SSIS Use Case

Requirement

We have a Sales data we need to do some cleaning. First, need to add a Full_Name column with the First_Name and Last_Name. Next, we don't need the Computer data. We need to find the Total_Sales with the formula [ItemSold*SoldPrice]. Sort TotalSales in Descending. Replace the ProductName Cell Phone with Mobiles. we need to separate the data based on the Region.

- The source Table contains Sales records with ID, First_Name, Last_Name, ProductName, ItemSold, SoldPrice, Country, and Region details.
- The target files should contain the following columns: ID, Full_Name, ProductName, ItemSold, SoldPrice, TotalSales, Country and Region details.

Note: Copy the below Source data paste it into the notepad and save it in txt format. And it should be a comma "," delimited file. The file name should be "Sales". The target file also should be in text format.

Sample Source Data:

ID	First_Name	Last_Name	ProductName	ItemSold	SoldPrice	Country	Region
1	Aamir	Shahzab	TV	1	700	USA	North America
2	M	Raza	Cell Phone	2	800	USA	North America
3	Christy	Ladson	TV	3	1600	USA	North America
4	John	Rivers	Laptop	5	2400	USA	North America
5	Najaf	Ali	Computer	1	300	Pakistan	Asia
6	Sukhjeet	Singh	TV	2	900	India	Asia
7	Chirag	Patel	Cell Phone	5	1500	India	Asia
8	Aleena	Aman	Laptop	2	800	Pakistan	Asia
9	Petra	Henry	TV	10	5000	France	Europe
10	Rita	Roger	Laptop	7	2100	France	Europe
11	Tamara	Tony	Cell Phone	2	1200	Germany	Europe

Sample Target North_America_Sales:

ID	Full_Name	ProductName	ItemSold	SoldPrice	Total_Sales	Country	Region
4	John Rivers	Laptop	5	2400	12000	USA	North America
3	Christy Ladson	TV	3	1600	4800	USA	North America
2	M Raza	Mobiles	2	800	1600	USA	North America
1	Aamir Shahzad	TV	1	700	700	USA	North America

Sample Target Asia_Sales:

ID	Full_Name	ProductName	ItemSold	SoldPrice	Total_Sales	Country	Region
7	Chirag Patel	Mobiles	5	1500	7500	India	Asia
6	Sukhjeet Singh	TV	2	900	1800	India	Asia
8	Aleena Aman	Laptop	2	800	1600	Pakistan	Asia

Sample Target Europe_Sales:

ID	Full_Name	ProductName	ItemSold	SoldPrice	Total_Sales	Country	Region
9	Petra Henry	TV	10	5000	50000	France	Europe
10	Rita Roger	Laptop	7	2100	14700	France	Europe
11	Tamara Tony	Mobiles	2	1200	2400	Germany	Europe

Table Format

Source File: - Source file records are separated by comma ","
Target Files: - These file records are separated by a comma ","

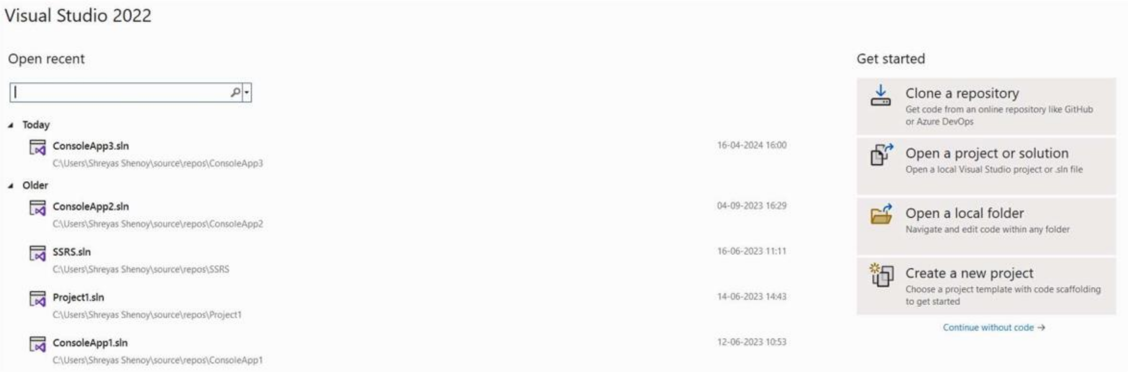
Source file details:

The Associates need to do the following tasks as per the naming standards. Source Table Name: TotalSales

Target Table Names: North_America_Sales, Asia_Sales and Europe_Sales

Implementation

- Open the Visual Studio. Click on Create a New Project.



Visual Studio 2022

Open recent

Today

ConsoleApp3.sln C:\Users\Shreyas Shenoy\source\repos\ConsoleApp3 16-04-2024 16:00

Older

ConsoleApp2.sln C:\Users\Shreyas Shenoy\source\repos\ConsoleApp2 04-09-2023 16:29

SSRS.sln C:\Users\Shreyas Shenoy\source\repos\SSRS 16-06-2023 11:11

Project1.sln C:\Users\Shreyas Shenoy\source\repos\Project1 14-06-2023 14:43

ConsoleApp1.sln C:\Users\Shreyas Shenoy\source\repos\ConsoleApp1 12-06-2023 10:53

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

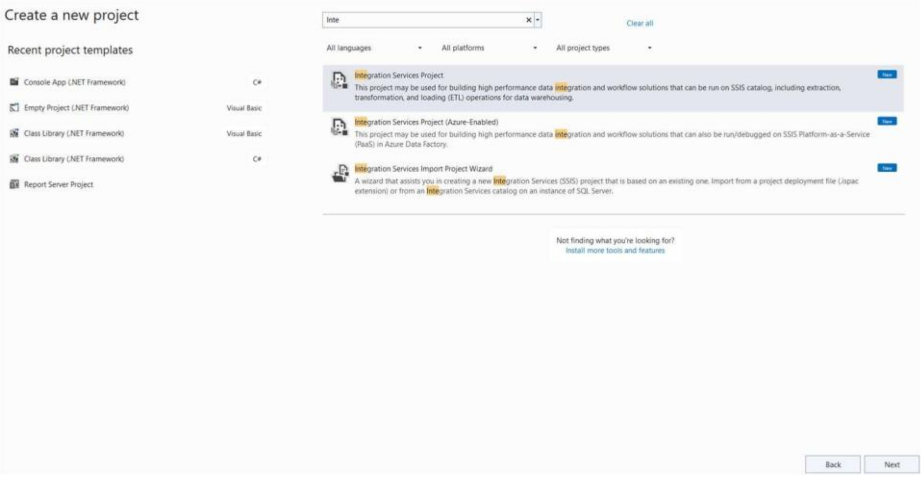
Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

- Select Integration Service and click on Next.



Create a new project

Recent project templates

Console App (.NET Framework) C#
Empty Project (.NET Framework) Visual Basic
Class Library (.NET Framework) Visual Basic
Class Library (.NET Framework) C#
Report Server Project

Inte

Clear all

All languages · All platforms · All project types

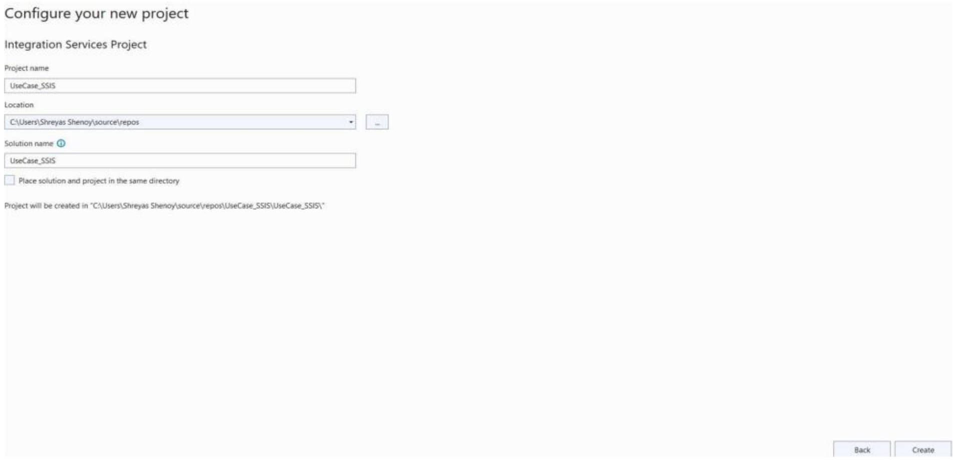
Integration Services Project
This project may be used for building high performance data integration and workflow solutions that can be run on SSIS catalog, including extraction, transformation, and loading (ETL) operations for data warehousing.

Integration Services Project (Azure-Enabled)
This project may be used for building high performance data integration and workflow solutions that can also be run/debugged on SSIS Platform-as-a-Service (PaaS) in Azure Data Factory.

Integration Services Import Project Wizard
A wizard that assists you in creating a new Integration Services (SSIS) project that is based on an existing one. Import from a project deployment file (.ispac extension) or from an Integration Services catalog on an instance of SQL Server.

Not finding what you're looking for?
Install more tools and features

- Give a Name and click on create.



