

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	Date	Name of the Experiment	Pg.No	Sign
1 (a)		Import the legacy data from different sources such as (Excel, Sql Server, Oracle etc.) and load in the target system	1	
1 (b)		Import the legacy data from different sources such as (excel, sqlserver, wikipedia etc.) And load in the target system	4	
2		Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI	13	
3		Data Visualization from ETL Process	18	
4		Creating a Cube in SQL server 2012	23	
5		Apply the what – if Analysis for data visualization to design and generate necessary reports based on the data warehouse data.	40	
6		Implementation of Classification algorithm in R Programming	48	
7		Practical Implementation of Decision Tree using R Tool	52	
8		k-means clustering using R	57	
9		Prediction Using Linear Regression	62	
10 (a)		Data Analysis using Time Series Analysis	68	
10 (b)		Data analysis using time series analysis	76	
11		Data Modelling and Analytics with Pivot Table in Excel	82	
12		Data Analysis and Visualization using Advanced Excel	84	

EX.NO : 1 (a)

DATE :

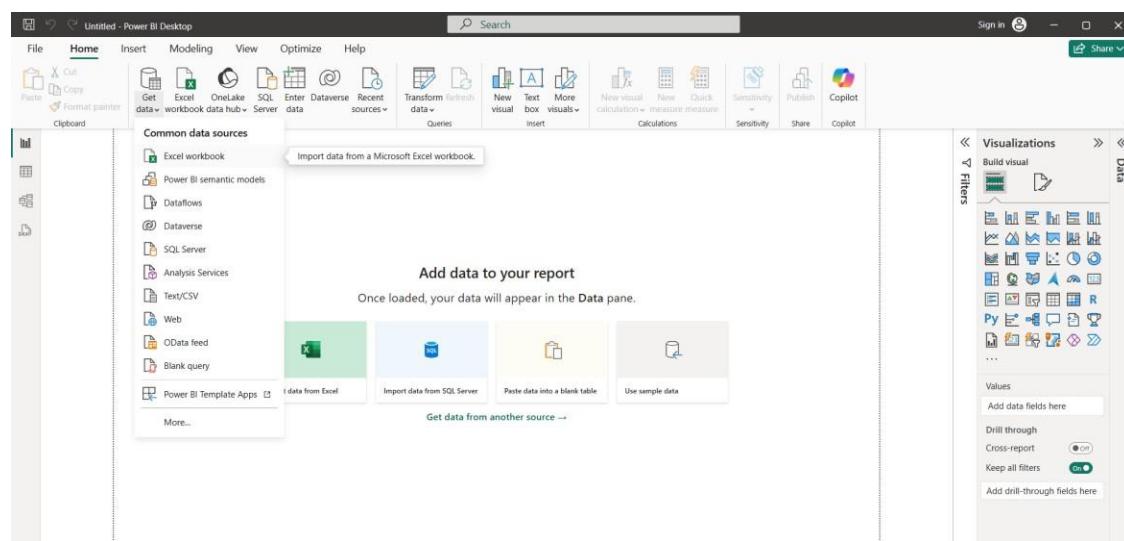
IMPORT THE LEGACY DATA FROM DIFFERENT SOURCES SUCH AS (EXCEL, SQLSERVER, ORACLE ETC.) AND LOAD IN THE TARGET SYSTEM

AIM:

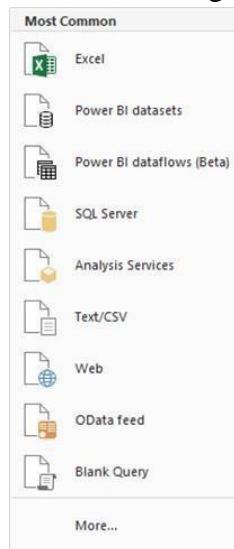
To import the legacy data from different sources such as (Excel, SQLServer, Oracle etc.) and load in the target system.

PROCEDURE:

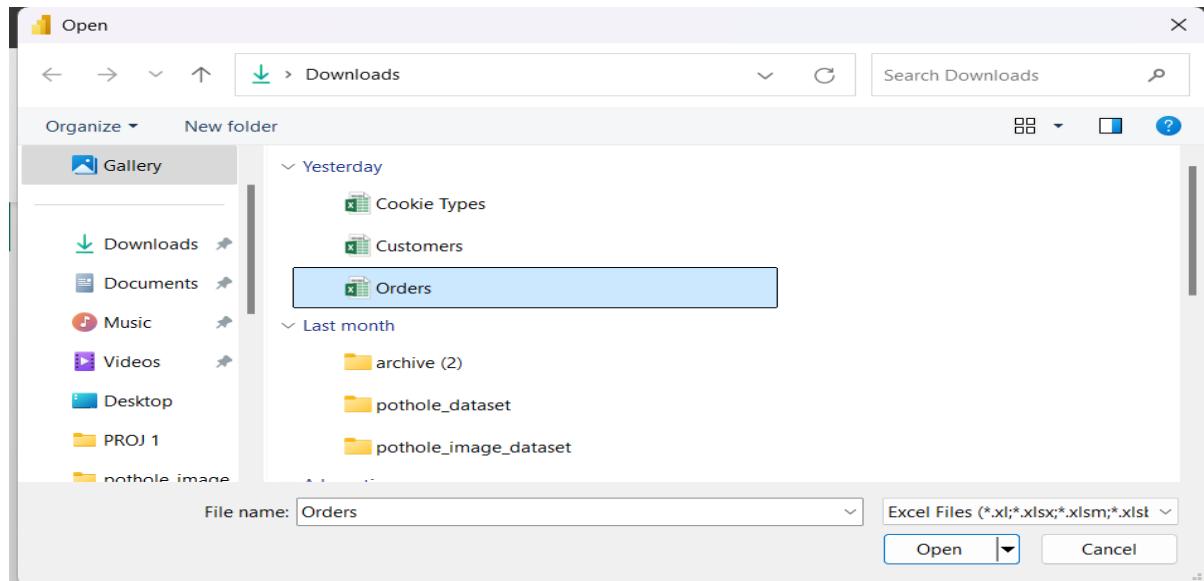
Step 1: Open Power BI



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

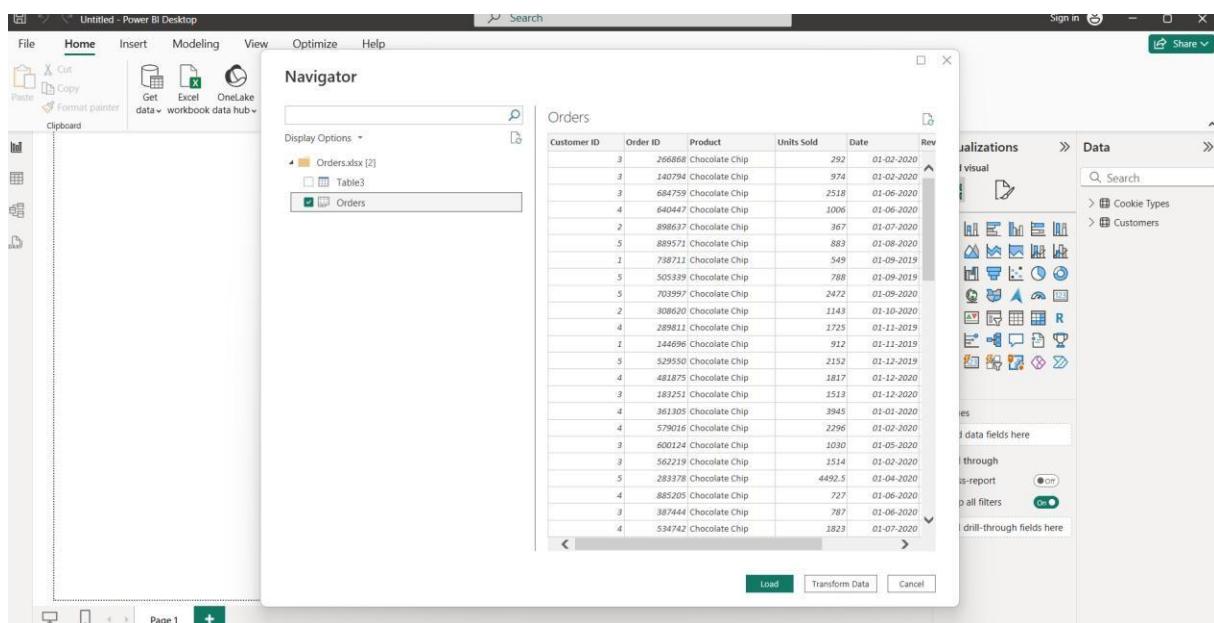
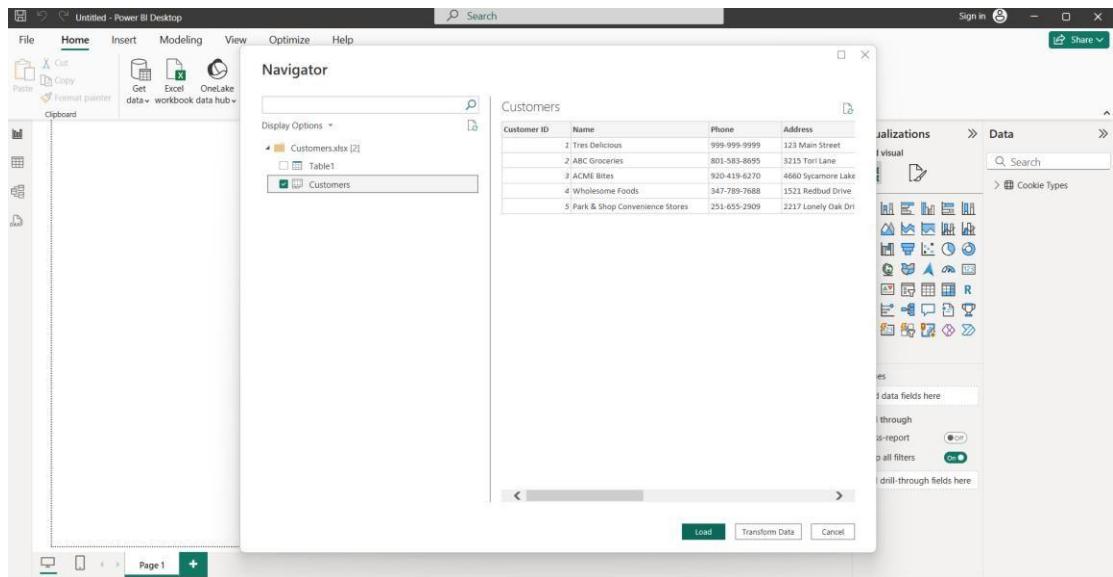


Step 4: Select file and load the data

A screenshot of Power BI Desktop. The 'Home' tab is selected. A 'Clipboard' ribbon tab is visible. In the center, a 'Navigator' window is open, showing the 'Cookie Types' table. The table has columns: Cookie Type, Units Sold, Revenue Per Cookie, and Cost Per Cookie. The data is as follows:

Cookie Type	Units Sold	Revenue Per Cookie	Cost Per Cookie
Chocolate Chip	338239.5	5	
Fortune Cookie	154198	1	
Oatmeal Raisin	155315	5	
Snickerdoodle	146846	4	
Sugar	168783	3	
White Chocolate Macadamia Nut	162424.5	6	

At the bottom of the Navigator window are 'Load', 'Transform Data', and 'Cancel' buttons. To the right, the 'Visualizations' pane shows various chart and report icons. The 'Filters' pane includes sections for 'Values', 'Drill through', 'Cross-report', 'Keep all filters', and 'Add drill-through fields here'.



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)

DATE :

IMPORT THE LEGACY DATA FROM DIFFERENT SOURCES SUCH AS (EXCEL, SQLSERVER, WIKIPEDIA ETC.) AND LOAD IN THE TARGET SYSTEM

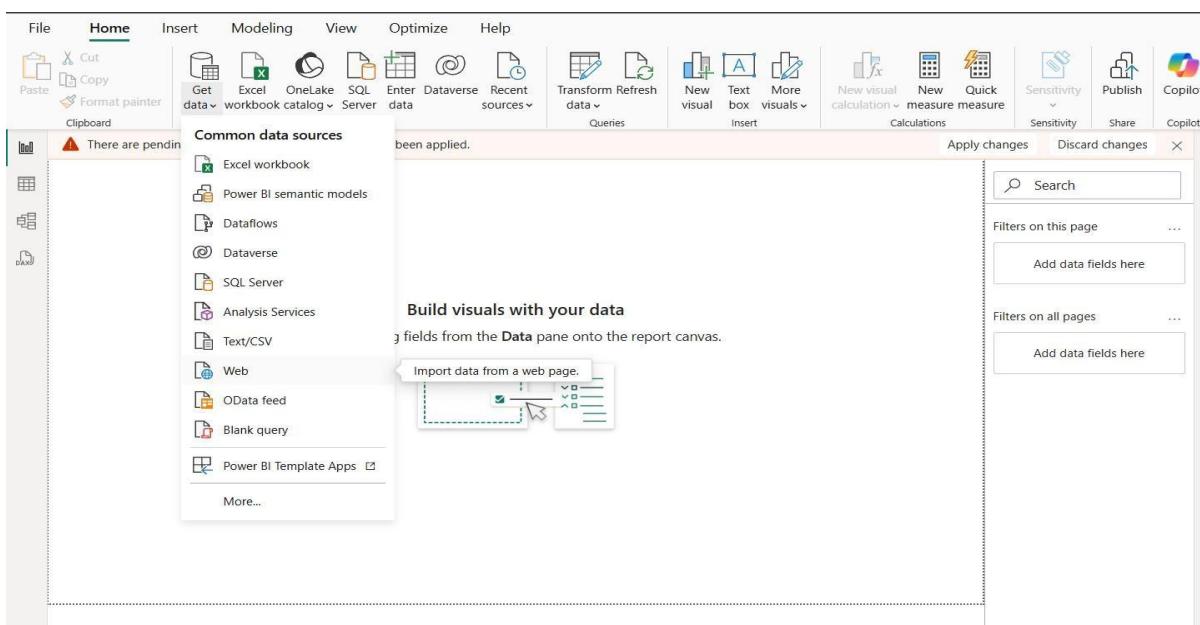
AIM:

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

- Click on Home → Select Get Data → Choose Web.



- 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.

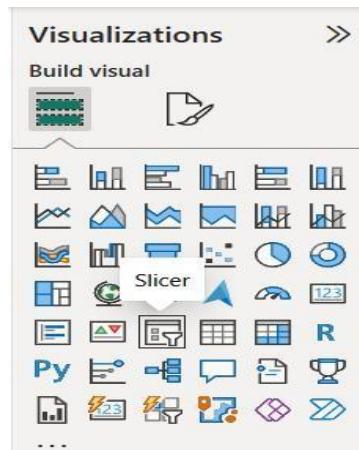
The screenshot shows the Power BI Navigator window. On the left, there is a tree view under 'HTML Tables [55]' containing items like Table 1 through Table 19. 'Table 11' is selected, indicated by a checked checkbox and highlighted with a light gray box. On the right, the 'Table View' tab is active, displaying 'Table 11' with the following data:

No.	Nation	Gold	Silver	Bronze	Total
1	Norway (NOR)	148	134	123	405
2	United States (USA)	114	121	95	330
3	Germany (GER)	104	98	65	267
4	Soviet Union (URS)	78	57	59	194
5	Canada (CAN)	77	72	76	225
6	Austria (AUT)	71	88	91	250
7	Sweden (SWE)	65	51	60	176
8	Switzerland (SUI)	63	47	58	168
9	Netherlands (NED)	53	49	45	147
10	Russia (RUS)	47	39	35	121

At the bottom of the window, there are buttons for 'Add Table Using Examples', 'Load' (which is green), 'Transform Data', and 'Cancel'.

Step 2: Create a Slicer for Country Selection.

- a) Go to Visualizations Pane → Click on Slicer.

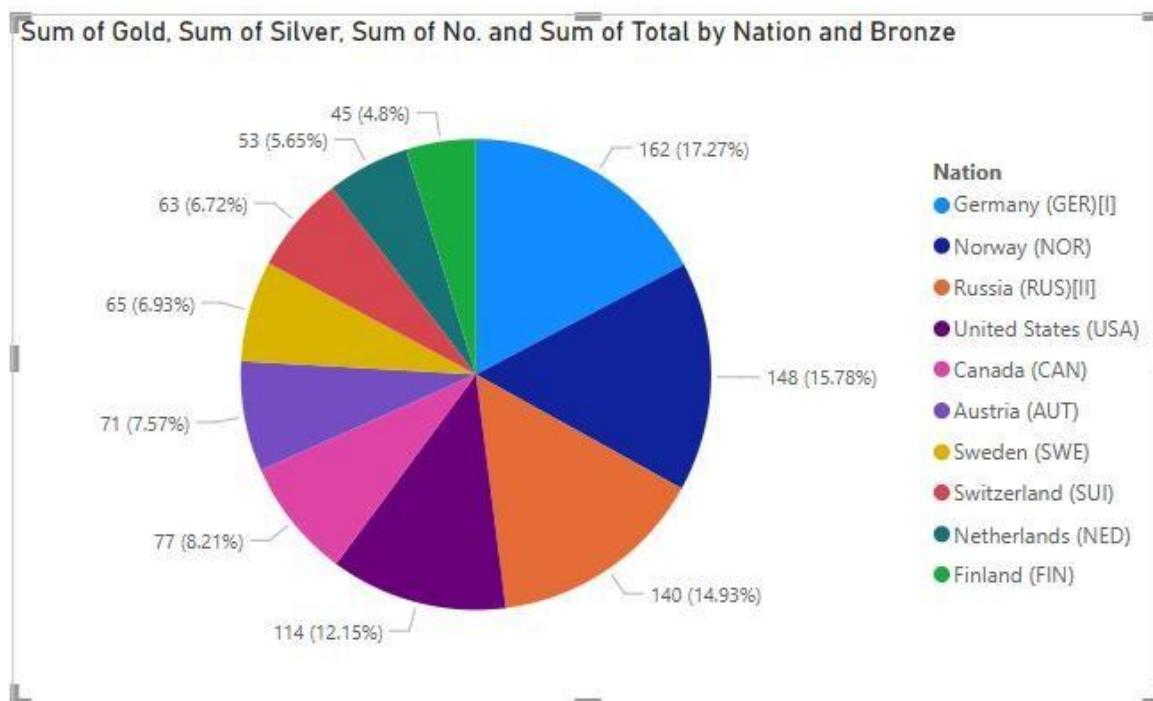


- b) Drag the Country field into the Slicer visualization.

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

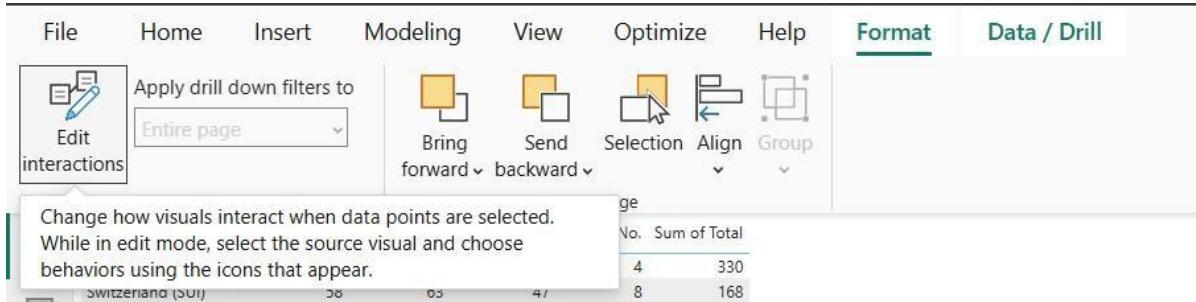
Step 3: Create a Visualization for Medal Data.

- 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.

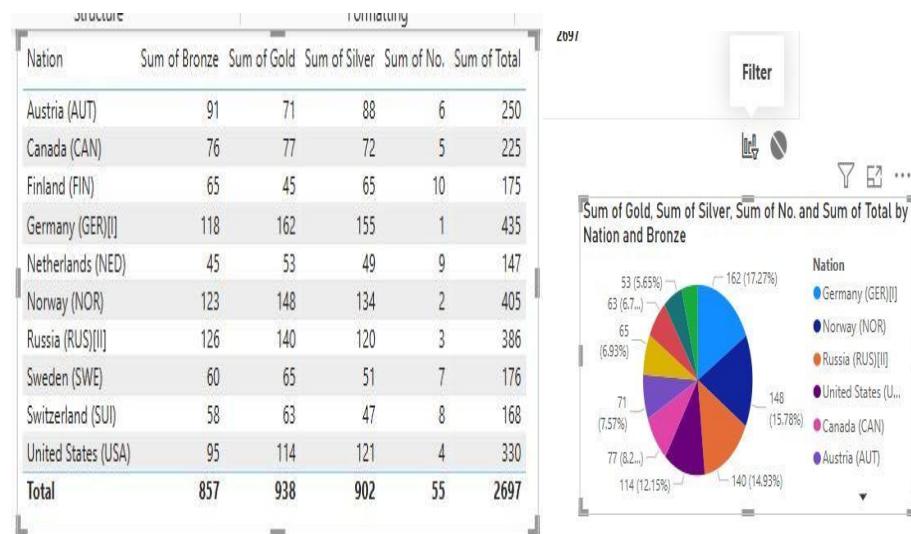


Step 4: Enable Interactive Filtering.

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



- b) Click on the Filter Icon on the visualization.

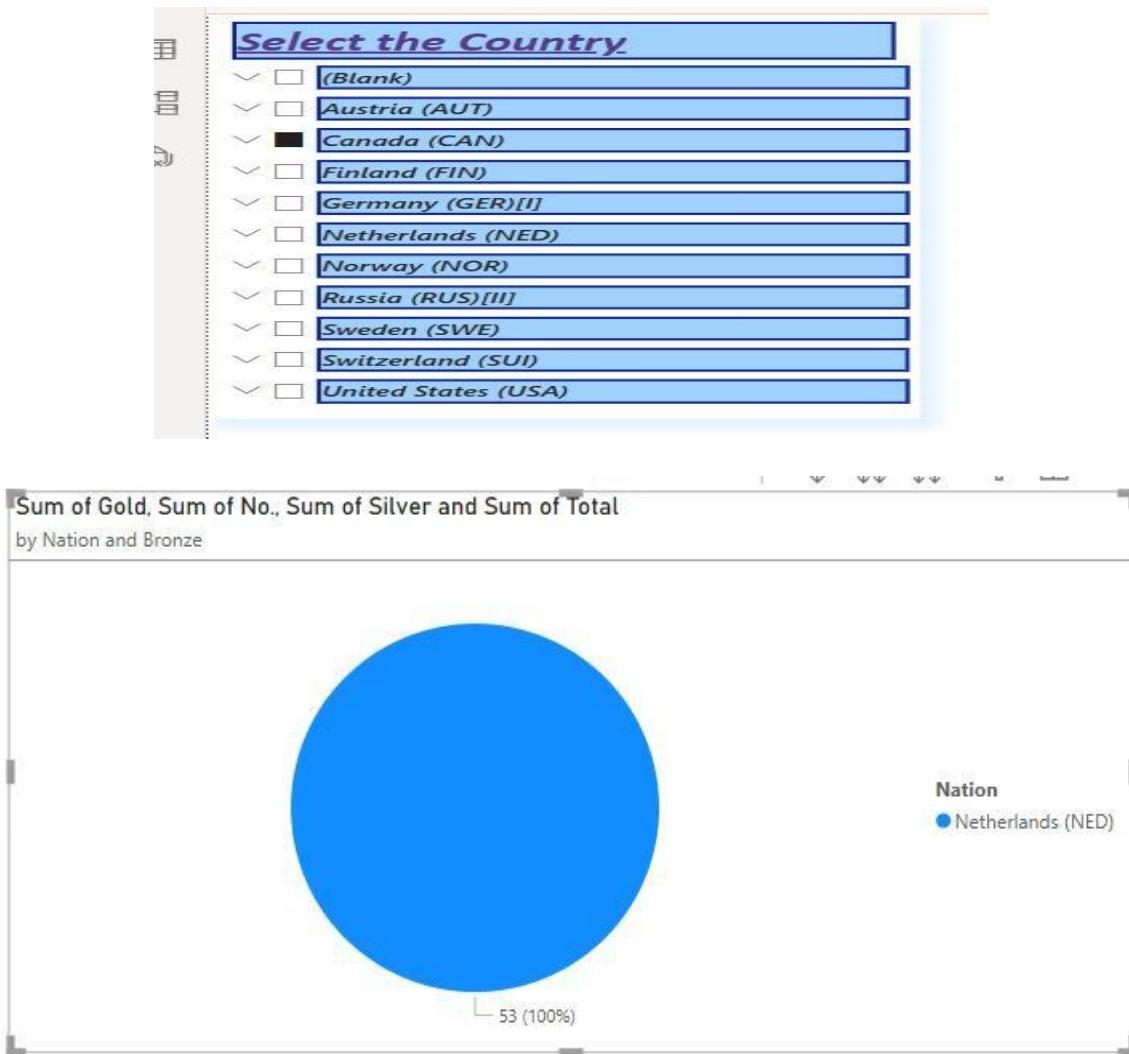


- c) Go to Visualization in the format visual can able to add the filter.



Step 5: Testing & Final Adjustments.

- Click on different countries in the slicer to test interactivity.



