Design and Implementation of a Data Warehouse for a Retail Store

Report 4

BI Report Design and Implementation for DFF along with the Integrated Group Report

Table of Contents

Section 1. Introduction

This project delves into the history of Dominick's Finer Foods, a prominent Chicago grocery chain. Established in 1918 by Dominick di Matteo, an Italian immigrant, the company rose to prominence in 1950 when it expanded into complete supermarkets, renowned for its innovative frozen food sections and in-store delicatessens. By the 1980s, Dominick's was a significant player, a contending market leader renowned for its consumer-focused strategies.

However, the late 1990s brought significant challenges due to changes in ownership, which posed difficulties in retaining market share and ensuring customer satisfaction. Despite initial success, Dominick encountered challenges pertaining to pricing, product selection, and the removal of customized services, all of which contributed to a reduction in the company's market presence.

In the late 1980s to mid-1990s, Dominick's collaborated with the University of Chicago Booth School of Business to advance store-level research. This research centered on randomized experiments conducted throughout the 100-store chain of Dominick's to examine effective shelf management and pricing strategies. Approximately nine years of store-level data pertaining to the sales of more than 3,500 UPCs became available as a beneficial outcome of this research collaboration.

The project aims to utilize this store-level data to extract valuable insights into the dynamics of the grocery retail industry. The dataset comprises six tables, encompassing a wide range of information related to sales, products, and demographics. However, the considerable size and diverse data formats of these files can pose challenges for analysis. Establishing meaningful relationships between these files is another complex task due to data quality issues, which might hinder the retrieval of reliable and accurate insights.

The report provides a detailed analysis aimed at answering crucial business questions that offer insights into customer preferences, sales data, and shopping behavior in order to identify key performance indicators. These insights can then serve as a guide for Dominick's Finer Foods in making informed, data-driven decisions that have the potential to enhance sales and profitability.

Section 2. Understanding of the Data

One of the essential parts of this report is understanding the raw data we have at our disposal. We will list the files in this report to better display the information.

Data Source

The store-level scanner data captured at Dominick's Finer Foods over more than seven years is included in Dominick's dataset. We have two categories of files: 1. General Files and 2. Category-specific files are both included in the dataset.

Data Description

General Files

1. The Customer Count File
    a. The customer count file includes information about in-store behavior. It refers to the number of customers buying products at a specific store. This file also contains the total sales value, including coupons redeemed.

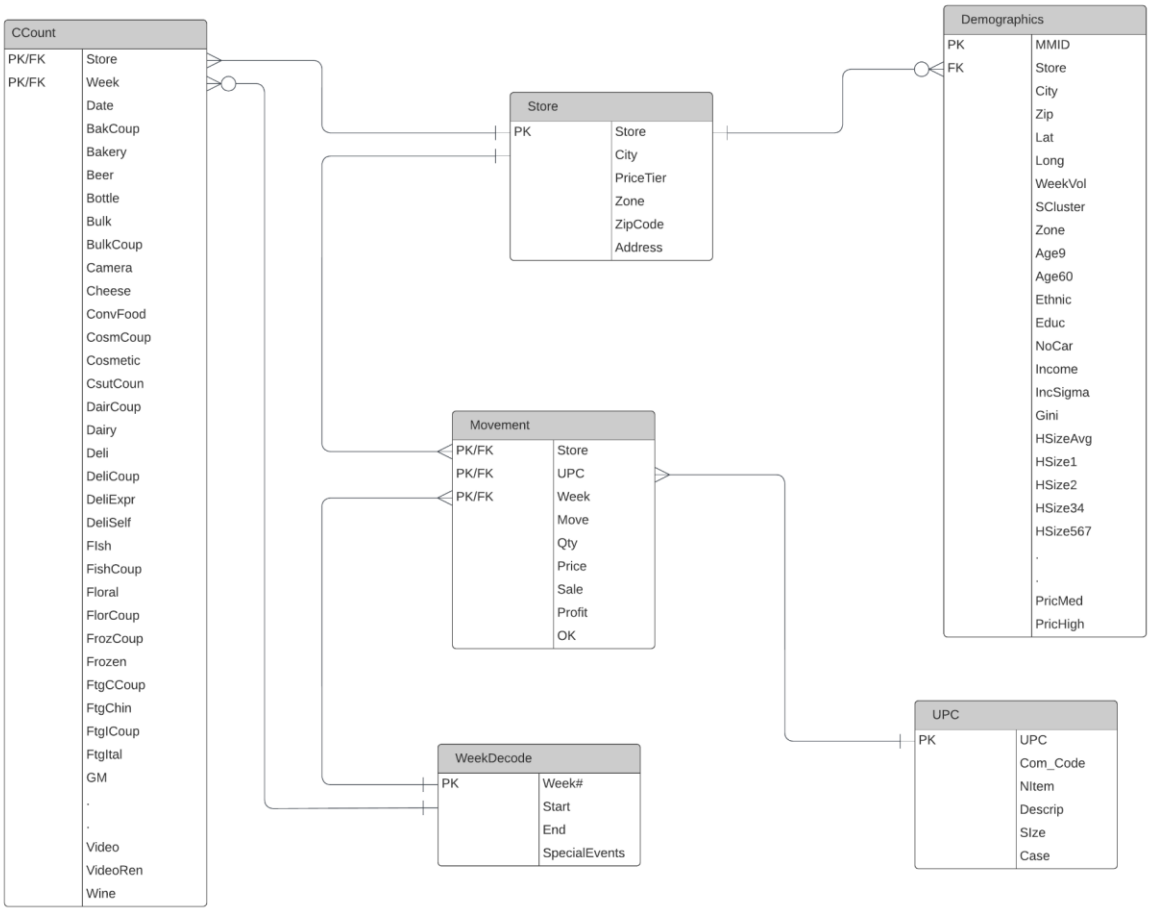
2. The Store-Level Demographics File
    a. This file contains census data from the US government. Includes the Chicago metropolitan area. This data contains demographic profiles, which can be linked to store-specific data.

Category Specific Files

1. The UPC files
    a. The Universal Product Code (UPC) files contain one record for each UPC in a category; files are named with the category acronym. They have information about product name, size, commodity code, etc. (Chicago Booth, 2011)

2. The movement files
    a. The movement files contain weekly sales data for each UPC in each store for over five years. The variables included in these files comprise price, unit sold, profit margin, deal code, etc. (Chicago Booth, 2011)

Entity Relationship (ER) Diagram



Section 3. Business Questions

Selected Business Questions (BQ)

BQ 1

Which are the top 5 categories for products across different stores between the years 1980 to 1997?

Values	Years 1980	1987	1988	1989	1990	1991	1992	1993	1994	1996	1997	Grand Total
Sum of BOTTLE	$0.00	$0.00	$85,684.98	$200,809.59	$71,850.45	$38,560.86	$146,149.94	$195,052.42	-$8,668.95	$2,004.32	$23,792.50	$755,236.11
Sum of BEER	$545.52	$13,233.77	$33,730,113.69	$20,922,872.25	$22,670,229.17	$26,194,510.51	$27,829,738.59	$28,643,916.19	$30,179,810.46	$34,890,088.41	$9,457,990.13	$234,533,048.69
Sum of WINE	$1,365.54	$0.00	$1,674,788.85	$13,086,642.23	$14,875,630.52	$16,019,131.14	$18,408,560.45	$18,831,875.95	$20,162,115.58	$24,690,145.67	$9,288,865.48	$137,039,121.41
Sum of CAMERA	$0.00	$0.00	$1,481,564.30	$1,544,838.38	$1,742,020.24	$2,037,994.35	$1,902,048.53	$2,196,794.62	$2,702,151.08	$4,057,030.77	$1,453,730.75	$19,118,173.02
Sum of SPIRITS	$537.14	$0.00	$1,775,608.93	$10,329,071.50	$12,070,396.78	$12,501,200.24	$14,256,839.76	$14,298,140.02	$14,132,063.85	$15,566,284.18	$4,552,700.09	$99,482,842.49
Sum of COSMETIC	$0.00	$0.00	$2,793,802.15	$4,540,557.01	$5,888,238.13	$6,719,181.00	$7,296,986.16	$6,951,340.65	$6,592,177.30	$5,287,156.35	$1,662,694.30	$47,732,133.05
Sum of BAKERY	$1,948.61	$4,592.18	$37,420,096.72	$45,280,061.73	$47,022,423.57	$48,904,591.75	$52,591,025.21	$54,028,874.41	$54,536,231.36	$56,894,129.81	$18,802,321.33	$415,486,296.68
Sum of JEWELRY	$0.00	$0.00	$611,637.99	$980,652.36	$966,107.84	$922,553.68	$419,812.90	$1,058,747.05	$160,628.64	$614,762.28	$8.70	$5,734,911.44
Sum of CHEESE	$995.25	$0.00	$4,686,111.52	$7,782,832.72	$8,737,232.88	$9,466,306.12	$10,739,618.58	$12,736,015.60	$12,909,413.29	$12,922,467.31	$4,132,094.41	$84,113,087.68
Sum of PHARMACY	$0.00	$0.00	$6,594,730.15	$10,371,910.77	$15,146,955.05	$21,762,319.83	$31,162,555.89	$40,906,207.48	$47,947,360.44	$63,443,242.10	$23,070,362.94	$260,405,644.65
Sum of CONVFOOD	$221.52	$0.00	$481,600.63	$4,434,897.46	$4,096,065.09	$4,476,420.83	$4,746,924.54	$4,973,600.26	$5,278,915.90	$3,886,006.29	$1,108,599.91	$33,483,252.43
Sum of DELI	$2,478.29	$17,463.66	$128,238,886.39	$77,854,830.44	$72,920,829.06	$76,903,238.74	$81,759,471.44	$92,466,578.50	$87,975,514.75	$91,526,918.39	$30,535,994.51	$740,202,204.17
Sum of GROCERY	$28,295.18	$115,573.67	$863,962,035.50	$824,551,230.89	$842,013,772.53	$901,611,105.12	$895,953,628.02	$897,045,701.64	$872,692,597.35	$922,328,335.95	$324,162,494.89	$7,344,464,770.74
Sum of MEAT	$6,955.11	$28,062.32	$194,781,008.07	$200,998,662.95	$205,718,922.81	$214,747,092.12	$211,983,261.00	$207,473,624.43	$201,025,816.71	$200,349,886.63	$69,878,855.94	$1,706,992,148.09
Sum of FLORAL	$745.06	$461.73	$9,556,123.43	$12,252,905.87	$13,171,667.45	$14,360,989.36	$16,267,077.00	$19,701,425.60	$18,980,309.86	$24,848,678.19	$9,529,789.26	$138,670,172.81
Sum of FROZEN	$4,735.50	$0.00	$30,316,926.54	$126,919,968.05	$140,919,335.24	$150,023,820.78	$152,037,367.31	$151,383,395.65	$151,992,829.85	$152,980,244.48	$56,997,179.56	$1,113,575,802.96
Sum of DAIRY	$7,214.63	$0.00	$38,534,261.87	$169,965,548.07	$191,872,716.81	$196,141,588.66	$203,928,125.35	$196,812,464.27	$194,215,282.65	$207,521,630.47	$76,671,371.10	$1,475,670,203.88
Sum of SALADBAR	$1,517.23	$0.00	$10,463,940.61	$12,347,731.68	$12,088,850.27	$11,534,443.17	$10,908,185.64	$11,153,529.64	$10,099,834.66	$10,320,436.86	$3,354,693.38	$92,273,163.14
Sum of FISH	$1,564.89	$3,055.23	$21,767,879.18	$22,730,910.05	$22,982,329.01	$23,502,128.80	$25,668,019.29	$29,578,427.56	$27,731,622.79	$29,699,456.05	$11,659,139.74	$215,324,532.59

Figure: Data Table

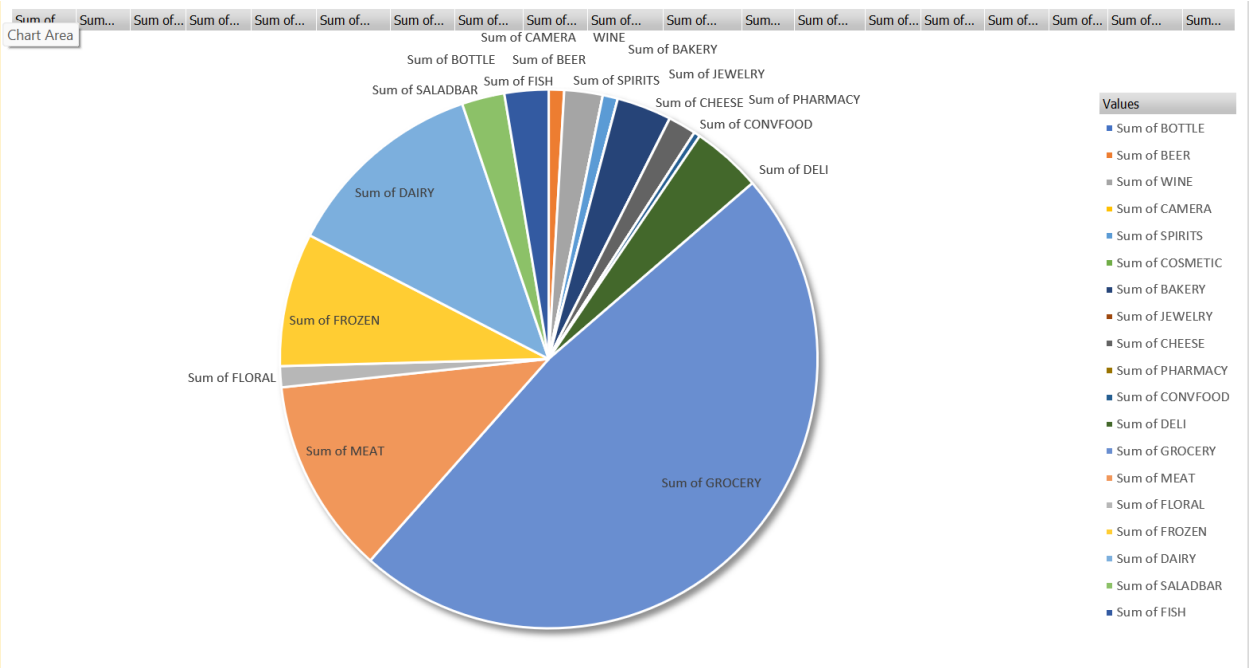


Figure: Data Chart

Assumption: We're only assuming the stores which are present in the stores table.

Observation: The top ranking products/category of products are - Grocery, Meat, Dairy, Frozen, and Deli.

Rationale: Understanding the top 5 product categories across different stores is crucial for several reasons. Amongst the most important are sales and revenue optimization, customer satisfaction, inventory management, and better space utilization. When an organization identifies the most essential products, it can allocate resources towards reducing promotional activities or diversifying its brand portfolio within a given category.

BQ 2

What are the year-over-year trends and patterns in alcohol (Beer, Wine, Spirit) sales from 1987 to 1997?

Years	Sum of BEER	Sum of WINE	Sum of SPIRITS
1987	$13,233.77	$0.00	$0.00
1988	$33,730,113.69	$1,674,788.85	$1,775,608.93
1989	$20,922,872.25	$13,086,642.23	$10,329,071.50
1990	$22,670,229.17	$14,875,630.52	$12,070,396.78
1991	$26,194,510.51	$16,019,131.14	$12,501,200.24
1992	$27,829,738.59	$18,408,560.45	$14,256,839.76
1993	$28,643,916.19	$18,831,875.95	$14,298,140.02
1994	$30,179,810.46	$20,162,115.58	$14,132,063.85
1995	$32,729,201.96	$22,058,673.43	$15,088,746.75
1996	$34,890,088.41	$24,690,145.67	$15,566,284.18
1997	$9,457,990.13	$9,288,865.48	$4,552,700.09
Grand Total	**$267,261,705.13**	**$159,096,429.30**	**$114,571,052.10**

Figure: Data Table

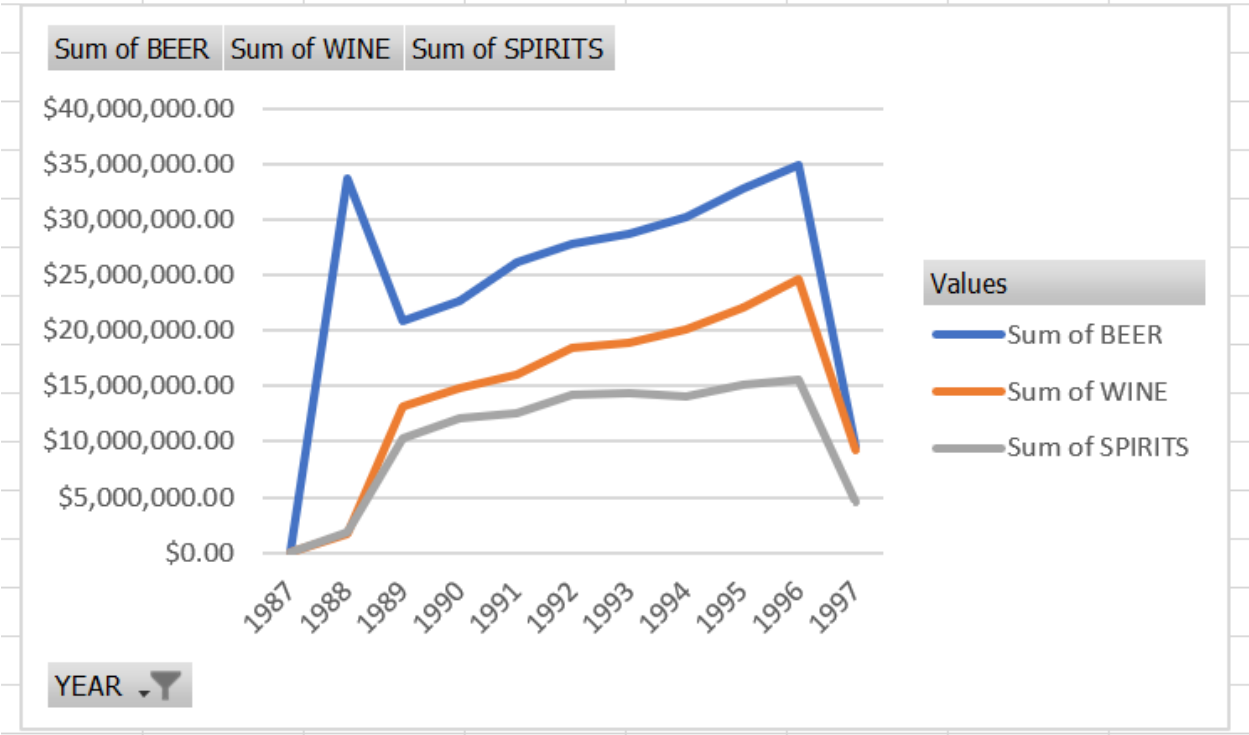


Figure: Line Chart

Observation: The sales trend indicates that beer has consistently ranked first in the alcohol category over the years.

Rationale: Analyzing past sales trends of various alcoholic beverages can offer valuable insights into changing consumer preferences over different years. In the context of DFF, this information emphasizes a consistent trend - their stores consistently exhibit the highest demand for beer among various alcoholic beverages. This knowledge empowers companies to refine marketing and product strategies, enhancing their competitiveness and adaptability to consumer needs.

BQ 3

What are the annual sales patterns for wine during peak seasons in the United States?

WINE Sales	Special Events									
Years	4th of July	Christmas	Easter	Halloween	Labor Day	Memorial Day	New-Year	Presidents Day	Thanksgiving	Grand Total
1989		687024.94		251312.5			561489.93		299353.29	1799180.66
1990	366999.21	867237.47	238864.63	270119.48	305149.09	299486.42	775648.75	249739.96	304045.23	3677290.24
1991	258058.87	894678.26	420812.06	266221.18	303508.53	355208.11	758167.16	271454.19	347778.3	3875886.66
1992	426755.94	720824.83	306440.04	323571.27	370662.21	386127.7	379612.03	302588.99	383512.63	3600095.64
1993	470182.18	790082.23	304045.19	304527.01	385537.53	433926.3	772801.68	308704.82	409375	4179181.94
1994	529494.48	980689.58	484375.61	349089.85	459822.2	504018.24	932427.37	317499.23	455226.77	5012643.33
1995	576608.31	1066866.67	362122.6	402388.1	448061.95	514153.44	976243.58	318945.36	466245.66	5131635.67
1996	431122.65	1387231.19	403082.37	324512.52	500926.19	539240.61	1168102.72	390774.02	546964.66	5691956.93
1997			564362.93				53547.02	436947.2		1054857.15
Grand Total	**3059221.64**	**7394635.17**	**3084105.43**	**2491741.91**	**2773667.7**	**3032160.82**	**6378040.24**	**2596653.77**	**3212501.54**	**34022728.22**

Figure: Data Table

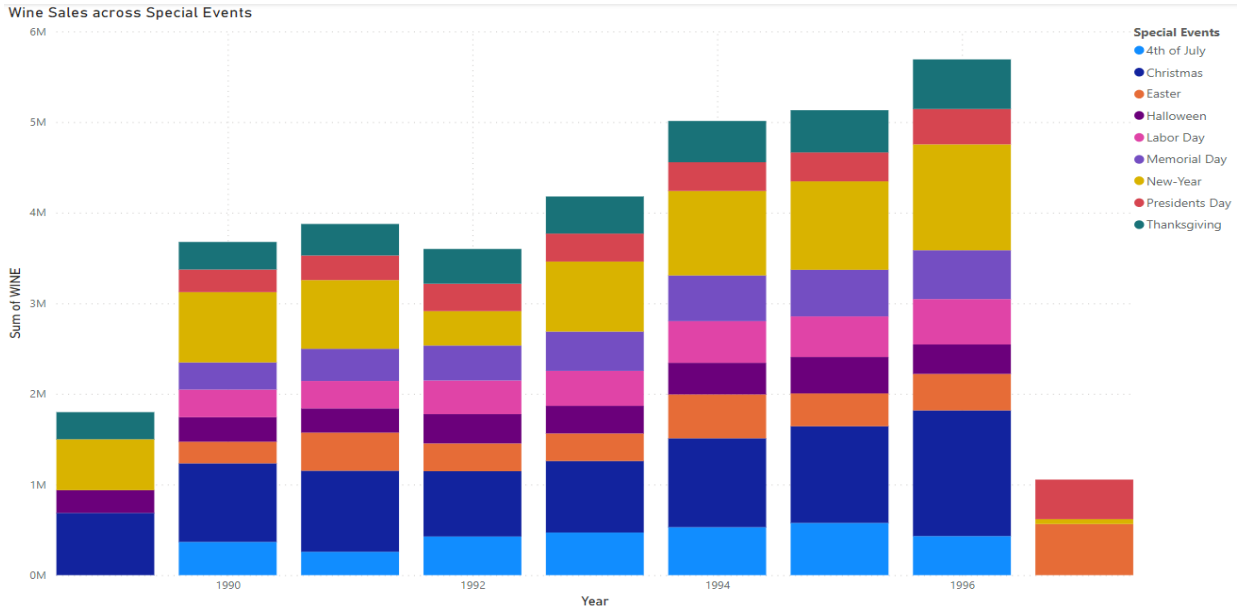


Figure: Bar Char

Observation: Most alcohol sales were observed in the year 1996 during the holiday seasons of Christmas and the New Year's.

Rationale: The US market has very strong sales behavior during specific times of the year. For example, Thanksgiving, Christmas, Memorial Day, the 4th of July, and the Back-to-School season, among others. Wine consumption can see a substantial increase in these peak seasons. DFF can capitalize on these trends to secure a competitive advantage and increase sales and customer base by taking proactive steps just before these specific periods. Additionally, this approach offers precise inventory management, ensuring that the store doesn't end up with excess unsold products on their shelves during slower sales seasons.

BQ 4

How does the percentage of avid, hurried, and strange shoppers vary across different cities, and how can we tailor marketing strategies accordingly?

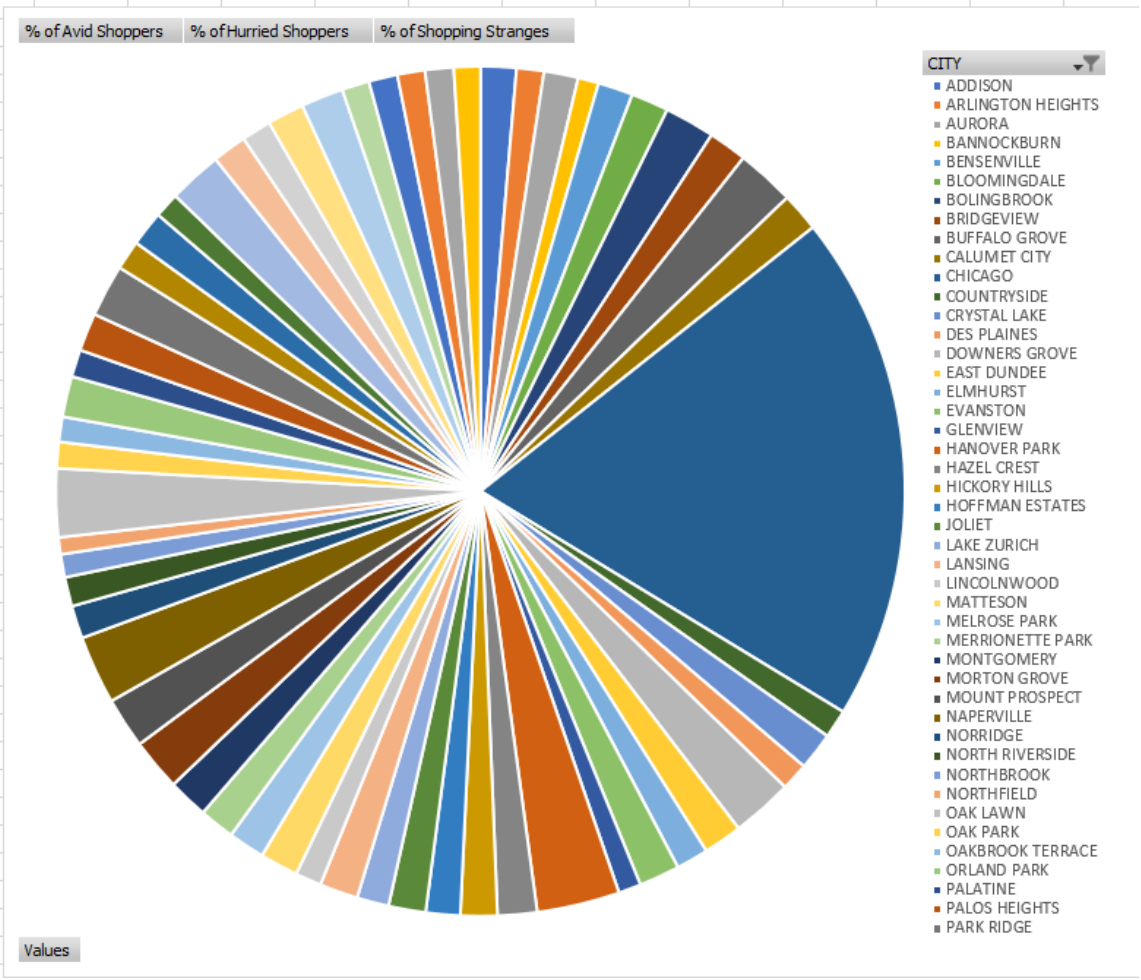


Figure: Pie Chart

Row Labels	% of Avid Shoppers	% of Hurried Shoppers	% of Shopping Stranges
ADDISON	1.36%	1.29%	0.312622118
ARLINGTON HEIGHTS	1.04%	1.40%	0.300879716
AURORA	1.30%	1.24%	0.29512982
BANNOCKBURN	0.79%	2.14%	0.217582858
BENSENVILLE	1.33%	1.10%	0.330180568
BLOOMINGDALE	1.43%	1.44%	0.358920287
BOLINGBROOK	1.91%	1.67%	0.275685662
BRIDGEVIEW	1.47%	1.00%	0.277556782
BUFFALO GROVE	2.22%	2.93%	0.684552931
CALUMET CITY	1.48%	0.71%	0.281401731
CHICAGO	19.36%	11.92%	5.015997734
COUNTRYSIDE	1.09%	1.27%	0.276860896
CRYSTAL LAKE	1.44%	1.82%	0.249688474
DES PLAINES	1.05%	1.02%	0.301388313
DOWNERS GROVE	2.33%	3.00%	0.647323215
EAST DUNDEE	1.48%	1.42%	0.264900662
ELMHURST	1.23%	1.36%	0.261970376
EVANSTON	1.55%	2.29%	0.646554197
GLENVIEW	0.86%	1.65%	0.216425446
HANOVER PARK	3.15%	3.48%	0.604308306
HAZEL CREST	1.50%	1.36%	0.241834378
HICKORY HILLS	1.40%	0.97%	0.32142039
HOFFMAN ESTATES	1.31%	1.63%	0.345028355
JOLIET	1.41%	0.98%	0.284918242
LAKE ZURICH	1.22%	2.02%	0.234881055
LANSING	1.46%	0.86%	0.270861169
LINCOLNWOOD	1.00%	1.18%	0.19057117
MATTESON	1.46%	1.32%	0.277539987
MELROSE PARK	1.39%	1.15%	0.228544284
MERRIONETTE PARK	1.35%	0.95%	0.237902546
MONTGOMERY	1.53%	1.35%	0.274316671
MORTON GROVE	1.98%	2.31%	0.447411625
MOUNT PROSPECT	1.93%	2.35%	0.612765645
NAPERVILLE	2.61%	3.65%	0.666400901
NORRIDGE	1.25%	0.77%	0.221238037
NORTH RIVERSIDE	1.11%	0.75%	0.268782953
NORTHBROOK	0.88%	1.61%	0.313467049
NORTHFIELD	0.64%	1.84%	0.18371703
OAK LAWN	2.60%	1.88%	0.463116159
OAK PARK	1.00%	0.72%	0.340653831
OAKBROOK TERRACE	0.95%	1.23%	0.318855932

Figure: Data Table

Observation: Chicago has observed the greatest population of avid, hurried, and strange shoppers throughout the years.

Rationale: Consumer behaviors and preferences can vary across different cities. By closely examining the shopping habits of distinct customer segments in different cities, DFF can tailor their marketing strategies more precisely. This data-driven approach enables us to align our tactics with the local consumer preferences specific to the stores in Chicago. Ultimately, this can lead to

more effective marketing campaigns and enhanced customer engagement, contributing to the store's success and growth.

BQ 5

What is the most popular coupon used for purchasing the top 20 juice categories?

Number of Coupons of Specific Type				
Row Labels	B	C	S	Grand Total
$INDIAN SUMMMER APPL	1455		14	1469
~OS REFRESHERS CITRU	1297		135	1432
~TROP TWISTER ORG/ST	1245		127	1372
DOLE PURE & LITE PEA	1748		78	1826
DOLE PURE&LITE CHERR	1673		89	1762
DOLE PURE&LITE RASBE	1706		84	1790
DOLE PUTE&LITE TANGE	1604		77	1681
GATORADE FRUIT PUNCH	1607		311	1918
GATORADE LEMON/LIME	1291		173	1464
GATORADE LEMONADE	1347		180	1527
GATORADE TROPICAL FR	1179		235	1414
HH APPLE JUICE	3045	14	583	3642
O S GRAPEFRUIT JUICE	994		411	1405
O S PINK GRAPEFRUIT	1002		419	1421
REALEMON LEMON JUICE	2255		83	2338
SUNSWEET PRUNE JUICE	1756		28	1784
TROP TWSTR. ORG/CRAN	1573		203	1776
TROPICANA LT ORG/CRA	1152		232	1384
WELCH'S GRAPE JUICE	3138	26	20	3184
WELCHS WHITE GRAPE J	1392			1392
Grand Total	32459	40	3482	35981

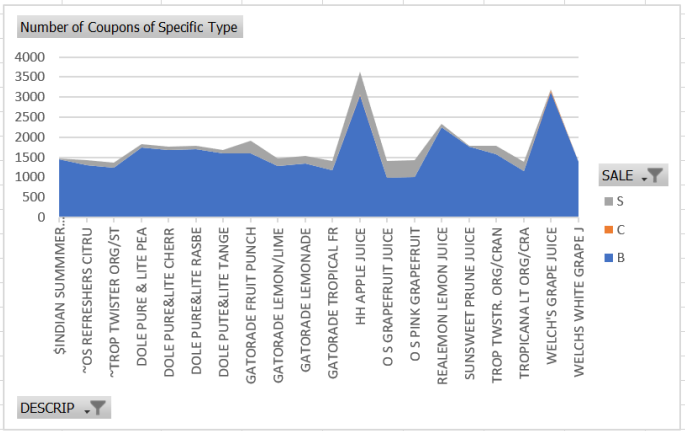


Figure: Data Table & Line Chart

Observation: 'B' indicates a Bonus Buy, 'C' indicates a Coupon, 'S' indicates a simple price reduction. As per the above analysis, Bonus Buy is the most popular coupon being used for the different categories of juices.

Rationale: By identifying the coupon that is most frequently utilized, the store can gain insight into consumer preferences and behaviors when it comes to purchasing a specific juice product. This information can aid DFF to tailor their promotional efforts to match consumer preferences effectively. Additionally, understanding the most popular coupon helps in fine-tuning marketing and promotional campaigns to better serve the needs and desires of customers within the juice category, ultimately resulting in improved sales and customer satisfaction.

Not Selected Business Questions (NSBQ)

NSBQ 1

Which geographical area within Chicago consistently generates the highest sales for frozen foods across multiple years?

City	Chicago
Row Labels	**Sum of FROZEN**
60617	**7104957.15**
3454 E. 118th St.	7104957.15
60618	**21195708.1**
3300 W. Belmont	9423444.81
3350 Western Ave.	11772263.29
60620	**10902572.99**
122 W. 79th St.	10902572.99
60625	**7894328.97**
5233 N. Lincoln Ave.	7894328.97
60629	**14026951.45**
7050 S. Pulaski Rd.	14026951.45
60630	**10991034.79**
4014 W. Lawrence	10991034.79
60632	**7583287.11**
4700 S. Kedzie Ave.	7583287.11
60634	**9012556.53**
3649 N. Central Ave.	9012556.53
60638	**16110954.59**
5829 S. Archer Ave.	16110954.59
60640	**9237411.85**
5235 N. Sheridan Rd.	9237411.85
60645	**16094915.68**
6623 N. Damen Ave.	16094915.68
60646	**12941742.58**
6312 N. Nagle Ave.	12941742.58
60649	**10329702.62**

Figure: Data Table

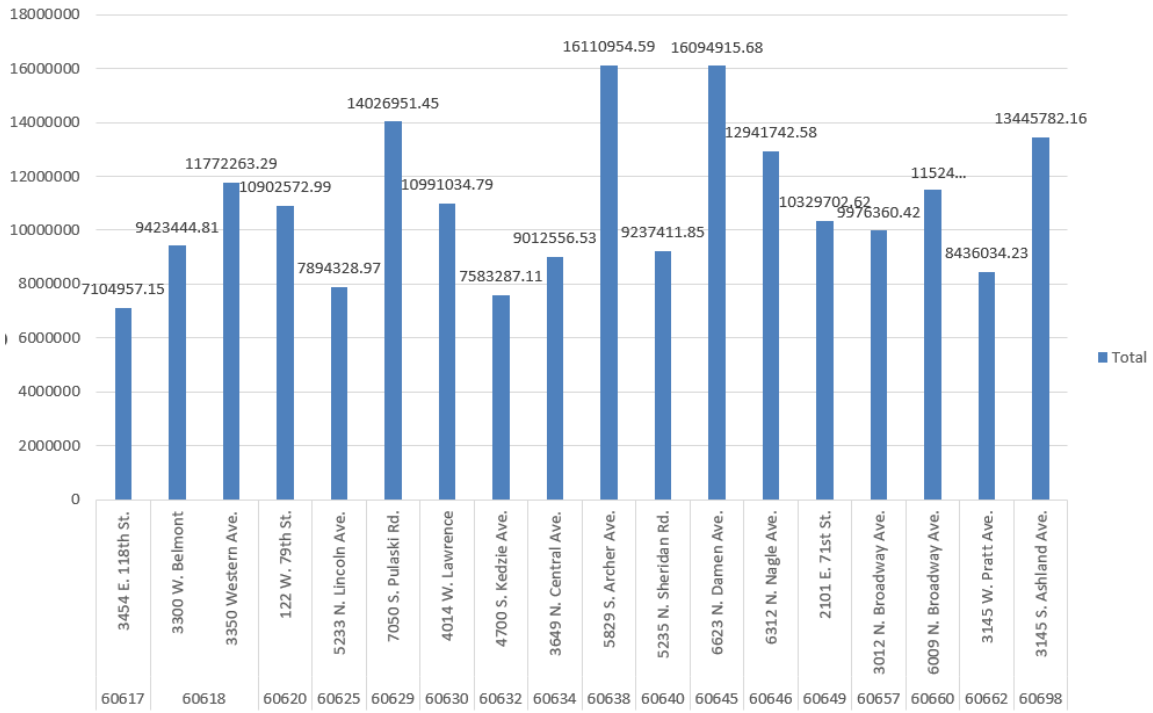


Figure: Line Chart

Observation: Throughout the years, the regions, Archer Avenue and Damen Avenue in Chicago have generated the highest frozen food sales.

Rationale: This emphasis on frozen food is supported by its status as one of DFF's top-selling categories, ensuring that the store capitalizes on a consistent source of revenue. By identifying the regions and zones that purchase the greatest quantity of inventory, the store can develop focused advertising and marketing campaigns for those specific areas (zip codes 5829 and 6623). Additionally, it can assist with inventory management, planning, and prioritization.

NSBQ 2

What is the distribution of average income levels among customers across different store tiers?

Row Labels	Avg of Income
High	**126.6738935**
2	10.55320518
12	9.996659083
14	11.04392933
32	10.67447502
33	10.34592726
52	11.05101698
53	10.60000405
62	11.23619652
68	10.1883657
71	10.40483843
72	10.71219307
75	9.867082871
Low	**42.40091081**
8	10.59700966
18	10.39197554
67	10.79695991
73	10.61496569
Medium	**445.7652583**
4	10.64697132
5	10.92237097
9	10.78715178
28	10.79853422
44	10.86915875
45	10.74537796
47	10.63532646
48	10.75602879
49	10.80675383
50	10.58930796
51	10.62083942
54	10.91089788
56	10.83182543
64	10.60581744
65	10.75462091
74	10.48001664
76	10.1406129

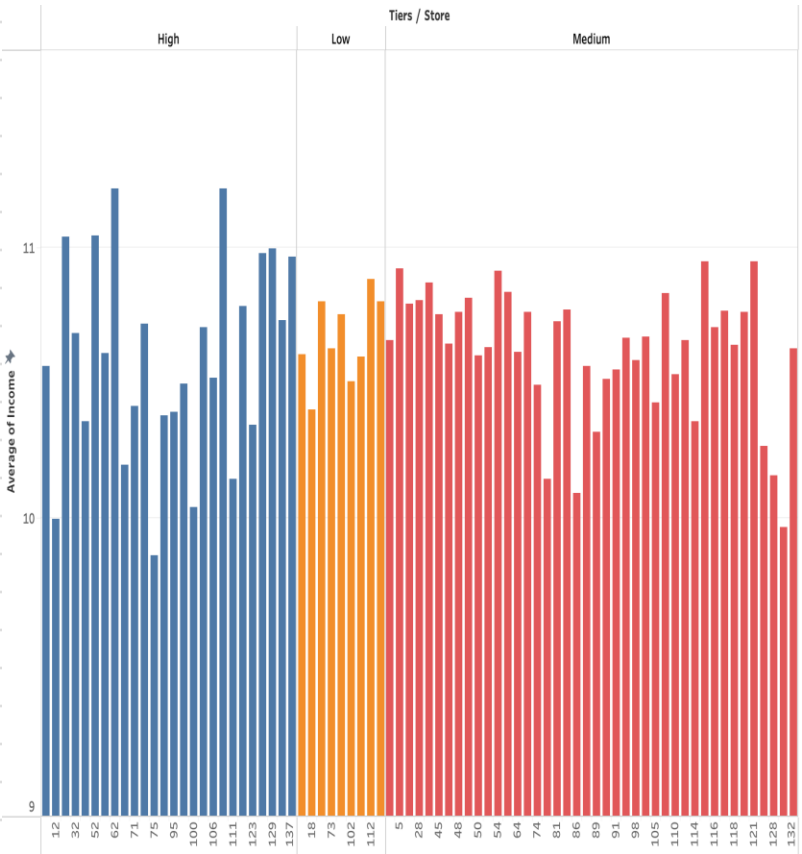


Figure: Data Table & Line Chart

Observation: As an observation, a notable majority of stores fall within the medium-tier category, and the store with the highest average customer income falls into the high-tier category.

Rationale: Analyzing the distribution of average income levels among customers across various store tiers is vital for optimizing business strategies. DFF can customize their marketing efforts, pricing, product offerings, and the overall customer experience to cater to specific income groups effectively. It also facilitates competitive analysis, economic insights, and demographic understanding, ultimately ensuring that stores can align their offerings with the preferences and needs of their customer base, enhancing their market performance and competitiveness.

NSBQ 3

Which are the top 20 soft drink brands that contribute the most to the overall profit in the soft drink category?

Row Labels	Sum of PROFIT
~DIET COKE CANS	391512.28
CANADA DRY TONIC WAT	610098.38
CHERRY COKE	421637.97
CLEARLY CANADIAN BLA	314343.28
CRUSH GRAPE CANS	381732.56
DADS ROOT BEER CANS	448976.04
DADS ROOT BEER DIET	339554.61
DIET PEPSI COLA	249448.58
DIET RITE PEACH	520213.33
DIET WILD CHERRY PEP	284696.4
DOM COLA	414914.88
DOM DIET ROOT BEER	1069630.31
DOM ROOT BEER	405498.54
FRESCA SODA	503436.17
MOUNTAIN DEW 20 OZ N	503981.1329
MT DEW DIET	454048.56
ORANGE SLICE PPD	351145.07
SNAPPLE STRAW KIWI	581148.86
TAB	398333.93
WELCH GRAPE	272218.08
Grand Total	**8916568.963**

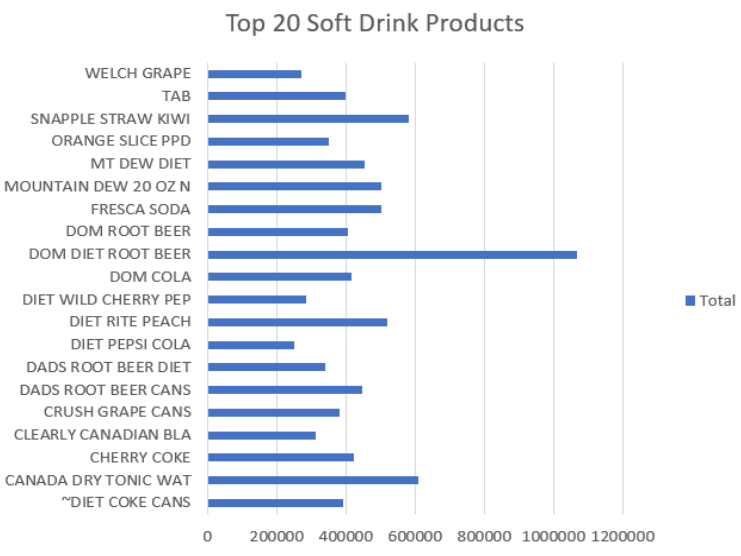


Figure: Data Table & Line Chart

Observation: Dom Diet Root Beer has the highest profit margin of the top 20 soft drink brand categories.

Rationale: Recognizing the top-profit contributors can enable DFF to develop highly focused marketing and promotional campaigns, prominently featuring these particular brands to harness their profitability. By emphasizing these best-performing products, the store can increase sales and profitability, enhancing its position within the competitive landscape.

NSBQ 4

What is the trend in the demographic composition of women (working vs. non-working and with vs. without children) across the top 10 stores?

Store	working women with children	non-working women with children	working women with no children
4	0.1457773	0.107405415	0.304417243
12	0.109006392	0.066937827	0.333294891
33	0.071343028	0.047733105	0.459542506
45	0.202506855	0.164120642	0.318058778
48	0.214338101	0.157315732	0.308082639
54	0.184847029	0.218644882	0.308904649
81	0.19017218	0.13687367	0.313671128
93	0.145332717	0.070471349	0.348427673
128	0.14331827	0.07703895	0.34241908
304	0.125235334	0.1369079	0.304772558
Grand Total	**1.531877205**	**1.183449472**	**3.341591145**

Figure: Data Table

Figure: Line Chart

Observation: The top ten stores have the highest demographic composition of working women with no children.

Rationale: Understanding the demographic trends of working and non-working women can aid in the development of marketing strategies and product offerings that cater to female customers. For instance, we can see that working women with no children have the highest demographic composition, which can guide the development of convenient shopping solutions to meet their specific needs. This data-driven approach can help DFF to adapt to changing consumer dynamics, improve customer engagement, and gain a competitive advantage by aligning their offerings with the distinct characteristics of their female customer base within the top ten stores.

NSBQ 5

Which product exhibits the highest popularity(sales) across the four tiers of stores?

Row Labels	Sum of DAIRY	Sum of FROZEN	Sum of MEAT	Sum of FISH	Sum of BAKERY	Sum of BEER	Sum of WINE	Sum of JEWELRY	Sum of CAMERA	Sum of DELI	Sum of FLORAL
CubFighter	151610882	119215420.3	185015464.6	21226359.88	48739401.67	20004124.67	12173753.9	302270.06	1448650.38	87959549.15	15997811.34
High	381242816.8	280599886	426744862.2	68367145.47	113509751.8	49025988.66	38550991.58	642934.45	3104172.35	185984309	47394457.13
Low	169578519.4	130165613.3	200867339.3	21541671.94	51279680.13	29059986.32	14781070.12	682406.32	2435448.36	95705052.07	16345933.66
Medium	566026093.1	435738646.9	659523756.5	79060077.73	184734979.2	79049063.06	51231106.08	1417108.15	7193176.51	301808255.2	61006852.85
Grand Total	**1268458311**	**965719566.4**	**1472151423**	**190195255**	**398263812.8**	**177139162.7**	**116736921.7**	**3044718.98**	**14181447.6**	**671457165.4**	**140745055**

Figure: Data Table

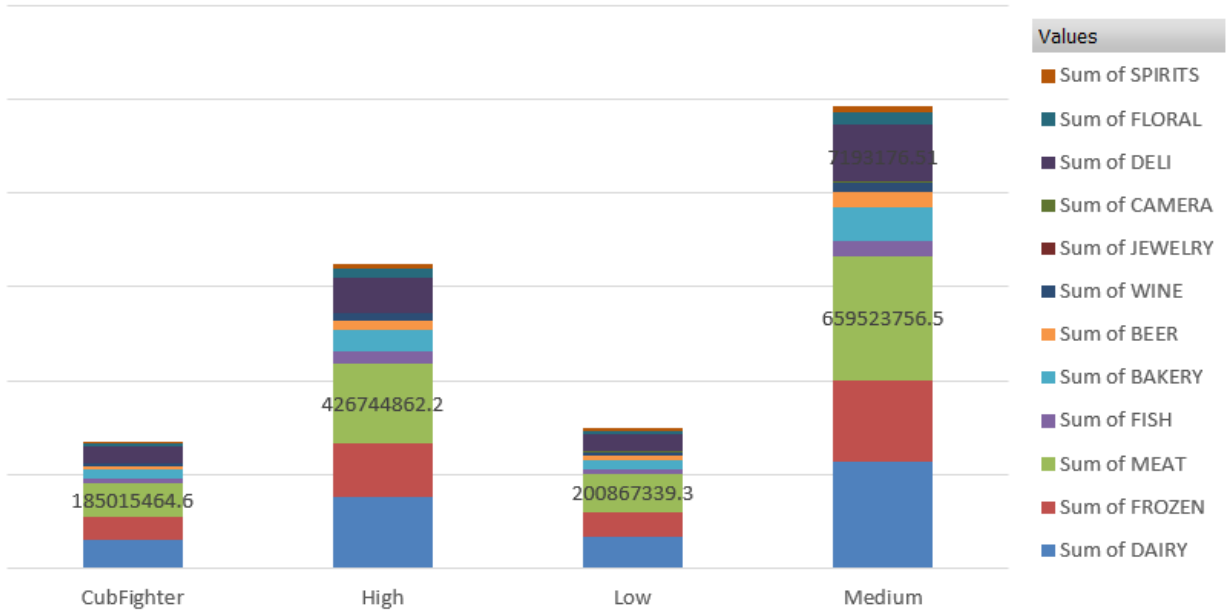


Figure: Bar Char

Observation: The bar chart offers valuable information regarding the most sought-after products across all four tiers, with meat being the most popular

Rationale: Identifying this top-selling product across all tiers allows for targeted inventory management, maintaining consistent stock levels to meet consumer demand efficiently and prevent inventory-related inefficiencies. The store can make use of product diversification, bundling, and complementary offerings as strategies to enhance revenue.

Section 4. Independent Data Marts design using Kimball's approach

Kimball's methodology, often known as the Kimball Approach or the Kimball Lifecycle, is a recognized important technique for developing and putting into use business intelligence (BI) and data warehousing solutions. Ralph Kimball, a distinguished authority on data warehousing, created it. Building business-centric, scalable, and adaptable data warehouses (DW) and business intelligence (BI) solutions is the main goal of Kimball's technique, which enables enterprises to efficiently evaluate and make data-driven choices.

The four key decisions made during the design of a dimensional model include: (The Kimball Group, 1)

1. **Select the business process:** The operational tasks carried out by your company, such as receiving orders, handling insurance claims, scheduling classes, or collecting monthly account snapshots, are known as business processes. Events related to business processes produce or record metrics for success, which are then converted into data in a fact table. The outcomes of a particular business process are the main subject of most fact tables. (The Kimball Group, 2023)

2. **Declare the grain:** A fact table row's specific representation is determined by the grain. Prior to selecting any dimensions or facts, the grain must be specified because each potential dimension or fact needs to be in line with the grain. The uniformity that this consistency maintains on all dimensions of designs is essential to the usability of BI applications. Different grains cannot be combined in a single fact table; each proposed fact table grain produces a distinct physical table. (The Kimball Group, 2023)

3. **Identify the dimensions:** A business process event's dimensions provide the "who, what, where, when, why, and how" context. The descriptive characteristics that BI applications employ to filter and organize the data are contained in dimension tables. All the potential dimensions can be identified by keeping the grain of a fact table firmly in mind. When a dimension is linked to a certain fact row, it should, whenever feasible, have a single value. (The Kimball Group, 2023)

4. **Identify the facts:** Measurements produced by business process events are called facts, and they are typically numerical. The fact table's grain indicates a one-to-one relationship between a single row and a measurement event. A fact table is in line with a concrete, observable event rather than the specifications of a specific report. Only information that

is compatible with the declared grain may be included in a fact table. (The Kimball Group, 2023)

DW Logical Design (Star Schema Design)

Dimension Tables

A dimension table is a data structure in a star schema that provides descriptive attributes, context, and categorization for measures in a data warehouse.

Product_Dim

Purpose	To store information related to product categories derived from the ccount table data source.
Attributes	**ProductKey** – Surrogate key to uniquely identify each record in this dimension. **ProductName** – It gives the name of the specific product/category of a product.

Store_Dim

Purpose	To store data specific to a particular store, providing a dedicated repository for information associated with that store.
Attributes	**StoreKey** – Surrogate key to uniquely identify each record in this dimension. **Store –** It gives the store number.

Date_Dim

Purpose	To record and store date-related information, encompassing details about weeks and specific holidays.
Attributes	**DateKey** – Surrogate key to uniquely identify each record in this dimension. **StartDate** – It gives the start date of the week. **EndDate –** It gives the end date of the week. **Week –** It gives the week number. **Year –** It gives the year information. **SpecialEvents –** This identifier is used to identify and track the presence of specific events during a particular year.

UPC_Dim

Purpose	To provide a structured repository of unique product identifiers, enabling efficient tracking and analysis of individual products.

Attributes	**UPCKey** – Surrogate key to uniquely identify each record in this dimension. **UPCID** – It gives the UPC id of a product. **UPCDesc** – It gives the full brand name of a product.

Coupon_Dim

Purpose	To store information about specific types of coupons used in promotions, aiding in understanding their usage.
Attributes	**CouponKey** – Surrogate key to uniquely identify each record in this dimension. **CouponCategory** – It gives information about each coupon category (A, B, and C).

Demo_Dim

Purpose	To store demographic data providing insights into customer characteristics and behavior.
Attributes	**DemoKey** – Surrogate key to uniquely identify each record in this dimension. **Store –** It gives the store number. **City** – It gives the name of the specific city a store is located in. **Avid -** The percentage of Avid shoppers. **Hurried** – The percentage of Hurried Shoppers. **Strange** – The percentage of Strange Shoppers.

Fact Tables

Fact tables are data tables that store numerical data and are key for analyzing business data by connecting with dimension tables to gain insights.

Fact_Sales

Purpose	The Fact_sales table is a data table used to store information about sales transactions for different product categories across stores and years.
Attributes	**ProductKey –** It is a surrogate key in Product_Dim table and a foreign key in this table. **StoreKey** – It is a surrogate key in Store_Dim table and a foreign key in this table. **DateKey –** It is a surrogate key in Date_Dim table and a foreign key in this table. **TotalSales** – It represents the cumulative sales of all items or products sold during that transaction.

Fact_Purchases

Purpose	The Fact_Purchases is a data table used to store information related to purchase transactions, including details about the specific coupon used.
Attributes	**UPCKey –** It is a surrogate key in UPC_Dim table and a foreign key in this table. **CouponKey** – It is a surrogate key in Coupon_Dim table and a foreign key in this table. **TotalPurchases** – It counts the quantity of items purchased for each unique coupon used in transactions, serving as a key metric for analyzing the popularity and impact of coupons on purchases.

Fact_Demo

Purpose	The Fact_Demo table is a data table used to store demographic data related to shopper behavior, specifically the percentages of avid, hurried, and strange shoppers.
Attributes	**DemoKey –** It is a surrogate key in Demo_Dim table and a foreign key in this table. **TotalPercentAvid** – It represents the overall percentage of avid shoppers. **TotalPercentHurried** – It represents the overall percentage of Hurried shoppers. **TotalPercentStrange** – It represents the overall percentage of strange shoppers.

Dimension Matrix

Data Mart	**Dimension**					
	Product_Dim	**Store_Dim**	**Date_Dim**	**UPC_Dim**	**Coupon_Dim**	**Demo_Dim**
Sales_Data_Mart	×	×	×			
Coupon_Purchases_Data_Mart				×	×	
Demo_Data_Mart						×

Data Mart Schema

A star schema is a multidimensional denormalized data model optimized for querying large data sets. It is composed of a single fact table, which contains all the measurable items of the data that link to other data dimensions such as time dimensions, categories, etc. We have designed three data marts for our report: the Sales_Data_Mart, the Demo_Data_Mart, and the Coupon_Purchases_Data_Mart, respectively. These data marts are structured using the star schema approach, empowering us to effectively address our business questions. This schema allows us to perform in-depth analysis, enabling us to explore and analyze data through operations like slicing and dicing for meaningful insights

Sales_Data_Mart

The Fact_Sales data mart contains three dimensional tables: Product_Dim, Store_Dim, Date_Dim and one fact table Fact_Sales.

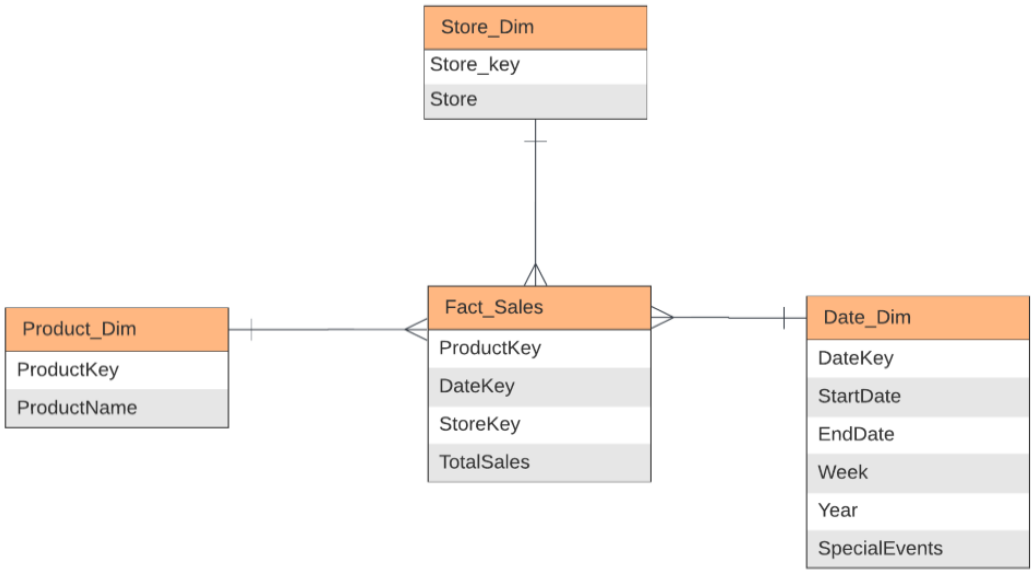


Figure: Sales_Data_Mart Schema

Demo_Data_Mart

The Demo data mart contains a one-dimensional table: Demo_Dim and one fact table Fact_Demo.

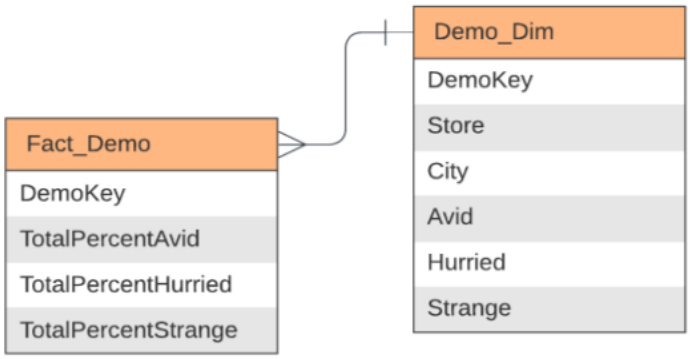


Figure 2: Demo_Data_Mart Schema

Coupon_Purchases_Data_Mart

The Coupon Purchases data mart contains two-dimensional tables: UPC_Dim, Coupon_Dim and one fact table Fact_Purchases

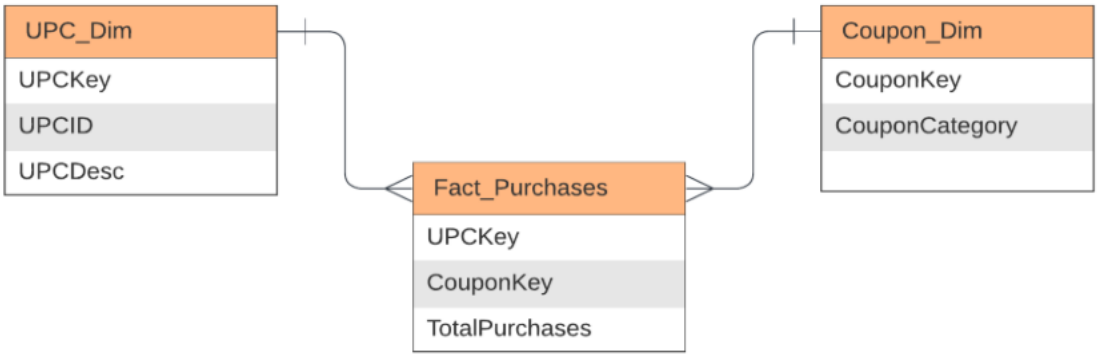


Figure 3: Coupon_Purchases_Data_Mart Schema

Selected Business Question Justification

1. Which are the top 5 categories for products across different stores between the years 1980 to 1997?
    a. The analysis seeks to identify the sales trend from 1980 to 1997 which helps in comprehending the performance of products, including which specific product has performed well and any product that might require special attention. It offers

valuable insights into market trends which can be beneficial to target consumer preferences and optimize inventory based on consumer demands.

To address the business question Product_Dim, Store_Dim, Date_Dim and Fact_Sales tables are utilized from the schema. Data is filtered based on the years 1980 to 1997 using “Year” from the Date dimension. The ProductKey in the Product Dimension associates each sale with a specific product, enabling the calculation of total sales for each product category. The store key in the Fact_Sales table links each sale to a particular store. Thus, this analysis is essential for optimizing our product offerings and store strategies to meet the demands of our customers and enhance our business's competitiveness.

2. What is the year-over-year trends and patterns in alcohol (Beer, Wine, Spirit) sales from 1987 to 1997?
    a. Analyzing year-over-year trends in alcohol sales provides critical insights into consumer preferences and market dynamics. It empowers them to optimize inventory, tailor marketing strategies, and fine-tune pricing for different product categories. This data-driven approach enhances profitability and customer engagement within the alcohol category, delivering substantial business value.

    The schema provided comprises three-dimension tables (Product_Dim, Store_Dim, and Date_Dim) and one fact table (Fact_Sales), enabling a comprehensive analysis of year-over-year alcohol sales trends from 1987 to 1997. Product_Dim categorizes product types, allowing precise filtering of alcohol categories. Date_Dim's temporal attributes, including Year, provide the necessary time context to study year-over-year trends, with SpecialEvents tracking event impacts. Fact_Sales records TotalSales and serves as the central repository for deriving insights into alcohol sales patterns. The schema with its dimension and fact tables offers a robust foundation for in-depth analysis of year-over-year alcohol sales trends.

3. What are the annual sales patterns for wine during peak seasons in the United States?
    a. Particularly during the holiday season, wine is a popular choice for gift-giving, toasting, and other celebrations; and hence can witness a significant surge in demand during such occasions. Identifying holiday wine sales patterns can aid in tailoring marketing efforts and promotional activities during peak seasons. The analysis can shed light on consumer preferences, such as when customers buy more wine, which can help in making informed decisions related to product development and supply chain.

    To address this question Product_Dim, Date_Dim and Fact_Sales tables are used from the schema. For the purpose of identifying peak seasons and wine sales, the data is filtered using the SpecialEvents and "Wine" attributes from the Product_Dim and Date_Dim dimensions, respectively. TotalSales calculates total

wine sales for each peak season and for each year. The analysis is valuable because it provides insights into wine sales patterns during peak seasons, allowing us to align our strategies with these patterns.

4. How does the percentage of avid, hurried, and strange shoppers vary across different cities, and how can we tailor marketing strategies accordingly?
    a. Understanding variations in shopper behavior percentages across cities is crucial for tailoring effective marketing strategies. This insight enables businesses to personalize their approaches to different customer segments, improving customer engagement and satisfaction. By identifying the preferences and behaviors of avid, hurried, and strange shoppers in various locations, companies can allocate resources more efficiently, optimizing marketing campaigns and ultimately driving higher sales and revenue.

       The schema effectively addresses the business question by utilizing the Demo_Dim dimension table and the Fact_Demo fact table. Demo_Dim houses demographic data that characterize customer behavior with attributes like Store, City, and percentages of Avid, Hurried, and Strange shoppers. The Fact_Demo table aggregates overall percentages while linking to Demo_Dim, offering a comprehensive view of shopper behavior. This schema enables businesses to understand variations in shopper characteristics across cities, facilitating the customization of marketing strategies for enhanced customer engagement and optimized campaigns, ultimately driving better business outcomes.

5. What is the most popular coupon used for purchasing the top 20 juice categories?
    a. Identifying the top 20 juice categories based on coupon usage allows us to assess the performance of various juice brands. This analysis helps us optimize our coupon strategy and concentrate our efforts on the most effective coupons. By determining which coupons are most frequently used, it can guide us in the creation of future marketing campaigns and promotions. Analyzing popular coupons provides insights into customer behavior and preferences, allowing us to understand which incentives and discounts resonate with our juice-buying customers.

       To answer this question, we utilize the UPC_Dim, Coupon_Dim, and Fact_Purchases tables from the schema. The CategoryKey links each purchase to a specific juice category, and the CouponKey associates the purchase with a specific coupon. The data is filtered on top 20 juice categories to determine the total purchases for each coupon within these categories. By knowing which coupons are the most popular for purchasing juice products, we can make informed decisions to enhance our customer experience and drive sales.