Configure your new project

Integration Services Project

Project name: UseCase_SSIS

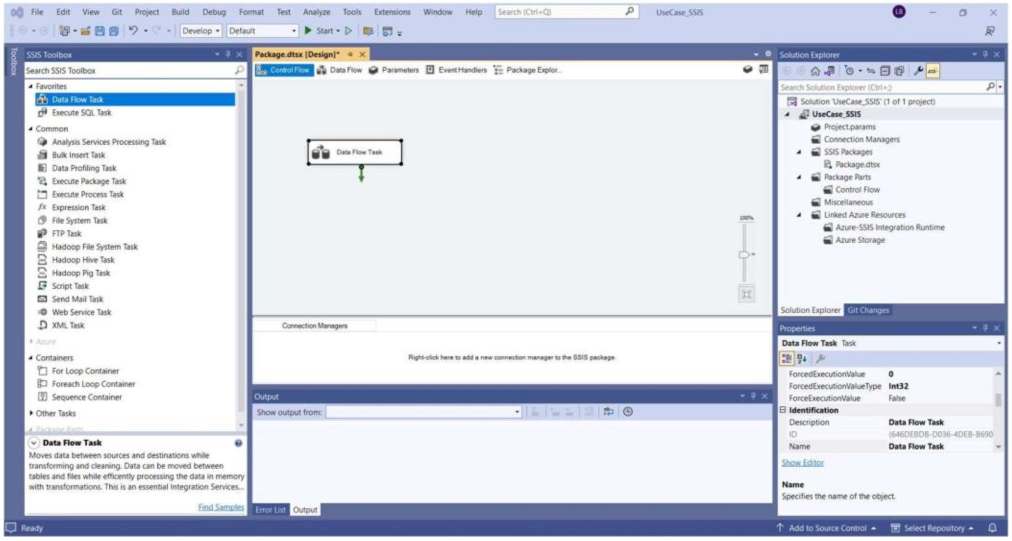
Location: C:\Users\Shreyas Shenoy\source\repos

Solution name: UseCase_SSIS

□ Place solution and project in the same directory

Project will be created in "C:\Users\Shreyas Shenoy\source\repos\UseCase_SSIS\UseCase_SSIS\"

- This is our Integration Service package, drag and drop the Data Flow task into the package.



Data Flow Task

Moves data between sources and destinations while transforming and cleaning. Data can be moved between tables and files while efficently processing the data in memory with transformations. This is an essential Integration Services...

Find Samples

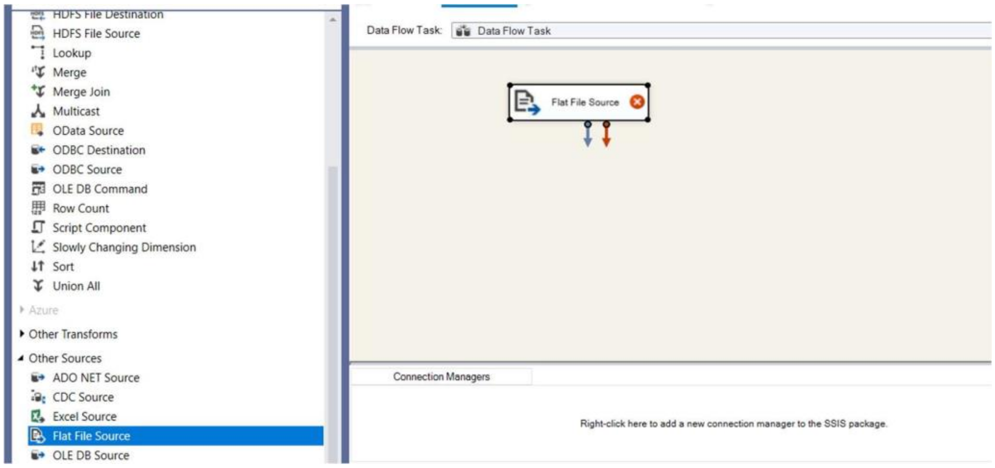
Data Flow Task Task

ForcedExecutionValue 0
ForcedExecutionValueType Int32
ForceExecutionValue False
Identification
Description Data Flow Task
ID {646DEBDB-D036-4DEB-B690
Name Data Flow Task

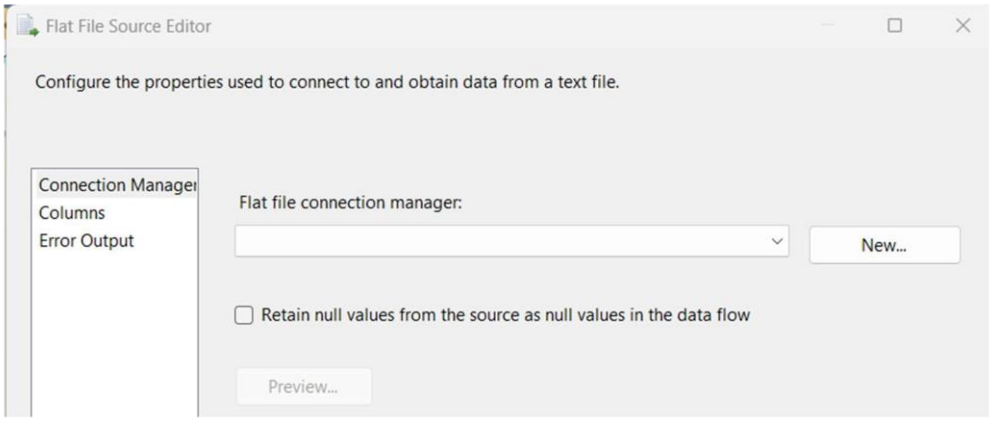
Show Editor

Name
Specifies the name of the object.

- Double-click on it and drag and drop the Flat File source into the flow.



- Double-click on it, and Click on New.

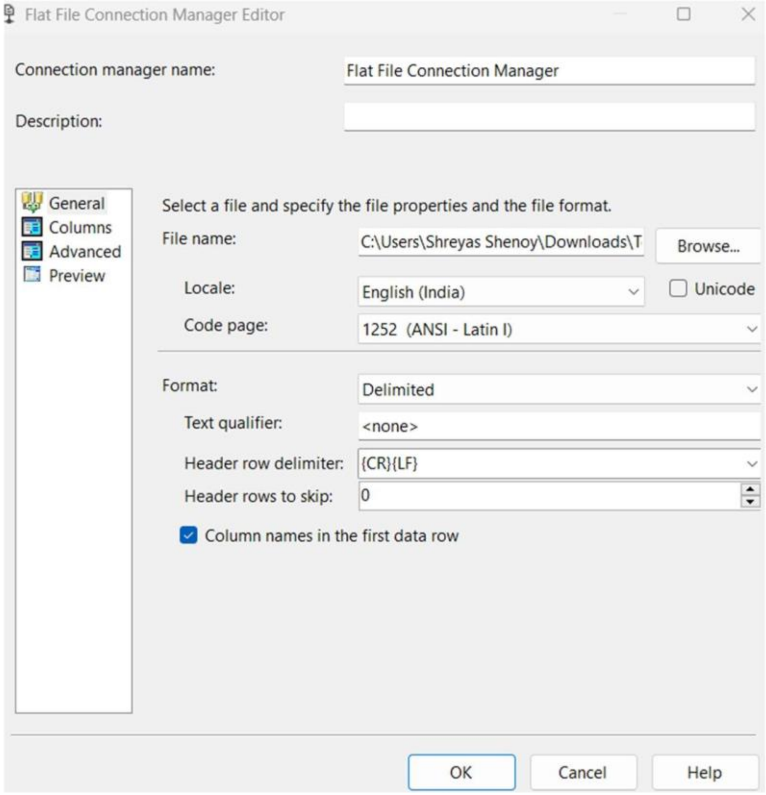


Configure the properties used to connect to and obtain data from a text file.

Flat file connection manager:

☐ Retain null values from the source as null values in the data flow

- Browse the file from your local system and click on ok.



Connection manager name: Flat File Connection Manager

Description:

Select a file and specify the file properties and the file format.

File name: C:\Users\Shreyas Shenoy\Downloads\T

Locale: English (India)

☐ Unicode

Code page: 1252 (ANSI - Latin I)

Format: Delimited

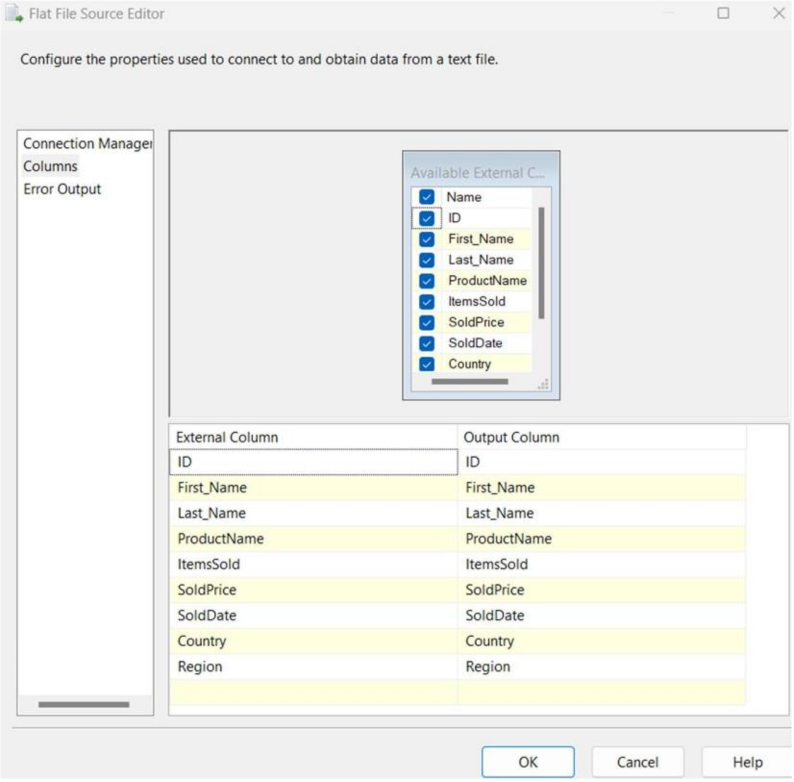
Text qualifier: <none>

Header row delimiter: {CR}{LF}

Header rows to skip: 0

☑ Column names in the first data row

- Next check the columns and click on OK.