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

DAX queries for the slicer selection:

1. Get the Selected Country Name:

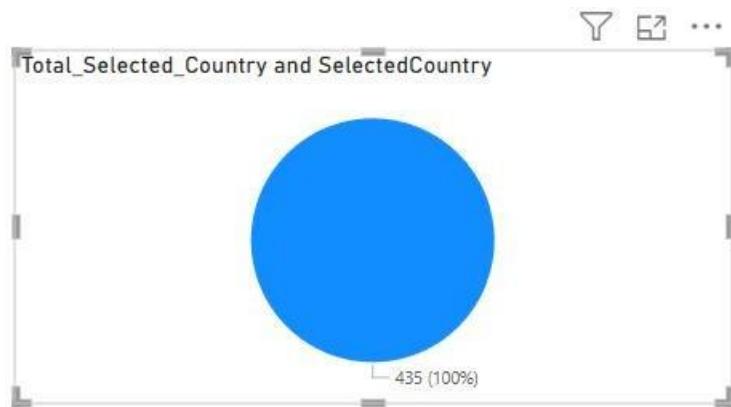
```
DAX
```

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

2. Filter Total Medals for Selected Country:

```
DAX
```

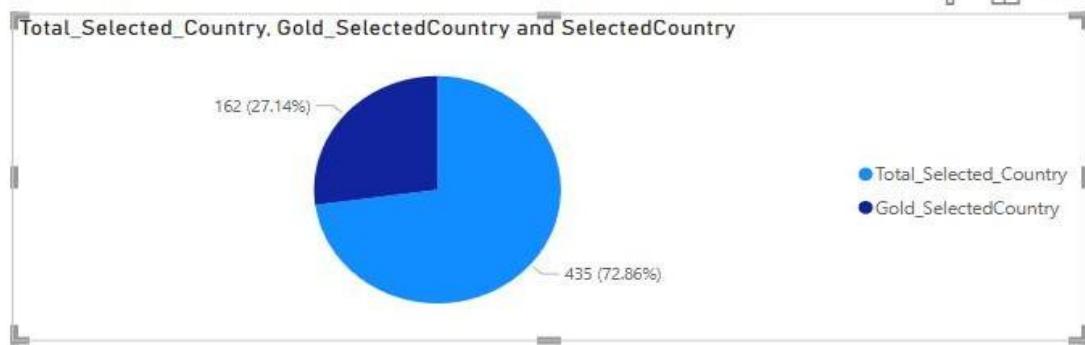
```
Total_Selected_Country =  
CALCULATE(  
    SUM('MedalsData'[Total]),  
    'MedalsData'[Nation1] = SELECTEDVALUE('MedalsData'[Nation1])  
)
```



3. Get Individual Medal Counts:

```
DAX
```

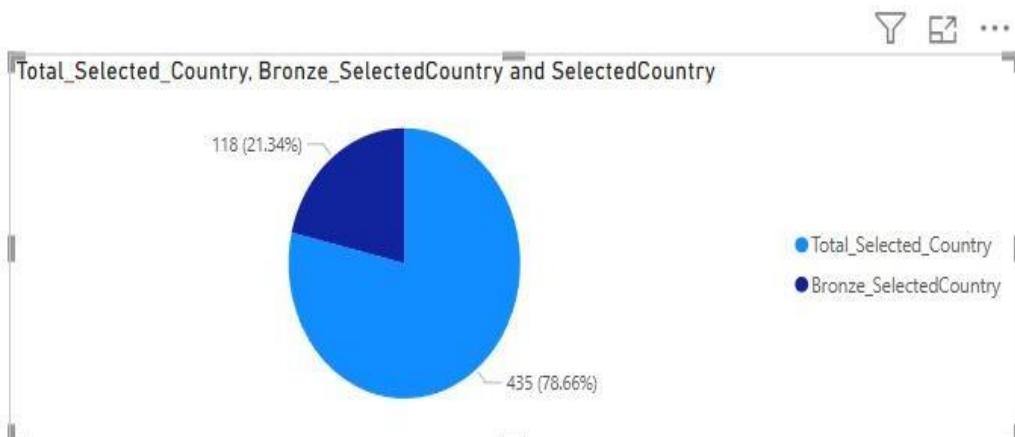
```
Gold_SelectedCountry =  
CALCULATE(  
    SUM('MedalsData'[Gold]),  
    'MedalsData'[Nation1] = SELECTEDVALUE('MedalsData'[Nation1]))
```



a) Bronze Medal:

DAX

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

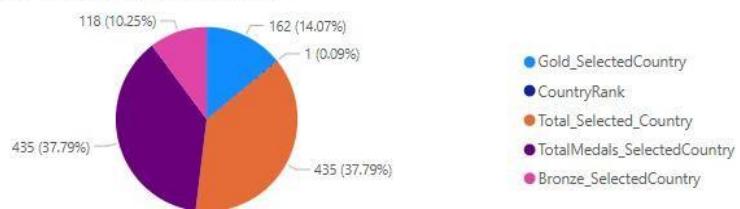


1. Rank Countries Based on Total Medals

DAX

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

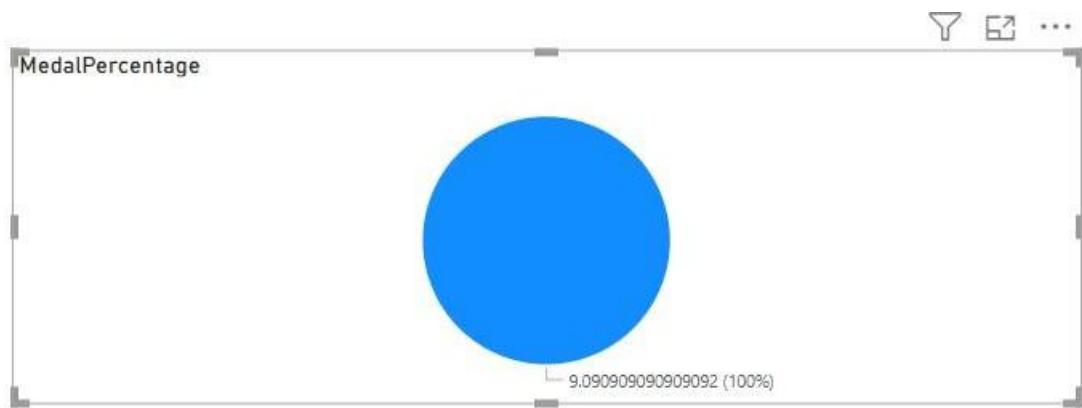
Gold_SelectedCountry, CountryRank, Total_Selected_Country, TotalMedals_SelectedCountry, Bronze_SelectedCountry and SelectedCountry



1. Percentage of Medals Won by the Selected Country

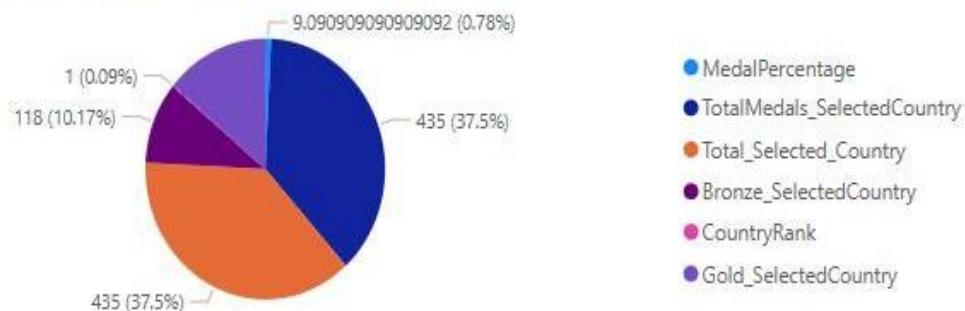
DAX

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

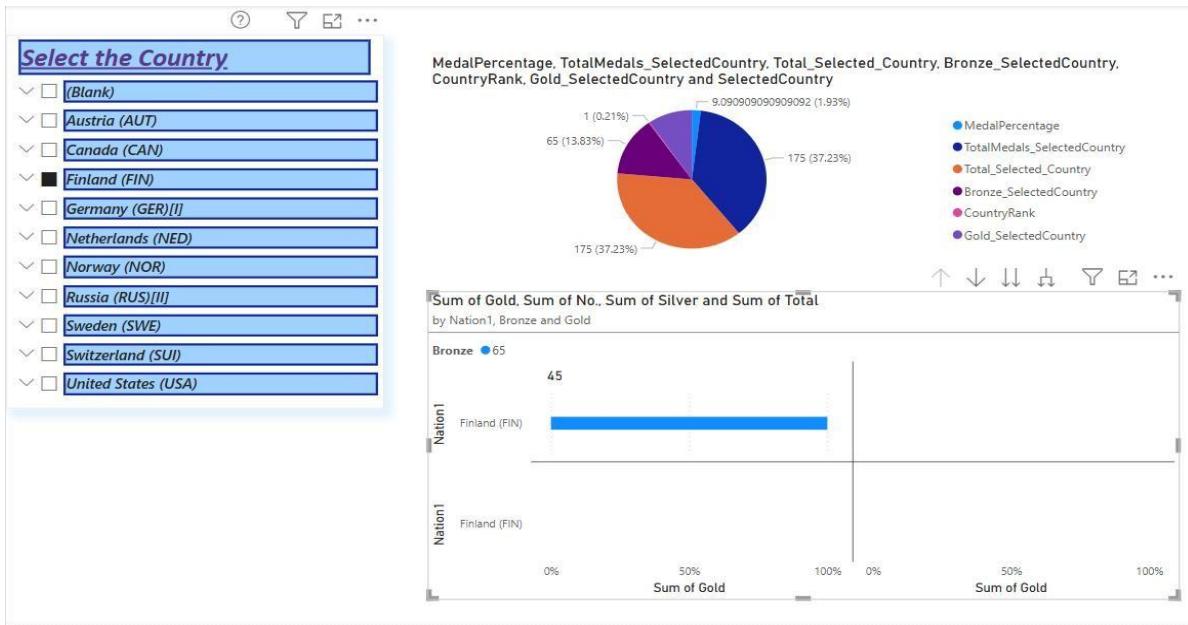


1. Over all DAX Query Visualization:

MedalPercentage, TotalMedals_SelectedCountry, Total_Selected_Country, Bronze_SelectedCountry, CountryRank, Gold_SelectedCountry and SelectedCountry



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

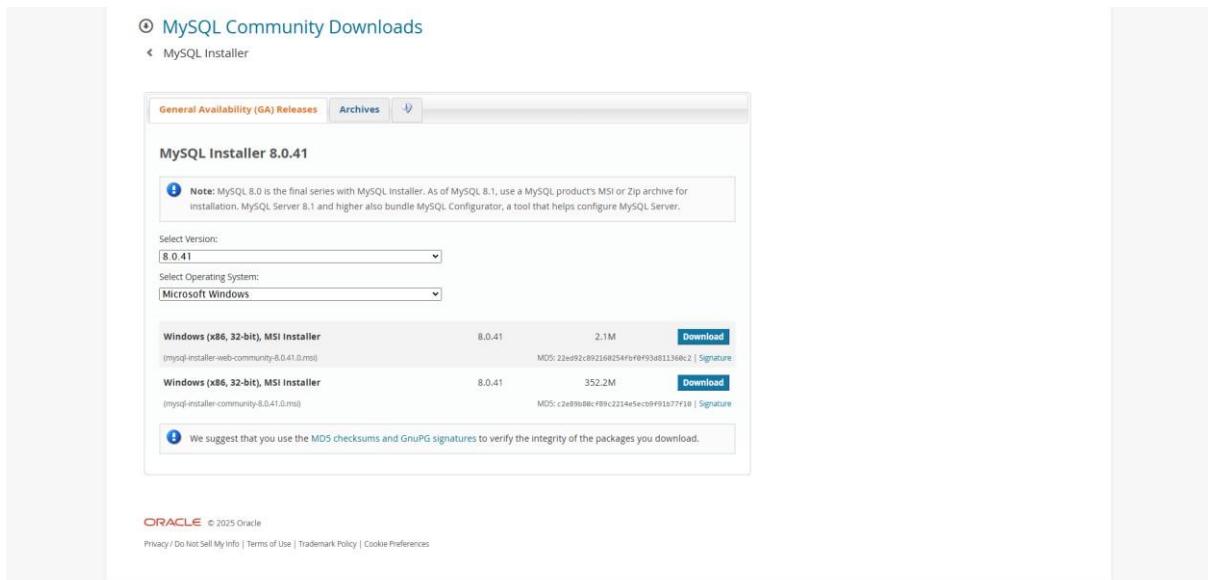
DATE :

IMPORT THE LEGACY DATA FROM SQLSERVER AND LOAD IN THE TARGET SYSTEM

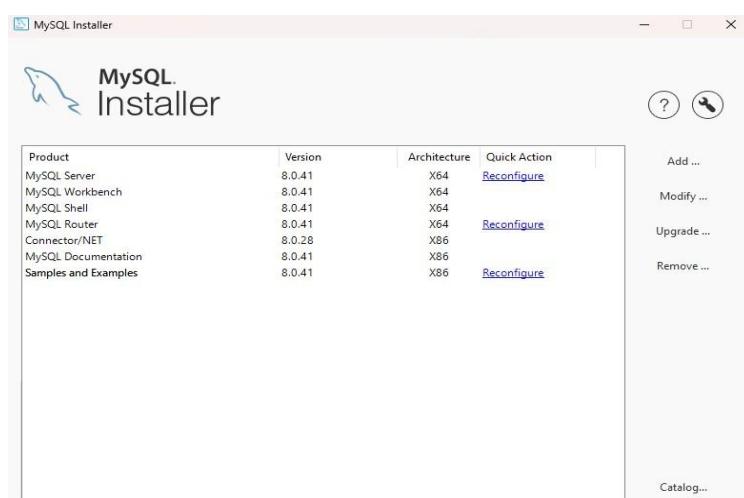
AIM:To import the data from SQLServer and load in the target system.

PROCEDURE:

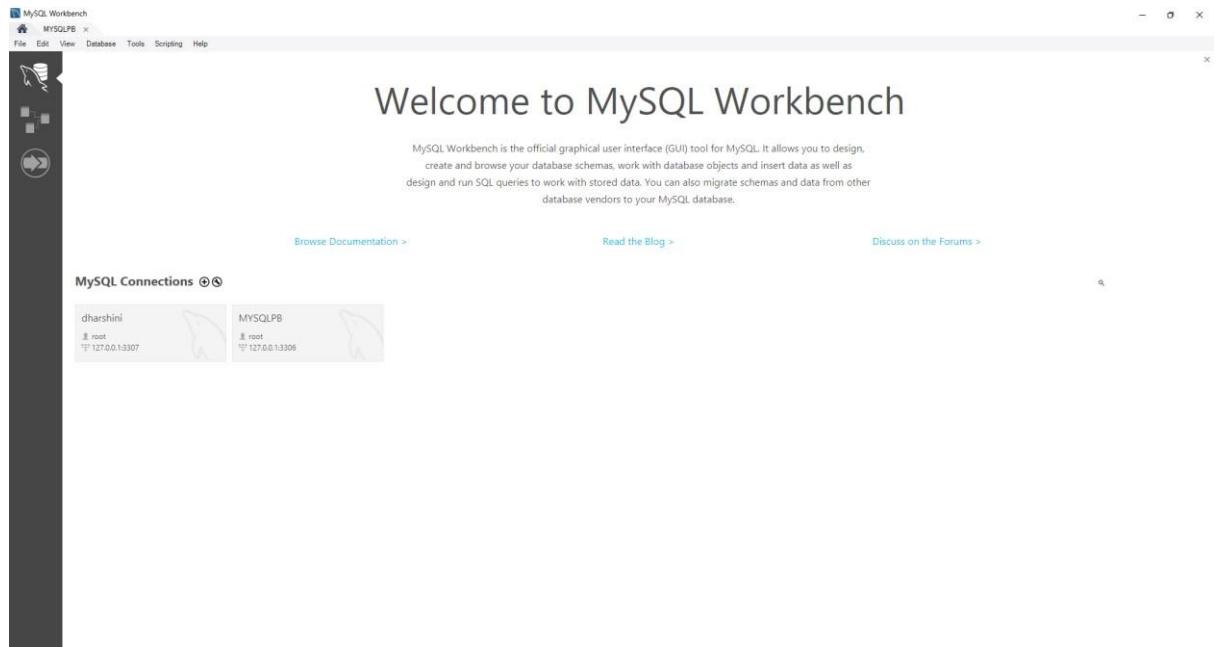
Step 1: Download MYSQL installer from <https://dev.mysql.com/downloads/installer/>.



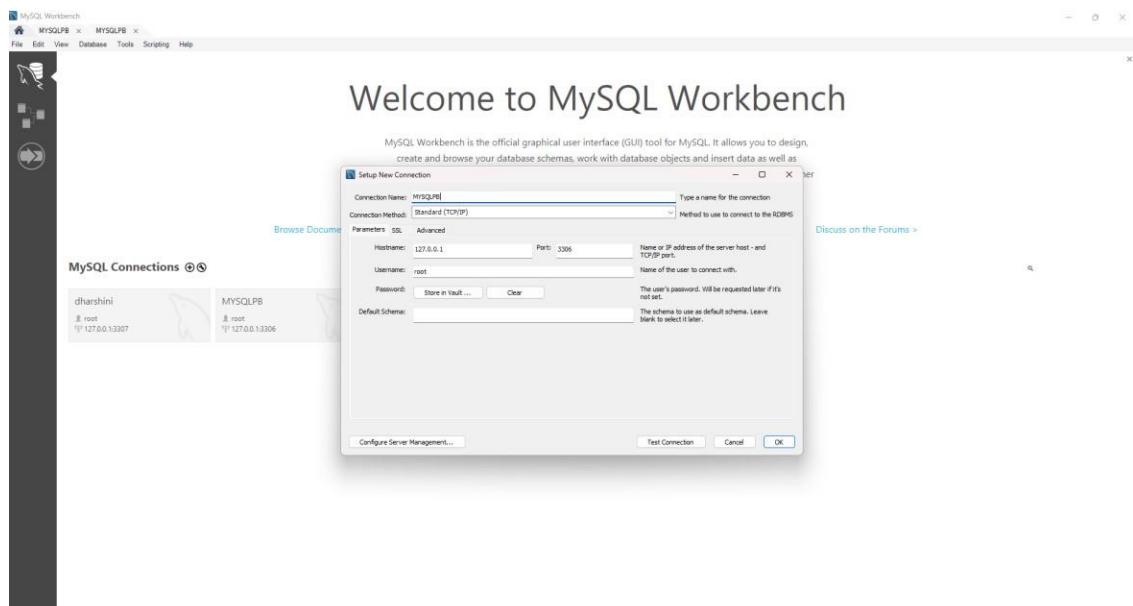
Step 2: Execute MYSQL installer and Add Server Files and configure it by giving server name and password for the database.



Step 3: Open MYSQL Workbench



Step4: Create a MYSQL connection



Step 5: Upload the SQL data into the MYSQL Workbench

A screenshot of the MySQL Workbench application. The main window shows a database schema tree on the left with 'mysqldb' selected. A central pane displays a query editor with the following SQL code:

```
1 * INSERT INTO `user_details` (`user_id`, `username`, `first_name`, `last_name`, `gender`, `password`, `status`) VALUES
2     ('roger885', 'david', 'John', 'Edwards', 'Male', 'edae33ee100007e565d74fecdc200b0', 1)
```

The status bar at the bottom indicates 'No object selected'. On the right, there's a 'File Explorer' window showing files in the 'Downloads' folder, including 'Sample-SQL-File-100-Rows.sql'. A tooltip on the right says: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.' The bottom right corner has 'Context Help' and 'Snippets' buttons.

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

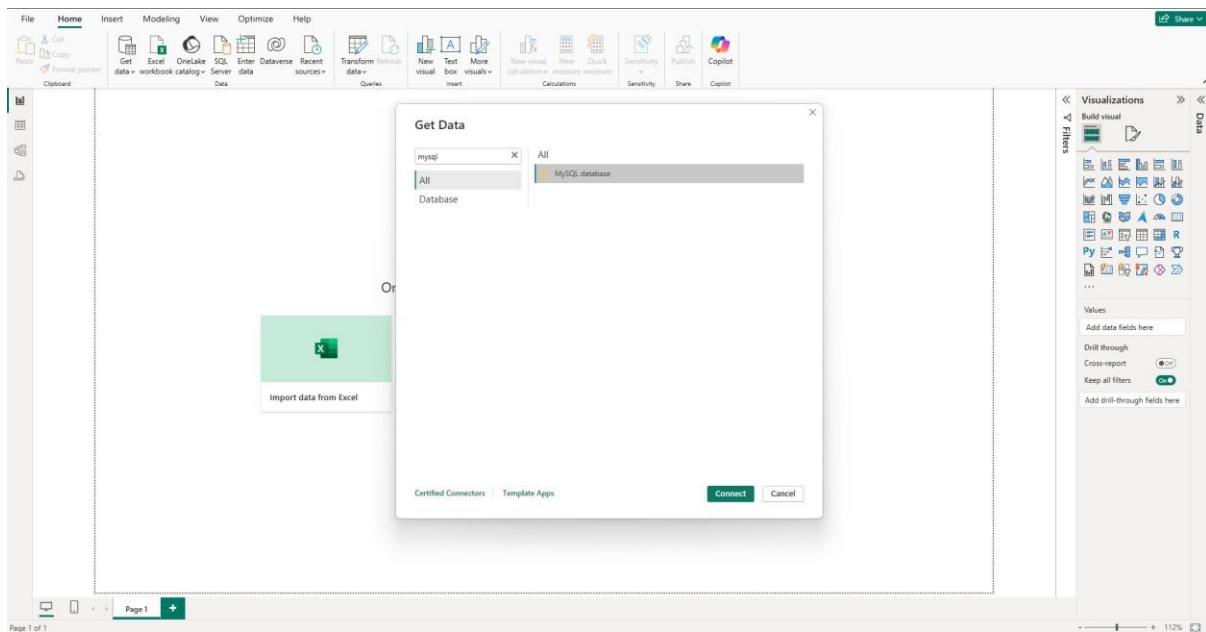
The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help, and a toolbar with various icons. The left sidebar has a Navigator section with SCHEMAS, showing databases like admin, sample, and sample_db. The main area is titled "Query 1" and contains the following SQL code:

```
1 -- Database: 'samplevideo_db'
2 --
3 --
4 --
5 -- -----
6 -- Table structure for table 'user_details'.
7 --
8 -- Table structure for table 'user_details'.
9 --
10 * CREATE DATABASE sample_db;
11
12 * USE sample_db;
13
14 * CREATE TABLE IF NOT EXISTS `user_details` (
15 *   `user_id` int(11) NOT NULL AUTO_INCREMENT,
16 *   `username` varchar(255) DEFAULT NULL,
17 *   `first_name` varchar(50) DEFAULT NULL,
18 *   `last_name` varchar(50) DEFAULT NULL,
19 *   `gender` varchar(10) DEFAULT NULL,
20 *   `password` varchar(100) DEFAULT NULL,
21 *   `status` tinyint(1) DEFAULT NULL,
22 *   PRIMARY KEY (`user_id`)
23 * ) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=10001 ;
24
25 --
26 -- Dumping data for table 'user_details'
27 --
28
29 * INSERT INTO `user_details`(`user_id`, `username`, `first_name`, `last_name`, `gender`, `password`, `status`)
30 VALUES (1, 'roger123', 'david', 'John', 'Female', 'edad3eev1980d7c63d74feedc2c60bd', 1),
```

The results pane at the bottom shows the execution of the commands:

Time	Action	Message	Duration / Fetch
1 10:58:34	CREATE DATABASE sample_db	1 row(s) affected	0.015 sec
2 10:58:34	USE sample_db;	0 rows(s) affected	0.000 sec
3 10:58:34	CREATE TABLE IF NOT EXISTS `user_details` (`user_id` int(11) NOT NULL AUTO_INCREMENT , `username` varchar(255) DEFAULT NULL , `first_n...	0 rows(s) affected, 2 warning(s): 1601 Integer display width is deprecated and will be removed in a future release. 1601 Integer display width is deprecated a...	0.000 sec
4 10:58:40	INSERT INTO `user_details`(`user_id`, `username`, `first_name`, `last_name`, `gender`, `password`, `status`) VALUES (1, 'roger123', 'david', 'John', 'Female', 'edad3eev1980d7c63d74feedc2c60bd', 1)	100 rows(s) affected, 0 warning(s), 0 errors(s)	0.000 sec
5 10:58:40	USE sample_db;	0 rows(s) affected	0.000 sec

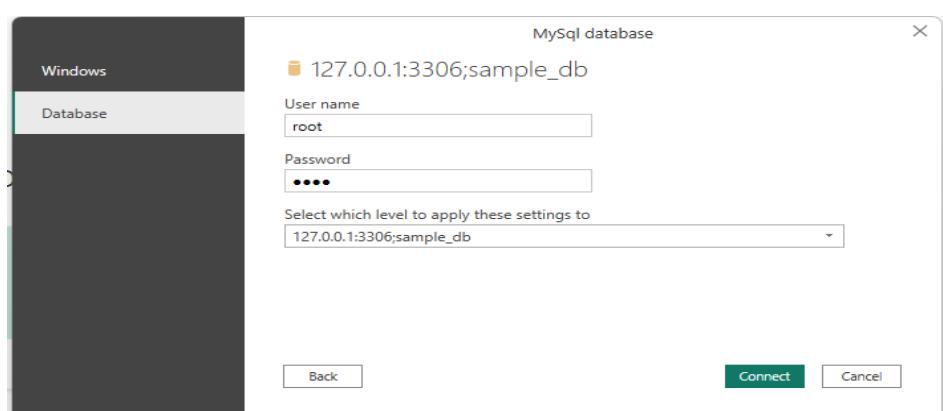
Step 7: Open PowerBi and Import the Database to PowerBi



Step 8: Enter your Server and Database



Step 9: Enter your Username and Password.



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

The screenshot shows the Navigator tool interface. On the left, there's a sidebar with 'Display Options' and a tree view showing '127.0.0.1:3306: sample_db [1]' and 'sample_db.user_details'. The main area displays the 'sample_db.user_details' table with the following schema:

user_id	username	first_name	last_name	gender	password
1	rogers63	david	john	Female	e6a33
2	mike28	rogers	paul	Male	2e7dc
3	rivera92	david	john	Male	1c3a8
4	ross95	maria	sanders	Male	62f0a1
5	paul85	morris	miller	Female	61bd0
6	smith34	daniel	michael	Female	7055b
7	james84	sanders	paul	Female	b7f72
8	daniel53	mark	mike	Male	299cb
9	brooks80	morgan	maria	Female	aa736
10	morgan65	paul	miller	Female	a28dc
11	sanders84	david	miller	Female	0629e
12	maria40	chrishaydon	bell	Female	17f28l
13	brown71	michael	brown	Male	fa0c4f
14	james63	morgan	james	Male	b9454
15	jenny0993	rogers	chrishaydon	Female	38882
16	john96	morgan	wright	Male	d0bb9
17	miller64	morgan	wright	Male	58b20
18	mark46	david	ross	Female	21cdcl
19	jenny0988	maria	morgan	Female	ec9ed
20	mark80	mike	bell	Male	08448
21	morris72	miller	michael	Male	bdb04
22	wright39	ross	rogers	Female	1b685
23	paul68	brooks	mike	Male	12d83

At the bottom, there are buttons for 'Select Related Tables', 'Load', 'Transform Data', and 'Cancel'.

