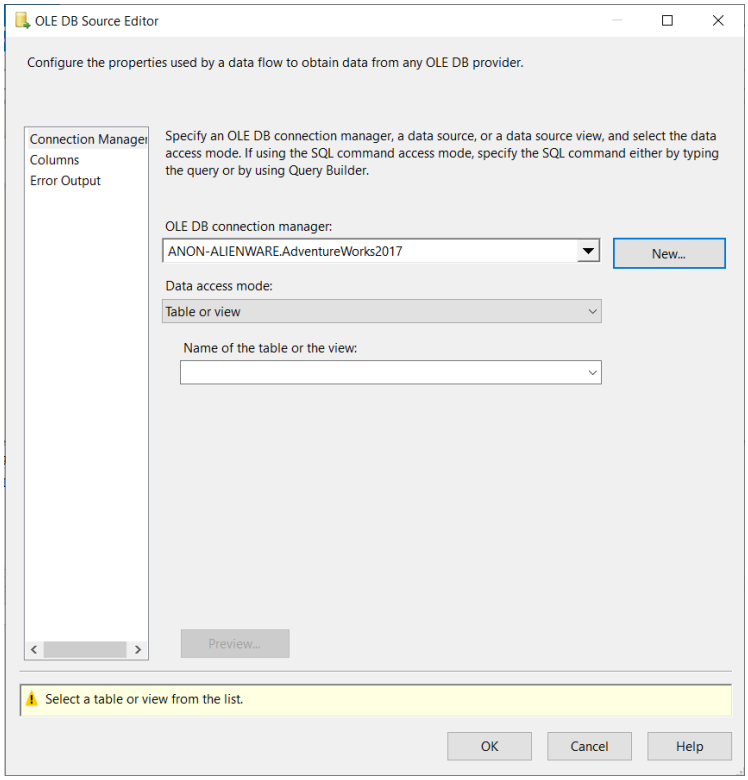
Step 14: Drag and drop **OLE DB Source** from other source into **Data Flow tab**

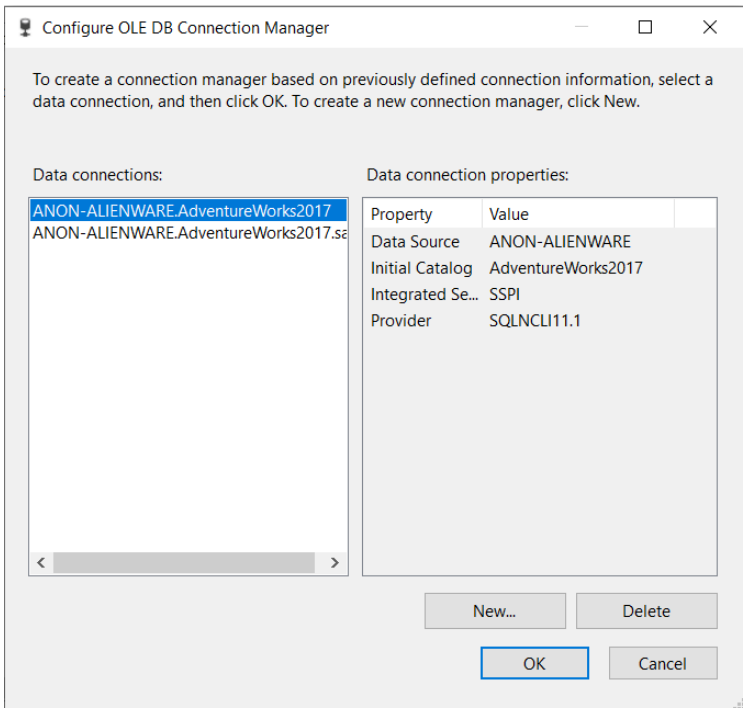


Add derived column from SSIS toolbox -> common

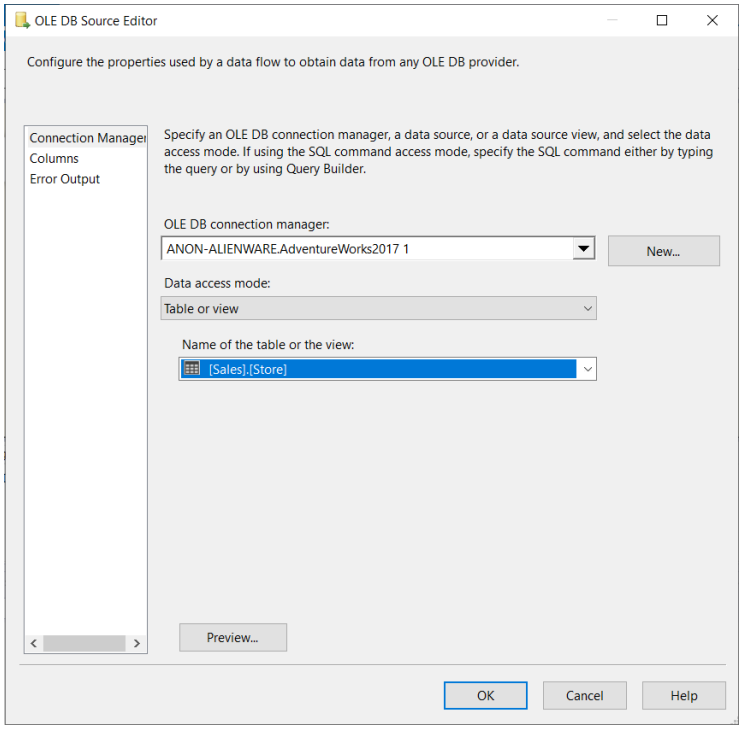


Step 15: Double click on OLE DB Source -> OLE DB source editor appears -> click on new to add connection manager

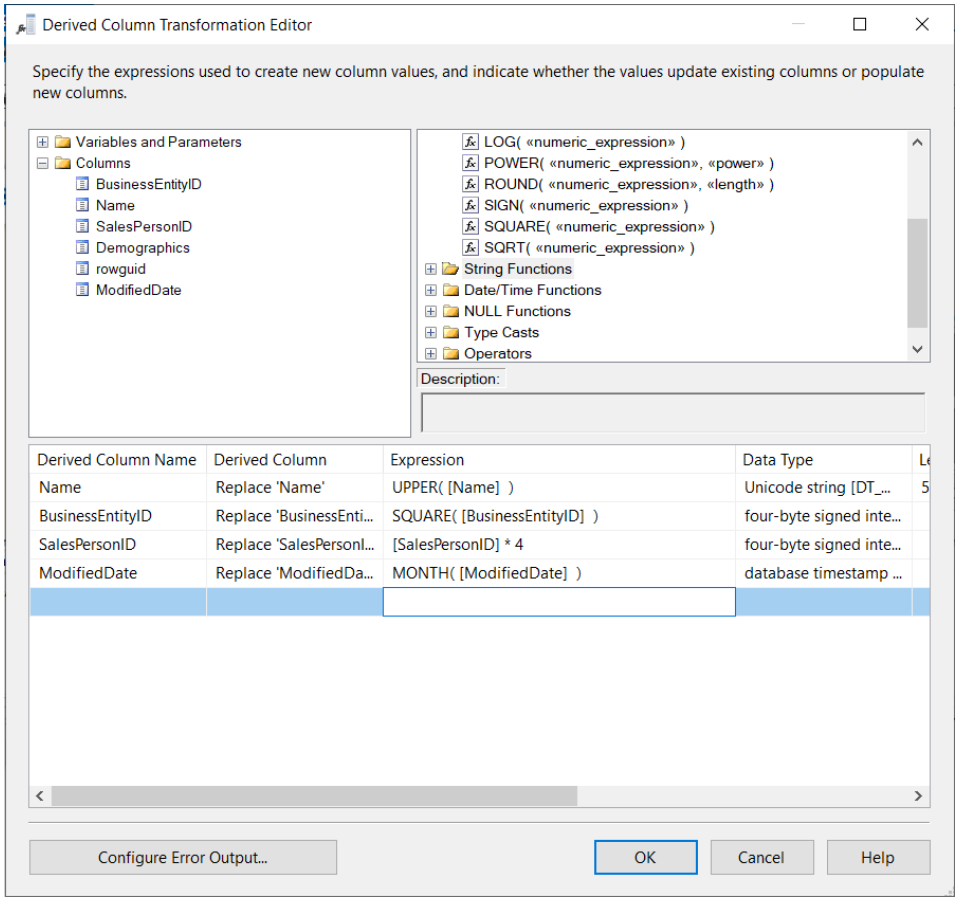




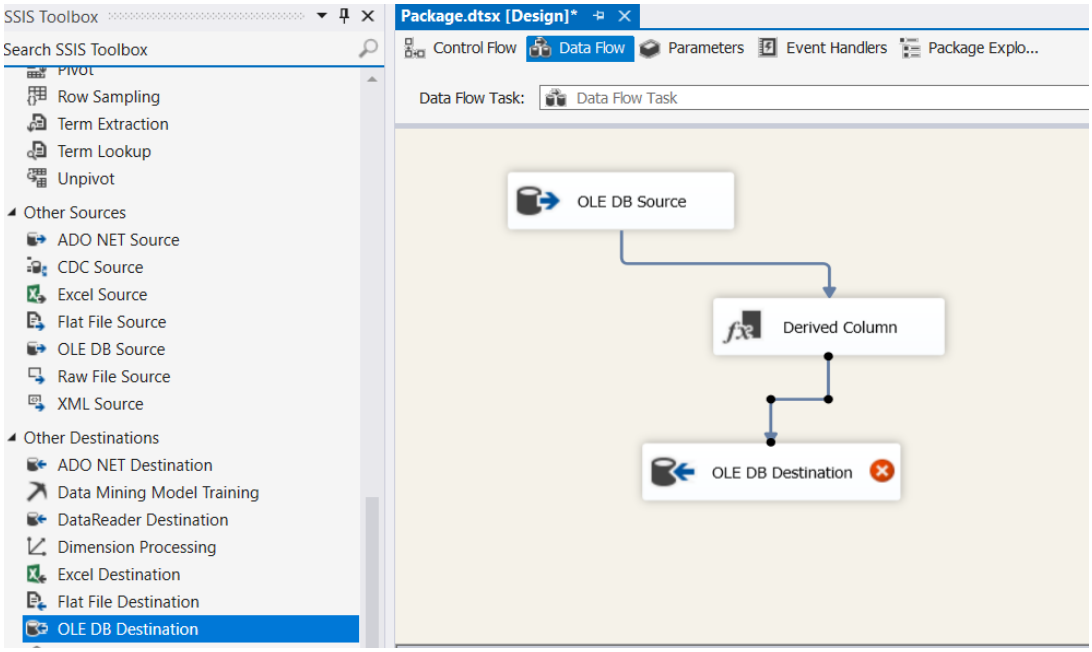
Select [Sales].[Store] table from drop down -> OK



Step 16: Connect OLE DB Source to derived column and Double click on derived column and add the expressions of ETL

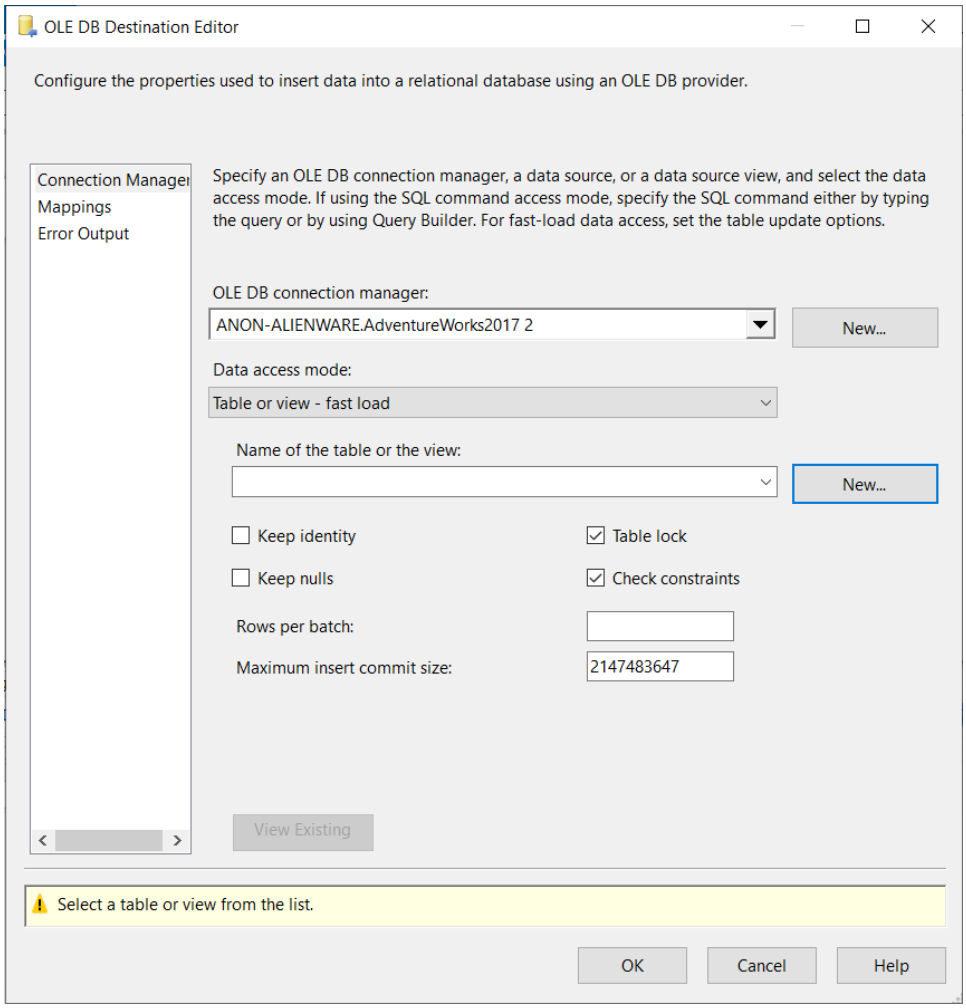


Step 17: Add OLE DB Destination in data flow and connect the derived column to it

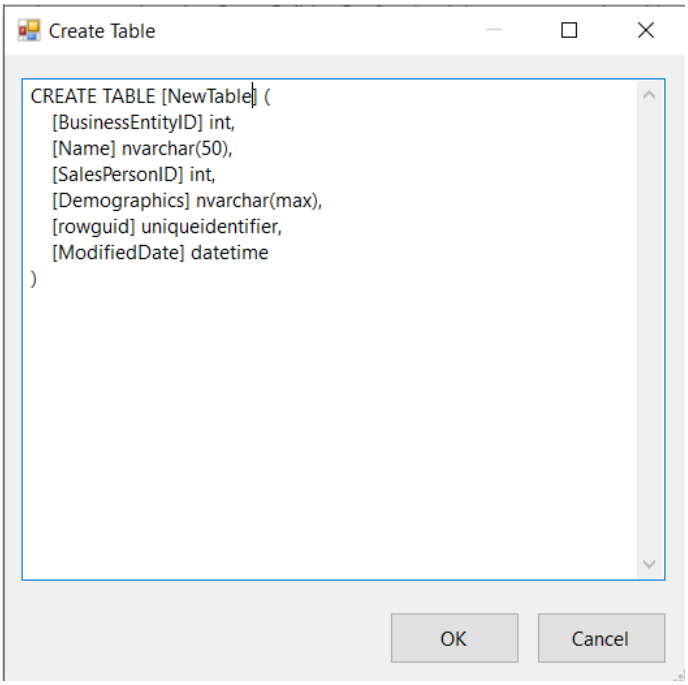


Step 18: Double click on OLE DB Destination

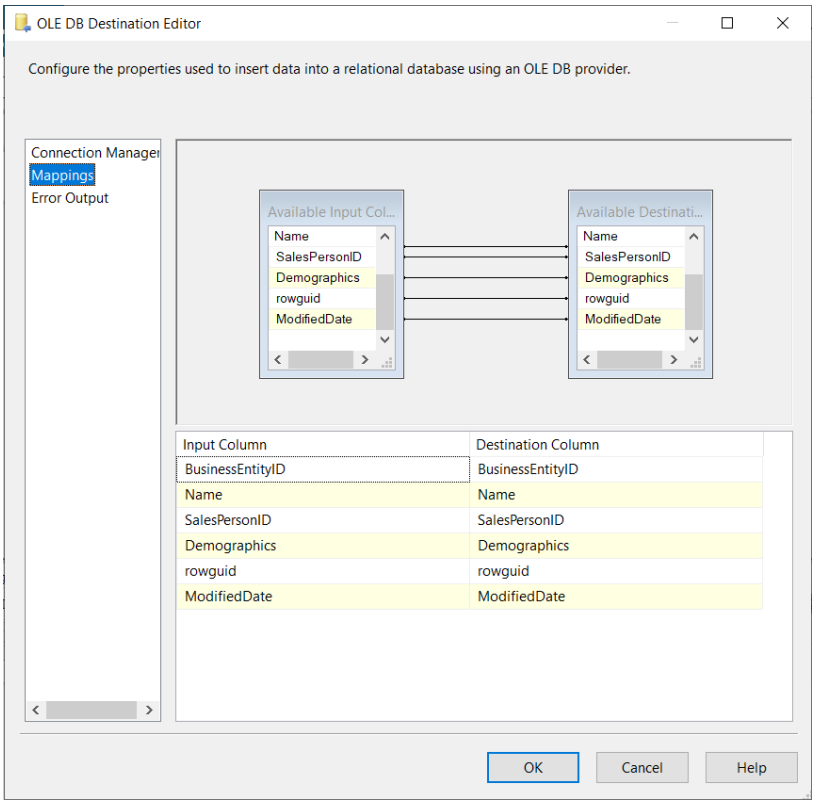
Click on new to run the query to get [OLE DB DESTINATION] in name of the table or the view