Develop Two Mapping Tables

Source to Staging Table Mapping

Source data	Source data field	Mapping	Staging table type	Staging table name	Attribute
CCount.csv		Surrogate Key	Temp	Sales_Staging	ProductKey
	Week	Direct Copy			Week
	Store				Store
	Different Product Category Sales				ProductSales
	Different Column Headers	Direct Copy (column headers)			ProductName
Week's Decode Table	Week#	Direct Copy			StartDate
	Start				EndDate
	End				Year
	Special Events				SpecialEvents
Demo.csv		Surrogate Key	Temp	Demo_Staging	DemoKey
	STORE				Store
	CITY				City
	SHPHURR				Hurried
	SHPAVID				Avid
	SHPKSTR				Strange

UPCBJC.csv		Surrogate Key	Temp	Coupon_Staging	UPCKey
	UPC	Direct Copy			UPCID
	DESCRIP				UPCDesc
	Store				Store
	Week				Week
DONE-WBJC.csv	UPC	Direct Copy			CouponCategory
	SALE				

Staging to Data Mart Mapping

Staging table (source data in staging)	Staging table data field/attribute	Mapping	Data Mart Table type	Data Mart Table Name	Field Name/Attribute
DIMENSION TABLES					
Sales_Staging	ProductKey	Direct Copy	Dimension	Product_Dim	ProductKey
	ProductName	Direct Copy			ProductName
		Surrogate Key		Store_Dim	StoreKey
	Store	Direct Copy			Store
		Surrogate Key		Date_Dim	DateKey
	StartDate	Direct Copy			StartDate
	EndDate				EndDate
	Week				Week
	Year				Year

	SpecialEvents				SpecialEvents
Demo_Staging	DemoKey	Direct Copy	Dimension	Demo_Dim	DemoKey
	Store				Store
	City				City
	Hurried				Hurried
	Avid				Avid
	Strange				Strange
Coupon_Staging	UPCKey	Direct Copy	Dimension	UPC_Dim	UPCKey
	UPCID				UPCID
	UPCDesc				UPCDesc
	Store				Store
	Week				Week
		Surrogate Key		Coupon_Dim	CouponKey
	CouponCategory	Direct Copy			CouponCategory
FACT TABLES					
Sales_Staging	ProductKey	Direct Copy	Fact	Fact_Sales	ProductKey
		Surrogate Key			DateKey
		Surrogate Key			StoreKey
	ProductSales	Sum of Sales of different Product categories			TotalSales
Demo_Staging	DemoKey	Direct Copy			DemoKey
	Hurried	=Hurried*100	Fact	Fact_Demo	TotalPercentAvid

	Avid	=Avid*100			TotalPercentHurried
	Strange	=Strange*100			TotalPercentStrange
Coupon_Staging	UPCKey	Direct Copy	Fact	Fact_Purchases	UPCKey
		Surrogate Key			CouponKey
	CouponCategory	Sum of categories of different coupons			TotalPurchases

Physical Design

Data aggregate plan

The initial dataset contains granular information about sales and products. While this level of atomicity is required for transactional systems, analytical systems do not need this level of detail. Considering how business decisions are made in the retail industry and to decrease the complexity of queries and compute resources, we will summarize the initial information in the staging. Aggregation will be made for the following tables.

1. Sales data in the Fact_Sales table will be aggregated over the time dimension to reflect the monthly sales across different stores and years.

2. Purchases data in the Fact_Purchases table will also be aggregated over coupon dimension to reflect the monthly purchases to gather more information on the total sales per type of coupon.

Indexing plan

Following Microsoft's "SQL Server and Azure SQL index architecture and design" guide to implementing indexing best practices (Microsoft, 2023), we choose to implement Columnstore indexes because it is best suited for data warehousing databases that need to process large datasets quickly. Microsoft claims that Columnstore indexes can gain up to 10 times query performance starting with SQL Server 2016 (Microsoft, 2023).

Data standardization plan

Standardization of the data is crucial in a data mart (and data warehousing) for many reasons. The most important are: 1. Consistency, 2. Accuracy, 3. Integration from Multiple Sources, 4. Maintainability and, 5. Scalability.

Database Object	Standard
Data Mart	Department_Data_Mart, the naming standard will follow the snake case standard, with the first letter of every word in uppercase. Example: Marketing_Data_Mart
Fact Table	Fact_Table, will follow the same standard as Data Mart. Example: Fact_Purchases
Dimension Table	Table_Dim, will follow the same standard as Data Mart. Example: Coupon_Dim
Column name	Will follow the Pascal Case standard. Example: TotalPercentStrange

Storage plan

The initial dataset has data for five years, from 1989 to 1994. The uncompressed, uncleaned raw data size is around 6 GB. The initial data must be exponentially more significant to consider implementing advanced storage solutions. If we split the data evenly, the average increase by year is 1.2 GB; taking into consideration that the company can offer more products and gather more data from each product and sale, we can safely assume the annual data size will be 2.4 GB for each year (double from the previous five years).

We are estimating a total database size of around 19 GB for a database containing ten years' worth of information. With today's storage capacity, any modern server can handle this data. This is why we have chosen to keep all data in the Mays Business School's on-premises server

Section 5. Data Cleaning and Integration

The dataset is from the Dominick's Fine Foods dataset and consists of real-world data. Real data requires a thorough cleaning and transformation process to extract meaningful insights and facilitate OLAP analysis. Our preprocessing efforts involved eliminating rows with null values, rectifying date formats, and optimizing the data types of prices and sales values for effective metric calculations. Furthermore, in alignment with specific business requirements, we streamlined the dataset by removing unnecessary attributes that did not contribute to our analytical objectives.

ETL Development Plan

Identify the target data

All the target data will be stored in the database named - **ISTM_637_602_Group10_dw_area**.

Dimension Table

Target Tables	Column Names	Data Type
Coupon_Dim	CouponKey	INT
	CouponCategory	VARCHAR (5)

Target Tables	Column Names	Data Type
UPC_Dim	UPCKey	INT
	UPCID	INT
	UPCDesc	VARCHAR (50)
	Week	INT
	Store	INT

Target Tables	Column Names	Data Type
Product_Dim	ProductKey	INT
	ProductName	VARCHAR (50)

Target Tables	Column Names	Data Type
Date_Dim	DateKey	INT
	StartDate	DATE
	EndDate	DATE
	Week	INT
	Year	YEAR
	SpecialEvents	VARCHAR (50)

Target Tables	Column Names	Data Type
	StoreKey	INT

Store_Dim	Store	INT

Target Tables	Column Names	Data Type
Demo_Dim	DemoKey	INT
	City	VARCHAR (50)
	Store	INT
	Hurried	DECIMAL
	Avid	DECIMAL
	Strange	DECIMAL

Fact Tables

Target Tables	Column Names	Data Type
Fact_Purchases	UPCKey	INT
	CouponKey	INT
	TotalPurchases	INT

Target Tables	Column Names	Data Type
Fact_Sales	ProductKey	INT
	DateKey	INT
	StoreKey	INT
	TotalPurchases	DECIMAL

Target Tables	Column Names	Data Type
Fact_Demo	DemoKey	INT
	TotalPercentHurried	DECIMAL
	TotalPercentAvid	DECIMAL
	TotalPercentStrange	DECIMAL

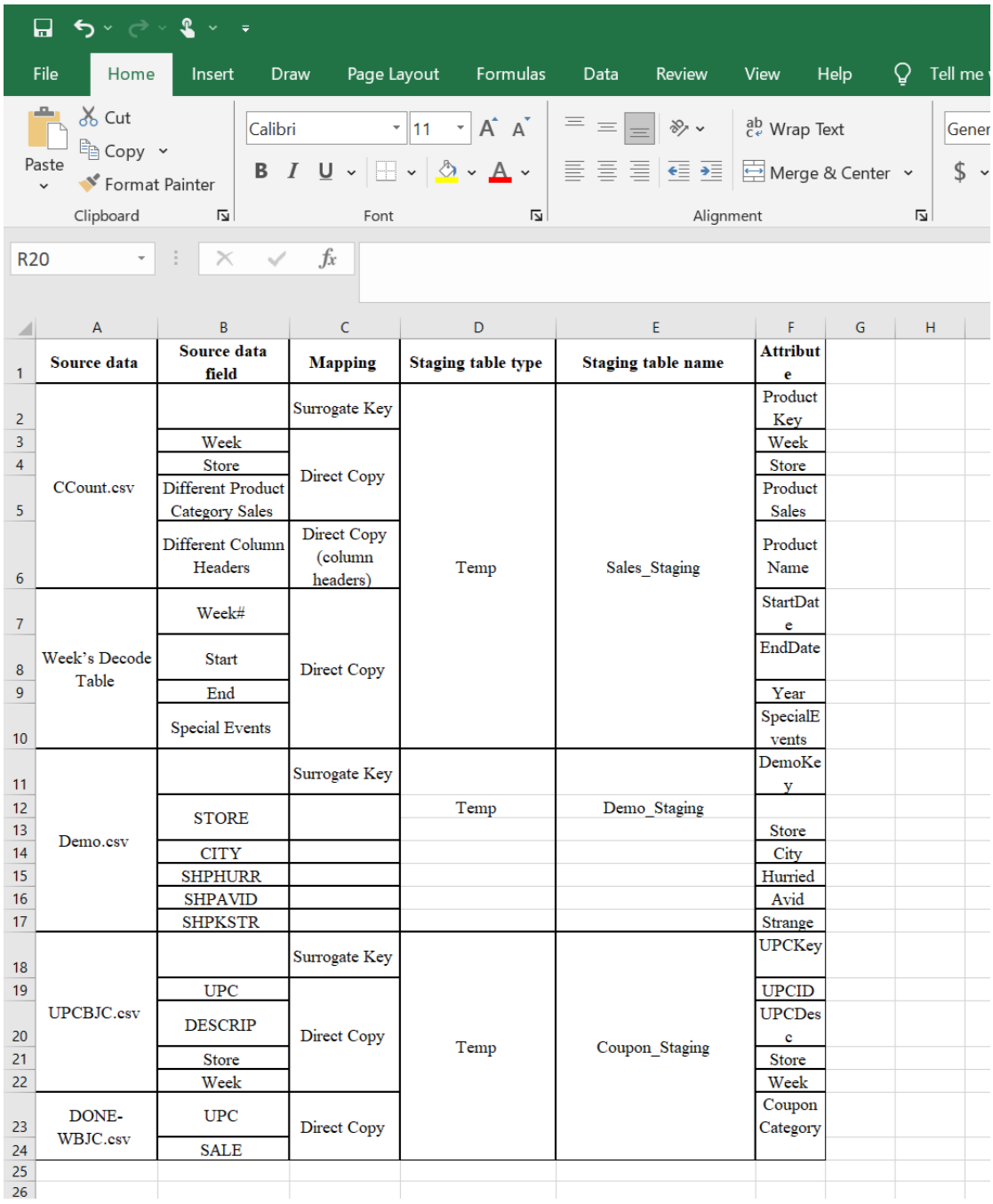
Identify the source data

The ETL (Extract, Transform, Load) process comprises extracting data from source files, which can be in the form of CSV, Excel, or PDF files. The extracted data is temporarily stored in interim tables. After performing necessary joins, the consolidated information is then stored in staging tables. Subsequently, the data undergoes cleaning and transformation based on specific requirements before being transferred to the data warehouse area, which includes various data marts.

Source File	Staging Area Tables	Datamart Tables
Ccount.csv	Sales_Staging	Fact_Sales Product_Dim Store_Dim DateDim
WeekDecode (Dominick.pdf)		
Demo.csv	Demo_Staging	Fact_Demo Demo_Dim
DONE_WBJC.csv	Coupon_Staging	Fact_Purchases Coupon_Dim UPC_Dim
UPCBJC.csv		

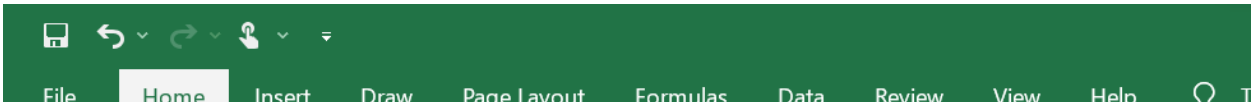
Mapping Tables

Source to Staging Table Mapping



	A	B	C	D	E	F	G	H
1	**Source data**	**Source data field**	**Mapping**	**Staging table type**	**Staging table name**	**Attribute**		
2	CCount.csv		Surrogate Key	Temp	Sales_Staging	Product Key		
3		Week	Direct Copy			Week		
4		Store				Store		
5		Different Product Category Sales				Product Sales		
6		Different Column Headers	Direct Copy (column headers)			Product Name		
7	Week's Decode Table	Week#	Direct Copy			StartDate		
8		Start				EndDate		
9		End				Year		
10		Special Events				SpecialEvents		
11	Demo.csv		Surrogate Key	Temp	Demo_Staging	DemoKey		
12		STORE						
13						Store		
14		CITY				City		
15		SHPHURR				Hurried		
16		SHPAVID				Avid		
17		SHPKSTR				Strange		
18	UPCBJC.csv		Surrogate Key	Temp	Coupon_Staging	UPCKey		
19		UPC	Direct Copy			UPCID		
20		DESCRIP				UPCDesc		
21		Store				Store		
22		Week				Week		
23	DONE-WBJC.csv	UPC	Direct Copy			Coupon Category		
24		SALE						
25								
26								

Staging to Data Mart Mapping



Staging table (source data in staging)	Staging table data field/attribute	Mapping	Data Mart Table type	Data Mart Table Name	Field Name/Attribute
DIMENSION TABLES					
Sales_Staging	ProductKey	Direct Copy	Dimension	Product_Dim	ProductKey
	ProductName	Direct Copy			ProductName
		Surrogate Key		Store_Dim	StoreKey
	Store	Direct Copy			Store
		Surrogate Key		Date_Dim	DateKey
	StartDate	Direct Copy			StartDate
	EndDate				EndDate
	Week				Week
	Year				Year
	SpecialEvents				SpecialEvents
Demo_Staging	DemoKey	Direct Copy	Dimension	Demo_Dim	DemoKey
	Store				Store
	City				City
	Hurried				Hurried
	Avid				Avid
	Strange				Strange
Coupon_Staging	UPCKey	Direct Copy	Dimension	UPC_Dim	UPCKey
	UPCID				UPCID
	UPCDesc				UPCDesc
	Store				Store
	Week				Week
		Surrogate Key		Coupon_Dim	CouponKey
	CouponCategory	Direct Copy			CouponCategory
FACT TABLES					
Sales_Staging	ProductKey	Direct Copy	Fact	Fact_Sales	ProductKey
		Surrogate Key			DateKey
		Surrogate Key			StoreKey
	ProductSales	Sum of Sales of different Product categories			TotalSales
Demo_Staging	DemoKey	Direct Copy	Fact	Fact_Demo	DemoKey
	Hurried	=Hurried*100			TotalPercentAvid
	Avid	=Avid*100			TotalPercentHurried
	Strange	=Strange*100			TotalPercentStrange
Coupon_Staging	UPCKey	Direct Copy	Fact	Fact_Purchases	UPCKey
		Surrogate Key			CouponKey
	CouponCategory	Sum of categories of different coupons			TotalPurchases

Data Extraction Rules

The initial phase of the ETL (Extract, Transform, Load) process is data extraction, where information is fetched from diverse sources, undergoes preparation and processing, and then gets stored in a data warehouse. This extraction process entails defining specific rules to ensure the accuracy and consistency of the extracted data, essential for analysis. The source data is derived from the Chicago Booth dataset, which consists of multiple files in various data formats. To meet our business needs, we first identified the files required for our data warehouse. The chosen extraction files are provided in diverse formats:

1. CSV Files: CCount-Copy.csv, Demo.csv, UPCJBC.csv, DONE-WBJC.csv
2. PDF File: Week Decode.pdf

The extraction rules for these source files include:

1. Directory Location: All data must be organized within a single directory location.
2. Data Format: The data must be formatted in the same data type format. For this purpose, all the files must be converted into CSV data format.
3. Data Validation: All data must be validated in the CSV files prior to the extraction process to ensure its integrity
4. Delimiter Checks: The CSV files must undergo a check for the delimiters used. The correct delimiter must be used during the extraction process to ensure proper column separations.

Based on the above rules the data must be imported in the SSIS import export wizard.

Data Transformation and Cleaning Rules

After the data extraction phase in the ETL process, the next critical step is data transformation and cleaning. This step is imperative to ensure that the extracted data is suitable for comprehensive analysis and reporting. The data transformation process involves cleaning, manipulating, and potentially enriching the data to align with the requirements of the desired data output. The following transformation rules were defined:

1. Identification of Missing Values: The data must be validated for any missing records. The significance of missing values must be evaluated for analysis. Removal is recommended if the data is not important, otherwise, enrichment should be applied.
2. Removal of Null Values: Any null values present in the dataset must be removed. The null values in tables such as CCount and Demographic must be addressed for improved data quality.

3. Removal of Unwanted Columns: The data must be transformed based on the columns essential for business processes. Any unwanted columns must be dropped. For example, only relevant product columns should be selected from the CCount.csv table.
4. Removal of Special Character: Exclude any special characters, such as "", $, or spaces, from the files. Remove quotes ("") found in files like Demographic and UPCBJC to enhance data cleanliness.
5. Removal of Negative Values: Exclude records containing inappropriate negative values for specific columns.
6. Data Conversion: Convert values in the dataset to their appropriate data types to facilitate necessary operations. For instance, ensure that sales values are in a numeric format for additive operations during analysis.
7. Table Merging: Perform join operations as required for certain tables to create independent data marts. These join operations should be executed in the staging area.
8. SSIS Manipulation Operations: Utilize SSIS functions, such as UNPIVOT, for necessary data manipulations. For instance, transpose the Product_Dim table from the Sales_Staging table to extract the names of all product categories as rows.

Plan for aggregate tables

To create aggregates for the fact tables in our data marts, we have considered the dimensions and measures involved in each fact table. Aggregations are pre-calculated summaries of data that can improve query performance. Here's a plan for creating aggregates for each fact table, along with explanations for each aggregate function:

Fact_Sales

Total Sales: Calculate the sum of TotalSales for different product categories across stores and years.
Group by: ProductKey, StoreKey, DateKey
Attributes:

1. ProductKey: Uniquely identifies each product.
2. StoreKey: Uniquely identifies each store.
3. DateKey: Represents each individual date.

This provides the most detailed information, allowing analysis of sales for each product in each store on each specific date.

Fact_Purchases

Total Purchases: Calculate the count of transactions for each combination of product and coupon.

Group by: UPCKey, CouponKey
Attributes:

1. UPCKey: Uniquely identifies each product.
2. CouponKey: Uniquely identifies each coupon category.

This provides the most detailed information on purchases for each unique product and coupon combination.

Fact_Demo

Total Percent Avid, Hurried, Strange: Calculate the percentage of avid, hurried, and strange shoppers for each demographic.
Group by: DemoKey
Attributes:

1. DemoKey: Uniquely identifies each demographic record.

This provides a comprehensive view of demographic data, summarizing percentages of avid, hurried, and strange shoppers for each demographic.

For general consideration, it is important to ensure that all the attributes are properly indexed for fast retrieval.

Write out the organization of data staging area

The organization of the data staging area involves structuring separate staging tables to store raw data extracted from diverse sources. These tables remain unaltered and are categorized by source. Metadata tables accompany staging tables to capture information about data structure, while we can also flag issues for cleansing before further processing. The staging area may include temporary work tables for complex transformations, and considerations such as partitioning, indexing, and security measures are implemented to optimize performance and maintain data integrity.

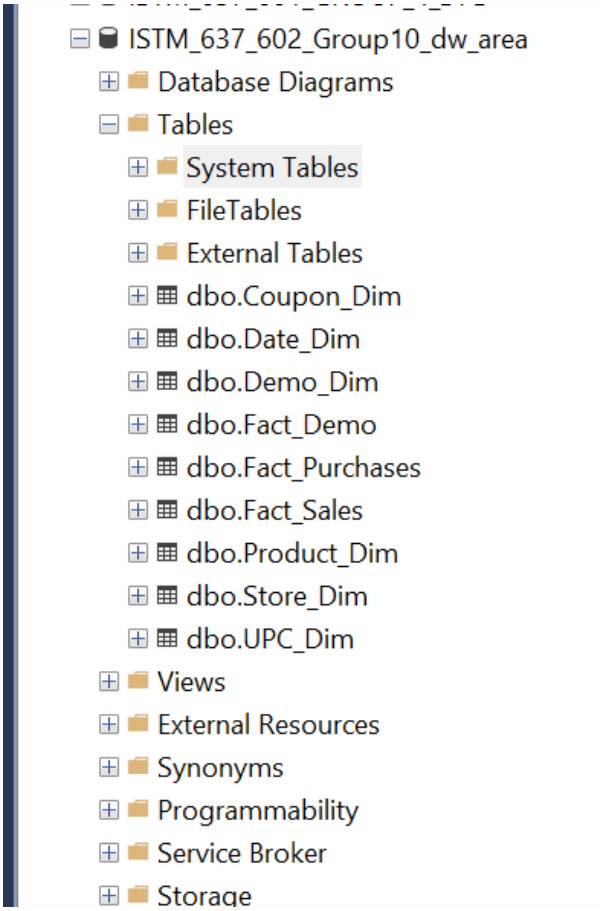


Figure: SQL Server with all the tables in the Data Staging and Data Warehousing area

We have the following tables in our ISTM_637_602_Group10_staging_area database

Temp Tables

Ccount	All the data from the customer count CSV files has been directly loaded into this table.
Week_Decode_Clean	All the data from the WeekDecode CSV file (derived from the Dominicks.pdf file) has been directly loaded into this table.
UPCBJC	All the data taken from the UPC table for the juices is directly loaded into this table
DONE-WBJC	All the data taken from the Movement table for the juices is directly loaded into this table.

All the temp tables mentioned above are stored temporarily in the database to facilitate JOIN operations in the Staging tables.

Staging Tables

Sales_Staging	Sales_Staging table contains a cleaner version of Ccount table following the data integrity and consistency constraints. It is a combined table for both Ccount and Week_Decode_Clean temp tables by performing JOIN operations.
Demo_Staging	All the data from the Demographic CSV file has been directly loaded into this table and then the cleansing and transformation tasks have been applied to make it relevant to the requirements.
Coupon_Staging	Coupon_Staging table contains a cleaner version of UPCBJC and movement table following the data integrity and consistency constraints. It is a combined table for both UPCBJC and DONE-WBJC temp tables by performing JOIN operations

Procedures for all data extractions and loadings

Described below we will find the procedures we did for answer our five business questions selected for this project:

1. We determined which data we needed to answer each question.
2. We located the source files where the data was
3. From the location of the source data, we start planning our load procedures
4. At first, we load the data from the different sources to the database 601_Group2_Staging_area
5. The data in 601_Group2_Staging_area is not considered clean, but we have it on SQL Server
6. We created SSIS packages to clean, join and remove outliers from the data
7. Within the package we create and drop temporary tables
8. After cleaning the data, we proceed to load it to the final destination.
9. We also calculated some values for the fact tables

10. The last step is to create the fact tables from the different dimension tables in the final database

ETL for Dimension Tables

Date_Dim Table: The Dim_Date dimension table incorporates details derived from the Week Decode table related to dates. It includes columns such as Week number, Start date, End date and Special Events which have been extracted from the Week Decode table and loaded in the temporary table. These columns were then merged with the Date column from the CCount-Copy table in the staging area. After cleaning the data, it was loaded into the Dim_Date dimension table, where each entry is assigned a surrogate key known as DateKey.

Store_Dim Table: The dimension table provides information regarding stores. The data is extracted from the Sales_Staging table having Store column which consists of various store numbers. The data was then cleaned and loaded into the Store_Dim dimension table with the surrogate key StoreKey.

Product_Dim Table: The dimension table contains details about different product categories from the CCount-Copy table. Relevant product categories were extracted from the CCount table, and the remining columns were excluded before loading into the Sales_Staging table. Further, the data was transformed using filters and pivot operations to achieve the intended format in the dimension table. Each product category in the dimension table was assigned a surrogate key, ProductKey.

Demo_Dim Table: The dimension table contains details about different stores regarding Hurried, Avid or Strange shoppers. The initial data is extracted from Demo.CSV file and it contains details about the City, Store, Hurried, Avid and Strange shoppers. The remaining columns were excluded before loading into the Demo_Dim table. Furthermore, the data was transformed from a purely string to decimal and integer values. The Store attribute is an integer, previously varchar. The Strange, Hurried and Avid attributes are decimal with a precision of 18 and 10 decimal points. The table was loaded into the Demo_Dim dimension table with surrogate key DemoKey.

UPC_Dim Table: The UPC_Dim dimension table incorporates details derived from the Coupon_Staging table for juices that in turn has been derived from the JOIN operation of UPC and Movement tables for juice brands. It includes columns such as UPC ID, UPC Description, Week number and Store numbers which have been extracted from the UPC and Movement tables for juices the temporary table. We have removed the double quotes from the UPCDesc attribute to make it clean. After cleaning the data, it was loaded into the UPC_Dim dimension table, where each entry is assigned a surrogate key known as UPCKey.

Coupon_Dim Table: The Coupon_Dim dimension table incorporates details derived from the Coupon_Staging table for juices that in turn has been derived from the JOIN operation of UPC and Movement tables for juice brands. It includes CouponCategory as the name which has been derived from the 'sales' column of Movement table. The CouponCategory only has four values - 'B' indicating a Bonus Buy, 'C' indicating a Coupon, 'S' indicating a simple price reduction and 'G' indicating undefined coupon category. Here, we are trying to understand how promotions work for juices and which coupon is the most popular one for a specific store and week. After cleaning the data, it was loaded into the Coupon_Dim dimension table, where each entry is assigned a surrogate key known as CouponKey.

ETL for Fact Tables

Fact_Sales: The fact table provides information regarding the total sales for all the product categories for different years and stores. The data was extracted from the Sales_Staging table.
It was transformed to derive total sales for each product category. DateKey, StoreKey and ProductKey were retrieved by Lookup from the Dimension tables.

Fact_Demo: The fact table provides information regarding the percentage of shoppers defined in a specific category for different stores. The data was extracted from the Demo_Staging table. It was transformed to express the percentage of Hurried, Avid and Strange shoppers in a store. TotalPercentHurried, TotalPercentAvid and TotalPercentStrange were retrieved by multiplying each corresponding field by 100. Also, these new columns were truncated to only store 2 decimal places.

Fact_Purchases: The Fact_Purchases table serves as a repository for purchase transaction information, emphasizing details about the specific coupons utilized. Extracted from the Coupon_Staging table, the data is transformed to capture the count of items purchased for each unique coupon. The table includes the following attributes: UPCKey, functioning as a surrogate key sourced from the UPC_Dim table and a foreign key in Fact_Purchases; CouponKey, serving as a surrogate key from the Coupon_Dim table and a foreign key in Fact_Purchases. The TotalPurchases attribute tallies the quantity of items bought for each distinct coupon in transactions, providing a metric for evaluating the popularity of various coupons on overall purchases.

SQL Scripts Details

```
SELECT
  [Week #] AS Week,
  [Start],
  [End],
  [Special Events] INTO Week_Decode_Clean
```

```
FROM
   [Week Decode]
WHERE
   ([Week #] IS NOT NULL)
   AND ([Week #] <> ' ');

SELECT
   DISTINCT Product_Name
FROM
   ISTM_637_602_Group10_Staging.dbo.Sales_Staging UNPIVOT (
      VALUE FOR Product_Name IN (
         Grocery,
         Dairy,
         Frozen,
         Bottle,
         Meat,
         Fish,
         Floral,
         Deli,
         Cheese,
         Bakery,
         Pharmacy,
         Jewelry,
         Beer,
         Wine,
         SPIRITS,
         Camera,
         Saladbar,
         Cosmetic
      )
   ) AS unpvt;

WITH cteas (
   SELECT
      Week,
      Store,
      sum (cast (Grocery AS money)) AS Grocery,
      sum (cast (Dairy AS money)) AS Dairy,
      sum (cast (Frozen AS money)) AS Frozen,
      sum (cast (Bottle AS money)) AS Bottle,
```

```
        sum (cast (Meat AS money)) AS Meat,
        sum (cast (Fish AS money)) AS Fish,
        sum (cast (Floral AS money)) AS Floral,
        sum (cast (Deli AS money)) AS Deli,
        sum (cast (Cheese AS money)) AS Cheese,
        sum (cast (Bakery AS money)) AS Bakery,
        sum (cast (Pharmacy AS money)) AS Pharmacy,
        sum (cast (Jewelry AS money)) AS Jewelry,
        sum (cast (Beer AS money)) AS Beer,
        sum (cast (Wine AS money)) AS Wine,
        sum (cast (SPIRITS AS money)) AS SPIRITS,
        sum (cast (Camera AS money)) AS Camera,
        sum (cast (Saladbar AS money)) AS Saladbar,
        sum (cast (Cosmetic AS money)) AS Cosmetic,
        sum (cast (ConvFood AS money)) AS ConvFood
    FROM
        ISTM_637_602_Group10_Staging.dbo.Sales_Staging
    group by
        [Week],
        [Store]
),
cte_2 AS (
    SELECT
        week,
        store,
        Product_Name,
        Total_Sales
    FROM
        (
            SELECT
                Week,
                store,
                Grocery,
                Dairy,
                Frozen,
                Bottle,
                Meat,
                Fish,
                Floral,
                Deli,
```

```
            Cheese,
            Bakery,
            Pharmacy,
            Jewelry,
            Beer,
            Wine,
            SPIRITS,
            Camera,
            Saladbar,
            Cosmetic,
            ConvFood
        FROM
            cte
    ) a UNPIVOT (
        Total_Sales FOR Product_Name IN (
            Grocery,
            Dairy,
            Frozen,
            Bottle,
            Meat,
            Fish,
            Floral,
            Deli,
            Cheese,
            Bakery,
            Pharmacy,
            Jewelry,
            Beer,
            Wine,
            SPIRITS,
            Camera,
            Saladbar,
            Cosmetic
        )
    ) AS unpvt
);

SELECT
    p.ProductKey,
    d.DateKey,
```

```
   s.Store_Key,
   Total_Sales
FROM
   cte_2
   INNER JOIN Date_Dim d ON cte_2.Week = d.Week
   INNER JOIN Product_dim p ON p.ProductName = cte_2.Product_Name
   INNER JOIN Store_Dim s ON s.Store = cte_2.Store 4.SELECT UPCKey,
   CouponKey,
   COUNT(c.CouponCategory) AS TotalPurchases
FROM
   ISTM_637_602_Group10_Staging.dbo.Coupon_Staging AS cs
   INNER JOIN UPC_Dim AS u ON cs.UPCID = u.UPCID
   INNER JOIN Coupon_Dim c ON cs.CouponCategory = c.CouponCategory
GROUP BY
   UPCKey,
   CouponKey;

UPDATE
   ISTM_637_602_Group10_staging_area.dbo.Coupon_Staging
SET
   UPCDesc = REPLACE (UPCDesc, "", '');

UPDATE
   ISTM_637_602_Group10_dw_area.dbo.Demo_Dim
SET
   City = REPLACE (City, "", '');

SELECT
   DISTINCT REPLACE (CouponCategory, "", '') AS CouponCategory
FROM
   Coupon_Staging
where
   CouponCategory NOT IN ("""", '1', '') 8.CREATE TABLE [dbo].[Product_Dim](
      [ProductKey] int identity (1, 1),
      [ProductName] date
   );

DROP TABLE [dbo].[Demo_Staging];

DROP TABLE [dbo].[Demo_Staging_Temp];
```

```
SELECT
   [dbo].[Demo_Staging_Temp].["CITY"] AS City,
   [dbo].[Demo_Staging_Temp].["STORE"] AS Store,
   [dbo].[Demo_Staging_Temp].["SHPHURR"] AS Hurried,
   [dbo].[Demo_Staging_Temp].["SHPAVID"] AS Avid,
   [dbo].[Demo_Staging_Temp].["SHPKSTR"] AS Strange INTO Demo_Staging
FROM
   [dbo].[Demo_Staging_Temp];

DROP TABLE [dbo].[Demo_Dim];

CREATE TABLE [dbo].[Demo_Dim] (
   [DemoKey] INT IDENTITY (1, 1),
   [City] [varchar](50),
   [Store] [int],
   [Hurried] [decimal](18, 10),
   [Avid] [decimal](18, 10),
   [Strange] [decimal](18, 10)
);

SELECT
   City,
   Store,
   cast(Hurried AS decimal(18, 10)) AS Hurried,
   cast(Avid AS decimal(18, 10)) AS Avid,
   cast(Strange AS decimal(18, 10)) AS Strange
FROM
   [dbo].[Demo_Staging]
WHERE
   (City <> """");

CREATE TABLE [dbo].[Fact_Demo] (
   [DemoKey] INT NOT NULL,
   [TotalPercentHurried] decimal(18, 2),
   [TotalPercentAvid] decimal(18, 2),
   [TotalPercentStrange] decimal(18, 2)
);

DROP TABLE [dbo].[Fact_Demo];
```

```
SELECT
   DemoKey,
   Hurried * 100 AS TotalPercentHurried,
   Avid * 100 AS TotalPercentAvid,
   Strange * 100 AS TotalPercentStrange
FROM
   Demo_Dim;
```

ETL Implementation

ETL for Staging Tables

Coupon_Staging Table

Data extraction for this table will be done using the two csv files from Movement and UPC for the specific product - Juices and then performing a JOIN operation on them to get the relevant data.

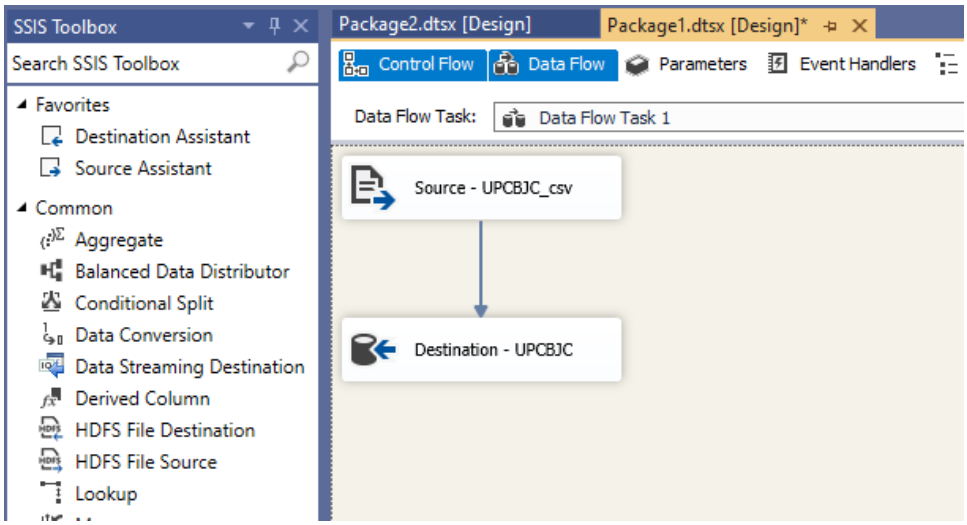


Figure: Extracting values from UPC.csv file to the database

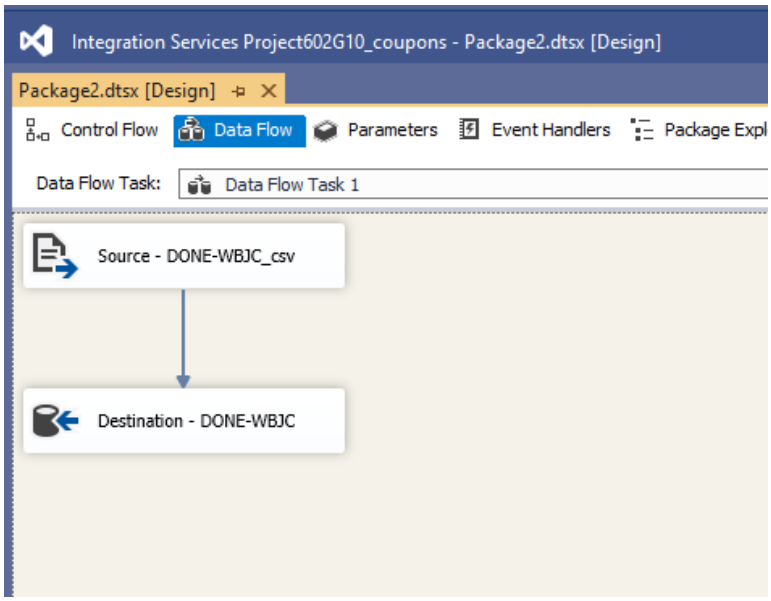


Figure: Extracting values from DONE-WBJC.csv (Movement) file to the database

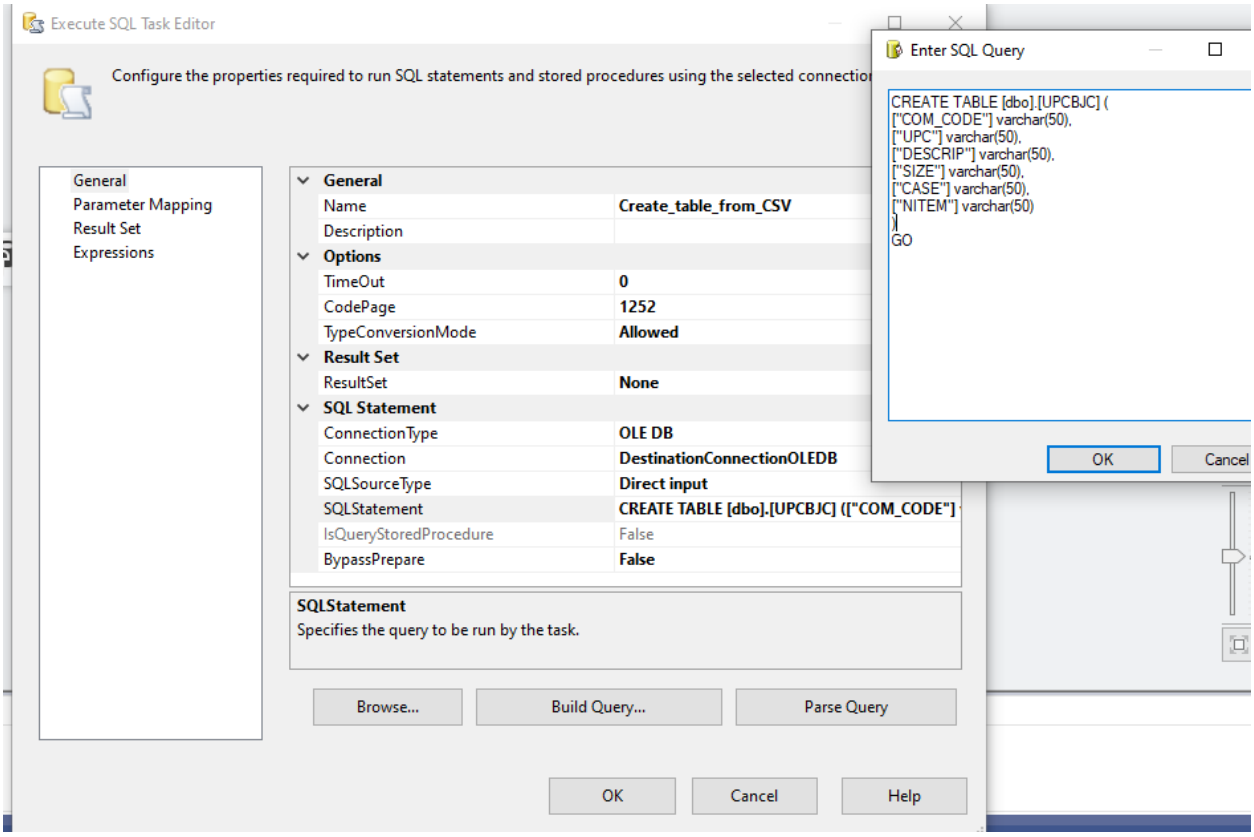


Figure: Creating a table on the destination for UPC

Now, once we have flat file data in the database, we can perform an execute SQL task to join the tables.

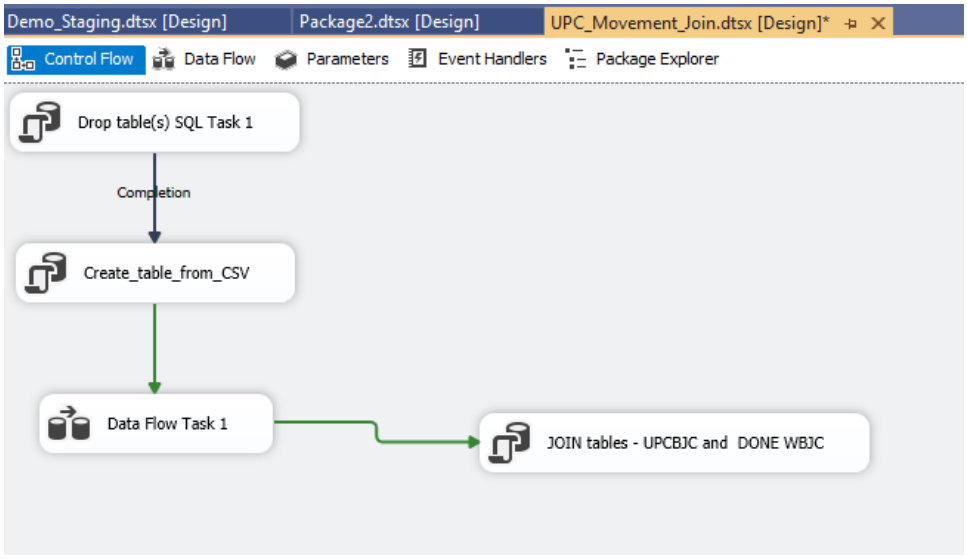


Figure: Execute SQL Task to JOIN the UPC and Movement tables

We are extracting the data from the UPCJBC and Movement table - DONE WBJC in the SSMS and joining it to make the final staging table for Coupons.

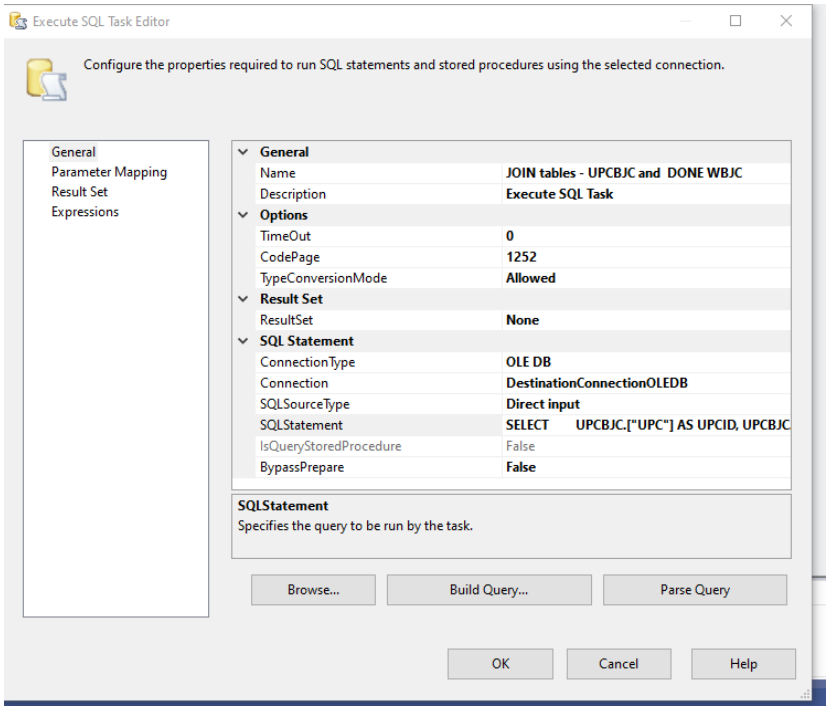


Figure: Mapping containing the SQL query to JOIN the two tables

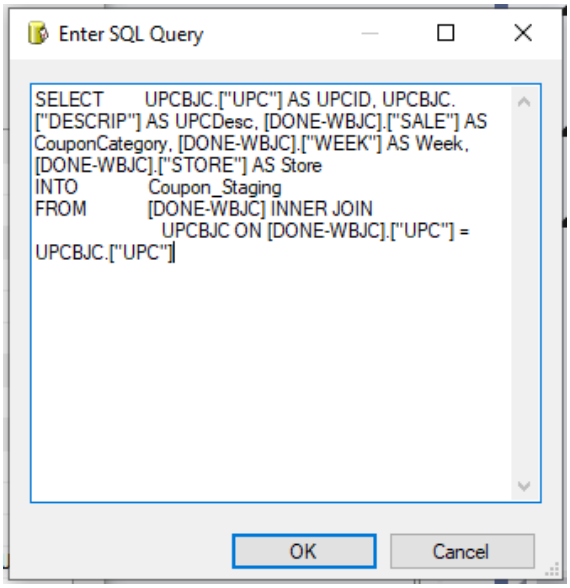


Figure: SQL query to perform JOIN operation

After building the query, you can define the target table as Coupon_Staging table. Here, we looked at the mapping and chose the desired number of attributes from the tables being joined to form the target staging table.

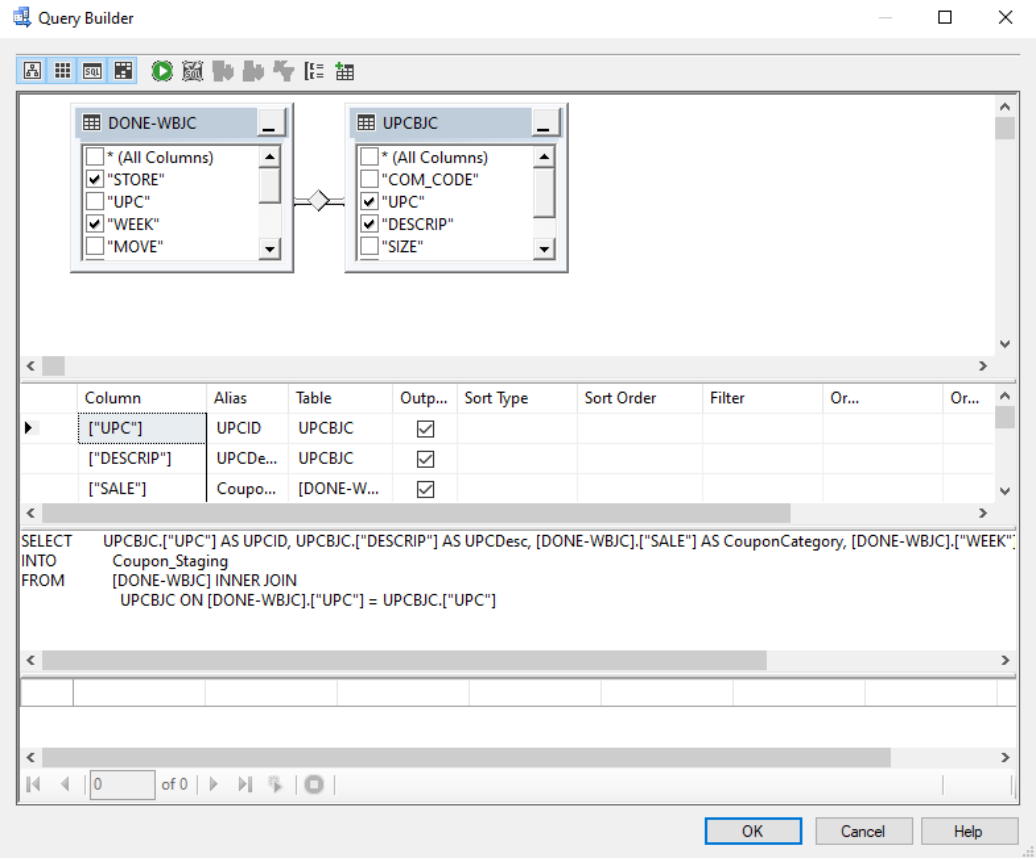


Figure: Mapping containing the SQL query to JOIN the two tables

Run an execute task to join tables to load the Coupon_Staging table:

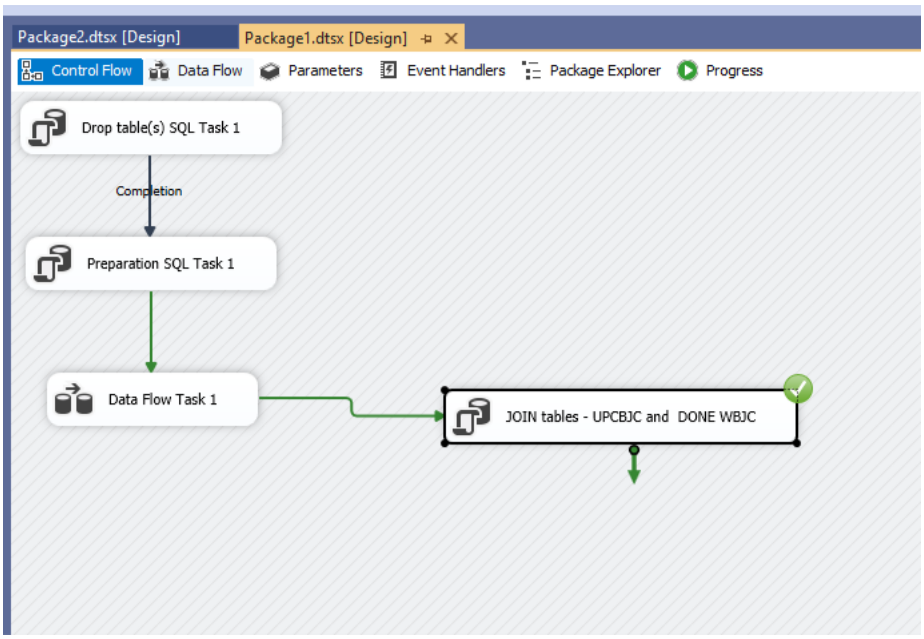


Figure: Running the Execute SQL task to merge two tables

The table Coupon_Staging has been successfully loaded in SSMS.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [UPCID]
      ,[UPCDesc]
      ,[CouponCategory]
      ,[Week]
      ,[Store]
  FROM [ISTM_637_602_Group10_Staging].[dbo].[Coupon_Staging]
```

100 %

Results | Messages

	UPCID	UPCDesc	CouponCategory	Week	Store
1	1060840005	ENDURO GRAPEFRUIT/TA	B	180	104
2	1060840005	ENDURO GRAPEFRUIT/TA		181	104
3	1060840005	ENDURO GRAPEFRUIT/TA	B	182	104
4	1060840005	ENDURO GRAPEFRUIT/TA		183	104
5	1060840005	ENDURO GRAPEFRUIT/TA		184	104
6	1060840005	ENDURO GRAPEFRUIT/TA		185	104
7	1060840005	ENDURO GRAPEFRUIT/TA		186	104
8	1060840005	ENDURO GRAPEFRUIT/TA		187	104
9	1060840005	ENDURO GRAPEFRUIT/TA		188	104
10	1060840005	ENDURO GRAPEFRUIT/TA		189	104
11	1060840005	ENDURO GRAPEFRUIT/TA		190	104
12	1060840005	ENDURO GRAPEFRUIT/TA		191	104
13	1060840005	ENDURO GRAPEFRUIT/TA		192	104
14	1060840005	ENDURO GRAPEFRUIT/TA		193	104

Figure: Coupon_Staging in SSMS

Sales_Staging Table

Data extraction for the Sales_Staging table was done using the two csv files namely, CCount.csv and Week Decode.csv. The tables were merged by performing a JOIN operation on them to retrieve the relevant products and the dates.

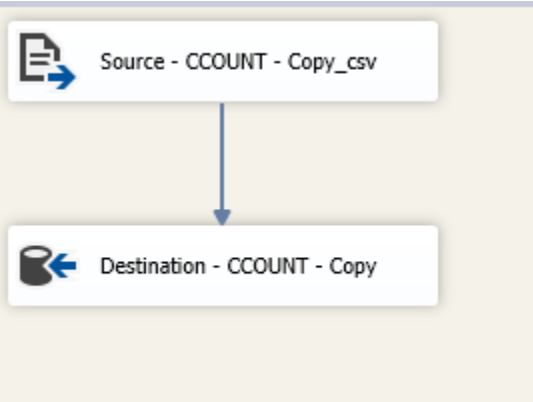


Figure: Extracting CCount-Copy.csv

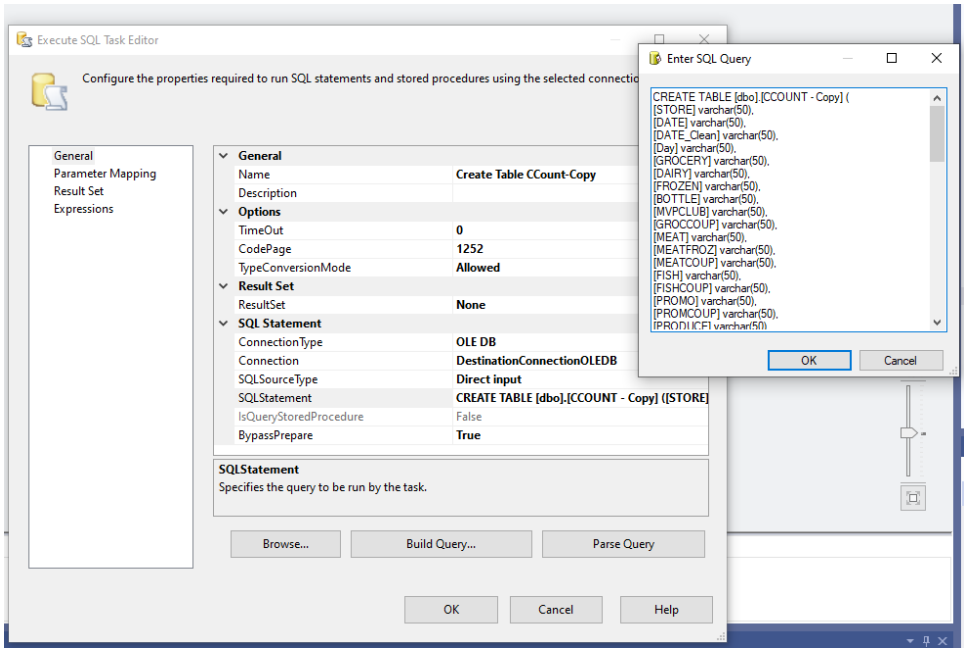


Figure: SQL Create query for extracting Products from CCount

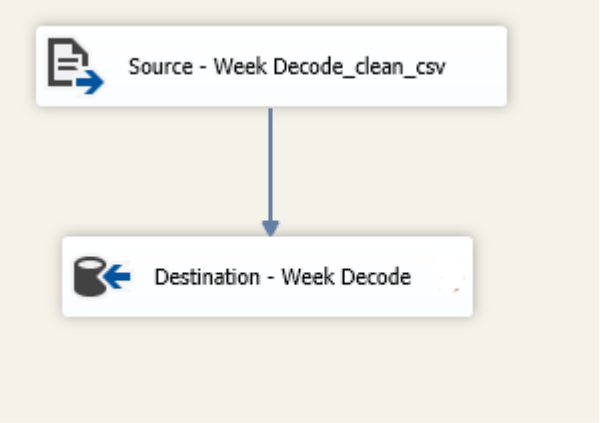


Figure: Extracting Week Decode.csv

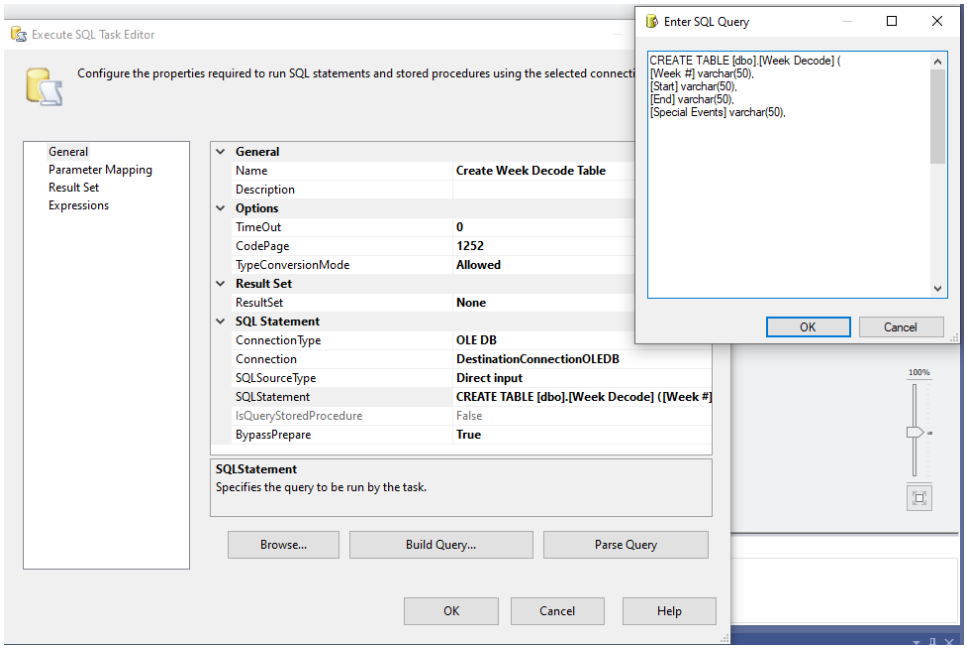


Figure: SQL Create query for extracting Products from Week Decode

After extracting, the tables were cleaned by executing SQL task flow. During the cleaning process, Null values as well as blank values were removed from the tables.

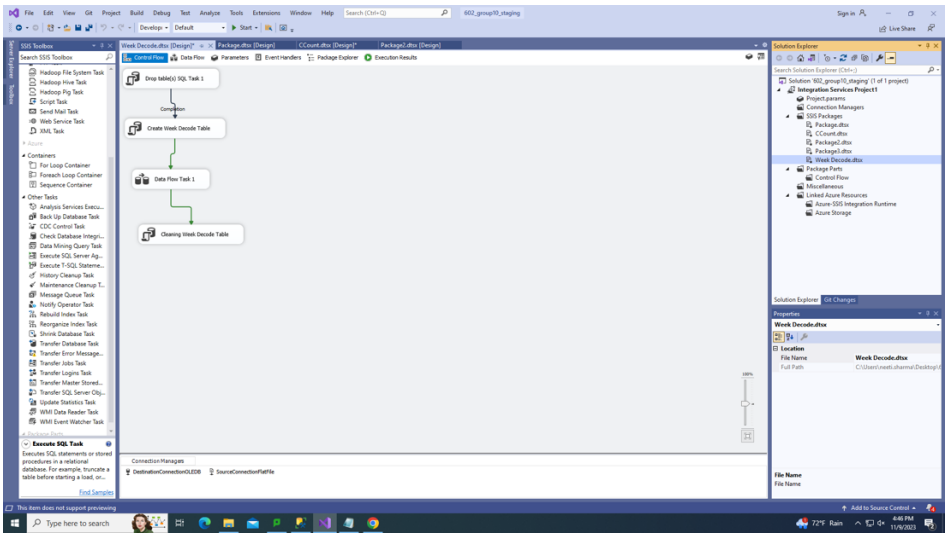


Figure: Package for cleaning Week Decode Table

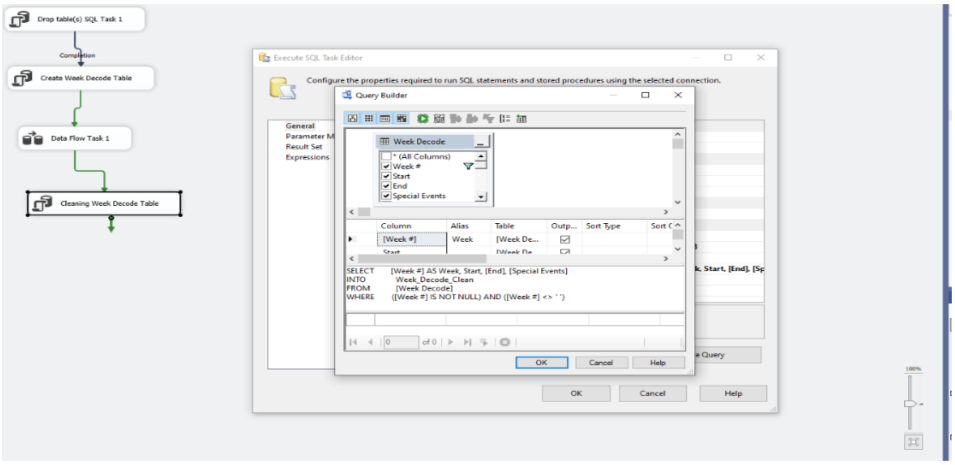


Figure: SQL query for cleaning Week Decode table

Now, once we have flat file data in the database, we can perform an execute SQL task to join the tables.

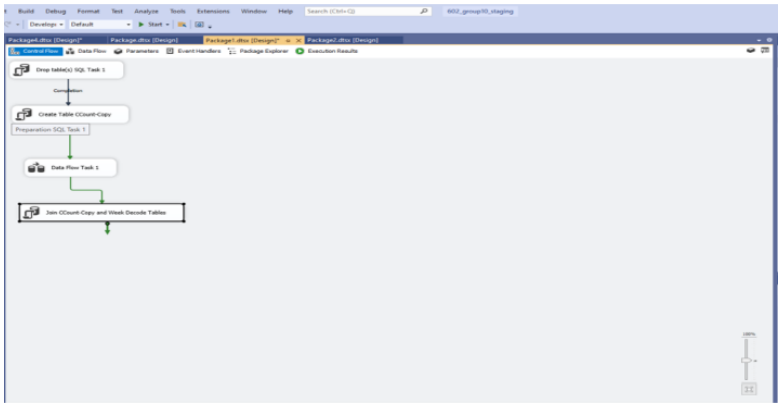


Figure: Package for Joining tables in SSIS

We are extracting the data from the CCount and Week_Decode_Clean table in the SSMS and joining it to make the final staging table for Product sales.

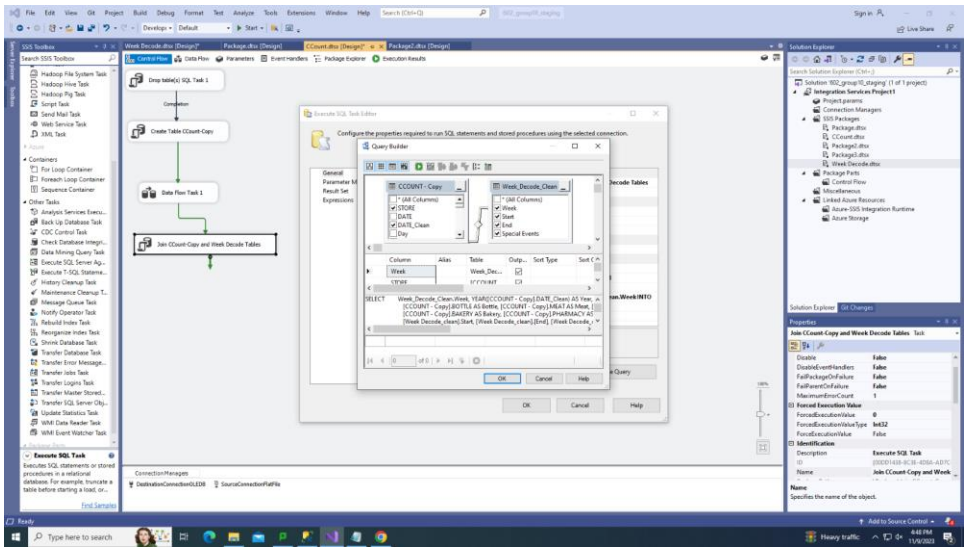


Figure: Joining the Week Decode Clean and CCount Table

After building the query, the target table is defined as Sales_Staging table. In the above screenshot we looked at the mapping and chose the desired number of attributes from the tables being joined to form the target staging table.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Week]
      ,[Year]
      ,[Store]
      ,[Grocery]
      ,[Dairy]
      ,[Frozen]
      ,[Bottle]
      ,[Meat]
      ,[Fish]
      ,[Floral]
      ,[Deli]
      ,[Cheese]
      ,[Bakery]
      ,[Pharmacy]
      ,[Jewelry]
      ,[Beer]
      ,[Wine]
      ,[SPIRITS]
      ,[Camera]
      ,[Saladbar]
      ,[Cosmetic]
      ,[ConvFood]
      ,[Start]
      ,[End]
      ,[Special Events]
```

Results | Messages

	Week	Year	Store	Grocery	Dairy	Frozen	Bottle	Meat	Fish	Floral	Deli	Cheese	Bakery	Pharmacy	Jewelry	Beer	Wine	SPIRITS	Camera	Saladbar	Cosmetic	ConvFood	Start	End	Special Events
1	250	1994	101	23306.27	4964.6	4342.38	0	5204.15	643.15	364.5	2330.73	146.67	1780.08	2665.87	0	1020.28	366.06	482.34	129.26	461.56	242.03	142.7	6/23/1994	6/29/1994	
2	250	1994	101	27255.65	5735.66	5468.04	0	6257.77	708.94	742.32	2936.63	214.11	1939.75	3074.66	0	1669.9	658.72	697.98	163.61	499.13	400.72	150.68	6/23/1994	6/29/1994	
3	250	1994	101	27927.31	5944.99	5658.06	0	7070.66	736.09	617.82	3385.8	186.7	2147.22	2443.75	0	1967.85	601.49	743.19	210.7	434.54	399.95	186.79	6/23/1994	6/29/1994	
4	250	1994	101	26191.56	5270.86	5095.66	1.7	5996.99	551.21	629.69	2998.85	176.52	1658.41	784.21	0	1273.73	461.79	497.98	265.55	380.57	403.59	137.21	6/23/1994	6/29/1994	
5	250	1994	101	19353.65	4119.47	3469.26	0	4359.41	378.18	269.93	2041.56	150.22	1193.4	3427.25	0	942.88	255.13	375.53	201.49	709.39	238.88	138.71	6/23/1994	6/29/1994	
6	250	1994	101	17165.77	3676.22	3559.81	0	3602.99	374.05	225.77	1887.12	113.22	1096.7	2110.5	0	729.08	209.1	305.67	104.03	620.64	166.26	132.72	6/23/1994	6/29/1994	
7	250	1994	101	16878.82	3490.52	3418.74	0	3862.85	269.37	290.2	1861.53	118.96	1258.2	2441.79	0	938.47	325.67	263.05	143.06	513.43	206.69	132.72	6/23/1994	6/29/1994	
8	252	1994	101	21680.57	4814.34	4561.33	0	5272.07	513.93	283.36	2686.85	226.62	1671.03	2989.61	0	945.53	427.49	418.25	157.69	625.17	312.49	113.26	7/7/1994	7/13/1994	
9	252	1994	101	23708.23	4934.83	4845.15	-0.6	6221.72	609.86	624.55	3056.47	275.05	2040.34	3742.37	0	1610.05	603.65	754.65	145.62	524.37	336.73	137.21	7/7/1994	7/13/1994	
10	252	1994	101	28484.32	6138.49	5818.1	-1.6	7921.46	505.03	440.77	3714.2	295.73	2396.75	2485.11	0	1717.65	547.74	835.02	198.37	419.6	544.82	178.11	7/7/1994	7/13/1994	
11	252	1994	101	24053.57	5173.67	4927.46	0	5547.25	351.83	452.42	2907.85	228.81	1477.55	984.22	0	1078.78	356.5	671.76	210.48	483.62	348.03	160.45	7/7/1994	7/13/1994	
12	252	1994	101	18954.8	3868.54	3682.07	0	4228.76	240.12	224.63	2197.27	144.3	1034.09	2772.18	0	678.77	250.39	392.78	152.23	665.03	244.38	127.73	7/7/1994	7/13/1994	
13	252	1994	101	17612.99	3871.59	3683.73	0	3709.41	312.97	177.46	2194.72	93.09	1126.08	2208.94	0	846.25	222.63	348.9	103.71	574.64	337.13	108.77	7/7/1994	7/13/1994	
14	252	1994	101	16535.69	3859.64	3885.85	0	4188.68	355.51	176.62	1793.45	159.97	1188.17	2116.08	0	906.85	297.84	492.55	57.69	506.55	237.58	122.74	7/7/1994	7/13/1994	
15	255	1994	101	23578.93	6950.73	4993.15	0	6123.47	681.48	276.12	2375.08	149.14	1868.26	2690.82	0	1133.29	525.33	615.83	165.1	526.06	168.31	134.73	7/28/1994	8/3/1994	
16	255	1994	101	25364.67	6765.62	5174.63	0	6624.19	752.33	412.17	2615.73	206.17	1908.44	1681.7	0	1670.86	462.56	577.27	134.26	550.41	343.22	163.17	7/28/1994	8/3/1994	
17	255	1994	101	29841.54	7853.69	6414.29	0.55	7608.84	650.65	475.85	3347.35	218.99	2298.65	2520.36	0	2092.43	668.58	753.45	138.42	383.31	290.97	150.08	7/28/1994	8/3/1994	
18	255	1994	101	23982.13	6130.51	5225.61	-1.6	5264	465.48	323.37	2930.79	177.19	1417.88	1427.51	0	1005.99	278.99	315.65	204.33	392.13	269.02	140.72	7/28/1994	8/3/1994	
19	255	1994	101	20183.1	5549.07	4942.2	0	4096.03	359	290.67	2462.96	117.05	1205.83	3416.45	0	670.68	226.15	316.16	157.86	663.67	270.96	186.63	7/28/1994	8/3/1994	

Figure: The table Sales_Staging has been successfully loaded in SSMS.

