Report-4:
BI Report Design and Implementation for DFF along with the Integrated Group Report

For

Design and Implementation of a Data Warehouse for a Retail Store

In partial fulfillment of *ISTM 637-602 Data Warehousing*

Submitted by:

Group-10

Chirag Karkera (chirag.karkera@tamu.edu)
Tania Manhas (tania.manhas@tamu.edu)
Vanshika Varshney (vanshika@tamu.edu)

Table of Contents

SSMS Login details:

Username: ka0919

Password: Mays0919

Datawarehouse: Group10_602-dw-area

Staging area: Group10_602-staging area

Reporting Server Login details:

Username: ka0919

Password: Mays2020

SSRS Report (BQ 5): 602 Group 10 Report 5 SSRS

SSRS Report (BQ 1): 602 Group 10 Report 1

Analysis Services:

602 Group 10 Report 3

602 Group10 Report 5

1. Introduction

1.1. About Dominick's Finer Foods

Dominick's Finer Foods was a grocery store chain based in Chicago, IL area. It was a subsidiary of large Safeway Group Inc. The chain had its distribution center at Northlake in Illinois and all of its management related activities were carried out at Oak Brook, IL. By 1968, the store grew in size and had 19 stores overall. But almost all the stores of this grocery store chain were closed in Chicago area by December 28, 2013. This was majorly due to the poor sales and an overall weak performance. During the years from 1988 through 1994, DFF partnered with Chicago Booth, University of Chicago. During this partnership, various experiments were carried out for the purpose of research into shelf management and pricing strategies. Experiments were conducted in more than 25 categories across all of its stores. Resulting from these experiments, we now have a sales data of almost 9 years (1988-1994). The size of this dataset is almost 5 GBs and contains information regarding customer sales, demographic data for its various stores and information regarding the items sold by different stores. This dataset can be very useful in analyzing and answering a number of business questions which will provide understanding about the sales and the various stores. The main idea is to build and develop a data warehouse so as to be able to answer a number of business questions through the project and also to gain strategic and business-oriented insights into the operations and functioning of Dominick Finer Foods.

In order to carry out initial analysis and to list out various business questions, our team used a number of tools including Microsoft Excel (charts, pivot charts) and JMP(SAS) and worked on the data. Our aim was to strategically analyze the data given the tools at hand and to make use of data creatively in order to be logically clear regarding the direction in which the team is moving forward. The subsequent sections provide a list of questions that the team has decided to work on along with the explanations. The sections also provide relevant analyses of the data regarding the respective business questions which aids in explaining the theory behind choosing those questions.

1.2. Challenges

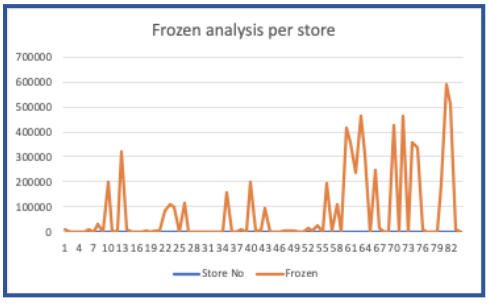
Although the objective of the project is clear, there are numerous challenges that our team has faced and addressed while working on the data and drawing out desired outcomes. The problems vary in their nature – ranging from the sources of data to the types and quality of data. All of these problems and challenges have been listed below:

- The majority of data is contained in CSV files, but there was some data which was contained in the form of pdf files and even HTML data. For an instance, event calendar and monthly calendar days were required to be extracted from pdf files.
- Being a team having no prior experience working in the field of retail and sales, analyzing the data to come up with relevant business questions appeared to be a gigantic task. In order to cope with this problem, the team comprehensively went through all the provided materials to understand the field and related data and ultimately come up with the desired outcome.
- The data is not clean and is extremely dirty in the sense that there are numerous special characters and even blank spaces. This made the task of analyzing and extracting data time consuming and difficult.
- The data was widespread, and the team had to come up with an ERD in order to understand and explain the data and to have a better picture of the data. Due to this widespread nature of the data, the team faced a few difficulties and had to spend substantial time so as to understand the model and come up with a common idea of the data.
- The team faced challenge in keeping everyone on the same page due to the problem of data being dirty. Extra care had to be taken to use the cleaned version of the data to carry out analysis for business questions in order to avoid rework and to maintain a consistent version of data.

2. Details About the Data

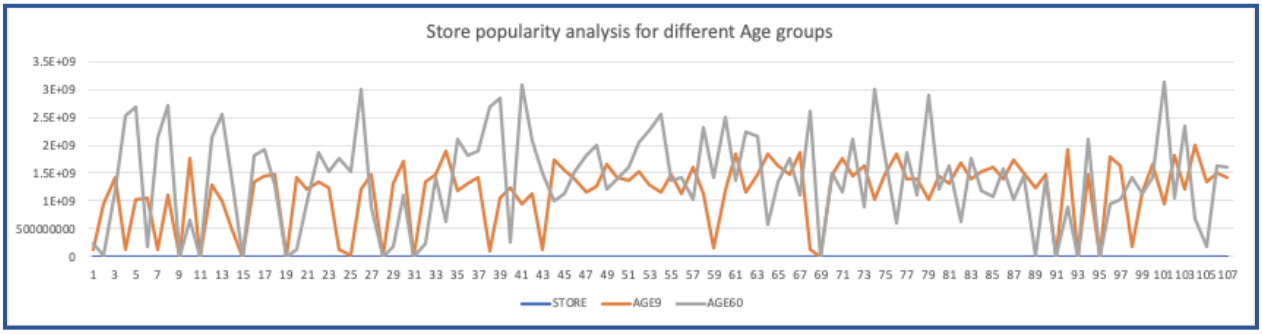
This dataset covers data at the store-level for DFF's stores for a period of more than seven years. There are majorly two types of files in the data – category-specific files and general files. Category-specific files contain data for about 29 categories and general files have demographic and customer count data. We have discussed in detail about each of the files in the sections below. We have also discussed about some of the files by taking samples from the data and have also shown trends for some of them below:

We have analyzed data of CCOUNT file for frozen products as shown-



The trends cannot be clearly analyzed from these charts as it has lots of garbage values and null values, so we have understood that we need to clean the data before loading it into the data warehouse.

We also analyzed the demographics file as shown-



From this chart, we understood that we can compare store popularity amongst different age groups. Hence, we can use this dataset to compare different income groups, ethnic groups, educational levels etc.

We analyzed data for a Movement file as shown-

STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK
62	1242409401	384	0	1	0		0	1
62	1242409401	385	3	1	4.79		27.72	1
62	1242409401	386	3	1	4.79		27.72	1
62	1242409401	387	5	1	4.79		27.72	1
62	1242409401	388	3	1	4.79		27.72	1
62	1242409401	389	3	1	4.79		27.72	1
62	1242409401	390	5	1	4.79		27.72	1
62	1242409401	391	4	1	4.79		27.72	1
62	1242409401	392	2	1	4.79		27.72	1
62	1242409401	393	2	1	4.79		27.72	1
62	1242409401	394	1	1	4.79		27.72	1
62	1242409401	395	1	1	4.79		27.72	1
62	1242409401	396	0	1	0		0	1
62	1242409401	397	2	1	4.79		45.72	1
62	1242409401	398	0	1	0		0	1
62	1242409401	399	0	1	0		0	1
62	1242409402	384	0	1	0		0	1
62	1242409402	385	8	1	4.79		27.72	1
62	1242409402	386	5	1	4.79		27.72	1
62	1242409402	387	8	1	4.79		27.72	1
62	1242409402	388	5	1	4.79		27.72	1
62	1242409402	389	5	1	4.79		27.72	1
62	1242409402	390	6	1	4.79		27.72	1
62	1242409402	391	6	1	4.79		27.72	1
62	1242409402	392	3	1	4.79		27.72	1
62	1242409402	393	1	1	4.79		27.72	1
62	1242409402	394	3	1	4.79		27.72	1
62	1242409402	395	3	1	4.79		27.72	1
62	1242409402	396	3	1	4.79		45.72	1
62	1242409402	397	4	1	4.79		45.72	1
62	1242409402	398	0	1	0		0	1
62	1242409402	399	9	1	4.79		45.72	1
62	1242409403	384	0	1	0		0	1
62	1242409403	385	1	1	4.79		27.72	1
62	1242409403	386	4	1	4.79		27.72	1
62	1242409403	387	1	1	4.79		27.72	1

In this data we have been provided information regarding UPC and their associated move, qty, price and profit values. In the data manual it has been mentioned that we can calculate sales values using formula: Sales = Price * Move / Qty. Hence, we can use this data to evaluate sale amount for a specific UPC at store and week level. Since, sales analysis is very crucial for any business,

we can use it to evaluate sales and compare it at store or quarter level based on the business requirements.

2.1. Understanding the Data

a. Customer Count File:

- This file contains data about weekly sales data about various product categories such as beer, camera, cheese, fish, dairy, etc.
- The file also contains the data about coupons that have been redeemed while purchase of all these categories.
- This file helps us to understand the weekly sales patterns of the products sold by DFF across their multiple stores. We can also understand the usage of coupons by the customers during their purchases which can help us to understand the impact of promotional schemes on the sales of the product.

b. Store Level Demographics

- This file contains the demographic data of Chicago Metropolitan Area obtained from US government census data.
- The file gives the store-wise weekly distribution of customers within different age groups.
- The file also contains information about household income, number of dependent members, employment status, and retired status of all customers.
- This data will help us in coming up with various targeting strategies for customers based on the demographic information given about them.

c. UPC Files

- UPC (Unique Product Code) files contain data about products and each product category is mapped to a separate file.
- Each UPC file contains data about products in a specific category and these files can be used to make product specific strategies.

d. Movement Files

- These files contain weekly movement of each category of product in each store.
- These files also give the information about profits and price of the products which will help us to understand the distribution of profit margins for various of product categories.

- We can also understand which products are incurring losses and develop strategies for them.

2.2. Metadata for all OLTP Files

a. CCOUNT file

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

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

SALADBAR	Salad Bar Sales in Dollars	Numeric	8
SALCOUP	Salad Coupons Redeemed	Numeric	8
SPIRITS	Spirits Sales in Dollars	Numeric	8
SSDELICP	Self Service Deli Sales in Dollars	Numeric	8
VIDCOUP	Video Coupons Redeemed	Numeric	8
VIDEO	Video Sales in Dollars	Numeric	8
VIDEOREN	Video Rentals (Dollar Amounts)	Numeric	8
WINE	Wine Sales in Dollars	Numeric	8

b. Demography File

Variable Name	Description	Type	Length
age9	% Population under age 9	Numeric	8
age60	% Population over age 60	Numeric	8
ethnic	% Blacks & Hispanics	Numeric	8
educ	% College Graduates	Numeric	8
nocar	% With No Vehicles	Numeric	8
income	Log of Median Income	Numeric	8
incsigma	Std dev of Income Distribution (Approximated)	Numeric	8
hsizeavg	Average Household Size	Numeric	8
hsize1	% of households with 1 person	Numeric	8
hsize2	% of households with 2 persons	Numeric	8
hsize34	% of households with 3 or 4 persons	Numeric	8
hsize567	% of households with 5 or more persons	Numeric	8
hh3plus	% of households with 3 or more persons	Numeric	8
hh4plus	% of households with 4 or more persons	Numeric	8
hhsingle	% of households with 1 person	Numeric	8
hhlarge	% of households with 5 or more persons	Numeric	8
workwom	% Working Women with full-time jobs	Numeric	8
sinhouse	% Detached Houses	Numeric	8

density	Trading Area in Sq Miles per Capita	Numeric	8
hval150	% of Households with Value over $150,000	Numeric	8
hval200	% of Households with Value over $200,000	Numeric	8
hvalmean	Mean Household Value (Approximated)	Numeric	8
single	% of Singles	Numeric	8
retired	% of Retired	Numeric	8
unemp	% of Unemployed	Numeric	8
wrkch5	% of working women with children under 5	Numeric	8
wrkch17	% of working women with children 6 - 17	Numeric	8
nwrkch5	% of non-working women with children under 5	Numeric	8
nwrkch17	% of non-working women with children 6 - 17	Numeric	8
wrkch	% of working women with children	Numeric	8
nwrkch	% of non-working women with children	Numeric	8
wrkwch	% of working women with children under 5	Numeric	8
wrkwnch	% of working women with no children	Numeric	8
telephn	% of households with telephones	Numeric	8
mortgage	% of households with mortgages	Numeric	8
nwhite	% of population that is non-white	Numeric	8
poverty	% of population with income under $15,000	Numeric	8
shopcons	% of Constrained Shoppers	Numeric	8
shophurr	% of Hurried Shoppers	Numeric	8
shopavid	% of Avid Shoppers	Numeric	8
shopstr	% of Shopping Stranges	Numeric	8

shopunft	% of Unfettered Shoppers	Numeric	8
shopbird	% of Shopper Birds	Numeric	8
shopindx	Ability to Shop (Car and Single-Family House)	Numeric	8
shpindx	Ability to Shop (Car and Single-Family House)	Numeric	8

c. UPC Files

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

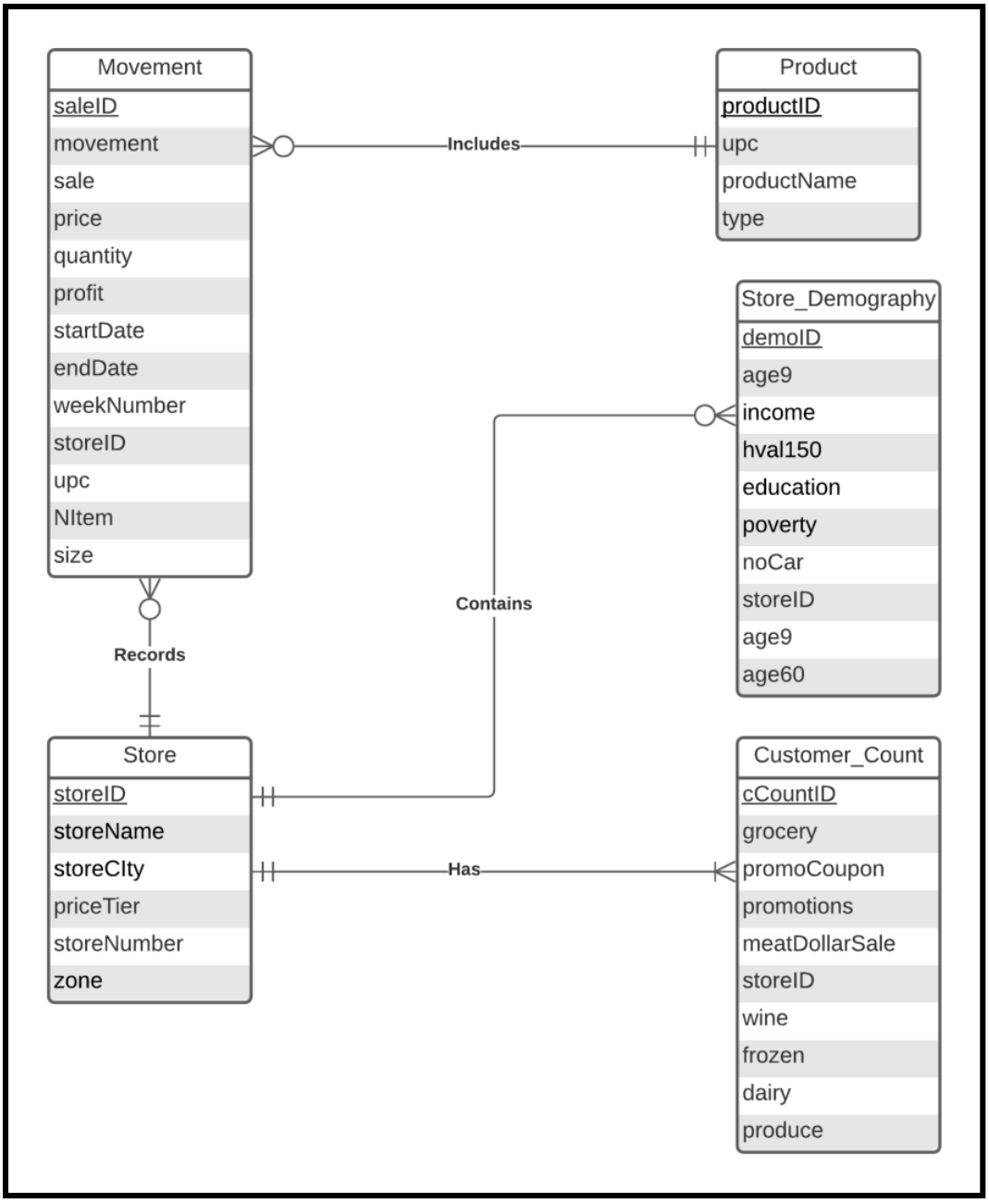
d. Movement Files

Variable	Description	Type	Length
upc	UPC Number	Numeric	8
store	Store Number	numeric	3
week	Week Number	Numeric	3
move	Number of units sold	Numeric	8
price	Retail Price	Numeric	8
qty	Number of items bundled together	Numeric	3
profit	Gross margin	Numeric	8

sale	Sale code (B, C, S)	Character	8
ok	1 for valid data, 0 for trash	Numeric	3

2.3. Entity-Relationship Diagram

Below diagram displays the relationships between various entities in the dataset. The underlined fields denote Primary Keys for the tables.



3. Domain Understanding

Retail industry is an extremely competitive market with unpredictable customer behavior and slim margins. Thus, with the desire to understand customers' needs and their purchase drivers so as to gain competitive advantage, companies in retail industry began to adopt data warehousing technology. Infact, it was one of the early adopters of the data warehousing technologies in early 1990's. The implementation of data warehouse in retail industry has offered it numerous benefits which includes cross-organizational analysis, in-depth analysis of business trends, deep understanding of customer behavior, customer preferences, product pricing, customized promotional campaigns, etc. Now let's deep dive into the retail industry to analyze the various trends which impacts its sales.

Li H. et al., [1] discusses about the major benefit that data warehousing provides to retail industry is the ability to conduct budgetary analysis. As a result, organizations use reports and dashboards to analyze the expected and actual budget expenditure. The insights derived from these reports help them in analyzing their current expenditure in promotional, pricing campaigns and based on the results obtained from different campaigns they can update their budgets. Since, Dominick Fine Foods has used coupons as part of their promotional campaigns in different stores, the implementation of data warehouse could provide them the ability to analyze the impact of the promotional campaign, conduct budgetary analysis, analyze cost overruns and based on the performance they could revisit the budget allocated to different stores.

In retail industry, forecasting plays a critical role as it helps the retailers in managing their inventory in an efficient manner. Huang et al., [2] depicts how gaining competitive information (prices and promotions) can be used to increase forecasting accuracy at UPC level. In this paper authors have discussed conducting forecasting in two stages. The first stages involve gathering competitive information and in second stage Autoregressive Distributed Lag (ADL) model has been used. They have further investigated the forecasting accuracy for promoted and non-promoted period to support their findings. The study has been supplemented by conducting research on Dominick Fine Foods as it has sales data at UPC level with three different types of promotions which had been employed i.e. Simple price reduction, Bonus buy, and Coupon.

Furthermore, Nevo et al., [3] discusses that during the periods when coupons are issued shelf lives of the product is reduced. Coupons are not only used to attract customers to purchase new products but also induce sales of products which face huge competition. They have also provided evidence

regarding how coupons help in inducing repurchase of the products amongst customers. In case of Dominick Fine Foods, the same approach could be applied as the coupon's had varied impact on sale for different products. Hence, this analysis could be used to plan the promotional strategy with coupons.

In retail industry, consumers' behavior gets very easily impacted by the changes in product attributes. Jami et al., [4] discusses that when value of any attribute is changed while keeping other attributes constant, customers' behaviors and judgements are impacted. The paper provides empirical evidence to show how consumers preference would change towards the product where change was desirable for them, eventually leading to increase in sales. This research has used Dominick data to test the hypothesis. Since, in case of Dominick's the undesirable price increase led to drop in sales and desirable price reduction led to increase in sales. Thus, the impact of change of any business attribute on customer behavior can be used to form marketing strategies.

In retail industry, seasonal demand has a lot of impact on the price of the products in the competitive business environment. Chevalier et al., [5] provides empirical evidence regarding how changes in retail margins has led to reduction in prices during seasonal demand peaks for a product. They have conducted study to analyze the price and demand relationships and determined how the results are based on "loss-leader" models of retailer competition. Prices are mainly reduced by the retailer to attract customers who end up buying more once they visit the shops. In case of Dominick Fine Foods, the same approach could be applied as there are multiple stores facing stiff competition from different retailers. Hence, by strategically reducing prices sales could be improved.

4. Business Questions and their Substantiations and Explanations

Following are the business questions that we have come up with by understanding and analyzing the data. The business questions have been categorized in the following categories:

1) High: Takes into account the questions which have humongous impact on the business

2) Medium: Takes into account the questions which have moderate impact on the business but the outcome is not as high as the factors in the "High" category.

3) Low: Takes into account the factors which are of less importance to the business.

The justification for this has been provided after each question. To sum up, it is important for a retail industry to maximize their revenue by understanding various trends and factors – sales of products during a particular occasion, variation in products sales due to difference in customer age groups, variation in customer demographics (age, salary, etc.), effects of marketing and

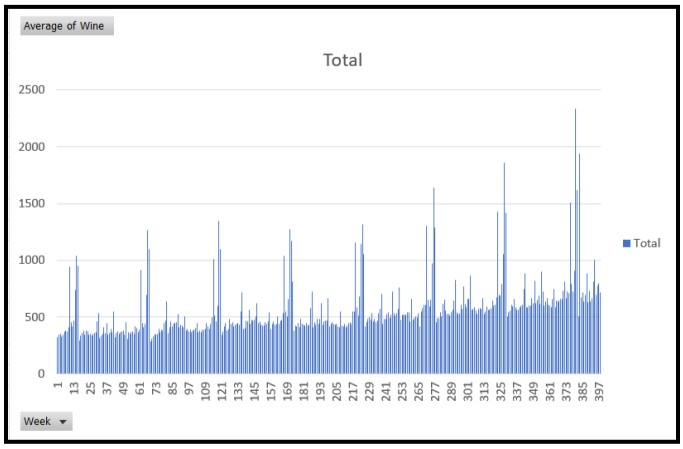
promotional strategies, location of stores, etc. We have focused on these factors and prioritized the questions accordingly.

Q1) Determine the trend of how the sales of Wine is varying during the time of Christmas for the entire period? (High)

Row Labels	Average of Wine
1	321.4626242
2	341.6825155
3	351.722764
4	328.0494565
5	341.677205
6	374.112
7	384.4183564
8	376.7986482
9	409.7342089
10	945.4827189
11	457.3984025
12	414.9251306
13	466.9039969
14	742.4641033
15	1037.20352
16	953.5611498
17	293.5134887
18	335.6266917
19	360.0199699
20	381.0853083
21	342.4440752
22	379.2401504

Grand Total	563.1149138

Average Wine Sales Overall



Average sales of wine across all the weeks

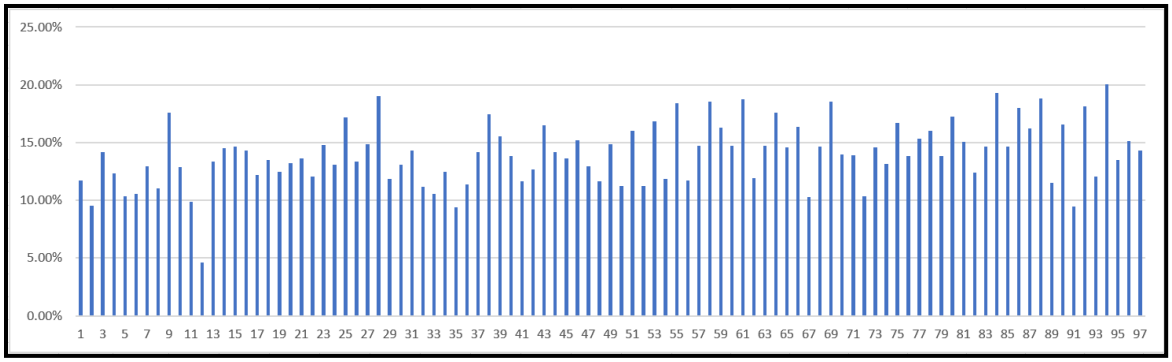
Justification

Along with festive seasons, the sales of certain products also vary a lot. Here, we have analyzed the trend of Wine sales over the season of Christmas. As we can evidently see from the graph, there are sudden hikes in the average sales throughout the entire period. When we analyzed the numbers of the weeks on which hike was occurring, we found that those were the weeks of/around Christmas. We came to know about the information of week number on which Christmas occurs through data information given on the website. By analyzing sales of such products which have

sudden growth during festive occasions, DFF can use that period to their advantage and manage the supplies of the products accordingly. This is one of the aspects of efficient shelf management and would allow DFF to effectively organize their shelf space and make use of increased sales opportunities. All in all, this kind of analysis can prove to be very beneficial for DFF.

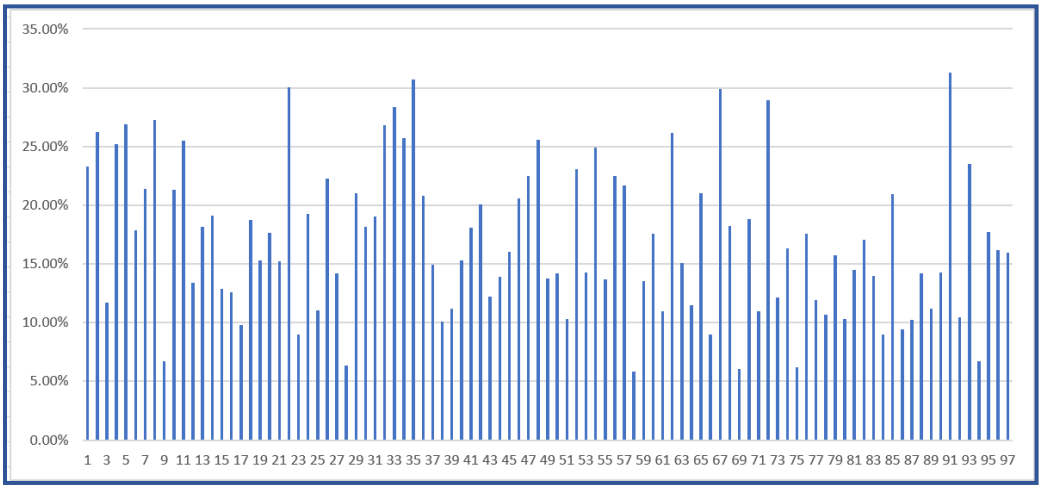
Q2) Which are the stores that have higher popularity amongst kids and amongst elderly groups pf people? (Medium) ***(Implemented)***

STORE	AGE9
2	11.75%
4	9.51%
5	14.14%
8	12.32%
9	10.35%
12	10.57%
14	12.96%
18	11.01%
21	17.59%
28	12.89%
32	9.91%
33	4.61%
40	13.37%
44	14.49%
45	14.67%
47	14.30%
48	12.18%
49	13.49%
50	12.44%



Store-wise percentage visit of kids(age<9)

STORE	AGE60
2	23.29%
4	26.20%
5	11.74%
8	25.24%
9	26.91%
12	17.83%
14	21.39%
18	27.23%
21	6.69%
28	21.33%
32	25.50%
33	13.42%
40	18.19%
44	19.10%
45	12.89%
47	12.58%
48	9.79%
49	18.75%
50	15.34%



Store-wise percentage visit of Elderly(age>60)

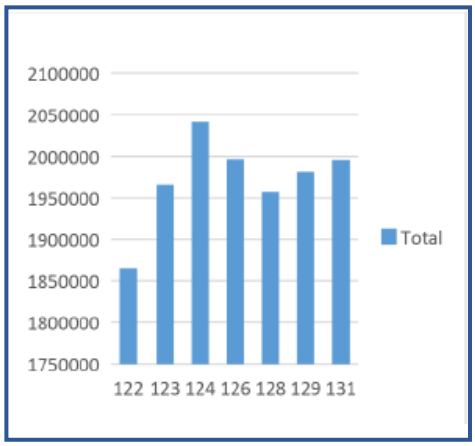
Justification

The graphs displayed above gives us information regarding the stores which are visited the most or the least by kids and by elders. By analyzing such information, Retail companies can strategize their marketing tactics and thus can be utilized by their sales and marketing department. For example, DFF can use this information to keep supplies of products which the kids like in the stores which are visited by kids the most. In a similar way, stores that are more popular amongst the elderly can have more supplies of products which are famous among elderly people. This will be beneficial for them and will aid in increasing their sales. Also, this question come under Medium priority as we are taking into account only kids and elders and not a narrow division of customers on the basis of their age.

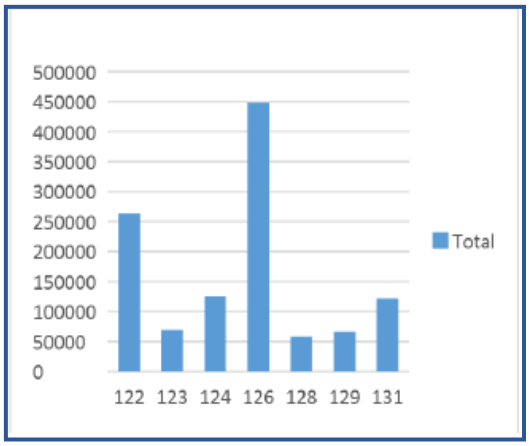
Q3) What is the effect on customer count (or frequency of customers) by the number of coupons? (High)

Row Labels	Sum of CUSTCOUN	Sum of Sum of coupons redeemed
122	1864750	264391.96
123	1965707	69975.38
124	2041411	125162.55
126	1996714	448490.56
128	1957184	57894.85
129	1980994	66385.66
131	1996128	121204.53
Grand Total	13802888	1153505.49

Count of Customers and Coupons Redeemed per week



Count of Customers per week



Count of Coupons Redeemed per week

Justification

In the above graphs, we are analyzing the effect of coupons on the frequency of visits of the customer. Here, we are considering the count of coupons that are being redeemed is proportional to the number of coupons being created. This kind of analysis can be used so as to analyze the impact of marketing/promotional strategies on the frequency of visits of the customer. DFF can also make use of this information to compare between various strategies for promotion and come up with the one which is the most effective amongst all. All in all, this kind of analysis will prove to be very beneficial for the company and will help in boosting their revenue. This business question is of High importance as coupons form a part of promotional strategies and informs the company about their marketing decisions.

Q4) Which stores have profit below the average profit margin of Cereal (WCER)? (High)

(Implemented)

Average Profit
10.93381561

Overall Average Profit

Store no	Average Profit
2	13.1010827
5	10.01901856
8	11.87108841
9	11.21213245
12	13.37669353
14	12.96915902
18	10.03424951
21	9.780566657
28	10.79777501
32	13.03314656
33	11.29250807
40	9.187376785
44	11.12345606
45	8.909152716
47	10.52265988
48	9.547635523
49	9.876613898
50	8.873816456
51	10.49016176
52	12.36324011
53	13.49437943
54	10.96060323
56	10.66600397
59	9.004972121
62	13.43324998
64	8.571727996
67	10.55053032
68	12.55991864
70	8.959704398
71	12.26592575
72	12.60496081
73	10.12717887
74	11.006259
75	13.10601031
76	10.708809

Average Profit across stores for Cereal

Justification

The movement data provides the ability to analyze average profit across different stores for different product categories. This data can be used by the company to analyze performance of different stores and eventually modify the current strategies to improve profitability of different stores. In this case, we are analyzing the average profit earned across different stores for cereals. This data will help us in understanding the top performers and the poor performers. It can further be used to analyze reasons behind performance and the marketing strategies can be updated to improve profitability of poor performing stores. This business question is of High importance as it provides insights into profitability of a product across stores.

Q5) What is the trend for the demand of grocery in different price tiers? (High)

Store details	Average Grocery Sales
0	19018.66667
1	24325.63
2	16991.81348
4	12883.49004
5	27110.76209
8	31370.3167
9	18971.198
10	15692.08559
12	22311.21109
14	20995.15031
16	15016.32068
18	28823.48707
19	10460.63832
21	20897.74365
25	9938.9989
28	14985.20773
32	30102.20702
33	17318.6352
39	10969.27785
40	23398.17992
44	24734.62243
45	15661.04519
46	19234.47009
47	17049.72335
48	14213.48109
49	13377.67024
50	14018.92465
51	19404.27551
52	23726.09478
53	17614.60341
54	16550.13213
55	12914.7759
56	18467.48803

Average Grocery Sales for different stores

Price Tier	Total
Low	48
Medium	66
High	12

Price Tier

Demand of grocery sales across different price tiers

Justification

In this case, we are analyzing the trend of demand of grocery sales in different price tiers. There are three price tiers i.e. high, low, medium. The price tiers are created by the organization based on their requirements. In this case, I have assumed the range for each category as follows:

Range	Price Tier
0-20000	Low
20000-4000	Medium
4000-6000	High

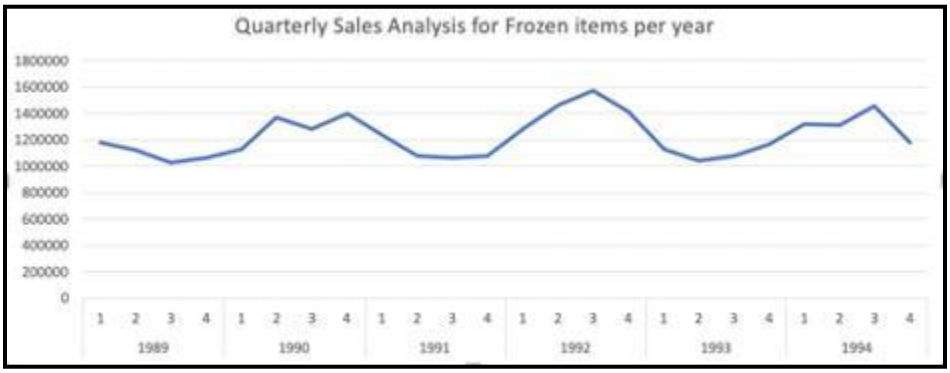
Price ranges for Price Tier

We have then created a plot to analyze stores in each of these price tiers. From this analysis we can see that maximum stores grocery sales lie in medium price tier. This analysis can be used to manage inventory and stock of the products which are being brought in different stores. It also The organization can further look into the trend to analyze which price tier is least popular (in this case its high), they can work to modify the marketing strategies to boost the sales in that price range. This business question is of High importance as it distinguishes the various price tiers in which grocery items will fall and helps the company analyze demand for day-to-day products.

Q6) How have the sales of frozen products varied over quarters in a particular year? (Medium) ***(Implemented)***

Quarter	Total Frozen Sale
1989	4399850.28
1	1181021.57
2	1123985.87
3	1030140.42
4	1064702.42
1990	5178531.85
1	1130595.24
2	1369334.57
3	1280191.14
4	1398410.9
1991	4452518.93
1	1237713.78
2	1076575.04
3	1062130.64
4	1076099.47
1992	5729941.34
1	1280897.9
2	1466638.97
3	1572689.7
4	1409714.77
1993	4412482.38
1	1127029.64
2	1042971.39
3	1077804.46
4	1164676.89
1994	5262247.9
1	1320195.19
2	1307456.17
3	1457336.56
4	1177259.98
Grand Total	29435572.68

Total Sales of Frozen Products per quarter per year



Total Sales of Frozen Products per quarter per year

Justification

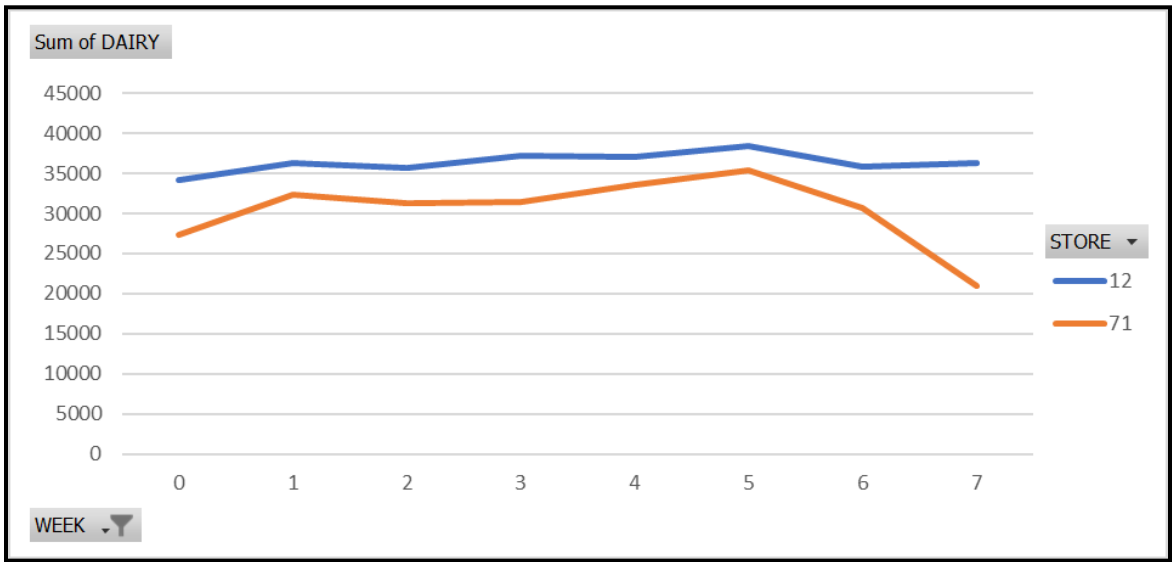
In the graph above, quarterly analysis of sales of frozen products have been conducted. This helps in analyzing how were the sales across the different years in different quarters. This can help management in analyzing if there are certain factors which led to growth/decline in sales of the

frozen product in different quarters over the years. Further, the analysis of the sales trends can help in forecasting demand in different time periods as well. Hence, quarter analysis helps in finding underlying trends in the sales and reasons associated with that. This question of Medium importance as the focus is on a niche product category.

Q7) How is the demand for a particular product (Dairy) changing in different price-tiers? (Medium) ***(Implemented)***

Sum of DAIRY	Column Labels		
Row Labels	12	71	Grand Total
0	34126.73	27360.65	61487.38
1	36233.3	32413.04	68646.34
2	35746.24	31286.16	67032.4
3	37155.84	31456.08	68611.92
4	37133.18	33594.3	70727.48
5	38482.39	35440.59	73922.98
6	35913.24	30654.99	66568.23
7	36371.86	20971.25	57343.11
Grand Total	291162.78	243177.06	534339.84

Weekly sales of Dairy in store 71(medium-price tier) store 12(high-price tier)



Justification

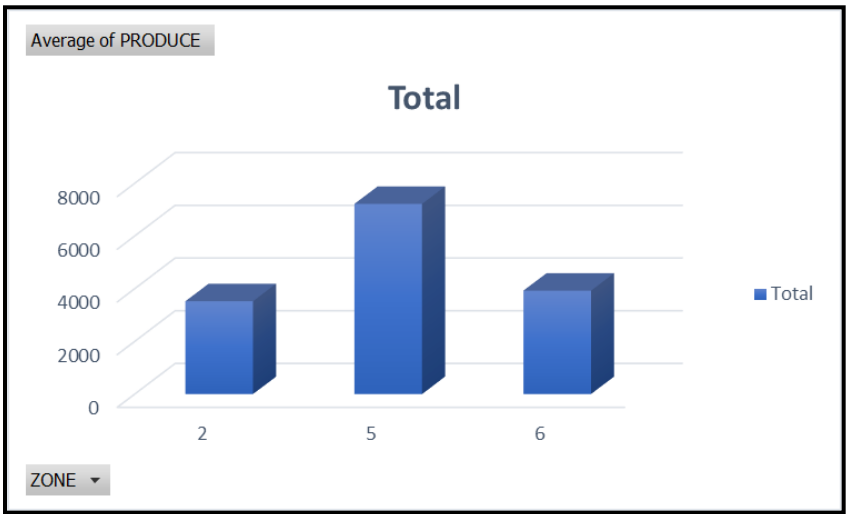
The above plot shows the sales of dairy products in in stores from two different price tiers i.e. high price tier (store 12) and medium-price tier (store 71). We can see a clear trend that people in high price tier zones buy higher amounts of dairy products than people in medium price tier zone. One reason for this may be that buying power of people in high price tier zones is higher and hence they can afford to buy more protein rich foods like dairy. Similarly, we can analyze other zones as well to develop a trend of products and their movement in the stores. This will help DFF to manage

their inventory as well as target customers accordingly. This question of Medium importance as the focus is on a niche product category to analyze the trends.

Q8. What is the trend of average sales of a particular product in different zones in (Produce)? (High)

Row Labels	Average of PRODUCE
2	3524.699804
5	7216.530204
6	3925.39365
Grand Total	**4003.957419**

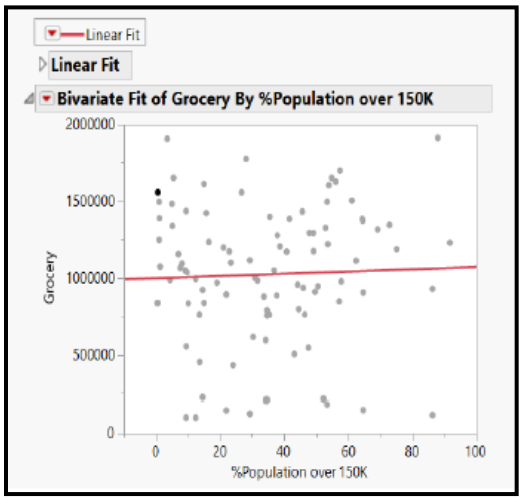
Average sales of produce in zones 2, 5 and 6

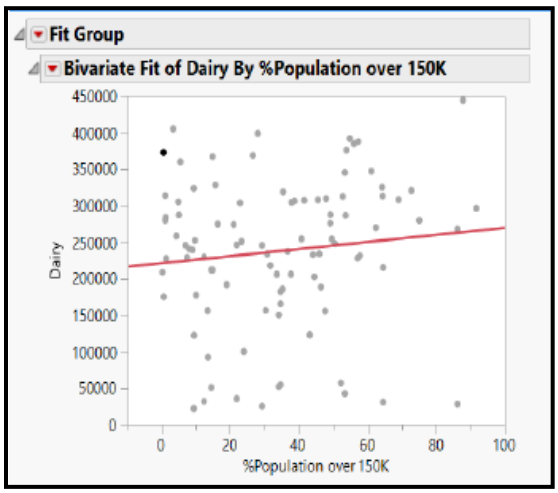


Justification

Above graph shows the average sales of produce stores in different zones. This question will help us to understand the trend of sales of a particular product on the basis of zones. This will help DFF to analyze which zones are performing well as well as those which are underperforming and if required make changes for the stores in underperforming zones. This analysis will also help DFF to improve their supply chain and warehouse management. This question of High importance as it provides the business insights into various zones.

Q9) How are the sales of various items varying for stores with difference in salaries (over 150K)? (Low)





Justification

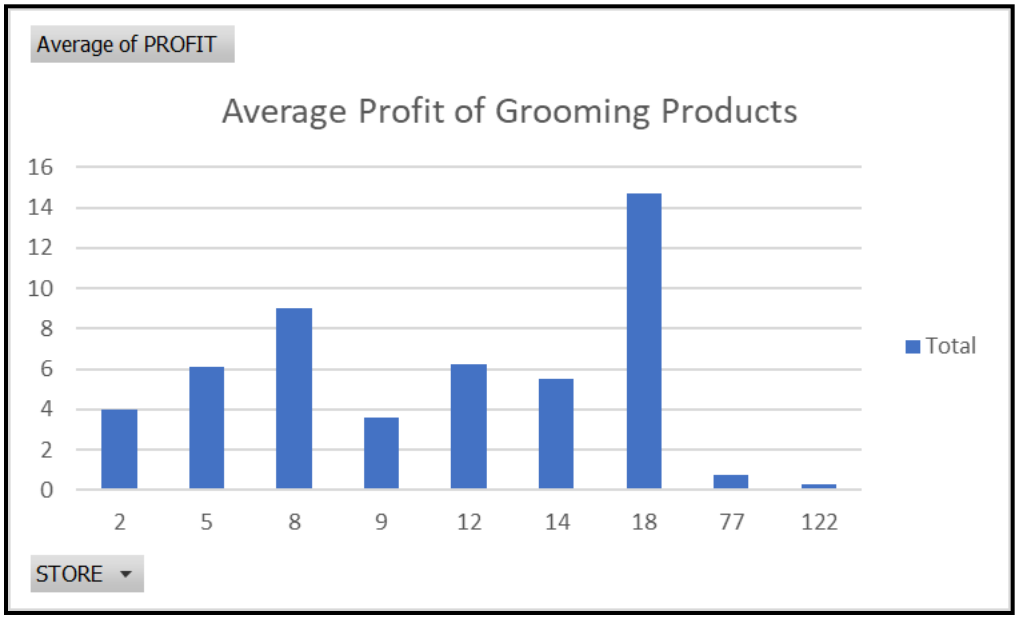
It is of immense importance for the retail companies to make the pricing related decisions suitably. They need to keep into account the various demographics of the location in which they are selling products. One such demographics is the demographic of salary which plays a huge role in buying patterns of the customers. By comparing the sales of products with salaries, we can define how the salaries are impacting sales in a particular location of the store and can be used to analyze whether the store has suitable price ranges to match the salaries of customers in that area. Wrong pricing decisions can highly impact the sales of products and can be held accountable for considerable number of losses.

Here, we have made use of the data from CCOUNT file and the Demographics file and combined both the tables. We have analyzed the data for Dairy and Grocery products and similar analyses can be carried out for other products as well. As we can see from the graphs above, the sales of items are increasing with the proportion of people having salaries amounting to or greater than 150K. Thus, we can imply that the higher the average salary, the higher are the sales of products. DFF can make use of this data to decide on pricing strategies for luxury items (products which are costlier than other products). These pricing decisions highly impact the sales and in turn affect the revenue of the company. This question is of low importance as we are analyzing this for luxury items whose sales may not be that common as compared the necessities.

Q.10 What is the store-wise average profit of particular product category (Grooming products)? (Medium) ***(Implemented)***

Row Labels	Average of PROFIT
2	4.012042254
5	6.129084507
8	9.028732394
9	3.598098592
12	6.224647887
14	5.541126761
18	14.67545455
77	0.729204545
122	0.287613636
Grand Total	**5.062980952**

Average store-wise profit on grooming products



Justification

All businesses aim to maximize their profits and thus understanding products and stores which are profitable will help the higher management pick out the products in stores which are not performing well. The management can then take decision whether to plan a strategy to improve performance of these products or to discontinue the products. This question will also help DFF cut their losses. This will ultimately lead to more revenue for the entire retail chain. This question of Medium importance as the focus is on a niche product category.

