PROJECT REPORT

Dominick's Finer Foods OLAP Data Warehouse

By
Rohan Dhadwal

Table of Contents

Section 1: Introduction 1

Understanding the data 1

Customer Count file 1
Demographics file 2
UPC file 3
Movement File 4
Dominick Stores data 5
Week Decode Table 5

Details about Metadata 5

Customer Count File 6
Demographic File 8
UPC file 9
Movement File 10
Dominick Store Data 10
Week Decode Data 11

Entity Relationship Diagram 12

Section 2: Subject Area Understanding 13

Research on Retail Domain 13

Business Questions 14

Selected Business Questions 21

Section 3: Independent Data Marts design using Kimball's approach 21

Steps for Kimball's Methodology 22

Significance of Kimball's methodology for independent data marts 24

DW Logical Design (Star Schema Design) 24

Step 1: Requirement Analysis 25
Step 2: Building the Data Matrix 25
Step 3: Design Fact Tables 25
Step 4: Design Dimension Tables 26
Step 5: Finalized Design (Star Schema) 28

Mapping Table 1 – Source Files to Staging Tables 32

Mapping Table 2 – Staging Tables to Data Mart Tables 33

Physical Design 35

Section 4: Data Cleaning and Integration 35

Data Quality issues in DFF dataset 35

ETL Plan 36

Step 1: Target Tables 36
Step 2: Data Source Identification 39
Step 3: Data Mapping 40
Step 4: Data extraction rules 44
Step 5: Data Transformation and Cleansing Rules: 45

Step 6: Plan for Aggregate Table 47
Step 7: Organization of data staging area 47
Step 8: Procedures for all data extractions and loadings: 48
Step 9: ETL for Dimension Table: 48
Step 10: ETL for Fact Table: 49

Implementation of the ETL Plan 50
Data Extraction 50
Data Cleaning 60
Data Loading into Independent Data marts 86
Data Mart 1 – Product Category Specific 86
Data Mart 2 –Store Specific 91
Data Mart 3 – Movement Specific 100

List OF Temporary Tables Created 109

Section 5: BI Reporting 110

Reporting Plan 110

Reporting Implementation 113

Section 1: Introduction

Dominick's Finer Foods (DFF) was a major grocery store chain based in the Chicago area. It was found in 1918 by a Sicilian immigrant Dominick di Matteo. Under Matteo by 1990's, DFF grew to become the second largest grocery chain, having 13% market share in Chicago, just behind Jewel-Osco grocery store which had 35% market share.

In 1995, after Mr Matteo's death, Dominick was acquired by Yucaipa Companies. Unfortunately, their expansion strategy led to increasing debt and operational challenges. In 1998. Dominick's was sold to Safeway Inc. for $1.2 billion. Under Safeway's ownership, Dominick struggled to keep the same market share as New market rivals like Walmart bloomed. Some of the key problems we want to delve deeper into which Dominick faced are:

1. Pricing and Profitability Challenges: The company had to balance competitive pricing and maintaining profit margins.

2. Product Assortment and Branding Issues: When Dominick was acquired by Safeway in 1998, Safeway replaced many Dominick's local products with their own private label brands. Since it was unfamiliar with the customers, the sales for such products decreased affecting the overall sales of DFF.

3. *Changing Consumer Preferences:* Dominick could not adapt to the evolving customer demands, hence the Consumer's shifted from DFF to other brands which satisfied their wants.

For this project, our team is constructing a data warehouse using the historical store-level data from DFF stores in Chicago collected between 1989-1994. This comprehensive dataset includes detailed information of stores, brands, categories, sales, promotions etc. By leveraging the rich historical data, we aim to uncover insights into critical business challenges like the ones mentioned above and support data-driven decision making for the effective operation of such businesses.

Understanding the data

During the initial stage of developing and implementing a Data Warehouse for the Dominicks Store, our team needs to thoroughly understand the source data for the DFF. This information is stored in various files, including UPC, Movement, Demographics, and CCount. To analyze the source data and its metadata, we extract small sample datasets from each of these files. We then use Excel tools to display and visualize the information for our analysis. Additionally, we have included images of the Pivot Tables used to identify overall data patterns, as well as Pivot Charts to illustrate the presented data.

Customer Count file

This file represents sales data for the store, tracking daily sales for various product categories such as grocery, dairy, frozen items, bottled goods, and meat, along with promotional metrics. Each row

corresponds to a specific date and store, with sales recorded across multiple categories. It gives an overview of the store's performance over time, helping analyze trends in sales and inventory.

A	B	C	D	E	F	G	H	I	J	K
STORE	DATE	GROCERY	DAIRY	FROZEN	BOTTLE	MVPCLUB	GROCCOUP	MEAT	MEATFROZ	MEATCOU
2	880101	17245.41	0	0	0	0	0	3926.1	0	0
2	880102	34917.48	0	0	0	0	0	7730.39	0	0
2	880103	28757.35	0	0	0	0	0	4474.85	0	0
2	880104	13533	3030	3636	0	0	-1	4144	0	0
2	880105	21542.94	0	0	0	0	0	4692.51	0	0
2	880106	24075.55	0	0	0	0	0	5532.89	0	0
2	880107	28791.28	0	0	0	0	0	6470.54	0	0
2	880108	28527.38	0	0	0	0	0	6815.86	0	0
2	880109	44203.2	0	0	0	0	0	10705.73	0	0
2	880110	30178.96	0	0	0	0	0	5413.91	0	0
2	880111	22643.04	0	0	0	0	0	4568.02	0	0
2	880112	22026.01	0	0	0	0	0	4493.82	0	0
2	880113	20134.51	0	0	0	0	0	4605.94	0	0
2	880114	27217.22	0	0	0	0	0	6850.41	0	0
2	880115	29321.51	0	0	0	0	0	6620.72	0	0
2	880116	43604.31	0	0	0	0	0	10351.57	0	0
2	880117	30183.34	0	0	0	0	0	5693.98	0	0

Figure 1: Ccount File uploaded in Excel

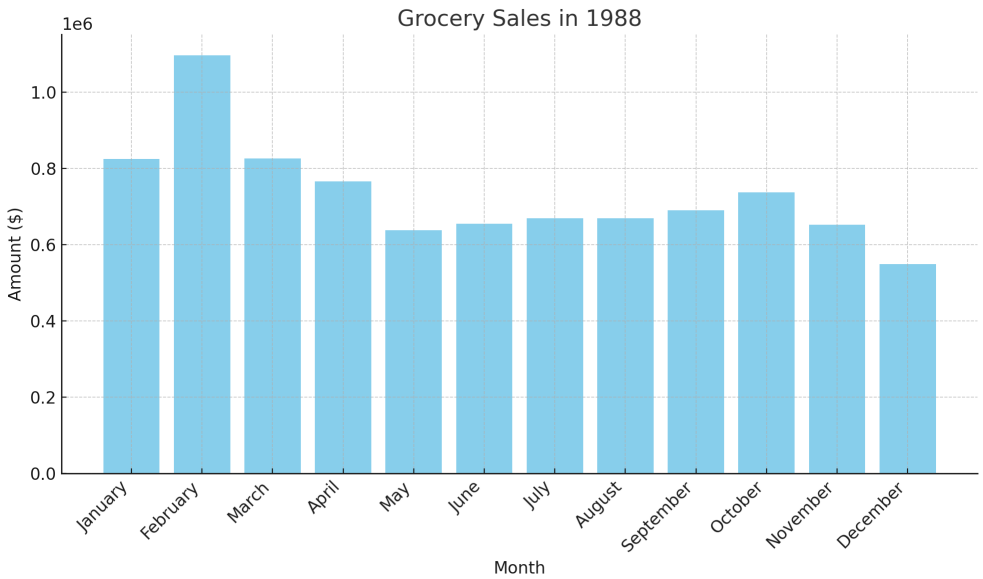


Figure 2: Month wise sales in Grocery for the year 1988

Demographics file

This demographics table offers store-specific profiles based on 1990 U.S. census data for Chicago. It includes metrics such as age, ethnic composition, education, income, household size, housing values, vehicle ownership, employment (especially for working women), and shopper behaviors. The data provides insights into each store's trading area, helping understand consumer demographics and market potential for targeted strategies.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
MMID	NAME	CITY	ZIP	LAT	LONG	WEEKVOL	STORE	SCLUSTER	ZONE	AGE9	AGE60	ETHNIC	EDUC	NOCAR	INCOME
16892	DOMINICKS 2	RIVER FOREST	60305	419081	878131	350	2	C	1	0.117509	0.232865	0.11428	0.248935	0.124603	10.55321
16893	DOMINICKS 4	PARK RIDGE	60068	420392	878425	300	4	A	2	0.09509	0.26203	0.062161	0.220789	0.055567	10.64697
16894	DOMINICKS 5	PALATINE	60067	421203	880431	550	5	D	2	0.141433	0.117368	0.053875	0.321226	0.02557	10.92237
16895	DOMINICKS 8	OAK LAWN	60453	417331	877436	600	8	C	5	0.123155	0.252394	0.035243	0.095173	0.075113	10.59701
16896	DOMINICKS 9	MORTON GROVE	60053	420411	877994	450	9	A	2	0.103503	0.269119	0.032619	0.222172	0.040128	10.78715
16898	DOMINICKS 12	CHICAGO	60660	419928	876592	450	12	B	7	0.105697	0.178341	0.380698	0.253413	0.483518	9.996659
16899	DOMINICKS 14	GLENVIEW	60025	420733	877994	400	14	A	1	0.129589	0.213949	0.034179	0.348293	0.026586	11.04393
16901	DOMINICKS 18	RIVER GROVE	60171	419364	878331	600	18	A	5	0.110095	0.272313	0.074417	0.072246	0.141975	10.39198
							19								
16903	DOMINICKS 21	HANOVER PARK	60103	420058	881411	500	21	D	6	0.175926	0.066896	0.105039	0.177503	0.017598	10.71619
							25								
16905	DOMINICKS 28	MOUNT PROSPECT	60056	420686	879208	275	28	A	2	0.12888	0.213309	0.055935	0.233163	0.054855	10.79853
16906	DOMINICKS 32	PARK RIDGE	60068	419872	878378	575	32	C	1	0.099061	0.254953	0.031939	0.19826	0.071701	10.67448
16907	DOMINICKS 33	CHICAGO	60657	419386	876447	300	33	B	7	0.046071	0.13417	0.130127	0.419688	0.506224	10.34593
							39								
16909	DOMINICKS 40	BRIDGEVIEW	60455	417317	877969	500	40	D	6	0.133685	0.181852	0.044053	0.072129	0.04633	10.55025
16912	DOMINICKS 44	WESTERN SPRINGS	60558	418033	878903	325	44	A	2	0.144883	0.190983	0.037632	0.329738	0.040766	10.86916
16913	DOMINICKS 45	WHEELING	60090	421403	879300	300	45	D	2	0.146719	0.128857	0.087234	0.28015	0.020232	10.74538
							46	D	5						
16915	DOMINICKS 47	ADDISON	60101	419364	880022	350	47	D	2	0.142962	0.125798	0.120676	0.140599	0.021297	10.63533
16916	DOMINICKS 48	SCHAUMBURG	60193	420503	880775	325	48	D	2	0.121767	0.097922	0.099492	0.30326	0.021209	10.75603
16917	DOMINICKS 49	DOWNERS GROVE	60515	418111	879869	275	49	A	2	0.134878	0.187473	0.038353	0.31995	0.054382	10.80675
16918	DOMINICKS 50	HICKORY HILLS	60457	417169	878347	275	50	A	2	0.12442	0.153357	0.070926	0.128764	0.036434	10.58931
16919	DOMINICKS 51	PALOS HEIGHTS	60463	416594	877775	400	51	D	3	0.132472	0.17616	0.025426	0.171917	0.025436	10.62084
16920	DOMINICKS 52	NORTHBROOK	60062	421364	878825	450	52	A	1	0.136606	0.152241	0.084899	0.372927	0.0149	11.05102
16921	DOMINICKS 53	CHICAGO	60662	420039	877069	300	53	C	7	0.120839	0.300279	0.065722	0.270383	0.145363	10.6
16922	DOMINICKS 54	NAPERVILLE	60540	417975	881225	375	54	D	2	0.147915	0.090222	0.046641	0.421126	0.02084	10.9109

Figure 3: Demographics file uploaded in Excel

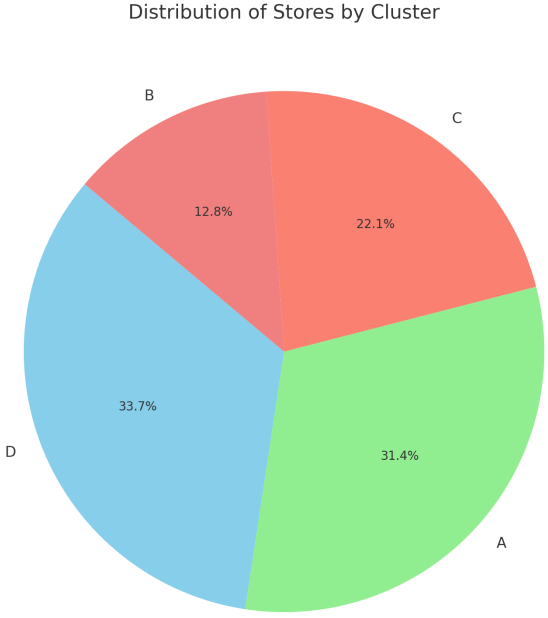


Figure 4: Distribution of Stores per cluster

UPC file

The UPC files categorize and describe products sold in different categories, identifying both the product and manufacturer. They help track items, even when updated versions exist.

A	B	C	D	E	F
COM_CODE	UPC	DESCRIP	SIZE	CASE	NITEM
953	1192603016	CAFFEDRINE CAPLETS 1	16 CT	6	7342431
953	1192662108	SLEEPINAL SOFTGEL	8 CT	6	7333311
953	1650001020	NERVINE TABS	30 CT	1	8430820
953	1650001022	NERVINE SLEEP AID	12 CT	1	8430840
953	1650004106	ALKA-SELTZER GOLD	20 CT	1	8430880
953	1650004108	ALKA-SELTZER GOLD	36 CT	1	8430900
953	1650004703	ALKA MINTS	30 CT	1	8430700
953	2140649030	LEGATRIN PM	30 CT	1	8435810
953	2586600493	PERCOGESIC A/F ANALG	50 CT	1	8416280
953	2586610493	PERCOGESIC A/F ANALG	50 CT	1	8416280
953	2586610501	ALEVE TABLETS	24 CT	6	6122441
953	2586610502	ALEVE CAPLETS	24 CT	6	6122741
953	2586610503	ALEVE TABLETS	50 CT	6	6122451
953	2586610504	ALEVE CAPLETS	50 CT	6	6122751
953	2586610505	ALEVE TABLETS	100 CT	6	6122461
953	2586610506	ALEVE CAPLETS	100 CT	6	6122761
953	3225259620	SUNBEAM HEAT WRAP MS	1 CT	1	8402470
953	3680012732	TC MOTION SICKNESS T	12 CT	12	6190791

Figure 5: UPC file data

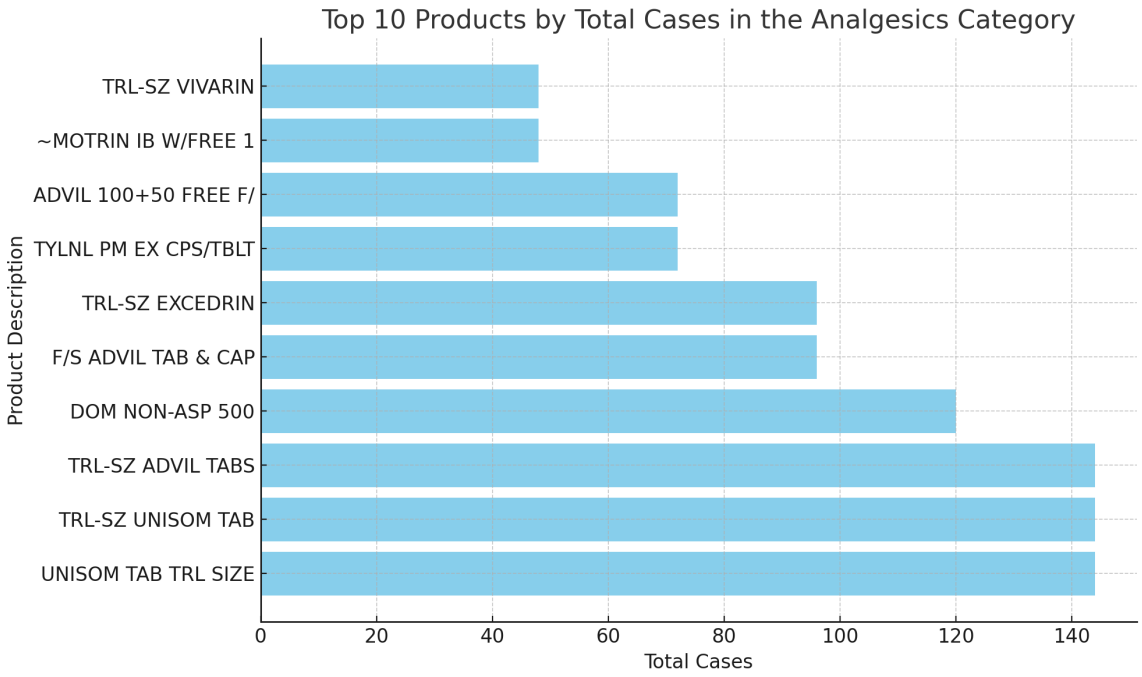


Figure 6: Top ten products by total cases in the Analgesics Category

Movement File

The movement files contain weekly sales data by UPC for each store and category. Sales are calculated as (Price * Move) / Qty, with "move" indicating individual items sold. Profit is the gross margin percentage, based on Average Acquisition Cost (AAC). Promotional sales are indicated by codes like 'B' (Bonus Buy) and 'C' (Coupon).

STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK
76	1192603016	306	0	1	0		0	1
76	1192603016	307	1	1	2.99		35.61	1
76	1192603016	308	0	1	0		0	1
76	1192603016	309	0	1	0		0	1
76	1192603016	310	0	1	0		0	1
76	1192603016	311	0	1	0		0	1
76	1192603016	312	0	1	0		0	1
76	1192603016	313	0	1	0		0	1
76	1192603016	314	0	1	0		0	1
76	1192603016	315	0	1	0		0	1
76	1192603016	316	0	1	0		0	1
76	1192603016	317	0	1	0		0	1
76	1192603016	318	0	1	0		0	1
76	1192603016	319	0	1	0		0	1
76	1192603016	320	0	1	0		0	1
76	1192603016	321	0	1	0		0	1
76	1192603016	322	0	1	0		0	1
76	1192603016	323	0	1	0		0	1
76	1192603016	324	0	1	0		0	1
76	1192603016	325	0	1	0		0	1
76	1192603016	326	0	1	0		0	1
76	1192603016	327	0	1	0		0	1

Figure 7: Movement File data

Dominick Stores data

This data provides details about various Dominick's store locations which includes the store number, city, price tier, pricing zone, zip code, and address. The price tiers are categorized as High, Medium, Low, and Cub-Fighter, reflecting different pricing strategies within 16 zones across the Chicago area. Some stores are marked as closed, while others lack zone assignments, likely because they opened after 1992. This dataset helps in understanding the geographical and pricing distribution of Dominick's stores during the research period.

Week Decode Table

This table provides a mapping of calendar weeks to database weeks used in the SAS files of the Dominick's database. Each row lists a specific week number along with its corresponding start and end dates. Special events such as holidays like Halloween, Thanksgiving, Christmas, and New Year's are also noted. This data helps identify the exact week for which a data point was recorded, making it easier to correlate events with sales and other relevant metrics in the database.

Details about Metadata

Metadata provides descriptive details about the underlying data, offering insights without the need to examine the actual values. This information enhances understanding and facilitates quicker access to relevant data. Now, let's review the metadata for all the OLTP files listed below:

Customer Count File

The customer count file contains key in-store data, including the store's unique identifier, sales figures, customer purchase counts, sales volume by item category, and the number of coupons redeemed for different product types. It offers a detailed overview of the store's sales activities.

Variable	Description	Type	Length
DATE	Date of the Observation	Character	6
Week	Week Number	Numeric	8
Store	Store Code	Numeric	8
BAKCOUP	Bakery Coupons Redeemed	Numeric	8
BAKERY	Bakery Sales in Dollars	Numeric	8
BEER	Beer Sales in Dollars	Numeric	8
BOTTLE	Bottle Sales in Dollars	Numeric	8
BULK	Bulk Sales in Dollars	Numeric	8
BULKCOUP	Bulk Coupons Redeemed	Numeric	8
CAMERA	Camera Sales in Dollars	Numeric	8
CHEESE	Cheese Sales in Dollars	Numeric	8
CONVFOOD	Conventional Foods Sales in Dollars	Numeric	8
COSMCOUP	Cosmetics Coupons Redeemed	Numeric	8
COSMETIC	Cosmetics Sales in Dollars	Numeric	8
CUSTCOUN	Customer Count	Numeric	8
DAIRCOUP	Dairy Coupons Redeemed	Numeric	8
DAIRY	Dairy Sales in Dollars	Numeric	8
DELI	Deli Sales in Dollars	Numeric	8
DELICOUP	Deli Coupons Redeemed	Numeric	8
DELIEXPR	Deli Express Sales in Dollars	Numeric	8
DELISELF	Deli Self Service Sales in Dollars	Numeric	8
FISH	Fish Sales in Dollars	Numeric	8
FISHCOUP	Fish Coupons Redeemed	Numeric	8
FLORAL	Floral Sales in Dollars	Numeric	8
FLORCOUP	Floral Coupons Redeemed	Numeric	8
FROZCOUP	Frozen Items Coupons Redeemed	Numeric	8
FROZEN	Frozen Items Sales	Numeric	8

FTGCCOUP	Food-to-Go Coupons Redeemed	Numeric	8
FTGCHIN	Food-to-Go Chinese Sales in Dollars	Numeric	8
FTGICOUP	Food-to-Go Coupons Redeemed	Numeric	8
FTGITAL	Food-to-Go Italian Sales in Dollars	Numeric	8
GM	General Merchandise Sales in Dollars	Numeric	8
GMCOUP	General Coupons Redeemed	Numeric	8
GROCCOUP	Grocery Coupons Redeemed	Numeric	8
GROCERY	Grocery Sales in Dollars	Numeric	8
HABA	Health and Beauty Aids Sales in Dollars	Numeric	8
HABACOUP	Health and Beauty Aids Coupons Redeemed	Numeric	8
JEWELRY	Jewelry Sales in Dollars	Numeric	8
LIQCOUP	Liquor Coupons Redeemed	Numeric	8
MANCOUP	Manufacturer Coupons Redeemed	Numeric	8
MEAT	Meat Sales in Dollars	Numeric	8
MEATCOUP	Meat Coupons Redeemed	Numeric	8
MEATFROZ	Meat-Frozen Sales in Dollars	Numeric	8
MISCSCP	Misc. Coupons Redeemed	Numeric	8
MVPCLUB	MVP	Numeric	8
PHARCOUP	Pharmacy Coupons Redeemed	Numeric	8
PHARMACY	Pharmacy Sales in Dollars	Numeric	8
PHOTCOUP	Photo Coupons Redeemed	Numeric	8
PHOTOFIN	Photo	Numeric	8
PRODCOUP	Produce Coupons Redeemed	Numeric	8
PRODUCE	Produce Sales in Dollars	Numeric	8
PROMCOUP	Promotion Coupons Redeemed	Numeric	8
PROMO	Promotion Sales in Dollars	Numeric	8
SALADBAR	Salad Bar Sales in Dollars	Numeric	8
SALCOUP	Salad Coupons Redeemed	Numeric	8
SPIRITS	Spirits Sales in Dollars	Numeric	8
SSDELICP	Self Service Deli Sales in Dollars	Numeric	8

VIDCOUP	Video Coupons Redeemed	Numeric	8
VIDEO	Video Sales in Dollars	Numeric	8
VIDEOREN	Video Rentals (Dollar Amounts)	Numeric	8
WINE	Wine Sales in Dollars	Numeric	8

Demographic File

The demographic information offers essential statistics about individual stores, covering aspects such as the percentage of customers in different age groups, marital status, education levels, household characteristics, and various other pertinent factors.

Variable Name	Description
age9	% Population under age 9
age60	% Population over age 60
ethnic	% Blacks & Hispanics
educ	% College Graduates
nocar	% With No Vehicles
income	Log of Median Income
incsigma	Std dev of Income Distribution (Approximated)
hsizeavg	Average Household Size
hsize1	% of households with 1 person
hsize2	% of households with 2 persons
hsize34	% of households with 3 or 4 persons
hsize567	% of households with 5 or more persons
hh3plus	% of households with 3 or more persons
hh4plus	% of households with 4 or more persons
hhsingle	% of households with 1 person
hhlarge	% of households with 5 or more persons
workwom	% Working Women with full-time jobs
sinhouse	% Detached Houses
density	Trading Area in Sq Miles per Capita

hval150	% of Households with Value over $150,000
hval200	% of Households with Value over $200,000
hvalmean	Mean Household Value (Approximated)
single	% of Singles
retired	% of Retired
unemp	% of Unemployed
wrkch5	% of working women with children under 5
wrkch17	% of working women with children 6 - 17
nwrkch5	% of non-working women with children under 5
nwrkch17	% of non-working women with children 6 - 17
wrkch	% of working women with children
nwrkch	% of non-working women with children
wrkwch	% of working women with children under 5
wrkwnch	% of working women with no children
telephn	% of households with telephones
mortgage	% of households with mortgages
nwhite	% of population that is non-white
poverty	% of population with income under $15,000
shopcons	% of Constrained Shoppers
shophurr	% of Hurried Shoppers
shopavid	% of Avid Shoppers
shopstr	% of Shopping Strangers
shopunft	% of Unfettered Shoppers
shopbird	% of Shopper Birds
shopindx	Ability to Shop (Car and Single Family House)
shpindx	Ability to Shop (Car and Single Family House)

UPC file

The metadata in the UPC files provides valuable insights into individual UPC entries within a particular category. It includes details such as the commodity code, item code, product name, size,

and the quantity of items in a case. This information is essential for deepening our understanding of the products linked to these UPCs.

Variable	Description	Type	Length
upc	UPC Number	Numeric	8
com_code	Dominick's Commodity Code	Numeric	8
nitem	Dominick's Item Code	Numeric	8
descrip	Product Name	Character	20
size	Product Size	Character	6
case	Number of Items in a Case	Numeric	8

Movement File

The metadata for movement data gives detailed insights into sales for individual UPCs at the store level within a specific category. It includes important information like the number of units sold per UPC at a particular store during a specific week, along with details about pricing and profit.

Variable	Description	Type	Length
upc	UPC Number	Numeric	8
store	Store Number	Numeric	3
week	Week Number	Numeric	3
move	Number of Units Sold	Numeric	8
price	Retail Price	Numeric	8
qty	Number of Items Bundled Together	Numeric	3
profit	Gross Margin	Numeric	8
sale	Sale Code (B, C, S)	Character	8
ok	1 for Valid Data, 0 for Trash	Numeric	3

Dominick Store Data

The metadata provided by Dominick's Store gives us detailed information about the stores that are currently open. This includes their unique identification codes, full physical addresses, and details about the level of the city they are in.

Variable	Description	Type	Length
Store	Store Code	Numeric	3
City	Name of city	Character	20
Price Tier	Segmentation of prices	Character	10
Zone	Zone where the store is located	Numeric	6

Zip Code	Zip code of the store	Numeric	8
Address	Address of the store	Character	24

Week Decode Data

The Decode Table for the week provides crucial information about important events, such as holidays and peak shopping days. It contains relevant details, including the week number and the start and end dates of these events.

Variable	Description	Type	Length
Week#	Week Number	Numeric	3
Start	Start Date of that particular week	Date	20
End	End Date of that particular week	Date	20
Special Events	Holidays or Festival periods	Character	20

Entity Relationship Diagram

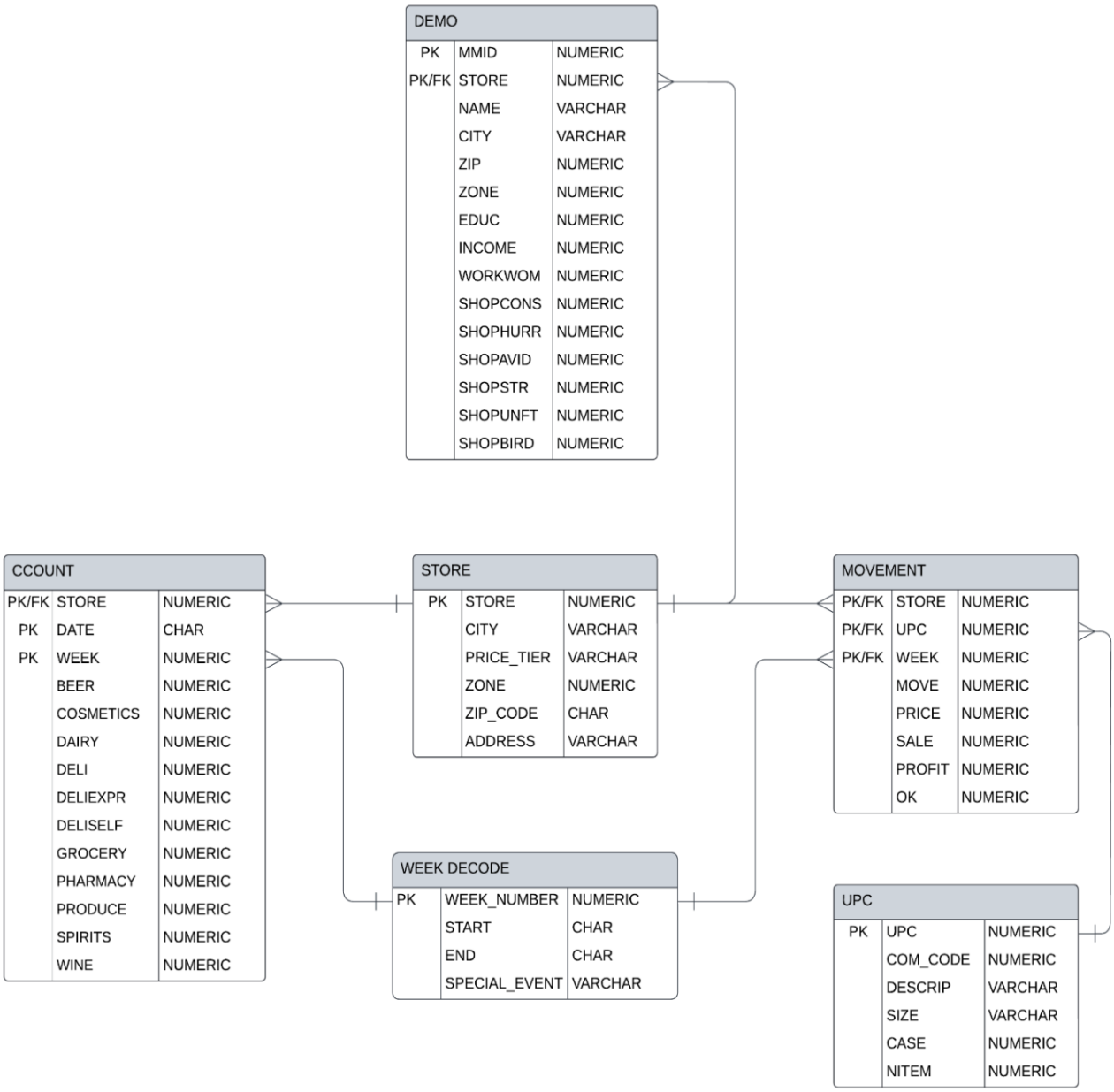


Fig 8: The Entity Relationship Diagram

Section 2: Subject Area Understanding

Research on Retail Domain

Implementing a data warehouse solution in the retail domain requires a deep understanding of data complexities and the ability to translate insights into actionable strategies. Through the analysis of several research papers, we explored how data can address key retail challenges, particularly in the context of Dominick's Finer Foods (DFF). These studies offered insights into pricing strategies, promotional effectiveness, and assortment planning, which are critical areas of concern for DFF, given its historical struggles with profitability, customer retention, and adapting to market dynamics.

The research on store-level price elasticity [1] identifies several factors influencing pricing strategies, including demographic profiles, category-specific dynamics, and temporal variations. For example, stores in high-income neighborhoods may exhibit lower price sensitivity compared to those in more cost-conscious areas. The Demographics File from the Dominick's dataset, which includes variables such as income levels, education percentages, and household sizes, can help DFF customize pricing strategies for different regions. Additionally, the Movement File's detailed sales data enables an analysis of how specific product categories, such as frozen goods or beverages, respond to price changes over time, allowing for more nuanced pricing models. Temporal variations, such as increased grocery purchases during holiday seasons (identified using the Week Decode Table), can further guide pricing adjustments to maximize revenue during peak periods.

Another study focusing on promotion profitability [2] sheds light on how different types of promotions—such as Bonus Buys, Coupons, and Simple Price Reductions—affect sales and margins. For instance, the Movement File categorizes promotional activities, enabling an analysis of which methods work best for specific products. A detailed comparison of national versus private label brands in the UPC File can reveal patterns; for example, national brands might attract higher sales during price reductions, whereas private labels may benefit more from bonus buy schemes. Additionally, understanding the interaction effects between promotions, brand types, and store locations can help DFF design store-specific campaigns, boosting overall profitability.

The third paper, focused on assortment planning [3], highlights the importance of dynamic inventory management and demand forecasting. DFF faced significant issues with product assortment, especially after Safeway's acquisition when local products were replaced by unfamiliar private-label brands. Using the UPC and Movement Files, DFF can analyze product substitution patterns during stock-outs—for example, identifying if customers switch from one brand of coffee to another or choose a different product category altogether. By integrating demographic data, DFF can also predict demand for specific items based on local preferences, ensuring popular products are always in stock. For example, stores in family-dense neighborhoods might prioritize high-volume staples like dairy or cereals, while urban locations could focus on ready-to-eat and convenience foods.

Several problems emerge as particularly relevant to DFF's case:

1. **Pricing and Profitability Challenges:** Balancing competitive pricing with profit margins was a major hurdle for DFF, especially as rivals like Walmart entered the market with aggressive pricing strategies.

2. **Product Assortment and Branding Issues:** The replacement of local favorites with unfamiliar private-label brands caused customer dissatisfaction, impacting sales.

3. **Adapting to Consumer Preferences:** Failing to respond to changing consumer demands and regional preferences led to a gradual decline in market share.

By leveraging insights from the research and utilizing the rich historical data in Dominick's dataset, DFF can construct a data-driven strategy to address these challenges. For instance, by identifying the most elastic product categories, they can implement targeted promotions and pricing schemes. Similarly, dynamic assortment planning, informed by substitution patterns and long-term demand forecasting, can minimize stockouts and overstock situations. Overall, the integration of these research findings into DFF's data warehouse initiative can not only mitigate past issues but also position the company to thrive in an increasingly competitive retail landscape.

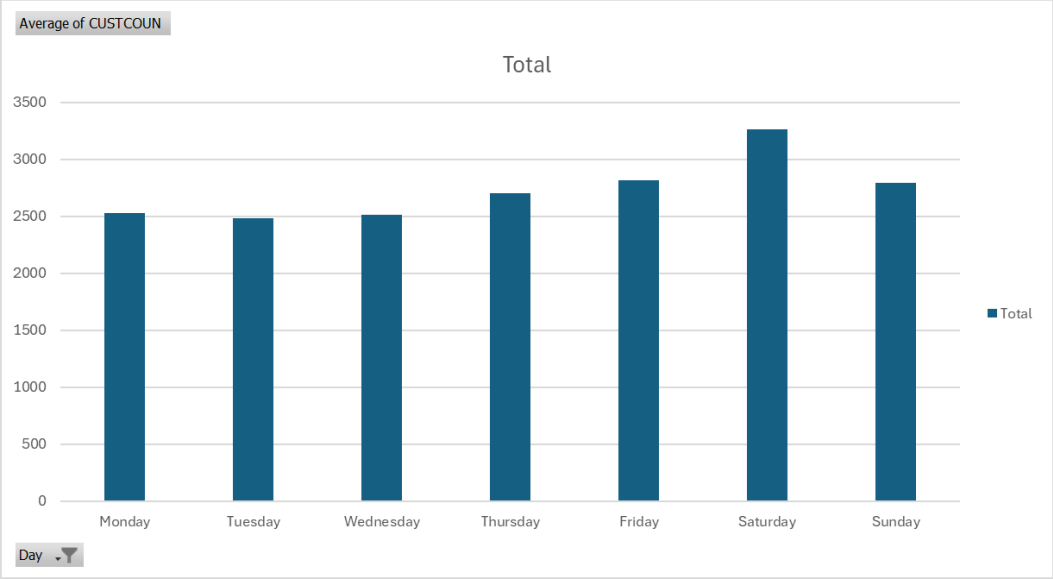
Business Questions

Here are the ten business questions we proposed:

1. Which day of the week has the highest customer traffic?

Why it's useful
By identifying the days with the highest volume of visitors, the business can make sure that staff members are available when needed to help customers have a better experience.
Data being used:
The data is present in the Customer Count Files where variables like CUSTCOUN (Customer Count) and DATE can help determine the average number of customers on each day of the week.

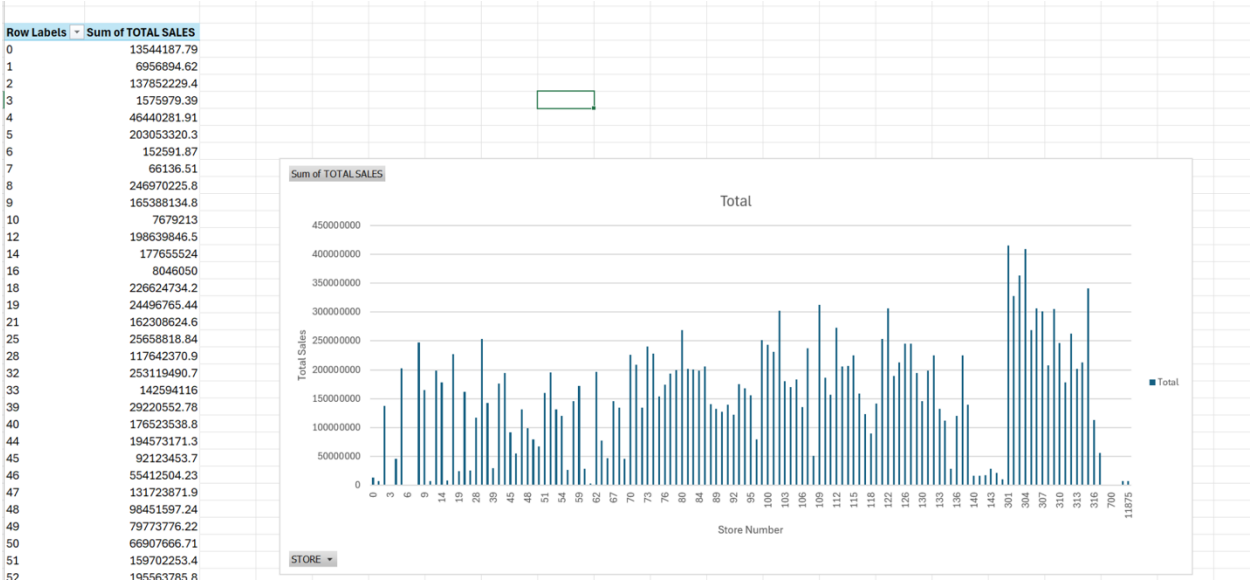
Row Labels	Average of CUSTCOUN
Monday	2530.54536
Tuesday	2483.945205
Wednesday	2512.851477
Thursday	2705.874419
Friday	2813.80885
Saturday	3266.168533
Sunday	2794.256307
Grand Total	**2730.009776**



Pivot Table and Bar Chart Containing Days and the Average of the Customer Count

2. Which store has the highest total sales?

Why it's useful:
Identifying top-performing stores can help the business analyze what factors are contributing to their success. They can make informed decisions on the product, brands and categories which these stores possess and make sure they are always stocked up in the inventory. It also provides a good means to compare other stores and see what success factors they can implement in other stores too.
Data being used:
We would be using the customer count file to sum up all the category sales for each store and check which store has the maximum sales.

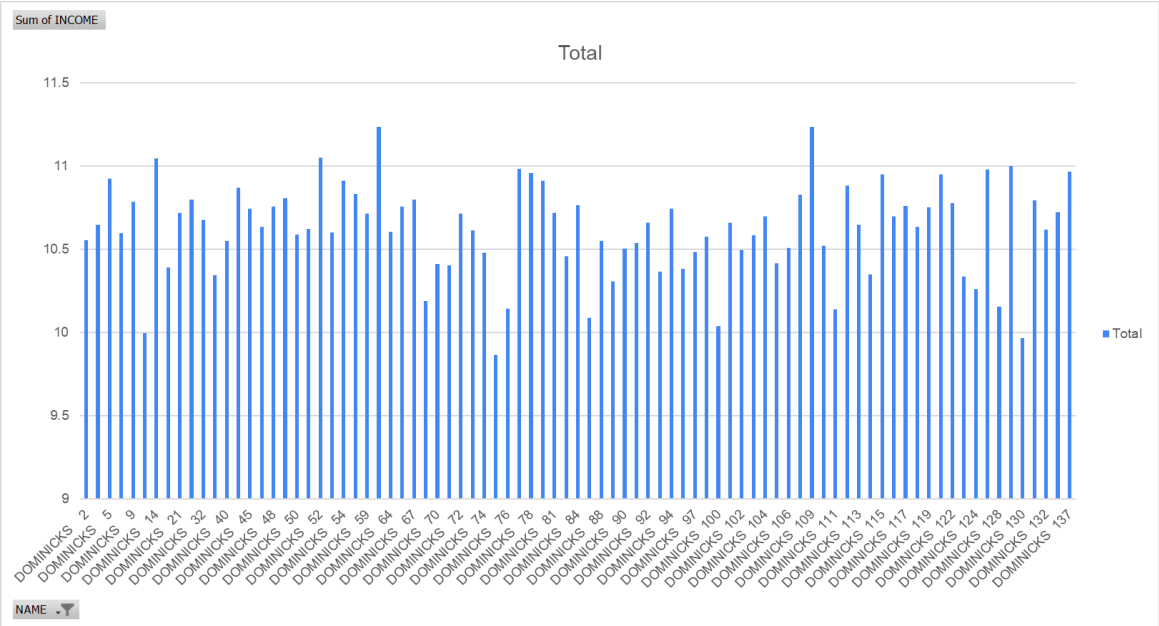


Row Labels	Sum of TOTAL SALES
0	13544187.79
1	6956894.62
2	137852229.4
3	1575979.39
4	46440281.91
5	203053320.3
6	152591.87
7	66136.51
8	246970225.8
9	165388134.8
10	7679213
12	198639846.5
14	177655524
16	8046050
18	226624734.2
19	24496765.44
21	162308624.6
25	25658818.84
28	117642370.9
32	253119490.7
33	142594116
39	29220552.78
40	176523538.8
44	194573171.3
45	92123453.7
46	55412504.23
47	131723871.9
48	98451597.24
49	79773776.22
50	66907666.71
51	159702253.4
52	195563785.8

Using Pivot Table and Charts we analyzed Store 301 has the highest total sales.

3. Which stores are located in areas with highest median income?

Why it's useful:
DFF can better fit the preferences of higher-income consumers by customizing product offers, pricing policies, and marketing initiatives with the help of this study.
Data being used:
The Store-specific Demographic data is to be analyzed in this question, specifically the income variable.



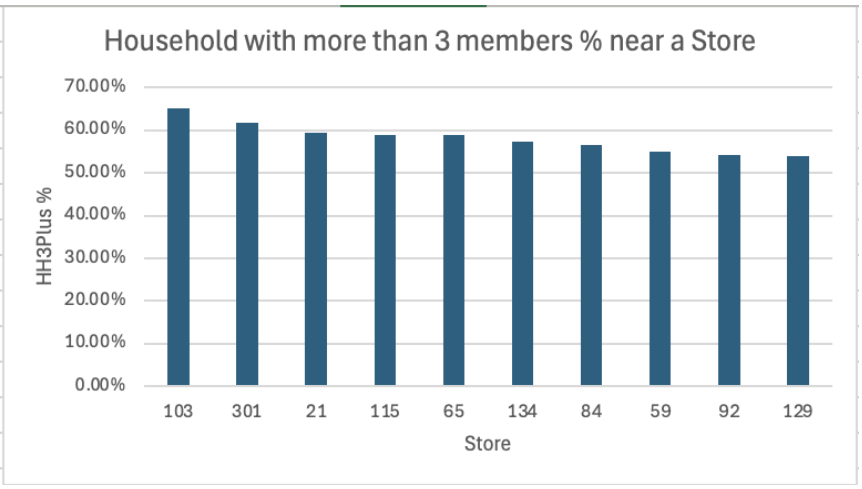
Based on the above chart, Dominick's 64 has the highest median income.

4. Which store has the highest percentage of households with more than 3 people?

Why it's useful:
Understanding the demographic composition of households can help DFF tailor its products' marketing strategies to better meet the needs of families. Stores with higher percentages of larger households may benefit from promotions, leading to increased sales and customer satisfaction.

Data being used:
This iincludes household size HHS3PLUS data from the Demographics File (demo.csv), specifying the percentage of households with more than three people, linked with store locations to identify which stores have the highest percentages.

Store	Average of HH3PLUS
103	65.04%
301	61.81%
21	59.50%
115	58.97%
65	58.85%
134	57.23%
84	56.51%
59	54.87%
92	54.24%
129	54.00%
Grand Total	**58.10%**



5. How many cosmetics coupons were redeemed in each store?

Why it's useful:
Cosmetics promotions can boost sales. This question helps DFF measure the impact of cosmetics-specific promotions across stores.
Data being used:
The data used for this question comes from the ccount.csv file, which provides coupon redemption data for cosmetics products. We will summarize this data to show the total number of cosmetics coupons redeemed across different stores.

Store	Sum of COSMCOUP
2	2428
3	2108
4	1263
5	636
6	168
7	137
Grand Total	**6741**

6. What is the average price of frozen dinners for specific UPCs across all stores?

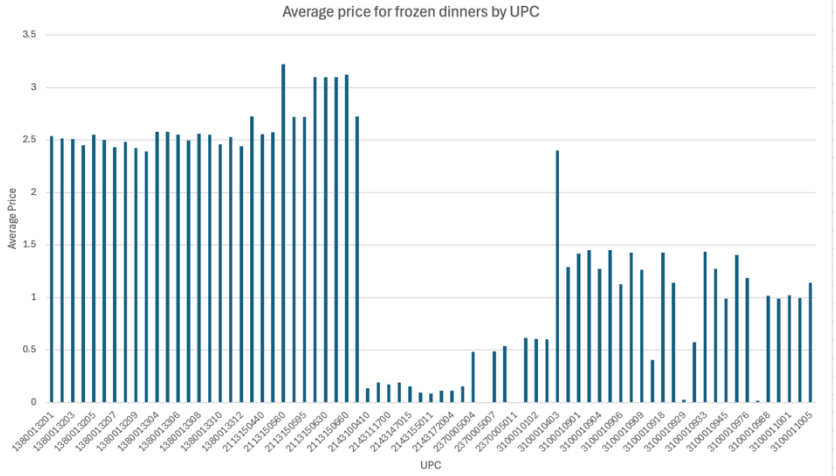
Why it's useful:
Frozen items are crucial for long shelf-life products. Monitoring the average price of frozen dinners across various stores allows DFF to ensure competitive pricing while maximizing profitability.

They can also maintain consistency in pricing across different stores. Understanding pricing dynamics can help identify opportunities for promotions or adjustments in pricing strategies.

Data being used:

This question leverages data from the wfrd.csv. By examining specific UPCs, DFF can assess pricing trends across different stores

UPC	Average of PRICE
1380013201	2.535350939
1380013202	2.51583547
1380013203	2.509768119
1380013204	2.451051396
1380013205	2.549174633
1380013206	2.501674435
1380013207	2.432633918
1380013208	2.482808203
1380013209	2.42186672
1380013210	2.390757576
1380013304	2.578802751
1380013305	2.580820686
1380013306	2.55305575
1380013307	2.495028096
1380013308	2.55954471
1380013309	2.551895941
1380013310	2.458884113
1380013311	2.526152321
1380013312	2.441966316
2113150434	2.723371631
2113150440	2.555020408
2113150475	2.575780533
2113150560	3.222186524
2113150575	2.720095847
2113150595	2.719414151
2113150605	3.098280142
2113150630	3.097879195



7. Which store has the higher % of college graduates in its region?

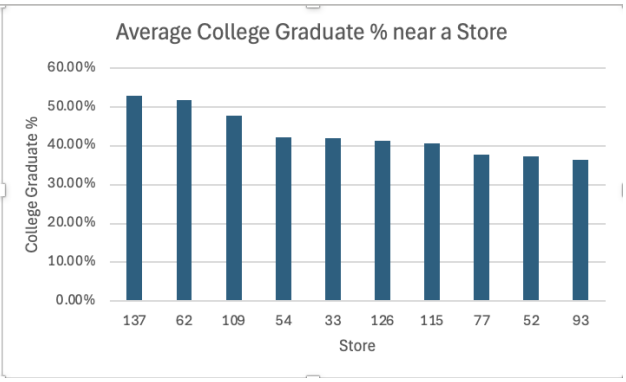
Why it's useful:

Finding stores in areas with more college graduates can help DFF diversify its product offerings. This group of consumers often prefers premium or health-focused products. By adjusting marketing strategies to appeal to educated shoppers, DFF can boost customer satisfaction and increase sales.

Data being used:

The demographic data from the demo.csv file is used to assess the percentage of college graduates in different regions across stores. This information aids DFF in understanding consumer preferences and aligning their product offerings accordingly.

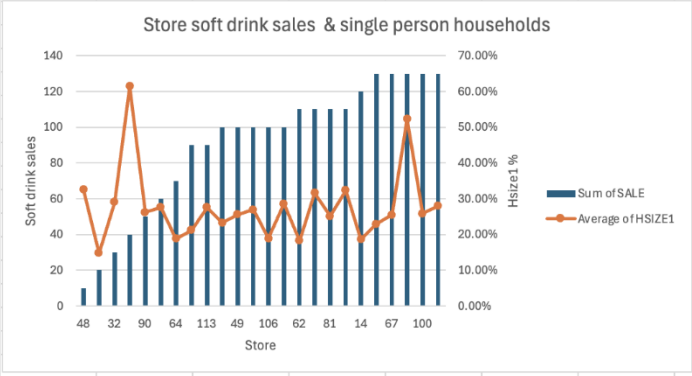
Store	Average of EDUC
137	52.84%
62	51.78%
109	47.69%
54	42.11%
33	41.97%
126	41.32%
115	40.63%
77	37.69%
52	37.29%
93	36.30%
Grand Total	**42.96%**



8. Which stores have the lowest sales of soft drinks and the highest percentage of single-person households?

Why it's useful:
Analyzing stores with low soft drink sales alongside high percentages of single-person households can help DFF devise targeted marketing strategies. Promotions tailored to convenience and smaller packaging may enhance sales in these regions.
Data being used:
This analysis utilizes sales data from the wsdr.csv (soft drinks) and demographic data from the demo.csv file. The combination provides insights into consumer behavior related to soft drink purchases and household composition.

Store	Sum of SALE	Average of HSIZE1
48	10	32.49%
109	20	14.77%
32	30	29.03%
33	40	61.40%
90	50	26.20%
91	60	27.55%
64	70	18.80%
9	90	21.15%
113	90	27.57%
88	100	23.23%
49	100	25.55%
76	100	26.97%
106	100	18.86%
95	100	28.60%
62	110	18.26%
68	110	31.61%
81	110	25.14%
86	110	32.37%
14	120	18.63%
73	130	22.90%
67	130	25.49%
75	130	52.29%
100	130	25.86%
53	130	27.94%
Grand Total	**2170**	**26.67%**

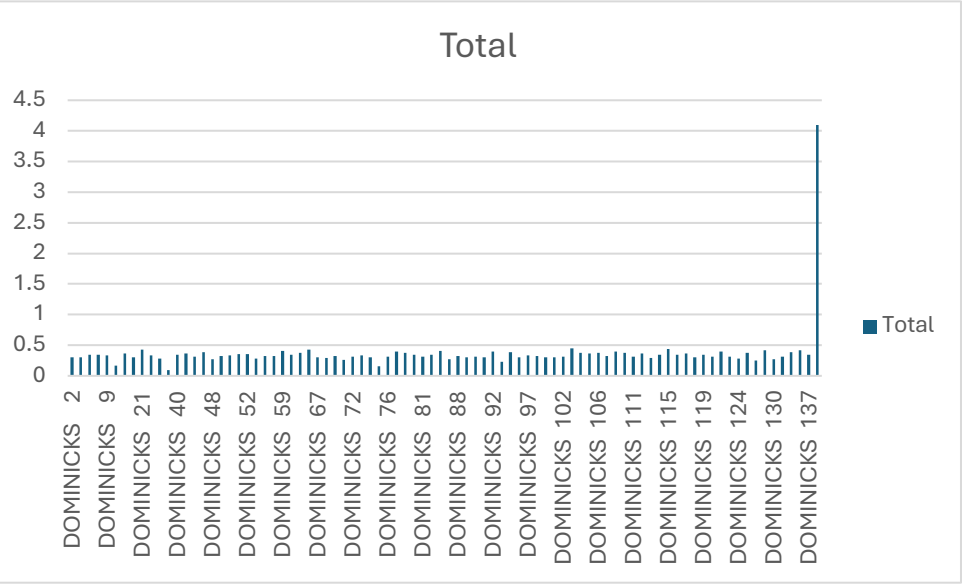


9. Which store is located in an area with the highest concentration of families and how does this impact sales of family-oriented products?

Why it's useful:
Identifying the stores located in areas with the highest concentration of larger families (households with 3 or more people) can help optimize product offerings and promotions. Family-oriented products, bulk items, or deals designed for larger households can be better targeted, improving sales and customer satisfaction. This also helps in managing inventory more effectively in stores that serve a predominantly family-based demographic.

Data being used:
The data being used comes from the demographic file and includes the fields hsize34, representing the percentage of households with 3 or 4 persons.

Row Labels	Sum of HSIZE34
DOMINICKS 2	0.301094246
DOMINICKS 4	0.303677092
DOMINICKS 5	0.342980009
DOMINICKS 8	0.343011107
DOMINICKS 9	0.332945624
DOMINICKS 12	0.16737675
DOMINICKS 14	0.366941592
DOMINICKS 18	0.306565707
DOMINICKS 21	0.430864545
DOMINICKS 28	0.32944198
DOMINICKS 32	0.285876172
DOMINICKS 33	0.092028602
DOMINICKS 40	0.342959825
DOMINICKS 44	0.364044028
DOMINICKS 45	0.315641899
DOMINICKS 47	0.386598945
DOMINICKS 48	0.270638147
DOMINICKS 49	0.322921348
DOMINICKS 50	0.335853379
DOMINICKS 51	0.354280964
DOMINICKS 52	0.349756517
DOMINICKS 53	0.276557146
DOMINICKS 54	0.324789152
DOMINICKS 56	0.319131021
DOMINICKS 59	0.408007478
DOMINICKS 62	0.348528739
DOMINICKS 64	0.374283374
DOMINICKS 65	0.423807318

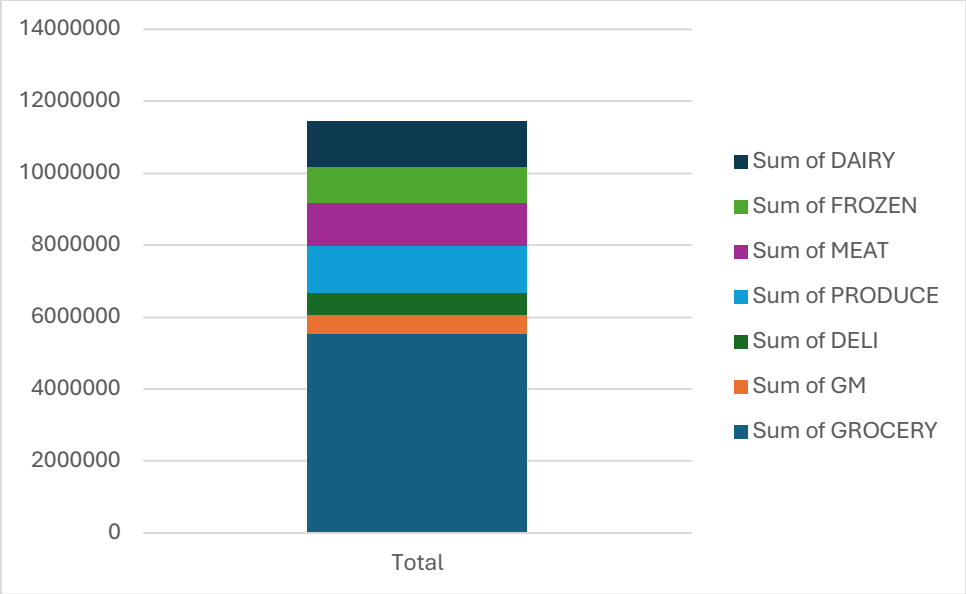


10. Which are the top 7 product categories that saw the highest selling record over the 1 year(1995)?

Why it's useful:
This insight helps optimize inventory management, marketing strategies, and product allocation. By identifying the top-selling product category, the business can make data-driven decisions about restocking popular items, enhancing supply chain efficiency, and targeting marketing efforts towards these high-demand products. It also provides the opportunity to adapt to changing consumer trends and focus on categories that generate the most revenue and profitability.

Data being used:
The data comes from the ccount file and includes sales records for each product category over the year 1995, focusing on both the highest and lowest-selling categories.

Sum of GROCERY	Sum of GM	Sum of DELI	Sum of PRODUCE	Sum of MEAT	Sum of FROZEN	Sum of DAIRY
5534382.85	520004.01	623559.16	1300791.03	1201166.2	1014070.4	1256179.14



Selected Business Questions

1. Which stores are located in areas with highest median income?
2. What is the average price of frozen dinners for specific UPCs across all stores?
3. Which stores have the lowest sales of soft drinks and the highest percentage of single-person households?
4. Which store is located in an area with the highest concentration of families and how does this impact sales of family-oriented products?
5. Which are the top 7 product categories that saw the highest selling record over the 1 year(1995)?

With key retail business inquiries now clarified through the review of academic research, Ralph Kimball's methodology offers a practical framework for developing data marts tailored to address these specific needs. By employing Kimball's dimensional modeling approach, retail data marts can be designed to effectively support and resolve the identified business challenges.

Section 3: Independent Data Marts design using Kimball's approach

Kimball's methodology for data warehouse design is an iterative and modular approach that primarily focuses on developing independent data marts aligned with specific business processes or departments. These independent data marts are later integrated into a cohesive enterprise data warehouse through conformed dimensions which ensures consistency across the entire organization.
This methodology is particularly effective in addressing business needs while maintaining scalability and flexibility for future expansion.

Steps for Kimball's Methodology

Below is a comprehensive breakdown of each step involved in Kimball's methodology.

Step 1. Requirement Analysis

The foundation or the starting point of Kimball's methodology is a thorough requirements analysis which involves gathering and understanding the key business questions, objectives, and data analysis needs. This step sets the scope and the direction for the entire data warehouse project. It involves –

- Get Business Questions: We completed this step in Report 1. As part of this report, we gathered requirements to address critical business questions or operational issues for Dominick's Finer Foods like –
    - *Which stores are located in areas with highest median income?*
    - *What is the average price of frozen dinners for specific UPCs across all stores?*
    - *Which stores have the lowest sales of soft drinks and the highest percentage of single-person households?*
    - *Which store is located in an area with the highest concentration of families and how does this impact sales of family-oriented products?*
    - *Which are the top 7 product categories that saw the highest selling record over the 1 year(1995)?*
- Looking for issues and Business Objectives: This includes understanding the broader business context which includes the objectives and challenges that the data warehouse should address. At DFF, we can address challenges such as optimizing product sales in family-concentrated areas. This is just one example of many.
- Collecting Analysis Requirements: To answer the business questions, we require the specific data that pertains to that business. This includes identifying the data sources, metrics, and dimensions that will be used in the data warehouse.