Demo_Staging Table

The following subsection shows the procedure for how we created the Demo_Staging table. We will explain it below in more detail

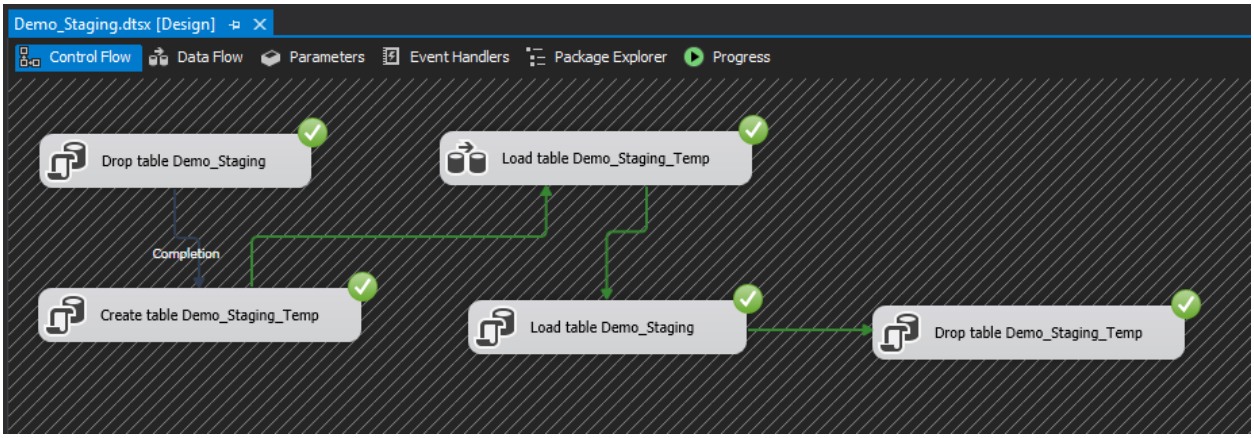


Figure: Overview of the Demo_Staging SSIS package.

The first step consists of dropping the Demo_Staging table, then, we create a Demo_Staging_Temp table to load all data from the Demo.CSV file to a SQL table. As shown below in the load process of Demo_Staging_Temp table.

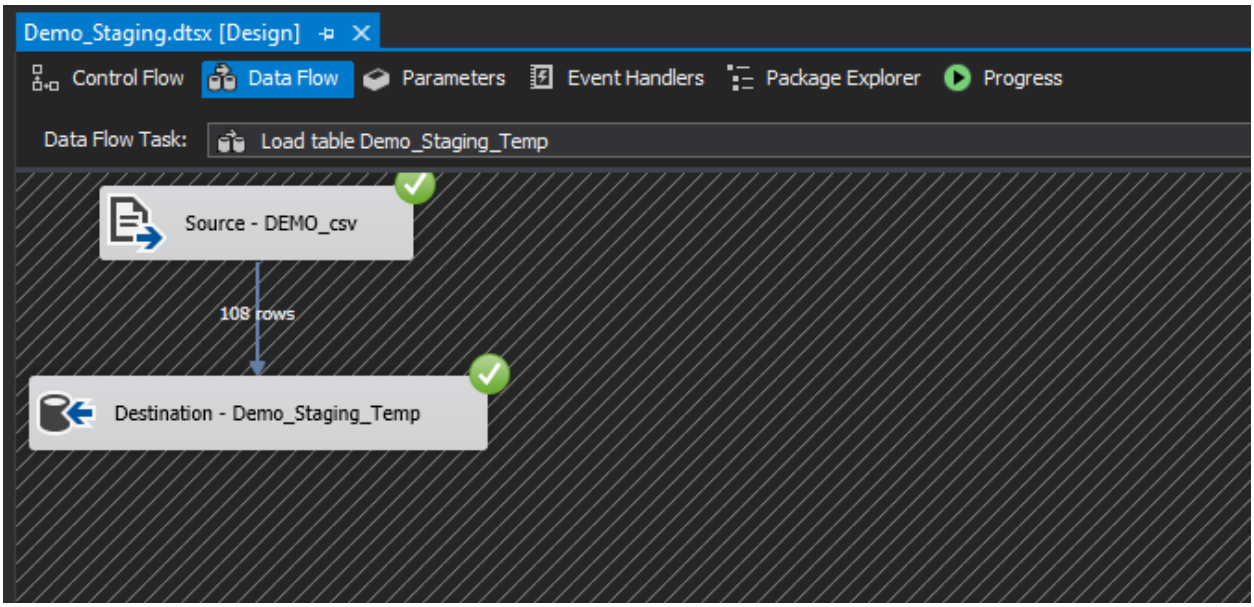


Figure: Overview of the Data Flow for Demo_Staging SSIS package.

The Data Flow shows that we read the Demo.CSV file from our computer and load all information into the Demo_Staging_Temp table.

Before loading Demo_Staging_Temp we create the table with all the attributes we have in the CSV file.

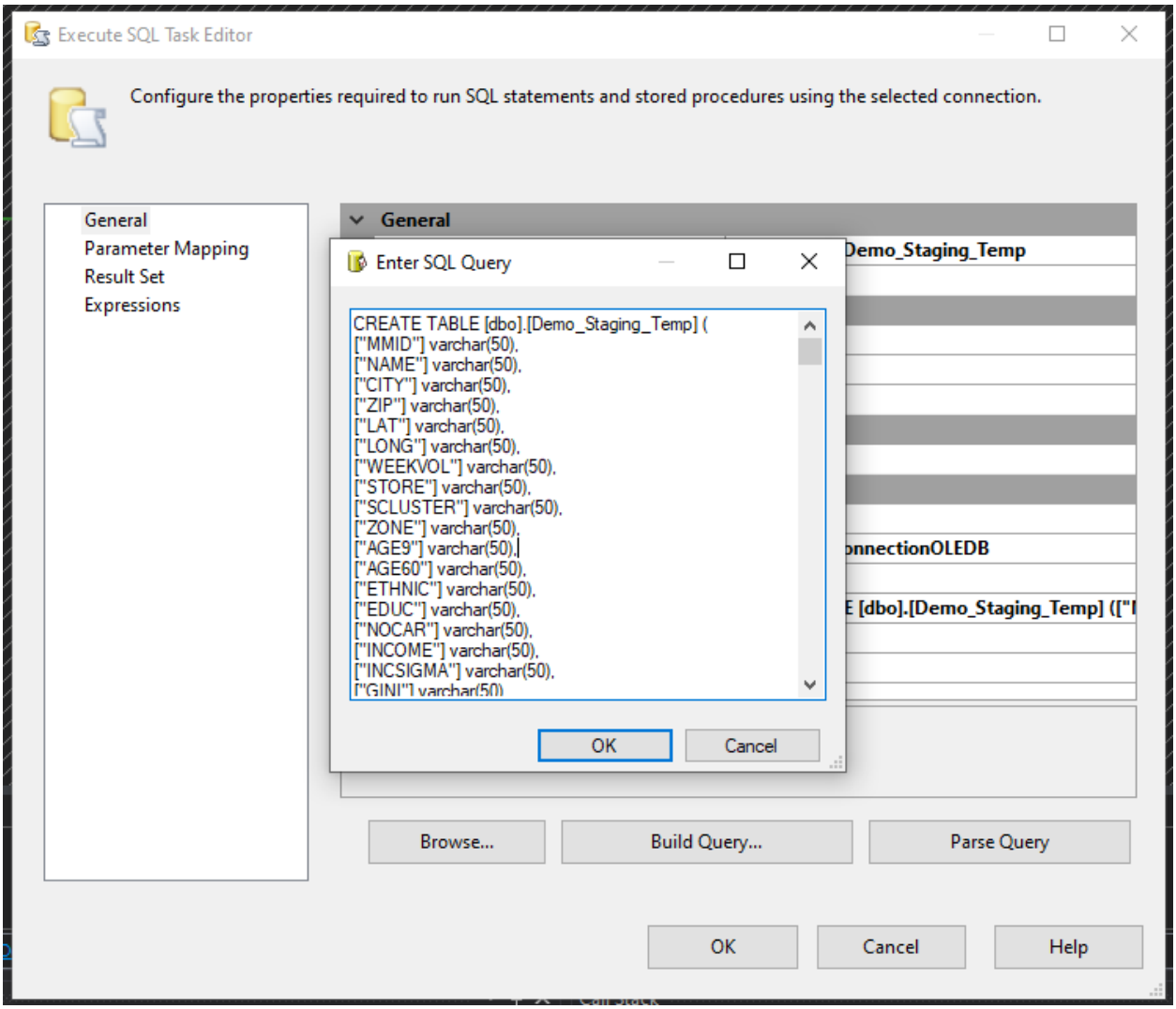


Figure: Create table query for Demo_Staging_Temp table

After that step, we only query the columns that we need and create a new table in the same step

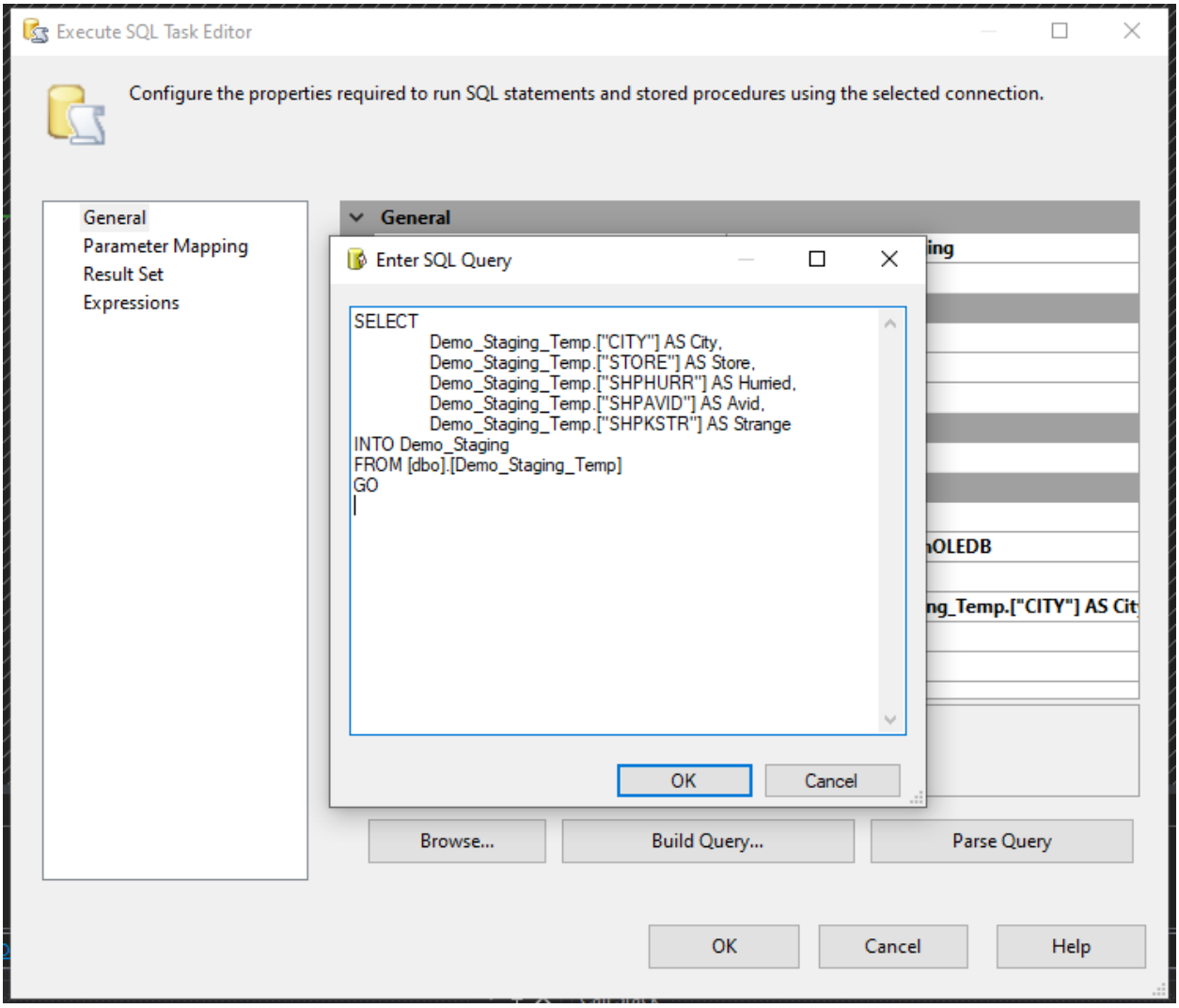


Figure: Select query with renaming and inserting values into a new table

We load the Demo_Staging table with the correct names that we want to use. Finally, we delete the Demo_Staging_Temp table and keep the Demo_Staging talbe with the data we need.

SQLQuery3.sql - in...nishq Dayma (211))

```
SELECT TOP (1000) [City]
      ,[Store]
      ,[Hurried]
      ,[Avid]
      ,[Strange]
  FROM [ISTM_637_602_Group10_staging_area].[dbo].[Demo_Staging]
```

100 %

Results | Messages

	City	Store	Hurried	Avid	Strange
1	""	.	.	.	.
2	"RIVER FOREST"	2	0.1210730674	0.1812566184	0.2322626191
3	"PARK RIDGE"	4	0.1183181914	0.1628987957	0.316923727
4	"PALATINE"	5	0.1906090191	0.1680613668	0.330195258
5	"OAK LAWN"	8	0.1351737791	0.2174439067	0.2161240651
6	"MORTON GROVE"	9	0.1586216924	0.1536783949	0.236405932
7	"CHICAGO"	12	0.0459478649	0.11045745	0.3630185617
8	"GLENVIEW"	14	0.215415685	0.1386738472	0.216425446
9	"RIVER GROVE"	18	0.1023112174	0.2156804144	0.231022116
10	""	19	.	.	.
11	"HANOVER PARK"	21	0.226978316	0.2502112081	0.29583216
12	""	25	.	.	.
13	"MOUNT PROSPECT"	28	0.165810369	0.1630079402	0.2659971976
14	"PARK RIDGE"	32	0.1147743664	0.1584586853	0.2906449619
15	"CHICAGO"	33	0.0263506356	0.0613082627	0.5577330508
16	""	39	.	.	.
17	"BRIDGEVIEW"	40	0.130373412	0.2380341332	0.2775567817
18	"WESTERN SPRINGS"	44	0.2046056654	0.1770939474	0.2052170369
19	"WHEELING"	45	0.1574519231	0.1756810897	0.3922275641
20	""	46	.	.	.
21	"ADDISON"	47	0.1686205549	0.2196170379	0.312622118
22	"SCHAUMBURG"	48	0.129397207	0.1384125862	0.4518295917
23	"DOWNERS GROVE"	49	0.1719101124	0.1896629213	0.304494382
24	"HICKORY HILLS"	50	0.1264604811	0.2258877434	0.3214203895
25	"PALOS HEIGHTS"	51	0.1429758936	0.2334164589	0.2661679135
26	"NORTHBROOK"	52	0.2107449857	0.1418338109	0.3134670487
27	"CHICAGO"	53	0.1429435484	0.1610887097	0.1951612903
28	"NAPERVILLE"	54	0.1014090173	0.1011601748	0.4036712419

Query executed successfully.

Figure: Demo_Staging in SSMS

ETL for Dimension Tables

UPC_Dim table creation

Creating a workflow to take data from the Coupon_Staging table and then creating a dimension table called Coupon_Dim. It will contain information on different types of coupon categories.

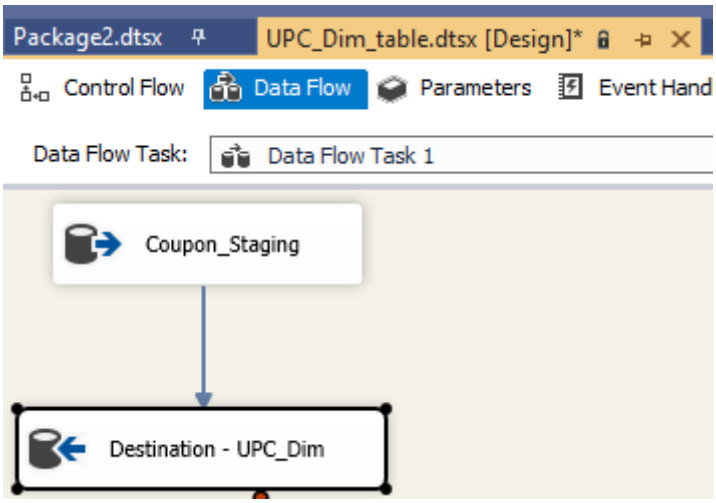


Figure: Extracting data from Staging table to UPC_Dim table

The below screenshot specifies the data that is present in the Coupon_Staging table in the staging_area database. This table will serve as the primary source for the Coupon_Purchases_Data_Mart.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [UPCID]
      ,[UPCDesc]
      ,[CouponCategory]
      ,[Week]
      ,[Store]
  FROM [ISTM_637_602_Group10_Staging].[dbo].[Coupon_Staging]
```

100 %

Results | Messages

	UPCID	UPCDesc	CouponCategory	Week	Store
1	1060840005	ENDURO GRAPEFRUIT/TA	B	180	104
2	1060840005	ENDURO GRAPEFRUIT/TA		181	104
3	1060840005	ENDURO GRAPEFRUIT/TA	B	182	104
4	1060840005	ENDURO GRAPEFRUIT/TA		183	104
5	1060840005	ENDURO GRAPEFRUIT/TA		184	104
6	1060840005	ENDURO GRAPEFRUIT/TA		185	104
7	1060840005	ENDURO GRAPEFRUIT/TA		186	104
8	1060840005	ENDURO GRAPEFRUIT/TA		187	104
9	1060840005	ENDURO GRAPEFRUIT/TA		188	104
10	1060840005	ENDURO GRAPEFRUIT/TA		189	104
11	1060840005	ENDURO GRAPEFRUIT/TA		190	104
12	1060840005	ENDURO GRAPEFRUIT/TA		191	104
13	1060840005	ENDURO GRAPEFRUIT/TA		192	104
14	1060840005	ENDURO GRAPEFRUIT/TA		193	104

Figure: Source will be the Coupon_Staging table

We took values from the Coupon_Staging table and mapped it to the UPC_Dim table. It will contain all the values except for the CouponCategory attribute field. The below screenshot shows the mapping between the two tables in the staging_area and dw_area databases. In addition to the columns that are present in the Coupon_Staging table we have an additional surrogate key column called UPCKey.

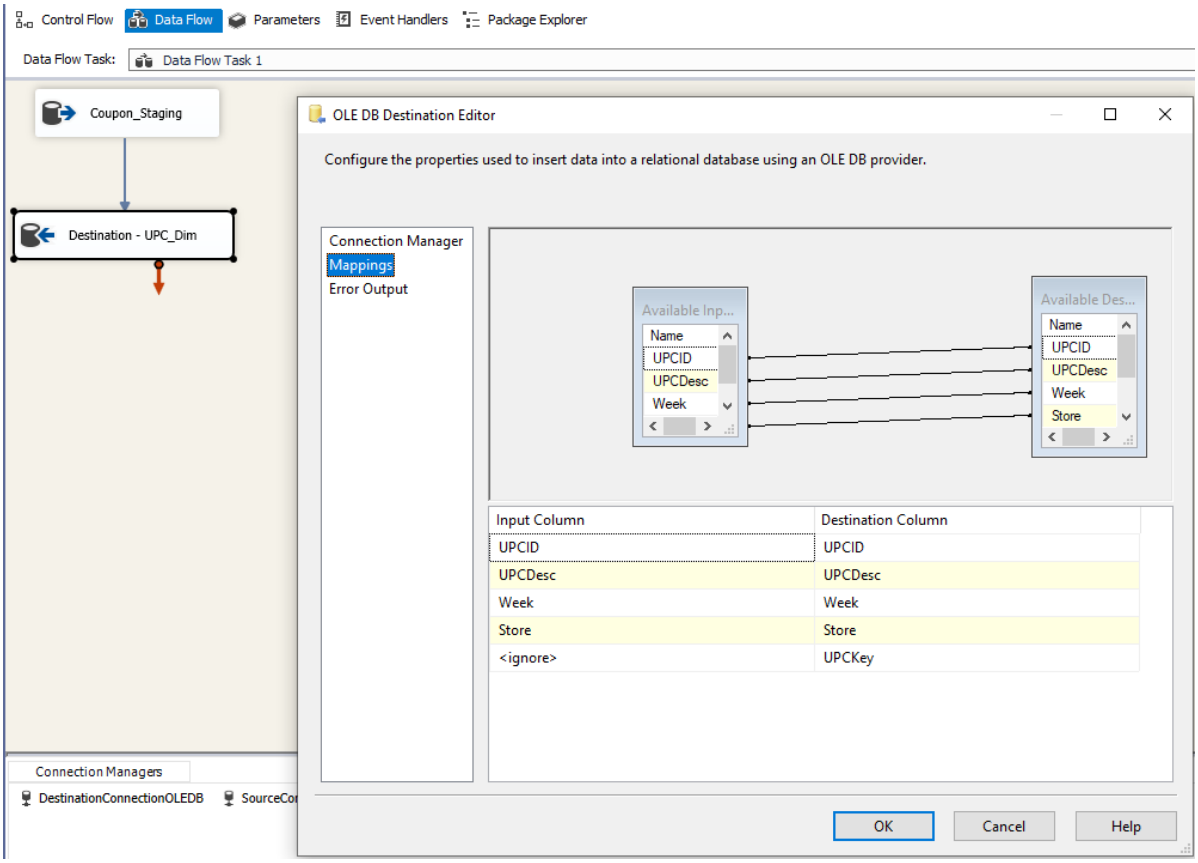


Figure: Mapping of data from source Staging table to Destination UPC dimension table

Here is the query of creating a table with an auto increment surrogate key for UPC_Dim Table.

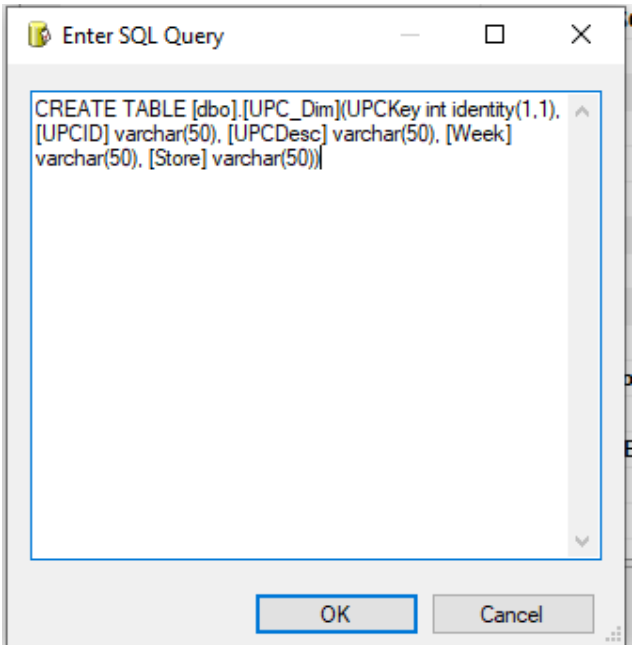


Figure: Create table query on the destination with the surrogate key

After the execution of this flow, we can get over 6 million rows in the UPC_Dim table containing the information about every UPCID of a brand of Juice.

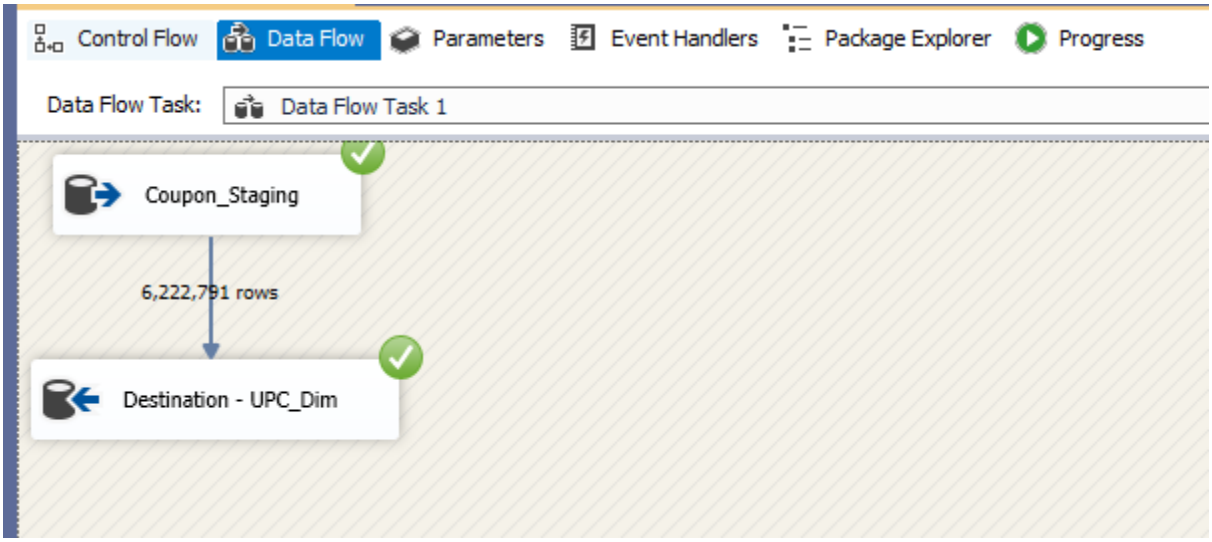
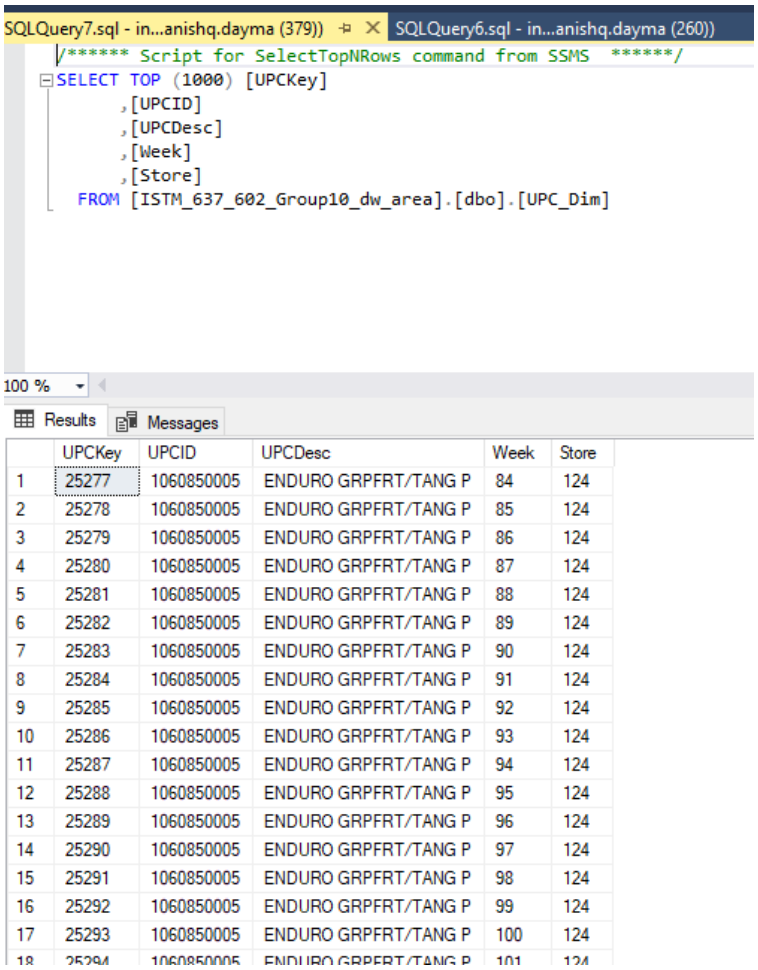


Figure: Loading and Transforming rows to the final dimension table



	UPCKey	UPCID	UPCDesc	Week	Store
1	25277	1060850005	ENDURO GRPFRT/TANG P	84	124
2	25278	1060850005	ENDURO GRPFRT/TANG P	85	124
3	25279	1060850005	ENDURO GRPFRT/TANG P	86	124
4	25280	1060850005	ENDURO GRPFRT/TANG P	87	124
5	25281	1060850005	ENDURO GRPFRT/TANG P	88	124
6	25282	1060850005	ENDURO GRPFRT/TANG P	89	124
7	25283	1060850005	ENDURO GRPFRT/TANG P	90	124
8	25284	1060850005	ENDURO GRPFRT/TANG P	91	124
9	25285	1060850005	ENDURO GRPFRT/TANG P	92	124
10	25286	1060850005	ENDURO GRPFRT/TANG P	93	124
11	25287	1060850005	ENDURO GRPFRT/TANG P	94	124
12	25288	1060850005	ENDURO GRPFRT/TANG P	95	124
13	25289	1060850005	ENDURO GRPFRT/TANG P	96	124
14	25290	1060850005	ENDURO GRPFRT/TANG P	97	124
15	25291	1060850005	ENDURO GRPFRT/TANG P	98	124
16	25292	1060850005	ENDURO GRPFRT/TANG P	99	124
17	25293	1060850005	ENDURO GRPFRT/TANG P	100	124
18	25294	1060850005	ENDURO GRPFRT/TANG P	101	124

Figure: UPC_Dim table successfully loaded in the database

Coupon_Dim table creation

Creating a workflow to take data from the Coupon_Staging table and then creating a dimension table called Coupon_Dim. It will contain information on different types of coupon categories.

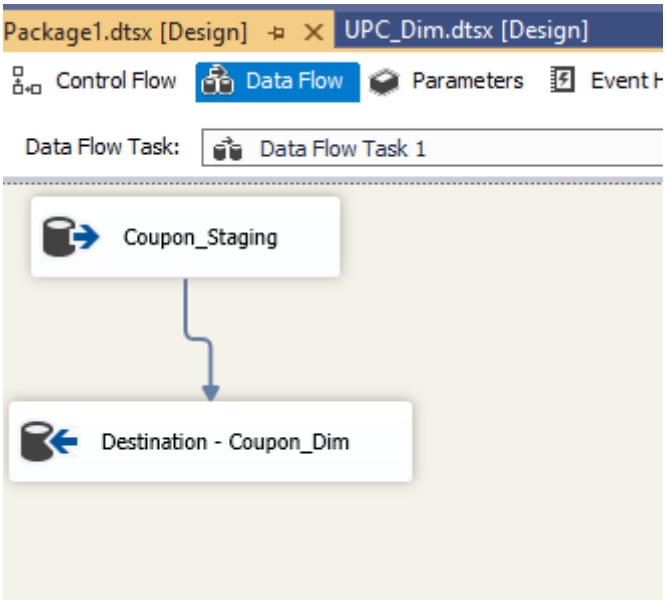


Figure: Extracting data from Staging table to Coupon_Dim table

The below screenshot specifies the data that is present in the Coupon_Staging table in the staging_area database. This table will serve as the primary source for the Coupon_Purchases_Data_Mart.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [UPCID]
      ,[UPCDesc]
      ,[CouponCategory]
      ,[Week]
      ,[Store]
  FROM [ISTM_637_602_Group10_Staging].[dbo].[Coupon_Staging]
```

100 %

Results | Messages

	UPCID	UPCDesc	CouponCategory	Week	Store
1	1060840005	ENDURO GRAPEFRUIT/TA	B	180	104
2	1060840005	ENDURO GRAPEFRUIT/TA		181	104
3	1060840005	ENDURO GRAPEFRUIT/TA	B	182	104
4	1060840005	ENDURO GRAPEFRUIT/TA		183	104
5	1060840005	ENDURO GRAPEFRUIT/TA		184	104
6	1060840005	ENDURO GRAPEFRUIT/TA		185	104
7	1060840005	ENDURO GRAPEFRUIT/TA		186	104
8	1060840005	ENDURO GRAPEFRUIT/TA		187	104
9	1060840005	ENDURO GRAPEFRUIT/TA		188	104
10	1060840005	ENDURO GRAPEFRUIT/TA		189	104
11	1060840005	ENDURO GRAPEFRUIT/TA		190	104
12	1060840005	ENDURO GRAPEFRUIT/TA		191	104
13	1060840005	ENDURO GRAPEFRUIT/TA		192	104
14	1060840005	ENDURO GRAPEFRUIT/TA		193	104

Figure: Source will be the Coupon_Staging table

We took values from the Coupon_Staging table and mapped it to the Coupon_Dim table. It will contain only the values of the CouponCategory attribute field. The below screenshot shows the mapping between the two tables in the staging_area and dw_area databases. In addition to the CouponCategory column that is present in the Coupon_Staging table we have an additional surrogate key column called CouponKey.

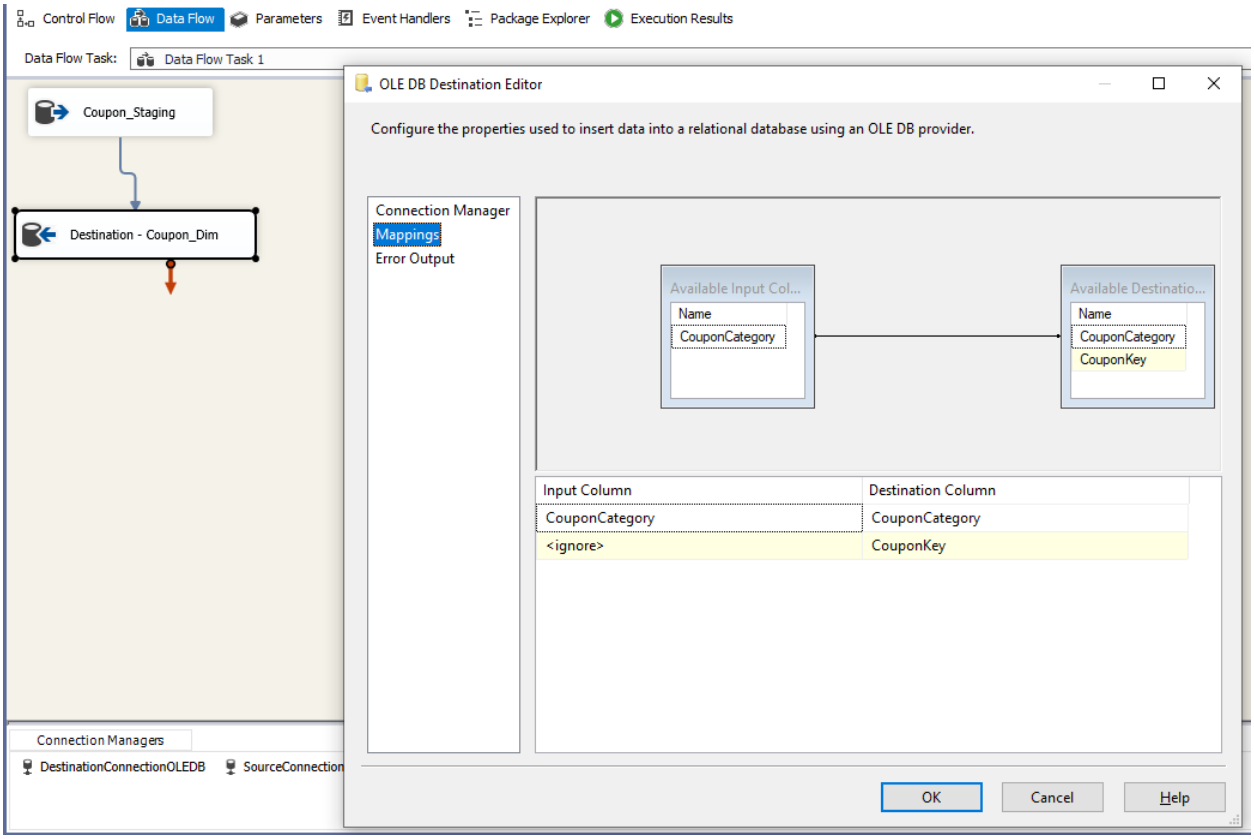


Figure: Mapping of data from source Staging table to Destination Coupon dimension table

Here is the query of creating a table with an auto increment surrogate key for Coupon_Dim Table.

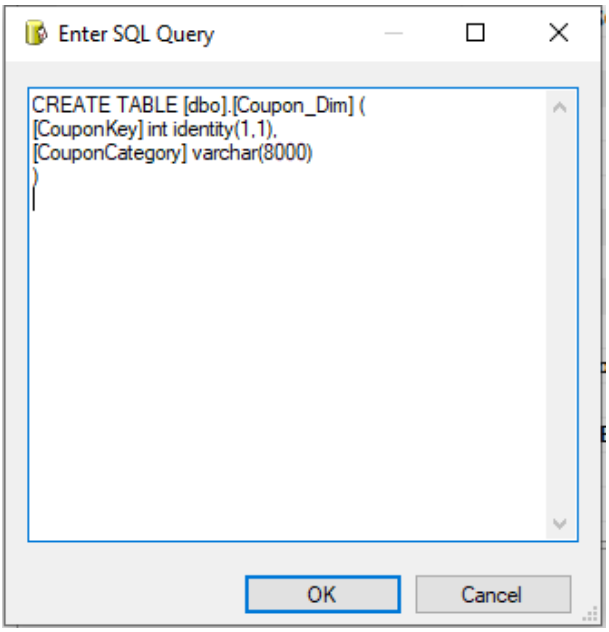


Figure: Create table query on the destination with the surrogate key

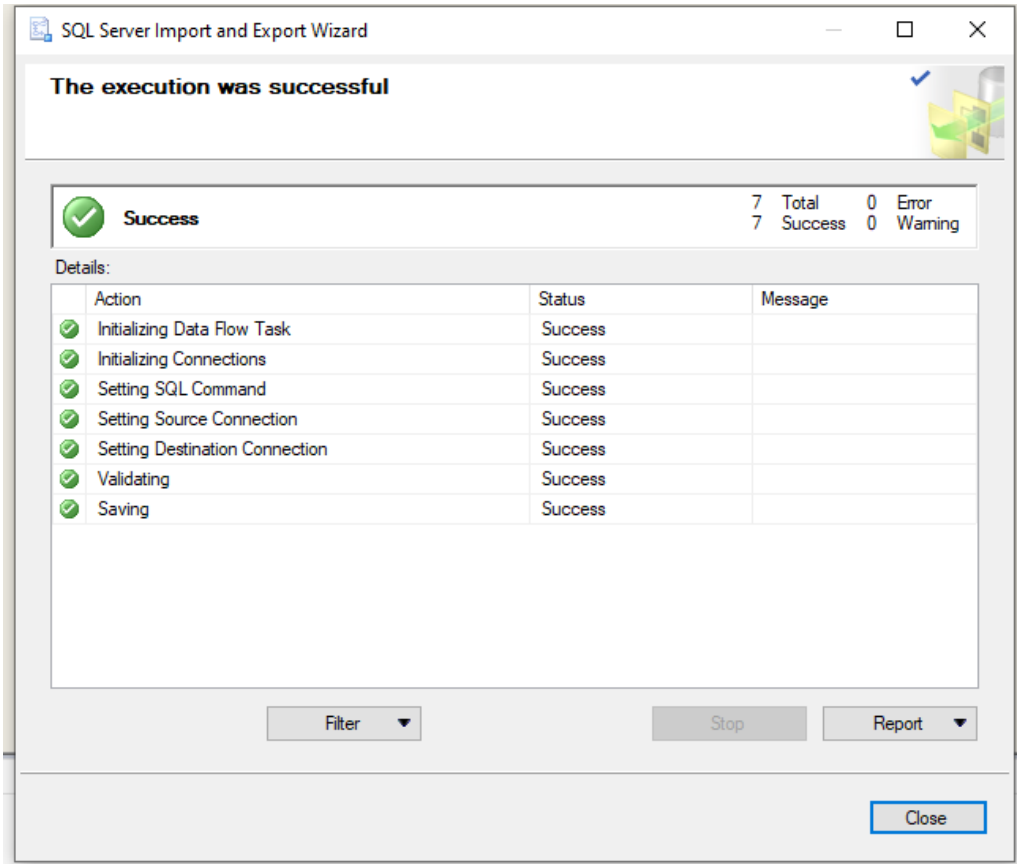


Figure: Execution successful to load the Coupon_Dim table

After the execution of this flow, we will be getting 4 rows in the Coupon_Dim table containing the information about every CouponCategory for a brand of Juice.

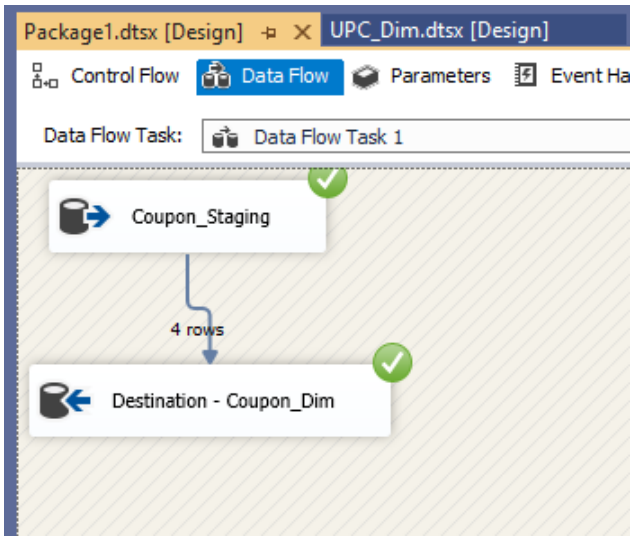


Figure: Loading and Transforming rows to the final dimension table

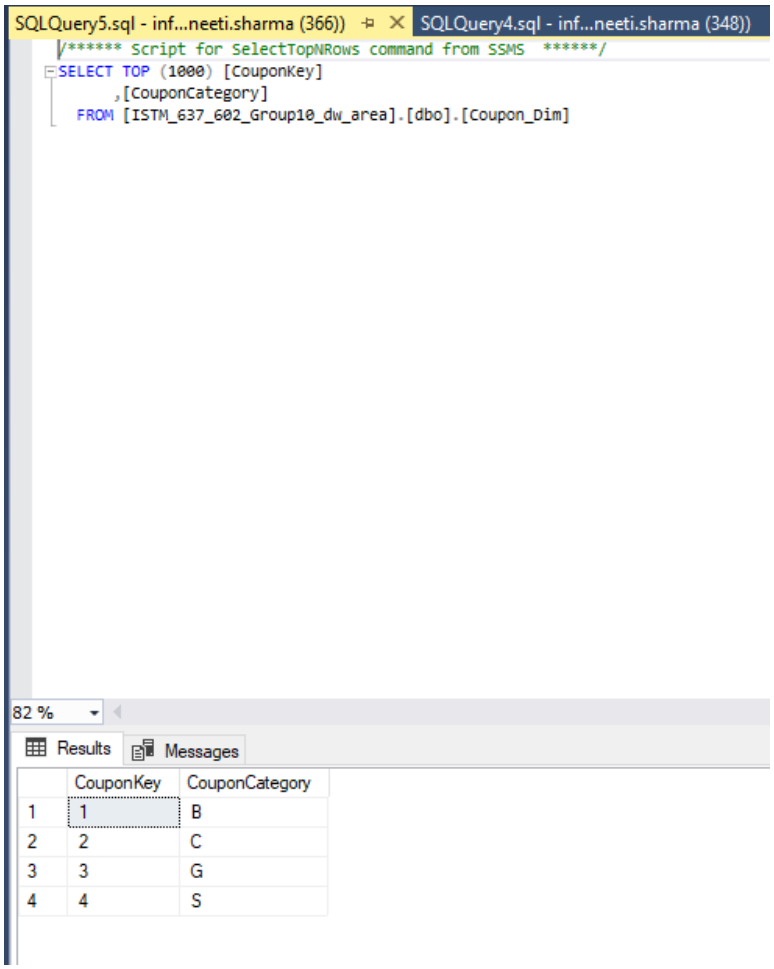


Figure: Coupon_Dim table successfully loaded in the database

Demo_Dim table creation

For the Demo_Dim table, we created an SSIS package that consists of the following overall processes. We first drop the current table, then we create the table again and finally we load the data into the newly created table

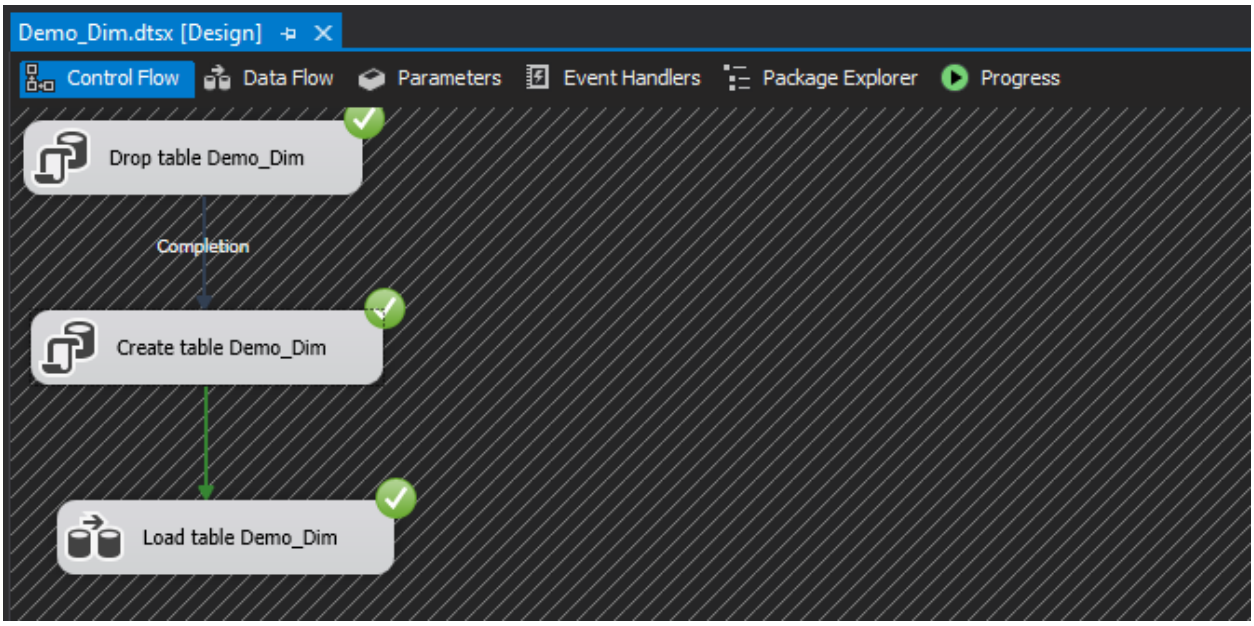


Figure: Overview of Demo_Dim package

As shown in the screenshot below, we have a CREATE TABLE query with the attributes we need. In this step we will also transform the data types of Store, Hurried, Avid and Strange and create a surrogate key called DemoKey.

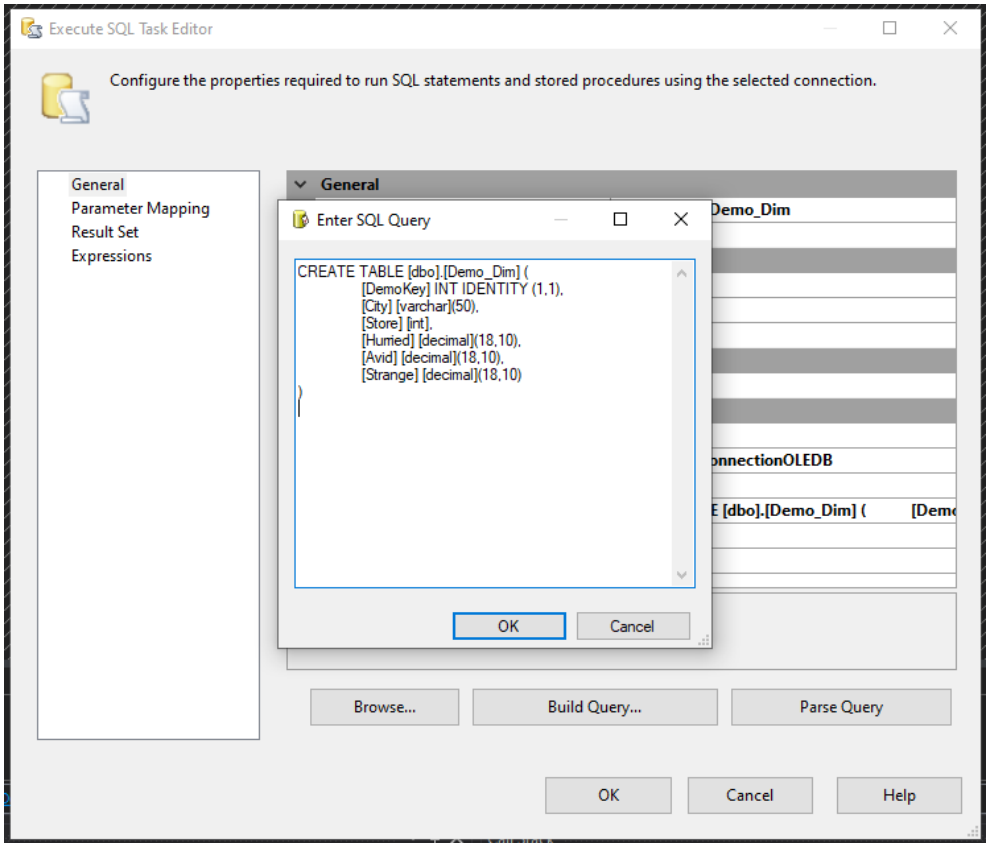


Figure: Create table query on the destination with the surrogate key

As shown in the Data Flow tab, the source data is a custom query, and the final destination is the newly created Demo_Dim table.

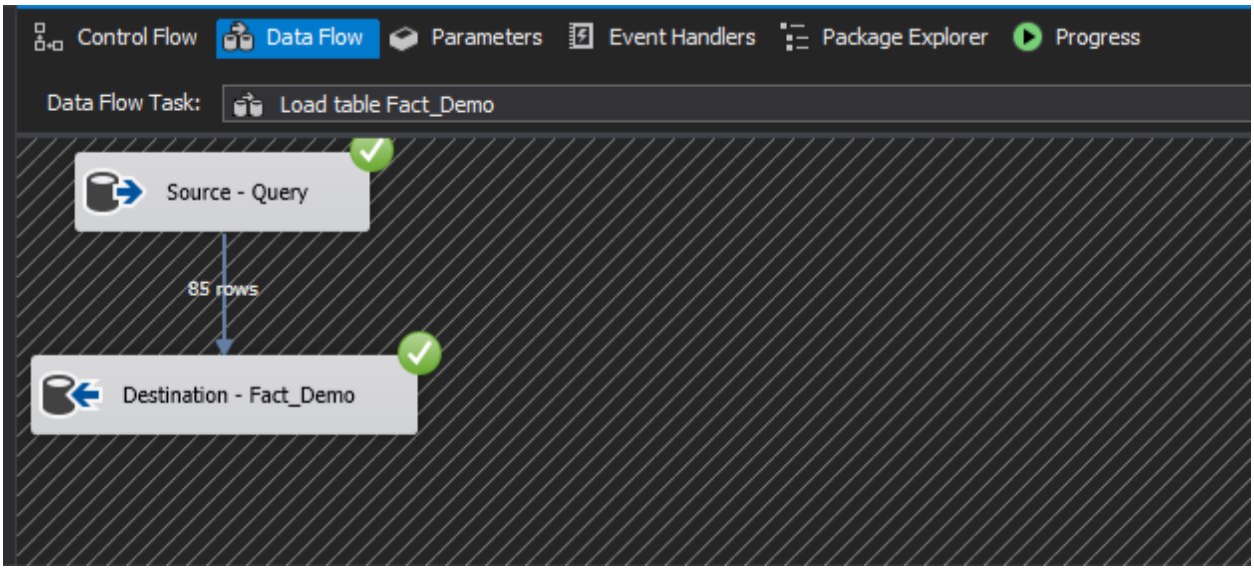


Figure: Loading and Transforming rows to the final dimension table

Following, we have the SELECT query that is in charge of performing the data transformation for Hurried, Avid and Store. They are originally varchar (50) but we need to use decimal (18,10). In this query we also clean the records where the City field is an empty string.

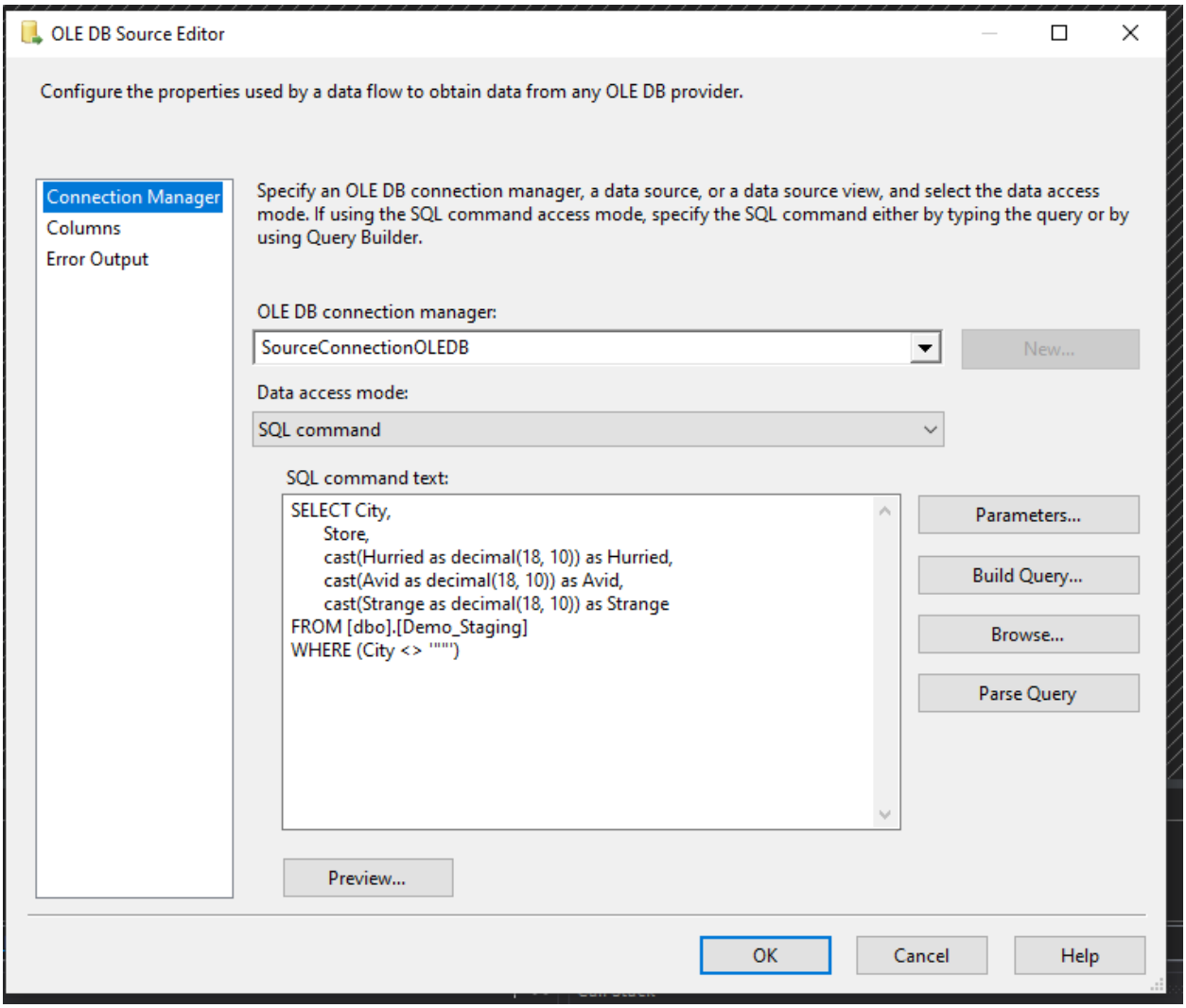


Figure: Select query used as source to load Demo_Dim

As shown in the mapping figure, we already create the correct name on the source query, and we will add the surrogate key of DemoKey to the destination table.

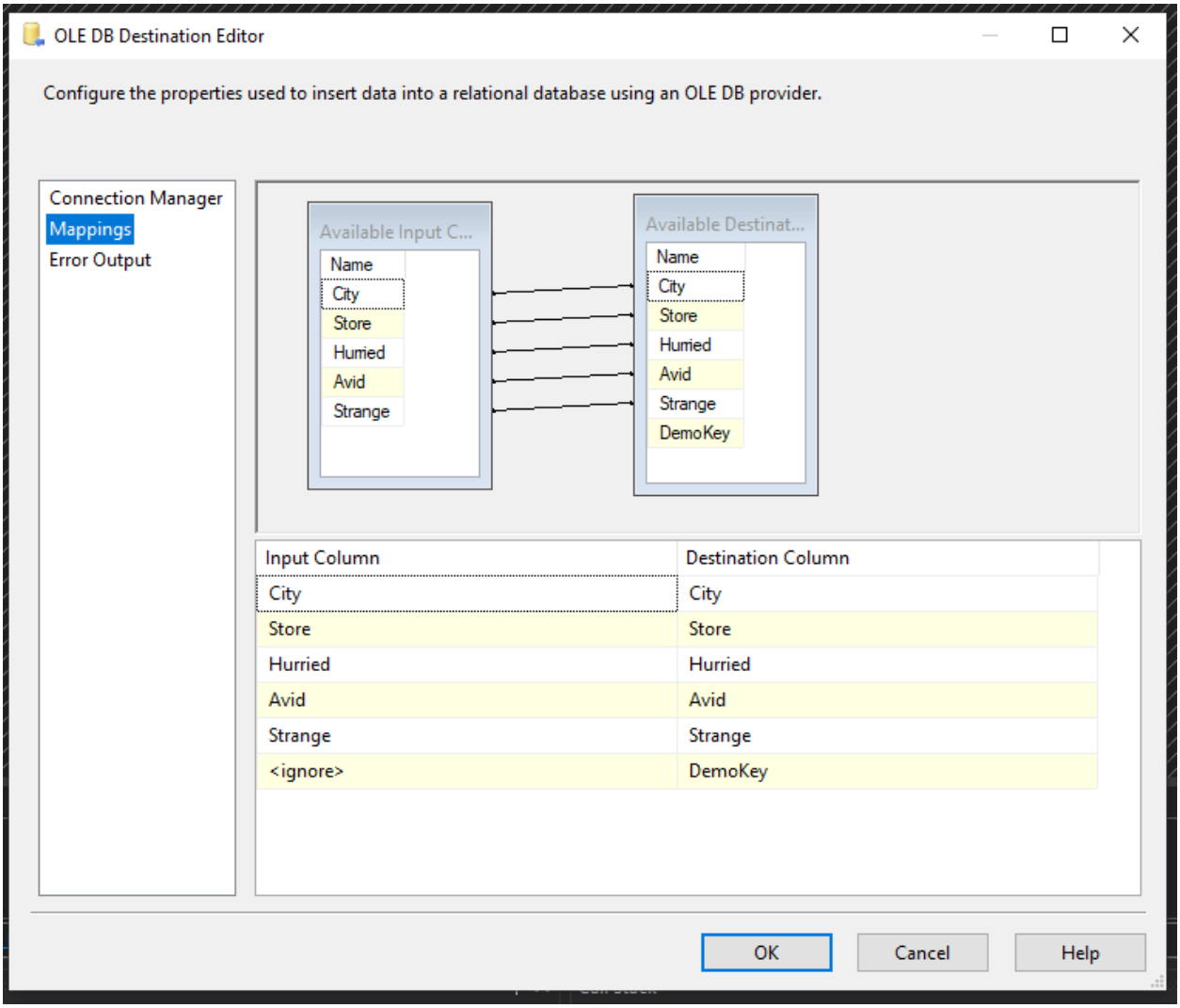


Figure: Mapping of the Demo_Dim table from source to destination

The last capture for Demo_Dim is a select query for all attributes to show that everything was imported correctly.

SQLQuery5.sql - in...nishq Dayma (329)) | SQLQuery4.sql - in...Tanishq Dayma (86))*

```
SELECT TOP (1000) [DemoKey]
      ,[City]
      ,[Store]
      ,[Hurried]
      ,[Avid]
      ,[Strange]
  FROM [ISTM_637_602_Group10_dw_area].[dbo].[Demo_Dim]
```

100 %

Results | Messages

	DemoKey	City	Store	Hurried	Avid	Strange
1	1	RIVER FOREST	2	0.1210730674	0.1812566184	0.2322626191
2	2	PARK RIDGE	4	0.1183181914	0.1628987957	0.3169237270
3	3	PALATINE	5	0.1906090191	0.1680613668	0.3301952580
4	4	OAK LAWN	8	0.1351737791	0.2174439067	0.2161240651
5	5	MORTON GROVE	9	0.1586216924	0.1536783949	0.2364059320
6	6	CHICAGO	12	0.0459478649	0.1104574500	0.3630185617
7	7	GLENVIEW	14	0.2154156850	0.1386738472	0.2164254460
8	8	RIVER GROVE	18	0.1023112174	0.2156804144	0.2310221160
9	9	HANOVER PARK	21	0.2269783160	0.2502112081	0.2958321600
10	10	MOUNT PROSPECT	28	0.1658103690	0.1630079402	0.2659971976
11	11	PARK RIDGE	32	0.1147743664	0.1584586853	0.2906449619
12	12	CHICAGO	33	0.0263506356	0.0613082627	0.5577330508
13	13	BRIDGEVIEW	40	0.1303734120	0.2380341332	0.2775567817
14	14	WESTERN SPRINGS	44	0.2046056654	0.1770939474	0.2052170369
15	15	WHEELING	45	0.1574519231	0.1756810897	0.3922275641
16	16	ADDISON	47	0.1686205549	0.2196170379	0.3126221180
17	17	SCHAUMBURG	48	0.1293972070	0.1384125862	0.4518295917
18	18	DOWNERS GROVE	49	0.1719101124	0.1896629213	0.3044943820
19	19	HICKORY HILLS	50	0.1264604811	0.2258877434	0.3214203895
20	20	PALOS HEIGHTS	51	0.1429758936	0.2334164589	0.2661679135
21	21	NORTHBROOK	52	0.2107449857	0.1418338109	0.3134670487
22	22	CHICAGO	53	0.1429435484	0.1610887097	0.1951612903
23	23	NAPERVILLE	54	0.1914999173	0.1911691748	0.4036712419
24	24	COUNTRYSIDE	56	0.1663859885	0.1756483665	0.2768608959
25	25	CRYSTAL LAKE	59	0.2375389408	0.2325545171	0.2496884735
26	26	NORTHFIELD	62	0.2404329928	0.1036743406	0.1837170300
27	27	VILLA PARK	64	0.1716158100	0.2262953103	0.2959246365

Query executed successfully.

