

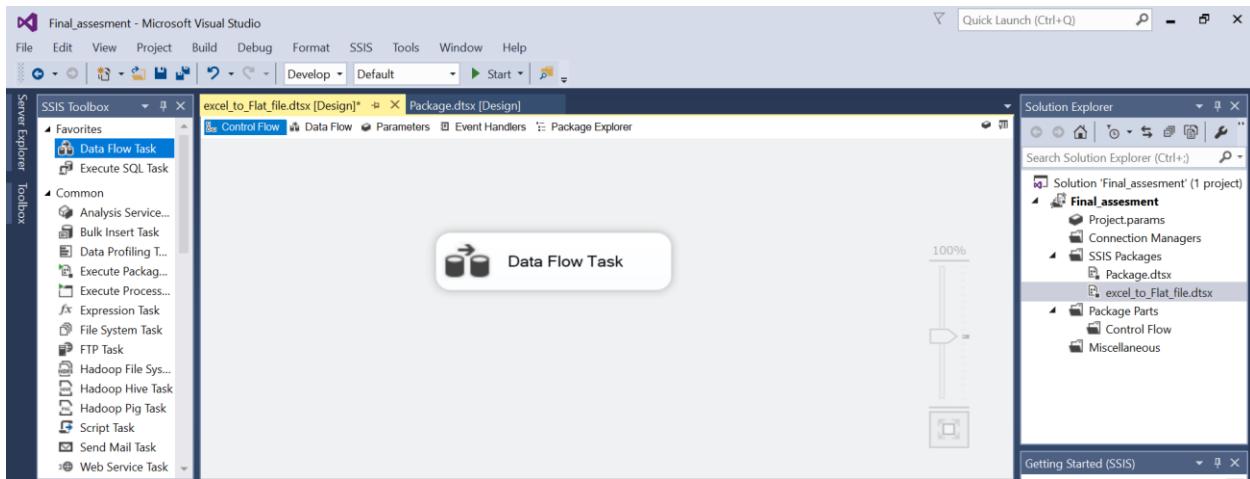
Module 2 SSIS

Qn1. Make connection to different data source

Solution:

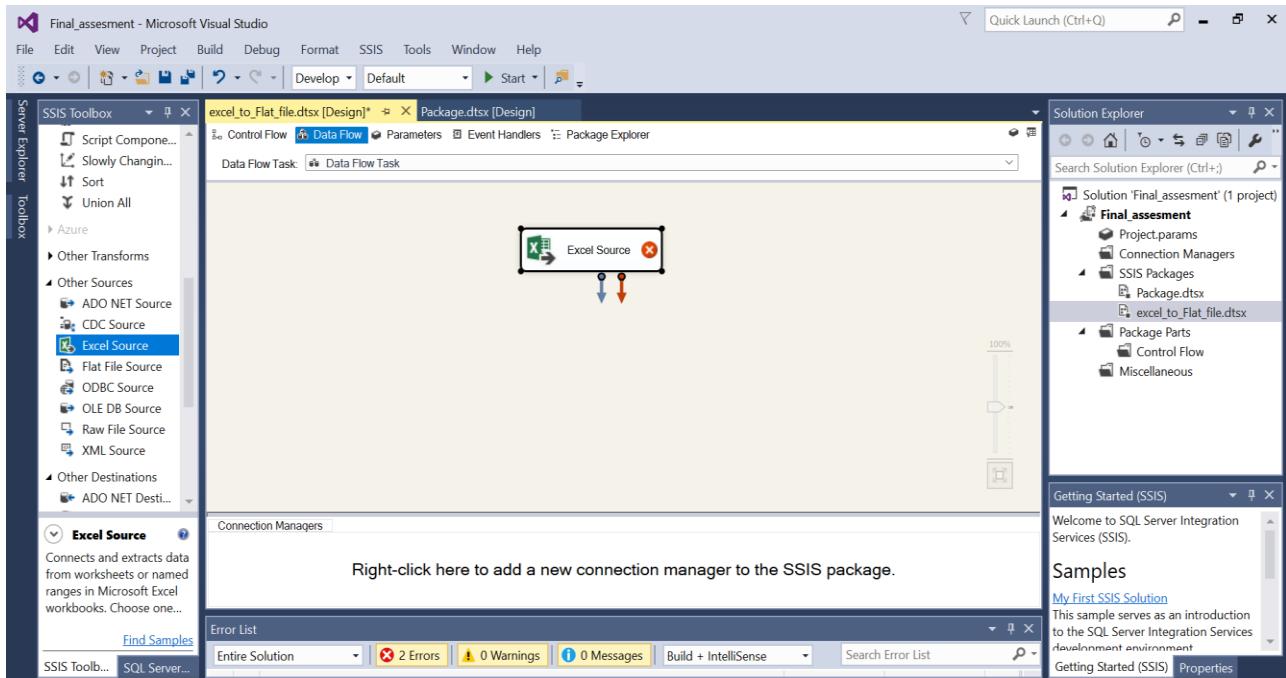
1.Extractiong the data from excel source, loading into the FLAT file

Step1: drag and drop the data flow task from ssis tool box



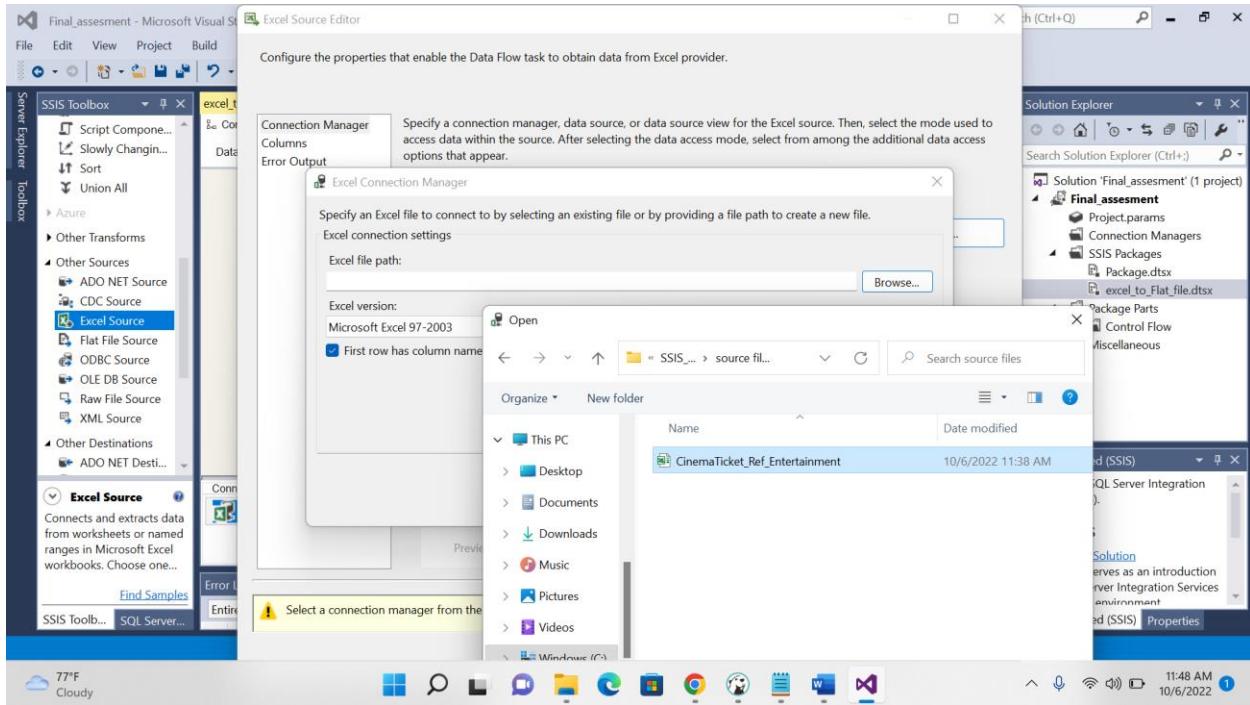
Step2: Double click on the data flow task

Step3: drag and drop the Excel source from the ssis tool box

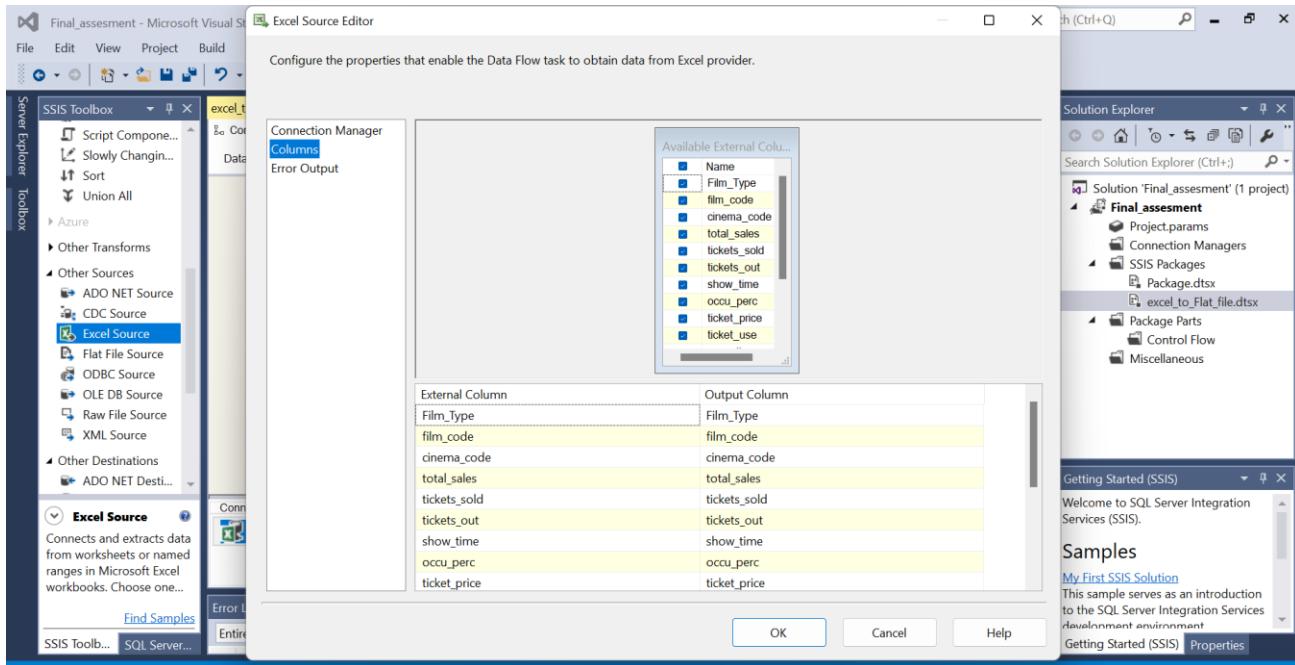


Step3: Double click on excel source we will get new window click on new.

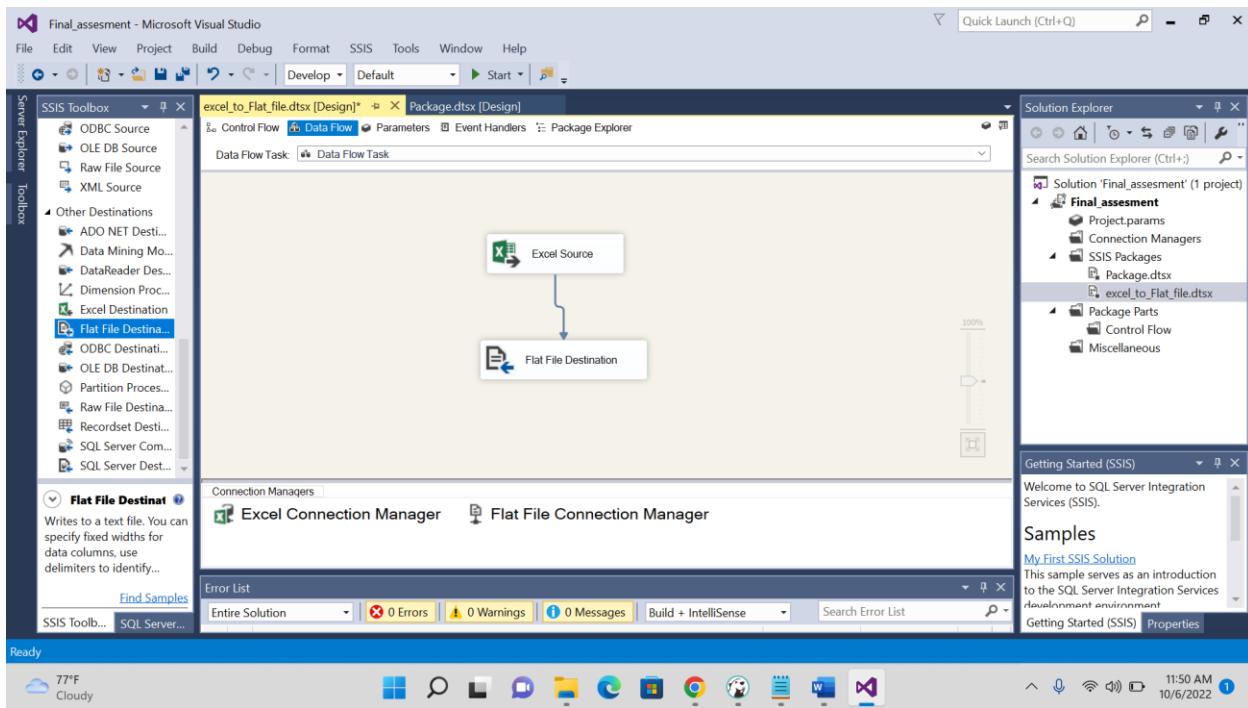
Step4: Browse for the file in local computer and select the file



