

**DESIGN AND IMPLEMENTATION OF A DATA WAREHOUSE**

Joint Consulting Project

**Section 602 - Group 6**

**Team Members**

Naresh Choudhary

Pallavi Tiwari

Chetan Joshi

Mays Business School - Texas A&M University

**Table of Contents**

1. **Introduction**
   1. **Details about DFF data**
   2. **Domain understanding**
2. **Business Questions and Dimension Models**
   1. **Kimball’s Matrix for Data Marts**
   2. **Business questions and corresponding models**
3. **Warehouse design**
4. **Marts and mapping**
5. **ETL**
6. **Physical design**

**1. Introduction**

Dominick's Finer Food chain, founded in 1918, was a Chicago-area grocery store chain which operated over 30+ locations during 1990 to 2000. DFF operated in sales of wide range of products including dairy, meat, cosmetics, alcohol, pharmacy and fresh produce. DFF was known for its experimentation over market to generate revenue. It experimented with ‘food and drug’ combo, walkie-talkie for communication, the ways through which the announcements were made to the customer and so on. After running for more than 20 years, on Dec 2013, its parent company announced to close all its stores in Chicago. [1]

To create a data warehouse for DFF, major concern is the complex data set. It contains files from 5 critical tables of DFF recording customer in-traffic, product details and inventory count over 10 years. Complexity is magnified by erroneous and incomplete information in the tables resulting in multiple interpretation of data, for example there are negative values in quantity of a commodity. It can be interpreted as junk or loss incurred to store due to breakage or spoilage leading to uncertainty.

Beside complex dataset, lack of subject knowledge of Retail and Marketing might hinder the development of a data warehouse which actually brings value to the business.

**i. Details about DFF Data**

*DFF majorly consist of 4 data files: CCOUNT, DEMOGRAPHICS, UPC and MOVMENT.*

*Understanding the data:*

CCOUNT**:** It is in-store traffic information file which contains number of customers visited and sales of each product in dollars segregated by date and stores. Information is also grouped by weeks. It also stores the coupons redeemed for various products on daily basis for each store. Data is dirty comprising of missing primary columns such as store id and date and includes unknown characters and negative values. Each store’s demographic information is further described in Store-Specific Demographics excel file.

Store-Specific Demographics: It contains demographic details containing location of each store. For example, % of household with 1 person or % of College Graduates. Many tables hold repetitive data (i.e. contains same information) and can be removed/ignored.

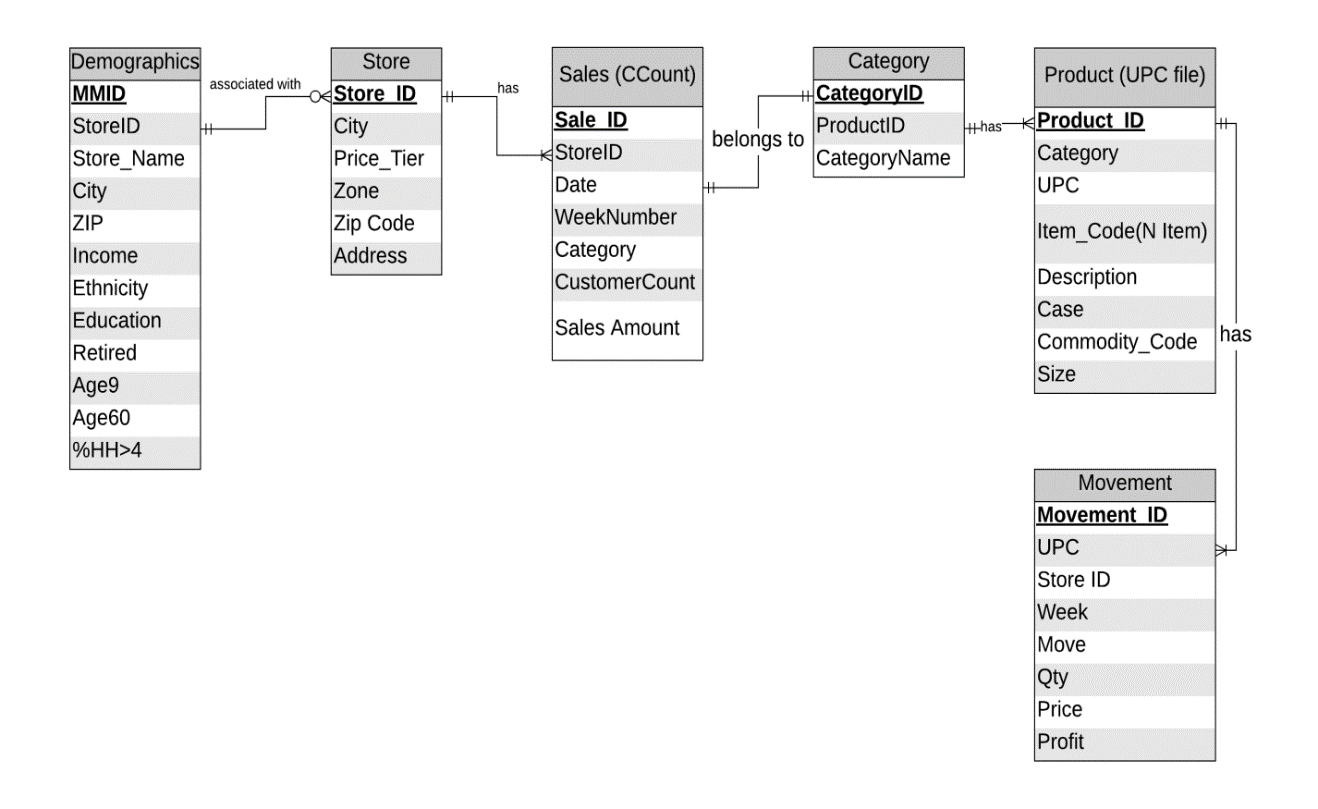
UPC: UPC files contain the unique key to identify products which are followed throughout market i.e. among different vendors, distributors and manufacturers. UPC key contains the product key associated with manufacturer key. Each file in UPC corresponds to single category which in turn has multiple commodities (identified by commodity code). DFF uses Item Code which uniquely defines a product within DFF database. Same key also helps to identify if the product was delivered to the Dominick's warehouse or directly to the store.