Figure: Demo_Dim table successfully loaded in the database

Date_Dim table creation

Creating a workflow to take data from the Sales_Staging table and then creating a dimension table called Date_Dim.

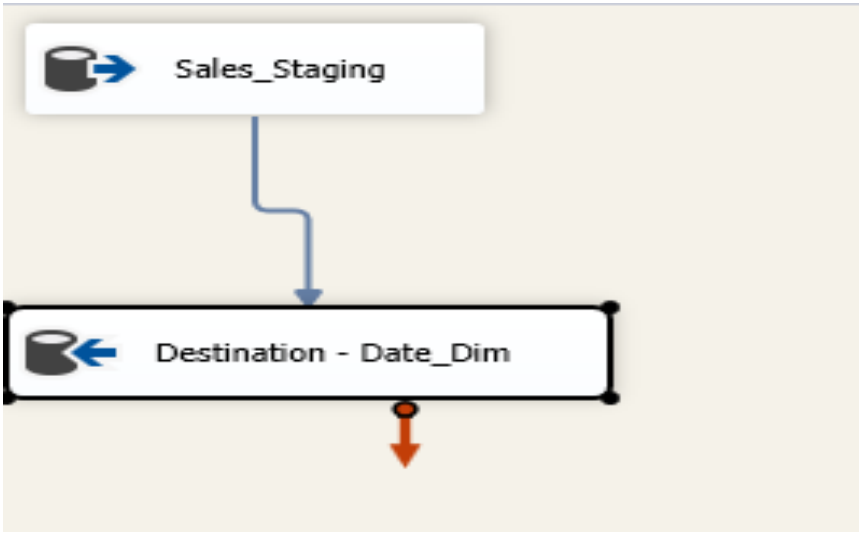
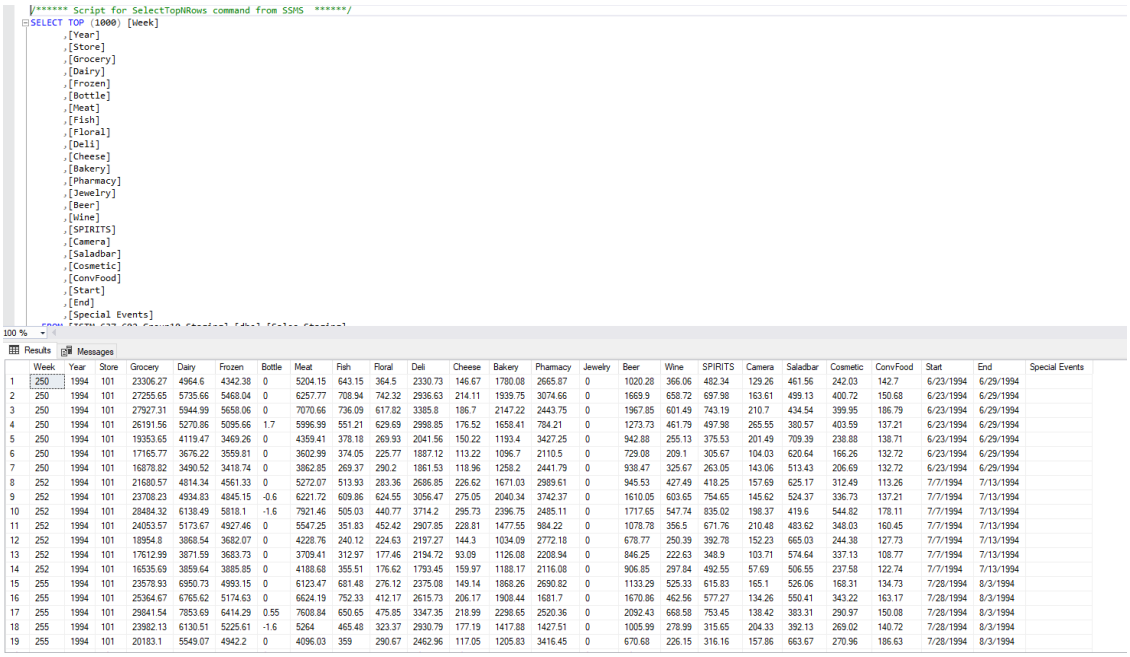


Figure: Extracting data from Sales_Staging table

The below screenshot specifies the data that is present in the Sales_Staging table in the staging_area database. This table will serve as the primary source for the Product_Sales_Data_Mart.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Week]
      ,[Year]
      ,[Store]
      ,[Grocery]
      ,[Dairy]
      ,[Frozen]
      ,[Bottle]
      ,[Meat]
      ,[Fish]
      ,[Floral]
      ,[Deli]
      ,[Cheese]
      ,[Bakery]
      ,[Pharmacy]
      ,[Jewelry]
      ,[Beer]
      ,[Wine]
      ,[SPIRITS]
      ,[Camera]
      ,[Saladbar]
      ,[Cosmetic]
      ,[ConvFood]
      ,[Start]
      ,[End]
      ,[Special Events]
```

Results Messages

	Week	Year	Store	Grocery	Dairy	Frozen	Bottle	Meat	Fish	Floral	Deli	Cheese	Bakery	Pharmacy	Jewelry	Beer	Wine	SPIRITS	Camera	Saladbar	Cosmetic	ConvFood	Start	End	Special Events
1	250	1994	101	23306.27	4964.6	4342.38	0	5204.15	643.15	364.5	2330.73	146.67	1780.08	2665.87	0	1020.28	366.06	482.34	129.26	461.56	242.03	142.7	6/23/1994	6/29/1994	
2	250	1994	101	27255.65	5735.66	5468.04	0	6257.77	708.94	742.32	2936.63	214.11	1939.75	3074.66	0	1669.9	658.72	697.98	163.61	499.13	400.72	150.68	6/23/1994	6/29/1994	
3	250	1994	101	27927.31	5944.99	5658.06	0	7070.66	736.09	617.82	3385.8	186.7	2147.22	2443.75	0	1967.85	601.49	743.19	210.7	434.54	399.95	186.79	6/23/1994	6/29/1994	
4	250	1994	101	26191.56	5270.86	5095.66	1.7	5996.99	551.21	629.69	2998.85	176.52	1658.41	784.21	0	1273.73	461.79	497.98	265.55	380.57	403.59	137.21	6/23/1994	6/29/1994	
5	250	1994	101	19353.65	4119.47	3469.26	0	4359.41	378.18	269.93	2041.56	150.22	1193.4	3427.25	0	942.88	255.13	375.53	201.49	709.39	238.88	138.71	6/23/1994	6/29/1994	
6	250	1994	101	17165.77	3676.22	3559.81	0	3602.99	374.05	225.77	1887.12	113.22	1096.7	2110.5	0	729.08	209.1	305.67	104.03	620.64	166.26	132.72	6/23/1994	6/29/1994	
7	250	1994	101	16878.82	3490.52	3418.74	0	3862.85	269.37	290.2	1861.53	118.96	1258.2	2441.79	0	938.47	325.67	263.05	143.06	513.43	206.69	132.72	6/23/1994	6/29/1994	
8	252	1994	101	21680.57	4814.34	4561.33	0	5272.07	513.93	283.36	2686.85	226.62	1671.03	2989.61	0	945.53	427.49	418.25	157.69	625.17	312.49	113.26	7/7/1994	7/13/1994	
9	252	1994	101	23708.23	4934.83	4845.15	-0.6	6221.72	609.86	624.55	3056.47	275.05	2040.34	3742.37	0	1610.05	603.65	754.65	145.62	524.37	336.73	137.21	7/7/1994	7/13/1994	
10	252	1994	101	28484.32	6138.49	5818.1	-1.6	7921.46	505.03	440.77	3714.2	295.73	2396.75	2485.11	0	1717.65	547.74	835.02	198.37	419.6	544.82	178.11	7/7/1994	7/13/1994	
11	252	1994	101	24053.57	5173.67	4927.46	0	5547.25	351.83	452.42	2907.85	228.81	1477.55	984.22	0	1078.78	356.5	671.76	210.48	483.62	348.03	160.45	7/7/1994	7/13/1994	
12	252	1994	101	18954.8	3868.54	3682.07	0	4228.76	240.12	224.63	2197.27	144.3	1034.09	2772.18	0	678.77	250.39	392.78	152.23	665.03	244.38	127.73	7/7/1994	7/13/1994	
13	252	1994	101	17612.99	3871.59	3683.73	0	3709.41	312.97	177.46	2194.72	93.09	1126.08	2208.94	0	846.25	222.63	348.9	103.71	574.64	337.13	108.77	7/7/1994	7/13/1994	
14	252	1994	101	16535.69	3859.64	3885.85	0	4188.68	355.51	176.62	1793.45	159.97	1188.17	2116.08	0	906.85	297.84	492.55	57.69	506.55	237.58	122.74	7/7/1994	7/13/1994	
15	255	1994	101	23578.93	6950.73	4993.15	0	6123.47	681.48	276.12	2375.08	149.14	1868.26	2690.82	0	1133.29	525.33	615.83	165.1	526.06	168.31	134.73	7/28/1994	8/3/1994	
16	255	1994	101	25364.67	6765.62	5174.63	0	6624.19	752.33	412.17	2615.73	206.17	1908.44	1681.7	0	1670.86	462.56	577.27	134.26	550.41	343.22	163.17	7/28/1994	8/3/1994	
17	255	1994	101	29841.54	7853.69	6414.29	0.55	7608.84	650.65	475.85	3347.35	218.99	2298.65	2520.36	0	2092.43	668.58	753.45	138.42	383.31	290.97	150.08	7/28/1994	8/3/1994	
18	255	1994	101	23982.13	6130.51	5225.61	-1.6	5264	465.48	323.37	2930.79	177.19	1417.88	1427.51	0	1005.99	278.99	315.65	204.33	392.13	269.02	140.72	7/28/1994	8/3/1994	
19	255	1994	101	20183.1	5549.07	4942.2	0	4096.03	359	290.67	2462.96	117.05	1205.83	3416.45	0	670.68	226.15	316.16	157.86	663.67	270.96	186.63	7/28/1994	8/3/1994	

Figure: Source will be the Sales_Staging table in SSMS

We took values from the Sales_Staging table and mapped it to the Date_Dim table. The below screenshot shows the mapping between the two tables in the staging_area and dw_area databases. In addition to the columns that are present in the Sales_Staging table we have an additional surrogate key column called DateKey.

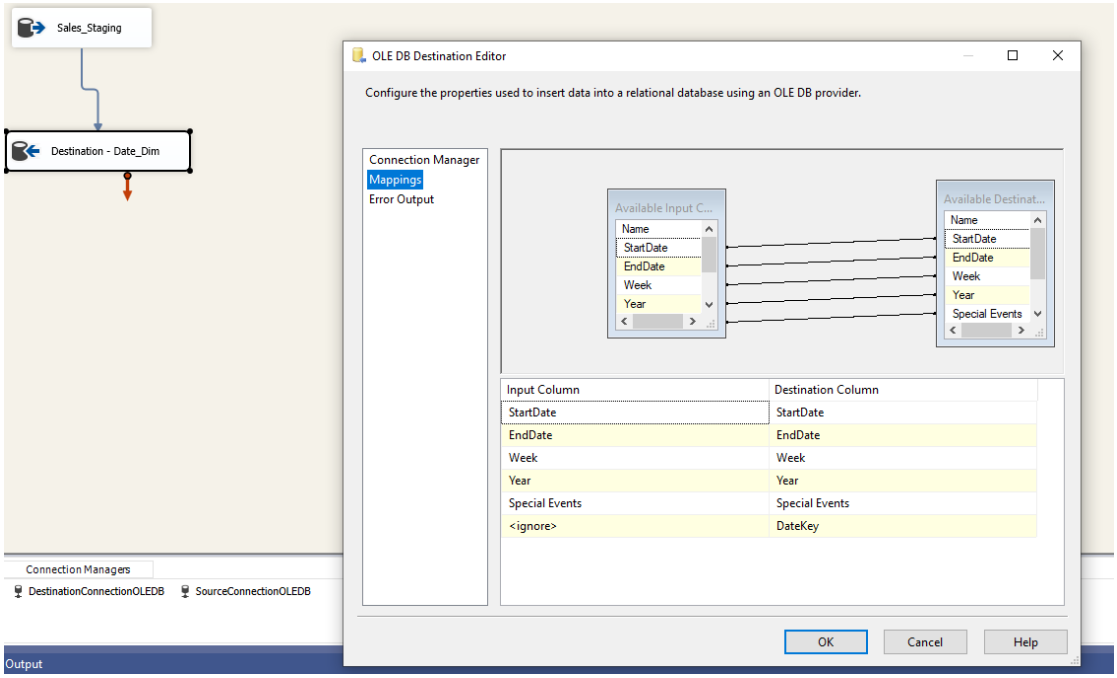


Figure: Mapping of data from source Staging table to Destination Date dimension table

The below table shows the query to create an auto increment surrogate key for Date_Dim table.

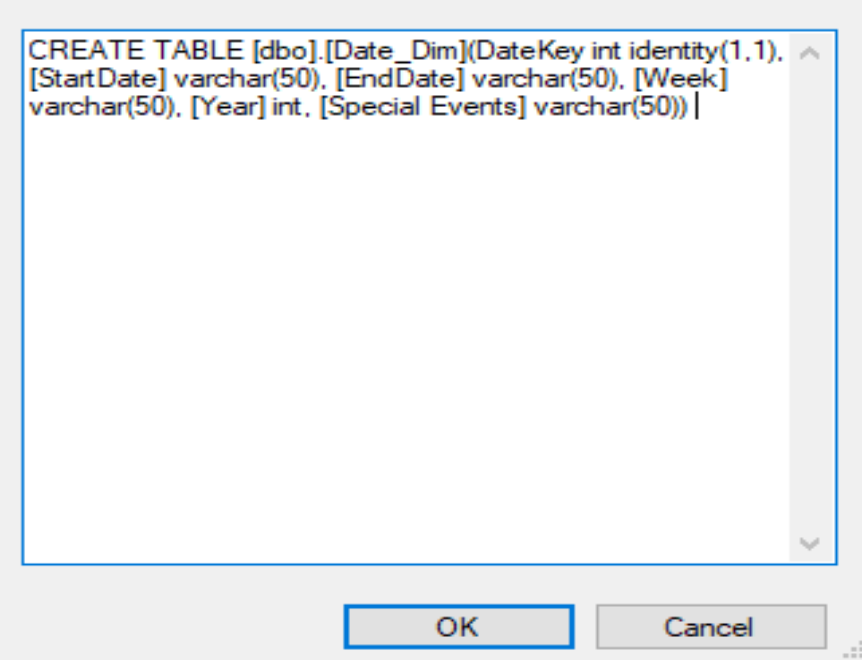


Figure: Create table query on the destination with the surrogate key

The flow is executed to load the table. Almost 406 rows were loaded in the Date_Dim table

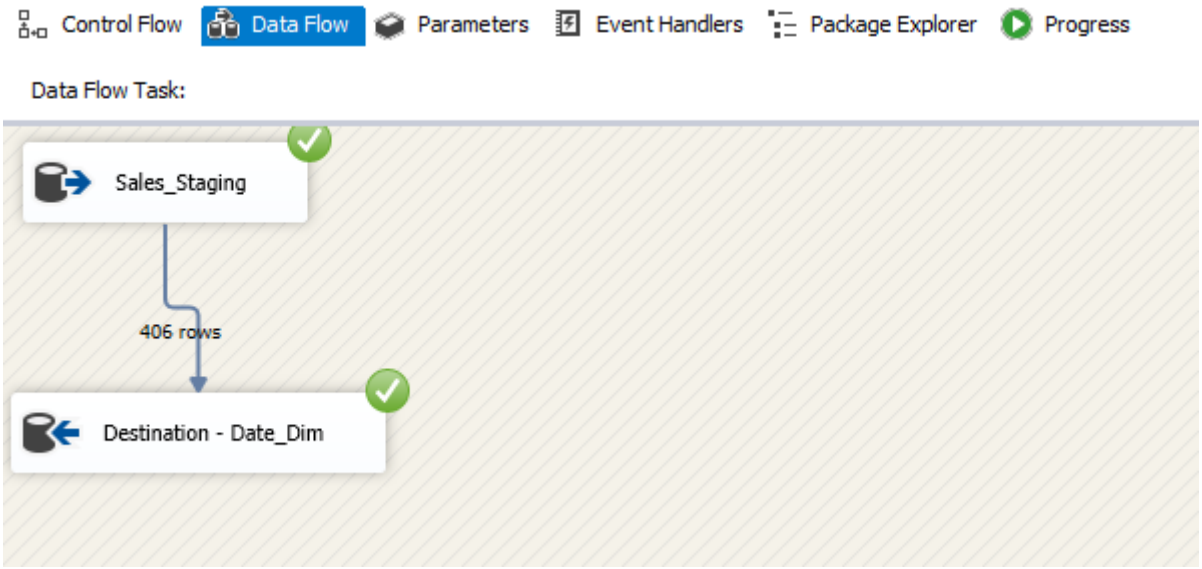
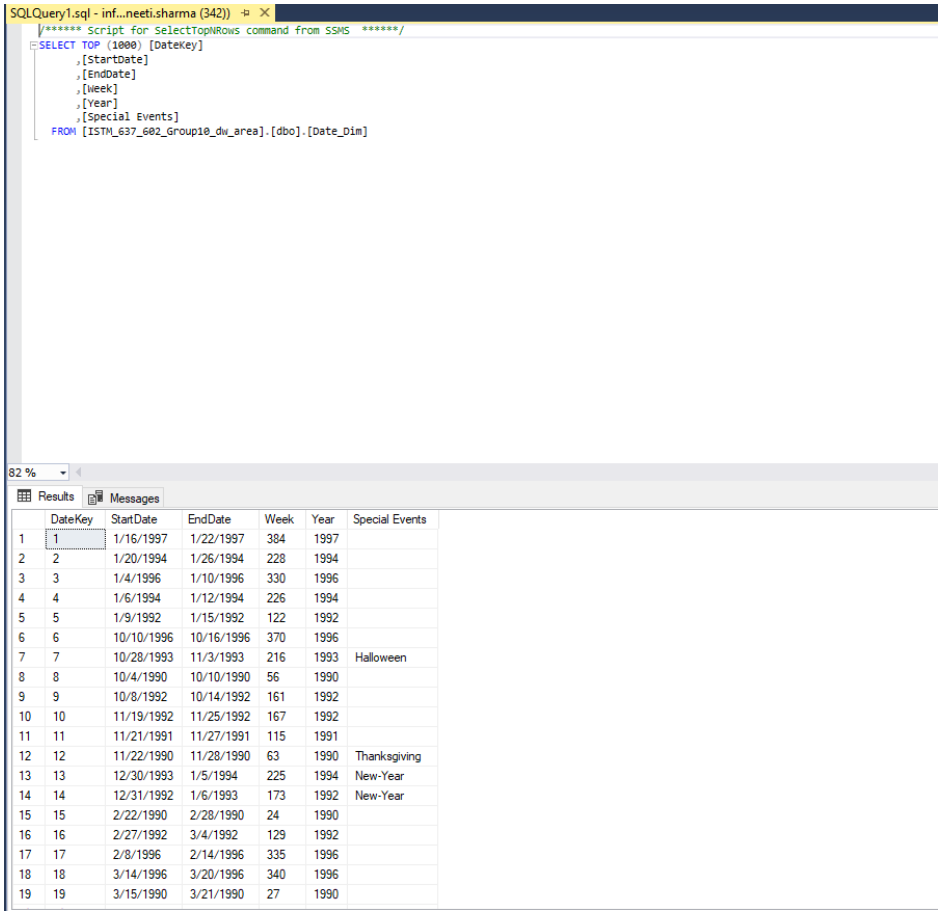


Figure: Loading and Transforming rows to the final dimension table

Data has been successfully loaded in the Date_Dim table in the SSMS.



	DateKey	StartDate	EndDate	Week	Year	Special Events
1	1	1/16/1997	1/22/1997	384	1997	
2	2	1/20/1994	1/26/1994	228	1994	
3	3	1/4/1996	1/10/1996	330	1996	
4	4	1/6/1994	1/12/1994	226	1994	
5	5	1/9/1992	1/15/1992	122	1992	
6	6	10/10/1996	10/16/1996	370	1996	
7	7	10/28/1993	11/3/1993	216	1993	Halloween
8	8	10/4/1990	10/10/1990	56	1990	
9	9	10/8/1992	10/14/1992	161	1992	
10	10	11/19/1992	11/25/1992	167	1992	
11	11	11/21/1991	11/27/1991	115	1991	
12	12	11/22/1990	11/28/1990	63	1990	Thanksgiving
13	13	12/30/1993	1/5/1994	225	1994	New-Year
14	14	12/31/1992	1/6/1993	173	1992	New-Year
15	15	2/22/1990	2/28/1990	24	1990	
16	16	2/27/1992	3/4/1992	129	1992	
17	17	2/8/1996	2/14/1996	335	1996	
18	18	3/14/1996	3/20/1996	340	1996	
19	19	3/15/1990	3/21/1990	27	1990	

Figure: Date_Dim Table successfully loaded in SSMS

Store_Dim table creation

Creating a workflow to take data from the Sales_Staging table and then creating a dimension table called Store_Dim.

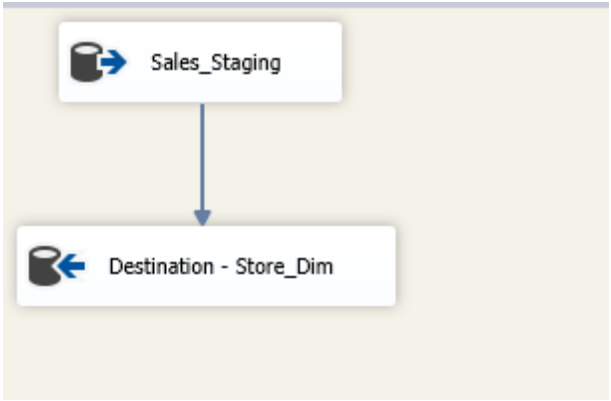
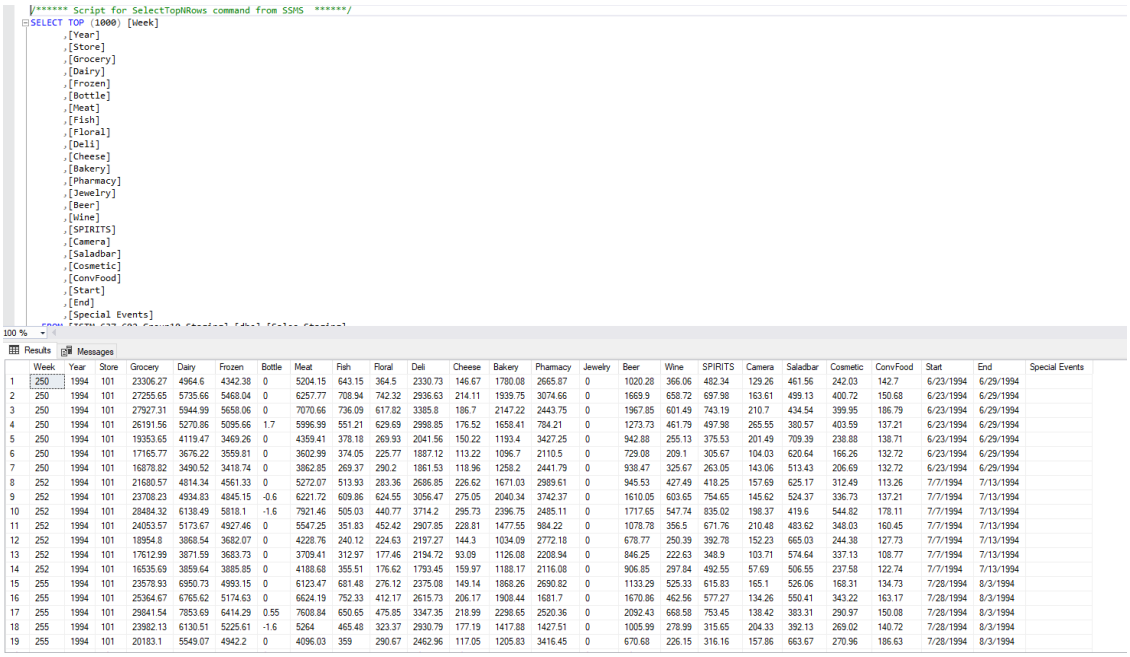


Figure: Extracting data from Sales_Staging table



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Week]
      ,[Year]
      ,[Store]
      ,[Grocery]
      ,[Dairy]
      ,[Frozen]
      ,[Bottle]
      ,[Meat]
      ,[Fish]
      ,[Floral]
      ,[Deli]
      ,[Cheese]
      ,[Bakery]
      ,[Pharmacy]
      ,[Jewelry]
      ,[Beer]
      ,[Wine]
      ,[SPIRITS]
      ,[Camera]
      ,[Saladbar]
      ,[Cosmetic]
      ,[ConvFood]
      ,[Start]
      ,[End]
      ,[Special Events]
```

100 % · Results · Messages

	Week	Year	Store	Grocery	Dairy	Frozen	Bottle	Meat	Fish	Floral	Deli	Cheese	Bakery	Pharmacy	Jewelry	Beer	Wine	SPIRITS	Camera	Saladbar	Cosmetic	ConvFood	Start	End	Special Events
1	250	1994	101	23306.27	4964.6	4342.38	0	5204.15	643.15	364.5	2330.73	146.67	1780.08	2665.87	0	1020.28	366.06	482.34	129.26	461.56	242.03	142.7	6/23/1994	6/29/1994	
2	250	1994	101	27255.65	5735.66	5468.04	0	6257.77	708.94	742.32	2936.63	214.11	1939.75	3074.66	0	1669.9	658.72	697.98	163.61	499.13	400.72	150.68	6/23/1994	6/29/1994	
3	250	1994	101	27927.31	5944.99	5658.06	0	7070.66	736.09	617.82	3385.8	186.7	2147.22	2443.75	0	1967.85	601.49	743.19	210.7	434.54	399.95	186.79	6/23/1994	6/29/1994	
4	250	1994	101	26191.56	5270.86	5095.66	1.7	5996.99	551.21	629.69	2998.85	176.52	1658.41	784.21	0	1273.73	461.79	497.98	265.55	380.57	403.59	137.21	6/23/1994	6/29/1994	
5	250	1994	101	19353.65	4119.47	3469.26	0	4359.41	378.18	269.93	2041.56	150.22	1193.4	3427.25	0	942.88	255.13	375.53	201.49	709.39	238.88	138.71	6/23/1994	6/29/1994	
6	250	1994	101	17165.77	3676.22	3559.81	0	3602.99	374.05	225.77	1887.12	113.22	1096.7	2110.5	0	729.08	209.1	305.67	104.03	620.64	166.26	132.72	6/23/1994	6/29/1994	
7	250	1994	101	16878.82	3490.52	3418.74	0	3862.85	269.37	290.2	1861.53	118.96	1258.2	2441.79	0	938.47	325.67	263.05	143.06	513.43	206.69	132.72	6/23/1994	6/29/1994	
8	252	1994	101	21680.57	4814.34	4561.33	0	5272.07	513.93	283.36	2686.85	226.62	1671.03	2989.61	0	945.53	427.49	418.25	157.69	625.17	312.49	113.26	7/7/1994	7/13/1994	
9	252	1994	101	23708.23	4934.83	4845.15	-0.6	6221.72	609.86	624.55	3056.47	275.05	2040.34	3742.37	0	1610.05	603.65	754.65	145.62	524.37	336.73	137.21	7/7/1994	7/13/1994	
10	252	1994	101	28484.32	6138.49	5818.1	-1.6	7921.46	505.03	440.77	3714.2	295.73	2396.75	2485.11	0	1717.65	547.74	835.02	198.37	419.6	544.82	178.11	7/7/1994	7/13/1994	
11	252	1994	101	24053.57	5173.67	4927.46	0	5547.25	351.83	452.42	2907.85	228.81	1477.55	984.22	0	1078.78	356.5	671.76	210.48	483.62	348.03	160.45	7/7/1994	7/13/1994	
12	252	1994	101	18954.8	3868.54	3682.07	0	4228.76	240.12	224.63	2197.27	144.3	1034.09	2772.18	0	678.77	250.39	392.78	152.23	665.03	244.38	127.73	7/7/1994	7/13/1994	
13	252	1994	101	17612.99	3871.59	3683.73	0	3709.41	312.97	177.46	2194.72	93.09	1126.08	2208.94	0	846.25	222.63	348.9	103.71	574.64	337.13	108.77	7/7/1994	7/13/1994	
14	252	1994	101	16535.69	3859.64	3885.85	0	4188.68	355.51	176.62	1793.45	159.97	1188.17	2116.08	0	906.85	297.84	492.55	57.69	506.55	237.58	122.74	7/7/1994	7/13/1994	
15	255	1994	101	23578.93	6950.73	4993.15	0	6123.47	681.48	276.12	2375.08	149.14	1868.26	2690.82	0	1133.29	525.33	615.83	165.1	526.06	168.31	134.73	7/28/1994	8/3/1994	
16	255	1994	101	25364.67	6765.62	5174.63	0	6624.19	752.33	412.17	2615.73	206.17	1908.44	1681.7	0	1670.86	462.56	577.27	134.26	550.41	343.22	163.17	7/28/1994	8/3/1994	
17	255	1994	101	29841.54	7853.69	6414.29	0.55	7608.84	650.65	475.85	3347.35	218.99	2298.65	2520.36	0	2092.43	668.58	753.45	138.42	383.31	290.97	150.08	7/28/1994	8/3/1994	
18	255	1994	101	23982.13	6130.51	5225.61	-1.6	5264	465.48	323.37	2930.79	177.19	1417.88	1427.51	0	1005.99	278.99	315.65	204.33	392.13	269.02	140.72	7/28/1994	8/3/1994	
19	255	1994	101	20183.1	5549.07	4942.2	0	4096.03	359	290.67	2462.96	117.05	1205.83	3416.45	0	670.68	226.15	316.16	157.86	663.67	270.96	186.63	7/28/1994	8/3/1994	

Figure: Source will be the Sales_Staging table in SSMS

We took values from the Sales_Staging table and mapped it to the Store_Dim table. The below screenshot shows the mapping between the two tables in the staging_area and dw_area databases. In addition to the columns that are present in the Sales_Staging table we have an additional surrogate key column called StoreKey.

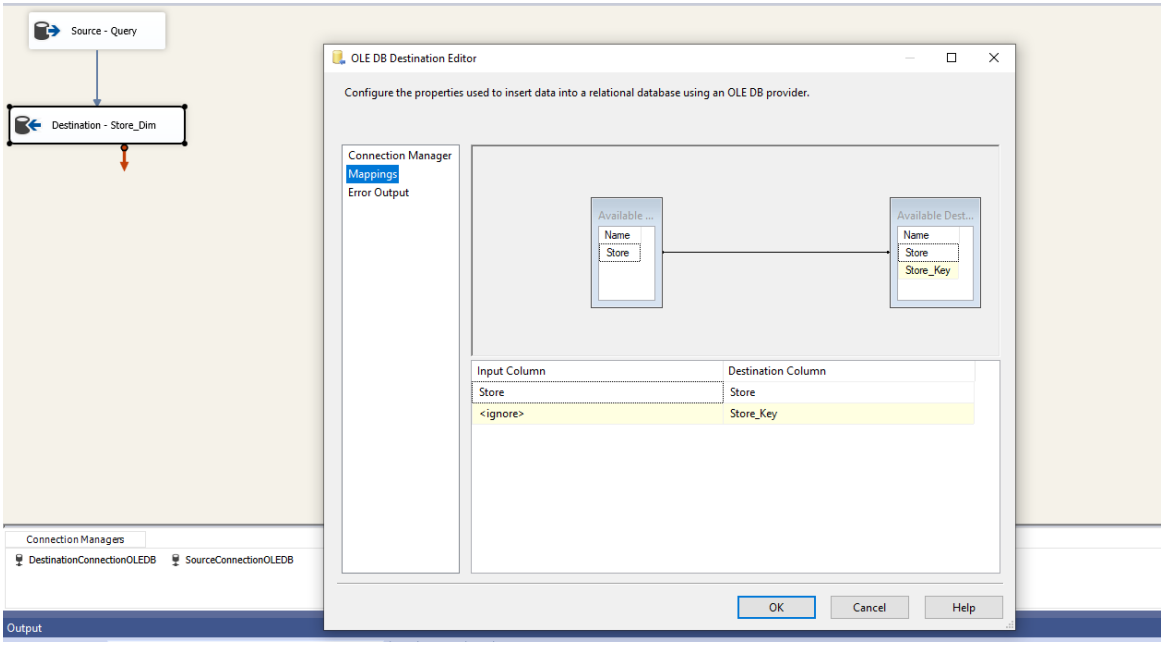


Figure: Mapping of data from source Staging table to Destination Store dimension table

The below table shows the query to create an auto increment surrogate key for Store_Dim table.

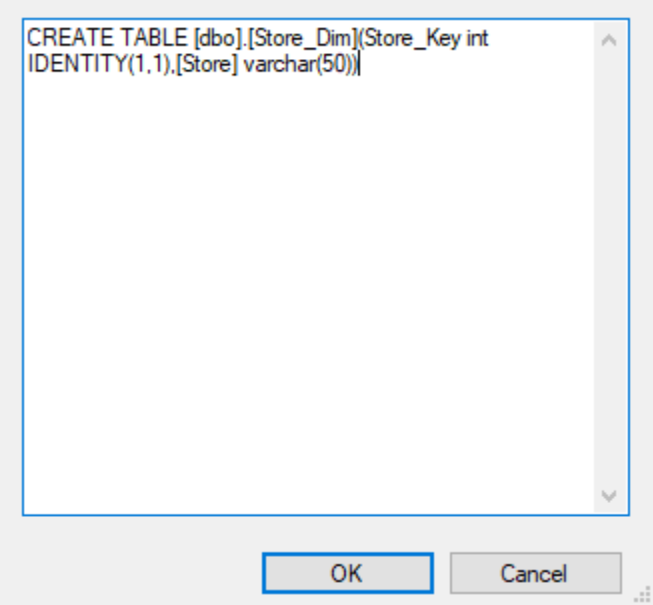


Figure: Create table query on the destination with the surrogate key

The flow is executed to load the table. Almost 125 rows were loaded in the Store_Dim table

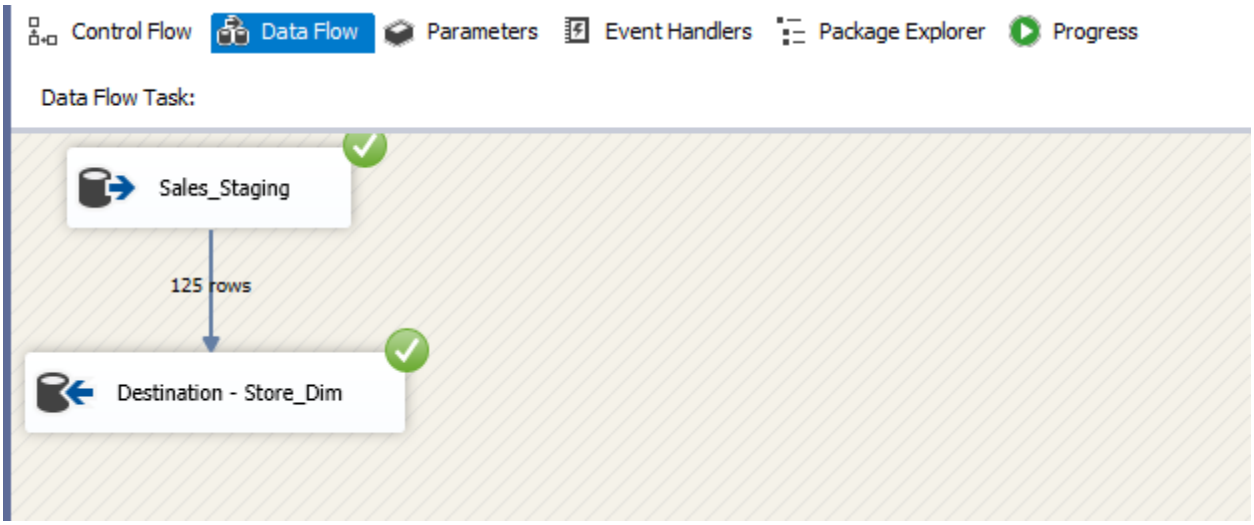
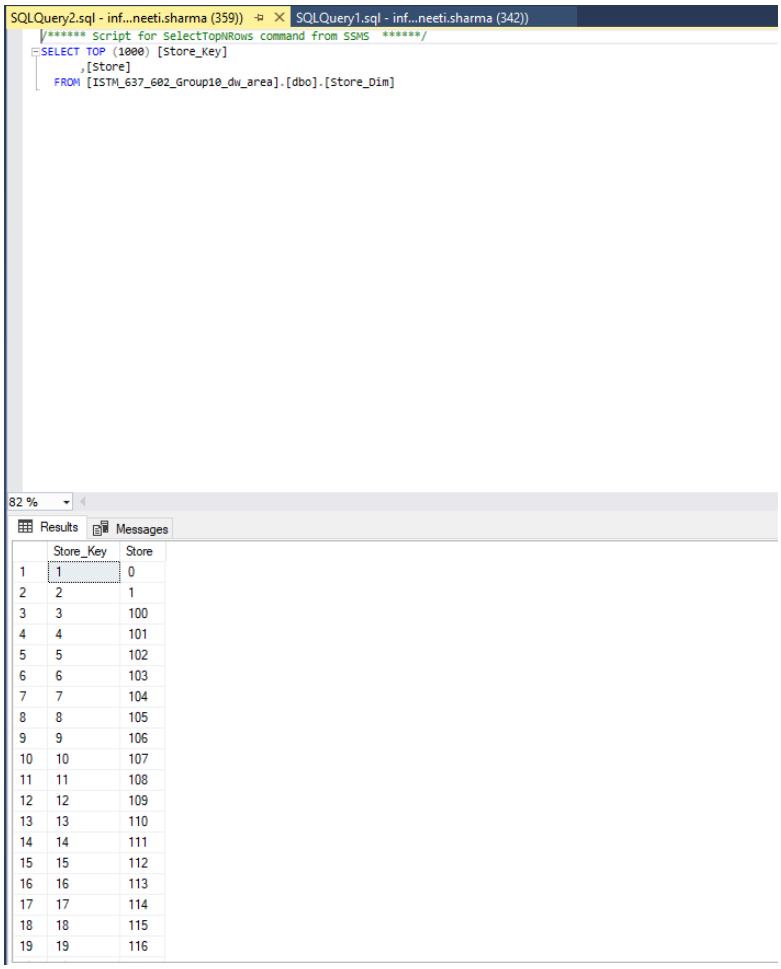


Figure: Loading and Transforming rows to the final dimension table

Data has been successfully loaded in the Store_Dim table in the SSMS.



	Store_Key	Store
1	1	0
2	2	1
3	3	100
4	4	101
5	5	102
6	6	103
7	7	104
8	8	105
9	9	106
10	10	107
11	11	108
12	12	109
13	13	110
14	14	111
15	15	112
16	16	113
17	17	114
18	18	115
19	19	116

Figure: Store_Dim Table successfully loaded in SSMS

Product_Dim table creation

Creating a workflow to take data from the Sales_Staging table and then creating a dimension table called Product_Dim.

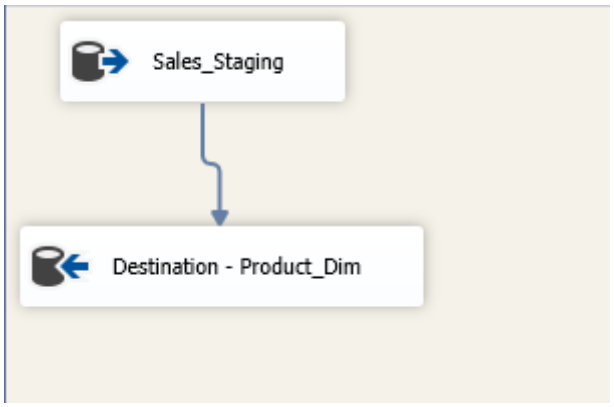


Figure: Extracting data from Sales_Staging table

We took values from the Sales_Staging table and mapped it to the Product_Dim table. The below screenshot shows the mapping between the two tables in the staging_area and dw_area databases. The columns from the Sales_Staging table were manipulated using the unpivot function to transpose the columns from the Sales_Staging table. The query used for the same is:

```
SELECT
  DISTINCT Product_Name
FROM
  ISTM_637_602_Group10_Staging.dbo.Sales_Staging UNPIVOT (
    value for Product_Name IN (
      Grocery, Dairy, Frozen, Bottle, Meat,
      Fish, Floral, Deli, Cheese, Bakery,
      Pharmacy, Jewelry, Beer, Wine, SPIRITS,
      Camera, Saladbar, Cosmetic
    )
  ) AS unpvt;
```

In addition to the columns that are present in the Sales_Staging table we have an additional surrogate key column called ProductKey.

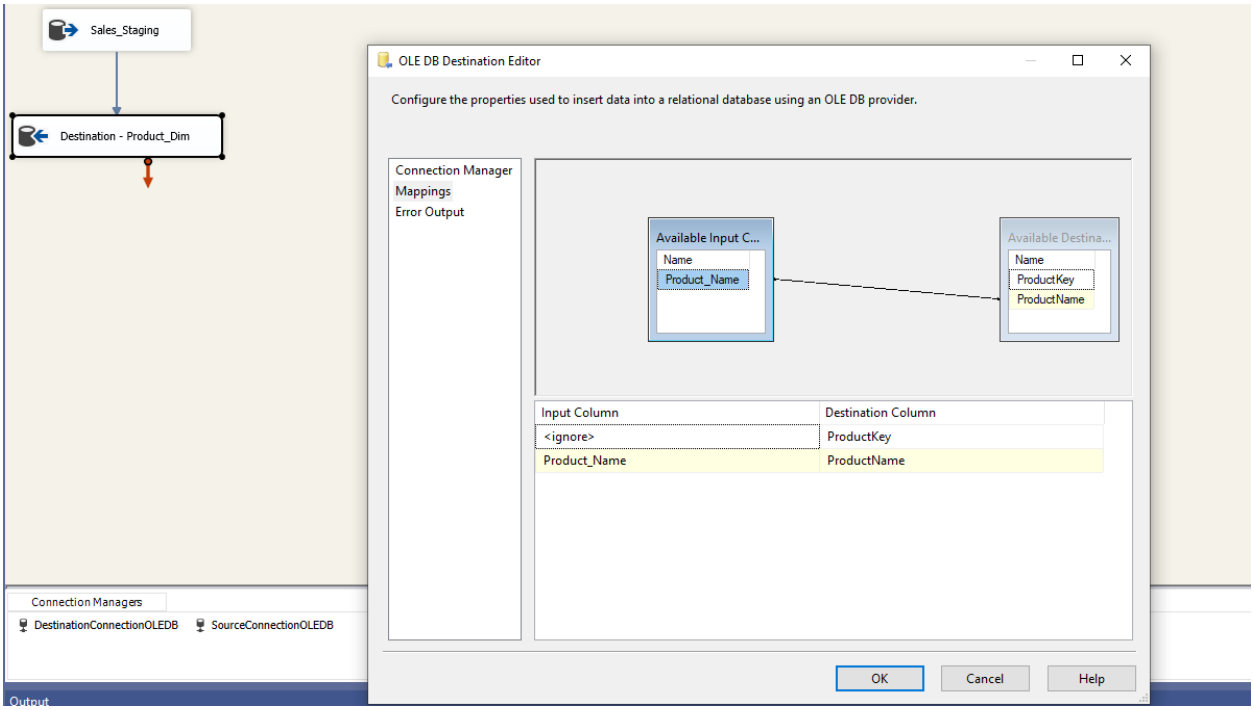


Figure: Mapping of data from source Staging table to Destination Product dimension table

The below table shows the query to create an auto increment surrogate key for the Product_Dim table.

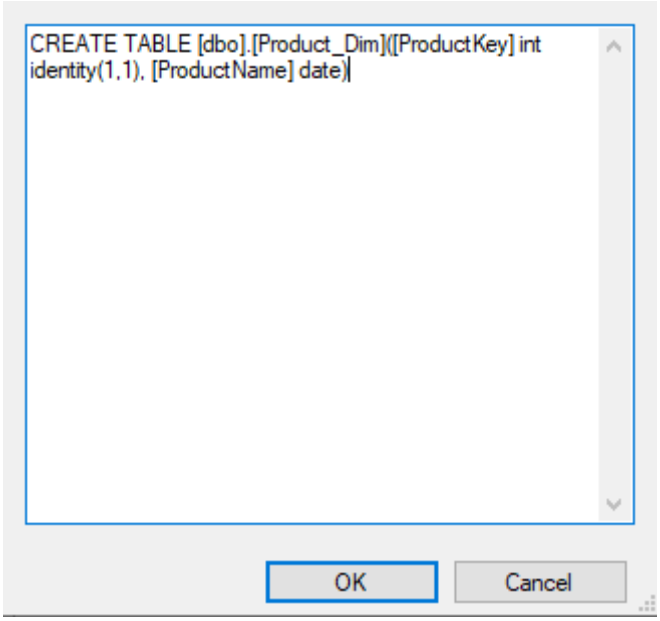


Figure: Create table query on the destination with the surrogate key

The flow is executed to load the table. Almost 18 rows were loaded in the Product_Dim table

Figure: Loading and Transforming rows to the final dimension table

Data has been successfully loaded in the Product_Dim table in the SSMS.

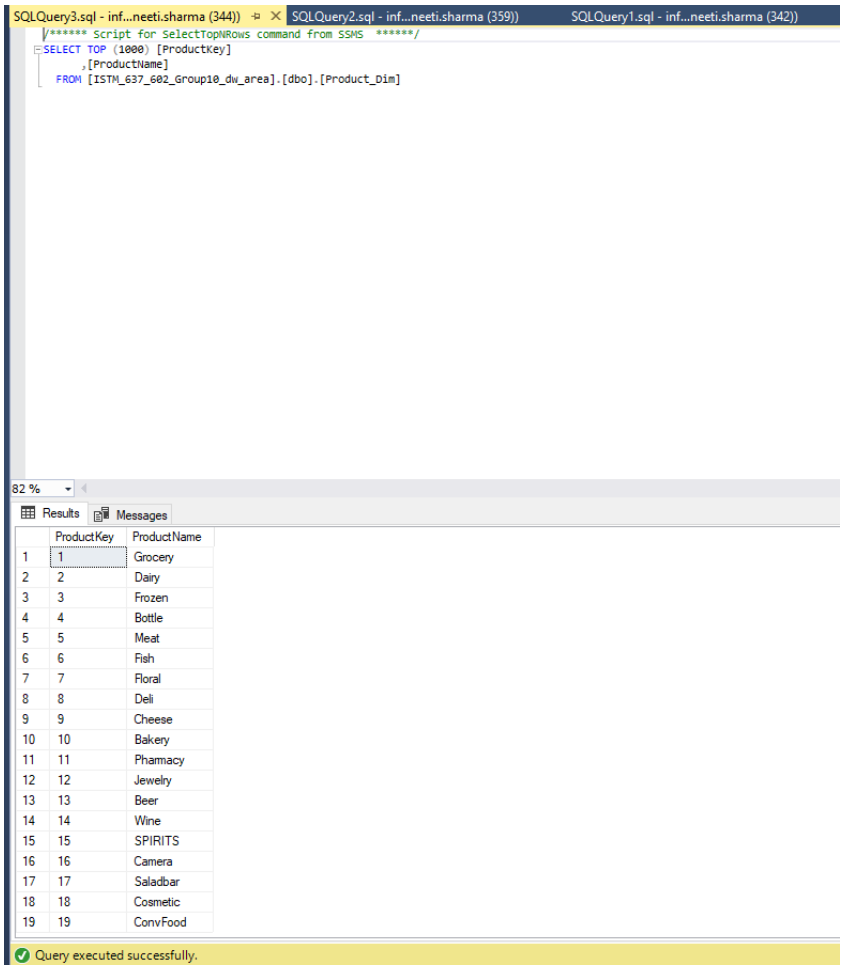


Figure: Product_Dim Table successfully loaded in SSMS

ETL for Fact Tables

Fact_Purchases table creation

Creating a workflow to take data from the Coupon_Staging table and performing some aggregate functions on it to create a Fact table called Fact_Purchases.