Step 5: check the columns and data in the excel file

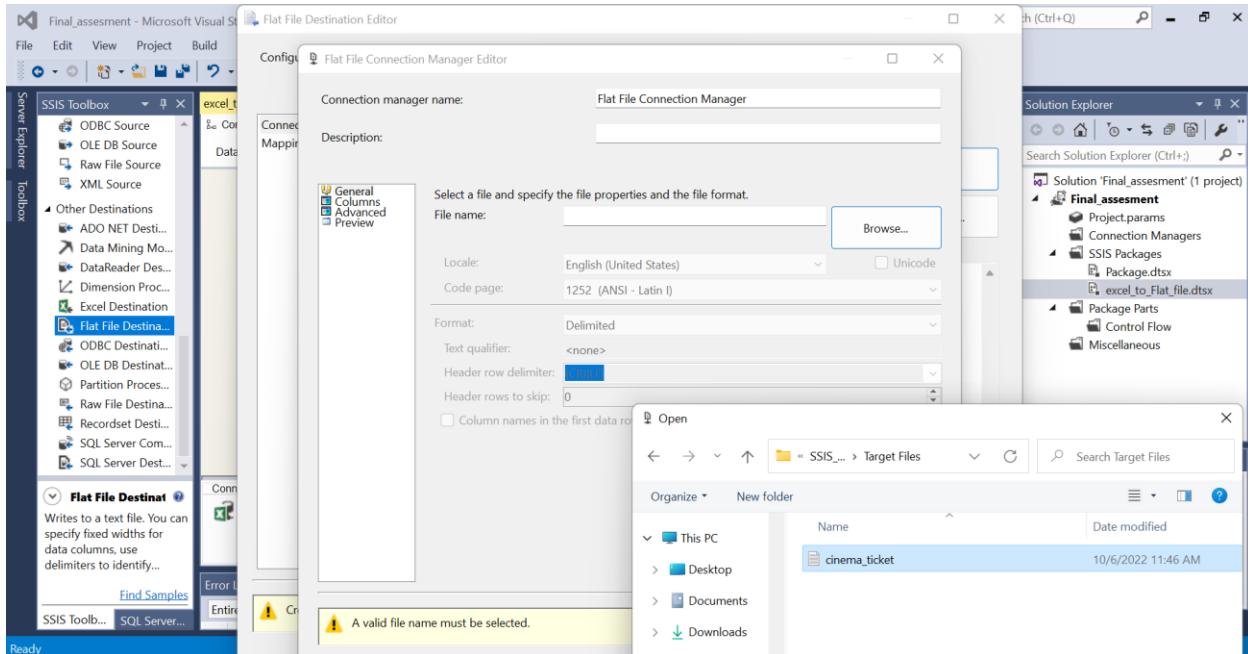


Step6: drag and drop the flat file Destination then connect to the source



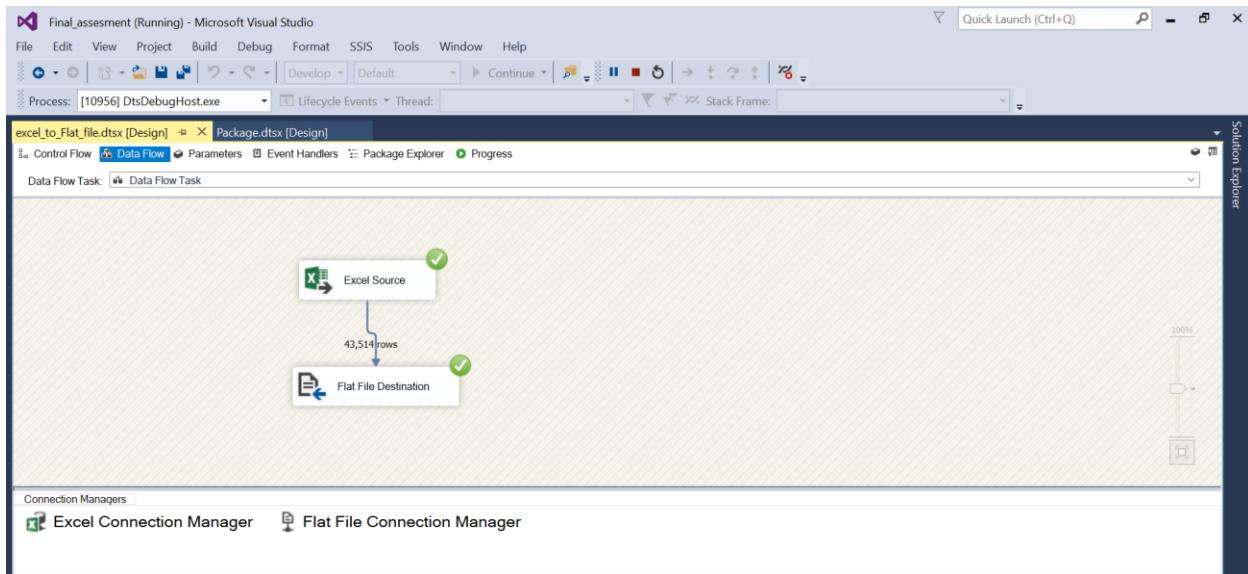
Step7: double click on the flat file destination

Step8: click on new and select the empty flat file



Step9: run the package

Result of successful execution



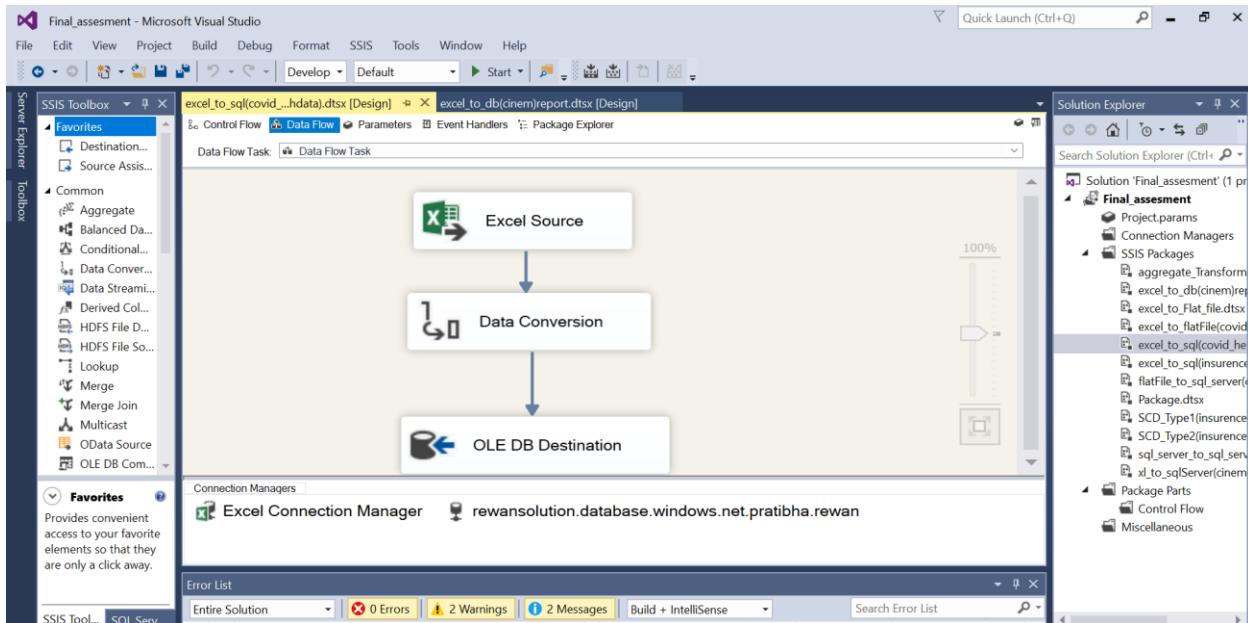
Below snapshot shows the data loaded in the flat file

```
cinema_ticket - Notepad
File Edit View
Film_Type,film_code,cinema_code,total_sales,tickets_sold,tickets_out,show_time,occu_perc,ticket_price,ticket_u
Romance,1492,304,3900000,26,0,4,4,2599999999999998,150000,26,610,5/5/2018,5,2,5
Romance,1492,352,3360000,42,0,5,8,0800000000000001,80000,42,520,5/5/2018,5,2,5
Romance,1492,489,2560000,32,0,4,20,80000,32,160,5/5/2018,5,2,5
Romance,1492,429,1200000,12,0,1,11,01,100000,12,109,5/5/2018,5,2,5
Romance,1492,524,1200000,15,0,3,16,6700000000000002,80000,15,90,5/5/2018,5,2,5
Romance,1492,71,1050000,7,0,3,0,9799999999999998,150000,7,714,5/5/2018,5,2,5
Romance,1492,163,1020000,10,0,3,7,6900000000000004,102000,10,130,5/5/2018,5,2,5
Romance,1492,450,750000,5,0,3,1,5700000000000001,150000,5,318,5/5/2018,5,2,5
Romance,1492,51,750000,11,0,2,0,9499999999999996,68181.818180000002,11,1158,5/5/2018,5,2,5
Romance,1492,522,600000,4,0,3,1,55,150000,4,258,5/5/2018,5,2,5
Romance,1492,43,480000,6,0,3,0,44,80000,6,1364,5/5/2018,5,2,5
Romance,1492,529,480000,4,0,3,2,96,120000,4,135,5/5/2018,5,2,5
Romance,1492,82,400000,5,0,6,0,5300000000000003,80000,5,943,5/5/2018,5,2,5
Romance,1492,344,300000,2,0,3,0,25,150000,2,800,5/5/2018,5,2,5
Romance,1492,73,240000,2,0,1,2,04,120000,2,98,5/5/2018,5,2,5
Romance,1492,304,16500000,112,0,4,18,32999999999998,147321.4286000001,112,611,5/4/2018,5,2,4
Romance,1492,352,13950000,93,0,5,10,57,150000,93,880,5/4/2018,5,2,4
Romance,1492,344,10200000,68,0,3,8,5399999999999991,150000,68,796,5/4/2018,5,2,4
Romance,1492,71,6600000,44,0,3,6,1399999999999997,150000,44,717,5/4/2018,5,2,4
Romance,1492,163,3360000,31,0,3,24,8000000000000001,108387.0968,31,125,5/4/2018,5,2,4
Romance,1492,522,3000000,20,0,3,7,75,150000,20,258,5/4/2018,5,2,4
Romance,1492,485,2400000,16,0,3,11,59,150000,16,138,5/4/2018,5,2,4
Romance,1492,524,1800000,12,0,3,13,33,150000,12,90,5/4/2018,5,2,4
Romance,1492,518,1680000,14,1,3,8,4800000000000004,120000,13,165,5/4/2018,5,2,4
Romance,1492,51,1400000,17,0,1,2,9300000000000002,82352.941179999994,17,580,5/4/2018,5,2,4
Romance,1492,448,1350000,9,0,2,2,3700000000000001,150000,9,380,5/4/2018,5,2,4
Romance,1492,429,1200000,12,0,1,11,01,100000,12,109,5/4/2018,5,2,4
Romance,1492,450,1050000,7,0,2,3,2999999999999998,150000,7,712,5/4/2018,5,2,4
```

The screenshot shows a Notepad window titled 'cinema_ticket - Notepad'. The content of the window displays a large amount of comma-separated data, representing the contents of a flat file. The data includes columns such as Film_Type, film_code, cinema_code, total_sales, tickets_sold, tickets_out, show_time, occu_perc, ticket_price, and ticket_u. The data spans multiple rows and columns, with some rows containing more than 10 columns. The Notepad window has standard Windows-style controls at the top and bottom.

2.Extracting the data from excel source, loading into the SQL server

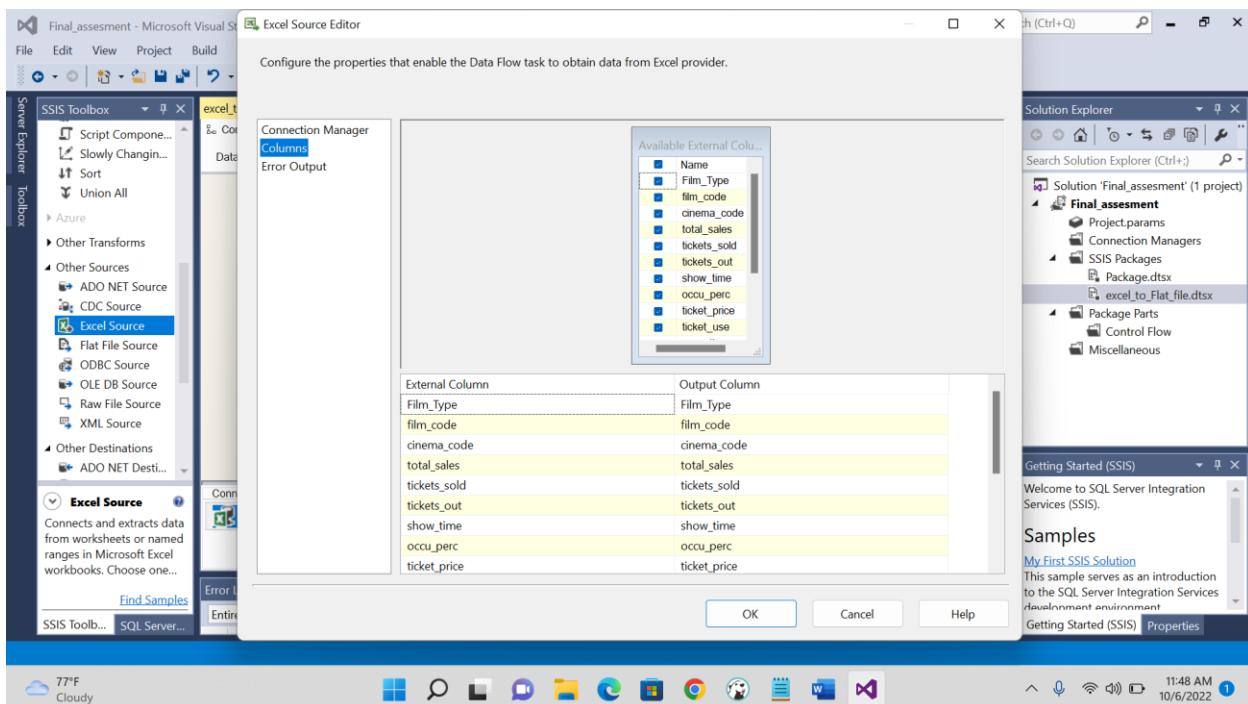
Step1: make connection ass shown in the below



Step2: double click on excel source and choose the source file

Step3: double click on the data conversion you will get below window

Step4: change the data type as per the sql server data type



Step5: create table in sql server as same as source columns

```

DBBeaver 22.2.0 - <pratibha> Final assessment
File Edit Navigate Search SQL Editor Database Window Help
Data... Proj... <Auto_> pratibha dbo@pratibha General dbo SJ_HEALTH <pratibha> Script-3 <pratibha> Final assessment <pratibha> Query
Enter a part of object
> <Auto_> - rewasolution.a
> oracle-88094-0.cloudl...
> pratibha - rewasolutio...
  > Databases
  > Security
  > Administrator
(
  Film_Type varchar(100),
  film_code varchar(100),
  cinema_code varchar(100),
  total_sales varchar(50),
  tickets_sold varchar(50),
  tickets_out varchar(50),
  show_time varchar(50),
  occu_perc varchar(100),
  ticket_price varchar(50),
  ticket_use varchar(50),
  capacity varchar(100),
  show_date varchar(50),
  show_month varchar(100),
  quarters varchar(50),
  show_day varchar(50)
)
select * from DFA_CINEMA_TICKET

```

Results 1 x

Name	Film_Type	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price	ticket_use	capacity
Record											

Save Cancel Script IST en Writable Smart Insert 99 : 1 : 3004 Sel: 0 | 0 12:33 PM 10/6/2022

Step6: Double click on the OLDB destination select the column as we have changed in data conversion

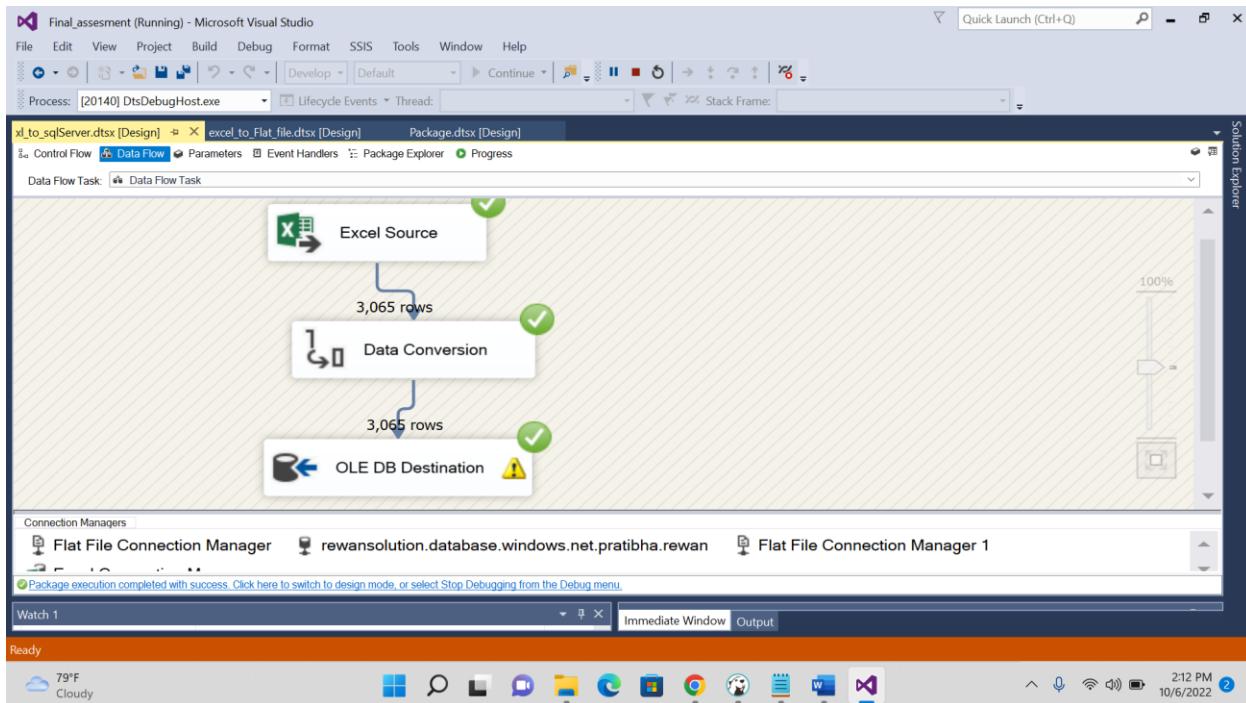
Data Conversion Transformation Editor

Configure the properties used to convert the data type of an input column to a different data type. Depending on the data type to which the column is converted, set the length, precision, scale, and code page of the column.