5. Independent Data Marts Using Kimball's Approach

5.1. Summary of Proposed Schema

In the next step we will be designing the data marts for our data warehouse that will be used to answer the business questions using Kimball's approach. For our design we have decided to use Star Schema for our data marts with one fact table and multiple dimension tables. We have proposed two data marts namely SALES Data Mart and STORE_PERFORMANCE Data Mart. These data marts consist of SALES_FACT and STORE_PERFORMANCE_FACT fact tables respectively. Our dimension tables will be PROD_DIM, STORE_DIM and TIME_DIM for these data marts. We will be discussing the structure of these fact and dimension tables in detail later in the section.

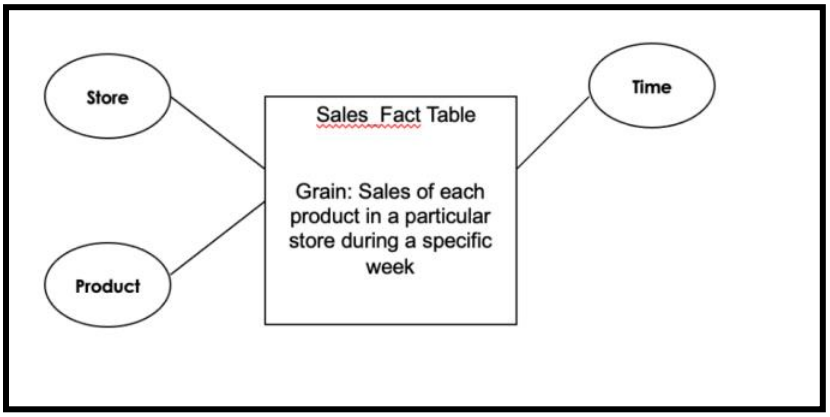
5.2. Dimensional Matrix for Data Marts

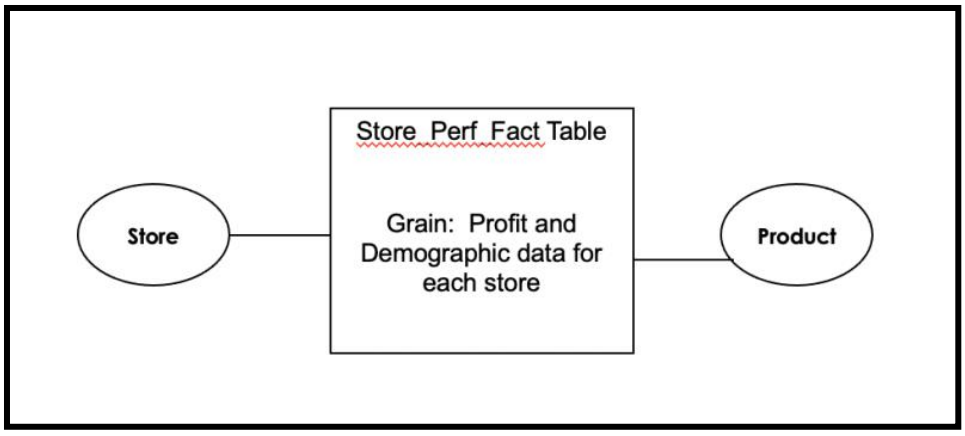
Dimension / Data Mart	PROD_DIM	STORE_DIM	TIME_DIM
STORE_PERFORMANCE	x	x	
SALES	x	x	x

In the above diagram, we can see the dimension tables that are being used in specific data marts of our dimensional model. Now let us discuss the structure of these fact and dimension tables in order design as well as their mappings with each other.

5.3. Dimensional Model

FACT TABLE DIAGRAM





SALES_FACT TABLE

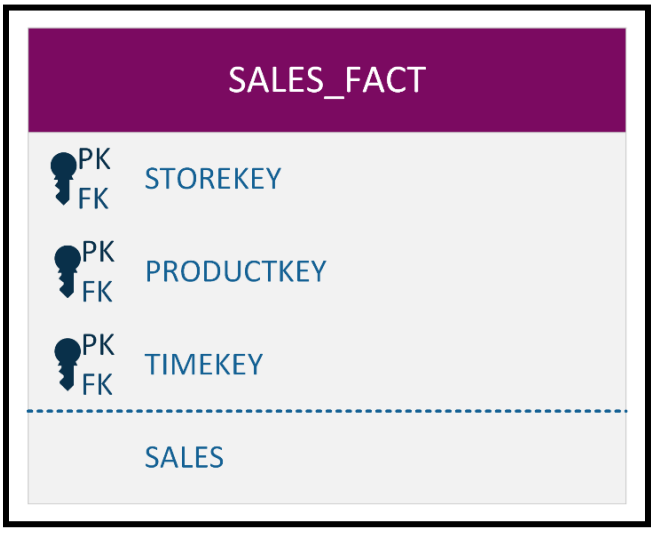
Let's have a look at the SALES_FACT table in the SALES data mart. This table contains information about the sales across all the Dominick's stores for different products in a given time frame. This table has been primarily derived from Movement Files. The movement files contain data for movement, price and quantity of a particular product in a particular week. We have derived the sales data by using the formula sales = price * movement / quantity. The SALES_FACT table consists of the following columns:

STOREKEY: This is the primary key of STORE_DIM table which is a foreign key in the SALES_FACT table. This is used to map stores with their sales data.

PRODUCTKEY: This is the primary key of PRODUCT_DIM table which is a foreign key in the SALES_FACT table. This is used to map the products with their sales data.

TIMEKEY: This is the primary key of TIME_DIM table which is a foreign key in the SALES_FACT table. This is used to map week numbers with sales of a particular store in a specific store.

SALES: As discussed earlier, this column is a derived column from price, movement and quantity of the movement files. This is the actual measure of this fact table that will help us to answer the related business questions.



Sales Fact Table

STORE_PERFORMANCE_FACT TABLE

This table contains information about the metrics that will be used to evaluate the performance of a store. Let's take a look at the structure of this table. This table is derived from movement, UPC and demographics files.

STOREKEY: This is the primary key of STORE_DIM table which is a foreign key in the STORE_PERFORMANCE_FACT table. This is used to map stores with their store performance data.

PRODUCTKEY: This is the primary key of PRODUCT_DIM table which is a foreign key in the STORE_PERFORMANCE_FACT table. This is used to map the products with their store performance data.

PROFIT: This column denotes the profit value in dollars for a particular product in a store. This column is derived from the gross_profit column in the movement files which gives percentage profit for each week.

BELOW_9%: This column indicates the percentage of population visiting a store that is below 9 years of age i.e. children visiting the store.

ABOVE_60%: This column indicates the percentage of population visiting a store that is above 60 years of age i.e. senior citizens visiting the store.

Store Performance Fact Table

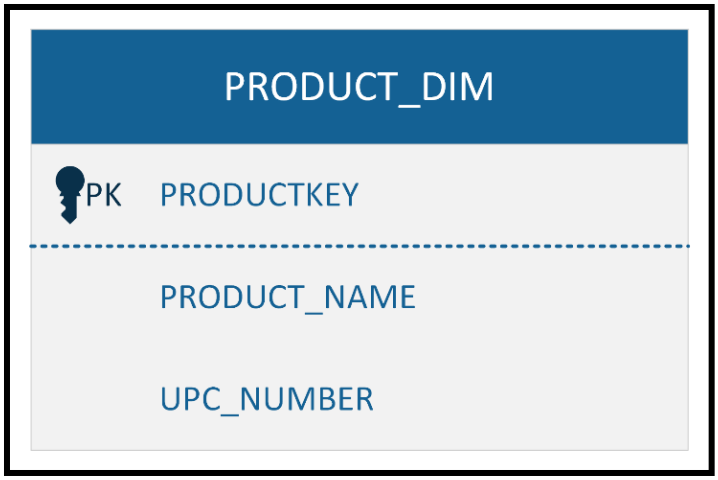
PRODUCT_DIM TABLE

This table consists data about the various products sold by DFF across all their stores. The data for this table is derived from the UPC files.

PRODUCTKEY: This is the primary key of PRODUCT_DIM table which is a foreign key in the SALES_FACT and STORE_PERFORMANCE_FACT table. This is used to map the products with their sales and store performance data. This field will be system generated.

UPC_NUMBER: This is a number assigned by DFF to identify each product uniquely.

PRODUCT_NAME: This column indicates the name of the product.



Product Dimension Table

STORE_DIM TABLE

This table consists information about individual stores. This information is derived from the demographics file and store table given in the data manual.

STOREKEY: This is the primary key of STORE_DIM table which is a foreign key in the STORE_PERFORMANCE_FACT and SALES_FACT table. This is used to map stores with their store performance and sales data. This field will be system generated.

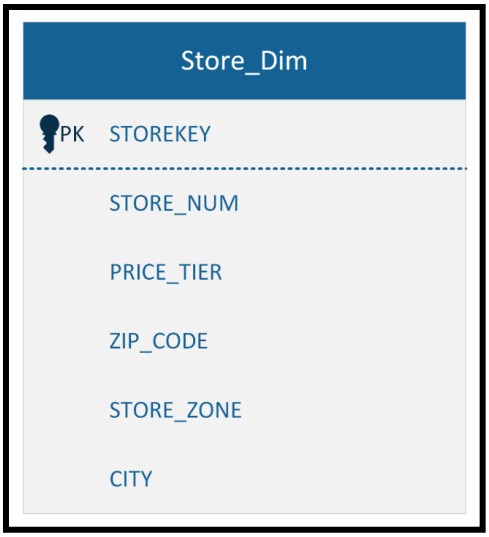
STORE_NUM: This is number given to each store which helps DFF to uniquely identify each store.

PRICE_TIER: This column indicates the price tier for a particular store as identified by DFF. They have three categories of price tiers i.e. high, medium, and low.

ZIP_CODE: This column indicates the zip code for the store.

STORE_ZONE: This column contains the zone information for each store.

CITY: This column contains the city information for each store.



Store Dimension Table

TIME_DIM TABLE

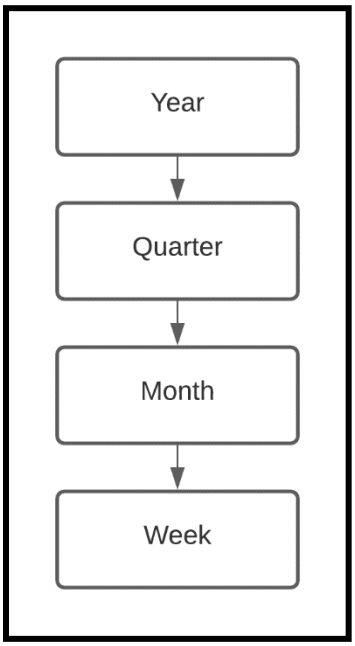
TIMEKEY: TIMEKEY is used to identify a combination of data about a particular day like the Day number, week number, month, and quarter data. This field will be system generated.

WEEK, MONTH, QUARTER, YEAR: The time information aggregated at various levels for assisting with easy usage when working on the business questions. The lowest granularity that can be achieved for this data is week number.

Time Dimension Table

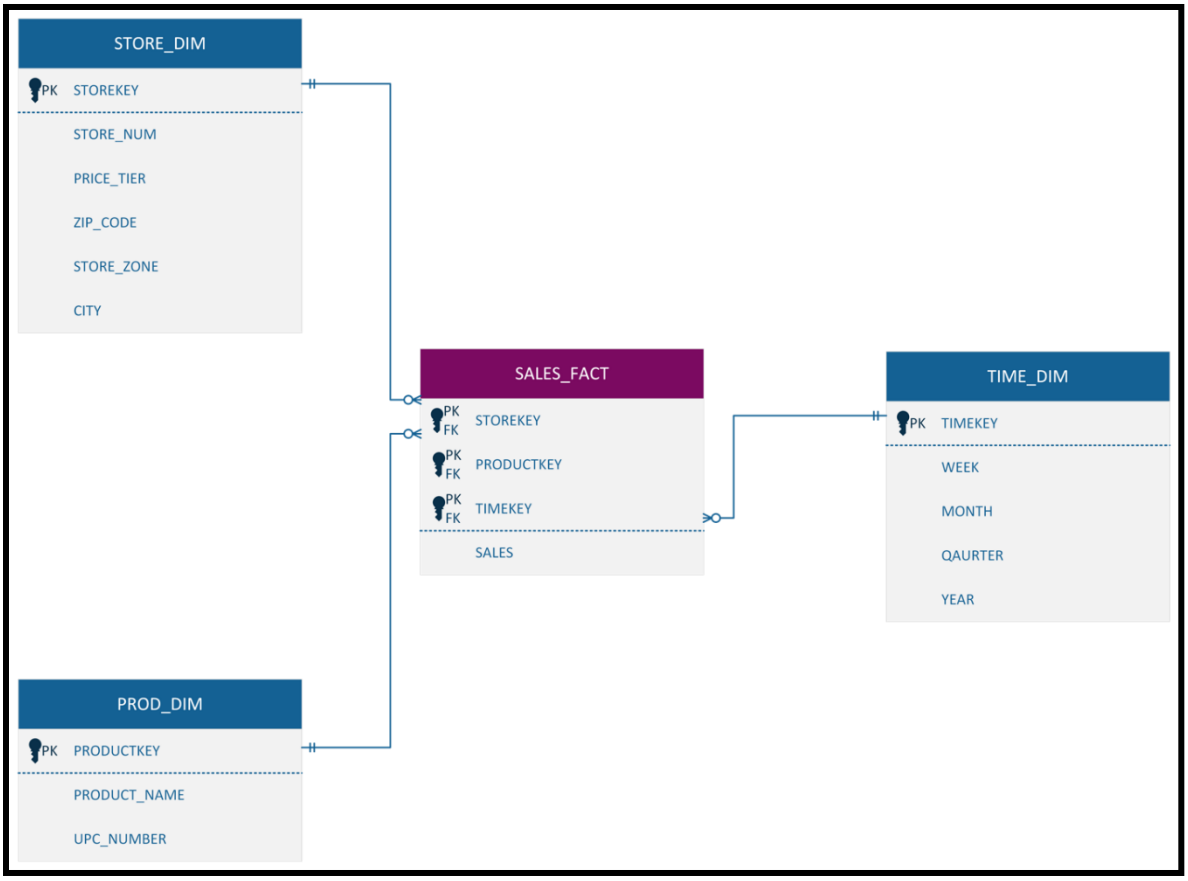
Dimension Hierarchy:

This diagram shows the drill-down path for data in Time dimension. The lowest granularity that can be achieved for this data is week number. Hence, from week we can aggregate to get data at month, quarter or year level.

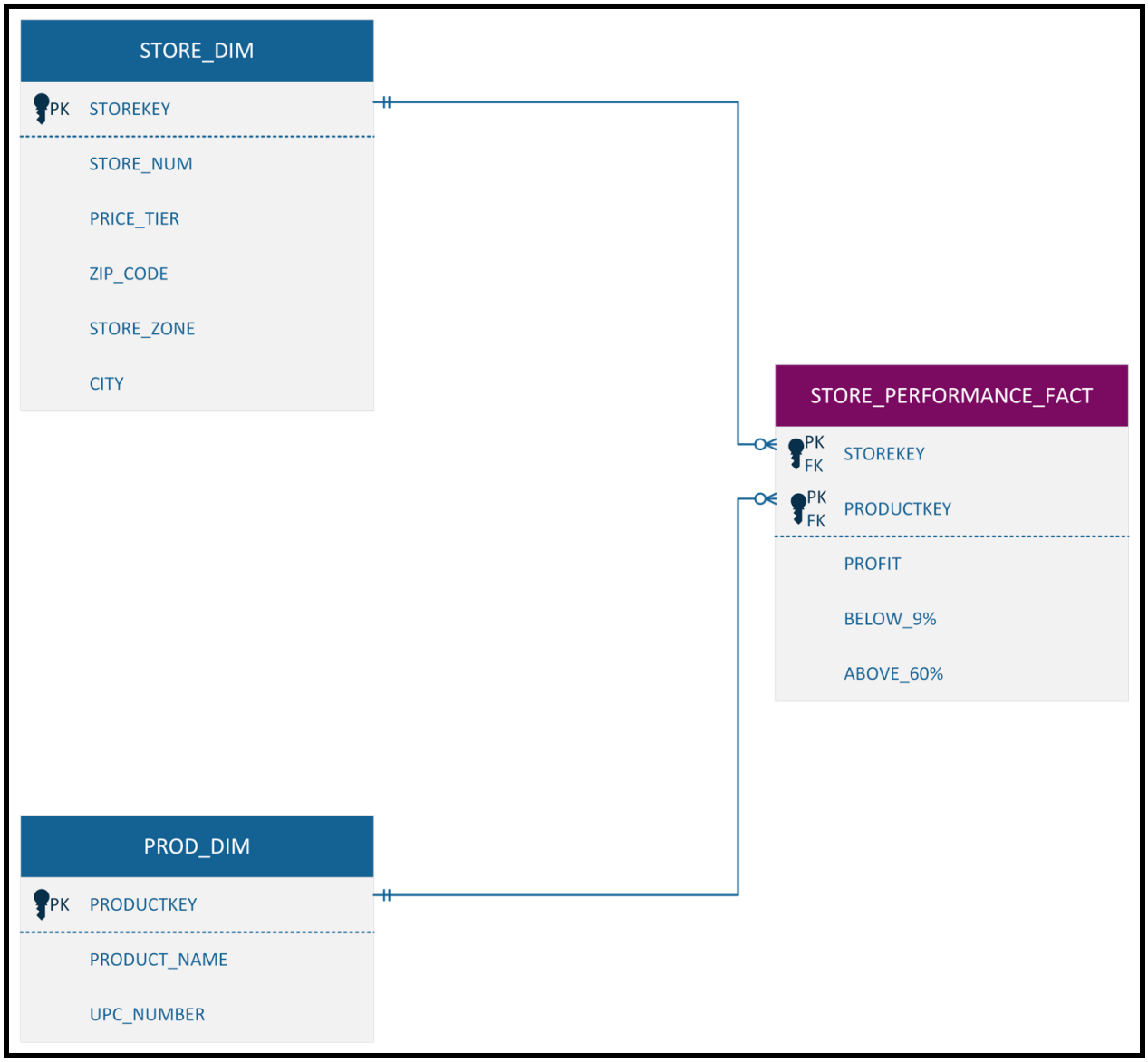


6. Star Schema Representation

Let's have a look at the star-schema design for our data marts. The first data mart is the SALES data mart, which is formed by joining SALES_FACT table with PROD_DIM, STORE_DIM and TIME_DIM tables. This data mart uses, Movement, UPC files and Store Data from the manual. The next data mart is the STORE_PERFORMANCE data mart which is formed by joining STORE_PERFORMANCE_FACT table to PROD_DIM and STORE_DIM tables. This uses Movement and Demographic files. We can also see the primary keys and foreign keys in the diagram.



SALES Data Mart



STORE_PERFORMNACE Data Mart

7. Mapping Tables

DIMENSION TABLE: STORE_DIM						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
	Surrogate Key	STORE_DIM	STOREKEY	int	Not Null	New column
Dominick_Store.csv	Store		STORE_NUM	int	Not Null	
	Price Tier		PRICE_TIER	varchar(30)		
	Zip Code		ZIP_CODE	int		
	Zone		STORE_ZONE	int		
	City		CITY	varchar(30)		

DIMENSION TABLE: PRODUCT_DIM						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
	Surrogate Key		PRODUCTKEY	int	Not Null	New column
UPC.csv	Product name	PRODUCT_DIM	PRODUCT_NAME	varchar(30)		
	UPC number		UPC_NUMBER	int	Not Null	

DIMENSION TABLE: TIME_DIM						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
	Surrogate Key	TIME_DIM	TIMEKEY	int	Not null	New column
Week Decode Table	Week number		WEEK	int	Week 1-400	
	Start Date		MONTH	int	Map month from start dates of week decode table	
	MONTH		QUARTER	int	Map quarter from MONTH column	
	Start Date		YEAR	int	Map year from start dates of week decode table	

FACT TABLE: SALES_FACT						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
STORE_DIM	STOREKEY	SALES_FACT	STOREKEY	int		Replicate
PRODUCT_DIM	PRODUCTKEY		PRODUCTKEY	int		Replicate
TIME_DIM	TIMEKEY		TIMEKEY	int		Replicate

Movement.csv	Movement, Quantity, Price		SALES	float	Computer from movement files	Sales = Price * Movement / Quantity

DIMENSION TABLE: STORE_PERFORMANCE_FACT						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
STORE_DIM	STOREKEY	STORE_PERFORMANCE_FACT	STOREKEY	int	Not Null	Replicate
TIME_DIM	TIMEKEY		TIMEKEY	int	Not Null	Replicate
Movement Files	Profit Margin		Profit	float		Profit Percent * Sales
Demography.csv	Below_9		BELOW_9%	float		Replicate
	Above_60		ABOVE_60%	float		Replicate

8. Physical Design

In order to get a complete physical model in hand, we followed all the necessary steps in the process of physical design. All these steps are explained in the sections below:

8.1. Standard Development

In the starting, we need to perform a very important step of developing standards so as to ensure consistency throughout different areas of the data warehouse. This practice improves usability and also aids in reducing complexities as both the users and the data are from different disciplines. Some of these standardization practices are:

- Naming the database objects: All the attributes, fact tables and dimension tables should be named clearly. We can use delineators to separate multiple words in complex and derived attributes. In our case, we have used underscore for the same. In addition to that, we are required to make sure that there should be a specific data type for the data which is stored in each of the field of the table. Also, it is a good practice to have the same name for the objects in both the logical model and the physical model.
- Naming tables and files in the data staging area: In the staging area, a number of intermediate and loading files are created. It is, therefore, necessary that we have some sort of standards in place so as to be able to match the various files to their respective functions. The files names should clearly indicate their purposes and should also denote the respective process. For an example, sales_units_weekly_stage
- There should also be a standard naming convention for various physical files (database files, source code files and application documents).

8.2. Aggregate Plan Creation

In the data warehouse, data can be utilized in various unpredictable ways due to which we need to store data at the level of lowest granularity. Storing data at the atomic level in data warehouses provides us with the flexibility of using the data warehouse in unpredictable ways. However, to answer the business questions for our project, we are required to aggregate profit and sales by products for separate time periods across various different stores. Thus, storing sales data and

profit data by product for different stores increases the run-time of the query and reduces performance. Consequently, to improve the performance of the system and to save the query from going through a large number of records, we can store the summarized data in tables for sales, profit and for percentage of people of certain age visiting different stores. We are going to summarize sales for different time periods (by week, by quarter and by year).

8.3. Data Partitioning

Data partitioning is important for the purpose of improving performance, functionality and also for ease of use. By this process, we can divide tables, indexes and indexed tables into additional partitions. For our project, we will use partitioning strategy based on WEEK data and QUARTER data. The business question under consideration make use of the TIME_DIM in order to calculate weekly and quarterly averages for sales and profits. This step of data partitioning will further aid us in making the process of querying easier, faster and efficient.

8.4. Indexing Strategy

Indexing is the process of creating indexes in a database to assist with the search activities when querying is performed. It eases the search process and makes it faster. Indexing creates a level of aggregation without influencing the structure of the table. Normally in data warehouse indexing is performed by recording the primary key (dimension key) in dimension table. While in fact table, one can either index and cluster on the date key or index the combination of date and time key. In our project, we have decided to use indexes on the price tiers for stores. So, we will perform indexing on the STORE_DIM dimension table where it will help us in categorizing the stores and their sales under various price tiers (high, medium, low). Further, the average profit margin from the PROD_DIM table will be indexed to help categorize the stores and the average profit margin for different products. This can be used as an aggregation criterion for the team which will help in evaluating the business question on a broader level. This will enhance the query performance and help us in answering business questions in an efficient manner. Since, over time data warehouses will change to suit what is going on due to the evolving business needs, we might update the indexing strategy to ensure that all business needs are met.

8.5. Storage Plan

The data that we are using for our project is from DFF's operational systems and is contained in a number of files. There is an area for data staging between this operational data and the data warehouse. We will perform data cleaning operations and will also transform the data into a required format at the data staging area. Operations like sorting and merging of data will also be performed here and the data is prepared for loading into the data warehouse. When the data is finally loaded in the warehouse, we can make use of this data to answer strategic questions. The size of data here is expected to increase over the years and thus we calculate the expected size of integrated and extracted data. This aids in ensuring that the performance of the system is not affected over the years as the size of the data increases.

8.6. Physical Model

TABLE/ COLUMN NAME	DATATYPE	NULLS ALLOWED ?	COMMENTS
STORE_DIM			
STOREKEY	int	N	Primary Key of STORE_DIM
STORE_NUM	int	N	Store Number for Dominick Fine Foods
PRICE_TIER	varchar (30)	N	Price Tier specific to store
ZIP_CODE	int	N	Zip Code for the store
STORE_ZONE	int	N	Detail of the Zone for the store
CITY	varchar (30)	N	Detail of the City for the store
PRODUCT_DIM			
PRODUCTKEY	int	N	Primary Key of PRODUCT_DIM
PRODUCT_NAME	varchar (30)	N	Name of the Product
UPC_NUMBER	varchar (30)	N	Unique Product Code
TIME_DIM			
TIMEKEY	int	N	Primary Key of TIME_DIM
WEEK	int	N	Week Number for the sale
MONTH	int	N	Derived from start date, MM

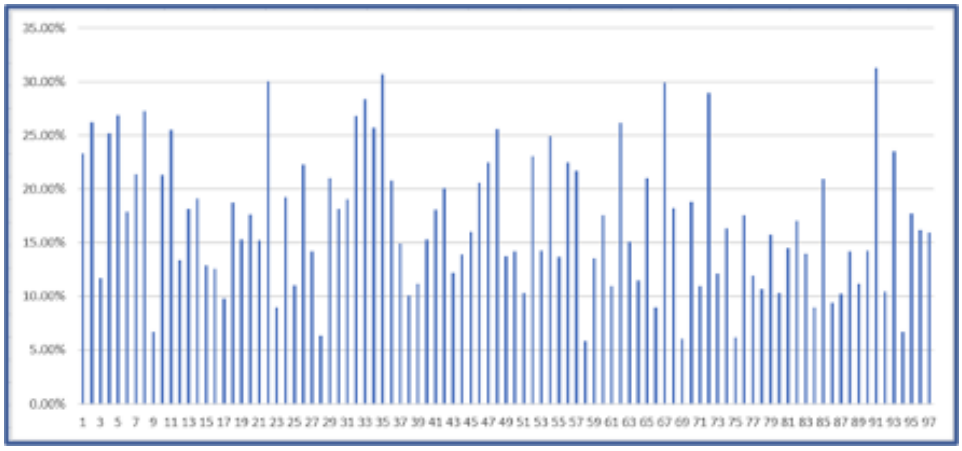
			gives us the detail of the Month
QUARTER	int	N	According to MM information in start date, quarter is derived
YEAR	int	N	Derived from start date, YY is the Year in which sales occurred
SALES_FACT			
STOREKEY	int	N	Partial primary key; foreign key referencing STORE_DIM
PRODUCTKEY	int	N	Partial primary key; foreign key referencing PRODUCT_DIM
TIMEKEY	int	N	Partial primary key; foreign key referencing TIME_DIM
SALES	float (20)	Y	Total Sales Amount of the product

TABLE/ COLUMN NAME	DATATYPE	NULLS ALLOWED?	COMMENTS
STORE_DIM			
STOREKEY	int	N	Primary Key of STORE_DIM
STORE_NUM	int	N	Store Number for Dominick Fine Foods
PRICE_TIER	varchar (30)	N	Price Tier specific to store
ZIP_CODE	int	N	Zip Code for the store
STORE_ZONE	int	N	Detail of the Zone for the store
CITY	varchar (30)	N	Detail of the City for the store
PRODUCT_DIM			
PRODUCTKEY	int	N	Primary Key of PRODUCT_DIM
PRODUCT_NAME	varchar (30)	N	Name of the Product
UPC_NUMBER	varchar (30)	N	Unique Product Code
STORE_PERFORMANCE_FACT			
PRODUCTKEY	int	N	Partial primary key; foreign key referencing PRODUCT_DIM
STOREKEY	int	N	Partial primary key; foreign key referencing STORE_DIM
PROFIT	float (20)	Y	Gross Profit of the product

BELOW_9%	float (20)	Y	% Population under age 9
ABOVE_60%	float (20)	Y	% Population over age 60

9. Schema Justification for each Business Question

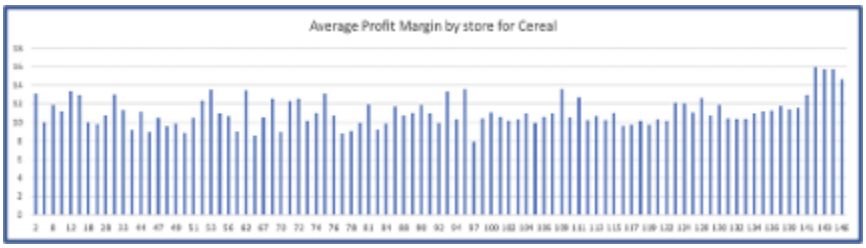
9.1. Which are the stores that have higher popularity amongst kids and amongst elderly groups of people?



Store-wise percentage visit of Elderly(age>60)

In order to answer this question, we will evaluate and compare the Store-wise percentage visit of kids (age<9) and Store-wise percentage visit of Elderly(age>60). This can be achieved by using STORE_PERFORMANCE data mart. The STORE_DIM dimension table will provide information regarding the different stores using the attribute STOREKEY. The STORE_PERFORMANCE_FACT fact table has attributes BELOW_9% and ABOVE_60%. This information will then be used to obtain the percentage visits for different stores by kids and elderly groups for and they will be analyzed separately to answer this question.

9.2. Which stores have profit below the average profit margin of Cereal (WCER)?



Average Profit across stores for Cereal

In order to answer this question, we will evaluate the average profit margin for each store and compare it to the overall average profit margin. This can be achieved through the STORE_PERFORMANCE data mart. The PROD_DIM dimension table will provide information the specific product i.e. cereal. The STORE_DIM dimension table will provide information regarding the different stores using the attribute STOREKEY. Furthermore, the STORE_PERFORMANCE_FACT fact table has attribute PROFIT which will be used to evaluate the average profit margin for the cereal for different stores.

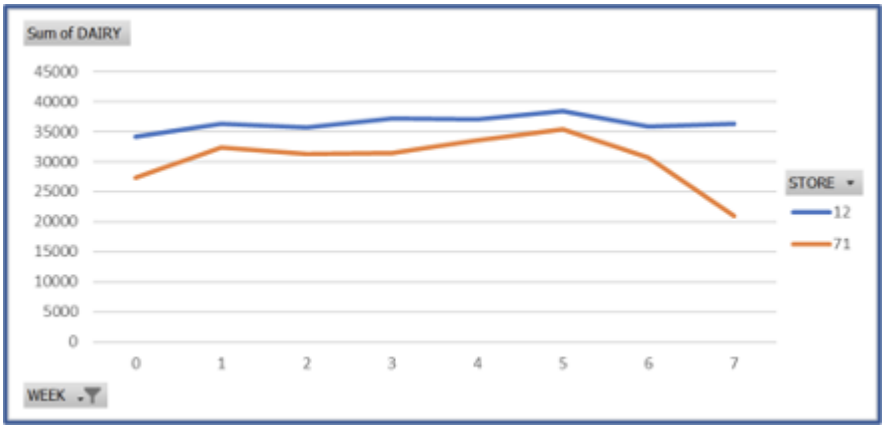
9.3. How have the sales of frozen products varied over quarters in a particular year?

Total Sales of Frozen Products per quarter per year

In order to answer this question, we will aggregate the sales (weekly) of frozen products across different quarters for every year. This can be achieved by using the SALES data mart. To this end, PROD_DIM dimension table will provide information the specific product i.e. Frozen via PRODUCTKEY. The TIME_DIM dimension table will be used to obtain the week details via

attribute WEEK. Finally, the SALES_FACT fact table which will provide information about the sales via attribute Sales. In this way, we will be able to evaluate the quarterly sales of frozen products and we will then use it to compare the sales of frozen products over different quarter in different years.

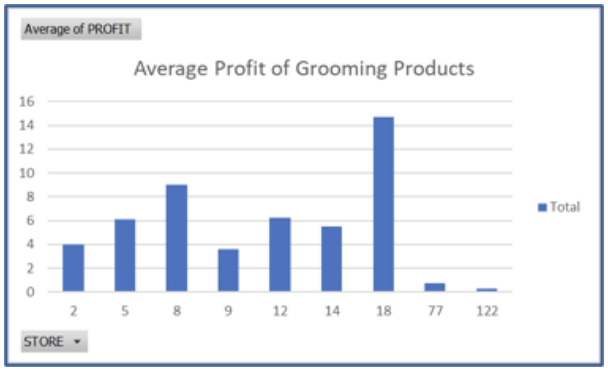
9.4. How is the demand for a particular product (Dairy) changing in different price-tiers?



Weekly sales of Dairy in store 71(medium-price tier) store 12(high-price tier)

In order to answer this business question, we need to analyze the sales of the dairy product for different stores belonging to different product tiers. This can be achieved by using SALES data mart. To this end, PROD_DIM dimension table will provide information the specific product i.e. Dairy via PRODUCTKEY. The STORE_DIM dimension table will provide information regarding the different stores using the attribute STOREKEY. It will also provide information regarding different price tiers using the attribute PRICE_TIER. The TIME_DIM dimension table will be used to obtain the week information via attribute WEEK. Furthermore, the information regarding the average sales can be derived using Sales attribute in SALES_FACT fact table. This will eventually be used to compare the weekly sales of dairy products in different stores belonging to different product tiers.

9.5. What is the store-wise average profit of particular product category (Grooming products)?



Average store-wise profit on grooming products

In order to answer this business question, we need to compute the average profit margin for each store in grooming product category. This can be achieved by using STORE_PERFORMANCE data mart. To this end, PROD_DIM dimension table will provide information about the specific product i.e. Grooming product via attribute PRODUCTKEY. The STORE_DIM dimension table will provide information regarding the different stores using the attribute STOREKEY. Finally, the STORE_PERFORMANCE_FACT fact table which will provide information about the sales via attribute Sales. Thus, we will compute the average profit margin per store and eventually obtain comparative analysis of average store-wise profit on grooming products.

10. Development of ETL Plan (Data Cleaning and Integration)

10.1. Determining the Target Data

The solution that we have chosen for our project consists of three dimensions and two fact tables. All of the tables along with all their attributes and data types are listed out below:

Dimension Tables:

DESTINATION: STORE_DIM		
DW Target Table	DW Target Column	Target Datatype
Group10_602-dw-area.STORE_DIM	STOREKEY	int
	STORE_NUM	int

	PRICE_TIER	varchar (30)
	ZIP_CODE	varchar (30)
	STORE_ZONE	varchar (30)
	CITY	varchar (30)

DESTINATION: PRODUCT_DIM		
DW Target Table	DW Target Column	Target Datatype
Group10_602-dw-area.PRODUCT_DIM	PRODUCTKEY	int
	PRODUCT_NAME	varchar (30)
	UPC_NUMBER	int

DESTINATION: TIME_DIM		
DW Target Table	DW Target Column	Target Datatype
Group10_602-dw-area.TIME_DIM	TIMEKEY	int
	WEEK	int
	MONTH	int
	QUARTER	int
	YEAR	int

Fact Tables:

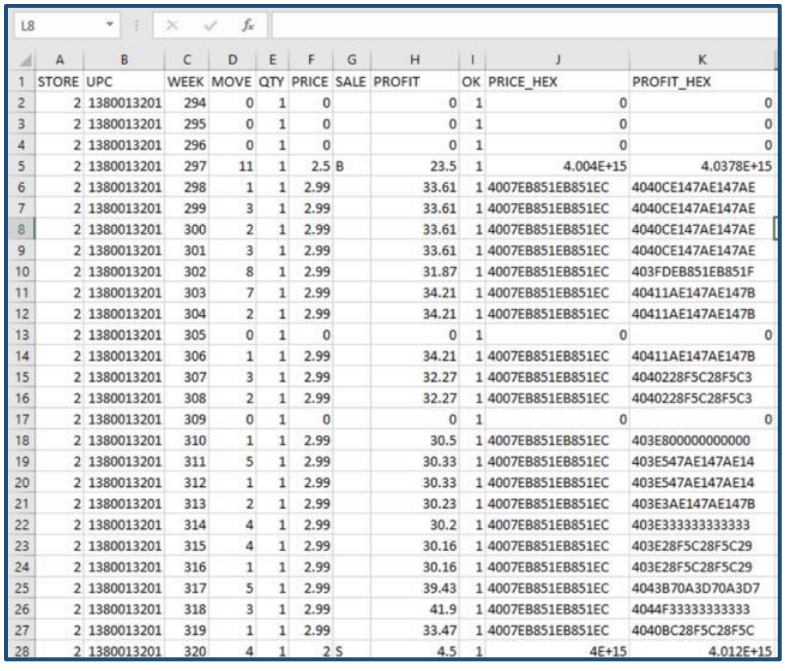
DESTINATION: SALES_FACT		
Target Table	Target Column	Target Datatype
Group10_602-dw-area.SALES_FACT	STOREKEY	int
	PRODUCTKEY	int
	TIMEKEY	int
	SALES	float

DESTINATION: STORE_PERF_FACT		
Target Table	Target Column	Target Datatype
Group10_602-dw-area.STORE_PERFORMANCE_FACT	STOREKEY	int
	PRODUCTKEY	int
	PROFIT	float
	BELOW_9%	float
	ABOVE_60%	float

10.2. Determining the Source Data

Source data for the schema proposed here is coming from DEMO.csv, MOVEMENT files (WFRD, WFRE, WFRJ, WGRO, WCHE, WCER), UPC files (UPCCER, UPCCHE, UPCFRD, UPCFRE, UPCFRJ, UPCGRO), Dominick's Stores Table (Part 6) and Weeks' Decode Table (Part 8) given in Dominick's FF Data Manual.

I) WFRD.CSV



	A	B	C	D	E	F	G	H	I	J	K
1	STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK	PRICE_HEX	PROFIT_HEX
2	2	1380013201	294	0	1	0		0	1	0	0
3	2	1380013201	295	0	1	0		0	1	0	0
4	2	1380013201	296	0	1	0		0	1	0	0
5	2	1380013201	297	11	1	2.5	B	23.5	1	4.004E+15	4.0378E+15
6	2	1380013201	298	1	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
7	2	1380013201	299	3	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
8	2	1380013201	300	2	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
9	2	1380013201	301	3	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
10	2	1380013201	302	8	1	2.99		31.87	1	4007EB851EB851EC	403FDEB851EB851F
11	2	1380013201	303	7	1	2.99		34.21	1	4007EB851EB851EC	40411AE147AE147B
12	2	1380013201	304	2	1	2.99		34.21	1	4007EB851EB851EC	40411AE147AE147B
13	2	1380013201	305	0	1	0		0	1	0	0
14	2	1380013201	306	1	1	2.99		34.21	1	4007EB851EB851EC	40411AE147AE147B
15	2	1380013201	307	3	1	2.99		32.27	1	4007EB851EB851EC	4040228F5C28F5C3
16	2	1380013201	308	2	1	2.99		32.27	1	4007EB851EB851EC	4040228F5C28F5C3
17	2	1380013201	309	0	1	0		0	1	0	0
18	2	1380013201	310	1	1	2.99		30.5	1	4007EB851EB851EC	403E800000000000
19	2	1380013201	311	5	1	2.99		30.33	1	4007EB851EB851EC	403E547AE147AE14
20	2	1380013201	312	1	1	2.99		30.33	1	4007EB851EB851EC	403E547AE147AE14
21	2	1380013201	313	2	1	2.99		30.23	1	4007EB851EB851EC	403E3AE147AE147B
22	2	1380013201	314	4	1	2.99		30.2	1	4007EB851EB851EC	403E333333333333
23	2	1380013201	315	4	1	2.99		30.16	1	4007EB851EB851EC	403E28F5C28F5C29
24	2	1380013201	316	1	1	2.99		30.16	1	4007EB851EB851EC	403E28F5C28F5C29
25	2	1380013201	317	5	1	2.99		39.43	1	4007EB851EB851EC	4043B70A3D70A3D7
26	2	1380013201	318	3	1	2.99		41.9	1	4007EB851EB851EC	4044F33333333333
27	2	1380013201	319	1	1	2.99		33.47	1	4007EB851EB851EC	4040BC28F5C28F5C
28	2	1380013201	320	4	1	2	S	4.5	1	4E+15	4.012E+15

II) WFRE.CSV

1	STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK	PRICE_HEX	PROFIT_HEX
2	62	1242409401	384	0	1	0		0	1	0	0
3	62	1242409401	385	3	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
4	62	1242409401	386	3	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
5	62	1242409401	387	5	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
6	62	1242409401	388	3	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
7	62	1242409401	389	3	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
8	62	1242409401	390	5	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
9	62	1242409401	391	4	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
10	62	1242409401	392	2	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
11	62	1242409401	393	2	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
12	62	1242409401	394	1	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
13	62	1242409401	395	1	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
14	62	1242409401	396	0	1	0		0	1	0	0
15	62	1242409401	397	2	1	4.79		45.72	1	401328F5C28F5C29	4046DC28F5C28F5C
16	62	1242409401	398	0	1	0		0	1	0	0
17	62	1242409401	399	0	1	0		0	1	0	0
18	62	1242409402	384	0	1	0		0	1	0	0
19	62	1242409402	385	8	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
20	62	1242409402	386	5	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
21	62	1242409402	387	8	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
22	62	1242409402	388	5	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
23	62	1242409402	389	5	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
24	62	1242409402	390	6	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
25	62	1242409402	391	6	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
26	62	1242409402	392	3	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
27	62	1242409402	393	1	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
28	62	1242409402	394	3	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8
29	62	1242409402	395	3	1	4.79		27.72	1	401328F5C28F5C29	403BB851EB851EB8

III) WFRJ.CSV

1	STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK	PRICE_HEX	PROFIT_HEX
2	2	1110000139	1	14	1	1.79		43.18	1	3FFCA3D70A3D70A4	4045970A3D70A3D7
3	2	1110000139	2	48	1	1.79		43.18	1	3FFCA3D70A3D70A4	4045970A3D70A3D7
4	2	1110000139	3	156	1	1.29	B	21.16	1	3FF4A3D70A3D70A4	403528F5C28F5C29
5	2	1110000139	4	4	1	1.29	B	22.17	1	3FF4A3D70A3D70A4	40362B851EB851EC
6	2	1110000139	5	51	1	1.79		43.91	1	3FFCA3D70A3D70A4	4045F47AE147AE14
7	2	1110000139	6	145	1	1.29	B	22.17	1	3FF4A3D70A3D70A4	40362B851EB851EC
8	2	1110000139	7	27	1	1.79		43.91	1	3FFCA3D70A3D70A4	4045F47AE147AE14
9	2	1110000139	8	22	1	1.79		43.91	1	3FFCA3D70A3D70A4	4045F47AE147AE14
10	2	1110000139	9	14	1	1.62		38.02	1	3FF9EB851EB851EC	4043028F5C28F5C3
11	2	1110000139	10	65	1	1.39		28.27	1	3FF63D70A3D70A3D	403C451EB851EB85
12	2	1110000139	11	10	1	1.39		33.6	1	3FF63D70A3D70A3D	4040CCCCCCCCCCD
13	2	1110000139	12	14	1	1.62		43.02	1	3FF9EB851EB851EC	4045828F5C28F5C3
14	2	1110000139	13	4	1	1.62		43.02	1	3FF9EB851EB851EC	4045828F5C28F5C3
15	2	1110000139	14	17	1	1.62		43.02	1	3FF9EB851EB851EC	4045828F5C28F5C3
16	2	1110000139	15	13	1	1.62		43.02	1	3FF9EB851EB851EC	4045828F5C28F5C3
17	2	1110000139	16	23	1	1.62		43.02	1	3FF9EB851EB851EC	4045828F5C28F5C3
18	2	1110000139	17	13	1	1.62		43.02	1	3FF9EB851EB851EC	4045828F5C28F5C3
19	2	1110000139	18	16	1	1.62		43.02	1	3FF9EB851EB851EC	4045828F5C28F5C3
20	2	1110000139	19	9	1	1.62		42.41	1	3FF9EB851EB851EC	4045347AE147AE14
21	2	1110000139	20	16	1	1.62		42.41	1	3FF9EB851EB851EC	4045347AE147AE14
22	2	1110000139	21	10	1	1.79		47.88	1	3FFCA3D70A3D70A4	4047F0A3D70A3D71
23	2	1110000139	22	0	1	0		0	1	0	0
24	2	1110000139	23	12	1	2.19		57.4	1	4001851EB851EB85	404CB33333333333
25	2	1110000139	24	277	1	1.39	S	24.32	1	3FF63D70A3D70A3D	403851EB851EB852
26	2	1110000139	25	7	1	2.19		51.96	1	4001851EB851EB85	4049FAE147AE147B
27	2	1110000139	26	6	1	2.19		51.96	1	4001851EB851EB85	4049FAE147AE147B
28	2	1110000139	27	16	1	2.13	B	50.61	1	40010A3D70A3D70A	40494E147AE147AE
29	2	1110000139	28	14	1	2.13	B	50.61	1	40010A3D70A3D70A	40494E147AE147AE

IV) WGRO.CSV

UPC	STORE	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK	PRICE_HEX	PROFIT_HEX
707300856	77	128	0	1	0		0	1	0	0
707300856	77	129	0	1	0		0	1	0	0
707300856	77	130	0	1	0		0	1	0	0
707300856	77	131	0	1	0		0	1	0	0
707300856	77	132	0	1	0		0	1	0	0
707300856	77	133	0	1	0		0	1	0	0
707300856	77	134	0	1	0		0	1	0	0
707300856	77	135	0	1	0		0	1	0	0
707300856	77	136	0	1	0		0	1	0	0
707300856	77	137	0	1	0		0	1	0	0
707300856	77	138	0	1	0		0	1	0	0
707300856	77	139	0	1	0		0	1	0	0
707300856	77	140	0	1	0		0	1	0	0
707300856	77	141	0	1	0		0	1	0	0
707300856	77	142	0	1	0		0	1	0	0
707300856	77	143	0	1	0		0	1	0	0
707300856	77	144	0	1	0		0	1	0	0
707300856	77	145	0	1	0		0	1	0	0
707300856	77	146	0	1	0		0	1	0	0
707300856	77	147	0	1	0		0	1	0	0
707300856	77	148	0	1	0		0	1	0	0
707300856	77	149	0	1	0		0	1	0	0
707300856	77	150	0	1	0		0	1	0	0
707300856	77	151	1	1	4.29		30.54	1	401128F5C28F5C29	403E8A3D70A3D70A
707300856	77	152	0	1	0		0	1	0	0
707300856	77	153	1	1	4.49		33.63	1	4011F5C28F5C28F6	4040D0A3D70A3D71
707300856	77	154	0	1	0		0	1	0	0