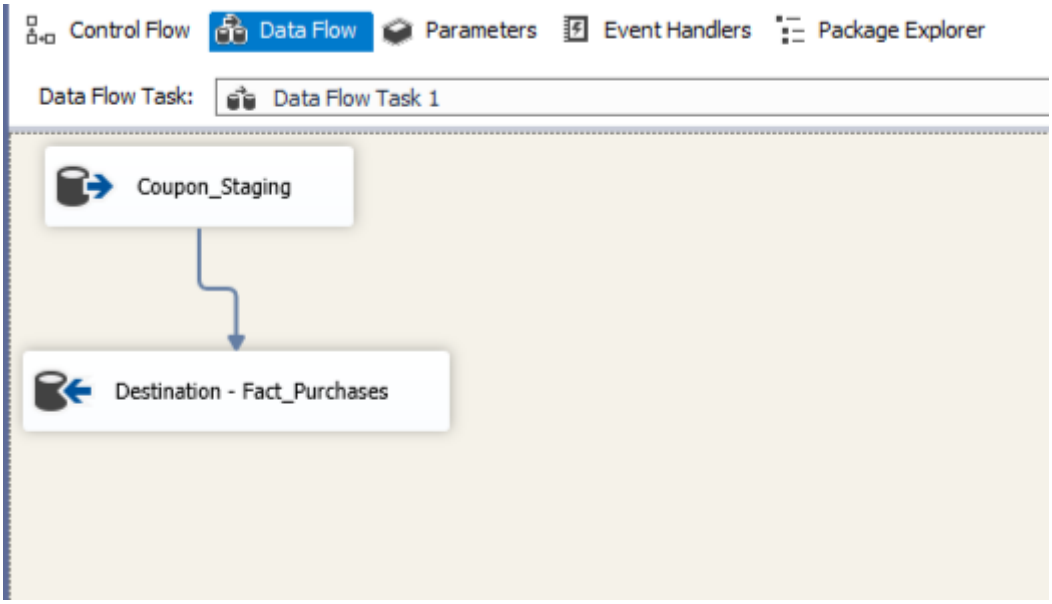


Figure: Extracting data from Staging table to Fact_Purchases table

The below screenshot specifies the data that is present in the Coupon_Staging table in the staging_area database. This table will serve as the primary source for the Coupon_Purchases_Data_Mart.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [UPCID]
      ,[UPCDesc]
      ,[CouponCategory]
      ,[Week]
      ,[Store]
  FROM [ISTM_637_602_Group10_Staging].[dbo].[Coupon_Staging]
```

100 %

Results | Messages

	UPCID	UPCDesc	CouponCategory	Week	Store
1	1060840005	ENDURO GRAPEFRUIT/TA	B	180	104
2	1060840005	ENDURO GRAPEFRUIT/TA		181	104
3	1060840005	ENDURO GRAPEFRUIT/TA	B	182	104
4	1060840005	ENDURO GRAPEFRUIT/TA		183	104
5	1060840005	ENDURO GRAPEFRUIT/TA		184	104
6	1060840005	ENDURO GRAPEFRUIT/TA		185	104
7	1060840005	ENDURO GRAPEFRUIT/TA		186	104
8	1060840005	ENDURO GRAPEFRUIT/TA		187	104
9	1060840005	ENDURO GRAPEFRUIT/TA		188	104
10	1060840005	ENDURO GRAPEFRUIT/TA		189	104
11	1060840005	ENDURO GRAPEFRUIT/TA		190	104
12	1060840005	ENDURO GRAPEFRUIT/TA		191	104
13	1060840005	ENDURO GRAPEFRUIT/TA		192	104
14	1060840005	ENDURO GRAPEFRUIT/TA		193	104

Figure: Source will be the Coupon_Staging table

In order to perform the extraction for the Fact table, I am using the option to select the data from the query. The below screenshot contains a query used as a source of data. This Fact table defines the number of purchases for each coupon category for each upc id (brand of juice) and then counts the values for it. We will also be taking data from the dimension tables of this data mart - Coupon_Dim and UPC_Dim.

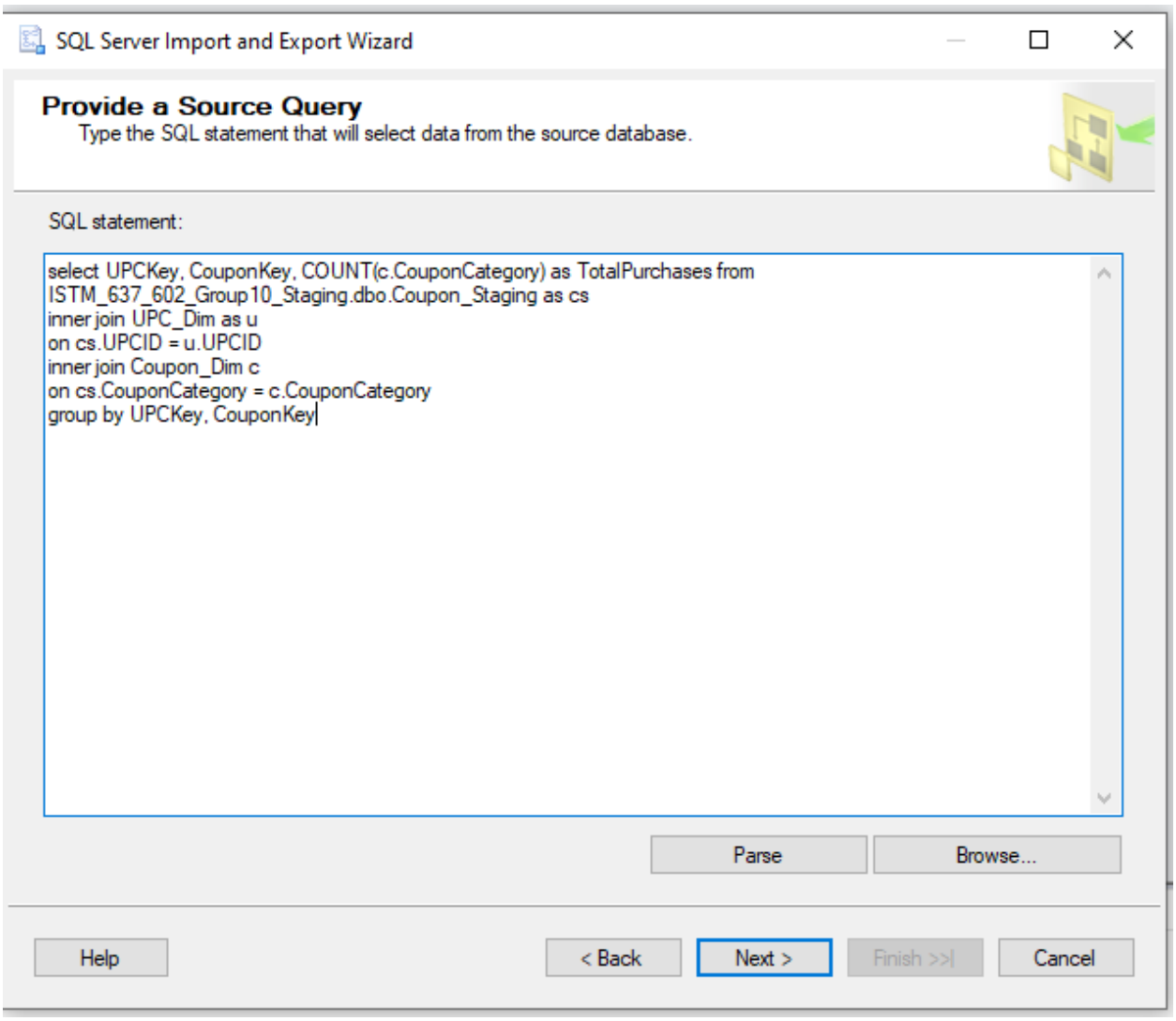


Figure: SQL query used to calculate the aggregate values of the TotalPurchases

We took values from the Coupon_Staging table and mapped it to the Fact_Purchases table. It will contain the surrogate keys from the dimension tables (UPC_Dim and Coupon_Dim) of the data mart - Coupon_Purchases_Data_Mart. Additionally, the fact table contains the metric TotalPurchases, storing the count of each type of coupon categories.

The below screenshot shows the mapping between the two tables in the staging_area and dw_area databases. In addition to the surrogate keys (UPCKey and CouponKey) column that are present in the dimension tables, we are defining a new attribute called TotalPurchases that will contain the total count of each category for a present in the Coupon_Staging table we have an additional surrogate key column called CouponKey.

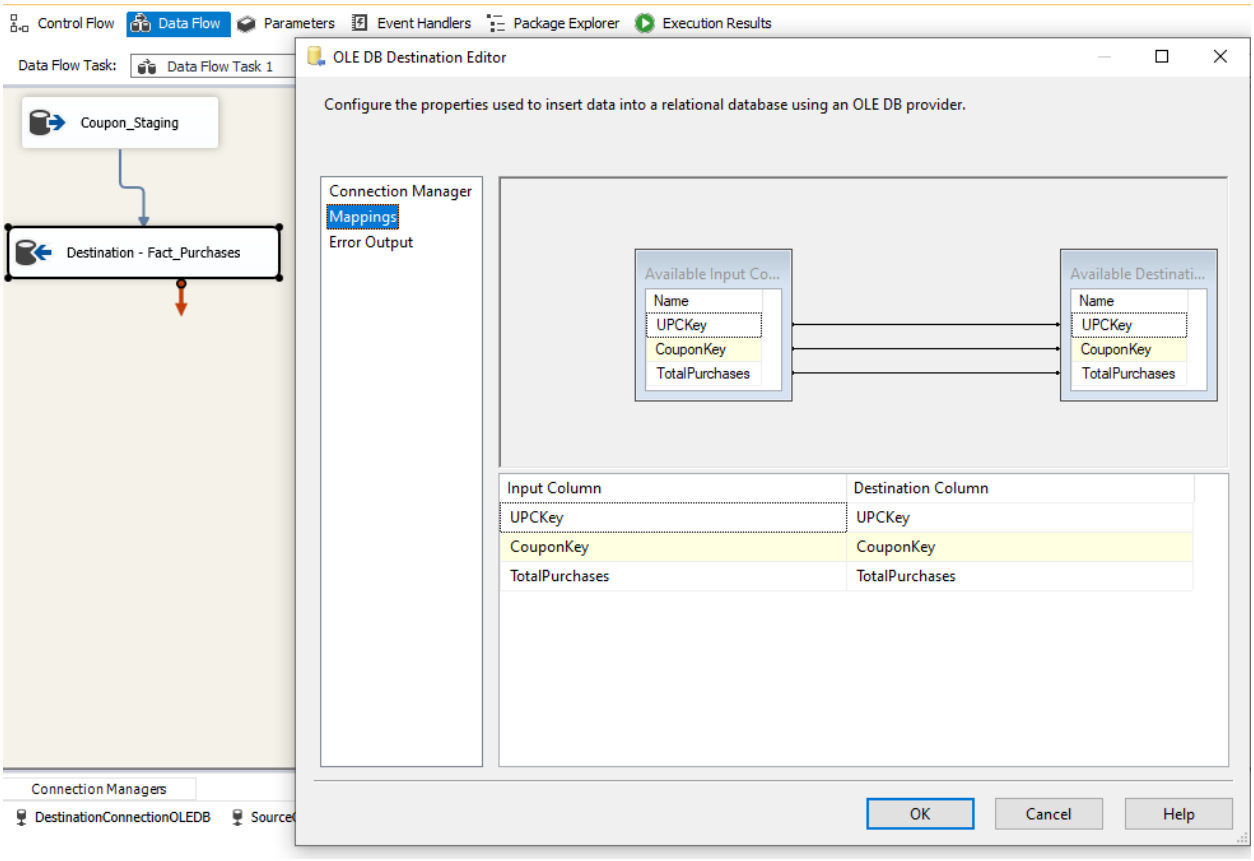


Figure: Mapping of data from source Staging table to Destination Fact_Purchases table

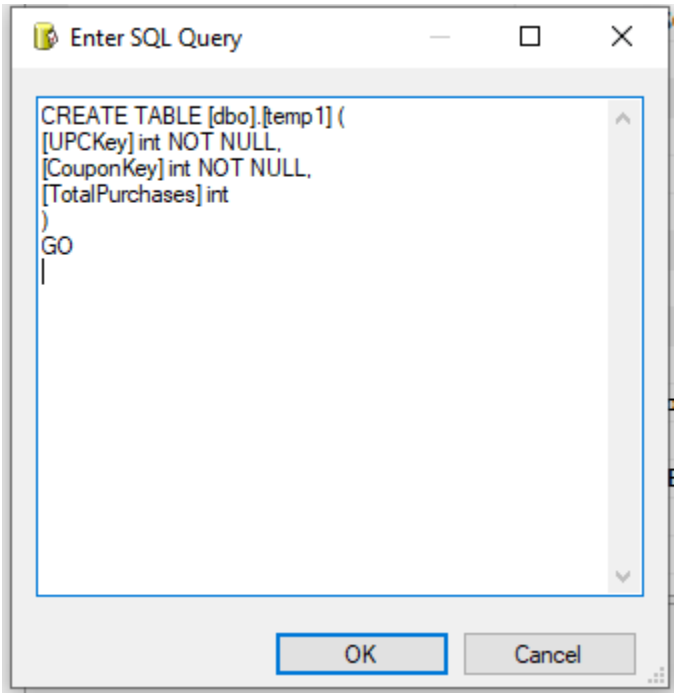


Figure: Create table query for the destination fact table

Run the package to execute the task and create a table in the ISTM_637_602_Group10_dw_area from the data being extracted and transformed using the dimension tables and Staging tables.

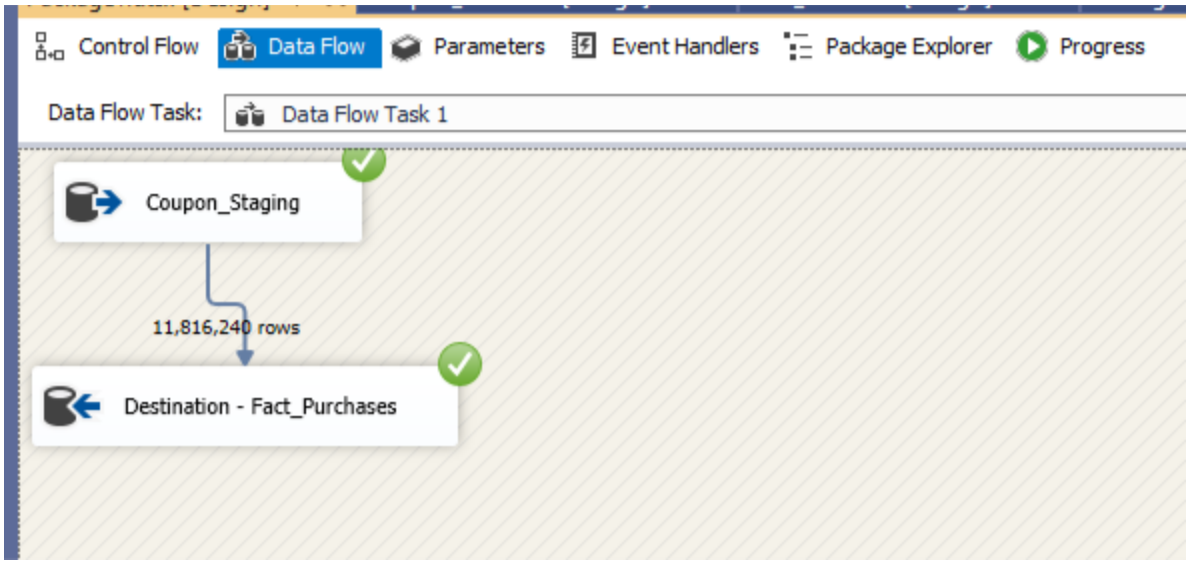
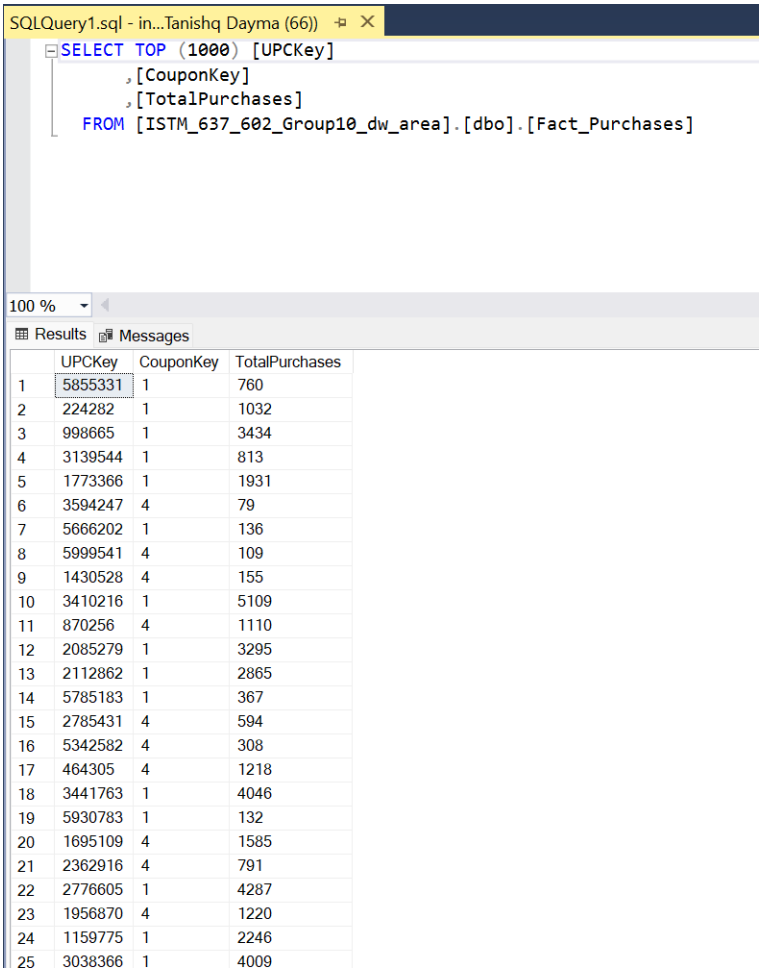


Figure: Loading and Transforming rows to the final fact table

Data has been successfully loaded in the Fact_Purhcases table in the SSMS.



	UPCKey	CouponKey	TotalPurchases
1	5855331	1	760
2	224282	1	1032
3	998665	1	3434
4	3139544	1	813
5	1773366	1	1931
6	3594247	4	79
7	5666202	1	136
8	5999541	4	109
9	1430528	4	155
10	3410216	1	5109
11	870256	4	1110
12	2085279	1	3295
13	2112862	1	2865
14	5785183	1	367
15	2785431	4	594
16	5342582	4	308
17	464305	4	1218
18	3441763	1	4046
19	5930783	1	132
20	1695109	4	1585
21	2362916	4	791
22	2776605	1	4287
23	1956870	4	1220
24	1159775	1	2246
25	3038366	1	4009

Figure: Fact_Purchases table successfully loaded in the database

Fact_Demo table creation

For the Fact_Demo table we used the SSIS package named Fact_Demo.

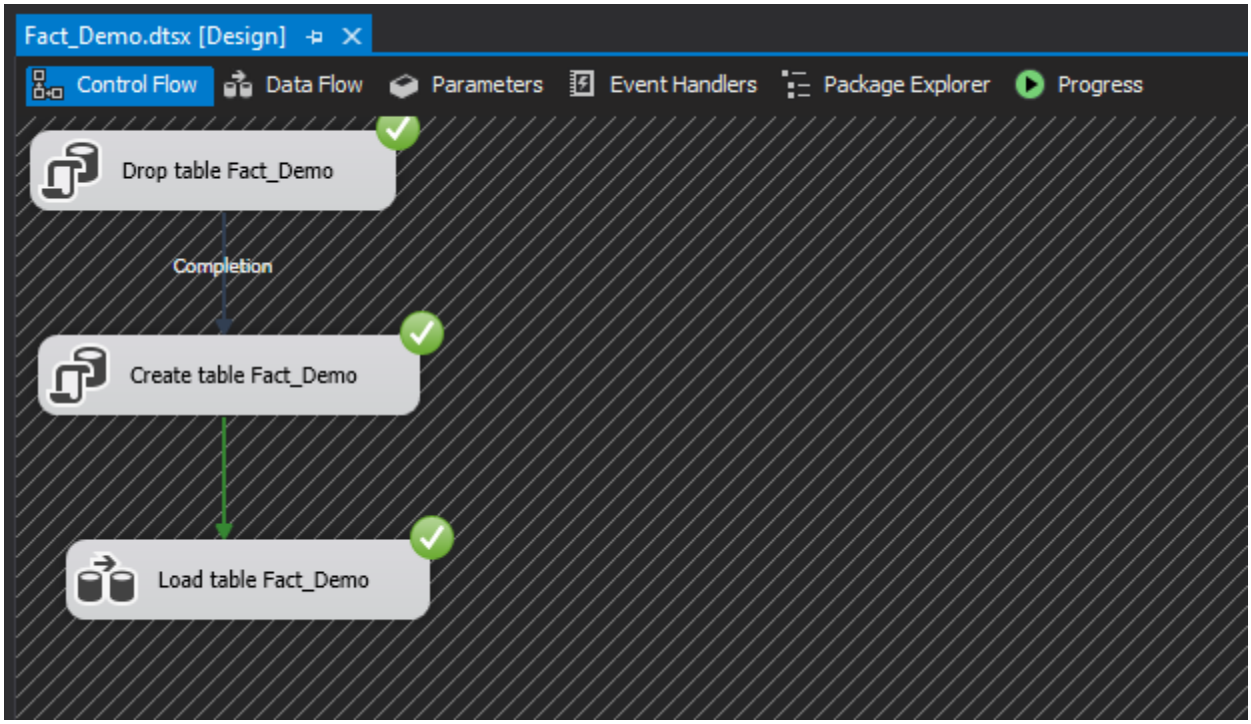


Figure: Overview of Fact_Demo package

As shown above, at the beginning we drop the Fact_Demo table to perform a full refresh of the data, then, we create the table and finally we load it.

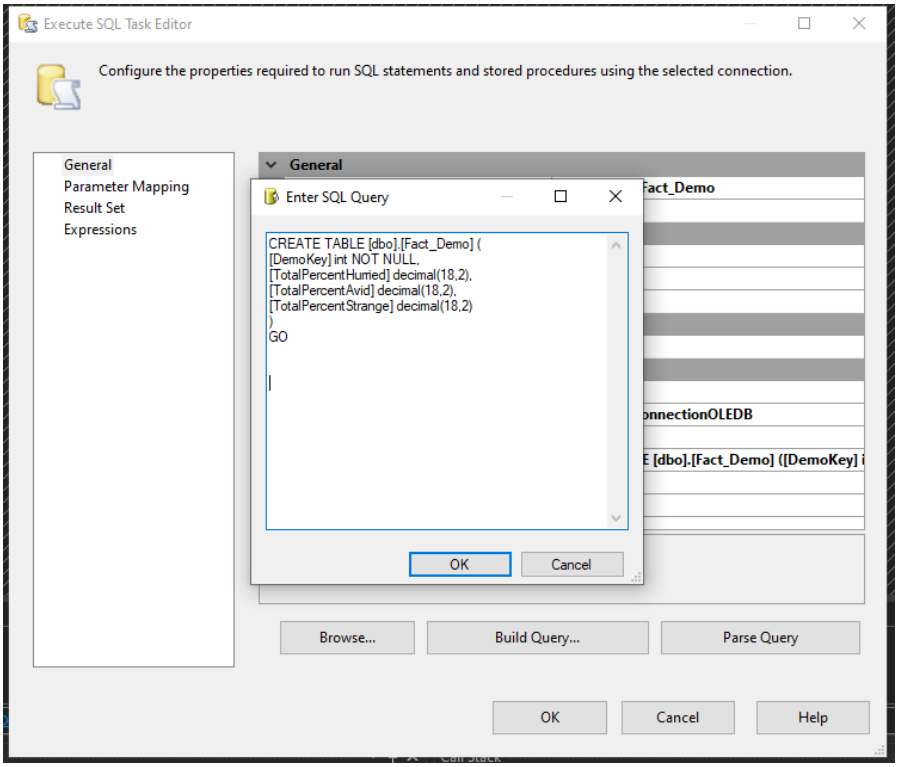


Figure: Create table query for the destination fact table

The screenshot shows the CREATE TABLE query that is performed before loading the data. We use a custom query to extract the right data from the source table. As shown in the screenshot below

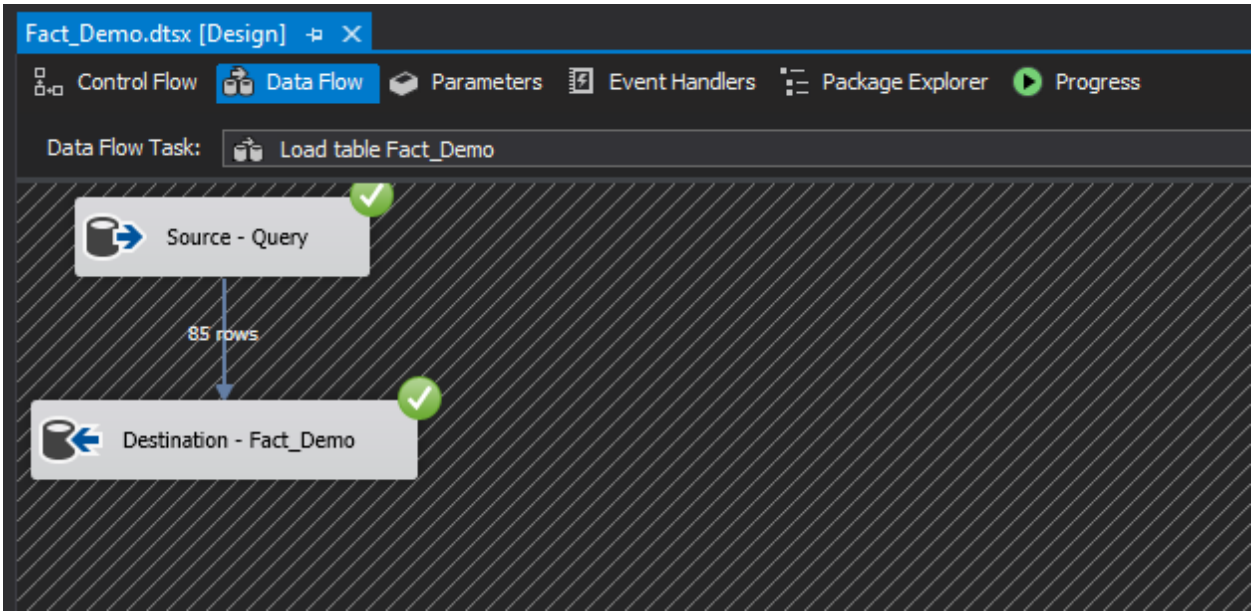


Figure: Overview of Data Flow for Fact_Demo package

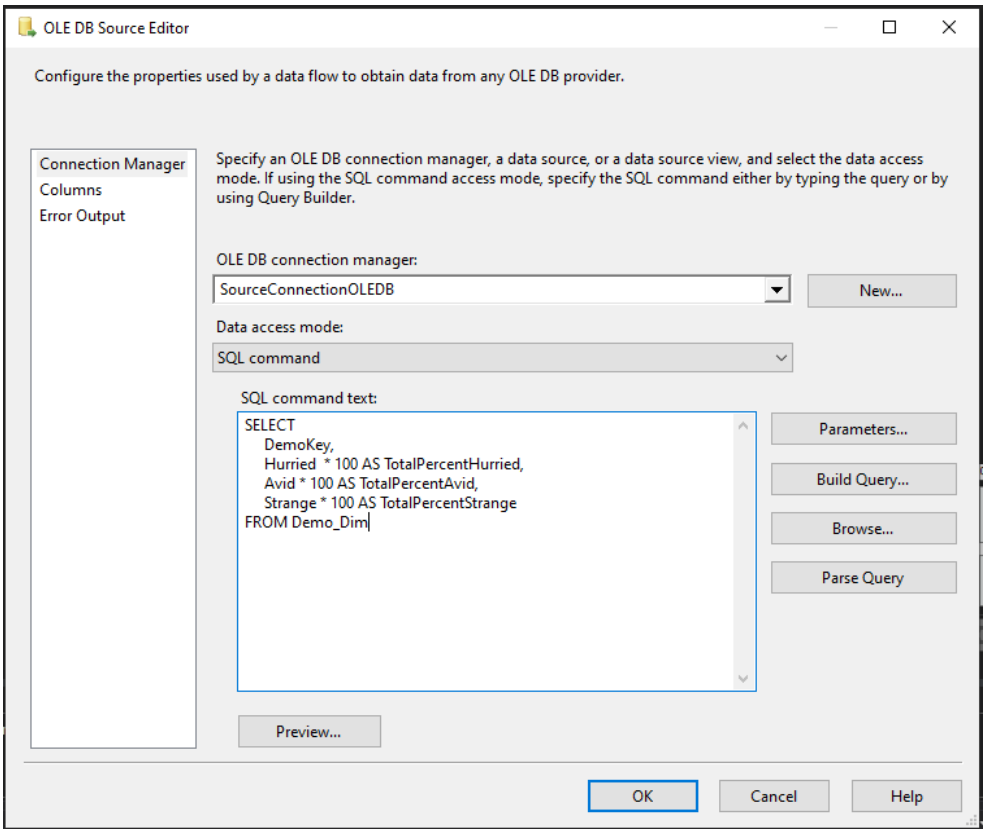


Figure: Select query for source data to Fact_Demo

In the query shown above, we are selecting the information from Demo_Dim and transforming it to the correct format we expect in the Fact_Demo table. Since we use a custom query, we already prepare the names of the final mapping to make it easier to load into the final table.

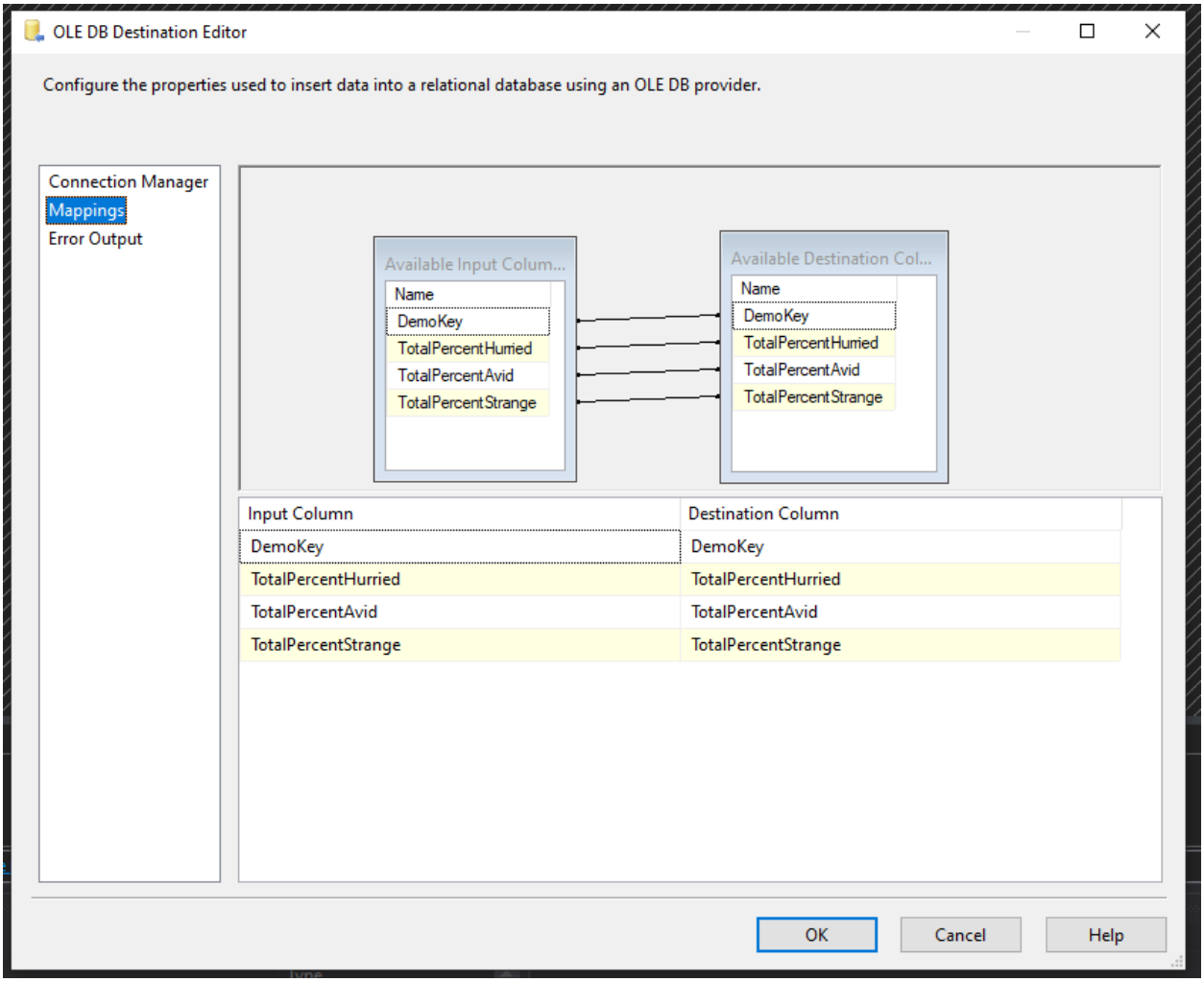


Figure: Mapping of data from source Staging table to Destination Fact_Demo table

Finally, we can show the data successfully loaded to the Fact_Demo table

SQLQuery5.sql - in...nishq Dayma (210))

```
SELECT TOP (1000) [DemoKey]
      ,[TotalPercentHurried]
      ,[TotalPercentAvid]
      ,[TotalPercentStrange]
  FROM [ISTM_637_602_Group10_dw_area].[dbo].[Fact_Demo]
```

100 %

Results | Messages

	DemoKey	TotalPercentHurried	TotalPercentAvid	TotalPercentStrange
1	1	12.10	18.12	23.22
2	2	11.83	16.28	31.69
3	3	19.06	16.80	33.01
4	4	13.51	21.74	21.61
5	5	15.86	15.36	23.64
6	6	4.59	11.04	36.30
7	7	21.54	13.86	21.64
8	8	10.23	21.56	23.10
9	9	22.69	25.02	29.58
10	10	16.58	16.30	26.59
11	11	11.47	15.84	29.06
12	12	2.63	6.13	55.77
13	13	13.03	23.80	27.75
14	14	20.46	17.70	20.52
15	15	15.74	17.56	39.22
16	16	16.86	21.96	31.26
17	17	12.93	13.84	45.18
18	18	17.19	18.96	30.44
19	19	12.64	22.58	32.14
20	20	14.29	23.34	26.61
21	21	21.07	14.18	31.34
22	22	14.29	16.10	19.51
23	23	19.14	19.11	40.36
24	24	16.63	17.56	27.68
25	25	23.75	23.25	24.96
26	26	24.04	10.36	18.37
27	27	17.16	22.62	29.59
28	28	22.83	26.03	30.84
29	29	16.08	15.41	31.88

Query executed successfully.

Figure: Fact_Demo table successfully loaded in the database

Fact_Sales table creation

Creating a workflow to take data from the Sales_Staging table and performing some aggregate functions on it to create a Fact table called Fact_Sales.

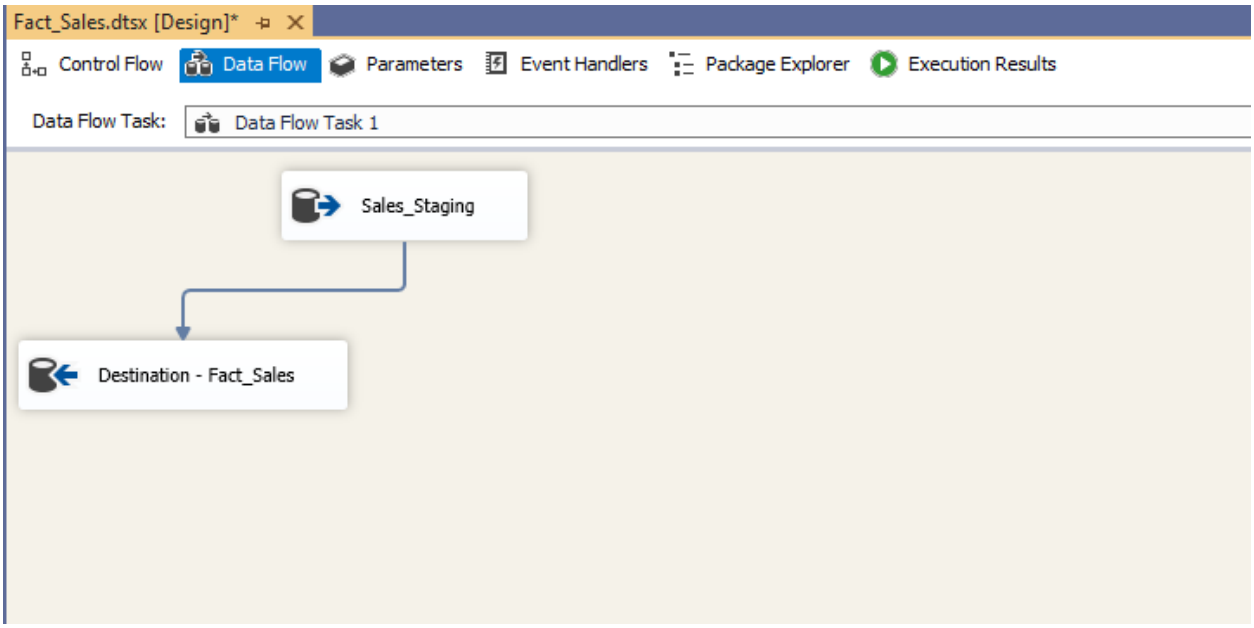
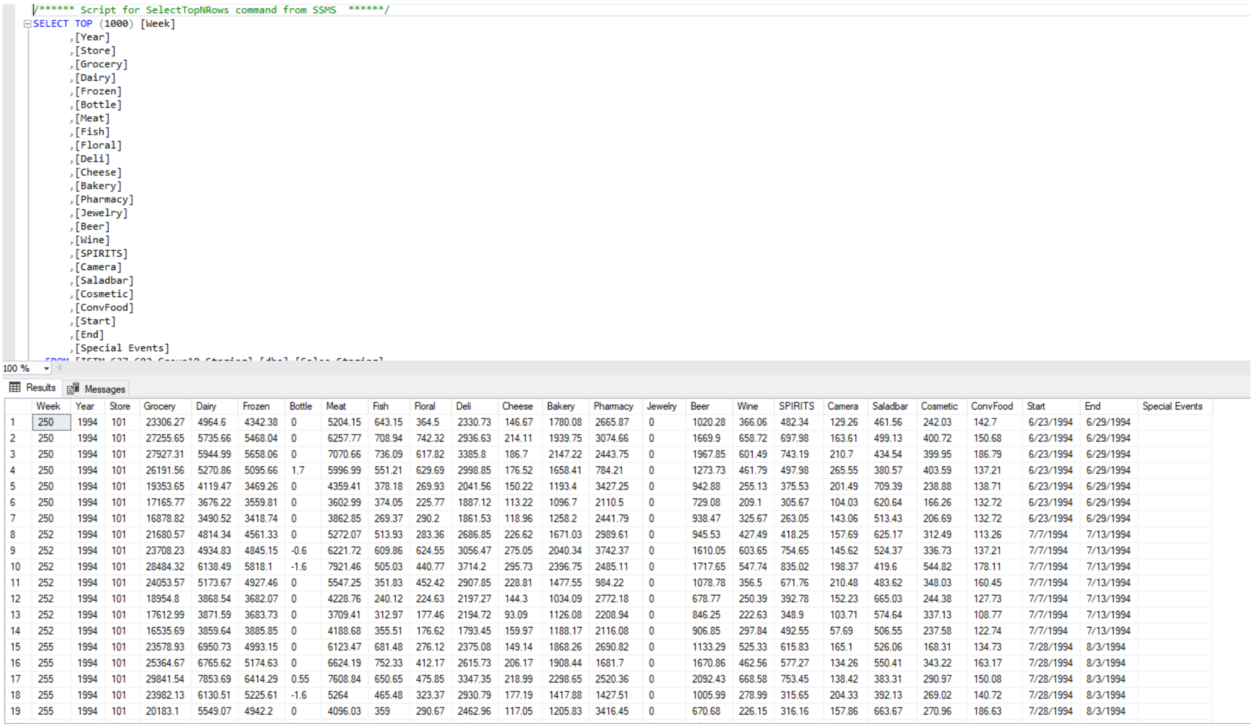


Figure: Extracting data from Staging table to Fact_Sales table

The below screenshot specifies the data that is present in the Sales_Staging table in the staging_area database.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Week]
      ,[Year]
      ,[Store]
      ,[Grocery]
      ,[Dairy]
      ,[Frozen]
      ,[Bottle]
      ,[Meat]
      ,[Fish]
      ,[Floral]
      ,[Deli]
      ,[Cheese]
      ,[Bakery]
      ,[Pharmacy]
      ,[Jewelry]
      ,[Beer]
      ,[Wine]
      ,[SPIRITS]
      ,[Camera]
      ,[Saladbar]
      ,[Cosmetic]
      ,[ConvFood]
      ,[Start]
      ,[End]
      ,[Special Events]
```

	Week	Year	Store	Grocery	Dairy	Frozen	Bottle	Meat	Fish	Floral	Deli	Cheese	Bakery	Pharmacy
1	250	1994	101	23306.27	4964.6	4342.38	0	5204.15	643.15	364.5	2330.73	146.67	1780.08	2665.87
2	250	1994	101	27255.65	5735.66	5468.04	0	6257.77	708.94	742.32	2936.63	214.11	1939.75	3074.66
3	250	1994	101	27927.31	5944.99	5658.06	0	7070.66	736.09	617.82	3385.8	186.7	2147.22	2443.75
4	250	1994	101	26191.56	5270.86	5095.66	1.7	5996.99	551.21	629.69	2998.85	176.52	1658.41	784.21
5	250	1994	101	19353.65	4119.47	3469.26	0	4359.41	378.18	269.93	2041.56	150.22	1193.4	3427.25
6	250	1994	101	17165.77	3676.22	3559.81	0	3602.99	374.05	225.77	1887.12	113.22	1096.7	2110.5
7	250	1994	101	16878.82	3490.52	3418.74	0	3862.85	269.37	290.2	1861.53	118.96	1258.2	2441.79
8	252	1994	101	21680.57	4814.34	4561.33	0	5272.07	513.93	283.36	2686.85	226.62	1671.03	2989.61
9	252	1994	101	23708.23	4934.83	4845.15	-0.6	6221.72	609.86	624.55	3056.47	275.05	2040.34	3742.37
10	252	1994	101	28484.32	6138.49	5818.1	-1.6	7921.46	505.03	440.77	3714.2	295.73	2396.75	2485.11
11	252	1994	101	24053.57	5173.67	4927.46	0	5547.25	351.83	452.42	2907.85	228.81	1477.55	984.22
12	252	1994	101	18954.8	3868.54	3682.07	0	4228.76	240.12	224.63	2197.27	144.3	1034.09	2772.18
13	252	1994	101	17612.99	3871.59	3683.73	0	3709.41	312.97	177.46	2194.72	93.09	1126.08	2208.94
14	252	1994	101	16535.69	3859.64	3885.85	0	4188.68	355.51	176.62	1793.45	159.97	1188.17	2116.08
15	255	1994	101	23578.93	6950.73	4993.15	0	6123.47	681.48	276.12	2375.08	149.14	1868.26	2690.82
16	255	1994	101	25364.67	6765.62	5174.63	0	6624.19	752.33	412.17	2615.73	206.17	1908.44	1681.7
17	255	1994	101	29841.54	7853.69	6414.29	0.55	7608.84	650.65	475.85	3347.35	218.99	2298.65	2520.36
18	255	1994	101	23982.13	6130.51	5225.61	-1.6	5264	465.48	323.37	2930.79	177.19	1417.88	1427.51
19	255	1994	101	20183.1	5549.07	4942.2	0	4096.03	359	290.67	2462.96	117.05	1205.83	3416.45

	Jewelry	Beer	Wine	SPIRITS	Camera	Saladbar	Cosmetic	ConvFood	Start	End	Special Events
1	0	1020.28	366.06	482.34	129.26	461.56	242.03	142.7	6/23/1994	6/29/1994	
2	0	1669.9	658.72	697.98	163.61	499.13	400.72	150.68	6/23/1994	6/29/1994	
3	0	1967.85	601.49	743.19	210.7	434.54	399.95	186.79	6/23/1994	6/29/1994	
4	0	1273.73	461.79	497.98	265.55	380.57	403.59	137.21	6/23/1994	6/29/1994	
5	0	942.88	255.13	375.53	201.49	709.39	238.88	138.71	6/23/1994	6/29/1994	
6	0	729.08	209.1	305.67	104.03	620.64	166.26	132.72	6/23/1994	6/29/1994	
7	0	938.47	325.67	263.05	143.06	513.43	206.69	132.72	6/23/1994	6/29/1994	
8	0	945.53	427.49	418.25	157.69	625.17	312.49	113.26	7/7/1994	7/13/1994	
9	0	1610.05	603.65	754.65	145.62	524.37	336.73	137.21	7/7/1994	7/13/1994	
10	0	1717.65	547.74	835.02	198.37	419.6	544.82	178.11	7/7/1994	7/13/1994	
11	0	1078.78	356.5	671.76	210.48	483.62	348.03	160.45	7/7/1994	7/13/1994	
12	0	678.77	250.39	392.78	152.23	665.03	244.38	127.73	7/7/1994	7/13/1994	
13	0	846.25	222.63	348.9	103.71	574.64	337.13	108.77	7/7/1994	7/13/1994	
14	0	906.85	297.84	492.55	57.69	506.55	237.58	122.74	7/7/1994	7/13/1994	
15	0	1133.29	525.33	615.83	165.1	526.06	168.31	134.73	7/28/1994	8/3/1994	
16	0	1670.86	462.56	577.27	134.26	550.41	343.22	163.17	7/28/1994	8/3/1994	
17	0	2092.43	668.58	753.45	138.42	383.31	290.97	150.08	7/28/1994	8/3/1994	
18	0	1005.99	278.99	315.65	204.33	392.13	269.02	140.72	7/28/1994	8/3/1994	
19	0	670.68	226.15	316.16	157.86	663.67	270.96	186.63	7/28/1994	8/3/1994	

Figure: Source will be the Sales_Staging table

To perform the extraction for the Fact table, the option to select the data from the query is used. The below screenshot contains a query used as a source of data. This Fact table defines the total sales for each product category for every year in different stores. We will also be taking data from the dimension tables of this data mart – Date_Dim, Store_Dim and Product_Dim

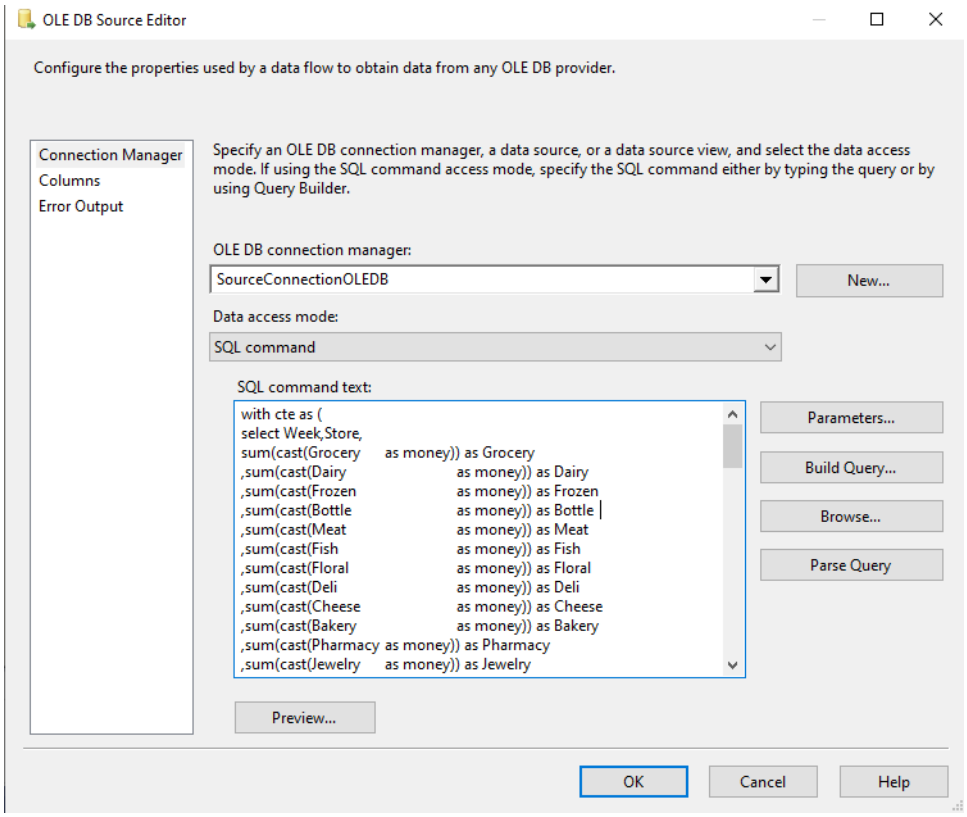


Figure: SQL query used to calculate the aggregate values of the TotalSales

We took values from the Sales_Staging table and mapped it to the Fact_Sales table. It will contain the surrogate keys from the dimension tables (Date_Dim, Store_Dim and Product_Dim) of the data mart - Sales_Data_Mart. Additionally, the fact table contains the metric TotalSales calculating sales for each product category.

The below screenshot shows the mapping between the two tables in the staging_area and dw_area databases. In addition to the surrogate keys (DateKey, StoreKey and CouponKey) columns that are present in the dimension tables, we are defining a new attribute called TotalSales that will contain the total sales of each product category present in the Sales_Staging table.

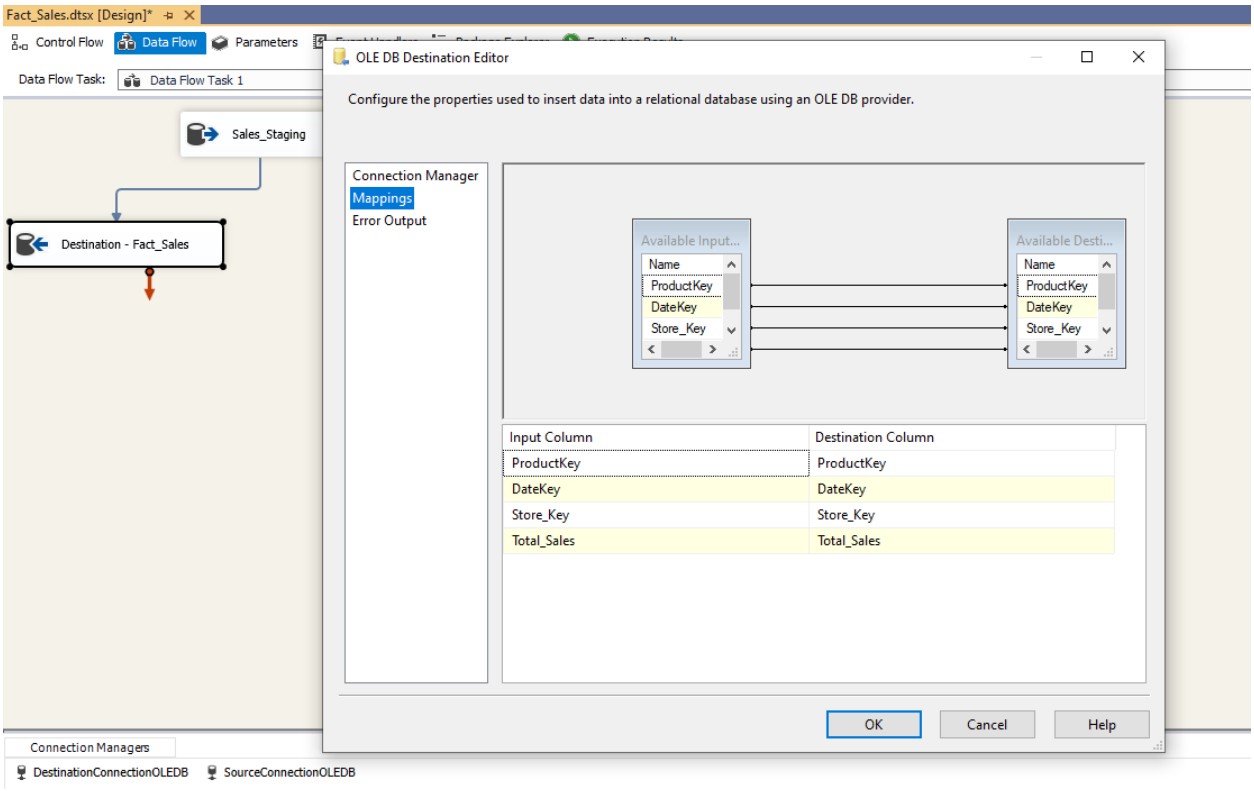


Figure: Mapping of data from source Staging table to Destination Fact_Sales table

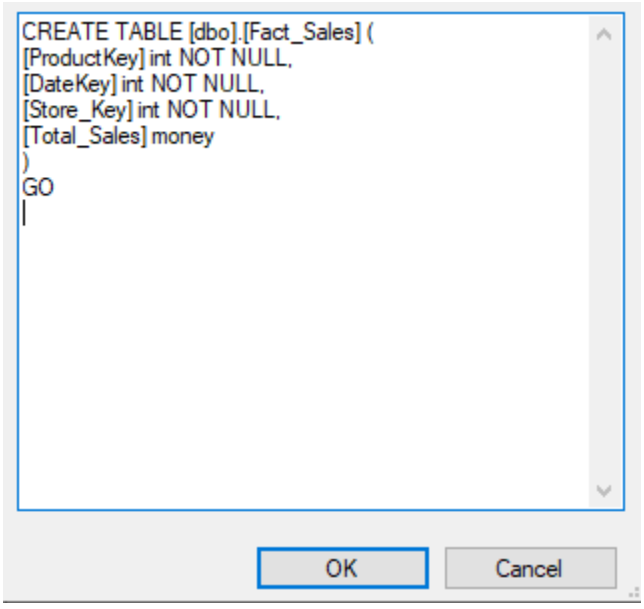


Figure: Create table query for the destination fact table

Run the package to execute the task and create a table in the ISTM_637_602_Group10_dw_area from the data being extracted and transformed using the dimension tables and Staging tables.

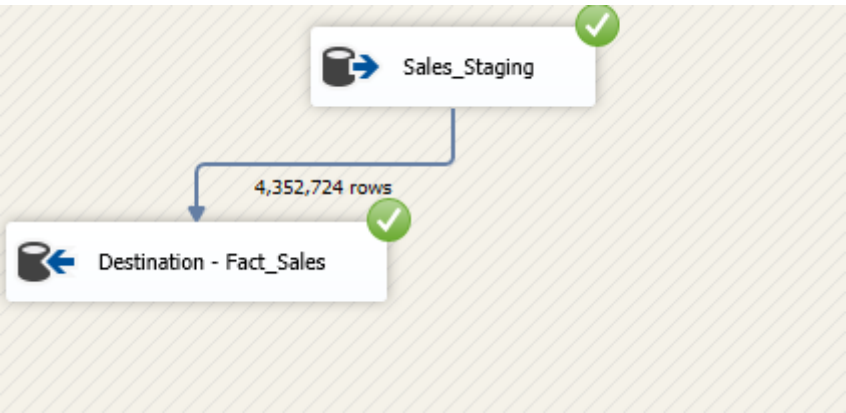
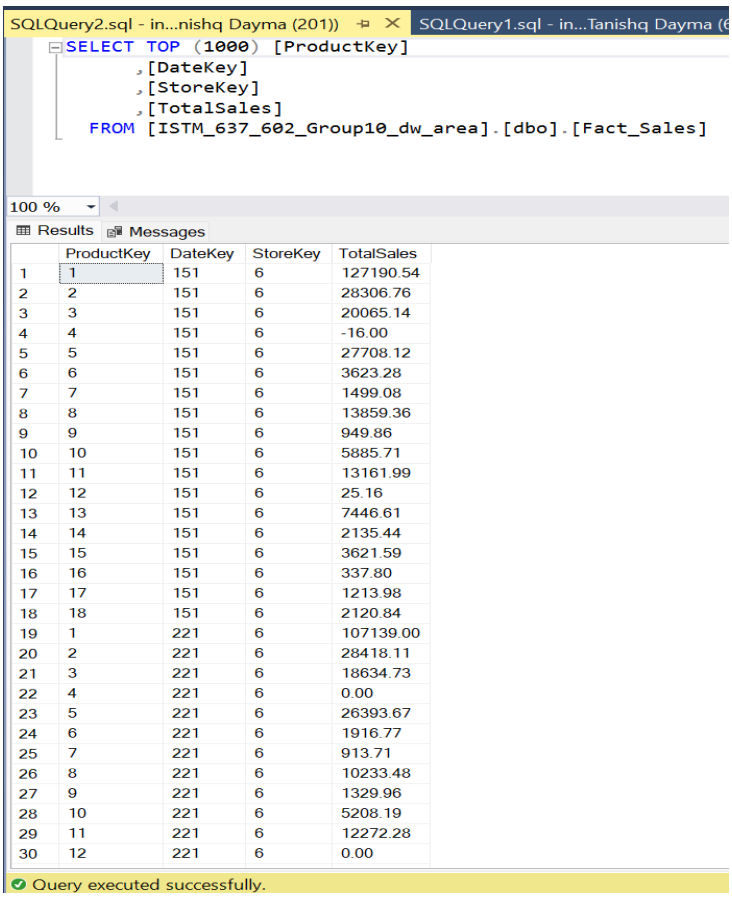


Figure: Loading and Transforming rows to the final fact table

Data has been successfully loaded in the Fact_Sales table in the SSMS.



SQLQuery2.sql - in...nishq Dayma (201)) | SQLQuery1.sql - in...Tanishq Dayma (

```
SELECT TOP (1000) [ProductKey]
      ,[DateKey]
      ,[StoreKey]
      ,[TotalSales]
  FROM [ISTM_637_602_Group10_dw_area].[dbo].[Fact_Sales]
```

100 %

Results | Messages

	ProductKey	DateKey	StoreKey	TotalSales
1	1	151	6	127190.54
2	2	151	6	28306.76
3	3	151	6	20065.14
4	4	151	6	-16.00
5	5	151	6	27708.12
6	6	151	6	3623.28
7	7	151	6	1499.08
8	8	151	6	13859.36
9	9	151	6	949.86
10	10	151	6	5885.71
11	11	151	6	13161.99
12	12	151	6	25.16
13	13	151	6	7446.61
14	14	151	6	2135.44
15	15	151	6	3621.59
16	16	151	6	337.80
17	17	151	6	1213.98
18	18	151	6	2120.84
19	1	221	6	107139.00
20	2	221	6	28418.11
21	3	221	6	18634.73
22	4	221	6	0.00
23	5	221	6	26393.67
24	6	221	6	1916.77
25	7	221	6	913.71
26	8	221	6	10233.48
27	9	221	6	1329.96
28	10	221	6	5208.19
29	11	221	6	12272.28
30	12	221	6	0.00

Query executed successfully.

Figure: Fact_Sales table successfully loaded in the database

Data Granularity in the Independent Data Marts

In the context of independent data marts, each mart focuses on specific business areas and exhibits unique granularity levels.

Sales_Data_Mart
Granularity Level: Finest granularity at the Product-Store-Date level in the Fact_Sales table. It allows detailed analysis of sales, providing insights into the performance of each product in each store on specific dates.

Coupon_Purchases_Data_Mart
Granularity Level: Finest granularity at the UPC-Coupon level in the Fact_Purchases table. It focuses on the impact of coupons on purchases, offering detailed information on the quantity of items bought for each unique product and coupon combination.

Demo_Data_Mart
Granularity Level: Finest granularity at the Demo level in the Fact_Demo table. It provides a comprehensive view of demographic data, offering percentages of avid, hurried, and strange shoppers for each demographic.

Each data mart is designed to meet specific business requirements, enabling targeted analysis and reporting. The granularity levels are tailored to capture the most detailed information relevant to the business questions each data mart aims to address.

Remove all Temporary Tables from Staging Area

The following temp tables were removed from the staging area after loading all the tables in the independent data marts.

Temp Tables	
Ccount	Consists of data from customer count CSV file
Week_Decode_Clean	Consists of data from the WeekDecode CSV file (derived from the Dominicks.pdf file).
UPCBJC	Consists of data from the UPC table for the juices

DONE-WBJC	Consists of data from the Movement table for the juices.

Section 6. Business Intelligence (BI) Reporting

Reporting Plan

Target Reports

BQ 1 Target Report
Which are the top 5 categories for products across different stores between the years 1980 to 1997?

The reporting tool we used to answer this business question was Power BI. We found that Power BI is a powerful tool that allows us to visualize and share insights to answer this question. To begin, Power BI has a user-friendly interface, it is intuitive making it accessible to users without in-depth technical knowledge to generate reports and visualizations. It has many drag-and-drop features. Since Power BI and SQL Server are provided by Microsoft, they have a seamless data integration making it easy to connect to the database and add supporting data from cloud-based sources, Excel spreadsheets, and more. Finally, using Power BI we can build interactive visualization such as charts, graphs, and maps.

BQ 2 Target Report
What are the year-over-year trends and patterns in alcohol (Beer, Wine, Spirit) sales from 1987 to 1997?

We used cubes from SSAS and reporting using SSRS to answer this business question. We found that using cubes from SSAS in combination with SSRS we had several advantages for answering this question. To answer this question we had to analyze multiple dimensions, SSAS allows the creation of multidimensional data models which are improved for analytical queries. Cubes can clearly show complex business hierarchies and relationships. Furthermore, using cubes from SSAS we can create dimensional hierarchies, this allows us to drill down into data at different levels of granularity. We can have a report as detailed as we need it. Finally, SSRS offers rich reporting capabilities, we can create interactive and parameterized reports. This enables us to develop reports that can meet diverse business needs

BQ 3 Target Report
What are the annual sales patterns for wine during peak seasons in the United States?

For this business question, we used Power BI again. Similarly, as business question 1, we used Power BI for this report because of the user-friendliness and seamless data integration with our SQL Server. Because we need basic charts to answer this question we decided to use Power BI again. Finally, the robust performance when handling large datasets made it easier to choose this reporting tool.

BQ 4 Target Report
How does the percentage of avid, hurried, and strange shoppers vary across different cities, and how can we tailor marketing strategies accordingly?

We used cubes from SSAS to answer this business question. We found that SSAS offers consistent and reliable results. Since we use cubes, and cubes store pre-aggregated data and calculations, we are sure that our results in the analytical report are consistent and reliable. Also, SSAS provides an enhanced business user experience, since business users can create their own ad-hoc reports and analysis without in-depth technical knowledge. Finally, cubes support the creation of dimensional hierarchies, this enables users to navigate data at different levels of granularity.

BQ 5 Target Report
What is the most popular coupon used for purchasing the top 20 juice categories?

This report used SSRS. SSRS offers robust reporting capabilities on its own, it gives a lot of flexibility when it comes to creating reports that suit different business needs and preferences. Some of the types of reporting are tabular reports, charts, and graphs. Subsequently, SSRS supports drill-down capabilities, this way users can navigate through different levels of data hierarchy and verify different summaries. To conclude, the tool can generate reports in various formats, PDF, Excel, Word, and a web server. It is very flexible and can accommodate different preferences for different business users.

Mappings From the Independent Data Marts to the Report Attributes

BQ 1 Mapping
Which are the top 5 categories for products across different stores between the years 1980 to 1997?

Data Mart Attributes	Dimension/Fact Tables	Filter	Report Attributes
Year	Date_Dim		Year
ProductName	Product_Dim	<Filter for top 5 product categories>	Product Name

TotalSales	Fact_Sales		Total Sales

BQ 2 Mapping

What are the year-over-year trends and patterns in alcohol (Beer, Wine, Spirit) sales from 1987 to 1997?

Data Mart Attributes	Dimension/Fact Tables	Filter	Report Attributes
Year	Date_Dim		Year
ProductName	Product_Dim	<Filter for Beer, Wine, Spirit>	Product Name
TotalSales	Fact_Sales		Total Sales

BQ 3 Mapping

What are the annual sales patterns for wine during peak seasons in the United States?

Data Mart Attributes	Dimension/Fact Tables	Filter	Report Attributes
Year	Date_Dim		Year
ProductName	Product_Dim	Wine	Product Categories
SpecialEvents	Date_Dim		Special Events
TotalSales	Fact_Sales		Total Sales

BQ 4 Mapping

How does the percentage of avid, hurried, and strange shoppers vary across different cities, and how can we tailor marketing strategies accordingly?

Data Mart Attributes	Dimension/Fact Tables	Filter	Report Attributes
City	Demo_Dim		City
TotalPercentAvid	Fact_Demo		Total Percent Avid
TotalPercentHurried	Fact_Demo		Total Percent Hurried

TotalPercentStrange	Fact_Demo		Total Percent Strange

BQ 5 Mapping
What is the most popular coupon used for purchasing the top 20 juice categories?

Data Mart Attributes	Dimension/Fact Tables	Filter	Report Attributes
CouponCategory	Coupon_Dim		Coupon Category
UPCDesc	UPC_Dim	<Grouped within each Coupon Category>	UPC Desc
TotalPurchases	Fact_Purchases		Total Purchases

Report Templates

Through the implementation of this project, we discovered a variety of visualization tools, which we have highlighted below:

SSAS – Microsoft SQL Server Analysis Services
Analysis services provide company insights and decision support through the use of an analytical data engine. Deploying a multidimensional model as a database to a server instance is a typical step in the deployment of a data analysis solution using SSAS.

SSRS – Microsoft SQL Server Reporting Services
Reports are generated for organizational data visualization using a server-based platform. This program makes it simple to create drill-down and ad hoc reports for the SQL Server data warehouse because it is directly related to Visual Studio and Microsoft SQL formatting tools.

Microsoft Power BI
Power BI is a business intelligence product that is more recent and technologically sophisticated because of its extensive graphical features. Most of the functions are available without charge, and it can be accessed through both desktop and online browser platforms.

Report Implementation

BQ 1 Report

Which are the top 5 categories for products across different stores between the years 1980 to 1997?

Visualization Method: Report using Microsoft Power BI

Final Report:

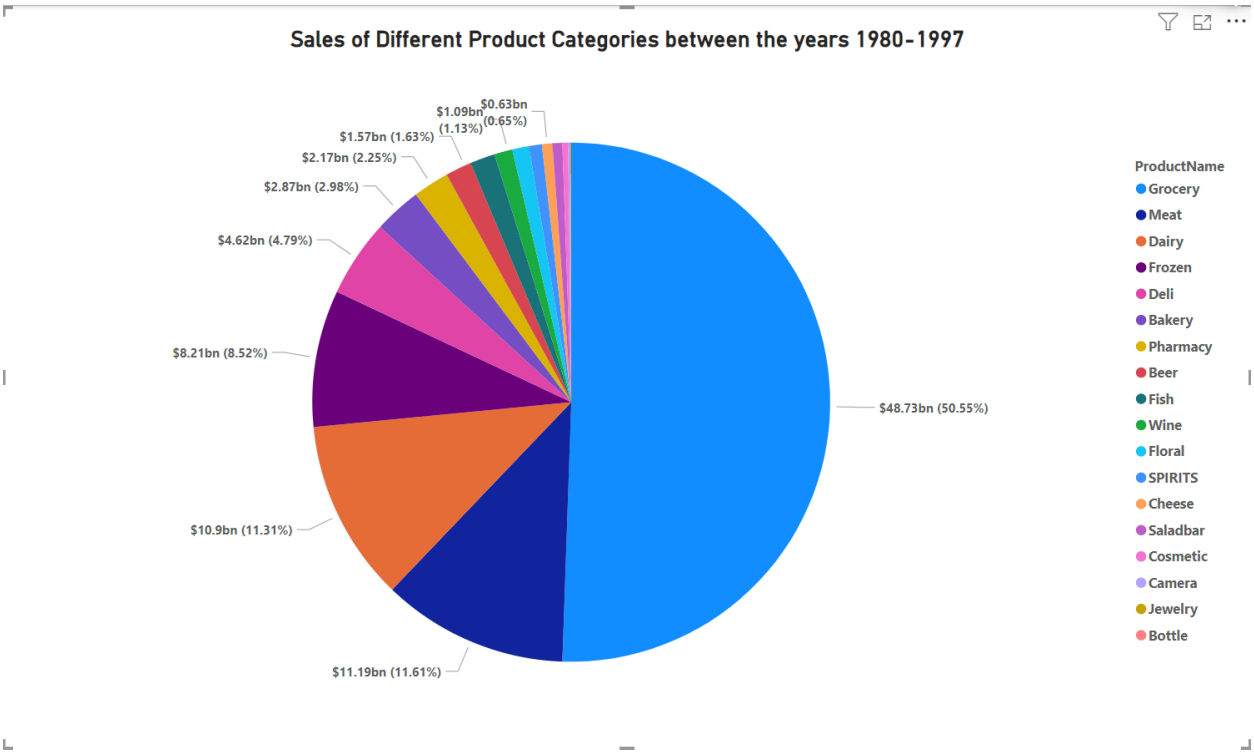


Figure: Power BI visualization showing sales distribution across different years

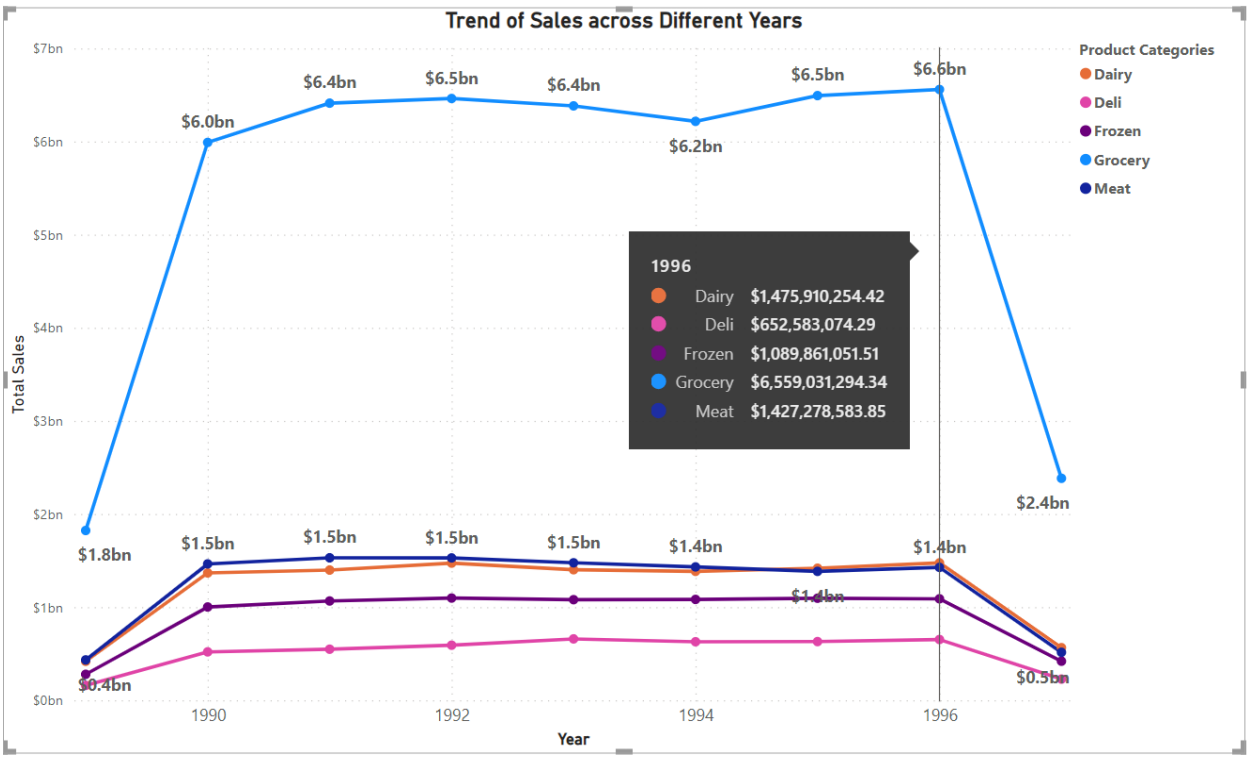


Figure: Power BI visualization showing yearly trend of top five product categories

Analysis: The Power BI report displays revenue across all product categories, highlighting the top products within each category. The analysis identifies Grocery, Meat, Dairy, Frozen, and Deli as the top 5 products across all categories. The pie chart displays the sales distribution of each category and also enables us to understand the proportion contribution of each product category. Thus, Grocery was identified as the top contributing category among the other top five products. Understanding these top 5 products can help the store optimize their revenue. The visualization of yearly trends of the top performing products helps us derive valuable insights into the dynamics of these products over time.

By gaining insights into these leading products, the store can strategically optimize revenue. This includes allocating resources to high-performing items, enhancing promotional activities, and focusing on the most frequently sold products for increased efficiency and profitability.

Implementation Steps: The analysis was conducted using Microsoft Power BI, utilizing data from the Sales_Data_Mart. The data was imported from the SQL Server Database and the tables Date_Dim, Store_Dim, Product_Dim and Fact_Sales were selected.

The data is filtered for the top five product categories in the yearly sales trends. The visualization includes a legend showing the impact of Product Categories on sales across years.

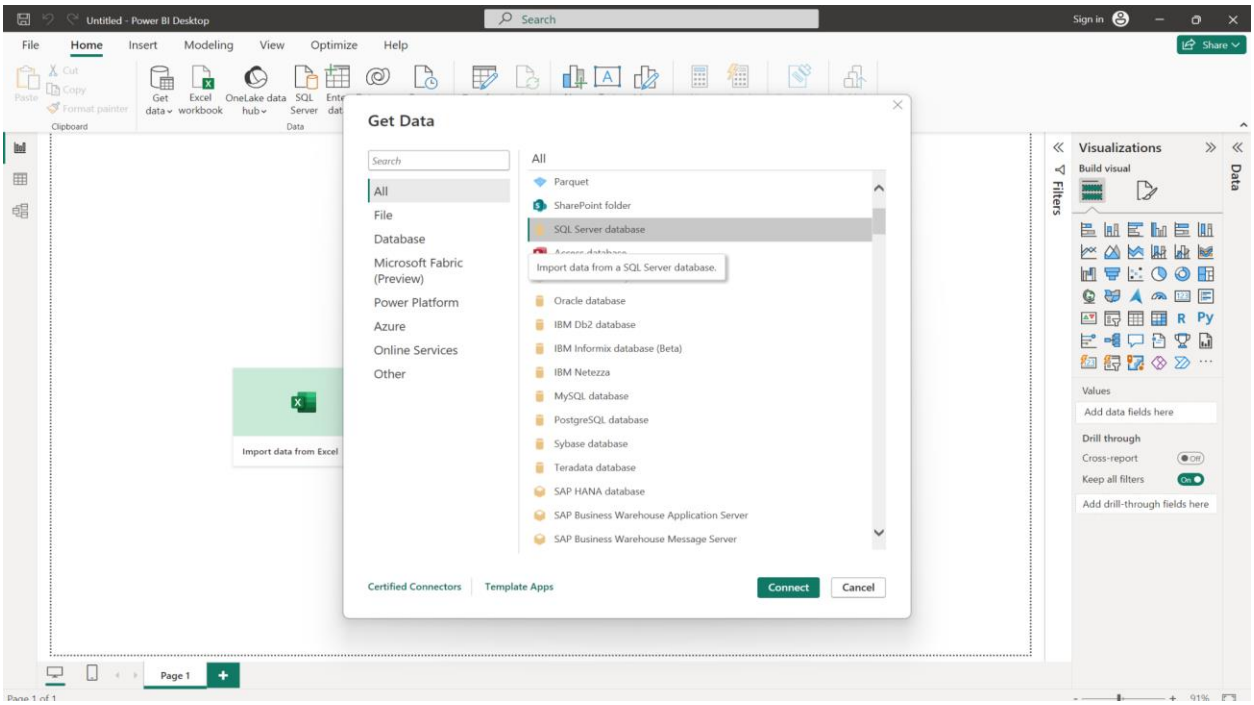


Figure: Importing data from SQL Server Database

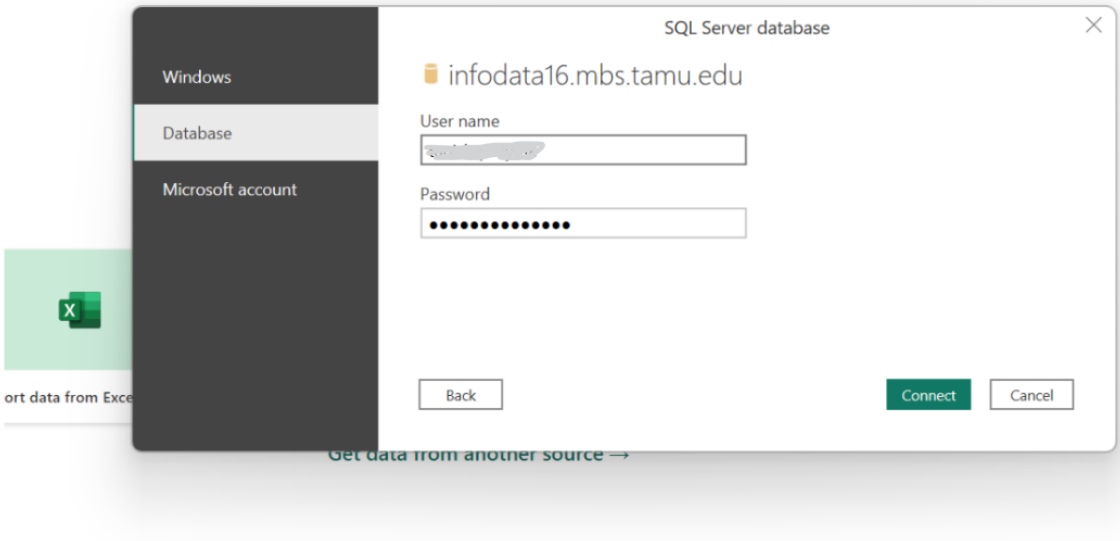


Figure: Establishing connection to the SQL Server

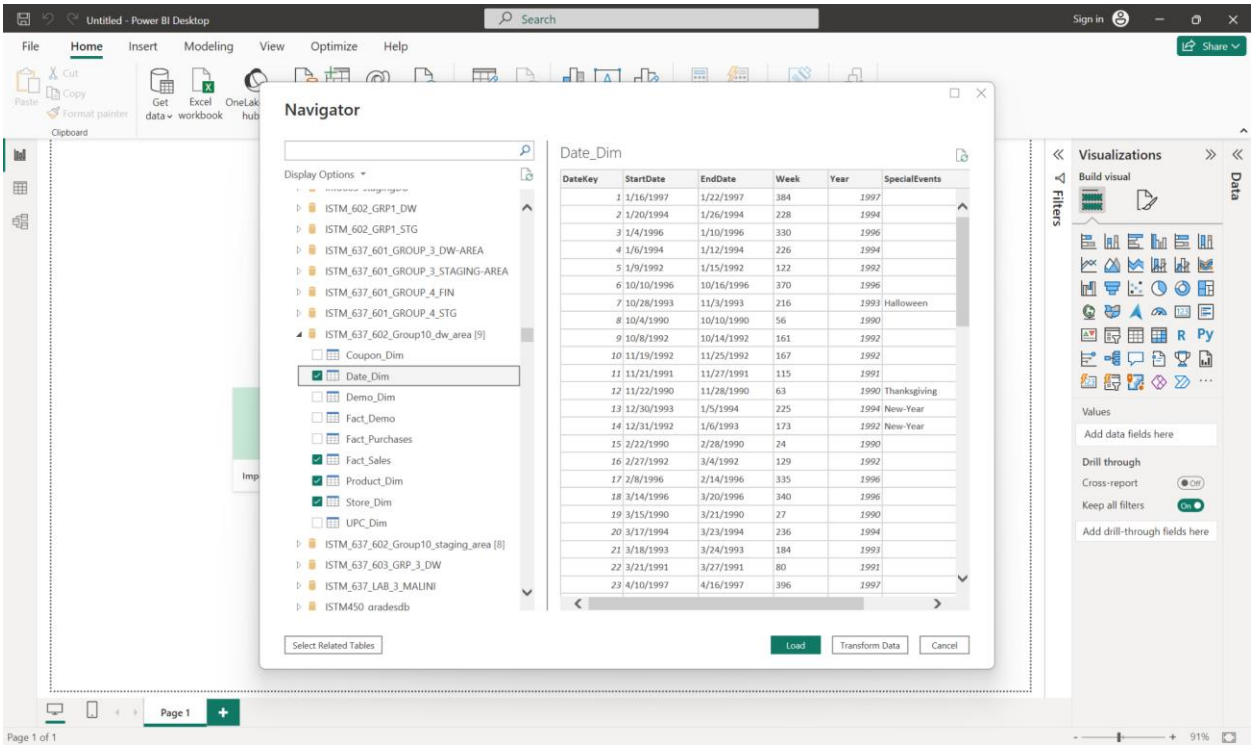


Figure: Selecting Sales_Data_Mart tables for visualization

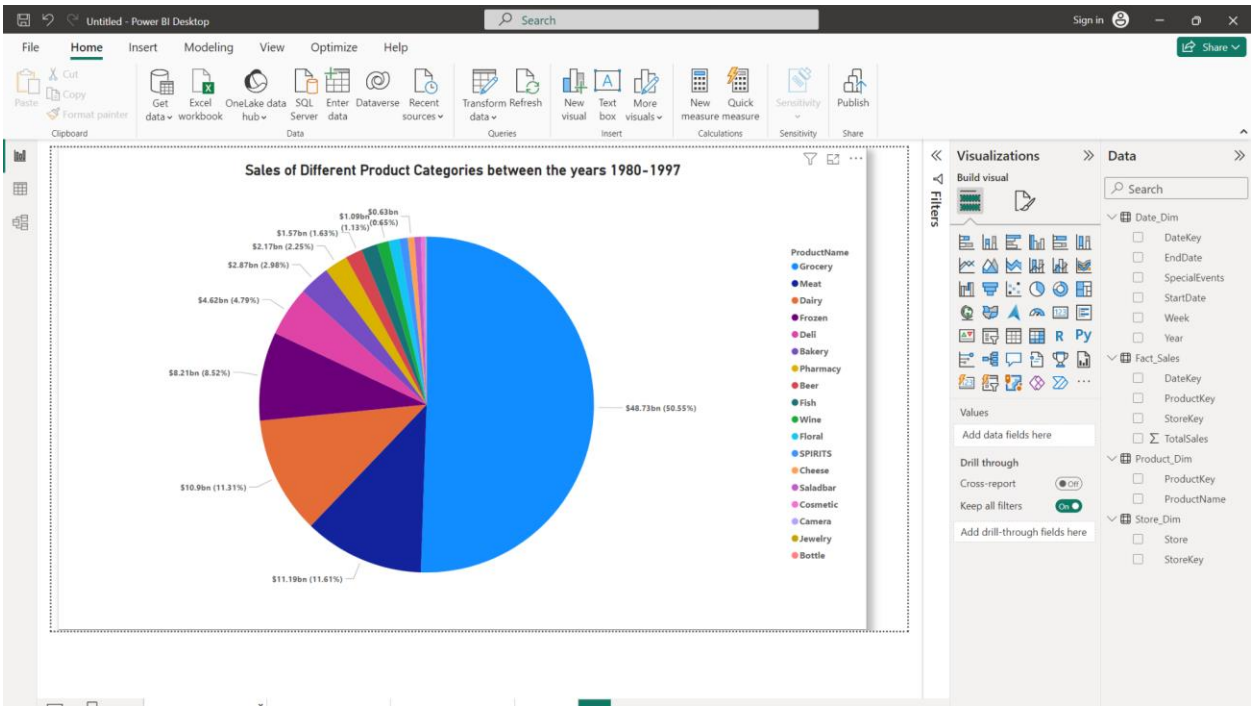


Figure: Power BI visualization showing sales distribution across different years

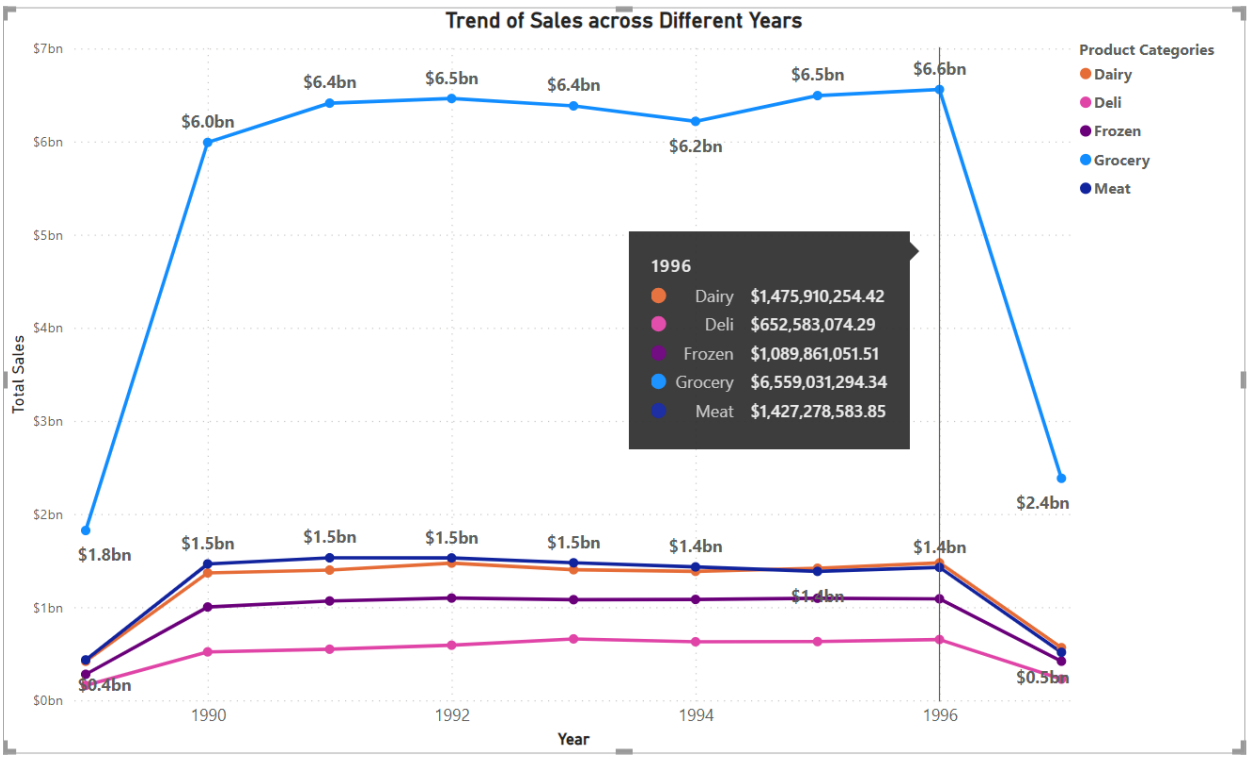


Figure: Power BI visualization showing yearly trend of top five product categories

BQ 2 Report

What are the year-over-year trends and patterns in alcohol (Beer, Wine, Spirit) sales from 1987 to 1997?