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
Film_Type	Copy of Film_Type	string [DT_STR]	255			1252 (ANSI - Le)
film_code	Copy of film_code	string [DT_STR]	50			1252 (ANSI - Le)
cinema_code	Copy of cinema_code	string [DT_STR]	50			1252 (ANSI - Le)
total_sales	Copy of total_sales	string [DT_STR]	50			1252 (ANSI - Le)
tickets_sold	Copy of tickets_sold	string [DT_STR]	50			1252 (ANSI - Le)
tickets_out	Copy of tickets_out	string [DT_STR]	50			1252 (ANSI - Le)
show_time	Copy of show_time	string [DT_STR]	50			1252 (ANSI - Le)
occu_perc	Copy of occu_perc	string [DT_STR]	50			1252 (ANSI - Le)
ticket_price	Copy of ticket_price	string [DT_STR]	50			1252 (ANSI - Le)
ticket_use	Copy of ticket_use	string [DT_STR]	50			1252 (ANSI - Le)
capacity	Copy of capacity	string [DT_STR]	50			1252 (ANSI - Le)
date	Copy of date	string [DT_STR]	50			1252 (ANSI - Le)

Step7: Run the package

Below snapshot is for execution of package



Below snapshot shows the data loaded in to the sql server

The screenshot shows the DBeaver interface with a SQL Editor window displaying the following query and results:

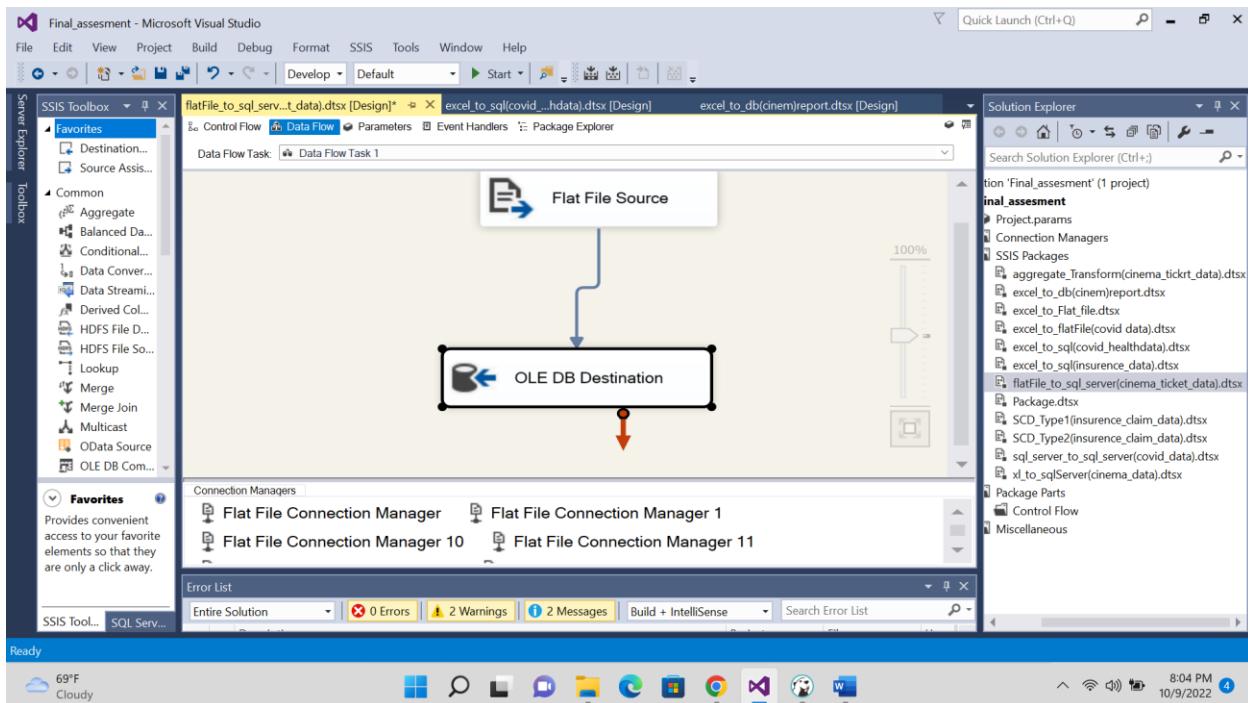
```
select * from DFA_CINEMA_TICKET
```

The results grid shows 21 rows of data:

	Film_Type	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price	ticket_use
1	Romance	1492	304	3900000	26	0	4	4.259999999999999	150000	26
2	Romance	1492	352	3360000	42	0	5	8.080000000000001	80000	42
3	Romance	1492	489	2560000	32	0	4	20	80000	32
4	Romance	1492	429	1200000	12	0	1	11.01	100000	12
5	Romance	1492	524	1200000	15	0	3	16.670000000000002	80000	15
6	Romance	1492	71	1050000	7	0	3	0.979999999999998	150000	7
7	Romance	1492	163	1020000	10	0	3	7.690000000000004	102000	10
8	Romance	1492	450	750000	5	0	3	1.570000000000001	150000	5
9	Romance	1492	51	750000	11	0	2	0.949999999999996	68181.81818000002	11
10	Romance	1492	522	600000	4	0	3	1.55	150000	4
11	Romance	1492	43	480000	6	0	3	0.44	80000	6
12	Romance	1492	529	480000	4	0	3	2.96	120000	4
13	Romance	1492	82	400000	5	0	6	0.530000000000003	80000	5
14	Romance	1492	344	300000	2	0	3	0.25	150000	2
15	Romance	1492	73	240000	2	0	1	2.04	120000	2
16	Romance	1492	304	1650000	112	0	4	18.32999999999998	147321.4286000001	112
17	Romance	1492	352	1395000	93	0	5	10.57	150000	93
18	Romance	1492	344	1020000	68	0	3	8.539999999999991	150000	68
19	Romance	1492	71	660000	44	0	3	6.139999999999997	150000	44
20	Romance	1492	163	336000	31	0	3	24.800000000000001	108387.0968	31
21	Romance	1492	522	300000	20	0	3	7.75	150000	20

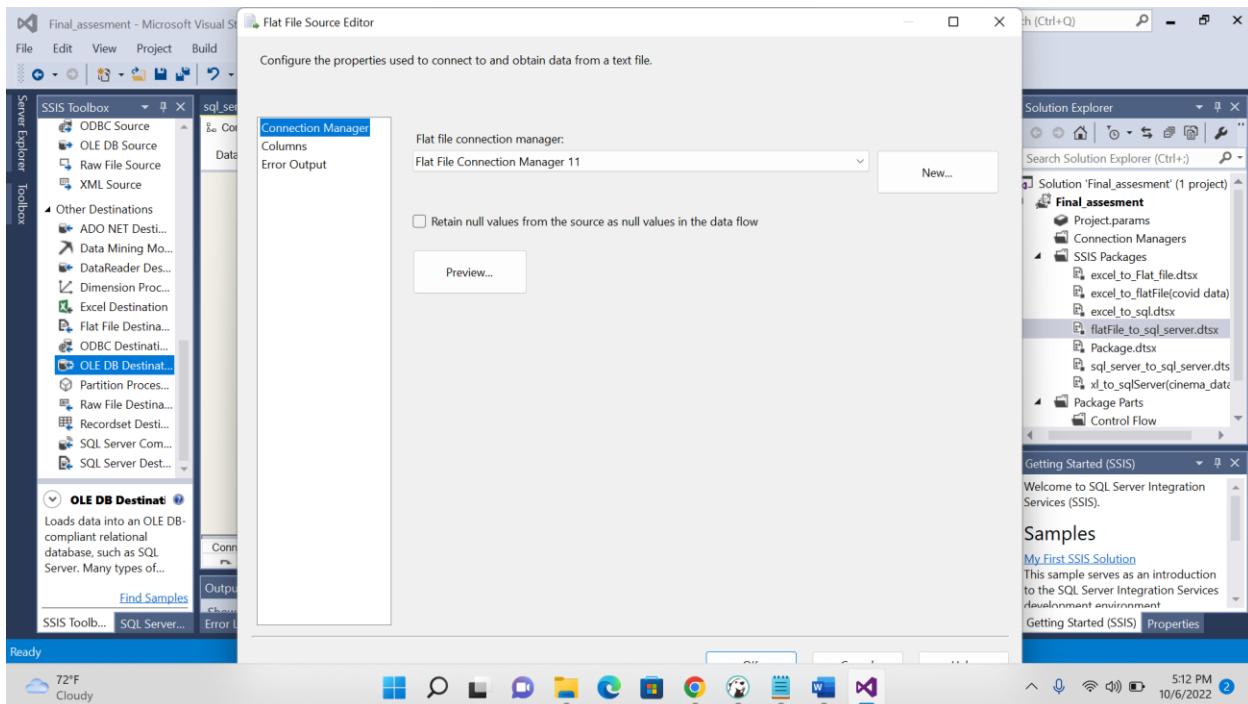
3.Extractiong the data from FLAT file source, loading into the sql server

Step1: make the connection as shown in the below



Step2: double click on the flat file

Step3: select the flat file



Step4: create destination table in SQL server

DBeaver 22.2.0 - <pratibha> Final assessment

File Edit Navigate Search SQL Editor Database Window Help

Data... × Proj... × <dlithe - rewansolution.a> <oracle-88094-0.cloudclusters.net> Scri... × General dbo SJ_HEALTH <pratibha> Script-3 × *<pratibha> Final assesment × <pratibha> Query

Enter a part of object

> <dlithe - rewansolution.a>

> oracle-88094-0.cloudclusters.net

> <pratibha - rewansolutio>

Databases Security Administrator

```
(  
    Film_Type varchar(100),  
    film_code varchar(100),  
    cinema_code varchar(100),  
    total_sales varchar(50),  
    tickets_sold varchar(50),  
    tickets_out varchar(50),  
    show_time varchar(50),  
    occu_perc varchar(100),  
    ticket_price varchar(50),  
    ticket_use varchar(50),  
    capacity varchar(100),  
    show_date varchar(50),  
    show_month varchar(100),  
    quarters varchar(50),  
    show_day varchar(50)  
)  
select * from DFA_CINEMA_TICKET
```

Results 1 ×

select * from DFA_CINEMA_TICKET Enter a SQL expression to filter results (use Ctrl+Space)

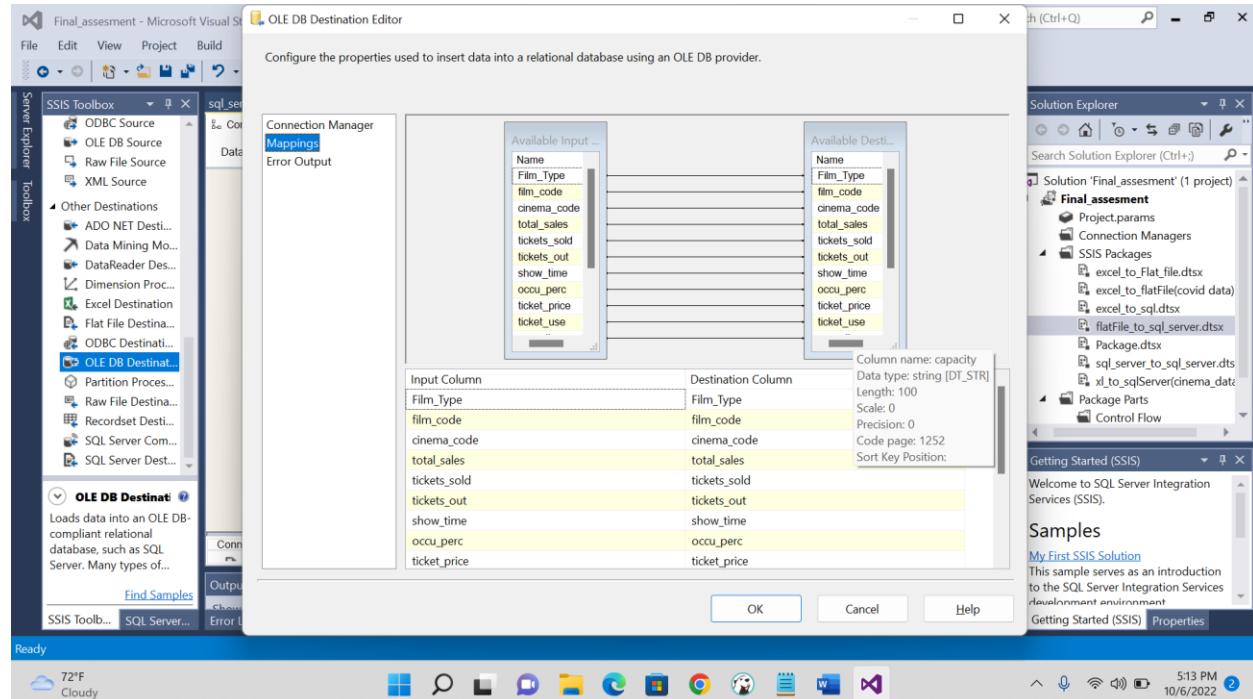
Name	Film_Type	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price	ticket_use	capacity
Bookmarks											
Diagrams											
Scripts											

Record Save Cancel Script IST en Writable Rows: 0 No data- 230ms. on 2022-10-06 at 12:33:45 Smart Insert 99 : 1 : 3004 Sel: 0 | 0