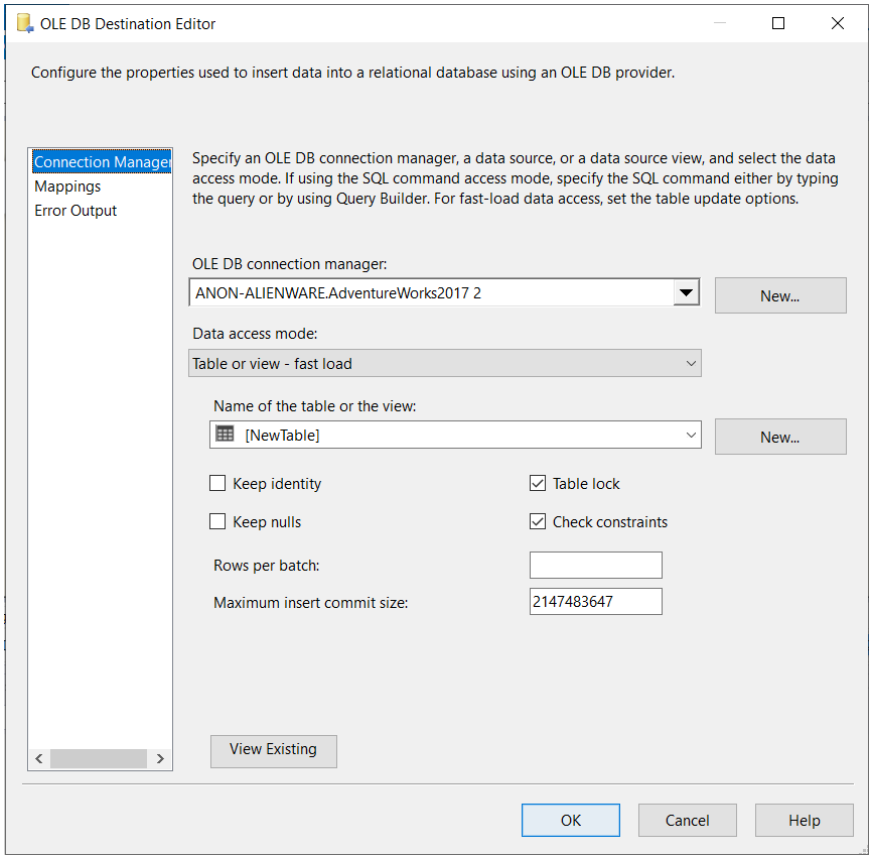


Change the name of table

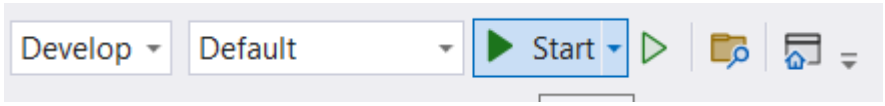


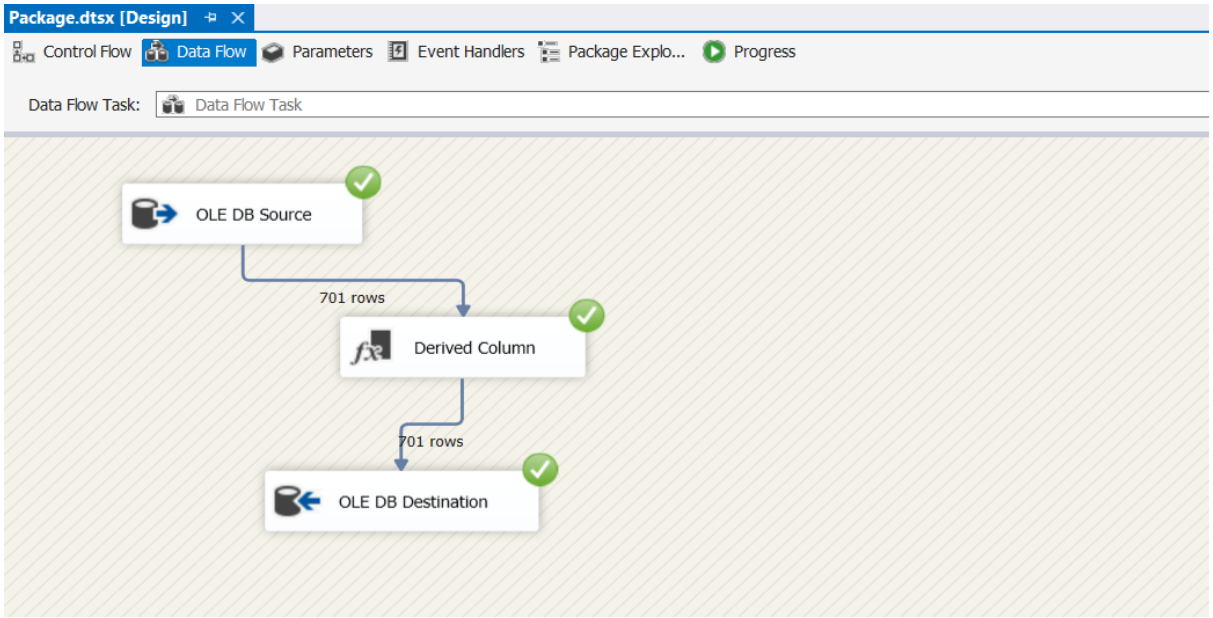
Map the columns



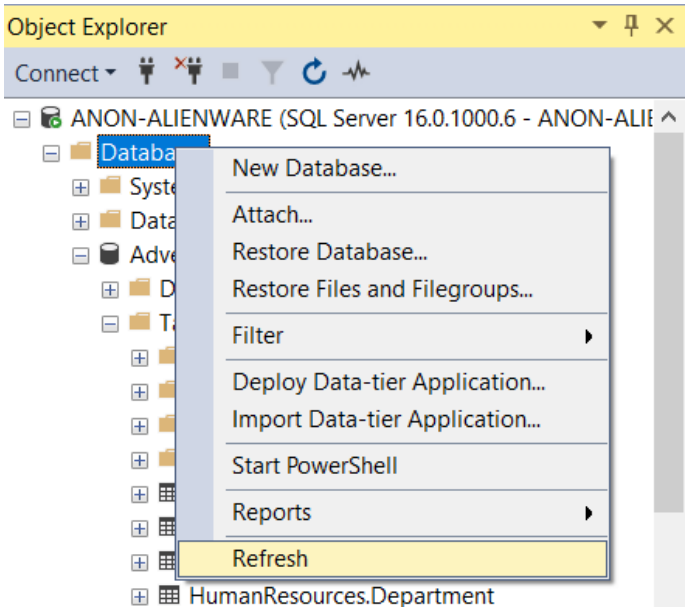


Click on Start button



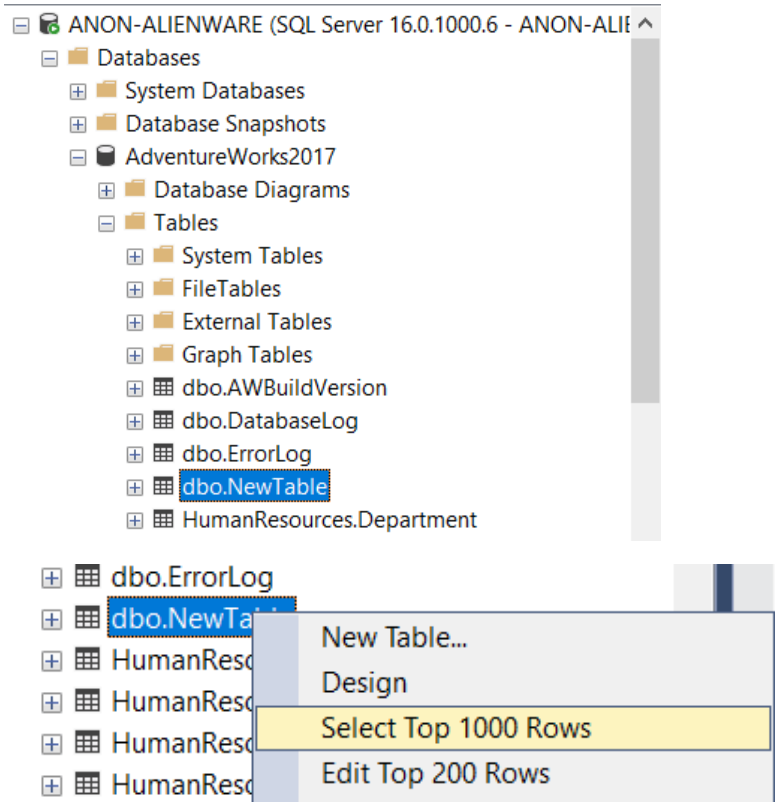


Refresh the Database

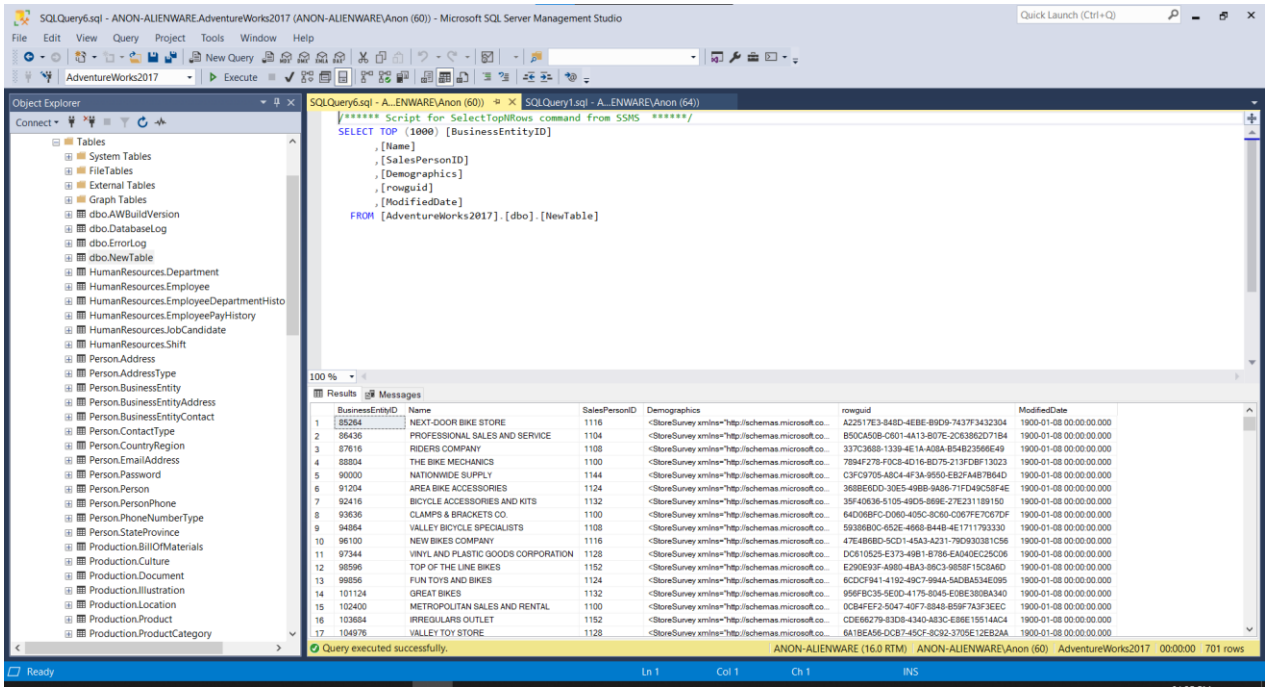


New table will be added to database

Right click on new table -> select top 100 rows



Final Output



Practical 3

AIM: Create the cube with suitable dimensions and flat table olap/holap

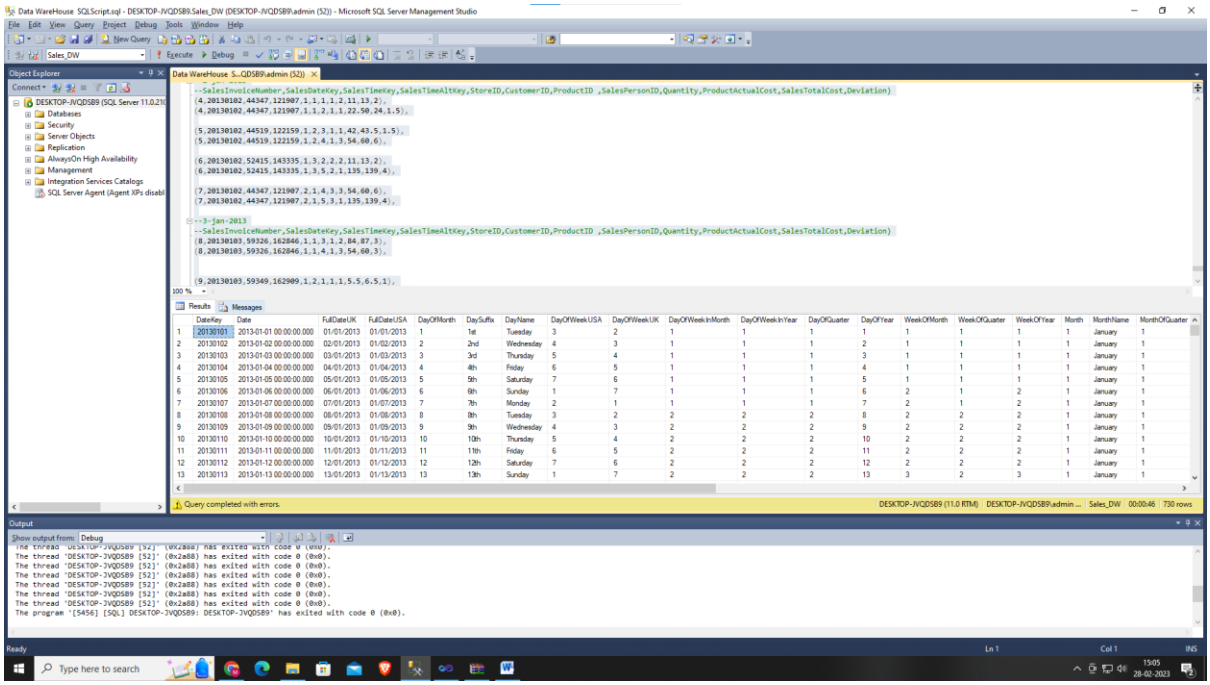
Solution:

Steps

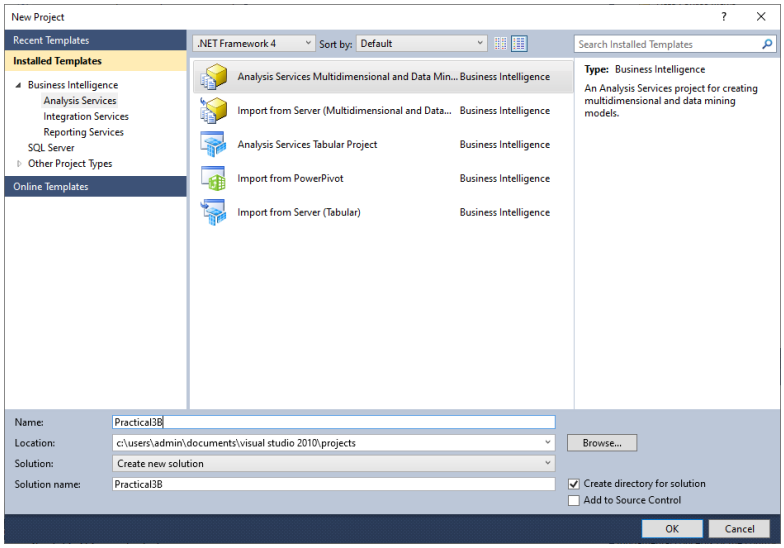
Download the script file:

https://www.codeproject.com/KB/database/652108/Data_WareHouse__SQLScript.zip

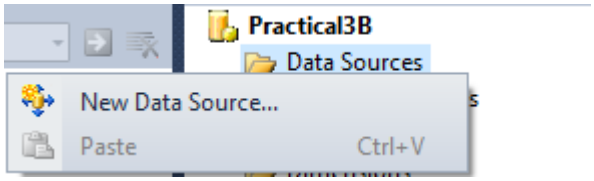
And execute it in the Sql server management tools;



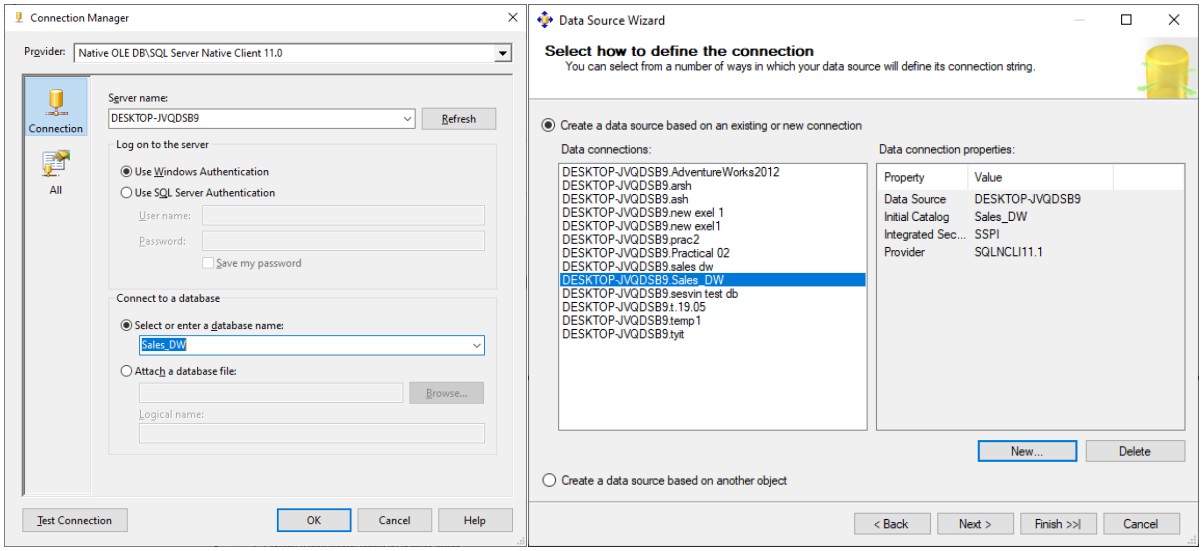
Now open, SQL Server Data Tools and create new project



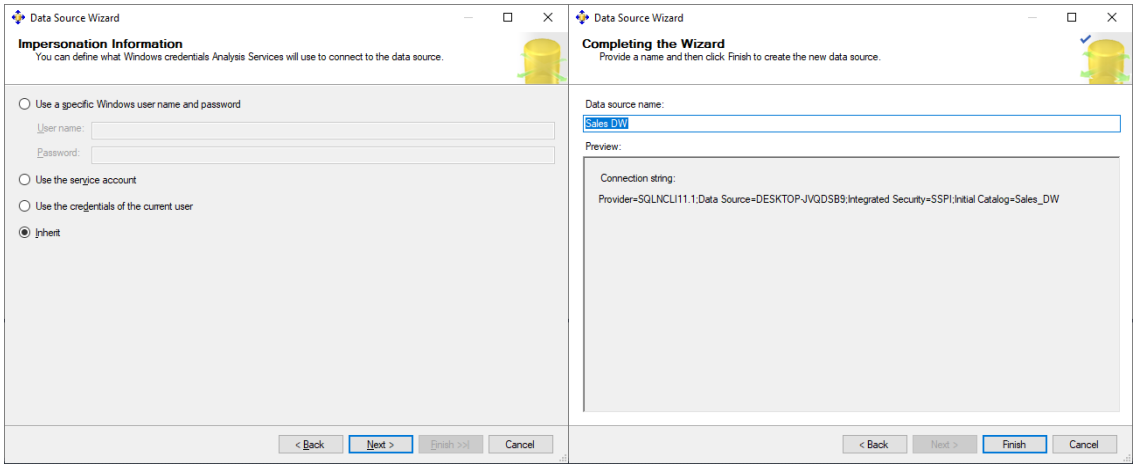
Right click on data sources and click add new Data Source



