

Practical 1

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

Solution:

In Power BI:

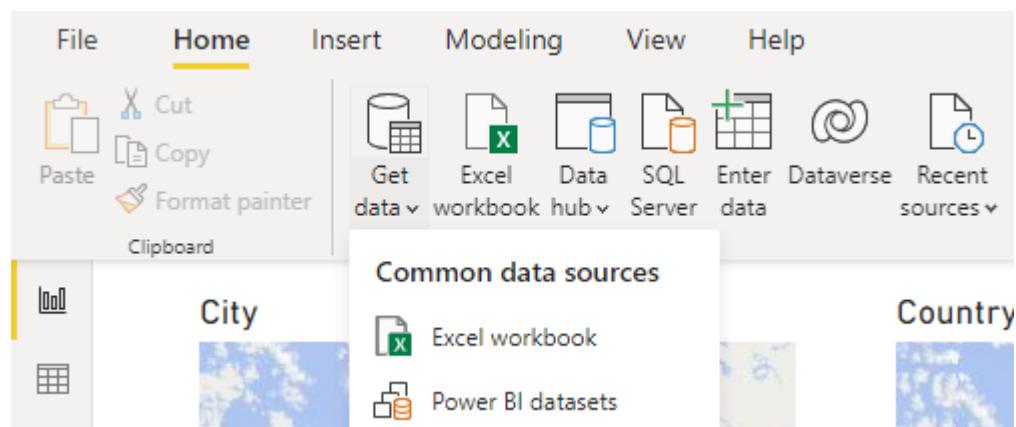
Steps:

<https://github.com/microsoft/powerbi-desktop-samples/blob/main/AdventureWorks%20Sales%20Sample/AdventureWorks%20Sales.xlsx>

Download excel sheet

Import excel sheet in Power BI

Select Get Data from home tab and select Excel workbook.



Recent files			
AdventureWorks Sales	06-12-2022 14:21	Microsoft Excel W...	13,988 KB
SYBSc-CS, Sem 3, Reg, batch 2021-24	05-12-2022 11:55	Microsoft Excel W...	178 KB
ChapterY.1.0.6	22-11-2022 12:13	File folder	
PortableGit	11-11-2022 14:13	File folder	



Navigator

Display Options

X

AdventureWorks Sales.xlsx [14]

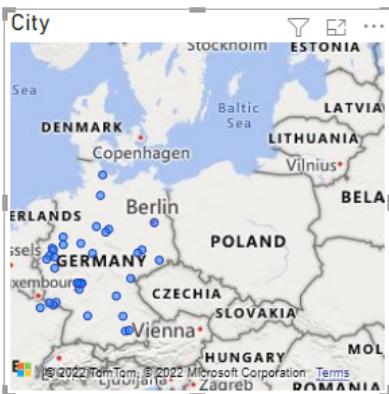
- Customer
- Date
- Product
- Reseller
- Sales
- SalesOrder
- SalesTerritory
- Customer_data
- Date_data
- Product_data
- Reseller_data
- Sales Order_data
- Sales Territory_data
- Sales_data

Date

DateKey	Date	Fiscal Year	Fiscal Quarter	Month	Full
20170701	01-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170702	02-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170703	03-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170704	04-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170705	05-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170706	06-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170707	07-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170708	08-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170709	09-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170710	10-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170711	11-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170712	12-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170713	13-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170714	14-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170715	15-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170716	16-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170717	17-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170718	18-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170719	19-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170720	20-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170721	21-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170722	22-07-2017	FY2018	FY2018 Q1	01-07-2017	
20170723	23-07-2017	FY2018	FY2018 Q1	01-07-2017	

Load
Transform Data
Cancel

Now select the fields which you want to see.



Filters

Search

Filters on this visual

City
is (All)

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Visualizations

Build visual





R Py





Fields

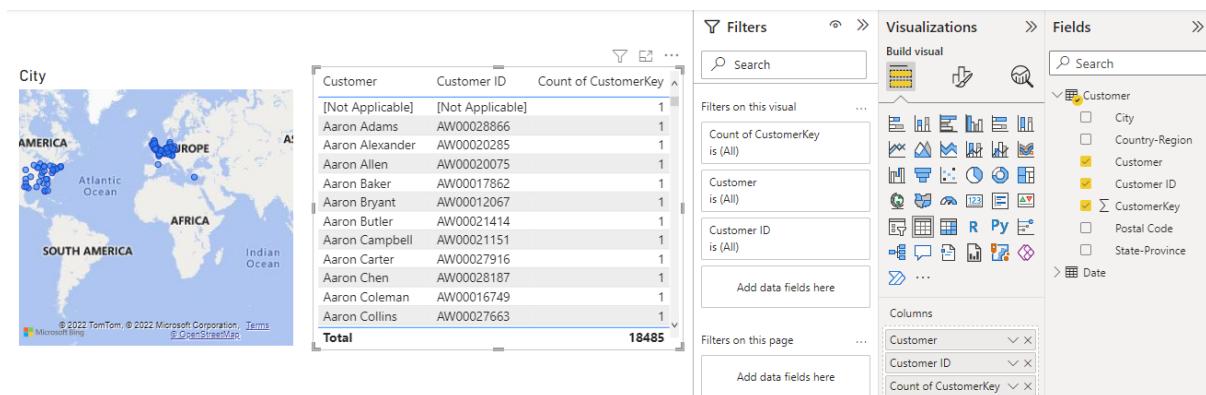
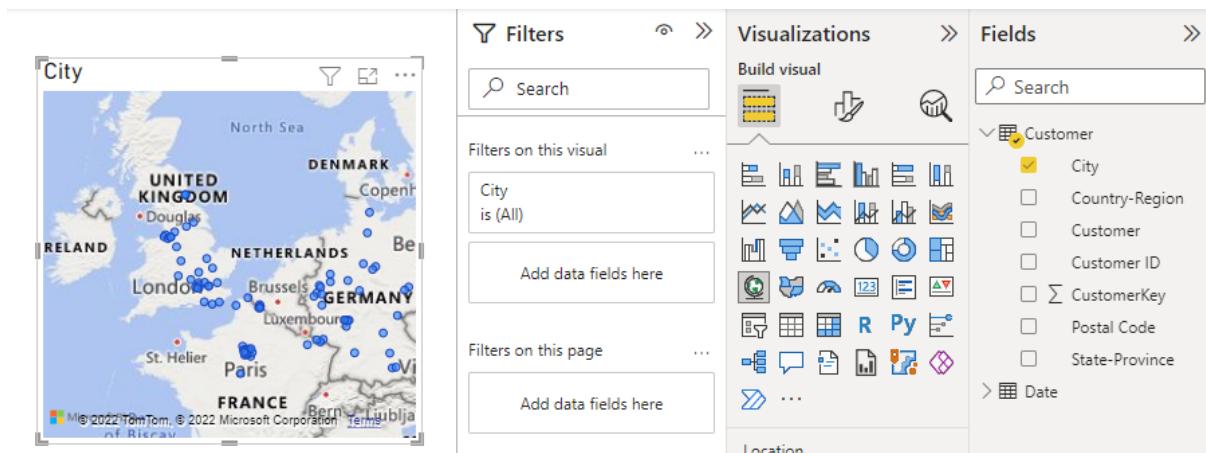
Search

Customer

- City
- Country-Region
- Customer
- Customer ID
- CustomerKey
- Postal Code
- State-Province

Date

Swapnil Ramgire T.20.83

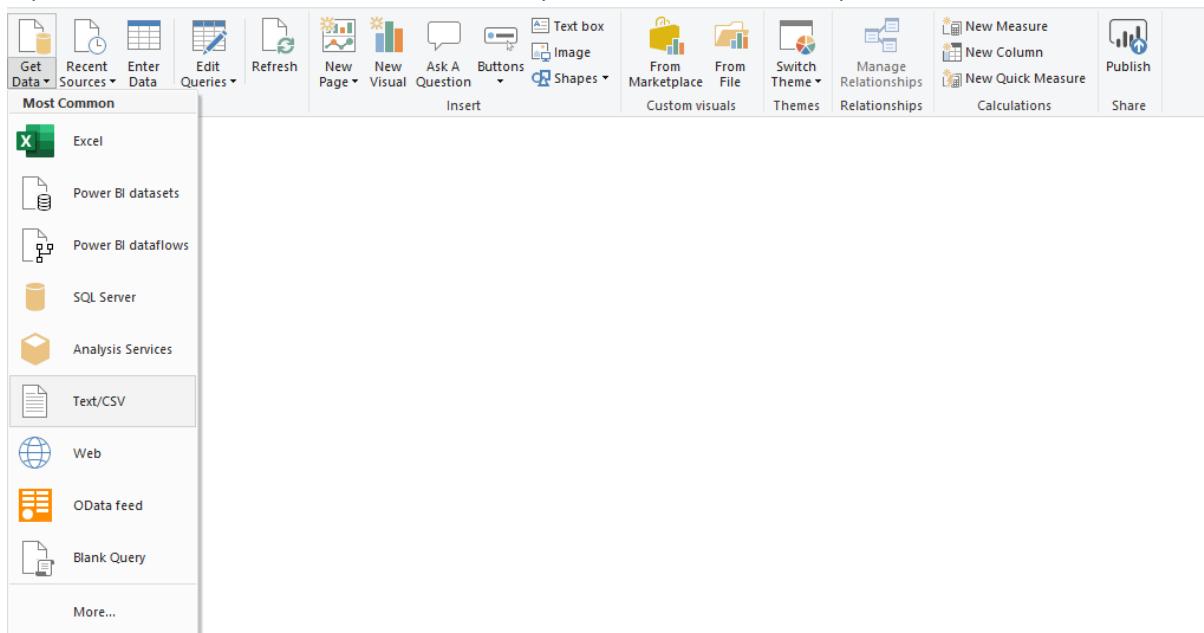


Importing through FlatFile/TextFile.

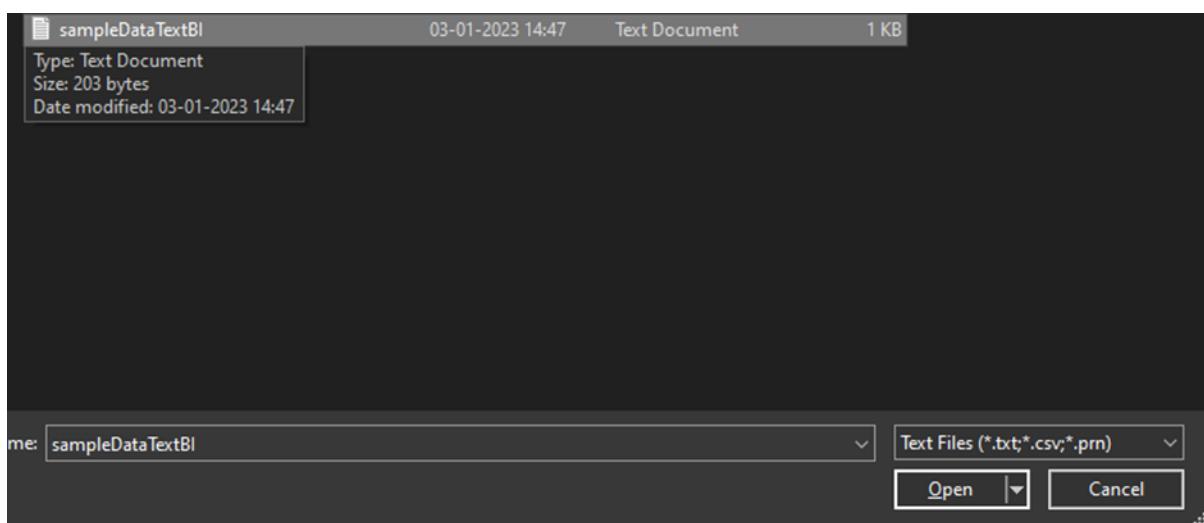
sampleDataTextBI - Notepad
File Edit Format View Help

```
name,rollNum
Adams,1
Baker,2
Clark,3
Davis,4
Evans,5
Frank,6
Ghosh,7
Hills,8
Irwin,9
Jones,10
Klein,11
Lopez,12
Mason,13
Nalty,14
Ochoa,15
Patel,16
Quinn,17
Reily,18
Smith,19
Trott,20
```

Open Power BI and select Text/CSV option from Get Data present on Home tab.



Select the saved TextFile



sampleDataTextBl.txt

File Origin: 1252: Western European (Windows) Delimiter: Comma Data Type Detection: Based on first 200 rows

name	rollNum
Adams	1
Baker	2
Clark	3
Davis	4
Evans	5
Frank	6
Ghosh	7
Hills	8
Irwin	9
Jones	10
Klein	11
Lopez	12
Mason	13
Nalty	14
Ochoa	15
Patel	16
Quinn	17
Reilly	18
Smith	19
Trott	20

Load Transform Data Cancel

Select the fields.

name	rollNum
Adams	1
Baker	2
Clark	3
Davis	4
Evans	5
Frank	6
Ghosh	7
Hills	8
Irwin	9
Jones	10
Klein	11
Total	210

Filters

Filters on this visual

- name is (All)
- rollNum is (All)
- Add data fields here

Filters on this page

- Add data fields here

Visualizations

Fields

Search

sampleDataTextBl

- name
- rollNum

Values

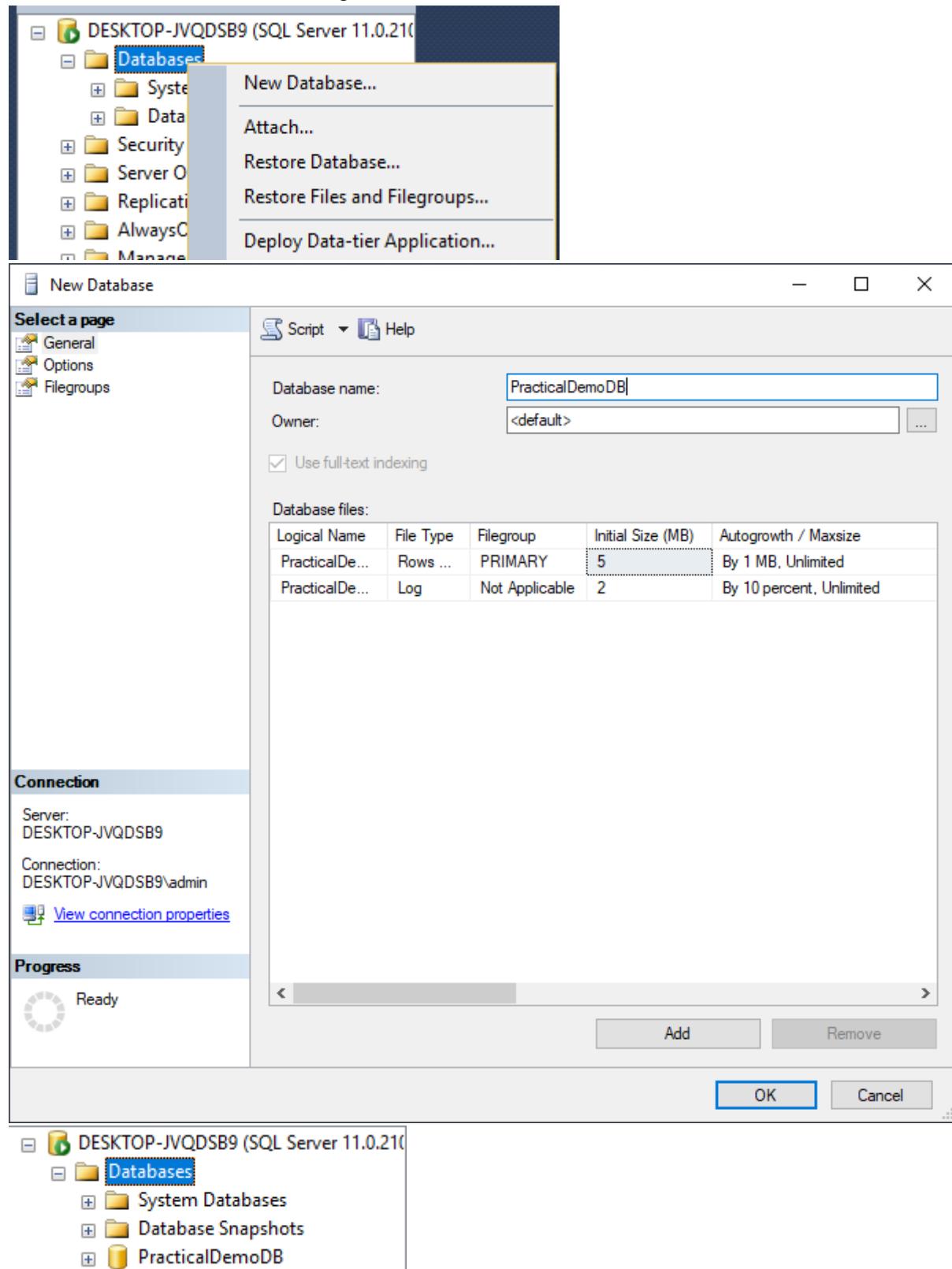
name	▼	X
rollNum	▼	X

In SQL Server Management Studio and SQL Server Data Tools

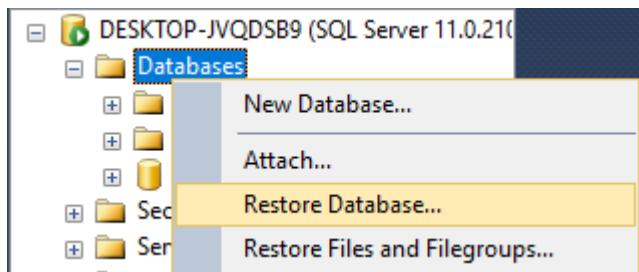
Importing data from Database

Steps:

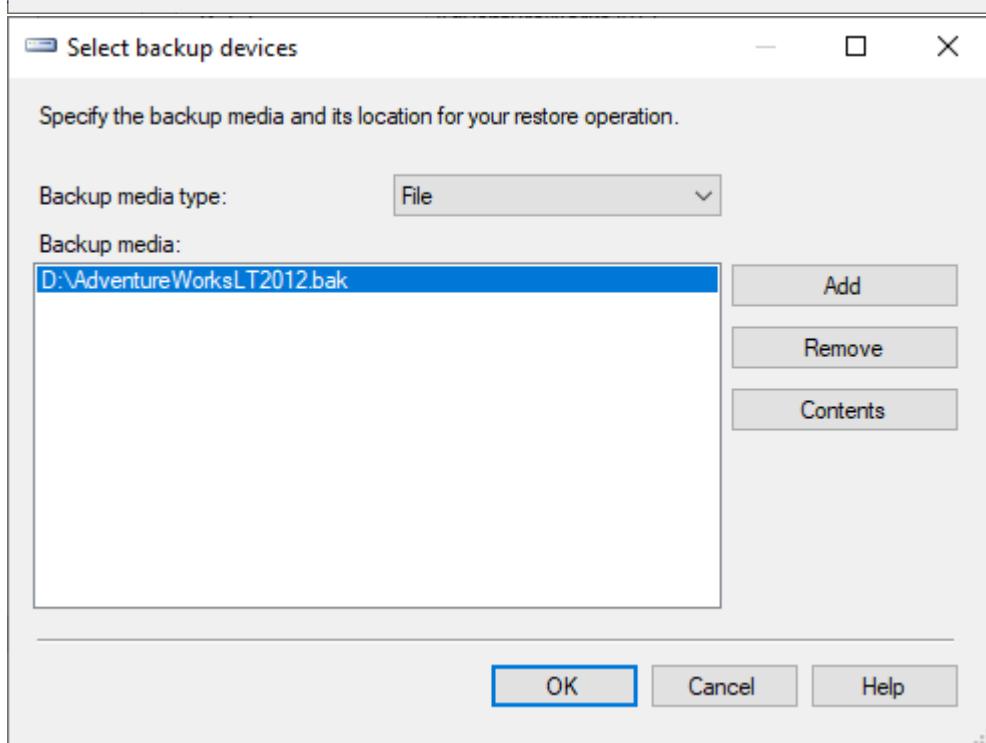
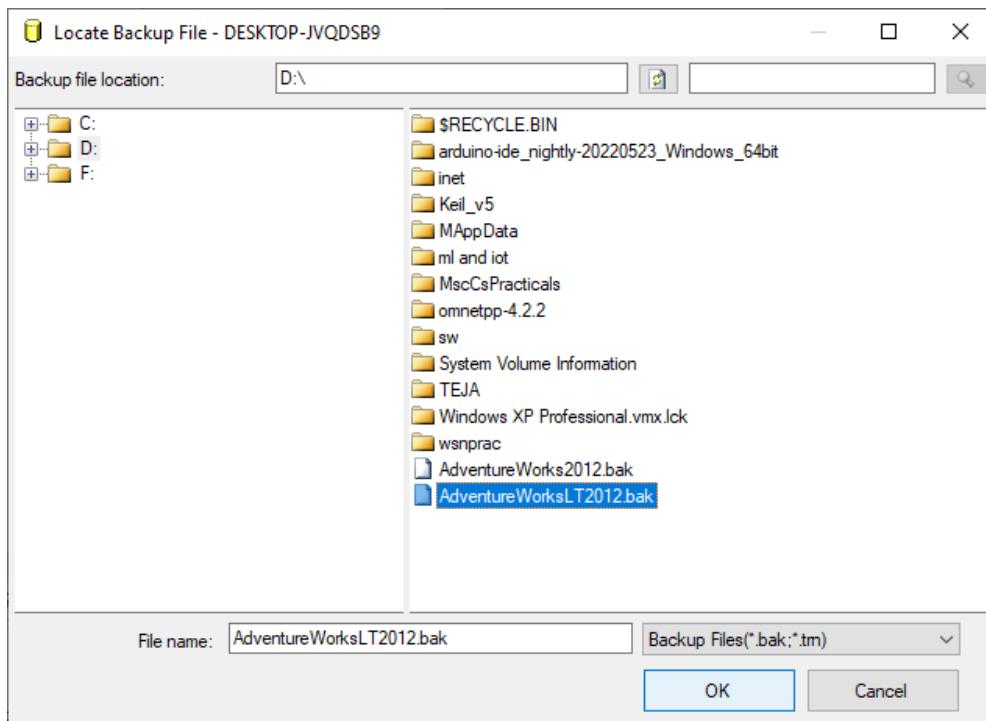
Launch the SQL server management studio and Create new database

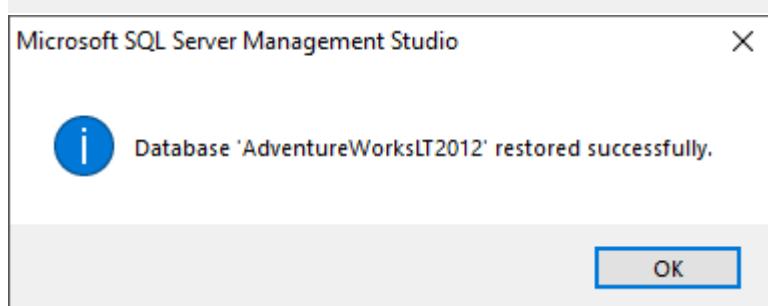
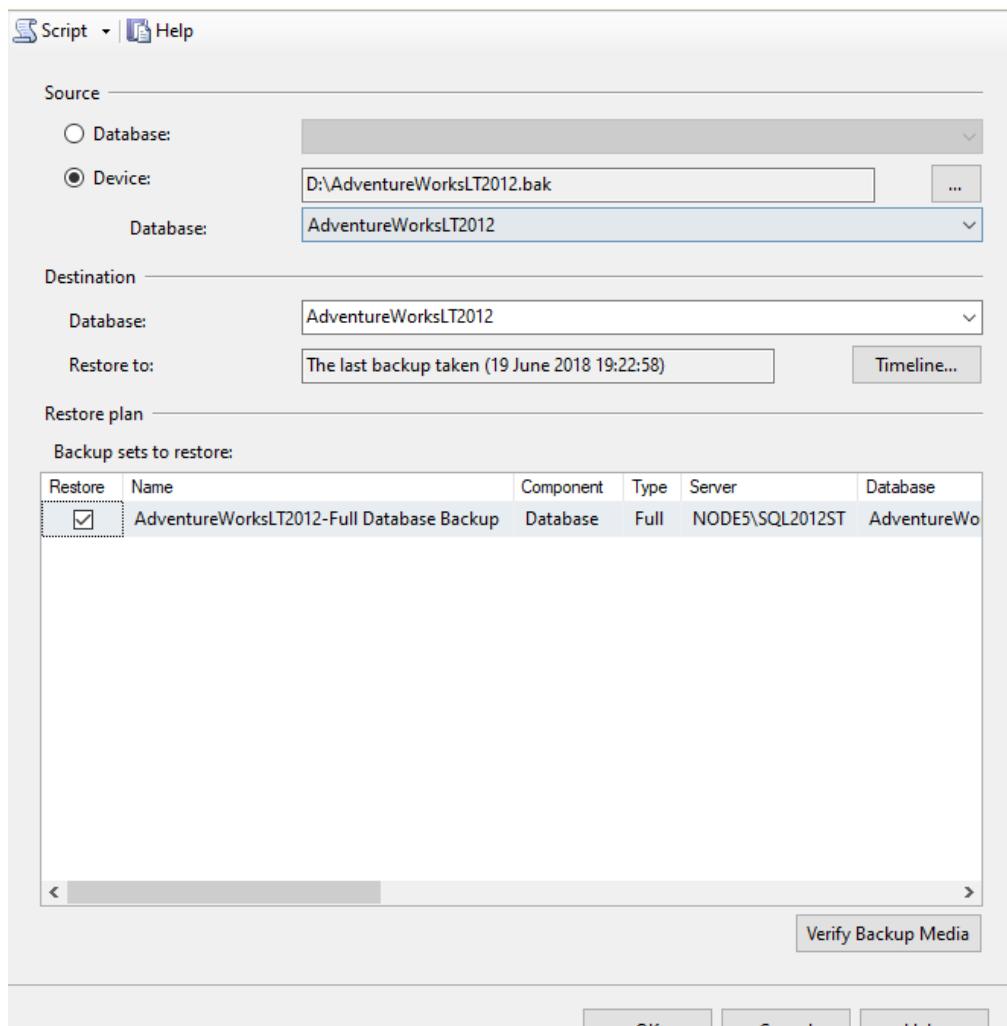


Now restore database



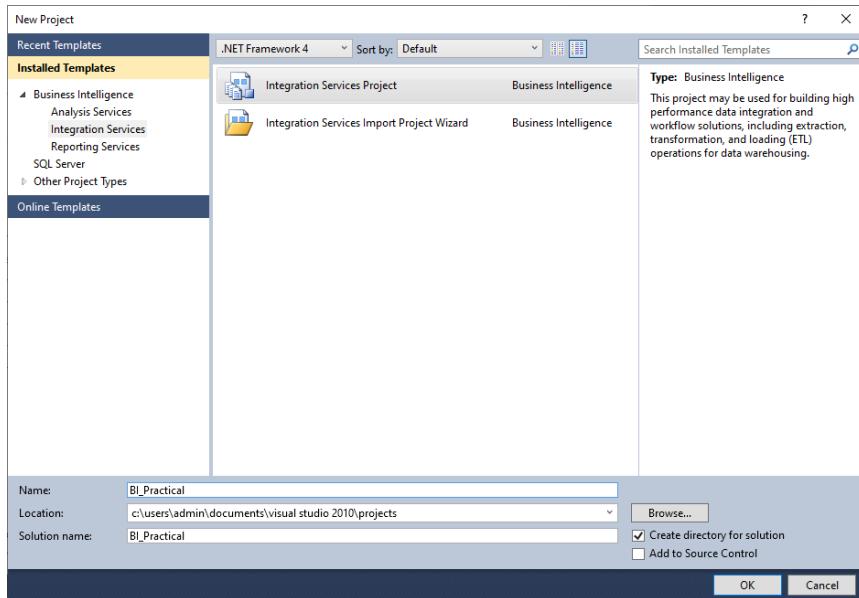
Select the downloaded database file.