87°F Cloudy 12:33 PM 10/6/2022

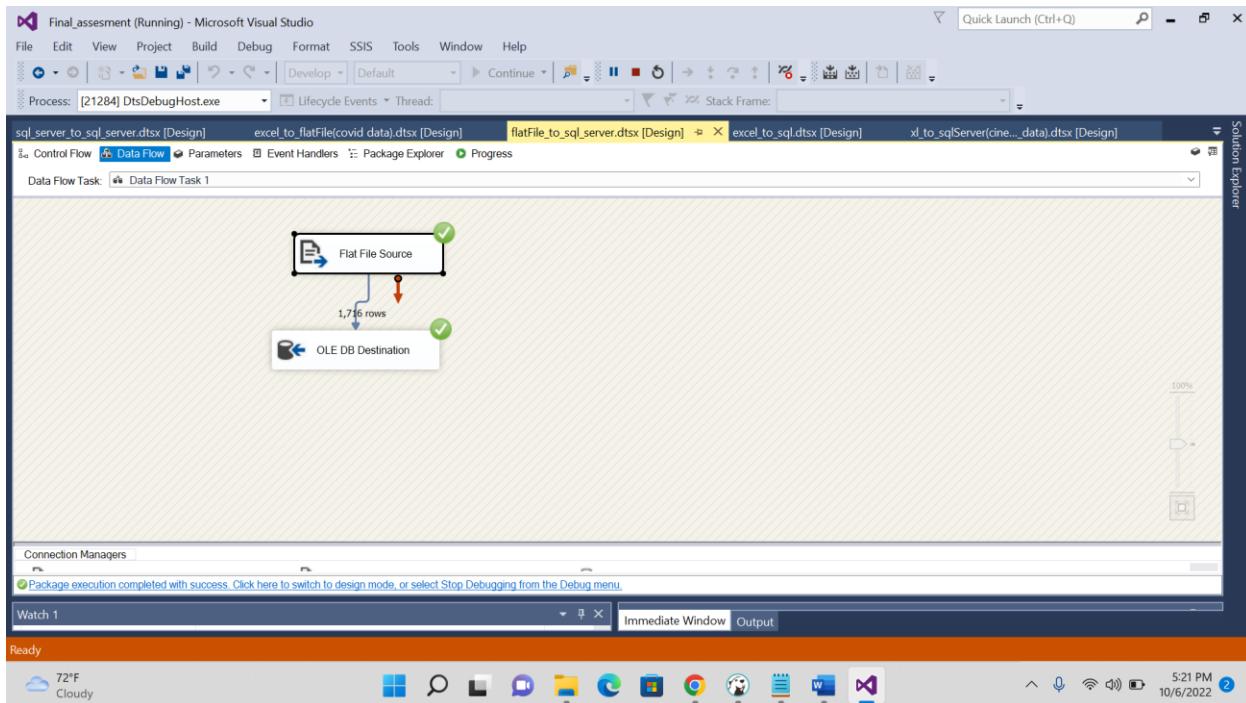
Step5: Double click on the OLDB destination

Step 6: select the destination table and map the coloums



Step8: Run the package

Below snapshot shows the execution of package

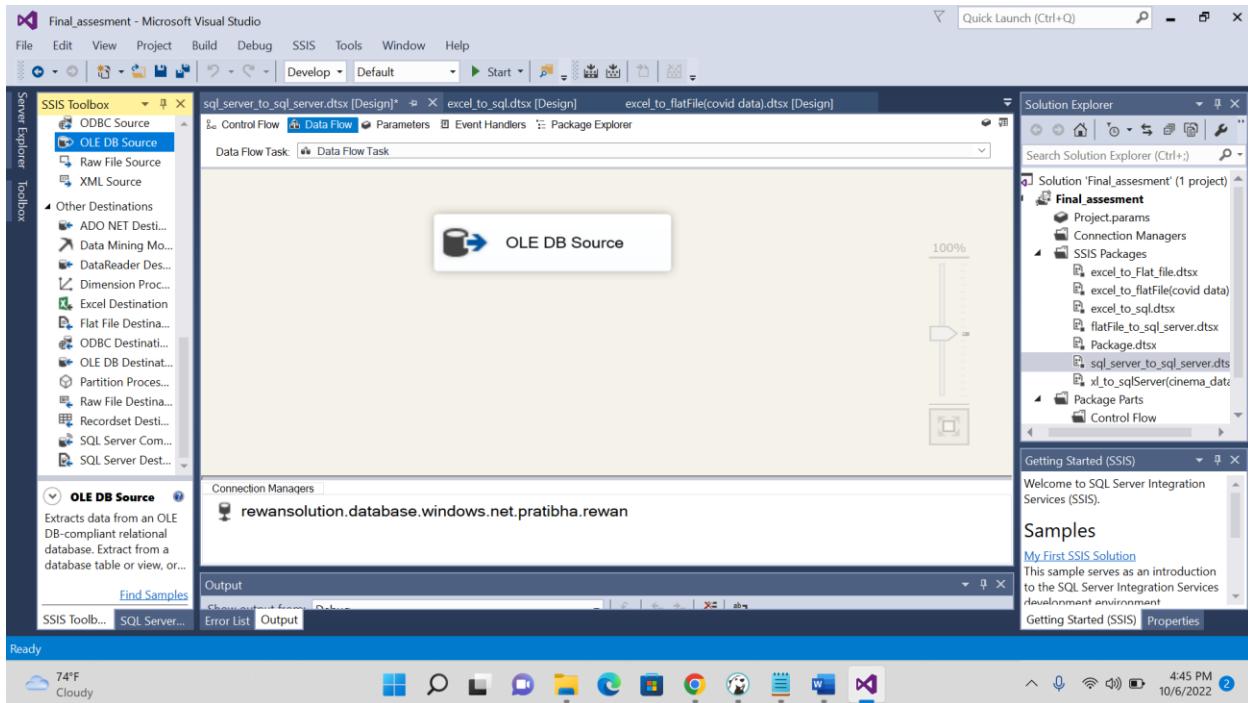


Below snapshot shows the data loaded into the sql server

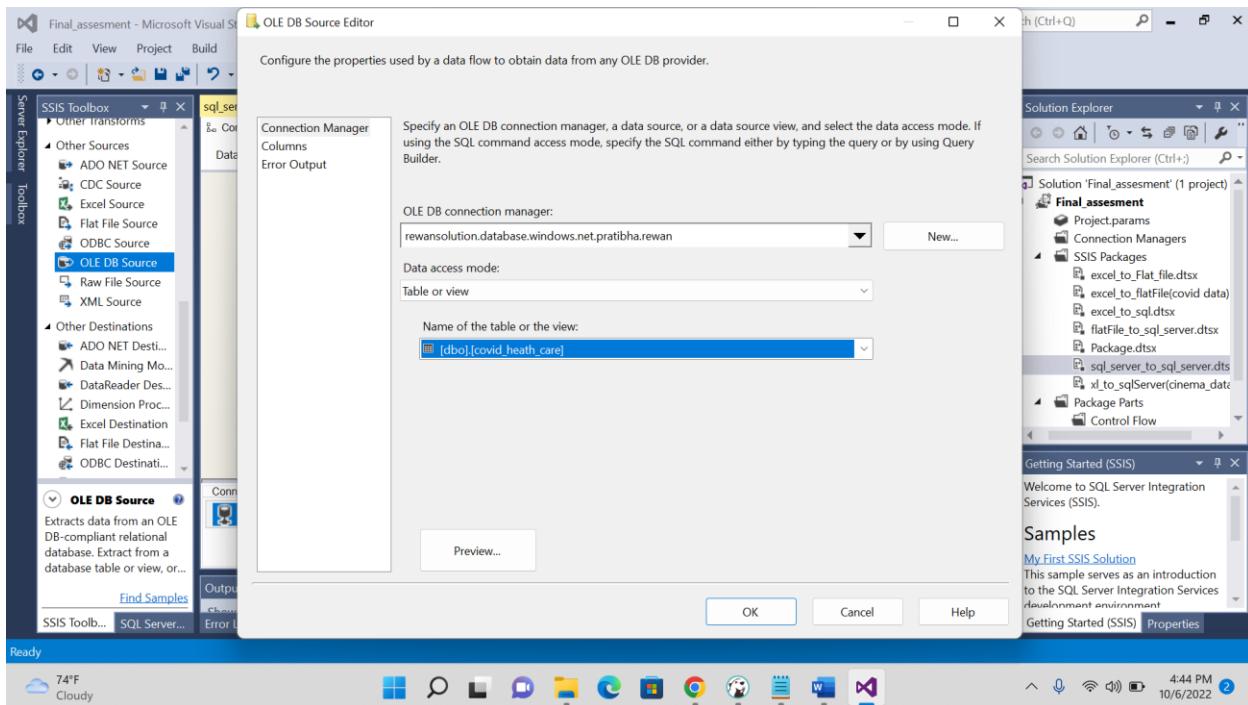
Film_Type	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price
Romance	1492	304	3900000	26	0	4	4.259999999999998	150000
Romance	1492	352	3360000	42	0	5	8.080000000000001	80000
Romance	1492	489	2560000	32	0	4	20	80000
Romance	1492	429	1200000	12	0	1	11.01	100000
Romance	1492	524	1200000	15	0	3	16.670000000000002	80000
Romance	1492	71	1050000	7	0	3	0.9799999999999998	150000
Romance	1492	163	1020000	10	0	3	7.690000000000004	102000
Romance	1492	450	750000	5	0	3	1.570000000000001	150000
Romance	1492	51	750000	11	0	2	0.9499999999999996	68181.8181800000
Romance	1492	522	600000	4	0	3	1.55	150000
Romance	1492	43	480000	6	0	3	0.44	80000
Romance	1492	529	480000	4	0	3	2.96	120000
Romance	1492	82	400000	5	0	6	0.5300000000000003	80000
Romance	1492	344	300000	2	0	3	0.25	150000
Romance	1492	73	240000	2	0	1	2.04	120000
Romance	1492	304	1650000	112	0	4	18.32999999999998	147321.428600000
Romance	1492	352	1395000	93	0	5	10.57	150000