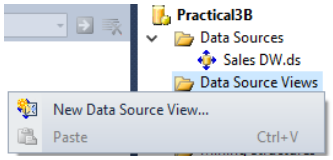
And click on New.. to add database

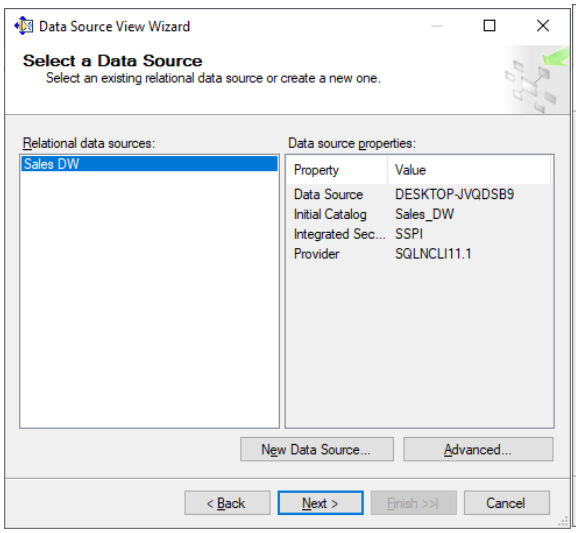


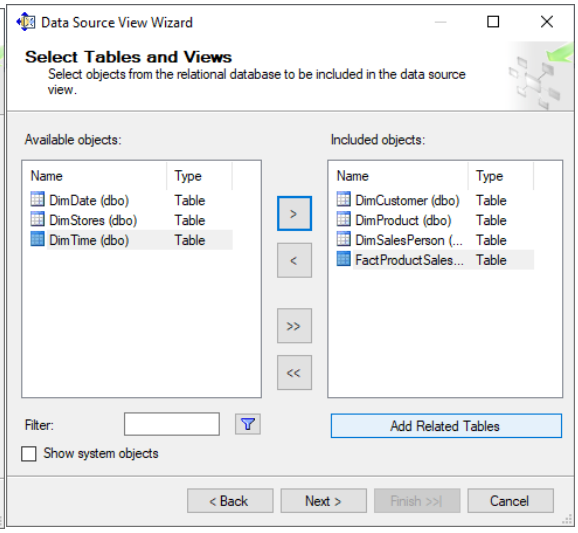
Select inherit



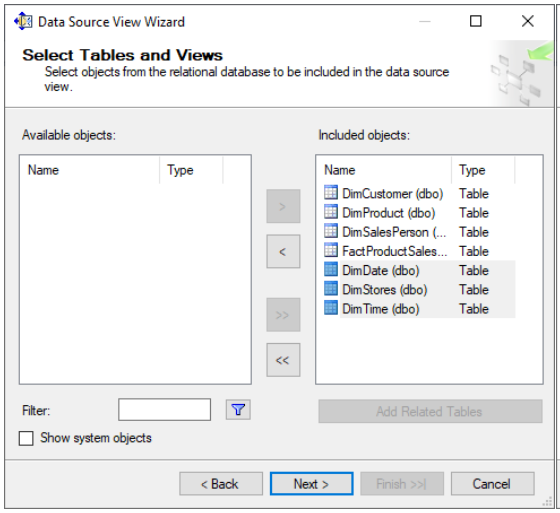
Now, Add new Data Source View

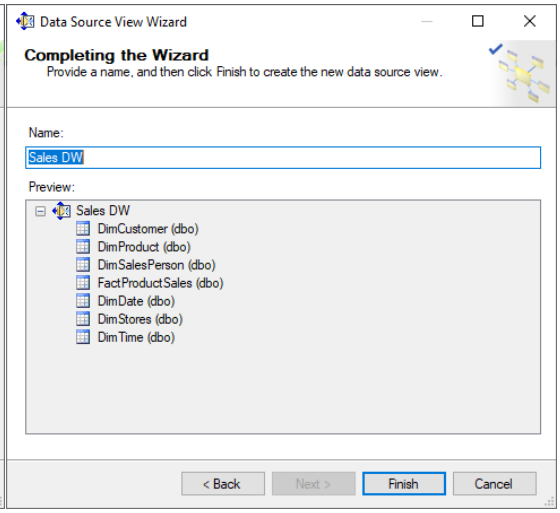




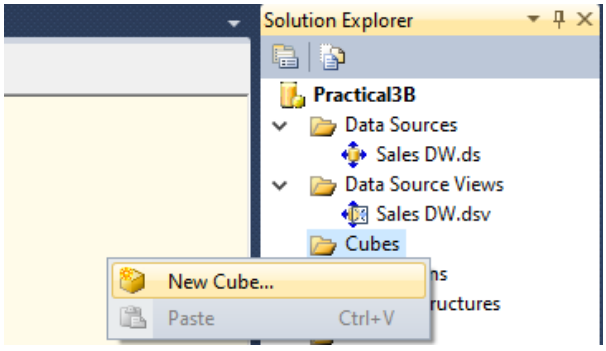


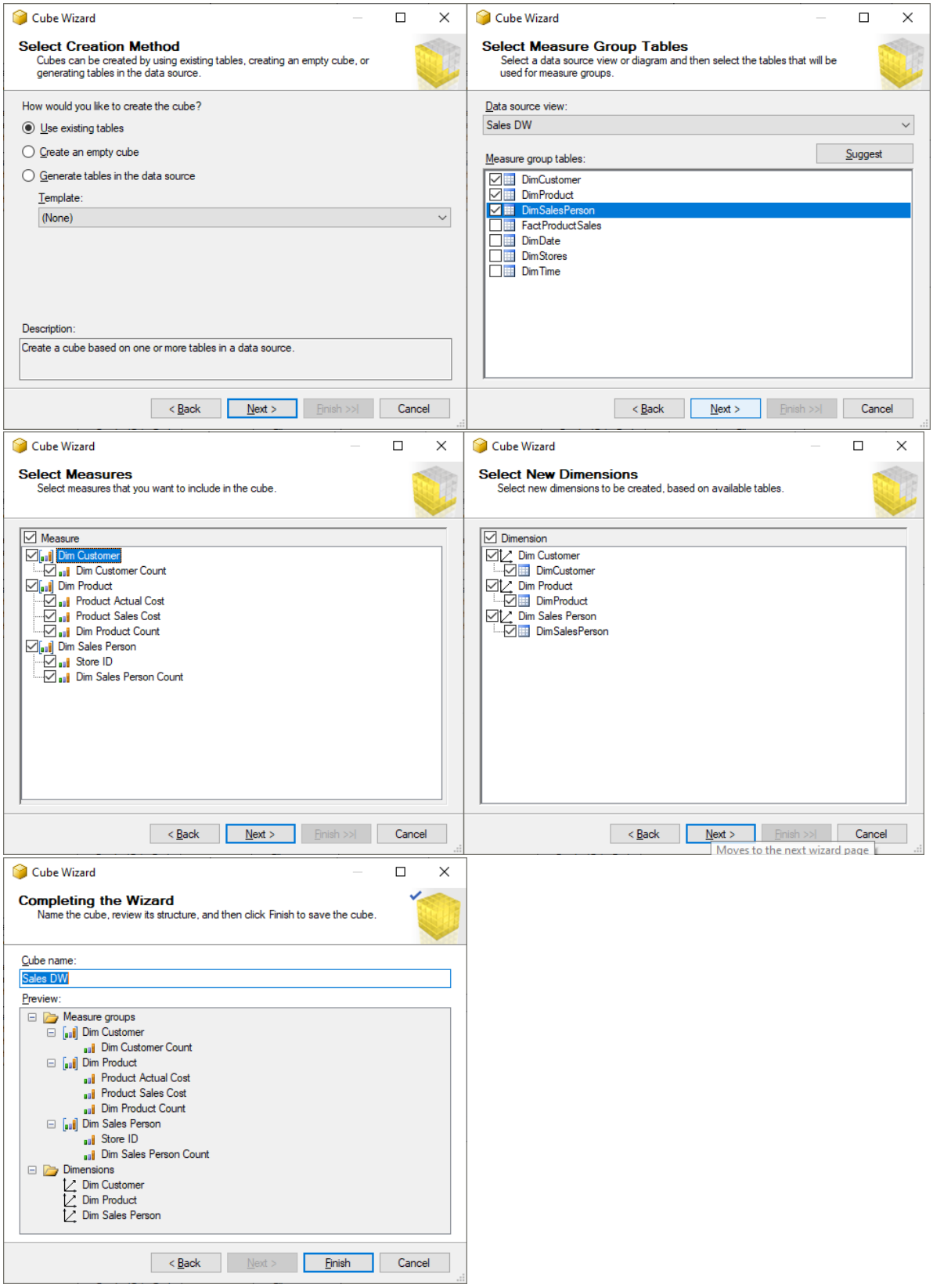
Click on Add Related Table





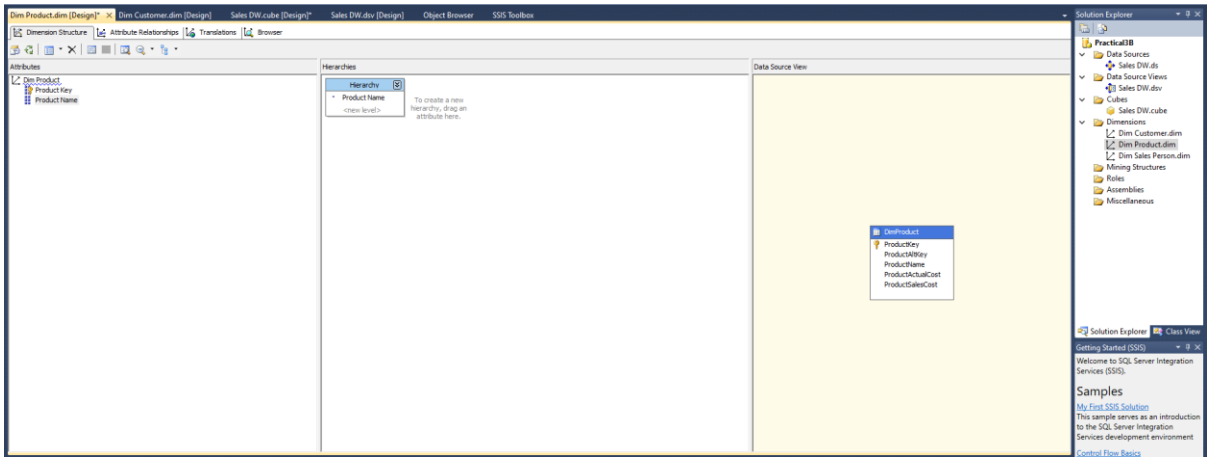
Now, Create new Cube



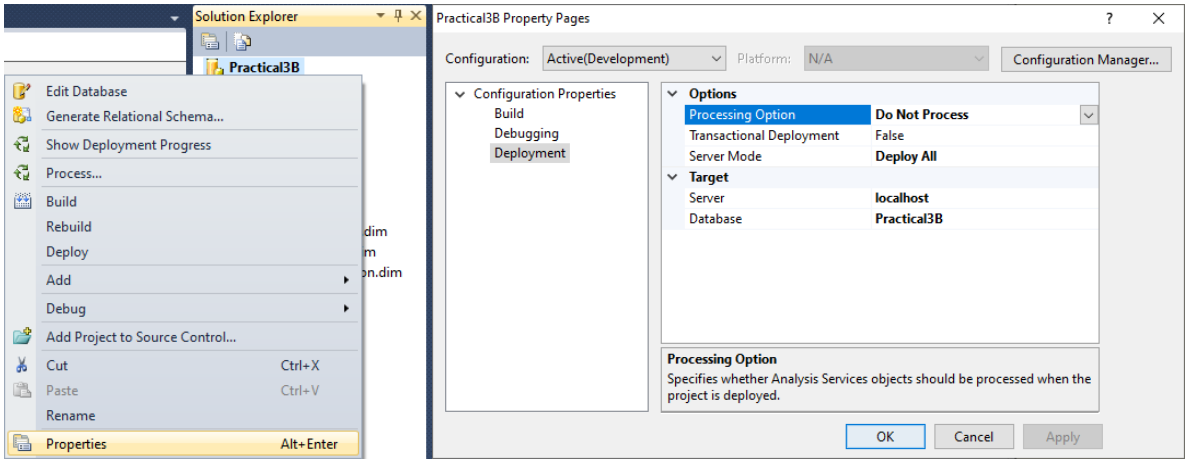

Cube Wizard
Select Creation Method
Cubes can be created by using existing tables, creating an empty cube, or generating tables in the data source.
How would you like to create the cube?
Use existing tables
Create an empty cube
Generate tables in the data source
Template:
(None)
Description:
Create a 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:
Sales DW
Suggest
Measure group tables:
DimCustomer
DimProduct
DimSalesPerson
FactProductSales
DimDate
DimStores
DimTime
< Back
Next >
Finish >>|
Cancel
Cube Wizard
Select Measures
Select measures that you want to include in the cube.
Measure
Dim Customer
Dim Customer Count
Dim Product
Product Actual Cost
Product Sales Cost
Dim Product Count
Dim Sales Person
Store ID
Dim Sales Person Count
< Back
Next >
Finish >>|
Cancel
Cube Wizard
Select New Dimensions
Select new dimensions to be created, based on available tables.
Dimension
Dim Customer
DimCustomer
Dim Product
DimProduct
Dim Sales Person
DimSalesPerson
< Back
Next >
Finish >>|
Cancel
Moves to the next wizard page
Cube Wizard
Completing the Wizard
Name the cube, review its structure, and then click Finish to save the cube.
Cube name:
Sales DW
Preview:
Measure groups
Dim Customer
Dim Customer Count
Dim Product
Product Actual Cost
Product Sales Cost
Dim Product Count
Dim Sales Person
Store ID
Dim Sales Person Count
Dimensions
Dim Customer
Dim Product
Dim Sales Person
< Back
Next >
Finish
Cancel

Now, open any dimensions

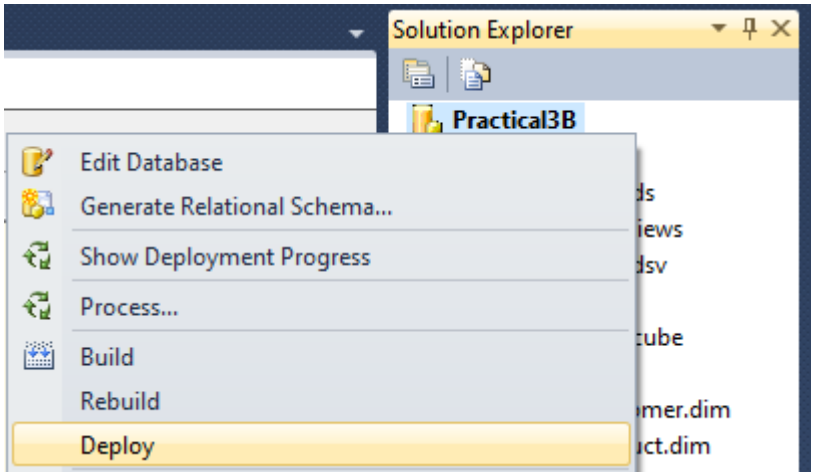
Drag any attribute from Data Source View to Attribute panel then, Drag any attribute from Attribute panel to Hierarchies Panel.



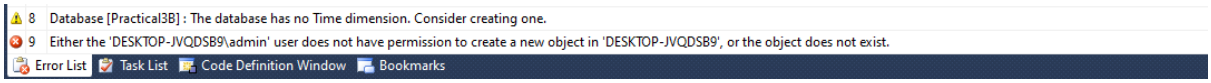
Now,



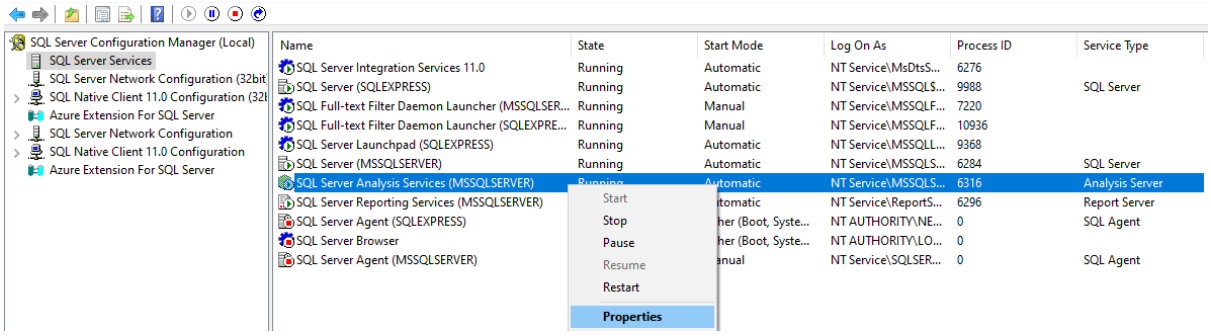
Now Deploy



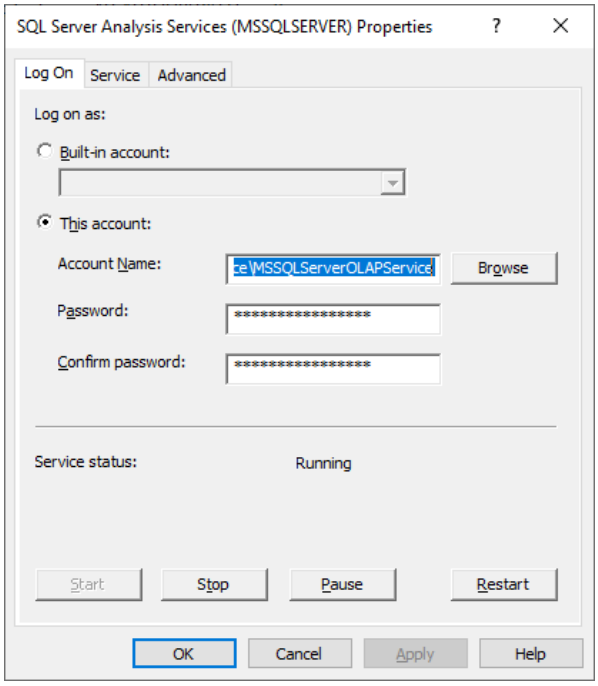
To solve this error,



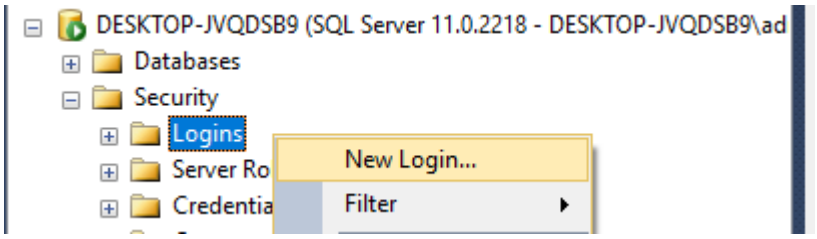
Open Sql server configuration manager:

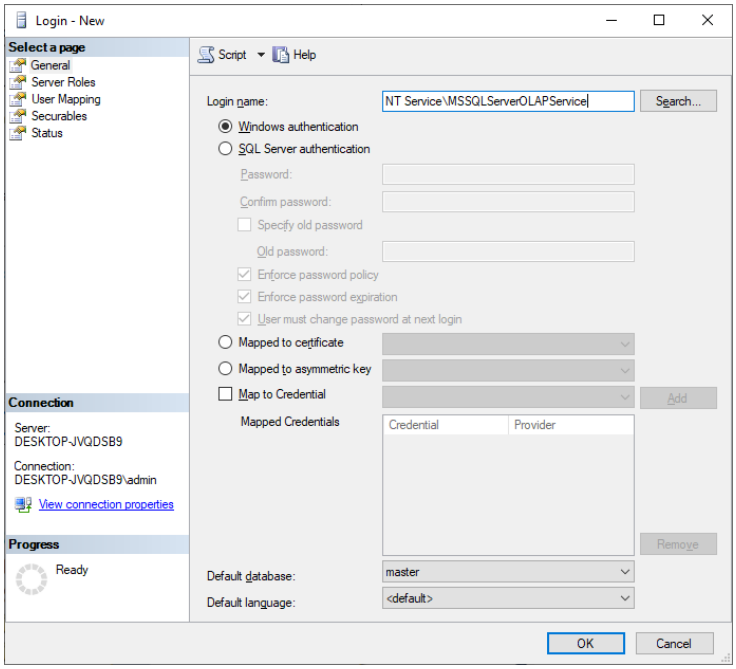


Open properties and copy account name

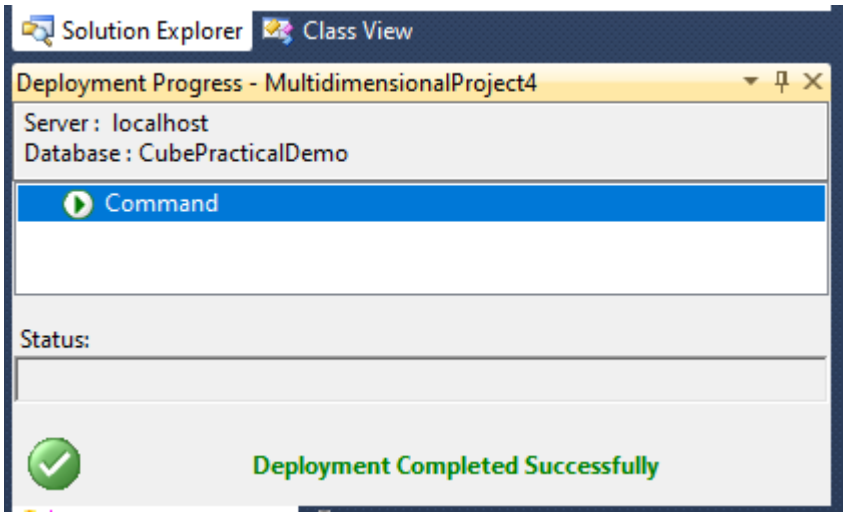


Then open SQL Server management studio

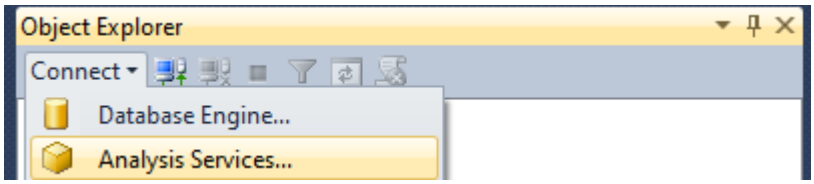


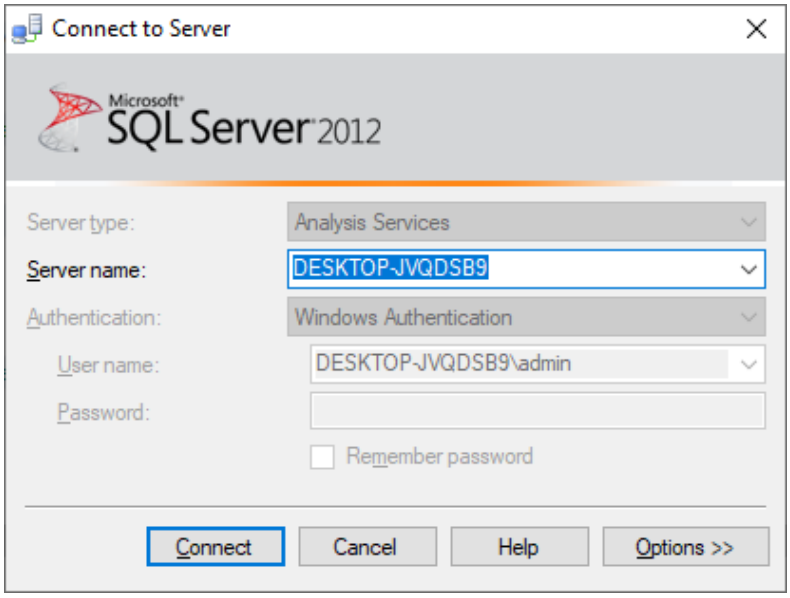


Now try again deploying the cube from SQL server data tools.