Step 2. Build the Matrix

In this step, we focus on organizing the data warehouse structure by building a matrix that maps the data marts against the suitable dimensions. The matrix is a visual representation of the relationship between data marts and dimensions on the x-y axis. It aims to ensure that each data mart contains the necessary dimensions for analysis to support the business questions we identified earlier. This includes –

- Listing the data marts in the matrix: We start by identifying the data marts to be included in the matrix. It is suggested to initiate with single-source data marts. These marts are easier to handle and provide quicker results. Multiple-source data marts can be created by combining similar single-source marts.
- Including the dimensions in the matrix: The next step is to mention the dimensions that will be used across data marts. These dimensions will be essential for analyzing the facts in the data marts. Common possible dimensions in the case of DFF could consist of Product, Time, Store, etc.

- Marking the intersection in the matrix: The matrix is completed when we mark the intersection points where the data and dimensions overlap. This step makes sure that the required dimensions are linked to each data mart.

Step 3. Design Fact Tables

The design of fact tables is central to the data warehouse's functionality. Fact tables store the quantitative metrics that answer the business questions. Each fact table is linked with dimension tables that provide context for the data such as which product was sold. This includes –

- Choosing the Data Mart: We select a data mart from the matrix as the foundation for the fact table
- Declaring the grain: It is crucial to define the level of detail for each fact record.
- Choosing facts: Identifying the specific metrics to be included in fact tables like total sales or units sold.
- Including derived facts: In this step, we determine if any calculated metrics should be included in the fact table.
- Identifying base facts and derived facts: This step involves removing duplicate facts, identifying additional calculations, cross referencing base facts and getting the approval for derived facts.
- Designing a fact table diagram: We develop a visual representation of the fact table and its relationship to the relevant dimensions.

Step 4. Design the Dimension Tables

Dimension tables provide descriptive information related to the facts, such as store locations, product categories, or time periods. The dimension tables enable the users to query and analyze information across different attributes like which products performed best in specific stores.

This step involves –

- Drawing a dimension table: We create a diagram to display each dimension's structure, grain, and key attributes.
- Showing Hierarchies: We identify and display any hierarchies within the dimensions. A common example is date hierarchy like year, quarter, month, week, and day for the time dimension
- Providing descriptive data: Ensure that each dimension contains the required descriptive information to support the analysis.

Step 5. Feedback for Design

When the design is ready to be finalized, it is essential to gather feedback from both technical teams, business users, and other stakeholders. This ensures that the data warehouse schema meets the technical requirements and provides the necessary insights to the business.

Feedback for design includes –

- Getting inputs from the IS team: It is crucial to validate the design with the information security team to ensure that it supports the necessary data infrastructure.
- Feedback from business users: Confirm with business stakeholders that the model aligns with their reporting and analysis needs.
- Present to the business users: The finalized design is then presented to the business users to verify that it handles the identified business issues.

Step 6. Data Sourcing

Data sourcing involves identifying the appropriate data sources, transforming the data as needed, and then loading that transformed data into the data warehouse. This step is significant for ensuring that the data warehouse contains accurate, clean, and up-to-date information.

Efficient data sourcing enables effective analysis and reporting through data marts supporting high-quality data. Data sourcing consists of –

- Identifying Data Sources: It is crucial to identify the informal (personal databases) and formal (operation systems, OLTP) sources of data.
- Creating Data Source Definitions: This step includes defining attributes such as the source system, business owner, IS owner, platform, description, & location for each data source. These definitions help standardize the data extraction process.
- Plan data extraction: Planning the process for extracting data from source systems and loading it into the staging area ensures data accessibility, accuracy, and mapping source data to staging tables.
- Data loading from staging to presentation server: Once, the data is cleaned and transformed in the staging area, it is loaded into the presentation server or data marts. The final steps involve mapping staging tables to the appropriate fact and dimension tables in data marts.

Significance of Kimball's methodology for independent data marts

While creating independent data marts, Kimball's methodology provides a structured approach that aligns with business requirements. Following are the reasons why Kimball's methodology can be advantageous for creating independent data marts –

- Targeted Insights: Each data mart is created to address specific business questions that allow for quicker delivery of insights without any need to develop a thorough enterprise data warehouse upfront.
- Conformed Dimensions: The use of consistent dimensions across multiple data marts allows for seamless integration because it provides a unified view of the organization and its data.
- Scalability & Flexibility: New data marts can be added as the business evolves, allowing the organization to grow its data warehouse while maintaining existing structures with minimal downtime
- Data consistency and accuracy: A well-defined ETL process ensures that data is cleaned, standardized, and transformed through data staging before being loaded into the data marts, which encourages reliable analysis and reporting

DW Logical Design (Star Schema Design)

In this section, we will be following Kimball's Methodology to build a star schema that corresponds to each business question we have chosen for the DFF data warehouse project.

Step 1: Requirement Analysis

In Report 1, we completed the requirements analysis and came up with 5 business questions (mentioned in section 2) based on which we would be creating the data marts:

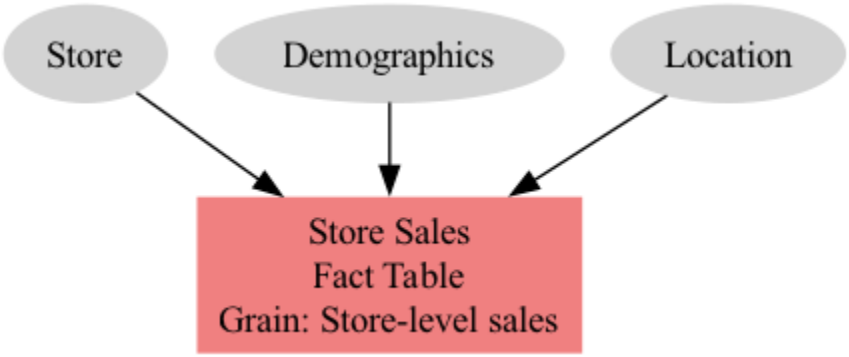
Step 2: Building the Data Matrix

Kimball Matrix for the Data Marts

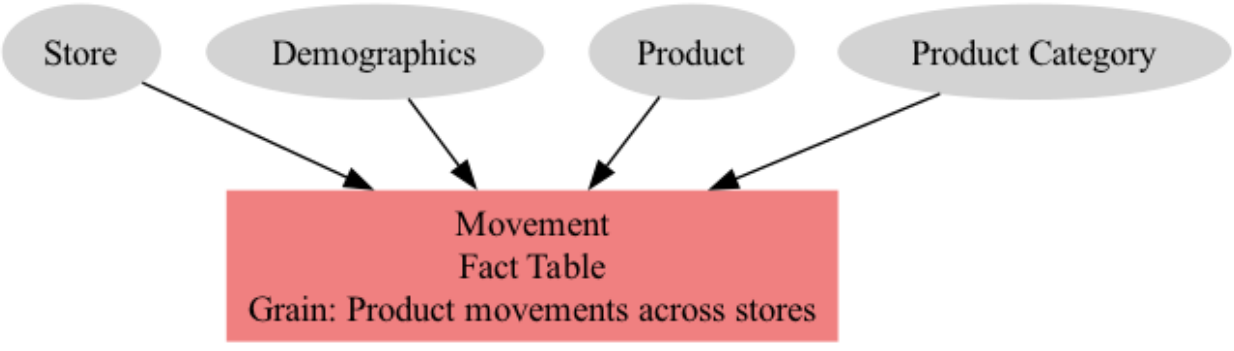
Data Mart ↓ / Dimension →	**Store_dim**	**Demographic_dim**	**Prod_dim**	**Time_dim**
Store Sales	X	X		
Movement details	X	X	X	
Top Category Sales			X	X

Step 3: Design Fact Tables

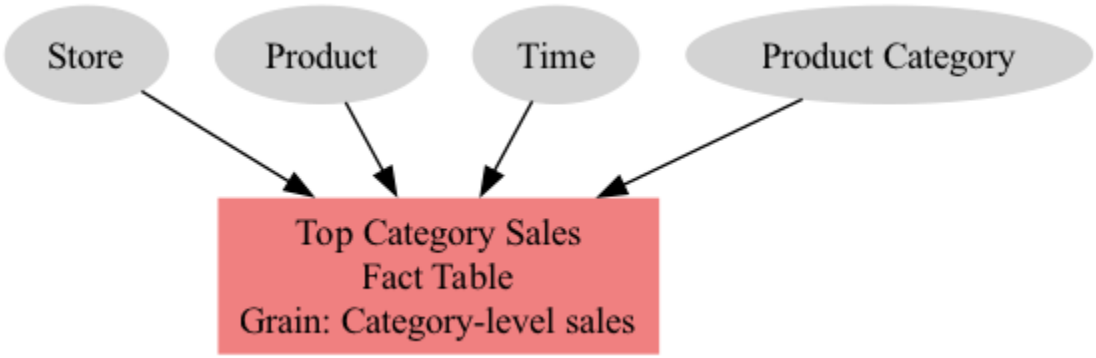
1. **Store Sales Fact Table**



2. **Movement Fact Table**

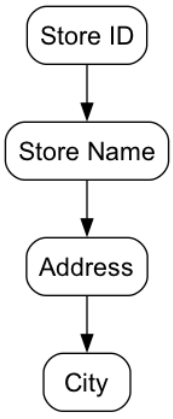


3. **Top Category Sales Fact Table**

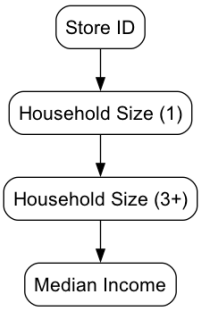


Step 4: Design Dimension Tables

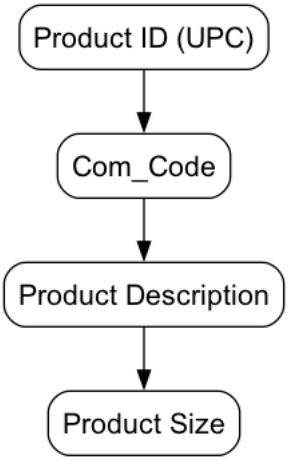
1. **Store Dimension (store_dim) Table**



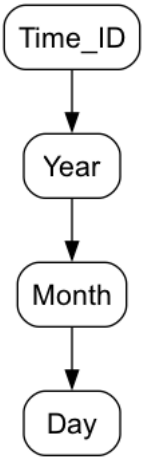
2. **Demographic Dimension (demographic_dim) Table**



3. Product Dimension (product_dim) Table

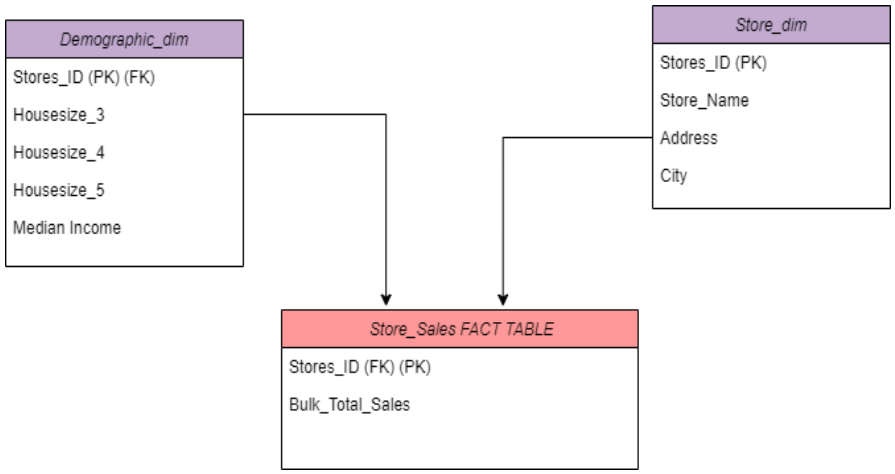


4. Time (time_dim) Dimension Table



Step 5: Finalized Design (Star Schema)

- **Store Specific**



The store specific data mart structure supports two of our business questions - **4.3** and **4.9**. Here's how:

Business Question 4.3:

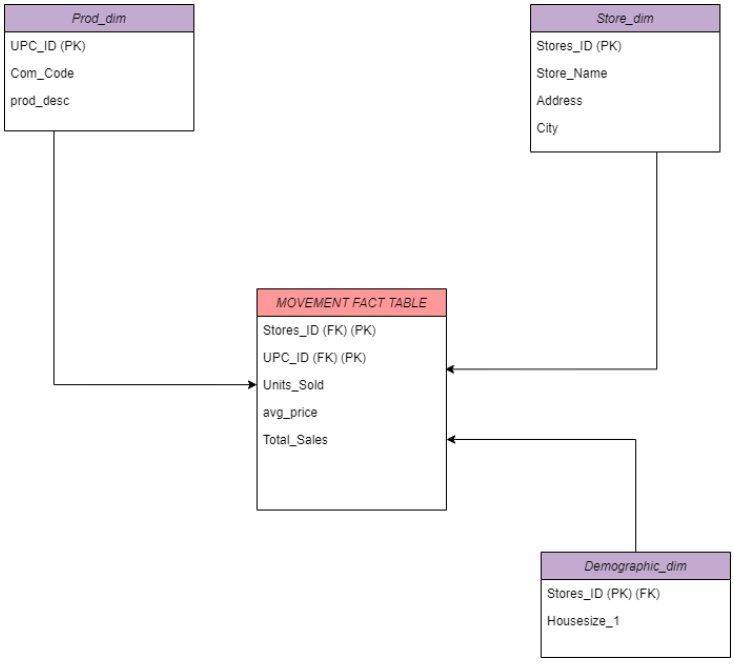
Goal: Find stores located in areas with the highest median income.
Fact Table Attributes: Stores_ID
Explanation:
The Store Sales Fact Table Stores_ID which is a unique store identifier. By joining it with the Demographic_dim (which includes Median_Income for each store location), the schema allows identification of specific stores located in areas with the highest median income. This can help businesses understand which stores cater to wealthier customers and can guide decisions about product pricing.

Business Question 4.9

Goal: Understand the impact of family size on sales of family-oriented products in specific stores
Fact TableAttributes: Stores_ID, Bulk_Total_Sales
Explanation:

Bulk sales can be used to determine the sale of family-oriented products because families tend to buy in bulk compared to individuals or smaller households. Stores (identified by Store_ID) located in areas with larger households (as identified in demographic dimensions - Housesize_3, Housesize_4, and Housesize_5) are more likely to have higher bulk sales. These bulk purchases often include family-oriented products such as larger packages of food, household supplies, or products designed for multiple family members.

- **Movement Specific**



The movement-specific data mart structure supports two of our business questions – **4.6** and **4.8**. Here's how:

Business Question 4.6:

Goal: Determine the average price for all frozen dinners across all stores.
Fact Table Attributes: UPC_ID, price, Stores_ID
Explanation:
The fact table has three important attributes namely: UPC_ID(which stands for unique product codes- unique product codes for all frozen dinners); price, the price of the specific frozen dinner; and Stores_ID, or store identifiers.
With the fact table we can calculate the average prices of all frozen dinners (denoted by UPC ID) across different stores. This will help us analyze the pricing across different stores and optimize pricing strategy based on store-specific locations.

Business Question 4.8:

Goal: Analyze the low sales of soft drinks in areas with a high single-person household.
Fact Table Attributes: Units_Sold, Price, Total_Sales, Stores_ID
Explanation:
The Movement Fact Table enables analysis of Total Sales for specific soft drinks by including key fields such as UPC_ID (which represents the unique product code for soft drinks), Stores_ID (store identifier), Units_Sold, Price, and Total_Sales.
Total Sales is calculated as Units Sold * Price, offering a clear view of product revenue.
When this fact table is joined with the Demographic Dimension—which includes Stores_ID and % of single-person households in each store's location—it becomes possible to assess which locations experience low soft drink sales despite having a high percentage of single-person households. This insight can help businesses refine their targeted marketing strategies in those areas.

- **Product category specific**

The product category specific schema caters to Business Question **4.10** in the following way:

Goal: Identify the top 7 product categories with the highest sales in 1995.
Explanation:
The top category-level fact of Sales consists of the aggregated sales sum of each Product category and time_id. It can be connected to the Time Dimension table through time_id (unique identifier), since it contains date-related information - year, month, and date for the analysis of sales over time. Filtering for the year 1995, the schema provides the ability to determine the top 7 product categories with peak sales records within that particular year. This analysis is used to evaluate the best product categories.

Mapping Table 1 – Source Files to Staging Tables

Source File Name	Source File Attribute	Mapping	Staging Table Type	Staging Table Name	Staging Table Attribute
Ccount.csv	Store Dairy, Frozen, Meat, Produce, Deli, GM, Grocery, Bakery Beer Bottle Bulk Camera Cheese ConvFood Cosmetic Fish Jewelry HABA Video Wine Spirits FTGCHIN FTGITAL Floral, SaladBar, MeatFroze n	Copy	Relation	stg_CCOUNT	store_id, Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum, Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum
Ccount.csv	Date	Transformatio n	Relation	stg_time	Year, Month, Day
Demo.csv	Name, City, Zip, Store, Income, hh3plus,	Copy	Relation	stg_demo	Store_name, Store_City, Store_Zip, Store_ID Median_income, Housesize_3,

	hh4plus, hhlarge, hhsingle				Housesize_4, Housesize_5, Housesize_1
UPCFRD.csv	Upc, Com_code, desc	Copy	Relation	stg_upcfrd	Upc_id_frd, Com_code_frd, prod_desc_frd
wfrd.csv	Mov, Price, Store, Upc,	Copy	Relation	stg_wfrd	Units_sold, Price, store_id_frd, upc_id_frd
wfrd.csv	Price, Move	Transformation	Relation	stg_wfrd	total_sales
upcsdr.csv	Upc, Com_code, Descrip	Copy	Relation	stg_upcsdr	Upc_id_sdr, Com_code_sdr, prod_desc_sdr
wsdr.csv	Mov, Price, Store, Upc	Copy	Relation	stg_wsdr	Units_sold_sdr, Price_sdr, store_id_sdr, upc_id_sdr
wsdr.csv	Price, Move	Transformation	Relation	stg_wsdr	total_sales

Mapping Table 2 – Staging Tables to Data Mart Tables

Staging Table Name	**Staging Table Attribute**	**Mapping**	**Data Mart Table Type**	**Data Mart Table Name**	**Data Mart Table Attribute**
stg_CCOUNT	store_id, store_name, store_City, store_Zip,	Copy	Dimension	store_dim	store_id, store_name, city, zip
stg_CCOUNT	store_id	Copy	Fact	store_sales	store_id
stg_CCOUNT	sales_bulk	Copy	Fact	store_sales	bulk_total_sales
stg_time	Year	Transformation (Filtering)	Dimension	time_dim	Year

stg_time	Month, Day, Time_ID	Copy	Dimension	time_dim	Month, Day, Time_ID
stg_CCOUNT	Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum, Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum	Copy	Fact	fact_ category_sales	Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum, Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum
stg_demo	Median_income, Housesize_3, Housesize_4, Housesize_5, Store_ID	Copy	Dimension	demo_dim (Datamart 2)	Median_income, Housesize_3, Housesize_4, Housesize_5, Store_ID
stg_demo	Housesize_1, Store_ID	Copy	Dimension	demo_dim (Datamart 3)	Housesize_1, Store_ID
stg_demo	store_name, store_id, store_city, store_zip	Copy	Dimension	store_dim	store_name, store_id, city, zip
stg_movement	Upc_id,	Copy	Dimension	prod_dim	Upc_id,

	Units_sold, Price, Total_sales, Com_code, Prod_desc, Category, Stored_ID				Units_sold, Price, Total_sales, Com_code, Prod_desc, Category, Stored_ID
stg_movement	Units_sold, Stored_ID, Upc_id, Price, Total_sales,	Copy	Fact	fact_movement	Units_sold store_id, upc_id, avg_price, total_sales

Physical Design

To develop the physical design plan for Dominick's Finer Foods (DFF), we will focus on optimizing performance, accommodating data mart growth through scalability, and manage storage using Microsoft SQL Server. Our data aggregate plan will primarily include creating summary tables like the 'top_category_sales_fact_table' which contains the aggregate data across various product categories. This approach is advantageous in reducing the need for real-time calculations and enhances query performance. We will implement indexing mostly through B-Tree indexes on high-selectivity columns such as 'Stores_ID' in the 'Store_Sales_Fact_Table' for faster results.

Section 4: Data Cleaning and Integration

Data Quality issues in DFF dataset

1. Missing Values in Ccount: The Store column in CCount contains blank entries for some records. It hinders accurate analysis because of incompleteness in data.

2. Null Values in Ccount: The CCount file has null entries in critical columns such as Store and Date, leading to gaps in the dataset that could result in skewed analyses.

3. Invalid Data in Sales Column in Ccount: The Sales column in the CCount includes non-numeric entries such as 'ddd'. As this column is expected to contain numeric dollar values, these entries need to be corrected to ensure proper calculations.

4. Date Format Issues: Dates in the CCount file are stored as non-standard strings, such as "b'880101", rather than a proper date datatype. This inconsistency needs to be removed by performing certain transformations.

5. Unreliable Store Mapping: Store mapping in the dataset was performed using store names instead of store IDs, which is a less reliable approach.

6. Invalid Zip Codes in DEMO.csv: The DEMO file includes invalid zip codes like '00000'. Valid zip codes should consist of either 5 or 9 digits.

7. Extra Spaces in Data: Many columns across the dataset contain leading or trailing spaces. These inconsistencies can cause issues with string matching and comparisons, and trimming these spaces is necessary for clean data processing.

In summary, the dataset has from a range of quality issues, including missing values, invalid entries, inconsistent formats, and unreliable mappings. Addressing these problems is essential to ensure the dataset's readiness for meaningful and accurate analysis.

ETL Plan

Step 1: Target Tables

1. **Demographic_dim Table**

Target Table	Target Column	Target Data Type
[601_grp2_DATAMART2].[dbo].[DEMO_DIM], [601_grp2_DATAMART3].[dbo].[demo_dim]	Store_ID	PK (Primary Key) integer
[601_grp2_DATAMART2].[dbo].[DEMO_DIM]	Housesize_3	integer
[601_grp2_DATAMART2].[dbo].[DEMO_DIM]	Housesize_4	integer
[601_grp2_DATAMART2].[dbo].[DEMO_DIM]	Housesize_5	integer
[601_grp2_DATAMART2].[dbo].[DEMO_DIM]	Median_Income	Decimal
[601_grp2_DATAMART3].[dbo].[demo_dim]	Housesize_1	integer

2. **Store_dim Table**

Target Table	Target Column	Target Data Type
[601_grp2_DATAMART2].[dbo].[store_dim], [601_grp2_DATAMART3].[dbo].store_dim]	store_ID	PK (Primary Key) integer
[601_grp2_DATAMART2].[dbo].[store_dim], [601_grp2_DATAMART3].[dbo].store_dim]	store_name	Varchar(50)
[601_grp2_DATAMART2].[dbo].[store_dim], [601_grp2_DATAMART3].[dbo].store_dim]	Zip	integer

[601_grp2_DATAMART2].[dbo].[store_dim], [601_grp2_DATAMART3].[dbo].store_dim]	City	Varchar(50)

3. **Prod_dim Table**

Target Table	Target Column	Target Data Type
[601_grp2_DATAMART3].[dbo].[product_dim]	UPC_ID	PK (Primary Key)
[601_grp2_DATAMART3].[dbo].[product_dim]	COM_CODE	Varchar(50)
[601_grp2_DATAMART3].[dbo].[product_dim]	PROD_DESC	Varchar(50)

4. **Time_dim Table**

Target Table	Target Column	Target Data Type
[601_grp2_DATAMART1].[dbo].[time_dim]	Time_ID	PK (Primary Key)
[601_grp2_DATAMART1].[dbo].[time_dim]	Year	Integer
[601_grp2_DATAMART1].[dbo].[time_dim]	Month	Integer
[601_grp2_DATAMART1].[dbo].[time_dim]	Date	Date

5. **Store_Sales_FACT Table**

Target Table	Target Column	Target Data Type
[601_grp2_DATAMART2].[dbo].[Store_Sales]	Stores_ID	FK (Foreign Key)
[601_grp2_DATAMART2].[dbo].[Store_Sales]	Bulk_Total_Sales	Decimal

6. **Movement_FACT Table**

Target Table	Target Column	Target Data Type
[601_grp2_DATAMART3].[dbo].[Movement_FACT]	Stores_ID	FK (Foreign Key)
[601_grp2_DATAMART3].[dbo].[Movement_FACT]	UPC_ID	FK (Foreign Key)
[601_grp2_DATAMART3].[dbo].[Movement_FACT]	Units_Sold	Integer
[601_grp2_DATAMART3].[dbo].[Movement_FACT]	Avg_Price	Decimal
[601_grp2_DATAMART3].[dbo].[Movement_FACT]	Total_Sales	Decimal

7. **FACT_CATEGORY_SALES Table**

Target Table	Target Column	Target Data Type
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Time_ID	FK (Foreign Key)
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Dairy_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Frozen_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Meat_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Produce_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Deli_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	GM_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Grocery_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Bakery_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Beer_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Bottle_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Bulk_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	ConvFood_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Cosmetic_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Fish_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Floral_Sum	Decimal

[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	FTGChinese_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	FTGItalian_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	HABA_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Jewelry_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	MeatFroz_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Pharmacy_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Promo_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Salad_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Spirits_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	SSDeli_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Video_Sum	Decimal
[601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]	Wine_Sum	Decimal

Step 2: Data Source Identification

Data Source	**Data Source File**
Customer Count	Ccount.csv
Time	Ccount.csv
Demography	Demo.csv
Product Information	UPCFRD.csv, upcsdr.csv
Product Movement	wfrd.csv, wsdr.csv

Fig: Data Sources

We started our data analysis by using multiple key tables, including the Customer Count table, the Demographicstable, and additional data from files such as Ccount.csv, Demo.csv, wfrd.csv, and

wsdr.csv. To answer our business questions, we extracted, cleaned, and transformed data from these sources, ensuring it was tailored to meet analytical needs.

For example, one of our questions involved identifying stores located in areas with the highest median income. We identified the relevant information within the Demographics (Demo.csv) and Store data. The required fields, such as Median_income, Store_ID, and location attributes, were extracted, transformed, and mapped to appropriate staging tables to suit our analytical framework. This allowed us to pinpoint stores situated in affluent areas effectively.

Another key analysis focused on determining the average price of frozen dinners for specific UPCs across all stores. We extracted data from the Movement File (wfrd.csv) and UPC data (UPCFRD.csv). By isolating frozen dinner-related UPCs and their sales and price data, we computed the average price across stores. The processed data was then summarized and mapped to fact tables for further analysis.

Another question was to explore stores with the lowest sales of soft drinks and the highest percentage of single-person households. This analysis involved integrating sales data from the Customer Count file (Ccount.csv) and demographic attributes from the Demographics file (Demo.csv). By combining these datasets, we identified stores that exhibited both low soft drink sales and high single-person household demographics.
The data warehouse's target data was acquired by leveraging various combinations of data sources to generate both the staging area tables and the data mart tables in the following manner:

Data Source	**Source File Name**	**Staging Area Table Name**	**Data Mart Table Name**
Customer Count	Ccount.csv	stg_CCOUNT	store_sales, fact_category_sales
Time	Ccount.csv, wfrd.csv, wsdr.csv	stg_time	time_dim
Demographics	Demo.csv	stg_demo	demo_dim, store_dim
Product Information	UPCFRD.csv, upcsdr.csv	stg_upcfrd, stg_upcsdr	prod_dim
Sales Movement	wfrd.csv, wsdr.csv	stg_wfrd, stg_wsdr	fact_movement, prod_dim

Step 3: Data Mapping

Mapping Table 1 – Source Files to Staging Tables

Source File Name	**Source File Attribute**	**Mapping**	**Staging Table Type**	**Staging Table Name**	**Staging Table Attribute**
Ccount.csv	Store	Copy	Relation	stg_CCOUNT	store_id,

	Dairy, Frozen, Meat, Produce, Deli, GM, Grocery, Bakery Beer Bottle Bulk Camera Cheese ConvFood Cosmetic Fish Jewelry HABA Video Wine Spirits FTGCHIN FTGITAL Floral, SaladBar, MeatFroze n				Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum, Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum
Ccount.csv	Date	Transformatio n	Relation	stg_time	Year, Month, Day
Demo.csv	Name, City, Zip, Store, Income, hh3plus, hh4plus, hhlarge, hhsingle	Copy	Relation	stg_demo	Store_name, Store_City, Store_Zip, Store_ID Median_income, Housesize_3, Housesize_4, Housesize_5, Housesize_1
UPCFRD.cs v	Upc, Com_code, desc	Copy	Relation	stg_upcfrd	Upc_id_frd, Com_code_frd, prod_desc_frd
wfrd.csv	Mov, Price,	Copy	Relation	stg_wfrd	Units_sold, Price,

	Store, Upc,				store_id_frd, upc_id_frd
wfrd.csv	Price, Move	Transformatio n	Relation	stg_wfrd	total_sales
upcsdr.cs v	Upc, Com_code, Descrip	Copy	Relation	stg_upcsdr	Upc_id_sdr, Com_code_sdr, prod_desc_sdr
wsdr.csv	Mov, Price, Store, Upc	Copy	Relation	stg_wsdr	Units_sold_sdr, Price_sdr, store_id_sdr, upc_id_sdr
wsdr.csv	Price, Move	Transformatio n	Relation	stg_wsdr	total_sales

Mapping Table 2 – Staging Tables to Data Mart Tables

Staging Table Name	Staging Table Attribute	Mapping	Data Mart Table Type	Data Mart Table Name	Data Mart Table Attribute
stg_CCOUNT	store_id, store_name, store_City, store_Zip,	Copy	Dimension	store_dim	store_id, store_name, city, zip
stg_CCOUNT	store_id	Copy	Fact	store_sales	store_id
stg_CCOUNT	sales_bulk	Copy	Fact	store_sales	bulk_total_sales
stg_time	Year	Transformation (Filtering)	Dimension	time_dim	Year
stg_time	Month, Day, Time_ID	Copy	Dimension	time_dim	Month, Day, Time_ID
stg_CCOUNT	Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum,	Copy	Fact	fact_ category_sales	Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum,

	Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum				Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum
stg_demo	Median_income, Housesize_3, Housesize_4, Housesize_5, Store_ID	Copy	Dimension	demo_dim (Datamart 2)	Median_income, Housesize_3, Housesize_4, Housesize_5, Store_ID
stg_demo	Housesize_1, Store_ID	Copy	Dimension	demo_dim (Datamart 3)	Housesize_1, Store_ID
stg_demo	store_name, store_id, store_city, store_zip	Copy	Dimension	store_dim	store_name, store_id, city, zip
stg_movement	Upc_id, Units_sold, Price, Total_sales, Com_code, Prod_desc, Category, Stored_ID	Copy	Dimension	prod_dim	Upc_id, Units_sold, Price, Total_sales, Com_code, Prod_desc, Category, Stored_ID
stg_movement	Units_sold, Stored_ID,	Copy	Fact	fact_movement	Units_sold store_id,

	Upc_id, Price, Total_sales,				upc_id, avg_price, total_sales

Step 4: Data extraction rules

We have collected all the relevant CSV Files for our business questions and loaded the data as it is from the CSV Files into the Staging database (601_grp2_stg_db). These were the steps we have followed as part of our extraction plan:

1. Importing Ccount File:

1. Utilized the SSIS Import and Export Wizard to map values from the source file to the destination database.
2. In the Source, loaded the ccount.csv file as a flat file source.
3. Extracted all the fields from the ccount source file
4. Renamed all the product columns by adding "SUM" at the end to denote the total sales for the following products- Eg: GROCERY_SUM, DAIRY_SUM etc.
5. Renamed Store to Store_ID to denote unique store identifier.
6. Validated mapping for each column
7. In the destination, loaded all data in the stg_CCOUNT table in the staging database.

2. Importing Demo File:

1. Utilized the SSIS Import and Export Wizard to map values from the source file to the destination database.
2. In the Source, loaded the demo.csv file as a flat file source.
3. Extracted all the fields from the demo source file.
4. Renamed the following relevant column names while creating the table and mapping columns:
   HH3PLUS: Housesize_3
   HH4PLUS: Housesize_4
   HHLARGE: Housesize_5
   HHSINGLE: Housesize_1
   INCOME: Median_Income
   NAME: STORE_NAME
   STORE: STORE_ID
   CITY: STORE_CITY
   ZIP: STORE_ZIP
5. Validated mapping for each column
6. In the destination, loaded all data in the stg_demo table in the staging database.

3. Importing Movement File:

1. Utilized the SSIS Import and Export Wizard to map values from the source file to the destination database.

2. We required two files from the Movement Files- wfrd.csv and wsdr.csv.
3. In the Source, we both these csv files as a flat file source.
4. Extracted all the fields from these source files.
5. Added “FRD” at the end of upc_id and store_id in the wfrd to denote unique identifiers for frozen dinner
6. Added “SDR” at the end of upc_id and store_id in the wsdr to denote unique identifiers for soft drinks
7. Renamed “Move” Column to “Units_Sold”
8. Validated mapping for all the columns
9. In the destination, loaded all wfrd.csv and wsdr.csv data in the stg_wfrd and stg_wsdr table respectively in the staging database.

4. Import UPC Files:

1. Utilized the SSIS Import and Export Wizard to map values from the source file to the destination database.
2. We required two files from the UPC Files- upcfrd.csv and upcsdr.csv.
3. In the Source, we both these csv files as a flat file source.
4. Extracted all the fields from these source files.
5. Added “FRD” at the end of upc_id, com_code and prod_desc in the upcfrd to denote unique identifiers for frozen dinner
6. Added “SDR” at the end of upc_id, com_code and prod_desc in the upcsdr to denote unique identifiers for soft drinks
7. Validated mapping for all the columns
8. In the destination, loaded all upcfrd.csv and upcsdr.csv data in the stg_upcfrd and stg_upcsdr table respectively in the staging database.

We carefully devised a data extraction plan, ensuring a delicate balance between flexibility and data integrity.

Step 5: Data Transformation and Cleansing Rules:

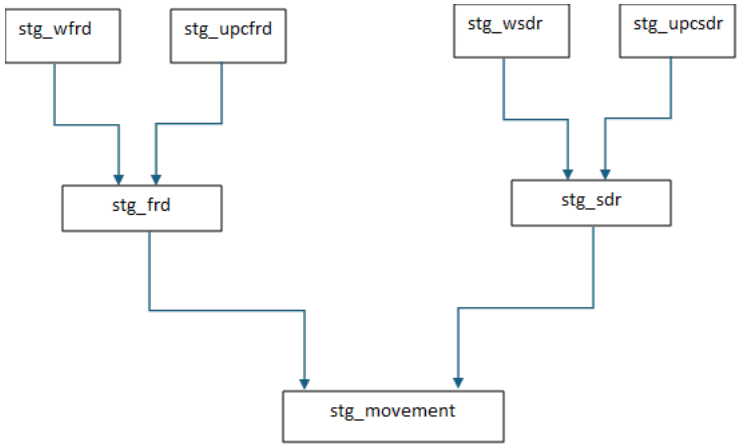
We have transformed all the staging tables mentioned above in the Staging database (601_grp2_stg_db) itself. These were the steps we have followed as part of our transformation and cleaning plan:

1. We have removed all the existing columns that we did not require as part of our business questions.
2. Removed all the missing and null values in the data.
3. Added the following New Columns:

Column Name	Table	Description
Year	Stg_CCOUNT	Extracted Year from DATE Column

Month	Stg_CCOUNT	Extracted Month from DATE Column
Day	Stg_CCOUNT	Extracted Day from DATE Column
Category	Stg_UPCFRD	To explain Category of the products. Set to Frozen Dinner
Category	Stg_UPCSDR	To explain Category of the products. Set to Soft Drinks
Total_Sales	Stg_WFRD	Total sales for each frozen dinner product Total Sales = units_sold * price
Total_Sales	Stg_WSDR	Total sales for each soft drink product Total Sales = units_sold * price

4. We also wanted to join stg_wfrd, stg_wsdr, stg_upcfrd and stg_upcsdr into a single table in the staging database that would help us define a product's category, upc, units_sold, price, total sales and the store it is sold in. To do this we created temporary tables that could help us join movement and upc files on upc_id and finally join both the temporary files into stg_movement. Following is the diagram to explain this better:



The stg_movement would be the final table that will help us to create product dimension.

Overall, the data transformation phase focuses on ensuring data integrity and consistency by cleansing and organizing it into the appropriate structure for analysis and reporting. When executed effectively, this phase enhances the quality of insights derived in subsequent business processes.

Step 6: Plan for Aggregate Table

To address our business question focused on identifying the highest-selling product categories, we plan to perform aggregation of product sales across key temporal dimensions, including Year, Month, and Day. By organizing and summarizing sales data over a one-year period, we plan to prepare data for further analysis and reporting

For other business questions involving movement data and store sales, we have intentionally retained the lowest level of granularity in our dataset. This approach ensures that we have detailed, transaction-level data available to meet the specific requirements of our business questions. Maintaining this level of granularity allows for flexible analysis, enabling us to drill down into specific records to understand sales dynamics, consumer behavior, and other key metrics at the most detailed level possible.

Looking ahead, for advanced analysis and reporting, we plan to implement targeted aggregations based on product category, store characteristics, and location attributes.

Step 7: Organization of data staging area

The data staging area plays a crucial role in the ETL process, acting as a vital bridge between raw source data and the final data warehouse. The organization of the data staging area is essential for efficiently managing and processing source data. In this project, the staging area serves as the foundation for cleansing, transforming, and enriching the raw data.
By carefully structuring this step, we ensure that the data is properly prepared and optimized before it is loaded into the target data warehouse, setting the stage for accurate and meaningful analysis and reporting.

Our staging database is called 601_grp2_stg_db and it contains all the cleaned and transformed source data. Below is the mapping of source csv files to staging and the transformation/ cleansing applied to each:

Source	Staging Table	Transformation Applied
Ccount.csv	Stg_CCOUNT	Columns not required dropped, missing values removed, new additional columns added
Demo.csv	Stg_demo	Columns not required dropped, missing values removed
Stg_CCOUNT	Stg_time	Relevant Columns (Year, Month and Day) stored
Wfrd.csv	Stg_wfrd	Columns not required dropped, missing values removed, new additional columns added

upcfrd.csv	Stg_upcfrd	Columns not required dropped, missing values removed
Wsdr.csv	Stg_wsdr	Columns not required dropped, missing values removed, new additional columns added
upcsdr.csv	Stg_upcsdr	Columns not required dropped, missing values removed
Stg_upcfrd Stg_wfrd	Stg_frd	Join tables
Stg_upcsdr Stg_wsdr	Stg_sdr	Join tables
Stg_frd Stg_sdr	Stg_movement	Join tables

Step 8: Procedures for all data extractions and loadings:

All the data extraction and loading are executed in SQL Server Integration Services (SSIS).

In data extraction, we made different packages to extract data from flat file sources to the OLE DB destination in the staging database. We gave special attention to ensure all the sources and destinations were properly defined and all the column mapping was validated.

In data loading, each SSIS Package contained transformed data flow from the staging database to the independent data marts. We ensured that all the independent data marts had the necessary dimension and fact tables to solve our business questions.

Step 9: ETL for Dimension Table:

In this step, we mainly focus on the ETL process for Dimension tables. Efficient ETL processes for dimension tables are essential to ensure data accuracy and enable effective analysis of data from fact tables. The following table explains the Loading of transformed data from the staging database to the independent data marts:

Source Table in Staging Database	Dimension Table
Stg_demo	Demo_dim
Stg_demo	Store_dim
Stg_time	Time_dim

Stg_movement	Prod_dim

The process involved first creating dimension tables- **Demo_dim**, **Store_dim**, **Time_dim** and **Prod_dim** in the SQL Server Management Studio (SSMS).
Once these databases were created, we created new SSIS Packages for each dimension table to ensure simplicity and accuracy of data.
We then mapped the appropriate columns from the above staging tables to the newly created database columns in SSIS
Finally, all the data flow tasks were executed in SSIS and the data in all the dimension tables in SSMS was validated.

Step 10: ETL for Fact Table:

In this final step, we focused on the ETL process for Fact Tables. We have three independent data marts, and each data mart has a fact table. The following are the fact tables- **fact_top_category_sales**, **fact_movemen**t and **fact_store_sales**. Below table explains the ETL Process for each fact table:

Dimension Tables	Source Tables	Lookup Transformation Columns	Destination Tables
Demo_dim	Stg_Ccount	Store_id	Fact_store_sales
Store_dim		Store_id	Fact_store_sales
Prod_dim	Stg_movement	Upc_id	Fact_movement
Store_dim		Store_id	Fact_movement
Demo_dim		Store_id	Fact_movement
Time_dim	Stg_Ccount	Time_id	Fact_top_category_sales

The process began with creating all the fact tables within SQL Server Management Studio (SSMS).Next, individual SSIS packages were developed for the ETL of each fact table to ensure the use of accurate dimension tables during fact table creation. An OLE DB Source was first configured to define the source tables for each fact table. Subsequently, a Lookup Transformation was performed on the specified columns for each dimension table. Finally, the data was loaded into the destination tables designated as the fact tables.

Implementation of the ETL Plan

Data Extraction

We have imported all relevant csv files for our business questions into the staging database.
Database: 601_grp2_stg_db

Steps:

1. CCOUNT.csv -> stg_CCOUNT

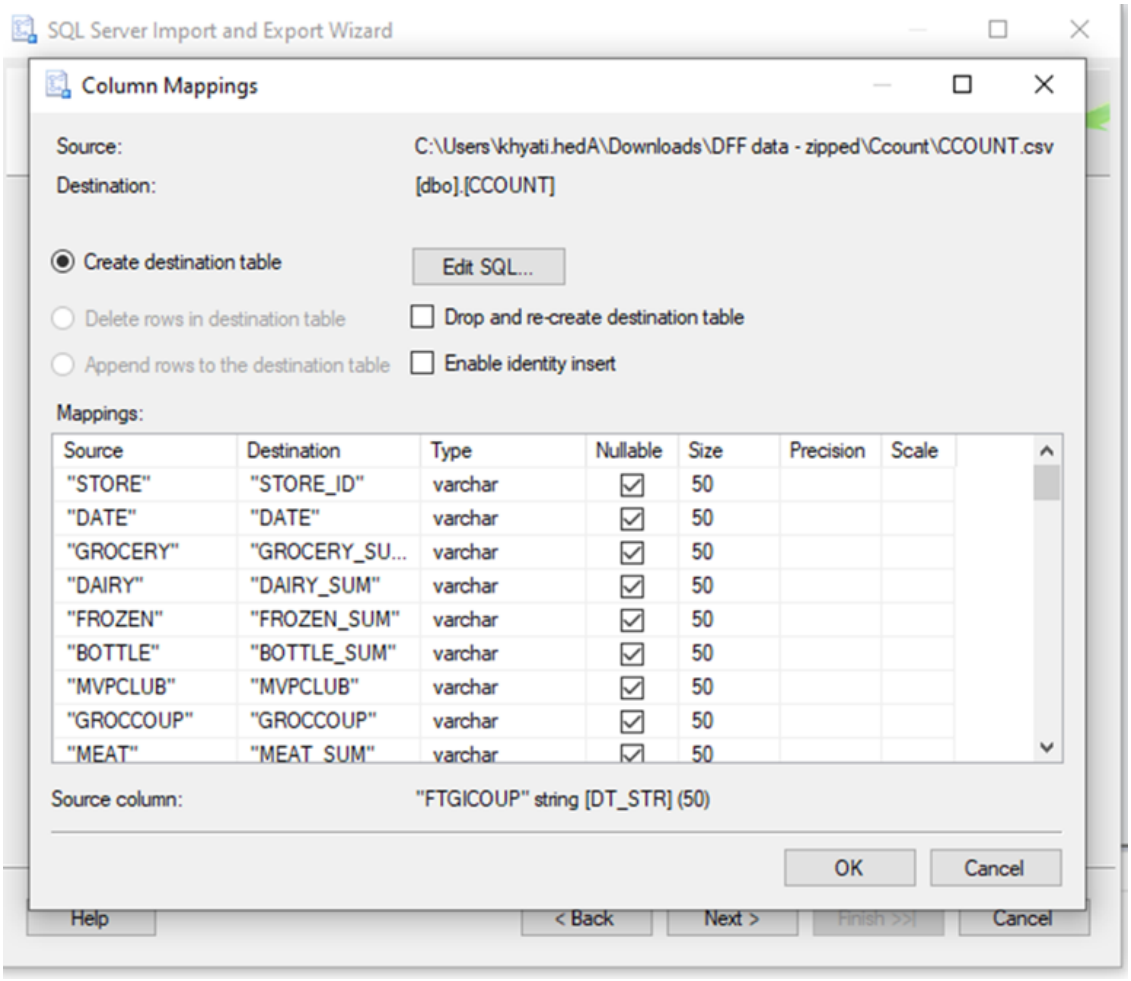


Fig: Validating the Mapping from source to destination and changing Column names in the destination columns

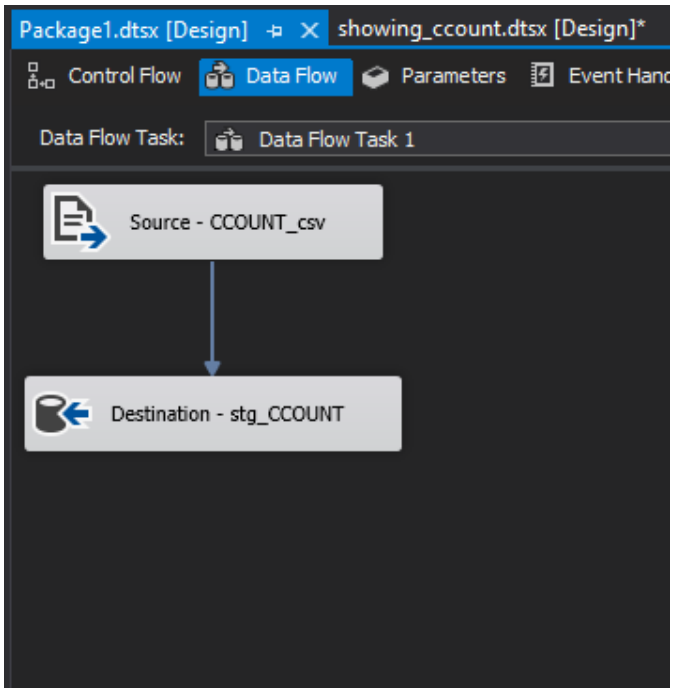


Fig: Data Flow Task in SSIS

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) ["STORE_ID"]
      ,["DATE"]
      ,["GROCERY_SUM"]
      ,["DAIRY_SUM"]
      ,["FROZEN_SUM"]
      ,["BOTTLE_SUM"]
      ,["MVPCLUB"]
      ,["GROCCOUP"]
      ,["MEAT_SUM"]
      ,["MEATFROZ_SUM"]
      ,["MEATCOUP"]
      ,["FISH_SUM"]
      ,["FISHCOUP"]
      ,["PROMO_SUM"]
      ,["PROMCOUP"]
      ,["PRODUCE_SUM"]
      ,["BULK_SUM"]
      ,["SALADBAR_SUM"]
      ,["PRODCOUP"]
      ,["BULKCOUP"]
      ,["SALCOUP"]
      ,["FLORAL_SUM"]
      ,["FLORCOUP"]
      ,["DELI_SUM"]
      ,["DELISELF"]
```

100 %

Results | Messages

	"STORE_ID"	"DATE"	"GROCERY_SUM"	"DAIRY_SUM"	"FROZEN_SUM"	"BOTTLE_SUM"	"MVPCLUB"	"GROCCOUP"	"MEAT_SUM"	"MEATFROZ_SUM"	"MEATCOUP"	"FISH_SUM"	"FISHCOUP"	"PROMO_SUM"	"PROMC
1	308	"910724"	29265.42	4198.1	4094.82	0	0	-12.03	5845.95	923.53	0	541.52	0	0	0
2	308	"910725"	39500.43	5863.06	8848.43	0	0	-8.61	8804.54	1383.6	0	1107.81	0	0	0
3	308	"910726"	43789.7	6807.61	8335.76	0	0	-12.03	10862.65	1402.85	0	1256.25	0	0	0
4	308	"910727"	48870.43	8150.73	8500.96	0	0	-28.07	13316.15	1602.51	0	1435.16	0	0	0
5	308	"910728"	38760	6487.5	6912.68	0	0	-0.2	9031.35	1425.63	0	1267.81	0	0	0
6	308	"910729"	28137.5	4486.44	5208.78	0	0	-0.8	5838.87	1047.41	0	571.43	0	0	0
7	308	"910730"	27626.39	4246.11	4699.16	0	0	0	4995.5	805.94	0	663.83	0	0	0
8	308	"910731"	27210.85	4502.61	4667.65	0	0	0	5384.71	857.62	0	622.37	0	0	0
9	308	"910801"	54566.96	6673.18	6040.31	0	0	0	11351.02	1253.63	0	864.28	0	0	0
10	308	"910802"	53770.67	7015.09	6378.94	0	0	0	13653.54	1438.24	0	1199.81	0	0	0
11	308	"910803"	65082.96	9070.08	7735.73	0	0	0	17888.12	1913.44	0	1090.55	0	0	0
12	308	"910804"	46294.37	6525.79	6039.81	0	0	-11.29	9573.17	1341.82	0	967.77	0	0	0
13	308	"910805"	34485.66	4966.38	4140.21	0	0	0	7336.1	894.85	0	472.41	0	0	0
14	308	"910806"	30779.27	4368.78	4059.22	0	0	-1	7056.6	1043.22	0	457.45	0	0	0
15	308	"910807"	29691.39	4096.54	3803.69	0	0	0	6651.57	888.28	0	558.95	0	0	0
16	308	"910808"	37167.78	5518.55	4727.6	0	0	-1.78	9587.47	1312.83	0	959.34	0	0	0
17	308	"910809"	39893.9	5626.42	4575.59	0	0	0	10824.8	1672.35	0	1068.48	0	0	0
18	308	"910810"	48289.8	7247.9	6468.94	0	0	0	14873.34	2768.96	0	1355.46	0	0	0

Fig: stg_CCOUNT in the staging database

2. DEMO.csv -> stg_DEMO

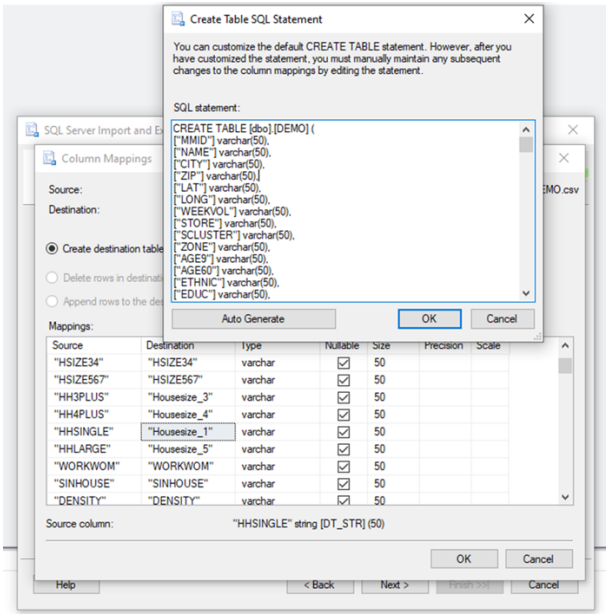


Fig: Create Table query for the destination table and validating the mapping from source to destination

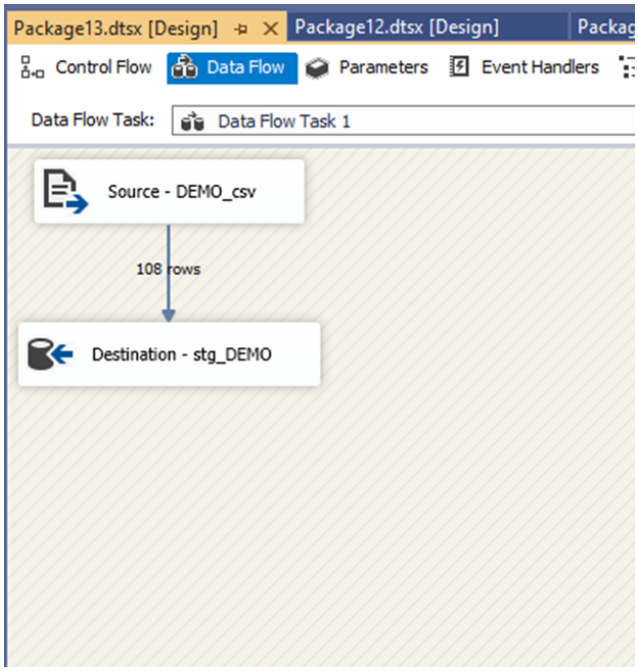


Fig: Data Flow Task in SSIS successfully executed

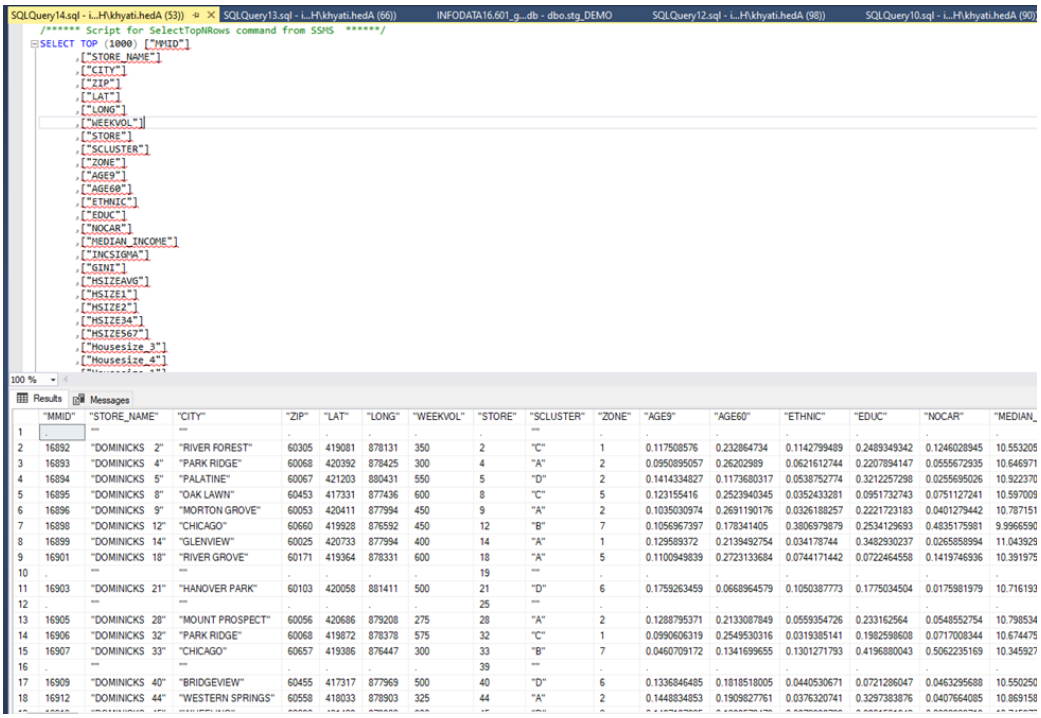


Fig: stg_DEMO in staging database.

3. wfrd.csv -> stg_wfrd

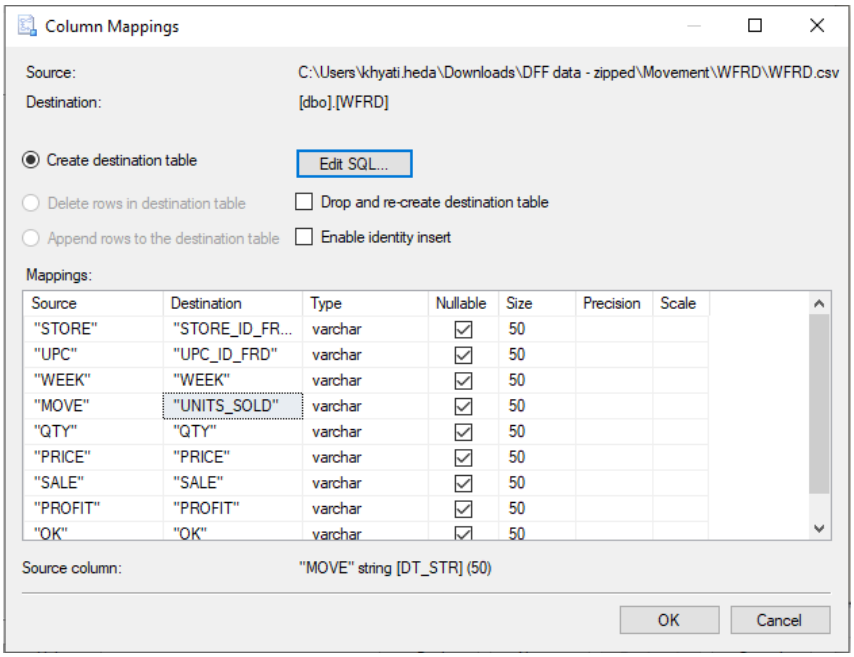


Fig: Validating the Mapping from source to destination and changing some Column names in the destination columns

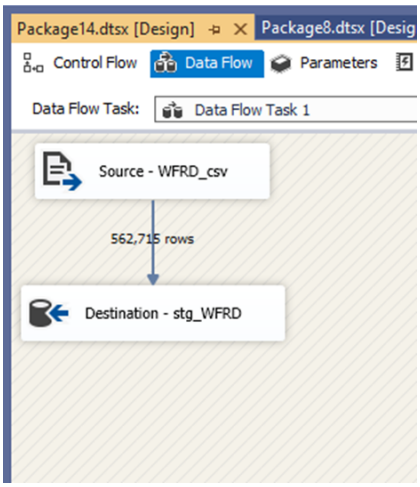
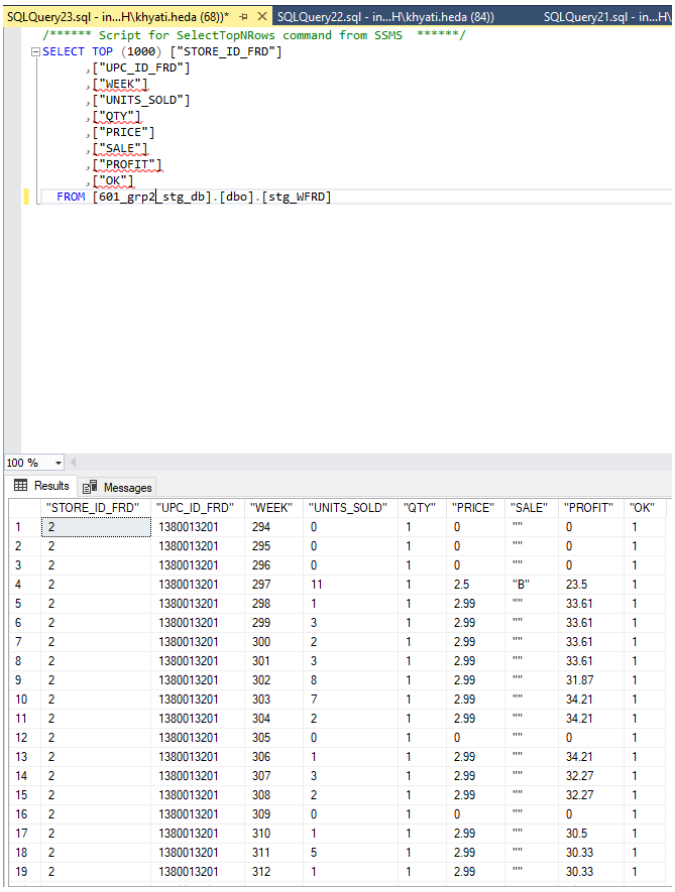


Fig: Data Flow Task in SSIS successfully executed



	"STORE_ID_FRD"	"UPC_ID_FRD"	"WEEK"	"UNITS_SOLD"	"QTY"	"PRICE"	"SALE"	"PROFIT"	"OK"
1	2	1380013201	294	0	1	0	""	0	1
2	2	1380013201	295	0	1	0	""	0	1
3	2	1380013201	296	0	1	0	""	0	1
4	2	1380013201	297	11	1	2.5	"B"	23.5	1
5	2	1380013201	298	1	1	2.99	""	33.61	1
6	2	1380013201	299	3	1	2.99	""	33.61	1
7	2	1380013201	300	2	1	2.99	""	33.61	1
8	2	1380013201	301	3	1	2.99	""	33.61	1
9	2	1380013201	302	8	1	2.99	""	31.87	1
10	2	1380013201	303	7	1	2.99	""	34.21	1
11	2	1380013201	304	2	1	2.99	""	34.21	1
12	2	1380013201	305	0	1	0	""	0	1
13	2	1380013201	306	1	1	2.99	""	34.21	1
14	2	1380013201	307	3	1	2.99	""	32.27	1
15	2	1380013201	308	2	1	2.99	""	32.27	1
16	2	1380013201	309	0	1	0	""	0	1
17	2	1380013201	310	1	1	2.99	""	30.5	1
18	2	1380013201	311	5	1	2.99	""	30.33	1
19	2	1380013201	312	1	1	2.99	""	30.33	1

Fig: stg_wfrd in staging database

4. upcfrd.csv -> stg_upcfrd

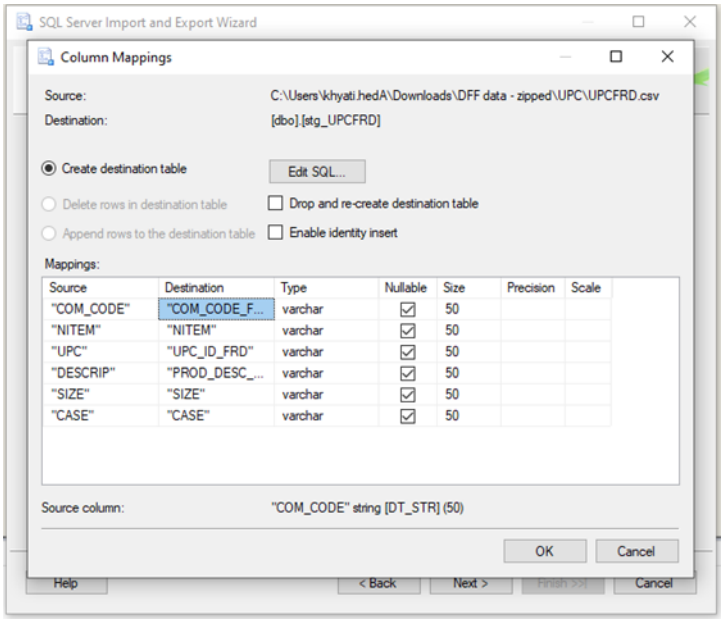


Fig: Validating the Mapping from source to destination and changing some Column names in the destination columns

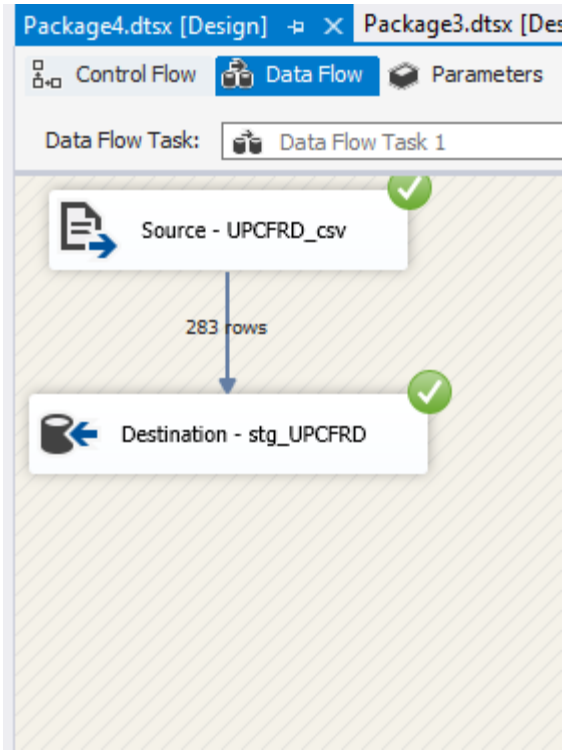
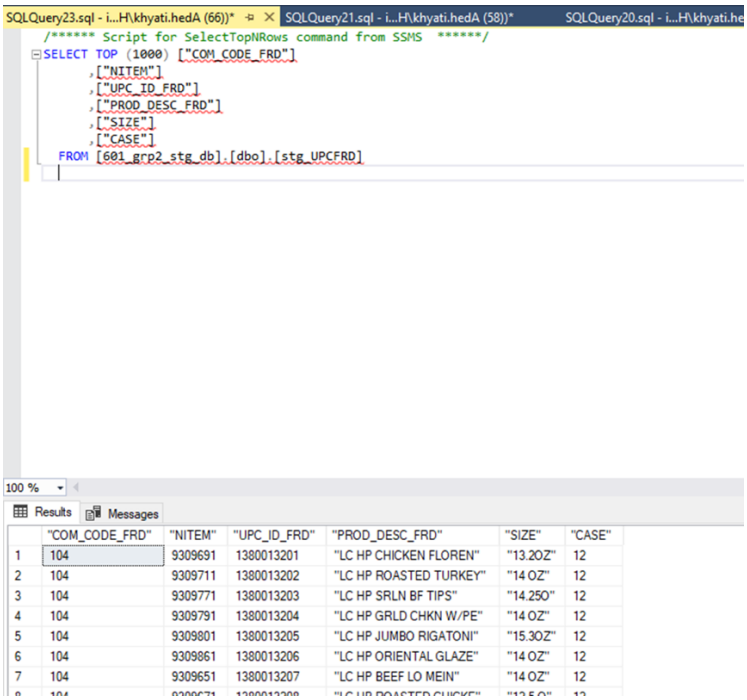


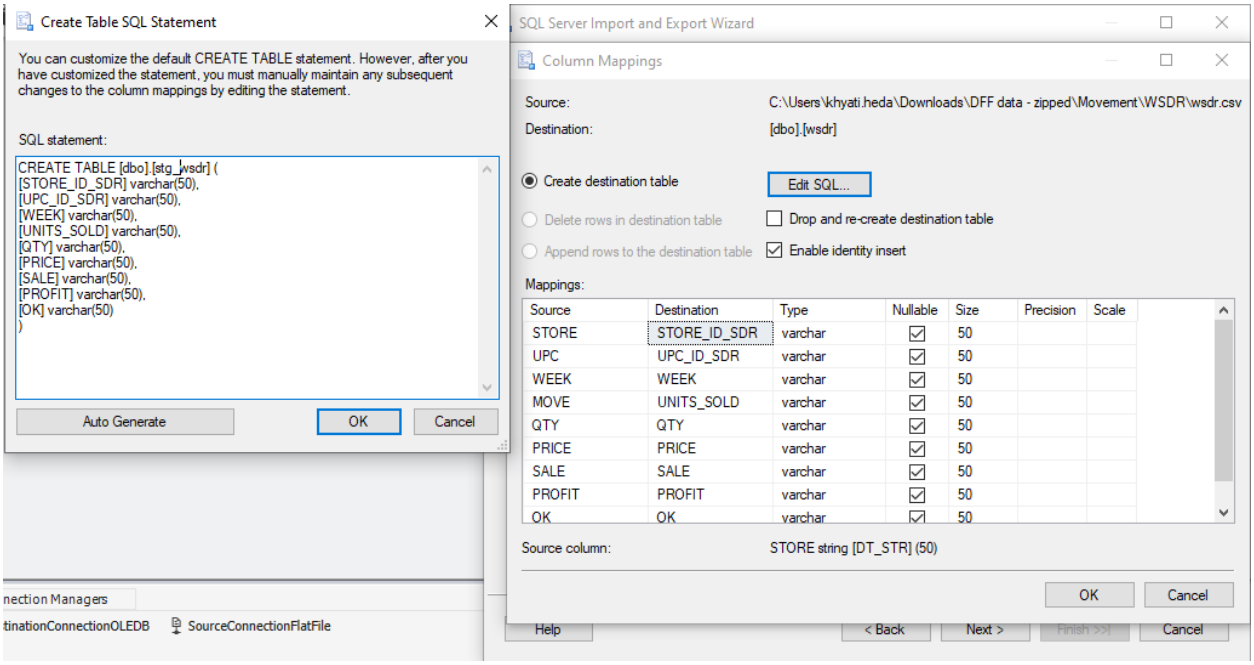
Fig: Data Flow Task in SSIS successfully executed



	"COM_CODE_FRD"	"NITEM"	"UPC_ID_FRD"	"PROD_DESC_FRD"	"SIZE"	"CASE"
1	104	9309691	1380013201	"LC HP CHICKEN FLOREN"	"13.2OZ"	12
2	104	9309711	1380013202	"LC HP ROASTED TURKEY"	"14 OZ"	12
3	104	9309771	1380013203	"LC HP SRLN BF TIPS"	"14.25O"	12
4	104	9309791	1380013204	"LC HP GRLD CHKN W/PE"	"14 OZ"	12
5	104	9309801	1380013205	"LC HP JUMBO RIGATONI"	"15.3OZ"	12
6	104	9309861	1380013206	"LC HP ORIENTAL GLAZE"	"14 OZ"	12
7	104	9309651	1380013207	"LC HP BEEF LO MEIN"	"14 OZ"	12

Fig: stg_upcfrd in staging database.

5. wsdr.csv -> stg_wsdr



Source	Destination	Type	Nullable	Size	Precision	Scale
STORE	STORE_ID_SDR	varchar	☑	50		
UPC	UPC_ID_SDR	varchar	☑	50		
WEEK	WEEK	varchar	☑	50		
MOVE	UNITS_SOLD	varchar	☑	50		
QTY	QTY	varchar	☑	50		
PRICE	PRICE	varchar	☑	50		
SALE	SALE	varchar	☑	50		
PROFIT	PROFIT	varchar	☑	50		
OK	OK	varchar	☑	50		

Fig: Create Table query for the destination table (stg_wsdr) and validating the mapping from source to destination

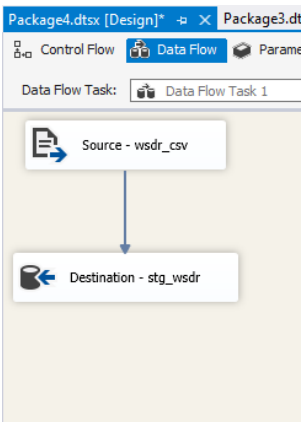


Fig: Data Flow Task in SSIS successfully executed

SQLQuery24.sql - in...\khyati.heda (132))* | SQLQuery23.sql - in...H\khyati.heda (68))* | SQLQuery22.sql - in...H\khyati.heda (8

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [STORE_ID_SDR]
      ,[UPC_ID_SDR]
      ,[WEEK]
      ,[UNITS_SOLD]
      ,[QTY]
      ,[PRICE]
      ,[SALE]
      ,[PROFIT]
      ,[OK]
  FROM [601_grp2_stg_db].[dbo].[stg_wsdr]
```