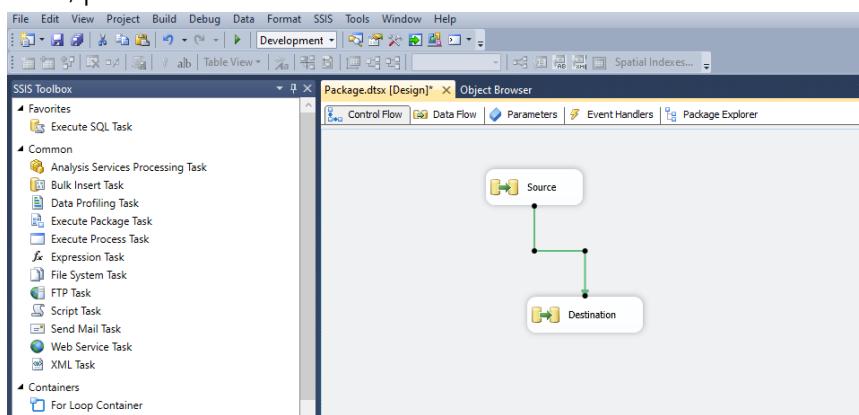


Now launch SQL server data tools

Create new project

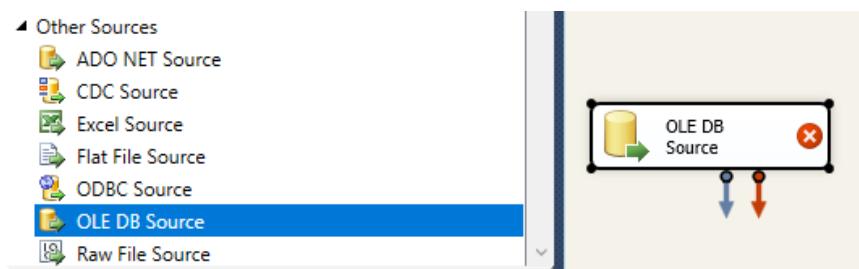


Now, place 2 Data Flow task

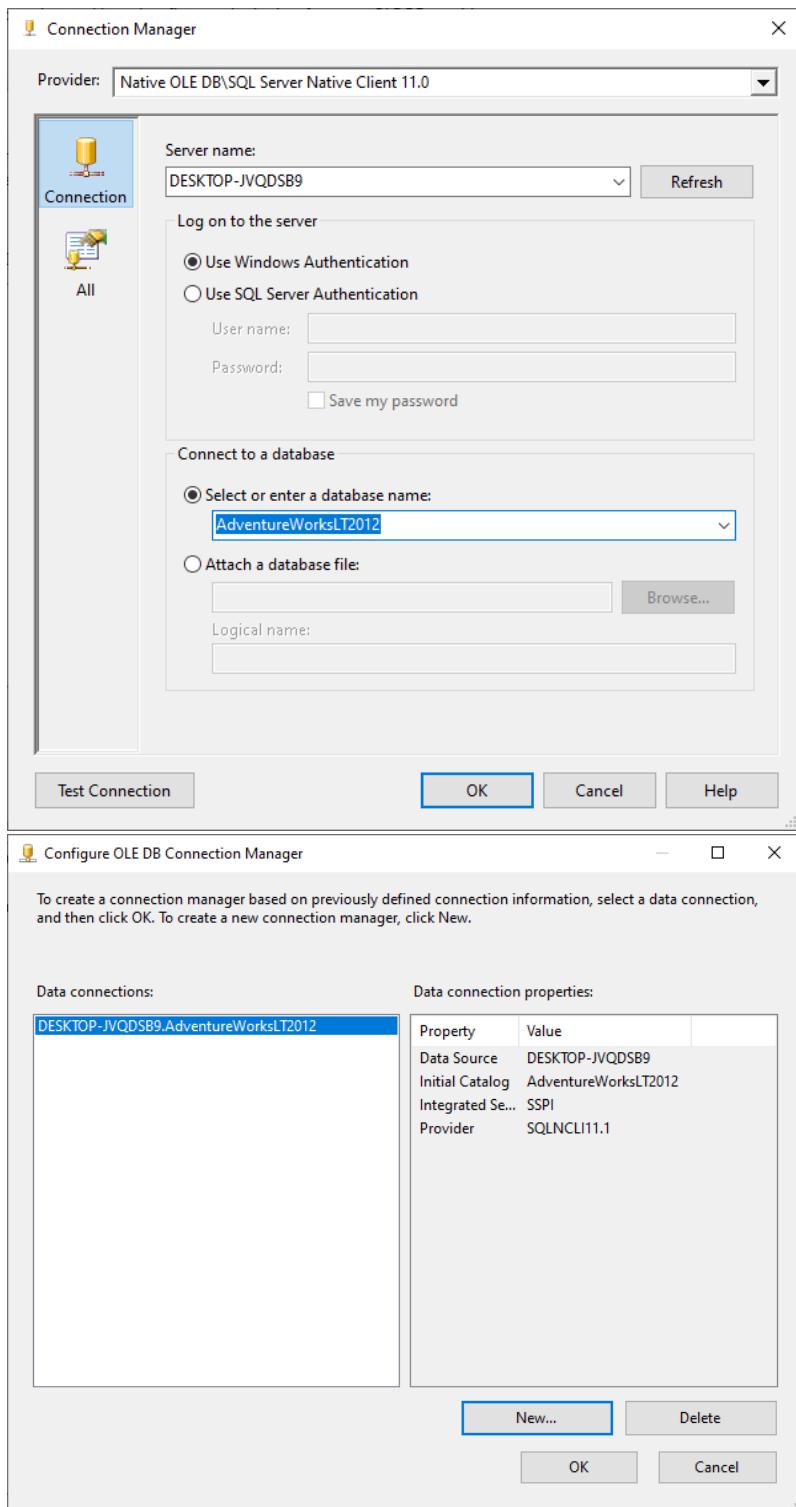


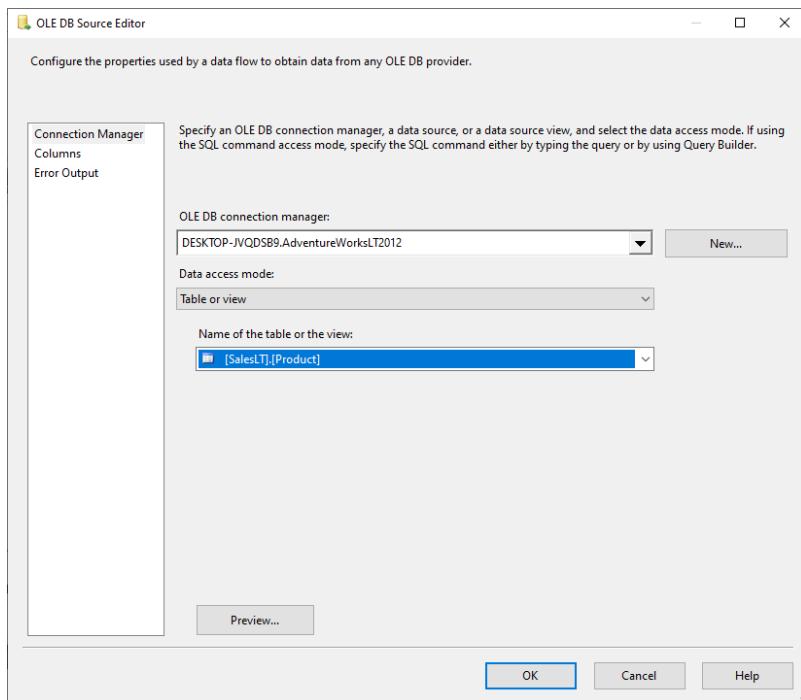
Go to source:

Add ole DB source



Double click on OLE DB source and add connection



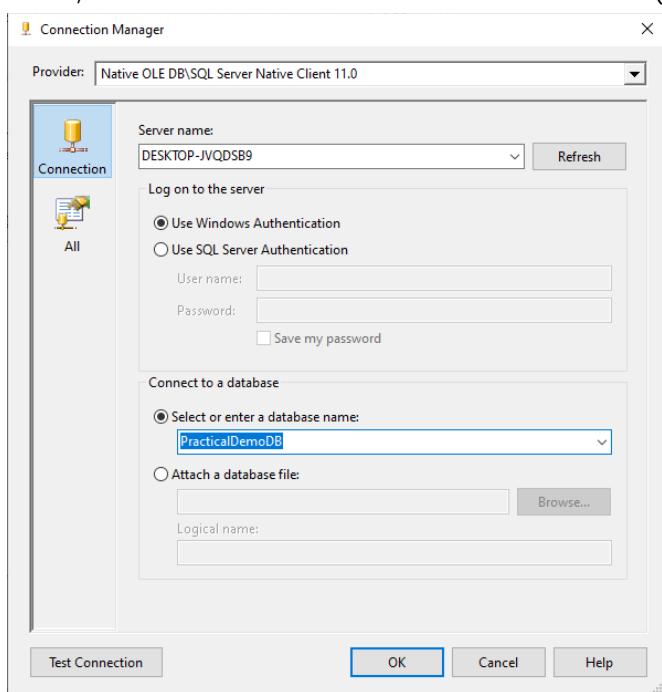


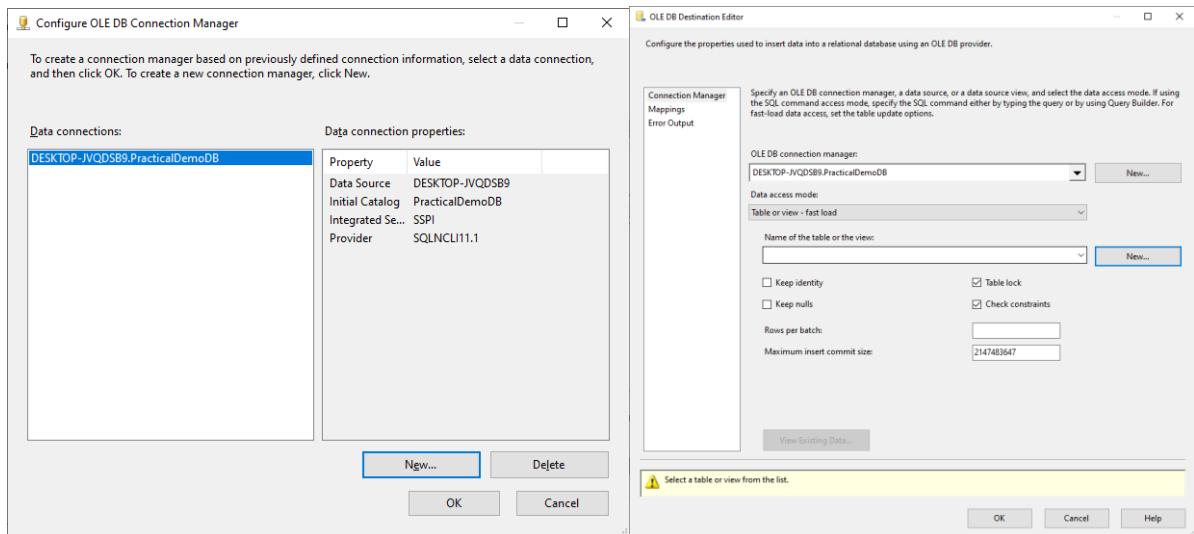
Now add one OLE DB destination

- ▲ Other Destinations
 - ADO NET Destination
 - Data Mining Model Training
 - DataReader Destination
 - Dimension Processing
 - Excel Destination
 - Flat File Destination
 - ODBC Destination
 - OLE DB Destination**
 - Partition Processing

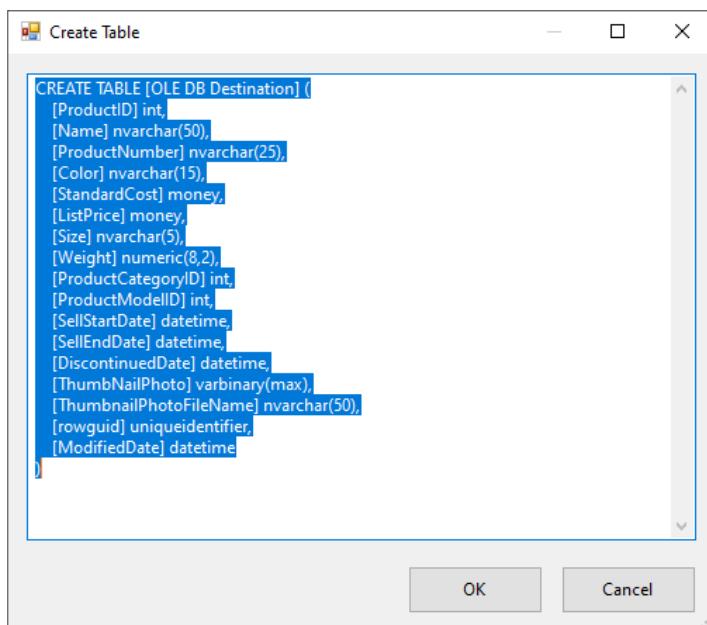


Then, add connection for created database (PracticalDemoDB)

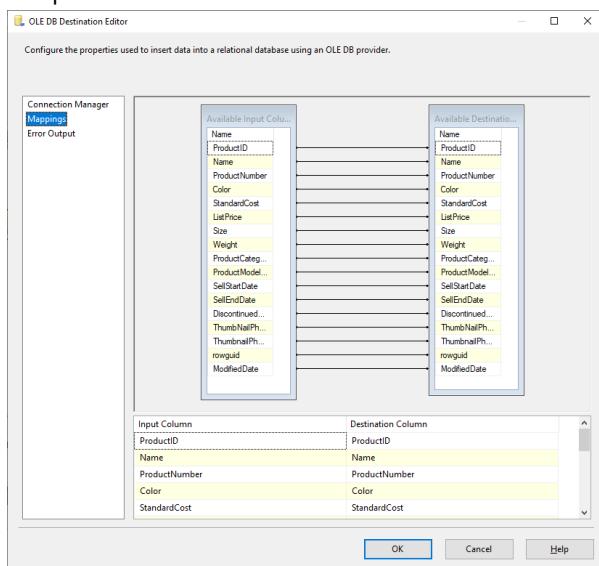


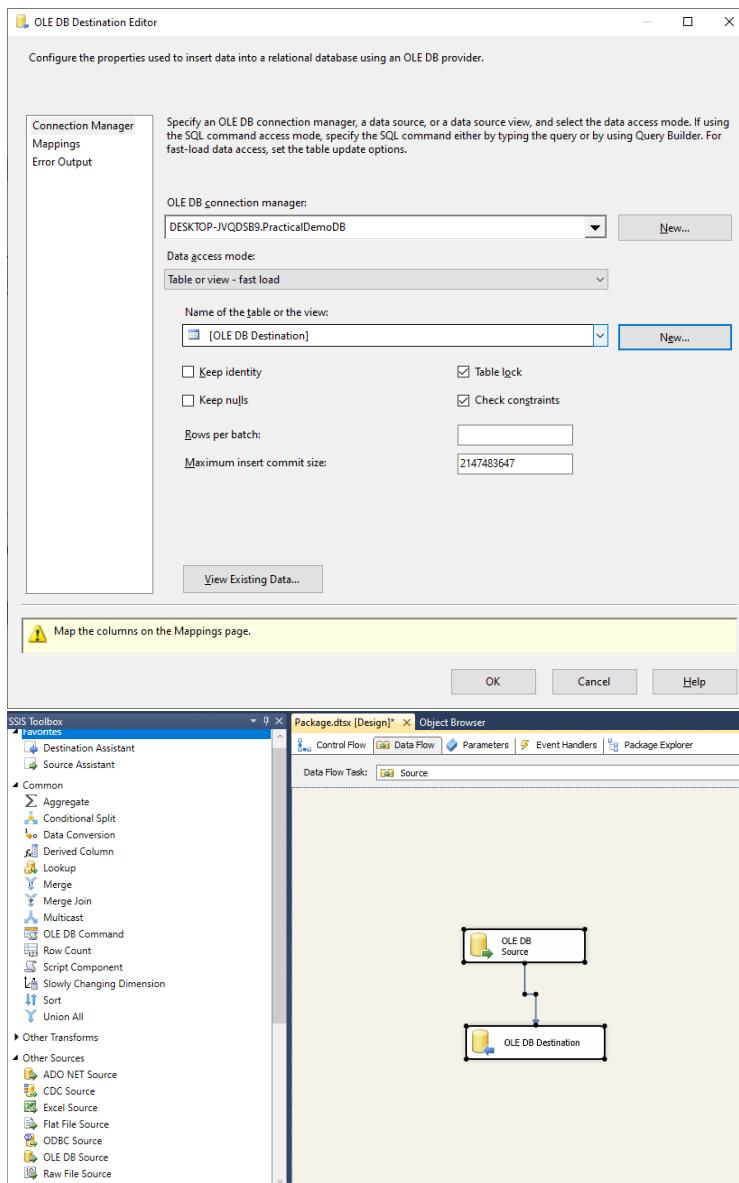


Click on “New...” to create table

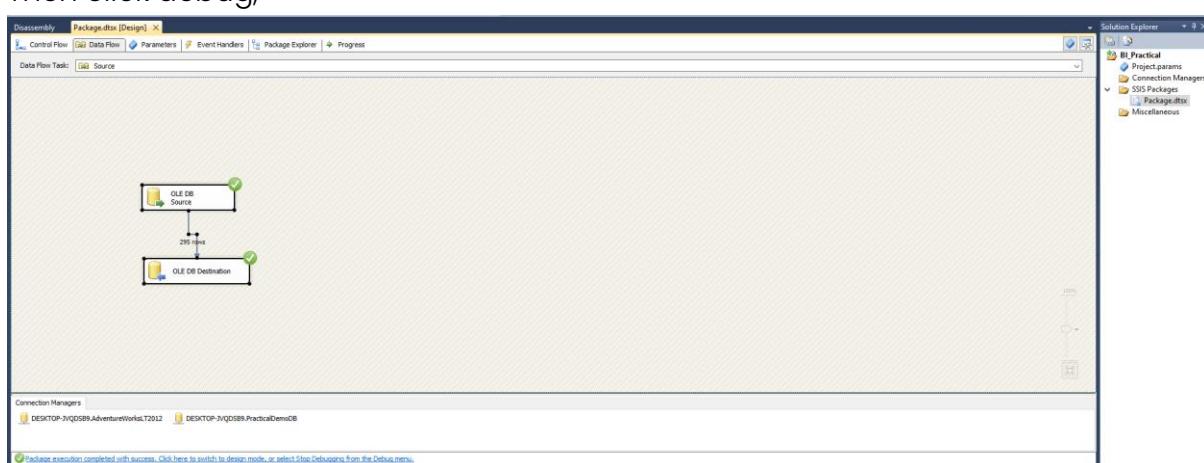


Map the columns





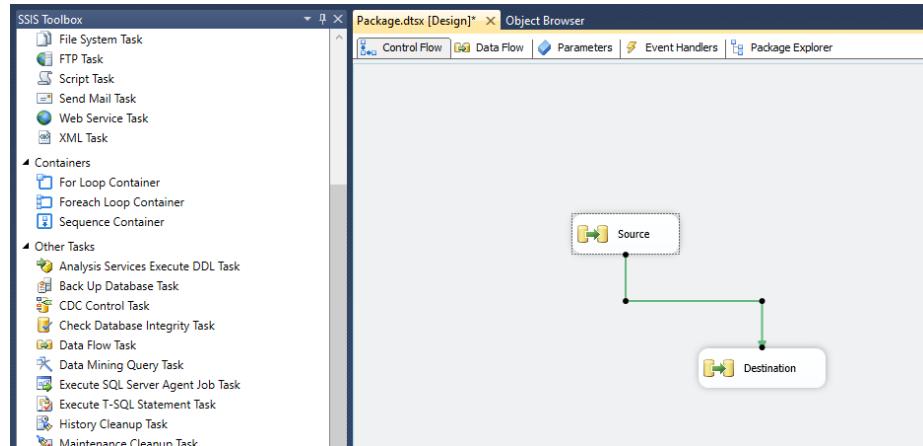
Then click debug,



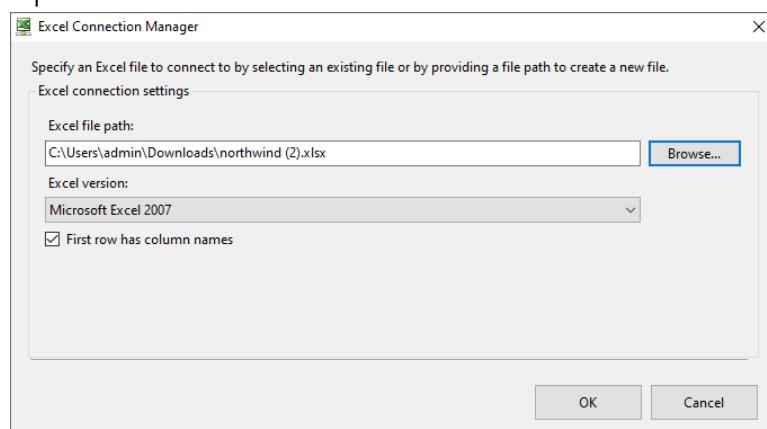
OUTPUT (Verifying it in SQL server management studio)

Importing Excel and loading it the system.

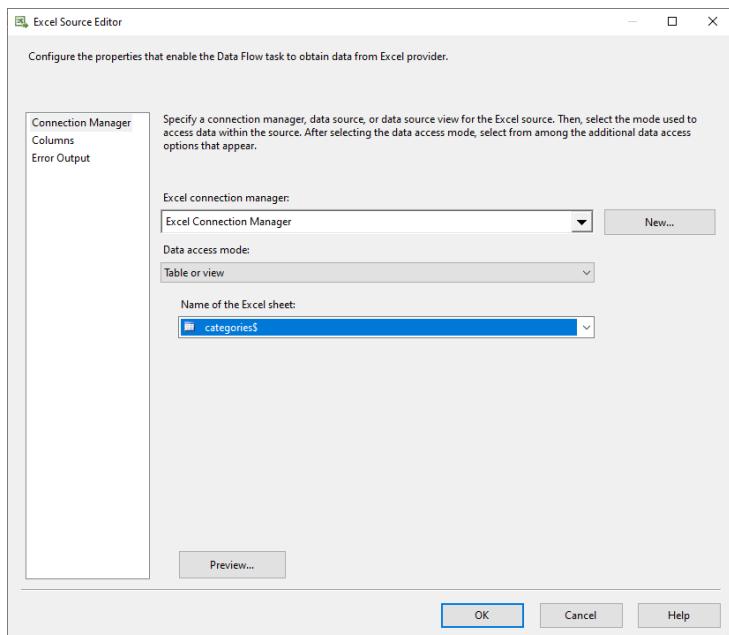
Place 2 Data Flow task.



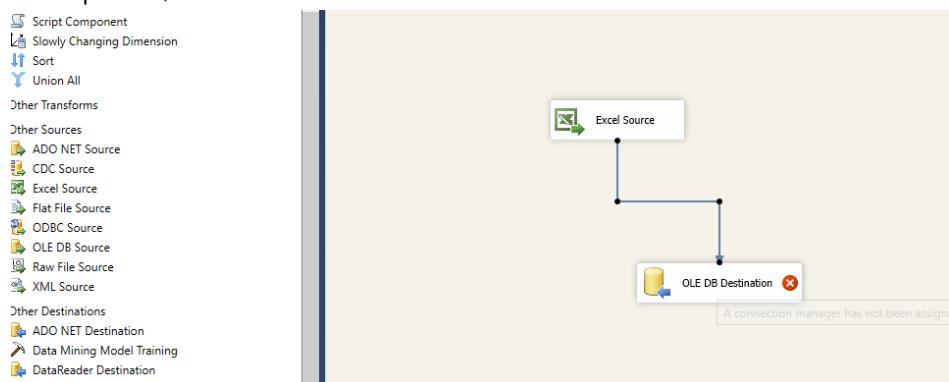
Open source and Place one Excel Source and choose your excel sheet



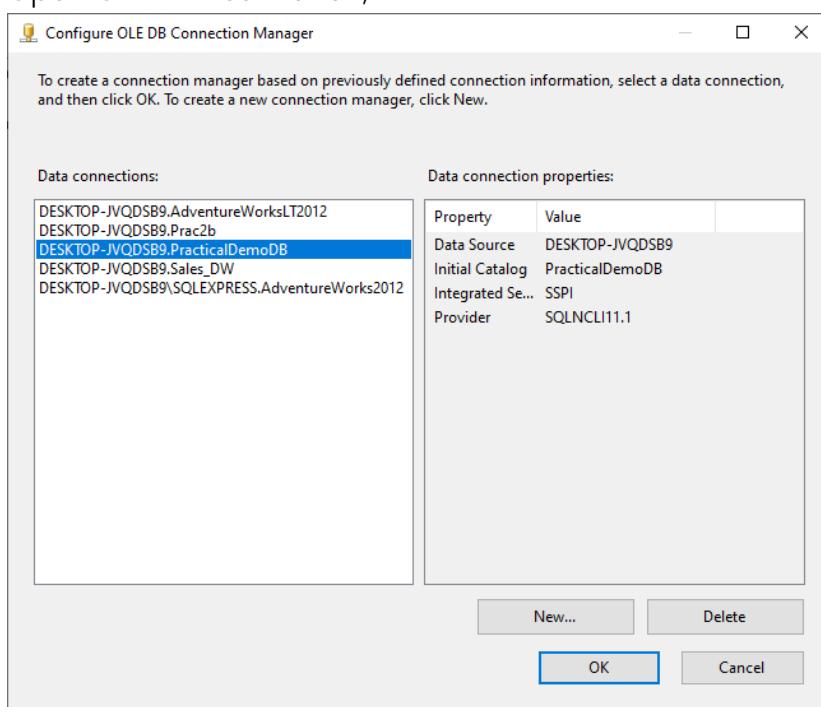
Swapnil Ramgire T.20.83

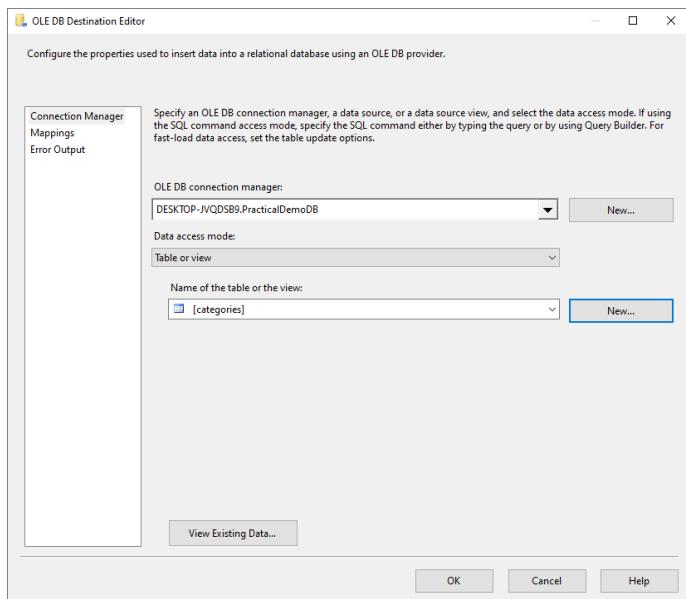


Now place, OLE DB Destination

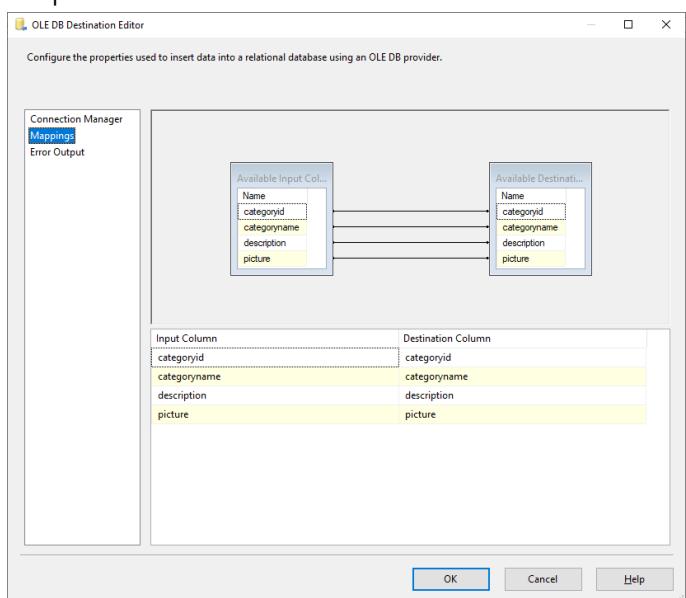


Open OLE DB Destination,



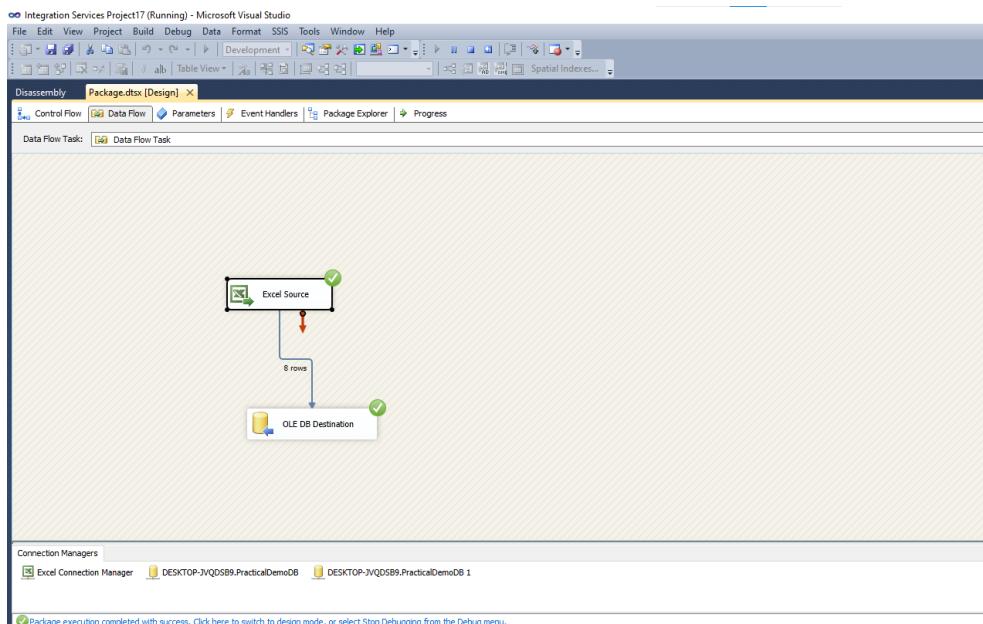


Map the columns



Now execute it,





Verifying it in Sql management studio

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The left pane shows the Object Explorer with the database 'PracticalDemoDB' selected. The right pane contains a query window titled 'SQLQuery1.sql - DESKTOP-JVQDSB9\administrator (51)' displaying the following T-SQL script:

```

***** Script for SelectTopRows command from SSMS *****
SELECT TOP 400 [categoryid]
      ,[categoryname]
      ,[description]
      ,[picture]
  FROM [PracticalDemoDB].[dbo].[categories]

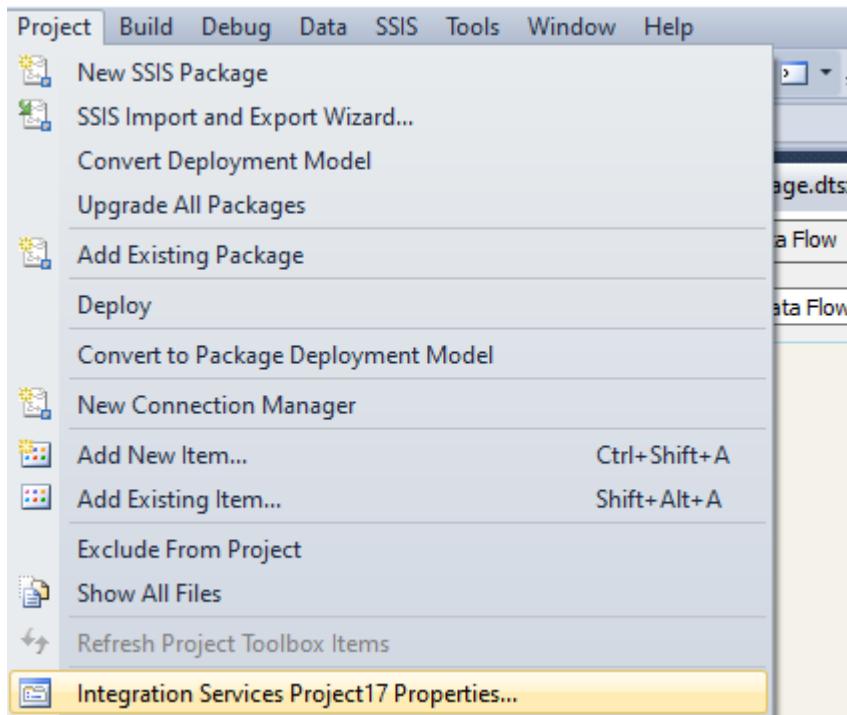
```

Below the script, the 'Results' tab shows the output of the query as a table:

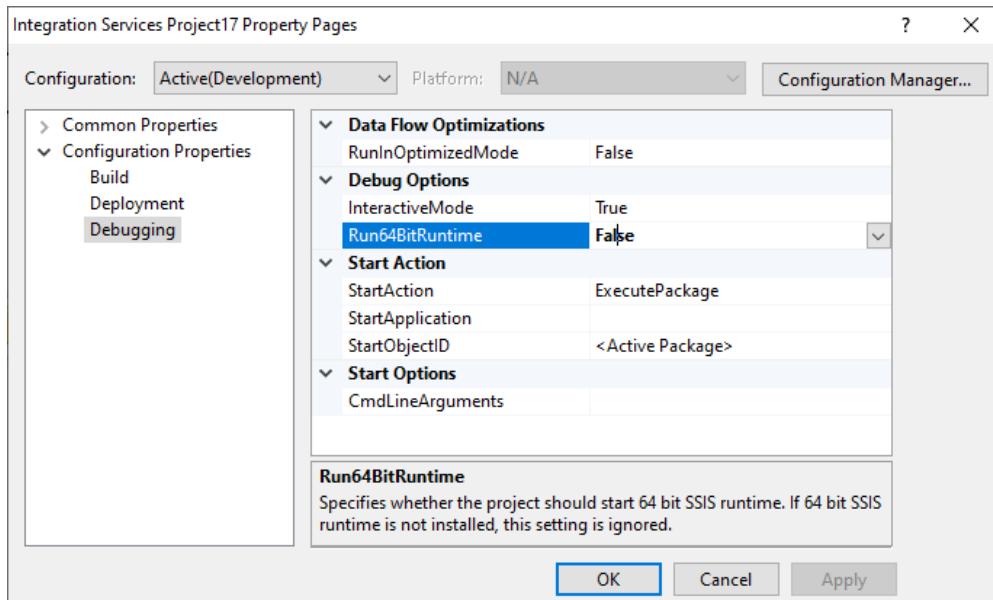
	categoryid	categoryname	description	picture
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales	☒
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	☒
3	3	Confections	Desserts, candies, and sweet breads	☒
4	4	Dairy Products	Cheeses	☒
5	5	Grains/Cereals	Bread, crackers, pasta, and cereal	☒
6	6	Meat/Poultry	Prepared meats	☒
7	7	Produce	Dried fruit and bean curd	☒
8	8	Seafood	Seaweed and fish	☒
9	1	Beverage	Soft drinks, coffees, teas, beers, and ales	☒
10	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	☒
11	3	Confections	Desserts, candies, and sweet breads	☒
12	4	Dairy Products	Cheeses	☒
13	5	Grains/Cereals	Bread, crackers, pasta, and cereal	☒
14	6	Meat/Poultry	Prepared meats	☒
15	7	Produce	Dried fruit and bean curd	☒
16	8	Seafood	Seaweed and fish	☒

If error occurs while importing the excel data, follow below steps.