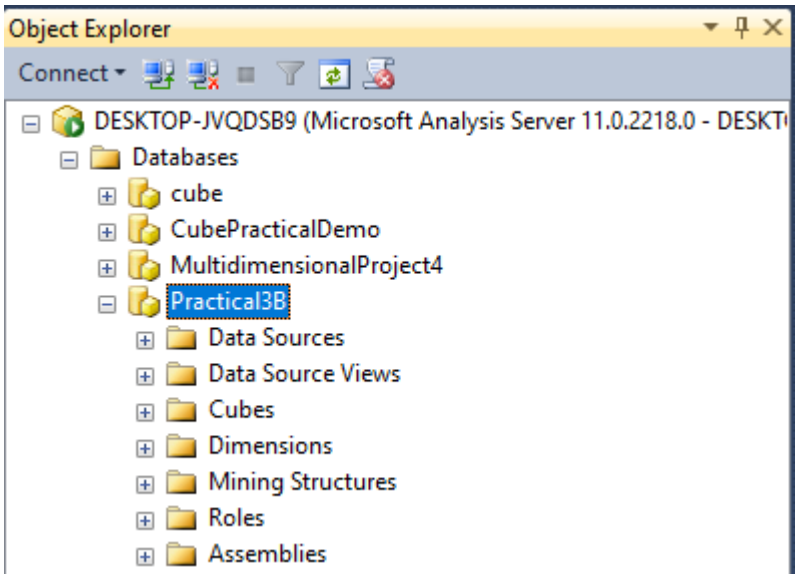


Now, again open the SQL Server Management Studio,




Connect to Server
Microsoft SQL Server 2012
Server type:
Analysis Services
Server name:
DESKTOP-JVQDSB9
Authentication:
Windows Authentication
User name:
DESKTOP-JVQDSB9\admin
Password:
Remember password
Connect
Cancel
Help
Options >>

OUTPUT

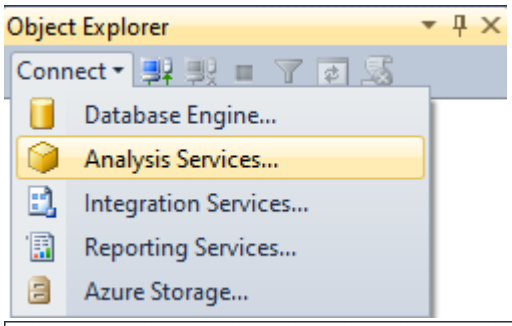

Object Explorer
Connect
DESKTOP-JVQDSB9 (Microsoft Analysis Server 11.0.2218.0 - DESKT
Databases
cube
CubePracticalDemo
MultidimensionalProject4
Practical3B
Data Sources
Data Source Views
Cubes
Dimensions
Mining Structures
Roles
Assemblies

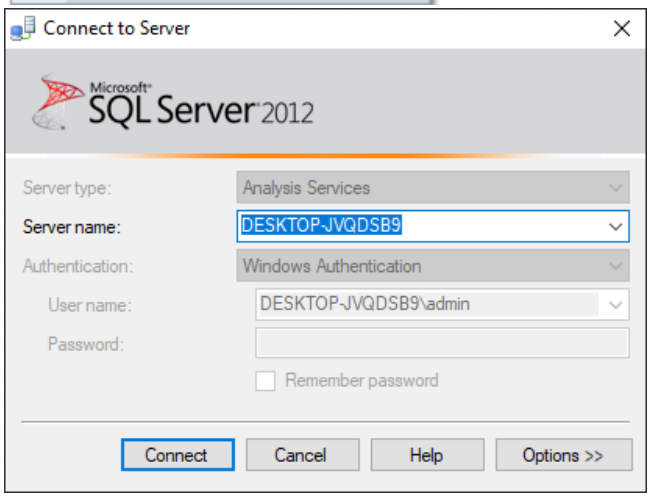
Practical 4

AIM: Execute the mdx queries to extract the data from data warehouse

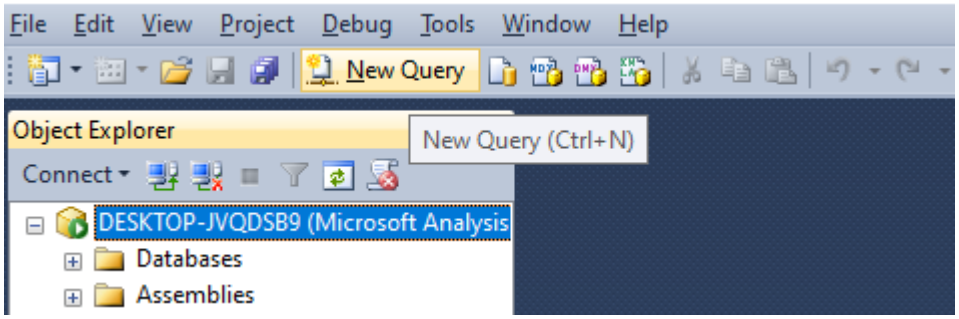
Solution:

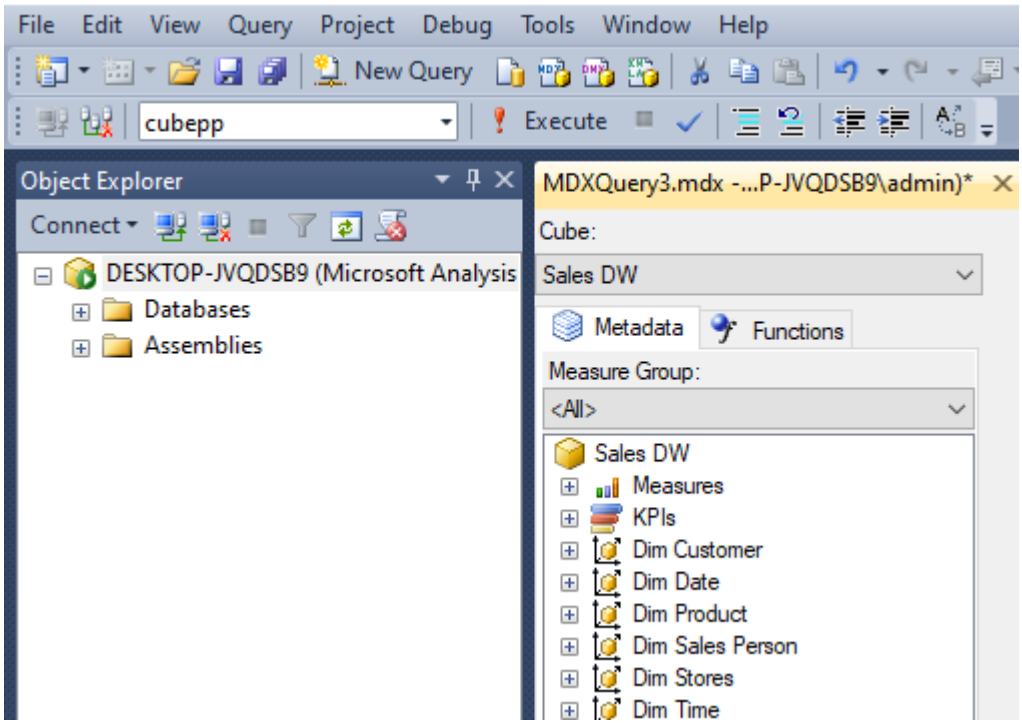
Open SQL Server Management Studio and connect to analysis services





Now, create new query,



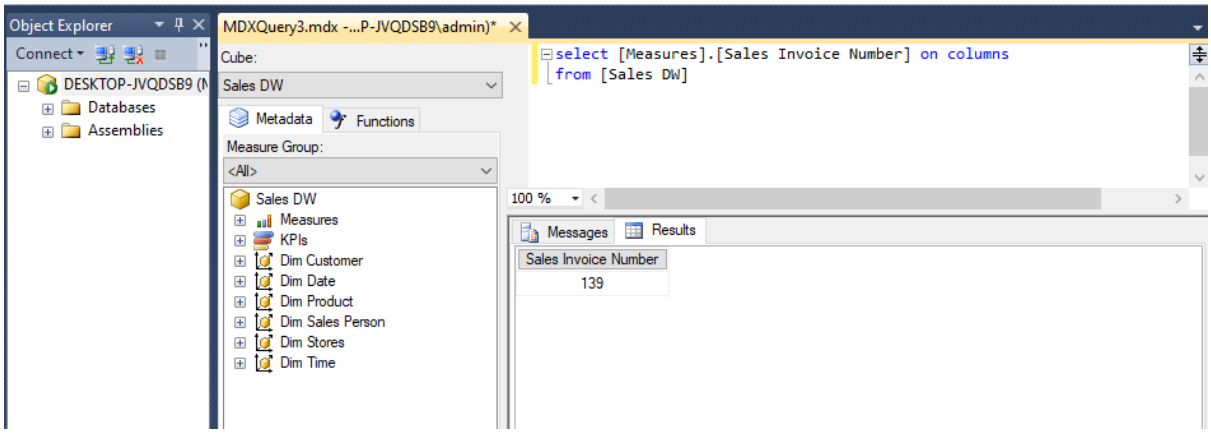


1st Query

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

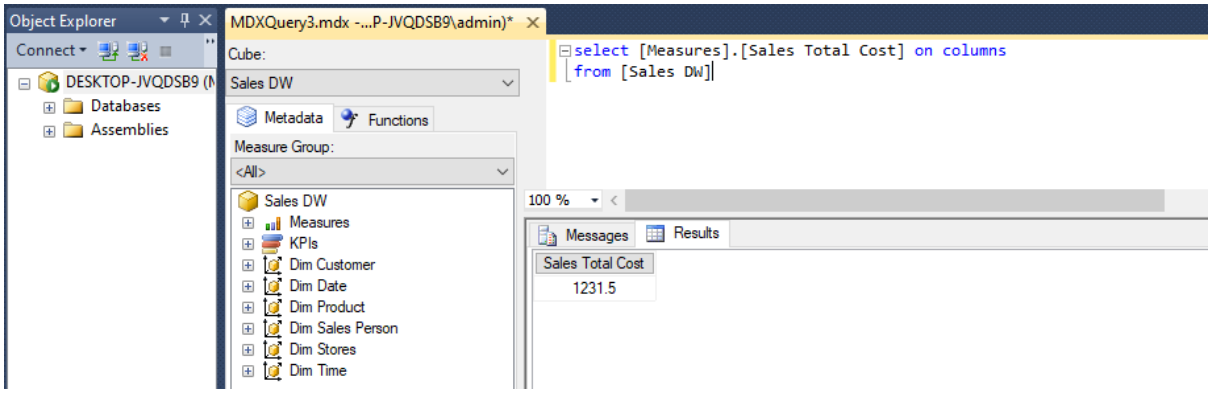
from [Sales DW]

Click on execute



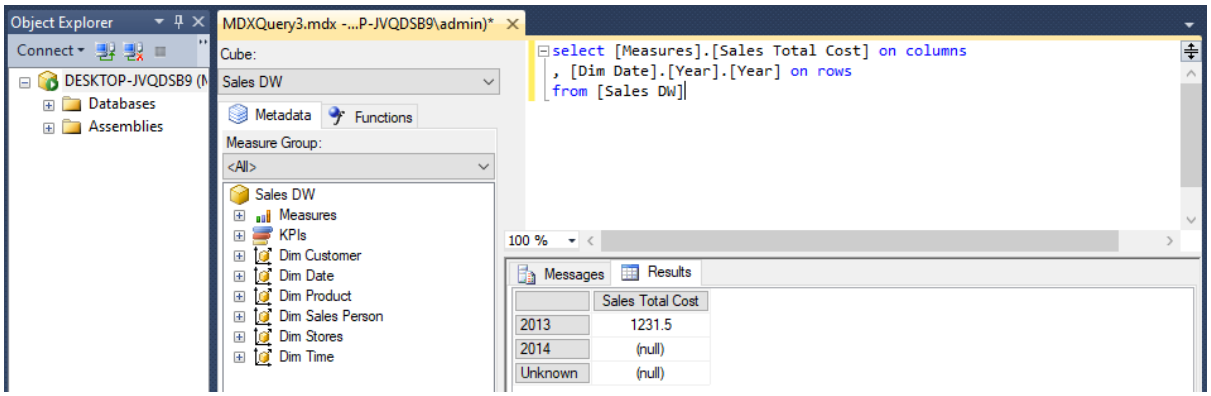
2nd Query

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



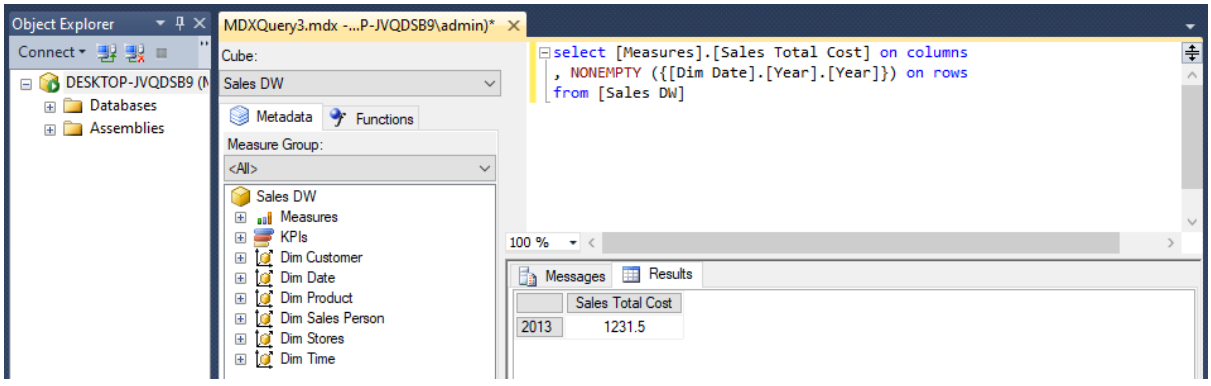
3rd Query

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



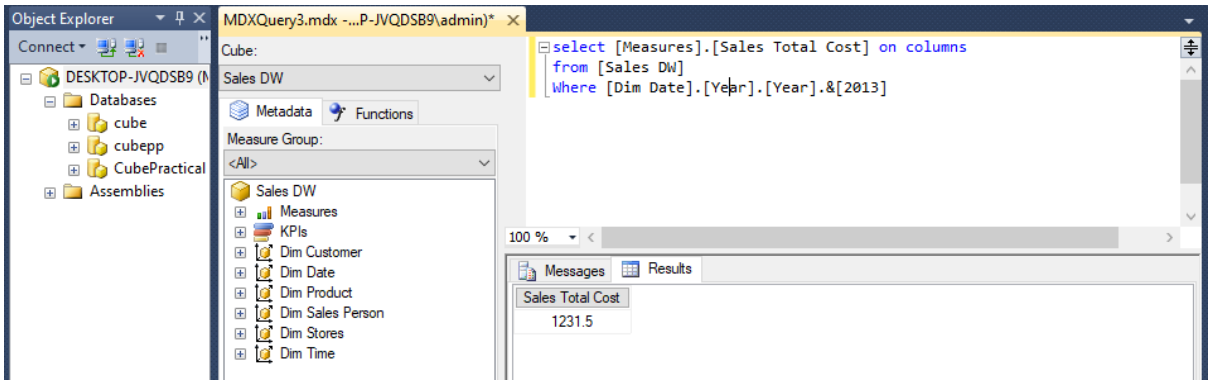
4th Query

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



5th Query

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



Practical 5:

AIM:

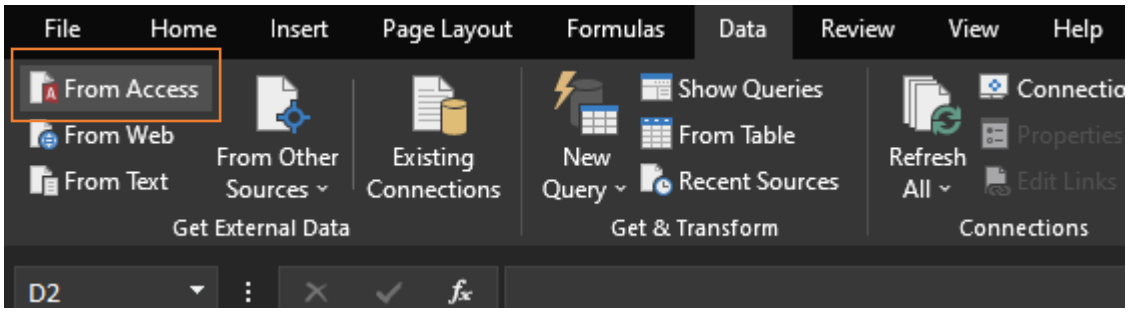
a. Import the datawarehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.
b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis

Solution:

a)

