RAJALAKSHMI ENGINEERING COLLEGE

[AUTONOMOUS]

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

AI19P82 BUSINESS INTELLIGENCE AND ANALYTICS
Laboratory Record Notebook

Name : . . . .

Year / Branch / Section : . . . . -

Register No. : . . . .

College Roll No. : . . . .

Semester : . . . .

Academic Year : . . . .

RAJALAKSHMI ENGINEERING COLLEGE

[AUTONOMOUS]

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name : . . . . . .

Academic Year : . . . . . .Semester : . . . . Branch:AIML

Register No. | |

Certified that this is the bonafide record of work done by the above student in the AI19P82 – Business Intelligence and Analytics Laboratory during the year 2024 - 2025.

Signature of Faculty in-charge

Submitted for the Practical Examination held on . . . . . .

Internal Examiner **External Examiner**

RAJALAKSHMI ENGINEERING COLLEGE[AUTONOMOUS]

INDEX

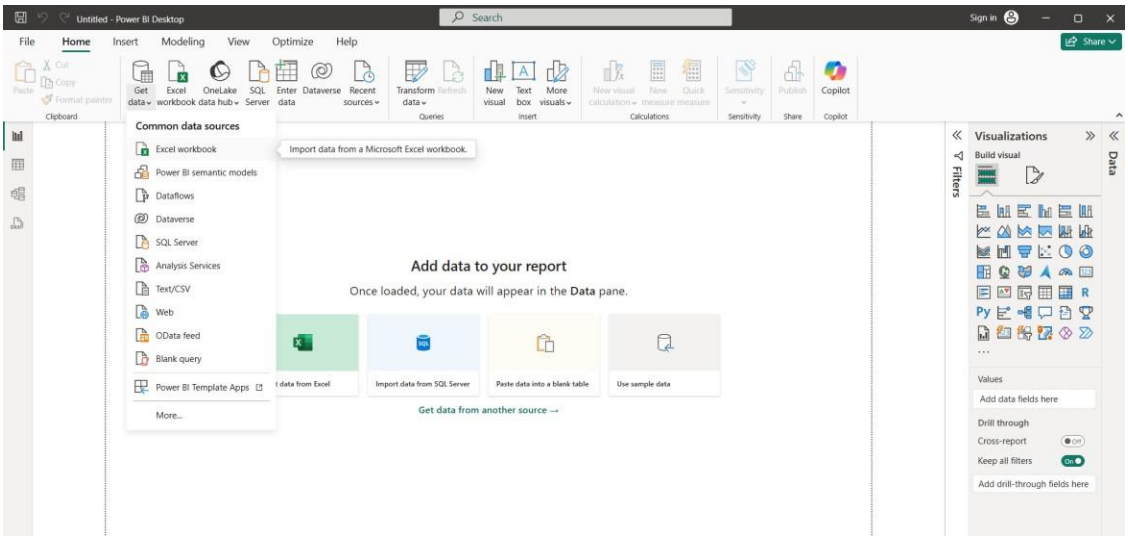
EX.NO : 1 (a)	IMPORT THE LEGACY DATA FROM DIFFERENT SOURCES SUCH AS (EXCEL, SQLSERVER, ORACLE ETC.) AND LOAD IN THE TARGET SYSTEM
DATE :	

AIM:

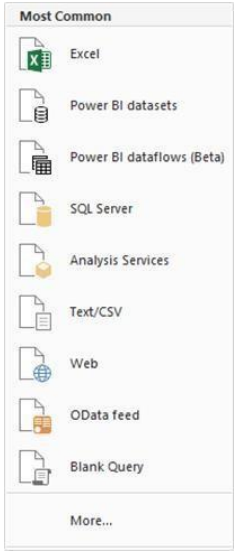
To import the legacy data from different sources such as (Excel, SQLServer, Oracle etc.) and load in the target system.

PROCEDURE:

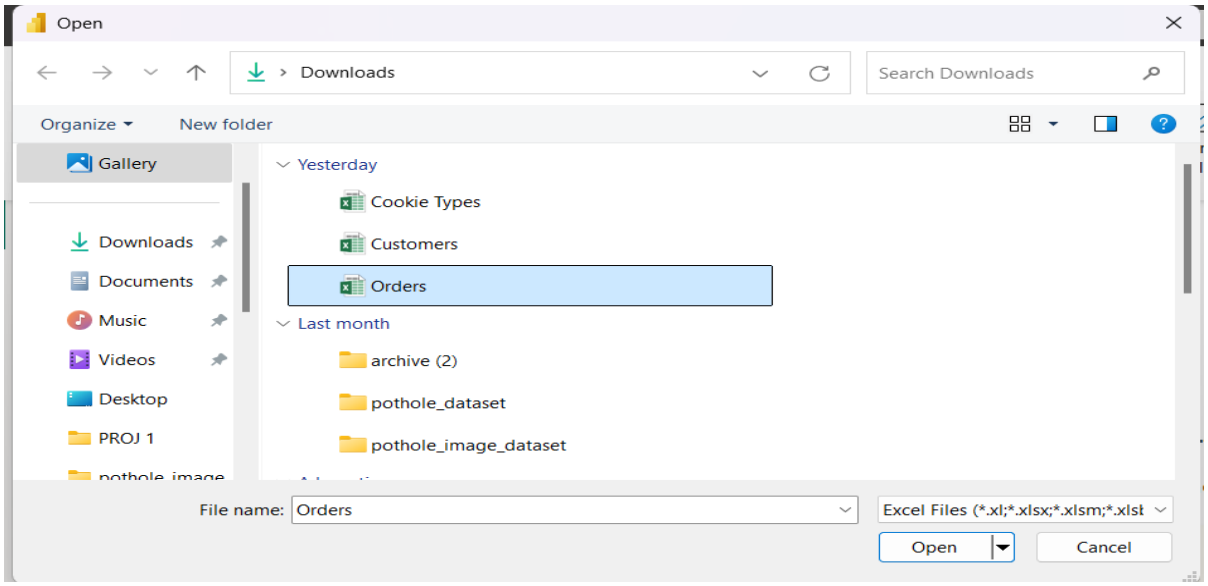
Step 1: Open Power BI



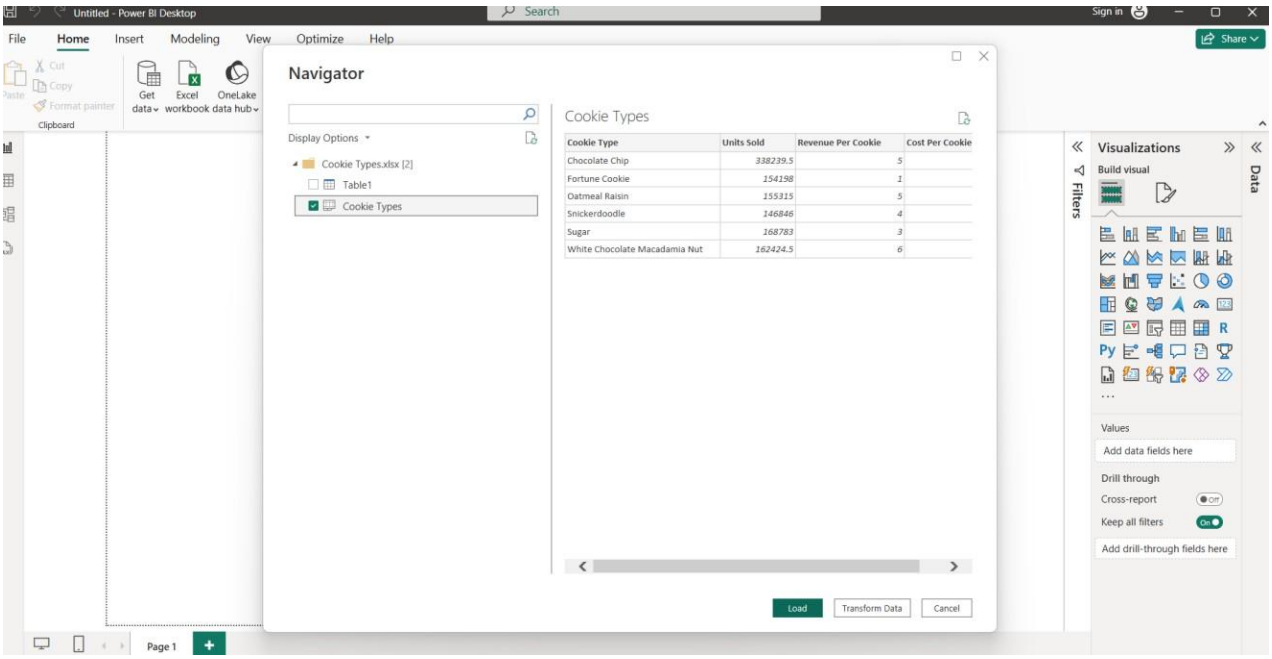
Step 2: Click on Get data following list will be displayed →select Excel

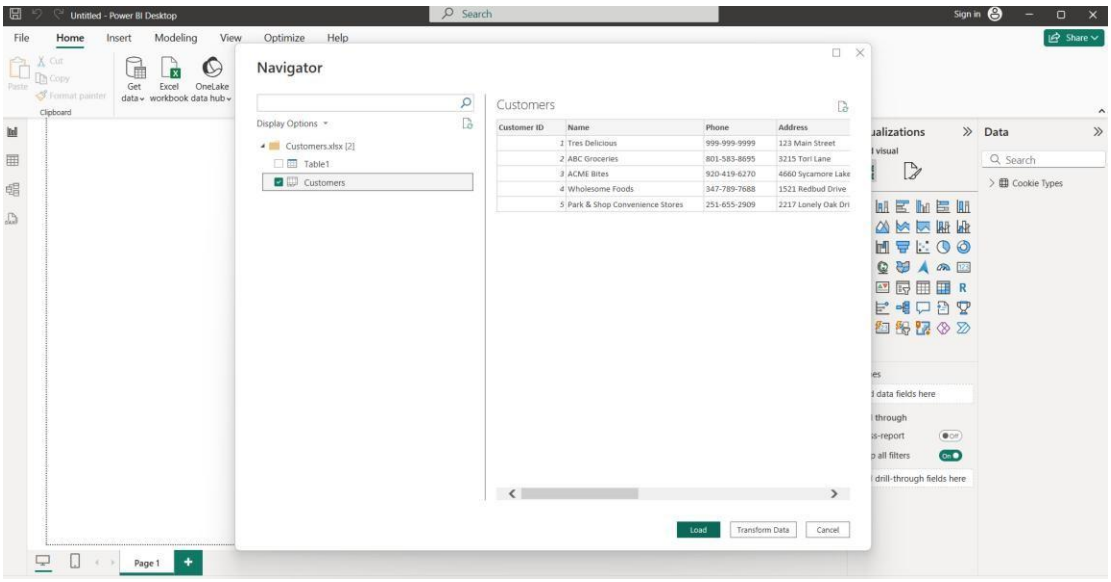


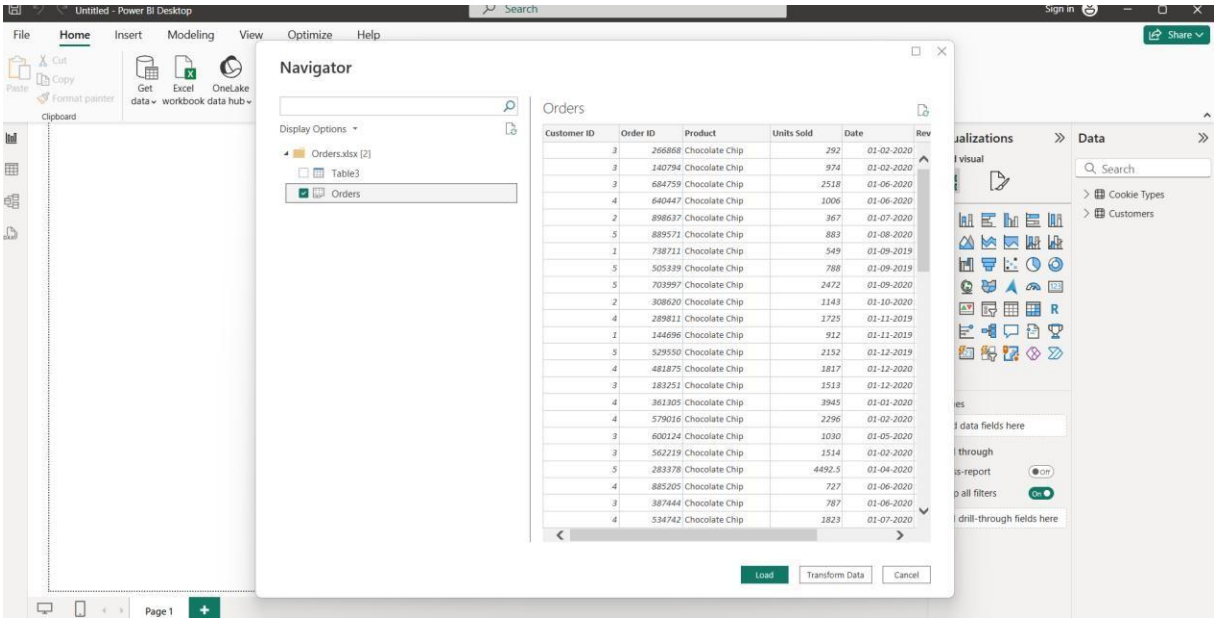
Step 3: Select required file and click on Open, Navigator screen appears



Step 4: Select file and load the data







RESULT:

Thus the procedure to import the legacy data from different sources such as (Excel, SQLServer, Oracle etc.) and loaded in the target system successfully and the output is verified

EX.NO : 1 (b)	IMPORT THE LEGACY DATA FROM DIFFERENT SOURCES SUCH AS (EXCEL, SQLSERVER, WIKIPEDIA ETC.) AND LOAD IN THE TARGET SYSTEM
DATE :	

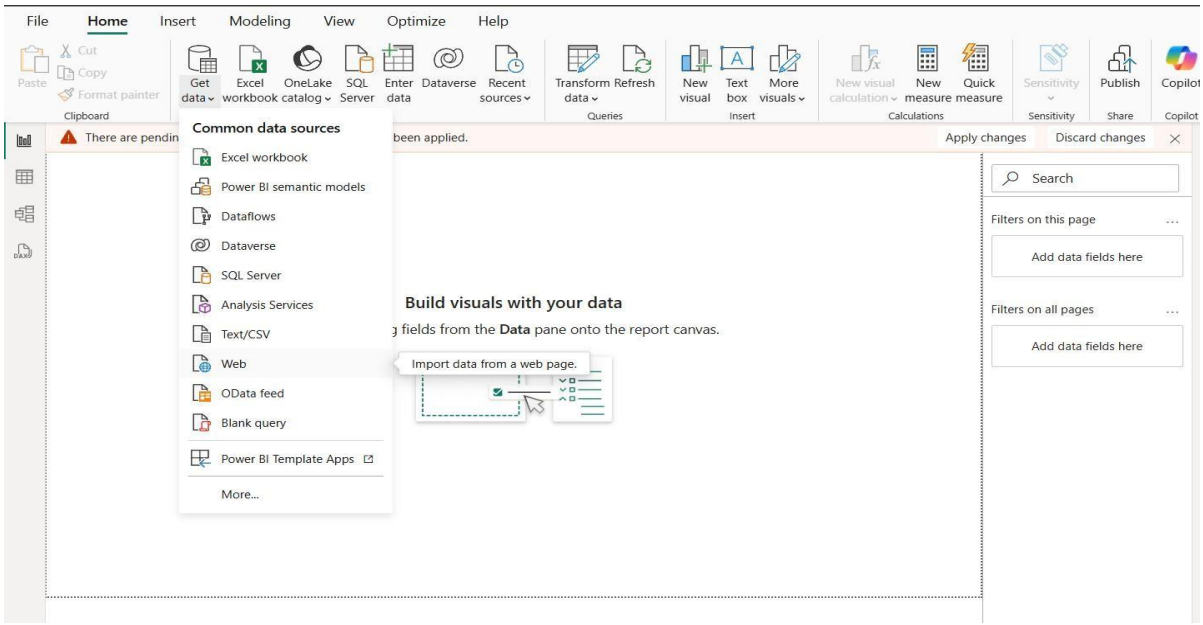
AIM:

To create an interactive Power BI report that dynamically filters data based on a selected country using a slicer and DAX queries.

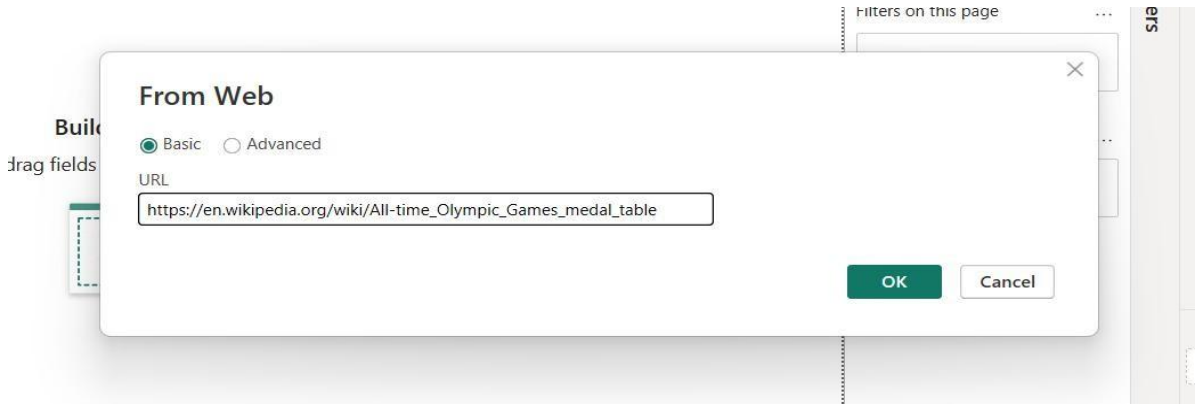
PROCEDURE:

Step 1: Open Power BI Desktop

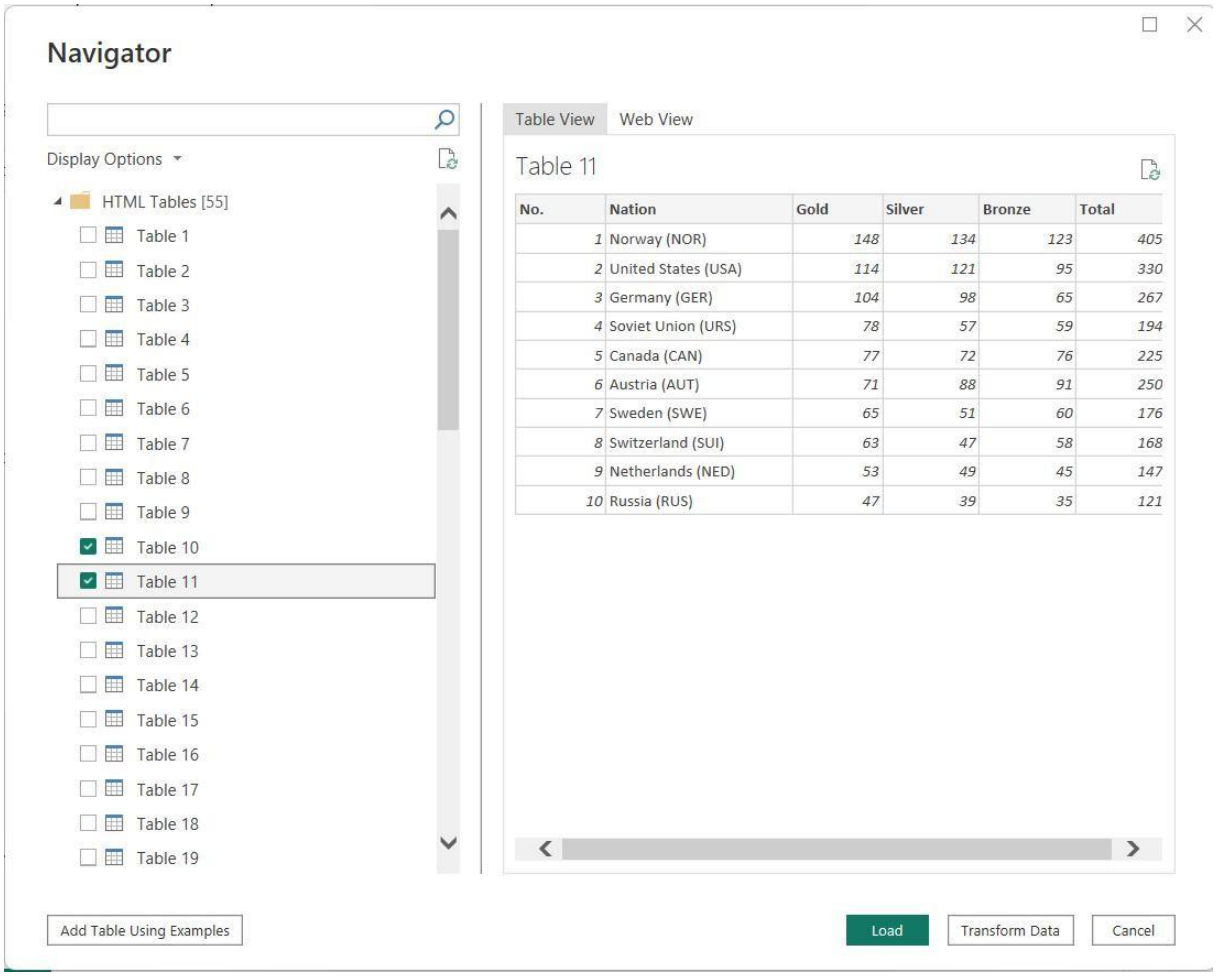
a) Click on Home → Select Get Data → Choose Web.



b) Paste the url: https://en.wikipedia.org/wiki/All-time_Olympic_Games_medal_table, And click OK let Power BI fetch the data.

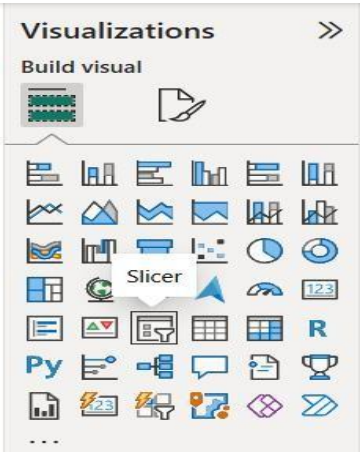


c) In the Navigator window, select the appropriate table and Click **Load** to import the data into Power BI.



Step 2: Create a Slicer for Country Selection.

a) Go to Visualizations Pane → Click on Slicer.

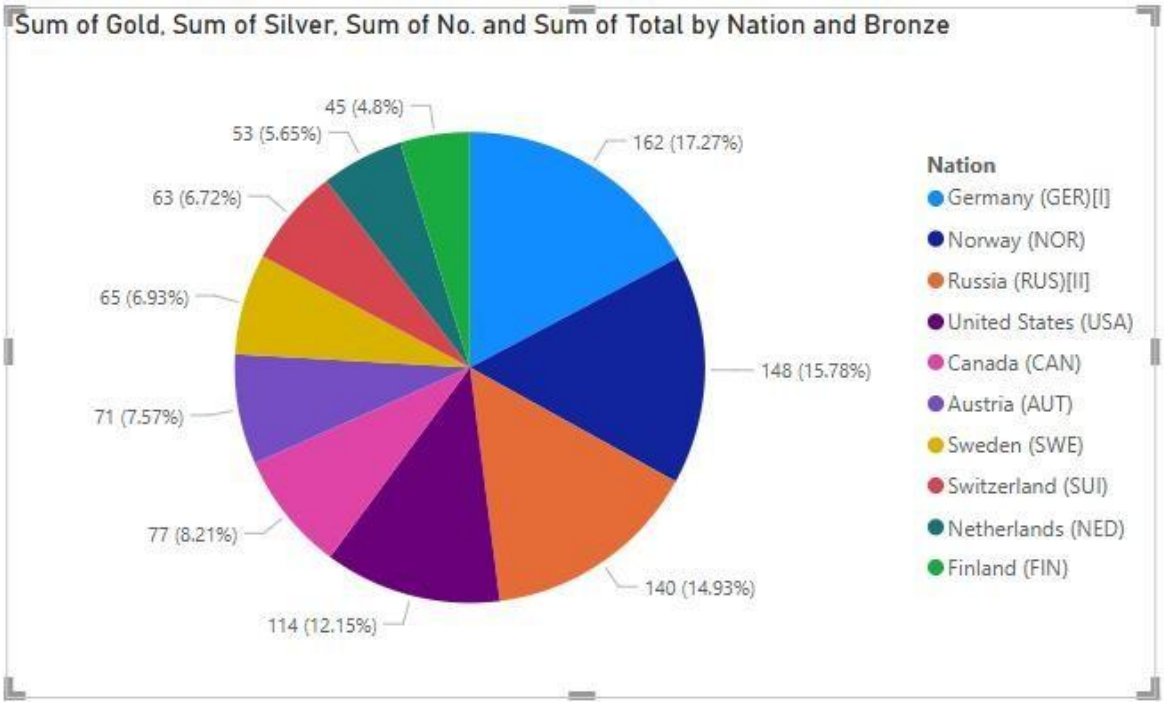


b) Drag the Country field into the Slicer visualization.

Nation	Sum of Bronze	Sum of Gold	Sum of Silver	Sum of No.	Sum of Total
Austria (AUT)	91	71	88	6	250
Canada (CAN)	76	77	72	5	225
Finland (FIN)	65	45	65	10	175
Germany (GER)[I]	118	162	155	1	435
Netherlands (NED)	45	53	49	9	147
Norway (NOR)	123	148	134	2	405
Russia (RUS)[II]	126	140	120	3	386
Sweden (SWE)	60	65	51	7	176
Switzerland (SUI)	58	63	47	8	168
United States (USA)	95	114	121	4	330
Total	**857**	**938**	**902**	**55**	**2697**

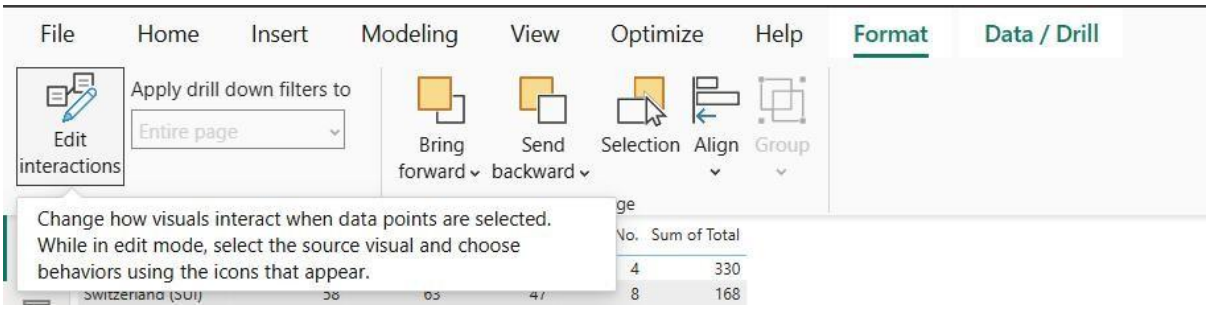
Step 3: Create a Visualization for Medal Data.

a) Add a Table, Bar Chart, or Column Chart from the Visualizations Pane and Drag the following fields into the visualization and Adjust formatting and design as needed.

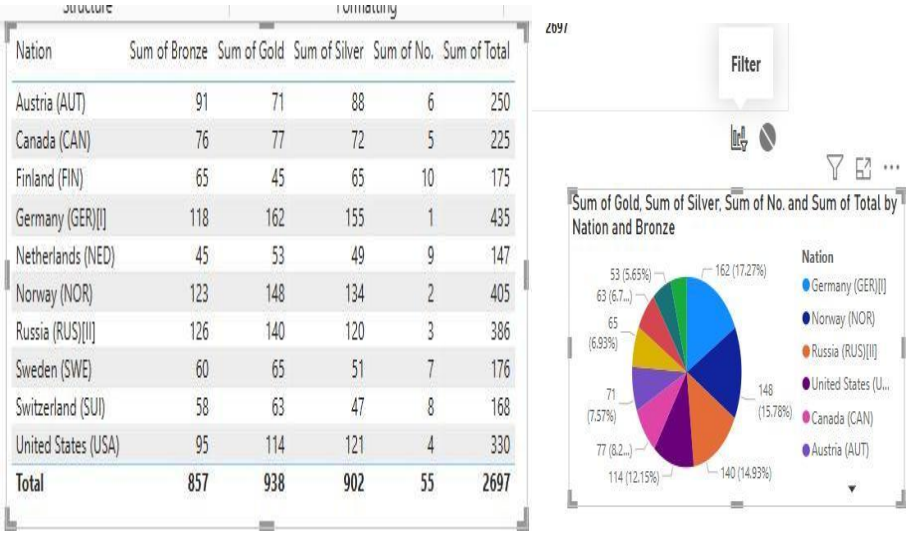


Step 4: Enable Interactive Filtering.

a) Click on the Slicer (Country List) and Go to Format Pane → Select Edit Interactions.



b) Click on the Filter Icon on the visualization.

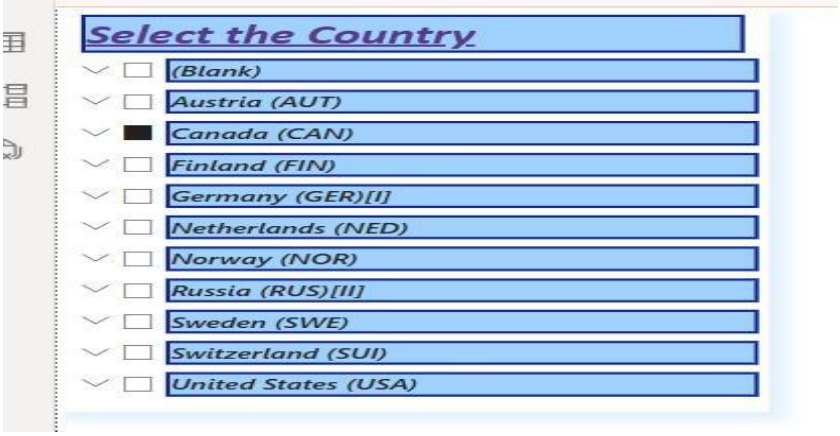


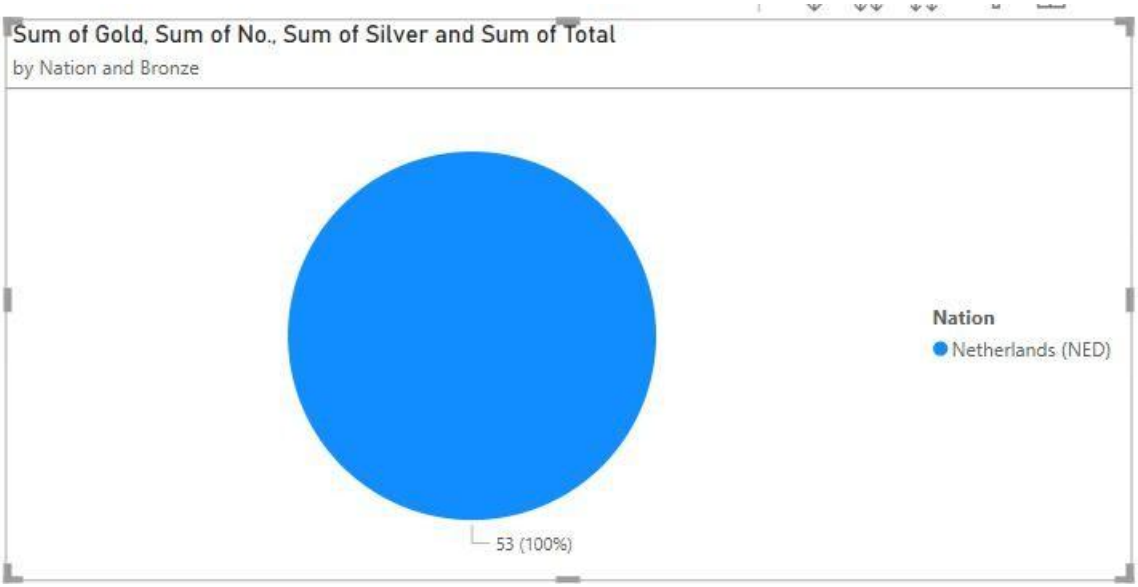
Nation	Sum of Bronze	Sum of Gold	Sum of Silver	Sum of No.	Sum of Total
Austria (AUT)	91	71	88	6	250
Canada (CAN)	76	77	72	5	225
Finland (FIN)	65	45	65	10	175
Germany (GER)[I]	118	162	155	1	435
Netherlands (NED)	45	53	49	9	147
Norway (NOR)	123	148	134	2	405
Russia (RUS)[II]	126	140	120	3	386
Sweden (SWE)	60	65	51	7	176
Switzerland (SUI)	58	63	47	8	168
United States (USA)	95	114	121	4	330
Total	857	938	902	55	2697

c) Go to Visualization in the format visual can able to add the filter.

Step 5: Testing & Final Adjustments.

a) Click on different countries in the slicer to test interactivity.





b) Adjust chart formatting (colors, labels, titles) for better visualization.
c) Save the report as .pbix file.