Go to Project properties.



Set Run64BitRuntime to false.



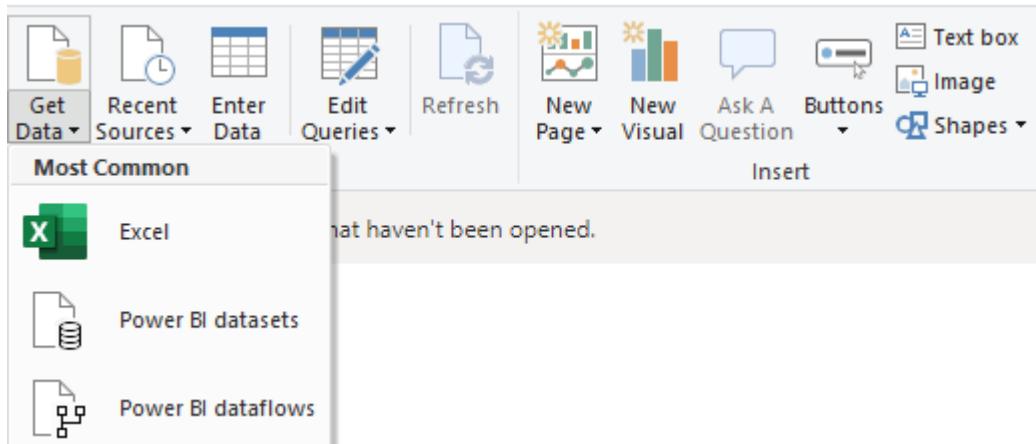
Practical 2

AIM: Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver.

Solution:

In Power Bi,

Import your desired data,



Select Tables.

A screenshot of the Microsoft Power BI Navigator dialog box. On the left, there is a tree view of data sources and tables. Under 'AdventureWorks Sales.xlsx [14]', the 'Customer' table is selected, indicated by a checked checkbox. Other tables listed include Date, Product, Reseller, Sales, SalesOrder, SalesTerritory, Customer_data, Date_data, Product_data, Reseller_data, Sales Order_data, Sales Territory_data, and Sales_data. On the right, a preview of the 'Date' table is shown in a grid format. The columns are DateKey, Date, Fiscal Year, Fiscal Quarter, Month, and Full. The data shows dates from July 2017, categorized by year, quarter, month, and day. At the bottom of the dialog box, there are 'Load', 'Transform Data', and 'Cancel' buttons.

Click on Transform Data

Untitled - Power Query Editor

File Home Transform Add Column View Help

Queries [1] Customer

CustomerKey CustomerID Customer City State-Province CountryRegion PostalCode

1 [Not Applicable] [Not Applicable] [Not Applicable] [Not Applicable] [Not Applicable] [Not Applicable]

2 11000 AW00011000 Jon Yang Rockhampton Queensland Australia 4700

3 11001 AW00011001 Eugene Huang Seaford Victoria Australia 3198

4 11002 AW00011002 Ruben Torres Hobart Tasmania Australia 7001

5 11003 AW00011003 Christy Zhu North Ryde New South Wales Australia 2113

6 11004 AW00011004 Elizabeth Johnson Wollongong New South Wales Australia 2500

7 11005 AW00011005 Julio Ruiz East Brisbane Queensland Australia 4169

8 11006 AW00011006 Janet Alvarez Matraville New South Wales Australia 2036

9 11007 AW00011007 Marco Mehta Warrambool Victoria Australia 3280

10 11008 AW00011008 Rob Verhoff Bendigo Victoria Australia 3550

11 11009 AW00011009 Shannon Carlson Hervey Bay Queensland Australia 4655

12 11010 AW00011010 Jacquelyn Suarez East Brisbane Queensland Australia 4169

13 11011 AW00011011 Curtis Lu East Brisbane Queensland Australia 4169

14 11012 AW00011012 Lauren Walker Fremington Washington United States 98312

15 11013 AW00011013 Ian Jenkins Lebanon Oregon United States 97355

16 11014 AW00011014 Sydney Bennett Redmond Washington United States 98052

17 11015 AW00011015 Chloe Young Burbank California United States 91502

18 11016 AW00011016 Wyatt Hill Imperial Beach California United States 91952

19 11017 AW00011017 Shannon Wang Sunbury Victoria Australia 3429

20 11018 AW00011018 Clarence Rai Bendigo Victoria Australia 3550

21 11019 AW00011019 Luke Lal Langley British Columbia Canada V3A 4R2

22 11020 AW00011020 Jordan King Methow British Columbia Canada V9

23 11021 AW00011021 Destiny Wilson Beaverton Oregon United States 97005

24 11022 AW00011022 Ethan Zhang Bellingham Washington United States 98225

25 11023 AW00011023 Seth Edwards Bellflower California United States 90706

26 11024 AW00011024 Russell Xie Concord California United States 94519

27 11025 AW00011025 Alejandro Beck Hawthorne Queensland Australia 4171

28 11026 AW00011026 Harold Sali Goulburn New South Wales Australia 2580

29 11027 AW00011027 Jessie Zhao Warrambool Victoria Australia 3280

30 11028 AW00011028 Jill Jimenez St. Leonards New South Wales Australia 2065

7 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 1641

Query Settings

Transform Steps

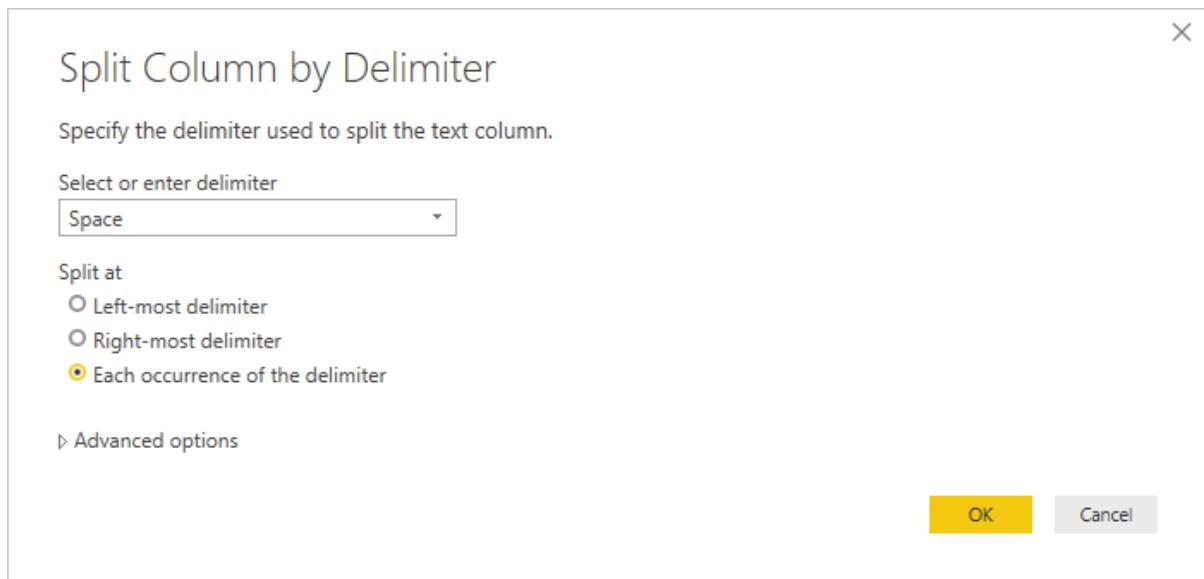
Remove Other Columns
Duplicate Column
Add Column From Examples...
Remove Duplicates
Remove Errors
Change Type
Transform
Replace Values...
Replace Errors...
Split Column
Group By...
Fill
Unpivot Columns
Unpivot Other Columns
Unpivot Only Selected Columns
Rename...
Move
 Drill Down
Add as New Query

Transformation 1:

Splitting customer column into FirstName and LastName

Customer	City	State-Province	Country-Region
[Not Applicable]			
Jon Yang	Rockhampton	Queensland	Australia
Eugene Huang	Seaford	Victoria	Australia
Ruben Torres	Hobart	Tasmania	Australia
Christy Zhu	North Ryde	New South Wales	Australia
Elizabeth Johnson	Wollongong	New South Wales	Australia
Julio Ruiz	East Brisbane	Queensland	Australia
Janet Alvarez	Matraville	New South Wales	Australia
Marco Mehta	Warrambool	Victoria	Australia
Rob Verhoff	Bendigo	Victoria	Australia
Shannon Carlson	Hervey Bay	Queensland	Australia
Jacquelyn Suarez	East Brisbane	Queensland	Australia
Curtis Lu	East Brisbane	Queensland	Australia
Lauren Walker	Fremington	Washington	United States
Ian Jenkins	Lebanon	Oregon	United States
Sydney Bennett	Redmond	Washington	United States
Chloe Young	Burbank	California	United States
Wyatt Hill	Imperial Beach	California	United States
Shannon Wang	Sunbury	Victoria	Australia
Clarence Rai	Bendigo	Victoria	Australia
Luke Lal	Langley	British Columbia	Canada V3A 4R2
Jordan King	Methow	British Columbia	Canada V9
Destiny Wilson	Beaverton	Oregon	United States 97005
Ethan Zhang	Bellingham	Washington	United States 98225
Seth Edwards	Bellflower	California	United States 90706
Russell Xie	Concord	California	United States 94519
Alejandro Beck	Hawthorne	Queensland	Australia 4171
Harold Sali	Goulburn	New South Wales	Australia 2580
Jessie Zhao	Warrambool	Victoria	Australia 3280
Jill Jimenez	St. Leonards	New South Wales	Australia 2065

Copy
Remove
Remove Other Columns
Duplicate Column
Add Column From Examples...
Remove Duplicates
Remove Errors
Change Type
Transform
Replace Values...
Replace Errors...
Split Column
Group By...
Fill
Unpivot Columns
Unpivot Other Columns
Unpivot Only Selected Columns
Rename...



FirstName	Customer	State-Province
[Not Applicable]	Yang	[Not Applicable]
Jon	Huang	Queensland
Eugene	Torres	Victoria
Ruben	Zhu	Tasmania
Christy	Johnson	New South Wales
Elizabeth	Ruiz	New South Wales
Julio	Alvarez	Queensland
Janet	Mehta	New South Wales
Marco	Verhoff	Victoria
Rob	Carlson	Victoria
Shannon	Suarez	Queensland
Jacquelyn	Lu	Queensland
Curtis	Walker	Queensland
Lauren	Jenkins	Washington
Ian	Bennett	Oregon
Sydney	Young	Washington
Chloe	Hill	California
Wyatt	Wang	California
Shannon	Rai	Victoria
Clarence	Lal	Victoria
Luke		British Columbia

A context menu is open over the 'Customer' column, showing options like Copy, Remove, Split Column, and Rename... The 'Rename...' option is highlighted.

Untitled - Power Query Editor

File Home Transform Add Column View Help

Close & Apply Close New Source Sources Enter Data Data source settings Data Sources Manage Parameters Refresh Preview Advanced Editor Properties Choose Columns Remove Columns Keep Rows Remove Rows Sort Split Group By Replace Values Data Type: Text Merge Queries Use First Row as Headers Append Queries Combine Files

Queries [1] Customer

	CustomerKey	CustomerID	FirstName	LastName	City	StateProvince	CountryRegion	PostalCode
1		-1 [Not Applicable]	[Not Applicable]					
2	11000	AW00011000	Jon	Yang	Rockhampton	Queensland	Australia	4700
3	11001	AW00011001	Eugene	Huang	Seaford	Victoria	Australia	3198
4	11002	AW00011002	Ruben	Torres	Hobart	Tasmania	Australia	7001
5	11003	AW00011003	Christy	Zhu	North Ryde	New South Wales	Australia	2113
6	11004	AW00011004	Elizabeth	Johnson	Wollongong	New South Wales	Australia	2500
7	11005	AW00011005	Julio	Ruiz	East Brisbane	Queensland	Australia	4169
8	11006	AW00011006	Janet	Alvarez	Matraville	New South Wales	Australia	2036
9	11007	AW00011007	Marco	Mehta	Warrnambool	Victoria	Australia	5280
10	11008	AW00011008	Rob	Vernhoff	Bendigo	Victoria	Australia	3550
11	11009	AW00011009	Shannon	Carlson	Hervey Bay	Queensland	Australia	4655
12	11010	AW00011010	Jacquelyn	Suarez	East Brisbane	Queensland	Australia	4169
13	11011	AW00011011	Curtis	Lu	East Brisbane	Queensland	Australia	4169
14	11012	AW00011012	Lauren	Walker	Bremerton	Washington	United States	98312
15	11013	AW00011013	Ian	Jenkins	Lebanon	Oregon	United States	97355
16	11014	AW00011014	Sydney	Bennett	Redmond	Washington	United States	98052
17	11015	AW00011015	Chloe	Young	Burbank	California	United States	91502
18	11016	AW00011016	Wyatt	Hill	Imperial Beach	California	United States	91952
19	11017	AW00011017	Shannon	Wang	Sunbury	Victoria	Australia	3429
20	11018	AW00011018	Clarence	Rai	Bendigo	Victoria	Australia	3550
21	11019	AW00011019	Luke	Lai	Langley	British Columbia	Canada	V3A 4R2
22	11020	AW00011020	Jordan	King	Methow	British Columbia	Canada	V9
23	11021	AW00011021	Destiny	Wilson	Beaverton	Oregon	United States	97005
24	11022	AW00011022	Ethan	Zhang	Bellingham	Washington	United States	98225
25	11023	AW00011023	Seth	Edwards	Bellflower	California	United States	90706
26	11024	AW00011024	Russell	Xie	Concord	California	United States	94519
27	11025	AW00011025	Alejandro	Beck	Hawthorne	Queensland	Australia	4171
28	11026	AW00011026	Harold	Sai	Goulburn	New South Wales	Australia	2580
29	11027	AW00011027	Jessie	Zhao	Warrnambool	Victoria	Australia	3280
30	11028	AW00011028	Ill	Leonard	Sr. Leonard	New South Wales	Australia	2065

8 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 1641

Transformation 2:

Replacing AW000 in Customer ID column with T20

CustomerKey CustomerID FirstName LastName City

Copy table [Not Applicable]
 Remove Rockhampton
 Remove Other Columns Seaford
 Duplicate Column Hobart
 Add Column From Examples... North Ryde
 Remove Duplicates Wollongong
 Remove Errors East Brisbane
 Change Type Matraville
 Transform Bendigo
 Replace Values... Hervey Bay
 Replace Errors... East Brisbane

Replace Values

Replace one value with another in the selected columns.

Value To Find

AW000

Replace With

T20

Advanced options

OK

Cancel

Customer

	CustomerKey	CustomerID	FirstName	LastName	City	StateProvince	CountryRegion	PostalCode
1	1	J [Not Applicable]	[Not Applicable]					
2	11000	T201000	Jon	Yang	Rockhampton	Queensland	Australia	4700
3	11001	T201001	Eugene	Huang	Seaford	Victoria	Australia	3198
4	11002	T201002	Ruben	Torres	Hobart	Tasmania	Australia	7001
5	11003	T201003	Christy	Zhu	North Ryde	New South Wales	Australia	2113
6	11004	T201004	Elizabeth	Johnson	Wollongong	New South Wales	Australia	2500
7	11005	T201005	Julio	Ruiz	East Brisbane	Queensland	Australia	4169
8	11006	T201006	Janet	Alvarez	Mattaville	New South Wales	Australia	2036
9	11007	T201007	Marco	Mehta	Warrnambool	Victoria	Australia	5280
10	11008	T201008	Rob	Vernhoff	Bendigo	Victoria	Australia	3550
11	11009	T201009	Shannon	Carlson	Hervey Bay	Queensland	Australia	4655
12	11010	T201010	Jacquelyn	Suarez	East Brisbane	Queensland	Australia	4169
13	11011	T201011	Curtis	Lu	East Brisbane	Queensland	Australia	4169
14	11012	T201012	Lauren	Walker	Bremerton	Washington	United States	98312
15	11013	T201013	Ian	Jenkins	Lebanon	Oregon	United States	97355
16	11014	T201014	Sydney	Bennett	Redmond	Washington	United States	98052
17	11015	T201015	Chloe	Young	Burbank	California	United States	91502
18	11016	T201016	Wyatt	Hill	Imperial Beach	California	United States	91952
19	11017	T201017	Shannon	Wang	Sunbury	Victoria	Australia	3429
20	11018	T201018	Clarence	Rai	Bendigo	Victoria	Australia	3550
21	11019	T201019	Luke	Lai	Langley	British Columbia	Canada	V3A 4R2
22	11020	T201020	Jordan	King	Methow	British Columbia	Canada	V9
23	11021	T201021	Destiny	Wilson	Beaverton	Oregon	United States	97005
24	11022	T201022	Ethan	Zhang	Bellingham	Washington	United States	98225
25	11023	T201023	Seth	Edwards	Bellflower	California	United States	90706
26	11024	T201024	Russell	Xie	Concord	California	United States	94519
27	11025	T201025	Alejandro	Beck	Hawthorne	Queensland	Australia	4171
28	11026	T201026	Harold	Sai	Goulburn	New South Wales	Australia	2580
29	11027	T201027	Jessie	Zhao	Warrnambool	Victoria	Australia	3280
30	11028	T201028	Ill	Imaneer	Sr Leonards	New South Wales	Australia	2065

Transformation 3:

Deleting first 100 records of all columns

Remove Top Rows

Specify how many rows to remove from the top.

Number of rows

100

OK Cancel

Screenshot of Power Query Editor showing a query named "Customer". The table has columns: CustomerKey, CustomerID, FirstName, LastName, City, StateProvince, CountryRegion, and PostalCode. The "CountryRegion" column is selected. The "Applied Steps" pane shows a step named "Removed Top Rows".

	CustomerKey	CustomerID	FirstName	LastName	City	StateProvince	CountryRegion	PostalCode
1	1109	T201109	Adam	Ross	Port Macquarie	New South Wales	Australia	2444
2	1110	T201100	Latache	Navarro	Cloverdale	South Australia	Australia	6105
3	1110	T201101	Abby	Sai	Crantbourne	Victoria	Australia	3977
4	1110	T201102	Julia	Nelson	Sydney	New South Wales	Australia	1002
5	1110	T201103	Cassie	Chande	Darlinghurst	New South Wales	Australia	2010
6	1110	T201104	Edgar	Sara	Gold Coast	Queensland	Australia	4217
7	1110	T201105	Candace	Fernandez	Mattaville	New South Wales	Australia	2036
8	1110	T201106	Jessie	Liu	Warrnambool	Victoria	Australia	3280
9	1110	T201107	Bianca	Lin	St Leonards	New South Wales	Australia	2065
10	1110	T201108	Kari	Alvarez	Port Macquarie	New South Wales	Australia	2444
11	1109	T201109	Ruben	Kapoor	South Melbourne	Victoria	Australia	3205
12	1110	T201110	Curtis	Yang	Darlinghurst	New South Wales	Australia	2010
13	1112	T201111	Meredith	Gutierrez	Geelong	Victoria	Australia	3220
14	1112	T201112	Crystal	Wang	Hervey Bay	Queensland	Australia	4655
15	1113	T201113	Michael	Bianco	Perth	South Australia	Australia	6006
16	1114	T201114	Leslie	Moreno	Rhodes	New South Wales	Australia	2138
17	1115	T201115	Alvin	Cal	St Leonards	New South Wales	Australia	2065
18	1116	T201116	Clinton	Carlson	Sydney	New South Wales	Australia	1002
19	1117	T201117	April	Deng	Mattaville	New South Wales	Australia	2036
20	1118	T201118	Alvin	Zeng	Lavender Bay	New South Wales	Australia	2060
21	1119	T201119	Evan	James	Port Macquarie	New South Wales	Australia	2444
22	1120	T201120	Beth	Jimenez	Sydney	New South Wales	Australia	1002
23	1121	T201121	Orlando	Suarez	Newcastle	New South Wales	Australia	2300
24	1122	T201122	Byron	Vazquez	North Sydney	New South Wales	Australia	2055
25	1124	T201123	Philip	Alvarez	Sunbury	Victoria	Australia	3429
26	1124	T201124	Ross	Jordan	Warrnambool	Victoria	Australia	3280
27	1125	T201125	Dana	Navarro	Seaford	Victoria	Australia	3198
28	1126	T201126	Shaun	Carson	North Ryde	New South Wales	Australia	2113
29	1127	T201127	Jan	Edwards	San Gabriel	California	United States	91776
30	1128	T201128	Samantha	Ione	Brentnn	Washington	United States	98055

Transformation 4:

Converting Country-Region column values to uppercase

Screenshot of Power Query Editor showing the "Country-Region" column selected. A context menu is open with the following options: Copy, Remove, Remove Other Columns, Duplicate Column, Add Column From Examples..., Remove Duplicates, Remove Errors, Change Type, Transform, Replace Values..., Replace Errors..., and Edit Column. The "Transform" option is highlighted. The "Applied Steps" pane shows a step named "Change Type" with "Country-Region" selected and "Type" set to "Text". A dropdown menu for "Text" type shows "UPPERCASE" selected.

Untitled - Power Query Editor

File **Home** **Transform** **Add Column** **View** **Help**

Queries [1] **CustomerKey** **Customer ID** **FirstName** **LastName** **City** **State-Province** **Country-Region** **Postal Code**

Customer

	CustomerKey	Customer ID	FirstName	LastName	City	State-Province	Country-Region	Postal Code
1	11099	T2011099	Adam	Ross	Port Macquarie	New South Wales	AUSTRALIA	2444
2	11100	T201100	Latache	Navarro	Cloverdale	South Australia	AUSTRALIA	6105
3	11101	T201101	Abby	Sai	Cronbourne	Victoria	AUSTRALIA	3977
4	11102	T201102	Julia	Nelson	Sydney	New South Wales	AUSTRALIA	1002
5	11103	T201103	Cassie	Chande	Darlinghurst	New South Wales	AUSTRALIA	2010
6	11104	T201104	Edgar	Sara	Gold Coast	Queensland	AUSTRALIA	4217
7	11105	T201105	Candace	Fernandez	Mattaville	New South Wales	AUSTRALIA	2036
8	11106	T201106	Jessie	Liu	Warrnambool	Victoria	AUSTRALIA	3280
9	11107	T201107	Bianca	Lin	St Leonards	New South Wales	AUSTRALIA	2065
10	11108	T201108	Kari	Alvarez	Port Macquarie	New South Wales	AUSTRALIA	2444
11	11109	T201109	Ruben	Kapoor	South Melbourne	Victoria	AUSTRALIA	3205
12	11110	T201110	Curtis	Yang	Darlinghurst	New South Wales	AUSTRALIA	2010
13	11111	T201111	Meredith	Gutierrez	Geelong	Victoria	AUSTRALIA	3220
14	11112	T201112	Crystal	Wang	Hervey Bay	Queensland	AUSTRALIA	4655
15	11113	T201113	Michael	Bianco	Perth	South Australia	AUSTRALIA	6006
16	11114	T201114	Leslie	Moreno	Rhodes	New South Wales	AUSTRALIA	2138
17	11115	T201115	Alvin	Cal	St Leonards	New South Wales	AUSTRALIA	2065
18	11116	T201116	Clinton	Carlson	Sydney	New South Wales	AUSTRALIA	1002
19	11117	T201117	April	Deng	Mattaville	New South Wales	AUSTRALIA	2036
20	11118	T201118	Alvin	Zeng	Lavender Bay	New South Wales	AUSTRALIA	2060
21	11119	T201119	Evan	James	Port Macquarie	New South Wales	AUSTRALIA	2444
22	11120	T201120	Beth	Jimenez	Sydney	New South Wales	AUSTRALIA	1002
23	11121	T201121	Orlando	Suarez	Newcastle	New South Wales	AUSTRALIA	2300
24	11122	T201122	Byron	Vazquez	North Sydney	New South Wales	AUSTRALIA	2055
25	11123	T201123	Philip	Alvarez	Sunbury	Victoria	AUSTRALIA	3429
26	11124	T201124	Ross	Jordan	Warrnambool	Victoria	AUSTRALIA	3280
27	11125	T201125	Dana	Navarro	Seaford	Victoria	AUSTRALIA	3198
28	11126	T201126	Shaun	Carson	North Ryde	New South Wales	AUSTRALIA	2113
29	11127	T201127	Jan	Edwards	San Gabriel	California	UNITED STATES	91776
30	11128	T201128	Samantha	Ione	Renton	Washington	UNITED STATES	98055

8 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED AT 1641

Finally click on "close and apply" to import

Untitled - Power Query Editor

File **Home** **Transform** **Add Column** **View** **Help**

Close & Apply **New Source** **Recent Sources** **Enter Data** **Data source settings** **Manage Parameters** **Ref Prev**

Close **New Query** **Data Sources** **Parameters**

Queries [1] **CustomerKey**

Customer

City	LastName	Postal Code	Country-Region	Customer ID	CustomerKey	FirstName
Ballard	Adams	98107	UNITED STATES	T2016877	1	Katelyn
Ballard	Allen	98107	UNITED STATES	T2019663	1	Alex
Ballard	Baker	98107	UNITED STATES	T2029036	1	Rebecca
Ballard	Brown	98107	UNITED STATES	T2027036	1	Hunter
Ballard	Cai	98107	UNITED STATES	T2025898	1	Brittney
Ballard	Chande	98107	UNITED STATES	T2016229	1	Wayne
Ballard	Clark	98107	UNITED STATES	T2028255	1	Robert
Ballard	Cox	98107	UNITED STATES	T2012366	1	Stephanie
Ballard	Cox	98107	UNITED STATES	T2027730	1	Isabella
Ballard	Davis	98107	UNITED STATES	T2014404	1	Rachel
Ballard	Diaz	98107	UNITED STATES	T2011785	1	Theodore
Ballard	Flores	98107	UNITED STATES	T2023193	1	Alexis
Ballard	Gill	98107	UNITED STATES	T2023246	1	Renee
Ballard	Green	98107	UNITED STATES	T2027548	1	Mason
Ballard	Hall	98107	UNITED STATES	T2021296	1	Luis
Ballard	Hayes	98107	UNITED STATES	T2015512	1	Jose
Ballard	He	98107	UNITED STATES	T2022741	1	Darryl
Ballard	Hernandez	98107	UNITED STATES	T2011144	1	Edward
Ballard	Hernandez	98107	UNITED STATES	T2016022	1	Devin
Ballard	Hernandez	98107	UNITED STATES	T2018813	1	Kristy
Ballard	Hernandez	98107	UNITED STATES	T2018887	1	Kari
Ballard	Hill	98107	UNITED STATES	T2013926	1	Dalton
Ballard	Hu	98107	UNITED STATES	T2023381	1	Bianca
Total					18385	

Filters on this visual

- City is (All)
- Country-Region is (All)
- Customer ID is (All)
- CustomerKey is (All)
- FirstName is (All)
- LastName is (All)
- Postal Code is (All)
- Country-Region is (All)
- Customer ID is (All)
- CustomerKey is (All)
- FirstName is (All)

Values

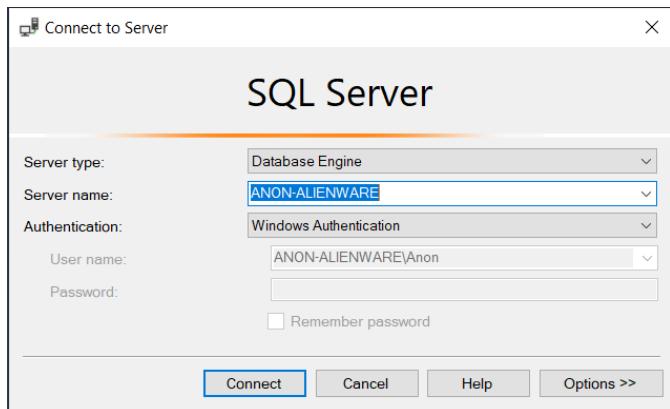
- City
- LastName
- Postal Code
- Country-Region
- Customer ID
- CustomerKey
- FirstName

Drillthrough

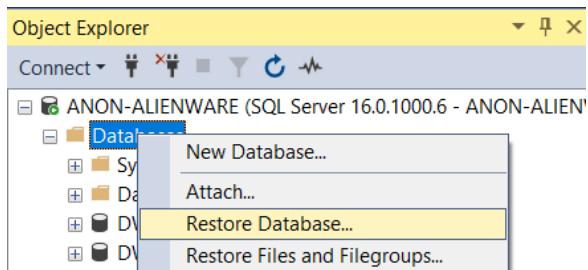
Cross-report

In SQL Server Data Tools.

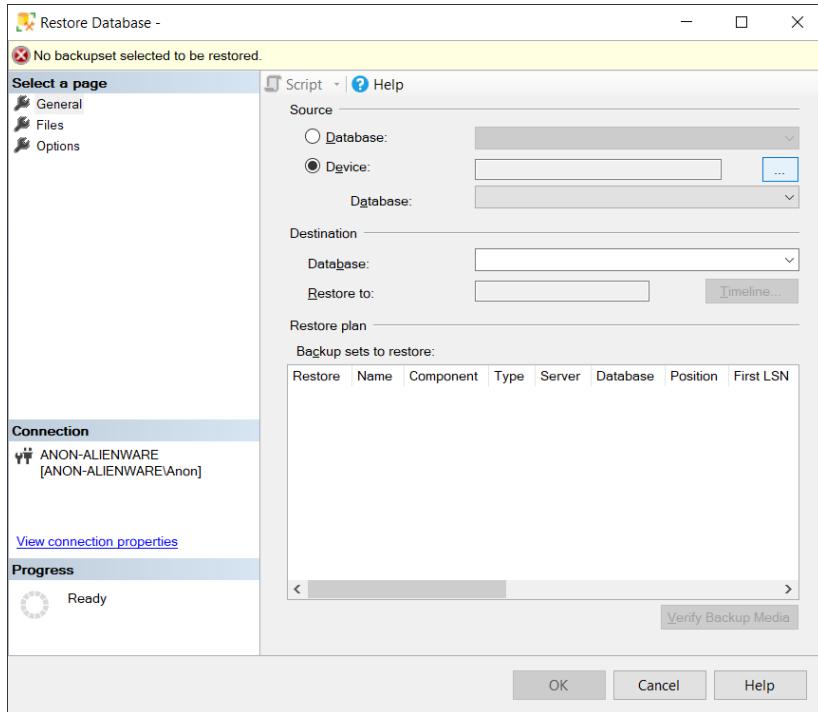
Step 1: Open SQL server management studio to restore backup file.