Visualization Method: Report from multidimensional cube using SSAS and SSRS

Final Report:

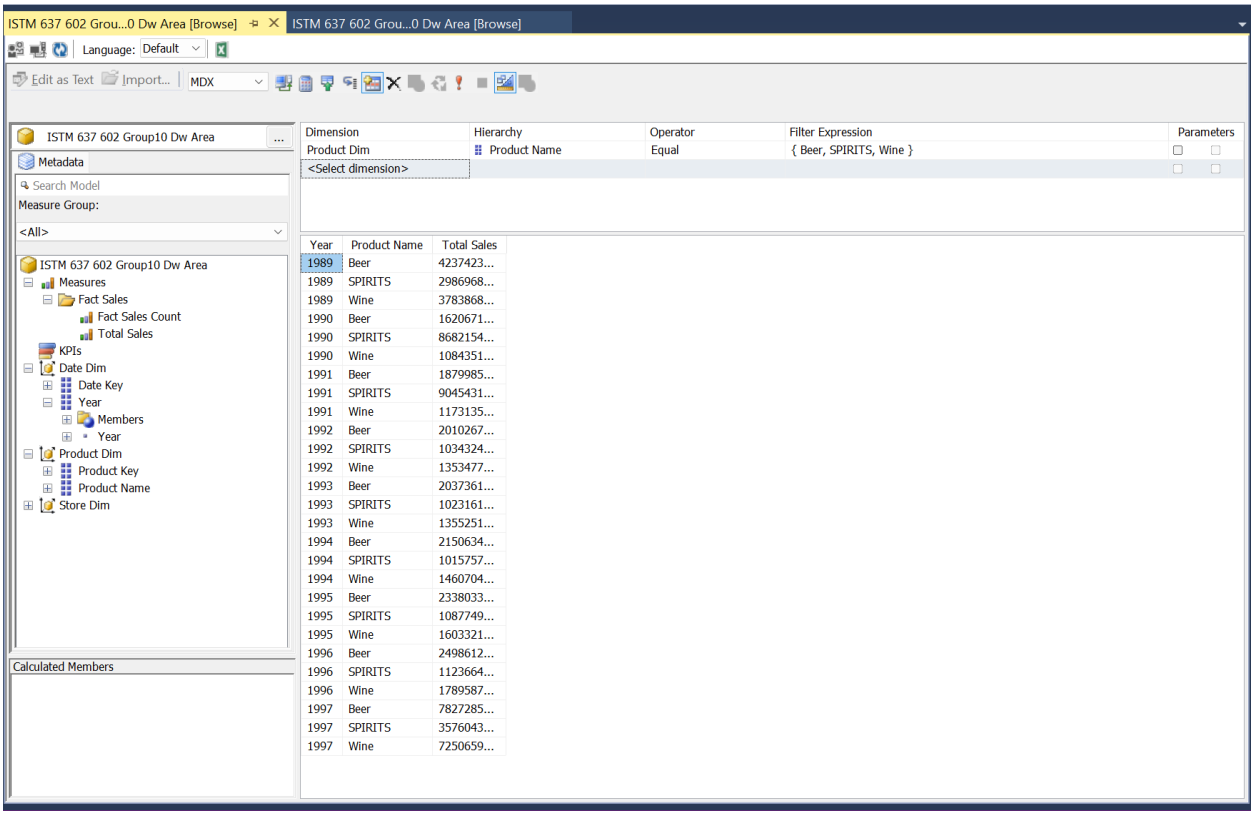


Figure: Multidimensional Analysis cube using SSAS

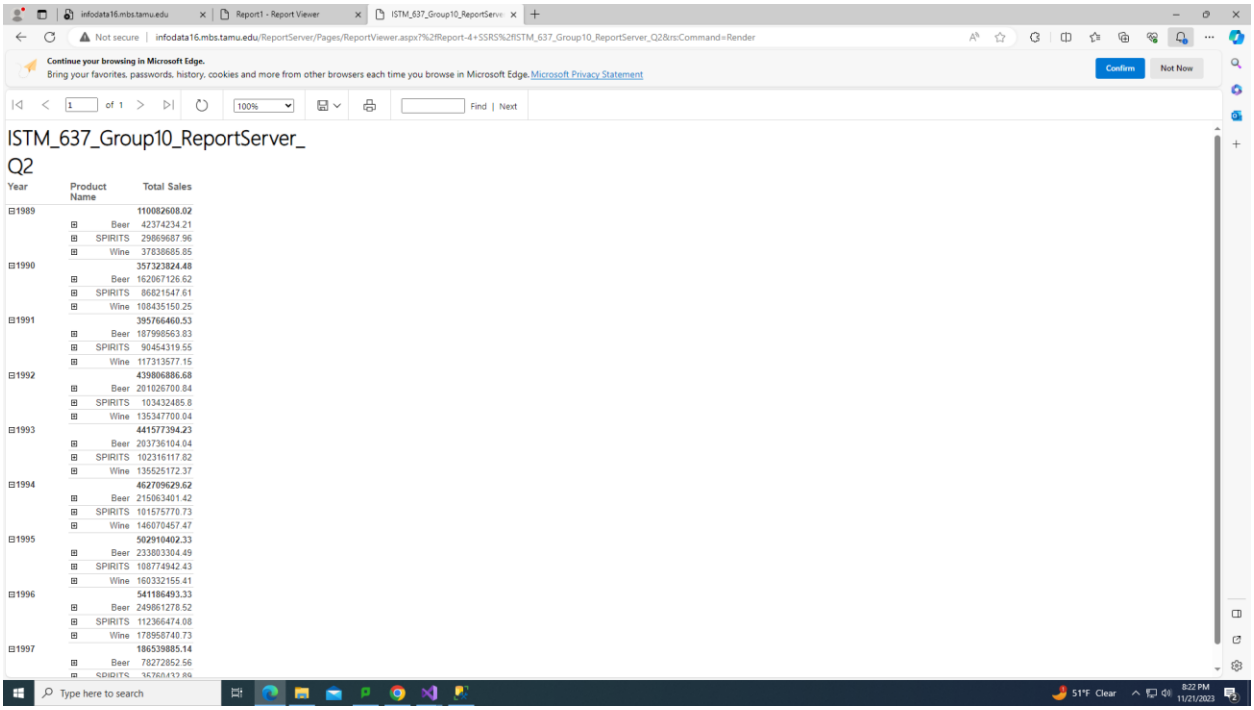


Figure: SSRS report built on SSAS Cube

Analysis: The report highlights the consistent top ranking of beer in the alcohol category, indicating that beer is a prevalent preference among consumers. This insight is valuable for DFF, as it can leverage the popularity of beer to attract more customers and boost sales. Implementing targeted promotions focused on beer, and linking loyalty programs to popular beer brands, could be effective strategies.

Additionally, analyzing the year-over-year trend in alcohol sales, it's evident that 1996 marked the peak for alcohol sales. This information provides a valuable opportunity for DFF to delve deeper into the factors contributing to this peak and explore strategies to replicate or sustain such scenarios.

Implementation: For this implementation, we have used multidimensional cube structure from SSAS and the reporting is done through SSRS. We have utilized data from the Sales_Data_Mart. The data was imported from the SQL Server Database and the tables Date_Dim, Store_Dim, Product_Dim and Fact_Sales were selected.

In SSRS, trends have been analyzed by first grouping data based on year and then further refining the analysis by specific alcohol product categories (Wine, Beer, Spirit) to obtain total sales.

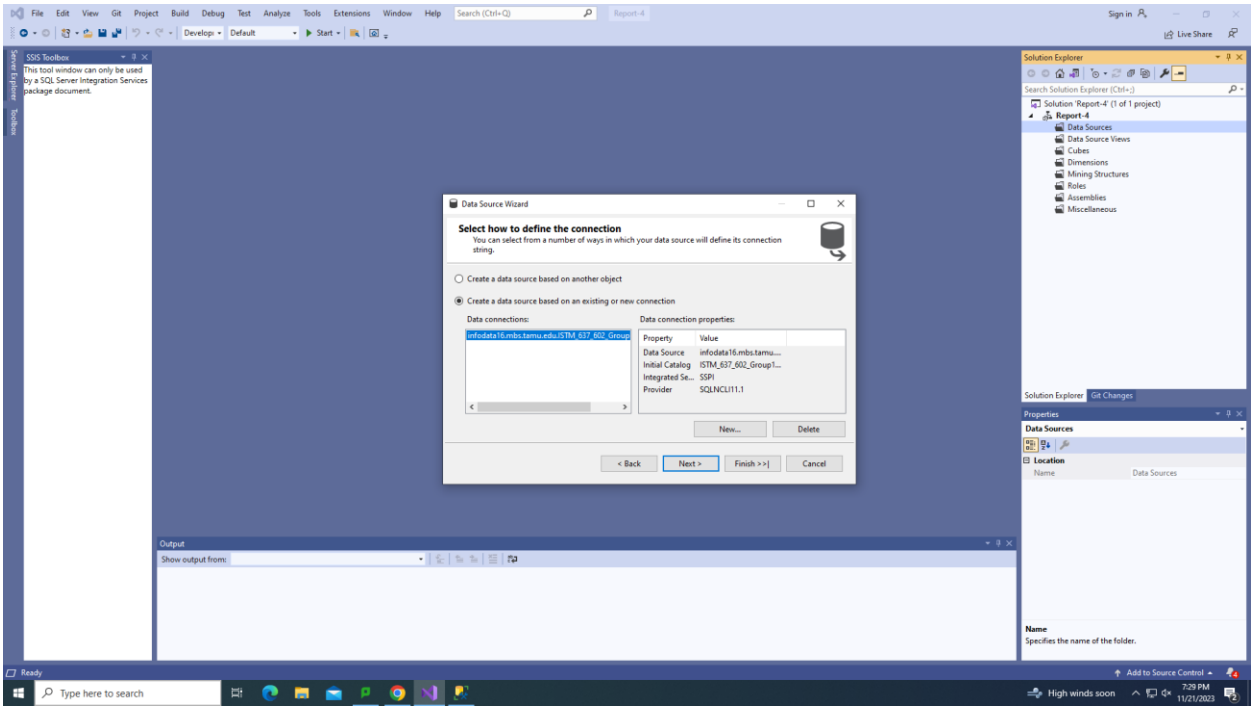


Figure: Creating datasource in SSAS

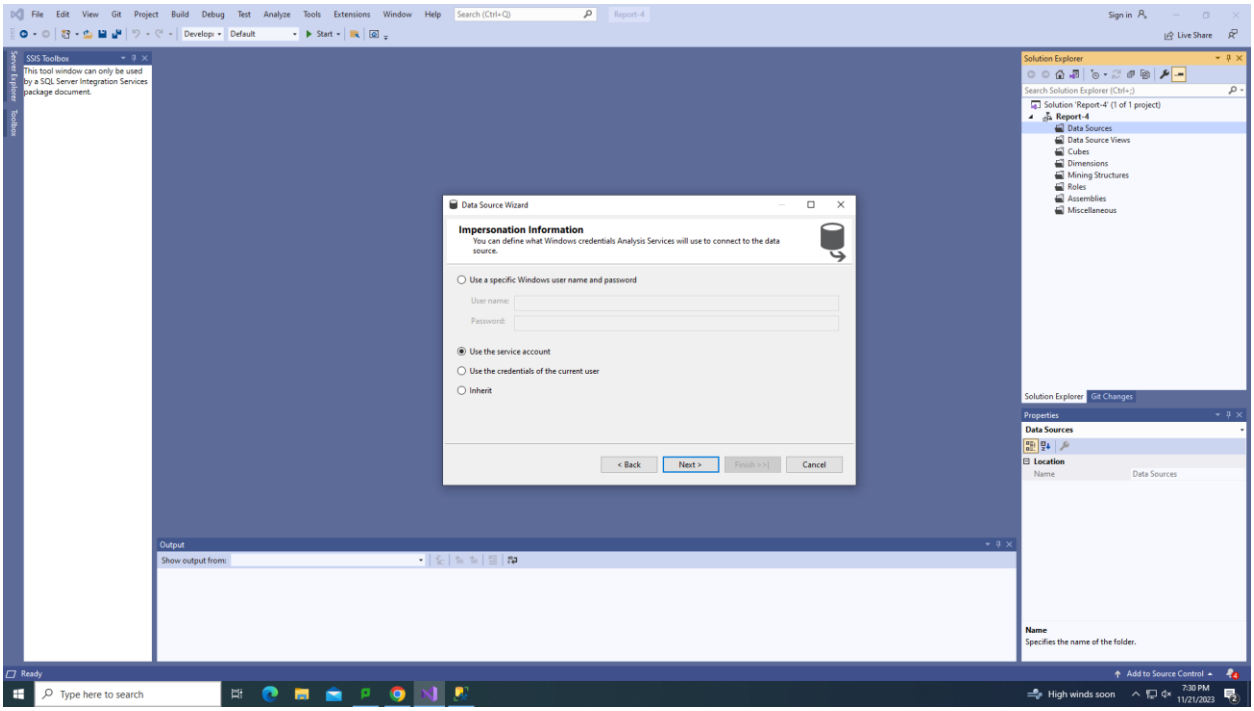


Figure: Using service account to access server

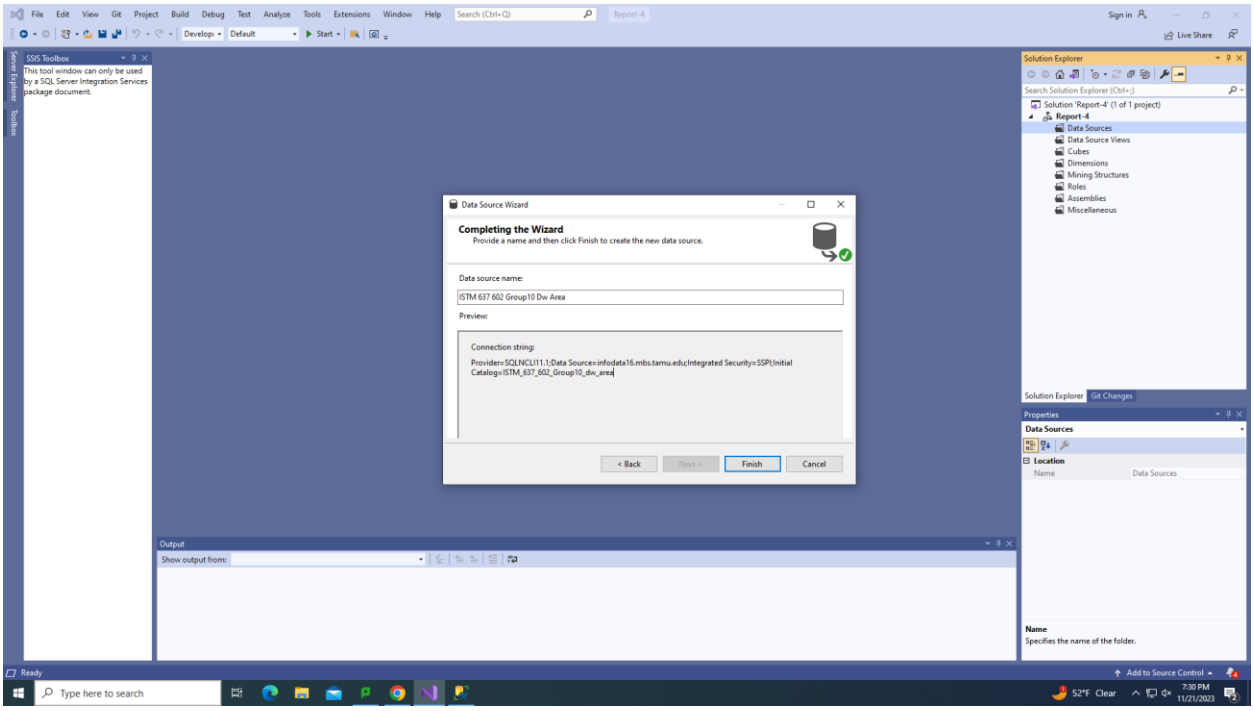


Figure: Establishing datasource connection

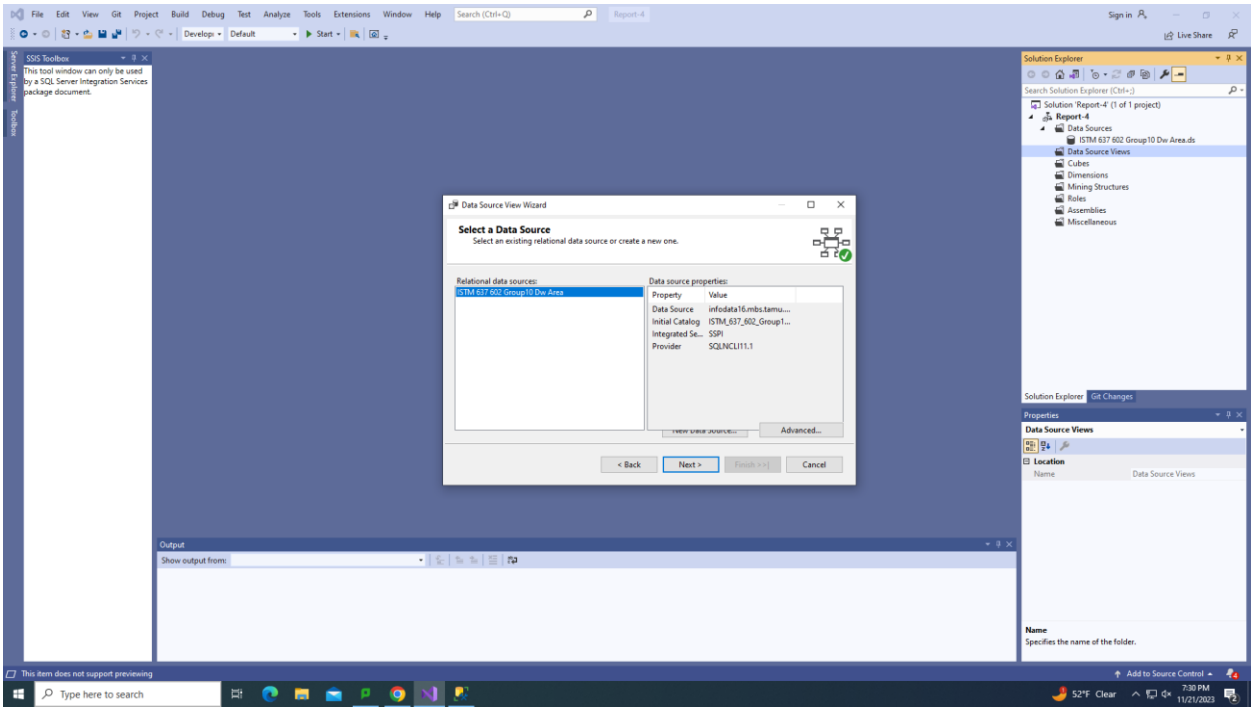


Figure: Creating data source view for the dw_area source

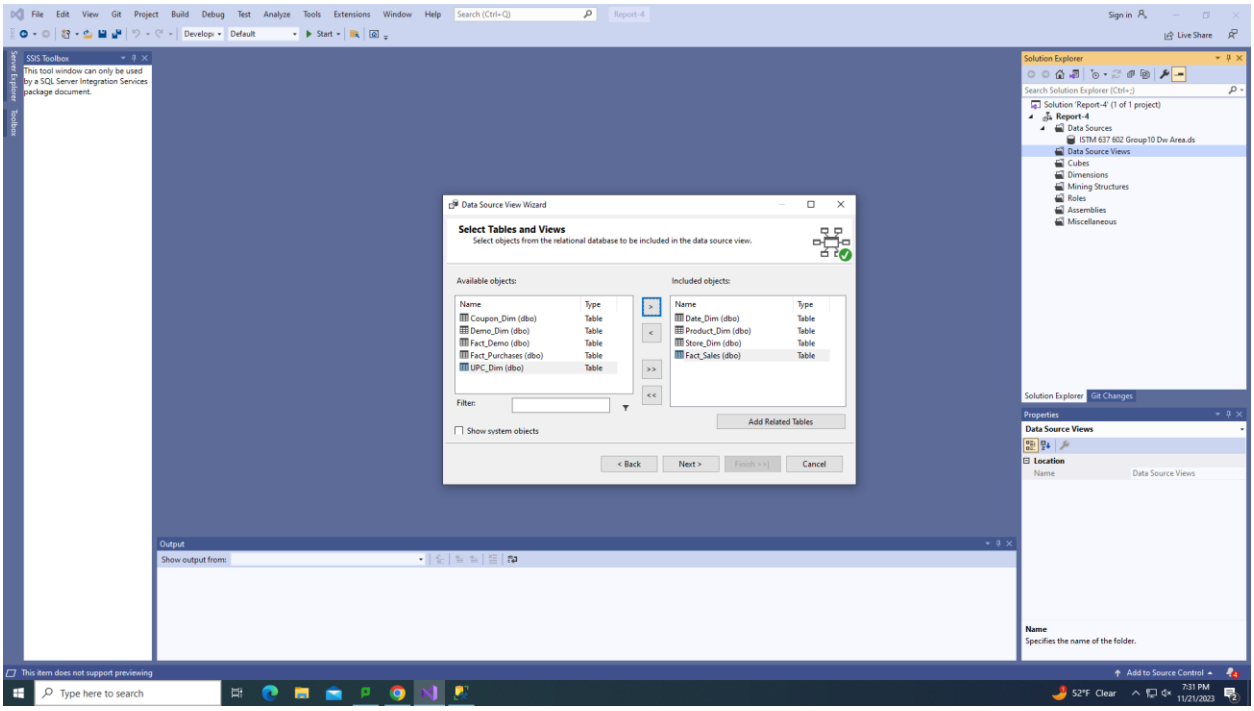


Figure: Mapping dimension and fact tables to the data source view

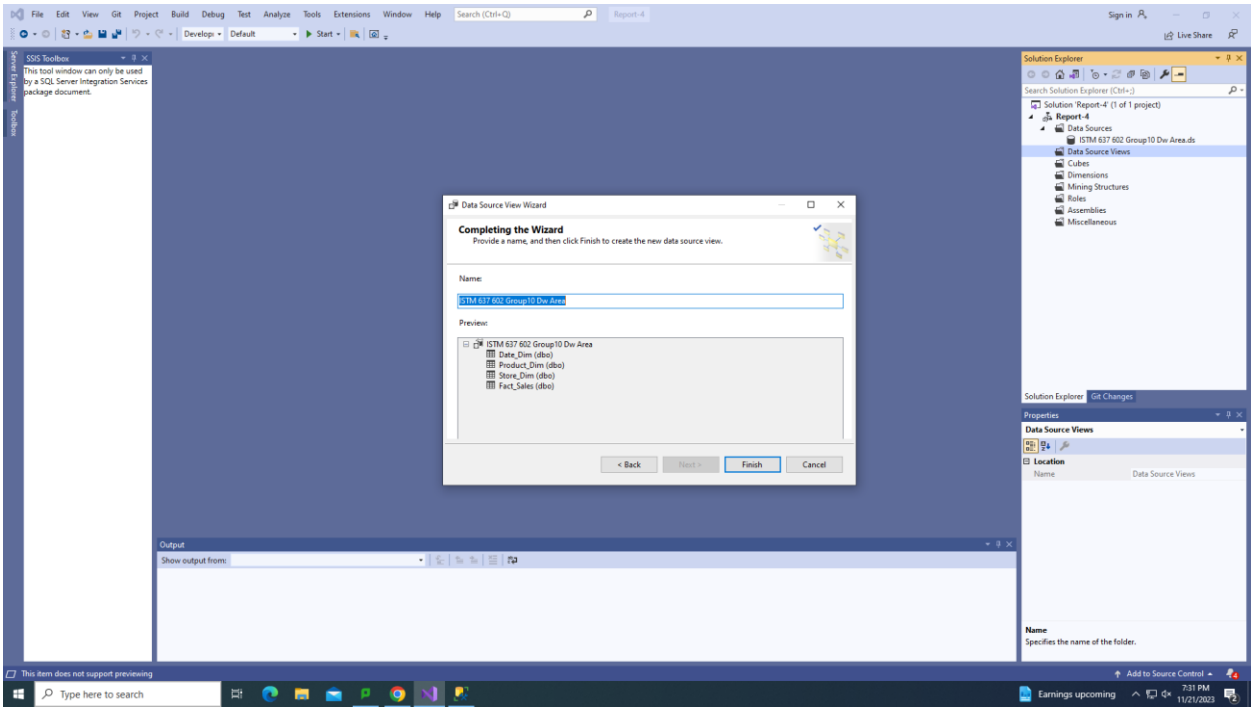


Figure: Finalizing the data source view

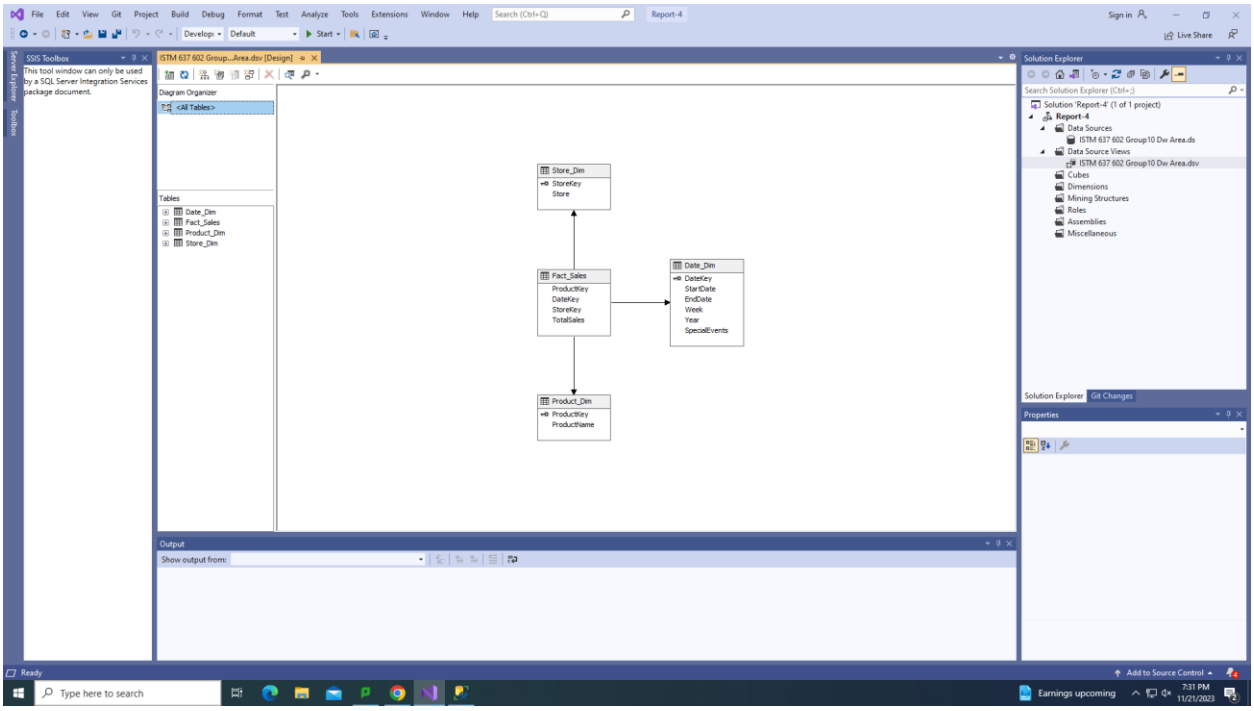


Figure: Data source view for Sales_Data_Mart

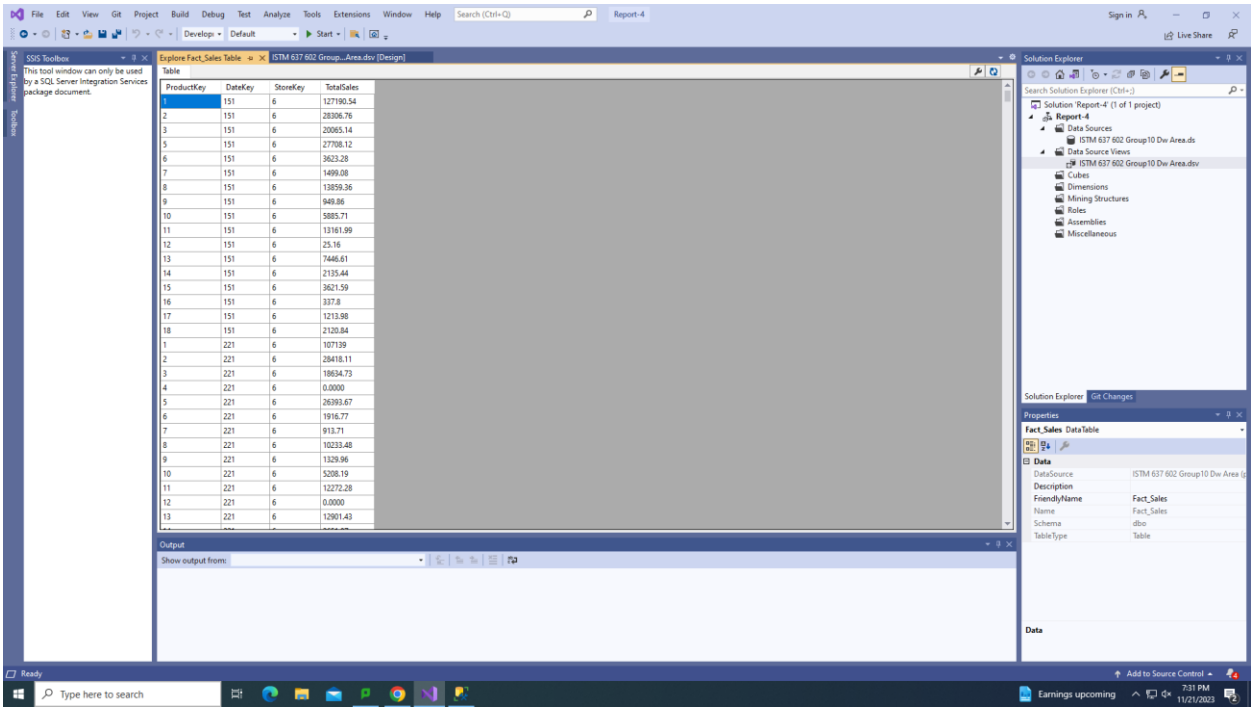


Figure: Expanded data source view for Sales_Data_Mart

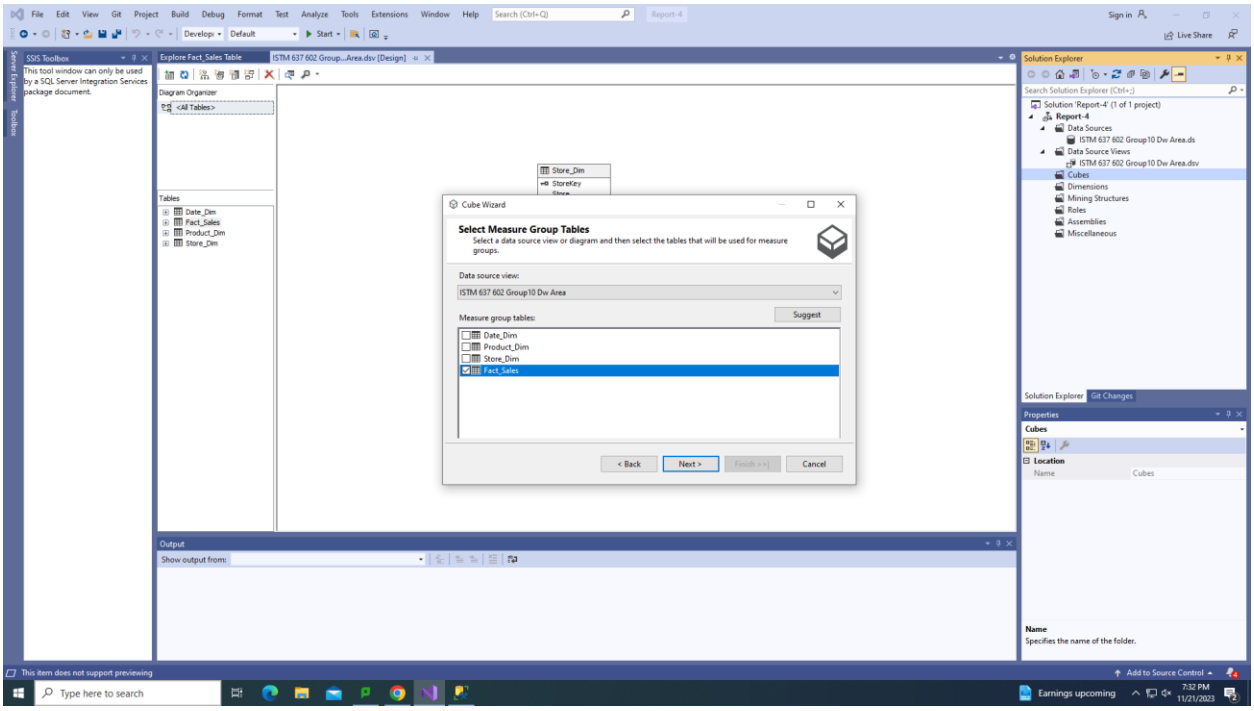


Figure: Initializing the cube structure

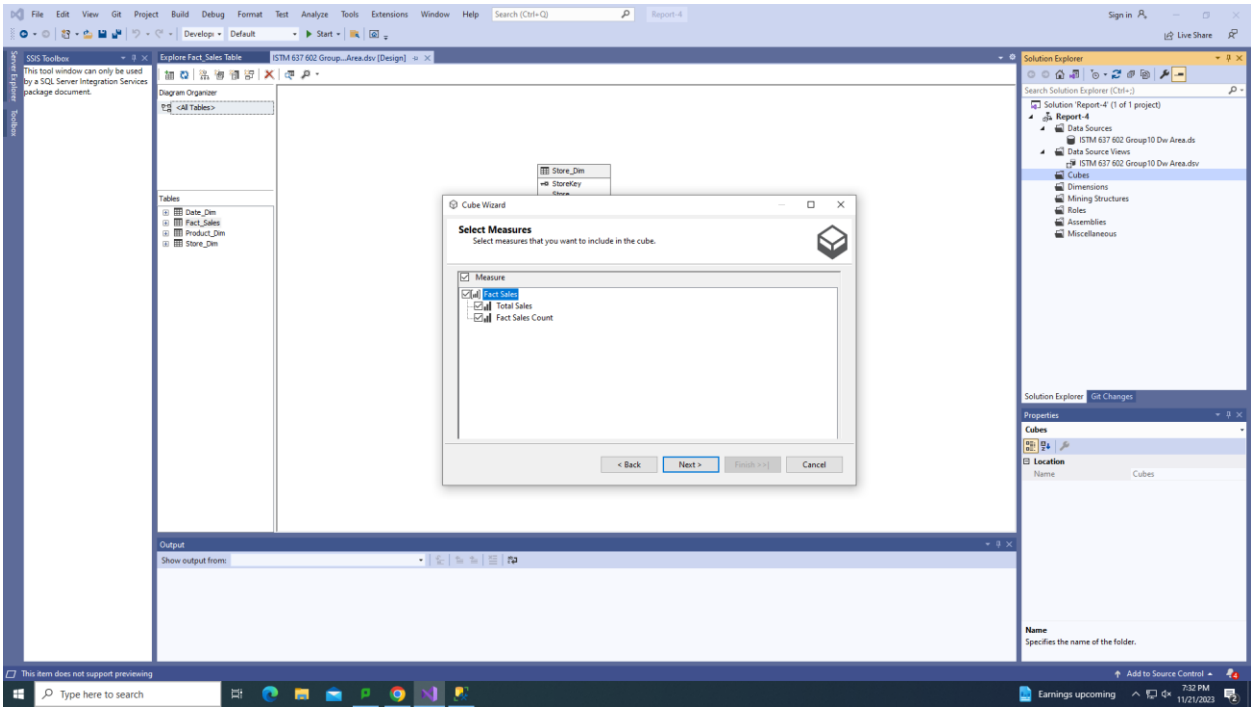


Figure: Selecting fact table attributes for cube structure

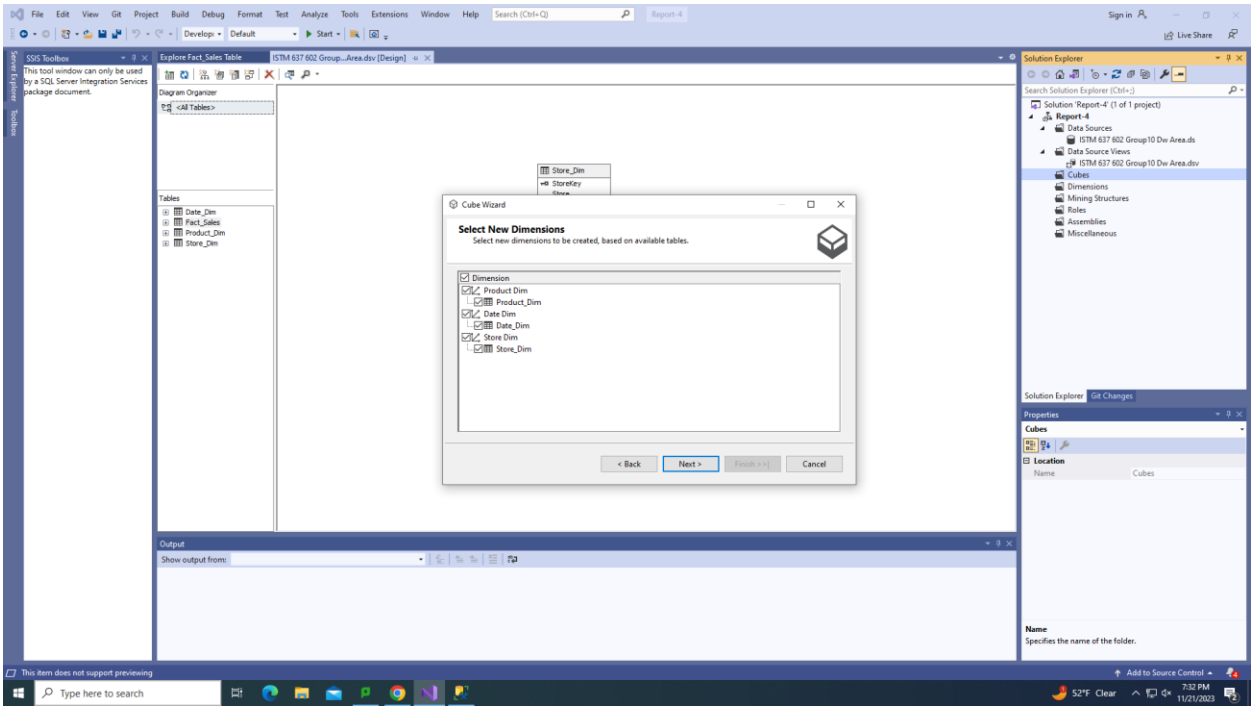


Figure: Selecting dimension table attributes for cube structure

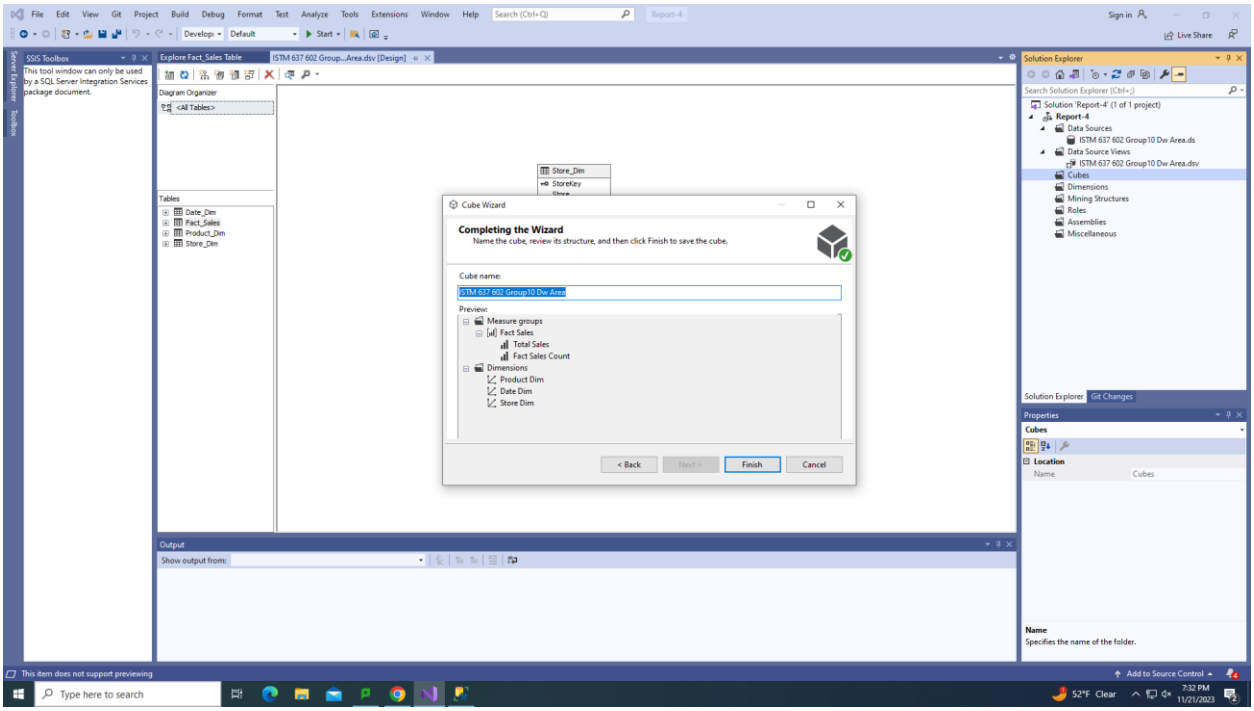


Figure: Finalizing creation of cube structure

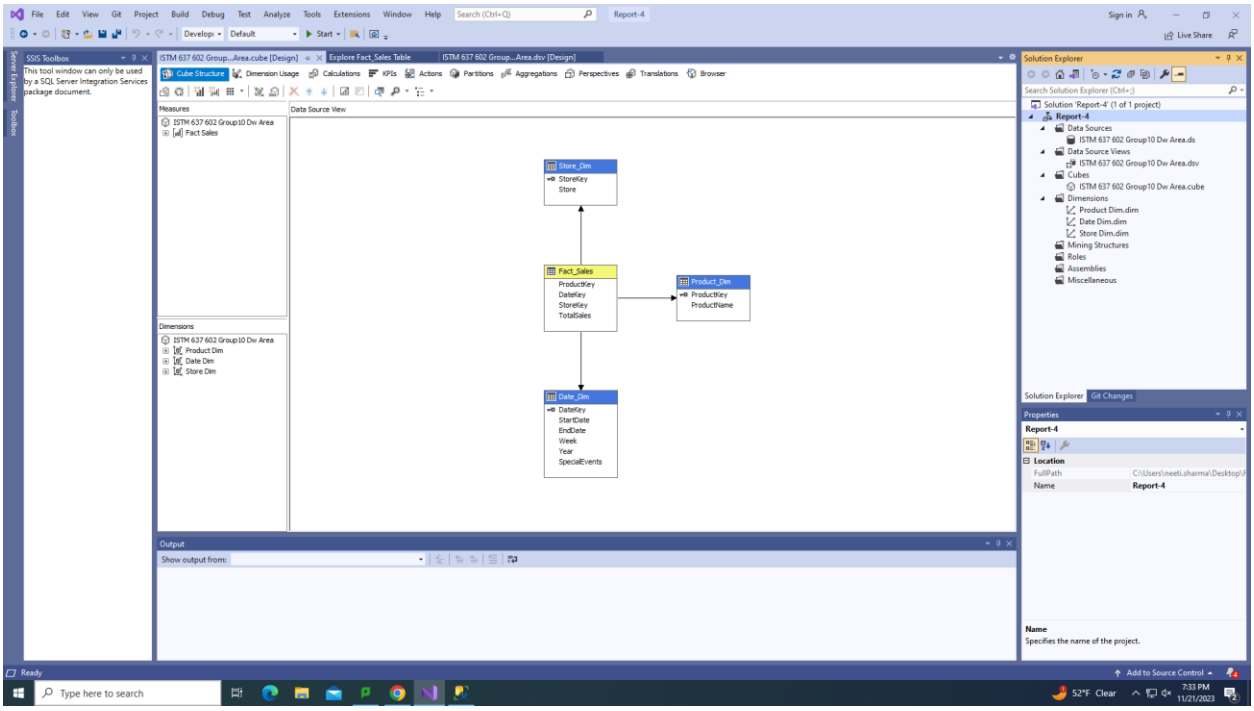


Figure: Cube structure for Sales_Data_Mart

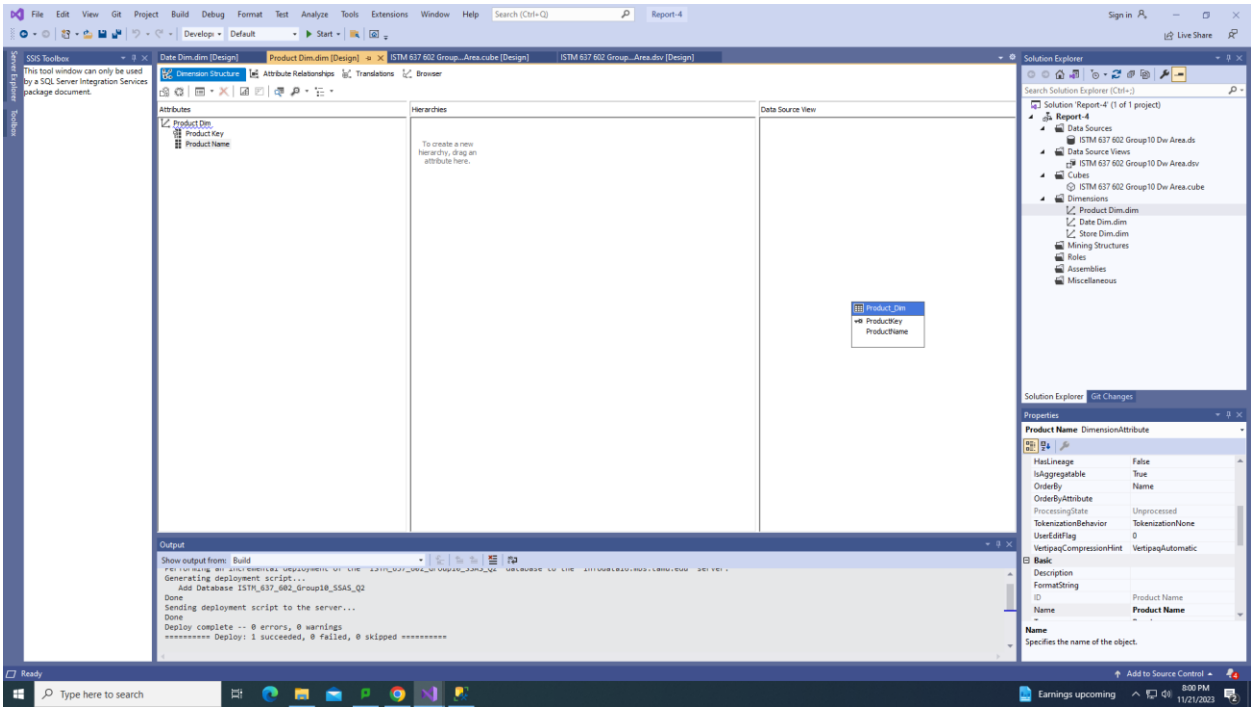


Figure: Selecting Dim table attributes in the cube structure

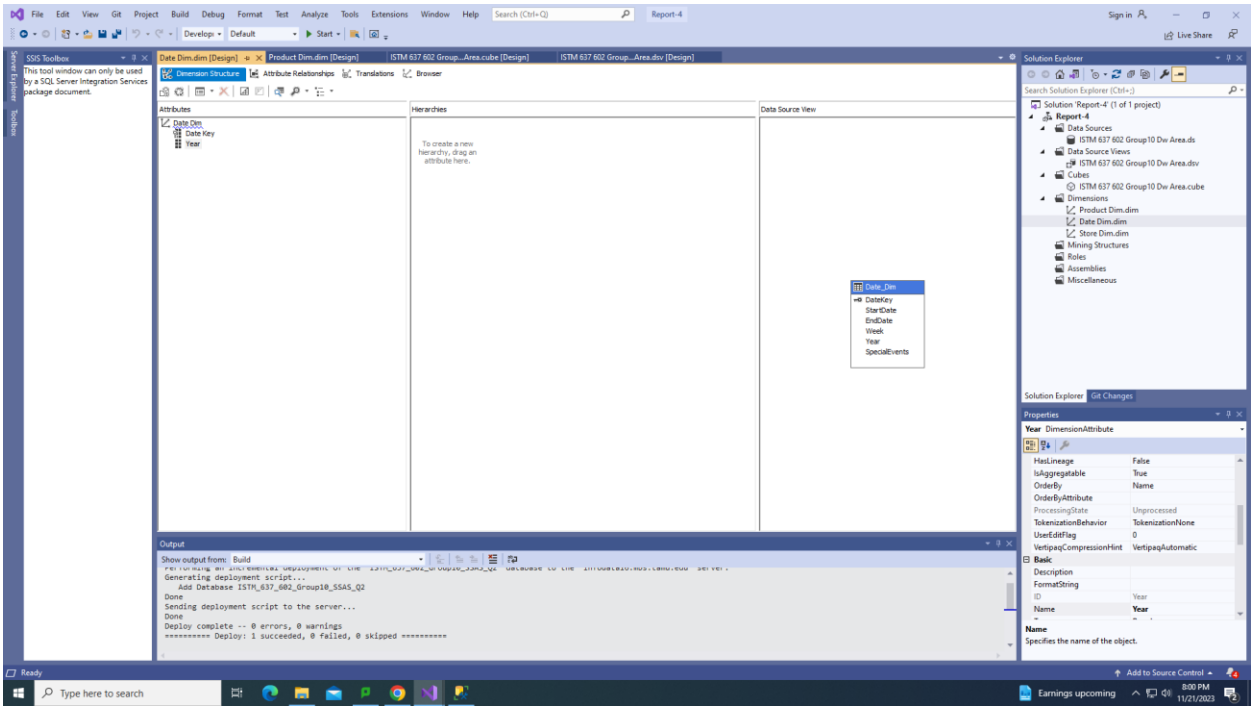


Figure: Selecting Dim table attributes in the cube structure

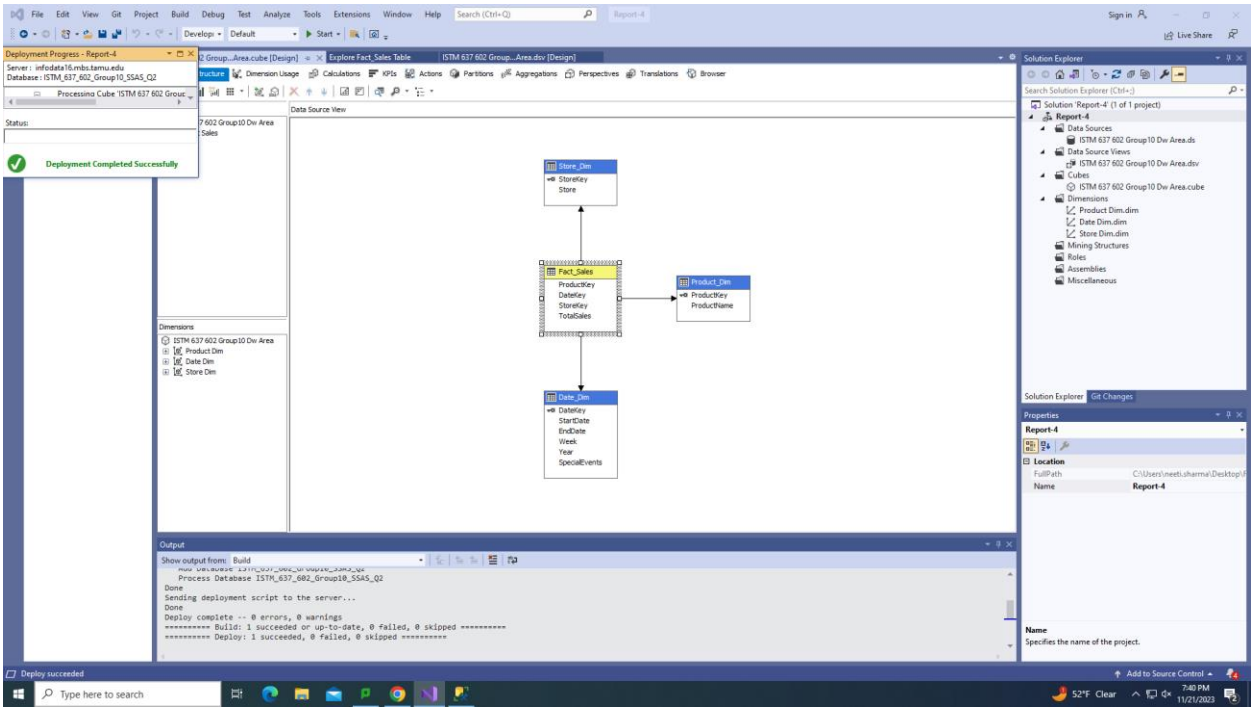


Figure: Deploying cube structure

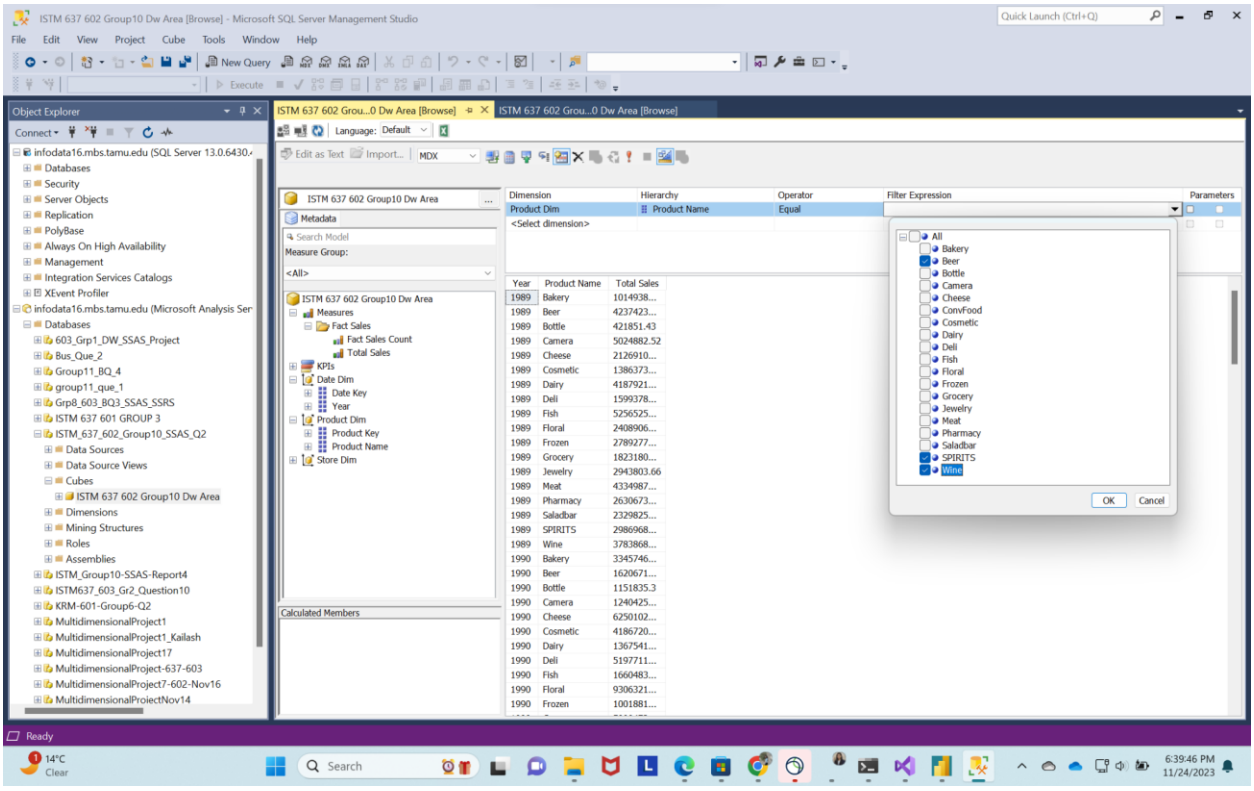


Figure: Filtering beer, wine and spirits for analysis in SSAS

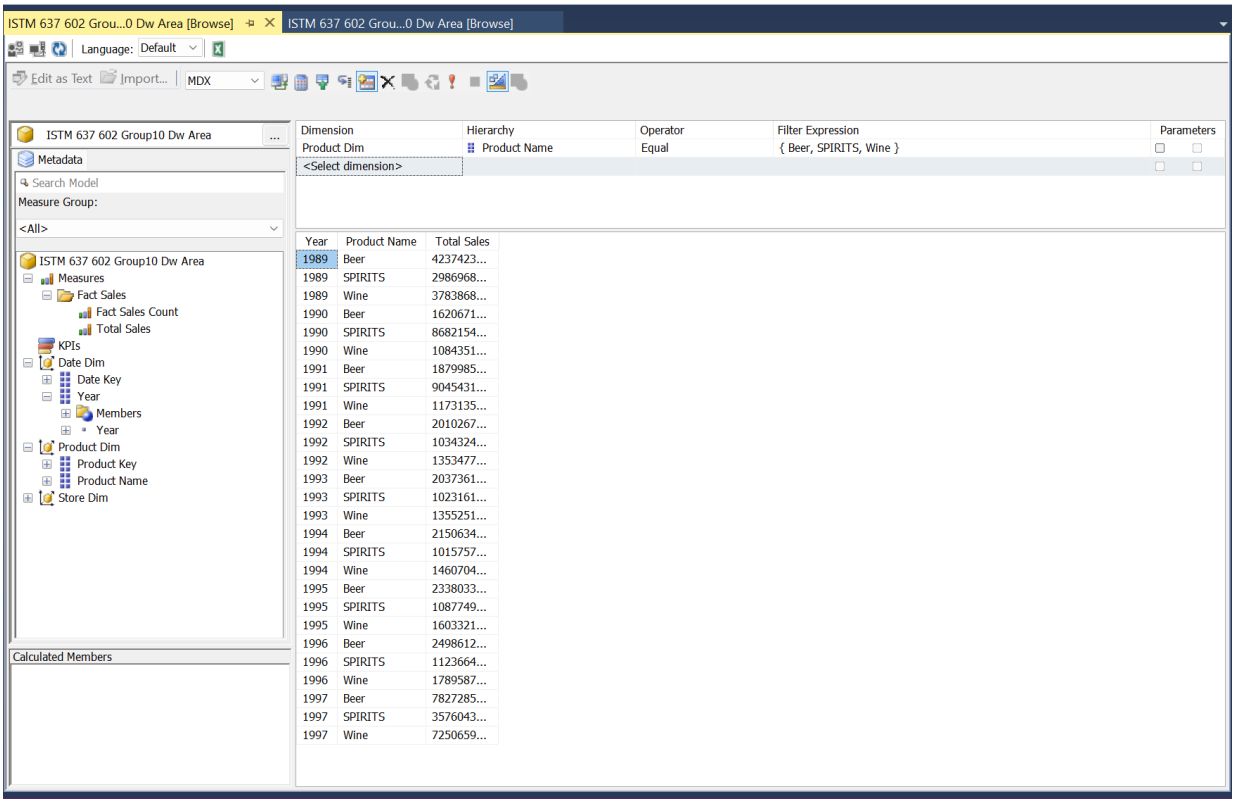


Figure: Multidimensional Analysis cube using SSAS

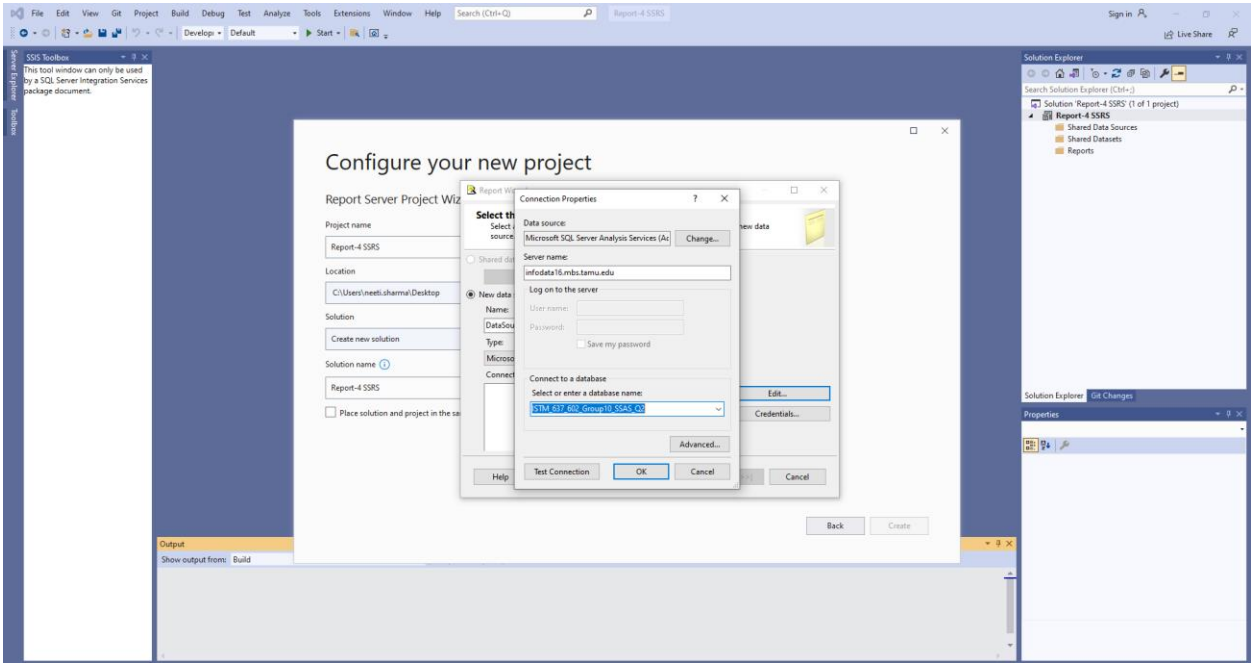


Figure: Selecting the source as Analysis Services for the Report Server

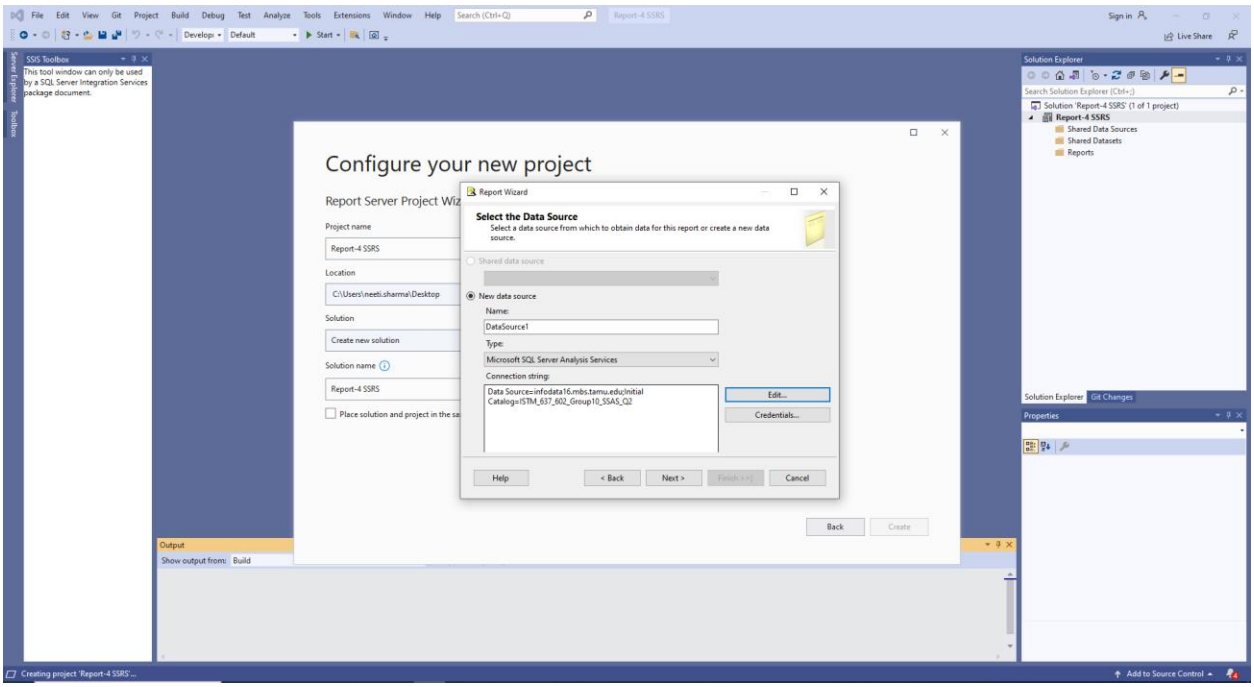


Figure: Creating a new data source from the Analysis Services

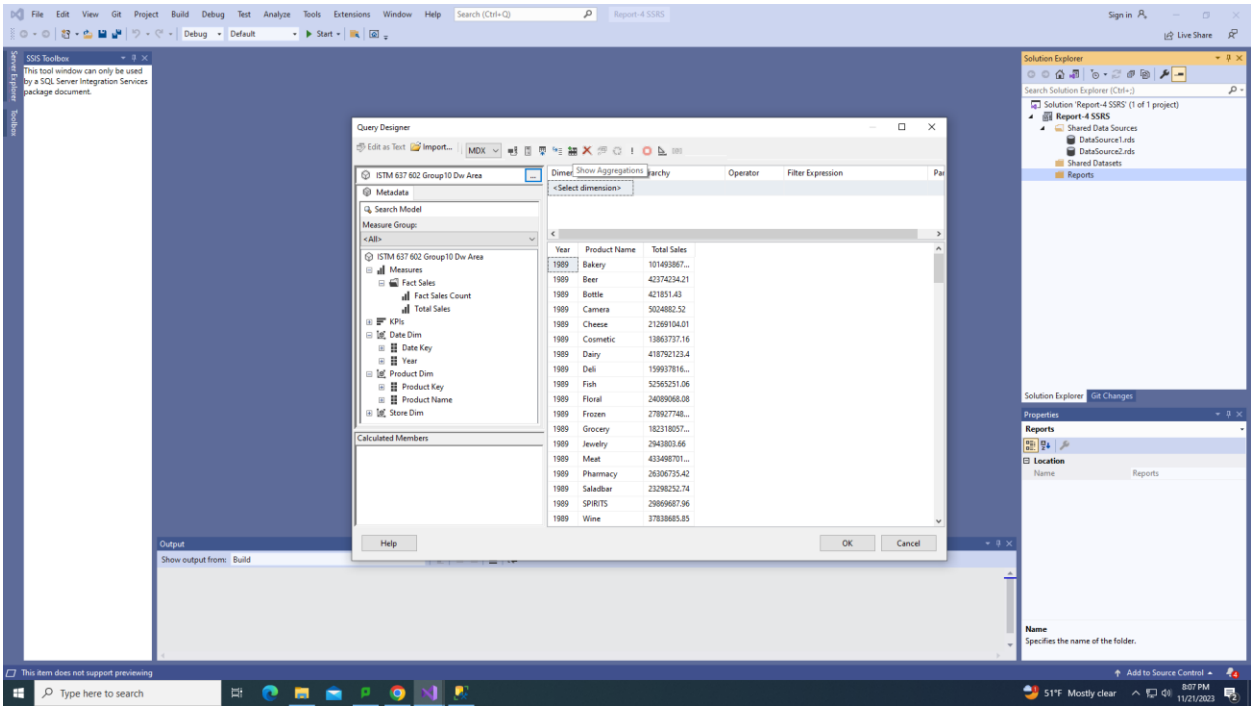


Figure: Selecting the relevant field for analysis - Year, Product Name, and Total Sales