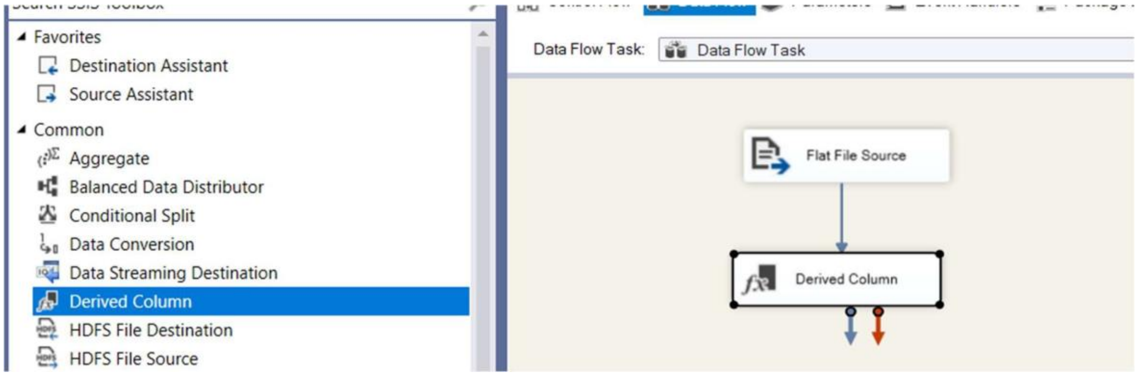


Configure the properties used to connect to and obtain data from a text file.

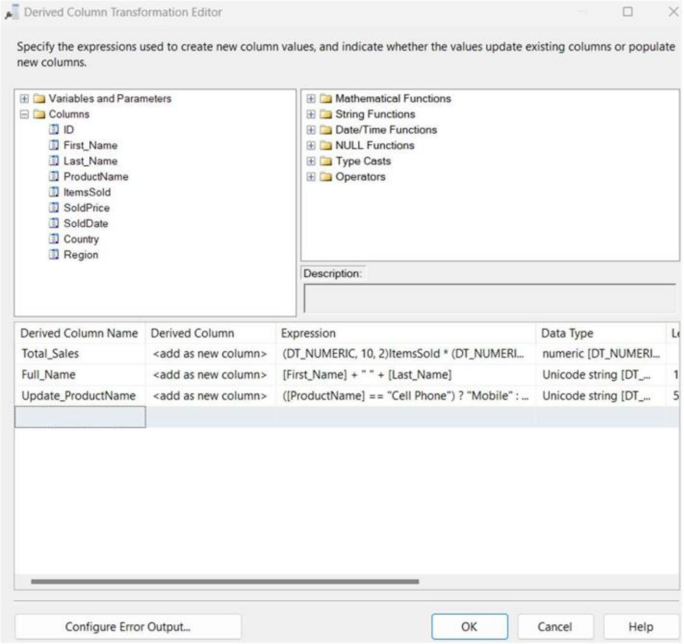
- ☑ Name
- ☑ ID
- ☑ First_Name
- ☑ Last_Name
- ☑ ProductName
- ☑ ItemsSold
- ☑ SoldPrice
- ☑ SoldDate
- ☑ Country

External Column	Output Column
ID	ID
First_Name	First_Name
Last_Name	Last_Name
ProductName	ProductName
ItemsSold	ItemsSold
SoldPrice	SoldPrice
SoldDate	SoldDate
Country	Country
Region	Region

- Drag and drop the Derived column and make the connection as shown below.



- Double-click on it. Add three columns as new and give the below expressions. Total_Sales: (DT_NUMERIC, 10, 2)ItemsSold * (DT_NUMERIC, 10, 2)SoldPrice Full_Name: [First_Name] + " " + [Last_Name]
    - Update_ProductName: ([ProductName] == "Cell Phone") ? "Mobile" : [ProductName]



Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type	Le
Total_Sales	<add as new column>	(DT_NUMERIC, 10, 2)ItemsSold * (DT_NUMERI...	numeric [DT_NUMERI...	
Full_Name	<add as new column>	[First_Name] + " " + [Last_Name]	Unicode string [DT_...	1
Update_ProductName	<add as new column>	([ProductName] == "Cell Phone") ? "Mobile" : ...	Unicode string [DT_...	5

- Add Lookup Transformation (For Region)

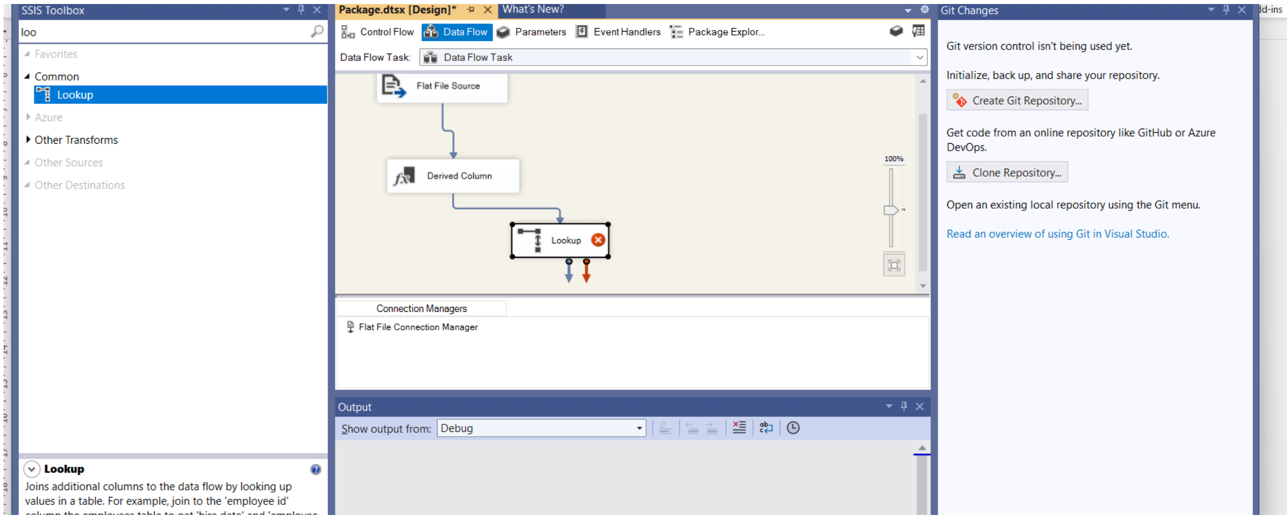
- Add Lookup Transformation:
- Drag a Lookup Transformation into the designer and connect it to the output of the Derived Column.

- In the Lookup transformation editor, choose to perform a lookup based on the Country (e.g., to lookup missing Region information).

Configure Lookup:

- Choose an OLE DB Connection or another file source where your lookup table is stored.
- Map the Country column from your source to your lookup reference.
- Map the resulting Region column back into the data flow.

Output: This will ensure that all records have the correct Region



Lookup

Joins additional columns to the data flow by looking up values in a table. For example, join to the 'employee id'

Git version control isn't being used yet.

Initialize, back up, and share your repository.

Get code from an online repository like GitHub or Azure DevOps.

Open an existing local repository using the Git menu.

