Dominick's Fine Food

Final Project Report

Design and Implementation of Data Warehouse for a Retail Store

Section 602 - Group 10

25th April 2019

Email ID: somya@tamu.edu

SOMYA SHARMA 827006361
BHAVISHYA TYAGI 627006433
AMISHA SINGH 527005276

Credentials for Microsoft SQL Server Management Studio, Analysis Services and Reporting Services

Server Name: infodata16.mbs.tamu.edu

User Id: sh6361

Password: Mays6361

1. Microsoft SQL Server Management
   - Staging Server: ***group10_602_stagingdb***
   - Data warehouse Server: ***group10_602_datawarehouse***

2. Analysis Services
   - Question 2: 602_Group10_Q2
   - Question 5: 602_Group10_Q5

3. Reporting Server

infodata16.mbs.tamu.edu/ReportServer - /602_Group10

[To Parent Directory]

Wednesday, April 24, 2019 5:52 PM	167129	Coupon Promotions Effect Report
Wednesday, April 24, 2019 5:47 PM	77847	Customer_visit_report
Wednesday, April 24, 2019 5:46 PM	119076	Profit Percentage Report
Wednesday, April 24, 2019 5:52 PM	198690	Thanksgiving Sales Report

Microsoft SQL Server Reporting Services Version 13.0.5081.1

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Data Warehouse is a system which enables business to support decision making and reporting. It aggregates data from multiple OLTP sources which contains past and current information. To design data warehouse for Dominick's Fine Foods the first step involved would be analyzing the business requirements.

Dominick's Fine Foods is a Chicago based retail store. It had approximately 116 stores by the year 1998. The retail store maintains almost 3500 UPCs containing a variety of grocery, pharmacy, bulk foods, video, coupons and promotion details. The chain was a giant chain in terms of the number of products and customers. The success of retail industry is based on multiple customer preferences, promotions offered and marketing. So making an appropriate strategic decision based on the past information is very essential. A data warehouse is primarily built to address the business needs. Hence, a strategy should be adopted such that that all the business questions are addressed effectively. The business questions identified would provide possibilities. The areas which we have touched upon are effects of demographics on sales, sale trends of various products, promotion strategies, customer footfall trends, store-wise sales trends, etc.

The objective of this report is to present a logical and physical design for Dominick's Fine Foods. We justify why dimensional modeling using Kimball's methodology is appropriate to address the business requirements. We have presented the logical model using star schema that satisfies the requirement to answer the selected business questions. Additionally, we have justified how the proposed schema satisfies the business needs and how the schema aligns with the requirement. We also present the physical design based on the data aggregate plan, indexing plan, data standardization plan, and storage plan. Finally, we describe the implementation steps involved in designing the data warehouse using the extraction, cleansing, transformation and load procedures.

2. Challenges in dealing with Project

Every real life project comes with its own set of issues and challenges. This data warehousing project as well had various challenges associated with it. It required understanding the retail domain and dealing with the large set of data to come up with unique and innovative business questions. Following were the main challenges identified during the first phase of the project:

- As the Dominick Fine Foods data was large and dirty, it was time-consuming to understand the data and identify the correlation between different files.
- Data in the file was in different data formats i.e. CSV, TXT, as well as some of the data like week decode and store information was available in the Dominick Fine Foods manual.
- As multiple attributes were involved from various files it was difficult to design the ER diagram identifying correct relationships between the various entities identified
- The major challenge was brainstorming to identify the most relevant business questions and justifying it appropriately.
- Since there is no standard naming convention used across multiple files, it is very tedious to understand what the attribute signifies.
- Lastly, the challenge was to identify the files related to the business questions and create insightful pivot charts

To overcome the above challenges we utilized Excel and JMP software to better understand the data available. We referred to Dominick's Data Manual to understand the data and metadata which enabled us in framing business questions and justifying it correctly.

3. Data Description

3.1. Understanding of Data

Dominick's Finer Foods data spans over a period of 9 years from 1989 to 1997. It's a chain, headquartered at Chicago, expands over a 100-stores across the United States. It sells about 25 different categories of products, having more than 3500 Unique products. This retail chain has a database covering the store-level scanner data. The entire research data is divided into general files and category-specific Files.

3.2 Metadata for OLTP source files

The data files consist of the following-

A. Customer Count files
   - It contains information about in-store traffic. The data pertaining to the number of customers making purchases on a daily basis, specific to the store are present in the file.
   - The total sales of the products and total coupons redeemed by the store are also listed.

Variable	Description	Type	Length
DATE	Date of the Observation	Character	6
Week	Week Number	Numeric	8
Store	Store Code	Numeric	8
BAKCOUP	Bakery Coupons Redeemed	Numeric	8
BAKERY	Bakery Sales in Dollars	Numeric	8
BEER	Beer Sales in Dollars	Numeric	8
BOTTLE	Bottle Sales in Dollars	Numeric	8
BULK	Bulk Sales in Dollars	Numeric	8
BULKCOUP	Bulk Coupons Redeemed	Numeric	8
CAMERA	Camera Sales in Dollars	Numeric	8
CHEESE	Cheese Sales in Dollars	Numeric	8
CONVFOOD	Conventional Foods Sales in Dollars	Numeric	8
COSMCOUP	Cosmetics Coupons Redeemed	Numeric	8
COSMETIC	Cosmetics Sales in Dollars	Numeric	8
CUSTCOUN	Customer Count	Numeric	8
DAIRCOUP	Dairy Coupons Redeemed	Numeric	8
DAIRY	Dairy Sales in Dollars	Numeric	8
DELI	Deli Sales in Dollars	Numeric	8
DELICOUP	Deli Coupons Redeemed	Numeric	8

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

SALCOUP	Salad Coupons Redeemed	Numeric	8
SPIRITS	Spirits Sales in Dollars	Numeric	8
SSDELICP	Self Service Deli Sales in Dollars	Numeric	8
VIDCOUP	Video Coupons Redeemed	Numeric	8
VIDEO	Video Sales in Dollars	Numeric	8
VIDEOREN	Video Rentals (Dollar Amounts)	Numeric	8
WINE	Wine Sales in Dollars	Numeric	8
DELIEXPR	Deli Express Sales in Dollars	Numeric	8
DELISELF	Deli Self Service Sales in Dollars	Numeric	8
FISH	Fish Sales in Dollars	Numeric	8
FISHCOUP	Fish Coupons Redeemed	Numeric	8

Table 1:Metadata for CCount

B. Store-Specific Demographics
   - It contains detailed information about the store-specific demographics based on the United States Government consensus for the Chicago Metropolitan area. Market Metrics processed the data to generate demographic profiles for each DFF store.
   - The demographics are divided on the basis of age, ethnicity, level of income, household size, household values, singles, retired, unemployed, working women, ability to shop, etc.

Variable Name	Description
age9	% Population under age 9
age60	% Population over age 60
ethnic	% Blacks & Hispanics
educ	% College Graduates
nocar	% With No Vehicles
income	Log of Median Income
incsigma	Std dev of Income Distribution (Approximated)
hsizeavg	Average Household Size
hsize1	% of households with 1 person
hsize2	% of households with 2 persons
hsize34	% of households with 3 or 4 persons
hsize567	% of households with 5 or more persons
hh3plus	% of households with 3 or more persons
hh4plus	% of households with 4 or more persons
hhsingle	% of households with 1 person
hhlarge	% of households with 5 or more persons
workwom	% Working Women with full-time jobs
sinhouse	% Detached Houses
density	Trading Area in Sq Miles per Capita
hval150	% of Households with Value over $150,000
hval200	% of Households with Value over $200,000
hvalmean	Mean Household Value (Approximated)
single	% of Singles
retired	% of Retired
unemp	% of Unemployed
wrkch5	% of working women with children under 5
wrkch17	% of working women with children 6 - 17
nwrkch5	% of non-working women with children under 5
nwrkch17	% of non-working women with children 6 - 17
wrkch	% of working women with children
nwrkch	% of non-working women with children
wrkwch	% of working women with children under 5

wrkwnch	% of working women with no children
telephn	% of households with telephones
mortgage	% of households with mortgages
nwhite	% of population that is non-white
poverty	% of population with income under $15,000
shopcons	% of Constrained Shoppers
shophurr	% of Hurried Shoppers
shopavid	% of Avid Shoppers
shopstr	% of Shopping Stranges
shopunft	% of Unfettered Shoppers
shopbird	% of Shopper Birds
shopindx	Ability to Shop (Car and Single Family House)
shpindx	Ability to Shop (Car and Single Family House)

Table 2: Metadata for Demographics

C. UPC files

- This file contains the description of each UPC belonging to each product category. Also, the file names are in the form of upcxxx. Here, xxx denotes the three-letter acronym for the category.
- The information has UPC name, commodity code, item code, description, item was drop-shipped or warehoused, etc.

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

Table 3: Metadata for UPC Files

D. Movement Data by UPC

- It has the sales information at every store level, for each UPC category, on a weekly basis. The files are named wxxx. The xxx defined the three- letter acronym for the respective category.
- Information such as the price of a product, units sold, profits gained on that particular product, etc.

Variable	Description	Type	Length
upc	UPC Number	Numeric	8
store	**Store Number**	numeric	3
week	**Week Number**	Numeric	3
move	Number of unit sold	Numeric	8
price	Retail Price	Numeric	8
qty	Number of item bundled together	Numeric	3
profit	Gross margin	Numeric	8
sale	Sale code (B,C,S)	Character	8
ok	1 for valid data, 0 for trash	Numeric	3

Table 4: Metadata for Movement Files

E. Category Description

- It includes the UPC's and their corresponding descriptions and size for a particular product category.
- The acronym for the product category, last update details, file size, number of observations, first week in the file, last week in the file, no. of UPC's are also mentioned.

Column Name	**Description**	**Example**
Category	Product category names has the Dominicks product categories	Analgesics
UPC	It contains three letter code of every category suffixed with upc	upcana
Movement	It contains three letter code of every category suffixed with w	wana

Table 5: Metadata for Category

F. Dominick's stores
- The city, price tier, zone, zip code and address information for each store are listed in this.

Column Name	Description	Example
Store	It has store number for Dominick's store	2
City	It contains city of the store	River Forest
Price Tier	It describes the price tier of the particular store	Low
Zone	It divided the store to different zones	2
Zip code	It contains zip code for the store	66054
Address	It contains address of stores	7502, North Avenue

Table 6: Metadata for Store

G. Week's Decode Table
- This SAS file contains a week variable that has been coded to give us the week corresponding to which a sales data is recorded. This also keeps a track of weeks having special events.

Column Name	Description	Example
Week#	It contains week number based on Dominick's start week	5
Start	It contains start date of the week	10/26/1989
End	It contains end date of the week	11/1/1989
Special Event	It contains special public event detail	Halloween

Table 7: Metadata for Weekdecode

3.3 Entity Relationship Diagram

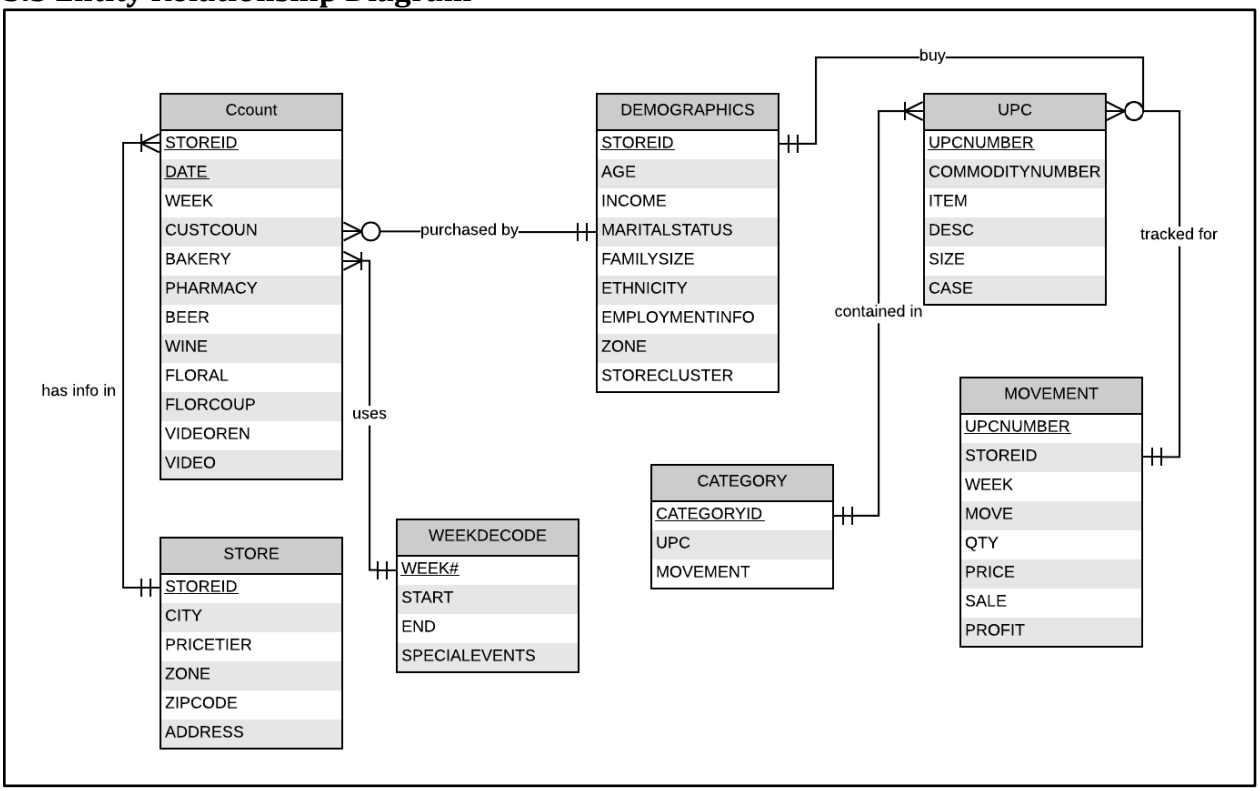


Figure 1: Entity Relationship Diagram

4. Domain Understanding

The retail industry demands that certain measures or decisions be undertaken to overcome the challenges associated with the highly competitive nature of the industry. Thus, it often utilizes data warehouses and BI system to enhance business activities. This helps make more strategic decisions using structured and unstructured data at all organisational levels. It also helps differentiate the Strategic Business unit (SBU) from all other firms. The three main stages in the process are- analysis of the business processes, dimensional data modelling, and implementation of the data warehouse.The systems used for generating the data warehouse are SSAS, SSIS and SSRS. All the phases are covered in this approach, like analysis, design and development.

Additionally, the current data should be studied for understanding. Next, the business requirements are taken and the questions to be worked on are developed. Then the logical and physical models for the warehouse are designed. The schema is penned down to show the relationship between the facts and the dimensions. This is how we will work on our DFF's project as well. Further, the data mart design is specified by specifying measures and then

reconciled in OLTP. Using ETL, data activities like cleaning are performed. Further, data is loaded into the warehouse and the dimension and fact tables are populated. Data in the warehouse are verified by generating reports. It is developed for entire enterprise using data marts to maintain the integrity.

For instance, a study of a paper, presented a retail analysis based on consumer behavior and retail trends. It used ESRI Tapestry demographics segmentation data and presented the market profile for each segment based on the data. The main objective was to identify the need to scale and support new businesses analyzing the current shopping trends. It exhibited summarized reports for the customer spending patterns and established SPI(Spending Potential Index) and MPI(Market Potential Index) for each category. It also obtained customer segmentation based on the socioeconomic data that enables us to understand which products a customer segment is interested in. A comparison of demand and sale trends to identify the retail gap was also presented. These findings enabled in recognizing certain areas of potential growth which would satisfy the market demand and reduce the retail gap.Thus, the report helped us understand the trends and issues common in the retail domain. As Dominick's Finer Foods also maintains store-wise customer demographics data based on age, ethnicity, the number of dependents and salary, we can perform a similar analysis to understand the sales based on the demographics.

In conclusion, from our domain research we gained perspective on the business needs to focus on pertaining to the problems when there are lot of structured and unstructured data which is very similar to the data that we have in DFF. Also, a good data design is key for optimal working warehouse. Data warehouse helps us decide how core business processes such as marketing, purchasing, managing customer relationship, etc. can be effectively implemented using warehousing in the retail industry. Thus, the domain research has laid out the relevant groundwork for our Dominick's Finer Foods project and how helped us establish our own approach in a similar fashion.

REFERENCES

1. Oketunji, Omadara (2011), "Design of Data Warehouse and Business Intelligence System", retrieved from https://www.diva-portal.org/smash/get/diva2:831050/FULLTEXT01.pdf

2. S. Habte, K. Ouazzane et al. (Vol:11, No:7, 2017), "Generic Data Warehousing for Consumer Electronics Retail Industry", retrieved from http://repository.londonmet.ac.uk/1268/1/Journal_paper_generic-data-warehousing-for-consumer-electronics-retail-industry.pdf

3. "Retail Market Analysis", (May 2018) retrieved from https://www.amherstma.gov/DocumentCenter/View/44885/Amherst-Retail-Analysis-Final?bidId=

4. "Oracle Retail Data Warehouse", http://www.oracle.com/us/industries/retail/046439.pdf

5. Business Questions

- Which brand of beer has the highest sale from 1992 - 1997 across all the stores?

Justification: The data provide total sales of Beer brands from the year 1992 to 1997 for all the stores. We have used this data to plot sales of beer store-wise. This data analysis would help business understand which Beer brands are more in demand and which brands have the least sales. Moreover, these trends would help in gauging the inventory and purchase changes required for the high selling brands. Additionally, the business can take measures to remove the lowest selling beer brands which would reduce the inventory maintenance. Similar trend analysis can be used for other products to identify the changes required to run the business more efficiently.

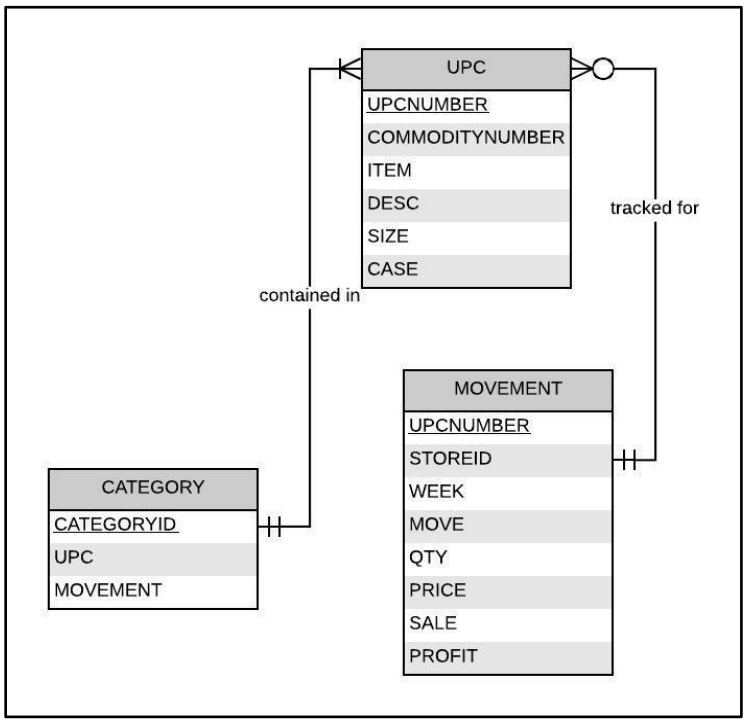


Figure 2: ER for Business Question 1

Sum of STORE	Store Number			
Beer Brands	Store 8	Store 9	Store 12	Grand Total
AMSTEL LIGHT BIER		720		720
BALLARD BITTER N.R.		342		342
BEER LIMIT	624	666	1152	2442
BUD ICE	400	756	960	2116
BUDWEISER BEER	6296	7956	14760	29012
BUDWEISER BEER LONG	2384	1989	3480	7853
BUDWEISER BEER N.R.B	4432	3978	7956	16366
BUDWEISER DRY BEER	2880	5589	7956	16425
BUDWEISER DRY LONGNE	1768	1989	2652	6409
BUDWEISER ICE DRAF	1920	1512	2952	6384
BUDWEISER ICE DRAFT	720	756	1080	2556
BUDWEISER LIGHT BEER	5432	9531	15912	30875
BUDWEISER LIGHT LONG	2400	1989	3552	7941
BUDWEISER LIGHT NR	1768	1989	2652	6409
BUSCH BEER	5144	3978	6372	15494
BUSCH LIGHT BEER	4288	3672	5388	13348
ELK MOUNTAIN RED LAG		342		342

Figure 3: Data for Business Question 1

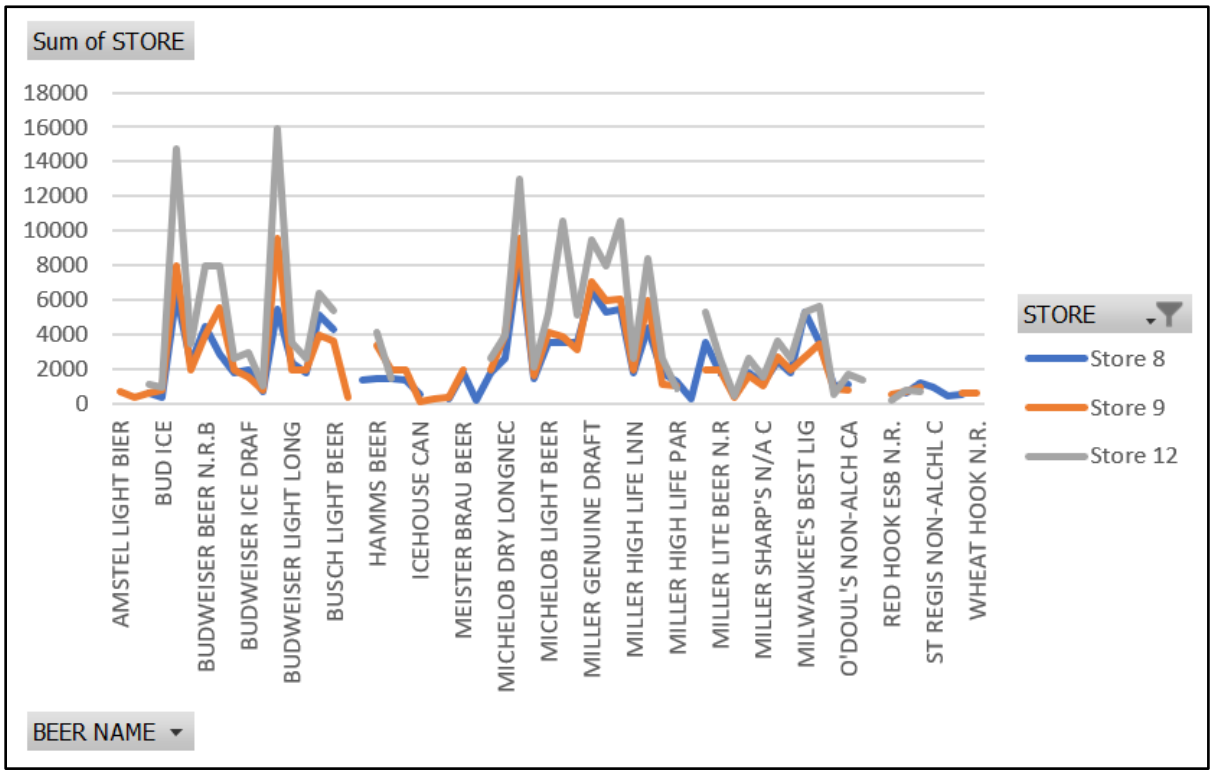


Figure 4: Graph for Business Question 1

✓ What is the sales trend for Thanksgiving week each year? Which products had the highest sale over the years during this time?

Justification: The sales of products generally increase during the holiday period. For this analysis, we have gathered product category sales data for Thanksgiving week during the year 1989 to 1996. We are here trying to capture product sales for various categories and trying to find which product has the highest sales during Thanksgiving week. The above graph presents the sales over the years during Thanksgiving week. The analysis of the highest selling product categories will help the business determine how to effectively handle holidays weeks better. It will enable the business to stock up as per the estimated sale trends. Moreover, additional coupons and promotions can be provided on products to attract more number of customers.

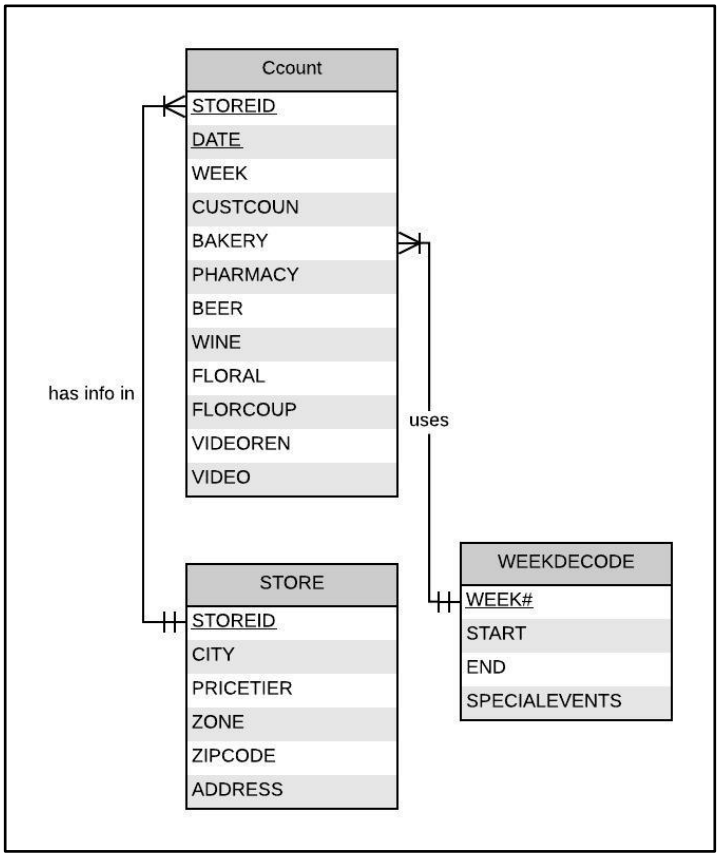


Figure 5: ER diagram for Business Question 2

Year	Sum of MEAT	Sum of DAIRY	Sum of FROZEN	Sum of FLORAL	Sum of DELI	Sum of CHEESE	Sum of BAKERY	Sum of BEER	Sum of WINE
1989	2572313.77	2905647.92	1796837.35	244795.44	1117125.95	127580.18	717438.27	339088.99	299353.29
1990	2749952.78	2962823.57	1928462.9	287800.94	1186132.95	130314.72	747661.12	388883.38	304045.23
1991	2652114.72	3450683.98	2076980.29	283666.29	1275933.62	148982.53	825725.8	404957.35	347778.3
1992	2587609.26	3341385.84	2050864.37	345450.62	1537585.96	175695.78	856059.81	452393.71	383512.63
1993	2541409.09	3211195.79	2135779.35	355921.39	1649440.47	183104.85	925901.8	466144.23	409375
1994	2565596.89	3290635.67	2204499.93	404636.2	1608853.53	204039.65	944087.05	490550.03	455226.77
1995	2365282.98	3115135.68	2147368.5	434240.23	1538157.86	193687.07	936721.69	518835.83	466245.66
1996	2523714.45	3430776.31	2258155.32	483542.17	1622865.34	216727.39	998041.11	564903.94	546964.66
Grand Total	**20557993.94**	**25708284.76**	**16598948.01**	**2840053.28**	**11536095.68**	**1380132.17**	**6951636.65**	**3625757.46**	**3212501.54**

Figure 6: Data for Business Question 2

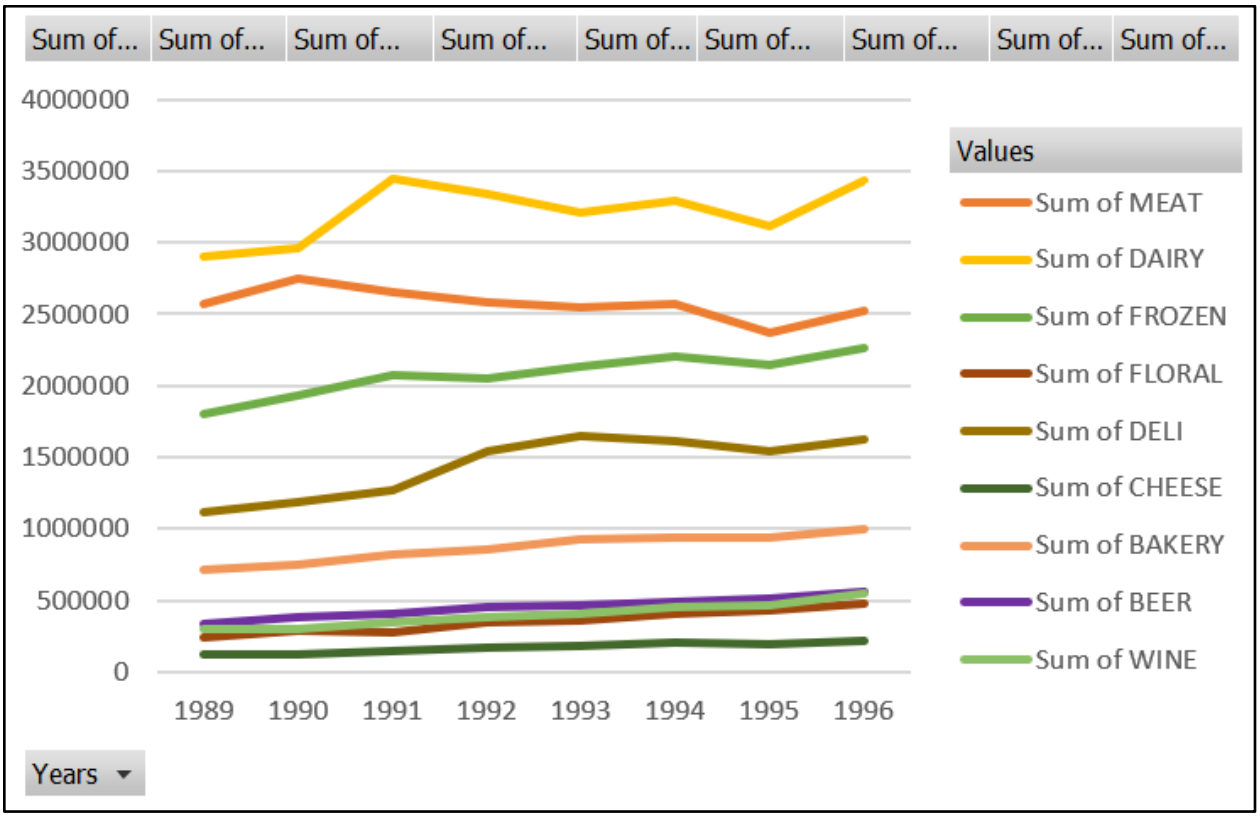


Figure 7: : Graph for Business Question 2

- What is the trend of frozen dinner sales for each price tiers over the years?

Justification: According to the Dominick Fine Food data we have 4 price tiers i.e. High, Medium, Low and Cub Fighter. We have analyzed Frozen foods data across these price tiers. The graph depicts the trend of Frozen food sales with respect to tiers. This analysis would help business in understanding the product needs in a particular tier. This will help business on focusing sales of products in the respective tier. We have taken a single product to demonstrate this idea. However, this analysis can be applied to various products and would help in making strategic business decisions.

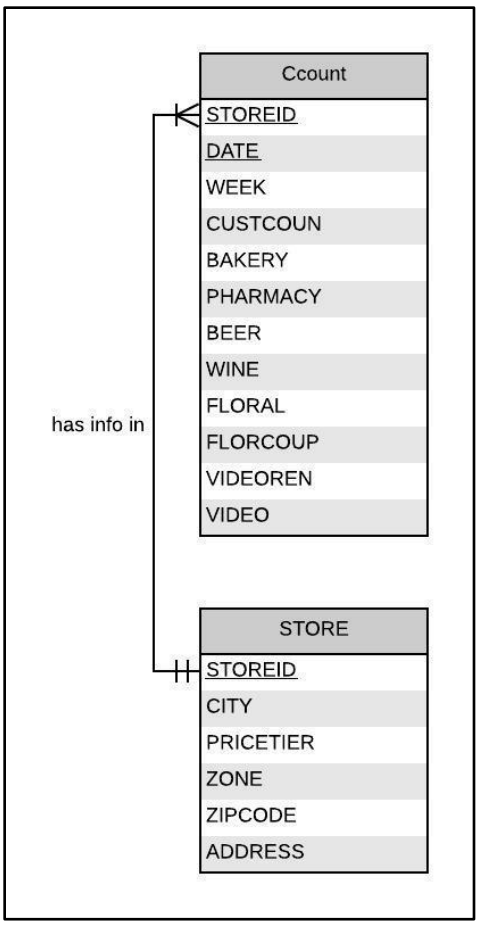


Figure 8: ER Diagram for Business Question 3

Sum of SALES	Year						
Price Tier	1991	1992	1993	1994	1995	1996	Grand Total
⊞CubFighter	60578.21333	62126.87	78726.35	152310.4	187074.57	56840.11	597656.5133
⊞High	129790.4433	121366.9	162379.74	332966.09	435774.34	122263.89	1304541.403
⊞Low	68298.23	65494.24	91176.77	156599.37	194849.1	57612.7	634030.41
⊞Medium	198096.3133	192589.28	251525.89	498331.23	573390.18	161797.64	1875730.533
Grand Total	456763.2	441577.29	583808.75	1140207.09	1391088.19	398514.34	4411958.86

Figure 9: Data for Business Question 3

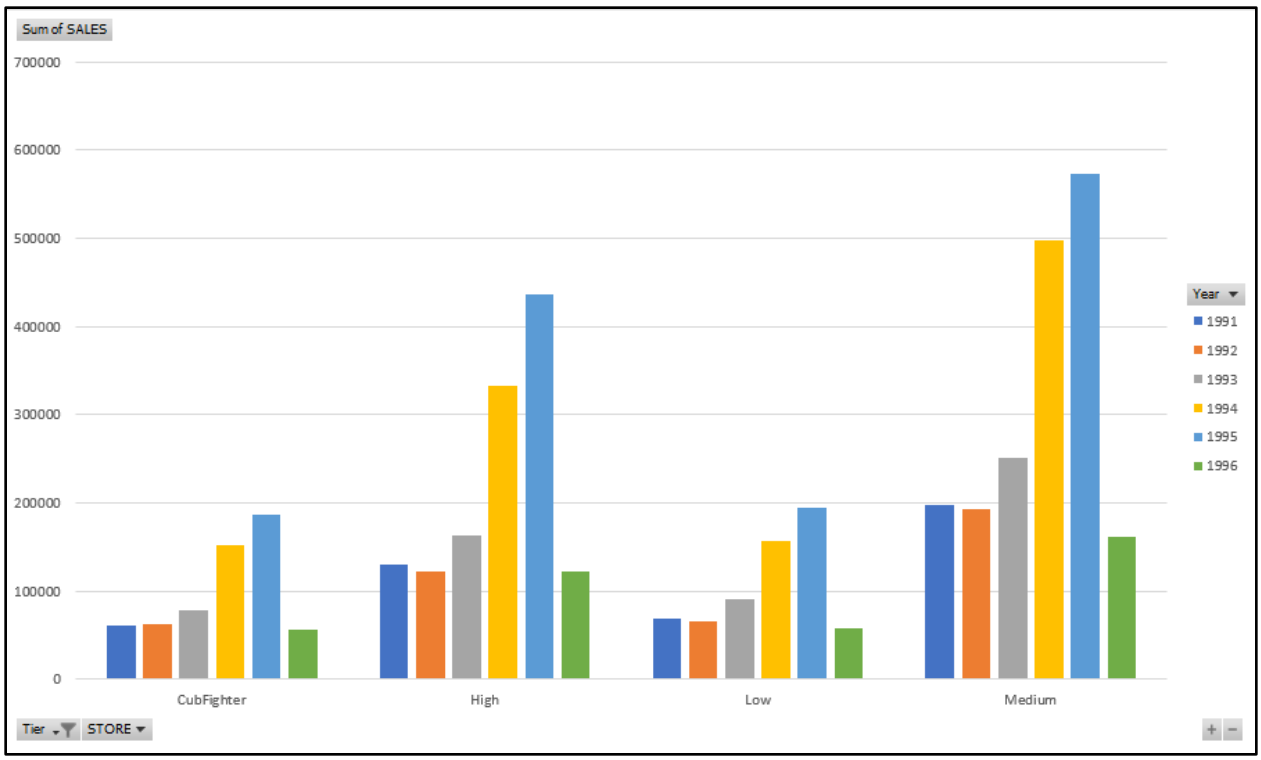


Figure 10: Graph for Business Question 3

✓ **What is the impact of age-wise regional demography on the sales of pharmacy products?**

Justification: Here we are considering the sales of pharmacy product for the entire life of DFF for every location. The differentiation factor is on the basis of age of the customer meaning if they are below age 9 or above 60 because either infants or senior people need more medications. The visualization will let us know the purchasing trend of analgesics and which stores are making maximum sales and which not so that we can increase or decrease the products based on customer behavior. No coupons have been considered here. This will also help to realize what are the reasons of low sales at certain locations and understanding what does the customer or buyers actually wants in terms of analgesics. We can also think of taking measures of increasing sales like including a delivery option of such products so that people do not have to travel much, or reduce waiting time. This becomes an important question when the major chunk of store sales are made through pharmacy products.

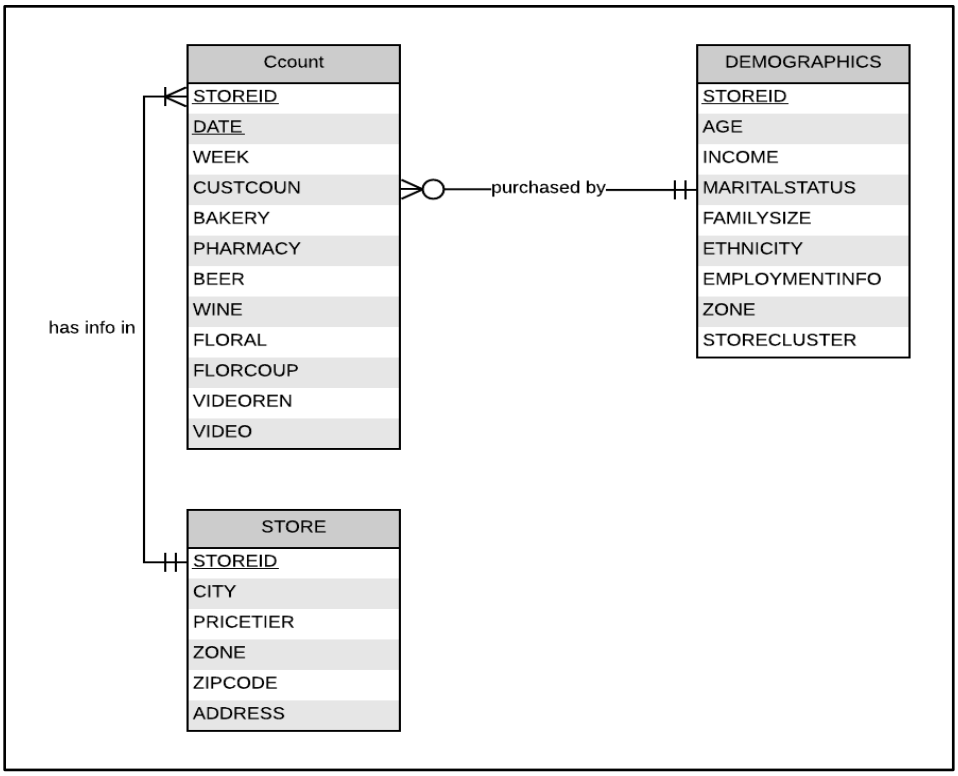


Figure 11: : ER Diagram for Business Question 4

Store Numb	Pharmacy products bought	Population below age 9	Population above age 60
5	2762022	0.141433483	0.117368032
8	1457135	0.123155416	0.252394035
9	771334	0.103503097	0.269119018
12	4383415	0.10569674	0.178341405
14	3125337	0.129589372	0.213949275
18	59	0.110094984	0.272313368
28	11	0.128879537	0.213308785
33	50	0.046070917	0.134169966
44	2658	0.144883485	0.190982776
49	6	0.134877735	0.187473188
67	21.36	0.118820051	0.210272984
95	3722879	0.118820051	0.210272984
109	24.65	0.147519804	0.151055656
137	4257971	0.146442828	0.20960245
Grand Total	20482923.01	1.6997875	2.810623919

Figure 12: : Data for Business Question 4

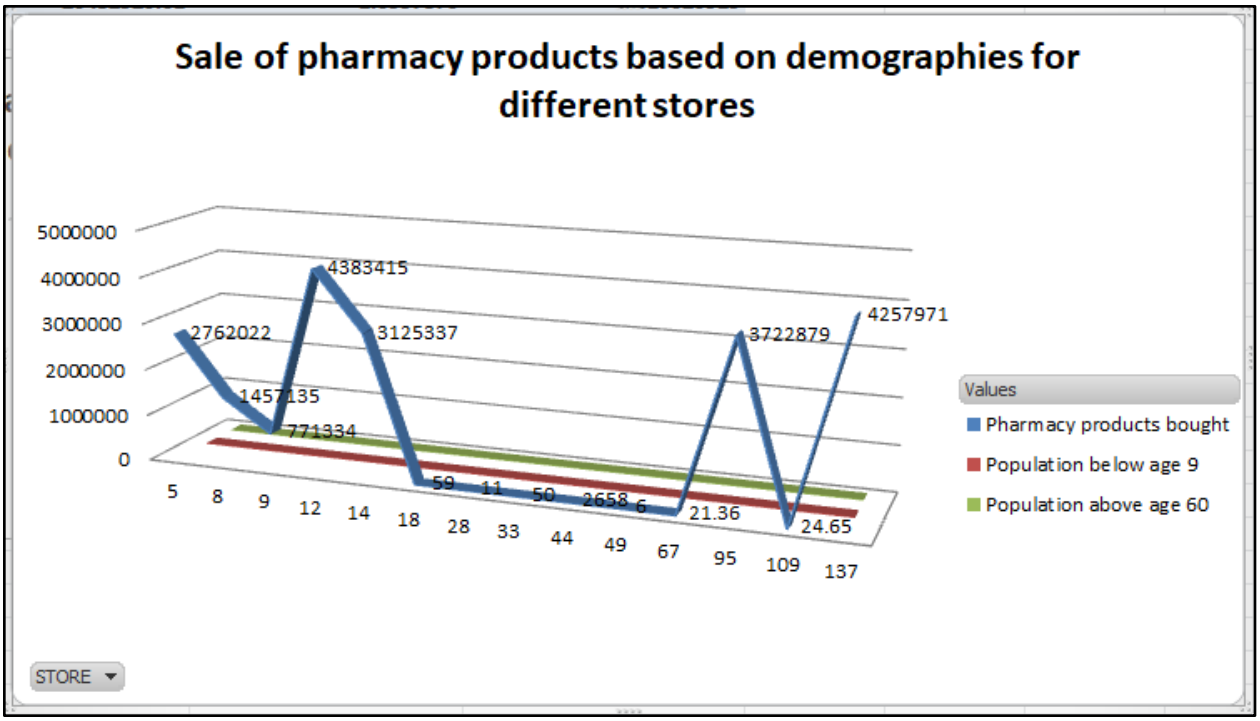


Figure 13: Graph for Business Question 4

✓ **What is the effect of coupon promotions on the sale of different products store-wise? Do coupons impact product sales?**

Justification: Here the sales for different products and their corresponding coupons are considered for various store locations for the entire life span of DFF. Considering the sales product wise and then comparing them to the coupon sales. It will let us know what impact the coupon promotions has on purchase of product or any new product introduced in the store is making sales because of coupons or not. This is further segregated based on store locations to have a deeper view and do analysis in more detail. This is done to have required level of granularity. Here for justification, we just considered bakery coupons, bulk coupons, floral coupons and photo coupons to have an idea of how to approach the problem and what repercussions will it have on the overall sale process and the enterprise as a whole. This can be also used to know if the customers are attracted to a store because they offer more coupon offers and can company use this as a medium to attract the customers and do marketing for their store. Also if the sales are less for a location then can coupon promotion boost the sale of that store and make a balance of coupons across different locations.