100 %

Results | Messages

	STORE_ID_SDR	UPC_ID_SDR	WEEK	UNITS_SOLD	QTY	PRICE	SALE	PROFIT	OK
1	32	5490000029	350	3	1	2.99		22.24	1
2	32	5490000029	351	6	1	2.99		22.24	1
3	32	5490000029	352	11	1	2.5	B	7	1
4	32	5490000029	353	9	1	2.99		22.24	1
5	32	5490000029	354	7	1	2.99		22.24	1
6	32	5490000029	355	9	1	2.99		22.24	1
7	32	5490000029	356	23	1	2.04	S	-13.8	1
8	32	5490000029	357	446	1	1.99	S	-16.83	1
9	32	5490000029	358	5	1	2.99		39.39	1
10	32	5490000029	359	6	1	2.99		39.39	1
11	32	5490000029	360	7	1	2.99		39.39	1
12	32	5490000029	361	5	1	2.99		39.39	1
13	32	5490000029	362	7	1	2.99		22.24	1
14	32	5490000029	363	10	1	2.99		22.24	1
15	32	5490000029	364	11	1	1.89	B	-22.42	1
16	32	5490000029	365	13	1	1.72	B	-35.11	1
17	32	5490000029	366	4	1	2.99		22.24	1
18	32	5490000029	367	3	1	2.99		22.24	1
19	32	5490000029	368	4	1	2.99		22.24	1

Fig: stg_wsdr in staging database.

6. Upcsdr.csv-> stg_upcsdr

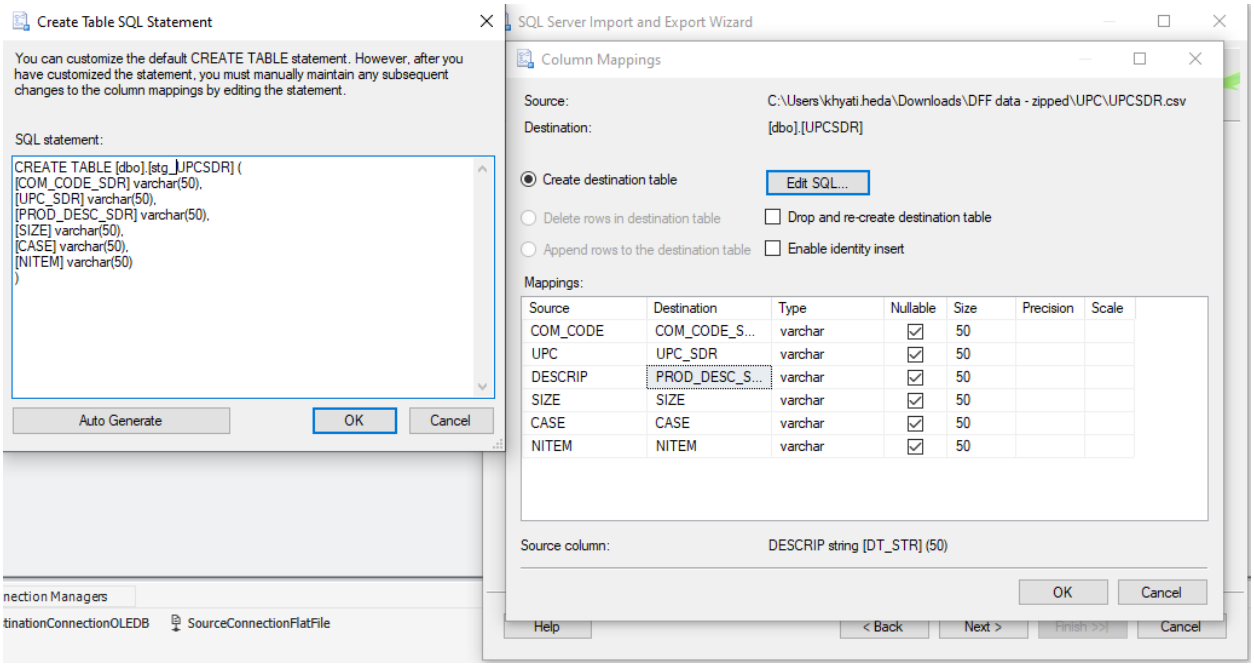


Fig: Create Table query for the destination table (stg_upcsdr) and validating the mapping from source to destination

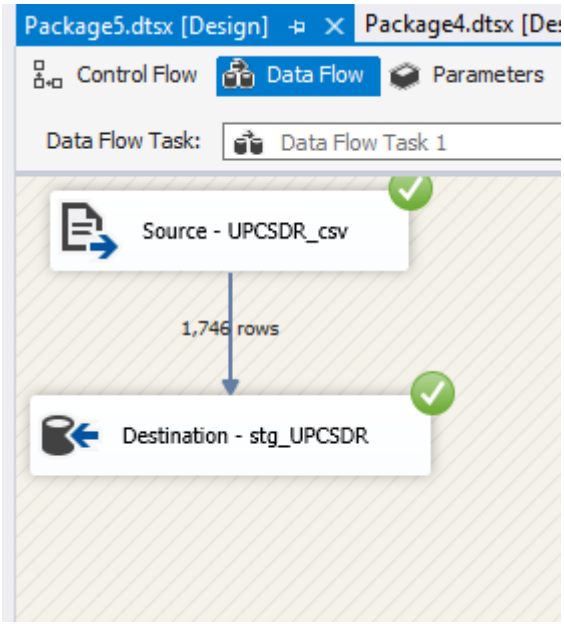
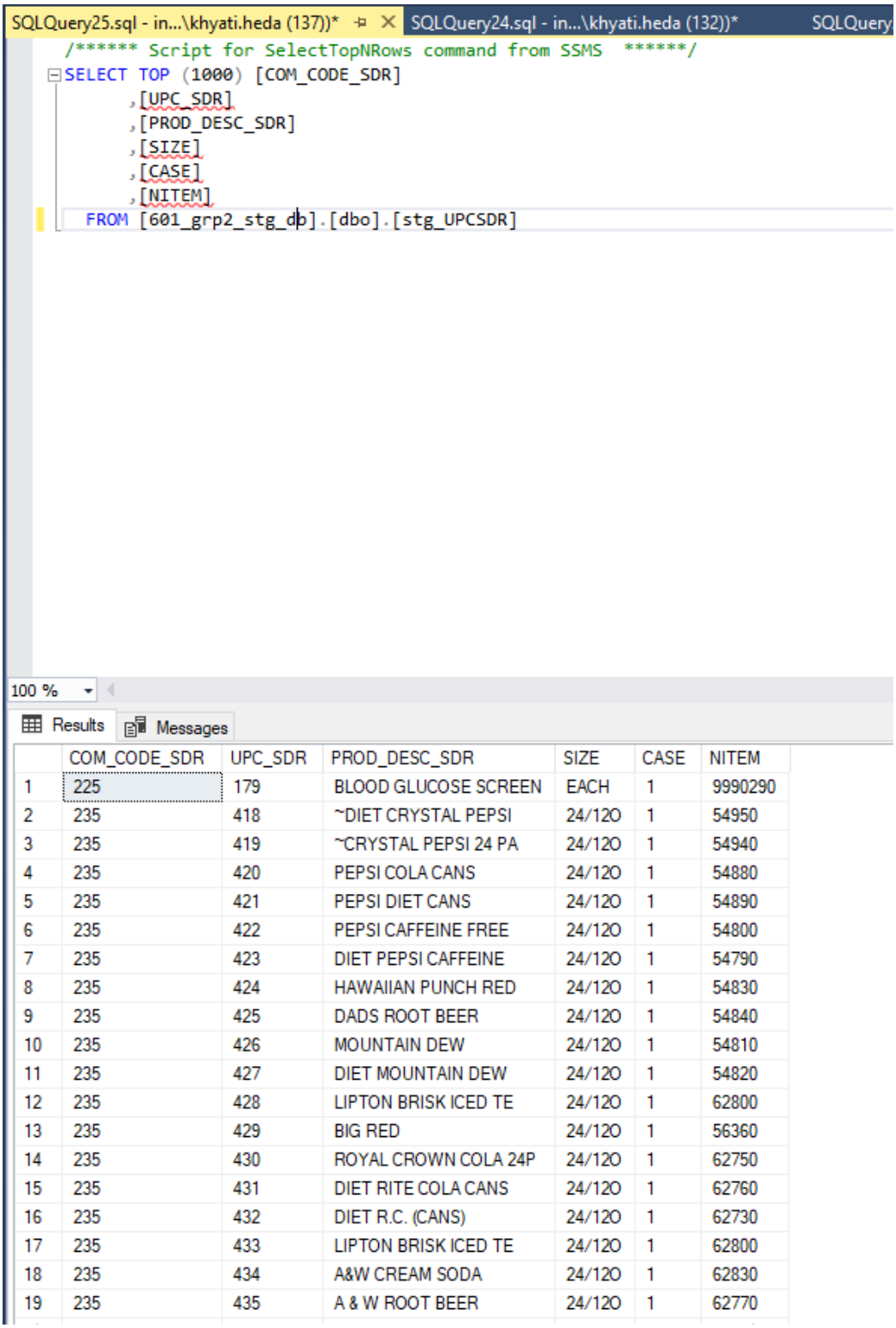


Fig: Data Flow Task in SSIS successfully executed



	COM_CODE_SDR	UPC_SDR	PROD_DESC_SDR	SIZE	CASE	NITEM
1	225	179	BLOOD GLUCOSE SCREEN	EACH	1	9990290
2	235	418	~DIET CRYSTAL PEPSI	24/12O	1	54950
3	235	419	~CRYSTAL PEPSI 24 PA	24/12O	1	54940
4	235	420	PEPSI COLA CANS	24/12O	1	54880
5	235	421	PEPSI DIET CANS	24/12O	1	54890
6	235	422	PEPSI CAFFEINE FREE	24/12O	1	54800
7	235	423	DIET PEPSI CAFFEINE	24/12O	1	54790
8	235	424	HAWAIIAN PUNCH RED	24/12O	1	54830
9	235	425	DADS ROOT BEER	24/12O	1	54840
10	235	426	MOUNTAIN DEW	24/12O	1	54810
11	235	427	DIET MOUNTAIN DEW	24/12O	1	54820
12	235	428	LIPTON BRISK ICED TE	24/12O	1	62800
13	235	429	BIG RED	24/12O	1	56360
14	235	430	ROYAL CROWN COLA 24P	24/12O	1	62750
15	235	431	DIET RITE COLA CANS	24/12O	1	62760
16	235	432	DIET R.C. (CANS)	24/12O	1	62730
17	235	433	LIPTON BRISK ICED TE	24/12O	1	62800
18	235	434	A&W CREAM SODA	24/12O	1	62830
19	235	435	A & W ROOT BEER	24/12O	1	62770

Fig: stg_upcsdr in staging database.

Data Cleaning

1. Cleaning stg_CCOUNT:

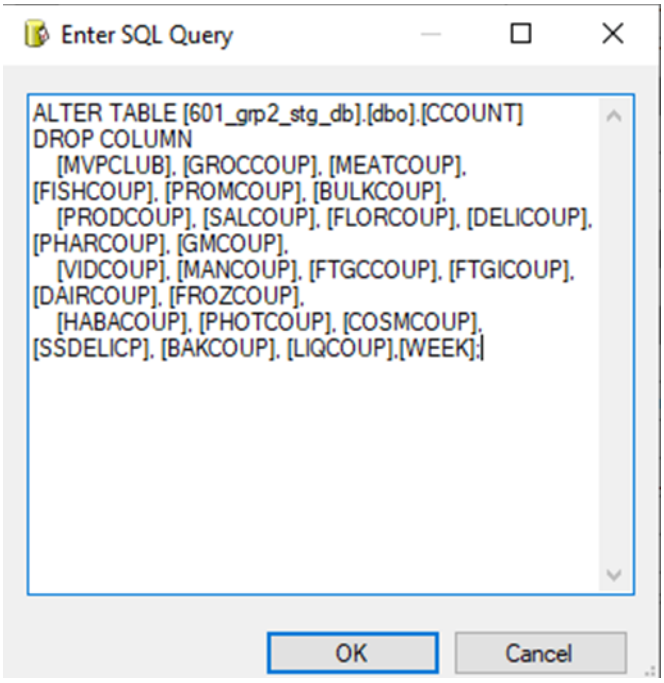


Fig: Dropping Columns not required

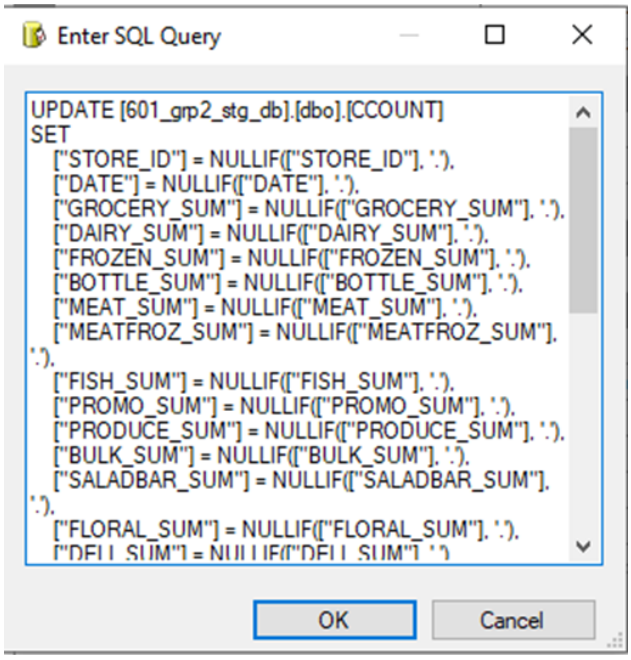


Fig: Replacing all "." values with NULL Values

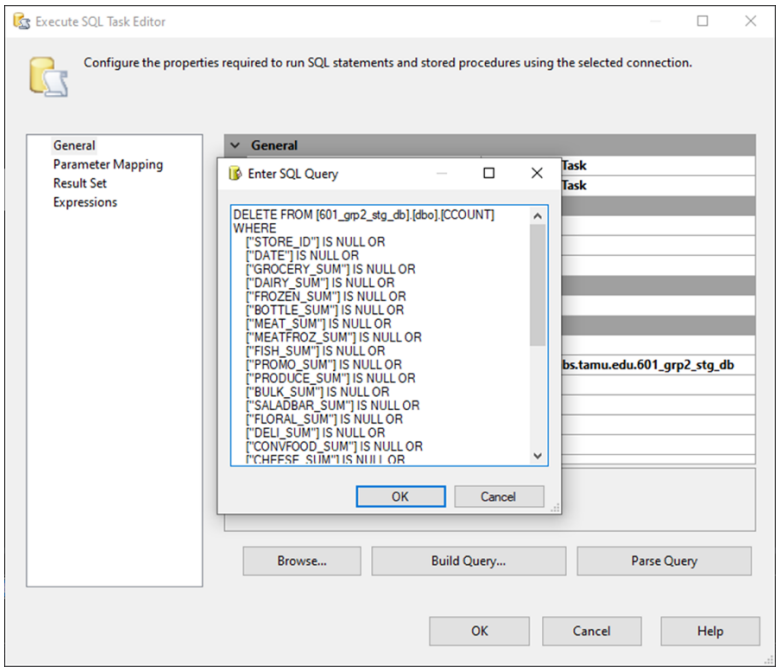


Fig: Deleting All Null Values

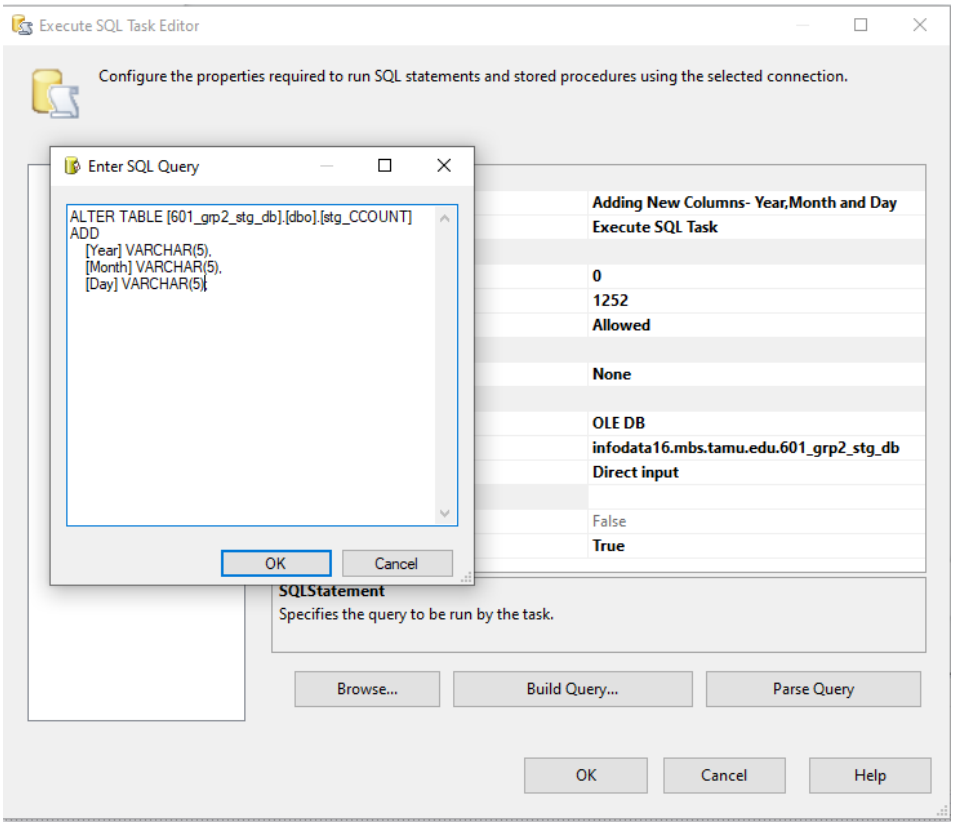


Fig: Adding New Columns- Year, Month and Day

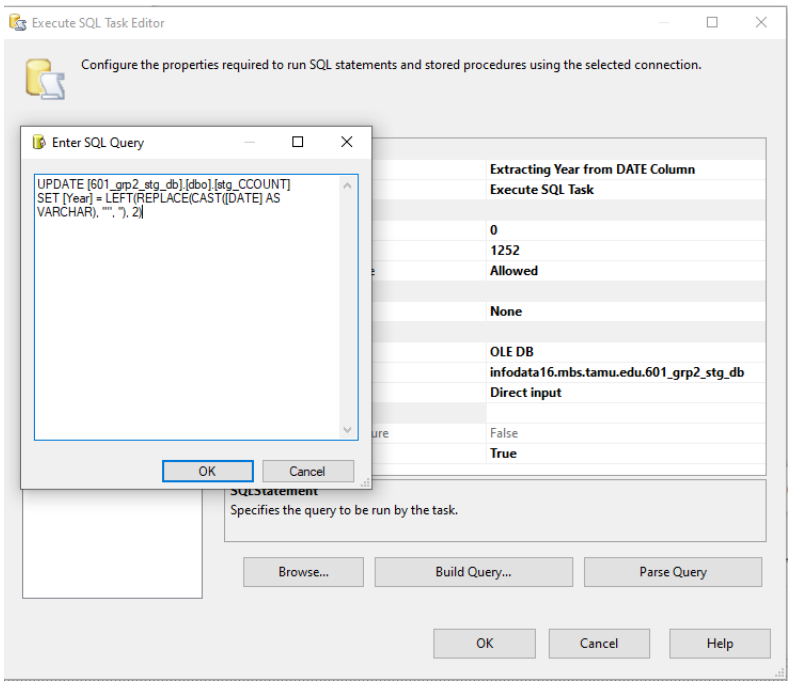


Fig: Extracting Year from DATE Column

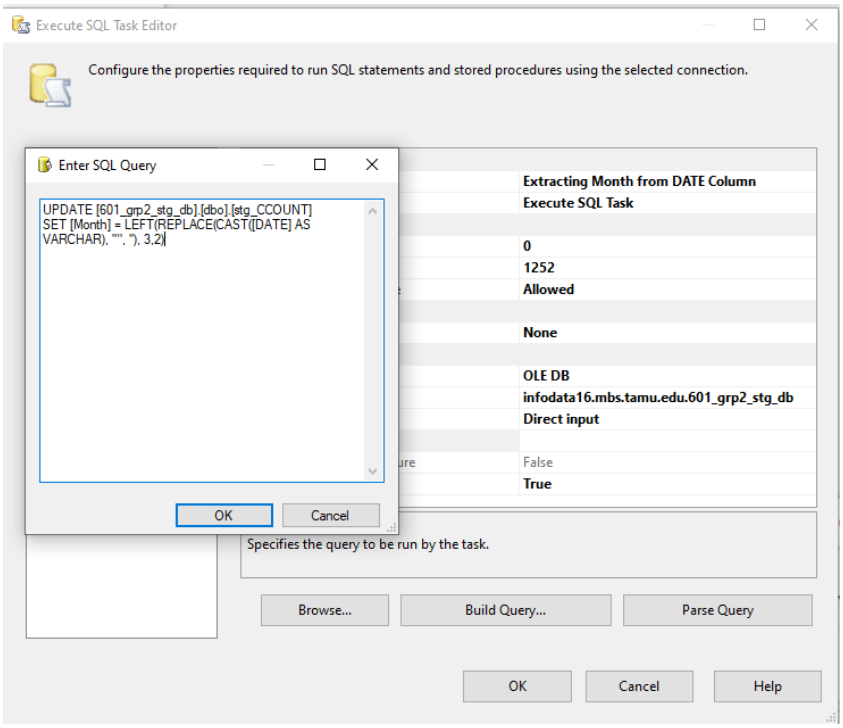


Fig: Extracting Month from DATE Column

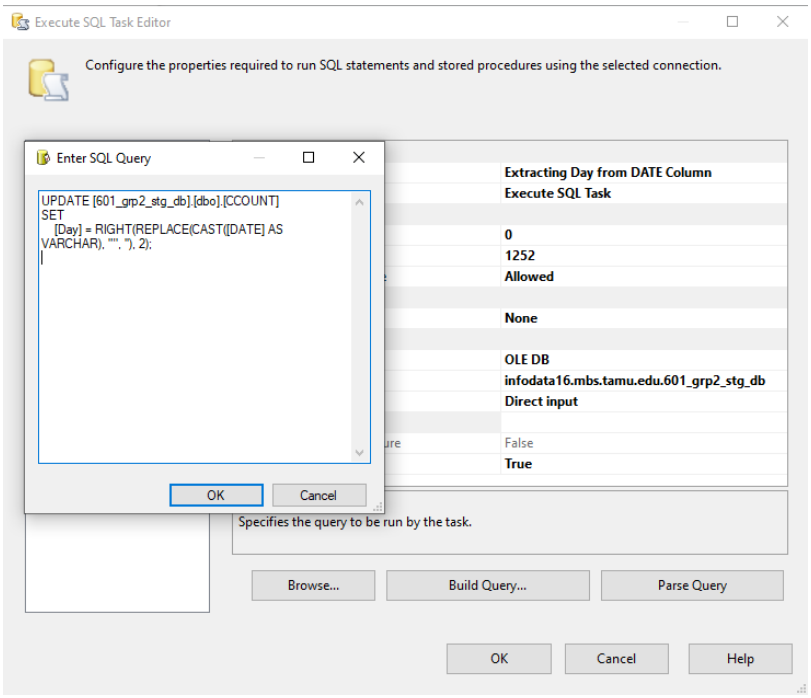


Fig: Extracting Day from DATE Column

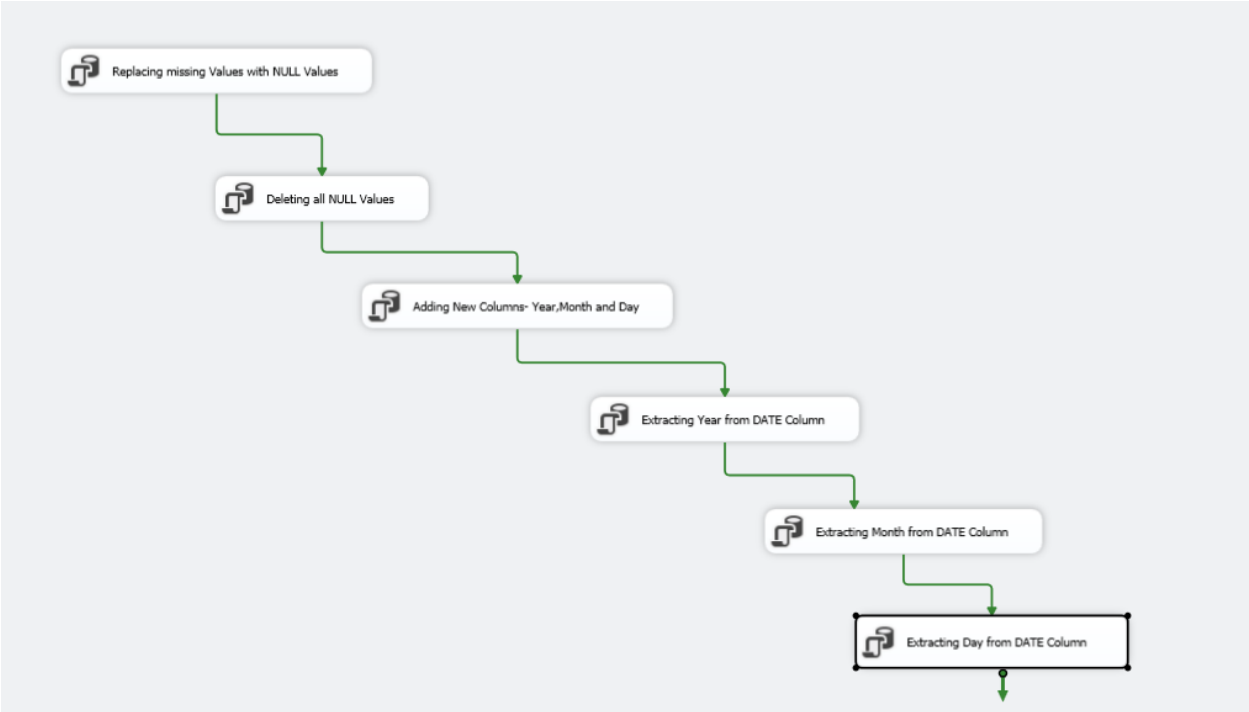
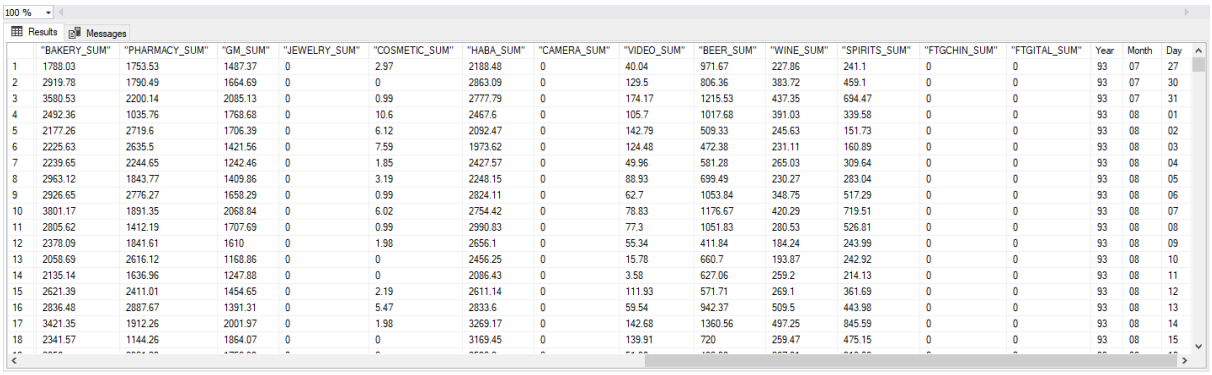


Fig: SQL Tasks in SSIS



	"BAKERY_SUM"	"PHARMACY_SUM"	"GM_SUM"	"JEWELRY_SUM"	"COSMETIC_SUM"	"HABA_SUM"	"CAMERA_SUM"	"VIDEO_SUM"	"BEER_SUM"	"WINE_SUM"	"SPIRITS_SUM"	"FTGCHIN_SUM"	"FTGITAL_SUM"	Year	Month	Day
1	1788.03	1753.53	1487.37	0	2.97	2188.48	0	40.04	971.67	227.86	241.1	0	0	93	07	27
2	2919.78	1790.49	1664.69	0	0	2863.09	0	129.5	806.36	383.72	459.1	0	0	93	07	30
3	3580.53	2200.14	2085.13	0	0.99	2777.79	0	174.17	1215.53	437.35	694.47	0	0	93	07	31
4	2492.36	1035.76	1768.68	0	10.6	2467.6	0	105.7	1017.68	391.03	339.58	0	0	93	08	01
5	2177.26	2719.6	1706.39	0	6.12	2092.47	0	142.79	509.33	245.63	151.73	0	0	93	08	02
6	2225.63	2635.5	1421.56	0	7.59	1973.62	0	124.48	472.38	231.11	160.89	0	0	93	08	03
7	2239.65	2244.65	1242.46	0	1.85	2427.57	0	49.96	581.28	265.03	309.64	0	0	93	08	04
8	2963.12	1843.77	1409.86	0	3.19	2248.15	0	88.93	699.49	230.27	283.04	0	0	93	08	05
9	2926.65	2776.27	1658.29	0	0.99	2824.11	0	62.7	1053.84	348.75	517.29	0	0	93	08	06
10	3801.17	1891.35	2068.84	0	6.02	2754.42	0	78.83	1176.67	420.29	719.51	0	0	93	08	07
11	2805.62	1412.19	1707.69	0	0.99	2990.83	0	77.3	1051.83	280.53	526.81	0	0	93	08	08
12	2378.09	1841.61	1610	0	1.98	2656.1	0	55.34	411.84	184.24	243.99	0	0	93	08	09
13	2058.69	2616.12	1168.86	0	0	2456.25	0	15.78	660.7	193.87	242.92	0	0	93	08	10
14	2135.14	1636.96	1247.88	0	0	2086.43	0	3.58	627.06	259.2	214.13	0	0	93	08	11
15	2621.39	2411.01	1454.65	0	2.19	2611.14	0	111.93	571.71	269.1	361.69	0	0	93	08	12
16	2836.48	2887.67	1391.31	0	5.47	2833.6	0	59.54	942.37	509.5	443.98	0	0	93	08	13
17	3421.35	1912.26	2001.97	0	1.98	3269.17	0	142.68	1360.56	497.25	845.59	0	0	93	08	14
18	2341.57	1144.26	1864.07	0	0	3169.45	0	139.91	720	259.47	475.15	0	0	93	08	15

Fig: Clean Data in SSMS

2. Creating new temp table called stg_time:

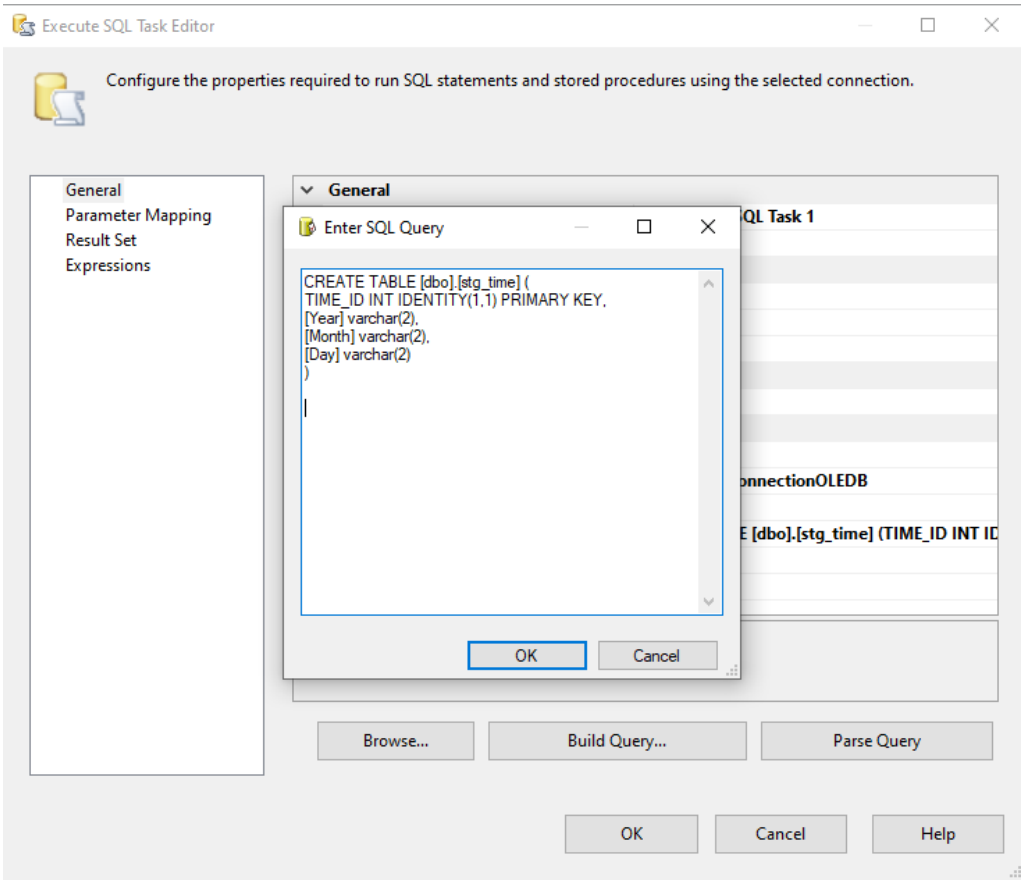


Fig: Creating stg_time table

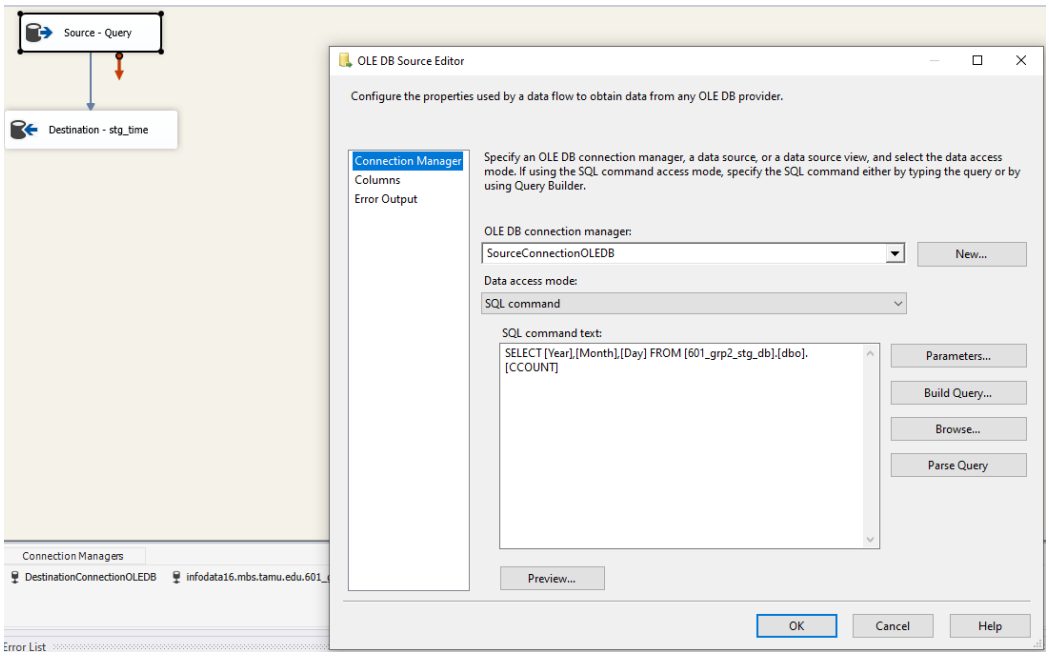


Fig: Query to load data to stg_time

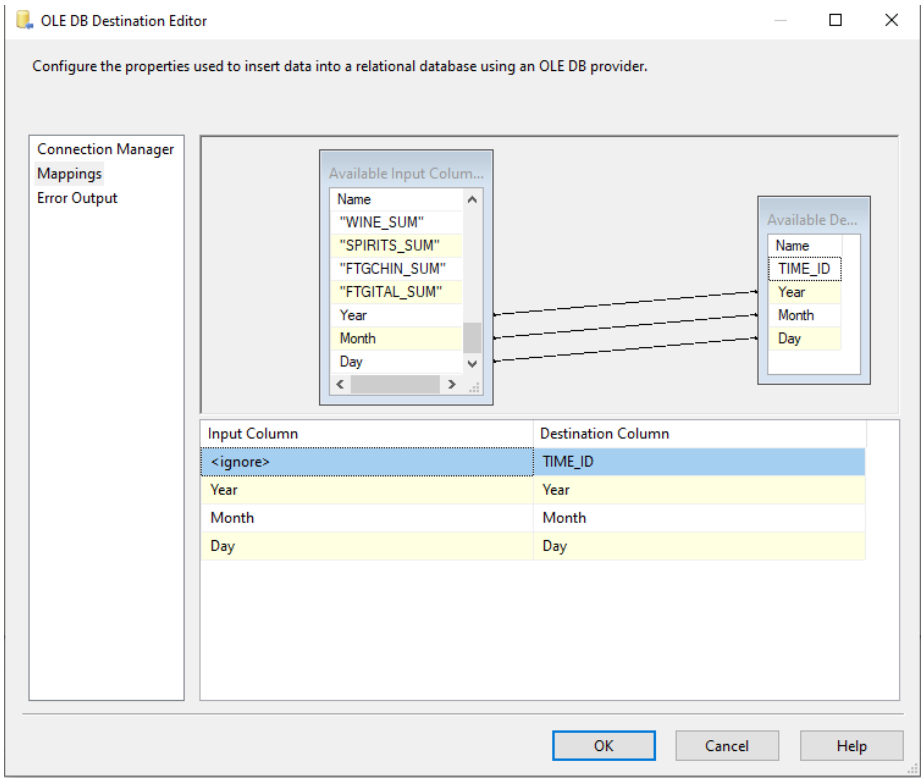


Fig: Column mapping from CCount to stg_time

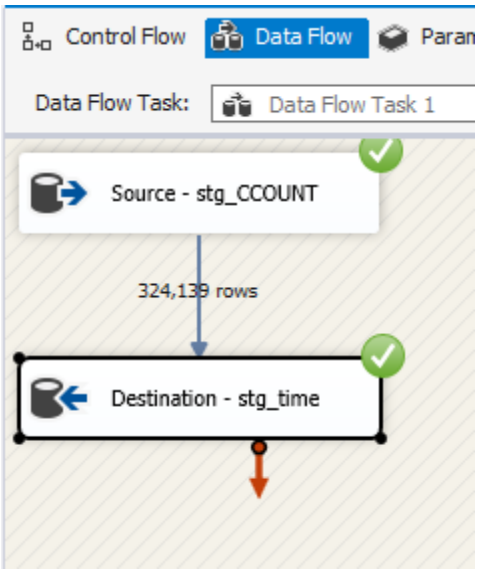


Fig: Loading data into temp table stg_time

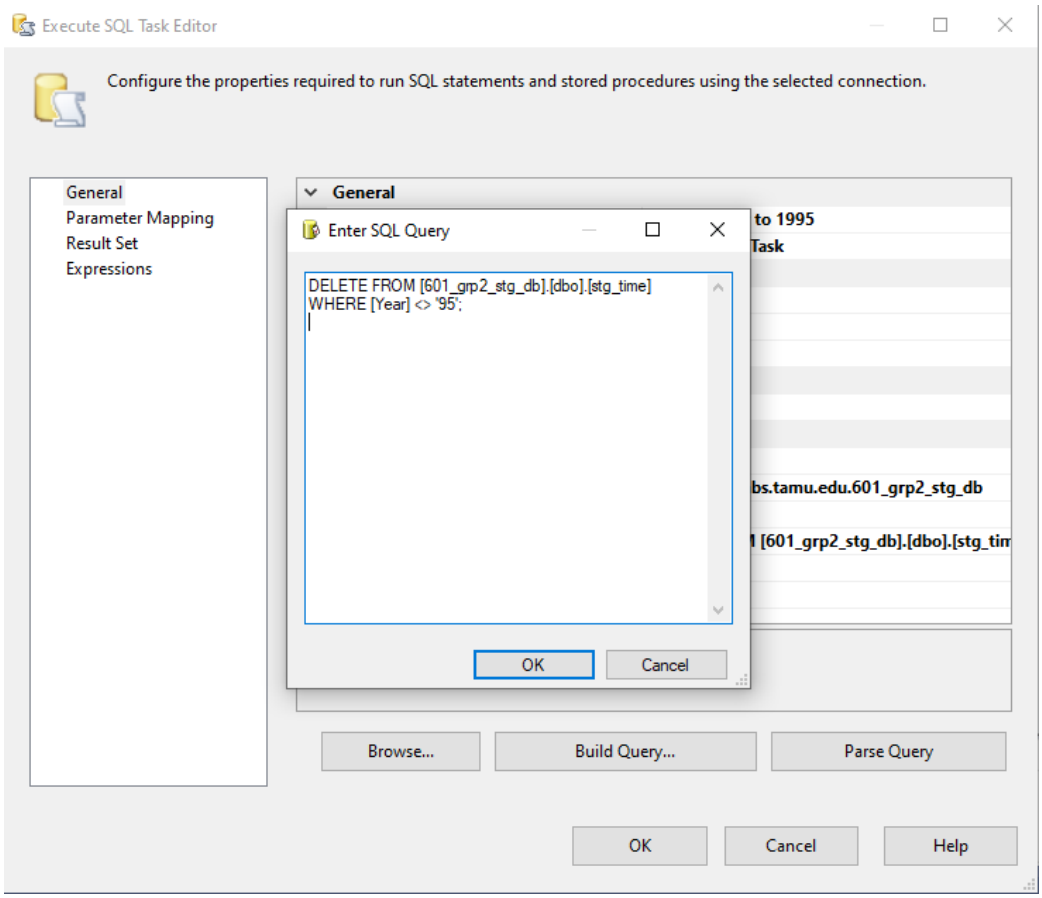


Fig: Filtering year to 1995 in stg_time

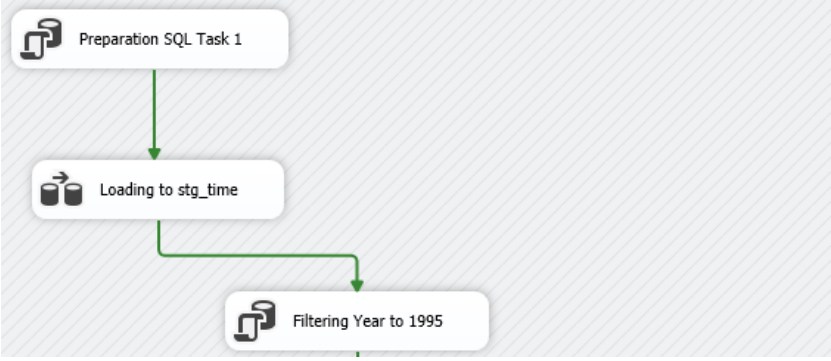
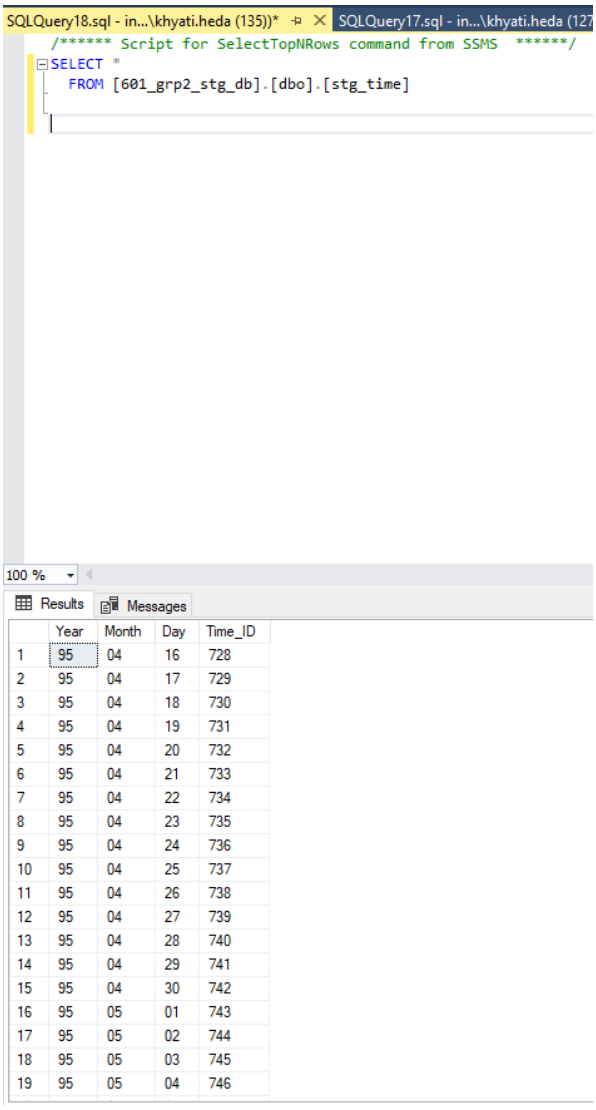


Fig: SQL Tasks in SSIS



	Year	Month	Day	Time_ID
1	95	04	16	728
2	95	04	17	729
3	95	04	18	730
4	95	04	19	731
5	95	04	20	732
6	95	04	21	733
7	95	04	22	734
8	95	04	23	735
9	95	04	24	736
10	95	04	25	737
11	95	04	26	738
12	95	04	27	739
13	95	04	28	740
14	95	04	29	741
15	95	04	30	742
16	95	05	01	743
17	95	05	02	744
18	95	05	03	745
19	95	05	04	746