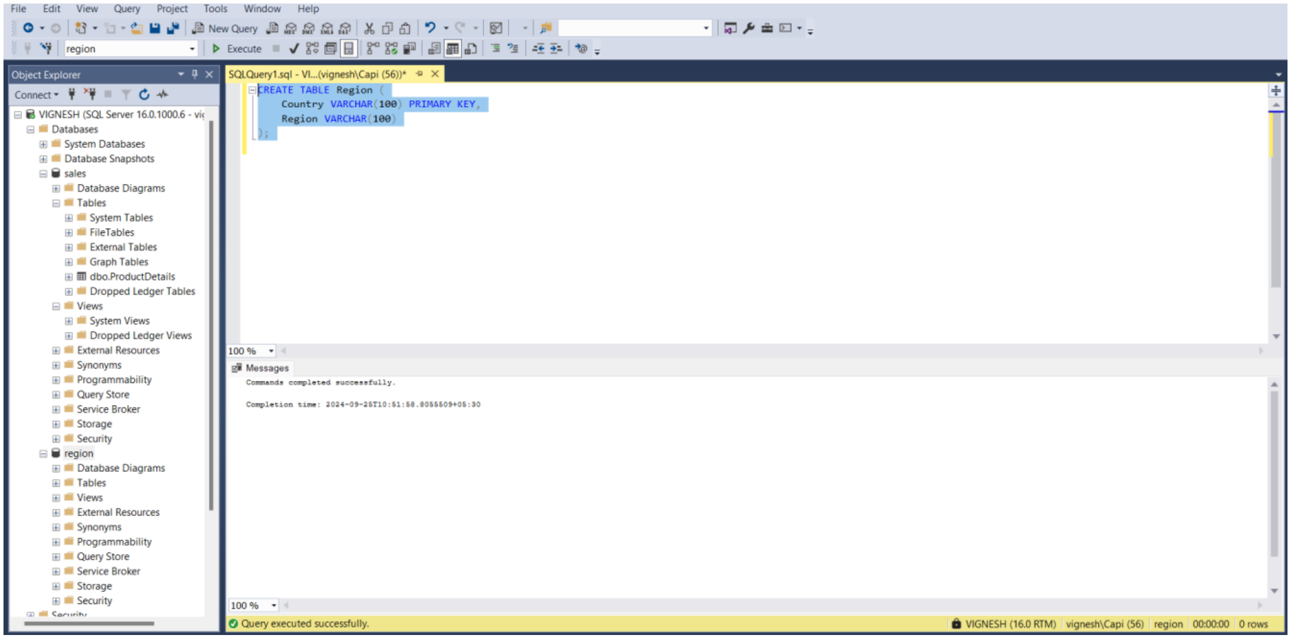
In the SSMS you need to create database name region

Use database Region

```
CREATE TABLE Region (
    Country VARCHAR(100) PRIMARY KEY,
    Region VARCHAR(100)
);


INSERT INTO Region (Country, Region)
VALUES
    ('USA', 'North America'),
    ('Pakistan', 'Asia'),
    ('India', 'Asia'),
    ('France', 'Europe'),
```

('Germany', 'Europe');

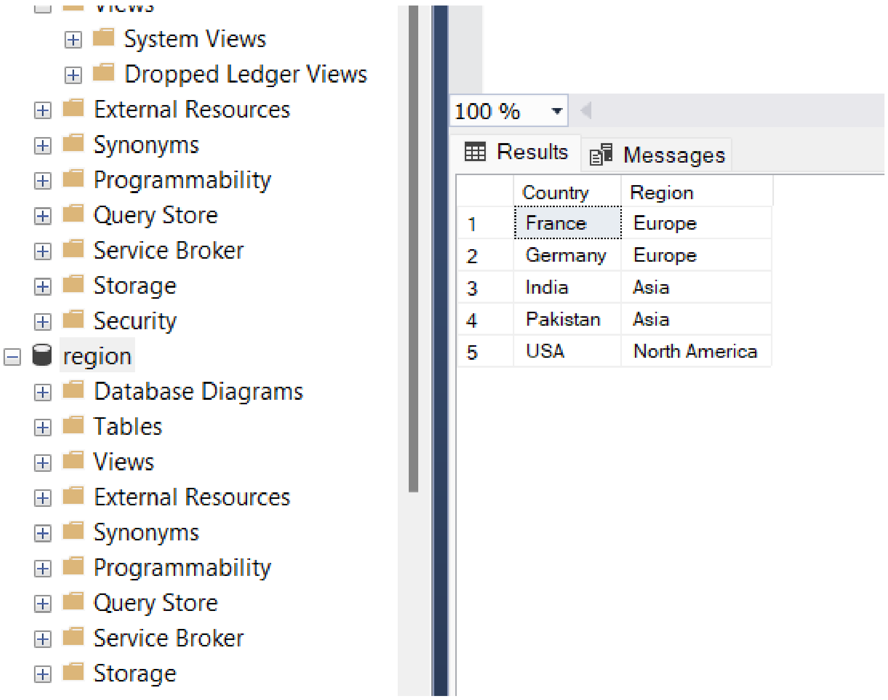


```
CREATE TABLE Region (
    Country VARCHAR(100) PRIMARY KEY,
    Region VARCHAR(100)
);
```

```
Commands completed successfully.

Completion time: 2024-09-25T10:51:58.8055509+05:30
```

Query executed successfully. VIGNESH (16.0 RTM) | vignesh\Capi (56) | region | 00:00:00 | 0 rows



	Country	Region
1	France	Europe
2	Germany	Europe
3	India	Asia
4	Pakistan	Asia
5	USA	North America

- Select the appropriate table or reference file where your Country to Region mapping is stored (e.g., CountryRegionReference).

Configure Lookup Columns:

- Go to the Columns tab in the Lookup editor.
- Under Available Input Columns, check the Country column from your input (main data flow).
- Under Available Lookup Columns, check the Region column from your reference table (CountryRegionReference).

Set the Lookup Matching Condition:

- In the same Columns tab, set the Lookup Condition:
    - Map the Country from your input columns to Country in the lookup reference table.
    - This will ensure the Lookup matches based on the Country.

Handle Lookup Failures:

- In the General tab, configure what should happen if a lookup fails:
    - Redirect rows to a failure output: If no match is found, you can send these rows to a different path for further handling.
    - Use a default value: For instance, you can use a default region if no matching record is found in the lookup table.

11)Map the Resulting Region into Data Flow

Output the Region Column:

- After configuring the lookup, the Region column from the reference table will now be available as part of the data flow.
- In the Columns tab, make sure the Region column is checked for inclusion in the output.

Replace or Add the Region Column:

- You can decide whether to replace the existing Region column (if one exists in the input) or add the Region column from the lookup as a new column.
    - If there is an existing Region column, rename the new Region column from the lookup in the Derived Column transformation or within the Mapping in the destination.
- If the existing Region column in your source is empty or incomplete, you can use the Lookup Region column to fill in the missing data.
- In the Derived Column transformation, you can add logic like:

ISNULL([Region]) ? [Lookup_Region] : [Region]

SSIS Lookup Transformation for Country to Region

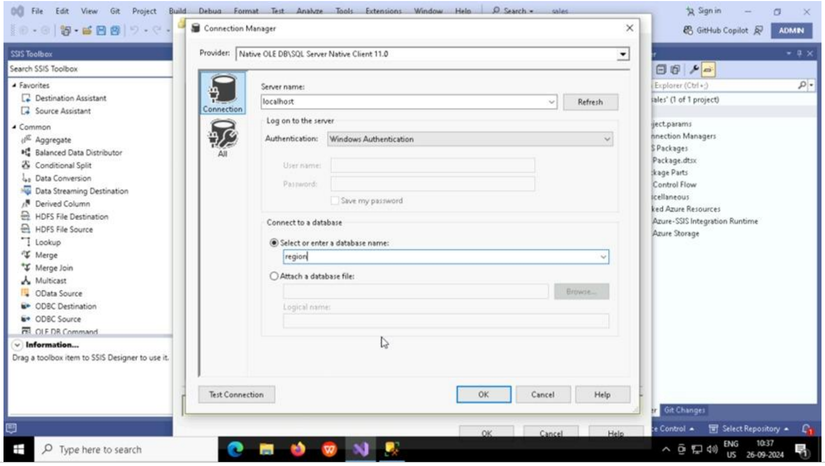
Example Input Data Before Lookup:

ID	First_Name	Last_Name	ProductName	ItemsSold	SoldPrice	Country	Region (incomplete)
1	Aamir	Shahzad	TV	1	700	USA	NULL
2	M	Raza	Cell Phone	2	800	USA	NULL
3	Christy	Ladson	TV	3	1600	USA	NULL
4	John	Rivers	Laptop	5	2400	USA	NULL
5	Najaf	Ali	Computer	1	300	Pakistan	NULL

Example Output After Lookup Transformation:

ID	First_Name	Last_Name	ProductName	ItemsSold	SoldPrice	Country	Region
1	Aamir	Shahzad	TV	1	700	USA	North America
2	M	Raza	Mobiles	2	800	USA	North America
3	Christy	Ladson	TV	3	1600	USA	North America
4	John	Rivers	Laptop	5	2400	USA	North America
5	Najaf	Ali	Computer	1	300	Pakistan	Asia

To connect to the database you use the server name localhost and select windows authentication give the database as “**region** “



Connection Manager

Provider: Native OLE DB\SQL Server Native Client 11.0

Server name: localhost

Log on to the server

Authentication: Windows Authentication

User name:

Password:

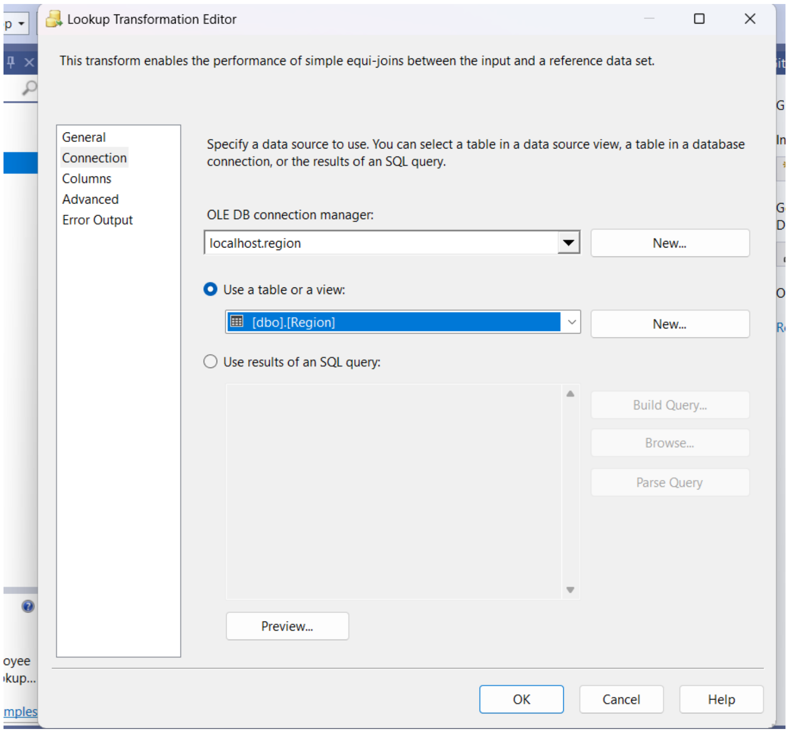
□ Save my password

Connect to a database

● Select or enter a database name: region

○ Attach a database file:

Logical name:



Lookup Transformation Editor

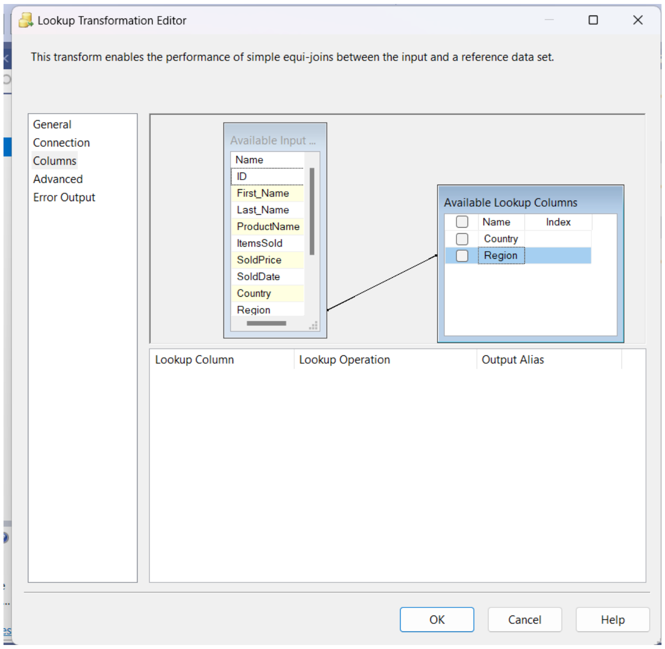
This transform enables the performance of simple equi-joins between the input and a reference data set.

Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.

OLE DB connection manager: localhost.region

● Use a table or a view: [dbo].[Region]

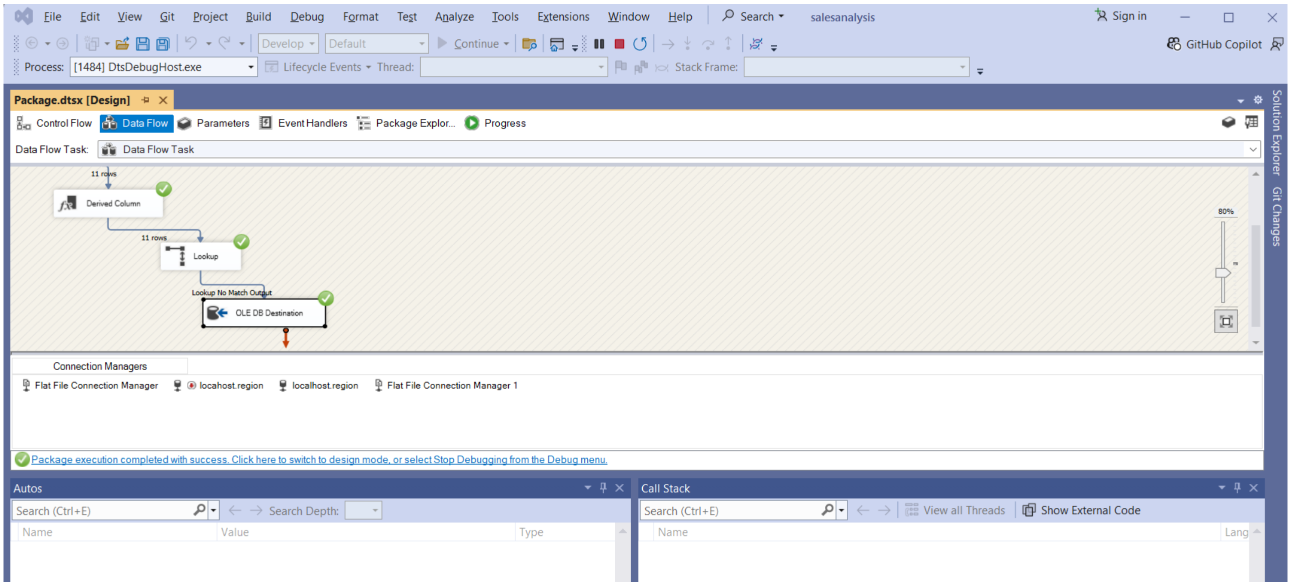
○ Use results of an SQL query:



Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

Verify the lookup data in the table after the lookup is performed



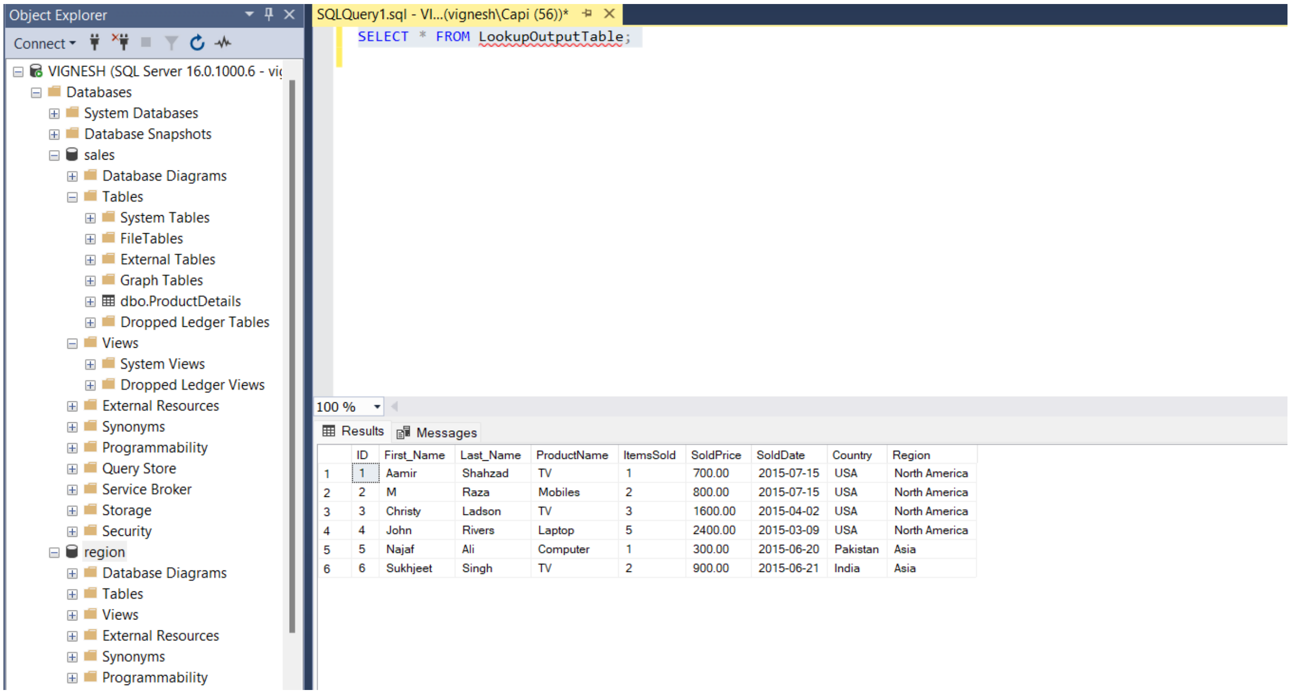
Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Add an OLE DB Destination:

- After your Lookup Transformation, drag an OLE DB Destination into the data flow.
- Connect the output of the Lookup to the OLE DB Destination.

Configure OLE DB Destination:

- Select the LookupOutputTable as the destination table.
- Map the columns from the Lookup Transformation output to the respective fields in the LookupOutputTable.