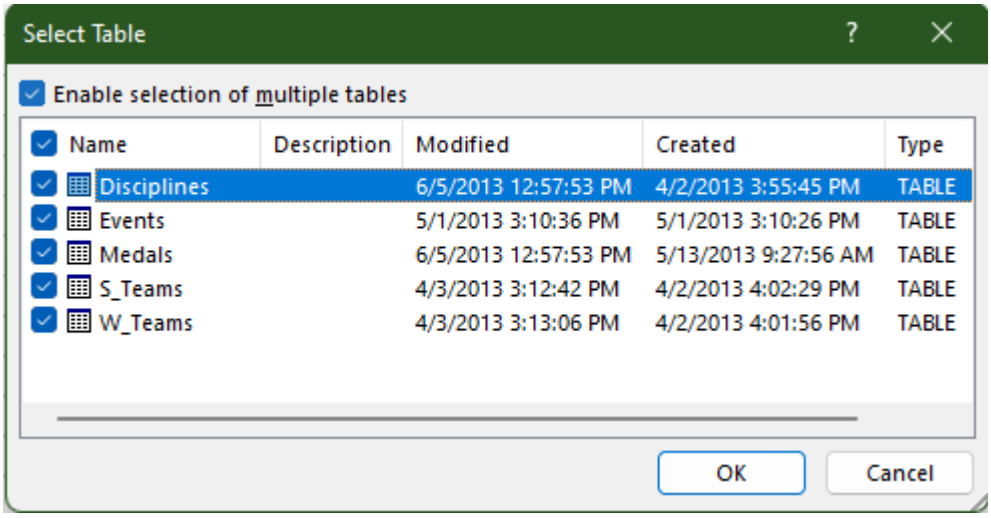
Download OlympicMedal.accdl

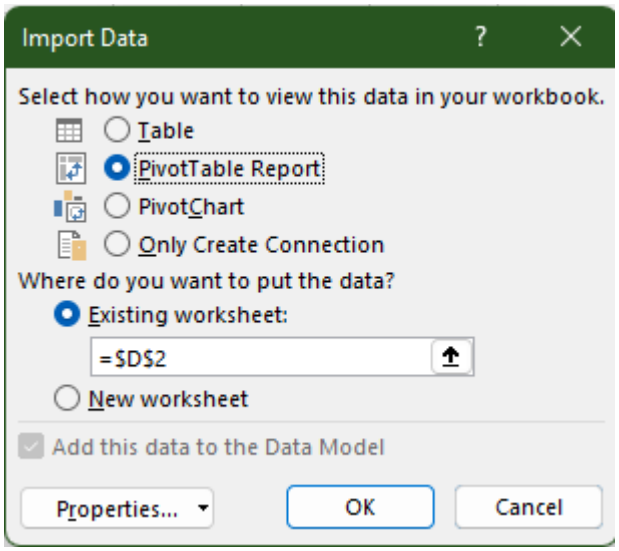
Now open Microsoft Excel, then go to Data tab and select "From Access"

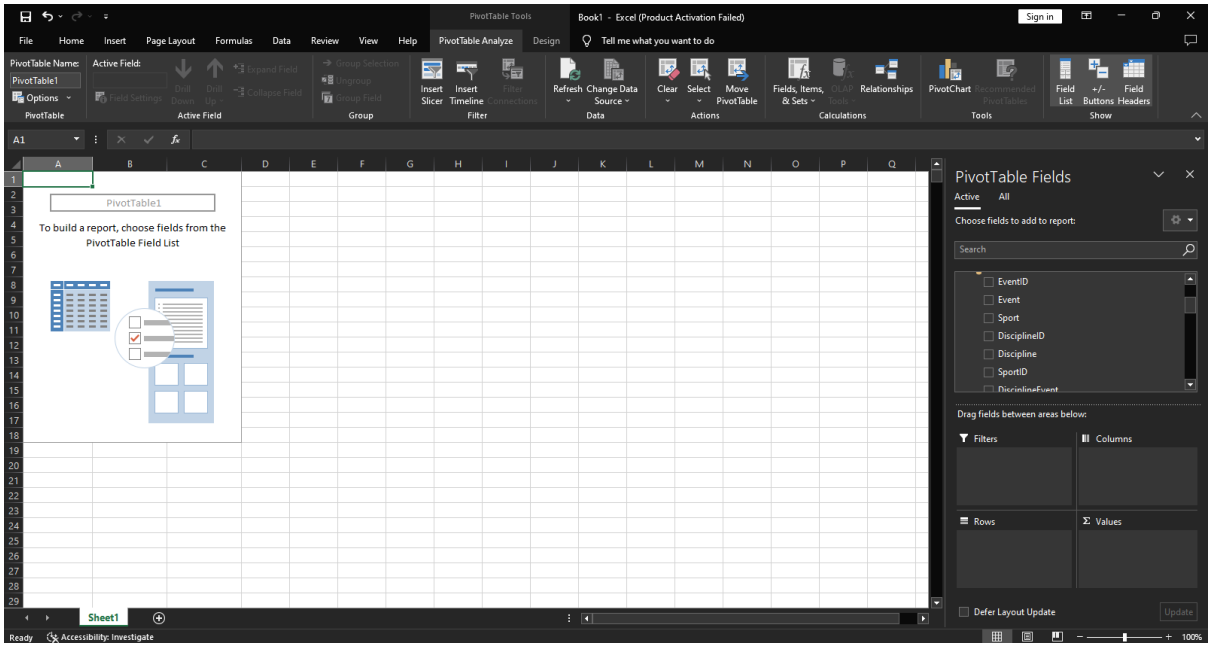


Then, select the downloaded OlympicMedal.accdl file.

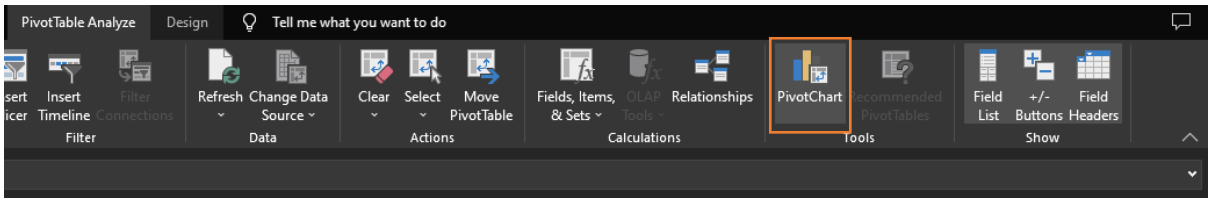
Enable selection of multiple tables and then select all the available tables.

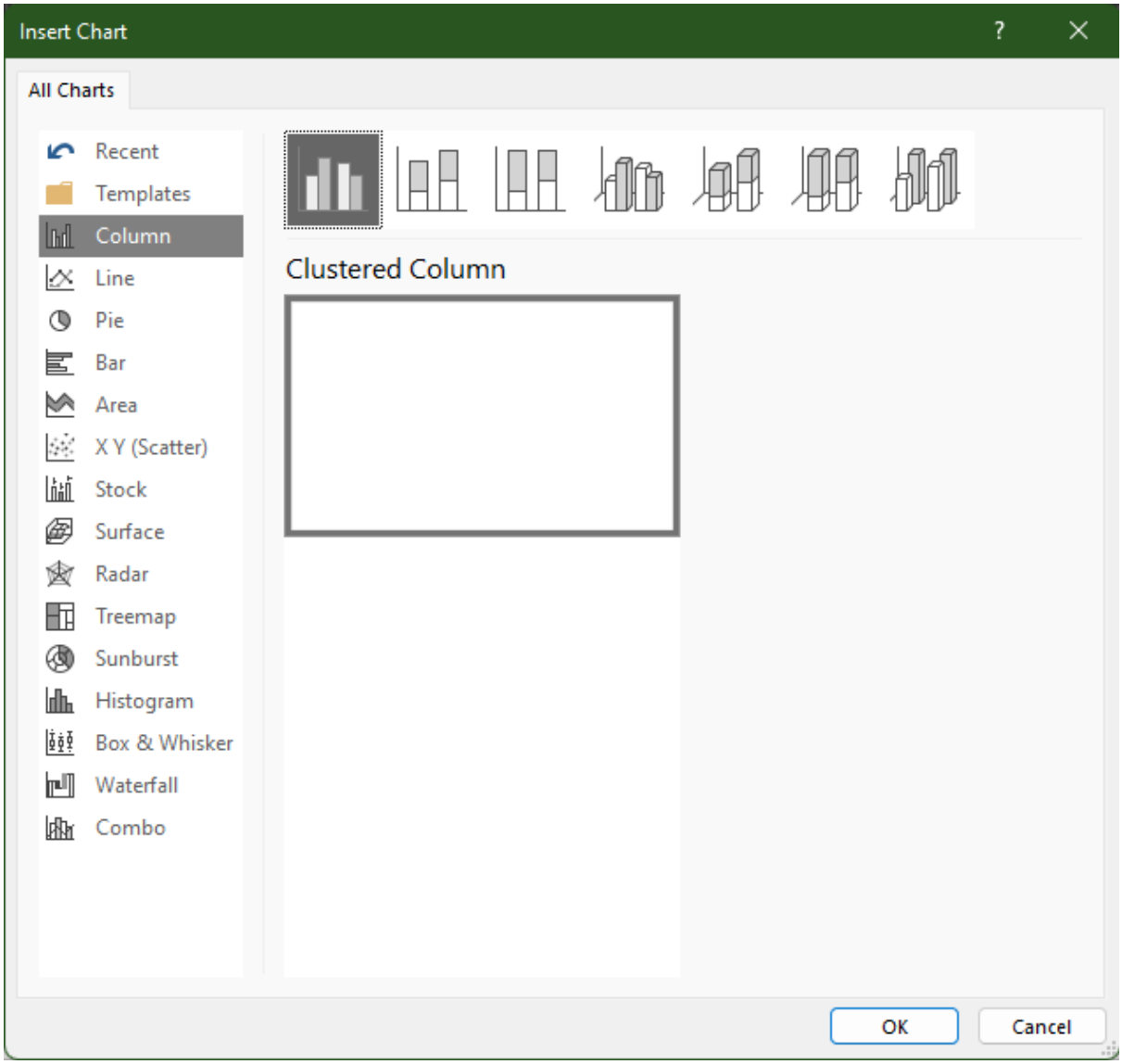


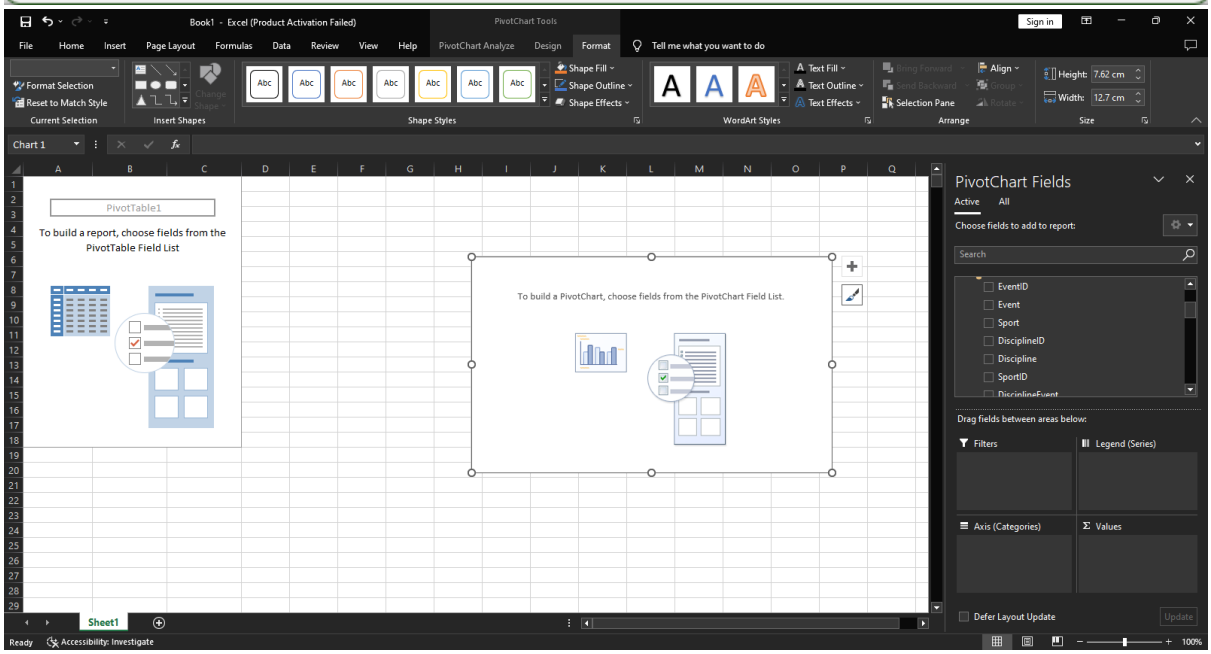




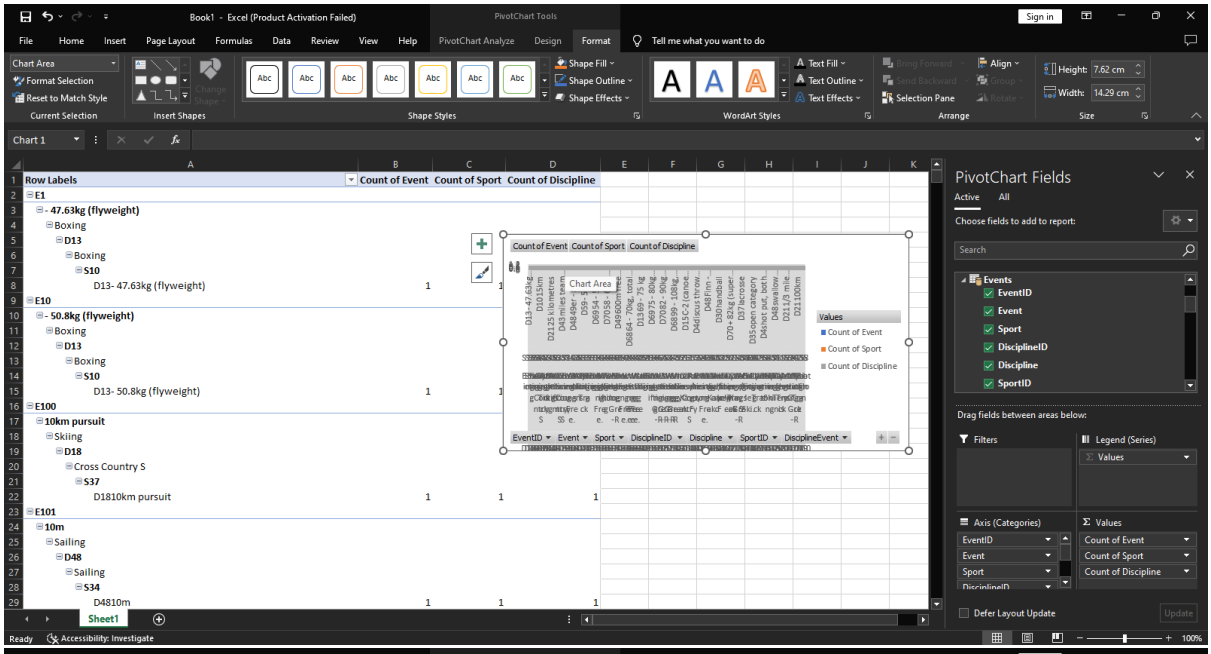
Now in pivot table analysis select PivotChart

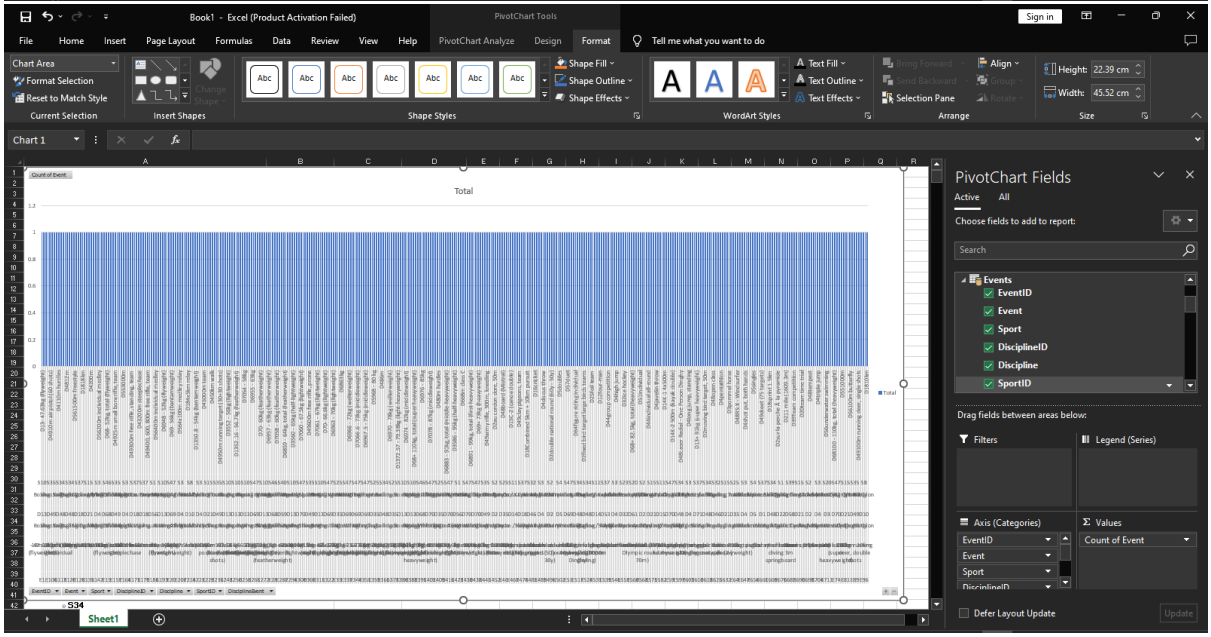



Insert Chart
All Charts
Recent
Templates
Column
Line
Pie
Bar
Area
X Y (Scatter)
Stock
Surface
Radar
Treemap
Sunburst
Histogram
Box & Whisker
Waterfall
Combo
Clustered Column
OK
Cancel


PivotChart Fields
PivotTable1
To build a report, choose fields from the PivotTable Field List
To build a PivotChart, choose fields from the PivotChart Field List.
Sheet1

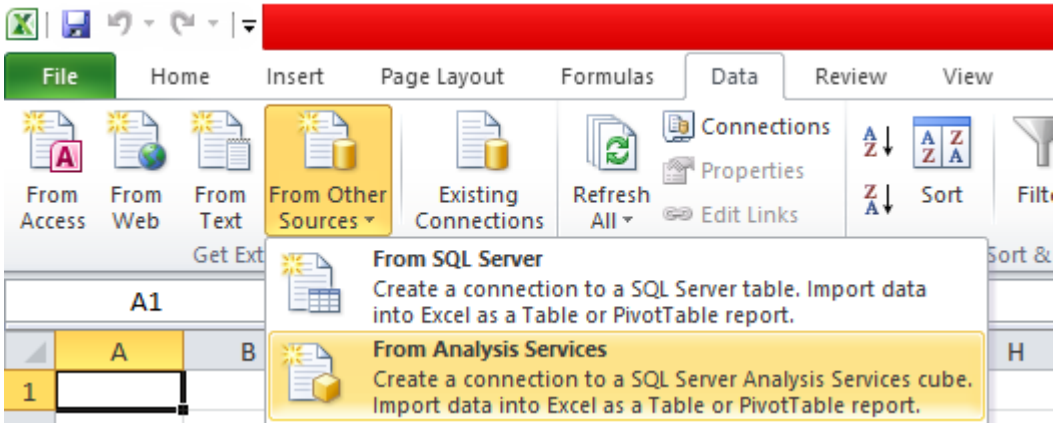
Now, choose the fields accordingly.



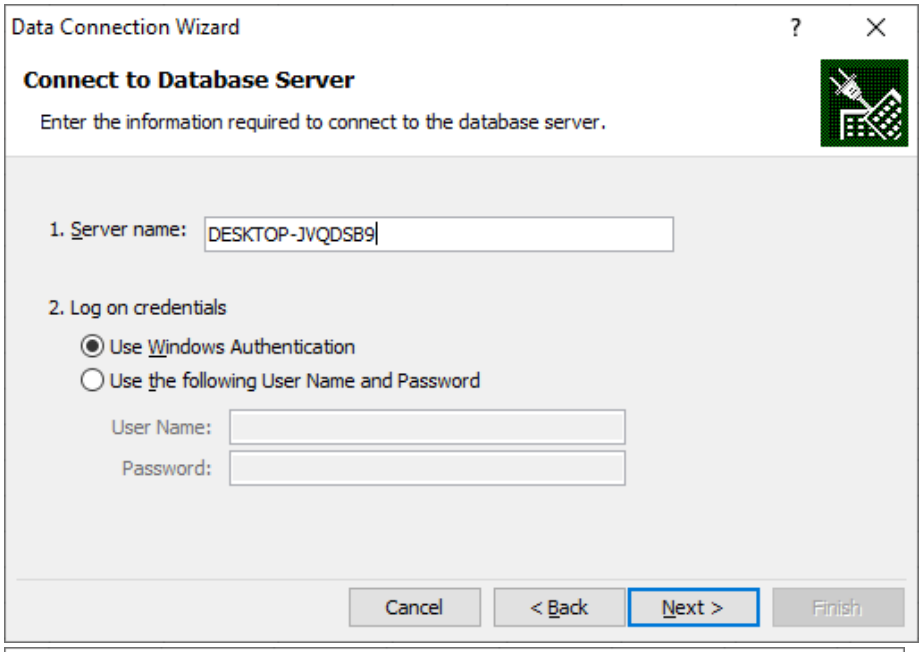


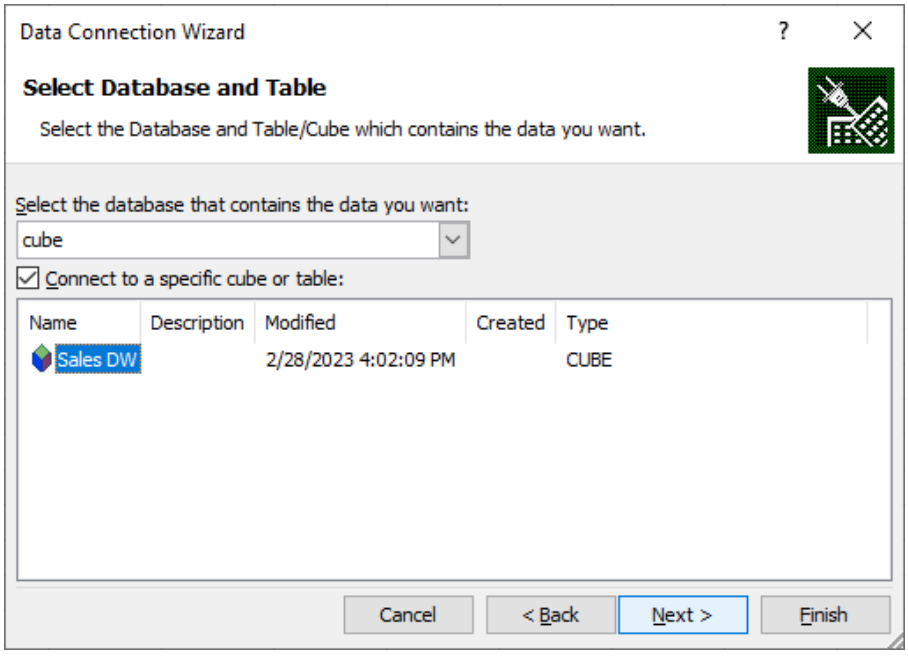
b)

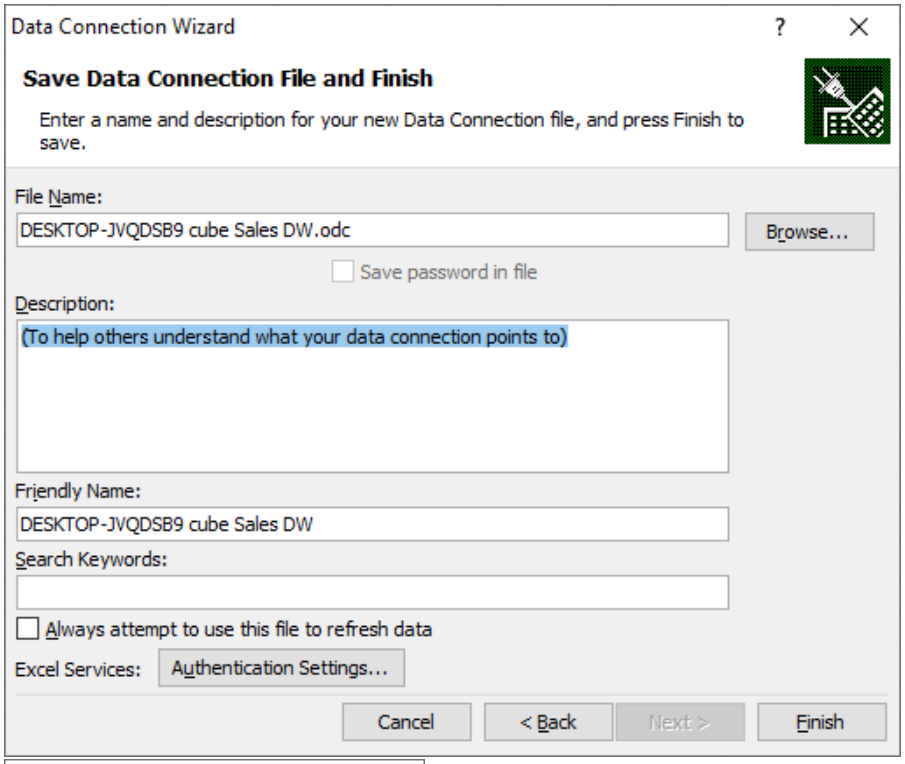
Open excel and go to data:

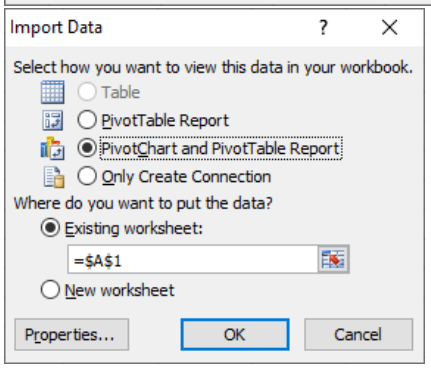


Provide server-name where the cube is present:






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:
DESKTOP-JVQDSB9 cube Sales DW.odc
Browse...
Save password in file
Description:
(To help others understand what your data connection points to)
Friendly Name:
DESKTOP-JVQDSB9 cube Sales DW
Search Keywords:
Always attempt to use this file to refresh data
Excel Services:
Authentication Settings...
Cancel
< Back
Next >
Finish


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:
=A1
New worksheet
Properties...
OK
Cancel

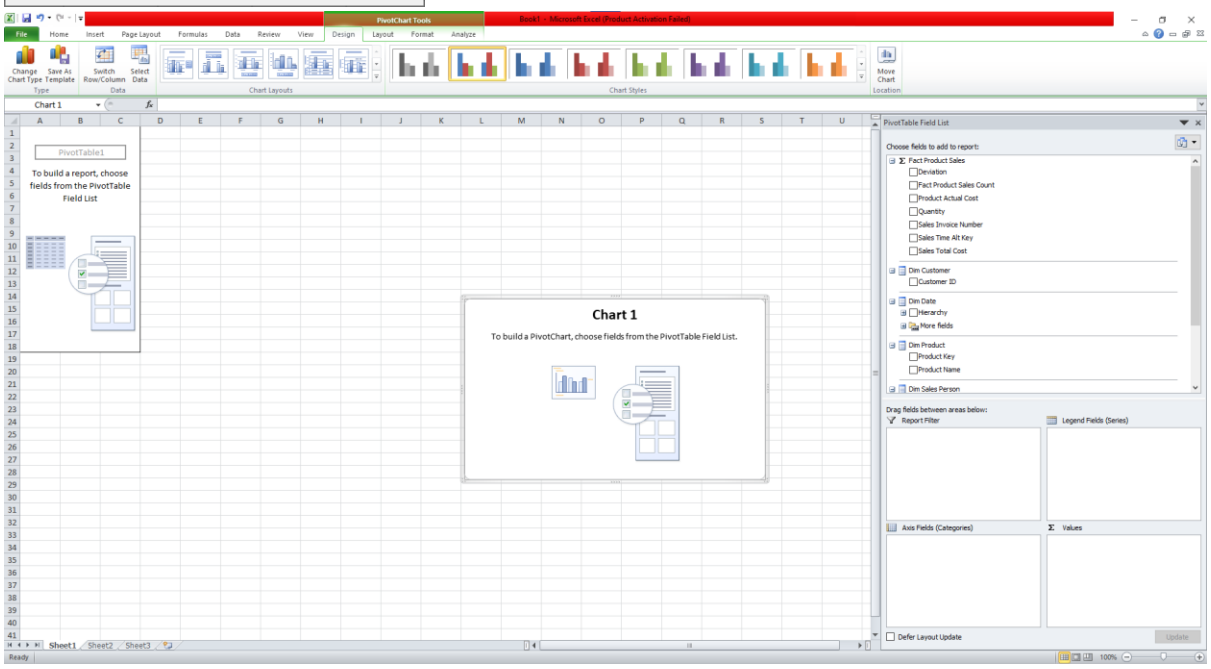
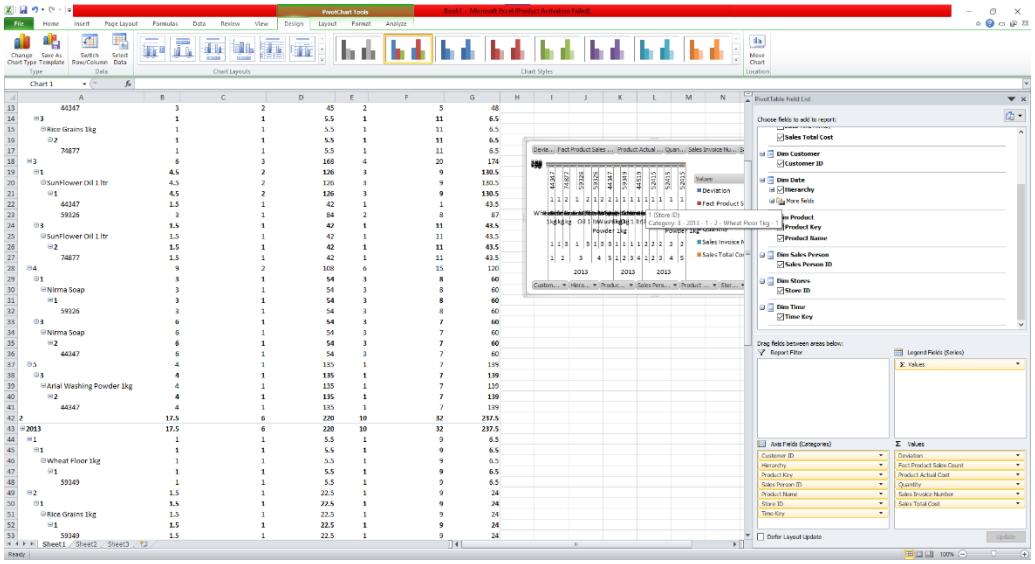
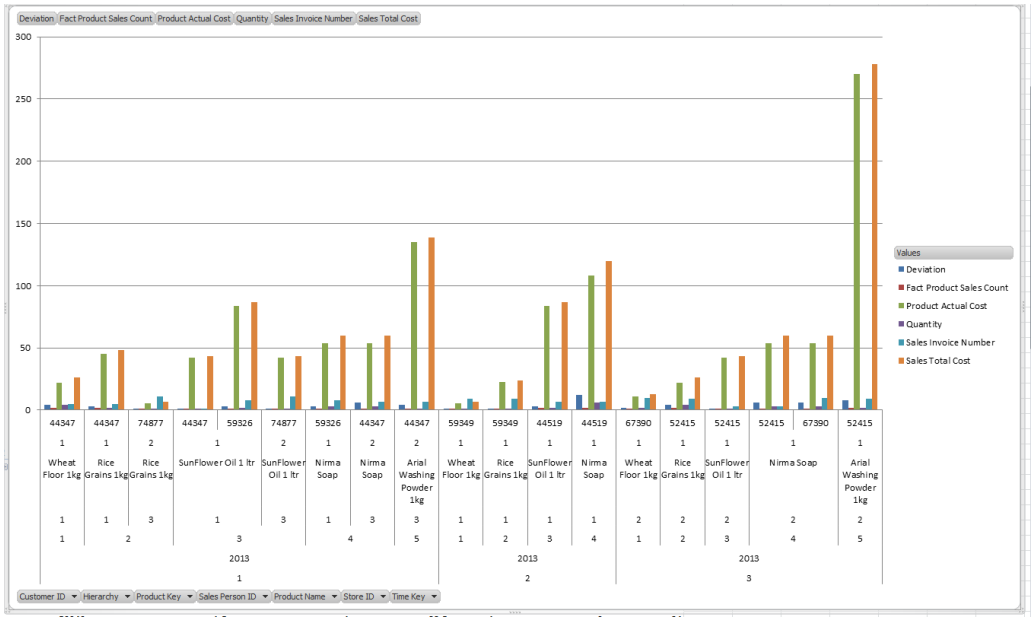


Chart 1
To build a PivotChart, choose fields from the PivotTable Field List.

OUTPUT




Deviation
Fact Product Sales Count
Product Actual Cost
Quantity
Sales Invoice Number
Sales Total Cost
300
250
200
150
100
50
0
Values
Deviation
Fact Product Sales Count
Product Actual Cost
Quantity
Sales Invoice Number
Sales Total Cost
Wheat Floor 1kg
Rice Grains 1kg
Rice Grains 1kg
SunFlower Oil 1 ltr
SunFlower Oil 1 ltr
Nirma Soap
Nirma Soap
Arial Washing Powder 1kg
Wheat Floor 1kg
Rice Grains 1kg
SunFlower Oil 1 ltr
Nirma Soap
Wheat Floor 1kg
Rice Grains 1kg
SunFlower Oil 1 ltr
Nirma Soap
Arial Washing Powder 1kg
2013
2013
2013
Customer ID
Hierarchy
Product Key
Sales Person ID
Product Name
Store ID
Time Key

Practical 7

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

Solution:

Cell calculations

C7 =B4*C4

	A	B	C	D	E
1	Grain Store				
2					
3		total weight of wheat	% sold for the highest price		
4		150	68%		
5					
6			weight of wheat	unit profit	
7		highest price	102	50	
8		lower price	48	35	
9					
10			total profit	6780	

C8 =B4-C7

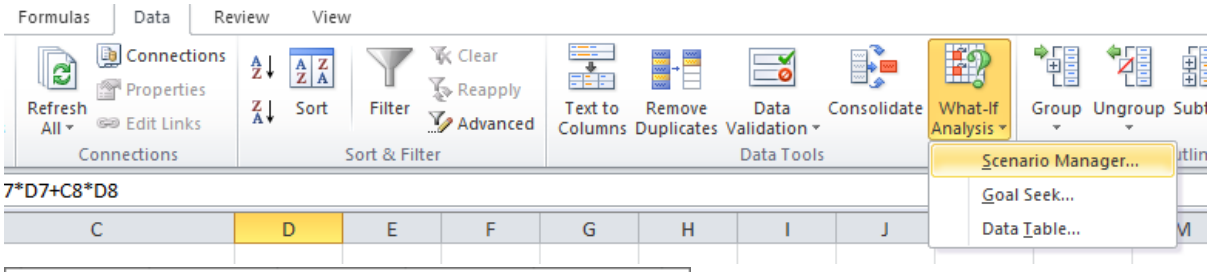
	A	B	C	D	E	F
1	Grain Store					
2						
3		total weight of wheat	% sold for the highest price			
4		150	68%			
5						
6			weight of wheat	unit profit		
7		highest price	102	50		
8		lower price	48	35		
9						
10			total profit	6780		

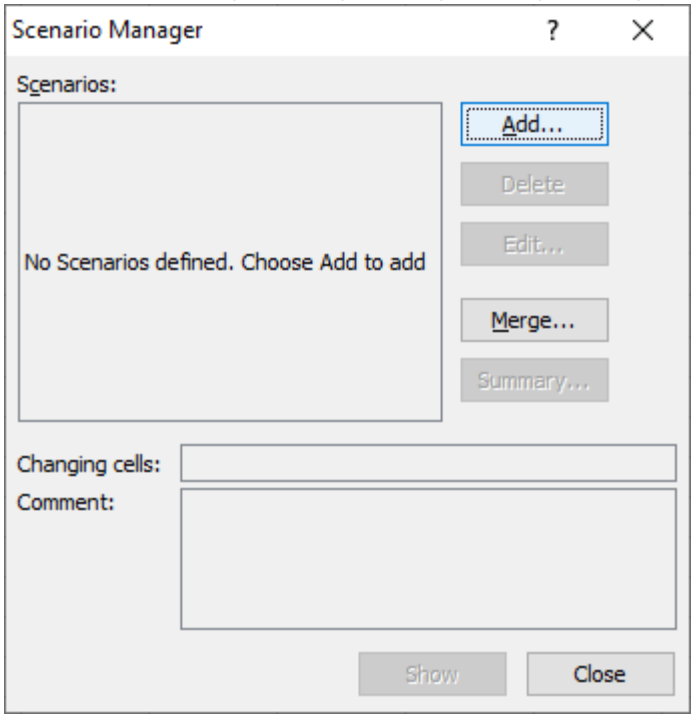
D10 =C7*D7+C8*D8

	A	B	C	D	E
1	Grain Store				
2					
3		total weight of wheat	% sold for the highest price		
4		150	68%		
5					
6			weight of wheat	unit profit	
7		highest price	102	50	
8		lower price	48	35	
9					
10			total profit	6780	

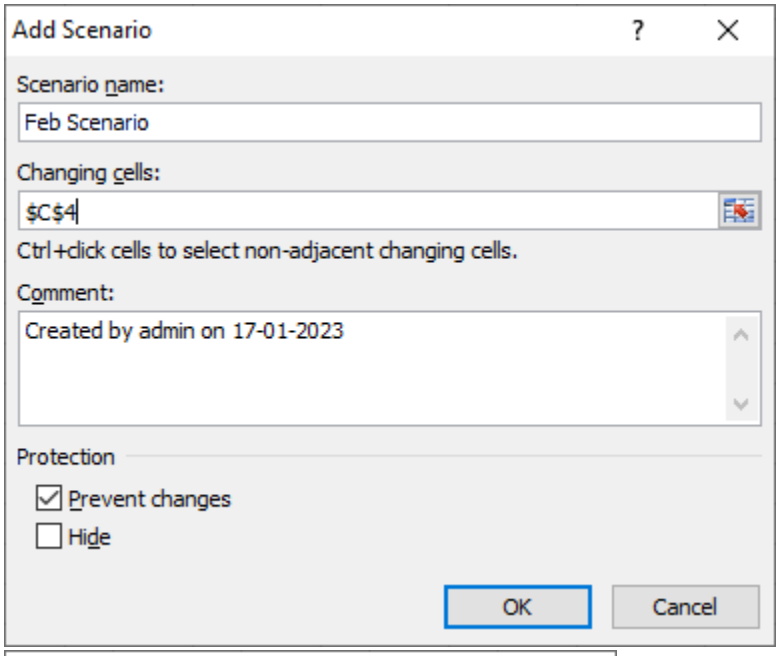
	A	B	C	D	E
1	Grain Store				
2					
3		total weight of wheat	% sold for the highest price		
4		150	68%		
5					
6			weight of wheat	unit profit	
7		highest price	102	50	
8		lower price	48	35	
9					
10			total profit	6780	
11					
12					

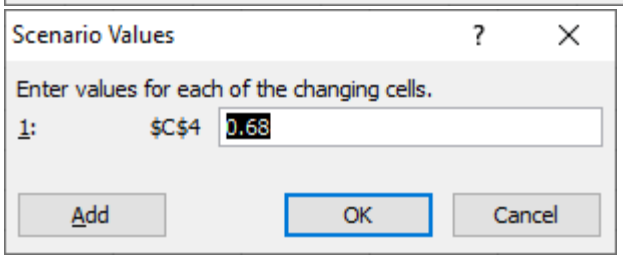
What if analysis (Scenario manager configuration)