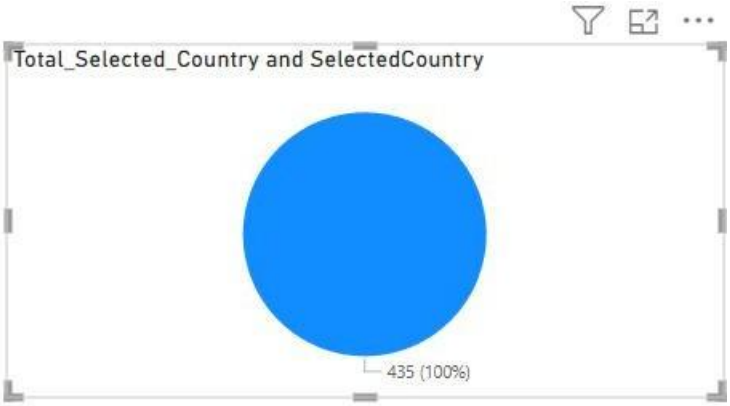
DAX queries for the slicer selection:

1. Get the Selected Country Name:

```
DAX


SelectedCountry =
SELECTEDVALUE('MedalsData'[Nation1], "Select a Country")
```

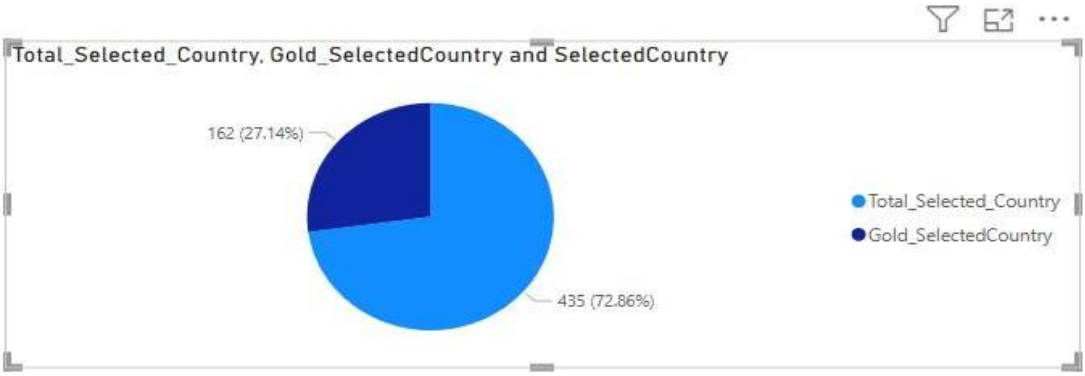
2. Filter Total Medals for Selected Country:

```
DAX
Total_Selected_Country =
CALCULATE(
    SUM('MedalsData'[Total]),
    'MedalsData'[Nation1] = SELECTEDVALUE('MedalsData'[Nation1])
)
```



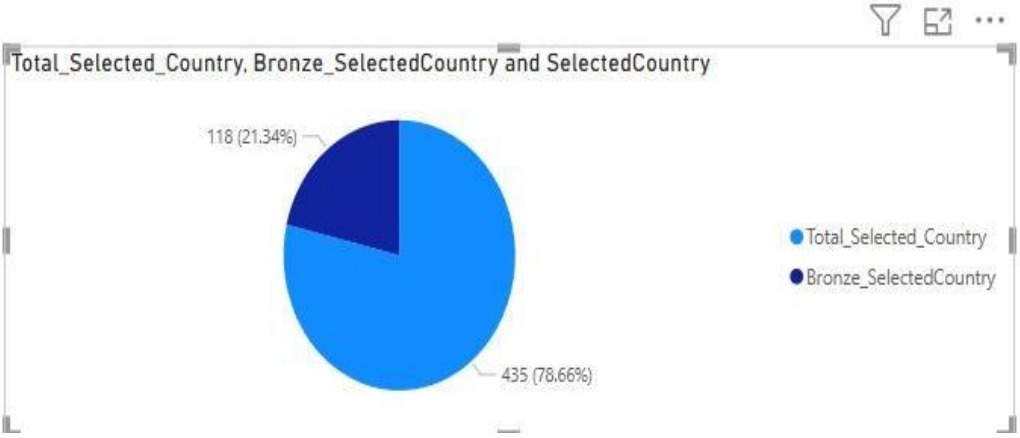
3. Get Individual Medal Counts:

```
DAX
Gold_SelectedCountry =
CALCULATE(
    SUM('MedalsData'[Gold]),
    'MedalsData'[Nation1] = SELECTEDVALUE('MedalsData'[Nation1])
```



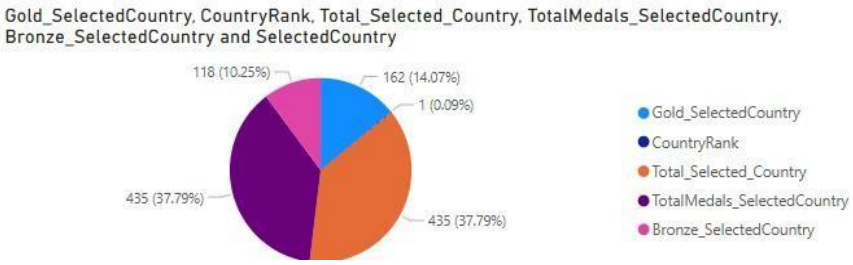
a) Bronze Medal:

```
DAX
Bronze_SelectedCountry =
CALCULATE(
    SUM('MedalsData'[Bronze]),
    'MedalsData'[Nation1] = SELECTEDVALUE('MedalsData'[Nation1])
)
```



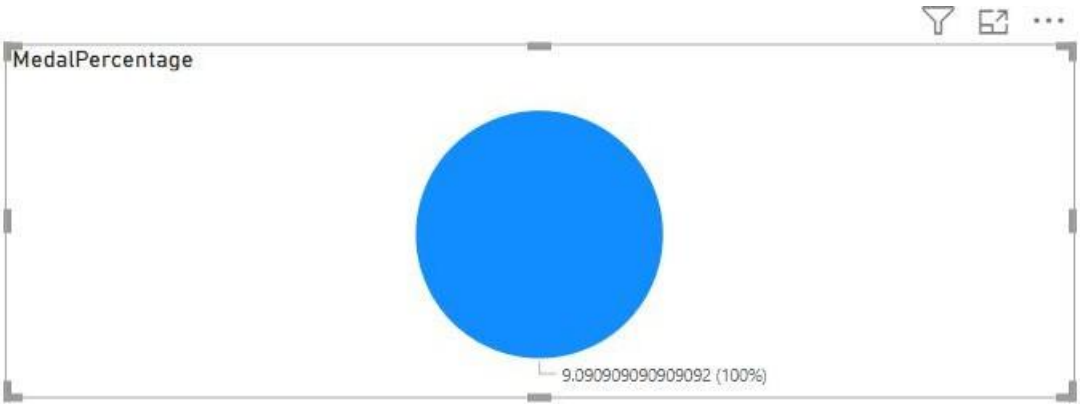
1. Rank Countries Based on Total Medals

```
DAX
CountryRank = RANKX(
    ALL('MedalsData'),
    SUM('MedalsData'[Total]),
    , DESC,
    DENSE
)
```

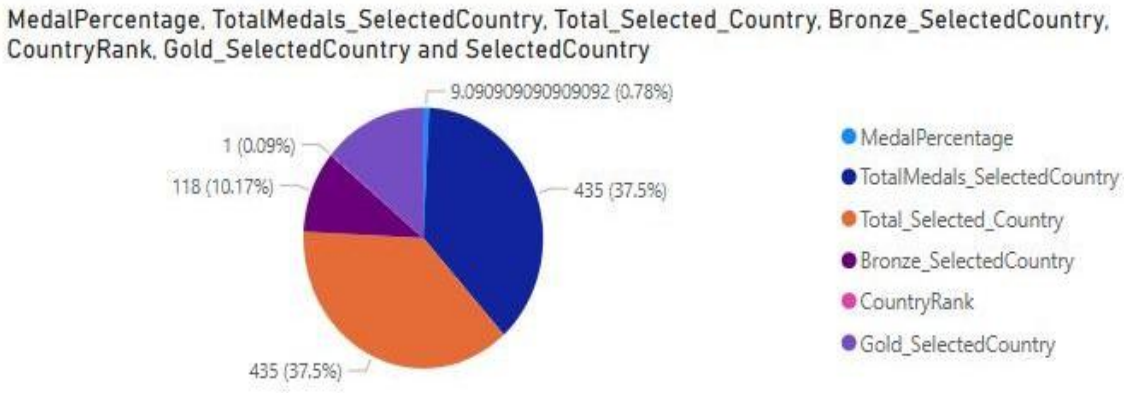


1. Percentage of Medals Won by the Selected Country

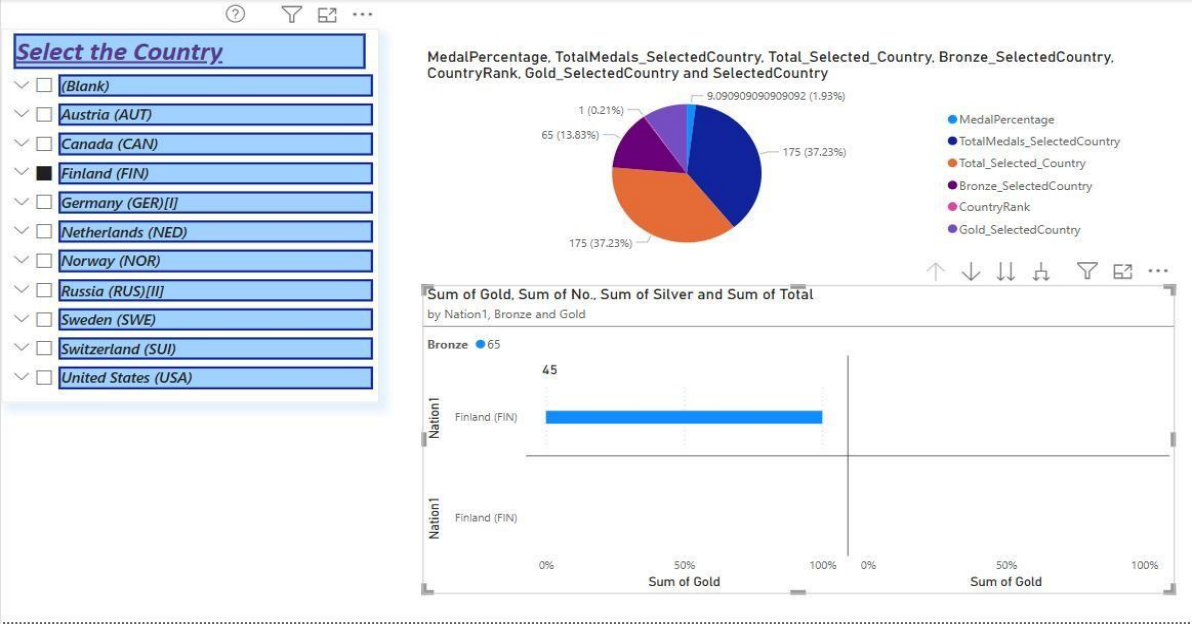
```
DAX
MedalPercentage =
DIVIDE(
                [TotalMedals_SelectedCountry],
    SUMX(ALL('MedalsData'), SUM('MedalsData'[Total])),
                                0
) * 100
```



1. Over all DAX Query Visualization:



Over all Dashboard:



RESULT:

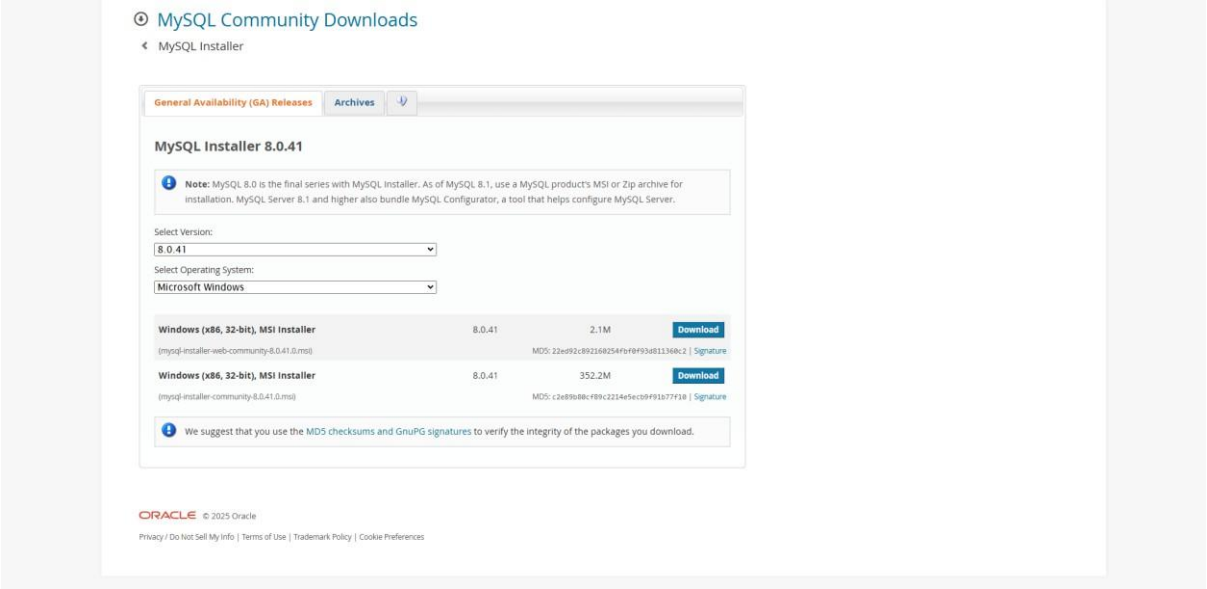
The slicer dynamically filters data based on the selected country, updating visualizations accordingly. DAX queries like SELECTEDVALUE and FILTER ensure efficient data retrieval and calculation for interactive analysis.

EX.NO : 2	IMPORT THE LEGACY DATA FROM SQLSERVER AND LOAD IN THE TARGET SYSTEM
DATE :	

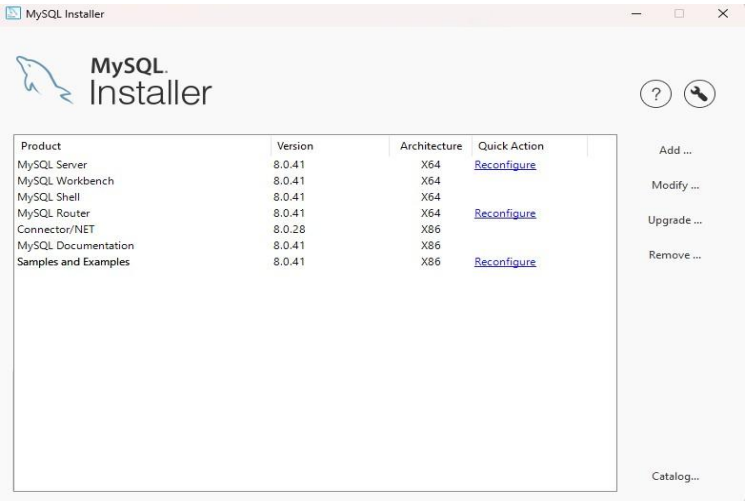
AIM:To import the data from SQLServer and load in the target system.

PROCEDURE:

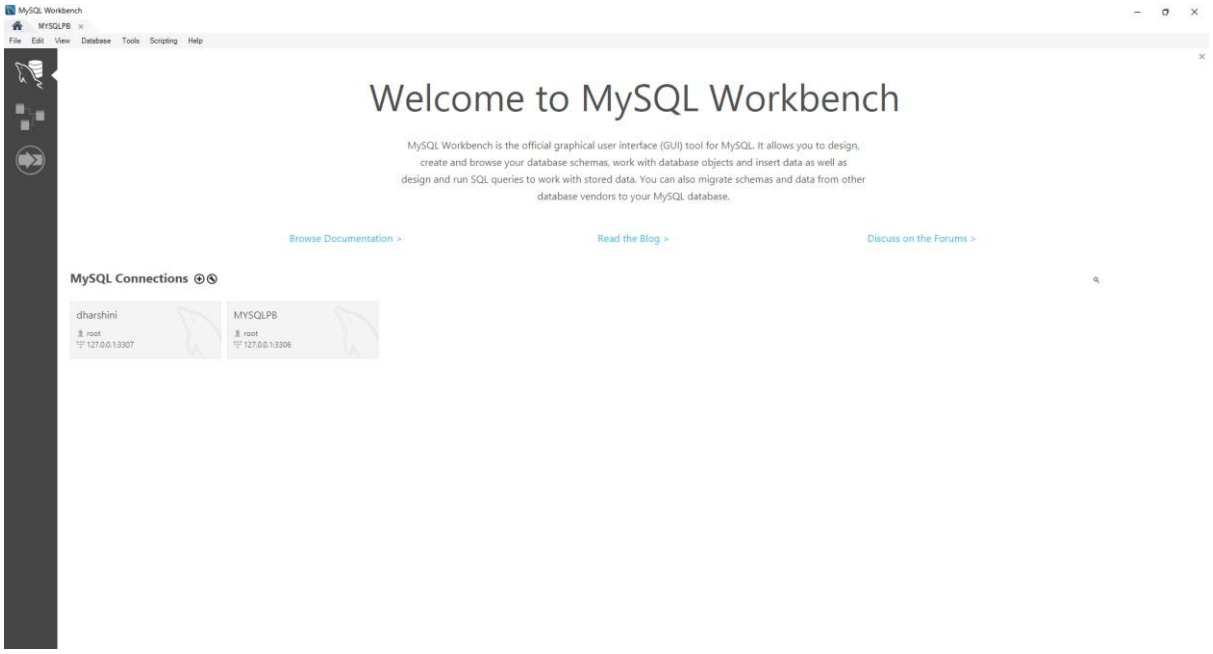
Step 1: Download MYSQL installer from https://dev.mysql.com/downloads/installer/.



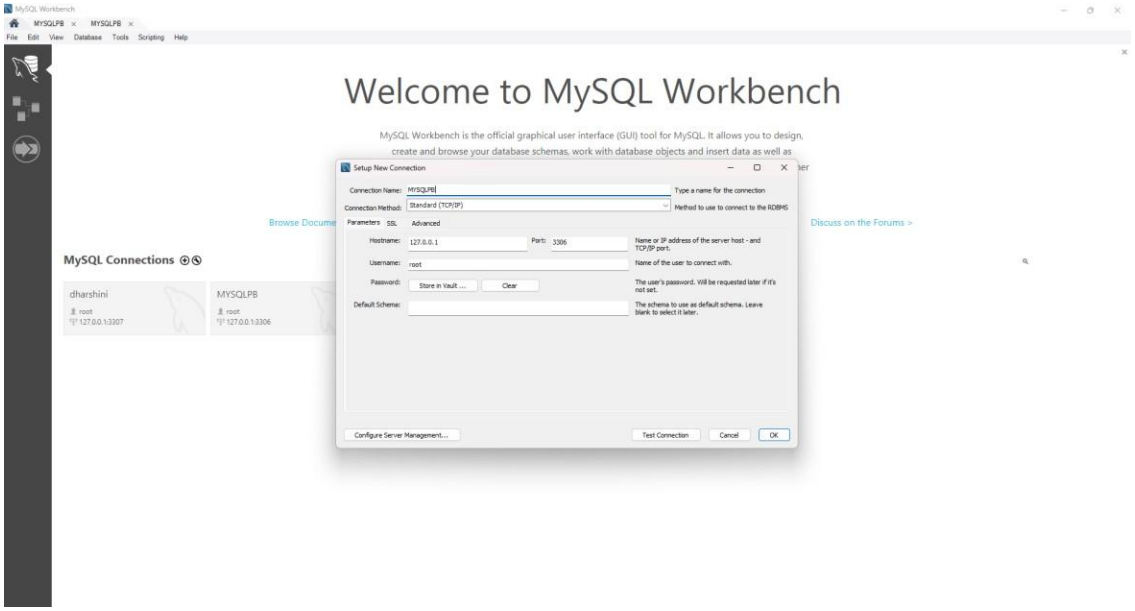
Step 2: Execute MYSQL installer and Add Server Files and configure it by giving server name and password for the database.



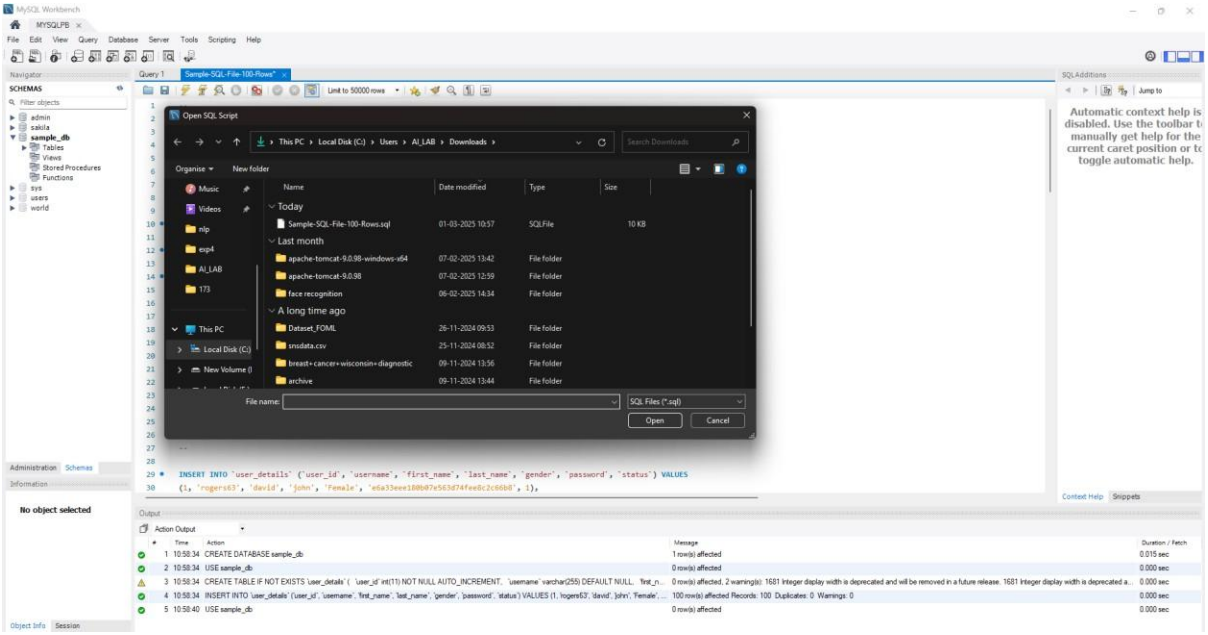
Step 3: Open MYSQL Workbench



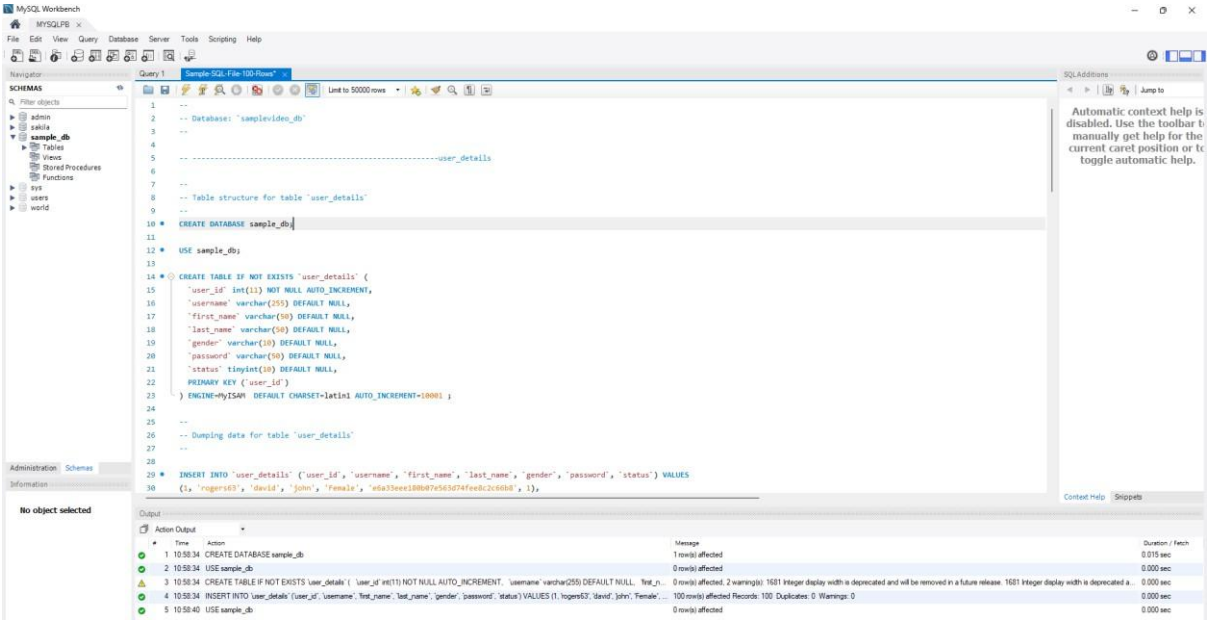
Step4: Create a MYSQL connection



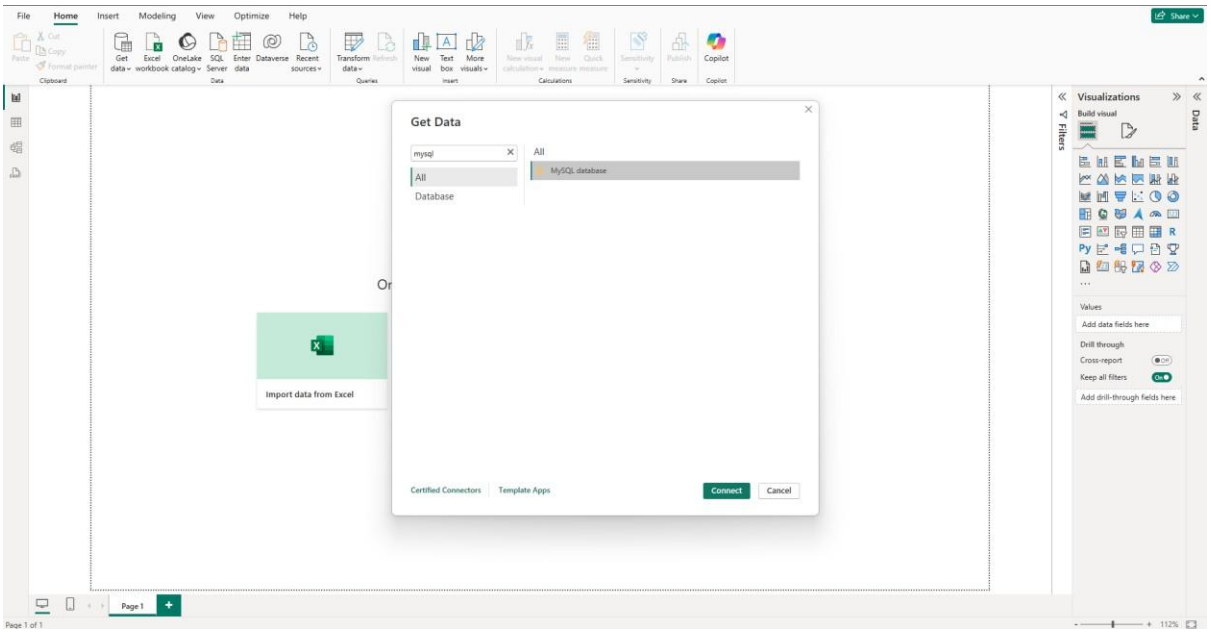
Step 5: Upload the SQL data into the MYSQL Workbench.



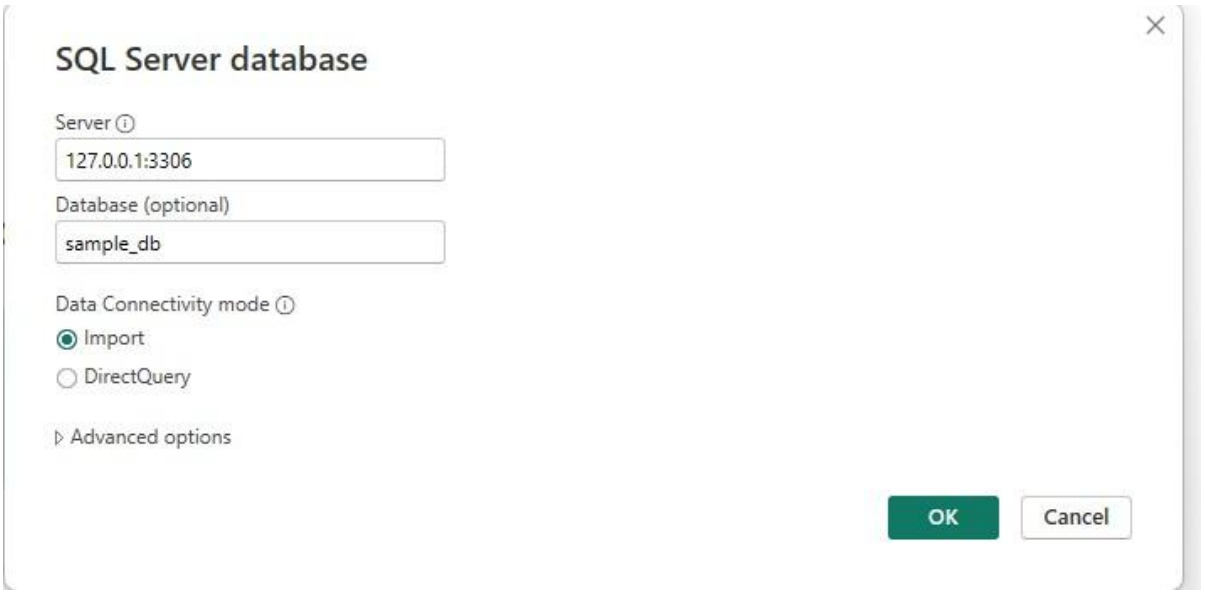
Step 6: Create database and insert the data into it.



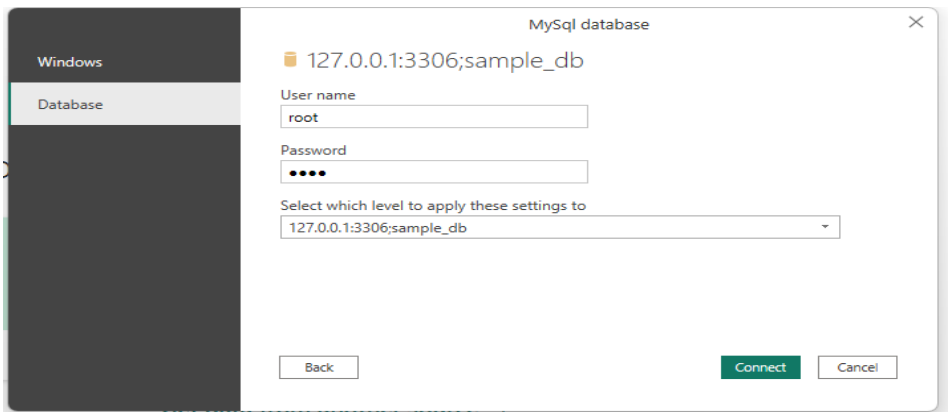
Step 7: Open PowerBi and Import the Database to PowerBi



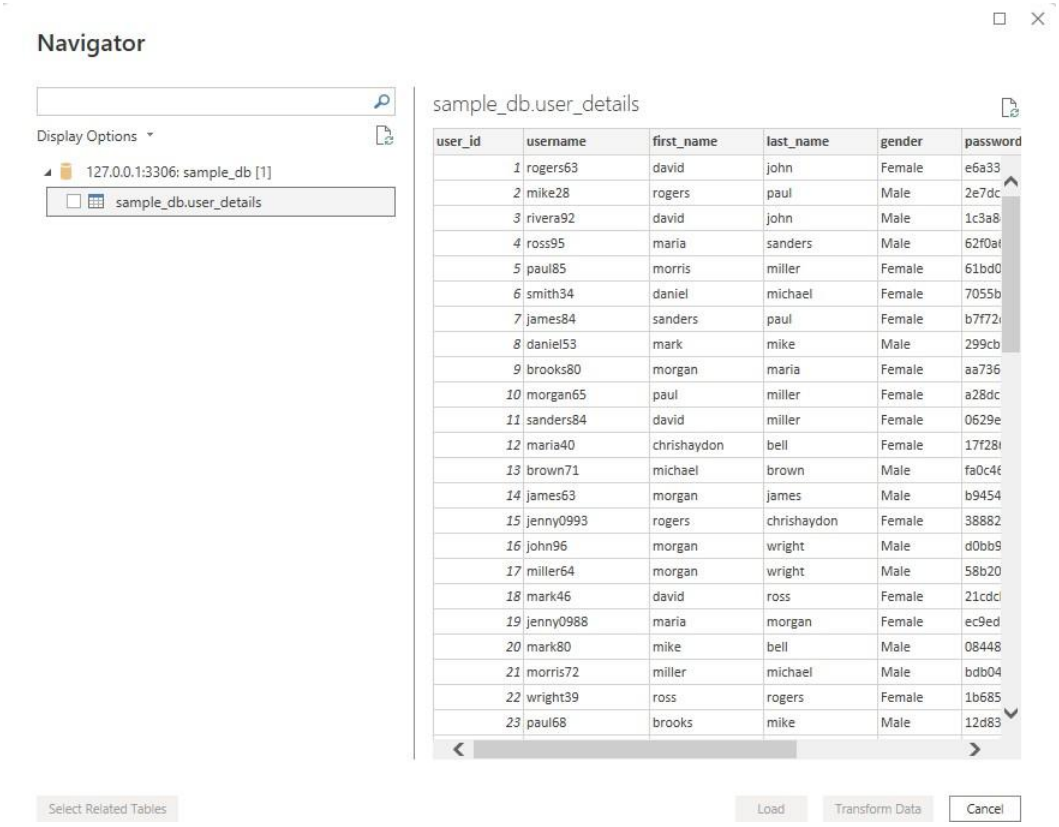
Step 8: Enter your Server and Database



Step 9: Enter your Username and Password.



Step 10: Use Navigator to Load and Transform the Data.



RESULT:

Thus, the procedure to import the Legacy data from SQLServer and loaded in the target system successfully and the output is verified

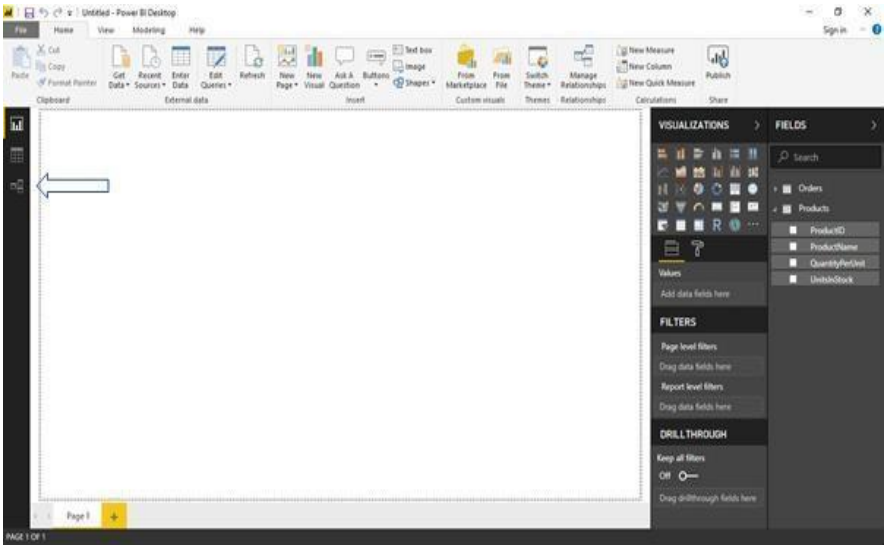
Ex.No: 3	Data Visualization from ETL Process
Date:	

AIM:

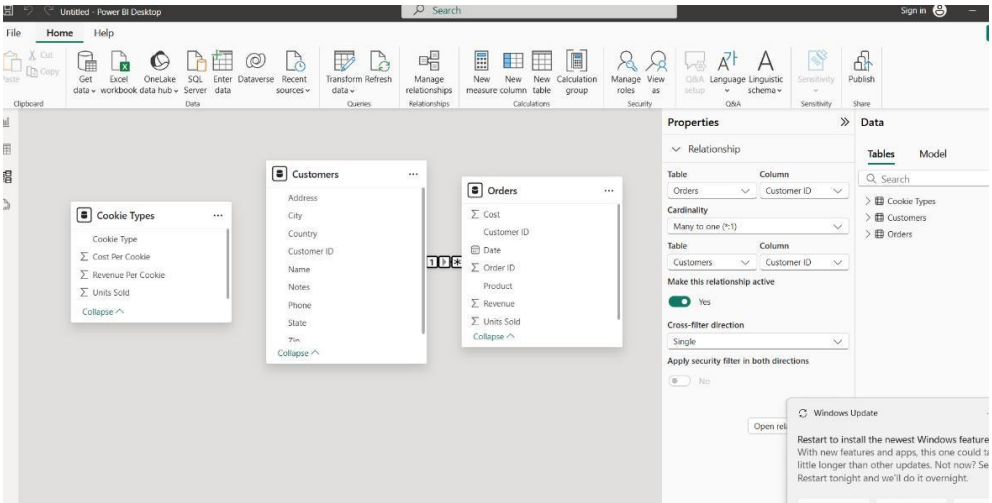
To perform the Extraction Transformation and Loading (ETL) process to construct the database in the SQL Server / Power BI.

PROCEDURE:

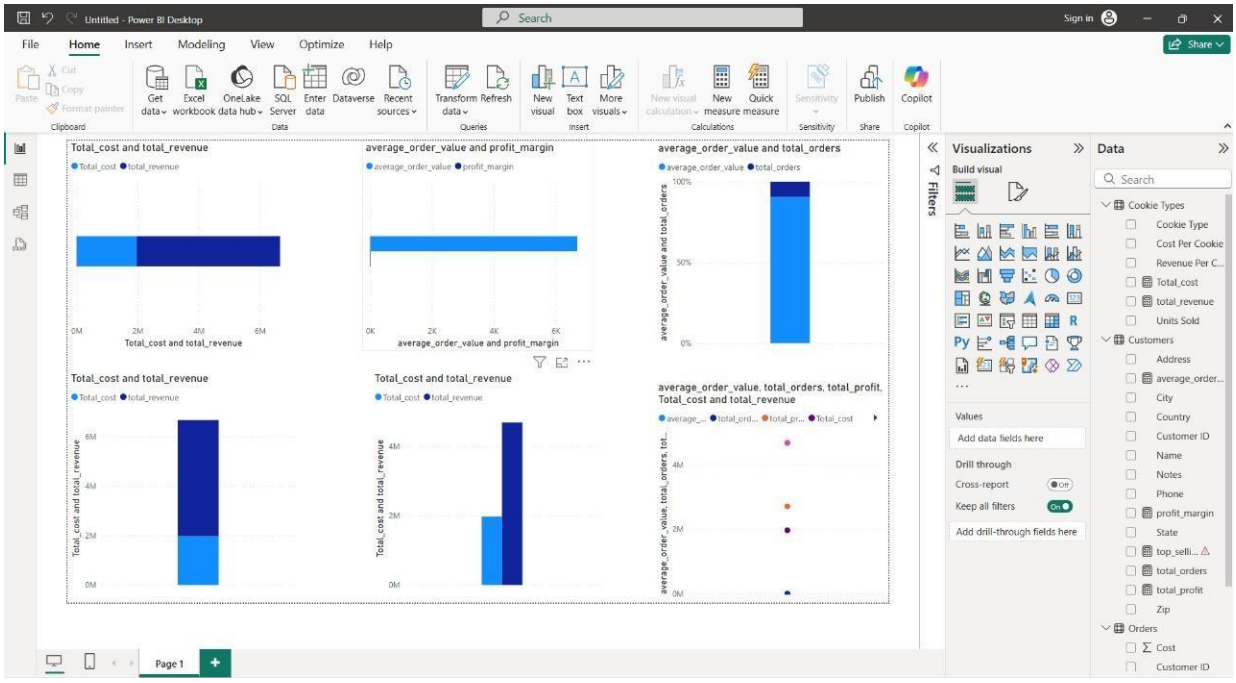
Step 1: Once the data is loaded, select the Model view and create relationship.

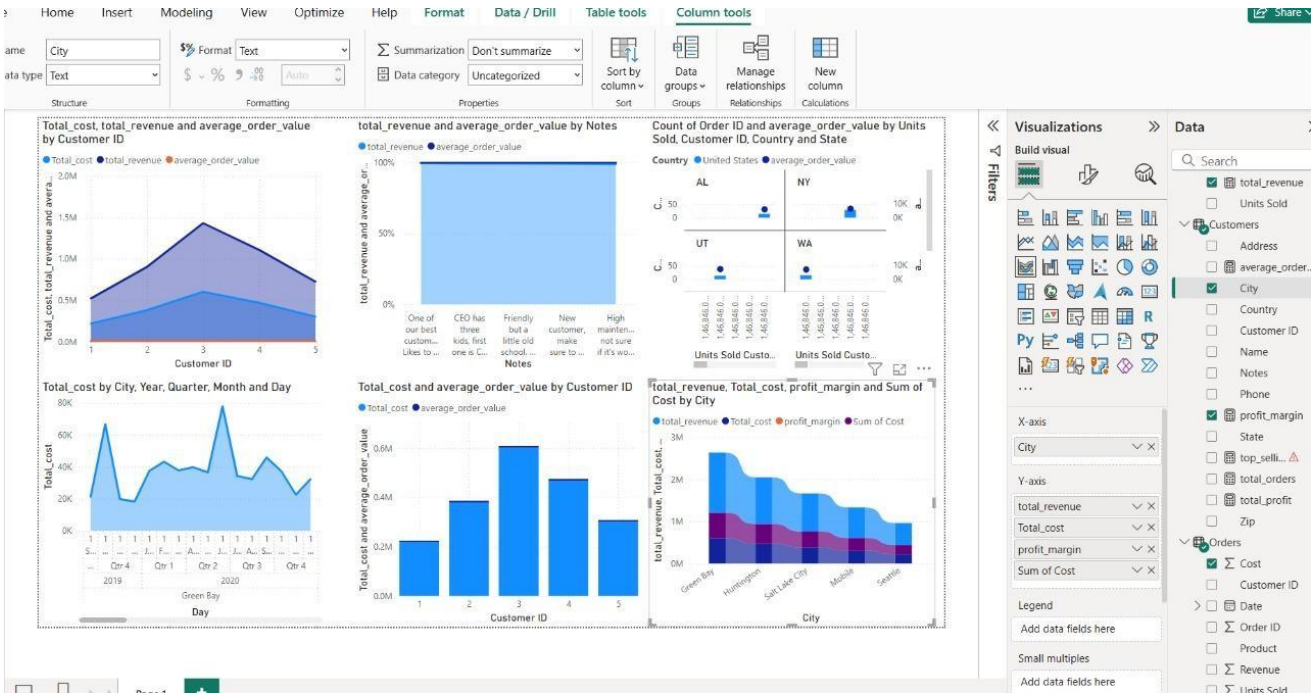


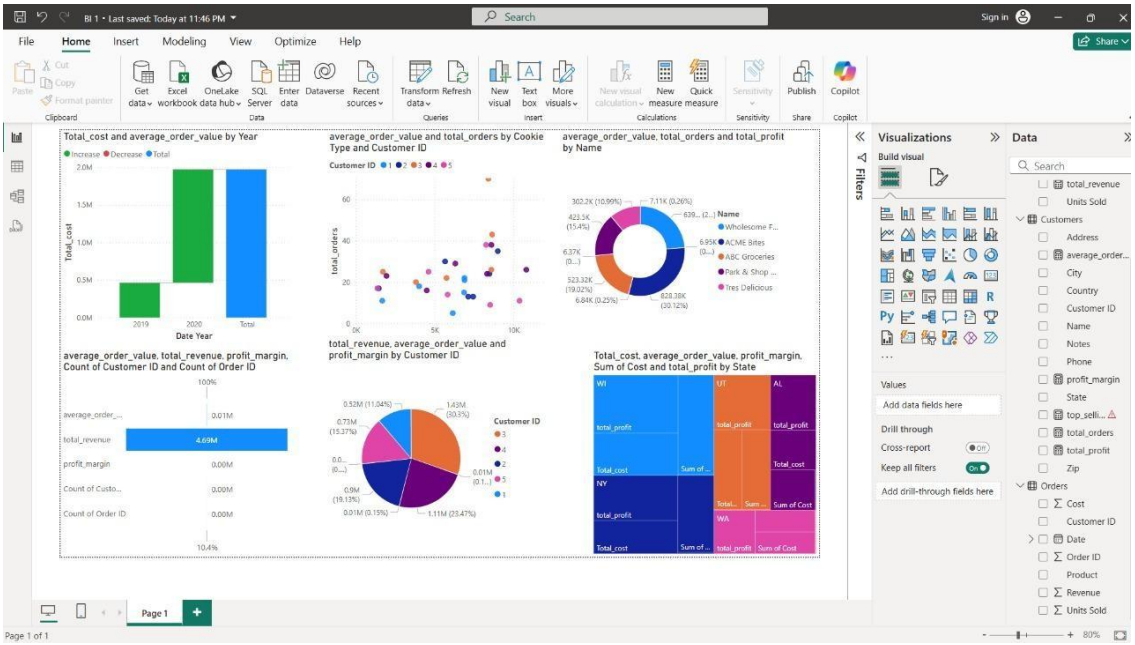
Step 2: We see the following, which visualizes the relationship between the queries.