Movement:Movement file stores number of units, profit and Quantity segregated by week, UPC number and stores.

*OLTP Metadata:*

|  |  |  |
| --- | --- | --- |
| **File** | **Metadata** | **Description** |
| CCount | Store, Date, Week, CustCount, Grocery, Dairy, Frozen, Bottle, Meat, MeatFroz, MeatCoup, Fish, FishCoup, Beer, Wine, Spirits, Pharmacy, DairyCoup, PharCoup, etc. | CCount has customer count data for a store for a particular date. It also consists of coupons redeemed and sales data for categories such as fish, dairy, etc. |
| Demographics | Age9, age60, ethnic, educ, nocar, income, incsigma, hsizeavg, hsize1, hsize2, hsize34, hsize567, Hh3plus, hh4plus, hhsingle, hhlarge, workwo, retired, unemp, wrkch17, nwrkch5, nwrkch17, wrkch, nwrkch, wrkwch, wrkwnch, telephn, mortgage, nwhite, poverty, etc. | This file consists of demographics data which is distributed store wise. It provides with data based on age, income, gender, family size etc. |
| Movement | Store, UPC, Move, Profit, Qty, Price, Sale, Week | Movement file stores the data of number of items that were sold for a store in a week |
| UPC | Com\_Code, UPC, Description, Case, Size, Nitem | Product data from with description and commodity codes |

*Entity-Relation Diagram*



*Database Representation of Tables*

CCount

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sales ID | StoreID | Date | Category | Sales | Customer Count |
| 1 | 47 | 880101 | GROCERY | 14900 | 1546 |
| 2 | 47 | 880101 | DAIRY | 3321 | 1546 |
| 3 | 47 | 880101 | FROZEN | 2625 | 1546 |
| 4 | 47 | 880101 | BOTTLE | 0 | 1546 |
| . | . | . | . | . | . |
| 100 | 47 | 880102 | GROCERY | 15881 | 1749 |
| 101 | 47 | 880102 | DAIRY | 3321 | 1749 |
| 102 | 47 | 880102 | FROZEN | 2621 | 1749 |
| 103 | 47 | 880102 | BOTTLE | 0 | 1749 |
| . | . | . | . | . | . |

Store

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| StoreID | City | Price\_Tier | Zone | ZipCode | Address |
| 2 | Chicago | High | B | 60439 | 5400 N. Lakewood Avenue |

Demographic

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MMID | StoreID | Store\_Name | City | Zip | Income | Ethnicity | Education | Retired | Age9 | Age60 | %HH>4 |
| 12345 | 2 | Dominick Chicago | Chicago | 60439 | 10.2 | 0.6 | 0.8 | 0.67 | 0.54 | 0.46 | 0.33 |

Category

|  |  |  |
| --- | --- | --- |
| CategoryID | CategoryName | ProductID |
| WTPA | toothpaste | 100002 |
| WTPA | toothpaste | 100001 |