3.Extracting the data from sql server, loading into the sql server

Step1: drag and drop the OLE DB source and connect to the OLEDB desination

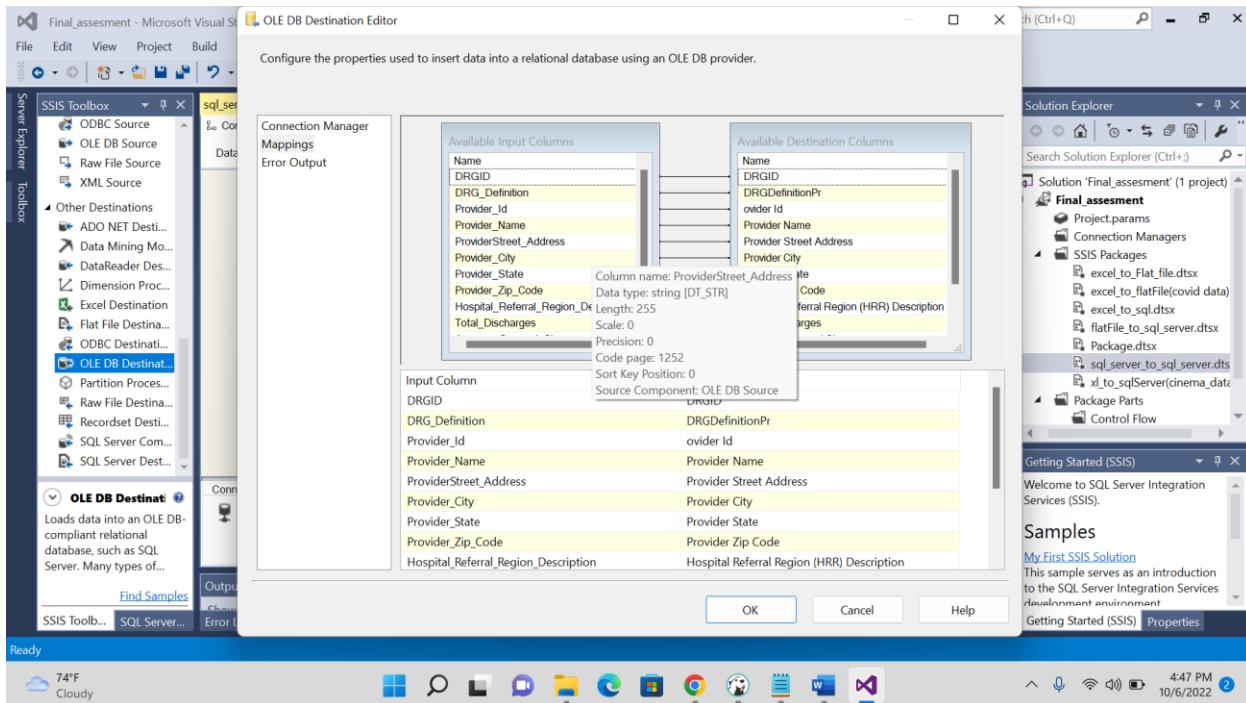


Step2: Double click on the source and select the source table

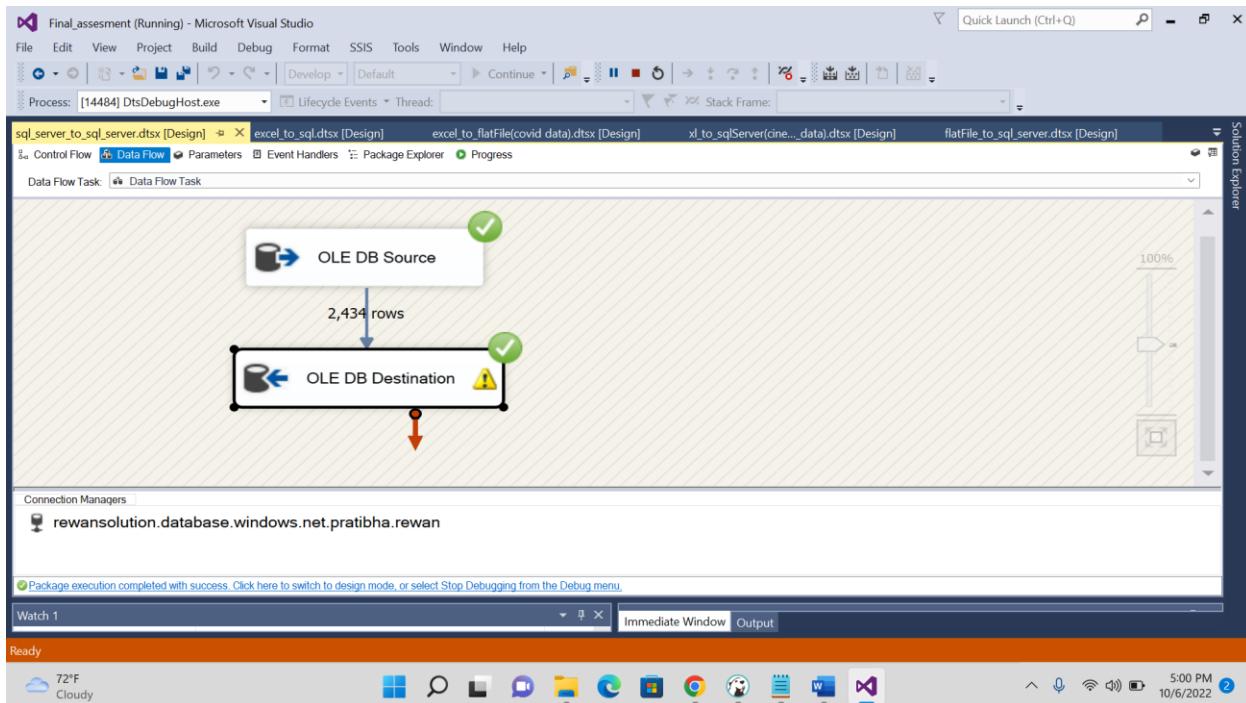


Step3: double click on the OLE DB destination

Step4: Map the columns



Step5: RUN the package



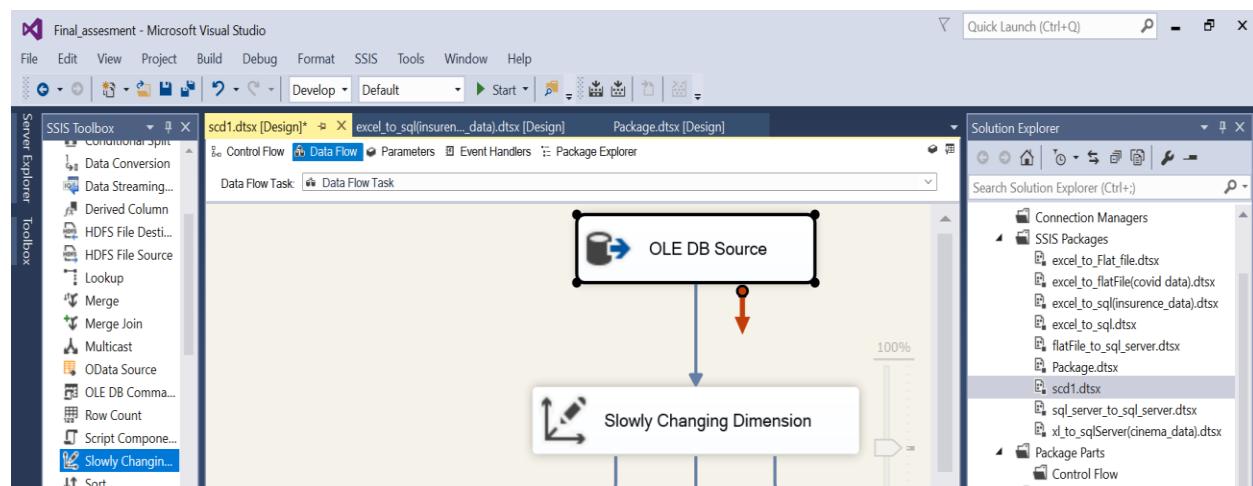
Snapshot shows the data loaded to the target table

	DRGID	DRGDefinitionP	Provider Id	Provider Name	Provider Street Address	Provider City	Provider State	Provider Zip Code
1	39	EXTRACRANIAL PROC 10001		SOUTHEAST ALABAMA 1108 ROSS CLARK CIRCLE	DOTHAN	AL	36301	
2	39	EXTRACRANIAL PROC 10005		MARSHALL MEDICAL 2505 U S HIGHWAY 431 NOR BOAZ	AL	35957		
3	39	EXTRACRANIAL PROC 10006		ELIZA COFFEE MEMO 205 MARENGO STREET	FLORENCE	AL	35631	
4	39	EXTRACRANIAL PROC 10011		ST VINCENT'S EAST 50 MEDICAL PARK EAST DRIV BIRMINGHAM	ALBASTER	AL	35235	
5	39	EXTRACRANIAL PROC 10016		SHELBY BAPTIST MED 1000 FIRST STREET NORTH	ALABASTER	AL	35007	
6	39	EXTRACRANIAL PROC 10023		BAPTIST MEDICAL CE 2105 EAST SOUTH BOULEVAD MONTGOMERY	AL	36116		
7	39	EXTRACRANIAL PROC 10029		EAST ALABAMA MED 2000 PEPPERELL PARKWAY	OPELIKA	AL	36801	
8	39	EXTRACRANIAL PROC 10033		UNIVERSITY OF ALAB 619 SOUTH 19TH STREET	BIRMINGHAM	AL	35233	
9	39	EXTRACRANIAL PROC 10039		HUNTSVILLE HOSPITAL 101 SIVLEY RD	HUNTSVILLE	AL	35801	
10	39	EXTRACRANIAL PROC 10040		GADSDEN REGIONAL 1007 GOODYEAR AVENUE	GADSDEN	AL	35903	
11	39	EXTRACRANIAL PROC 10046		RIVERVIEW REGIONA 600 SOUTH THIRD STREET	GADSDEN	AL	35901	
12	39	EXTRACRANIAL PROC 10055		FLOWERS HOSPITAL 4370 WEST MAIN STREET	DOTHAN	AL	36305	
13	39	EXTRACRANIAL PROC 10056		ST VINCENT'S BIRMIN 810 ST VINCENT'S DRIVE	BIRMINGHAM	AL	35205	
14	39	EXTRACRANIAL PROC 10078		NORTHEAST ALABAM 400 EAST 10TH STREET	ANNISTON	AL	36207	
15	39	EXTRACRANIAL PROC 10083		SOUTH BALDWIN RE: 1613 NORTH MCKENZIE STRE FOLEY	AL	36535		
16	39	EXTRACRANIAL PROC 10085		DECATUR GENERAL F 1201 7TH STREET SE	DECATUR	AL	35609	
17	39	EXTRACRANIAL PROC 10090		PROVIDENCE HOSPITAL 6801 AIRPORT BOULEVARD	MOBILE	AL	36608	
18	39	EXTRACRANIAL PROC 10092		D C H REGIONAL ME 809 UNIVERSITY BOULEVARD TUSCALOOSA	AL	35401		
19	39	EXTRACRANIAL PROC 10100		THOMAS HOSPITAL 750 MORPHY AVENUE	FAIRHOPE	AL	36532	
20	39	EXTRACRANIAL PROC 10102		PARTIST MEDICAL CE 201 PRINCETON AVENUE SOL RUMINCEAL			36211	

Perform SCD 1 & SCD2 dimension table modelling

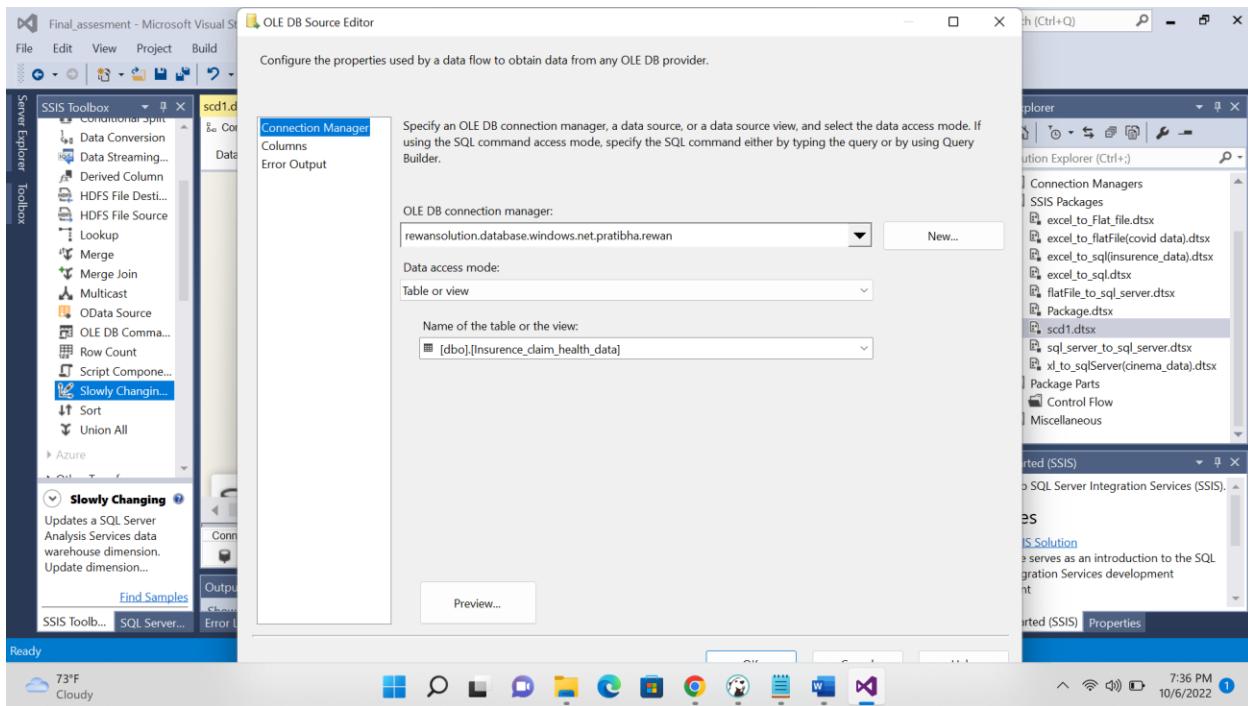
Solution:

Step1: make the as shown in the below



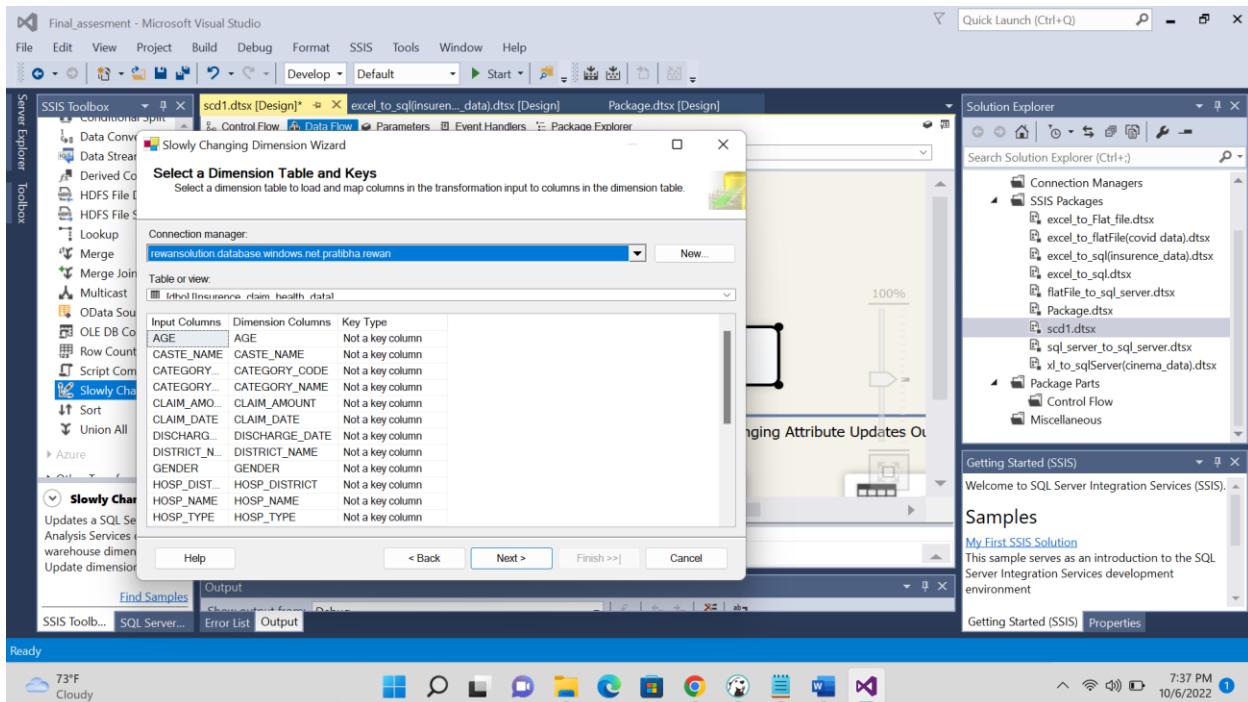
Step2: click on the OLE DB source and select the source table

Step3: check the column and preview the data

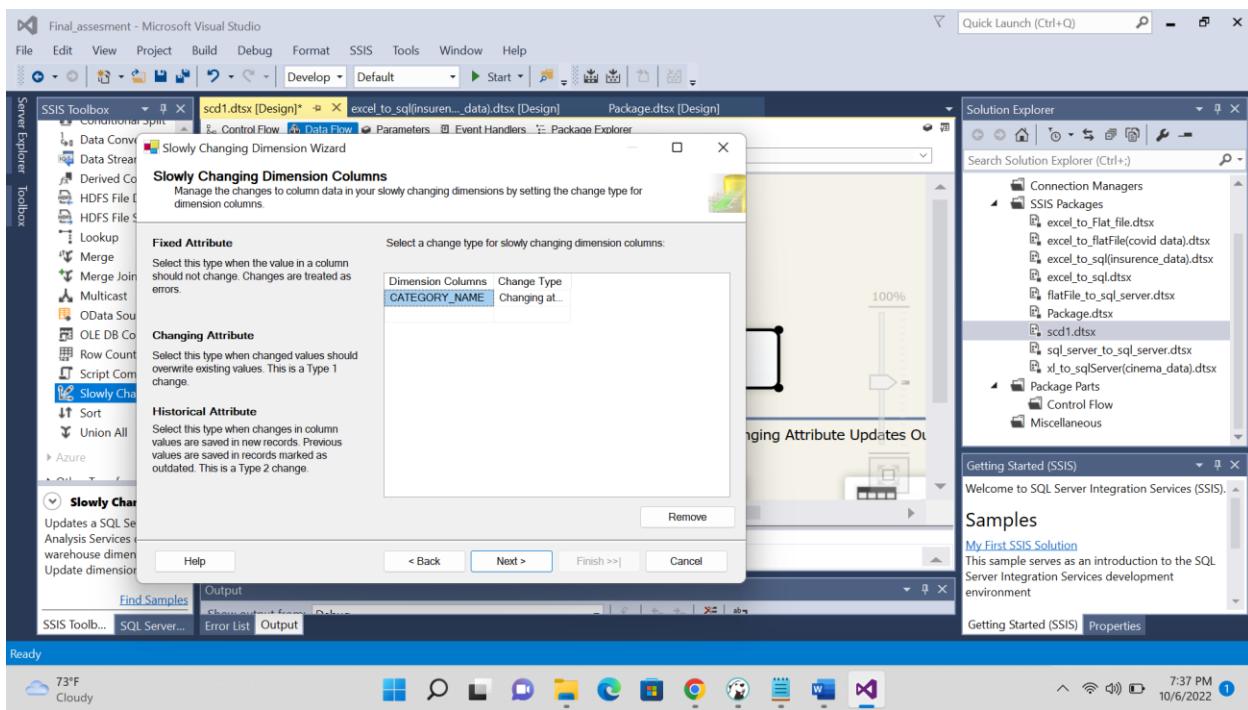


Step4: Double click on the slowly changing Dimension and choose the target table

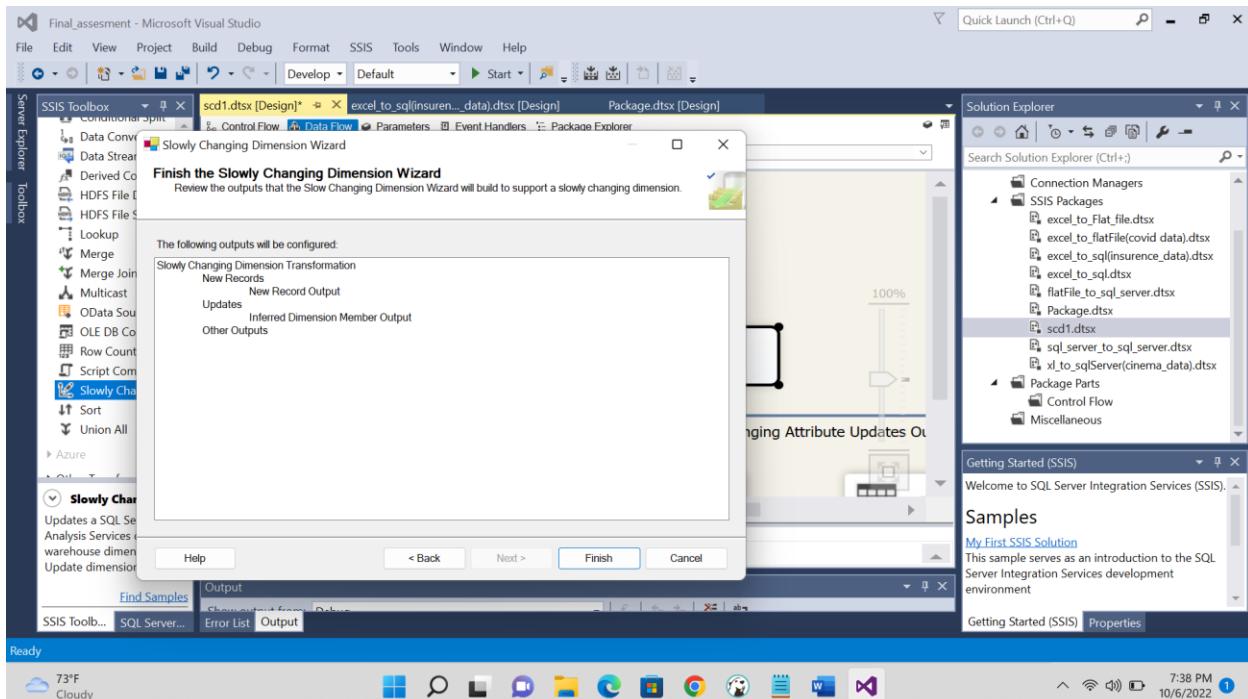
Step5: choose the business key from available column and click on next