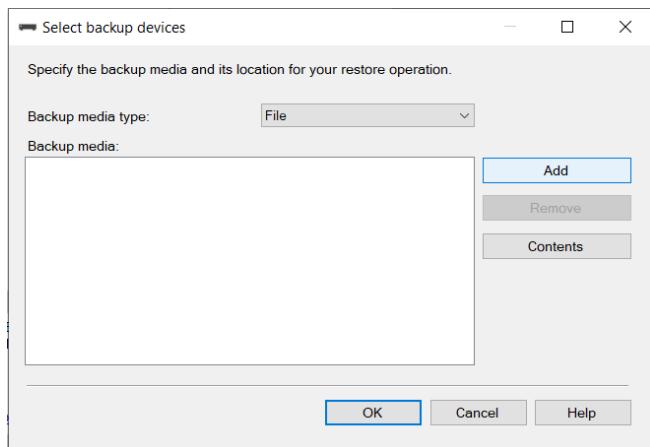


Step 2: Right click on the databases -> restore database

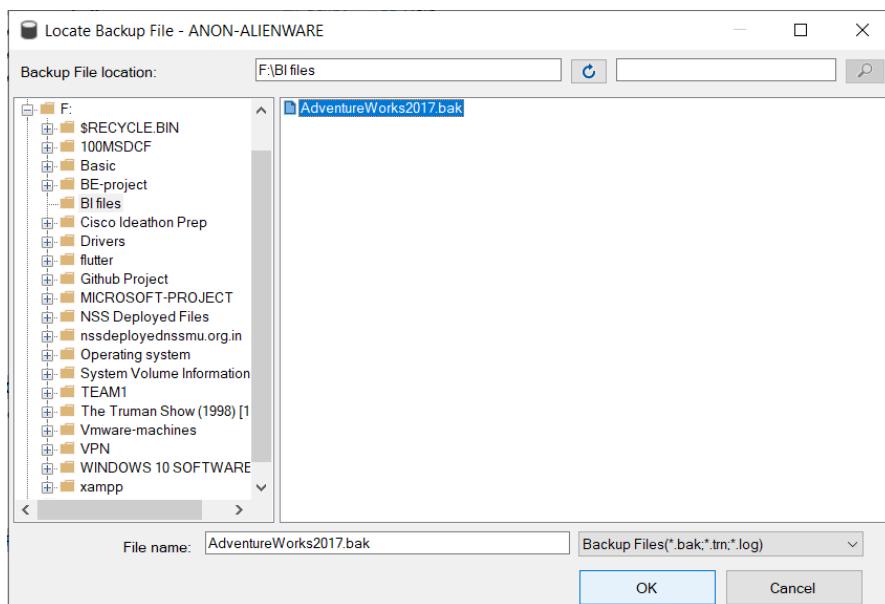


Step 3 : Select device -> click on 3 dots icon towards the end of device box

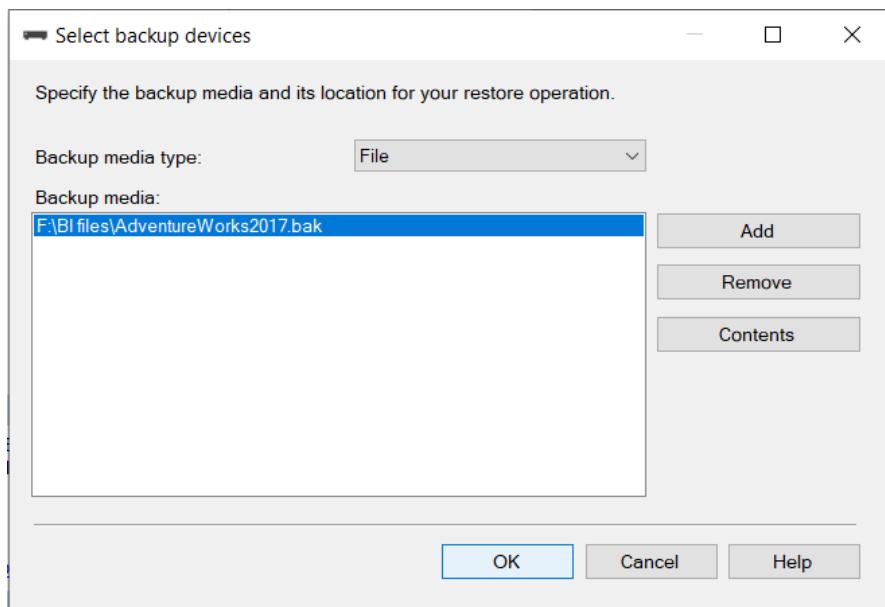




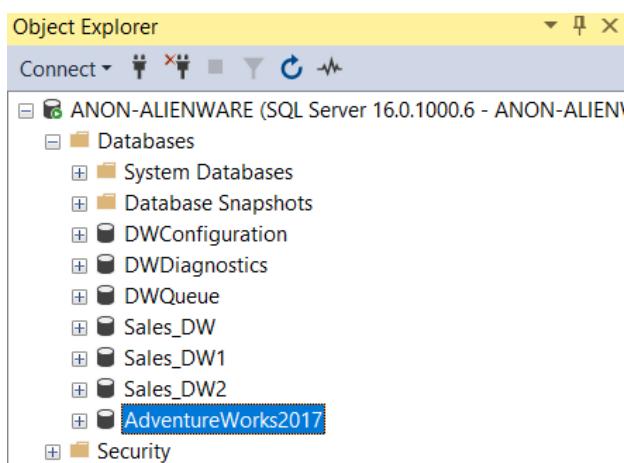
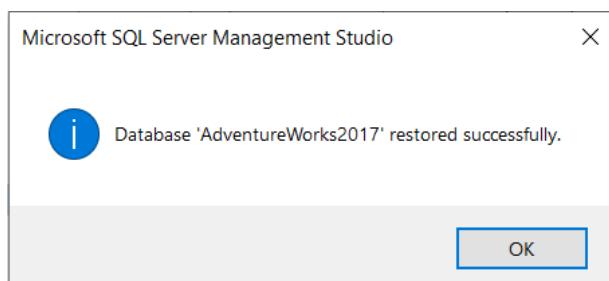
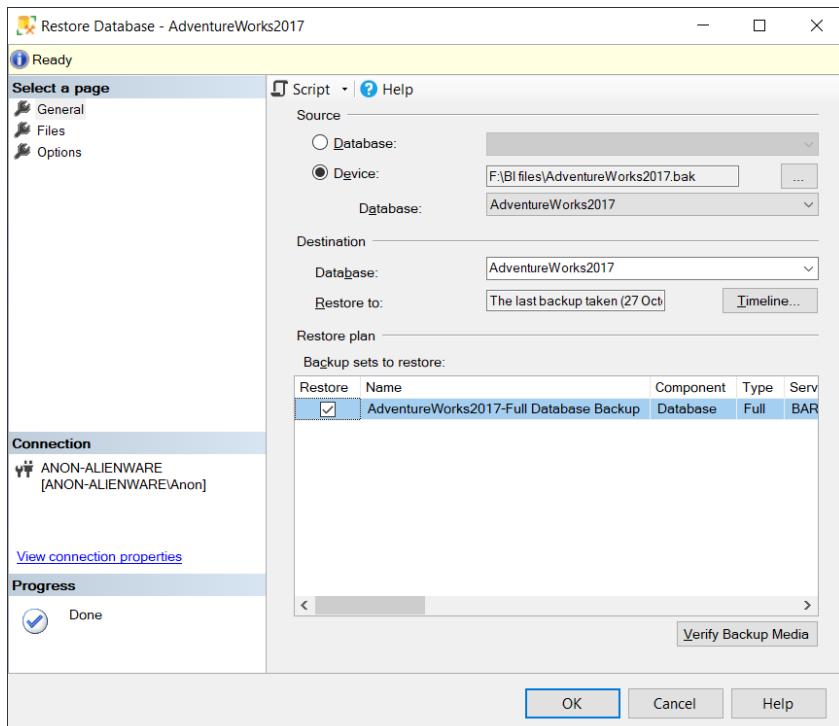
Step 4: Click on Add -> the path of backup file



Step 5: Select the file



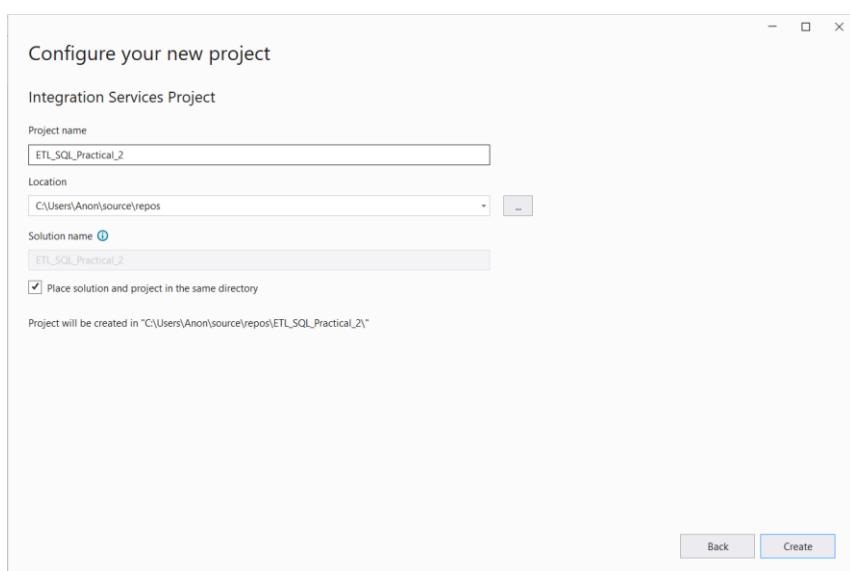
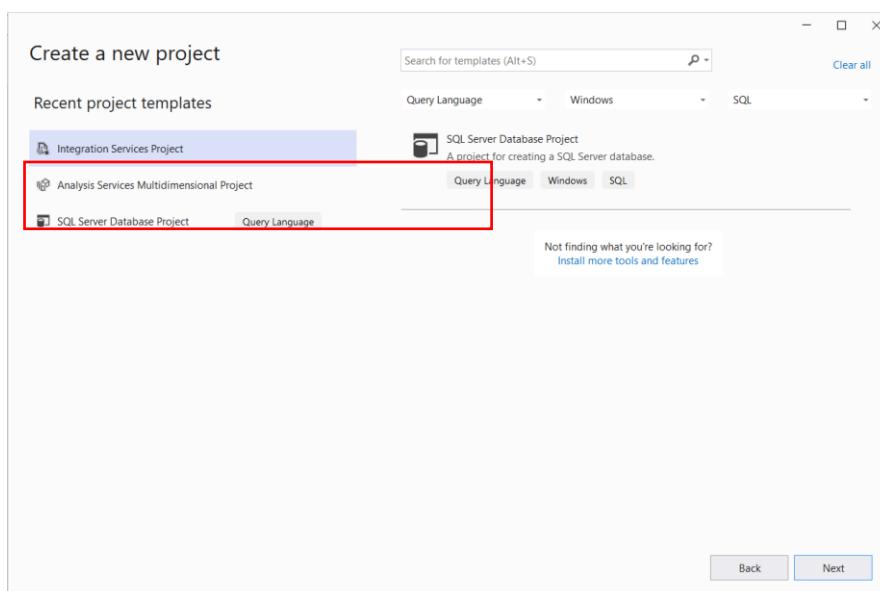
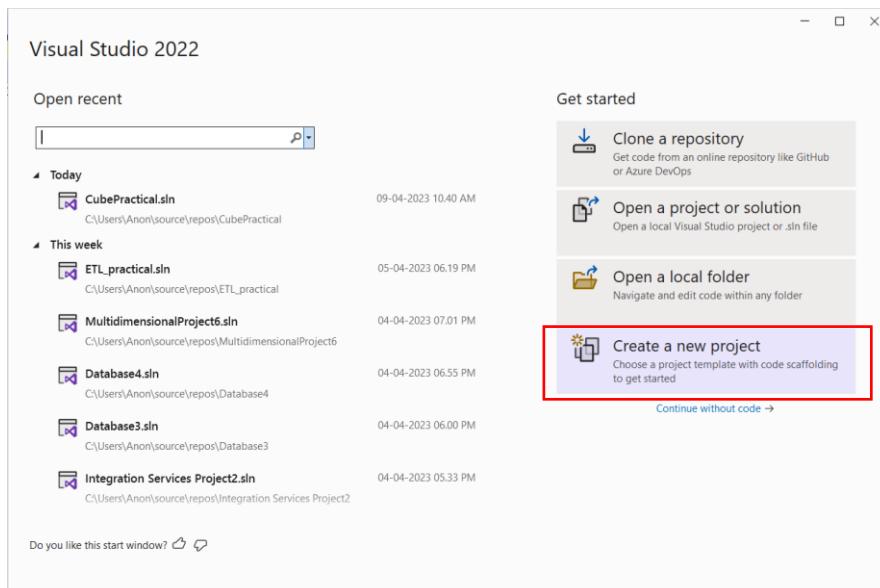
Step 6: Click on OK



The database is restored successfully.

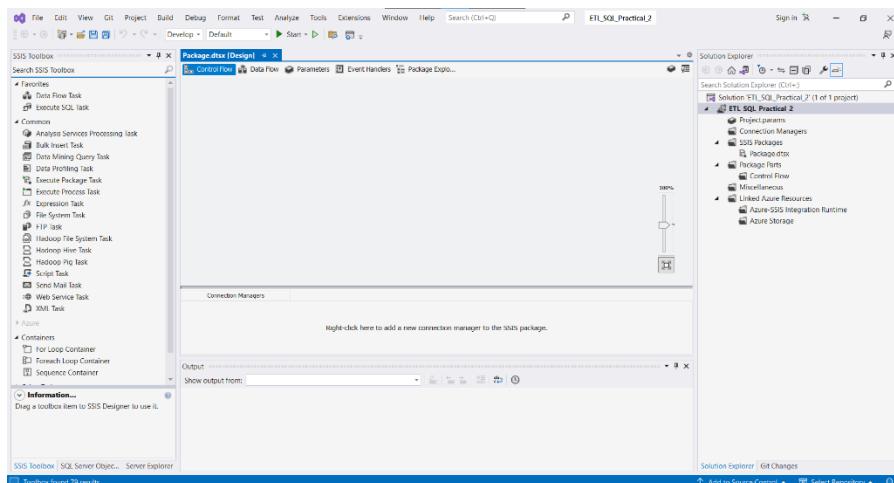
Step 7: Open SQL server data tools

Select File -> new -> project -> Business Intelligence -> **Integration services project** & give project name

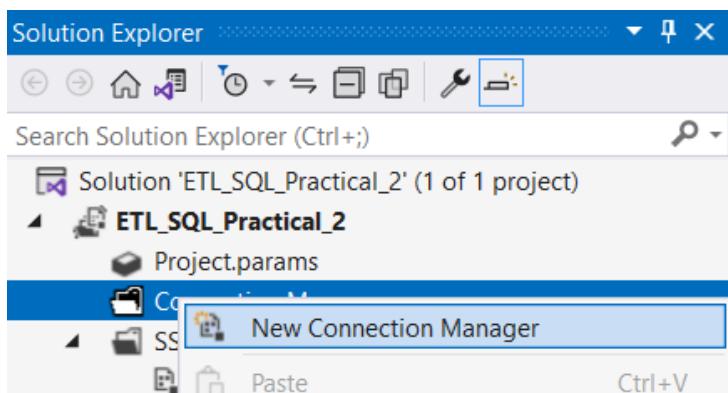


Environment consist of SQL Server Integration Services (SSIS)

Swapnil Ramgire T.20.83

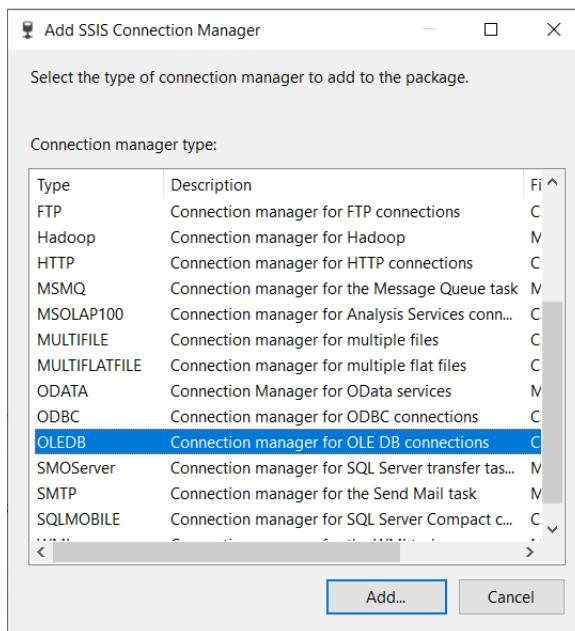


Step 8: Right click on connection manager in solution explorer and click on new connection manager

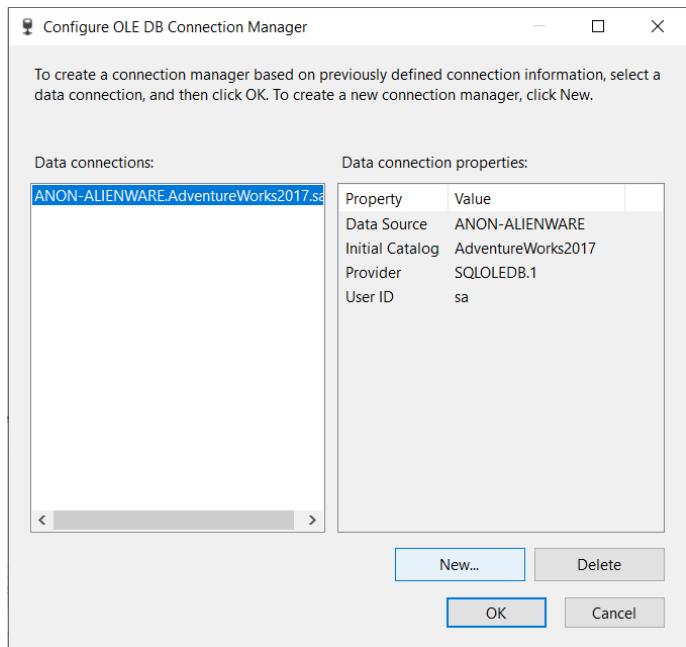


Add SSIS connection manager window appears

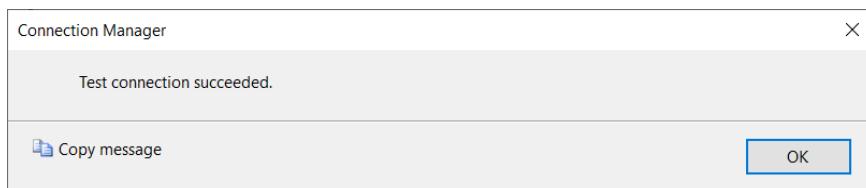
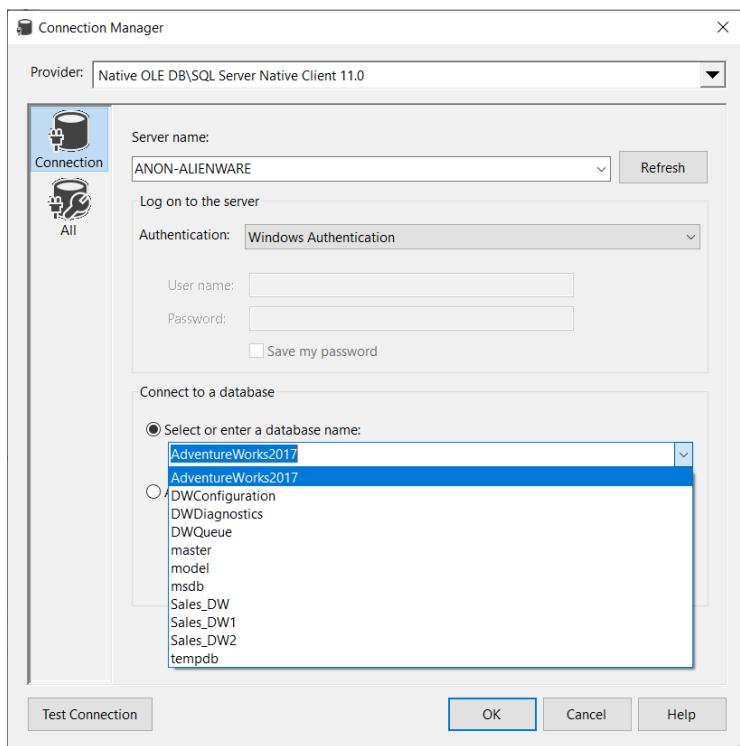
Step 9: Select OLEDB Connection Manager and Click on Add



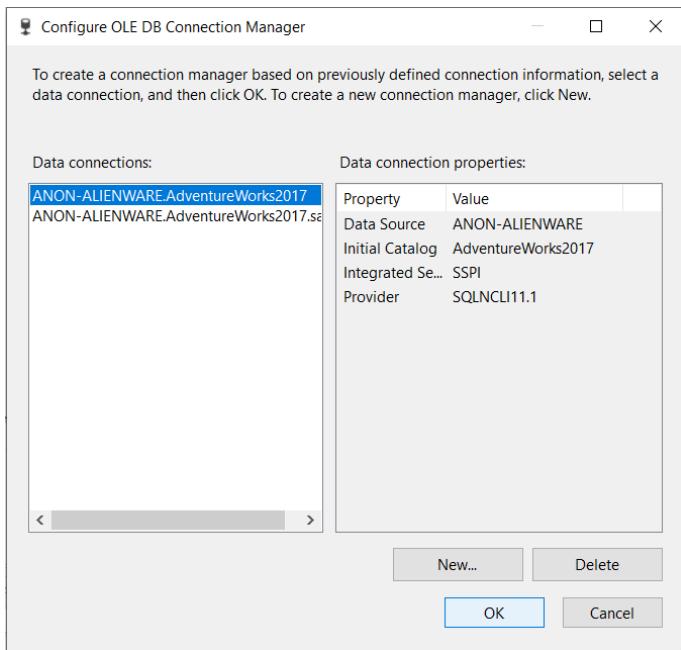
Step 10: Configure OLE DB Connection Manager window appears -> click on new



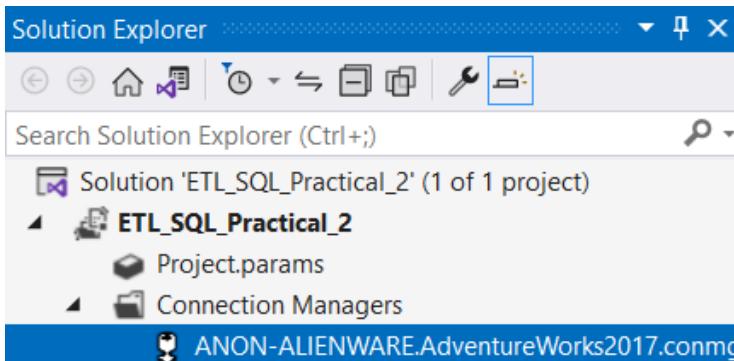
Step 11: Select Server name from drop down and database name and click on test connection



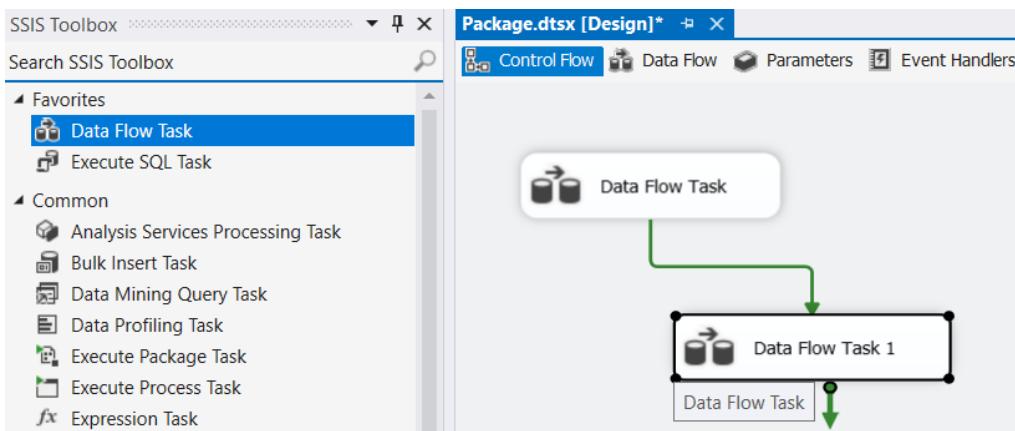
Step 12: Click on OK



Connection is added in connection manager

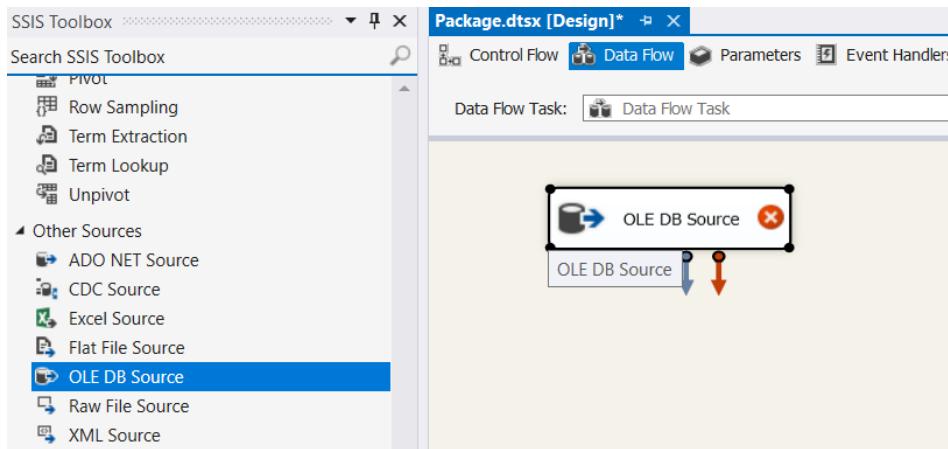


Step 13: Drag and drop Data Flow Task in Control Flow Tab

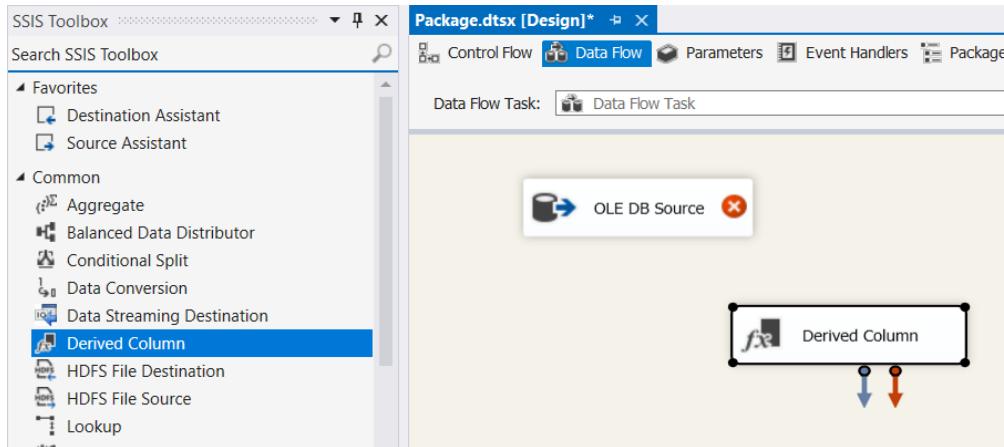


Double click on data flow task

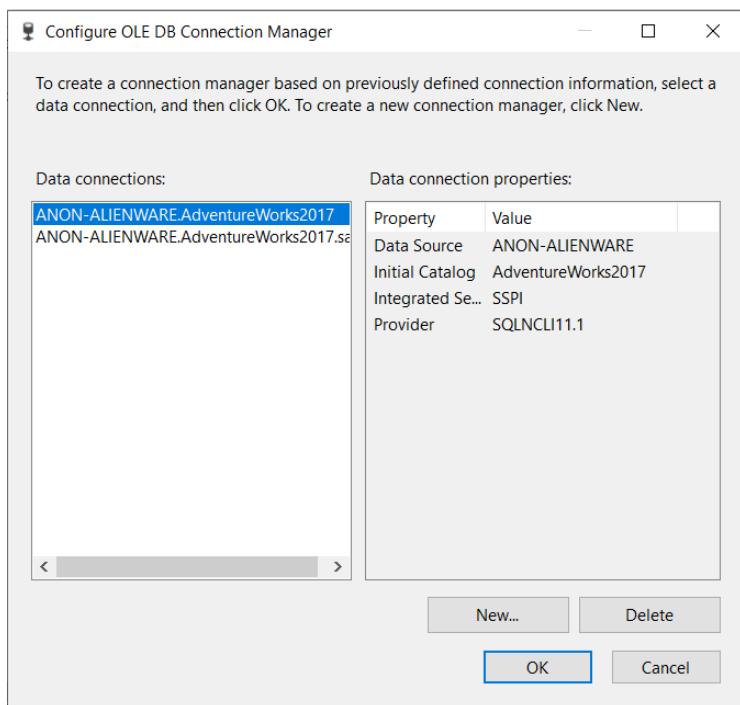
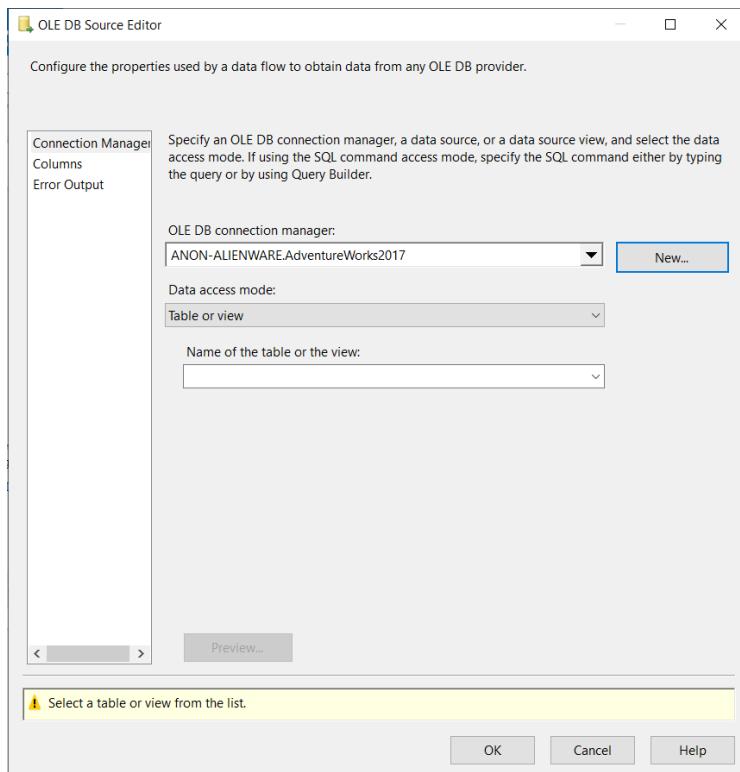
Step 14: Drag and drop OLE DB Source from other source into Data Flow tab



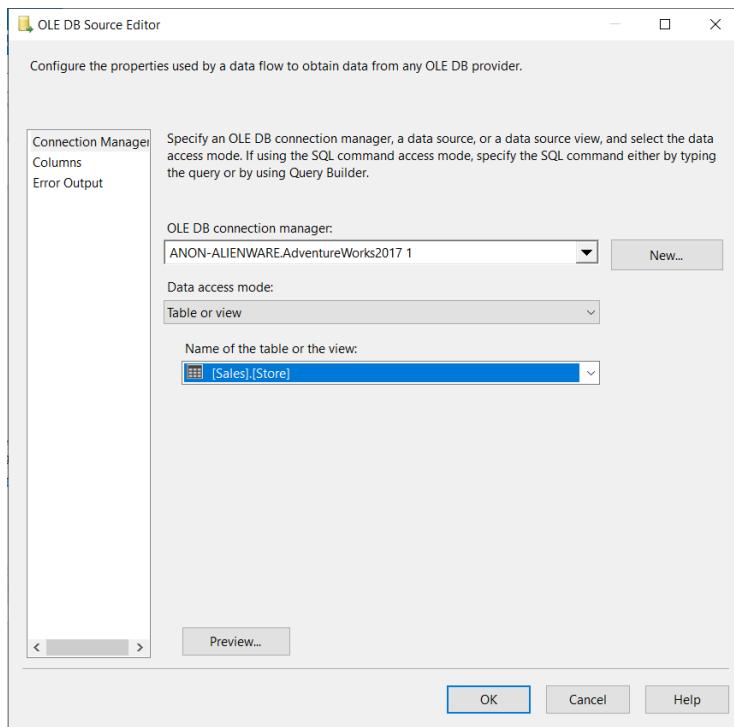
Add derived column from SSIS toolbox -> common