Product

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ProductID | UPC | Category | Item\_code | Description | Case | CommodityCode | Size |
| 1 | 1192603016 | ANA | 7342431 | CAFFEDRINE CAPLETS 1 | 6 | 953 | 16 CT |
| 2 | 1192662108 | ANA | 7333311 | SLEEPINAL SOFTGEL | 6 | 953 | 8 CT |
| 3 | 1192603016 | ANA | 7342431 | CAFFEDRINE CAPLETS 1 | 12 | 953 | 16 CT |

Movement

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MovementID | UPC | StoreID | Week | MOve | QTY | DiscountType | Price | Profit |
| 1 | 1060831115 | 5 | 298 | 7 | 1 |  | 0.59 | 15.25 |
| 2 | 1060831115 | 5 | 335 | 2 | 1 | B | 0.51 | 50.98 |

**ii.** Domain **Understanding**

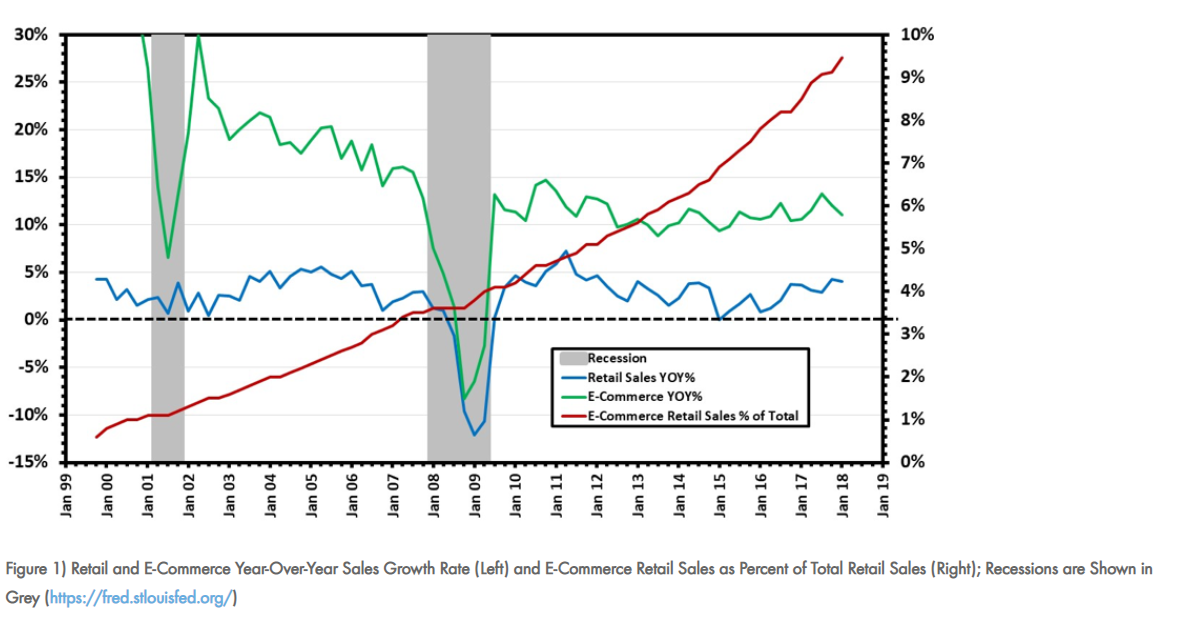
“Retail is the process of selling consumer goods or services to customers through multiple channels of distribution to earn a profit. Retailers satisfy demand identified through a supply chain” [2]. There are four core principles to be followed in retail [3]:

1. Customer is most important
2. Understanding your customer
3. 4 Ps - Product, Price, Promotion, Place
4. Location

*Current trends, Strategies & problems in Retails Market*

*Digital Disruption*

The rise of e-commerce has disruptive effect on retail market. Currently, more than 10% of US retail market is dominated by e-commerce. According to Credit Suisse, more than 8640 stores with 147m sq. feet of space closed in 2018 [4]. Below graph supports the underlying trend:



Retails chains should come up with ideas to annul the loss in revenue due to e-commerce by providing in-home service and delivery.

*Customer Customization according to Demographics and Psychographics*

Customers change their shopping pattern frequently and look for personalized experience. Currently retailers analyze demographic data to understand their customer and provide variety in products in which those customers are interested in. Steps involve analyzing customer base, checking competition, understanding product/service, choosing specific demographic to target and finally expanding product base [4]. However, targeting a specific population doesn’t mean you are excluding people who doesn’t fit criteria. [5]

*Marketing Trends by One of the Smartest Supermarket in the United States*

Texas based super mart H-E-B is notorious for promoting local produce by associating them with symbol of pride towards their state. Products are branded like Houston Blend Coffee and Hill Country, which trigger the sentiments in the customers and lead to similar amount of sales as compared to branded products.