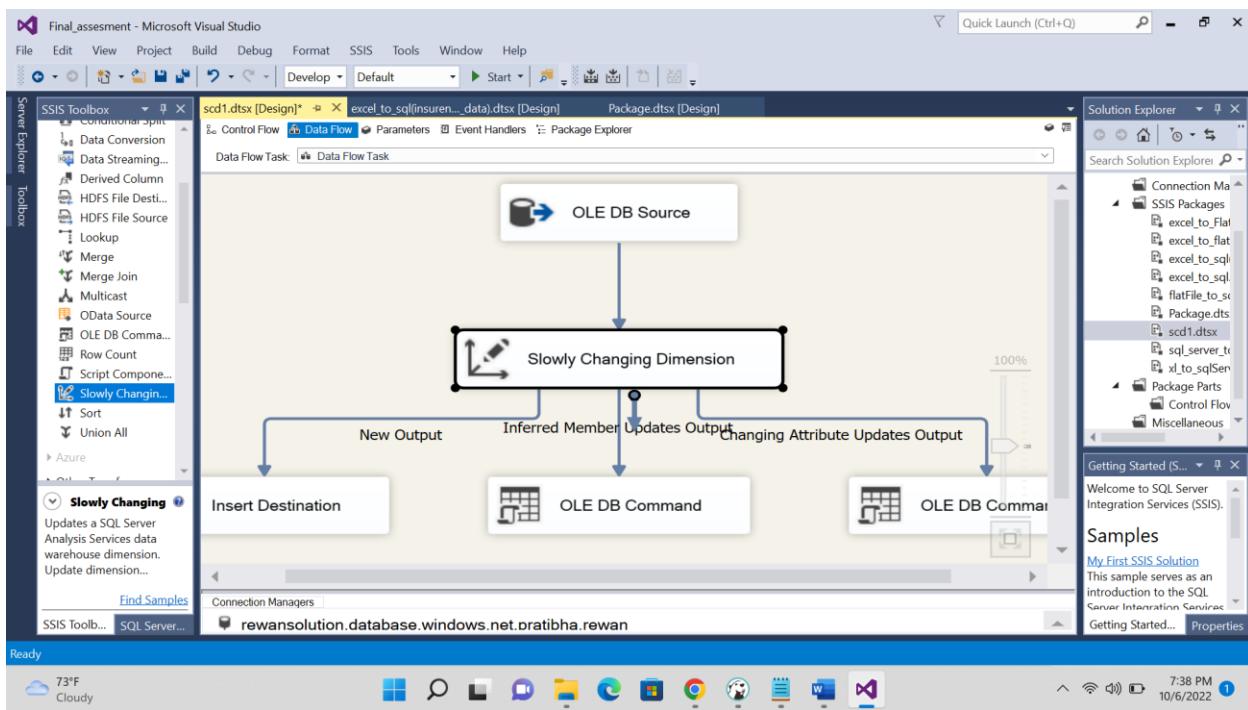
Step6: choose the changing attribute fixed attribute click on next



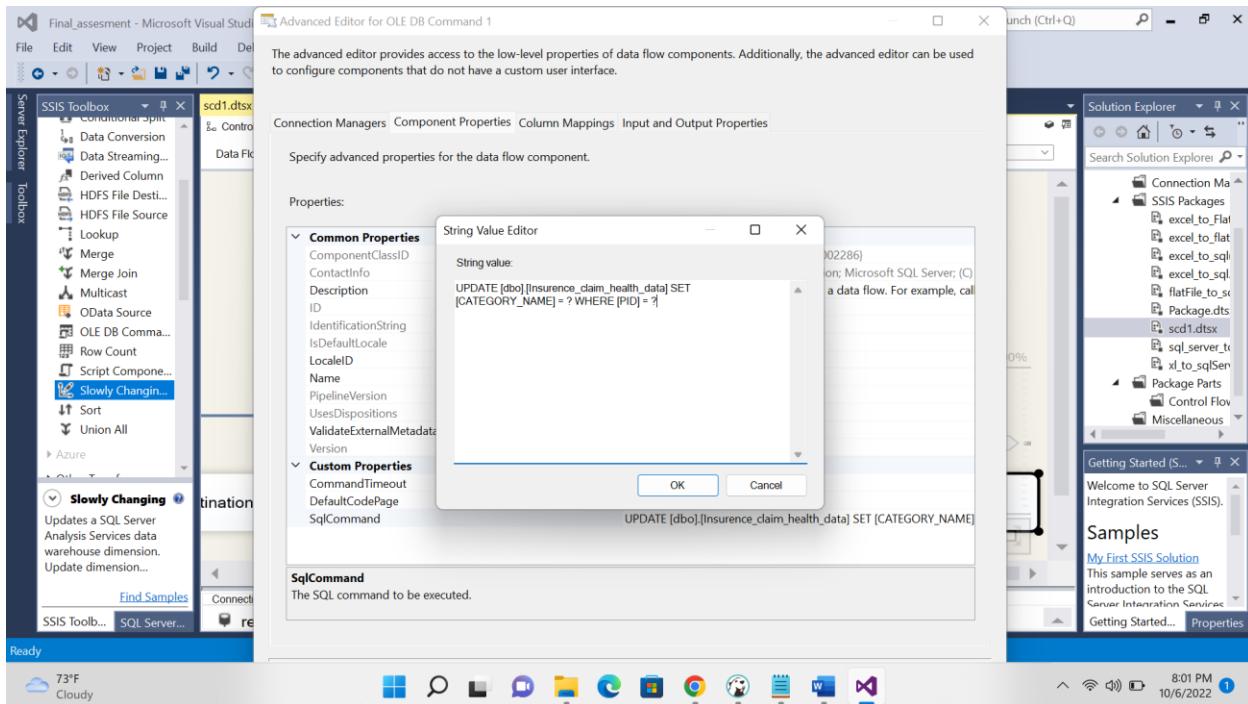
Step 7: click on finish



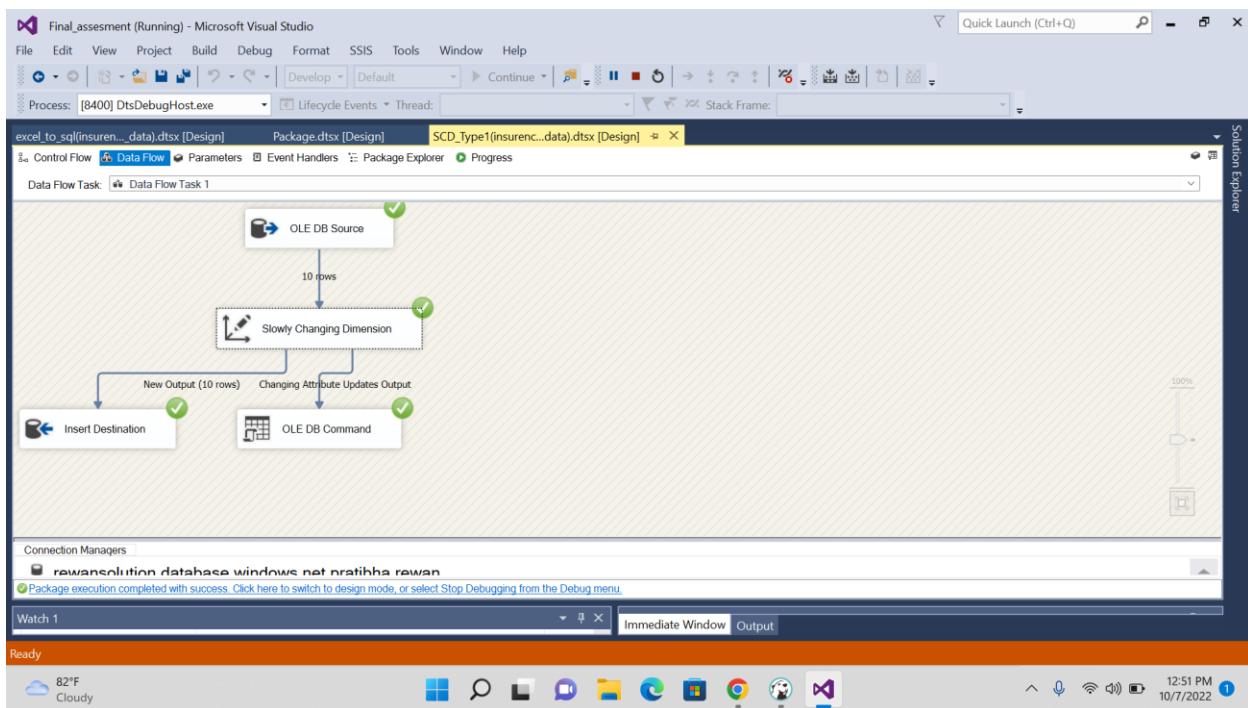
After performing these steps you will get connection as shown in the below snapshot



Step 8: view the update commands



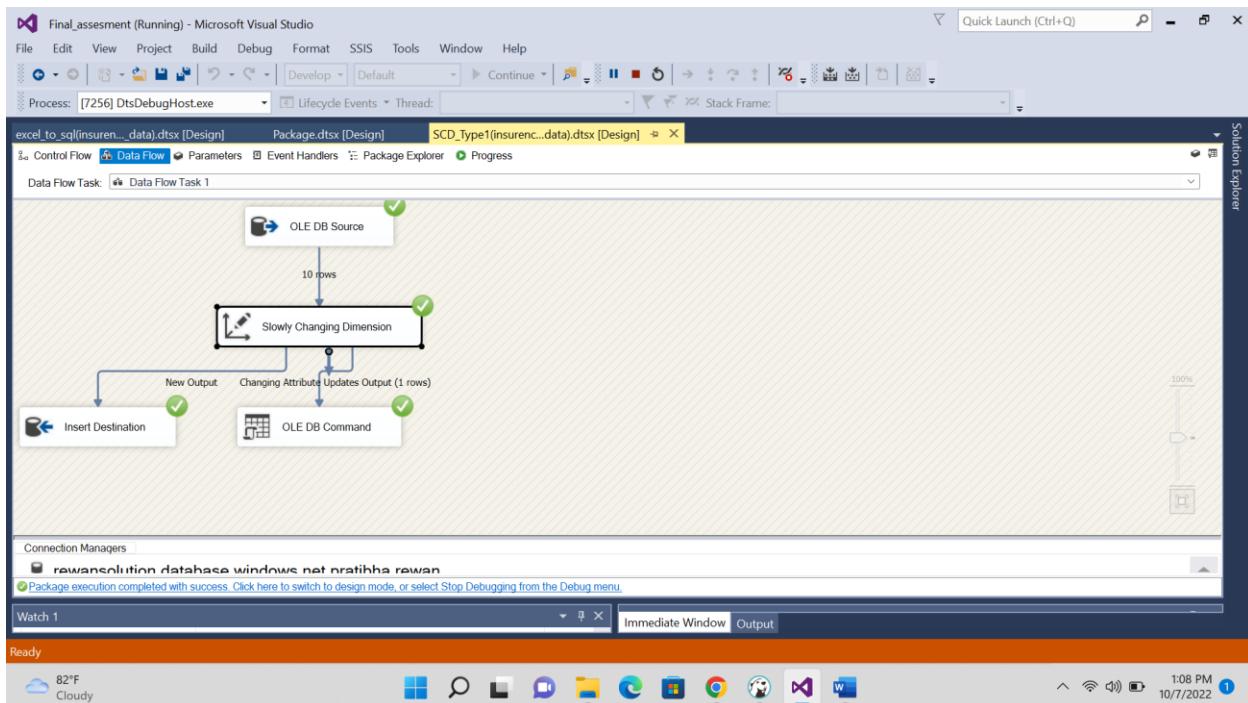
Step9: Run the package



Step 10: after execution update the source column

```
update Insurance_claim_healthdata_scd_type1 set claim_amount='25000'
where pid='1'
```

step 11: again, run the package after updating



Step12: check the target table

below snap shot show the updated target table

DBeaver 22.2.0 - <pratibha> Final assesment

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto_ pratibha dbo@pratibha @ @

<dlithe> sql_queries.sql <oracle-88094-0.cloudclusters.net> Script-2 General dbo SJ_HEALTH <pratibha> Script-3 <pratibha> Final assesment

Enter a part of c

pratibha

update Insurance_claim_healthdata_scd_type1 set claim_amount='25000' where pid='1'