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

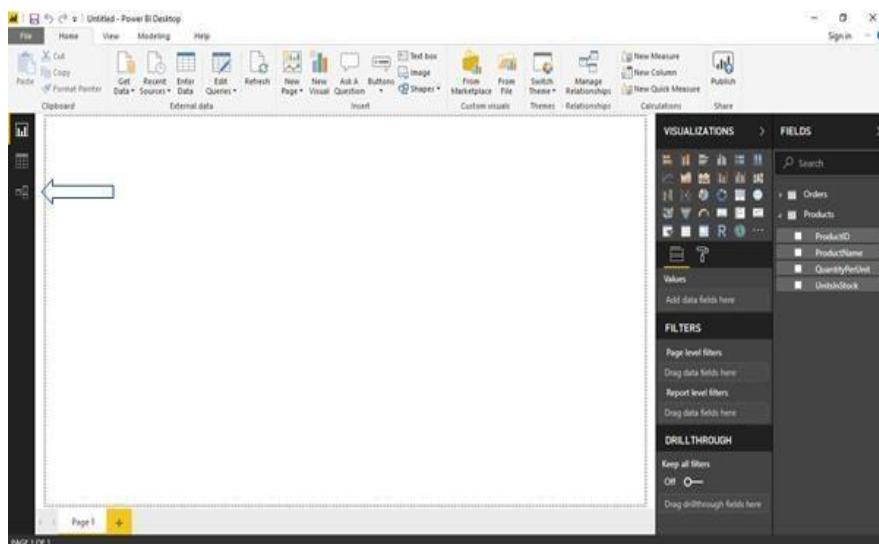
Date:

AIM:

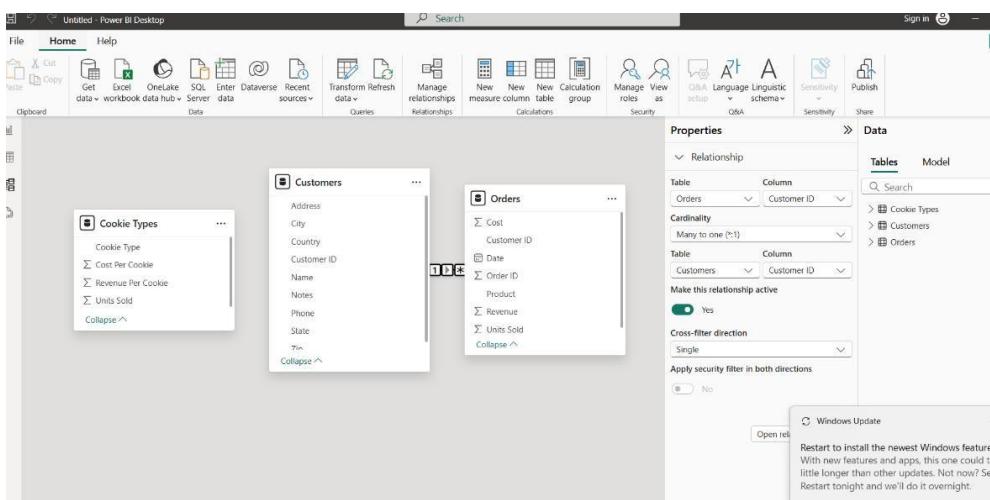
To perform the Extraction Transformation and Loading (ETL) process to construct the database in the SQL Server / Power BI.

PROCEDURE:

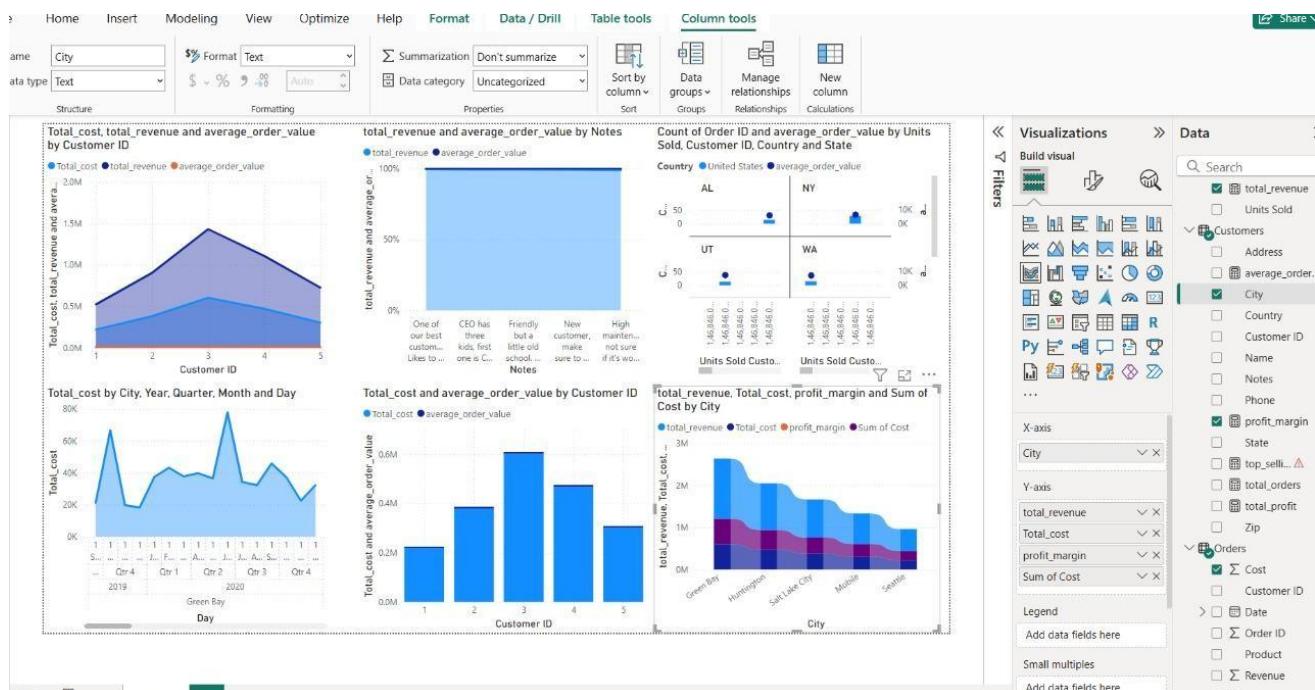
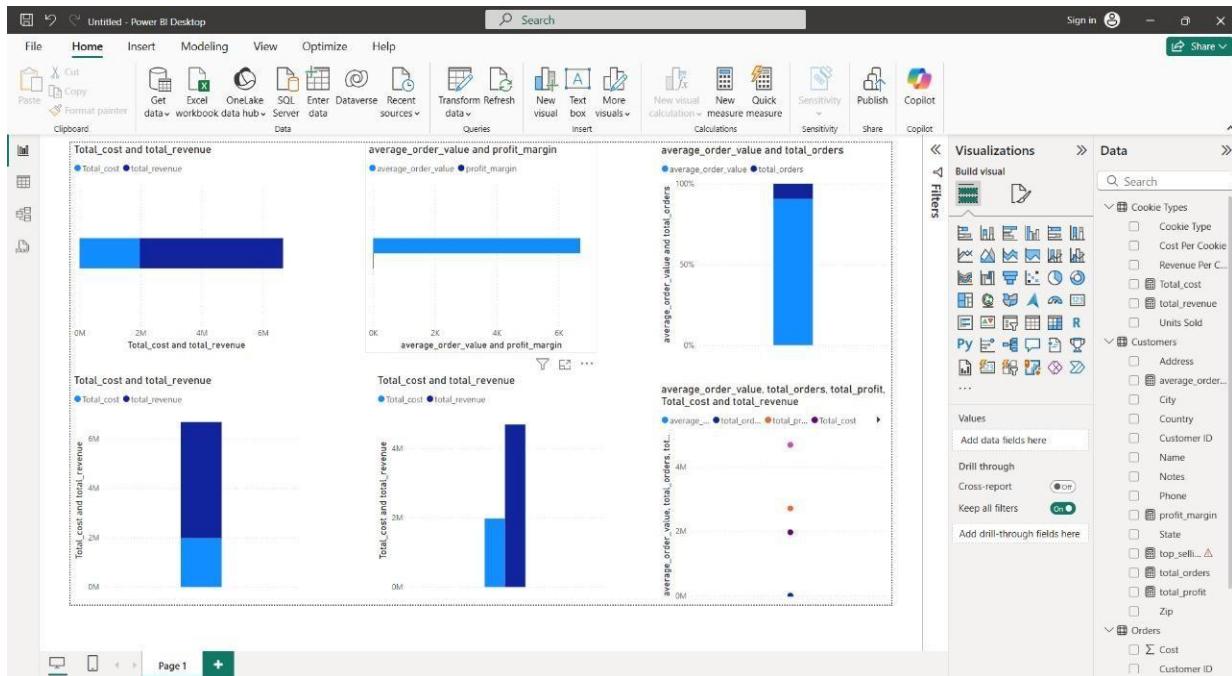
Step 1: Once the data is loaded, select the Model view and create relationship.

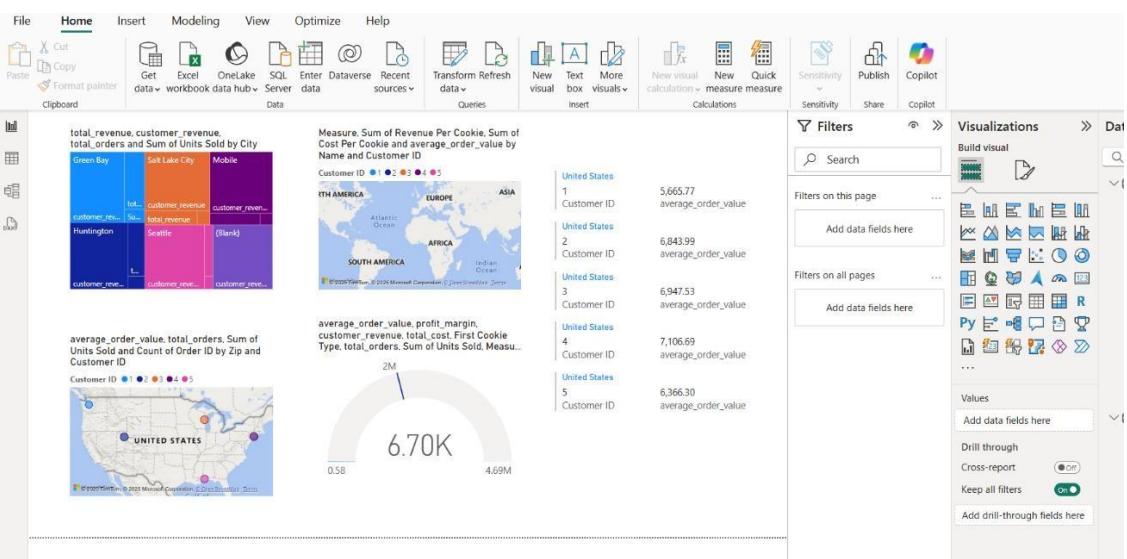
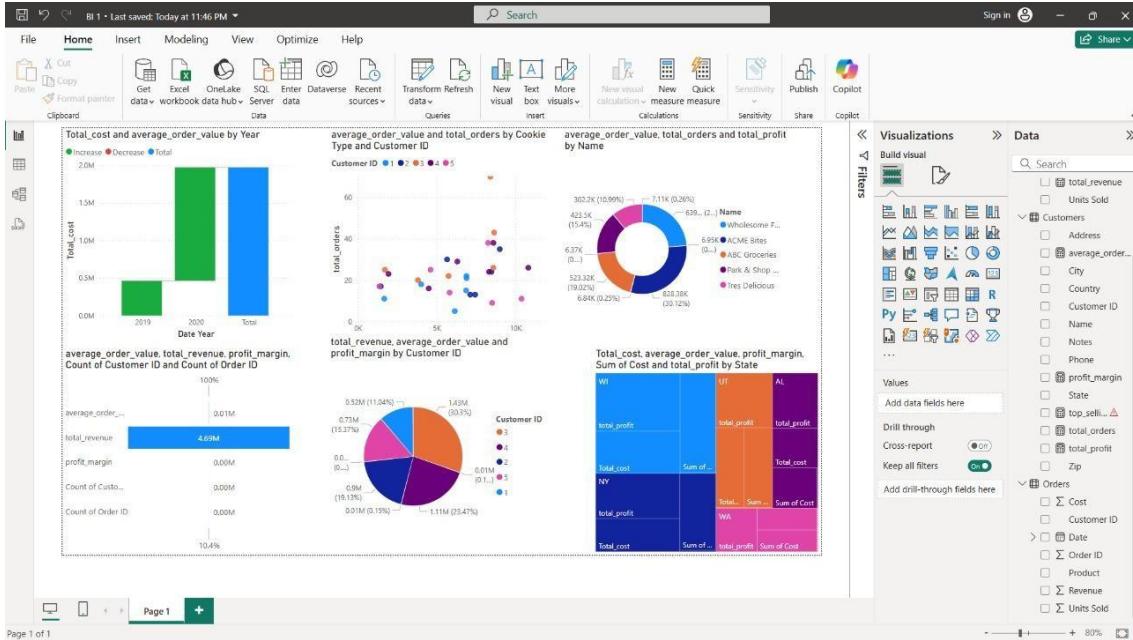


Step 2: We see the following, which visualizes the relationship between the queries.



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





File Home Insert Modeling View Optimize Help

Clipboard

Data

Queries

Insert Calculations

Sensitivity Share Copilot

Visualizations

Filters

Customer ID Sum of Cost Year Quarter

3	21193	2019	Qtr 3
3	66669	2019	Qtr 4
3	19801	2019	Qtr 4
3	1851	2020	Qtr 1
3	37383	2020	Qtr 1
3	43202	2020	Qtr 1
3	37779	2020	Qtr 1
3	39751	2020	Qtr 2
3	36481	2020	Qtr 2
3	77844	2020	Qtr 2
Total	602812		

Sum of Revenue Per Cookie.

total_revenue and Sum of Units Sold by Cookie Type

(Blank)

Notes
One of our best customers! Likes to talk about sports.

Country

Cookie Type

United States average

Chocolate Chip
Fortune Cookie
Oatmeal Raisin
Snickerdoodle
Sugar
White Chocolate Macadamia Nut
Total

File Home Insert Modeling View Optimize Help

Clipboard

Data

Queries

Insert Calculations

Sensitivity Share Copilot

Visualizations

Filters

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

4	3	2
5	1	--

Choose a narrative type

Use Copilot to create a narrative with AI, or choose Custom for more control. [Read Copilot terms](#)

Copilot (preview)

Custom

Sum of Revenue

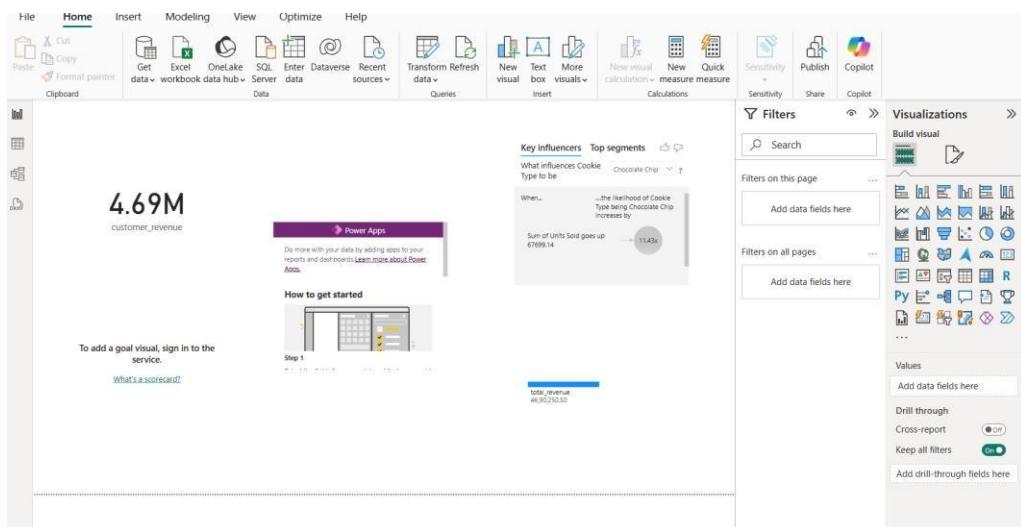
4.6...

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).

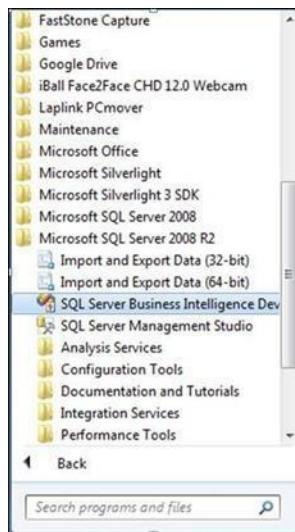
1. Open SQL Server Management Studio 2008
2. Connect Database Engine
3. Open **New Query** editor
4. Copy paste Scripts given below in various steps in new query editor window one by one
5. To run the given SQL Script, press **F5**
6. It will create and populate “Sales_DW” database on your SQL Server

Developing an OLAP Cube:

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

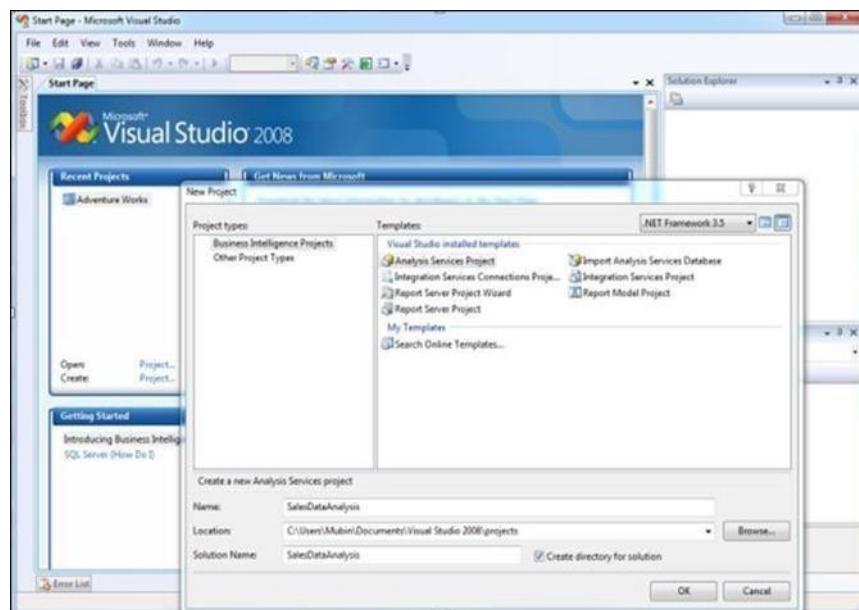
Step 1: Start BIDS Environment

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



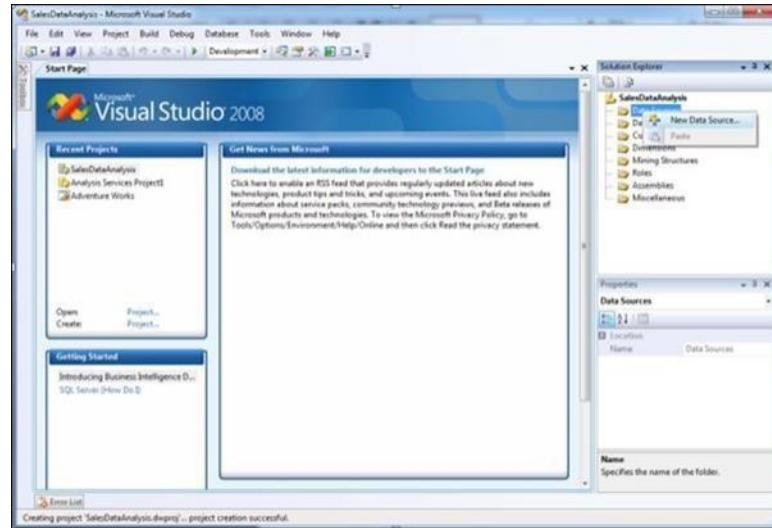
Step 2: Start Analysis Services Project

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



Step 3: Creating New Data Source

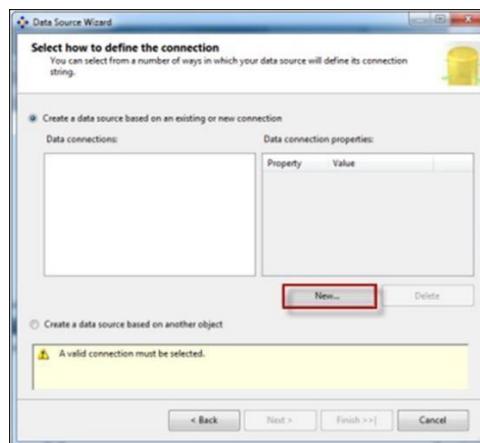
- In Solution Explorer, Right click on **Data Source** -> Click **New Data Source**



- Click on **Next**

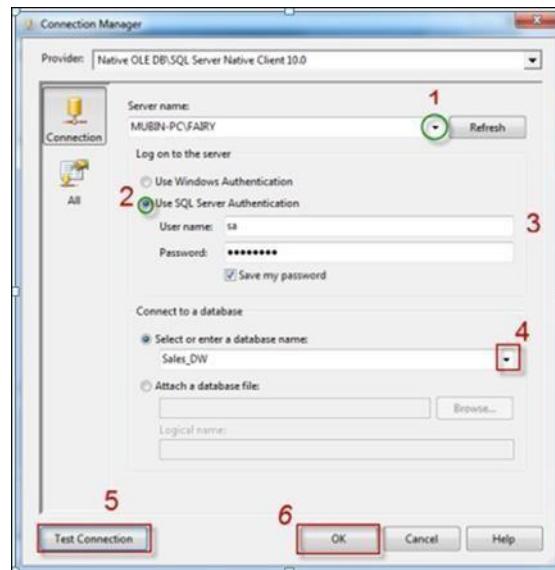


- Click on **New** Button

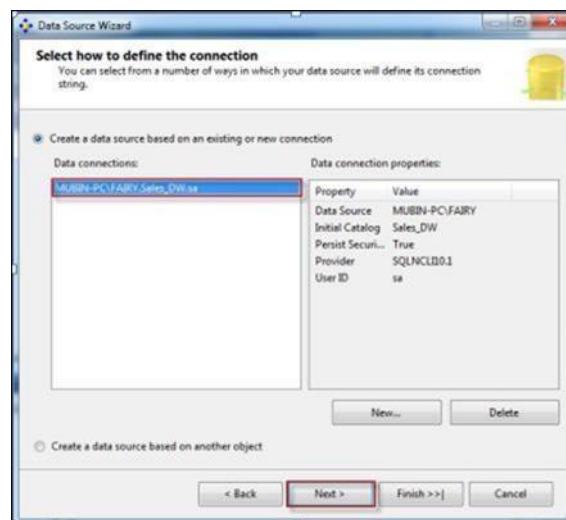


- Creating **New connection**

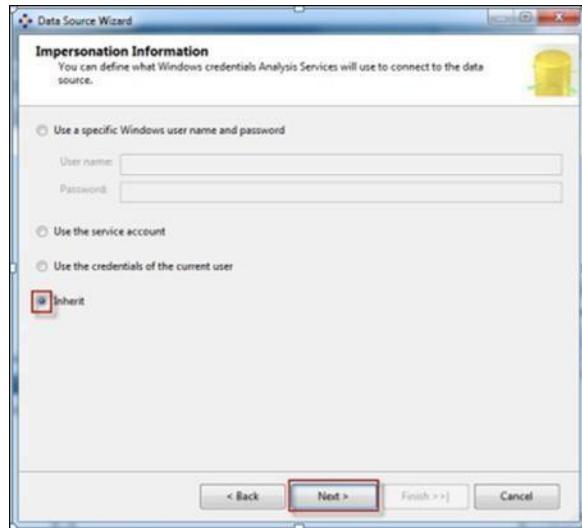
1. Specify Your **SQL Server Name** where your Data Warehouse was created
2. Select Radio Button according to your **SQL Server Authentication** mode
3. Specify your **Credentials** using which you can connect to your SQL Server
4. Select database Sales_DW.
5. Click on **Test Connection** and verify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



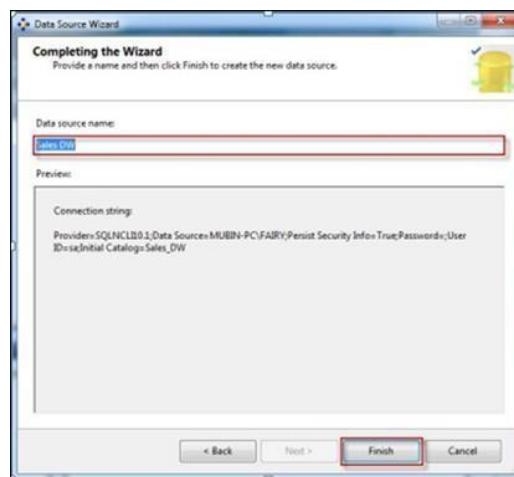
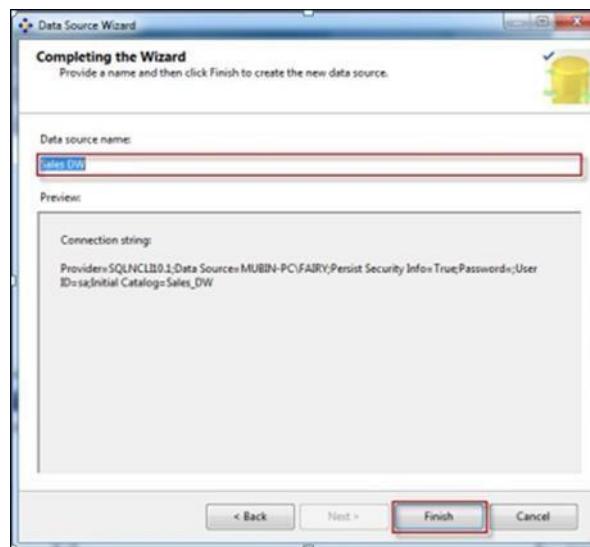
- Select Connection created in **Data Connections**-> Click **Next**