Step 15: Double click on OLE DB Source -> OLE DB source editor appears -> click on new to add connection manager



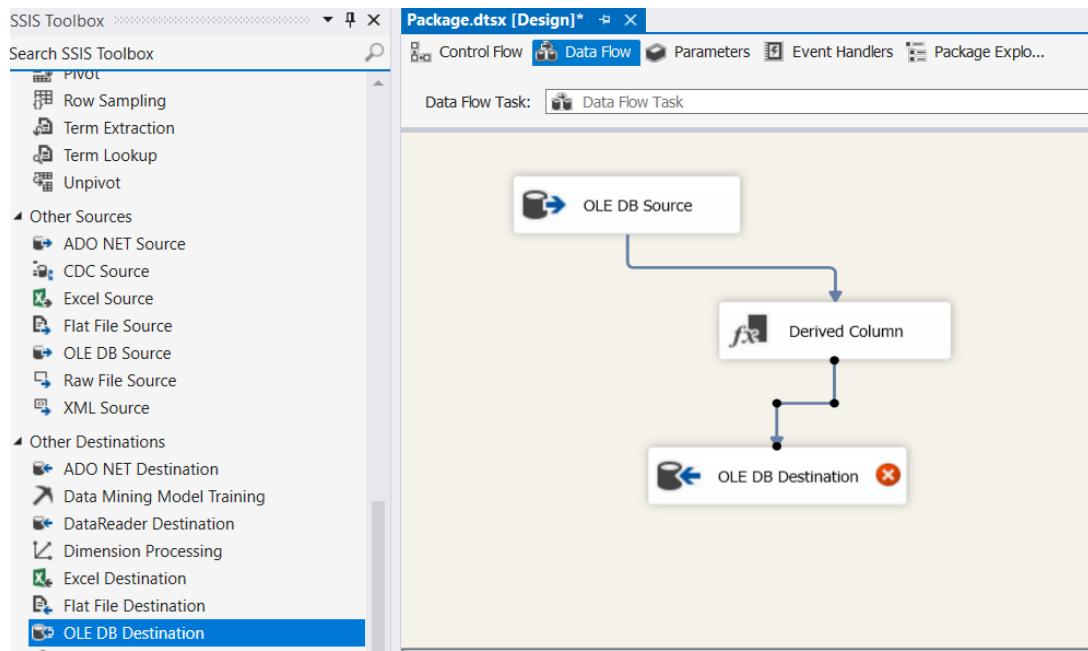
Select [Sales].[Store] table from drop down -> OK



Step 16: Connect OLE DB Source to derived column and Double click on derived column and add the expressions of ETL

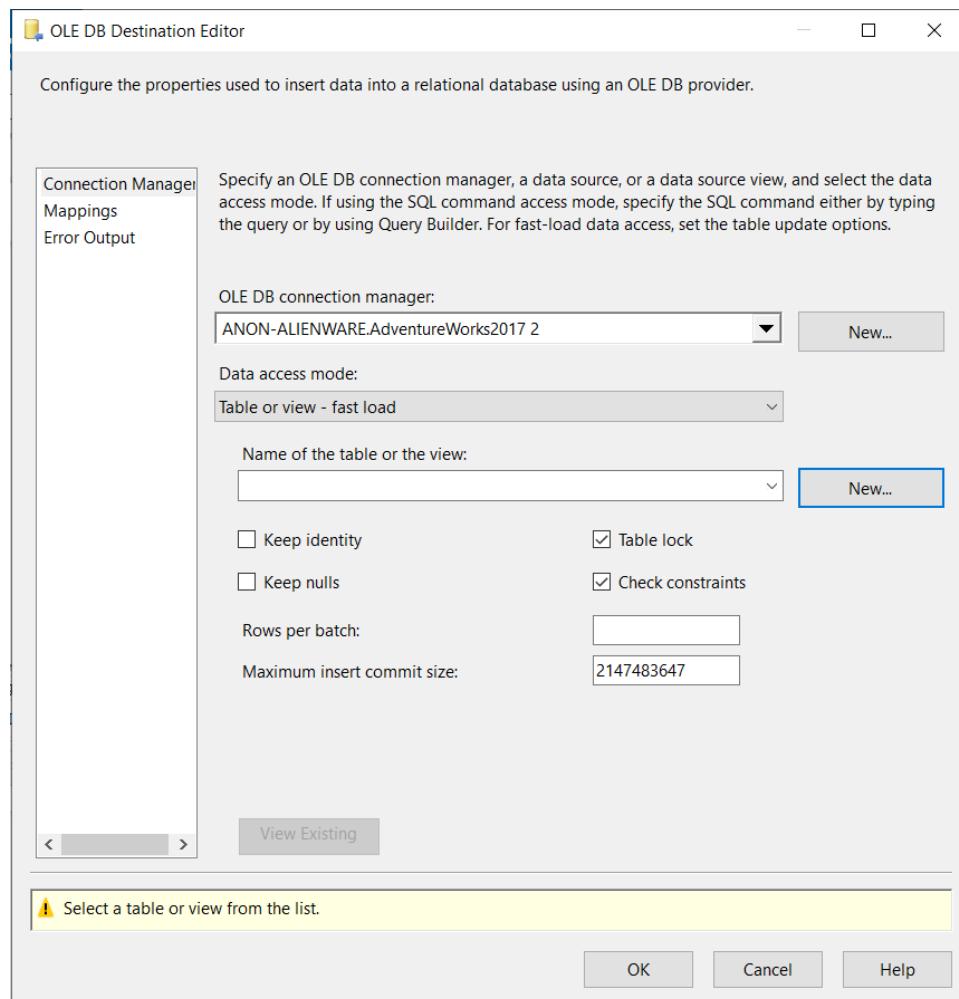
Derived Column Name	Derived Column	Expression	Data Type	Length
Name	Replace 'Name'	UPPER([Name])	Unicode string [DT_WSTR]	5
BusinessEntityID	Replace 'BusinessEnti...'.	SQUARE([BusinessEntityID])	four-byte signed integer [DT_I4]	
SalesPersonID	Replace 'SalesPersonl...'.	[SalesPersonID] * 4	four-byte signed integer [DT_I4]	
ModifiedDate	Replace 'ModifiedDa...'.	MONTH([ModifiedDate])	database timestamp [DT_DBTIMESTAMP]	

Step 17: Add OLE DB Destination in data flow and connect the derived column to it

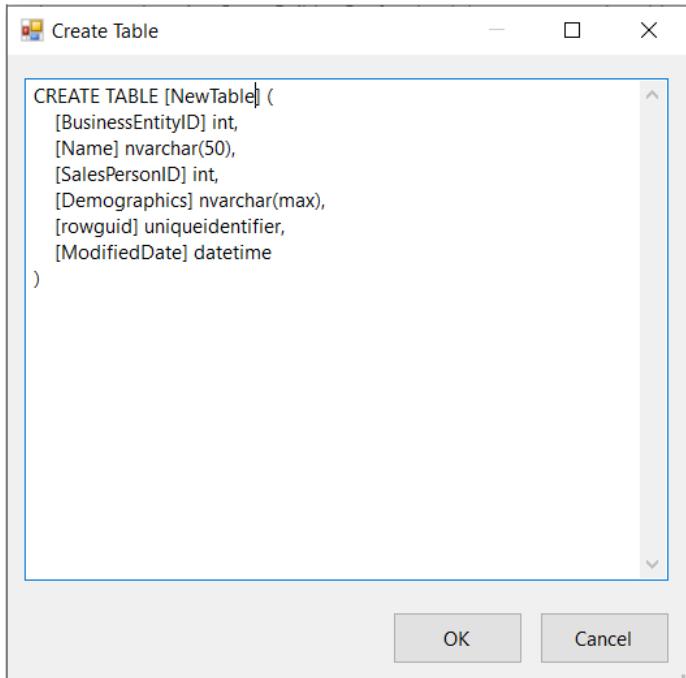


Step 18: Double click on OLE DB Destination

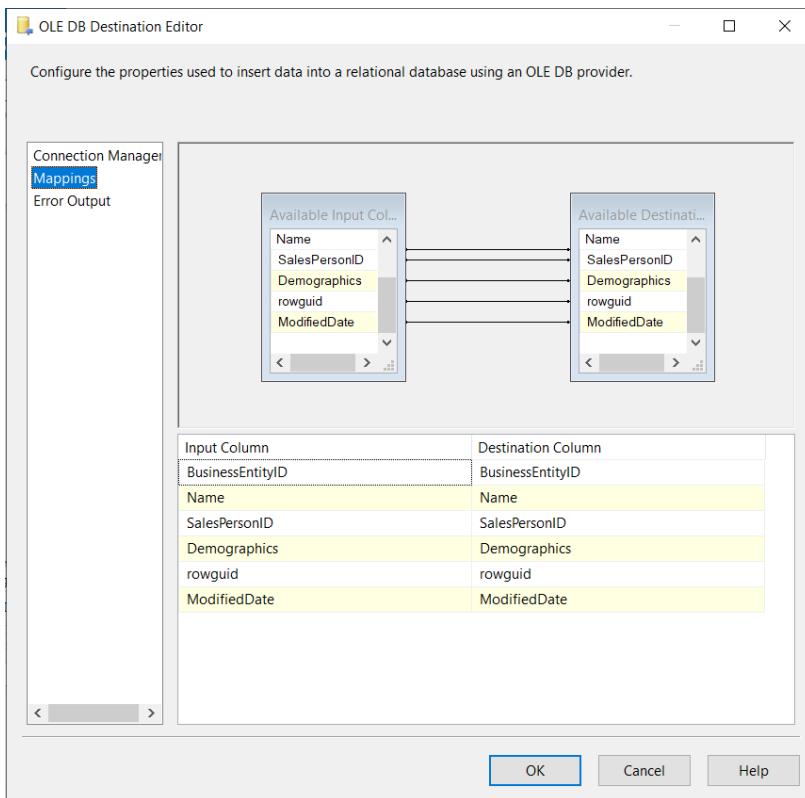
Click on new to run the query to get [OLE DB DESTINATION] in name of the table or the view

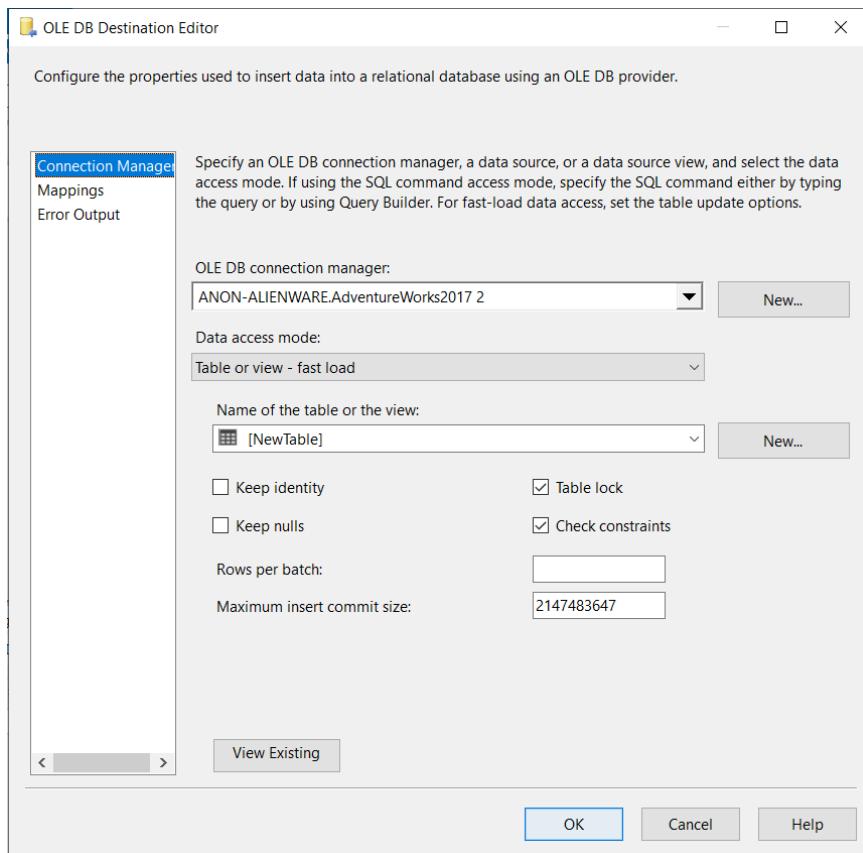


Change the name of table

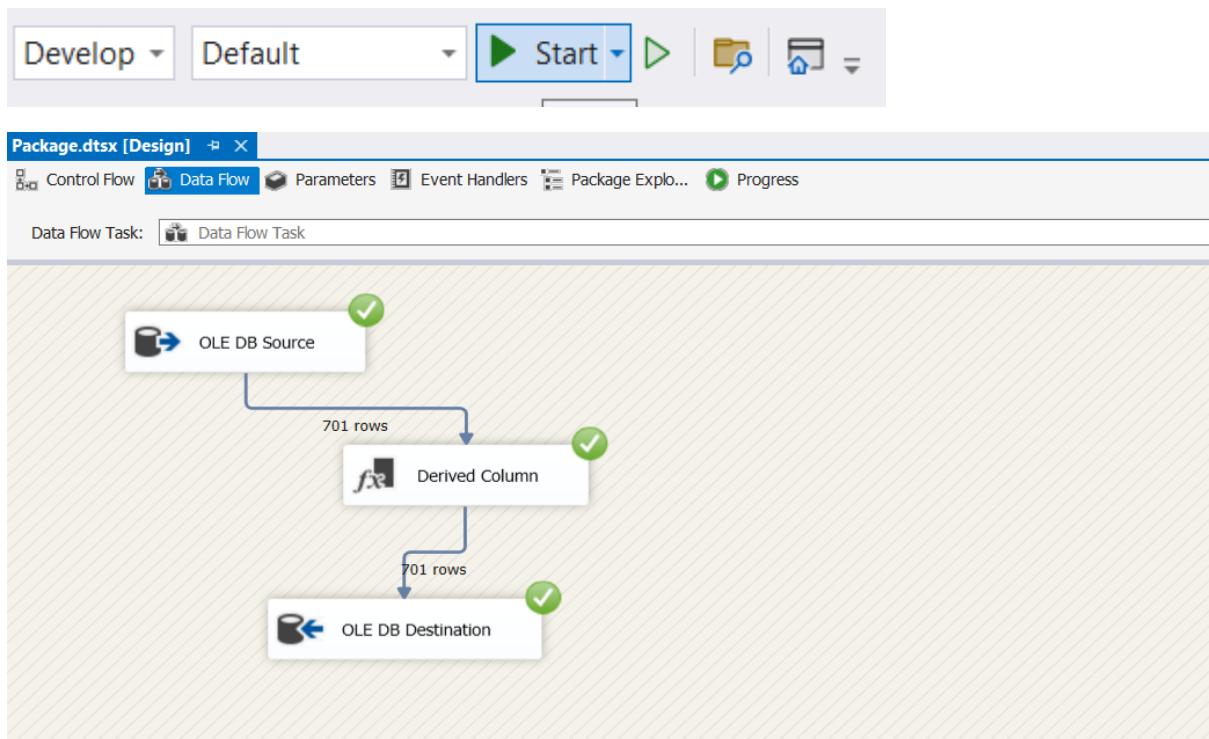


Map the columns

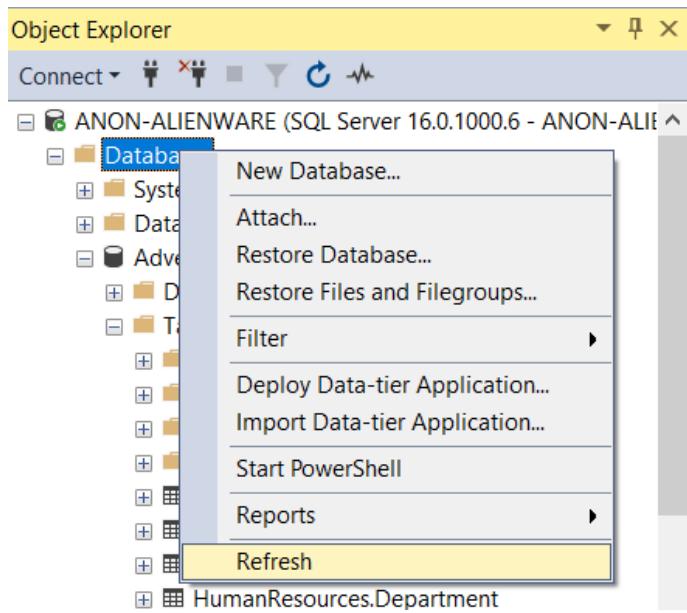




Click on Start button

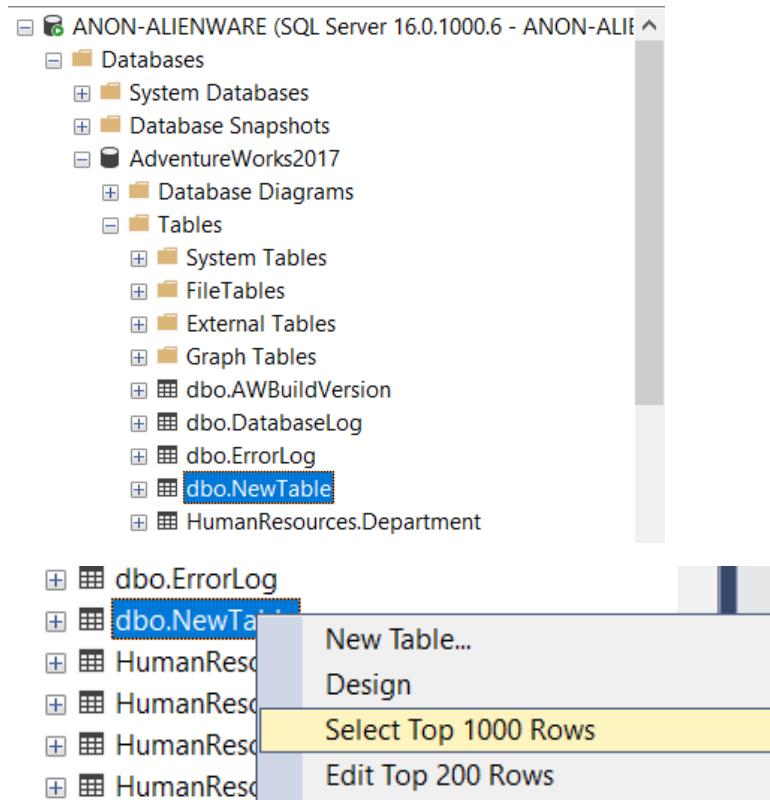


Refresh the Database



New table will be added to database

Right click on new table -> select top 100 rows



Final Output

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the session is for 'ANON-ALIENWARE.AdventureWorks2017 (ANON-ALIENWARE\Anon (60)) - Microsoft SQL Server Management Studio'. The Object Explorer pane on the left lists various database objects like Tables, System Tables, FileTables, External Tables, Graph Tables, and specific tables such as 'dbo.NewTable' and 'dbo.NewTable1'. The main Results pane displays the output of a T-SQL script:

```
***** Script for SelectTopNRows command from SSMS *****
SELECT TOP (1000) [BusinessEntityID]
      ,[Name]
      ,[SalesPersonID]
      ,[Demographics]
      ,[rowguid]
      ,[ModifiedDate]
  FROM [AdventureWorks2017].[dbo].[NewTable]
```

The results grid shows 701 rows of data from the 'NewTable' table. The columns include BusinessEntityID, Name, SalesPersonID, Demographics, rowguid, and ModifiedDate. The first few rows of data are as follows:

BusinessEntityID	Name	SalesPersonID	Demographics	rowguid	ModifiedDate
1	NEXT-DOOR BIKE STORE	1104	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	A2251F3E-348D-4EBE-B909-743773432304	1900-01-08 00:00:00.000
2	PROFESSIONAL SALES AND SERVICE	1104	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	B50CA50B-C001-4A13-B07E-2C386291B4	1900-01-08 00:00:00.000
3	RECYCLE IT	1106	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	3377A880-9A80-4850-90E9-000000000000	1900-01-08 00:00:00.000
4	THE BIKE MECHANICS	1100	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	7804F778-F032-4D16-BD75-213D8F13023	1900-01-08 00:00:00.000
5	NATIONWIDE SUPPLY	1144	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	C3F09705-A001-473A-9950-E374AB7804D	1900-01-08 00:00:00.000
6	AREA BIKE ACCESSORIES	1124	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	3638E600-30E5-49B9-9486-71F0409C594E	1900-01-08 00:00:00.000
7	BICYCLE ACCESSORIES AND KITS	1132	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	35F40E36-5105-49D5-88E6-27E2311891D0	1900-01-08 00:00:00.000
8	CLAMPS & BRACKETS CO.	1100	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	64D069FC-D090-405C-8C60-C067FE7067DF	1900-01-08 00:00:00.000
9	VALLEY BICYCLE SPECIALISTS	1108	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	5933680C-0522-4668-8448-4E1711793320	1900-01-08 00:00:00.000
10	NEW BIKES COMPANY	1116	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	47E4B9BD-5CD1-45A3-A231-790930381C56	1900-01-08 00:00:00.000
11	VINYL AND PLASTIC GOODS CORPORATION	1128	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	D0C10525-E373-40B1-8786-EA040E0C5C06	1900-01-08 00:00:00.000
12	TOP OF THE LINE BIKES	1152	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	E290E93F-A903-4B43-86C3-985F15C8AE	1900-01-08 00:00:00.000
13	FUN TOYS AND BIKES	1124	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	6CCCF941-4192-49C7-9944-5ADBA534E095	1900-01-08 00:00:00.000
14	GREAT BIKES	1132	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	9596FB35-5ED0-4175-8045-E0BE380BA340	1900-01-08 00:00:00.000
15	METROPOLITAN SALES AND RENTAL	1100	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	0CB4FE2-5047-40F7-8848-B897A3C3E3EC	1900-01-08 00:00:00.000
16	IRREGULARS OUTLET	1152	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	CDE66279-83D8-3430-A33C-E86E15914AC4	1900-01-08 00:00:00.000
17	VALLEY TOY STORE	1128	<StoreSurvey xmlns='http://schemas.microsoft.com/2009/08/StoreSurvey'>	6A1BEA56-DCB7-450F-8C92-3705E12EB2A2	1900-01-08 00:00:00.000

At the bottom of the Results pane, it says 'Query executed successfully.' and shows the status bar with 'Ready' and 'ANON-ALIENWARE (16.0 RTM) ANON-ALIENWARE\Anon (60) AdventureWorks2017 00:00:00 701 rows'.

Practical 3

AIM: Create the cube with suitable dimensions and flat table olap/holap

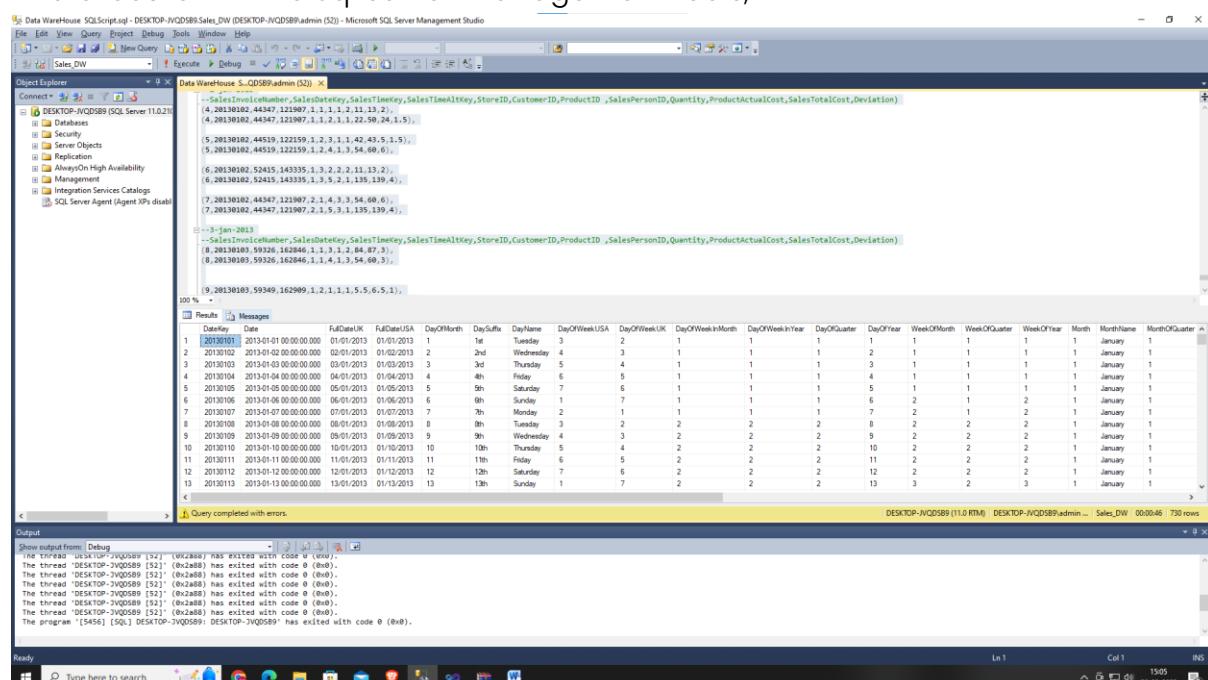
Solution:

Steps

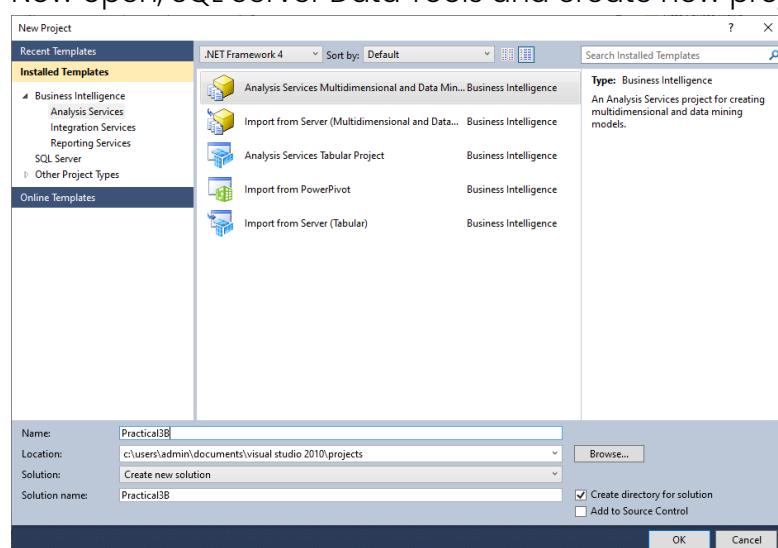
Download the script file:

https://www.codeproject.com/KB/database/652108/Data_Warehouse_SQLScript.zip

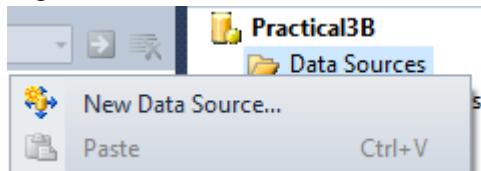
And execute it in the Sql server management tools;



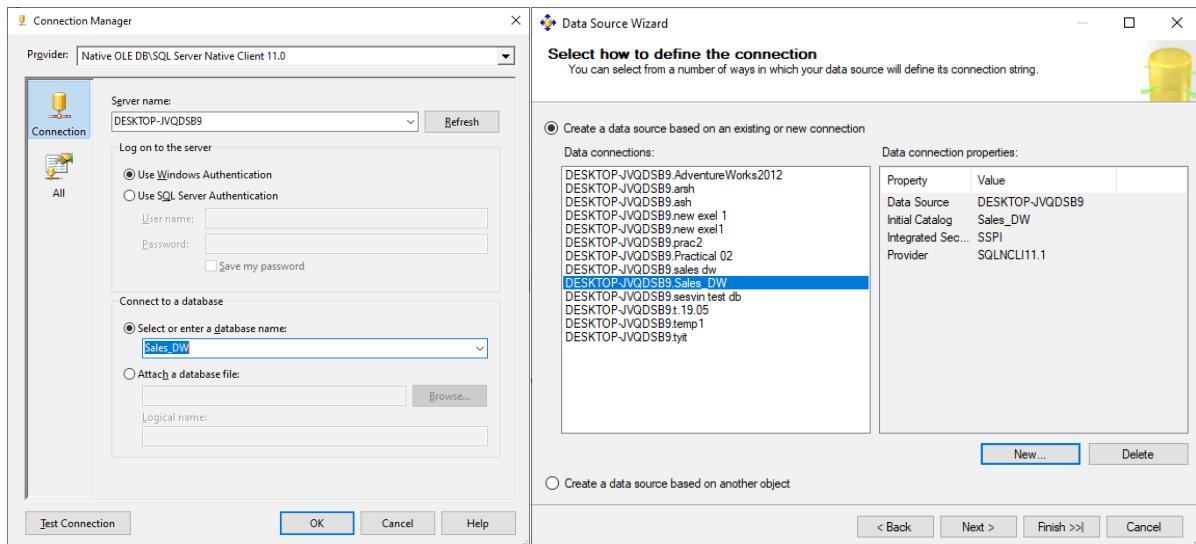
Now open, SQL Server Data Tools and create new project



Right click on data sources and click add new Data Source



And click on New.. to add database



Provider: Native OLE DB\SQL Server Native Client 11.0

Server name: DESKTOP-JVQDSB9

Log on to the server:

- Use Windows Authentication (selected)
- Use SQL Server Authentication

User name: [empty]

Password: [empty]

Save my password:

Connect to a database:

- Select or enter a database name: SalesDW
- Attach a database file:

Logical name: [empty]

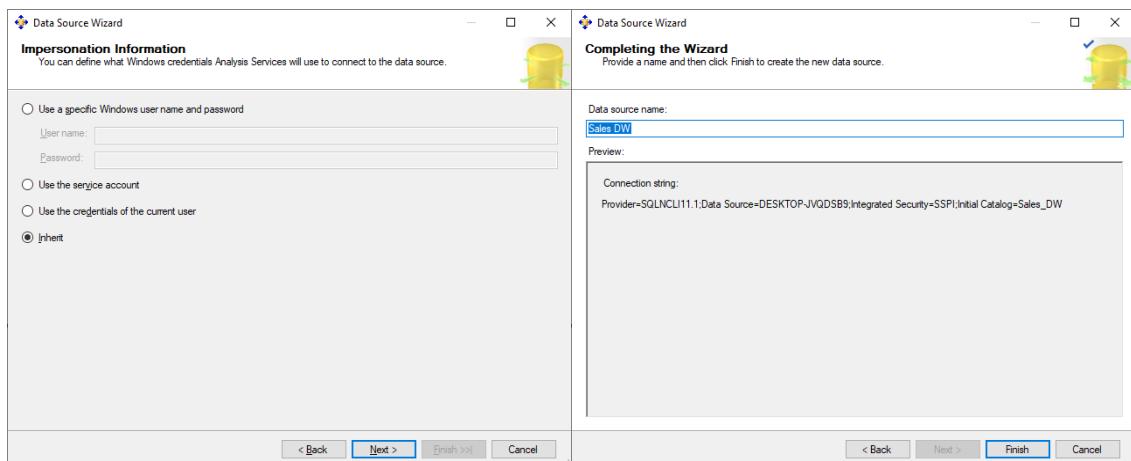
Data connections:

- DESKTOP-JVQDSB9.AdventureWorks2012
- DESKTOP-JVQDSB9.ash
- DESKTOP-JVQDSB9.ash
- DESKTOP-JVQDSB9.new excel
- DESKTOP-JVQDSB9.new excel
- DESKTOP-JVQDSB9.prac2
- DESKTOP-JVQDSB9.Practical 02
- DESKTOP-JVQDSB9.sales dw
- DESKTOP-JVQDSB9.Sales DW**
- DESKTOP-JVQDSB9.sevin test db
- DESKTOP-JVQDSB9t.19.05
- DESKTOP-JVQDSB9.temp 1
- DESKTOP-JVQDSB9.tyt

Data connection properties:

Property	Value
Data Source	DESKTOP-JVQDSB9
Initial Catalog	Sales_DW
Integrated Sec...	SSPI
Provider	SQLNCLI11.1

Select inherit



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

Use the service account

Use the credentials of the current user

Inherit (selected)

Completing the Wizard

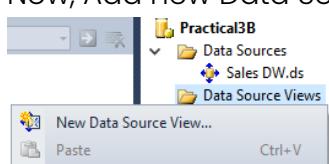
Provide a name and then click Finish to create the new data source.

