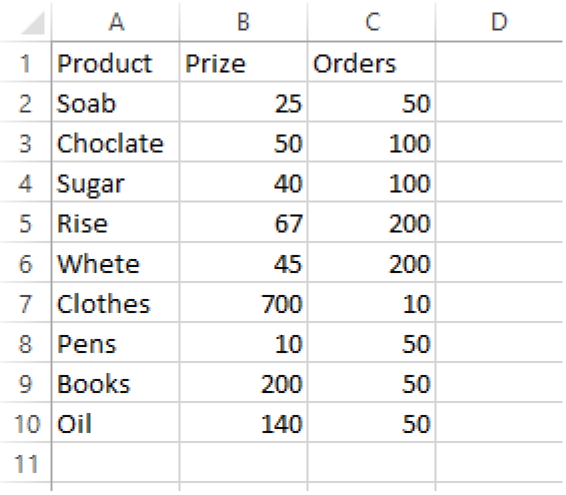
Practical No.1

Aim: Import legacy data from Excel and load in the targeted system.

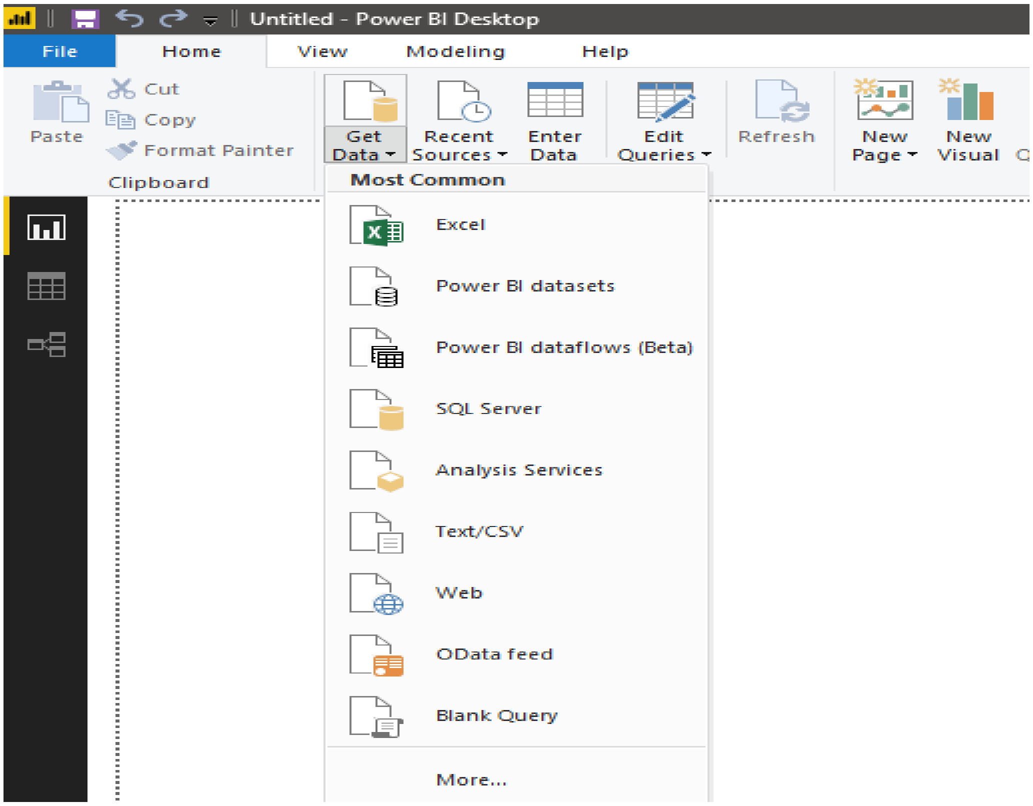
Step 1: For this first create Excel file.



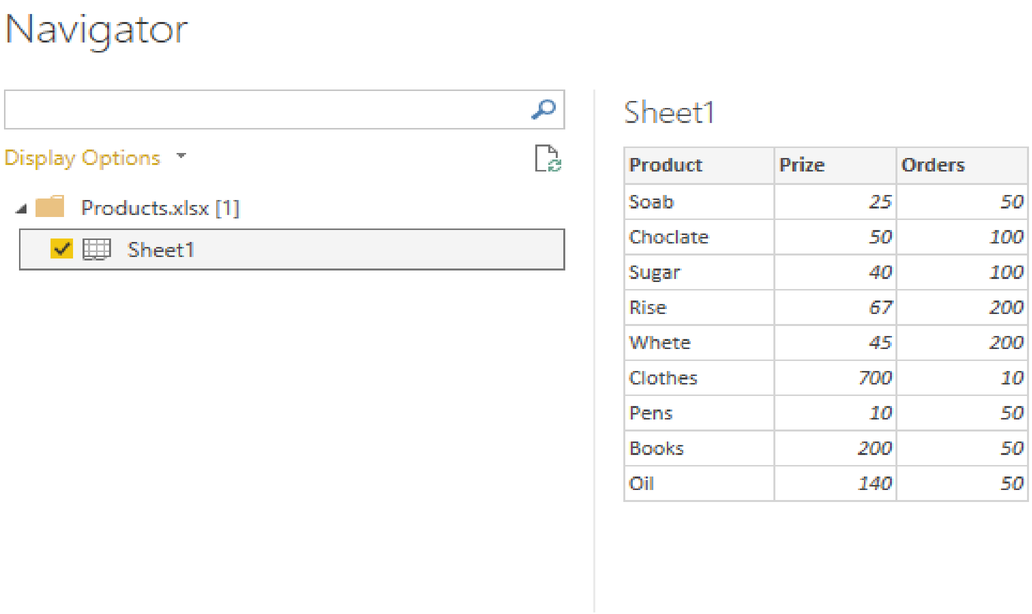
	A	B	C	D
1	Product	Prize	Orders	
2	Soab	25	50	
3	Choclate	50	100	
4	Sugar	40	100	
5	Rise	67	200	
6	Whete	45	200	
7	Clothes	700	10	
8	Pens	10	50	
9	Books	200	50	
10	Oil	140	50	
11				

Step 2: Open Power Bi Desktop.

Step 3: Go to Home -> Get data -> Excel and browse your excel file.



Step 4: In the navigation tab, select your (sheet1) from database (products.xlsx).

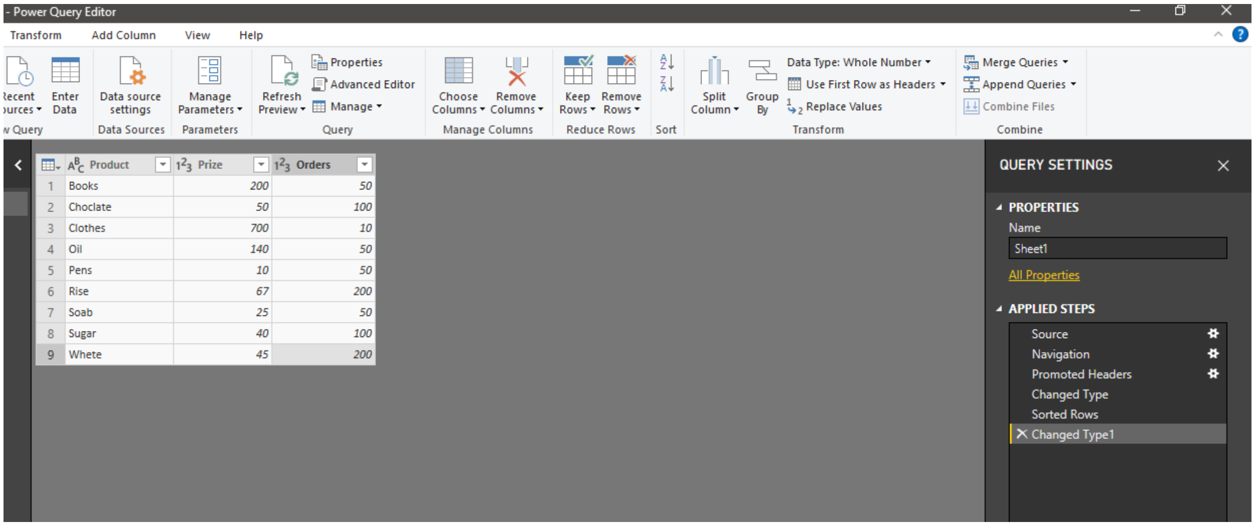


☑ Sheet1

Sheet1

Product	Prize	Orders
Soab	25	50
Choclate	50	100
Sugar	40	100
Rise	67	200
Whete	45	200
Clothes	700	10
Pens	10	50
Books	200	50
Oil	140	50

Step 5: You will obtain this screen for queries.



	Product	Prize	Orders
1	Books	200	50
2	Choclate	50	100
3	Clothes	700	10
4	Oil	140	50
5	Pens	10	50
6	Rise	67	200
7	Soab	25	50
8	Sugar	40	100
9	Whete	45	200

QUERY SETTINGS

PROPERTIES

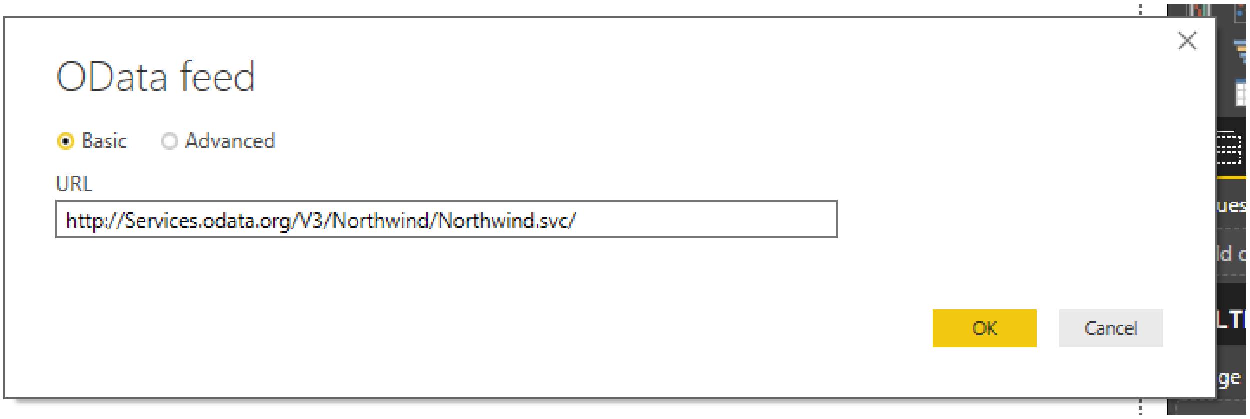
Name: Sheet1

All Properties

APPLIED STEPS

- Source
- Navigation
- Promoted Headers
- Changed Type
- Sorted Rows
- Changed Type1

Step 6: Step 3: Go to Home -> Get data -> Odata feed and type url.



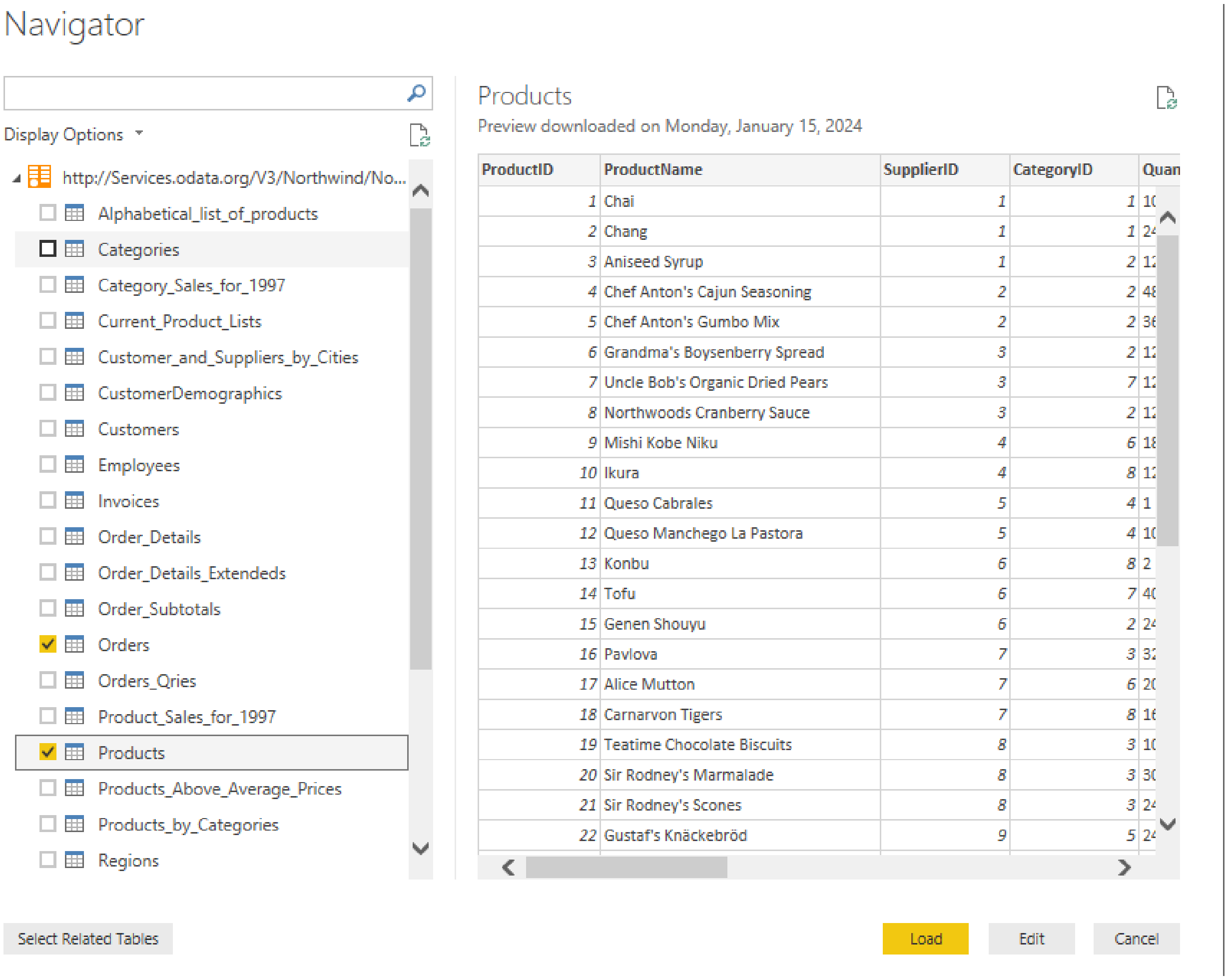
OData feed

● Basic ○ Advanced

URL

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

Step 7: From server Select orders and products table -> click on load



Navigator

http://Services.odata.org/V3/Northwind/No...

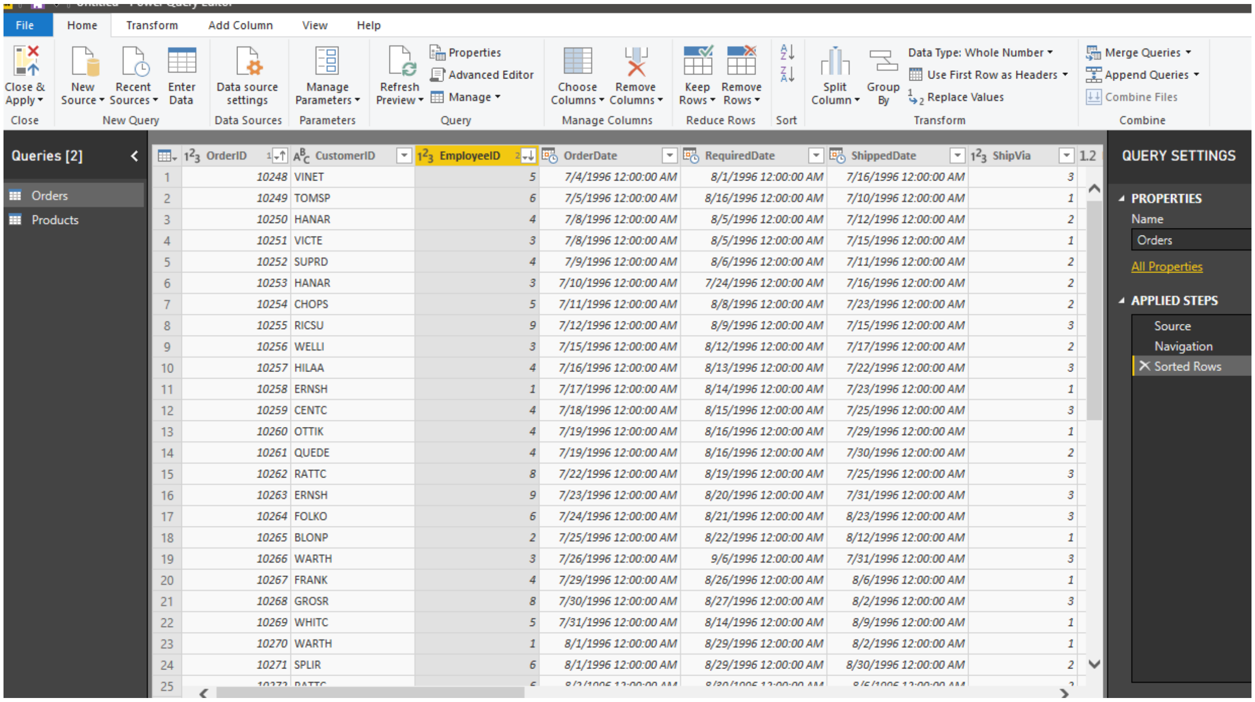
- □ Alphabetical_list_of_products
- □ Categories
- □ Category_Sales_for_1997
- □ Current_Product_Lists
- □ Customer_and_Suppliers_by_Cities
- □ CustomerDemographics
- □ Customers
- □ Employees
- □ Invoices
- □ Order_Details
- □ Order_Details_Extendeds
- □ Order_Subtotals
- ☑ Orders
- □ Orders_Qries
- □ Product_Sales_for_1997
- ☑ Products
- □ Products_Above_Average_Prices
- □ Products_by_Categories
- □ Regions

Products

Preview downloaded on Monday, January 15, 2024

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

Step 8: You will obtain this screen for queries.



Queries [2]
- Orders
- Products

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	1.2
1	10248	VINET	5	7/4/1996 12:00:00 AM	8/1/1996 12:00:00 AM	7/16/1996 12:00:00 AM	3	
2	10249	TOMSP	6	7/5/1996 12:00:00 AM	8/16/1996 12:00:00 AM	7/10/1996 12:00:00 AM	1	
3	10250	HANAR	4	7/8/1996 12:00:00 AM	8/5/1996 12:00:00 AM	7/12/1996 12:00:00 AM	2	
4	10251	VICTE	3	7/8/1996 12:00:00 AM	8/5/1996 12:00:00 AM	7/15/1996 12:00:00 AM	1	
5	10252	SUPRD	4	7/9/1996 12:00:00 AM	8/6/1996 12:00:00 AM	7/11/1996 12:00:00 AM	2	
6	10253	HANAR	3	7/10/1996 12:00:00 AM	7/24/1996 12:00:00 AM	7/16/1996 12:00:00 AM	2	
7	10254	CHOPS	5	7/11/1996 12:00:00 AM	8/8/1996 12:00:00 AM	7/23/1996 12:00:00 AM	2	
8	10255	RICSU	9	7/12/1996 12:00:00 AM	8/9/1996 12:00:00 AM	7/15/1996 12:00:00 AM	3	
9	10256	WELLI	3	7/15/1996 12:00:00 AM	8/12/1996 12:00:00 AM	7/17/1996 12:00:00 AM	2	
10	10257	HILAA	4	7/16/1996 12:00:00 AM	8/13/1996 12:00:00 AM	7/22/1996 12:00:00 AM	3	
11	10258	ERNSH	1	7/17/1996 12:00:00 AM	8/14/1996 12:00:00 AM	7/23/1996 12:00:00 AM	1	
12	10259	CENTC	4	7/18/1996 12:00:00 AM	8/15/1996 12:00:00 AM	7/25/1996 12:00:00 AM	3	
13	10260	OTTIK	4	7/19/1996 12:00:00 AM	8/16/1996 12:00:00 AM	7/29/1996 12:00:00 AM	1	
14	10261	QUEDE	4	7/19/1996 12:00:00 AM	8/16/1996 12:00:00 AM	7/30/1996 12:00:00 AM	2	
15	10262	RATTC	8	7/22/1996 12:00:00 AM	8/19/1996 12:00:00 AM	7/25/1996 12:00:00 AM	3	
16	10263	ERNSH	9	7/23/1996 12:00:00 AM	8/20/1996 12:00:00 AM	7/31/1996 12:00:00 AM	3	
17	10264	FOLKO	6	7/24/1996 12:00:00 AM	8/21/1996 12:00:00 AM	8/23/1996 12:00:00 AM	3	
18	10265	BLONP	2	7/25/1996 12:00:00 AM	8/22/1996 12:00:00 AM	8/12/1996 12:00:00 AM	1	
19	10266	WARTH	3	7/26/1996 12:00:00 AM	9/6/1996 12:00:00 AM	7/31/1996 12:00:00 AM	3	
20	10267	FRANK	4	7/29/1996 12:00:00 AM	8/26/1996 12:00:00 AM	8/6/1996 12:00:00 AM	1	
21	10268	GROSR	8	7/30/1996 12:00:00 AM	8/27/1996 12:00:00 AM	8/2/1996 12:00:00 AM	3	
22	10269	WHITC	5	7/31/1996 12:00:00 AM	8/14/1996 12:00:00 AM	8/9/1996 12:00:00 AM	1	
23	10270	WARTH	1	8/1/1996 12:00:00 AM	8/29/1996 12:00:00 AM	8/2/1996 12:00:00 AM	1	
24	10271	SPLIR	6	8/1/1996 12:00:00 AM	8/29/1996 12:00:00 AM	8/30/1996 12:00:00 AM	2	
25	10272	RATTC	6	8/2/1996 12:00:00 AM	8/30/1996 12:00:00 AM	8/6/1996 12:00:00 AM	2	

PROPERTIES

Name

Orders

All Properties

APPLIED STEPS

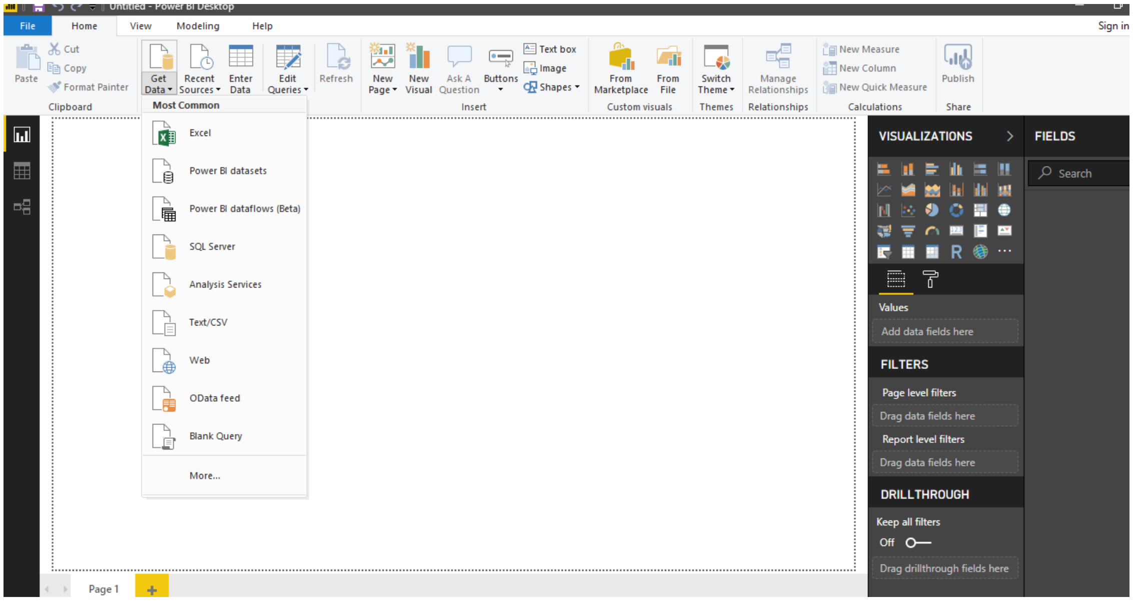
- Source
- Navigation
- Sorted Rows

PRACTICAL 2

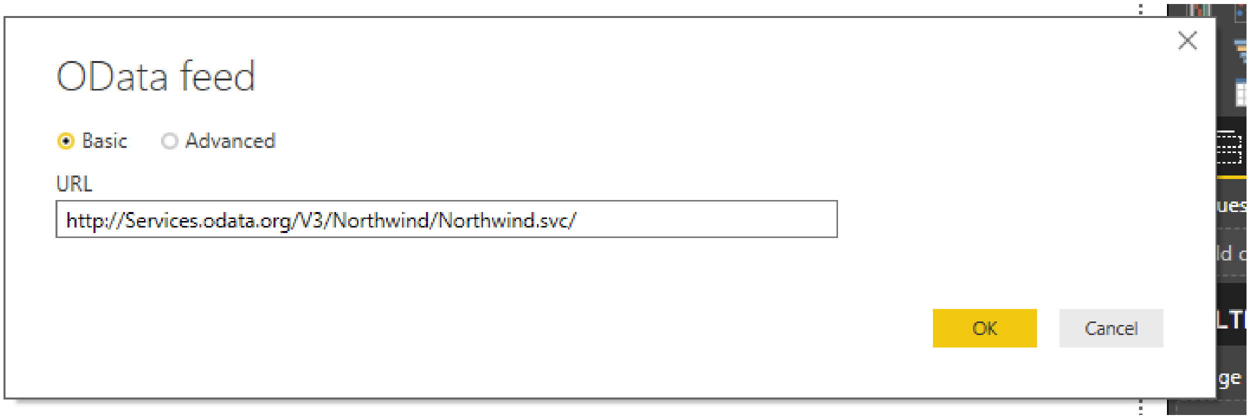
Aim:- Perform the extraction transformation and loading(ETL) process to construct database in the SQL server/Power BI.

Step 1: Open Power Bi desktop.

Step 2: Go to Home -> Get Data -> Odata Feed



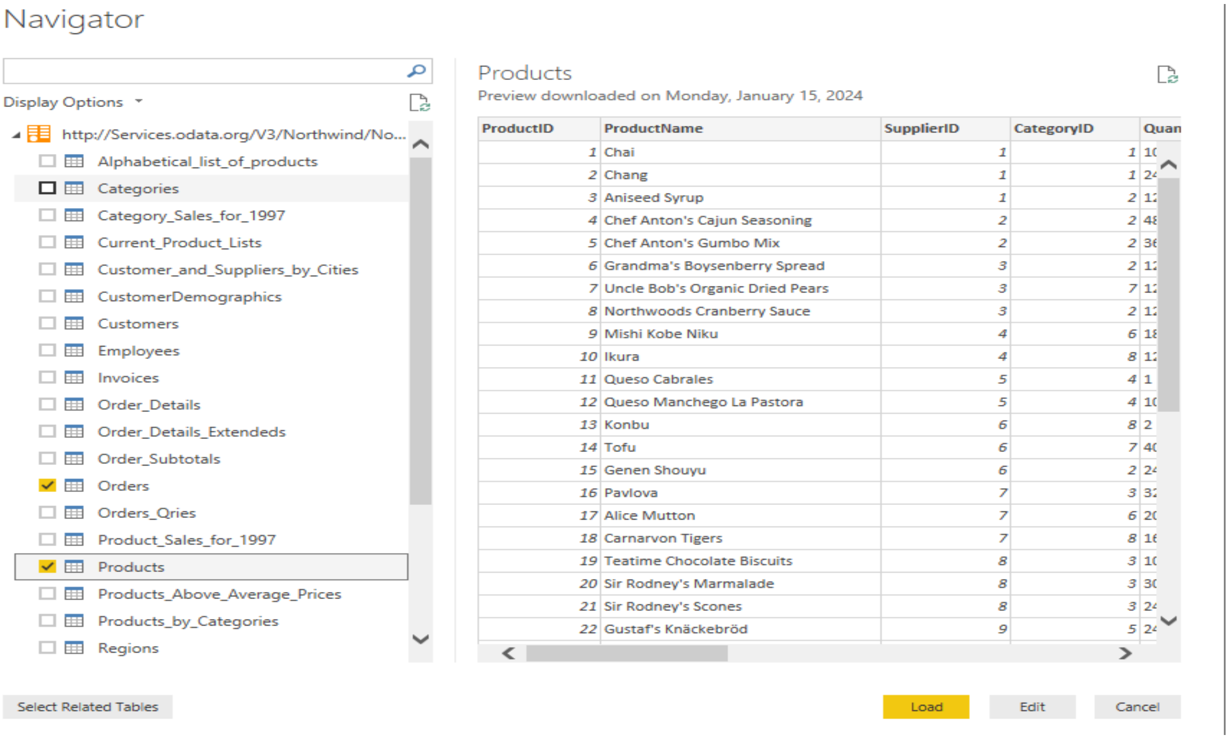
Step 3: Paste Url



URL

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

Step 4: From server Select order and products table -> click on edit .



Navigator

Display Options

http://Services.odata.org/V3/Northwind/No...

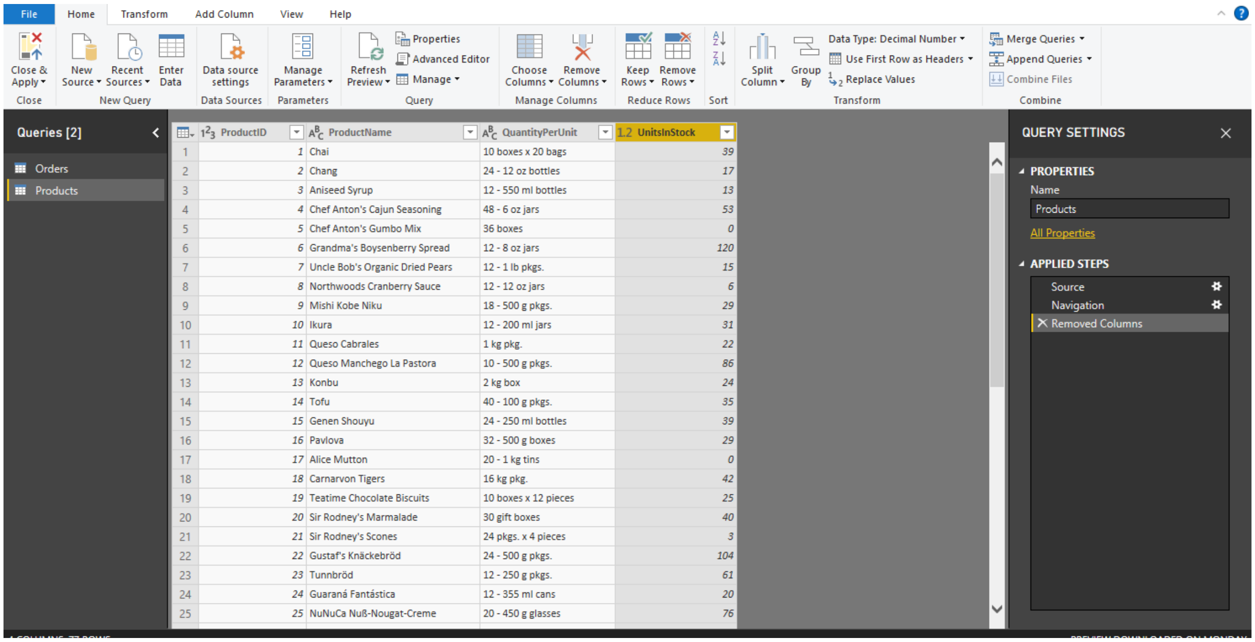
- □ Alphabetical_list_of_products
- □ Categories
- □ Category_Sales_for_1997
- □ Current_Product_Lists
- □ Customer_and_Suppliers_by_Cities
- □ CustomerDemographics
- □ Customers
- □ Employees
- □ Invoices
- □ Order_Details
- □ Order_Details_Extendeds
- □ Order_Subtotals
- ☑ Orders
- □ Orders_Qries
- □ Product_Sales_for_1997
- ☑ Products
- □ Products_Above_Average_Prices
- □ Products_by_Categories
- □ Regions

Products

Preview downloaded on Monday, January 15, 2024

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

Step 5: Select Product Table -> Delete Unrequired Columns -> Keep 4 Required Column.



Queries [2]

- Orders
- Products

	ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	1	Chai	10 boxes x 20 bags	39
2	2	Chang	24 - 12 oz bottles	17
3	3	Aniseed Syrup	12 - 550 ml bottles	13
4	4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	5	Chef Anton's Gumbo Mix	36 boxes	0
6	6	Grandma's Boysenberry Spread	12 - 8 oz jars	120
7	7	Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
8	8	Northwoods Cranberry Sauce	12 - 12 oz jars	6
9	9	Mishi Kobe Niku	18 - 500 g pkgs.	29
10	10	Ikura	12 - 200 ml jars	31
11	11	Queso Cabrales	1 kg pkg.	22
12	12	Queso Manchego La Pastora	10 - 500 g pkgs.	86
13	13	Konbu	2 kg box	24
14	14	Tofu	40 - 100 g pkgs.	35
15	15	Genen Shouyu	24 - 250 ml bottles	39
16	16	Pavlova	32 - 500 g boxes	29
17	17	Alice Mutton	20 - 1 kg tins	0
18	18	Carnarvon Tigers	16 kg pkg.	42
19	19	Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
20	20	Sir Rodney's Marmalade	30 gift boxes	40
21	21	Sir Rodney's Scones	24 pkgs. x 4 pieces	3
22	22	Gustaf's Knäckebröd	24 - 500 g pkgs.	104
23	23	Tunnbröd	12 - 250 g pkgs.	61
24	24	Guaraná Fantástica	12 - 355 ml cans	20
25	25	NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	76

QUERY SETTINGS

PROPERTIES

Name

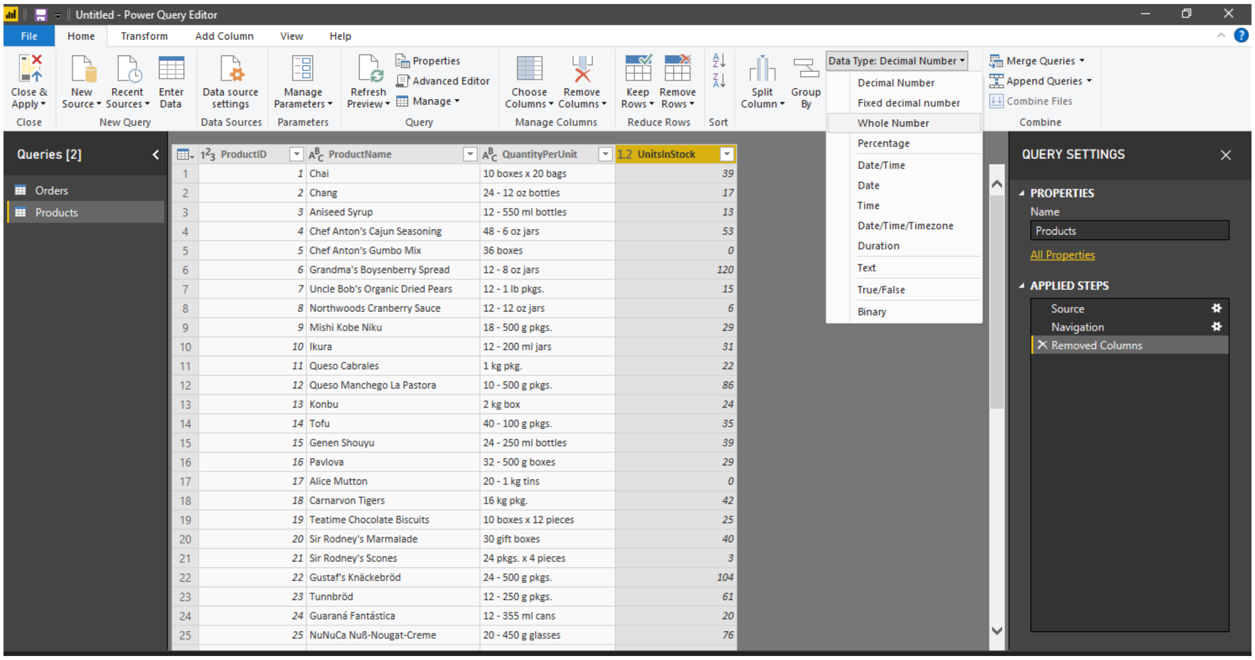
Products

All Properties

APPLIED STEPS

- Source
- Navigation
- Removed Columns

Step 6 : Change UnitsInStock Column to data type from decimal to whole number.



Data Type: Decimal Number

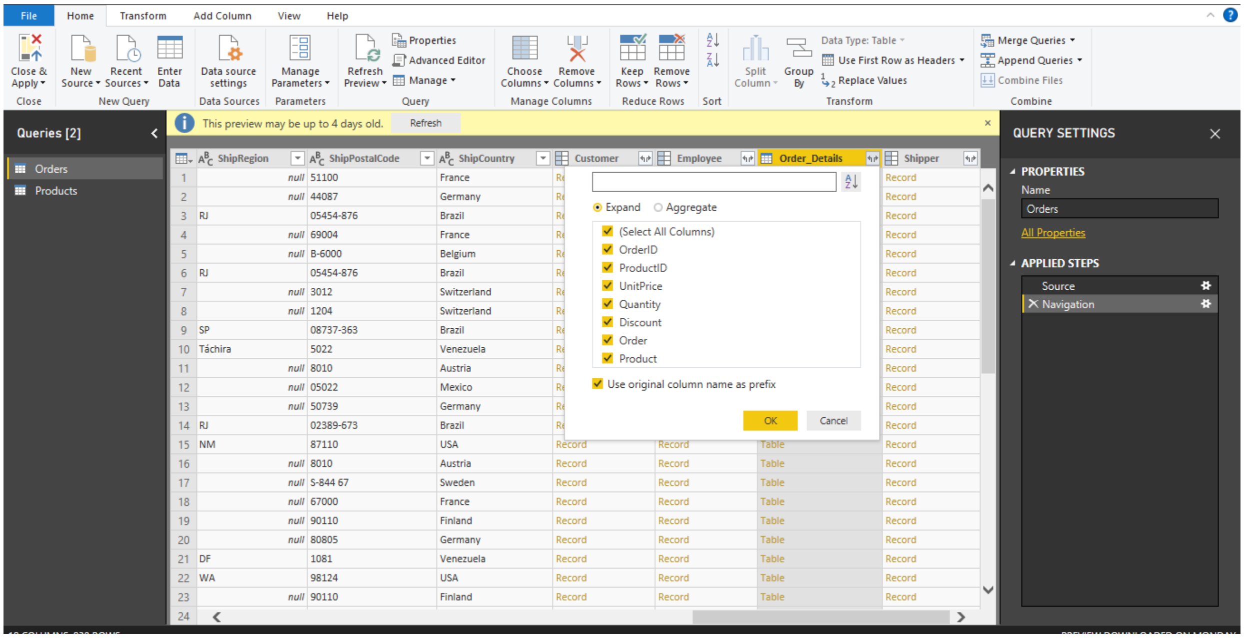
- Decimal Number
- Fixed decimal number
- Whole Number
- Percentage
- Date/Time
- Date
- Time
- Date/Time/Timezone
- Duration
- Text
- True/False
- Binary

	ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	1	Chai	10 boxes x 20 bags	39
2	2	Chang	24 - 12 oz bottles	17
3	3	Aniseed Syrup	12 - 550 ml bottles	13
4	4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	5	Chef Anton's Gumbo Mix	36 boxes	0
6	6	Grandma's Boysenberry Spread	12 - 8 oz jars	120
7	7	Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
8	8	Northwoods Cranberry Sauce	12 - 12 oz jars	6
9	9	Mishi Kobe Niku	18 - 500 g pkgs.	29
10	10	Ikura	12 - 200 ml jars	31
11	11	Queso Cabrales	1 kg pkg.	22
12	12	Queso Manchego La Pastora	10 - 500 g pkgs.	86
13	13	Konbu	2 kg box	24
14	14	Tofu	40 - 100 g pkgs.	35
15	15	Genen Shouyu	24 - 250 ml bottles	39
16	16	Pavlova	32 - 500 g boxes	29
17	17	Alice Mutton	20 - 1 kg tins	0
18	18	Carnarvon Tigers	16 kg pkg.	42
19	19	Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
20	20	Sir Rodney's Marmalade	30 gift boxes	40
21	21	Sir Rodney's Scones	24 pkgs. x 4 pieces	3
22	22	Gustaf's Knäckebröd	24 - 500 g pkgs.	104
23	23	Tunnbröd	12 - 250 g pkgs.	61
24	24	Guaraná Fantástica	12 - 355 ml cans	20
25	25	NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	76

Name: Products

Applied steps:
- Source
- Navigation
- Removed Columns

Step 7: Select Order Table -> Go to column OrderDetails -> Expand Column -> Click OK



This preview may be up to 4 days old.

	ShipRegion	ShipPostalCode	ShipCountry	Customer	Employee	Order_Details	Shipper
1	null	51100	France	Re			Record
2	null	44087	Germany	Re			Record
3	RJ	05454-876	Brazil	Re			Record
4	null	69004	France	Re			Record
5	null	B-6000	Belgium	Re			Record
6	RJ	05454-876	Brazil	Re			Record
7	null	3012	Switzerland	Re			Record
8	null	1204	Switzerland	Re			Record
9	SP	08737-363	Brazil	Re			Record
10	Táchira	5022	Venezuela	Re			Record
11	null	8010	Austria	Re			Record
12	null	05022	Mexico	Re			Record
13	null	50739	Germany	Re			Record
14	RJ	02389-673	Brazil	Re			Record
15	NM	87110	USA	Record	Record	Table	Record
16	null	8010	Austria	Record	Record	Table	Record
17	null	S-844 67	Sweden	Record	Record	Table	Record
18	null	67000	France	Record	Record	Table	Record
19	null	90110	Finland	Record	Record	Table	Record
20	null	80805	Germany	Record	Record	Table	Record
21	DF	1081	Venezuela	Record	Record	Table	Record
22	WA	98124	USA	Record	Record	Table	Record
23	null	90110	Finland	Record	Record	Table	Record
24							

● Expand ○ Aggregate

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

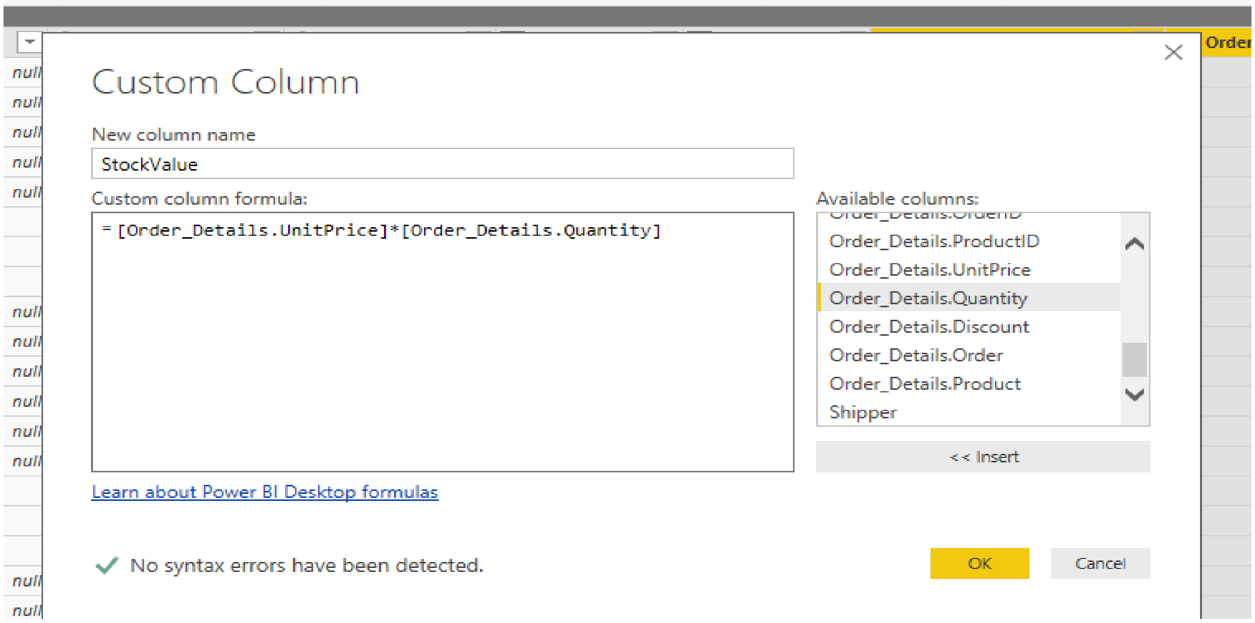
☑ Use original column name as prefix

OK Cancel

Name: Orders

Applied steps:
- Source
- Navigation

Step 8: Go to Add Column -> select Custom column -> Give name StockValue and Insert UnitPrice and Quantity -> click ok.



Custom Column

New column name

StockValue

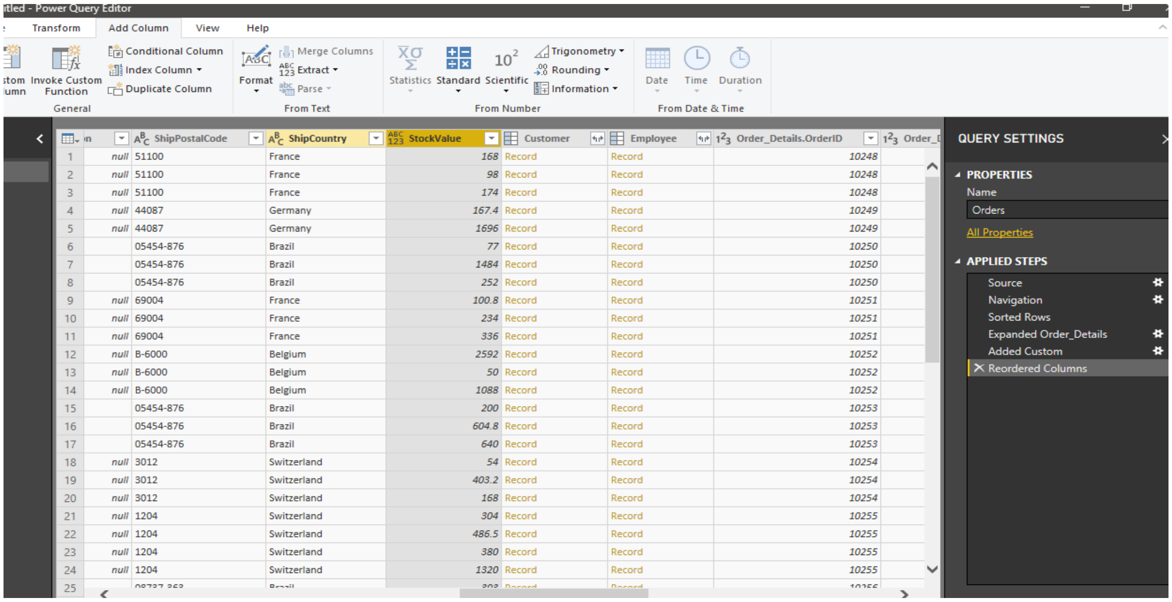
Custom column formula:

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

Available columns:

No syntax errors have been detected.

Step 9: Drag and drop the column StockValue.



	n	ShipPostalCode	ShipCountry	StockValue	Customer	Employee	Order_Details.OrderID
1	null	51100	France	168	Record	Record	10248
2	null	51100	France	98	Record	Record	10248
3	null	51100	France	174	Record	Record	10248
4	null	44087	Germany	167.4	Record	Record	10249
5	null	44087	Germany	1696	Record	Record	10249
6		05454-876	Brazil	77	Record	Record	10250
7		05454-876	Brazil	1484	Record	Record	10250
8		05454-876	Brazil	252	Record	Record	10250
9	null	69004	France	100.8	Record	Record	10251
10	null	69004	France	234	Record	Record	10251
11	null	69004	France	336	Record	Record	10251
12	null	B-6000	Belgium	2592	Record	Record	10252
13	null	B-6000	Belgium	50	Record	Record	10252
14	null	B-6000	Belgium	1088	Record	Record	10252
15		05454-876	Brazil	200	Record	Record	10253
16		05454-876	Brazil	604.8	Record	Record	10253
17		05454-876	Brazil	640	Record	Record	10253
18	null	3012	Switzerland	54	Record	Record	10254
19	null	3012	Switzerland	403.2	Record	Record	10254
20	null	3012	Switzerland	168	Record	Record	10254
21	null	1204	Switzerland	304	Record	Record	10255
22	null	1204	Switzerland	486.5	Record	Record	10255
23	null	1204	Switzerland	380	Record	Record	10255
24	null	1204	Switzerland	1320	Record	Record	10255

QUERY SETTINGS

PROPERTIES

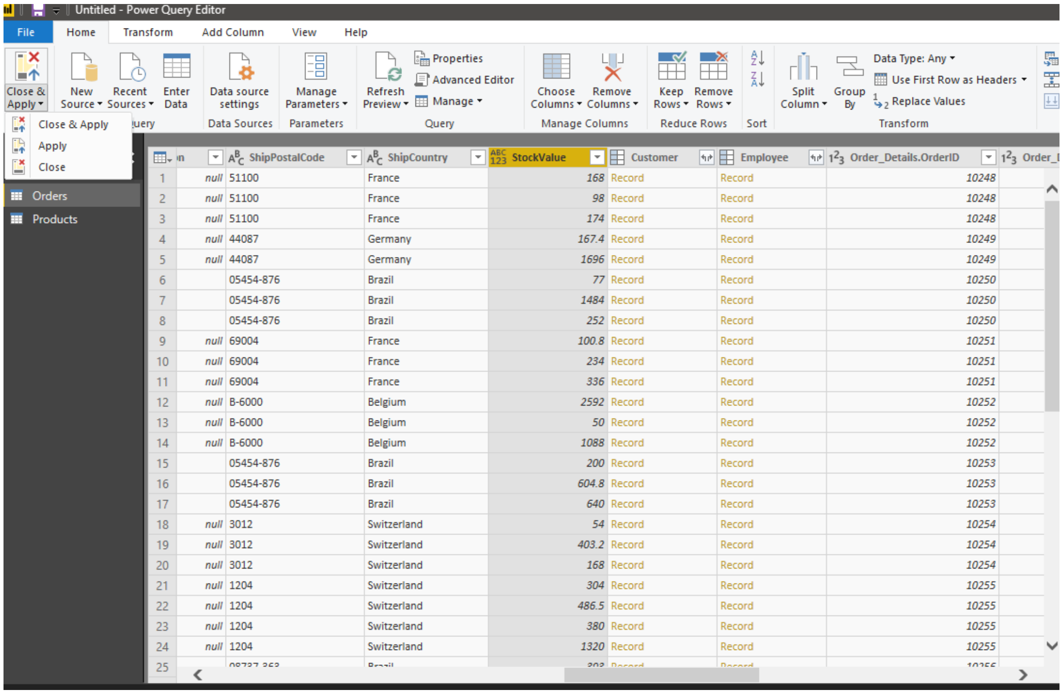
Name: Orders

All Properties

APPLIED STEPS

- Source
- Navigation
- Sorted Rows
- Expanded Order_Details
- Added Custom
- Reordered Columns

Step 10: select Close and Apply.



	n	ShipPostalCode	ShipCountry	StockValue	Customer	Employee	Order_Details.OrderID	Order_
1	null	51100	France	168	Record	Record	10248	
2	null	51100	France	98	Record	Record	10248	
3	null	51100	France	174	Record	Record	10248	
4	null	44087	Germany	167.4	Record	Record	10249	
5	null	44087	Germany	1696	Record	Record	10249	
6		05454-876	Brazil	77	Record	Record	10250	
7		05454-876	Brazil	1484	Record	Record	10250	
8		05454-876	Brazil	252	Record	Record	10250	
9	null	69004	France	100.8	Record	Record	10251	
10	null	69004	France	234	Record	Record	10251	
11	null	69004	France	336	Record	Record	10251	
12	null	B-6000	Belgium	2592	Record	Record	10252	
13	null	B-6000	Belgium	50	Record	Record	10252	
14	null	B-6000	Belgium	1088	Record	Record	10252	
15		05454-876	Brazil	200	Record	Record	10253	
16		05454-876	Brazil	604.8	Record	Record	10253	
17		05454-876	Brazil	640	Record	Record	10253	
18	null	3012	Switzerland	54	Record	Record	10254	
19	null	3012	Switzerland	403.2	Record	Record	10254	
20	null	3012	Switzerland	168	Record	Record	10254	
21	null	1204	Switzerland	304	Record	Record	10255	
22	null	1204	Switzerland	486.5	Record	Record	10255	
23	null	1204	Switzerland	380	Record	Record	10255	
24	null	1204	Switzerland	1320	Record	Record	10255	
25		08737-363	Brazil	303	Record	Record	10256	

Practical No 3

Aim: Create the cube with suitable dimension and fact tables based on OLAP.

1. 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 in your SQL Server. Downloading "Data Warehouse SQLScript.zip" from the article. https://www.codeproject.com/Articles/652108/Create-First-DataWareHouse

2. After downloading extract file in folder. Follow the given steps to run the query in SSMS (SQL Server Management Studio). Open SQL Server Management Studio 2012

3.Connect Database Engine

Click Connect. Open New Query editor

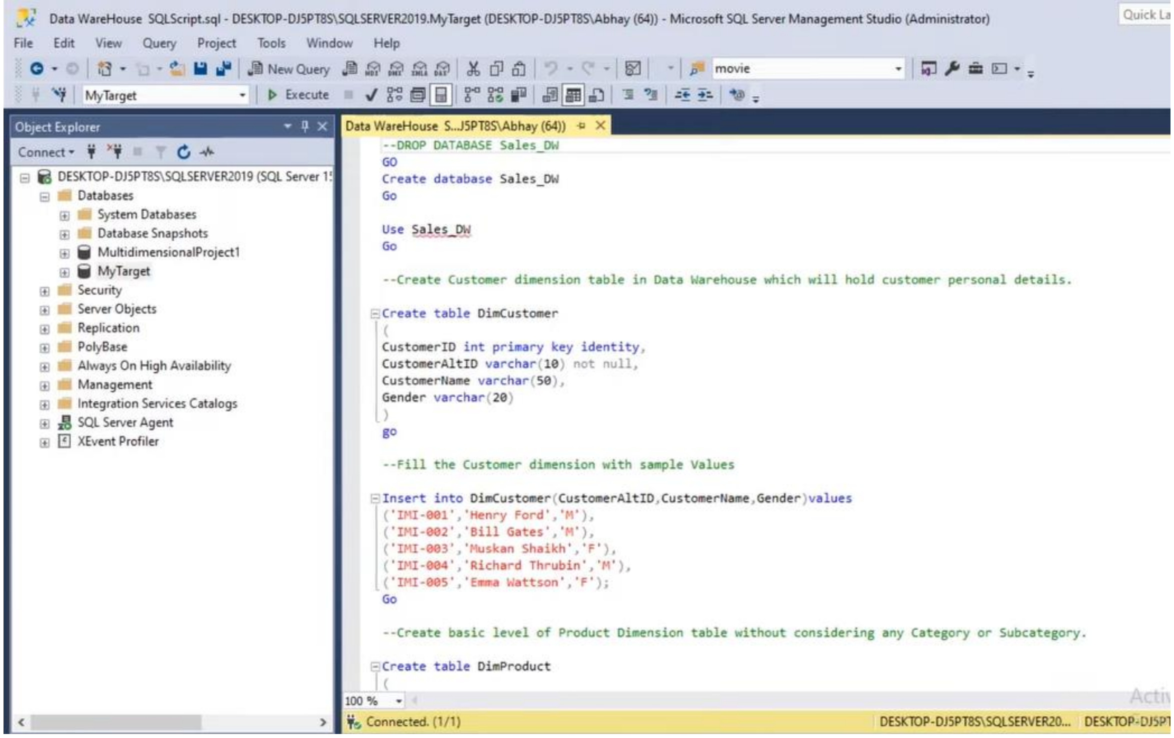
4. Copy paste Scripts given below in various steps in new query editor window one by one

5. To run the given SQL Script, press F5

6. It will create and populate "Sales DW" database on your SQL Server

OR

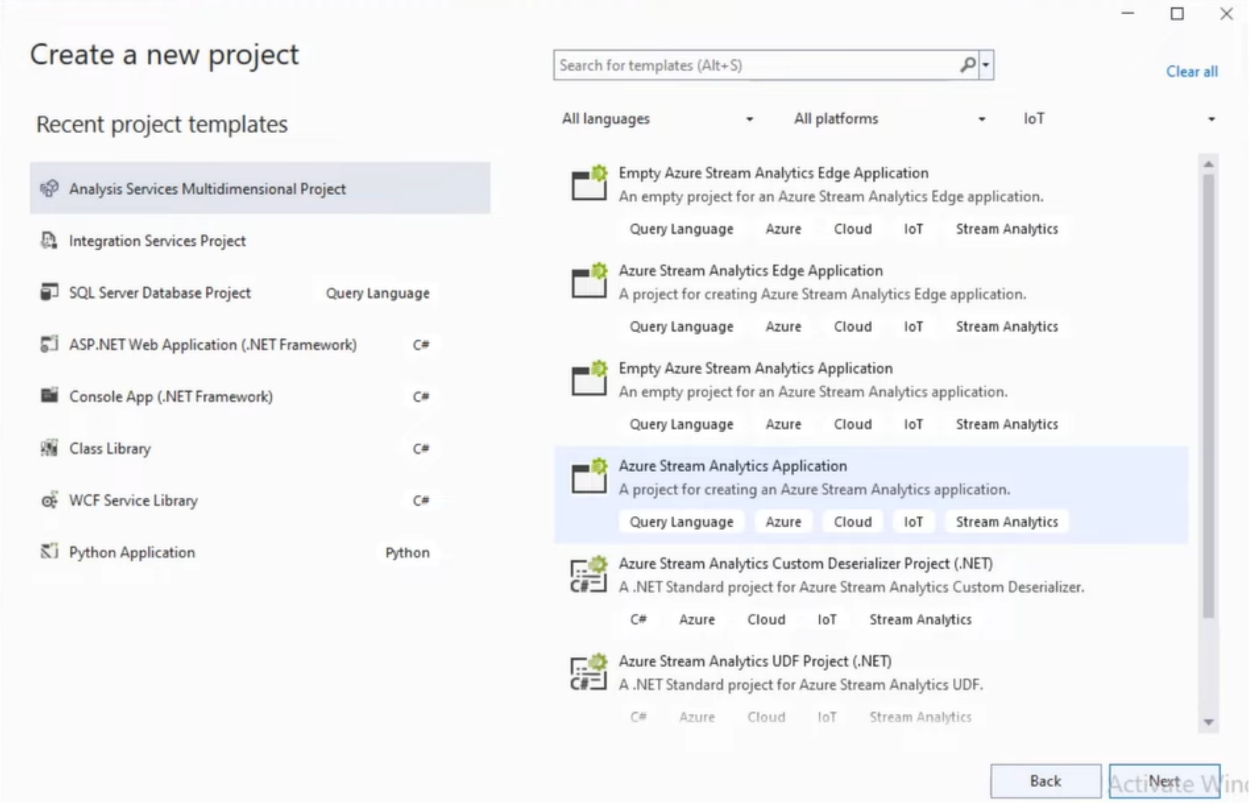
1. Go to the extracted SQL file and double click on it. 2. New SQL Query Editor will be opened containing



Data WareHouse SQLScript.sql - DESKTOP-DJ5PT8S\SQLSERVER2019.MyTarget (DESKTOP-DJ5PT8S\Abhay (64)) - Microsoft SQL Server Management Studio (Administrator)
File Edit View Query Project Tools Window Help
New Query
movie
MyTarget
Execute
Object Explorer
Connect
DESKTOP-DJ5PT8S\SQLSERVER2019 (SQL Server 1
Databases
System Databases
Database Snapshots
MultidimensionalProject1
MyTarget
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
Data WareHouse S...J5PT8S\Abhay (64))
100 %
Connected. (1/1)
DESKTOP-DJ5PT8S\SQLSERVER20...
DESKTOP-DJ5PT

```
--DROP DATABASE Sales_DW
GO
Create database Sales_DW
Go

Use Sales_DW
Go

--Create Customer dimension table in Data Warehouse which will hold customer personal details.

Create table DimCustomer
(
CustomerID int primary key identity,
CustomerAltID varchar(10) not null,
CustomerName varchar(50),
Gender varchar(20)
)
go

--Fill the Customer dimension with sample Values

Insert into DimCustomer(CustomerAltID,CustomerName,Gender)values
('IMI-001','Henry Ford','M'),
('IMI-002','Bill Gates','M'),
('IMI-003','Muskan Shaikh','F'),
('IMI-004','Richard Thrubin','M'),
('IMI-005','Emma Wattson','F');
Go

--Create basic level of Product Dimension table without considering any Category or Subcategory.

Create table DimProduct
(
```

Step 2: Start SSDT environment and create New Data Source. Go to SQL Server Data Tools → Right click and run as administrator Click on File → New Project In Business Intelligence → Analysis Services Multidimensional Project → appropriate project name → click OK.



Create a new project

Recent project templates

- Analysis Services Multidimensional Project
- Integration Services Project
- SQL Server Database Project — Query Language
- ASP.NET Web Application (.NET Framework) — C#
- Console App (.NET Framework) — C#
- Class Library — C#
- WCF Service Library — C#
- Python Application — Python

Search for templates (Alt+S)

All languages · All platforms · IoT

Empty Azure Stream Analytics Edge Application
An empty project for an Azure Stream Analytics Edge application.
Query Language · Azure · Cloud · IoT · Stream Analytics

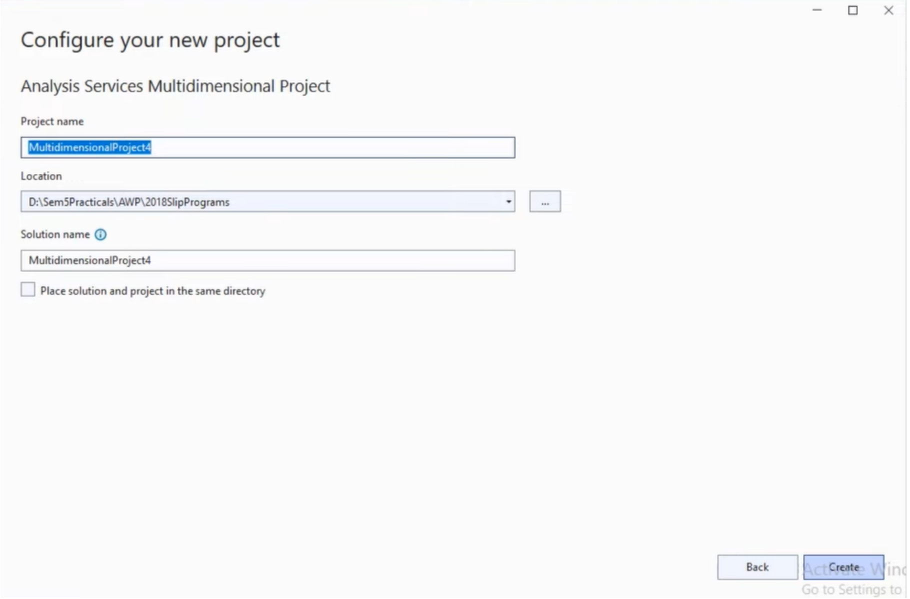
Azure Stream Analytics Edge Application
A project for creating Azure Stream Analytics Edge application.
Query Language · Azure · Cloud · IoT · Stream Analytics

Empty Azure Stream Analytics Application
An empty project for an Azure Stream Analytics application.
Query Language · Azure · Cloud · IoT · Stream Analytics

Azure Stream Analytics Application
A project for creating an Azure Stream Analytics application.
Query Language · Azure · Cloud · IoT · Stream Analytics

Azure Stream Analytics Custom Deserializer Project (.NET)
A .NET Standard project for Azure Stream Analytics Custom Deserializer.
C# · Azure · Cloud · IoT · Stream Analytics

Azure Stream Analytics UDF Project (.NET)
A .NET Standard project for Azure Stream Analytics UDF.
C# · Azure · Cloud · IoT · Stream Analytics



Configure your new project

Analysis Services Multidimensional Project

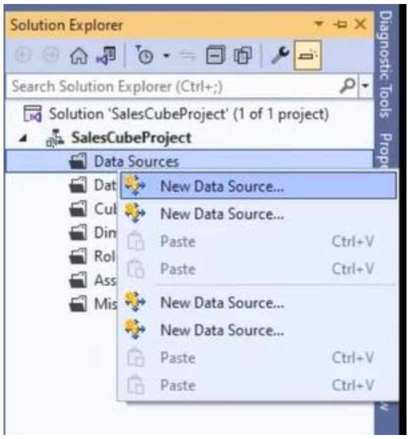
Project name: MultidimensionalProject4

Location: D:\Sem5Practicals\AWP\2018SlipPrograms

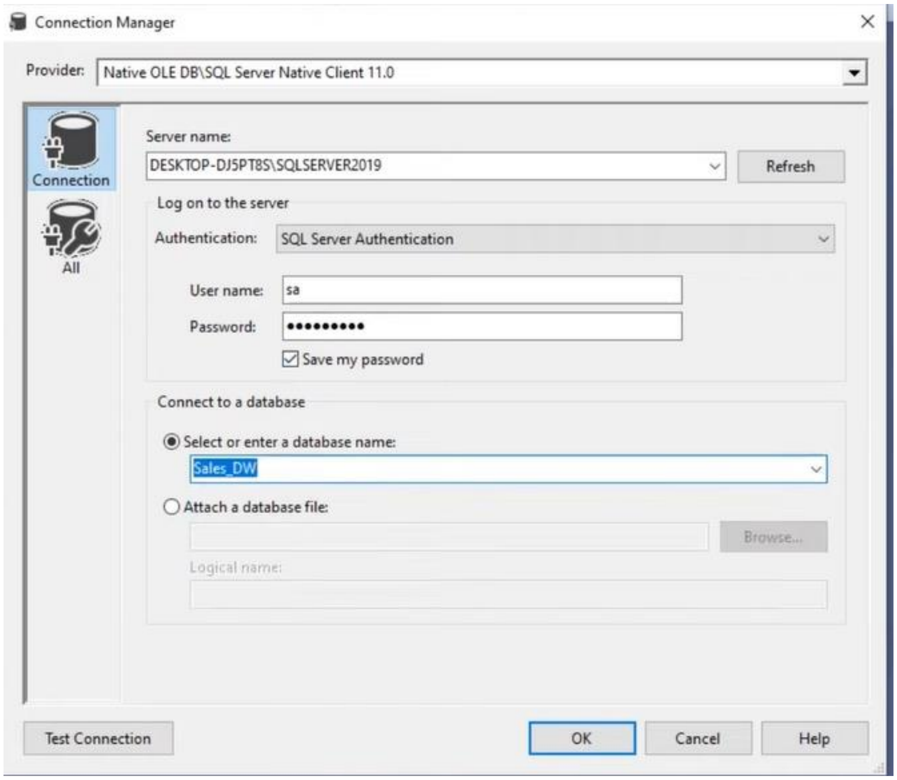
Solution name: MultidimensionalProject4

□ Place solution and project in the same directory

Right click on Data Sources in solution explorer → New Data Source.



Select Server → Name select Use SQL Server Authentication → Select or enter a database name (Sales DW). Click Next


Connection Manager

Provider: Native OLE DB\SQL Server Native Client 11.0

Server name: DESKTOP-DJ5PT8S\SQLSERVER2019

Log on to the server

Authentication: SQL Server Authentication

User name: sa

Password: •••••••••

☑ Save my password

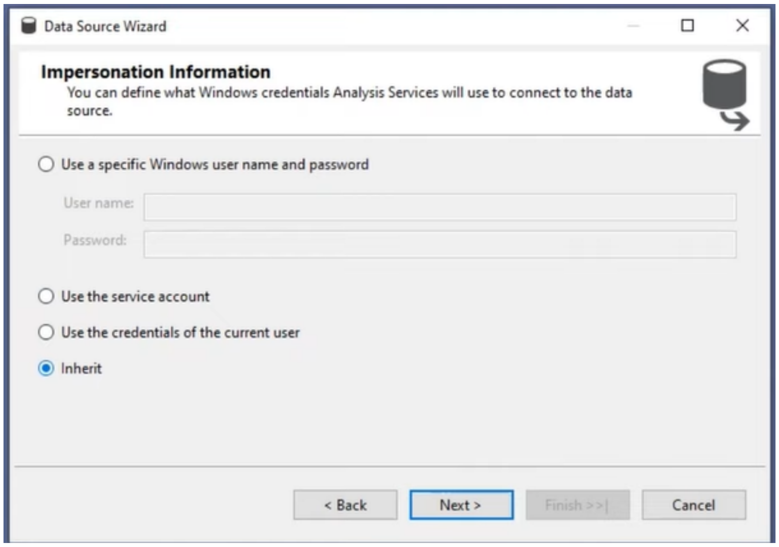
Connect to a database

● Select or enter a database name: Sales_DW

○ Attach a database file:

Logical name:

Select Inherit → Next



Data Source Wizard

Impersonation Information

You can define what Windows credentials Analysis Services will use to connect to the data source.

○ Use a specific Windows user name and password

User name:

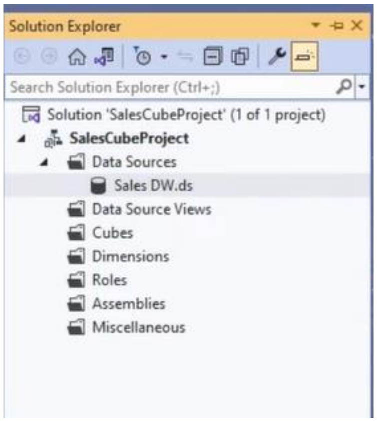
Password:

○ Use the service account

○ Use the credentials of the current user

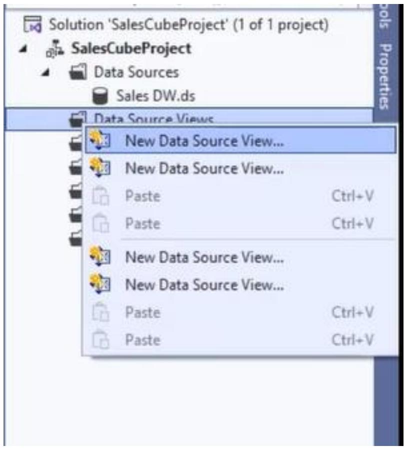
● Inherit

Sales_DW.ds gets created under Data Sources in Solution Explorer



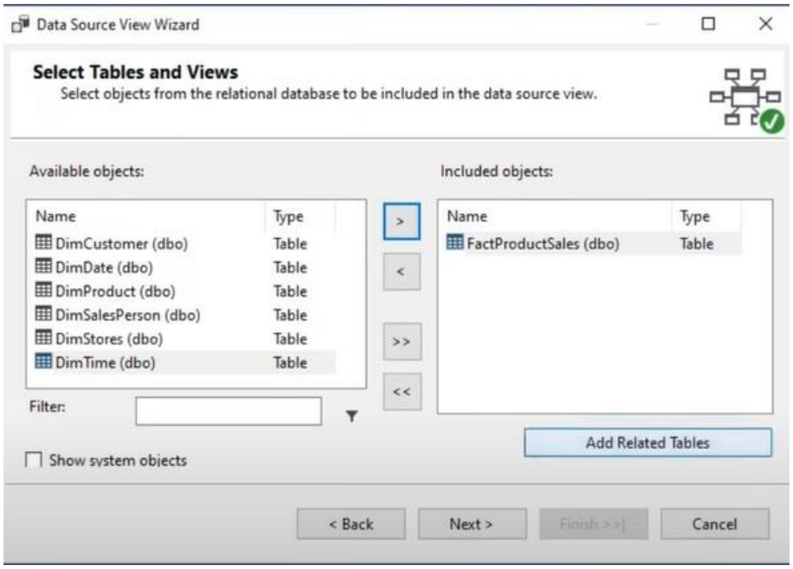
Solution 'SalesCubeProject' (1 of 1 project)
- SalesCubeProject
  - Data Sources
    - Sales DW.ds
  - Data Source Views
  - Cubes
  - Dimensions
  - Roles
  - Assemblies
  - Miscellaneous

Step 3: Creating New Data Source View in Solution explorer right click on Data Source View Select New Data



Solution 'SalesCubeProject' (1 of 1 project)
- SalesCubeProject
  - Data Sources
    - Sales DW.ds
  - Data Source Views

Click Next Select FactProductSales(dbo) from Available objects and put in Includes Objects by click.



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

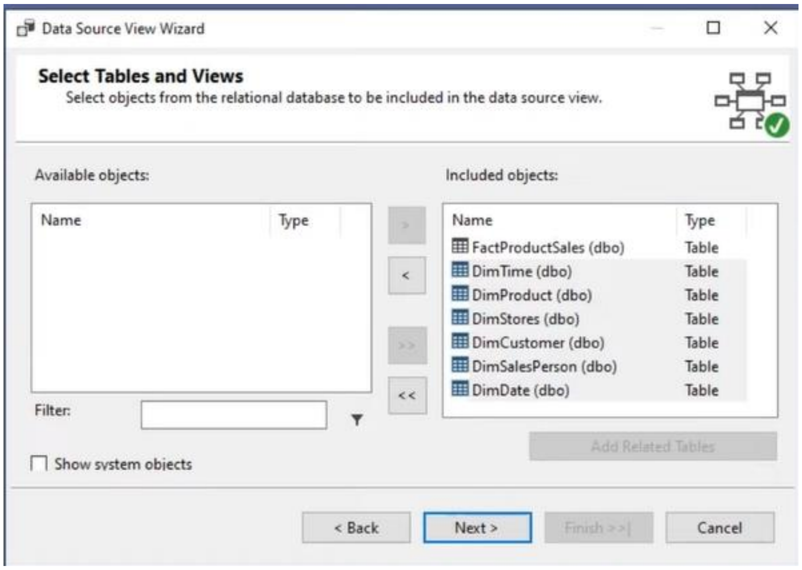
Name	Type
DimCustomer (dbo)	Table
DimDate (dbo)	Table
DimProduct (dbo)	Table
DimSalesPerson (dbo)	Table
DimStores (dbo)	Table
DimTime (dbo)	Table

Included objects:

Name	Type
FactProductSales (dbo)	Table

Filter:

□ Show system objects



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

Name	Type

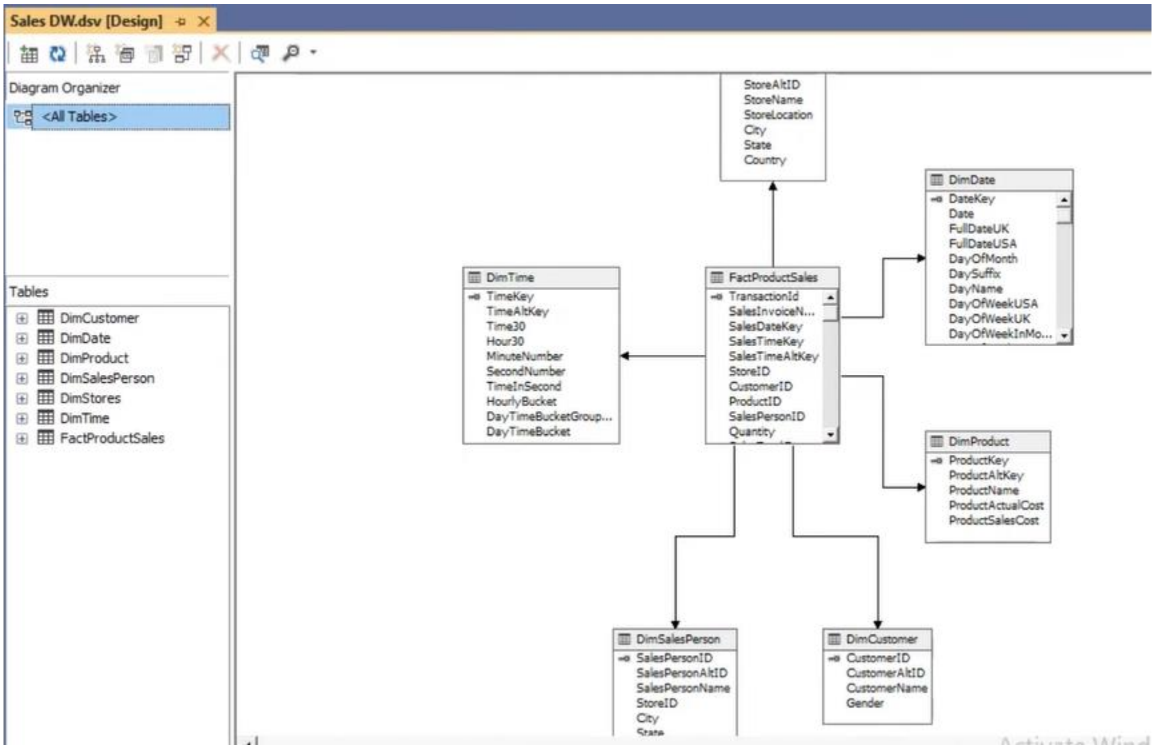
Included objects:

Name	Type
FactProductSales (dbo)	Table
DimTime (dbo)	Table
DimProduct (dbo)	Table
DimStores (dbo)	Table
DimCustomer (dbo)	Table
DimSalesPerson (dbo)	Table
DimDate (dbo)	Table

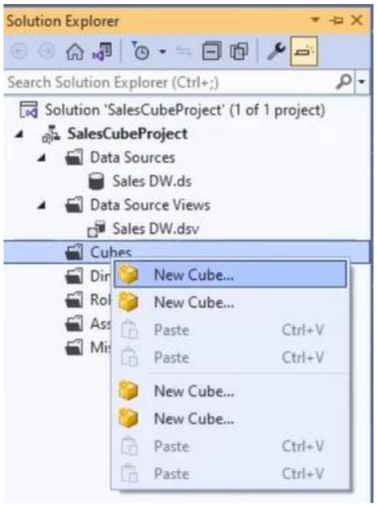
Filter:

□ Show system objects

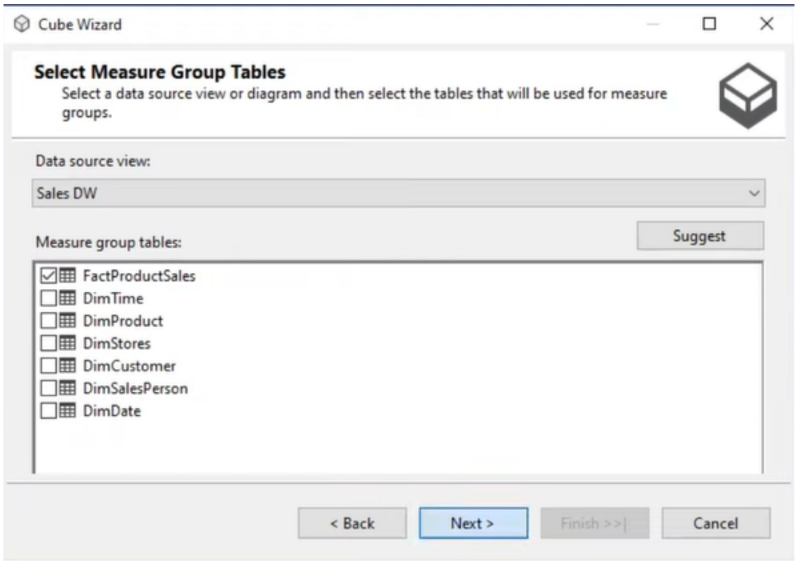
Click Finish Sales DW.dsv appears in Data Source Views in Solution Explorer



Step 4: Creating new cube Right click on cubes → New Cube



Select Use existing tables in Select Creation Method → Next In Select Measure Group Tables → Select FactProductSales → Click Next



Cube Wizard

Select Measure Group Tables

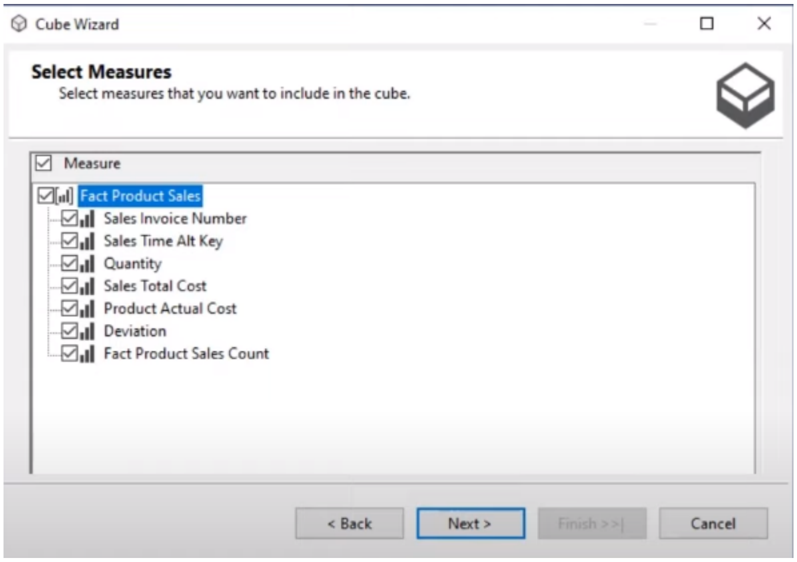
Select a data source view or diagram and then select the tables that will be used for measure groups.

Data source view: Sales DW

Measure group tables:

- ☑ FactProductSales
- □ DimTime
- □ DimProduct
- □ DimStores
- □ DimCustomer
- □ DimSalesPerson
- □ DimDate

In Select Measures → check all measures → Next



Cube Wizard

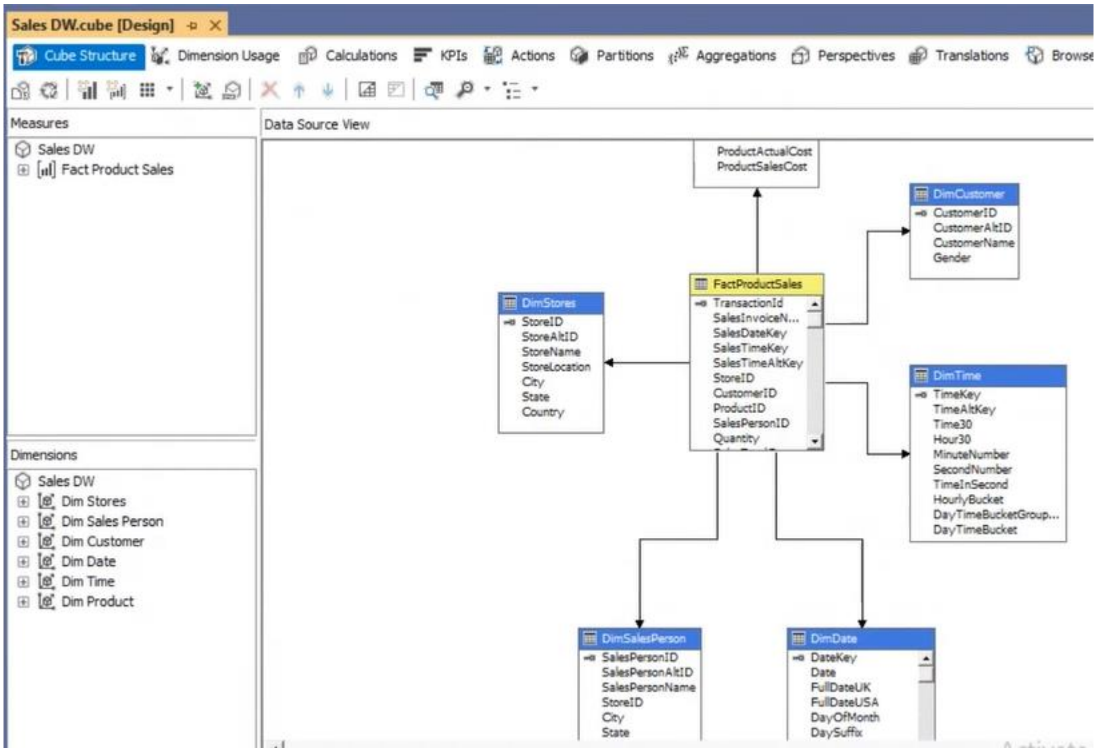
Select Measures

Select measures that you want to include in the cube.

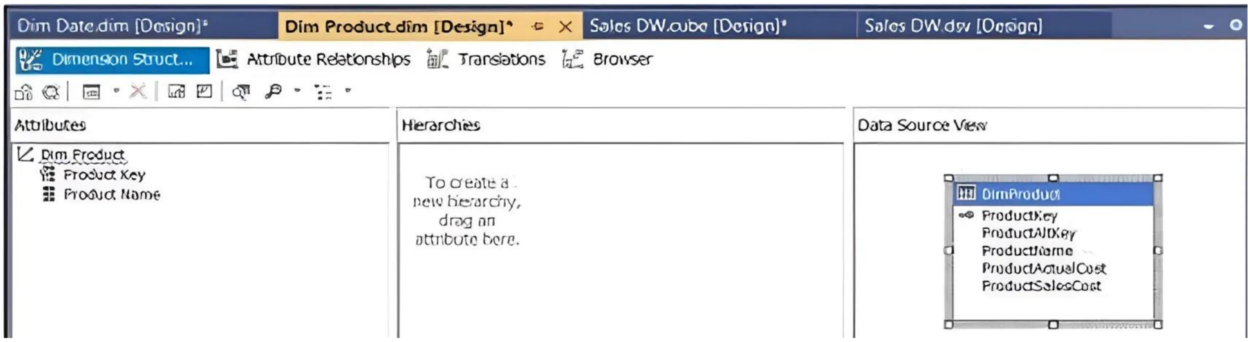
☑ Measure

- ☑ Fact Product Sales
  - ☑ Sales Invoice Number
  - ☑ Sales Time Alt Key
  - ☑ Quantity
  - ☑ Sales Total Cost
  - ☑ Product Actual Cost
  - ☑ Deviation
  - ☑ Fact Product Sales Count

In Select New Dimensions → Check all Dimensions→ Next. Click on Finish. Sales_DW.cube is created.

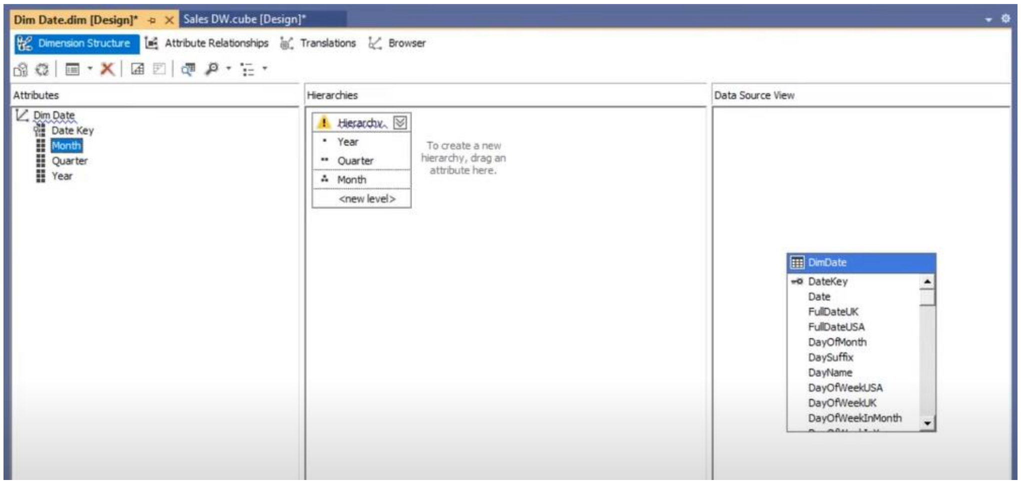


Step 5: Dimension Modification In dimension tab → Double Click Dim Product.dim Step 6: Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side.

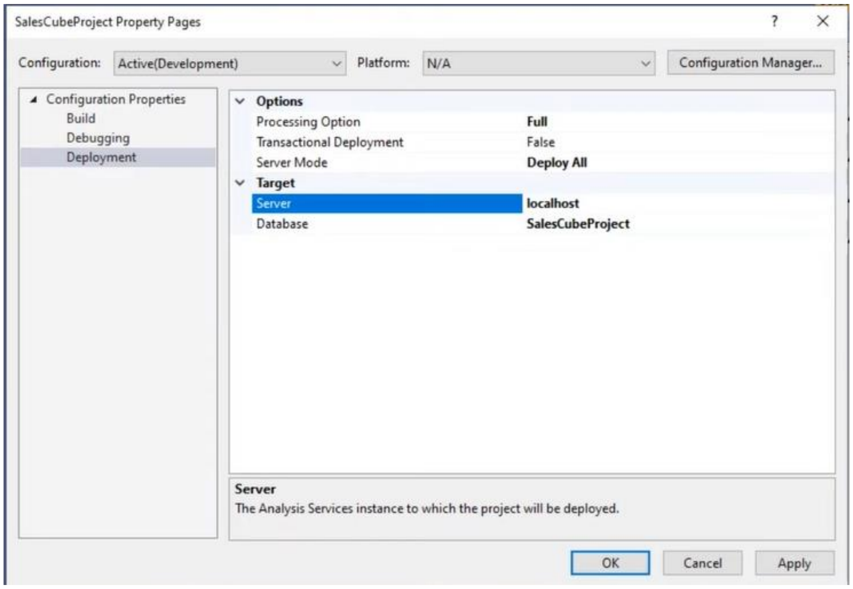


Step 7: Creating Attribute Hierarchy in Date Dimension.

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 Cube Right click on Project name → Properties Do following changes and click on Apply & OK

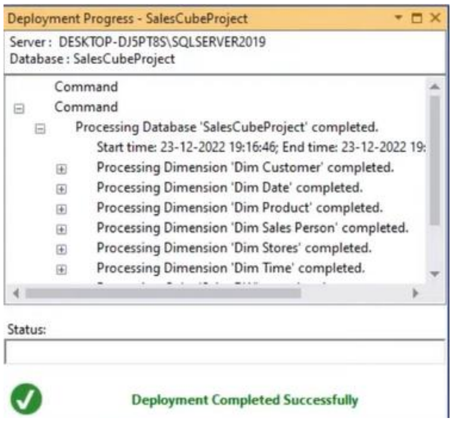


Configuration: Active(Development) Platform: N/A

Options	
Processing Option	**Full**
Transactional Deployment	False
Server Mode	**Deploy All**
Target	
Server	**localhost**
Database	**SalesCubeProject**

Server
The Analysis Services instance to which the project will be deployed.

Right click on project name → Deploy



```
Server :  DESKTOP-DJ5PT8S\SQLSERVER2019
Database : SalesCubeProject

Command
Command
    Processing Database 'SalesCubeProject' completed.
        Start time: 23-12-2022 19:16:46; End time: 23-12-2022 19:
        Processing Dimension 'Dim Customer' completed.
        Processing Dimension 'Dim Date' completed.
        Processing Dimension 'Dim Product' completed.
        Processing Dimension 'Dim Sales Person' completed.
        Processing Dimension 'Dim Stores' completed.
        Processing Dimension 'Dim Time' completed.
```

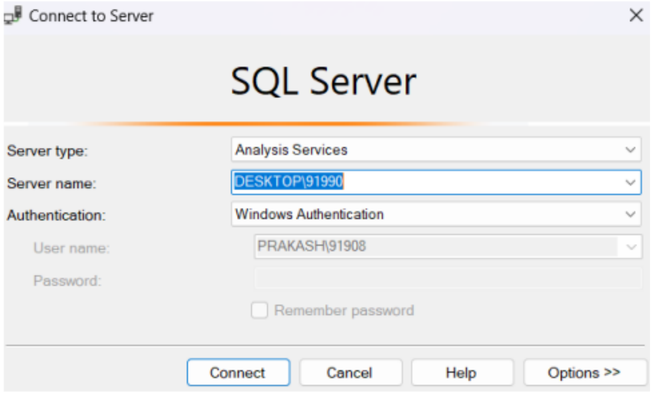
✓ Deployment Completed Successfully

PRACTICAL NO:4

Aim: a) Create the ETL map and set up the schedule for execution.

b) Execute the MDX queries to extract the data from the Datawarehouse.

Step 1: Open SQL server Management studio and Connect to Analysis Services.



Connect to Server

SQL Server

Server type: Analysis Services

Server name: DESKTOP\91990

Authentication: Windows Authentication

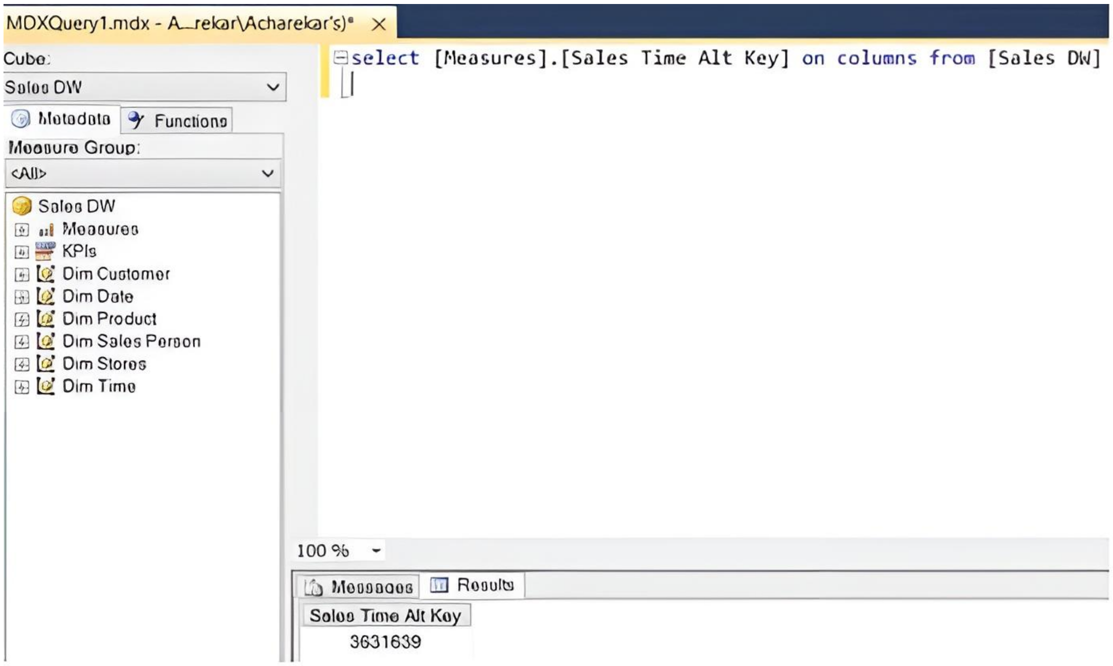
User name: PRAKASH\91908

Password:

□ Remember password

Step 2: Click on new query -> type following queries based on Sales_DW -> then click on Execute.

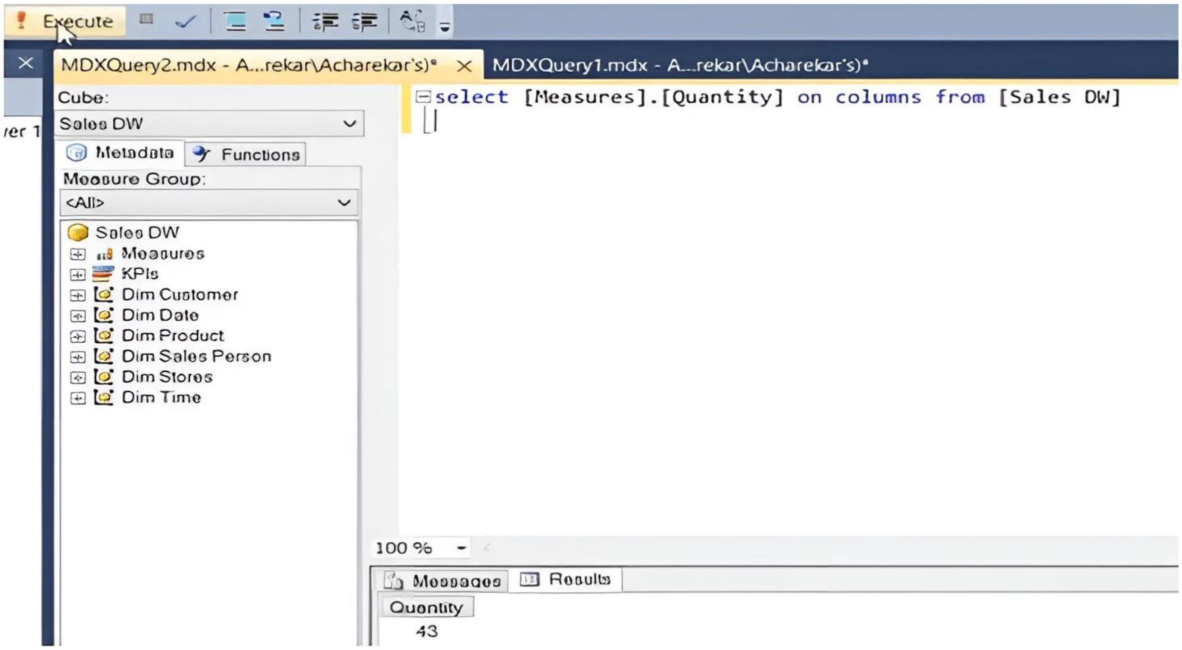
1.select[Measures].[Sales Time Alt Key] on columns from [Sales DW]



```
select [Measures].[Sales Time Alt Key] on columns from [Sales DW]
```

```
Sales Time Alt Key
3631639
```

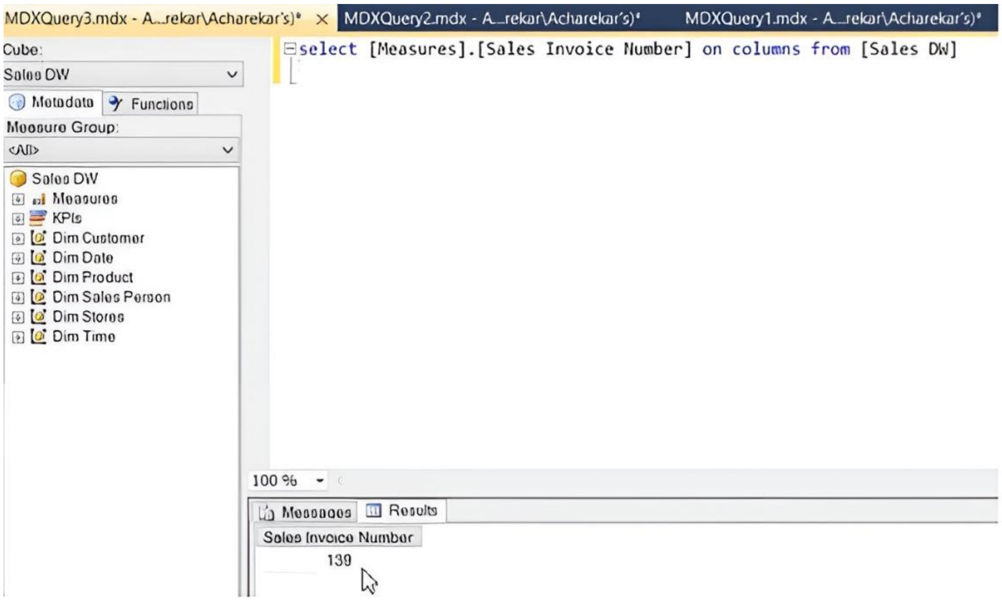
2. select [Measures].[Quantity] columns from [Sales DW]



```
select [Measures].[Quantity] on columns from [Sales DW]
```

```
Quantity
43
```

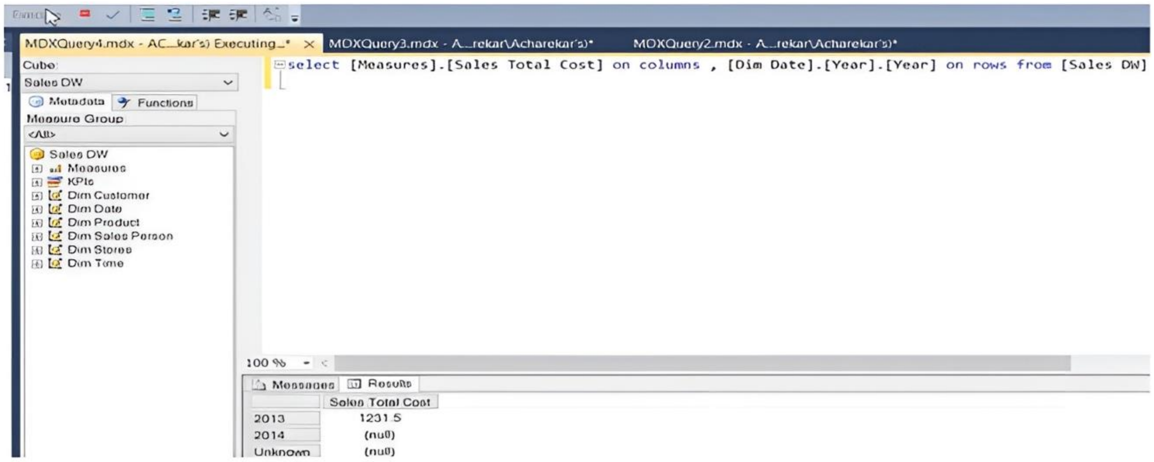
3. select [Measures].[Sales Invoice Number] columns from [Sales DW]



```
select [Measures].[Sales Invoice Number] on columns from [Sales DW]
```

```
Sales Invoice Number
139
```

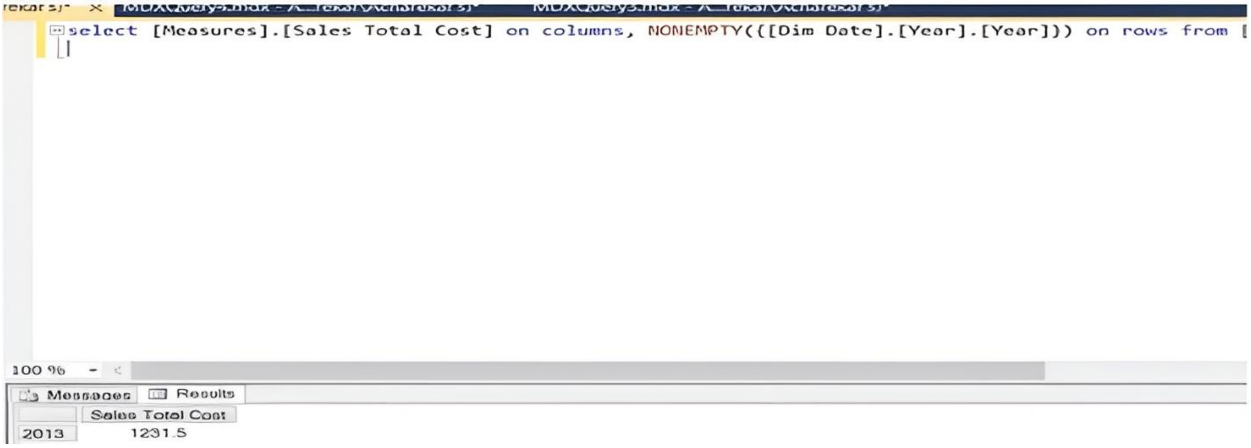
4. Select [Measures].[Sales Total Cost] on columns, [Dim Date].[Year].[Year] on rows from [Sales_DW]



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

```
         Sales Total Cost
2013     1231.5
2014     (null)
Unknown  (null)
```

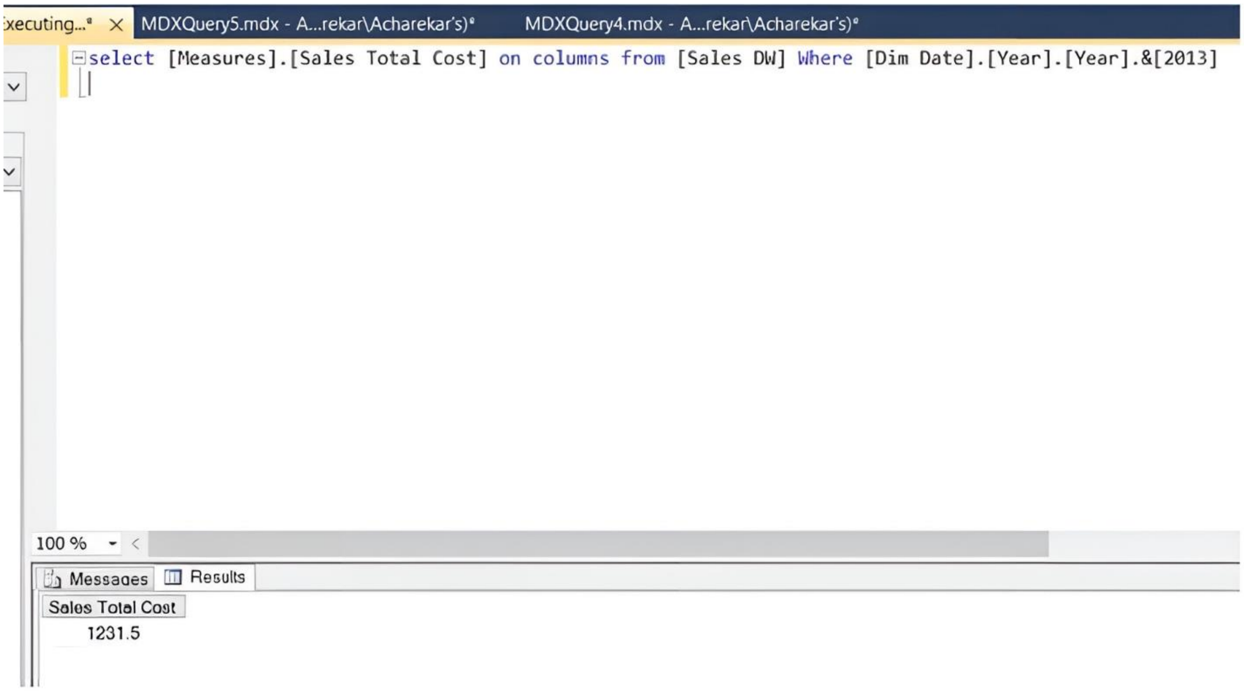
5. Select [Measures].[Sales Total Cost] on columns, NONEMPTY({[Dim Date].[Year].[Year]}) on rows from [Sales_DW]



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

```
      Sales Total Cost
2013  1231.5
```

6.Select [Measures].[Sales Total Cost] on columns from [Sales_DW] where [Dim Date].[Year].[Year].&[2013]



xecuting...*
MDXQuery5.mdx - A...rekar\Acharekar's)*
MDXQuery4.mdx - A...rekar\Acharekar's)*
100 %
Messages
Results

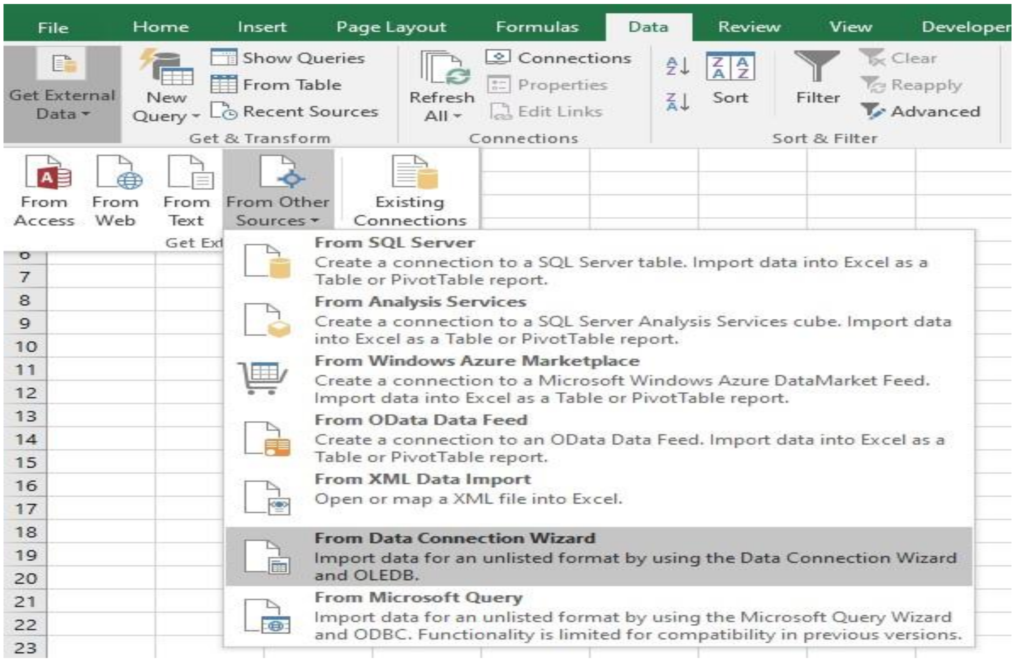
```
select [Measures].[Sales Total Cost] on columns from [Sales DW] Where [Dim Date].[Year].[Year].&[2013]
```

```
Sales Total Cost
1231.5
```

PRACTICAL NO: 5(A)

Aim: Import the Data Warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.

Step 1: Open Excel Go to Data tab → Get External Data → From Other Sources → From Data Connection Wizard



From SQL Server
Create a connection to a SQL Server table. Import data into Excel as a Table or PivotTable report.

From Analysis Services
Create a connection to a SQL Server Analysis Services cube. Import data into Excel as a Table or PivotTable report.

From Windows Azure Marketplace
Create a connection to a Microsoft Windows Azure DataMarket Feed. Import data into Excel as a Table or PivotTable report.

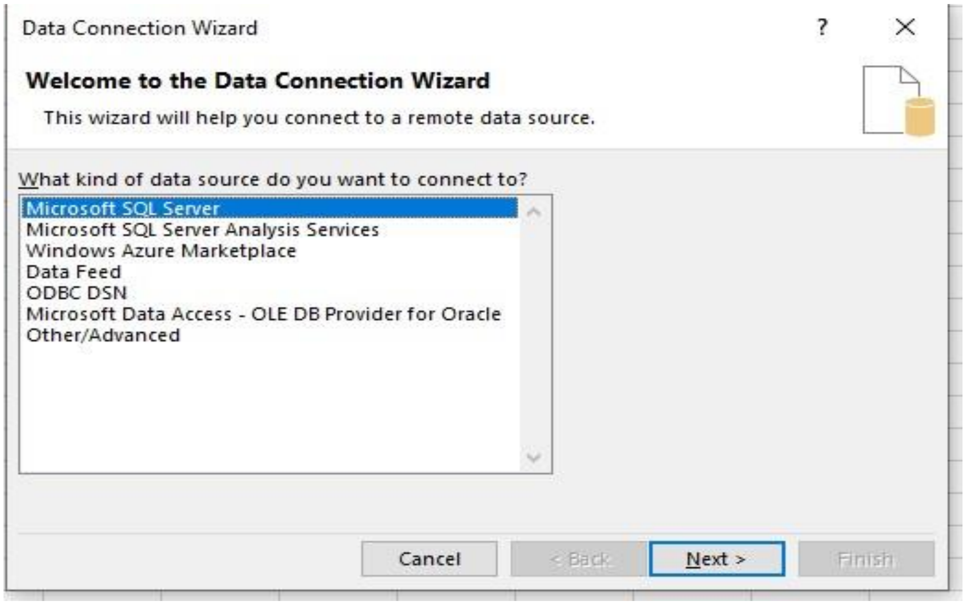
From OData Data Feed
Create a connection to an OData Data Feed. Import data into Excel as a Table or PivotTable report.

From XML Data Import
Open or map a XML file into Excel.

From Data Connection Wizard
Import data for an unlisted format by using the Data Connection Wizard and OLEDB.

From Microsoft Query
Import data for an unlisted format by using the Microsoft Query Wizard and ODBC. Functionality is limited for compatibility in previous versions.

Step 2: In Data Connection Wizard → Select Microsoft SQL Server → Click on Next



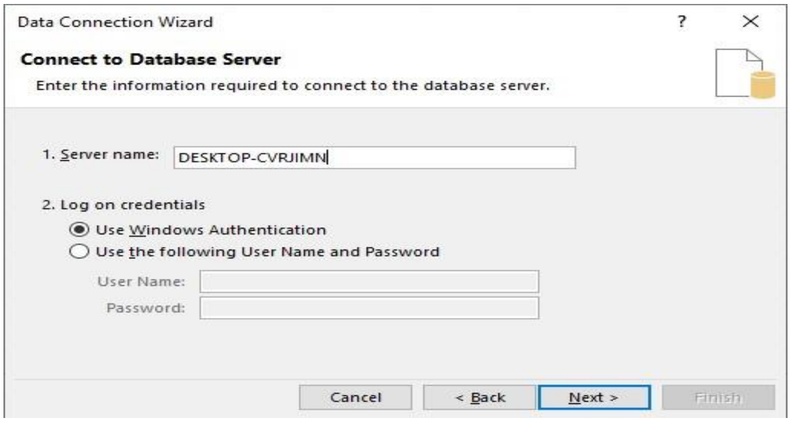
Welcome to the Data Connection Wizard

This wizard will help you connect to a remote data source.

What kind of data source do you want to connect to?

- Microsoft SQL Server (selected)
- Microsoft SQL Server Analysis Services
- Windows Azure Marketplace
- Data Feed
- ODBC DSN
- Microsoft Data Access - OLE DB Provider for Oracle
- Other/Advanced

Step 3: In connect to Database Server provide Server name-> Click on Next



Data Connection Wizard

Connect to Database Server

Enter the information required to connect to the database server.

1. Server name: DESKTOP-CVRJIMN

2. Log on credentials

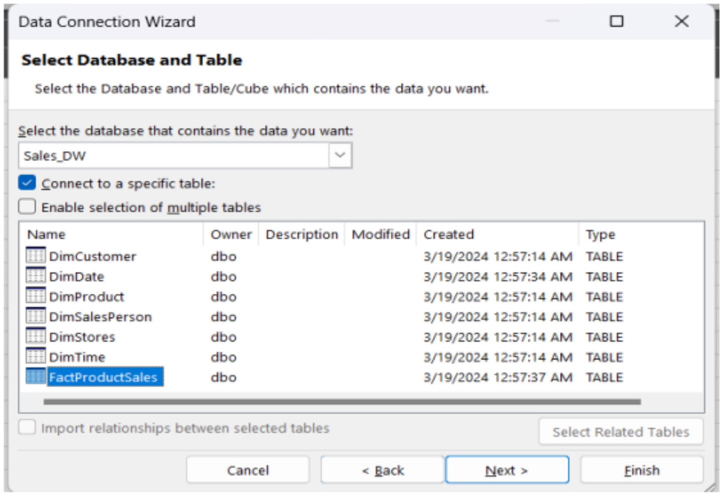
● Use Windows Authentication

○ Use the following User Name and Password

User Name:

Password:

Step 4: In Select Database and Table→ Select Adventure works (already created in SQL) → check all dimensions and import relationships between selected tables -> Click on next.



Data Connection Wizard

Select Database and Table

Select the Database and Table/Cube which contains the data you want.

Select the database that contains the data you want:

Sales_DW

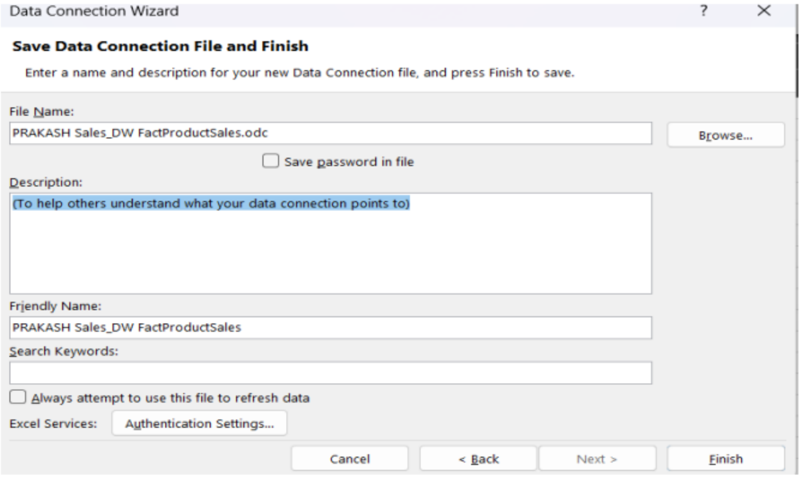
☑ Connect to a specific table:

☐ Enable selection of multiple tables

Name	Owner	Description	Modified	Created	Type
DimCustomer	dbo			3/19/2024 12:57:14 AM	TABLE
DimDate	dbo			3/19/2024 12:57:34 AM	TABLE
DimProduct	dbo			3/19/2024 12:57:14 AM	TABLE
DimSalesPerson	dbo			3/19/2024 12:57:14 AM	TABLE
DimStores	dbo			3/19/2024 12:57:14 AM	TABLE
DimTime	dbo			3/19/2024 12:57:14 AM	TABLE
FactProductSales	dbo			3/19/2024 12:57:37 AM	TABLE

☐ Import relationships between selected tables

Step 5: In save data connection files browse path and click on Finish



Data Connection Wizard

Save Data Connection File and Finish

Enter a name and description for your new Data Connection file, and press Finish to save.

File Name:

PRAKASH Sales_DW FactProductSales.odc

☐ Save password in file

Description:

(To help others understand what your data connection points to)

Friendly Name:

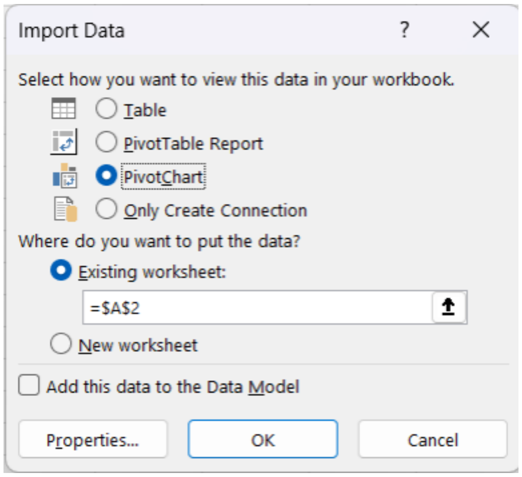
PRAKASH Sales_DW FactProductSales

Search Keywords:

☐ Always attempt to use this file to refresh data

Excel Services:

Step 6: In import data select Pivot Chart and click on OK



Import Data ? ✕

Select how you want to view this data in your workbook.

○ Table

○ PivotTable Report

● PivotChart

○ Only Create Connection

Where do you want to put the data?

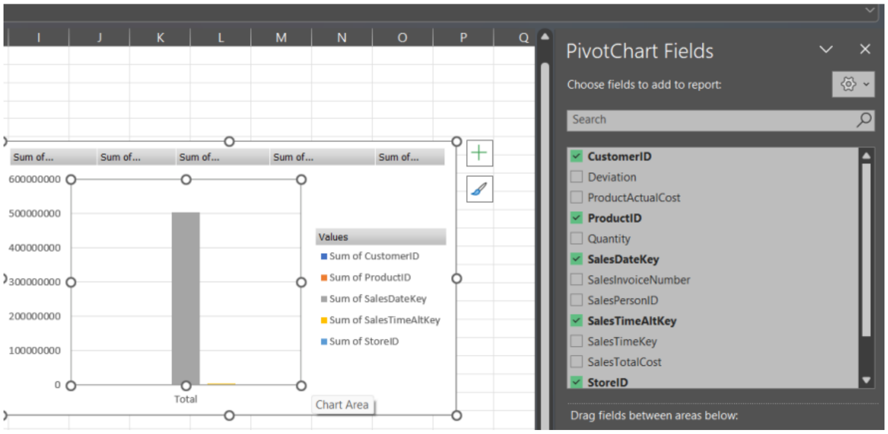
● Existing worksheet:

=$A$2

○ New worksheet

□ Add this data to the Data Model

Step 7: Select field to create Pivot Chart



PivotChart Fields

Choose fields to add to report:

Search

☑ CustomerID

□ Deviation

□ ProductActualCost

☑ ProductID

□ Quantity

☑ SalesDateKey

□ SalesInvoiceNumber

□ SalesPersonID

☑ SalesTimeAltKey

□ SalesTimeKey

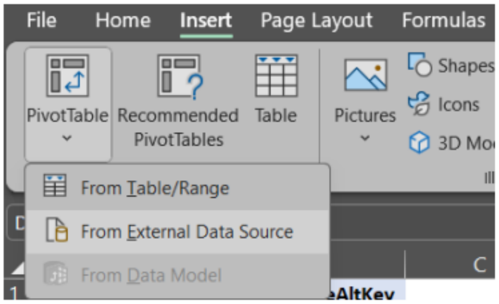
□ SalesTotalCost

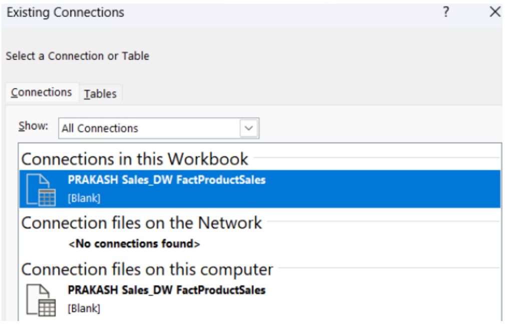
☑ StoreID

Drag fields between areas below:

Step 8: In Insert Tab -> Go to Pivot Table

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





Existing Connections ? ✕

Select a Connection or Table

Show: All Connections

Connections in this Workbook

PRAKASH Sales_DW FactProductSales
[Blank]

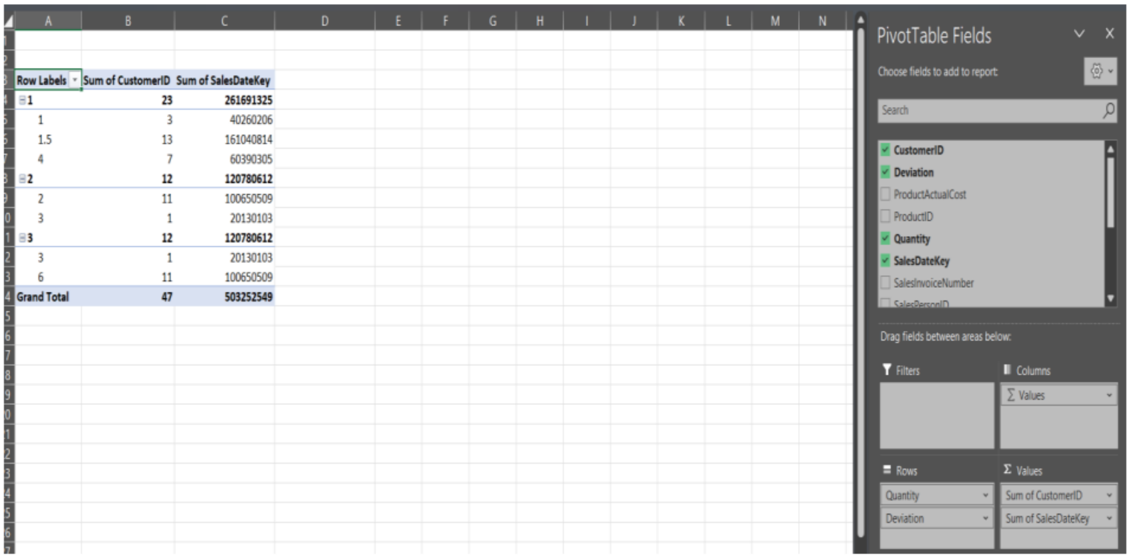
Connection files on the Network

<No connections found>

Connection files on this computer

PRAKASH Sales_DW FactProductSales
[Blank]

Step 10: Select Field to create Pivot Table. You will get this.



Row Labels	Sum of CustomerID	Sum of SalesDateKey
⊟1	**23**	**261691325**
1	3	40260206
1.5	13	161040814
4	7	60390305
⊟2	**12**	**120780612**
2	11	100650509
3	1	20130103
⊟3	**12**	**120780612**
3	1	20130103
6	11	100650509
Grand Total	**47**	**503252549**

PivotTable Fields

Choose fields to add to report:

Search

☑ **CustomerID**
☑ **Deviation**
□ ProductActualCost
□ ProductID
☑ **Quantity**
☑ **SalesDateKey**
□ SalesInvoiceNumber
□ SalesPersonID

Drag fields between areas below:

Filters

Columns: Σ Values

Rows: Quantity, Deviation

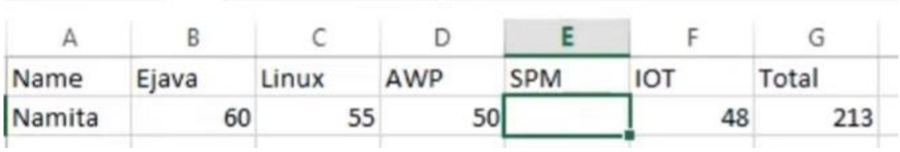
Σ Values: Sum of CustomerID, Sum of SalesDateKey

PRACTICAL NO: 6

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

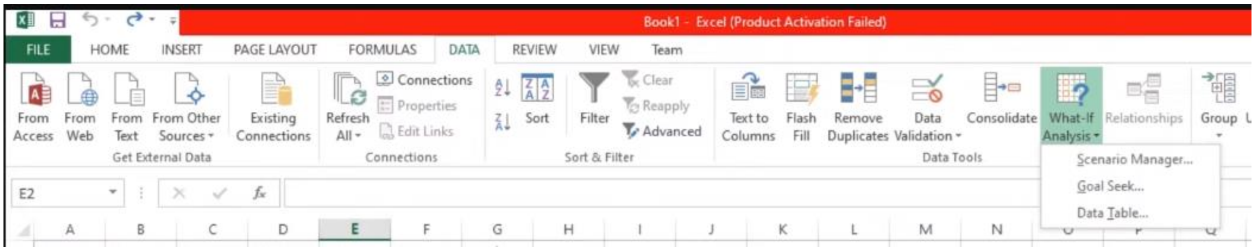
1) Goal Seek Method :

Step 1 : Go to Data Tab -> Select the What- if Analysis -> click on goal seek

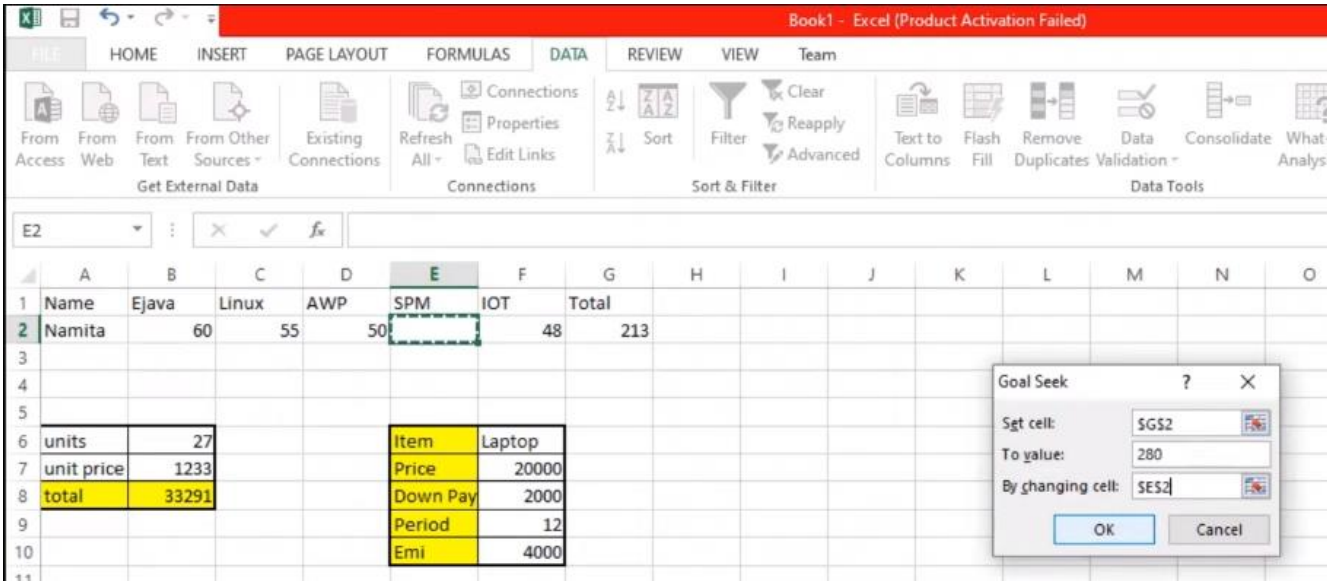


A	B	C	D	E	F	G
Name	Ejava	Linux	AWP	SPM	IOT	Total
Namita	60	55	50		48	213

Step 2: Go to Data tab -> What-if Analysis->Click on Goal Seek



Step 3: In “set cells” select the total column and in “By changing sell” select the missing value and in “to value” give any random number.



	A	B	C	D	E	F	G
1	Name	Ejava	Linux	AWP	SPM	IOT	Total
2	Namita	60	55	50		48	213
3							
4							
5							
6	units	27			Item	Laptop	
7	unit price	1233			Price	20000	
8	total	33291			Down Pay	2000	
9					Period	12	
10					Emi	4000	

Goal Seek

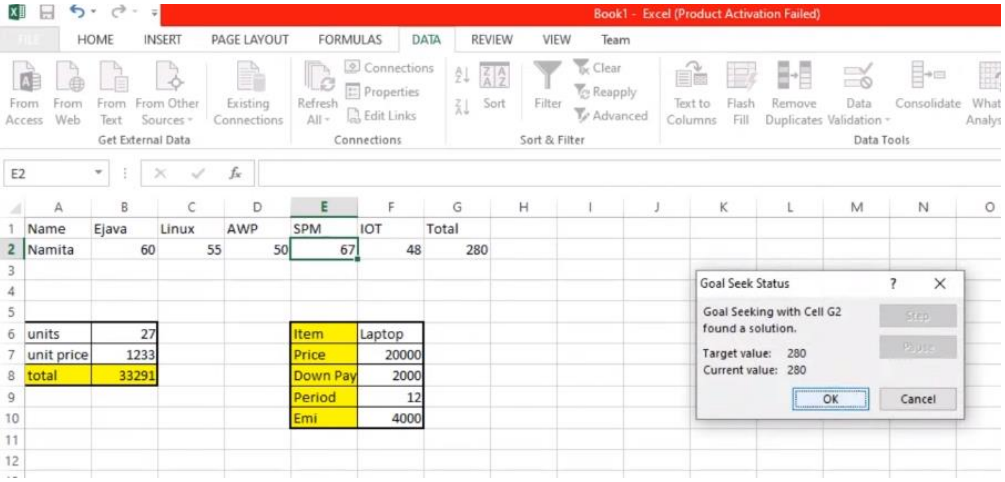
Set cell: $G$2

To value: 280

By changing cell: $E$2

OK Cancel

Step 4: The empty value is been filled according to the given condition.


Book1 - Excel (Product Activation Failed)
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW Team
From Access From Web From Text From Other Sources Existing Connections
Get External Data
Refresh All Connections Properties Edit Links
Connections
Sort Filter Clear Reapply Advanced
Sort & Filter
Text to Columns Flash Fill Remove Duplicates Data Validation Consolidate What-If Analysis
Data Tools
E2
Step
Pause
OK
Cancel

Name	Ejava	Linux	AWP	SPM	IOT	Total
Namita	60	55	50	67	48	280

units	27
unit price	1233
total	33291

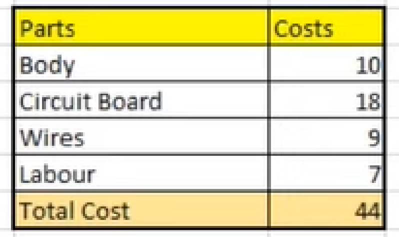
Item	Laptop
Price	20000
Down Pay	2000
Period	12
Emi	4000

Goal Seek Status

Goal Seeking with Cell G2 found a solution.

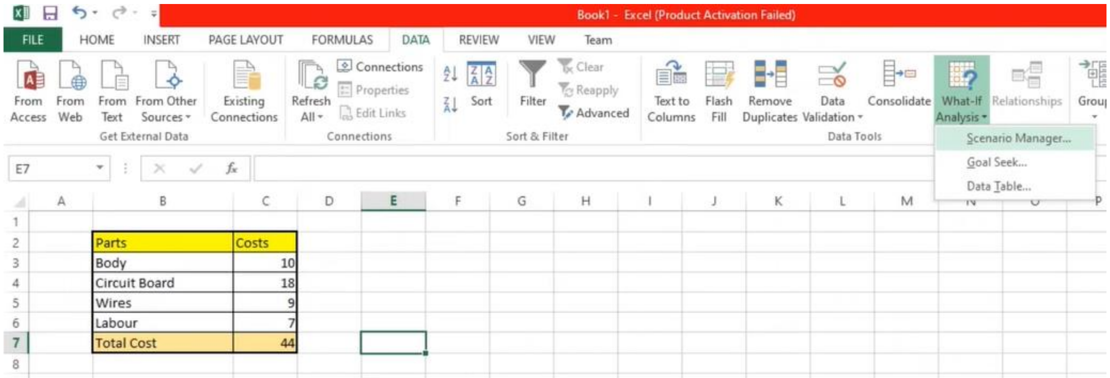
Target value: 280
Current value: 280

Step 5: Also create one more table.

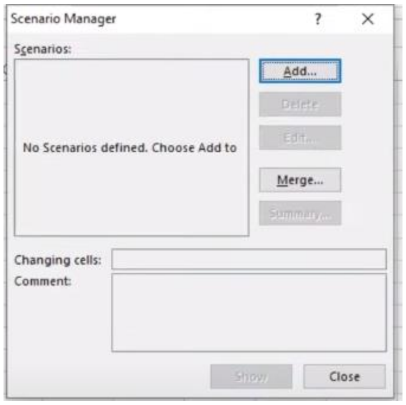


Parts	Costs
Body	10
Circuit Board	18
Wires	9
Labour	7
Total Cost	44

Step 6: Go to Data tab -> What-if Analysis -> Click on Scenario Manager -> Then Click on Add.


Book1 - Excel (Product Activation Failed)
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW Team
From Access From Web From Text From Other Sources Existing Connections
Get External Data
Refresh All Connections Properties Edit Links
Connections
Sort Filter Clear Reapply Advanced
Sort & Filter
Text to Columns Flash Fill Remove Duplicates Data Validation Consolidate
Data Tools
What-If Analysis
Relationships
Group
Scenario Manager...
Goal Seek...
Data Table...
E7

Parts	Costs
Body	10
Circuit Board	18
Wires	9
Labour	7
Total Cost	44



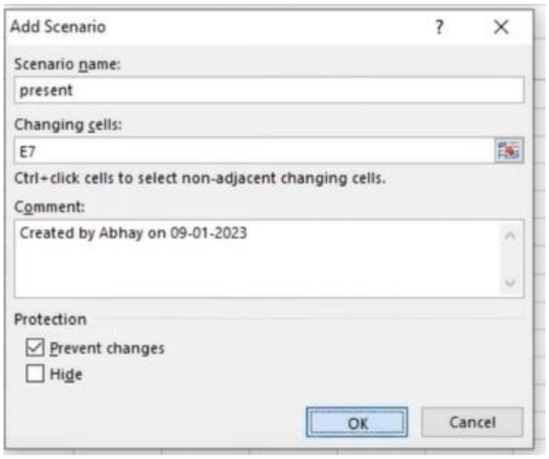
Scenario Manager

Scenarios:

No Scenarios defined. Choose Add to

Changing cells:

Comment:



Add Scenario

Scenario name: present

Changing cells: E7

Ctrl+click cells to select non-adjacent changing cells.

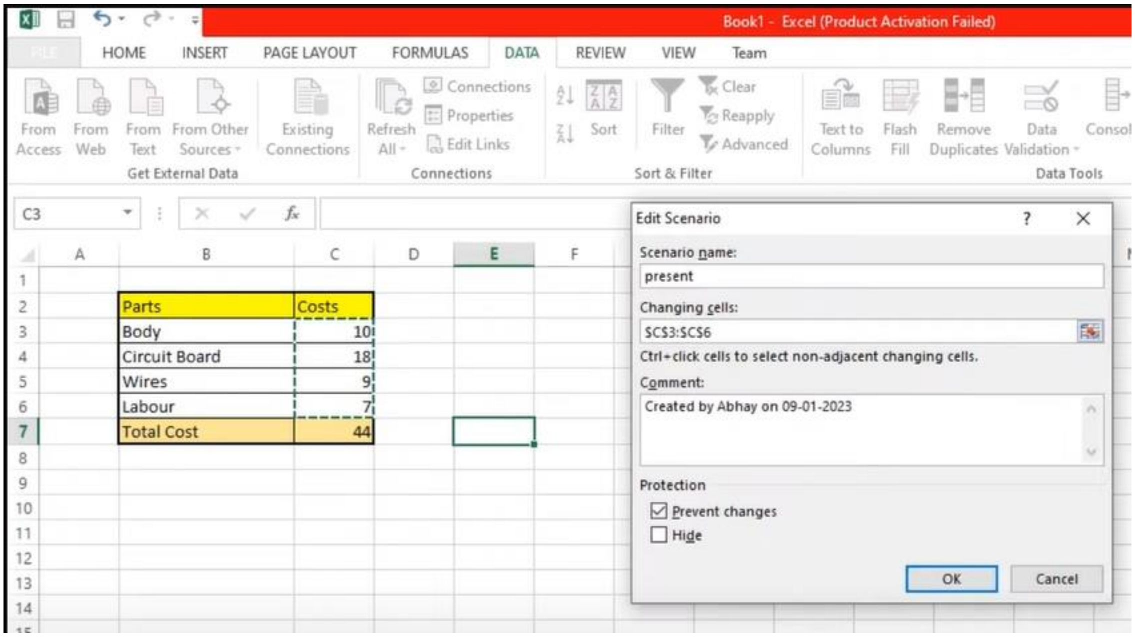
Comment: Created by Abhay on 09-01-2023

Protection

☑ Prevent changes

□ Hide

Step 7: Give the scenario name Present -> select the changing cells-> Click on ok



	A	B	C
1			
2		Parts	Costs
3		Body	10
4		Circuit Board	18
5		Wires	9
6		Labour	7
7		Total Cost	44

Edit Scenario

Scenario name: present

Changing cells: $C$3:$C$6

Ctrl+click cells to select non-adjacent changing cells.