- Select Option **Inherit**

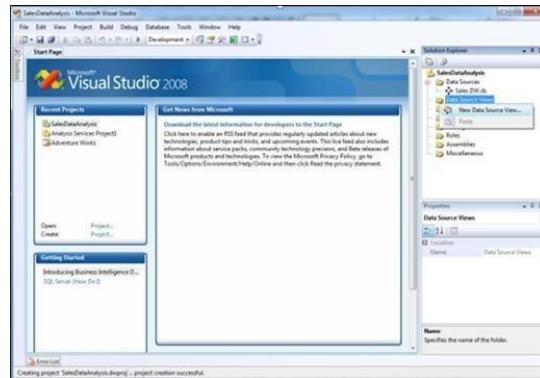


- Assign Data Source Name -> Click Finish



Step 4: Creating New Data Source View

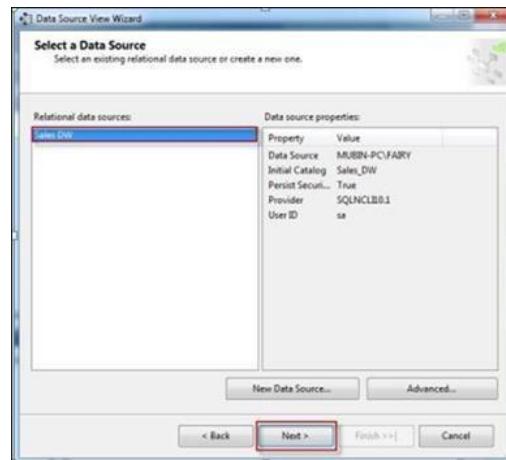
- In the Solution Explorer, Right Click on **Data Source View** -> Click on **New Data Source View**



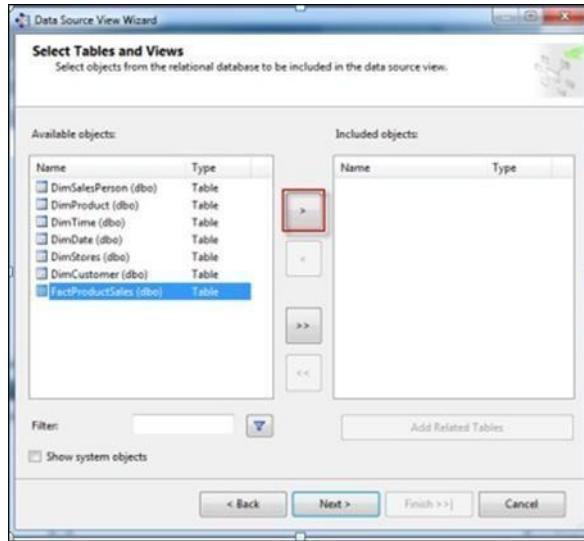
- Click **Next**



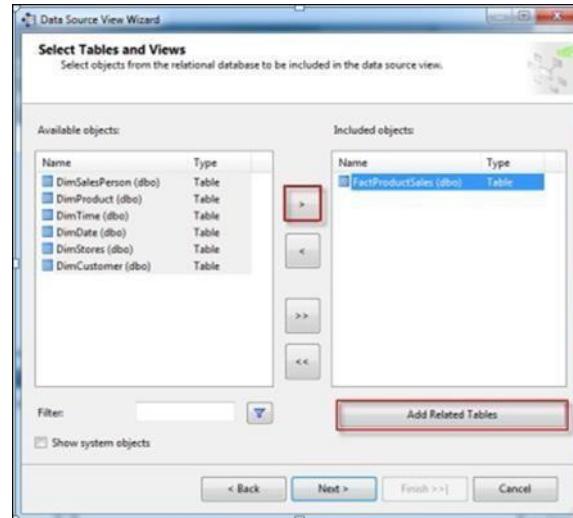
- Select **Relational Data Source** we have created previously (Sales_DW)-> Click **Next**



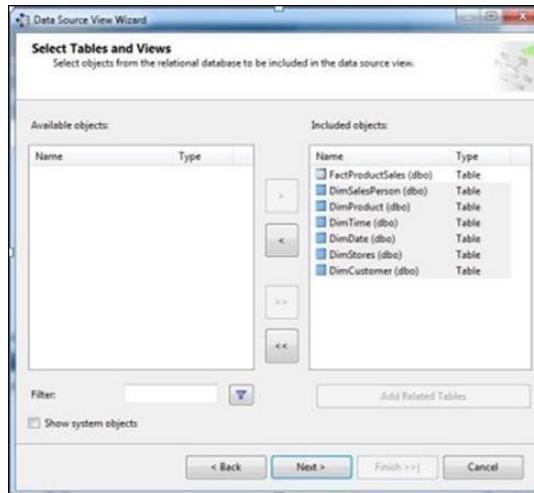
- First move your **Fact Table** to the right side to include in object list.



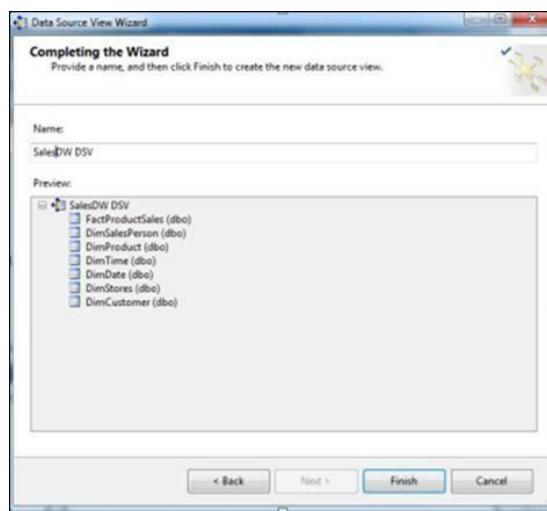
- Select FactProductSales Table -> Click on Arrow Button to move the selected object to Right Pane.
- Now to **add dimensions** which are **related** to your **Fact Table**, follow the given steps:
Select **Fact Table** in Right Pane (Fact product Sales) -> Click On **Add Related Tables**



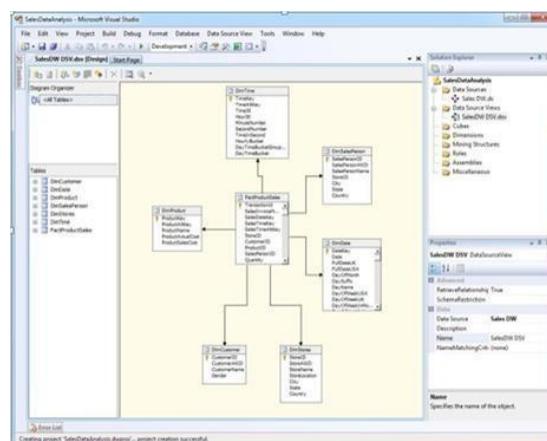
- It will add all associated dimensions to your Fact table as per relationship specified in your SQL DW (Sales_DW).
- Click **Next**.



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

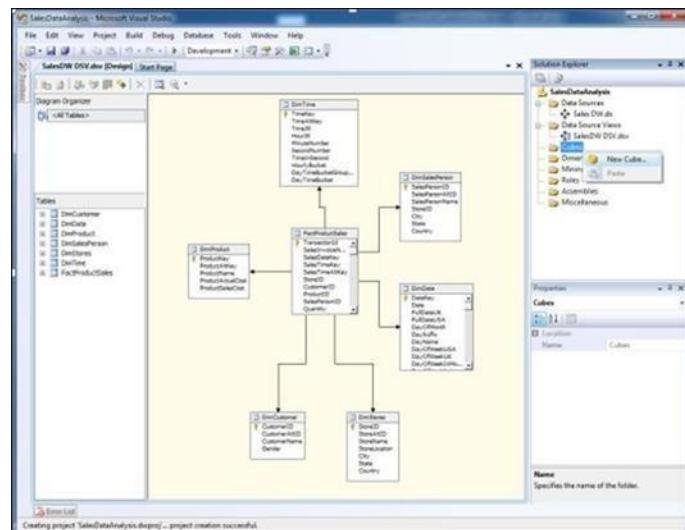


- Now Data Source View is ready to use.



Step 5: Creating New Cube

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



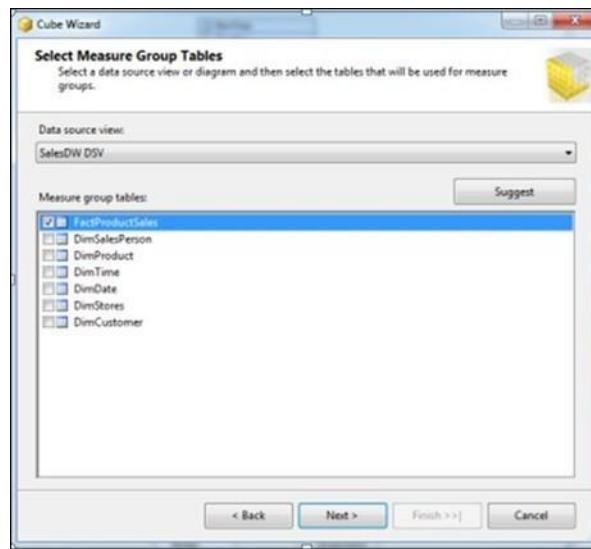
- Click **Next**



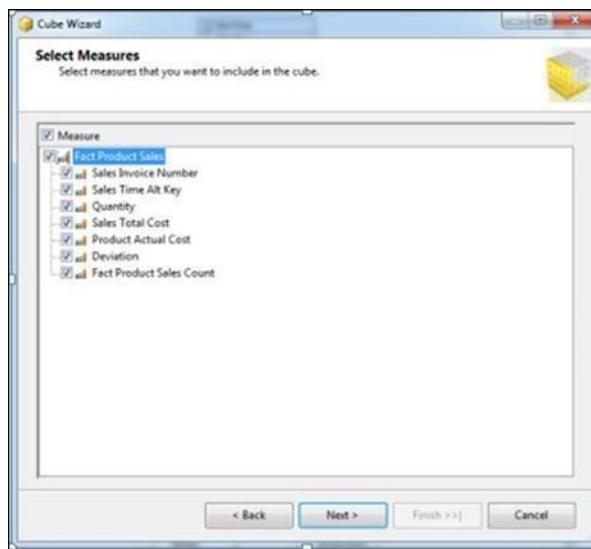
- Select Option **Use existing Tables** -> Click **Next**



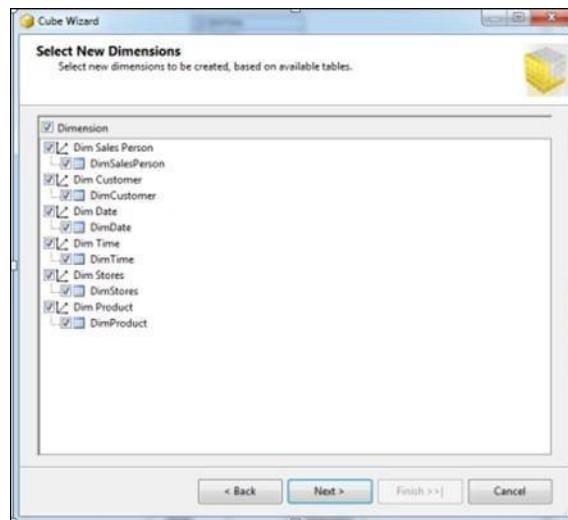
- Select Fact Table Name from **Measure Group Tables (FactProductSales)** -> Click **Next**



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



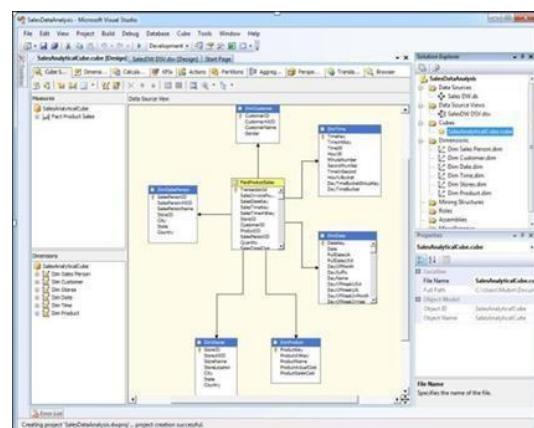
- Select All **Dimensions** here which are associated with your Fact Table-> Click **Next**



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

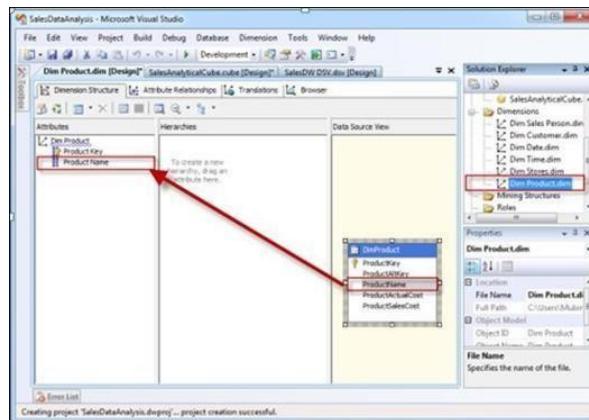


- Now your Cube is ready, you can see the newly created cube and dimensions added in your solution explorer.



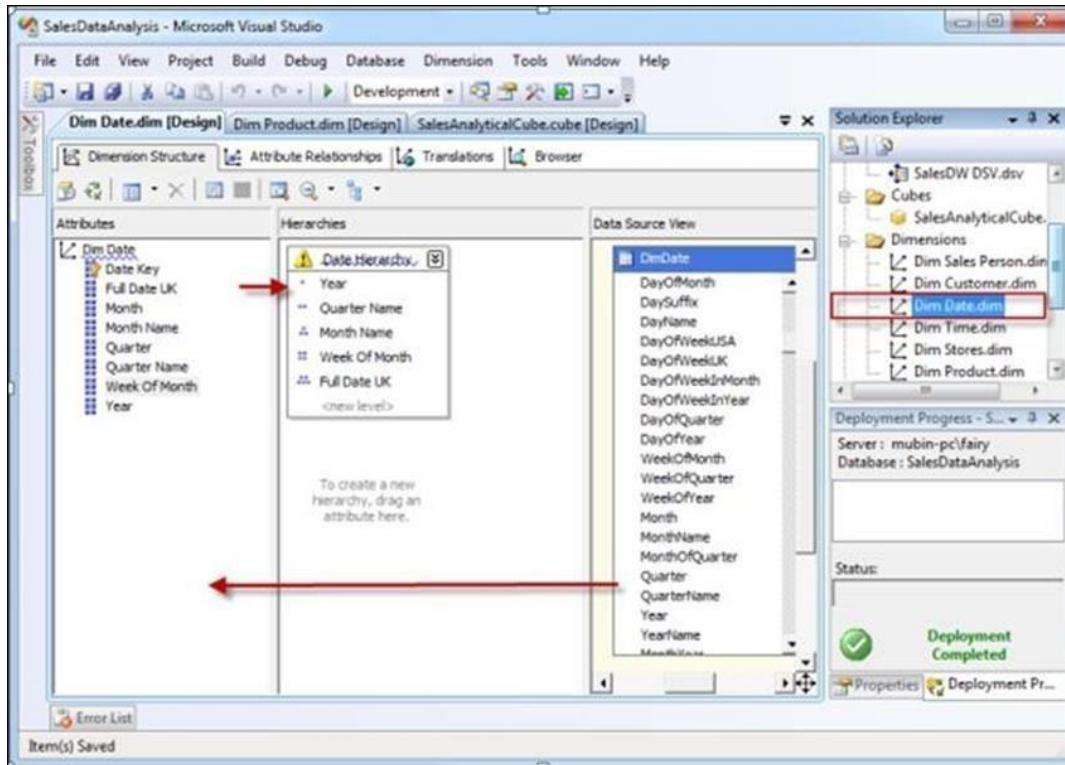
Step 6: Dimension Modification

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.



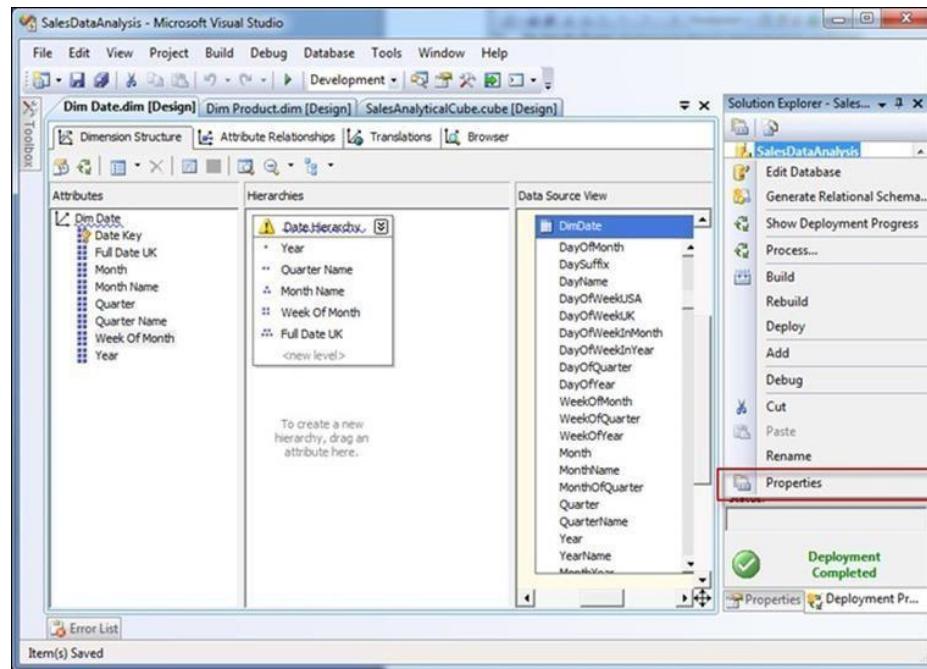
Step 7: Creating Attribute Hierarchy in Date Dimension

- 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)

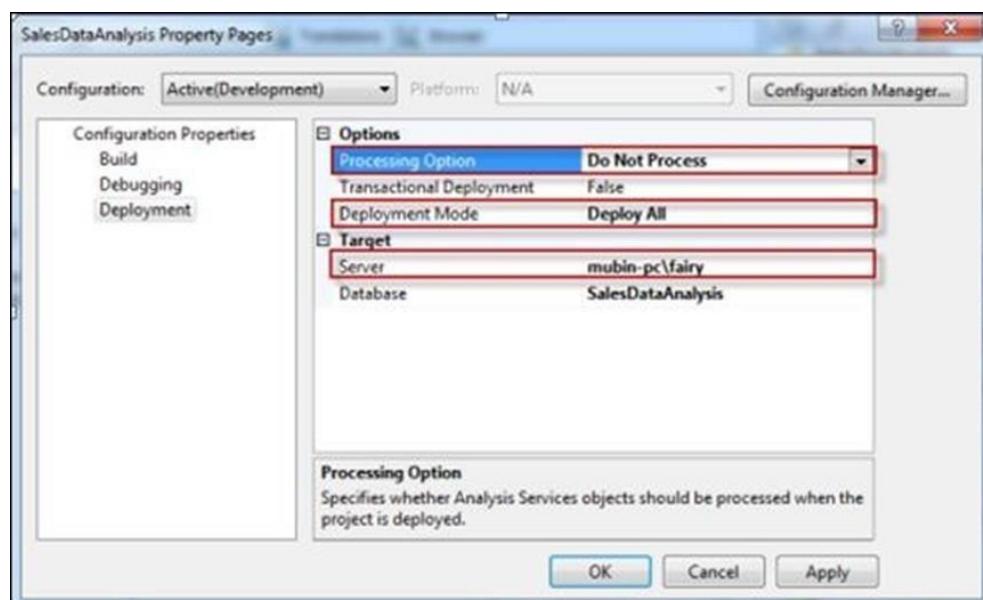


Step 8: Deploy the Cube

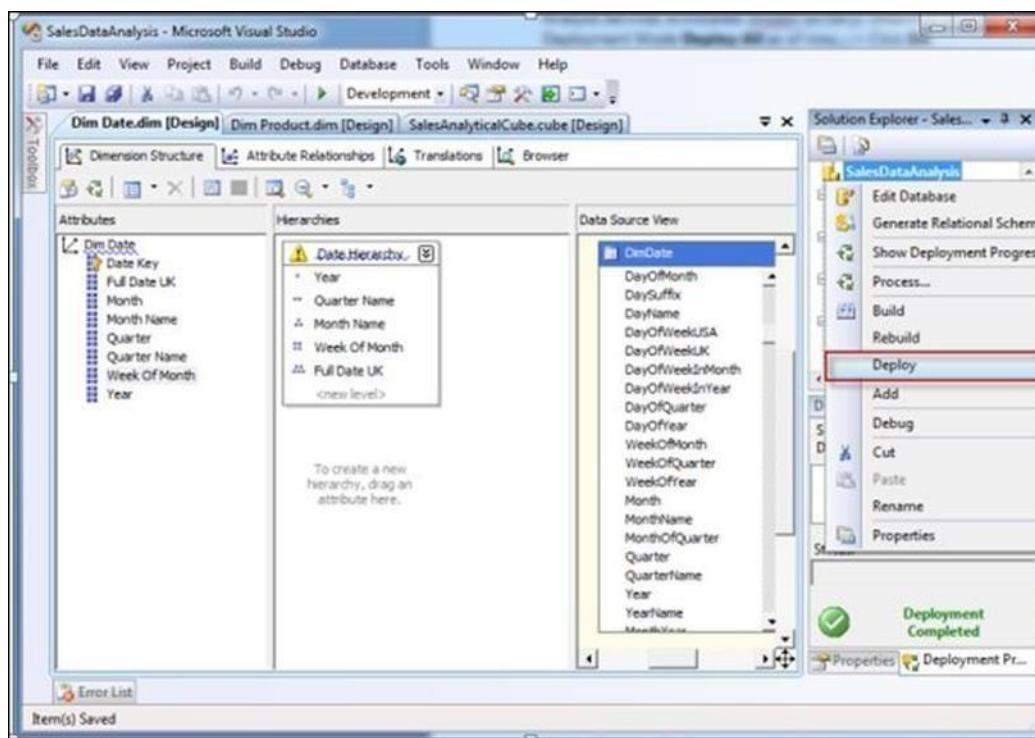
- In Solution Explorer, right click on Project Name (SalesDataAnalysis) --> Click Properties



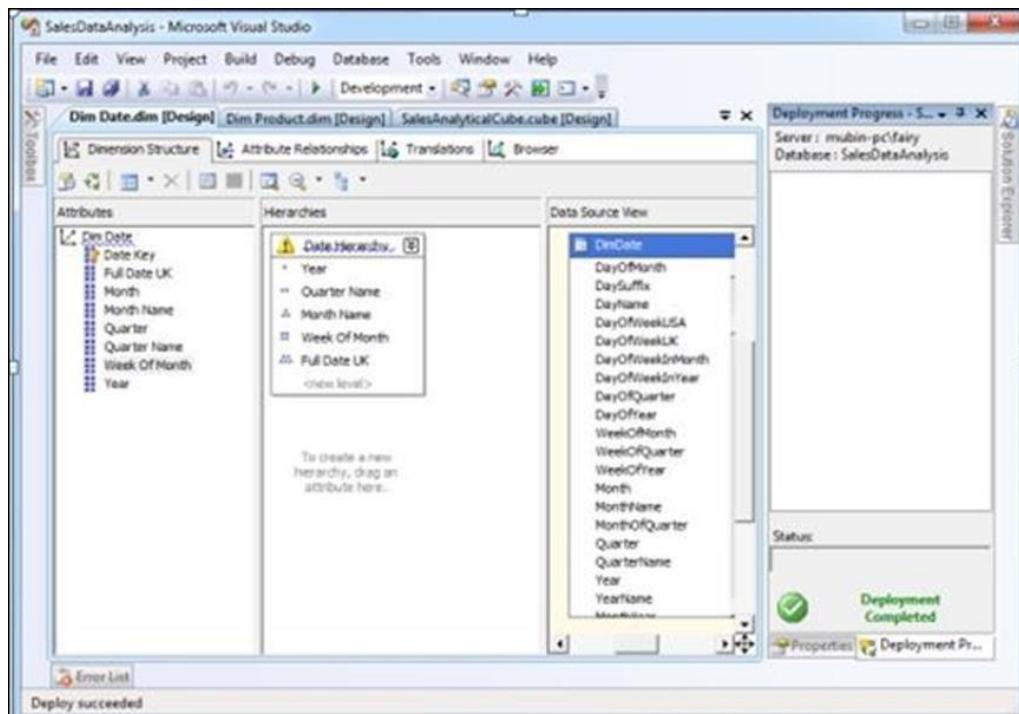
- Set Deployment Properties First
- In Configuration Properties, Select Deployment-> Assign Your SQL Server Instance Name Where Analysis Services Is Installed (mubin-pc\fairy) (Machine Name\Instance Name) -> Choose Deployment Mode Deploy All as of now ->Select Processing Option Do Not Process -> Click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



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

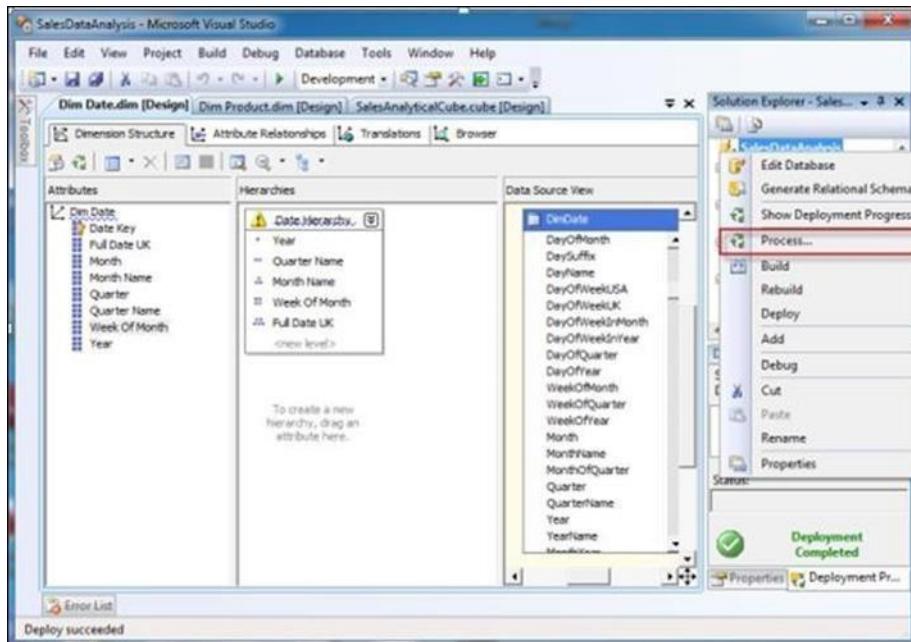


- Once Deployment will finish, you can see the message **Deployment Completed** in deployment Properties.