Fig: Clean data in SSMS

3. Cleaning stg_demo:

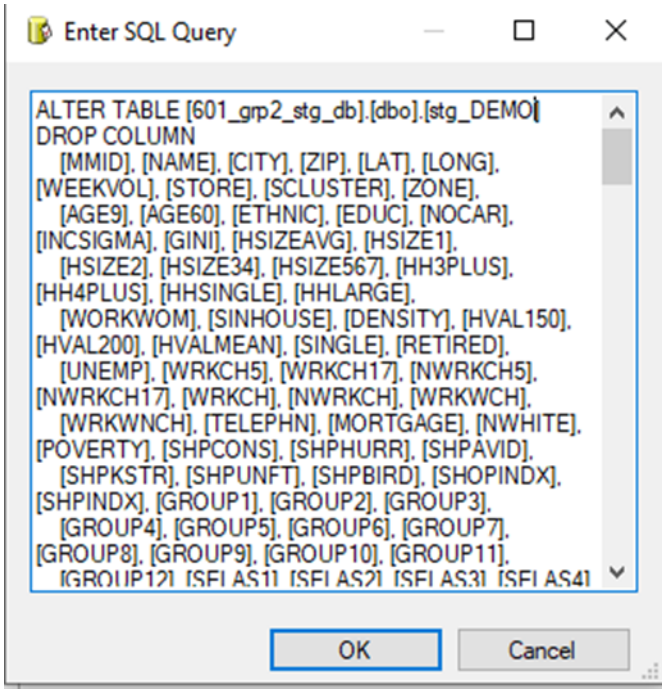


Fig: Dropping Columns not required

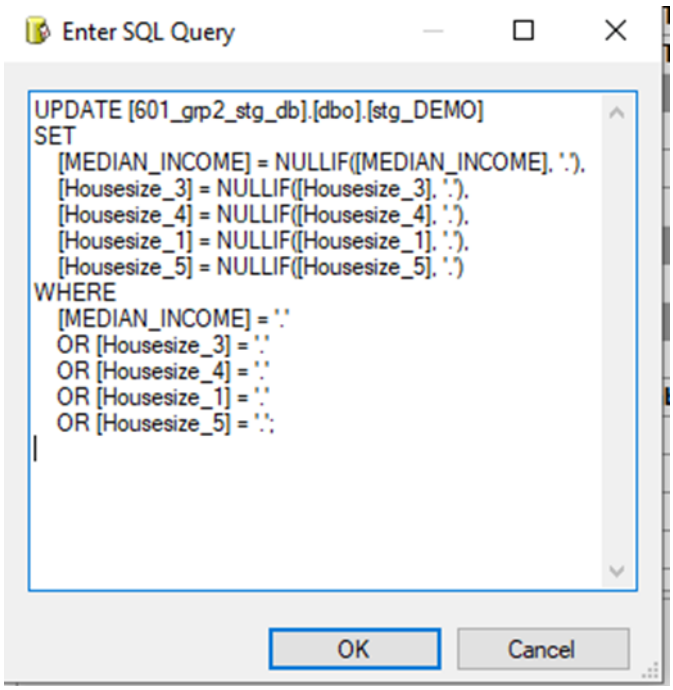


Fig: Replacing all "." values with NULL Values

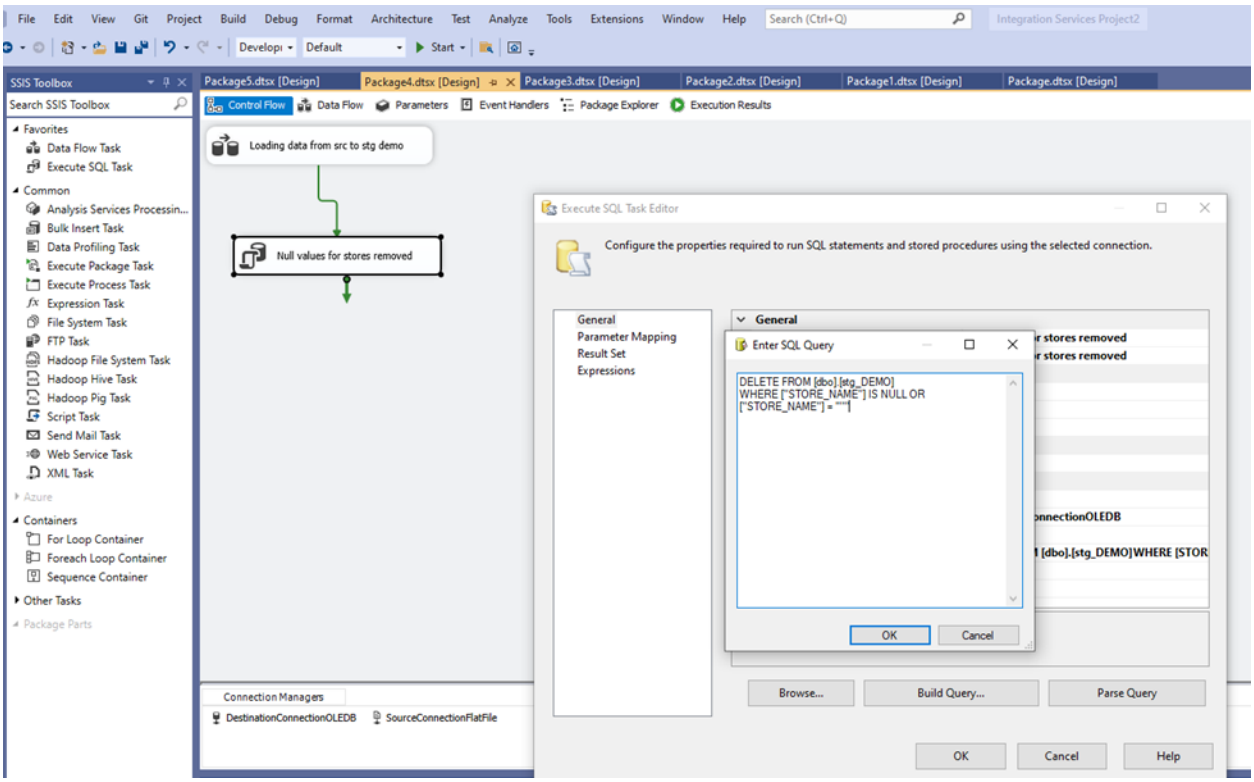


Fig: Dropping Tables where Store Name is NULL

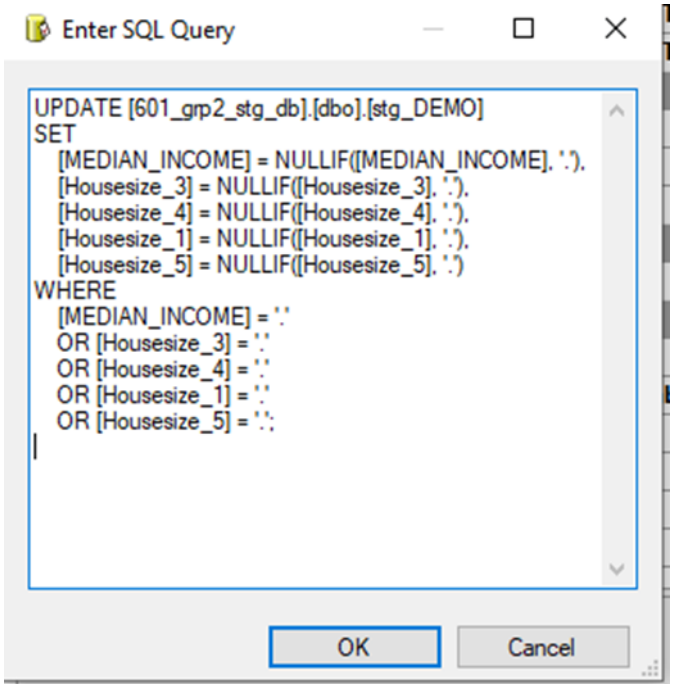


Fig: Replacing all "." values with NULL Values

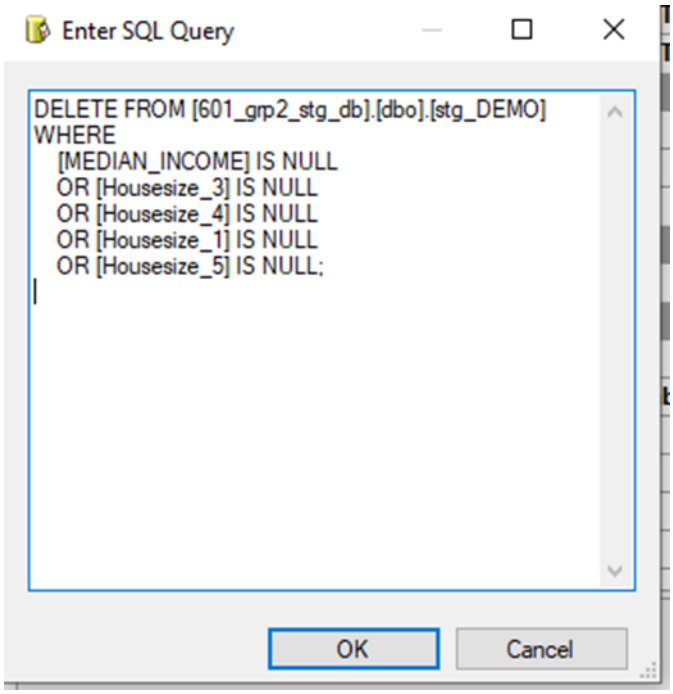


Fig: Deleting all NULL Values

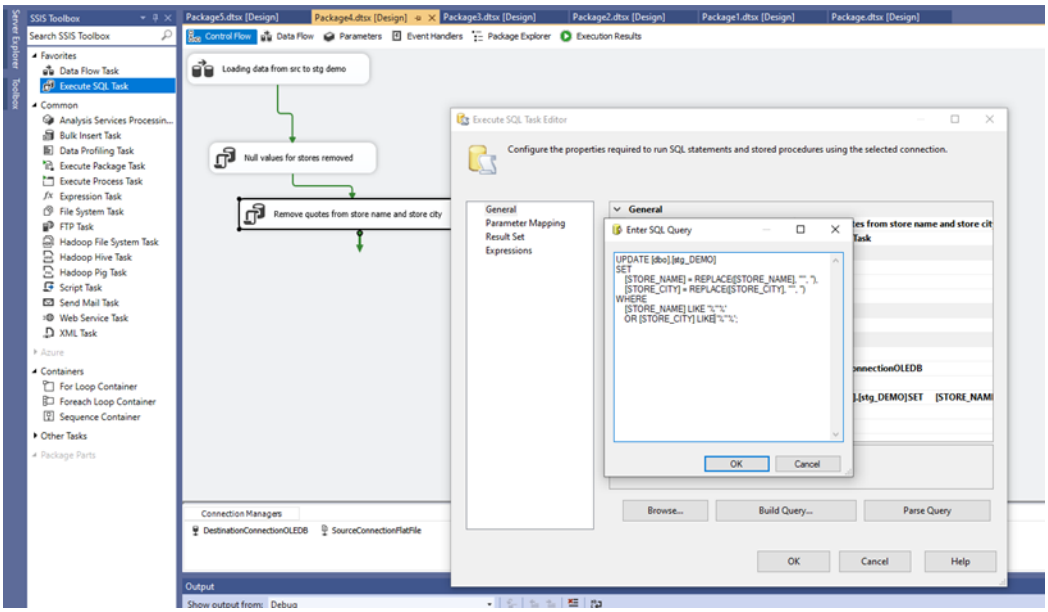


Fig: Removing Extra Double Quotes

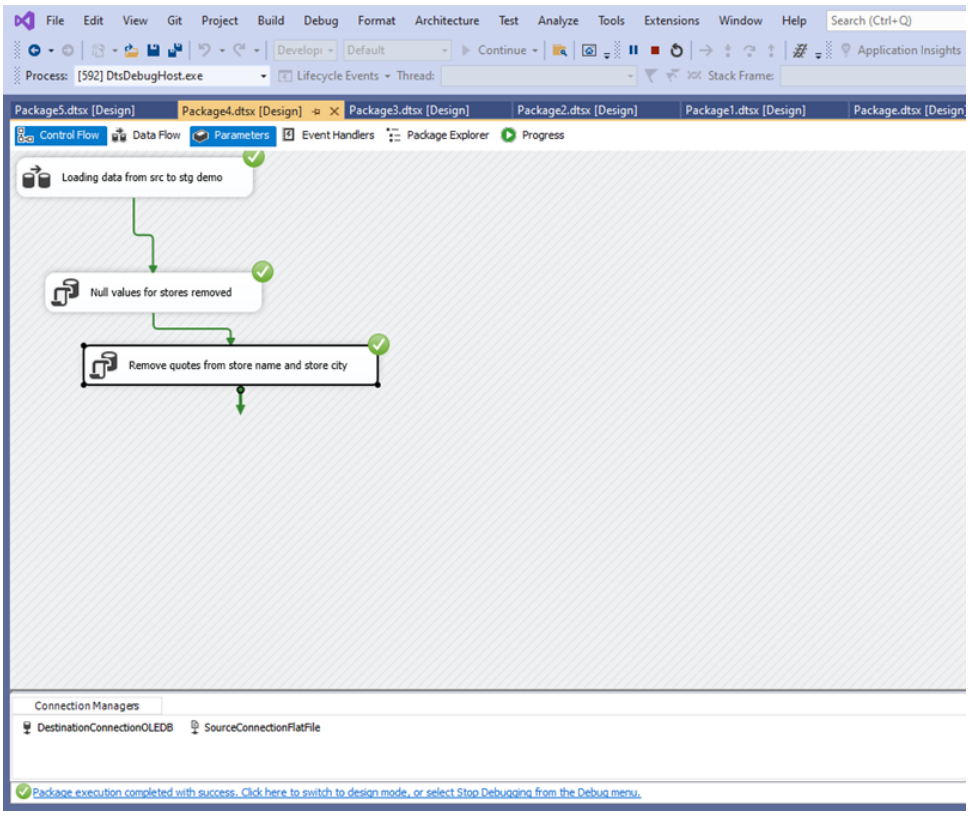
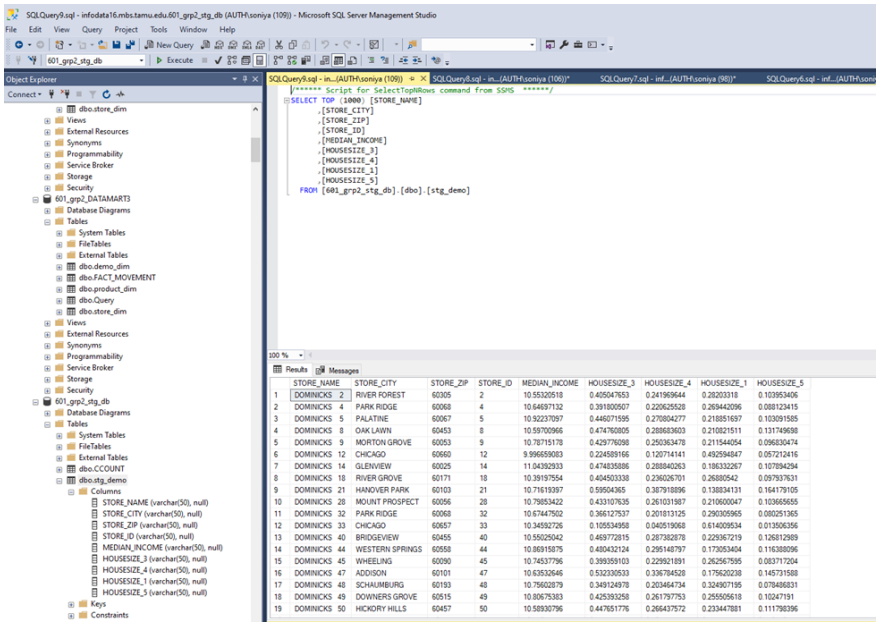


Fig: SQL Tasks in SSIS



	STORE_NAME	STORE_CITY	STORE_ZIP	STORE_ID	MEDIAN_INCOME	HOUSESIZE_3	HOUSESIZE_4	HOUSESIZE_1	HOUSESIZE_5
1	DOMINICKS 2	RIVER FOREST	60305	2	10.55320518	0.405047653	0.241969644	0.28203318	0.103953406
2	DOMINICKS 4	PARK RIDGE	60068	4	10.64697132	0.391800507	0.220625528	0.269442096	0.088123415
3	DOMINICKS 5	PALATINE	60067	5	10.92237097	0.446071595	0.270804277	0.218851697	0.103091585
4	DOMINICKS 8	OAK LAWN	60453	8	10.59700966	0.474760805	0.288683603	0.210821511	0.131749698
5	DOMINICKS 9	MORTON GROVE	60053	9	10.78715178	0.429776098	0.250363478	0.211544054	0.096830474
6	DOMINICKS 12	CHICAGO	60660	12	9.966859093	0.224589166	0.120714141	0.492594847	0.057212416
7	DOMINICKS 14	GLENVIEW	60025	14	11.04392933	0.474835886	0.288840263	0.186332267	0.107894294
8	DOMINICKS 18	RIVER GROVE	60171	18	10.39197554	0.404503338	0.236026701	0.26880542	0.097937631
9	DOMINICKS 21	HANOVER PARK	60103	21	10.71619397	0.59504365	0.387918896	0.138834131	0.164179105
10	DOMINICKS 28	MOUNT PROSPECT	60056	28	10.79853422	0.433107635	0.261031987	0.210600047	0.103665655
11	DOMINICKS 32	PARK RIDGE	60068	32	10.67447502	0.366127537	0.201813125	0.290305965	0.080251365
12	DOMINICKS 33	CHICAGO	60657	33	10.34592726	0.105534958	0.040519068	0.614009534	0.013506356
13	DOMINICKS 40	BRIDGEVIEW	60455	40	10.55025042	0.469772815	0.287382878	0.225367219	0.125812989
14	DOMINICKS 44	WESTERN SPRINGS	60558	44	10.86915875	0.480432124	0.295148797	0.173053404	0.116388096
15	DOMINICKS 45	WHEELING	60090	45	10.74537796	0.399359103	0.229921891	0.262567595	0.083717204
16	DOMINICKS 47	ADDISON	60101	47	10.63532646	0.532330533	0.336784528	0.175620238	0.145731588
17	DOMINICKS 48	SCHAUMBURG	60193	48	10.75602879	0.349124978	0.203464734	0.324907195	0.078486831
18	DOMINICKS 49	DOWNERS GROVE	60515	49	10.80675383	0.425393258	0.261797753	0.255505618	0.10247191
19	DOMINICKS 50	HICKORY HILLS	60457	50	10.58930796	0.447651776	0.266437572	0.233447881	0.111798396

Fig: Clean Data in SSMS

4. Cleaning stg_upcfrd:

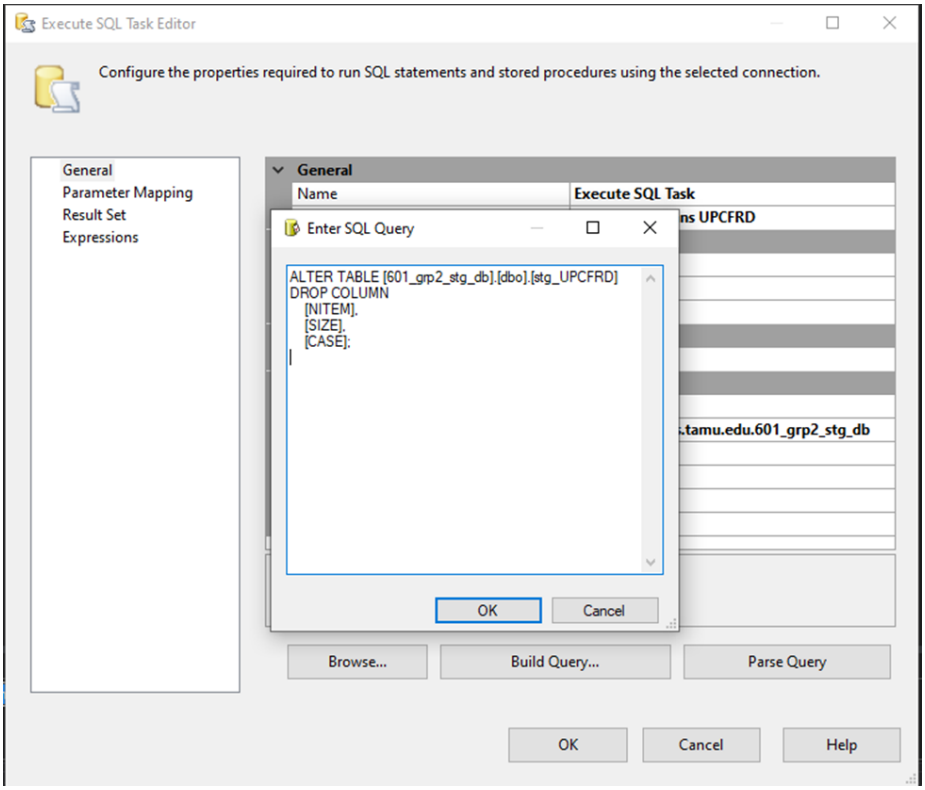


Fig: Dropping Columns not required

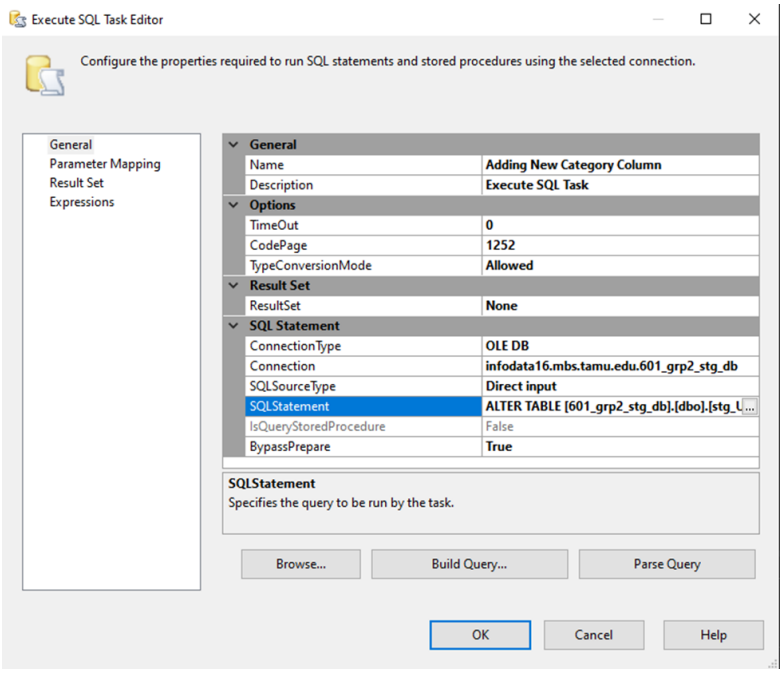


Fig: Adding New Column called Category and setting value to Frozen Dinner

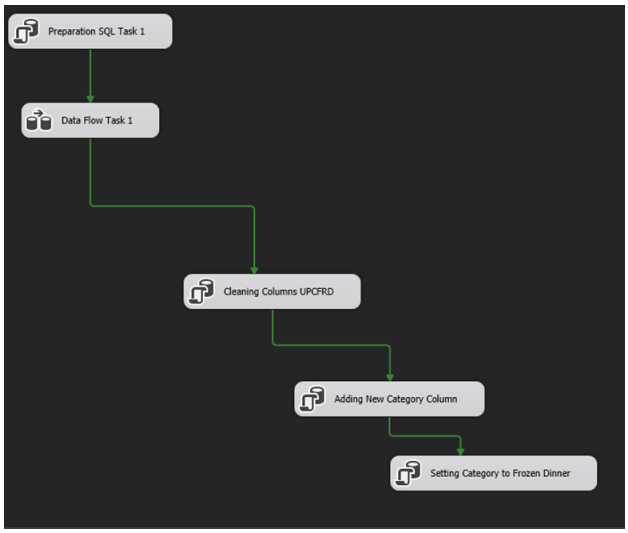


Fig: SQL Task in SSIS

SQLQuery19.sql - in...H\khyati.heda (68)) SQLQuery17.sql - in...H\khyati.heda (55)) SQLQuery

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) ["COM_CODE_FRD"]
      ,["UPC_ID_FRD"]
      ,["PROD_DESC_FRD"]
      ,["CATEGORY"]
  FROM [601_grp2_stg_db].[dbo].[stg_UPCFRD]
```

100 %

Results Messages

	"COM_CODE_FRD"	"UPC_ID_FRD"	"PROD_DESC_FRD"	"CATEGORY"
1	104	1380013201	"LC HP CHICKEN FLOREN"	Frozen Dinner
2	104	1380013202	"LC HP ROASTED TURKEY"	Frozen Dinner
3	104	1380013203	"LC HP SRLN BF TIPS"	Frozen Dinner
4	104	1380013204	"LC HP GRLD CHKN W/PE"	Frozen Dinner
5	104	1380013205	"LC HP JUMBO RIGATONI"	Frozen Dinner
6	104	1380013206	"LC HP ORIENTAL GLAZE"	Frozen Dinner
7	104	1380013207	"LC HP BEEF LO MEIN"	Frozen Dinner
8	104	1380013208	"LC HP ROASTED CHICKE"	Frozen Dinner
9	104	1380013209	"LC HEARTY PORTIONS L"	Frozen Dinner
10	104	1380013210	"LC HRTY PORTIONS CHS"	Frozen Dinner
11	104	1380013304	"STOUFFER'S HRTY PRTN"	Frozen Dinner
12	104	1380013305	"STOUFFER'S HRTY PRTN"	Frozen Dinner
13	104	1380013306	"STOUFFER'S HRTY PRTN"	Frozen Dinner
14	104	1380013307	"STOUFFER'S HRTY PRTN"	Frozen Dinner
15	104	1380013308	"STOUFFER'S HRTY PRTN"	Frozen Dinner
16	104	1380013309	"STOUFFER'S HRTY PRTN"	Frozen Dinner
17	104	1380013310	"STOUFFERS HP CNTRY F"	Frozen Dinner
18	104	1380013311	"STOUFFER'S HRTY PRTN"	Frozen Dinner
19	104	1380013312	"STOUFFERS HEARTY POR"	Frozen Dinner

Fig: Clean Data in SSMS

5. Cleaning stg_Wfrd:

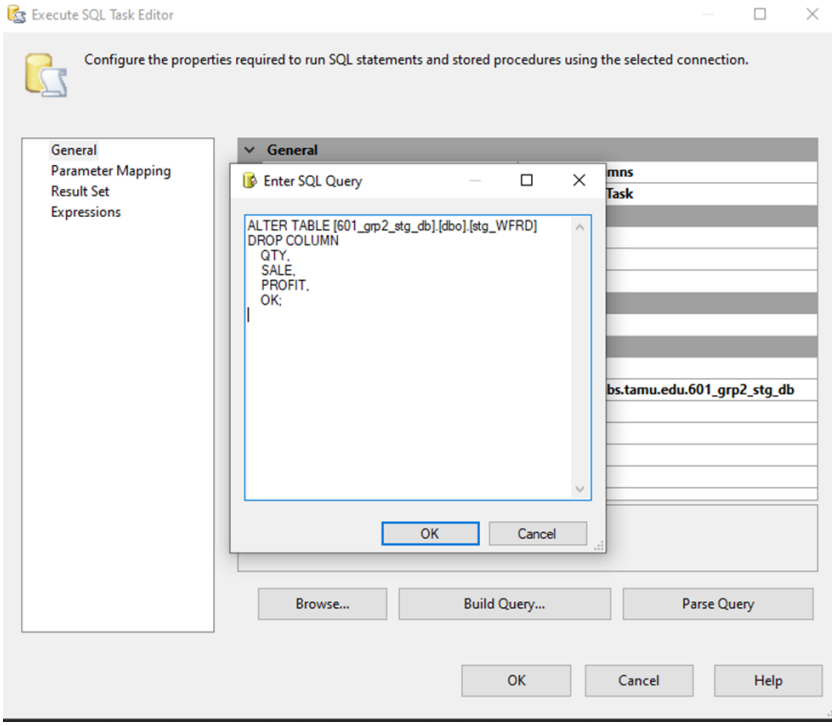


Fig: Dropping Columns not required

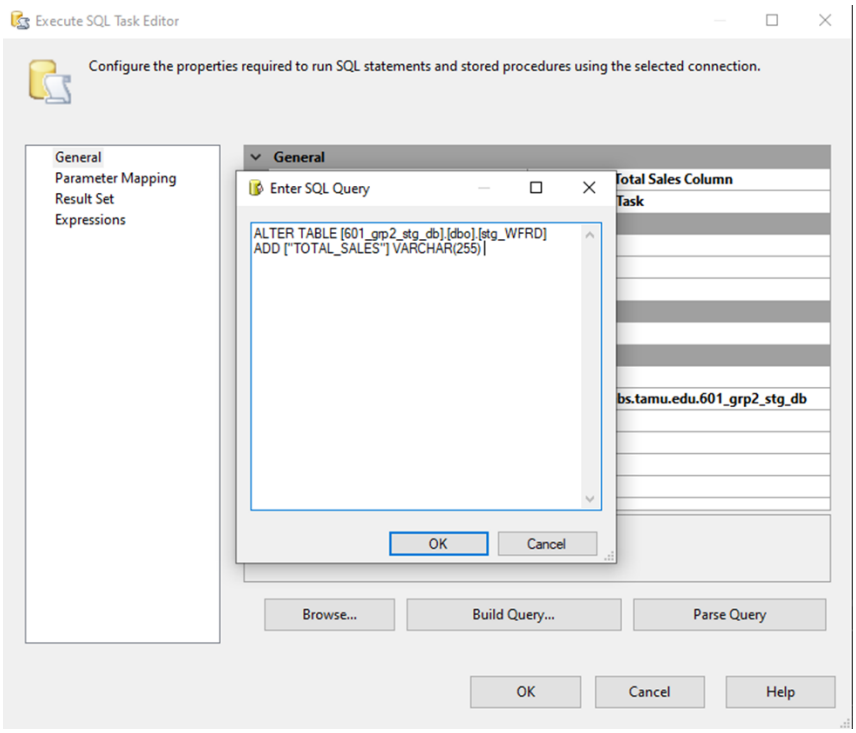


Fig: Adding New Column called Total_Sales

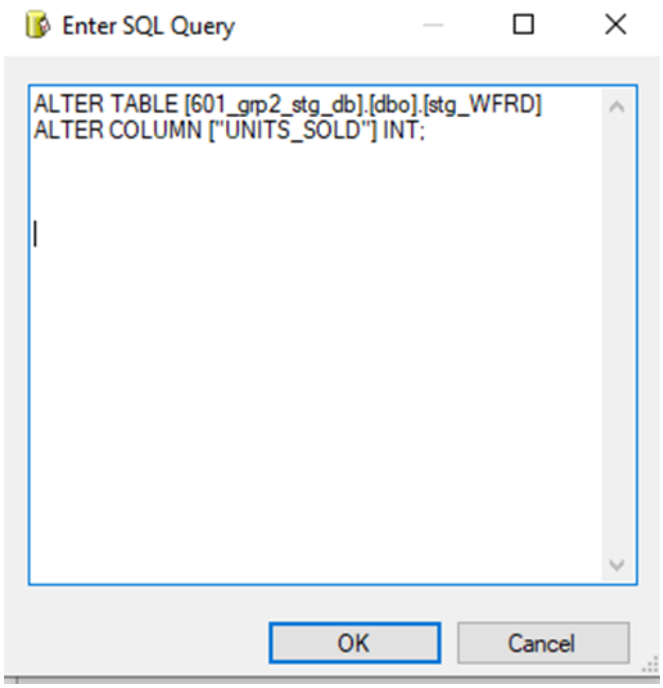


Fig: Altering data type of Units_Sold

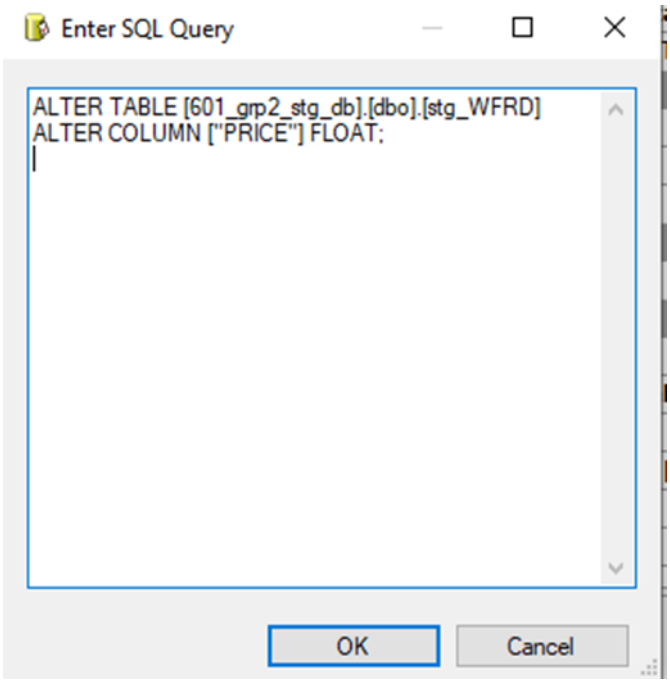


Fig: Altering data type of Price

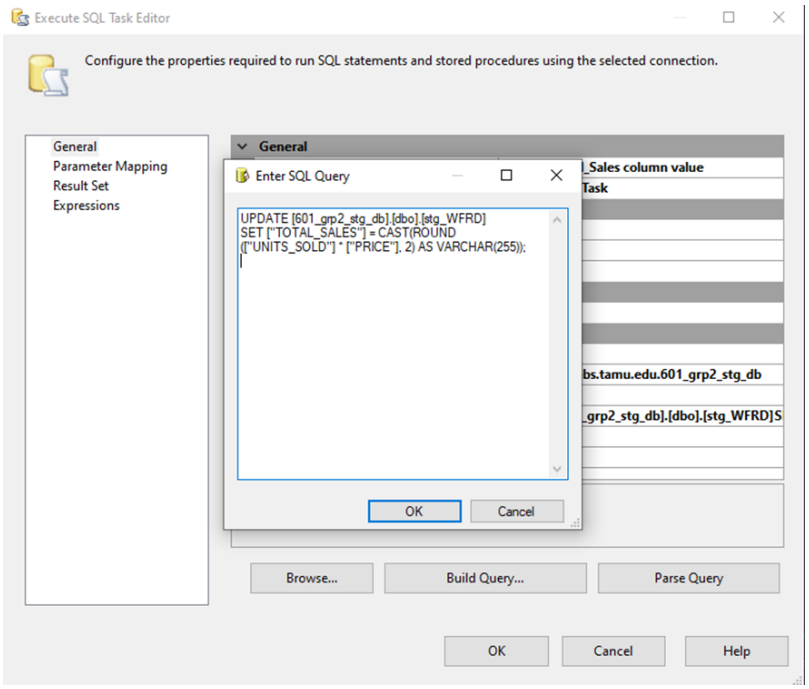


Fig: Updating Total Sales = Units_sold * Price

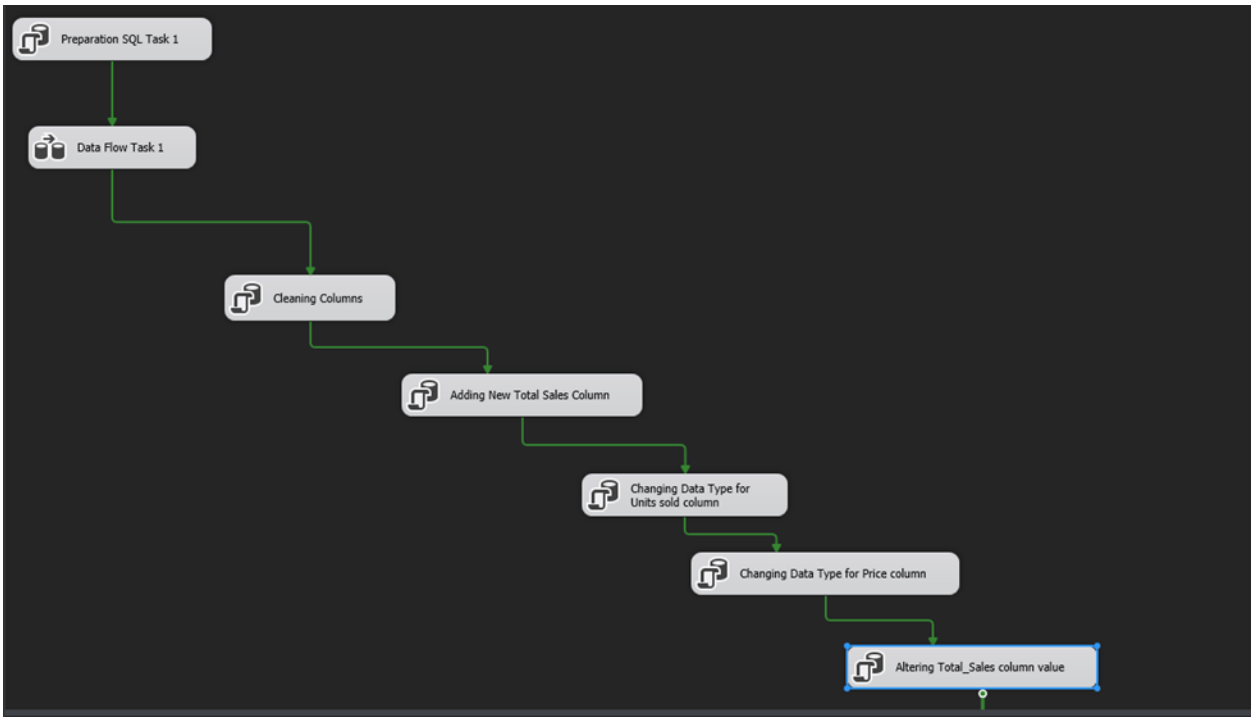


Fig: SQL Task in SSIS

SQLQuery20.sql - in...H\khyati.heda (96)) | SQLQuery19.sql - in...H\khyati.heda (68)) | SQLQue

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) ["STORE_ID_FRD"]
      ,["UPC_ID_FRD"]
      ,["UNITS_SOLD"]
      ,["PRICE"]
      ,["TOTAL_SALES"]
  FROM [601_grp2_stg_db].[dbo].[stg_WFRD]
```

100 %

Results | Messages

	"STORE_ID_FRD"	"UPC_ID_FRD"	"UNITS_SOLD"	"PRICE"	"TOTAL_SALES"
4	2	1380013201	11	2.5	27.50
5	2	1380013201	1	2.99	2.99
6	2	1380013201	3	2.99	8.97
7	2	1380013201	2	2.99	5.98
8	2	1380013201	3	2.99	8.97
9	2	1380013201	8	2.99	23.92
10	2	1380013201	7	2.99	20.93
11	2	1380013201	2	2.99	5.98
12	2	1380013201	0	0	0.00
13	2	1380013201	1	2.99	2.99
14	2	1380013201	3	2.99	8.97
15	2	1380013201	2	2.99	5.98
16	2	1380013201	0	0	0.00
17	2	1380013201	1	2.99	2.99
18	2	1380013201	5	2.99	14.95
19	2	1380013201	1	2.99	2.99
20	2	1380013201	2	2.99	5.98
21	2	1380013201	4	2.99	11.96
22	2	1380013201	4	2.99	11.96

Fig: Clean Data in SSMS

6. Creating new temp table: stg_FRD:

- Now we are creating a temp table called as stg_FRD which will join the stg_upcfrd and stg_wfrd

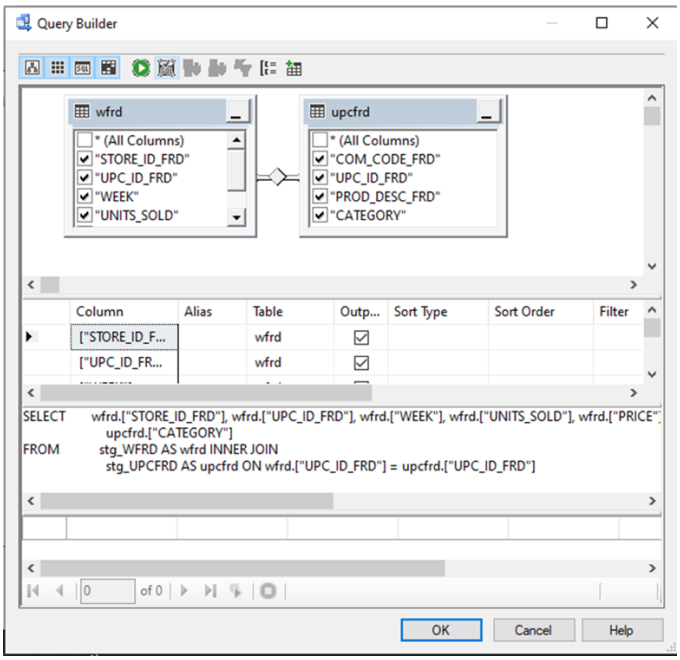


Fig: Joining both the tables and storing it into stg_FRD

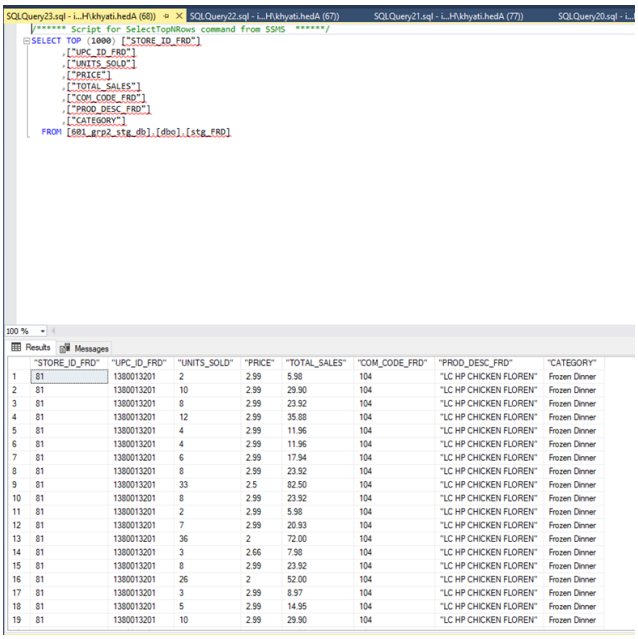


Fig: stg_FRD Table in SSMS

7. Cleaning stg_upcsdr:

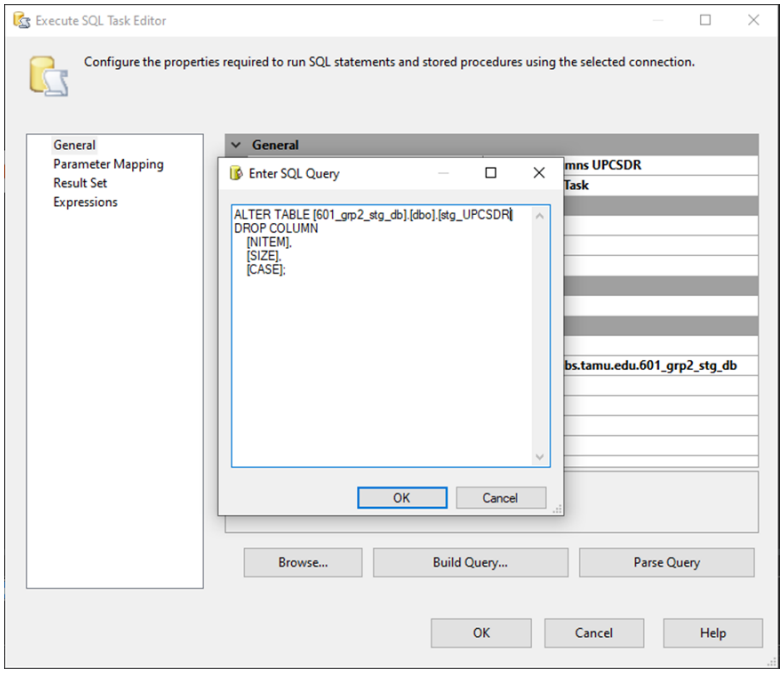


Fig: Dropping Columns not required

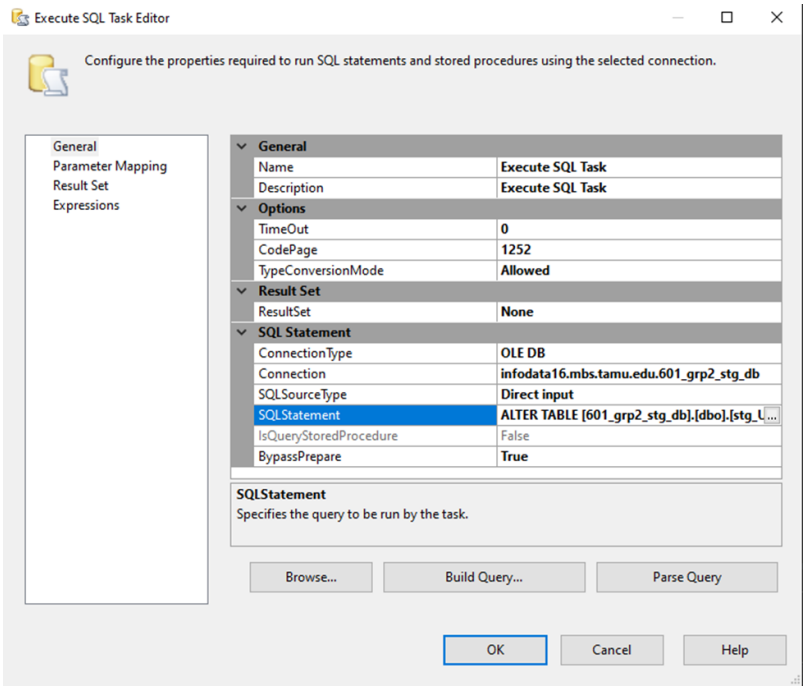


Fig: Adding New Column called Category and setting value to Soft Drinks

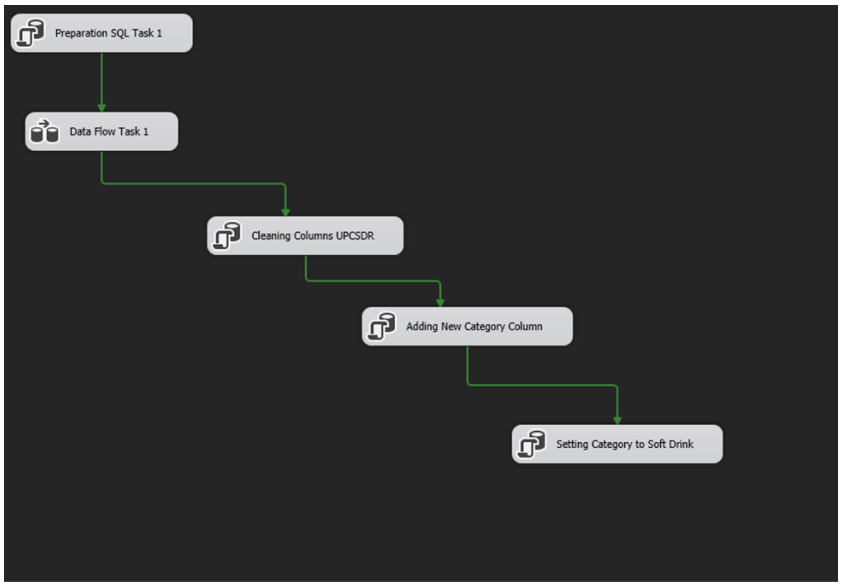
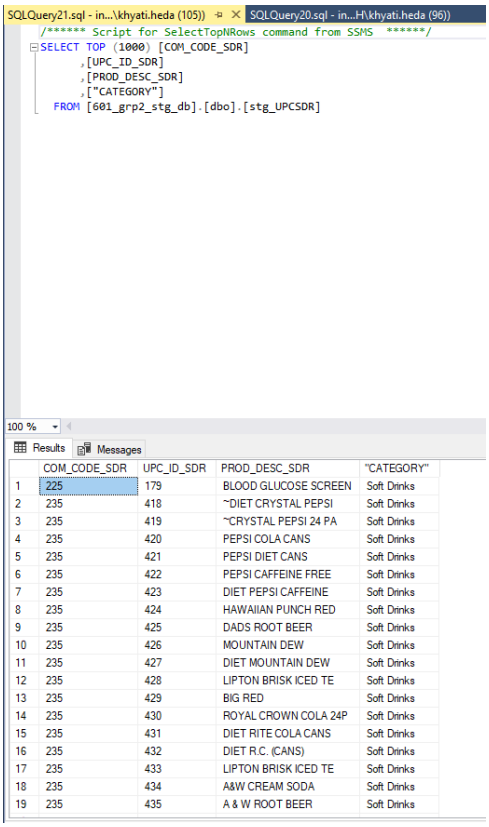


Fig: SQL Task in SSIS



SQLQuery21.sql - in...\khyati.heda (105)) | SQLQuery20.sql - in...H\khyati.heda (96))

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [COM_CODE_SDR]
      ,[UPC_ID_SDR]
      ,[PROD_DESC_SDR]
      ,["CATEGORY"]
  FROM [601_grp2_stg_db].[dbo].[stg_UPCSDR]
```

100 %

Results | Messages

	COM_CODE_SDR	UPC_ID_SDR	PROD_DESC_SDR	"CATEGORY"
1	225	179	BLOOD GLUCOSE SCREEN	Soft Drinks
2	235	418	~DIET CRYSTAL PEPSI	Soft Drinks
3	235	419	~CRYSTAL PEPSI 24 PA	Soft Drinks
4	235	420	PEPSI COLA CANS	Soft Drinks
5	235	421	PEPSI DIET CANS	Soft Drinks
6	235	422	PEPSI CAFFEINE FREE	Soft Drinks
7	235	423	DIET PEPSI CAFFEINE	Soft Drinks
8	235	424	HAWAIIAN PUNCH RED	Soft Drinks
9	235	425	DADS ROOT BEER	Soft Drinks
10	235	426	MOUNTAIN DEW	Soft Drinks
11	235	427	DIET MOUNTAIN DEW	Soft Drinks
12	235	428	LIPTON BRISK ICED TE	Soft Drinks
13	235	429	BIG RED	Soft Drinks
14	235	430	ROYAL CROWN COLA 24P	Soft Drinks
15	235	431	DIET RITE COLA CANS	Soft Drinks
16	235	432	DIET R.C. (CANS)	Soft Drinks
17	235	433	LIPTON BRISK ICED TE	Soft Drinks
18	235	434	A&W CREAM SODA	Soft Drinks
19	235	435	A & W ROOT BEER	Soft Drinks

Fig: Clean Data in SSMS

8. Cleaning stg_wsdr:

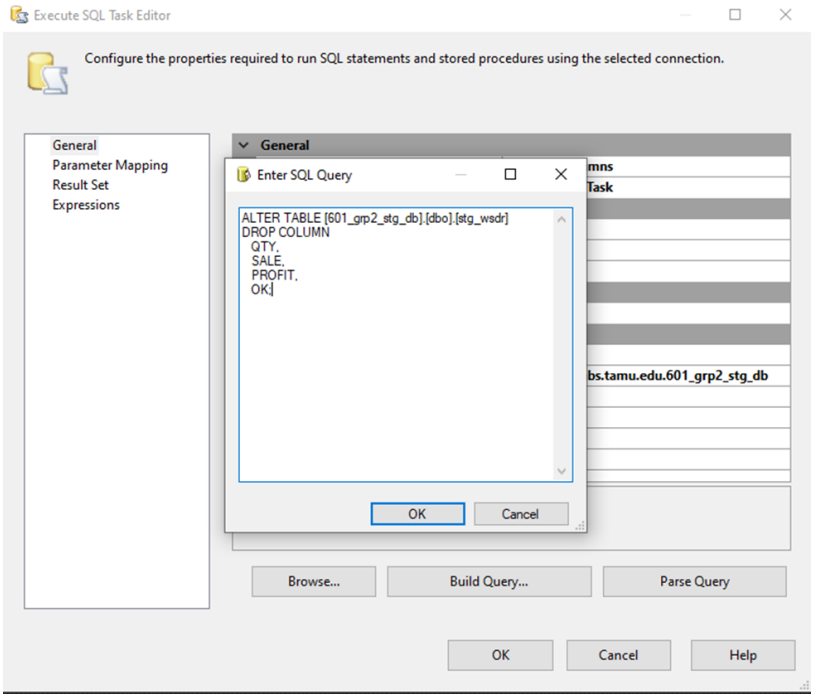


Fig: Dropping Columns not required

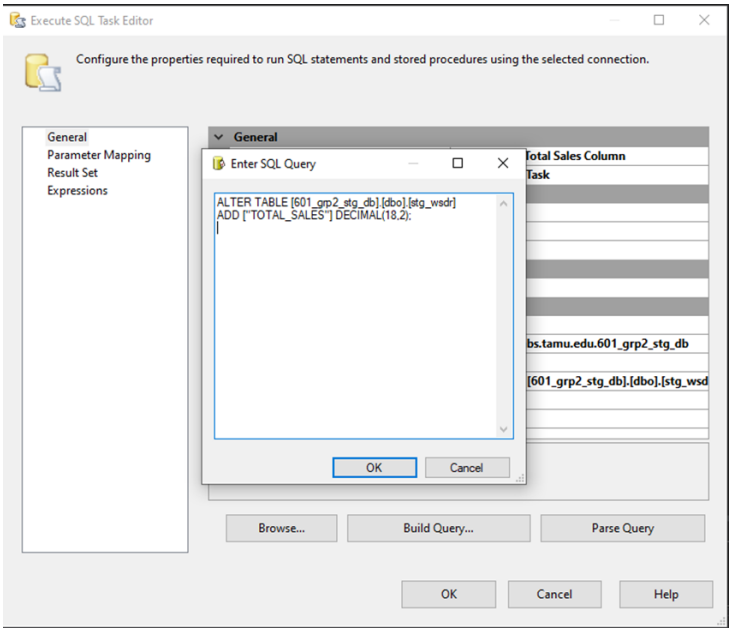


Fig: Adding New Column called Total_Sales

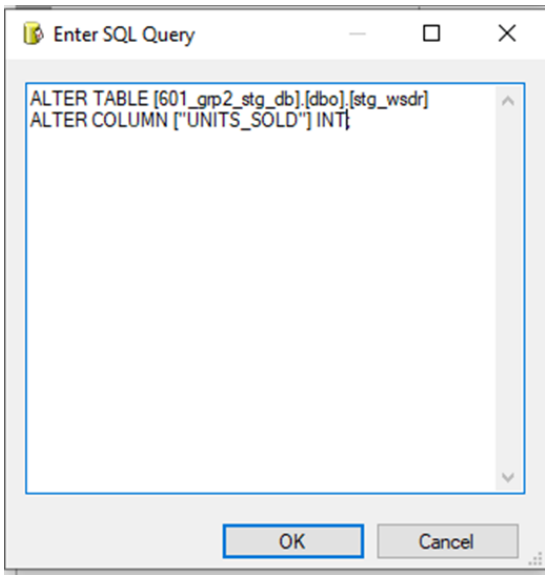


Fig: Altering data type of Units_Sold

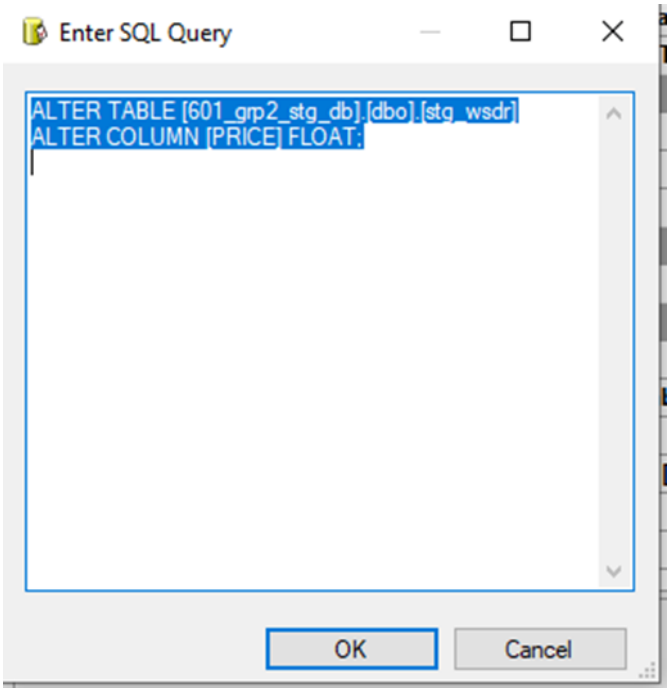


Fig: Altering data type of Price

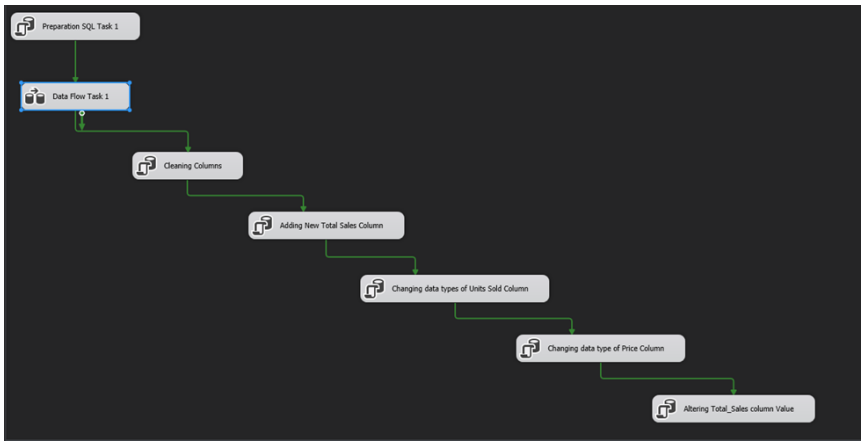
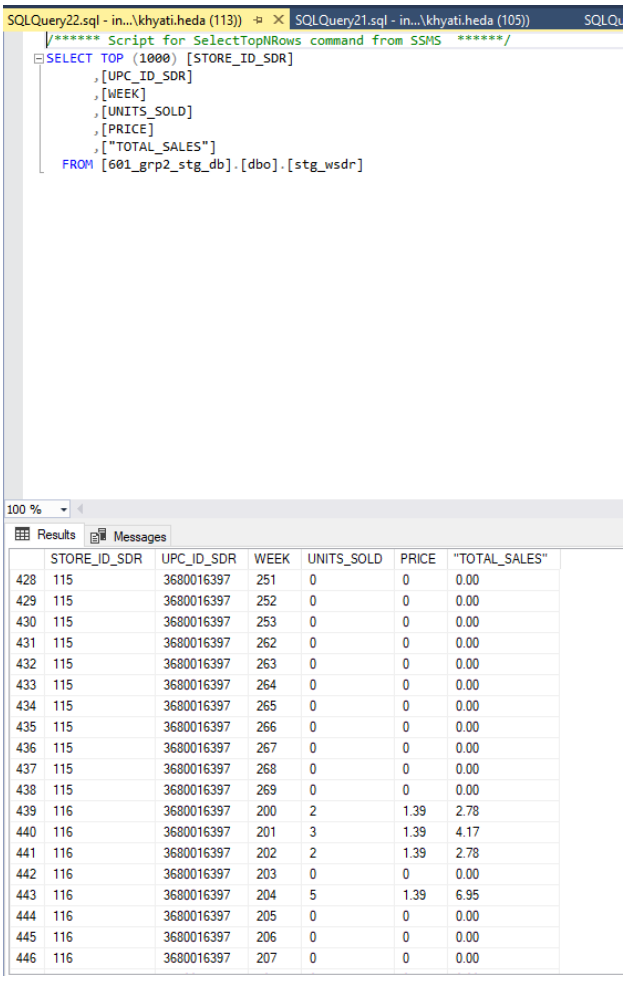


Fig: SQL Task in SSIS



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [STORE_ID_SDR]
      ,[UPC_ID_SDR]
      ,[WEEK]
      ,[UNITS_SOLD]
      ,[PRICE]
      ,["TOTAL_SALES"]
  FROM [601_grp2_stg_db].[dbo].[stg_wsdr]
```

	STORE_ID_SDR	UPC_ID_SDR	WEEK	UNITS_SOLD	PRICE	"TOTAL_SALES"
428	115	3680016397	251	0	0	0.00
429	115	3680016397	252	0	0	0.00
430	115	3680016397	253	0	0	0.00
431	115	3680016397	262	0	0	0.00
432	115	3680016397	263	0	0	0.00
433	115	3680016397	264	0	0	0.00
434	115	3680016397	265	0	0	0.00
435	115	3680016397	266	0	0	0.00
436	115	3680016397	267	0	0	0.00
437	115	3680016397	268	0	0	0.00
438	115	3680016397	269	0	0	0.00
439	116	3680016397	200	2	1.39	2.78
440	116	3680016397	201	3	1.39	4.17
441	116	3680016397	202	2	1.39	2.78
442	116	3680016397	203	0	0	0.00
443	116	3680016397	204	5	1.39	6.95
444	116	3680016397	205	0	0	0.00
445	116	3680016397	206	0	0	0.00
446	116	3680016397	207	0	0	0.00

Fig: Clean Data in SSMS

9. Creating new temp table: stg_SDR:

- Now we are creating a temp table called as stg_SDR which will join the stg_upcsdr and stg_wsdr

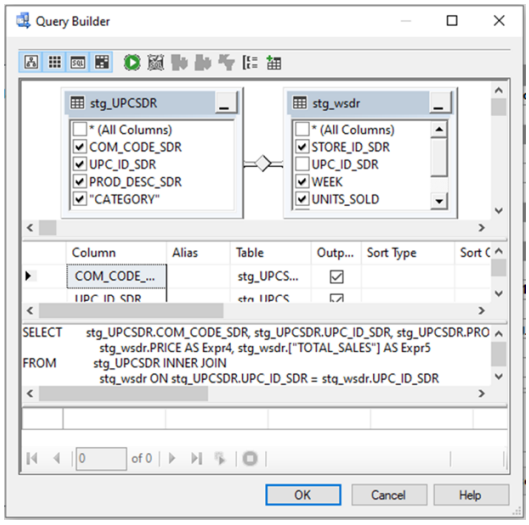
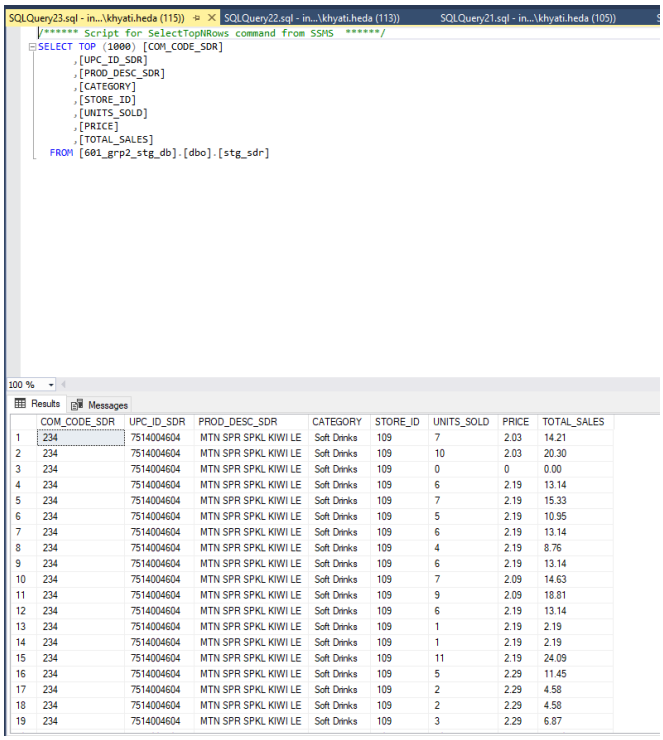


Fig: Joining both the tables and storing it into stg_SDR



	COM_CODE_SDR	UPC_ID_SDR	PROD_DESC_SDR	CATEGORY	STORE_ID	UNITS_SOLD	PRICE	TOTAL_SALES
1	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	7	2.03	14.21
2	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	10	2.03	20.30
3	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	0	0	0.00
4	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	6	2.19	13.14
5	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	7	2.19	15.33
6	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	5	2.19	10.95
7	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	6	2.19	13.14
8	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	4	2.19	8.76
9	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	6	2.19	13.14
10	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	7	2.09	14.63
11	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	9	2.09	18.81
12	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	6	2.19	13.14
13	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	1	2.19	2.19
14	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	1	2.19	2.19
15	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	11	2.19	24.09
16	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	5	2.29	11.45
17	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	2	2.29	4.58
18	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	2	2.29	4.58
19	234	7514004604	MTN SPR SPKL KIWI LE	Soft Drinks	109	3	2.29	6.87