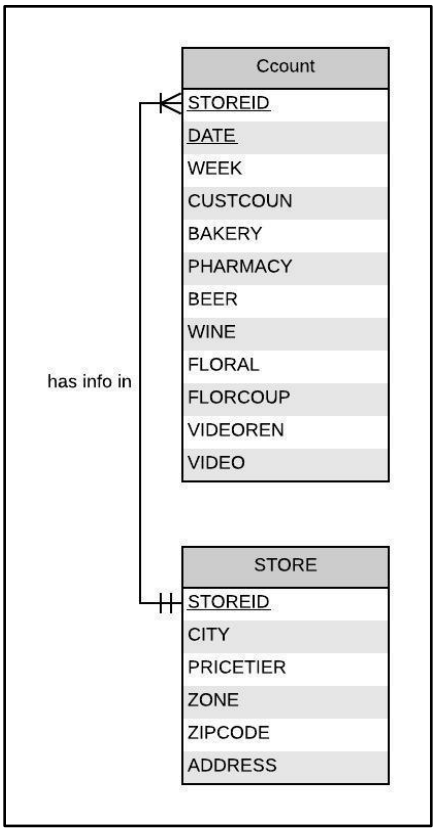


Figure 14: ER Diagram for Business Question 5

-

STORES	(All)								
BakerySales	BakeryCouponSale	BulkSales	BulkCouponSales	CosmeticSales	CosmeticCouponsSa	FloralSales	FloralCouponSales	PhotoSales	PhotoCouponSales
45544827.82	683254.17	15711437.29	24365.73	757305	3491.17	18132702.02	312835.56	8198371.51	586736.38

Figure 15: Data for Business Question 5

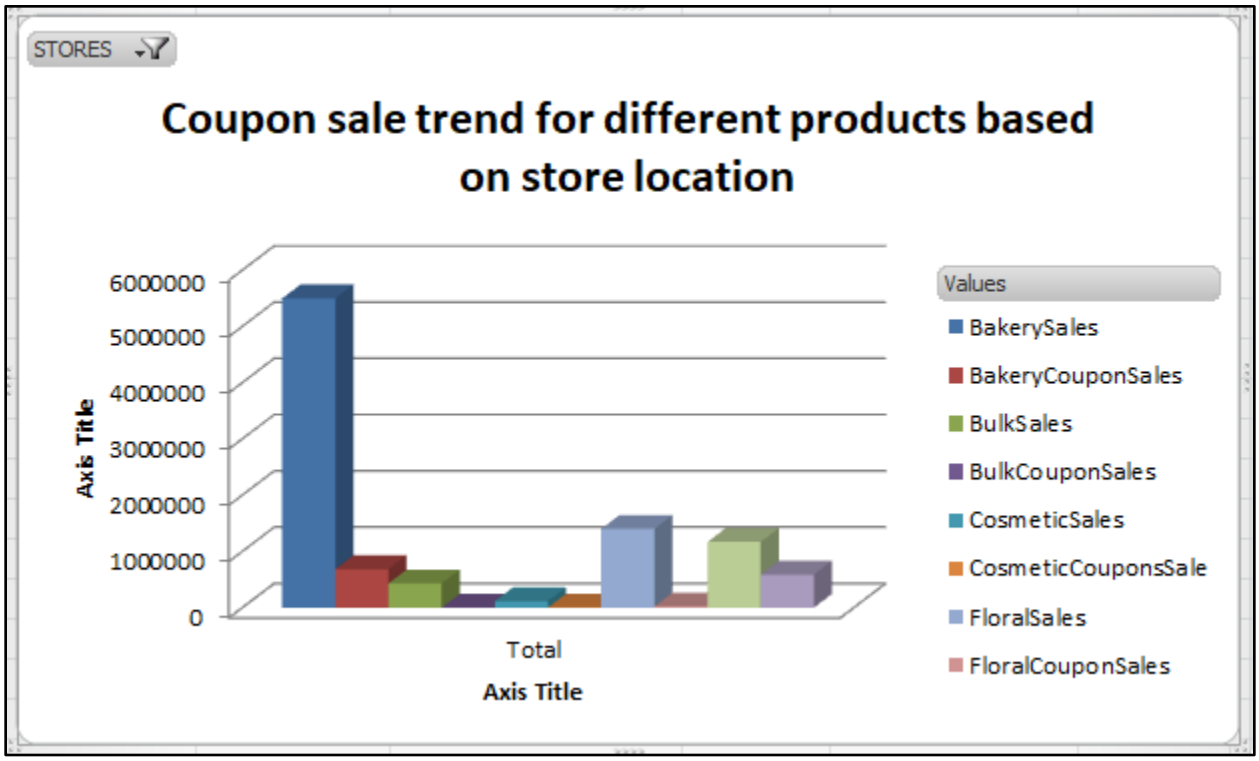


Figure 16: Graph for Business Question 5

- Which stores lie in the bottom 25% in terms of sales of video rentals?

Justification: Video rentals data can be used to see what stores are making maximum sales and data is segregated in quartiles. This can be used to see which stores lie in bottom 25% in terms of rentals and required action can be taken to either improve the sales through coupon promotions or remove the rentals options for those stores. Also this can be used to further analyse how often customers rent videos and this can be compared with people buying videos and renting videos.

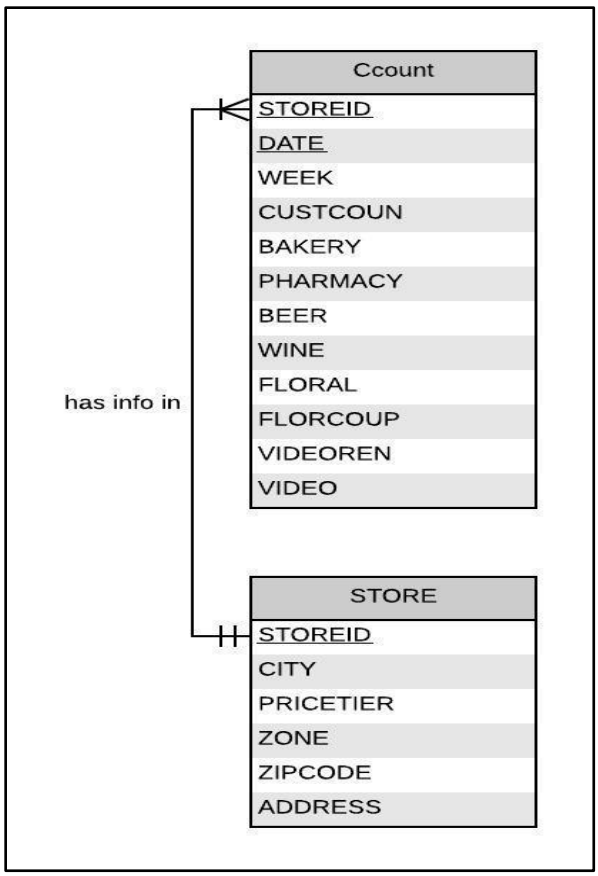


Figure 17: ER Diagram for Business Question 6

Quaterly Distribution	(All)
Store Number	Sum of VIDEORENTALS
2	30354
8	401227.52
14	410.71
18	263
28	480.98
33	58.11
40	429980.64
46	285.25
54	54.86
59	360332.86
67	322.12
70	292.37
74	148064.87
80	12212.35
95	71.6
107	645813.65
111	41.01
121	634856.49
126	935063.48
137	108.94
Grand Total	3600294.81

Figure 18: Data for Business Question 6

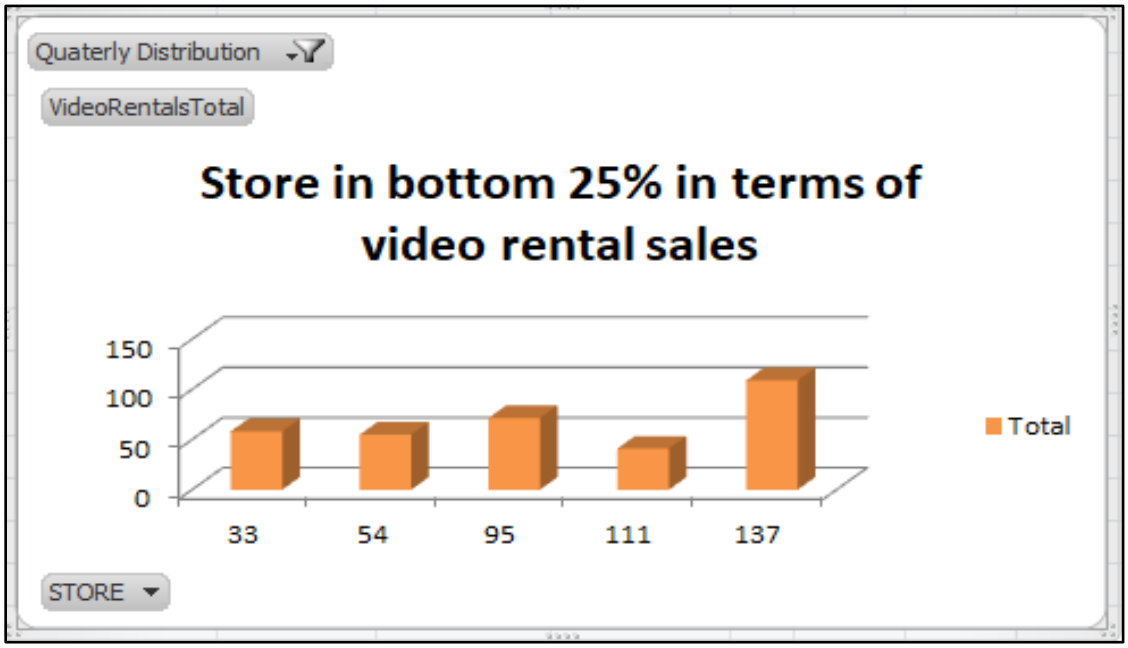


Figure 19: Graph for Business Question 6

✓ What is the customer increase or decrease over years based on the store?

Justification: The CCount file contains the store-specific details of customer count based on date. This data can be used in analyzing the customer visiting trends of the stores. The data provides with the customer count day wise for every store. Customers are the sole source of the retail business. Projecting the customer count numbers of stores comparatively will help us make strategic decisions related to staffing and resourcing. Customer counting is an important management tool to measure store performance and analysis. This analysis also helps in-store profile analysis by finding the store footfall details indicating the store surging and falling trends.

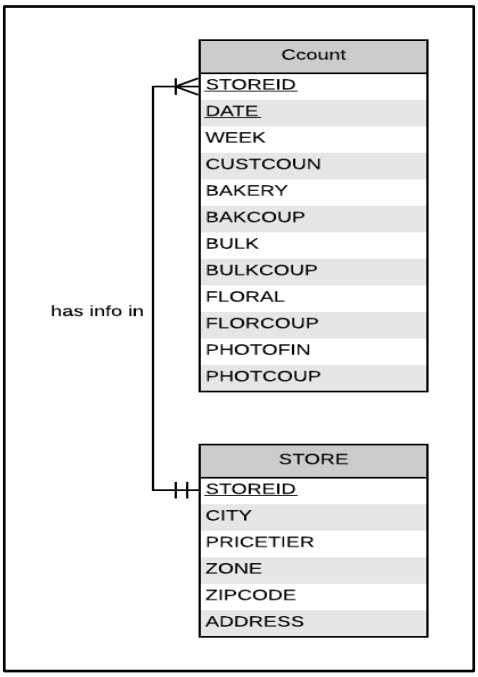


Figure 20: ER Diagram for Business Question 7

Row Labels	Sum of CUSTCOUN
1988	**22596**
Store 8	13699
Store 14	8897
1989	**30574**
Store 8	19497
Store 14	11077
1990	**29069**
Store 8	18234
Store 14	10835
1991	**28197**
Store 8	17629
Store 14	10568
1992	**30150**
Store 8	18290
Store 14	11860
1993	**29256**
Store 8	16863
Store 14	12393

Figure 21: Data for Business Question 7

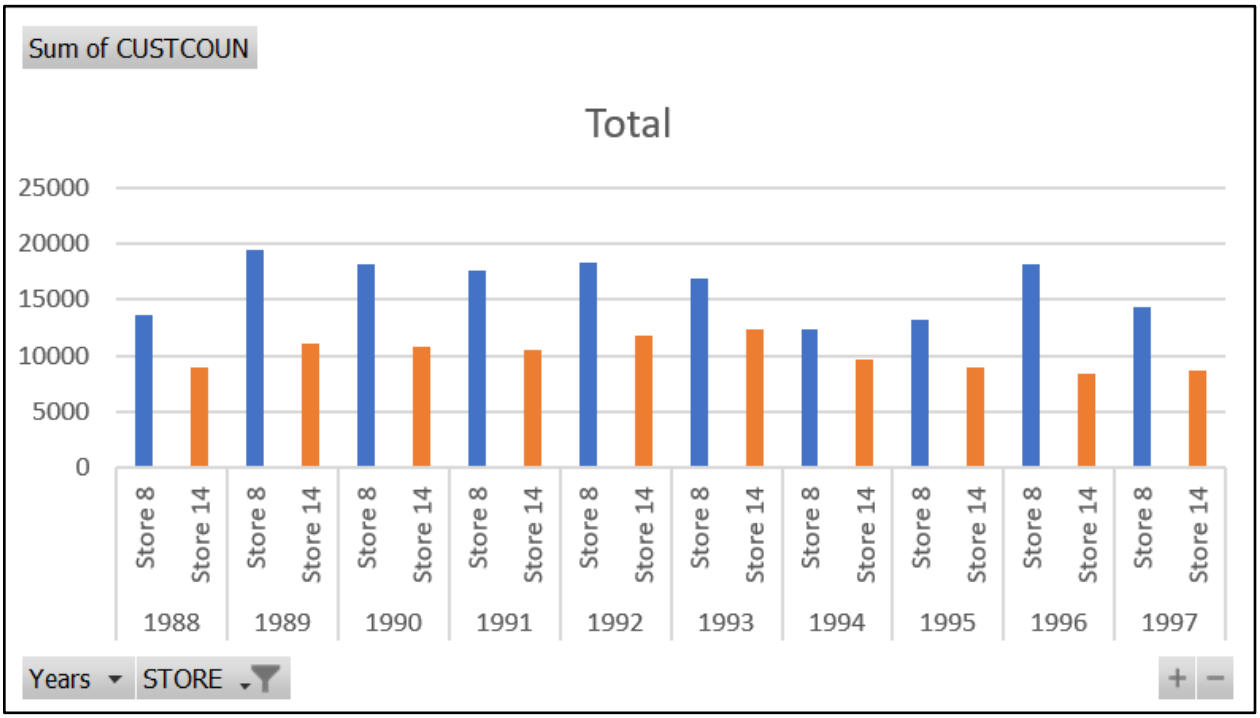


Figure 22: Graph for Business Question 7

- What is the location wise sales and which of these stores have above average sales between the years 1992 to 1997?

Justification: The CCount file for Dominick's data provides us with the store numbers and the yearly sales trend in every store. This data can provide us insight about several things. It will tell us about our optimum store location. The stores that fetch us maximum sales or above average returns should focus their strategy on retaining the customers. The stores with below average returns, should be shut down if returns are very low, or focus on attracting customers through providing incentives in the form of coupons, etc.

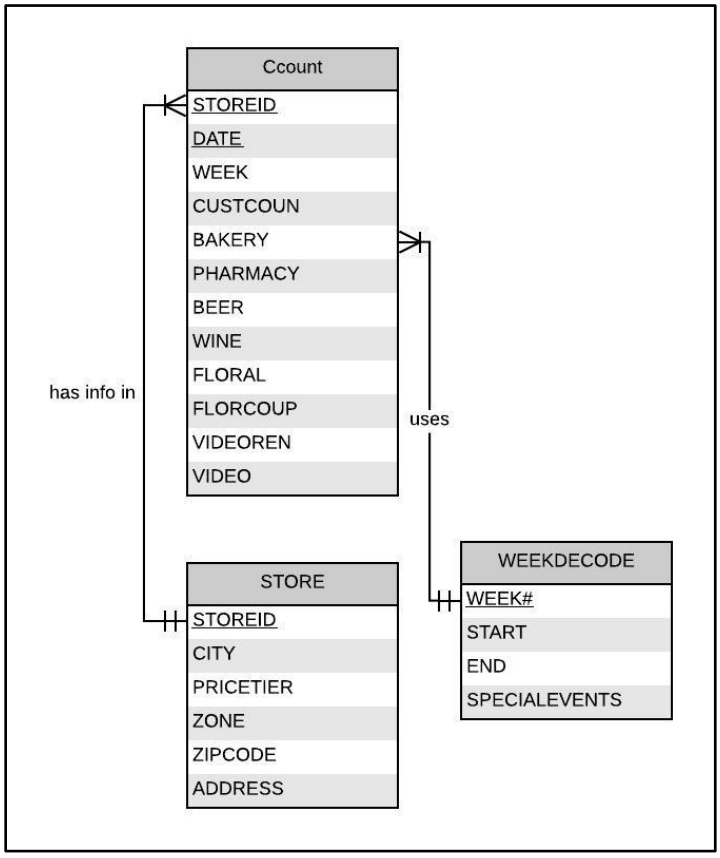


Figure 23: ER Diagram for Business Question 8

	Column Labels							
	Store 8		Store 14		Store 44		Total Sum of Total Sales	Total % of Sales
Row Labels	Sum of Total Sales	% of Sales	Sum of Total Sales	% of Sales	Sum of Total Sales	% of Sales		
⊞1992	459397.74	39.34%	335547.7	28.74%	372753.39	31.92%	1167698.83	100.00%
⊞1993	295179.56	36.87%	274024.68	34.22%	231453.76	28.91%	800658	100.00%
⊞1994	269364.41	31.52%	267932.13	31.35%	317230.34	37.12%	854526.88	100.00%
⊞1995	436057.42	41.85%	274440.08	26.34%	331481.61	31.81%	1041979.11	100.00%
⊞1996	379822.78	40.23%	261822.99	27.73%	302454.44	32.04%	944100.21	100.00%
⊞1997	412853.78	39.79%	282543.02	27.23%	342161.65	32.98%	1037558.45	100.00%
Grand Total	2252675.69	38.53%	1696310.6	29.01%	1897535.19	32.46%	5846521.48	100.00%

Figure 24: Data for Business Question 8

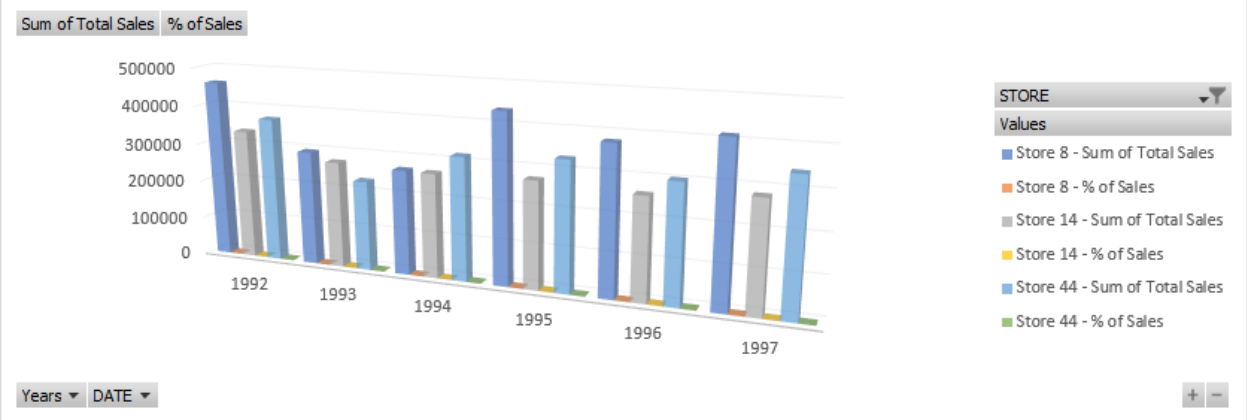


Figure 25: Graph for Business Question 8

✓ How are the product category profits changing in every store over the years? What are the product categories with the highest and least profits?

Justification: The movement data for Dominick's has the files for each product category, named after the product category itself. Each file enlists the Unique product code (UPC) of the products belonging to that category. The profit on each product, every week, in each store is also listed. From this, we have calculated the yearly profits for each product category, store-wise. This data is relevant to the business, as it gives us an overview of the product categories that are being sold more and those that are not, and the profits incurred. This data will be beneficial for us to minimize the stock keeping unit (SKU's) of products that are not being sold, and clear more shelf space for products that provide higher profits. Thus, in turn, we can keep more SKU's of the high selling product categories depending on the sales in the stores over the years.

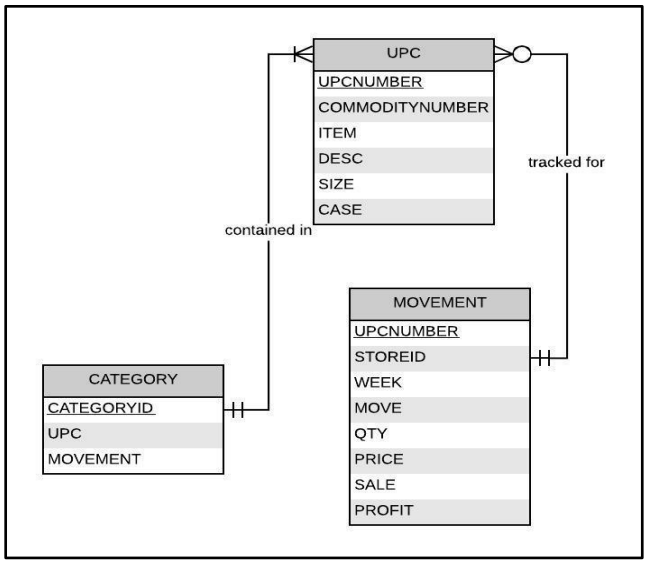


Figure 26: ER Diagram for Business Question 9

Sum of PROFIT	Column Labels							
	Store 2			Store 2 Total	Store 8			Store 8 Total	Grand Total
	Cheese	Cigarettes	Cookies		Cheese	Cigarettes	Cookies		
Products									
1988	112.32	656.65	1910.22	2679.19	160977.51	659.01	1910.02	163546.54	166225.73
1989	31.36	373.52	2048.7	2453.58	150404.67	819.79	1867.25	153091.71	155545.29
1990		3599.69		3599.69		1989.88		1989.88	5589.57
1991	96.84	3902.19	1722.83	5721.86	187645.27	1552.75	1589.14	190787.16	196509.02
1992	105.62	3041.25	1821.16	4968.03	165591.71	1932.89	1428.1	168952.7	173920.73
1993		2843.68	1422.67	4266.35	70237.96	2614.79	1404.18	74256.93	78523.28
1994		1729.78	1188.3	2918.08	196414.37	946.51	1028.32	198389.2	201307.28
1995		1667.66	1324.1	2991.76	197227.42	1265.7	1130	199623.12	202614.88
Grand Total	**346.14**	**17814.42**	**11437.98**	**29598.54**	**1128498.91**	**11781.32**	**10357.01**	**1150637.24**	**1180235.78**

Figure 27: Data for Business Question 9

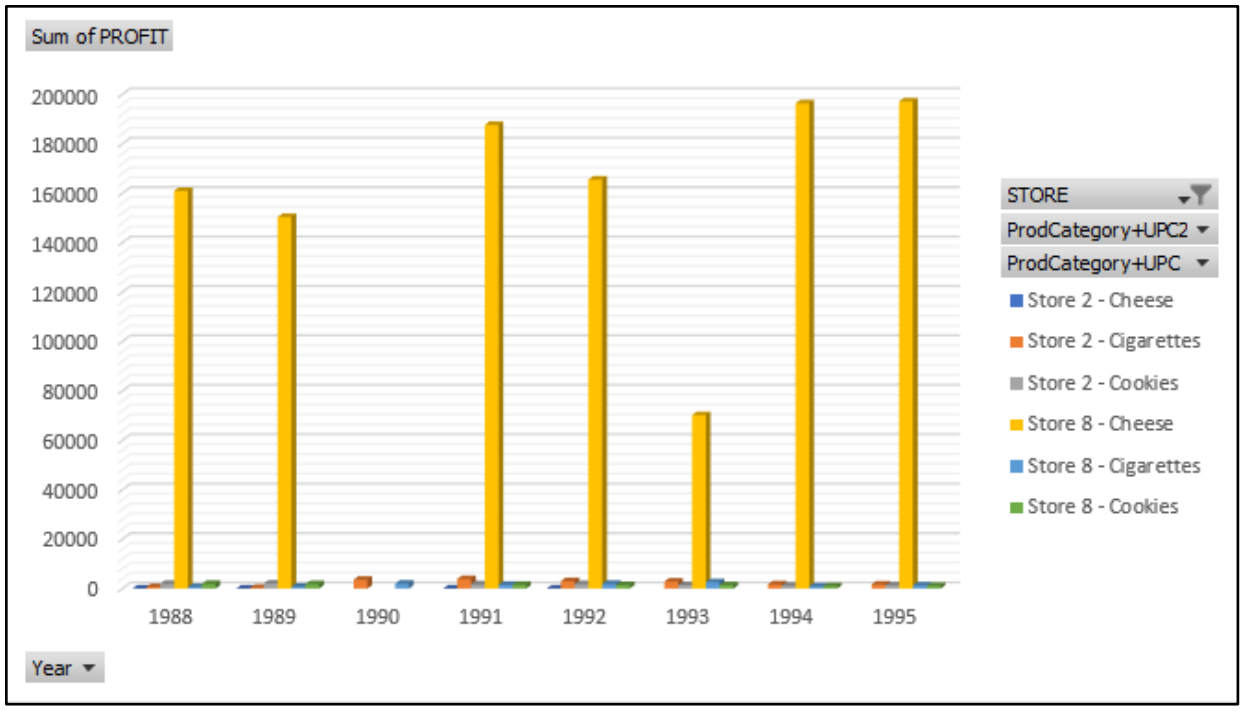


Figure 28: Graph for Business Question 9

× How is the average transaction value for every year changing for each store?

Justification: Dominick's data provides us with the sales of every product date-wise and across the various stores. It also shows the customer count data in the CCount file. From this, we can deduce the average transaction value for every store and it's change over the years. The formula is as follows-

Average Transaction=Total revenue/ Number of Transactions

In the above mentioned formula, the total revenue is the same as total sales, as we are assuming our product sales are the only source of income. The number of transactions can be deduced from the customer count number. Thus, we can appropriately find the Average Transaction value. This will allow the business to know how much people are spending on an average, on each purchase they make. If the amount is higher, customers are purchasing more expensive products or more number of items. If the value is low, customers are purchasing more low prices products or less number of items. This data will help us determine whether we need to rethink our pricing strategy, use sale tactics such as using coupons for promotions and so on, based on the results.

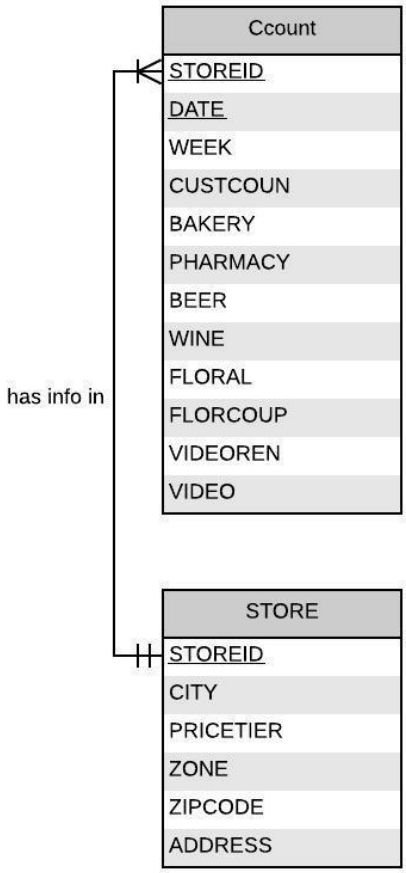


Figure 29: ER Diagram for Business Question 10

Sum of Avg Trans. Value	Column Labels			
Row Labels	Store 2	Store 4	Store 5	Grand Total
1988	87.63225411	59.74795926		147.3802134
1989	70.48313558	60.88863196	67.20147013	198.5732377
1990	73.75692061	63.0895545	76.81188326	213.6583584
1991	63.49587395	54.48834824	83.75646356	201.7406857
1992	60.29472898	49.28827105	70.69983787	180.2828379
1993	69.73458918		73.0512419	142.7858311
1994	66.60876499		78.6090383	145.2178033
1995	67.15844303		80.32349797	147.481941
1996	59.71894962		76.48602975	136.2049794
1997	84.75315468			84.75315468
Grand Total	**703.6368147**	**287.502765**	**606.9394627**	**1598.079042**

Figure 30: Data for Business QUestion 10

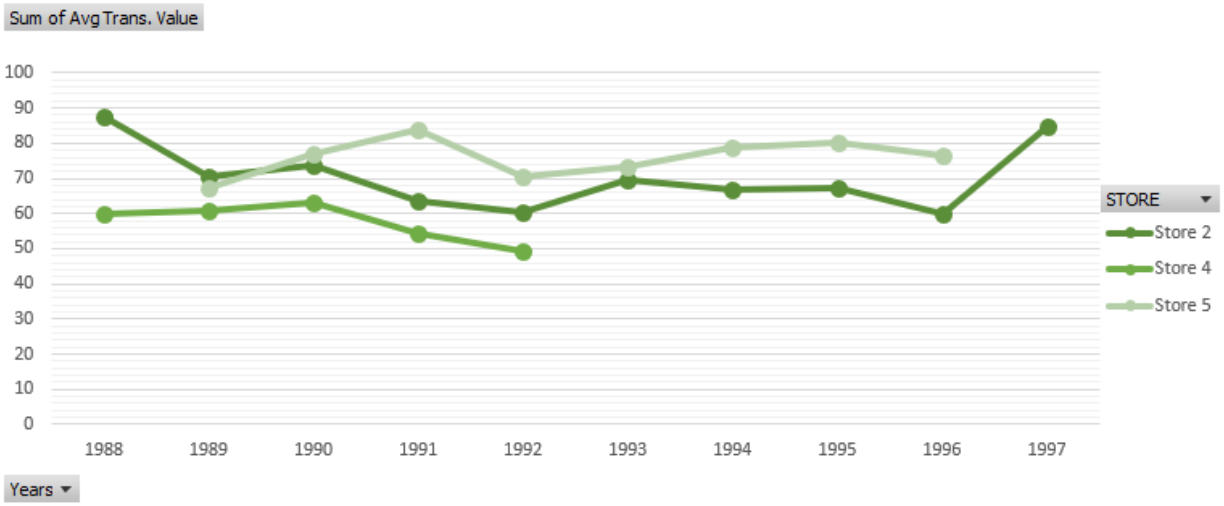


Figure 31: Graph for Business Question 10

We would address 5 questions from the above 10 questions by implementation a data warehouse. One of the 2 symbols have been used against every question which has significance as below:

- ✓ - means that question is selected to be further worked on
- × - means that question is not selected

6. Logical Design

6.1 Why Kimball's methodology?

We will use the Kimball approach for the business questions which requires the creation of a dimensional model (star schema). Kimball's approach requires the creation of data marts first as it is a bottom-up approach. Kimball's strategy is appropriate in this scenario as it would enable us to see quick results without much overhead to having a complete full-proof master plan of a data warehouse as in case of Inmon's methodology. Dimensional modeling enables ease of end-user accessibility and high-level performance. Moreover, this approach will take less time and less initial cost. Dimension modeling will also enable us to capture the critical measures which are intuitive for the business users.

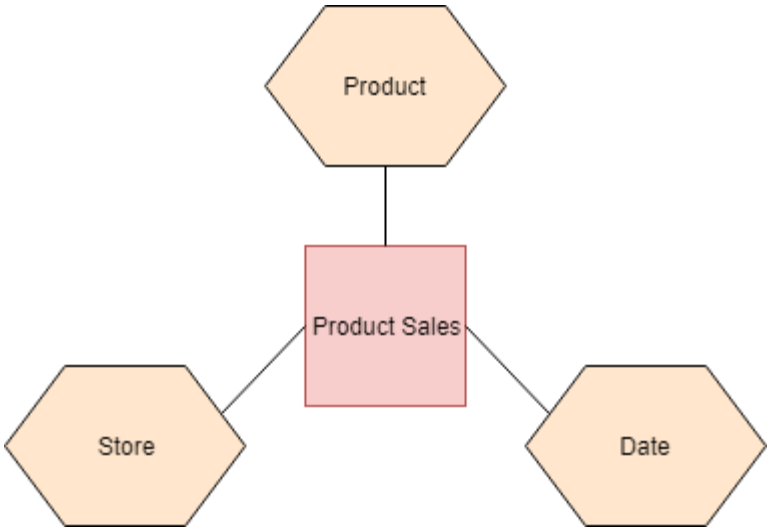


Figure 32: Star Diagram

To address all the business questions, we have proposed the schema including three dimensions i.e Product, Store, and Date. Dim_Store has details of all Dominick's Fine foods store located across the United States. The dimension is populated based on Dominick's research manual "Dominick's Stores". It contains store hierarchy attributes like store number, zip code, city, and zone. The store_key attribute is the surrogate key for the store. The dim_product dimension contains the data of all the products sold at DFF. The data for the products is taken from the UPC files available for all the product categories at DFF. The product hierarchy has the following attributes: UPC code, category code, and category name. The product_key attribute is the surrogate key for the store. The dim_date dimension contains time instances at the lowest granularity of a daily basis. Date_key is the surrogate key which is auto-incremented. The "Week's decode table" contains data for 400 weeks starting from 14th September 1989 to 14th May 1997. To record the occurrence at the lowest atomic level we include day_of_week and day_of_month fields, we need to split the start and end dates and capture all the days between that period.

6.2 Dimensional Modeling

1. **Dimension Product**

Product dimension stores all the product available and sold at Dominick's Fine Foods. The dimension is named as dim_product. Following is the description of the attributes of the dim_product table:

- ***product_key*** - This is the surrogate key that uniquely identifies the product dimension.
- ***upc_code*** - The last five digits of the code identify the product and the remaining digits identify the manufacturer.
- ***product_desc*** - These are the names of the product.
- ***category_code*** - This is the Dominick's item code
- ***category_name*** - Three letter acronym for the product category.

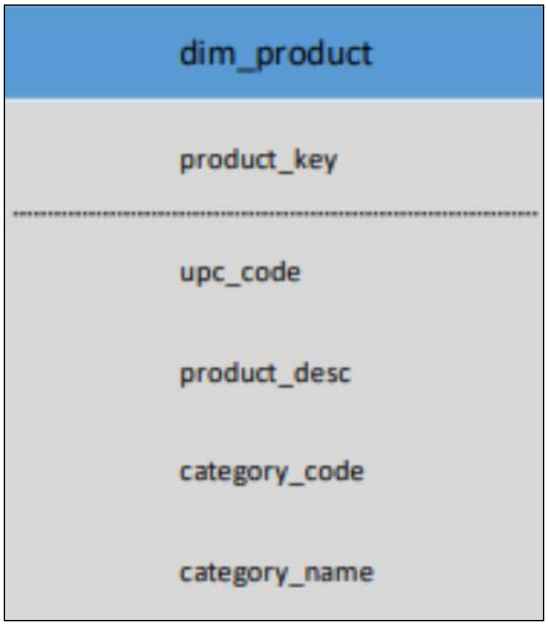


Figure 33: Product Dimension

2. Dimension Date

Date dimension is used to store time-related attributes. It is named as dim_date. Following is the description of the attributes of the dim_date table:

- ***date_key*** - This is the surrogate key that uniquely identifies the date dimension.
- ***day_of_month*** -This identifies the date of observation
- ***week_no*** - This identifies the week number of the observation

- ***month*** - This identifies the month of the observation
- ***Year*** - This identifies the year of the observation
- ***holiday_flag*** - If the flag is 1 or true, then it means it is a holiday. Otherwise, if the flat is 0 or false, then there isn't a holiday.
- ***event_desc*** -The identifier indicates the special event in the year eg- Christmas, thanksgiving, etc.

Figure 34: Date Dimension

3. Dimension Store

Store dimension stores the store-level details of all the retail chains of DFF. It is named as dim_store. Following are the attributes of dim_store:

- ***store_key*** - This is the surrogate key that uniquely identifies the store dimension.
- ***store_num*** - This is the number assigned to a particular store.
- ***zipcode*** - The zipcode of the corresponding store.
- ***city*** -The city where the store is located.

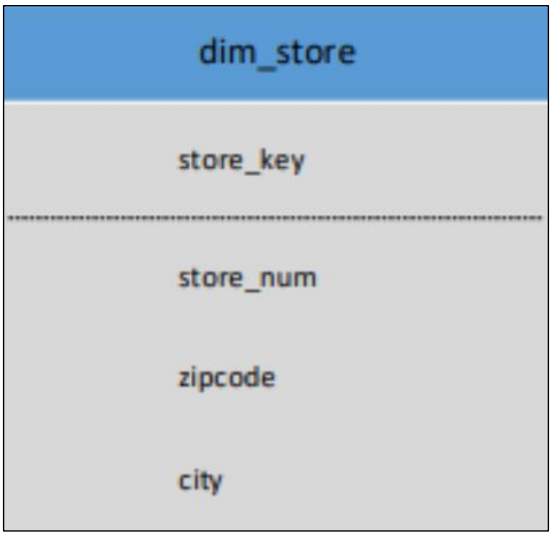


Figure 35: Store Dimension

4. Fact Product Sales

This table consists of the quantitative sales information of the products across the stores.

The fact table has the following concatenated keys:

- ***store_key*** - It is the unique identifier which has been generated for the store dimension as a surrogate key.
- ***date_key*** - It is the unique identifier which has been generated for date dimension as a surrogate key.
- ***product_key*** - It is the unique identifier which has been generated for the product dimension as a surrogate key.

The fact table has the following facts (measures):

- ***unit_price*** - This is the unit price of the bundled product
- ***quantity*** - Size of the bundle
- ***number_of_units_sold -*** This indicates the no. of units sold.
- ***product_category_sales*** - It indicates the dollar value of the product sales for a particular product category.

- ***product_sales*** - It indicates the dollar value of the product sales for a particular product. Where product sales are calculated as-

  *Product_sales = (Unit_price * Number_of_units_sold) / Quantity*

- ***product_category_profit*** - This indicates the profit margin generated by DFF for a specific product category
- ***product_profit*** - This indicates the profit margin made by DFF by selling a particular product.
- ***coupon_sales*** - This indicates the dollar value of the promotional coupon for each product category
- ***population%_below_age9*** - This indicates the percentage population of the customer base that is below the age of 9.
- ***population%_above_age60*** - This indicates the percentage population of the customer base that is above the age of 6.
- ***customer_count*** - This indicates the number of customer buying or making a purchase at the stores

Figure 36: Fact Product Sales

6.3 Kimball Rules

Kimball states that for greater understanding and superior performance, it is good to use the dimensional model instead of the ER model. And recommends following the 10 rules to achieve granularity, flexibility and proofed information resource. Here is the list of rules:

Rule 1: Dimensional structures should be loaded with detailed atomic data.

For the star schema generated, the dimension tables such as date, product, and store are all created to maintain the atomic level of granularity. For example, let's look at the date dimension it has the lowest level of granularity, date, which is further summarized as the week, month and year. Same for the store where data is maintained for the individual store then zip and further aggregated to the city. For product also the lowest level of granularity is maintained by having UPC codes and further summing up to category codes. According to this rule, the level of atomicity is maintained while creating dimensions.

Rule 2: Structure dimensional model around business processes

Here we have one fact table which will contain all the metrics for the product sales, category sales and the customer count.

Rule 3: Date dimension table should be associated with every fact table

The date is an important dimension when designing a dimensional model and here also we created date dimension because each transaction is taking place based on date. This date dimension is required to have a different level of granularities such as date, week, month and year which is further connected to fact table as a foreign key.

Rule 4: Master fact table containing all fact should have them in the same level of detail

For the fact table we have maintained the granularity for all the metrics. For the product sales fact table, the sales based on product and then higher hierarchy for category is provided to have required level of hierarchies.

Rule 5: There should be no existence of many-to-many relationship in fact table

There is no many-to-many relationship in the schema prepared. We do have product sales fact table to avoid this violation.

Rule 6: There should be no existence of many-to-one relationship in dimension table

Many-to-one relationship has been resolved by including metrics in the fact table for business events. Measures are included to have product sales and further rolled up to category sales. This is done to have required level of granularity.

Rule 7: Dimension tables should store report labels and filter domain values

There will be no null values in dimension tables. All transformations will be performed to check for null values and work on it if any is found before loading it to the warehouse. The events and description of any acronym used are properly stated and defined to avoid any sort of confusion.

Rule 8: Surrogate key should be used by dimension table

Every dimension has its own surrogate key. Product dimension has product_key, store has store_key and date has date_key as surrogate keys.

Rule 9: Data should be integrated across the enterprise using conformed Dimension

Here we have one data mart called *sales* having one schema. This eliminates the need to have integration for dimensions.

Rule 10: Match business requirements to realities to have DW/BI solution acceptable by users

All the business questions are taken into consideration while creating the schema. This is done so that when the end user runs any query they get the required results easily and summarization are created to reduce processing time.

	Dim_Store	Dim_Date	Dim_Product
Fact_Product_Sales	X	X	X

Table 8: Dimension-Fact Mapping

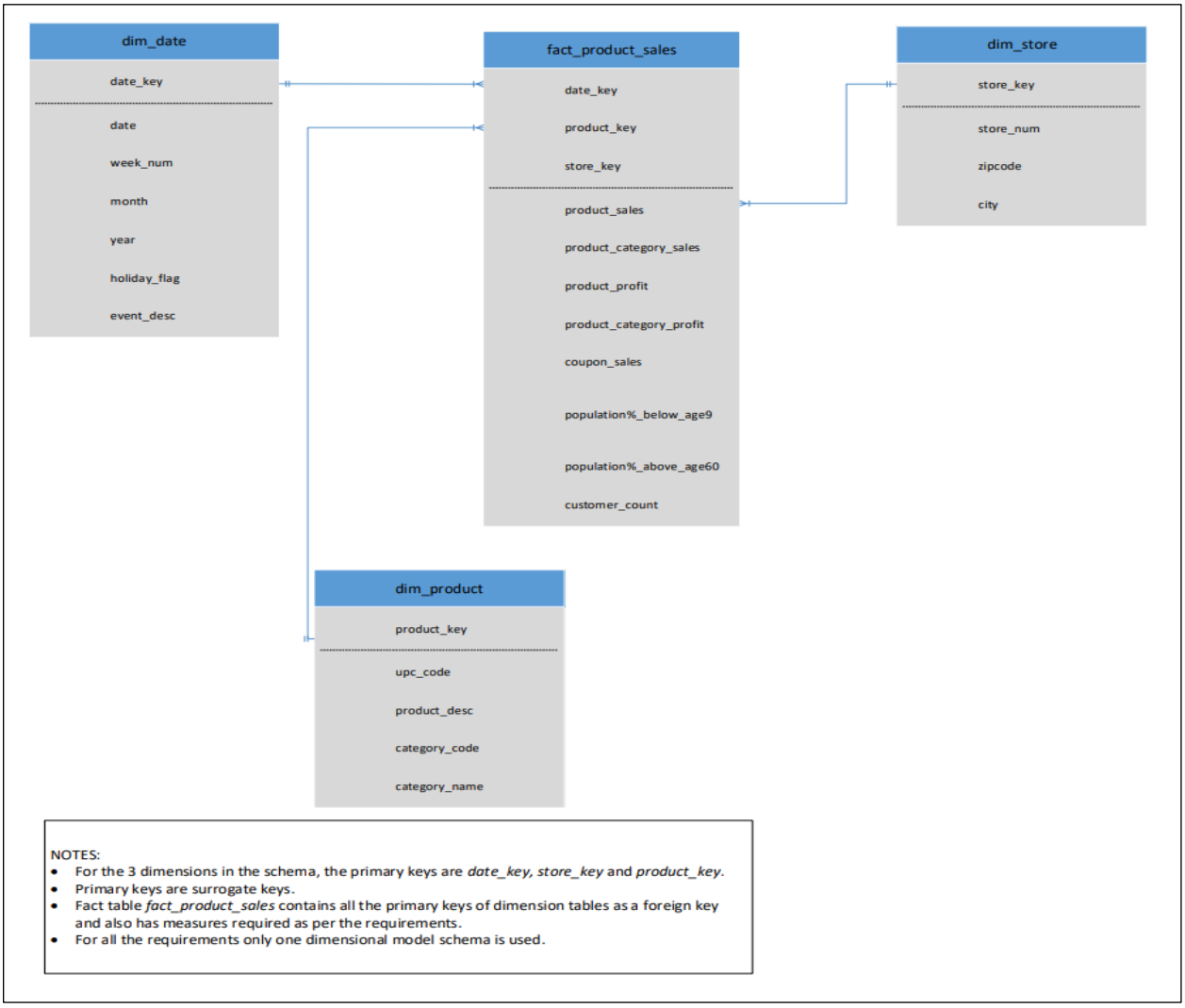


Figure 37: Star Schema for Product Sales

6.4 Star Schema

Using the dimensions and fact tables defined above, we have created a data mart. The data mart is for Product Sales. This data mart is sufficient to address all the business questions.

Below data mart is the Product Sales data mart. It comprises of three dimensions i.e. Store, Product, and Date. The below star schema is sufficient to answer questions 1 to 5 described further in the section.

7. Mapping Table

7.1 Dimension Date mapping table

Source System/ Table	Source Table Attribute	DW Dimension Target Table	DW Dimension Target Table Attribute	Data Type	Mapping Function	Handling Rules
Surrogate key		dim_date	date_key	Int	Autogenerated	
Week's Decode Table	Start/End		day_of_month	int	transform	Day_of_month is populated based on the dates between the start date and end date period
	Week #		week_no	int	copy	
	Start/End		month	varchar	transform	Trim the month from the date
	Start/End		year	int	transform	Trim the year from the date
	Special events		holiday_flag	int	transform	Set flag to 1 incase of holiday else 0
	Special events		event_desc	Varchar	copy	

Table 9: Mapping for Date Dimension

7.2 Dimension Product mapping table

Source System/ Table	Source Table Attribute	DW Dimension Target Table	DW Dimension Target Table Attribute	Data Type	Mapping Function	Handling Rules
Surrogate key		dim_product	product_key	Int	Autogenerated	
upcxxx.csv	UPC		upc_code	Int	copy	
	COM_CODE		category_code	Int	copy	
	Xxx from upcxxx filename		category_name	varchar	transform	Category name is taken from the filename trimming string
	DESCRIP		product_desc	varchar	copy	

Table 10: Mapping for Product Dimension

7.3 Store mapping table

Source System/ Table	Source Table Attribute	DW Dimension Target Table	DW Dimension Target Table Attribute	Data Type	Mapping Function	Handling Rules
Surrogate key		dim_store	store_key	int	Autogenerated	
Dominick's Store	STORE		store_num	int	copy	
	ZIP		zipcode	int	copy	
	CITY		city	varchar	copy	

Table 11: Mapping for Store Dimension

7.4 Fact Sales mapping table

Source System/ Table	Source Table Attribute	DW Fact Target Table	DW Fact Target Table Attribute	Data Type	Mapping Function	Handling Rules
Foreign key of dim_date		fact_product_sales	date_key	int	copy	
Foreign key of dim_product			product_key	int	copy	
Foreign key of dim_store			store_key	int	copy	
			product_category_sales	float		

			product_sales	float	transform	Sales = (Unit_price * Number_of_units_sold) / Quantity
			product_category_profit	float		
			product_profit	float		
			coupon_sales	float		
demo.csv	AGE9		%population_below_age9	float	copy	
demo.csv	AGE60		%population_below_age60	float	copy	
ccount.csv	CUSTCOUN		customer_count	int	copy	

Table 12: : Mapping for Fact Product Sales

8. Schema Justification for Business Questions

1. What is the sales trend for Thanksgiving week each year? Which product categories had the highest sale over the years during this time?

Justification: For this business question, we need to analyze the trend of sales of product categories over the years. We prepared the fact table schema for *fact_product_sales* based on the dimension stores, products, and date. Hence, the measure *product_sales* will have sales of a particular product in a particular store during a particular period of time. We need to identify the sales for the Thanksgiving week, and hence we summarized the data on a weekly basis as well as filtering sales based on the Thanksgiving week. This will help us analyze how over the years of DFF sales, which products were sold highest during this time.

