4. Laboratory Experiment list

Sr. No	**Title**
	Prerequisite practical assignments or installation (if any)
	Companion Course: Elective V (417531), Elective VI (417532)
	List of Assignments
1	Import Data from different Sources such as (Excel, Sql Server, Oracle etc.) and load in targeted systems.
2	Data Visualization from Extraction Transformation and Loading (ETL) Process
3	Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI.
4	Data Analysis and Visualization using Advanced Excel.
5	Perform the data classification algorithm using any Classification algorithm
6	Perform the data clustering algorithm using any Clustering algorithm
	Content Beyond Syllabus
1	
2	

4.1 Experiment No. 1

Aim:

Practical 1: Import the legacy data from different sources such as (Excel, SqlServer, Oracle etc.) and load in the target system.

Objective:

In this experiment, we will be able to

- Gather data from various sources and load it in the targeted system, Data Extraction and Loading operation.

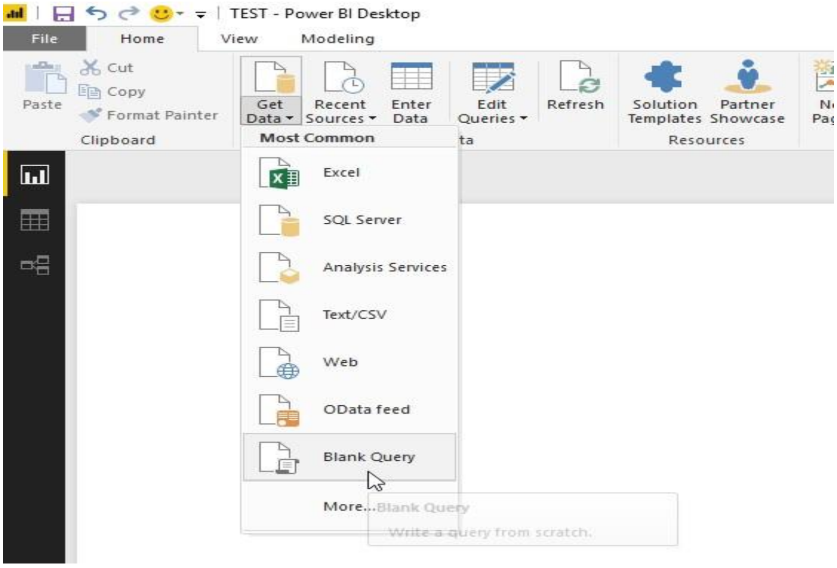
Theory:

Importing Excel Data and loading in BI Tool (Power BI)

1) Launch Power BI Desktop.

2) From the Home ribbon, select Get Data.

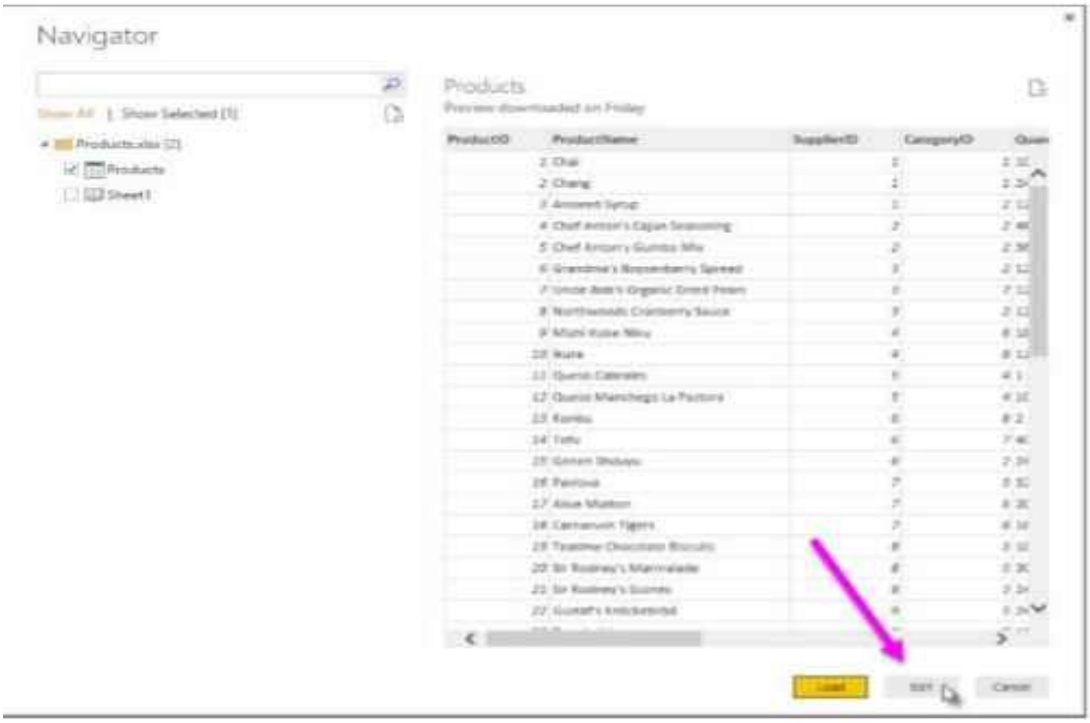
Excel is one of the Most Common data connections, so you can select it directly from the Get Data menu.



3) If you select the Get Data button directly, you can also select File > Excel and select Connect.

4) In the Open File dialog box, select the Products.xlsx file.

5) In the Navigator pane, select the Products table and then select Edit**.**



Navigator

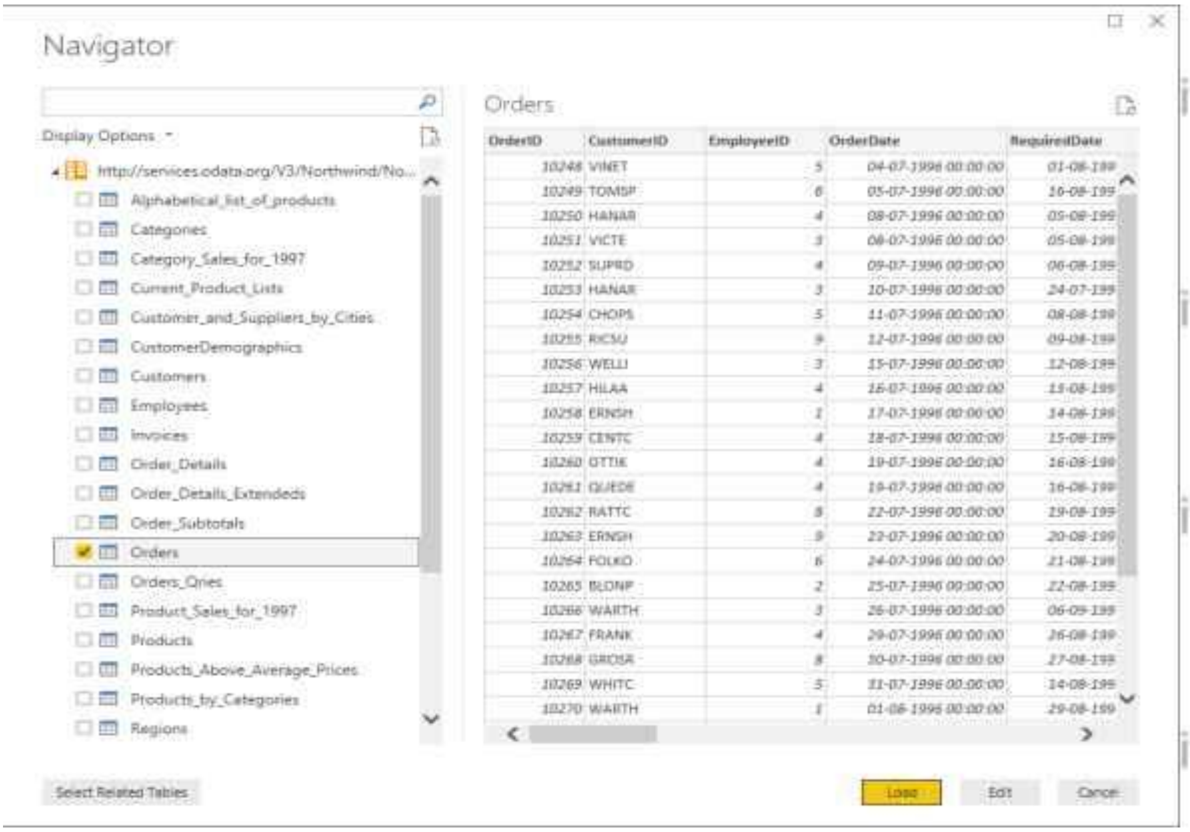
Show All | Show Selected (1)

- Products.xlsx [2]
  - ☑ Products
  - □ Sheet1

Products

Preview downloaded on Friday

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



Navigator

Display Options

- http://services.odata.org/V3/Northwind/No...
  - □ Alphabetical_list_of_products
  - □ Categories
  - □ Category_Sales_for_1997
  - □ Current_Product_Lists
  - □ Customer_and_Suppliers_by_Cities
  - □ CustomerDemographics
  - □ Customers
  - □ Employees
  - □ Invoices
  - □ Order_Details
  - □ Order_Details_Extendeds
  - □ Order_Subtotals
  - ☑ Orders
  - □ Orders_Qries
  - □ Product_Sales_for_1997
  - □ Products
  - □ Products_Above_Average_Prices
  - □ Products_by_Categories
  - □ Regions

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	04-07-1996 00:00:00	01-08-199
10249	TOMSP	6	05-07-1996 00:00:00	16-08-199
10250	HANAR	4	08-07-1996 00:00:00	05-08-199
10251	VICTE	3	08-07-1996 00:00:00	05-08-199
10252	SUPRD	4	09-07-1996 00:00:00	06-08-199
10253	HANAR	3	10-07-1996 00:00:00	24-07-199
10254	CHOPS	5	11-07-1996 00:00:00	08-08-199
10255	RICSU	9	12-07-1996 00:00:00	09-08-199
10256	WELLI	3	15-07-1996 00:00:00	12-08-199
10257	HILAA	4	16-07-1996 00:00:00	13-08-199
10258	ERNSH	1	17-07-1996 00:00:00	14-08-199
10259	CENTC	4	18-07-1996 00:00:00	15-08-199
10260	OTTIK	4	19-07-1996 00:00:00	16-08-199
10261	QUEDE	4	19-07-1996 00:00:00	16-08-199
10262	RATTC	8	22-07-1996 00:00:00	19-08-199
10263	ERNSH	9	23-07-1996 00:00:00	20-08-199
10264	FOLKO	6	24-07-1996 00:00:00	21-08-199
10265	BLONP	2	25-07-1996 00:00:00	22-08-199
10266	WARTH	3	26-07-1996 00:00:00	06-09-199
10267	FRANK	4	29-07-1996 00:00:00	26-08-199
10268	GROSR	8	30-07-1996 00:00:00	27-08-199
10269	WHITC	5	31-07-1996 00:00:00	14-08-199
10270	WARTH	1	01-08-1996 00:00:00	29-08-199

Importing Data from OData Feed

In this task, you'll bring in order data. This step represents connecting to a sales system. You import data into Power BI Desktop from the sample Northwind OData feed at the following

URL, which you can copy (and then paste) in the steps below:

http://services.odata.org/V3/Northwind/Northwind.svc/

Connect to an OData feed:

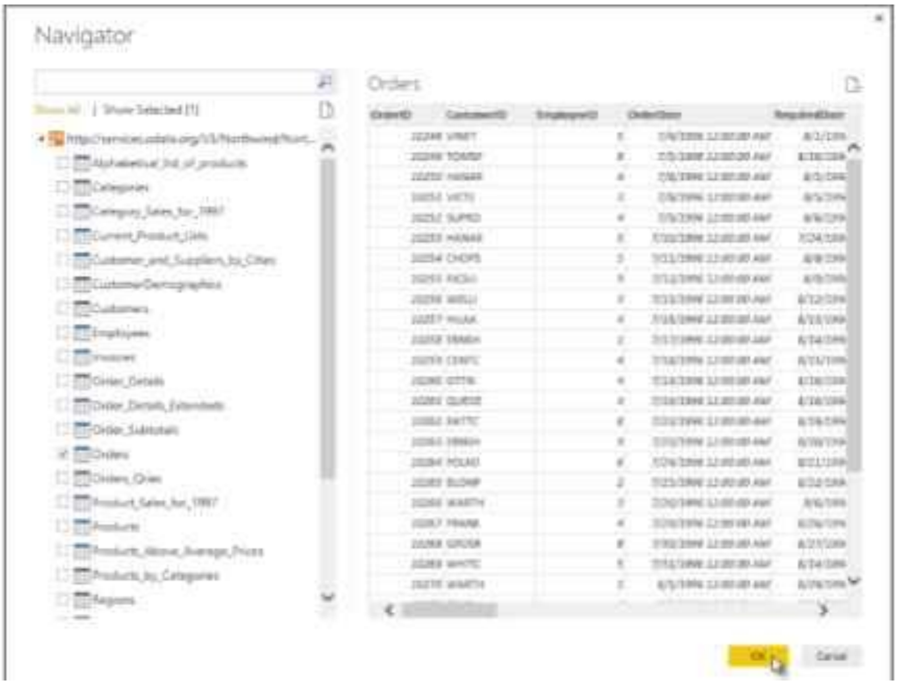
1) From the Home ribbon tab in the Query Editor, select Get Data.

2) Browse to the OData Feed data source.

3) In the OData Feed dialog box, paste the URL for the Northwind OData feed.

4) Select OK.

5) In the Navigator pane, select the Orders table, and then select Edit.



Note - You can click a table name, without selecting the checkbox, to see a preview

Conclusion: In This way we have studied data loading operation from various sources into power BI.

Outcome: Successfully able to Import the legacy data from different sources such as (Excel, SqlServer, Oracle etc.) and load in the targeted system

Application:

Data Integration from Multiple Sources

ETL can be employed to extract data from these diverse sources such as different databases, spreadsheets, and external APIs., transform it into a unified format, and load it into the SQL Server database. This integration ensures a consolidated and consistent dataset for analysis in Power BI.

Questions:

Q 1: How do you ensure data integrity and consistency during the import process from different sources?

Q 2: What strategies would you employ to handle large volumes of data during the import process from various sources?

Q 3: Can you describe the role of data mapping and transformation in the context of importing data from different sources into the target system?

Q 4: What security measures would you implement to safeguard the integrity and confidentiality of the imported data during the transfer process?

4.2 *Experiment No. 2*

Aim: Data Visualization from Extraction Transformation and Loading (ETL) Process

Objective: In this experiment, we will be able to transform raw data into valuable insights, thereby empowering businesses with the tools needed for data-driven decision-making.

Theory:

A data representation tool is the user interface of the whole business intelligence system. Before it can be used for creating visuals, the data goes through a long process. This is basically a description of how BI works

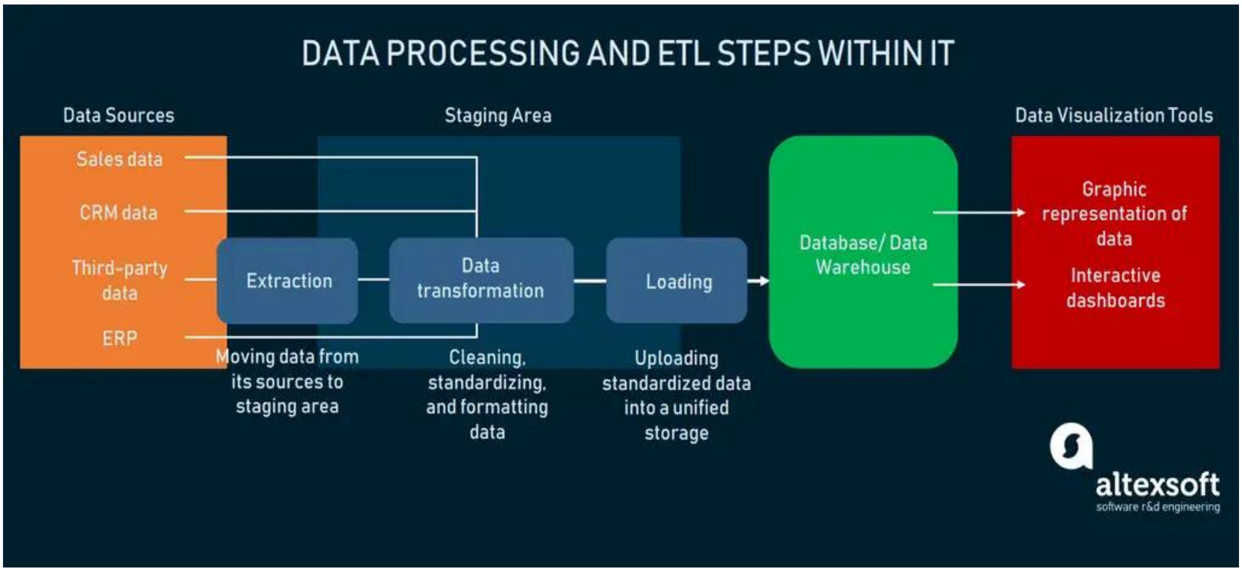
First things first, we should define data sources and data types that will be used.

Then transformation methods and database qualities are determined.

Following that, the data is sourced from its initial storages, for example, Google Analytics, ERP, CRM, or SCM system.

Using API channels, the data is moved to a staging area where it is transformed. Transformation assumes data cleaning, mapping, and standardizing to a unified format.

Further, cleaned data can be moved into a storage: a usual database or data warehouse. To make it possible for the tools to read data, the original base language of datasets can also be rewritten.


DATA PROCESSING AND ETL STEPS WITHIN IT
Data Sources
Staging Area
Data Visualization Tools
Sales data
CRM data
Third-party data
ERP
Extraction
Data transformation
Loading
Database/ Data Warehouse
Graphic representation of data
Interactive dashboards
Moving data from its sources to staging area
Cleaning, standardizing, and formatting data
Uploading standardized data into a unified storage
altexsoft
software r&d engineering

Data visualization actually takes place in the whole process. Most modern BI interfaces have a wide number of options concerning the choice of how to use data for visuals. In most cases, there is a command dashboard with a drag-and-drop interface that allows to:

Connect the data source to the system via API (or custom integration)
Choose the dataset to work with
Choose the type of visualization
Place multiple visuals on the dashboard
Create interactive elements to manipulate the data
Modify visuals as the data updates
Type information manually
Save reports
Share reports

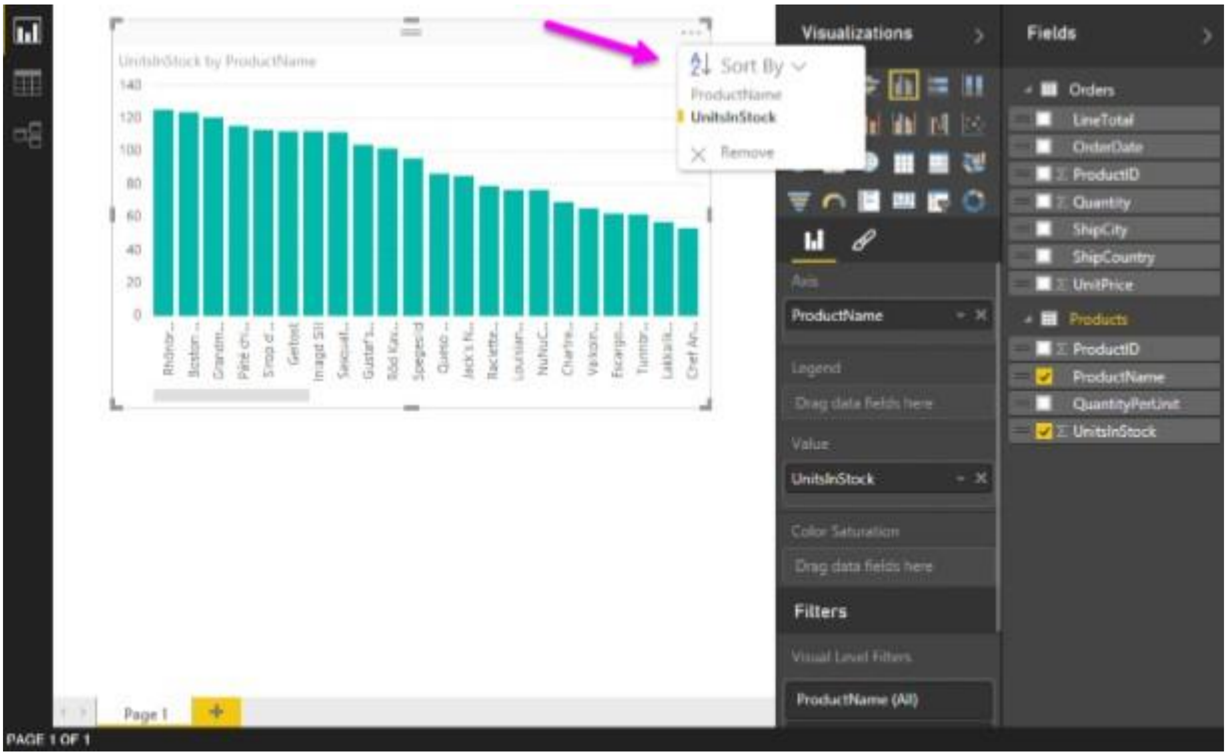
Data Visualization from ETL Process

Power BI Desktop lets you create a variety of visualizations to gain insights from your data. You can build reports with multiple pages and each page can have multiple visuals. You can interact with your visualizations to help analyze and understand your data

In this task, you create a report based on the data previously loaded. You use the Fields pane to select the columns from which you create the visualizations.

Step 1: Create charts showing Units in Stock by Product and Total Sales by Year

1. Drag UnitsInStock from the Field pane (the Fields pane is along the right of the screen) onto a blank space on the canvas. A Table visualization is created. Next, drag ProductName to the Axis box, found in the bottom half of the Visualizations pane. Then we then select Sort By > UnitsInStock using the skittles in the top right corner of the visualization.

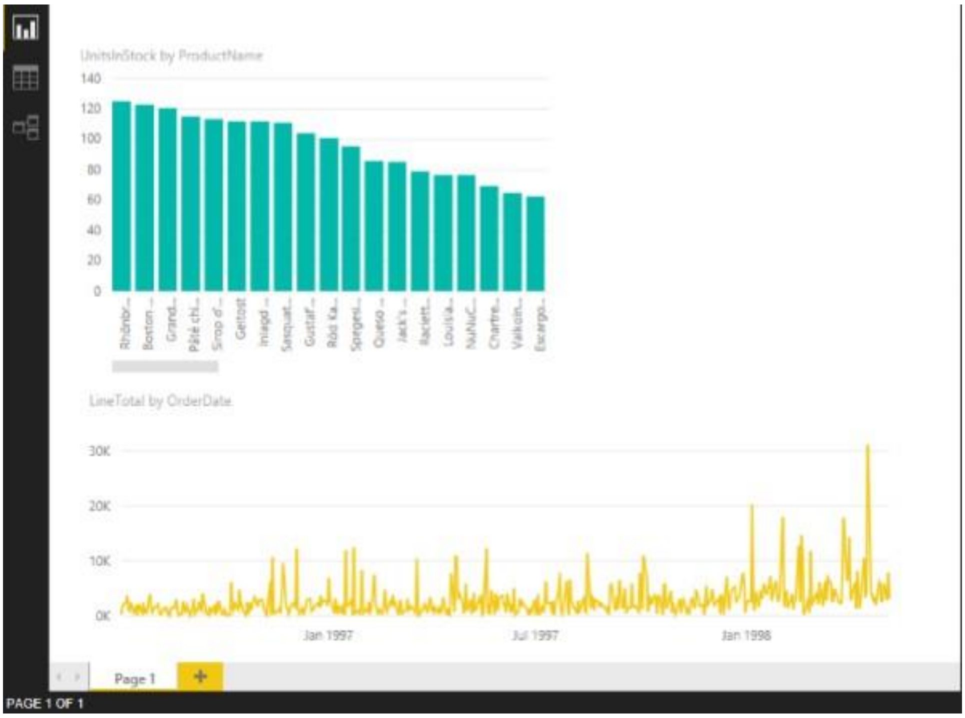


Axis: ProductName

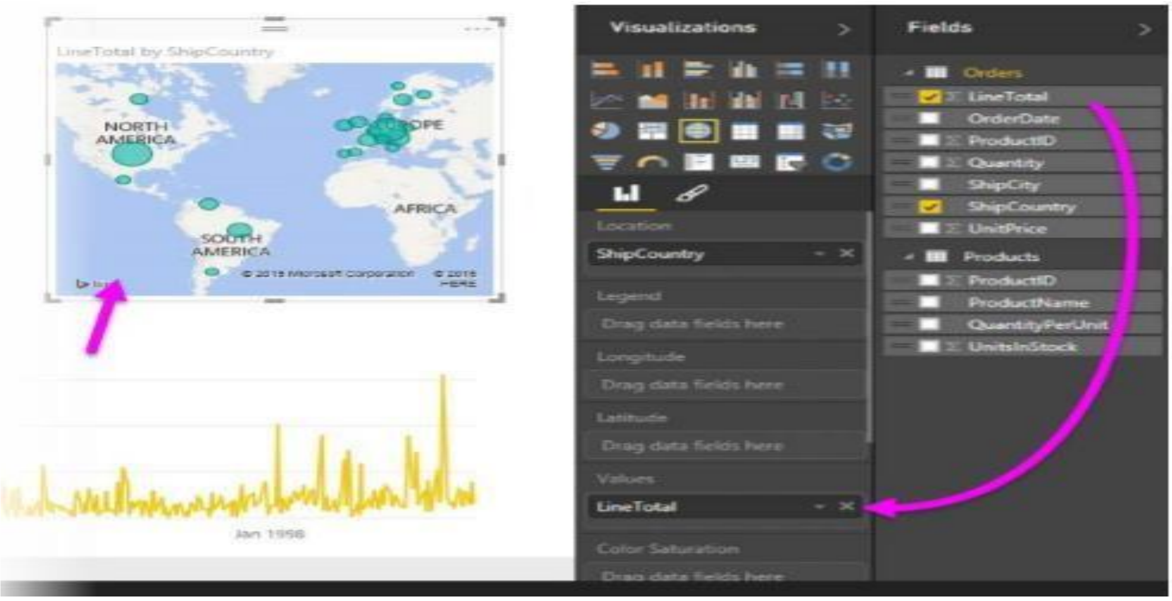
Value: UnitsInStock

ProductName (All)

2. Drag OrderDate to the canvas beneath the first chart, then drag LineTotal (again, from the Fields pane) onto the visual, then select Line Chart. The following visualization is created.



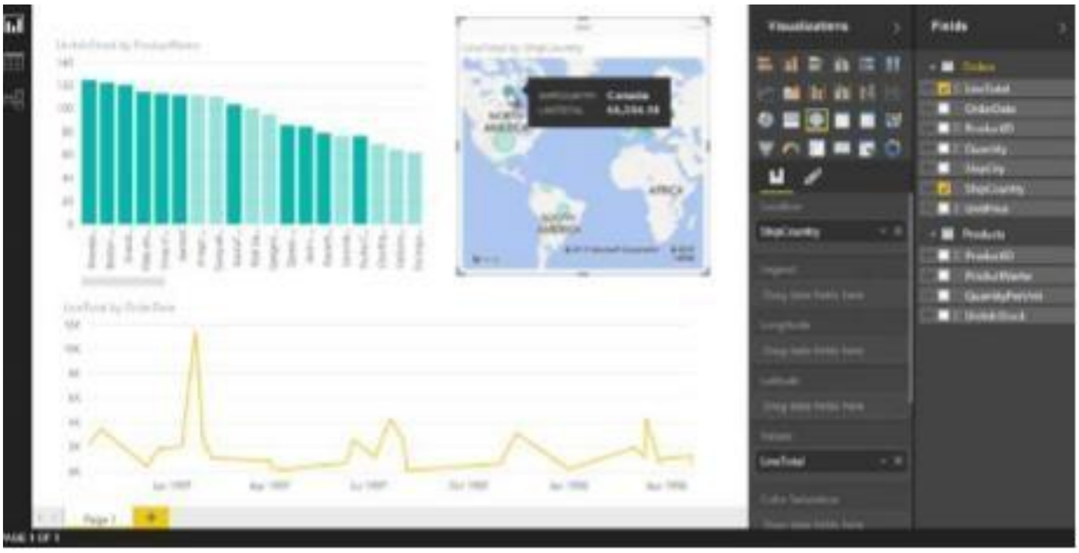
3. Next, drag ShipCountry to a space on the canvas in the top right. Because you selected a geographic field, a map was created automatically. Now drag LineTotal to the Values field; the circles on the map for each country are now relative in size to the LineTotal for orders shipped to that country.



Step 2: Interact with your report visuals to analyze further

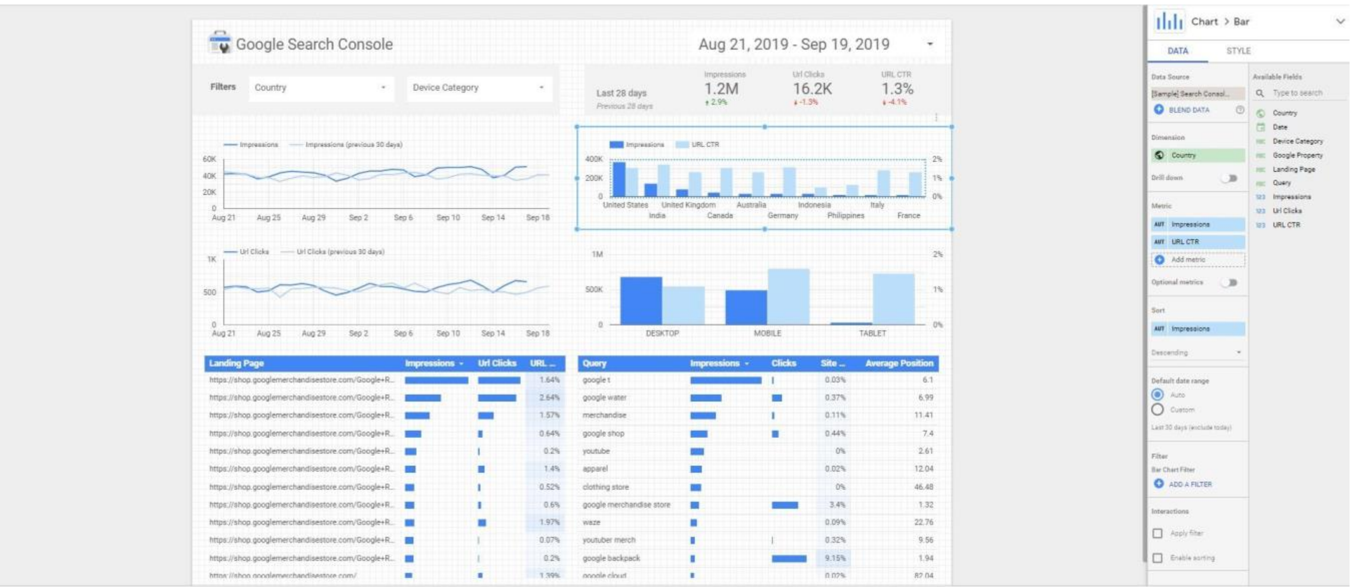
Power BI Desktop lets you interact with visuals that cross-highlight and filter each other to uncover further trends.

1. Click on the light blue circle centered in Canada. Note how the other visuals are filtered to show Stock (ShipCountry) and Total Orders (LineTotal) just for Canada.



Input: Students can use different dataset

Output: Create your own Dashboard



Google Search Console

Aug 21, 2019 - Sep 19, 2019

Filters: Country, Device Category

Last 28 days Previous 28 days	Impressions	Url Clicks	URL CTR
	1.2M ↑ 2.9%	16.2K ↓ -1.3%	1.3% ↓ -4.1%

Landing Page	Impressions ▾	Url Clicks	URL ...
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	1.64%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	2.64%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	1.57%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	0.64%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	0.2%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	1.4%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	0.52%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	0.6%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	1.97%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	0.07%
https://shop.googlemerchandisestore.com/Google+R...	▬	▬	0.2%
https://shop.googlemerchandisestore.com/	▬	▬	1.39%

Query	Impressions ▾	Clicks	Site ...	Average Position
google t	▬	▬	0.03%	6.1
google water	▬	▬	0.37%	6.99
merchandise	▬	▬	0.11%	11.41
google shop	▬	▬	0.44%	7.4
youtube	▬		0%	2.61
apparel	▬		0.02%	12.04
clothing store	▬		0%	46.48
google merchandise store	▬	▬	3.4%	1.32
waze	▬		0.09%	22.76
youtuber merch	▬	▬	0.32%	9.56
google backpack	▬	▬	9.15%	1.94
google cloud	▬		0.02%	82.04

Chart > Bar

DATA | STYLE

Data Source: [Sample] Search Consol...

BLEND DATA

Dimension: Country

Drill down

Metric: AUT Impressions; AUT URL CTR

Add metric

Optional metrics

Sort: AUT Impressions

Descending

Default date range: ● Auto ○ Custom

Last 30 days (exclude today)

Filter

Bar Chart Filter

ADD A FILTER

Interactions

□ Apply filter

□ Enable sorting

Available Fields: Type to search; Country; Date; Device Category; Google Property; Landing Page; Query; Impressions; Url Clicks; URL CTR

Applications:

Power BI Data Modeling and Visualization:

Scenario: Constructing meaningful data models in Power BI for insightful visualizations and reports.

Application: The ETL process shapes the data in a way that aligns with Power BI data models. Relationships between tables are established, and calculations may be pre-aggregated during the ETL to enhance Power BI performance. This ensures that the data is ready for effective reporting and analysis.

Conclusion:

Summarize the overall process, emphasizing the effectiveness of Power BI in turning raw data into compelling visualizations and actionable insights.

Questions:

1. How can ETL tools improve data visualization?

2. What are some best practices for using ETL tools?

3. ETL vs ELT

4.What is data visualization: how it works, types of data to visualize, visualization formats

4.3 *Experiment No. 3*

Aim: Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver / Power BI.

Objective: In this experiment, Student's will be able to execute the Extraction, Transformation, and Loading (ETL) process to construct a robust and optimized database in SQL Server, with a subsequent integration into Power BI.

Successfully achieving these objectives, the ETL process will result in a well-constructed and optimized database in SQL Server, seamlessly integrated with Power BI for effective data visualization and reporting.

Theory:

Step 1 : Data Extraction : The data extraction is the first step of ETL. There are two Types of Data Extraction

1. Full Extraction : All the data from source systems or operational systems gets extracted to the staging area. (Initial Load)

2. Partial Extraction : Sometimes we get notification from the source system to update a specific date. It is called Delta load. Source System Performance: The Extraction strategies should not affect source system performance.

Step 2 : Data Transformation : The data transformation is the second step.After extracting the data there is a big need to do the transformation as per the target system.

Data Transformation.

Data Extracted from the source system is in Raw format.

We need to transform it before loading into the target server.

Data has to be cleaned, mapped and transformed

There are following important steps of Data Transformation :

1. Selection : Select data to load in target

2. Matching : Match the data with target system

3. Data Transforming : We need to change data as per target table structures

Real life examples of Data Transformation :

- Standardizing data : Data is fetched from multiple sources so it needs to be standardized as per the target system.
- Character set conversion : Need to transform the character sets as per the target systems. (Firstname and last name example)
- Calculated and derived values: In the source system there is first val and second val and in target we need the calculation of first val and second val.
- Data Conversion in different formats : If in source system data is in DDMMYY format and in target the date is in DDMONYYYY format then this transformation needs to be done at transformation phase.

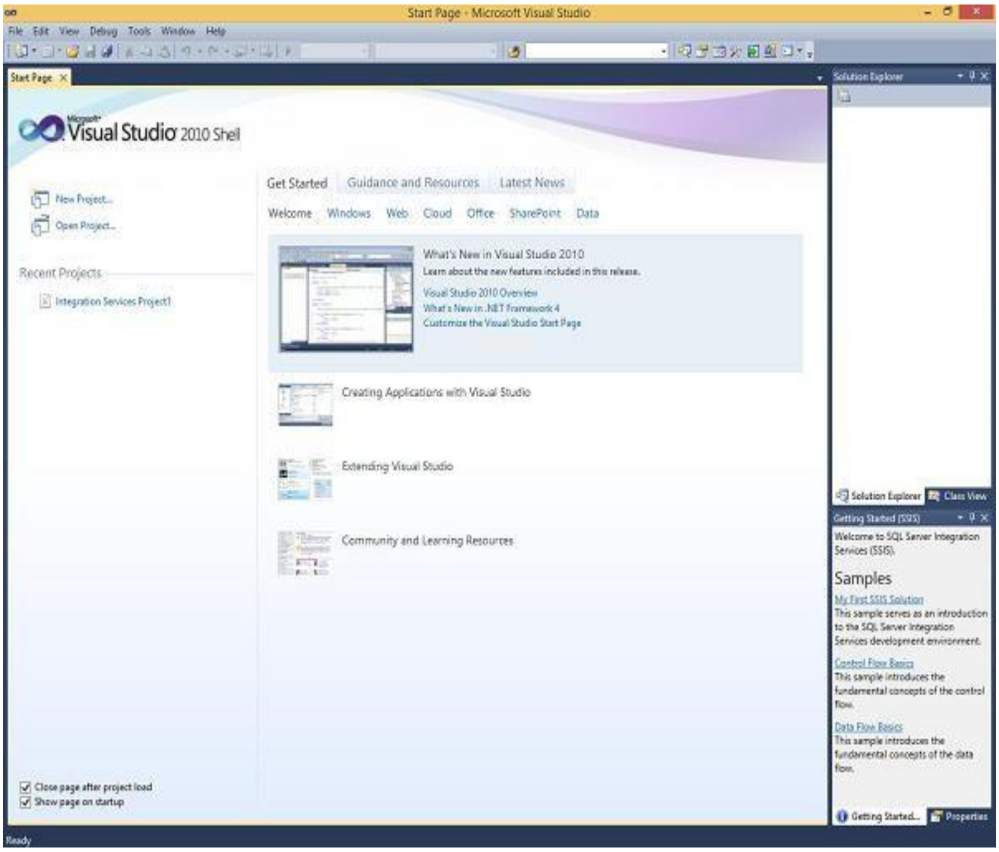
Step 3 : Data Loading

Data loading phase loads the prepared data from staging tables to main tables

ETL process in SQL Server:

Following are the steps to open BIDS\SSDT.

Step 1 − Open either BIDS\SSDT based on the version from the Microsoft SQL Server programs group. The following screen appears**.**



What's New in Visual Studio 2010

Learn about the new features included in this release.

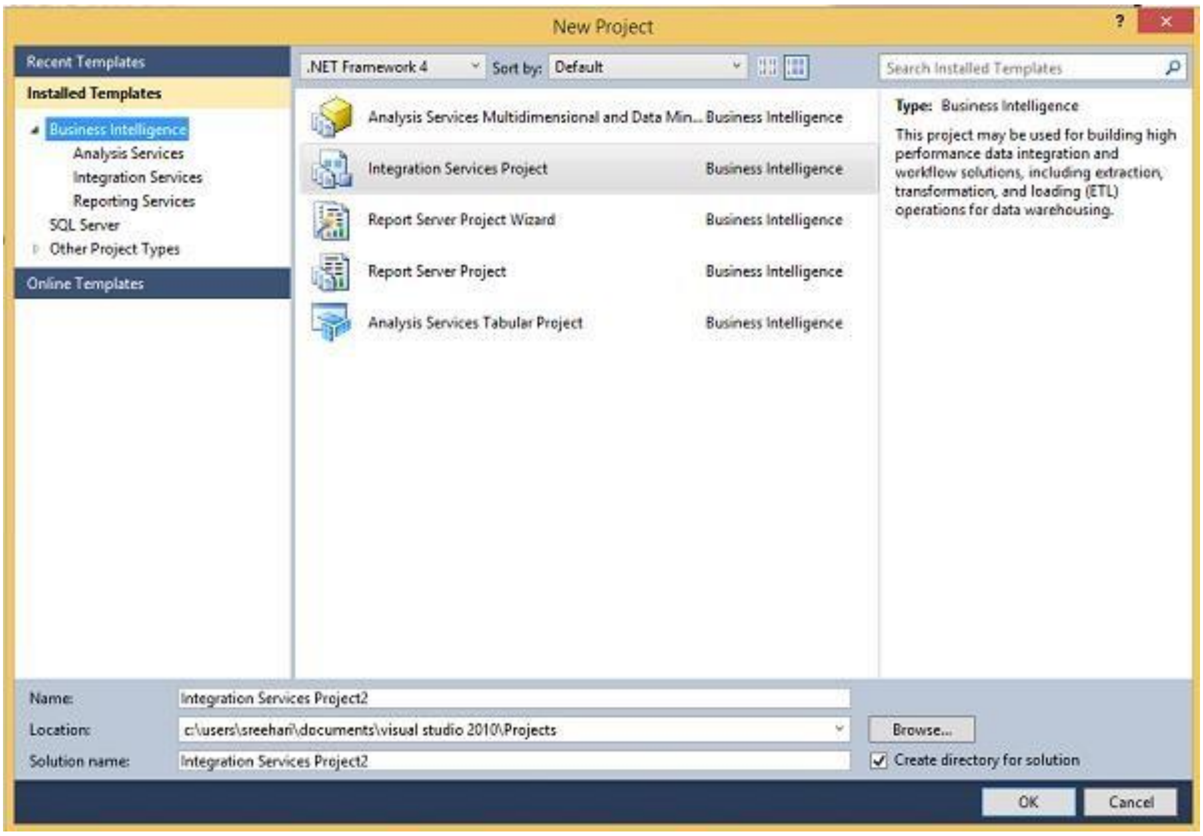
Welcome to SQL Server Integration Services (SSIS).

This sample serves as an introduction to the SQL Server Integration Services development environment.

This sample introduces the fundamental concepts of the control flow.

This sample introduces the fundamental concepts of the data flow.

Step 2 − The above screen shows SSDT has opened. Go to file at the top left corner in the above image and click New. Select project and the following screen opens.


New Project
Recent Templates
Installed Templates
Business Intelligence
Analysis Services
Integration Services
Reporting Services
SQL Server
Other Project Types
Online Templates
.NET Framework 4
Sort by: Default
Search Installed Templates
Analysis Services Multidimensional and Data Min... Business Intelligence
Integration Services Project Business Intelligence
Report Server Project Wizard Business Intelligence
Report Server Project Business Intelligence
Analysis Services Tabular Project Business Intelligence
Browse...
OK
Cancel

Type: Business Intelligence

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

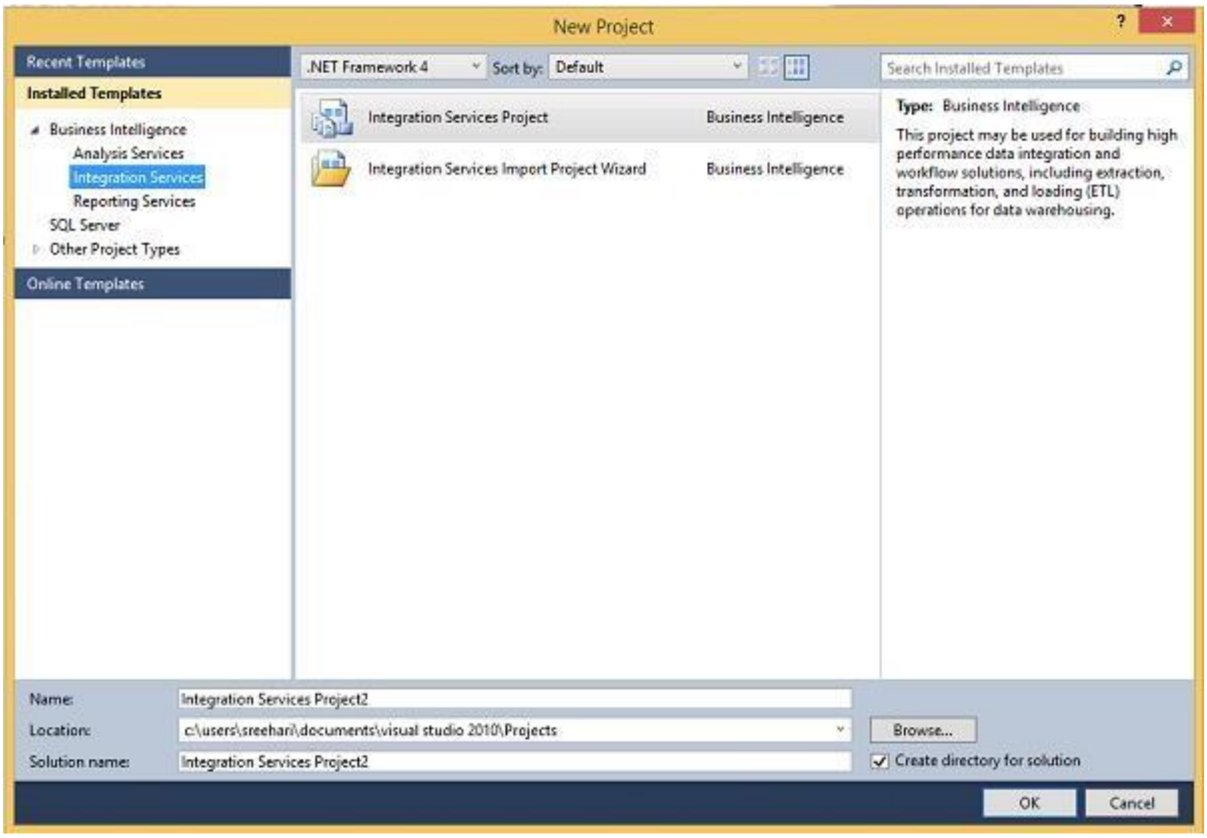
Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution

Step 3 − Select Integration Services under Business Intelligence on the top left corner in the above screen to get the following screen.



Type: Business Intelligence

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution

Step 4 − In the above screen, select either Integration Services Project or Integration Services Import Project Wizard based on your requirement to develop\create the package.

Modes

There are two modes − Native Mode (SQL Server Mode) and SharePoint Mode.

Models

There are two models − Tabular Model (For Team and Personal Analysis) and Multi Dimensions Model (For Corporate Analysis).

The BIDS (Business Intelligence Studio till 2008 R2) and SSDT (SQL Server Data Tools from 2012) are environments to work with SSAS.

Step 1 − Open either BIDS\SSDT based on the version from the Microsoft SQL Server programs group. The following screen will appear.

Step 2 − The above screen shows SSDT has opened. Go to file on the top left corner in the above image and click New. Select project and the following screen opens.

Step 3 − Select Analysis Services in the above screen under Business Intelligence as seen on the top left corner. The following screen pops up.

Step 4 − In the above screen, select any one option from the listed five options based on your requirement to work with Analysis services.

ETL Process in Power BI

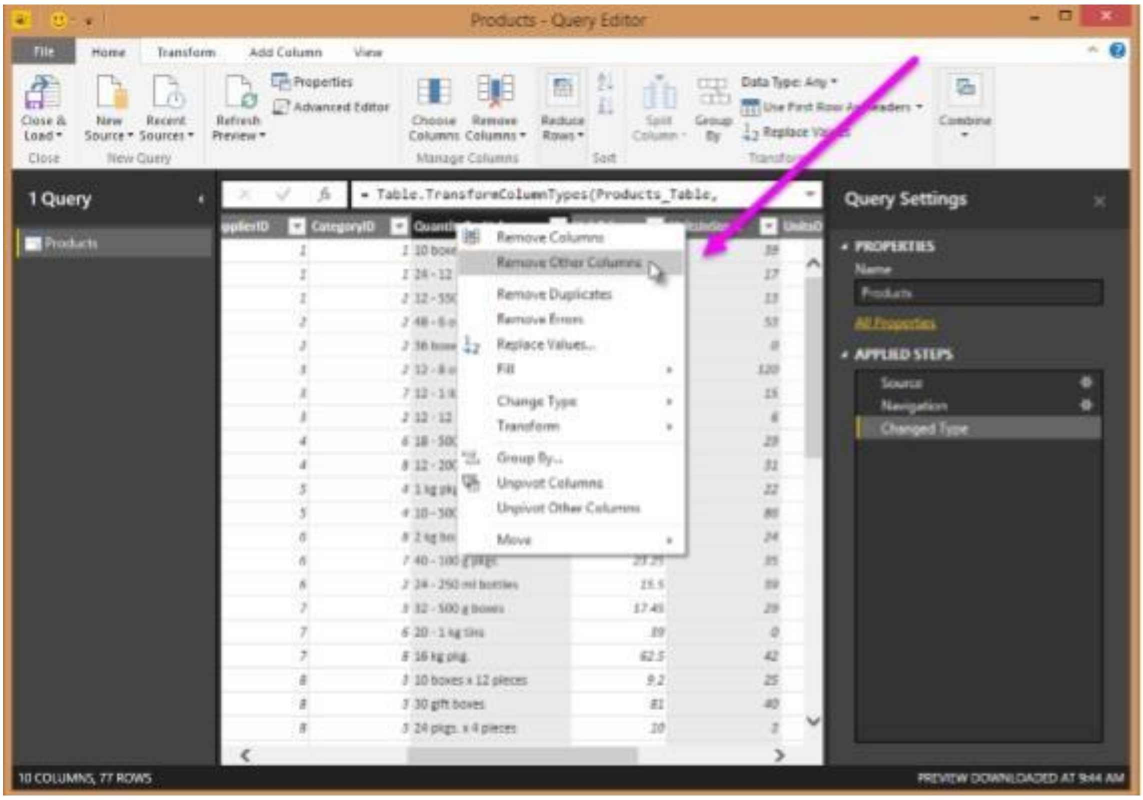
1. **Remove other columns to only display columns of interest**

   In this step you remove all columns except **ProductID, ProductName, UnitsInStock, and QuantityPerUnit** Power BI Desktop includes Query Editor, which is where you shape and transform your data connections. Query Editor opens automatically when you select Edit from Navigator.

   You can also open the Query Editor by selecting Edit Queries from the Home ribbon in Power BI Desktop. The following steps are performed in Query Editor.

   1. In Query Editor, select the **ProductID, ProductName, QuantityPerUnit, and UnitsInStock** columns (use Ctrl+Click to select more than one column, or Shift+Click to select columns that are beside each other).

   2. Select **Remove Columns > Remove** Other Columns from the ribbon, or right-click on a column header and click Remove Other Columns.

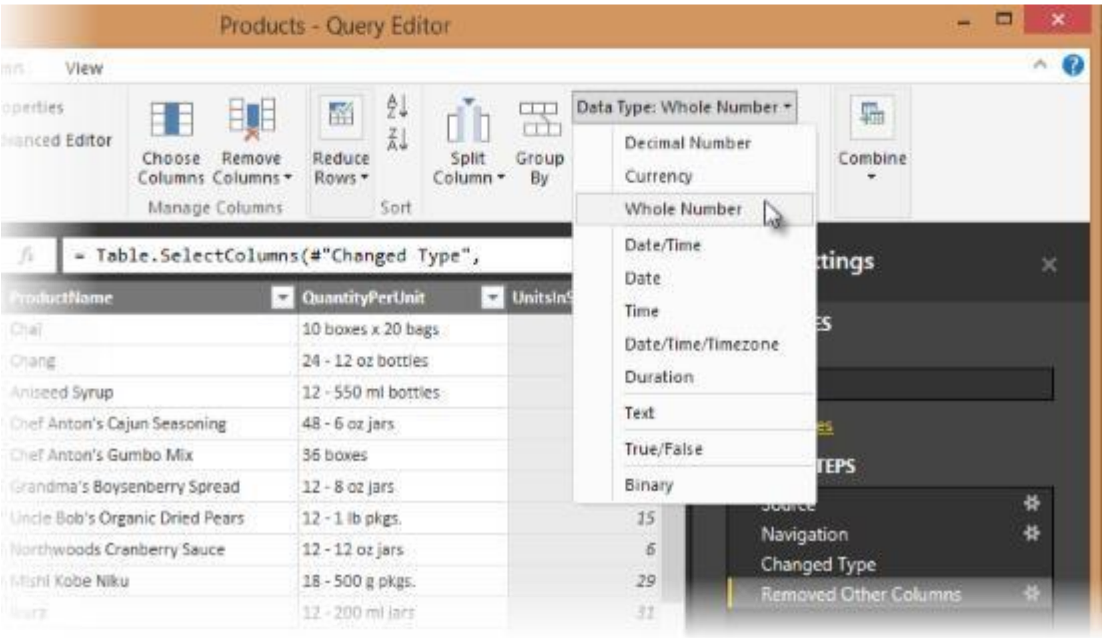


```
= Table.TransformColumnTypes(Products_Table,
```

2. Change the data type of the UnitsInStock column

When the Query Editor connects to data, it reviews each field and determines the best data type. For the Excel workbook, products in stock will always be a whole number, so in this step you confirm the **UnitsInStock** column's datatype is Whole Number.

1. Select the **UnitsInStock** column.
2. Select the **Data Type drop-down button** in the **Home ribbon**.
3. If not already a Whole Number, select **Whole Number** for data type from the drop down (the Data Type: button also displays the data type for the current selection).



```
= Table.SelectColumns(#"Changed Type",
```

ProductName	QuantityPerUnit	UnitsIn
Chai	10 boxes x 20 bags	
Chang	24 - 12 oz bottles	
Aniseed Syrup	12 - 550 ml bottles	
Chef Anton's Cajun Seasoning	48 - 6 oz jars	
Chef Anton's Gumbo Mix	36 boxes	
Grandma's Boysenberry Spread	12 - 8 oz jars	
Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
Northwoods Cranberry Sauce	12 - 12 oz jars	6
Mishi Kobe Niku	18 - 500 g pkgs.	29
Ikura	12 - 200 ml jars	31

3. Expand the Order_Details table

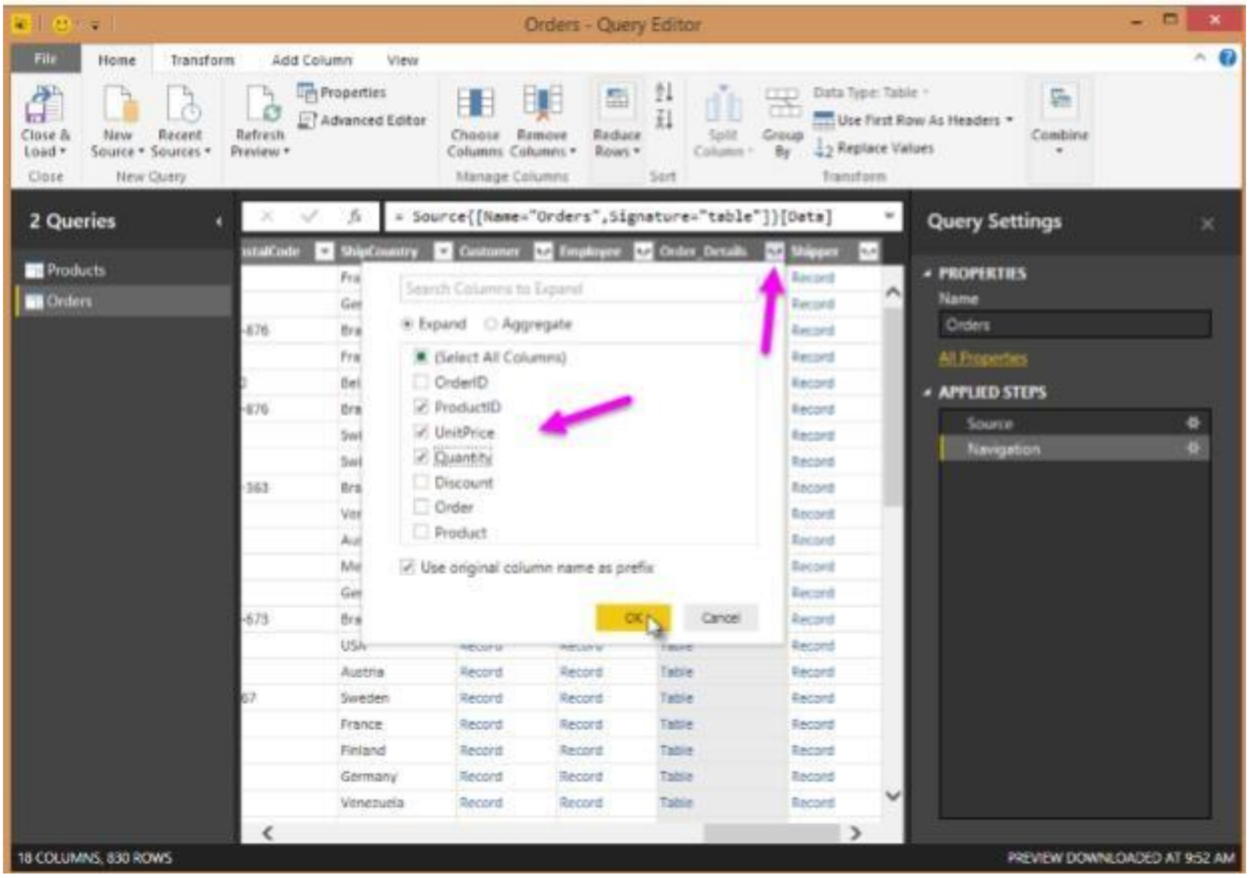
The Orders table contains a reference to a Details table, which contains the individual products that were included in each Order. When you connect to data sources with multiples tables (such as a relational database) you can use these references to build up your query

In this step, you expand the **Order_Details** table that is related to the Orders table, to combine the **ProductID**, **UnitPrice**, and **Quantity** columns from **Order_Details** into the **Orders table**. This is a representation of the data in these tables:

The Expand operation combines columns from a related table into a subject table. When the query runs, rows from the related table (**Order_Details**) are combined into rows from the subject table (**Orders**).

After you expand the Order_Details table, three new columns and additional rows are added to the Orders table, one for each row in the nested or related table.

1. In the Query View, scroll to the Order_Details column.
2. In the Order_Details column, select the expand icon ( ).
3. In the Expand drop-down:
    1. Select (Select All Columns) to clear all columns.
    2. Select ProductID, UnitPrice, and Quantity.
    3. Click OK.



```
= Source{[Name="Orders",Signature="table"]}[Data]
```

Search Columns to Expand

● Expand ○ Aggregate

■ (Select All Columns)
□ OrderID
☑ ProductID
☑ UnitPrice
☑ Quantity
□ Discount
□ Order
□ Product

☑ Use original column name as prefix

OK Cancel

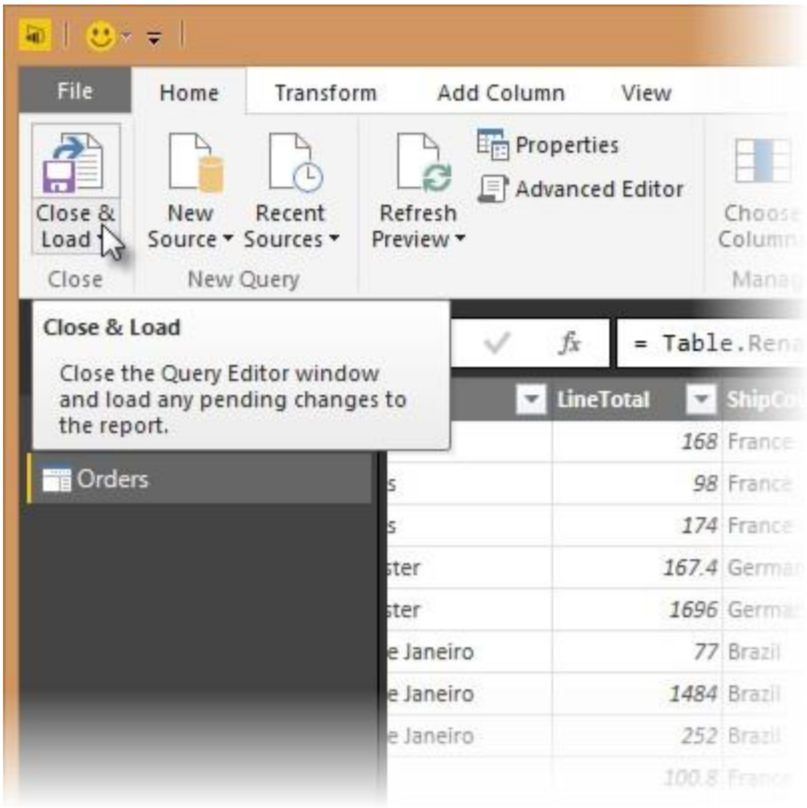
18 COLUMNS, 830 ROWS

4. Calculate the line total for each Order_Details row

Power BI Desktop creates calculations based on the columns you are importing, so you can enrich the data that you connect to. In this step, you create a Custom Column to calculate the line total for each Order_Details row.

Calculate the line total for each Order_Details row:

1. In the Add Column ribbon tab, click Add Custom Column.



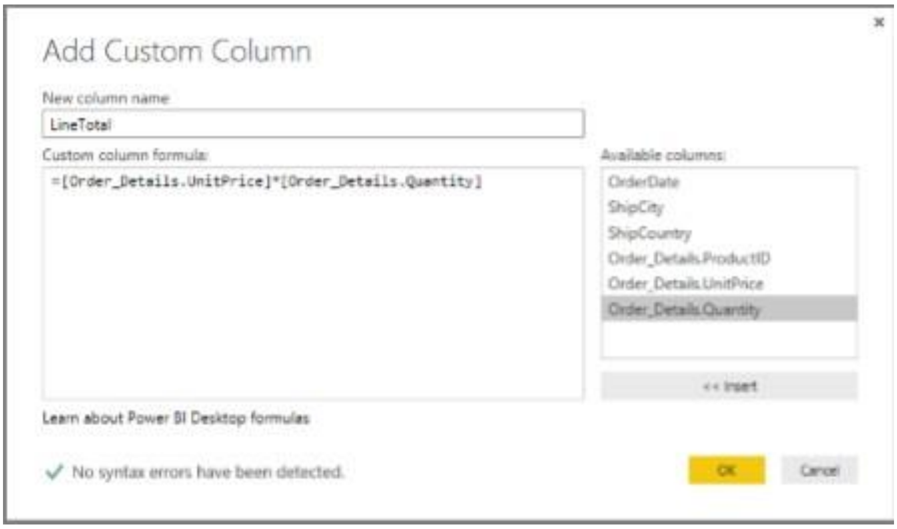
Close & Load

Close the Query Editor window and load any pending changes to the report.

2. In the Add Custom Column dialog box, in the Custom Column Formula textbox, enter [Order_Details.UnitPrice] * [Order_Details.Quantity].

3. In the New column name textbox, enter LineTotal.

4. Click OK.



Add Custom Column

New column name: LineTotal

Custom column formula:

```
=[Order_Details.UnitPrice]*[Order_Details.Quantity]
```

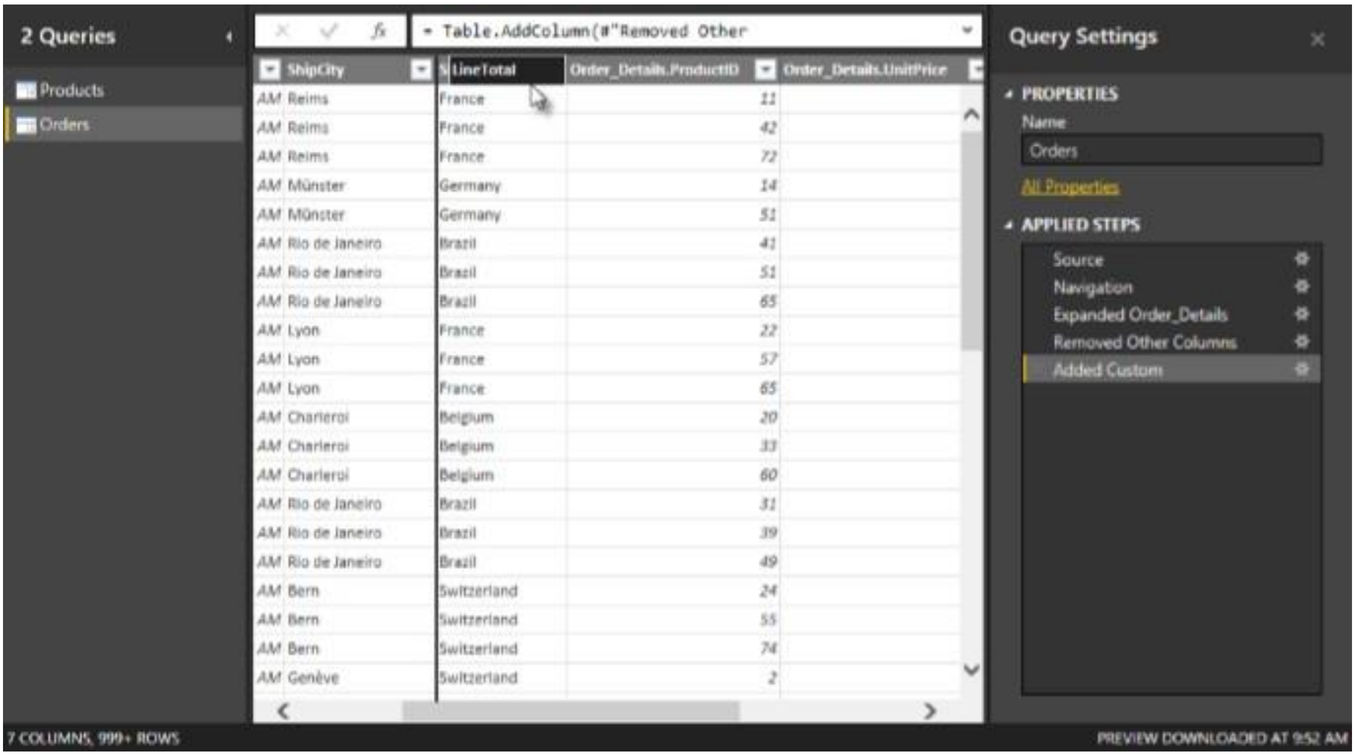
Available columns:

✓ No syntax errors have been detected.

5. Rename and reorder columns in the query

In this step you finish making the model easy to work with when creating reports, by renaming the final columns and changing their order.

1. In Query Editor, drag the LineTotal column to the left, after ShipCountry.



```
= Table.AddColumn(#"Removed Other
```

	ShipCity	LineTotal	Order_Details.ProductID	Order_Details.UnitPrice
AM	Reims	France	11	
AM	Reims	France	42	
AM	Reims	France	72	
AM	Münster	Germany	14	
AM	Münster	Germany	51	
AM	Rio de Janeiro	Brazil	41	
AM	Rio de Janeiro	Brazil	51	
AM	Rio de Janeiro	Brazil	65	
AM	Lyon	France	22	
AM	Lyon	France	57	
AM	Lyon	France	65	
AM	Charleroi	Belgium	20	
AM	Charleroi	Belgium	33	
AM	Charleroi	Belgium	60	
AM	Rio de Janeiro	Brazil	31	
AM	Rio de Janeiro	Brazil	39	
AM	Rio de Janeiro	Brazil	49	
AM	Bern	Switzerland	24	
AM	Bern	Switzerland	55	
AM	Bern	Switzerland	74	
AM	Genève	Switzerland	2	

Name: Orders

Applied steps:
- Source
- Navigation
- Expanded Order_Details
- Removed Other Columns
- Added Custom

7 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 9:52 AM

2. Remove the Order_Details. prefix from the Order_Details.ProductID, Order_Details.UnitPrice and Order_Details.Quantity columns, by double-clicking on each column header, and then deleting that text from the column name.

6. Combine the Products and Total Sales queries

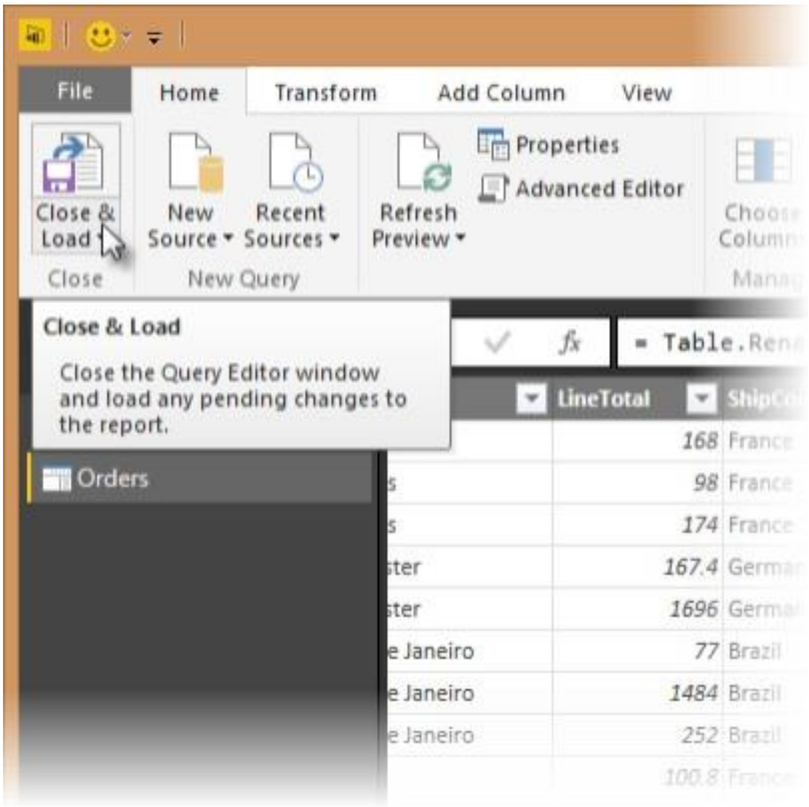
Power BI Desktop does not require you to combine queries to report on them. Instead, you can create Relationships between datasets. These relationships can be created on any column that is common to your datasets

We have Orders and Products data that share a common 'ProductID' field, so we need to ensure there's a relationship between them in the model we're using with Power BI Desktop. Simply specify in Power BI Desktop that the columns from each table are related (i.e. columns that have the same values). Power BI Desktop works out the direction and cardinality of the relationship for you. In some cases, it will even detect the relationships automatically.

In this task, you confirm that a relationship is established in Power BI Desktop between the Products and Total Sales queries

Step 1: Confirm the relationship between Products and Total Sales

1. First, we need to load the model that we created in the Query Editor into Power BI Desktop. From the Home ribbon of the Query Editor, select Close & Load.

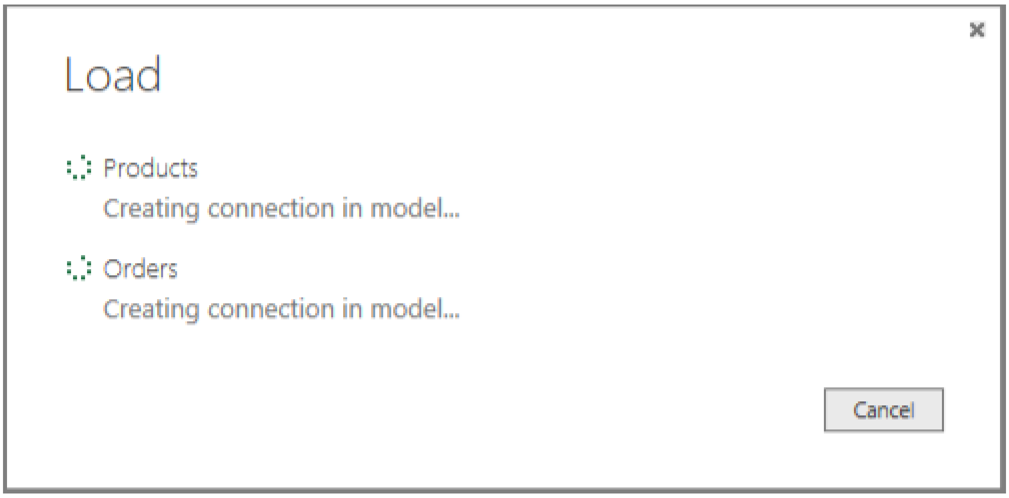


Close & Load

Close the Query Editor window and load any pending changes to the report.

	LineTotal	ShipCo
	168	France
s	98	France
s	174	France
ster	167.4	Germa
ster	1696	Germa
e Janeiro	77	Brazil
e Janeiro	1484	Brazil
e Janeiro	252	Brazil
	100.8	France

2. Power BI Desktop loads the data from the two queries.

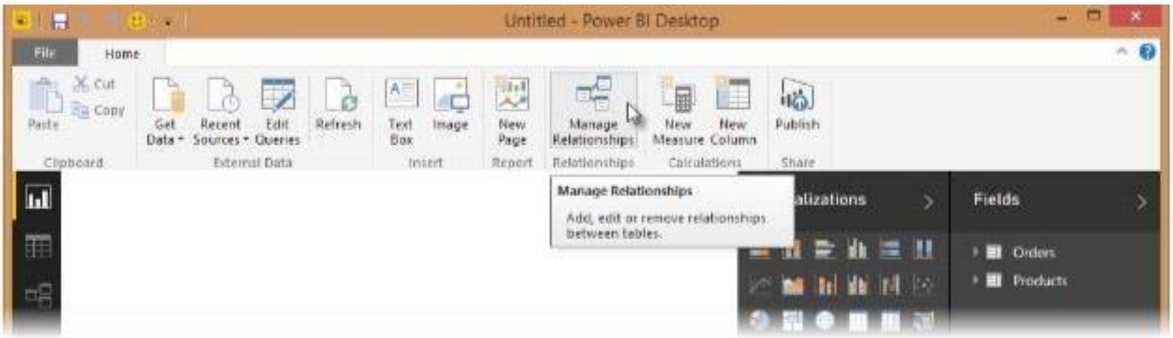


Load

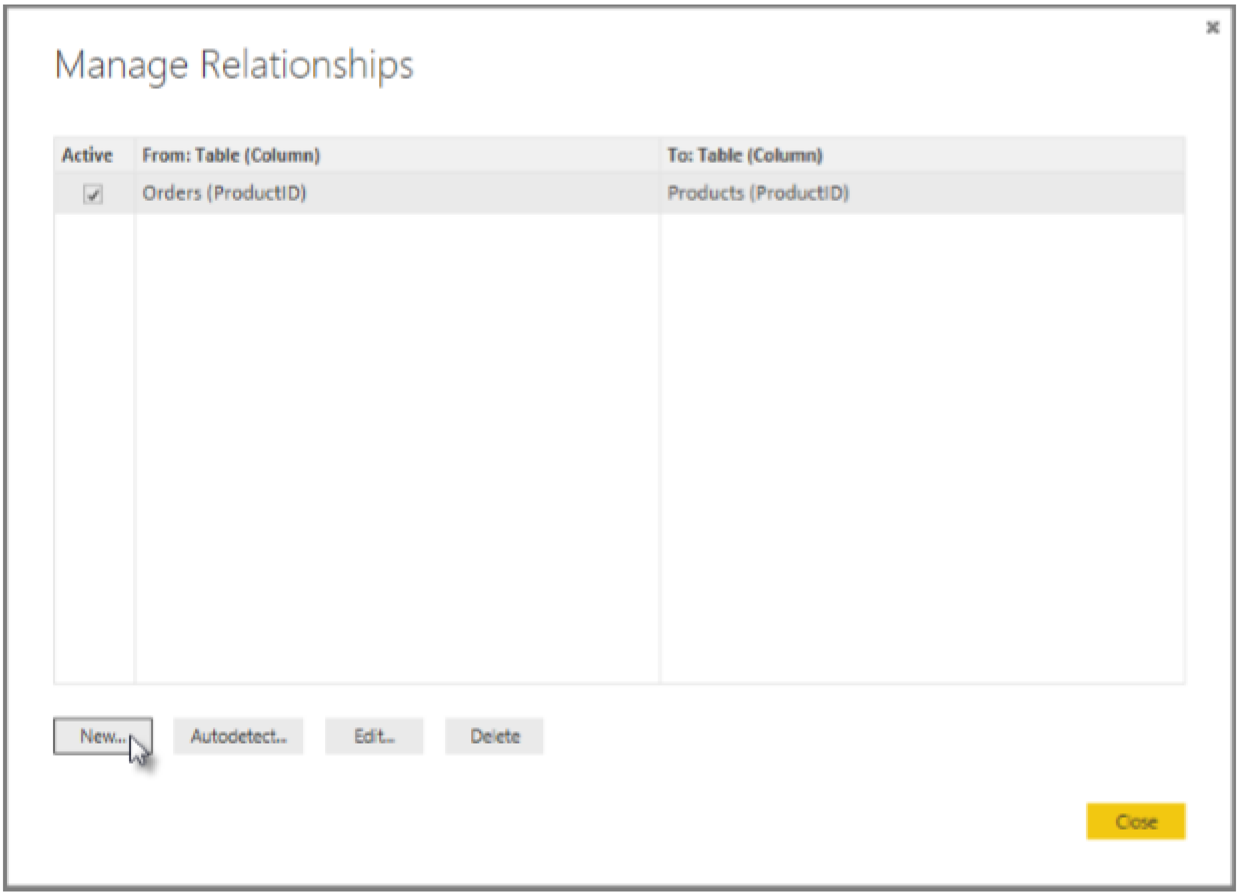
Products
Creating connection in model...

Orders
Creating connection in model...

3. Once the data is loaded, select the Manage Relationships button Home ribbon.



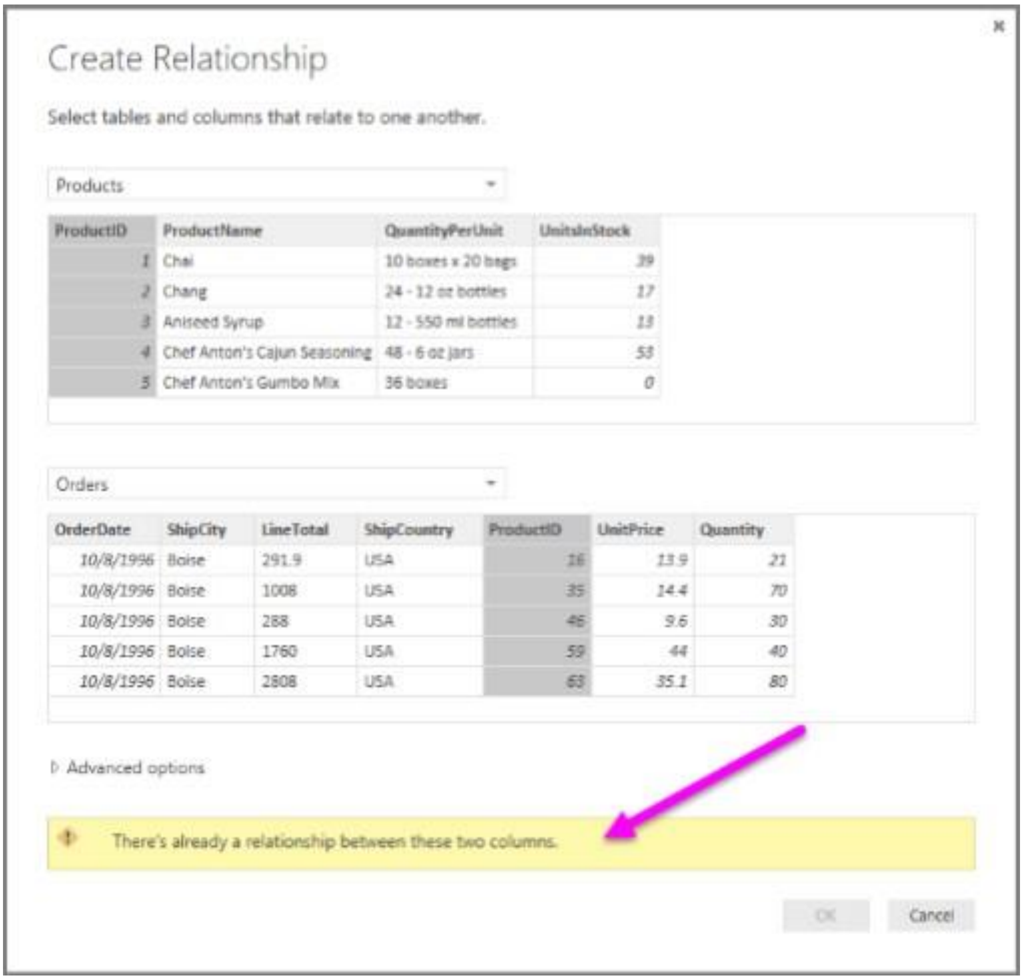
4. Select the New… button



Manage Relationships

Active	From: Table (Column)	To: Table (Column)
☑	Orders (ProductID)	Products (ProductID)

5. When we attempt to create a relationship, we see that one already exists! As shown in the Create Relationship dialog (by the shaded columns), the ProductsID fields in each query already have an established relationship.



Create Relationship

Select tables and columns that relate to one another.

Products

ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	Chef Anton's Gumbo Mix	36 boxes	0

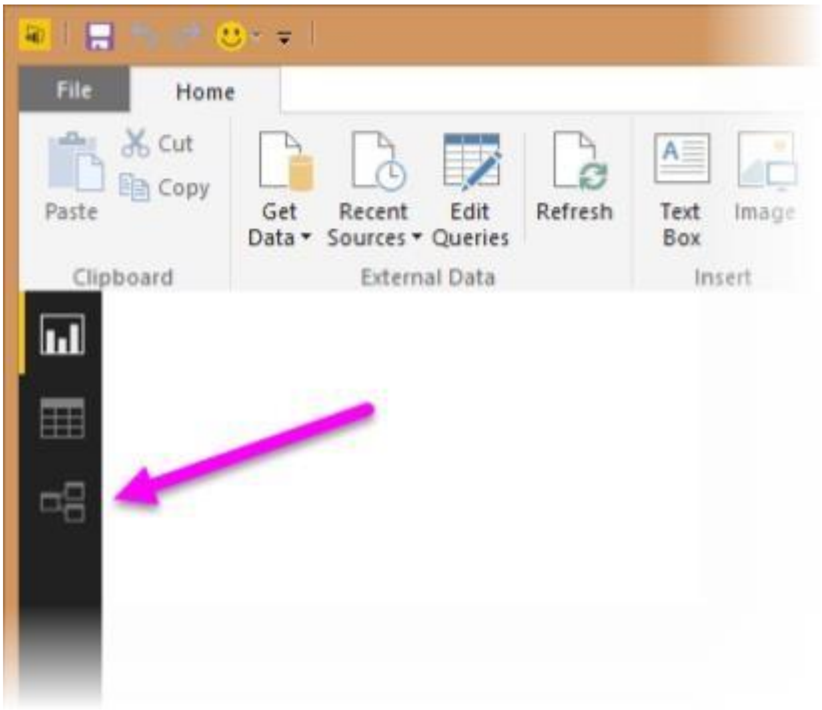
Orders

OrderDate	ShipCity	LineTotal	ShipCountry	ProductID	UnitPrice	Quantity
10/8/1996	Boise	291.9	USA	16	13.9	21
10/8/1996	Boise	1008	USA	35	14.4	70
10/8/1996	Boise	288	USA	46	9.6	30
10/8/1996	Boise	1760	USA	59	44	40
10/8/1996	Boise	2808	USA	63	35.1	80

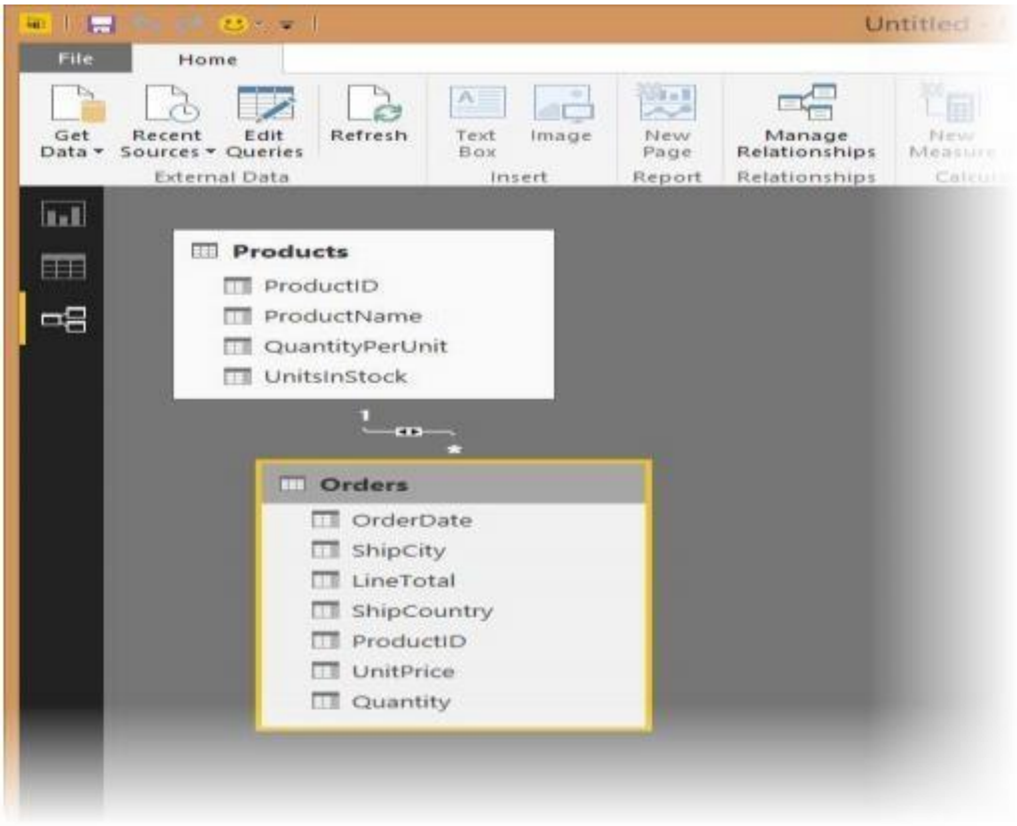
▷ Advanced options

There's already a relationship between these two columns.

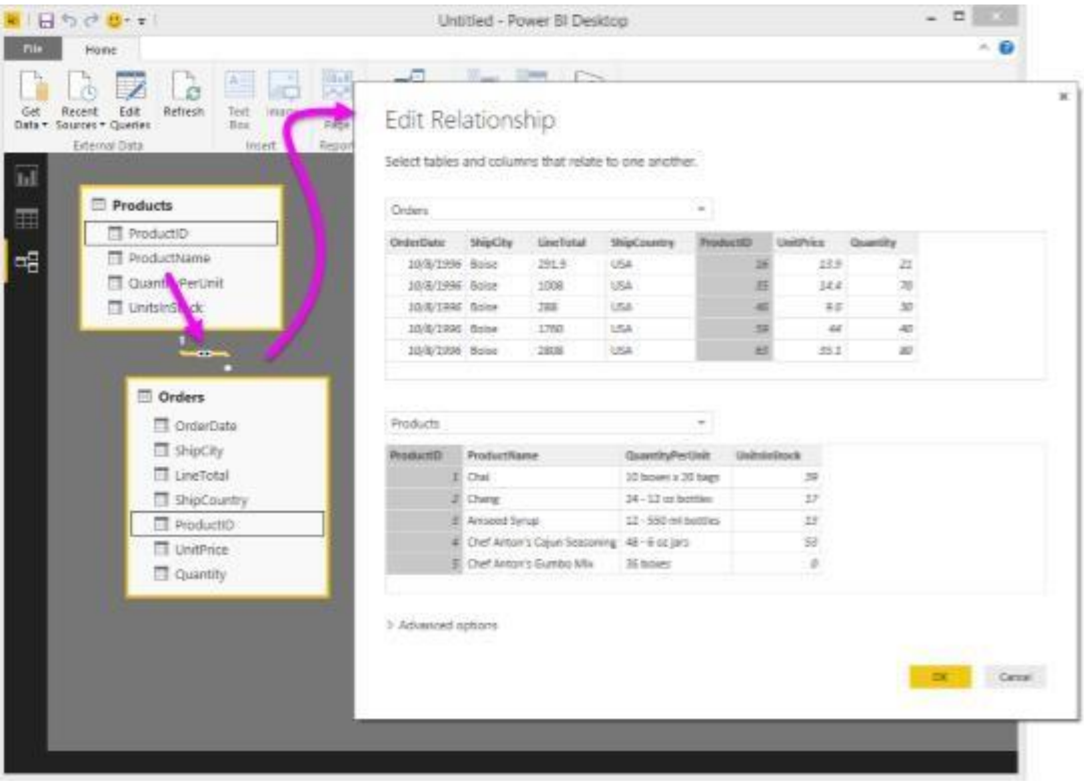
6. Select Cancel, and then select Relationship view in Power BI Desktop.



7. Visualizes the relationship between the queries.



8. When you double-click the arrow on the line that connects the two queries, an Edit Relationship dialog appears.



Edit Relationship

Select tables and columns that relate to one another.

Orders

OrderDate	ShipCity	LineTotal	ShipCountry	ProductID	UnitPrice	Quantity
10/8/1996	Boise	291.9	USA	28	13.9	21
10/8/1996	Boise	1008	USA	35	14.4	70
10/8/1996	Boise	288	USA	40	9.6	30
10/8/1996	Boise	1760	USA	59	44	40
10/8/1996	Boise	2808	USA	65	35.1	80

Products

ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	Chef Anton's Gumbo Mix	36 boxes	0

No need to make any changes, so we'll just select Cancel to close the Edit Relationship dialog.

Applications: The application of the Extraction, Transformation, and Loading (ETL) process to construct a database in SQL Server and integrate it with Power BI is crucial for efficiently managing and analyzing data. following are the applications of ETL:

Data Cleaning and Standardization

Database Construction and Schema Design

Scenario: Setting up a new database or enhancing an existing one to optimize performance and facilitate efficient querying.

Application: ETL plays a crucial role in designing the database schema and populating tables. This involves defining relationships, indexing key columns, and organizing data for faster retrieval. A well-structured database enhances Power BI performance by providing a solid foundation for reporting.

Conclusion: the Extraction, Transformation, and Loading (ETL) process is a fundamental aspect of constructing a robust database in SQL Server and integrating it seamlessly with Power BI for insightful analysis and reporting.

Outcome:

Successful construction of a database in SQL Server or Power BI containing the extracted, transformed, and loaded data, improved data quality and consistency through data cleansing and transformation.

Questions:

1. What are the two types of data extraction methods commonly used in the ETL process, and how do they differ from each other?

2. How does data extraction affect the performance of the source system, and what strategies can be implemented to mitigate any negative impact?

3. What are the key steps involved in the data transformation phase of the ETL process, and why is it essential for preparing data for loading into the target system?

4. Can you provide real-life examples of data transformation tasks and explain how they contribute to ensuring data quality and consistency?

5. What is the purpose of the data loading phase in the ETL process, and how does it facilitate the transfer of prepared data from staging tables to main tables in the target system?

4.4 Experiment No. 4

Aim: Data Analysis and Visualization using Advanced Excel

Objective: To leverage advanced Excel techniques for data analysis and visualization in order to extract actionable insights, facilitate informed decision-making, and enhance organizational performance.

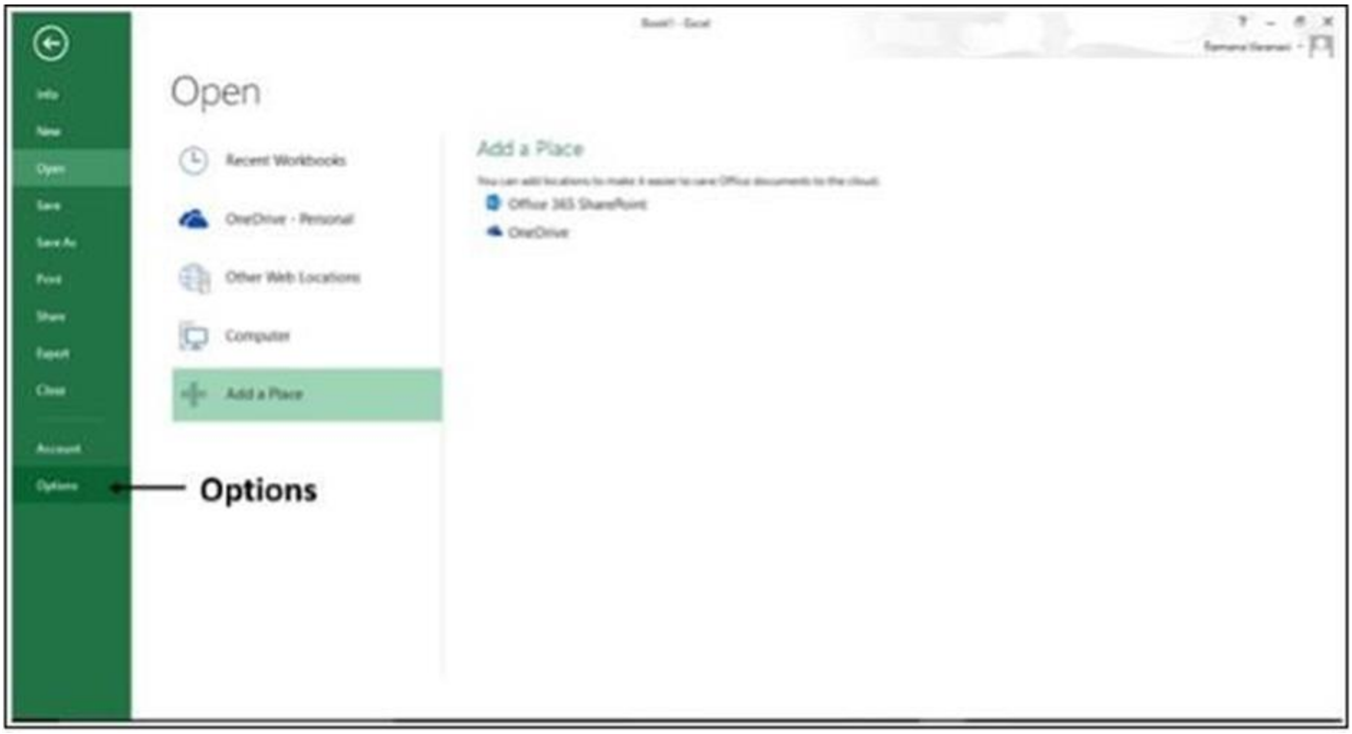
Theory:

Power View is a feature of Microsoft Excel 2013 that enables interactive data exploration, visualization, and presentation encouraging intuitive ad-hoc reporting.

Create a Power View Sheet

Make sure Power View add-in is enabled in Excel 2013.

Step 1 – Click on the File menu and then Click on Options.



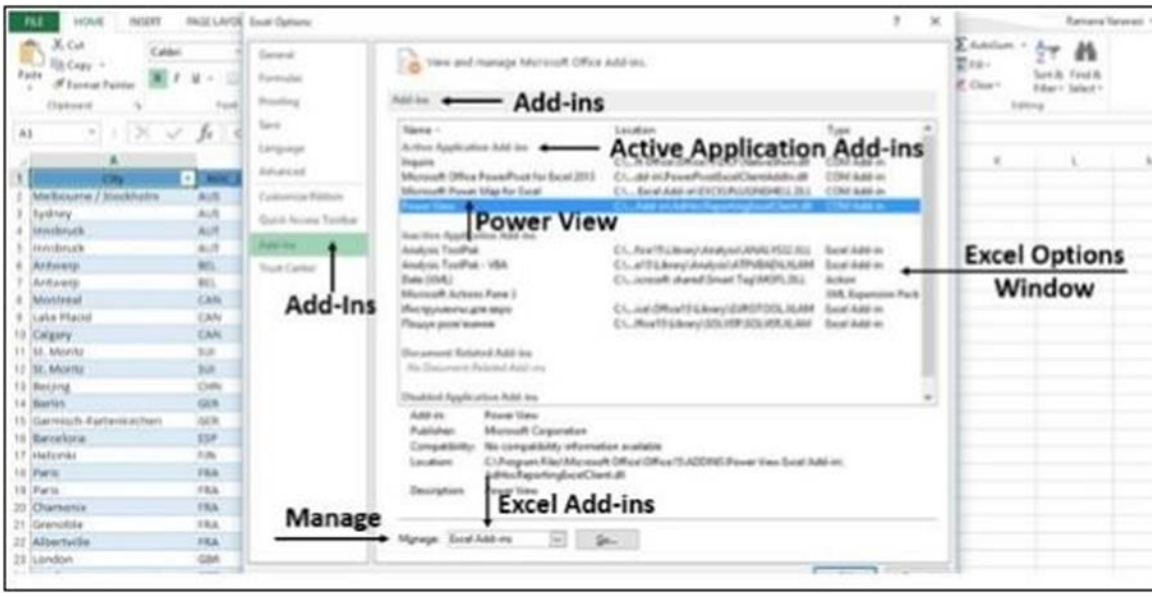
You can add locations to make it easier to save Office documents to the cloud.

The Excel Options window appears.

Step 2 – Click on Add-Ins.

Step 3 – In the Manage box, click the drop-down arrow and select Excel Add-ins.

Step 4 − All the available Add-ins will be displayed. If Power View Add-in is enabled, it appears in Active Application Add-ins.



If it does not appear, follow these steps −

Step 1 − In the Excel Options Window, Click on Add-Ins.

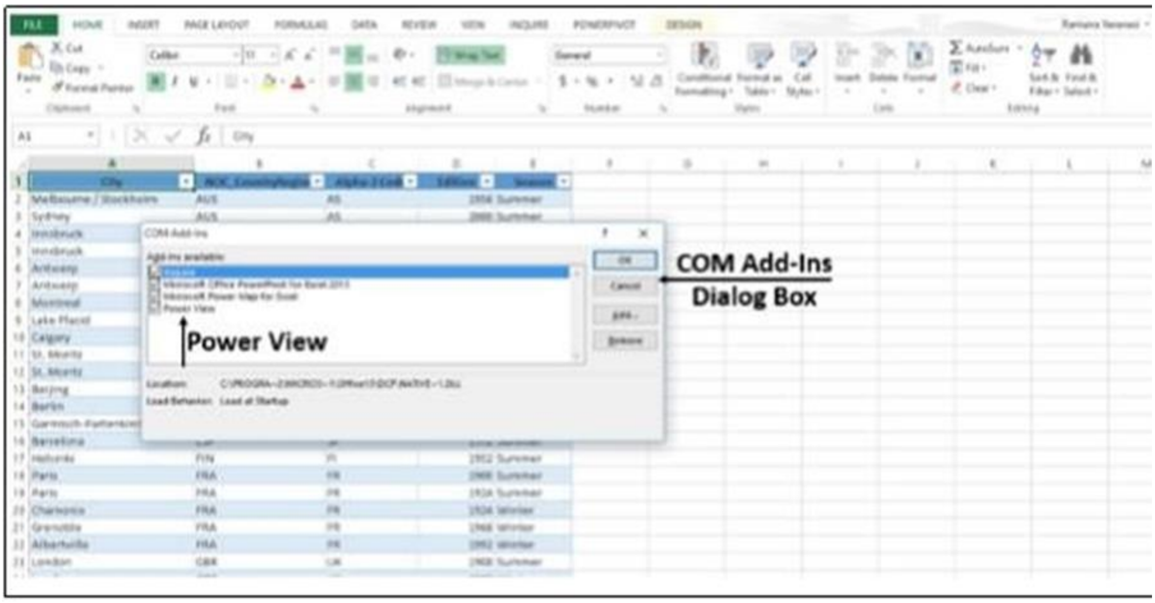
Step 2 − In the Manage box, click the drop-down arrow and select

COM Add-ins Step 3 − Click on the Go button. A COM Add-Ins

Dialog Box appears.

Step 4 − Check the Power View Check Box.

Step 5 − Click OK.



COM Add-Ins

Add-Ins available:

- ☐ Inquire
- ☐ Microsoft Office PowerPivot for Excel 2013
- ☐ Microsoft Power Map for Excel
- ☐ Power View

Location: [illegible]

Load Behavior: Load at Startup

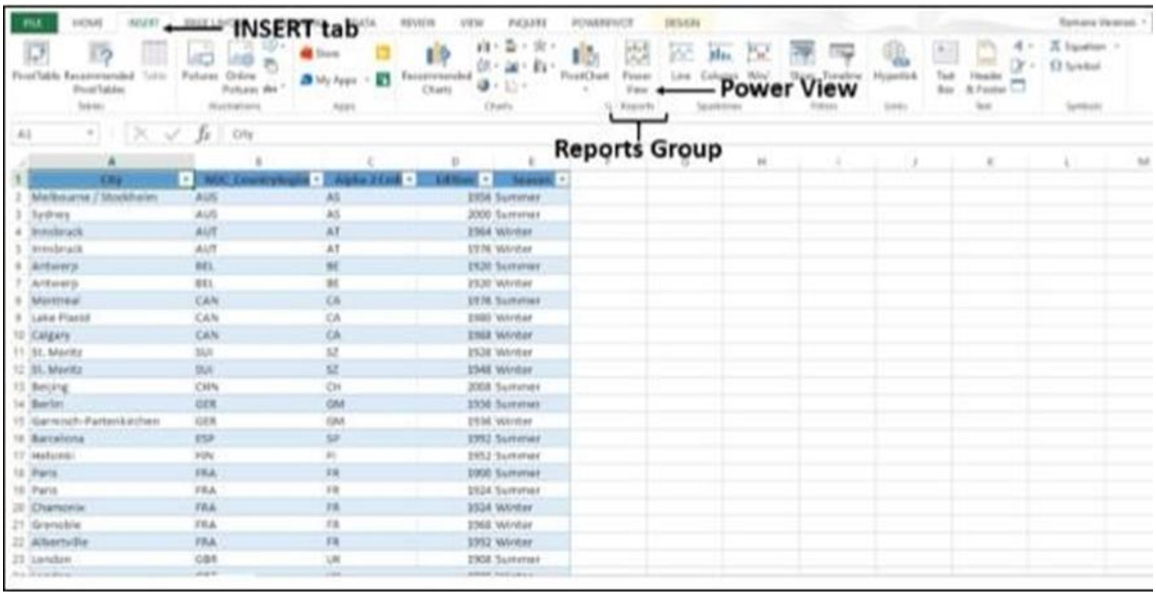
Now, you are ready to create the

Power View sheet. Step 1 − Click on

the Data Table.

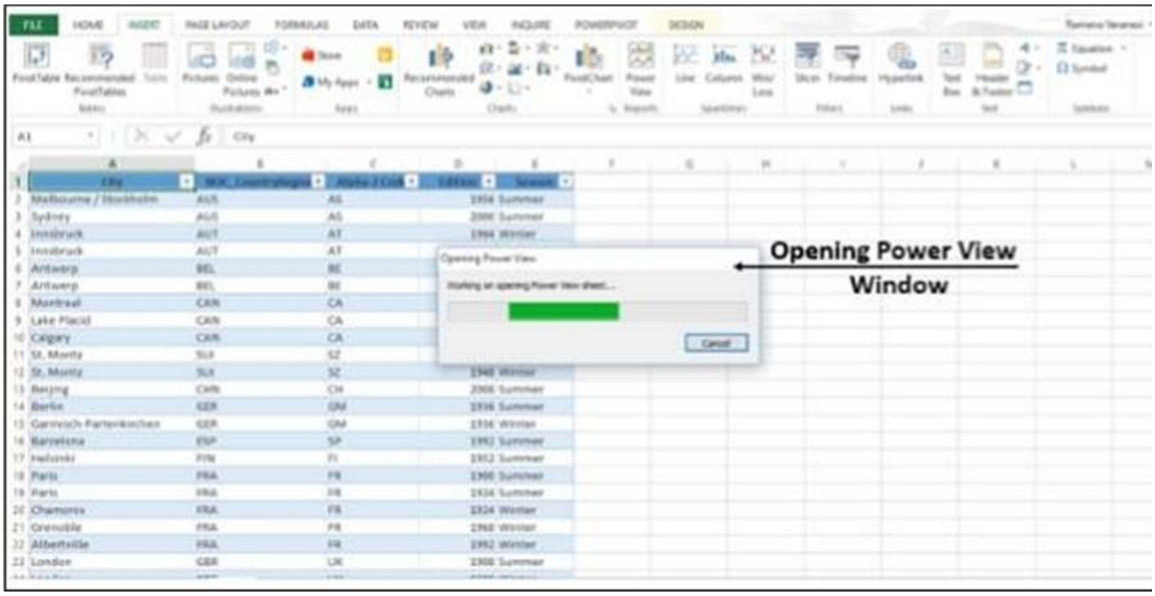
Step 2 − Click on Insert tab.

Step 3 − Click on Power View in Reports Group.



City	NOC_CountryRegion	Alpha 2 Code	Edition	Season
Melbourne / Stockholm	AUS	AS	1956	Summer
Sydney	AUS	AS	2000	Summer
Innsbruck	AUT	AT	1964	Winter
Innsbruck	AUT	AT	1976	Winter
Antwerp	BEL	BE	1920	Summer
Antwerp	BEL	BE	1920	Winter
Montreal	CAN	CA	1976	Summer
Lake Placid	CAN	CA	1980	Winter
Calgary	CAN	CA	1988	Winter
St. Moritz	SUI	SZ	1928	Winter
St. Moritz	SUI	SZ	1948	Winter
Beijing	CHN	CH	2008	Summer
Berlin	GER	GM	1936	Summer
Garmisch-Partenkirchen	GER	GM	1936	Winter
Barcelona	ESP	SP	1992	Summer
Helsinki	FIN	FI	1952	Summer
Paris	FRA	FR	1900	Summer
Paris	FRA	FR	1924	Summer
Chamonix	FRA	FR	1924	Winter
Grenoble	FRA	FR	1968	Winter
Albertville	FRA	FR	1992	Winter
London	GBR	UK	1908	Summer

An Opening Power View window opens, showing the progress of Working on opening Power View sheet.



	City	NOC_CountryRegion	Alpha-2 Code	Edition	Season
2	Melbourne / Stockholm	AUS	AS	1956	Summer
3	Sydney	AUS	AS	2000	Summer
4	Innsbruck	AUT	AT	1964	Winter
5	Innsbruck	AUT	AT		
6	Antwerp	BEL	BE		
7	Antwerp	BEL	BE		
8	Montreal	CAN	CA		
9	Lake Placid	CAN	CA		
10	Calgary	CAN	CA		
11	St. Moritz	SUI	SZ		
12	St. Moritz	SUI	SZ	1948	Winter
13	Beijing	CHN	CH	2008	Summer
14	Berlin	GER	GM	1936	Summer
15	Garmisch-Partenkirchen	GER	GM	1936	Winter
16	Barcelona	ESP	SP	1992	Summer
17	Helsinki	FIN	FI	1952	Summer
18	Paris	FRA	FR	1900	Summer
19	Paris	FRA	FR	1924	Summer
20	Chamonix	FRA	FR	1924	Winter
21	Grenoble	FRA	FR	1968	Winter
22	Albertville	FRA	FR	1992	Winter
23	London	GBR	UK	1908	Summer

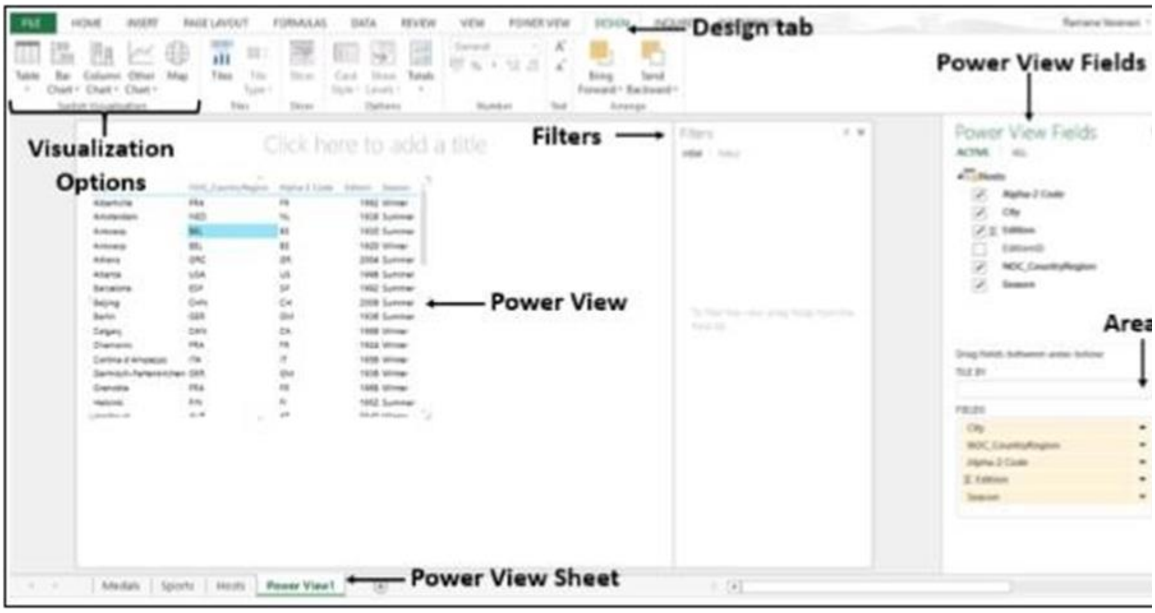
Opening Power View

Working on opening Power View sheet...

Cancel

The Power View sheet is created for you and added to your Workbook with the Power View. On the Right-side of the Power View, you find the Power View Fields. Under the Power View Fields you will find Areas.

In the Ribbon, if you click on Design tab, you will find various Visualization options.



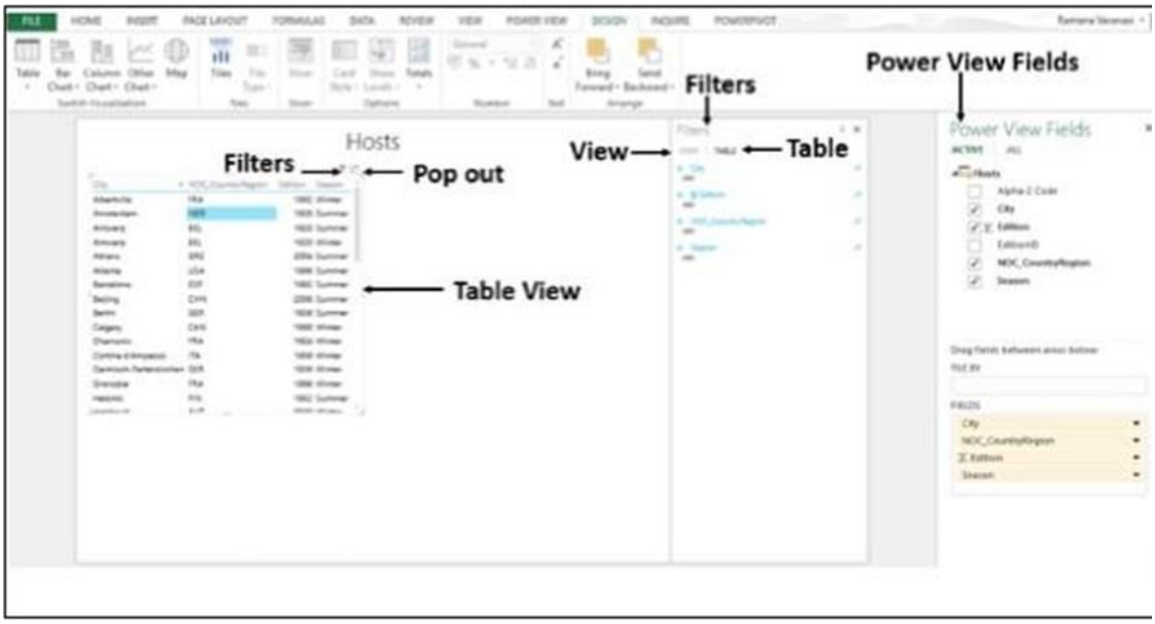
Create Charts and other Visualizations

For every visualization you want to create, you start on a Power View sheet by creating a table, which you then easily convert to other visualizations, to find one that best illustrates your Data.

Step 1 − Under the Power View Fields, select the fields you want to visualize.

Step 2 − By default, the Table View will be displayed. As you move across the Table, on the top-right corner, you find two symbols – Filters and Pop out.

Step 3 − Click on the Filters symbol. The filters will be displayed on the right side. Filters has two tabs. View tab to filter all visualizations in this View and Table tab to filter the specific values in this table only.



Visualization – Matrix

A Matrix is made up of rows and columns like a Table. However, a Matrix has the following capabilities that a Table does not have −

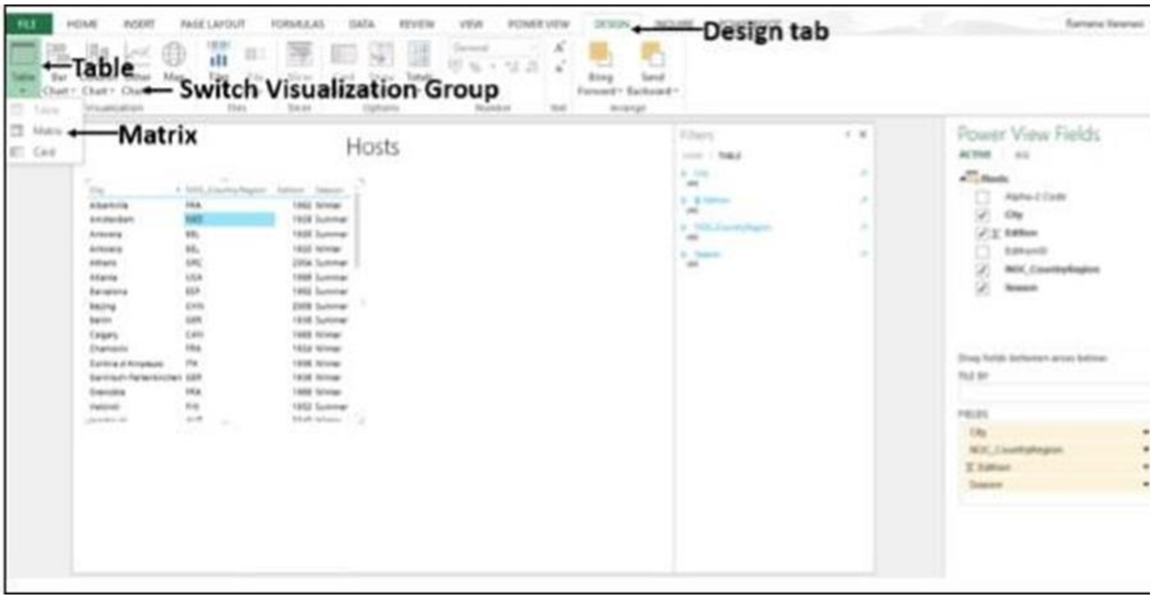
- Display data without repeating values.
- Display totals and subtotals by row and column.
- With a hierarchy, you can drill up/drill down.

Collapse and Expand the Display

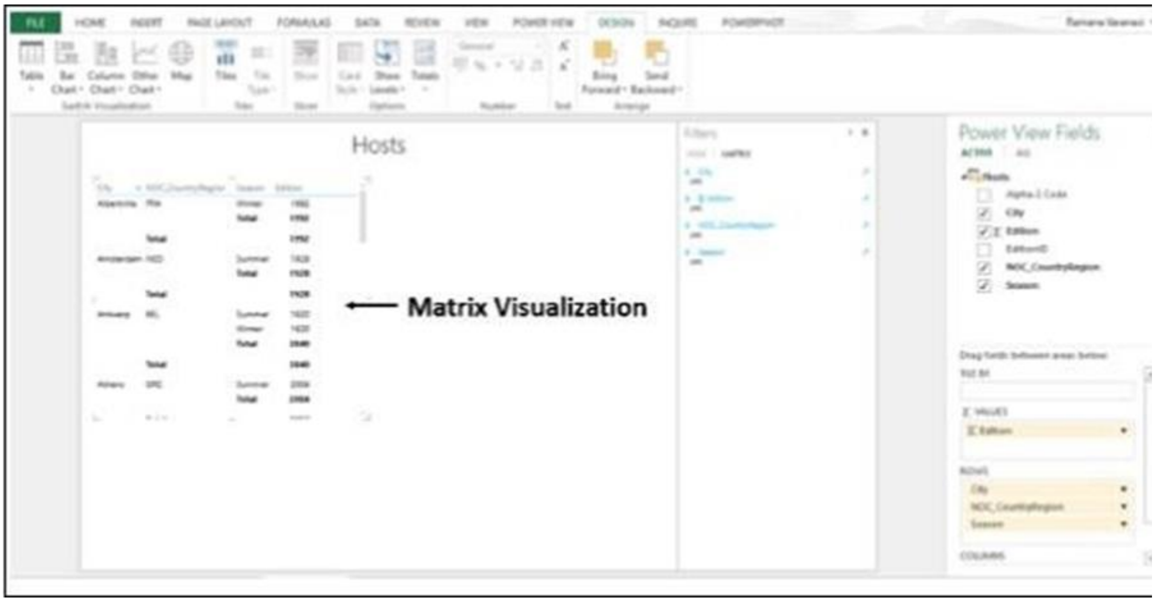
Step 1 – Click on the DESIGN tab.

Step 2 – Click on Table in the Switch Visualization Group.

Step 3 – Click on Matrix.



The Matrix Visualization appears.



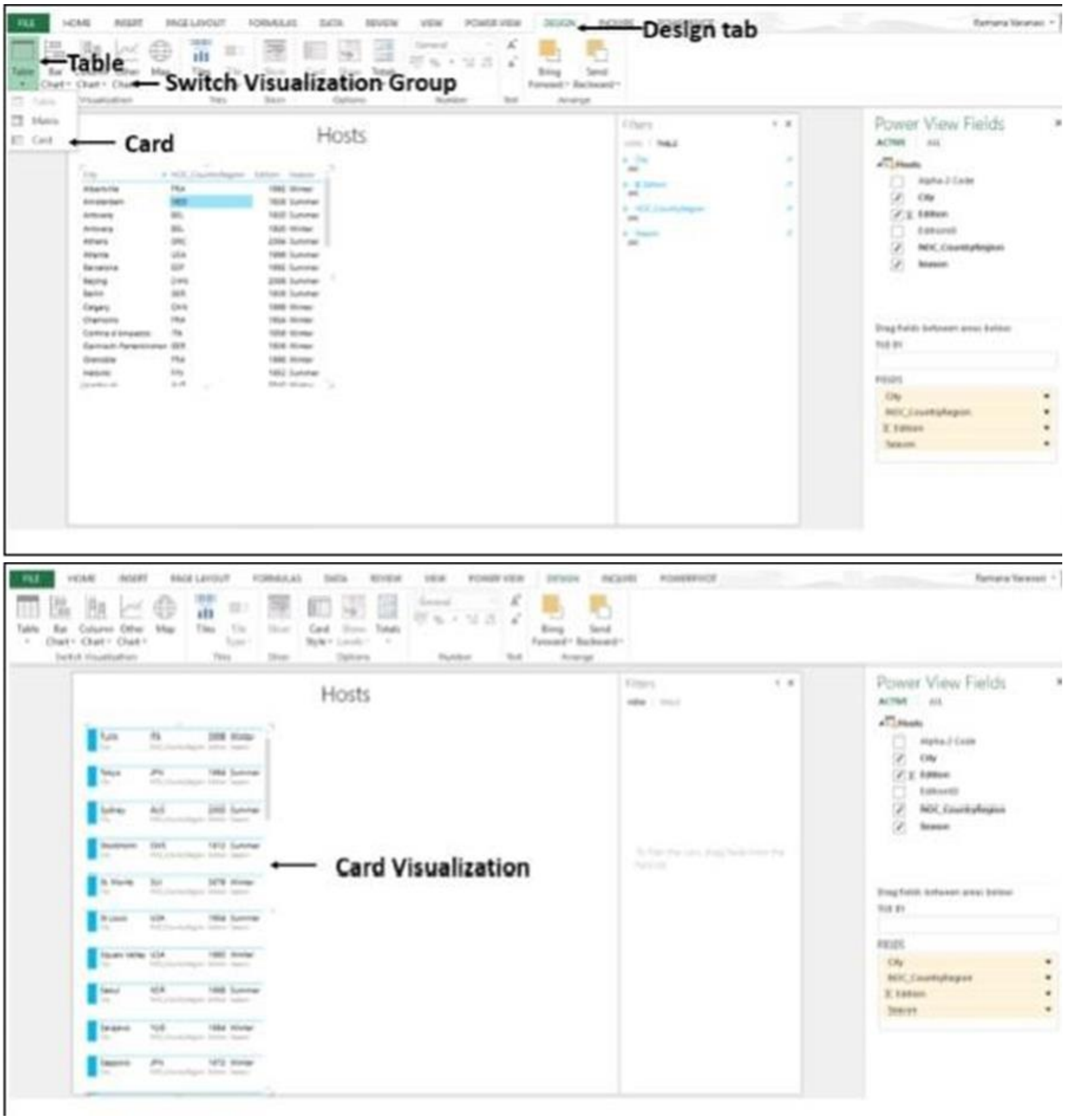
Visualization – Card

You can convert a Table to a series of Cards that display the data from each row in the table laid out in a Card format, like an index Card.

Step 1 − Click on the DESIGN tab.

Step 2 − Click on Table in the Switch Visualization Group.

Step 3 − Click on Card.



Visualization – Charts

In Power View, you have a number of Chart options: Pie, Column, Bar, Line, Scatter, and Bubble. You can use several design options in a chart such as showing and hiding labels, legends, and titles. Charts are interactive. If you click on a Value in one Chart −

- the Value in that chart is highlighted.
- All the Tables, Matrices, and Tiles in the report are filtered to that Value.
- That Value in all the other Charts in the report is highlighted.

The charts are interactive in a presentation setting also.

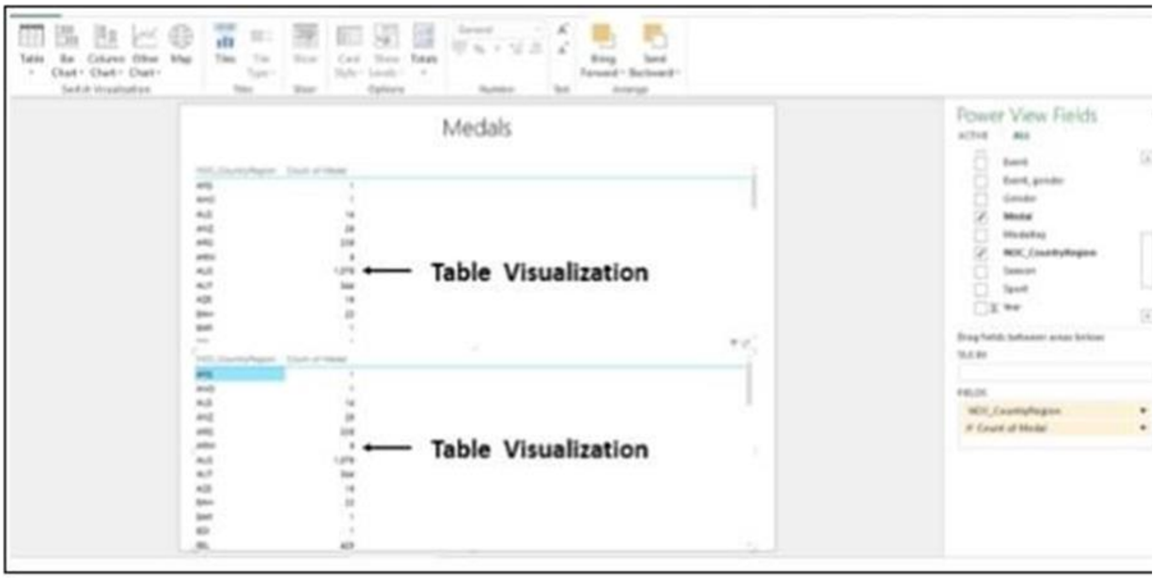
Step 1 − Create a Table Visualization from Medals data.

You can use Line, Bar and Column Charts for comparing data points in one or more data series. In these Charts, the x-axis displays one field and the y-axis displays another, making it easy to see the relationship between the two values for all the items in the Chart.

Line Charts distribute category data evenly along a horizontal (category) axis, and all numerical value data along a vertical (value) axis.

Step 2 − Create a Table Visualization for two Columns, NOC_CountryRegion and Count of Medal.

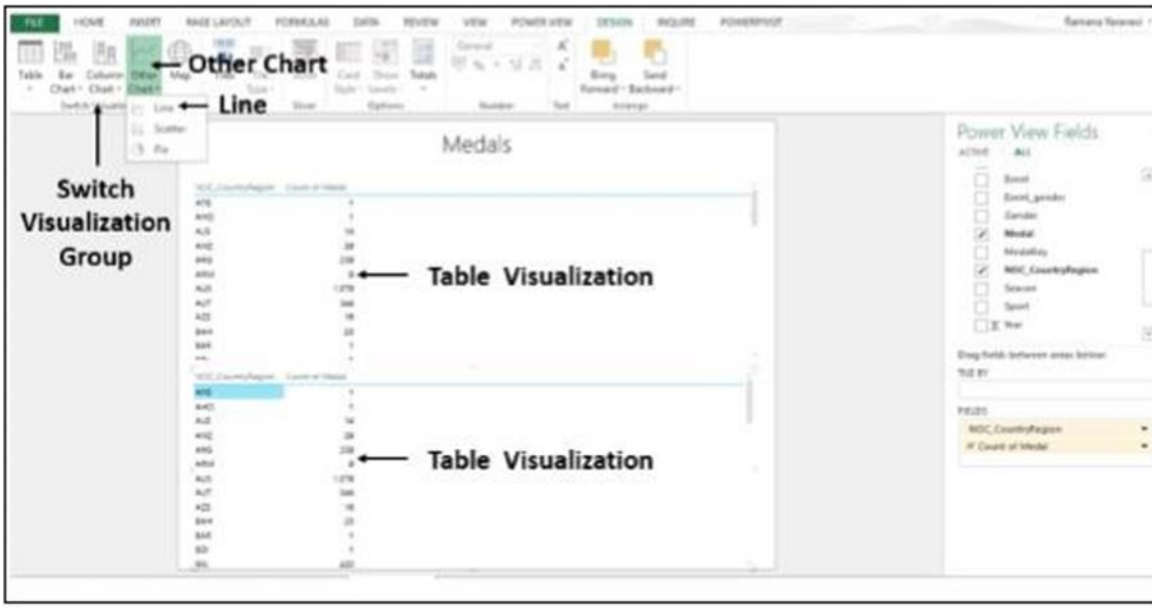
Step 3 − Create the same Table Visualization below



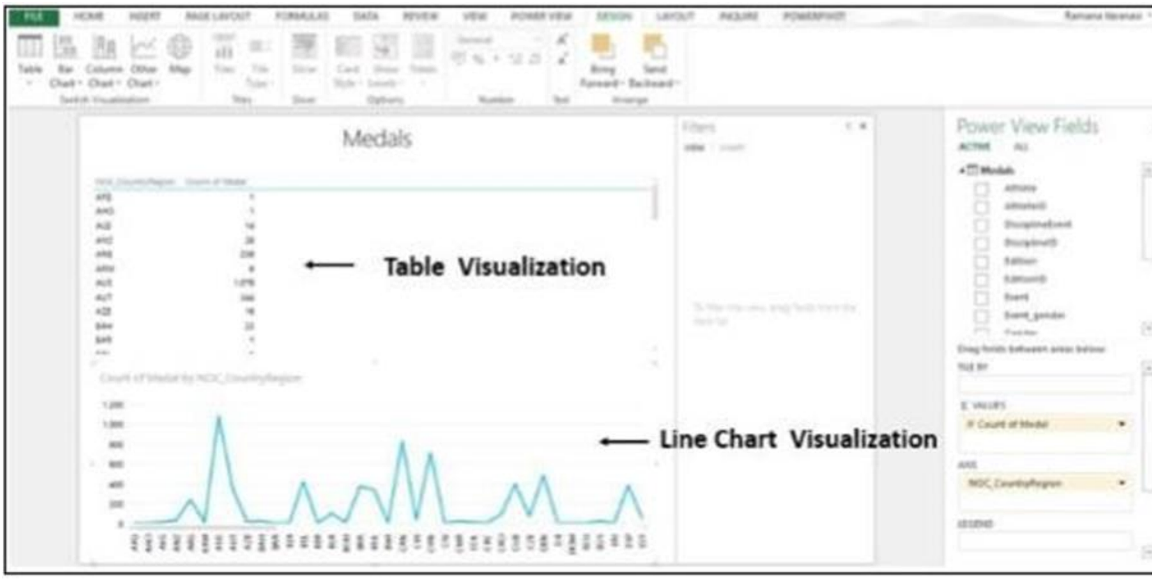
Step 4 − Click on the Table Visualization below.

Step 5 − Click on Other Chart in the Switch Visualization group.

Step 6 − Click on Line.



The Table Visualization converts into Line Chart Visualization.

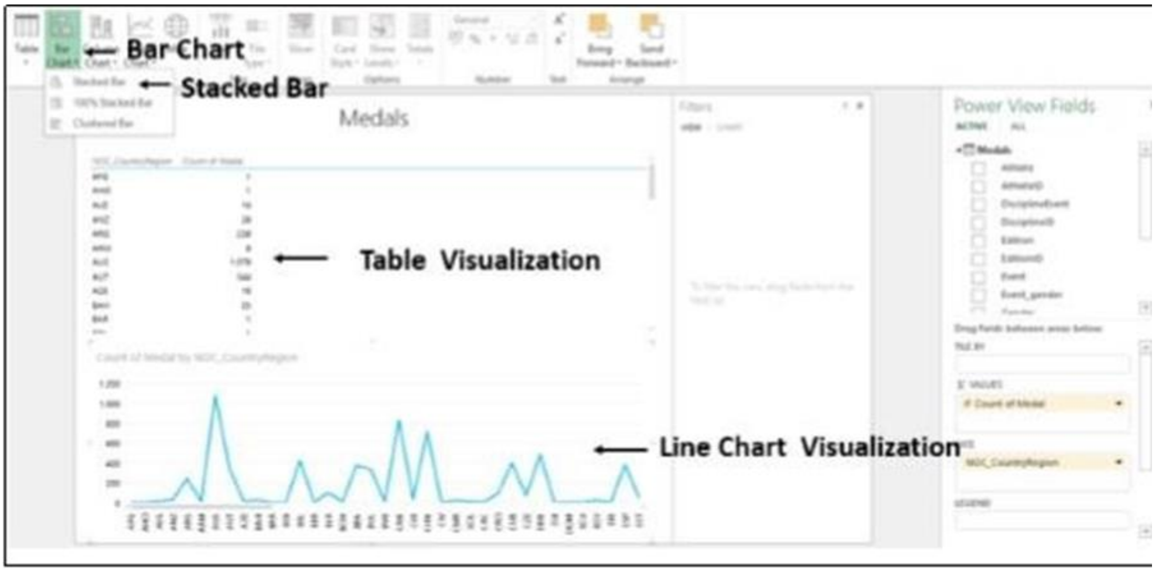


In a Bar Chart, categories are organized along the vertical axis and values along the horizontal axis. In Power View, there are three subtypes of the Bar Chart: Stacked, 100% stacked, and Clustered.

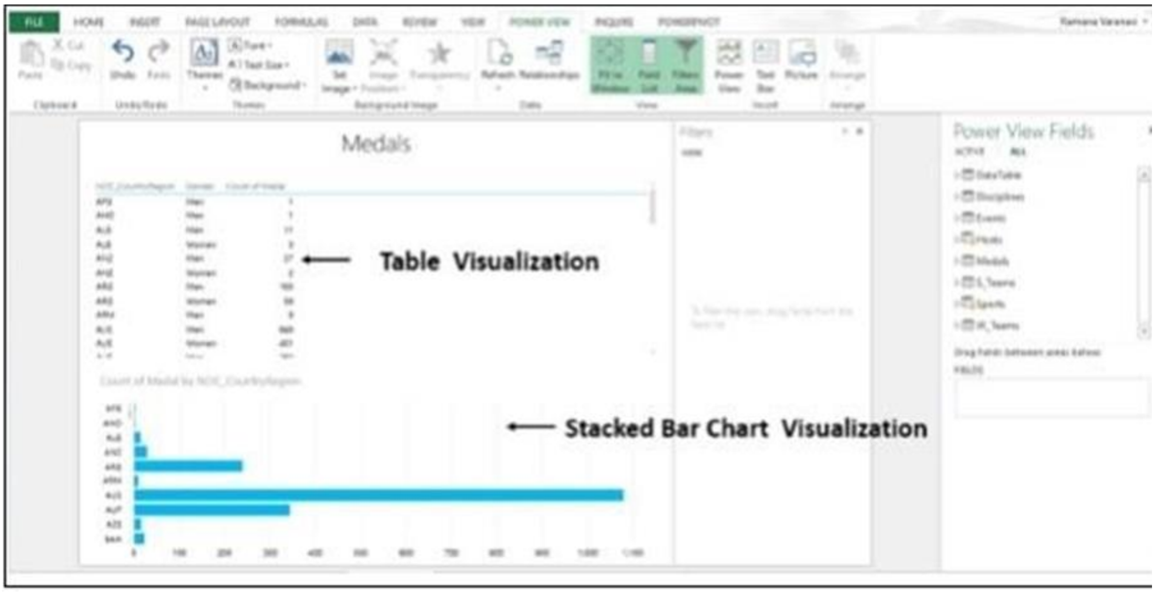
Step 7 − Click on the Line Chart Visualization.

Step 8 − Click on Bar Chart in the Switch Visualization Group.

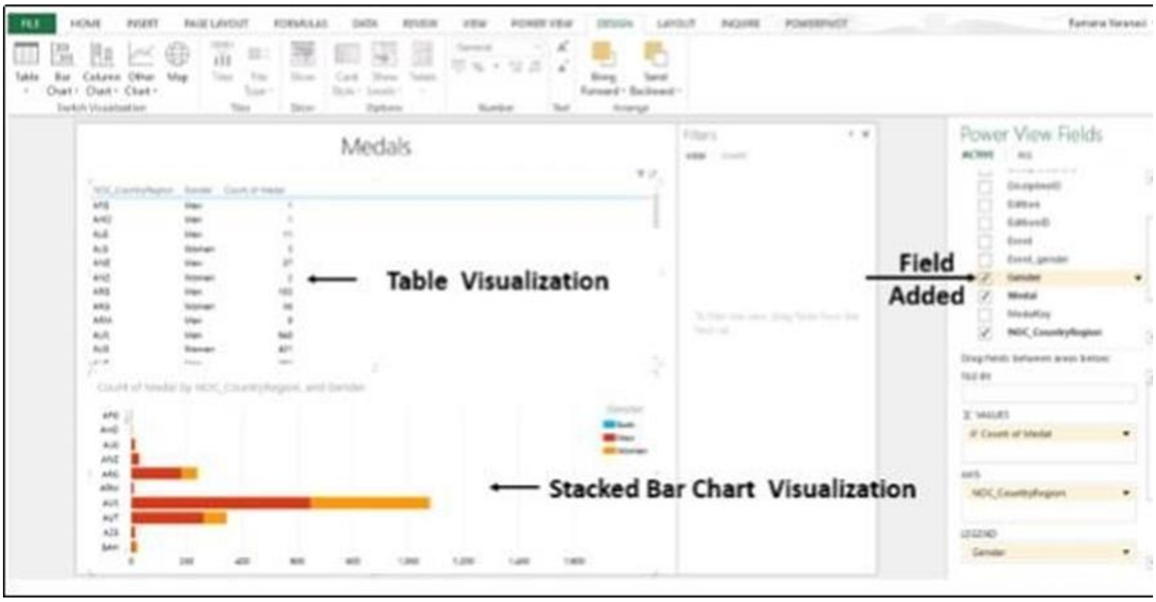
Step 9 − Click on the Stacked Bar option.



The Line Chart Visualization converts into Stacked Bar Chart Visualization.

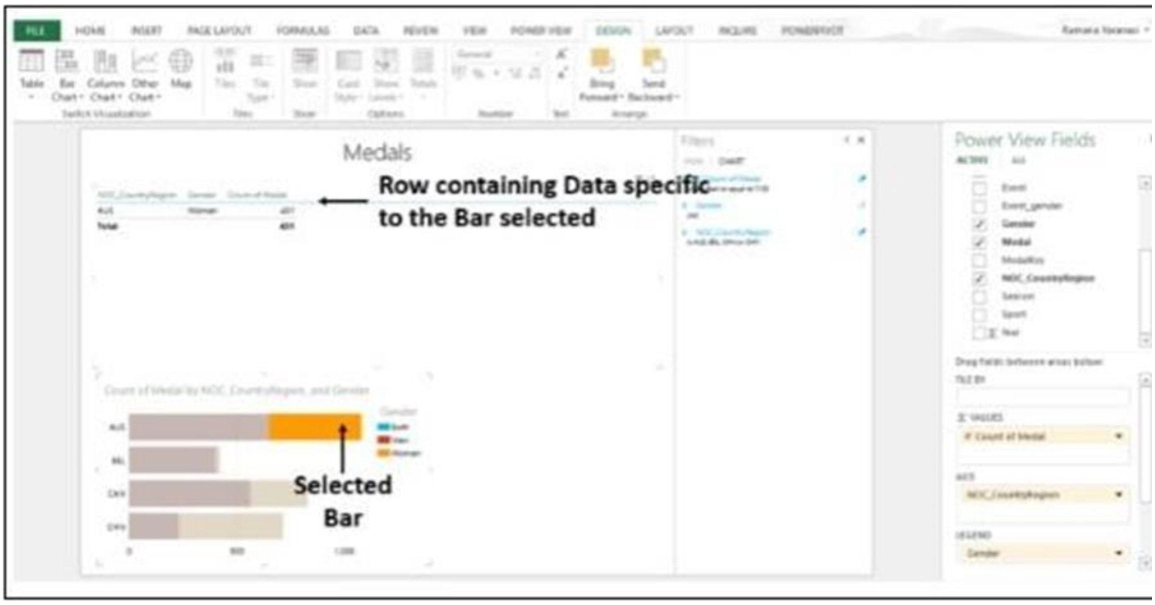


Step 10 – In the Power View Fields, in the Medals Table, select the Field Gender also.



Step 11 – Click on one of the bars. That portion of the bar is highlighted. Only the row containing the

Data specific to the selected bar is displayed in the table above.



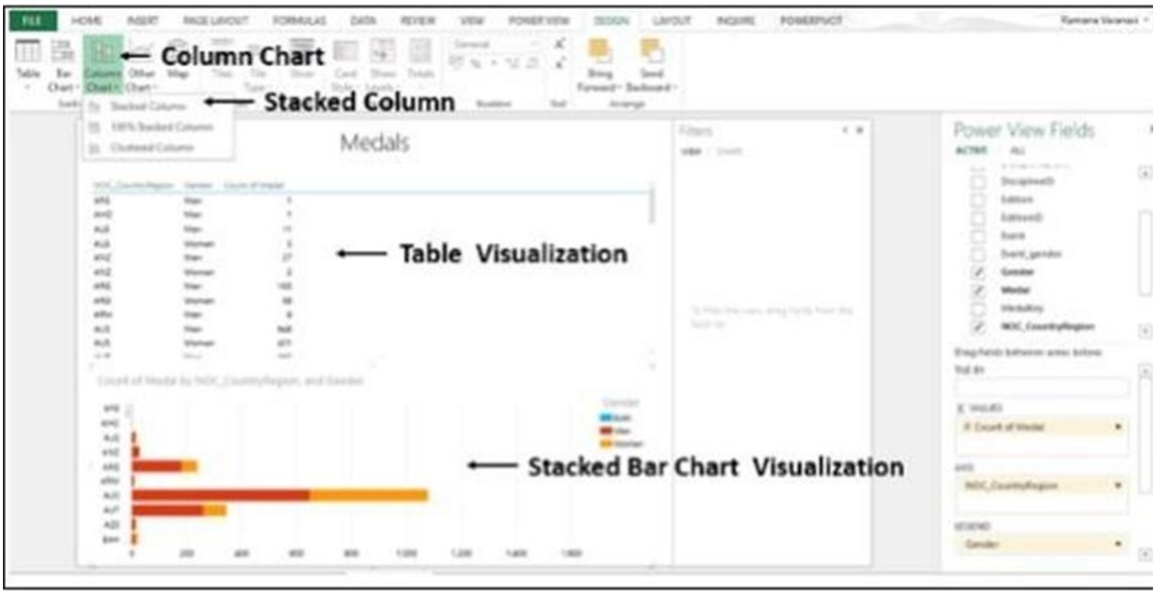
You can use the column charts for showing data changes over a period of time or for illustrating comparison among different items. In a Column Chart, the categories are along the horizontal axis and values are along the vertical axis.

In Power View, there are three Column Chart subtypes: Stacked, 100% stacked, and Clustered.

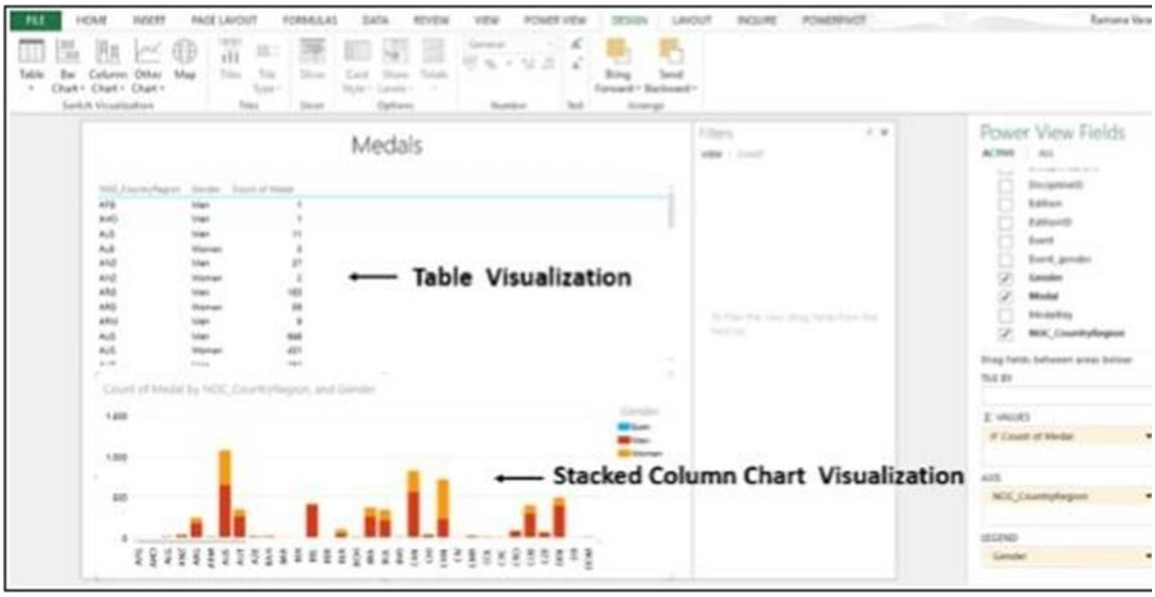
Step 12 − Click on the Stacked Bar Chart Visualization.

Step 13 − Click on Column Chart in the Switch Visualization group.

Step 14 − Click on Stacked Column.



The Stacked Bar Chart Visualization converts into Stacked Column Chart Visualization.

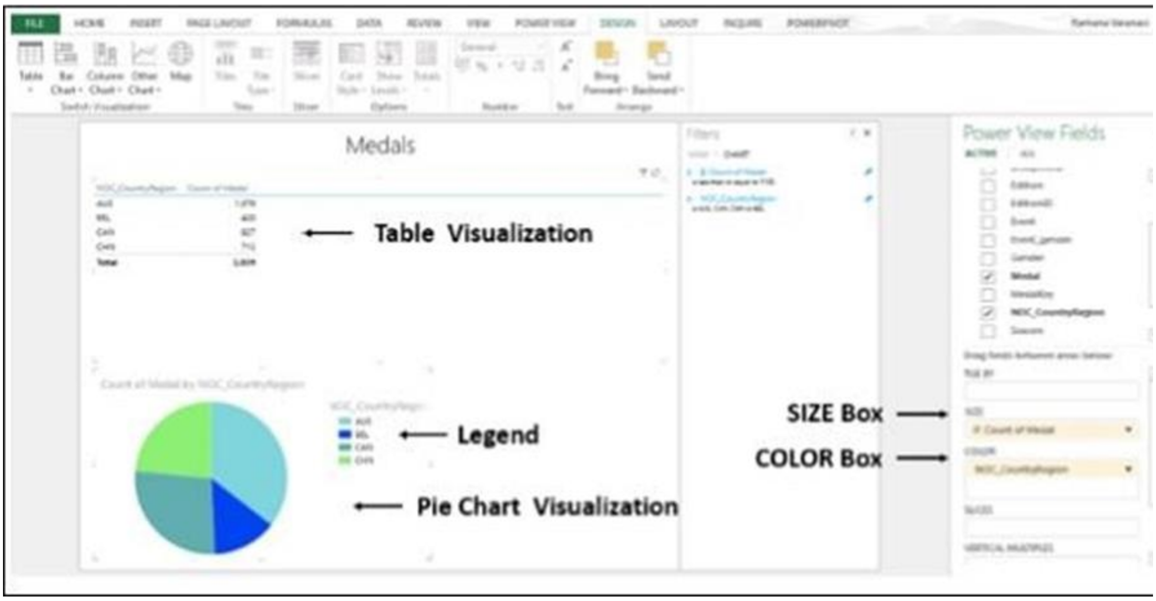


You can have simple Pie Chart Visualizations in Power View.

Step 1 – Click on the Table Visualization as shown below.

Step 2 – Click on Other Chart in the Switch Visualization group.

Step 3 – Click on Pie as shown in the image given below.



Applications:

Financial Analysis: Analyzing financial statements, conducting budgeting and forecasting, and creating financial models using Excel's advanced functions and features.

Supply Chain Management: Optimizing inventory levels, analyzing supplier performance, and identifying cost-saving opportunities by analyzing supply chain data in Excel.

Business Intelligence: Building interactive dashboards, performing ad-hoc analysis, and generating actionable insights from large datasets to support strategic decision-making.

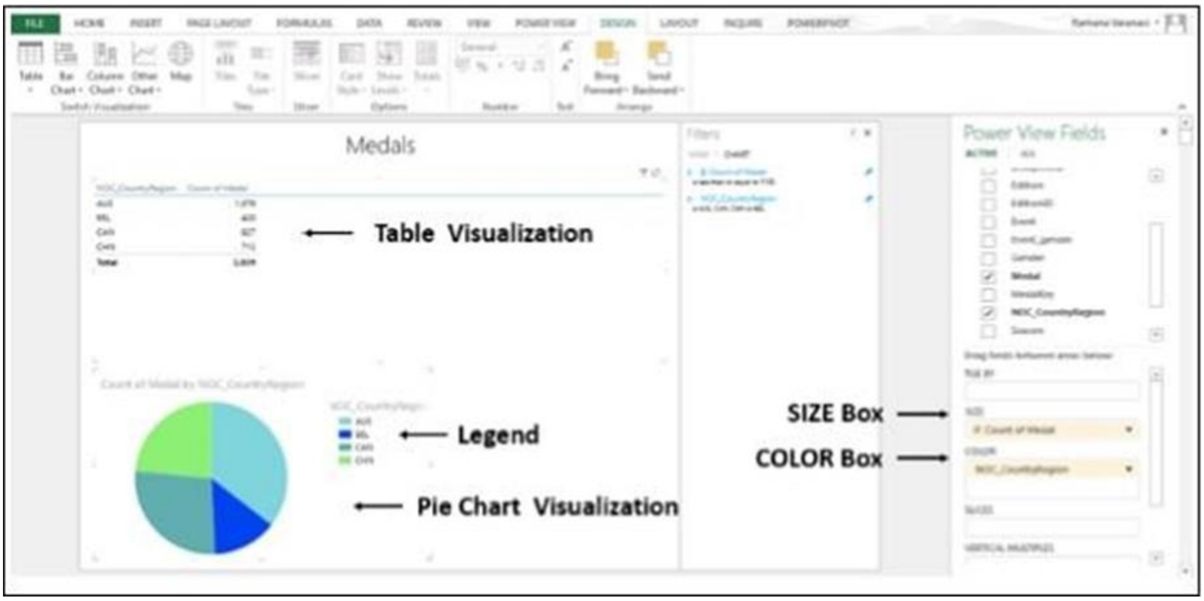
Education Analytics: Analyzing student performance data, identifying areas for improvement in educational programs, and tracking learning outcomes using Excel's analytical capabilities.

Input:

Students need to use any sample dataset or

Select **Events.accdb**, Events Access Database file.

Output:



Conclusion:

Outcome: The successful completion of the Data Analysis and Visualization using Advanced Excel project will result in the creation of visually compelling and insightful reports, presentations that effectively communicate complex data trends, patterns, and relationships.

This will enable stakeholders to make data-driven decisions with confidence, leading to improved efficiency, productivity, and strategic outcomes within the organization.

Questions:

1. How can advanced Excel features such as conditional formatting and data validation be used to improve the visual presentation and accuracy of analytical findings?

2. How can Excel be integrated with other data analysis and visualization tools or platforms to streamline workflows and enhance collaboration among team members?

3. What strategies can be employed to ensure the security and integrity of sensitive data when conducting analysis and visualization projects in Excel?

4.5 *Experiment No. 5*

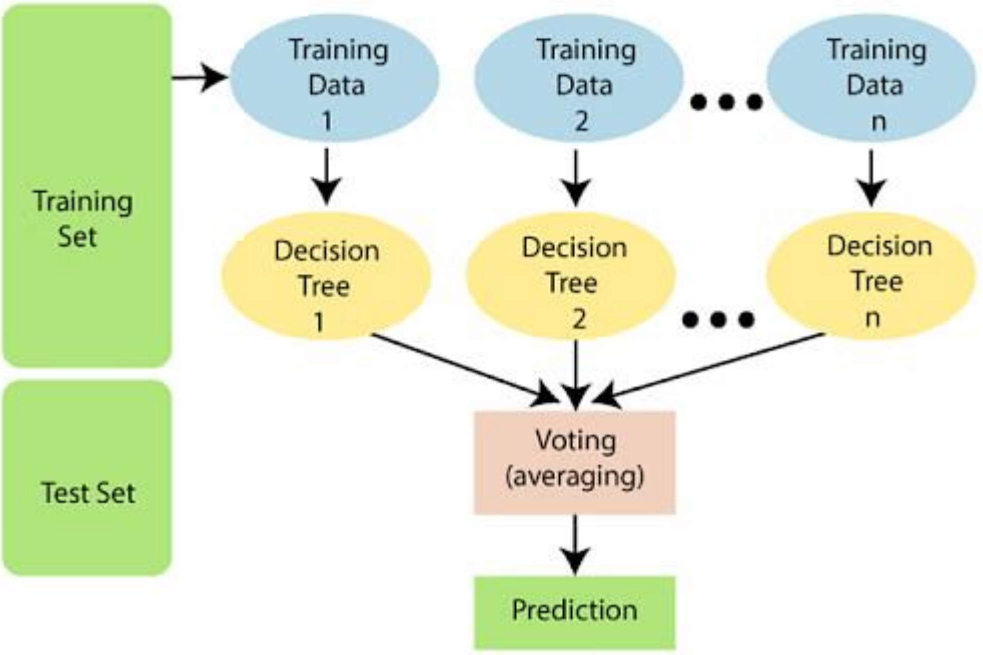
Title of Assignment: Practical Implementation of Data Classification Using Random Forest Algorithm

Objective: To implement and evaluate the practical application of data classification using the Random Forest algorithm, aiming to develop robust predictive models that accurately classify data instances into predefined categories.

Theory: Business Intelligence (BI) leverages data analysis tools and techniques to transform raw data into actionable insights for informed decision-making. Classification algorithms play a crucial role in BI by categorizing data into meaningful groups or classes. In this practical write-up, we will demonstrate how to perform data classification using the Random Forest algorithm, a powerful ensemble learning technique.

Dataset:

For our demonstration, we will use a hypothetical dataset from a retail business containing customer transaction data. The dataset includes various features such as customer demographics, purchase history, and product preferences. Our goal is to classify customers into different segments based on their purchasing behavior.



The following steps explain the working Random Forest Algorithm:

Step 1: Select random samples from a given data or training set.

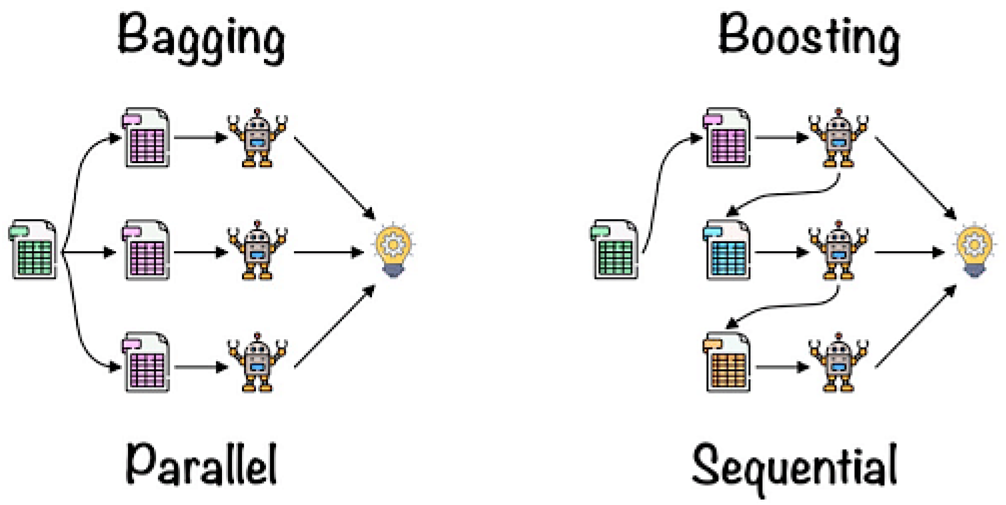
Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision tree.