Fig: Stg_SDR in SSMS

10. Creating new table: stg_movement:

This will join distinct stg_SDR and stg_FRD

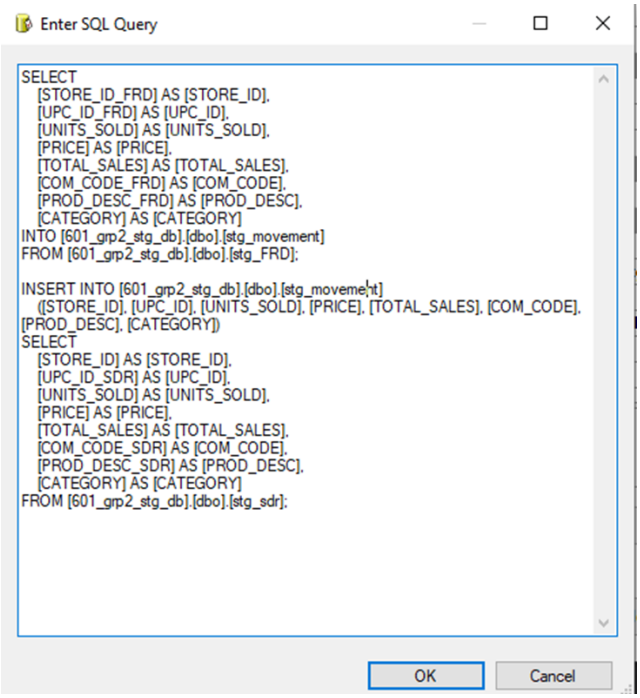
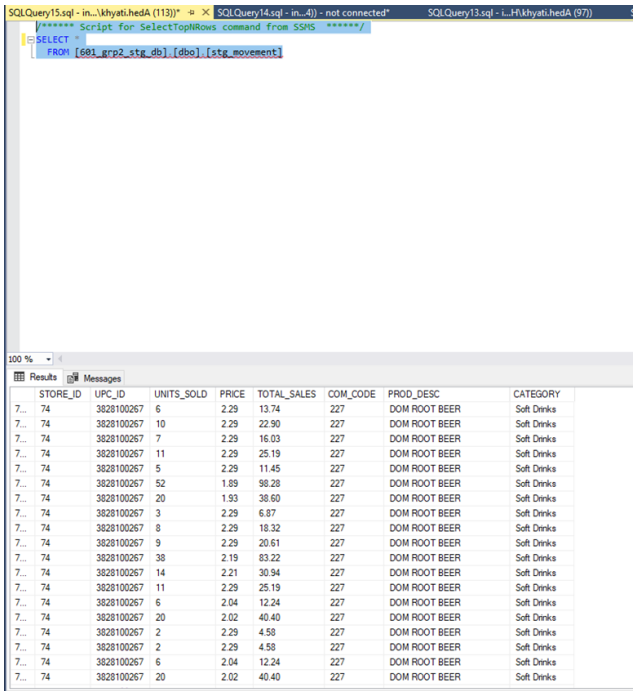


Fig: SQL Query to join stg_frd and stg_sdr into stg_movement



	STORE_ID	UPC_ID	UNITS_SOLD	PRICE	TOTAL_SALES	COM_CODE	PROD_DESC	CATEGORY
7...	74	3828100267	6	2.29	13.74	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	10	2.29	22.90	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	7	2.29	16.03	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	11	2.29	25.19	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	5	2.29	11.45	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	52	1.89	98.28	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	20	1.93	38.60	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	3	2.29	6.87	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	8	2.29	18.32	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	9	2.29	20.61	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	38	2.19	83.22	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	14	2.21	30.94	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	11	2.29	25.19	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	6	2.04	12.24	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	20	2.02	40.40	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	2	2.29	4.58	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	2	2.29	4.58	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	6	2.04	12.24	227	DOM ROOT BEER	Soft Drinks
7...	74	3828100267	20	2.02	40.40	227	DOM ROOT BEER	Soft Drinks

Fig: stg_MOVEMENT in SSMS

Data Loading into Independent Data marts

Data Mart 1 – Product Category Specific

Database: 601_grp2_DATAMART1

1. Time Dim:

```
CREATE TABLE [601_grp2_DATAMART1].[dbo].[time_dim](
TIME_ID INT IDENTITY(1,1) PRIMARY KEY,
[Year] varchar(2),
[Month] varchar(2),
[Day] varchar(2)
)
```

```
%
Messages
Commands completed successfully.

Completion time: 2024-11-15T19:10:54.9495604-06:00
```

Fig: Creating time_dim table

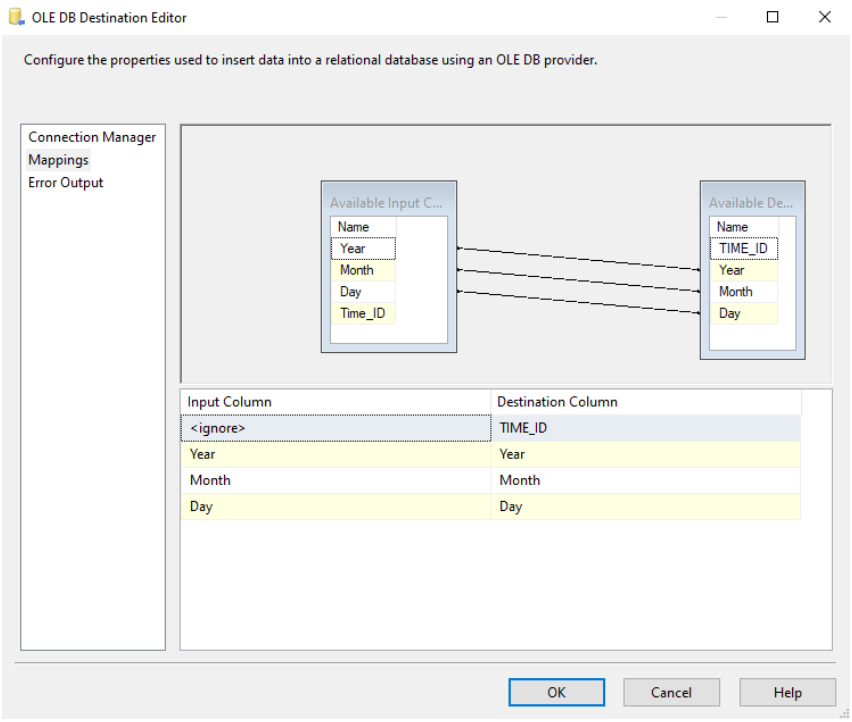


Fig: Mapping columns for time_dim table

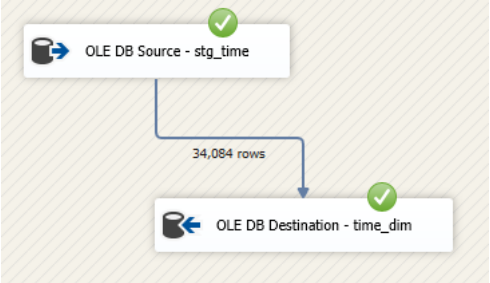


Fig: Data Flow Task in SSIS from stg_time to time_dim

```
SELECT TOP (1000) [TIME_ID]
      ,[Year]
      ,[Month]
      ,[Day]
  FROM [601_grp2_DATAMART1].[dbo].[time_dim]
```

Results | Messages

	TIME_ID	Year	Month	Day
1	1	95	04	16
2	2	95	04	17
3	3	95	04	18
4	4	95	04	19
5	5	95	04	20
6	6	95	04	21
7	7	95	04	22
8	8	95	04	23
9	9	95	04	24
10	10	95	04	25
11	11	95	04	26
12	12	95	04	27

Fig: Data loaded in time_dim

2. Fact Table – FACT_CATEGORY_SALES

```
CREATE TABLE FACT_CATEGORY_SALES (
    GROCERY_SUM VARCHAR(50),
    DAIRY_SUM VARCHAR(50),
    FROZEN_SUM VARCHAR(50),
    BOTTLE_SUM VARCHAR(50),
    MEAT_SUM VARCHAR(50),
    MEATFROZ_SUM VARCHAR(50),
    FISH_SUM VARCHAR(50),
    PROMO_SUM VARCHAR(50),
    PRODUCE_SUM VARCHAR(50),
    BULK_SUM VARCHAR(50),
    SALADBAR_SUM VARCHAR(50),
    FLORAL_SUM VARCHAR(50),
    DELI_SUM VARCHAR(50),
    CONVFOOD_SUM VARCHAR(50),
    CHEESE_SUM VARCHAR(50),
    BAKERY_SUM VARCHAR(50),
    PHARMACY_SUM VARCHAR(50),
    GM_SUM VARCHAR(50),
    JEWELRY_SUM VARCHAR(50),
    COSMETIC_SUM VARCHAR(50),
    HABA_SUM VARCHAR(50),
    CAMERA_SUM VARCHAR(50),
    VIDEO_SUM VARCHAR(50),
    BEER_SUM VARCHAR(50),
    WINE_SUM VARCHAR(50),
    SPIRITS_SUM VARCHAR(50),
    FTGCHIN_SUM VARCHAR(50),
    FTGITAL_SUM VARCHAR(50),
    TIME_ID INT,
    FOREIGN KEY (TIME_ID) REFERENCES [601_grp2_DATAMART1].[dbo].[time_dim] (time_id)
);
```

```
%
Messages
Commands completed successfully.

Completion time: 2024-11-12T17:46:54.7239526-06:00
```

Fig: Creating the fact table

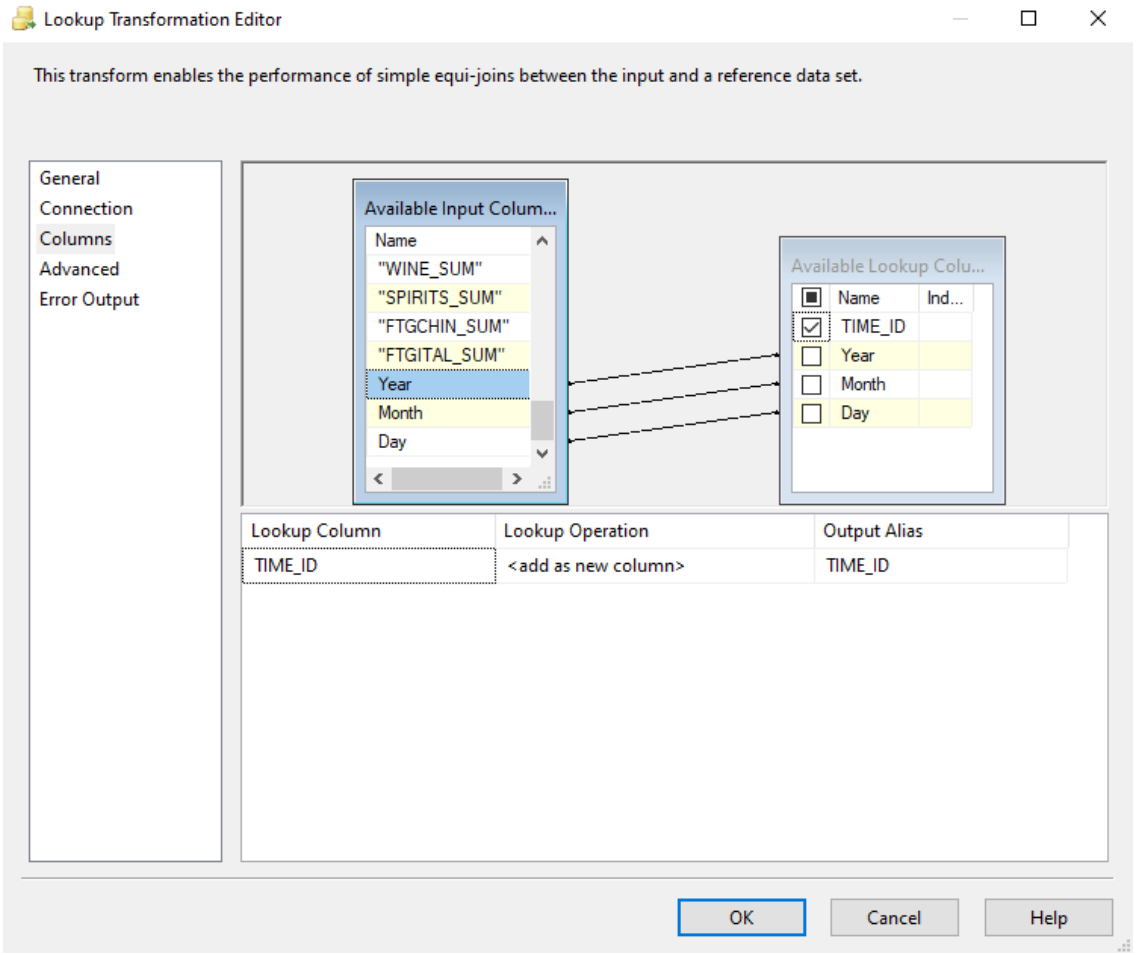


Fig: Performing Lookup transformation for time_dim

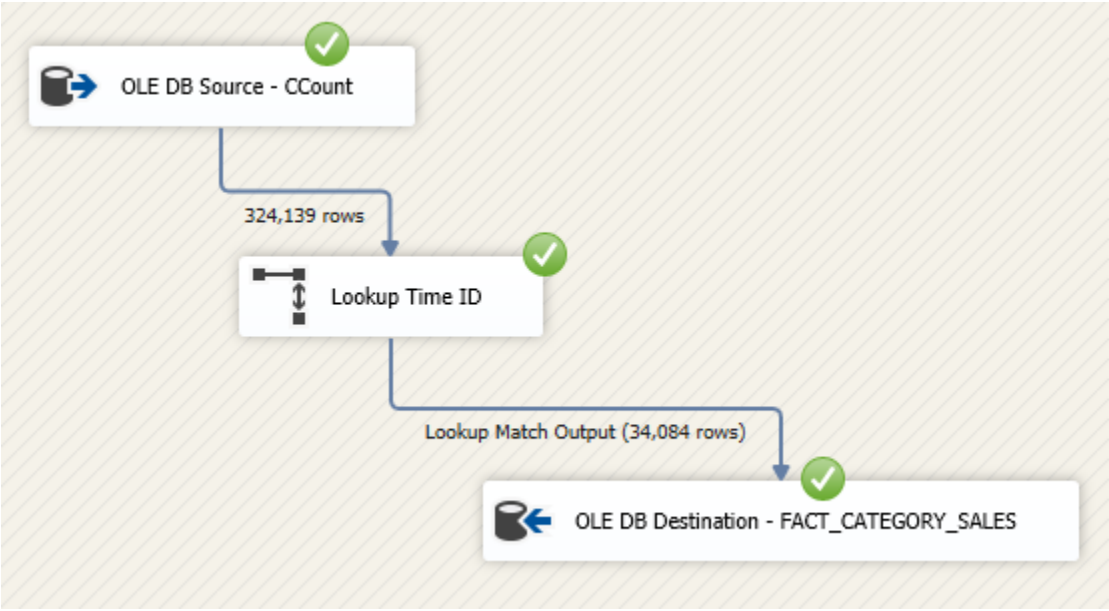


Fig: Lookup transformation data flow

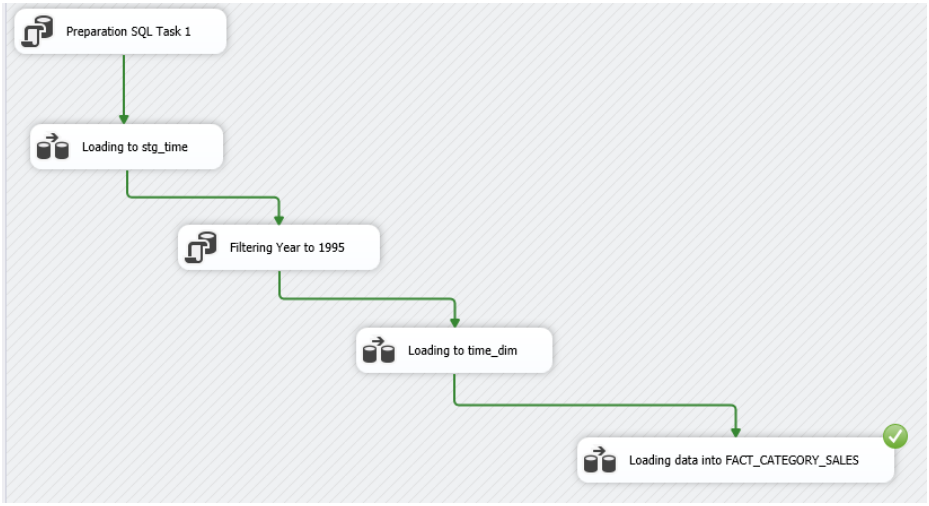


Fig: SQL tasks in SSIS successfully executed

```
SELECT TOP (1000) [GROCERY_SUM]
      ,[DAIRY_SUM]
      ,[FROZEN_SUM]
      ,[BOTTLE_SUM]
      ,[MEAT_SUM]
      ,[MEATFROZ_SUM]
      ,[FISH_SUM]
      ,[PROMO_SUM]
      ,[PRODUCE_SUM]
      ,[BULK_SUM]
      ,[SALADBAR_SUM]
      ,[FLORAL_SUM]
      ,[DELI_SUM]
      ,[CONVFOOD_SUM]
      ,[CHEESE_SUM]
      ,[BAKERY_SUM]
      ,[PHARMACY_SUM]
      ,[GM_SUM]
      ,[JEWELRY_SUM]
      ,[COSMETIC_SUM]
      ,[HABA_SUM]
      ,[CAMERA_SUM]
      ,[VIDEO_SUM]
      ,[BEER_SUM]
      ,[WINE_SUM]
      ,[SPIRITS_SUM]
      ,[FTGCHIN_SUM]
      ,[FTGITAL_SUM]
      ,[TIME_ID]
  FROM [601_grp2_DATAMART1].[dbo].[FACT_CATEGORY_SALES]
```

Results | Messages

	GROCERY_SUM	DAIRY_SUM	FROZEN_SUM	BOTTLE_SUM	MEAT_SUM	MEATFROZ_SUM	FISH_SUM	PROMO_SUM	PRODUCE_SUM	BULK_SUM
	10917.36	2971.97	1880.57	0	1875.97	192.31	383.32	17.97	3392.09	324.980000000000002
2	16641.04	4788.41	2749.1	0	3049.77	358.56	331.4	49.99	5367.54	316.079999999999998
3	18574.18	5219.21	3514.09	0	3421.89	463.95	499.51	0	5520.73	433.740000000000001
4	17343.11	4729.72	3221.03	0	3217.18	486.92	555.96	29.99	4738.74	415.88
5	19897.43	4569.59	3775.91	0	5216.77	552.52	652.11	69.99	6483.72	665.240000000000001
6	23937.46	5542.99	4160.97	0	5940.2	612.51	1129.73	2.99	7829.53	657.649999999999998
7	32874.57	7362.44	6068.85	0	10003.04	777.81	1387.01	127.95	10871.44	945.470000000000003
8	32870.12	7342.59	6158.29	0	8195.94	872.61	690.2	314.94	9142.76	761.360000000000001
9	18513.84	4387.15	3574.1	0	3613.04	598.51	410.18	0	6066.62	423.660000000000003
10	17473.2	3992.66	2999.1	0	4348.31	512.5	406.01	8.97	5234.78	475.720000000000003
11	17684.37	4218.97	3067.78	0	4266.21	339.18	461.92	0	5202.95	459.540000000000002
12	23633.54	5737.47	4009.14	0	5046.23	537.42	739.67	129.95	6922.72	494.949999999999999
13	25996.14	6287.04	4420.81	0	5611.31	580.46	1096	242.87	8226.62	476.94
14	35963.14	8311.58	6370.75	0	8036.73	846.43	1173.47	361.88	10514.68	815.190000000000005

Fig: FACT_CATEGORY_SALES data in SSMS

Data Mart 2 –Store Specific
Database: 601_grp2_DATAMART2

1. Demo Dim:

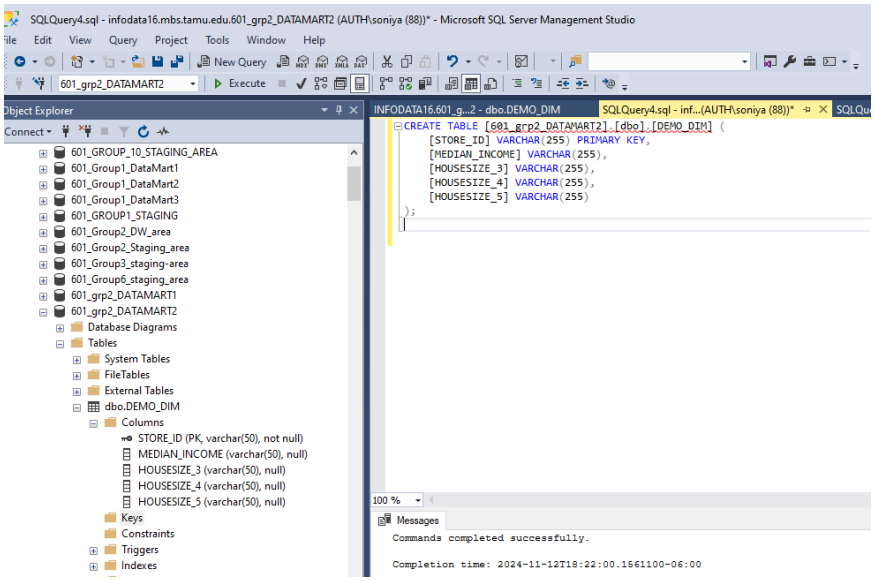


Fig: Creating DEMO_DIM table

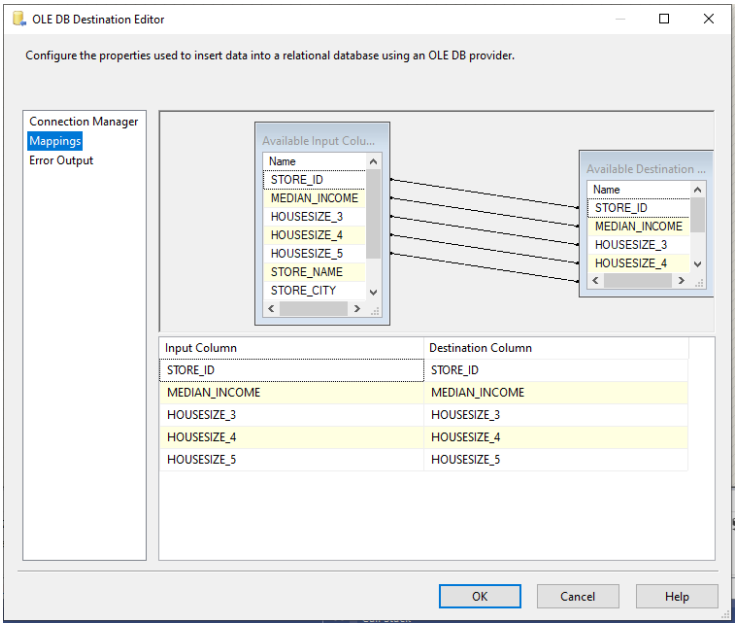


Fig: Mapping columns for the DEMO_DIM table

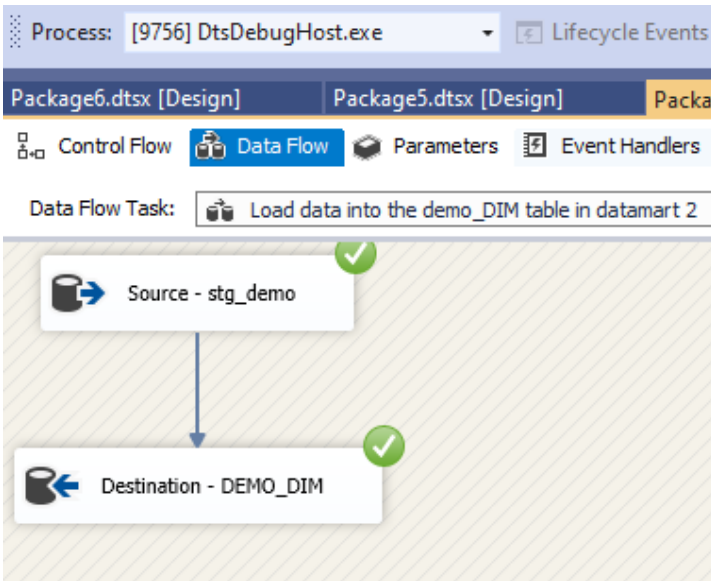
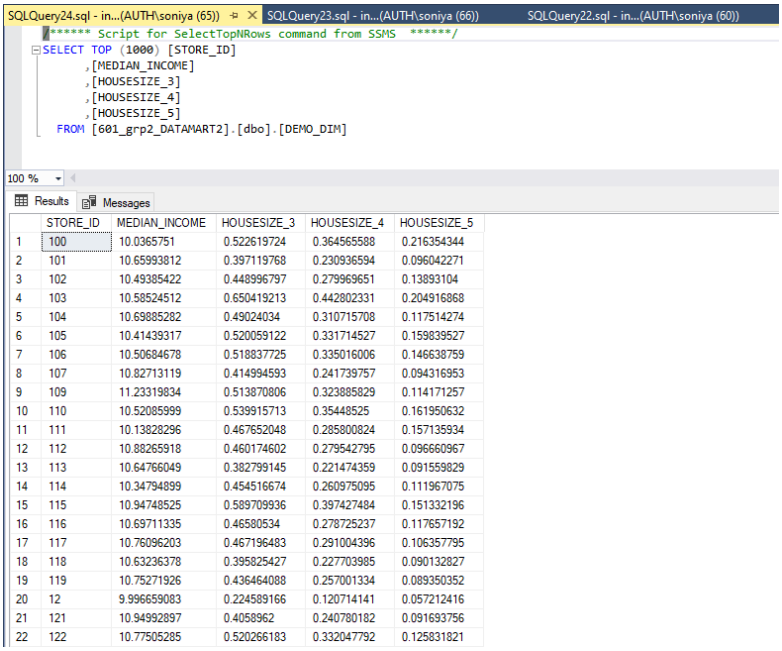


Fig: Data Flow task in SSIS



```
SQLQuery24.sql - in...(AUTH\soniya (65))   SQLQuery23.sql - in...(AUTH\soniya (66))   SQLQuery22.sql - in...(AUTH\soniya (60))
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [STORE_ID]
      ,[MEDIAN_INCOME]
      ,[HOUSESIZE_3]
      ,[HOUSESIZE_4]
      ,[HOUSESIZE_5]
  FROM [601_grp2_DATAMART2].[dbo].[DEMO_DIM]
```

100 %

Results | Messages

	STORE_ID	MEDIAN_INCOME	HOUSESIZE_3	HOUSESIZE_4	HOUSESIZE_5
1	100	10.0365751	0.522619724	0.364565588	0.216354344
2	101	10.65993812	0.397119768	0.230936594	0.096042271
3	102	10.49385422	0.448996797	0.279969651	0.13893104
4	103	10.58524512	0.650419213	0.442802331	0.204916868
5	104	10.69885282	0.49024034	0.310715708	0.117514274
6	105	10.41439317	0.520059122	0.331714527	0.159839527
7	106	10.50684678	0.518837725	0.335016006	0.146638759
8	107	10.82713119	0.414994593	0.241739757	0.094316953
9	109	11.23319834	0.513870806	0.323885829	0.114171257
10	110	10.52085999	0.539915713	0.35448525	0.161950632
11	111	10.13828296	0.467652048	0.285800824	0.157135934
12	112	10.88265918	0.460174602	0.279542795	0.096660967
13	113	10.64766049	0.382799145	0.221474359	0.091559829
14	114	10.34794899	0.454516674	0.260975095	0.111967075
15	115	10.94748525	0.589709936	0.397427484	0.151332196
16	116	10.69711335	0.46580534	0.278725237	0.117657192
17	117	10.76096203	0.467196483	0.291004396	0.106357795
18	118	10.63236378	0.395825427	0.227703985	0.090132827
19	119	10.75271926	0.436464088	0.257001334	0.089350352
20	12	9.996659083	0.224589166	0.120714141	0.057212416
21	121	10.94992897	0.4058962	0.240780182	0.091693756
22	122	10.77505285	0.520266183	0.332047792	0.125831821

Fig: Data loaded in DEMO_DIM – SSMS

2. Store Dim

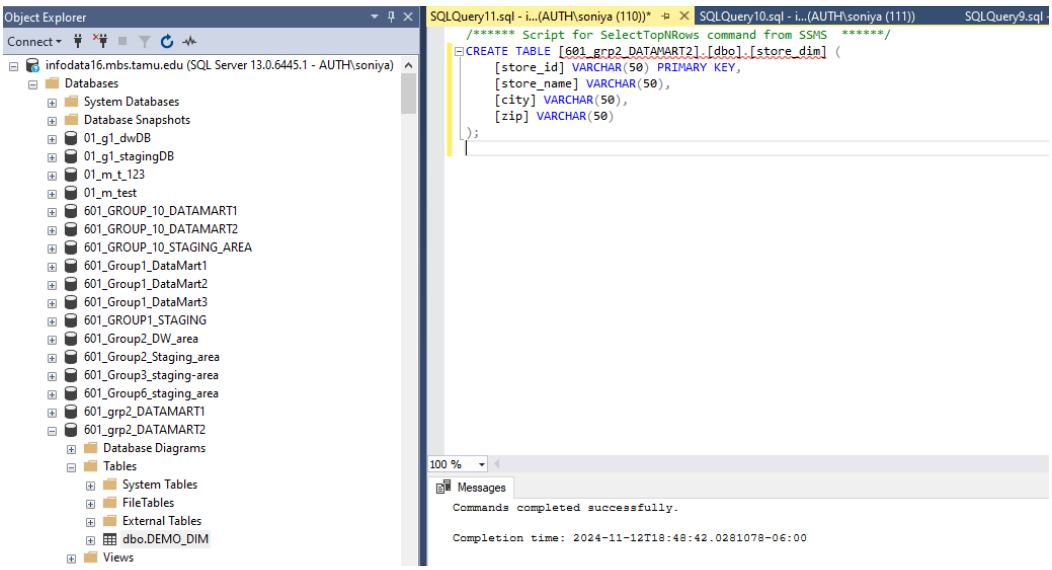


Fig: Create table Store_dim

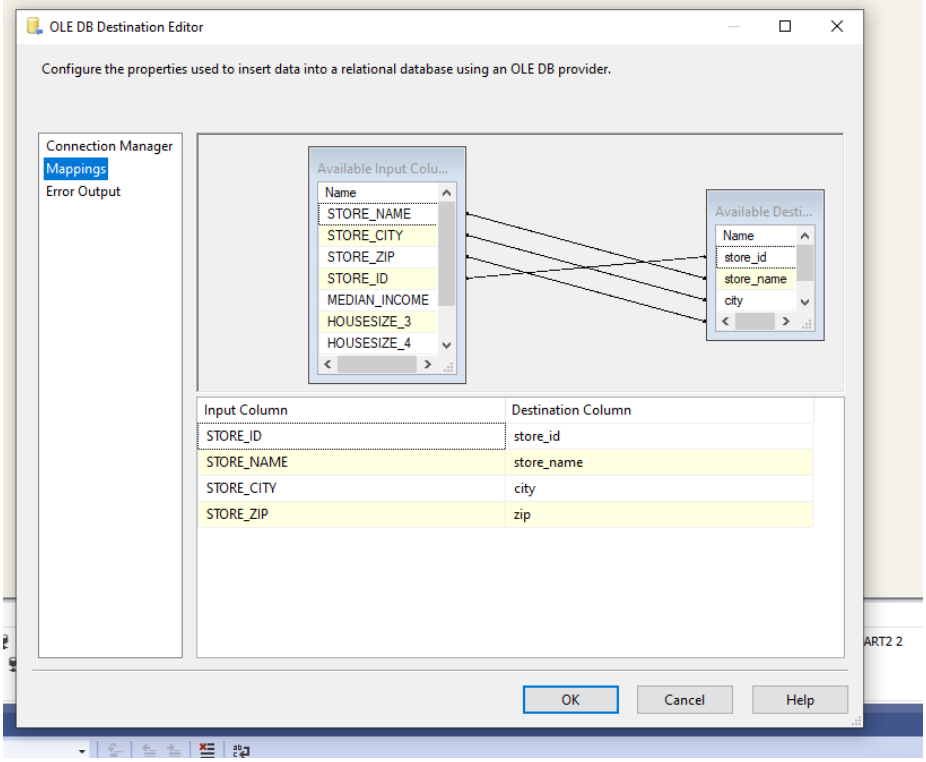


Fig: Mapping for store_dim

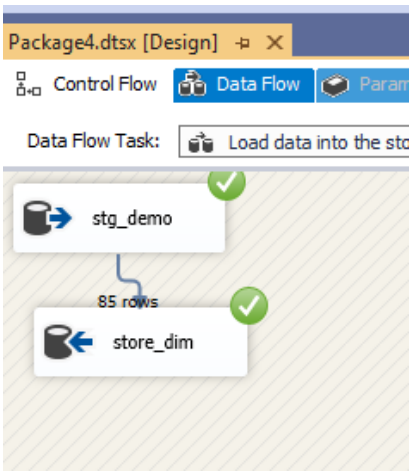


Fig: Data Flow task in SSIS

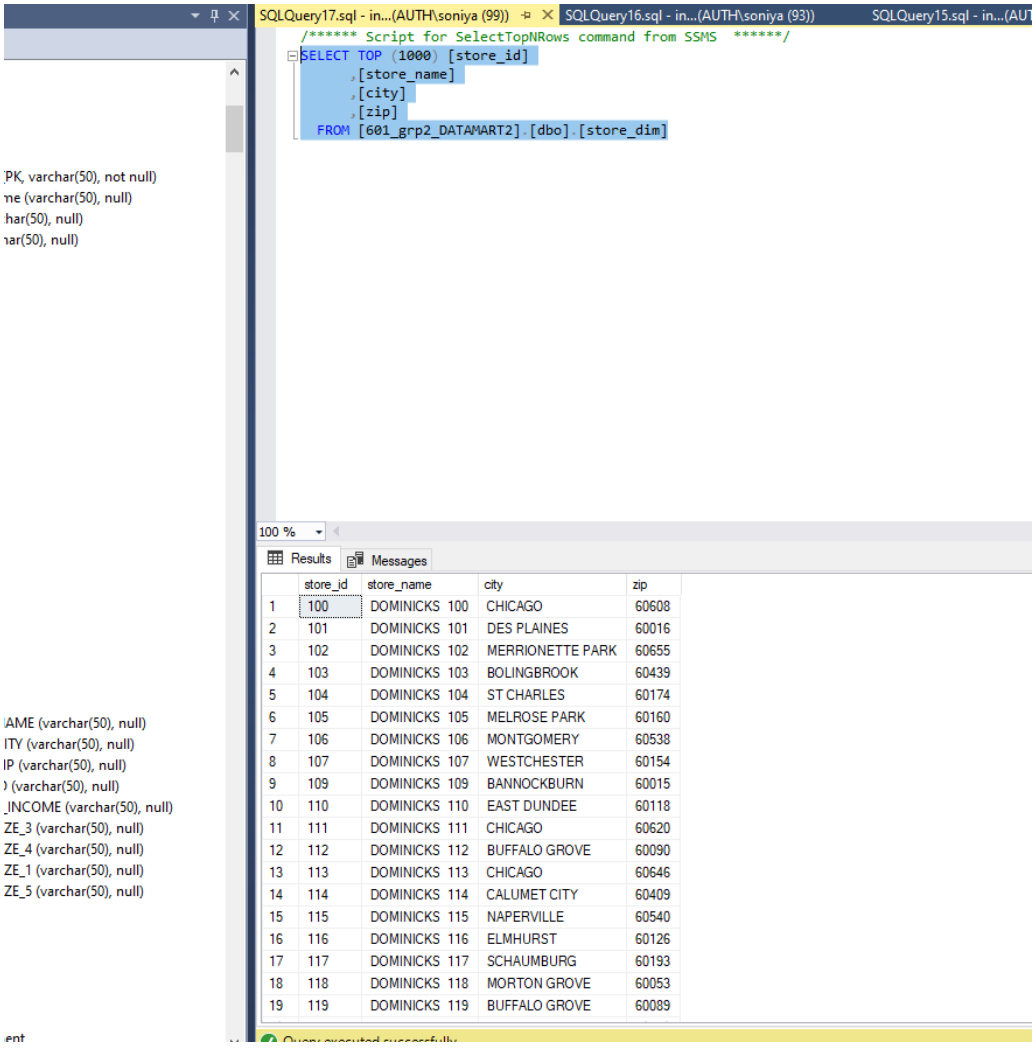


Fig: Data in store_dim in SSMS

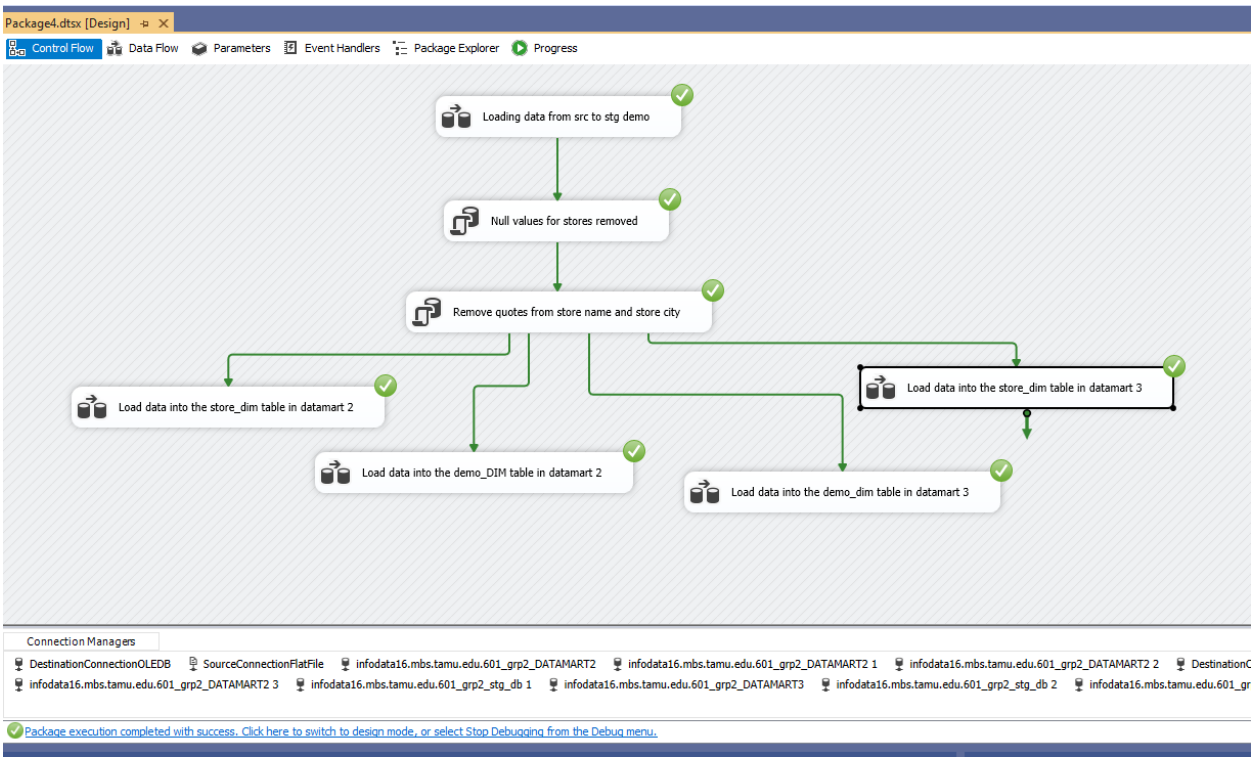


Fig: Data loaded in dim_demo and store_dim

3. Fact Table – Store_Sales

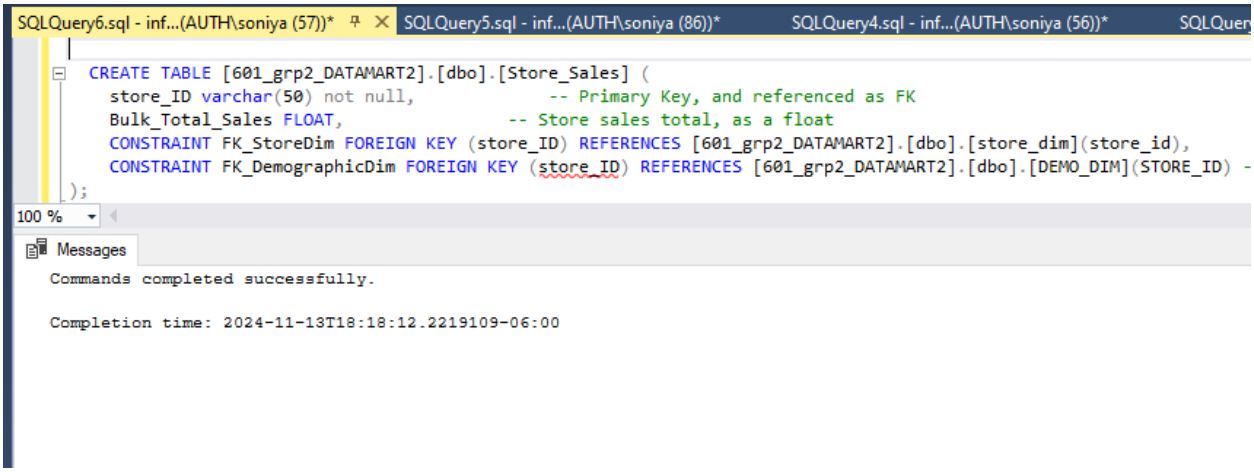


Fig: Create Fact table Store_Sales

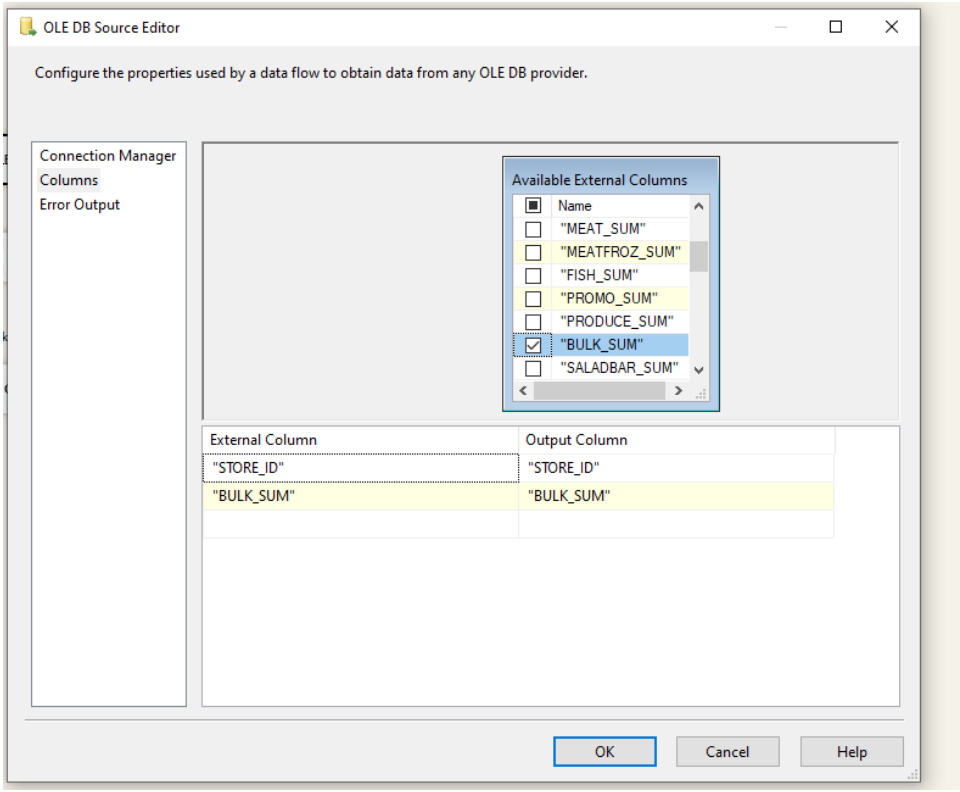


Fig: Setting the Source CCOUNT

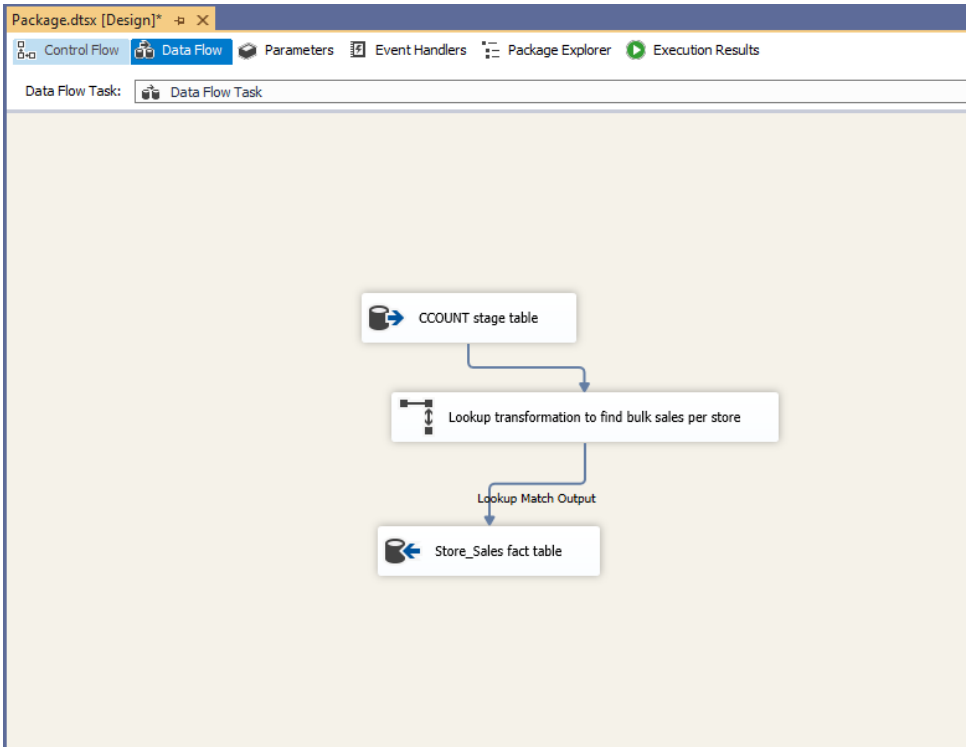
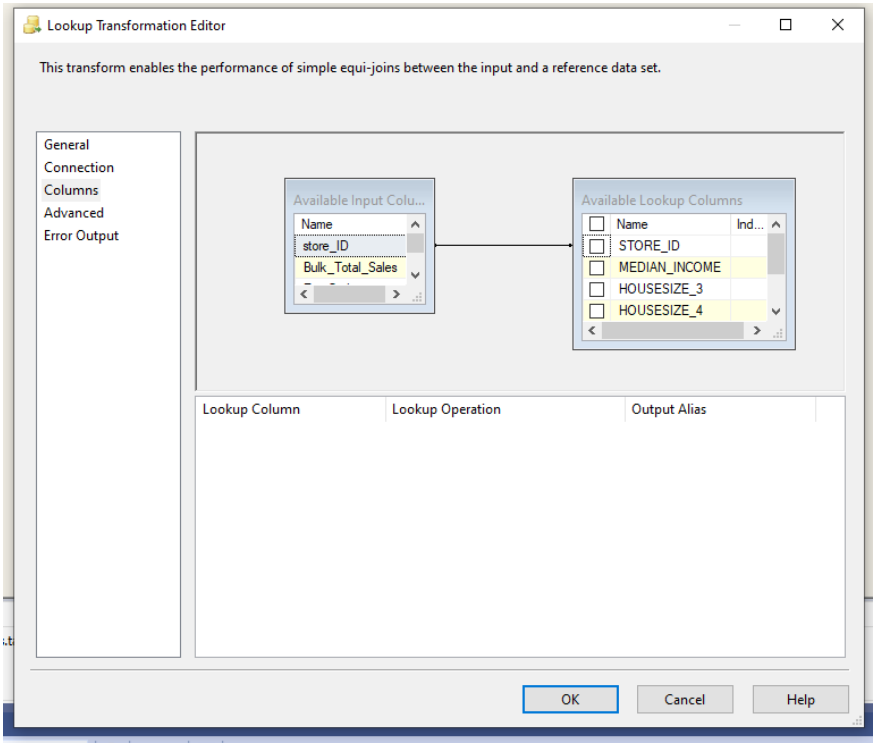


Fig: Data Flow Task in SSIS



Lookup transformation on Store_ID (Demo_dim)

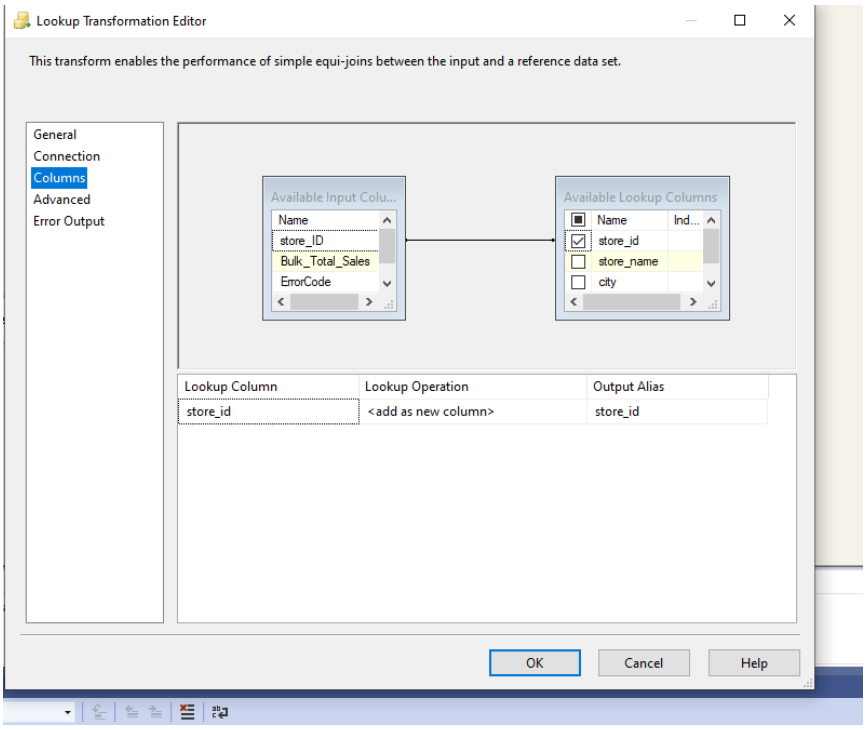


Fig: Lookup transformation on store_id (store_dim)