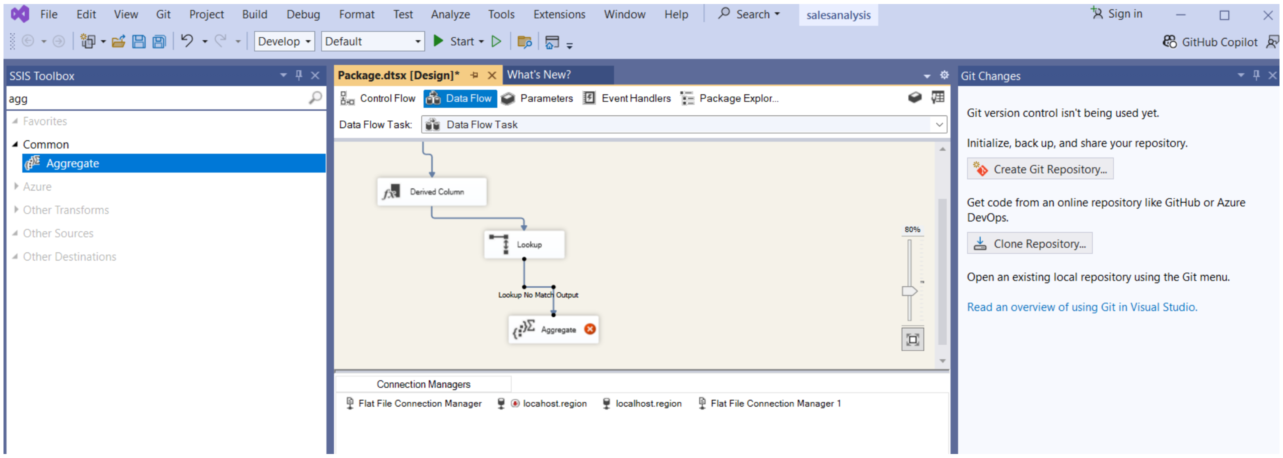


```sql
SELECT * FROM LookupOutputTable;
```

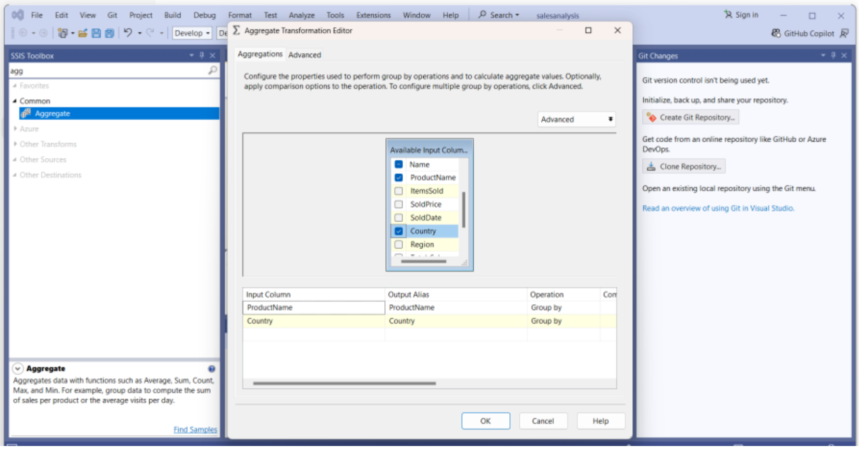
	ID	First_Name	Last_Name	ProductName	ItemsSold	SoldPrice	SoldDate	Country	Region
1	1	Aamir	Shahzad	TV	1	700.00	2015-07-15	USA	North America
2	2	M	Raza	Mobiles	2	800.00	2015-07-15	USA	North America
3	3	Christy	Ladson	TV	3	1600.00	2015-04-02	USA	North America
4	4	John	Rivers	Laptop	5	2400.00	2015-03-09	USA	North America
5	5	Najaf	Ali	Computer	1	300.00	2015-06-20	Pakistan	Asia
6	6	Sukhjeet	Singh	TV	2	900.00	2015-06-21	India	Asia

Add Aggregate Transformation (For Totaling Sales)

Add Aggregate Transformation:



- Drag an Aggregate Transformation onto the designer and connect it to the Lookup Transformation.
- In the Aggregate Transformation, group by:
  - ProductName
  - Country



Aggregate Transformation Editor

Aggregations | Advanced

Configure the properties used to perform group by operations and to calculate aggregate values. Optionally, apply comparison options to the operation. To configure multiple group by operations, click Advanced.

Advanced

Available Input Colum...

- ☑ Name
- ☑ ProductName
- ☐ ItemsSold
- ☐ SoldPrice
- ☐ SoldDate
- ☑ Country
- ☐ Region

Input Column	Output Alias	Operation	Com
ProductName	ProductName	Group by	
Country	Country	Group by	

Aggregate

Aggregates data with functions such as Average, Sum, Count, Max, and Min. For example, group data to compute the sum of sales per product or the average visits per day.

Find Samples

Git version control isn't being used yet.

Initialize, back up, and share your repository.

Get code from an online repository like GitHub or Azure DevOps.

Open an existing local repository using the Git menu.

Read an overview of using Git in Visual Studio.

- Add the following operations:
  - Total_Items_Sold: Sum the ItemsSold.
  - Total_Sales: Sum the calculated Total_Sales.

Output: This step calculates the total sales and items sold for each product by country

12 Add Conditional Split (For Region-Based Separation)

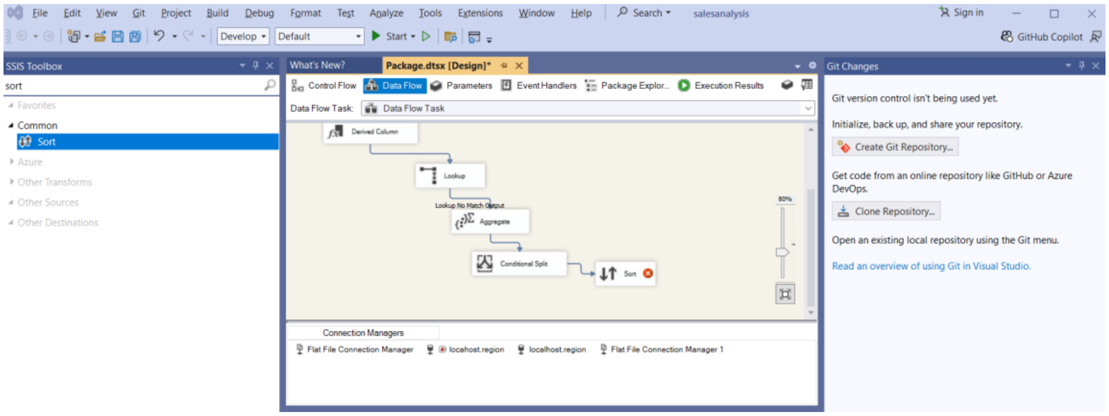
Add Conditional Split Transformation:

- Drag a Conditional Split transformation onto the designer and connect it to the output of the Aggregate Transformation.

Configure Conditional Split:

- Add conditions to split the data based on the Region:
  - Region == "North America" -> North America Output
  - Region == "Asia" -> Asia Output
  - Region == "Europe" -> Europe Output

Drag and drop the Conditional Split and make the connection.



Git version control isn't being used yet.

Initialize, back up, and share your repository.

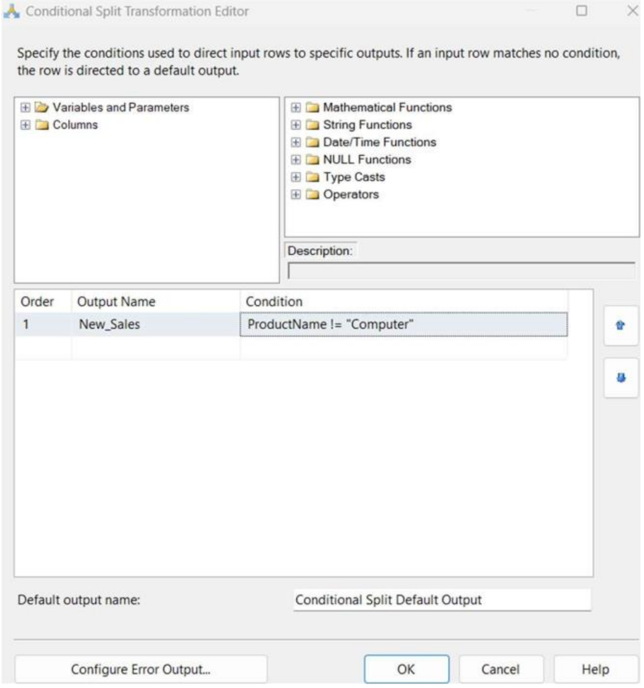
Get code from an online repository like GitHub or Azure DevOps.

Open an existing local repository using the Git menu.

Read an overview of using Git in Visual Studio.

Double-click on it and give an output and condition as shown below

. Exp: ProductName != "Computer"



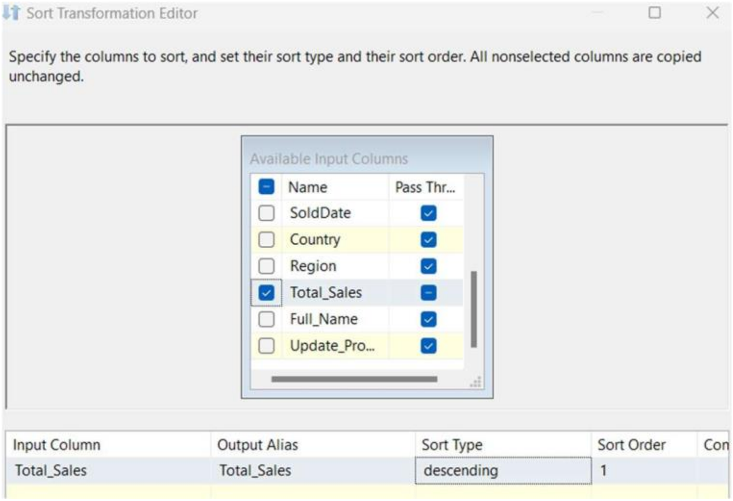
Specify the conditions used to direct input rows to specific outputs. If an input row matches no condition, the row is directed to a default output.

Order	Output Name	Condition
1	New_Sales	ProductName != "Computer"

Default output name: Conditional Split Default Output

- Drag and drop the Sort and make the connection as shown below.

    - Double-click on it and set the condition.



Sort Transformation Editor

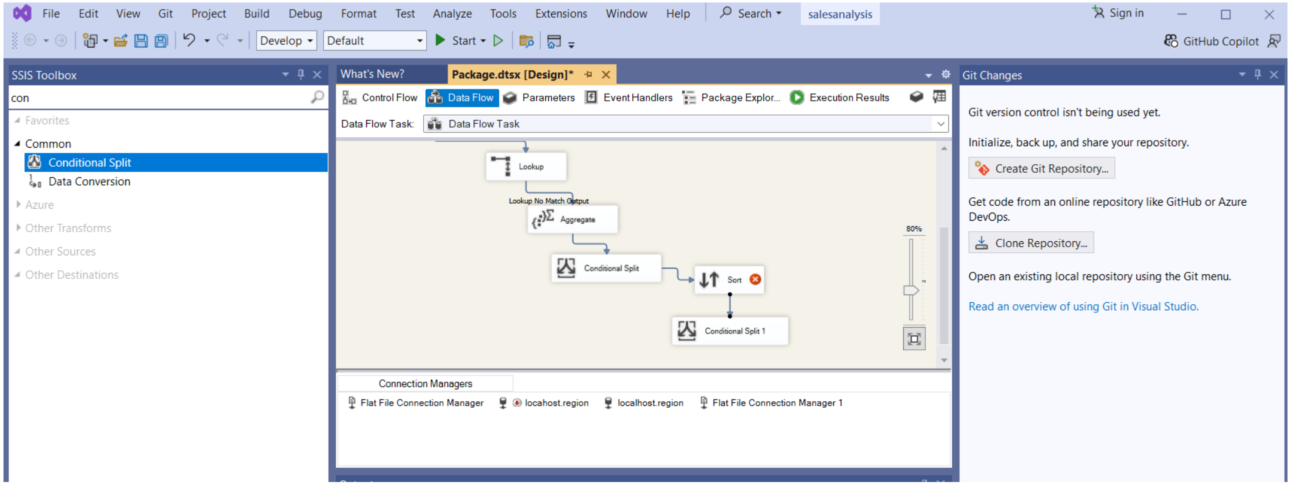
Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.

Available Input Columns

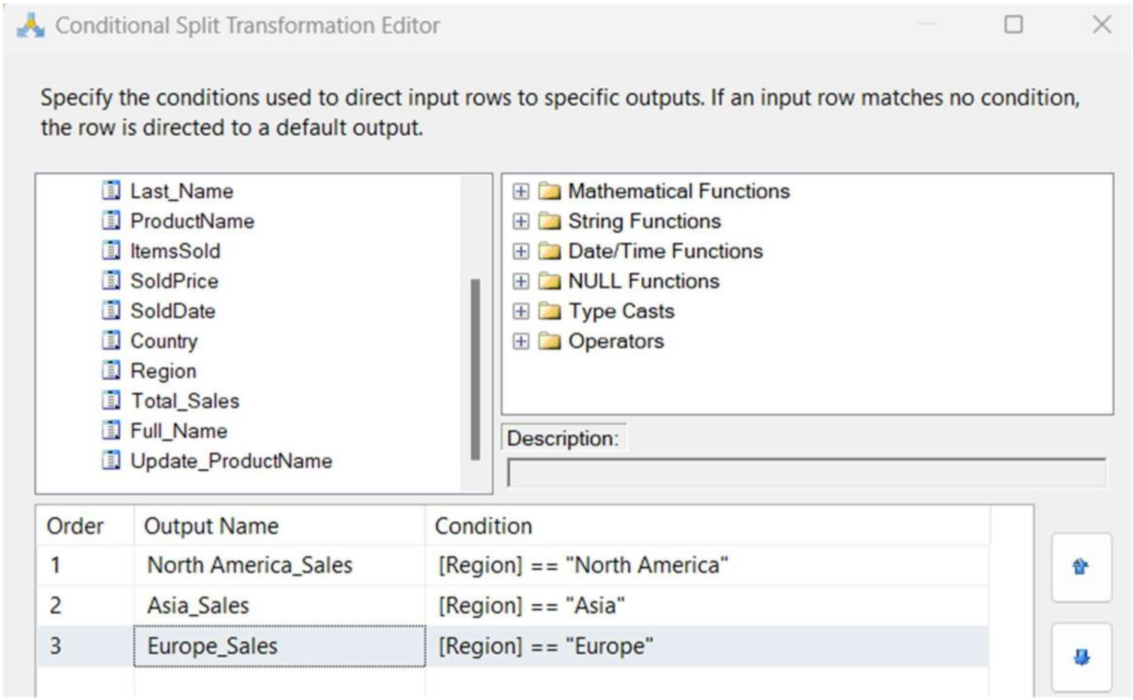
☑ Name	Pass Thr...
☐ SoldDate	☑
☐ Country	☑
☐ Region	☑
☑ Total_Sales	☑
☐ Full_Name	☑
☐ Update_Pro...	☑

Input Column	Output Alias	Sort Type	Sort Order	Con
Total_Sales	Total_Sales	descending	1	

Drag and drop the conditional split again and make the connection as shown below.



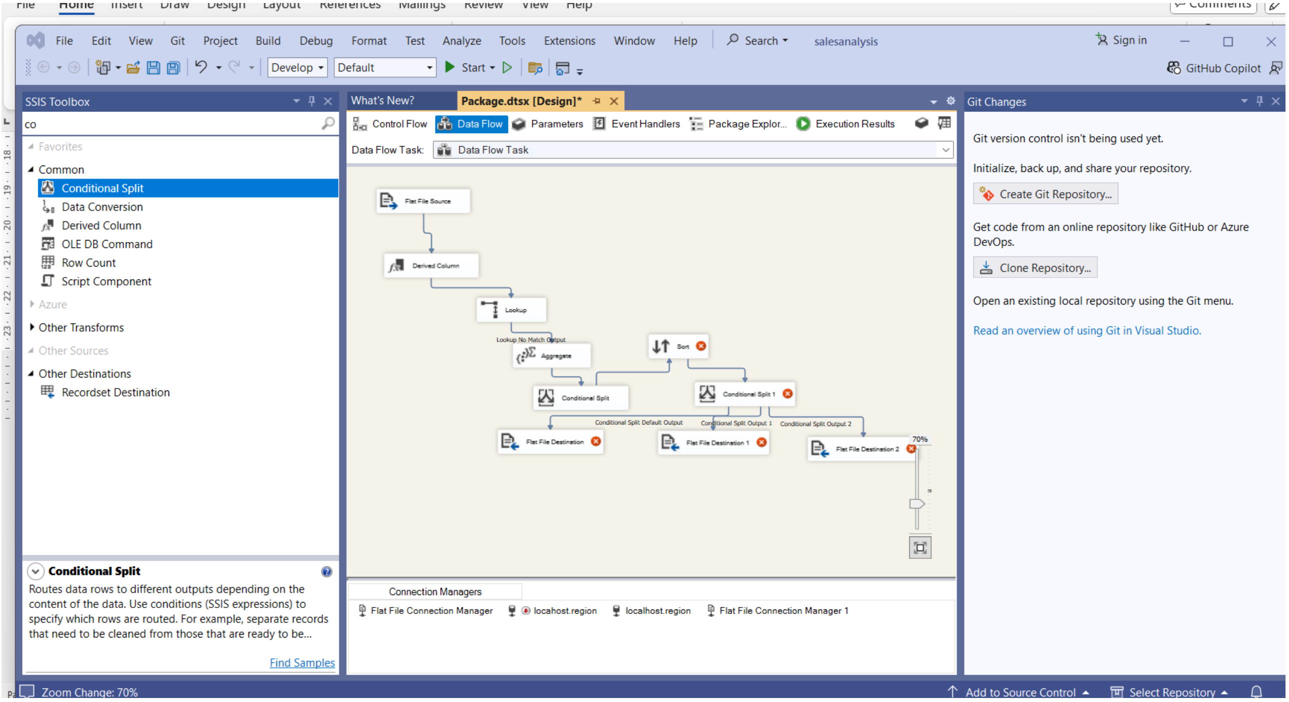
Double-click on it and add below three output conditions.



Specify the conditions used to direct input rows to specific outputs. If an input row matches no condition, the row is directed to a default output.