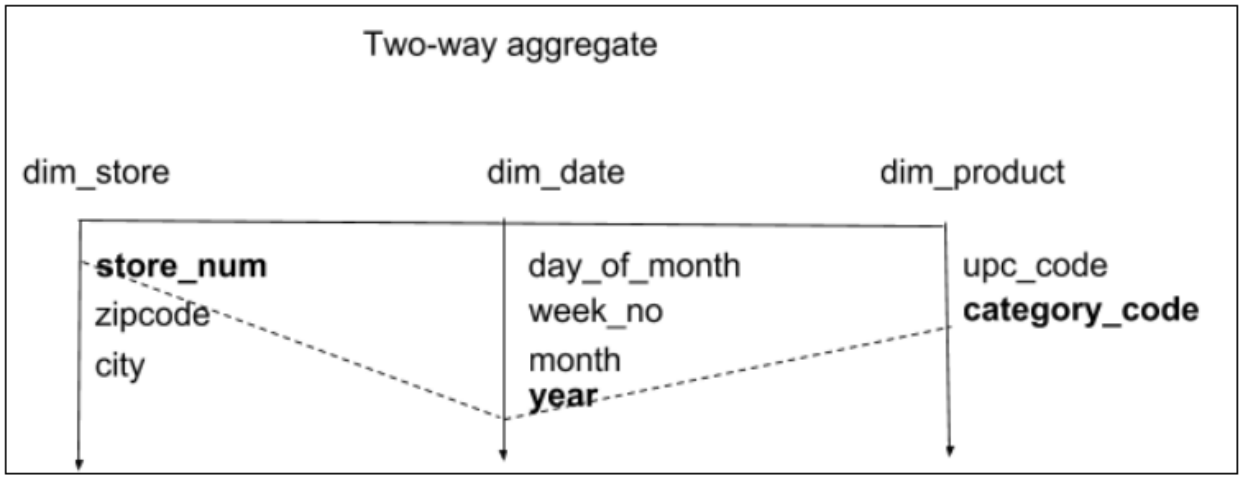


Figure 38: Two-way aggregate Business Question 1

Tables	Columns	Primary Keys (Dimension table)	Foreign Keys (Fact table)
dim_store	store_num	store_key	store_key
dim_date	week_num	date_key	date_key
	year		
	event_desc		
dim_product	category_code	product_key	product_key
fact_product_sales	product_category_sales		

Table 13: Schema for Business Question 1

2. What is the impact of age-wise regional demography on the sales of pharmacy products?

Justification: If we look at the prepared schema and this business question, then we see that the requirement demands to have pharmacy products sale based on store taking the demography into consideration. We prepared the *product_sales* schema using date, product, and store as dimensions and one fact table names *fact_product_sales*. If we look at the level of granularity needed for the product dimension for this question that total sales is needed for different stores based on pharmacy as a category as a whole. So the products are summarized as *category_code* in the product dimension table. Next we are considering the

sales on a yearly basis for the entire life of DFF, so it becomes crucial to have aggregation for date and have "*year*" as an attribute. This will help to easily look at the required data when the query is performed. Next, up for calculating the total pharmacy product sales, we included that as a metric named *product_category_sales* in the fact table since it is a measure and is an aggregated value for various years for different dates. *%population_below_age9* and *%populaation_above_age60* are also measured and required to find the sales based on demography for a store and hence included in the fact table.

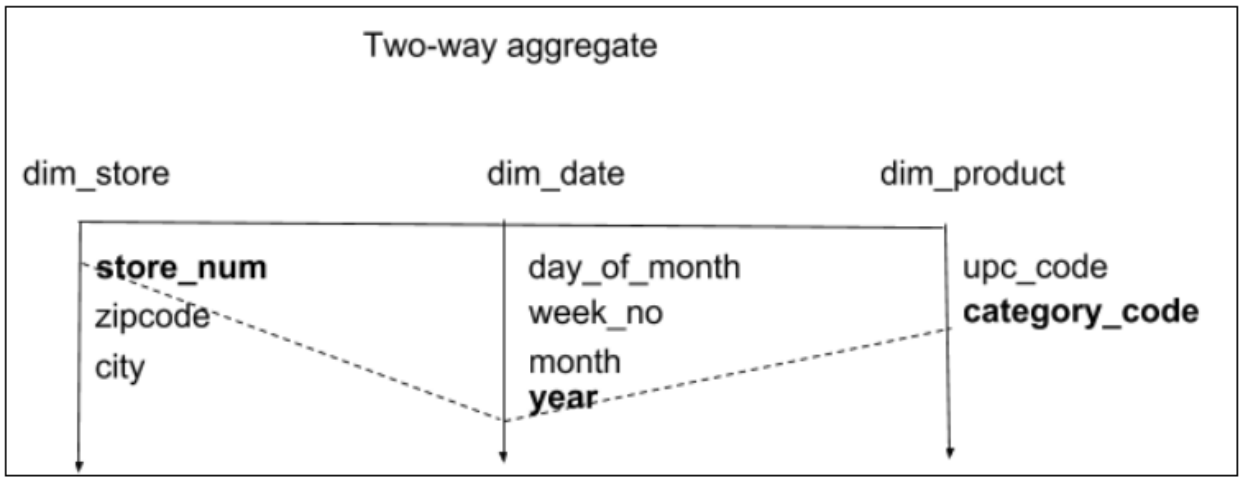


Figure 39: Two-way aggregate Business Question 2

The schema for this business question is as follows:

Tables	Columns	Primary Keys (Dimension table)	Foreign Keys (Fact table)
dim_date dim_product dim_store fact_product_sales	Year store_num category_code product_category_sales	date_key product_key store_key	date_key product_key store_key

Table 14: Schema for Business Question 2

3. What is the effect of coupon promotions on the sale of different products store-wise? Do coupons impact product sales?

Justification: For knowing the coupon sales we will again use the same schema called *product_sales* using the dimensions date, store and product. A fact table called *fact_product_sales* is created which has the total coupon sales as *coupon_sales* and the product sales based on categories. The coupon sales are found out based on different stores and on yearly basis. Hence the data is summarized on yearly basis and the products are also aggregated category wise to have required level of granularity. We will also need *product_category_sales* for this requirement to differentiate between the coupon and non-coupon sales and compare the two to know the actual values.

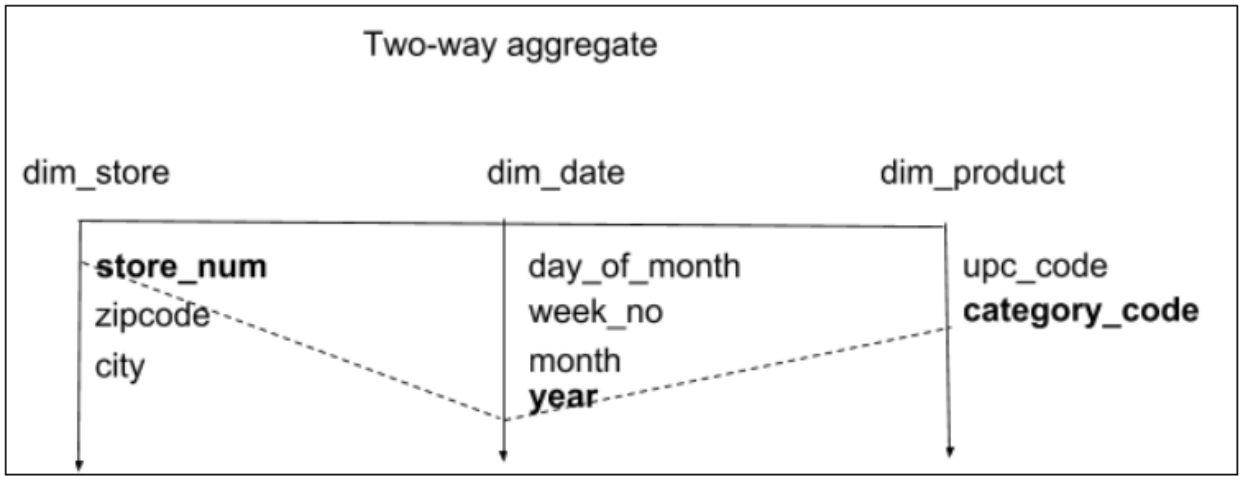


Figure 40: Two-way aggreagte for Business Question 3

The schema for this business question is as follows:

Tables	Columns	Primary Keys (Dimension table)	Foreign Keys (Fact table)
dim_date dim_product dim_store fact_product_sales	Year store_num category_code Product_category_sales Coupon_sales population%_below_age9 population%_above_age60	date_key product_key store_key	date_key product_key store_key

Table 15: Schema for Business Question 3

4. What is the customer increase or decrease over years based on the store?

Justification: This business question can be addressed by product sales data mart. For this, a attribute for *year* in *dim_date* dimension table and *store_num* in *dim_store* dimension table are created and analyzed to get the *customer_count* in the *fact_product_sales* fact table. This uses one-way aggregate. This makes it possible to make the customer count data available to the end-user for every date in each store and thus, we have a way of tracking the corresponding transactions taking place. Further, we also have day_of_month, week_no, month and year in dim_date and zipcode and city in dim_store to maintain the level of granularity and have data at the atomic level.

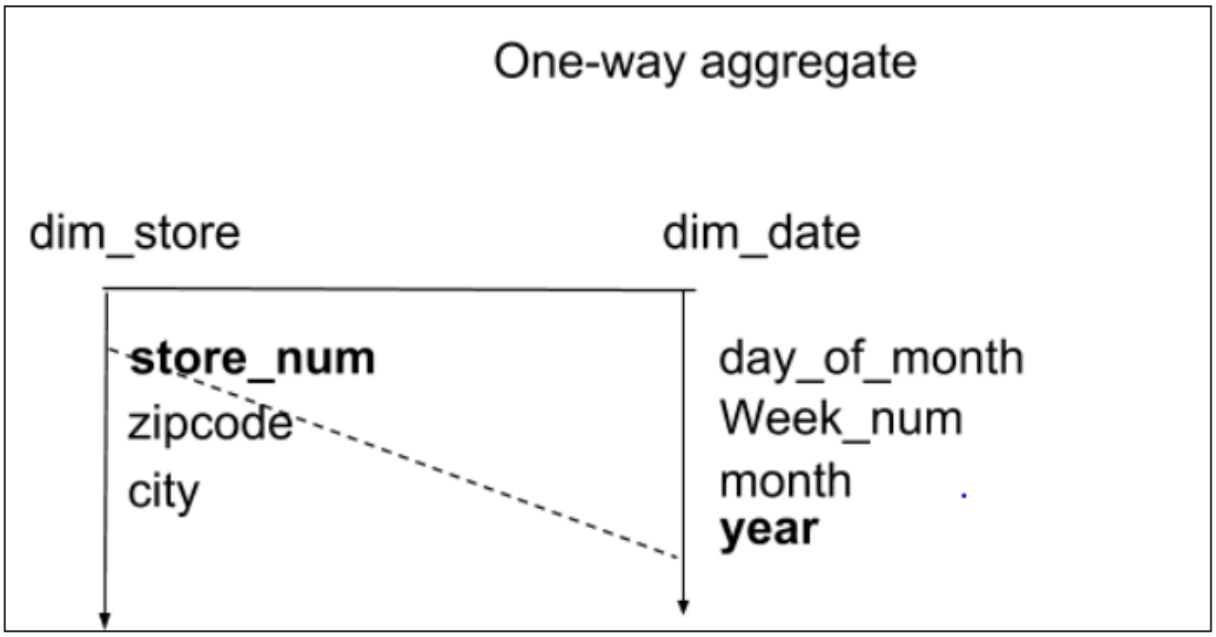


Figure 41: One-way aggregate for Business Question 4

The schema for this business question is as follows-

Tables	Columns	Primary Keys (Dimension table)	Foreign Keys (Fact table)
dim_store dim_date fact_product_sales	store_num year customer_count	store_key date_key	store_key date_key

Table 16: Schema for Business Question 4

5. How are the product category profits changing in every store over the years? What are the product categories with the highest and least profits?

Justification: This business question can be answered by making use of the product sales data mart. It is plotted based on each product category and the subsequent change in profits over the years are viewed. Thus, the *store_num* information is loaded to the *dim_store* dimension table, the *year* data to the *dim_date* dimension table and further the *category_code* to the *dim_product* dimension table is loaded and thus, this will subsequently will help us in

assessing this business question. This question uses a two-way aggregate. The *fact_Product_Sales* fact table with *product_category_profit* measure gives us all the detail related to product category profits every year. This data is aggregated from the profits for all products in a particular category.

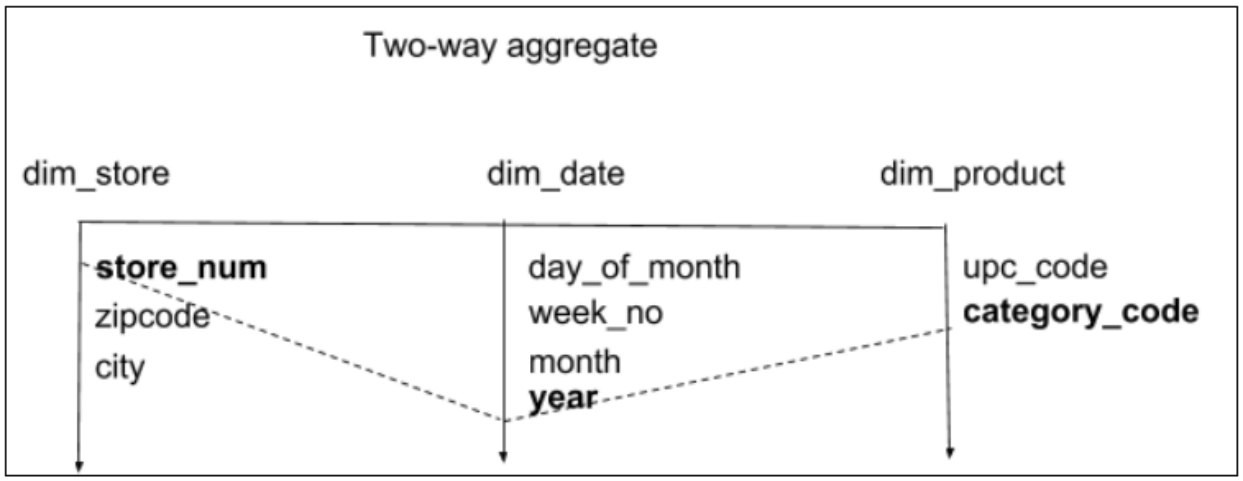


Figure 42: Two-way aggregate for Business Question 5

Tables	Columns	Primary Keys (Dimension table)	Foreign Keys (Fact table)
dim_store	store_num	store_key	store_key
dim_date	year	date_key	date_key
dim_product	category_code	product_key	product_key
fact_product_sales	product_category_profit		

Table 17: Schema for Business Question 5

9. Physical Design Plan

Physical design plan would enable improving performance and data management. The physical design consists of seven steps which include setting standards, creating an aggregate plan, partitioning data, establishing clustering options, deciding indexing strategy, deciding storage structures and designing complete physical model. To address these, we are designing a data aggregate plan, indexing plan, data standardization plan, and storage plan.

9.1 Data Aggregate Plan

As most of the business questions for the data warehouse are based on the summarization of the available data, it becomes crucial to decide on the level of summarization data needs. Almost all questions need to look for data either on a weekly or yearly basis, the attributes included for the dimensional model have been designed so. Let's say that the sales trend for Thanksgiving week each year needs to be determined. In this case, we are examining weekly data for every year, hence there is no need to include daily data. If we look at the requirement definition again then we will be able to conclude the actual level of summarization and hierarchies needed for the data warehouse so that it will easy for the end user to perform queries and carry out analysis. Creating aggregates is a great way to improve performance.

For example, if we consider one of the requirements which say to find the effect of coupons on sales of different products based on store, for this the data will be taken on a yearly basis for different stores for products having coupons. Here for this particular requirement, we need to sum all the coupon sales and total product sales for all years to be included in the dimension table. Originally the data was present for individual dates but the aggregated data was created by summing up the sales for all dates in a year. So this is a classic example of how the aggregation is done to meet the needs of the warehouse. Here we can also deduce that since the date wise data will never be looked up, including that in a warehouse will result in unnecessary storage and every time the query will run it will have to go through the number of hierarchies that are not actually required. So deciding on summarization becomes utmost important in designing the warehouse.

Dimension/Fact Tables	Summarization Level	Description
dim_date	Attributes such as week_no, month and year contain the aggregated values.	*Week_no* lies higher up in the hierarchy when compared to *day_of_week* and *day_of_month* and contains 7 days in it. *Month* is a higher level of summarization for *week_no* and contain data for specific months. *Year* is further higher up in the hierarchy to have more summarized data for months.
dim_product	Attributes that contain the aggregated values are *category_code* which is a sum of, or contain, all the products in that particular category.	It is a table because of summarization for *upc_code.*
fact_product_sales	*Product_category_sales* contains the aggregated values which contain the sales data for a particular product category and not by individual product.	This is a higher level of summarization of individual product sales. We have not included that in the fact table because it is not required as per our requirements.

Table 18: Data Aggregate Plan

9.2 Indexing Plan

Indexing in data warehousing is aimed to reduce the time in seeing results of the query and making sure the results are available to the end-user in an optimum manner. To ensure efficiency in this, we need to find the perfect balance between the number of indexes we utilize. This is because if we have few indexes, data loading is quick but this might

compromise the response time to the query. Whereas, having a lot of indexes means the data loading is slow and more of storage space is required but the response rate is faster. Also, the plan indicates the indexes in the sequence they are created. Many indexes also monitor data warehouse for some time and come up with the index plan. However, we would not be following this approach. Also, data warehouse data is read-only and thus, we can create indexes without the risk of incurring costs to modify index files.

The factors we considered for the Dominick's' Fine Foods project is an archival type data warehouse since we are not dealing with real-time data. Furthermore, the querying will be ad hoc and we have taken into consideration the size of the dimension and the fact tables. Out of the three indexing types- binary tree(B-Tree), bit-mapped, clustered indexing, considering our requirements we would be using the B-Tree approach would be the most efficient. Indexing is done by identifying the common queries and selecting the columns that are frequently used to constrain. Such columns are used for indexing. It stores hierarchical index records. We will be indexing on the surrogate or primary key, and the foreign keys to start off and we are using single column indexing. DBMS's automatically create indexes on the primary key because of its data retrieval speed, ease of maintenance, and simplicity. For the dimensions i.e, *date, product and store*, the indexes are their surrogate keys i.e, *date_key, product_key, store_key*, respectively. Now, the fact table's primary key is a concatenation of all the dimension tables of surrogate keys. Thus, a B-index tree for the same would be created by concatenating all the primary keys or "full" primary key in a specific order.

The following diagram shows how store no. can be stored in binary-tree for efficient retrieval-

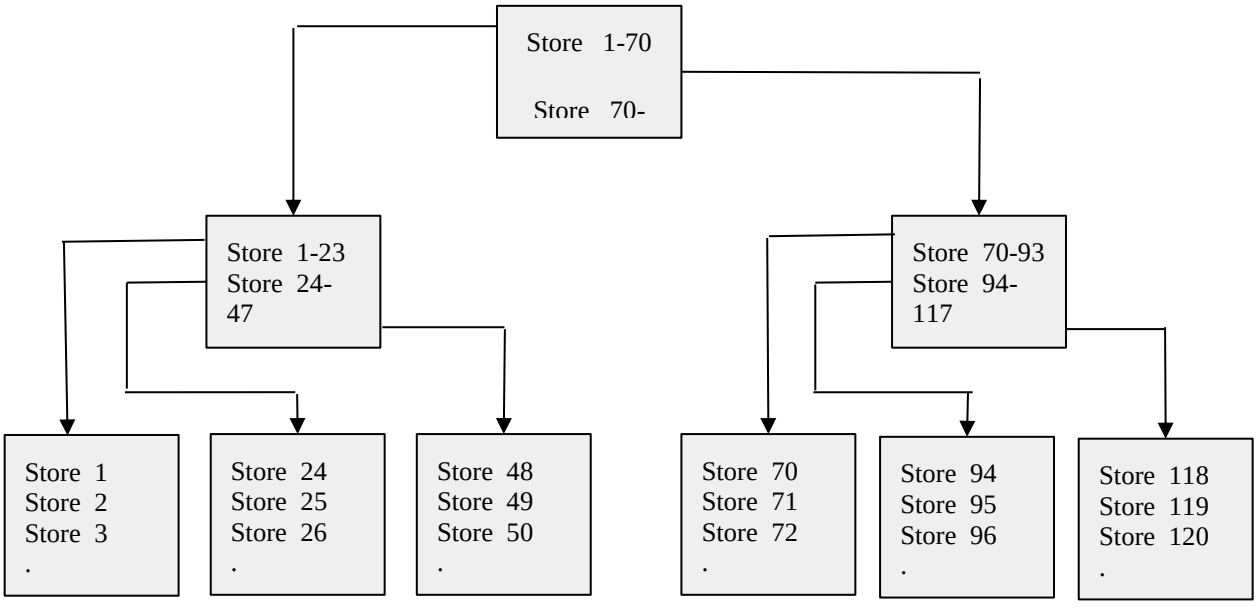


Figure 43: Indexing Plan

Consequently, using B-indexing will catalyze the query-centric retrieval process in data warehousing. Index files contain the primary keys. It has the value of the indexes and a pointer to the physical address where data for that index (primary key) is stored. Thus, it simplifies the querying process as the system creates part of the binary tree and swaps it as required, for a data set as big as ours and hence, improves the process.

9.3 Data Standardization Plan

Data Standardization plan is essential to ensure data quality as well as. As data in a data warehouse is collected from multiple sources, lacking standardization can lead to data inconsistency. In Dominick's Fine Foods data is collected from multiple sources and formats. It is important to spend time in hunting, gathering and standardizing data correctly for value addition. The most essential step is to name the fields and tables consistently.

	Naming Convention	Example
Surrogate Key	*xxx*_key Where xxx is the dimension name	product_key date_key store_key
Dimension Table	dim_*xxx* Where xxx is the name of the dimension	dim_product dim_date dim_store
Fact Table	fact_*xxx* Where xxx is the name of the fact table	fact_product_sales fact_customer_visits
Staging Table	stg_xxx	stg_demographics

	Where xxx is the name of the file extracted in the staging area	stg_custcount stg_store

Table 19: Data Standardization Plan

9.4 Storage Plan

Storage plan is important to make data rapidly accessible and enable quick analysis. As data warehouse generally has large volume is data, the efficiency of data retrieval is affected based on how data is stored.

The storage requirement for Staging:

	Initial estimate of rows	Average length of rows	Monthly increase in number of rows (anticipated)	Estimated current table size in MB	Calculated size in 6 months	Calculated size in 12 months
stg_demographics	109	4800	~1	~0.5 MB	~0.5 MB	~0.5 MB
stg_movement	~31457K	25	~34800	~750 MB	~755 MB	~ 760 MB
stg_week_decode	~400	30	~4	~0.1 MB	~0.1 MB	~0.15 MB
stg_product	~18200	45	~100	~ 1.2 MB	~ 1.4 MB	~ 1.6 MB
stg_stores	116	40	~1	~ 0.5 MB	~0.6 MB	~ 0.6 MB
stg_custco	~ 327 K	300	~3480	~89 MB	~92 MB	~95 MB

unt						

Table 20: Data Storage Plan for Staging

The storage requirement for OLAP tables:

	Initial estimate of rows	Average length of rows (bytes)	Monthly increase in number of rows (anticipated)	Estimated current table size in MB	Calculated size in 6 months	Calculated size in 12 months
dim_product	~18200	45	~100	~ 1 MB	~ 1.1 MB	~ 1.2 MB
dim_store	~130	35	~2	~ 0.2 MB	~ 0.3 MB	~ 0.4 MB
dim_date	~2800	35	~30	~ 0.5 MB	~ 0.6 MB	~ 0.7 MB
fact_product_sales	~11773K	35	~6000	~ 390 MB	~ 400 MB	~ 410 MB

Table 21: Data Storage Plan for Dimension and Fact

10. ETL Plan

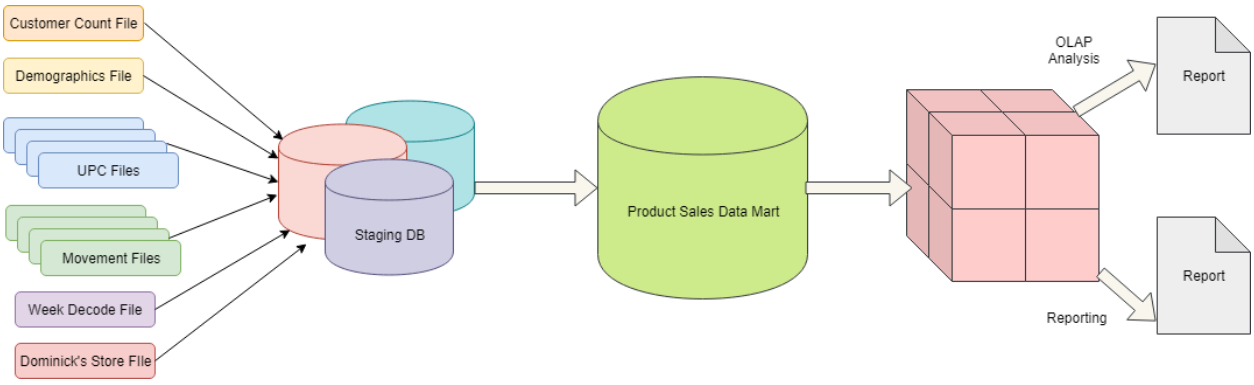


Figure 44: ETL Plan

The area of data acquisition, data storage is covered by data extraction, data transformation and data loading. Firstly, the extraction from source systems encompasses the data extraction. In DFF's case our sources were the following files- customer count, demographics, UPC, movement, week decode, and the store file. These source files were extracted from csv, txt, the DFF manual to the staging database. Next, functional and procedure changes to the source data are applied. These transformations are to change the data in the appropriate format and structures for storage in the data warehouse database.This is then followed by moving or loading the data into the data warehouse repository. This is when our product sales data mart is populated i.e. the dimension and fact tables. The OLAP data is stored in a star schema and follows Kimball's data mart approach. After successfully performing the extract, transform and load the OLAP cube is then further used for analysis and reporting. This will in turn be used to answer the business questions for Dominick's.

10.1 Target Data

As per the Data warehouse schema, the Data mart for Product Sales requires the below target data. We require three dimensions namely: date, product, and store and a fact table for product sales.

Index	Dimension / Fact	Table Name	Attribute Name	Attribute Data Type	Example
1	Dimension	Dim_Date	date_key	Integer	1
			date	Integer	29
			week_num	Integer	10

			month	Varchar	January
			year	Integer	1989
			holiday_flag	Bit	0 or 1
			event_desc	Varchar	Thanksgiving
2	Dimension	Dim_Store	store_key	Integer	2
			store_num	Integer	28
			zipcode	Varchar	77801
			city	Varchar	Chicago
3	Dimension	Dim_Product	product_key	Integer	3
			upc_code	Varchar	1192603016
			product_desc	Varchar	CAFFEDRINE CAPLETS 1
			category_code	Varchar	953
			category _name	Varchar	Analgesics
4	Fact	Fact_Product_ Sales	date_key	Integer	1
			product_key	Integer	3
			store_key	Integer	2
			product_sales	Float	36.2
			product_categor y_sales	Float	378.54
			product_profit	Float	27.53

			product_category_profit	Float	28.67
			coupon_sales	Float	35.98
			population%_below_age9	Float	
			population%_above_age60	Float	
			customer_count	Float	23

Table 22: Target Data

10.2 Data Sources

The data sources for Dominick's Fine Foods includes all the store data, customer data and sales data provided by the University of Chicago Booth School of Business. It also includes the data included in Dominick's Manual.

Source File	Data in Source and Description
UPCxxx.csv	This contains description of UPC's belonging to each product category. Also, the file names are in the form of upcxxx. The information has UPC name, commodity code, item code, description, item was drop-shipped or warehoused, etc. Here, xxx denotes the three-letter acronym for the category.
Wxxx.csv	It has the sales information at every store level, for each UPC category, on a weekly basis. Information such as price of a product, units sold, profits gained on that particular product, etc. The xxx in the filename defines the three- letter acronym for the respective category.
Store.csv	The city, price tier, zone, zip code and address information for each store are listed in the Dominick's manual.
WeekDecode.csv	This SAS file contains a week variable that has been coded to give us the week corresponding to which a sales data is

	recorded
CCount.csv	It contains information about in-store traffic. The data pertaining to the number of customers making purchases on a daily basis, specific to the store are present in the file. Total sales of the products and total coupons redeemed by the store are also listed.
Category.csv	It includes the UPC's and their corresponding descriptions and size for a particular product category. The acronym for the product category, last update details, file size, number of observations, first week in the file, last week in the file, no. of UPC's are also mentioned.
Demo.csv	It contains detailed information about the store-specific demographics based on the United States Government consensus for the Chicago Metropolitan area. Market Metrics processed the data to generate demographic profiles for each DFF store. Demographics are divided on the basis of age, ethnicity, level of income, household size, etc.

Table 23: Source Data

10.3 Data Mapping Table

The data mappings are prepared for all Dominick's file from source to staging database and staging database to the data warehouse. The mapping will include all the naming conventions and data type changes done to the source data.

DATA MAPPING TABLE

Mapping Index	Source	Source Attribute Name	Source Attribute Type	Staging Table	Staging Attribute Name	Staging Attribute Type	Business Rule	Target Warehouse Table	Target Attribute Name	Target Attribute Type	Business Rule
1	Multiple UPC Flat File Source	UPCXXX.csv	Flat Files (csv)	stg_product	COM_CODE	varchar(50)		dim_product	category_code	varchar(10)	
2					UPC	varchar(50)		dim_product	upc_code	varchar(15)	
3					DESCRIP	varchar(50)		dim_product	product_desc	varchar(50)	
4					SIZE	varchar(50)					
5					CASE	varchar(50)					
6					NITEM	varchar(50)					
7					FILENAME	varchar(50)	A Filename attribute has been added derived from the name of the file.				derived(map filename) used to compare with category upc filename
8	Domnick's store data from manual	store.csv	Flat File Source(csv)	stg_store	STORE	int		dim_store	store_num	varchar(10)	
9					CITY	varchar(50)		dim_store	city	varchar(20)	
10					PRICETIER	varchar(50)					
11					ZONE	varchar(50)					
12					ZIPCODE	varchar(50)		dim_store	zipcode	varchar(10)	
13					ADDRESS	varchar(50)					
14	Domnick's week decode table from manual	weekdecode.csv	Flat File Source(csv)	stg_weekdecode	Week#	int	Data conversion is applied	dim_date	week_num	varchar(3)	
15					Start	date					
16					End	date					
17					Special Events	varchar(50)					
18	Domnick's category table from manual	category.csv	Flat Files (csv)	stg_category	CATEGORY	varchar(50)		dim_product	category_name	varchar(25)	Mapped
19					UPC	varchar(50)					
20					MOVEMENT	varchar(50)					
21	Customer count data from flat file	ccount.csv	Flat Files (csv)	stg_custcount	STORE	varchar(50)	Stg_transformed_custcount table has been used as a intermediatary table for our fact_product_sales table. The stg_custcount talbe unpivoted and then split conditionally for negatve values. Data conversion followd by derived function is applied ro get resultant transformaned data				
22					DATE	varchar(50)					
23					GROCCOUP	varchar(50)		fact_product_sales			
24					MEATCOUP	varchar(50)					
25					FISHCOUP	varchar(50)					
26					PROMCOUP	varchar(50)					
27					PRODCOUP	varchar(50)					
28					BULKCOUP	varchar(50)					
29					SALCOUP	varchar(50)					
30					FLORCOUP	varchar(50)					
31					DELICOUP	varchar(50)					
32					PHARCOUP	varchar(50)					
33					GMCOUP	varchar(50)					
34					VIDCOUP	varchar(50)					
35					MANCOUP	varchar(50)					
36					FTGCCOUP	varchar(50)					
37					FTGICOUP	varchar(50)					
38					DAIRCOUP	varchar(50)					
39					FROZCOUP	varchar(50)					
40					HABACOUP	varchar(50)					

41					PHOTCOUP	varchar(50)					
42					COSMCOUP	varchar(50)					
43					BAKCOUP	varchar(50)					
44					LIQCOUP	varchar(50)			coupon_sales	float	Aggregated based on week
45					WEEK	varchar(50)					
46					CUSTCOUN	varchar(50)		fact_product_sales	customer_count	int	
47					NAME	varchar(50)		fact_product			
48	Demographics data from flat file	demo.csv	Flat Files (csv)		AGE9	varchar(50)		fact_product_sales	population_below_age9	float	
49				stg_demographics	AGE60	varchar(50)		fact_product_sales	population_above_age60	float	
50					STORE	varchar(50)					
51					UPC	varchar(50)					
52					WEEK	varchar(50)					
53					MOVE	varchar(50)					
54					QTY	varchar(50)					
55	Movement data from flat file	wxxx.csv	Flat Files (csv)	stg_movement	PRICE	varchar(50)		fact_product_sales	product_sales	float	Calculated using formula: Price * Move/Qty
								fact_product_sales	product_category_sales	float	Aggregated based on category code
56								fact_product_sales	product_profit	float	
					PROFIT	varchar(50)		fact_product_sales	product_category_profit	float	Aggregated based on category code
57					DateKey	int					
58					DateFull	date					Check whether between start_date and end_date. Then, the corresponding week is picked up from weekdecode
59					DayOfMonth	int		dim_date	date	int	
60	N/A	Autogenerated using script for Date	N/A	stg_date	Month	int		dim_date	month	varchar(10)	
61					Year	int		dim_date	year	int	
62					IsWorkDayKey	int					
63					IsPublicHolidayKey	int		dim_date	holiday_flag	bit	Set to 1 if IsWorkDayKey = 1 or IsPublicHolidayKey = 1
63					IsPublicHolidayDescription	varchar(50)		dim_date	event_desc	varchar(25)	
64								dim_date	date_key	int	surrogate key
65								dim_store	store_key	int	surrogate key
66								dim_product	product_key	int	surrogate key

10.4 Data Extraction Rules

For extracting the data into the staging area we had to collect data from various sources such as csv files, text files, and data from the Dominick's data manual. We extracted the data from from the flat file sources, This is the very first process that comes into picture when working with staging database. This is also very crucial stage since all the further processes are dependent on the correct extraction of files.
The steps included are as follows:

- Identify the source system: Here our source system are flat files and they need to be extracted to staging tables. Source data is looked for null values or any junk value.
- Method of Extraction: We used SSIS tool to extract all the data. The data is extracted with the help of a source area that specifies the use of csv file and then mapped to the staging area table.
- Extraction Frequency: We are extracting data just once as an initial extract and it will be further extracted on weekly basis to be furthered loaded to data warehouse.
- Time Window: Whole extraction needs to be completed within a certain time frame which has been fixed in the script file and scheduling scripts. The time is usually kept at a maximum value of an hour considering the worst case scenario.
- Job sequencing: A sequence is specified for determining the execution order of the jobs, it specifies which job is dependent on other jobs to complete. For example, a table using another table's key as foreign key, required that table to be extracted first and then use its keys.
- Exception Handling: There are times when there is some input that needs special care while extraction. These columns are either type casted or tried to be loaded like that only.

10.5 Data Transformation Rules

It consists of rules to prepare the data to be further loaded to the data warehouse systems. It consists of a series of functions to be applied to the extracted data. Data cleaning is the most important aspect of transforming the data. Data transformation rules are mainly considered for data quality purposes in a data warehouse. It includes the following:

- Data validation rules: This includes basic functions such as validating the data based on capturing the null values or may be other filters as required, this will throw out the erroneous data. This also includes functions such as selection, splitting, conversion, summarization and other basic transformations.
- Data cleansing rules: Here the values that contain dirty or not required data are removed and the extracted data becomes free from any toxicity it might have.
- Data deduplication rules: This involves only single data for one customer and pointing all the records for that customer in the source system.
- Data consolidation rules: Data in ETL comes from various sources and hence need to be think of before integrating them. This requires understanding the variation of source systems and planning what needs to be done before integrating. All relevant source data are integrated into coherent data structures. In our project, we had to combine flat files with OLE DB tables, so properly integrate them appropriately so that we have a consolidated view of data. There are various problems associated with like Entity Identification Problem, Multiple Source Problem, Dimension attribute transformation, etc.

10.6 Data Cleansing Rules

This deals with removing records with junk values. Data is cleansed for accuracy, completeness, consistency and uniformity. The cleansing rule can be one of the following:

- Eliminating missing fields: This includes removing null values, or values that have any junk values like periods(.) or asterisk (*). Data transformation tools make the tasks easy when it comes to cleaning data due to the availability of various options for cleaning.
- Modifying data type: At times values with different data types need to be matched and compared, this involves data type conversion. There are various functions available for type casting in SSIS. One of the stage available to do this task is Derived column. This helps to perform various operations when it comes to conversion of data type.
- Setting default value: It becomes necessary to provide a default value to certain columns so that it has a sensible meaning to ti even when it is left blank. It provides some legit meanings to a column.

10.7 Plan for aggregate table

Aggregate table is more or less a component of a fact table and dimension table. Aggregation is done to achieve data at different level of granularity. In our project we have created aggregates for all the dimension tables and fact table. Let's say that date dimension needs to be created, initially there is lowest level of granularity which is individual date. It is further aggregated to create week, month and then year. Same is the case with product dimension in terms of individual product and category and store dimension which has zipcode and city.

- First we look at the business questions to see what granularity of data is required.
- This will help us to look out for columns that are required to be aggregated.
- After the data has been extracted in the staging area, the columns are aggregated based on the requirements.
- Aggregate stage is used to create summarized data and then loaded to the warehouse tables.
- Data is required to achieve level of granularity.

10.8 Organization of Data Staging area

Data staging area is created as a database with the name *group10_602_stagingdb*. All the data sources from the excel files and text files are extracted into the staging area.

1. **Stg_demographics:** Demographics data is extracted into table stg_demographics from demo.csv.

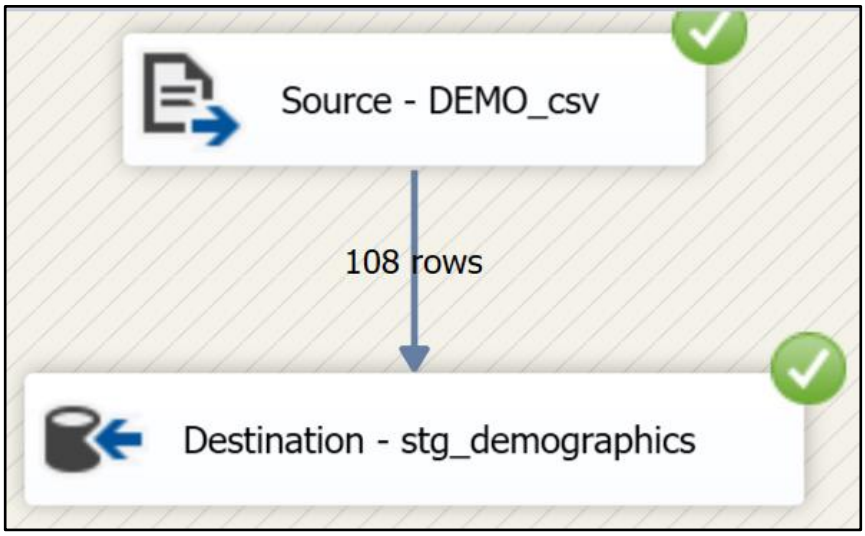
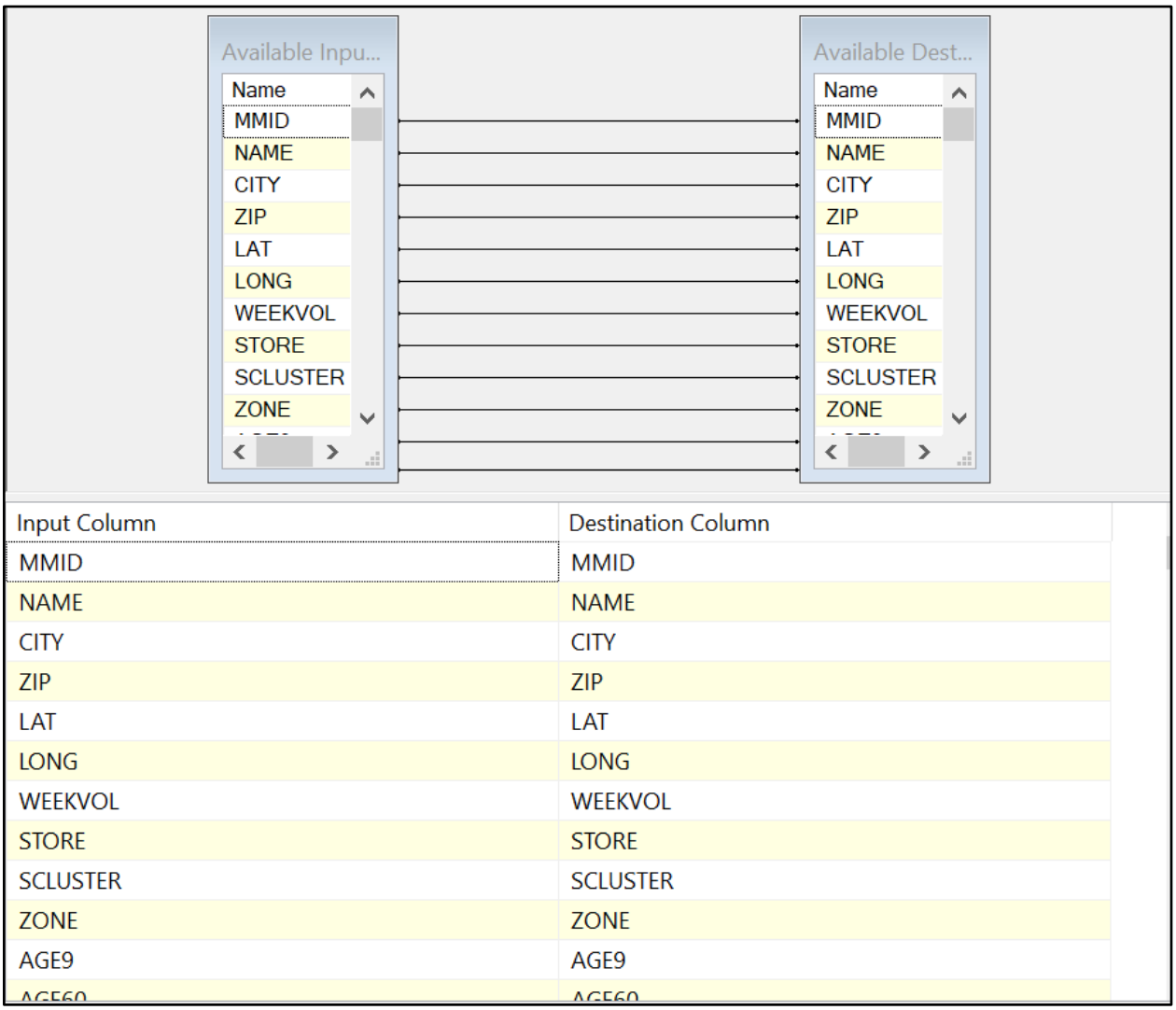


Figure 45: Demographics Staging Flow



Input Column	Destination Column
MMID	MMID
NAME	NAME
CITY	CITY
ZIP	ZIP
LAT	LAT
LONG	LONG
WEEKVOL	WEEKVOL
STORE	STORE
SCLUSTER	SCLUSTER
ZONE	ZONE
AGE9	AGE9
AGE60	AGE60

Figure 46: Data Mapping for Demographics

	MMID	NAME	CITY	ZIP	LAT	LONG	WEEKVOL	STORE	SCLUSTER	ZONE	AGE9	AGE60
1	16892	DOMINICKS 2	RIVER FOREST	60305	419081	878131	350	2	C	1	0.117508576	0.232864734
2	16893	DOMINICKS 4	PARK RIDGE	60068	420392	878425	300	4	A	2	0.0950895057	0.26202989
3	16894	DOMINICKS 5	PALATINE	60067	421203	880431	550	5	D	2	0.1414334827	0.1173680317
4	16895	DOMINICKS 8	OAK LAWN	60453	417331	877436	600	8	C	5	0.123155416	0.2523940345
5	16896	DOMINICKS 9	MORTON GROVE	60053	420411	877994	450	9	A	2	0.1035030974	0.2691190176
6	16898	DOMINICKS 12	CHICAGO	60660	419928	876592	450	12	B	7	0.1056967397	0.178341405
7	16899	DOMINICKS 14	GLENVIEW	60025	420733	877994	400	14	A	1	0.129589372	0.2139492754
8	16901	DOMINICKS 18	RIVER GROVE	60171	419364	878331	600	18	A	5	0.1100949839	0.2723133684
9	16903	DOMINICKS 21	HANOVER PARK	60103	420058	881411	500	21	D	6	0.1759263459	0.0668964579
10	16905	DOMINICKS 28	MOUNT PROSP...	60056	420686	879208	275	28	A	2	0.1288795371	0.2133087849
11	16906	DOMINICKS 32	PARK RIDGE	60068	419872	878378	575	32	C	1	0.0990606319	0.2549530316
12	16907	DOMINICKS 33	CHICAGO	60657	419386	876447	300	33	B	7	0.0460709172	0.1341699655
13	16909	DOMINICKS 40	BRIDGEVIEW	60455	417317	877969	500	40	D	6	0.1336846485	0.1818518005
14	16912	DOMINICKS 44	WESTERN SPR...	60558	418033	878903	325	44	A	2	0.1448834853	0.1909827761
15	16913	DOMINICKS 45	WHEELING	60090	421403	879300	300	45	D	2	0.1467187625	0.1288573479

Figure 47: Table for stg_demographics

2. Stg_custcount: Customer Count data from the ccount.csv is extracted into the stg_custcount.

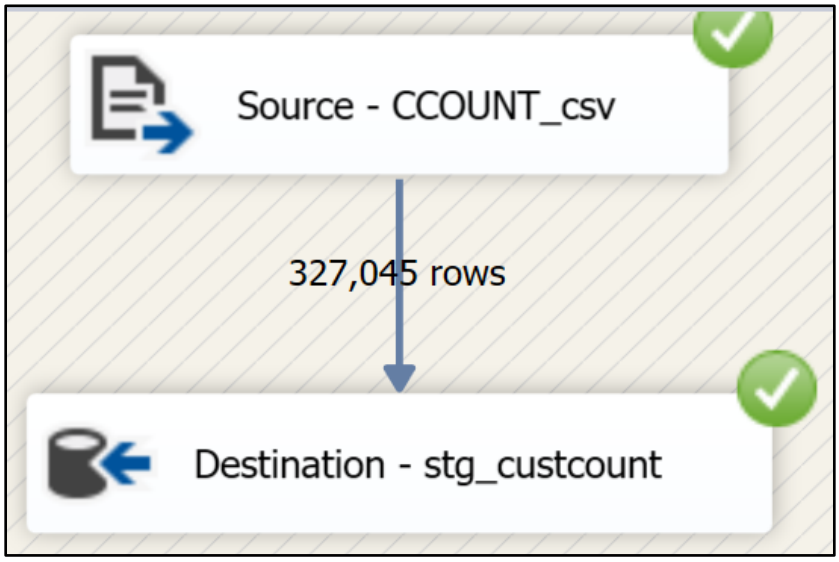
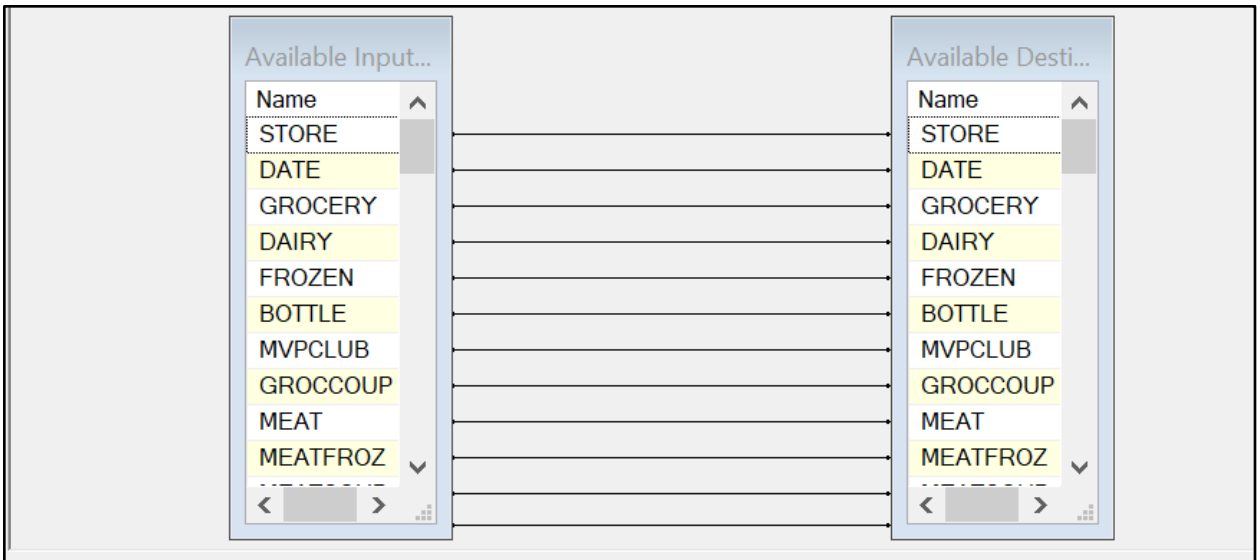


Figure 48: CCount Data Flow



Input Column	Destination Column
STORE	STORE
DATE	DATE
GROCERY	GROCERY
DAIRY	DAIRY
FROZEN	FROZEN
BOTTLE	BOTTLE
MVPCLUB	MVPCLUB
GROCCOUP	GROCCOUP
MEAT	MEAT
MEATFROZ	MEATFROZ
MEATCOUP	MEATCOUP

Figure 49: Data Mapping for CCount

	STORE	DATE	GROCERY	DAIRY	FROZEN	BOTTLE	MVPCLUB	MEAT	MEATFROZ	MEATCOUP	FISH	FISHCOUP
1	307	910426	47581.01	9057.78	7151.21	0	0	10342.07	1128.1	0	552.29	0
2	309	910527	35788.06	6518.17	5226.24	0	0	7635.04	1101.73	0	502.06	0
3	304	890803	34080.84	4761.27	3737.85	0	0	7289.86	1120.49	0	644.1	0
4	302	910517	59411.19	10505.36	9330.02	0	0	9909.6	1484.02	0	1104.19	0
5	303	900216	50066.33	10467.18	6141.67	0	0	12226.79	1647.38	0	1903.33	0
6	303	900311	56717.86	13130.54	7202.94	0	0	12508.43	1641.5	0	1710.23	0
7	303	910118	56342.43	10746.75	5615.35	0	0	11937.86	1624.93	0	2562.79	0
8	303	910309	83550.59	16745.71	8576.32	0	0	22182.48	2407.58	0	2583.63	0
9	302	900101	20703.25	5037.88	3415.91	0	0	3973.32	920.67	0	453.8	0
10	307	890423	24749.68	4461.76	3396.55	0	0	4609.2	730.35	0	280.96	0
11	301	910316	82419.75	13682.93	12636.52	0	0	16040.14	2818.68	0	1750.17	0
12	94	961010	15957.44	3562.32	2839.81	0	48.73	4090.08	333.4	0	337.07	0
13	134	960515	12077.19	2671.42	2303.82	0	125.48	2637.92	592	-27	405.41	0

Figure 50: Table for stg_custcount

3. Stg_product: All the UPC files are extracted from the UPCxxx.csv files to stg_product using the "Multiple Flat File Source". All the 29 UPC files are extracted using the below package. Derived function is used to map the filename i.e xxx in the UPCxxx.csv filename to the destination table.