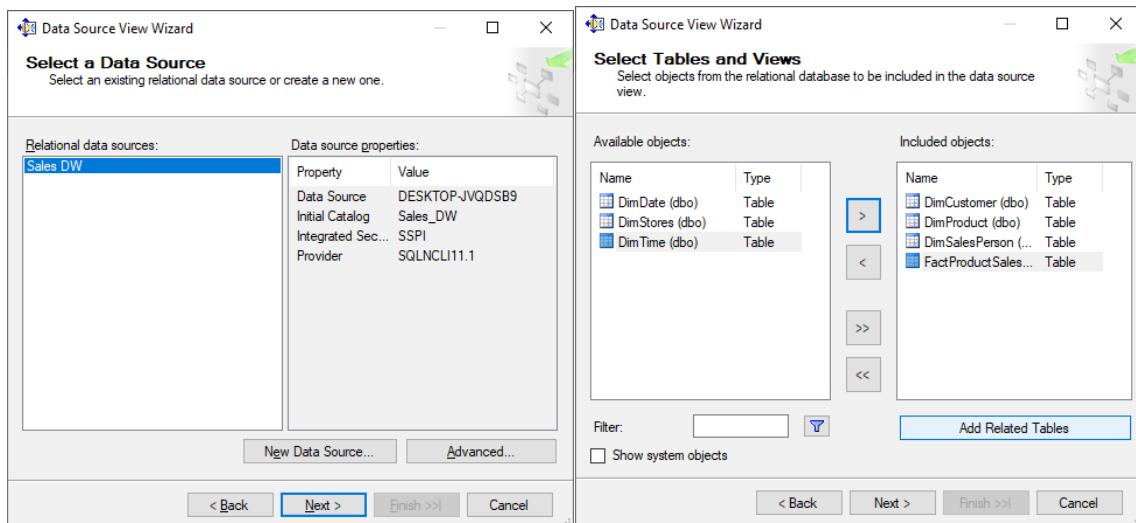
Data source name: Sales DW

Preview:

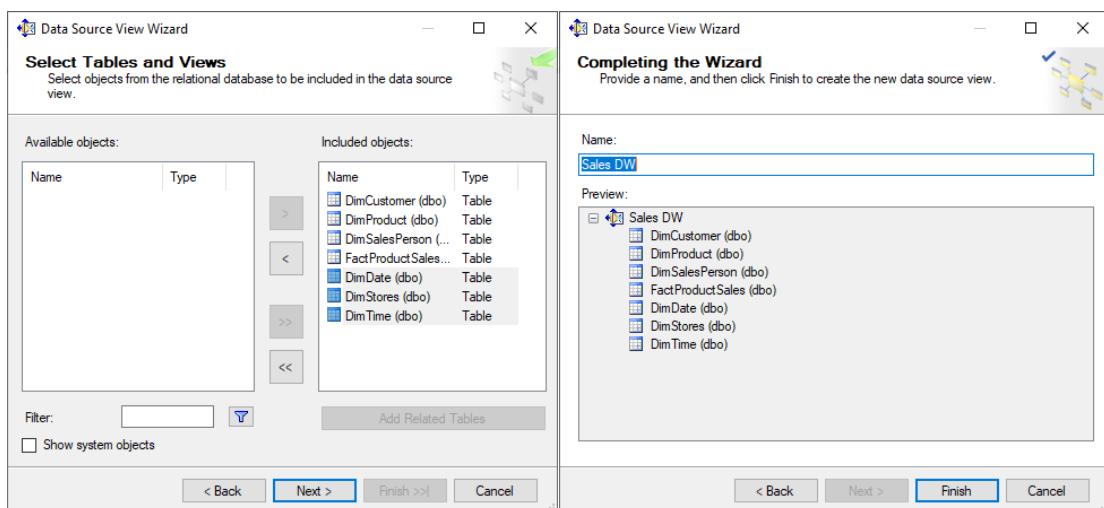
Connection string:
Provider=SQLNCLI11.1;Data Source=DESKTOP-JVQDSB9;Integrated Security=SSPI;Initial Catalog=Sales_DW

Now, Add new Data Source View

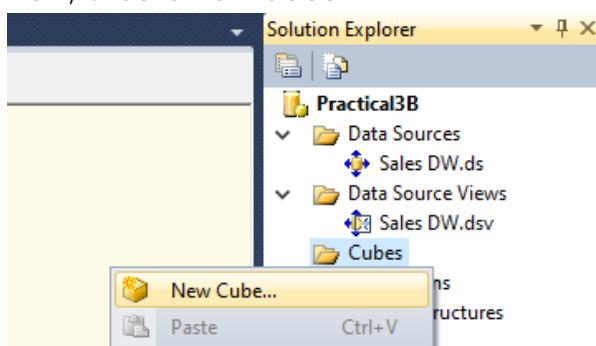




Click on Add Related Table



Now, Create new Cube



Cube Wizard

Select Creation Method
Cubes can be created by using existing tables, creating an empty cube, or generating tables in the data source.

How would you like to create the cube?

Use existing tables
 Create an empty cube
 Generate tables in the data source

Template:
(None)

Description:
Create a cube based on one or more tables in a data source.

< Back Next > Finish >> Cancel

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:

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

Suggest

< Back Next > Finish >> Cancel

Cube Wizard

Select Measures
Select measures that you want to include in the cube.

Measure

- Dim Customer
 - Dim Customer Count
 - Dim Product
 - Product Actual Cost
 - Product Sales Cost
 - Dim Product Count
 - Dim Sales Person
 - Store ID
 - Dim Sales Person Count

< Back Next > Finish >> Cancel

Cube Wizard

Select New Dimensions
Select new dimensions to be created, based on available tables.

Dimension

- Dim Customer
 - DimCustomer
- Dim Product
 - DimProduct
- Dim Sales Person
 - DimSalesPerson

< Back Next > Finish >> Cancel

Moves to the next wizard page

Cube Wizard

Completing the Wizard
Name the cube, review its structure, and then click Finish to save the cube.

Cube name:
Sales DW

Preview:

Measure groups

- Dim Customer
 - Dim Customer Count
- Dim Product
 - Product Actual Cost
 - Product Sales Cost
 - Dim Product Count
- Dim Sales Person
 - Store ID
 - Dim Sales Person Count

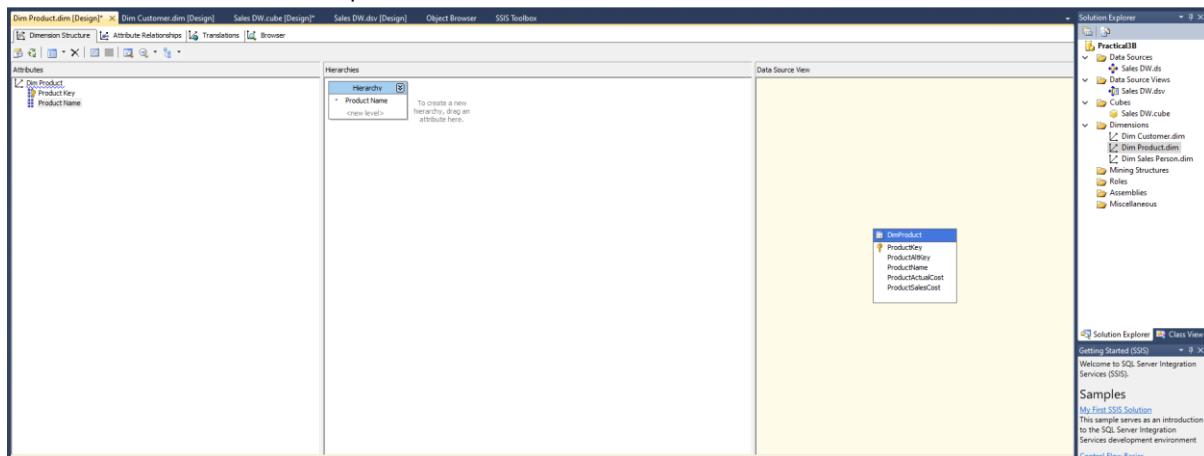
Dimensions

- Dim Customer
- Dim Product
- Dim Sales Person

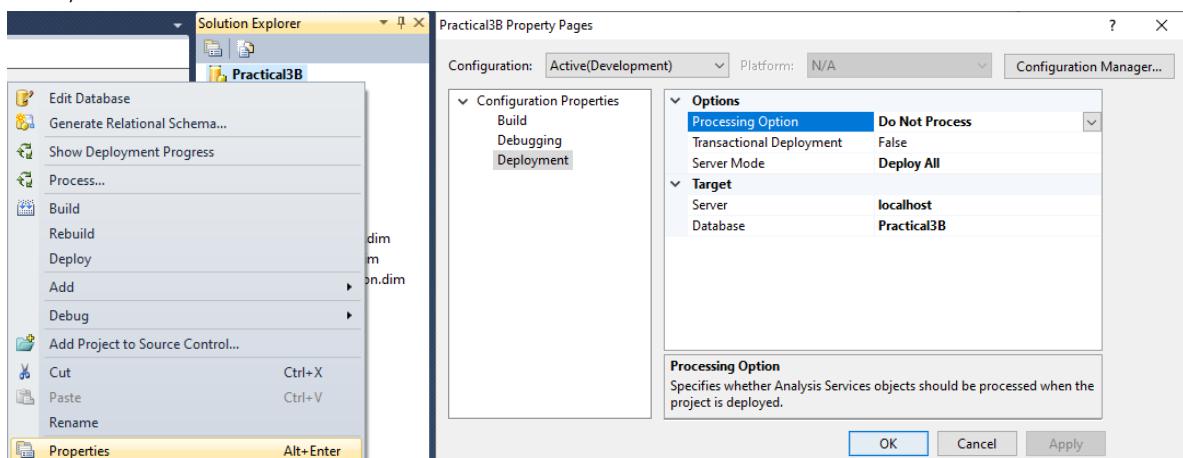
< Back Next > Finish Cancel

Now, open any dimensions

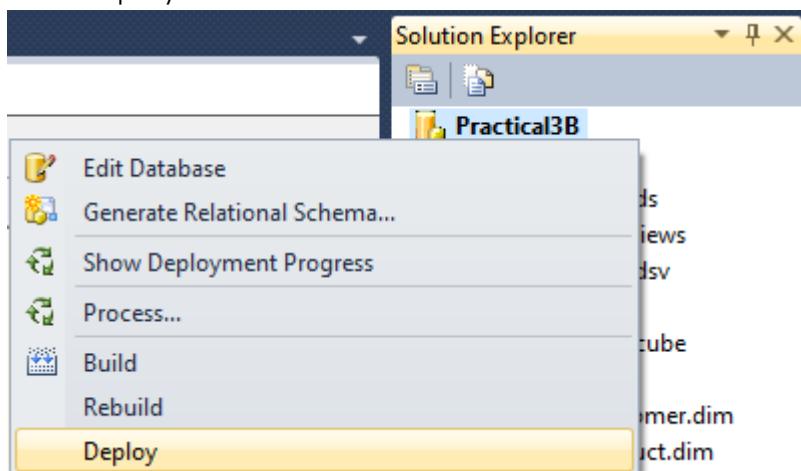
Drag any attribute from Data Source View to Attribute panel then, Drag any attribute from Attribute panel to Hierarchies Panel.



Now,



Now Deploy



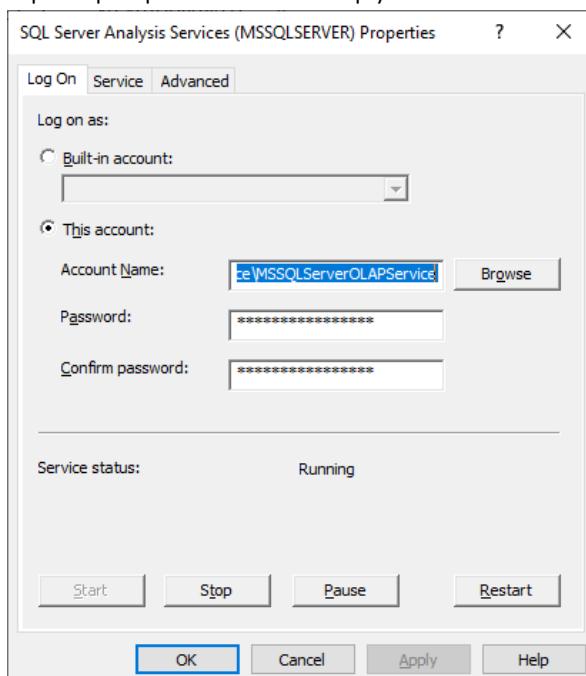
To solve this error,

- 8 Database [Practical3B] : The database has no Time dimension. Consider creating one.
- 9 Either the 'DESKTOP-JVQDSB9\administrator' user does not have permission to create a new object in 'DESKTOP-JVQDSB9', or the object does not exist.

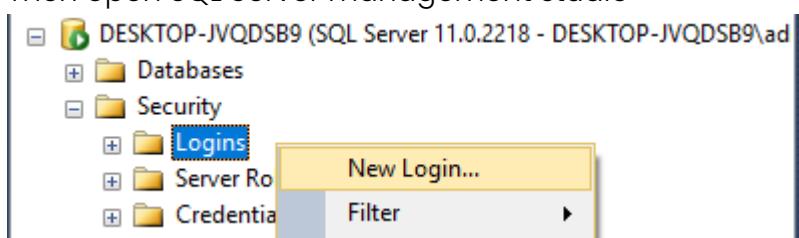
Open Sql server configuration manager:

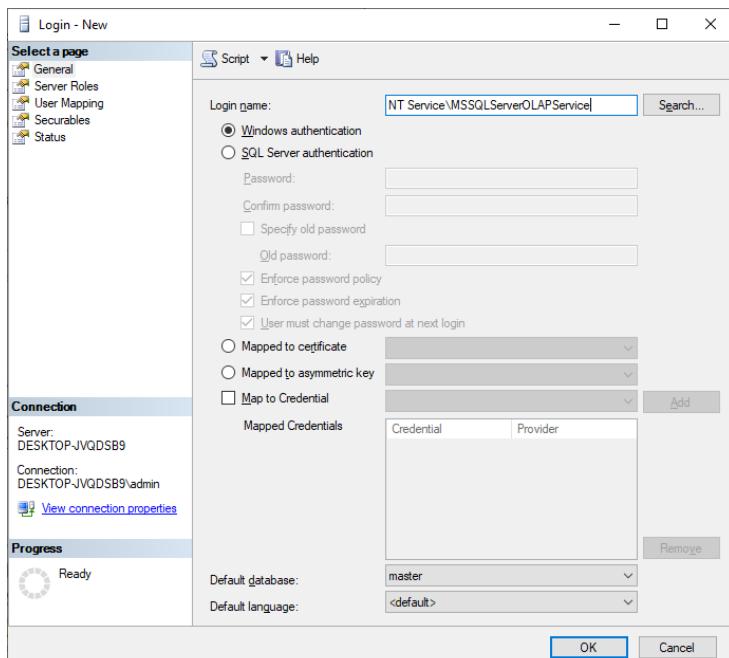
Name	State	Start Mode	Log On As	Process ID	Service Type
SQL Server Integration Services 11.0	Running	Automatic	NT Service\MsDtsS... 6276		
SQL Server (SQLEXPRESS)	Running	Automatic	NT Service\MSSQL\$... 9988		SQL Server
SQL Full-text Filter Daemon Launcher (MSSQLSERV...)	Running	Manual	NT Service\MSSQLF... 7220		
SQL Full-text Filter Daemon Launcher (SQLEXPRE...)	Running	Manual	NT Service\MSSQLF... 10936		
SQL Server Launchpad (SQLEXPRESS)	Running	Automatic	NT Service\MSSQLL... 9368		
SQL Server (MSSQLSERVER)	Running	Automatic	NT Service\MSSQLS... 6284		SQL Server
SQL Server Analysis Services (MSSQLSERVER)	Running	Automatic	NT Service\MSSQLS... 6316	Analysis Server	
SQL Server Reporting Services (MSSQLSERVER)	Start	Automatic	NT Service\ReportS... 6296		Report Server
SQL Server Agent (SQLEXPRESS)	Stop	Her (Boot, Syste... 0	NT AUTHORITY\NE...	0	SQL Agent
SQL Server Browser	Pause	Her (Boot, Syste... 0	NT AUTHORITY\LO...	0	
SQL Server Agent (MSSQLSERVER)	Resume	Manual	NT Service\SQLSER... 0		SQL Agent
	Restart				
		Properties			

Open properties and copy account name

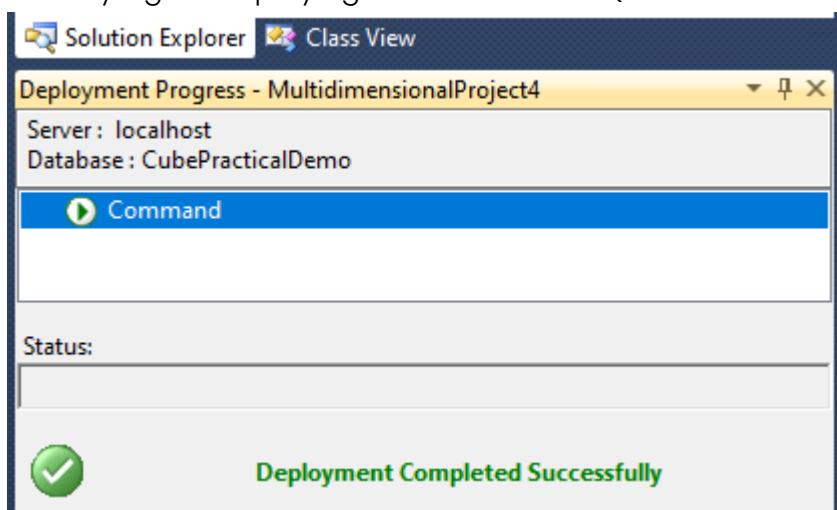


Then open SQL Server management studio

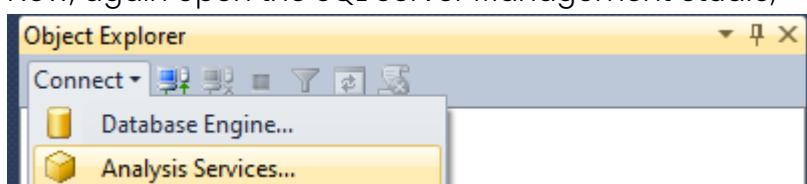


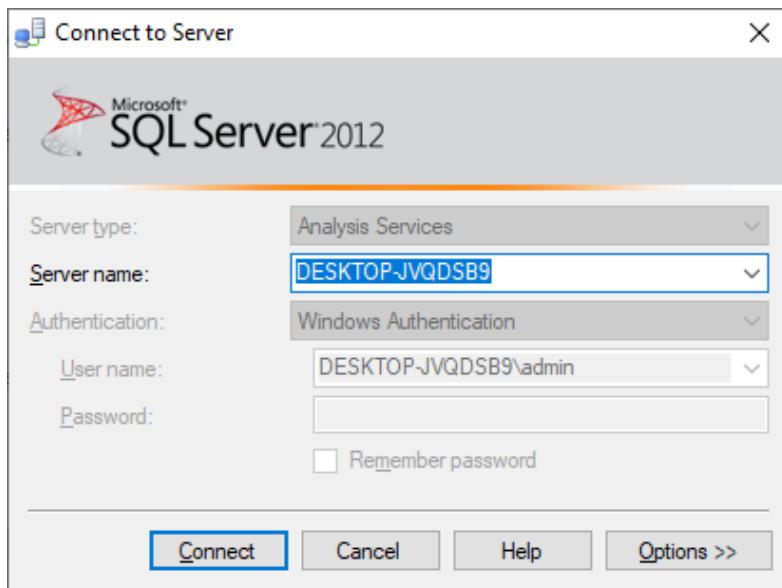


Now try again deploying the cube from SQL server data tools.



Now, again open the SQL Server Management Studio,





OUTPUT

Object Explorer

Connect ▾

DESKTOP-JVQDSB9 (Microsoft Analysis Server 11.0.2218.0 - DESKTOP-JVQDSB9)

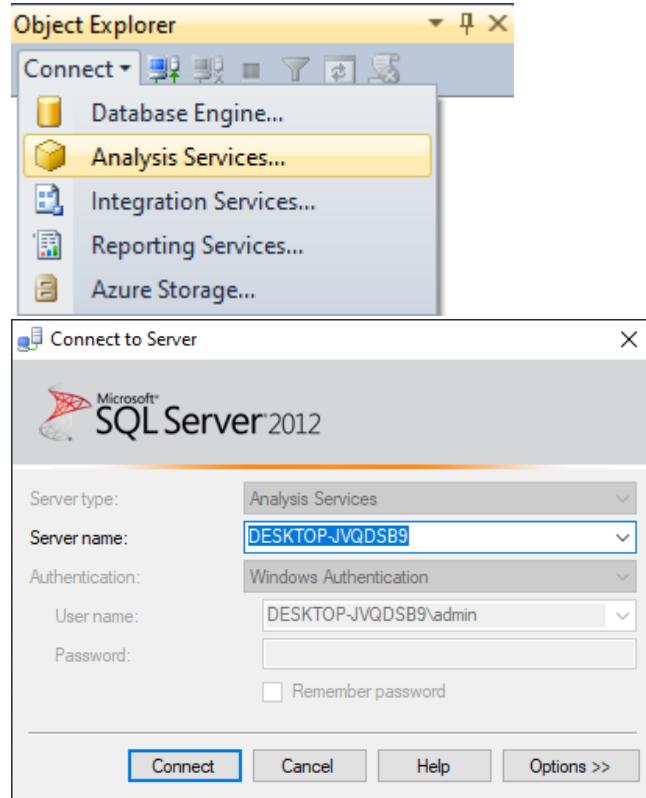
- Databases
 - cube
 - CubePracticalDemo
 - MultidimensionalProject4
 - Practical3B
 - Data Sources
 - Data Source Views
 - Cubes
 - Dimensions
 - Mining Structures
 - Roles
 - Assemblies

Practical 4

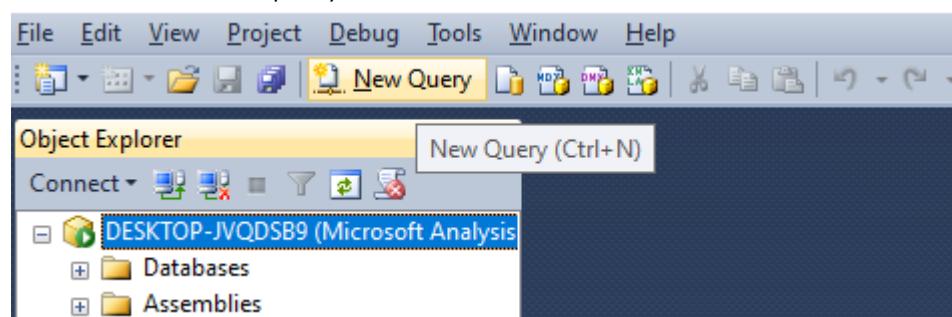
AIM: Execute the mdx queries to extract the data from data warehouse

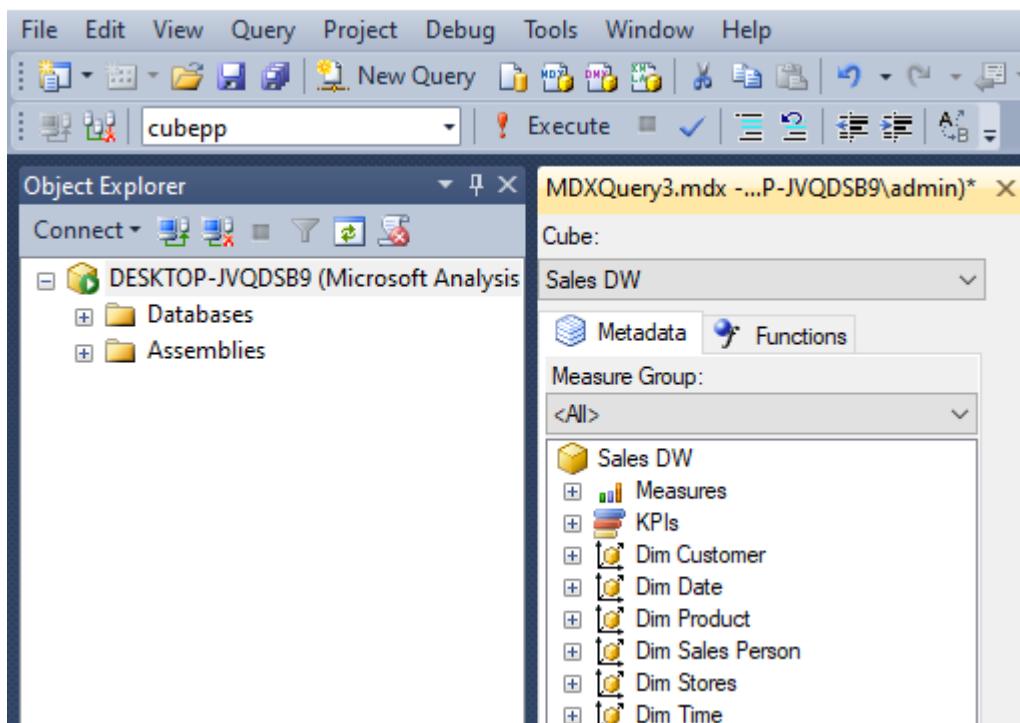
Solution:

Open SQL Server Management Studio and connect to analysis services



Now, create new query,

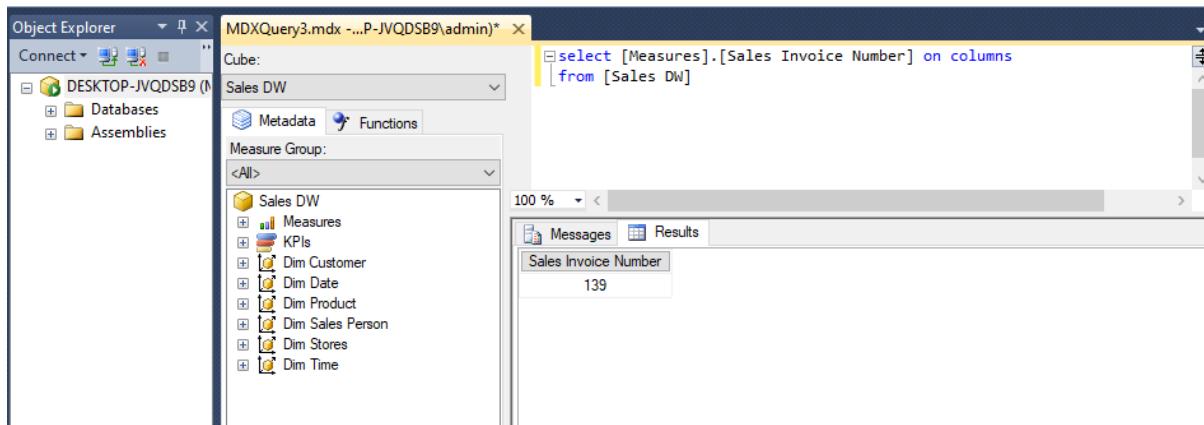




1st Query

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

Click on execute



2nd Query

select [Measures].[Sales Total Cost] on columns
from [Sales DW]

The screenshot shows the SSMS interface with the Object Explorer on the left and the MDXQuery3.mdx query window on the right. The query is:

```
select [Measures].[Sales Total Cost] on columns
from [Sales DW]
```

The results pane shows a single row:

Sales Total Cost
1231.5

3rd Query

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

The screenshot shows the SSMS interface with the Object Explorer on the left and the MDXQuery3.mdx query window on the right. The query is:

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

The results pane shows the following data:

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

4th Query

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

The screenshot shows the SSMS interface with the Object Explorer on the left and the MDXQuery3.mdx query window on the right. The query is:

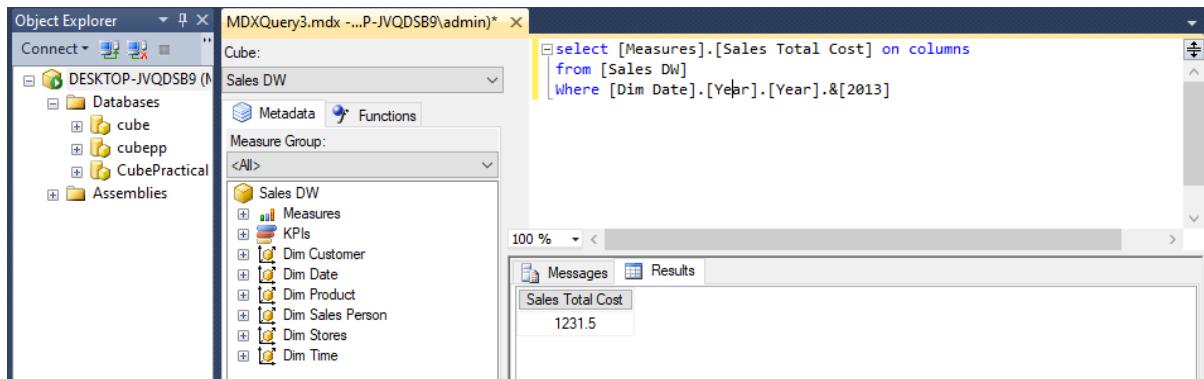
```
select [Measures].[Sales Total Cost] on columns
, NONEMPTY ({[Dim Date].[Year].[Year]}) on rows
from [Sales DW]
```

The results pane shows the following data:

Sales Total Cost	
2013	1231.5

5th Query

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



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer displays a database named 'DESKTOP-JVQDSB9' containing several objects like 'cube', 'cubapp', and 'CubePractical'. The central pane is titled 'MDXQuery3.mdx - ...P-JVQDSB9\admin)*' and contains an MDX query window. The query is:

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

Below the query, the Results tab shows the output:

Sales Total Cost
1231.5

Practical 5:

AIM:

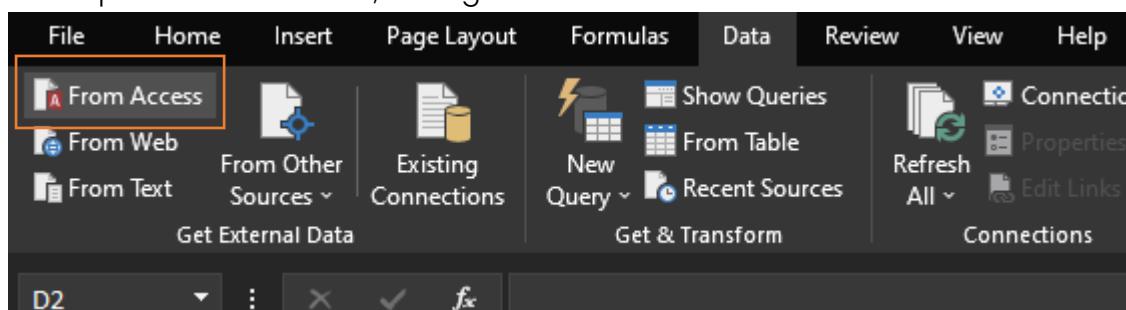
- a. Import the datawarehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.
- b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis

Solution:

a)

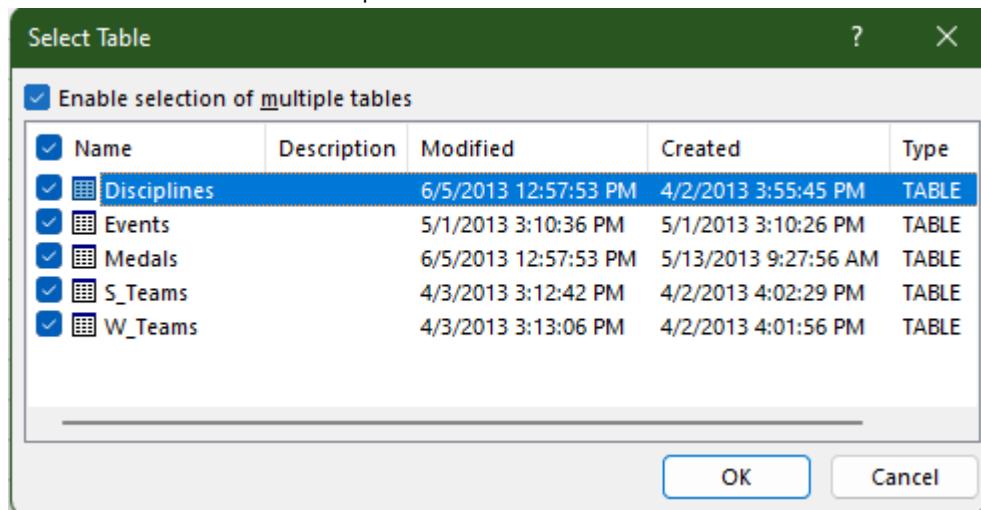
Download [OlympicMedal.accdb](#)

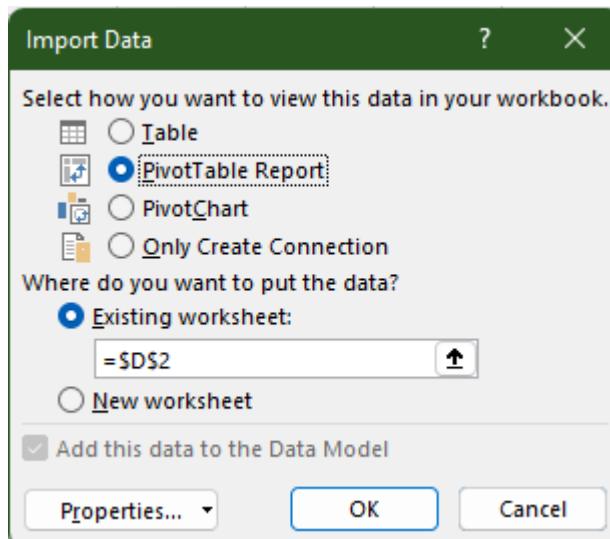
Now open Microsoft Excel, then go to Data tab and select "From Access"



Then, select the downloaded [OlympicMedal.accdb](#) file.