Another strategy they use is Neuro marketing by utilization of yellow coupons spread across the aisles. “Put a big yellow price sign next to something, and it will light up people's brains - even if it's not actually a great deal” [6]. In H-E-B’s case, they offer flagged coupons to tear off and to be carried to the checkout counter for scanning which takes Cialdini’s principles of persuasion in account by influencing the customers to engage rather than just pick up a product with discount pasted on the product label.

Utilizing this strategy can lead to promotion of certain products which may have been failing to grasp customer’s attention. Furthermore, by putting these strategies in action, Dominick’s can promote in-house products and utilize coupons in the stores.

*Revenue Vs Profitability*

Retailers face several challenges, the biggest one is to increase profitability. Customers have ever changing demands which are difficult to track. This causes problems in managing the supply chain. There may be products which are always in demand but there is low availability due to supply chain inefficiencies or lack of labor. Retailers also fail to evaluate the potential customers and markets which leads to lower sales. The inventory management system especially during the holiday season is not efficient to manage the inventory across the all the stores. Customers are ever demanding of good quality products which are cheap. To do this, they resort to multiple channels such as online shopping. Retailers are expected to be consistent across all the channels and provide convenience to the customers. Often, retailers fail to estimate their competitors’ capabilities. Small retailers cannot compete with big retailers on price. Conclusively, they need to focus on something other than price to create a good shopping experience. [7]

**2. Dimensional models**

* + 1. **Kimball’s Matrix for Data Marts:**

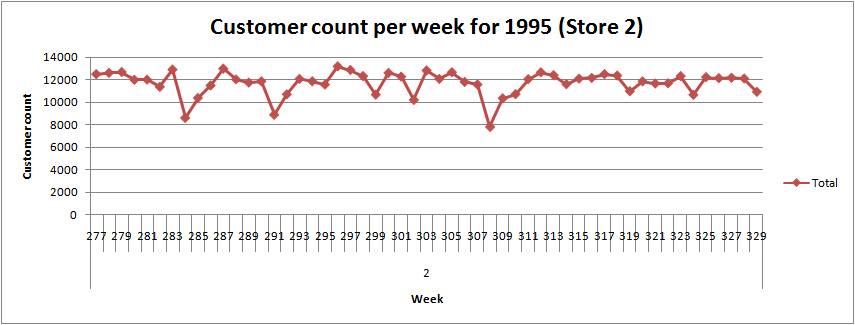
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Marts** | **Time** | **Product** | **Store Demo** | **Demographic** |
| Customer Count & Coupon Redemption Sales | X | X | X |  |
| Product Sales | X | X | X |  |

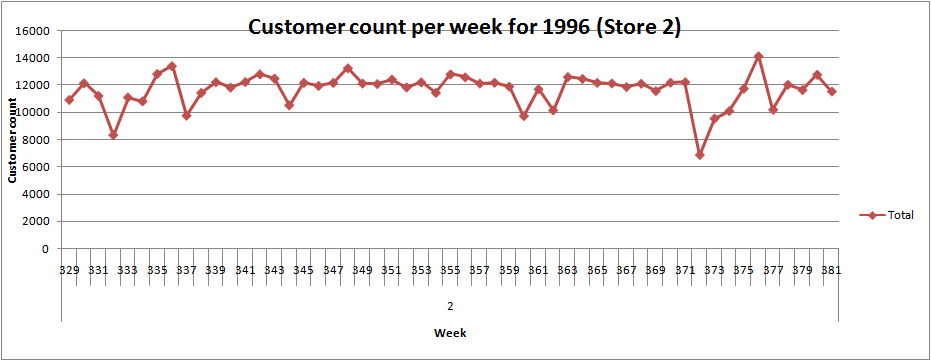
*The Data Mart Matrix*

* + 1. **Business questions and corresponding models**

1. **What is weekly trend of customer in traffic for store 2 between 1995 and 1996?**