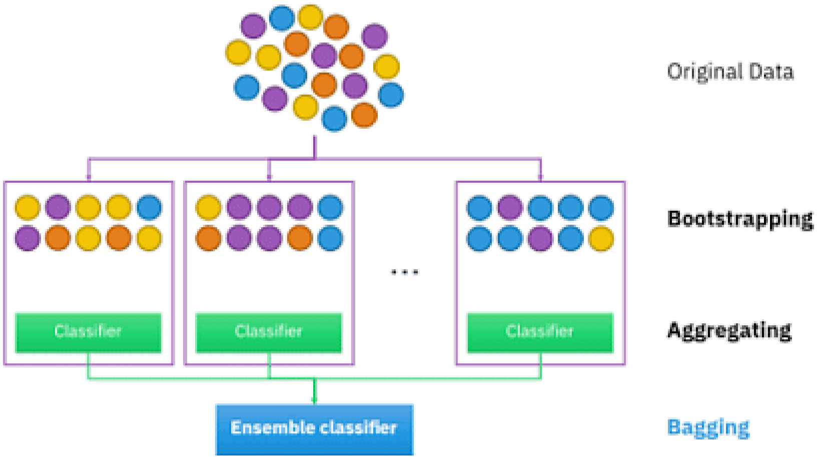
Step 4: Finally, select the most voted prediction result as the final prediction result.

This combination of multiple models is called Ensemble. Ensemble uses two methods:

1. Bagging: Creating a different training subset from sample training data with replacement is called Bagging. The final output is based on majority voting.
2. Boosting: Combing weak learners into strong learners by creating sequential models such that the final model has the highest accuracy is called Boosting. Example: ADA BOOST, XG BOOST.



Bagging: From the principle mentioned above, we can understand Random forest uses the Bagging code. Now, let us understand this concept in detail. Bagging is also known as Bootstrap Aggregation used by random forest. The process begins with any original random data. After arranging, it is organised into samples known as Bootstrap Sample. This process is known as Bootstrapping.Further, the models are trained individually, yielding different results known as Aggregation. In the last step, all the results are combined, and the generated output is based on majority voting. This step is known as Bagging and is done using an Ensemble Classifier.



Implementation Steps:

Data Collection and Preprocessing:

- Gather the customer transaction data from the retail database or CRM system.
- Preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features if necessary.

Feature Selection and Engineering:

- Identify relevant features that may influence customer segmentation.
- Perform feature engineering to create new features or transform existing ones to improve model performance.

Random Forest Classifier:

- Import the RandomForestClassifier from the Scikit-learn library.
- Split the dataset into training and testing sets using techniques like cross-validation.
- Initialize the Random Forest classifier and specify hyperparameters such as the number of trees, maximum depth, and minimum samples per leaf.
- Train the classifier using the training data.

Model Evaluation:

- Evaluate the performance of the trained classifier using metrics like accuracy, precision, recall, and F1-score.
- Utilize techniques such as confusion matrices and ROC curves to assess model performance and identify areas for improvement.

Segmentation Analysis:

- Analyze the results of the classification to identify distinct customer segments.
- Explore the characteristics and behaviors of each segment to gain insights into customer preferences and trends.

Why Use a Random Forest Algorithm?

There are a lot of benefits to using Random Forest Algorithm, but one of the main advantages is that it reduces the risk of overfitting and the required training time. Additionally, it offers a high level of accuracy. Random Forest algorithm runs efficiently in large databases and produces highly accurate predictions by estimating missing data.

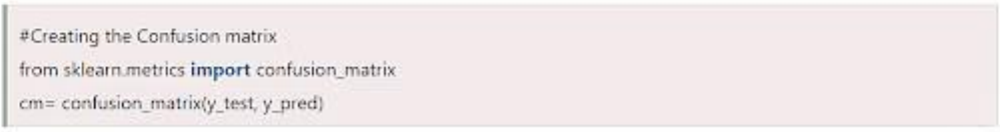
Applications:

Some of the applications of Random Forest Algorithm are listed below:

1. Banking: It predicts a loan applicant's solvency. This helps lending institutions make a good decision on whether to give the customer loan or not. They are also being used to detect fraudsters.
2. Health Care: Health professionals use random forest systems to diagnose patients. Patients are diagnosed by assessing their previous medical history. Past medical records are reviewed to establish the proper dosage for the patients.
3. Stock Market: Financial analysts use it to identify potential markets for stocks. It also enables them to remember the behaviour of stocks.
4. E-Commerce: Through this system, e-commerce vendors can predict the preference of customers based on past consumption behaviour.

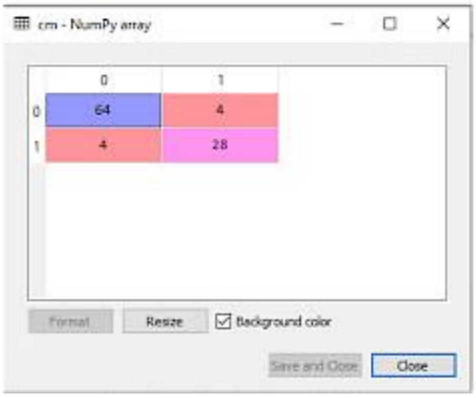
Input: user_data.csv

Output:



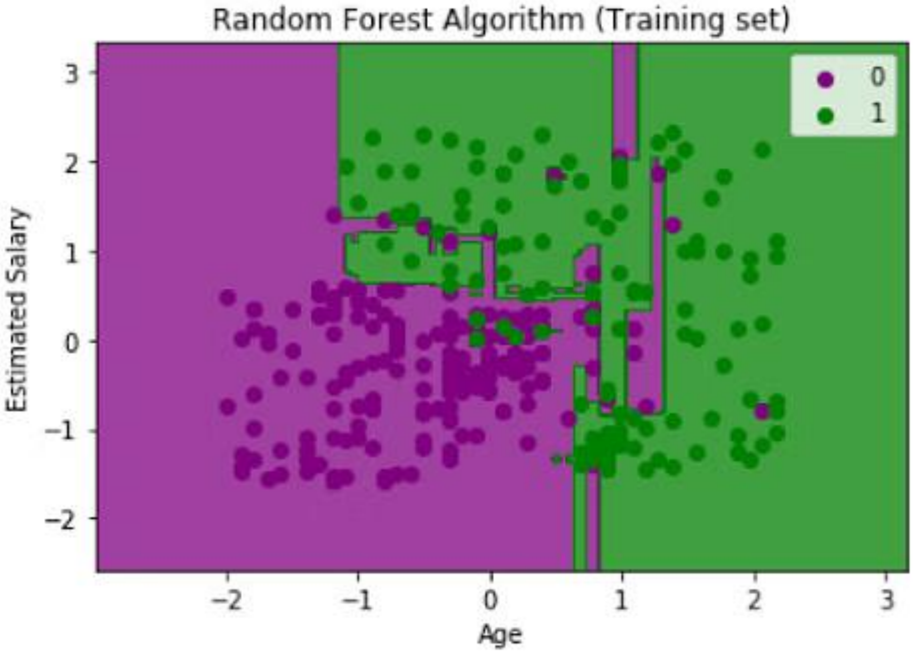
```
#Creating the Confusion matrix
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
```

Output:



	0	1
0	64	4
1	4	28

Output:



Conclusion: In this practical, we have studied the implementation of a data classification algorithm using the Random Forest algorithm in the context of business intelligence. By leveraging classification techniques, businesses can gain valuable insights into customer behavior, market trends, and other critical factors that drive decision-making processes. The Random Forest algorithm, with its ability to handle large datasets and capture complex relationships, offers a powerful tool for segmenting customers and identifying patterns that can drive business growth and innovation.

Outcome: Students will be successfully able to classify data Using Random Forest Algorithm

Questions:

Q1: What are the key steps involved in implementing a Random Forest model for data classification, from data preprocessing to model evaluation?

Q 2: How does the Random Forest algorithm handle overfitting, and what techniques can be employed to fine-tune its hyperparameters for optimal performance?

Q3: How does the computational complexity of Random Forest algorithm compare to other classification algorithms, particularly in terms of training time and prediction speed?

4.6 Experiment No. 6

Aim: Data Clustering Using Decision Tree Algorithm for Business Intelligence

Objective: to introduce students to the practical implementation of data clustering using the decision tree algorithm for business intelligence applications.

By the end of the session, students should be able to understand the concepts of decision tree clustering and apply it to analyze and interpret business datasets.

Theory:

Machine learning algorithms are used in almost every sector of business to solve critical problems and build intelligent systems and processes. Supervised machine learning algorithms, specifically, are used for solving classification and regression problems. one of the most popularly used supervised learning algorithms: decision trees in Python.

What is a Decision Tree?

A decision tree is a tree-based supervised learning method used to predict the output of a target variable. Supervised learning uses labeled data (data with known output variables) to make predictions with the help of regression and classification algorithms. Supervised learning algorithms act as a supervisor for training a model with a defined output variable. It learns from simple decision rules using the various data features. Decision trees in Python can be used to solve both classification and regression problems—they are frequently used in determining odds.

Advantages of Using Decision Trees

- Decision trees are simple to understand, interpret, and visualize
- They can effectively handle both numerical and categorical data
- They can determine the worst, best, and expected values for several scenarios
- Decision trees require little data preparation and data normalization
- They perform well, even if the actual model violates the assumptions

Important Terms Used in Decision Trees

1. Entropy: Entropy is the measure of uncertainty or randomness in a data set. Entropy handles how a decision tree splits the data.

It is calculated using the following formula:

$$\sum_{i=1}^{k} P(value_i).log_2(P(value_i)$$

2. Information Gain: The information gain measures the decrease in entropy after the data set is split.

It is calculated as follows:

IG(Y, X) = Entropy (Y) - Entropy (Y | X)

3. Gini Index: The Gini Index is used to determine the correct variable for splitting nodes. It measures how often a randomly chosen variable would be incorrectly identified.

4. Root Node: The root node is always the top node of a decision tree. It represents the entire population or data sample, and it can be further divided into different sets.

5. Decision Node: Decision nodes are subnodes that can be split into different subnodes; they contain at least two branches.

6. Leaf Node: A leaf node in a decision tree carries the final results. These nodes, which are also known as terminal nodes, cannot be split any further.

Building a Decision Tree in Python

We’ll now predict if a consumer is likely to repay a loan using the decision tree algorithm in Python. The data set contains a wide range of information for making this prediction, including the initial payment amount, last payment amount, credit score, house number, and whether the individual was able to repay the loan.

2. Dataset Exploration:

- Load the dataset for the clustering task.
- Explore the dataset to understand its structure, features, and distribution.
- Preprocess the dataset if necessary (e.g., handling missing values, scaling features).

3. Data Preparation:

- Select relevant features for clustering.
- Standardize or normalize the data if needed.

4. Building the Decision Tree Model:

- Import the necessary libraries (scikit-learn).
- Create an instance of the DecisionTreeClassifier for clustering.
- Train the decision tree model using the dataset.

5. Visualizing Clusters:

- Visualize the decision tree structure to understand how data is partitioned into clusters.
- Plot the clusters using appropriate visualization techniques (e.g., scatter plot, dendrogram).

6. Interpretation and Analysis:

- Analyze the clusters generated by the decision tree algorithm.
- Interpret the results in the context of business intelligence objectives (e.g., customer segmentation, market analysis).
- Discuss the implications of the clustering results for business decision-making.

7. Fine-tuning the Model:

- Discuss parameters such as tree depth, criterion, and splitting strategy.
- Experiment with different parameter values to optimize the clustering performance.

Hyperparameters for decision tree

The performance of a machine learning model can be improved by tuning its hyperparameters. Hyperparameters are those parameters that the user has to set in advance. They are not learned by the data during training.

Some of the most common hyperparameters for a decision tree are:

Criterion: This parameter determines how the impurity of a split will be measured. Possibilities are ‘gini’ or ‘entropy.

Splitter: How the decision tree searches the features for a split. The default is set to ‘best’ meaning for each node, the algorithm considers all the features and chooses the best split. If it is set to random, then a random subset of features will be considered. The split will be made by the best feature within the random subset. The size of the random subset is determined by the ‘max_features’ parameter.

Max_Depth: This determines how deep the tree will be. The default is none and this often results in overfitting. The max_depth parameter is one of the ways in which we can regularize the tree, or limit the way it grows to prevent over-fitting.

Min_samples_split: The minimum number of samples a node must contain to consider splitting. The default is set to 2. This again is another parameter to regularize the decision tree.

Max_features: The number of features to consider when looking for the best split. By default, the decision tree will consider all available features to make the best split.

Loan Repayment dataset

The Loan Repayment dataset is made up of 1000 rows and six columns. Each row represents information about a particular person that relates to loan repayment. This dataset is perfect for classification algorithms such as the decision tree. The six columns are the following.

- Initial payment.
- Last payment.
- Credit score.
- House number.
- Sum.

- Result.

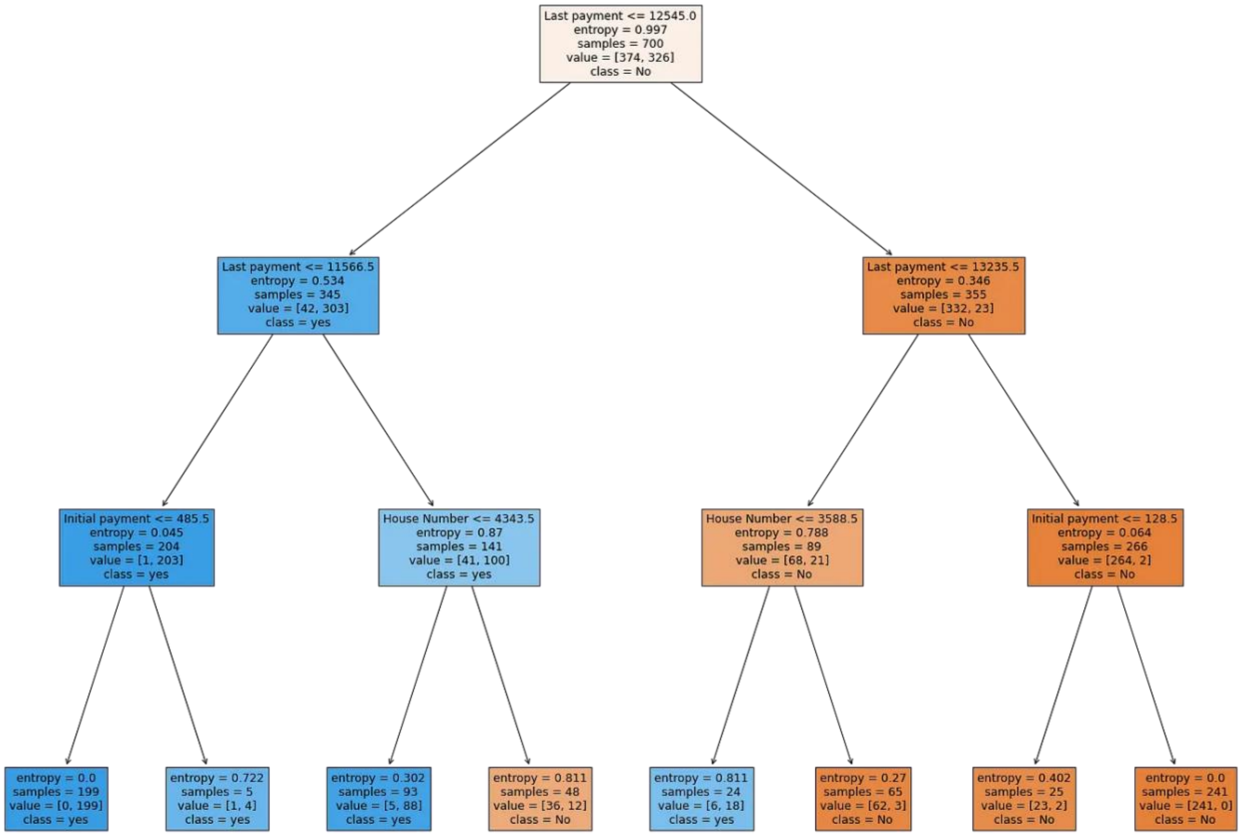
Applications: Decision Tree Applications

1. A decision tree is used to determine whether an applicant is likely to default on a loan.
2. It can be used to determine the odds of an individual developing a specific disease.
3. It can help ecommerce companies in predicting whether a consumer is likely to purchase a specific product.
4. Decision trees can also be used to find customer churn rates.

Input: Import the Loan Repayment dataset as a Pandas data frame

```
df = pd.read_csv('Decision_Tree_ Dataset.csv',
sep= ',', header= 0)
df
```

Output:



Conclusion:

In conclusion, the Decision Tree algorithm offers a powerful tool for data clustering in business intelligence, allowing for the effective analysis and interpretation of complex datasets to derive actionable insights.

Outcome:

The practical implementation of Decision Tree clustering facilitates a deeper understanding of business datasets, enabling students to apply decision tree algorithms for segmentation and analysis tasks in real-world scenarios

Questions:

1. How does the Decision Tree algorithm contribute to business intelligence applications?
2. What are the advantages of using Decision Trees for data clustering in comparison to other methods?

3. Can you explain the significance of hyperparameters in fine-tuning a Decision Tree model?

4. What are some real-world applications of Decision Trees in business and industry?

5 Appendix

Introduction

Python is a widely used high-level programming language. It is one of the most popular and flexible server-side programming languages.

Windows does not have the Python programming language installed by default. However, you can install Python on Windows in just a few easy steps.

This guide provides step-by-step instructions to install and set up Python on Windows.

Prerequisites

- A system running Windows 10 with administrator access.
- Access to the command prompt.
- Access to a web browser.

Python Installation on Windows

The installation requires downloading the official Python *.exe* installer and running it on your system. The sections below will explain several options and details during the installation process.

Step 1: Select Python Version

Deciding on a version depends on what you want to do in Python. The two major versions are Python 2 and Python 3. Choosing one over the other might be better depending on your project details. If there are no constraints, choose whichever one you prefer.

We recommend Python 3, as Python 2 reached its end of life in 2020. Download Python 2 only if you work with legacy scripts and older projects. Also, choose a stable release over the newest since the newest release may have bugs and issues.

Step 2: Download Python Executable Installer

Start by downloading the Python executable installer for Windows:

1. Open a web browser and navigate to the Downloads for Windows section of the official Python website.

2. Locate the desired Python version.

3. Click the link to download the file. Choose either the Windows 32-bit or 64-bit installer.

The download is approximately 25MB.

Step 3: Run Executable Installer

The steps below guide you through the installation process:

1. Run the downloaded **Python Installer**.

2. The installation window shows two checkboxes:

Admin privileges. The parameter controls whether to install Python for the current or all system users. This option allows you to change the installation folder for Python.
Add Python to PATH. The second option places the executable in the PATH variable after installation. You can also add Python to the PATH environment variable manually later.

For the most straightforward installation, we recommend ticking both checkboxes.

3. Select the **Install Now** option for the recommended installation (in that case, skip the next two steps).

To adjust the default installation options, choose **Customize installation** instead and proceed to the following step.

The default installation installs Python to *C:\Users\[user]\AppData\Local\Programs\Python\Python[version]* for the current user. It includes IDLE (the default Python editor), the PIP package manager, and additional documentation. The installer also creates necessary shortcuts and file associations.

Customizing the installation allows changing these installation options and parameters.

4. Choose the optional installation features. Python works without these features, but adding them improves the program's usability.

Click **Next** to proceed to the Advanced Options screen.

5. The second part of customizing the installation includes advanced options.

Choose whether to install Python for all users. The option changes the install location to *C:\Program Files\Python[version]*. If selecting the location manually, a common choice is *C:\Python[version]* because it avoids spaces in the path, and all users can access it. Due to administrative rights, both paths may cause issues during package installation.

Other advanced options include creating shortcuts, file associations, and adding Python to PATH.

After picking the appropriate options, click **Install** to start the installation.

6. Select whether to disable the path length limit. Choosing this option will allow Python to bypass the 260-character **MAX_PATH** limit.

The option will not affect any other system settings, and disabling it resolves potential name-length issues. We recommend selecting the option and closing the setup.

Step 4: Add Python to Path (Optional)

If the Python installer does not include the **Add Python to PATH** checkbox or you have not selected that option, continue in this step. Otherwise, skip to the next step.

Adding the Python path to the PATH variable alleviates the need to use the full path to access the Python program in the command line. It instructs Windows to review all the folders added to the PATH environment variable and to look for the *python.exe* program in those folders.

To add Python to PATH, do the following:

1. In the **Start** menu, search for **Environment Variables** and press **Enter**.

2. Click **Environment Variables** to open the overview screen.

3. Double-click **Path** on the list to edit it.

Alternatively, select the variable and click the **Edit** button.

4. Double-click the first empty field and paste the Python installation folder path.

Alternatively, click the **New** button instead and paste the path.

5. Click **OK** to save the changes. If the command prompt is open, restart it for the following step.

Step 5: Verify Python Was Installed on Windows

The first way to verify that Python was installed successfully is through the command line. Open the command prompt and run the following command:

```
python --version
```

The output shows the installed Python version.

The second way is to use the GUI to verify the Python installation. Follow the steps below to run the Python interpreter or IDLE:

1. Navigate to the directory where Python was installed on the system.

2. Double-click *python.exe* (the Python interpreter) or IDLE.

3. The interpreter opens the command prompt and shows the following window:

Running IDLE opens Python's built-in IDE:

In both cases, the installed Python version shows on the screen, and the editor is ready for use.

Step 6: Verify PIP Was Installed

To verify whether PIP was installed, enter the following command in the command prompt:

```
pip --version
```

If it was installed successfully, you should see the PIP version number, the executable path, and the Python version:

PIP has not been installed yet if you get the following output:

```
'pip' is not recognized as an internal or external command,
Operable program or batch file.
```

If an older version of Python is installed or the PIP installation option is disabled during installation, PIP will not be available. To install PIP, see our article How to Install PIP on Windows.

Step 7: Install virtualenv (Optional)

Python software packages install system-wide by default. Consequently, whenever a single project-specific package is changed, it changes for all your Python projects.

The **virtualenv** package enables making isolated local virtual environments for Python projects. Virtual environments help avoid package conflicts and enable choosing specific package versions per project.

To install **virtualenv**, run the following command in the command prompt:

```
pip install virtualenv
```

Wait for the installation to complete. Once done, it is installed on the system and available for use.