Statistics 1 Results 2

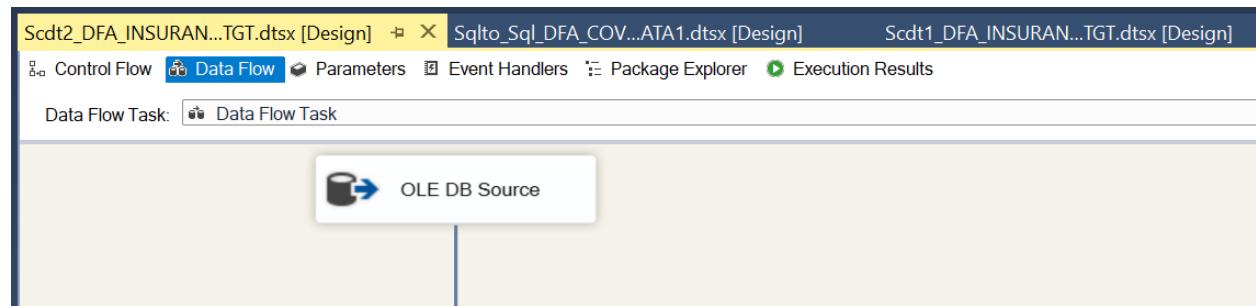
	VILLAGE	MANDAL_NAME	DISTRICT_NAME	PRAUTH_DATE	PRAUTH_AMT	CLAIM_DATE	CLAIM_AMOUNT	HOSP_NAME
1	Lolugu	Ponduru	Srikakulam	8/3/2013 8:38:00 PM	12500	3/22/2017 8:25:00 PM	25000	Rims Govt. General Hospital, Srika
2	Borivanka	Kaviti	Srikakulam	8/6/2013 7:26:00 AM	12500	3/22/2017 8:25:00 PM	11000	Rims Govt. General Hospital, Srika
3	Kapasakuddi	Kaviti	Srikakulam	8/9/2013 6:30:00 PM	12500	3/22/2017 8:25:00 PM	11500	Rims Govt. General Hospital, Srika
4	Telikipenta	Sarubujili	Srikakulam	8/24/2013 7:37:00 PM	12500	3/22/2017 8:25:00 PM	11000	Rims Govt. General Hospital, Srika
5	Thandemvalas	Srikakulam	Srikakulam	8/28/2013 5:03:00 PM	12500	3/22/2017 8:25:00 PM	11000	Rims Govt. General Hospital, Srika
6	Phasigangupeta	Pathapatnam	Srikakulam	8/28/2013 5:30:00 PM	12500	3/22/2017 8:25:00 PM	11000	Rims Govt. General Hospital, Srika
7	Kranti Nagar	Nandal	Kurnool	8/30/2013 7:53:00 AM	12500	3/22/2017 8:25:00 PM	11000	Govt General Hospital Kurnool
8	Bhogapuram	Bhogapuram	Vizianagaram	5/2/2014 3:11:00 PM	12500	10/11/2017 7:34:00 PM	5000	Queens Nri Hospitals
9	Vallur	Kakumanu	Guntur	6/10/2014 5:02:00 PM	40000	2/17/2017 7:21:00 PM	40000	Karumuri Hospital
10	Ward-15	Guntur(C)	Guntur	6/11/2014 8:41:00 PM	115846	2/17/2017 7:21:00 PM	115846	Karumuri Hospital

Save Cancel Script | IST en Writable | Smart Insert | 355 : 85 [82] | Sel: 82 | 1

Cloudy 82°F 10:09 PM 10/7/2022

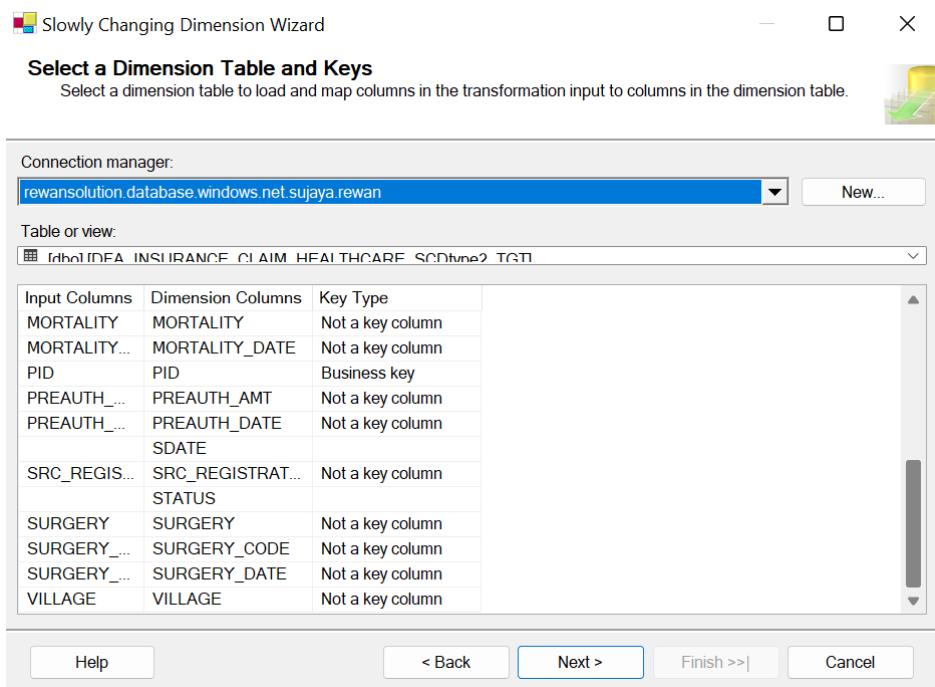
SCD-2

Step1: Drag and Drop the OLE DB source.

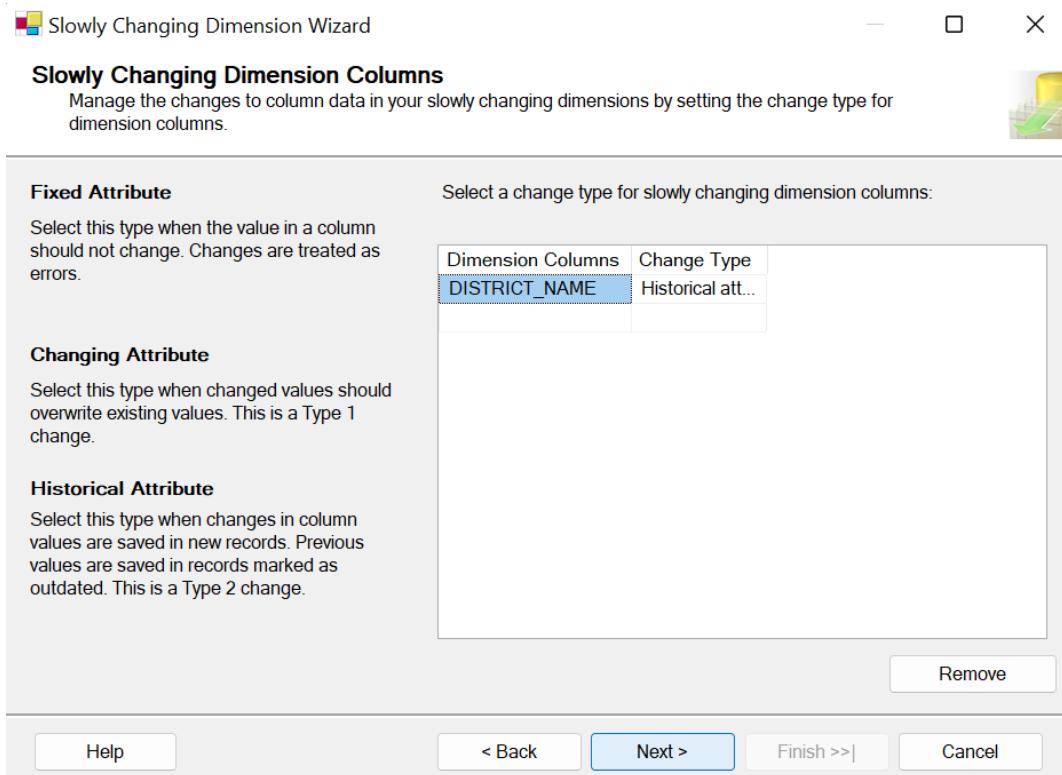


Step2: Double Click on OLE DB source and make the connection and select the source table in database.

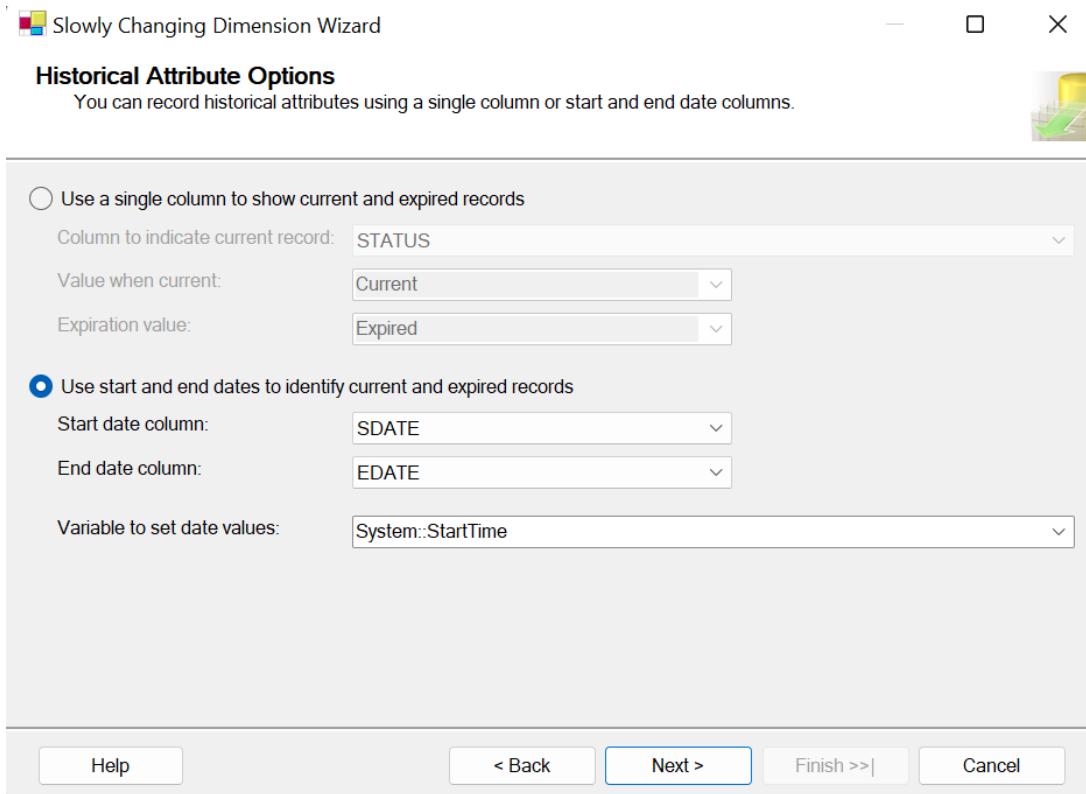
Step3: Drag and Drop the Slowly Changing Dimension and Connect the OLE DB to Scd and double click on Scd and Select the Target table and Business Key.



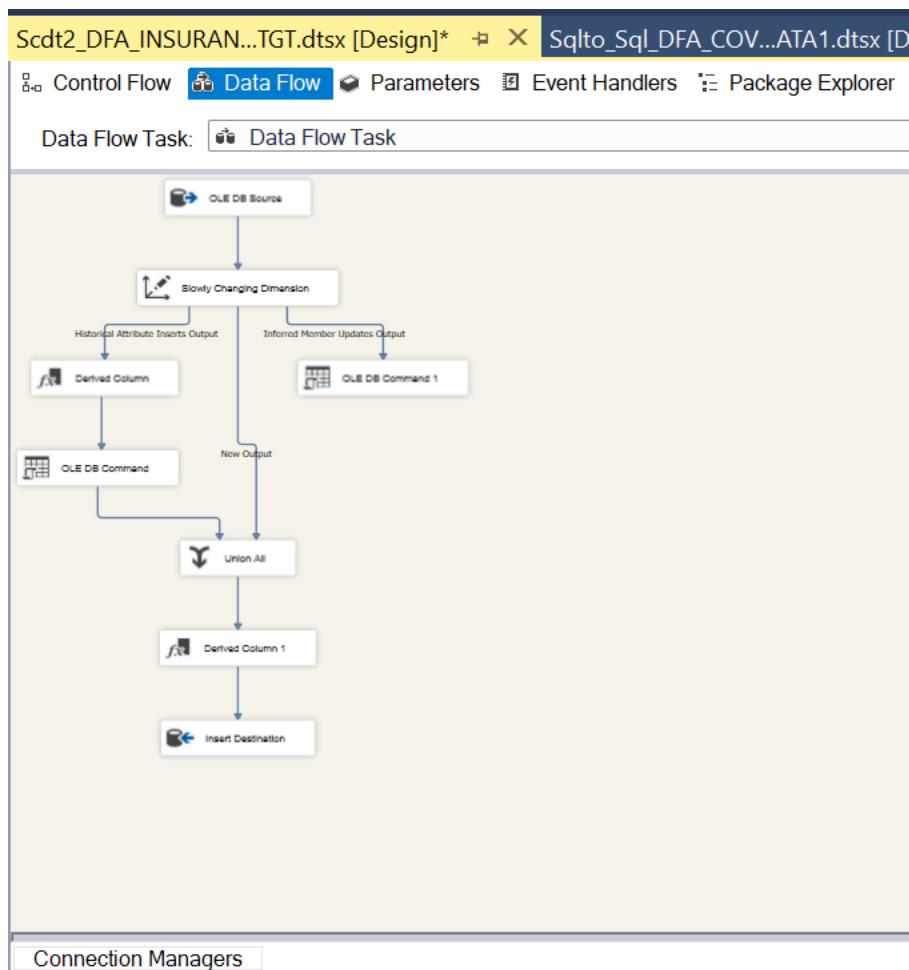
Step4:Select the Historical Attribute as change type for the column and give next.



Step5:Select the Start Date and End Date column and set the date values and click Finish.



After performing these steps you will get connection as shown in the below snapshot. And save and run the package.