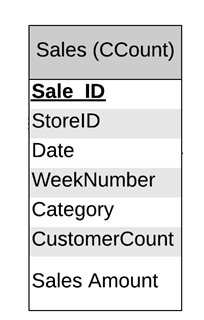
**Pivot:**

****

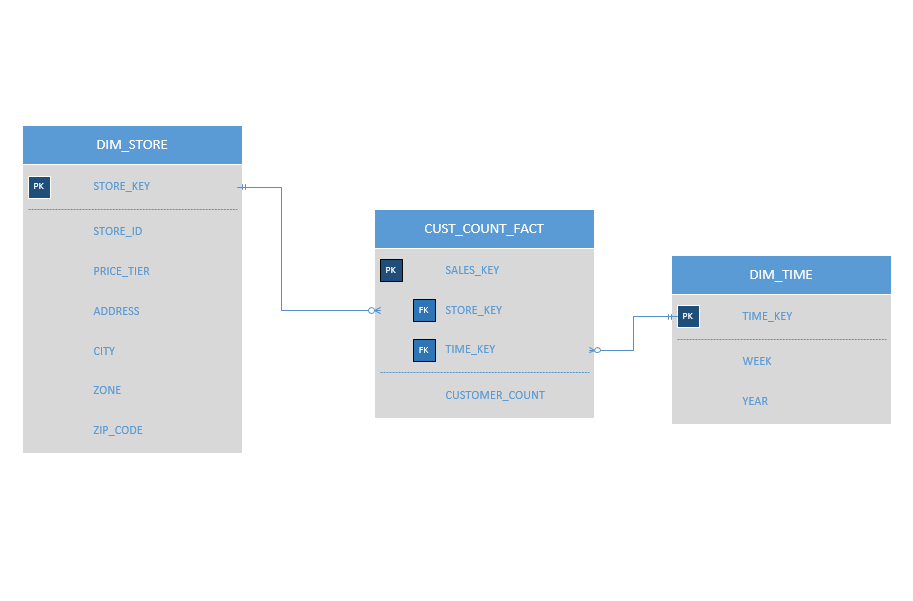
****

**Business Justification:** For a sample of stores, we have taken data for year 1995 and 1996 to find out the number of customers visiting the store weekly. Using this data, we have identified a general trend of reduction in customer count during the 1st week of months. During these trough points in graph i.e. the weeks with reduced customer count, we can reduce the staff or let the staff members plan their vacations. This will further lead to reduction of over-stocking and employees count can be reduced to save salary expenses.

**ERD:**

****

**Dimensional Model:**

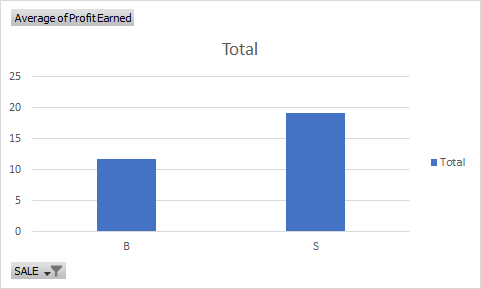


**Schema Justification:** To calculate customer count on a specific store on weekly basis, we need two dimensions namely *DIM\_TIME* and *DIM\_STORE*. Time dimension will aggregate the data on weeks and year. Similarly store dimension will help to find customer count on different stores.

*CUST\_COUNT\_FACT* will have auto incrementing primary key and foreign key to other dimensions. *CUST\_COUNT\_FACT* will have only one attribute *CUSTOMER\_COUNT* which corresponds to customer in traffic on a particular week on a particular store (summing daily customer count to aggregate it week wise: *sum (customer\_count\_daily) from sales\_table/CCount group\_by week, group\_by store)*

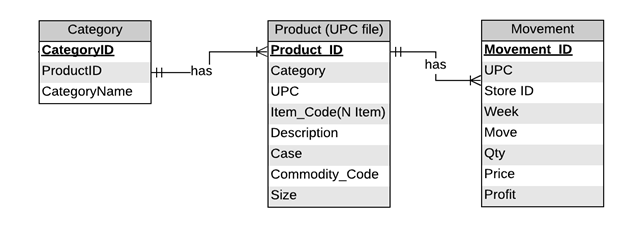
**2. In category of frozen products, which method of discount (Coupon or price reduction) gives more profit?**

**Pivot:**

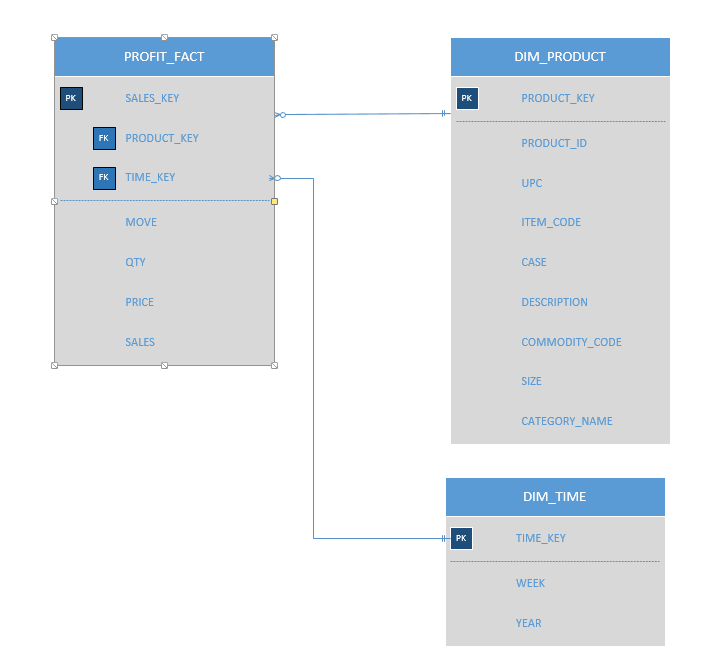


**Schema Justification**: Due to lower shelf life of frozen foods they are often sold on discounted rates to avoid loss. Analysis of profit earned in category frozen entree shows that when simple price reduction is offered, sale and profit is more. Dominick can consider offering simple price reduction rather than offering coupons.