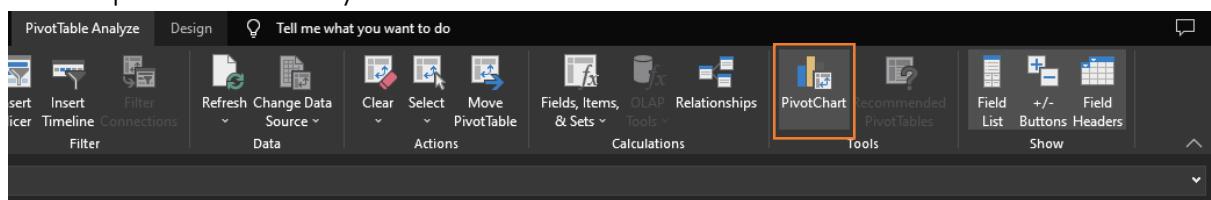
Enable selection of multiple tables and then select all the available tables.

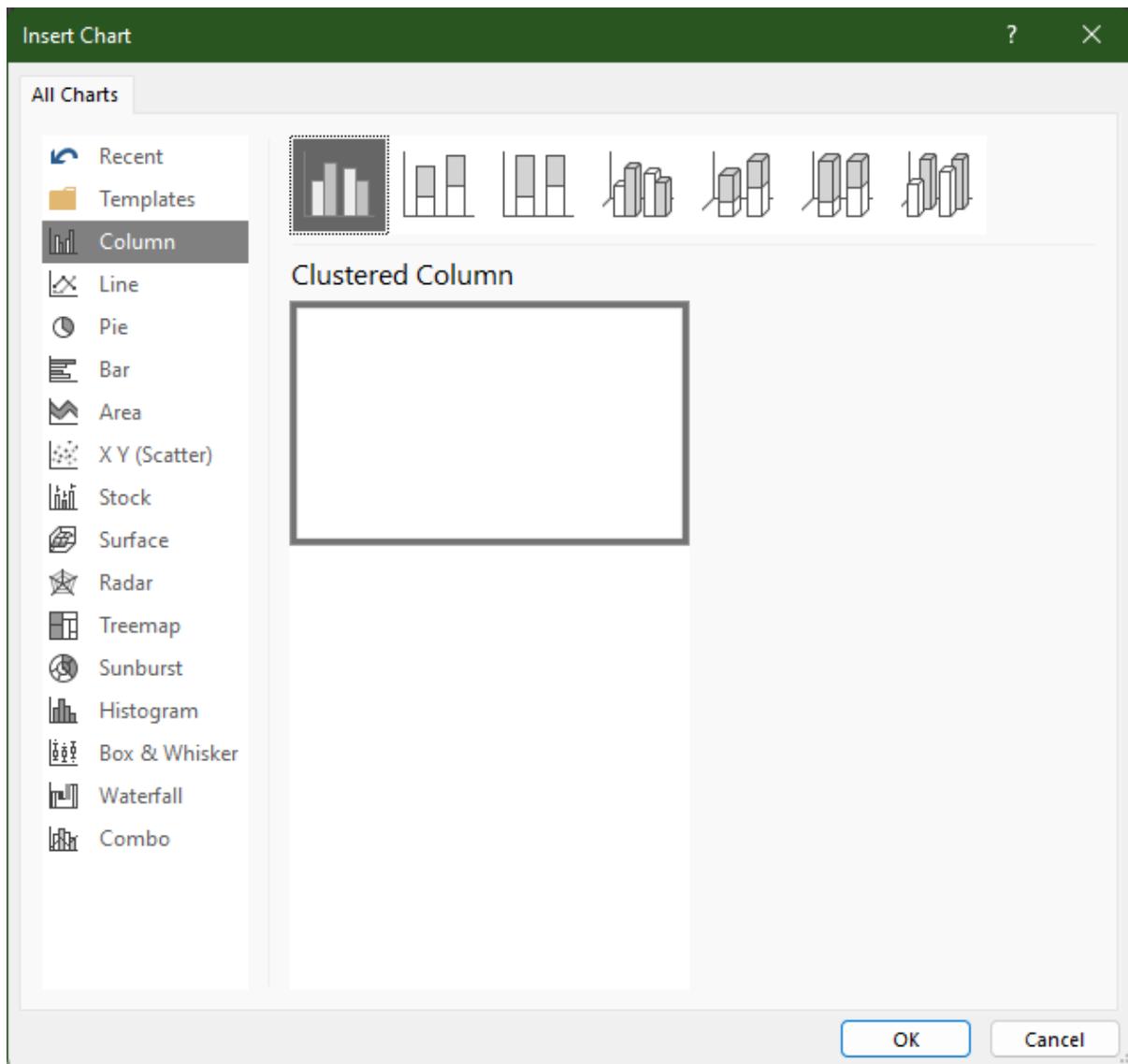




The screenshot shows an Excel spreadsheet with a PivotTable named "PivotTable1" located in cell A1. The PivotTable Fields pane on the right lists various fields such as EventID, Event, Sport, DisciplineID, Discipline, SportID, and DisciplineEvent. The PivotTable Tools ribbon tab is active.

Now in pivot table analysis select PivotChart





Book1 - Excel (Product Activation Failed)

File Home Insert Page Layout Formulas Data Review View Help PivotChart Tools Design Format

PivotTable1

To build a report, choose fields from the PivotTable Field List

Current Selection Insert Shapes Shape Styles WordArt Styles

A B C D E F G H I J K L M N O P Q

To build a PivotChart, choose fields from the PivotChart Field List.

EventID Event Sport Discipline Discipline SportID DisciplineEvent

Filters Legend (Series)

Axis (Categories) Values

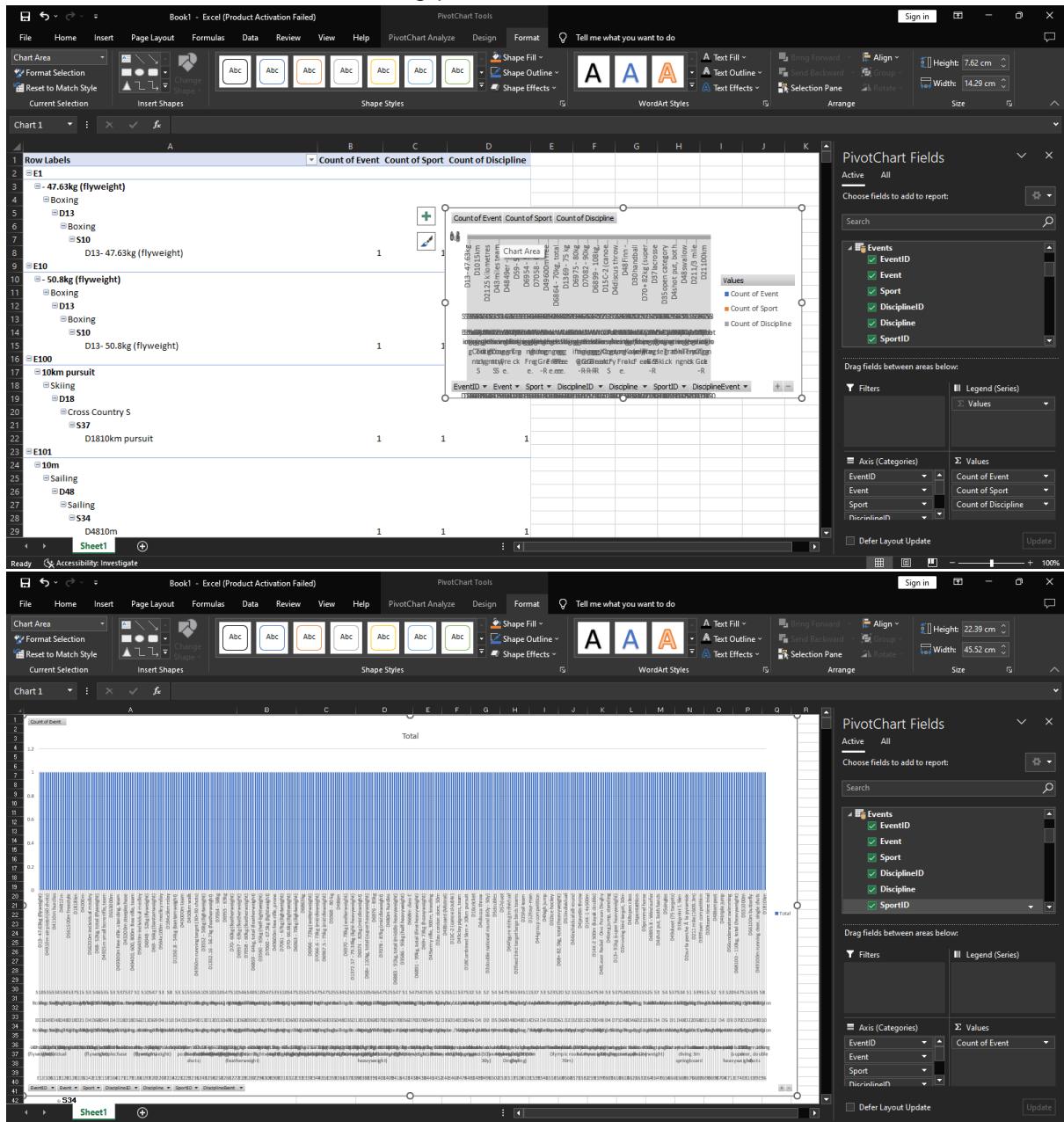
Defer Layout Update Update

Sheet1

Ready Accessibility: Investigate

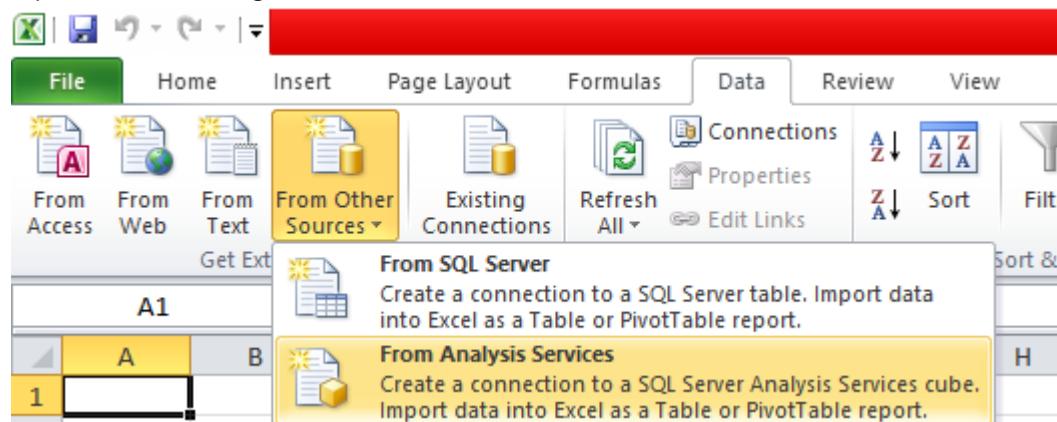
This screenshot shows the Excel ribbon with the 'PivotChart Tools' tab selected. A PivotTable is currently selected on the worksheet. The 'PivotChart Fields' pane is open on the right, displaying fields such as EventID, Event, Sport, Discipline, Discipline, SportID, and DisciplineEvent. The 'PivotTable Field List' is also visible on the left side of the worksheet area.

Now, choose the fields accordingly.

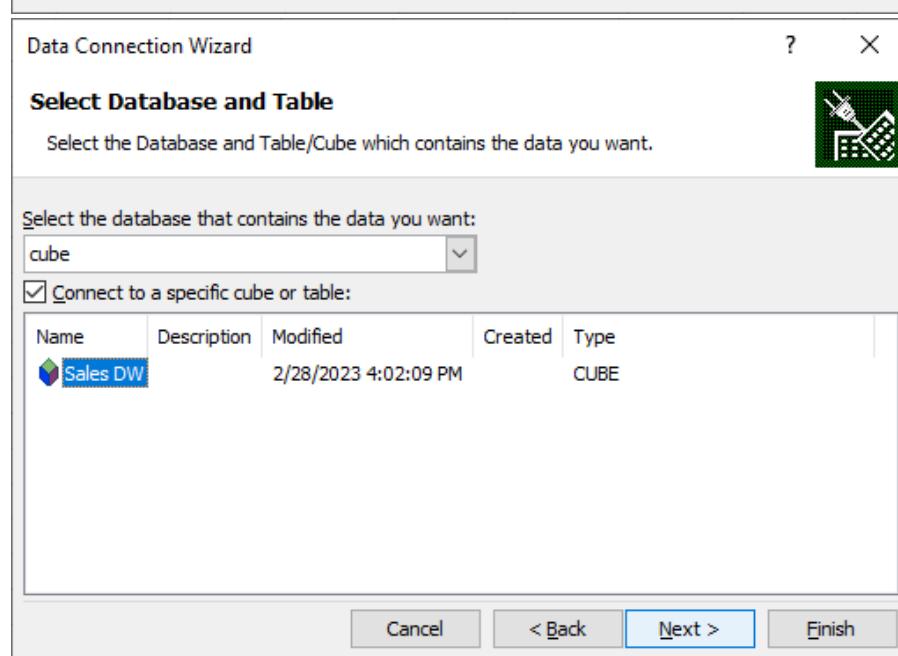
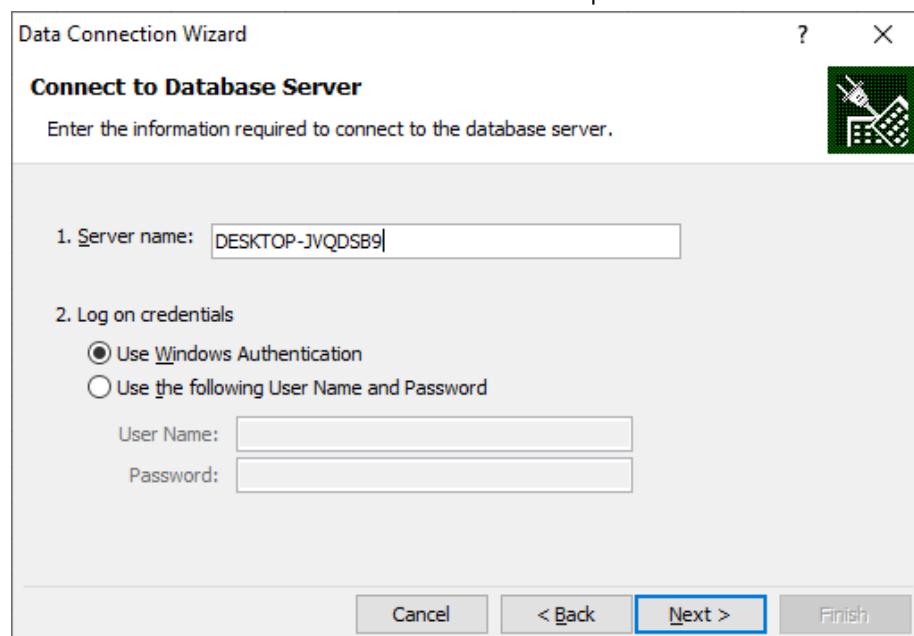


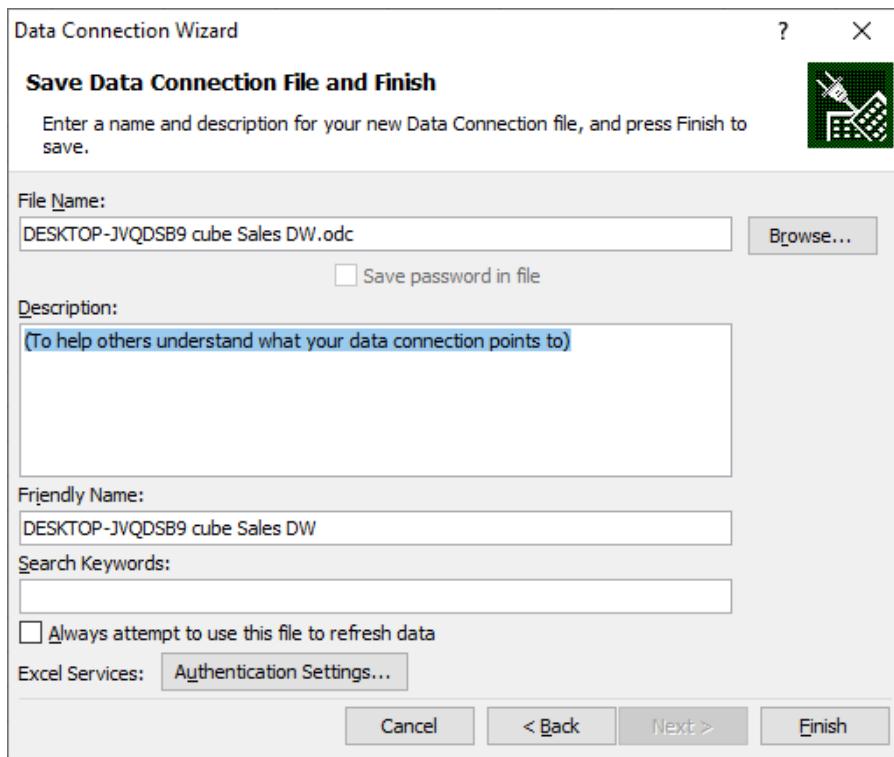
b)

Open excel and go to data:



Provide server-name where the cube is present:





Import Data

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

- Table
- PivotTable Report
- PivotChart and PivotTable Report
- Only Create Connection

Where do you want to put the data?

- Existing worksheet:
- New worksheet

PivotChart Tools

Chart 1

To build a report, choose fields from the PivotTable Field List.

PivotTable Field List

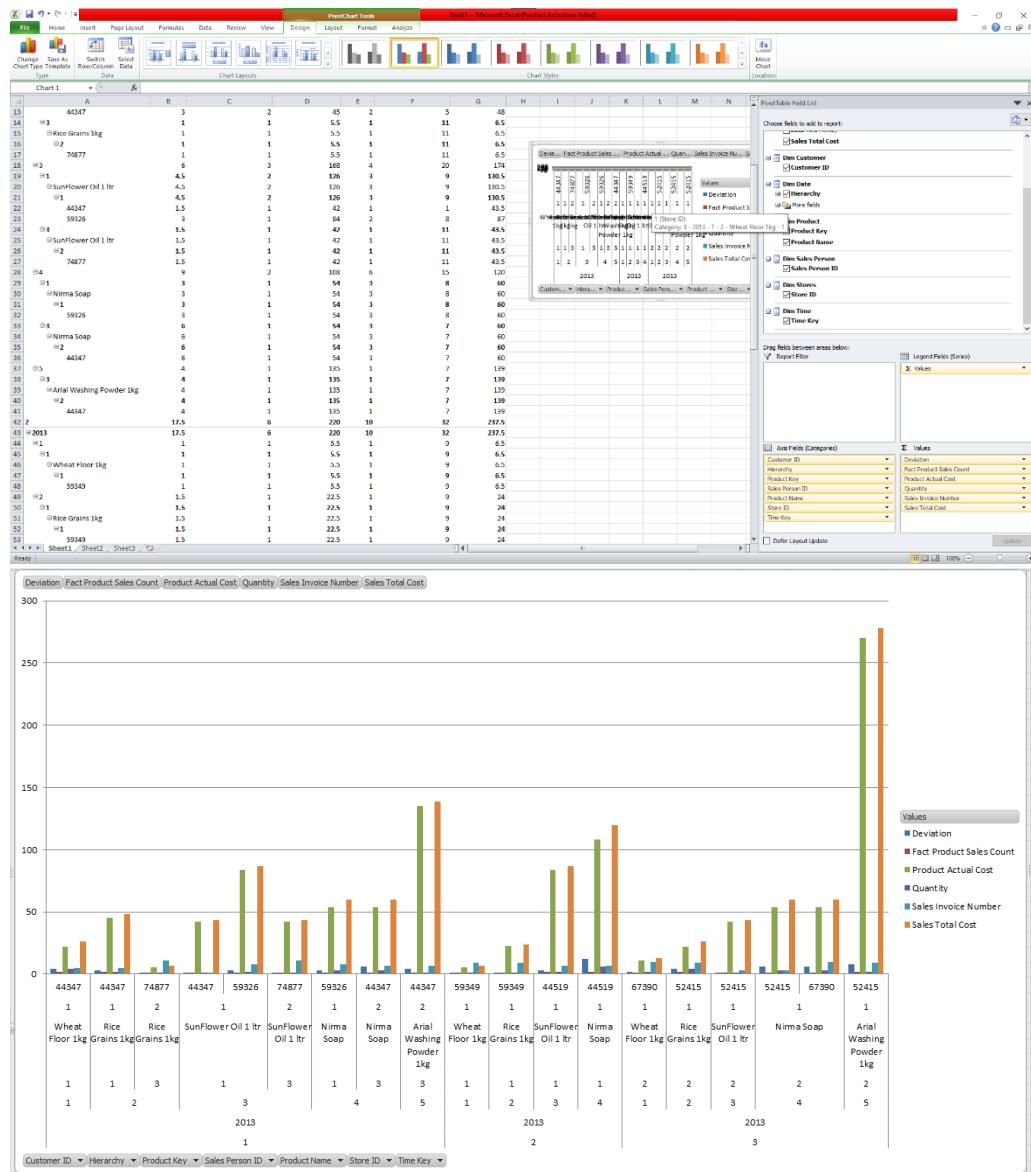
- Choose fields to add to report:
 - Fact Product Sales
 - Deviation
 - Fact Product Sales Count
 - Product Actual Cost
 - Quantity
 - Sales Invoice Number
 - Sales Time Att Key
 - Sales Total Cost
 - Dim Customer
 - Customer ID
 - Dim Date
 - Hierarchy
 - More fields
 - Dim Product
 - Product Key
 - Product Name
 - Dim Sales Person

Drag fields between areas below:

- Report Filter
- Legend Fields (Series)
- Axis Fields (Categories)
- Values

Defer Layout Update

OUTPUT



Practical 7

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

Solution:

Cell calculations

		C7	f _x	=B4*C4		
1	A	B	C	D	E	
1	Grain Store					
2						
3		total weight of wheat	% sold for the highest price			
4		150	68%			
5						
6			weight of wheat	unit profit		
7		highest price		102	50	
8		lower price		48	35	
9						
10			total profit		6780	

		C8	f _x	=B4-C7		
1	A	B	C	D	E	F
1	Grain Store					
2						
3		total weight of wheat	% sold for the highest price			
4		150	68%			
5						
6			weight of wheat	unit profit		
7		highest price		102	50	
8		lower price		48	35	
9						
10			total profit		6780	

		D10	f _x	=C7*D7+C8*D8		
1	A	B	C	D	E	
1	Grain Store					
2						
3		total weight of wheat	% sold for the highest price			
4		150	68%			
5						
6			weight of wheat	unit profit		
7		highest price		102	50	
8		lower price		48	35	
9						
10			total profit		6780	

	A	B	C	D	E
1	Grain Store				
2					
3		total weight of wheat	% sold for the highest price		
4		150	68%		
5					
6			weight of wheat	unit profit	
7		highest price		102	50
8		lower price		48	35
9					
10			total profit		6780
11					
12					

What if analysis (Scenario manager configuration)

Formulas Data Review View

Connections Properties Refresh All Edit Links Connections Z A Sort Filter Advanced Text to Columns Remove Duplicates Data Validation Consolidate What-If Analysis Group Ungroup Subt Scenario Manager... Goal Seek... Data Table...

7*D7+C8*D8 C D E F G H I J

Scenario Manager

Scenarios:

No Scenarios defined. Choose Add to add

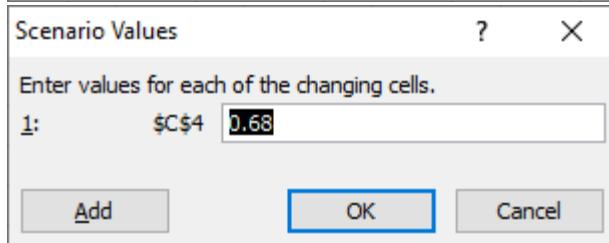
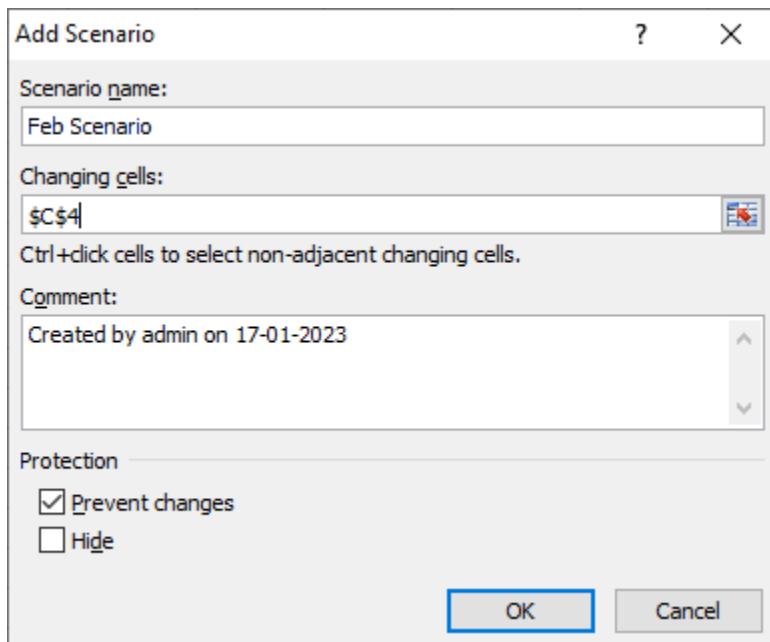
Add... Delete Edit... Merge... Summary...