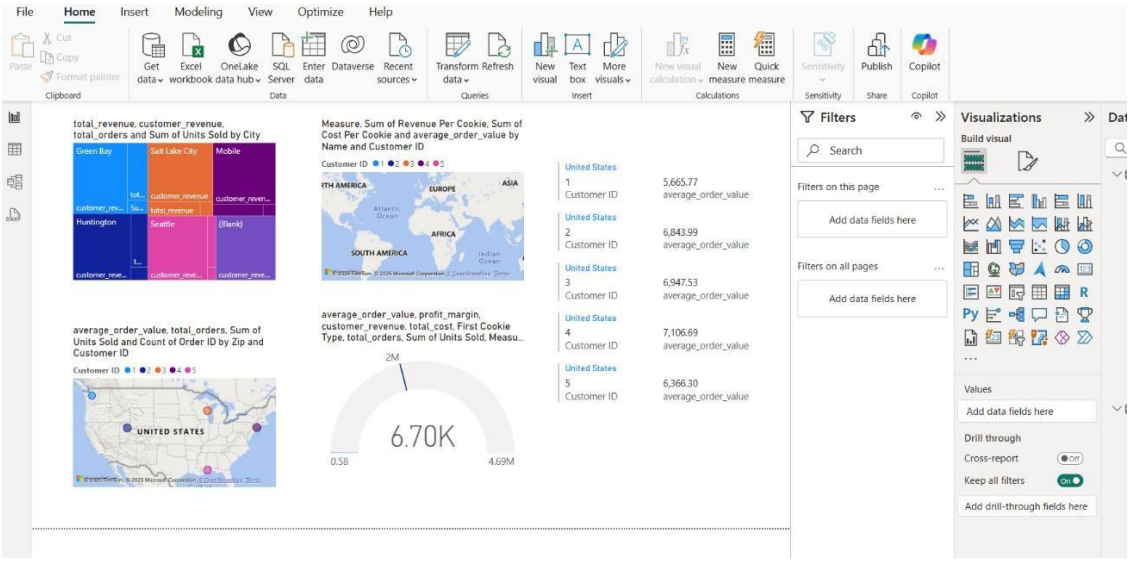


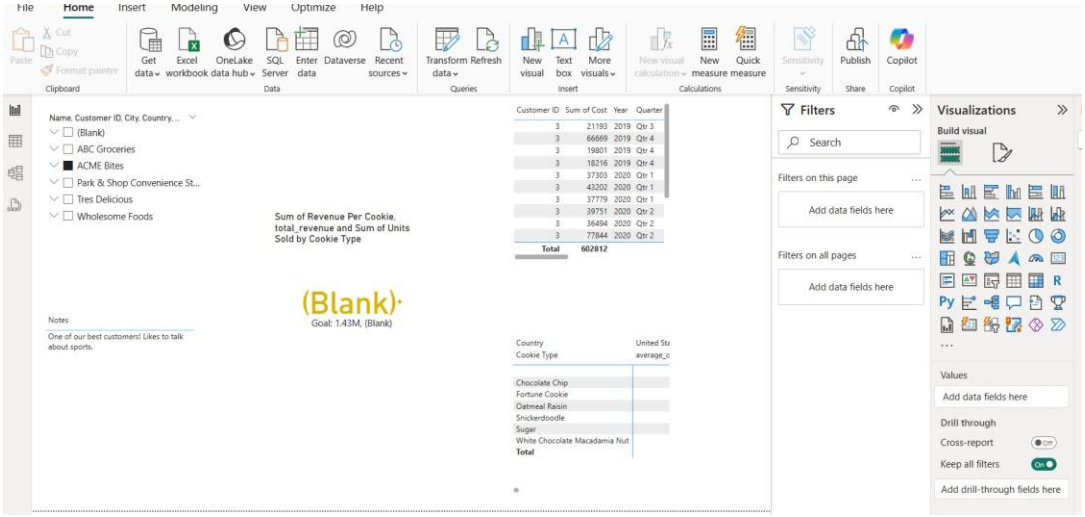
Step 3: After managing the relationships, create a dashboard using all graph visuals.

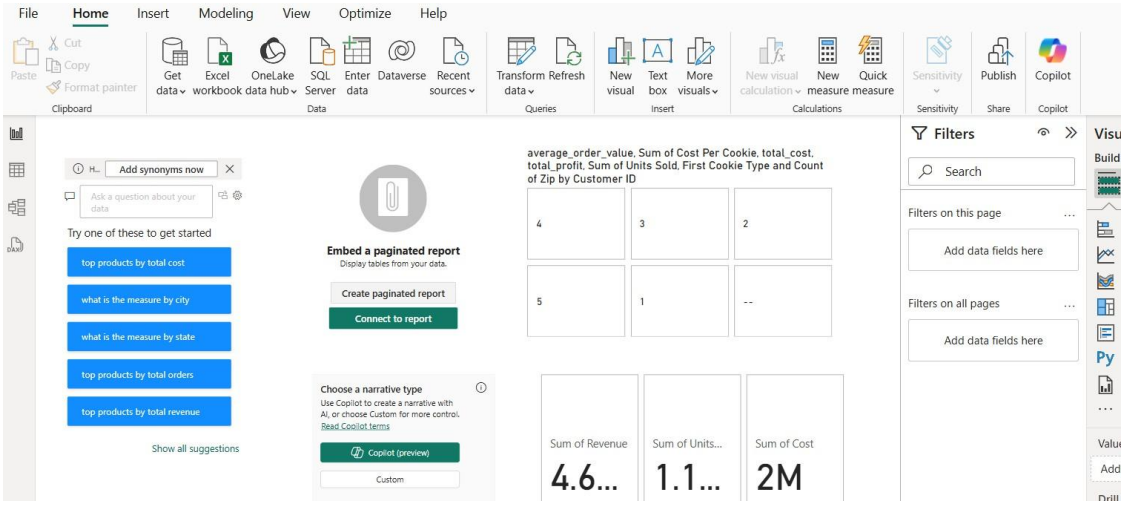


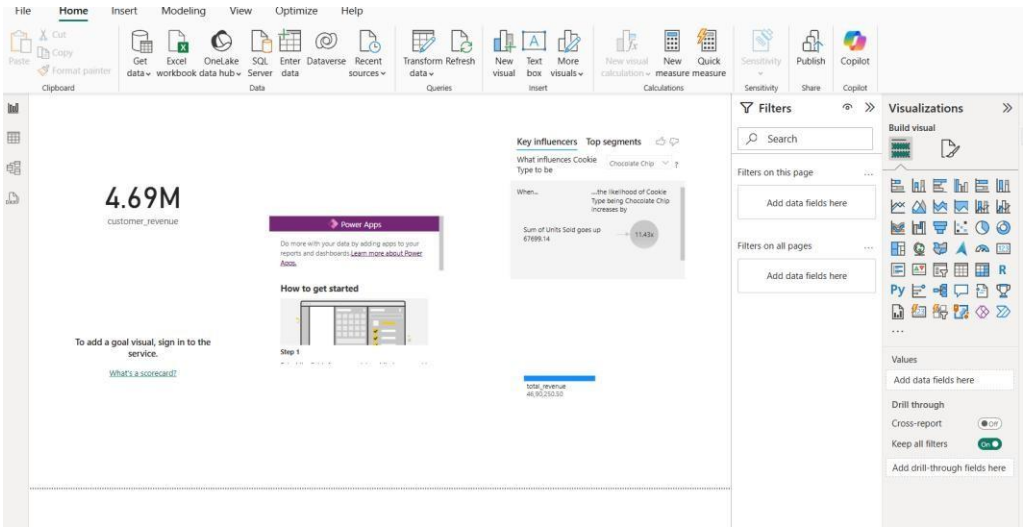



Total_cost and average_order_value by Year
average_order_value and total_orders by Cookie Type and Customer ID
average_order_value, total_orders and total_profit by Name
average_order_value, total_revenue, profit_margin, Count of Customer ID and Count of Order ID
total_revenue, average_order_value and profit_margin by Customer ID
Total_cost, average_order_value, profit_margin, Sum of Cost and total_profit by State
Visualizations
Data
Page 1


total_revenue, customer_revenue, total_orders and Sum of Units Sold by City
Measure, Sum of Revenue Per Cookie, Sum of Cost Per Cookie and average_order_value by Name and Customer ID
average_order_value, total_orders, Sum of Units Sold and Count of Order ID by Zip and Customer ID
average_order_value, profit_margin, customer_revenue, total_cost, First Cookie Type, total_orders, Sum of Units Sold, Measu...
6.70K
Filters
Visualizations


File
Home
Insert
Modeling
View
Optimize
Help
Filters
Search
Visualizations
Build visual
Filters on this page
Add data fields here
Filters on all pages
Values
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Sum of Revenue Per Cookie, total_revenue and Sum of Units Sold by Cookie Type
(Blank)
Goal: 1.43M, (Blank)


File
Home
Insert
Modeling
View
Optimize
Help
Filters
Search
Filters on this page
Add data fields here
Filters on all pages
Add data fields here
Add synonyms now
Try one of these to get started
top products by total cost
what is the measure by city
what is the measure by state
top products by total orders
top products by total revenue
Show all suggestions
Embed a paginated report
Display tables from your data.
Create paginated report
Connect to report
Choose a narrative type
Use Copilot to create a narrative with AI, or choose Custom for more control.
Read Copilot terms
Copilot (preview)
Custom
average_order_value, Sum of Cost Per Cookie, total_cost, total_profit, Sum of Units Sold, First Cookie Type and Count of Zip by Customer ID
Sum of Revenue
4.6...
Sum of Units...
1.1...
Sum of Cost
2M



RESULT:

Thus the Extraction, Transformation and Loading (ETL) process to construct the database in the Power BI was performed successfully and the output is verified

EX.NO : 4	CREATING A CUBE IN SQL SERVER 2012
DATE :	

AIM:

To create a OLAP cube with data warehouse fact tables and dimensions in SQL Server Management Studio.

PROCEDURE:

 # Creating Data Warehouse:

Let us execute our T-SQL Script to create data warehouse with fact tables, dimensions and populate them with appropriate test values.

Download T-SQL script attached with this article for creation of Sales Data Warehouse or download from this article **"Create First Data Warehouse**" and run it in your SQL Server.

Follow the given steps to run the query in **SSMS** (SQL Server Management Studio).

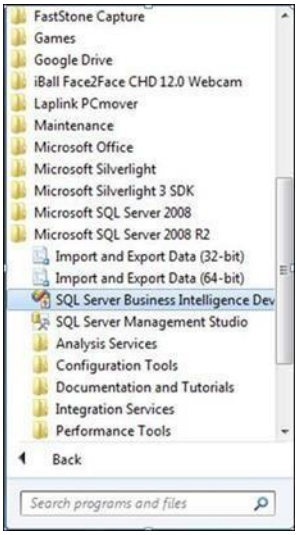
1. Open SQL Server Management Studio 2008
2. Connect Database Engine
3. Open **New Query** editor
4. Copy paste Scripts given below in various steps in new query editor window one by one
5. To run the given SQL Script, press **F5**
6. It will create and populate "Sales_DW" database on your SQL Server

 # Developing an OLAP Cube:

For creation of OLAP Cube in Microsoft BIDS Environment, follow the 10 easy steps given below.

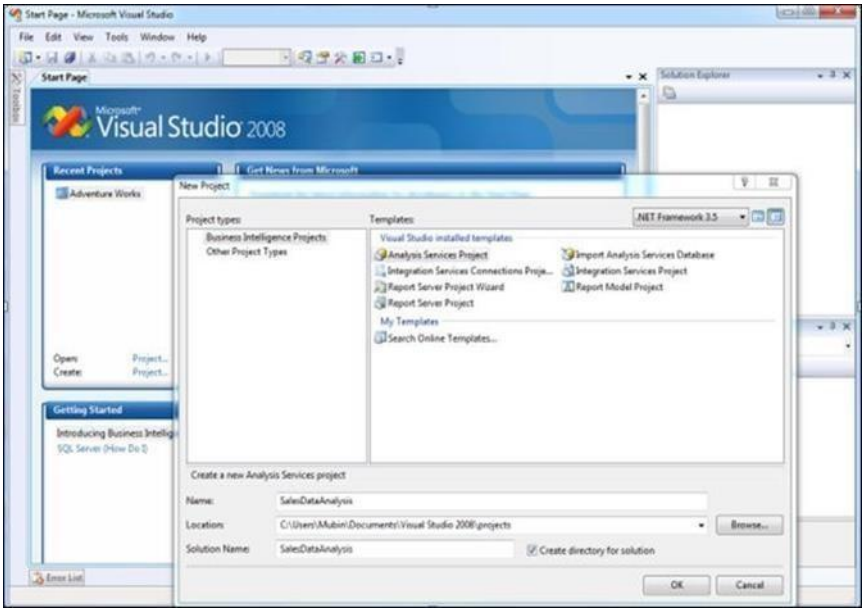
Step 1: Start BIDS Environment

Click on **Start Menu -> Microsoft SQL Server 2008 R2 ->** Click **SQL Server Business Intelligence Development Studio.**



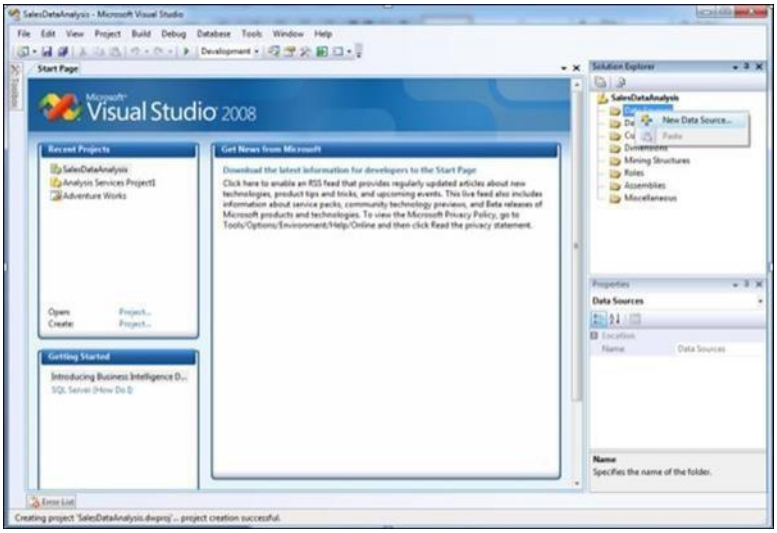
Step 2: Start Analysis Services Project

Click **File** -> **New** -> **Project** ->**Business Intelligence Projects** ->select **Analysis Services Project**-> Assign Project **Name** -> Click **OK**



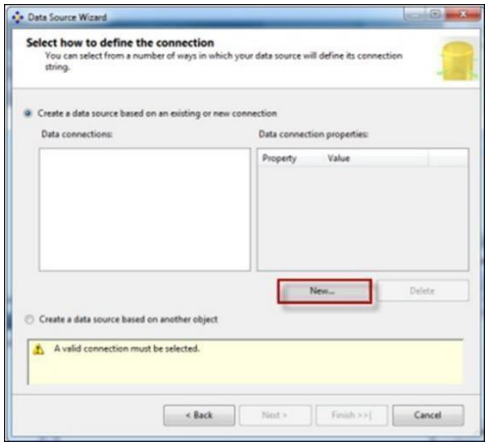
Step 3: Creating New Data Source

- In Solution Explorer, Right click on **Data Source** -> Click **New Data Source**



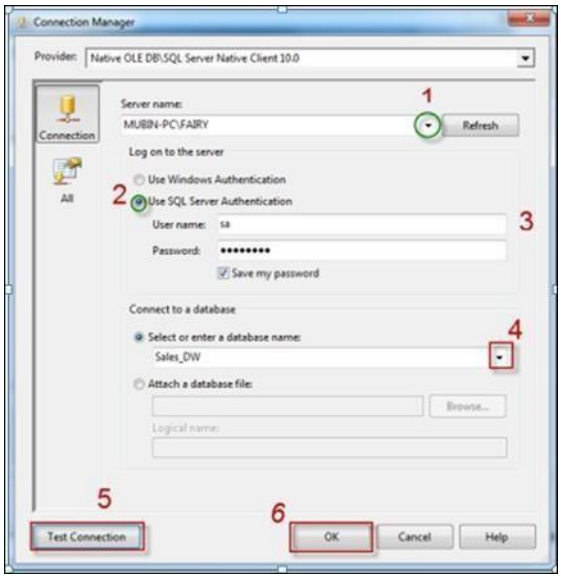
- Click on **Next**

- Click on New **Button**

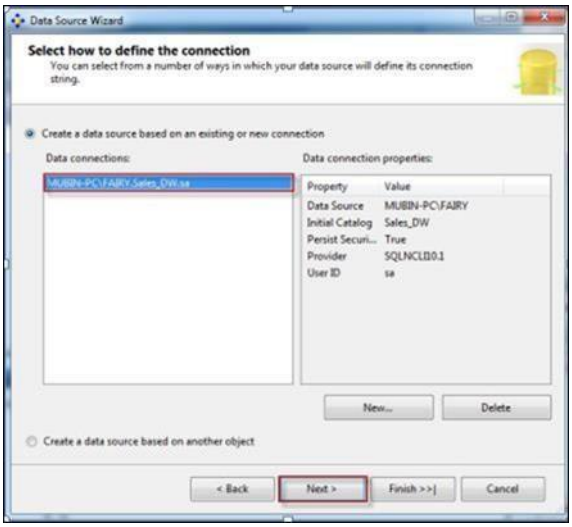


- Creating **New connection**

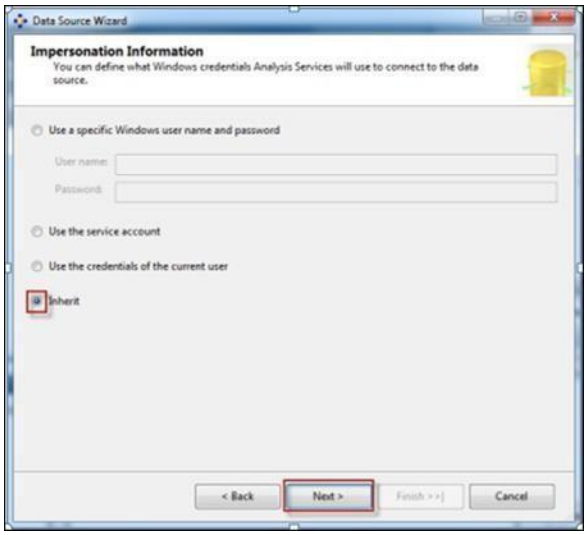
1. Specify Your SQL **Server Name** where your Data Warehouse was created
2. Select Radio Button according to your **SQL Server Authentication** mode
3. Specify your **Credentials** using which you can connect to your SQL Server
4. Select database Sales_DW.
5. Click on **Test Connection** and verify for its success click **OK**.



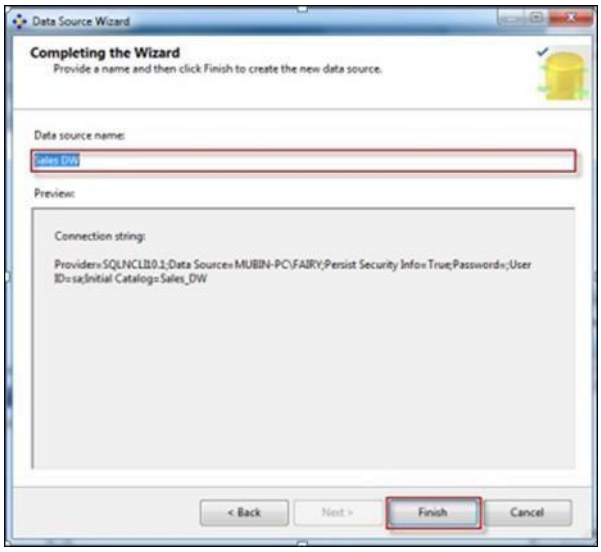
- Select Connection created in **Data Connections**-> Click **Next**

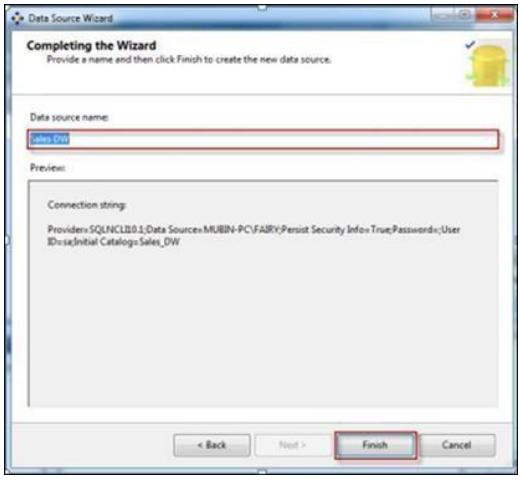


- Select Option **Inherit**



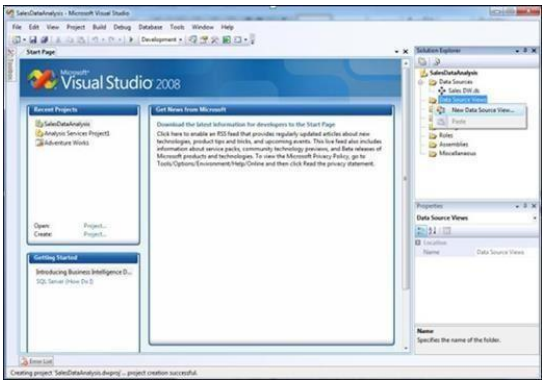
- Assign Data Source **Name** -> Click **Finish**



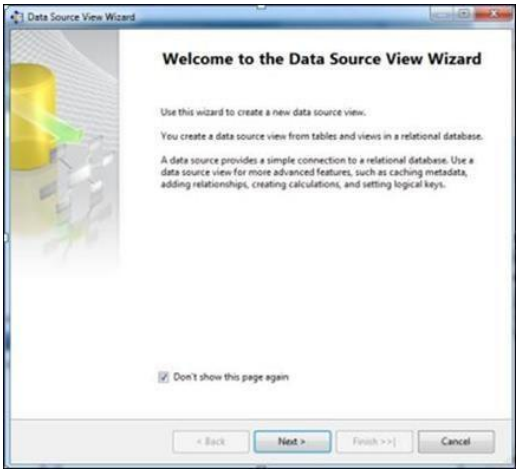


Step 4: Creating New Data Source View

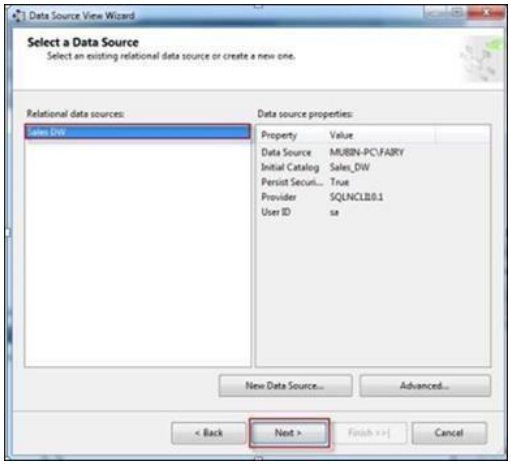
- In the Solution Explorer, Right Click on **Data Source View** -> Click on **New Data Source View**



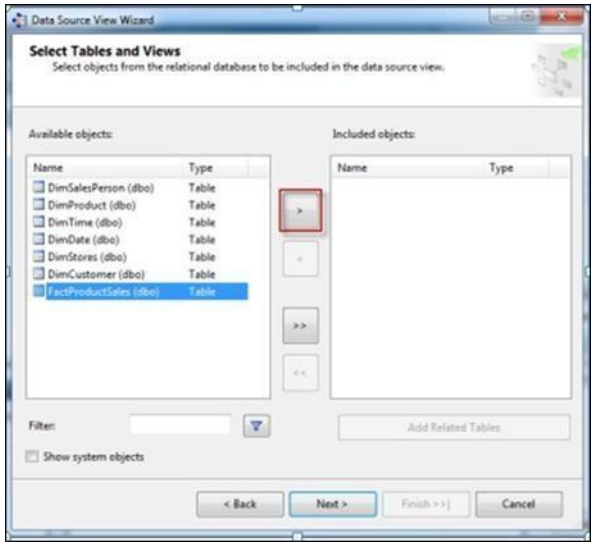
- Click **Next**



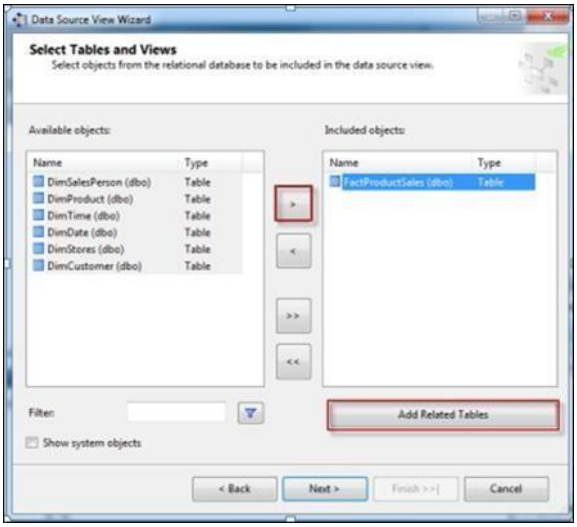
- Select **Relational Data Source** we have created previously (Sales_DW)-> Click **Next**



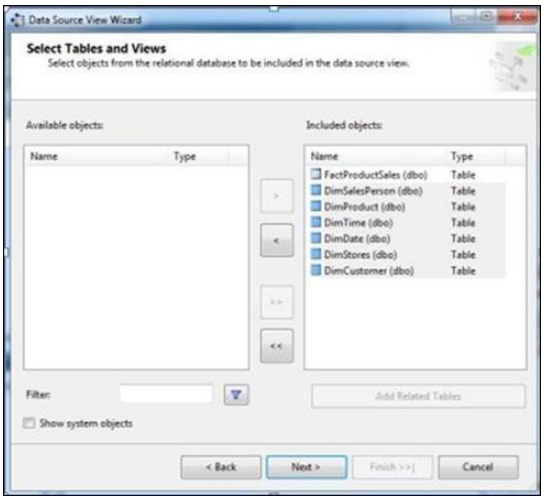
- First move your **Fact Table** to the right side to include in object list.



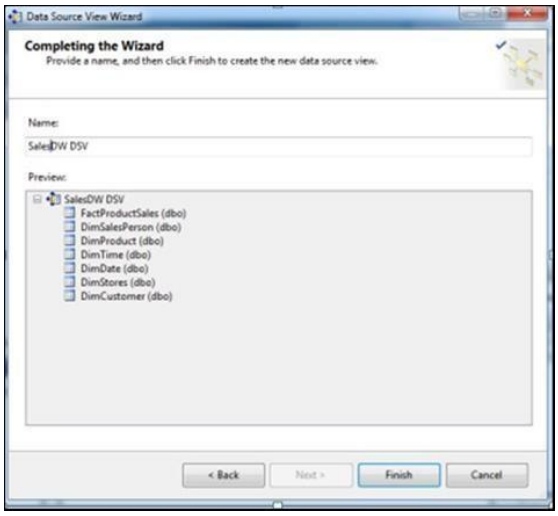
- Select FactProductSales Table -> Click on Arrow Button to move the selected object to Right Pane.
- Now to **add dimensions** which are **related** to your **Fact Table**, follow the given steps: Select **Fact Table** in Right Pane (Fact product Sales) -> Click On **Add Related Tables**



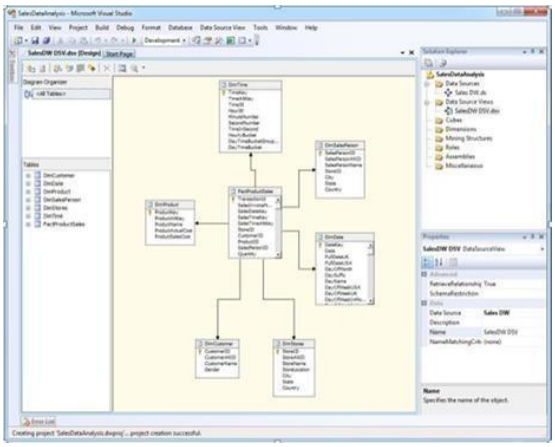
- It will add all associated dimensions to your Fact table as per relationship specified in your SQL DW (Sales_DW).
- Click **Next**.



- Assign **Name (SalesDW DSV)->** Click **Finish**

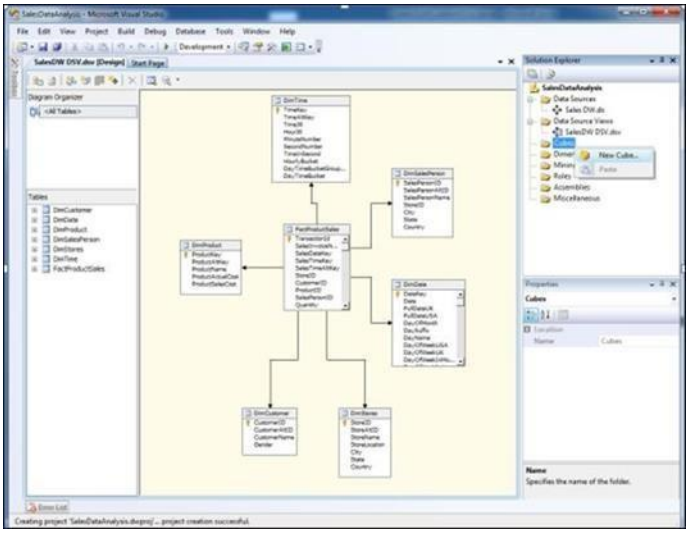


- Now Data Source View is ready to use.



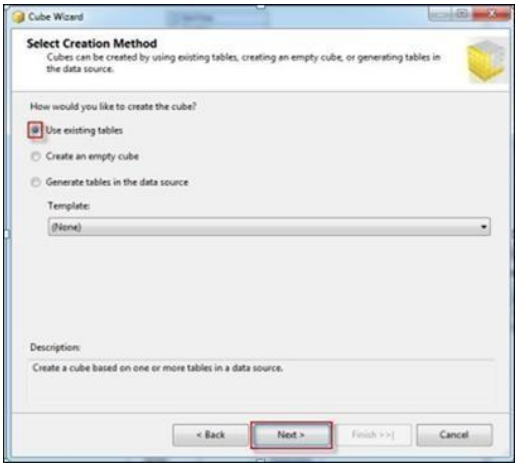
Step 5: Creating New Cube

- In Solution Explorer -> Right Click on **Cube->** Click **New Cube**

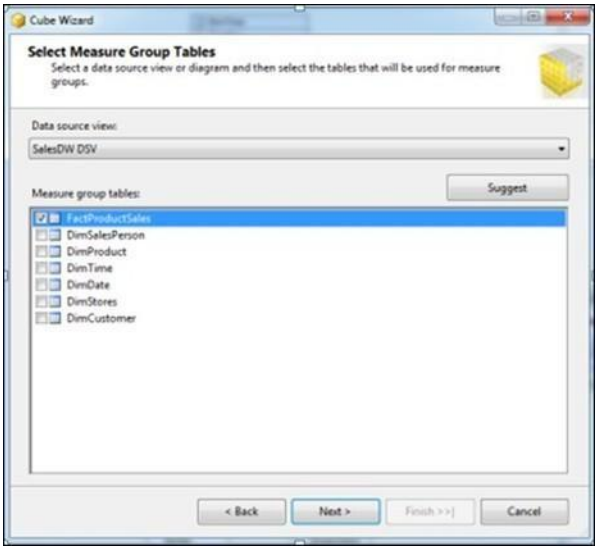


- Click **Next**

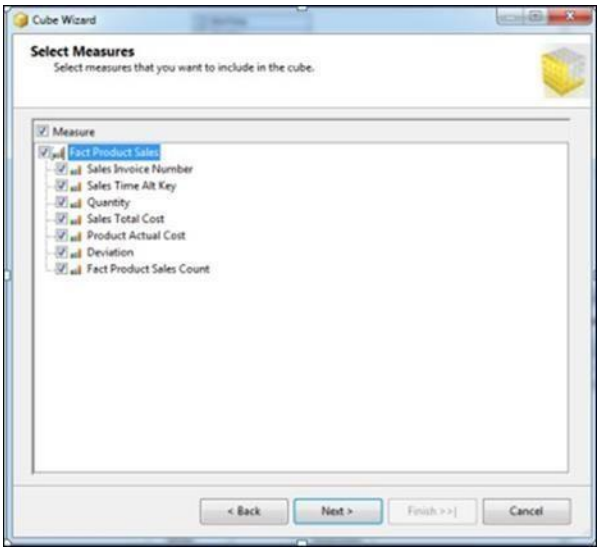
- Select Option **Use existing Tables ->** Click **Next**



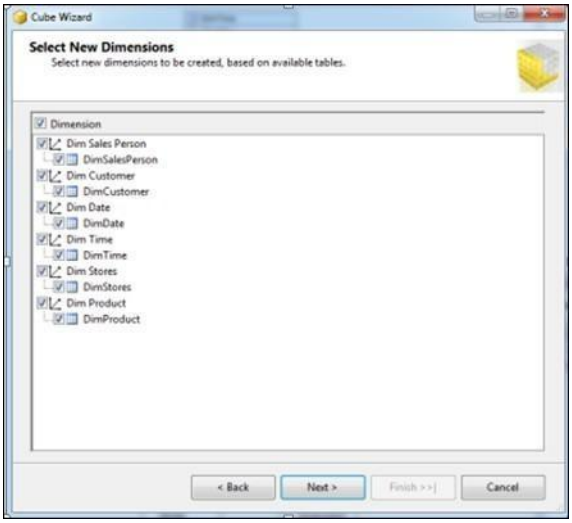
- Select Fact Table Name from **Measure Group Tables (FactProductSales) ->** Click **Next**



- Choose **Measures** from the List which you want to place in your Cube --> Click **Next**

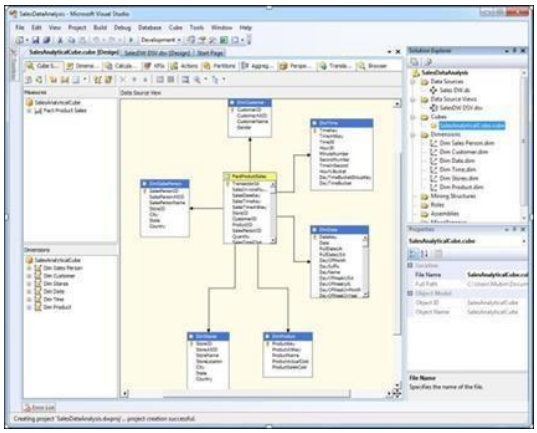


- Select All **Dimensions** here which are associated with your Fact Table-> Click **Next**



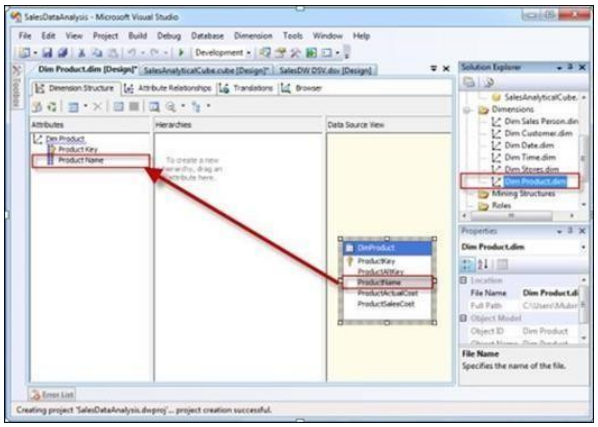
- Assign **Cube Name** (SalesAnalyticalCube) -> Click **Finish**

- Now your Cube is ready, you can see the newly created cube and dimensions added in your solution explorer.