Order	Output Name	Condition
1	North America_Sales	[Region] == "North America"
2	Asia_Sales	[Region] == "Asia"
3	Europe_Sales	[Region] == "Europe"

- Drag and drop three flat file destinations and make connections as shown below.



Conditional Split

Routes data rows to different outputs depending on the content of the data. Use conditions (SSIS expressions) to specify which rows are routed. For example, separate records that need to be cleaned from those that are ready to be...

Git version control isn't being used yet.

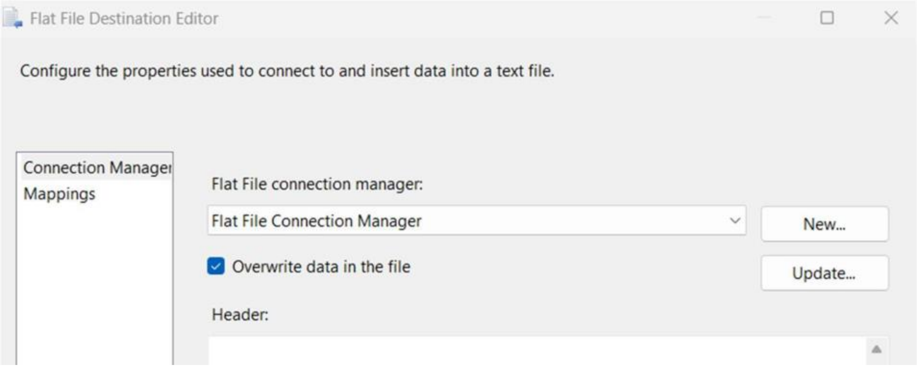
Initialize, back up, and share your repository.

Get code from an online repository like GitHub or Azure DevOps.

Open an existing local repository using the Git menu.

Read an overview of using Git in Visual Studio.

- Double-click on the first destination and click on New.



Flat File Destination Editor

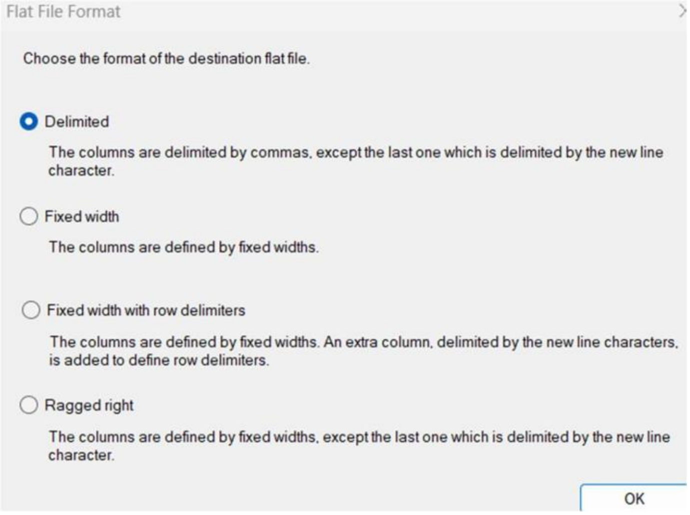
Configure the properties used to connect to and insert data into a text file.

Flat File connection manager: Flat File Connection Manager

☑ Overwrite data in the file

Header:

- Click on OK.



Flat File Format

Choose the format of the destination flat file.

● Delimited

The columns are delimited by commas, except the last one which is delimited by the new line character.

○ Fixed width

The columns are defined by fixed widths.

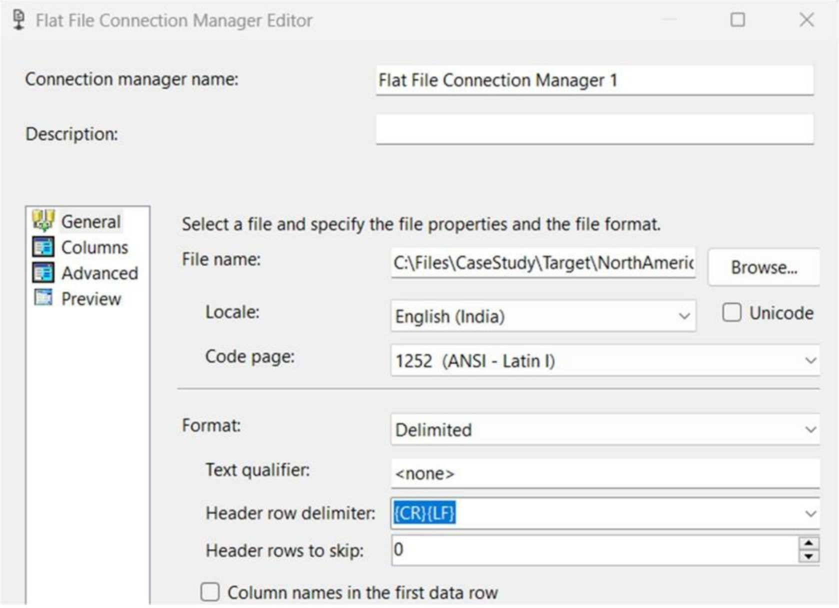
○ Fixed width with row delimiters

The columns are defined by fixed widths. An extra column, delimited by the new line characters, is added to define row delimiters.

○ Ragged right

The columns are defined by fixed widths, except the last one which is delimited by the new line character.

- Create a File in your local system with the North American sales.



Connection manager name: Flat File Connection Manager 1

Description:

Select a file and specify the file properties and the file format.

File name: C:\Files\CaseStudy\Target\NorthAmeric

Locale: English (India)

□ Unicode

Code page: 1252 (ANSI - Latin I)

Format: Delimited

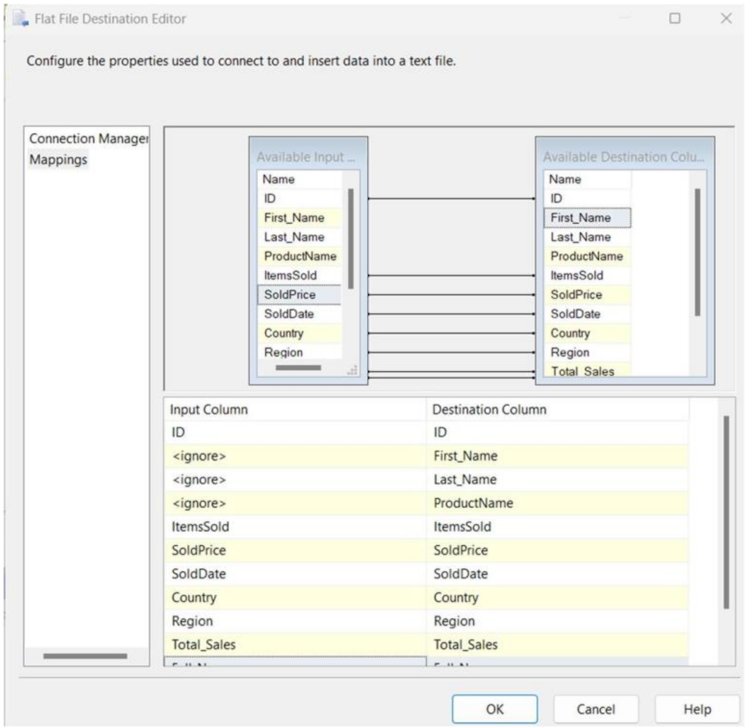
Text qualifier: <none>

Header row delimiter: {CR}{LF}

Header rows to skip: 0

□ Column names in the first data row

- Under Advance delete the First Name, Last Name, and Product Name columns.
- Click OK. Under Mapping Check the columns.

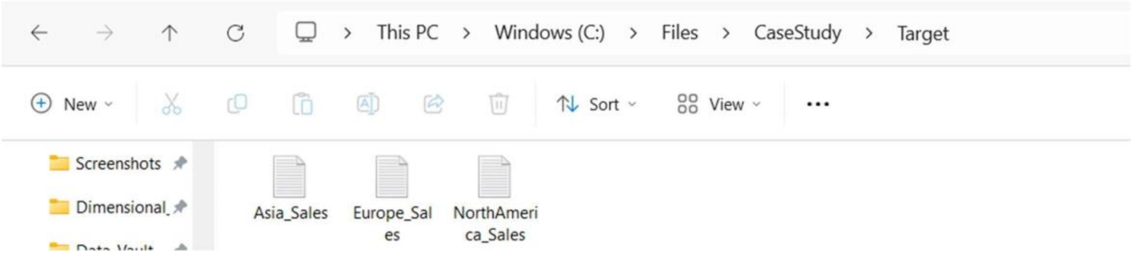


Configure the properties used to connect to and insert data into a text file.

Input Column	Destination Column
ID	ID
<ignore>	First_Name
<ignore>	Last_Name
<ignore>	ProductName
ItemsSold	ItemsSold
SoldPrice	SoldPrice
SoldDate	SoldDate
Country	Country
Region	Region
Total_Sales	Total_Sales

- Do the Same for Other Destinations also.
- Next save the package and click on Run.
- Here our flow was executed successfully.

- Here we have three text files created.



- Check the data in those files. Export all the table you have created in the SSMS
- Submit the work to the github repository with the proper document