**ERD:**



**Dimensional Model:**

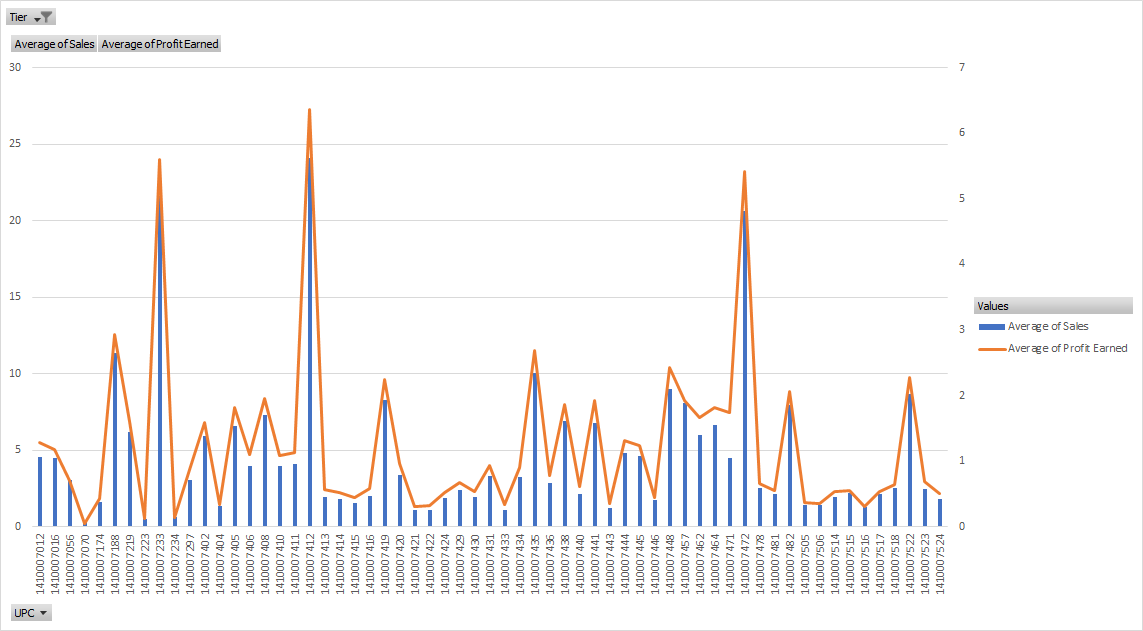


**Schema Justification:** For this question, we propose a schema which consists of a fact table PROFIT\_FACT and two dimension tables as DIM\_PRODUCT and DIM\_TIME. The basic requirement of this question is to calculate the sales of frozen products when products had simple discount vs same product when they had bonus buy offer. This schema holds these sales details in PROFIT\_FACT as ‘sales’ attribute. Since BQ asks for frozen products, PROFIT\_FACT is linked to DIM\_PRODUCT in which PRODUCT\_KEY is the surrogate key & PRODUCT\_ID is the OLTP database’s ID. Further, to figure out discount type product dimension for any week or year (coming from the DIM\_TIME) table will be used.

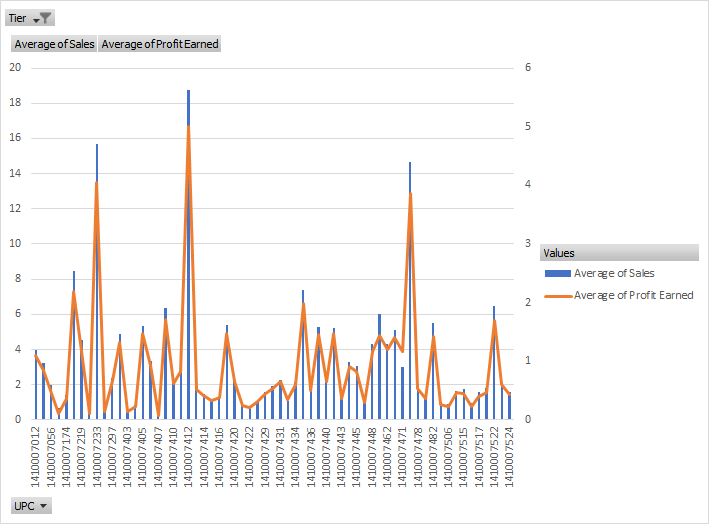
Data in PROFIT\_FACT table comes from movement table/movement file. Since ERD movement table (or movement file from Dominick) records data on a daily basis, sales will be aggregated week wise.