Step6: Go to database and update the source table and run the package again.

```


    UPDATE DFA_INSURANCE_CLAIM_HEALTHCARE_SCDtype2_SOURCE
    SET DISTRICT_NAME='Guntur'
    WHERE PID=10;
  

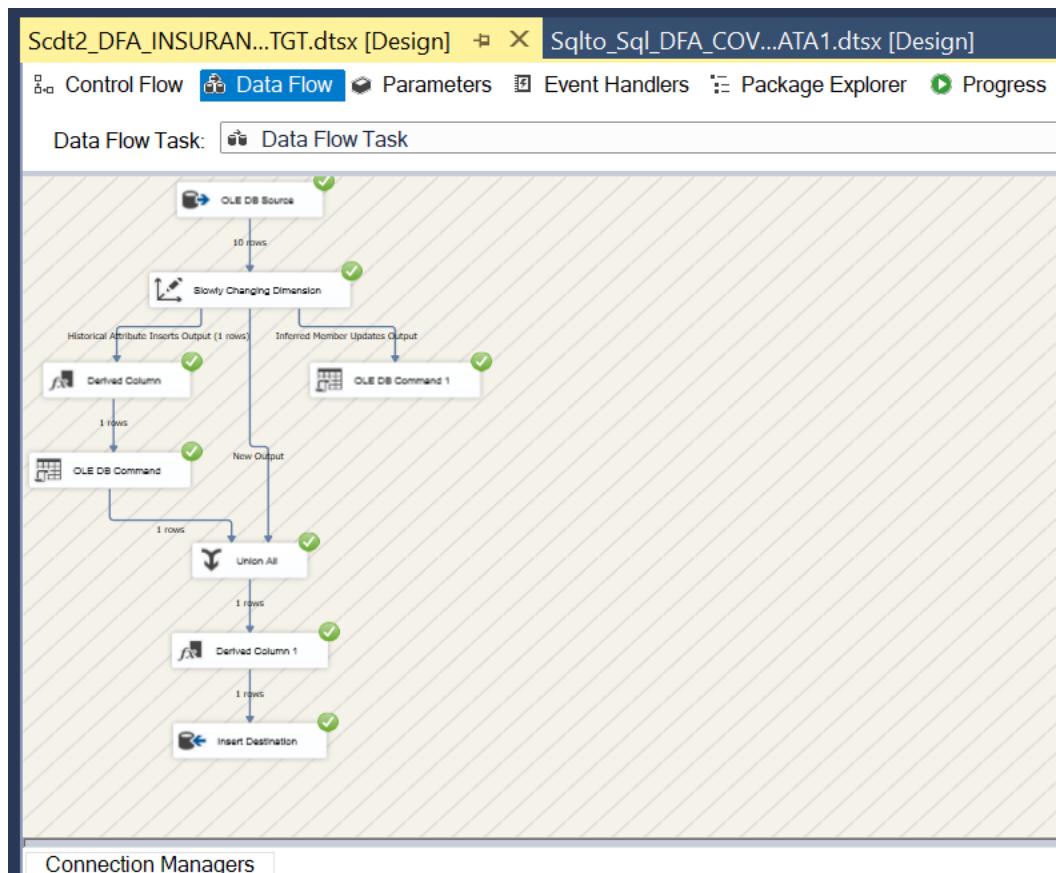
```

The screenshot shows a SQL Server Management Studio (SSMS) session window. The top bar displays connection details: <dlithe> SQL_qu..., dbo, SUJ_DEPT, <dlithe> Script-1, dlite, <sujaya> BasicS..., sujaya. The main area contains a single T-SQL UPDATE statement:

```


    UPDATE DFA_INSURANCE_CLAIM_HEALTHCARE_SCDtype2_SOURCE
    SET DISTRICT_NAME='Guntur'
    WHERE PID=10;
  

```



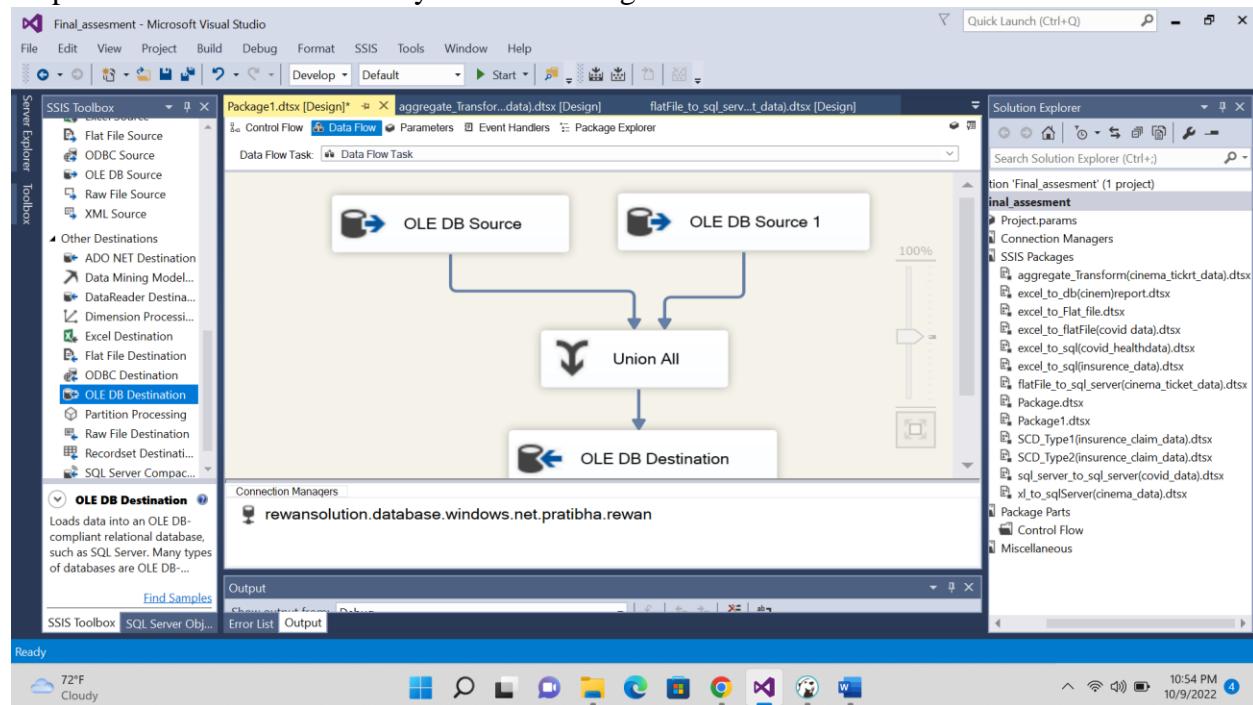
Step7:Check the target table to check whether the records are updated in the table.

Film_Type	total_sales	cinema_code	net_tickets_sold
Action	4850000	397	97
Thriller	2250000	448	155
Thriller	3760000	431	44
Science	142504998	448	1047
Romance	160000	144	2
Romance	160000	144	4
Romance	200000	43	5
Thriller	780000	380	9
Thriller	770000	43	11
Comedy	1505000	225	43
Thriller	2970000	489	213
Romance	26320000	304	329
Science	5880000	164	96
Action	50880000	160	636
Romance	3840000	163	64
Science	21150000	524	142
Romance	600000	506	15
Family	9960000	529	166
Romance	480000	144	6
Thriller	600000	531	10
Romance	750000	489	5
Romance	1680000	163	17
Thriller	16870000	485	241
Thriller	1450000	368	29

Using union all Transform

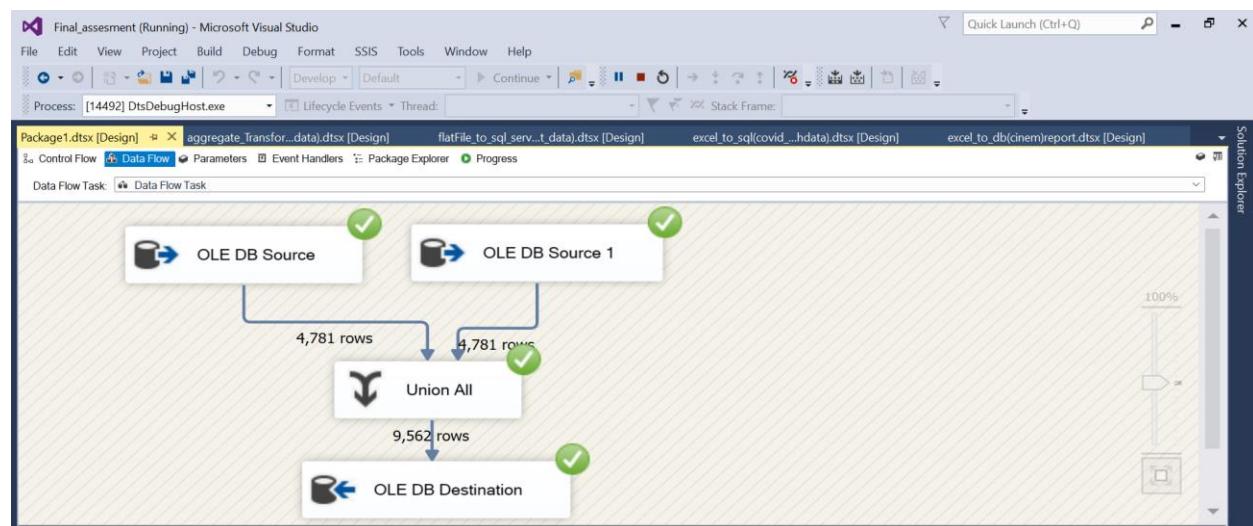
Step1: make connection as shown in below

Step 2: select the source table by double clicking OLE DB source



Step3: connect 2 sources to the union then connect to the destination

Step4 : run the package



Below snapshot shows the resultant table

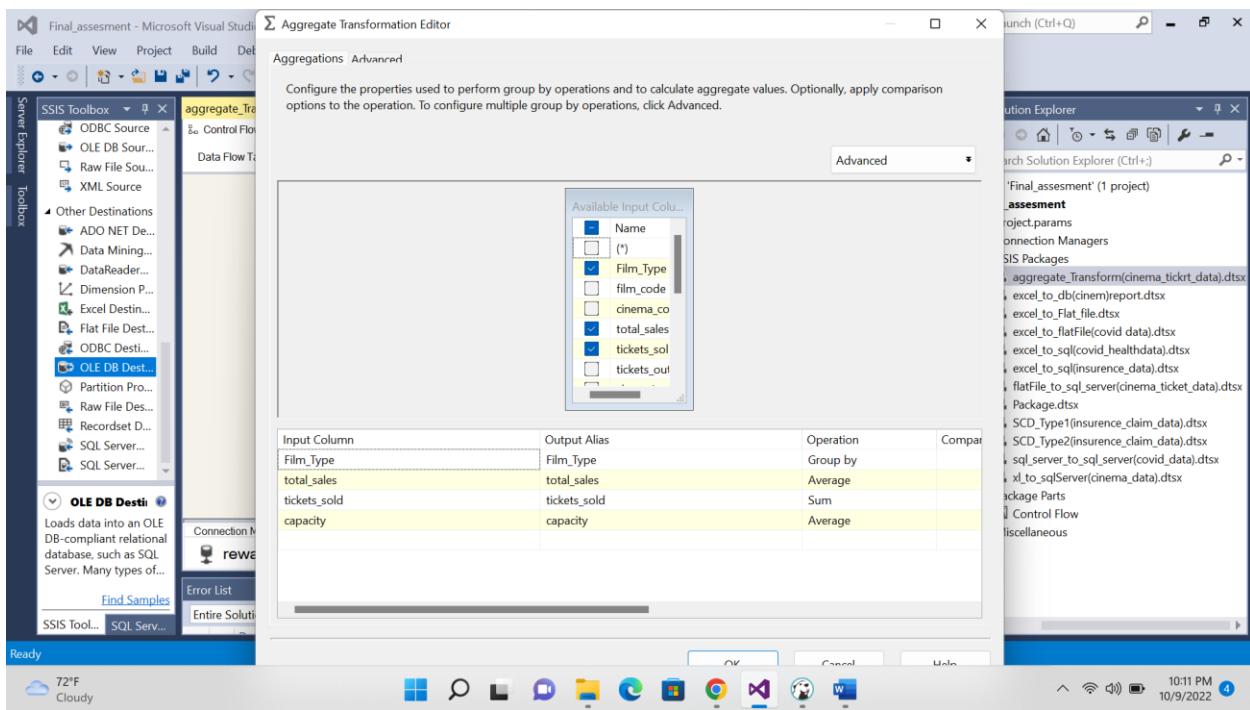
	Film_Type	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	...
1	Romance	1492	304	3900000	26	0	4	4.2599999999999998	150
2	Romance	1492	352	3360000	42	0	5	8.080000000000001	800
3	Romance	1492	489	2560000	32	0	4	20	800
4	Romance	1492	429	1200000	12	0	1	11.01	100
5	Romance	1492	524	1200000	15	0	3	16.670000000000002	800
6	Romance	1492	71	1050000	7	0	3	0.9799999999999998	150
7	Romance	1492	163	1020000	10	0	3	7.690000000000004	102
8	Romance	1492	450	750000	5	0	3	1.570000000000001	150
9	Romance	1492	51	750000	11	0	2	0.949999999999996	681
10	Romance	1492	522	600000	4	0	3	1.55	150
11	Romance	1492	43	480000	6	0	3	0.44	800
12	Romance	1492	529	480000	4	0	3	2.96	120
13	Romance	1492	82	400000	5	0	6	0.5300000000000003	800
14	Romance	1492	344	300000	2	0	3	0.25	150
15	Romance	1492	73	240000	2	0	1	2.04	120
16	Romance	1492	304	16500000	112	0	4	18.329999999999998	147
17	Romance	1492	352	13950000	93	0	5	10.57	150
18	Romance	1492	344	10200000	68	0	3	8.539999999999991	150
19	Romance	1492	71	6600000	44	0	3	6.139999999999997	150

Create aggregate table based on the column

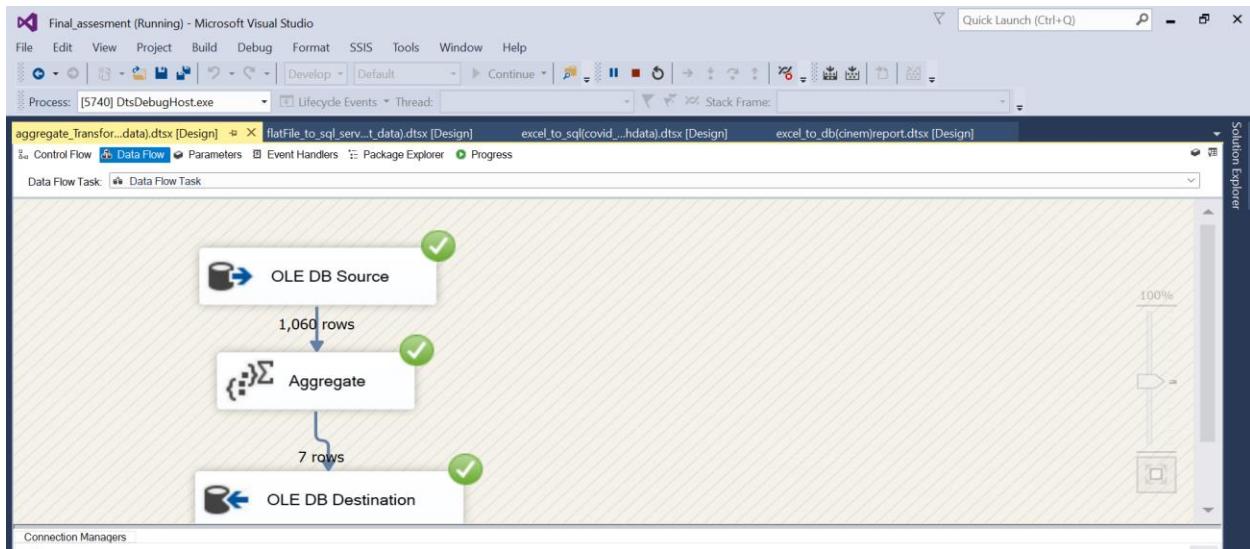
Step1: make connection shown below

Step2: select source table

Step3: double click on aggregate transform and choose operation as we need



Step4: Run the package



Below snap shows the resultant tabel

The screenshot shows a SQL query results grid. At the top, there is a toolbar with icons for file, edit, and database operations. Below the toolbar, the query text is displayed: `ALTER COLUMN total_sales bigint` and `select * from aggregate_function_cinema_data`. The results grid has three tabs: Results 1, Results 2, and Results 3. The Results 1 tab is selected and displays the following data:

	Film_Type	total_sales	tickets_sold	capacity
1	Fiction	5,390,076.290076336	11,308	630.4
2	Action	34,796,898.887978144	148,362	1,143.5546448087
3	Science	5,134,899.302013422	8,731	727.2751677852
4	Romance	1,795,401.4598540147	2,154	445.6569343066
5	Thriller	5,069,605.263157895	10,771	754.7631578947
6	Drama	2,686,018.5185185187	1,700	326.0185185185
7	Comedy	1,598,484.8484848484	1,102	741.6060606061