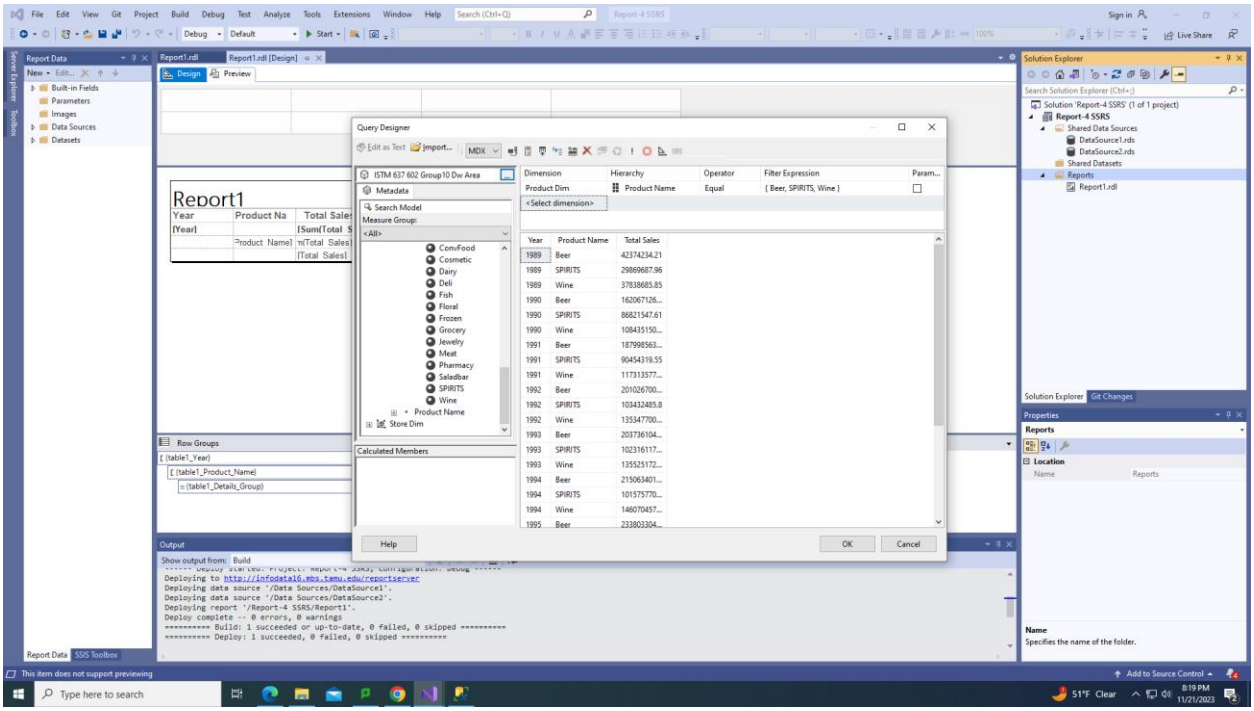


Figure: Applying a filter expression for Product Name to only include alcohol product categories

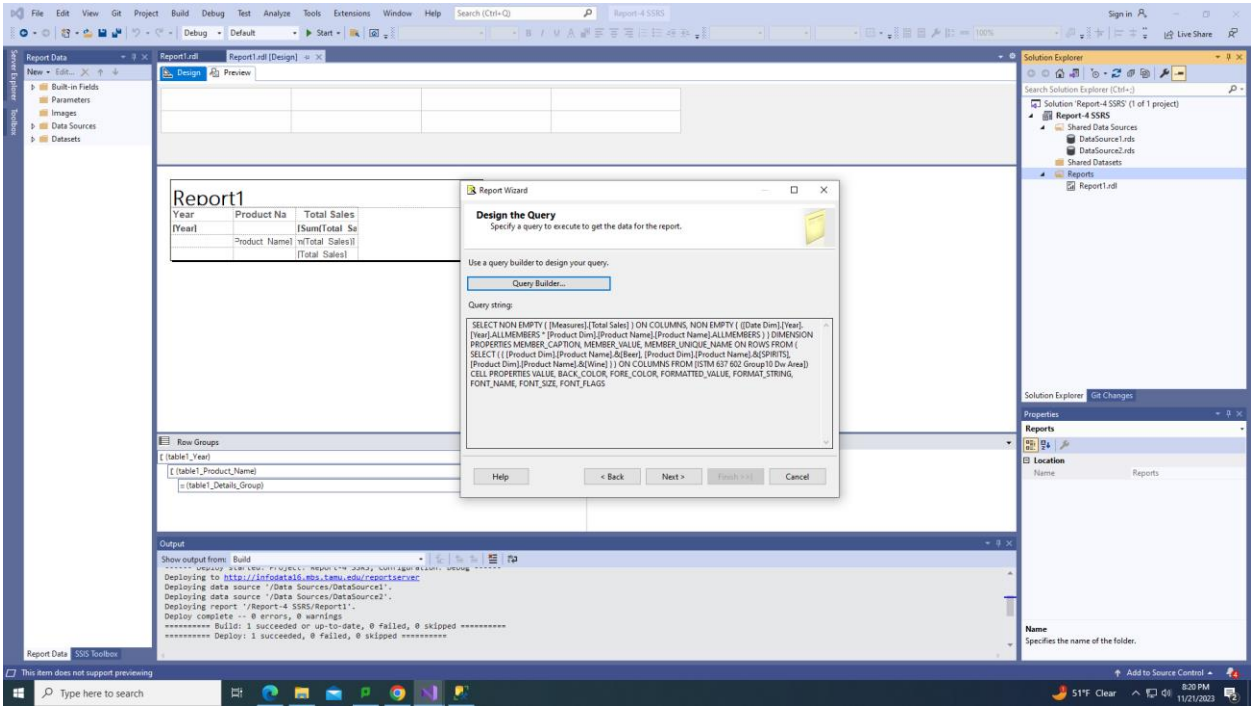


Figure: Query Builder for the Report Server

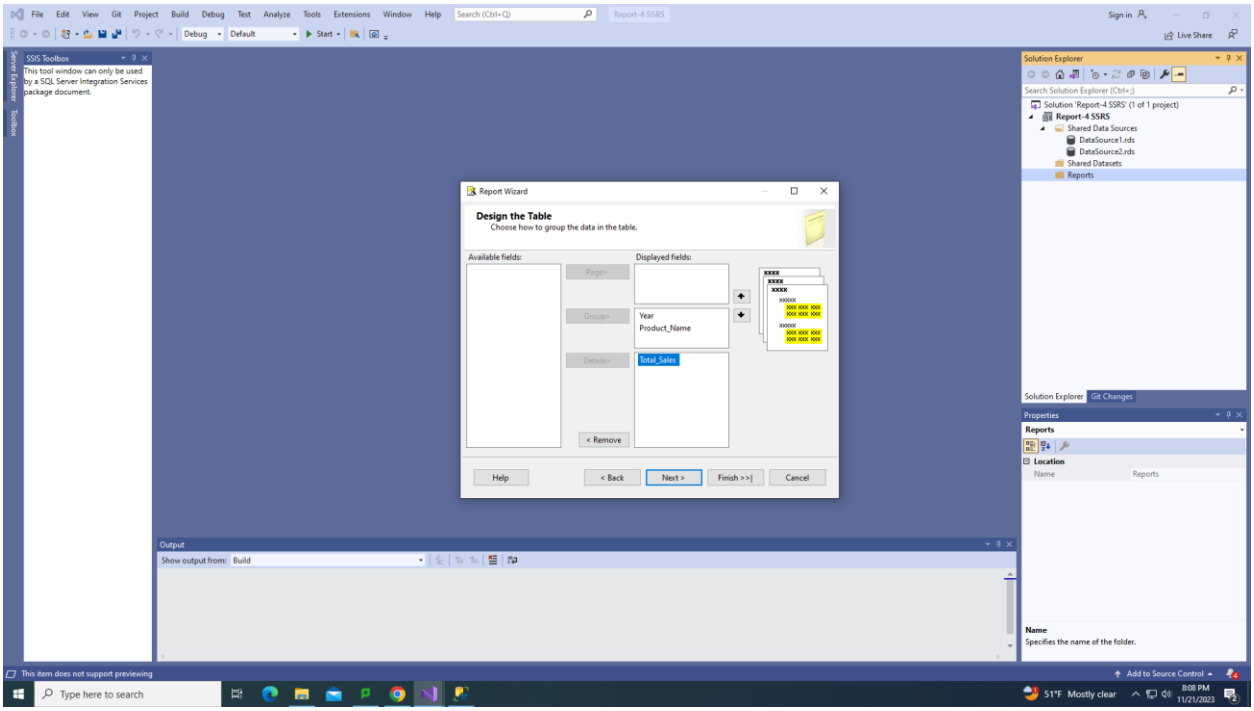


Figure: Designing the table by placing the Product Names within different Years and selecting Stepped Table Layout

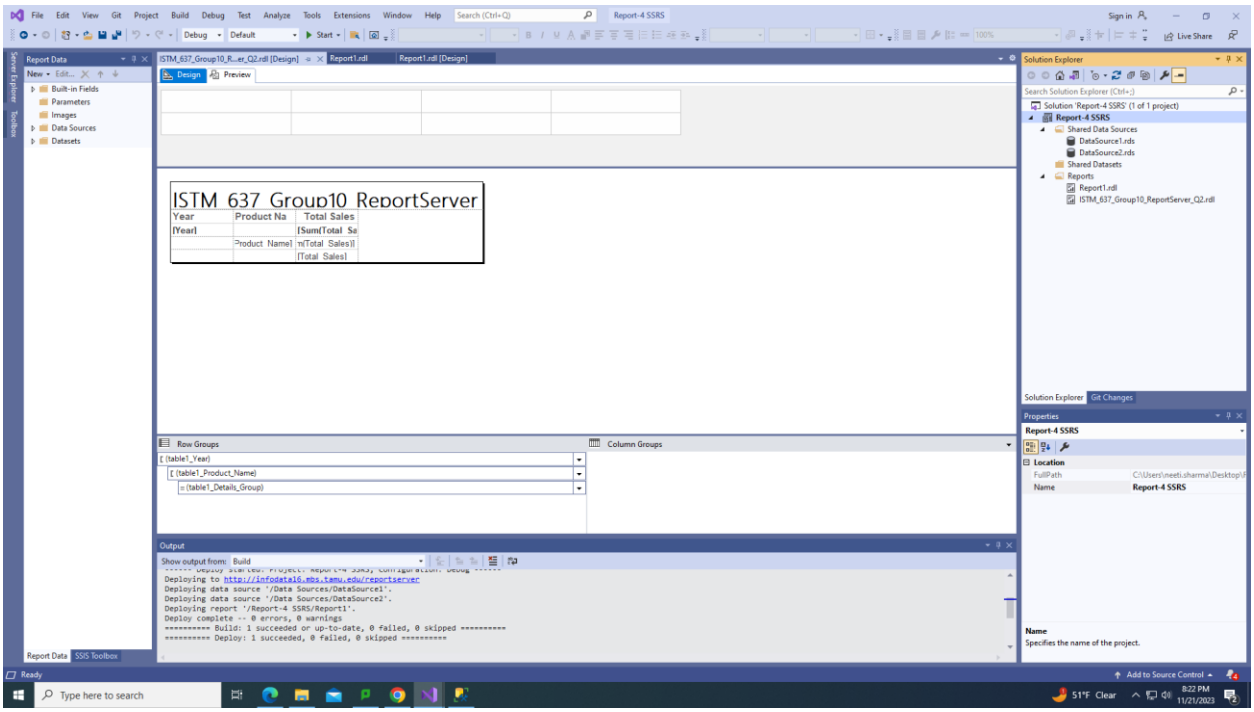


Figure: Changing the TargetServerURL and deploying the project on the server

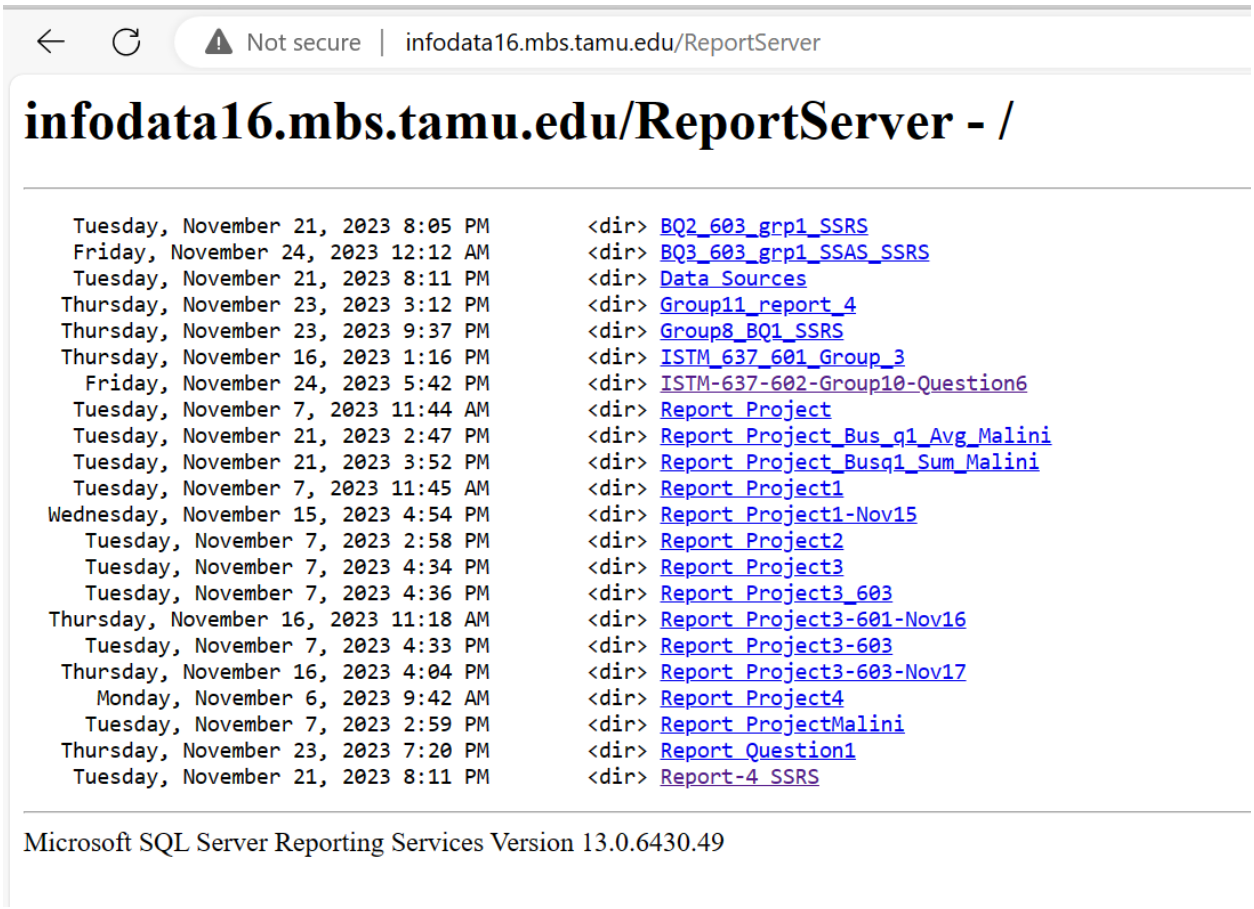


Figure: Clicking on the URL will take us to the list of reports deployed on the Report Server

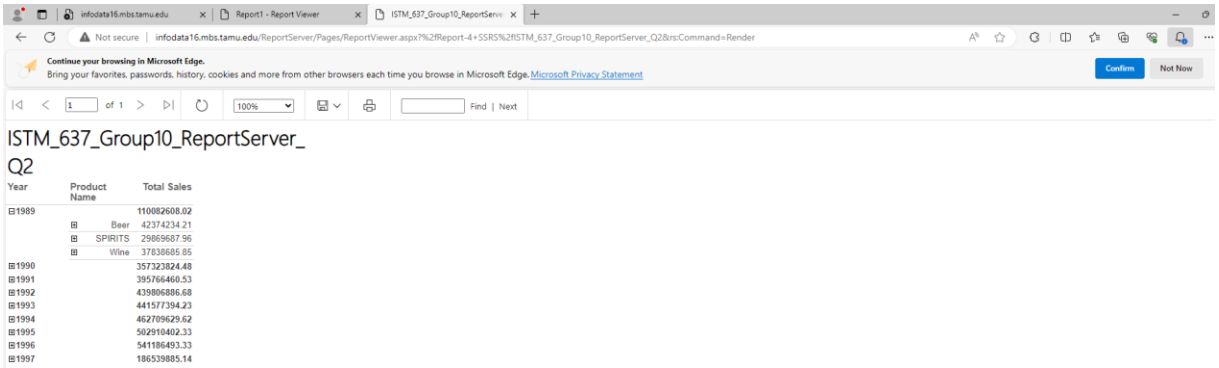


Figure: Report for Question 2 shows the total sales against each product categories of alcohol

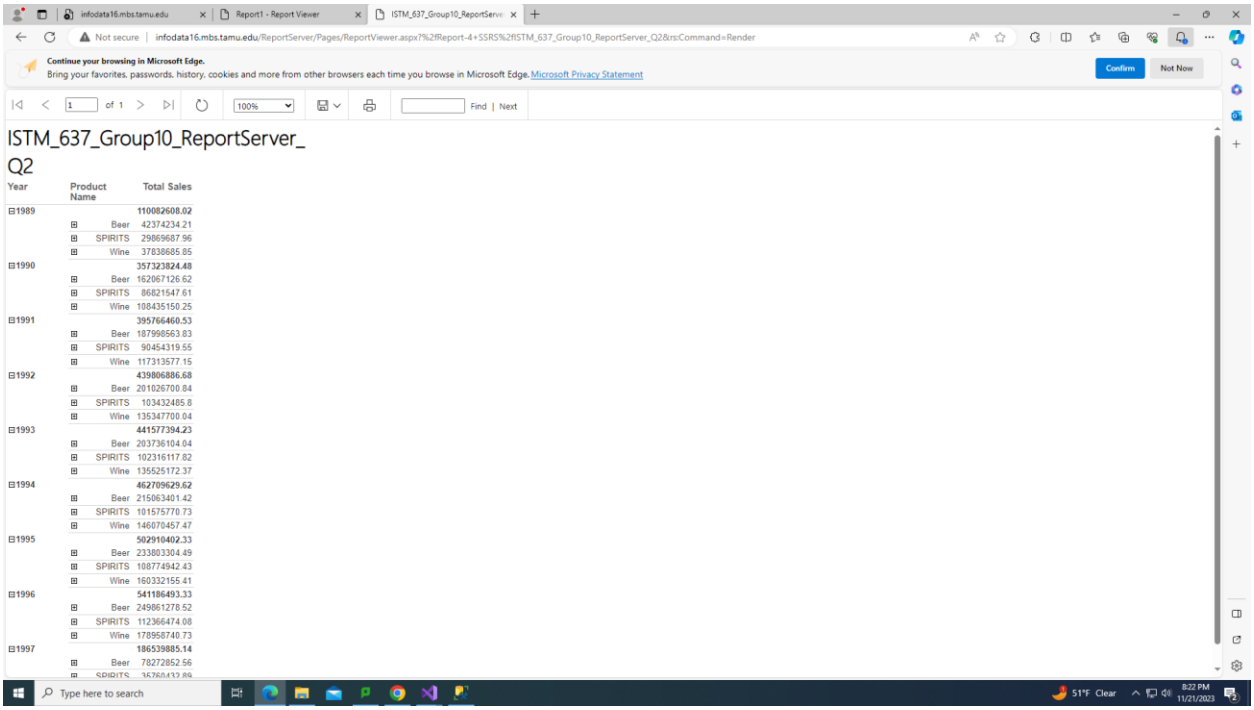


Figure: Drilling down to see different categories of alcohol and their corresponding sales across years

BQ 3 Report

What are the annual sales patterns for wine during peak seasons in the United States?

Visualization Method: Report using Microsoft Power BI

Final Report:

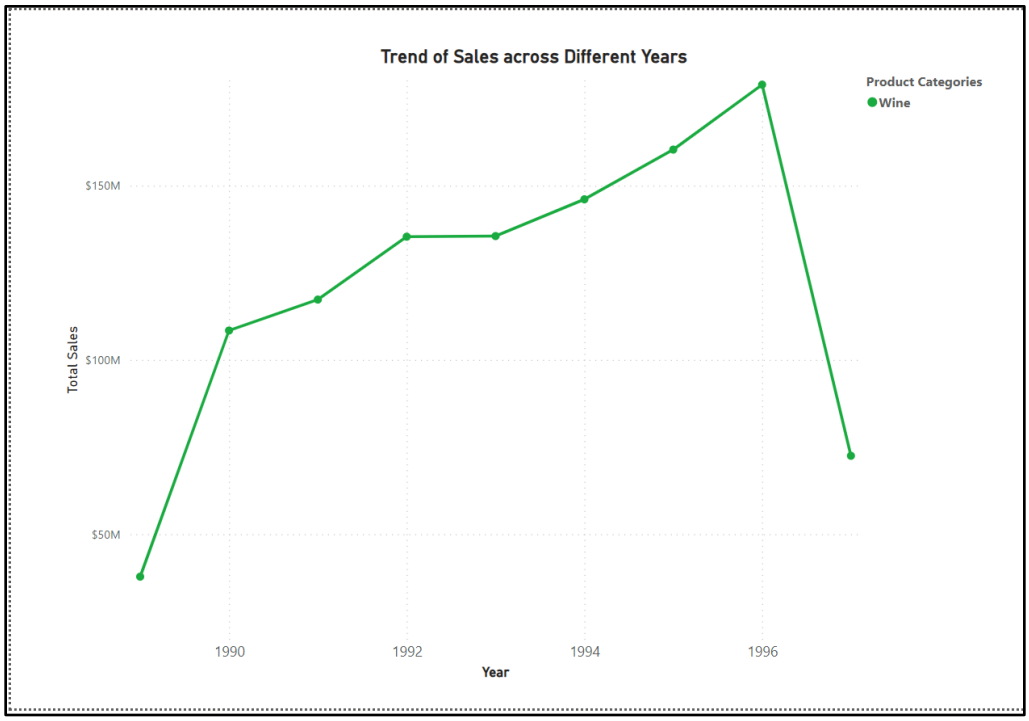


Figure: Power BI visualization showing trend of wine sales across different years

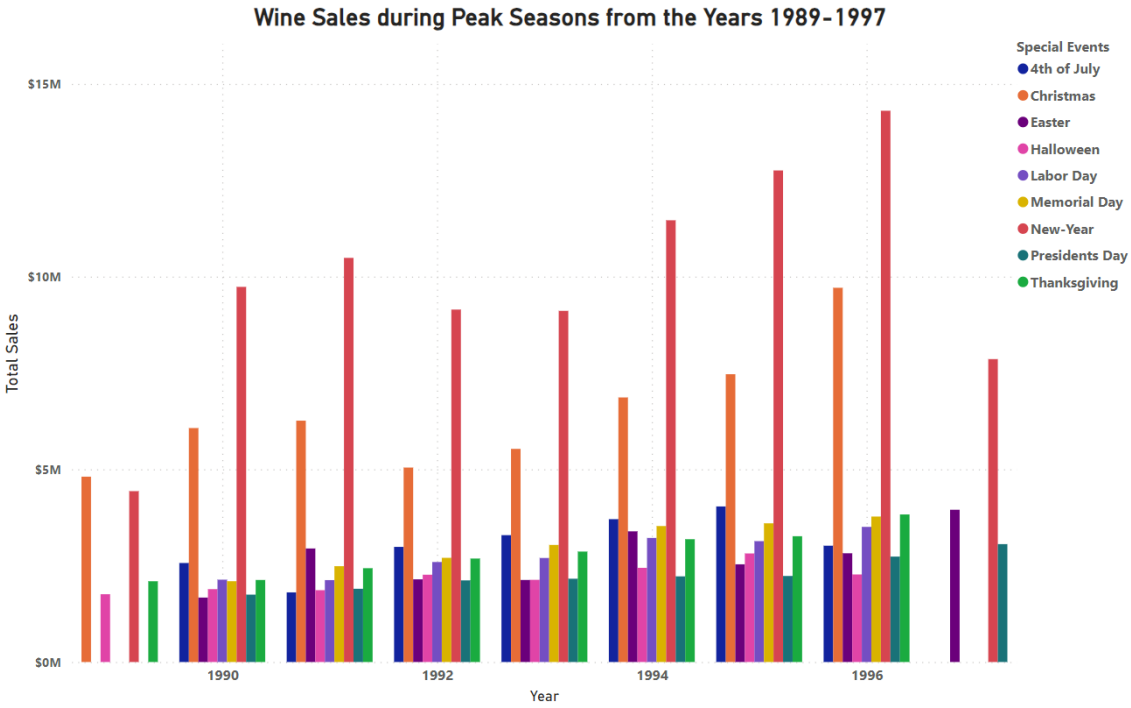


Figure: Stacked Bar chart showing the maximum sales of Wine around New Year followed by Christmas

Analysis: The Power BI analysis shows that Dominick's wine sales were highest in 1996 but dropped in 1997. This demands for a closer look into the reasons behind the decline. On a positive note, the data reveals that wine sells really well during New Year and Christmas. This suggests that Dominick could benefit from planning marketing events or launching new wines around these festive times. The analysis also highlights the combined holiday sales in 1996 stand out as the company's highest. In a nutshell, the insights offer practical ways for Dominick to adjust its strategies, making the most of high-sales periods and identifying the potential challenges.

The analysis goes beyond just numbers, offering practical guidance for Dominick to optimize its approach by understanding consumer behavior and timing marketing initiatives effectively.

Implementation: The analysis was conducted using Microsoft Power BI, utilizing data from the Sales_Data_Mart. The data was imported from the SQL Server Database and the tables Date_Dim, Store_Dim, Product_Dim and Fact_Sales were selected.

The Y-axis denotes Total Sales, filtered exclusively for the "Wine" product, while the X-axis represents the chronological Year. The visualization includes a legend showing the impact of Special Events on Wine sales across years.

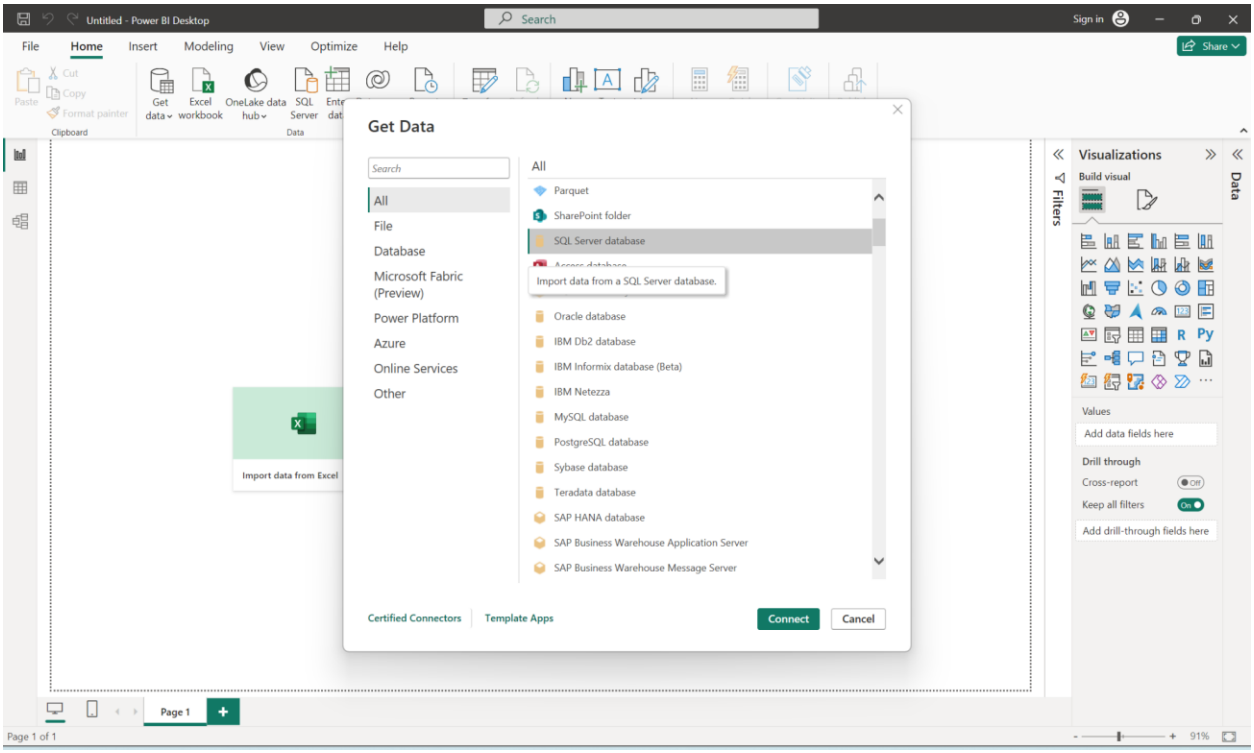


Figure: Importing data from SQL Server Database

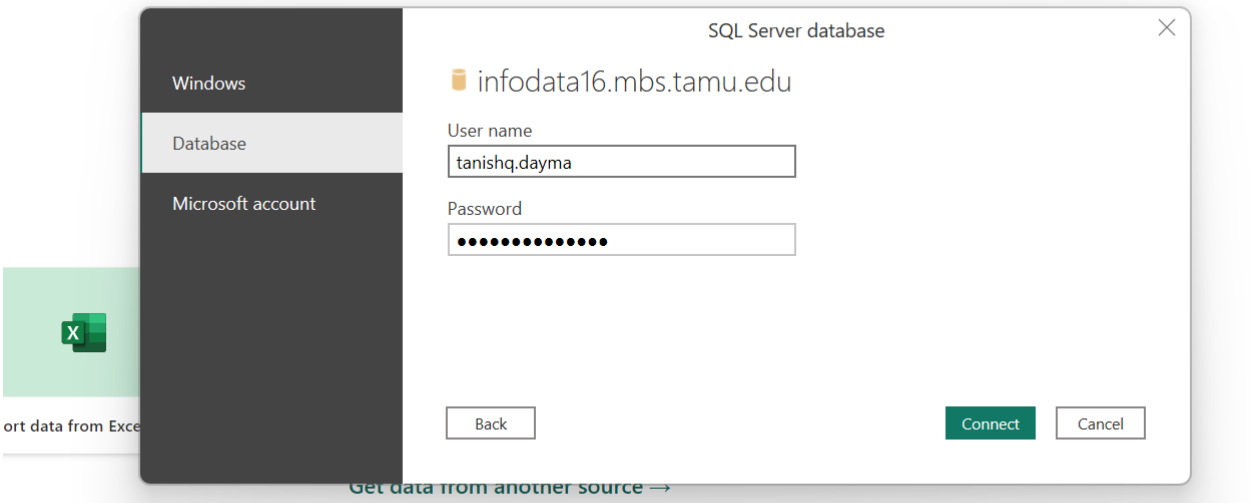


Figure: Establishing connection to the SQL Server

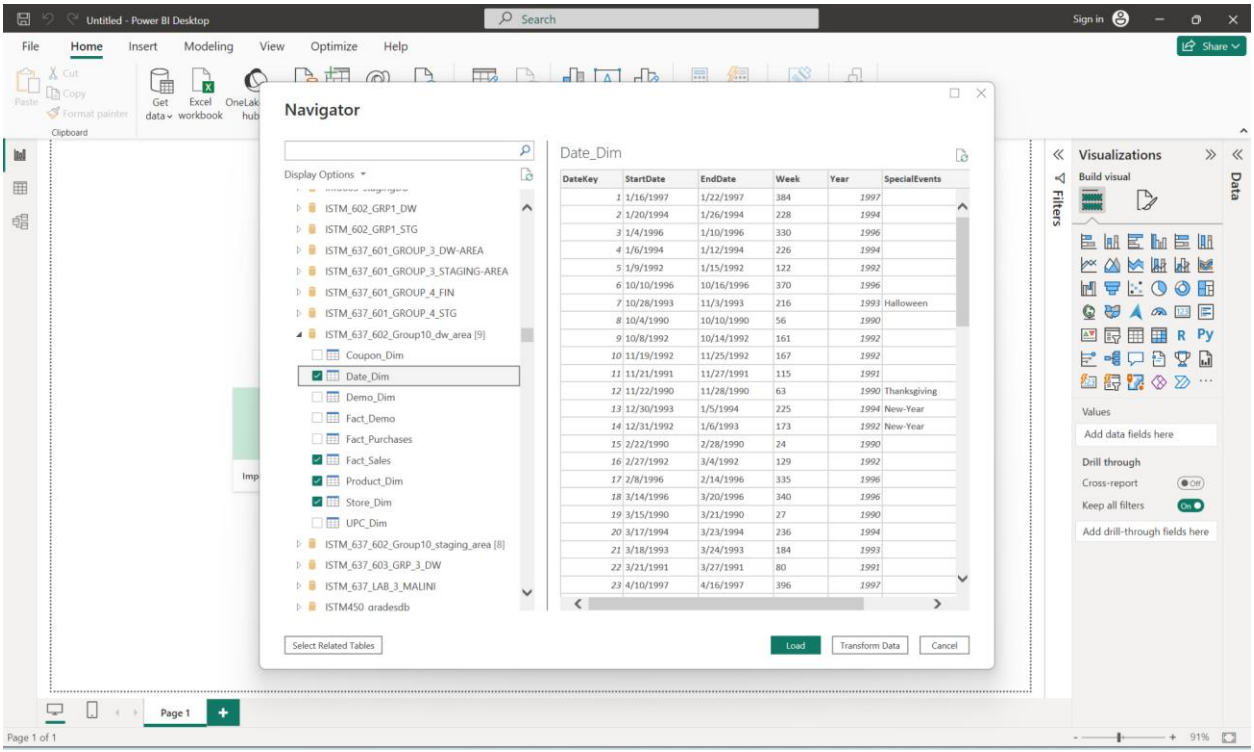


Figure: Selecting Sales_Date_Mart tables for visualizatio

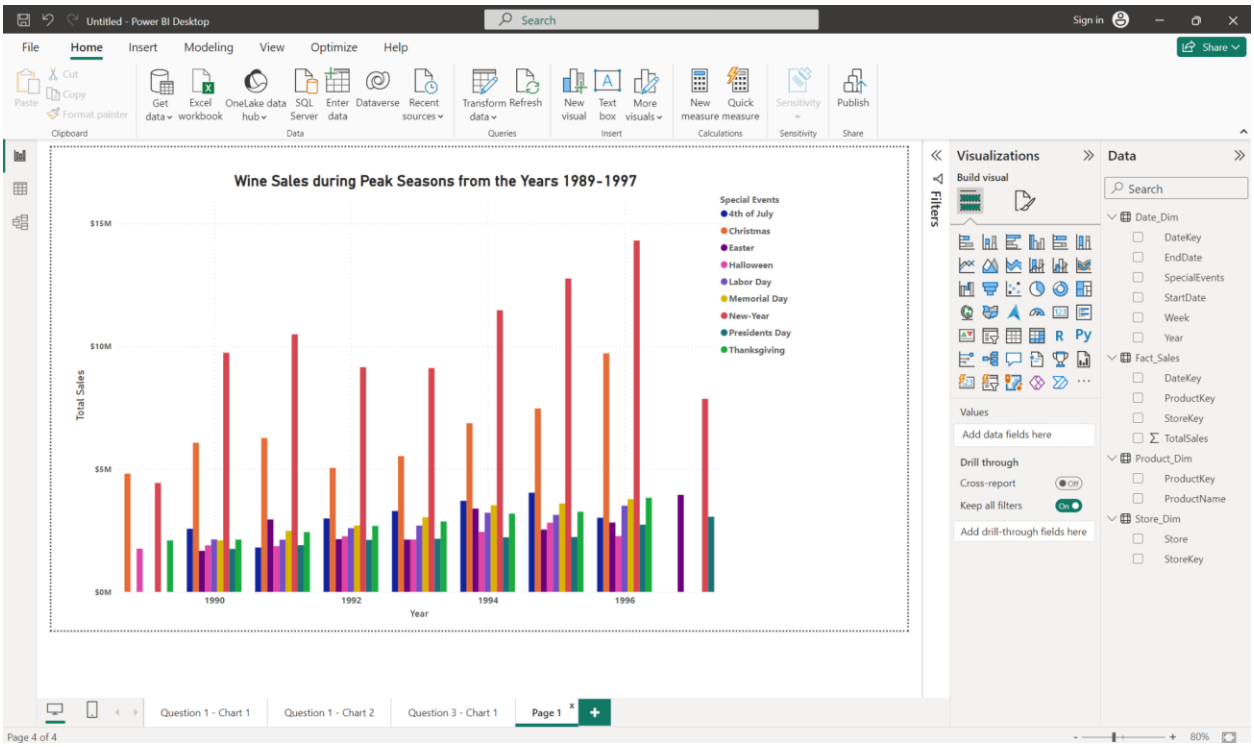


Figure: Stacked Bar chart showing the maximum sales of Wine around New Year followed by Christmas

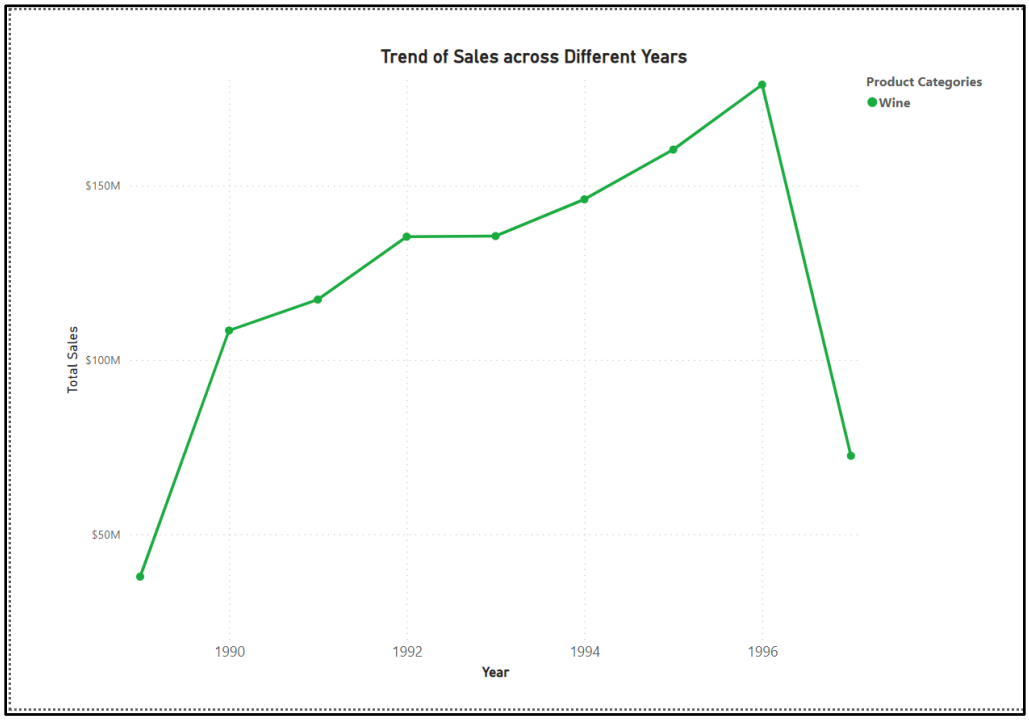


Figure: Power BI visualization showing trend of wine sales across different years

BQ 4 Report

How does the percentage of avid, hurried, and strange shoppers vary across different cities, and how can we tailor marketing strategies accordingly?

Visualization Method: Report from multidimensional cubes using SSAS

Final Report:

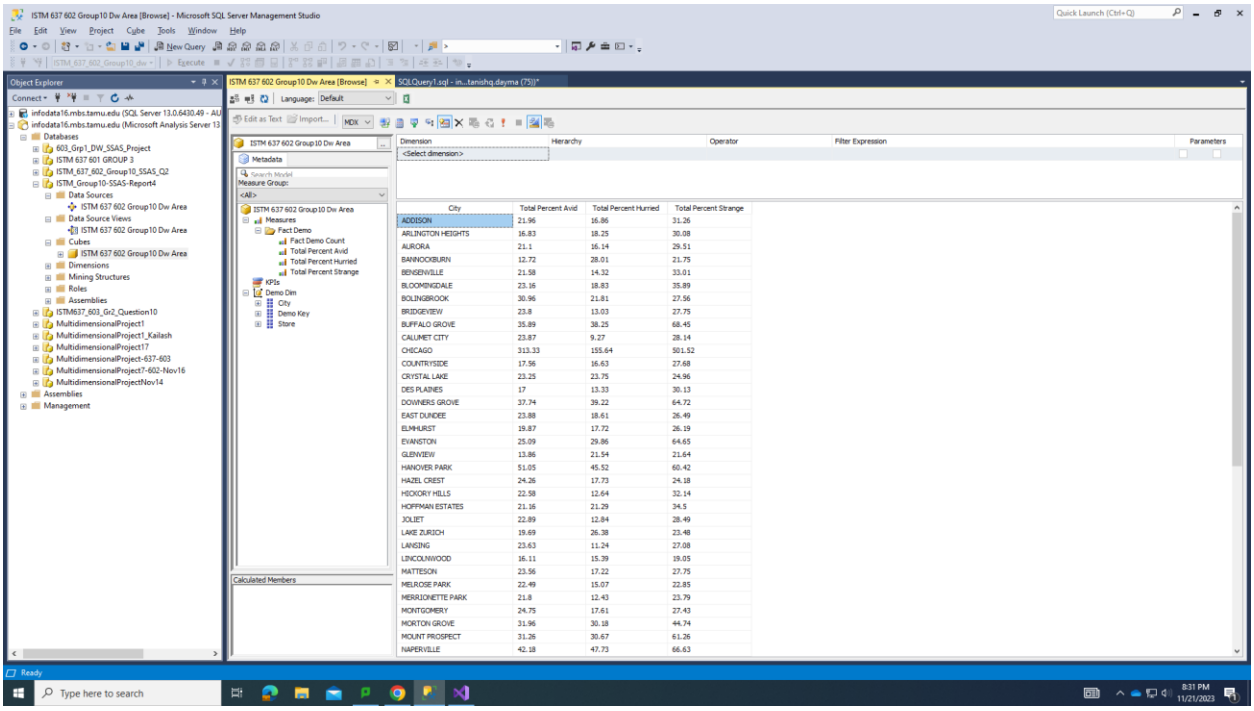


Figure: SSAS report showing population of shoppers in different cities

Analysis: The SSAS report reveals that Chicago has the largest population of avid, hurried, and strange shoppers. Identifying this allows DFF to strategically align its tactics with the distinct consumer preferences prevalent in Chicago's local stores. This insight can help DFF to target its marketing efforts more effectively by tailoring advertisements and product offerings to cater to the identified shopper profiles. Notably, among the various shopper categories, Chicago stands out with the highest population of avid shoppers, indicating a robust demand for shopping activities in the city. While this presents a significant opportunity for DFF to expand and thrive, it's crucial to assess the competitive landscape, as a high number of avid shoppers may intensify competition among retailers in Chicago.

Implementation: The analysis was performed through SSAS, utilizing information sourced from the Demo_Data_Mart. The source data was imported from the ISTM_637_602_Group10_dw_area and the tables Demo_Dim and Fact_Demo were selected.