V) WCER.CSV

1	STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK	PRICE_HEX	PROFIT_HEX
2	51	317	372	1	1	26.02		62.52	1	403A051EB851EB85	404F428F5C28F5C3
3	51	317	373	0	1	0		0	1	0	0
4	51	317	374	0	1	0		0	1	0	0
5	51	317	375	0	1	0		0	1	0	0
6	51	317	376	0	1	0		0	1	0	0
7	51	317	377	0	1	0		0	1	0	0
8	51	317	378	0	1	0		0	1	0	0
9	51	317	379	0	1	0		0	1	0	0
10	51	317	380	0	1	0		0	1	0	0
11	51	317	381	0	1	0		0	1	0	0
12	51	317	382	0	1	0		0	1	0	0
13	51	317	383	0	1	0		0	1	0	0
14	51	317	384	0	1	0		0	1	0	0
15	51	317	385	0	1	0		0	1	0	0
16	51	317	386	0	1	0		0	1	0	0
17	51	317	387	0	1	0		0	1	0	0
18	51	317	388	0	1	0		0	1	0	0
19	51	317	389	0	1	0		0	1	0	0
20	51	317	390	0	1	0		0	1	0	0
21	51	317	391	0	1	0		0	1	0	0
22	51	317	392	0	1	0		0	1	0	0
23	51	317	393	0	1	0		0	1	0	0
24	51	317	394	0	1	0		0	1	0	0
25	51	317	395	0	1	0		0	1	0	0
26	51	317	396	0	1	0		0	1	0	0
27	51	317	397	0	1	0		0	1	0	0
28	51	317	398	0	1	0		0	1	0	0
29	51	317	399	0	1	0		0	1	0	0

VI) WCHE.CSV

	STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK	PRICE_HEX	PROFIT_HEX
2	2	1570077445	1	7	1	2.41		37.59	1	400347AE147AE148	4042CB851EB851EC
3	2	1570077445	2	8	1	2.41		37.59	1	400347AE147AE148	4042CB851EB851EC
4	2	1570077445	3	4	1	2.41		37.14	1	400347AE147AE148	404291EB851EB852
5	2	1570077445	4	8	1	2.41		37.14	1	400347AE147AE148	404291EB851EB852
6	2	1570077445	5	7	1	2.41		37.14	1	400347AE147AE148	404291EB851EB852
7	2	1570077445	6	5	1	2.41		37.39	1	400347AE147AE148	4042B1EB851EB852
8	2	1570077445	7	12	1	2.41		36.85	1	400347AE147AE148	40426CCCCCCCCCCD
9	2	1570077445	8	10	1	2.09		27.18	1	4000B851EB851EB8	403B2E147AE147AE
10	2	1570077445	9	12	1	2.09		27.42	1	4000B851EB851EB8	403B6B851EB851EC
11	2	1570077445	10	17	1	2.09	B	27.42	1	4000B851EB851EB8	403B6B851EB851EC
12	2	1570077445	11	3	1	2.09		27.37	1	4000B851EB851EB8	403B5EB851EB851F
13	2	1570077445	12	9	1	2.41		37.01	1	400347AE147AE148	40428147AE147AE1
14	2	1570077445	13	10	1	2.41		25.31	1	400347AE147AE148	40394F5C28F5C28F
15	2	1570077445	14	10	1	2.41		25.39	1	400347AE147AE148	403963D70A3D70A4
16	2	1570077445	15	9	1	2.63		31.75	1	40050A3D70A3D70A	403FC00000000000
17	2	1570077445	16	4	1	2.63		31.75	1	40050A3D70A3D70A	403FC00000000000
18	2	1570077445	17	14	1	2.63		31.67	1	40050A3D70A3D70A	403FAB851EB851EC
19	2	1570077445	18	8	1	2.63		35.29	1	40050A3D70A3D70A	4041A51EB851EB85
20	2	1570077445	19	15	1	1.97		17.72	1	3FFF851EB851EB85	4031B851EB851EB8
21	2	1570077445	20	7	1	2.63		38.71	1	40050A3D70A3D70A	40435AE147AE147B
22	2	1570077445	21	9	1	2.63		36.2	1	40050A3D70A3D70A	404219999999999A
23	2	1570077445	22	7	1	2.63		34.87	1	40050A3D70A3D70A	40416F5C28F5C28F
24	2	1570077445	23	9	1	2.63		34.94	1	40050A3D70A3D70A	40417851EB851EB8
25	2	1570077445	24	7	1	2.63		35.25	1	40050A3D70A3D70A	4041A00000000000
26	2	1570077445	25	11	1	2.49		31.61	1	4003EB851EB851EC	403F9C28F5C28F5C
27	2	1570077445	26	4	1	2.49	B	31.61	1	4003EB851EB851EC	403F9C28F5C28F5C
28	2	1570077445	27	9	1	2.51	B	34.22	1	4004147AE147AE14	40411C28F5C28F5C
29	2	1570077445	28	10	1	2.51	B	35.22	1	4004147AE147AE14	40419C28F5C28F5C

VII) DEMO.CSV

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	MMID	NAME	CITY	ZIP	LAT	LONG	WEEKVOL	STORE	SCLUSTER	ZONE	AGE9	AGE60	ETHNIC	EDUC	NOCAR	INCOME	INCSIGMA	GINI	HSIZEAVG	HSIZE1	HSIZE2
2	.																				
3	16892	DOMINICK	RIVER FOF	60305	419081	878131	350	2	C	1	0.117509	0.232865	0.11428	0.248935	0.124603	10.55321	26296.9	.	2.531062	0.282033	0.312919
4	16893	DOMINICK	PARK RIDG	60068	420392	878425	300	4	A	2	0.09509	0.26203	0.062161	0.220789	0.055567	10.64697	24885.18	.	2.480347	0.269442	0.338757
5	16894	DOMINICK	PALATINE	60067	421203	880431	550	5	D	2	0.141433	0.117368	0.053875	0.321226	0.02557	10.92237	26779.61	.	2.656439	0.218852	0.335077
6	16895	DOMINICK	OAK LAWI	60453	417331	877436	600	8	C	5	0.123155	0.252394	0.035243	0.095173	0.075113	10.59701	24653.87	.	2.769603	0.210822	0.314418
7	16896	DOMINICK	MORTON	60053	420411	877994	450	9	A	2	0.103503	0.269119	0.032619	0.222172	0.040128	10.78715	26599.04	.	2.616894	0.211544	0.35868
8	16898	DOMINICK	CHICAGO	60660	419928	876592	450	12	B	7	0.105697	0.178341	0.380698	0.253413	0.483518	9.996659	22375.07	.	1.959018	0.492595	0.282816
9	16899	DOMINICK	GLENVIEW	60025	420733	877994	400	14	A	1	0.129589	0.213949	0.034179	0.348293	0.026586	11.04393	28371.71	.	2.735061	0.186332	0.338832
10	16901	DOMINICK	RIVER GRO	60171	419364	878331	600	18	A	5	0.110095	0.272313	0.074417	0.072246	0.141975	10.39198	23126.8	.	2.530338	0.268805	0.326691
11	.							19													
12	16903	DOMINICK	HANOVER	60103	420058	881411	500	21	D	6	0.175926	0.066896	0.105039	0.177503	0.017598	10.71619	21437.77	.	3.110391	0.138834	0.266122
13	.							25													
14	16905	DOMINICK	MOUNT PI	60056	420686	879208	275	28	A	2	0.12888	0.213309	0.055935	0.233163	0.054855	10.79853	26203.64	.	2.646509	0.2106	0.356292
15	16906	DOMINICK	PARK RIDG	60068	419872	878378	575	32	C	1	0.099061	0.254953	0.031939	0.19826	0.071701	10.67448	25506.95	.	2.401154	0.290306	0.343566
16	16907	DOMINICK	CHICAGO	60657	419386	876447	300	33	B	7	0.046071	0.13417	0.130127	0.419688	0.506224	10.34593	25921.61	.	1.55429	0.61401	0.280456
17	.							39													
18	16909	DOMINICK	BRIDGEVII	60455	417317	877969	500	40	D	6	0.133685	0.181852	0.044053	0.072129	0.04633	10.55025	22767.84	.	2.730972	0.229367	0.30086
19	16912	DOMINICK	WESTERN	60558	418033	878903	325	44	A	2	0.144883	0.190983	0.037632	0.329738	0.040766	10.86916	27842.3	.	2.778435	0.173053	0.346514
20	16913	DOMINICK	WHEELING	60090	421403	879300	300	45	D	2	0.146719	0.128857	0.087234	0.28015	0.020232	10.74538	25056.31	.	2.498898	0.262568	0.338073
21	.							46	D	5											
22	16915	DOMINICK	ADDISON	60101	419364	880022	350	47	D	2	0.142962	0.125798	0.120676	0.140599	0.021297	10.63533	23887.56	.	2.924399	0.17562	0.292049
23	16916	DOMINICK	SCHAUMB	60193	420503	880775	325	48	D	2	0.121767	0.097922	0.099492	0.30326	0.021209	10.75603	25334.74	.	2.352837	0.324907	0.325968
24	16917	DOMINICK	DOWNERS	60515	418111	879869	275	49	A	2	0.134878	0.187473	0.038353	0.31995	0.054382	10.80675	27361.59	.	2.586292	0.255506	0.319101
25	16918	DOMINICK	HICKORY H	60457	417169	878347	275	50	A	2	0.12442	0.153357	0.070926	0.128764	0.036434	10.58931	23495.48	.	2.651317	0.233448	0.3189
26	16919	DOMINICK	PALOS HE	60463	416594	877775	400	51	D	3	0.132472	0.17616	0.025426	0.171917	0.025436	10.62084	24312.73	.	2.784705	0.199667	0.315711
27	16920	DOMINICK	NORTHBR	60062	421364	878825	450	52	A	1	0.136606	0.152241	0.084899	0.372927	0.0149	11.05102	28792.41	.	2.678316	0.205958	0.342309
28	16921	DOMINICK	CHICAGO	60662	420039	877069	300	53	C	7	0.120839	0.300279	0.065722	0.270383	0.145363	10.6	26725.53	.	2.554727	0.279379	0.330377
29	16922	DOMINICK	NAPERVIL	60540	417975	881225	375	54	D	2	0.147915	0.090222	0.046641	0.421126	0.02084	10.9109	27006.45	.	2.578799	0.26724	0.304448

VIII) UPCCER.CSV

1	COM_CODE	UPC	DESCRIP	SIZE	CASE	NITEM
2	311	317	$TONY THE TIGER T-SH	ASST	1	9900090
3	311	1313000002	NAB SHREDDED WHEAT	10 OZ	12	2513101
4	311	1313000005	NABISCO SHREDDED WHE	15 OZ	12	2513120
5	311	1313000032	NABISCO 100% BRAN	17 OZ	12	2513951
6	311	1313000050	NAB SPOON SIZE SHRED	23.45O	12	2516761
7	311	1313000054	NAB SPOON SIZE SHRED	17.2 O	12	2513051
8	311	1313000148	NABISCO TEAM FLAKES	13 OZ	12	2513600
9	311	1313000698	NABISCO CREAM OF RIC	14 OZ	12	251690
10	311	1313001041	NABISCO OAT BRAN RAI	15.5 O	12	2513551
11	311	1313001059	NABISCO WHEAT N BRAN	18 OZ	12	2513131
12	311	1313001145	NABISCO FRUIT WHEATS	15.5 O	12	2513500
13	311	1313001146	NAB RASPBERRY FRUIT	15.5 O	12	251352
14	311	1313001854	NAB SHRED WHT W/OAT	18 OZ	12	2513121
15	311	1313002033	NABISCO 100% BRAN W/	17 OZ	12	2513900
16	311	1313002401	TEDDY GRAHAMS BRKFST	15 OZ	12	2513201
17	311	1313002402	TEDDY GRAHAMS BRKFST	15 OZ	12	2513251
18	311	1313002403	TEDDY GRAHAMS BRKFST	15 OZ	12	2513351
19	311	1313005152	NAB FROST WHEAT SQUA	15 OZ	12	2513161
20	311	1313005154	NAB FROSTED WHEAT	20 OZ	12	2508120
21	311	1313018155	~NABISCO FROSTED WHE	15 OZ	14	2513121
22	311	1600000000	BERRY KIX/WHT TOTAL/	1 CT	1	9825091
23	311	1600062360	TRIPLES/KRAFT MARSHM	18.7 O	21	2508221
24	311	1600062370	$ GEN MILLS TRIPLES	13.3 O	16	2508201
25	311	1600062380	G. MILLS TRIPLES	18.75O	14	2508221
26	311	1600062580	G. MILLS BENEFIT W/R	15 OZ	12	250440
27	311	1600062600	G. MILLS BENEFIT CER	12.5 O	12	250435
28	311	1600062640	G.M. BERRY BERRY KIX	18 OZ	14	2503600

IX) UPCCHE.CSV

1	COM_CODE	UPC	DESCRIP	SIZE	CASE	NITEM
2	153	1570077445	CTY LN COLBY MILD	10 OZ	12	5390551
3	153	1570077446	CTY LN CHEDDAR MILD	10 OZ	12	5390571
4	153	1570077454	CTY LN CHEDDAR MED/S	8 OZ	12	5390591
5	153	1570077455	CTY LN CHEDDAR SHARP	8 OZ	12	5390511
6	153	1570077456	CTY LN SWISS OLD WOR	10 OZ	12	5390531
7	153	1570077458	COUNTY LINE/COLBY JA	10 OZ	12	5390621
8	153	1570077501	CTY LN MONTEREY JACK	10 OZ	12	5390491
9	153	1570077503	~COUNTY LINE MUENSTE	8 OZ	12	5390441
10	153	1570077507	~COUNTY LINE MOZZ CH	8 OZ	12	5390421
11	153	1570077508	COUNTY LINE EXTRA SH	8 OZ	12	5390401
12	153	1700310035	KAUKAUNA RANCH BALLS	10 OZ	12	9835471
13	153	1700310039	KAUKAUNA GARDEN VEGT	10 OZ	12	9835461
14	153	1700310062	KAUKAUNA SHARP CHEES	10 OZ	12	9835491
15	153	1700310067	KAUKAUNA SHARP CHED	10 OZ	12	9835491
16	153	1700310068	KAUKAUNA PORT WINE C	10 OZ	12	9835481
17	156	1700310074	KAUK CHEDDAR CHEESE	7 OZ	12	9835501
18	156	1700310077	KAUK PORT WINE CHEES	7 OZ	1	9835511
19	156	1700310081	KAUKAUNA SHARP LITE	8 OZ	12	5394051
20	153	1700310084	KAUKAUNA PORTWINE CH	12 OZ	12	9835511
21	153	1700310087	KAUKAUNA SHARP BALLS	12 OZ	12	9835501
22	156	1700310091	KAUKAUNA SHARP CHEDD	8 OZ	12	5394031
23	156	1700310104	KAUKAUNA GARDEN VEG	8 OZ	12	5349071
24	156	1707720116	LIFEWAY'S FARMER'S C	16 OZ	18	5070300
25	156	1707730108	LIFEWAY'S SWEET KIS	8 OZ	42	5070350
26	153	2070068305	BLACK DIAMOND CHEDDA	8 OZ	12	5365750
27	157	2100000001	KR PHILADELPHIA CR C	3 OZ	24	5351501
28	157	2100000002	KR PIMENTO PHILLY CC	3 OZ	6	5343501

X) UPCFRD.CSV

1	COM_CODE	NITEM	UPC	DESCRIP	SIZE	CASE
2	104	9309691	1380013201	LC HP CHICKEN FLOREN	13.2OZ	12
3	104	9309711	1380013202	LC HP ROASTED TURKEY	14 OZ	12
4	104	9309771	1380013203	LC HP SRLN BF TIPS	14.25O	12
5	104	9309791	1380013204	LC HP GRLD CHKN W/PE	14 OZ	12
6	104	9309801	1380013205	LC HP JUMBO RIGATONI	15.3OZ	12
7	104	9309861	1380013206	LC HP ORIENTAL GLAZE	14 OZ	12
8	104	9309651	1380013207	LC HP BEEF LO MEIN	14 OZ	12
9	104	9309671	1380013208	LC HP ROASTED CHICKE	12.5 O	12
10	104	9310121	1380013209	LC HEARTY PORTIONS L	15 OZ	12
11	104	9310141	1380013210	LC HRTY PORTIONS CHS	15.5 O	12
12	104	9310041	1380013304	STOUFFER'S HRTY PRTN	17 OZ	12
13	104	9309931	1380013305	STOUFFER'S HRTY PRTN	16 OZ	12
14	104	9310051	1380013306	STOUFFER'S HRTY PRTN	15.1OZ	12
15	104	9309951	1380013307	STOUFFER'S HRTY PRTN	16.75Z	12
16	104	9309991	1380013308	STOUFFER'S HRTY PRTN	16 OZ	12
17	104	9310001	1380013309	STOUFFER'S HRTY PRTN	17.5OZ	12
18	104	9310131	1380013310	STOUFFERS HP CNTRY F	16 OZ	12
19	104	9310011	1380013311	STOUFFER'S HRTY PRTN	13.5 O	12
20	104	9309881	1380013312	STOUFFERS HEARTY POR	15.4 O	12
21	104	9043031	2113150434	MARIE CALLENDAR SPAG	17 OZ	8
22	104	9043081	2113150440	MARIE CALLENDER RAVI	16 OZ	8
23	104	9043041	2113150475	FETTUCINI W/BROCCOLI	13 OZ	8
24	104	9043001	2113150560	MARIE CALLENDER MEAT	14 OZ	8
25	104	9043051	2113150575	ESCALLOPED NOODLES &	13 OZ	8
26	104	9043061	2113150595	MARIE CALLENDER LASA	15 OZ	8
27	104	9043011	2113150605	MARIE CALLENDER COUN	16 OZ	8
28	104	9043091	2113150630	MARIE CALL HERB ROAS	14OZ	8

XI) UPCFRE.CSV

1	COM_CODE	UPC	DESCRIP	SIZE	CASE	NITEM
2	105	1242409401	WOLFGANG PUCK 4 CHS	13.5	8	9310050
3	105	1242409402	WOLFGANG PUCK MSH/SP	13.5	8	9310060
4	105	1242409403	WOLFGANG PUCK CHN RA	13.5 O	8	9310010
5	105	1242409404	WOLFGANG PUCK SWTPPO	13.5 O	8	9310020
6	105	1242409406	WOLFGANG PUCK MSH SP	15 OZ	8	9310090
7	105	1242409407	WOLFGANG PUCK EGPLNT	15 OZ	8	9310080
8	105	1242409408	WOLFGANG PUCK SP CKN	13.5 O	8	9310030
9	105	1380010002	STFRS BEEF CHOP SUEY	12 OZ	12	9129041
10	105	1380010011	~STOUFFER'S BEEF STR	9.7 OZ	12	9110430
11	105	1380010012	STOFR BEEF TERIYAKI	10 OZ	12	9111070
12	105	1380010017	STOFFERS CHILI W/BEA	8.75 O	12	9126951
13	105	1380010018	STFRS CRMD CHP BEEF	11 OZ	12	9124101
14	105	1380010019	STFRS IRZ CHILI REAL	9.2 OZ	12	912398
15	105	1380010021	~STFR LXP ORIENTAL B	9.625	12	9120041
16	105	1380010023	STOUFFERS CHDR PASTA	11 OZ	12	9124741
17	105	1380010025	STFRS R C BEEF POT R	9.25 O	12	9134181
18	105	1380010032	STFRS GREEN PEPPER S	10.5 O	12	9124551
19	105	1380010036	STFRS R C BEEF DIJON	9.5 OZ	12	9134201
20	105	1380010037	STFRS R C BEEF CACCI	10.1 O	12	9134051
21	105	1380010040	STFRS TORTILLA GRAND	9.6 OZ	12	9124431
22	105	1380010045	STFRS H/S RIGATONI W	9 OZ	12	9124261
23	105	1380010050	LC BEEF/BEAN ENCHANA	9.25OZ	12	9310761
24	105	1380010055	STFRS IRZ JAMBALAYA	10 OZ	12	912400
25	105	1380010060	STOFR SALISBURY STEA	9.78 O	12	9129420
26	105	1380010065	STOFR RIBS OF BEEF	9 OZ	12	9129550
27	105	1380010066	STOUFFERS SNGL SERVE	10 OZ	12	9124311
28	105	1380010067	STFRS STUFFED PEPPER	15.5 O	12	9123901

XII) UPCFRJ.CSV

1	COM_CODE	UPC	DESCRIP	SIZE	CASE	NITEM
2	111	1110000139	FLORIDA GOLD VALENCI	12 OZ	24	9005521
3	111	1110000390	FLORIDAGOLD OLD FASH	12 OZ	24	9006021
4	111	1110000391	FLORIDAGOLD PULP FRE	12 OZ	24	9005531
5	111	1110000550	~FLORIDA GOLD ORANGE	64 OZ	6	9005511
6	111	1450000124	DEAN FOODS RED RASPB	10 OZ	12	9002321
7	111	2080006311	SPEAS PARENTS CHOICE	12 OZ	12	9005321
8	111	2080006319	SPEAS PARENTS CHOICE	12 OZ	12	9005361
9	111	2080006326	SPEAS PARENTS CHOICE	12 OZ	12	9005341
10	111	2190896511	FARM APPLE JUICE 12	12 OZ	12	9310200
11	111	2190896512	FARM GRAPE JUICE 12	12 OZ	12	9310210
12	111	2400036600	HAWAIIAN PUNCH FRT J	12 OZ	24	9005651
13	111	2500000300	~MM ORANGE JUICE	6 OZ	24	9005301
14	111	2500000320	MM TANGERINE JCE	6 OZ	24	9900207
15	111	2500000425	5 ALIVE FRUIT BEVERA	12 OZ	24	9013031
16	111	2500000610	BACARDI RASP DAIQUIR	6 OZ	24	9002121
17	111	2500000611	BACARDI PINA COLADA	6 OZ	24	9012851
18	111	2500000612	BACARDI STRAW DAIQUI	6 OZ	24	9012871
19	111	2500000615	BACARDI MAGARITA	6 OZ	24	9012921
20	111	2500000616	BACARKI PEACH DAQUIR	6 OZ	24	9002110
21	111	2500000621	~BACARDI PINA COLADA	10 OZ	12	9012851
22	111	2500000622	BACARDI STRAW DAIQUI	10 OZ	12	9012871
23	111	2500000625	~BACARDI MARGARITA	10 OZ	12	9012891
24	111	2500000627	BACARDI RASP DAIQUIR	10 OZ	24	9002121
25	111	2500000632	BACARDI BANANA DAQ	10 OZ	24	9012881
26	111	2500000636	~BACARDI RUM RUNNER	10 OZ	12	9012881
27	111	2500002404	~ MM TROPICAL PUNCH	12 OZ	24	9012941
28	111	2500002408	~MINUTE MAID CRANBER	12 OZ	24	9012971

XIII) UPCGRO.CSV

1	COM_CODE	NITEM	UPC	DESCRIP	SIZE	CASE
2	972	6570051	707300856	BRUT 33 SPLSH LOT W/	7 OZ	6
3	969	6428441	1204400005	OLD SPICE PUMP ORIGI	1.75 O	6
4	969	6428451	1204400006	OLD SPICE PUMP FRESH	1.75 O	6
5	967	6412021	1204412490	LADY CHOICE SLD REG	1.5 OZ	6
6	967	6412031	1204412500	~LADY CHIOCE SLD UNS	1.5 OZ	6
7	967	6412041	1204412510	LADY CHOICE SLD FRES	1.5 OZ	6
8	967	6412021	1204412520	LADY CHOICE SLD REG	1.5 OZ	6
9	967	6412031	1204412530	LADY CHOICE SLD UNS	1.5 OZ	6
10	967	6412041	1204412540	LADY CHOICE SLD FRSH	1.5 OZ	6
11	967	6412001	1204412630	~LADY CHOICE SLD MEA	1.5 OZ	6
12	967	6412011	1204412640	~LADYS CHCE SLD SPOR	1.5 OZ	6
13	968	6412071	1204412710	LADY CHCE R/O UNSC #	1.5 OZ	6
14	968	6412061	1204412720	LADY'S CHOICE R/O FR	1.5 OZ	6
15	967	6412001	1204412800	LADYS CHOICE MEAD BR	1.5 OZ	6
16	967	6412011	1204412810	~LDY CH SLD MUSK 30[	1.5 OZ	6
17	972	6571331	1204430225	OLD SPICE SOOTHING G	3.25 O	6
18	972	6571341	1204430226	OLD SPICE SOOTHING G	3.25 O	6
19	972	6304681	1204430620	OS COL/HI END DEOD G	1 CT	6
20	972	6304671	1204430630	OS ASL/COLOGNE GIFT	1 CT	6
21	972	6304691	1204430650	OS ASL/SHV CRM/STK O	1 CT	6
22	968	6428421	1204432110	OLD SPI DEOD LIQ MUS	1.5 OZ	6
23	968	6428411	1204432120	OLD SPI LIQ STK FRSH	1.5 OZ	6
24	972	6301821	1204433670	OLD SPICE FRAG GIFT	1 CT	18
25	972	6301821	1204433920	OLD SPICE FRAG GIFT	1 CT	18
26	972	6301821	1204433930	OLD SPICE FRAG GIFT	1 CT	18
27	967	6428461	1204434210	OLD SPICE HIGH ENDVR	3.25 O	6
28	967	6428511	1204434220	OLD SPICE HIGH ENDVR	3.25 O	6

XIV) WEEK.CSV

1	Week #	Start	End	Special Events
2	1	9/14/1989	9/20/1989	
3	2	9/21/1989	9/27/1989	
4	3	9/28/1989	10/4/1989	
5	4	10/5/1989	10/11/1989	
6	5	10/12/1989	10/18/1989	
7	6	10/19/1989	10/25/1989	
8	7	10/26/1989	11/1/1989	Halloween
9	8	11/2/1989	11/8/1989	
10	9	11/9/1989	11/15/1989	
11	10	11/16/1989	11/22/1989	
12	11	11/23/1989	11/29/1989	Thanksgiving
13	12	11/30/1989	12/6/1989	
14	13	12/7/1989	12/13/1989	
15	14	12/14/1989	12/20/1989	
16	15	12/21/1989	12/27/1989	Christmas
17	16	12/28/1989	1/3/1990	New-Year
18	17	1/4/1990	1/10/1990	
19	18	1/11/1990	1/17/1990	
20	19	1/18/1990	1/24/1990	
21	20	1/25/1990	1/31/1990	
22	21	2/1/1990	2/7/1990	
23	22	2/8/1990	2/14/1990	
24	23	2/15/1990	2/21/1990	Presidents Day
25	24	2/22/1990	2/28/1990	
26	25	3/1/1990	3/7/1990	
27	26	3/8/1990	3/14/1990	
28	27	3/15/1990	3/21/1990	
29	28	3/22/1990	3/28/1990	Easter

XV)STORE.CSV

Store	City	Price Tier	Zone	Zip Code	Address
2	River Forest	High	1	60305	7501 W. North Ave.
4	Park Ridge	Medium	2	60068	Closed
5	Palatine	Medium	2	60067	223 Northwest HWY.
8	Oak Lawn	Low	5	60435	8700 S. Cicero Ave.
9	Morton Grove	Medium	2	60053	6931 Dempster
12	Chicago	High	7	60660	6009 N. Broadway Ave.
14	Glenview	High	1	60025	1020 Waukegan Rd.
18	River Grove	Low	5	60171	8355 W. Belmont Ave.
19	Glen Ellyn			60137	Closed
21	Hanover Park	CubFighter	6	60103	1440 Irving Park Rd.
25	Chicago			60639	Closed
28	Mt. Prospect	Medium	2	60054	1145-55 Mt Prospect Pz.
32	Park Ridge	High	1	60068	1900 S. Cumberland Ave.
33	Chicago	High	7	60657	3012 N. Broadway Ave.
39	Waukegan			60085	Closed
40	Bridgeview	CubFighter	6	60455	8825 S. Harlem Ave.
44	Western Spring	Medium	2	60558	14 Garden Market St.
45	Wheeling	Medium	2	60090	550 W. Dundee Rd.
46	Carol Stream	Low	5	60187	Closed
47	Addison	Medium	2	60101	545 W. Lake St.
48	Schaumburg	Medium	2	60193	20 E. Golf Rd.
49	Downers Grove	Medium	2	60515	120 E. Ogden Ave.

10.3. Mapping Tables for Staging and Data Mart Loads

(a) Mapping of data from .csv data sources to staging area:

- Dominick_Store.csv to STORE_STAGING:

Dominick_Store.csv to STORE_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
Dominick_Store.csv	Store	STORE_STAGING	Store	int
	Price Tier		Price_Tier	varchar (30)
	Zip Code		Zip_Code	int
	Zone		Zone	int
	City		City	varchar (30)

- Dominick_WeekDecode.csv to TIME_STAGING:

Dominick_Week_Decode.csv to TIME_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
Dominick_WEEK_Decode.csv	Week#	WEEK_STAGING	Week#	int
	Start		Start	int
	End		End	int
	Special Events		Special Events	int

- Demo.csv to DEMO_STAGING:

Demo.csv to DEMO_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
Demo.csv	MMID	DEMO_STAGING	MMID	varchar (30)
	Name		Name	varchar (30)
	City		City	varchar (30)
	Zip		Zip	varchar (30)
	Store		Store	varchar (30)

	Zone		Zone	int
	Age9		Age9	float
	Age60		Age60	float

- wcer.csv to WCER_STAGING:

wcer.csv to WCER_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
wcer.csv	Store	WCER_STAGING	Store	varchar (30)
	UPC		UPC	varchar (30)
	Week		Week	varchar (30)
	Move		Move	int
	Qty		Qty	int
	Price		Price	float
	Sale		Sale	float
	Profit		Profit	float

- wche.csv to WCHE_STAGING:

wche.csv to WCHE_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
wche.csv	Store	WCHE_STAGING	Store	varchar (30)
	UPC		UPC	varchar (30)
	Week		Week	varchar (30)
	Move		Move	int
	Qty		Qty	int
	Price		Price	float
	Sale		Sale	float
	Profit		Profit	float

- wfrd.csv to WFRD_STAGING:

wfrd.csv to WFRD_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
wfrd.csv	Store	WFRD_STAGING	Store	varchar (30)
	UPC		UPC	varchar (30)
	Week		Week	varchar (30)

	Move		Move	int
	Qty		Qty	int
	Price		Price	float
	Sale		Sale	float
	Profit		Profit	float

- wfre.csv to WFRE_STAGING:

wfre.csv to WFRE_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
wfre.csv	Store	WFRE_STAGING	Store	varchar (30)
	UPC		UPC	varchar (30)
	Week		Week	varchar (30)
	Move		Move	int
	Qty		Qty	int
	Price		Price	float
	Sale		Sale	float
	Profit		Profit	float

- wfrj.csv to WFRJ_STAGING:

wfrj.csv to WFRJ_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
wfrj.csv	Store	WFRJ_STAGING	Store	varchar (30)
	UPC		UPC	varchar (30)
	Week		Week	varchar (30)
	Move		Move	Int
	Qty		Qty	Int
	Price		Price	float
	Sale		Sale	float
	Profit		Profit	float

- wgro.csv to WGRO_STAGING:

wgro.csv to WGRO_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
wgro.csv	Store	WGRO_STAGING	Store	varchar (30)

	UPC		UPC	varchar (30)
	Week		Week	varchar (30)
	Move		Move	Int
	Qty		Qty	Int
	Price		Price	float
	Sale		Sale	float
	Profit		Profit	float

- upccer.csv to UPCCER_STAGING:

upccer.csv to UPCCER_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
upccer.csv	COM_CODE	UPCCER_STAGING	COM_CODE	float
	UPC		UPC	float
	DESCRIP		DESCRIP	varchar (30)
	SIZE		SIZE	varchar (30)
	CASE		CASE	float
	NITEM		NITEM	float

- upcche.csv to UPCCHE_STAGING:

upcche.csv to UPCCHE_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
upcche.csv	COM_CODE	UPCCHE_STAGING	COM_CODE	float
	UPC		UPC	float
	DESCRIP		DESCRIP	varchar (30)
	SIZE		SIZE	varchar (30)
	CASE		CASE	float
	NITEM		NITEM	float

- upcfrd.csv to UPCFRD_STAGING:

upcfrd.csv to UPCFRD_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
upcfrd.csv	COM_CODE	UPCFRD_STAGING	COM_CODE	float
	UPC		UPC	float
	DESCRIP		DESCRIP	varchar (30)

	SIZE		SIZE	varchar (30)
	CASE		CASE	float
	NITEM		NITEM	float

- upcfre.csv to UPCFRE_STAGING:

upcfre.csv to UPCFRE_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
upcfre.csv	COM_CODE	UPCFRE_STAGING	COM_CODE	float
	UPC		UPC	float
	DESCRIP		DESCRIP	varchar (30)
	SIZE		SIZE	varchar (30)
	CASE		CASE	float
	NITEM		NITEM	float

- upcfrj.csv to UPCFRJ_STAGING:

upcfrj.csv to UPCFRJ_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
upcfrj.csv	COM_CODE	UPCFRJ_STAGING	COM_CODE	float
	UPC		UPC	float
	DESCRIP		DESCRIP	varchar (30)
	SIZE		SIZE	varchar (30)
	CASE		CASE	float
	NITEM		NITEM	float

- upcgro.csv to UPCGRO_STAGING:

upcgro.csv to UPCGRO_STAGING				
SOURCE		STAGING		
Source Table	Source Column	Staging Table	Staging Column	Staging Datatype
upcgro.csv	COM_CODE	UPCGRO_STAGING	COM_CODE	float
	UPC		UPC	float
	DESCRIP		DESCRIP	varchar (30)
	SIZE		SIZE	varchar (30)
	CASE		CASE	float
	NITEM		NITEM	float

(b) Mapping of data from Staging area to Data Marts:

- STORE_DIM_STAGING to STORE_DIM:

STORE_DIM_STAGING to STORE_DIM					
STAGING			DESTINATION		
Staging Table	Staging Column	Staging Datatype	Target Table	Target Column	Target Datatype
STORE_DIM_STAGING	STOREKEY	int	STORE_DIM	STOREKEY	int
	STORE_NUM	varchar (30)		STORE_NUM	int
	PRICE_TIER	varchar (30)		PRICE_TIER	varchar (30)
	ZIP_CODE	varchar (30)		ZIP_CODE	varchar (30)
	STORE_ZONE	varchar (30)		STORE_ZONE	varchar (30)
	CITY	varchar (30)		CITY	varchar (30)

- PRODUCT_DIM_STAGING to PRODUCT_DIM:

PRODUCT_DIM_STAGING to PRODUCT_DIM					
STAGING			DESTINATION		
Staging Table	Staging Column	Target Datatype	Target Table	Target Column	Target Datatype
PRODUCT_DIM_STAGING	PRODUCTKEY	int	PRODUCT_DIM	PRODUCTKEY	int
	UPC_NUMBER	float		UPC_NUMBER	float
	PRODUCT_NAME	varchar (30)		PRODUCT_NAME	varchar (30)

- TIME_DIM_STAGING to TIME_DIM:

STORE_STAGING to TIME_DIM					
STAGING			DESTINATION		
Staging Table	Staging Column	Staging Datatype	Target Table	Target Column	Target Datatype
TIME_DIM_STAGING	TIMEKEY	int	TIME_DIM	TIMEKEY	int
	Week	int		Week	int
	Month	int		Month	int

	Quarter	int		Quarter	int
	Year	int		Year	int

- SALES_FACT_STAGING to SALES_FACT:

SALES_FACT_STAGING to SALES_FACT					
STAGING			DESTINATION		
Staging Table	Staging Column	Staging Datatype	Target Table	Target Column	Target Datatype
SALES_FACT_STAGING	STOREKEY	int	SALES_FACT	STOREKEY	float
	PRODUCTKEY	int		PRODUCTKEY	int
	TIMEKEY	int		TIMEKEY	int
	SALES	float		SALES	float

- STORE_PERF_FACT_STAGING to STORE_PERF_FACT:

STORE_PERF_FACT_STAGING to STORE_PERF_FACT					
STAGING			DESTINATION		
Staging Table	Staging Column	Staging Datatype	Target Table	Target Column	Target Datatype
STORE_PERF_FACT_STAGING	STOREKEY	int	STORE_PERF_FACT	STOREKEY	int
	PRODUCTKEY	int		PRODUCTKEY	int
	PROFIT	float		PROFIT	float
	BELOW_9%	float		BELOW_9%	float
	ABOVE_60%	float		ABOVE_60%	float

10.4. Data Quality Issues in DFF's Data Sets

already mentioned, source data included various CSV files and some data from the DFF's data manual. The extraction rules which we have used for each are described below:

The Dominick's

The data quality issues associated with DFF's data sets are:

1. Duplicates in the data: We observed that in the UPC file like UPCCER, UPCCHE, UPCFRD, UPCFRE, UPCFRJ, UPCGRO and so on there was some redundant data. These redundant columns contained data with same UPC numbers and Product Descriptions.
2. Cryptic Values: We observed that in the CCOUNT file, there were many values which were cryptic like ".". We had to delete these values to get a better view and understanding of the data.
3. Absence of data values: There were many files in which important values for profits and even sales were missing. Many rows had null values for store numbers. We had to remove those values to analyze the data properly.

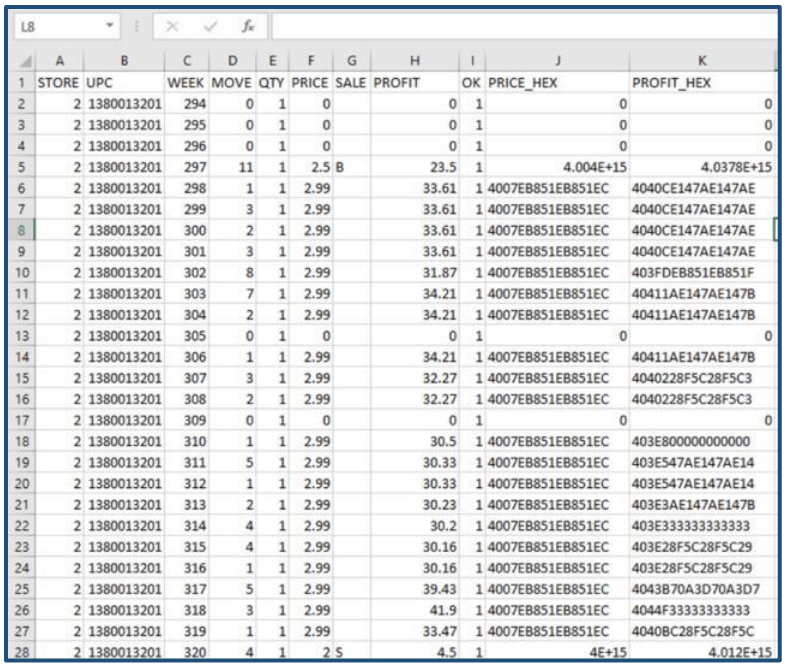
By identifying these data quality issues in our dataset, we formulated data extraction strategy which has been discussed in sections below.

10.5. Data Extraction Rules

As we have already mentioned, source data included various CSV files and some data from the DFF's data manual. The extraction rules which we have used for each are described below:

The Dominick's data set had data in two categories – category specific files and general files. The general files had customer count data and store level demographics data while the category files had UPC and Movement data for about 29 categories. Based on our business questions for the project, we identified the required files for analysis and loaded only those in the warehouse.

We have made use of 6 Movement files namely - WFRD, WFRE, WFRJ, WGRO, WCHE, WCER which is acting as a source for Sales fact table and Store_Performance fact table as well. We extracted all the required Movement files and then combined them by taking a union of all the files as per our requirement for the project. We are using the union of these files to get Sales data for the Sales fact table and to get Profit data for the Store_Performance fact table.



	A	B	C	D	E	F	G	H	I	J	K
1	STORE	UPC	WEEK	MOVE	QTY	PRICE	SALE	PROFIT	OK	PRICE_HEX	PROFIT_HEX
2	2	1380013201	294	0	1	0		0	1	0	0
3	2	1380013201	295	0	1	0		0	1	0	0
4	2	1380013201	296	0	1	0		0	1	0	0
5	2	1380013201	297	11	1	2.5	B	23.5	1	4.004E+15	4.0378E+15
6	2	1380013201	298	1	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
7	2	1380013201	299	3	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
8	2	1380013201	300	2	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
9	2	1380013201	301	3	1	2.99		33.61	1	4007EB851EB851EC	4040CE147AE147AE
10	2	1380013201	302	8	1	2.99		31.87	1	4007EB851EB851EC	403FDEB851EB851F
11	2	1380013201	303	7	1	2.99		34.21	1	4007EB851EB851EC	40411AE147AE147B
12	2	1380013201	304	2	1	2.99		34.21	1	4007EB851EB851EC	40411AE147AE147B
13	2	1380013201	305	0	1	0		0	1	0	0
14	2	1380013201	306	1	1	2.99		34.21	1	4007EB851EB851EC	40411AE147AE147B
15	2	1380013201	307	3	1	2.99		32.27	1	4007EB851EB851EC	4040228F5C28F5C3
16	2	1380013201	308	2	1	2.99		32.27	1	4007EB851EB851EC	4040228F5C28F5C3
17	2	1380013201	309	0	1	0		0	1	0	0
18	2	1380013201	310	1	1	2.99		30.5	1	4007EB851EB851EC	403E800000000000
19	2	1380013201	311	5	1	2.99		30.33	1	4007EB851EB851EC	403E547AE147AE14
20	2	1380013201	312	1	1	2.99		30.33	1	4007EB851EB851EC	403E547AE147AE14
21	2	1380013201	313	2	1	2.99		30.23	1	4007EB851EB851EC	403E3AE147AE147B
22	2	1380013201	314	4	1	2.99		30.2	1	4007EB851EB851EC	403E333333333333
23	2	1380013201	315	4	1	2.99		30.16	1	4007EB851EB851EC	403E28F5C28F5C29
24	2	1380013201	316	1	1	2.99		30.16	1	4007EB851EB851EC	403E28F5C28F5C29
25	2	1380013201	317	5	1	2.99		39.43	1	4007EB851EB851EC	4043B70A3D70A3D7
26	2	1380013201	318	3	1	2.99		41.9	1	4007EB851EB851EC	4044F33333333333
27	2	1380013201	319	1	1	2.99		33.47	1	4007EB851EB851EC	4040BC28F5C28F5C
28	2	1380013201	320	4	1	2	S	4.5	1	4E+15	4.012E+15

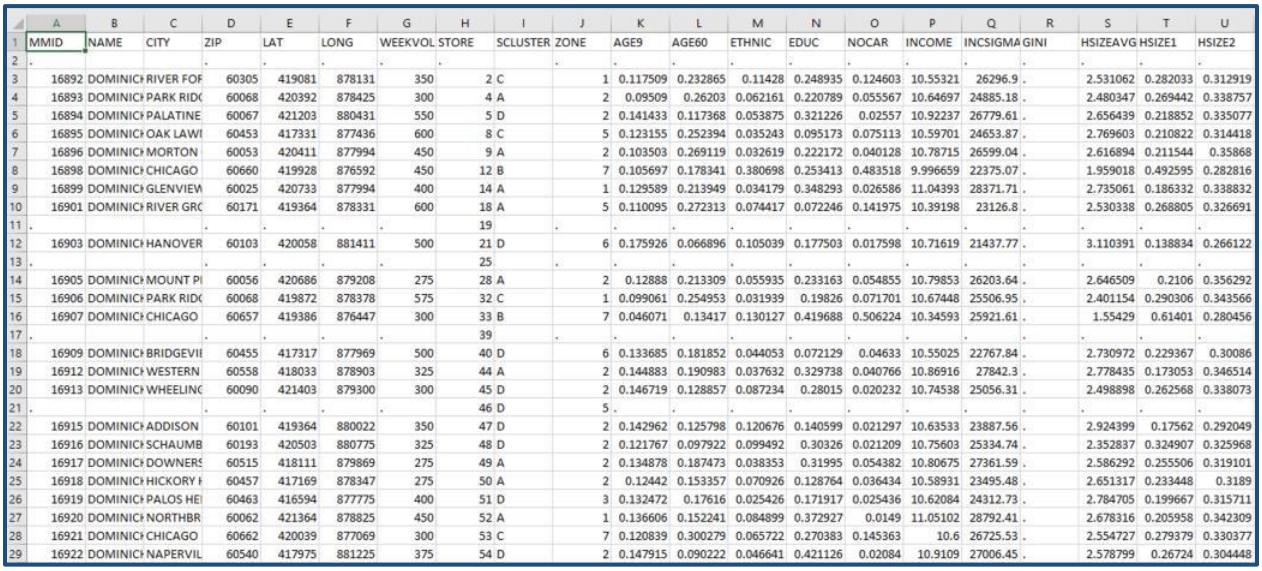
Example of Movement File before Extraction

Next, we made use of 6 UPC files namely - UPCCER, UPCCHE, UPCFRD, UPCFRE, UPCFRJ, UPCGRO which is acting as a source for Product Dimension Table. We extracted all the files and then combined them using Union to get relevant data for our purposes.

1	COM_CODE	UPC	DESCRIP	SIZE	CASE	NITEM
2	311	317	$TONY THE TIGER T-SH	ASST	1	9900090
3	311	1313000002	NAB SHREDDED WHEAT	10 OZ	12	2513101
4	311	1313000005	NABISCO SHREDDED WHE	15 OZ	12	2513120
5	311	1313000032	NABISCO 100% BRAN	17 OZ	12	2513951
6	311	1313000050	NAB SPOON SIZE SHRED	23.45O	12	2516761
7	311	1313000054	NAB SPOON SIZE SHRED	17.2 O	12	2513051
8	311	1313000148	NABISCO TEAM FLAKES	13 OZ	12	2513600
9	311	1313000698	NABISCO CREAM OF RIC	14 OZ	12	251690
10	311	1313001041	NABISCO OAT BRAN RAI	15.5 O	12	2513551
11	311	1313001059	NABISCO WHEAT N BRAN	18 OZ	12	2513131
12	311	1313001145	NABISCO FRUIT WHEATS	15.5 O	12	2513500
13	311	1313001146	NAB RASPBERRY FRUIT	15.5 O	12	251352
14	311	1313001854	NAB SHRED WHT W/OAT	18 OZ	12	2513121
15	311	1313002033	NABISCO 100% BRAN W/	17 OZ	12	2513900
16	311	1313002401	TEDDY GRAHAMS BRKFST	15 OZ	12	2513201
17	311	1313002402	TEDDY GRAHAMS BRKFST	15 OZ	12	2513251
18	311	1313002403	TEDDY GRAHAMS BRKFST	15 OZ	12	2513351
19	311	1313005152	NAB FROST WHEAT SQUA	15 OZ	12	2513161
20	311	1313005154	NAB FROSTED WHEAT	20 OZ	12	2508120
21	311	1313018155	~NABISCO FROSTED WHE	15 OZ	14	2513121
22	311	1600000000	BERRY KIX/WHT TOTAL/	1 CT	1	9825091
23	311	1600062360	TRIPLES/KRAFT MARSHM	18.7 O	21	2508221
24	311	1600062370	$ GEN MILLS TRIPLES	13.3 O	16	2508201
25	311	1600062380	G. MILLS TRIPLES	18.75O	14	2508221
26	311	1600062580	G. MILLS BENEFIT W/R	15 OZ	12	250440
27	311	1600062600	G. MILLS BENEFIT CER	12.5 O	12	250435
28	311	1600062640	G.M. BERRY BERRY KIX	18 OZ	14	2503600

Example of UPC File before Extraction

The DEMO.csv has been used as a source for Store_Performance fact table. During data extraction from Demographics.csv file we deleted the data where the store numbers were invalid from the Demographics table. Also, we had many columns in this file, we dropped some of the columns during extraction and loaded only those columns which were required for analysis.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	MMID	NAME	CITY	ZIP	LAT	LONG	WEEKVOL	STORE	SCLUSTER	ZONE	AGE9	AGE60	ETHNIC	EDUC	NOCAR	INCOME	INCSIGMA	GINI	HSIZEAVG	HSIZE1	HSIZE2
2	.			.	.	.	.	.		.	.	.	.	.	.	.	.	.	.	.	.
3	16892	DOMINICK	RIVER FOF	60305	419081	878131	350	2	C	1	0.117509	0.232865	0.11428	0.248935	0.124603	10.55321	26296.9	.	2.531062	0.282033	0.312919
4	16893	DOMINICK	PARK RIDG	60068	420392	878425	300	4	A	2	0.09509	0.26203	0.062161	0.220789	0.055567	10.64697	24885.18	.	2.480347	0.269442	0.338757
5	16894	DOMINICK	PALATINE	60067	421203	880431	550	5	D	2	0.141433	0.117368	0.053875	0.321226	0.02557	10.92237	26779.61	.	2.656439	0.218852	0.335077
6	16895	DOMINICK	OAK LAWN	60453	417331	877436	600	8	C	5	0.123155	0.252394	0.035243	0.095173	0.075113	10.59701	24653.87	.	2.769603	0.210822	0.314418
7	16896	DOMINICK	MORTON	60053	420411	877994	450	9	A	2	0.103503	0.269119	0.032619	0.222172	0.040128	10.78715	26599.04	.	2.616894	0.211544	0.35868
8	16898	DOMINICK	CHICAGO	60660	419928	876592	450	12	B	7	0.105697	0.178341	0.380698	0.253413	0.483518	9.996659	22375.07	.	1.959018	0.492595	0.282816
9	16899	DOMINICK	GLENVIEW	60025	420733	877994	400	14	A	1	0.129589	0.213949	0.034179	0.348293	0.026586	11.04393	28371.71	.	2.735061	0.186332	0.338832
10	16901	DOMINICK	RIVER GRO	60171	419364	878331	600	18	A	5	0.110095	0.272313	0.074417	0.072246	0.141975	10.39198	23126.8	.	2.530338	0.268805	0.326691
11	.			.	.	.	.	19		.	.	.	.	.	.	.	.	.	.	.	.
12	16903	DOMINICK	HANOVER	60103	420058	881411	500	21	D	6	0.175926	0.066896	0.105039	0.177503	0.017598	10.71619	21437.77	.	3.110391	0.138834	0.266122
13	.			.	.	.	.	25		.	.	.	.	.	.	.	.	.	.	.	.
14	16905	DOMINICK	MOUNT PI	60056	420686	879208	275	28	A	2	0.12888	0.213309	0.055935	0.233163	0.054855	10.79853	26203.64	.	2.646509	0.2106	0.356292
15	16906	DOMINICK	PARK RIDG	60068	419872	878378	575	32	C	1	0.099061	0.254953	0.031939	0.19826	0.071701	10.67448	25506.95	.	2.401154	0.290306	0.343566
16	16907	DOMINICK	CHICAGO	60657	419386	876447	300	33	B	7	0.046071	0.13417	0.130127	0.419688	0.506224	10.34593	25921.61	.	1.55429	0.61401	0.280456
17	.			.	.	.	.	39		.	.	.	.	.	.	.	.	.	.	.	.
18	16909	DOMINICK	BRIDGEVIE	60455	417317	877969	500	40	D	6	0.133685	0.181852	0.044053	0.072129	0.04633	10.55025	22767.84	.	2.730972	0.229367	0.30086
19	16912	DOMINICK	WESTERN	60558	418033	878903	325	44	A	2	0.144883	0.190983	0.037632	0.329738	0.040766	10.86916	27842.3	.	2.778435	0.173053	0.346514
20	16913	DOMINICK	WHEELING	60090	421403	879300	300	45	D	2	0.146719	0.128857	0.087234	0.28015	0.020232	10.74538	25056.31	.	2.498898	0.262568	0.338073
21	.			.	.	.	.	46	D	5	.	.	.	.	.	.	.	.	.	.	.
22	16915	DOMINICK	ADDISON	60101	419364	880022	350	47	D	2	0.142962	0.125798	0.120676	0.140599	0.021297	10.63533	23887.56	.	2.924399	0.17562	0.292049
23	16916	DOMINICK	SCHAUMB	60193	420503	880775	325	48	D	2	0.121767	0.097922	0.099492	0.30326	0.021209	10.75603	25334.74	.	2.352837	0.324907	0.325968
24	16917	DOMINICK	DOWNERS	60515	418111	879869	275	49	A	2	0.134878	0.187473	0.038353	0.31995	0.054382	10.80675	27361.59	.	2.586292	0.255506	0.319101
25	16918	DOMINICK	HICKORY H	60457	417169	878347	275	50	A	2	0.12442	0.153357	0.070926	0.128764	0.036434	10.58931	23495.48	.	2.651317	0.233448	0.3189
26	16919	DOMINICK	PALOS HE	60463	416594	877775	400	51	D	3	0.132472	0.17616	0.025426	0.171917	0.025436	10.62084	24312.73	.	2.784705	0.199667	0.315711
27	16920	DOMINICK	NORTHBR	60062	421364	878825	450	52	A	1	0.136606	0.152241	0.084899	0.372927	0.0149	11.05102	28792.41	.	2.678316	0.205958	0.342309
28	16921	DOMINICK	CHICAGO	60662	420039	877069	300	53	C	7	0.120839	0.300279	0.065722	0.270383	0.145363	10.6	26725.53	.	2.554727	0.279379	0.330377
29	16922	DOMINICK	NAPERVIL	60540	417975	881225	375	54	D	2	0.147915	0.090222	0.046641	0.421126	0.02084	10.9109	27006.45	.	2.578799	0.26724	0.304448

Example of Demo File before Extraction

Additionally, we used the DFF's data manual to extract Dominick's Stores table (Part 6) and Weeks' decode table (Part 8) as a source for Store dimension table and Time dimension table respectively. We built CSV files from the lab manual and then extracted them in our staging area.

Store	City	Price Tier	Zone	Zip Code	Address
2	River Forest	High	1	60305	7501 W. North Ave.
4	Park Ridge	Medium	2	60068	Closed
5	Palatine	Medium	2	60067	223 Northwest HWY.
8	Oak Lawn	Low	5	60435	8700 S. Cicero Ave.
9	Morton Grove	Medium	2	60053	6931 Dempster
12	Chicago	High	7	60660	6009 N. Broadway Ave.
14	Glenview	High	1	60025	1020 Waukegan Rd.
18	River Grove	Low	5	60171	8355 W. Belmont Ave.
19	Glen Ellyn			60137	Closed
21	Hanover Park	CubFighter	6	60103	1440 Irving Park Rd.
25	Chicago			60639	Closed
28	Mt. Prospect	Medium	2	60054	1145-55 Mt Prospect Pz.
32	Park Ridge	High	1	60068	1900 S. Cumberland Ave.
33	Chicago	High	7	60657	3012 N. Broadway Ave.
39	Waukegan			60085	Closed
40	Bridgeview	CubFighter	6	60455	8825 S. Harlem Ave.
44	Western Spring	Medium	2	60558	14 Garden Market St.
45	Wheeling	Medium	2	60090	550 W. Dundee Rd.
46	Carol Stream	Low	5	60187	Closed
47	Addison	Medium	2	60101	545 W. Lake St.
48	Schaumburg	Medium	2	60193	20 E. Golf Rd.
49	Downers Grove	Medium	2	60515	120 E. Ogden Ave.

Example of Store File before Extraction

1	Week #	Start	End	Special Events
2	1	9/14/1989	9/20/1989	
3	2	9/21/1989	9/27/1989	
4	3	9/28/1989	10/4/1989	
5	4	10/5/1989	10/11/1989	
6	5	10/12/1989	10/18/1989	
7	6	10/19/1989	10/25/1989	
8	7	10/26/1989	11/1/1989	Halloween
9	8	11/2/1989	11/8/1989	
10	9	11/9/1989	11/15/1989	
11	10	11/16/1989	11/22/1989	
12	11	11/23/1989	11/29/1989	Thanksgiving
13	12	11/30/1989	12/6/1989	
14	13	12/7/1989	12/13/1989	
15	14	12/14/1989	12/20/1989	
16	15	12/21/1989	12/27/1989	Christmas
17	16	12/28/1989	1/3/1990	New-Year
18	17	1/4/1990	1/10/1990	
19	18	1/11/1990	1/17/1990	
20	19	1/18/1990	1/24/1990	
21	20	1/25/1990	1/31/1990	
22	21	2/1/1990	2/7/1990	
23	22	2/8/1990	2/14/1990	
24	23	2/15/1990	2/21/1990	Presidents Day
25	24	2/22/1990	2/28/1990	
26	25	3/1/1990	3/7/1990	
27	26	3/8/1990	3/14/1990	
28	27	3/15/1990	3/21/1990	
29	28	3/22/1990	3/28/1990	Easter

Example of Week File before Extraction

10.6. Data Transformation and Cleansing Rules

(a) Calculating Sales Value in Movement Table:

We needed Sales value in the Sales fact table and for that, we created a new column called SALES in the MOVEMENT_STAGING table. In this column we calculated the Sales value using the following formula:

[SALES] = (PRICE * [MOVE])/QTY

For this, we made use of the following SQL Queries:

```
ALTER TABLE [Group10_602-staging-area].[dbo].[MOVEMENT_STAGING] ADD SALES
float;
UPDATE [Group10_602-staging-area].[dbo].[MOVEMENT_STAGING] SET [SALES] =
(PRICE * [MOVE])/QTY;
```

Results | Messages

	STORE	UPC	WEEK	MOVE	QTY	PRICE	PROFIT	SALES	PROFIT_VALUE
149693	9	2500000622	299	10	1	1.79	31.06	17.9	5.55974
149694	9	2500000622	300	12	1	1.79	31.06	21.48	6.671688
149695	9	2500000622	301	17	1	1.79	31.06	30.43	9.451558
149696	9	2500000622	302	14	1	1.79	31.06	25.06	7.783636
149697	9	2500000622	303	16	1	1.79	31.06	28.64	8.895584
149698	9	2500000622	304	12	1	1.79	31.06	21.48	6.671688
149699	9	2500000622	305	12	1	1.79	31.06	21.48	6.671688
149700	9	2500000622	306	16	1	1.79	31.06	28.64	8.895584
149701	9	2500000622	307	18	1	1.79	31.06	32.22	10.007532
149702	9	2500000622	308	5	1	1.79	31.06	8.95	2.77987
149703	9	2500000622	309	16	1	1.79	31.06	28.64	8.895584
149704	9	2500000622	310	15	1	1.79	31.06	26.85	8.33961
149705	9	2500000622	311	8	1	1.79	31.06	14.32	4.447792
149706	9	2500000622	312	5	1	1.79	31.06	8.95	2.77987
149707	9	2500000622	313	3	1	1.79	31.06	5.37	1.667922
149708	9	2500000622	314	3	1	1.79	31.06	5.37	1.667922
149709	9	2500000622	315	1	1	1.79	31.06	1.79	0.555974
149710	9	2500000622	316	1	1	1.79	31.06	1.79	0.555974
149711	9	2500000622	317	4	1	1.79	31.06	7.16	2.223896
149712	9	2500000622	318	3	1	1.79	31.06	5.37	1.667922
149713	9	2500000622	319	1	1	1.79	31.06	1.79	0.555974
149714	9	2500000622	320	7	1	0.99	-24.64	6.93	-1.707552

Movement Table after Transformation and Cleaning

(b) Calculating Profit Value in Movement Table:

We needed Profit values in the Store_Performance fact table and for that, we created a new column called PROFIT_VALUE in the MOVEMENT_STAGING table. In this column, we calculated the Profit value using the following formula:

[PROFIT_VALUE] = [PROFIT] * [SALES] /100

For this, we made use of the following SQL Queries:

```
ALTER TABLE [Group10_602-staging-area].[dbo].[MOVEMENT_STAGING] ADD
PROFIT_VALUE float;
UPDATE [Group10_602-staging-area].[dbo].[MOVEMENT_STAGING] SET
[PROFIT_VALUE] = [PROFIT] * [SALES] /100;
```

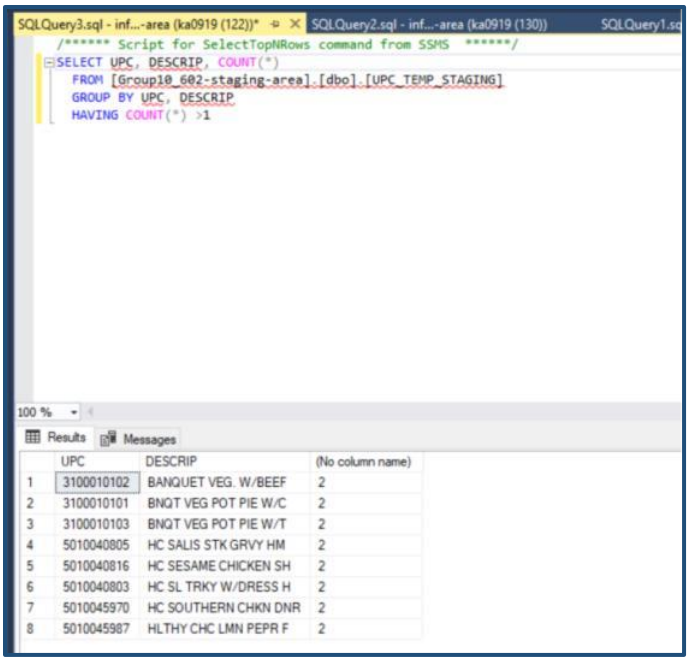
Results | Messages

	STORE	UPC	WEEK	MOVE	QTY	PRICE	PROFIT	SALES	PROFIT_VALUE
149693	9	2500000622	299	10	1	1.79	31.06	17.9	5.55974
149694	9	2500000622	300	12	1	1.79	31.06	21.48	6.671688
149695	9	2500000622	301	17	1	1.79	31.06	30.43	9.451558
149696	9	2500000622	302	14	1	1.79	31.06	25.06	7.783636
149697	9	2500000622	303	16	1	1.79	31.06	28.64	8.895584
149698	9	2500000622	304	12	1	1.79	31.06	21.48	6.671688
149699	9	2500000622	305	12	1	1.79	31.06	21.48	6.671688
149700	9	2500000622	306	16	1	1.79	31.06	28.64	8.895584
149701	9	2500000622	307	18	1	1.79	31.06	32.22	10.007532
149702	9	2500000622	308	5	1	1.79	31.06	8.95	2.77987
149703	9	2500000622	309	16	1	1.79	31.06	28.64	8.895584
149704	9	2500000622	310	15	1	1.79	31.06	26.85	8.33961
149705	9	2500000622	311	8	1	1.79	31.06	14.32	4.447792
149706	9	2500000622	312	5	1	1.79	31.06	8.95	2.77987
149707	9	2500000622	313	3	1	1.79	31.06	5.37	1.667922
149708	9	2500000622	314	3	1	1.79	31.06	5.37	1.667922
149709	9	2500000622	315	1	1	1.79	31.06	1.79	0.555974
149710	9	2500000622	316	1	1	1.79	31.06	1.79	0.555974
149711	9	2500000622	317	4	1	1.79	31.06	7.16	2.223896
149712	9	2500000622	318	3	1	1.79	31.06	5.37	1.667922
149713	9	2500000622	319	1	1	1.79	31.06	1.79	0.555974
149714	9	2500000622	320	7	1	0.99	-24.64	6.93	-1.707552

Movement Table after Transformation and Cleaning

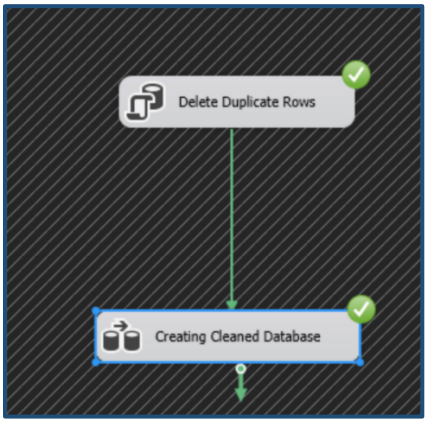
(c) Deleting Duplicates from UPC Table:

We found that there was some redundant data in the UPC_STAGING table which unnecessarily increased the size of the database. These redundant columns contained data with same UPC numbers and Product Descriptions. We deleted those rows using Execute SQL task in which we made use of the following SQL Query:



	UPC	DESCRIP	(No column name)
1	3100010102	BANQUET VEG. W/BEEF	2
2	3100010101	BNQT VEG POT PIE W/C	2
3	3100010103	BNQT VEG POT PIE W/T	2
4	5010040805	HC SALIS STK GRVY HM	2
5	5010040816	HC SESAME CHICKEN SH	2
6	5010040803	HC SL TRKY W/DRESS H	2
7	5010045970	HC SOUTHERN CHKN DNR	2
8	5010045987	HLTHY CHC LMN PEPR F	2

```
WITH a as
(
SELECT row_number() over (partition by UPC order by UPC) rn
FROM [Group10_602-staging-area].[dbo].[UPC_TEMP_STAGING]
)
delete from a
WHERE rn = 2
```



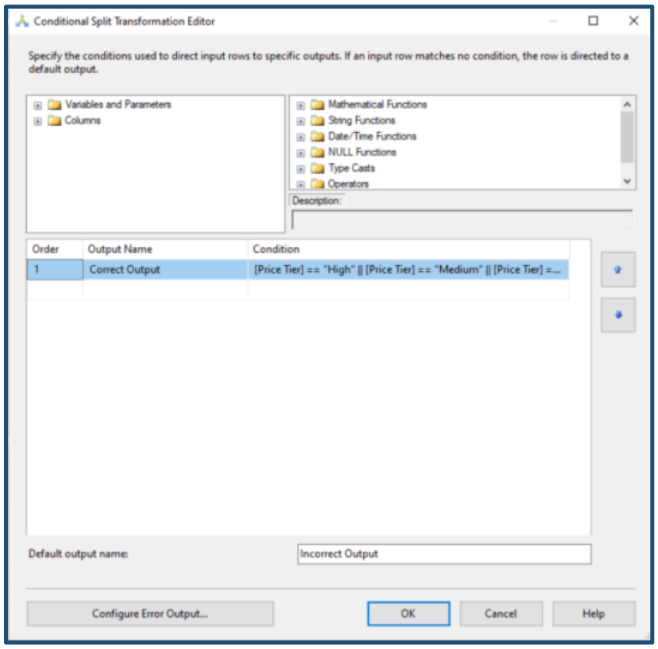
Results | Messages

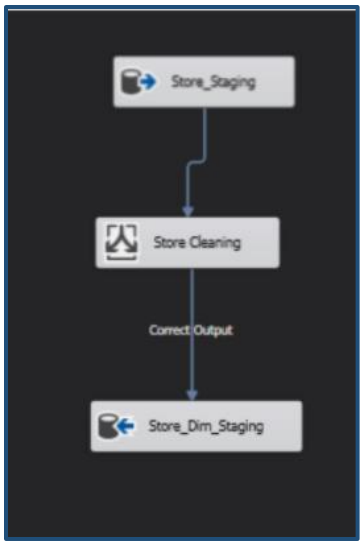
	UPC	DESCRIP
1	1570077445	CTY LN COLBY MILD
2	1570077446	CTY LN CHEDDAR MILD
3	1570077454	CTY LN CHEDDAR MED/S
4	1570077455	CTY LN CHEDDAR SHARP
5	1570077456	CTY LN SWISS OLD WOR
6	1570077458	COUNTY LINE/COLBY JA
7	1570077501	CTY LN MONTEREY JACK
8	1570077503	~COUNTY LINE MUENSTE
9	1570077507	~COUNTY LINE MOZZ CH
10	1570077508	COUNTY LINE EXTRA SH
11	1700310035	KAUKAUNA RANCH BALLS
12	1700310039	KAUKAUNA GARDEN VEGT
13	1700310062	KAUKAUNA SHARP CHEES
14	1700310067	KAUKAUNA SHARP CHED
15	1700310068	KAUKAUNA PORT WINE C
16	1700310074	KAUK CHEDDAR CHEESE
17	1700310077	KAUK PORT WINE CHEES
18	1700310081	KAUKAUNA SHARP LITE
19	1700310084	KAUKAUNA PORTWINE CH
20	1700310087	KAUKAUNA SHARP BALLS
21	1700310091	KAUKAUNA SHARP CHEDD
22	1700310104	KAUKAUNA GARDEN VEG

UPC Table after Transformation and Cleaning

(d) Removing Unknown Values from PRICE_TIER Column in Store Table:

There were many null rows in the STORE_STAGING table due to unknown values in the price_tier column. We deleted those rows using conditional split:





	STOREKEY	STORE_NUM	CITY	PRICE_TIER	STORE_ZONE	ZIP_CODE
1	1	2	River Forest	High	1	60305
2	2	4	Park Ridge	Medium	2	60068
3	3	5	Palatine	Medium	2	60067
4	4	8	Oak Lawn	Low	5	60435
5	5	9	Morton Grove	Medium	2	60053
6	6	12	Chicago	High	7	60660
7	7	14	Glenview	High	1	60025
8	8	18	River Grove	Low	5	60171
9	9	21	Hanover Park	CubFighter	6	60103
10	10	28	Mt. Prospect	Medium	2	60054
11	11	32	Park Ridge	High	1	60068
12	12	33	Chicago	High	7	60657
13	13	40	Bridgeview	CubFighter	6	60455
14	14	44	Western Spring	Medium	2	60558
15	15	45	Wheeling	Medium	2	60090
16	16	46	Carol Stream	Low	5	60187
17	17	47	Addison	Medium	2	60101
18	18	48	Schaumburg	Medium	2	60193
19	19	49	Downers Grove	Medium	2	60515
20	20	50	Hickory Hills	Medium	2	60457

Store Table after Transformation and Cleaning

(e) Transformation of Time data into Month, Quarter and Year details:

We made use of the WEEK_STAGING table to derive month, quarter and year respectively. We first transformed the Start column from string to date datatype and then used the Execute SQL task to get the required columns:

- Update Month:

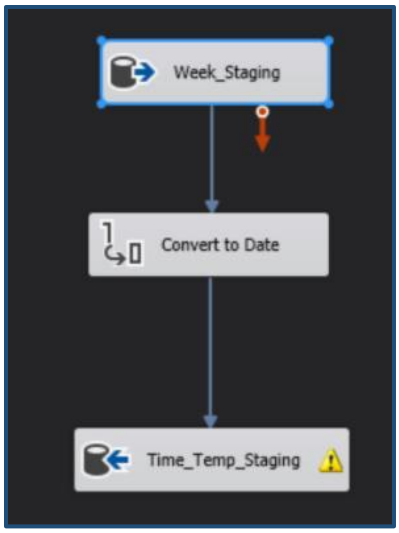
UPDATE [dbo].[WEEK_STAGING] SET [MONTH] =LEFT([START], 2);

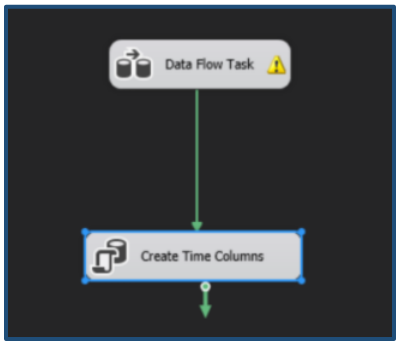
- Update Quarter Detail based on the month:

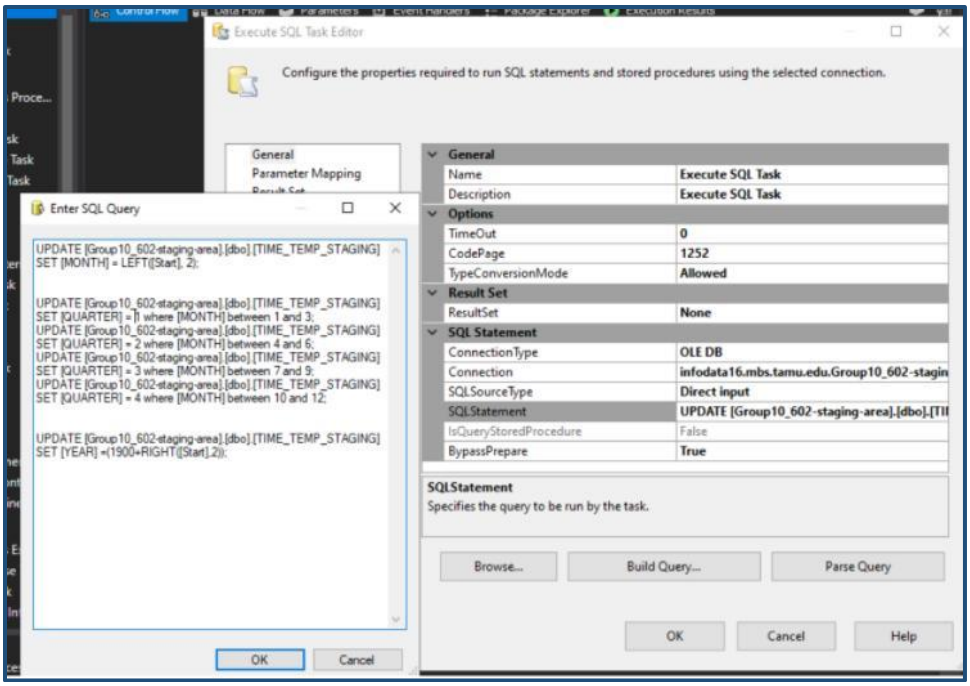
UPDATE [dbo].[WEEK_STAGING] SET [QUARTER] = 1 where [MONTH] between 1 and 3;
UPDATE [dbo].[WEEK_STAGING] SET [QUARTER] = 2 where [MONTH] between 4 and 6;
UPDATE [dbo].[WEEK_STAGING] SET [QUARTER] = 3 where [MONTH] between 7 and 9;
UPDATE [dbo].[WEEK_STAGING] SET [QUARTER] = 4 where [MONTH] between 10 and 12;

- Update Year:

UPDATE [dbo].[WEEK_STAGING] SET [YEAR] =(1900+RIGHT([START], 2));





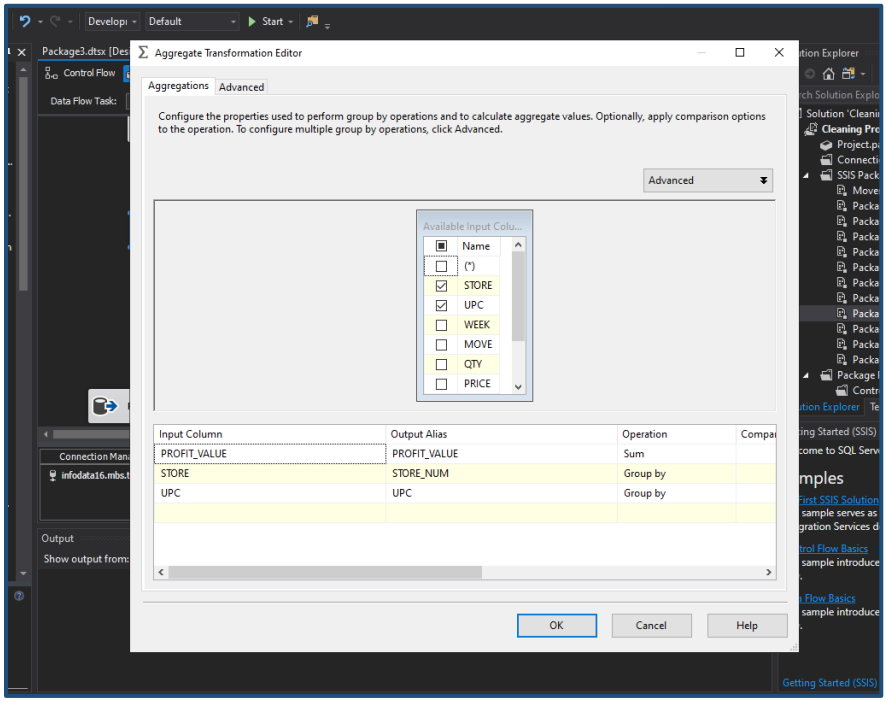


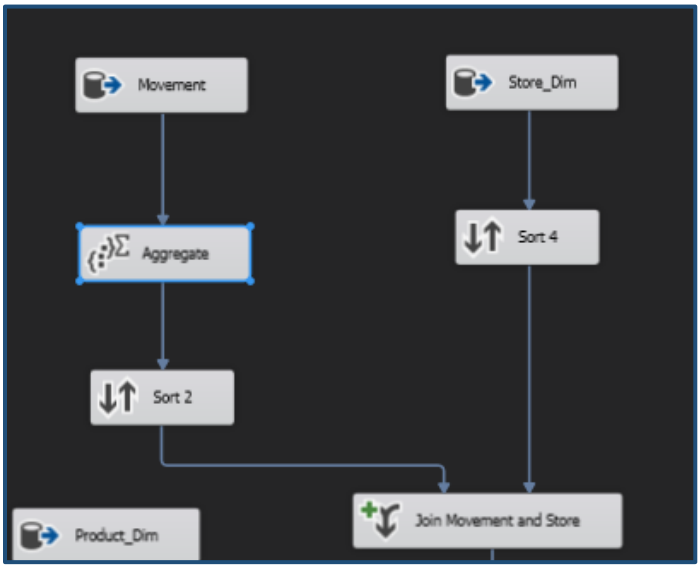
	TIMEKEY	WEEK	MONTH	QUARTER	YEAR
1	1	1	9	3	1989
2	2	2	9	3	1989
3	3	3	9	3	1989
4	4	4	10	4	1989
5	5	5	10	4	1989
6	6	6	10	4	1989
7	7	7	10	4	1989
8	8	8	11	4	1989
9	9	9	11	4	1989
10	10	10	11	4	1989
11	11	11	11	4	1989
12	12	12	11	4	1989
13	13	13	12	4	1989
14	14	14	12	4	1989
15	15	15	12	4	1989
16	16	16	12	4	1989
17	17	17	1	1	1990
18	18	18	1	1	1990
19	19	19	1	1	1990
20	20	20	1	1	1990
21	21	21	2	1	1990
22	22	22	2	1	1990
23	23	23	2	1	1990

Time Table after Transformation and Cleaning

10.7. Plan for Aggregation

We have performed aggregation in the STORE_PERFORMANCE fact table in order to achieve the right level of granularity for answering business questions. Data in the MOVEMENT_STAGING table is stored on a weekly basis for each product sold in a particular store. But for STORE_PERFORMANCE fact we needed the overall Profit value of a product in a particular store. Thus, we grouped the data based on respective store numbers and UPC numbers and ultimately summed up the Profit values.





10.8. Organization of Staging Area

The data extracted from the various data sources using respective extraction rules is stored in the database for staging area – Group10_602-staging-area. Various tables present in the staging area and their source files are shown below:

Table: [Group10_602-staging-area].[dbo].[MOVEMENT_STAGING]

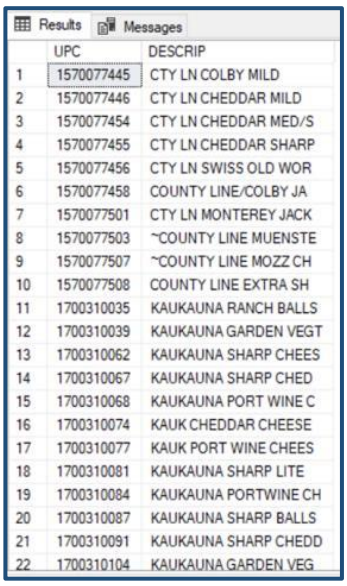
This table is derived using 6 movement files - WFRD, WFRE, WFRJ, WGRO, WCHE, WCER. These files are then combined in the staging area to create one MOVEMENT_STAGING table. It holds the sales and profit values for different stores.

Results Messages

	STORE	UPC	WEEK	MOVE	QTY	PRICE	PROFIT	SALES	PROFIT_VALUE
3...	124	1600067440	336	16	1	3.59	19.19	57.44	11.022736
3...	124	1600067440	337	7	1	3.59	19.19	25.13	4.822447
3...	124	1600067440	338	17	1	3.59	19.19	61.03	11.711657
3...	124	1600067440	339	11	1	3.59	19.19	39.49	7.578131
3...	124	1600067440	340	16	1	3.59	19.19	57.44	11.022736
3...	124	1600067440	341	18	1	3.59	19.19	64.62	12.400578
3...	124	1600067440	342	12	1	3.59	19.19	43.08	8.267052
3...	124	1600067440	343	14	1	3.59	19.19	50.26	9.644894
3...	124	1600067440	344	13	1	3.59	19.19	46.67	8.955973
3...	124	1600067440	345	29	1	3.59	29.18	104.11	30.379298
3...	124	1600067440	346	257	1	2.01	20.44	516.57	105.586908
3...	124	1600067440	347	14	1	3.59	55.4	50.26	27.84404
3...	124	1600067440	348	12	1	3.59	55.4	43.08	23.86632
3...	124	1600067440	349	9	1	3.59	27.23	32.31	8.798013
3...	124	1600067440	350	9	1	3.59	19.05	32.31	6.155055
3...	124	1600067440	351	5	1	3.59	18.99	17.95	3.408705
3...	124	1600067440	352	5	1	3.59	18.96	17.95	3.40332
3...	124	1600067440	353	14	1	3.59	18.93	50.26	9.514218
3...	124	1600067440	354	15	1	3.59	18.91	53.85	10.183035
3...	124	1600067440	355	19	1	3.42	30.01	64.98	19.500498
3...	124	1600067440	356	24	1	2.98	19.46	71.52	13.917792
3...	124	1600067440	357	27	1	2.98	19.46	80.46	15.657516

Table: [Group10_602-staging-area].[dbo].[UPC_CLEANED_STAGING]

We derived this table by combining 6 UPC files - UPCCER, UPCCHE, UPCFRD, UPCFRE, UPCFRJ, UPCGRO. This holds the columns of UPC_Number and Product_Descript which will further be utilized in the Product dimension table.

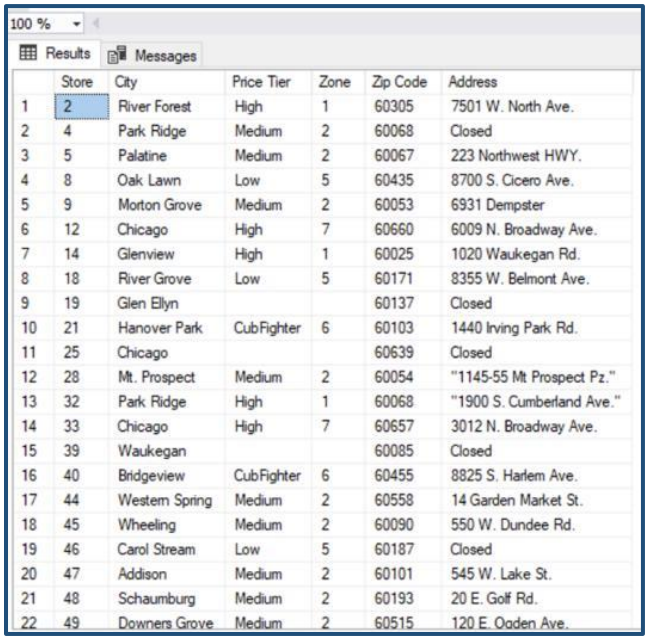


Results Messages

	UPC	DESCRIP
1	1570077445	CTY LN COLBY MILD
2	1570077446	CTY LN CHEDDAR MILD
3	1570077454	CTY LN CHEDDAR MED/S
4	1570077455	CTY LN CHEDDAR SHARP
5	1570077456	CTY LN SWISS OLD WOR
6	1570077458	COUNTY LINE/COLBY JA
7	1570077501	CTY LN MONTEREY JACK
8	1570077503	~COUNTY LINE MUENSTE
9	1570077507	~COUNTY LINE MOZZ CH
10	1570077508	COUNTY LINE EXTRA SH
11	1700310035	KAUKAUNA RANCH BALLS
12	1700310039	KAUKAUNA GARDEN VEGT
13	1700310062	KAUKAUNA SHARP CHEES
14	1700310067	KAUKAUNA SHARP CHED
15	1700310068	KAUKAUNA PORT WINE C
16	1700310074	KAUK CHEDDAR CHEESE
17	1700310077	KAUK PORT WINE CHEES
18	1700310081	KAUKAUNA SHARP LITE
19	1700310084	KAUKAUNA PORTWINE CH
20	1700310087	KAUKAUNA SHARP BALLS
21	1700310091	KAUKAUNA SHARP CHEDD
22	1700310104	KAUKAUNA GARDEN VEG

Table: [Group10_602-staging-area].[dbo].[STORE_STAGING]

This table is derived from Dominick's Stores table (Part 6) in DFF's lab manual. It contains information geographic information about all the stores of DFF and is further utilized in the creation of Store dimension table.



100 %

Results Messages

	Store	City	Price Tier	Zone	Zip Code	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.
11	25	Chicago			60639	Closed
12	28	Mt. Prospect	Medium	2	60054	"1145-55 Mt Prospect Pz."
13	32	Park Ridge	High	1	60068	"1900 S. Cumberland Ave."
14	33	Chicago	High	7	60657	3012 N. Broadway Ave.
15	39	Waukegan			60085	Closed
16	40	Bridgeview	CubFighter	6	60455	8825 S. Harlem Ave.
17	44	Western Spring	Medium	2	60558	14 Garden Market St.
18	45	Wheeling	Medium	2	60090	550 W. Dundee Rd.
19	46	Carol Stream	Low	5	60187	Closed
20	47	Addison	Medium	2	60101	545 W. Lake St.
21	48	Schaumburg	Medium	2	60193	20 E. Golf Rd.
22	49	Downers Grove	Medium	2	60515	120 E. Ogden Ave.

Table: [Group10_602-staging-area].[dbo].[DEMO_STAGING]

This table derived from the file called demo.csv. It contains information regarding the percentage of people of certain age visiting different stores (below_9% and above_60%). These values are further utilized in the creation of STORE_PERFORMANCE fact table.

Results Messages

	"MMID"	"CITY"	"ZIP"	"STORE"	"ZONE"	"AGE9"	"AGE60"
1	16892	"RIVER FOREST"	60305	2	1	0.117508576	0.232864734
2	16893	"PARK RIDGE"	60068	4	2	0.0950895057	0.26202989
3	16894	"PALATINE"	60067	5	2	0.1414334827	0.1173680317
4	16895	"OAK LAWN"	60453	8	5	0.123155416	0.2523940345
5	16896	"MORTON GROVE"	60053	9	2	0.1035030974	0.2691190176
6	16898	"CHICAGO"	60660	12	7	0.1056967397	0.178341405
7	16899	"GLENVIEW"	60025	14	1	0.129589372	0.2139492754
8	16901	"RIVER GROVE"	60171	18	5	0.1100949839	0.2723133684
9	16903	"HANOVER PARK"	60103	21	6	0.1759263459	0.0668964579
10	16905	"MOUNT PROSPECT"	60056	28	2	0.1288795371	0.2133087849
11	16906	"PARK RIDGE"	60068	32	1	0.0990606319	0.2549530316
12	16907	"CHICAGO"	60657	33	7	0.0460709172	0.1341699655
13	16909	"BRIDGEVIEW"	60455	40	6	0.1336846485	0.1818518005
14	16912	"WESTERN SPRINGS"	60558	44	2	0.1448834853	0.1909827761
15	16913	"WHEELING"	60090	45	2	0.1467187625	0.1288573479
16	16915	"ADDISON"	60101	47	2	0.1429616817	0.125798297
17	16916	"SCHAUMBURG"	60193	48	2	0.1217670803	0.0979219614
18	16917	"DOWNERS GROVE"	60515	49	2	0.1348777349	0.1874731875
19	16918	"HICKORY HILLS"	60457	50	2	0.1244204019	0.1533573759
20	16919	"PALOS HEIGHTS"	60463	51	3	0.132472108	0.1761597181
21	16920	"NORTHBROOK"	60062	52	1	0.13660619	0.1522411953
22	16921	"CHICAGO"	60662	53	7	0.1208391392	0.3002786809
23	16922	"NAPERVILLE"	60540	54	2	0.1479145854	0.0902222777

Table: [Group10_602-staging-area].[dbo].[WEEK_STAGING]

This table is derived from Weeks' decode table (Part 8) from the DFF's lab manual. This table was further transformed to derive month, quarter and year from the information provided using SQL queries.

Results Messages

	Week #	Start	End	Special Events
1	1	09/14/89	09/20/89	
2	2	09/21/89	09/27/89	
3	3	09/28/89	10/04/89	
4	4	10/05/89	10/11/89	
5	5	10/12/89	10/18/89	
6	6	10/19/89	10/25/89	
7	7	10/26/89	11/01/89	Halloween
8	8	11/02/89	11/08/89	
9	9	11/09/89	11/15/89	
10	10	11/16/89	11/22/89	
11	11	11/23/89	11/29/89	Thanksgiving
12	12	11/30/89	12/06/89	
13	13	12/07/89	12/13/89	
14	14	12/14/89	12/20/89	
15	15	12/21/89	12/27/89	Christmas
16	16	12/28/89	01/03/90	New-Year
17	17	01/04/90	01/10/90	
18	18	01/11/90	01/17/90	
19	19	01/18/90	01/24/90	
20	20	01/25/90	01/31/90	
21	21	02/01/90	02/07/90	
22	22	02/08/90	02/14/90	
23	23	02/15/90	02/21/90	Presidents Day

Table: [Group10_602-staging-area].[dbo].[TIME_DIM_STAGING]

This table is derived from WEEK_STAGING table and includes derived columns for quarter, month and year using SQL queries as already explained above.

	TIMEKEY	WEEK	MONTH	QUARTER	YEAR
1	1	1	9	3	1989
2	2	2	9	3	1989
3	3	3	9	3	1989
4	4	4	10	4	1989
5	5	5	10	4	1989
6	6	6	10	4	1989
7	7	7	10	4	1989
8	8	8	11	4	1989
9	9	9	11	4	1989
10	10	10	11	4	1989
11	11	11	11	4	1989
12	12	12	11	4	1989
13	13	13	12	4	1989
14	14	14	12	4	1989
15	15	15	12	4	1989
16	16	16	12	4	1989
17	17	17	1	1	1990
18	18	18	1	1	1990
19	19	19	1	1	1990
20	20	20	1	1	1990
21	21	21	2	1	1990
22	22	22	2	1	1990
23	23	23	2	1	1990

Temp Tables:

In order to derive at these tables, we made use of some temp tables in between. The list of all temp tables used is given below:

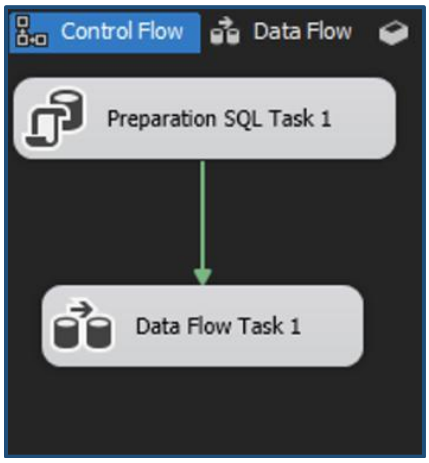
1. MOVEMENT_PRODUCT_TEMP_STAGING
2. TIME_TEMP_STAGING
3. UPC_TEMP_STAGING
4. DEMO_TEMP_STAGING
5. MOVEMENT_STORE_TEMP_STAGING

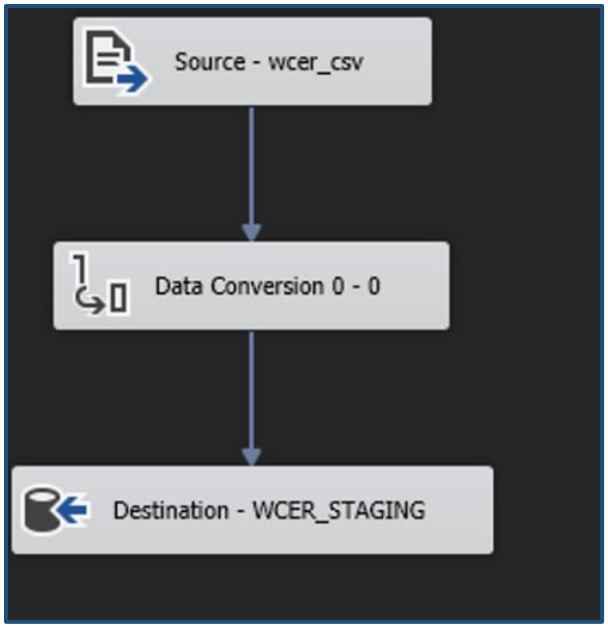
10.9. Data Extraction and Loading Procedures

MOVEMENT FILES

1. Wcer.csv to WCER_STAGING

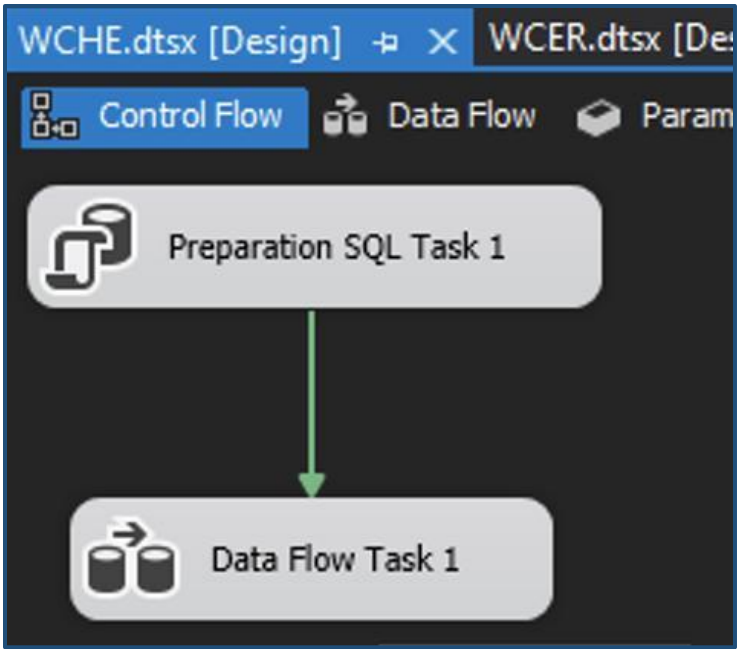
The data from wcer.csv has been loaded into the WCER_STAGING Table in the staging area. While loading the data, we transformed its data types in mapping section of import and export wizard of its package.

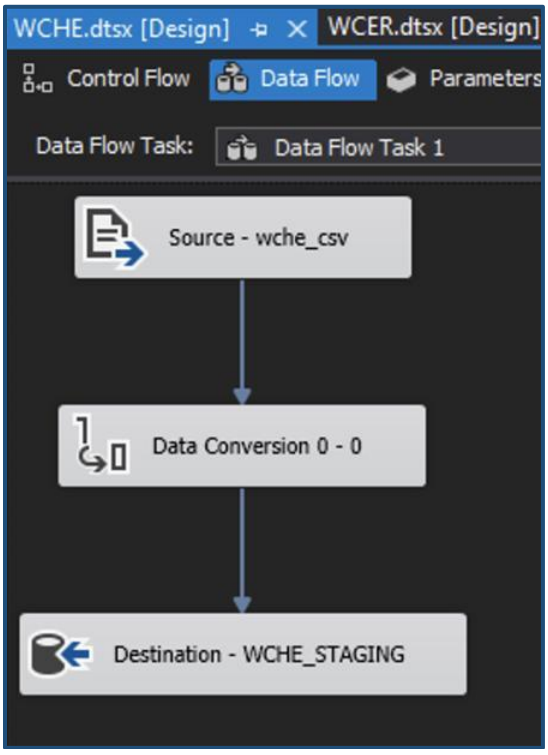




2. Wche.csv to WCHE_STAGING

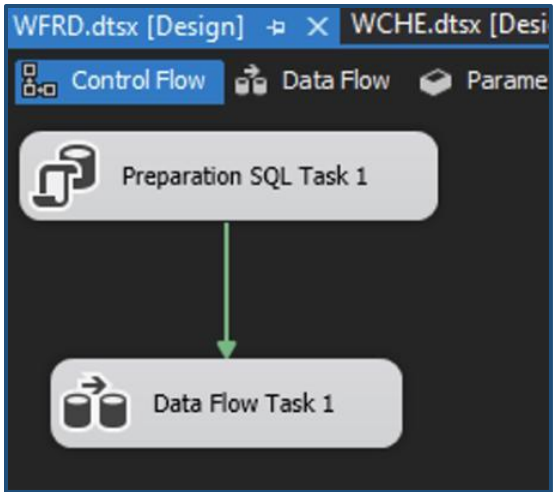
The data from wche.csv has been loaded into the WCHE_STAGING Table in the staging area. While loading the data, we transformed its data types in mapping section of import and export wizard of its package.

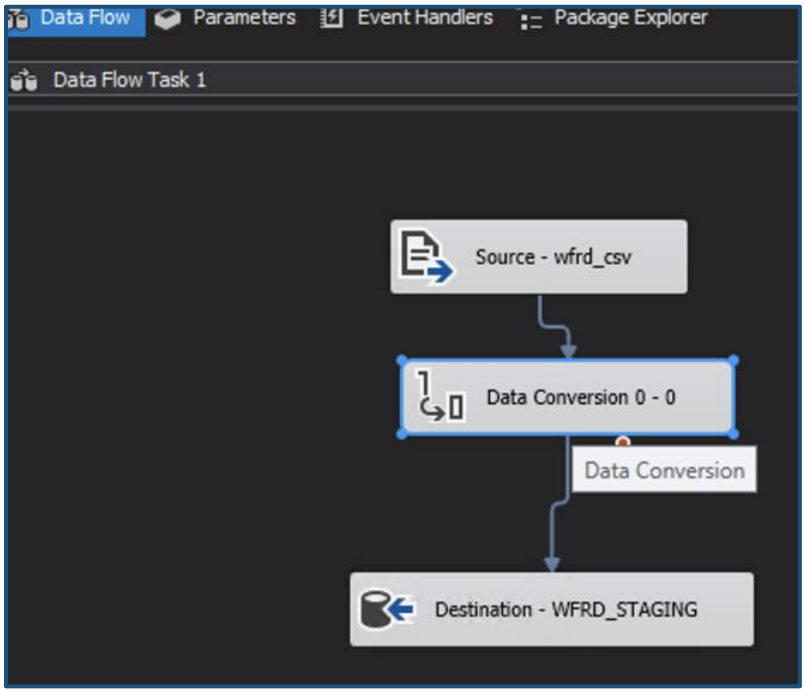




3. Wfrd.csv to WFRD_STAGING

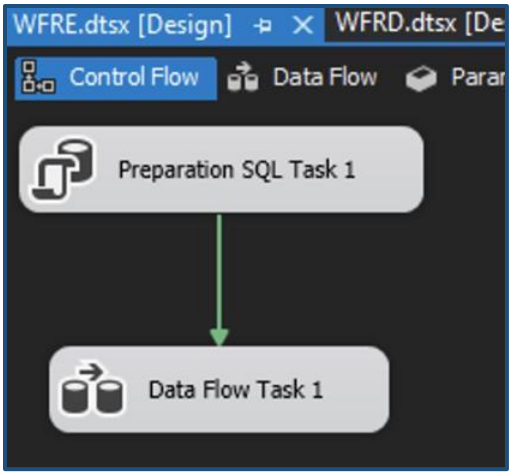
The data from wfrd.csv has been loaded into the WFRD_STAGING Table in the staging area. While loading the data, we transformed its data types in mapping section of import and export wizard of its package.

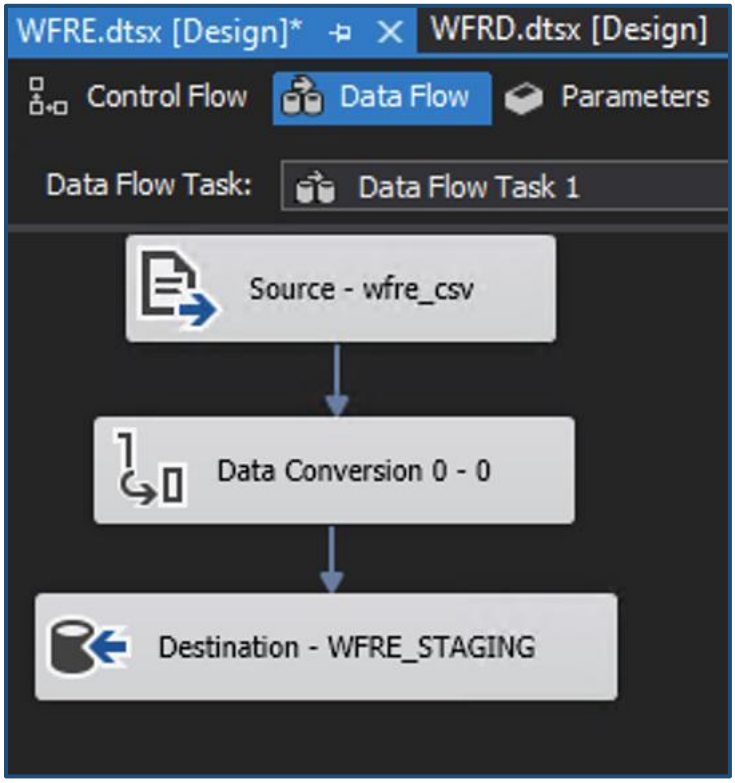




4. Wfre.csv to WFRE_STAGING

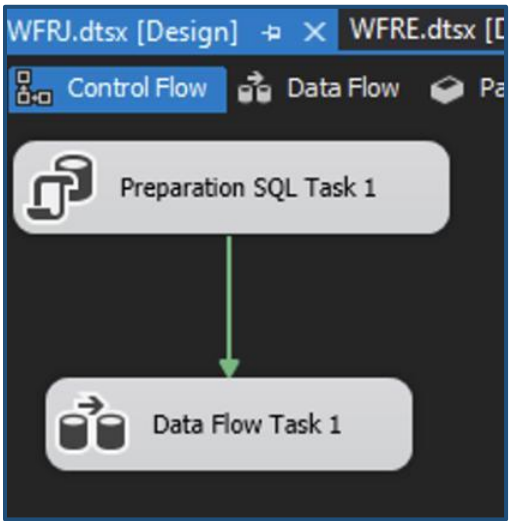
The data from wfre.csv has been loaded into the WFRE_STAGING Table in the staging area. While loading the data, we transformed its data types in mapping section of import and export wizard of its package.

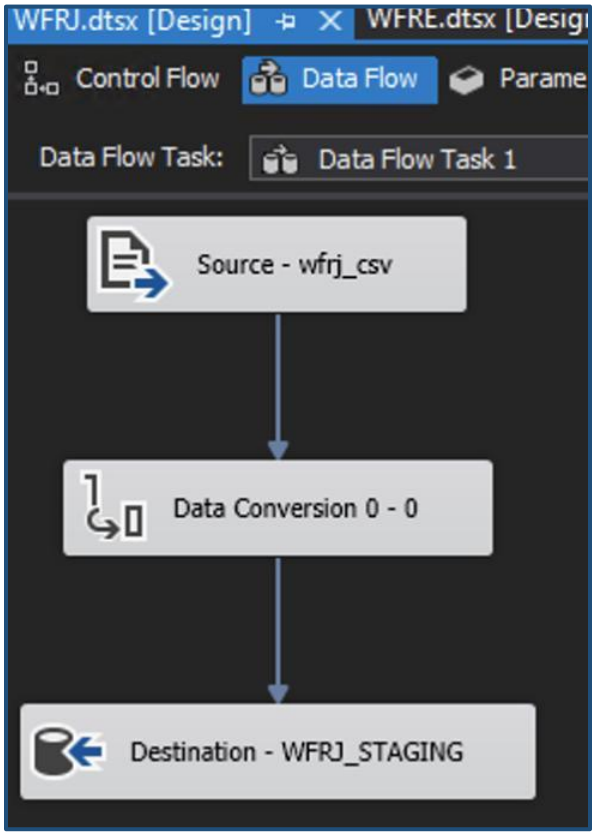




5. Wfrj.csv to WFRJ_STAGING

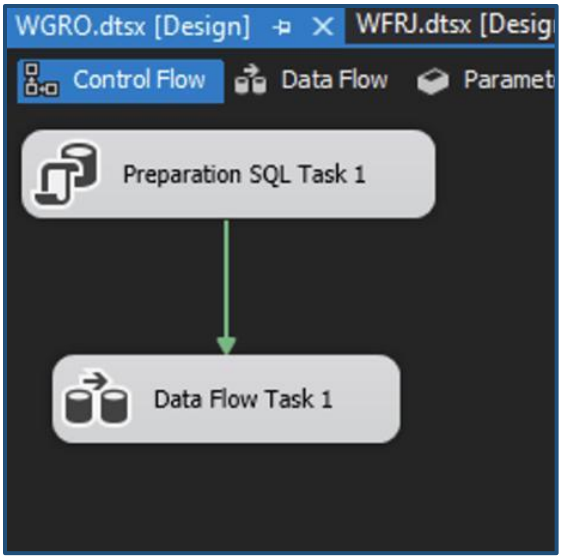
The data from wfrj.csv has been loaded into the WFRJ_STAGING Table in the staging area. While loading the data, we transformed its data types in mapping section of import and export wizard of its package.





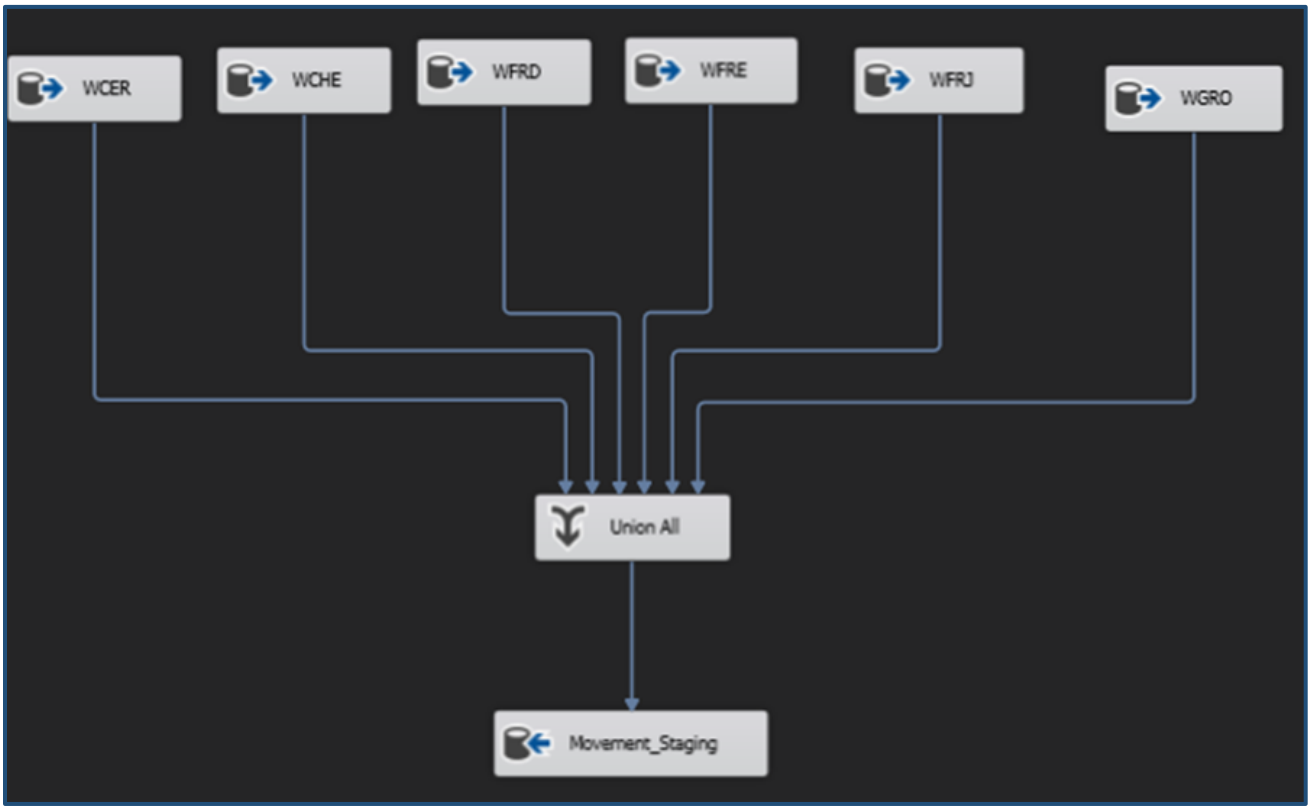
6. Wgro.csv to WGRO_STAGING

The data from wgro.csv has been loaded into the WGRO_STAGING Table in the staging area. While loading the data, we transformed its data types in mapping section of import and export wizard of its package.



7. MOVEMENT_STAGING

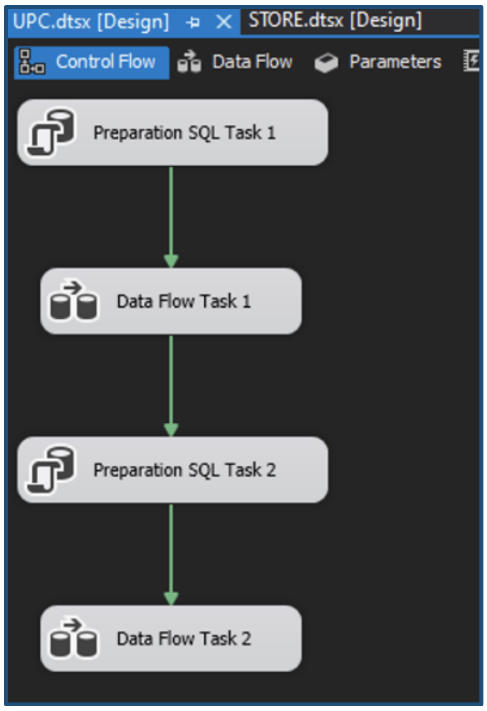
Once all the movement files were loaded into the staging area, union was performed on them and was eventually loaded into MOVEMENT_STAGING.

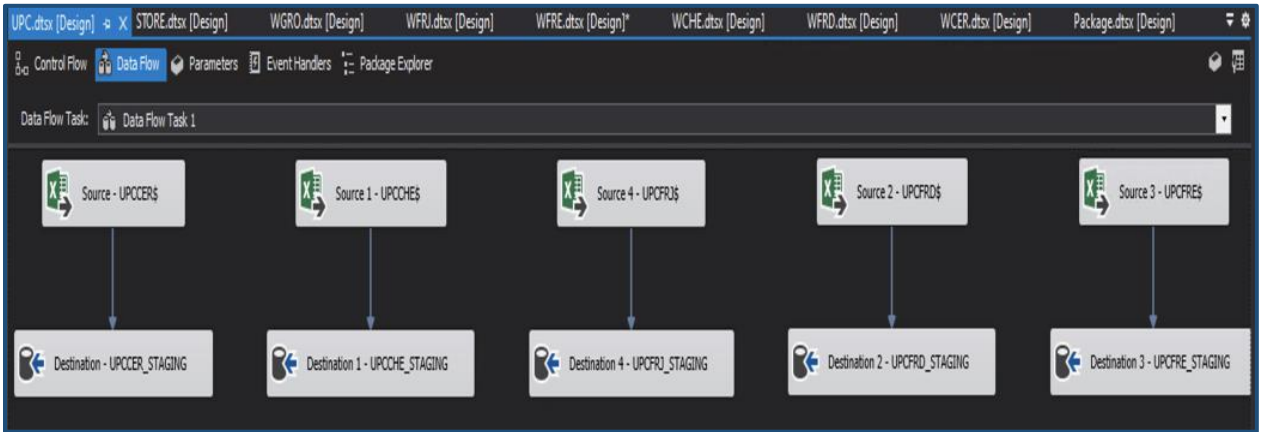


UPC Files

1. UPC_Staging

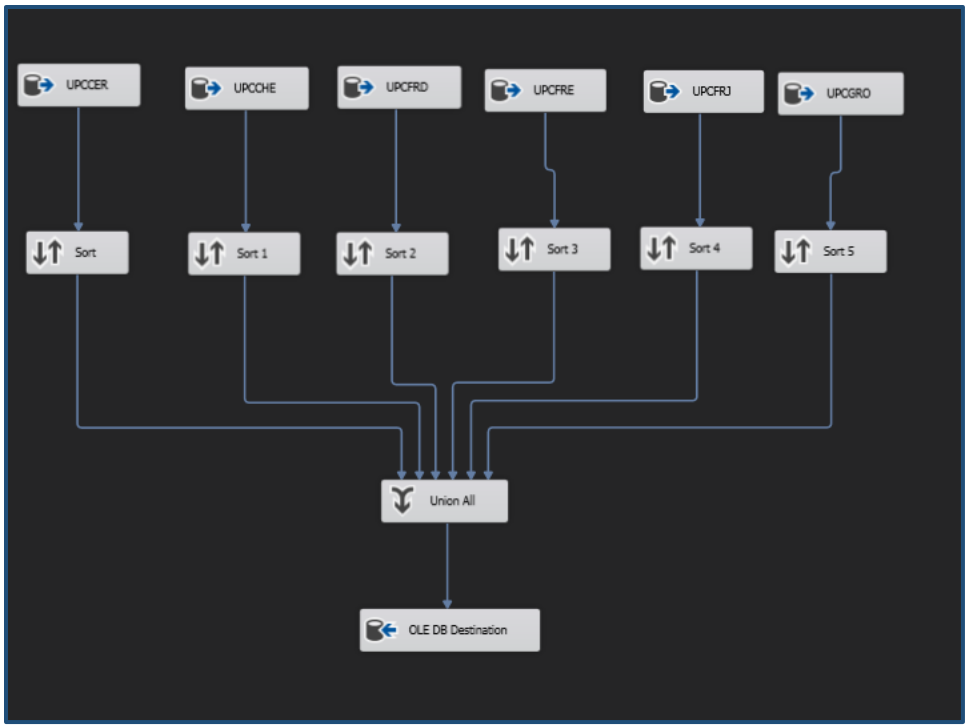
We had multiple UPC files i.e. UPCCER, UPCCHE, UPCFRD, UPCFRE, UPCFRJ, UPCGRO in csv format. All these files were combined in a UPC.xls file and then using SSIS import and export wizard they were loaded into UPCCER_STAGING, UPCCHE_STAGING, UPCFRD_STAGING, UPCFRE_STAGING, UPCFRJ_STAGING, UPCGRO_STAGING. While loading the files we also updated the data formats in the mapping section of the SSIS import and export wizard.

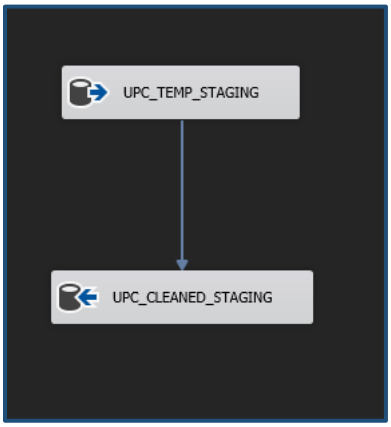




2. UPC_CLEANED_STAGING

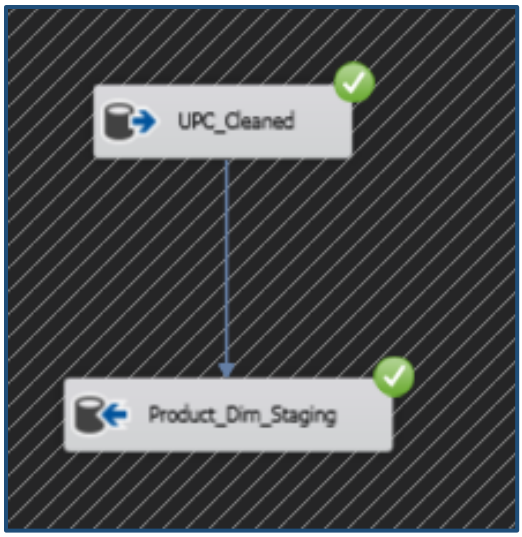
Once all the UPC files were loaded into their respective destinations, we performed union to combine all the UPC files into one table. After that we conducted transformations over the file by removing columns which were not required, and we also dealt with numerous duplicate values which were present in the file. The data was finally loaded into UPC_CLEANED_STAGING.





3. UPC_CLEANED_STAGING to PRODUCT_DIM

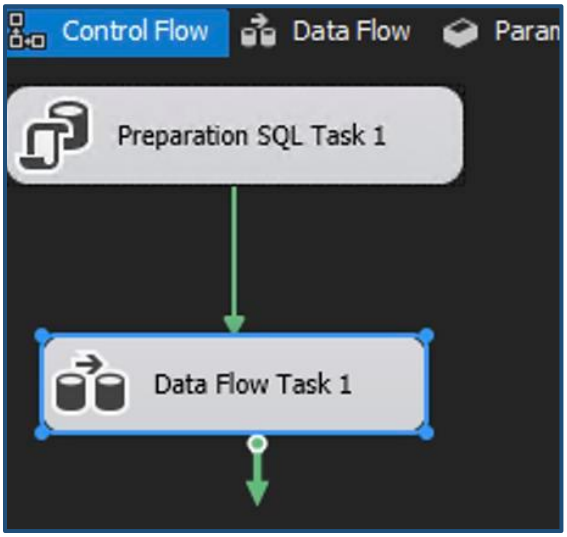
After getting the cleaned data moved into UPC_CLEANED_STAGING table, we generated surrogate key and then moved the data into PRODUCT_DIM_STAGING.

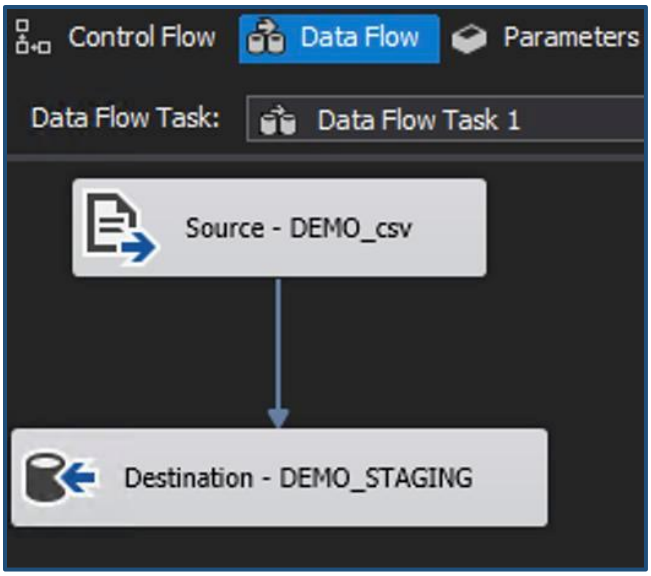


DEMO DATA

1. Demo.csv to DEMO_STAGING

Demo.csv file contains the demographics information. It had numerous columns. Since, we only needed a few of them for our analysis when we moved the data from demo.csv to DEMO_STAGING we only selected a few of them using SSIS import and export wizard.





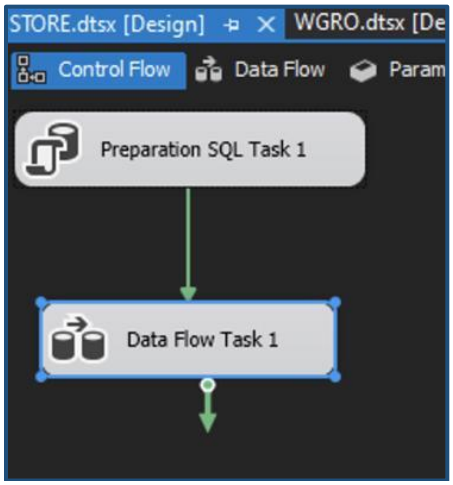
2. DEMO_STAGING to DEMO_CLEANED_STAGING

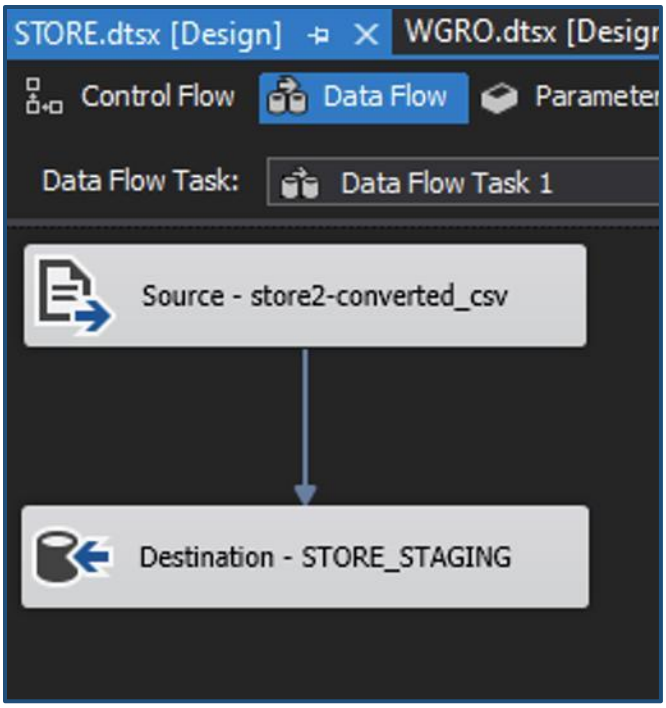
The data present in data staging area had numerous null values so we dealt with those null values by removing them from this table and loaded the cleaned data into the DEMO_CLEANED_STAGING table.

STORE DATA

1. Store.csv to STORE_STAGING

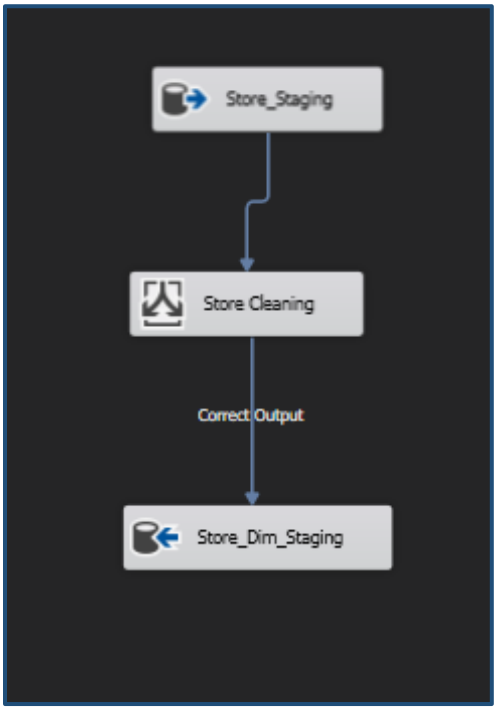
We derived the store.csv file from Dominick's Stores table (Part 6) in DFF's lab manual. It contains geographic information about all the stores of DFF and was used to create Store.csv. After this, we loaded the store.csv into STORE_STAGING.





2. STORE_DIM_STAGING

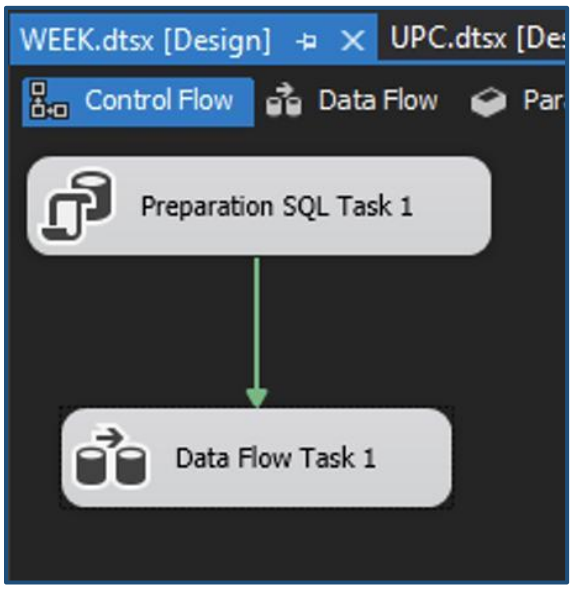
Once data was loaded in the STORE_STAGING table, we conducted transformations in this table to deal with the null values which were present in PRICE_TIER column. We used conditional split transformation editor to make this transformation and eventually the cleaned data was loaded into STORE_DIM_STAGING table.

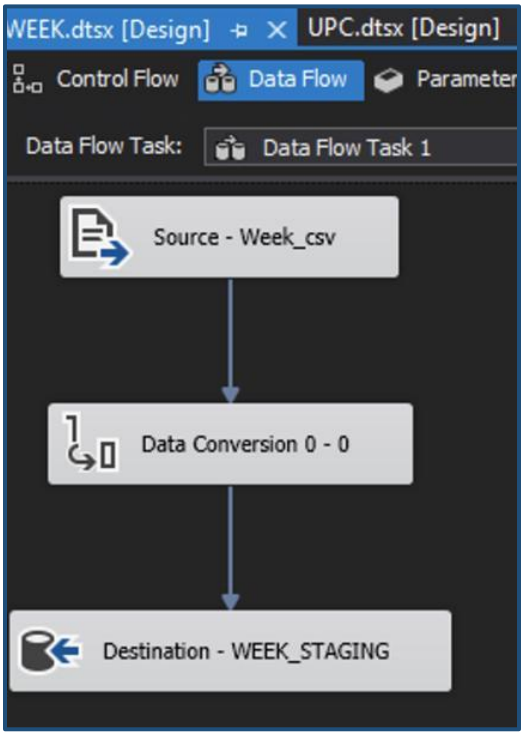


WEEK DATA

1. Week.csv to WEEK_STAGING

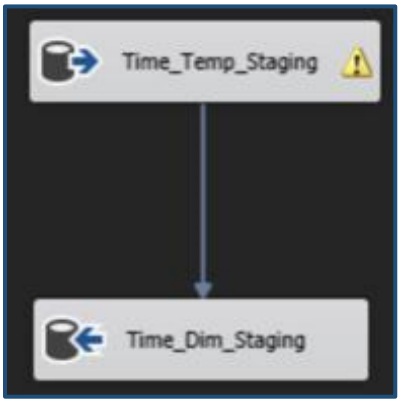
This table is derived from Weeks' decode table (Part 8) from the DFF's lab manual. We used SSIS import and export wizard to create Week.csv file and then we loaded the data into WEEK_STAGING.





2. TIME_DIM_STAGING

Once data was loaded into the WEEK_STAGING table, we conducted transformations over it to derive month, quarter and year from the information provided. We also dropped a few columns which were not needed for our analysis. The final data was loaded into the TIME_DIM_STAGING table.



Further, we also created SALES_FACT and STORE_PERFORMNACE fact tables in the staging area which are used to load data in the final fact tables in the data marts and its logic has been described in the later sections.

10.10. Data Granularity in Independent Data Marts:

SALES Data Mart:

The aim of this data mart is to analyze sales across Dominick's stores for different time periods. In order to answer our business questions, we required to have data at the week level (most atomic level). We can also drill-down and drill-up in the data whenever the need arises.

STORE_PERFORMANCE Data Mart:

The aim of this data mart is to evaluate performance of the different DFF stores. Our business questions require us to evaluate average profit margins across the stores for different product categories. Due to this, we have used aggregated data in the STORE_PERFORMANCE fact table at the store level.

10.11. ETL for Dimension Tables

STORE_DIM (DW)

The STORE_DIM_STAGING table from the staging area has been used to prepare the STORE_DIM table in the data warehouse area as shown below:

PRODUCT_DIM (DW)

The PRODUCT_DIM table has been derived from the PRODUCT_DIM_STAGING table which is further derived from UPC_STAGING table. This table holds information regarding UPC numbers and description of products. The data was cleaned and transformed as already shown above and then it is ultimately transferred to the PRODUCT_DIM table.

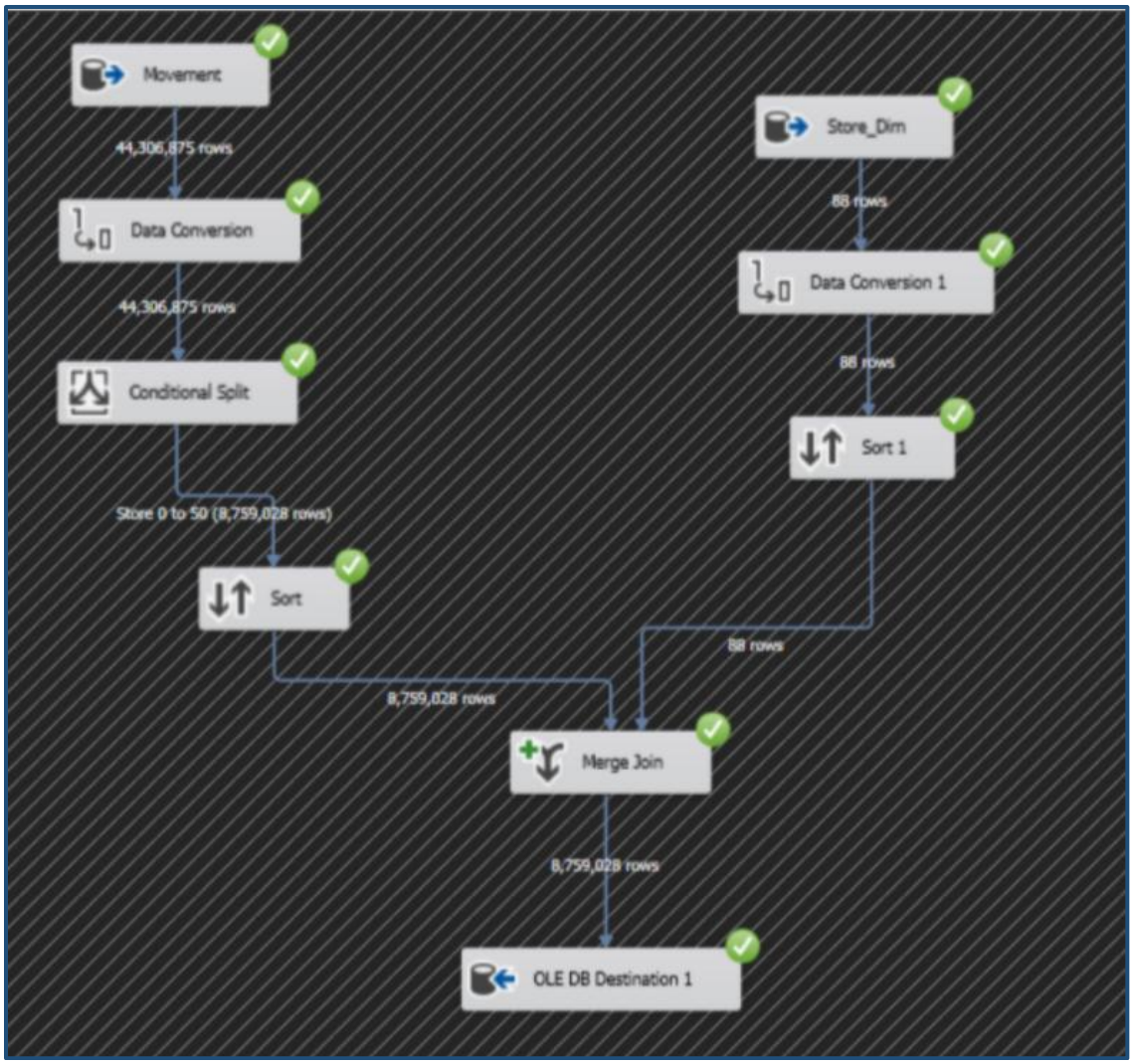
TIME_DIM (DW)

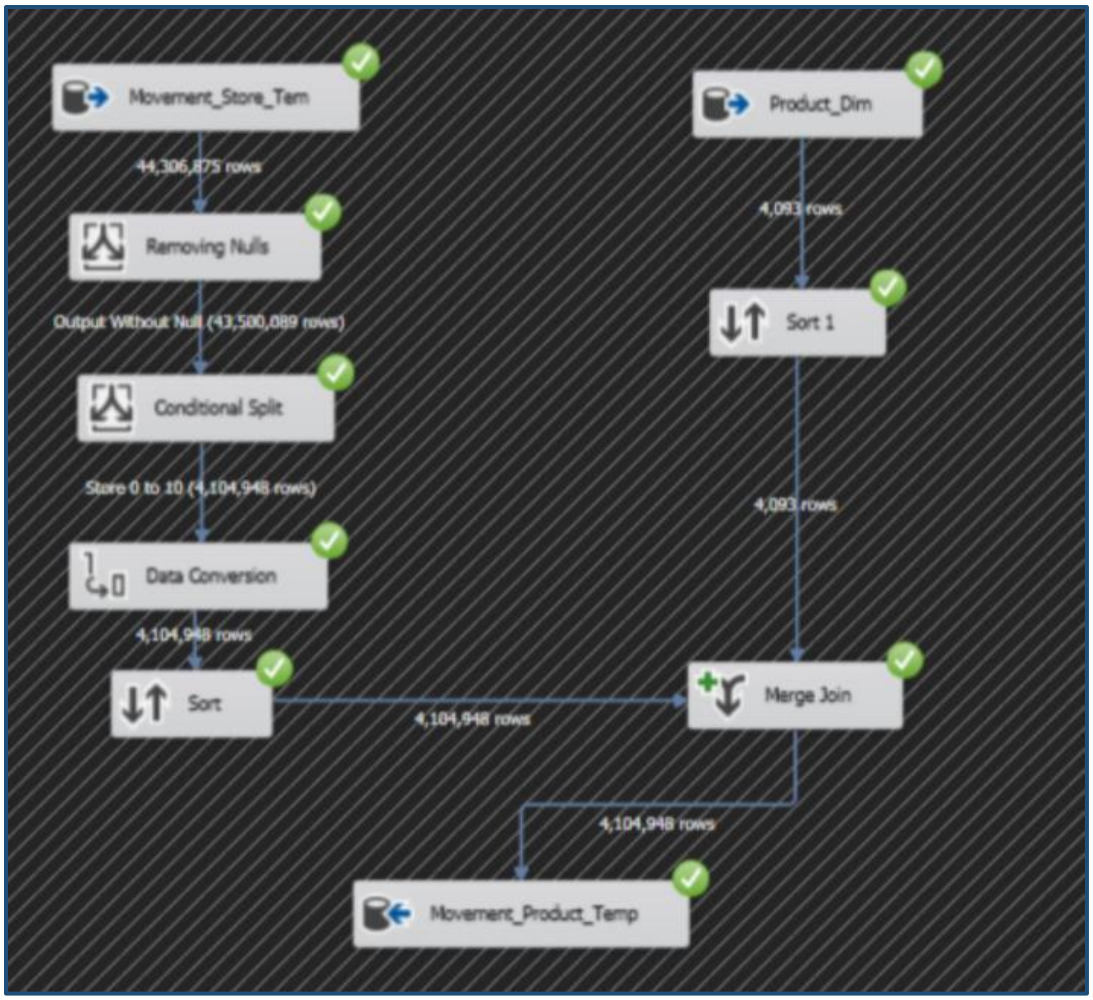
The TIME_DIM_STAGING table in the staging area contains information regarding dates, weeks, months, quarters and years. The columns of month, quarter and year were calculated from the WEEK_STAGING table as already shown above and then the data was loaded into TIME_DIM_STAGING table which is further transferred to TIME_DIM in the data warehouse area.

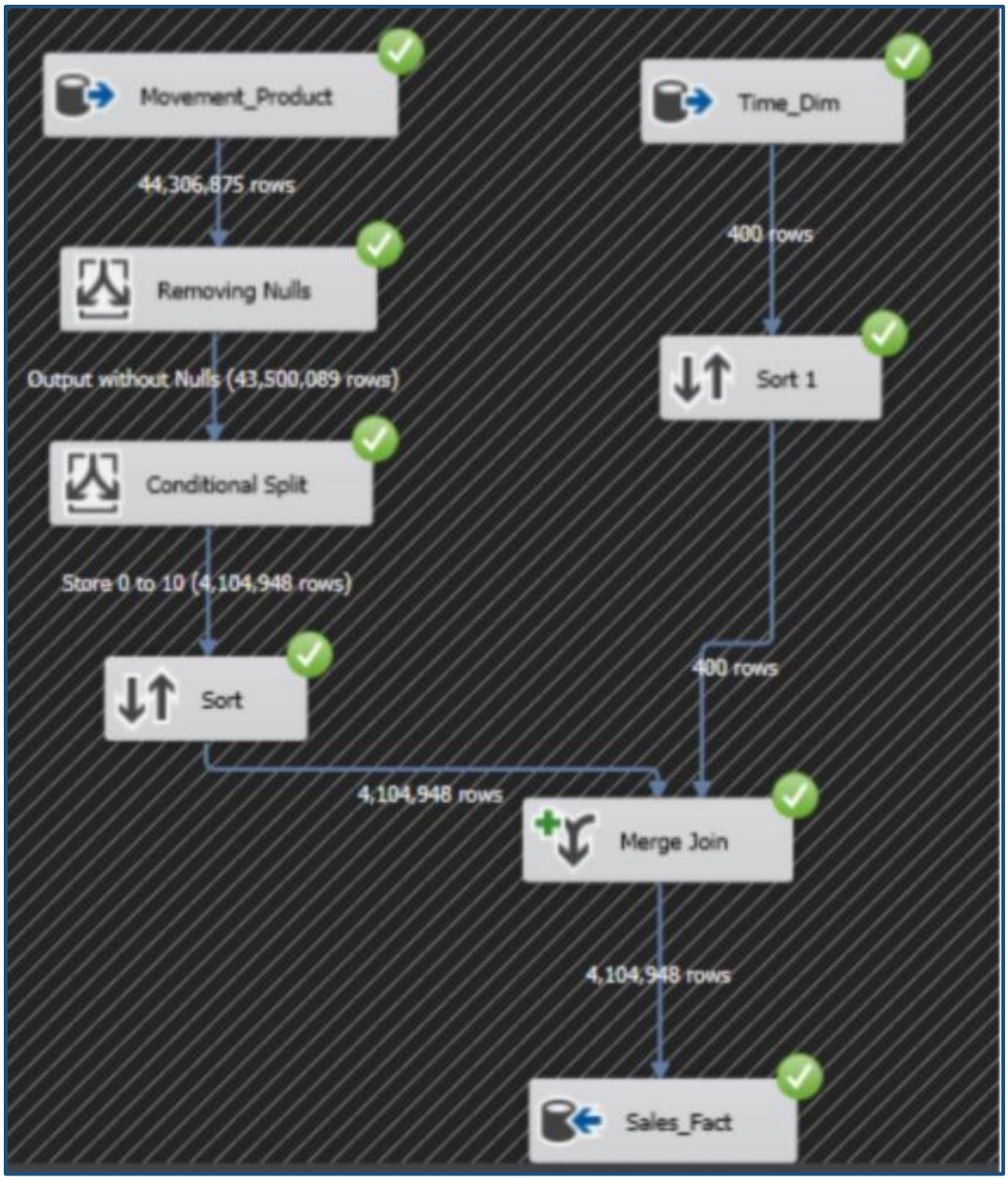
10.12. ETL for Fact Tables

SALES_FACT (DW)

The SALES_FACT table uses 4 tables as sources – MOVEMENT_STAGING, PRODUCT_DIM_STAGING, STORE_DIM_STAGING, TIME_DIM_STAGING from which the final data is extracted. We performed join operation in stages utilizing two tables at a time. This was done in order to deal with the huge amount of data in the tables (~44 million rows). We split the data on the basis of store numbers into chunks of ~5 million rows each. All the tables are then joined using Merge Joins on the source tables of the staging area after sorting each of these tables based on appropriate attributes.

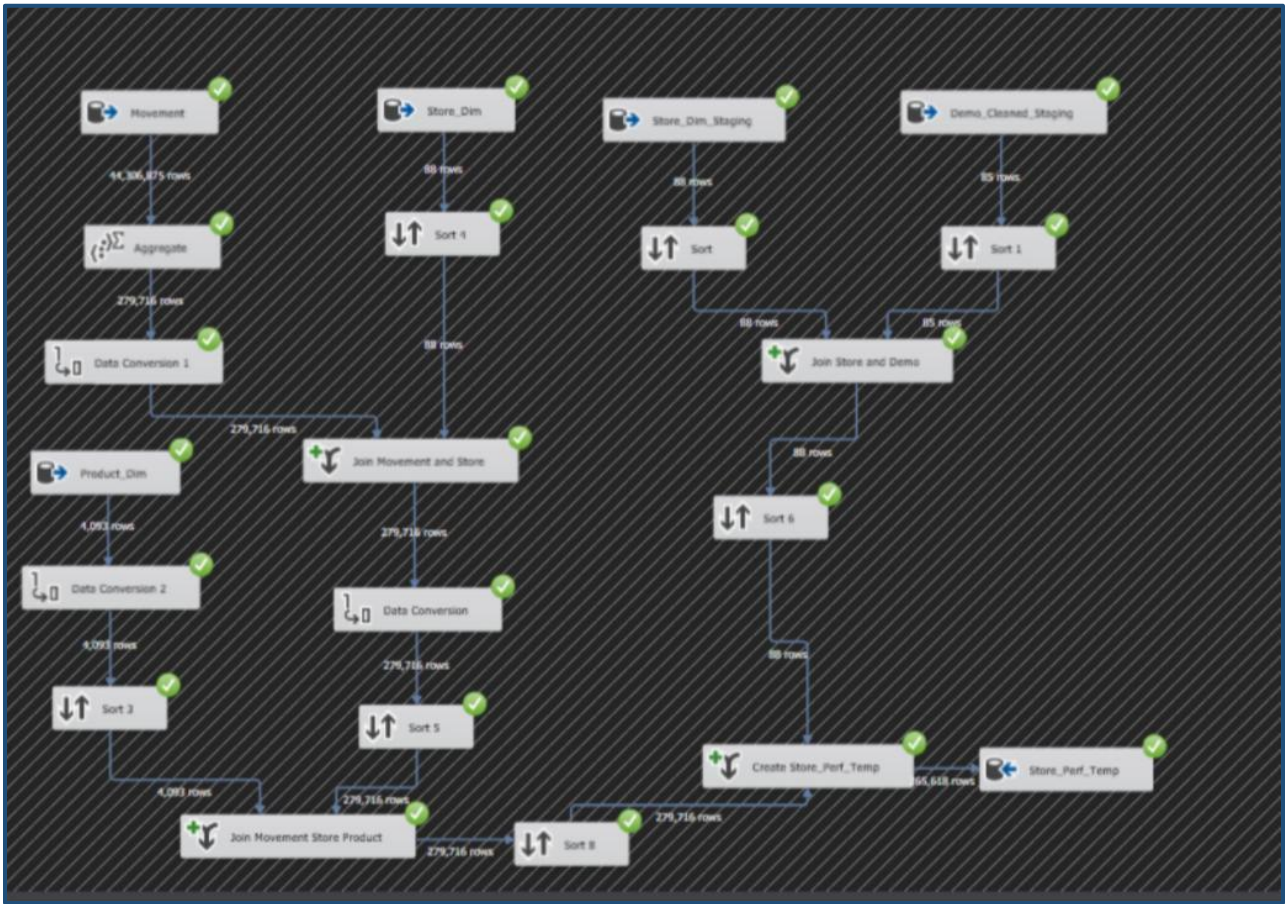

Movement
44,306,875 rows
Data Conversion
44,306,875 rows
Conditional Split
Store 0 to 50 (8,759,028 rows)
Sort
8,759,028 rows
Store_Dim
88 rows
Data Conversion 1
88 rows
Sort 1
88 rows
Merge Join
8,759,028 rows
OLE DB Destination 1


Movement_Store_Tem
44,306,875 rows
Removing Nulls
Output Without Null (43,500,089 rows)
Conditional Split
Store 0 to 10 (4,104,948 rows)
Data Conversion
4,104,948 rows
Sort
4,104,948 rows
Product_Dim
4,093 rows
Sort 1
4,093 rows
Merge Join
4,104,948 rows
Movement_Product_Temp


Movement_Product
44,306,875 rows
Removing Nulls
Output without Nulls (43,500,089 rows)
Conditional Split
Store 0 to 10 (4,104,948 rows)
Sort
Time_Dim
400 rows
Sort 1
400 rows
4,104,948 rows
Merge Join
4,104,948 rows
Sales_Fact

STORE_PERF_FACT (DW)

The STORE_PERFORMANCE_FACT table uses 3 tables as sources – MOVEMENT_STAGING, STORE_DIM_STAGING and PRODUCT_DIM_STAGING from which the final data has been extracted. The data from the MOVEMENT_STAGING table has been aggregated as we have already discussed above. All the tables are then joined using Merge Joins on the source tables of the staging area after sorting each of these tables based on appropriate attributes.



10.13. Mapping Definition describing the Source to End Table for all Dimension and Fact Tables

DIMENSION TABLE: STORE_DIM						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
	Surrogate Key	STORE_DIM	STOREKEY	int	Not Null	New column
Dominick_Store.csv	Store		STORE_NUM	int	Not Null	
	Price Tier		PRICE_TIER	varchar(30)		
	Zip Code		ZIP_CODE	int		
	Zone		STORE_ZONE	int		
	City		CITY	varchar(30)		

DIMENSION TABLE: PRODUCT_DIM						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
	Surrogate Key		PRODUCTKEY	int	Not Null	New column
UPC.csv	Product name	PRODUCT_DIM	PRODUCT_NAME	varchar(30)		
	UPC number		UPC_NUMBER	int	Not Null	

DIMENSION TABLE: TIME_DIM

SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
	Surrogate Key	TIME_DIM	TIMEKEY	int	Not null	New column
Week Decode Table	Week number		WEEK	int	Week 1-400	
	Start Date		MONTH	int	Map month from start dates of week decode table	
	MONTH		QUARTER	int	Map quarter from MONTH column	
	Start Date		YEAR	int	Map year from start dates of week decode table	

FACT TABLE: SALES_FACT						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
STORE_DIM	STOREKEY	SALES_FACT	STOREKEY	int		Replicate
PRODUCT_DIM	PRODUCTKEY		PRODUCTKEY	int		Replicate
TIME_DIM	TIMEKEY		TIMEKEY	int		Replicate

Movement.csv	Movement, Quantity, Price		SALES	float	Computer from movement files	Sales = Price * Movement / Quantity

DIMENSION TABLE: STORE_PERFORMANCE_FACT						
SOURCE		DESTINATION			RULES	
Source Table	Source Column	Target Table	Target Column	Target Datatype	Business Rules	Mapping Function
STORE_DIM	STOREKEY	STORE_PERFORMANCE_FACT	STOREKEY	int	Not Null	Replicate
TIME_DIM	TIMEKEY		TIMEKEY	int	Not Null	Replicate
Movement Files	Profit Margin		Profit	float		Profit Percent * Sales
Demography.csv	Below_9		BELOW_9%	float		Replicate
	Above_60		ABOVE_60%	float		Replicate

10.14. SQL Commands Used for ETL Operations

```
CREATE TABLE [STORE_DIM_STAGING] (
    [STOREKEY] int identity(1,1) primary key not null,
    [STORE_NUM] varchar(50),
    [CITY] varchar(100),
    [PRICE_TIER] varchar(50),
    [STORE_ZONE] varchar(100),
    [ZIP_CODE] varchar(100)
)

CREATE TABLE [PRODUCT_DIM_STAGING] (
    [PRODUCTKEY] int identity(1,1) primary key not null,
```

```
    [UPC] int,
    [DESCRIP] nvarchar(255)
)

CREATE TABLE [TIME_DIM_STAGING] (
    [TIMEKEY] int identity(1,1) primary key not null,
    [WEEK] varchar(50),
    [MONTH] int,
    [QUARTER] int,
    [YEAR] int
)

CREATE TABLE [SALES_FACT] (
    [TIMEKEY] int NOT NULL,
    [SALES] float,
    [STOREKEY] int NOT NULL,
    [PRODUCTKEY] int NOT NULL,
    PRIMARY KEY (STOREKEY, PRODUCTKEY, TIMEKEY),
    FOREIGN KEY (STOREKEY) REFERENCES[Group10_602-dw-
    area].[dbo].[STORE_DIM](STOREKEY),
    FOREIGN KEY (PRODUCTKEY) REFERENCES [Group10_602-dw-
    area].[dbo].[PRODUCT_DIM](PRODUCTKEY),
    FOREIGN KEY (TIMEKEY) REFERENCES [Group10_602-dw-
    area].[dbo].[TIME_DIM](TIMEKEY)
 )

CREATE TABLE [STORE_PERF_FACT] (
    [STOREKEY] int NOT NULL,
    [PRODUCTKEY] int NOT NULL,
    [PROFIT_VALUE] float,
    [AGE9] varchar(50),
    [AGE60] varchar(50),
    PRIMARY KEY (STOREKEY, PRODUCTKEY),
    FOREIGN KEY (STOREKEY) REFERENCES[Group10_602-dw-
    area].[dbo].[STORE_DIM](STOREKEY),
    FOREIGN KEY (PRODUCTKEY) REFERENCES [Group10_602-dw-
    area].[dbo].[PRODUCT_DIM](PRODUCTKEY)
 )
```

10.15. Staging and Data Mart Tables Screen Shots

- **STORE_DIM**

Results | Messages

	STOREKEY	STORE_NUM	CITY	PRICE_TIER	STORE_ZONE	ZIP_CODE
1	1	2	River Forest	High	1	60305
2	2	4	Park Ridge	Medium	2	60068
3	3	5	Palatine	Medium	2	60067
4	4	8	Oak Lawn	Low	5	60435
5	5	9	Morton Grove	Medium	2	60053
6	6	12	Chicago	High	7	60660
7	7	14	Glenview	High	1	60025
8	8	18	River Grove	Low	5	60171
9	9	21	Hanover Park	CubFighter	6	60103
10	10	28	Mt. Prospect	Medium	2	60054
11	11	32	Park Ridge	High	1	60068
12	12	33	Chicago	High	7	60657
13	13	40	Bridgeview	CubFighter	6	60455
14	14	44	Western Spring	Medium	2	60558

- **PRODUCT_DIM**

Results | Messages

	PRODUCTKEY	UPC_NUMBER	PRODUCT_NAME
1	1	1570077445	CTY LN COLBY MILD
2	2	1570077446	CTY LN CHEDDAR MILD
3	3	1570077454	CTY LN CHEDDAR MED/S
4	4	1570077455	CTY LN CHEDDAR SHARP
5	5	1570077456	CTY LN SWISS OLD WOR
6	6	1570077458	COUNTY LINE/COLBY JA
7	7	1570077501	CTY LN MONTEREY JACK
8	8	1570077503	~COUNTY LINE MUENSTE
9	9	1570077507	~COUNTY LINE MOZZ CH
10	10	1570077508	COUNTY LINE EXTRA SH
11	11	1700310035	KAUKAUNA RANCH BALLS
12	12	1700310039	KAUKAUNA GARDEN VEGT
13	13	1700310062	KAUKAUNA SHARP CHEES
14	14	1700310067	KAUKAUNA SHARP CHED

- **TIME_DIM**

Results Messages

	TIMEKEY	WEEK	MONTH	QUARTER	YEAR
1	1	1	9	3	1989
2	2	2	9	3	1989
3	3	3	9	3	1989
4	4	4	10	4	1989
5	5	5	10	4	1989
6	6	6	10	4	1989
7	7	7	10	4	1989
8	8	8	11	4	1989
9	9	9	11	4	1989
10	10	10	11	4	1989
11	11	11	11	4	1989
12	12	12	11	4	1989
13	13	13	12	4	1989
14	14	14	12	4	1989

- **SALES_FACT**

Results Messages

	TIMEKEY	SALES	STOREKEY	PRODUCTKEY
1	1	16.87	1	1
2	2	19.28	1	1
3	3	9.64	1	1
4	4	19.28	1	1
5	5	16.87	1	1
6	6	12.05	1	1
7	7	28.92	1	1
8	8	20.9	1	1
9	9	25.08	1	1
10	10	35.53	1	1
11	11	6.27	1	1
12	12	21.69	1	1
13	13	24.1	1	1
14	14	24.1	1	1

- **STORE_PERF_FACT**

Results | Messages

	STOREKEY	PRODUCTKEY	PROFIT_VALUE	AGE9	AGE60
1	1	1	1124.529984	0.117508576	0.232864734
2	1	2	1796.089695	0.117508576	0.232864734
3	1	3	1386.434283	0.117508576	0.232864734
4	1	4	2310.607852	0.117508576	0.232864734
5	1	5	3820.317127	0.117508576	0.232864734
6	1	6	1606.248498	0.117508576	0.232864734
7	1	7	1380.462945	0.117508576	0.232864734
8	1	8	52.714589	0.117508576	0.232864734
9	1	9	50.962011	0.117508576	0.232864734
10	1	10	125.967837	0.117508576	0.232864734
11	1	14	25.295745	0.117508576	0.232864734
12	1	15	12.097965	0.117508576	0.232864734
13	1	19	17.116788	0.117508576	0.232864734
14	1	20	60.567096	0.117508576	0.232864734

11. BI Reporting

11.1 Reporting Plan

For each of the business questions chosen for our project, we have made use of SSRS, SSAS and Tableau to generate the reports. The following table shows the reporting and analysis methods corresponding to each of the questions:

BUSINESS QUESTION	REPORTING TOOL USED
Business Question 1	SSRS
Business Question 2	Tableau
Business Question 3	SSAS
Business Question 4	Tableau
Business Question 4	SSRS+SSAS

11.2 Target Report that satisfies Business Questions

BQ 1. Which are the stores that have higher popularity amongst kids and amongst elderly groups of people? (Reporting using SSRS)

For this question, we created a report showing popularity of different DFF stores amongst kids and elders. The report shows popularity by means of percentage of kids (below age 9) and elders (above age 60) visiting the stores. We made use of STORE_DIM table and STORE_PERFORMANCE_FACT tables to derive these values. Store numbers of respective stores are extracted from the STORE_DIM table and corresponding percentage values for visits are derived from the STORE_PERFORMANCE_FACT table. SSRS is used here for the purpose of implementation. By analyzing such information, Retail companies can strategize their marketing tactics and thus can be utilized by their sales and marketing department. For example, DFF can use this information to keep supplies of products which the kids like in the stores which are visited by kids the most. In a similar way, stores that are more popular amongst the elderly can have more supplies of products which are famous among elderly people. This will be beneficial for them and will aid in increasing their sales.

BQ 2. Which stores have profit below the average profit margin of Cereal (WCER)? (Reporting using Tableau)

For this question, we plotted a graph showing average profit for Cereal of each of the DFF's stores. On the Y axis of the plot, we have average profit and on the X axis, we have different store numbers. PRODUCT_DIM, STORE_DIM and STORE_PERFORMANCE_FACT tables were used in the creation of this plot. We are getting respective store numbers from the STORE_DIM table and we have applied filter on PRODUCT_DIM such that the product type is Cereal. The value of profit is then extracted from the STORE_PERFORMANCE_FACT table. Tableau has been used in the creation of this report. This data can be used by the company to analyze performance of different stores and eventually modify the current strategies to improve profitability of different stores. In this case, we are analyzing the average profit earned across different stores for cereals. This data will help us in understanding the top performers and the poor performers.

BQ 3. How have the sales of frozen products varied over quarters in a particular year? (Reporting using SSAS)

For this question, we created a report showing quarterly sales of frozen products. This was achieved by analyzing the cube created using SALES_FACT table as the measure table and PRODUCT_DIM and TIME_DIM as the dimensions. The cube was then deployed at the SSAS server. Filter was applied on the PRODUCT_DIM table such that the product type was Frozen. The sales values were then obtained using SALES_FACT table grouped by quarter using the TIME_DIM. This helps in analyzing how were the sales across the different years in different quarters. This can help management in analyzing if there are certain factors which led to growth/decline in sales of the frozen product in different quarters over the years. Further, the analysis of the sales trends can help in forecasting demand in different time periods as well. Hence, quarter analysis helps in finding underlying trends in the sales and reasons associated with that.

BQ 4. How is the demand for a particular product (Dairy) changing in different price tiers? (Reporting using Tableau)

In this question, we have created a plot showing the sales of Dairy products across different price tiers (High, Medium, Low, Cub Fighter). The graph on the Y axis displays the sales values and on the X axis it displays week numbers. We have made use of PRODUCT_DIM, TIME_DIM, STORE_DIM and SALES_FACT table to create this report. Filter is applied on the PRODUCT_DIM table such that the product type is Dairy and then sales values for different price tiers on a weekly basis are obtained using SALES_FACT, STORE_DIM and TIME_DIM tables respectively. We can use this question to analyze various price tiers as well to develop a trend of products and their movement in the stores. This will help DFF to manage their inventory as well as target customers accordingly.

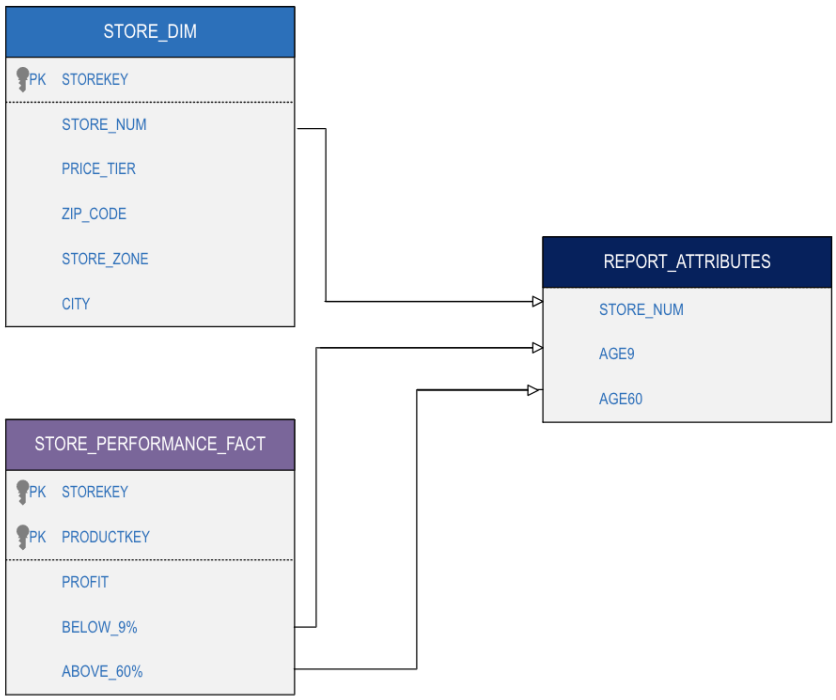
BQ 5. What is the store wise average profit of particular product category (Grooming)? (Reporting using SSRS + SSAS)

For this question, we have created a report displaying store-wise average profit values for Grooming products and the data is then analyzed using a bar graph. We have made use of

PRODUCT_DIM, STORE_DIM and STORE_PERFORMANCE_FACT tables. We first created a cube in SSAS using the STORE_PERFORMANCE_FACT table as measure and PRODUCT_DIM and STORE_DIM tables as dimensions. We then deployed the cube on the SSAS server. After this, we created a report showing store wise average profit on SSRS by taking average of profit values and applying filter on PRODUCT_DIM such that product type was Grooming products. We then created a bar graph in SSRS to carry on the further analysis. All businesses aim to maximize their profits and thus understanding products and stores which are profitable will help the higher management pick out the products in stores which are not performing well. The management can then take decision whether to plan a strategy to improve performance of these products or to discontinue the products. This question will also help DFF cut their losses. This will ultimately lead to more revenue for the entire retail chain.

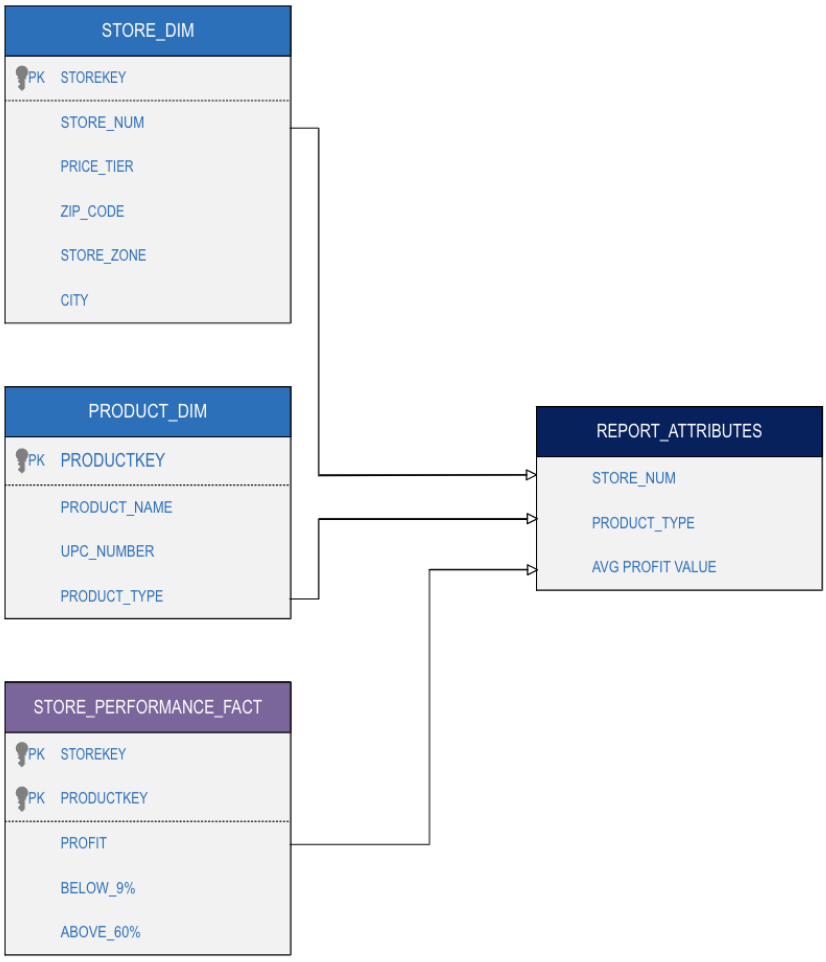
11.3 Mapping from Independent Data Marts to Report Attributes

BQ 1. Which are the stores that have higher popularity amongst kids and amongst elderly groups of people?



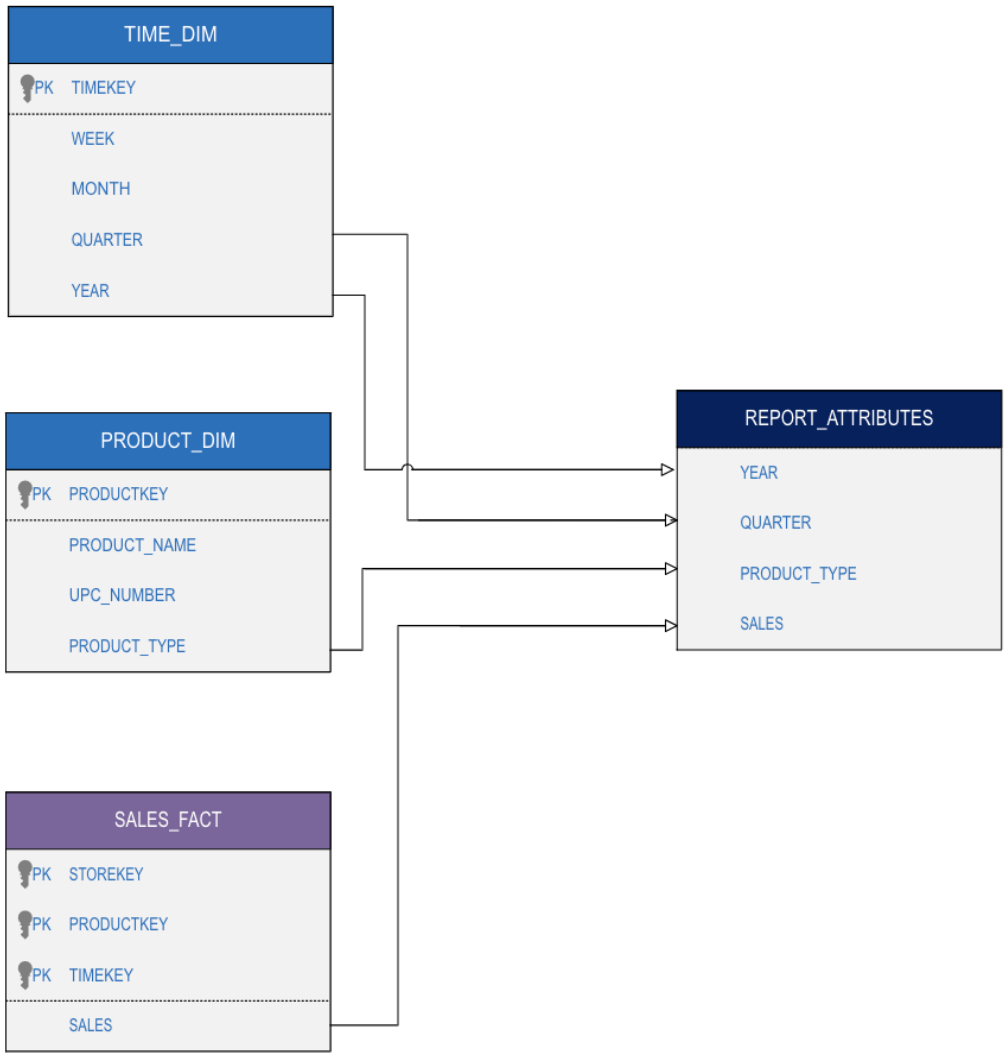
Source Table	Attribute Name	Report Attribute
STORE_DIM	Store_Num	Store_Num
STORE_PERFORMANCE_FACT	Below_9%	Age9
STORE_PERFORMANCE_FACT	Above_60%	Age60

BQ 2. Which stores have profit below the average profit margin of Cereal (WCER)?



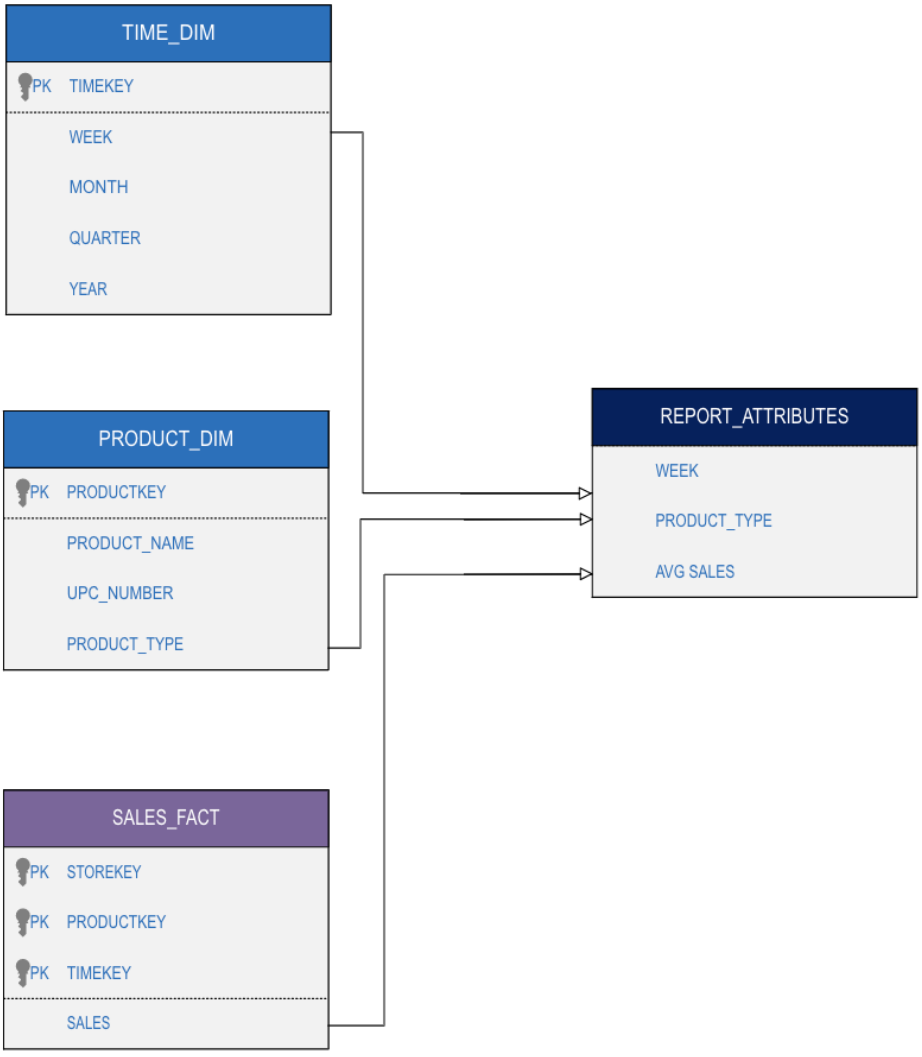
Source Table	Attribute Name	Report Attribute
STORE_DIM	Store_Num	Store_Num
PRODUCT_DIM	Product_Type	Product_Type
STORE_PERFORMANCE_FACT	Profit	Avg Profit Value

BQ 3. How have the sales of frozen products varied over quarters in a particular year?



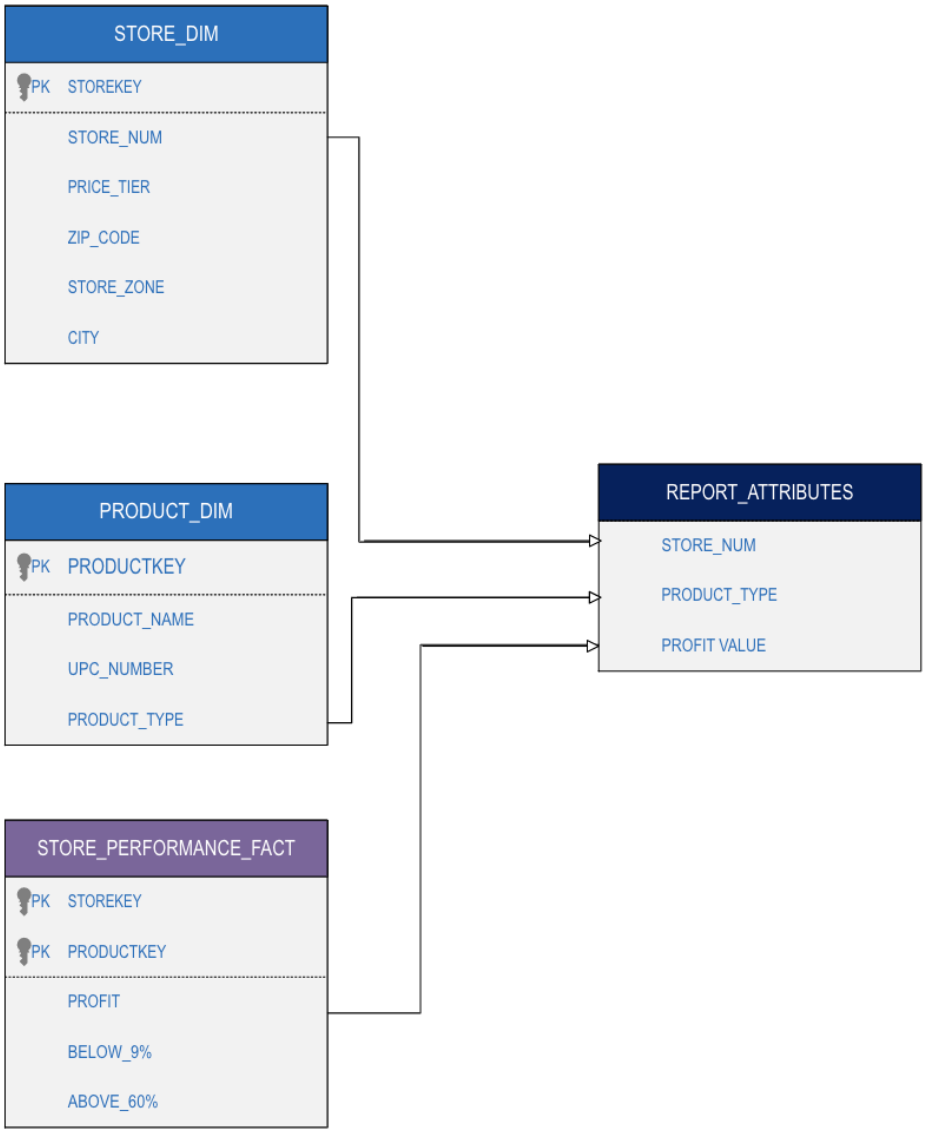
Source Table	Attribute Name	Report Attribute
TIME_DIM	Year	Year
TIME_DIM	Quarter	Quarter
PRODUCT_DIM	Product_Type	Product_Type
SALES_FACT	Sales	Sales

BQ 4. How is the demand for a particular product (Dairy) changing in different price tiers?



Source Table	Attribute Name	Report Attribute
TIME_DIM	Week	Week
PRODUCT_DIM	Product_Type	Product_Type
SALES_FACT	Sales	Avg Sales

BQ 5. What is the store wise average profit of particular product category (Grooming)?

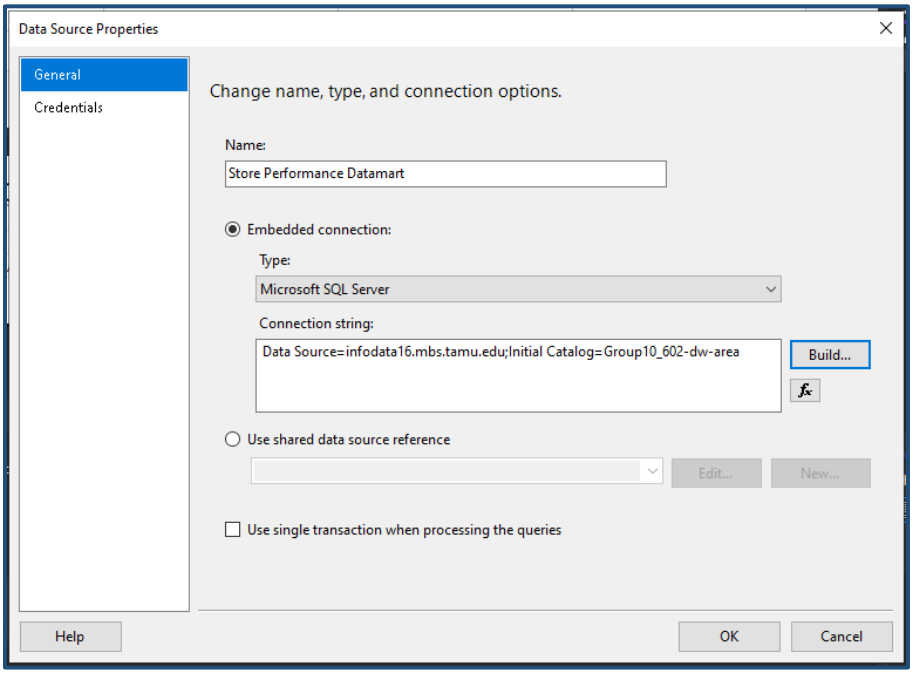


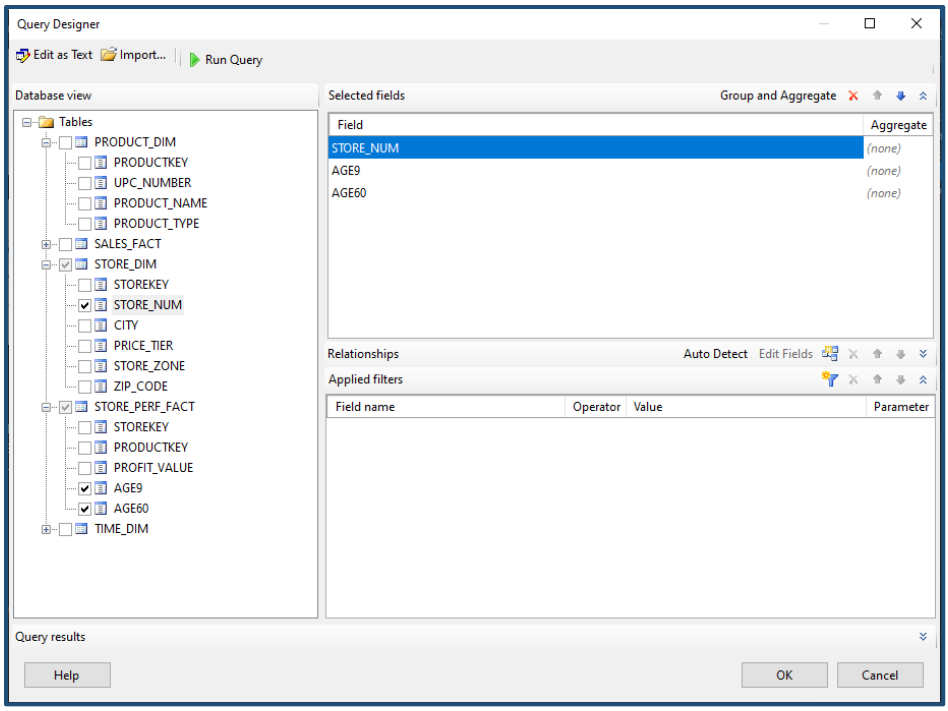
Source Table	Attribute Name	Report Attribute
STORE_DIM	Store_Num	Store_Num
PRODUCT_DIM	Product_Type	Product_Type
STORE_PERFORMANCE_FACT	Profit	Profit Value

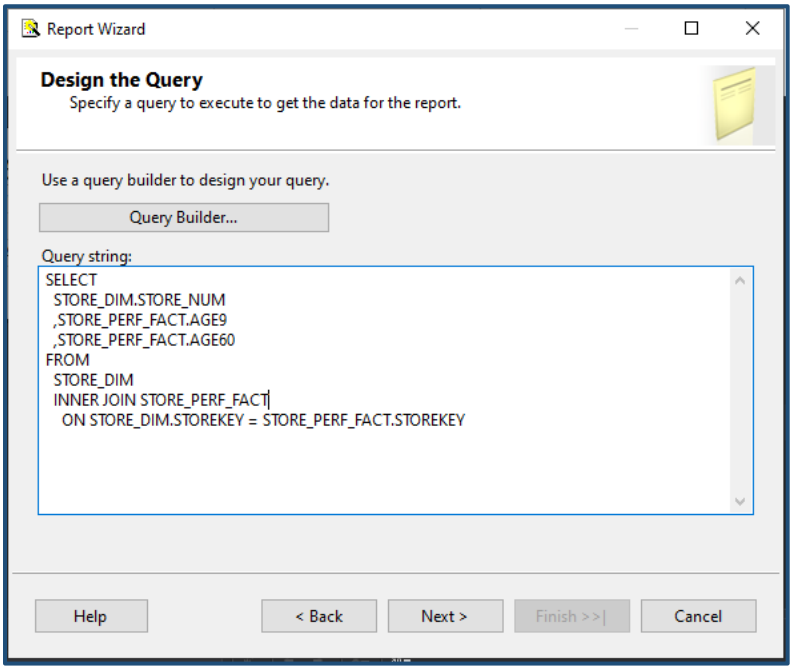
11.4 Generated Reports using SSRS for Business Question 1

BQ 1. Which are the stores that have higher popularity amongst kids and amongst elderly groups of people? (Reporting using SSRS)

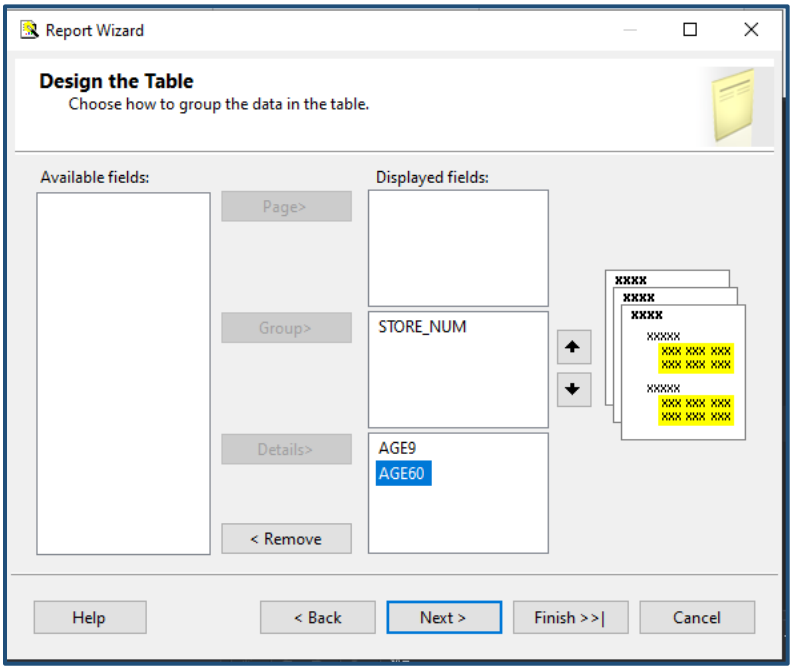
To answer question 1, we are generating a report using SSRS by using the Store Performance data mart and then deploying the project to the SSRS at the server side.

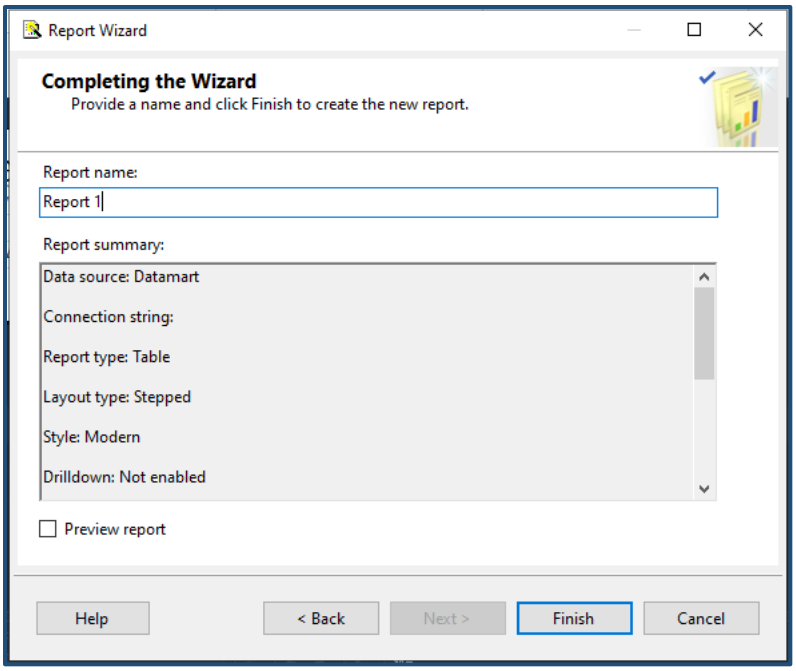


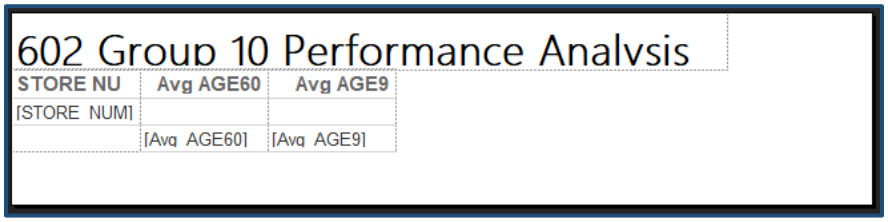

Query Designer
Edit as Text
Import...
Run Query
Database view
Tables
PRODUCT_DIM
PRODUCTKEY
UPC_NUMBER
PRODUCT_NAME
PRODUCT_TYPE
SALES_FACT
STORE_DIM
STOREKEY
STORE_NUM
CITY
PRICE_TIER
STORE_ZONE
ZIP_CODE
STORE_PERF_FACT
STOREKEY
PRODUCTKEY
PROFIT_VALUE
AGE9
AGE60
TIME_DIM
Selected fields
Group and Aggregate
Field
Aggregate
STORE_NUM
(none)
AGE9
(none)
AGE60
(none)
Relationships
Auto Detect
Edit Fields
Applied filters
Field name
Operator
Value
Parameter
Query results
Help
OK
Cancel

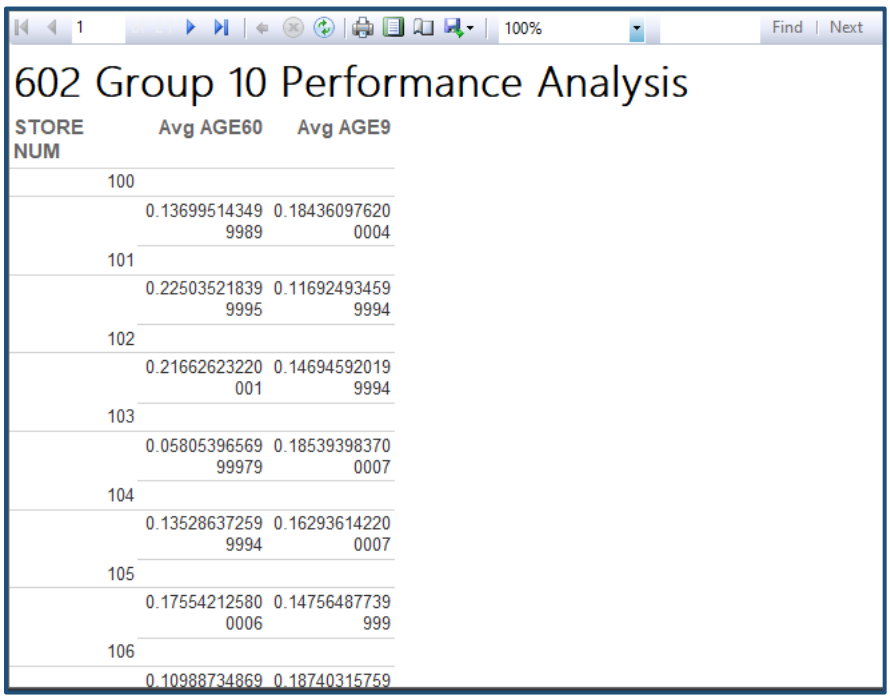

Report Wizard
Design the Query
Specify a query to execute to get the data for the report.
Use a query builder to design your query.
Query Builder...
Query string:
SELECT
STORE_DIM.STORE_NUM
,STORE_PERF_FACT.AGE9
,STORE_PERF_FACT.AGE60
FROM
STORE_DIM
INNER JOIN STORE_PERF_FACT
ON STORE_DIM.STOREKEY = STORE_PERF_FACT.STOREKEY
Help
< Back
Next >
Finish >>|
Cancel

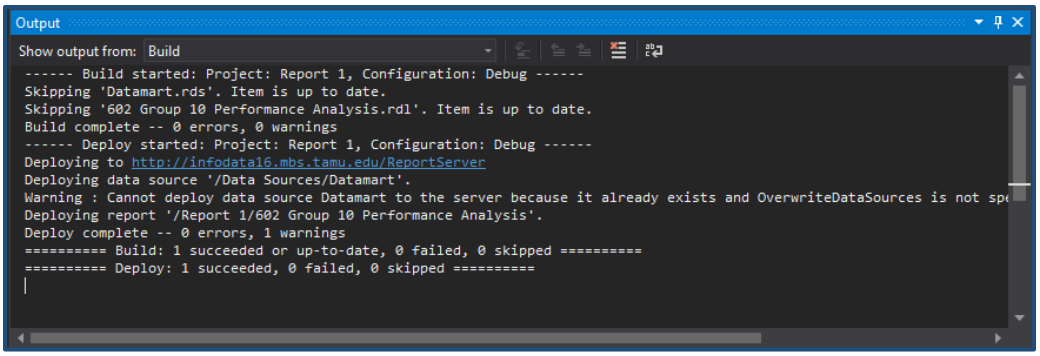
```
SELECT
        STORE_DIM.STORE_NUM
       ,STORE_PERF_FACT.AGE9
       ,STORE_PERF_FACT.AGE60
FROM
        STORE_DIM
        INNER JOIN STORE_PERF_FACT
        ON STORE_DIM.STOREKEY = STORE_PERF_FACT.STOREKEY
```

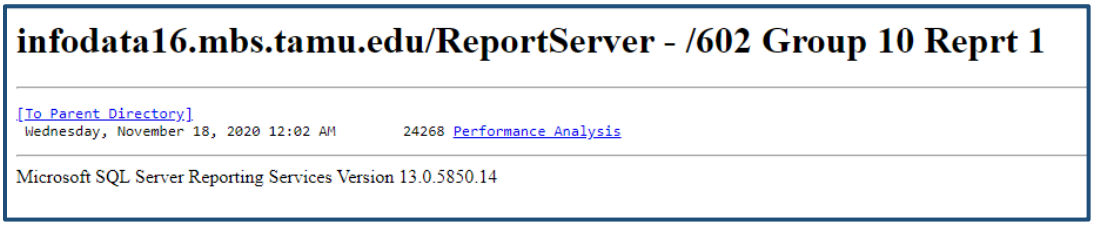


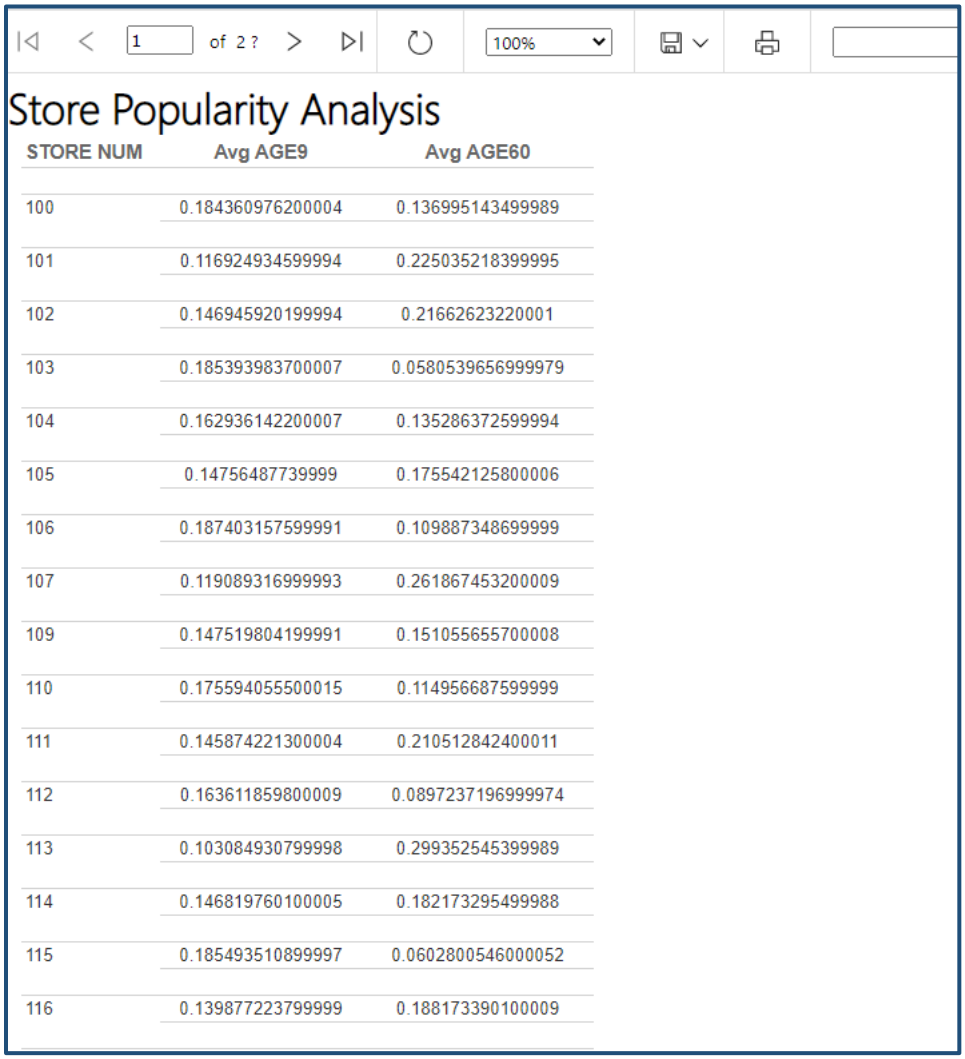

Report Wizard
Completing the Wizard
Provide a name and click Finish to create the new report.
Report name:
Report 1
Report summary:
Data source: Datamart
Connection string:
Report type: Table
Layout type: Stepped
Style: Modern
Drilldown: Not enabled
Preview report
Help
< Back
Next >
Finish
Cancel


602 Group 10 Performance Analysis
STORE NU
Avg AGE60
Avg AGE9
[STORE_NUM]
[Avg_AGE60]
[Avg_AGE9]


1
100%
Find
Next
602 Group 10 Performance Analysis
STORE NUM
Avg AGE60
Avg AGE9
100
0.136995143499989
0.184360976200004
101
0.225035218399995
0.116924934599994
102
0.216626232200001
0.146945920199994
103
0.058053965699979
0.185393983700007
104
0.135286372599994
0.162936142200007
105
0.175542125800006
0.14756487739999
106
0.10988734869
0.18740315759







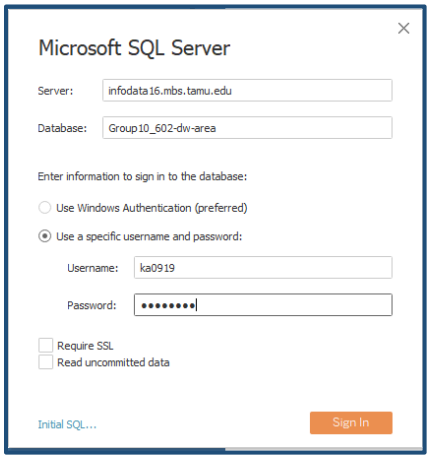
STORE NUM	Avg AGE9	Avg AGE60
100	0.184360976200004	0.136995143499989
101	0.116924934599994	0.225035218399995
102	0.146945920199994	0.21662623220001
103	0.185393983700007	0.0580539656999979
104	0.162936142200007	0.135286372599994
105	0.14756487739999	0.175542125800006
106	0.187403157599991	0.109887348699999
107	0.119089316999993	0.261867453200009
109	0.147519804199991	0.151055655700008
110	0.175594055500015	0.114956687599999
111	0.145874221300004	0.210512842400011
112	0.163611859800009	0.0897237196999974
113	0.103084930799998	0.299352545399989
114	0.146819760100005	0.182173295499988
115	0.185493510899997	0.0602800546000052
116	0.139877223799999	0.188173390100009

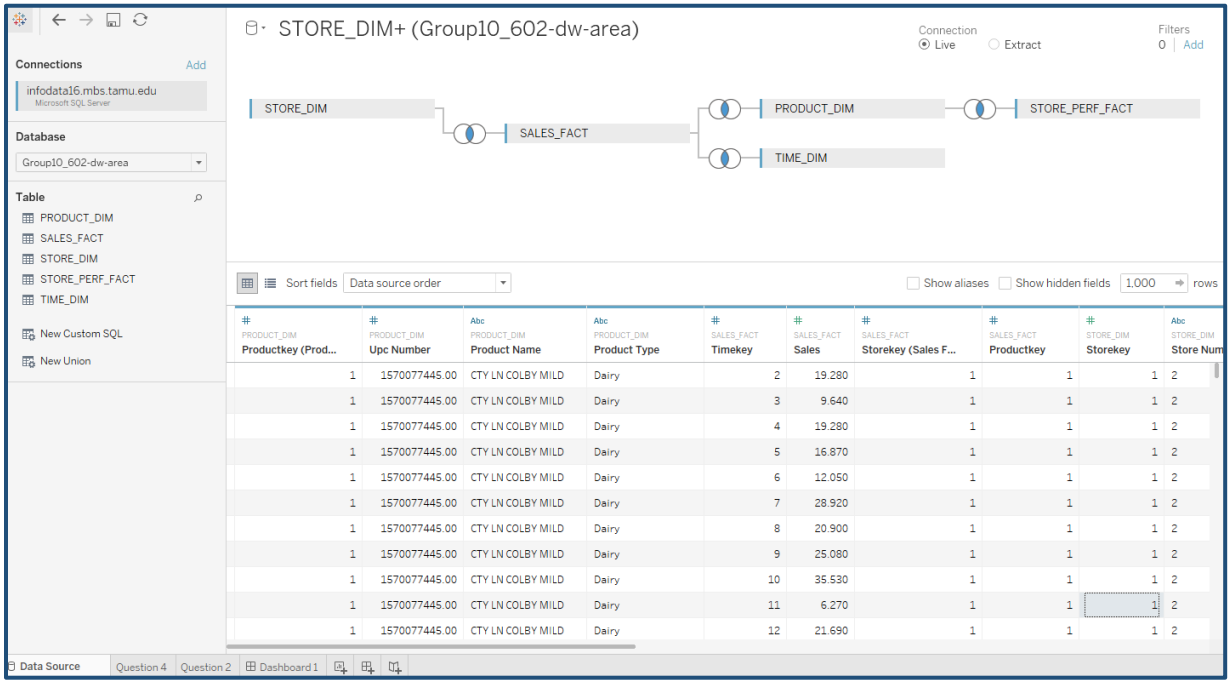
Result: The results above show that the popularity of different stores by means of percentage of kids (below age 9) and elders (above age 60) visiting the stores. We can note that the most popular store amongst kids is store 106 while the most popular store amongst elderly is 113. Hence, business should update their inventory based on the most important demographics for them.

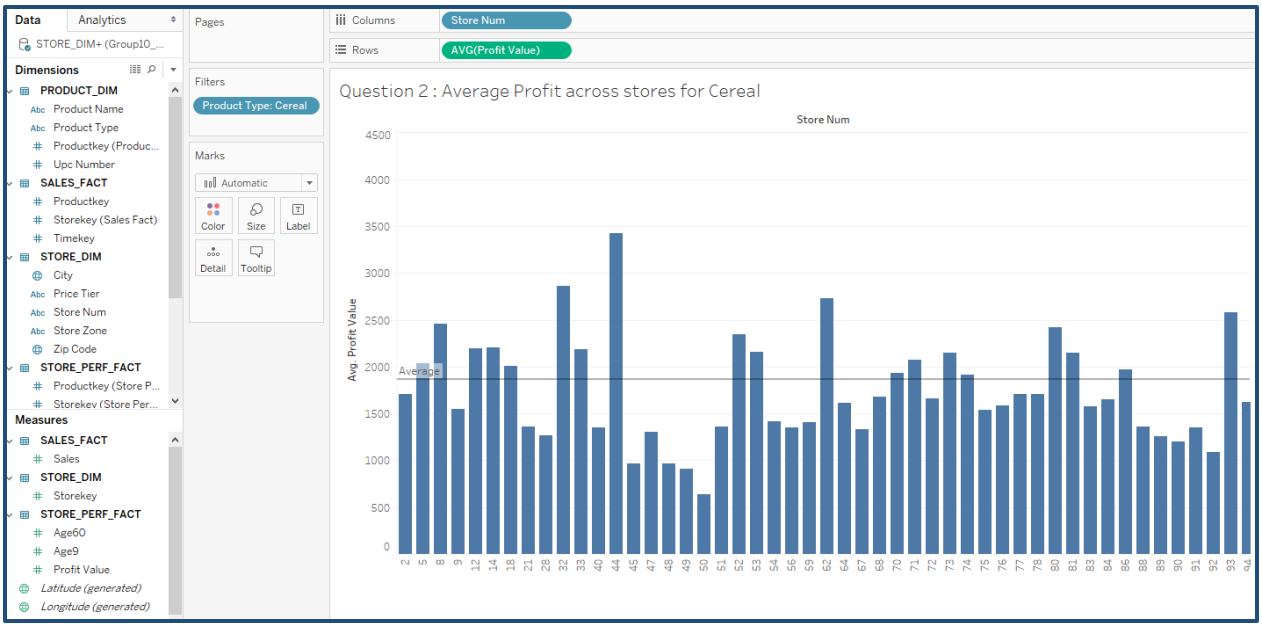
11.5 Generated Reports using Tableau for Business Question 2

BQ 2. Which stores have profit below the average profit margin of Cereal (WCER)? (Reporting using Tableau)

For this question, we are using Tableau along with Store Performance datamart to plot average profit of stores for cereals.





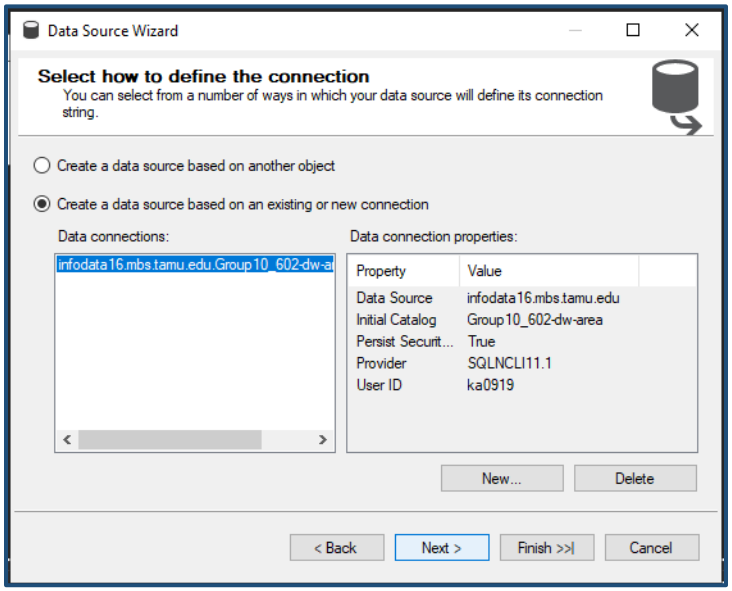


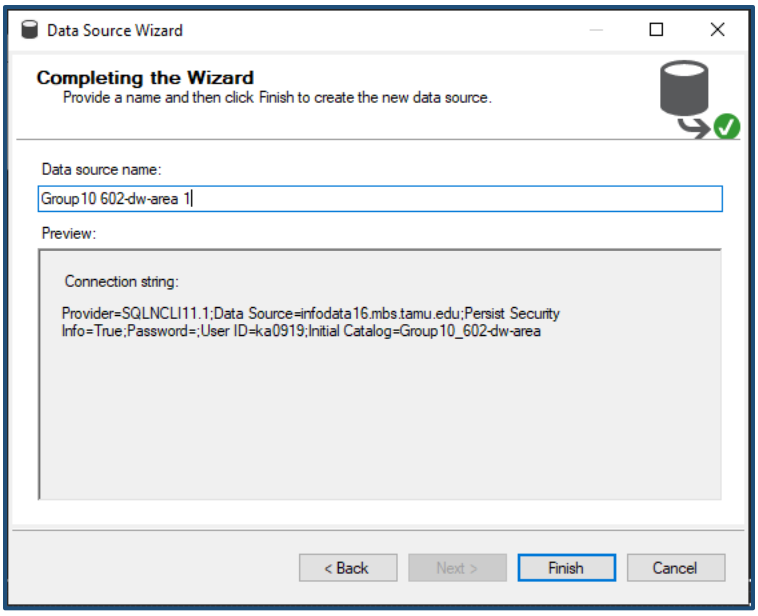
Result: The above results show the average profit value for cereals across different stores using bar chart. There is a reference line in the report stores which implies the average profit value for cereals. Thus, all the stores whose bars lie above the reference line have their average store profit above the overall average store profit for cereal category.

11.6 Generated Reports using SSAS for Business Question 3

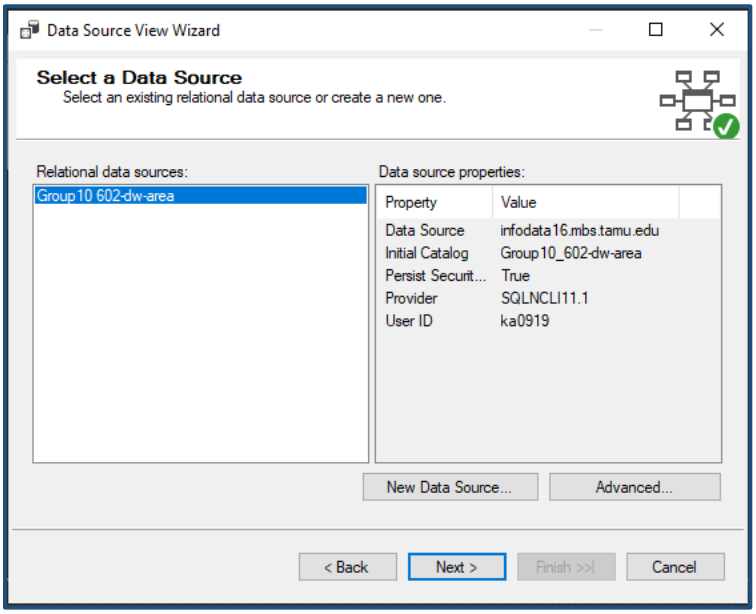
BQ 3. How have the sales of frozen products varied over quarters in a particular year? (Reporting using SSAS)

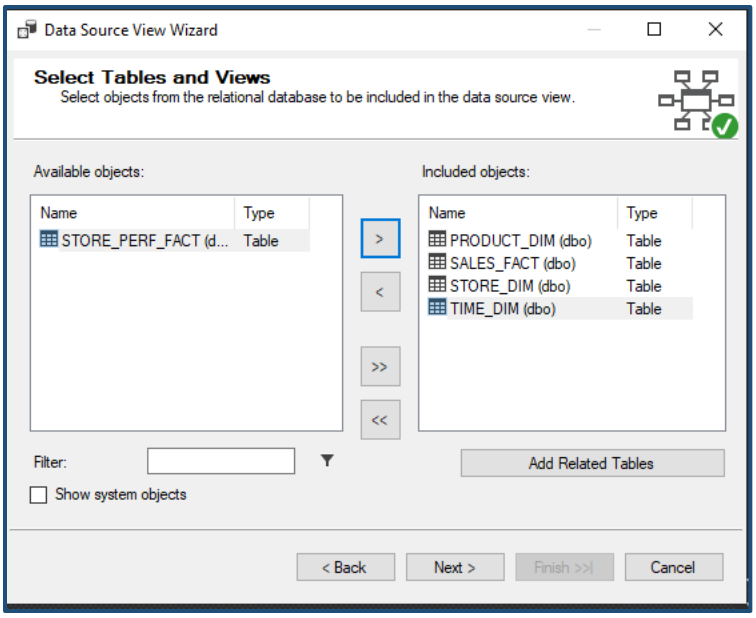
Create a connection

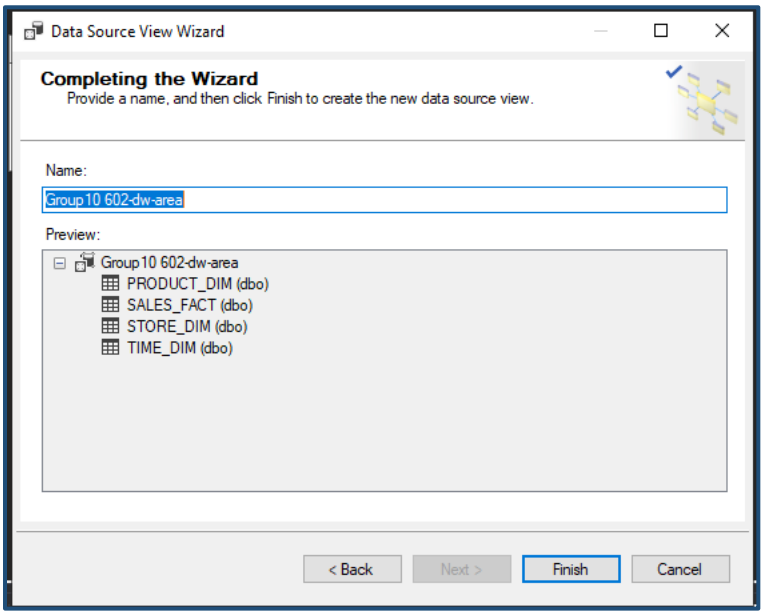




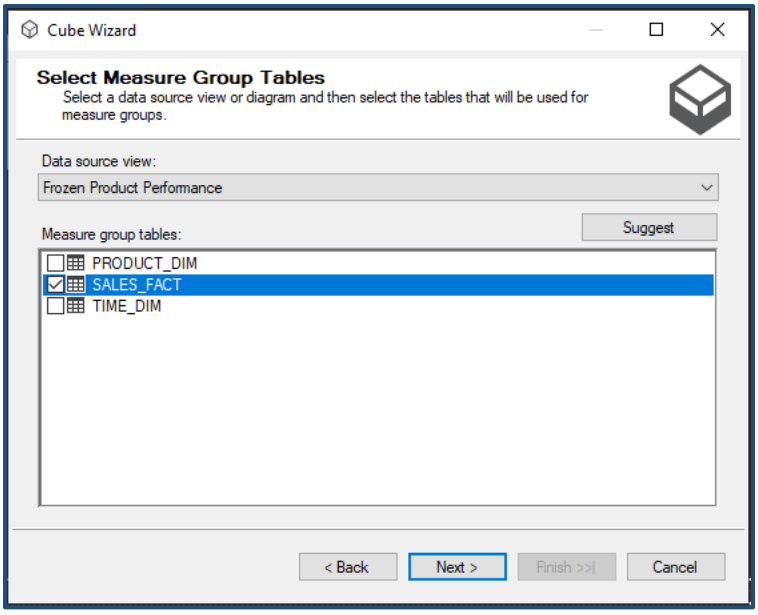
Create Data Source View

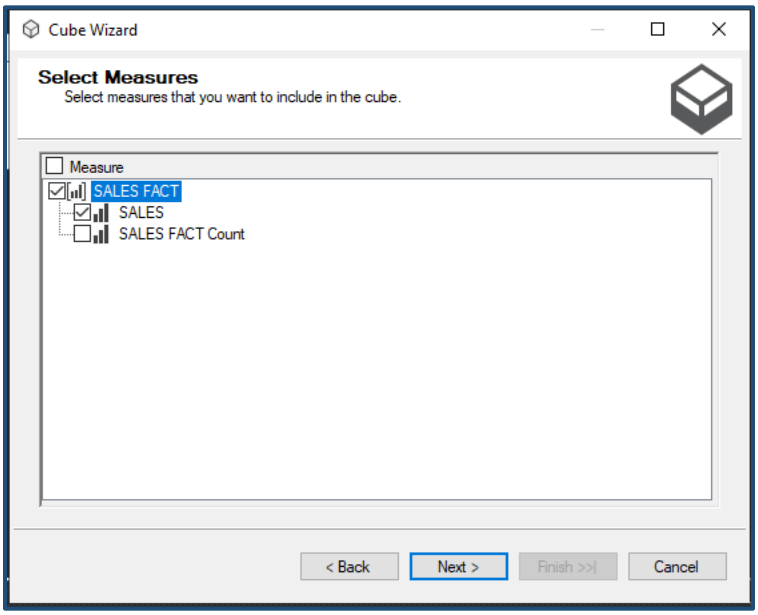


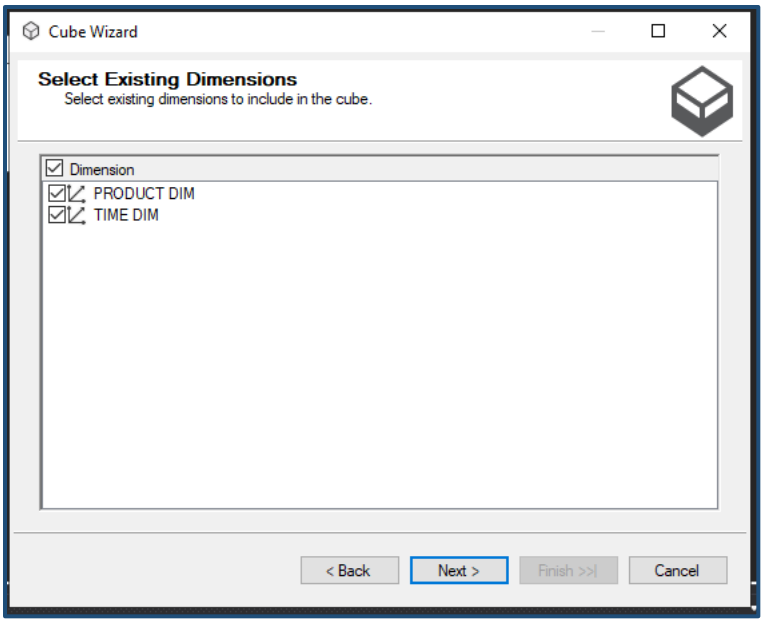

Data Source View Wizard
Select Tables and Views
Select objects from the relational database to be included in the data source view.
Available objects:
Name
Type
STORE_PERF_FACT (d...
Table
Included objects:
Name
Type
PRODUCT_DIM (dbo)
Table
SALES_FACT (dbo)
Table
STORE_DIM (dbo)
Table
TIME_DIM (dbo)
Table
>
<
>>
<<
Filter:
Add Related Tables
Show system objects
< Back
Next >
Finish >>|
Cancel

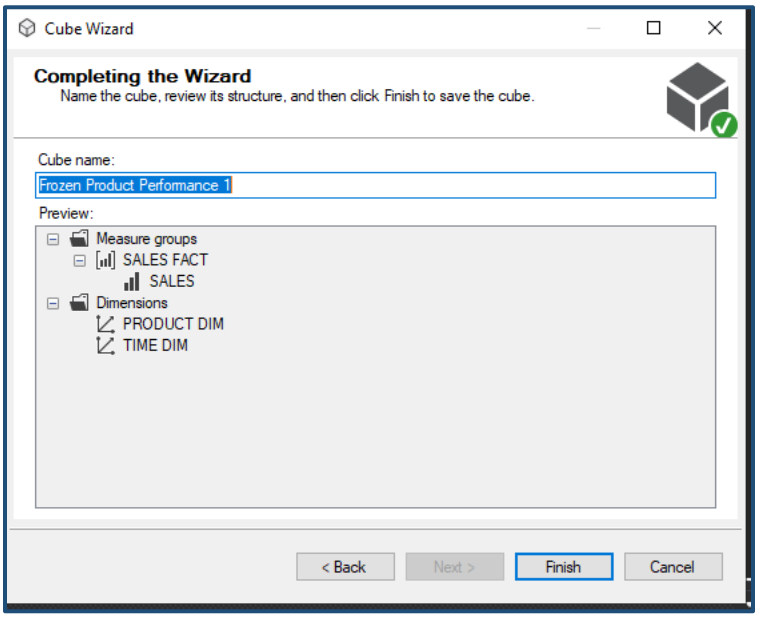

Data Source View Wizard
Completing the Wizard
Provide a name, and then click Finish to create the new data source view.
Name:
Group10 602-dw-area
Preview:
Group10 602-dw-area
PRODUCT_DIM (dbo)
SALES_FACT (dbo)
STORE_DIM (dbo)
TIME_DIM (dbo)
< Back
Next >
Finish
Cancel

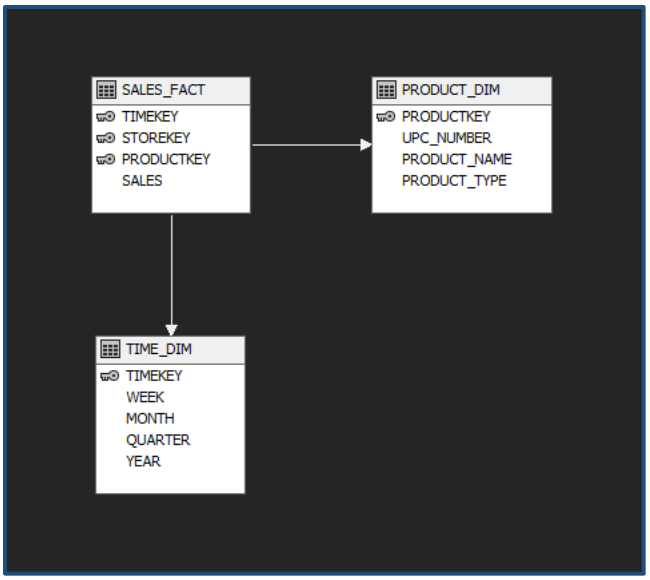
Create the cube



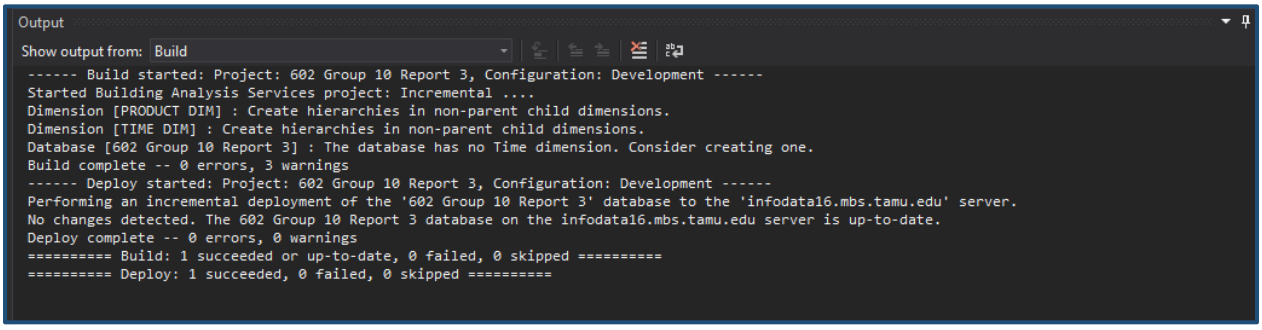



Cube Wizard
Select Existing Dimensions
Select existing dimensions to include in the cube.
Dimension
PRODUCT DIM
TIME DIM
< Back
Next >
Finish >>|
Cancel

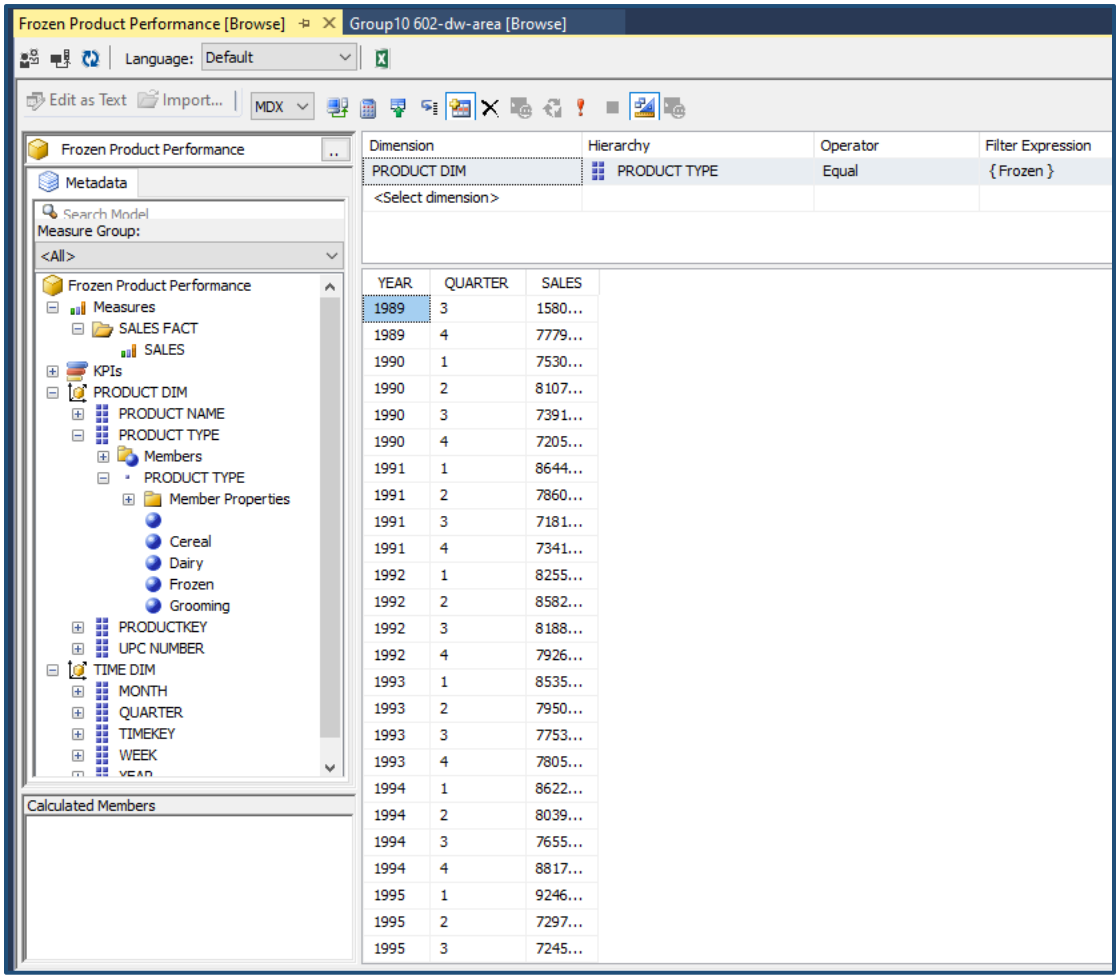

Cube Wizard
Completing the Wizard
Name the cube, review its structure, and then click Finish to save the cube.
Cube name:
Frozen Product Performance 1
Preview:
Measure groups
SALES FACT
SALES
Dimensions
PRODUCT DIM
TIME DIM
< Back
Next >
Finish
Cancel



Deploying the Cube on Analysis Services



Browsing the Cube

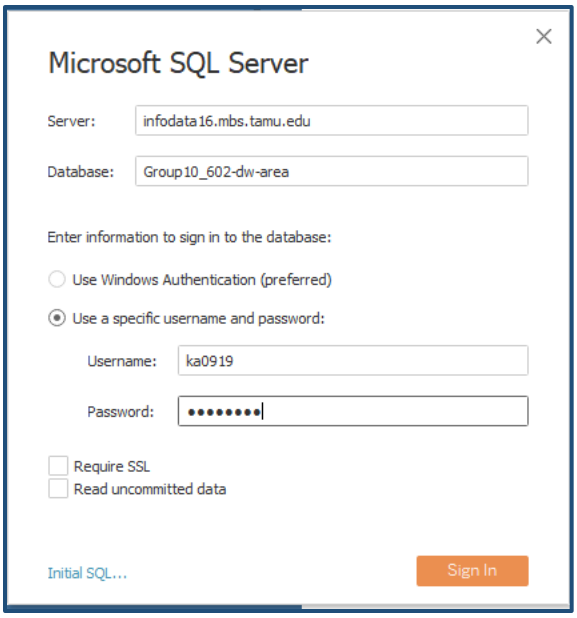


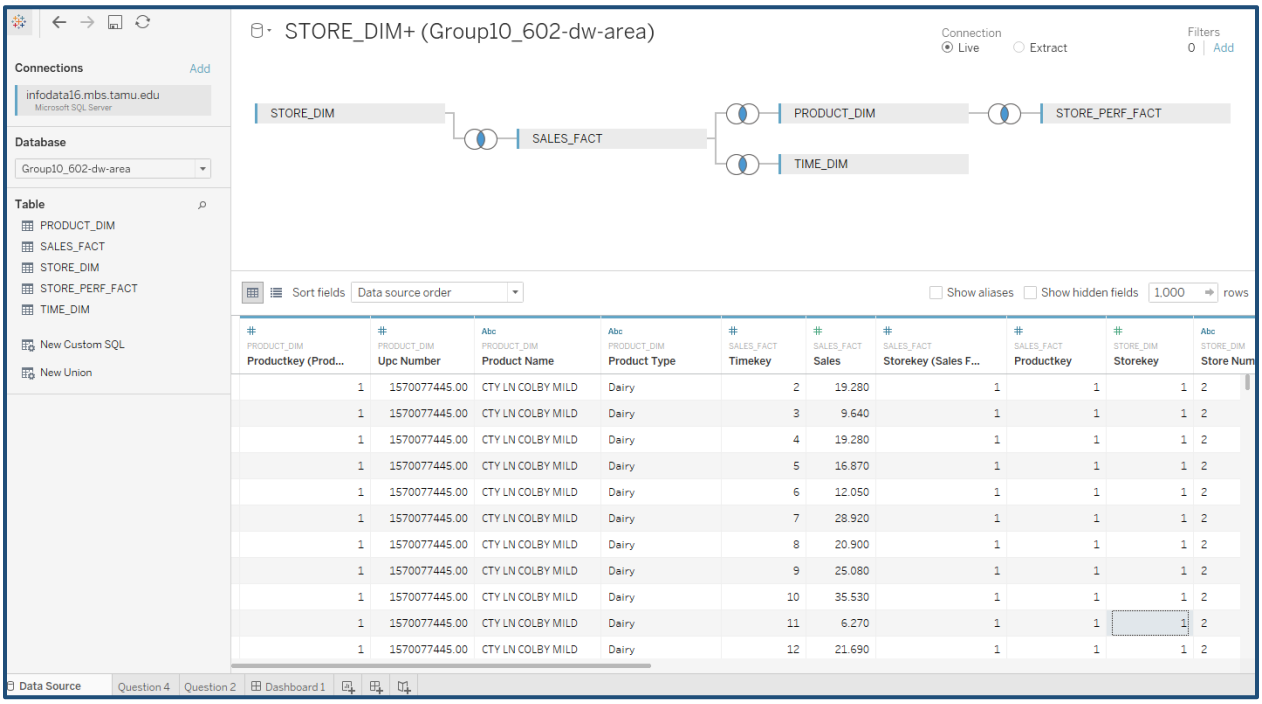
Result: The above results of the cube show the total sales for frozen products which took place in different years across different quarters. This is a view which can be easily accessed by the management to analyze the quarterly sales and compare sales of different quarters to manage their strategies.

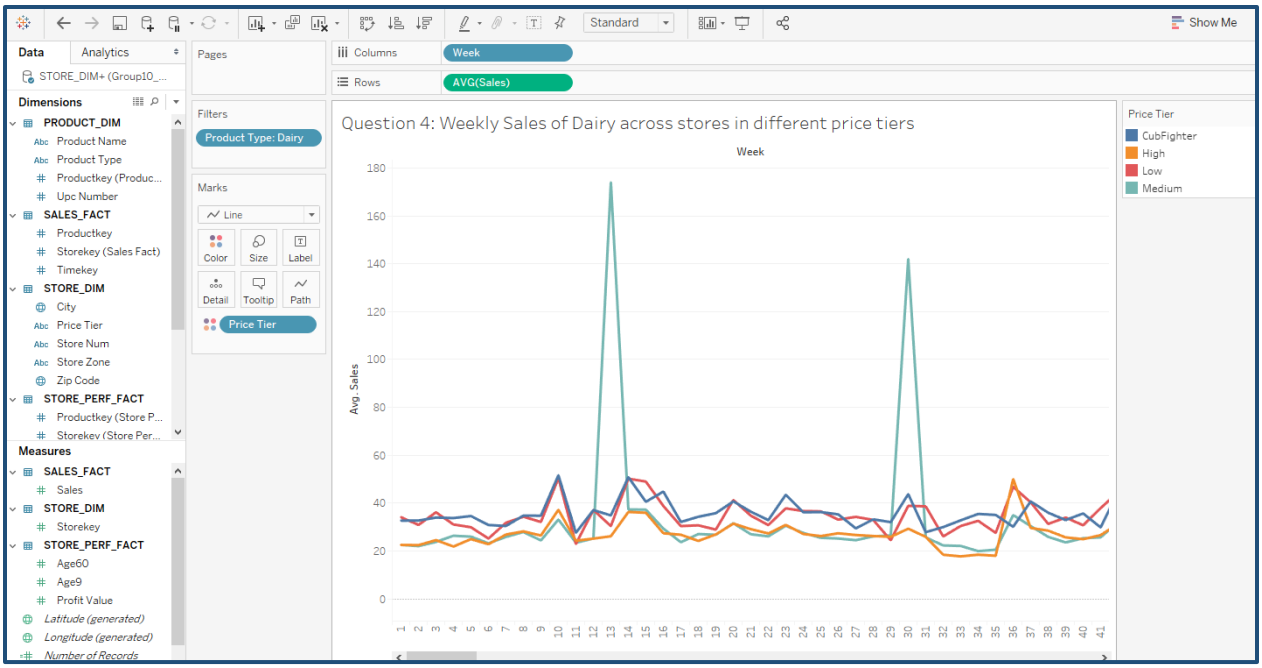
11.7 Generated Reports using Tableau for Business Question 4

BQ 4. How is the demand for a particular product (Dairy) changing in different price tiers? (Reporting using Tableau)

For this question, we are using Tableau along with Sales datamart to plot sales of Dairy products in different price tiers.


Microsoft SQL Server
Server:
infodata16.mbs.tamu.edu
Database:
Group10_602-dw-area
Enter information to sign in to the database:
Use Windows Authentication (preferred)
Use a specific username and password:
Username:
ka0919
Password:
Require SSL
Read uncommitted data
Initial SQL...
Sign In


STORE_DIM+ (Group10_602-dw-area)
Connection
Live
Extract
Filters
0 | Add
Connections
Add
infodata16.mbs.tamu.edu
Microsoft SQL Server
Database
Group10_602-dw-area
Table
PRODUCT_DIM
SALES_FACT
STORE_DIM
STORE_PERF_FACT
TIME_DIM
New Custom SQL
New Union
STORE_DIM
SALES_FACT
PRODUCT_DIM
STORE_PERF_FACT
TIME_DIM
Sort fields
Data source order
Show aliases
Show hidden fields
1,000
rows
Productkey (Prod...
Upc Number
Product Name
Product Type
Timekey
Sales
Storekey (Sales F...
Productkey
Storekey
Store Num
1 1570077445.00 CTY LN COLBY MILD Dairy 2 19.280 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 3 9.640 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 4 19.280 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 5 16.870 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 6 12.050 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 7 28.920 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 8 20.900 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 9 25.080 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 10 35.530 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 11 6.270 1 1 1 2
1 1570077445.00 CTY LN COLBY MILD Dairy 12 21.690 1 1 1 2
Data Source
Question 4
Question 2
Dashboard 1

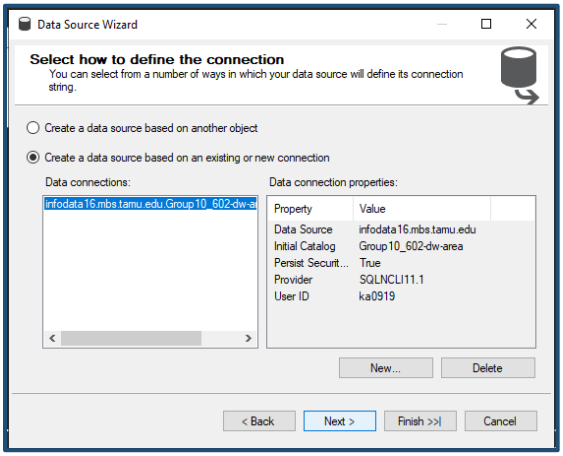


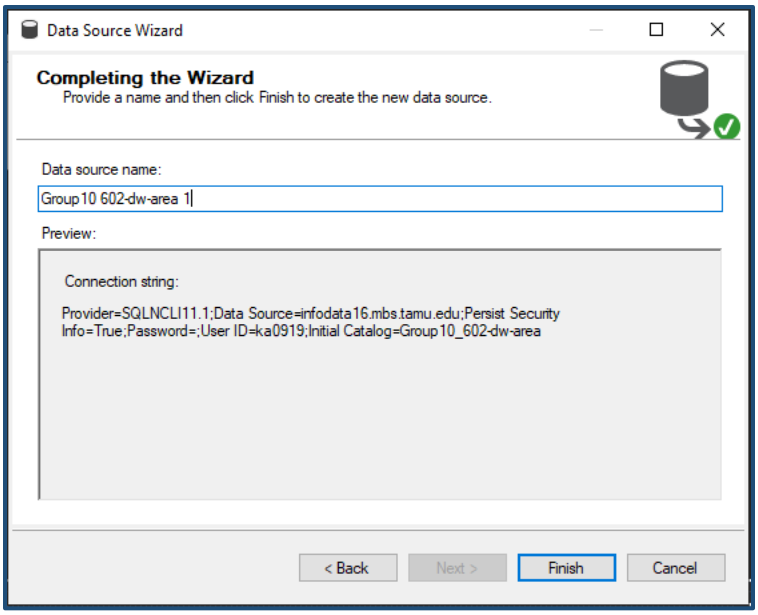
Result: The above results show how the weekly sales of dairy took place in different stores in different price tiers using a line chart. We can observe from the results above that there is a spike in the sales of medium price tier category in a few stores. However, the sales of all the other price tiers are following a similar trend which is evident from the line chart above.

11.8 Generated Reports using SSAS+SSRS for Business Question 5

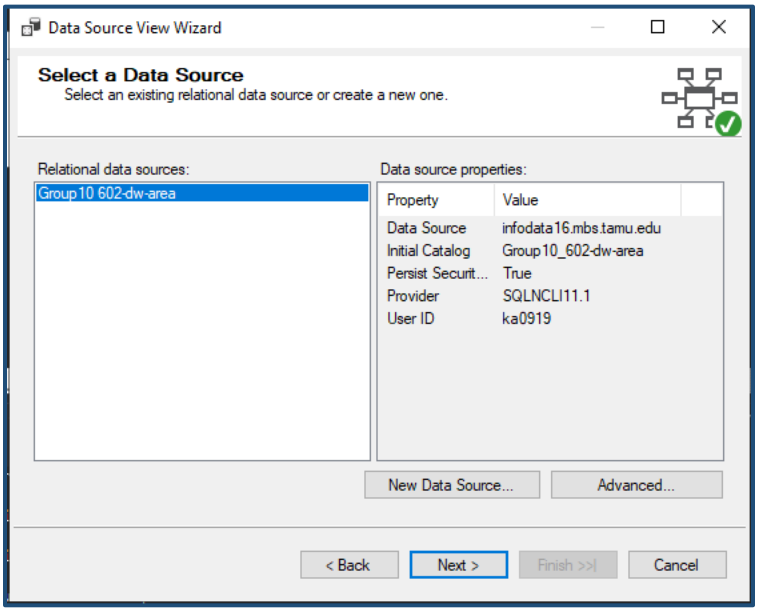
BQ 5. What is the store wise average profit of particular product category (Grooming)? (Reporting using SSAS + SSRS)

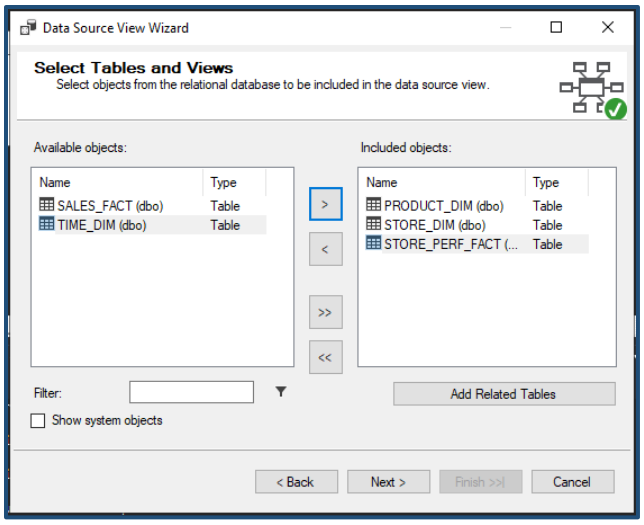
Create a connection

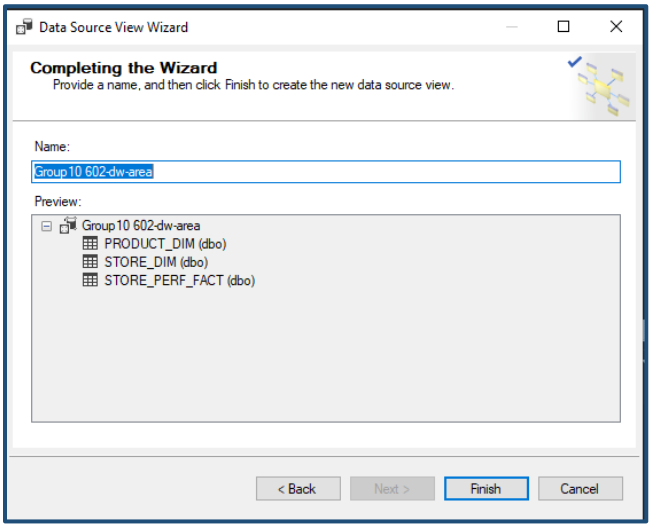




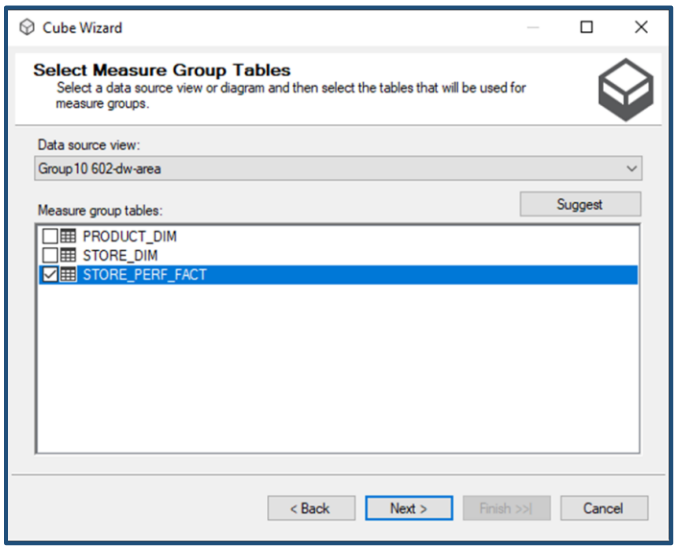
Create Data Source View

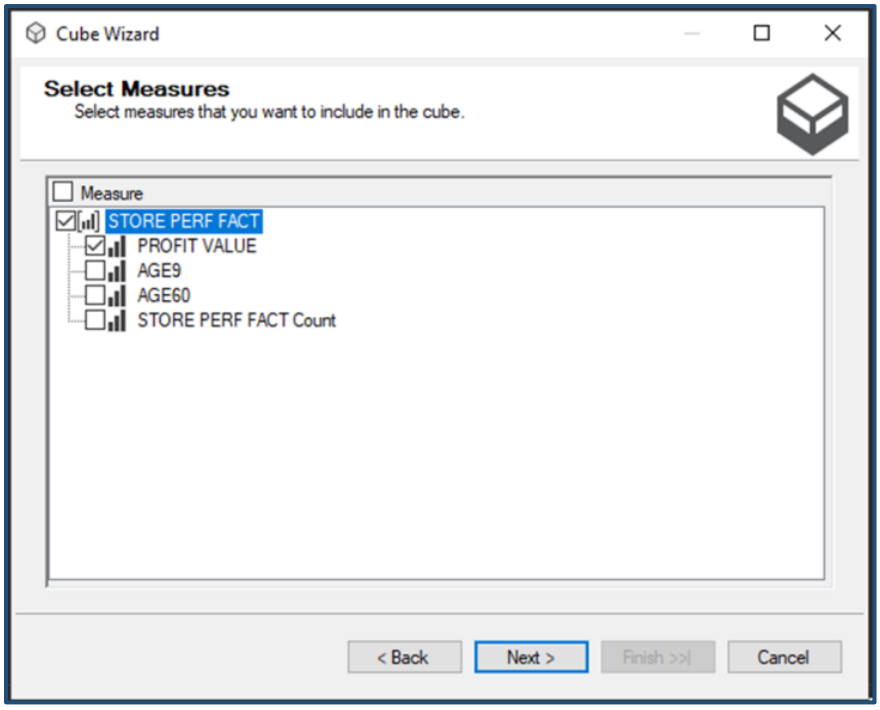


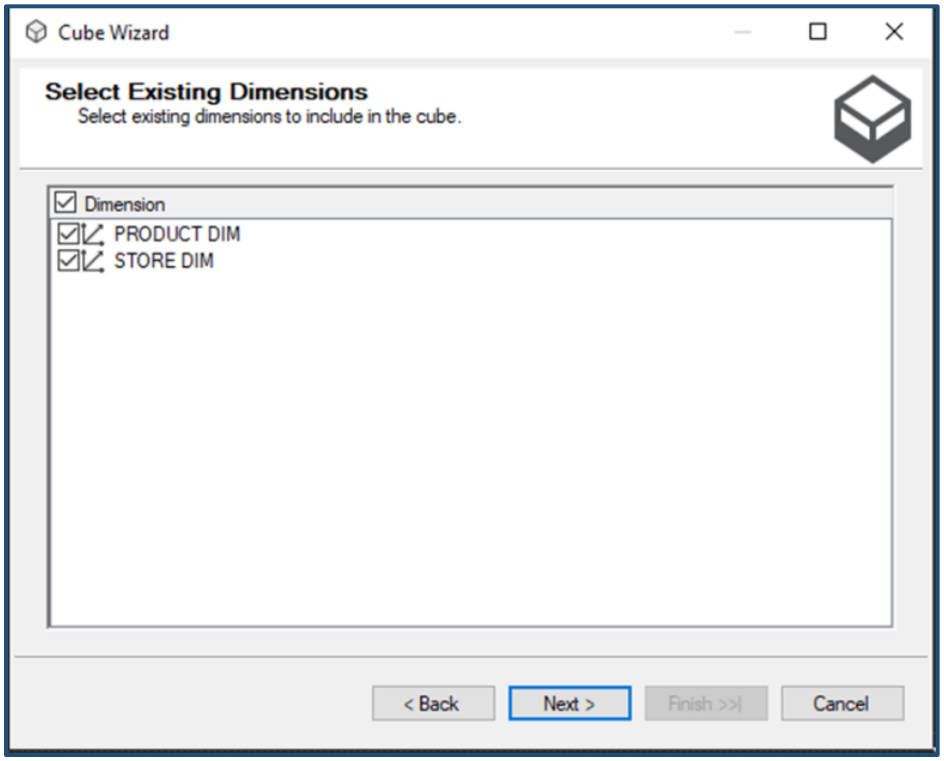


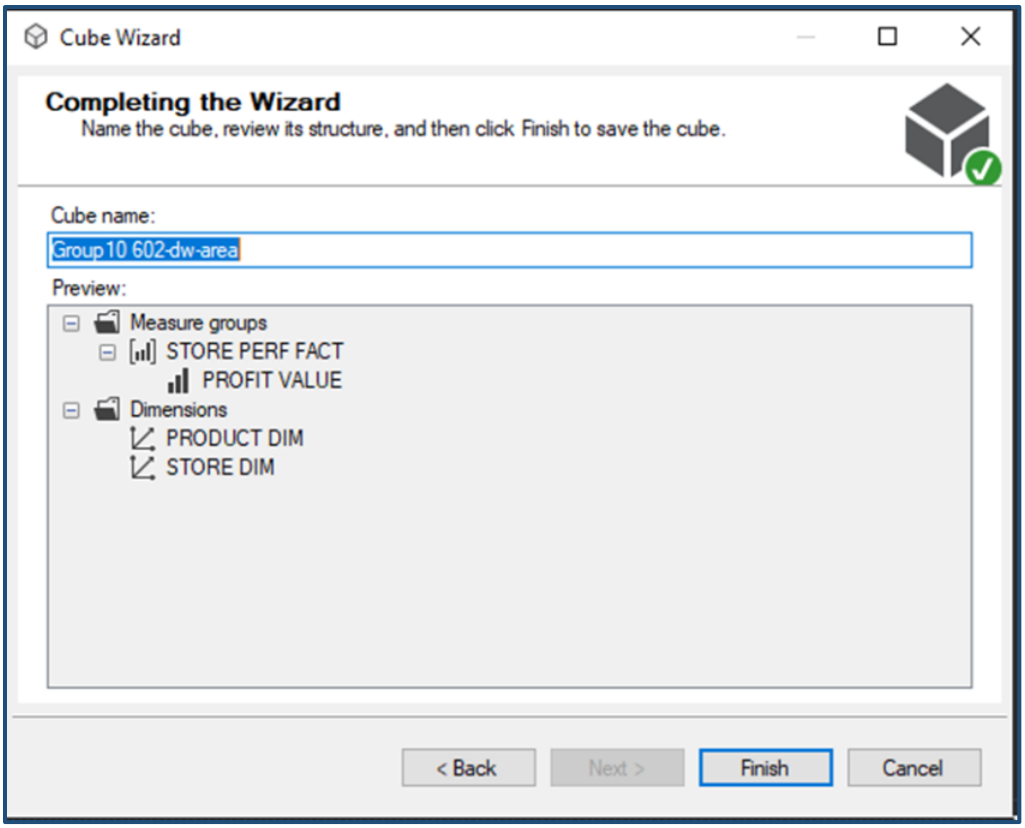


Creating the cube:

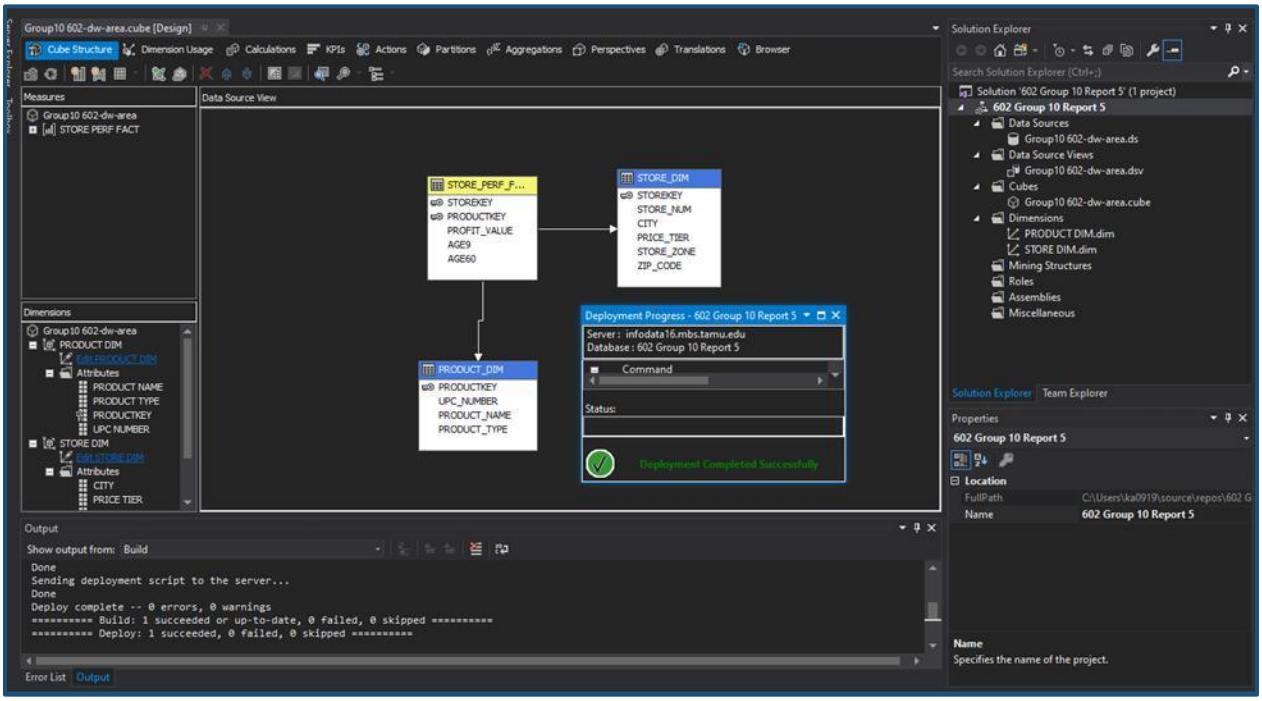


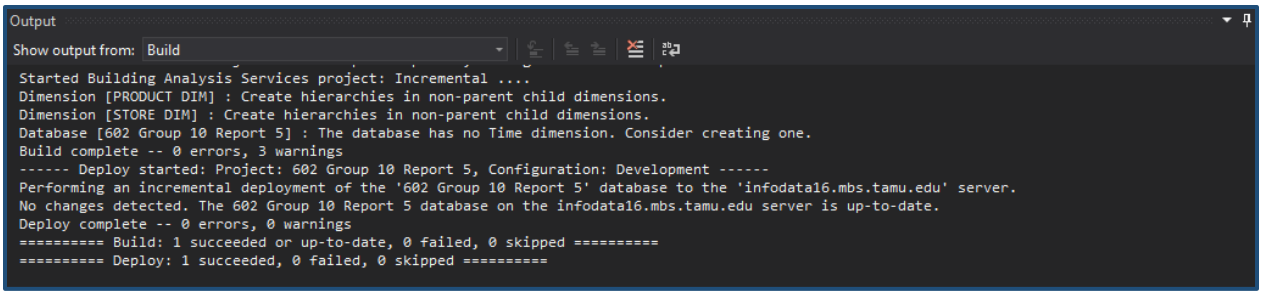

Cube Wizard
Select Measures
Select measures that you want to include in the cube.
Measure
STORE PERF FACT
PROFIT VALUE
AGE9
AGE60
STORE PERF FACT Count
< Back
Next >
Finish >>|
Cancel


Cube Wizard
Select Existing Dimensions
Select existing dimensions to include in the cube.
Dimension
PRODUCT DIM
STORE DIM
< Back
Next >
Finish >>|
Cancel

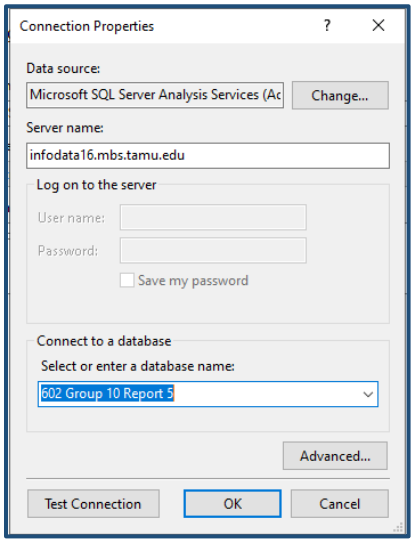


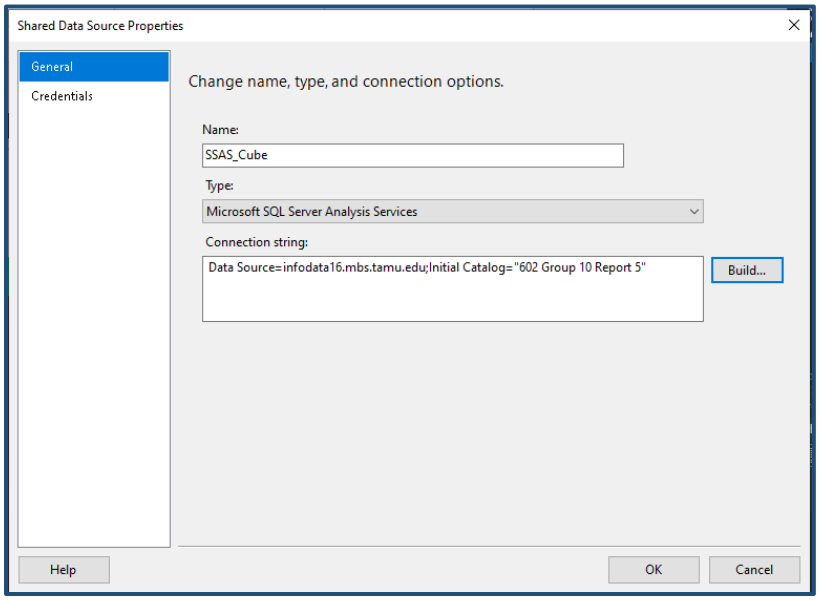
Deploying the Cube:



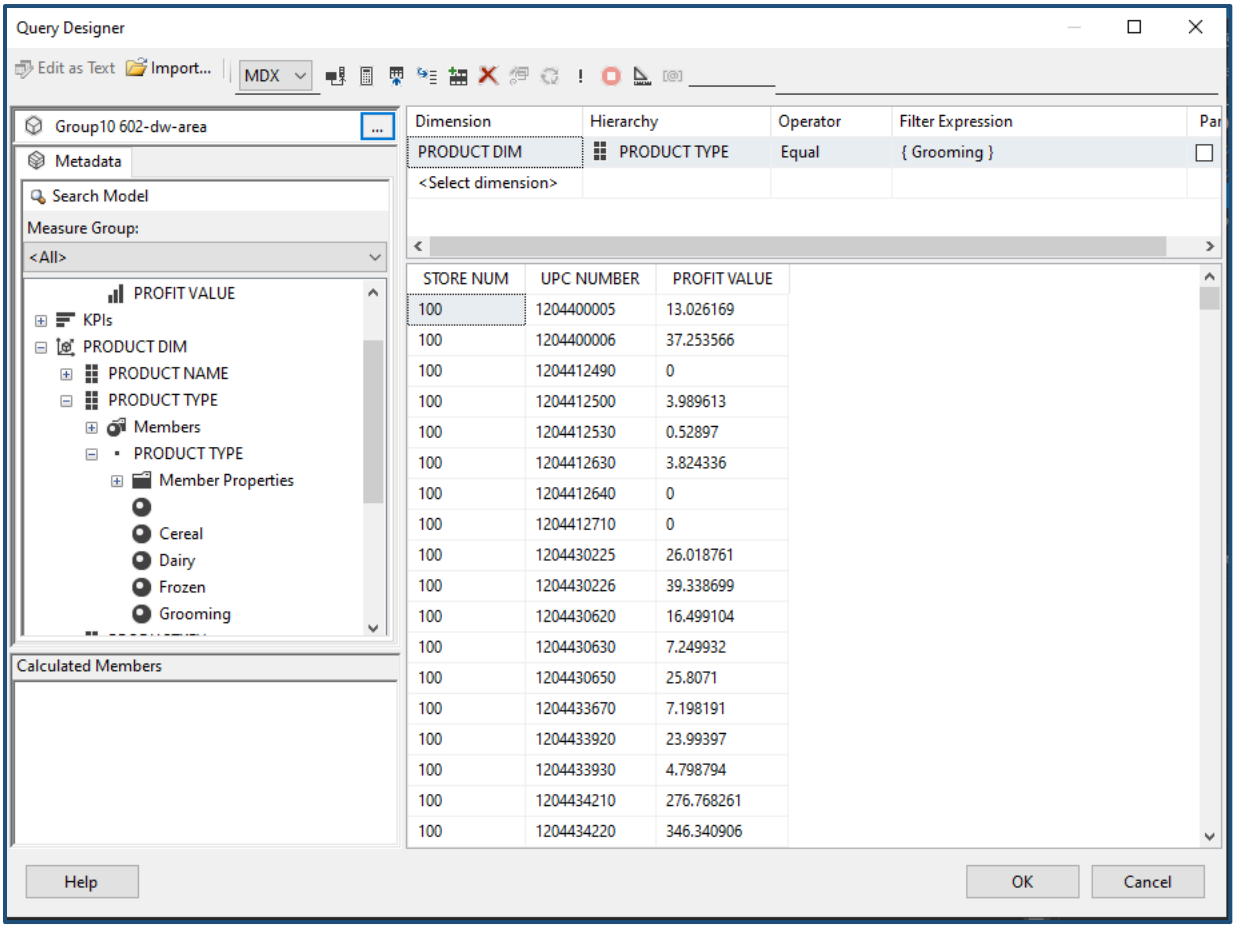


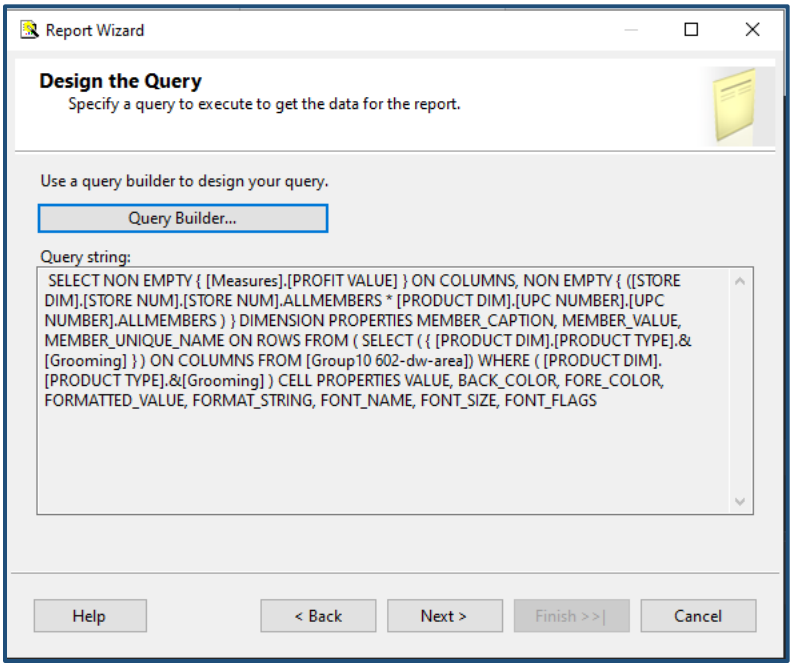
Connecting Cube to SSRS:





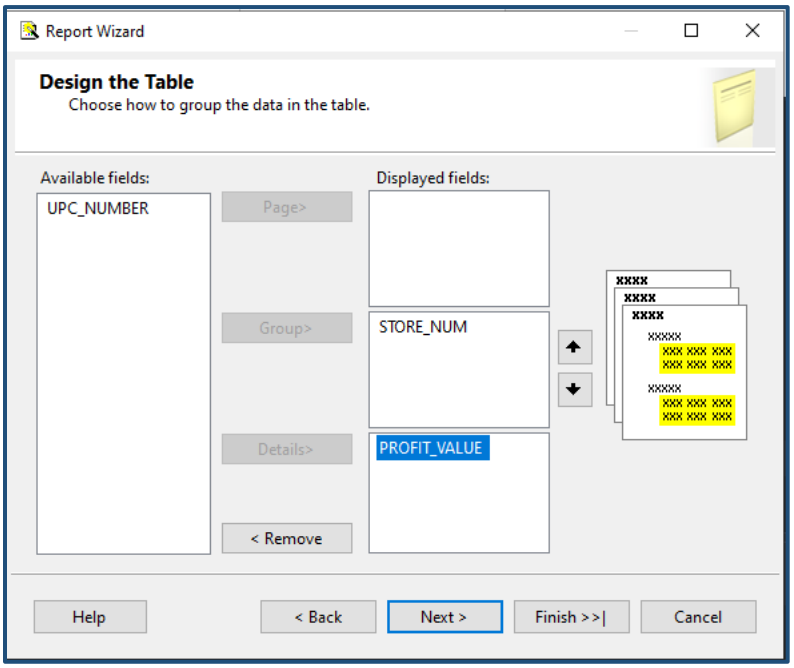
Creating the Report:



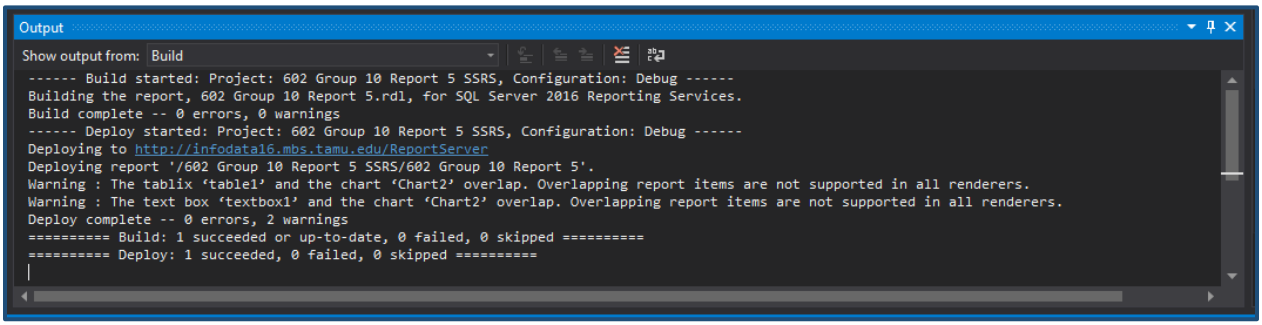


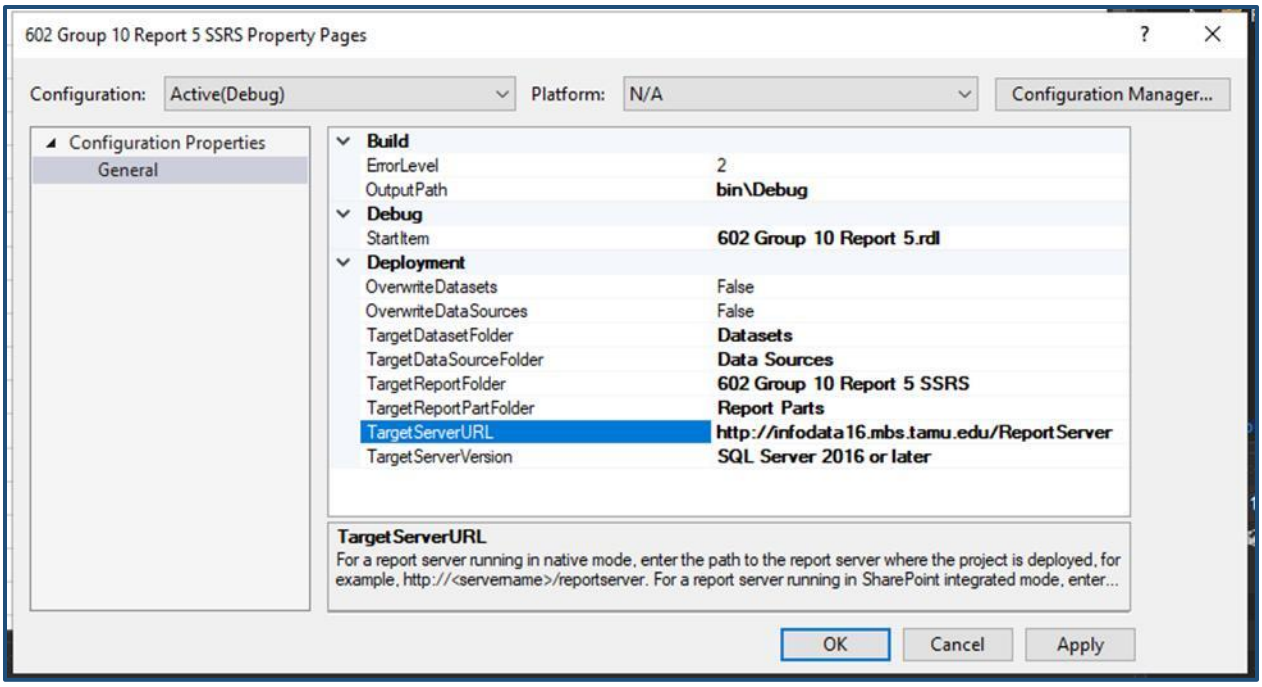
Query:

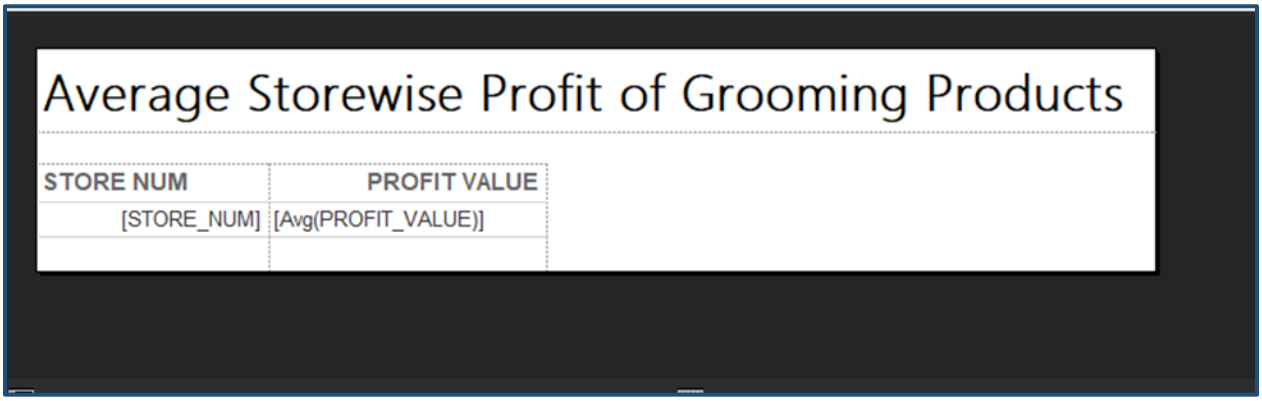
SELECT NON EMPTY { [Measures].[PROFIT VALUE] } ON COLUMNS, NON EMPTY { ([STORE DIM].[STORE NUM].[STORE NUM].ALLMEMBERS * [PRODUCT DIM].[UPC NUMBER].[UPC NUMBER].ALLMEMBERS) } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_VALUE, MEMBER_UNIQUE_NAME ON ROWS FROM (SELECT ({ [PRODUCT DIM].[PRODUCT TYPE].&[Grooming] }) ON COLUMNS FROM [Group10 602-dw-area]) WHERE ([PRODUCT DIM].[PRODUCT TYPE].&[Grooming]) CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR, FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS

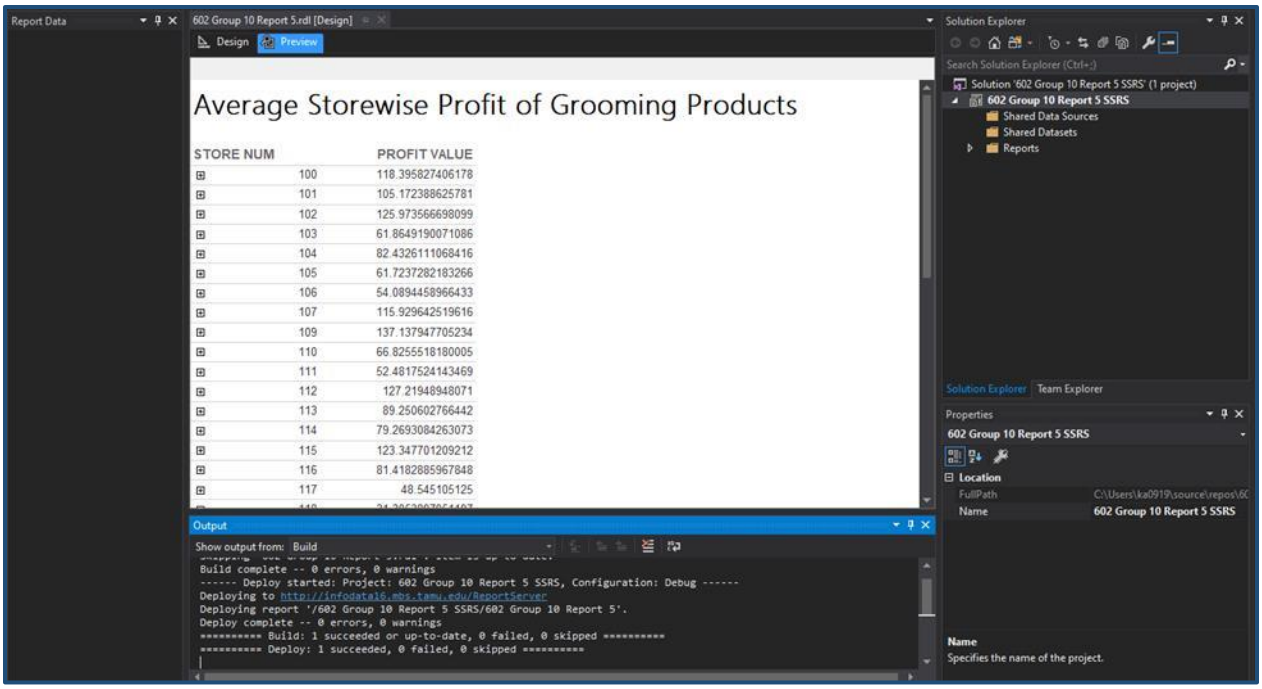


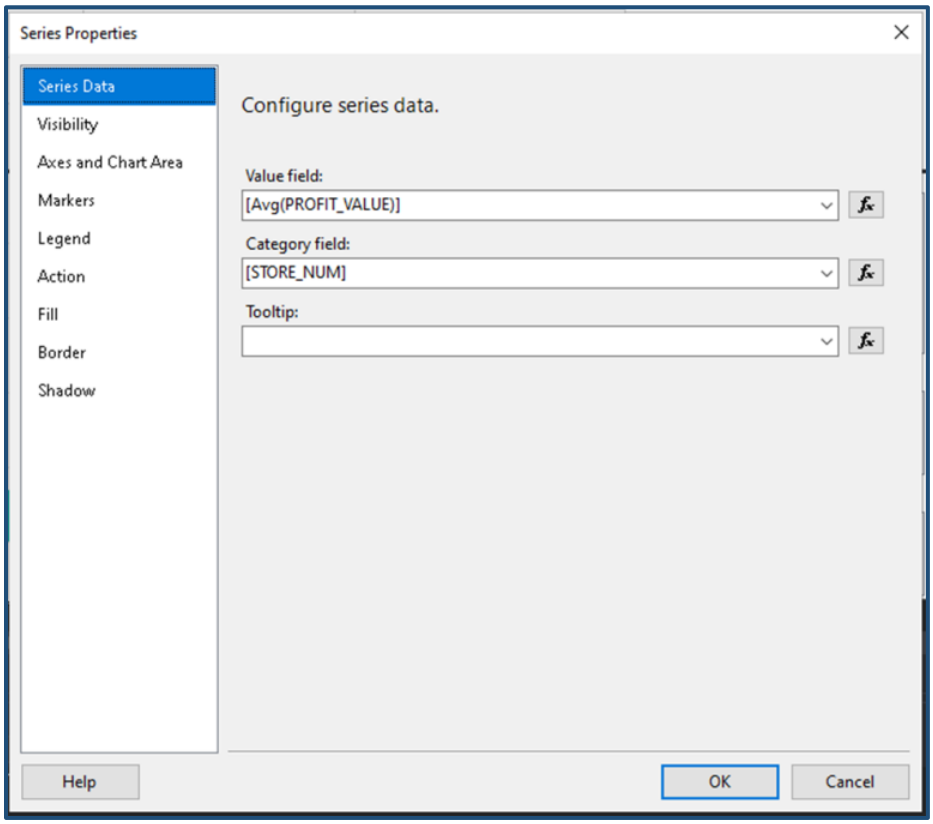
Deploying the Report

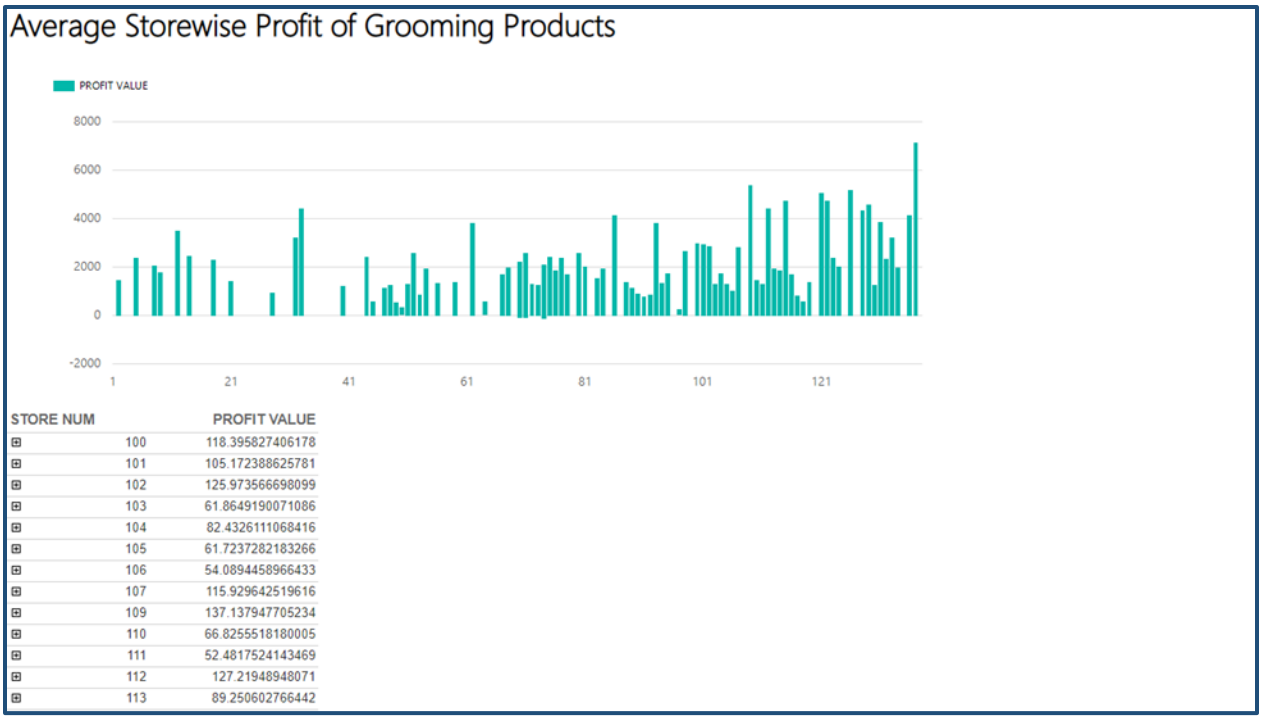



602 Group 10 Report 5 SSRS Property Pages
Configuration: Active(Debug)
Platform: N/A
Configuration Manager...
Configuration Properties
General
Build
ErrorLevel 2
OutputPath bin\Debug
Debug
StartItem 602 Group 10 Report 5.rdl
Deployment
OverwriteDatasets False
OverwriteDataSources False
TargetDatasetFolder Datasets
TargetDataSourceFolder Data Sources
TargetReportFolder 602 Group 10 Report 5 SSRS
TargetReportPartFolder Report Parts
TargetServerURL http://infodata16.mbs.tamu.edu/ReportServer
TargetServerVersion SQL Server 2016 or later
TargetServerURL
For a report server running in native mode, enter the path to the report server where the project is deployed, for example, http://<servername>/reportserver. For a report server running in SharePoint integrated mode, enter...
OK
Cancel
Apply


Average Storewise Profit of Grooming Products
STORE NUM
PROFIT VALUE
[STORE_NUM]
[Avg(PROFIT_VALUE)]


Report Data
602 Group 10 Report 5.rdl [Design]
Design
Preview
Average Storewise Profit of Grooming Products
STORE NUM
PROFIT VALUE
100 118.395827406178
101 105.172388625781
102 125.973566698099
103 61.8649190071086
104 82.4326111068416
105 61.7237282183266
106 54.0894458966433
107 115.929642519616
109 137.137947705234
110 66.8255518180005
111 52.4817524143469
112 127.21948948071
113 89.250602766442
114 79.2693084263073
115 123.347701209212
116 81.4182885967848
117 48.545105125
Output
Show output from: Build
Build complete -- 0 errors, 0 warnings
------ Deploy started: Project: 602 Group 10 Report 5 SSRS, Configuration: Debug ------
Deploying to http://infodata16.mys.tamu.edu/ReportServer
Deploying report '/602 Group 10 Report 5 SSRS/602 Group 10 Report 5'.
Deploy complete -- 0 errors, 0 warnings
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
========== Deploy: 1 succeeded, 0 failed, 0 skipped ==========
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution '602 Group 10 Report 5 SSRS' (1 project)
602 Group 10 Report 5 SSRS
Shared Data Sources
Shared Datasets
Reports
Solution Explorer
Team Explorer
Properties
602 Group 10 Report 5 SSRS
Location
FullPath
Name
602 Group 10 Report 5 SSRS
Name
Specifies the name of the project.


Series Properties
Series Data
Visibility
Axes and Chart Area
Markers
Legend
Action
Fill
Border
Shadow
Configure series data.
Value field:
[Avg(PROFIT_VALUE)]
Category field:
[STORE_NUM]
Tooltip:
Help
OK
Cancel



STORE NUM	PROFIT VALUE
100	118.395827406178
101	105.172388625781
102	125.973566698099
103	61.8649190071086
104	82.4326111068416
105	61.7237282183266
106	54.0894458966433
107	115.929642519616
109	137.137947705234
110	66.8255518180005
111	52.4817524143469
112	127.21948948071
113	89.250602766442

Result: The results above show the average store wise profits for grooming products using bar chart. It is evident from the chart that there is huge variation in the average profits across different stores. Store 141 has the highest average store wise profit for grooming products while store 95 has least average store wise profit.

References

1. Li, H. (2005). Applications of data warehousing and data mining in the retail industry. Proceedings of ICSSSM 05. 2005 International Conference on Services Systems and Services Management, 2005. doi:10.1109/icsssm.2005.1500153
2. Huang, Tao, Robert Fildes, and Didier Soopramanien. (2014). "The Value of Competitive Information in Forecasting FMCG Retail Product Sales and the Variable Selection Problem." European Journal of Operational Research
3. Nevo, A., & Wolfram, C. (2002). Why Do Manufacturers Issue Coupons? An Empirical Analysis of Breakfast Cereals. The RAND Journal of Economics, 33(2), 319. doi:10.2307/3087436
4. Jami, Ata and Himanshu Mishra. (2013). "Downsizing and Supersizing: How Changes in Product Attributes Influence Consumer Preferences." Journal of Behavioral Decision Making
5. Chevalier, J., Kashyap, A., & Rossi, P. (2000). Why Don't Prices Rise During Periods of Peak Demand? Evidence from Scanner Data. doi:10.3386/w7981

6. Ponniah, P. (2010). Data Warehousing Fundamentals for IT Professionals: A Comprehensive Guide for IT Professionals. Retrieved from http://business.baylor.edu/gina_green/teaching/dw/spr16/Ponniah_data-warehousing-fundamentals-for-it-professionalsSecondEdition.pdf
7. Michelle A. Poolet | Jul 29. (2019, January 16). Indexing the Data Warehouse. Retrieved from https://www.itprotoday.com/sql-server/indexing-data-warehouse
8. Database Data Warehousing Guide. (2017, January 17). Retrieved from http://www.dba86.com/docs/oracle/12.2/DWHSG/data-warehouse-physical-design.htm
9. Bhaskar, V. (2013, September 29). Logical and Physical Design of Data Warehouse. Retrieved from https://www.folkstalk.com/2011/04/logical-and-physical-design-of-data.html
10. S, V. (2018, February 27). Data Warehouse Physical Data Model. Retrieved from https://dwgeek.com/data-warehouse-physical-data-model.html/