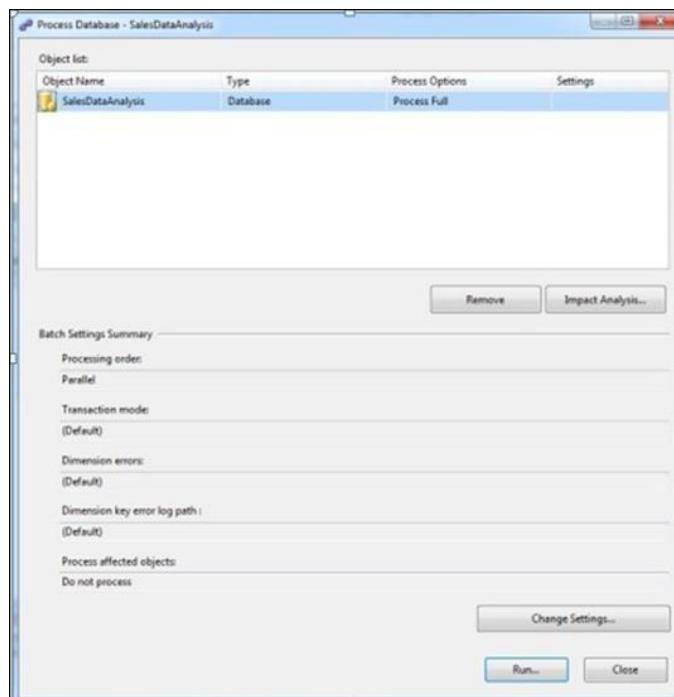


Step 9: Process the Cube

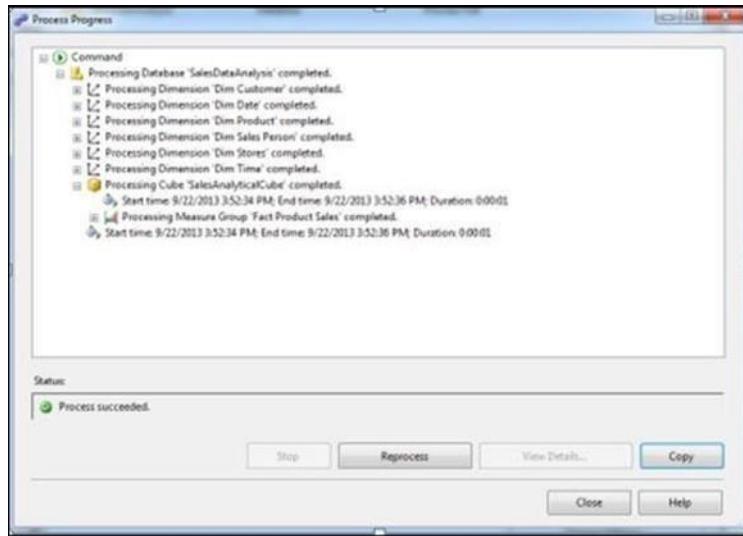
- In Solution Explorer, right click on Project Name (SalesDataAnalysis) --> Click **Process**



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

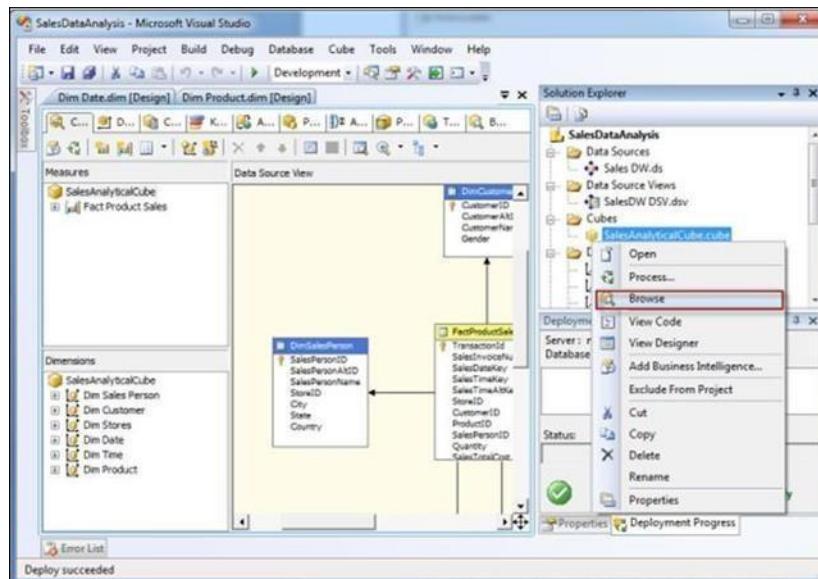


- Once processing is complete, you can see **Status** as **Process Succeeded** -->Click **Close** to close both the open windows for processing one after the other.



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

	Amul Heering Product	Big Name Soap	Rice Granary	SunFlower Oil Ltr	Wheat Flax	Grand Total
Full Date UK	2013-01-01	4	4	2	18	18
	2013-01-02	4	4	2	18	18
	2013-01-03	4	4	2	18	18
	2013-01-04	4	4	2	18	18
	GrandTotal	12	12	12	54	54

RESULT:

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

EX.NO: 5	
DATE:	

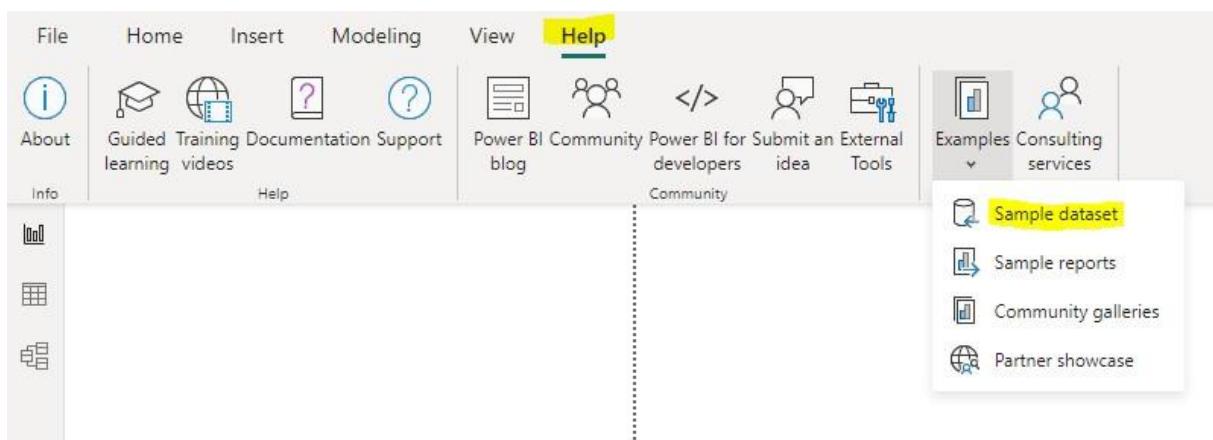
APPLY THE WHAT – IF ANALYSIS FOR DATA VISUALIZATION TO DESIGN AND GENERATE NECESSARY REPORTS BASED ON THE DATA WAREHOUSE DATA

AIM:

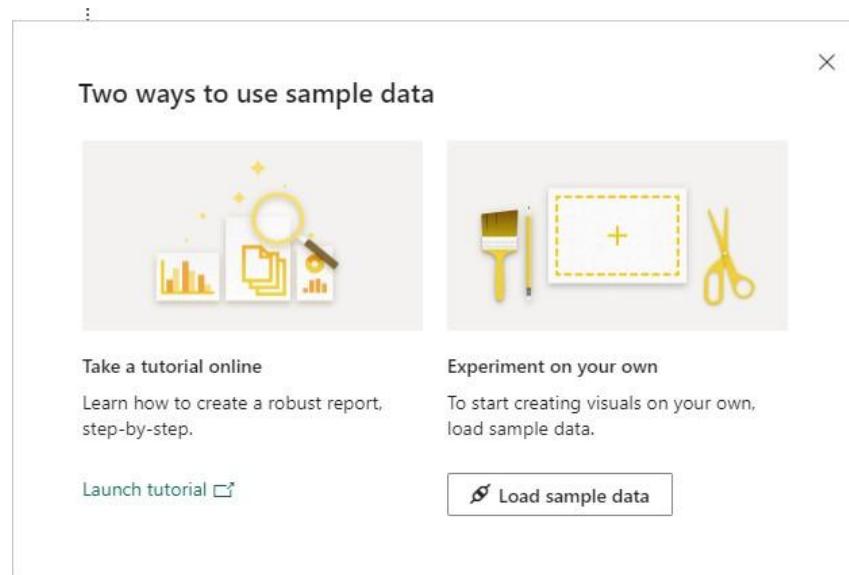
To apply the what – If analysis for Data Visualization to design and generate necessary reports based on the data warehouse data in Excel.

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.

Navigator

The screenshot shows the Power BI Navigator interface. On the left, there is a tree view of the 'Financial Sample.xlsx' file, with the 'financials' sheet selected. The main area displays a preview of the 'financials' data table. The table has four columns: Segment, Country, Product, and Discount Band. The data includes various market segments like Government, Midmarket, and Enterprise, along with countries like Canada, Germany, France, Mexico, and the United States. Products listed are Carretera, Montana, Paseo, and None. The discount band is also listed. At the bottom right of the preview area, there are 'Load', 'Transform Data', and 'Cancel' buttons. The 'Load' button is highlighted in green.

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.

The screenshot shows the Power BI ribbon with the 'Modeling' tab selected. In the 'Modeling' tab group, there is a 'New parameter' button. A tooltip for this button is displayed, showing the options: 'New parameter', 'Manage roles', 'View as', 'Q&A setup', 'Language schema', and 'Linguistic Q&A'. The 'Numeric range' option is highlighted with a yellow box. Below the tooltip, there is a 'Fields' section.

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.

Parameters

X

Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. [Learn more](#)

What will your variable adjust?

Name

Data type

Minimum

Maximum

Increment

Default



Add slicer to this page

Create

Cancel

After creating the parameter, we will get the new table ‘Parameter’ on “Fields”.

The screenshot shows the 'Fields' pane in Power BI. At the top, there is a search bar with a magnifying glass icon. Below it, a tree view shows the hierarchy: 'financials' > 'Parameter'. Under 'Parameter', there are two items: 'Parameter' (with a checked checkbox) and 'Parameter Value' (with an unchecked checkbox). The 'Parameter' item is highlighted with a gray background.

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.”**

The screenshot shows the 'Format visual' dialog for a Slicer. At the top, there are three icons: a grid, a paintbrush, and a magnifying glass. Below them is a search bar. The 'Visual' tab is selected. In the 'Slicer settings' section, under 'Options', the 'Style' dropdown is set to 'Vertical list'. Other sections include 'Selection', 'Reset to default', 'Slicer header' (which is turned 'On'), and 'Values'.

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.

X-axis

Segment

Column y-axis

Sum of Sales

Line y-axis

Add data fields here

Column legend

Add data fields here

Small multiples

Country

Tooltips

Add data fields here

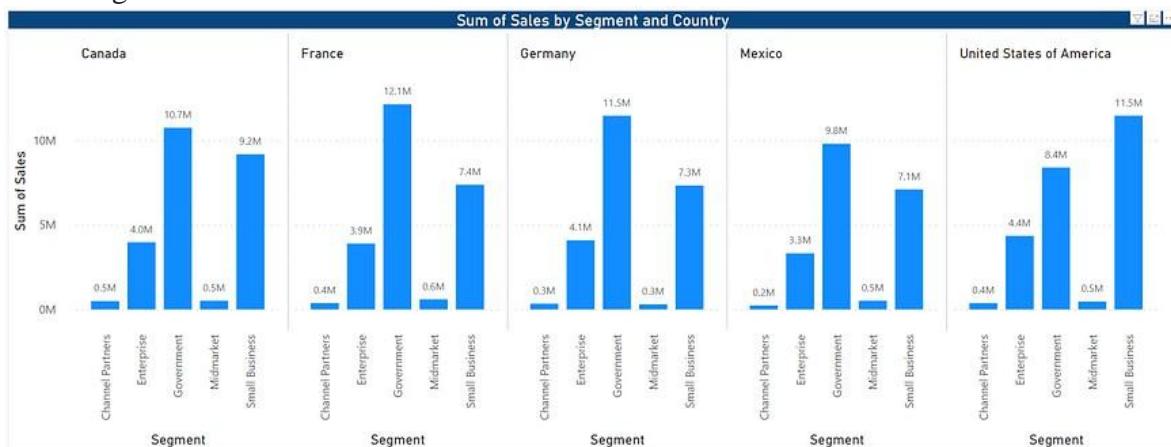
Drill through

Cross-report Off

Keep all filters

Add drill-through fields here

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]
```

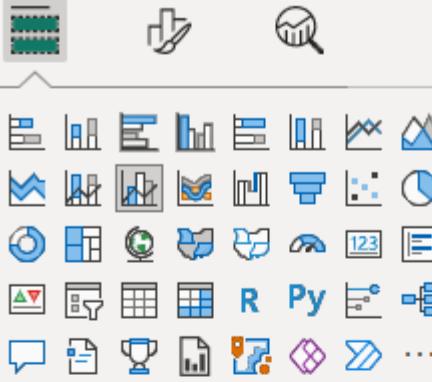
Step 3: Visual for what-if analysis.

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

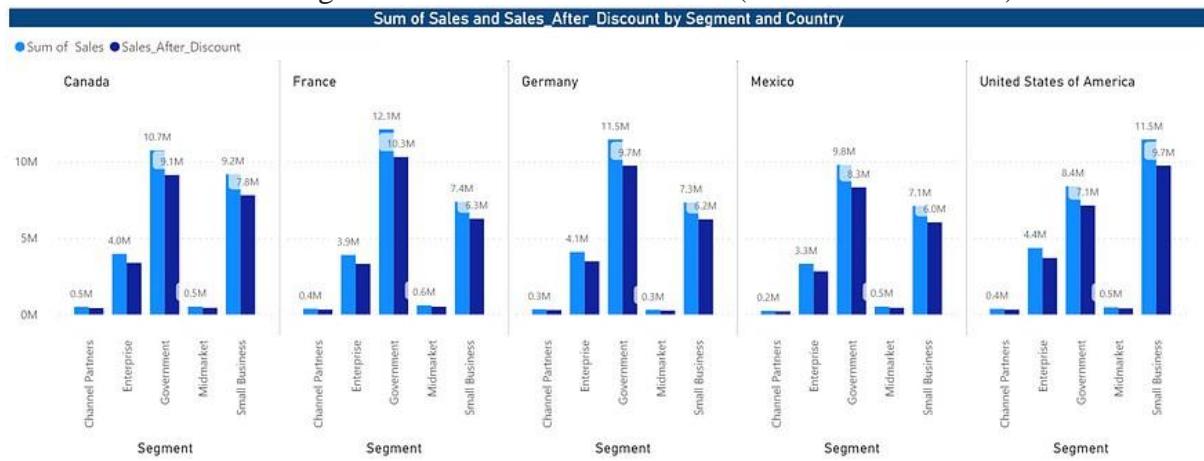
Drill through

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).

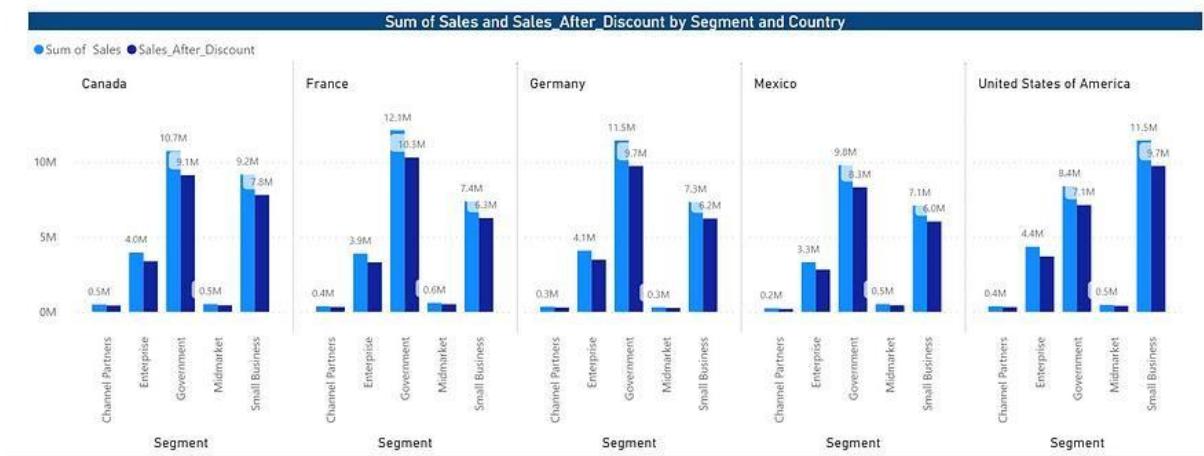


- Finally, we get the report, which includes information about each country, segment, and discount percentage.

% discount

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
Channel Partners	491,164.14	417,489.52	372,090.36	316,276.81	336,425.88	285,962.00
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
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
Midmarket	510,213.98	433,681.88	593,802.08	504,731.76	301,344.75	256,143.04
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



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

DATE:

IMPLEMENTATION OF CLASSIFICATION ALGORITHM IN R PROGRAMMING

AIM:

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

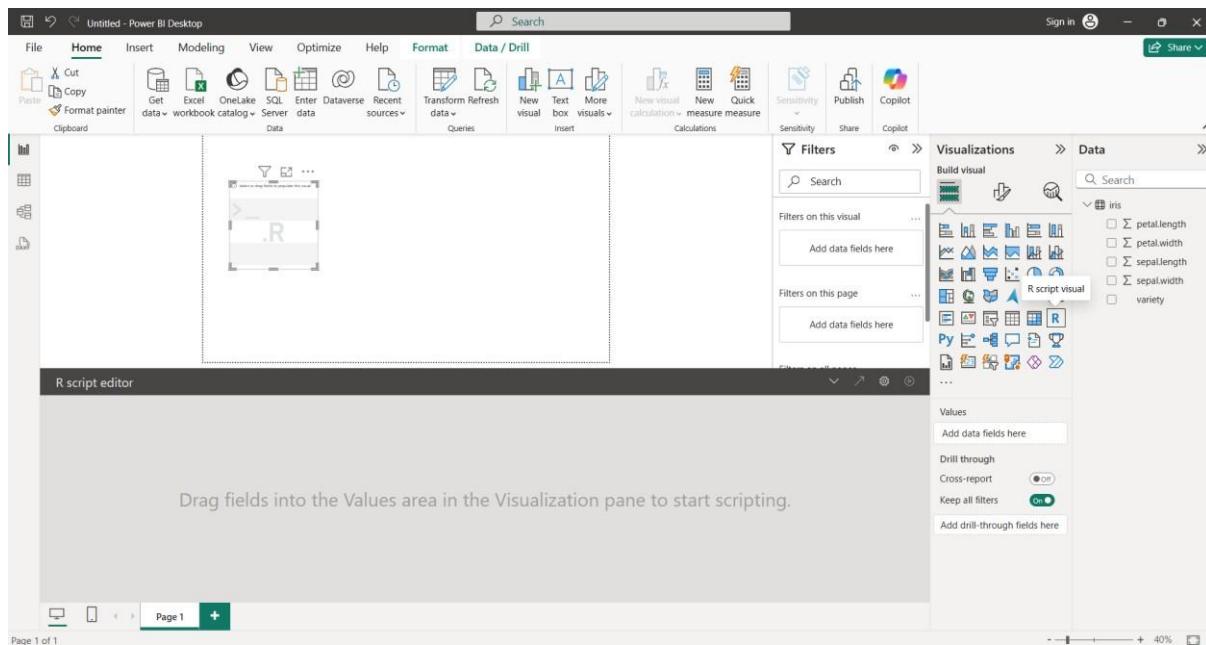
PROCEDURE:

Step 1: Import The Dataset into Power BI

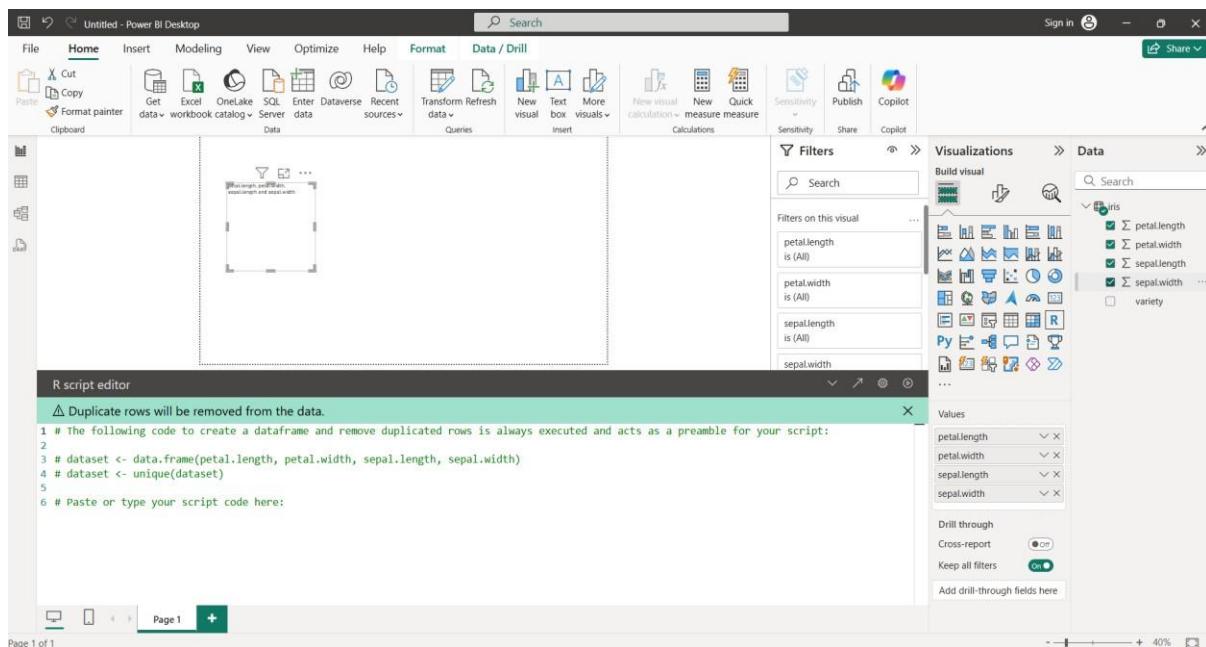
The screenshot shows the Power BI Desktop interface. The main area displays the message "Add data to your report" with options to "Import data from Excel", "Import data from SQL Server", "Paste data into a blank table", and "Use sample data". To the right is the "Visualizations" pane showing various chart and matrix options. Below the main area is the "Navigator" pane, which lists the "iris.xlsx [1]" file and the "iris" table. The "iris" table preview shows columns: sepal.length, sepal.width, petal.length, petal.width, and variety. The data consists of 150 entries for three species: Setosa, Versicolor, and Virginica. At the bottom of the Navigator pane are buttons for "Load", "Transform Data", and "Cancel".

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

The screenshot shows the Power BI Desktop interface. The script editor window contains the following R code:

```

7 library(rpart)
8 library(rpart.plot)
9 library(caret)
10
11 # Load dataset
12 dataset <- read.csv("C:/Users/AI_LAB/Desktop/iris.csv")
13
14 # Convert categorical variables to factors (if 'Species' column exists)
15 dataset$Species <- as.factor(dataset$Species)
16
17 # Split data into training (80%) and testing (20%)
18 set.seed(123)
19 trainIndex <- createDataPartition(dataset$Species, p = 0.8, list = FALSE)

```

The visualizations pane on the right shows a decision tree visualization for the 'Variety' column, with nodes for 'Setosa', 'Versicolor', and 'Virginia'. The data pane shows fields: petal.length, petal.width, sepal.length, sepal.width, and variety.

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

The screenshot shows the Power BI Desktop interface after running the script. The visualizations pane now displays a decision tree for the 'Variety' column, with nodes for 'Setosa', 'Versicolor', and 'Virginia'. The data pane shows fields: petal.length, petal.width, sepal.length, sepal.width, and variety.

The script editor window now contains the full R code for building the decision tree:

```

1 # The following code to create a data frame and remove duplicated rows is always executed and acts as a preamble for your script:
2
3 # dataset <- data.frame(petal.length, petal.width, sepal.length, sepal.width, variety)
4 # dataset <- unique(dataset)
5
6 # Posts or type your script code here:
7 library(rpart)
8 library(rpart.plot)
9 library(caret)
10
11 # Load dataset
12 dataset <- read.csv("C:/Users/aatre/OneDrive/Desktop/iris.csv")
13
14 # Convert categorical variables to factors (if 'Species' column exists)
15 dataset$Species <- as.factor(dataset$Species)
16
17 # Split data into training (80%) and testing (20%)
18 set.seed(123)
19 trainIndex <- createDataPartition(dataset$Species, p = 0.8, list = FALSE)
20 trainData <- dataset[trainIndex, ]
21 testData <- dataset[-trainIndex, ]
22
23 # Build Decision Tree Model

```

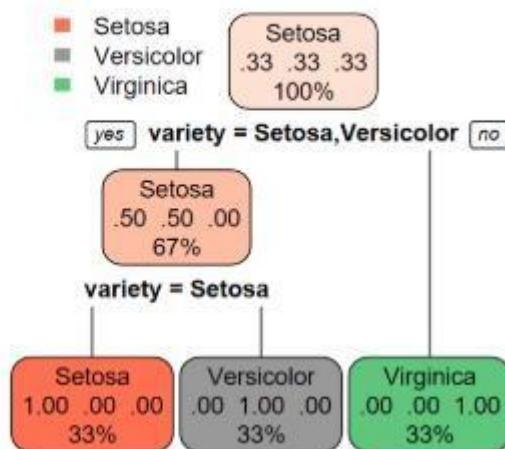
CODE:

```
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

DATE:

IMPLEMENTATION OF DECISION TREE USING R TOOL

AIM:

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

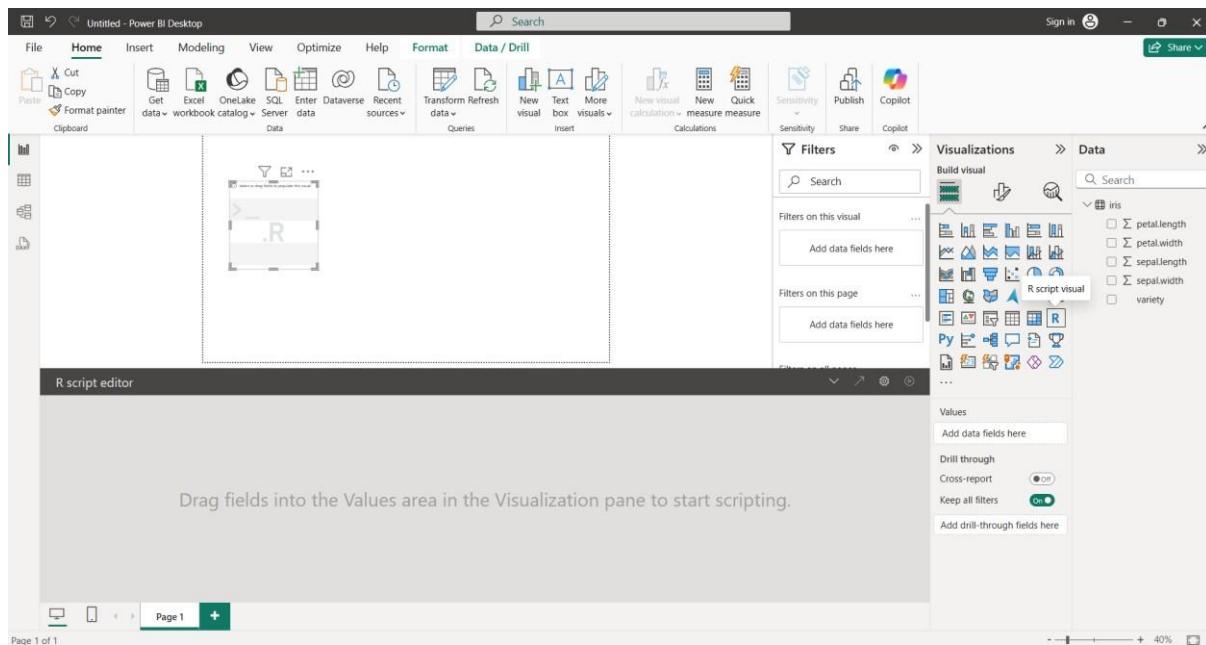
PROCEDURE:

Step 1: Import The Dataset into Power BI

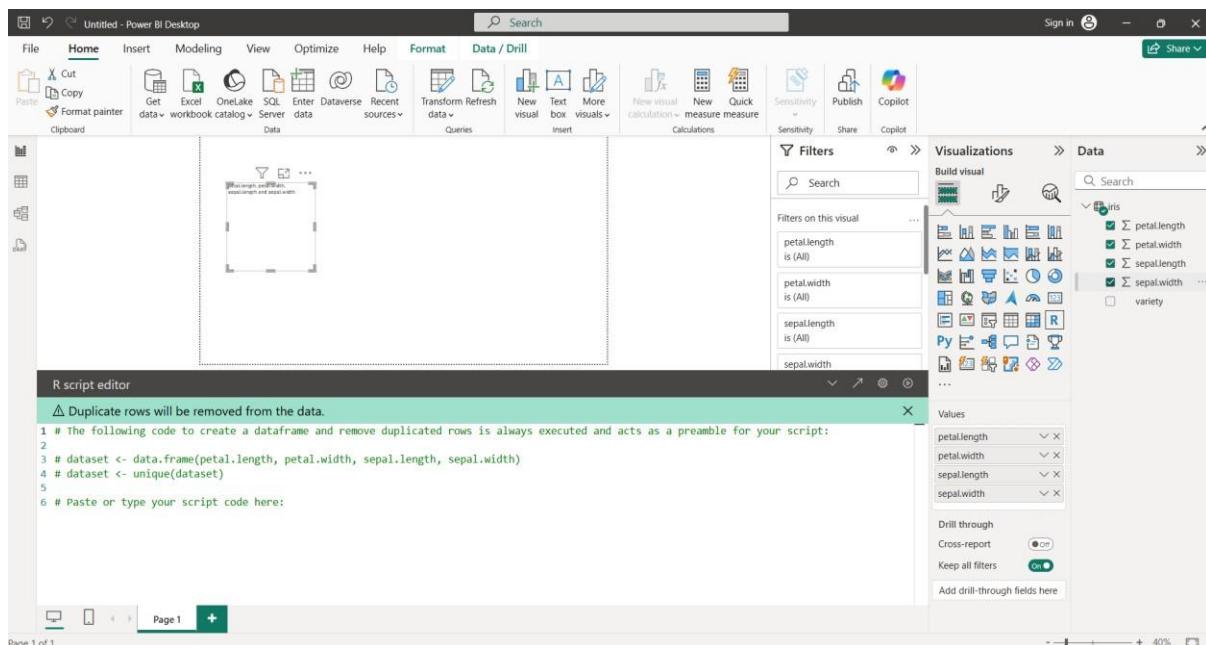
The screenshot shows the Power BI Desktop interface. The main canvas displays a message: "Add data to your report. Once loaded, your data will appear in the Data pane." Below this are four buttons: "Import data from Excel", "Import data from SQL Server", "Post data into a blank table", and "Use sample data". To the right of the canvas is the "Visualizations" pane, which lists various chart and matrix types. Below the canvas is the "Navigator" pane, which shows a file tree with "iris.xlsx [1]" selected. A preview of the "iris" table is displayed, showing columns: sepal.length, sepal.width, petal.length, petal.width, and variety. The data preview shows 50 rows of Iris flower measurements. At the bottom of the Navigator pane are buttons for "Load", "Transform Data", and "Cancel".

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

The screenshot shows the Power BI Desktop interface. On the left, the R script editor contains the following code:

```

15 dataset$variety <- as.factor(dataset$variety) # Target variable
16
17 # Split data into training (80%) and testing (20%)
18 set.seed(123)
19 trainIndex <- sample(1:nrow(dataset), 0.8 * nrow(dataset))
20 trainData <- dataset[trainIndex, ]
21 testData <- dataset[-trainIndex, ]
22
23 # Build Decision Tree Model
24 dt_model <- rpart(variety ~ ., data = trainData, method = "class")
25
26 # Plot the Decision Tree
27 rpart.plot(dt_model)
  
```

The main canvas displays a decision tree visualization for the Iris dataset. The root node splits on petal.length < 2.5. The left branch leads to a node labeled "Setosa" with 100% accuracy and 33 samples. The right branch leads to a further split on petal.length < 4.8, resulting in two nodes: one for "Versicolor" (27% accuracy, 50 samples) and one for "Virginica" (40% accuracy, 58 samples).

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

The screenshot shows the Power BI Desktop interface with the decision tree visualization. The R script editor now includes the following code:

```

# The following code to create a data frame and remove duplicated rows is always executed and acts as a preamble for your script:
# dataset <- data.frame(petal.length, petal.width, sepal.length, sepal.width, variety)
# dataset <- unique(dataset)
#
# Paste or type your script code here:
1 library(rpart)
2 library(rpart.plot)
3 library(tidyverse)
4
5 # Load the dataset
6 dataset <- read.csv("C:/Users/aatre/OneDrive/Desktop/Iris.csv")
7
8 # Convert categorical variables to factors
9 dataset$variety <- as.factor(dataset$variety) # Target variable
10
11 # Split data into training (80%) and testing (20%)
12 set.seed(123)
13 trainIndex <- sample(1:nrow(dataset), 0.8 * nrow(dataset))
14 trainData <- dataset[trainIndex, ]
15 testData <- dataset[-trainIndex, ]
16
17 # Build Decision Tree Model
  
```

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")

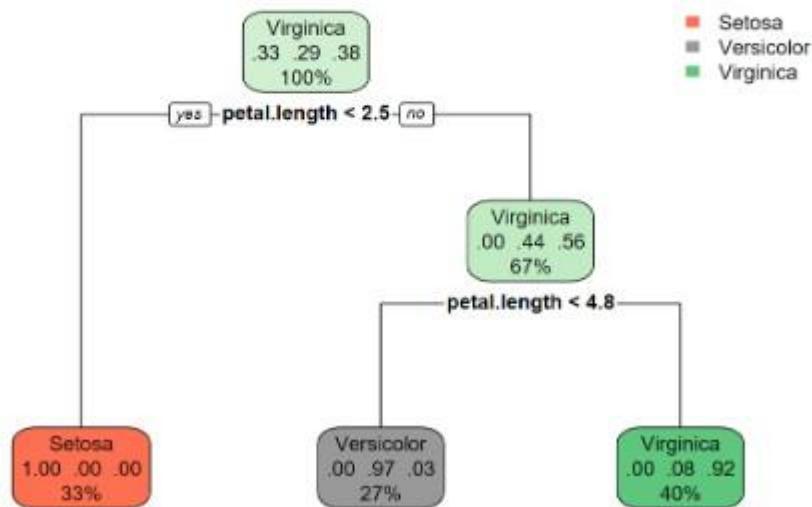
# Evaluate model performance

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

DATE:

K-MEANS CLUSTERING USING R PROGRAMMING

AIM:

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

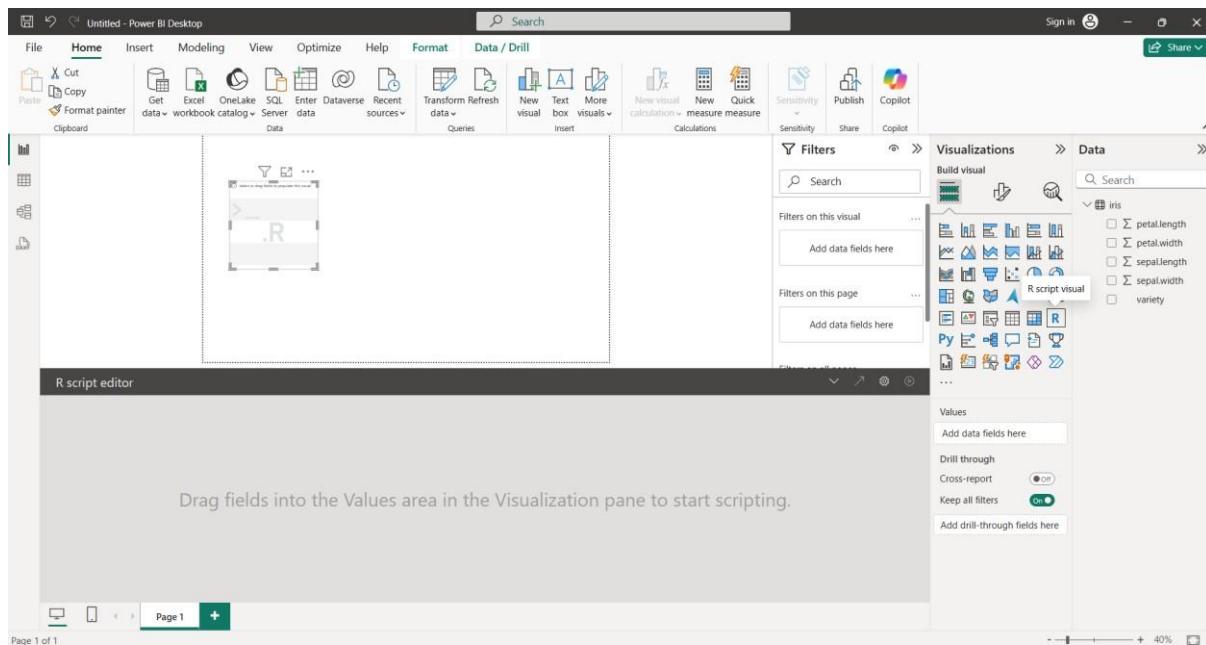
PROCEDURE:

Step 1: Import the Dataset into Power BI

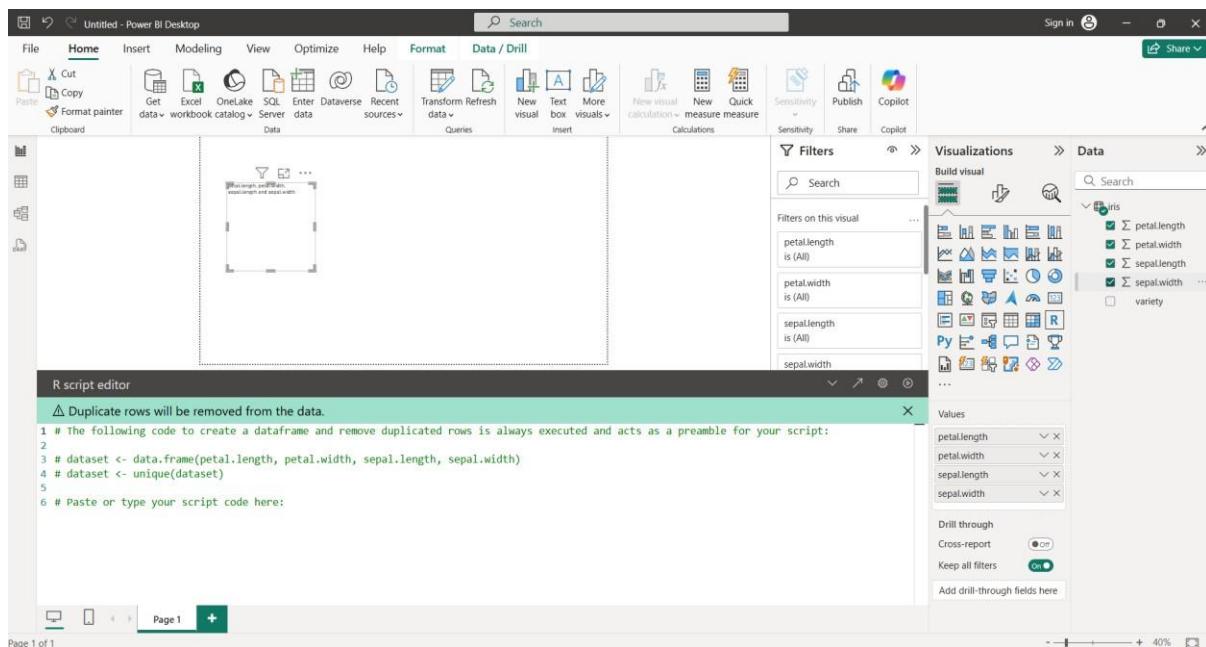
The screenshot shows the Power BI Desktop interface. The main canvas area displays a message: "Add data to your report. Once loaded, your data will appear in the Data pane." Below this are four buttons: "Import data from Excel", "Import data from SQL Server", "Paste data into a blank table", and "Use sample data". To the right of the canvas is the "Visualizations" pane, which lists various chart and matrix types. Below the canvas is the "Navigator" pane, which shows the "iris.xlsx [1]" file is selected. The "iris" sheet is open, displaying a table with columns: sepal.length, sepal.width, petal.length, petal.width, and variety. The data consists of 150 entries for three species: Setosa, Versicolor, and Virginica. At the bottom of the Navigator pane are "Load", "Transform Data", and "Cancel" buttons.

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

The screenshot shows the Power BI Desktop interface. The ribbon at the top includes File, Home, Insert, Modeling, View, Optimize, Help, Format, Data / Drill, Table tools, Column tools, and a sign-in button. The main area features the R script editor with the following code:

```


# Duplicate rows will be removed from the data.
20 # Apply K-Means clustering
21 kmeans_model <- kmeans(df, centers = k, nstart = 25)
22
23 # Add cluster labels to the dataset
24 dataset$Cluster <- as.factor(kmeans_model$cluster)
25
26 # Visualize the clusters (choose two dimensions to plot)
27 ggplot(dataset, aes(x = dataset[,1], y = dataset[,2], color = Cluster)) +
28   geom_point(size = 3) +
29   labs(title = "K-Means Clustering on Iris Dataset",
30       x = colnames(dataset)[1],
31       y = colnames(dataset)[2]) +
32   theme_minimal()


```

The visualizations pane on the right shows various chart types like bar charts, line graphs, and maps, with several items selected under the 'Iris' category. The data pane lists fields such as petal.length, petal.width, sepal.length, sepal.width, and variety.

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

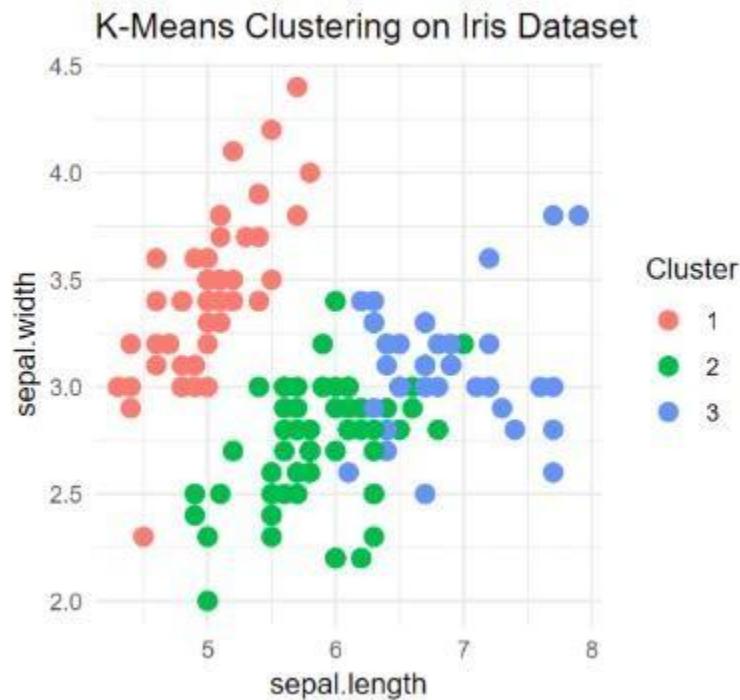
The screenshot shows the Power BI Desktop interface after running the script. The main area displays a scatter plot titled 'K-Means Clustering on Iris Dataset'. The x-axis is labeled 'sepal.length' and ranges from approximately 4.0 to 7.8. The y-axis is labeled 'petal.width' and ranges from approximately 1.0 to 4.5. The data points are colored by cluster, forming three distinct groups. The visualizations pane and data pane are visible on the right, and the script editor at the bottom shows the executed R 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:

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

**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

Date:

PREDICTION USING LINEAR REGRESSION

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**".

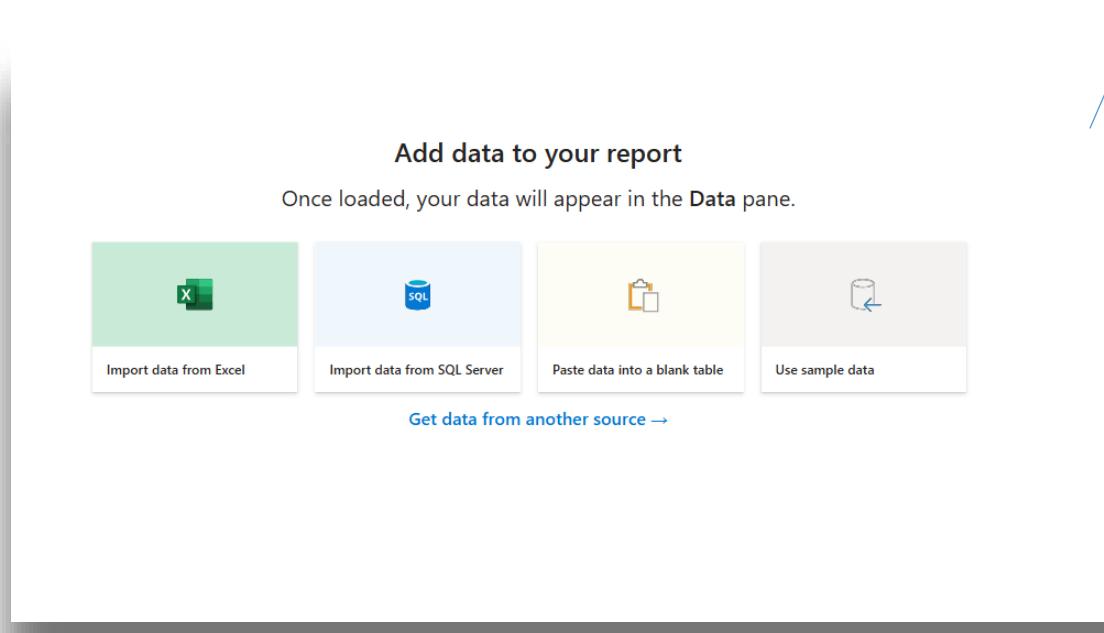
4. Import the Dataset into Power BI

- 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.

Data tab



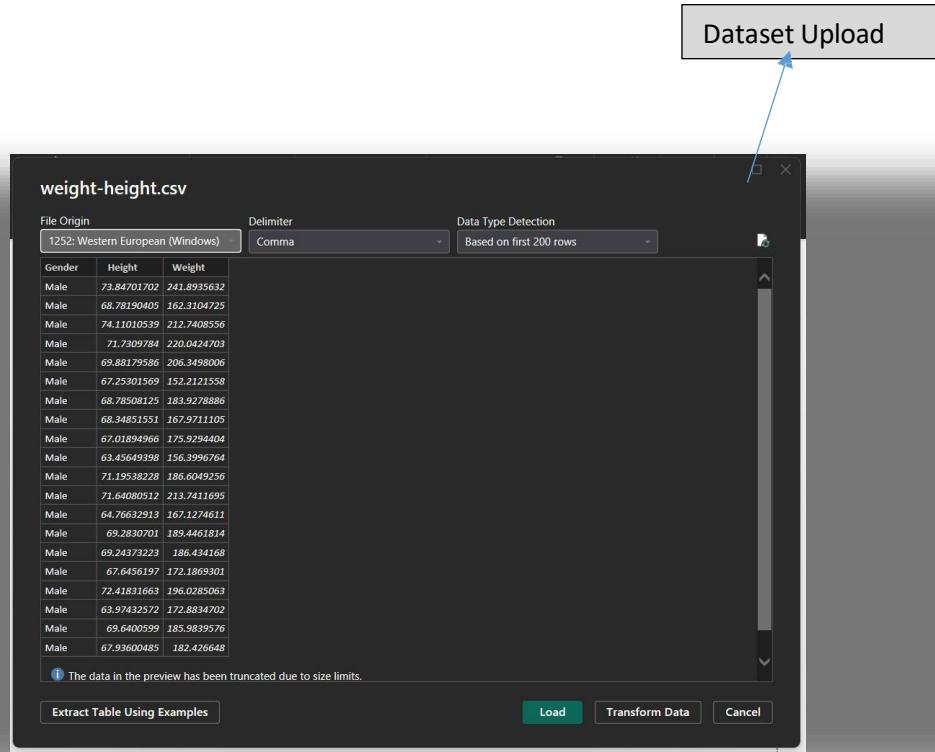
3. Visualize Time Series Data in Power BI

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**".



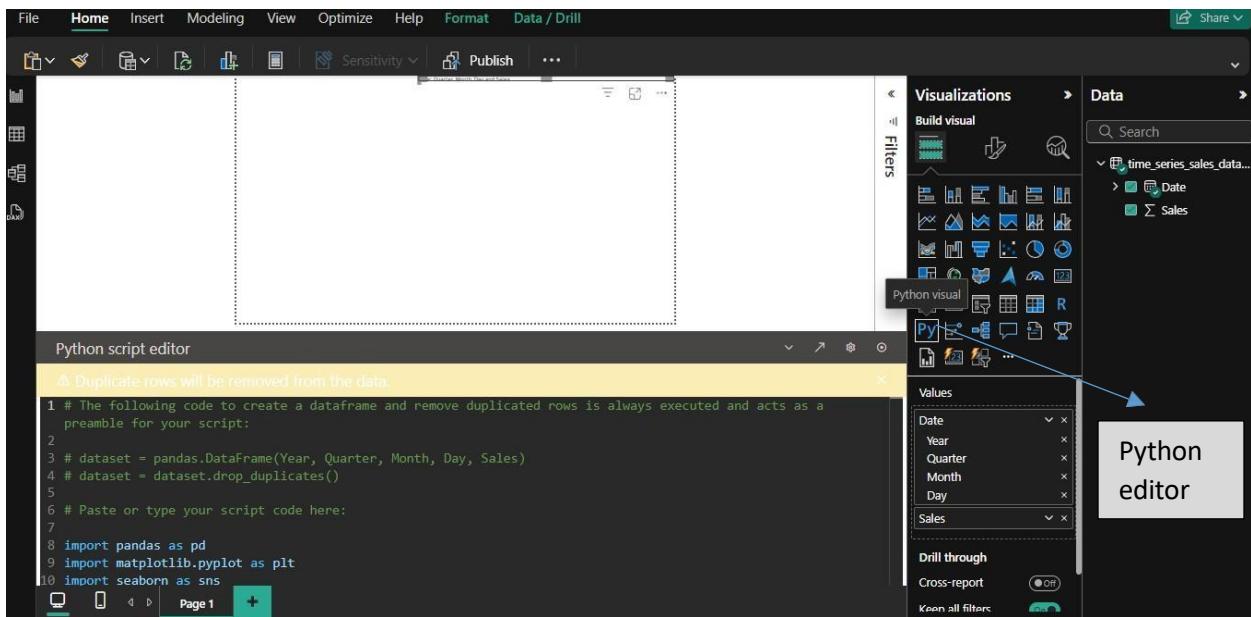
4. Run Python Script for Advanced Analysis

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.

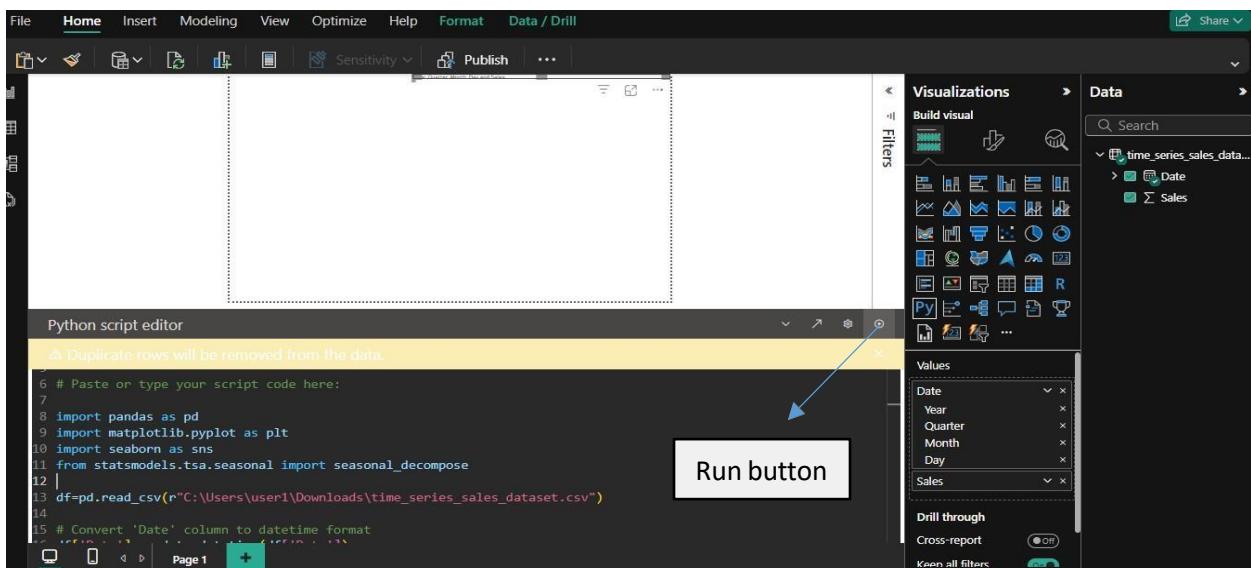


10. Paste and Run the Python Script

- 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

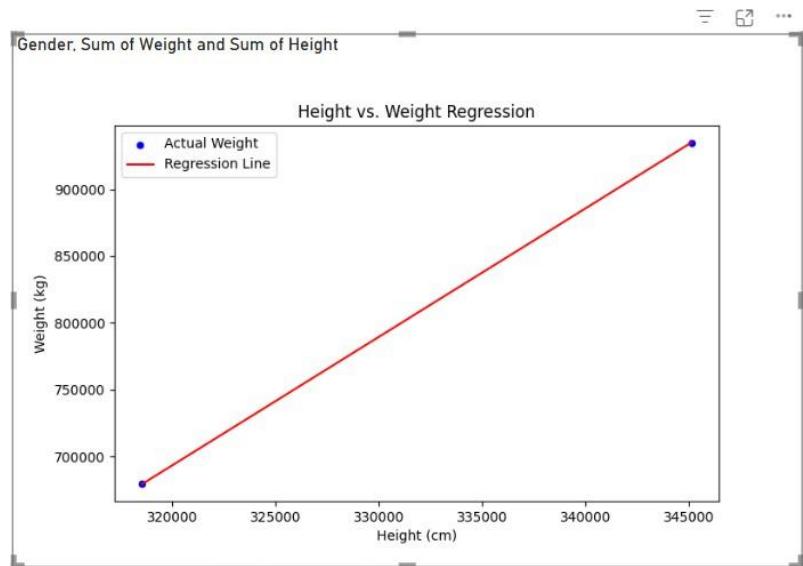
# Define independent (X) and dependent (Y) variables
X = dataset[['Height']] # Predictor
y = dataset['Weight'] # Response Variable

# 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

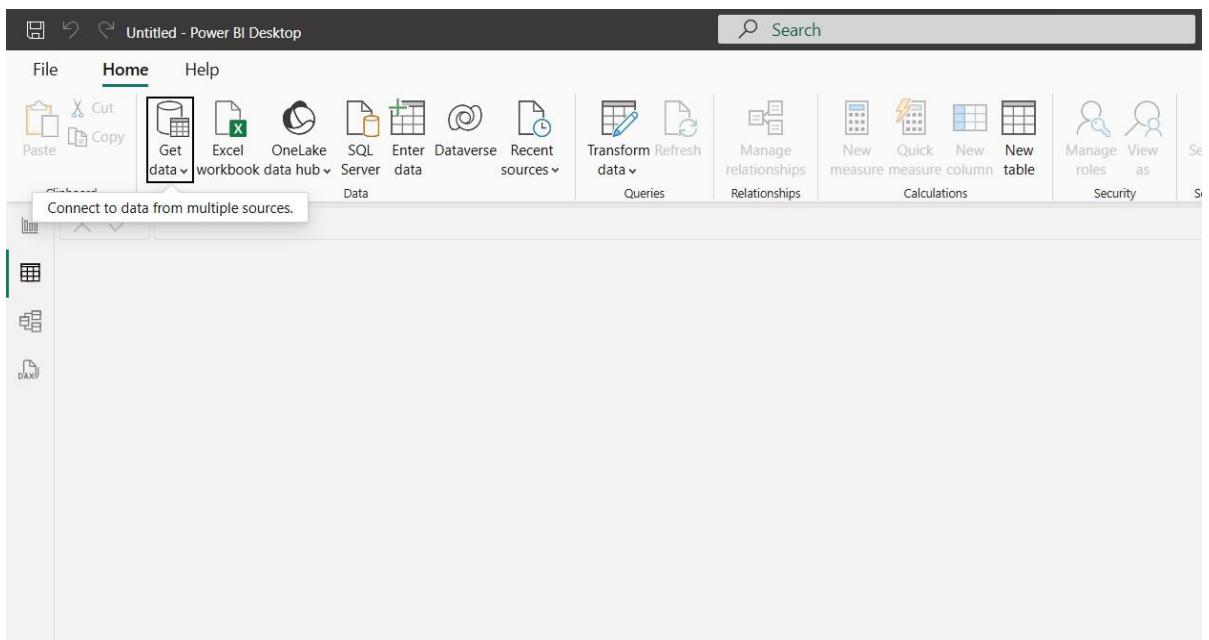
DATE:

AIM:

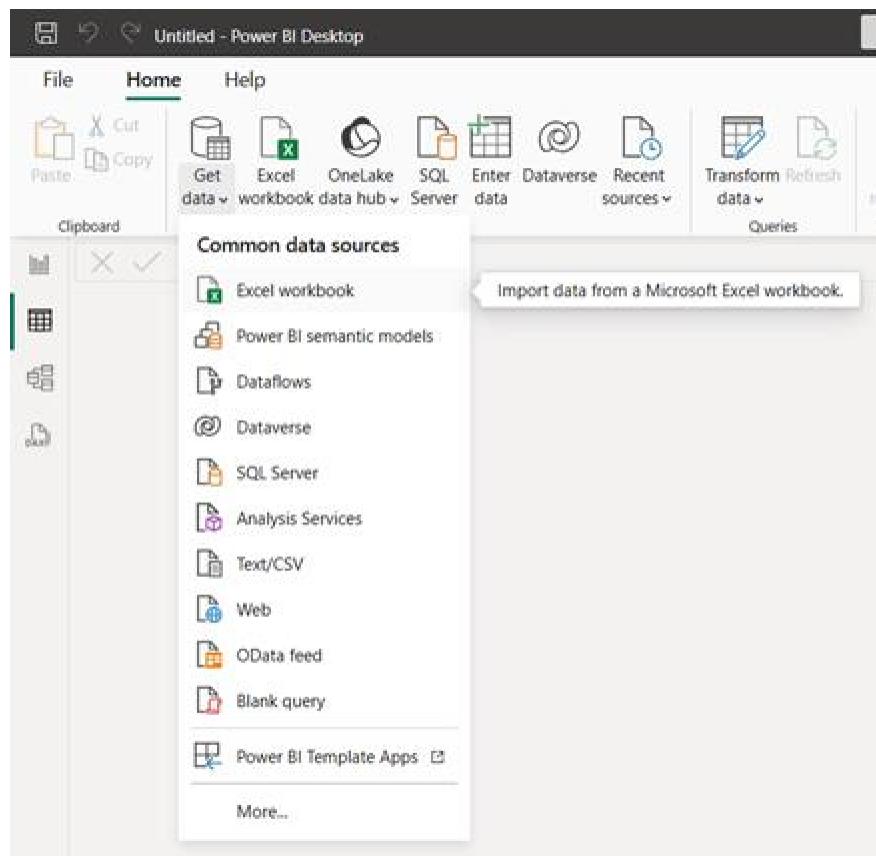
To Create a Dashboard Analytics on Real-Time Sales Using Time Series Dataset

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:

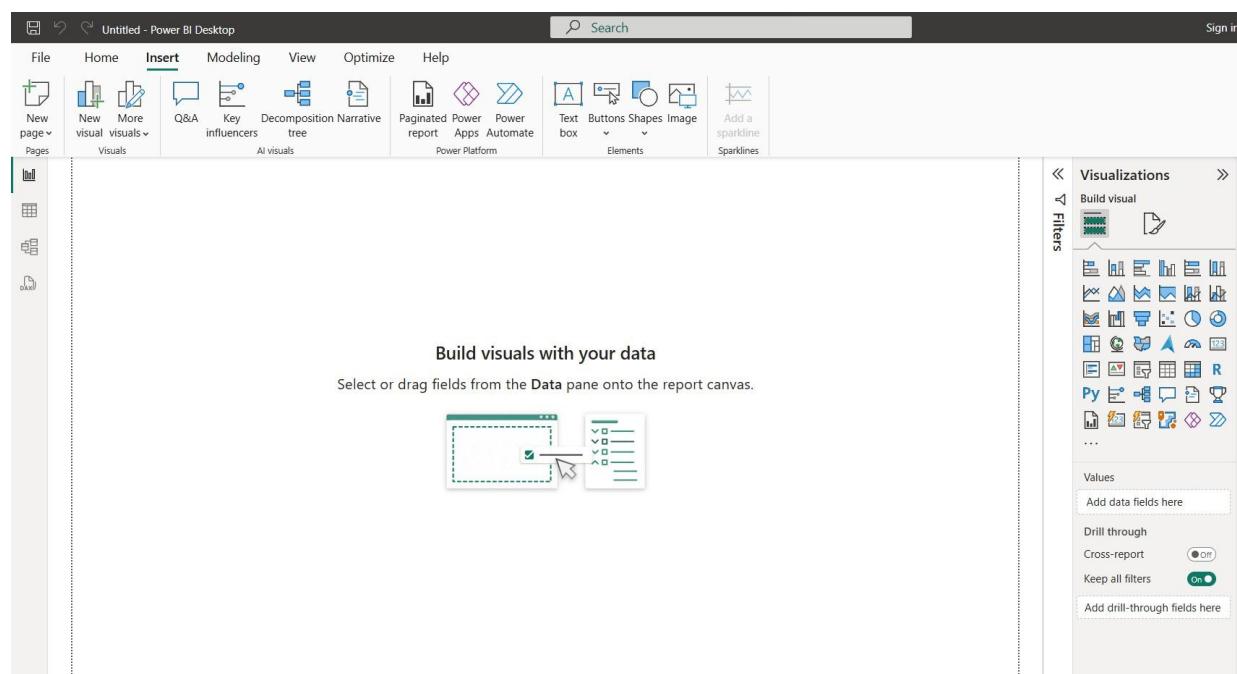
	Date	Year	Quarter	Month	Month Name	SerialNum	Year-Month	Weeks
7/1/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/2/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/3/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/4/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/5/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/6/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/7/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/8/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/9/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/10/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/11/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/12/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/13/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/14/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/15/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/16/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/17/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/18/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/19/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/20/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/21/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/22/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/23/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/24/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/25/2023	2023	Q3		7	Jul	202307	2023-Jul	3:
7/26/2023	2023	Q3		7	Jul	202307	2023-Jul	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:

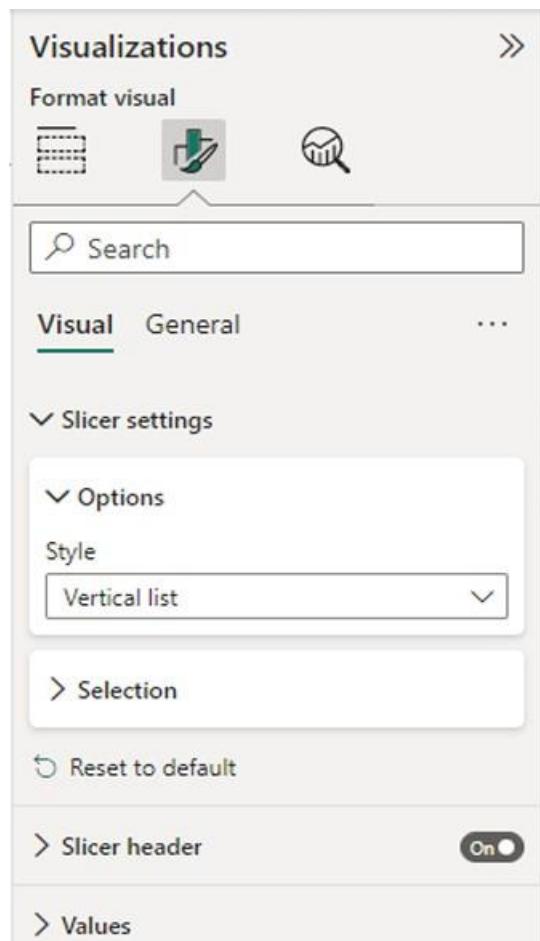


The next step was to click on "Insert" in the ribbon, then select "New visual" to start adding visualizations for your chocolate sales dataset. The current image shows Power BI Desktop with a blank report canvas and the "Visualizations" pane open on the right, indicating you've already followed that 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.

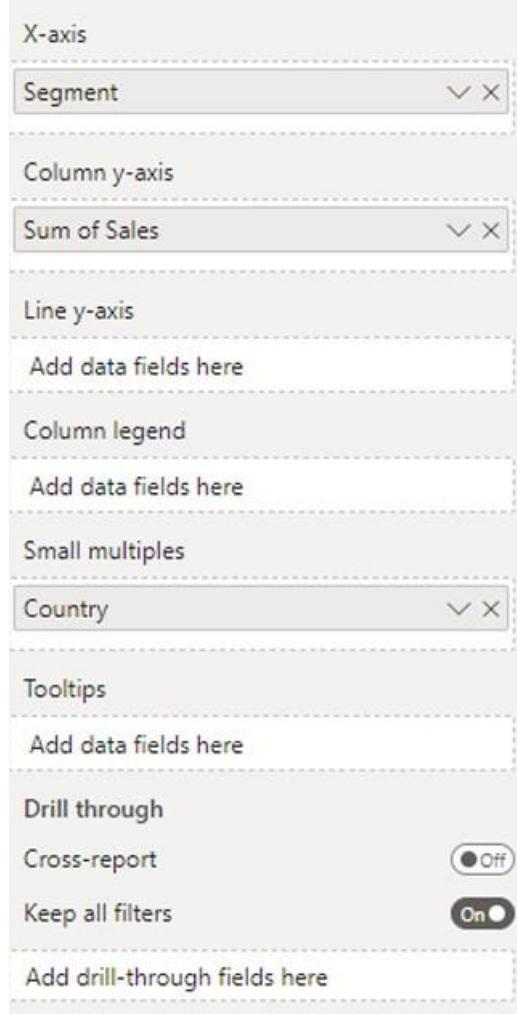
Step 5:

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.

- 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.
- 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.
- 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.
- 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."

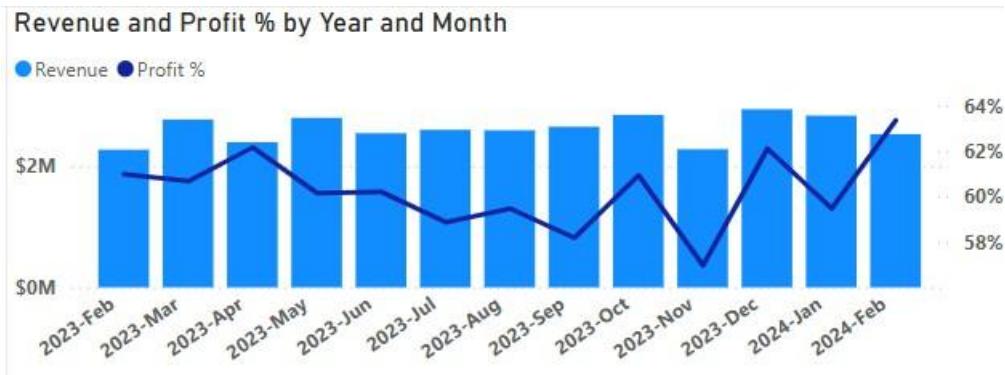
Step 7:

\$34M	2M	60.29%	\$21M	6113	624
Revenue	Total Boxes	Profit %	Total Profit	Total Transactions	Low-Box Shipments

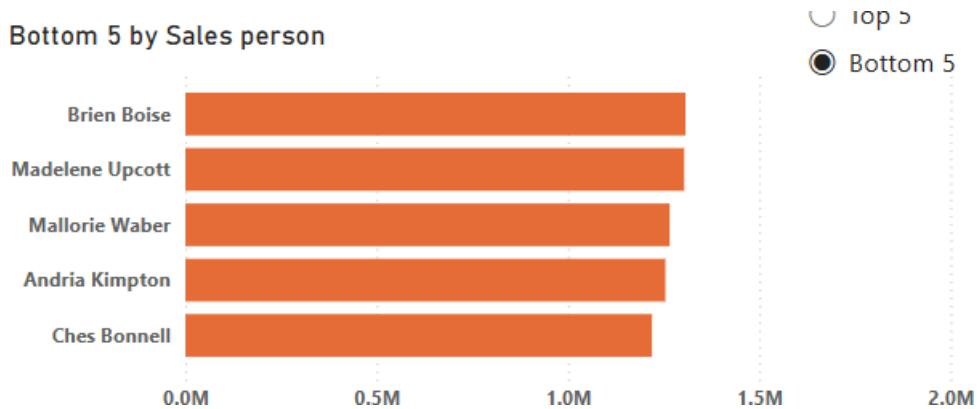
This visualized data in Power BI is likely represented using a Card visualization for each metric (Revenue, Total Boxes, Profit %, Total Profit, Total Transactions, Low-Box Shipments). Each card displays a single key value (\$34M, 2M, 60.29%, \$21M, 6113, 624) with its corresponding label, arranged horizontally 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%

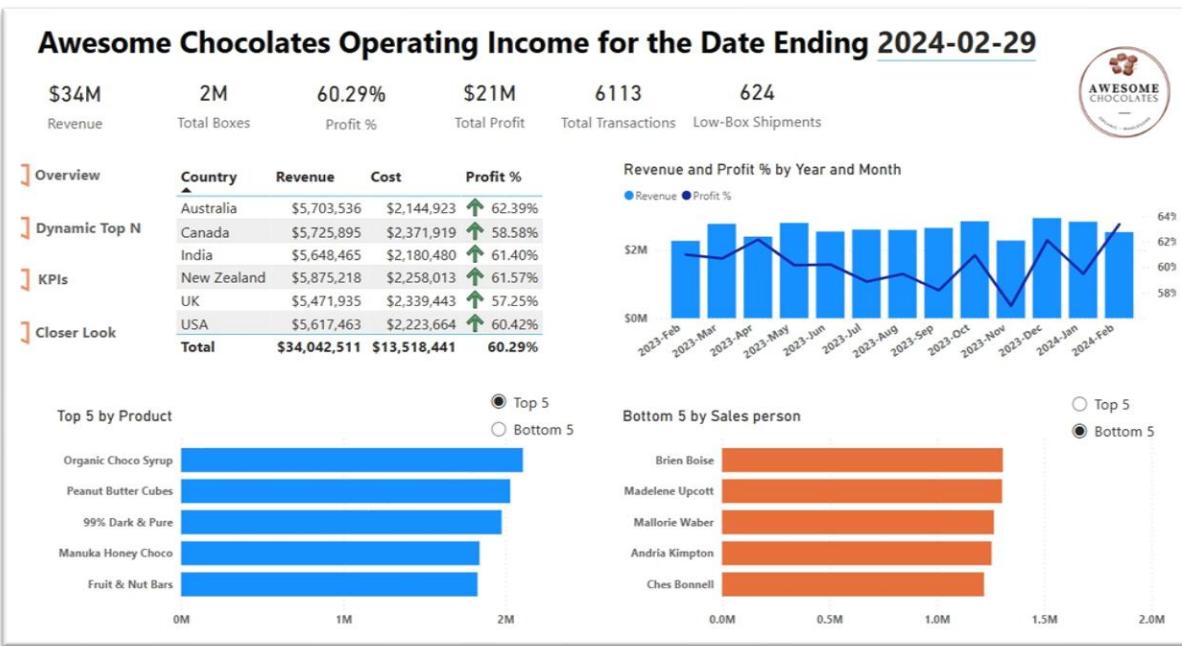
This data is visualized in Power BI using a Table visualization. The table displays columns for "Country," "Revenue," "Cost," and "Profit %," with rows for each country (Australia, Canada, India, New Zealand, UK, USA) and a total row. You can create this by selecting the Table icon from the "Visualizations" 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**

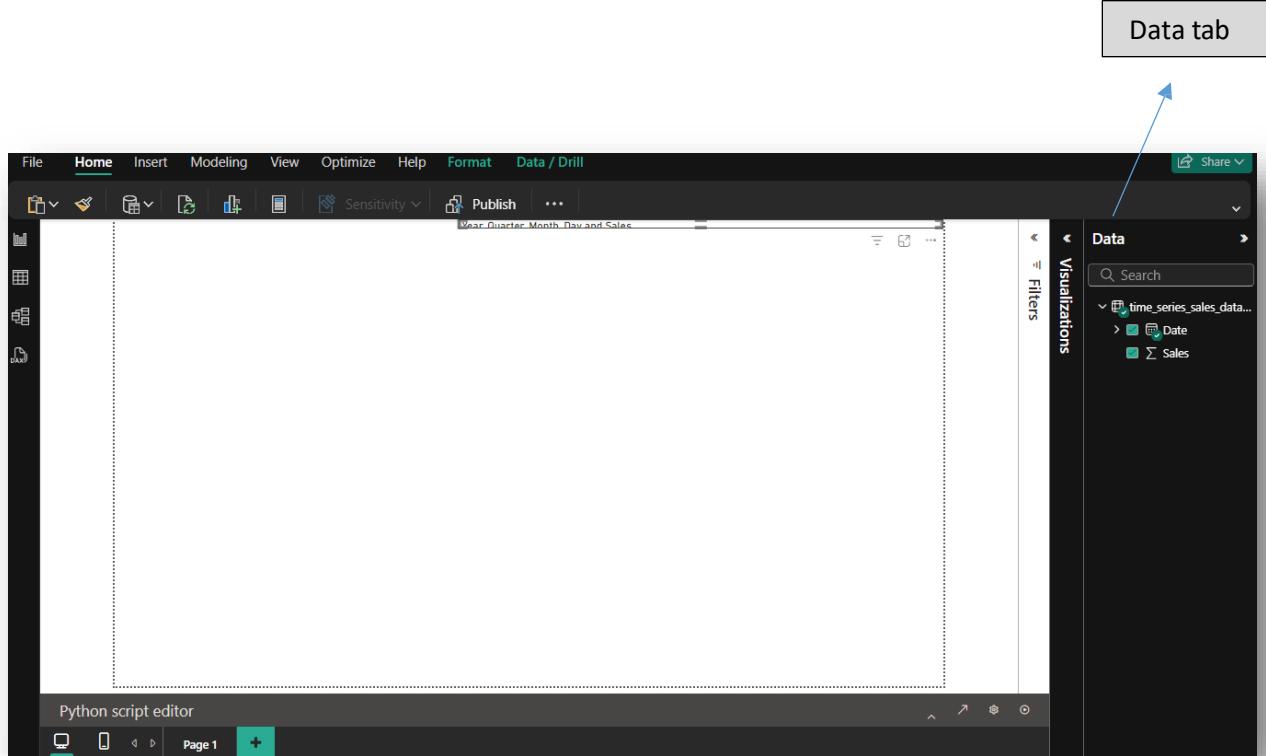
- Ensure you have a **time series dataset** in a suitable format (CSV, TXT, etc.).
- Example dataset: "**time_series_sales_dataset.csv**".

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.



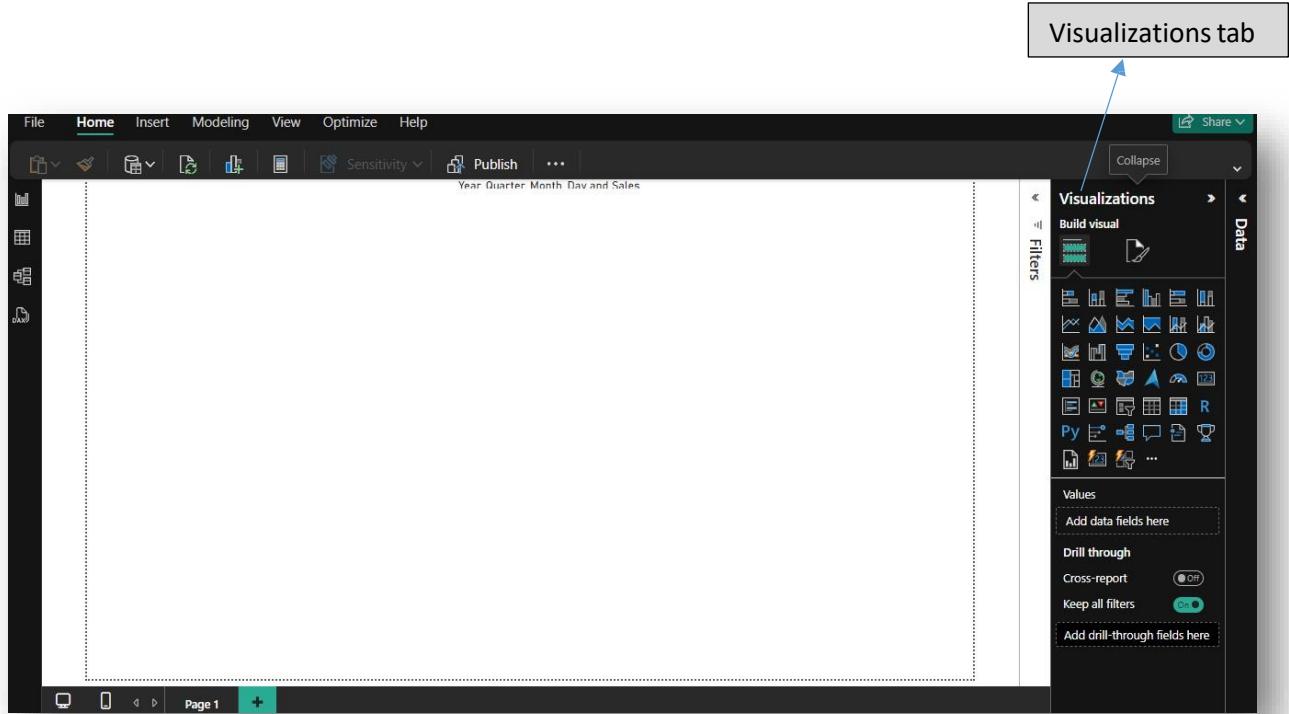
3. Visualize Time Series Data in Power BI

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**".



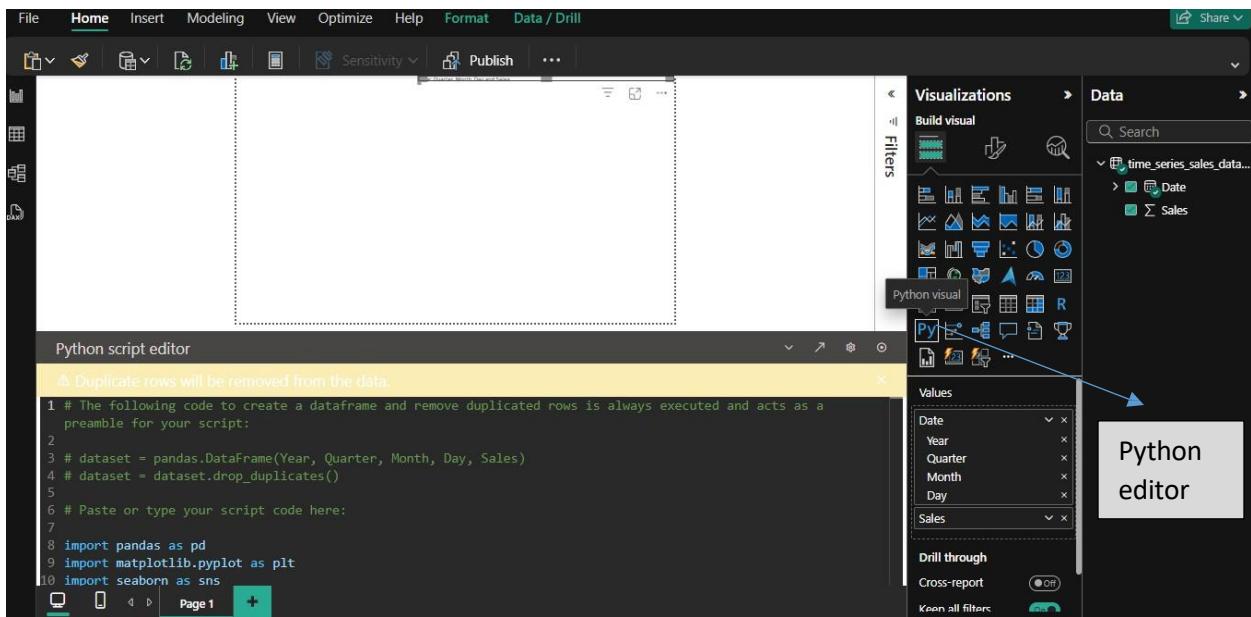
4. Run Python Script for Advanced Analysis

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.

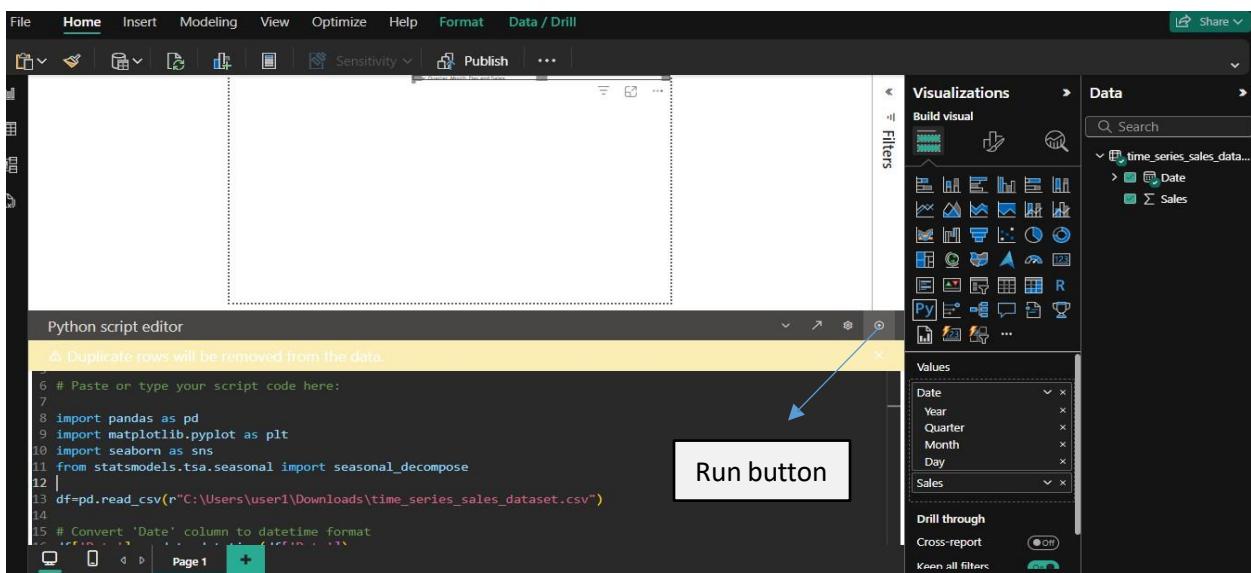


10. Paste and Run the Python Script

- 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()
```

```

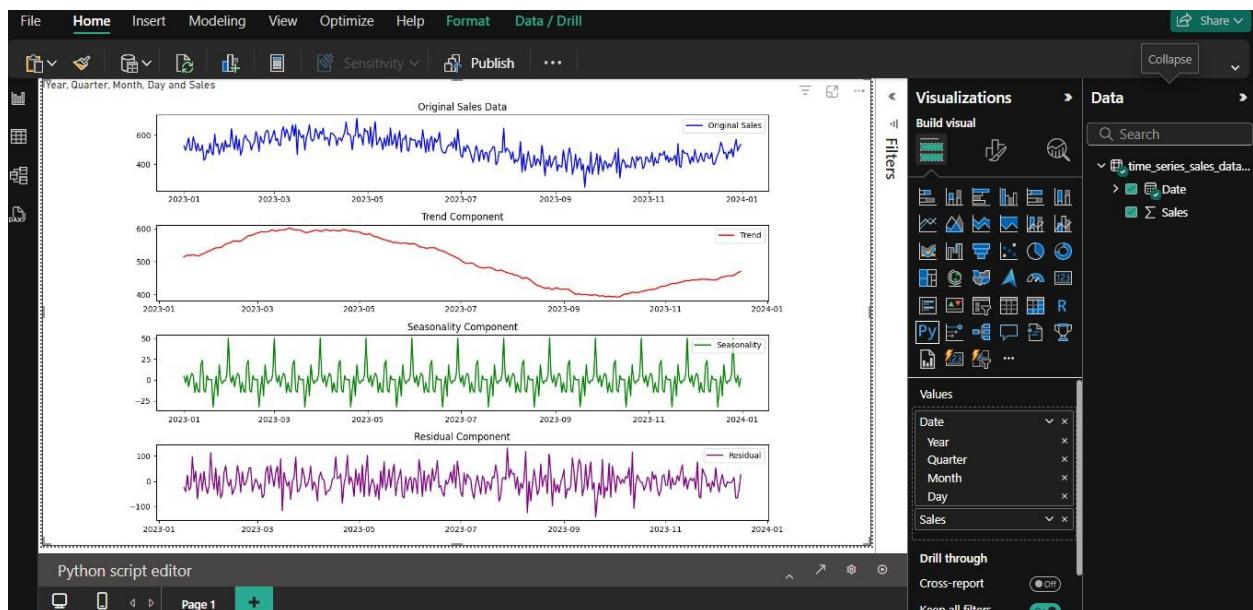
# Seasonal Component
plt.subplot(4, 1, 3)
plt.plot(df.index, df['Seasonality'], label='Seasonality', color='green')
plt.title('Seasonality Component')
plt.legend()

# 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

Date:

Data Modelling and Analytics with Pivot Table in Power BI

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

DATE:

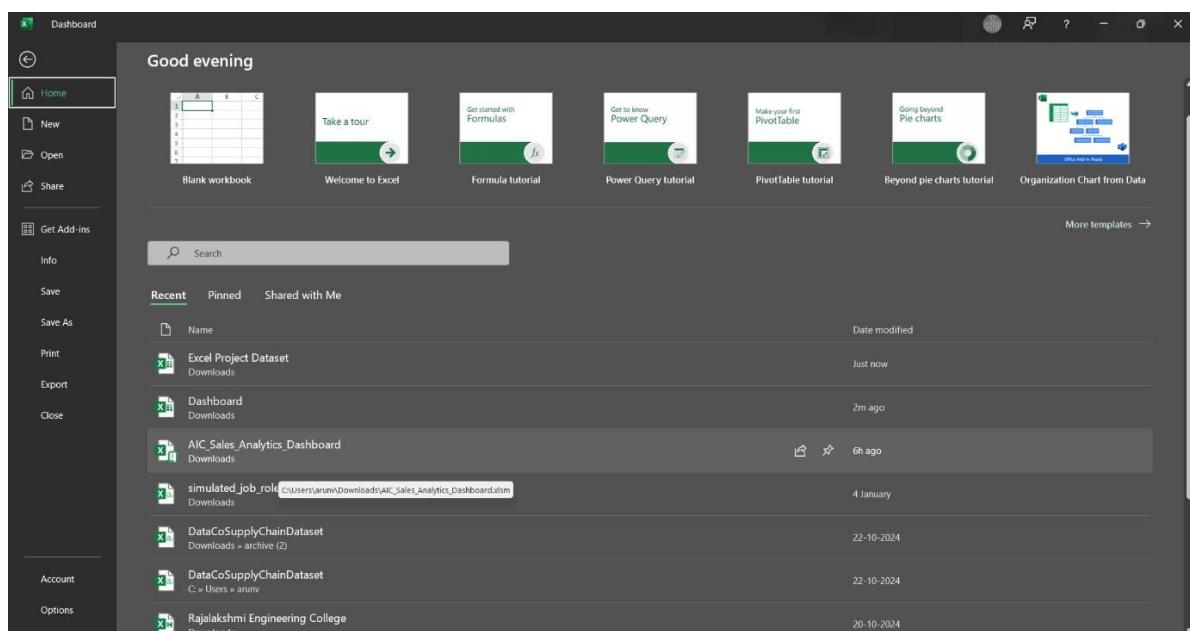
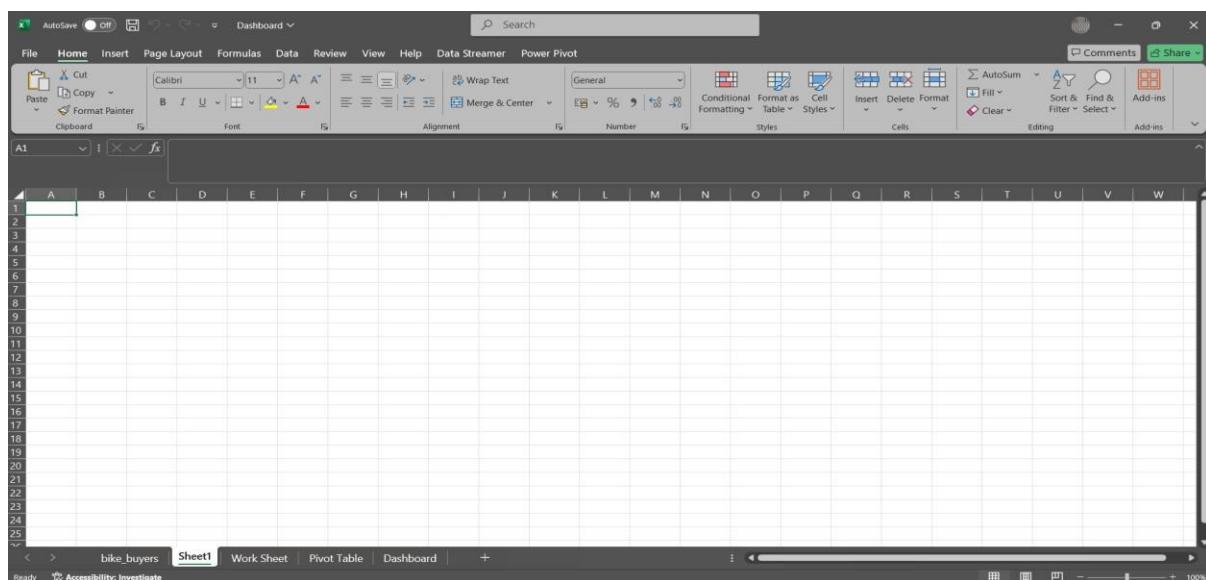
DATA ANALYSIS AND VISUALIZATION USING ADVANCED EXCEL

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.



The screenshot shows a Microsoft Excel spreadsheet titled "Marital Status". The table has 25 rows and 14 columns. The columns are labeled: ID, Marital Status, Gender, Income, Children, Education, Occupation, Home Owner, Cars, Commute Dist, Region, Age, and Purchased Bike. The "Purchased Bike" column contains many blank entries. The "Age" column has some numerical values like 42, 43, 60, etc., and some categorical values like "Middle Age" and "Old".

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.

The screenshot shows the same Microsoft Excel spreadsheet as before, but with a "Find and Replace" dialog box overlaid. The "Find what:" field is set to "M" and the "Replace with:" field is set to "Married". The "Replace All" button is highlighted. The table data remains the same, showing marital status categories like Married, Single, and Married again.

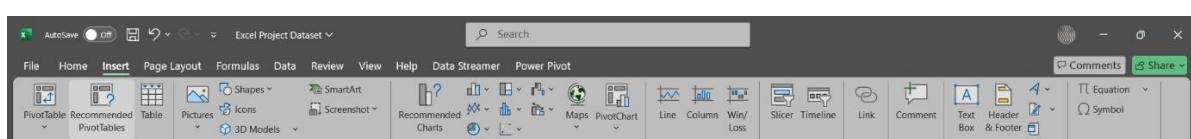
Step 3: For the Bike sales we need to capture the specific age group so include a row (Age Brackets) for the better view.

ID	Marital Status	Gender	Income	Children	Education	Occupation	Home Owner	Cars	Commute	Region	Age	Age Brackets	Purchased Bike
2	12496 Married	Female	\$40,000.00	1	Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	42	Middle Age	No
3	24107 Married	Male	\$30,000.00	3	Partial Col	Clerical	Yes	1	0-1 Miles	Europe	43	Middle Age	No
4	14177 Married	Male	\$80,000.00	5	Partial Col	Professional	No	2	2-5 Miles	Europe	60	Old	No
5	24381 Single	Male	\$70,000.00	0	Bachelors	Professional	Yes	1	5-10 Miles	Pacific	41	Middle Age	Yes
6	25597 Single	Male	\$30,000.00	0	Bachelors	Clerical	No	0	0-1 Miles	Europe	36	Middle Age	Yes
7	13507 Married	Female	\$10,000.00	2	Partial Col	Manual	Yes	0	1-2 Miles	Europe	50	Middle Age	No
8	27974 Single	Male	\$1,60,000.00	2	High Schoc	Management	Yes	4	0-1 Miles	Pacific	33	Middle Age	Yes
9	19364 Married	Male	\$40,000.00	1	Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	43	Middle Age	Yes
10	22155 Married	Male	\$20,000.00	2	Partial Hig	Clerical	No	2	5-10 Miles	Pacific	58	Old	No
11	19280 Married	Male	\$1,20,000.00	2	Partial Col	Manual	Yes	1	0-1 Miles	Europe	40	Middle Age	Yes
12	22173 Married	Female	\$30,000.00	3	High Schoc	Skilled Manual	No	2	1-2 Miles	Pacific	54	Middle Age	Yes
13	12697 Single	Female	\$90,000.00	0	Bachelors	Professional	No	4	More than	Pacific	36	Middle Age	No
14	11434 Married	Male	\$1,70,000.00	5	Partial Col	Professional	Yes	0	0-1 Miles	Europe	55	Old	No
15	25323 Married	Male	\$40,00,000	2	Partial Col	Clerical	Yes	1	1-2 Miles	Europe	35	Middle Age	Yes
16	23542 Single	Male	\$60,00,000	1	Partial Col	Skilled Manual	No	1	0-1 Miles	Pacific	45	Middle Age	Yes
17	20870 Single	Female	\$10,00,000	2	High Schoc	Manual	Yes	1	0-1 Miles	Europe	38	Middle Age	Yes
18	23316 Single	Male	\$30,00,000	3	Partial Col	Clerical	No	2	1-2 Miles	Pacific	59	Old	Yes
19	12610 Married	Female	\$30,00,000	1	Bachelors	Clerical	Yes	0	0-1 Miles	Europe	47	Middle Age	No
20	27183 Single	Male	\$40,00,000	2	Partial Col	Clerical	Yes	1	1-2 Miles	Europe	35	Middle Age	Yes
21	25940 Single	Male	\$20,00,000	2	Partial Hig	Clerical	Yes	2	5-10 Miles	Pacific	55	Old	Yes
22	25598 Married	Female	\$40,00,000	0	Graduate I	Clerical	Yes	0	0-1 Miles	Europe	36	Middle Age	Yes
23	21564 Single	Female	\$80,00,000	0	Bachelors	Professional	Yes	4	More than	Pacific	35	Middle Age	No
24	19193 Single	Male	\$40,00,000	2	Partial Col	Clerical	Yes	0	1-2 Miles	Europe	35	Middle Age	Yes
25	26412 Married	Female	\$80,00,000	5	High Schoc	Management	No	3	5-10 Miles	Europe	56	Old	No

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

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
25	26412 Married	Female	\$80,00,000	5	High Schoc	Management	No	3	5-10 Miles	Europe	56	Old	No						
26	27184 Single	Male	\$40,00,000	2	Partial Col	Clerical	No	1	0-1 Miles	Europe	34	Middle Age	No						
27	12590 Single	Male	\$30,00,000	1	Bachelors	Clerical	Yes	0	0-1 Miles	Europe	63	Old	No						
28	17841 Single	Male	\$30,00,000	0	Partial Col	Clerical	No	1	0-1 Miles	Europe	29	Adolescent	Yes						
29	18283 Single	Female	\$1,00,00,000	0	Bachelors	Professional	No	1	5-10 Miles	Pacific	40	Middle Age	No						
30	18299 Married	Male	\$70,00,000	5	Partial Col	Skilled Manual	Yes	2	5-10 Miles	Pacific	44	Middle Age	No						
31	16466 Single	Female	\$20,00,000	0	Partial Hig	Manual	No	2	0-1 Miles	Europe	32	Middle Age	Yes						
32	19273 Married	Female	\$20,00,000	2	Partial Col	Manual	Yes	0	0-1 Miles	Europe	63	Old	No						
33	22400 Married	Male	\$10,00,000	0	Partial Col	Manual	No	1	0-1 Miles	Pacific	26	Adolescent	Yes						
34	20942 Single	Female	\$20,00,000	0	High Schoc	Manual	No	1	5-10 Miles	Europe	31	Middle Age	No						
35	18484 Single	Male	\$80,00,000	2	High Schoc	Skilled Manual	No	2	1-2 Miles	Pacific	50	Middle Age	Yes						
36	12291 Single	Male	\$90,00,000	5	Partial Col	Professional	No	2	2-5 Miles	Europe	62	Old	Yes						
37	28380 Single	Female	\$10,00,000	5	Partial Hig	Manual	Yes	0	2-1 Miles	Europe	41	Middle Age	No						
38	17891 Married	Female	\$10,00,000	2	Partial Col	Manual	Yes	1	0-1 Miles	Europe	50	Middle Age	Yes						
39	27832 Single	Female	\$30,00,000	0	Partial Col	Clerical	No	1	2-5 Miles	Europe	30	Adolescent	No						
40	26863 Single	Male	\$20,00,000	0	High Schoc	Manual	No	1	2-5 Miles	Europe	28	Adolescent	No						
41	16259 Single	Female	\$10,00,000	4	Partial Hig	Manual	Yes	2	0-1 Miles	Europe	40	Middle Age	Yes						
42	27803 Single	Female	\$30,00,000	2	Partial Col	Clerical	No	0	0-1 Miles	Europe	43	Middle Age	No						
43	14347 Single	Female	\$40,00,000	2	Bachelors	Management	Yes	2	5-10 Miles	Pacific	65	Old	Yes						
44	17703 Married	Female	\$10,00,000	1	Graduate I	Manual	Yes	0	0-1 Miles	Europe	48	Middle Age	Yes						
45	17185 Married	Female	\$1,70,00,000	4	Partial Col	Professional	No	3	5-10 Miles	Europe	41	Middle Age	Yes						
46	29380 Married	Female	\$20,00,000	3	High Schoc	Manual	Yes	0	0-1 Miles	Europe	66	Old	Yes						
47	23986 Married	Female	\$20,00,000	1	Bachelors	Clerical	Yes	0	0-1 Miles	Europe	46	Middle Age	Yes						
48	24466 Married	Female	\$60,00,000	0	Partial Col	Skilled Manual	Yes	1	5-10 Miles	Pacific	52	Middle Age	Yes						
49	29097 Single	Female	\$40,00,000	2	Partial Col	Skilled Manual	Yes	2	5-10 Miles	Pacific	43	Middle Age	No						

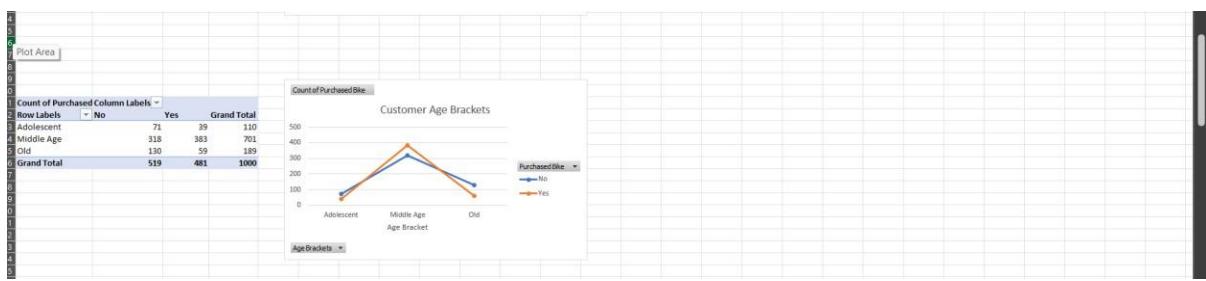
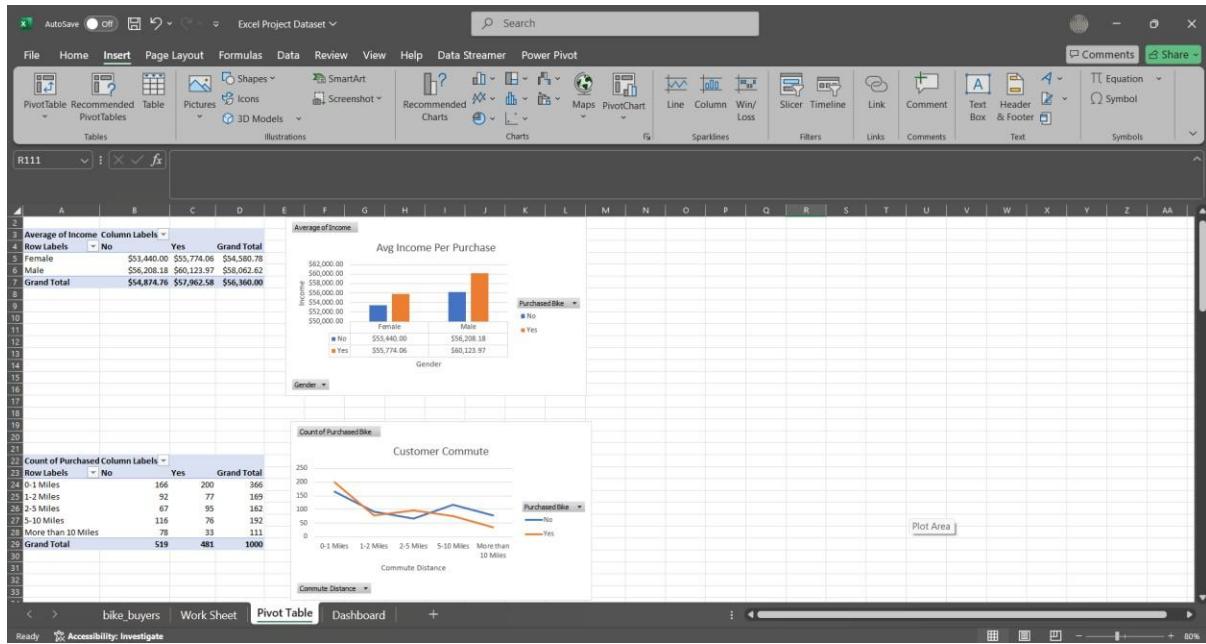
Step 5: After the completion of the data cleaning and pre-processing now we can start to create the pivot table. Make an add of a sheet in the same board to work pivot table in different table.



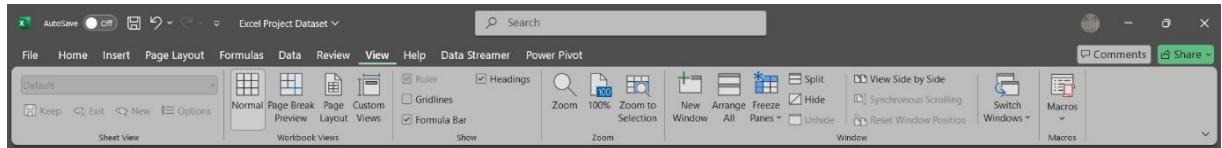
Step 6: Select the pivot table and in the down, you can see the dataset select that sheet and give **ctrl+a** to work on the pivot table.

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.

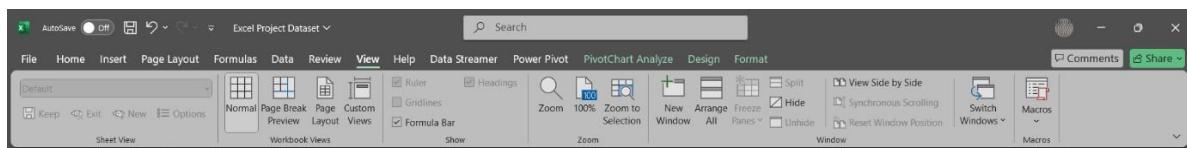
Step 8: Using recommended charts we can visualize the total value which we taken in pivot table.



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.