SQL Query:

```
CREATE TABLE [dbo].[stg_product](
        [COM_CODE] [varchar](50) NOT NULL,
        [UPC] [varchar](50) NOT NULL,
        [DESCRIP] [varchar](50) NULL,
        [SIZE] [varchar](50) NULL,
        [CASE] [varchar](50) NULL,
        [NITEM] [varchar](50) NULL,
        [FILENAME] [varchar](50) NOT NULL
) ON [PRIMARY]
GO
```

Derived Column is used to map the filename as:

(DT_STR,6,1252)LOWER(SUBSTRING(FileName,44,6))

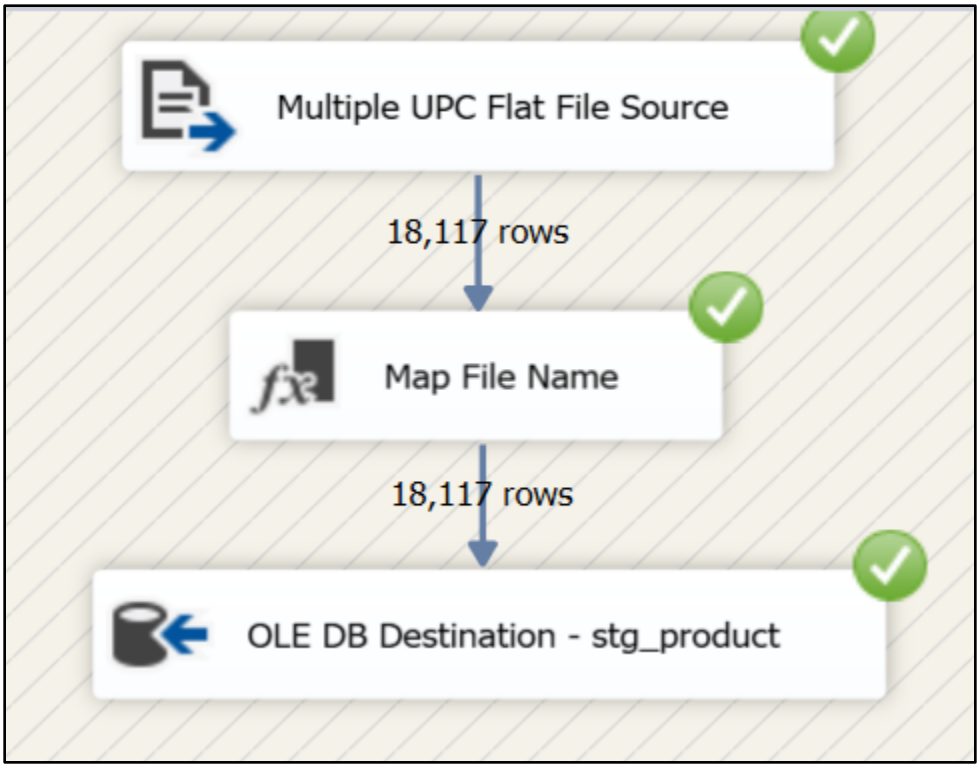
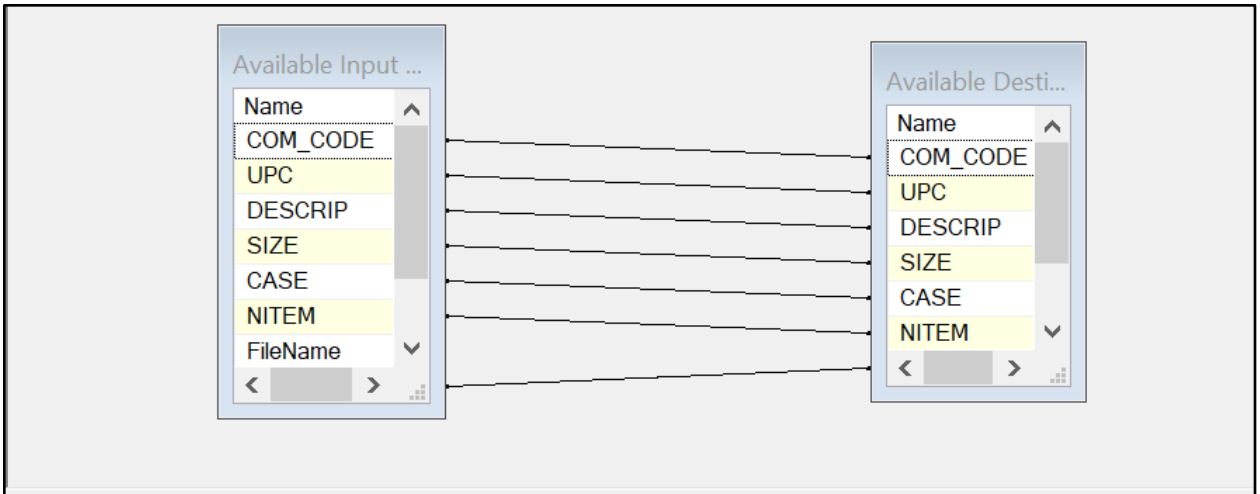


Figure 51: UPC Data Flow



Input Column	Destination Column
COM_CODE	COM_CODE
UPC	UPC
DESCRIP	DESCRIP
SIZE	SIZE
CASE	CASE
NITEM	NITEM
UPCFileName	FILENAME

Figure 52: Data Mapping for UPC

	COM_CODE	UPC	DESCRIP	SIZE	CASE	NITEM	FILENAME
1	953	1192603016	CAFFEDRINE CAPLETS 1	16 CT	6	7342431	upcana
2	953	1192662108	SLEEPINAL SOFTGEL	8 CT	6	7333311	upcana
3	953	1650001020	NERVINE TABS	30 CT	1	8430820	upcana
4	953	1650001022	NERVINE SLEEP AID	12 CT	1	8430840	upcana
5	953	1650004106	ALKA-SELTZER GOLD	20 CT	1	8430880	upcana
6	953	1650004108	ALKA-SELTZER GOLD	36 CT	1	8430900	upcana
7	953	1650004703	ALKA MINTS	30 CT	1	8430700	upcana
8	953	2140649030	LEGATRIN PM	30 CT	1	8435810	upcana
9	953	2586600493	PERCOGESIC A/F ANALG	50 CT	1	8416280	upcana
10	953	2586610493	PERCOGESIC A/F ANALG	50 CT	1	8416280	upcana
11	953	2586610501	ALEVE TABLETS	24 CT	6	6122441	upcana

Figure 53: Table for stg_product

4. Stg_store : The store data for 96 Dominick's store is extracted from the store.csv to the stg_store table.

SQL Query:

```
CREATE TABLE [dbo].[stg_store](
	[STORE] [int] NOT NULL,
	[CITY] [varchar](50) NULL,
	[PRICETIER] [varchar](50) NULL,
	[ZONE] [varchar](50) NULL,
	[ZIPCODE] [varchar](50) NULL,
	[ADDRESS] [varchar](50) NULL,
PRIMARY KEY CLUSTERED
(
	[STORE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

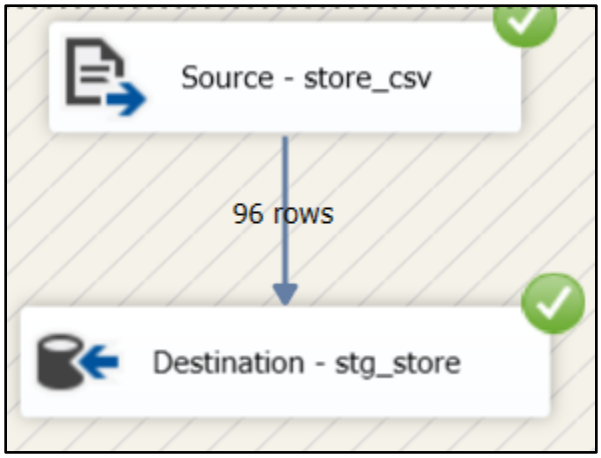
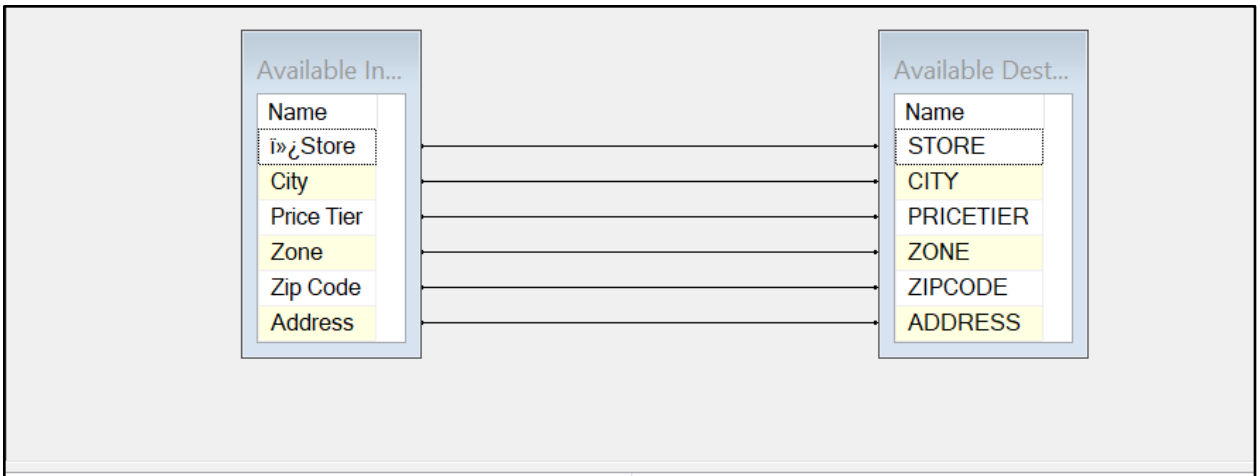


Figure 54: Store Staging Data Flow



Input Column	Destination Column
ï»¿Store	STORE
City	CITY
Price Tier	PRICETIER
Zone	ZONE
Zip Code	ZIPCODE
Address	ADDRESS

Figure 55: Data Mapping for Store

	STORE	CITY	PRICETIER	ZONE	ZIPCODE	ADDRESS
1	2	River Forest	High	1	60305	7501 W. North Ave.
2	4	Park Ridge	Medium	2	60068	Closed
3	5	Palatine	Medium	2	60067	223 Northwest HWY.
4	8	Oak Lawn	Low	5	60435	8700 S. Cicero Ave.
5	9	Morton Grove	Medium	2	60053	6931 Dempster
6	12	Chicago	High	7	60660	6009 N. Broadway Ave.
7	14	Glenview	High	1	60025	1020 Waukegan Rd.
8	18	River Grove	Low	5	60171	8355 W. Belmont Ave.
9	19	Glen Ellyn			60137	Closed
10	21	Hanover Park	CubFighter	6	60103	1440 Irving Park Rd.

Figure 56: Table for stg_store

5. Stg_category: Product category data is extracted to the database from the category.csv to the stg_category table. This will be used to map the three lettered UPC code to the product category name.

SQL Query:

```
CREATE TABLE [dbo].[stg_category](
	[CATEGORY] [varchar](50) NULL,
	[UPC] [varchar](50) NULL,
```

```
        [MOVEMENT] [varchar](50) NULL
) ON [PRIMARY]
GO
```

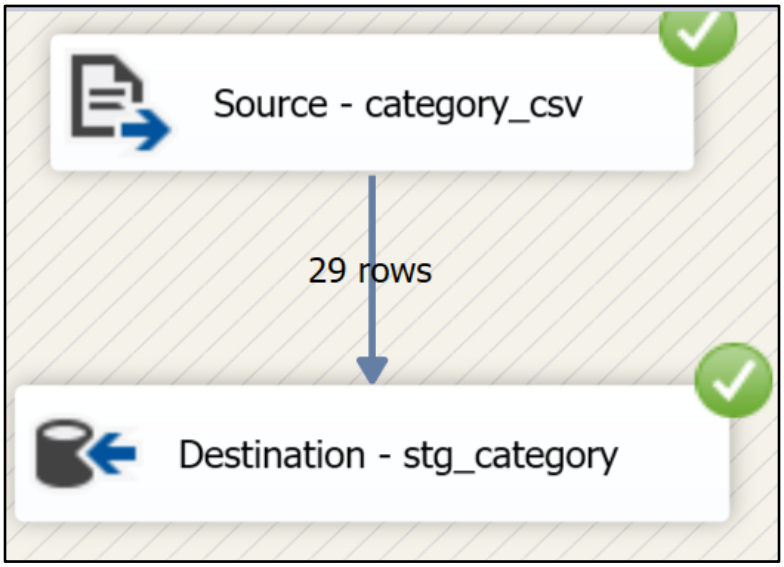


Figure 57: Category Staging Data Flow

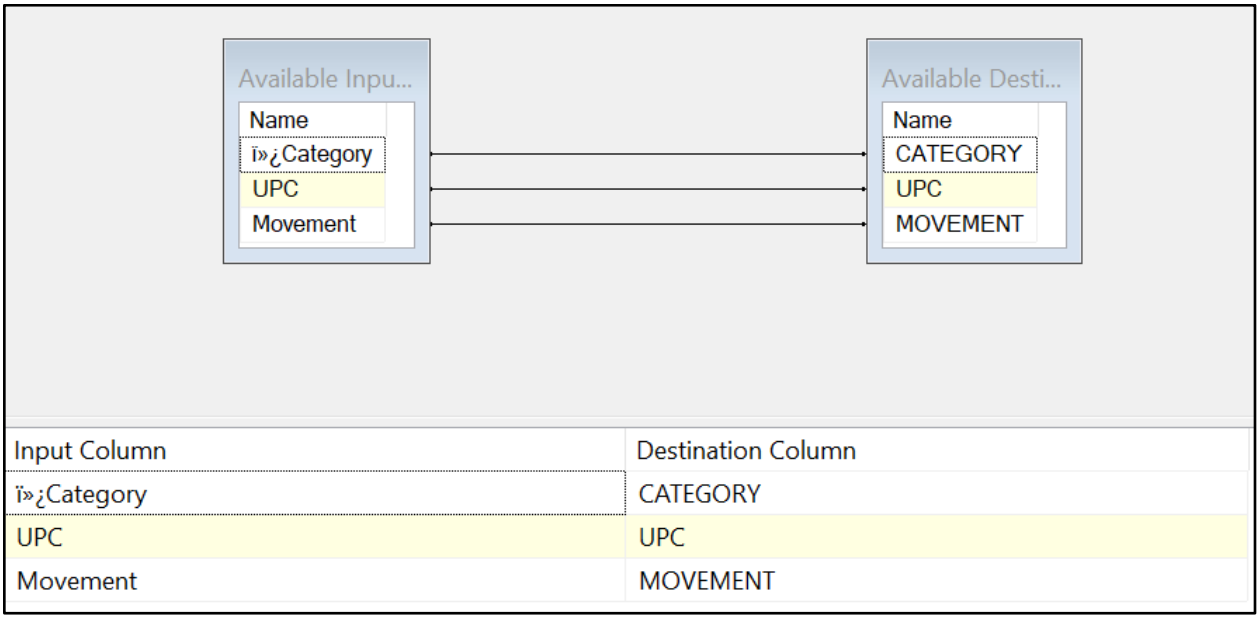


Figure 58: Data Mapping for Category

	CATEGORY	UPC	MOVEMENT
1	Analgesics	upcana	wana
2	Bath Soap	upcbat	wbat
3	Beer	upcber	wber
4	Bottled Juice	upcbjc	wbjc
5	Cereals	upccer	wcer
6	Cheese	upcche	wche
7	Cigarettes	upccig	wcig
8	Cookies	upccoo	wcoo
9	Crackers	upccra	wcra
10	Canned Soup	upccso	wcso
11	Dish Detergent	upcdid	wdid

Figure 59: Table for stg_category

6. Stg_weekdecode: Weekdecode data from the weekdecode.csv source file is extracted to the stg_weekdecode table. This will be utilized to map the week number to the date dimension. The start and end dates are converted to Unicode String [DT_WSTR].

SQL Query:

```
CREATE TABLE [dbo].[stg_weekdecode](
	[Week#] [int] NOT NULL,
	[Start] [date] NOT NULL,
	[End] [date] NOT NULL,
	[Special Events] [varchar](50) NULL,
PRIMARY KEY CLUSTERED
(
	[Week#] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

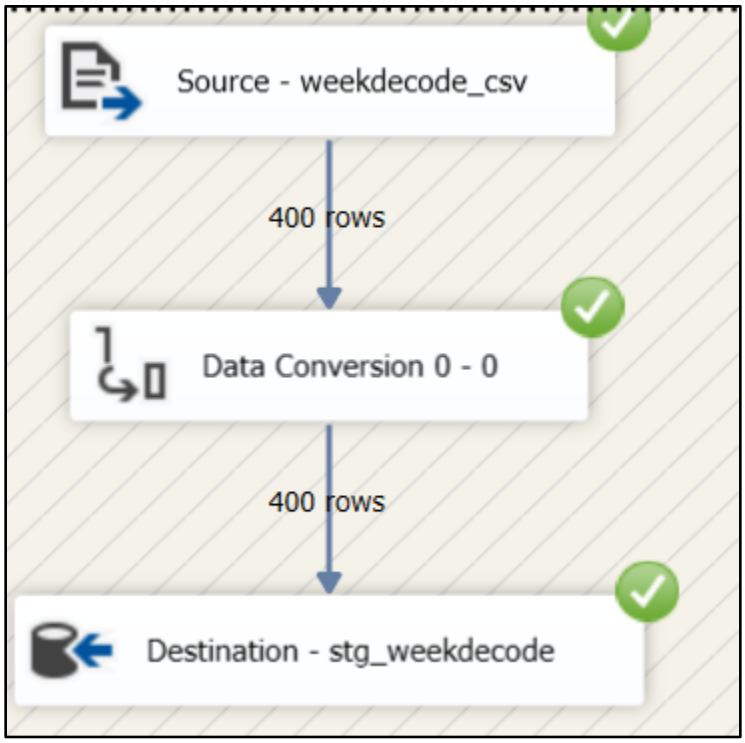
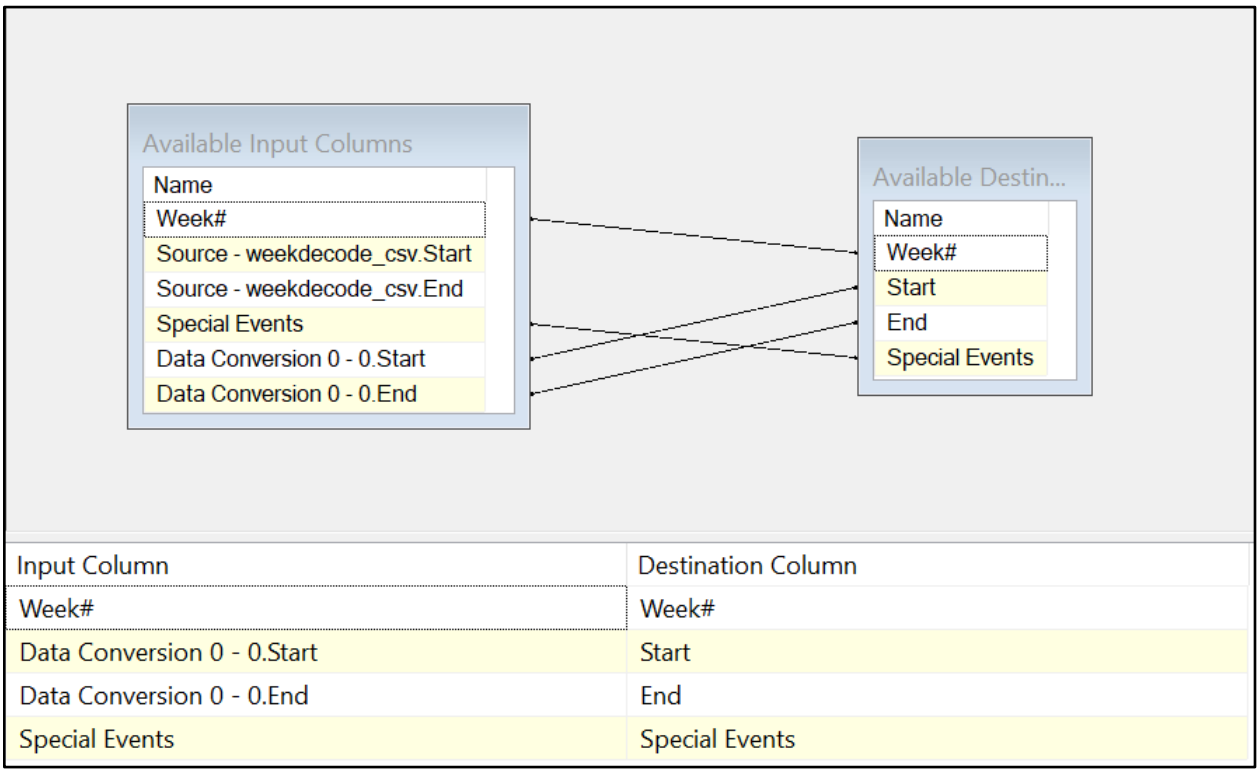


Figure 60: Weekdecode data flow



Input Column	Destination Column
Week#	Week#
Data Conversion 0 - 0.Start	Start
Data Conversion 0 - 0.End	End
Special Events	Special Events

Figure 61: Data mapping for weekdecode

	Week#	Start	End	Special Events
1	1	1989-09-14	1989-09-20	
2	2	1989-09-21	1989-09-27	
3	3	1989-09-28	1989-10-04	
4	4	1989-10-05	1989-10-11	
5	5	1989-10-12	1989-10-18	
6	6	1989-10-19	1989-10-25	
7	7	1989-10-26	1989-11-01	Halloween
8	8	1989-11-02	1989-11-08	
9	9	1989-11-09	1989-11-15	
10	10	1989-11-16	1989-11-22	
11	11	1989-11-23	1989-11-29	Thanksgiving

Figure 62: Table for stg_weekdecode

7. Stg_movementdata: Movement data for all the product categories is extracted from different files using "Multiple Flat Files Source" to table stg_movementdata.

SQL Query:

```
CREATE TABLE [dbo].[stg_movementdata](
	[STORE] [varchar](50) NULL,
	[UPC] [varchar](50) NULL,
	[WEEK] [varchar](50) NULL,
	[MOVE] [varchar](50) NULL,
	[QTY] [varchar](50) NULL,
	[PRICE] [varchar](50) NULL,
	[PROFIT] [varchar](50) NULL
) ON [PRIMARY]
GO
```

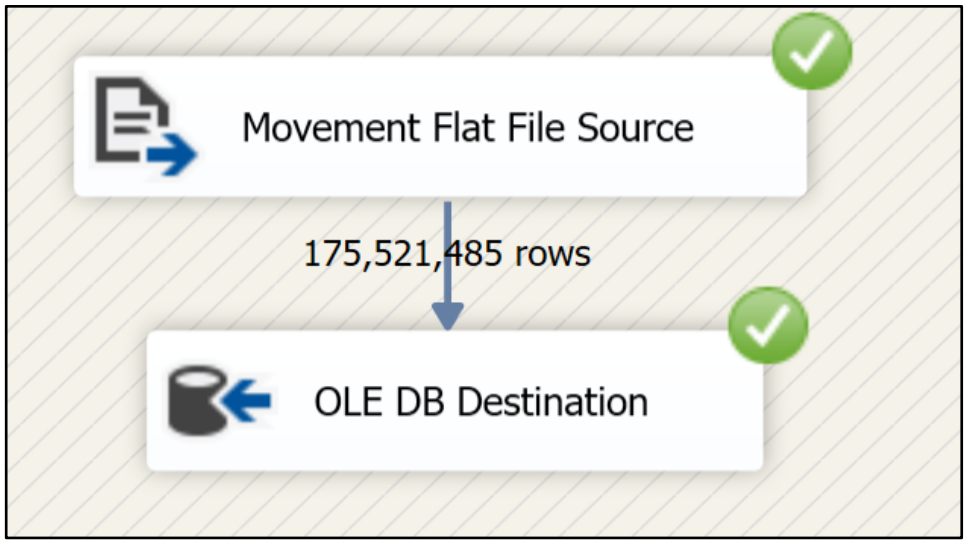
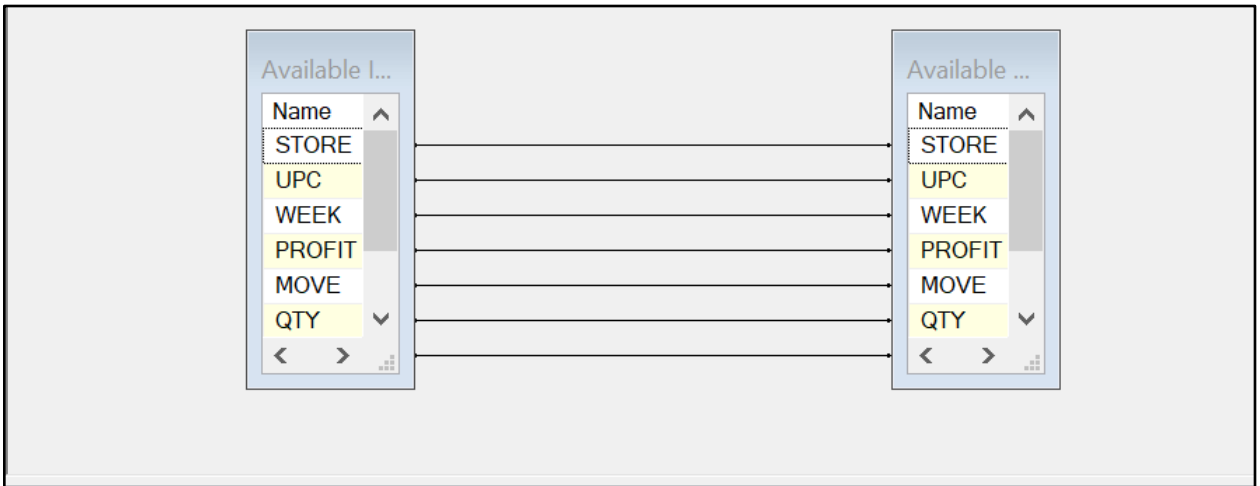


Figure 63: Movement Data Flow



Input Column	Destination Column
STORE	STORE
UPC	UPC
WEEK	WEEK
PROFIT	PROFIT
MOVE	MOVE
QTY	QTY
PRICE	PRICE

Figure 64: Data mapping for movement

	STORE	UPC	WEEK	MOVE	QTY	PRICE	PROFIT
1	44	3828161125	40	0	1	0	0
2	44	3828161125	41	0	1	0	0
3	44	3828161125	42	0	1	0	0
4	44	3828161125	43	0	1	0	0
5	44	3828161125	44	0	1	0	0
6	44	3828161125	45	0	1	0	0
7	44	3828161125	46	0	1	0	0
8	44	3828161125	47	0	1	0	0
9	44	3828161125	48	0	1	0	0
10	44	3828161125	49	0	1	0	0
11	44	3828161125	50	0	1	0	0

Figure 65: Table for stg_movementdata

8. Stg_date: Date is created using a SQL script from the SSIS. It contains dates from 1989 to 1997. Also, it creates holiday fields based on weekends and public holidays. This file would be mapped with the stg_weekdecode file to create date dimension.

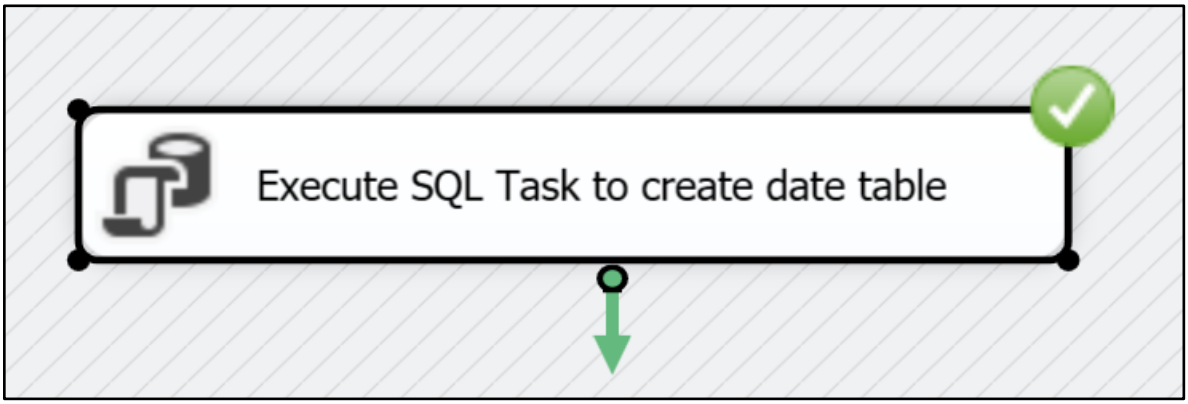


Figure 66: Date Staging control flow

	DateKey	DateFull	Year	Month	MonthKey	MonthName	DayOfMonth	NumberOf...	DayOfYear	IsWorkDayKey	IsWorkDayDescription
1	19890101	1989-01-01	1989	1	198901	January	1	31	1	1	Weekend
2	19890102	1989-01-02	1989	1	198901	January	2	31	2	0	Workday
3	19890103	1989-01-03	1989	1	198901	January	3	31	3	0	Workday
4	19890104	1989-01-04	1989	1	198901	January	4	31	4	0	Workday
5	19890105	1989-01-05	1989	1	198901	January	5	31	5	0	Workday
6	19890106	1989-01-06	1989	1	198901	January	6	31	6	0	Workday
7	19890107	1989-01-07	1989	1	198901	January	7	31	7	1	Weekend
8	19890108	1989-01-08	1989	1	198901	January	8	31	8	1	Weekend
9	19890109	1989-01-09	1989	1	198901	January	9	31	9	0	Workday
10	19890110	1989-01-10	1989	1	198901	January	10	31	10	0	Workday

Figure 67: Table for stg_date

9. Stg_transformed_custcount: Data in the Customer count file has been unpivoted. This is done to convert columns in the aforementioned source file to rows. Further, data conversion is applied to

have the correct data type in the warehouse. The derived function operator is then applied and the incoming data is then finally stored in the destination staging area after transformation.

SQL Query:

```
CREATE TABLE [dbo].[stg_transformed_custcount](
	[store] [varchar](10) NOT NULL,
	[date] [varchar](10) NOT NULL,
	[week] [varchar](10) NOT NULL,
	[Custcount] [int] NULL,
	[Coupon_Name] [varchar](10) NULL,
	[Coupon_Sales] [float] NULL
) ON [PRIMARY]
GO
```

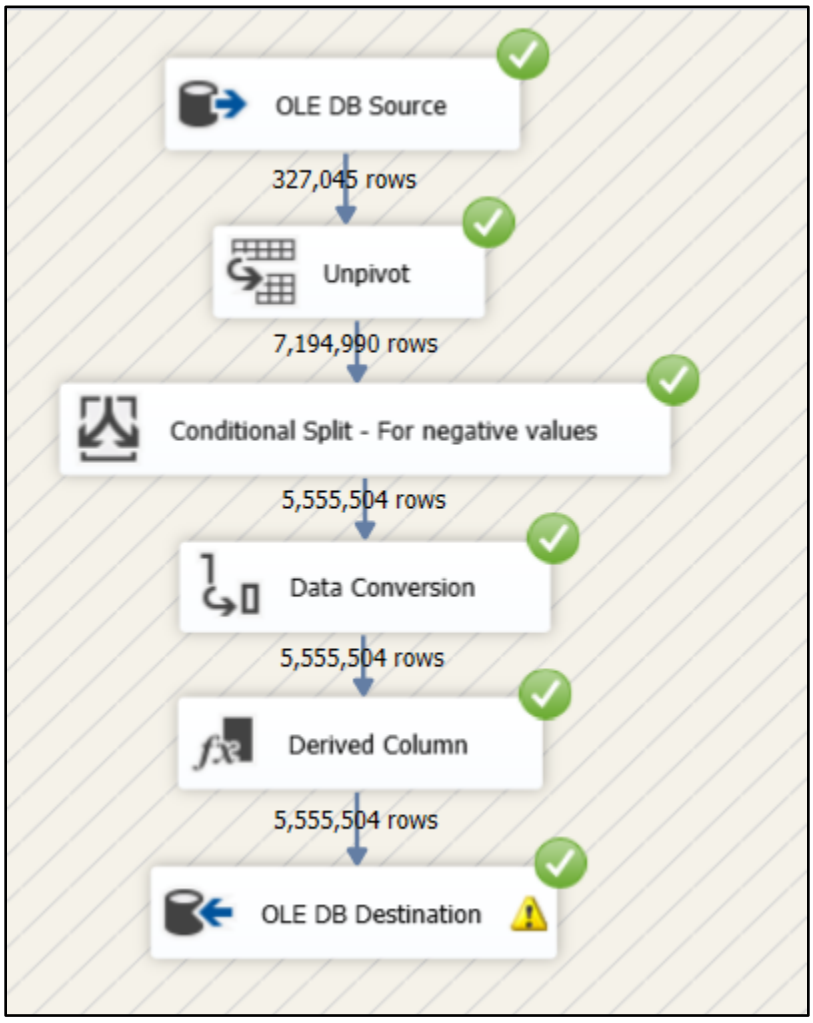
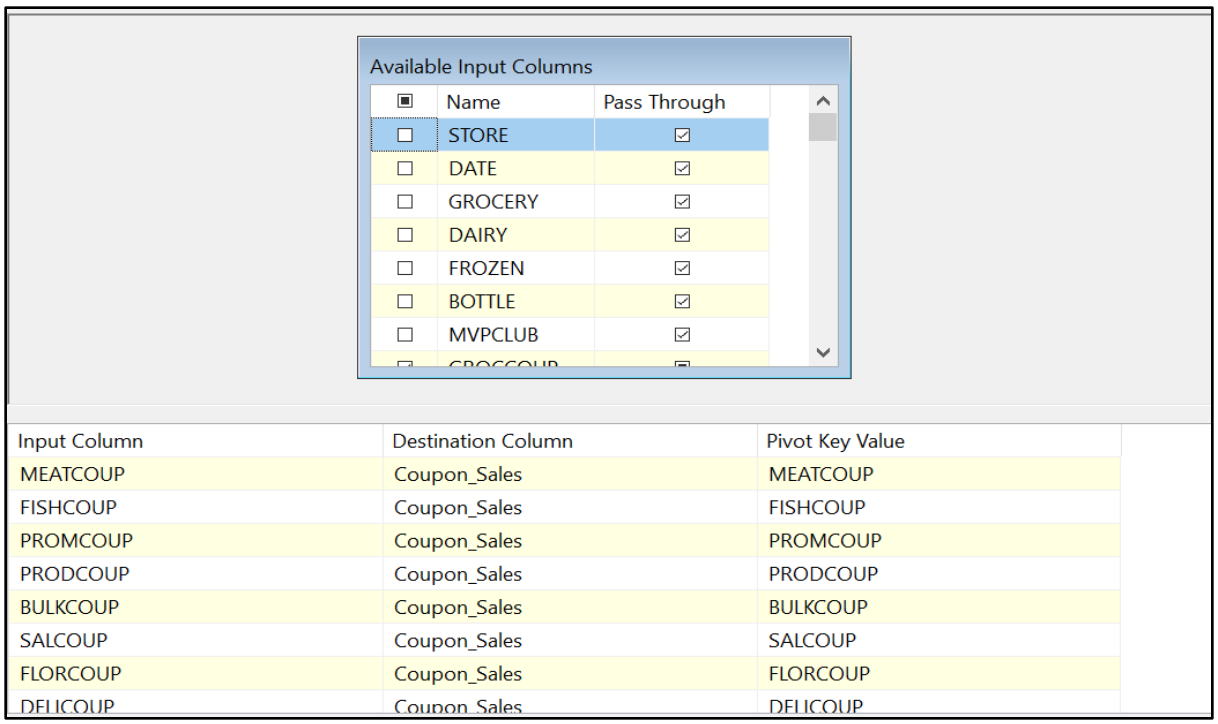
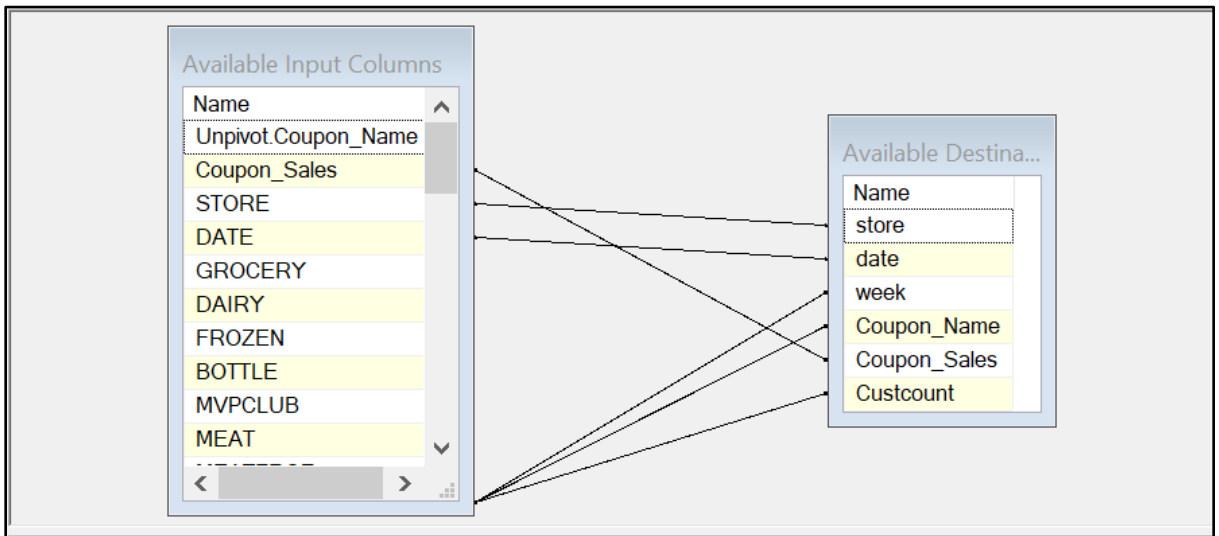


Figure 68: Transformed CCount Data Flow



Input Column	Destination Column	Pivot Key Value
MEATCOUP	Coupon_Sales	MEATCOUP
FISHCOUP	Coupon_Sales	FISHCOUP
PROMCOUP	Coupon_Sales	PROMCOUP
PRODCOUP	Coupon_Sales	PRODCOUP
BULKCOUP	Coupon_Sales	BULKCOUP
SALCOUP	Coupon_Sales	SALCOUP
FLORCOUP	Coupon_Sales	FLORCOUP
DELICOUP	Coupon_Sales	DELICOUP

Figure 69: Unpivot



Input Column	Destination Column
STORE	store
DATE	date
WEEK	week
Data Conversion.Coupon_Name	Coupon_Name
Coupon_Sales	Coupon_Sales
CUSTCOUN	Custcount

Figure 70: Data Mapping for transformed ccount

	store	date	week	Custcount	Coupon_Name	Coupon_Sales
1	121	921130	168	2795	COSMCOUP	0
2	121	921130	168	2795	DAIRCOUP	0
3	121	921130	168	2795	FISHCOUP	0
4	121	921130	168	2795	FLORCOUP	0
5	121	921130	168	2795	FROZCOUP	0
6	121	921130	168	2795	FTGCCOUP	0
7	121	921130	168	2795	FTGICOUP	0
8	121	921130	168	2795	HABACOUP	0
9	121	921130	168	2795	LIQCOUP	0
10	121	921130	168	2795	MANCOUP	451.48
11	121	921130	168	2795	MEATCOUP	0

Figure 71: Table for stg_transformed_ccount

10.8 Procedure for data extraction and loading

For extracting the data into the staging area we had to collect data from various sources such as csv files, text files, and data from the Dominick's data manual. We extracted the data from from the flat file sources, This is the very first process that comes into picture when working with staging database. This is also very crucial stage since all the further processes are dependent on the correct extraction of files.

The steps included are as follows:

- Identify the source system: Here our source system are flat files and they need to be extracted to staging tables. Source data is looked for null values or any junk value.
- Method of Extraction: We used SSIS tool to extract all the data. The data is extracted with the help of a source area that specifies the use of csv file and then mapped to the staging area table.
- Extraction Frequency: We are extracting data just once as an initial extract and it will be further extracted on weekly basis to be furthered loaded to data warehouse.
- Time Window: Whole extraction needs to be completed within a certain time frame which has been fixed in the script file and scheduling scripts. The time is usually kept at a maximum value of an hour considering the worst case scenario.
- Job sequencing: A sequence is specified for determining the execution order of the jobs, it specifies which job is dependent on other jobs to complete. For example, a table using another table's key as foreign key, required that table to be extracted first and then use its keys.
- Exception Handling: There are times when there is some input that needs special care while extraction. These columns are either type casted or tried to be loaded like that only.

For this project, the source table extracted are ccount, demo, UPC, movement, week decode, category and store that are provides either in csv files or in Dominick's manual. These are further used to do transformation in the staging area which may include basic or advanced operations.

For data loading:

All the transformed data needs to be loaded to the data warehouse. This is an area where the end user query is performed and needed data is extracted.

Procedure for data loading are as follows:

- The data transformed in the staging area is now ready to be loaded to the warehouse.
- The dimension tables are first created and loaded with a dimension_key in it.

- Required data are taken from staging tables and loaded in dimension table with required granularity.
- Summarization is created for various columns and loaded.
- Fact table are then created when all the dimension tables have been developed, with dimension table keys in the fact table to achieve referential integrity.
- Summarized data is also kept in the fact table.

10.9 ETL for dimension tables

1. **Dim_store** : Store dimension is created from the stg_store. Conditional split is used to filter the data and remove the rows with null values.

   SQL Query:

```
CREATE TABLE [dbo].[dim_store](
	[store_key] [int] IDENTITY(1,1) NOT NULL,
	[store_num] [varchar](10) NOT NULL,
	[zipcode] [varchar](10) NOT NULL,
	[city] [varchar](20) NOT NULL,
PRIMARY KEY CLUSTERED
(
	[store_key] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

   Rows with null zipcode and city is removed.

   ISNULL(ZIPCODE) || ISNULL(CITY) || ZIPCODE == "" || CITY == ""

   Steps to create dim_store:

   1. For creating store dimension, the source used is stg_store which is cleansed and filtered based on null values or any junk values.
   2. All the cleansed records are transferred to dim_store destination table.
   3. Dirty rows are filtered to other table.

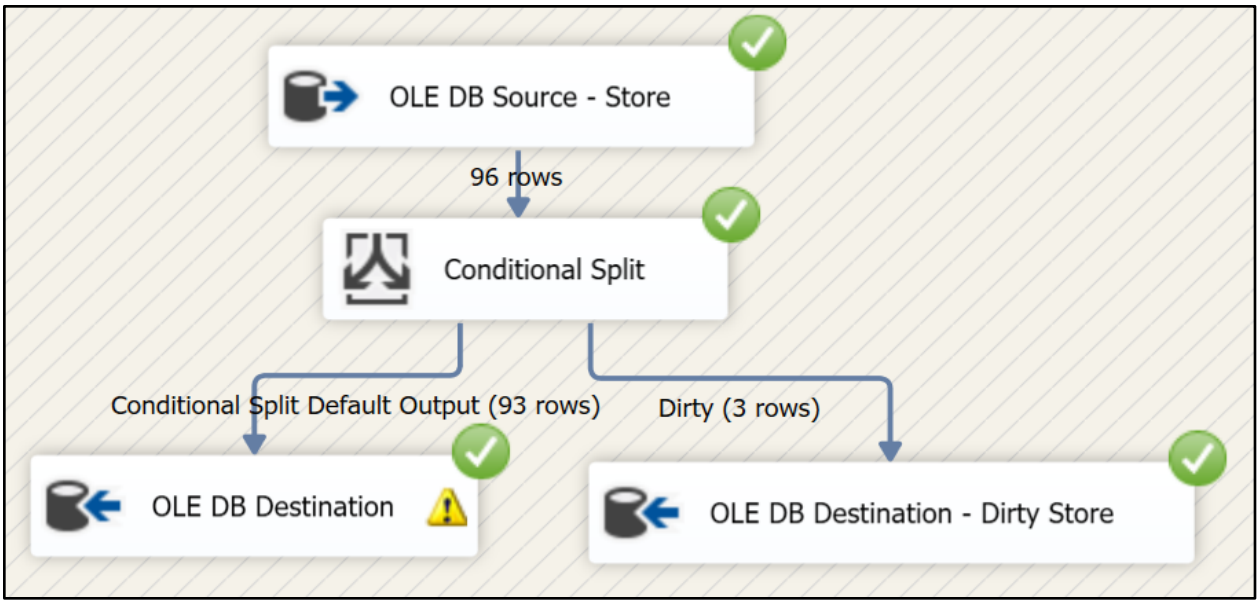
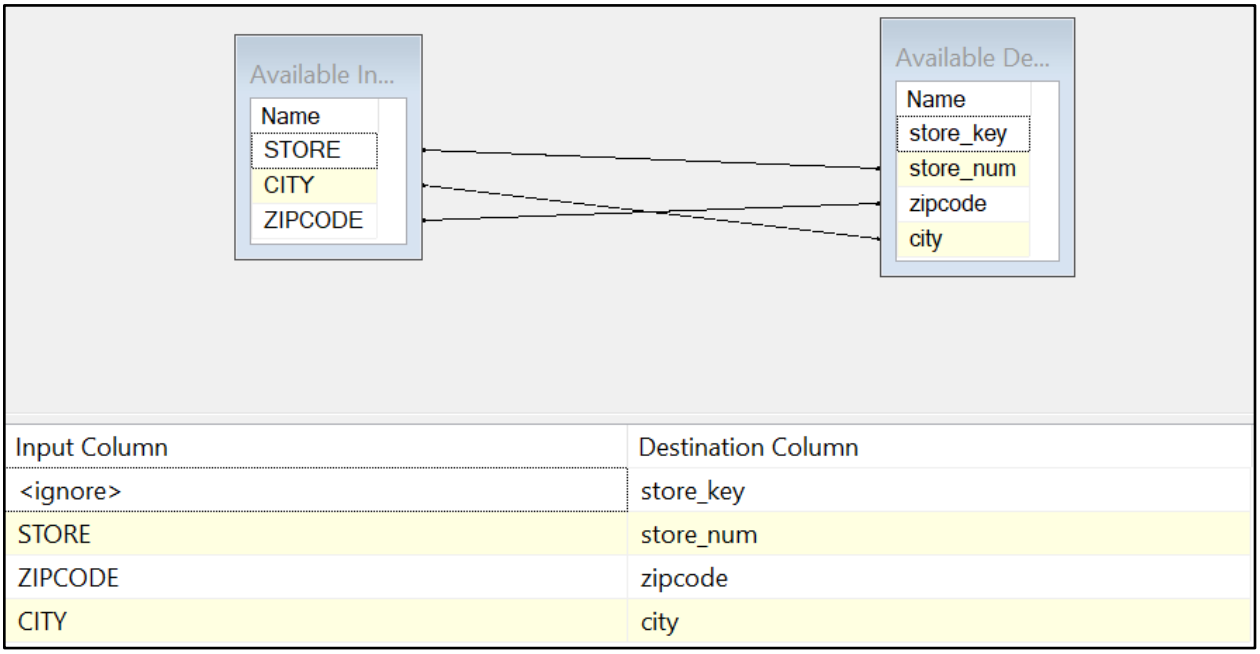


Figure 72: Store dimension data flow



Input Column	Destination Column
<ignore>	store_key
STORE	store_num
ZIPCODE	zipcode
CITY	city

Figure 73: Data Mapping for Store Dimension

	store_key	store_num	zipcode	city
1	1	2	60305	River Forest
2	2	4	60068	Park Ridge
3	3	5	60067	Palatine
4	4	8	60435	Oak Lawn
5	5	9	60053	Morton Grove
6	6	12	60660	Chicago
7	7	14	60025	Glenview
8	8	18	60171	River Grove
9	9	19	60137	Glen Ellyn
10	10	21	60103	Hanover Park
11	11	25	60639	Chicago

Figure 74: Table for dim_store

2. Dim_product : Product dimension is loaded from the stg_product table which has data from the stg_product table.

SQL Query:

```
CREATE TABLE [dbo].[dim_product](
	[product_key] [int] IDENTITY(1,1) NOT NULL,
	[upc_code] [varchar](15) NOT NULL,
	[product_desc] [varchar](50) NULL,
	[category_code] [varchar](10) NOT NULL,
	[category_name] [varchar](25) NOT NULL,
PRIMARY KEY CLUSTERED
(
	[product_key] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Conditional split is used to remove null/blank/dirty values using below expression:

ISNULL(COM_CODE) || ISNULL(UPC) || UPC == "." || UPC == "*"

Steps to create dim_product:

1. Source system to create this dimension is stg_product.
2. Next step is to pass the records to conditional split stage which will split based on null values or any junk values which is not expected.
3. The cleansed data is looked up with stg_category to have the category column as well.
4. Finally, the transformed data is loaded to dim_product.

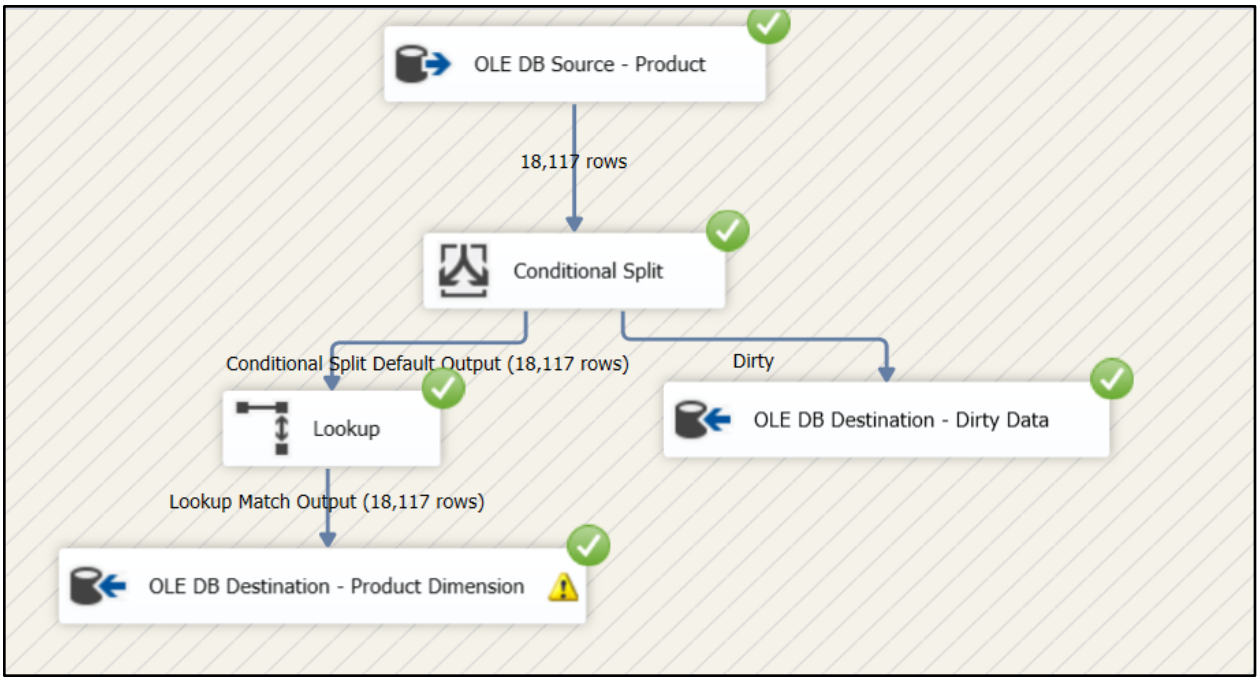


Figure 75: Product Dimension data flow

Lookup is used for combining the category names based on the filename in stg_product to upc in stg_category.

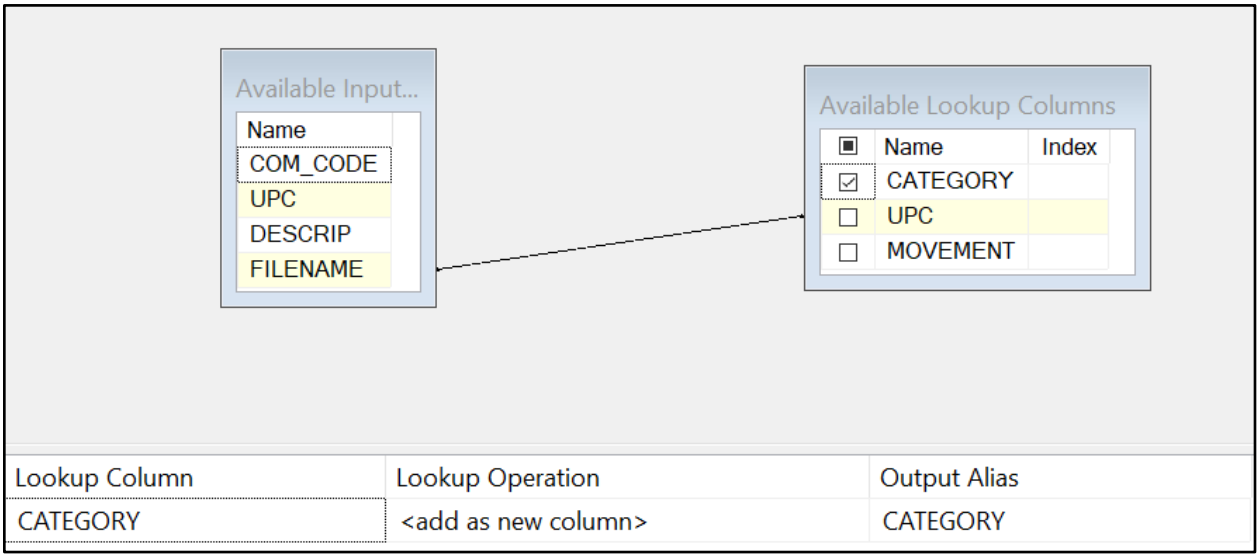
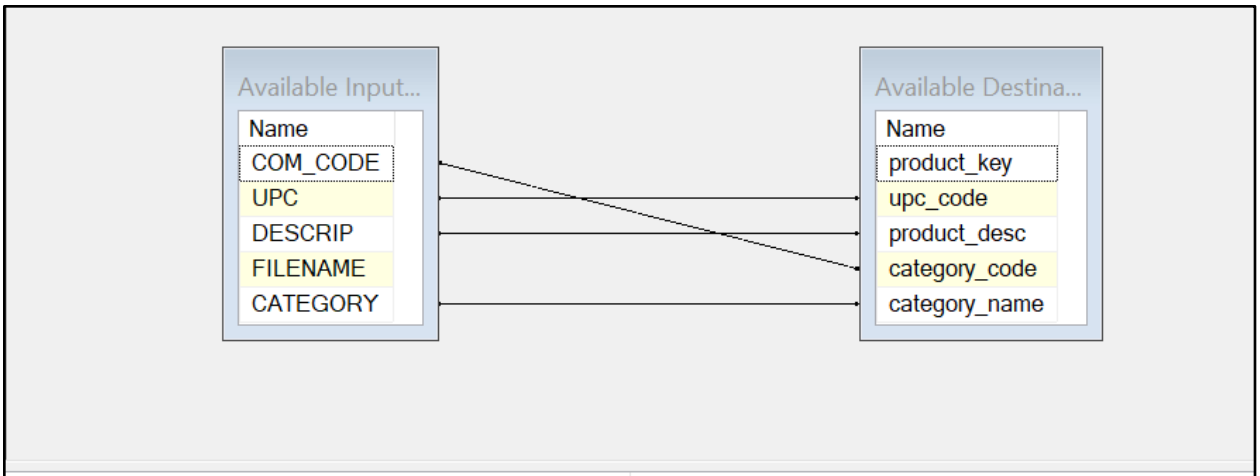


Figure 76: Lookup for category



Input Column	Destination Column
<ignore>	product_key
UPC	upc_code
DESCRIP	product_desc
COM_CODE	category_code
CATEGORY	category_name

Figure 77: Data mapping for product dimension

	product_key	upc_code	product_desc	category_code	category_name
1	1	1192603016	CAFFEDRINE CAPLETS 1	953	Analgesics
2	2	1192662108	SLEEPINAL SOFTGEL	953	Analgesics
3	3	1650001020	NERVINE TABS	953	Analgesics
4	4	1650001022	NERVINE SLEEP AID	953	Analgesics
5	5	1650004106	ALKA-SELTZER GOLD	953	Analgesics
6	6	1650004108	ALKA-SELTZER GOLD	953	Analgesics
7	7	1650004703	ALKA MINTS	953	Analgesics
8	8	2140649030	LEGATRIN PM	953	Analgesics
9	9	2586600493	PERCOGESIC A/F ANALG	953	Analgesics
10	10	2586610493	PERCOGESIC A/F ANALG	953	Analgesics
11	11	2586610501	ALEVE TABLETS	953	Analgesics

Figure 78: Table for product dimension

3. Dim_date : Date dimension is created from the stg_date looking up with stg_weekdecode to map the Dominick's week number. Date dimension has the attributes namely: date_key, date, week_num, month, year, holiday_flag, and event_desc.

SQL Query:

```
CREATE TABLE [dbo].[dim_date](
	[date_key] [int] IDENTITY(1,1) NOT NULL,
```

```
        [date_string] [varchar](6) NOT NULL,
        [date] [int] NOT NULL,
        [week_num] [varchar](3) NOT NULL,
        [month] [varchar](10) NOT NULL,
        [year] [int] NOT NULL,
        [holiday_flag] [bit] NOT NULL,
        [event_desc] [varchar](25) NULL,
PRIMARY KEY CLUSTERED
(
        [date_key] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Steps to create date dimension:

1. Source system to create this dimension is stg_date which has the complete data for the calendar dates.
2. Source data is looked up with stg_weekdecode table to have week# and event flags.
3. The matching rows is then sent to derived column stage where event flag is generated if that date has any event/holiday associated with it.
4. Non-matching rows are sent to separate table of non-matching dates.
5. Output from derived columns are loaded into dim_date.

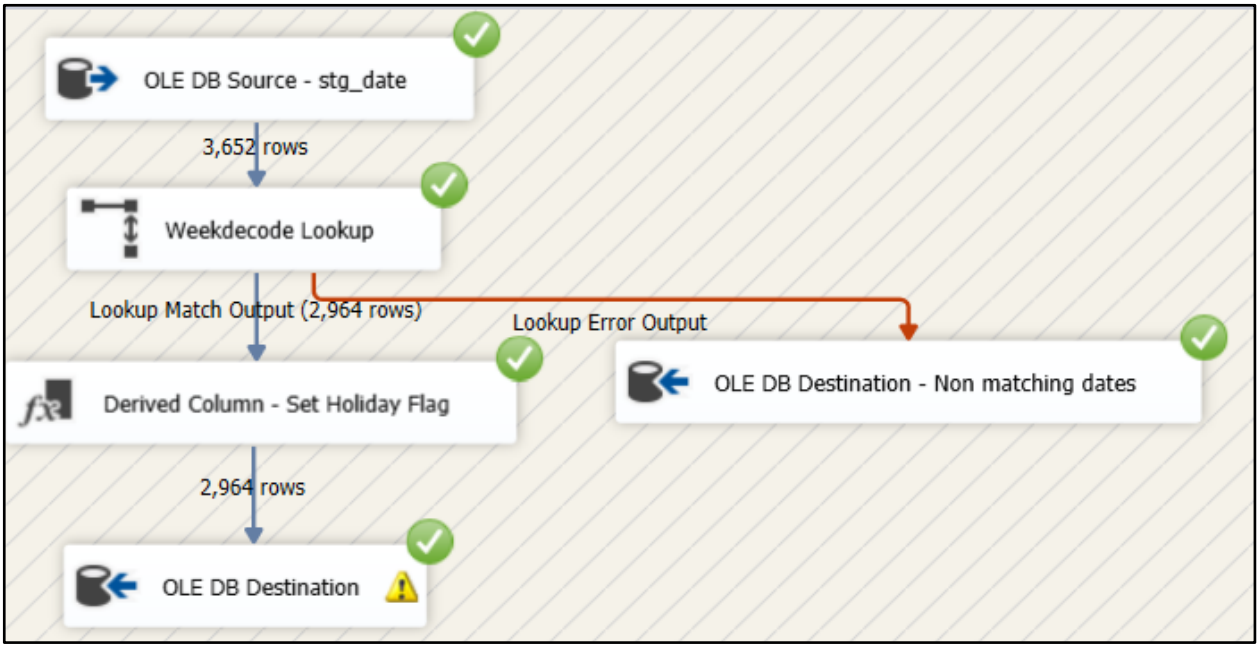


Figure 79: Date dimension data flow

Lookup is used to combine the weekdecode table based on the start and end date. The query in lookup is as below:

*select * from (select * from [dbo].[stg_weekdecode]) [refTable]*

where [refTable].[Start] <= ? and [refTable].[End] >= ?

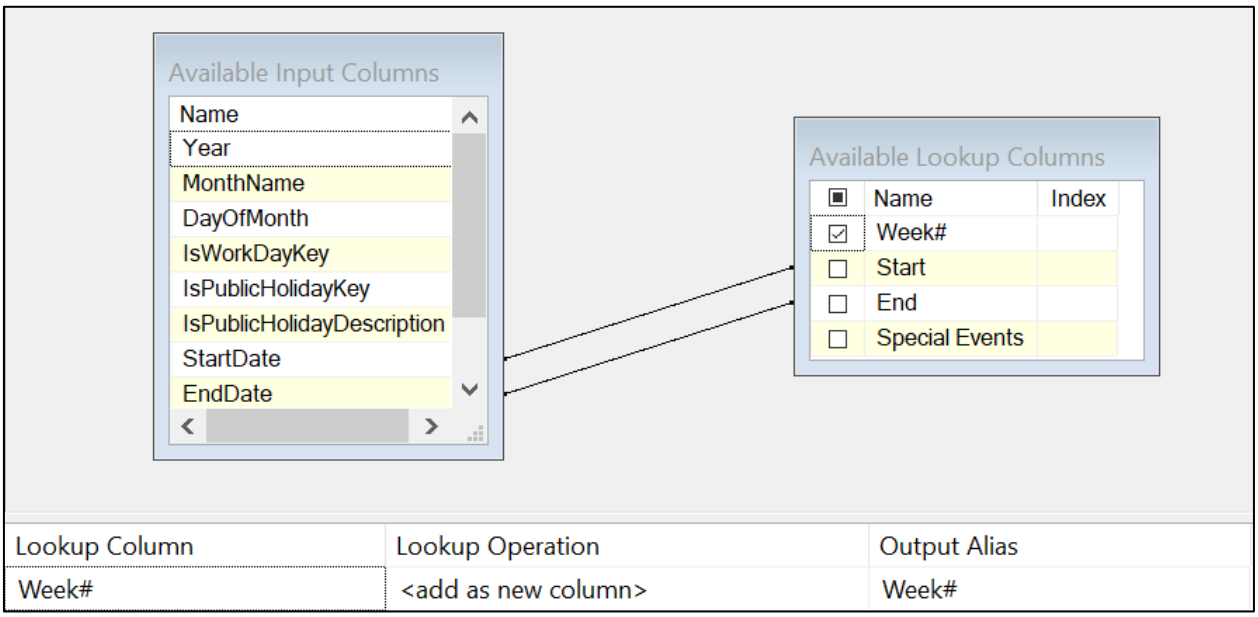


Figure 80: Lookup from weekdecode

Derived column is used to transform columns. For holiday flag below transformation is used:

((DT_WSTR,10)IsWorkDayKey == "1" || (DT_WSTR,10)IsPublicHolidayKey == "1") ? 1 : 0

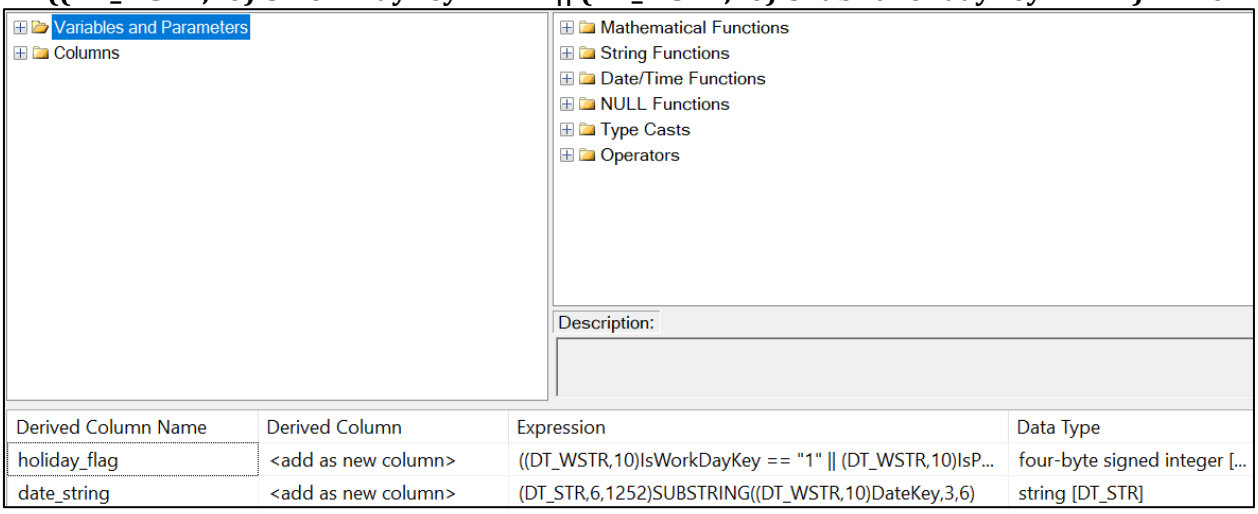
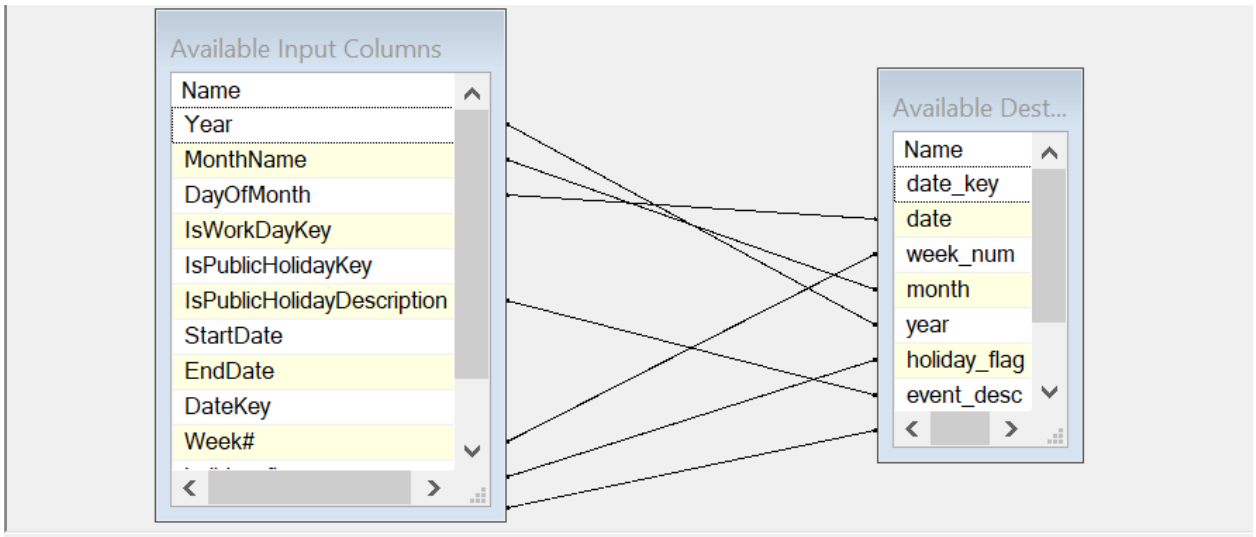


Figure 81: Derived column - Holiday flag



Input Column	Destination Column
<ignore>	date_key
DayOfMonth	date
Week#	week_num
MonthName	month
Year	year
holiday_flag	holiday_flag
IsPublicHolidayDescription	event_desc
date_string	date_string

Figure 82: Data mapping for date dimension

	date_key	date_string	date	week_num	month	year	holiday_flag	event_desc
1	1	890914	14	1	September	1989	0	N/A
2	2	890915	15	1	September	1989	0	N/A
3	3	890916	16	1	September	1989	1	N/A
4	4	890917	17	1	September	1989	1	N/A
5	5	890918	18	1	September	1989	0	N/A
6	6	890919	19	1	September	1989	0	N/A
7	7	890920	20	1	September	1989	0	N/A
8	8	890921	21	2	September	1989	0	N/A
9	9	890922	22	2	September	1989	0	N/A
10	10	890923	23	2	September	1989	1	N/A
11	11	890924	24	2	September	1989	1	N/A

Figure 83: Table for date dimension

10.10 ETL for fact tables

As discussed earlier fact tables are created when all the dimension tables have been created. Now that dim_product, dim_store and dim_date are created we will go on creating the fact table. The SQL query for creating the fact table is provided below, it has all the keys as primary keys and they are referenced to their corresponding dimension table. We will further go about discussing how the fact table is created by extracting the data from dimension tables and all other staging tables.

SQL Query:

```
CREATE TABLE [dbo].[fact_product_sales](
	[date_key] [int] NOT NULL,
	[product_key] [int] NOT NULL,
	[store_key] [int] NOT NULL,
	[product_sales] [float] NULL,
	[product_category_sales] [float] NULL,
	[product_profit] [float] NULL,
	[product_category_profit] [float] NULL,
	[coupon_sales] [float] NULL,
	[population_below_age9] [float] NULL,
	[population_above_age60] [float] NULL,
	[customer_count] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[date_key] ASC,
	[store_key] ASC,
	[product_key] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[fact_product_sales]  WITH CHECK ADD FOREIGN KEY([date_key])
REFERENCES [dbo].[dim_date] ([date_key])
GO

ALTER TABLE [dbo].[fact_product_sales]  WITH CHECK ADD FOREIGN
KEY([product_key])
REFERENCES [dbo].[dim_product] ([product_key])
GO

ALTER TABLE [dbo].[fact_product_sales]  WITH CHECK ADD FOREIGN KEY([store_key])
REFERENCES [dbo].[dim_store] ([store_key])
GO
```

Below is the diagram for creating the final fact table, the steps involved are follows:

1. Movement file from staging database are taken as source is checked for dirty data and removed with the help of conditional split stage.

2. This cleansed data is then put into derived columns and type casted to calculate total_category_sales and total_profit_sales.
3. This is further joined with category staging table using lookup, after which aggregation is performed and sales are aggregated based category_code.
4. Then demographics and customer count staging table are then looked up to have the required fact table columns i.e. custcount and population%.
5. At the end all the dimension tables are looked up to have the primary keys in fact table.

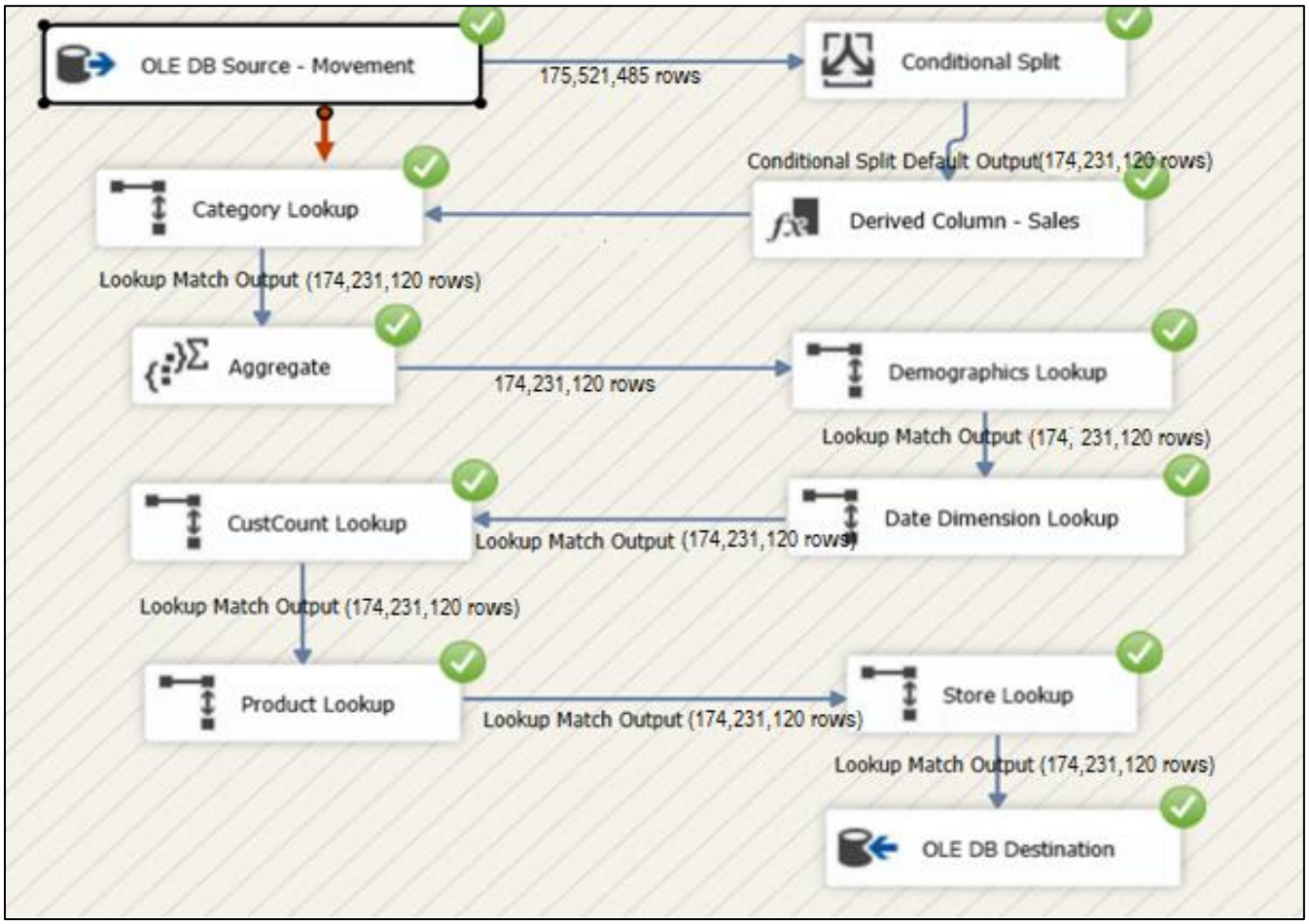
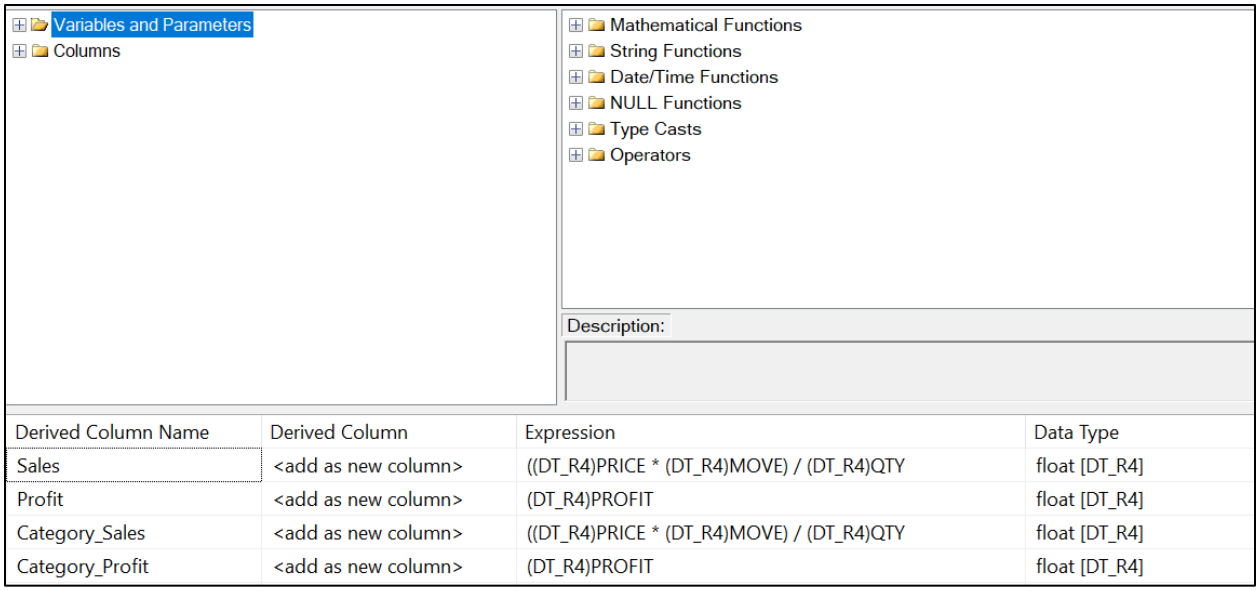


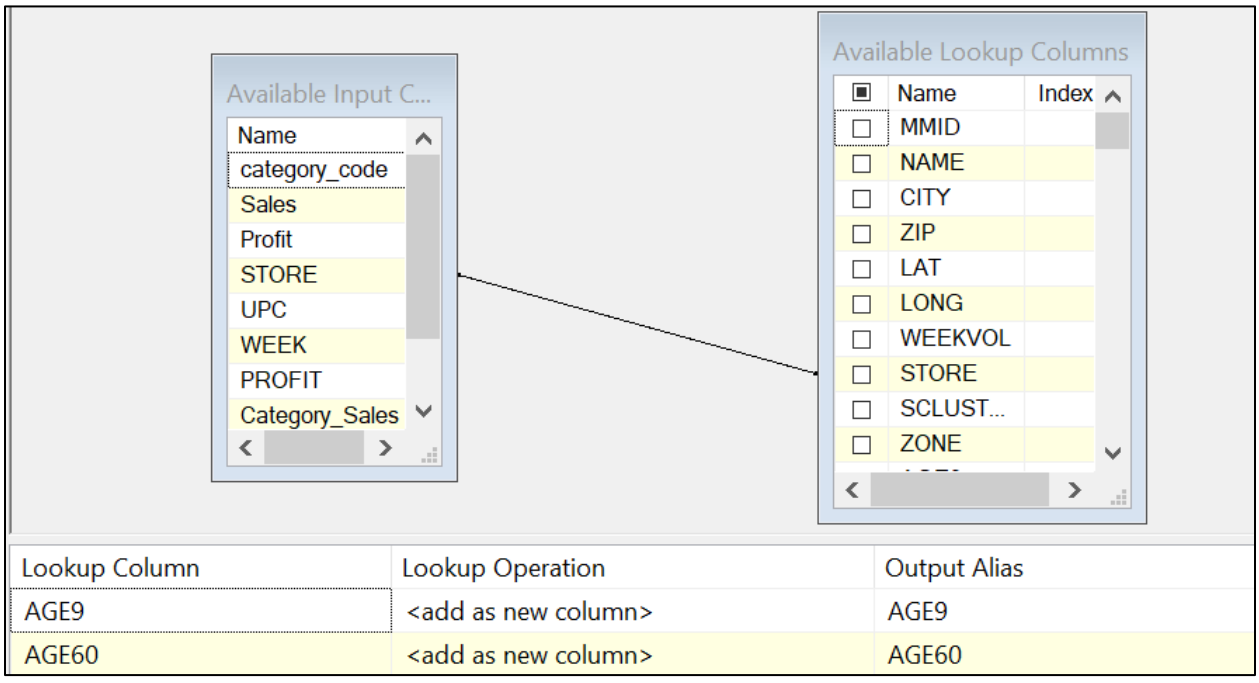
Figure 84: Fact Product Sales data flow

Below is the derived column where type cast is performed along with the appropriate calculations for calculating sales and profit.



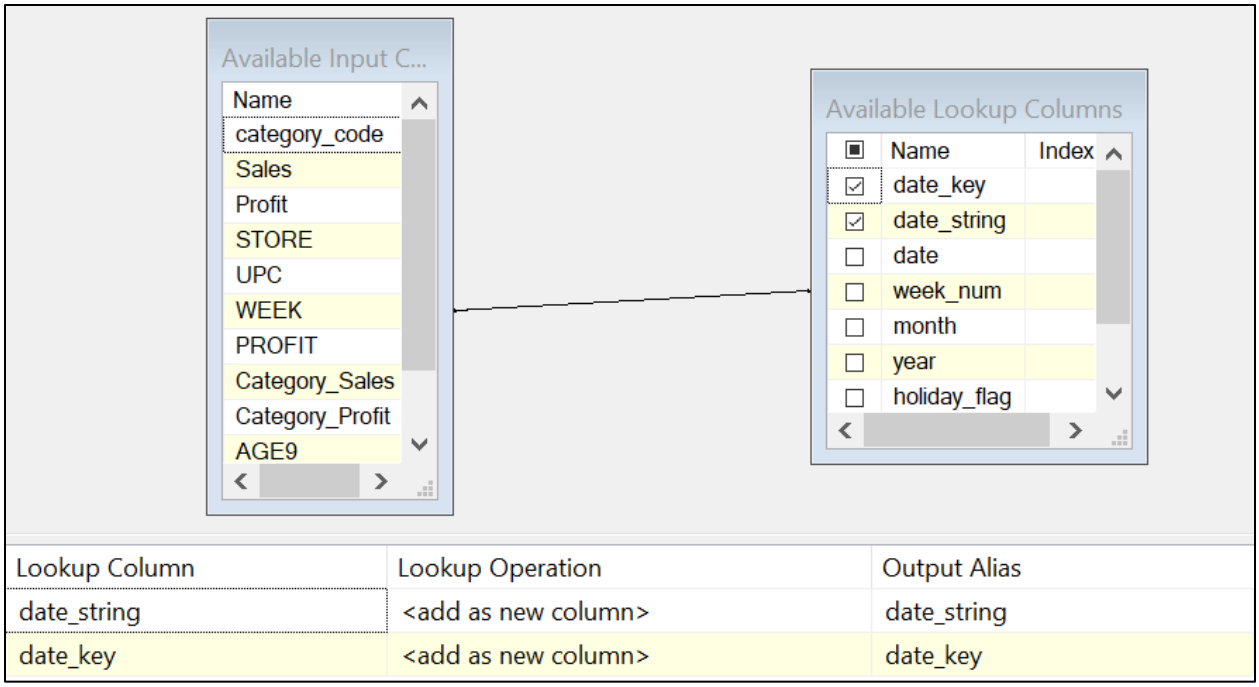
Derived Column Name	Derived Column	Expression	Data Type
Sales	<add as new column>	((DT_R4)PRICE * (DT_R4)MOVE) / (DT_R4)QTY	float [DT_R4]
Profit	<add as new column>	(DT_R4)PROFIT	float [DT_R4]
Category_Sales	<add as new column>	((DT_R4)PRICE * (DT_R4)MOVE) / (DT_R4)QTY	float [DT_R4]
Category_Profit	<add as new column>	(DT_R4)PROFIT	float [DT_R4]

Figure 85: Derived column - Sales



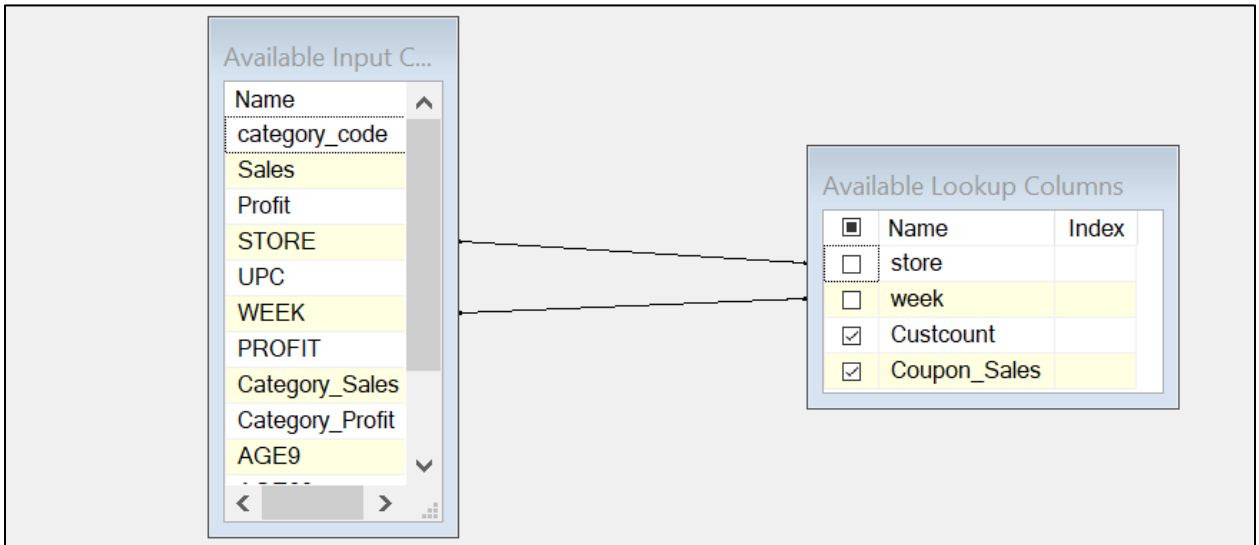
Lookup Column	Lookup Operation	Output Alias
AGE9	<add as new column>	AGE9
AGE60	<add as new column>	AGE60

Figure 86: Demographics lookup



Lookup Column	Lookup Operation	Output Alias
date_string	<add as new column>	date_string
date_key	<add as new column>	date_key

Figure 87: Date dimension lookup



Lookup Column	Lookup Operation	Output Alias
Custcount	<add as new column>	Custcount
Coupon_Sales	<add as new column>	Coupon_Sales

Figure 88: Transformed CCount lookup

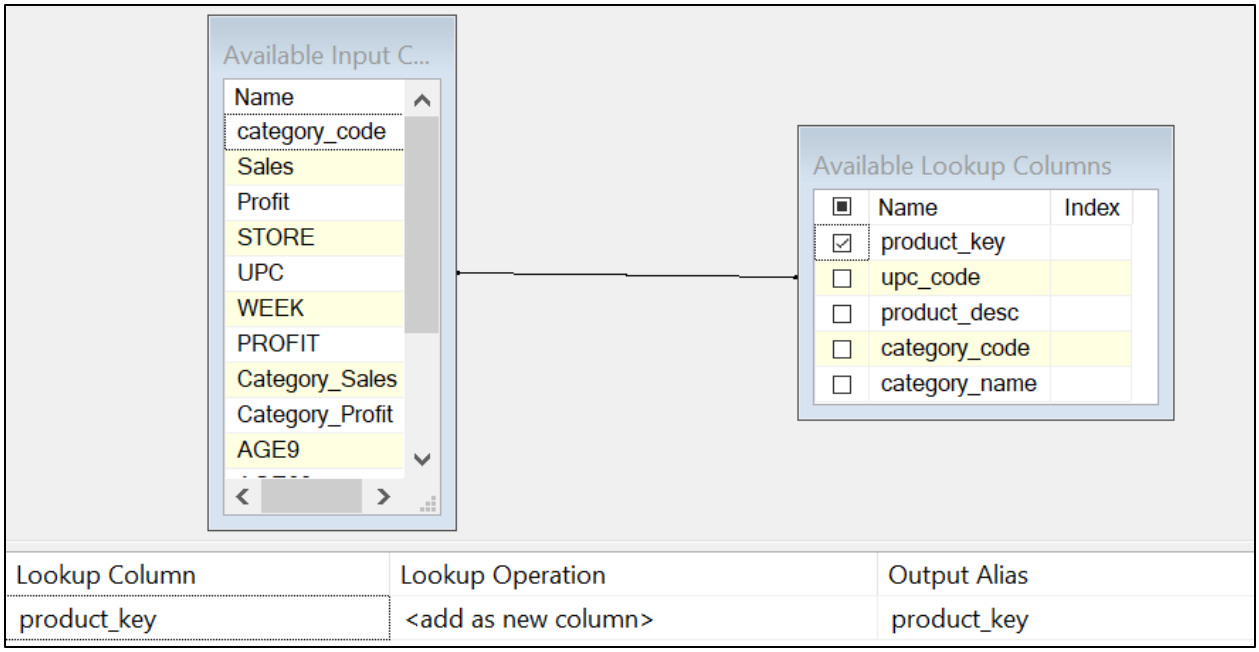


Figure 89: Product Dimension lookup

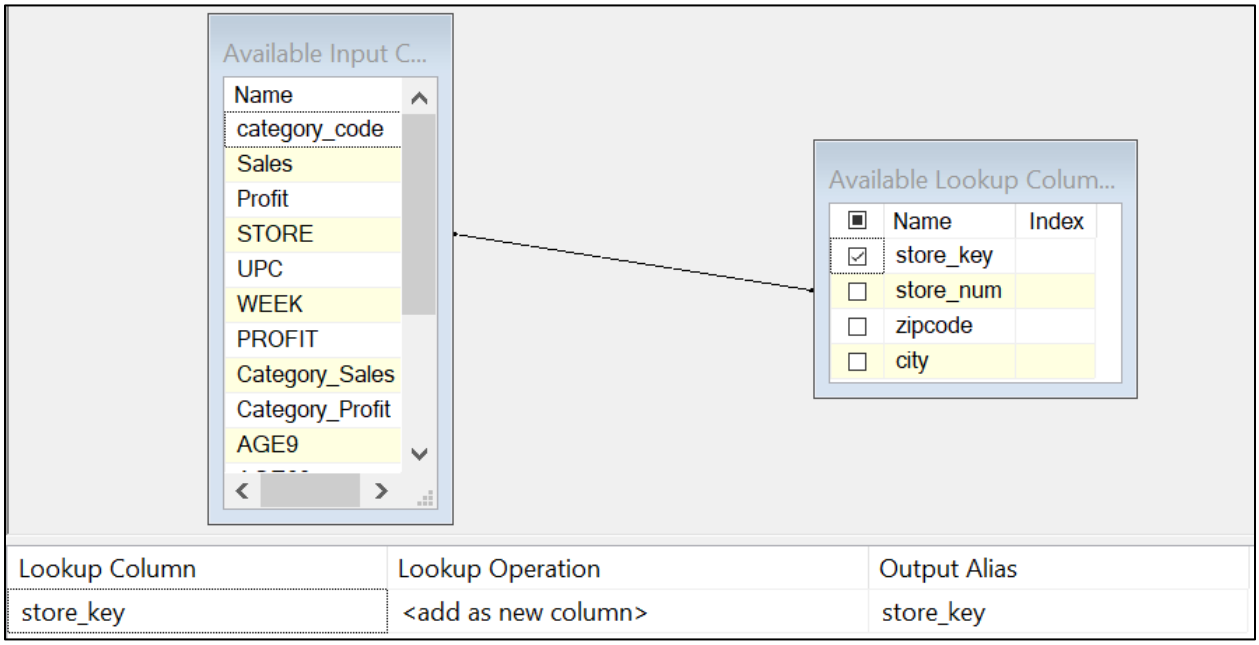
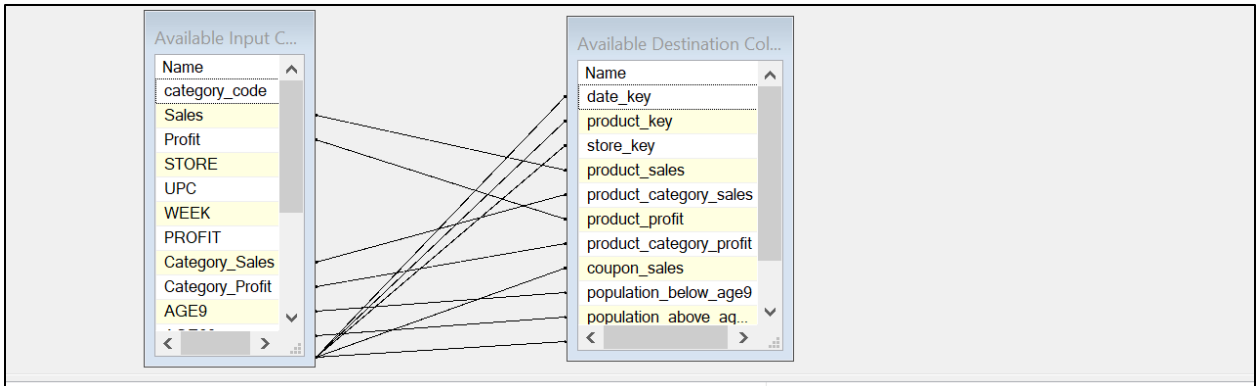


Figure 90: Store dimension lookup



Input Column	Destination Column
date_key	date_key
product_key	product_key
store_key	store_key
Sales	product_sales
Category_Sales	product_category_sales
Profit	product_profit
Category_Profit	product_category_profit
Coupon_Sales	coupon_sales
AGE9	population_below_age9

Figure 91: Data Mapping for fact table

	date_key	product_key	store_key	product_sales	product_category_sales	product_profit	product_category_profit	coupon_sales	population_below_age9	population_above_age60	customer_count
1	1	63	18	0	200.429998159409	0	2079.55003356934	0	0.1467187625	0.1288573479	218328
2	1	63	20	0	200.429998159409	0	2079.55003356934	2381.74	0.1429616817	0.125798297	215897
3	1	63	21	0	200.429998159409	0	2079.55003356934	0	0.1217670803	0.0979219614	211993
4	1	63	22	0	200.429998159409	0	2079.55003356934	0	0.1348777349	0.1874731875	186204
5	1	63	24	0	200.429998159409	0	2079.55003356934	2749	0.132472108	0.1761597181	243289
6	1	63	25	0	200.429998159409	0	2079.55003356934	0	0.13660619	0.1522411953	290655
7	1	63	26	0	200.429998159409	0	2079.55003356934	0	0.1208391392	0.3002786809	262195
8	1	63	27	0	200.429998159409	0	2079.55003356934	33517.22	0.1479145854	0.0902222777	246780
9	1	63	29	0	200.429998159409	0	2079.55003356934	0	0.1310138278	0.192888549	240582
10	1	63	30	0	200.429998159409	0	2079.55003356934	3287.03	0.1721453742	0.1108189134	219040
11	1	63	32	0	200.429998159409	0	2079.55003356934	0	0.1335093237	0.222534262	240132
12	1	63	33	0	200.429998159409	0	2079.55003356934	2233.03	0.1484041037	0.1419920205	198264
13	1	63	35	0	200.429998159409	0	2079.55003356934	2612.92	0.1188200509	0.2102729836	260798
14	1	63	37	0	200.429998159409	0	2079.55003356934	5938.46	0.1433459818	0.1902358043	332366
15	1	63	38	0	200.429998159409	0	2079.55003356934	0	0.111723683	0.2680708699	289476
16	8	63	18	0	200.429998159409	0	2079.55003356934	0	0.1467187625	0.1288573479	239585

Figure 92: Table for fact_product_sales

10.11 Data warehouse Diagram

The database group10_602_datawraehouse has 4 tables: 3 dimension tables and 1 fact table. Dimensions dim_date, dim_product and dim_store have auto-generated surrogate keys namely date_key, product_key and store_key which are referenced in fact table fact_product_sales. The combination of keys date_key, product_key and store_key are composite primary key for the fact table.

Figure 93: Database diagram for data mart - product sales

11. Business Intelligence Reporting

11.1 Target Report for Business Questions

Table 24: Reporting tools for Business Questions

Question Number	Method for Reporting
Question 1	SSRS
Question 2	SSAS
Question 3	SSRS
Question 4	Report Builder 3.0

Question 5	SSAS + SSRS

1. What is the sales trend for Thanksgiving week each year? Which products had the highest sale over the years during this time?

This business question has been completed using SSRS.

To address this question we have generated reports and charts using SSRS. We have sales data mart which has product category sales based on dimensions date, store, and product. In this report, the data is dynamically generated based on the store number entered by the user. This generates a dashboard which includes a report and 4 different types of graphs to show the Thanksgiving sales trend.

2. What is the impact of age-wise regional demography on the sales of pharmacy products?

This business question has been completed by using SSAS.

To approach this question we created cubes for data sources after creating views so that this multidimensional cube can be used for analysis on the basis of various facts and dimensions. Hierarchy has been taken into consideration for product, date and store when developing the cube. There are also derived measures that have been created using Calculate tab in SSAS. This cube is deployed on SSAS server from where filter has been placed for Analgesics category to have the required view of products on basis of demography.

3. What is the effect of coupon promotions on the sale of different products store-wise? Do coupons impact product sales?

This business question has been done using SSRS.

To answer this question we have reports and charts using SSRS. We have used the product sales data mart to show the coupon sales based on year, store number and category name. The report is made to dynamically generate based on the year entered by the user. It generates a report and a graph which clearly depicts the coupon sales for different product categories across the stores.

4. What is the customer increase or decrease over the years based on the store?

This business question has been done using Report Builder 3.0.

Table is created and query is written in it to join the tables and get data for creating reports. The categories and series are defined and a variable is included in the parameter section so that it can take inputs at the run time. Report is run to have both graphical and chart representation.

5. How are the product category profits changing in every store over the years? What are the product categories with the highest and least profits?

This business question has been done using SSRS on top of SSAS

To answer this question first a cube was created with the required hierarchies and calculated measures in Microsoft Visual Studio. We used the sales data mart which had the date, product and store dimensions, and the product sales fact table. The measure product profit in the cube was aggregated from the product profit attribute in the fact table. After deploying the cube successfully, the report was generated i.e., SSRS on top of SSAS. Finally, the report enabled drill-down abilities and the charts that we used helped us visualize our goals. We were able to see the product category profits changing for every store over the years.

11.2 Mapping – Data Mart to Report

1. What is the sales trend for Thanksgiving week each year? Which products had the highest sale over the years during this time?

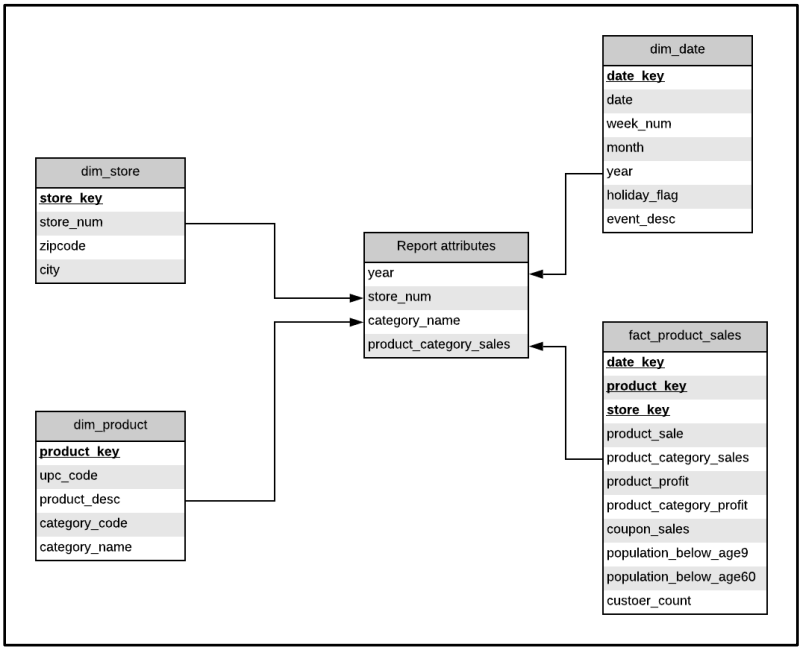


Figure 94: Report Mapping for Business Question 1

2. What is the impact of age-wise regional demography on the sales of pharmacy products?

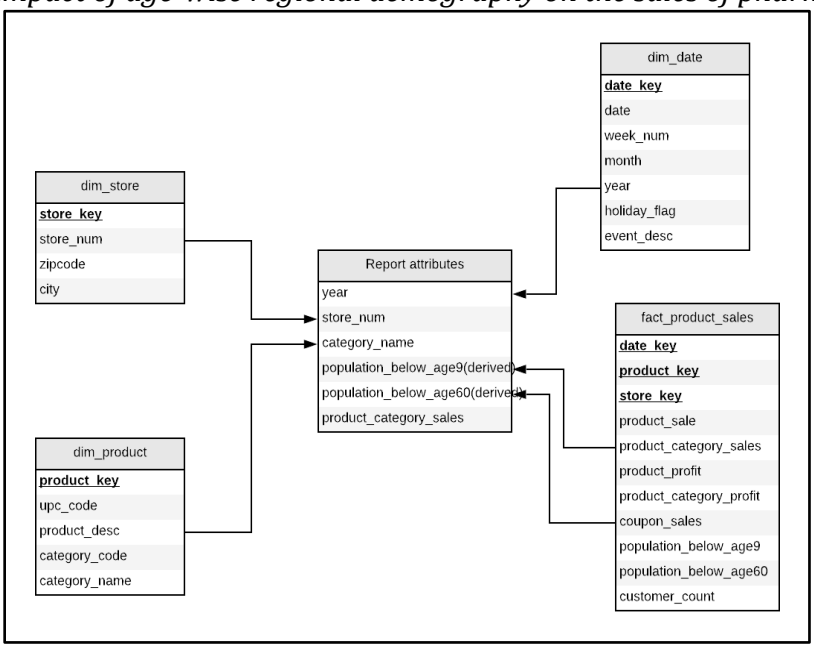


Figure 95: Report Mapping for Business Question 2

3. What is the effect of coupon promotions on the sale of different products store-wise? Do coupons impact product sales?

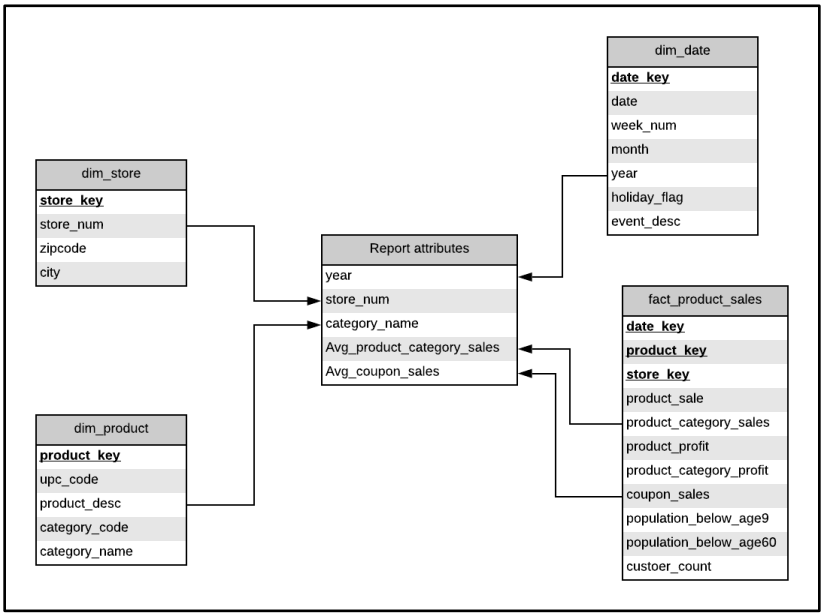


Figure 96: Report Mapping for Business Question 3

4. What is the customer increase or decrease over the years based on the store?

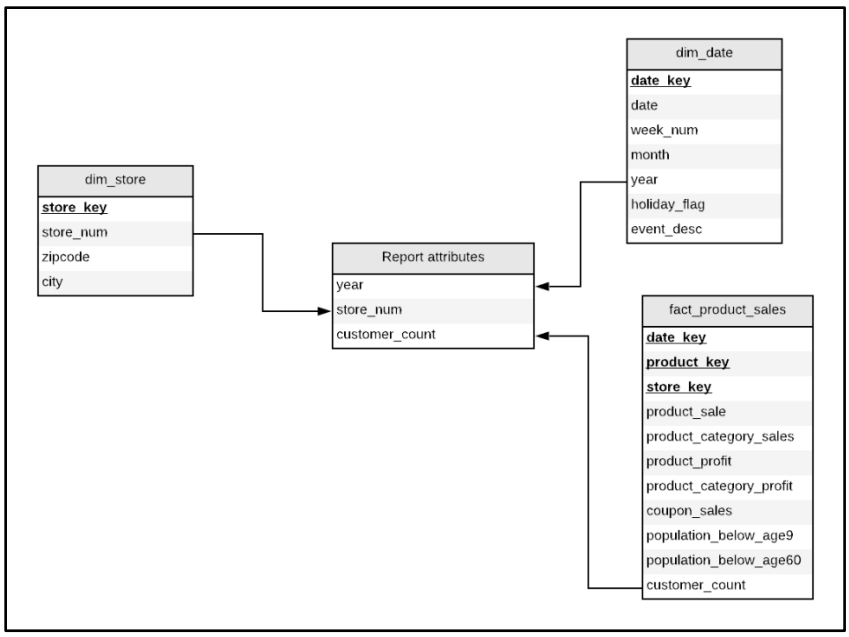


Figure 97: Report Mapping for Business Question 4

5. How are the product category profits changing in every store over the years? What are the product categories with the highest and least profits?

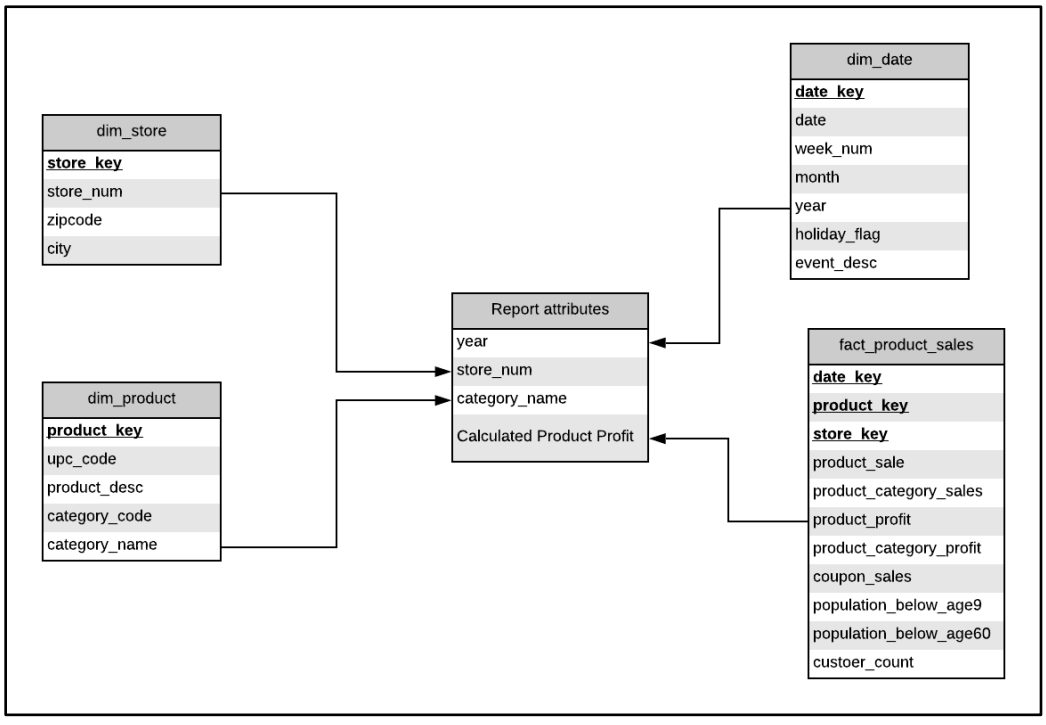


Figure 98: Report Mapping for Business Question 5

11.3 Reporting

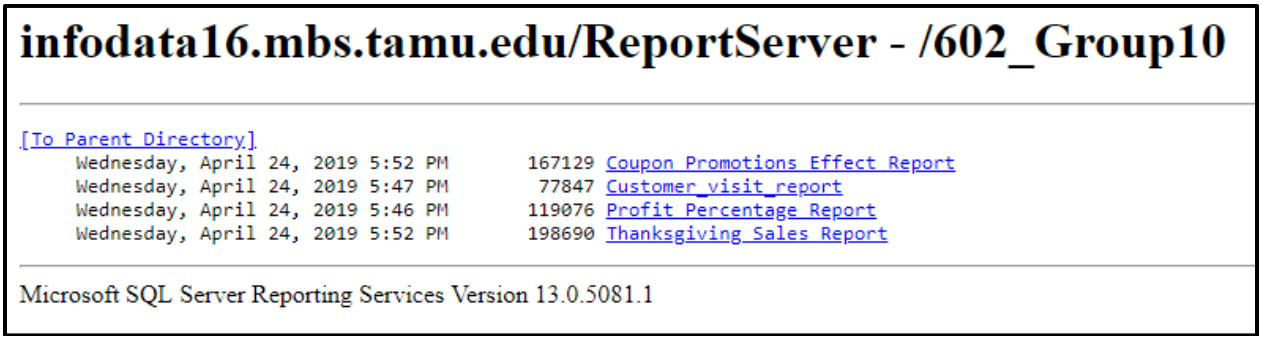


Figure 99: Report Server - Deployed Reports

1. What is the sales trend for Thanksgiving week each year? Which products had the highest sale over the years during this time?

To address the business question for Thanksgiving sales trend, we are using SSRS. The data source is selected as Microsoft SQL server and the required configuration details are entered. The data warehouse *group10_602_datawarehouse* is selected from the dropdown.

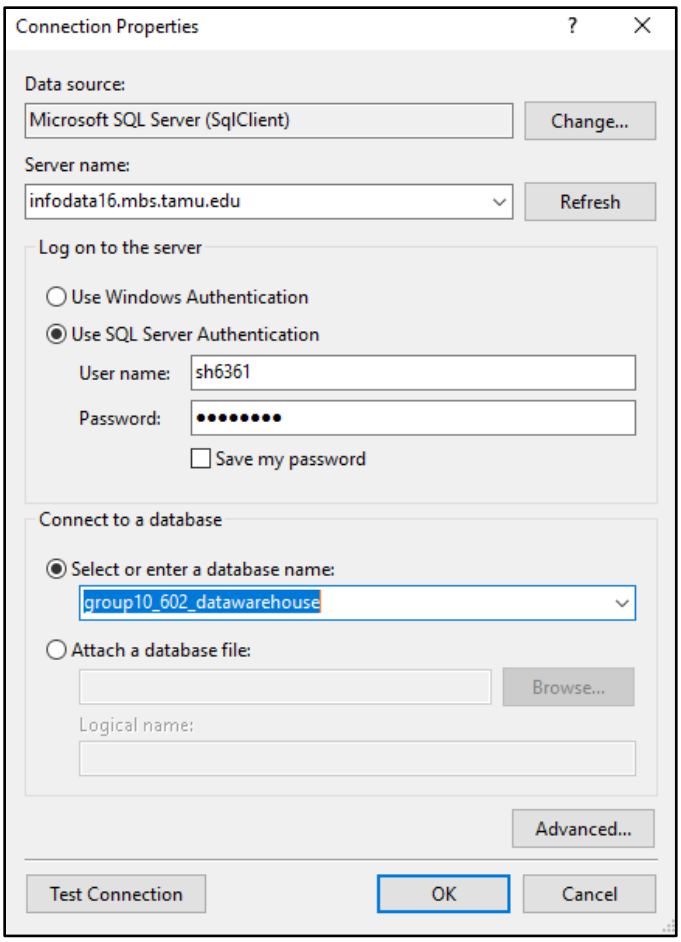


Figure 100: SSRS - Data Source

In query designer, dim_date, dim_store, dim_product and fact_product_sales is added. The attributes product_category_sales, store_num, year and category_name are selected. In the query filter for event_desc is added as "Thanksgiving".

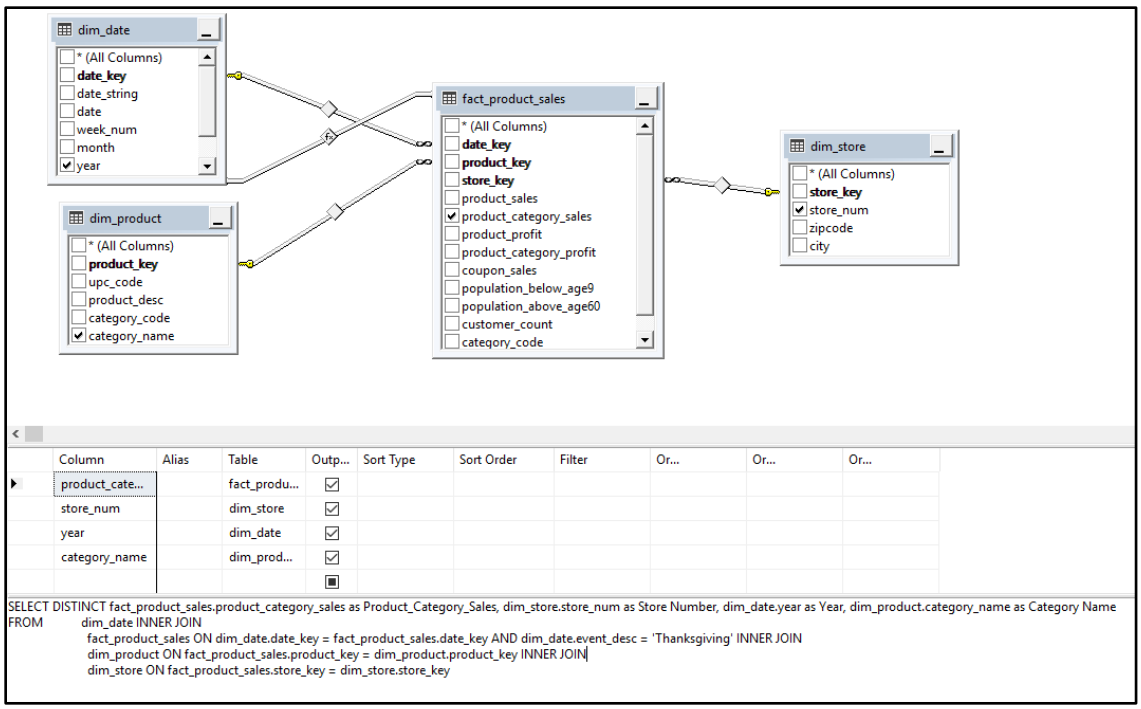


Figure 101: Query Designer

In the query, a dynamic parameter for store_num is added. This helps in generating custom reports based on the store number.

SQL Query:

```
SELECT DISTINCT fact_product_sales.product_category_sales AS Product_Category_Sales,
dim_store.store_num AS Store_Number, dim_date.year AS Year, dim_product.category_name
AS Category_Name
FROM          dim_date INNER JOIN
                fact_product_sales ON dim_date.date_key = fact_product_sales.date_key
AND dim_date.event_desc = 'Thanksgiving' INNER JOIN
                dim_product ON fact_product_sales.product_key = dim_product.product_key
INNER JOIN
                dim_store ON fact_product_sales.store_key = dim_store.store_key
WHERE (dim_store.store_num = @store_num)
```

In designing the reporting table, store_num is added as page field and grouped by year and category_name. The value added in the detailed section is product_category_sales.

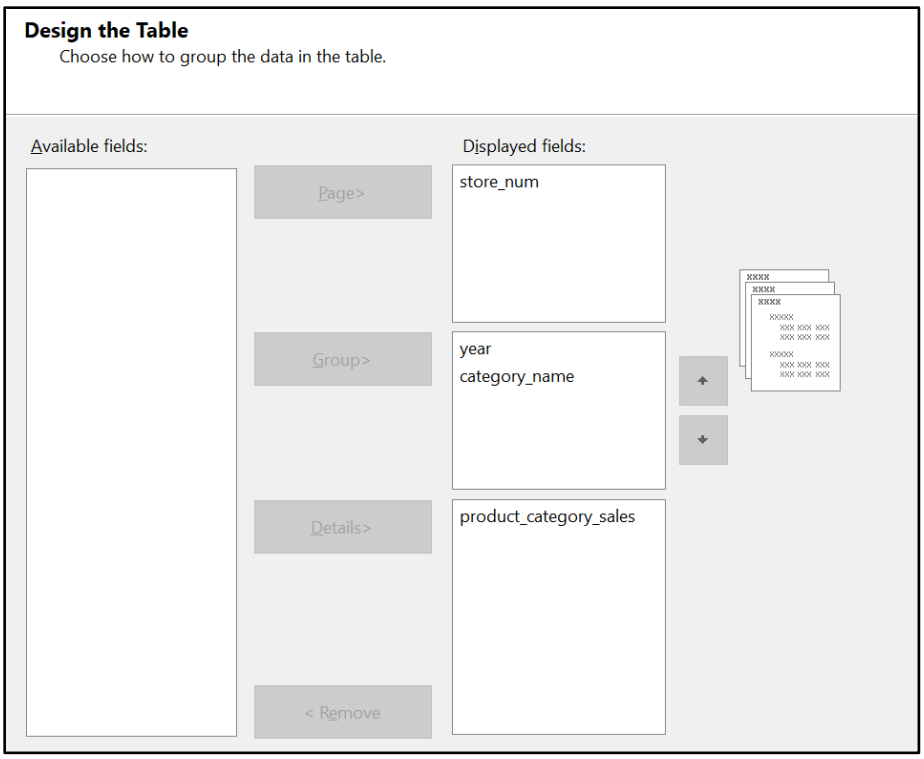


Figure 102: SSRS - Table Design for Thanksgiving Report

When the report is previewed, it asks for the store number which then generates the Thanksgiving Sales Trend report dynamically based on the store number entered.

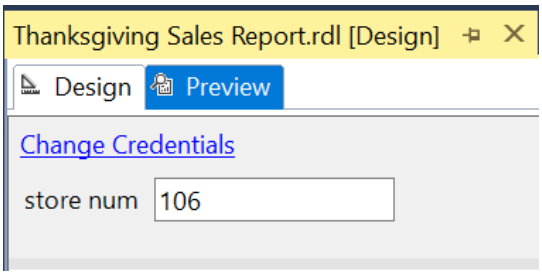


Figure 103: Thanksgiving Report - Filter by Store

The report is deployed to http://infodata16.mbs.tamu.edu/ReportServer in the folder "602_Group10" as "Thanksgiving Sales Report".

The report is displayed based on the store number entered. It presents a dashboard which includes reporting table and 4 charts corresponding to the sales trend.

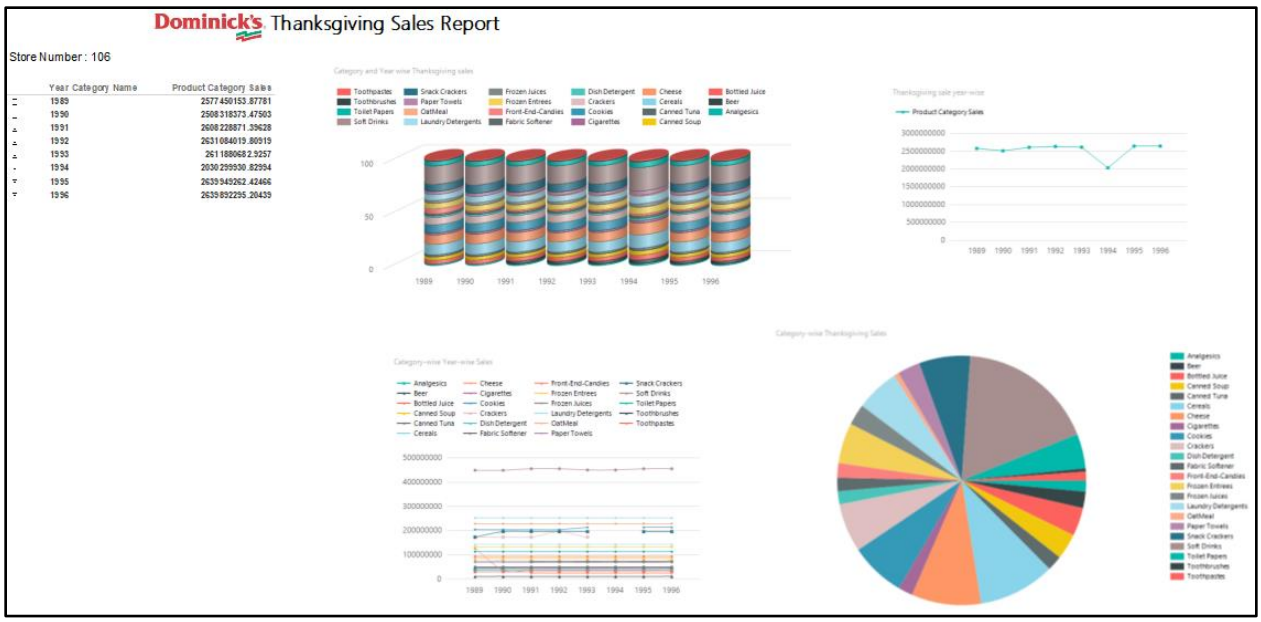


Figure 104: Thanksgiving Sales Dashboard

The report is for a single store includes Year, Category Name and Product Category Sales.

Store Number : 106

Year Category Name	Product Category Sales
1989	2577450153.87781
1990	2508318373.47503
1991	2608228871.39628
1992	2631084019.80919
1993	2611880682.9257
1994	2030299930.82994
1995	2639949262.42466
1996	2639892295.20439

Figure 105: Thanksgiving Sales Data Report

As the drill down is enabled, on expanding it shows the sales corresponding to each category.

Store Number : 106

Year	Category Name	Product Category Sales
1989		**2577450153.87781**
	Analgesics	36299228.2927702
	Bottled Juice	92866069.1501749
	Canned Soup	84919877.0880288
	Canned Tuna	49480225.5171132
	Cereals	252167235.769663
	Cheese	228568670.599841
	Cigarettes	47532560.0067981
	Cookies	204370132.716886
	Crackers	172820581.504008
	Dish Detergent	42815562.4802805
	Fabric Softener	43922240.5211457
	Front-End-Candies	127951332.707733

Figure 106: Thanksgiving Sales Report drilled down

Below bar chart shows the sales trend for multiple categories for Store 106 year-wise in Thanksgiving week.

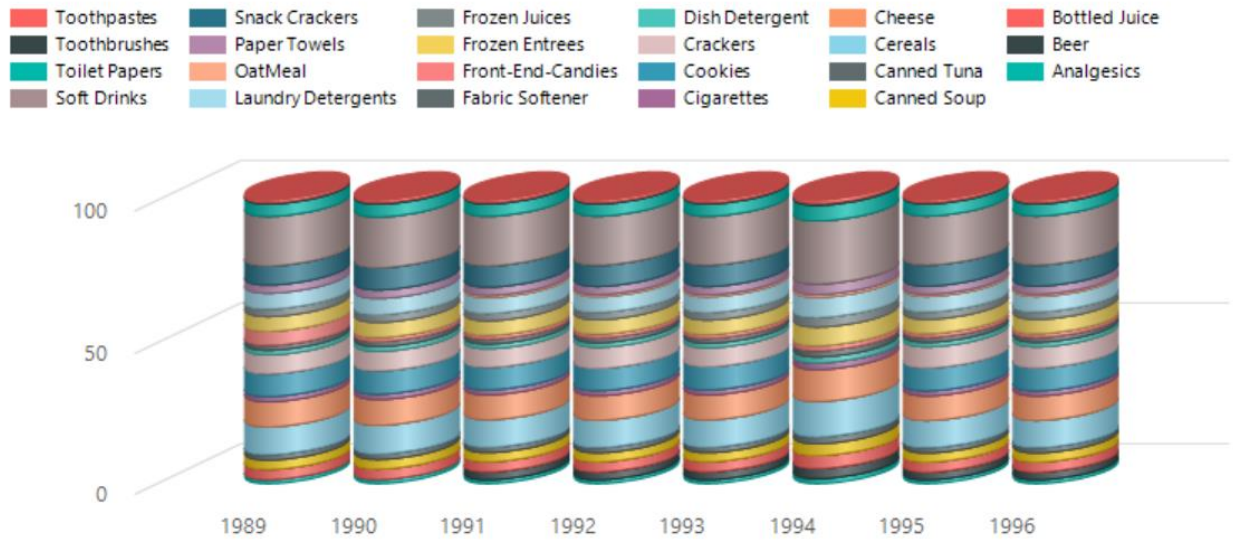


Figure 107: Thanksgiving Sales Chart - Year vs Category

Below line chart shows the sales in Thanksgiving week increase/ decrease year on year.

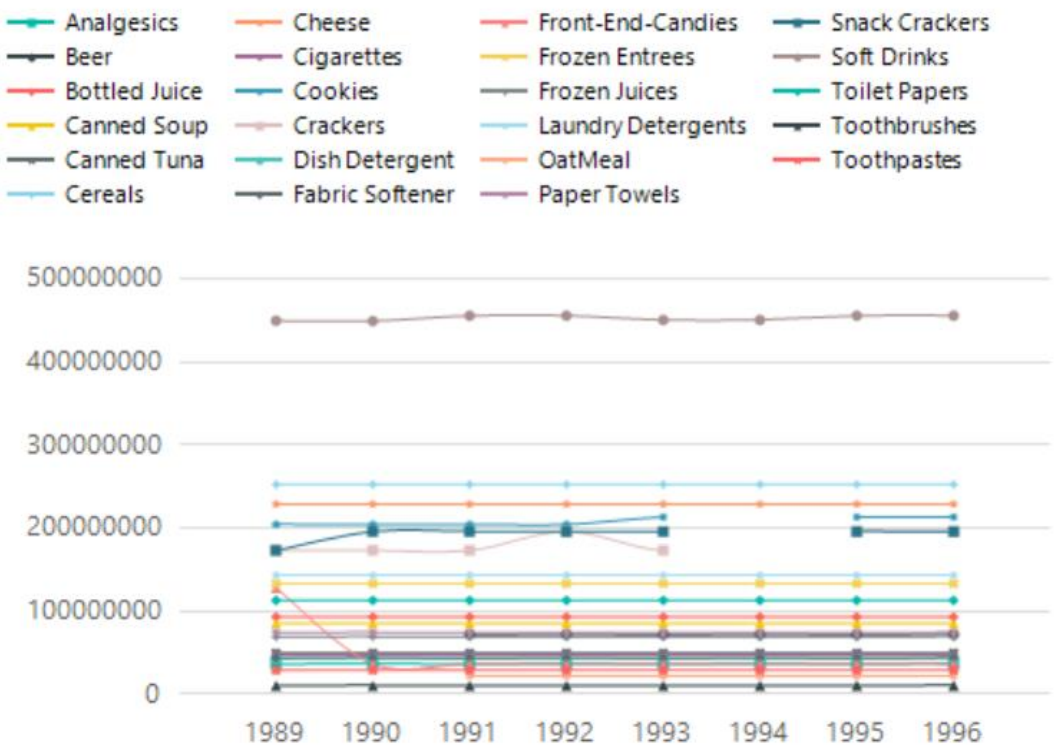


Figure 108: Thanksgiving Sales Chart - Increase / Decrease in Sales

Below pie chart gives a clear idea of which products have higher sales in the Thanksgiving week from the year 1989 - 1996.

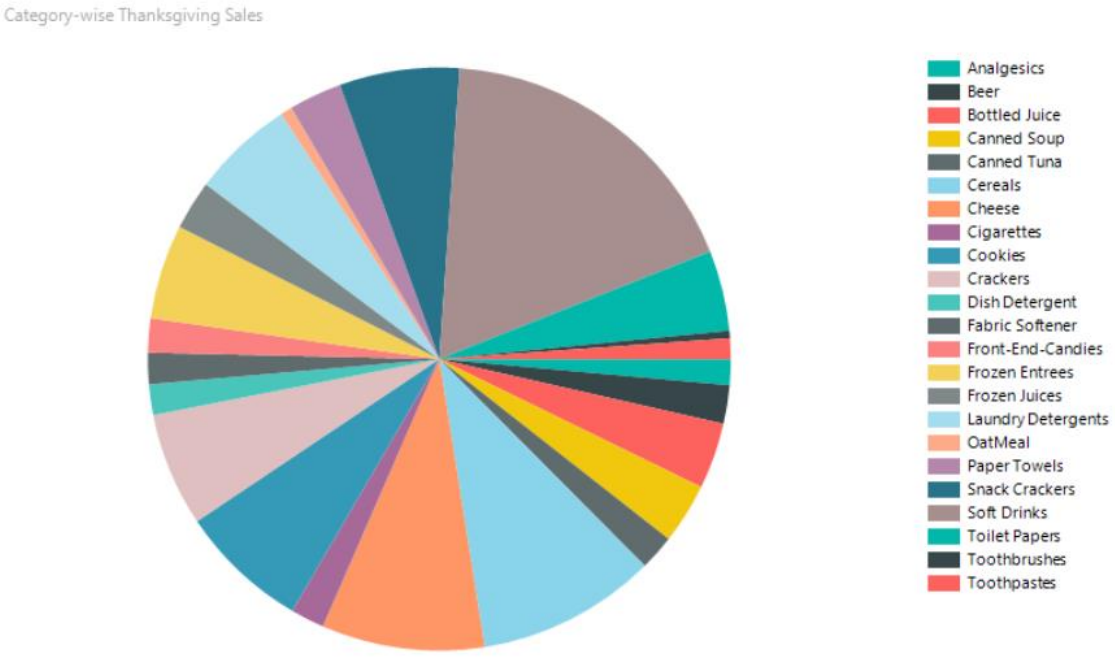


Figure 109: Thanksgiving Sales Chart – Category wise

Below line chart, shows the total Thanksgiving sale from the year 1989-1996.

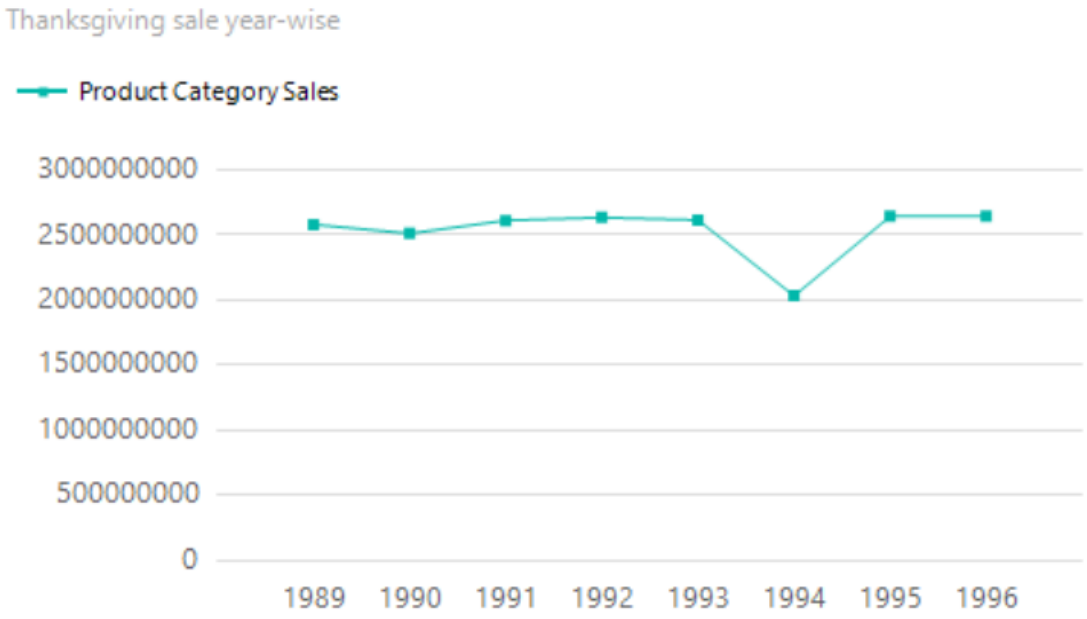


Figure 110: Thanksgiving Sales Chart - Year wise

2. What is the impact of age-wise regional demography on the sales of pharmacy products?

Firstly, a new project is created using "Analysis Service Multidimensional and Data Mining".

The steps followed after creating a multidimensional project are as follows:

A data source is created with the required database in it. Here we used our project's data warehouse database. New a data source view is generated where in all the tables (dimension and facts) are mapped to the "Included Objects" tab to be included further while making cube.

The view is checked for schema.

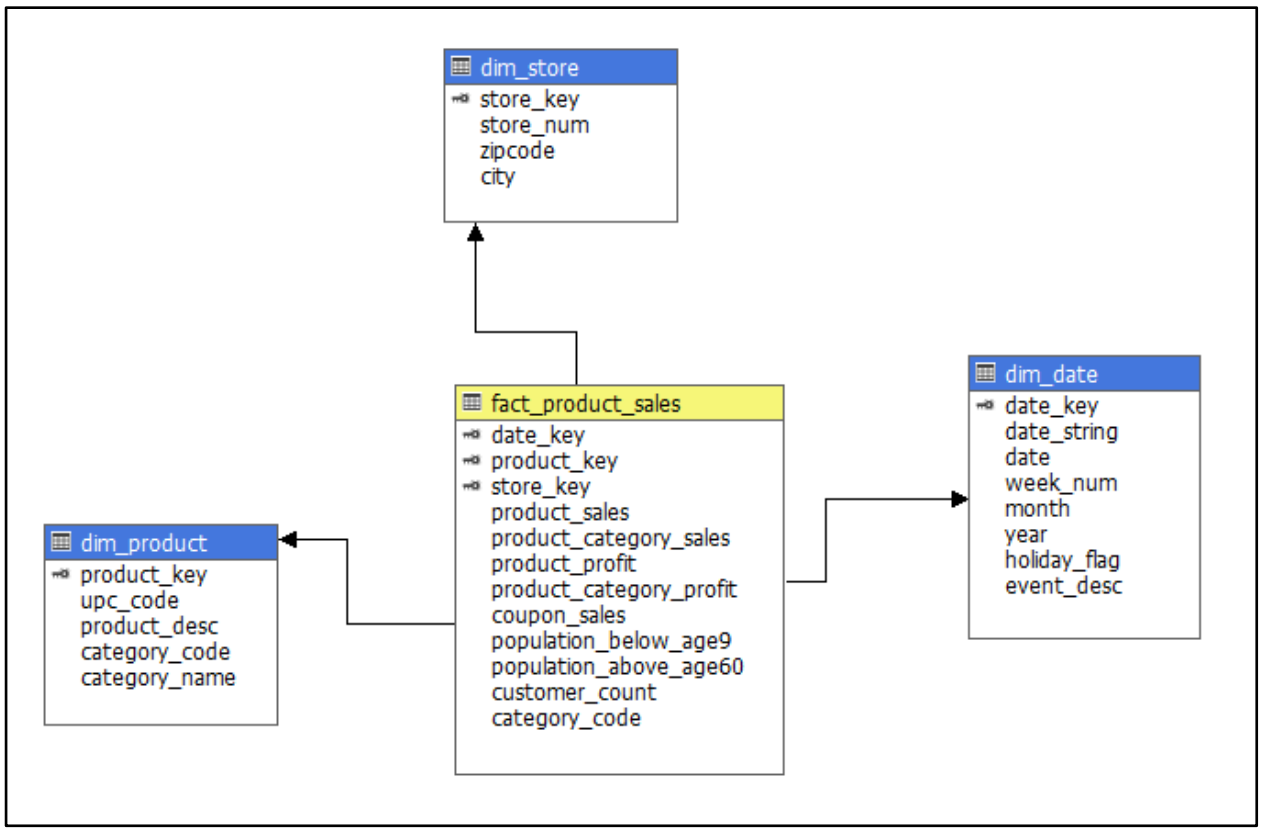


Figure 111: Cube - Demographics effect on Analgesics Sales

Now, is the time to create a multidimensional cube for the data source views. All the facts and dimension tables are selected to be included in the cubes. For this business question, hierarchies are created for dim_store, dim_date and dim_product to be used as attributes when browsing the cube.

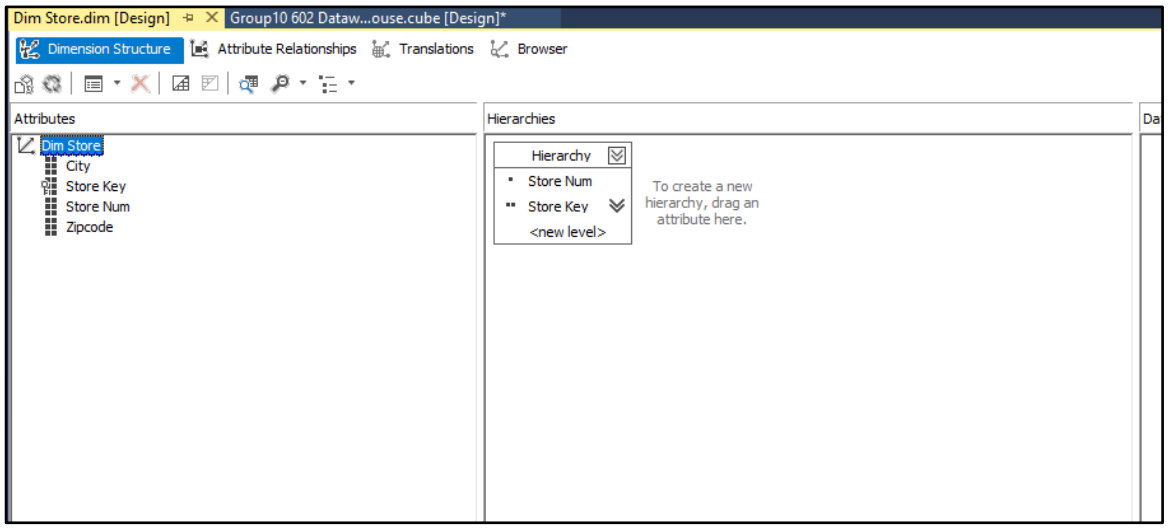


Figure 112: Store dimension hierarchy

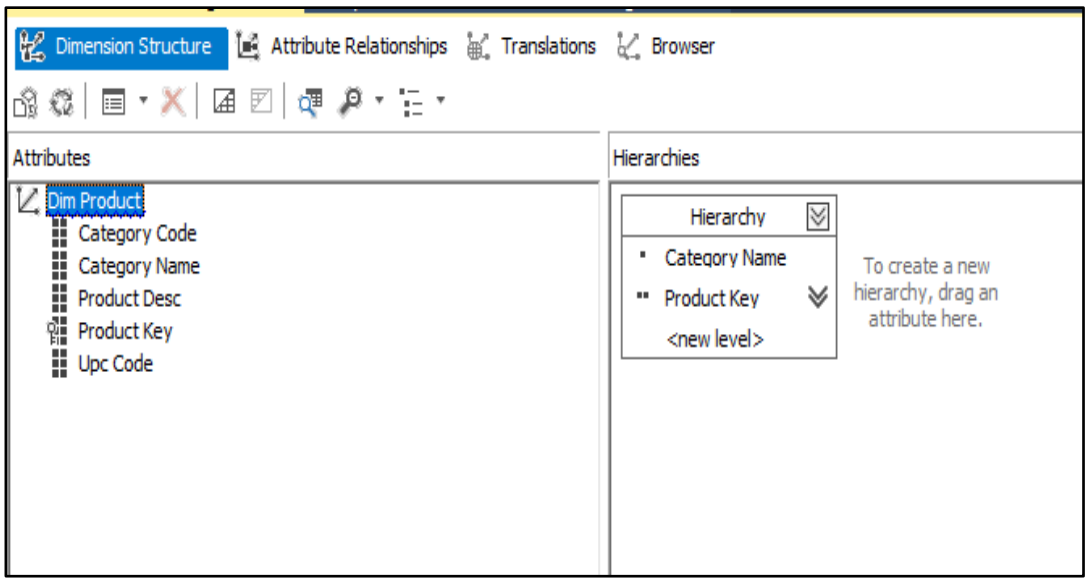


Figure 113: Product dimension hierarchy

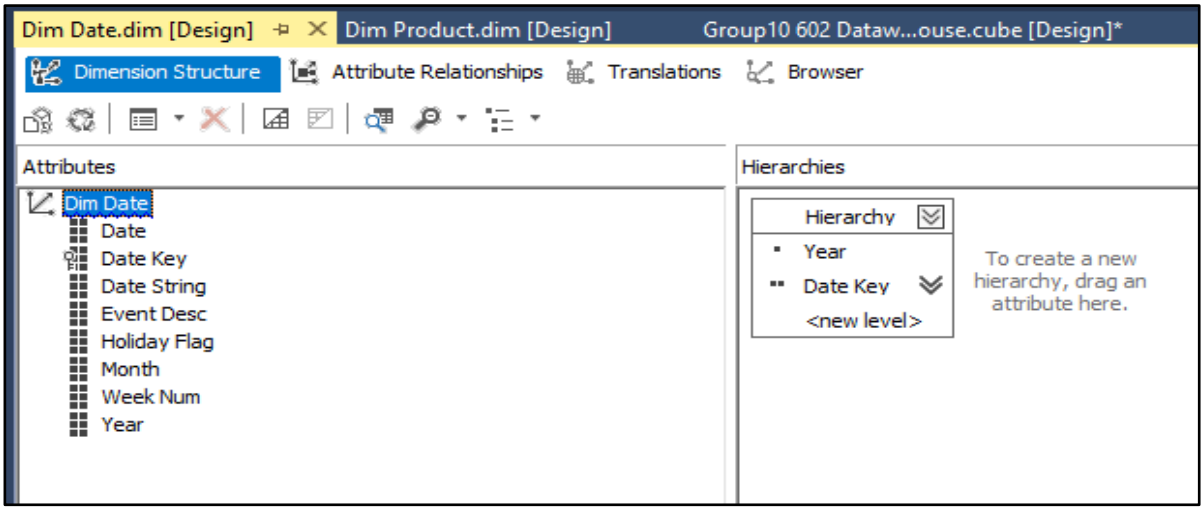


Figure 114: Date dimension hierarchy

Derived attributes are also created for 2 metrics to have a new calculated value pointed to the browser.

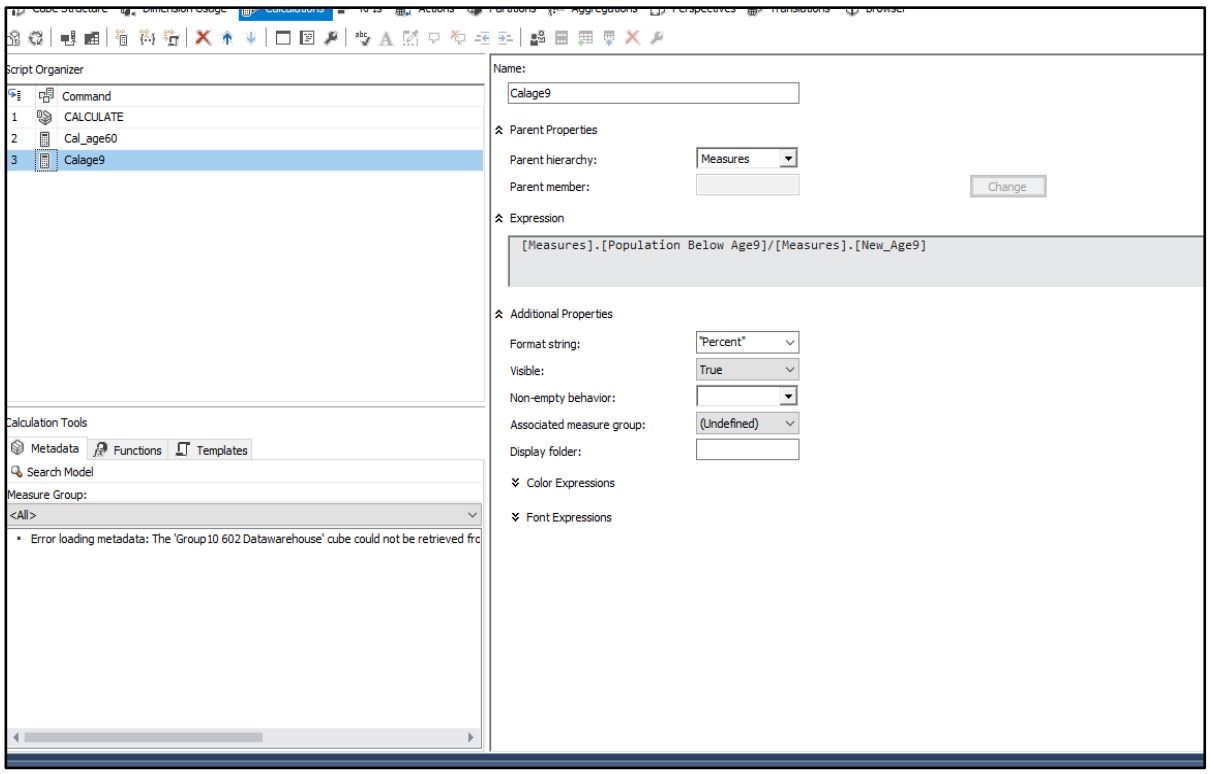


Figure 115: Calculated Measure – Age9

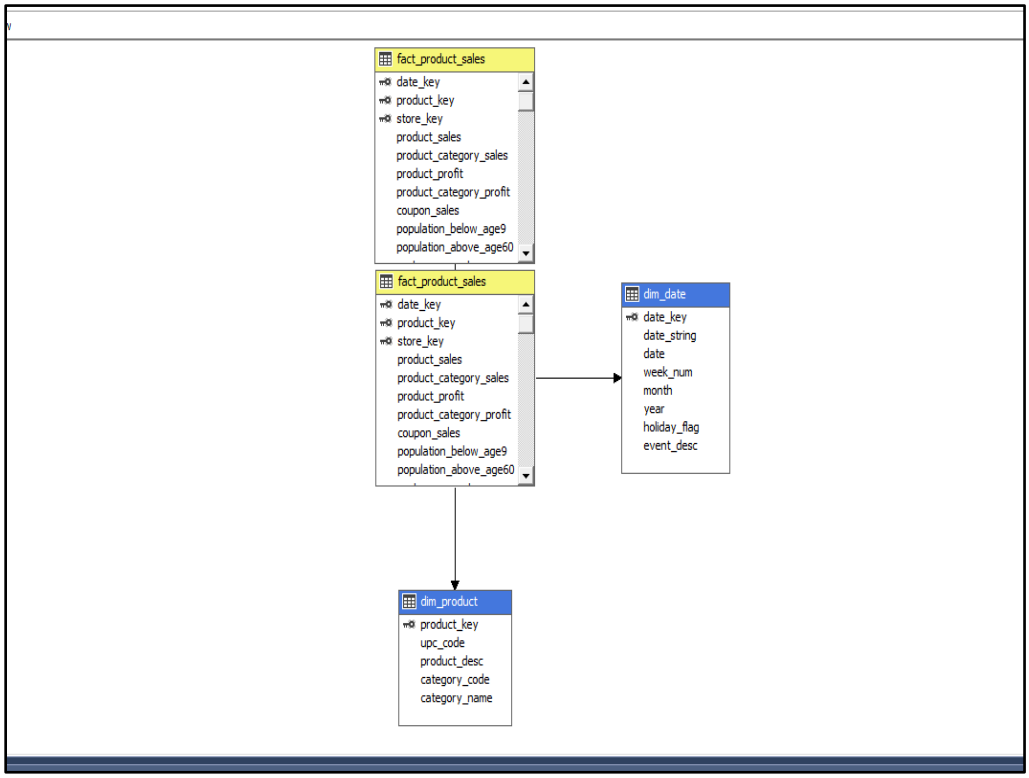


Figure 116: Cube after creating measure

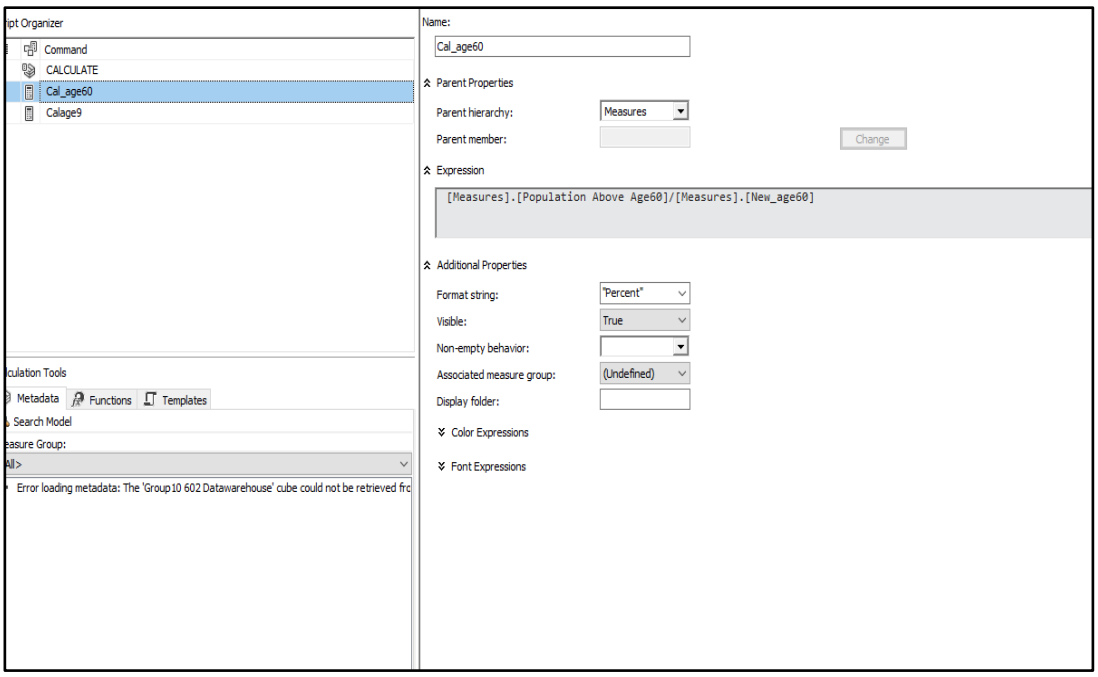


Figure 117: Calculated Measure – Age60

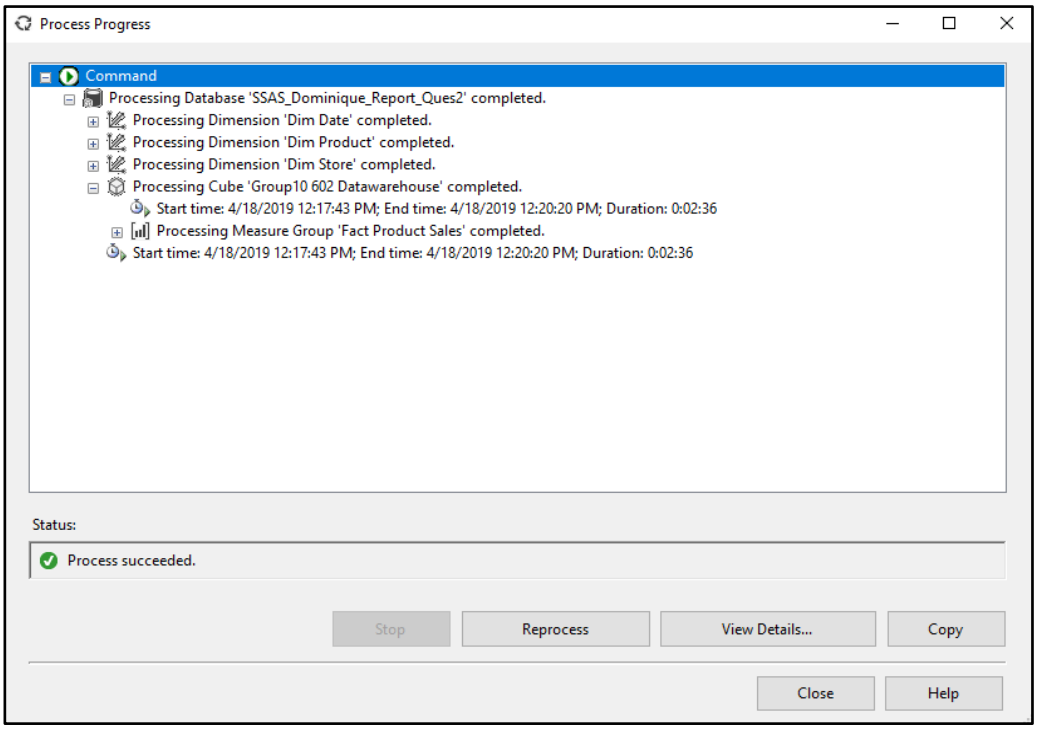
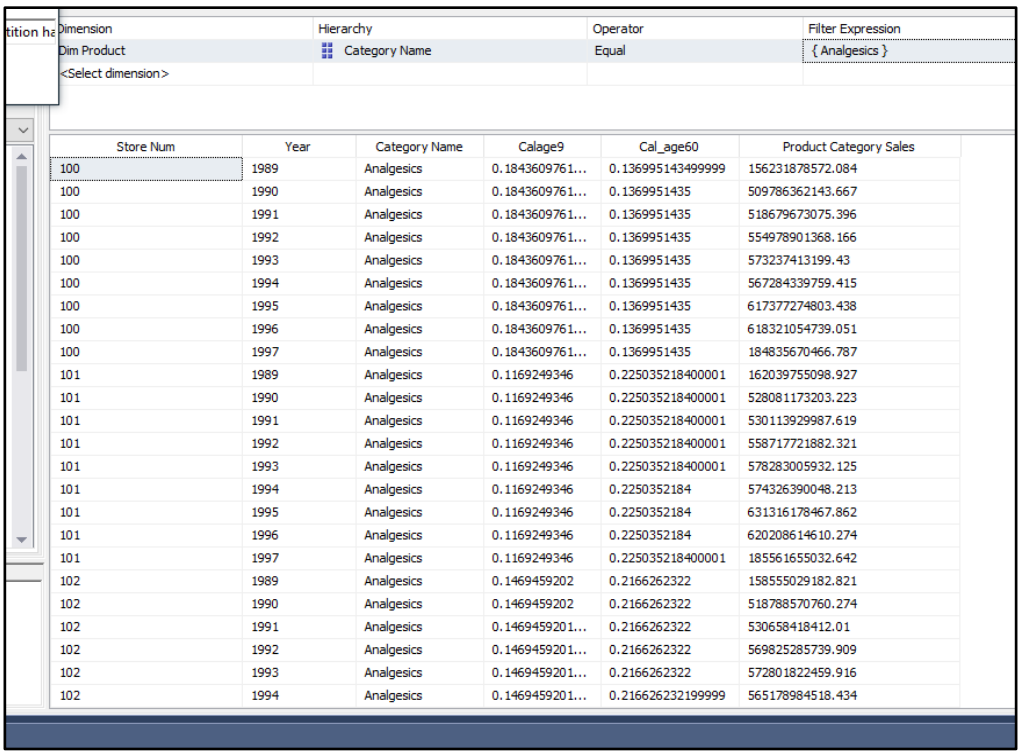


Figure 118: SSAS - Process Completed

This is the view of cube when browse in the visual studio 2015. Here a filter has been applied since we need to have a look at the products of Analgesics category based on demographics.

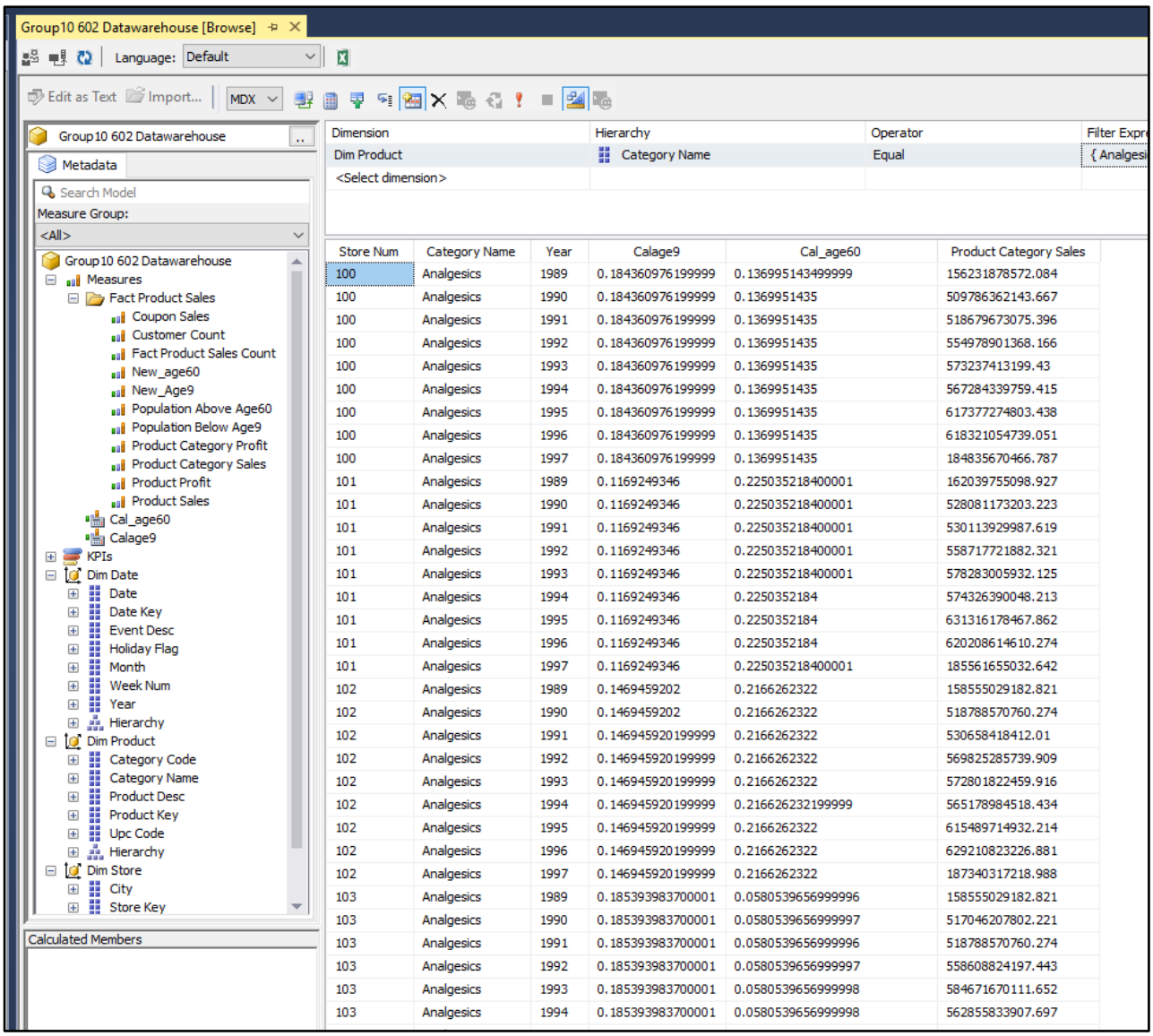


Dimension	Hierarchy	Operator	Filter Expression
Dim Product	Category Name	Equal	{Analgesics}
<Select dimension>			

Store Num	Year	Category Name	Calage9	Cal_age60	Product Category Sales
100	1989	Analgesics	0.1843609761...	0.136995143499999	156231878572.084
100	1990	Analgesics	0.1843609761...	0.1369951435	509786362143.667
100	1991	Analgesics	0.1843609761...	0.1369951435	518679673075.396
100	1992	Analgesics	0.1843609761...	0.1369951435	554978901368.166
100	1993	Analgesics	0.1843609761...	0.1369951435	573237413199.43
100	1994	Analgesics	0.1843609761...	0.1369951435	567284339759.415
100	1995	Analgesics	0.1843609761...	0.1369951435	617377274803.438
100	1996	Analgesics	0.1843609761...	0.1369951435	618321054739.051
100	1997	Analgesics	0.1843609761...	0.1369951435	184835670466.787
101	1989	Analgesics	0.1169249346	0.225035218400001	162039755098.927
101	1990	Analgesics	0.1169249346	0.225035218400001	528081173203.223
101	1991	Analgesics	0.1169249346	0.225035218400001	530113929987.619
101	1992	Analgesics	0.1169249346	0.225035218400001	558717721882.321
101	1993	Analgesics	0.1169249346	0.225035218400001	578283005932.125
101	1994	Analgesics	0.1169249346	0.2250352184	574326390048.213
101	1995	Analgesics	0.1169249346	0.2250352184	631316178467.862
101	1996	Analgesics	0.1169249346	0.2250352184	620208614610.274
101	1997	Analgesics	0.1169249346	0.225035218400001	185561655032.642
102	1989	Analgesics	0.1469459202	0.2166262322	158555029182.821
102	1990	Analgesics	0.1469459202	0.2166262322	518788570760.274
102	1991	Analgesics	0.1469459201...	0.2166262322	530658418412.01
102	1992	Analgesics	0.1469459201...	0.2166262322	569825285739.909
102	1993	Analgesics	0.1469459201...	0.2166262322	572801822459.916
102	1994	Analgesics	0.1469459201...	0.216626232199999	565178984518.434

Figure 119: Cube view in SSAS for effect of demographics on Analgesics Sales

After this the multidimensional cube is deployed over the SSAS server shown in the screenshot below. Browsing will present the screen as shown below. This can be viewed and used for analysis by anyone.



Store Num	Category Name	Year	Calage9	Cal_age60	Product Category Sales
100	Analgesics	1989	0.184360976199999	0.136995143499999	156231878572.084
100	Analgesics	1990	0.184360976199999	0.1369951435	509786362143.667
100	Analgesics	1991	0.184360976199999	0.1369951435	518679673075.396
100	Analgesics	1992	0.184360976199999	0.1369951435	554978901368.166
100	Analgesics	1993	0.184360976199999	0.1369951435	573237413199.43
100	Analgesics	1994	0.184360976199999	0.1369951435	567284339759.415
100	Analgesics	1995	0.184360976199999	0.1369951435	617377274803.438
100	Analgesics	1996	0.184360976199999	0.1369951435	618321054739.051
100	Analgesics	1997	0.184360976199999	0.1369951435	184835670466.787
101	Analgesics	1989	0.1169249346	0.225035218400001	162039755098.927
101	Analgesics	1990	0.1169249346	0.225035218400001	528081173203.223
101	Analgesics	1991	0.1169249346	0.225035218400001	530113929987.619
101	Analgesics	1992	0.1169249346	0.225035218400001	558717721882.321
101	Analgesics	1993	0.1169249346	0.225035218400001	578283005932.125
101	Analgesics	1994	0.1169249346	0.2250352184	574326390048.213
101	Analgesics	1995	0.1169249346	0.2250352184	631316178467.862
101	Analgesics	1996	0.1169249346	0.2250352184	620208614610.274
101	Analgesics	1997	0.1169249346	0.225035218400001	185561655032.642
102	Analgesics	1989	0.1469459202	0.2166262322	158555029182.821
102	Analgesics	1990	0.1469459202	0.2166262322	518788570760.274
102	Analgesics	1991	0.146945920199999	0.2166262322	530658418412.01
102	Analgesics	1992	0.146945920199999	0.2166262322	569825285739.909
102	Analgesics	1993	0.146945920199999	0.2166262322	572801822459.916
102	Analgesics	1994	0.146945920199999	0.216626232199999	565178984518.434
102	Analgesics	1995	0.146945920199999	0.2166262322	615489714932.214
102	Analgesics	1996	0.146945920199999	0.2166262322	629210823226.881
102	Analgesics	1997	0.146945920199999	0.2166262322	187340317218.988
103	Analgesics	1989	0.185393983700001	0.0580539656999996	158555029182.821
103	Analgesics	1990	0.185393983700001	0.0580539656999997	517046207802.221
103	Analgesics	1991	0.185393983700001	0.0580539656999996	518788570760.274
103	Analgesics	1992	0.185393983700001	0.0580539656999997	558608824197.443
103	Analgesics	1993	0.185393983700001	0.0580539656999998	584671670111.652
103	Analgesics	1994	0.185393983700001	0.0580539656999998	562855833907.697

Figure 120: Cube deployed

3. What is the effect of coupon promotions on the sale of different products store-wise? Do coupons impact product sales?

To address this business question, a ***"materialized view"*** is created in data warehouse named as PROMO_FACT. The data source is selected as *group10_602_datawarehouse.*

In the query designer, below query is used. The year is used as a dynamic selection parameter.

SQL Query:

```
SELECT
```

```
  PROMO_FACT.[year]
  ,PROMO_FACT.category_name
  ,PROMO_FACT.store_num
  ,PROMO_FACT.Avg_coupon_sales
  ,PROMO_FACT.Avg_product_category_sales
FROM
  PROMO_FACT WHERE WHERE (PROMO_FACT.[year]= @year)
```

While designing the table year is used as a page field and grouped by store_num and category_name. The values Avg_product_sales and Avg_product_category_sales is added in the details.

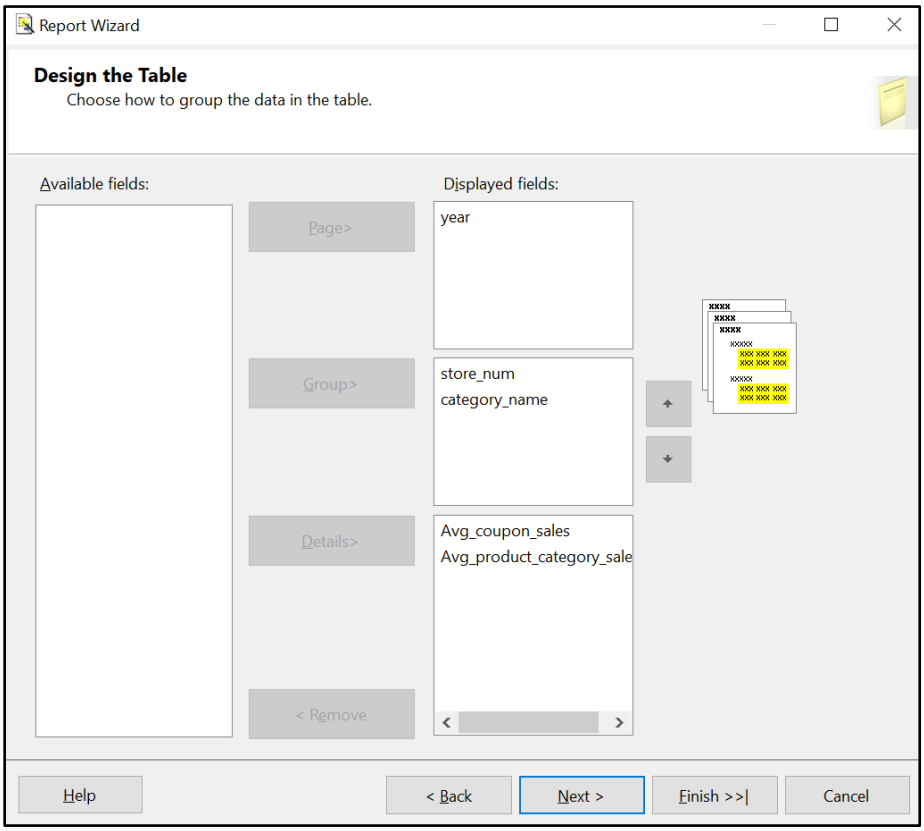


Figure 121: Table design for Coupon Promotion Report

When the report is previewed, the year can be entered to fetch the related coupon promotion report can be fetched.

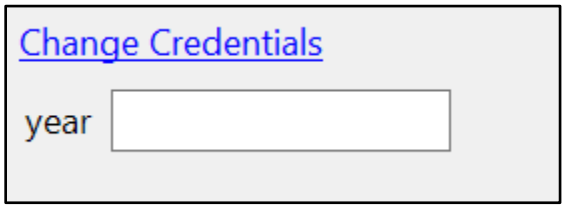


Figure 122: SSRS - Filter by year

The report is deployed to http://infodata16.mbs.tamu.edu/ReportServer in the folder "602_Group10" as "Coupon Promotions Effect Report".
On entering the year, below dashboard report can be viewed as below.

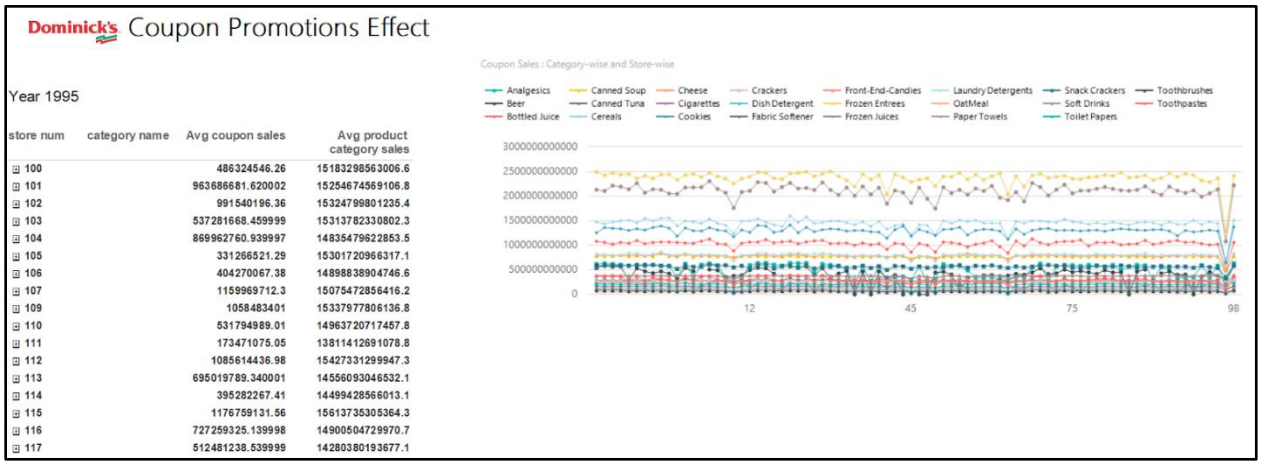


Figure 123: Coupon Promotions Effect Dashboard

The below graph shows, the coupon sales based on the category name for the stores.

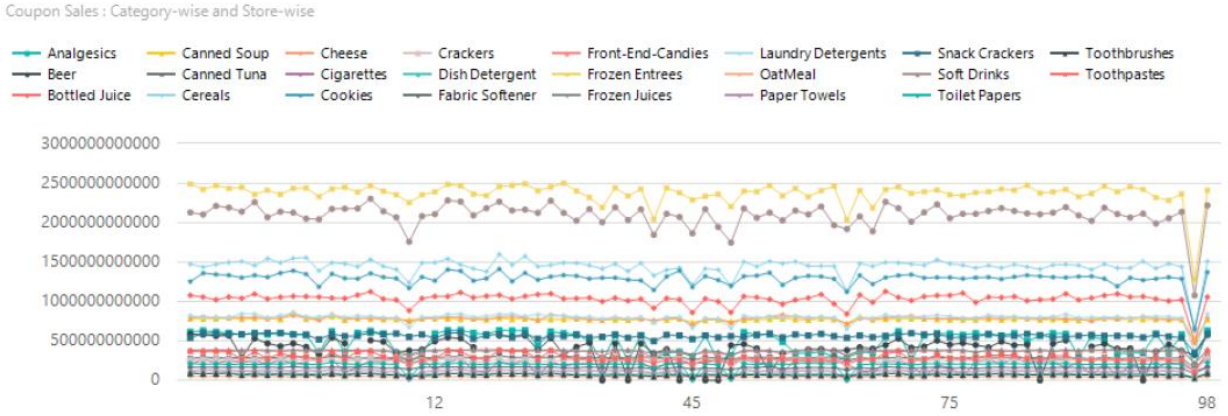


Figure 124: Coupon Promotion Effect Graph

The report can be drilled down for every store. On expanding the coupon sales for each category is shown.

Year 1995

store num	category name	Avg coupon sales	Avg product category sales
⊟ 100		**486324546.26**	**15183298563006.6**
	⊞ Analgesics	33733889.6300001	617377274803.434
	⊞ Beer	25653791.3199999	589010677783.597
	⊞ Bottled Juice	23013729.84	1080218116354.84
	⊞ Canned Soup	23991287.4500001	785096489731.395
	⊞ Canned Tuna	11183035.27	279563274171.69
	⊞ Cereals	12075392.68	1474169660309.45
	⊞ Cheese	30057279.3400001	792296018773.317
	⊞ Cigarettes	15824914.16	203720764874.371
	⊞ Cookies	44921533.7199998	1253033202953.31
	⊞ Crackers	11009176.91	255653709351.416
	⊞ Dish	10646292.5	89298835332.2606
	⊞ Fabric Softener	12766251.03	167172250469.662
	⊞ Front-End-Candies	20847906.92	384070073708.808
	⊞ Frozen Entrees	37174526.0899998	2490618149889.82

Figure 125: Coupon Promotion Effect Report

4. What is the customer increase or decrease over the years based on the store?

This business question has been done with the help of Report Builder 3.0.

SQL Query:

```
SELECT convert(int, dim_store.store_num) as store_num, dim_date.year,
Fact_product_sales.customer_count
FROM dim_store
INNER JOIN
Fact_product_sales ON fact_product_sales.store_key = dim_store.store_key
INNER JOIN
Dim_date ON dim_date_date_key = fact_product_sales.date_key where store_num =
(@store_number) group by dim_date.year, dim_store.store_num,
fact_product_sales.customer_count
```

The datasource is selected from which data needs to be selected which is Dominick's data warehouse created by our team for this project. SQL query is written in "Edit as text" field in design a query tab. Columns that are needed are selected from the tables as required.

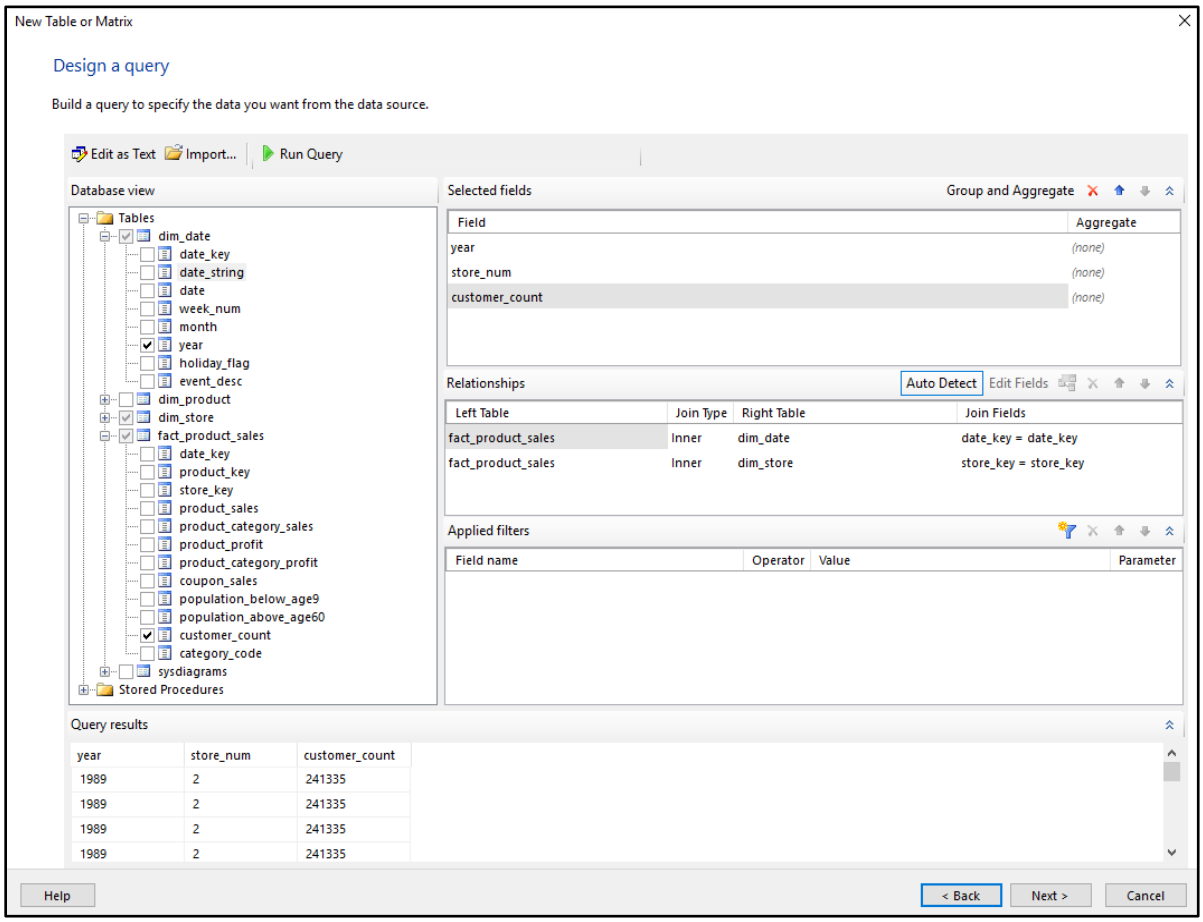


Figure 126: Report Builder - Query design

The columns are then separated on row and column groups in "Arrange Fields". Show subtotal and grand total is unchecked.

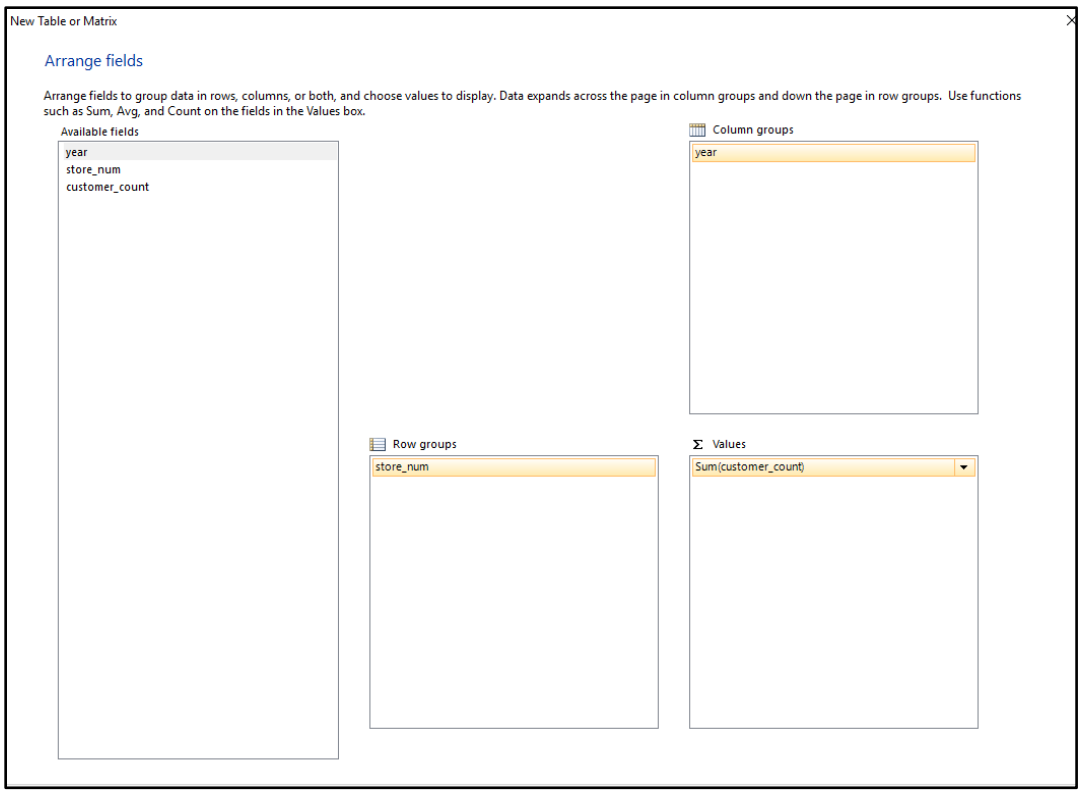


Figure 127: Report Builder - Table design

Run time parameters are created by mentioning in dataset and defining the parameter value. The query is ran to give below table and charts.

year	100	101	102	103	104	105	106	107	109	110	111	112
1989	6768210	5883932	6640027	5198718	3484986	5620611	3803054	4858258	5213817	4513678	5939509	6217134
1990	21900509	19348703	22256134	17381493	10654212	18232860	10032591	16326491	17584247	14149456	20425493	20011046
1991	23616088	20740483	23254560	18666894	10813030	18303285	10075946	17831614	18423253	15355470	22611354	19653206
1992	21596887	20003027	22032418	17826683	10578543	16680599	9289842	18240813	18592318	13303380	20698082	18466513
1993	20674543	18890692	20695040	16014913	10077840	15451065	8658316	17582440	18159288	11946840	19081018	17426169
1994	18551530	16783911	18470363	15682746	9295883	12948636	8110326	17089711	17282089	10342908	17161787	15905411
1995	18045211	16518997	18407495	14722804	11842135	11695843	7981627	17334234	16943245	10206685	17432128	17357600
1996	18255884	16148607	18467876	14283234	12758274	11675147	7796969	18422709	19882271	10641420	16272278	18593026
1997	4775997	3779725	4505010	3435763	2836770	2706898	1875158	4569416	4974618	2650665	4502554	4527136

Figure 128: Report Builder - Report

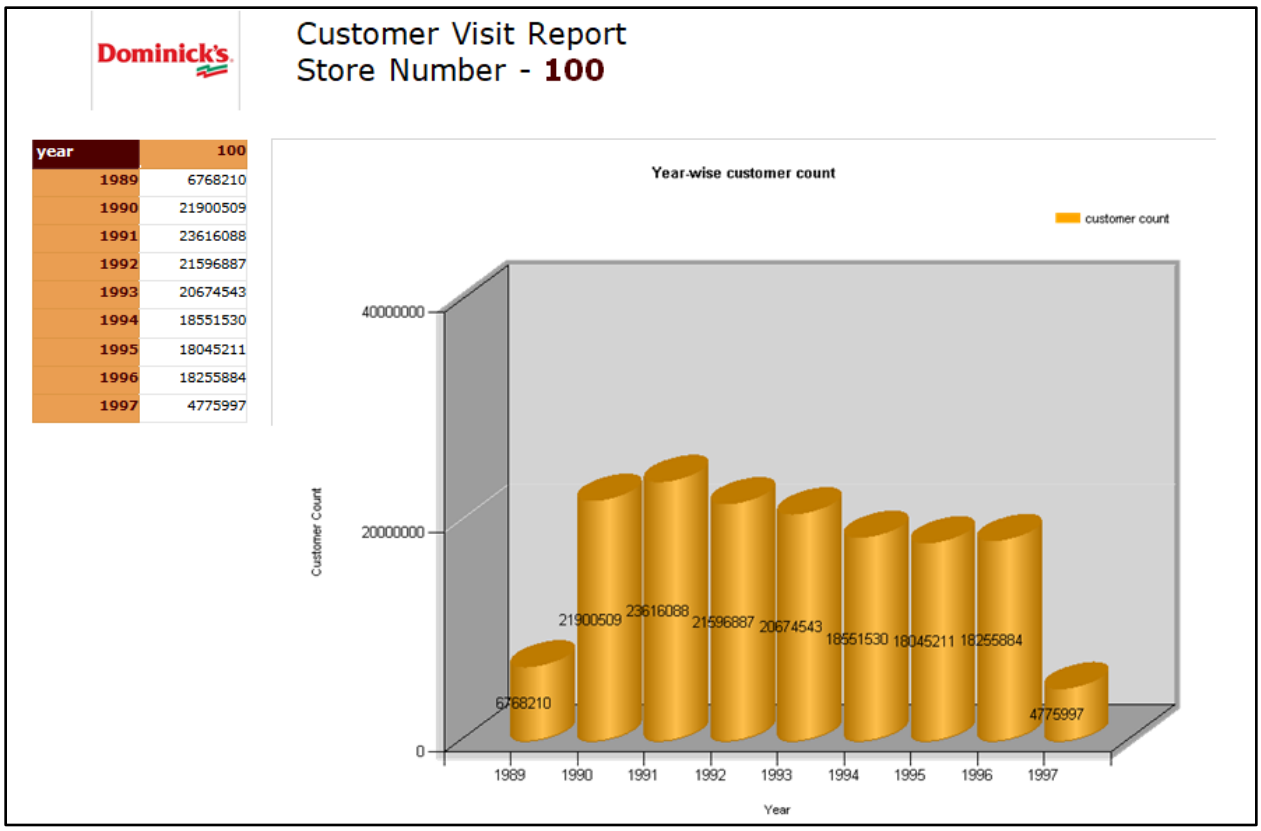


Figure 129: Report Builder - Dashboard

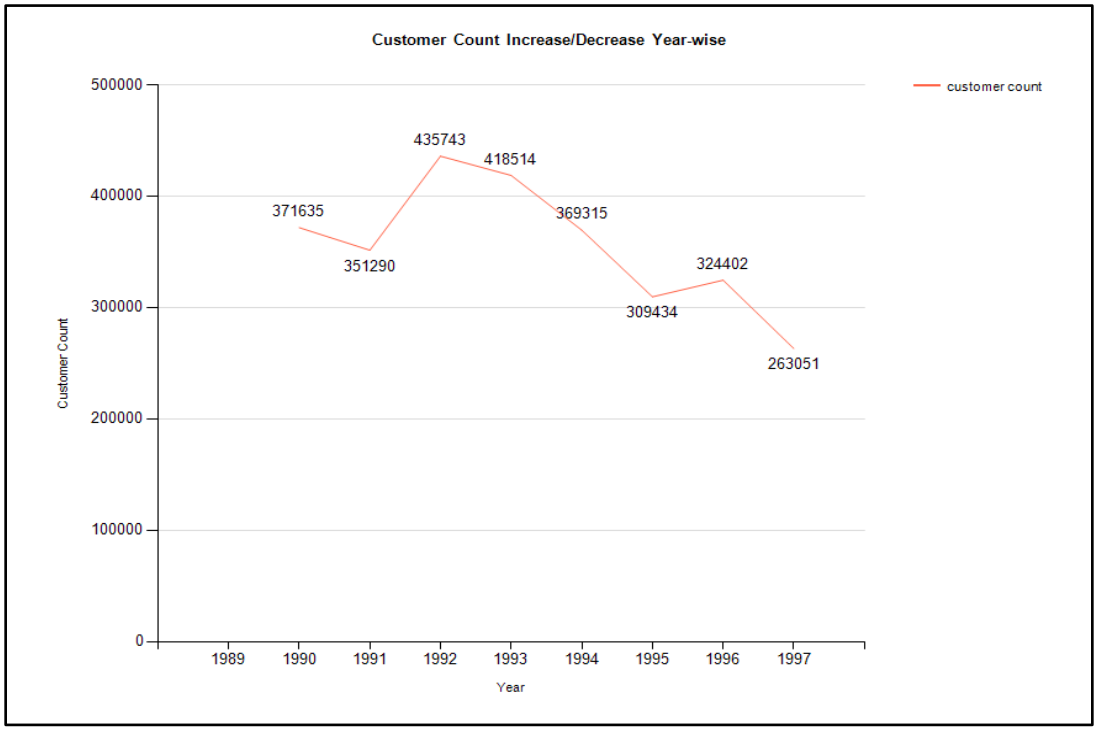


Figure 130: Report Builder - Customer Increase / Decrease Graph

5. How are the product category profits changing in every store over the years? What are the product categories with the highest and least profits?

In order to answer this question we used SSRS on top of SSAS. Firstly, to create the SSAS project the respective data source, data source view and the cube was generated with the required mappings. Also, the dimension table and fact tables were utilized for this purpose. For the cube the hierarchies were defined as seen below.

The hierarchy for dim_date followed the order year, month and week num. We need this to aggregate the data by year and then further drill down if necessary.

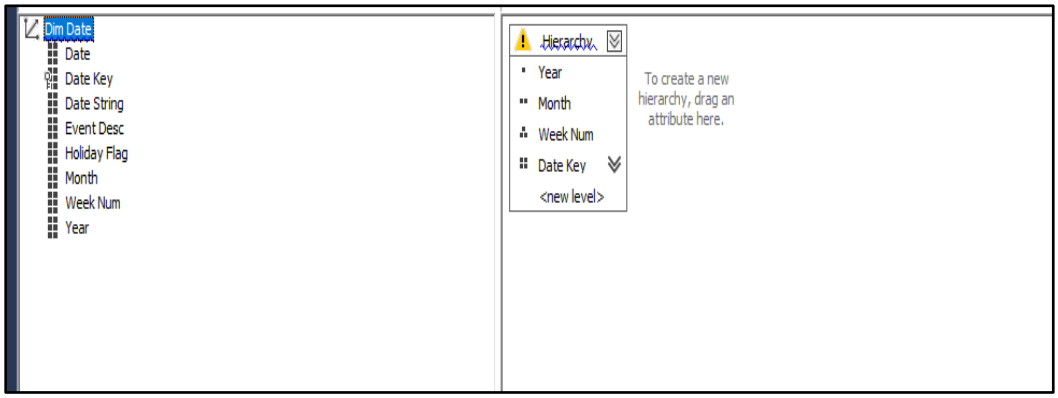


Figure 131: SSAS - Date dimension hierarchy

The dim_store hierarchy just had the store num as required by the business question.

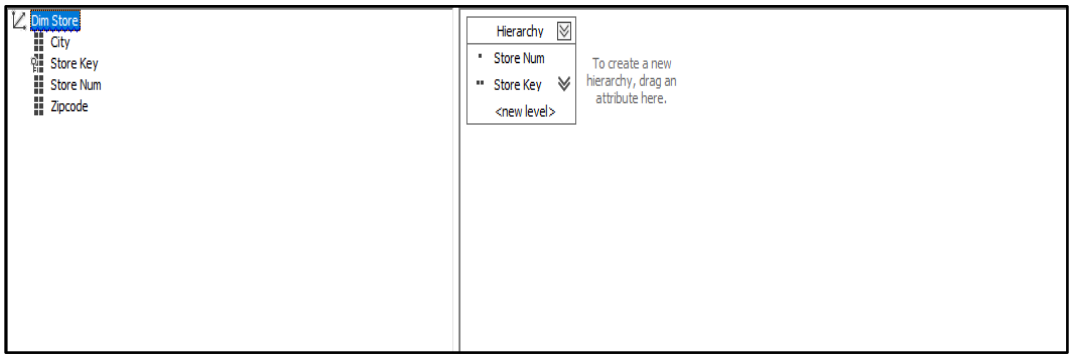


Figure 132: SSAS - Store dimension hierarchy

The dim_product hierarchy constituted of category name.

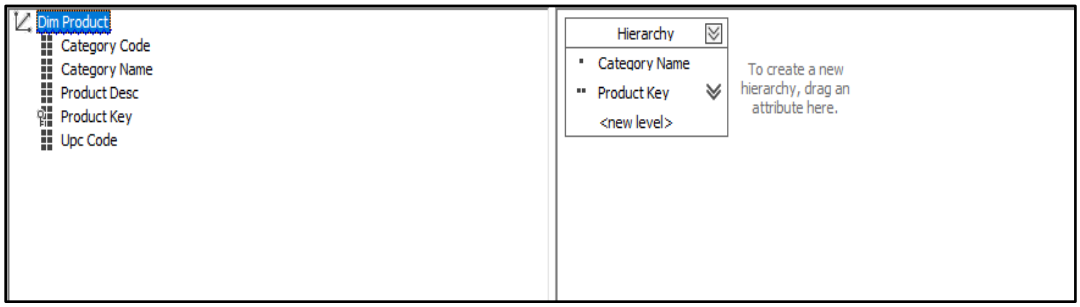


Figure 133: SSAS - Product dimension hierarchy

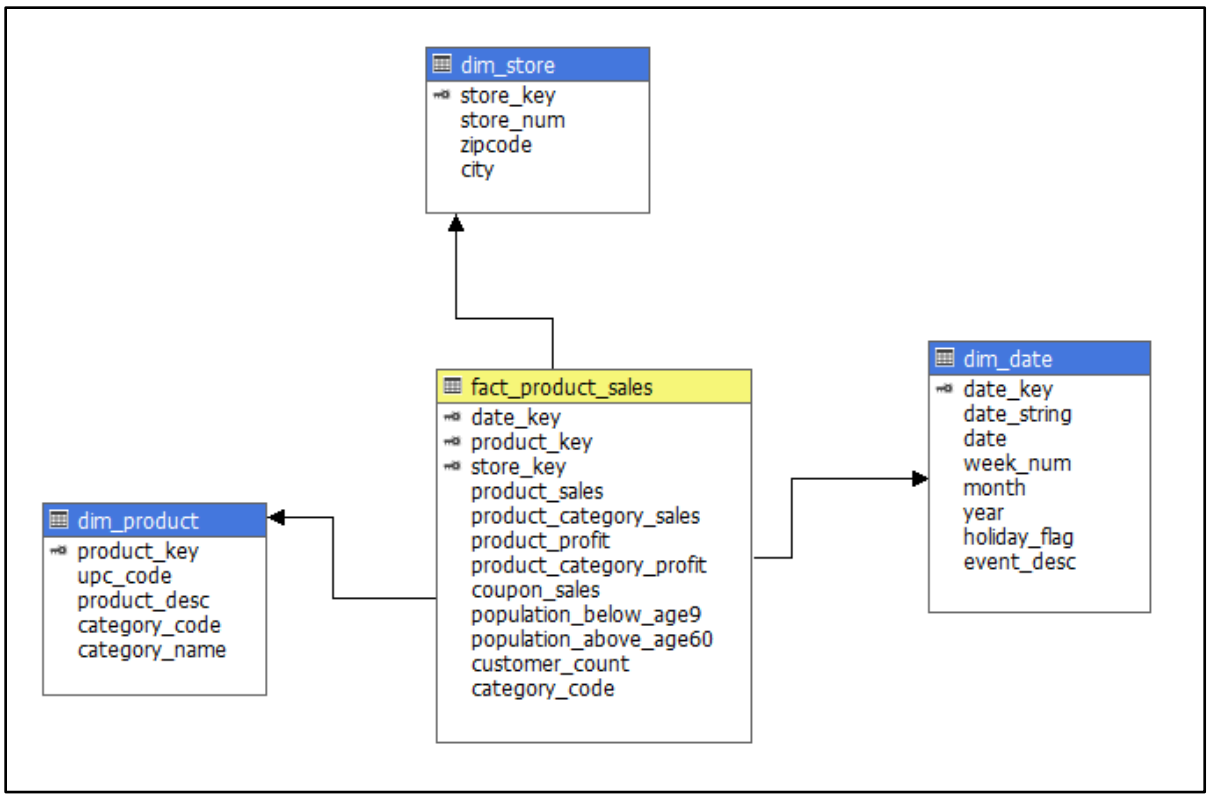


Figure 134: SSRS on top of SSAS - Cube

Now, to calculate the product category profits changing in every store over the years a new calculated member was added. The aggregate function for Product Profit in the properties was summed, another new measure wherein the aggregate function for Product Profit in the properties was set to count was added.
Further, a calculated measure called Calculated Product Profit was introduced which made use of the following expression-
[Measures].[Product Profit]/[Measures].[Count Product Profit]
The Format string was also set to percent. This is done to aggregate the profit percentage value coming in from the fact table (product_profit).

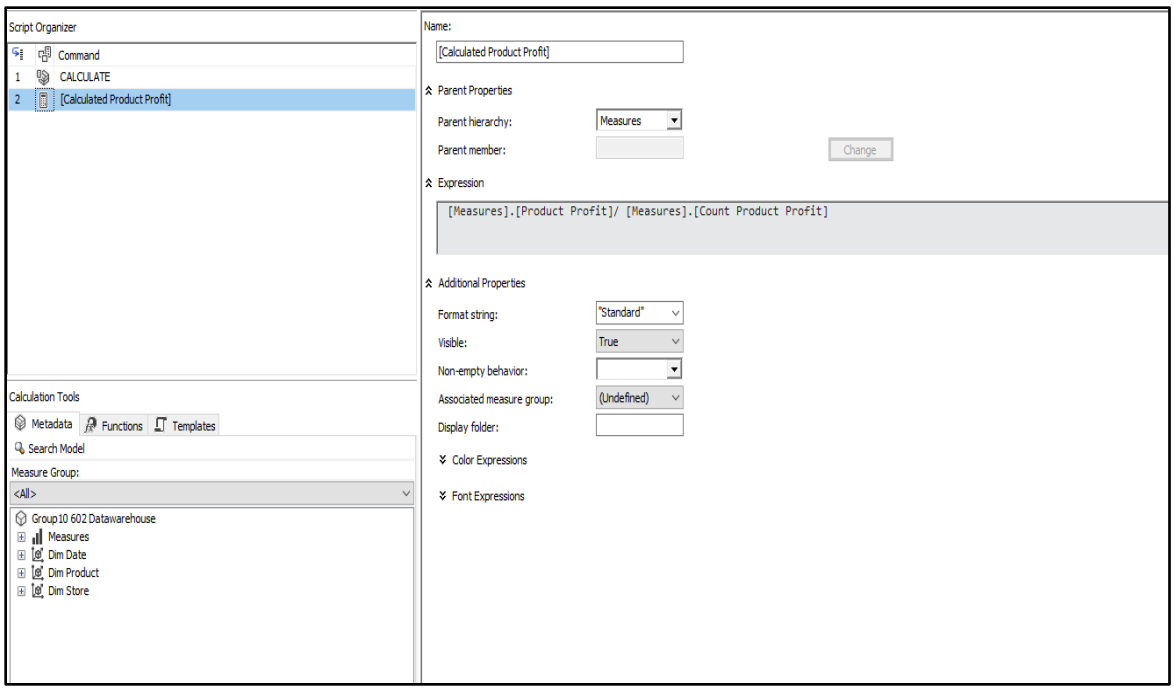
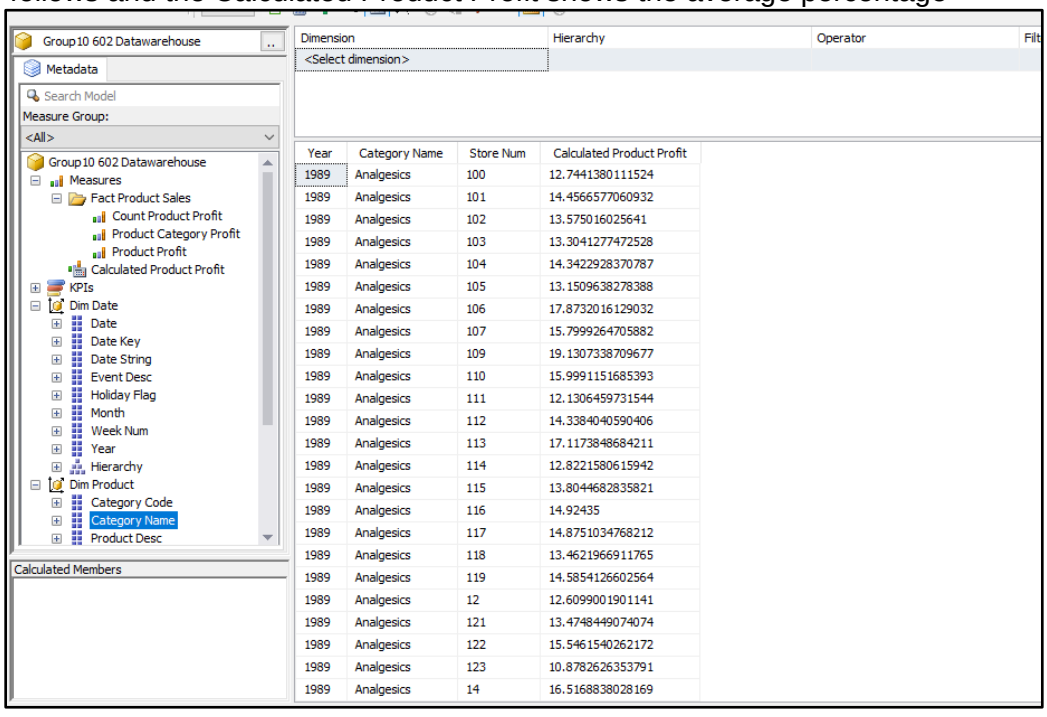


Figure 135: SSAS - Calculated Measure (Profit Percent)

Finally, the cube was processed and deployed. The query results after execution in the browser were as follows and the Calculated Product Profit shows the average percentage-



Year	Category Name	Store Num	Calculated Product Profit
1989	Analgesics	100	12.7441380111524
1989	Analgesics	101	14.4566577060932
1989	Analgesics	102	13.575016025641
1989	Analgesics	103	13.3041277472528
1989	Analgesics	104	14.3422928370787
1989	Analgesics	105	13.1509638278388
1989	Analgesics	106	17.8732016129032
1989	Analgesics	107	15.7999264705882
1989	Analgesics	109	19.1307338709677
1989	Analgesics	110	15.9991151685393
1989	Analgesics	111	12.1306459731544
1989	Analgesics	112	14.3384040590406
1989	Analgesics	113	17.1173848684211
1989	Analgesics	114	12.8221580615942
1989	Analgesics	115	13.8044682835821
1989	Analgesics	116	14.92435
1989	Analgesics	117	14.8751034768212
1989	Analgesics	118	13.4621966911765
1989	Analgesics	119	14.5854126602564
1989	Analgesics	12	12.6099001901141
1989	Analgesics	121	13.4748449074074
1989	Analgesics	122	15.5461540262172
1989	Analgesics	123	10.8782626353791
1989	Analgesics	14	16.5168838028169

Figure 136: SSAS - Cube browsing

When the project was deployed on the Microsoft SQL Server was deployed the cube showed the following result:

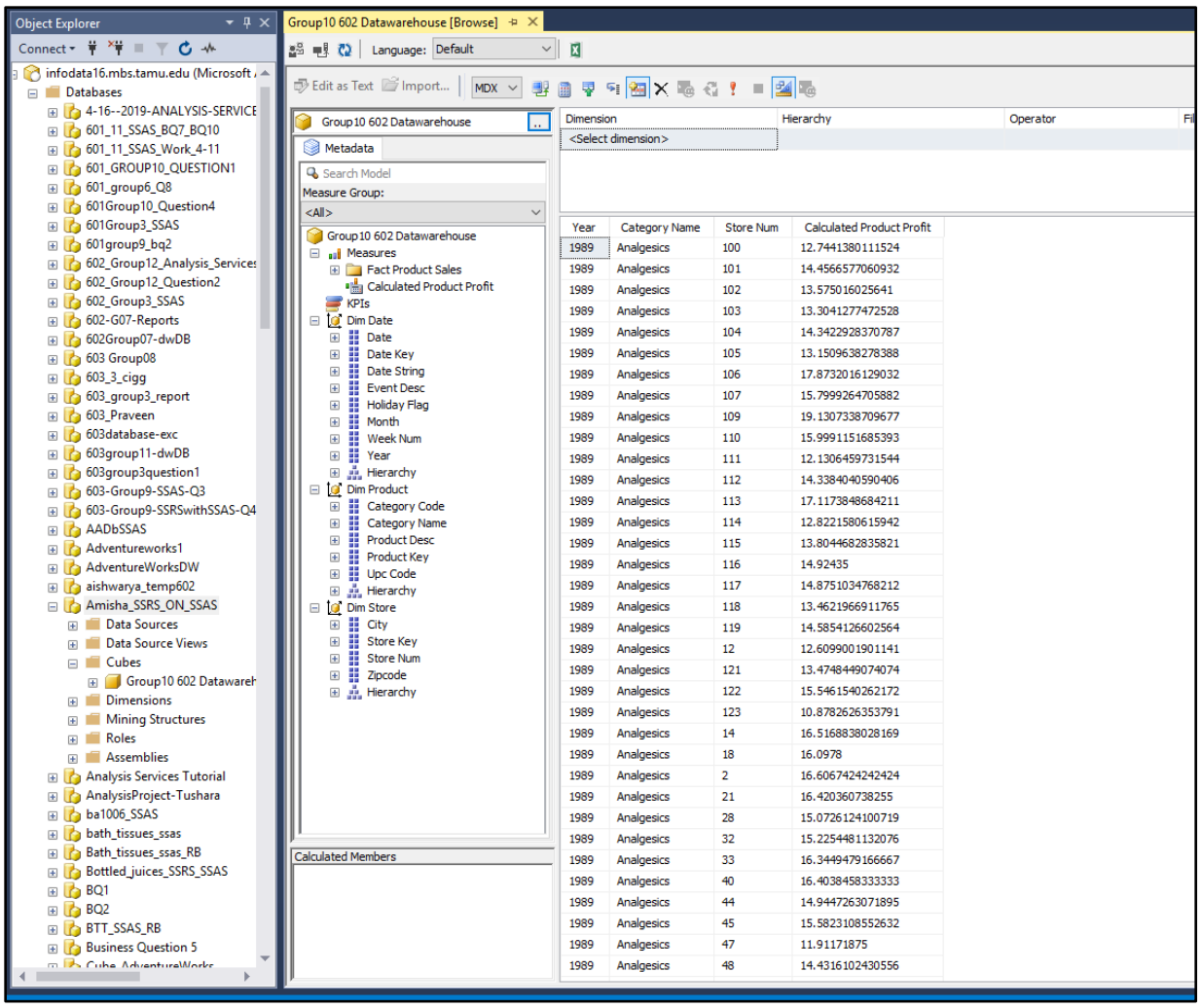


Figure 137: Cube deployed

After this to implement SSRS on top of SSAS, we choose the Report Wizard in Visual Studio. Now, the data source is chosen as Microsoft SQL Server Analysis Server.

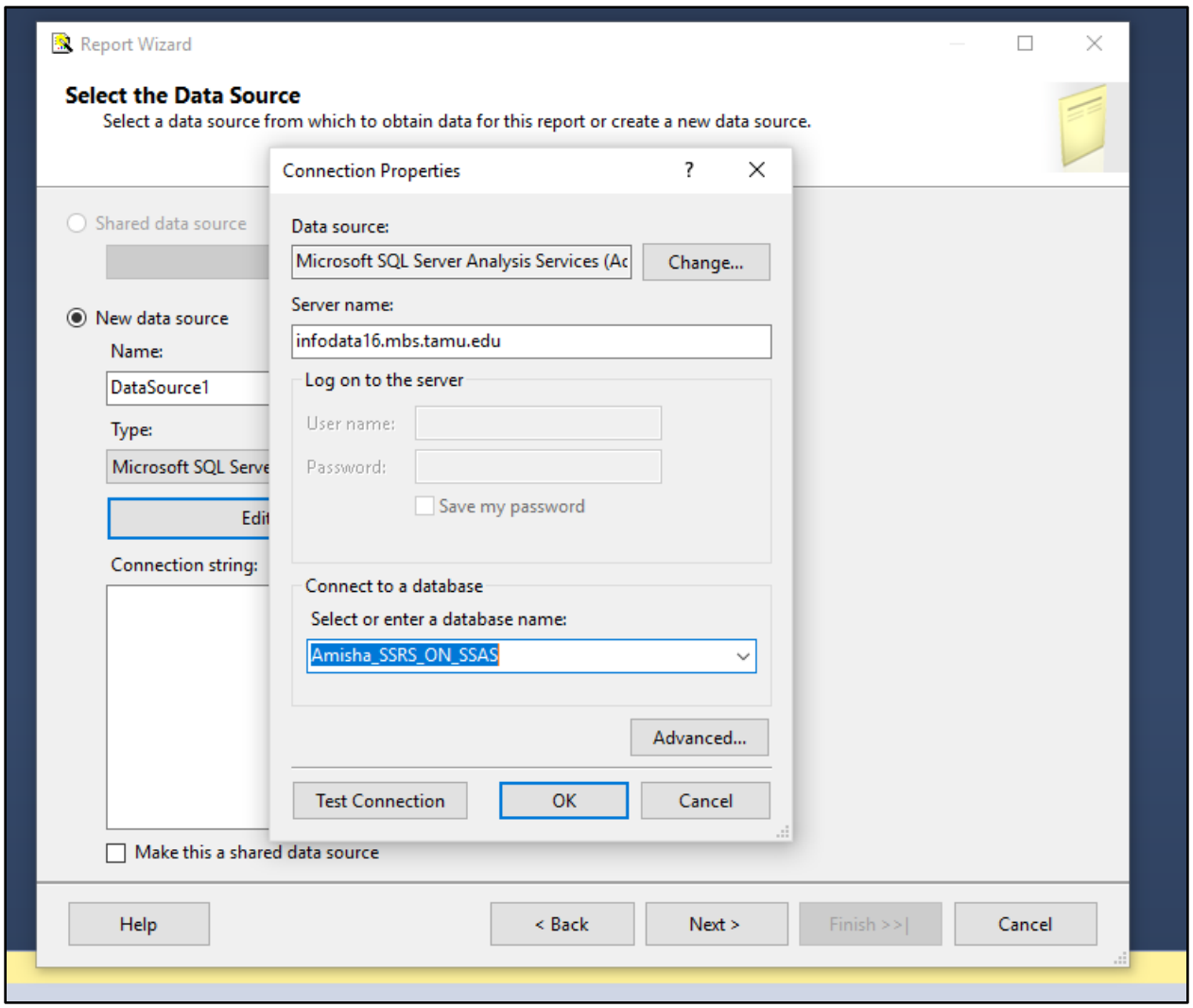


Figure 138: SSRS - Source Selection

The Query Designer then pops up and we can input out fields. The Query is built automatically after this step.

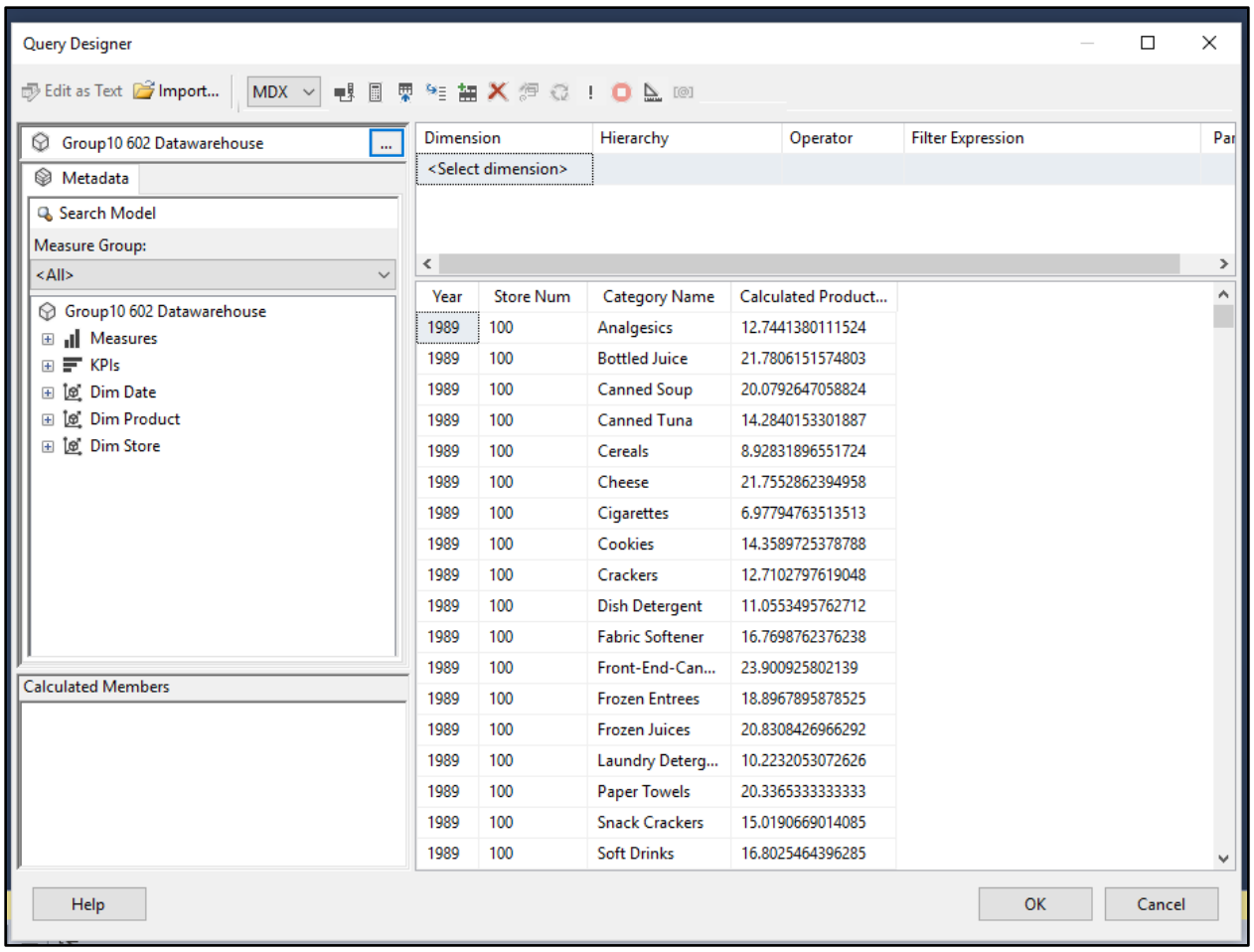


Figure 139: SSRS on top SSAS - Query Designer

SQL Query:

```
SELECT NON EMPTY { [Measures].[Calculated Product Profit] } ON COLUMNS, NON EMPTY { ([Dim Date].[Year].[Year].ALLMEMBERS * [Dim Store].[Store Num].[Store Num].ALLMEMBERS * [Dim Product].[Category Name].[Category Name].ALLMEMBERS ) } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_VALUE, MEMBER_UNIQUE_NAME ON ROWS FROM [Group10 602 Datawarehouse] CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR, FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS
```

Then the tabular design is chosen and the respective fields are dragged in. Drill down has also been enabled. Year, Store_Num and Category_name were dragged into group as per the business question and then Calculated_Product_Profit was dragged into the details.

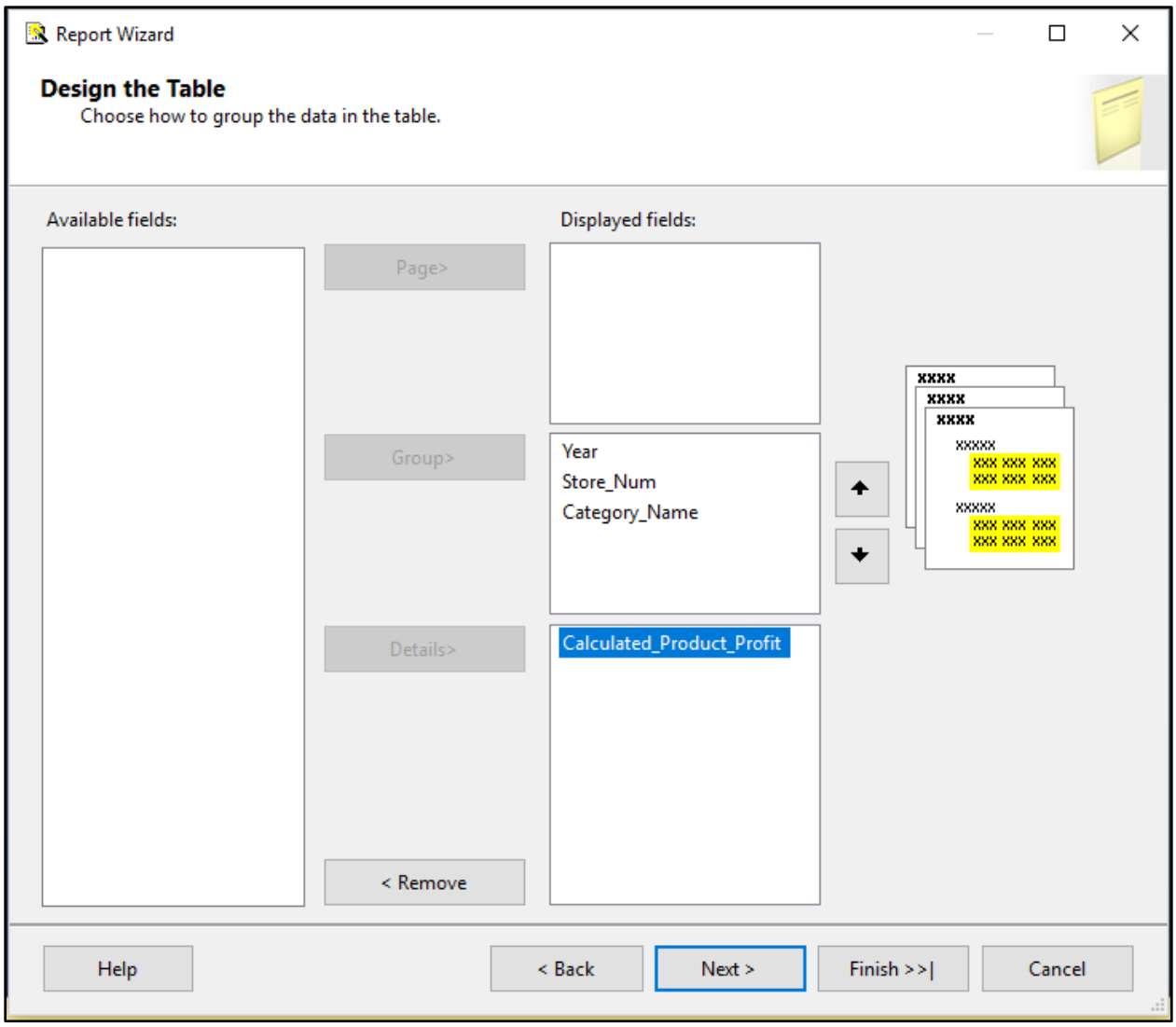


Figure 140: SSRS on top of SSAS - Table design

Finally, the profit percentage report in SSRS was generated as shown below-

Figure 141: Product Profit Report Dashboard

The table allows drill down and is viewed as below:

Year	Category Name	Calculated Product Profit
⊟ 1989		
	⊟ Analgesics	
		14.4566577060932
	⊟ Bottled Juice	
		21.2605167322835
	⊟ Canned Soup	
		16.6348341232227
	⊞ Canned Tuna	
	⊞ Cereals	
	⊞ Cheese	

Figure 142: Product Profit report

The first graph which is a line graph shows the profit percent for Product Category VS Year-

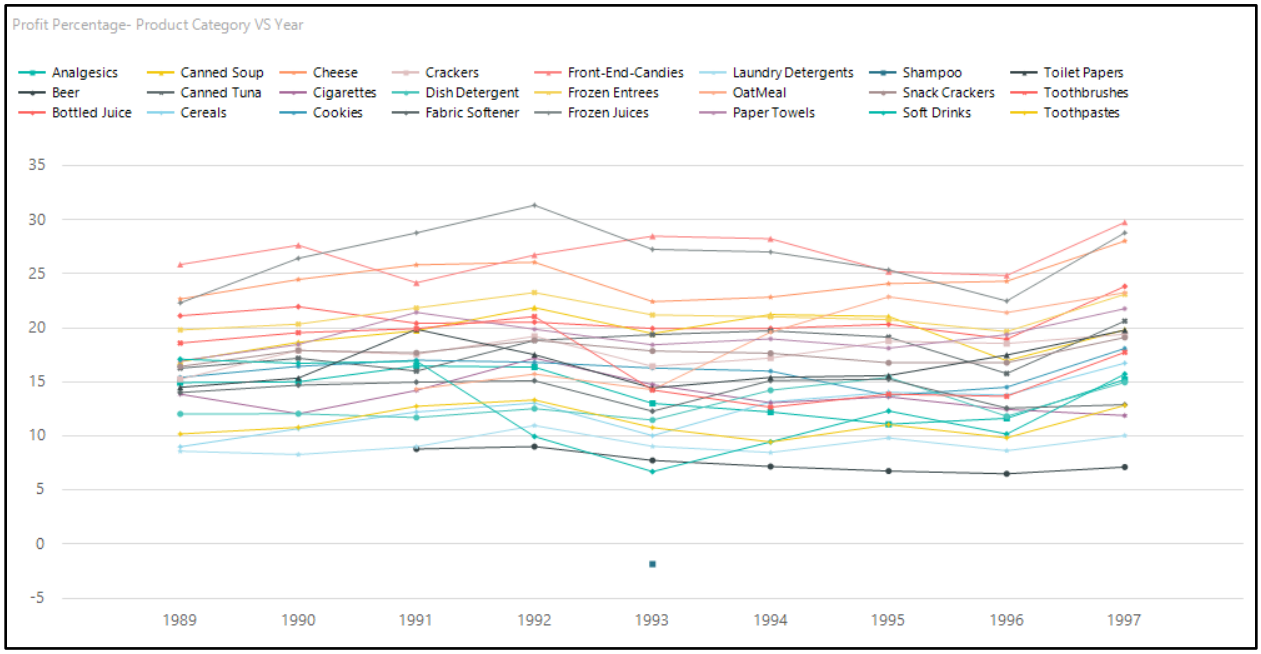


Figure 143: Product Profit graph – Product Category vs Year

The second graph which is a line graph shows the profit percent for Store Num VS Year-

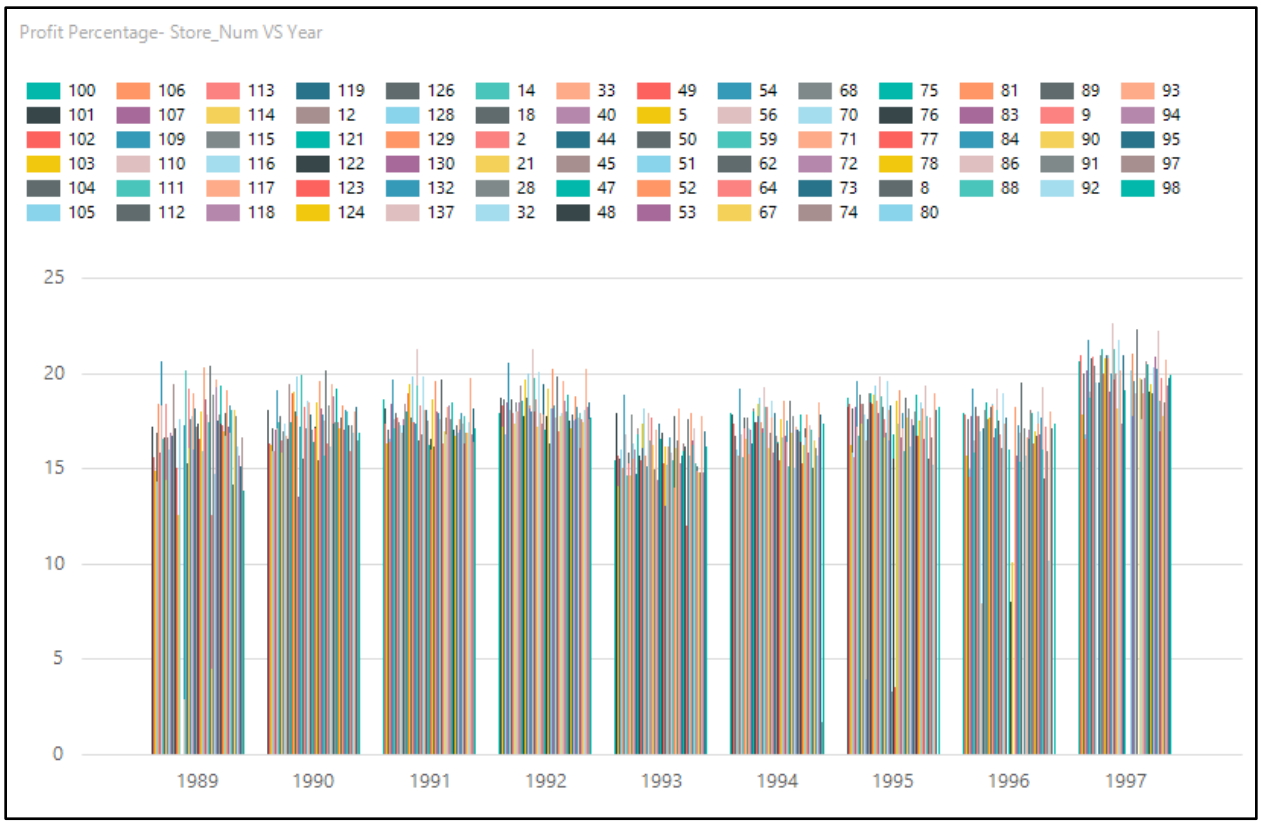


Figure 144: Product profit graph - Store vs Year

Finally, the highest and lowest product profits can also be identified from these graphs with respect to the store and year.

12. Conclusion

The identified business questions for Dominick's fine foods were successfully addressed based on the reporting and analysis services. To address the business questions we typically made use of the Kimball's approach. We made a sales data mart in MSSQL server using SSIS tools which has a sales fact table and 3 dimensions namely store, product and date. This data warehouse was used to generate reports and charts using SSRS and Report Builder 3.0 and cubes using SSAS that enabled to answer the business questions successfully.

12. Team Work

Table 25: Team Work

Task#	Task	Time Taken	Team Member
1	Introduction	30 Minutes	Somya, Amisha, Bhavishya
2	Details of Data and it's Understanding	3 hours	Somya, Amisha, Bhavishya
3	Business Question Identification	2.5 hours	Somya, Amisha, Bhavishya
4	Logical Design	1 hours	Somya, Amisha, Bhavishya
5	Mapping Table	1.5 hours	Somya, Amisha, Bhavishya
6	Schema Justification for Business Questions	40 Minutes	Somya, Amisha, Bhavishya
7	Physical Design Plan	2 hours	Somya, Amisha, Bhavishya
8	ETL Plan & Implmentatoin	15 hours	Somya, Amisha, Bhavishya
9	Target Data	30 minutes	Somya, Amisha, Bhavishya
10	Data Sources	20 minutes	Somya, Amisha, Bhavishya
11	Data Mapping Table	2 hours	Somya, Amisha, Bhavishya
12	Data Extraction Rules	20 minutes	Somya, Amisha, Bhavishya
13	Data Transformation Rules	40 minutes	Somya, Amisha, Bhavishya
14	Data Cleansing Rules	50 minutes	Somya, Amisha, Bhavishya
15	Plan for Aggregate Table	1 hour	Somya, Amisha, Bhavishya
16	Organization of Data Staging Area	45 minutes	Somya, Amisha, Bhavishya
17	Procedure for data extraction & loading	4 hours	Somya, Amisha, Bhavishya
18	ETL for Dimension Tables	2 hours	Somya, Amisha, Bhavishya
19	ETL for fact Tables	1.5 hours	Somya, Amisha, Bhavishya
20	Target Report for Business Question	40 minutes	Somya, Amisha, Bhavishya
21	Mapping to report attribute	2 hours	Somya, Amisha, Bhavishya
22	Report Templates & Implementation	7 hours	Somya, Amisha, Bhavishya