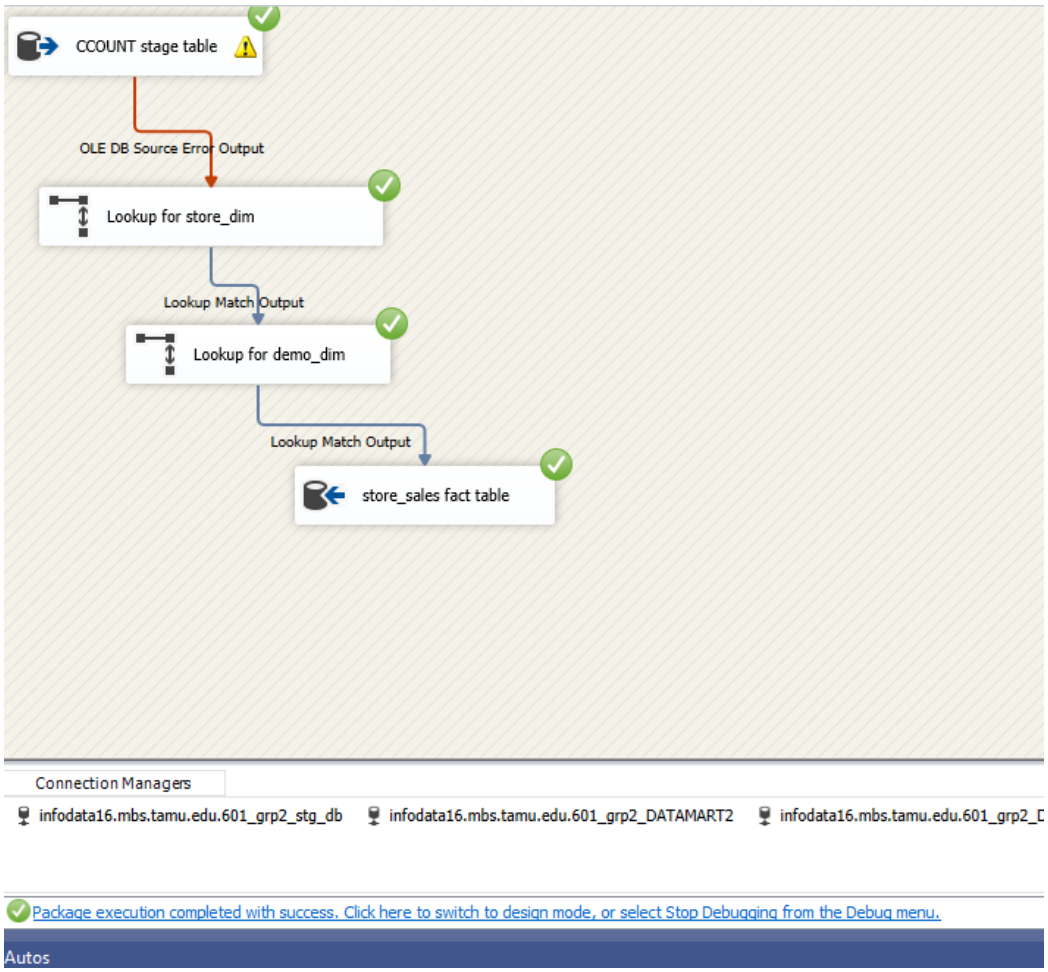


Fig: Data Flow tasks in SSIS

SQLQuery6.sql - inf...(AUTH\soniya (57))* SQLQuery13.sql - in...(AUTH\soniya (

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [store_ID]
      ,[Bulk_Total_Sales]
  FROM [601_grp2_DATAMART2].[dbo].[Store_Sales]
```

100 %

Results Messages

	store_ID	Bulk_Total_Sales
1	102	4172533.81
2	116	2549918.33
3	133	1473987.31
4	16	0
5	306	3057769.01
6	316	893613.279999999
7	44	2399007.66
8	45	978547.28
9	5	2892369.13
10	65	403082.49
11	71	2102623.47
12	72	988764.55
13	89	1253452.23
14	91	1698714.88
15	92	2112668.14
16	1	657.97
17	10	0
18	100	3326345.95
19	124	3096226.07
20	311	1366572.51
21	39	276073.24
22	49	610211.83
23	50	981863.919999999
24	59	2323421.5
25	62	770042.09
26	69	70061.58
27	74	3344568.01
28	88	2010013.5
29	90	2085018.05
30	114	2634598.13
31	12	2719576.21
32	135	291865.04
33	136	894015.170000001
34	141	126509.4
35	146	100505.38
36	18	2764367.79

Fig: Data in Store_Sales fact table in SSMS

Data Mart 3 – Movement Specific
Database: 601_grp2_DATAMART3

1. Prod Dim:

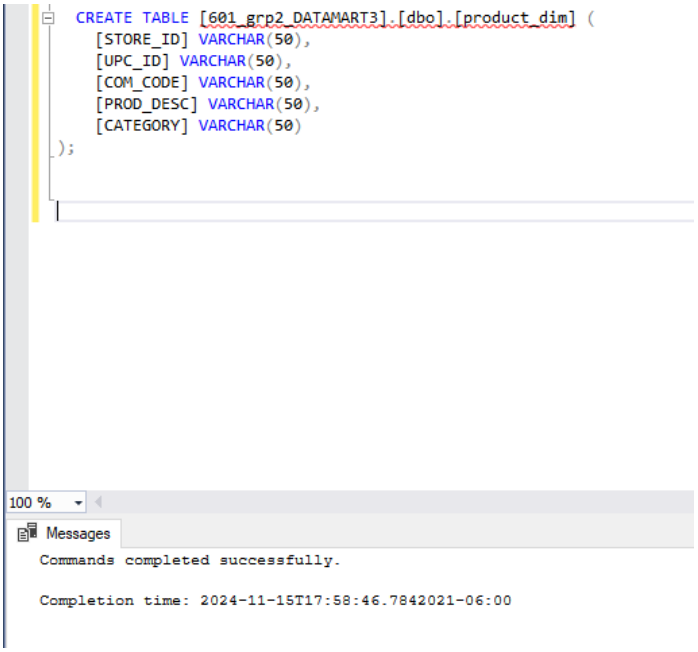


Fig: Creating table- Prod_dim

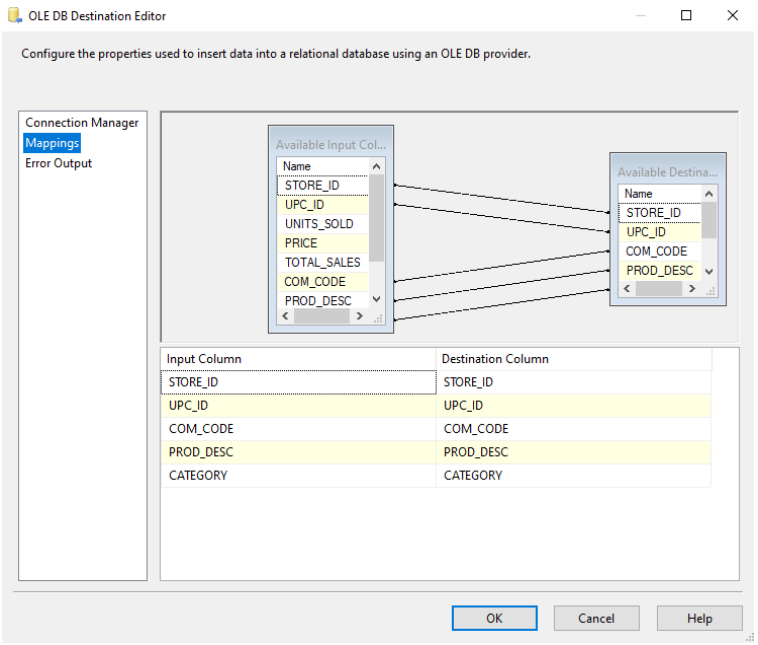


Fig: Mapping columns for the product_dim table

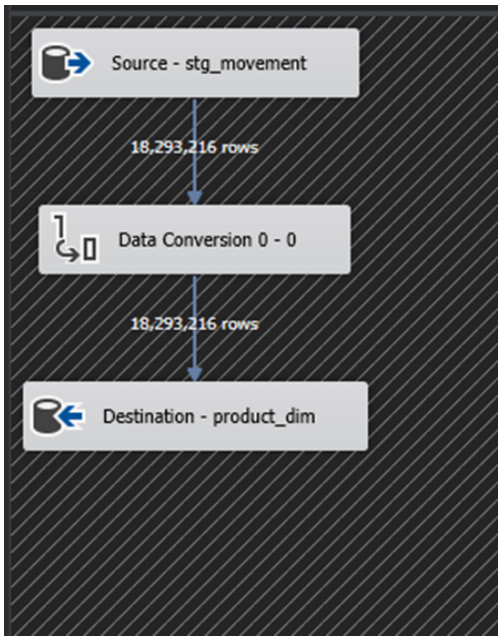
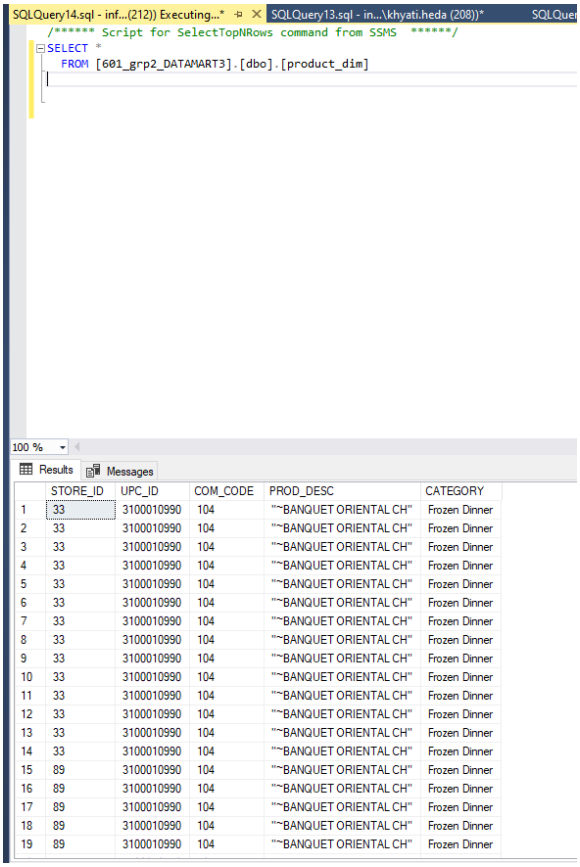


Fig: Data Flow Task in SSIS from stg_movement to product_dim



	STORE_ID	UPC_ID	COM_CODE	PROD_DESC	CATEGORY
1	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
2	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
3	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
4	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
5	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
6	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
7	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
8	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
9	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
10	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
11	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
12	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
13	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
14	33	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
15	89	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
16	89	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
17	89	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
18	89	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner
19	89	3100010990	104	""~BANQUET ORIENTAL CH""	Frozen Dinner

Fig: product_dim in SSMS

2. Demo dim

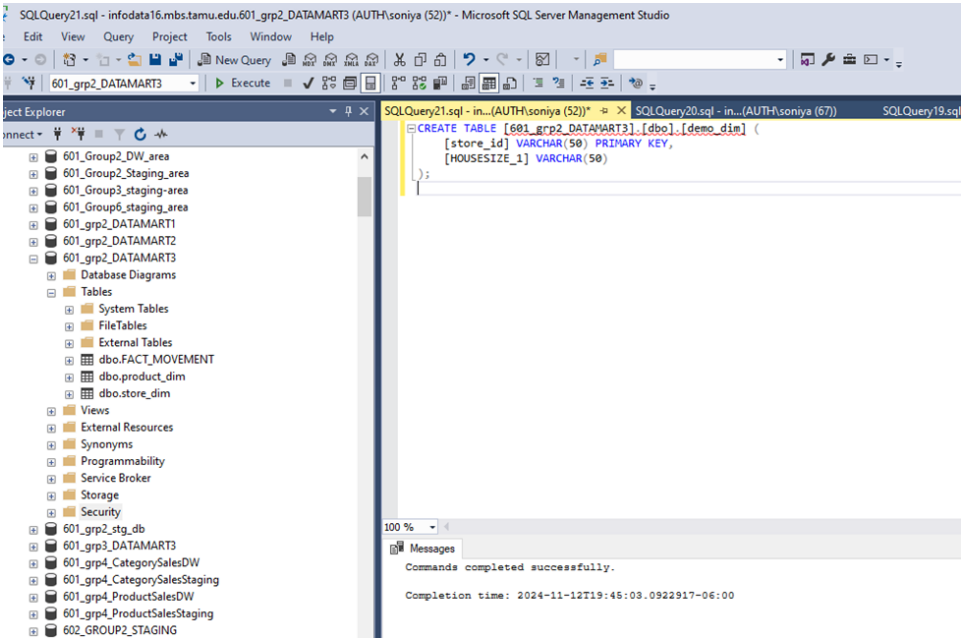


Fig: Creating table- demo_dim

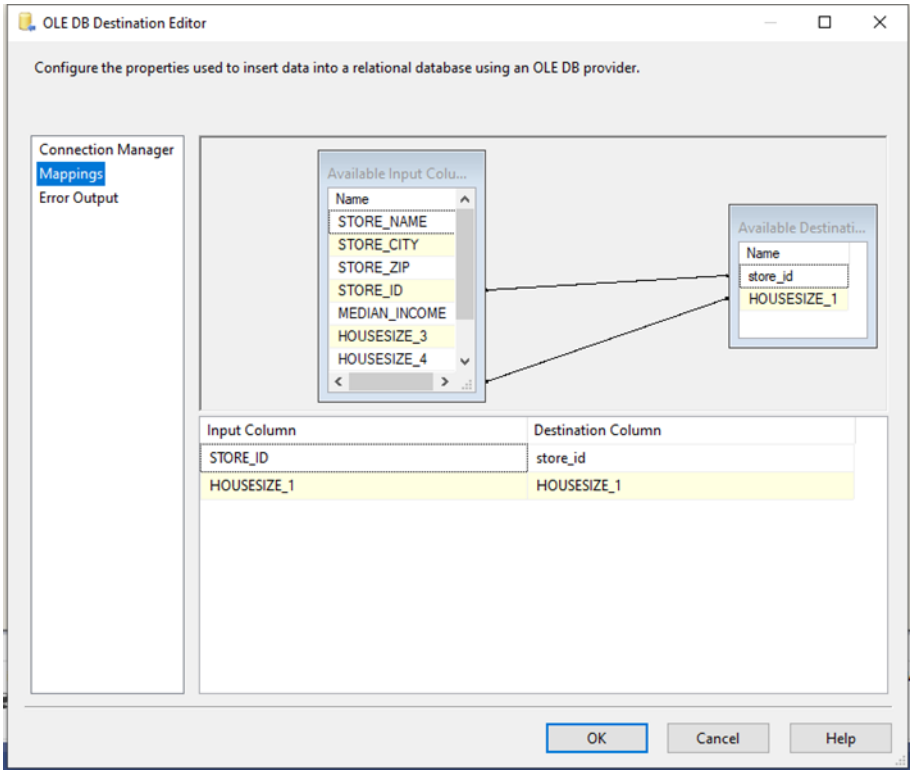


Fig: Mapping columns for the demo_dim table

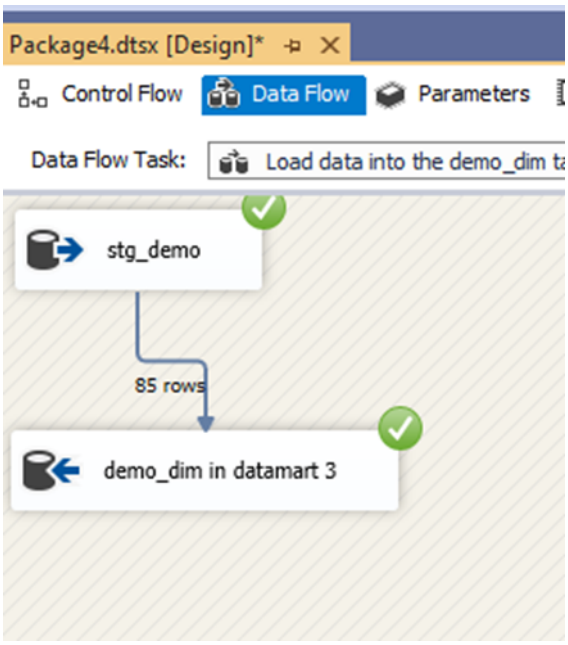


Fig: Data Flow Task in SSIS from stg_demo to demo_dim

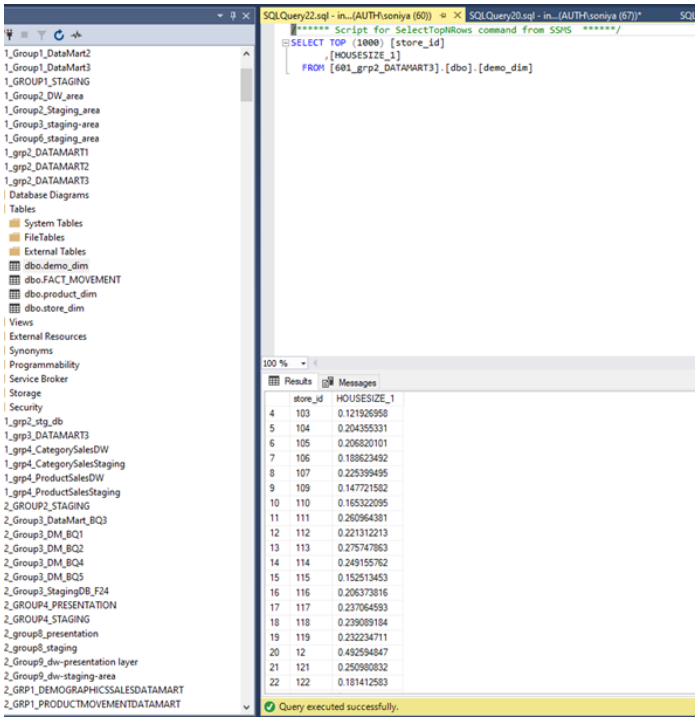


Fig: Demo_dim in SSMS

3. Store Dim:

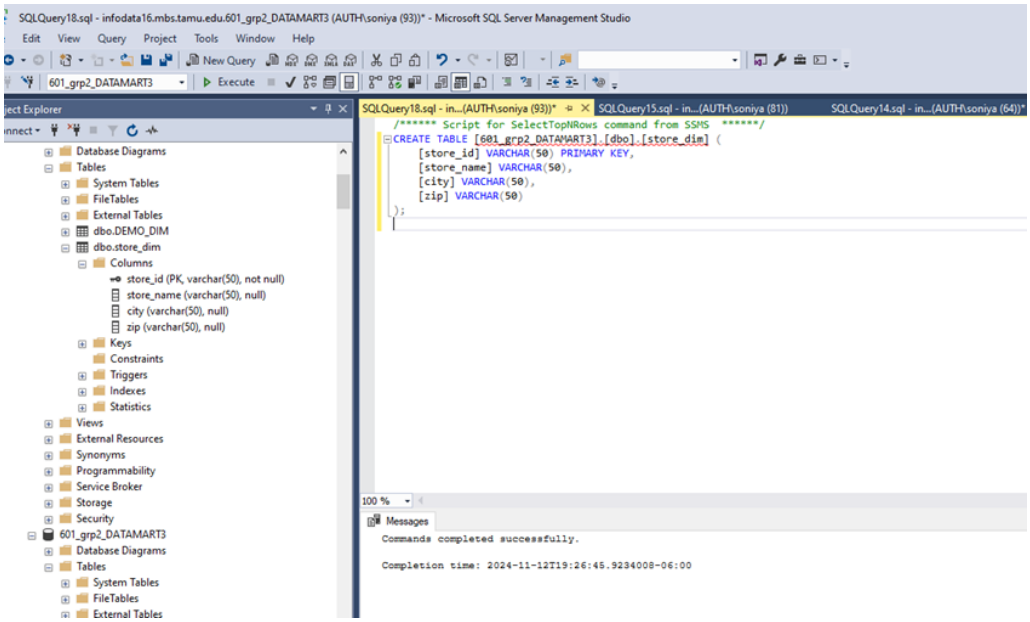


Fig: Creating table- store_dim

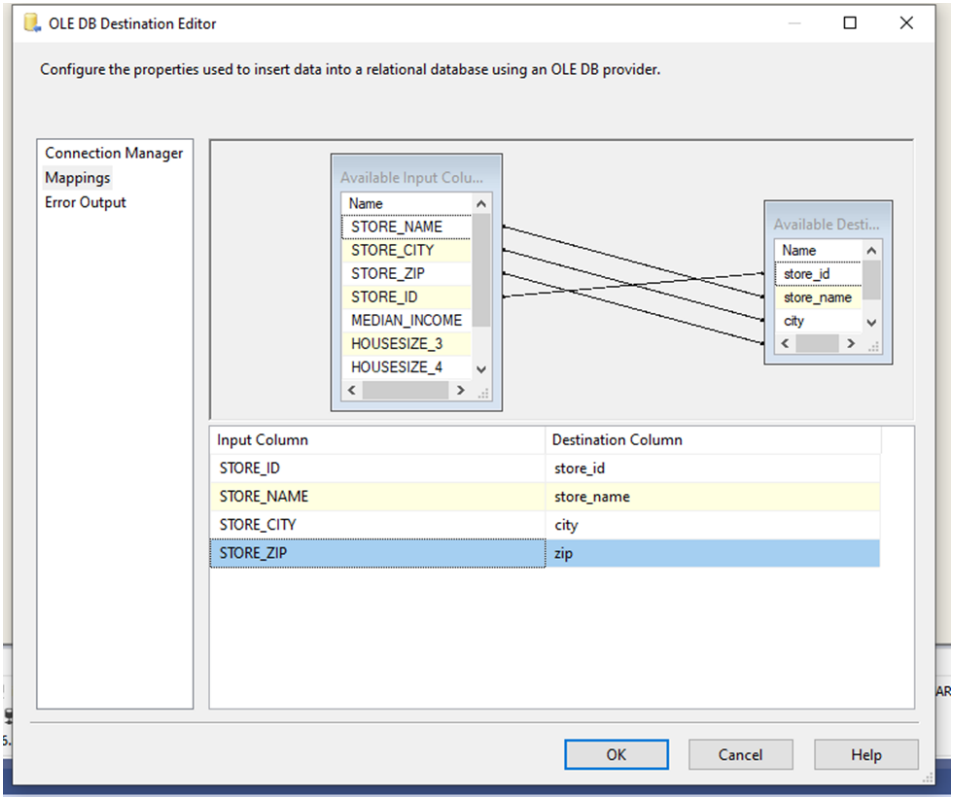


Fig: Mapping columns for the store_dim table

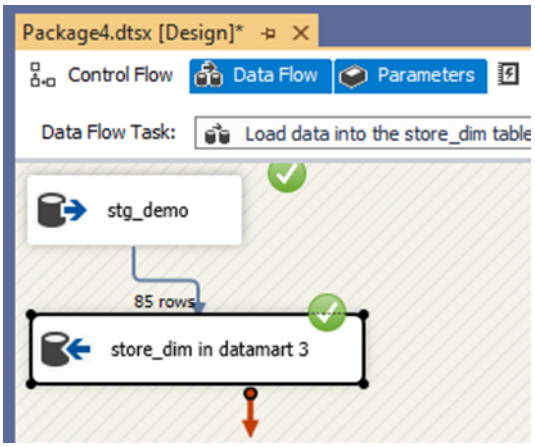
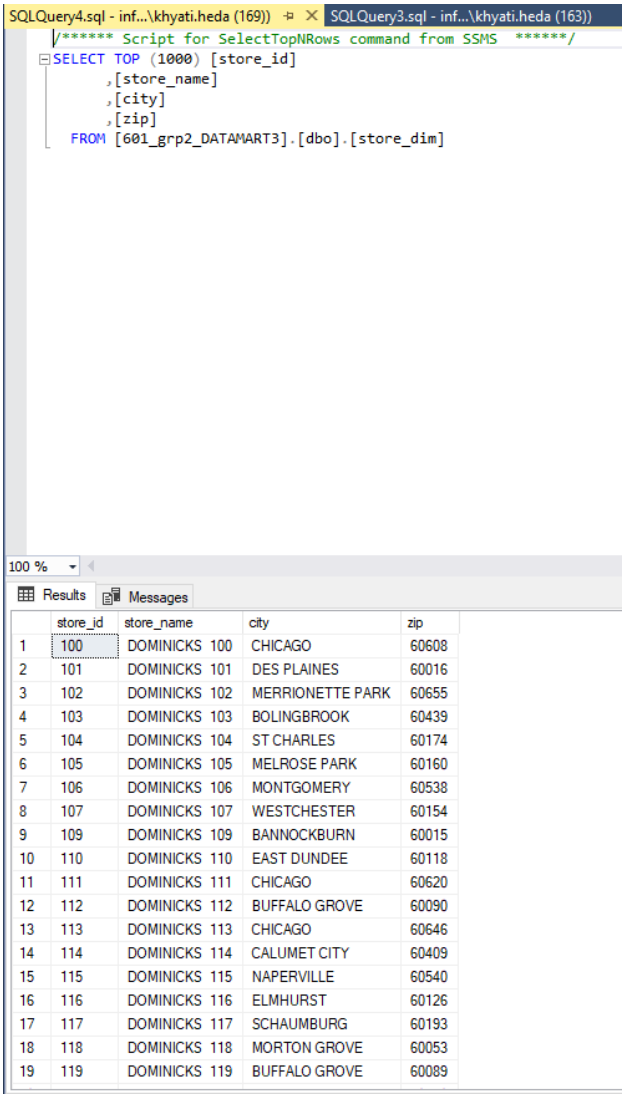


Fig: Data Flow Task in SSIS from stg_demo to store_dim



	store_id	store_name	city	zip
1	100	DOMINICKS 100	CHICAGO	60608
2	101	DOMINICKS 101	DES PLAINES	60016
3	102	DOMINICKS 102	MERRIONETTE PARK	60655
4	103	DOMINICKS 103	BOLINGBROOK	60439
5	104	DOMINICKS 104	ST CHARLES	60174
6	105	DOMINICKS 105	MELROSE PARK	60160
7	106	DOMINICKS 106	MONTGOMERY	60538
8	107	DOMINICKS 107	WESTCHESTER	60154
9	109	DOMINICKS 109	BANNOCKBURN	60015
10	110	DOMINICKS 110	EAST DUNDEE	60118
11	111	DOMINICKS 111	CHICAGO	60620
12	112	DOMINICKS 112	BUFFALO GROVE	60090
13	113	DOMINICKS 113	CHICAGO	60646
14	114	DOMINICKS 114	CALUMET CITY	60409
15	115	DOMINICKS 115	NAPERVILLE	60540
16	116	DOMINICKS 116	ELMHURST	60126
17	117	DOMINICKS 117	SCHAUMBURG	60193
18	118	DOMINICKS 118	MORTON GROVE	60053
19	119	DOMINICKS 119	BUFFALO GROVE	60089

Fig: store_dim in SSMS

4. Fact Table: fact_movement

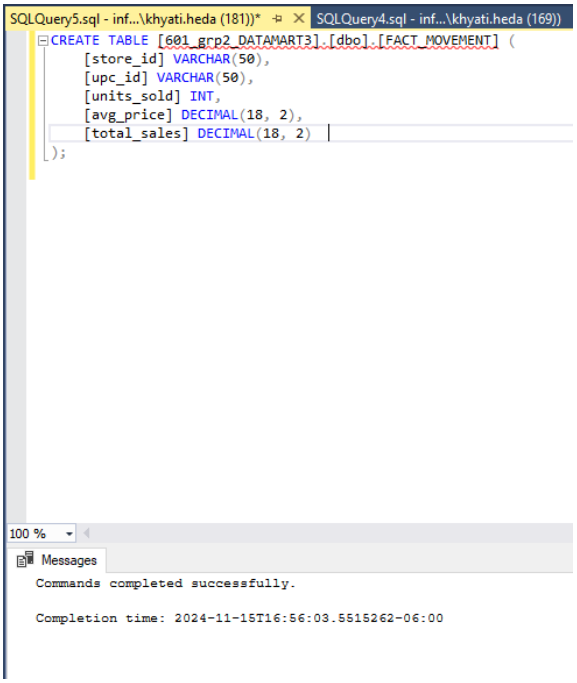


Fig: Creating Fact Table: fact_movement

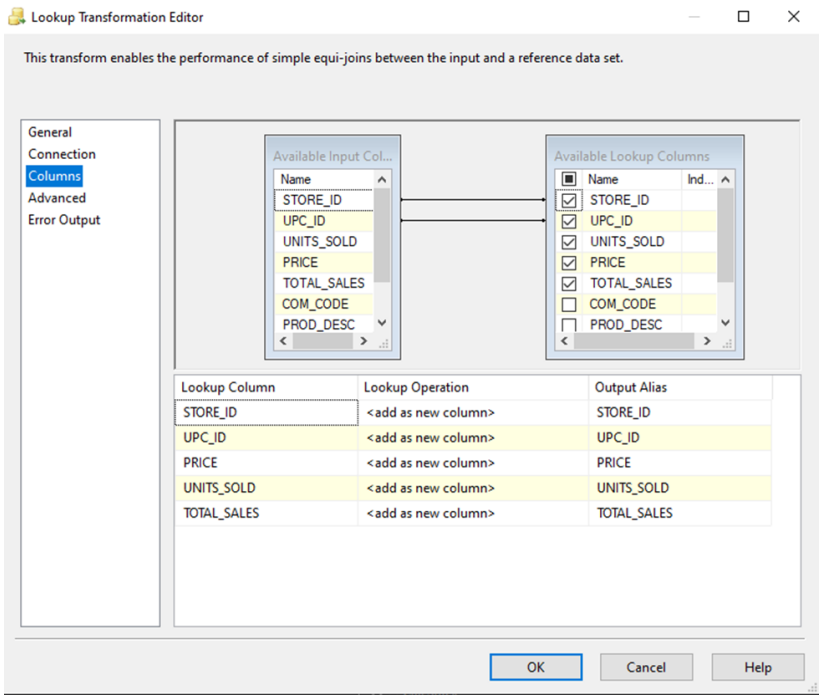


Fig: Performing Lookup transformation for Prod_dim

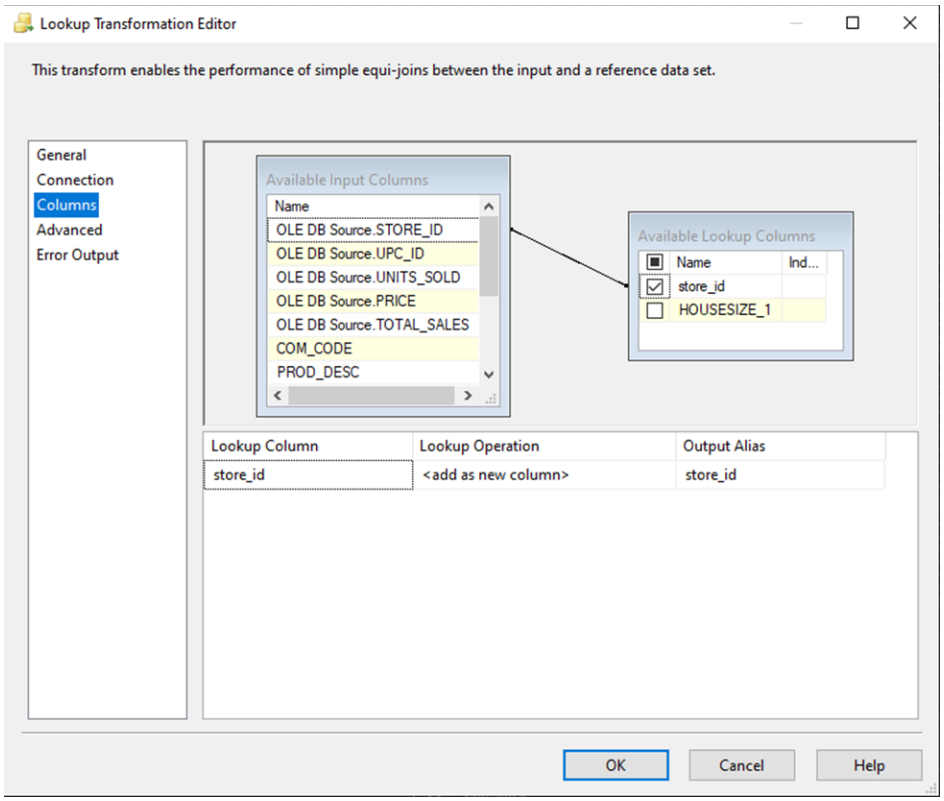


Fig: Performing lookup transformation for demo_dim

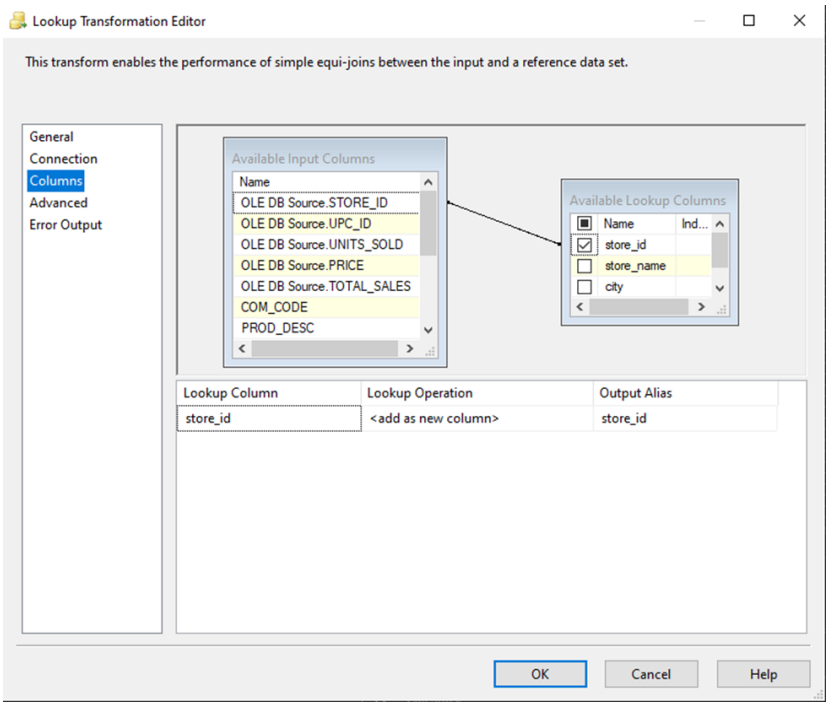


Fig: Performing lookup transformation for store_dim

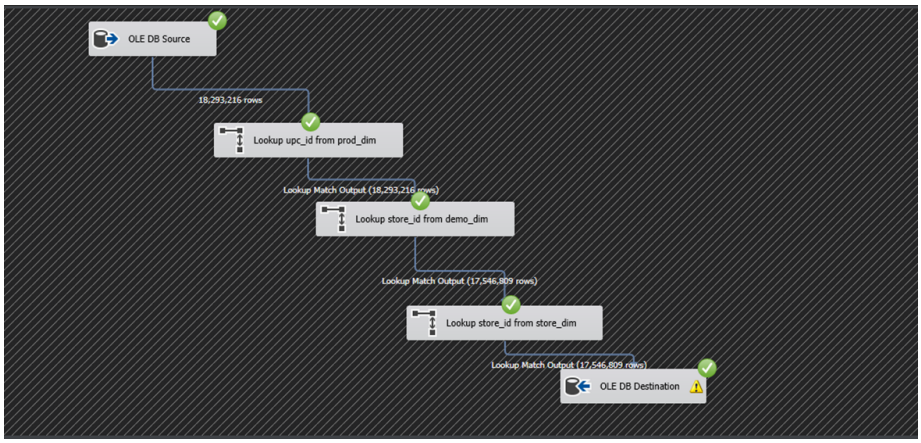
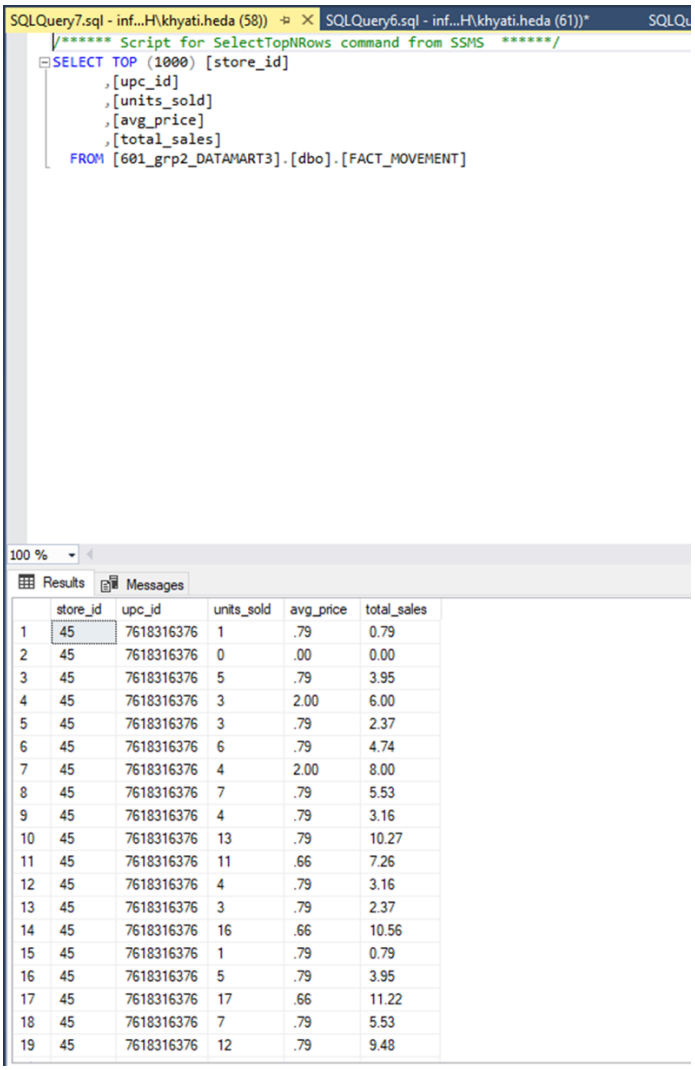


Fig: Lookup Transformation for fact_moevement with demo,store and prod



	store_id	upc_id	units_sold	avg_price	total_sales
1	45	7618316376	1	.79	0.79
2	45	7618316376	0	.00	0.00
3	45	7618316376	5	.79	3.95
4	45	7618316376	3	2.00	6.00
5	45	7618316376	3	.79	2.37
6	45	7618316376	6	.79	4.74
7	45	7618316376	4	2.00	8.00
8	45	7618316376	7	.79	5.53
9	45	7618316376	4	.79	3.16
10	45	7618316376	13	.79	10.27
11	45	7618316376	11	.66	7.26
12	45	7618316376	4	.79	3.16
13	45	7618316376	3	.79	2.37
14	45	7618316376	16	.66	10.56
15	45	7618316376	1	.79	0.79
16	45	7618316376	5	.79	3.95
17	45	7618316376	17	.66	11.22
18	45	7618316376	7	.79	5.53
19	45	7618316376	12	.79	9.48

Fig: fact_movement in SSMS

List OF Temporary Tables Created

TABLE NAME IN 601_grp2_stg_db	Description
Stg_SDR	Joining UPC and Movement for SDR
Stg_FRD	Joining UPC and Movement for FRD

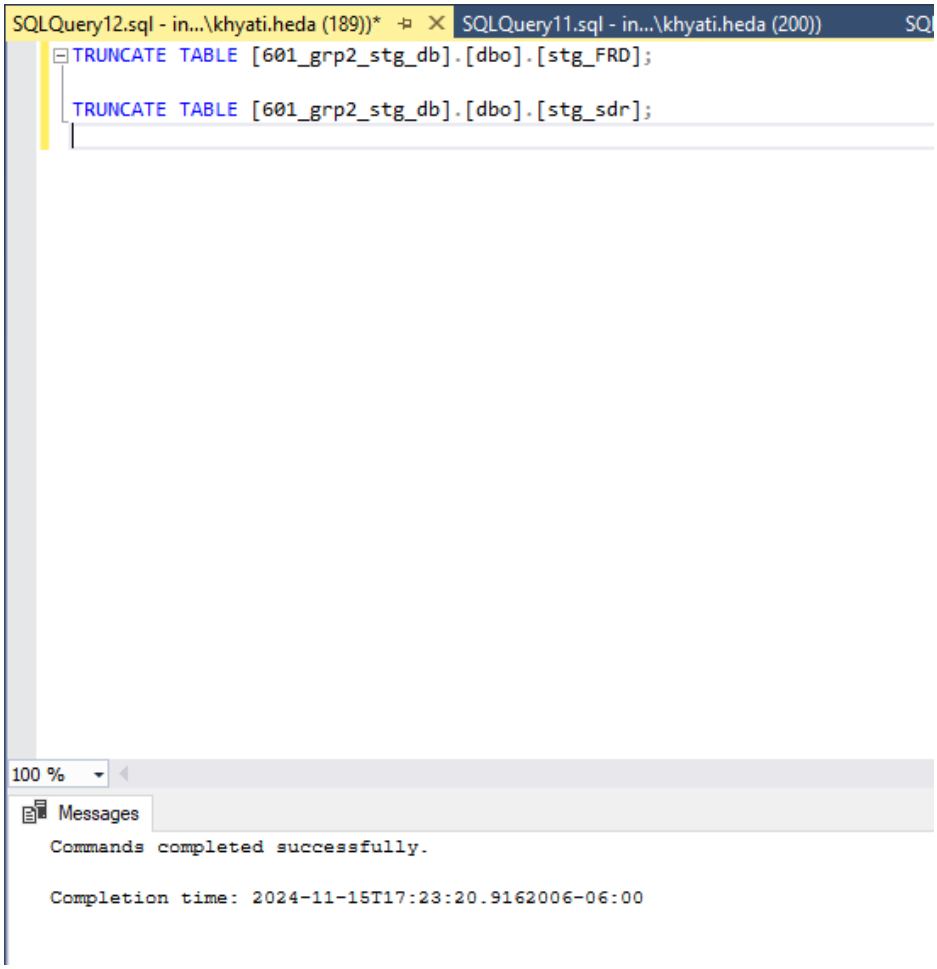


Fig: Deleted all temporary tables

Section 5: BI Reporting

Reporting Plan

During the reporting phase of our project, we used many businesses intelligence (BI) tools to generate insightful reports from the data warehouse created for Dominick Finer Foods (DFF) dataset. The main goal was to address key business questions, enabling system users to understand the business requirements and the significance of the data stored in the fact and dimension tables. Through the course of the project, we gained expertise in several visualization tools, which are summarized below:

1. **Microsoft Power BI**:

   Power BI is a modern and advanced business intelligence tool with powerful visualization capabilities, including trend charts, column graphs, and more. It offers flexibility by being accessible both on desktop and web browsers. Additionally, most of its core features are free, making it an attractive option for creating visually rich reports.

2. **SSRS – Microsoft SQL Server Reporting Services**:

   SSRS is a server-based platform designed for organizational reporting and data visualization. Its integration with Microsoft SQL Server and Visual Studio makes it ideal for generating drill-down and ad-hoc reports from the SQL Server data warehouse. This seamless compatibility ensures efficient reporting and visualization.

3. **SSAS – Microsoft SQL Server Analysis Services**:

   SSAS provides an analytical data engine to support decision-making and business analytics. It typically involves creating and deploying multidimensional models, such as cubes, which aggregate data across various measures. These cubes can then be used to generate reports in Analytical Services projects, providing a robust foundation for advanced analytics.

In summary, these tools collectively enabled us to deliver comprehensive and visually engaging reports, facilitating better decision-making and a deeper understanding of the data warehouse's impact.

a. Target Reports

The below table summarizes the target reports used for our business questions

Target Reports	Business Question	Justification
SSAS	Which stores are located in areas with the highest median income?	Understanding the median income of areas helps identify opportunities for premium product offerings, optimized pricing strategies, and tailored marketing campaigns for high-income consumers.
SSAS & SSRS	Which store is located in an area with the highest concentration of families, and how does this impact sales of family-oriented products?	Analyzing family demographics enables the store to tailor product inventory and marketing strategies, boosting sales of family-oriented products like bulk items, groceries, and toys in such areas.
Microsoft PowerBI	Which are the top 7 product categories that saw the highest selling record over the 1 year (1995)?	Tracking top-selling product categories enables the business to focus on inventory planning, marketing, and promotions for high-demand products, ensuring better customer satisfaction and profitability.
Microsoft PowerBI	Which stores have the lowest sales of soft drinks and the highest percentage of single-person households?	Identifying stores with low soft drink sales and high single-person households can help optimize product offerings, reduce waste, and target individual-serving products for these areas.
SSRS	What is the average price of frozen dinners for specific UPCs across all stores?	Calculating the average price for specific frozen dinner UPCs helps maintain consistent pricing strategies, ensures competitive pricing across stores, and identifies opportunities for cost optimization in procurement and pricing policies.

b. Mapping from the independent data marts to the report attributes in report template

Business Question 1: Which stores are located in areas with the highest median income?

DataMart Attribute	Table	Filters	Report Attribute
Store_id	Fact_store_sales	n/a	Store ID
store_name	store_dim	n/a	Store Name
city	store_dim	n/a	Store City
median_income	demo_dim	Count [median_income(previous)>median_income(current)] + 1	Rank

Business Question 2: Which store is in an area with the highest concentration of families and how does this impact sales of family-oriented products?

DataMart Attribute	Table	Filters	Report Attribute
Bulk_Total_Sales	Fact_Store_Sales	n/a	Bulk Total Sales
store_name	store_dim	n/a	Store Name
Housesize_3, Housesize_4, Housesize_5	demo_dim	SUM()	Total Family Households

Business Question 3: Which are the top 7 product categories that saw the highest selling record over the 1 year(1995)?

DataMart Attribute	Table	Filters	Report Attribute
Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum, Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum	Fact_Category_Sales	SUM()	Dairy_sum, Frozen_sum, Meat_sum, Produce_sum, Deli_sum, GM_Sum, Grocery_Sum, Bakery_Sum, Beer_Sum, Bottle_Sum, Bulk_Sum, Camera_Sum, Cheese_Sum, ConvFood_Sum, Cosmetic_Sum, Fish_Sum, Jewelry_Sum, HABA_Sum, Video_Sum, Wine_Sum, Spirits_Sum, FTGCHIN_Sum, FTGITAL_Sum, Floral_Sum, SaladBar_Sum, MeatFrozen_Sum
Year	time_dim	n/a	Year

Business Question 4: Which stores have the lowest sales of soft drinks and the highest percentage of single- person households?

DataMart Attribute	Table	Filters	Report Attribute
total_sales	Fact_Movement	SUM()	Sum of total_sales
store_name	store_dim	n/a	Store Name
Housesize_1	demo_dim	n/a	Housesize_1
Category	prod_dim	'Soft Drinks'	Category

Business Question 5: What is the average price of frozen dinners for specific UPCs across all stores?

DataMart Attribute	Table	Filters	Report Attribute
avg_price	Fact_Movement	Avg()	Avg Price
UPC_ID	prod_dim	n/a	UPC ID
Category	prod_dim	'Frozen Dinner'	Category

Reporting Implementation

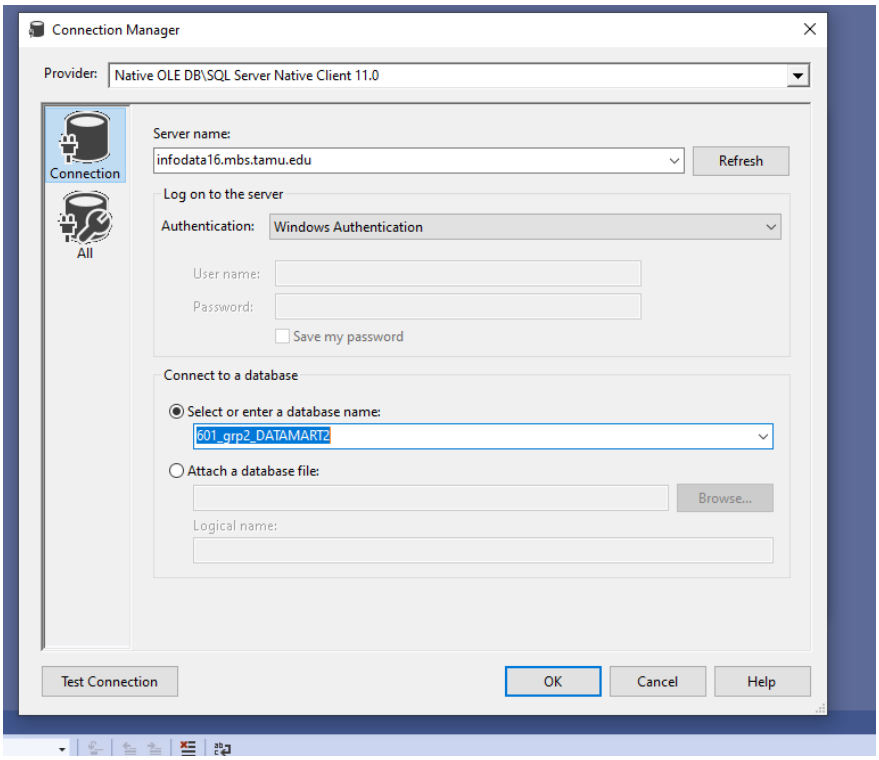
1. Which stores are in areas with the highest median income?

Rationale:

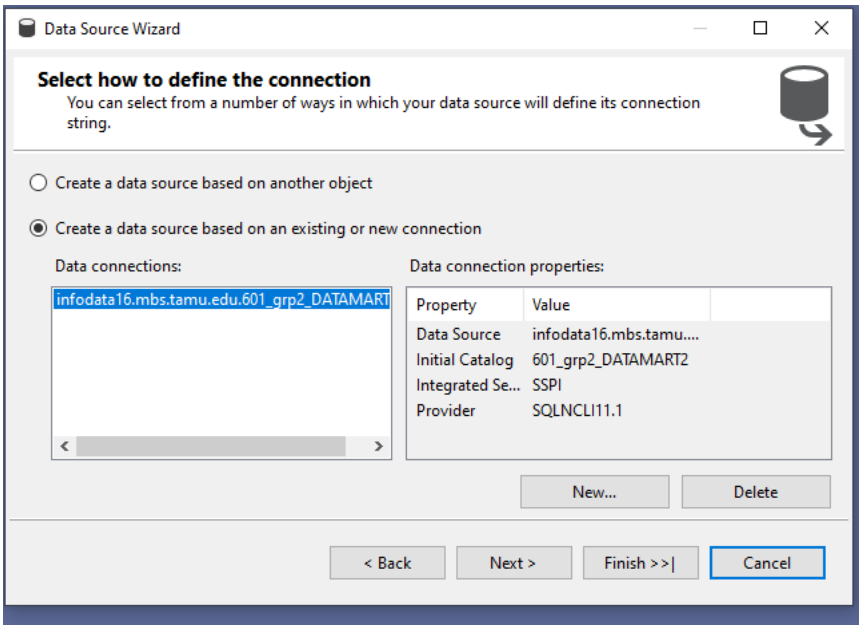
This analysis offers valuable insights into store locations situated in affluent areas, allowing Dominick's Fine Foods (DFF) to strategically target higher-income consumers. By leveraging these demographics, DFF can refine its approach by introducing premium or luxury product lines tailored to the preferences of affluent shoppers, optimizing pricing strategies to align with their higher purchasing power, and designing targeted marketing campaigns such as loyalty programs, exclusive deals, or eco-friendly product options. Additionally, this analysis can inform expansion plans, enabling DFF to identify high-income regions where opening new stores could maximize profitability and customer satisfaction.

Reporting Tool used: SSAS

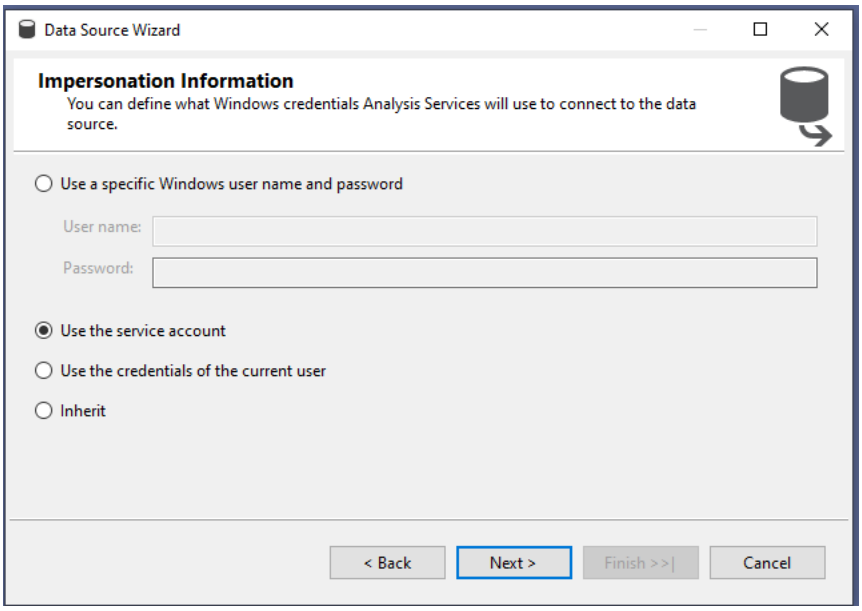
Step 1: Creating a new data source by specifying the server name and selecting the data mart.



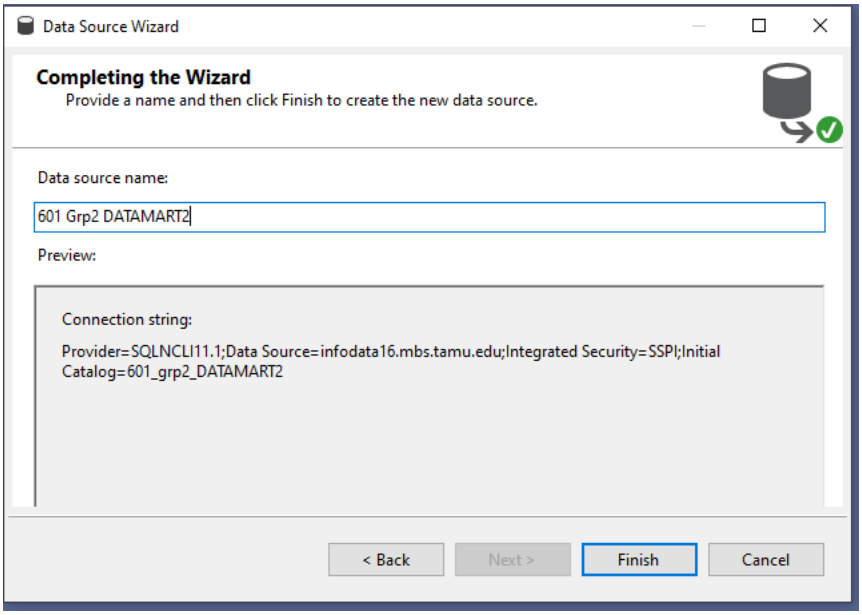
Step 2: The connection to the data mart is added in the Data Source Wizard



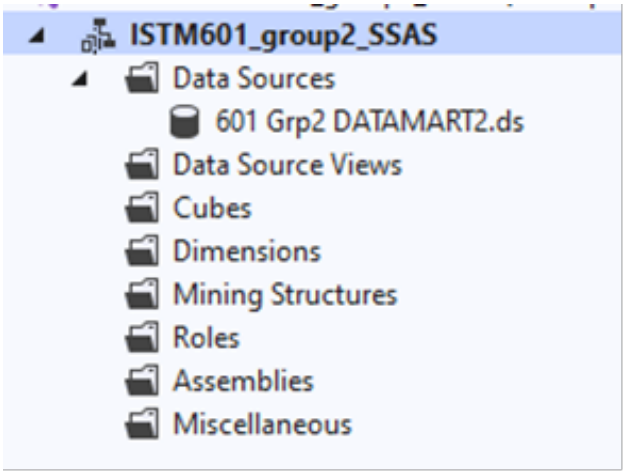
Step 3: For Impersonation Information, we select the option ‘Use the service account’



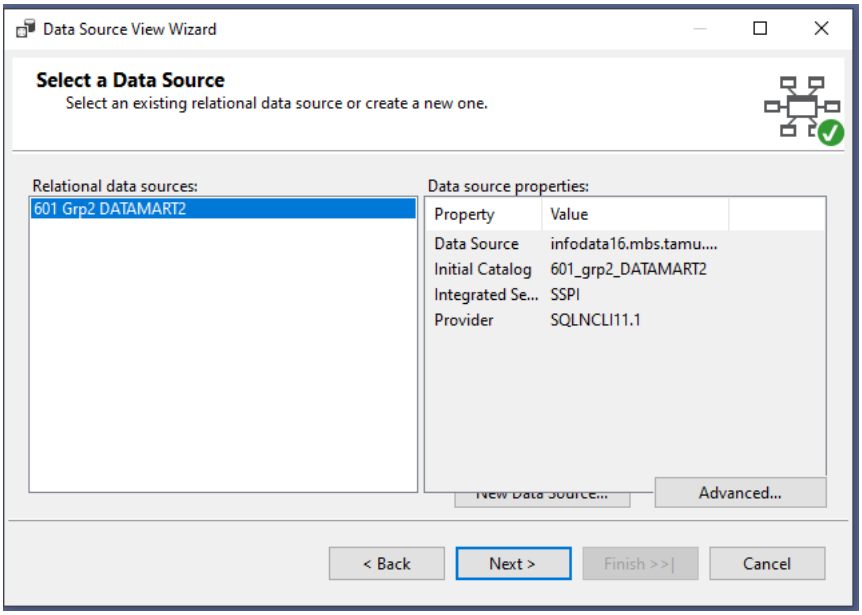
Step 4: Give a name to the data source and click on finish



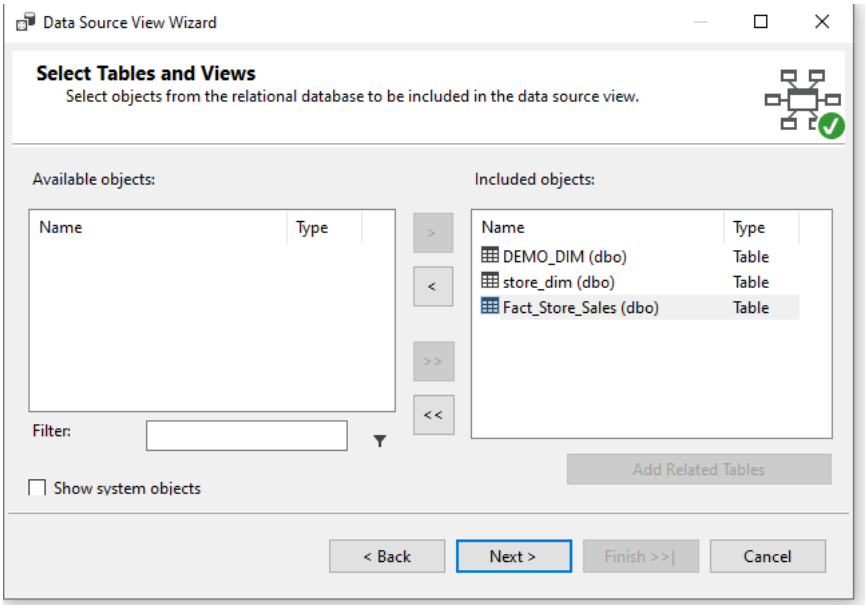
Step 5: Check where the data source appears in solution explorer



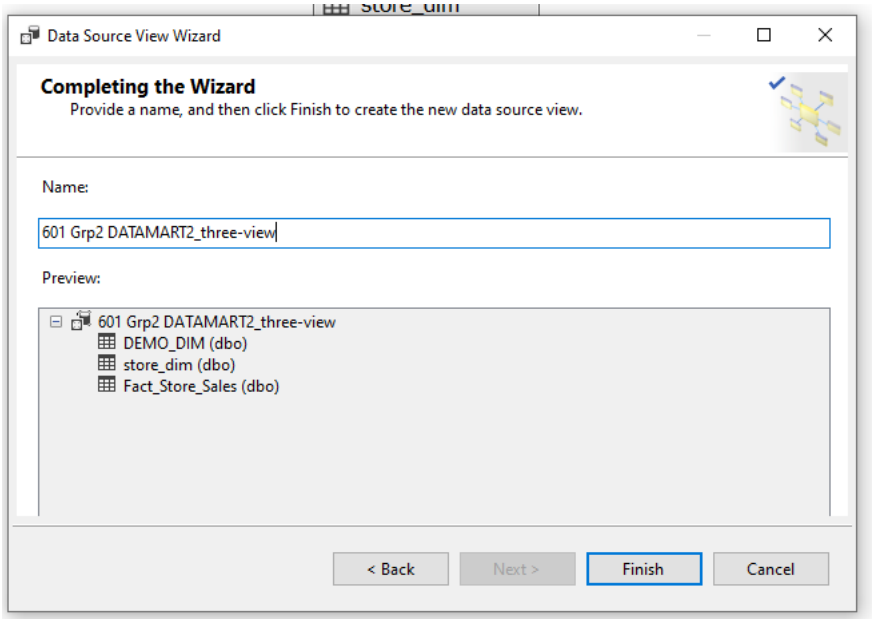
Step 6: Create a new data source view by selecting the data source created in previous steps



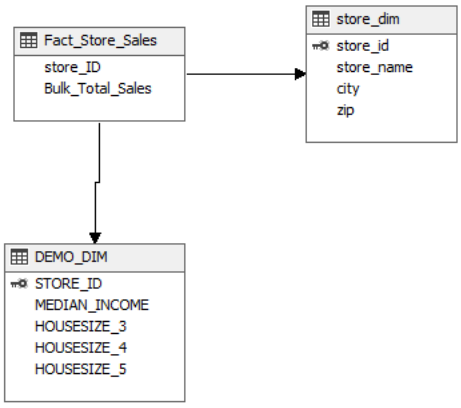
Step 7: Select all the required tables



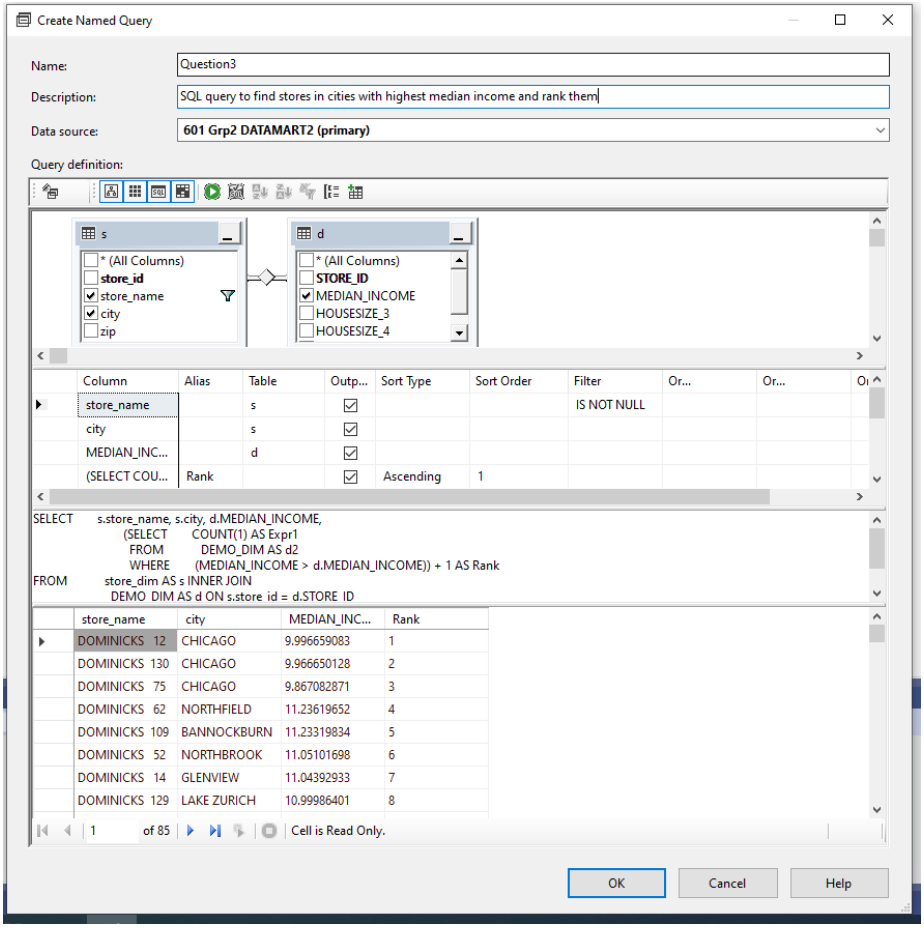
Step 8: Give a name to the data source view and click on ‘Finish’



Step 9: We can now view the tables and relationships in the data source view



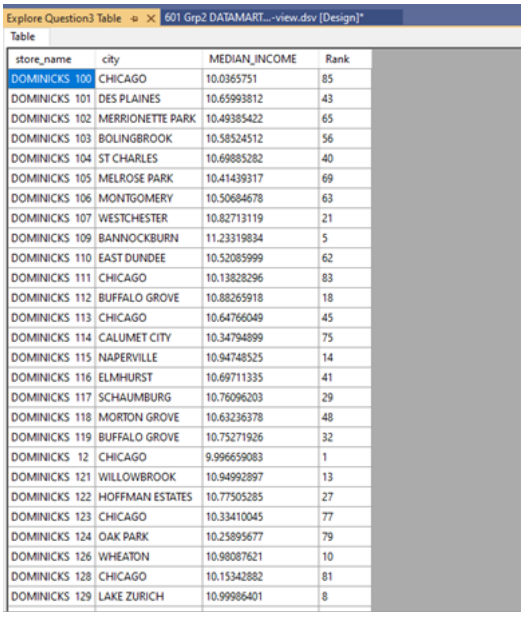
Step 10: Create a new named query to build the reporting data by adding an appropriate query name and SQL query



SQL query used:

```
SELECT
    s.store_name,
    s.city,
    d.MEDIAN_INCOME,
    (SELECT COUNT(1)
     FROM DEMO_DIM d2
     WHERE d2.MEDIAN_INCOME > d.MEDIAN_INCOME) + 1 AS Rank
FROM store_dim s
INNER JOIN DEMO_DIM d
    ON s.store_id = d.STORE_ID
WHERE s.store_name IS NOT NULL
ORDER BY Rank;
```

Step 11: Click on Explore data to view the reporting data

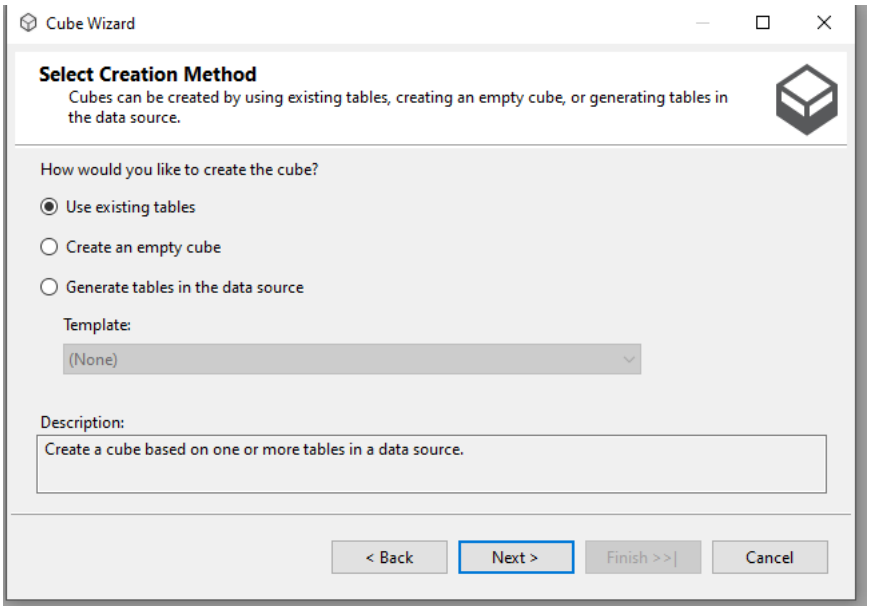


Explore Question3 Table | 601 Grp2 DATAMART...-view.dsv [Design]*

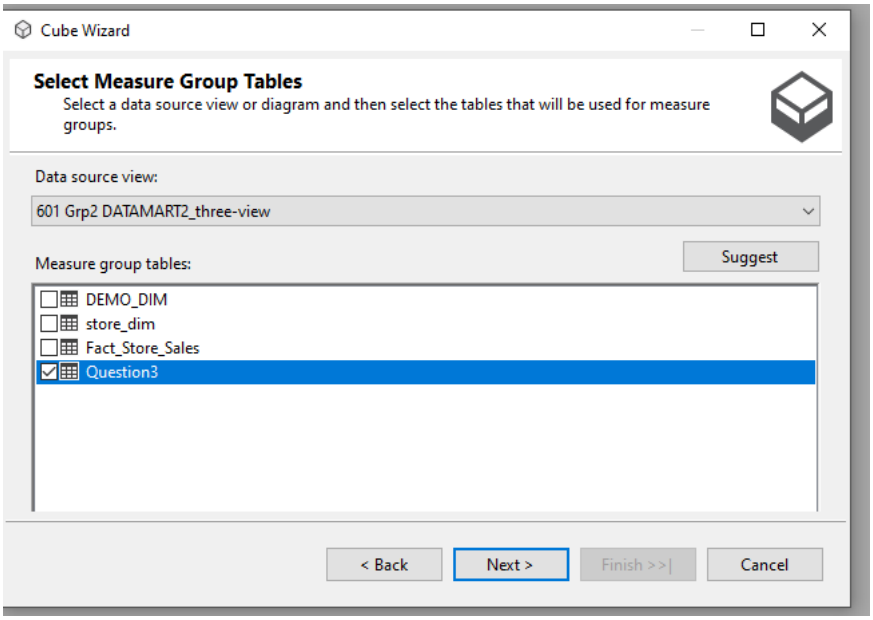
Table

store_name	city	MEDIAN_INCOME	Rank
DOMINICKS 100	CHICAGO	10.0365751	85
DOMINICKS 101	DES PLAINES	10.65993812	43
DOMINICKS 102	MERRIONETTE PARK	10.49385422	65
DOMINICKS 103	BOLINGBROOK	10.58524512	56
DOMINICKS 104	ST CHARLES	10.69885282	40
DOMINICKS 105	MELROSE PARK	10.41439317	69
DOMINICKS 106	MONTGOMERY	10.50684678	63
DOMINICKS 107	WESTCHESTER	10.82713119	21
DOMINICKS 109	BANNOCKBURN	11.23319834	5
DOMINICKS 110	EAST DUNDEE	10.52085999	62
DOMINICKS 111	CHICAGO	10.13828296	83
DOMINICKS 112	BUFFALO GROVE	10.88265918	18
DOMINICKS 113	CHICAGO	10.64766049	45
DOMINICKS 114	CALUMET CITY	10.34794899	75
DOMINICKS 115	NAPERVILLE	10.94748525	14
DOMINICKS 116	ELMHURST	10.69711335	41
DOMINICKS 117	SCHAUMBURG	10.76096203	29
DOMINICKS 118	MORTON GROVE	10.63236378	48
DOMINICKS 119	BUFFALO GROVE	10.75271926	32
DOMINICKS 12	CHICAGO	9.996659083	1
DOMINICKS 121	WILLOWBROOK	10.94992897	13
DOMINICKS 122	HOFFMAN ESTATES	10.77505285	27
DOMINICKS 123	CHICAGO	10.33410045	77
DOMINICKS 124	OAK PARK	10.25895677	79
DOMINICKS 126	WHEATON	10.98087621	10
DOMINICKS 128	CHICAGO	10.15342882	81
DOMINICKS 129	LAKE ZURICH	10.99986401	8

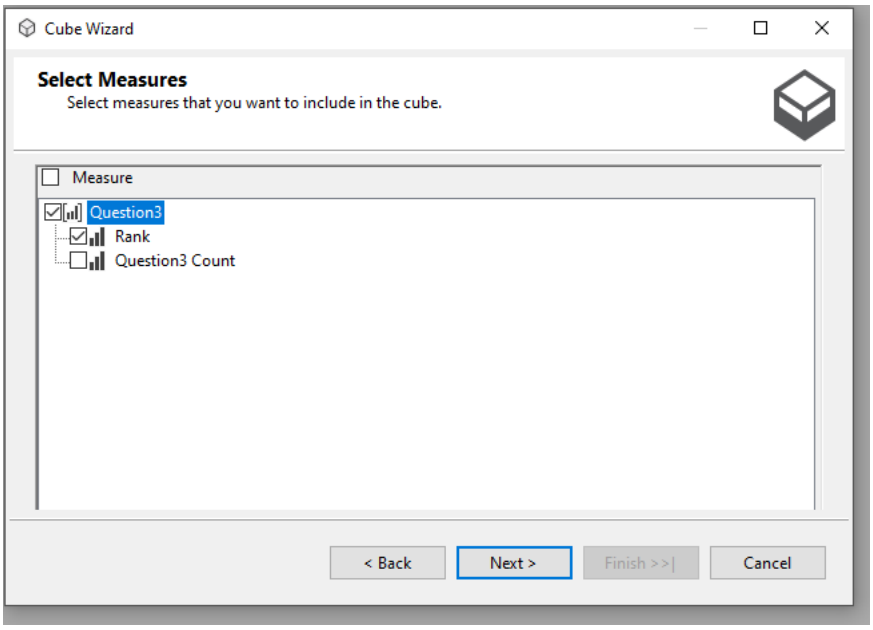
Step 12: Create a new cube by using Cube Wizard



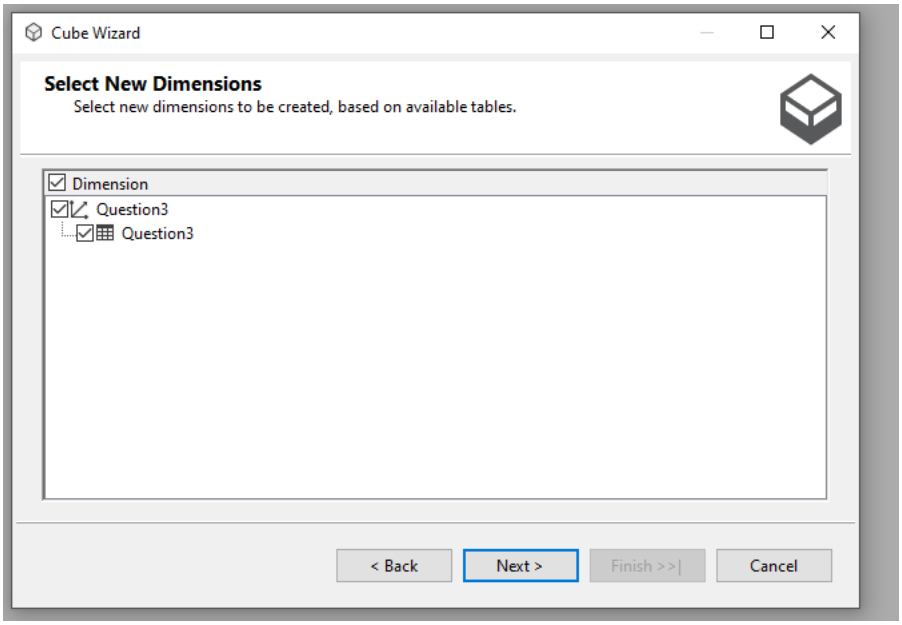
Step 13: Select Measure Group tables



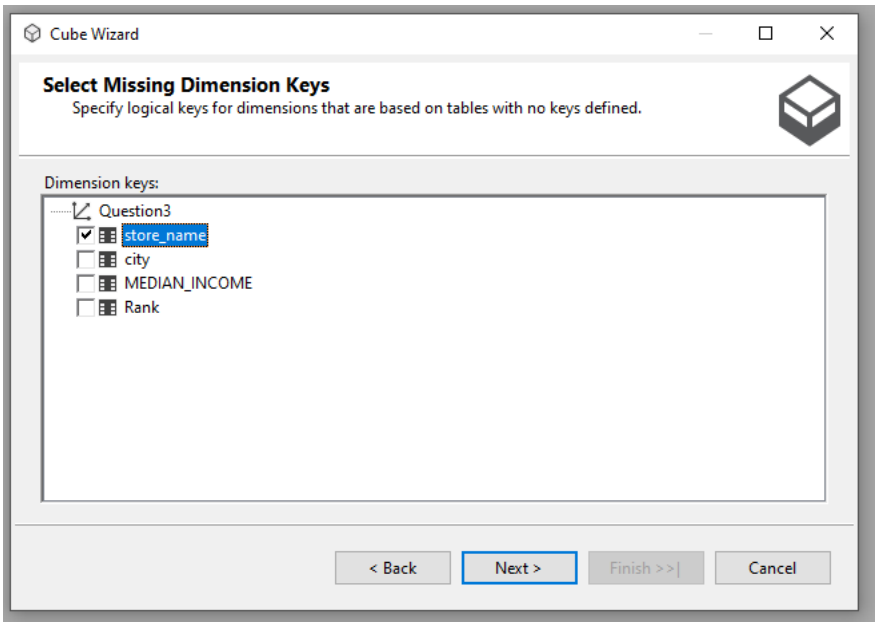
Step 14: Select measures



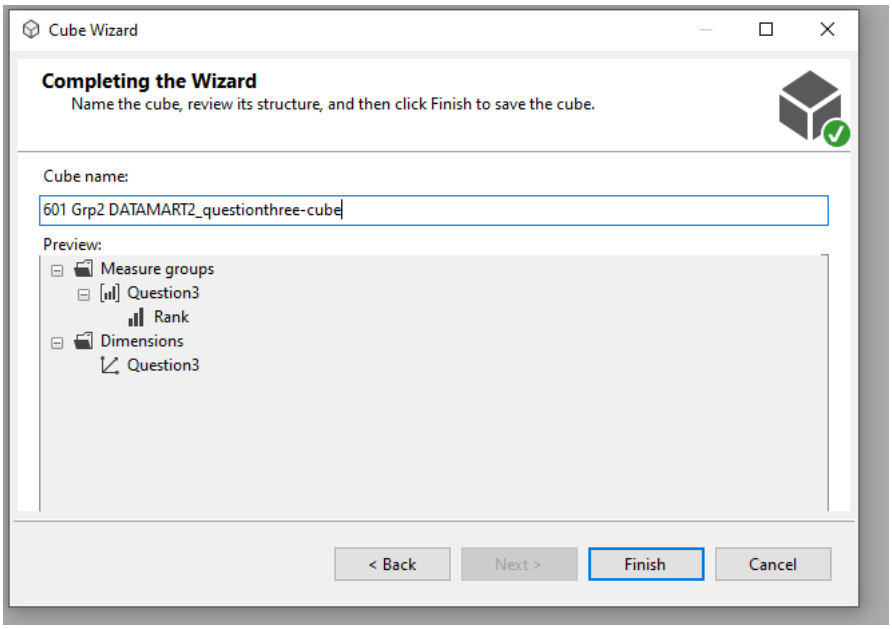
Step 15: Select new dimensions



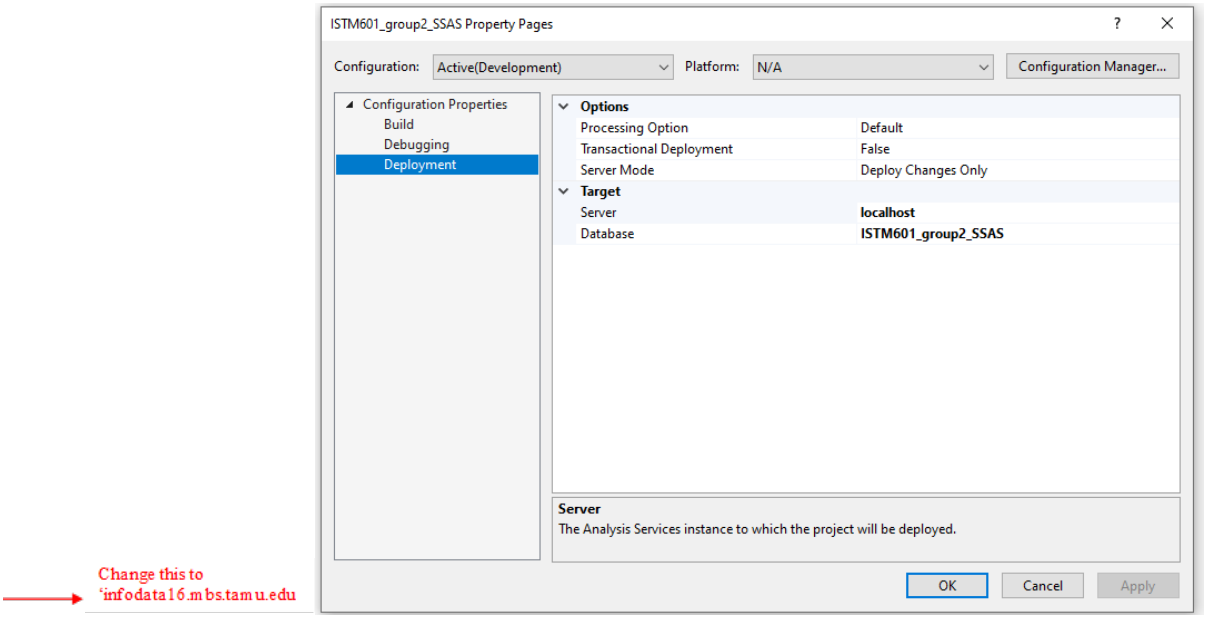
Step 16: Specify dimension key



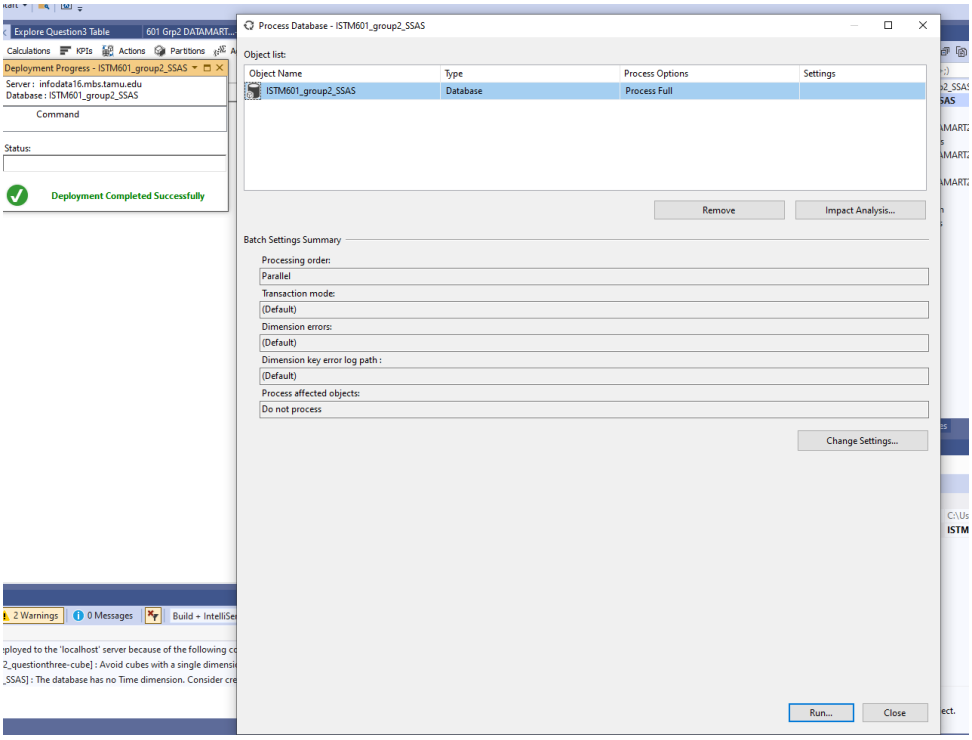
Step 17: Give the cube a name and click on ‘Finish’



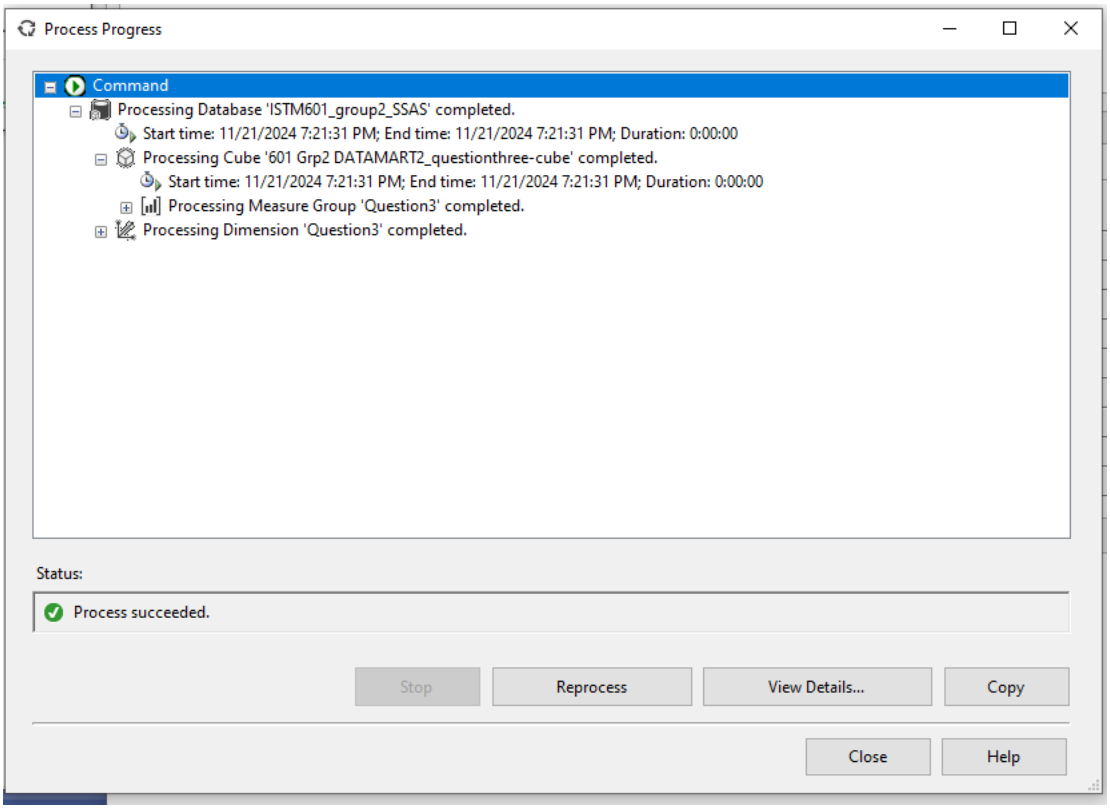
Step 18: In the property pages of the project, change the server to ‘infodata16.mbs.tamu.edu’



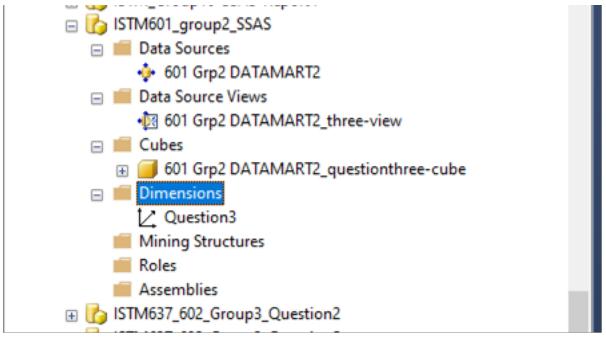
Step 19: Process the deployment of the project created



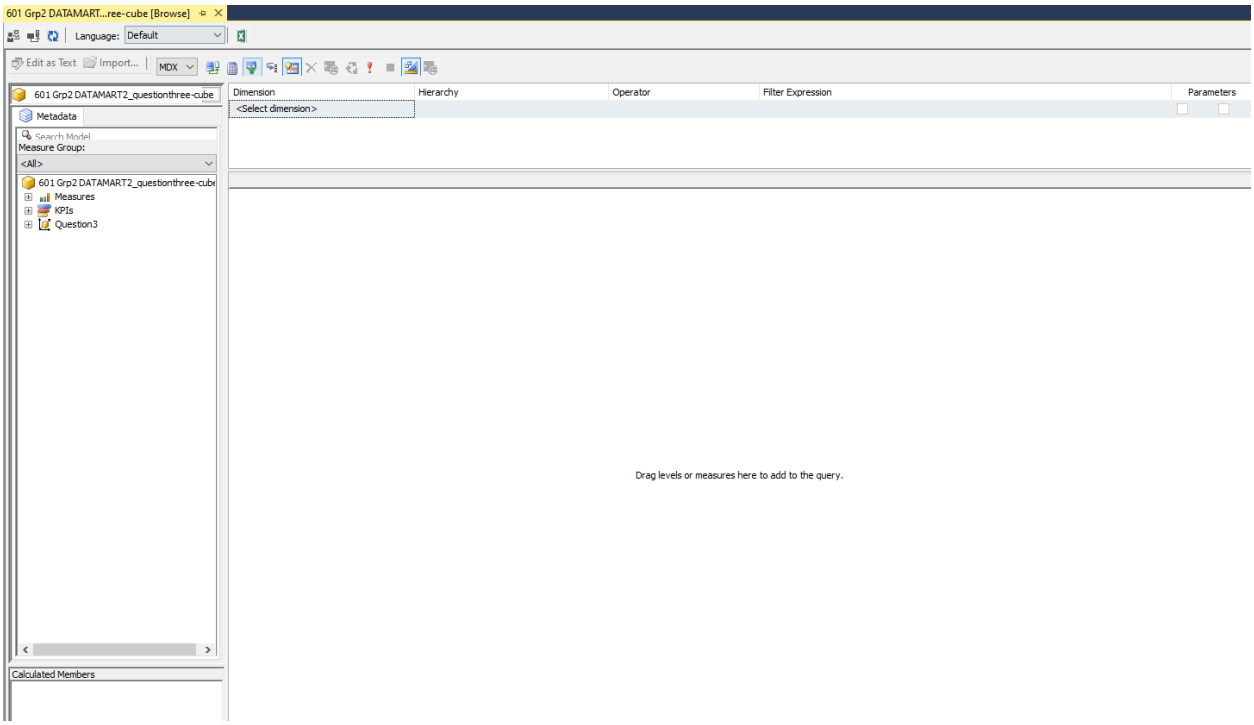
Step 20: Check if process has succeeded



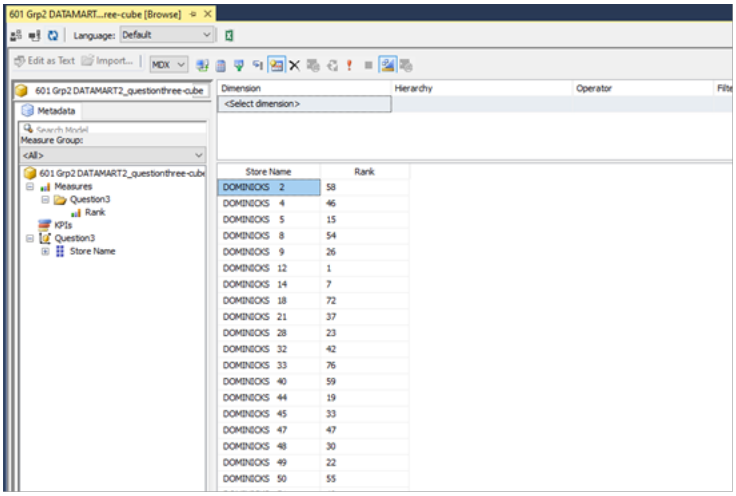
Step 21: Go to SSAS and check if analysis database is available



Step 22: Browse the cube to create the report



Step 23: Drag the required measures for justifying the business question and execute query to generate report



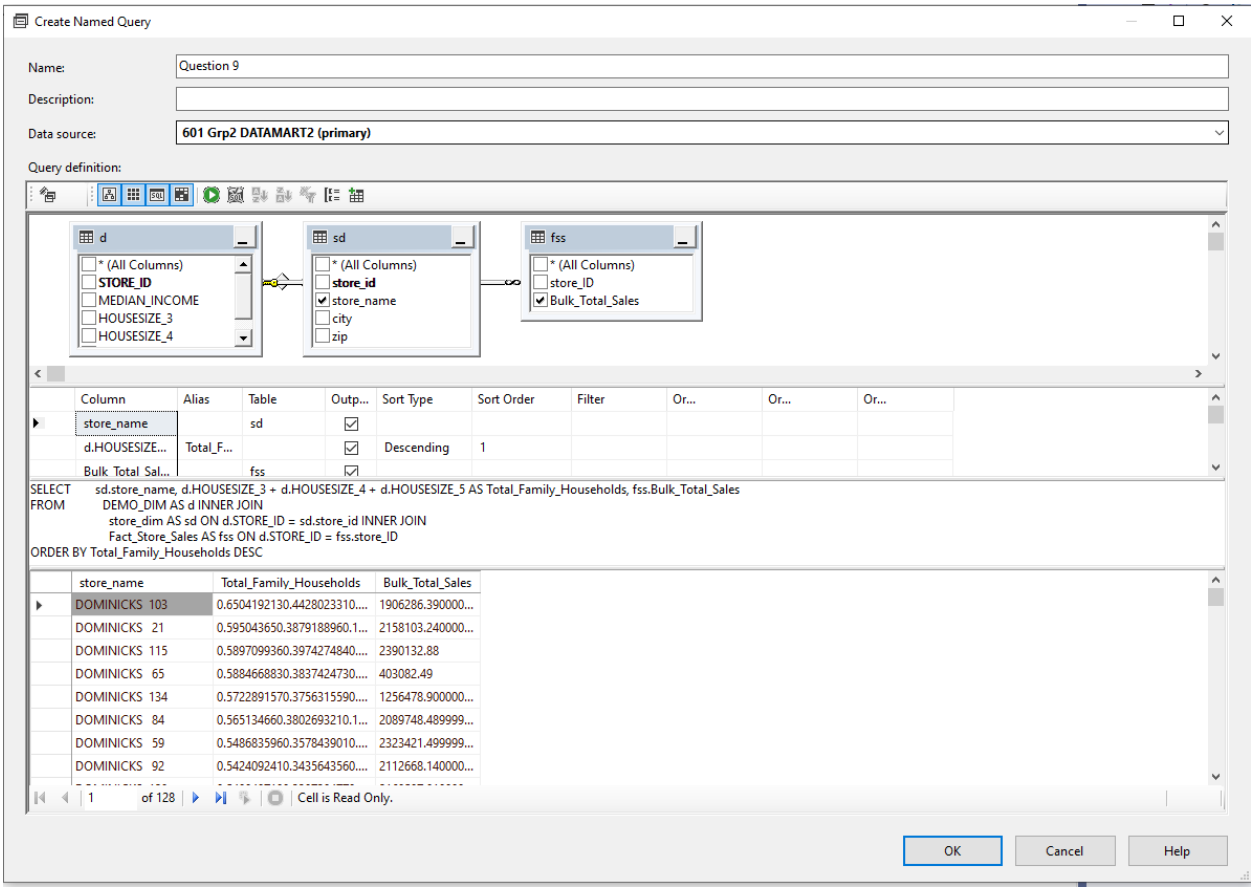
2. **Which store is in an area with the highest concentration of families and how does this impact sales of family-oriented products?**

Rationale:
Identifying the stores located in areas with the highest concentration of larger families (households with 3 or more people) can help optimize product offerings and promotions. Family-oriented products, bulk items, or deals designed for larger households can be better targeted, improving sales and customer satisfaction. This also helps in managing inventory more effectively in stores that serve a predominantly family-based demographic.

Reporting Tools used: SSAS and SSRS

While using this method, we already have the data source loaded from the previous question so we can create a modified view over the same data.

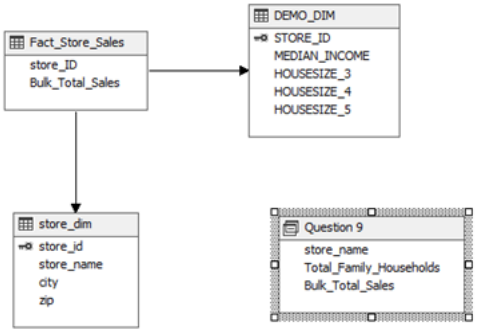
Step 1: Creating a new named query in SSAS



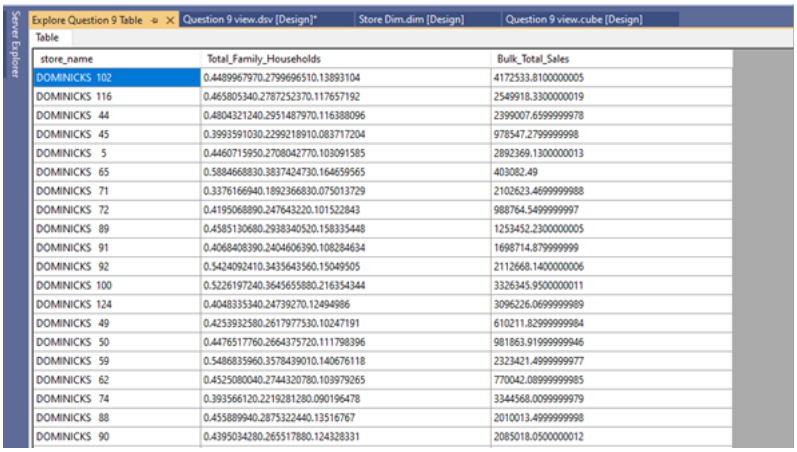
SQL query used:

```
SELECT sd.store_name, d.HOUSESIZE_3 + d.HOUSESIZE_4 + d.HOUSESIZE_5 AS
Total_Family_Households, fss.Bulk_Total_Sales
FROM DEMO_DIM AS d INNER JOIN
    store_dim AS sd ON d.STORE_ID = sd.store_id INNER JOIN
    Fact_Store_Sales AS fss ON d.STORE_ID = fss.store_ID
ORDER BY Total_Family_Households DESC;
```

Step 2: Modified Data Source View



Step 3: Click on explore data to see the data in the view

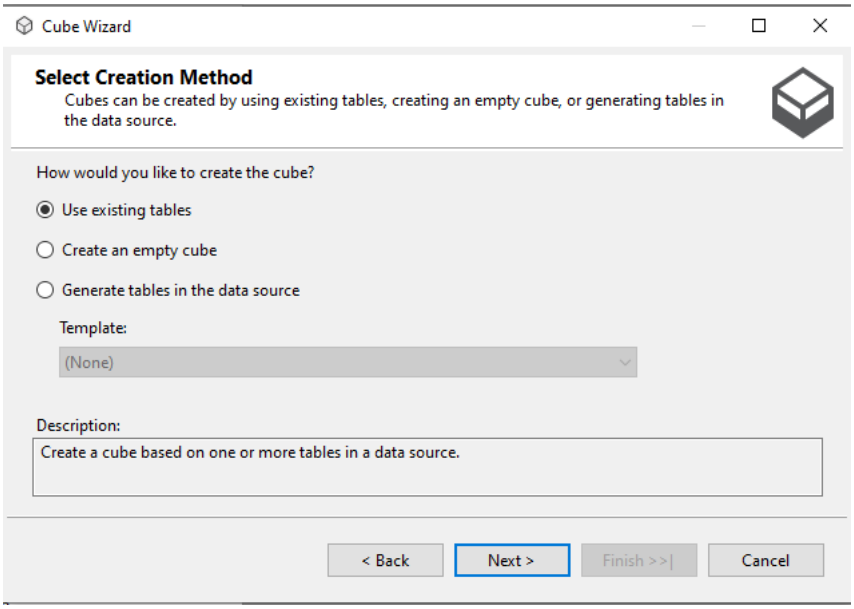


Explore Question 9 Table | Question 9 view.dsv [Design]* | Store Dim.dim [Design] | Question 9 view.cube [Design]

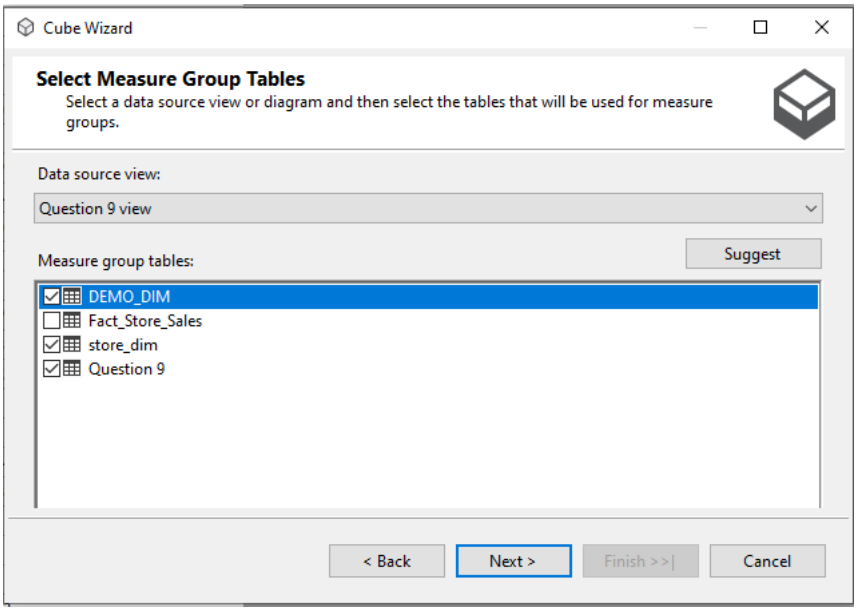
Table

store_name	Total_Family_Households	Bulk_Total_Sales
DOMINICKS 102	0.4489967970.2799696510.13893104	4172533.8100000005
DOMINICKS 116	0.465805340.2787252370.117657192	2549918.3300000019
DOMINICKS 44	0.4804321240.2951487970.116388096	2399007.6599999978
DOMINICKS 45	0.3993591030.2299218910.083717204	978547.2799999998
DOMINICKS 5	0.4460715950.2708042770.103091585	2892369.1300000013
DOMINICKS 65	0.5884668830.3837424730.164659565	403082.49
DOMINICKS 71	0.3376166940.1892366830.075013729	2102623.4699999988
DOMINICKS 72	0.4195068890.247643220.101522843	988764.5499999997
DOMINICKS 89	0.4585130680.2938340520.158335448	1253452.2300000005
DOMINICKS 91	0.4068408390.2404606390.108284634	1698714.879999999
DOMINICKS 92	0.5424092410.3435643560.15049505	2112668.1400000006
DOMINICKS 100	0.5226197240.3645655880.216354344	3326345.9500000011
DOMINICKS 124	0.4048335340.24739270.12494986	3096226.0699999989
DOMINICKS 49	0.4253932580.2617977530.10247191	610211.82999999984
DOMINICKS 50	0.4476517760.2664375720.111798396	981863.91999999946
DOMINICKS 59	0.5486835960.3578439010.140676118	2323421.4999999977
DOMINICKS 62	0.4525080040.2744320780.103979265	770042.08999999985
DOMINICKS 74	0.393566120.2219281280.090196478	3344568.0099999979
DOMINICKS 88	0.455889940.2875322440.13516767	2010013.4999999998
DOMINICKS 90	0.4395034280.265517880.124328331	2085018.0500000012

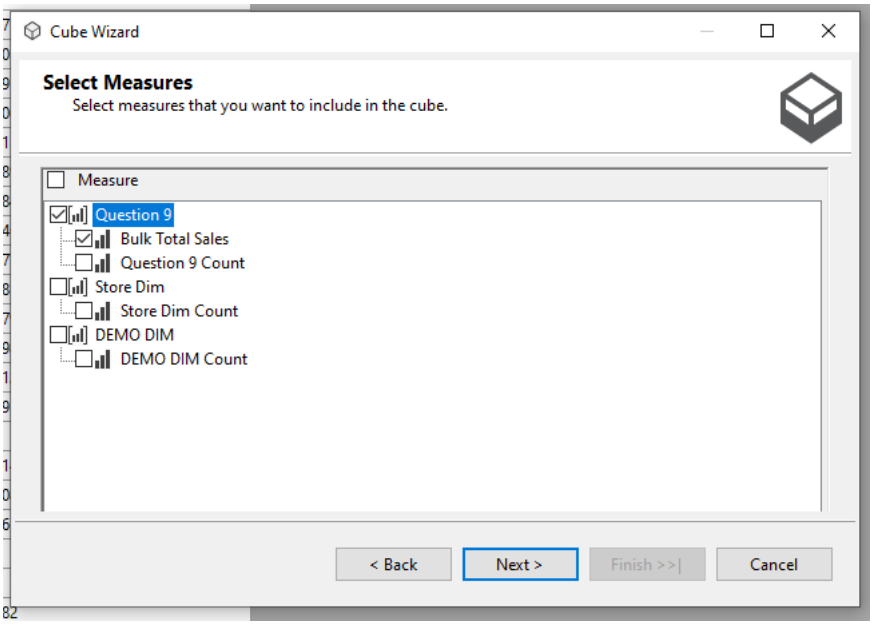
Step 4: We create a new cube using existing tables



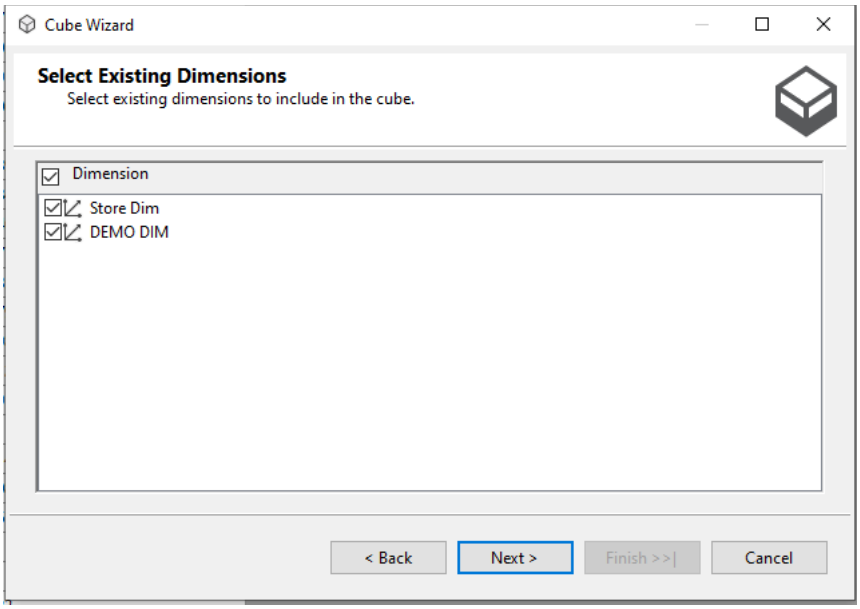
Step 5: Select measure group tables



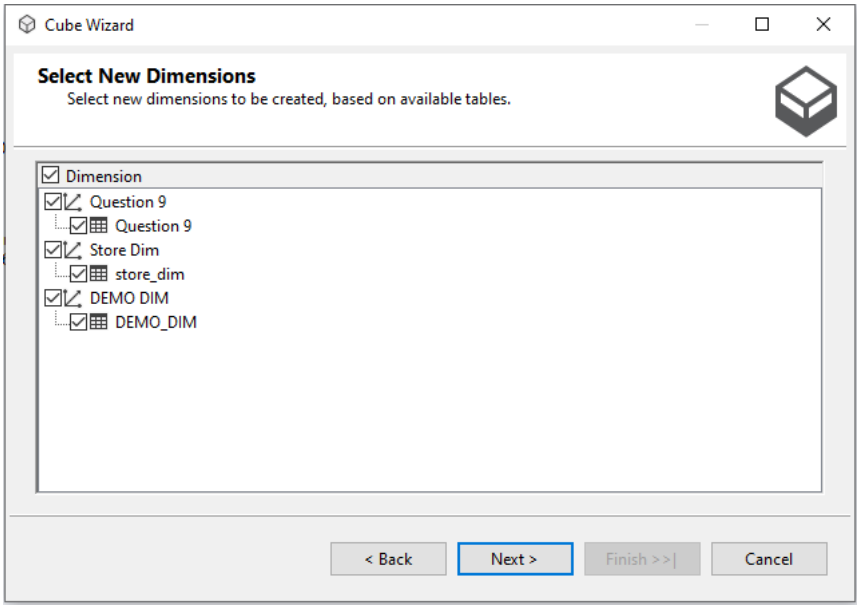
Step 6: Select measures



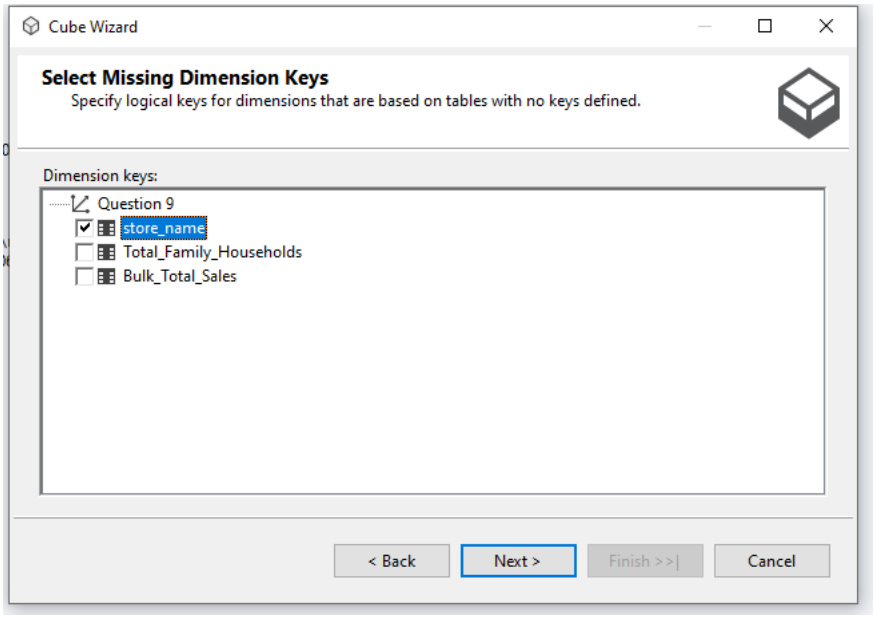
Step 7: Select from existing dimensions



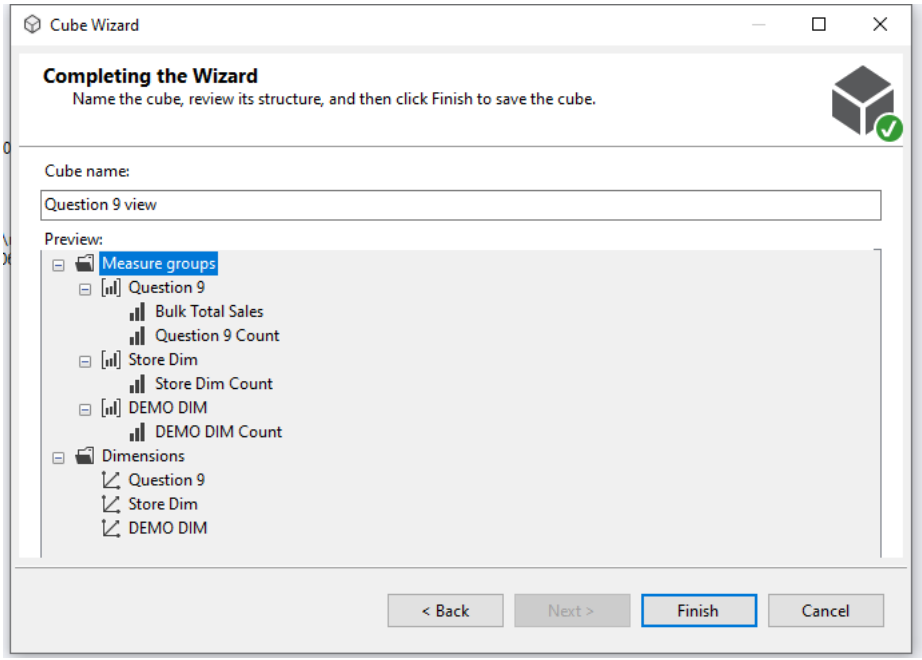
Step 8: Select from new dimensions



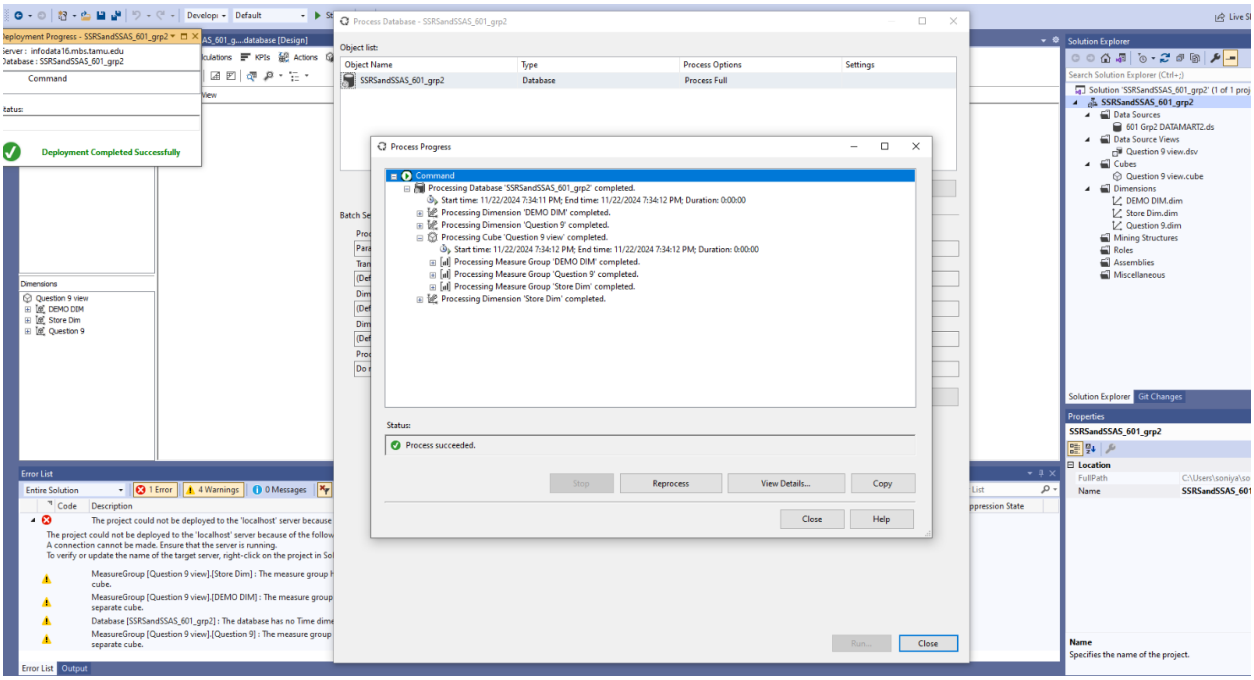
Step 9: Select dimension keys



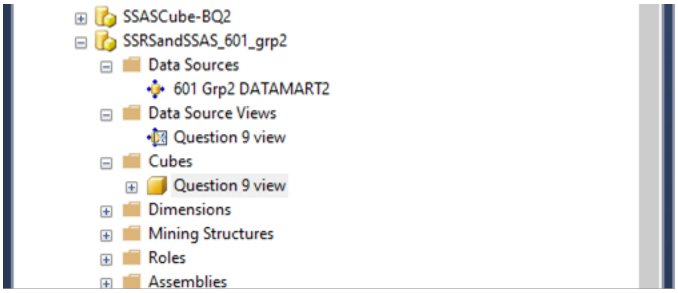
Step 10: Review all the data and complete creating the cube



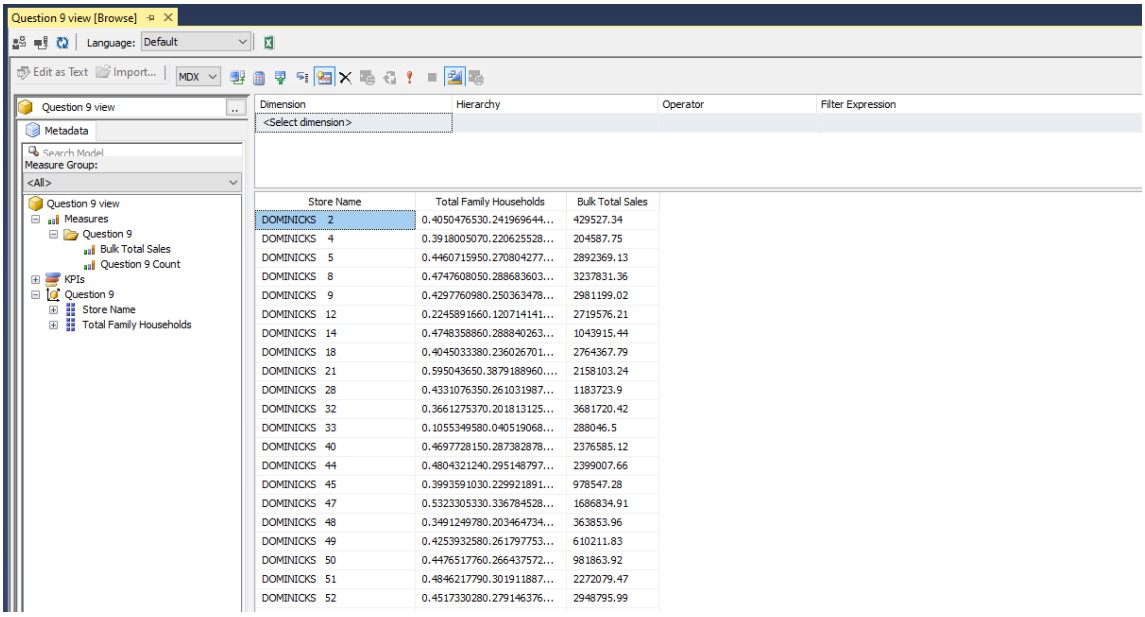
Step 11: Process the change to SSAS



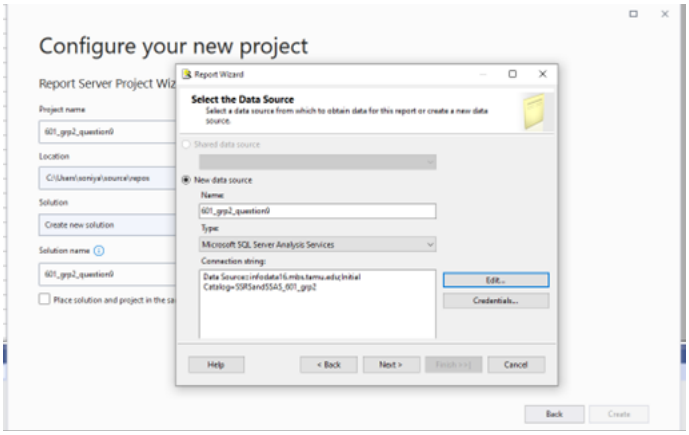
Step 12: Check for the availability of the new view and cube in SSAS



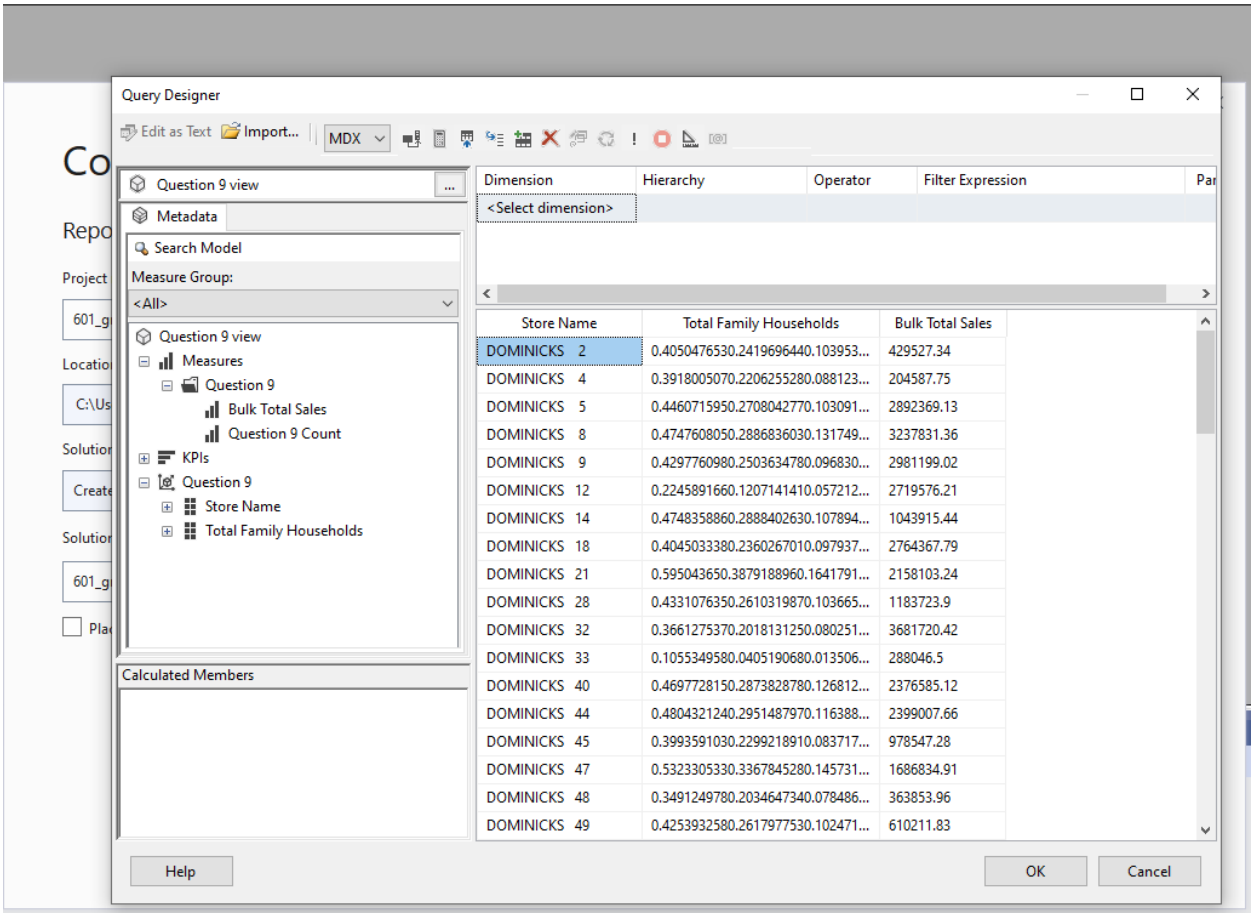
Step 13: Drag appropriate measures from the cube and run the query to view the reporting data in SSAS



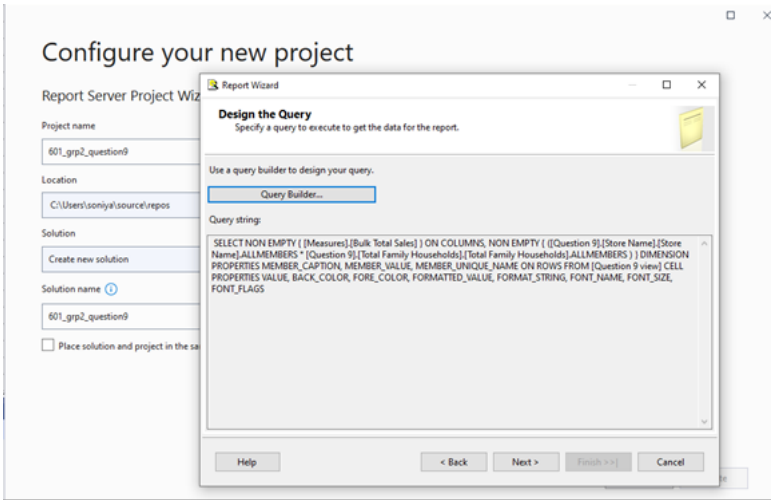
Step 14: Create a connection with SSAS while creating a new SSRS project



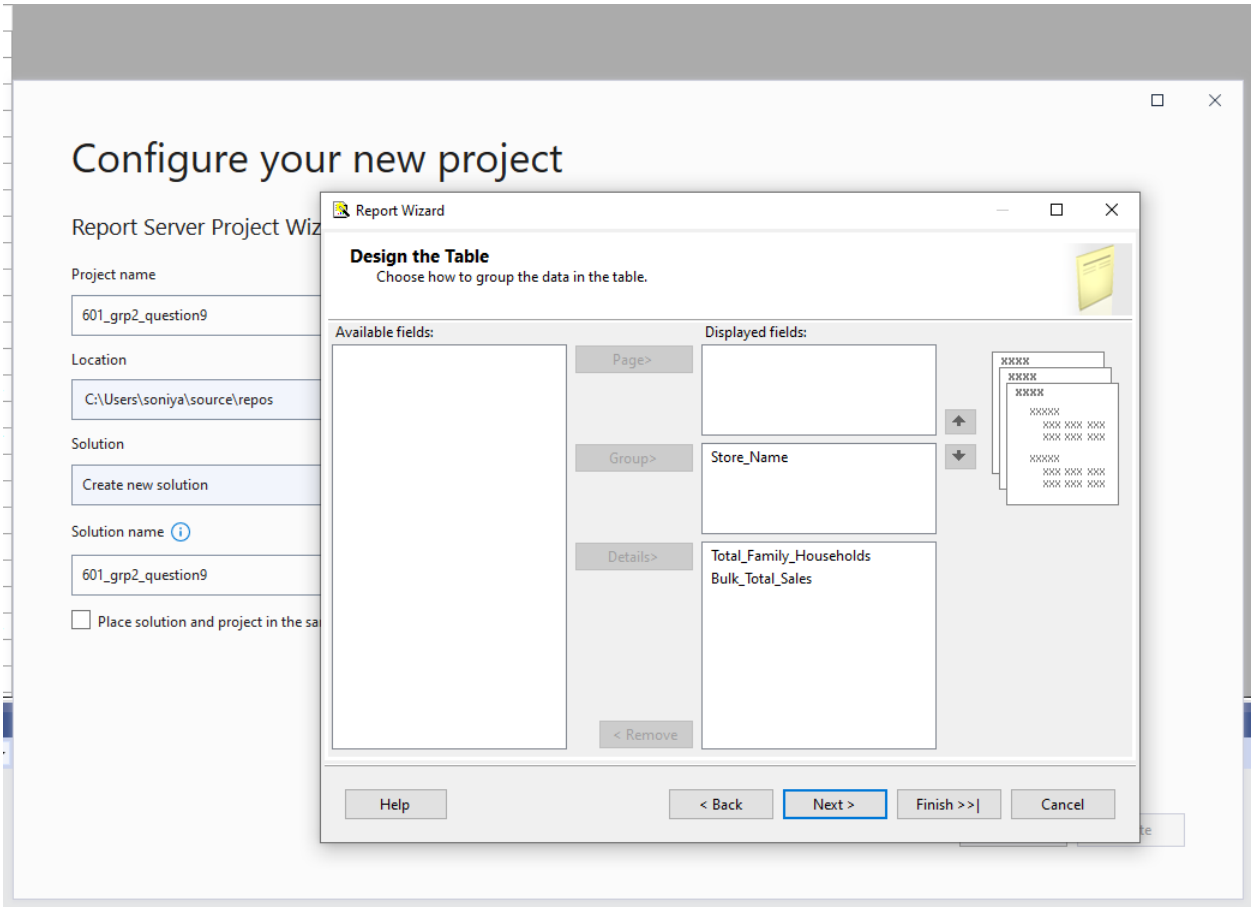
Step 15: In query designer, select the right measures and dimensions to create a query for building the SSRS report.



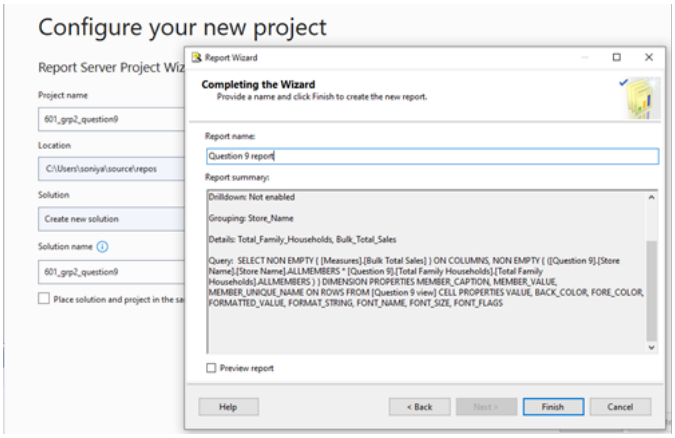
Step 16: The query is built by the query designer



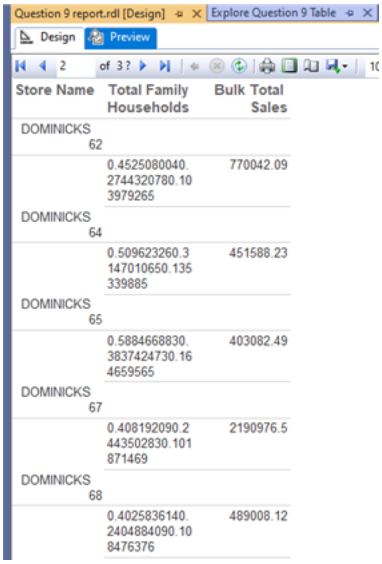
Step 17: Drag the columns in way that you want to structure/group data in your report



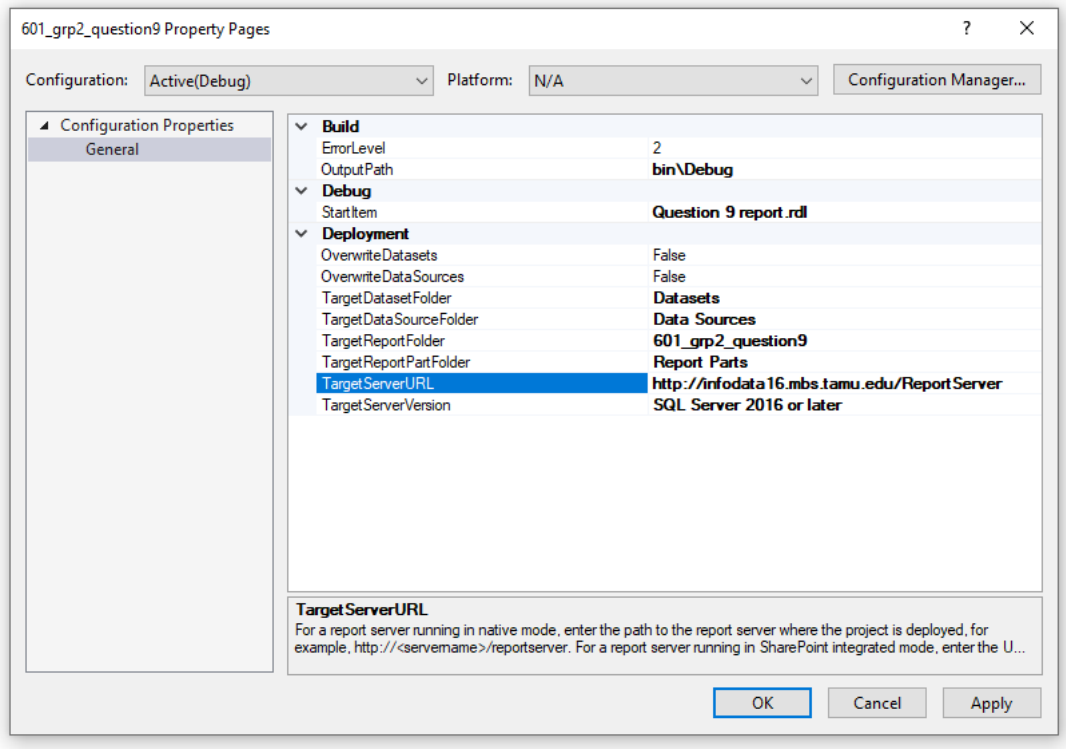
Step 18: Review all the selections and complete the wizard



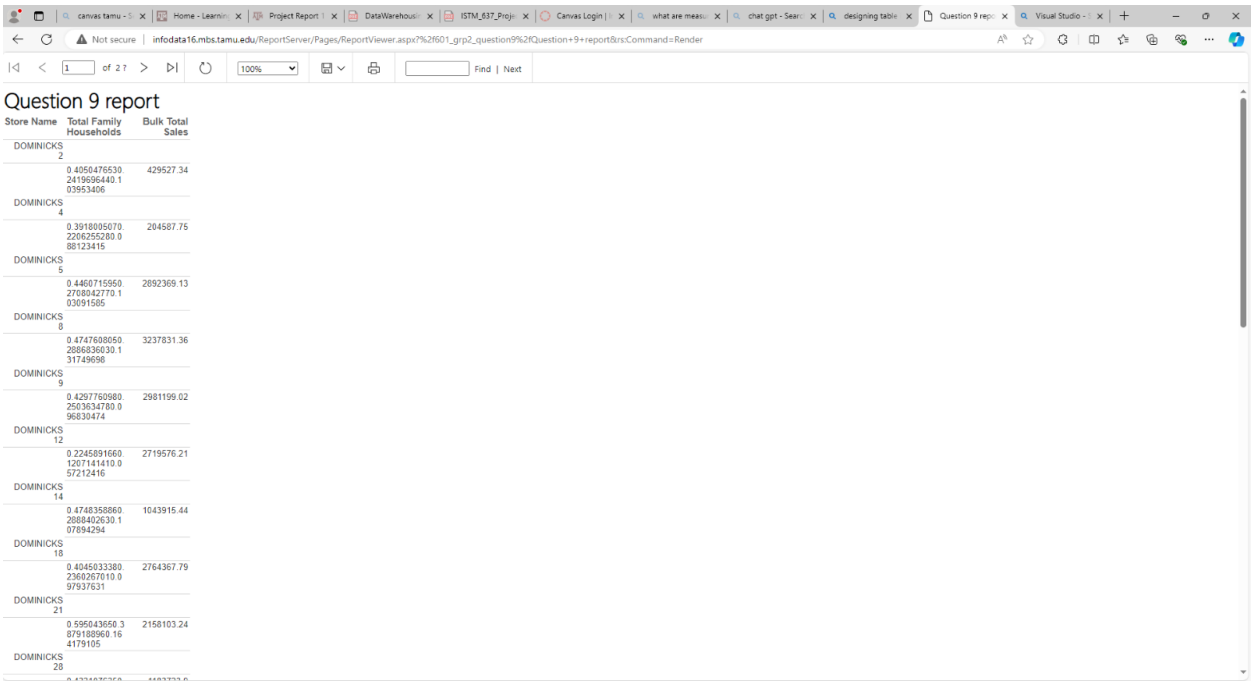
Step 19: Preview the report in the design tab



Step 20: To publish the report the server, change TargetServerURl in the property pages of the project



Step 21: Report is successfully deployed on target server

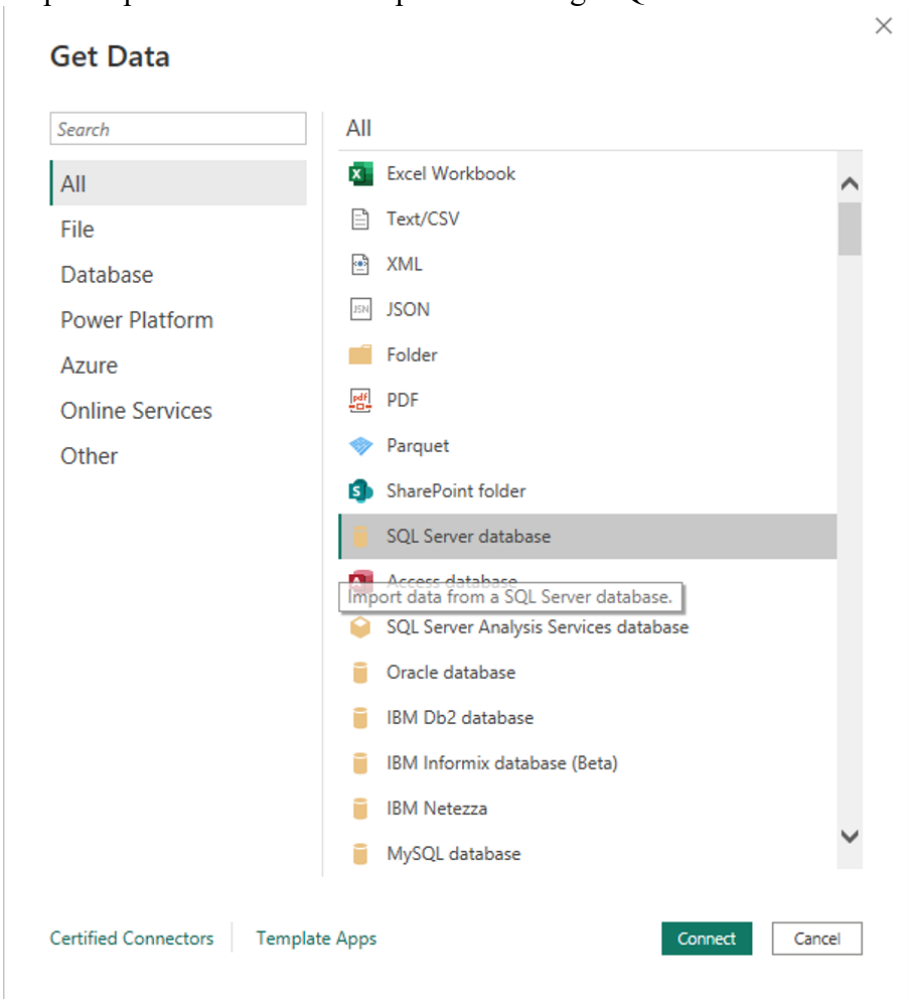


3. **Which are the top 7 product categories that saw the highest selling record over the 1 year(1995)?**

Rationale: Identifying the top 7 product categories with the highest sales over the year 1995 provides valuable insights into consumer purchasing behavior and trends. This information helps Dominick's Finer Foods (DFF) optimize inventory management, promotional strategies, and shelf space allocation for high-demand products. Analyzing these categories can also reveal seasonal patterns or recurring demand, enabling more accurate sales forecasting and better alignment of stock levels with customer preferences, ultimately enhancing profitability and customer satisfaction.

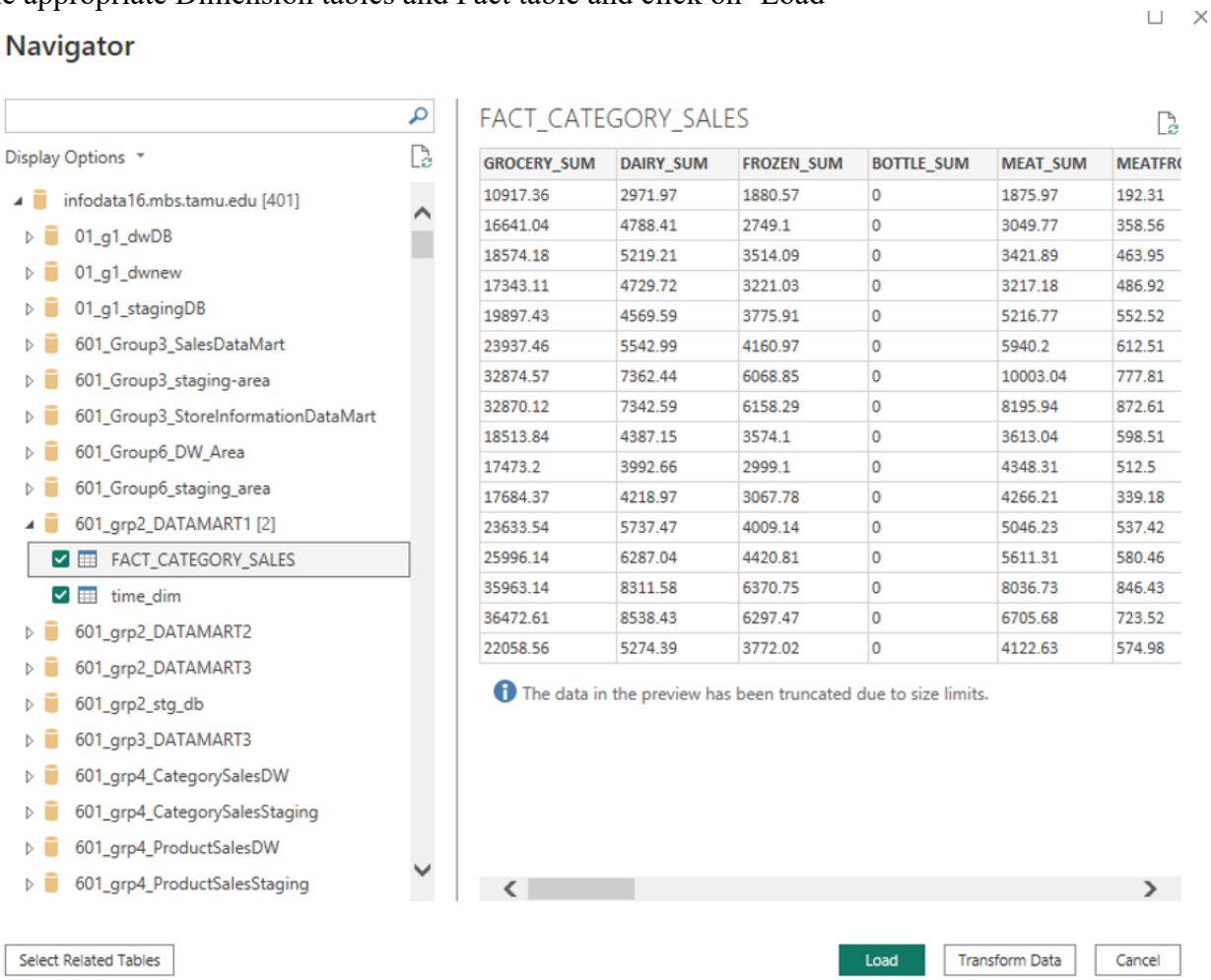
Reporting Tools Used: Power BI Only

Step 1: Open Power BI and import data using 'SQL Server Database' as source

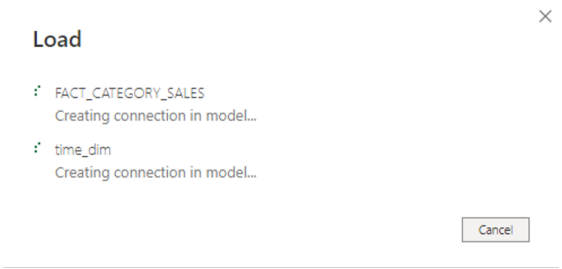


Step 2: Set the connection

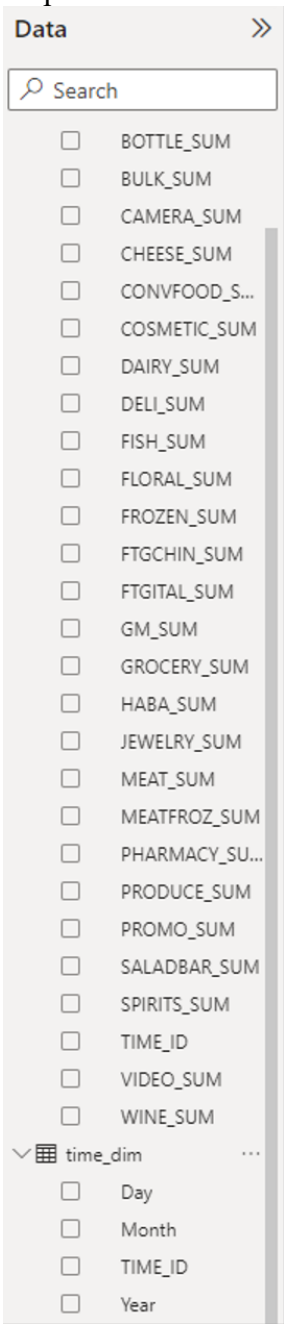
Step 3: Select the appropriate Dimension tables and Fact table and click on 'Load'



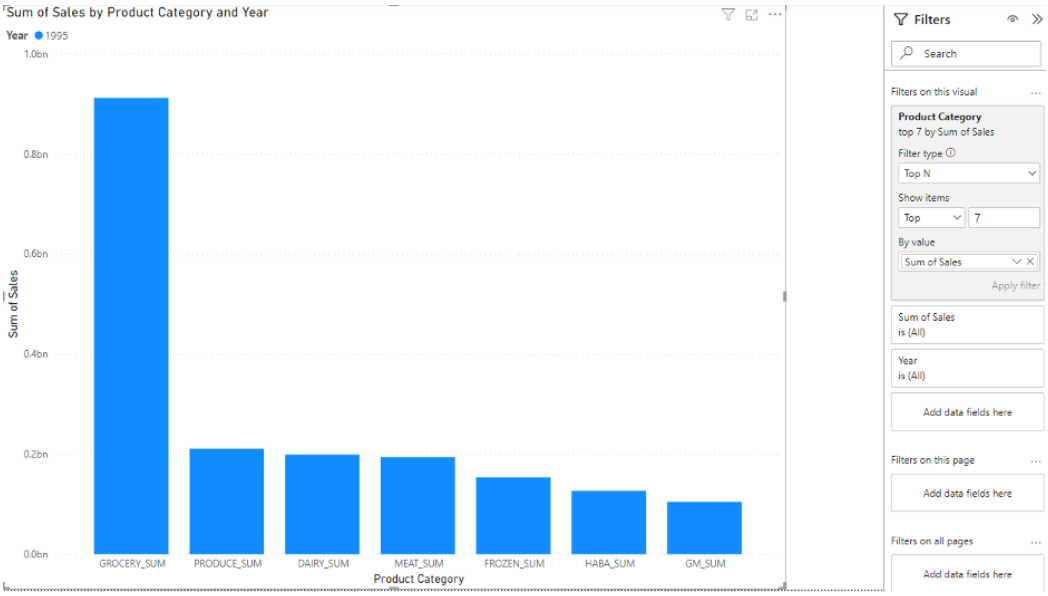
Step 4: Allow the tables to load



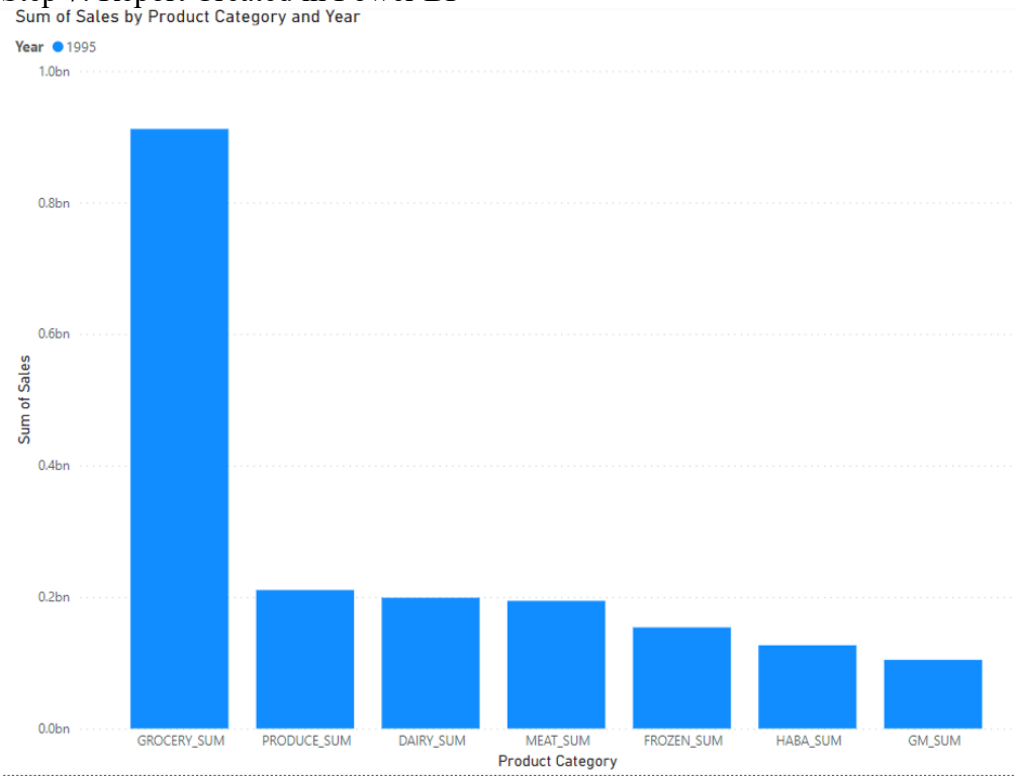
Step 5: Tables and attributes are loaded into Power BI



Step 6: Select the fields that are required and set the filters (Top 7 Product Categories by Sum of Sales)



Step 7: Report Created in Power BI

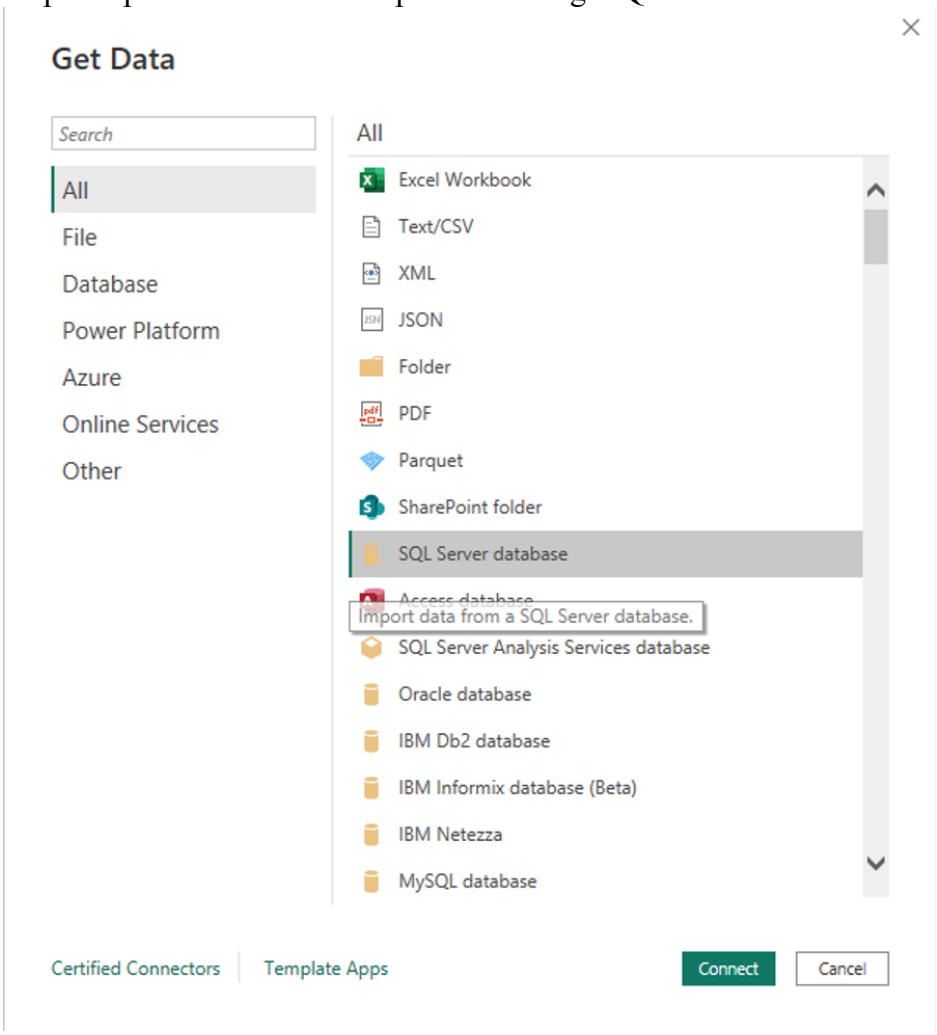


4. **Which stores have the lowest sales of soft drinks and the highest percentage of single-person households?**

Rationale: This question is designed to identify stores where specific demographic factors, such as a high percentage of single-person households, may correlate with lower sales of soft drinks. By analyzing this, DFF can uncover insights into consumer preferences and purchasing behavior in these areas. The findings can help target marketing strategies or adjust product assortments to better align with the needs and habits of single-person households, ultimately optimizing store performance and customer satisfaction.

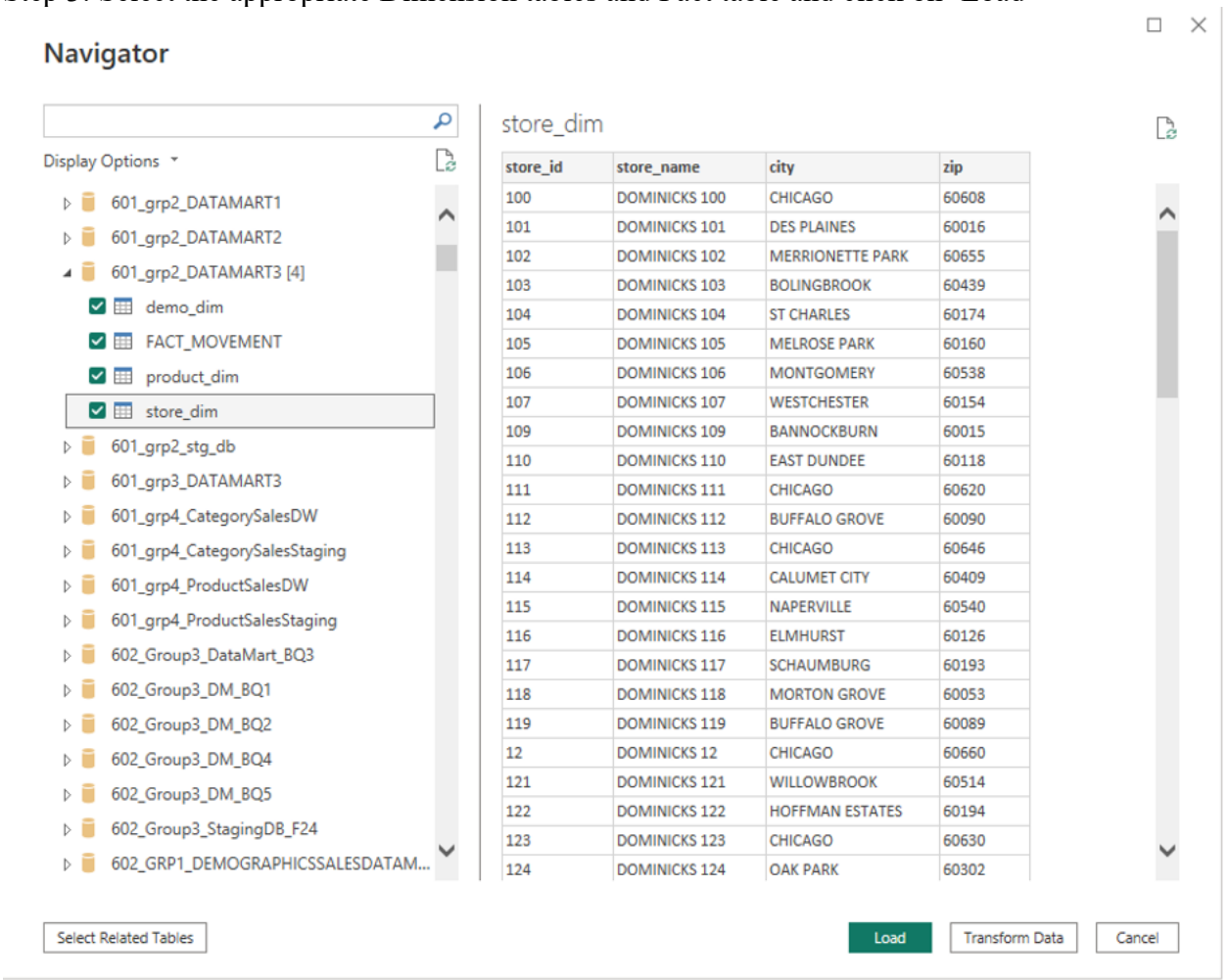
Reporting Tools Used: Power BI Only

Step 1: Open Power BI and import data using 'SQL Server Database' as source

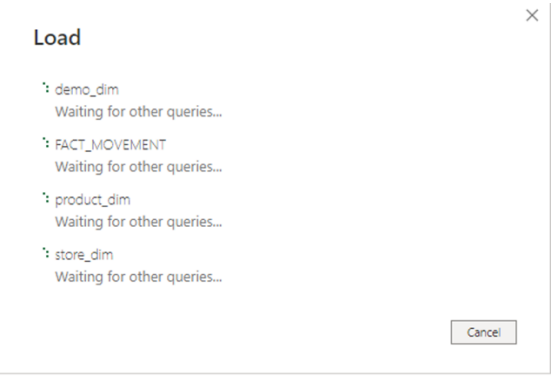


Step 2: Set the connection

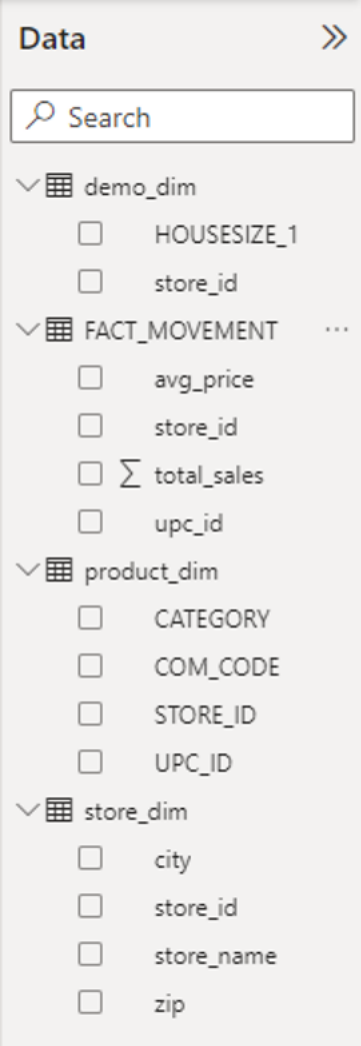
Step 3: Select the appropriate Dimension tables and Fact table and click on 'Load'



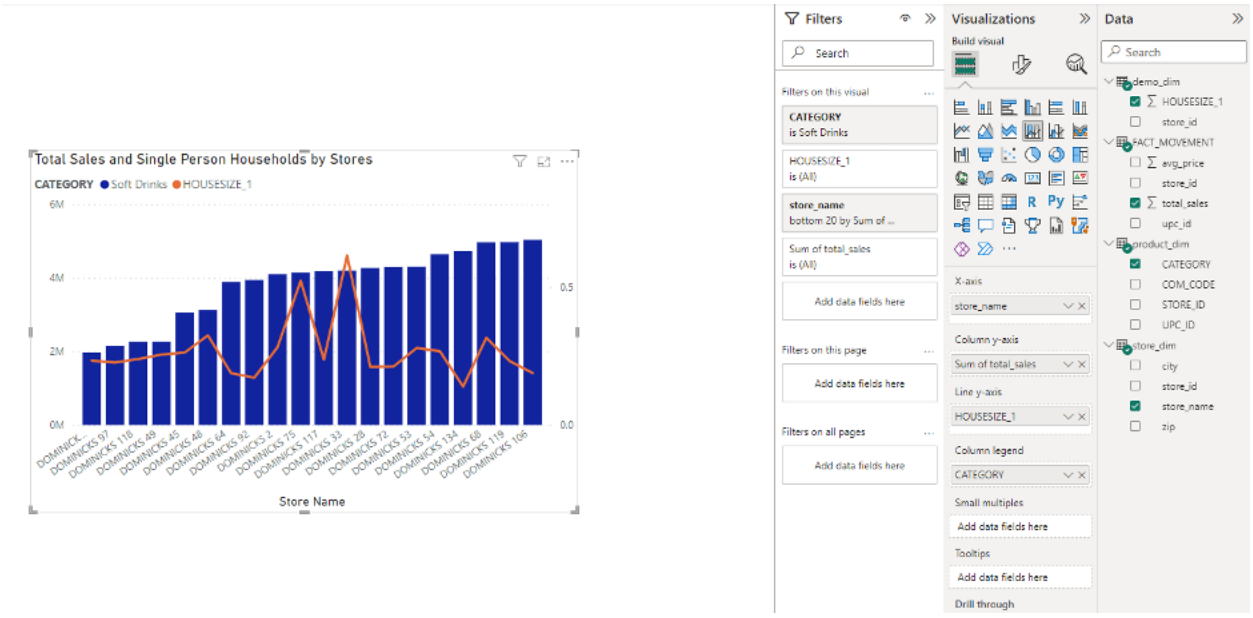
Step 4: Allow the tables to load



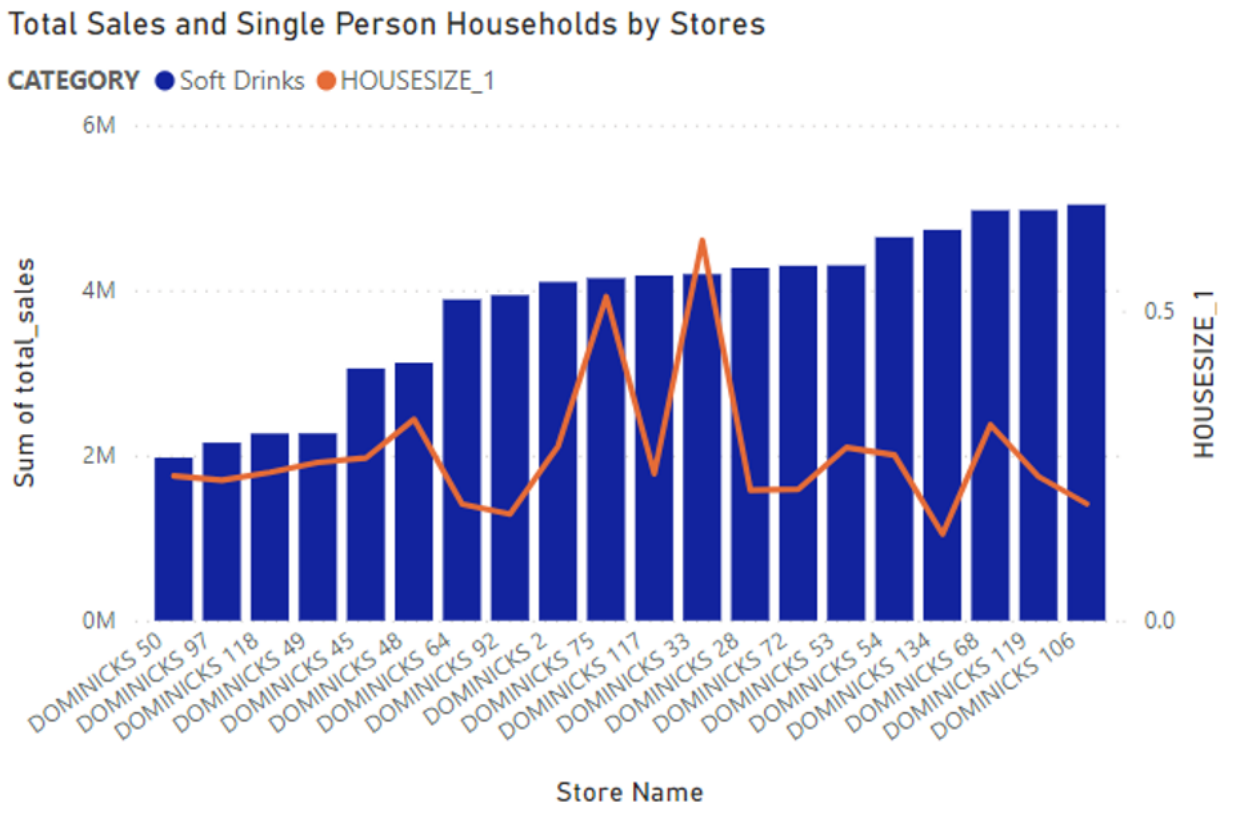
Step 5: Tables and attributes are loaded into Power BI



Step 6: Select the fields that are required to create the visualization and set the filters



Step 7: Report is created

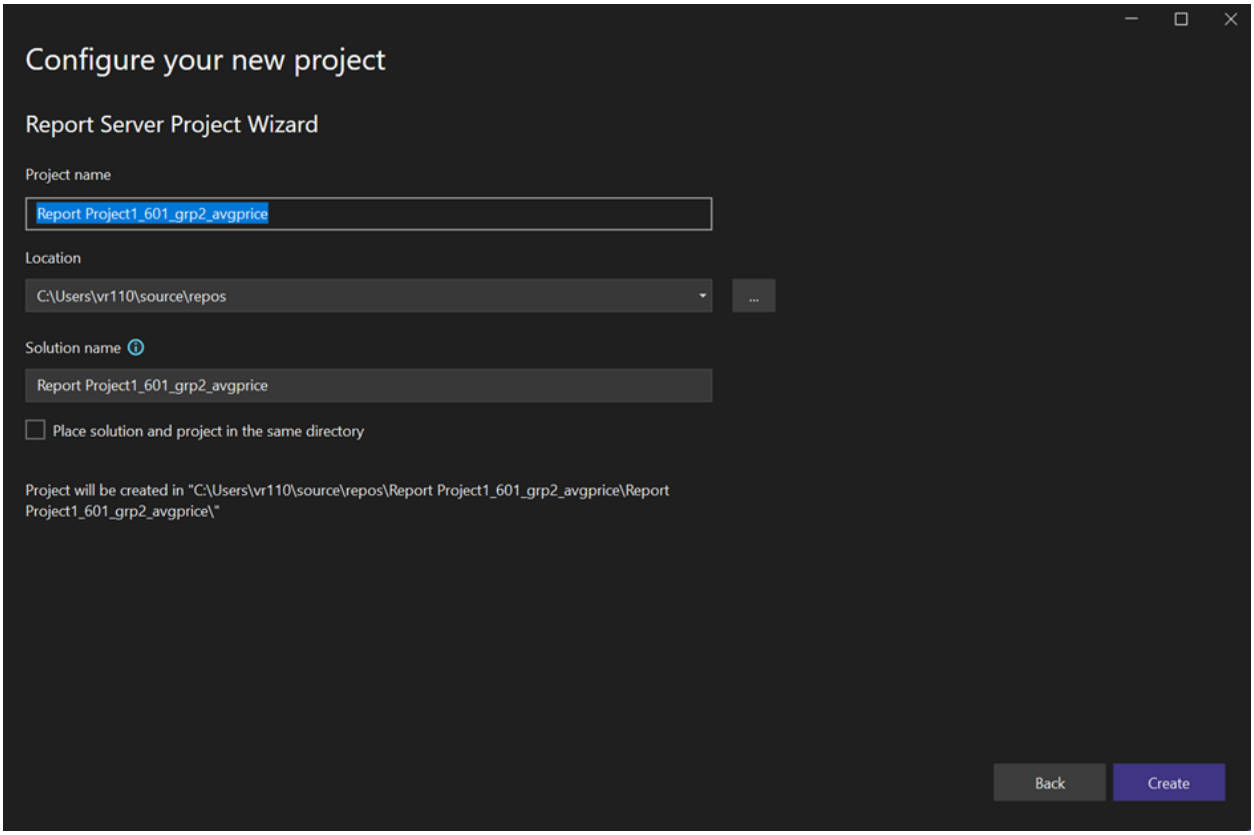


5. **What is the average price of frozen dinners for specific UPCs across all stores?**

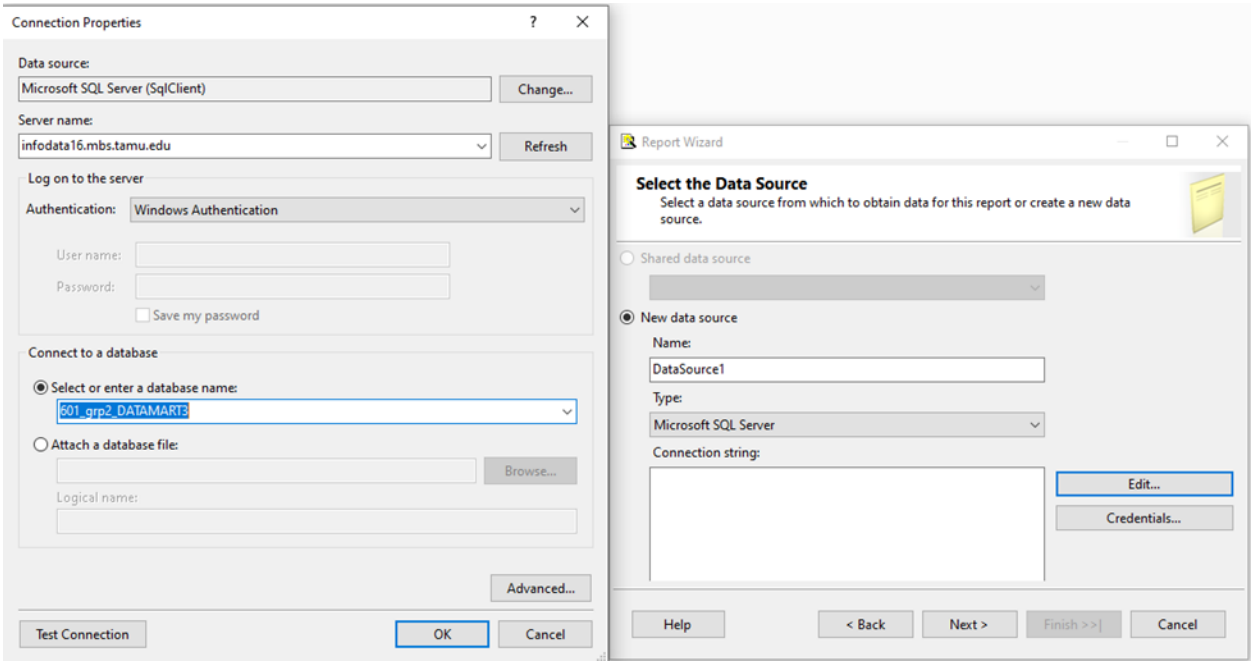
Rationale: By analyzing the average price, DFF can identify regional pricing discrepancies, optimize profit margins, and ensure competitive pricing across different locations. This data also aids in making informed decisions on supplier negotiations, sales forecasting, and promotional strategies, helping to maintain profitability and customer satisfaction while staying aligned with market demands and trends**.**

Reporting Tools Used: SSRS Only

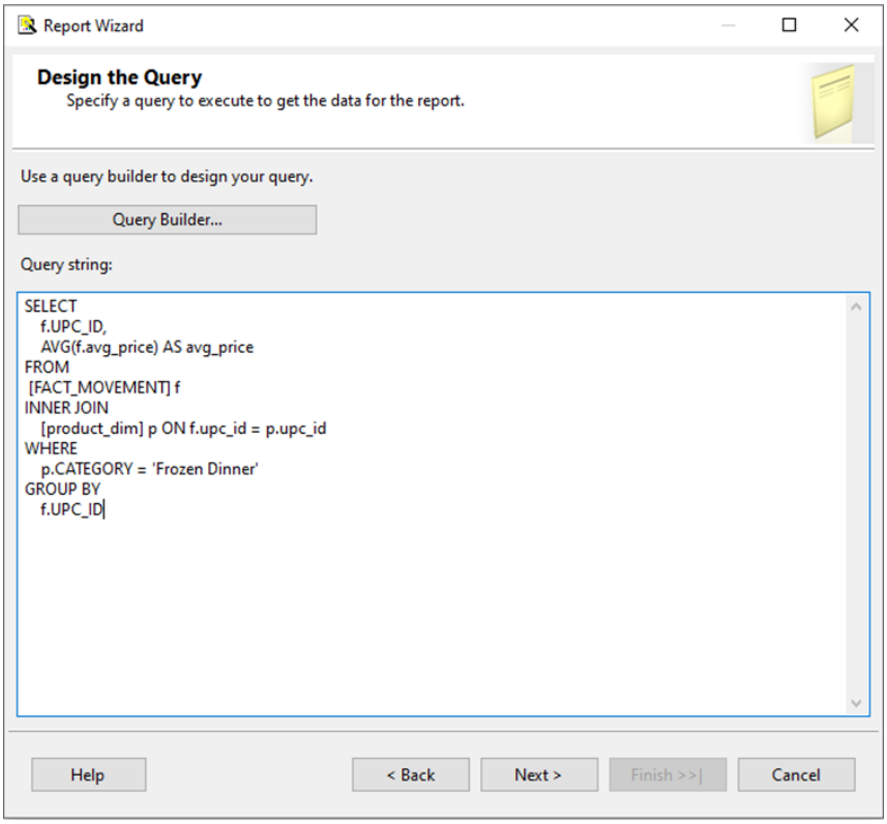
Step 1: Create New Reporting Service Project using the Report Server Project Wizard



Step 2: Selecting data source by Setting connection with infodata16.mbs.tamu.edu (SSMS) and choose database- 601_grp2_DATAMART3



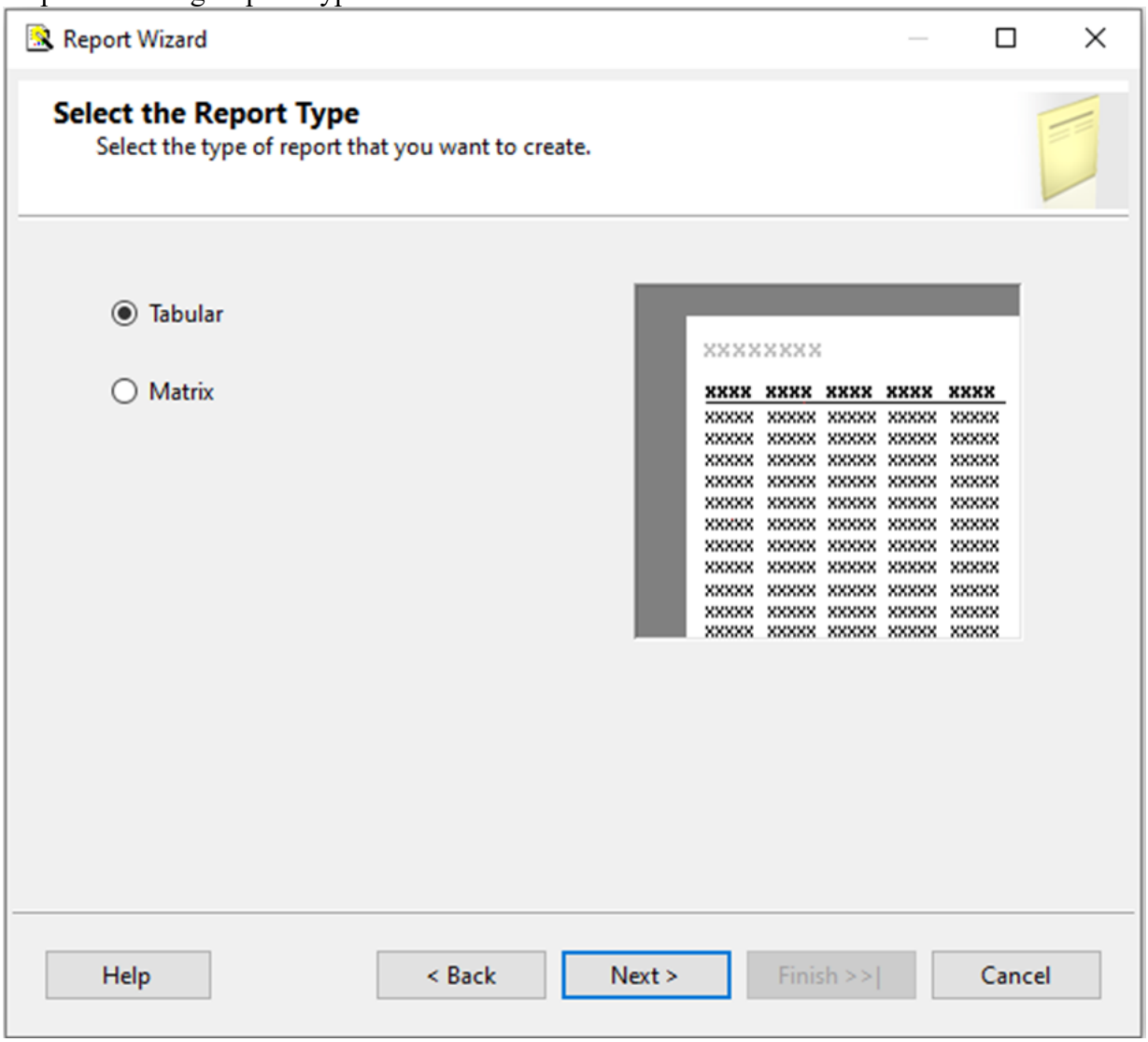
Step 3: Write SQL Query using the Query Builder



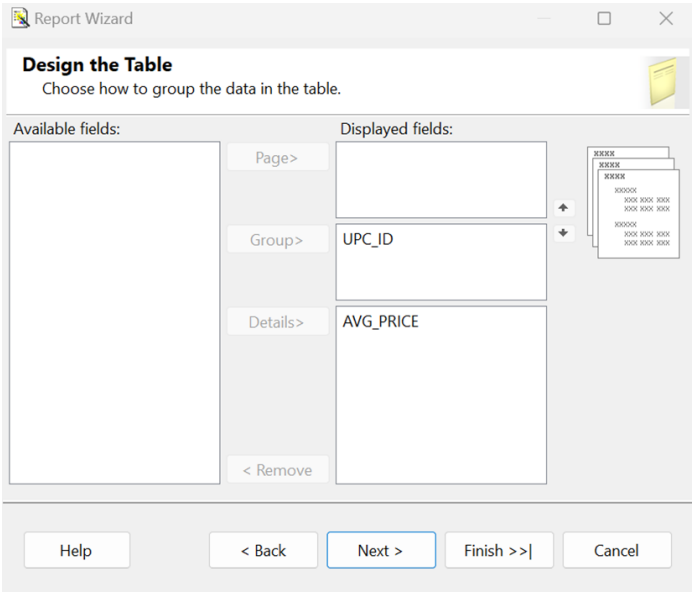
SQL Query:

```
SELECT
p.UPC_ID,
AVG (f.avg_price) AS avg_price
FROM
[FACT_MOVEMENT] f
INNER JOIN
[product_dim] p ON f.upc_id = p.upc_id
WHERE
P.CATEGORY = 'Frozen Dinner'
GROUP BY
F.UPC_ID
```

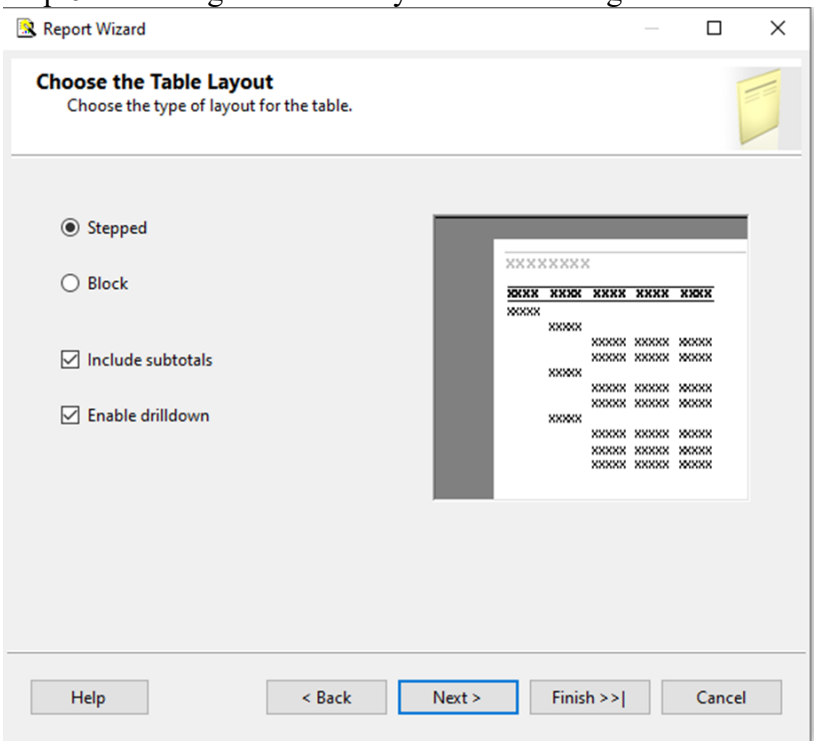
Step 4: Selecting Report Type:



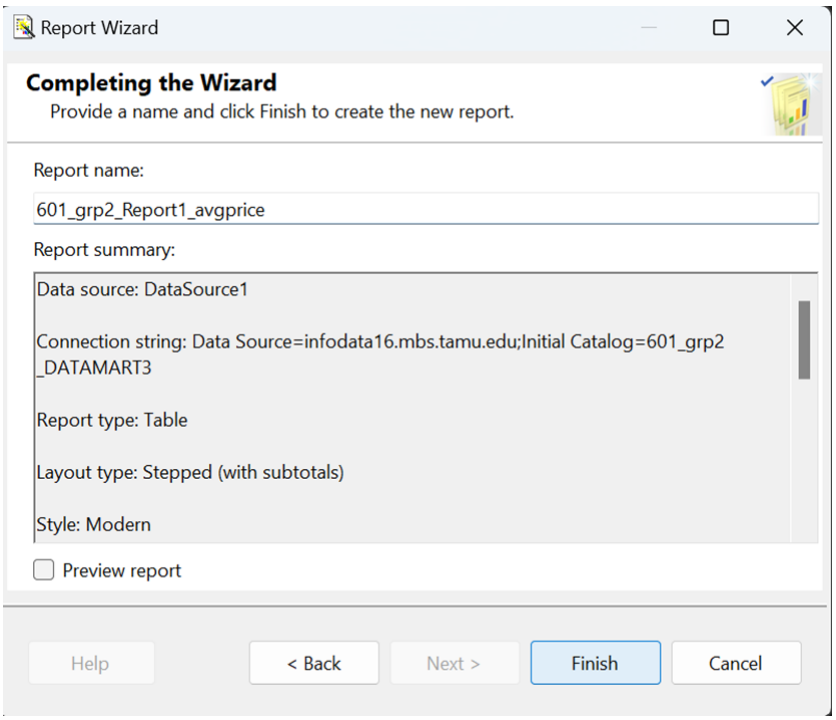
Step 5: Designing the table by dragging the UPC_ID in the heading (Group) and Avg_price in the Content (Details)



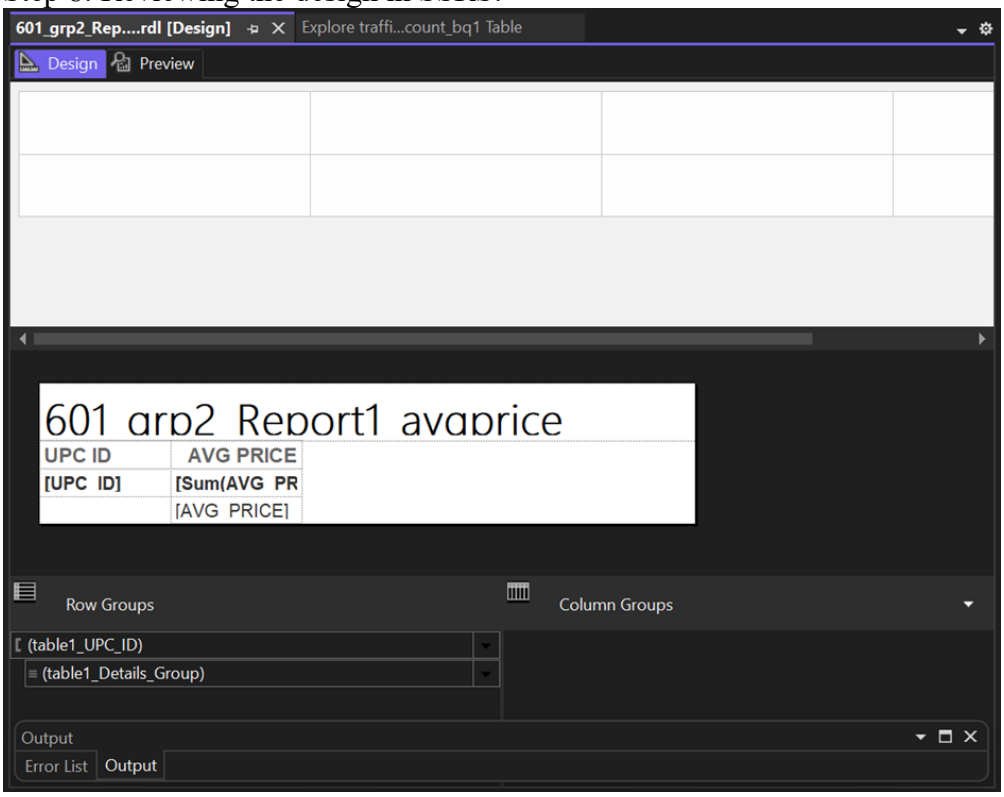
Step 6: Choosing the Table Layout and enabling subtotals and drilldown:



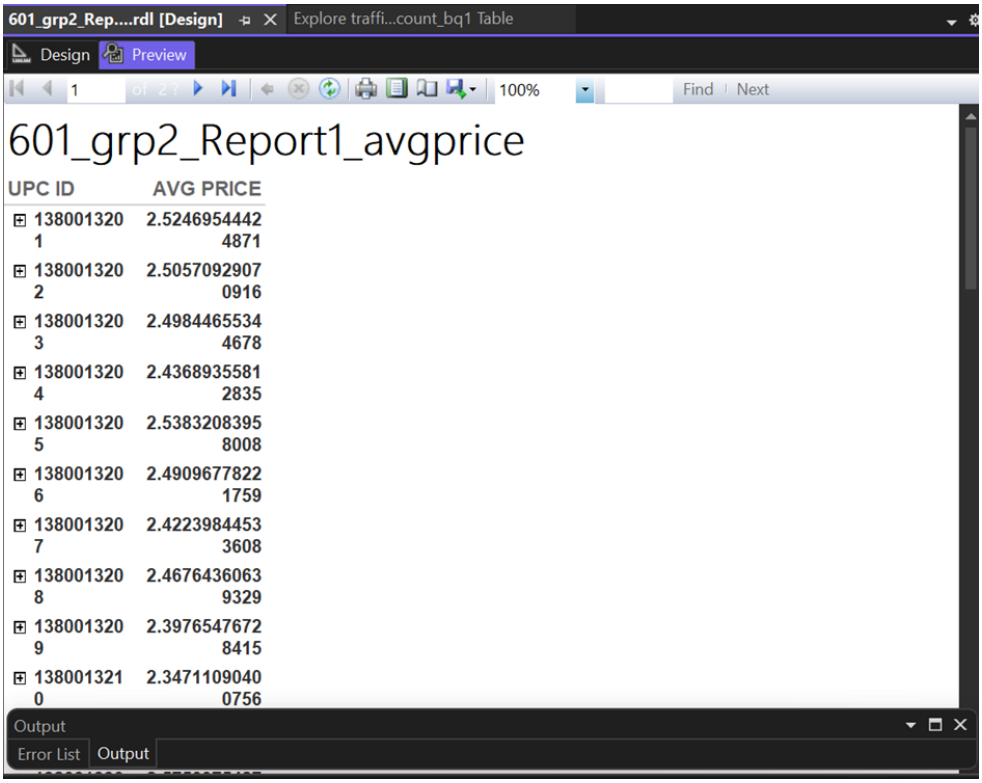
Step 7: Validating Report and Completing the Wizard:



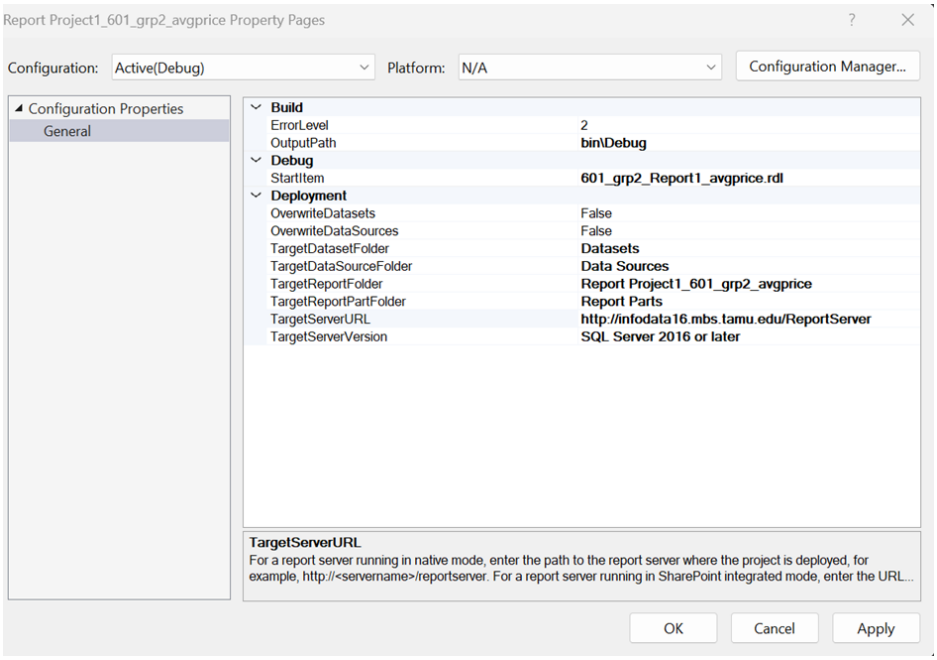
Step 8: Reviewing the design in SSRS:



Step 9: Report Preview:



Step 10: In the properties, changing the TargetServerURL to http://infodata16.mbs.tamu.edu/reportServer



Step 11: Successful Deployment in the Target Server:

Report Name: Report_Project1_601_grp2_avgprice

```
      Monday, February 5, 2024 7:51 PM       <dir> Report Project_OLAP_valeriep09
      Sunday, February 4, 2024 1:15 PM       <dir> Report Project_Week8Assignment
    Tuesday, November 7, 2023 11:45 AM       <dir> Report Project1
     Sunday, February 11, 2024 8:45 PM       <dir> Report Project1 online
     Friday, November 29, 2024 9:28 PM       <dir> Report Project1_601_grp2_avgprice
    Tuesday, November 28, 2023 3:55 AM       <dir> Report Project10
    Saturday, November 2, 2024 4:51 AM       <dir> Report Project1-11022024
  Wednesday, November 20, 2024 4:13 PM       <dir> Report Project-11202024-arunsen
  Wednesday, November 20, 2024 2:31 PM       <dir> Report Project1125_601
  Wednesday, November 20, 2024 2:31 PM       <dir> Report Project1125_601_tanay
      Monday, November 4, 2024 3:10 PM       <dir> Report Project1-601-11042024
   Wednesday, November 6, 2024 2:46 PM       <dir> Report Project1-601-11062024-PVerma
   Wednesday, November 6, 2024 2:46 PM       <dir> Report Project1-601-11062024-sen
  Thursday, November 30, 2023 11:26 PM       <dir> Report Project19
```

Report in target server: