

**BUNTS SANGHA'S
S.M. SHETTY COLLEGE OF SCIENCE, COMMERCE &
MANAGEMENT STUDIES (AUTONOMOUS),
POWAI, MUMBAI-76**



SUBJECT: Business Intelligence and Data Analytics

CLASS: T.Y.B.Sc.I.T (2024-25)

SEMESTER: VI

SUBMITTED BY (Name): _____

ROLL NUMBER: _____

**SUBMITTED TO
DEPARTMENT OF INFORMATION TECHNOLOGY**

Asst. Prof. Raveena Shetty

Subject In-charge

Dr. Tushar Sambare

Coordinator

External Examiner

For Partial Fulfilment for Degree of
Bachelor of Science on IT (Information Technology) in 2024-25

Index

Sr. No.	Topic	Date	Sign
1	<p>Perform the analysis for the following:</p> <p>a. Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.</p> <p>b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis.</p>		
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 a 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).		

BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS

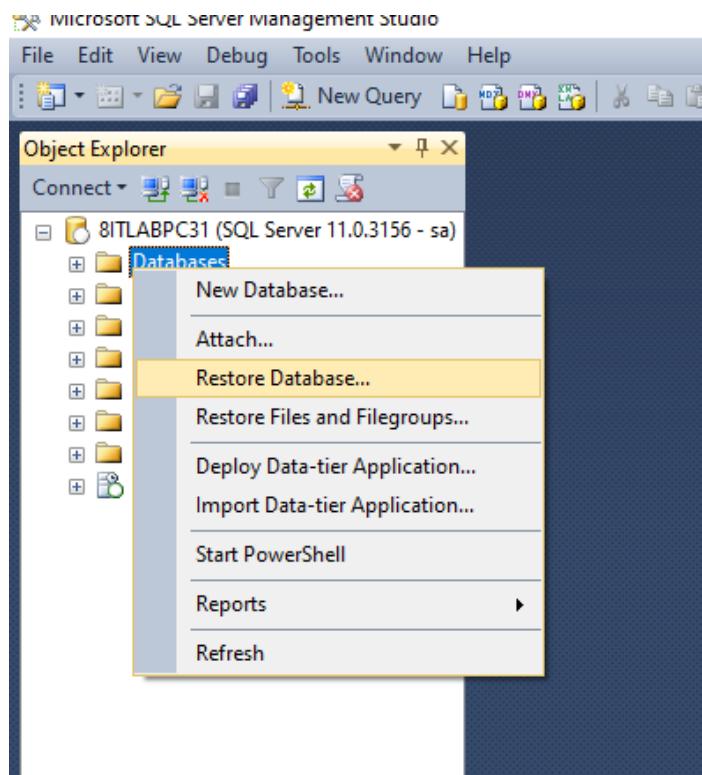
8	Perform data visualization: a. Perform data visualization using Python on any sales data. b. Perform data visualization using PowerBI on any sales data.		
9	Create the Data staging area for the selected database using SQL.		
10	Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.		

Practical No. 1

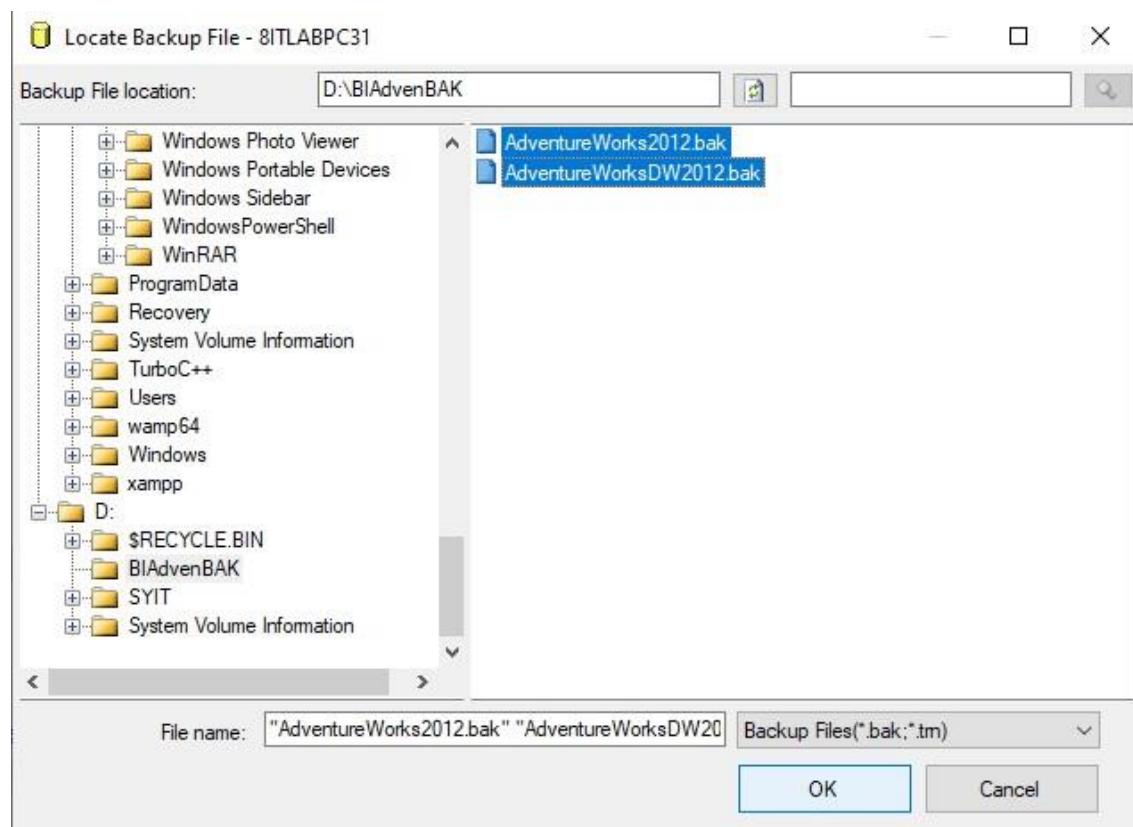
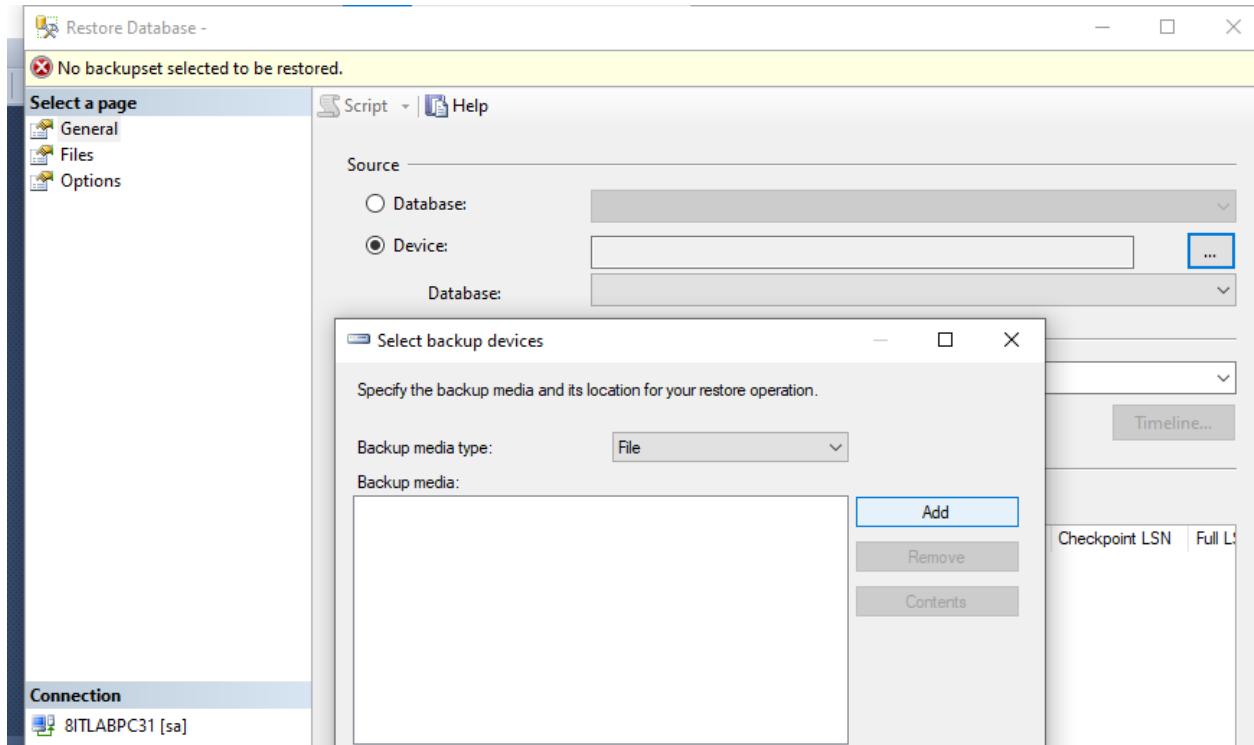
Perform the analysis for the following

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

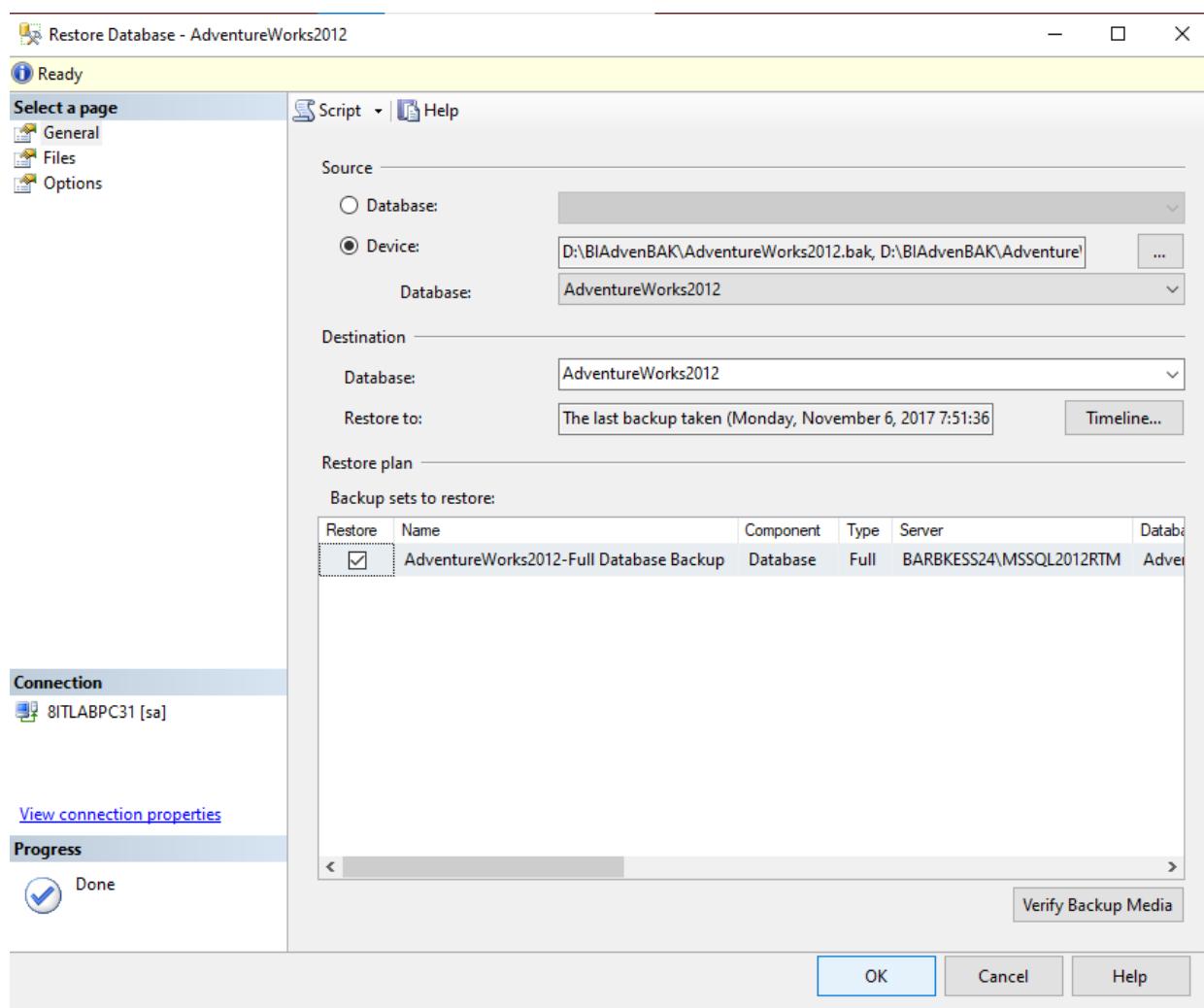
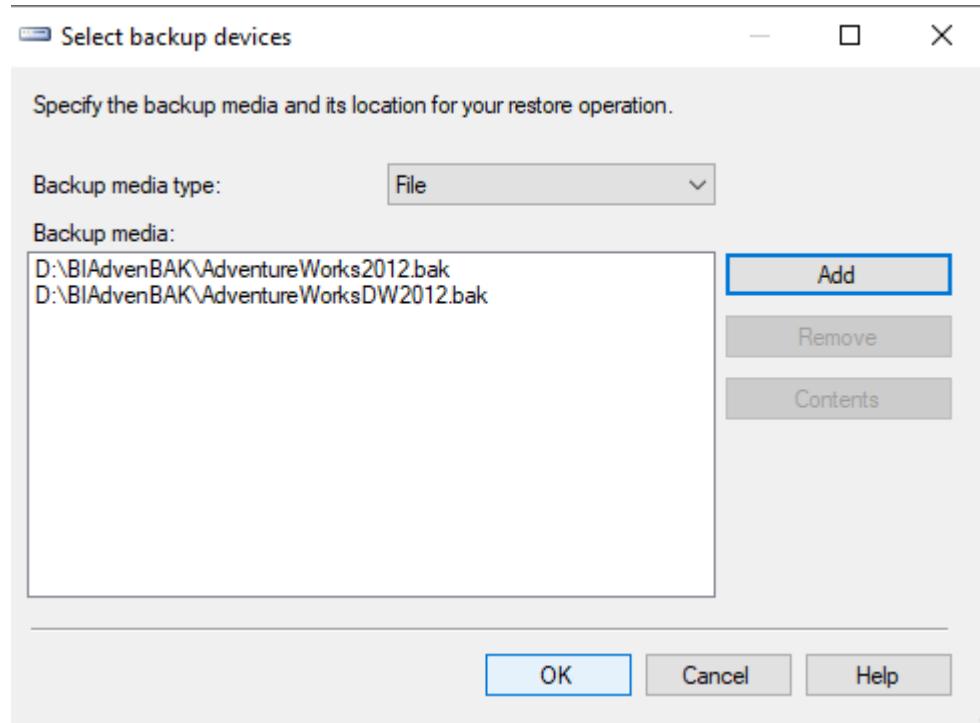
1. From ‘BI Dataset’ (located in Desktop) copy AdventureWorks2012.bak and AdventureWorksDW2012.bak file → create a new directory in ‘D’ drive (Name it as BI) paste it there.
2. Open SSMS → connect it to the sever.
3. Right click on ‘Database’ → select Restore Database → add both the files into it.



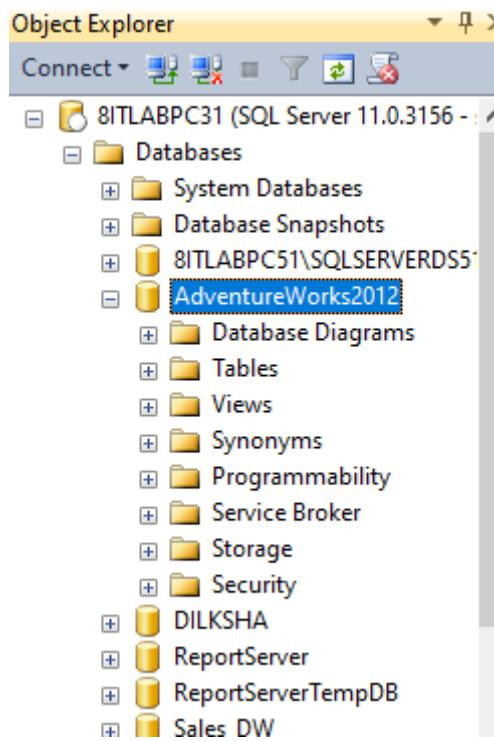
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



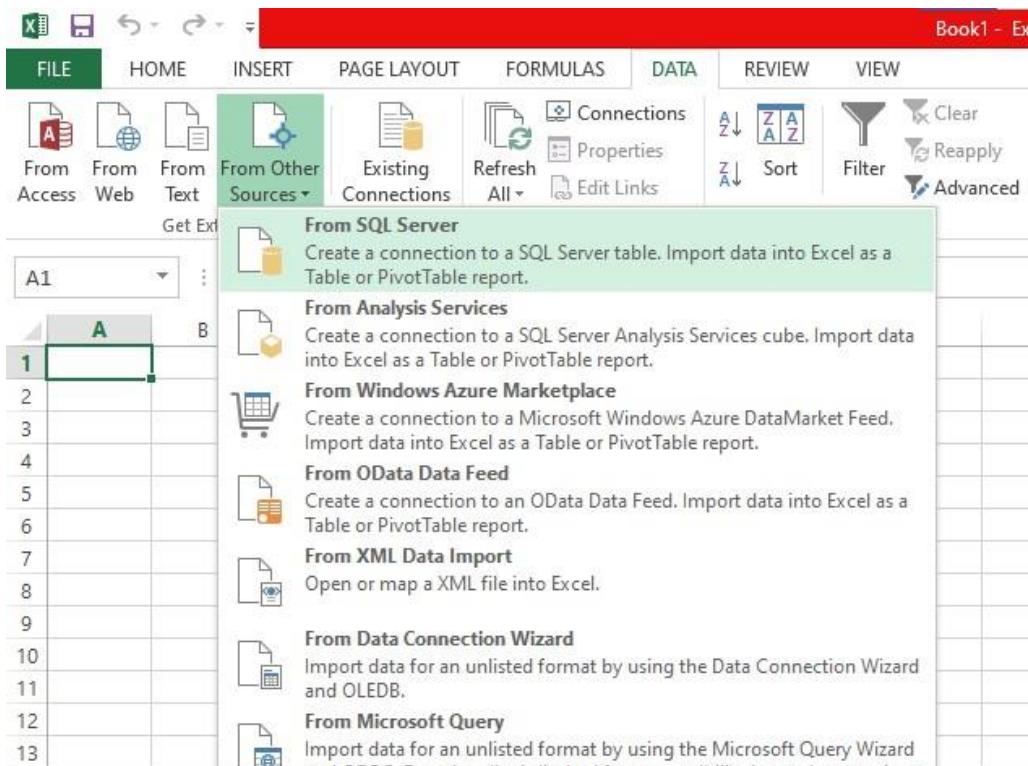
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



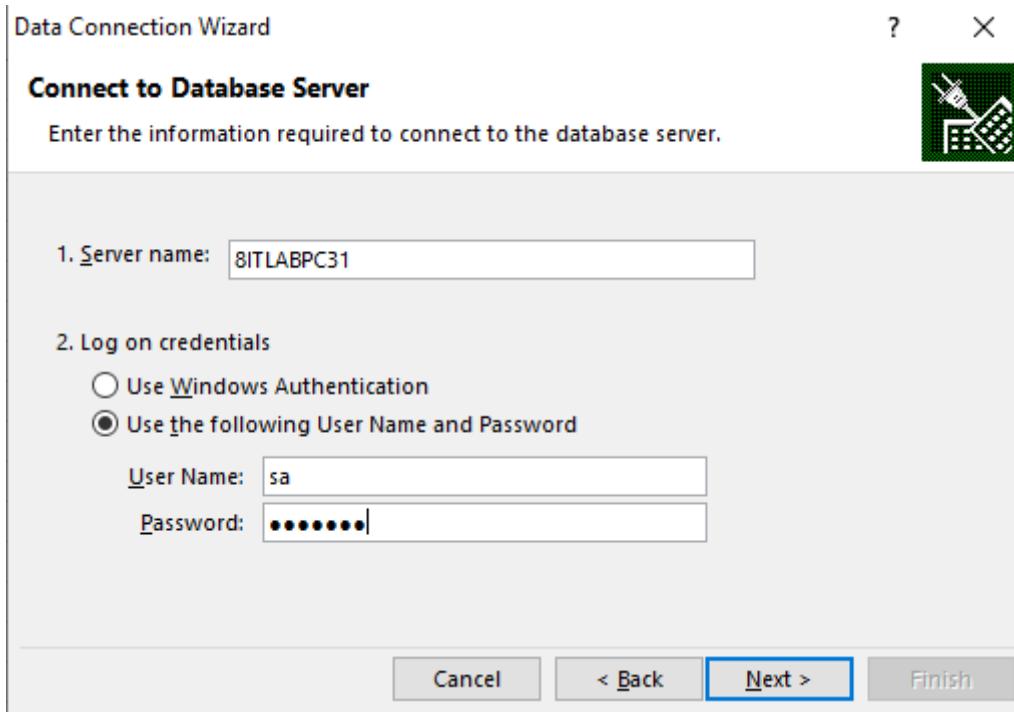
4. After adding the file we should see the 'AdventureWorks2012' database.



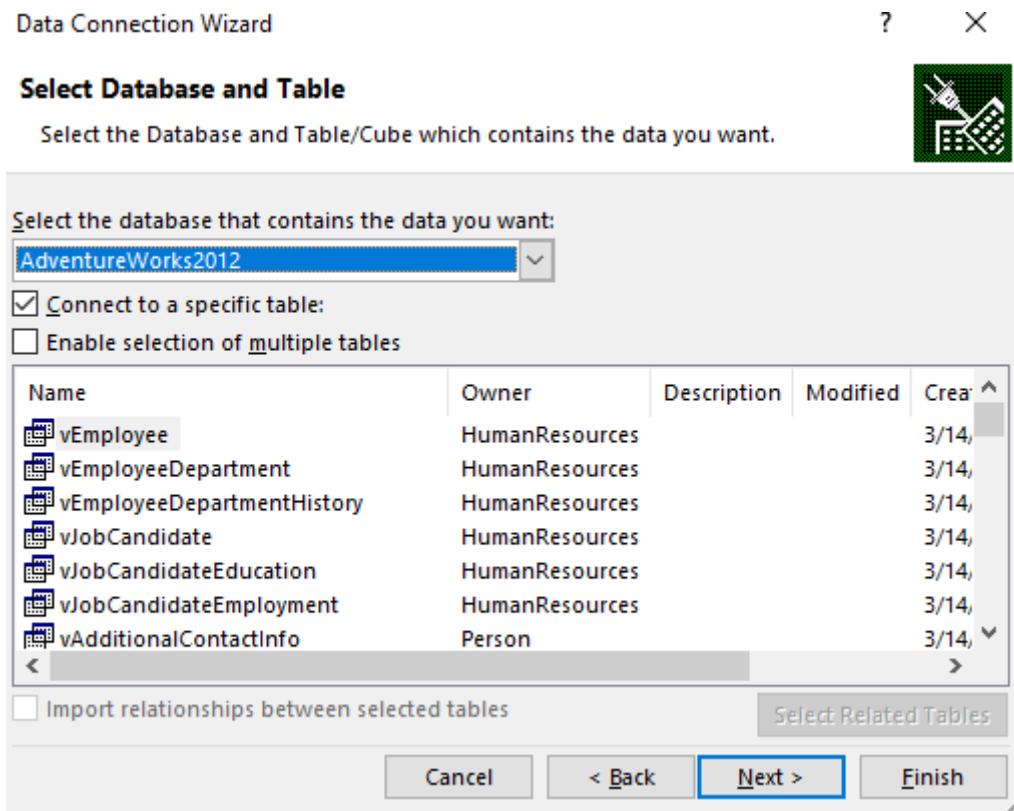
5. Open Microsoft Excel → Click on the "Data" tab → Select "Get Data" > "From Other Sources" > "From SQL Server"



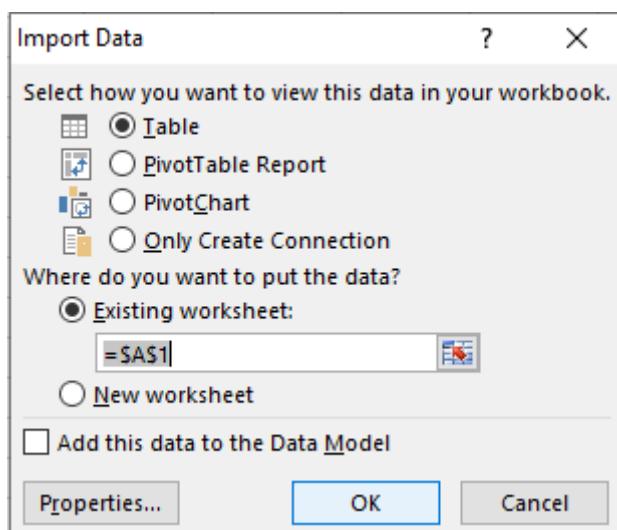
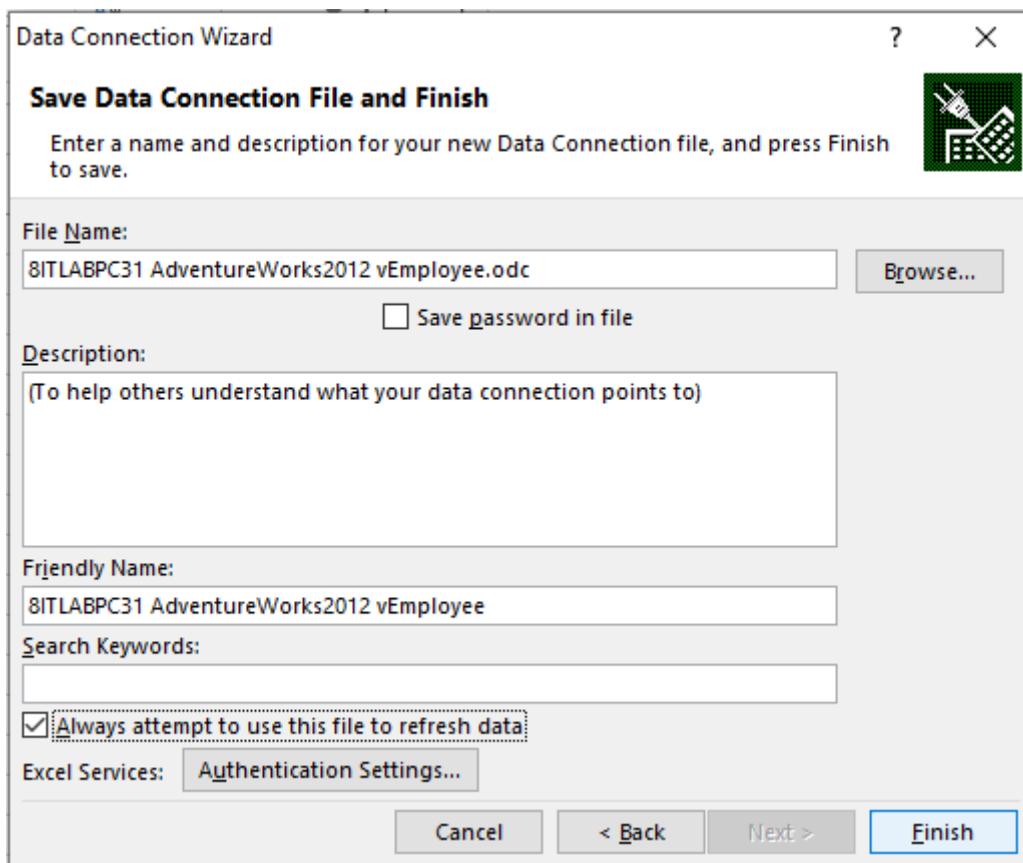
6. Enter the server name and database credentials.



7. Choose the required data table from the warehouse.

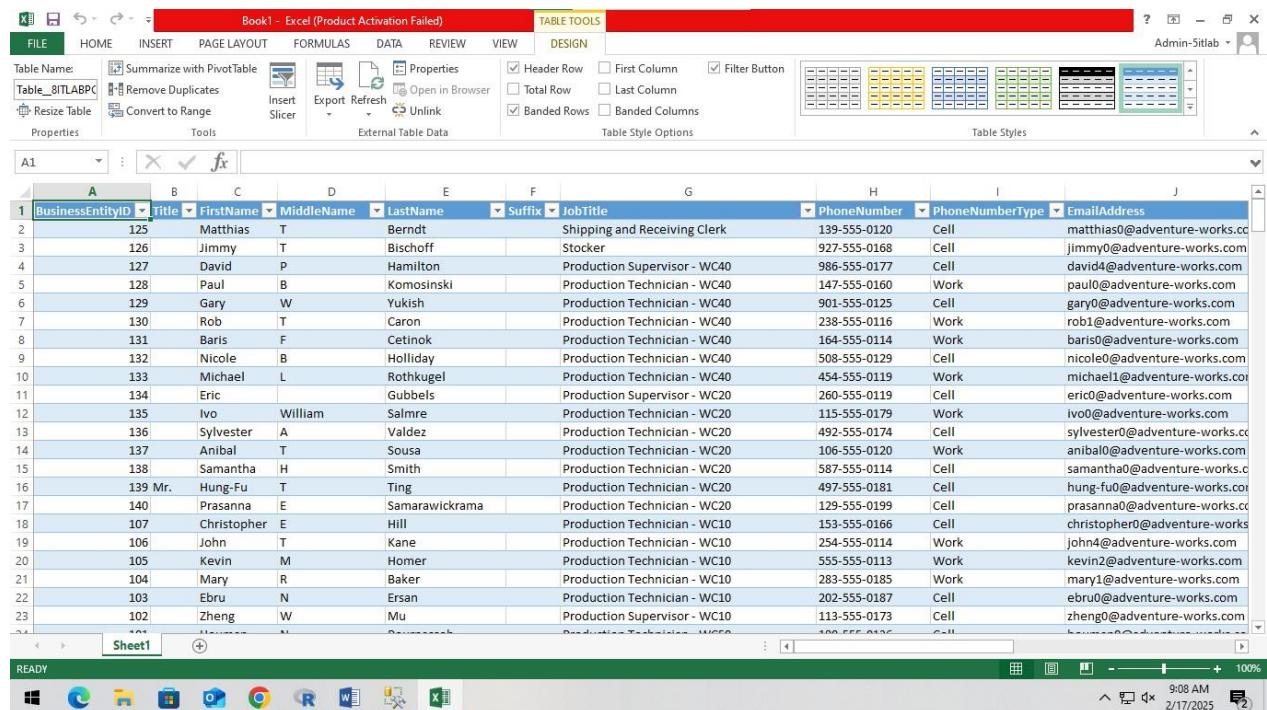


8. Check ‘Always attempt to use this file to refresh data’ → finish.



BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS

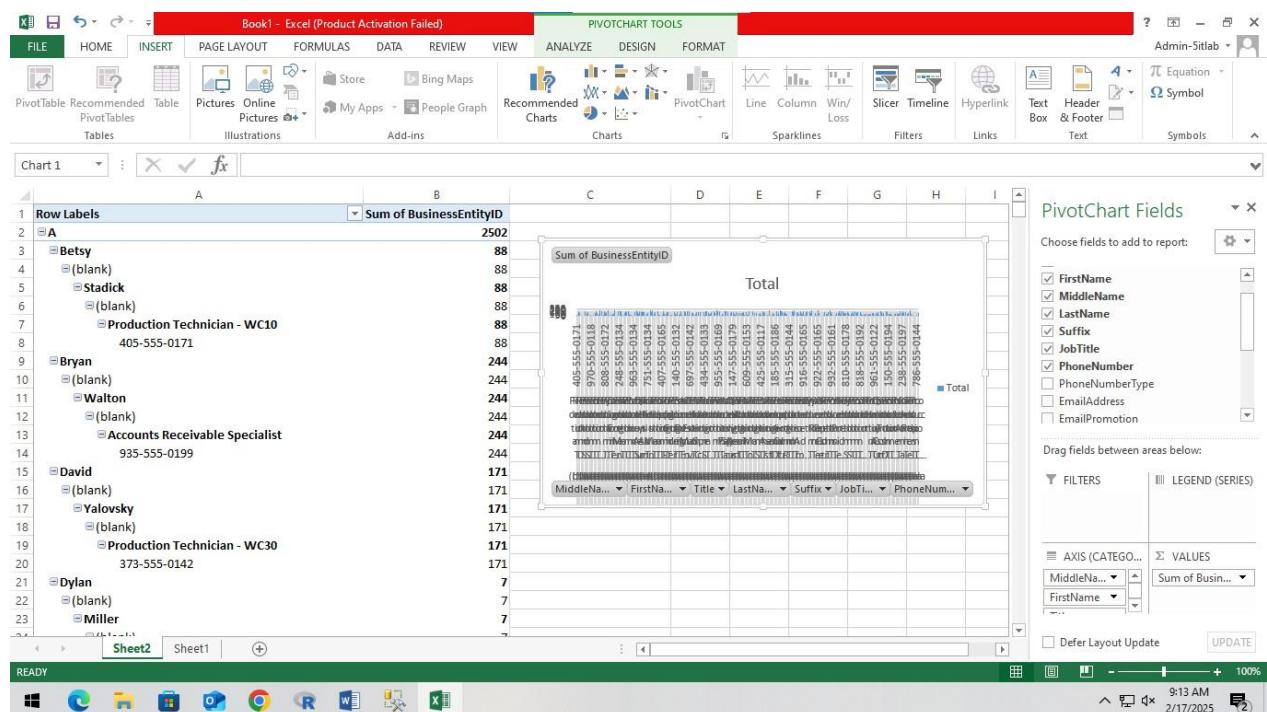
9. Data is loaded from the server.



The screenshot shows a Microsoft Excel spreadsheet titled "Book1 - Excel (Product Activation Failed)". The table has columns: BusinessEntityID, Title, FirstName, MiddleName, LastName, Suffix, JobTitle, PhoneNumber, PhoneNumberType, and EmailAddress. The data consists of 23 rows of employee information. The "DESIGN" tab is selected in the ribbon. The status bar at the bottom right shows the date as 2/17/2025 and the time as 9:08 AM.

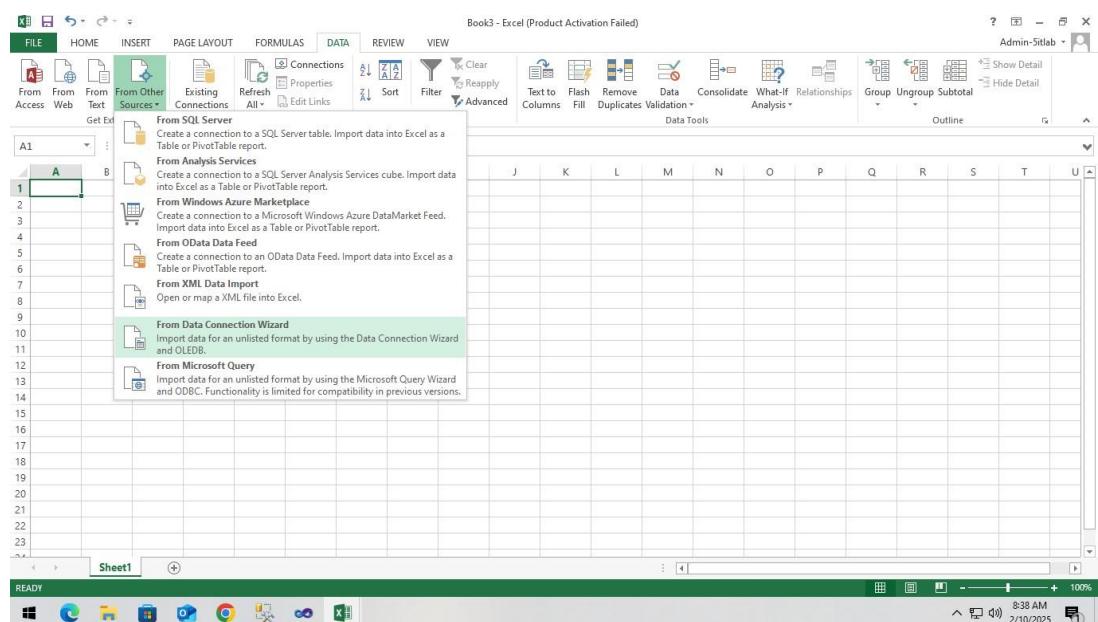
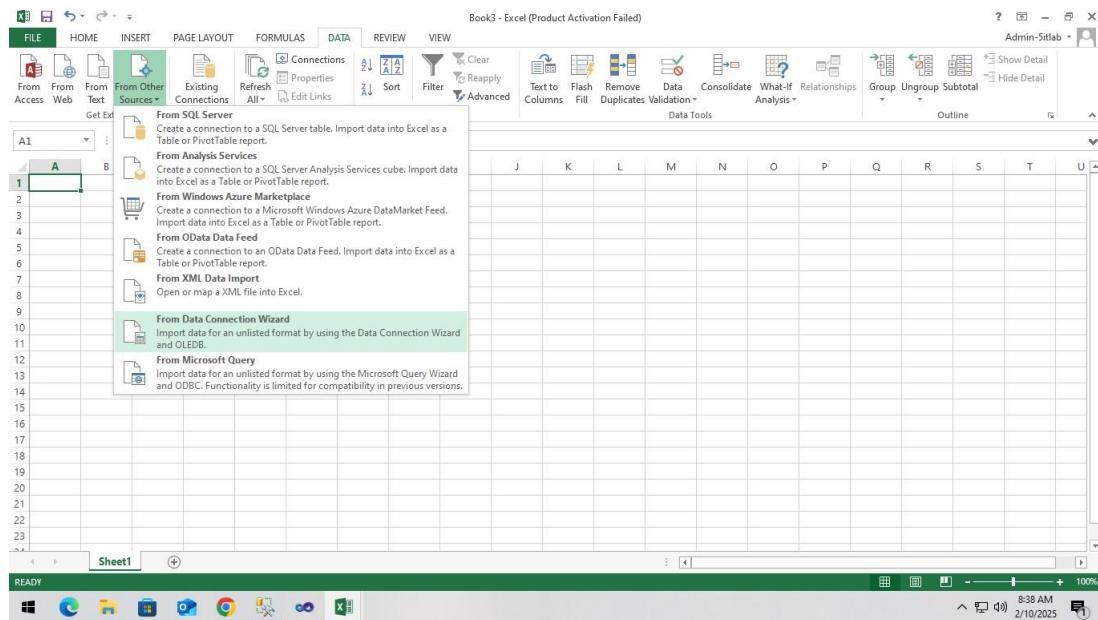
BusinessEntityID	Title	FirstName	MiddleName	LastName	Suffix	JobTitle	PhoneNumber	PhoneNumberType	EmailAddress
125	Matthias	T		Berndt		Shipping and Receiving Clerk	139-555-0120	Cell	matthias0@adventure-works.com
126	Jimmy	T		Bischoff		Stocker	927-555-0168	Cell	jimmy0@adventure-works.com
127	David	P		Hamilton		Production Supervisor - WC40	986-555-0177	Cell	david0@adventure-works.com
128	Paul	B		Komosinski		Production Technician - WC40	147-555-0160	Work	paul0@adventure-works.com
129	Gary	W		Yukish		Production Technician - WC40	901-555-0125	Cell	gary0@adventure-works.com
130	Rob	T		Caron		Production Technician - WC40	238-555-0116	Work	robi@adventure-works.com
131	Baris	F		Cetinok		Production Technician - WC40	164-555-0114	Work	baris0@adventure-works.com
132	Nicole	B		Holliday		Production Technician - WC40	508-555-0129	Cell	nicole0@adventure-works.com
133	Michael	L		Rothkugel		Production Technician - WC40	454-555-0119	Work	michael1@adventure-works.com
134	Eric			Gubbels		Production Supervisor - WC20	260-555-0119	Cell	eric0@adventure-works.com
135	Ivo	William		Salmre		Production Technician - WC20	115-555-0179	Work	ivo0@adventure-works.com
136	Sylvester	A		Valdez		Production Technician - WC20	492-555-0174	Cell	sylvester0@adventure-works.com
137	Anibal	T		Sousa		Production Technician - WC20	106-555-0120	Work	anibal0@adventure-works.com
138	Samantha	H		Smith		Production Technician - WC20	587-555-0114	Cell	samantha0@adventure-works.com
139	Mr. Hung-Fu	T		Ting		Production Technician - WC20	497-555-0181	Cell	hung-fu0@adventure-works.com
140	Prasanna	E		Samarawickrama		Production Technician - WC20	129-555-0199	Cell	prasanna0@adventure-works.com
107	Christopher	E		Hill		Production Technician - WC10	153-555-0166	Cell	christopher0@adventure-works.com
106	John	T		Kane		Production Technician - WC10	254-555-0114	Work	john4@adventure-works.com
105	Kevin	M		Homer		Production Technician - WC10	555-555-0113	Work	kevin2@adventure-works.com
104	Mary	R		Baker		Production Technician - WC10	283-555-0185	Work	mary1@adventure-works.com
103	Ebru	N		Ersan		Production Technician - WC10	202-555-0187	Cell	ebru0@adventure-works.com
102	Zheng	W		Mu		Production Supervisor - WC10	113-555-0173	Cell	zheng0@adventure-works.com

10. Then make the 'Pivot Chart & Table' accordingly.

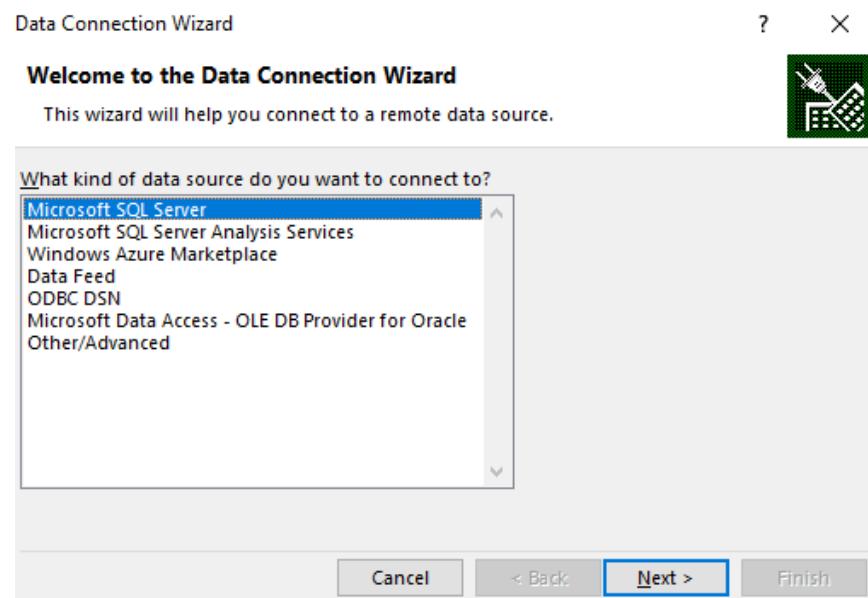


b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis.

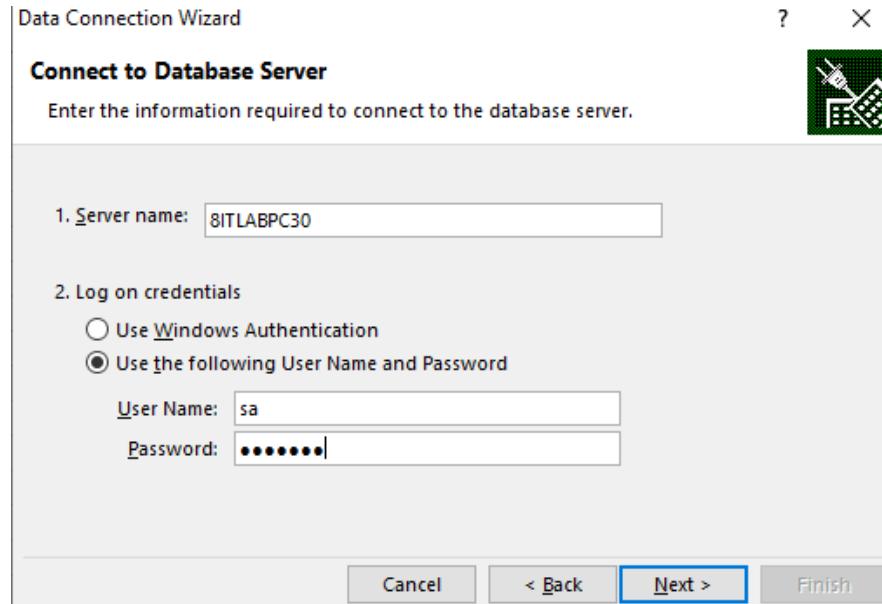
- Kindly perform Practical 9 & 10 → Open Excel → Go to ‘Data Tab’ → get external data → from other source → from data connection wizard.



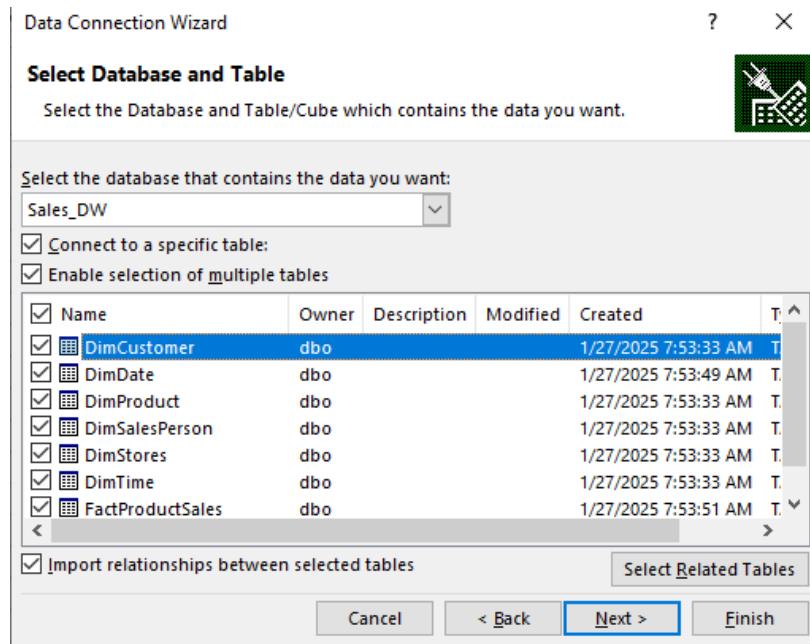
2. In data connection wizard → select Microsoft sql server → click on next.



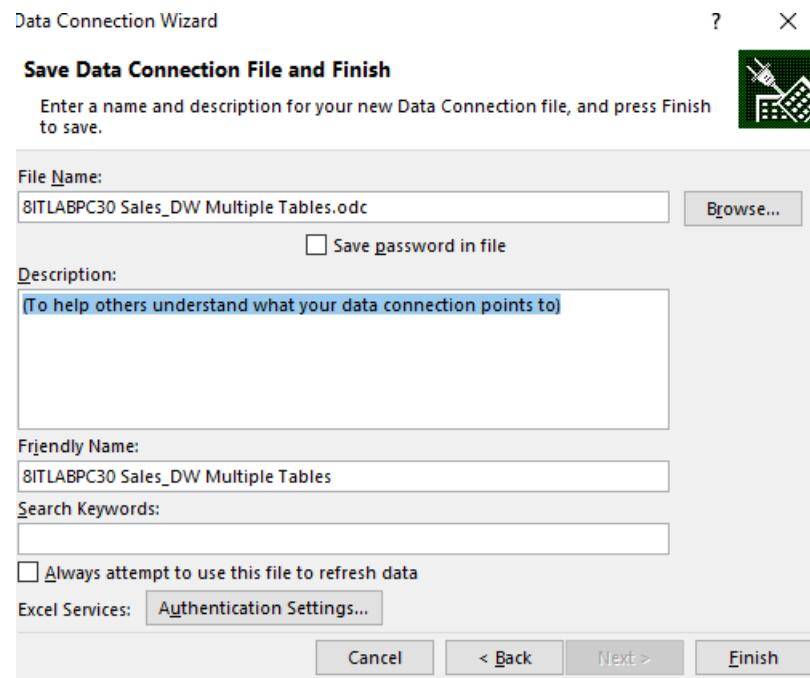
3. In connect to database server provide server (name: - Microsoft sql server) → Use the following username & psw. [enter the name & psw] → click on next.



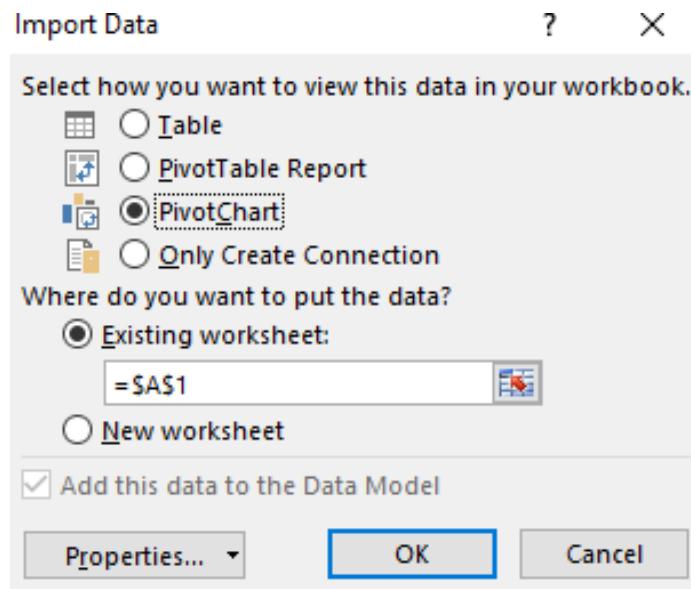
4. In select Database table → select Sales_DW (already created in SQL) → check all dimensions and import relationships between selected tables → next.



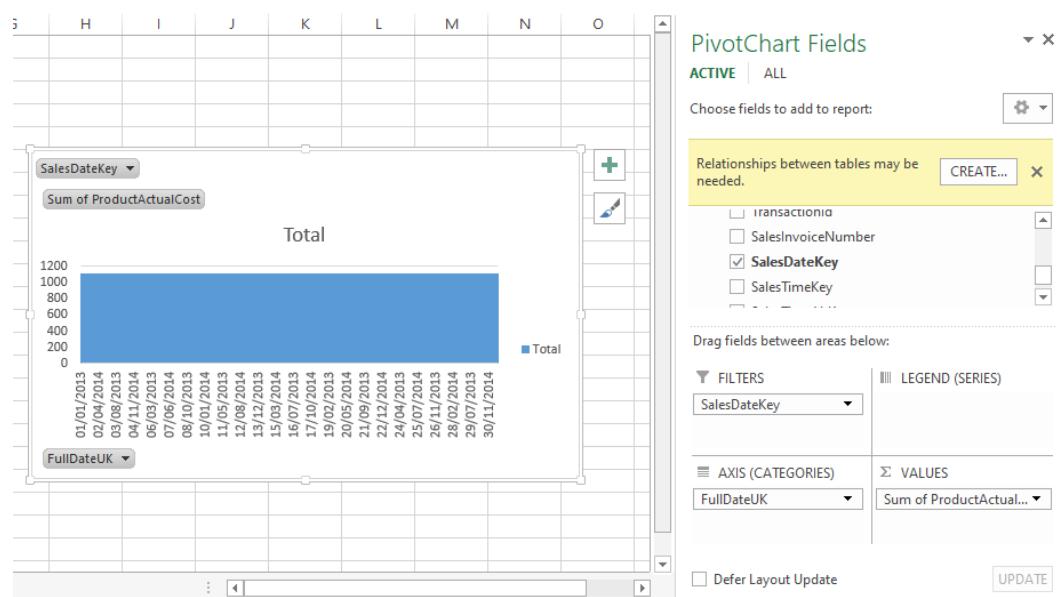
5. In save data connection files browse path and click on finish.



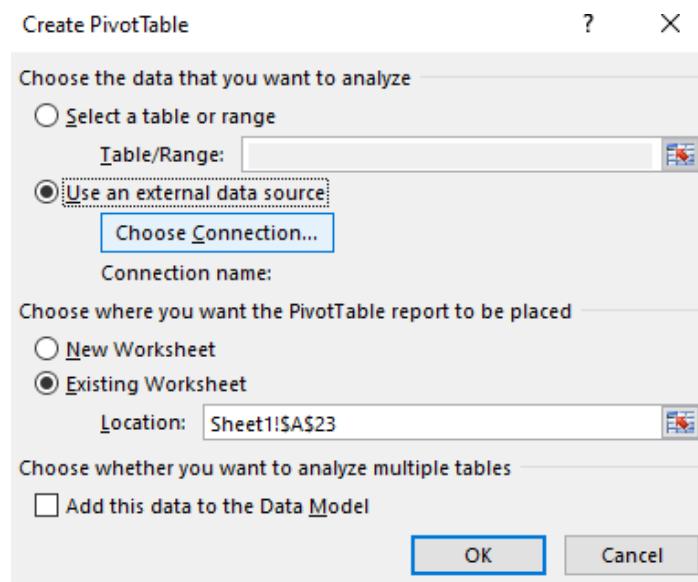
6. In import data select Pivot Chart and click on OK.



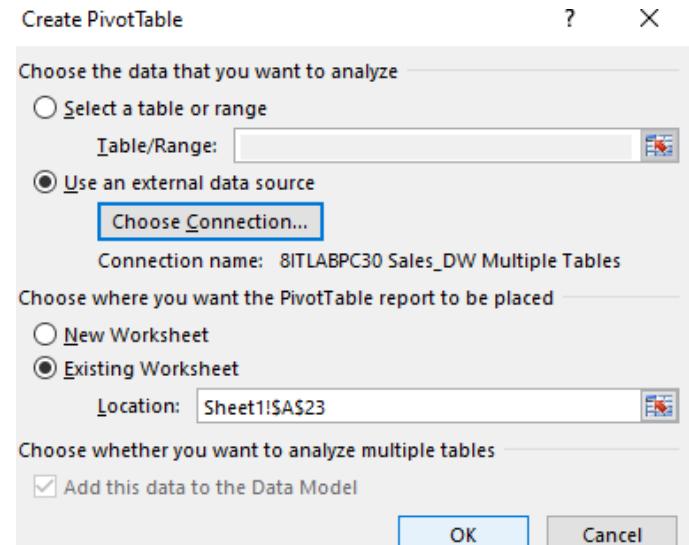
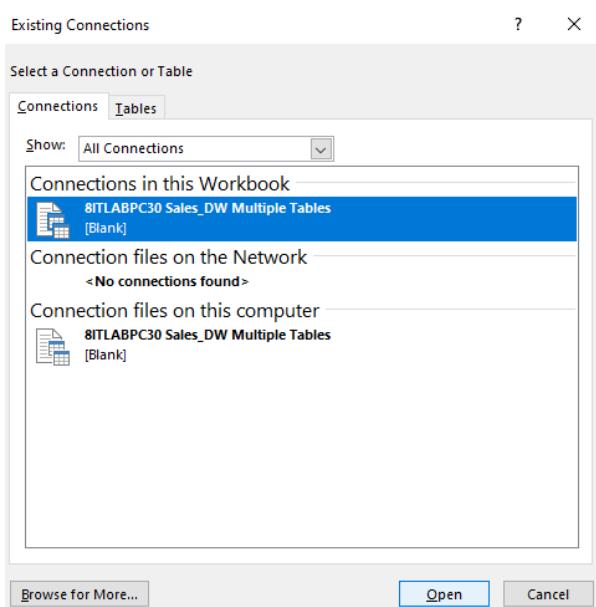
7. From fields put SalesDateKey in ‘filters’, FullDateUK in ‘axis’ and ProductActualCost in ‘values’.



8. Go to 'Insert' tab → Pivot Table.



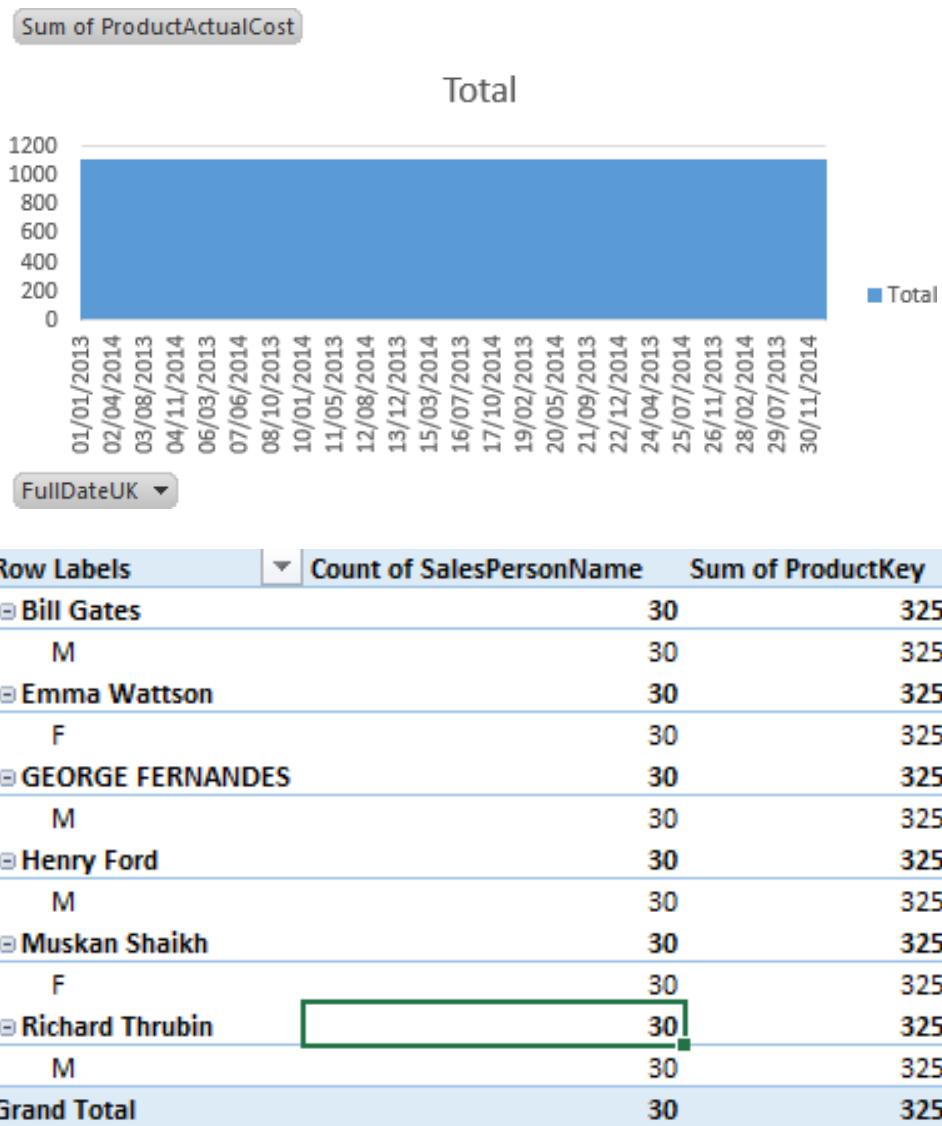
9. Click on choose connection to select existing connection with Sales_DW and click on open.



10. From fields put CustomerName & Gender in ‘Rows’, SalesPerson & ProductKey in ‘values’.

	Count of SalesPersonName	Sum of ProductKey
Bill Gates	30	325
M	30	325
Emma Wattson	30	325
F	30	325
GEORGE FERNANDES	30	325
M	30	325
Henry Ford	30	325
M	30	325
Muskan Shaikh	30	325
F	30	325
Richard Thrubin	30	325
M	30	325
Grand Total	30	325

11. Pivot Chart & Table is created.



Practical No. 2

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

1. Prepare data in Excel.

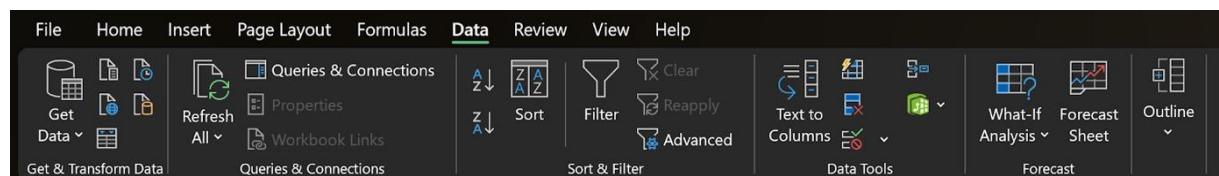
	A	B	C	D	E	F	G
1	Name	STQA	IS	BIDA	FGIS	ITIM	Total
2	Yo	55	50		65	53	

2. In ‘Total’ type =sum(select cell from B2 till F2) & hit enter.

	A	B	C	D	E	F	G	H
1	Name	STQA	IS	BIDA	FGIS	ITIM	Total	
2	Yo	55	50		65	=sum(B2:F2)		
3							SUM(number1, [number2], ...)	

	A	B	C	D	E	F	G
1	Name	STQA	IS	BIDA	FGIS	ITIM	Total
2	Yo	55	50		65	53	223

3. To do What if Analysis (Goal Seek) → From ribbon go to ‘Data’ → select What if Analysis.



4. Select Goal Seek and enter the values → Ok.

	A	B	C	D	E	F	G
1	Name	STQA	IS	BIDA	FGIS	ITIM	Total
2	Yo	55	50		65	53	223
3	Goal Seek		?	X			
4							
5	Set cell:		\$G\$2	▲			
6	To value:		280				
7	By changing cell:		\$D\$2	▲			
8							
9							
10							
	<input type="button" value="OK"/>		<input type="button" value="Cancel"/>				

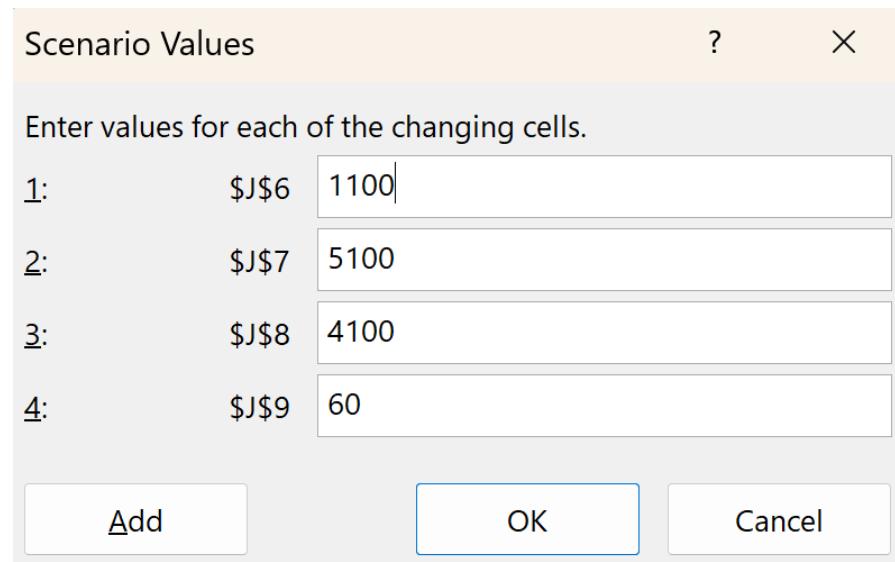
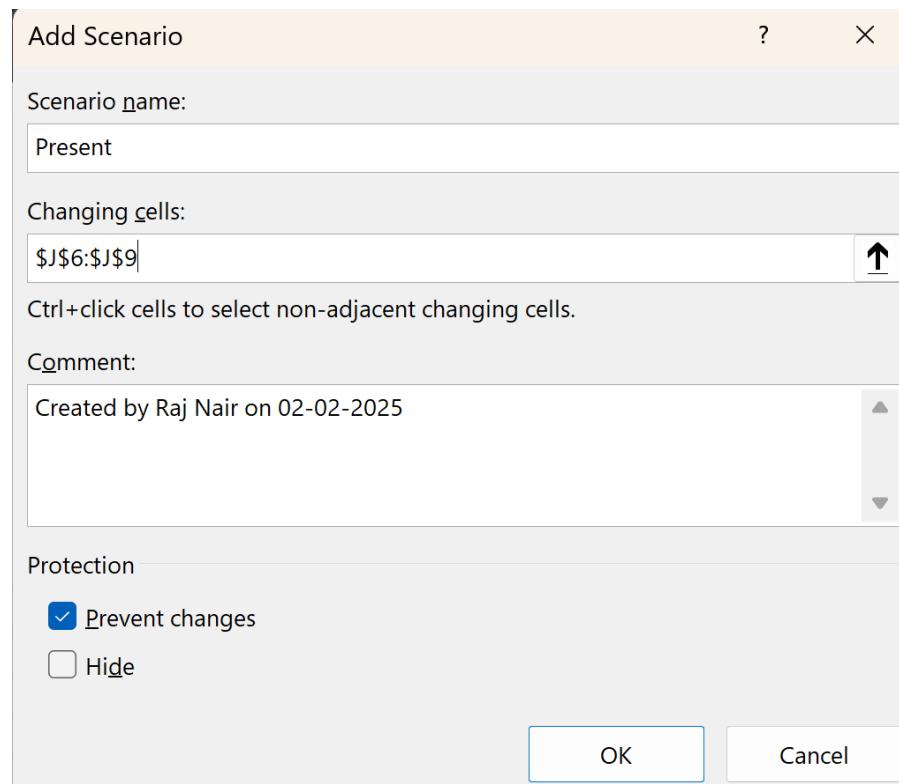
	A	B	C	D	E	F	G
1	Name	STQA	IS	BIDA	FGIS	ITIM	Total
2	Yo	55	50	57	65	53	280
3	Goal Seek Status		?	X			
4							
5	Goal Seeking with Cell G2 found a solution.			Step			
6							
7	Target value: 280			Pause			
8	Current value: 280						
9							
10	<input type="button" value="OK"/>		<input type="button" value="Cancel"/>				
11							

5. Prepare new data to perform What if Analysis (Scenario Manager) → From ribbon go to 'Data' → select What if Analysis → Scenario Manager.

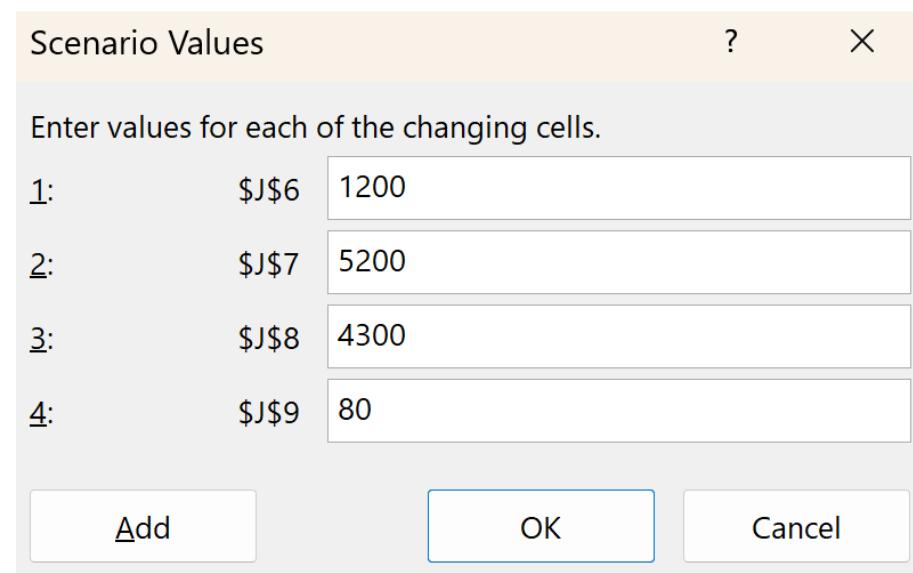
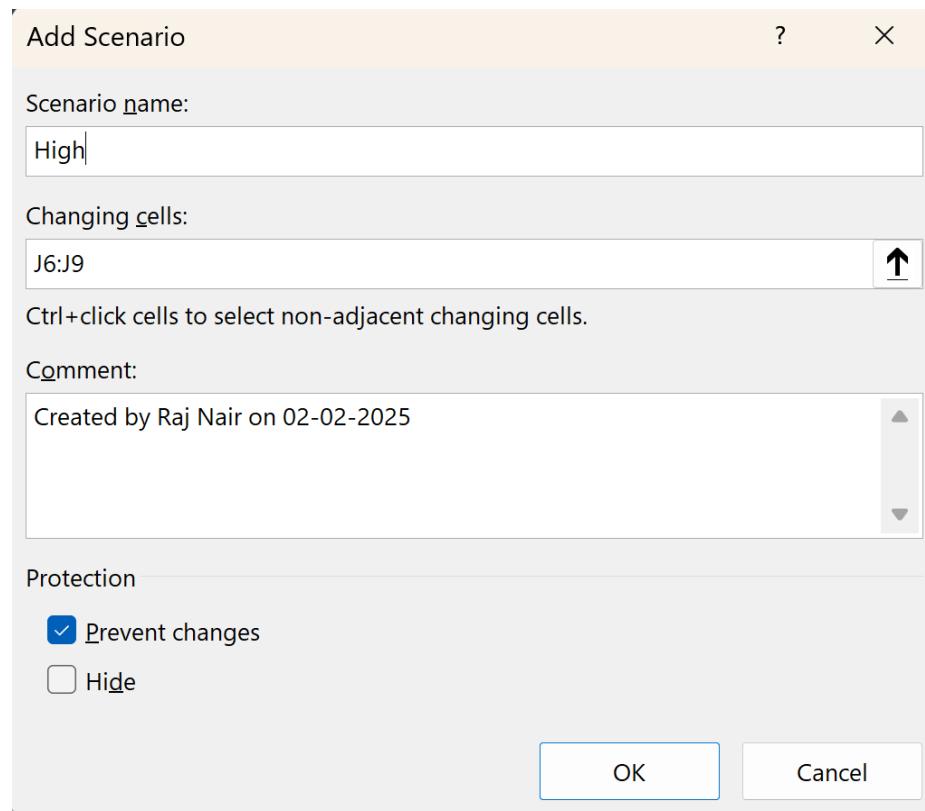
Parts	Cost
Mouse	1100
Chip	5100
Monitor	4100
Keyboard	60
Total	10360

The screenshot shows the 'Scenario Manager' dialog box in Excel. The 'Scenarios:' section is empty, displaying the message 'No Scenarios defined. Choose Add to add scenarios.' Below it are buttons for 'Add...', 'Delete', 'Edit...', 'Merge...', and 'Summary...'. The 'Changing cells:' and 'Comment:' sections are also present. At the bottom are 'Show' and 'Close' buttons. To the right of the dialog, a portion of the worksheet is visible, showing a table with columns 'Parts' and 'Cost'. The rows include Mouse (1100), Chip (5100), Monitor (4100), Keyboard (60), and a summary row 'Total' (10360). The 'Total' cell is highlighted with a green border.

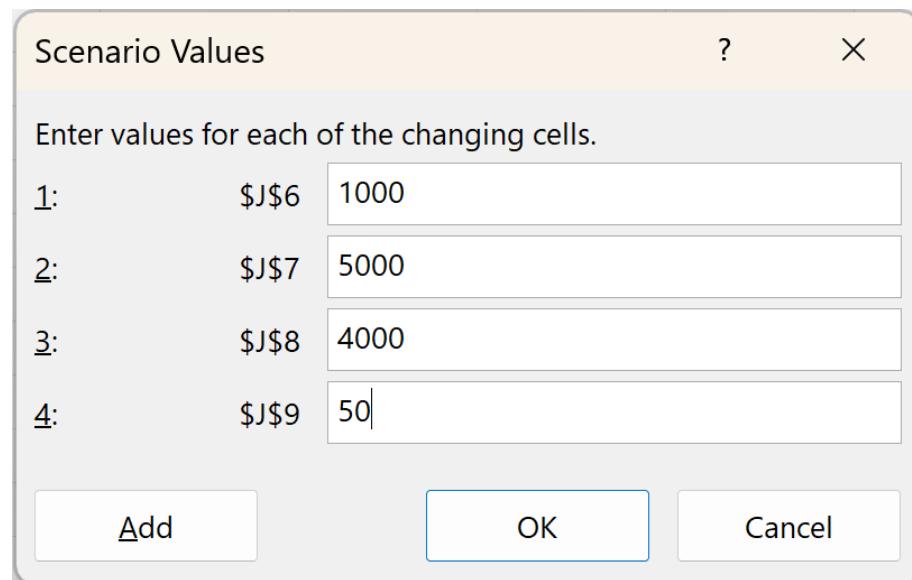
6. Click on 'Add' → write the following → Ok.



7. Click on 'Add' → write the following → Ok.



8. Click on 'Add' → write the following → Ok.



9. Select 'Present' → then 'Show'.

Scenarios:

- Present
- High
- Low

Changing cells: \$J\$6:\$J\$9

Comment: Created by Raj Nair on 02-02-2025

Add... Delete Edit... Merge... Summary...

Parts	Cost
Mouse	1100
Chip	5100
Monitor	4100
Keyboard	60
Total	10360

Show Close

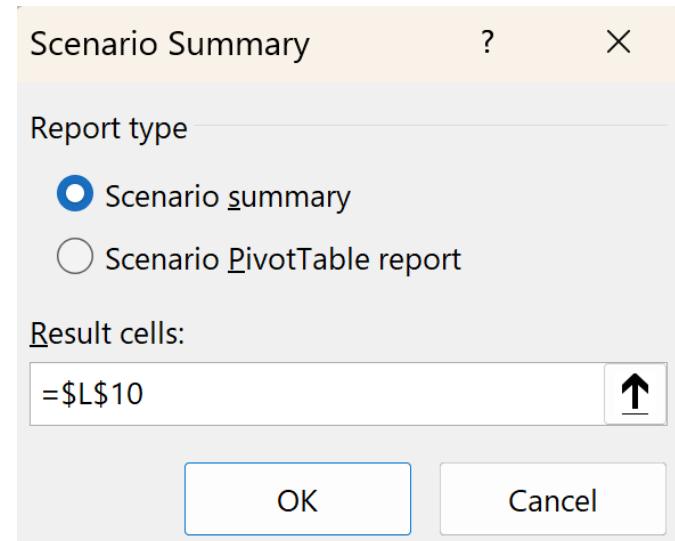
10. Select ‘High’ → then ‘Show’.

The screenshot shows the 'Scenario Manager' dialog box. In the 'Scenarios:' list, 'Present' is at the top, followed by 'High' which is highlighted with a blue selection bar, and 'Low' below it. To the right of the list are five buttons: 'Add...', 'Delete', 'Edit...', 'Merge...', and 'Summary...'. Below the list, there are two input fields: 'Changing cells:' containing '\$J\$6:\$J\$9' and 'Comment:' containing 'Created by Raj Nair on 02-02-2025'. At the bottom are two buttons: 'Show' and 'Close'. To the right of the dialog box, a portion of a spreadsheet is visible, showing a table with columns 'Parts' and 'Cost'. The rows include Mouse (1200), Chip (5200), Monitor (4300), Keyboard (80), and a summary row with 'Total' (10780) in bold.

11. Select ‘Low’ → then ‘Show’.

The screenshot shows the 'Scenario Manager' dialog box. In the 'Scenarios:' list, 'Present' is at the top, followed by 'High' and 'Low' which is highlighted with a blue selection bar. To the right of the list are five buttons: 'Add...', 'Delete', 'Edit...', 'Merge...', and 'Summary...'. Below the list, there are two input fields: 'Changing cells:' containing '\$J\$6:\$J\$9' and 'Comment:' containing 'Created by Raj Nair on 02-02-2025'. At the bottom are two buttons: 'Show' and 'Close'. To the right of the dialog box, a portion of a spreadsheet is visible, showing a table with columns 'Parts' and 'Cost'. The rows include Mouse (1000), Chip (5000), Monitor (4000), Keyboard (50), and a summary row with 'Total' (10050) in bold.

12. Select ‘Low’ → ‘Summary’ → then ‘Ok.



	A	B	C	D	E	F	G	H																																							
1																																															
2																																															
3																																															
4																																															
5																																															
6																																															
7																																															
8																																															
9																																															
10																																															
11																																															
12																																															
13																																															
14																																															
Scenario Summary																																															
<table border="1"> <thead> <tr> <th></th> <th>Current Values:</th> <th>Present</th> <th>High</th> <th>Low</th> </tr> </thead> <tbody> <tr> <td>Changing Cells:</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>\$J\$6</td> <td>1000</td> <td>1100</td> <td>1200</td> <td>1000</td> </tr> <tr> <td>\$J\$7</td> <td>5000</td> <td>5100</td> <td>5200</td> <td>5000</td> </tr> <tr> <td>\$J\$8</td> <td>4000</td> <td>4100</td> <td>4300</td> <td>4000</td> </tr> <tr> <td>\$J\$9</td> <td>50</td> <td>60</td> <td>80</td> <td>50</td> </tr> <tr> <td>Result Cells:</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>\$L\$10</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>									Current Values:	Present	High	Low	Changing Cells:					\$J\$6	1000	1100	1200	1000	\$J\$7	5000	5100	5200	5000	\$J\$8	4000	4100	4300	4000	\$J\$9	50	60	80	50	Result Cells:						\$L\$10			
	Current Values:	Present	High	Low																																											
Changing Cells:																																															
\$J\$6	1000	1100	1200	1000																																											
\$J\$7	5000	5100	5200	5000																																											
\$J\$8	4000	4100	4300	4000																																											
\$J\$9	50	60	80	50																																											
Result Cells:																																															
	\$L\$10																																														
<p>Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.</p>																																															

13. Prepare new data to perform What if Analysis (Data Table) → select the data from → ribbon go to ‘Data’ → select What if Analysis → ‘Data table’.

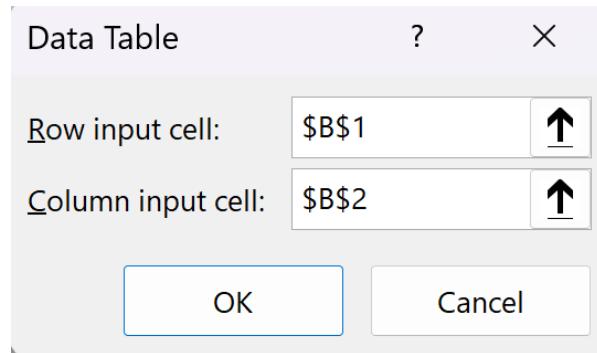
	A	B	C	D	E	F	G
1	Qty	500					
2	Price	5					
3	GM	60%					
4	Profit	1500					
5							
6							
7		Quantity					
8		1500	1500	300	400	500	600
9		3					
10		4					
11	Price	5					
12		6					
13		7					
14		8					
15							

The screenshot shows a Microsoft Excel interface with the following details:

- Excel Title Bar:** Shows 'AutoSave Off', 'excel_data_with_wh...', and a search bar.
- ribbon:** The 'Data' tab is selected.
- Context Menu:** An arrow points to the 'What-if Analysis' button in the Data Tools group, and a dropdown menu is open with 'Data Table...' highlighted.
- Worksheet Data:**

	A	B	C	D	E	F	G
1	Qty	500					
2	Price	5					
3	GM	60%					
4	Profit	1500					
5							
6							
7		Quantity					
8		1500	1500	300	400	500	600
9		3					
10		4					
11	Price	5					
12		6					
13		7					
14		8					
15							

14. In the Row Input Cell, select B1 (Qty) & the Column Input Cell, select B2 (Price).

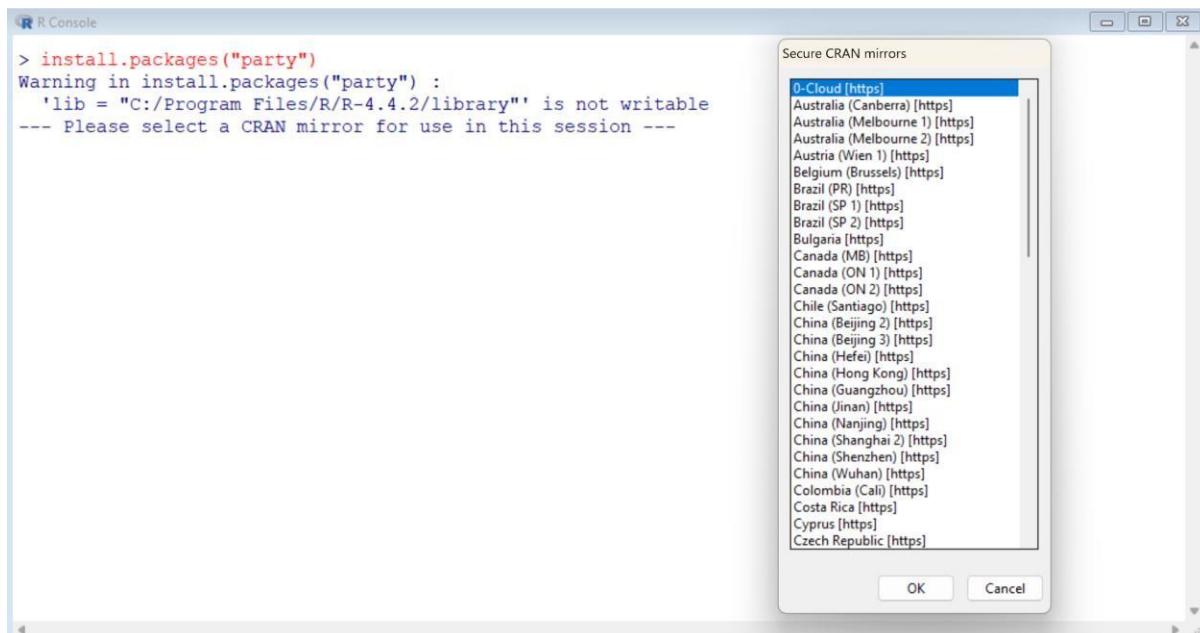


15. Click **OK**, and Excel will calculate the profit for different quantity and price combinations.

		Quantity				
	1500	1500	300	400	500	600
Price	3	2700	540	720	900	1080
4	3600	720	960	1200	1440	
5	4500	900	1200	1500	1800	
6	5400	1080	1440	1800	2160	
7	6300	1260	1680	2100	2520	
8	7200	1440	1920	2400	2880	

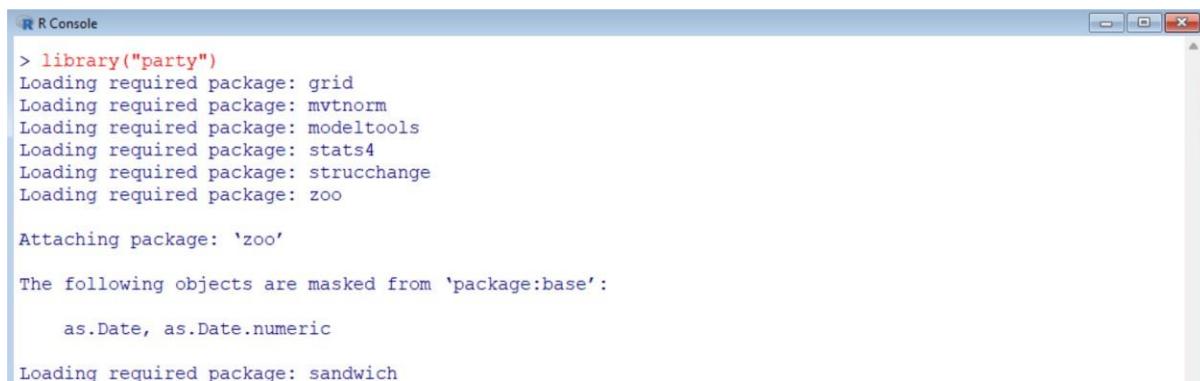
Practical No. 3**Perform the data classification using classification algorithm using R/Python****Code: -**

>> install.packages("party") → Select 0-Cloud



The downloaded binary packages are in
C:\Users\rajn2\AppData\Local\Temp\RtmpwNlnBO\downloaded_packages

>> library("party")



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

>> png(file="decision_tree.png")

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

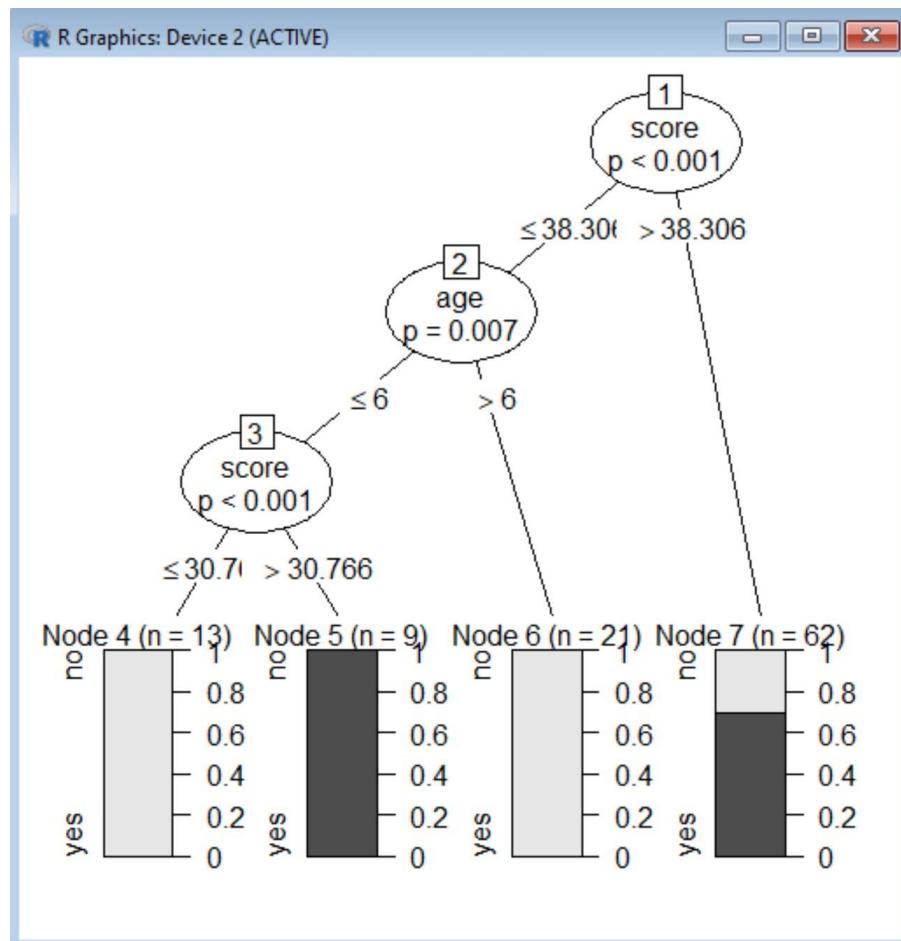
>> plot(output.tree)

>> dev.off()
```

```
> dev.off()
null device
1
```

Note: - Re-plot it again i.e. run the following line again,

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



Practical No. 4

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

Code: -

```
>> newiris <- iris
```

```
>> print(newiris)
```

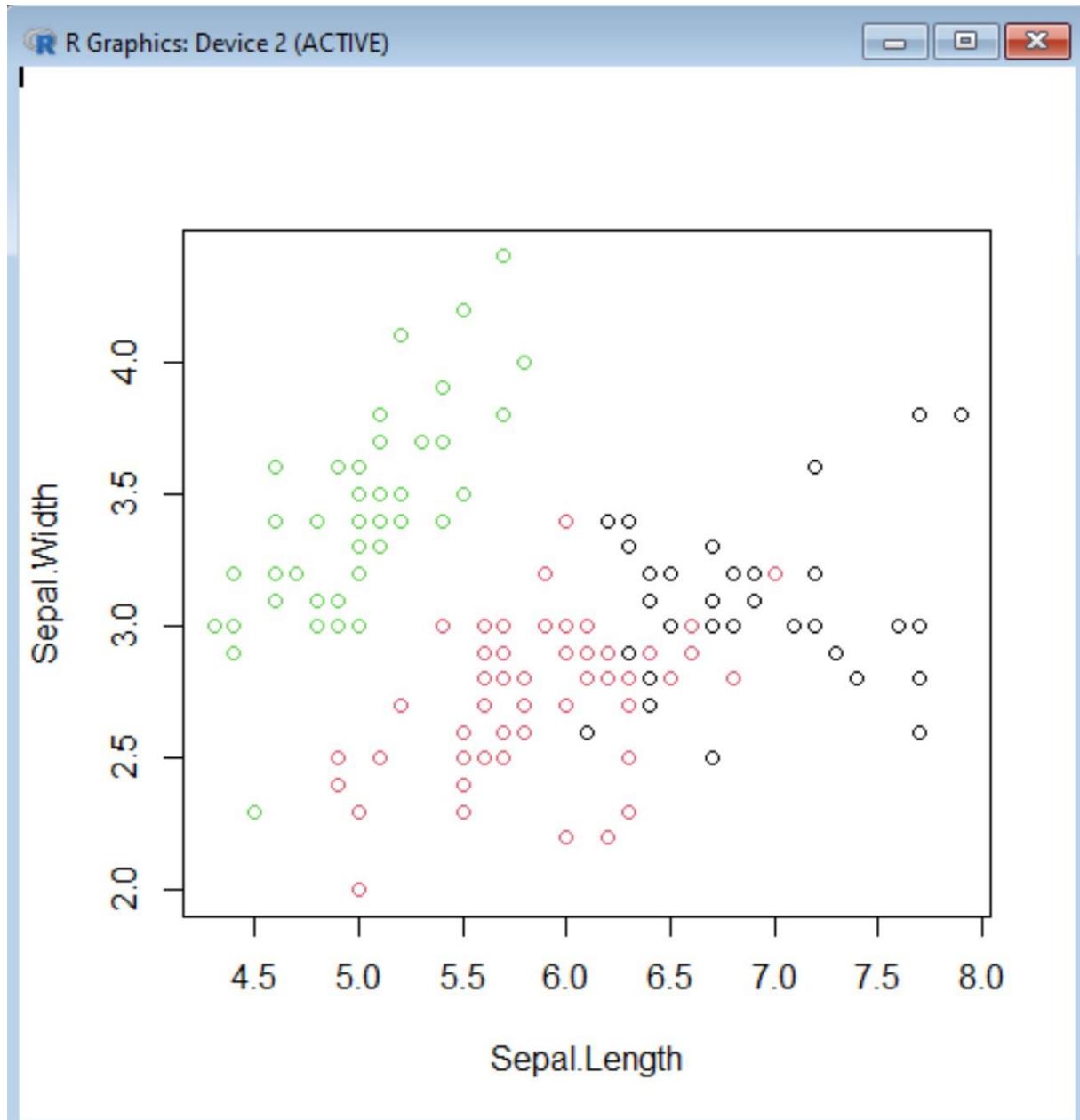
```
R Console
```

```
> newiris <- iris
> print(newiris)
   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
```

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

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

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



Practical No. 5

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

Code: -

```
>> x <- c(100,110,130,143,156,170)
>> y <- c(50,55,60,72,66,89)
>> r <- lm(y ~ x)
>> print(r)
```

```
R Console
> x <- c(100,110,130,143,156,170)
> y <- c(50,55,60,72,66,89)
> r <- lm(y ~ x)
> print(r)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)           x
1.0991          0.4764
```

```
>> print(summary(r))
```

```
R Console
> print(summary(r))

Call:
lm(formula = y ~ x)

Residuals:
    1     2     3     4     5     6 
1.261 1.497 -3.031 2.776 -9.417 6.913 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.0991    14.3091   0.077   0.9425    
x            0.4764     0.1044   4.563   0.0103 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

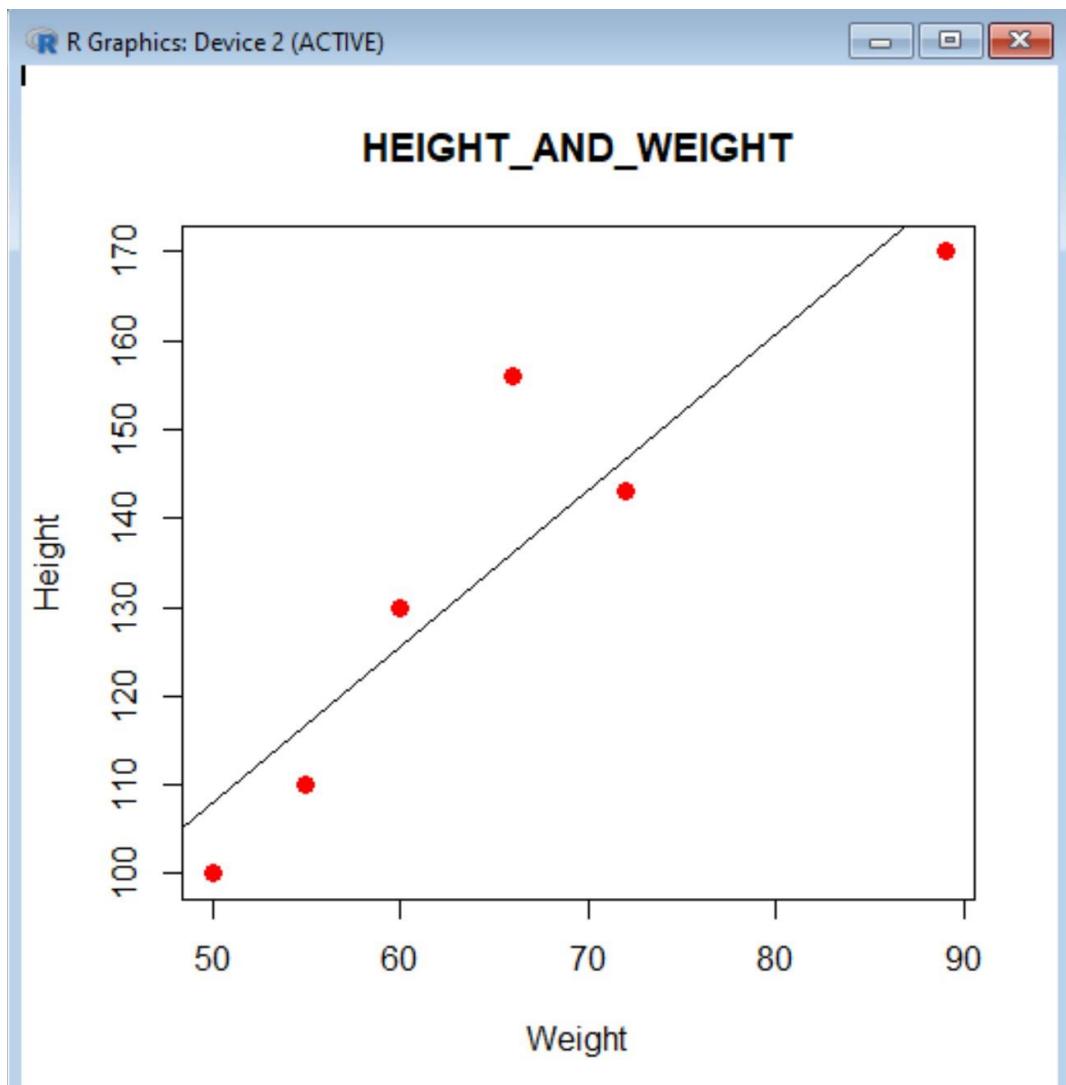
Residual standard error: 6.269 on 4 degrees of freedom
Multiple R-squared:  0.8388,    Adjusted R-squared:  0.7985 
F-statistic: 20.82 on 1 and 4 DF,  p-value: 0.01032
```

```
>> a <- data.frame(x=130)
>> res <- predict(r,a)
>> print(res)
```



```
R Console
> a <- data.frame(x=130)
> res <- predict(r,a)
> print(res)
1
63.03075
```

```
>> plot(y,x,col="red",main =
"HEIGHT_AND_WEIGHT",abline(lm(x~y)),cex=1.3,pch=16,xlab="Weight",ylab="Height")
```



Practical No. 6**Perform the logistic regression on the given data warehouse data using R/Python****Code: -**

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, roc_auc_score

# Generate dataset and split into train/test sets
X, y = make_classification(n_samples=1000, n_classes=2, random_state=1)
trainX, testX, trainy, testy = train_test_split(X, y, test_size=0.5, random_state=2)

# Fit logistic regression model
model = LogisticRegression(solver='lbfgs').fit(trainX, trainy)

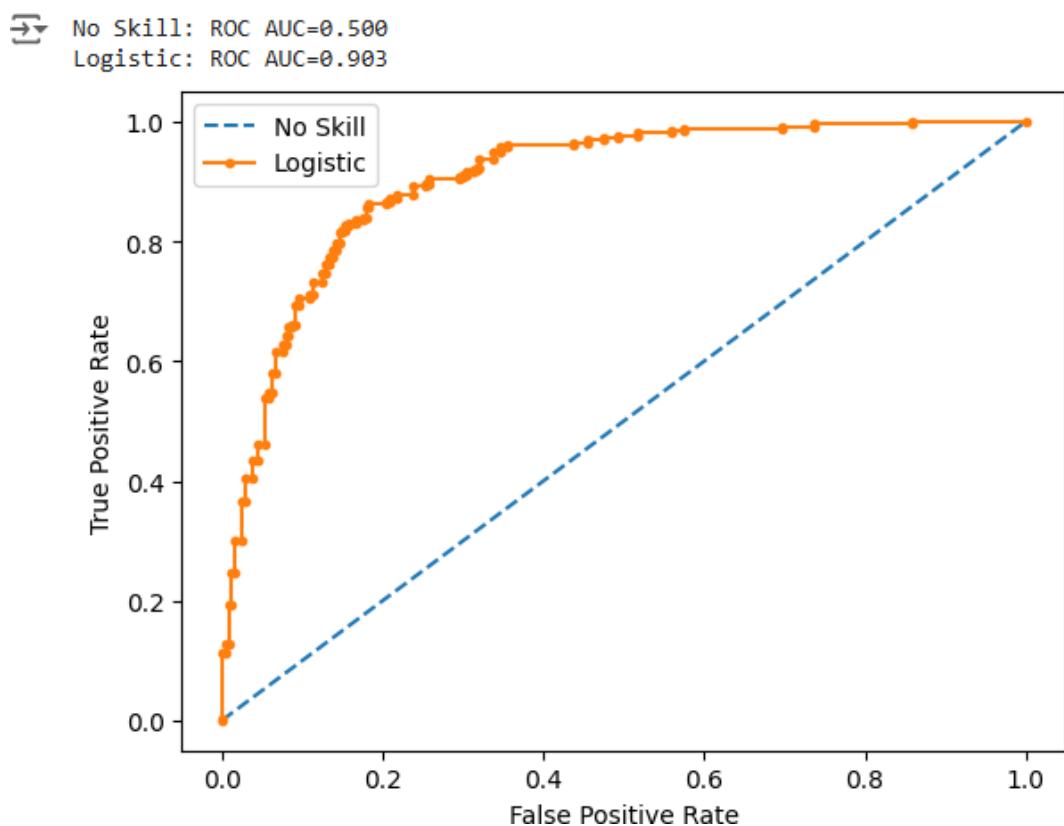
# Predict probabilities
lr_probs = model.predict_proba(testX)[:, 1]
ns_probs = [0] * len(testy) # No skill model

# Calculate AUC scores
ns_auc = roc_auc_score(testy, ns_probs)
lr_auc = roc_auc_score(testy, lr_probs)

# Print AUC scores
print(f'No Skill: ROC AUC={ns_auc:.3f}')
print(f'Logistic: ROC AUC={lr_auc:.3f}')

# Calculate ROC curves
ns_fpr, ns_tpr, _ = roc_curve(testy, ns_probs)
lr_fpr, lr_tpr, _ = roc_curve(testy, lr_probs)
```

```
# Plot ROC curves  
plt.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')  
plt.plot(lr_fpr, lr_tpr, marker='.', label='Logistic')  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
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)

Code: -

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.cluster import KMeans

dataset = pd.read_csv("Mall_Customers.csv")

x = dataset.iloc[:, [3, 4]].values

kmeans = KMeans(n_clusters=5, init='k-means++', max_iter=300, n_init=10,
random_state=0)

y_kmeans = kmeans.fit_predict(x)

colors = ['red', 'blue', 'green', 'cyan', 'magenta']

for i in range(5):

    plt.scatter(x[y_kmeans == i, 0], x[y_kmeans == i, 1], s=100, c=colors[i], label=f'Cluster
{i+1}'')

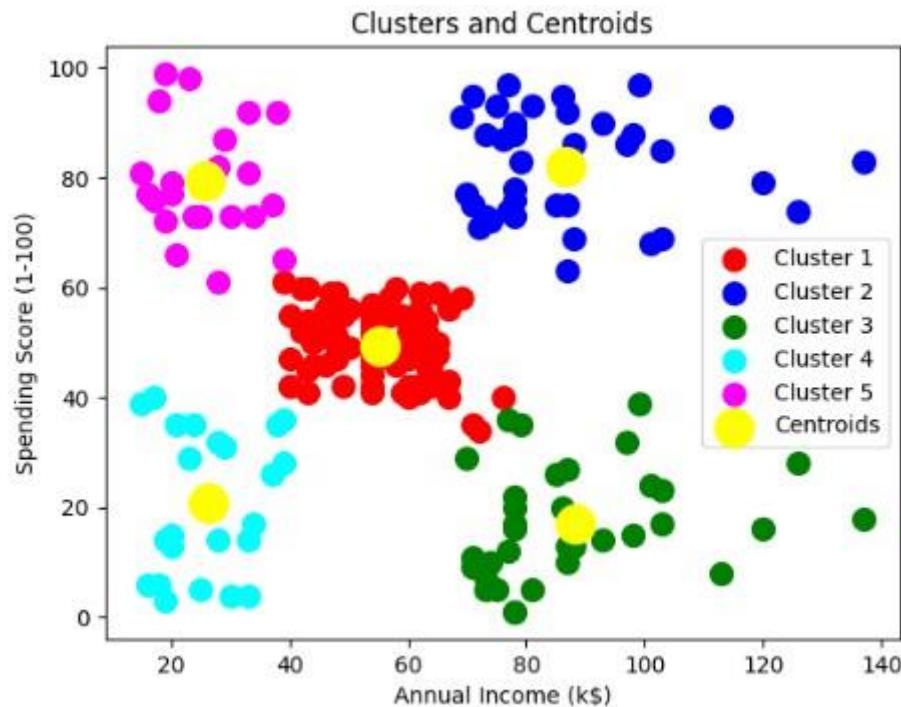
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='yellow',
label='Centroids')

plt.title("Clusters and Centroids")

plt.xlabel("Annual Income (k$)")

plt.ylabel("Spending Score (1-100)")

plt.legend() plt.show()
```

Output:**KNN**

```

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

X, y = load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
print(f'Accuracy: {accuracy_score(y_test, y_pred):.2f}')

```

Output:

Accuracy: 1.00

Practical No. 8
Perform data visualization

- a. Perform data visualization using Python on any sales data.

1. create CSV file and upload in Collab.

Date,Sales
2025-01-01,150
2025-01-02,200
2025-01-03,180
1/4/2025,220
1/5/2025,190
2025-01-06,250
2025-01-07,230
1/8/2025,270
1/9/2025,210
2025-01-10,280

2. Open collab & push the csv file.

pip install pandas matplotlib seaborn

3. Code:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the data from the CSV file (with correct path)

df = pd.read_csv('/content/Book1.csv')

df.columns = df.columns.str.strip()
```

Plot

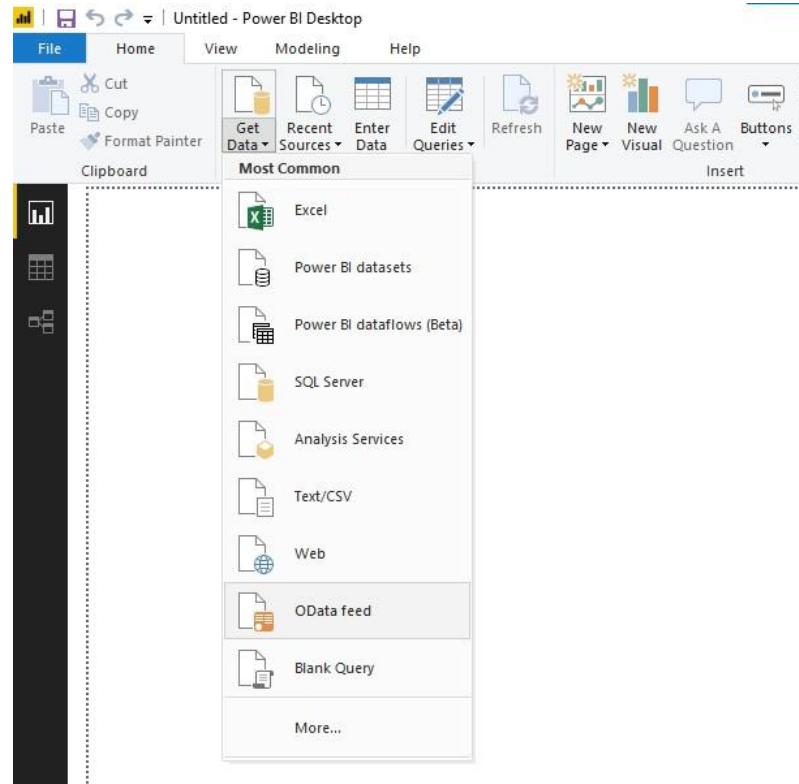
```
sns.lineplot(data=df, x='Date', y='Sales', marker='o')
plt.title('Sales Over Time')
plt.xticks(rotation=45)
plt.show()
```

Output:

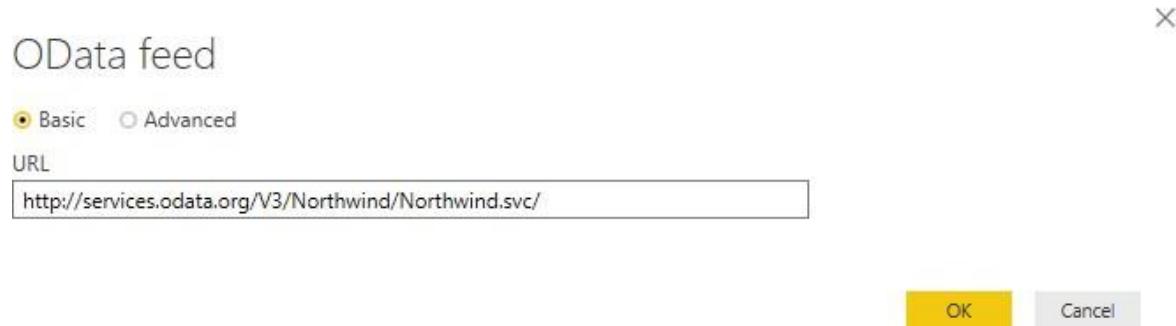


b. Perform data visualization using PowerBI on any sales data.

1. Open Power BI Desktop → Go to Get Data → OData feed.



2. Give URL :- <http://services.odata.org/V3/Northwind/Northwind.svc/> → click OK

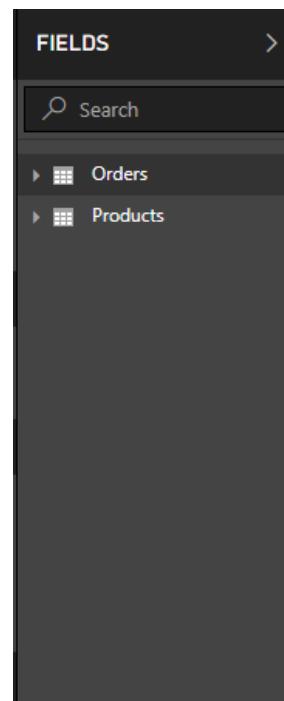


3. Select Tables Orders & Products → Load.

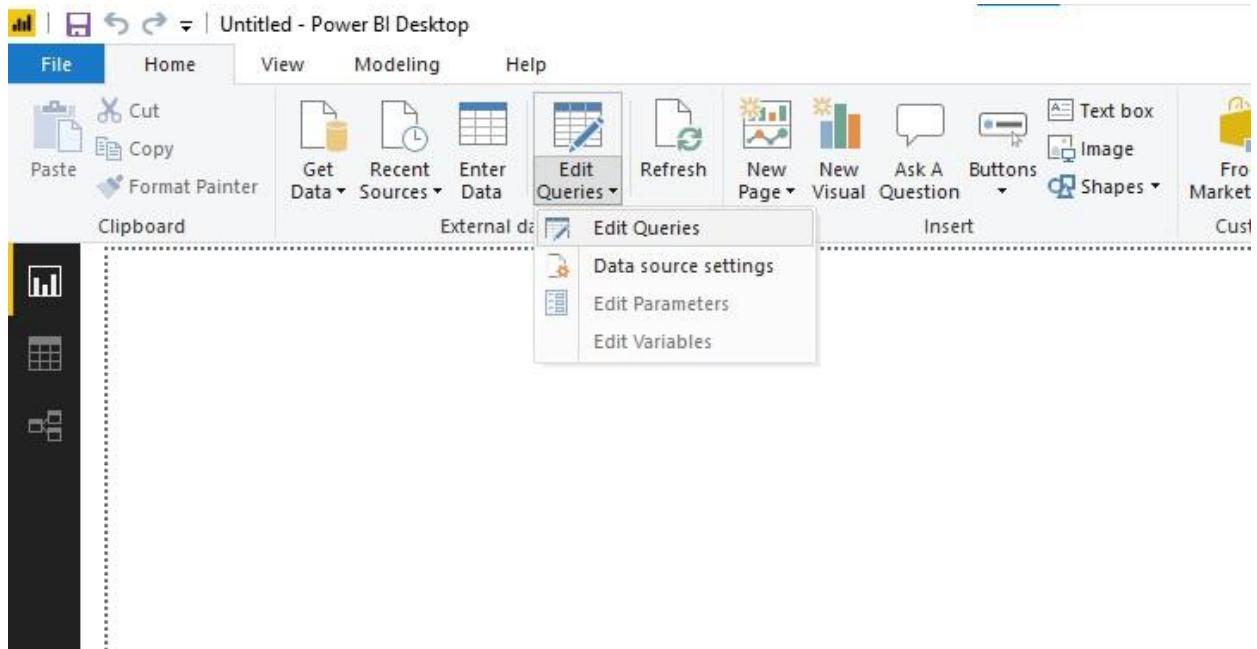
Navigator

The screenshot shows the 'Navigator' window with the title 'Products'. It displays a preview of data downloaded on Saturday, January 18, 2025. On the left, a tree view lists various tables from the Northwind database, with 'Orders' and 'Products' selected. On the right, a grid shows the first 22 products from the 'Products' table. At the bottom, there are buttons for 'Select Related Tables', 'Load' (highlighted in yellow), 'Edit', and 'Cancel'.

ProductID	ProductName	SupplierID	CategoryID	Quan
1	Chai	1	1	10
2	Chang	1	1	2
3	Aniseed Syrup	1	2	1
4	Chef Anton's Cajun Seasoning	2	2	48
5	Chef Anton's Gumbo Mix	2	2	36
6	Grandma's Boysenberry Spread	3	2	12
7	Uncle Bob's Organic Dried Pears	3	7	12
8	Northwoods Cranberry Sauce	3	2	12
9	Mishi Kobe Niku	4	6	18
10	Ikura	4	8	1
11	Queso Cabrales	5	4	1
12	Queso Manchego La Pastora	5	4	10
13	Konbu	6	8	2
14	Tofu	6	7	40
15	Genen Shouyu	6	2	24
16	Pavlova	7	3	32
17	Alice Mutton	7	6	20
18	Carnarvon Tigers	7	8	16
19	Teatime Chocolate Biscuits	8	3	10
20	Sir Rodney's Marmalade	8	3	30
21	Sir Rodney's Scones	8	3	24
22	Gustaf's Knäckebröd	9	5	24



4. Go to Edit Queries → select Edit Queries.



5. In Edit Queries select ProductID, QuantityForUnit, UnitPrice, UnitInStock → using Ctrl key → Right Click on on columns and Remove Other Columns.

The screenshot shows the Power Query Editor interface. The 'Edit Queries' step is selected in the ribbon. A context menu is open over a table preview, specifically over the 'QuantityPerUnit' column. The menu path 'Remove Other Columns' is highlighted. The table preview shows data from the 'Products' table, including columns like ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, and UnitInStock. The status bar at the bottom indicates '13 COLUMNS, 77 ROWS'.

	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitInStock
1	Chai	1	1	10 boxes x 20 bags	18	39
2	Chang	1	1	24 - 12 oz bottles	19	45
3	Aniseed Syrup	1	2	2 12 - 550 ml bottles	10	13
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22	15
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35	5
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25	15
7	Uncle Bob's Organic Dried Pears	3	2	12 - 1 lb pkgs.	30	12
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40	60
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	50
10	Ikura	4	8	12 - 200 ml jars	31	40
11	Queso Cabrales	5	4	1 kg pkg.	21	10
12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38	20
13	Konbu	6	8	2 kg box	6	10
14	Tofu	6	7	40 - 100 g pkgs.	23.25	15
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	10
16	Pavlova	7	3	32 - 500 g boxes	17.45	5
17	Alice Mutton	7	6	20 - 1 kg tins	39	10
18	Carnarvon Tigers	7	8	16 kg pkg.	62.5	42
19	Teatime Chocolate Biscuits	8	3	10 boxes x 12 pieces	9.2	25
20	Sir Rodney's Marmalade	8	3	30 gift boxes	81	40
21	Sir Rodney's Scones	8	3	24 pkgs x 4 pieces	10	3
22	Gustaf's Knäckebrot	9	5	24 - 500 g pkgs.	21	104
23	Tunnbröd	9	5	12 - 250 g pkgs.	9	61
24						

6. Go to Replace Values and make changes in number.

The screenshot shows the Power Query Editor interface. At the top, there's a ribbon with tabs like 'Enter Data', 'Data source settings', 'Manage Parameters', 'Properties', 'Advanced Editor', 'Query', 'Manage Columns', 'Reduce Rows', 'Sort', 'Split Column', 'Group By', and 'Transform'. Below the ribbon is a table with 15 rows of data. The columns are labeled 'ProductID' (containing values 1 through 15), 'QuantityPerUnit' (containing descriptions like '10 boxes x 20 bags'), 'UnitPrice' (containing values like 18, 19, 21.35, etc.), and 'UnitsInStock' (containing values like 39, 17, 0, etc.).

ProductID	QuantityPerUnit	UnitPrice	UnitsInStock
1	10 boxes x 20 bags	18	39
2	24 - 12 oz bottles	19	17
3	3 12 - 550 ml bottles	10	13
4	4 48 - 6 oz jars	22	53
5	5 36 boxes	21.35	0
6	6 12 - 8 oz jars	25	120
7	7 12 - 1 lb pkgs.	30	15
8	8 12 - 12 oz jars	40	6
9	9 18 - 500 g pkgs.	97	29
10	10 12 - 200 ml jars	31	31
11	11 1 kg pkg.	21	22
12	12 10 - 500 g pkgs.	38	86
13	13 2 kg box	6	24
14	14 40 - 100 g pkgs.	23.25	35
15	15 24 - 250 ml bottles	15.5	39

Replace Values

Replace one value with another in the selected columns.

Value To Find

Replace With

OK

Cancel

7. To Add new Column → Go to Add Column Tab select Custom Column.

8. Give Column Name → Select [UnitPrice]*[UnitInStock] By inserting columns.

New column name
Custom

Custom column formula:
= [UnitPrice]*[UnitsInStock]

Available columns:

- ProductID
- QuantityPerUnit
- UnitPrice
- UnitsInStock

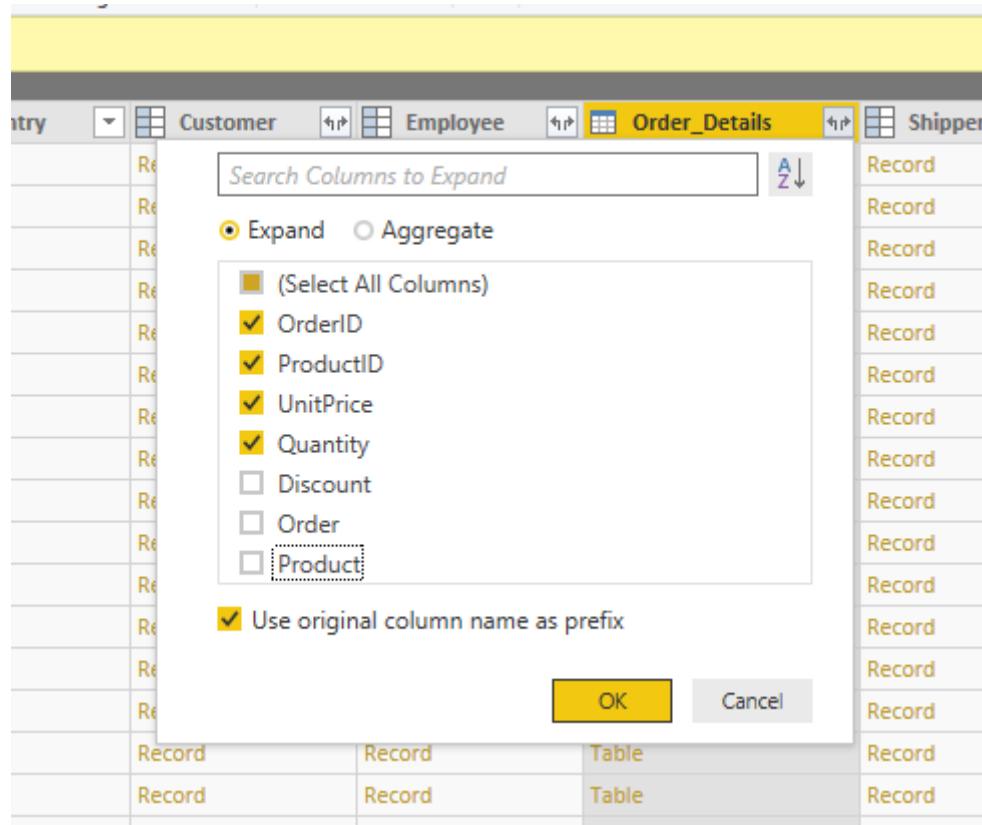
<< Insert

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

OK Cancel

9. Now in Order Table Go to Order_Details column and select the columns given below
 → then OK.



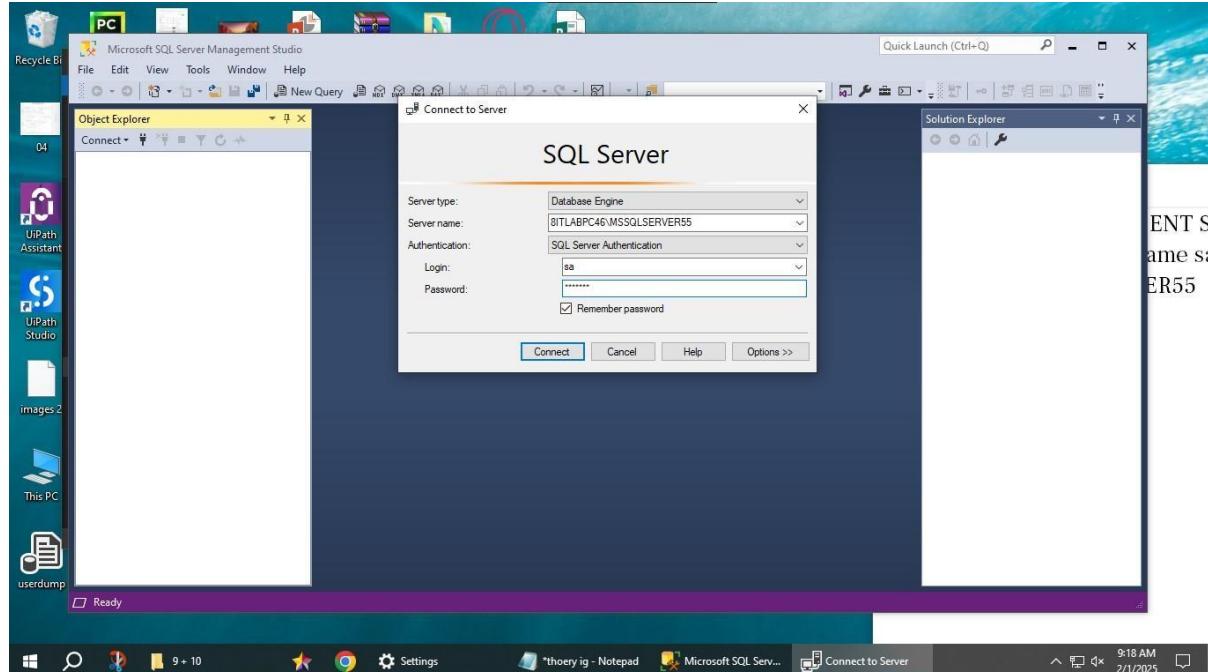
10. Select the columns Order_Details.OrderId, Order_Details.ProductId,
 Order_Details.UnitPrice and Order_Details.Quantity -> Right click on the column and
 select Remove Other Columns.

1. Order_Details.OrderId	2. Order_Details.ProductId	3. Order_Details.UnitPrice	4. Order_Details.Quantity
10248	11	14	
10248	42	9.8	
10248	72	34.8	
10249	14	18.6	
10249	51	42.4	
10250	41	7.7	
10250	51	42.4	
10250	65	16.8	
10251	22	16.8	
10251	57	15.6	
10251	65	16.8	
10252	20	64.8	
10252	33	2	
10252	60	27.2	
10253	31	10	
10253	39	14.4	
10253	49	16	
10254	24	3.6	
10254	55	19.2	
10254	74	8	
10255	2	15.2	
10255	16	13.9	
10255	36	15.2	

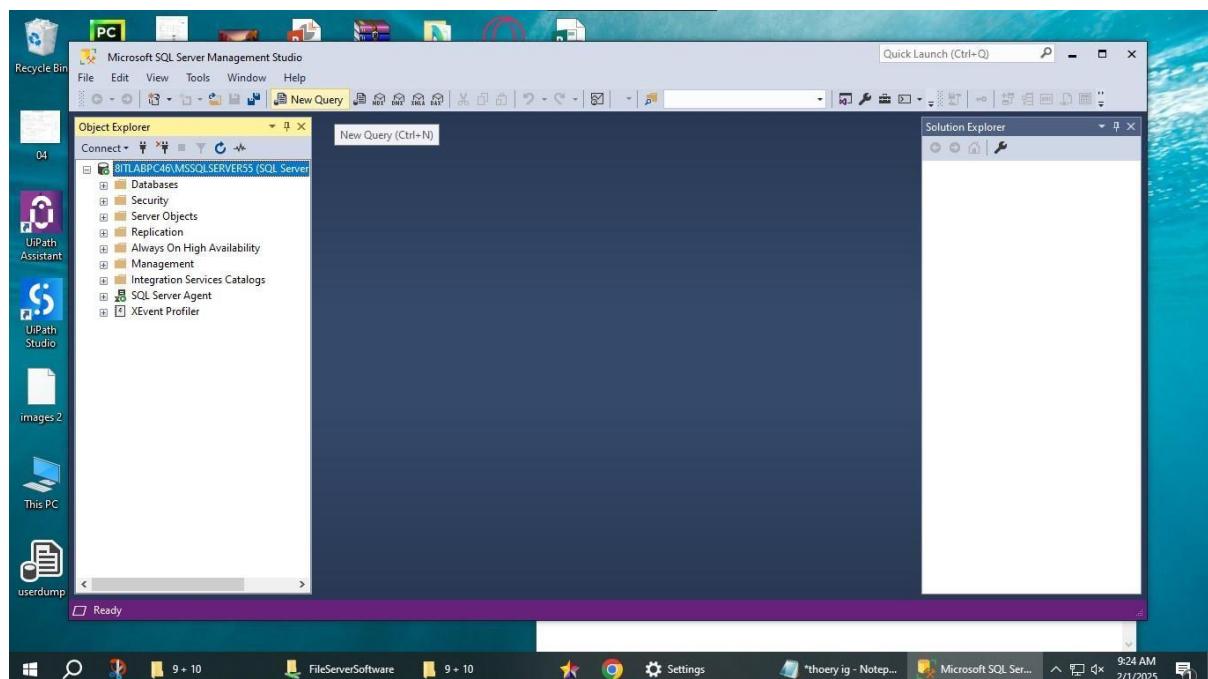
Practical No. 9

Create the Data staging area for the selected database using SQL

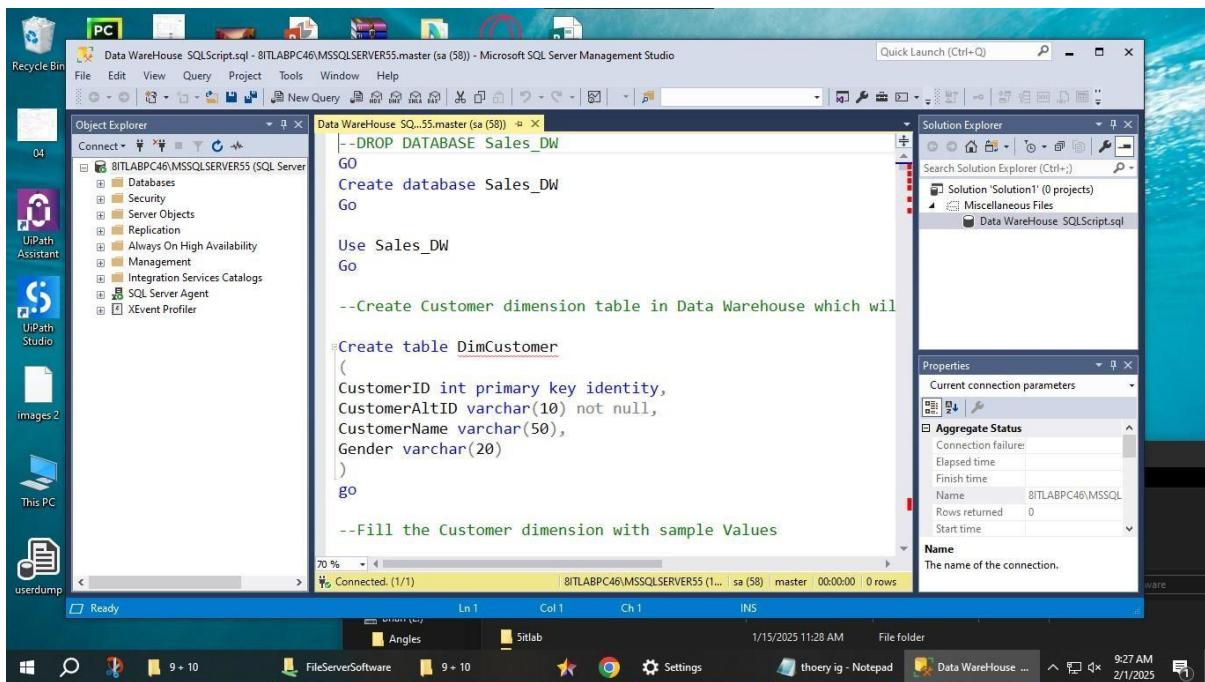
1. Open SQL Server Management Studio (SSMS).



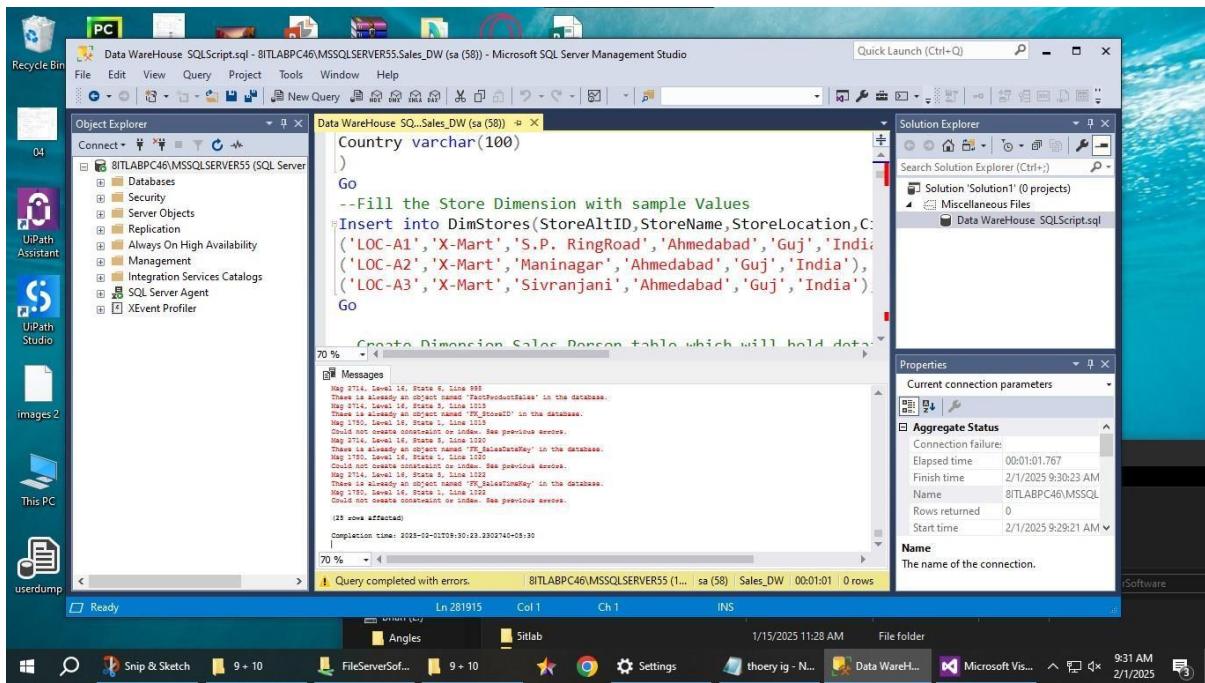
2. Connect to the database engine using 8ITLABPC46\MSSQLSERVER55, SQL Server authentication (sa/sms@123).



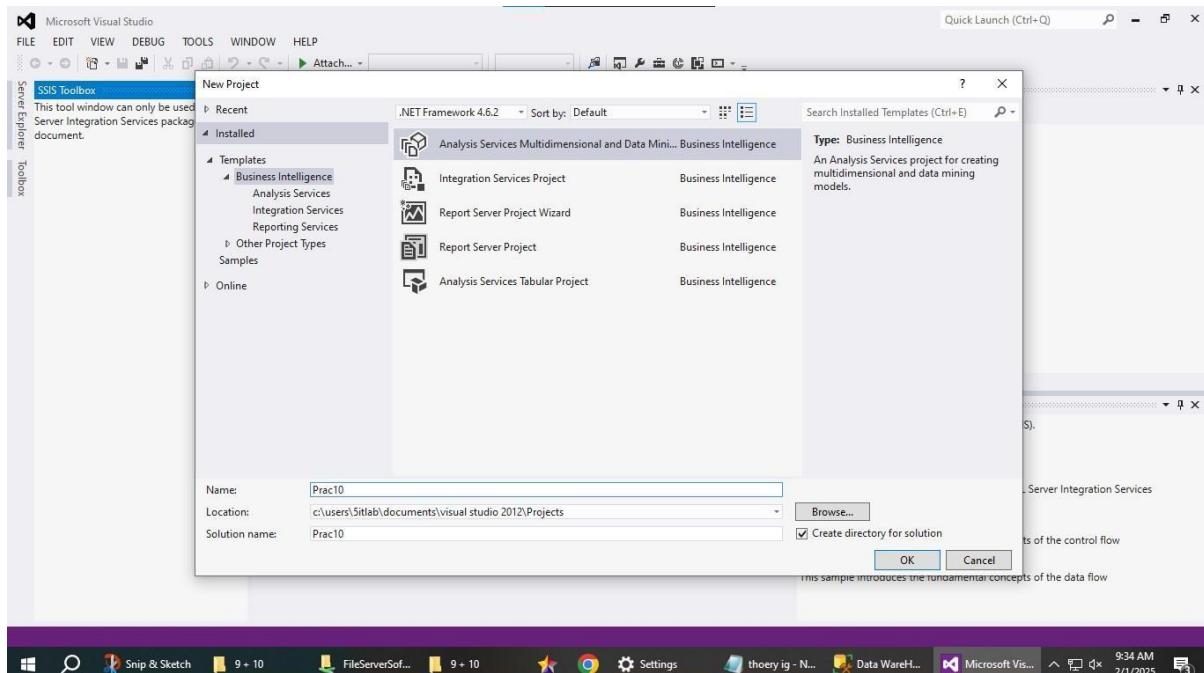
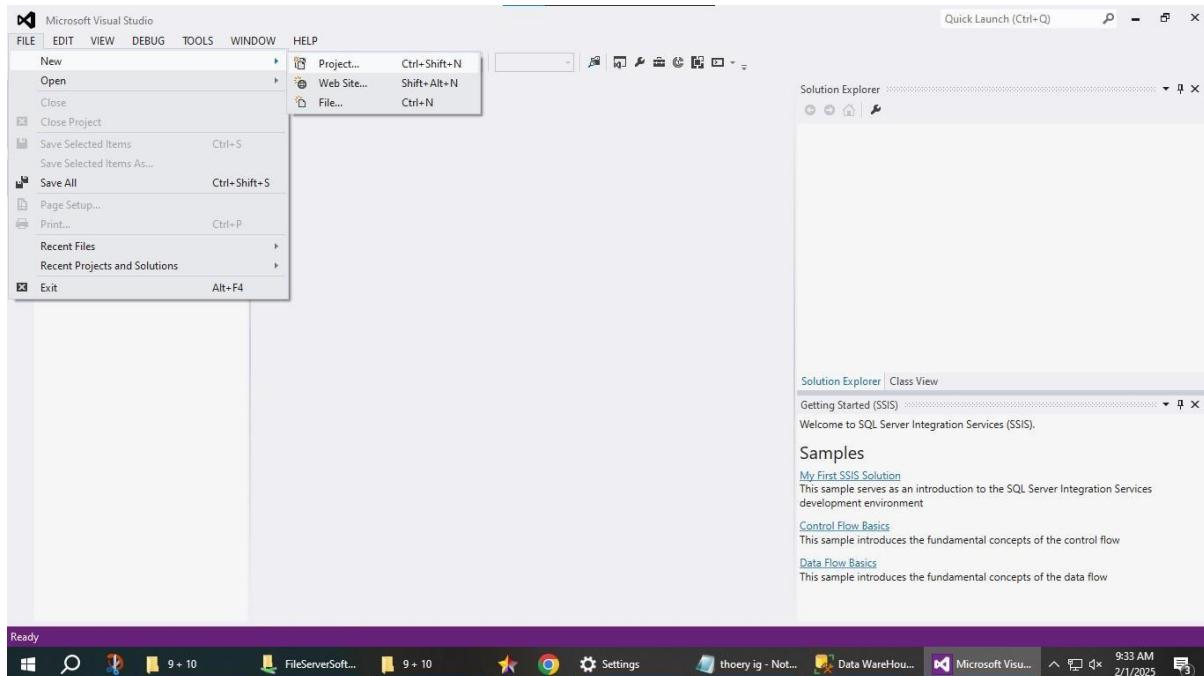
3. Navigate to Run > \\...98\file server\docs\bsql script file.



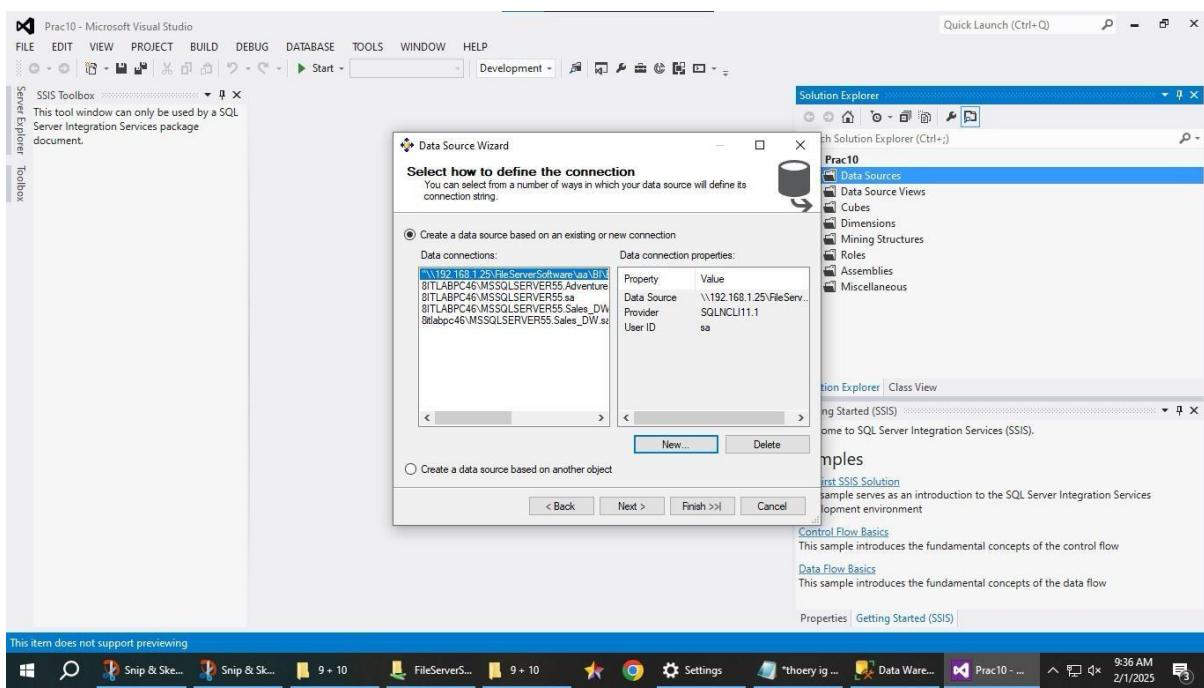
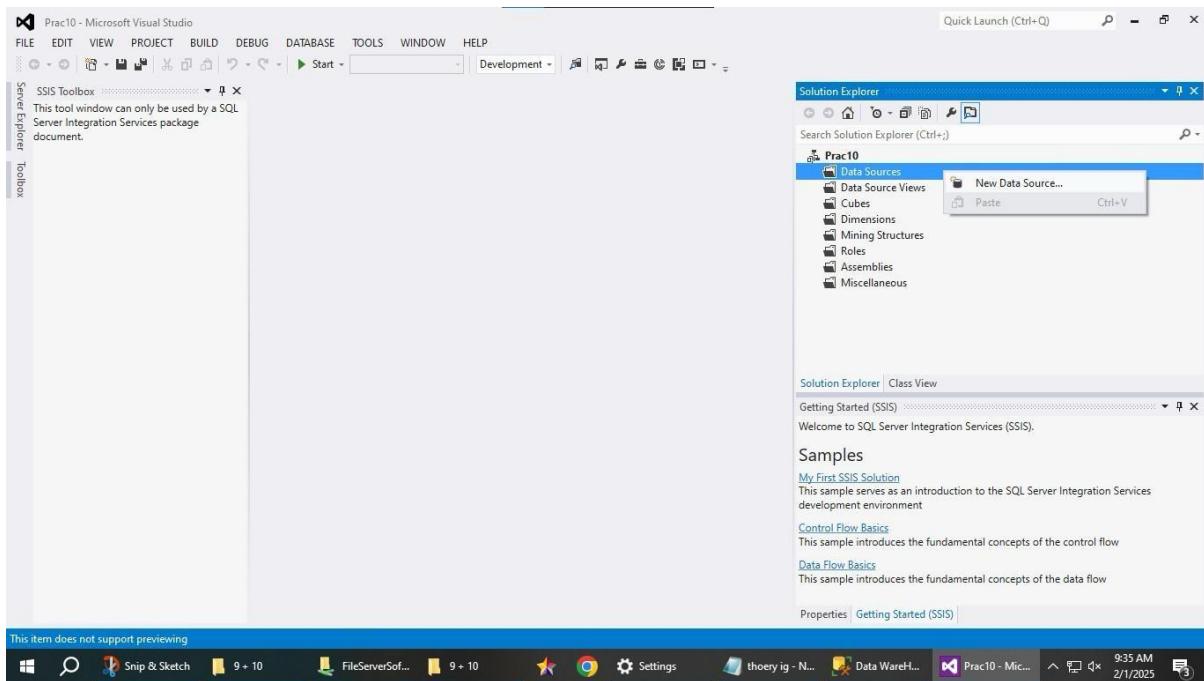
4. Open a new query, double-click the DW script file, right-click, and execute (errors may appear if already executed).



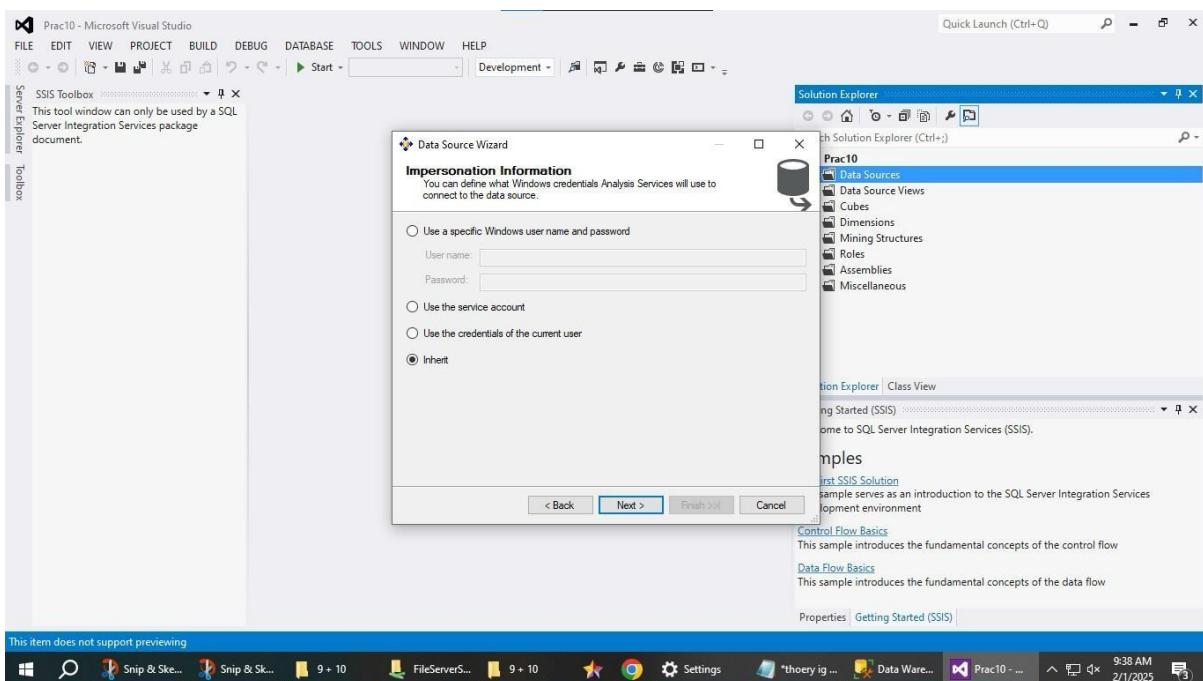
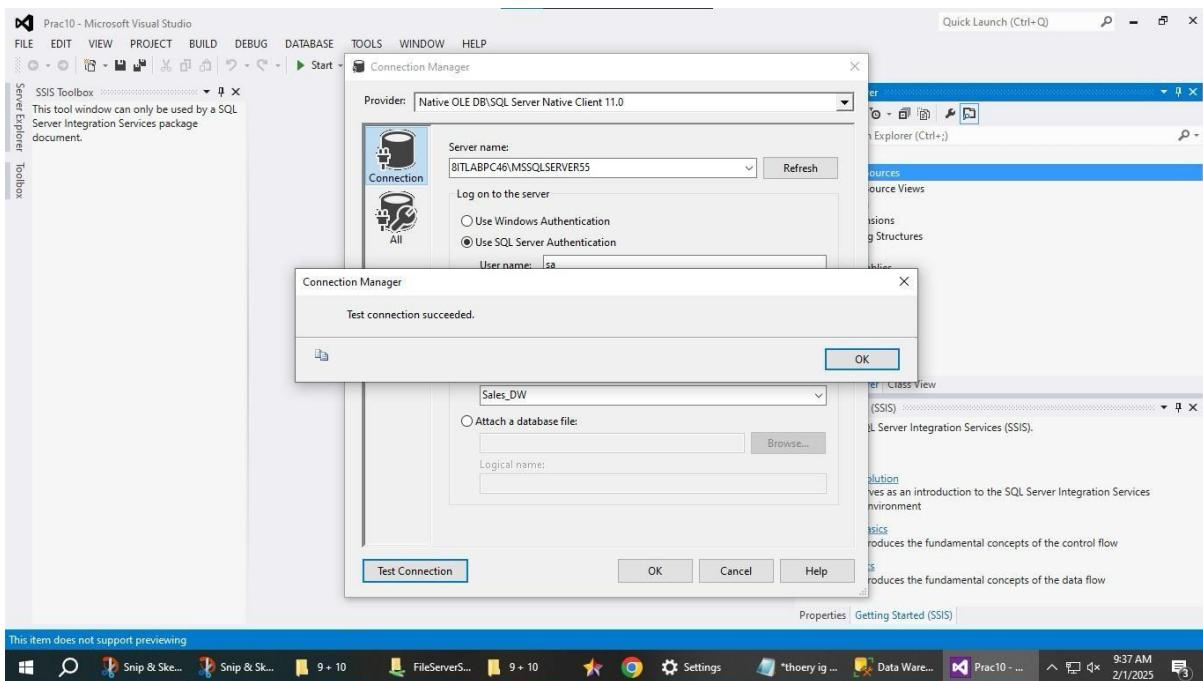
5. Open SQL Server Data Tools (SSDT) and create a new project: Analysis Services Multidimensional > "Prac10". If Solution Explorer is missing, go to Window > Reset Layout.



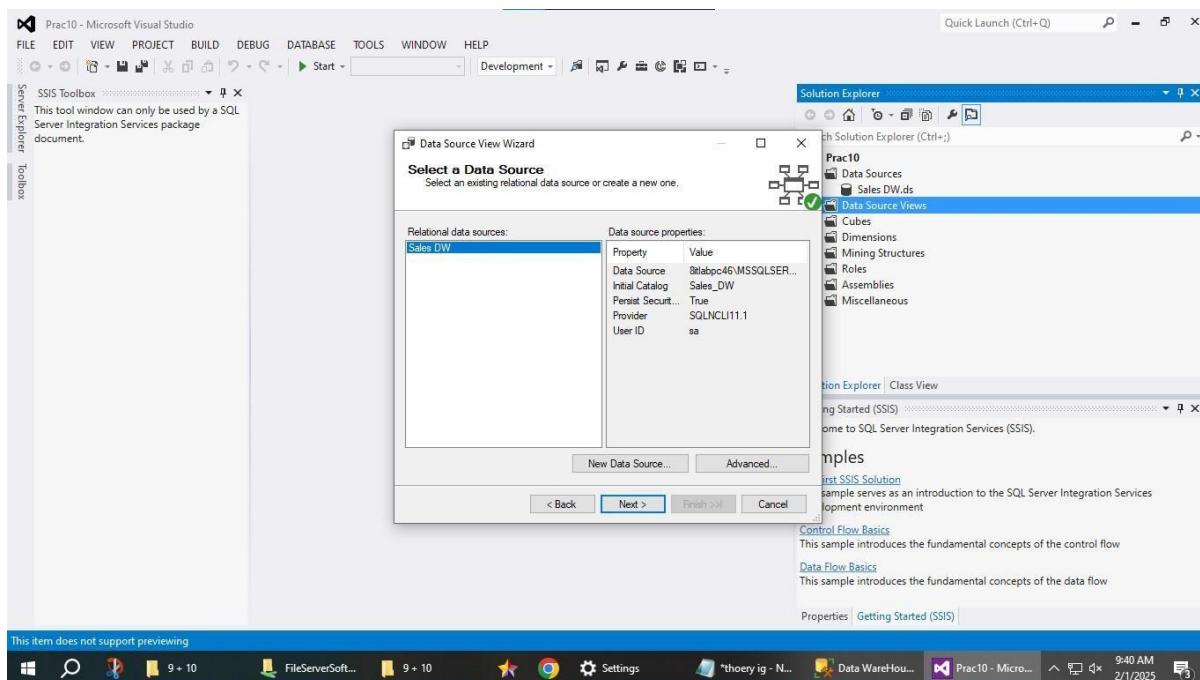
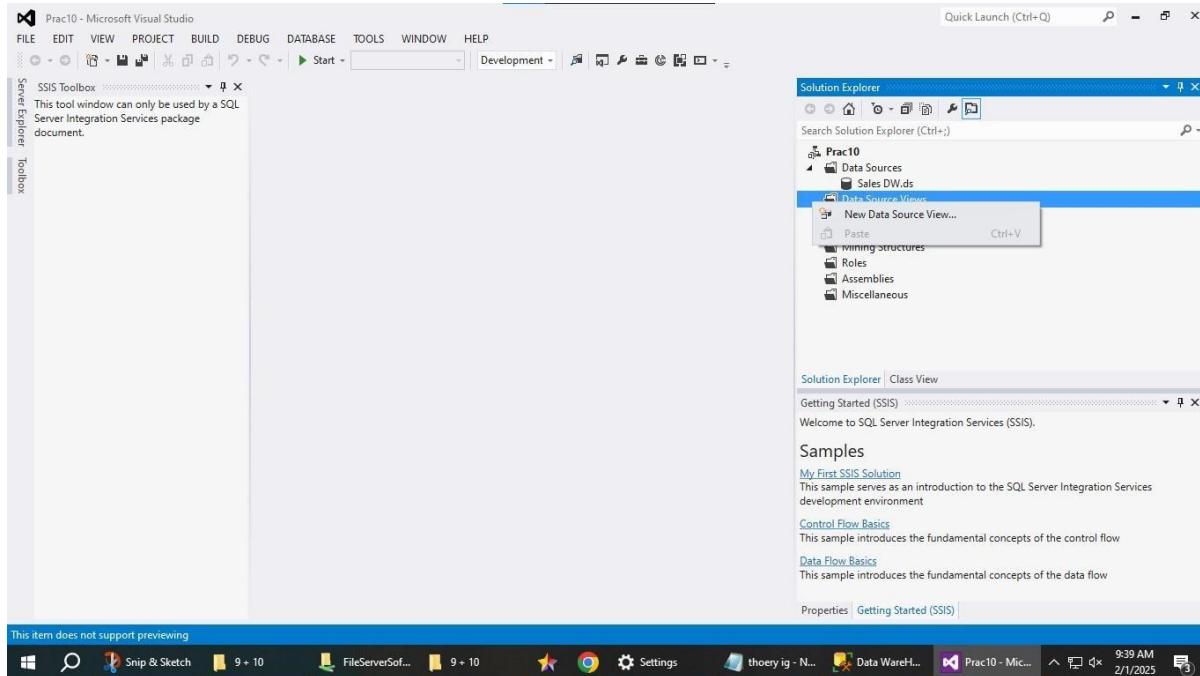
6. Create a new Data Source, using the copied server name, same credentials, select Sales DW, test connection, inherit settings, and finish.



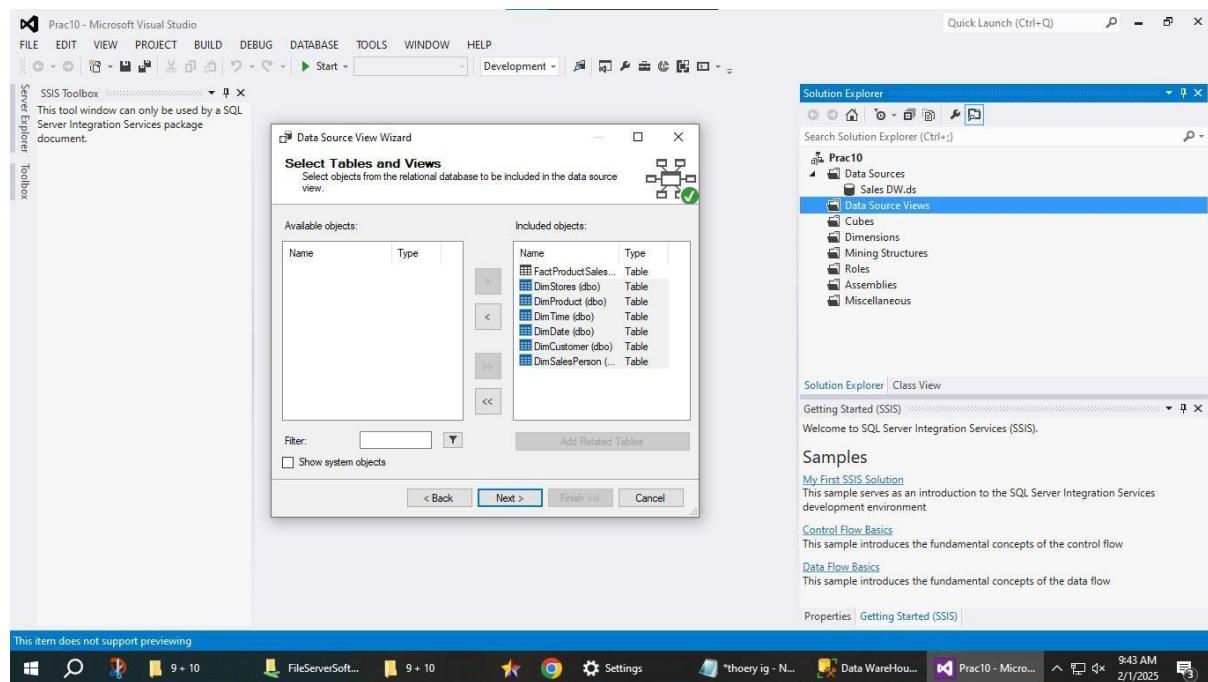
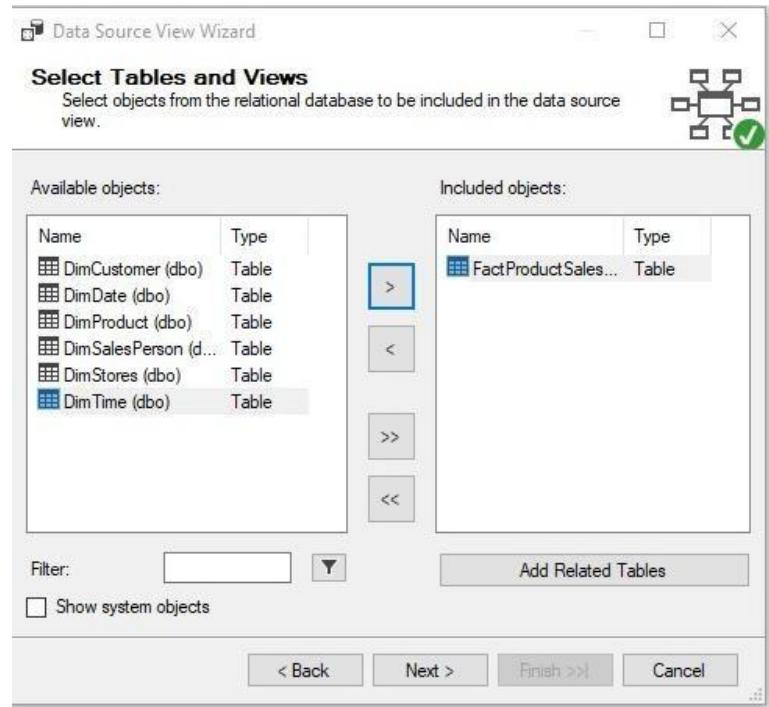
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



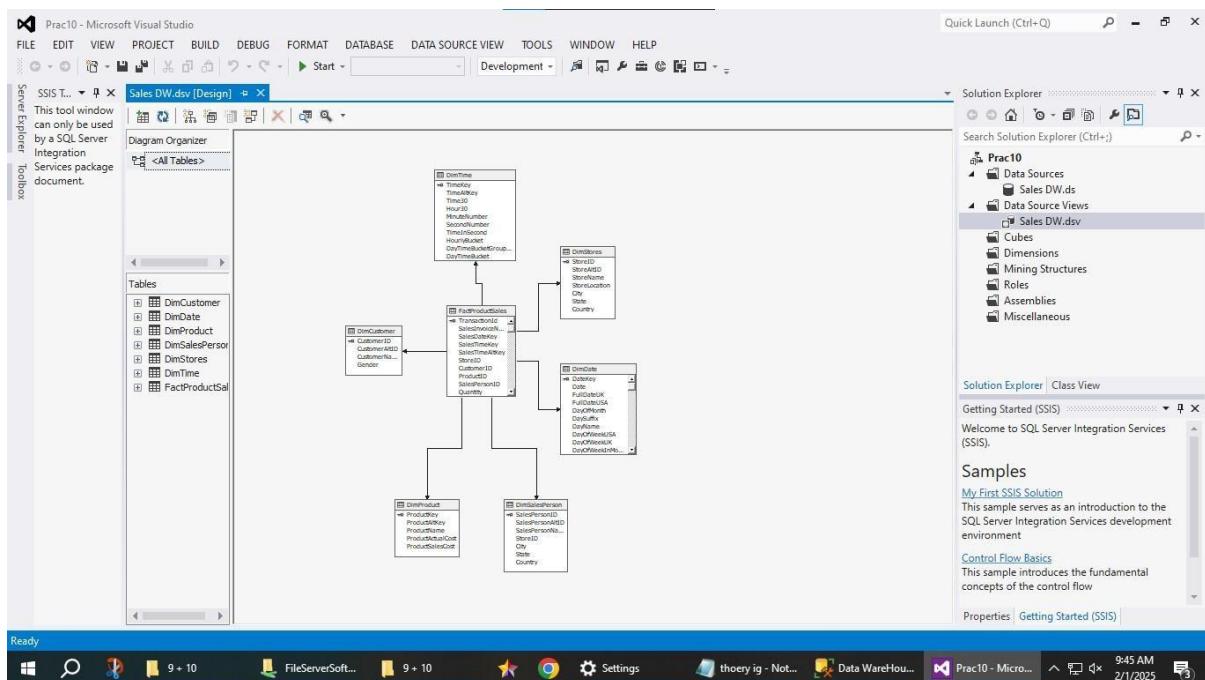
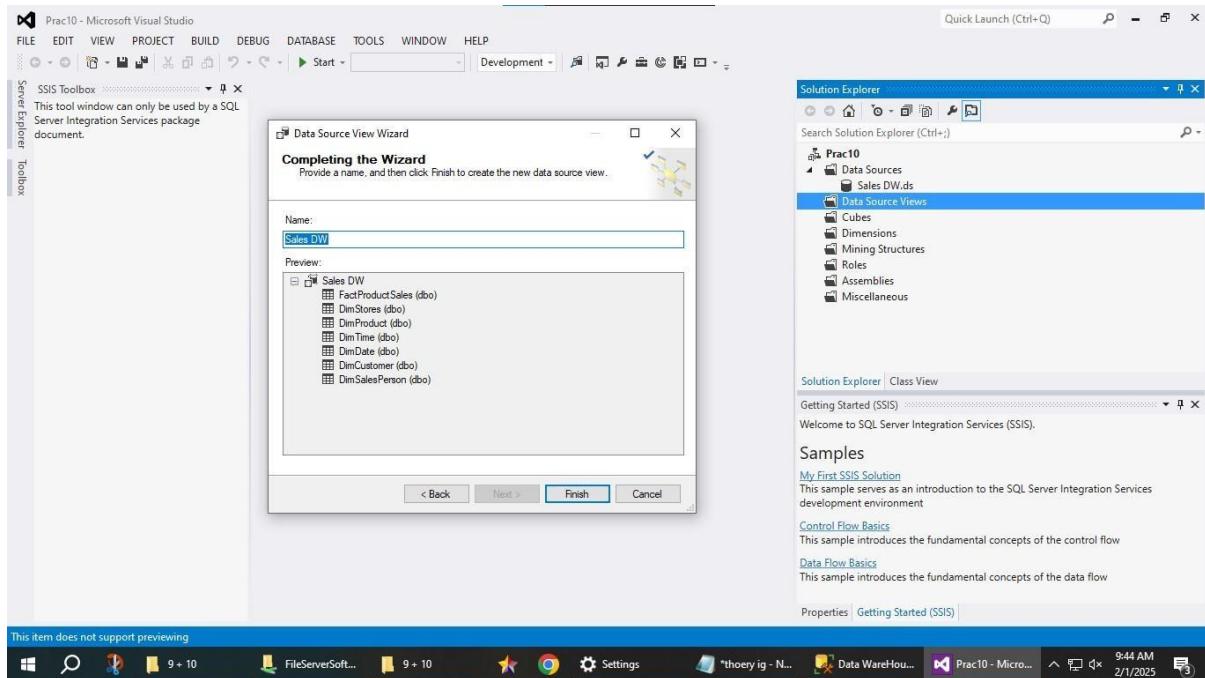
7. Create a Data Source View, select Sales DW, add the fact table and related dimension tables, then finish.



BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



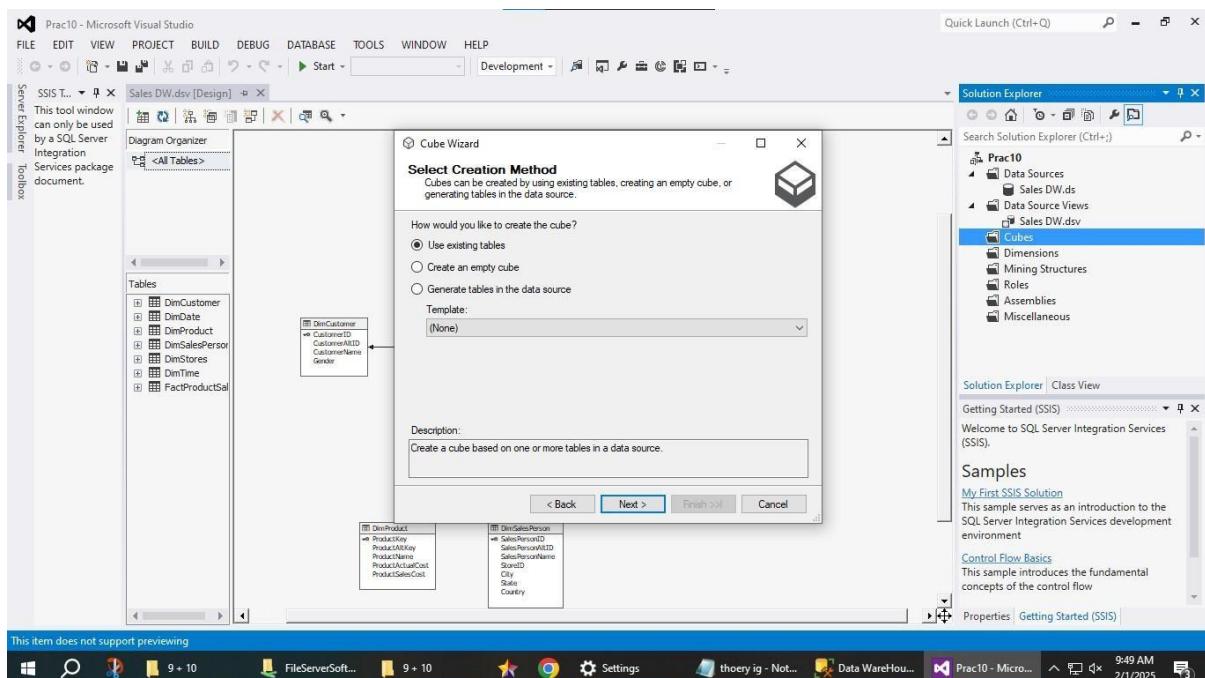
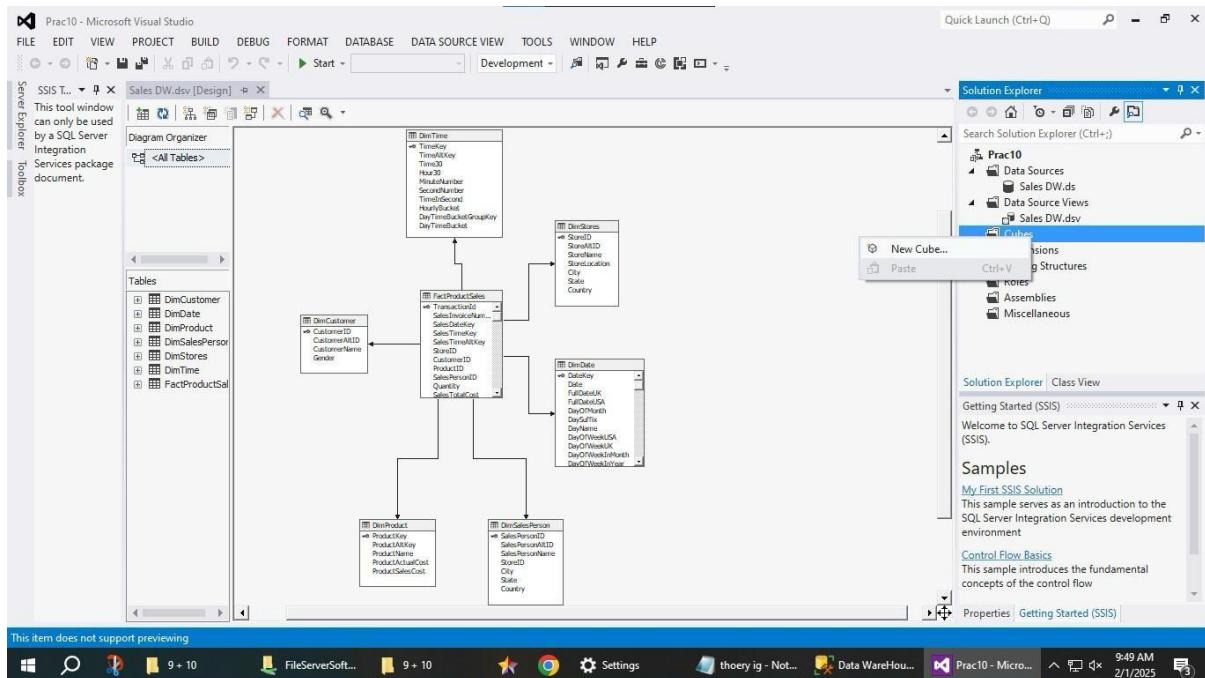
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



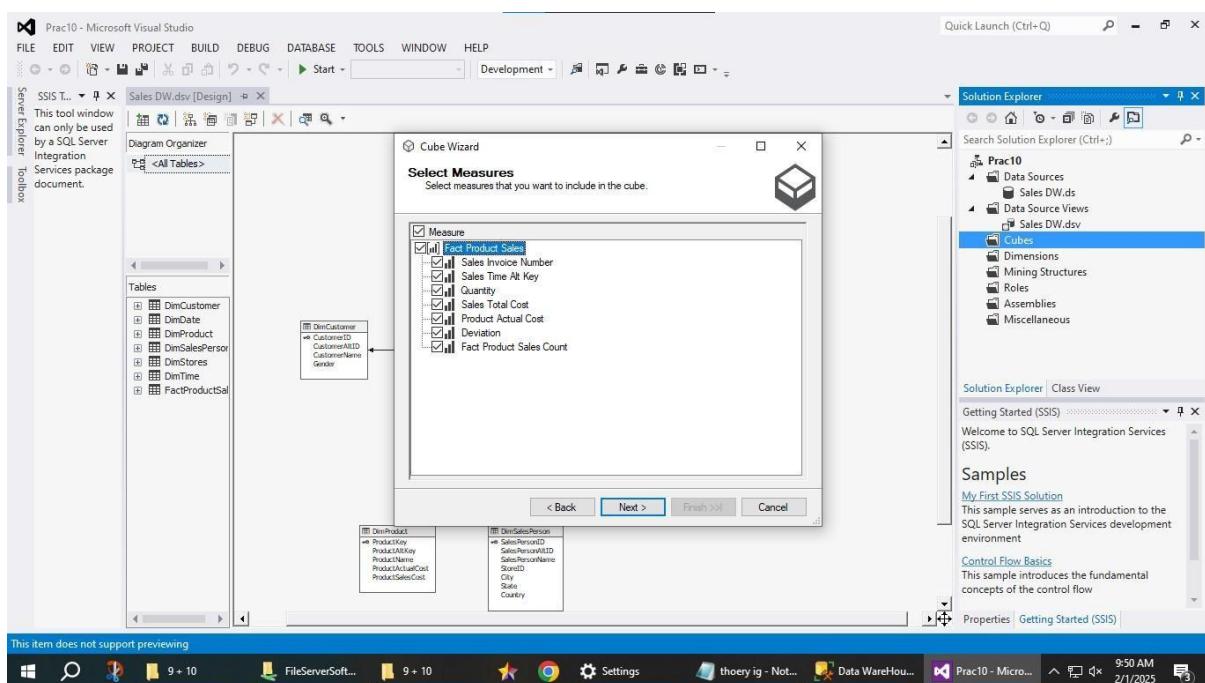
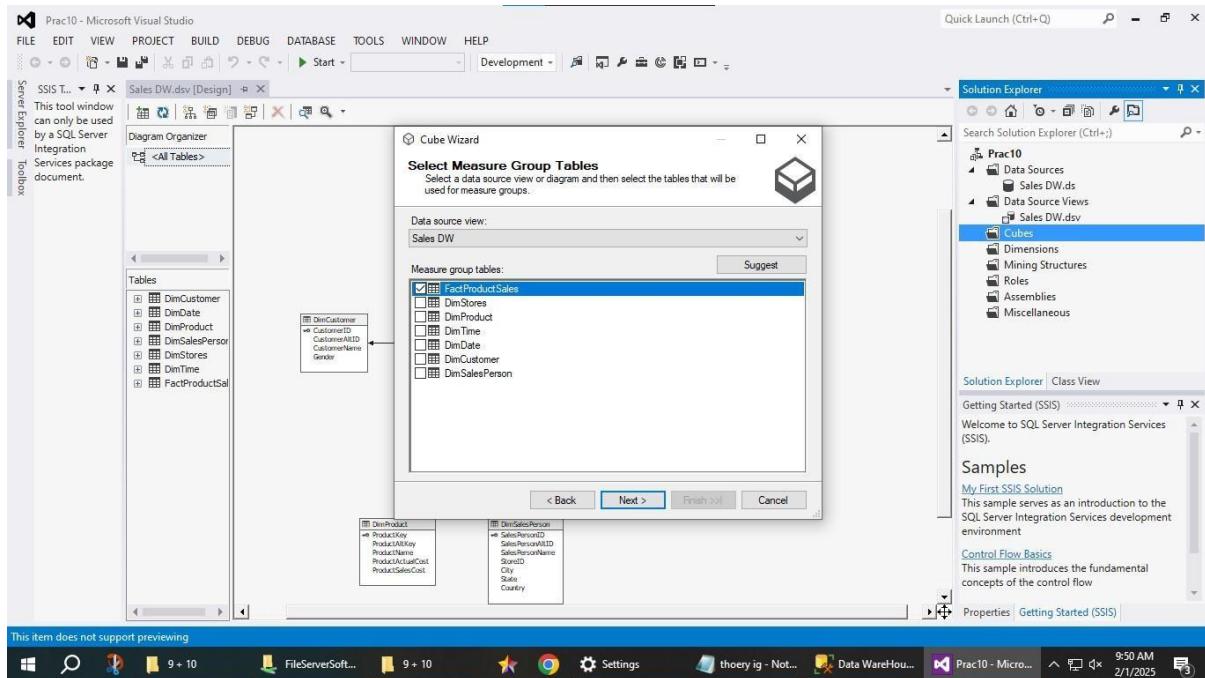
Practical No. 10

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

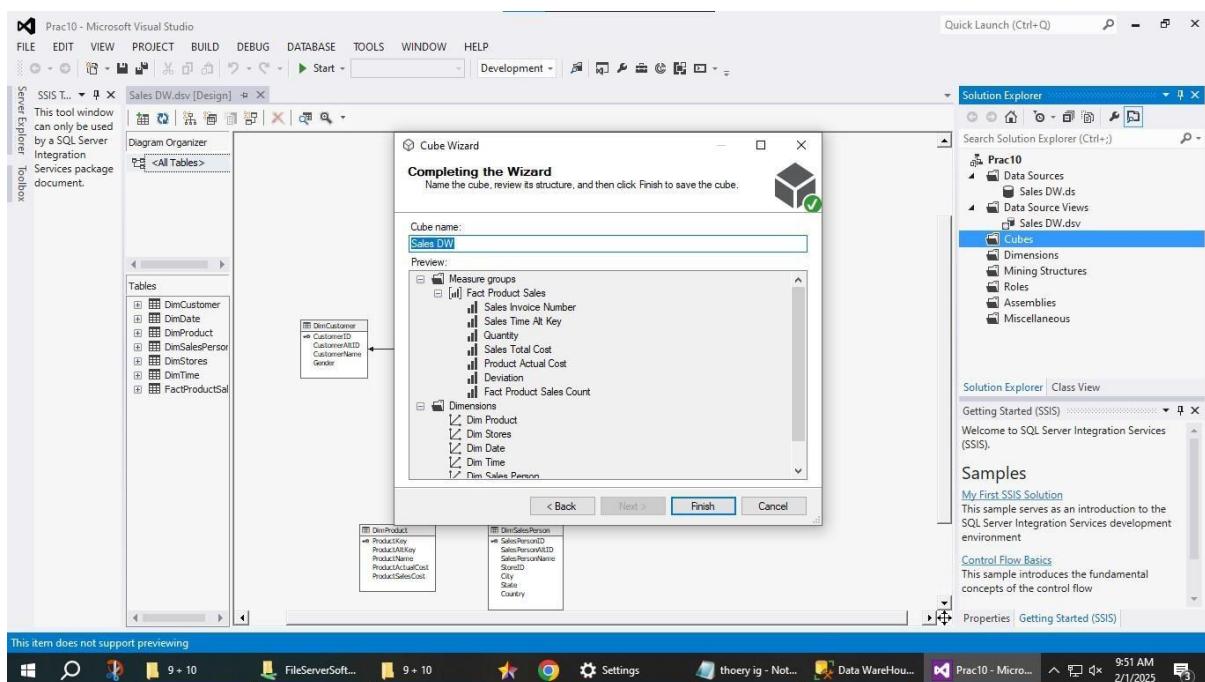
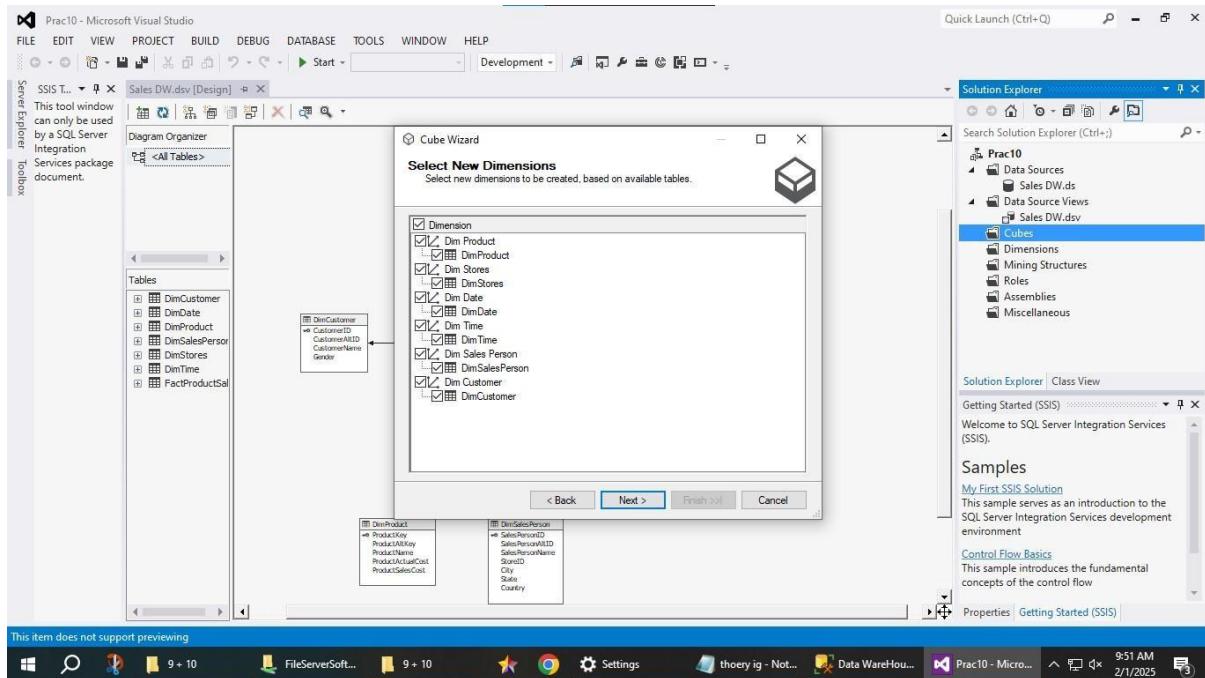
1. Create a new Cube, use an existing table, select the fact table, and complete the wizard.



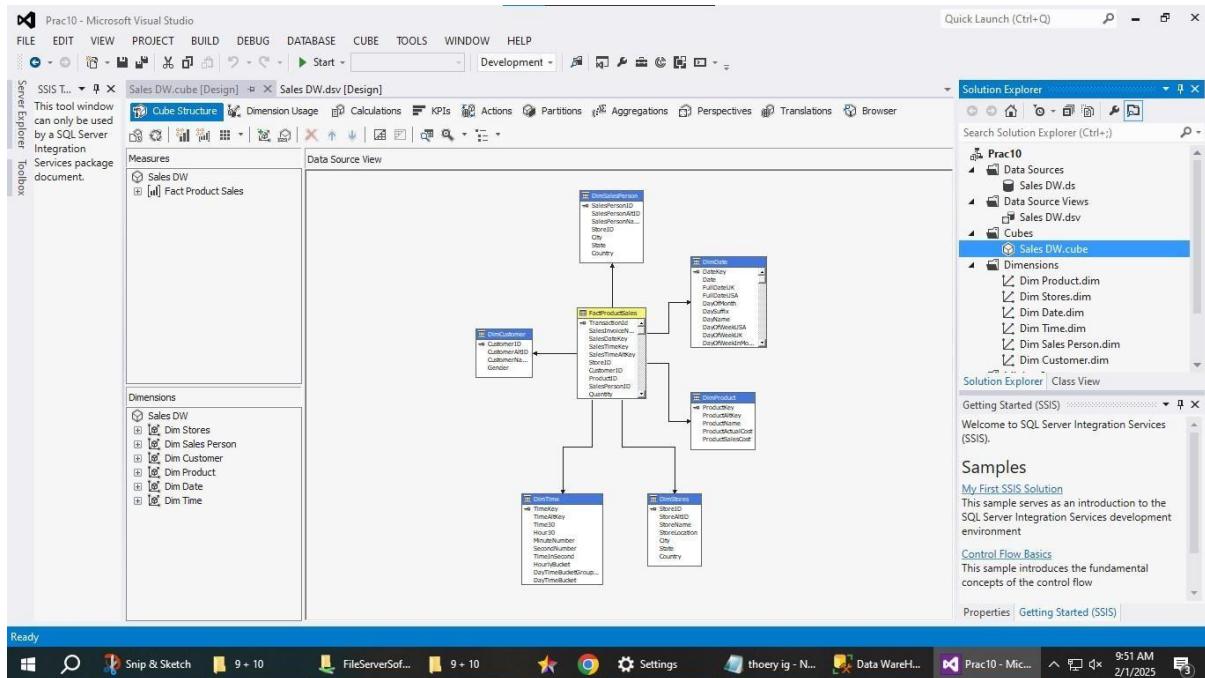
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



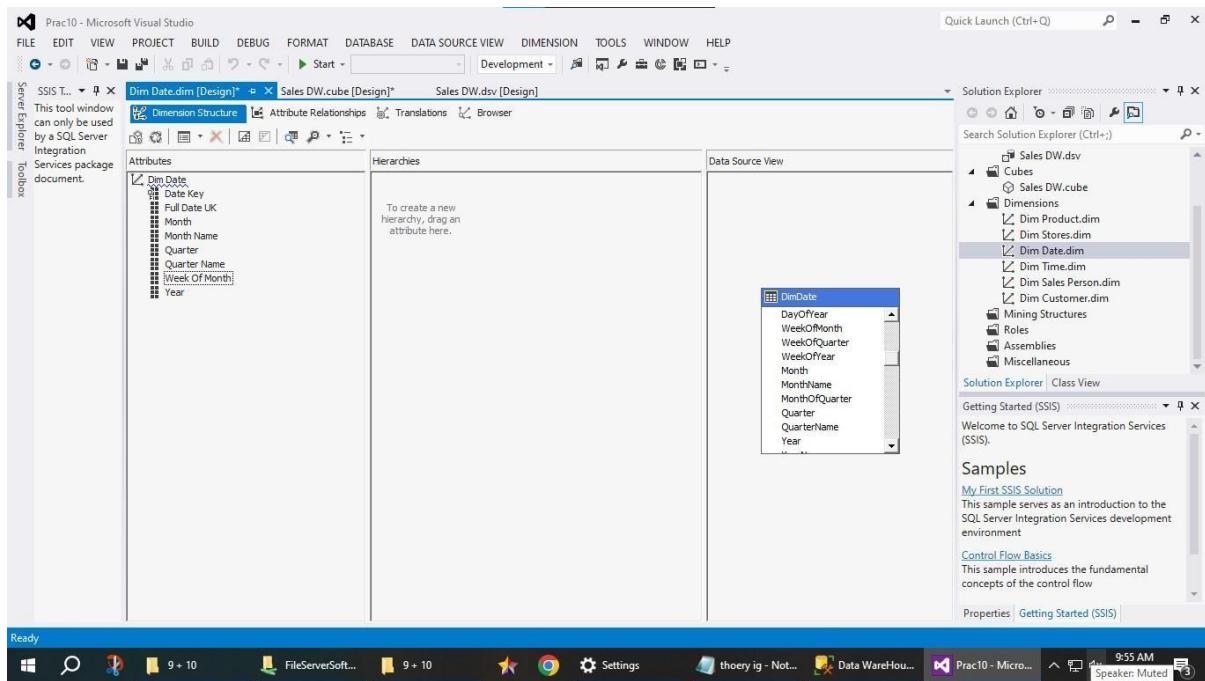
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



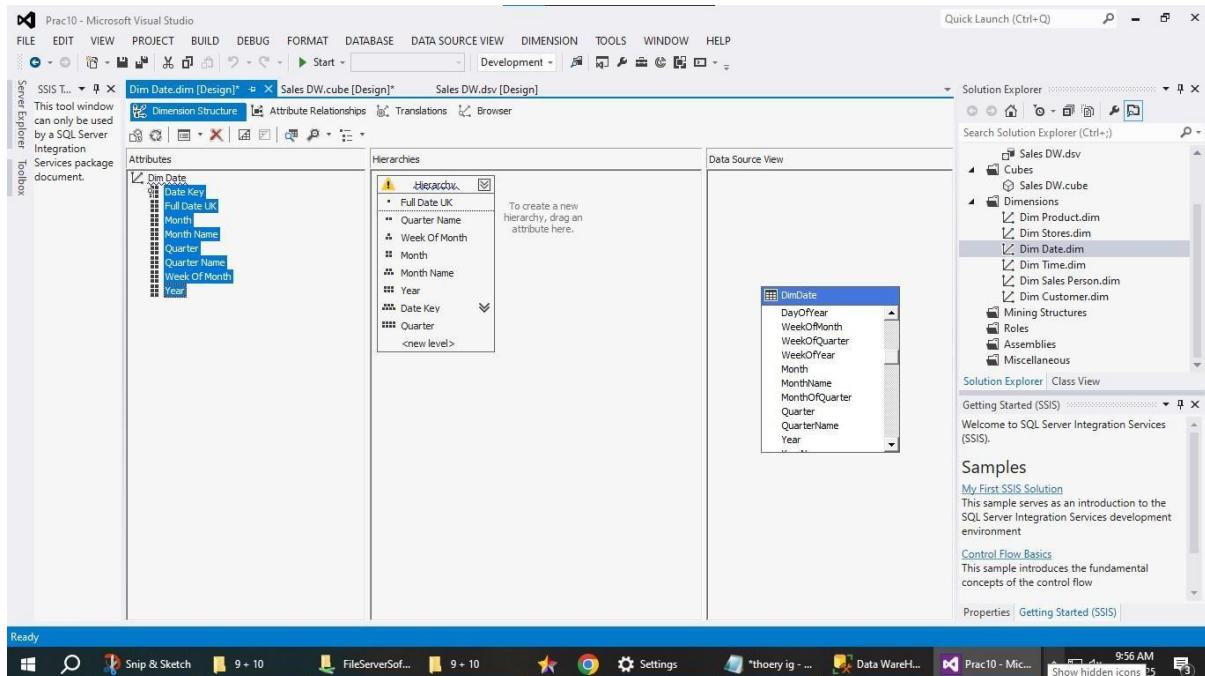
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



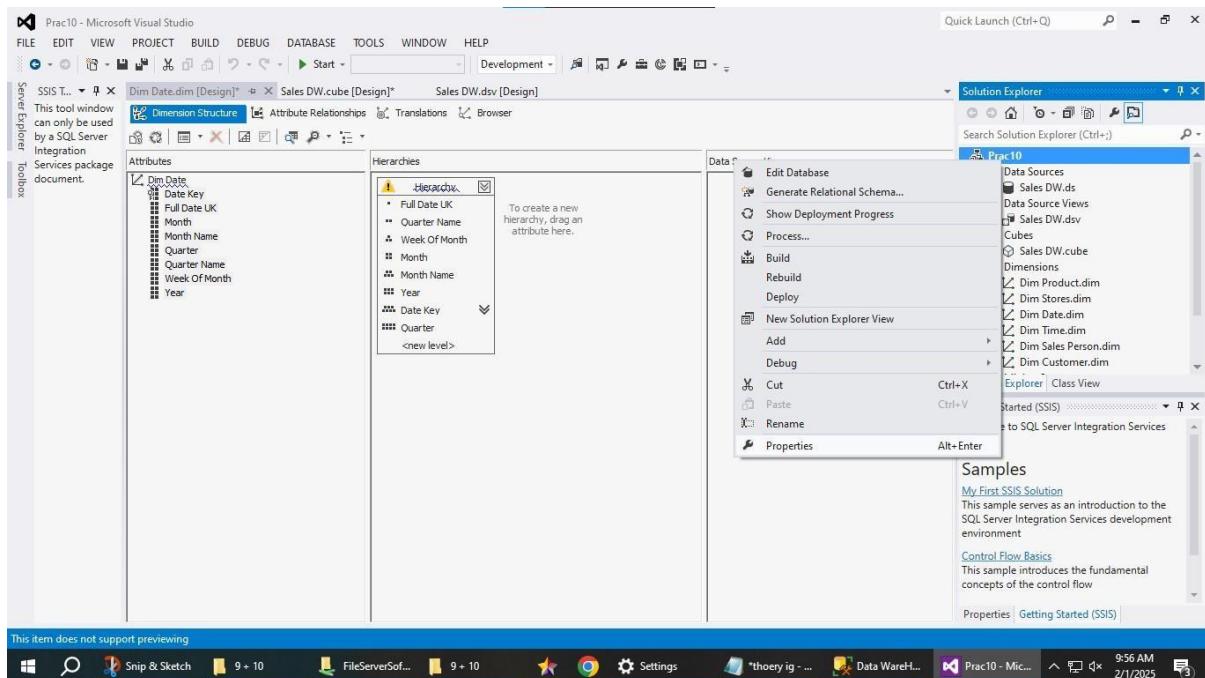
2. Drag hierarchy fields (Full Date UK, Month, Quarter, Year, etc.) into the hierarchy.



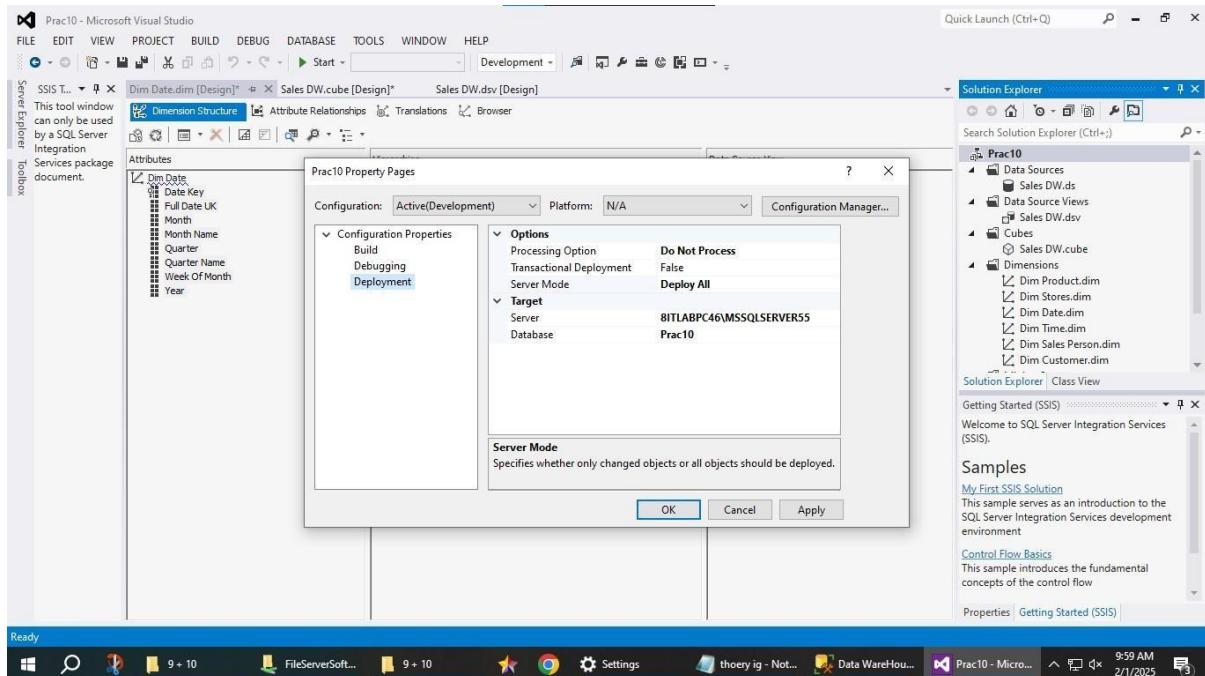
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



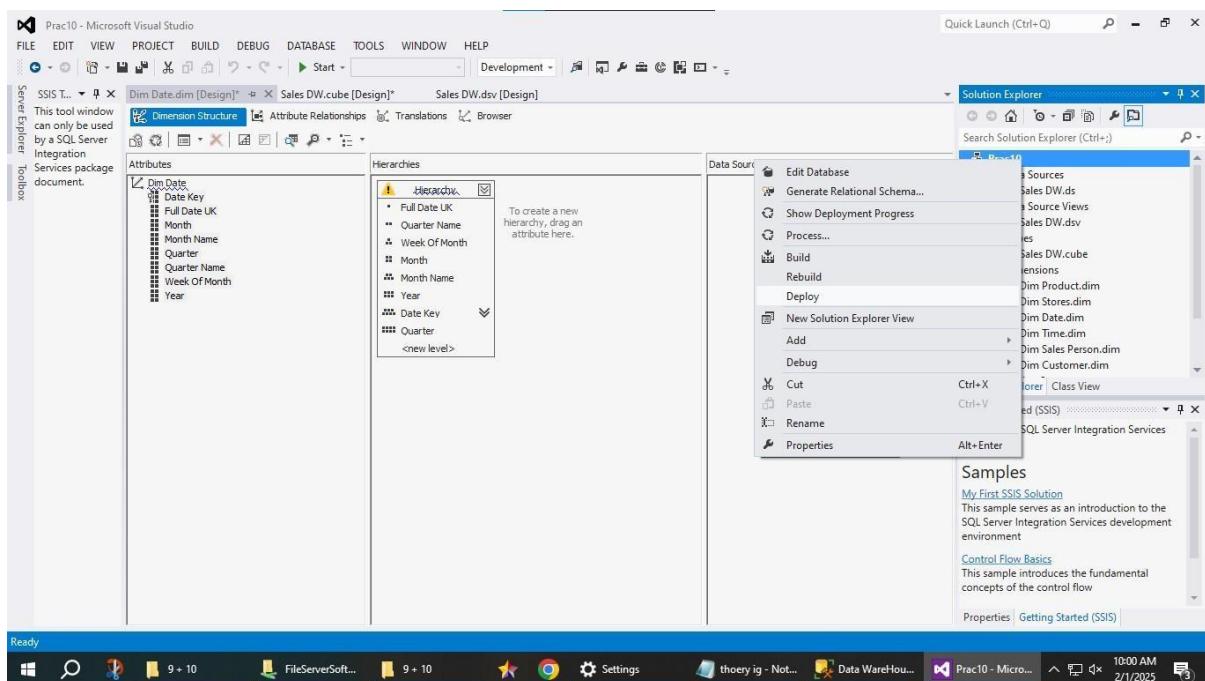
- In Prac10 properties, set deployment to "Do Not Process," server mode to "Deploy All," target database as copied name, and database as Prac10.



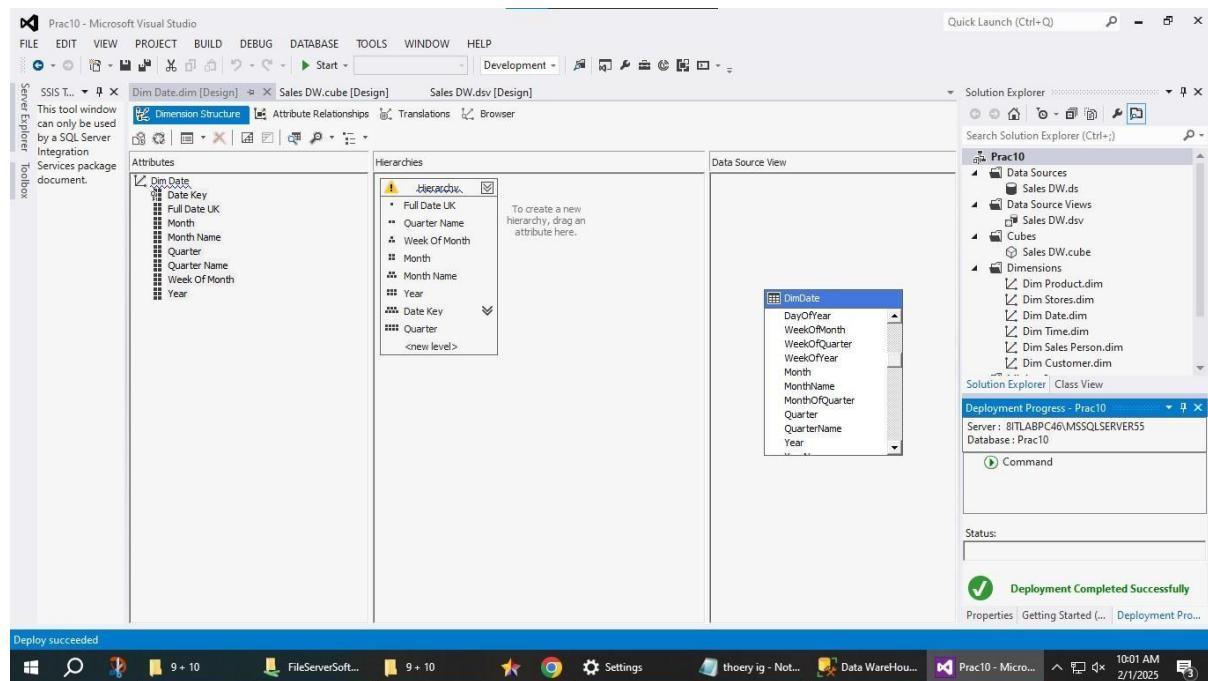
BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



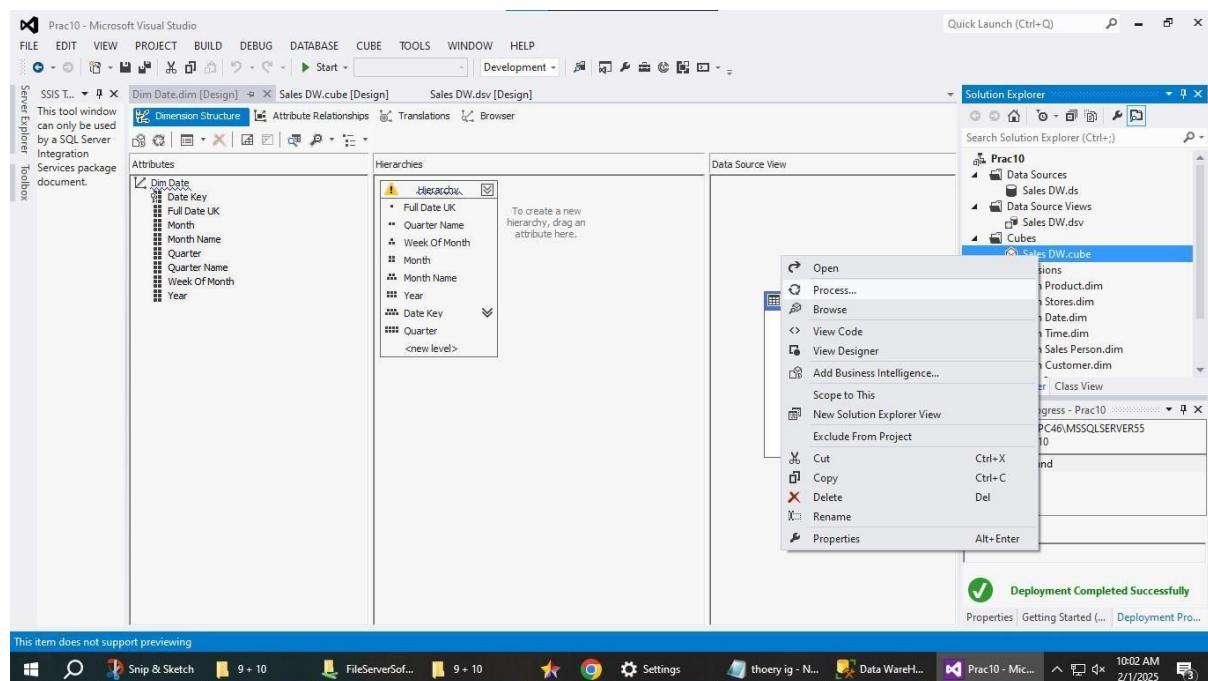
4. Deploy Prac10 (should show a green tick if successful).



BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS



5. Open SalesDW Cube, run it, and confirm "Process Succeeded."



BUSINESS INTELLIGENCE AND DATA ANALYTICS PRACTICALS