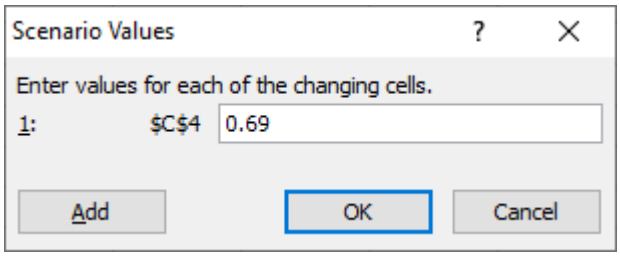


Scenario 1

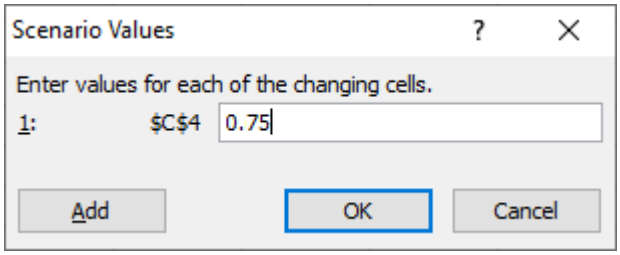




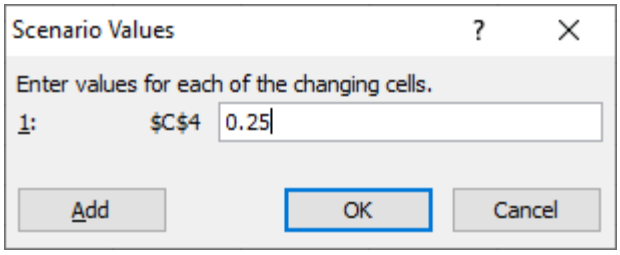
Scenario 2

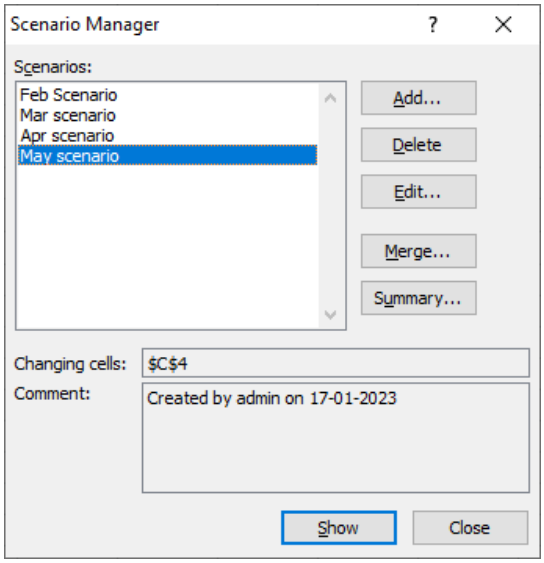


Scenario 3

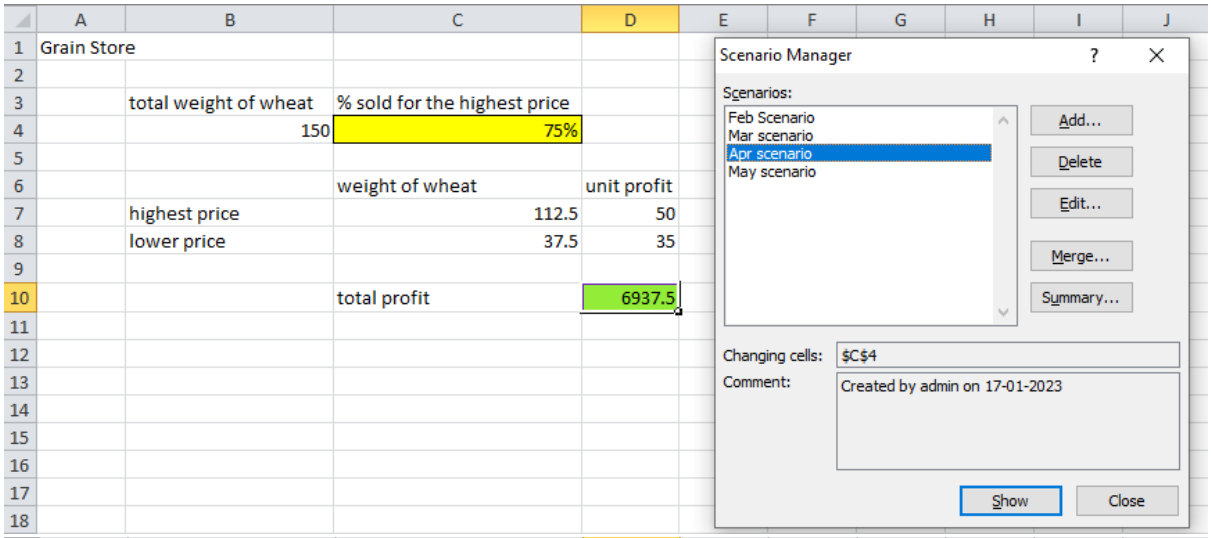


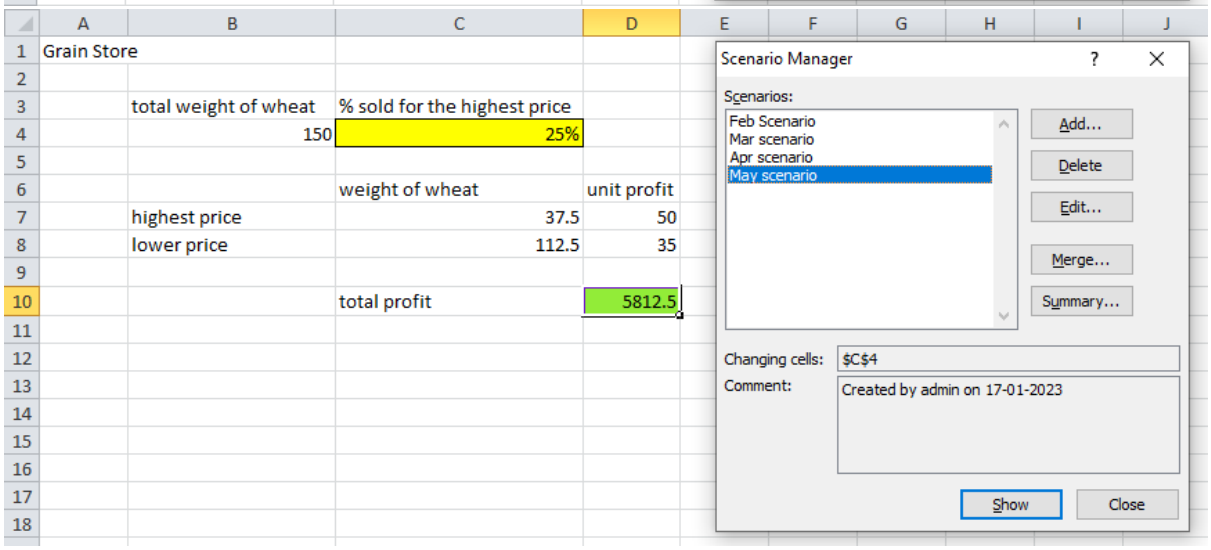
Scenario 4




Scenario Manager
Scenarios:
Feb Scenario
Mar scenario
Apr scenario
May scenario
Add...
Delete
Edit...
Merge...
Summary...
Changing cells: C4
Comment:
Created by admin on 17-01-2023
Show
Close

Click show


Grain Store
total weight of wheat
% sold for the highest price
150
75%
weight of wheat
unit profit
highest price
112.5
50
lower price
37.5
35
total profit
6937.5
Scenario Manager
Scenarios:
Feb Scenario
Mar scenario
Apr scenario
May scenario
Add...
Delete
Edit...
Merge...
Summary...
Changing cells: C4
Comment:
Created by admin on 17-01-2023
Show
Close


Grain Store
total weight of wheat
% sold for the highest price
150
25%
weight of wheat
unit profit
highest price
37.5
50
lower price
112.5
35
total profit
5812.5
Scenario Manager
Scenarios:
Feb Scenario
Mar scenario
Apr scenario
May scenario
Add...
Delete
Edit...
Merge...
Summary...
Changing cells: C4
Comment:
Created by admin on 17-01-2023
Show
Close

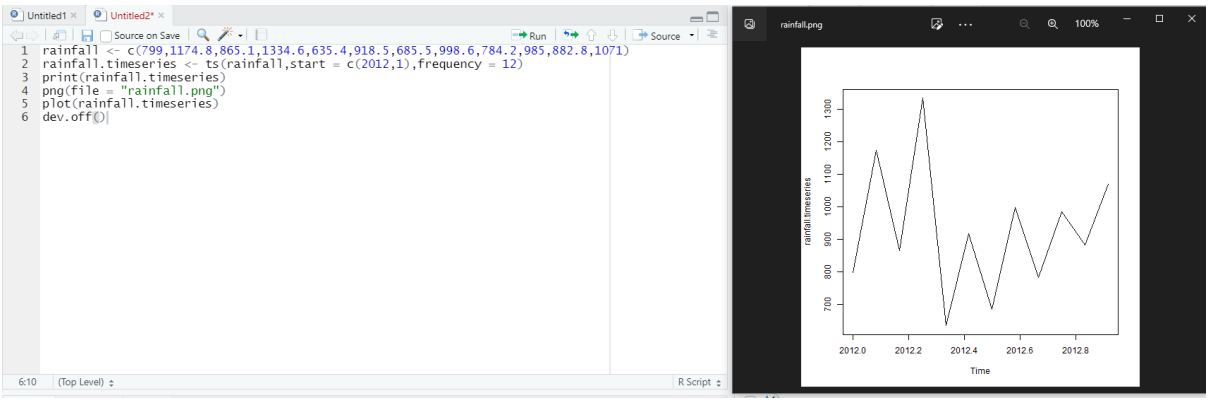
Practical 7

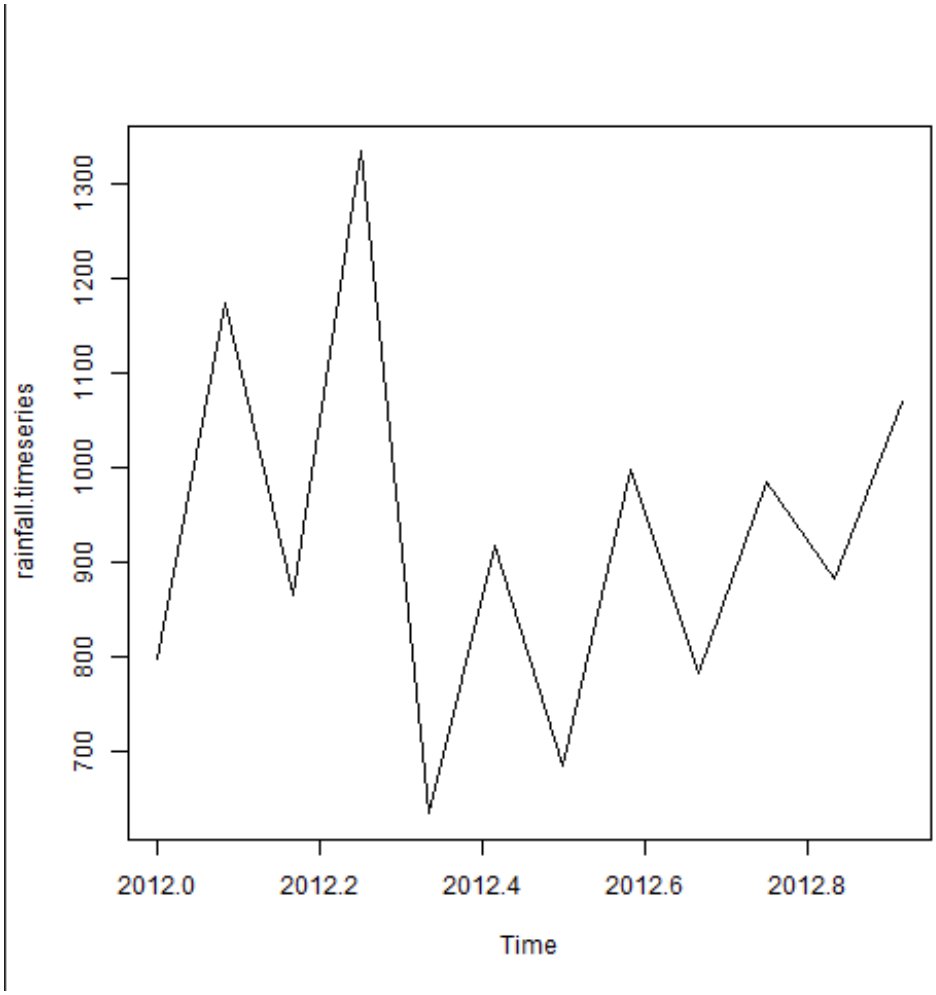
AIM: Perform the data classification using classification algorithm.

Solution:

R studio code:

```
# Get the data points in form of a R vector.
rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
# Convert it to a time series object.
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
# Print the timeseries data.
print(rainfall.timeseries)
# Give the chart file a name.
png(file = "rainfall.png")
# Plot a graph of the time series.
plot(rainfall.timeseries)
# Save the file.
dev.off()
```




rainfall.timeseries
700
800
900
1000
1100
1200
1300
2012.0
2012.2
2012.4
2012.6
2012.8
Time

Practical 8

AIM: Perform the data clustering using clustering algorithm.

Solution:

Step 1: In RStudio run below command

print(iris)

```
> print(iris)
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
11           5.4         3.7          1.5         0.2     setosa
12           4.8         3.4          1.6         0.2     setosa
13           4.8         3.0          1.4         0.1     setosa
14           4.3         3.0          1.1         0.1     setosa
15           5.8         4.0          1.2         0.2     setosa
16           5.7         4.4          1.5         0.4     setosa
17           5.4         3.9          1.3         0.4     setosa
18           5.1         3.5          1.4         0.3     setosa
19           5.7         3.8          1.7         0.3     setosa
20           5.1         3.8          1.5         0.3     setosa
21           5.4         3.4          1.7         0.2     setosa
22           5.1         3.7          1.5         0.4     setosa
23           4.6         3.6          1.0         0.2     setosa
24           5.1         3.3          1.7         0.5     setosa
25           4.8         3.4          1.9         0.2     setosa
```

Step 2: Creating copy of iris dataset and removing the Species column

newiris <- iris

newiris$Species <- NULL

print(newiris)

```
> newiris <- iris
> newiris$Species <- NULL
> print(newiris)
    Sepal.Length Sepal.Width Petal.Length Petal.Width
1            5.1         3.5          1.4         0.2
2            4.9         3.0          1.4         0.2
3            4.7         3.2          1.3         0.2
4            4.6         3.1          1.5         0.2
5            5.0         3.6          1.4         0.2
6            5.4         3.9          1.7         0.4
7            4.6         3.4          1.4         0.3
8            5.0         3.4          1.5         0.2
9            4.4         2.9          1.4         0.2
10           4.9         3.1          1.5         0.1
11           5.4         3.7          1.5         0.2
12           4.8         3.4          1.6         0.2
13           4.8         3.0          1.4         0.1
14           4.3         3.0          1.1         0.1
15           5.8         4.0          1.2         0.2
16           5.7         4.4          1.5         0.4
17           5.4         3.9          1.3         0.4
18           5.1         3.5          1.4         0.3
19           5.7         3.8          1.7         0.3
20           5.1         3.8          1.5         0.3
```

Step 3: Applying K-means to iris and storing result

print(iris)

newiris <- iris

newiris$Species <- NULL

print(newiris)

(kc <- kmeans(newiris,4))

```
print(iris)
newiris <- iris
newiris$Species <- NULL
print(newiris)
(kc <- kmeans(newiris,4))
```

```
> (kc <- kmeans(newiris,4))
K-means clustering with 4 clusters of sizes 27, 28, 50, 45

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     7.014815    3.096296     5.918519    2.155556
2     5.532143    2.635714     3.960714    1.228571
3     5.006000    3.428000     1.462000    0.246000
4     6.264444    2.884444     4.886667    1.666667

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 2 4 2 4 2 4 2 2 2 2 4 2 4 2 2 4 2 4
 [71] 4 2 4 4 4 4 4 4 4 2 2 2 2 4 2 4 4 4 2 2 2 4 2 2 2 2 2 2 4 2 2 1 4 1 4 1
[106] 1 2 1 1 1 4 4 1 4 4 4 4 1 1 4 1 4 1 4 1 1 4 4 1 1 1 1 1 4 4 1 1 4 4 1
[141] 1 1 4 1 1 1 4 4 4 4

Within cluster sum of squares by cluster:
[1] 15.351111  9.749286 15.151000 17.014222
 (between_SS / total_SS =  91.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
>
```

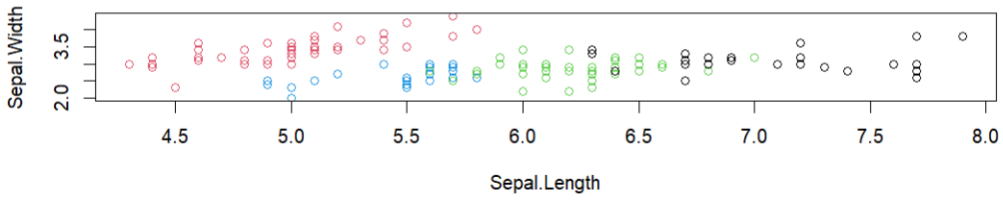
Step 4: Compare the species label with clustring result

table(iris$Species, kc$cluster)

```
> table(iris$Species, kc$cluster)

            1  2  3  4
  setosa    0 50  0  0
  versicolor  0  0 23 27
  virginica  27  0 22  1
```

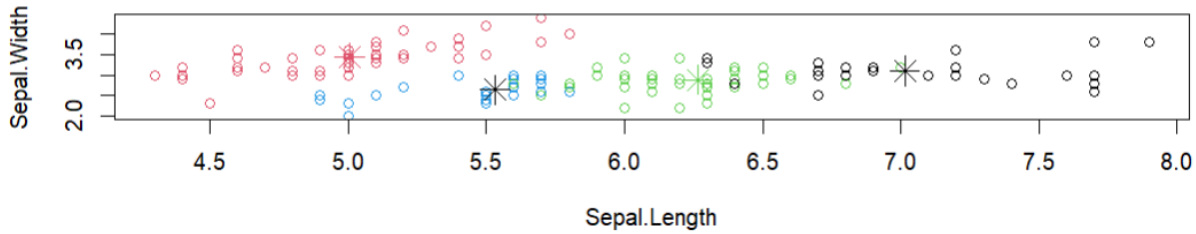
Step 5: Plot the cluster and their centers

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



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

points(kc$centers[,c("Sepal.Length", "Sepal.Width")],col=1:3,pch=8,cex=2)



Complete code

```
print(iris)
newiris <- iris
newiris$Species <- NULL
print(newiris)
(kc <- kmeans(newiris,4))
table(iris$Species, kc$cluster)
plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)
points(kc$centers[,c("Sepal.Length", "Sepal.Width")],col=1:3,pch=8,cex=2)
```

Practical 9

AIM: Perform the Linear regression on the given data warehouse data.

Solution:

Step 1: Setting up data

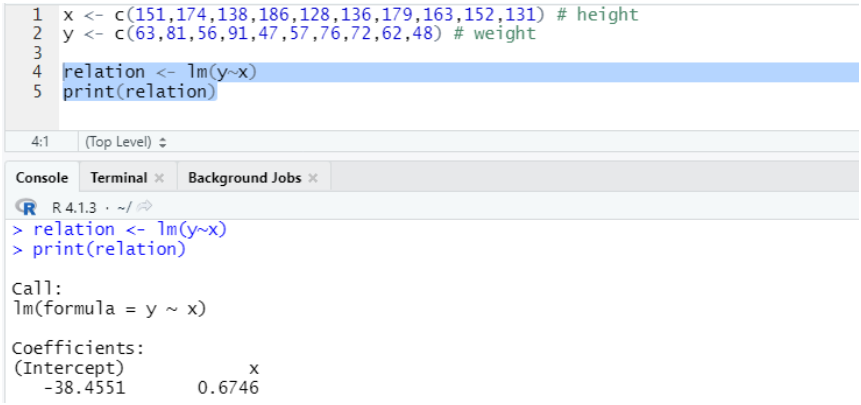
x <- c(151,174,138,186,128,136,179,163,152,131) # height

y <- c(63,81,56,91,47,57,76,72,62,48) # weight

Step 2: printing relation

relation <- lm(y~x)

print(relation)



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

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

```
Console  Terminal  Background Jobs
R 4.1.3 · ~/
> relation <- lm(y~x)
> print(relation)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551       0.6746
```

Step 3: printing summary of the relation

print(summary(relation))

```
> print(summary(relation))

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002 -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

Step 4: Predicting the weight of new person

a <- data.frame(x = 170)

result <- predict(relation,a)

print(result)

```
7  a <- data.frame(x = 170)
8  result <- predict(relation,a)
9  print(result)
```

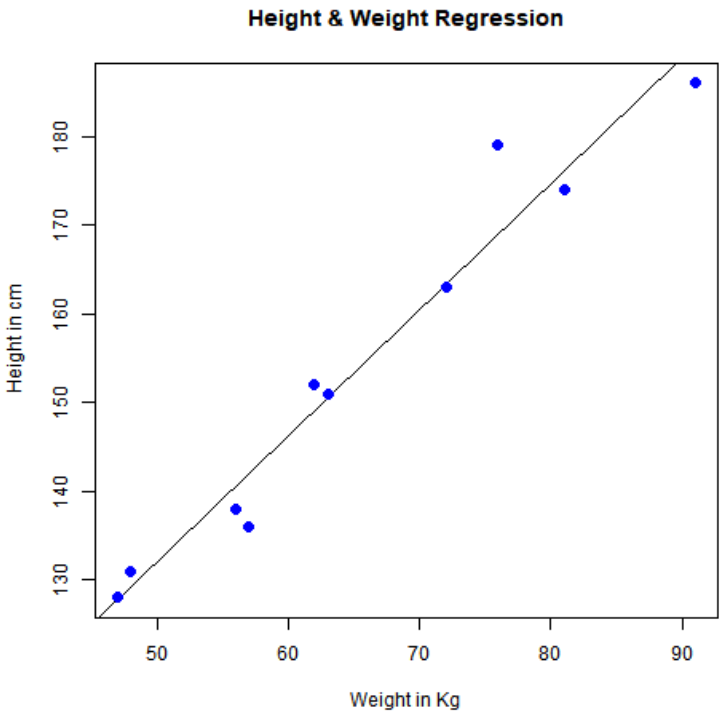
```
7:1   (Top Level)
```

```
Console   Terminal ×   Background Jobs ×
R 4.1.3 · ~/
> a <- data.frame(x = 170)
> result <- predict(relation,a)
> print(result)
       1
76.22869
>
```

Step 5: Visualize the Regression Graphically

```
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()
```

```
Console   Terminal ×   Background Jobs ×
R 4.1.3 · ~/
> 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()
RStudioGD
        2
>
```



Complete Code

```
x <- c(151,174,138,186,128,136,179,163,152,131) # height
y <- c(63,81,56,91,47,57,76,72,62,48) # weight
relation <- lm(y~x)
print(summary(relation))
a <- data.frame(x = 170)
result <- predict(relation,a)
print(result)
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()
```

Practical 10

AIM: Perform the logistic regression on the given data warehouse data.

Solution:

To perform this download the data from the link given below:
https://github.com/TarekDib03/Analytics/blob/master/Week3%20-%20Logistic%20Regression/Data/quality.csv

Code:

```
> quality <- read.csv('C:/Users/Anon/Desktop/BI Practical/quality.csv')
> str(quality)
```

Output:

```
'data.frame': 131 obs. of  14 variables:
 $ MemberID            : int  1 2 3 4 5 6 7 8 9 10 ...
 $ InpatientDays       : int  0 1 0 0 8 2 16 2 2 4 ...
 $ ERVisits            : int  0 1 0 1 2 0 1 0 1 2 ...
 $ OfficeVisits        : int  18 6 5 19 19 9 8 8 4 0 ...
 $ Narcotics           : int  1 1 3 0 3 2 1 0 3 2 ...
 $ DaysSinceLastERVisit: num  731 411 731 158 449 ...
 $ Pain                : int  10 0 10 34 10 6 4 5 5 2 ...
 $ TotalVisits         : int  18 8 5 20 29 11 25 10 7 6 ...
 $ ProviderCount       : int  21 27 16 14 24 40 19 11 28 21 ...
 $ MedicalClaims       : int  93 19 27 59 51 53 40 28 20 17 ...
 $ ClaimLines          : int  222 115 148 242 204 156 261 87 98 66 .
..
 $ StartedOnCombination: logi  FALSE FALSE FALSE FALSE FALSE FALSE .
..
 $ AcuteDrugGapSmall   : int  0 1 5 0 0 4 0 0 0 0 ...
 $ PoorCare            : int  0 0 0 0 0 1 0 0 1 0 ...
```

```
> table(quality$PoorCare)

 0 1
98 33
```

```
> 98/131
[1] 0.7480916
```

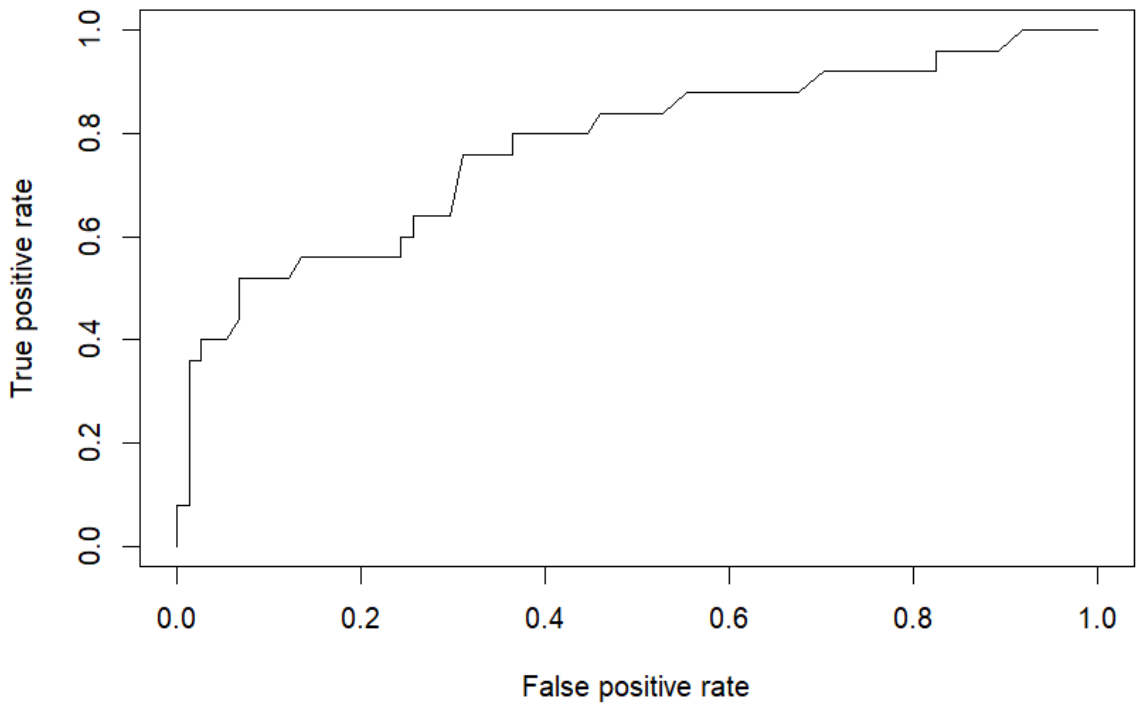
```
> install.packages("caTools")
```

```
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Anon/AppData/Local/R/win-library/4.2’
(as ‘lib’ is unspecified)
also installing the dependency ‘bitops’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/bitops_1.0-7.zip'
Content type 'application/zip' length 31679 bytes (30 KB)
downloaded 30 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caTools_1.18.2.zip'
Content type 'application/zip' length 245985 bytes (240 KB)
downloaded 240 KB
```

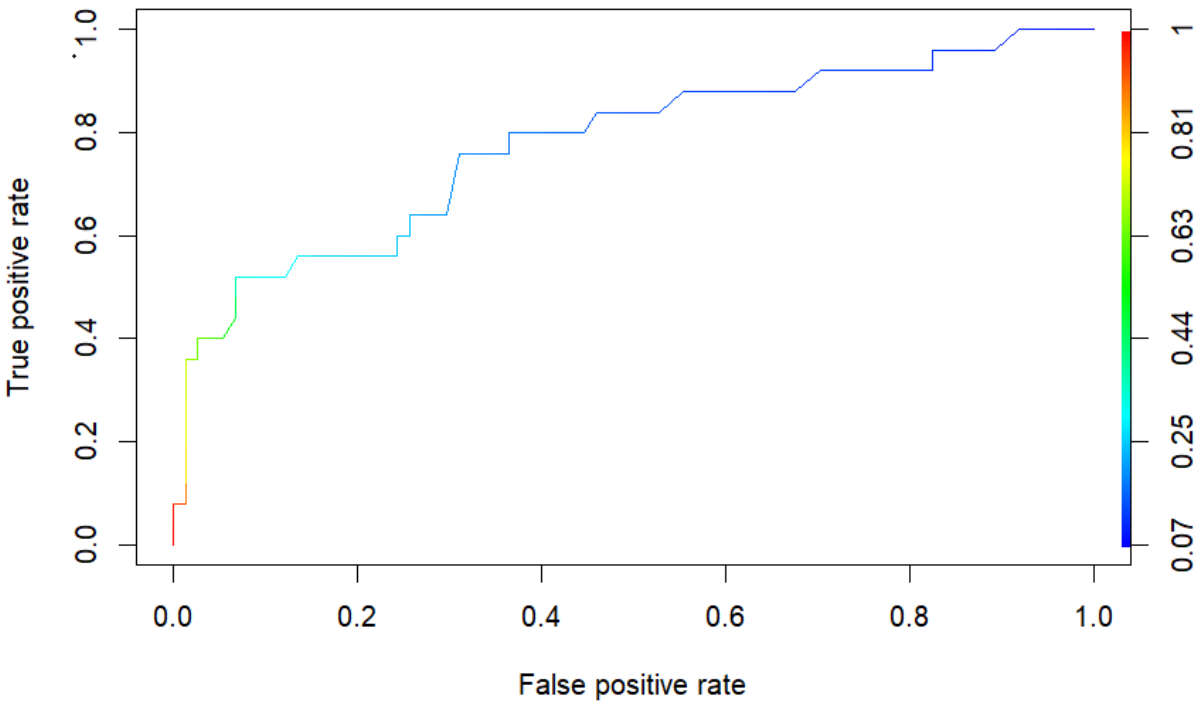
```
package 'bitops' successfully unpacked and MD5 sums checked
package 'caTools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Anon\AppData\Local\Temp\RtmpCyEvnb\downloaded_packag
es

> library(caTools)
> set.seed(88)
> split = sample.split(quality$PoorCare, SplitRatio = 0.75)
> split
  [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  T
RUE FALSE
 [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  T
RUE  TRUE
 [25] FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  T
RUE  TRUE
 [37] FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FA
LSE  TRUE
 [49] FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  T
RUE  TRUE
 [61]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FA
LSE  TRUE
 [73]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  T
RUE FALSE
 [85]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  T
RUE  TRUE
 [97]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  T
RUE FALSE
[109]  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  T
RUE  TRUE
[121] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FA
LSE

> qualityTrain = subset(quality, split == TRUE)
> qualityTest = subset(quality, split == FALSE)
> nrow(qualityTrain)
[1] 99
> nrow(qualityTest)
[1] 32

> QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics, data = quali
tyTrain, family = binomial)
> summary(QualityLog)

Call:
glm(formula = PoorCare ~ OfficeVisits + Narcotics, family = binomial
,
    data = qualityTrain)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.06303  -0.63155  -0.50503  -0.09689   2.16686

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.64613    0.52357  -5.054 4.33e-07 ***
OfficeVisits  0.08212    0.03055   2.688  0.00718 **
Narcotics     0.07630    0.03205   2.381  0.01728 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 111.888  on 98  degrees of freedom
Residual deviance:  89.127  on 96  degrees of freedom
AIC: 95.127

Number of Fisher Scoring iterations: 4

> predictTrain = predict(QualityLog, type = "response")
> summary(predictTrain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.06623 0.11912 0.15967 0.25253 0.26765 0.98456

> tapply(predictTrain, qualityTrain$PoorCare, mean)
        0         1
0.1894512 0.4392246
> table(qualityTrain$PoorCare, predictTrain > 0.5)

    FALSE TRUE
  0    70    4
  1    15   10
> 10/25
[1] 0.4
> 70/74
[1] 0.9459459
> table(qualityTrain$PoorCare, predictTrain > 0.7)

    FALSE TRUE
  0    73    1
  1    17    8

> 8/25
[1] 0.32
> 73/74
[1] 0.9864865
> table(qualityTrain$PoorCare, predictTrain > 0.2)

    FALSE TRUE
  0    54   20
  1     9   16
> 16/25
[1] 0.64
> 54/74
[1] 0.7297297

> install.packages("ROCR")

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Anon/AppData/Local/R/win-library/4
.2'
(as 'lib' is unspecified)
also installing the dependencies 'gtools', 'gplots'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/gtools_
3.9.4.zip'
Content type 'application/zip' length 359411 bytes (350 KB)
downloaded 350 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/gplots_
3.1.3.zip'
Content type 'application/zip' length 603183 bytes (589 KB)
downloaded 589 KB
```

```
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/ROCR_1.
0-11.zip'
Content type 'application/zip' length 453649 bytes (443 KB)
downloaded 443 KB

package ‘gtools’ successfully unpacked and MD5 sums checked
package ‘gplots’ successfully unpacked and MD5 sums checked
package ‘ROCR’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Anon\AppData\Local\Temp\RtmpCyEvnb\downloaded_packag
es

> library(ROCR)
> ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)
> ROCRpref = performance(ROCRpred, "tpr", "fpr")
> plot(ROCRpref)
```




```
> plot(ROCRpref, colorize=TRUE)
```



Complete code:

```
quality <- read.csv('C:/Users/Anon/Desktop/BI Practical/quality.csv'
)
str(quality)
table(quality$PoorCare)
98/131
install.packages("caTools")
library(caTools)
set.seed(88)
split = sample.split(quality$PoorCare, SplitRatio = 0.75)
split
qualityTrain = subset(quality, split == TRUE)
qualityTest = subset(quality, split == FALSE)
nrow(qualityTrain)
nrow(qualityTest)
QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics, data = quality
Train, family = binomial)
summary(QualityLog)
predictTrain = predict(QualityLog, type = "response")
summary(predictTrain)
tapply(predictTrain, qualityTrain$PoorCare, mean)
table(qualityTrain$PoorCare, predictTrain > 0.5)
10/25
70/74
table(qualityTrain$PoorCare, predictTrain > 0.7)
8/25
73/74
table(qualityTrain$PoorCare, predictTrain > 0.2)
16/25
```

```
install.packages("ROCR")
library(ROCR)
ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)
ROCRpref = performance(ROCRpred, "tpr", "fpr")
plot(ROCRpref)
plot(ROCRpref, colorize=TRUE)
plot(ROCRpref, colorize=TRUE, print, print.cutoffs.at = seq(0,1,by =
0.1), text.adj = c(-0.2,0.7))
```

Practical 11

AIM: Practical implementation of Decision tree using R tool

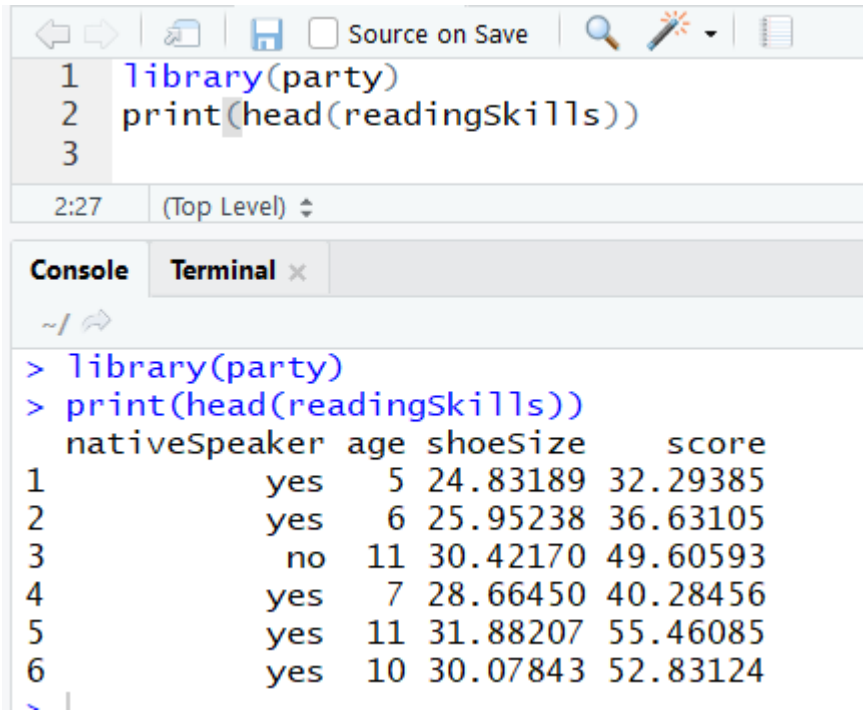
Solution:

Step 1: Install “party package”

>install.packages("party")

Step 2: Code

```
library(party)
print(head(readingSkills))
```




```
input.dat <- readingSkills[c(1:105),]
png(file = "decision_tree.png")
output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)
plot(output.tree)
dev.off()
```

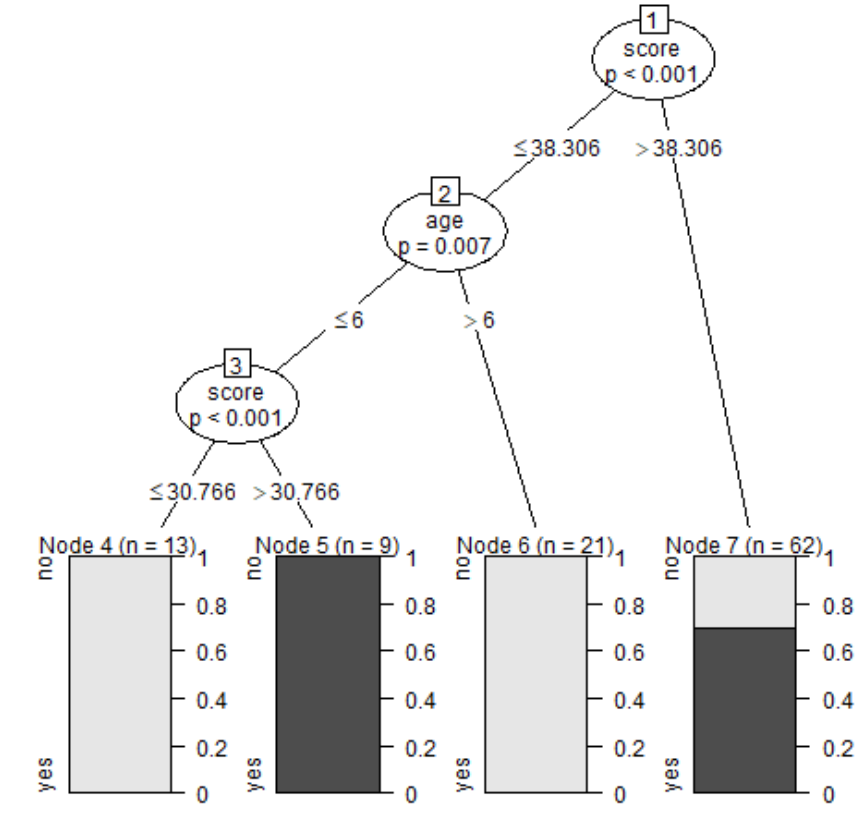
OUTPUT

```
4  input.dat <- readingSkills[c(1:105),]
5  png(file = "decision_tree.png")
6  output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)
7  plot(output.tree)
8  dev.off()
```

8:10 (Top Level)

Console Terminal

```
> input.dat <- readingSkills[c(1:105),]
> png(file = "decision_tree.png")
> output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)
> plot(output.tree)
> dev.off()
null device
          1
>
```



Complete Code

```
library(party)
print(head(readingSkills))
input.dat <- readingSkills[c(1:105),]
png(file = "decision_tree.png")
output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)
plot(output.tree)
dev.off()
```