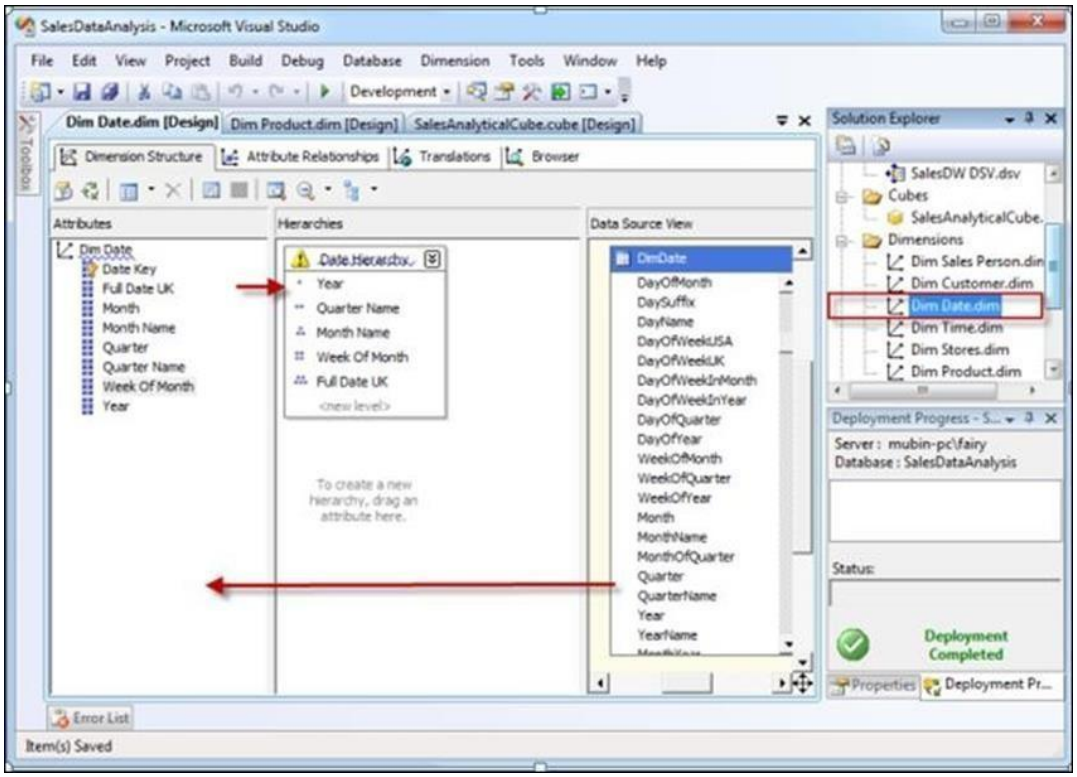
Step 6: Dimension Modification

In Solution Explorer, double click on dimension **Dim Product ->** Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side.



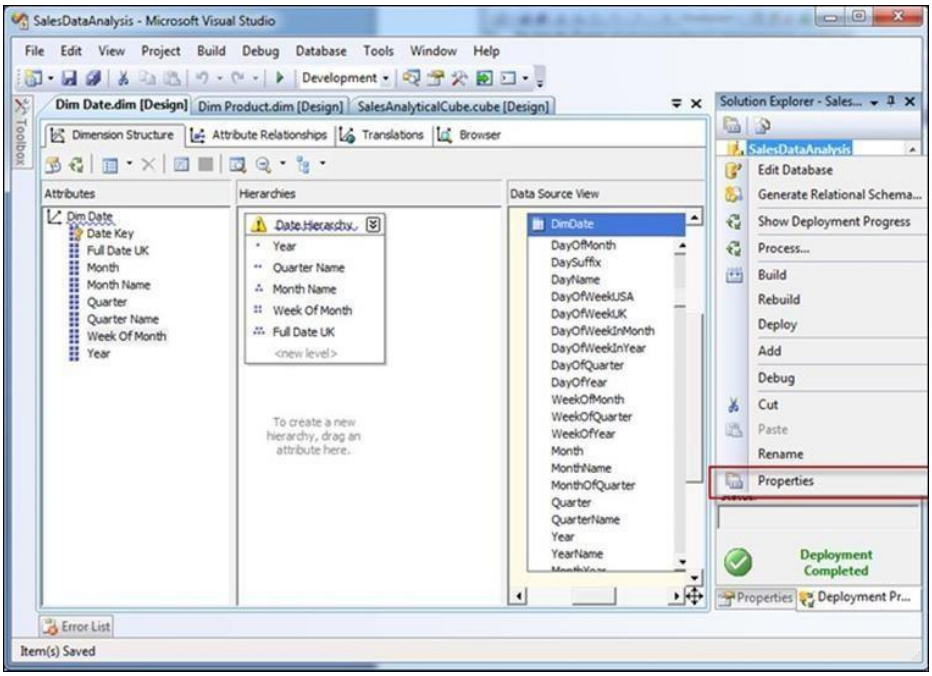
Step 7: Creating Attribute Hierarchy in Date Dimension

- Double click On **Dim Date** dimension -> Drag and Drop Fields from Table shown in Data Source View to Attributes-> Drag and Drop attributes from leftmost pane of attributes to middle pane of Hierarchy.
- Drag fields in sequence from Attributes to Hierarchy window (Year, Quarter Name, Month Name, Week of the Month, Full Date UK)

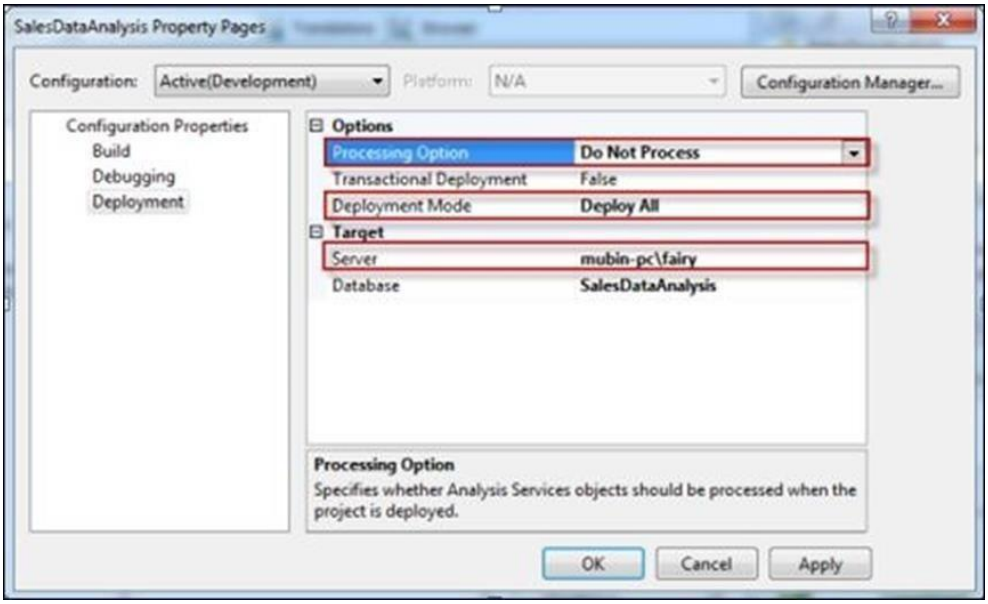


Step 8: Deploy the Cube

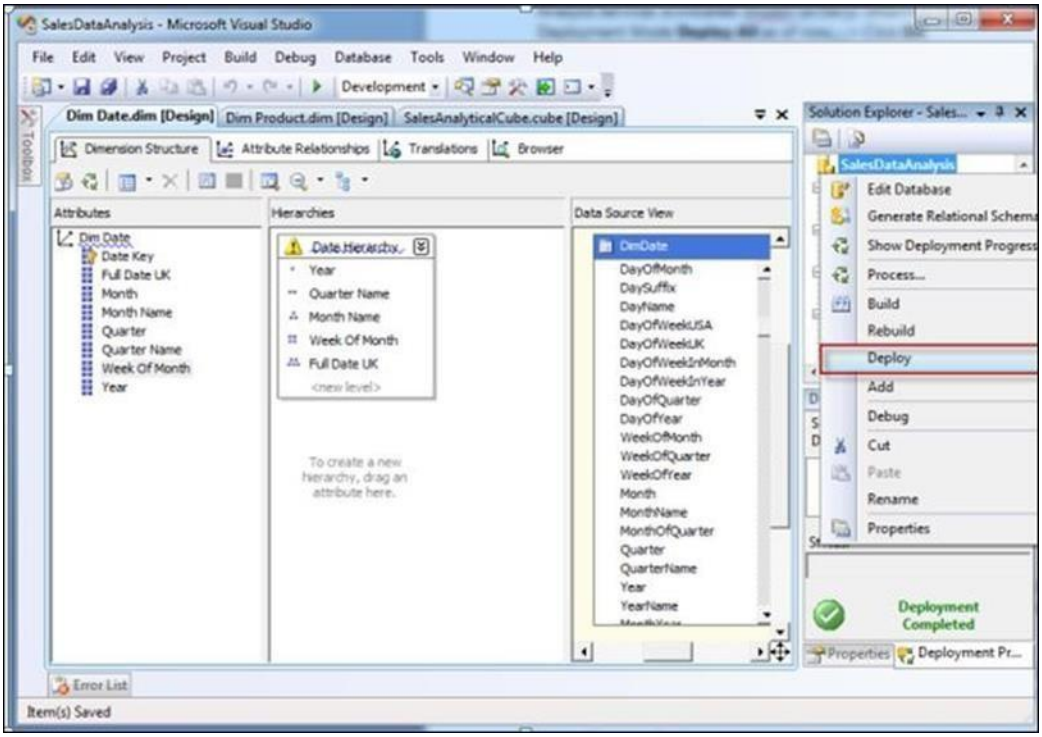
- In Solution Explorer, right click on Project Name (SalesDataAnalysis) -- > Click **Properties**



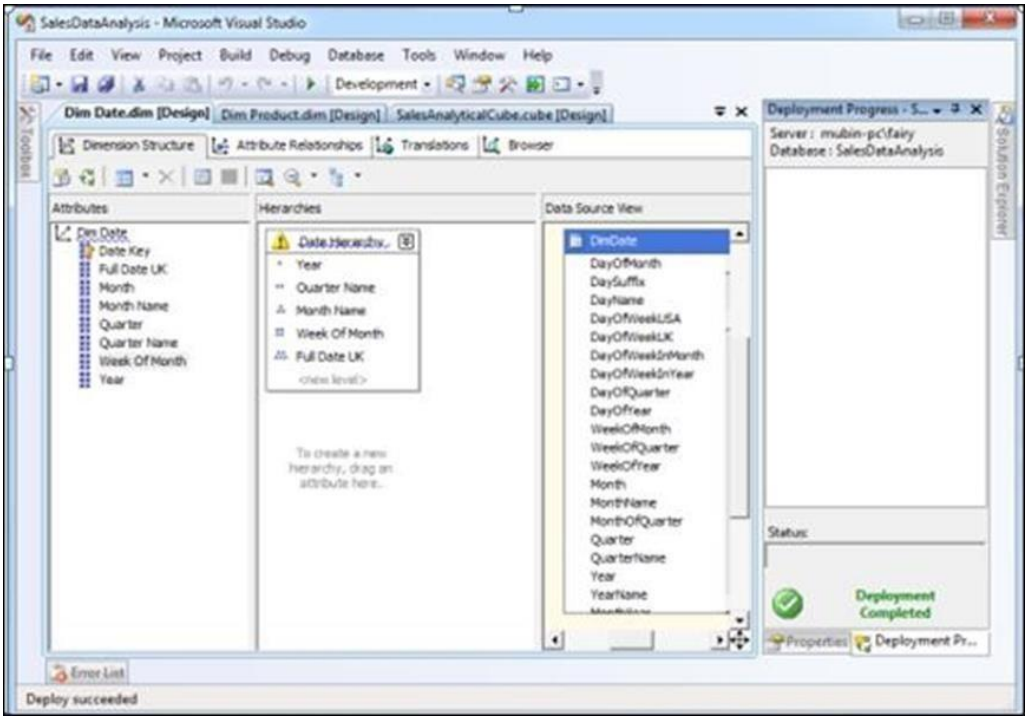
- Set **Deployment Properties** First
- In Configuration Properties, Select Deployment-> Assign Your SQL Server Instance Name Where Analysis Services Is Installed (mubin-pc\fairy) (Machine Name\Instance Name) -> Choose Deployment Mode Deploy All as of now ->Select Processing Option Do Not Process -> Click OK



- In Solution Explorer, right click on **Project Name** (SalesDataAnalysis) -- > Click **Deploy**

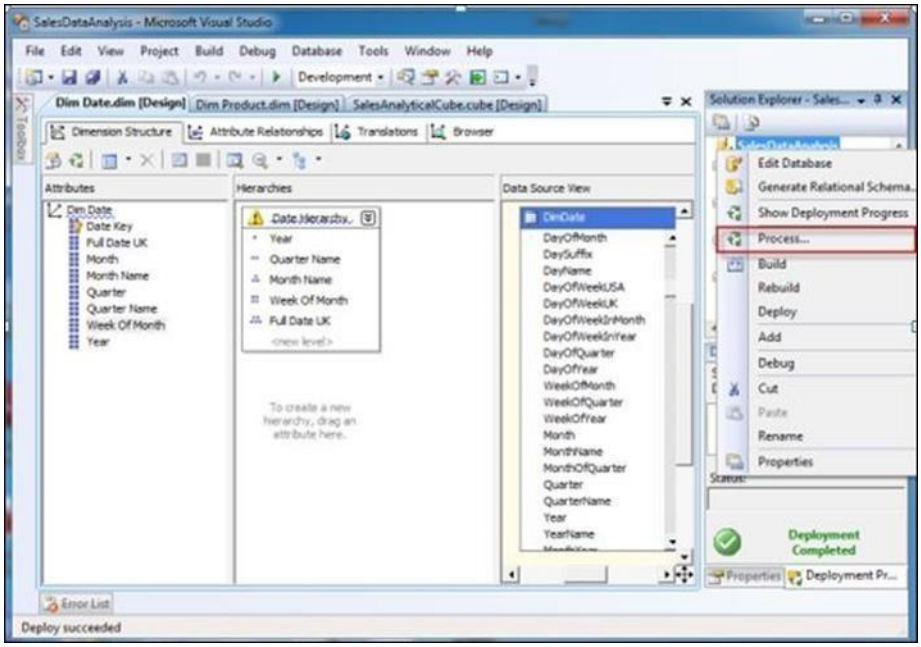


- Once Deployment will finish, you can see the message **Deployment Completed** in deployment Properties.

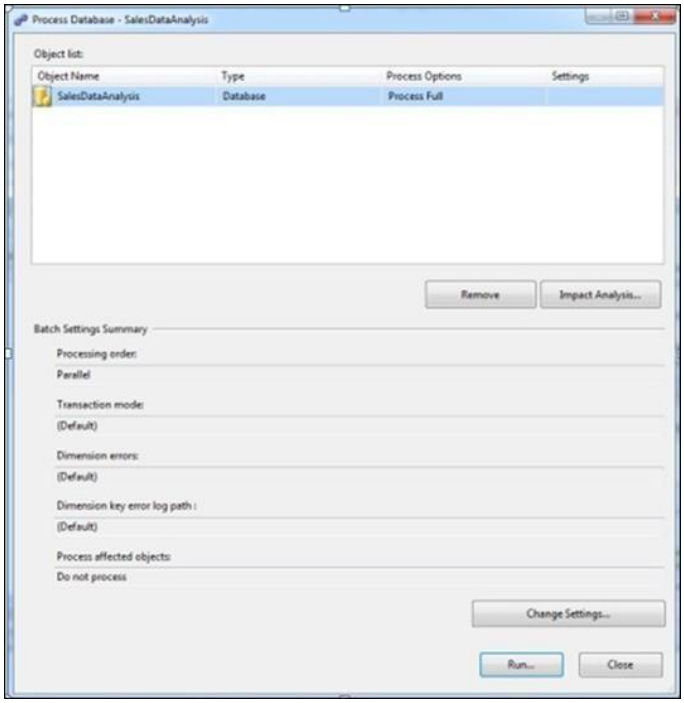


Step 9: Process the Cube

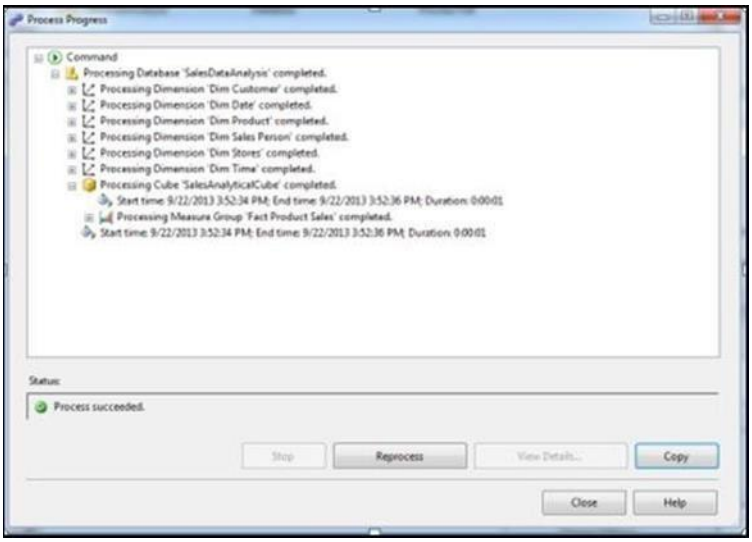
- In Solution Explorer, right click on Project Name (SalesDataAnalysis) -- > Click **Process**



- Click on **Run** button to process the Cube



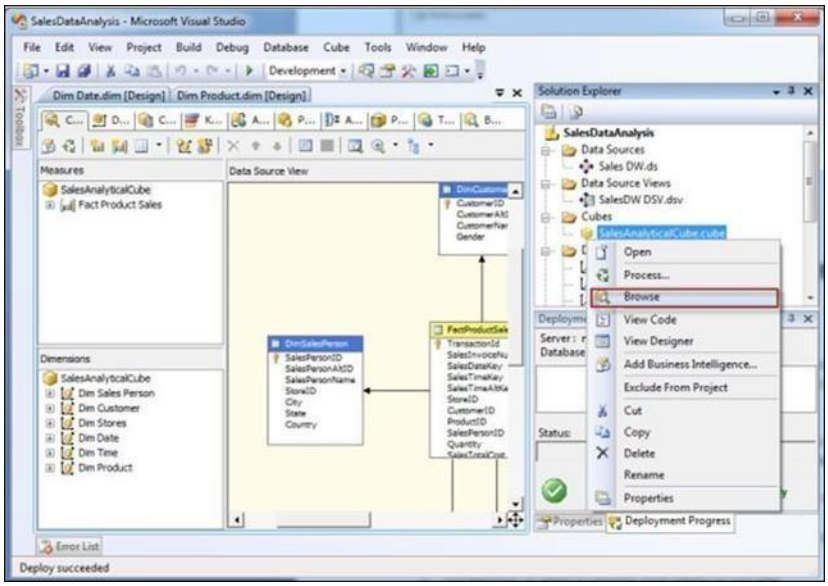
- Once processing is complete, you can see **Status** as **Process Succeeded** -->Click **Close** to close both the open windows for processing one after the other.



Step 10: Browse the Cube for Analysis

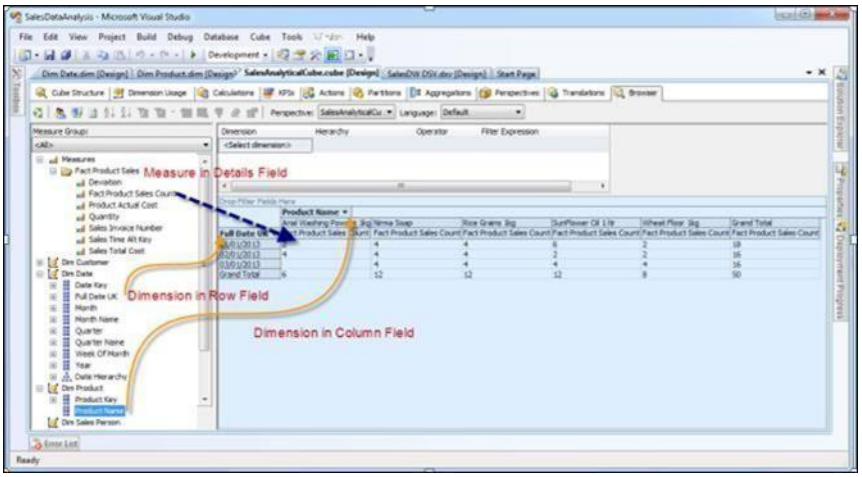
- In Solution Explorer, right click on Cube Name (SalesDataAnalysisCube) -- > Click

Browse



- Drag and drop measures in to Detail fields, & Drag and Drop Dimension Attributes in Row Field or Column fields.
- Now to **Browse Our Cube**

1. Product Name Drag & Drop into Column
2. Full Date UK Drag & Drop into Row Field
3. FactProductSalesCount Drop this measure in Detail area



RESULT:

Thus the OLAP cube with data warehouse fact tables and dimensions were created, deployed and processed in SQL Server Management Studi

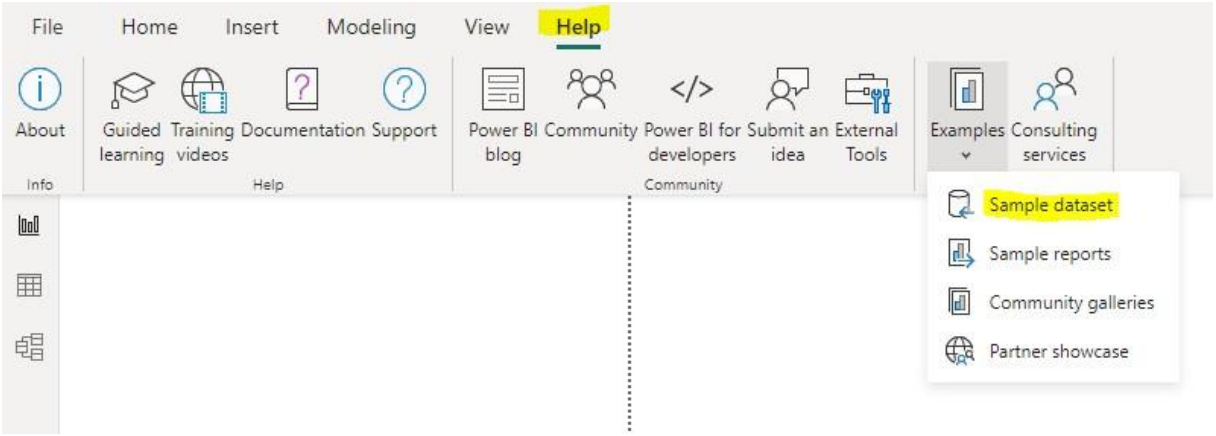
EX.NO: 5	APPLY THE WHAT – IF ANALYSIS FOR DATA VISUALIZATION TO DESIGN AND GENERATE NECESSARY REPORTS BASED ON THE DATA WAREHOUSE DATA
DATE:	

AIM:

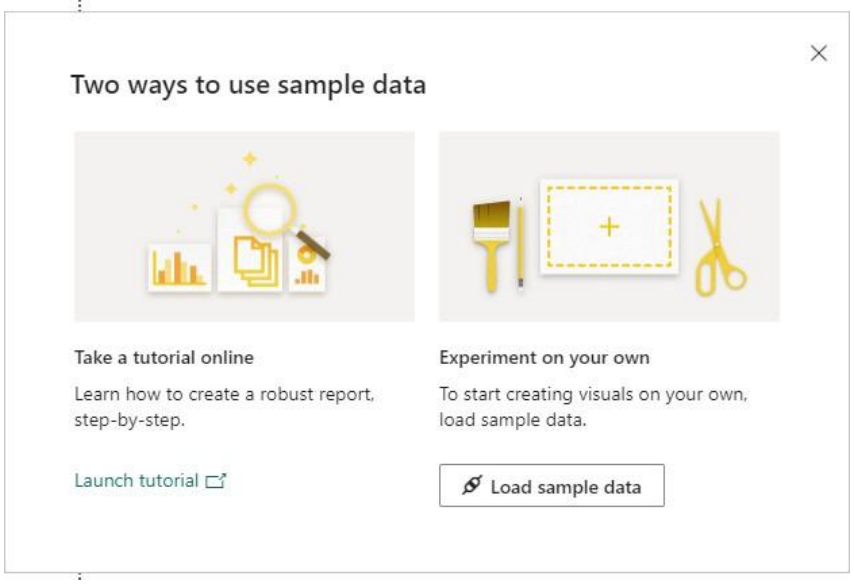
To apply the what – If analysis for Data Visualization to design and generate necessary reports based on the data warehouse data in Excel.

PROCEDURE:

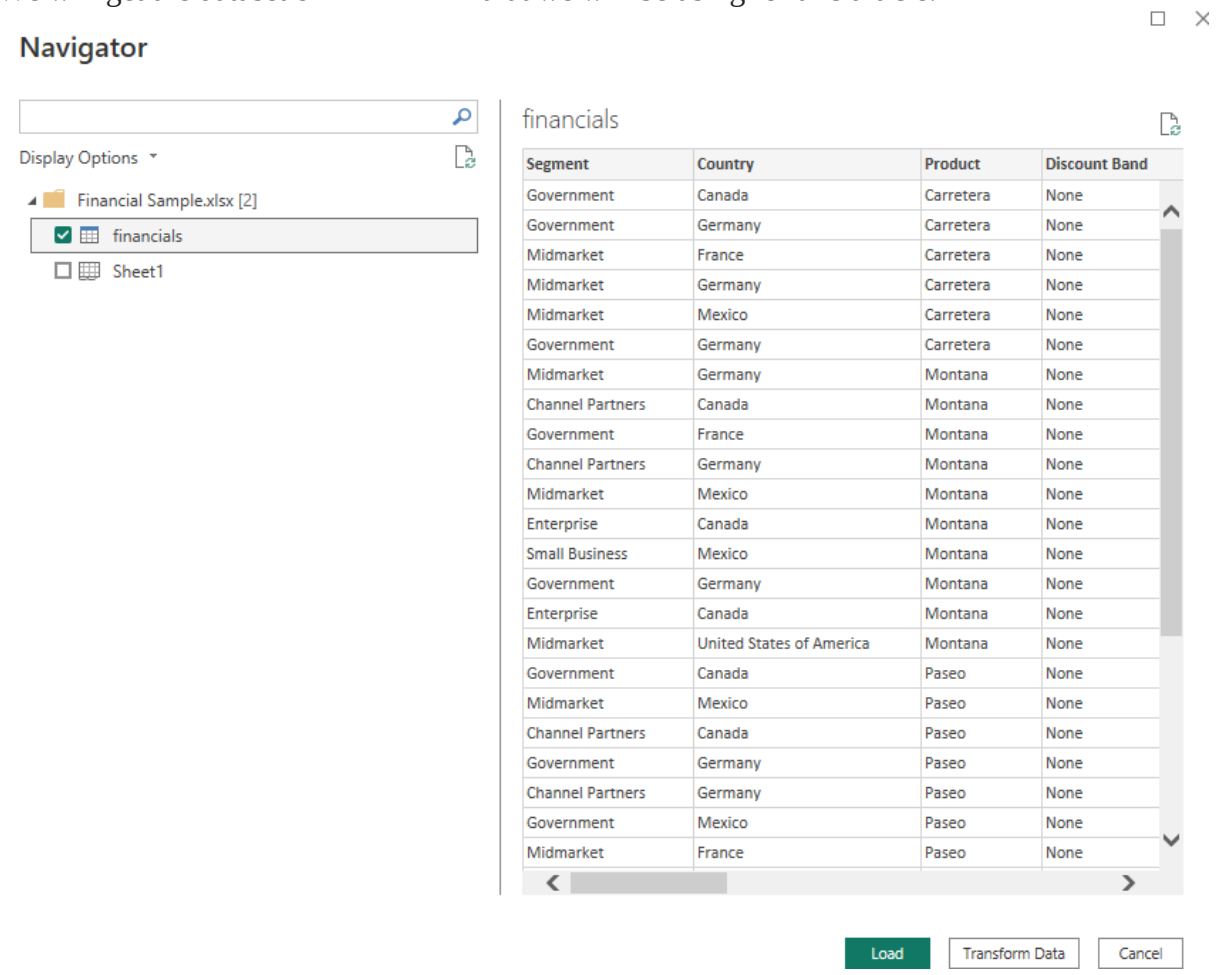
Step 1: Go to Power BI Desktop, click 'Help' , then choose the icon 'Examples' and press on 'Sample dataset'.



Step 2: The box below appears after selecting "Sample dataset." Next, select "Load sample data."



We will get the dataset of ‘financials’ that we will be using for this article.



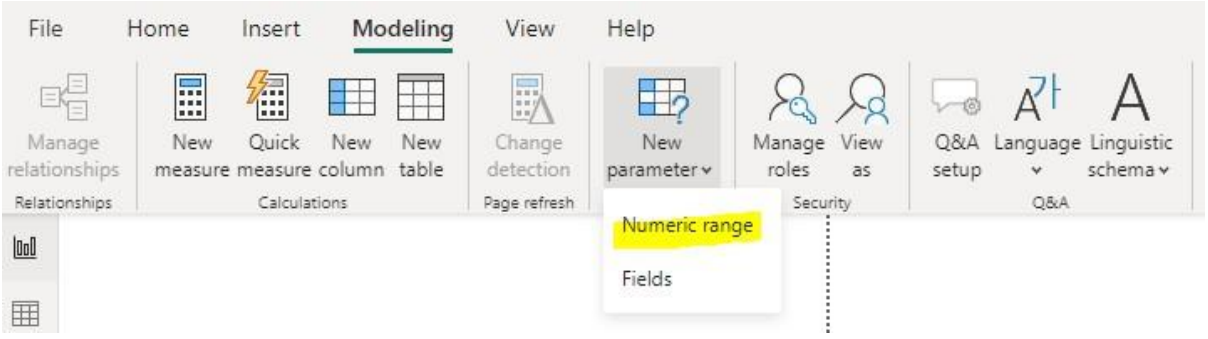
Segment	Country	Product	Discount Band
Government	Canada	Carretera	None
Government	Germany	Carretera	None
Midmarket	France	Carretera	None
Midmarket	Germany	Carretera	None
Midmarket	Mexico	Carretera	None
Government	Germany	Carretera	None
Midmarket	Germany	Montana	None
Channel Partners	Canada	Montana	None
Government	France	Montana	None
Channel Partners	Germany	Montana	None
Midmarket	Mexico	Montana	None
Enterprise	Canada	Montana	None
Small Business	Mexico	Montana	None
Government	Germany	Montana	None
Enterprise	Canada	Montana	None
Midmarket	United States of America	Montana	None
Government	Canada	Paseo	None
Midmarket	Mexico	Paseo	None
Channel Partners	Canada	Paseo	None
Government	Germany	Paseo	None
Channel Partners	Germany	Paseo	None
Government	Mexico	Paseo	None
Midmarket	France	Paseo	None

Steps for creating what-if parameters.

Step 1: Create a parameter.

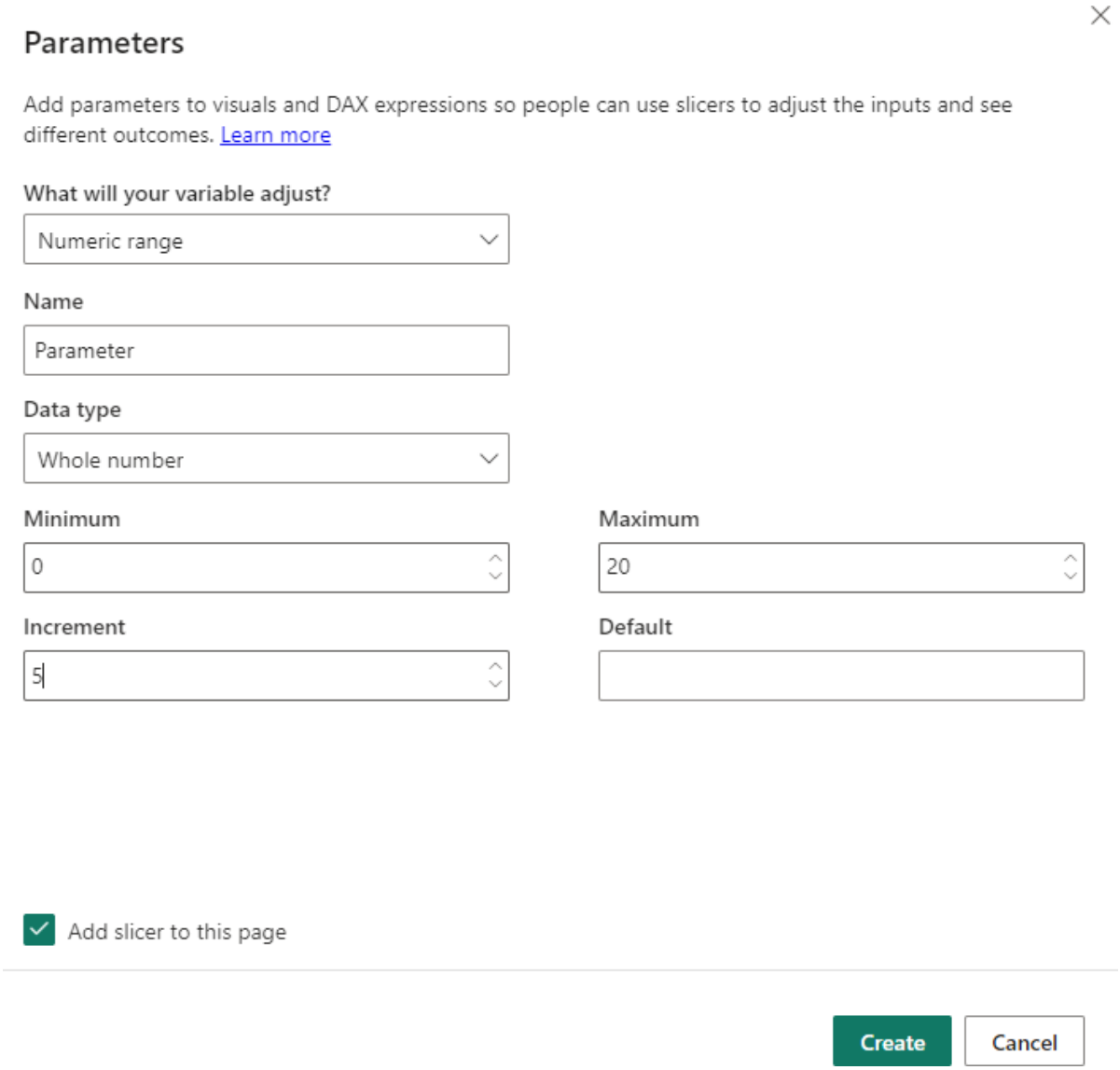
First, go to the tab “Modeling” and click on “New parameter”. Then choose “Numeric range”.

- If we have a strategy to set promotions for clients by offering discounts (such as 5%, 10%, 15%, and 20%), we will compare the number of “Sales” for this offer. Therefore, the “**Numeric range**” was chosen for that purpose.

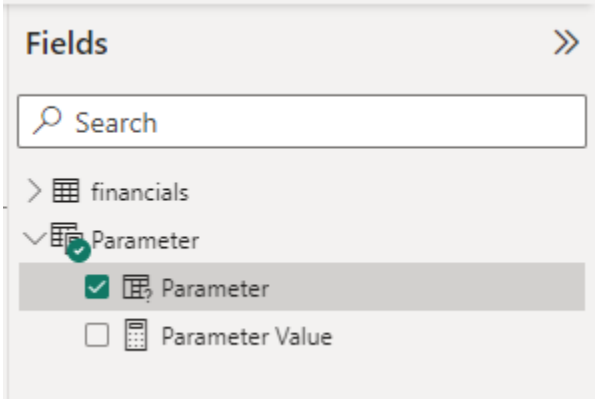


We can choose the value for each parameters below.

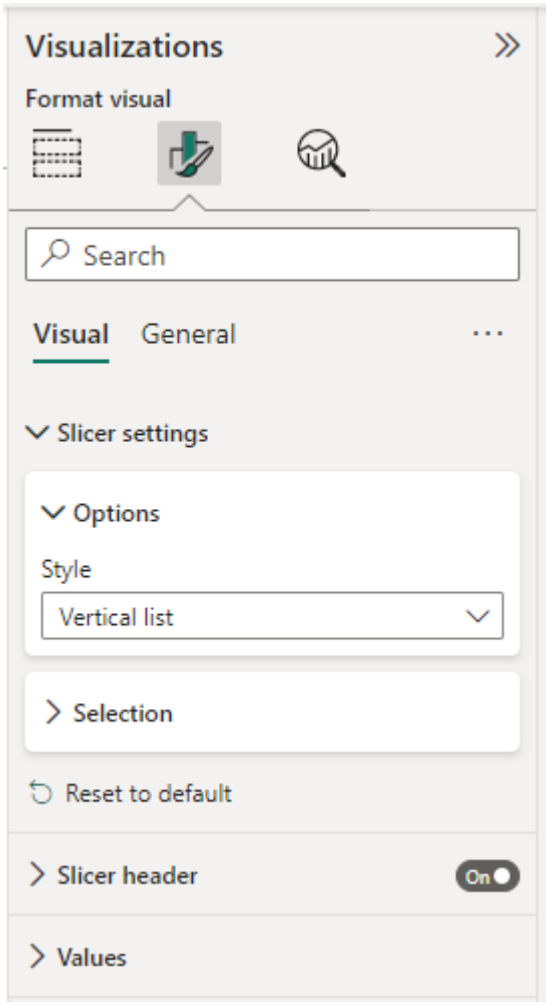
- What will your variable adjust? : For choosing an appropriate filter for your data.
- Name: To edit the table name as required.
- Data type: To choose the appropriate filter for your data.
- Minimum: To set a minimum value.
- Maximum: To set a maximum value.
- Increment: A series of regular consecutive additions.
- To display the slicer on canvas, check the "**Add a slicer to this page**" box.



After creating the parameter, we will **get the new table 'Parameter'** on "Fields".



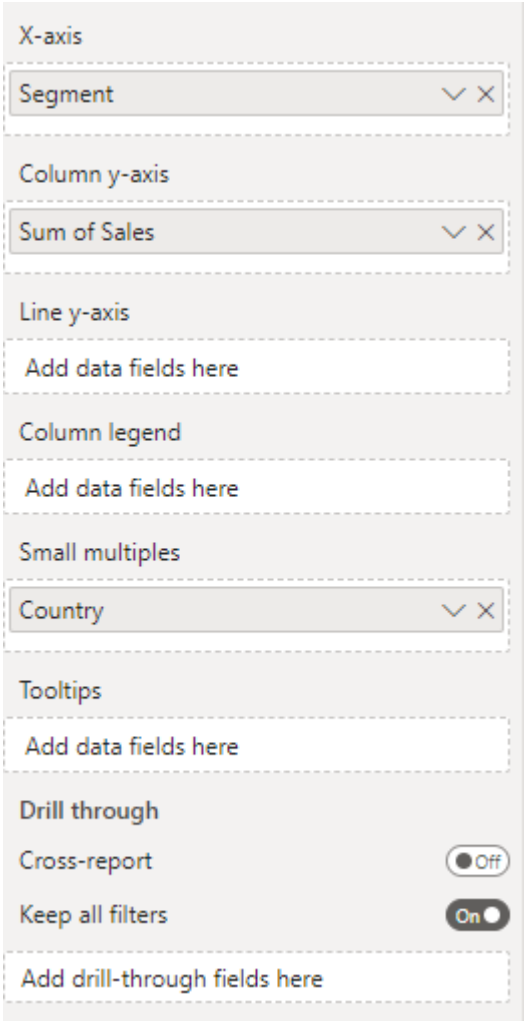
We can do this on the tab "Visual" of the menu "Format visual" on "Visualizations" and then change the field at "Slicer settings" on "Options" to **change the pattern slicer from "Single value" to "Vertical list."**



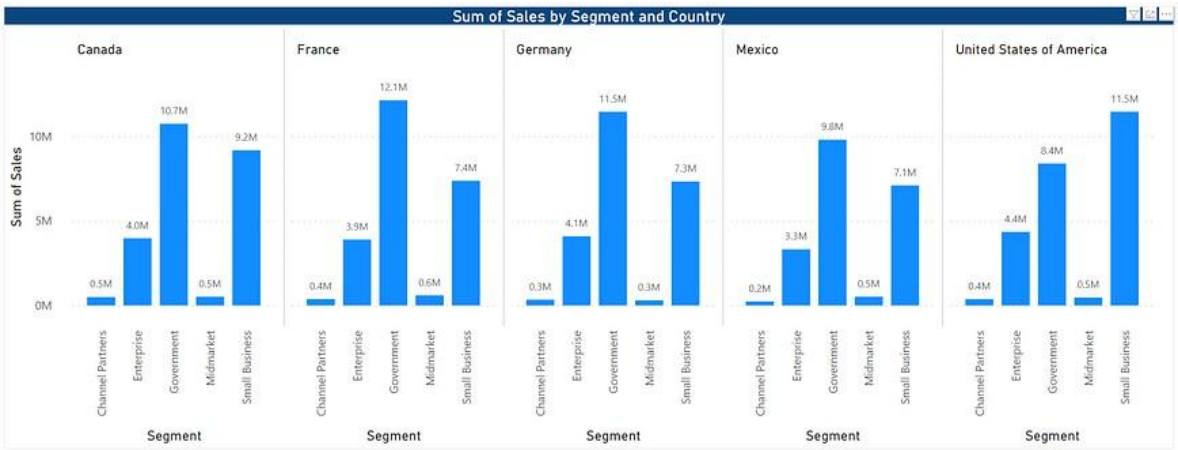
We will also need to create a measure to generate an interactive value and graph. Then go to the next step.

Step 2: Create a “measure”.

We create a visualization “Line and clustered column chart”. And fill value for each field as below.



We will get the chart as shown below.



Next, start to create a DAX measure

- **Parameter-SalesValue:** Create a measure on the “Parameter” table.

```
Parameter-SalesValue = SELECTEDVALUE('Parameter'[Parameter])
```

- **Sales Discount:** We create a new measure called “Sales Discount” that is calculated by multiplying the sum of Sales (on the “financials” table) by Parameter-SalesValue (on the “Parameter” table). Then, to convert the unit to a percentage, divide the result by 100.

```
Sales_Discount = ((sum(financials[ Sales])*Parameter[Parameter-SalesValue])/100)
```

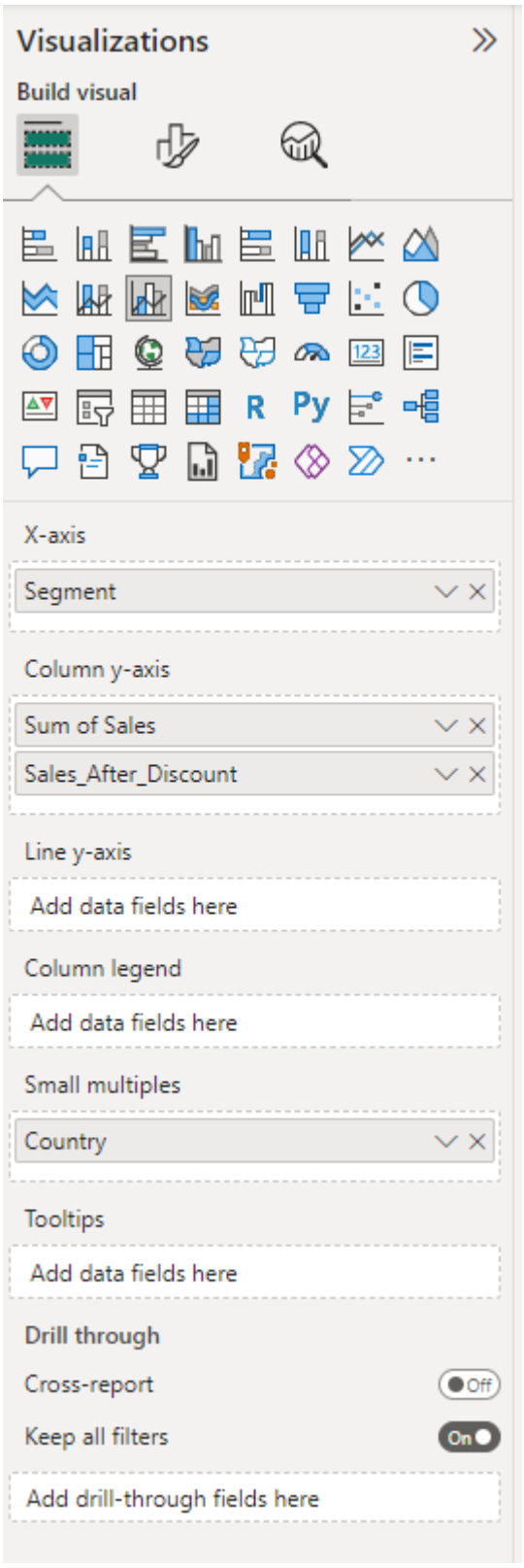
- **Sales After Discount:** This is a measure for calculating the formula that will show the result of the final price after getting the discount (refer to % discount on the parameter).

```
Sales_After_Discount = sum(financials[ Sales]) - Parameter[Sales_Discount]
```

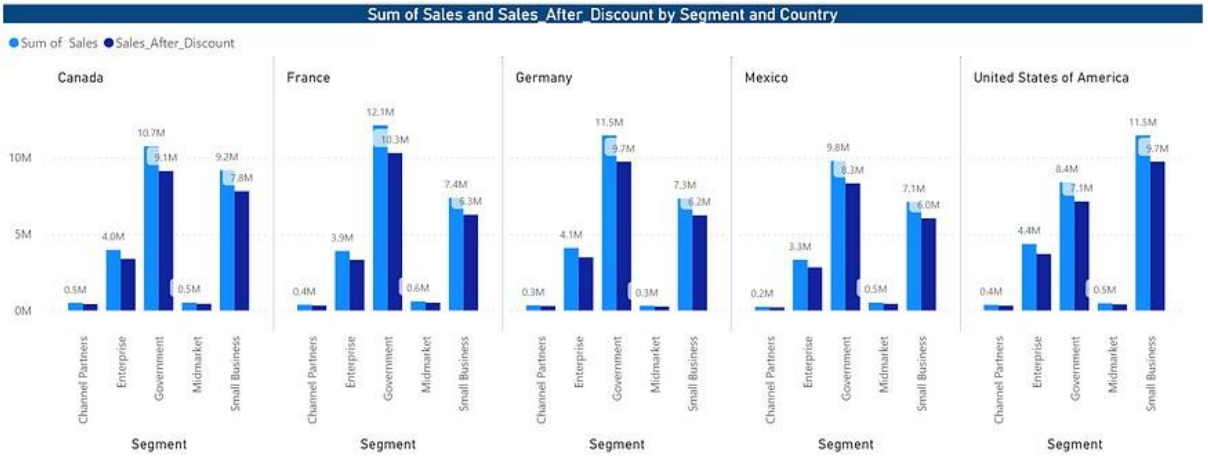
Step 3: Visual for what-if analysis.

It’s time to visual our data !!

- Fill the column in each field as below. Don’t forget to choose “Line and clustered column char”.


Visualizations
Build visual
X-axis
Segment
Column y-axis
Sum of Sales
Sales_After_Discount
Line y-axis
Add data fields here
Column legend
Add data fields here
Small multiples
Country
Tooltips
Add data fields here
Drill through
Cross-report
Off
Keep all filters
On
Add drill-through fields here

- Then we will get the bar chart is shown below (with a 15% discount).



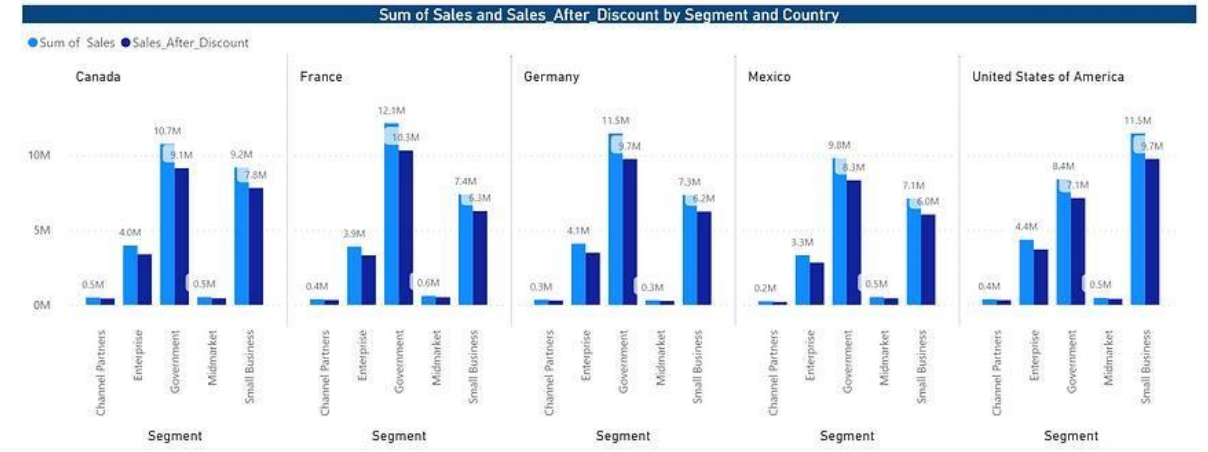
- Finally, we get the report, which includes information about each country, segment, and discount percentage.

-

% discount
0
5
10
15
20

Sales after get 15 % Discount

Country	Canada		France		Germany		Mexico		United States of America	
Segment	Sum of Sales	Sales_After_Discount	Sum of Sales	Sales_After_Discount	Sum of Sales	Sales_After_Discount	Sum of Sales	Sales_After_Discount	Sum of Sales	Sales_After_Discount
Channel Partners	491,164.14	417,489.52	372,090.36	316,276.81	336,425.88	285,962.00	234,379.08	199,222.22	366,534.18	311,554.05
Enterprise	3,967,491.25	3,372,367.56	3,890,890.63	3,307,257.03	4,086,826.25	3,473,802.31	3,315,881.25	2,818,499.06	4,350,605.00	3,698,014.25
Government	10,741,236.52	9,130,051.04	12,127,782.72	10,308,615.31	11,452,895.94	9,734,961.55	9,791,599.38	8,322,859.47	8,390,746.11	7,132,134.19
Midmarket	510,213.98	433,681.88	593,802.08	504,731.76	301,344.75	256,143.04	511,136.40	434,465.94	465,385.88	395,577.99
Small Business	9,177,549.00	7,800,916.65	7,369,606.50	6,264,165.53	7,327,848.00	6,228,670.80	7,096,356.00	6,031,902.60	11,456,559.00	9,738,075.15



RESULT:

Thus the What – If analysis for Data Visualization is applied to design and generate necessary on the data warehouse data in Excel successfully.

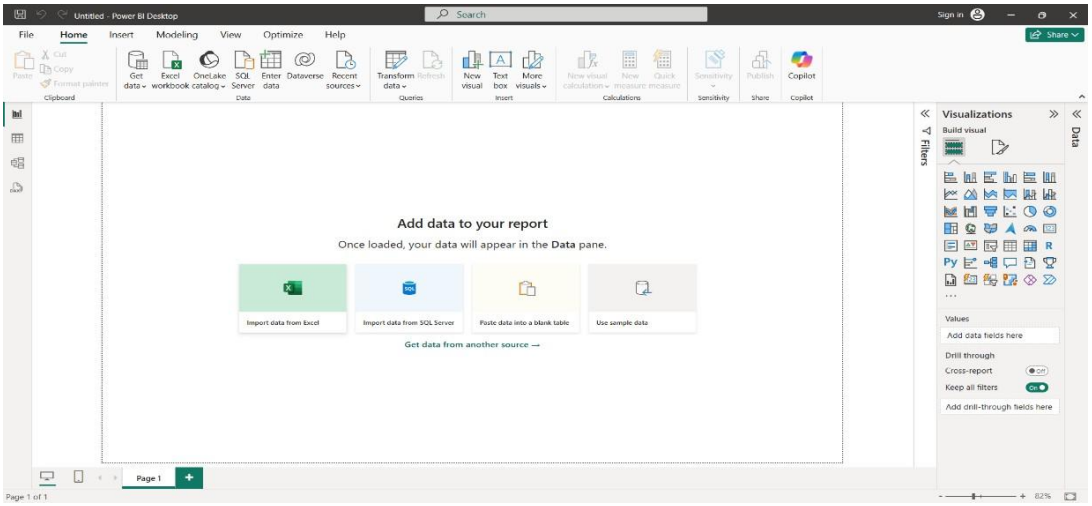
EX.NO: 6	IMPLEMENTATION OF CLASSIFICATION ALGORITHM IN R PROGRAMMING
DATE:	

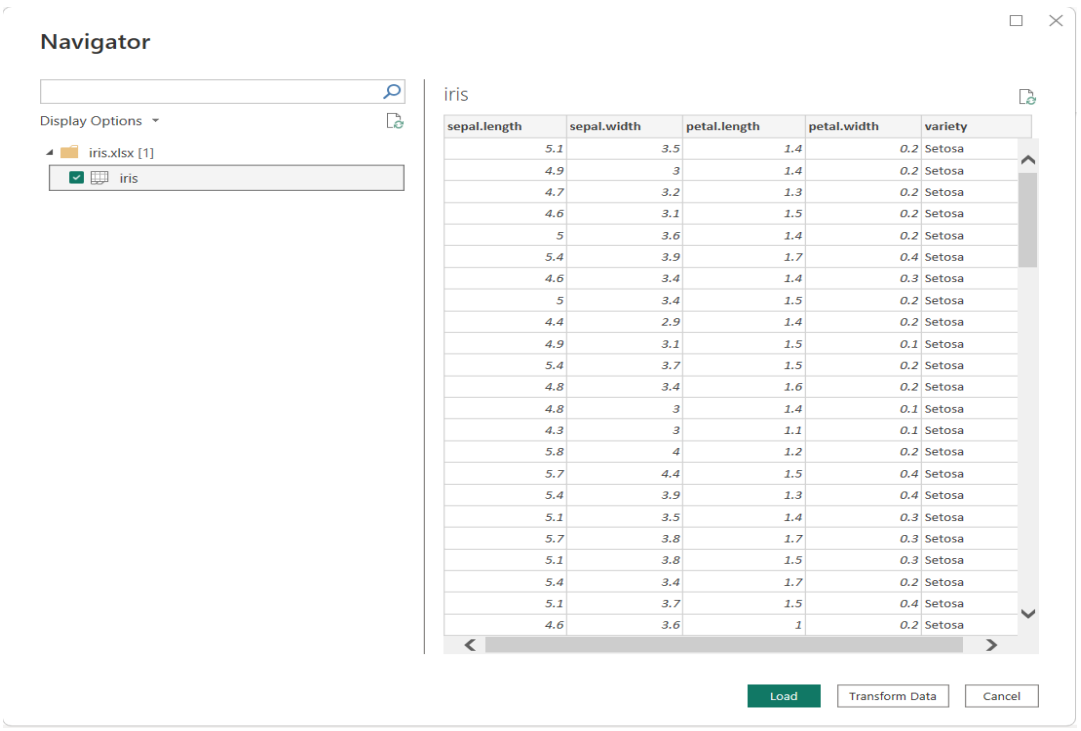
AIM:

To implement the classification algorithm using R programming in Power BI.

PROCEDURE:

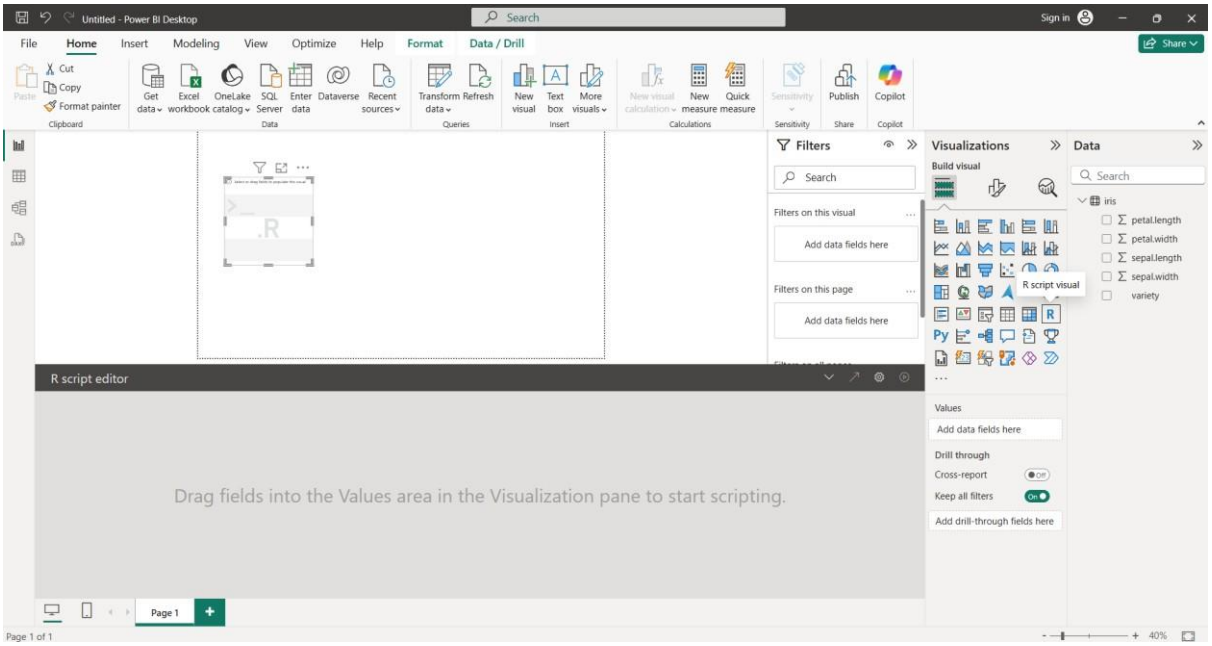
Step 1: Import The Dataset into Power BI



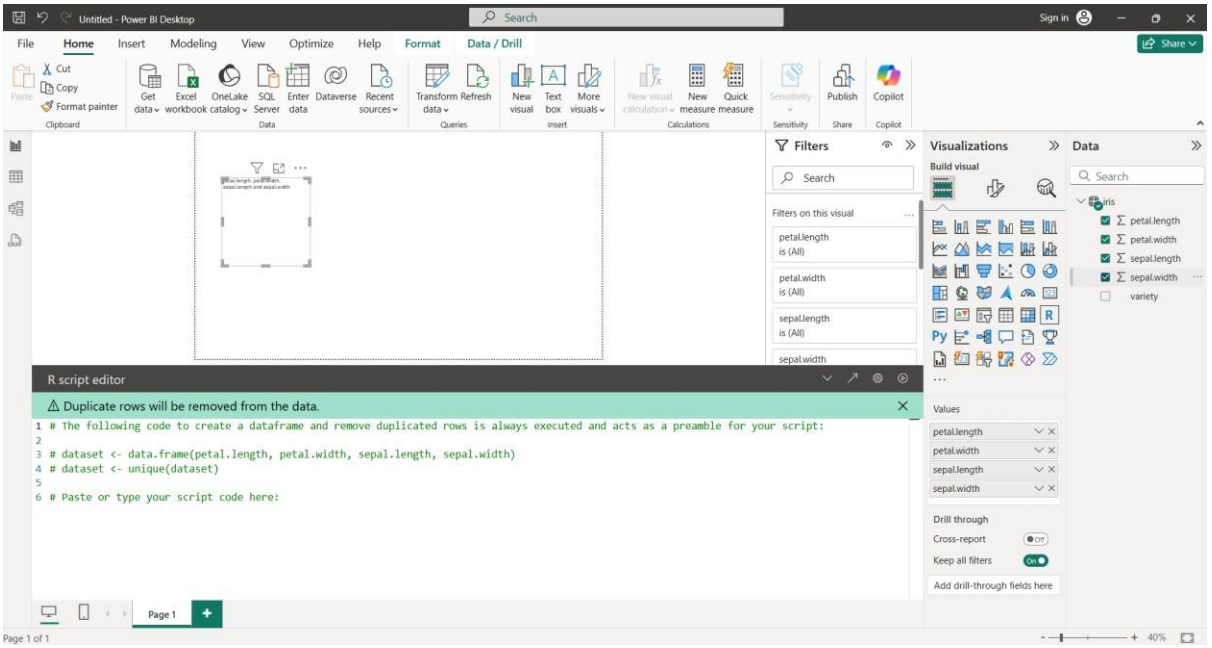


sepal.length	sepal.width	petal.length	petal.width	variety
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.2	Setosa
5.4	3.9	1.7	0.4	Setosa
4.6	3.4	1.4	0.3	Setosa
5	3.4	1.5	0.2	Setosa
4.4	2.9	1.4	0.2	Setosa
4.9	3.1	1.5	0.1	Setosa
5.4	3.7	1.5	0.2	Setosa
4.8	3.4	1.6	0.2	Setosa
4.8	3	1.4	0.1	Setosa
4.3	3	1.1	0.1	Setosa
5.8	4	1.2	0.2	Setosa
5.7	4.4	1.5	0.4	Setosa
5.4	3.9	1.3	0.4	Setosa
5.1	3.5	1.4	0.3	Setosa
5.7	3.8	1.7	0.3	Setosa
5.1	3.8	1.5	0.3	Setosa
5.4	3.4	1.7	0.2	Setosa
5.1	3.7	1.5	0.4	Setosa
4.6	3.6	1	0.2	Setosa

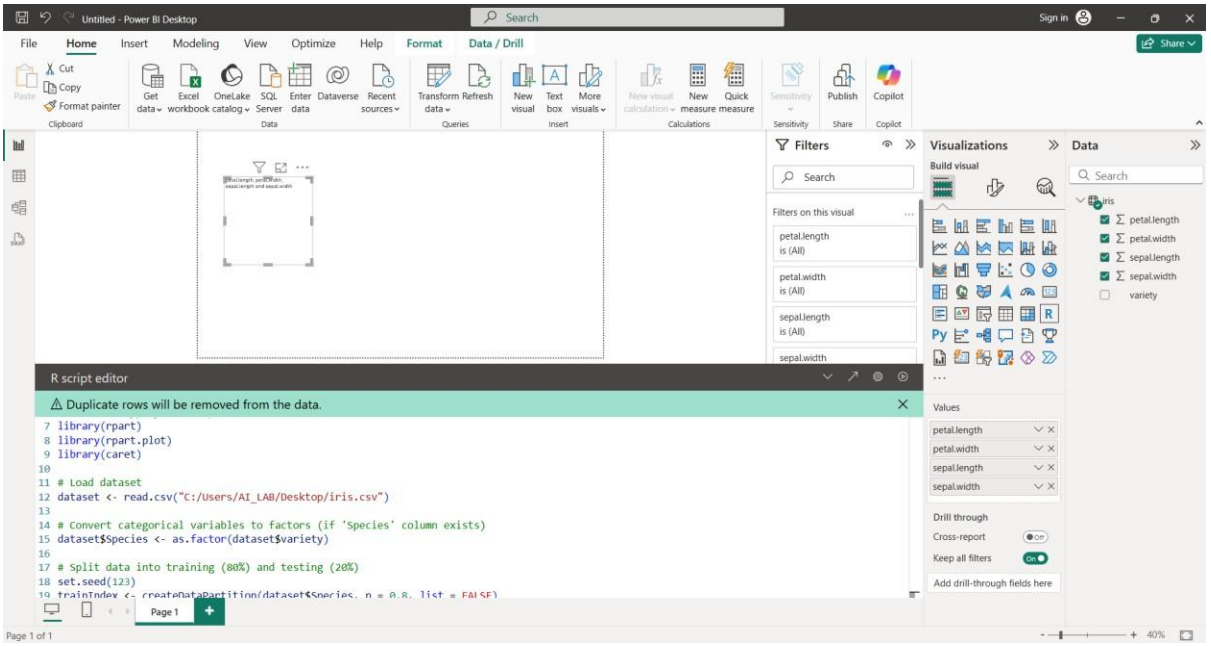
Step 2: Then select R script visual from the Visualization



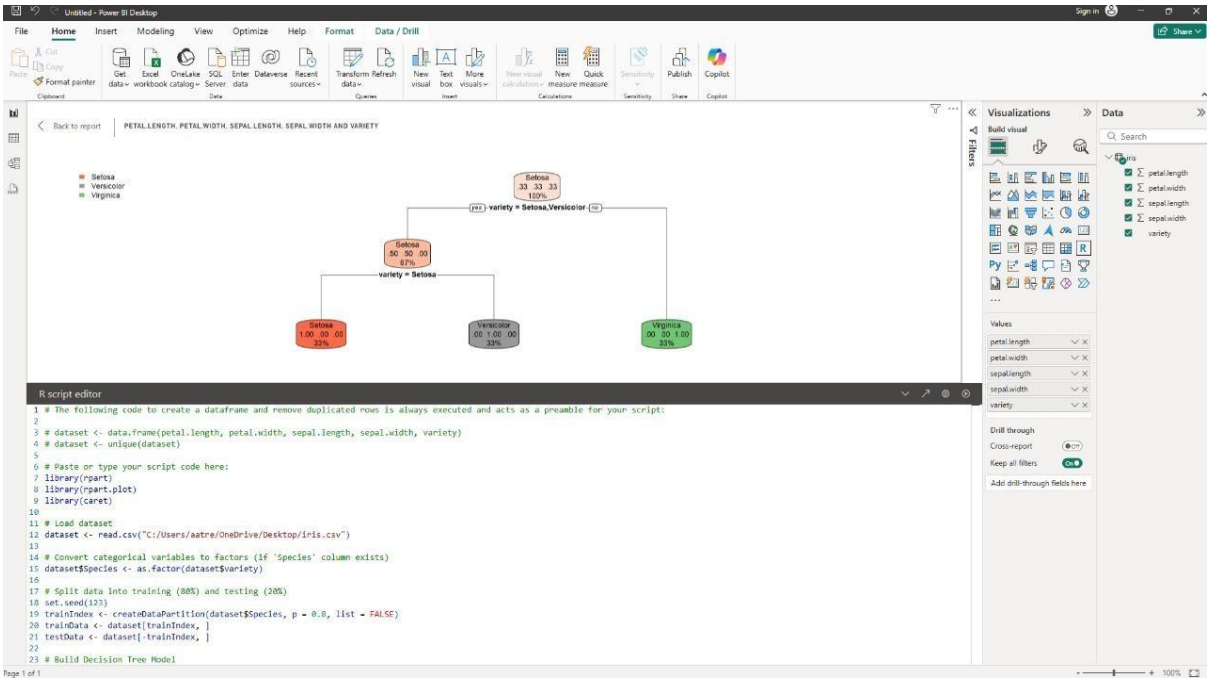
Step 3: Select No of Parameters from the dataset to work within it.



Step 4: In the Script editor Type the Classification algorithm using R for visualization



Step 5: Run the script for the visualization of the code

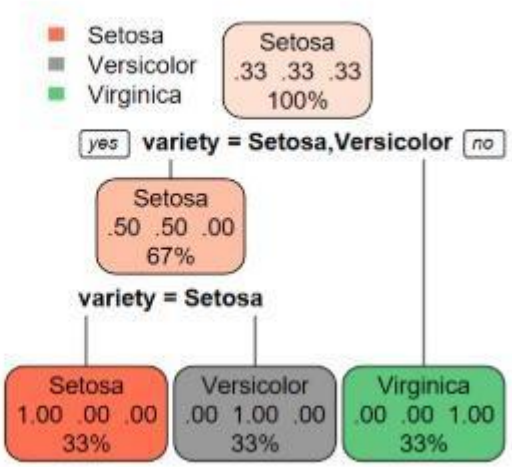


CODE:

```
library(rpart)
library(rpart.plot)
library(caret)
# Load dataset

dataset <- read.csv("C:/Users/AI_LAB/Desktop/iris.csv")
# Convert categorical variables to factors (if 'Species' column exists)
dataset$Species <- as.factor(dataset$variety)
# Split data into training (80%) and testing (20%)
set.seed(123)
trainIndex <- createDataPartition(dataset$Species, p = 0.8, list = FALSE)
trainData <- dataset[trainIndex, ]
testData <- dataset[-trainIndex, ]
# Build Decision Tree Model
dt_model <- rpart(Species ~ ., data = trainData, method = "class")
# Plot the Decision Tree
rpart.plot(dt_model)
# Predict on test data
predictions <- predict(dt_model, testData, type = "class")
# Evaluate model performance
confMatrix <- confusionMatrix(predictions, testData$Species)
print(confMatrix)
```

OUTPUT:

petal.length



RESULT:

Thus, the implementation of classification algorithm using Decision Tree using R programming in Power BI was executed successfully and the output also verified.

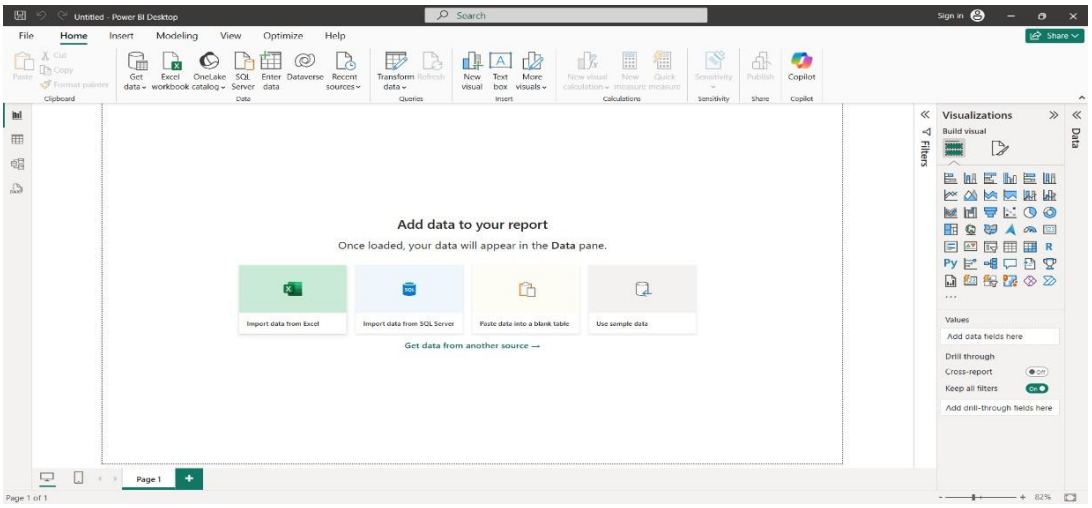
EX.NO: 7	IMPLEMENTATION OF DECISION TREE USING R TOOL
DATE:	

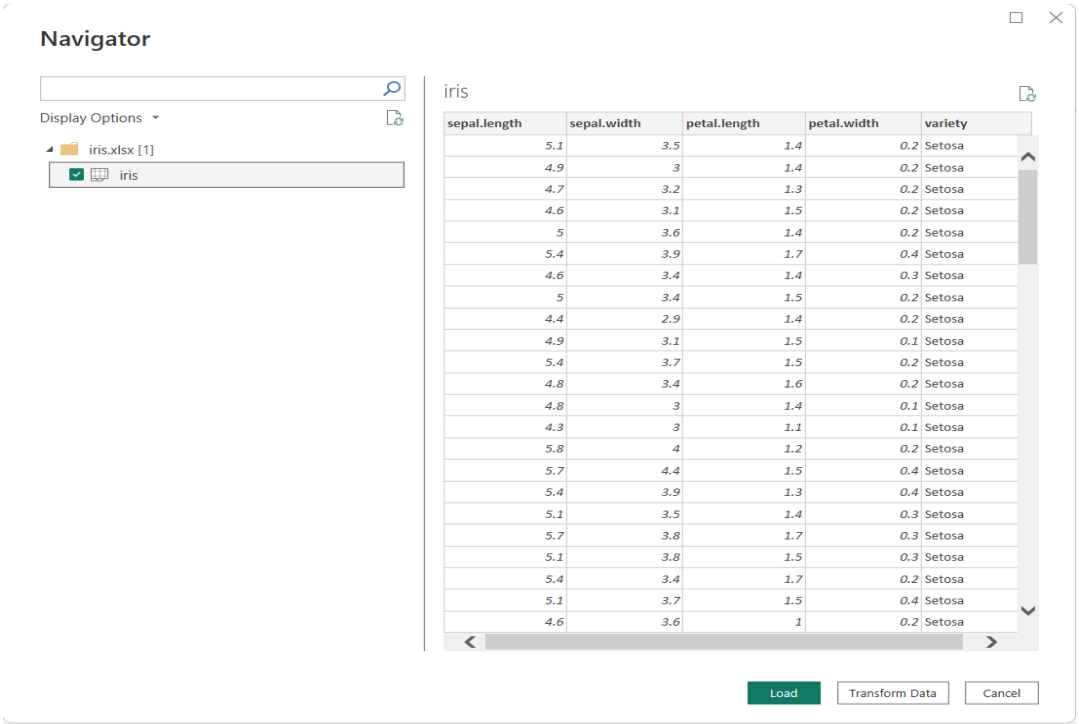
AIM:

To implement the Decision Tree using R programming in Power BI.

PROCEDURE:

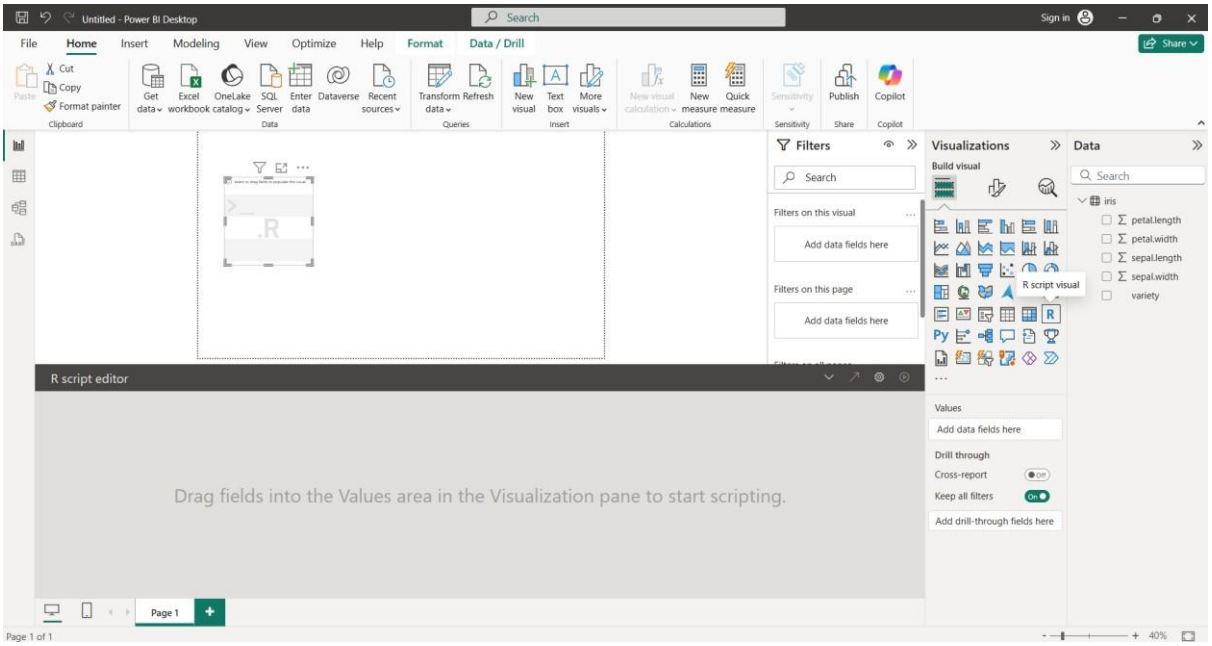
Step 1: Import The Dataset into Power BI



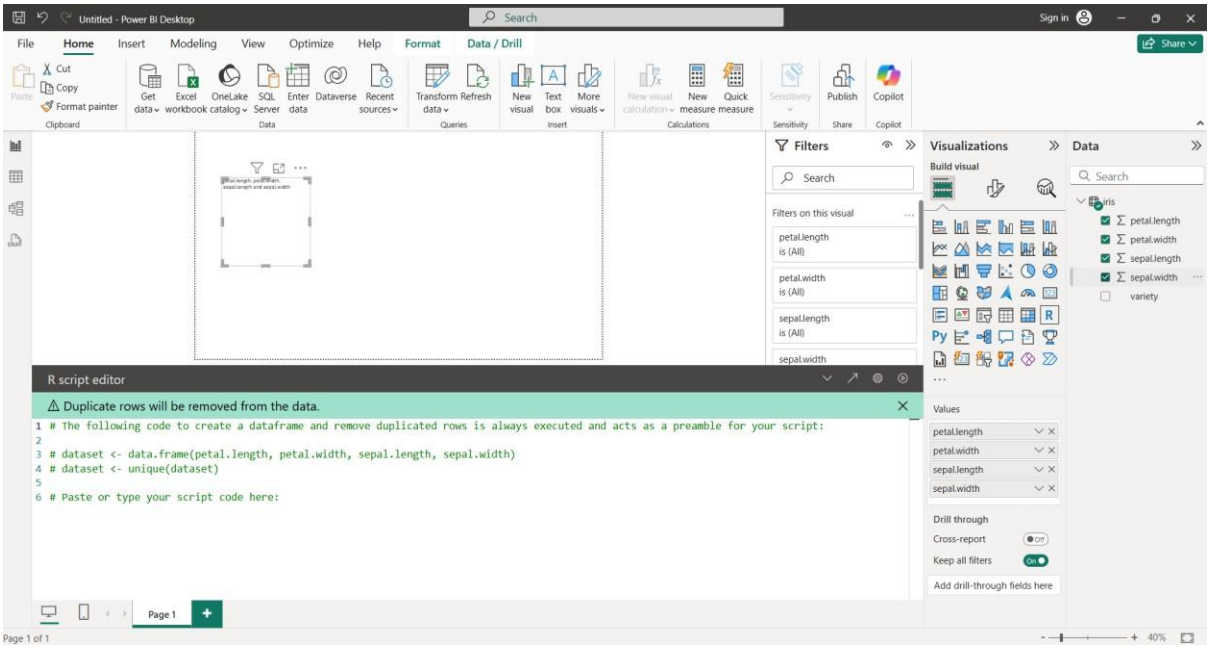


sepal.length	sepal.width	petal.length	petal.width	variety
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.2	Setosa
5.4	3.9	1.7	0.4	Setosa
4.6	3.4	1.4	0.3	Setosa
5	3.4	1.5	0.2	Setosa
4.4	2.9	1.4	0.2	Setosa
4.9	3.1	1.5	0.1	Setosa
5.4	3.7	1.5	0.2	Setosa
4.8	3.4	1.6	0.2	Setosa
4.8	3	1.4	0.1	Setosa
4.3	3	1.1	0.1	Setosa
5.8	4	1.2	0.2	Setosa
5.7	4.4	1.5	0.4	Setosa
5.4	3.9	1.3	0.4	Setosa
5.1	3.5	1.4	0.3	Setosa
5.7	3.8	1.7	0.3	Setosa
5.1	3.8	1.5	0.3	Setosa
5.4	3.4	1.7	0.2	Setosa
5.1	3.7	1.5	0.4	Setosa
4.6	3.6	1	0.2	Setosa

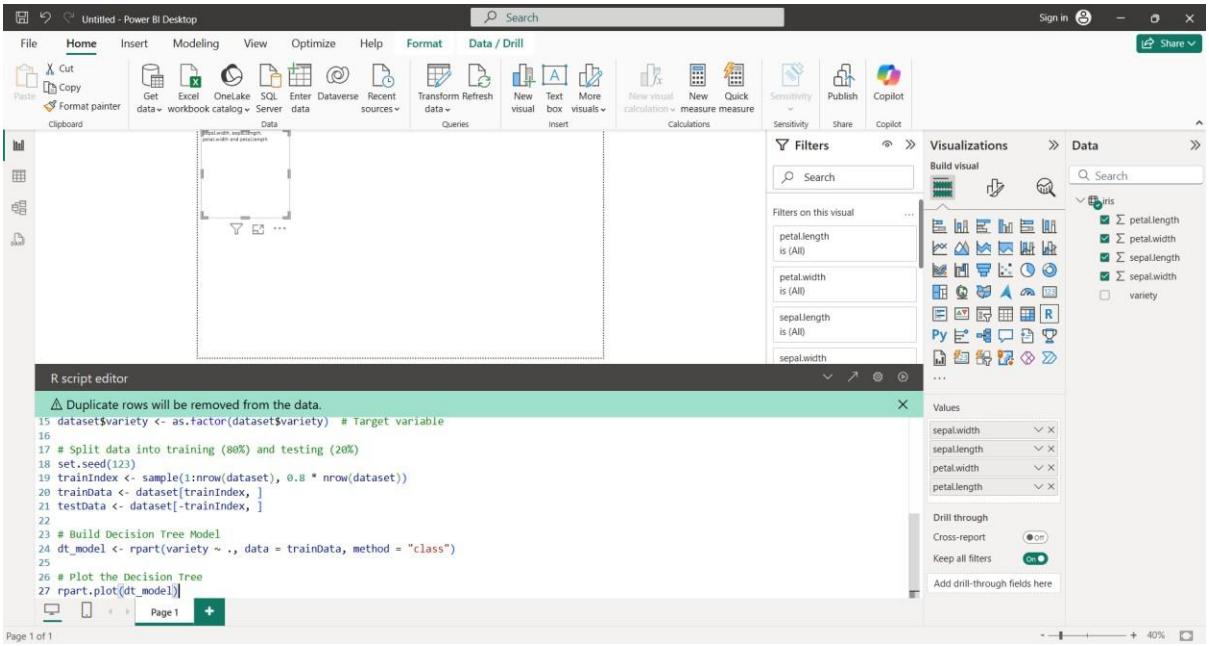
Step 2: Then select R script visual from the Visualization



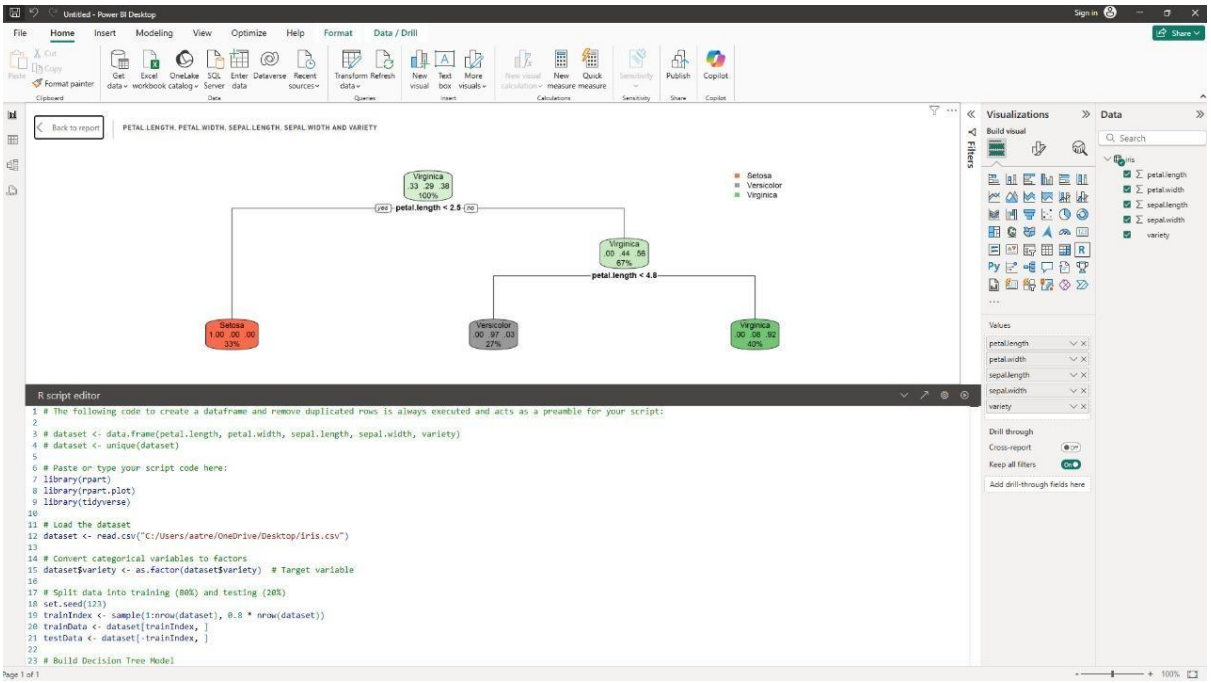
Step 3: Select No of Parameters from the dataset to work within it.



Step 4: In the Script editor Type the Decision Tree algorithm using R for visualization



Step 5: Run the script for the visualization of the code

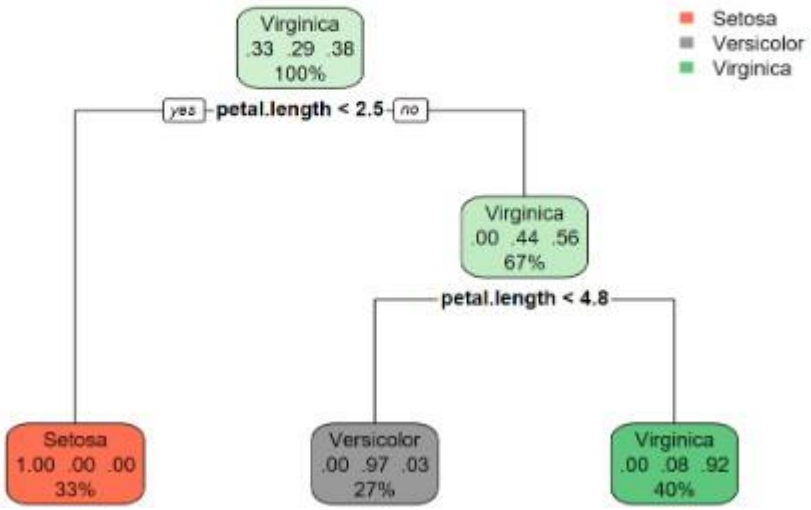


CODE:

```
library(rpart)
library(rpart.plot)
library(tidyverse)
# Load the dataset
dataset <- read.csv("C:/Users/AI_LAB/Desktop/iris.csv")
# Convert categorical variables to factors
dataset$variety <- as.factor(dataset$variety)  # Target variable
# Split data into training (80%) and testing (20%)
set.seed(123)
trainIndex <- sample(1:nrow(dataset), 0.8 * nrow(dataset))
trainData <- dataset[trainIndex, ]
testData <- dataset[-trainIndex, ]
# Build Decision Tree Model
dt_model <- rpart(variety ~ ., data = trainData, method = "class")
# Plot the Decision Tree
rpart.plot(dt_model)
predictions <- predict(dt_model, testData, type = "class")
# Evaluate model performance
confMatrix <- confusionMatrix(predictions, testData$Species)
print(confMatrix)
```

OUTPUT:

Sum of petal.length, Sum of petal.width, sepal.width, sepal.length and variety



RESULT:

Thus, the implementation of Decision Tree using R programming in Power BI was executed successfully and the output also verified.

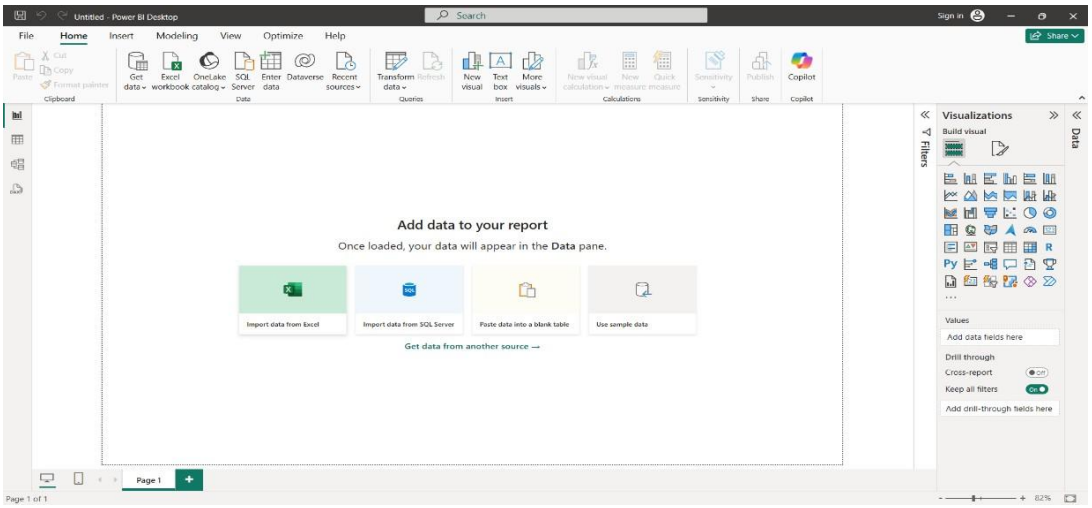
EX.NO: 8	K-MEANS CLUSTERING USING R PROGRAMMING
DATE:	

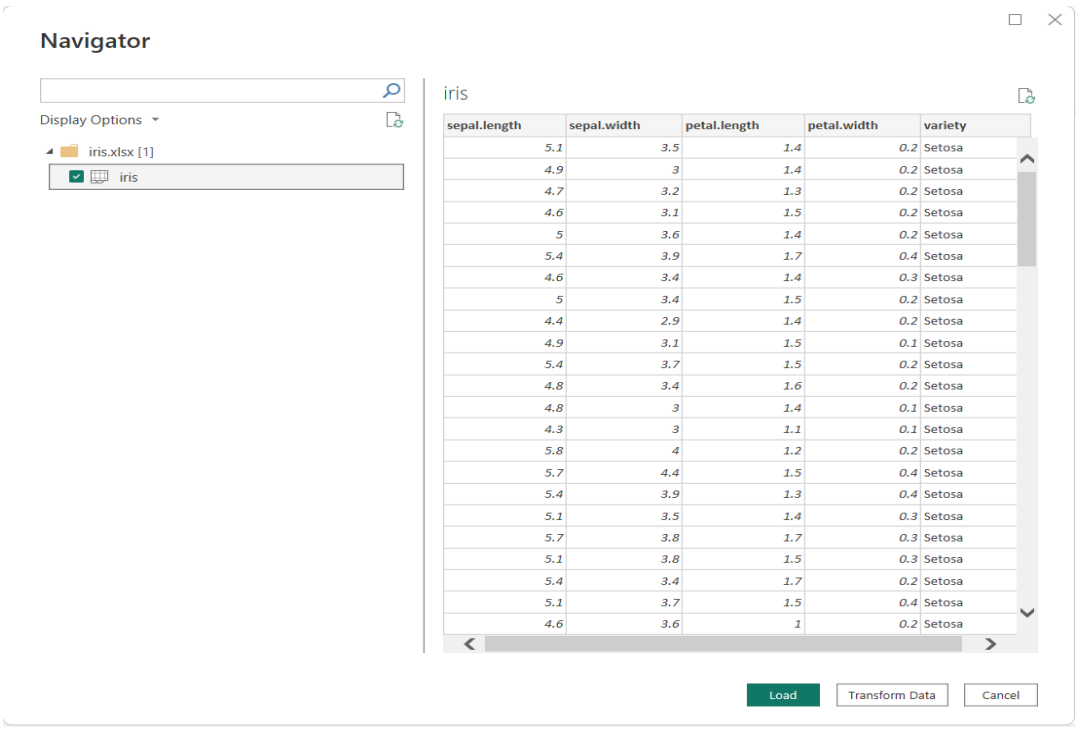
AIM:

To implement the K-Means Clustering using R programming in Power BI.

PROCEDURE:

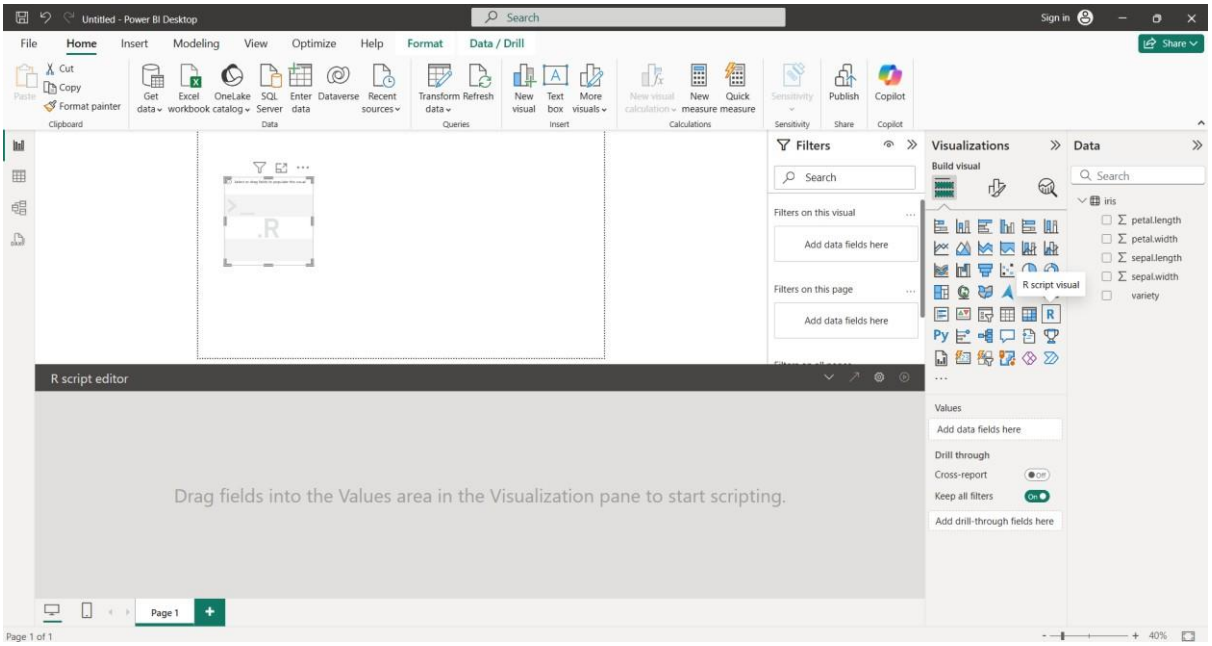
Step 1: Import the Dataset into Power BI



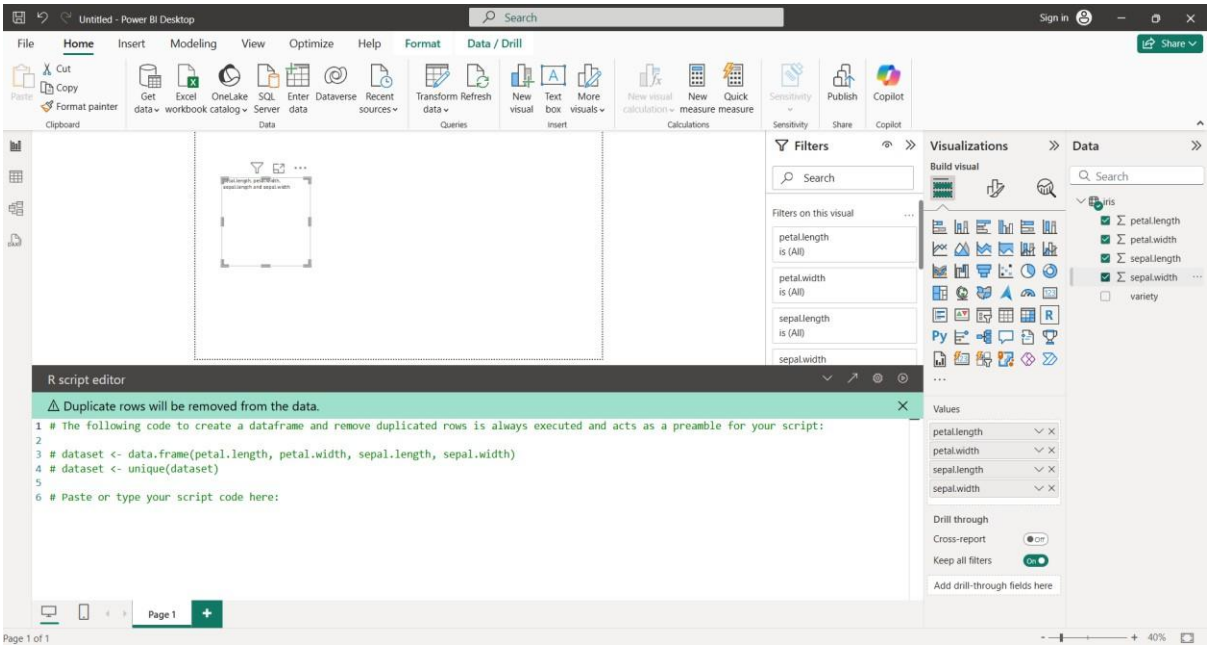


sepal.length	sepal.width	petal.length	petal.width	variety
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.2	Setosa
5.4	3.9	1.7	0.4	Setosa
4.6	3.4	1.4	0.3	Setosa
5	3.4	1.5	0.2	Setosa
4.4	2.9	1.4	0.2	Setosa
4.9	3.1	1.5	0.1	Setosa
5.4	3.7	1.5	0.2	Setosa
4.8	3.4	1.6	0.2	Setosa
4.8	3	1.4	0.1	Setosa
4.3	3	1.1	0.1	Setosa
5.8	4	1.2	0.2	Setosa
5.7	4.4	1.5	0.4	Setosa
5.4	3.9	1.3	0.4	Setosa
5.1	3.5	1.4	0.3	Setosa
5.7	3.8	1.7	0.3	Setosa
5.1	3.8	1.5	0.3	Setosa
5.4	3.4	1.7	0.2	Setosa
5.1	3.7	1.5	0.4	Setosa
4.6	3.6	1	0.2	Setosa

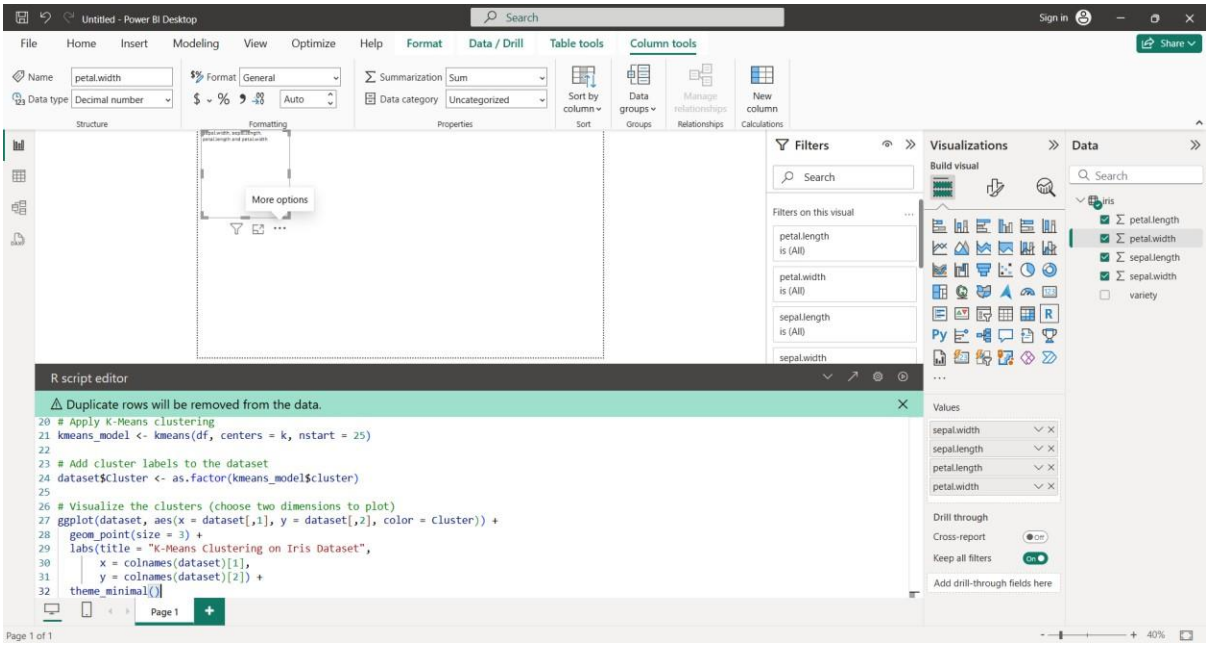
Step 2: Then select R script visual from the Visualization



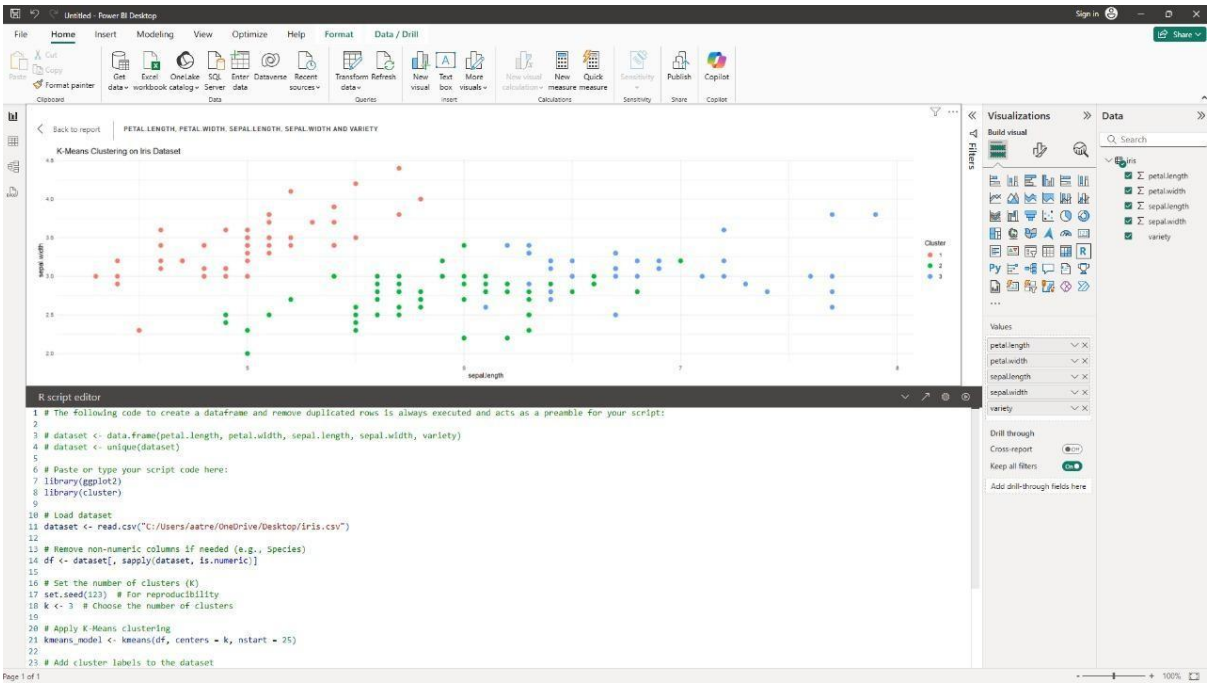
Step 3: Select No of Parameters from the dataset to work within it.



Step 4: In the Script editor Type the K-Means algorithm using R for visualization



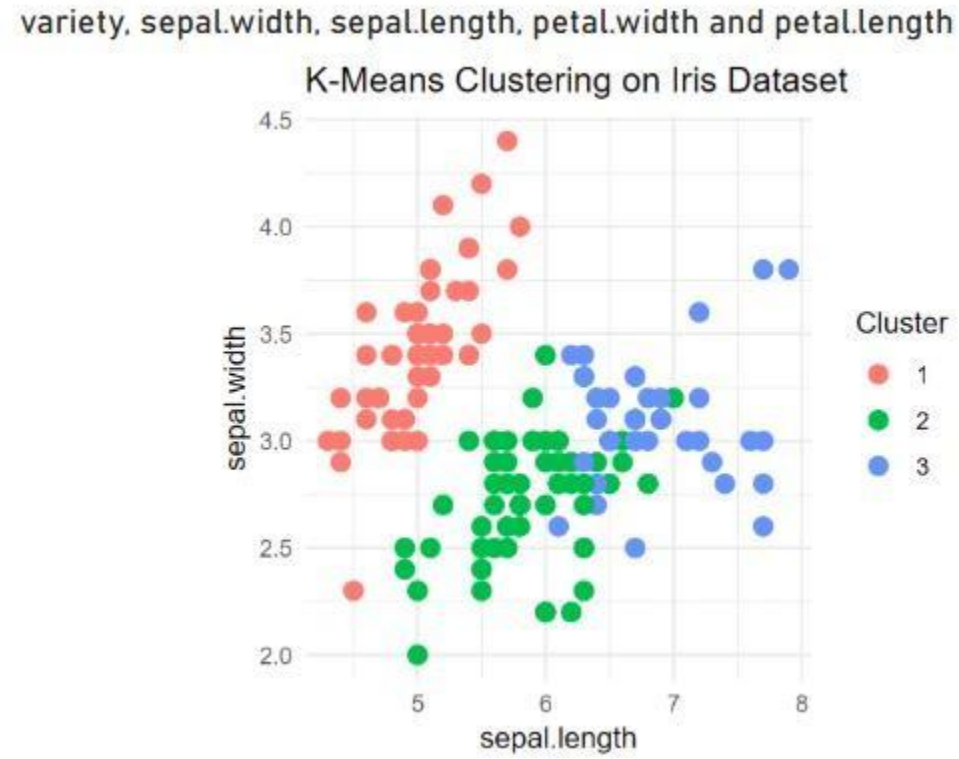
Step 5: Run the script for the visualization of the code



CODE:

```
library(ggplot2)
library(cluster)
# Load dataset
dataset <- read.csv("C:/Users/AI_LAB/Desktop/iris.csv")
# Remove non-numeric columns if needed (e.g., Species)
df <- dataset[, sapply(dataset, is.numeric)]
# Set the number of clusters (K)
set.seed(123)  # For reproducibility
k <- 3  # Choose the number of clusters
# Apply K-Means clustering
kmeans_model <- kmeans(df, centers = k, nstart = 25)
# Add cluster labels to the dataset
dataset$Cluster <- as.factor(kmeans_model$cluster)
# Visualize the clusters (choose two dimensions to plot)
ggplot(dataset, aes(x = dataset[,1], y = dataset[,2], color = Cluster)) +
geom_point(size = 3) +
  labs(title = "K-Means Clustering on Iris Dataset",
       x = colnames(dataset)[1],
       y = colnames(dataset)[2]) +
  theme_minimal()
```

OUTPUT:



RESULT:

Thus, the implementation of K-Means Clustering using R programming in Power BI was executed successfully and the output also verified.

Ex.No: 09	PREDICTION USING LINEAR REGRESSION
Date:	

AIM:

To perform the time series analyze in POWER BI by using python script and visualize the charts.

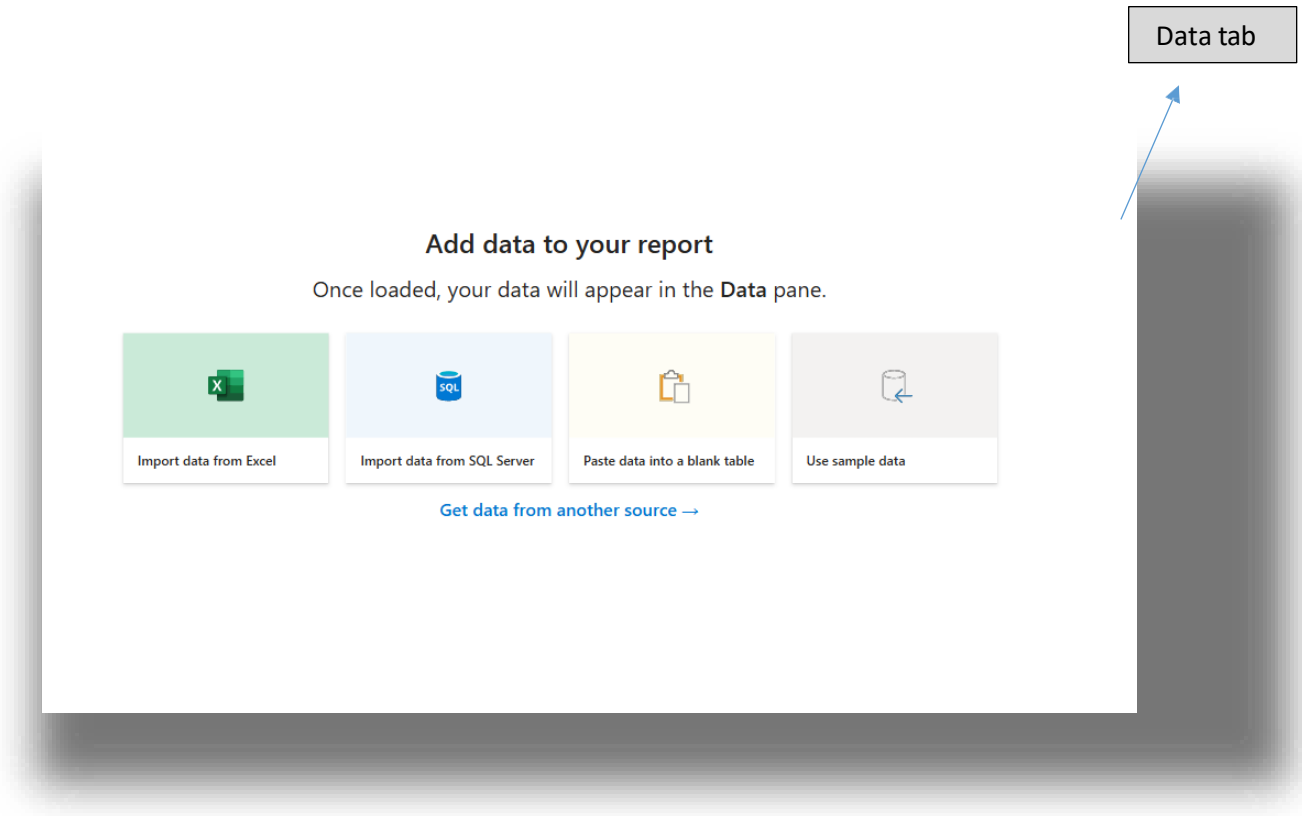
IMPLEMENTATION:

1. Install and Open Power BI

1. **Download & Install Power BI**
   - Visit the official Power BI Download Page.
   - Download and install **Power BI Desktop**.
2. **Open Power BI Application**
   - Launch the Power BI Desktop application after installation.

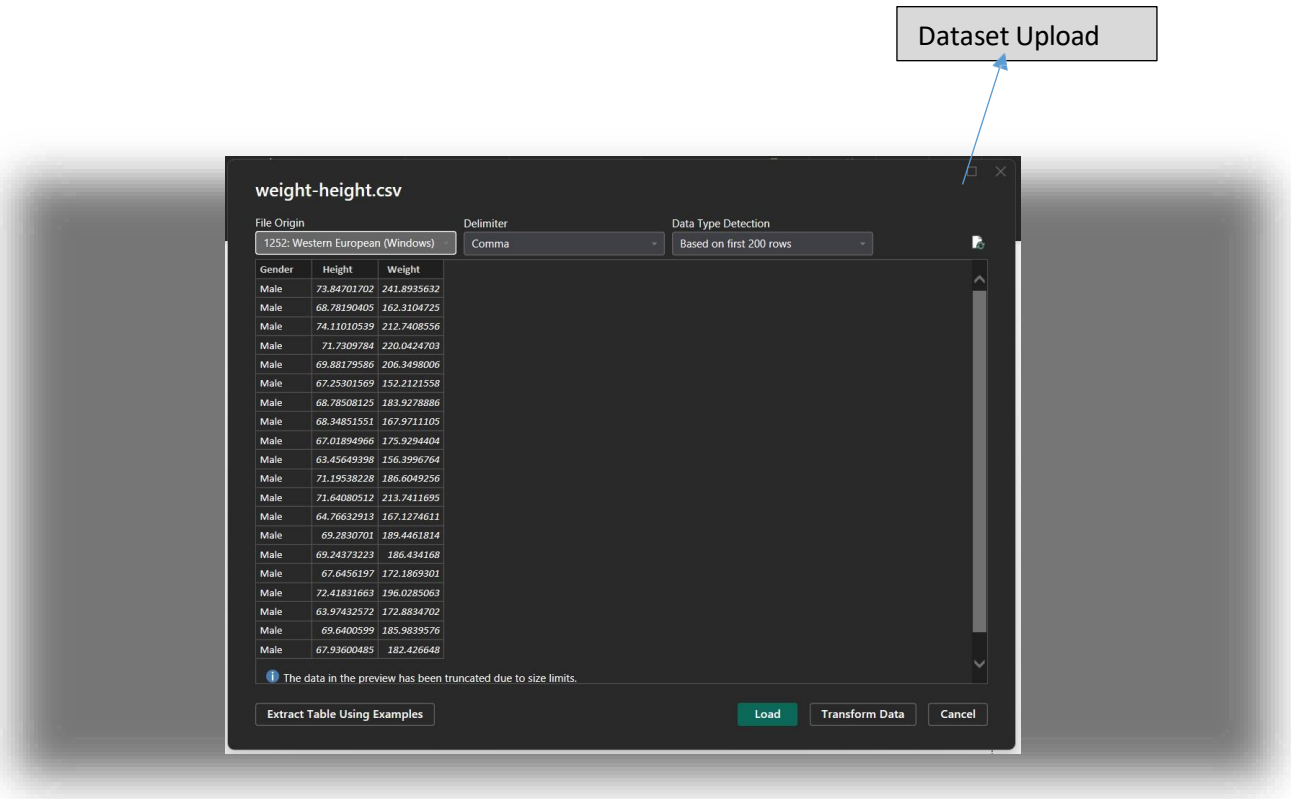
2. Load the Time Series Dataset

3. **Download the Dataset**
   - Ensure you have a **time series dataset** in a suitable format (CSV, TXT, etc.).
   - Example dataset: **"time_series_sales_dataset.csv"**.
4. **Import the Dataset into Power BI**
   - Open **Power BI Desktop**.
   - Click **Home > Get Data > Text/CSV**.
   - Select **"time_series_sales_dataset.csv"** and click **Load**.
   - Ensure that the **Date column** is formatted as **Date/Time**.
5. **Verify the Imported Data**
   - Click the **Data View** option (right panel) to preview the dataset.



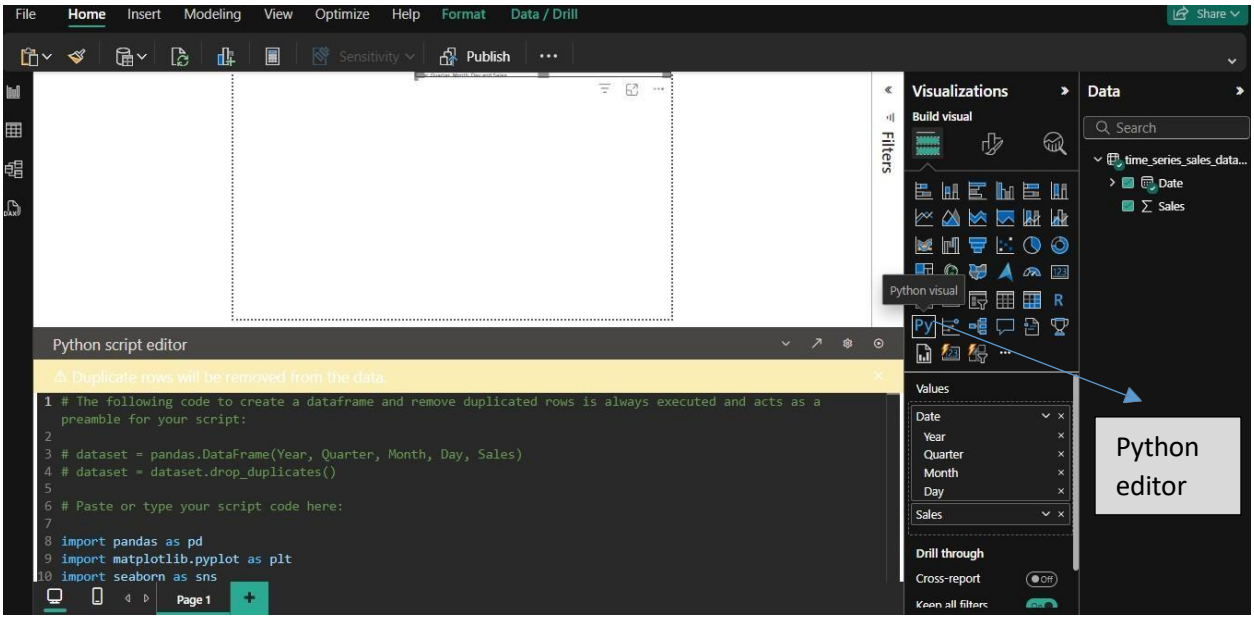
3. Visualize Time Series Data in Power BI

6. **Open the Visualization Pane**
   - Click the **Report View** tab.
   - Click the **"Visualization"** pane to see all available visualizations.
7. **Create a Time Series Line Chart (optional)**
   - From **Visualizations**, select the **Line Chart**.
   - **Drag Date** to the **X-axis**.
   - **Drag Sales** to the **Y-axis**.
   - Title the chart as **"Time Series Sales Analysis"**.



4. Run Python Script for Advanced Analysis

8. **Enable Python Scripting in Power BI**
   - Click **File > Options and Settings > Options**.
   - Under **Python scripting**, set the Python home directory (Anaconda/Miniconda).
9. **Add Python Script in Power BI**
   - Click **"Python Visual" (Py icon)** in the **Visualization Pane**.
   - Select the required **columns** from the dataset.

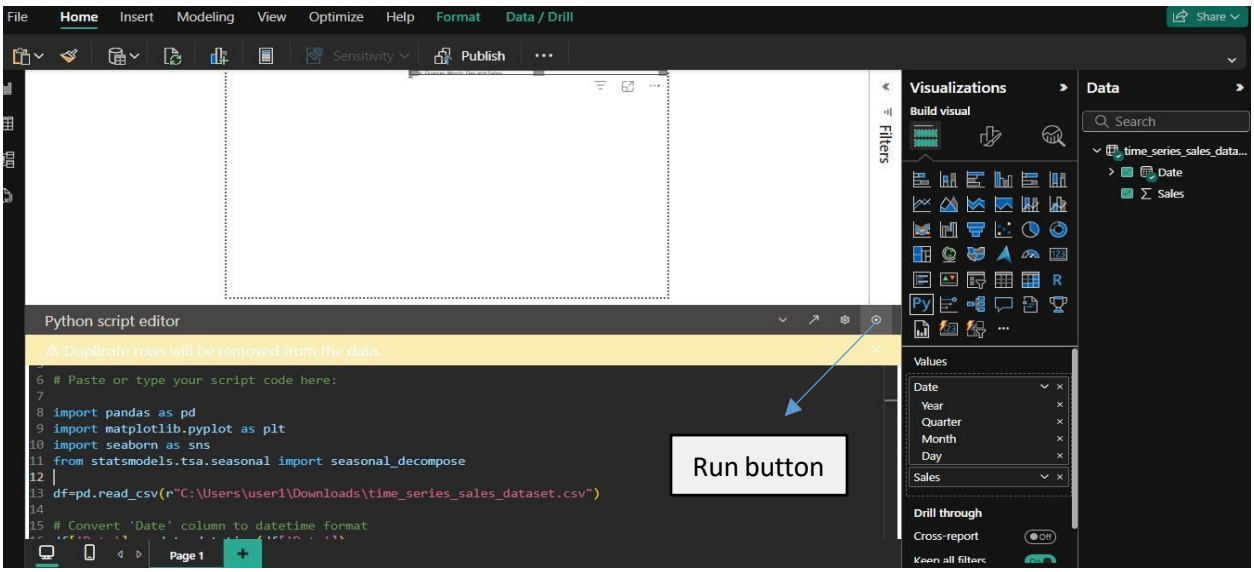


10. **Paste and Run the Python Script**

- Open the **Python Script Editor** in Power BI.
- Paste the following **Python script**:

11. **Run the Python Script**

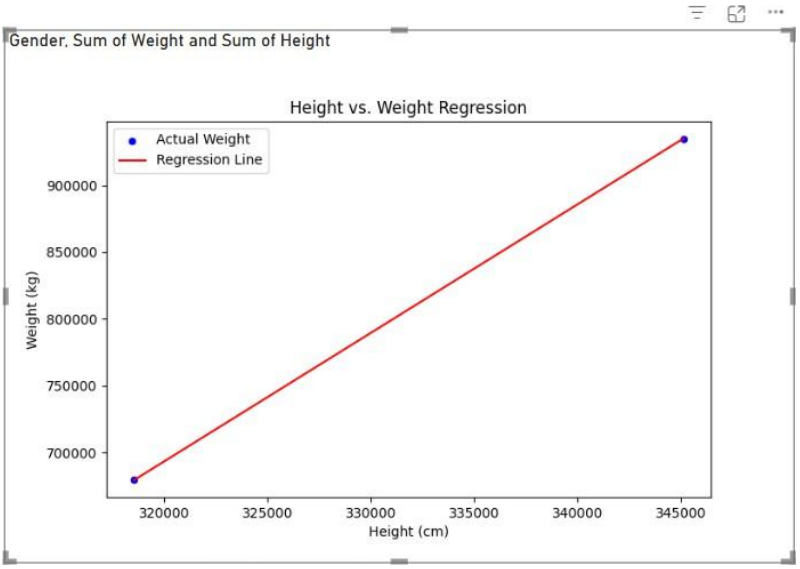
- Click the **Run button** on the top of the Python script editor.
- Wait for execution to complete.



CODE:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

# Load data from Power BI
dataset = dataset  # Power BI automatically passes the dataset

# Define independent (X) and dependent (Y) variables
X = dataset[['Height']]  # Predictor
y = dataset['Weight']  # Response Variable

# Create and train the Linear Regression model
model = LinearRegression()
model.fit(X, y)

# Predict weight using height
dataset['Predicted_Weight'] = model.predict(X)

# Plot the Regression Line
plt.figure(figsize=(8,5))
sns.scatterplot(x=X['Height'], y=y, color='blue', label='Actual Weight')
sns.lineplot(x=X['Height'], y=dataset['Predicted_Weight'], color='red', label='Regression
Line')
plt.xlabel("Height (cm)")
plt.ylabel("Weight (kg)")
plt.title("Height vs. Weight Regression")
plt.legend()
plt.show()
```

OUTPUT:



RESULT:

The program was executed successfully.

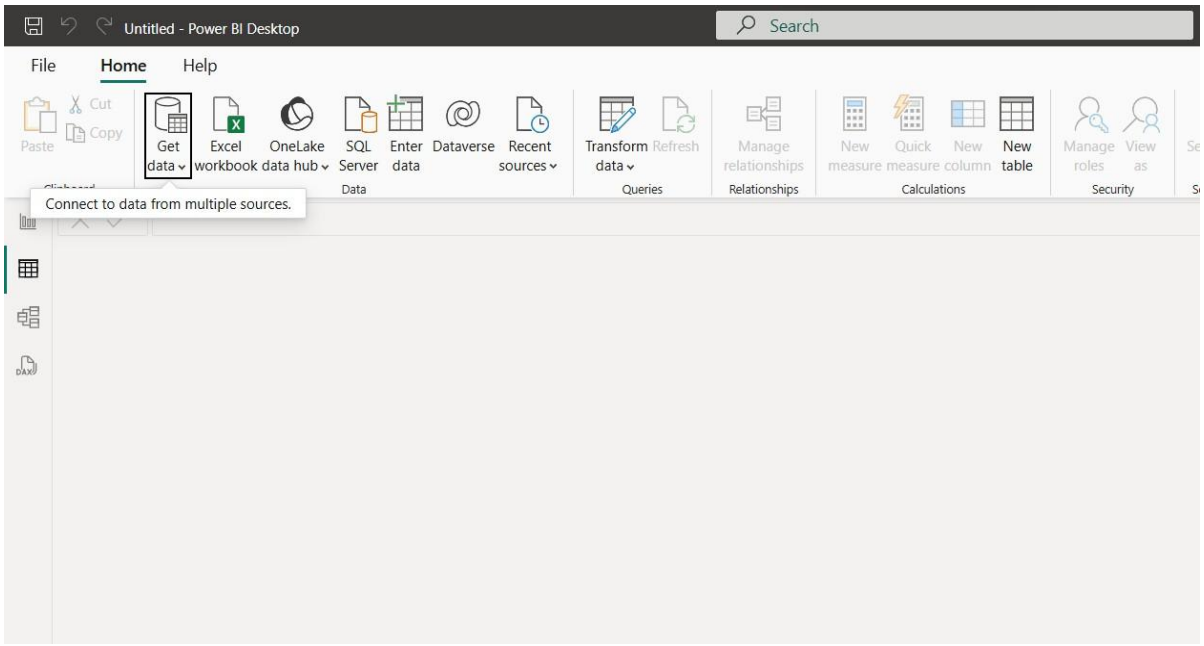
EX.NO: 10 (a)	DATA ANALYSIS USING TIME SERIES ANALYSIS
DATE:	

AIM:

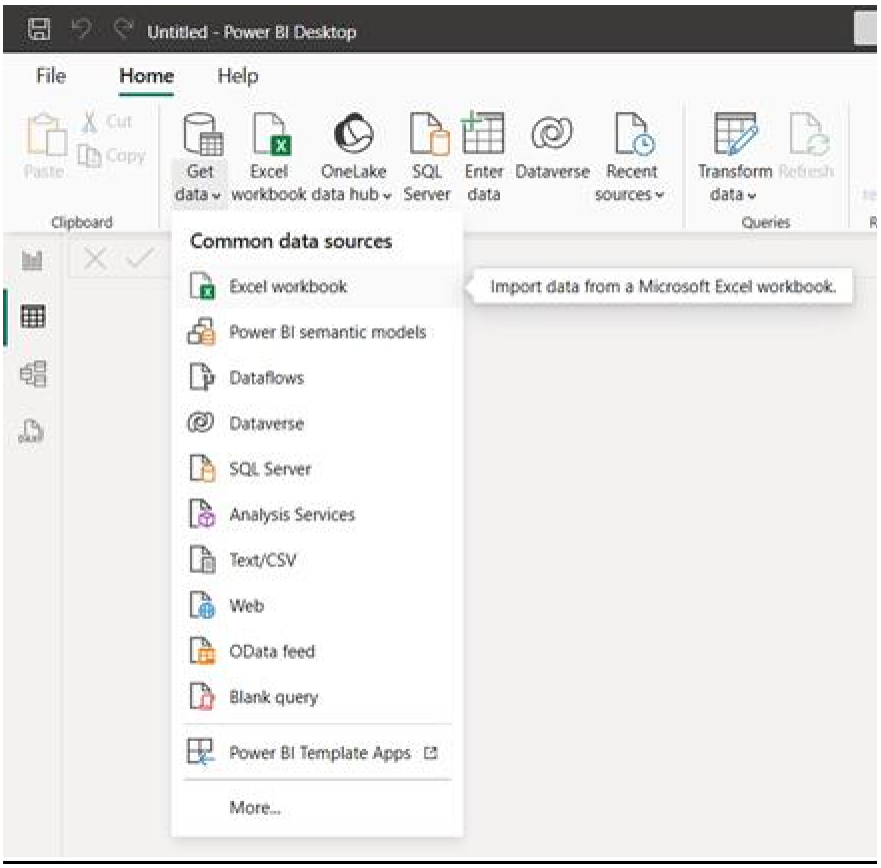
To Create a Dashboard Analytics on Real-Time Sales Using Time Series Dataset

PROCEDURE:

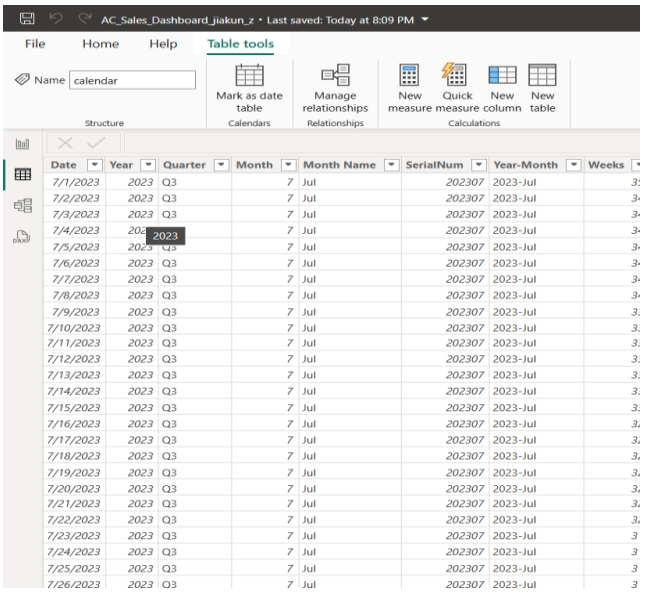
Step1: To begin creating a dashboard for real-time sales analytics using the chocolate sales dataset in Power BI, start by clicking on "Get data" in the ribbon at the top of the screen. This will allow you to import your dataset into Power BI for further analysis and visualization.



Step 2: After clicking "Get data" in Power BI Desktop, navigate to the dropdown menu that appears. From the list of common data sources, select "Excel workbook" to proceed with importing your chocolate sales dataset. This option is highlighted in the menu, allowing you to connect to your Excel file containing the dataset. Power BI will then guide you to locate and load your file for further analysis.



Step 3:

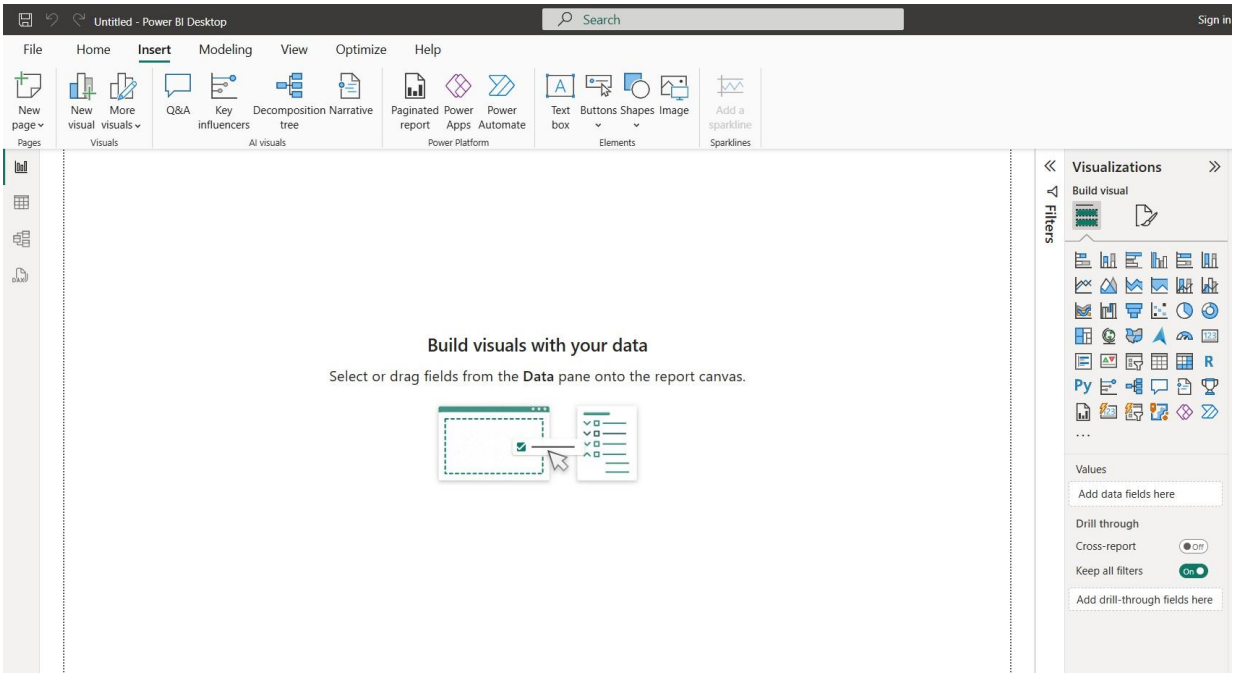


The image shows the Power BI Desktop interface with a table view of a dataset, likely part of your chocolate sales dataset, displayed in the "Fields" pane on the right. The table includes columns such as "Date," "Year," "Quarter," "Month," "Month Name," "SerialNum," and "Year-Month," with data entries for various dates in 2023. This indicates that you've already imported your dataset (e.g., the Excel workbook with chocolate sales data) and are now viewing or preparing it for analysis. The ribbon at the top includes options like "Home," "Insert," "View," and "Modeling," which you can use to transform data, create relationships, or build visualizations. The "Table tools" tab is active, suggesting you're working with a specific table, and you can use features like "Mark as date table," manage relationships, or create new measures and columns.

Next Step:

To proceed with creating a dashboard for real-time sales analytics using the time series dataset, click on "Insert" in the ribbon at the top of the screen, then select "New visual" to start adding visualizations (e.g., line charts, bar charts) to analyze sales trends over time, such as chocolate sales by date or month.

Step 4:

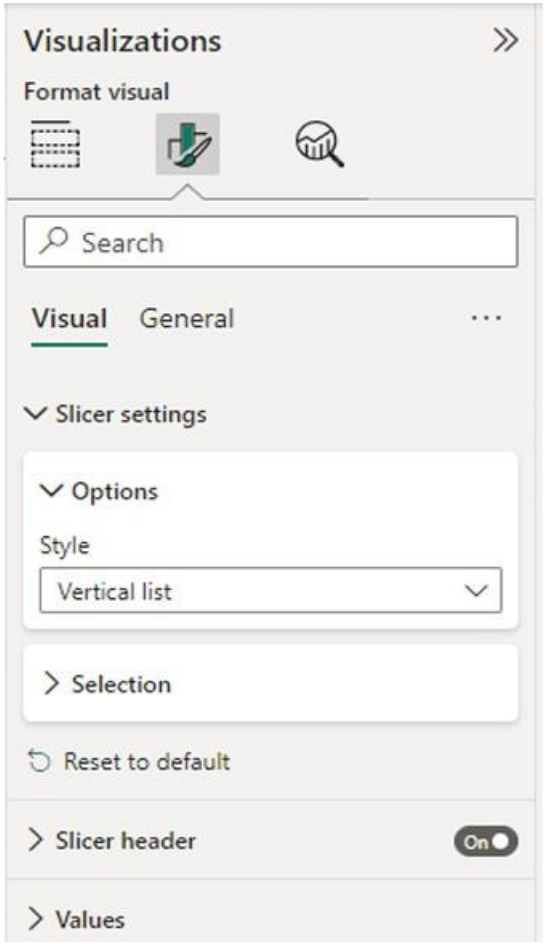


The next step was to click on "Insert" in the ribbon, then select "New visual" to start adding visualizations for your chocolate sales dataset. The current image shows Power BI Desktop with a blank report canvas and the "Visualizations" pane open on the right, indicating you've already followed that step and are now ready to create a specific visualization. The canvas prompts you to "Build visuals with your data" by selecting or dragging fields from the "Fields" pane (not visible in this image but typically on the right) onto the report canvas.

From the "Visualizations" pane on the right, select the line chart icon (a chart with lines and dots) to create a time series visualization for real-time sales analytics, such as tracking chocolate sales over time using the "Date" or "Year-Month" fields from your dataset. Then, drag the appropriate fields (e.g., "Date" to the Axis and "Sales" or a similar metric to the Values) from the "Fields" pane onto the line chart to visualize sales trends.

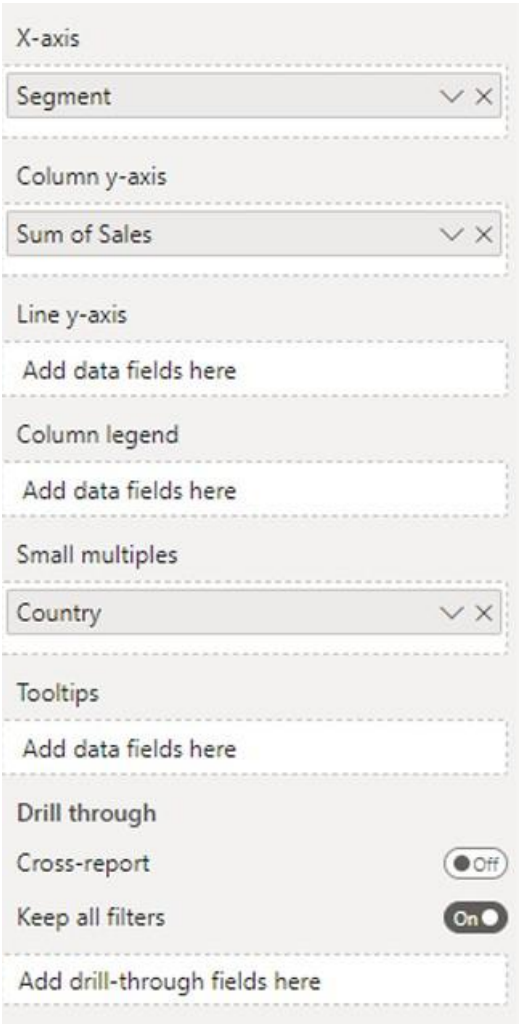
Step 5:

Next step was to select the line chart icon from the "Visualizations" pane and drag fields like "Date" and "Sales" from your chocolate sales dataset to create a time series visualization for real-time sales analytics. The current image shows the "Format visual" pane in Power BI, specifically for a slicer visual, with options like "Slicer settings," "Style" (set to "Vertical list"), "Selection," "Slicer header," and "Values." This indicates you've already added a slicer to your report, likely to filter data (e.g., by date, product, or region), and are now customizing its appearance or behavior.



To enhance your dashboard for real-time sales analytics, go back to the "Visualizations" pane, select the line chart icon again (if not already present), and ensure it's placed on the report canvas. Then, drag the "Date" field to the Axis and a sales-related field (e.g., "Sales Amount" or a similar metric) to the Values to create or refine the time series visualization of chocolate sales trends, allowing you to analyze data over time while using the slicer to filter interactively.

Step 6:



The image shows the "Fields" section of a combo chart (line and column chart) in the Power BI "Visualizations" pane, with "Segment" assigned to the X-axis, "Sum of Sales" to the Column y-axis, and "Country" to Small multiples. This configuration suggests you're creating a visualization to analyze chocolate sales data, comparing sales across segments (e.g., product types or categories) and breaking them down by country. To integrate this into your report for real-time sales analytics using the time series dataset, follow these steps:

1. Verify the Data Fields: Ensure the fields "Segment," "Sum of Sales," and "Country" are correctly pulled from your chocolate sales dataset in the "Fields" pane. If needed, adjust by dragging the appropriate fields (e.g., replace "Segment" with "Date" or "Year-Month" for a time series focus, and ensure "Sales" reflects your sales metric).

2. Adjust for Time Series: To align with real-time sales analytics, modify the X-axis to use a time-based field like "Date" or "Year-Month" instead of "Segment." Drag the "Date" or "Year-Month" field from the "Fields" pane to the X-axis, replacing "Segment," to create a time series visualization of sales trends.

3. Enhance the Visualization: Keep "Sum of Sales" on the Column y-axis to show sales amounts over time. If you want to compare sales across countries, retain "Country" in Small multiples to create separate charts for each country, or move it to the Column legend for a different breakdown.

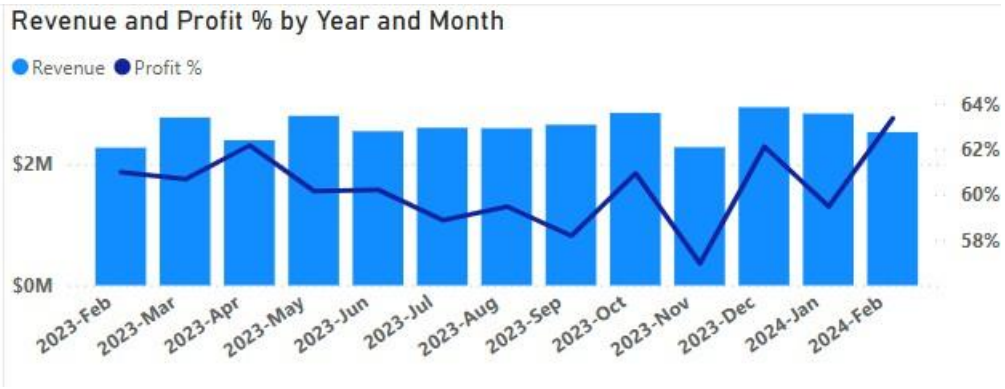
4. Format the Visual: Click on "Format visual" (next to the chart icon in the "Visualizations" pane) to customize the chart's appearance, such as colors, labels, and titles, to make it clear and visually appealing for your dashboard.
5. Add to the Report Canvas: Drag the combo chart from the "Visualizations" pane onto the report canvas if it's not already there. Position it alongside other visuals (e.g., slicers or line charts) to create a comprehensive dashboard for analyzing chocolate sales trends over time.
6. Interact with Slicers: If you have slicers (e.g., from previous steps), ensure they're connected to filter this chart by dragging relevant fields like "Date" or "Country" into the slicer visual. This allows real-time filtering of the sales data.
7. Save and Publish: Once satisfied with the visualization, save your Power BI file by clicking "File" > "Save" in the top-left corner. To share the dashboard, publish it to Power BI Service by clicking "File" > "Publish" > "Publish to Power BI."

Step 7:

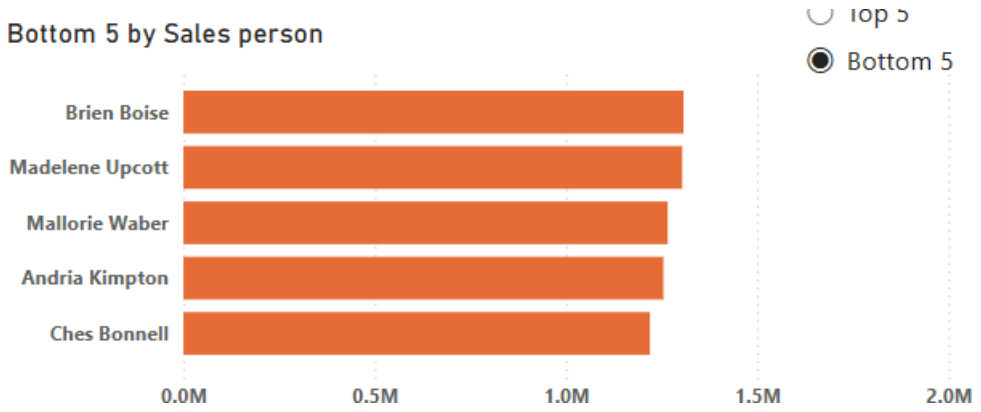
This visualized data in Power BI is likely represented using a Card visualization for each metric (Revenue, Total Boxes, Profit %, Total Profit, Total Transactions, Low-Box Shipments). Each card displays a single key value ($34M, 2M, 60.29%, $21M, 6113, 624) with its corresponding label, arranged horizontally across the report canvas. You can create this in Power BI by selecting the Card icon from the "Visualizations" pane, dragging each metric (e.g., Sum of Revenue, Count of Boxes) from the "Fields" pane into the card, and formatting the cards for alignment and style under "Format visual." Slicers can be added to filter the data dynamically for real-time analysis.

Country	Revenue	Cost	Profit %
Australia	$5,703,536	$2,144,923	62.39%
Canada	$5,725,895	$2,371,919	58.58%
India	$5,648,465	$2,180,480	61.40%
New Zealand	$5,875,218	$2,258,013	61.57%
UK	$5,471,935	$2,339,443	57.25%
USA	$5,617,463	$2,223,664	60.42%
Total	**$34,042,511**	**$13,518,441**	**60.29%**

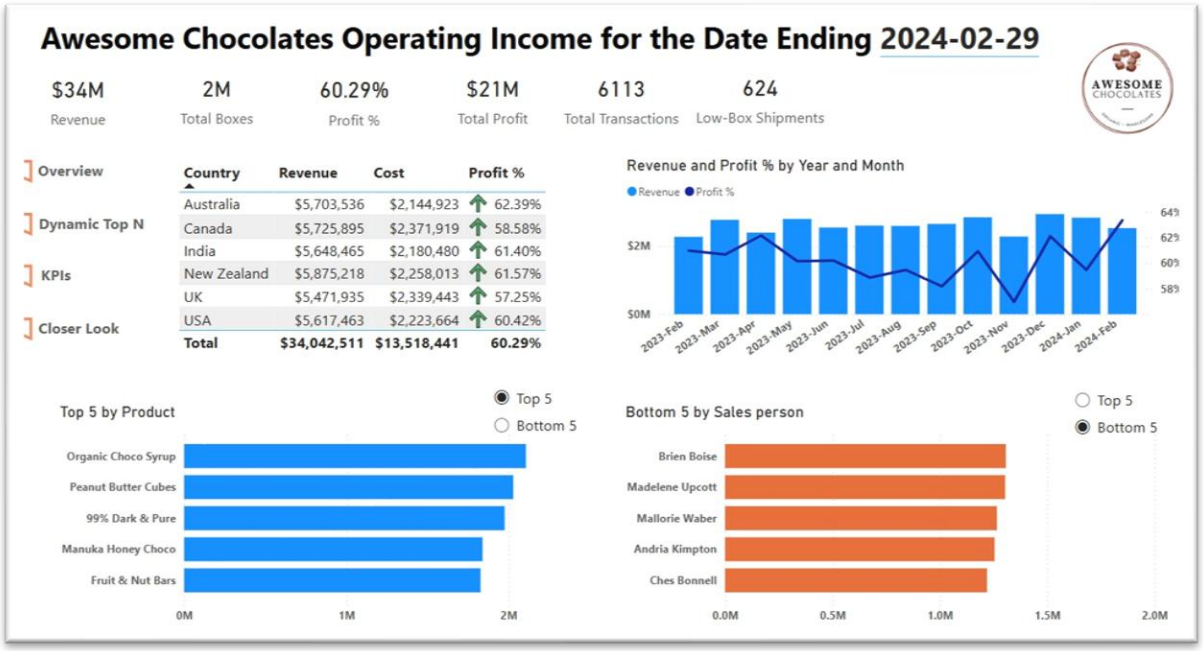
This data is visualized in Power BI using a Table visualization. The table displays columns for "Country," "Revenue," "Cost," and "Profit %," with rows for each country (Australia, Canada, India, New Zealand, UK, USA) and a total row. You can create this by selecting the Table icon from the "Visualizations" pane, dragging the "Country," "Revenue," "Cost," and "Profit %" fields from the "Fields" pane into the table, and formatting it under "Format visual" to add conditional formatting (e.g., green arrows for profit) and totals. Slicers can be added for interactive filtering by country or other fields.



This data is visualized in Power BI using a Combo Chart combining a Column Chart for Revenue and a Line Chart for Profit %. The X-axis represents months from January 2023 to February 2024, the Column y-axis shows Revenue in $M, and the Line y-axis (on the right) displays Profit % (58%-64%). To create this, select the Combo Chart icon in the "Visualizations" pane, drag "Month" to the Axis, "Revenue" to the Column values, and "Profit %" to the Line values, then format under "Format visual" to adjust colors and scales.



The chart displays the "Bottom 5 by Sales person" (Brien Boise, Madelene Upcott, Mallorie Waber, Andria Kimpton, Ches Bonnell) on the y-axis, with sales amounts (0M to 2M) on the x-axis. To create this, select the Bar Chart icon in the "Visualizations" pane, drag "Sales Person" to the Axis and a sales metric (e.g., Sum of Sales) to the Values, sort by sales in ascending order to show the bottom 5, and format under "Format visual" to adjust colors and labels.



This dashboard for Awesome Chocolates Operating Income is visualized in Power BI using a combination of visualizations:

- Card visualizations display key metrics (Revenue: $34M, Total Boxes: 2M, Profit %: 60.29%, Total Profit: $21M, Total Transactions: 6113, Low-Box Shipments: 624).
- A Table visualization under "Country" shows Revenue, Cost, and Profit % by country, with totals.
- A Combo Chart (Column for Revenue, Line for Profit %) tracks trends by year and month.
- Bar Charts highlight the Top 5 Products by sales and Bottom 5 Sales Persons, created by dragging respective fields to the Axis and Values, sorting, and formatting under "Format visual." Slicers or filters can enhance interactivity.

RESULT:

The analysis in Power BI, applied to the Excel chocolate sales data, generates a dynamic dashboard with adjustable Target Profit on Card visuals ($21M + adjustment), a Line Chart showing Profit % trends (58%-64%) with a projected line, a Table updating Country-wise metrics (e.g., Australia: $5.7M, 62.39% + impact), and Bar Charts for Top 5 Products and Bottom 5 Sales Persons. This setup enables real-time simulation of sales target impacts for enhanced decision-making.

Ex.No: 10 (b)	DATA ANALYSIS USING TIME SERIES ANALYSIS
Date:	

AIM:

To perform the time series analyze in POWER BI by using python script and visualize the charts.

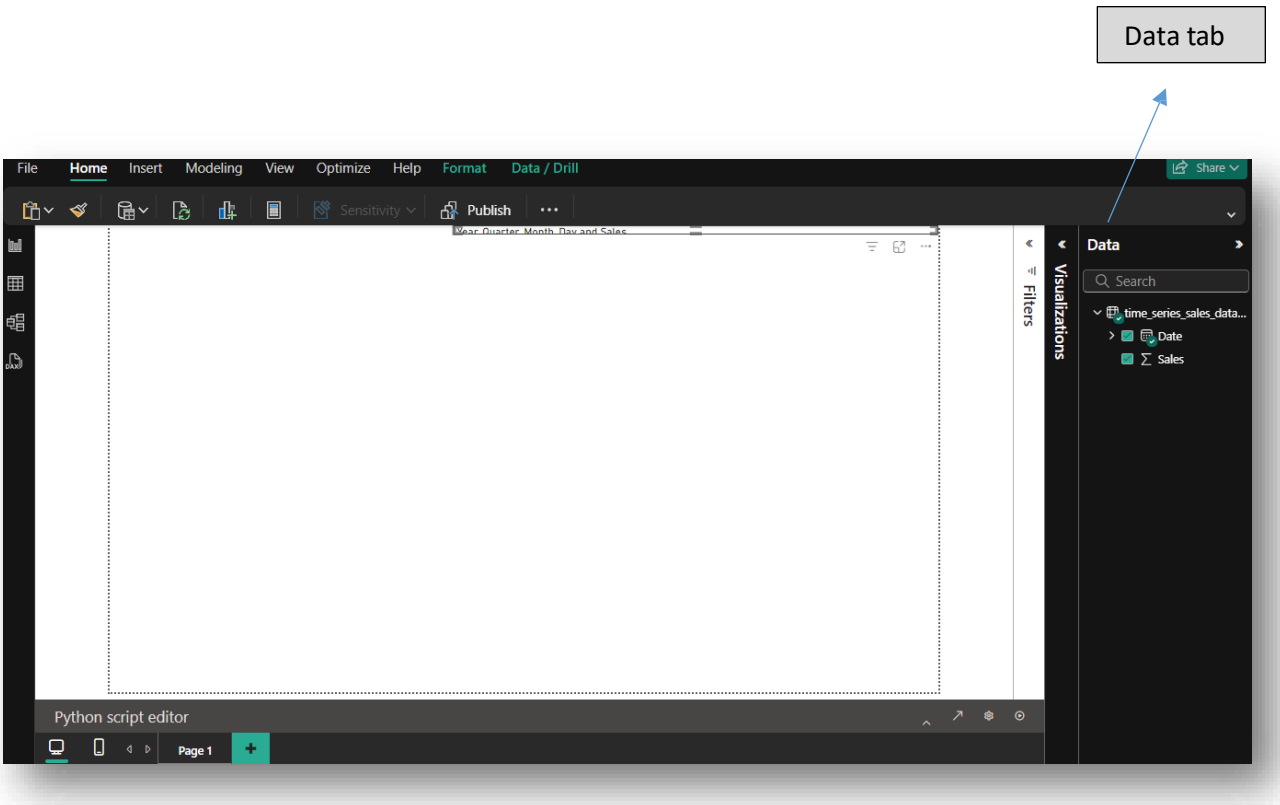
IMPLEMENTATION:

1. Install and Open Power BI

1. **Download & Install Power BI**
   - Visit the official Power BI Download Page.
   - Download and install **Power BI Desktop**.
2. **Open Power BI Application**
   - Launch the Power BI Desktop application after installation.

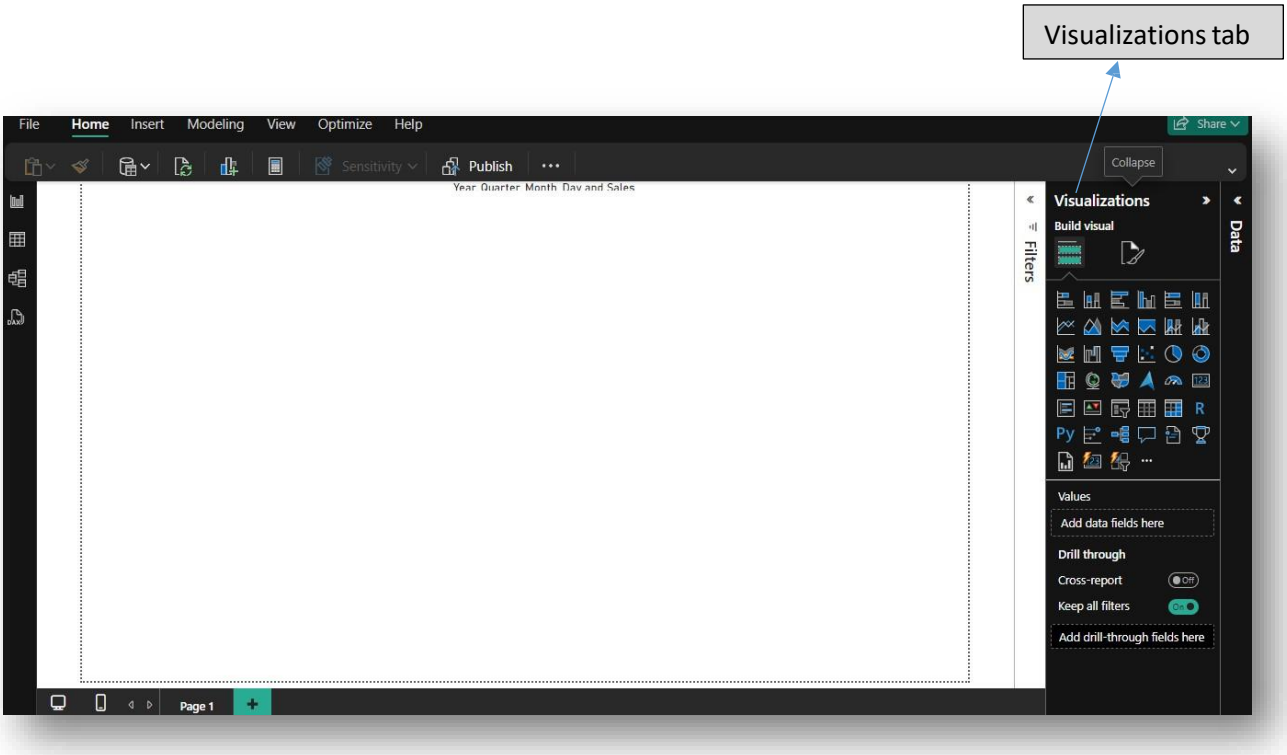
2. Load the Time Series Dataset

3. **Download the Dataset**
   - Ensure you have a **time series dataset** in a suitable format (CSV, TXT, etc.).
   - Example dataset: **"time_series_sales_dataset.csv"**.
4. **Import the Dataset into Power BI**
   - Open **Power BI Desktop**.
   - Click **Home > Get Data > Text/CSV**.
   - Select **"time_series_sales_dataset.csv"** and click **Load**.
   - Ensure that the **Date column** is formatted as **Date/Time**.
5. **Verify the Imported Data**
   - Click the **Data View** option (right panel) to preview the dataset.



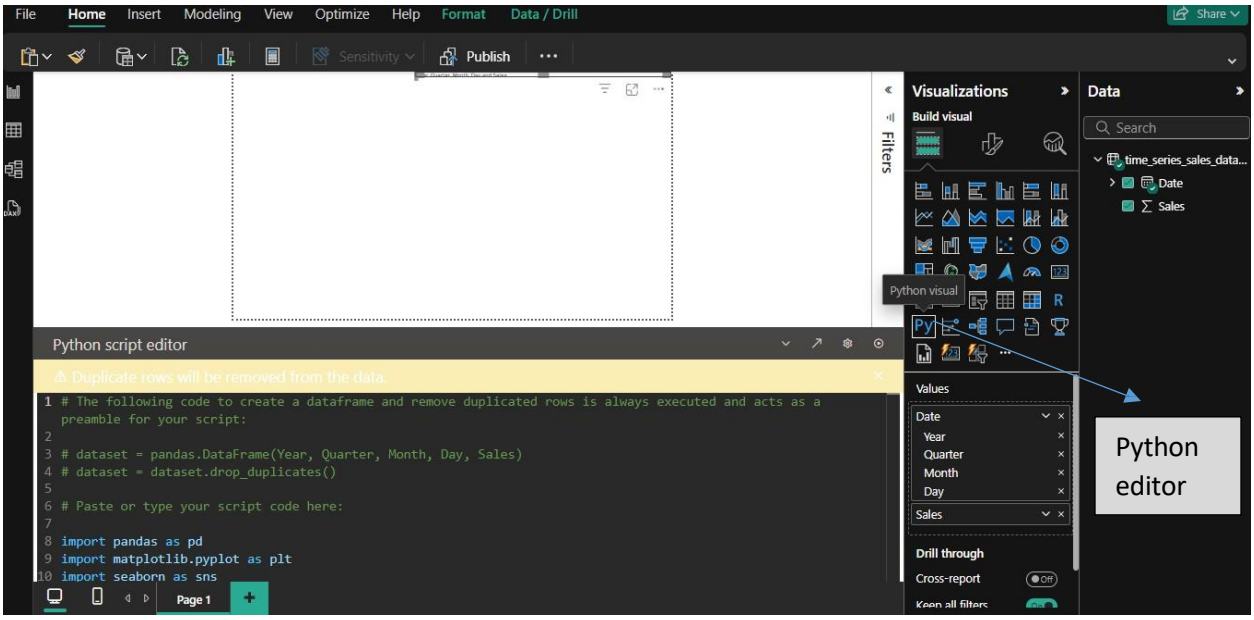
3. Visualize Time Series Data in Power BI

6. **Open the Visualization Pane**
   - Click the **Report View** tab.
   - Click the **"Visualization"** pane to see all available visualizations.
7. **Create a Time Series Line Chart (optional)**
   - From **Visualizations**, select the **Line Chart**.
   - **Drag Date** to the **X-axis**.
   - **Drag Sales** to the **Y-axis**.
   - Title the chart as **"Time Series Sales Analysis"**.



4. Run Python Script for Advanced Analysis

8. **Enable Python Scripting in Power BI**
   - Click **File > Options and Settings > Options**.
   - Under **Python scripting**, set the Python home directory (Anaconda/Miniconda).
9. **Add Python Script in Power BI**
   - Click **"Python Visual" (Py icon)** in the **Visualization Pane**.
   - Select the required **columns** from the dataset.

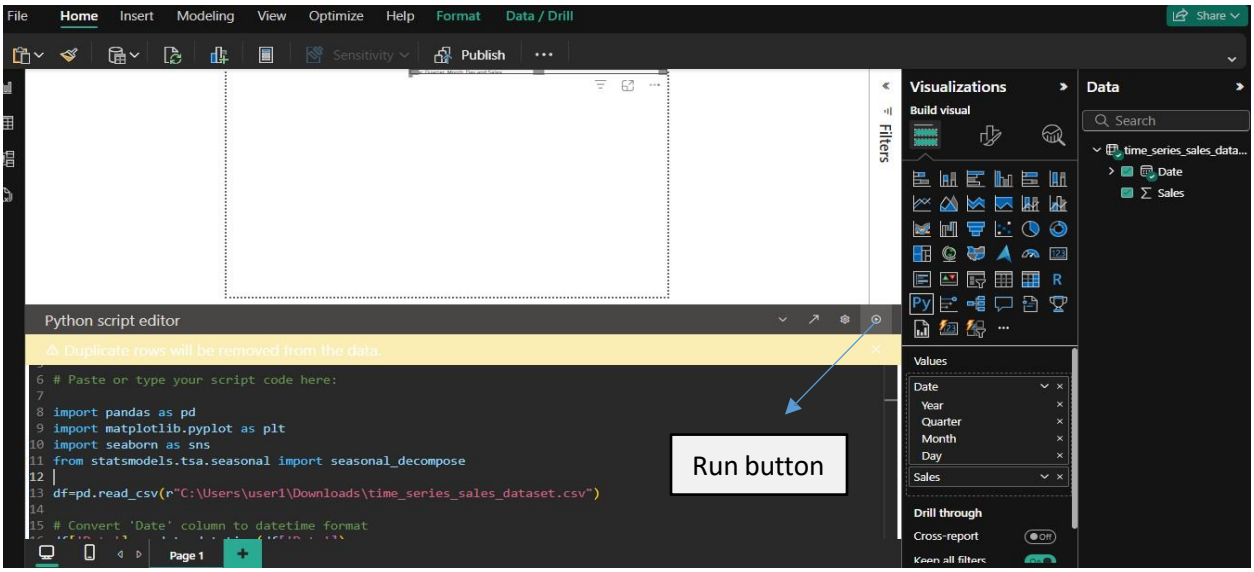


10. **Paste and Run the Python Script**

- Open the **Python Script Editor** in Power BI.
- Paste the following **Python script**:

11. **Run the Python Script**

- Click the **Run button** on the top of the Python script editor.
- Wait for execution to complete.



CODE:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.seasonal import seasonal_decompose

df=pd.read_csv(r"C:\Users\user1\Downloads\time_series_sales_dataset.csv")

# Convert 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Set 'Date' as the index (Required for time series analysis)
df.set_index('Date', inplace=True)

# Perform Seasonal Decomposition (Additive Model)
decomposition = seasonal_decompose(df['Sales'], model='additive', period=30)  # Assuming monthly seasonality

# Extract trend, seasonal, and residual components
df['Trend'] = decomposition.trend
df['Seasonality'] = decomposition.seasonal
df['Residual'] = decomposition.resid

# Create visualizations
plt.figure(figsize=(12, 8))

# Original Sales Data
plt.subplot(4, 1, 1)
plt.plot(df.index, df['Sales'], label='Original Sales', color='blue')
plt.title('Original Sales Data')
plt.legend()

# Trend Component
plt.subplot(4, 1, 2)
plt.plot(df.index, df['Trend'], label='Trend', color='red')
plt.title('Trend Component')
plt.legend()
```

```
# Seasonal Component
plt.subplot(4, 1, 3)
plt.plot(df.index, df['Seasonality'], label='Seasonality', color='green')
plt.title('Seasonality Component')
plt.legend()

# Residual Component
plt.subplot(4, 1, 4)
plt.plot(df.index, df['Residual'], label='Residual', color='purple')
plt.title('Residual Component')
plt.legend()

plt.tight_layout()
plt.show()
```

OUTPUT:

RESULT:

The program was executed successfully.

Ex.No: 11	Data Modelling and Analytics with Pivot Table in Power BI
Date:	

AIM:

To make a pivot table for data modelling and analytics using power bi.

PROCEDURE:

- **Key Performance Indicators (KPIs) - Card Visuals**

  - Total Revenue → SUM (Orders [Revenue])
  - Total Profit → [total_revenue] - [total_cost]
  - Profit Margin (%) → DIVIDE([total_profit], [total_revenue], 0)
  - Total Orders → DISTINCTCOUNT (Orders [Order ID])
  - Average Order Value (AOV) 🏷 → DIVIDE([total_revenue], [total_orders], 0)

Visualization: Use Card visuals for each KPI.

- **Sales Performance Over Time - Line Chart**

Measure:

- Monthly Sales Trend → SUMMARIZE (Orders, FORMAT (Orders [Date], "YYYY-MM"), "Monthly Revenue", SUM (Orders [Revenue]))

Visualization: Use a Line Chart with:

- X-axis → Month-Year (YYYY-MM)
- Y-axis → Monthly Revenue

- **Top 5 Best-Selling Cookies - Bar Chart**

Measure:

- Top-Selling Cookies → TOPN(5, SUMMARIZE(Orders, Orders[Product],"Total Sales", SUM(Orders[Units Sold])), [Total Sales], DESC)

Visualization: Use a Clustered Bar Chart with:

- X-axis → Total Sales
- Y-axis → Product Name (Descending Order)

- **Customer Insights - Table or Matrix**

Measures:

- Customer-wise Revenue → CALCULATE([total_revenue], ALLEXCEPT (Orders, Orders [Customer ID]))
- Total Orders per Customer → CALCULATE([total_orders], ALLEXCEPT (Orders, Orders [Customer ID]))
- Top 5 Customers by Revenue → TOPN (5, SUMMARIZE (Orders, Orders [Customer ID], "Total Revenue", SUM (Orders [Revenue])),

Visualization: Use a Matrix or Table visual showing:

- Columns: Customer Name, Total Revenue, Total Orders

- **Profitability by Cookie Type - Pie Chart**

Measure:

- Profit per Cookie Type → SUMMARIZE(Orders, Orders[Product], "Total Profit", SUM(Orders[Revenue]) - SUM(Orders[Cost]))

Visualization: Use a Pie Chart with:

- Legend → Product Name
- Values → Total Profit

OUTPUT:

RESULT:

Thus pivot table for data modelling and analytics are implemented using power bi has been verified.

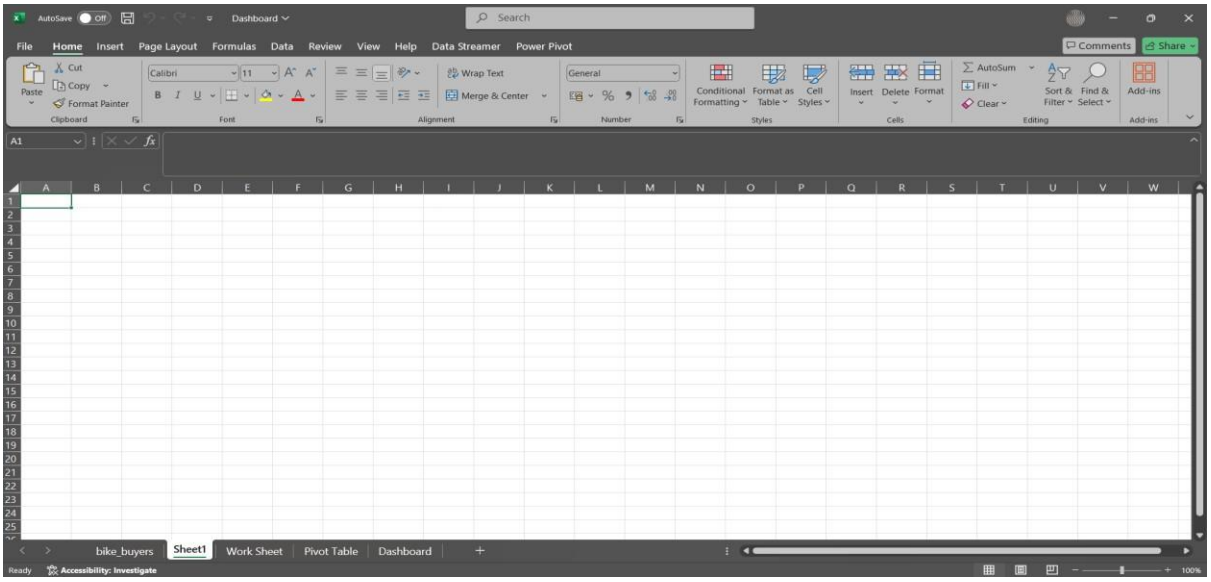
EX.NO: 12	DATA ANALYSIS AND VISUALIZATION USING ADVANCED EXCEL
DATE:	

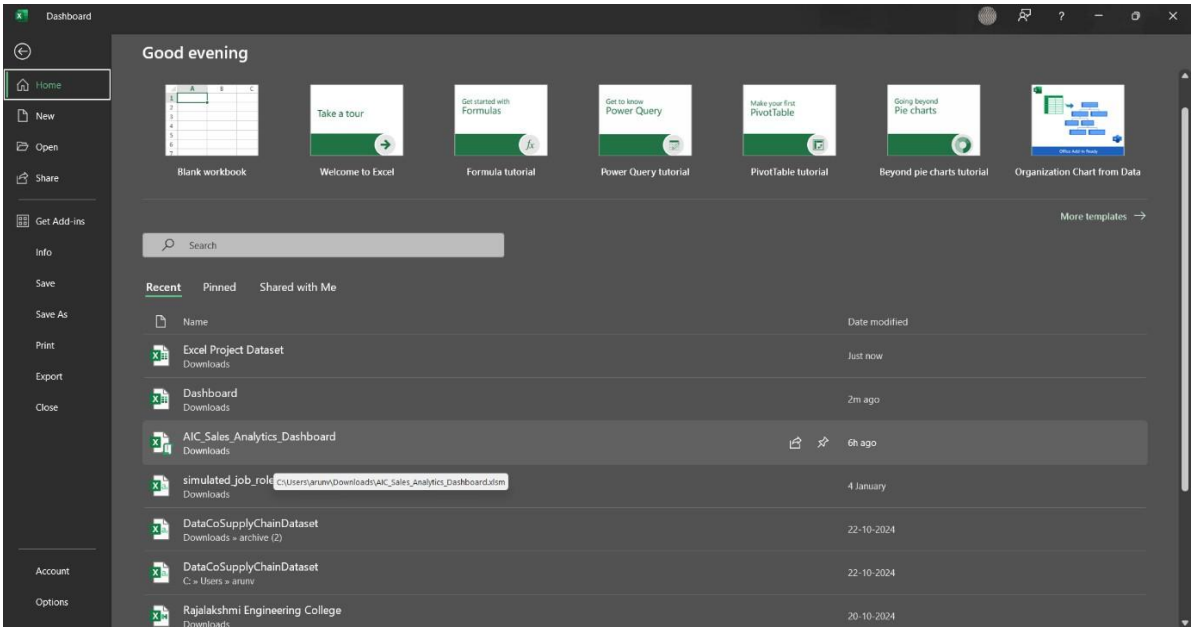
AIM:

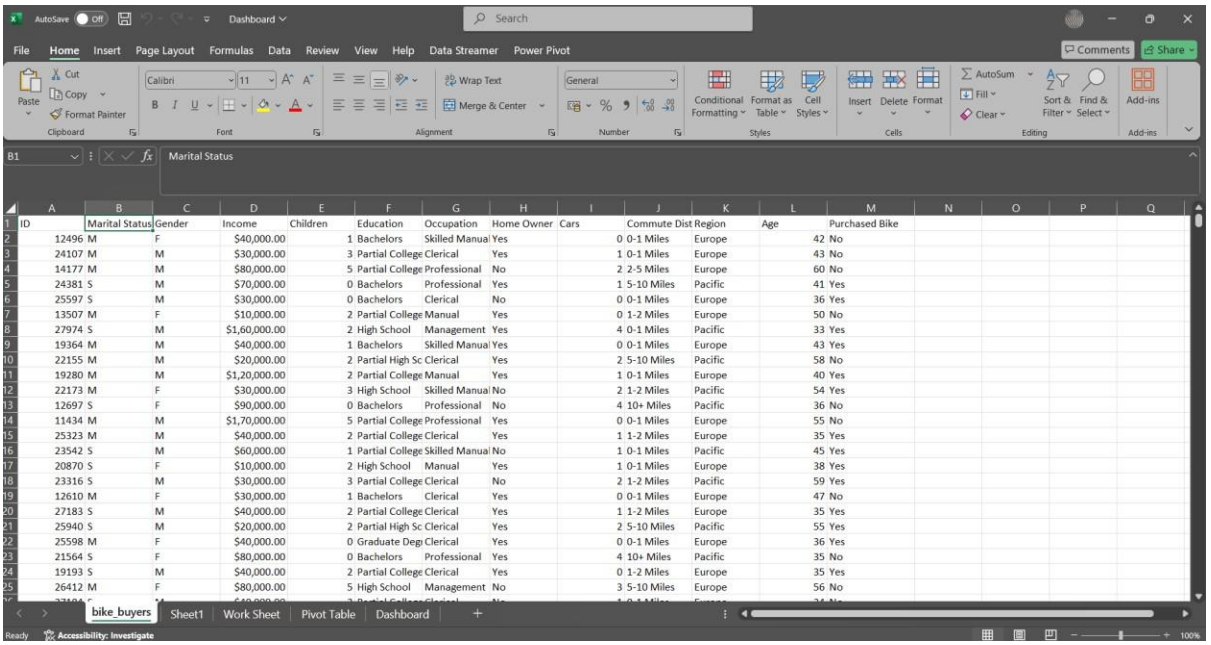
To implement the Data Analysis and Visualization using Advanced Excel.

PROCEDURE:

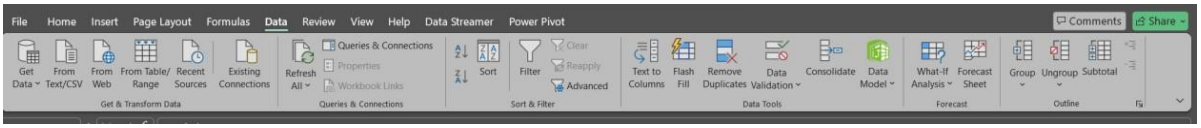
Step 1: Open the blank excel and import the Dataset into Microsoft Excel 2021.

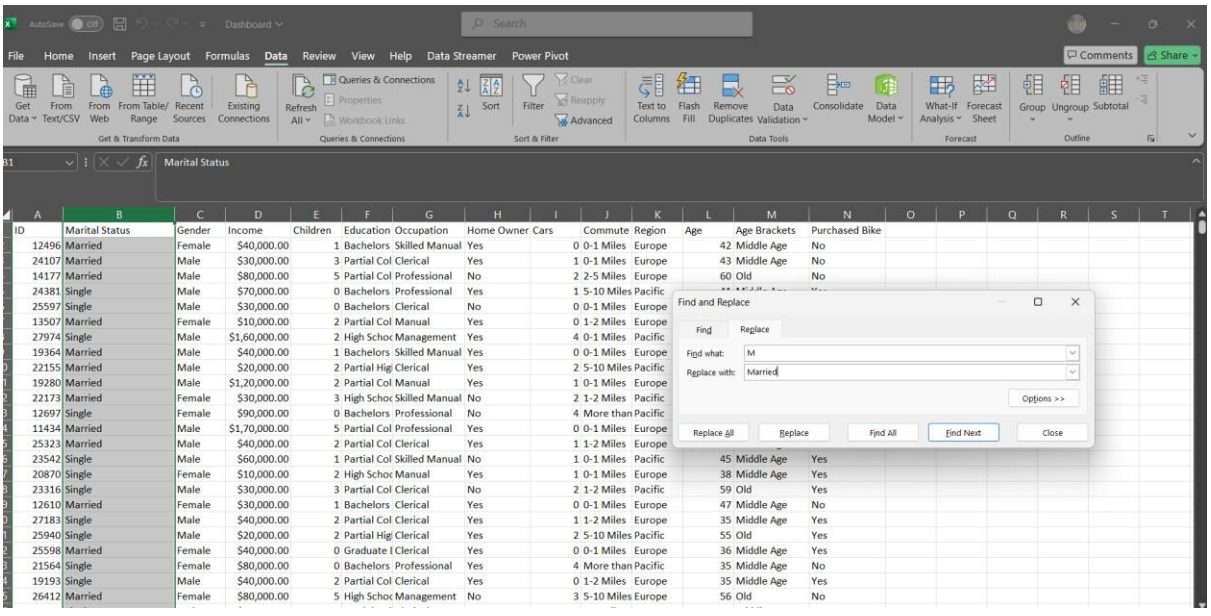




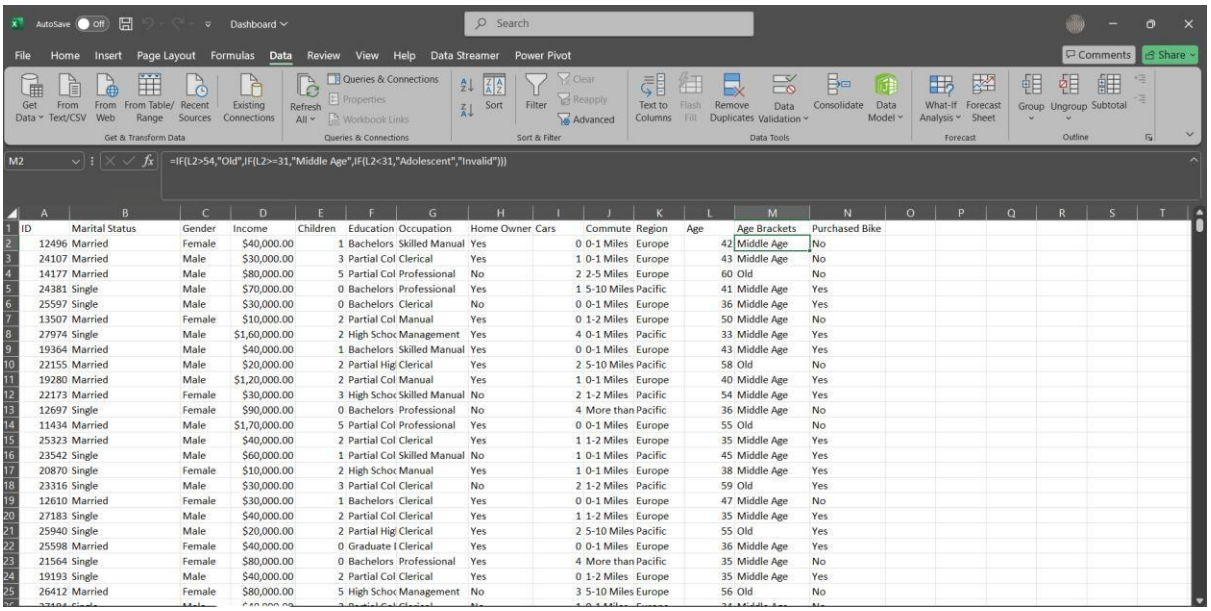


Step 2: Cleaning and pre-processing the dataset for the better visualization. Using the key Data you can remove the duplicates in the entries. To know clearly about the dataset for the user making changes in row 2 & 3.

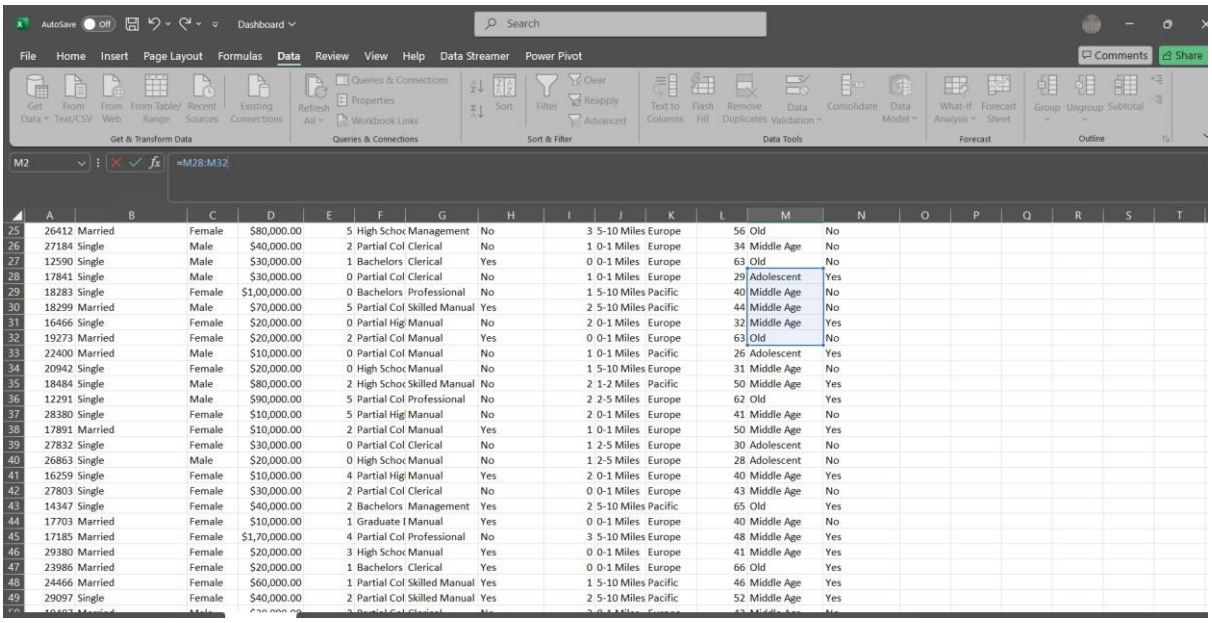




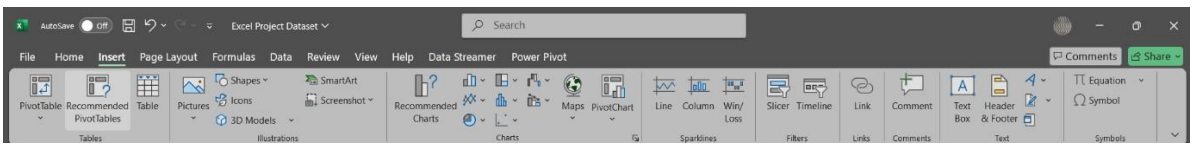
Step 3: For the Bike sales we need to capture the specific age group so include a row (Age Brackets) for the better view.



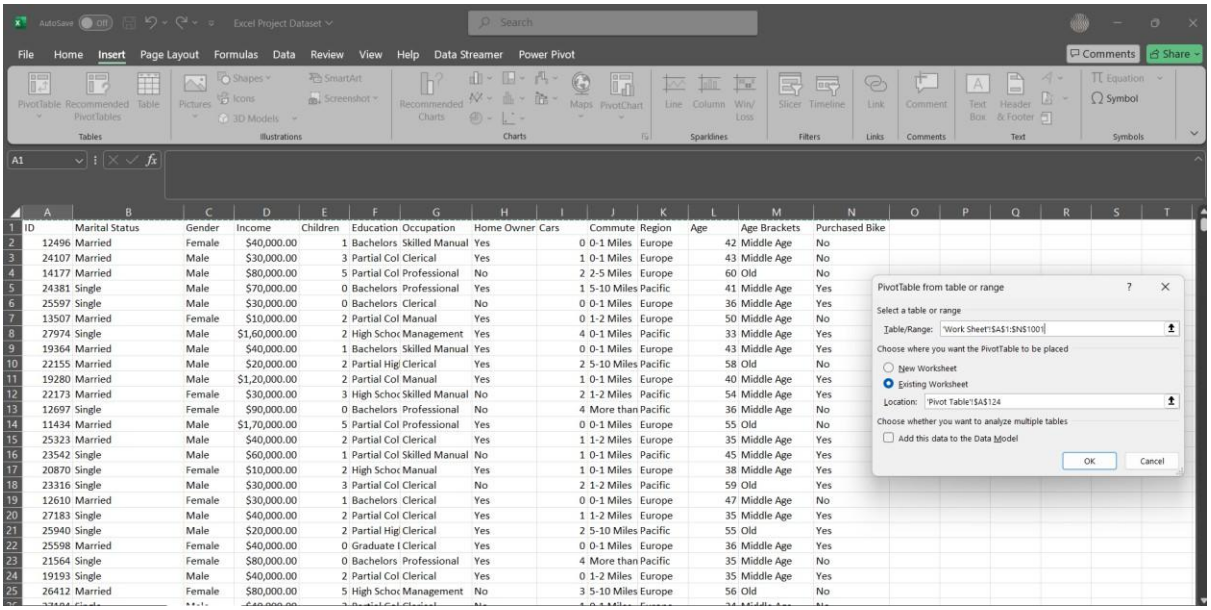
Step 4: For the different group age peoples we using the formula [=IF(L2>54,"Old",IF(L2>=31,"Middle Age",IF(L2<31,"Adolescent","Invalid")))]



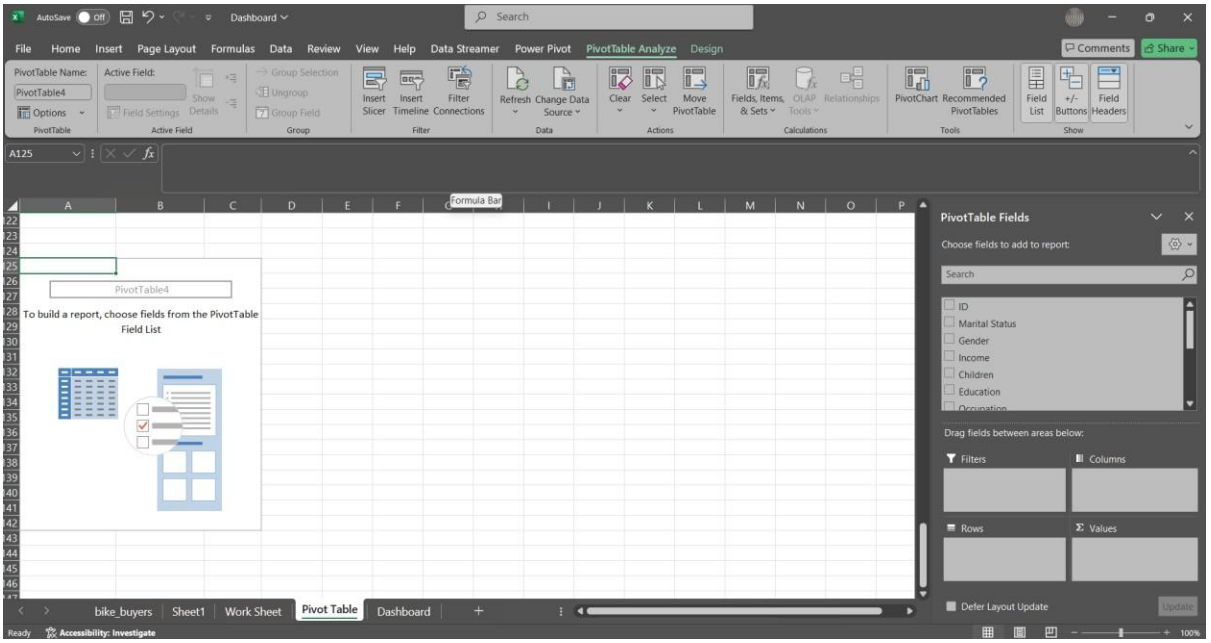
Step 5: After the completion of the data cleaning and pre-processing now we can start to create the pivot table. Make an add of a sheet in the same board to work pivot table in different table.



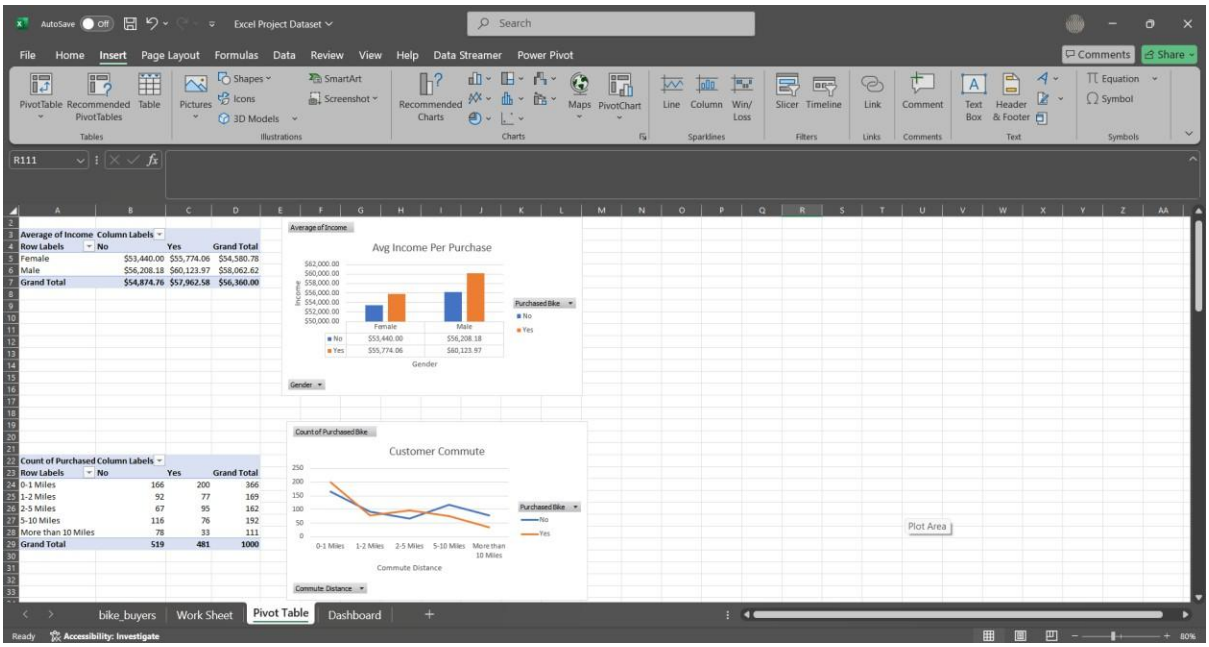
Step 6: Select the pivot table and in the down, you can see the dataset select that sheet and give ctrl+a to work on the pivot table.

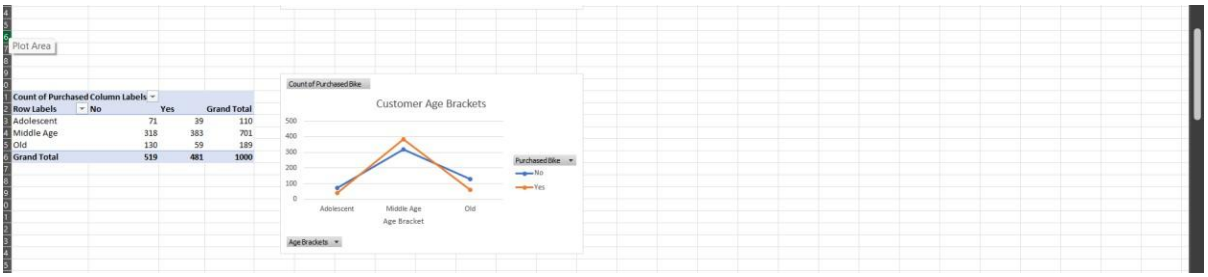


Step 7: For the user needs we can select and drag the option to make chart with following datasets to make understand the user we can use charts too.

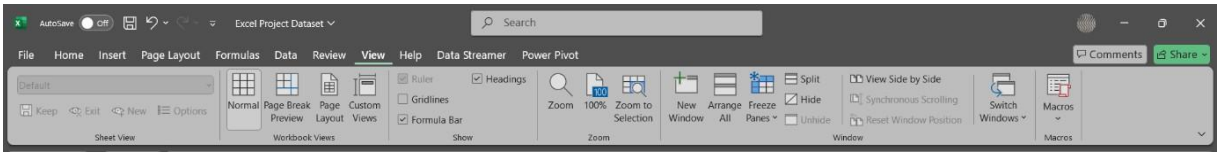


Step 8: Using recommended charts we can visualize the total value which we taken in pivot table.

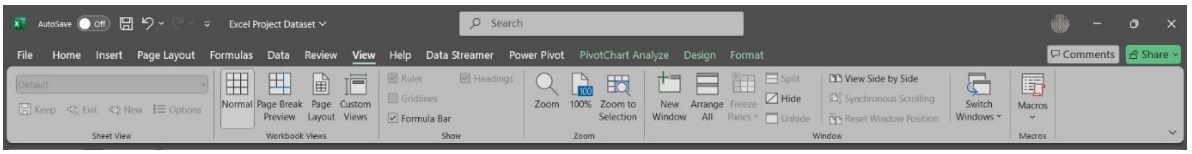




Step 9: Now creating a Dashboard in the excel sheet same like pivot table add a sheet for dashboard. Using the view option you can edit the grid lines.



Step 10: Design the dashboard to look with informative and user friendly. Using pivot chart analysis option, you can work on the design and slicer to know about the detail information of pivot chart.



RESULT:

Thus the Data Analysis and Visualization using Advanced Excel was implemented successfully.