Changing cells:

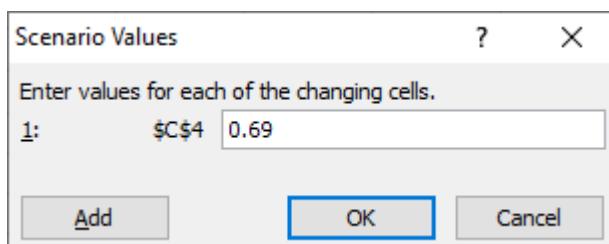
Comment:

Show Close

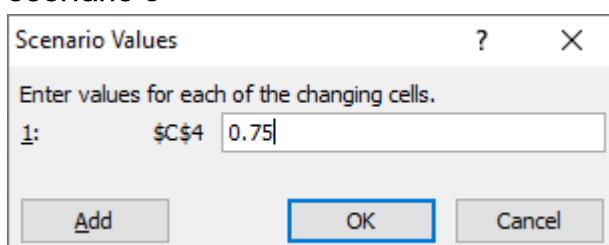
Scenario 1



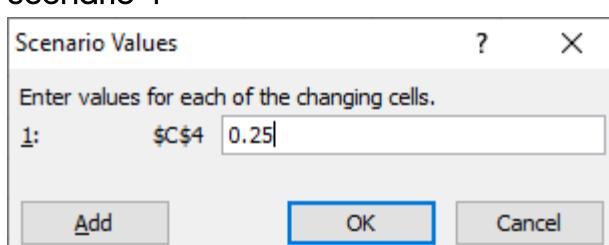
Scenario 2

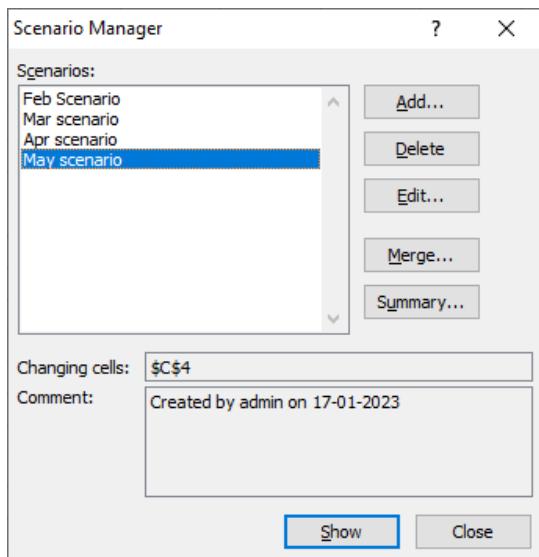


Scenario 3



Scenario 4





Click show

	A	B	C	D	E	F	G	H	I	J
1	Grain Store									
2										
3		total weight of wheat	% sold for the highest price							
4		150	75%							
5										
6			weight of wheat	unit profit						
7		highest price		112.5	50					
8		lower price		37.5	35					
9										
10			total profit	6937.5						
11										
12										
13										
14										
15										
16										
17										
18										

	A	B	C	D	E	F	G	H	I	J
1	Grain Store									
2										
3		total weight of wheat	% sold for the highest price							
4		150	25%							
5										
6			weight of wheat	unit profit						
7		highest price		37.5	50					
8		lower price		112.5	35					
9										
10			total profit	5812.5						
11										
12										
13										
14										
15										
16										
17										
18										

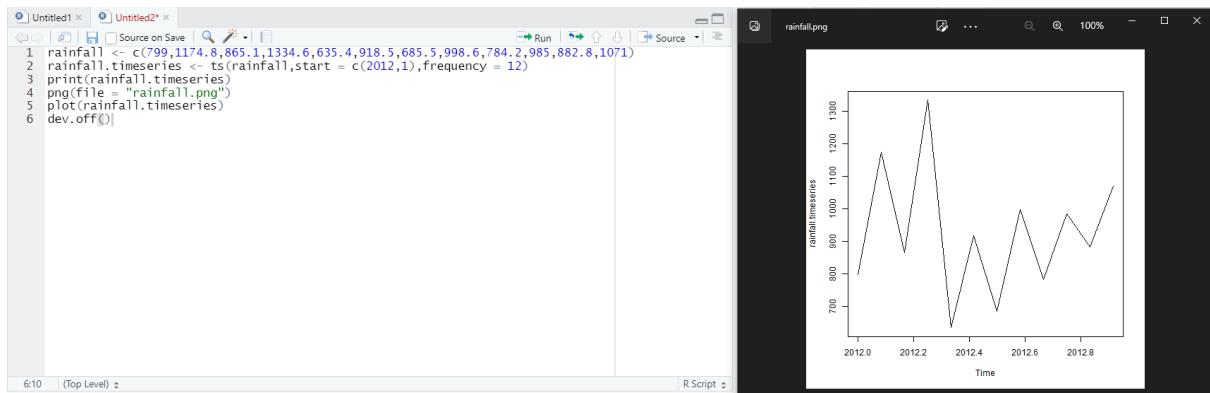
Practical 7

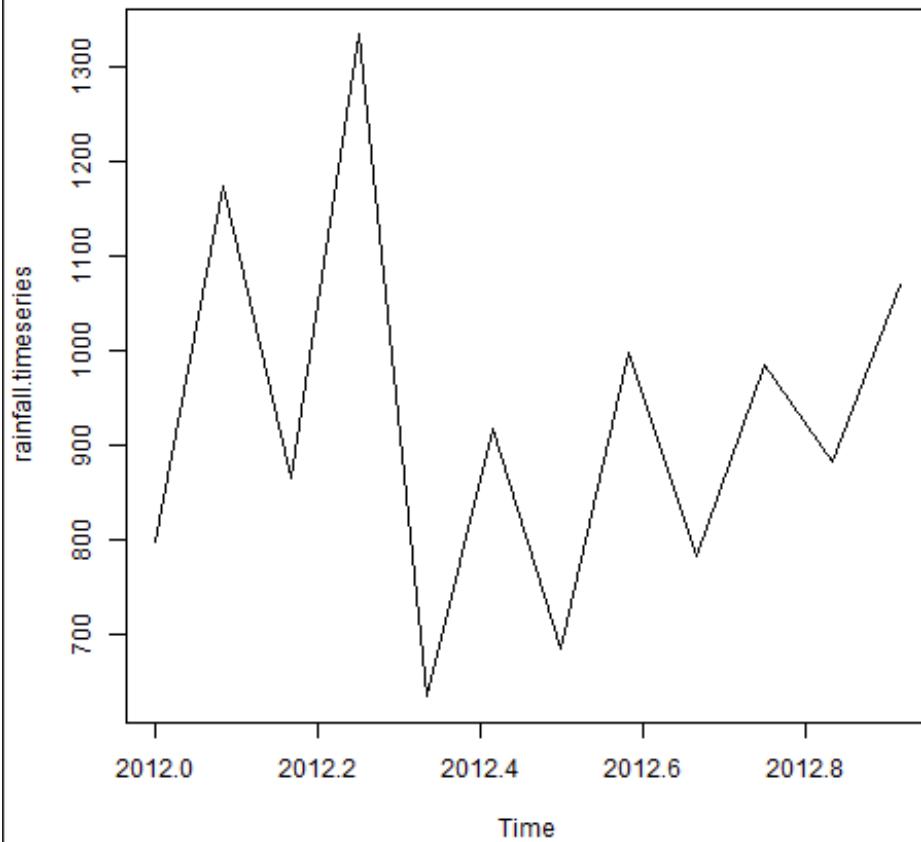
AIM: Perform the data classification using classification algorithm.

Solution:

R studio code:

```
# Get the data points in form of a R vector.  
rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)  
# Convert it to a time series object.  
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)  
# Print the timeseries data.  
print(rainfall.timeseries)  
# Give the chart file a name.  
png(file = "rainfall.png")  
# Plot a graph of the time series.  
plot(rainfall.timeseries)  
# Save the file.  
dev.off()
```





Practical 8

AIM: Perform the data clustering using clustering algorithm.

Solution:

Step 1: In RStudio run below command

```
print(iris)
```

```
> print(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
14         4.3       3.0        1.1       0.1    setosa
15         5.8       4.0        1.2       0.2    setosa
16         5.7       4.4        1.5       0.4    setosa
17         5.4       3.9        1.3       0.4    setosa
18         5.1       3.5        1.4       0.3    setosa
19         5.7       3.8        1.7       0.3    setosa
20         5.1       3.8        1.5       0.3    setosa
21         5.4       3.4        1.7       0.2    setosa
22         5.1       3.7        1.5       0.4    setosa
23         4.6       3.6        1.0       0.2    setosa
24         5.1       3.3        1.7       0.5    setosa
25         4.8       3.4        1.9       0.2    setosa
```

Step 2: Creating copy of iris dataset and removing the Species column

```
newiris <- iris
```

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

```
print(newiris)
```

```
> newiris <- iris
> newiris$Species <- NULL
> print(newiris)
   Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1       3.5        1.4       0.2
2          4.9       3.0        1.4       0.2
3          4.7       3.2        1.3       0.2
4          4.6       3.1        1.5       0.2
5          5.0       3.6        1.4       0.2
6          5.4       3.9        1.7       0.4
7          4.6       3.4        1.4       0.3
8          5.0       3.4        1.5       0.2
9          4.4       2.9        1.4       0.2
10         4.9       3.1        1.5       0.1
11         5.4       3.7        1.5       0.2
12         4.8       3.4        1.6       0.2
13         4.8       3.0        1.4       0.1
14         4.3       3.0        1.1       0.1
15         5.8       4.0        1.2       0.2
16         5.7       4.4        1.5       0.4
17         5.4       3.9        1.3       0.4
18         5.1       3.5        1.4       0.3
19         5.7       3.8        1.7       0.3
20         5.1       3.8        1.5       0.3
```

Step 3: Applying K-means to iris and storing result

```
print(iris)
```

```
newiris <- iris
```

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

```
print(newiris)
```

```
(kc <- kmeans(newiris,4))
1 print(iris)
2 newiris <- iris
3 newiris$Species <- NULL
4 print(newiris)
5 (kc <- kmeans(newiris,4))

5:26 (Top Level) R Script
Console Terminal x Background Jobs x
R 4.2.2 · ~/~
> (kc <- kmeans(newiris,4))
K-means clustering with 4 clusters of sizes 27, 28, 50, 45

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1    7.014815   3.096296    5.918519    2.155556
2    5.532143   2.635714    3.960714    1.228571
3    5.006000   3.428000    1.462000    0.246000
4    6.264444   2.884444    4.886667    1.666667

Clustering vector:
 [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 3 3 3 3 3 3 3 3
[36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[71] 4 2 4 4 4 4 4 4 4 2 2 2 2 4 2 4 4 4 2 2 2 4 2 2 2 2 4 2 2 1 4 1 4 1
[106] 1 2 1 1 1 4 4 1 4 4 4 1 1 4 1 4 1 4 1 1 4 1 1 1 4 4 1 1 1 4 4 1 1 4 4 1
[141] 1 1 4 1 1 1 4 4 4 4

Within cluster sum of squares by cluster:
[1] 15.351111 9.749286 15.151000 17.014222
(between_SS / total_SS = 91.6 %)

Available components:
[1] "cluster"      "centers"       "totss"        "withinss"
[5] "tot.withinss" "betweenss"     "size"         "iter"
[9] "ifault"
>
```

Step 4: Compare the species label with clustering result

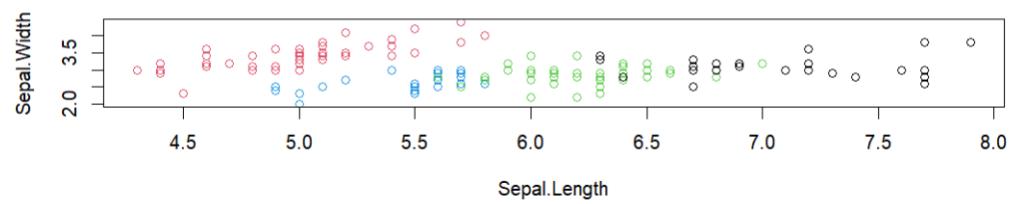
table(iris\$Species, kc\$cluster)

> **table(iris\$Species, kc\$cluster)**

	1	2	3	4
setosa	0	50	0	0
versicolor	0	0	23	27
virginica	27	0	22	1

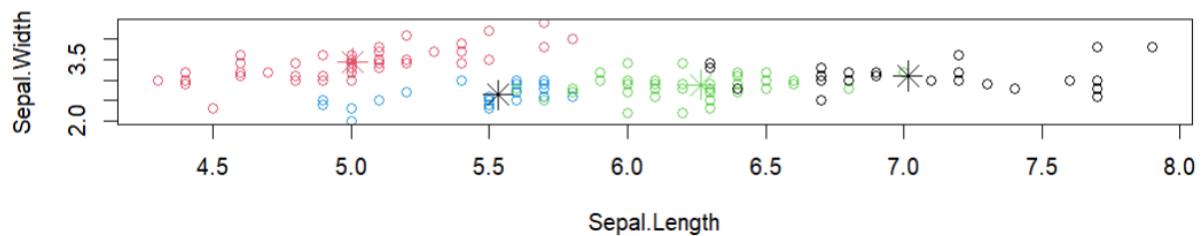
Step 5: Plot the cluster and their centers

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



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

points(kc\$centers[,c("Sepal.Length", "Sepal.Width")],col=1:3,pch=8,cex=2)



Complete code

```

print(iris)
newiris <- iris
newiris$Species <- NULL
print(newiris)
(kc <- kmeans(newiris,4))
table(iris$Species, kc$cluster)
plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)
points(kc$centers[,c("Sepal.Length", "Sepal.Width")],col=1:3,pch=8,cex=2)

```

Practical 9

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

Solution:

Step 1: Setting up data

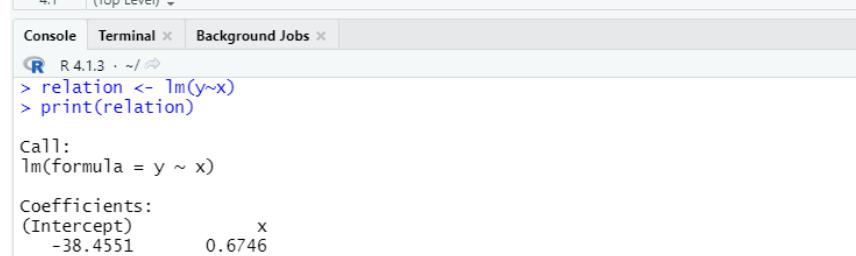
```
x <- c(151,174,138,186,128,136,179,163,152,131) # height  
y <- c(63,81,56,91,47,57,76,72,62,48) # weight
```

Step 2: printing relation

```
relation <- lm(y~x)
```

```
print(relation)
```

```
1 x <- c(151,174,138,186,128,136,179,163,152,131) # height  
2 y <- c(63,81,56,91,47,57,76,72,62,48) # weight  
3  
4 relation <- lm(y~x)  
5 print(relation)
```

4:1 (Top Level) 

```
R 4.1.3 · ~/R  
> relation <- lm(y~x)  
> print(relation)
```

Call:
lm(formula = y ~ x)

Coefficients:

(Intercept)	x
-38.4551	0.6746

Step 3: printing summary of the relation

```
print(summary(relation))
```

```
> print(summary(relation))  
  
Call:  
lm(formula = y ~ x)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-6.3002 -1.6629  0.0412  1.8944  3.9775  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -38.45509    8.04901 -4.778  0.00139 **  
x             0.67461   0.05191 12.997 1.16e-06 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 3.253 on 8 degrees of freedom  
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491  
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

Step 4: Predicting the weight of new person

```
a <- data.frame(x = 170)  
result <- predict(relation,a)  
print(result)
```

```

7 a <- data.frame(x = 170)
8 result <- predict(relation,a)
9 print(result)

7:1 (Top Level) ⇡

Console Terminal × Background Jobs ×

R 4.1.3 · ~/ ↗
> a <- data.frame(x = 170)
> result <- predict(relation,a)
> print(result)
 1
76.22869
> |

```

Step 5: Visualize the Regression Graphically

```

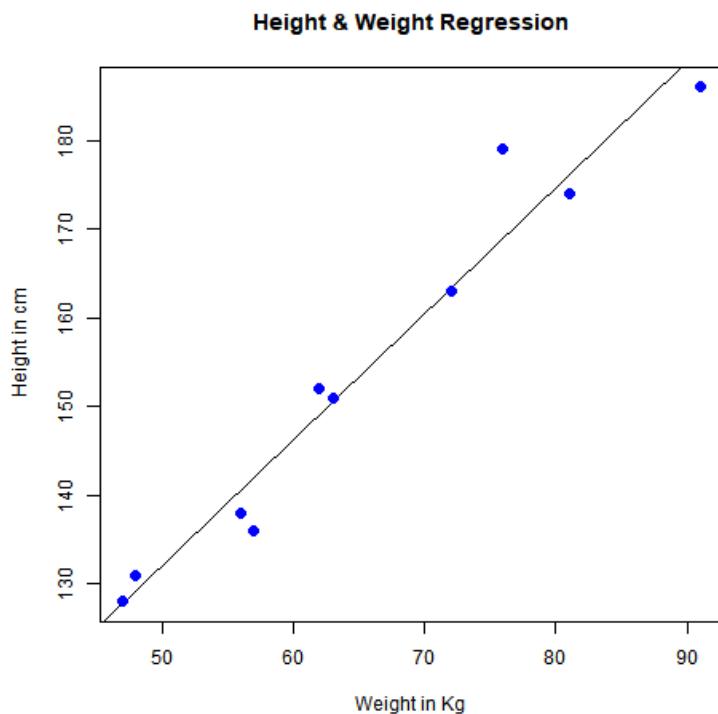
png(file = "linearregression.png")
# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in
cm")
# Save the file.
dev.off()

```

```

Console Terminal × Background Jobs ×
R 4.1.3 · ~/ ↗
> png(file = "linearregression.png")
> # Plot the chart.
> plot(y,x,col = "blue",main = "Height & weight Regression", abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "weight in Kg",ylab = "Height in cm")
> # Save the file.
> dev.off()
RStudioGD
2
> |

```



Complete Code

```
x <- c(151,174,138,186,128,136,179,163,152,131) # height
y <- c(63,81,56,91,47,57,76,72,62,48) # weight
relation <- lm(y~x)
print(summary(relation))
a <- data.frame(x = 170)
result <- predict(relation,a)
print(result)
png(file = "linearregression.png")
# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression", abline(lm(x~y)),cex =
1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in cm")
# Save the file.
dev.off()
```

Practical 10

AIM: Perform the logistic regression on the given data warehouse data.

Solution:

To perform this download the data from the link given below:

<https://github.com/TarekDib03/Analytics/blob/master/Week3%20-%20Logistic%20Regression/Data/quality.csv>

Code:

```
> quality <- read.csv('C:/Users/Anon/Desktop/BI Practical/quality.csv')
> str(quality)
```

Output:

```
'data.frame': 131 obs. of 14 variables:
$ MemberID      : int 1 2 3 4 5 6 7 8 9 10 ...
$ InpatientDays : int 0 1 0 0 8 2 16 2 2 4 ...
$ ERVisits       : int 0 1 0 1 2 0 1 0 1 2 ...
$ OfficeVisits   : int 18 6 5 19 19 9 8 8 4 0 ...
$ Narcotics      : int 1 1 3 0 3 2 1 0 3 2 ...
$ DaysSinceLastERVisit: num 731 411 731 158 449 ...
$ Pain           : int 10 0 10 34 10 6 4 5 5 2 ...
$ TotalVisits    : int 18 8 5 20 29 11 25 10 7 6 ...
$ ProviderCount  : int 21 27 16 14 24 40 19 11 28 21 ...
$ MedicalClaims  : int 93 19 27 59 51 53 40 28 20 17 ...
$ ClaimLines     : int 222 115 148 242 204 156 261 87 98 66 ...
...
$ StartedOnCombination: logi FALSE FALSE FALSE FALSE FALSE ...
...
$ AcuteDrugGapSmall : int 0 1 5 0 0 4 0 0 0 0 ...
$ PoorCare         : int 0 0 0 0 1 0 0 1 0 ...

> table(quality$PoorCare)

 0 1
98 33

> 98/131
[1] 0.7480916

> install.packages("caTools")

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Anon/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
also installing the dependency 'bitops'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/bitops_1.0-7.zip'
Content type 'application/zip' length 31679 bytes (30 KB)
downloaded 30 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caTools_1.18.2.zip'
Content type 'application/zip' length 245985 bytes (240 KB)
downloaded 240 KB
```

```
package 'bitops' successfully unpacked and MD5 sums checked
package 'caTools' successfully unpacked and MD5 sums checked
```

```
The downloaded binary packages are in
  C:\Users\Anon\AppData\Local\Temp\RtmpCyEvnb\downloaded_packages
```

```
> library(caTools)
> set.seed(88)
> split = sample.split(quality$PoorCare, splitRatio = 0.75)
> split
 [1] TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  T
RUE FALSE
 [13] TRUE  T
RUE TRUE
 [25] FALSE TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE TRUE  T
RUE TRUE
 [37] FALSE TRUE  TRUE  TRUE FALSE FALSE TRUE  TRUE FALSE TRUE  FA
LSE TRUE
 [49] FALSE TRUE  TRUE FALSE FALSE TRUE  TRUE TRUE  TRUE TRUE  T
RUE TRUE
 [61] TRUE  TRUE  TRUE  TRUE FALSE TRUE  TRUE TRUE  TRUE TRUE  FA
LSE TRUE
 [73] TRUE  T
RUE FALSE
 [85] TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE T
RUE TRUE
 [97] TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE TRUE  TRUE  T
RUE FALSE
[109] TRUE FALSE FALSE TRUE  TRUE FALSE TRUE  TRUE TRUE  TRUE FALSE T
RUE TRUE
[121] FALSE TRUE  TRUE FALSE TRUE  TRUE FALSE TRUE  TRUE TRUE  TRUE  FA
LSE

> qualityTrain = subset(quality, split == TRUE)
> qualityTest = subset(quality, split == FALSE)
> nrow(qualityTrain)
[1] 99
> nrow(qualityTest)
[1] 32

> QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics, data = qualityTrain, family = binomial)
> summary(QualityLog)

Call:
glm(formula = PoorCare ~ OfficeVisits + Narcotics, family = binomial
     , data = qualityTrain)

Deviance Residuals:
    Min      1Q      Median      3Q      Max 
-2.06303 -0.63155 -0.50503 -0.09689  2.16686 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -2.64613   0.52357 -5.054 4.33e-07 *** 
OfficeVisits  0.08212   0.03055  2.688  0.00718 **  
Narcotics     0.07630   0.03205  2.381  0.01728 *   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 111.888 on 98 degrees of freedom
Residual deviance: 89.127 on 96 degrees of freedom
AIC: 95.127

Number of Fisher Scoring iterations: 4

> predictTrain = predict(QualityLog, type = "response")
> summary(predictTrain)
   Min. 1st Qu. Median Mean 3rd Qu. Max.
0.06623 0.11912 0.15967 0.25253 0.26765 0.98456

> tapply(predictTrain, qualityTrain$PoorCare, mean)
0          1
0.1894512 0.4392246
> table(qualityTrain$PoorCare, predictTrain > 0.5)

    FALSE TRUE
0      70    4
1      15   10
> 10/25
[1] 0.4
> 70/74
[1] 0.9459459
> table(qualityTrain$PoorCare, predictTrain > 0.7)

    FALSE TRUE
0      73    1
1      17    8

> 8/25
[1] 0.32
> 73/74
[1] 0.9864865
> table(qualityTrain$PoorCare, predictTrain > 0.2)

    FALSE TRUE
0      54   20
1       9   16
> 16/25
[1] 0.64
> 54/74
[1] 0.7297297

> install.packages("ROCR")

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Anon/AppData/Local/R/win-library/4
.2'
(as 'lib' is unspecified)
also installing the dependencies 'gtools', 'gplots'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/gtools_3.9.4.zip'
Content type 'application/zip' length 359411 bytes (350 KB)
downloaded 350 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/gplots_3.1.3.zip'
Content type 'application/zip' length 603183 bytes (589 KB)
downloaded 589 KB

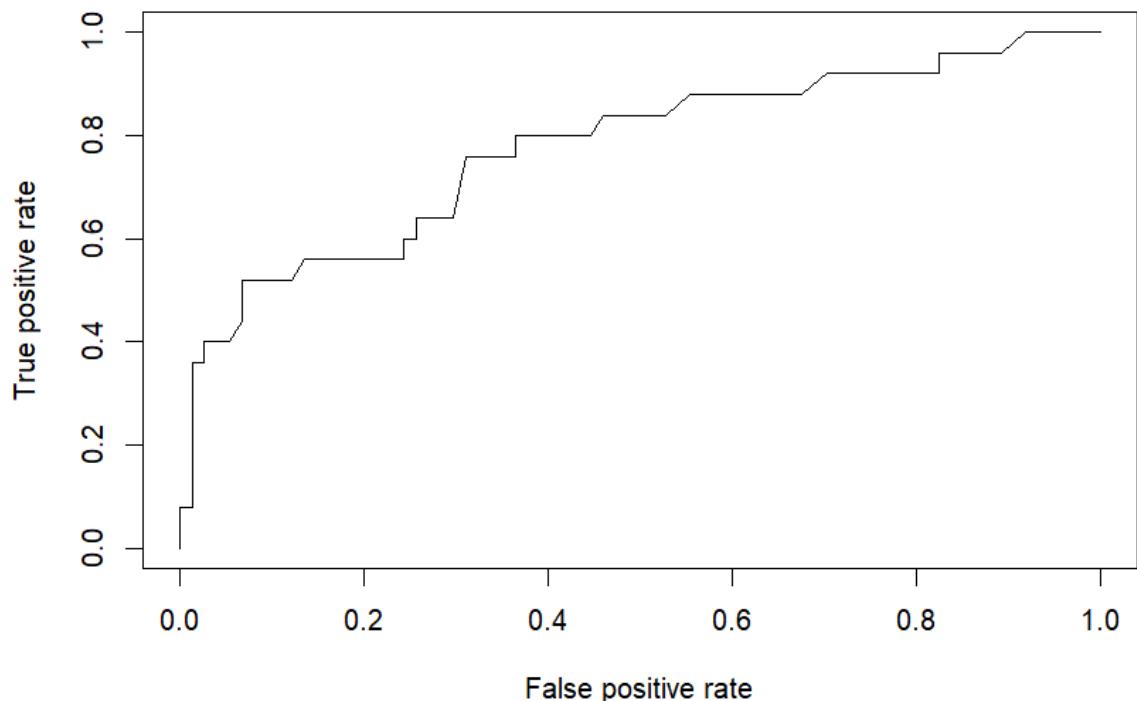
```

```
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/ROCR_1.0-11.zip'
Content type 'application/zip' length 453649 bytes (443 KB)
downloaded 443 KB
```

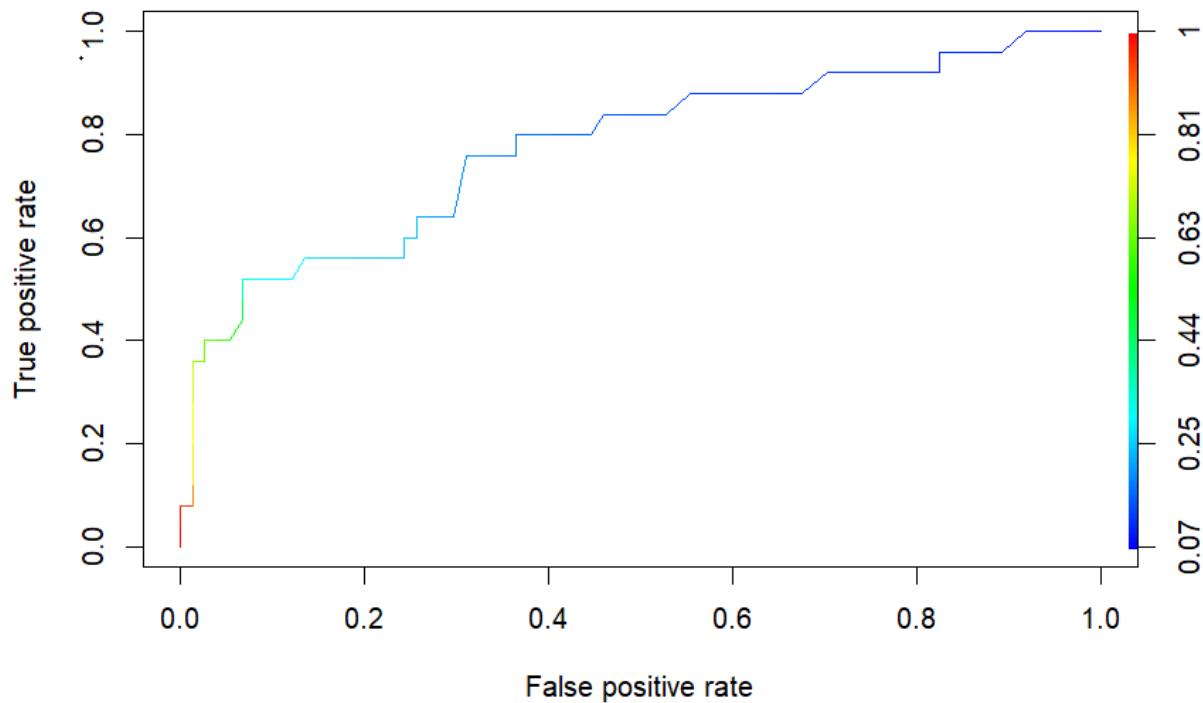
```
package 'gtools' successfully unpacked and MD5 sums checked
package 'gplots' successfully unpacked and MD5 sums checked
package 'ROCR' successfully unpacked and MD5 sums checked
```

```
The downloaded binary packages are in
  C:\Users\Anon\AppData\Local\Temp\RtmpCyEvnb\downloaded_packages
```

```
> library(ROCR)
> ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)
> ROCRpref = performance(ROCRpred, "tpr", "fpr")
> plot(ROCRpref)
```



```
> plot(ROCRpref, colorize=TRUE)
```



Complete code:

```

quality <- read.csv('C:/Users/Anon/Desktop/BI Practical/quality.csv')
str(quality)
table(quality$PoorCare)
98/131
install.packages("caTools")
library(caTools)
set.seed(88)
split = sample.split(quality$PoorCare, SplitRatio = 0.75)
split
qualityTrain = subset(quality, split == TRUE)
qualityTest = subset(quality, split == FALSE)
nrow(qualityTrain)
nrow(qualityTest)
QualityLog = glm(PoorCare ~ officeVisits + Narcotics, data = qualityTrain, family = binomial)
summary(QualityLog)
predictTrain = predict(QualityLog, type = "response")
summary(predictTrain)
tapply(predictTrain, qualityTrain$PoorCare, mean)
table(qualityTrain$PoorCare, predictTrain > 0.5)
10/25
70/74
table(qualityTrain$PoorCare, predictTrain > 0.7)
8/25
73/74
table(qualityTrain$PoorCare, predictTrain > 0.2)
16/25

```

54/74

```
install.packages("ROCR")
library(ROCR)
ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)
ROCRpref = performance(ROCRpred, "tpr", "fpr")
plot(ROCRpref)
plot(ROCRpref, colorize=TRUE)
plot(ROCRpref, colorize=TRUE, print, print.cutoffs.at = seq(0,1,by =
0.1), text.adj = c(-0.2,0.7))
```

Practical 11

AIM: Practical implementation of Decision tree using R tool

Solution:

Step 1: Install "party package"

```
>install.packages("party")
```

Step 2: Code

```
library(party)  
print(head(readingSkills))
```

The screenshot shows the RStudio interface. In the top-left pane, there is a code editor with three lines of R code:

```
1 library(party)  
2 print(head(readingSkills))  
3
```

In the bottom-left pane, the 'Console' tab is active, showing the same R code being run and the resulting output:

```
> library(party)  
> print(head(readingSkills))
```

	nativeSpeaker	age	shoeSize	score
1	yes	5	24.83189	32.29385
2	yes	6	25.95238	36.63105
3	no	11	30.42170	49.60593
4	yes	7	28.66450	40.28456
5	yes	11	31.88207	55.46085
6	yes	10	30.07843	52.83124

```
input.dat <- readingSkills[c(1:105),]  
png(file = "decision_tree.png")  
output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)  
plot(output.tree)  
dev.off()
```

OUTPUT

```

4 input.dat <- readingSkills[c(1:105),]
5 png(file = "decision_tree.png")
6 output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)
7 plot(output.tree)
8 dev.off()

```

8:10 (Top Level) ⇣

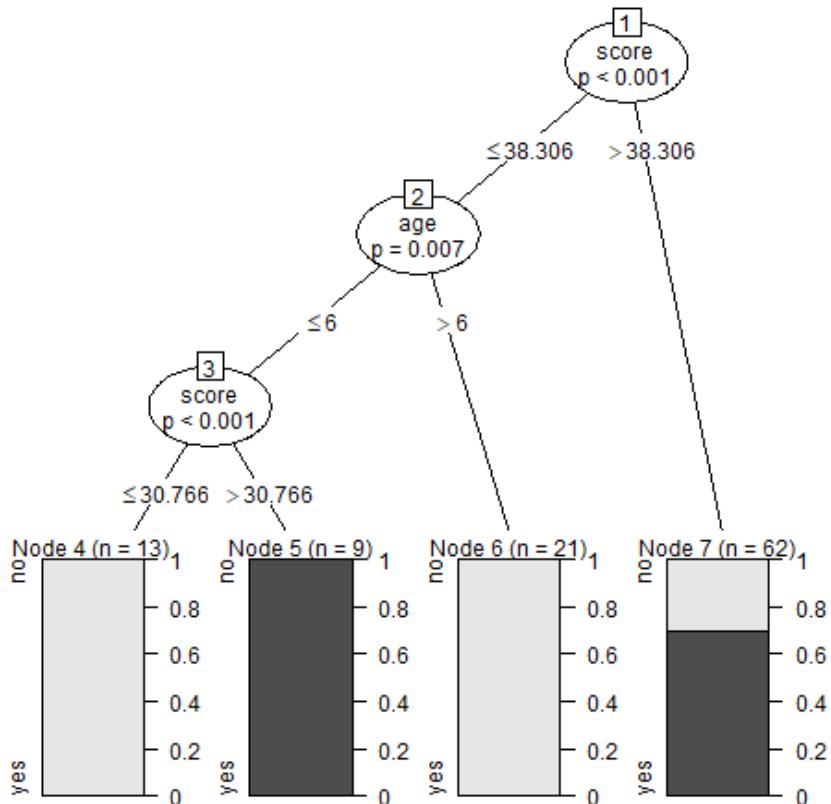
Console Terminal ✎

~/

```

> input.dat <- readingSkills[c(1:105),]
> png(file = "decision_tree.png")
> output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)
> plot(output.tree)
> dev.off()
null device
1
> |

```



Complete Code

```

library(party)
print(head(readingSkills))
input.dat <- readingSkills[c(1:105),]
png(file = "decision_tree.png")
output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score,data = input.dat)
plot(output.tree)
dev.off()

```