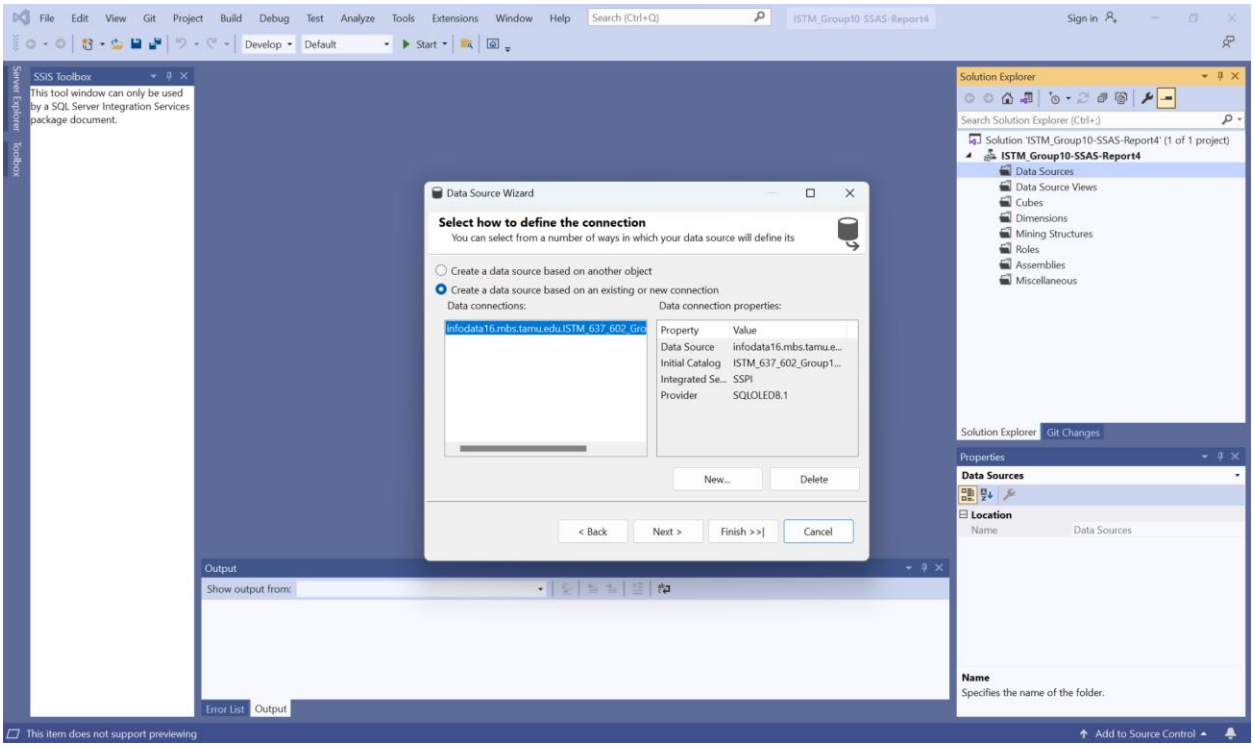


Figure: Establishing datasource connection

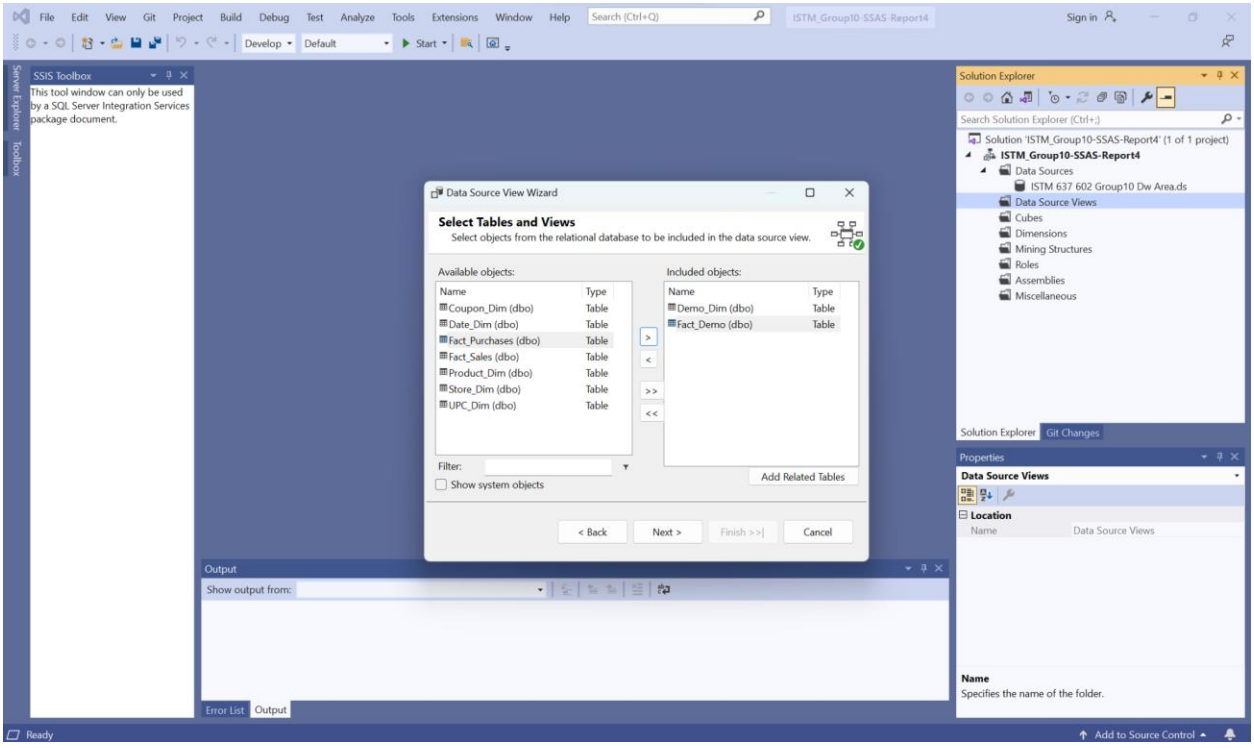


Figure: Mapping dimension and fact tables to the data source view

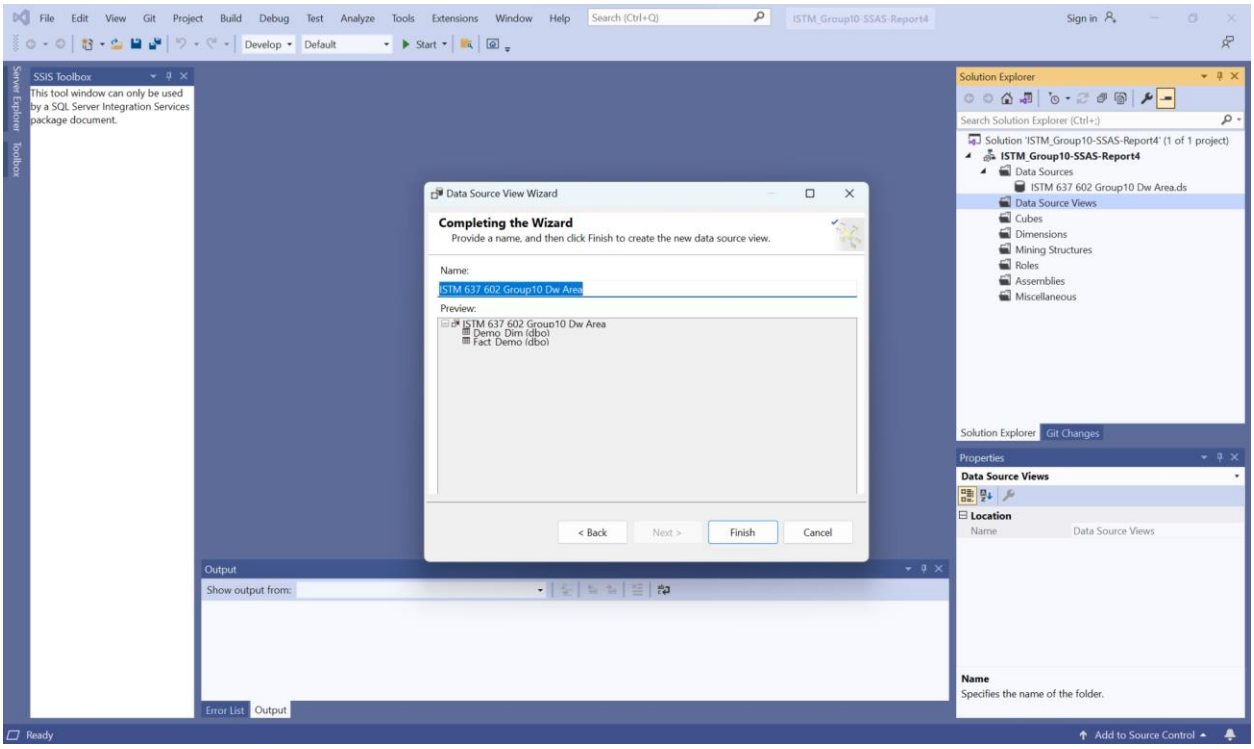


Figure: Finalizing the data source view

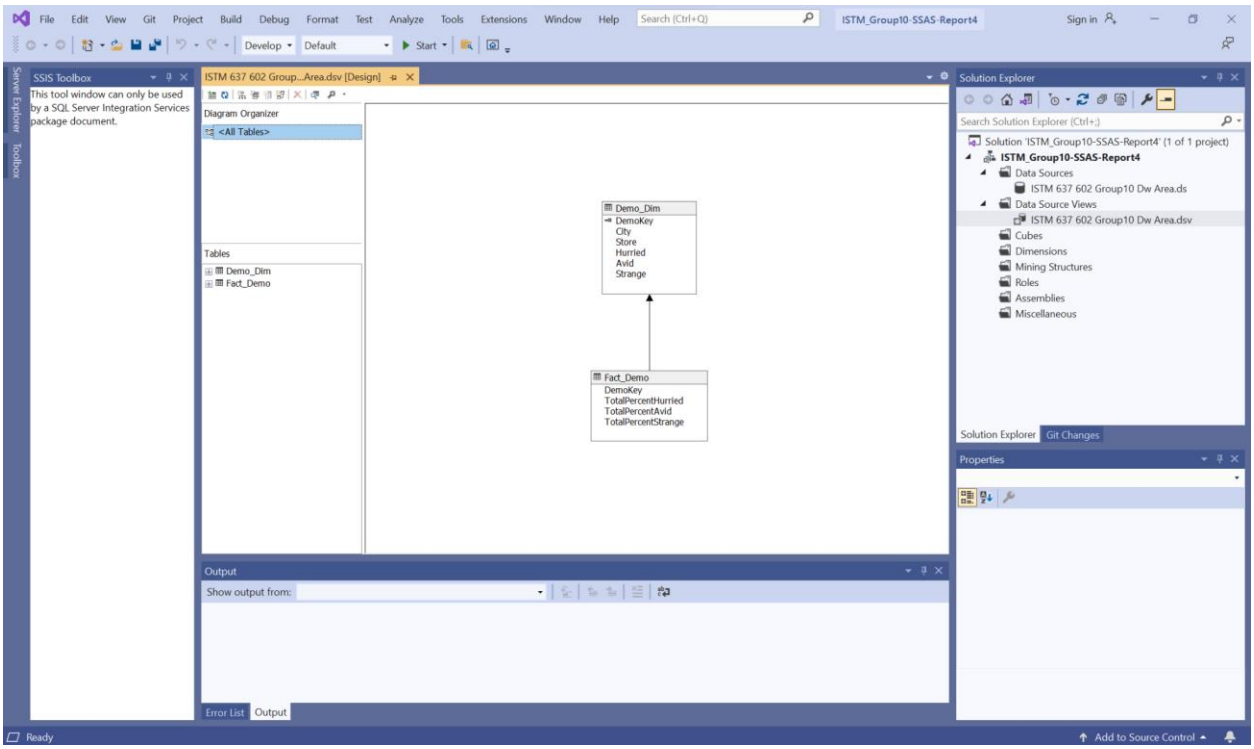


Figure: Data source view for Demo_Data_Mart

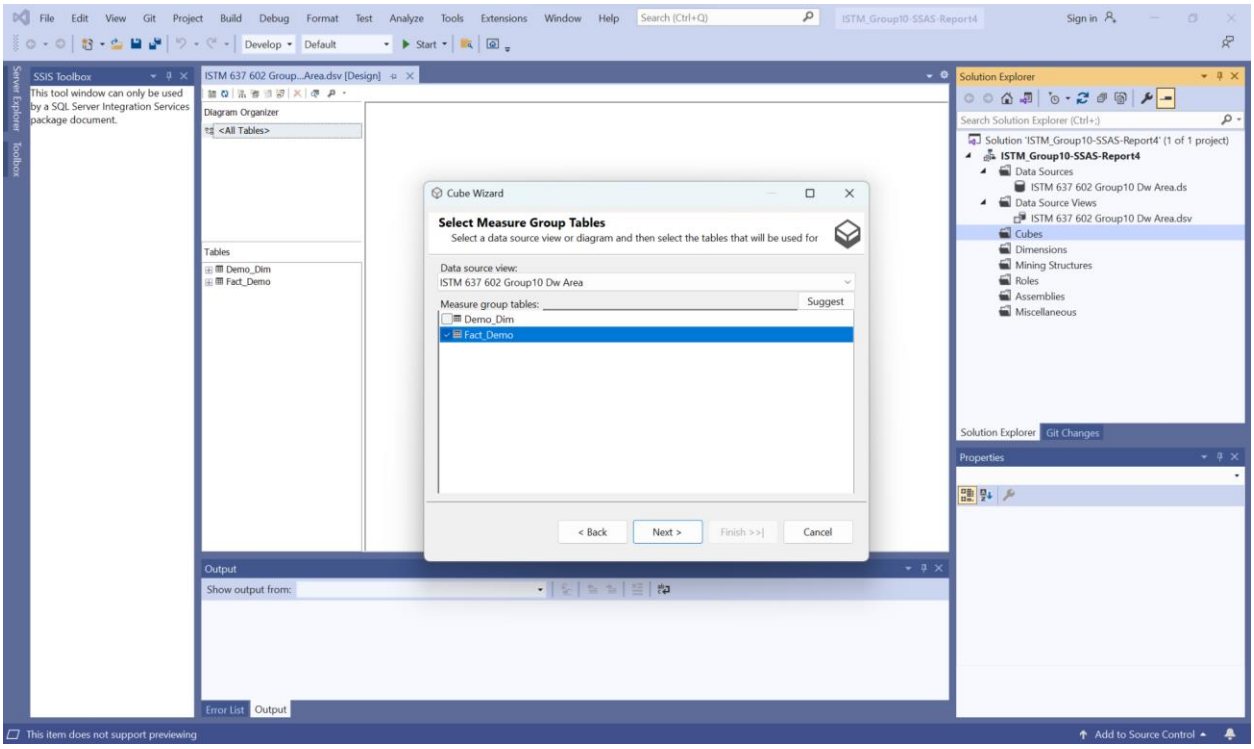


Figure: Initializing the cube structure

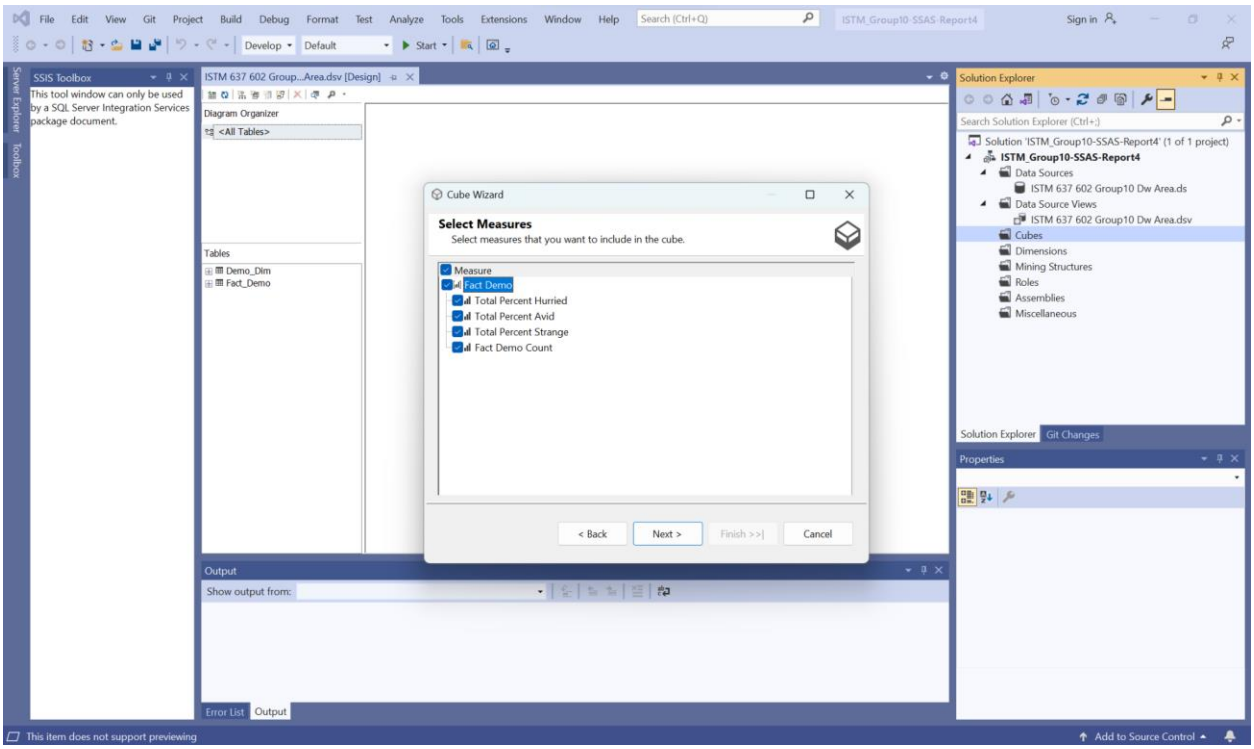


Figure: Selecting fact table attributes for cube structure

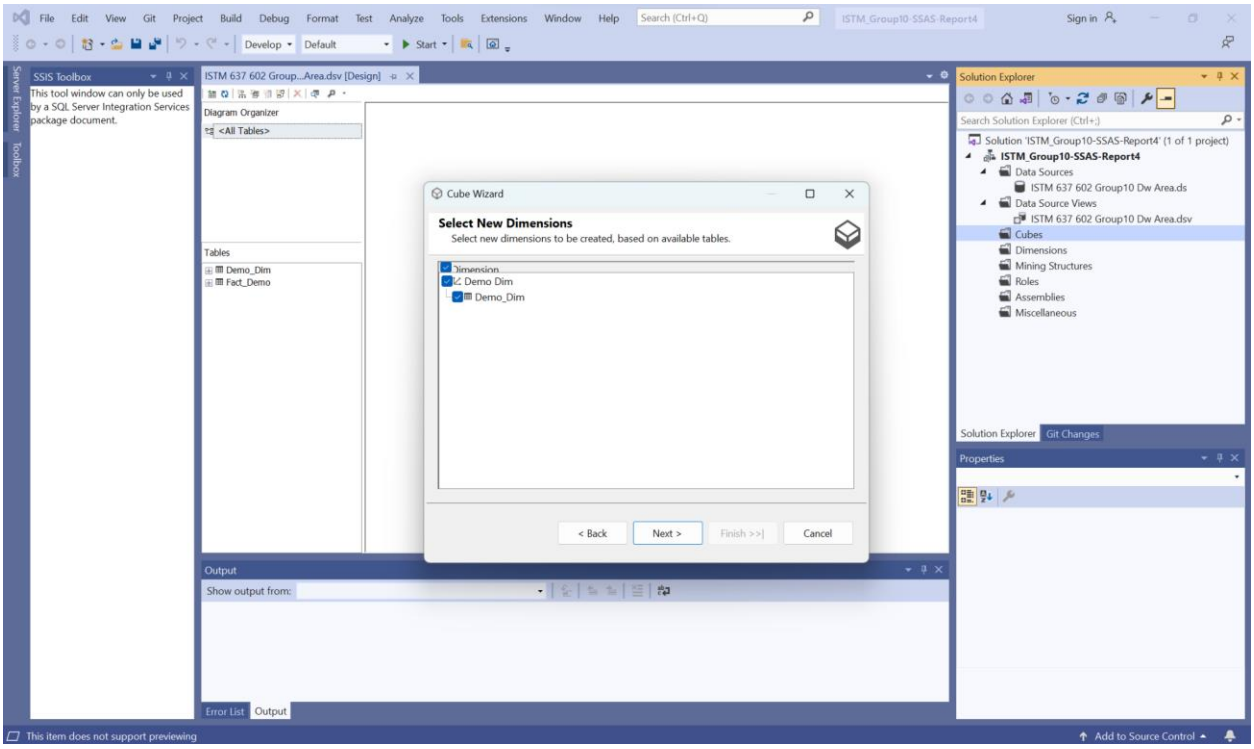


Figure: Selecting dimension table attributes for cube structure

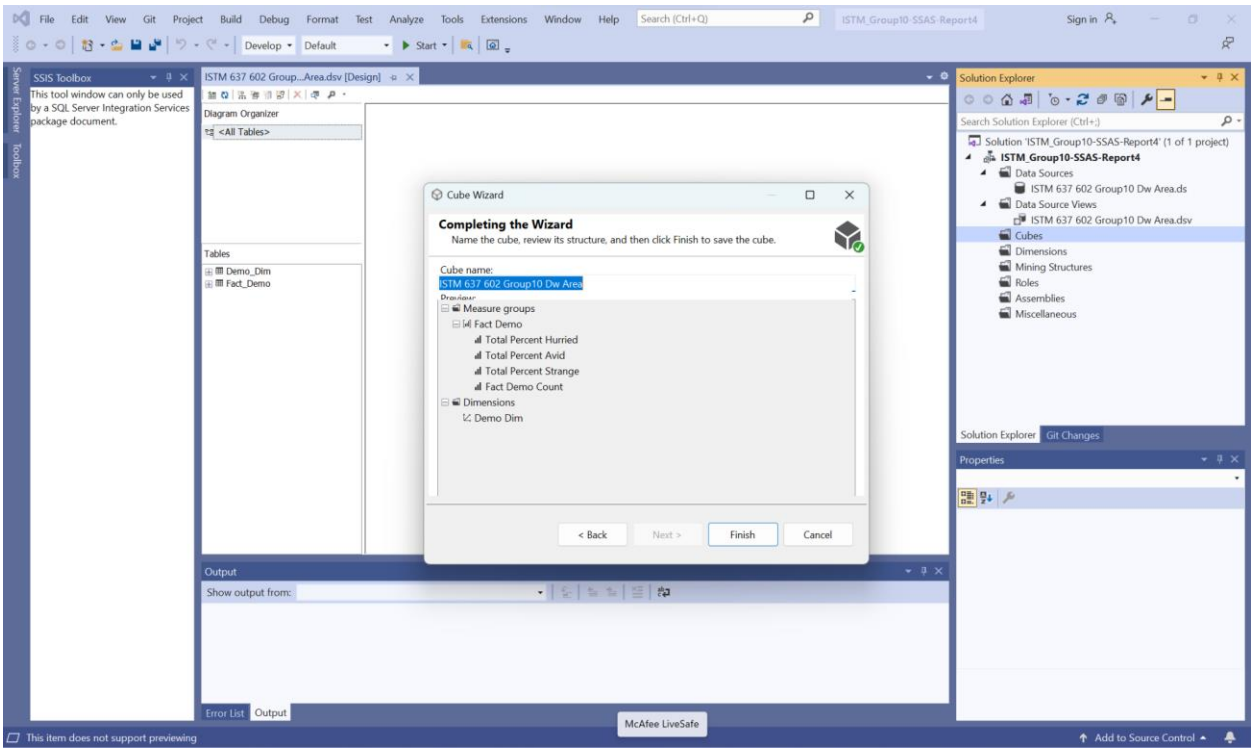


Figure : Finalizing creation of cube structure

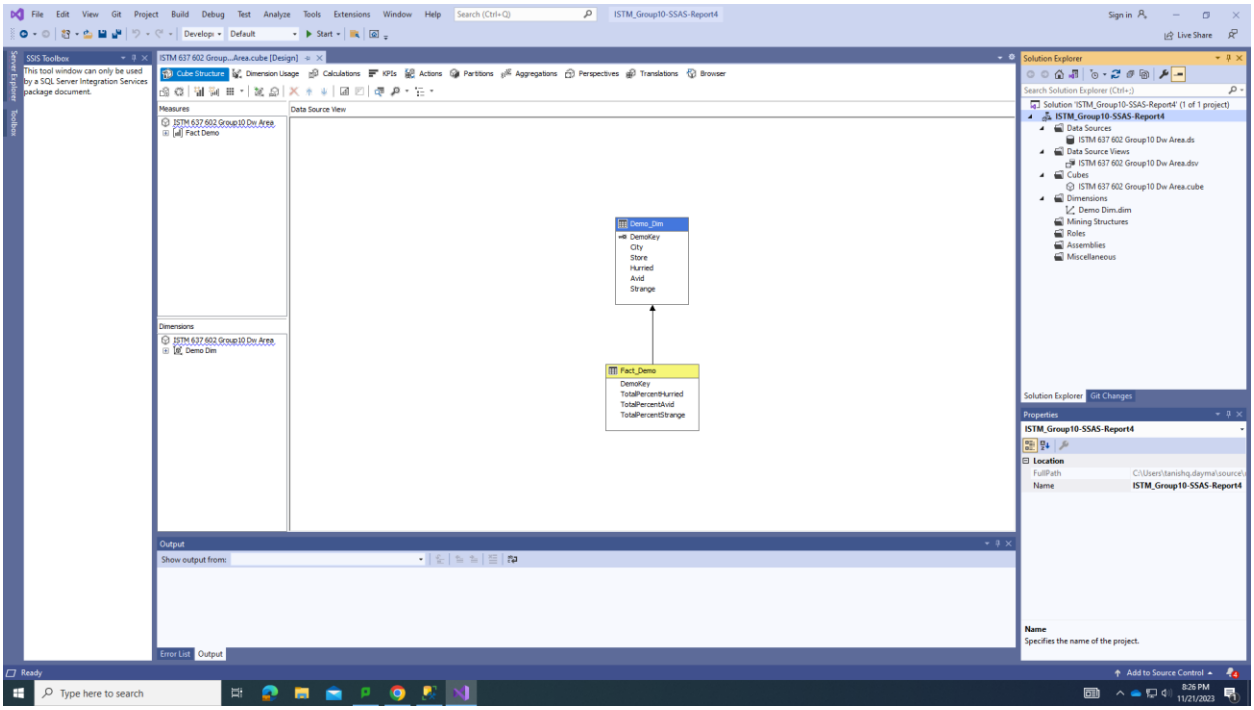


Figure: Cube structure for Demo_Data_Mart

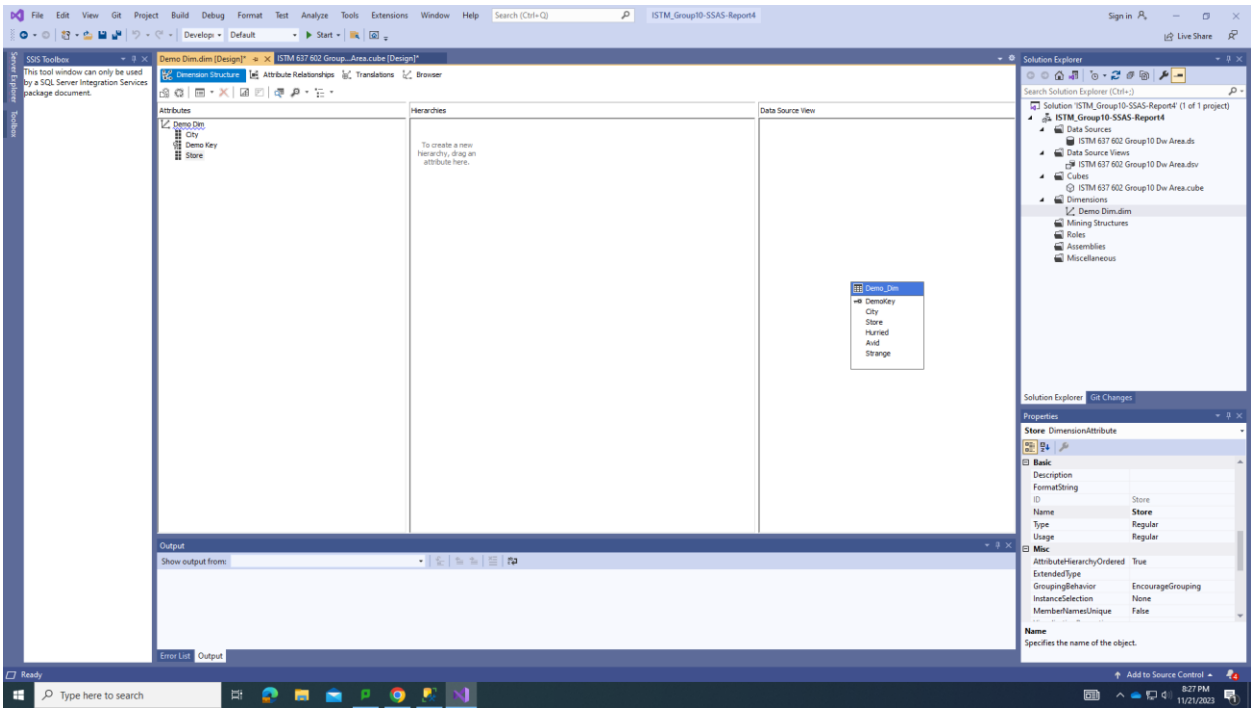


Figure: Selecting Dim table attributes in the cube structure

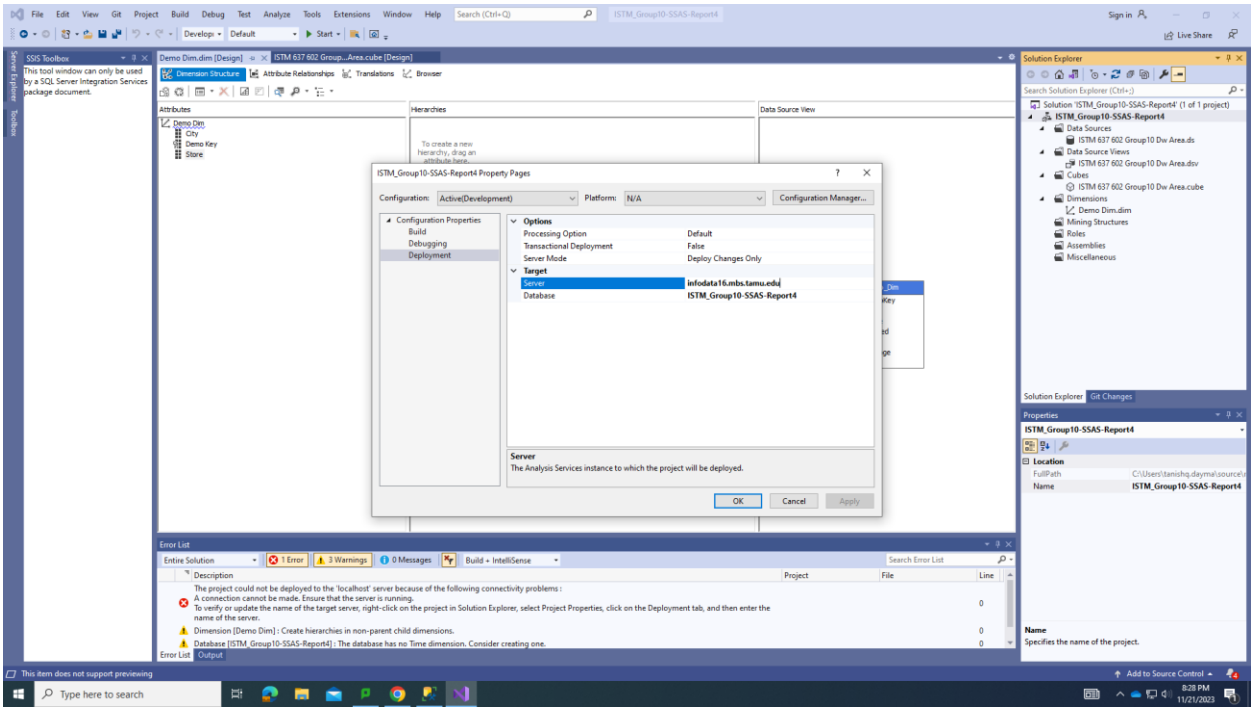


Figure: Setting properties to deploy the project to the infodata16.mbs.tamu.edu server

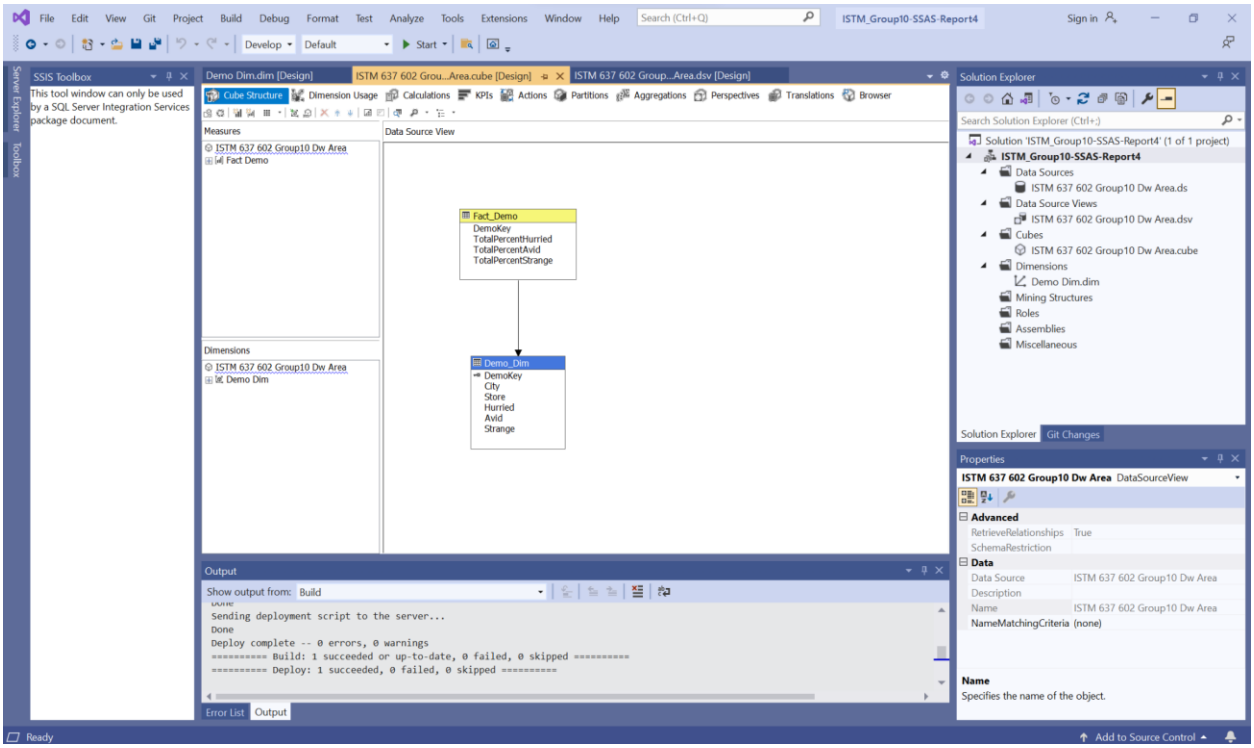


Figure: Deploying cube structure

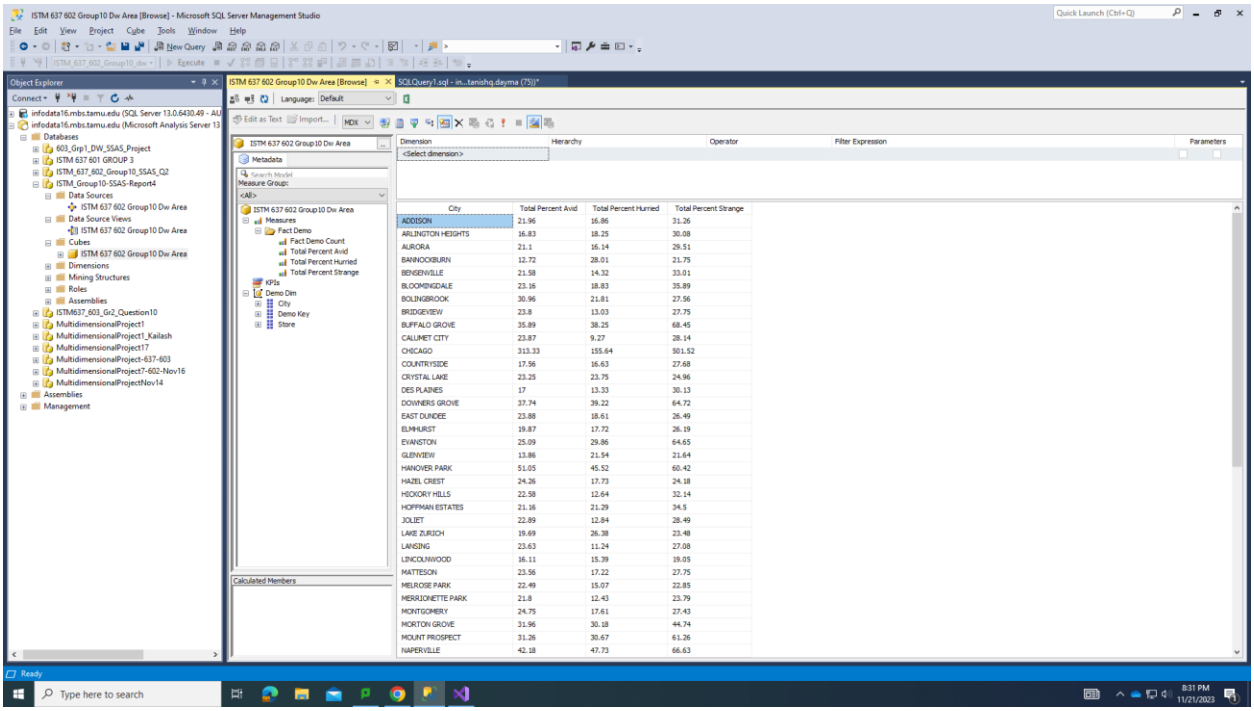


Figure: SSAS report showing population of shoppers in different cities

BQ 5 Report

What is the most popular coupon used for purchasing the top 20 juice categories?

Visualization Method: Report from SSRS

Final Report:

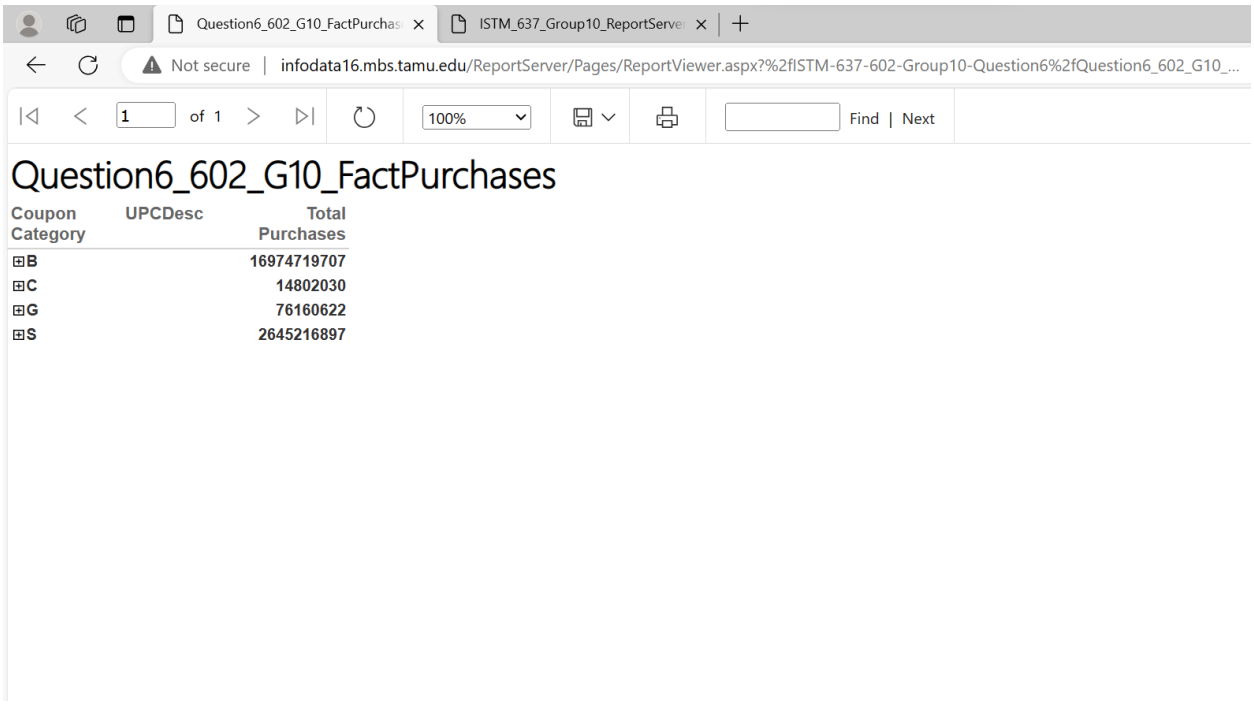


Figure: SSRS table output to understand which Coupon Category is the most popular

Analysis: SSRS can be leveraged to generate a detailed report on total purchases for each coupon category within the top 20 juice categories. It provides valuable insights for any business strategy, as SSRS's intuitive design can be used to organize and analyze data, grouping and sorting based on coupon categories and total purchases. This not only helps us identify the most popular coupons but also allows for dynamic adjustments like focusing on the top 20 juice categories.

The identification of a specific popular coupon category is beneficial, as it will allow the business team to focus on coupons/promotions that resonate the most with customers, maximizing their return on investment.

Implementation: The analysis was performed through SSRS, utilizing information sourced from the Coupon_Purchases_Data_Mart. The source data was imported from the ISTM_637_602_Group10_dw_area and the tables UPC_Dim, Coupon_Dim and Fact_Purchases were selected.

Implemented tabular reporting in SSRS to analyze trends by first grouping data based on coupon categories and then further refining the analysis by specific brand names (UPC descriptions) to obtain total purchases.

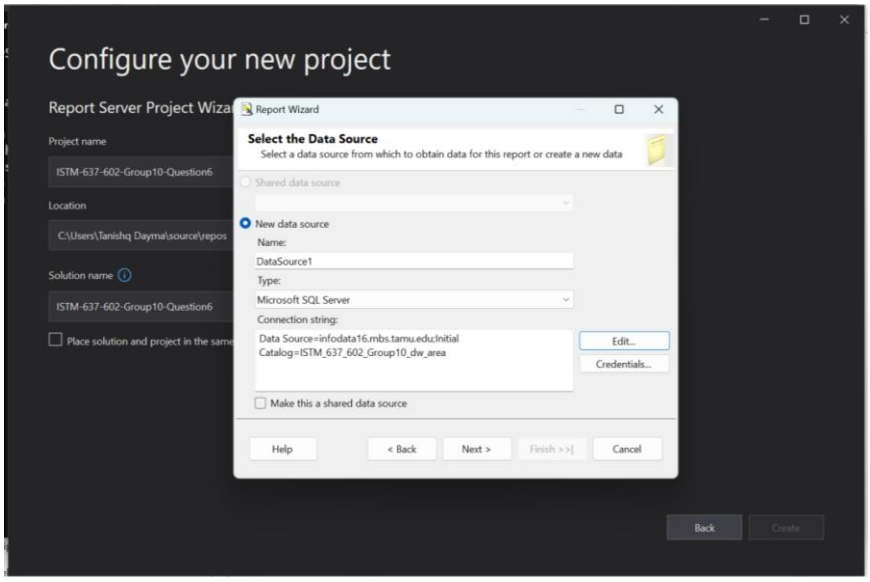


Figure: Creating a new Report Server project in Visual Studio Code

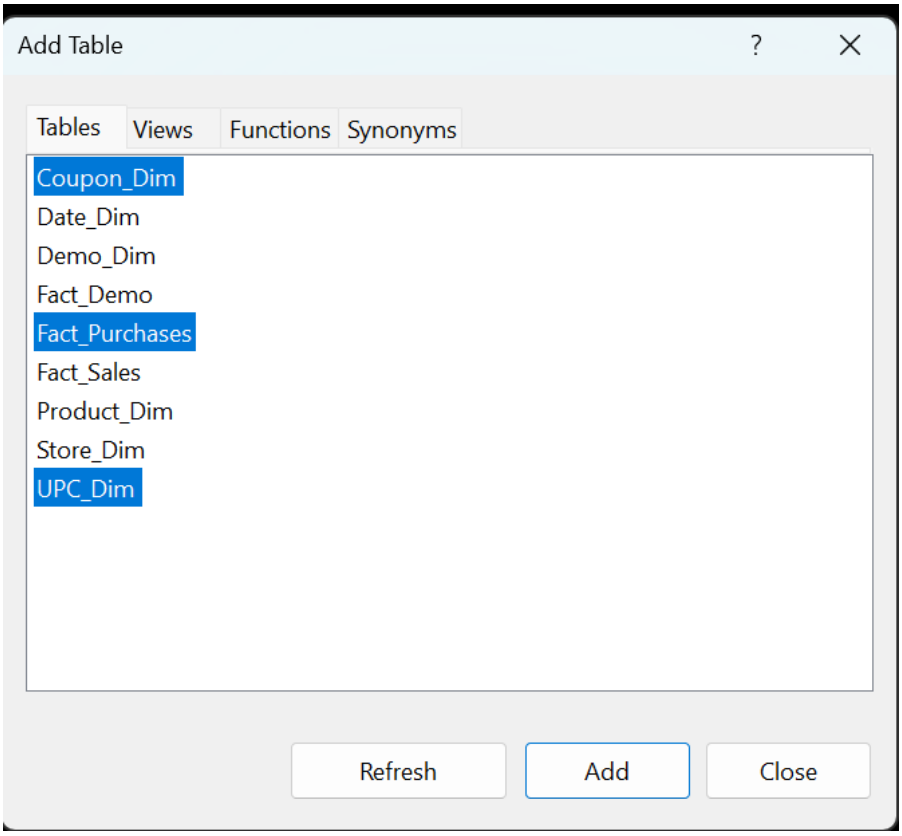


Figure: Selecting the relevant table from the data warehouse database

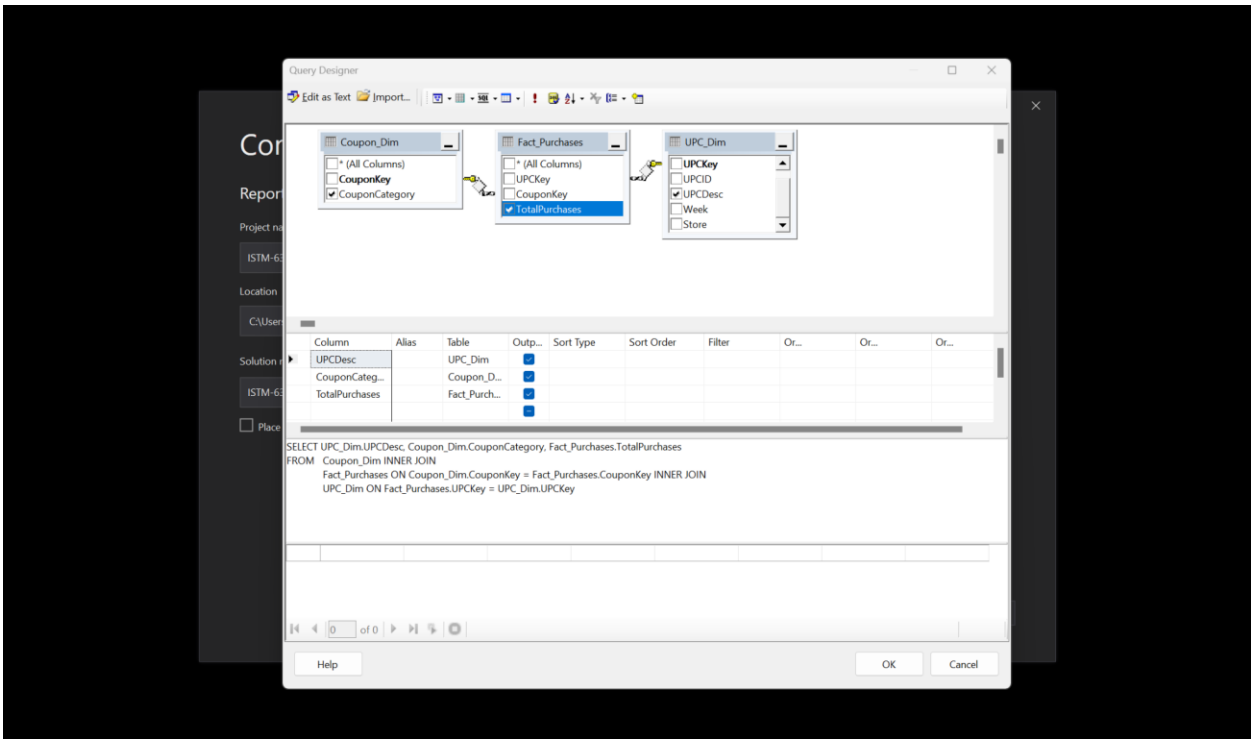


Figure: Selecting the relevant fields for analysis

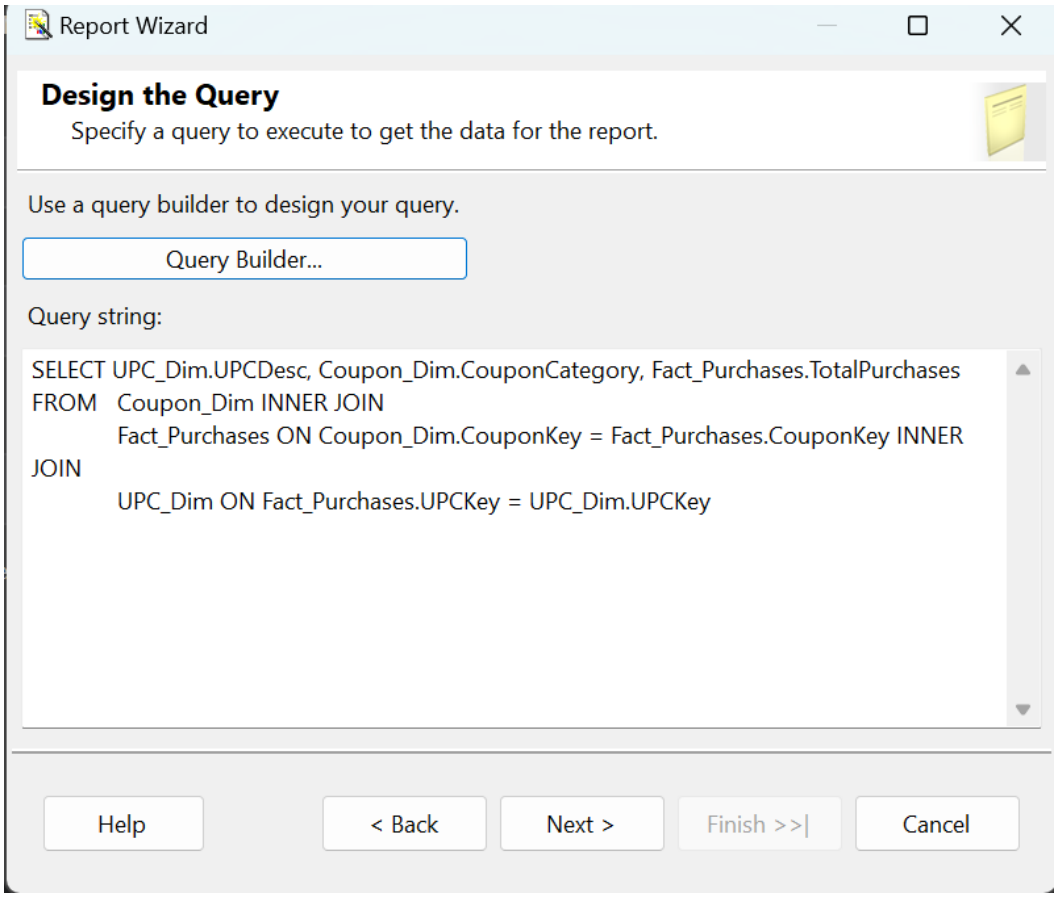


Figure: Query Builder for the Report Server

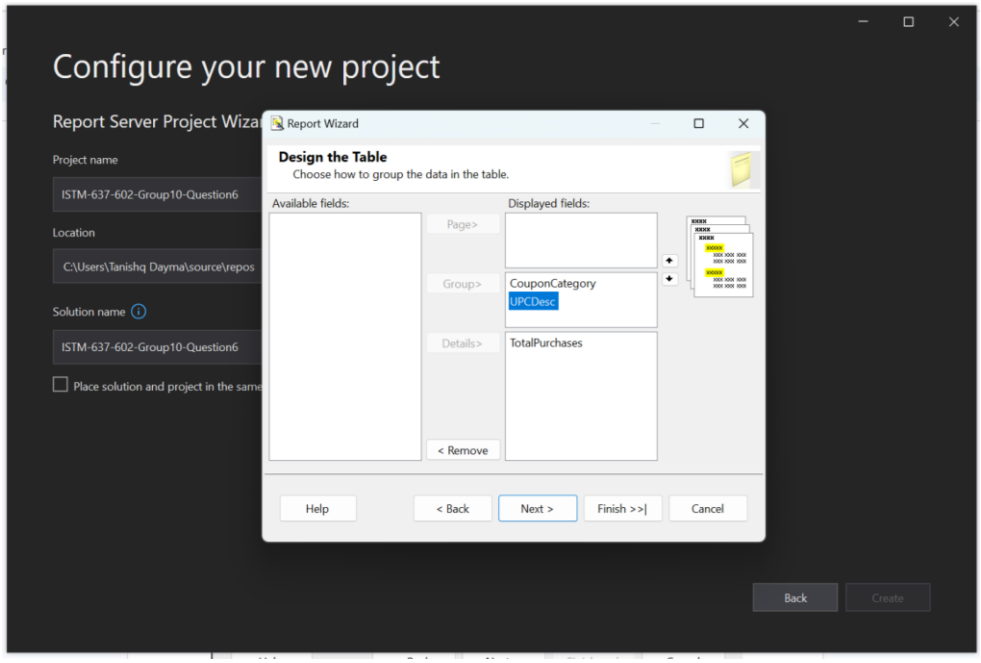


Figure: Designing the table by placing the juice brands within different coupon categories

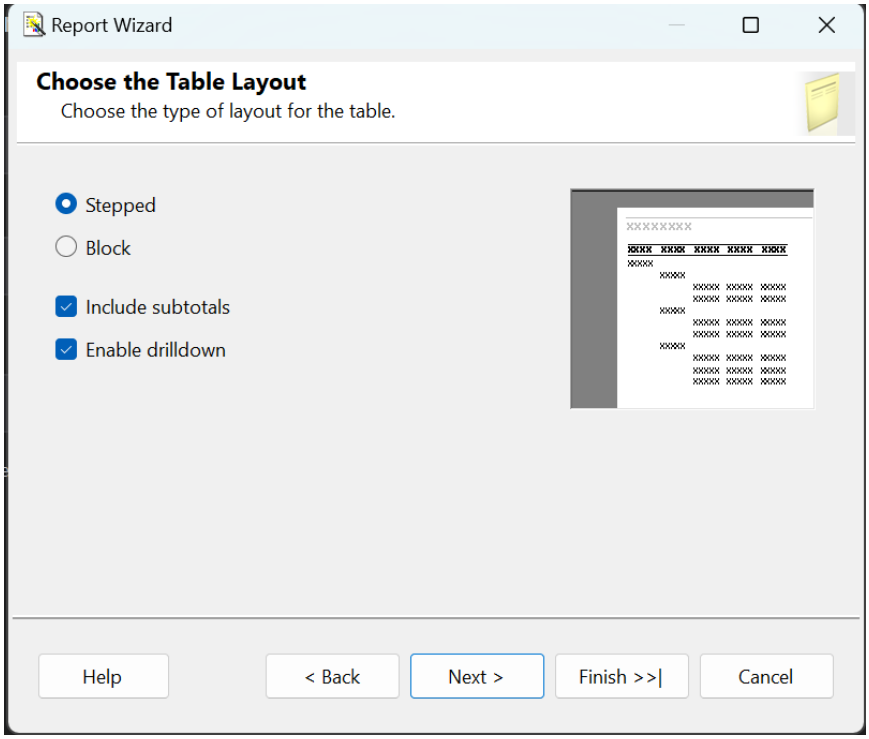


Figure: Choosing the table layout as 'Stepped' including subtotals and drilldowns

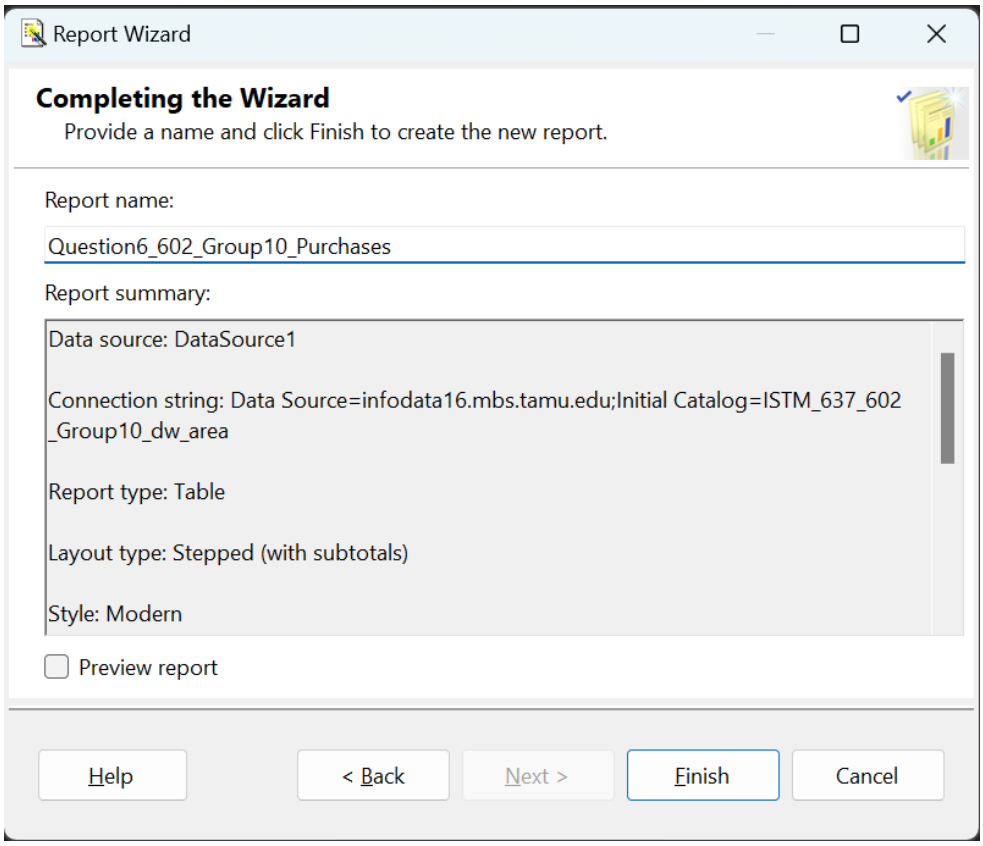


Figure: Giving the report a unique name and clicking on 'Finish'

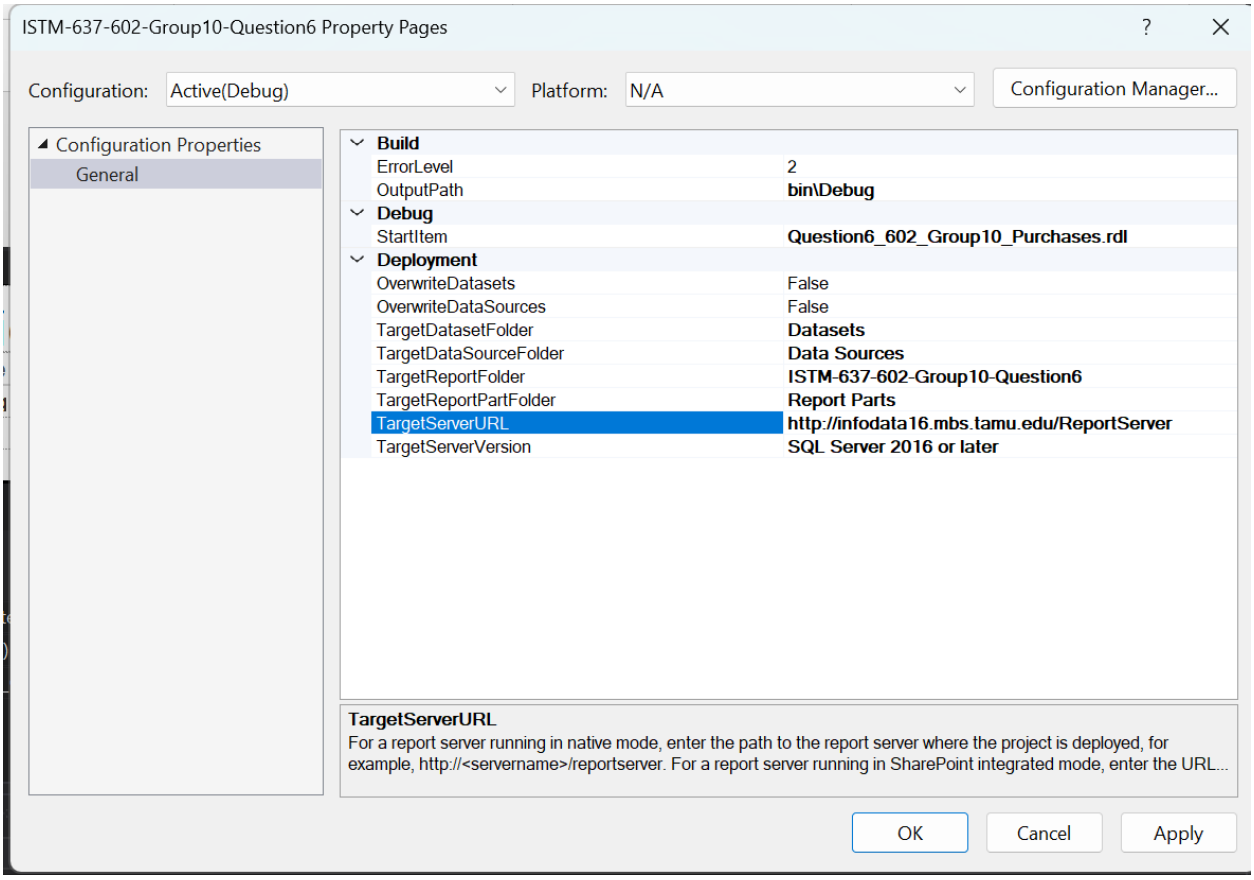


Figure: Clicking on Properties of the report and changing the TargetServerURL to our server

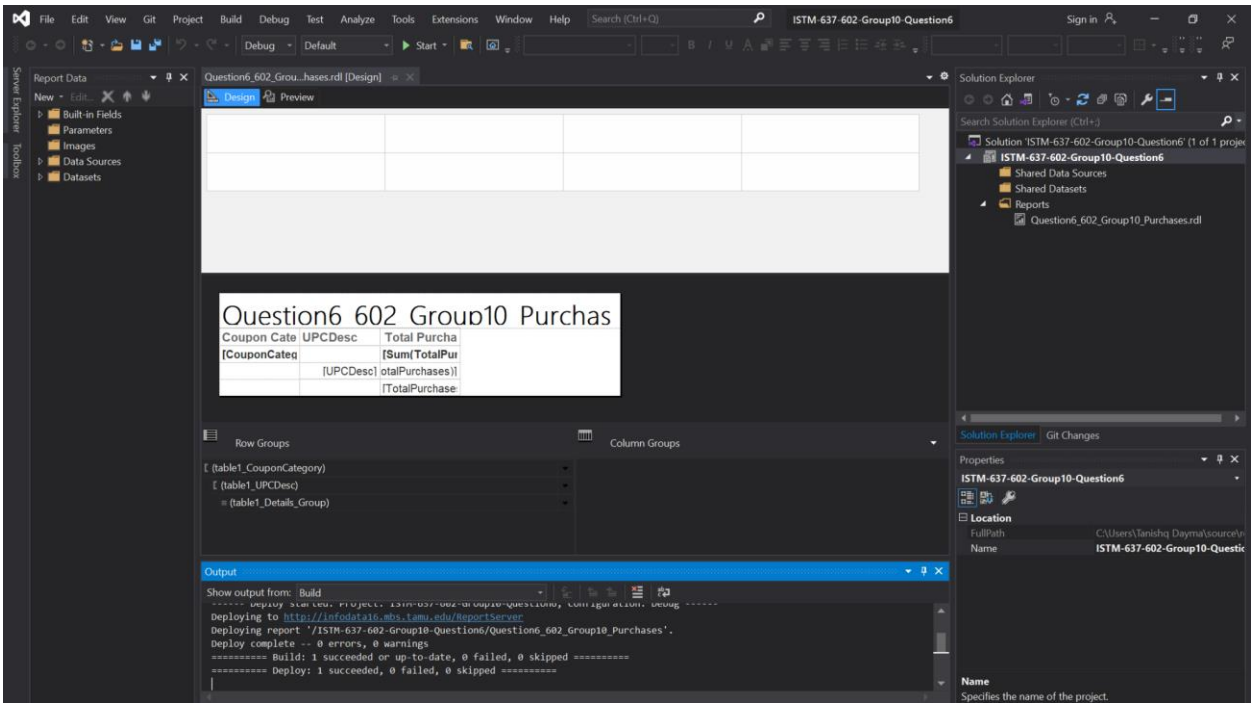


Figure: Deployment completed successfully

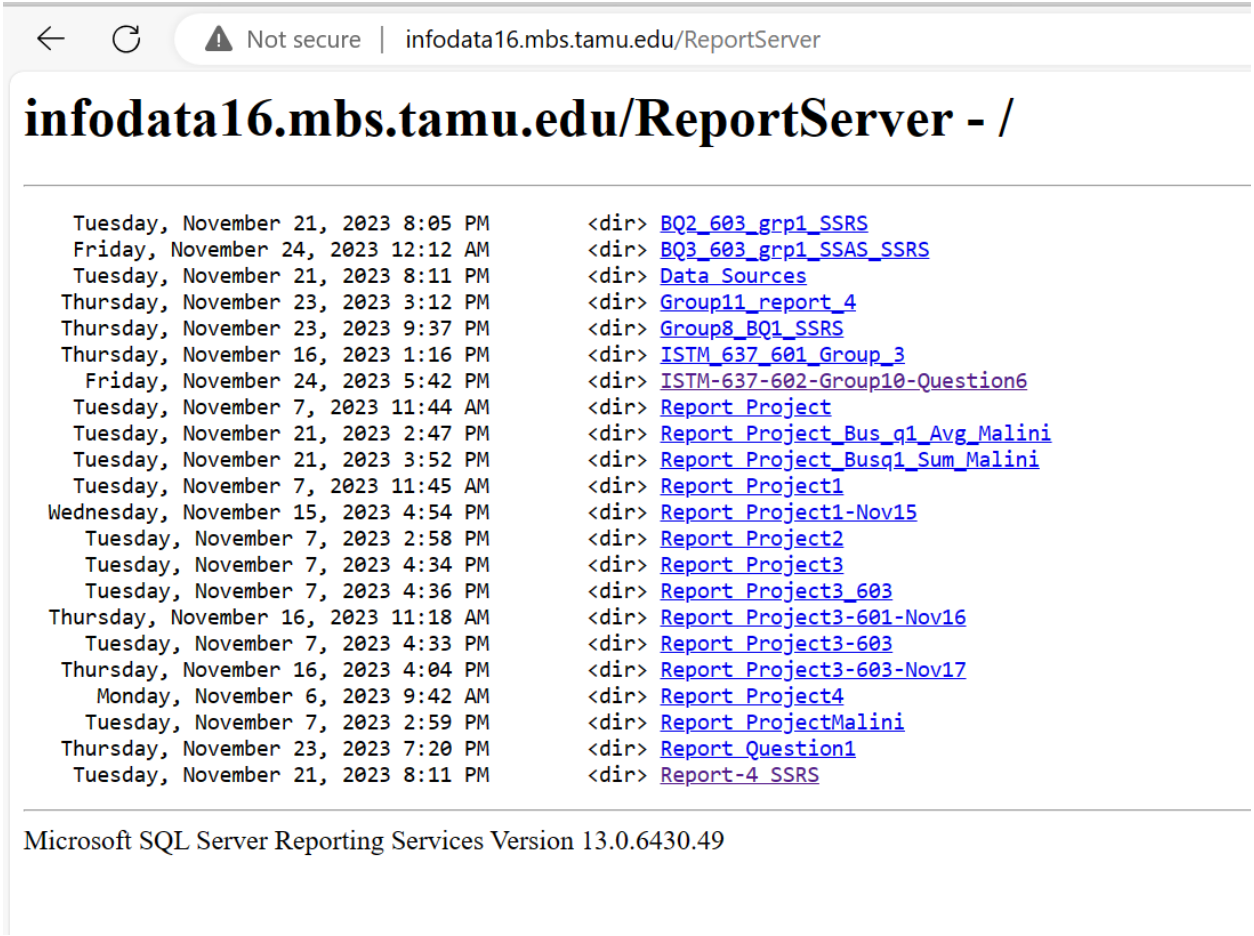


Figure: Clicking on the URL will take us to the list of reports deployed on the Report Server

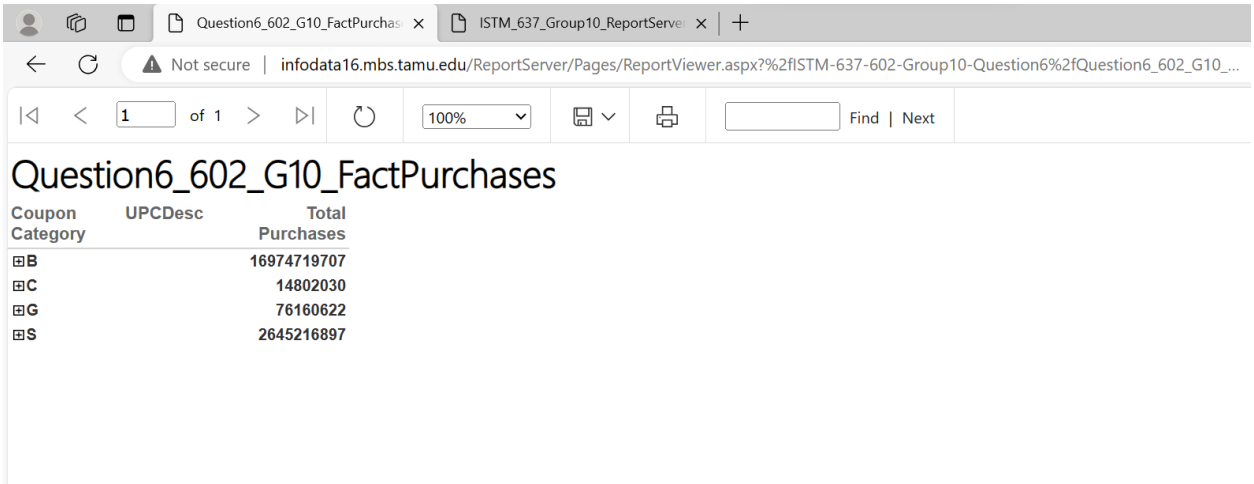


Figure: Report for Fact_Purchases shows the total purchases against each coupon category

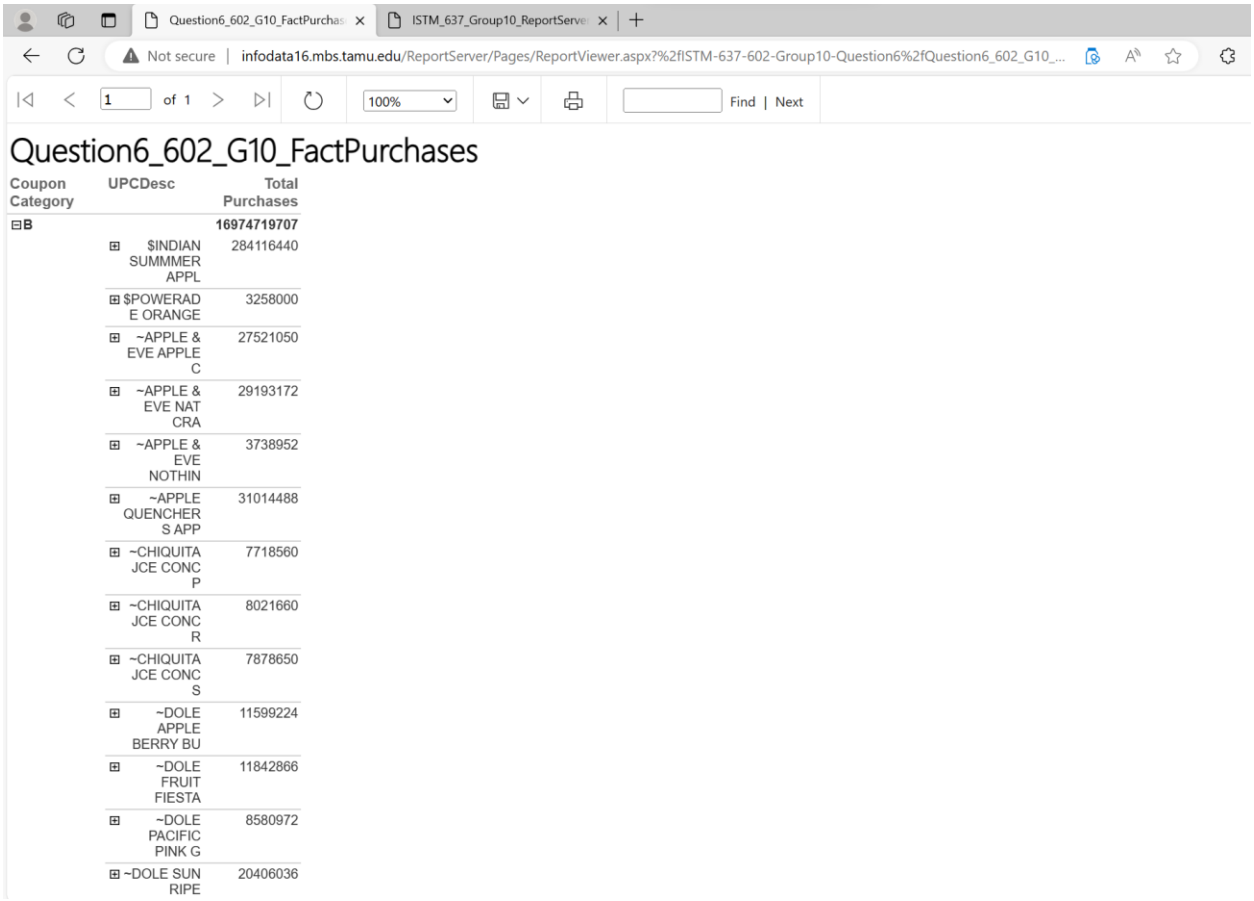


Figure: Drilling down to see different brands of juices against each juice category and their corresponding purchases

Section 7. Bibliography

Chicago Booth. (2011). Dominick's Dataset – Kilts Center. Retrieved from Chicago Booth: https://www.chicagobooth.edu/research/kilts/research-data/dominicks

Databricks. (2023, 1 1). Star Schema. Retrieved from Databricks: https://www.databricks.com/glossary/star-schema

J. Kaur, V. A. (2020). Influence of technological advances and change in marketing strategies using analytics in retail industry. International Journal of System Assurance Engineering and Management, 953–961.

Microsoft. (2023, 08 21). Columnstore indexes: Overview. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-overview?view=sql-server-ver16

Microsoft. (2023, 04 04). SQL Server and Azure SQL index architecture and design guide. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/sql/relational-databases/sql-server-index-design-guide?view=sql-server-2016

The Kimball Group. (1, 1 2023). Four-Step Dimensional Design Process. Retrieved from Kimball Group: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/four-4-step-design-process/

The Kimball Group. (2023, 1 1). Business Processes . Retrieved from Kimball Group: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/business-process/

The Kimball Group. (2023, 1 1). Dimensions for Descriptive Context. Retrieved from Kimball Group: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/dimensions-for-context/

The Kimball Group. (2023, 1 1). Facts for Measurements. Retrieved from Kimball Group: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/facts-for-measurement/

The Kimball Group. (2023, 1 1). Grain. Retrieved from Kimball Group: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/grain/

Chicago Booth. (2011). Dominick's Dataset – Kilts Center. Chicago Booth. Retrieved October 11, 2023, from https://www.chicagobooth.edu/research/kilts/research-data/dominicks

Kaur, J., Arora, V., & Bali, S. (2020). Influence of technological advances and change in marketing strategies using analytics in retail industry. International Journal of System Assurance Engineering and Management, 11(5), 953–961. https://doi.org/10.1007/s13198-020-01023-5

M, A. B., & Babu, H. S. (2018, April). Big Data Analytics – Its Impact on Changing Trends in Retail Industry. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 7(4), 10. ISSN: 2278 – 1323

Morimura, F., & Sakagawa, Y. (2023). The intermediating role of big data analytics capability between responsive and proactive market orientations and firm performance in the retail industry. In Journal of Retailing and Consumer Services (Vol. 71, p. 103193). Elsevier

BV. https://doi.org/10.1016/j.jretconser.2022.103193
Elgendy, N., & Elragal, A. (2016). Big Data Analytics in Support of the Decision Making Process. In Procedia Computer Science (Vol. 100, pp. 1071–1084). Elsevier BV.
https://doi.org/10.1016/j.procs.2016.09.251

Dominick’s Finer Foods, Inc. -- Company History. (n.d.). Dominick’s Finer Foods, Inc. -- Company History. https://www.company-histories.com/Dominicks-Finer-Foods-Inc-Company-History.html