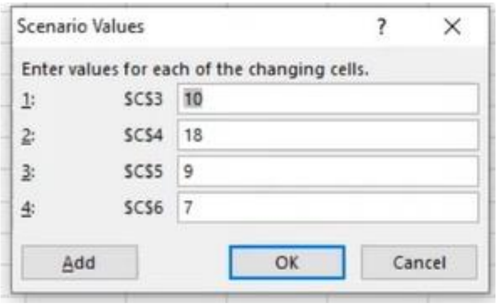
Comment: Created by Abhay on 09-01-2023

Protection

☑ Prevent changes

□ Hide

Step 8: Type Scenario present values -> Click on ok



Scenario Values

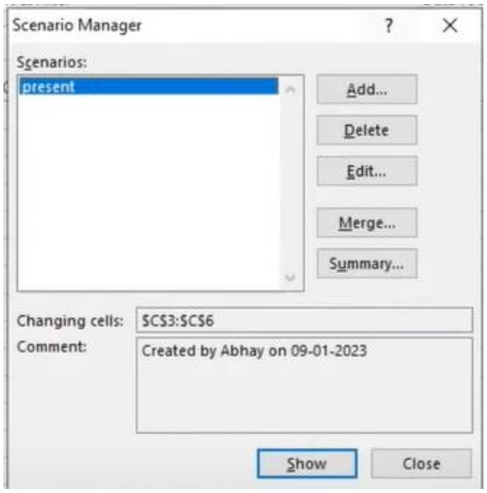
Enter values for each of the changing cells.

1: $C$3 10

2: $C$4 18

3: $C$5 9

4: $C$6 7

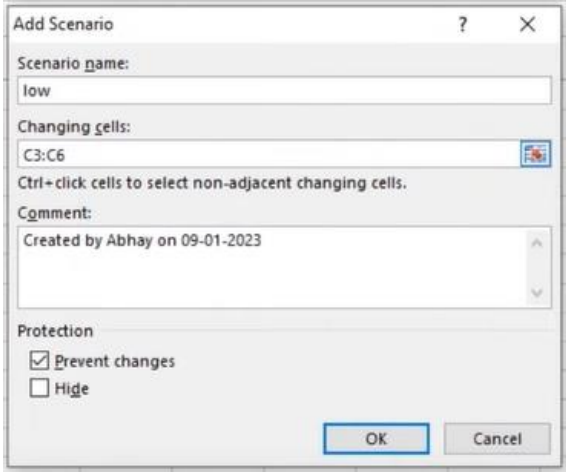


Scenarios: present

Changing cells: $C$3:$C$6

Comment: Created by Abhay on 09-01-2023

Step 9: Click on Add -> Give the scenario name Low -> select the changing cells -> Click on ok.



Scenario name: low

Changing cells: C3:C6

Ctrl+click cells to select non-adjacent changing cells.

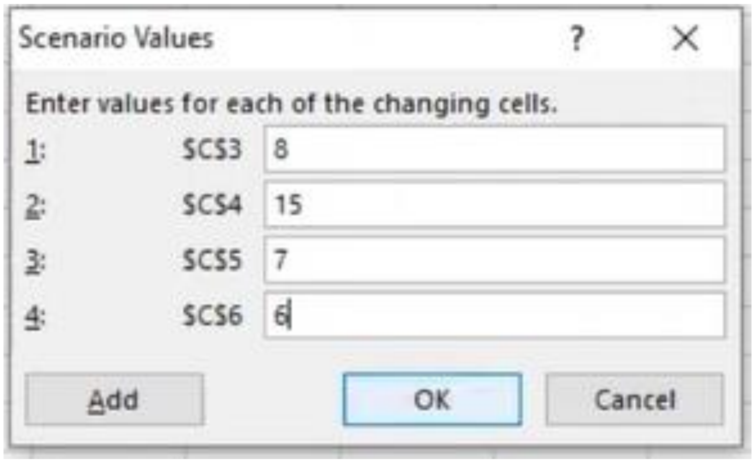
Comment: Created by Abhay on 09-01-2023

Protection

☑ Prevent changes

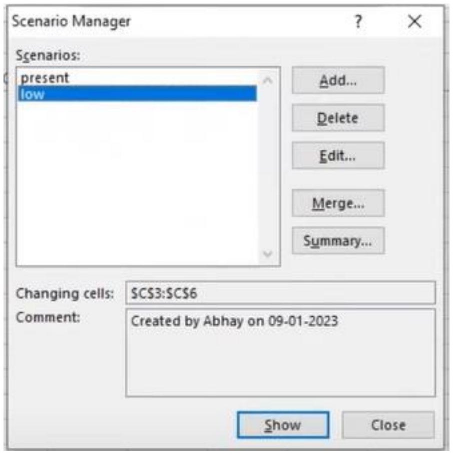
☐ Hide

Step 10: Type scenario Low values -> click ok



Enter values for each of the changing cells.

1:	$C$3	8
2:	$C$4	15
3:	$C$5	7
4:	$C$6	6

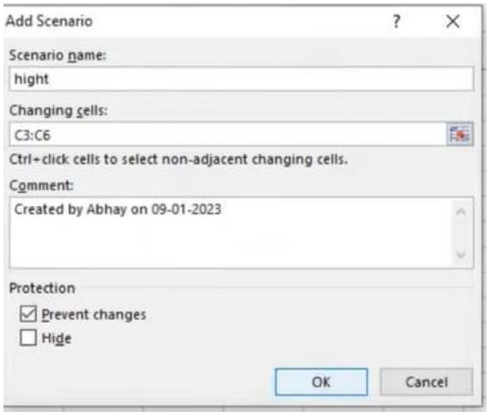

Scenario Manager

Scenarios:
present
low

Changing cells: $C$3:$C$6

Comment: Created by Abhay on 09-01-2023

Step 11: Click on Add -> Give the scenario name High ->Select the changing cells -> Click on ok.


Add Scenario

Scenario name: hight

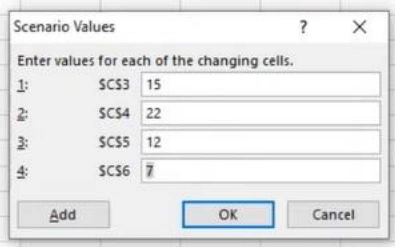
Changing cells: C3:C6

Ctrl+click cells to select non-adjacent changing cells.

Comment: Created by Abhay on 09-01-2023

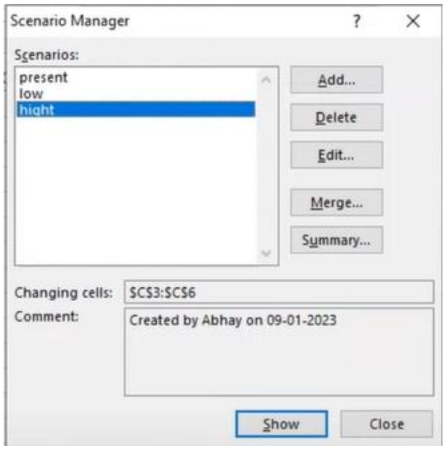
Protection

☑ Prevent changes
☐ Hide


Scenario Values

Enter values for each of the changing cells.

1:	$C$3	15
2:	$C$4	22
3:	$C$5	12
4:	$C$6	7

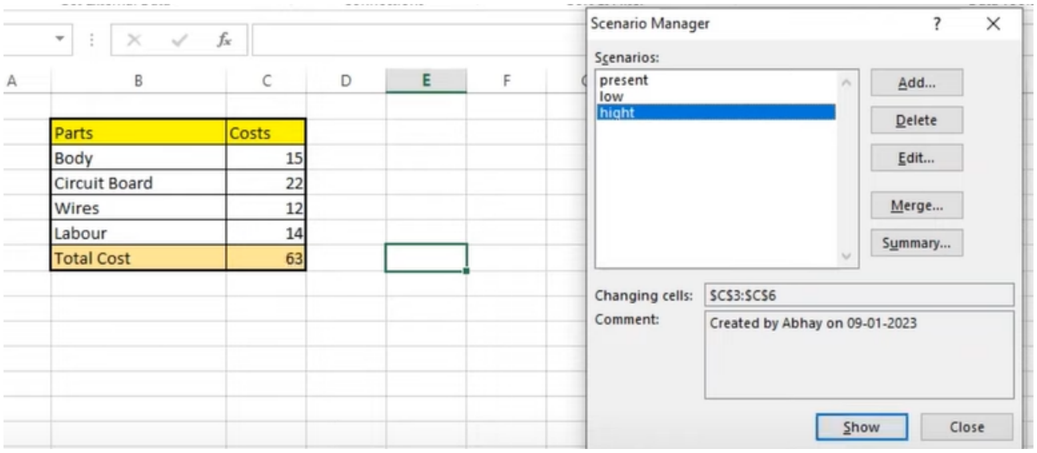

Scenario Manager

Scenarios:
present
low
hight

Changing cells: $C$3:$C$6

Comment: Created by Abhay on 09-01-2023

Step 13: Click on Show



Parts	Costs
Body	15
Circuit Board	22
Wires	12
Labour	14
Total Cost	63

Scenario Manager

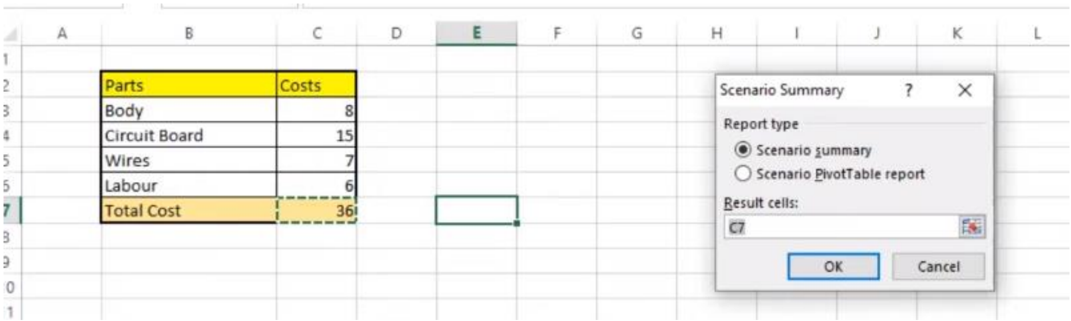
Scenarios:

present
low
hight

Changing cells: $C$3:$C$6

Comment: Created by Abhay on 09-01-2023

Step 14: Click on Summary -> select total cost cell in result cells -> Click on ok.



Parts	Costs
Body	8
Circuit Board	15
Wires	7
Labour	6
Total Cost	36

Scenario Summary

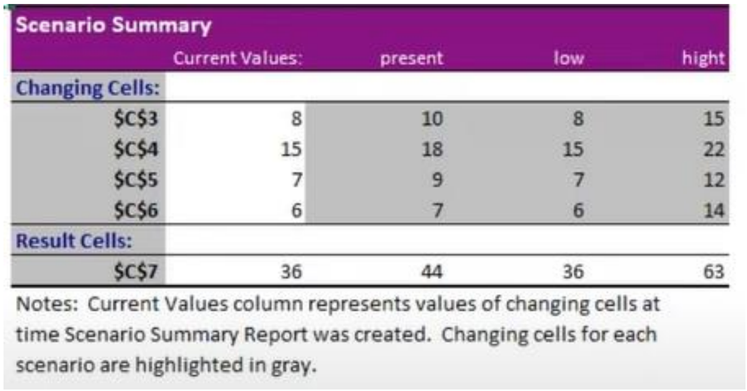
Report type

● Scenario summary

○ Scenario PivotTable report

Result cells: C7

Step 15: You will get Scenario Summary



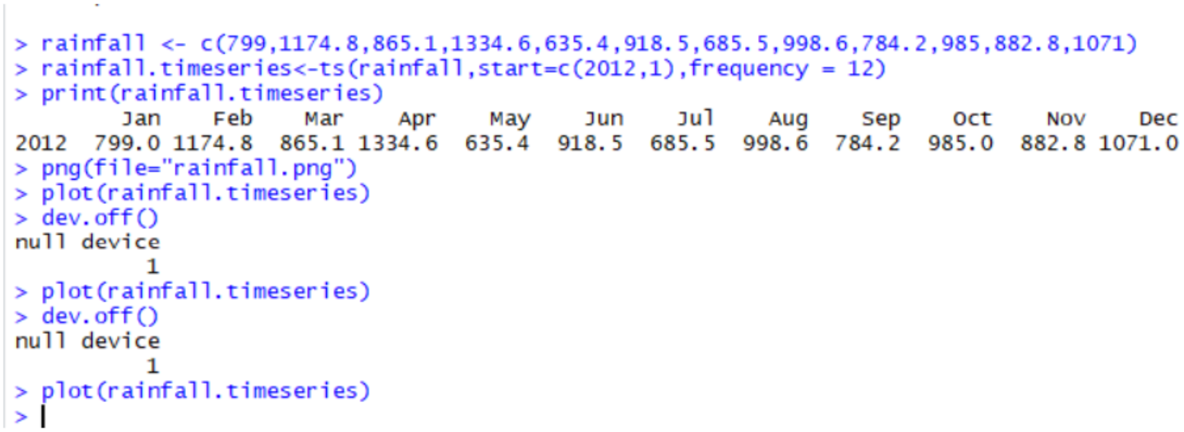
Scenario Summary				
	Current Values:	present	low	hight
Changing Cells:				
$C$3	8	10	8	15
$C$4	15	18	15	22
$C$5	7	9	7	12
$C$6	6	7	6	14
Result Cells:				
$C$7	36	44	36	63

Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.

PRACTICAL NO: 7

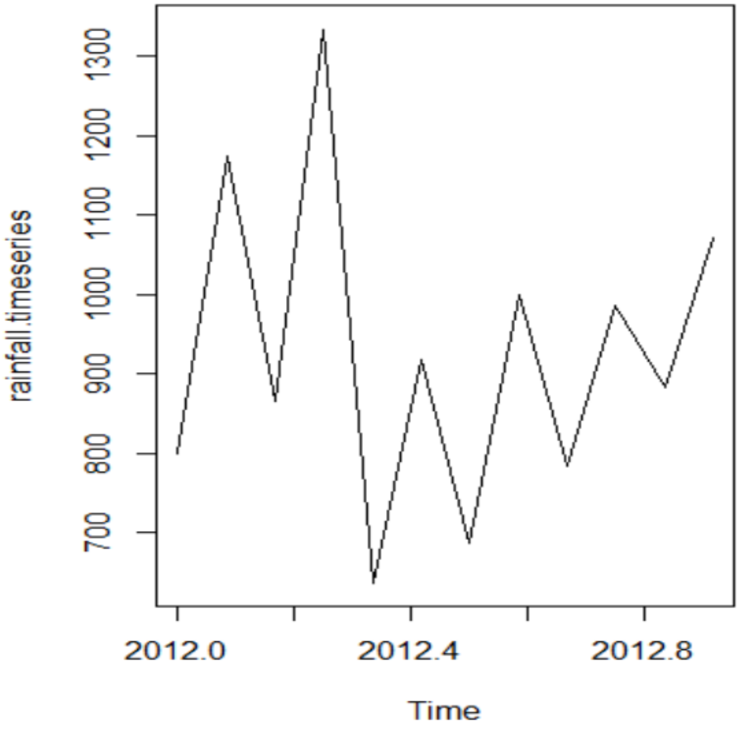
AIM : Perform the data classification using classification algorithm.

Step 1: write this code in Rstudio ide



```
> rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
> rainfall.timeseries<-ts(rainfall,start=c(2012,1),frequency = 12)
> print(rainfall.timeseries)
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
2012  799.0 1174.8  865.1 1334.6  635.4  918.5  685.5  998.6  784.2  985.0  882.8 1071.0
> png(file="rainfall.png")
> plot(rainfall.timeseries)
> dev.off()
null device
          1
> plot(rainfall.timeseries)
> dev.off()
null device
          1
> plot(rainfall.timeseries)
> |
```

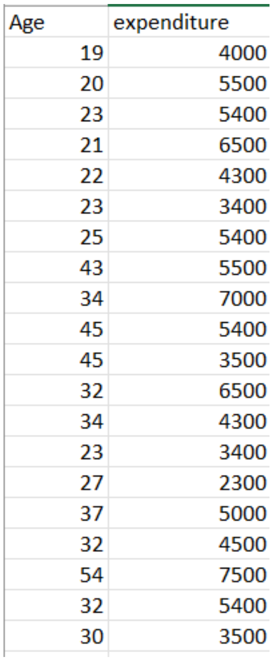
Final Output



PRACTICAL No: 8

Aim: Perform the data clustering using a clustering algorithm.

Step 1: Create Excel file with Age and Expenditure column -> save as CSV.

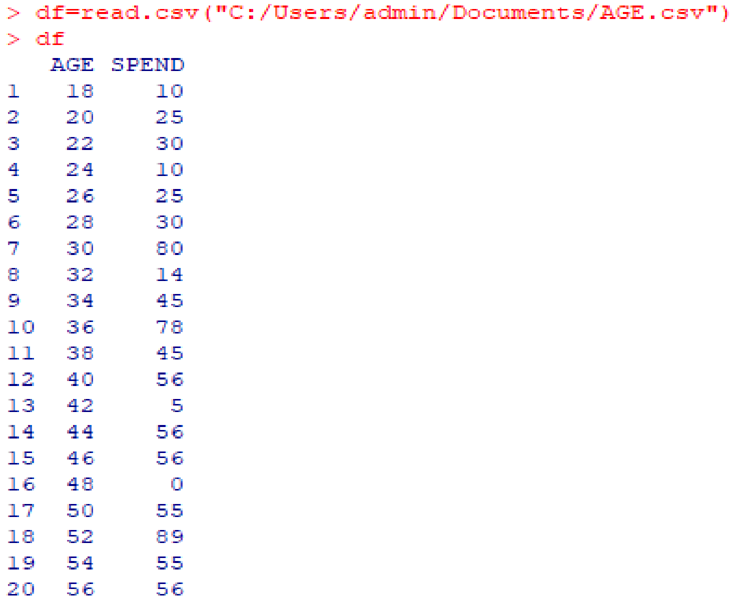


Age	expenditure
19	4000
20	5500
23	5400
21	6500
22	4300
23	3400
25	5400
43	5500
34	7000
45	5400
45	3500
32	6500
34	4300
23	3400
27	2300
37	5000
32	4500
54	7500
32	5400
30	3500

Step 2: Open R Studio -> Go to R console and type code.

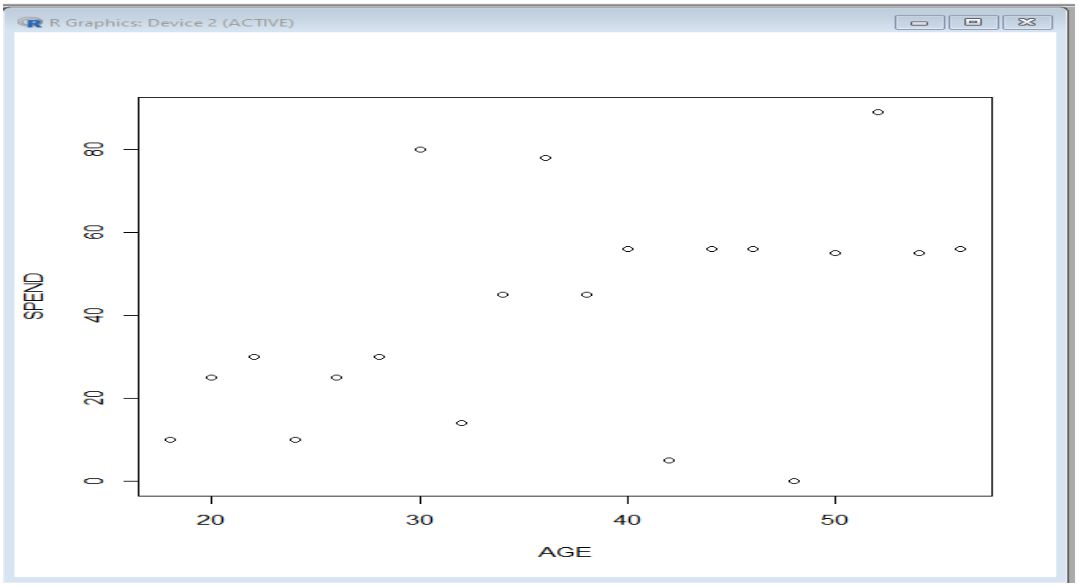
1.df=read.csv("C:/Users/admin/Documents/AGE.csv")

df

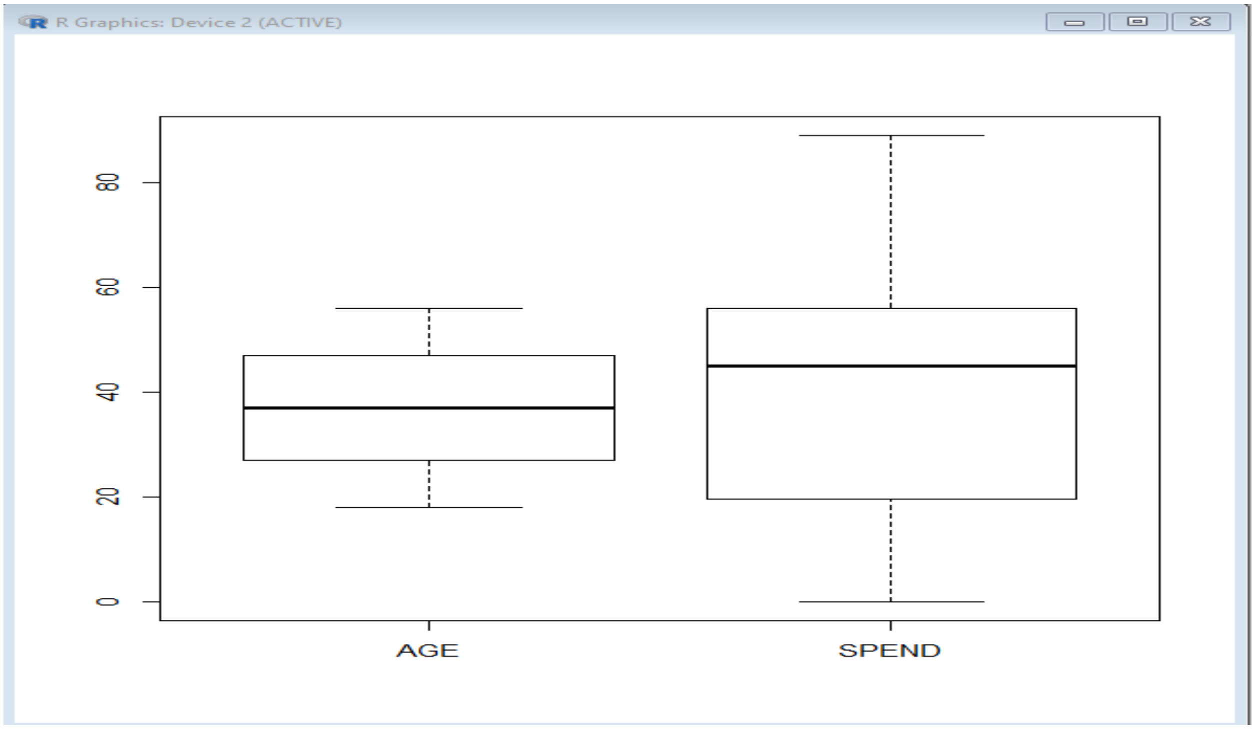


```
> df=read.csv("C:/Users/admin/Documents/AGE.csv")
> df
   AGE SPEND
1   18    10
2   20    25
3   22    30
4   24    10
5   26    25
6   28    30
7   30    80
8   32    14
9   34    45
10  36    78
11  38    45
12  40    56
13  42     5
14  44    56
15  46    56
16  48     0
17  50    55
18  52    89
19  54    55
20  56    56
```

4.plot(df)



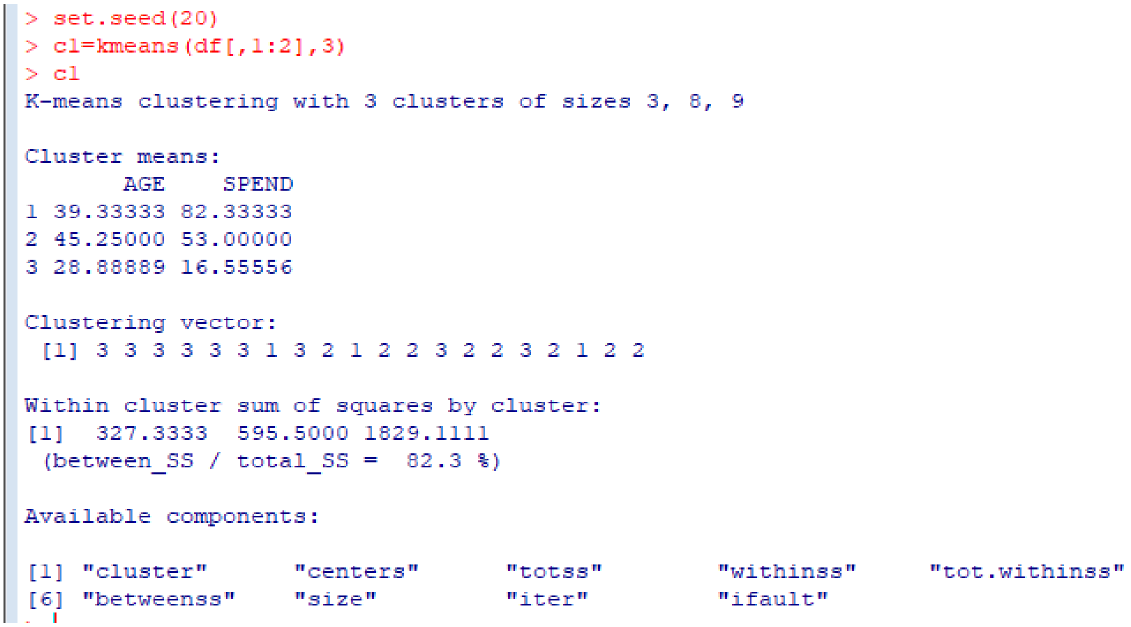
5.boxplot(df)



6.set.seed(20)

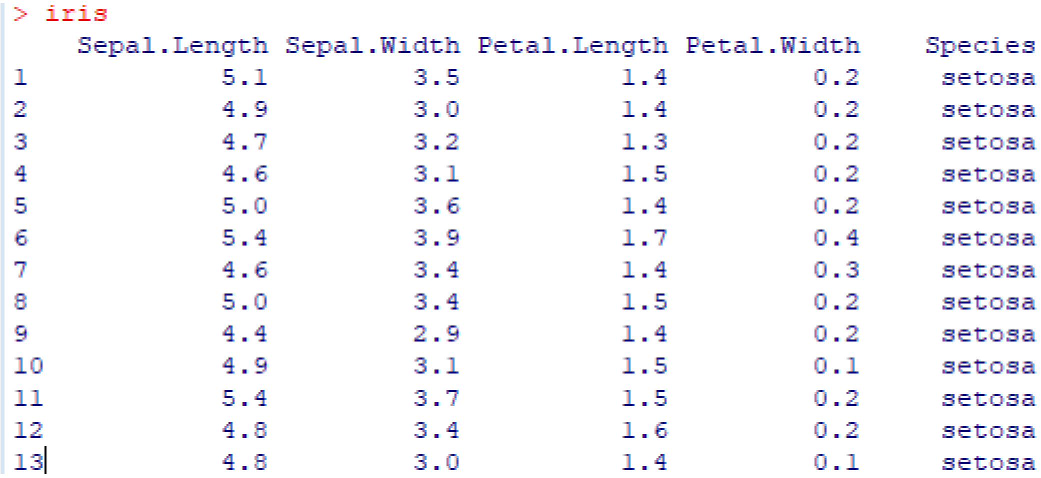
> c1=kmeans(df[,1:2],3)

> c1



```
> set.seed(20)
> c1=kmeans(df[,1:2],3)
> c1
K-means clustering with 3 clusters of sizes 3, 8, 9

Cluster means:
       AGE    SPEND
1 39.33333 82.33333
2 45.25000 53.00000
3 28.88889 16.55556

Clustering vector:
 [1] 3 3 3 3 3 3 1 3 2 1 2 2 3 2 2 3 2 1 2 2

Within cluster sum of squares by cluster:
[1]  327.3333  595.5000 1829.1111
 (between_SS / total_SS =  82.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

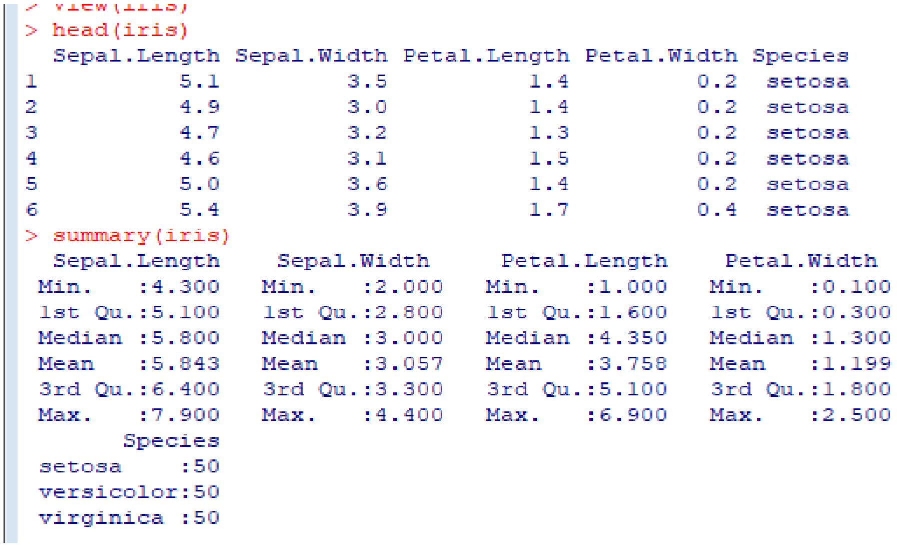
7.iris



```
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
11           5.4         3.7          1.5         0.2     setosa
12           4.8         3.4          1.6         0.2     setosa
13           4.8         3.0          1.4         0.1     setosa
```

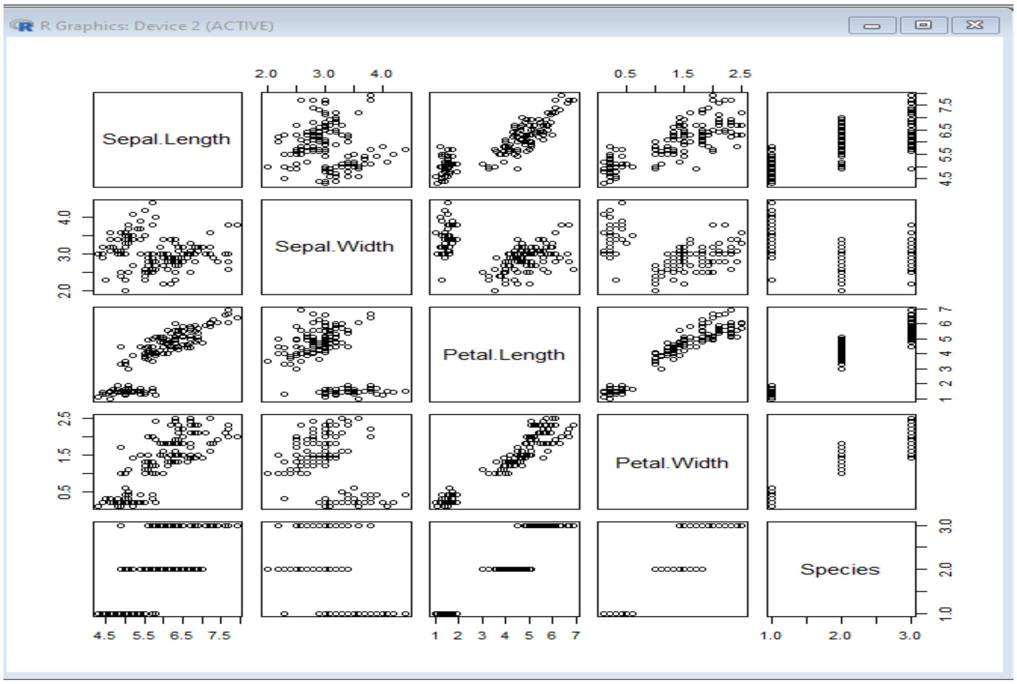
8.head(iris)

9.summary(iris)

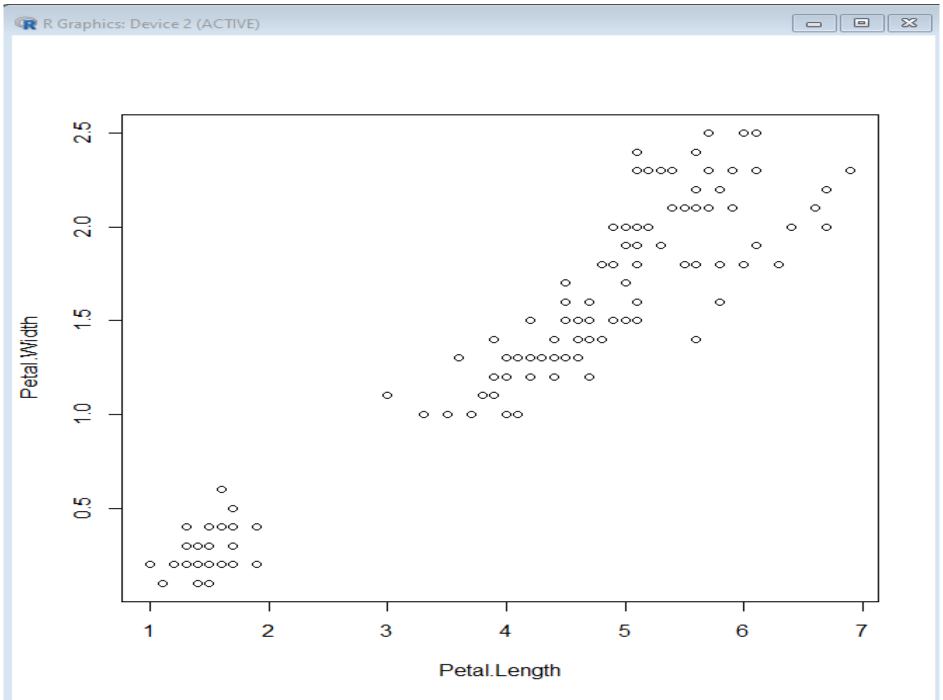


```
> View(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

10.plot(iris)

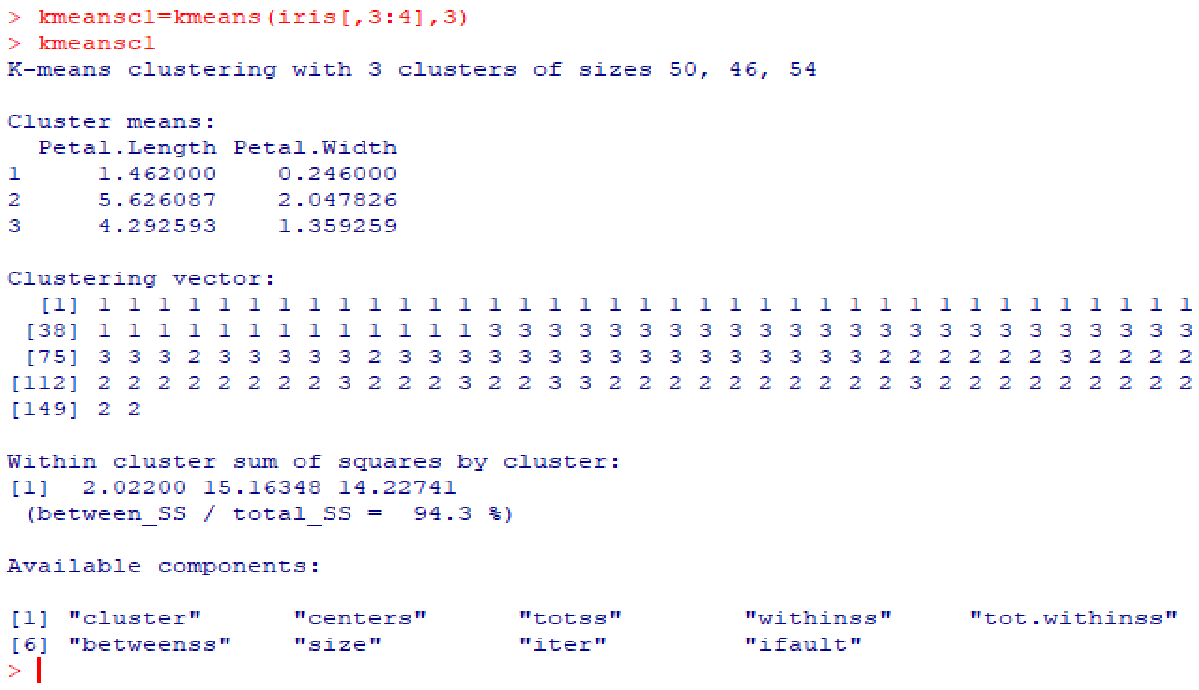


11.plot(iris[,3:4])



12.kmeansc1=kmeans(iris[,3:4],3)

kmeansc1

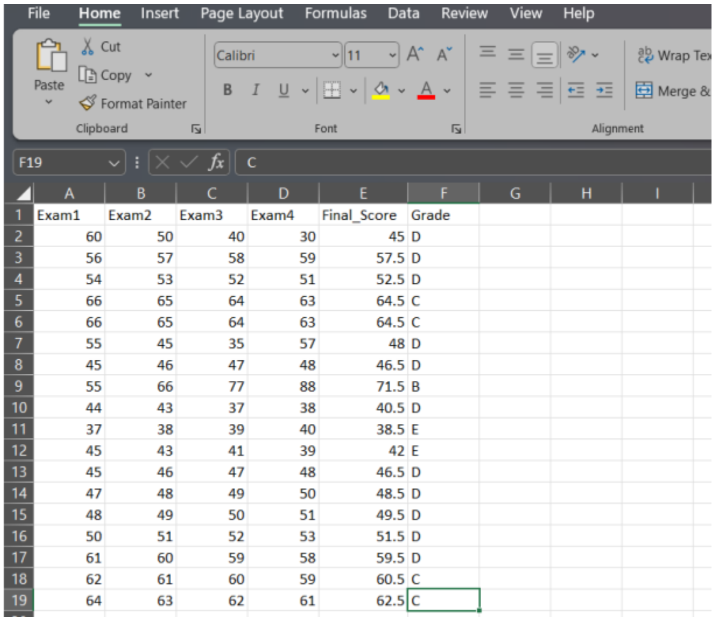


```
> kmeansc1=kmeans(iris[,3:4],3)
> kmeansc1
K-means clustering with 3 clusters of sizes 50, 46, 54

Cluster means:
  Petal.Length Petal.Width
1     1.462000    0.246000
2     5.626087    2.047826
3     4.292593    1.359259

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [75] 3 3 3 2 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 3 2 2 2 2
[112] 2 2 2 2 2 2 2 2 3 2 2 2 3 2 2 3 3 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2
[149] 2 2

Within cluster sum of squares by cluster:
[1]  2.02200 15.16348 14.22741
 (between_SS / total_SS =  94.3 %)

Available components:

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

PRACTICAL NO: 9

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

Step 1: Open Excel-> create 4 columns with names Exam1, Exam2, Exam3, Exam4, Final Score, Grade.

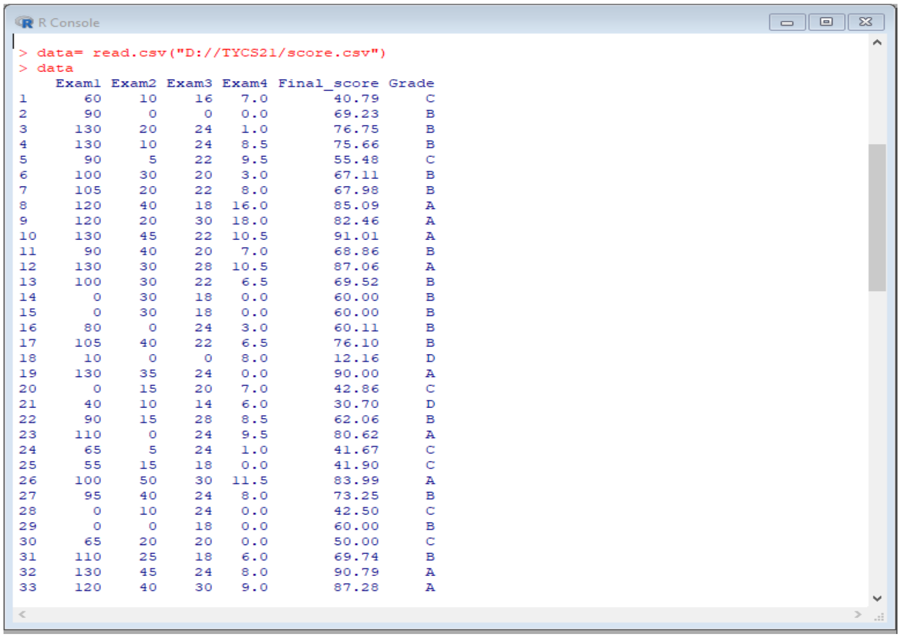


	A	B	C	D	E	F
1	Exam1	Exam2	Exam3	Exam4	Final_Score	Grade
2	60	50	40	30	45	D
3	56	57	58	59	57.5	D
4	54	53	52	51	52.5	D
5	66	65	64	63	64.5	C
6	66	65	64	63	64.5	C
7	55	45	35	57	48	D
8	45	46	47	48	46.5	D
9	55	66	77	88	71.5	B
10	44	43	37	38	40.5	D
11	37	38	39	40	38.5	E
12	45	43	41	39	42	E
13	45	46	47	48	46.5	D
14	47	48	49	50	48.5	D
15	48	49	50	51	49.5	D
16	50	51	52	53	51.5	D
17	61	60	59	58	59.5	D
18	62	61	60	59	60.5	C
19	64	63	62	61	62.5	C

Step 2: Open R Studio -> Go on R console and type code

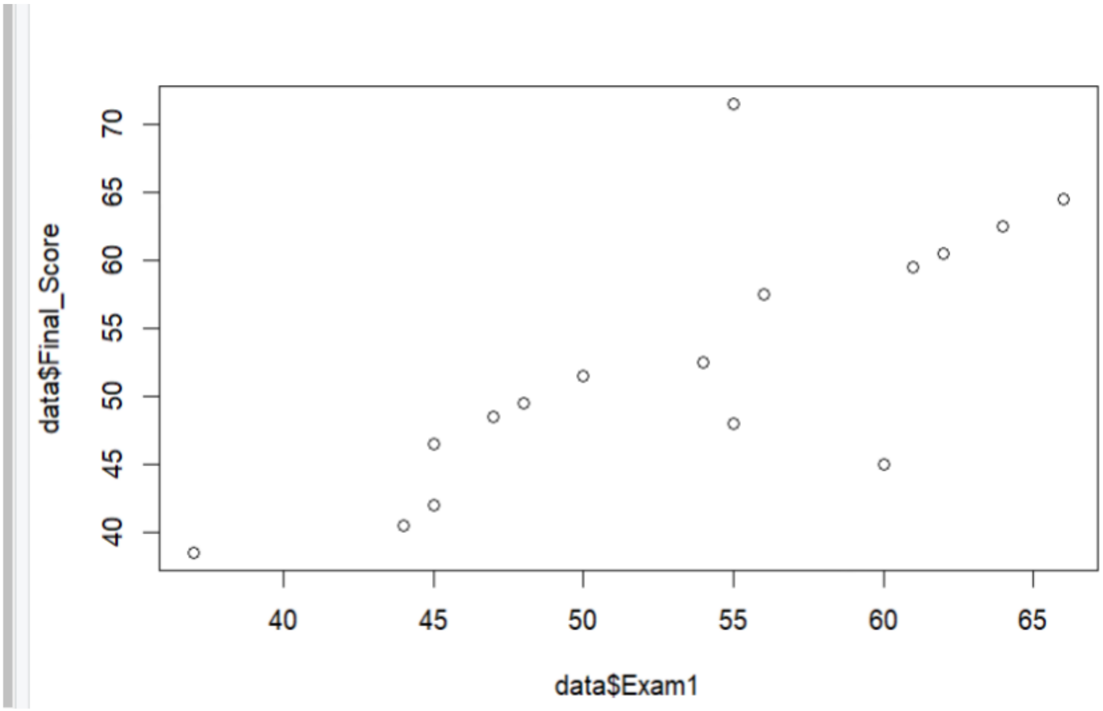
1.Data = read.csv(“D://TYCS21/score.csv”)

Data

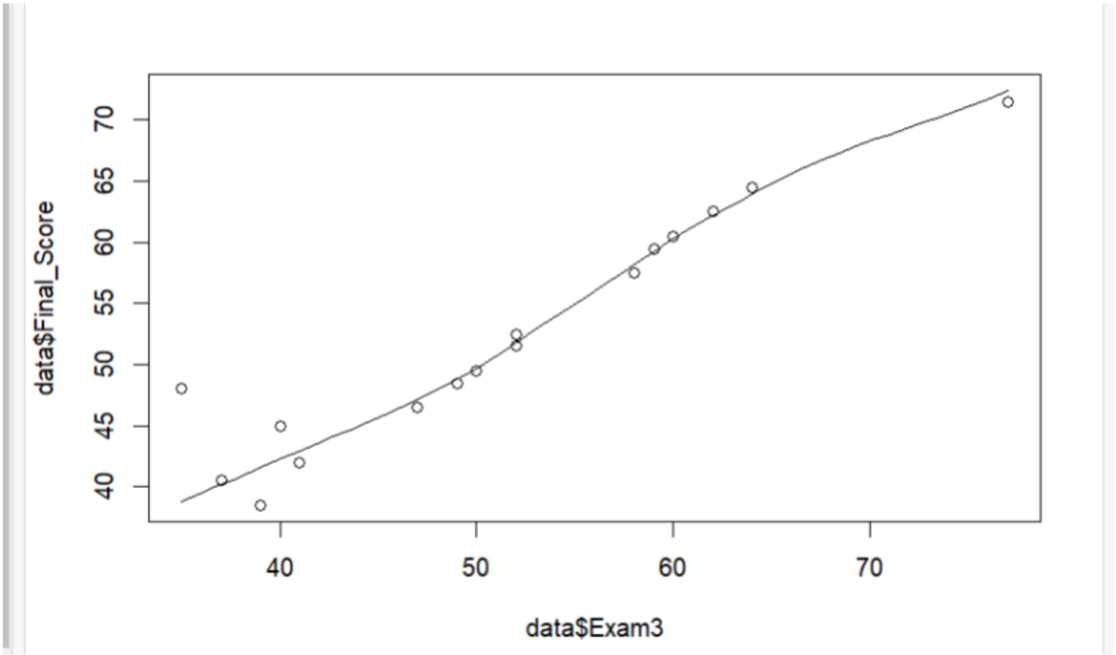


```
> data= read.csv("D://TYCS21/score.csv")
> data
   Exam1 Exam2 Exam3 Exam4 Final_score Grade
1     60    10    16   7.0       40.79     C
2     90     0     0   0.0       69.23     B
3    130    20    24   1.0       76.75     B
4    130    10    24   8.5       75.66     B
5     90     5    22   9.5       55.48     C
6    100    30    20   3.0       67.11     B
7    105    20    22   8.0       67.98     B
8    120    40    18  16.0       85.09     A
9    120    20    30  18.0       82.46     A
10   130    45    22  10.5       91.01     A
11    90    40    20   7.0       68.86     B
12   130    30    28  10.5       87.06     A
13   100    30    22   6.5       69.52     B
14     0    30    18   0.0       60.00     B
15     0    30    18   0.0       60.00     B
16    80     0    24   3.0       60.11     B
17   105    40    22   6.5       76.10     B
18    10     0     0   8.0       12.16     D
19   130    35    24   0.0       90.00     A
20     0    15    20   7.0       42.86     C
21    40    10    14   6.0       30.70     D
22    90    15    28   8.5       62.06     B
23   110     0    24   9.5       80.62     A
24    65     5    24   1.0       41.67     C
25    55    15    18   0.0       41.90     C
26   100    50    30  11.5       83.99     A
27    95    40    24   8.0       73.25     B
28     0    10    24   0.0       42.50     C
29     0     0    18   0.0       60.00     B
30    65    20    20   0.0       50.00     C
31   110    25    18   6.0       69.74     B
32   130    45    24   8.0       90.79     A
33   120    40    30   9.0       87.28     A
```

2.plot(x=data$Exam1,y=data$Final_score)



3.scatter.smooth(x=data$Exam3,y=data$Final_score)

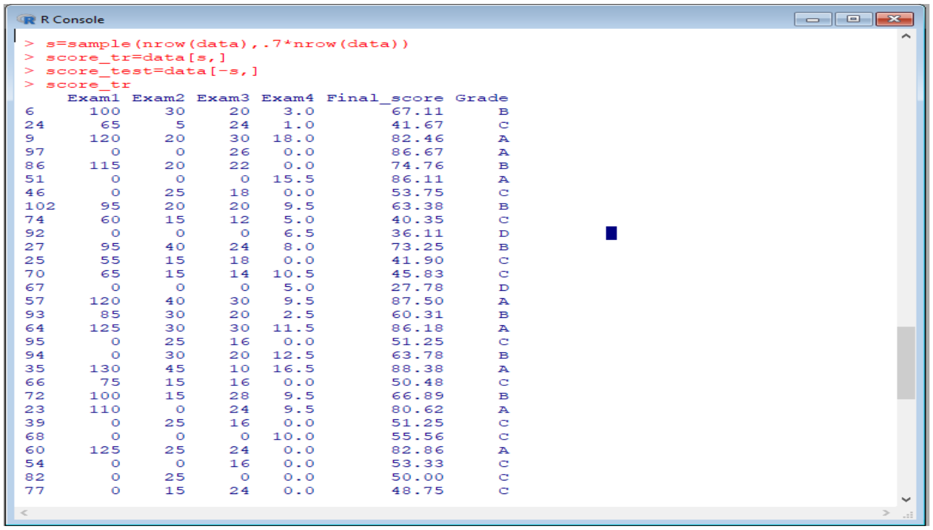


4.s=sample(nrow(data),.7*nrow(data))

>score_tr=data[s,]

>score_test=[-s,]

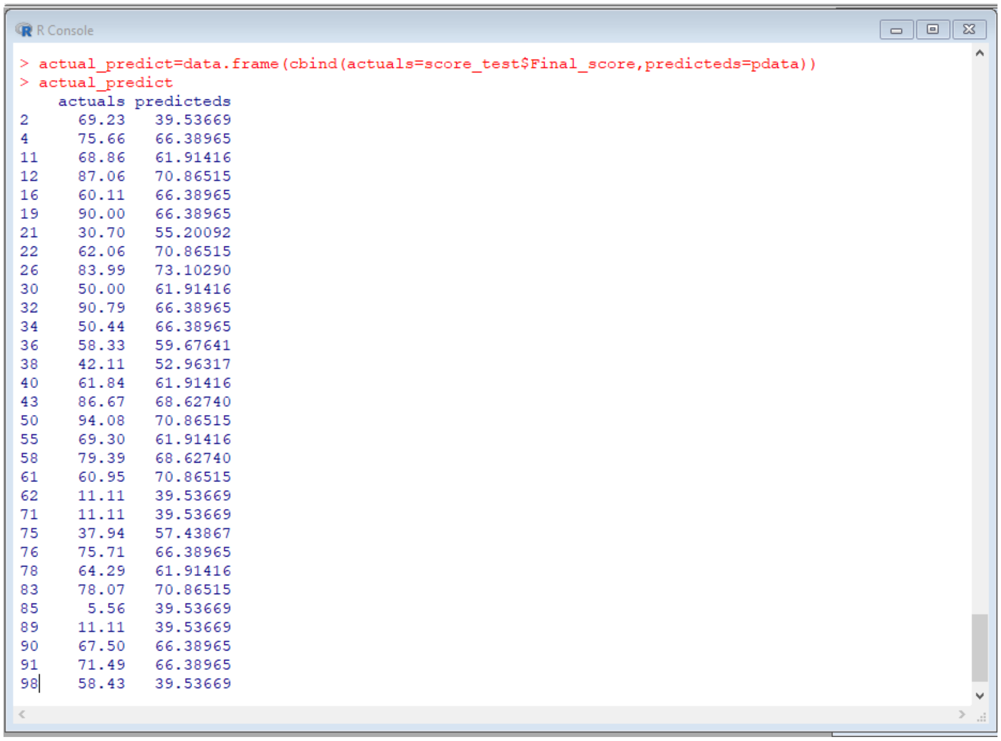
Score_tr



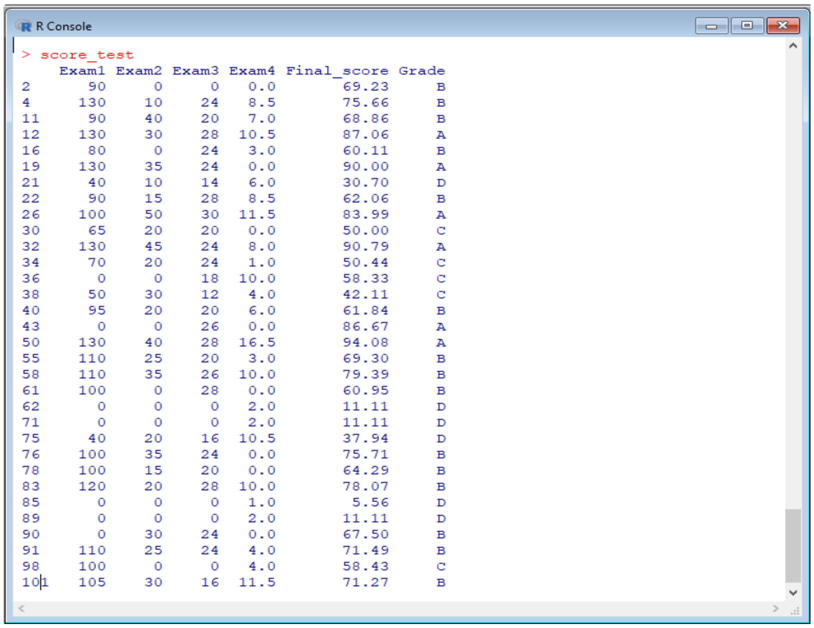
```
> s=sample(nrow(data),.7*nrow(data))
> score_tr=data[s,]
> score_test=data[-s,]
> score_tr
    Exam1 Exam2 Exam3 Exam4 Final_score Grade
6     100    30    20   3.0       67.11     B
24     65     5    24   1.0       41.67     C
9     120    20    30  18.0       82.46     A
97      0     0    26   0.0       86.67     A
86    115    20    22   0.0       74.76     B
51      0     0     0  15.5       86.11     A
46      0    25    18   0.0       53.75     C
102    95    20    20   9.5       63.38     B
74     60    15    12   5.0       40.35     C
92      0     0     0   6.5       36.11     D
27     95    40    24   8.0       73.25     B
25     55    15    18   0.0       41.90     C
70     65    15    14  10.5       45.83     C
67      0     0     0   5.0       27.78     D
57    120    40    30   9.5       87.50     A
93     85    30    20   2.5       60.31     B
64    125    30    30  11.5       86.18     A
95      0    25    16   0.0       51.25     C
94      0    30    20  12.5       63.78     B
35    130    45    10  16.5       88.38     A
66     75    15    16   0.0       50.48     C
72    100    15    28   9.5       66.89     B
23    110     0    24   9.5       80.62     A
39      0    25    16   0.0       51.25     C
68      0     0     0  10.0       55.56     C
60    125    25    24   0.0       82.86     A
54      0     0    16   0.0       53.33     C
82      0    25     0   0.0       50.00     C
77      0    15    24   0.0       48.75     C
```

5. linmod =lm(Final_score-Exam3,data=score_tr)

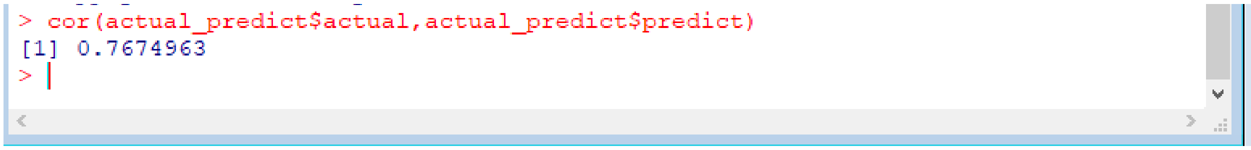
print(linmod)



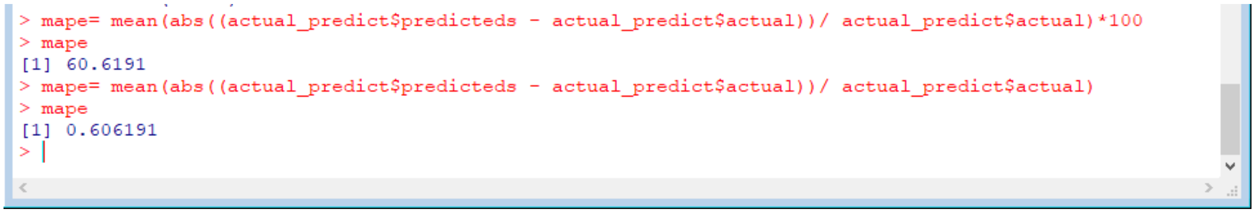
```
> actual_predict=data.frame(cbind(actuals=score_test$Final_score,predicteds=pdata))
> actual_predict
   actuals predicteds
2    69.23   39.53669
4    75.66   66.38965
11   68.86   61.91416
12   87.06   70.86515
16   60.11   66.38965
19   90.00   66.38965
21   30.70   55.20092
22   62.06   70.86515
26   83.99   73.10290
30   50.00   61.91416
32   90.79   66.38965
34   50.44   66.38965
36   58.33   59.67641
38   42.11   52.96317
40   61.84   61.91416
43   86.67   68.62740
50   94.08   70.86515
55   69.30   61.91416
58   79.39   68.62740
61   60.95   70.86515
62   11.11   39.53669
71   11.11   39.53669
75   37.94   57.43867
76   75.71   66.38965
78   64.29   61.91416
83   78.07   70.86515
85    5.56   39.53669
89   11.11   39.53669
90   67.50   66.38965
91   71.49   66.38965
98   58.43   39.53669
```



```
> score_test
    Exam1 Exam2 Exam3 Exam4 Final_score Grade
2      90     0     0   0.0       69.23     B
4     130    10    24   8.5       75.66     B
11     90    40    20   7.0       68.86     B
12    130    30    28  10.5       87.06     A
16     80     0    24   3.0       60.11     B
19    130    35    24   0.0       90.00     A
21     40    10    14   6.0       30.70     D
22     90    15    28   8.5       62.06     B
26    100    50    30  11.5       83.99     A
30     65    20    20   0.0       50.00     C
32    130    45    24   8.0       90.79     A
34     70    20    24   1.0       50.44     C
36      0     0    18  10.0       58.33     C
38     50    30    12   4.0       42.11     C
40     95    20    20   6.0       61.84     B
43      0     0    26   0.0       86.67     A
50    130    40    28  16.5       94.08     A
55    110    25    20   3.0       69.30     B
58    110    35    26  10.0       79.39     B
61    100     0    28   0.0       60.95     B
62      0     0     0   2.0       11.11     D
71      0     0     0   2.0       11.11     D
75     40    20    16  10.5       37.94     D
76    100    35    24   0.0       75.71     B
78    100    15    20   0.0       64.29     B
83    120    20    28  10.0       78.07     B
85      0     0     0   1.0        5.56     D
89      0     0     0   2.0       11.11     D
90      0    30    24   0.0       67.50     B
91    110    25    24   4.0       71.49     B
98    100     0     0   4.0       58.43     C
101   105    30    16  11.5       71.27     B
```



```
> cor(actual_predict$actual,actual_predict$predict)
[1] 0.7674963
> 
```



```
> mape= mean(abs((actual_predict$predicteds - actual_predict$actual))/ actual_predict$actual)*100
> mape
[1] 60.6191
> mape= mean(abs((actual_predict$predicteds - actual_predict$actual))/ actual_predict$actual)
> mape
[1] 0.606191
> 
```

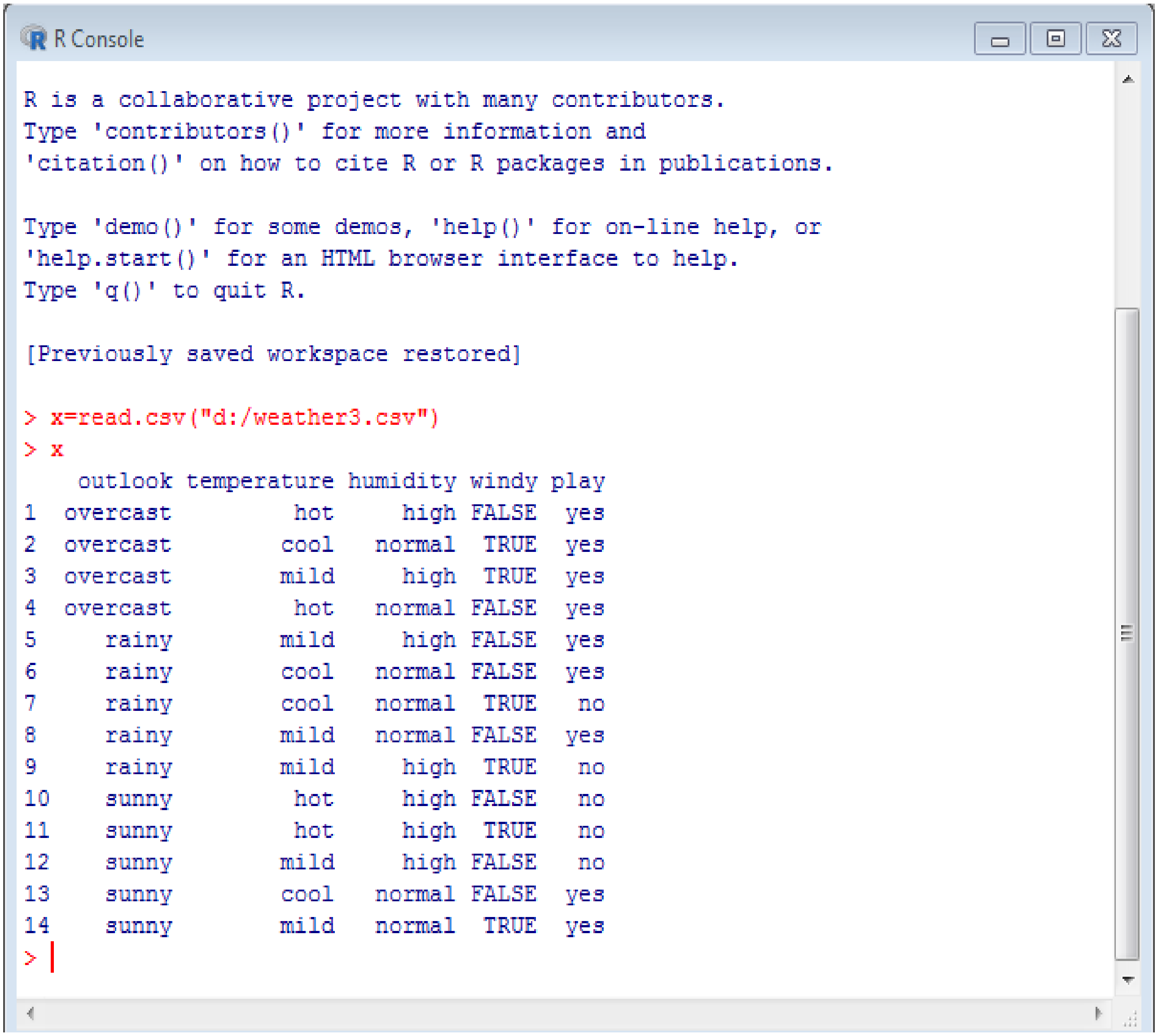
Practical N0: 10

Aim: Perform the logistic regression on the given data warehouse data

Step1: Open R studio -> Go on R console and type code.

1.X<-read.csv("C:/Users/Admin/Documents/SampleStudentData.csv")

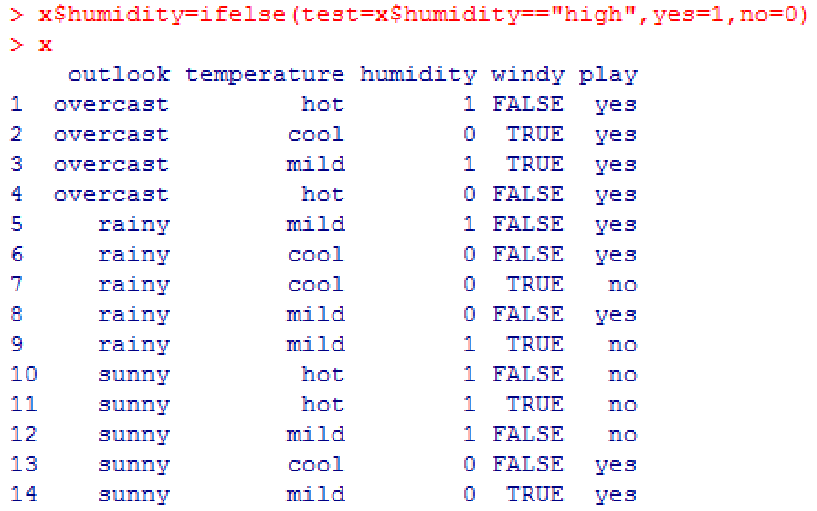
> X



```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> x=read.csv("d:/weather3.csv")
> x
    outlook temperature humidity windy play
1  overcast         hot     high FALSE  yes
2  overcast        cool   normal  TRUE  yes
3  overcast        mild     high  TRUE  yes
4  overcast         hot   normal FALSE  yes
5     rainy        mild     high FALSE  yes
6     rainy        cool   normal FALSE  yes
7     rainy        cool   normal  TRUE   no
8     rainy        mild   normal FALSE  yes
9     rainy        mild     high  TRUE   no
10    sunny         hot     high FALSE   no
11    sunny         hot     high  TRUE   no
12    sunny        mild     high FALSE   no
13    sunny        cool   normal FALSE  yes
14    sunny        mild   normal  TRUE  yes
> 
```

2.>x$humidity=ifelse(test=x$humidity=="high",yes=1,no=0)

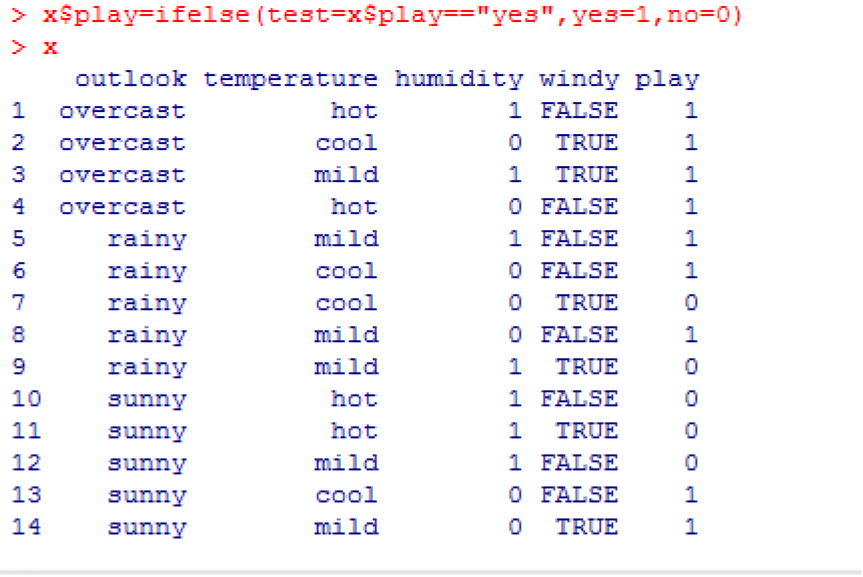
>x



```
> x$humidity=ifelse(test=x$humidity=="high",yes=1,no=0)
> x
    outlook temperature humidity windy play
1  overcast         hot        1 FALSE  yes
2  overcast        cool        0  TRUE  yes
3  overcast        mild        1  TRUE  yes
4  overcast         hot        0 FALSE  yes
5     rainy        mild        1 FALSE  yes
6     rainy        cool        0 FALSE  yes
7     rainy        cool        0  TRUE   no
8     rainy        mild        0 FALSE  yes
9     rainy        mild        1  TRUE   no
10    sunny         hot        1 FALSE   no
11    sunny         hot        1  TRUE   no
12    sunny        mild        1 FALSE   no
13    sunny        cool        0 FALSE  yes
14    sunny        mild        0  TRUE  yes
```

3.x$play=ifelse(test=x$play=="yes",yes=1,no=0)

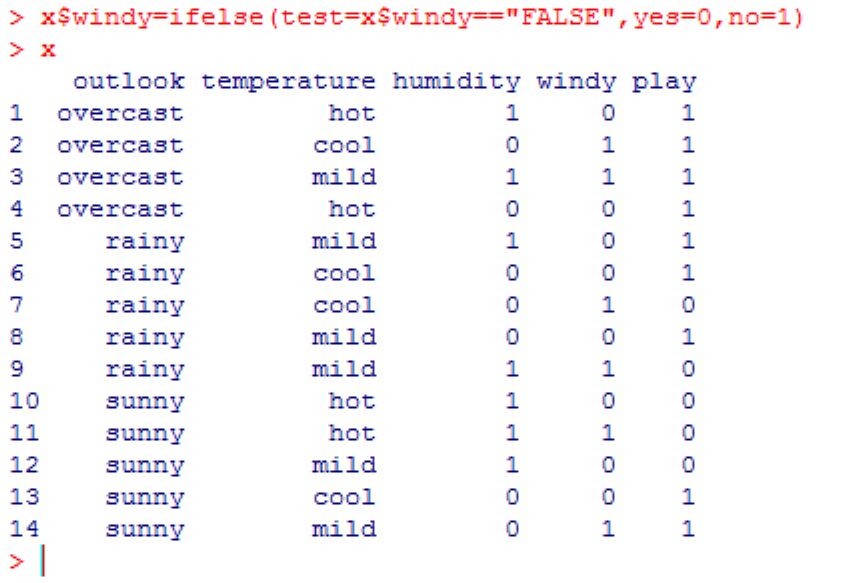
>x



```
> x$play=ifelse(test=x$play=="yes",yes=1,no=0)
> x
    outlook temperature humidity windy play
1  overcast         hot        1 FALSE    1
2  overcast        cool        0  TRUE    1
3  overcast        mild        1  TRUE    1
4  overcast         hot        0 FALSE    1
5     rainy        mild        1 FALSE    1
6     rainy        cool        0 FALSE    1
7     rainy        cool        0  TRUE    0
8     rainy        mild        0 FALSE    1
9     rainy        mild        1  TRUE    0
10    sunny         hot        1 FALSE    0
11    sunny         hot        1  TRUE    0
12    sunny        mild        1 FALSE    0
13    sunny        cool        0 FALSE    1
14    sunny        mild        0  TRUE    1
```

4.x$windy=ifelse(test=x$windy=="FALSE",yes=0,no=1)

>x



```
> x$windy=ifelse(test=x$windy=="FALSE",yes=0,no=1)
> x
    outlook temperature humidity windy play
1  overcast         hot        1     0    1
2  overcast        cool        0     1    1
3  overcast        mild        1     1    1
4  overcast         hot        0     0    1
5     rainy        mild        1     0    1
6     rainy        cool        0     0    1
7     rainy        cool        0     1    0
8     rainy        mild        0     0    1
9     rainy        mild        1     1    0
10    sunny         hot        1     0    0
11    sunny         hot        1     1    0
12    sunny        mild        1     0    0
13    sunny        cool        0     0    1
14    sunny        mild        0     1    1
> 
```

5. s=sample(nrow(x),.7*nrow(x))

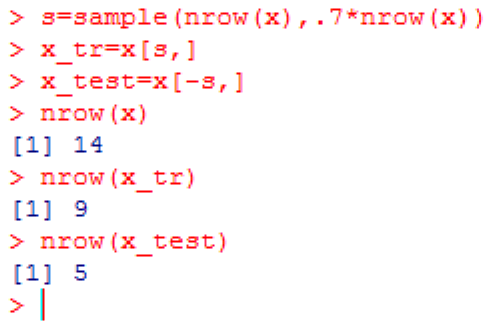
>x_tr=x[s,]

>x_test=x[-s,]

>nrow(x)

>nrow(x_tr)

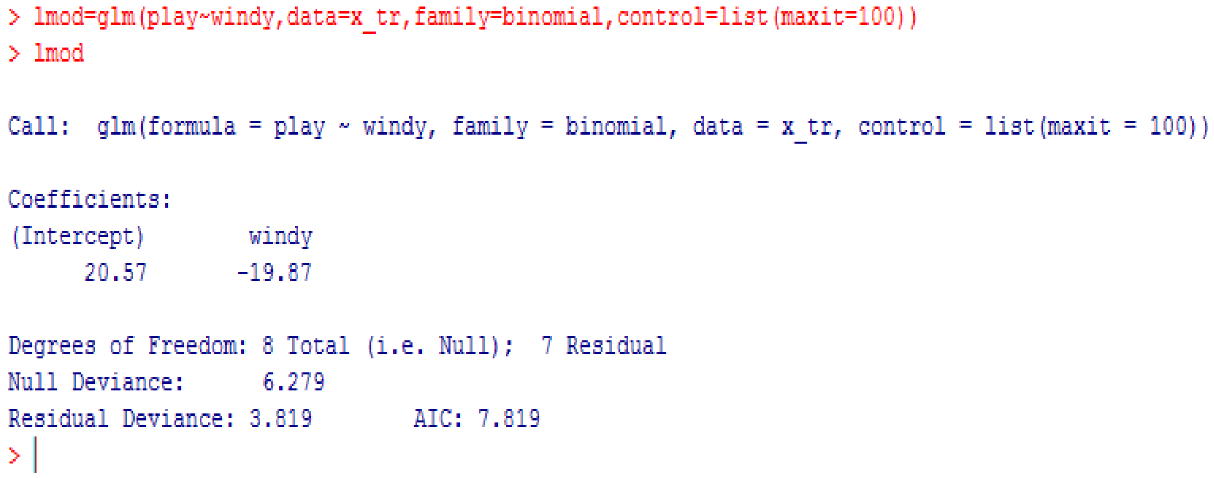
>nrow(x_test)



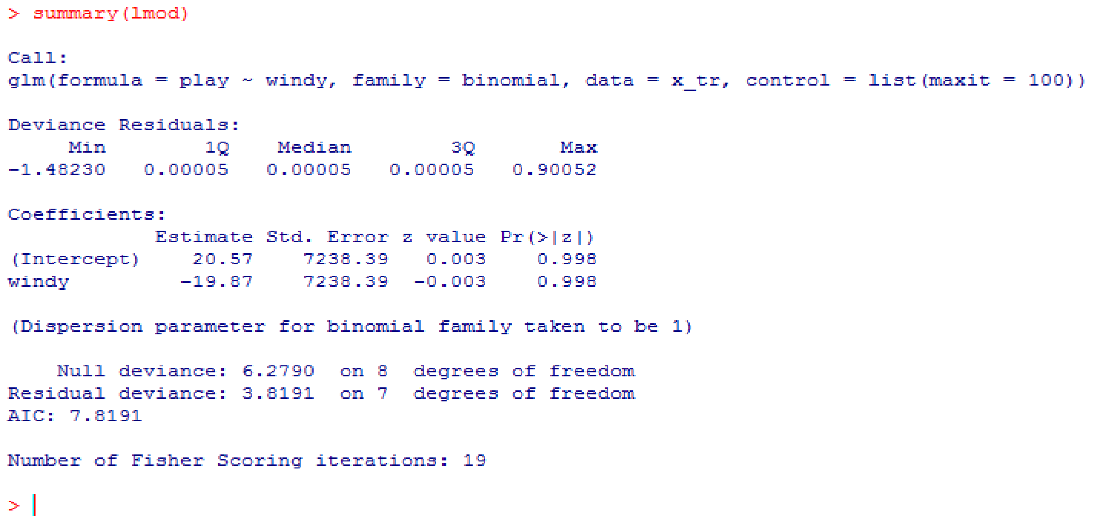
```
> s=sample(nrow(x),.7*nrow(x))
> x_tr=x[s,]
> x_test=x[-s,]
> nrow(x)
[1] 14
> nrow(x_tr)
[1] 9
> nrow(x_test)
[1] 5
> 
```

6.lmod=glm(play~windy,data=x_tr,family=binomial,control=list(maxit=100))

>lmod

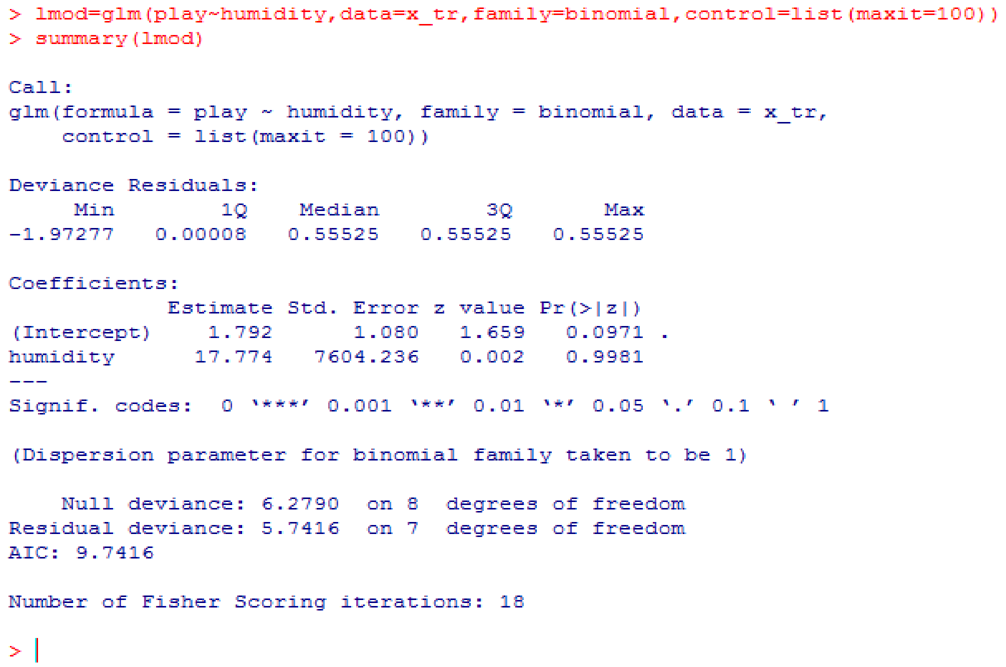


```
> lmod=glm(play~windy,data=x_tr,family=binomial,control=list(maxit=100))
> lmod

Call:  glm(formula = play ~ windy, family = binomial, data = x_tr, control = list(maxit = 100))

Coefficients:
(Intercept)        windy  
      20.57       -19.87  

Degrees of Freedom: 8 Total (i.e. Null);  7 Residual
Null Deviance:      6.279 
Residual Deviance: 3.819        AIC: 7.819
> |
```



```
> summary(lmod)

Call:
glm(formula = play ~ windy, family = binomial, data = x_tr, control = list(maxit = 100))

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.48230   0.00005   0.00005   0.00005   0.90052  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    20.57    7238.39   0.003    0.998
windy         -19.87    7238.39  -0.003    0.998

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6.2790  on 8  degrees of freedom
Residual deviance: 3.8191  on 7  degrees of freedom
AIC: 7.8191

Number of Fisher Scoring iterations: 19

> |
```

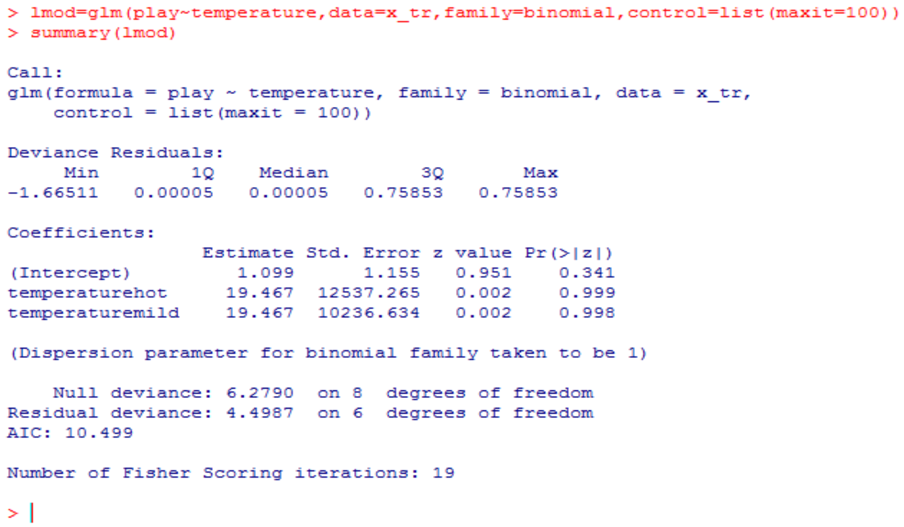
>summary(lmod)



```
> lmod=glm(play~humidity,data=x_tr,family=binomial,control=list(maxit=100))
> summary(lmod)

Call:
glm(formula = play ~ humidity, family = binomial, data = x_tr,
    control = list(maxit = 100))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.97277   0.00008   0.55525   0.55525   0.55525

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.792      1.080   1.659   0.0971 .
humidity      17.774   7604.236   0.002   0.9981
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6.2790  on 8  degrees of freedom
Residual deviance: 5.7416  on 7  degrees of freedom
AIC: 9.7416

Number of Fisher Scoring iterations: 18

> |
```

>lmod=glm(play~temperature,data=x_tr,family=binomial,control=list(maxit=100))

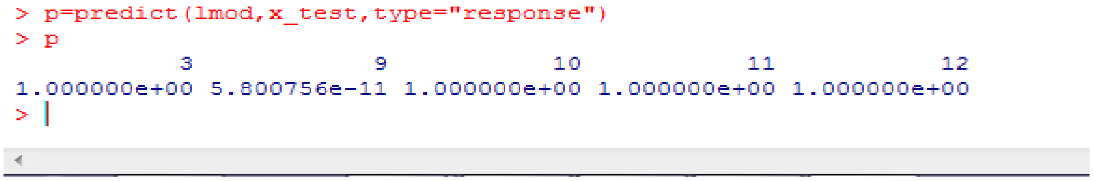
>summary(lmod)



```
> lmod=glm(play~temperature,data=x_tr,family=binomial,control=list(maxit=100))
> summary(lmod)

Call:
glm(formula = play ~ temperature, family = binomial, data = x_tr,
    control = list(maxit = 100))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.66511   0.00005   0.00005   0.75853   0.75853

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)        1.099      1.155   0.951    0.341
temperaturehot    19.467  12537.265   0.002    0.999
temperaturemild   19.467  10236.634   0.002    0.998

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6.2790  on 8  degrees of freedom
Residual deviance: 4.4987  on 6  degrees of freedom
AIC: 10.499

Number of Fisher Scoring iterations: 19

> |
```

8.p=predict(lmod,x_test,type="response")

>p



```
> p=predict(lmod,x_test,type="response")
> p
           3            9           10           11           12
1.000000e+00 5.800756e-11 1.000000e+00 1.000000e+00 1.000000e+00
> |
```