**3. Which UPCs had most sale in high, medium and low tier for cookies?**

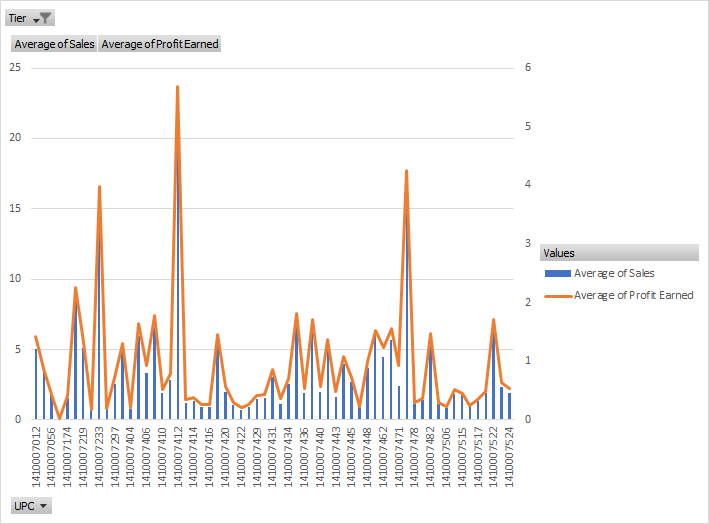
**Pivot (High tier stores):**



**Pivot (Low tier stores):**

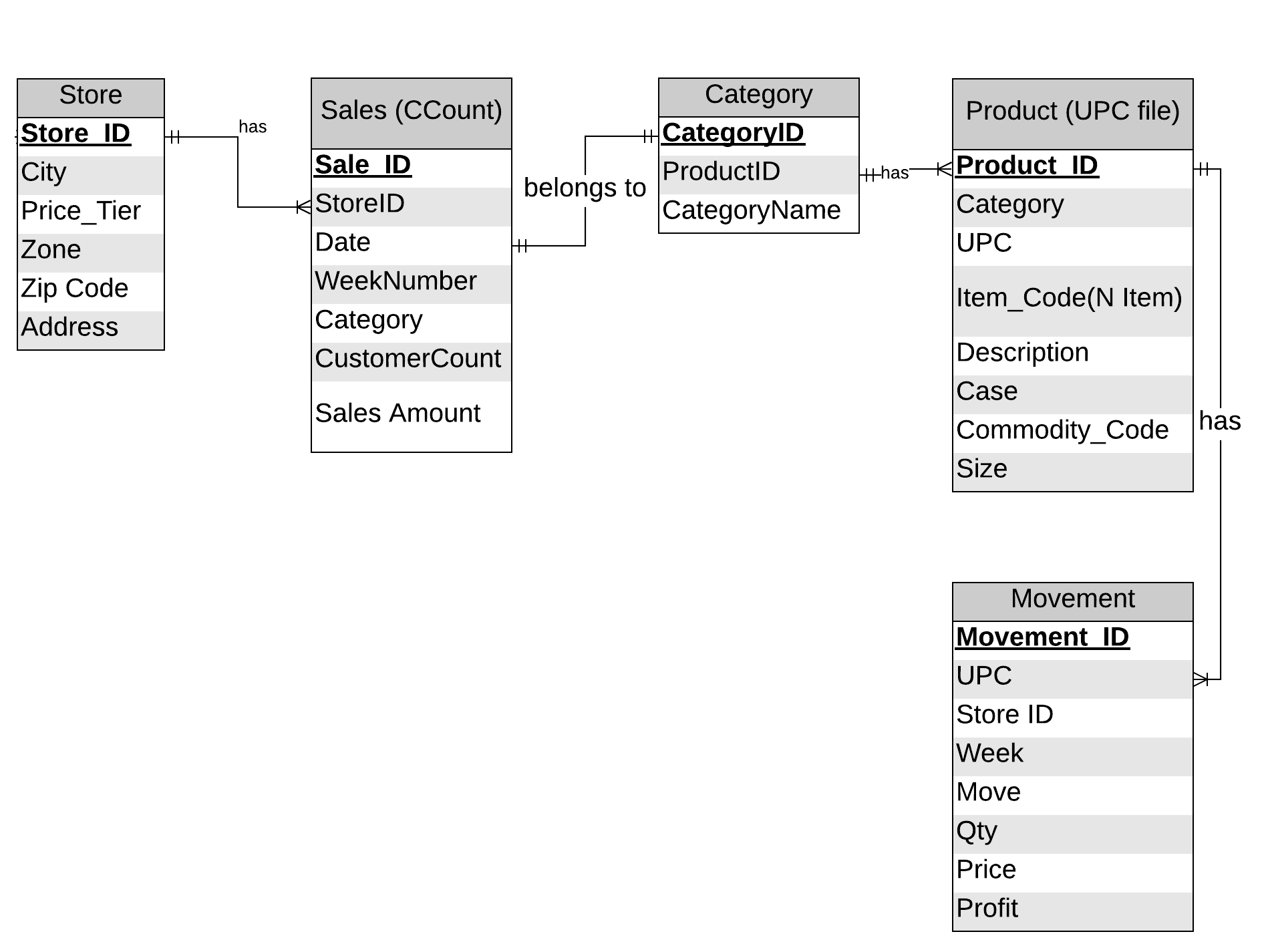
****

**Pivot (Medium tier stores):**

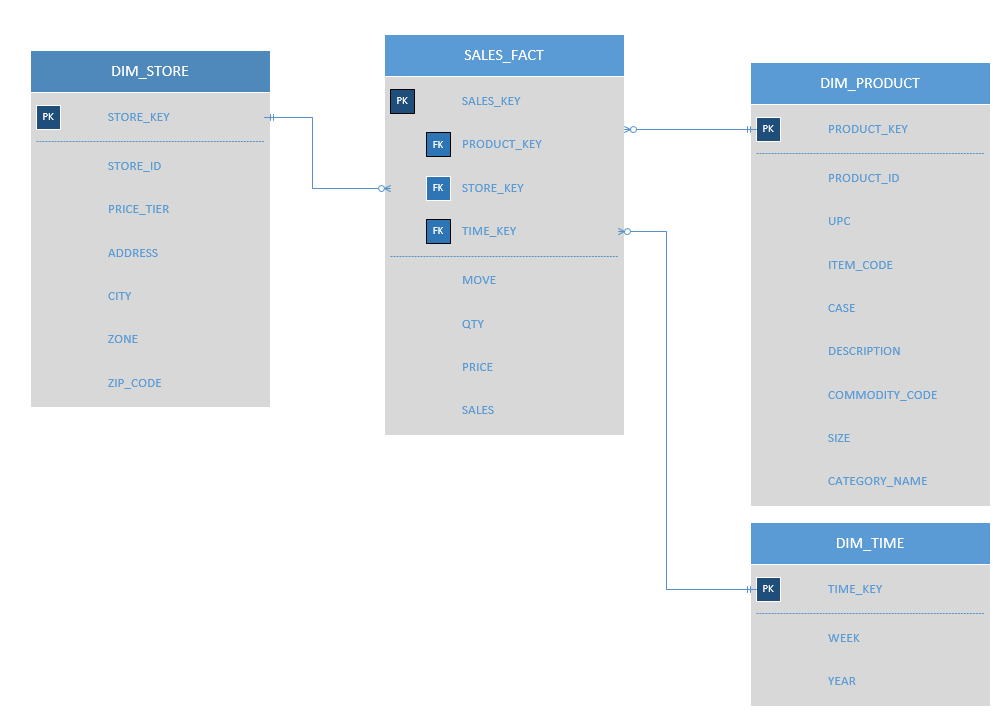
****

**Schema Justification**: For cookies, the pivot charts show data for different tiers i.e. high, medium and low. We observed that sale of certain UPCs was very high as compared to the others. For example, the sale and profit of UPC is high across all the tiers. This analysis can help the business to promote certain UPCs more than the others. Our analysis on different tiers for the cookies shows that known products, especially high and medium tiers one, are often bought. The business can then focus on the sale of these products by introducing coupons or discounts. They should particularly aim at the high/medium tier to increase profit margins substantially.

**ERD:**

****

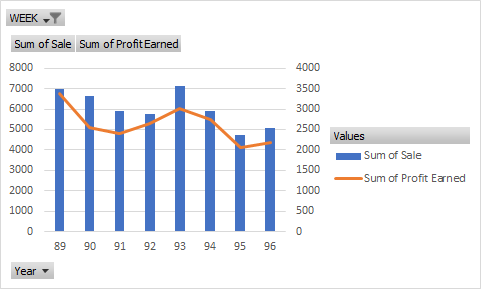
**Dimensional Model:**



**Schema Justification:** For this question, we propose a schema which consists of a fact table SALES\_FACT and three dimension tables as DIM\_PRODUCT, DIM STORE and DIM\_TIME. We need to find the UPCs (Cookie type) that had highest sales in different tiers. For this, we have designed the SALES\_FACT table to store the movement, quantity, price and sales data (from movement table). The UPC data comes from the Product dimension. DIM\_STORE will store the store details which will provide the tier data. In this way, we can capture the sale of a UPC in a particular store/tier.

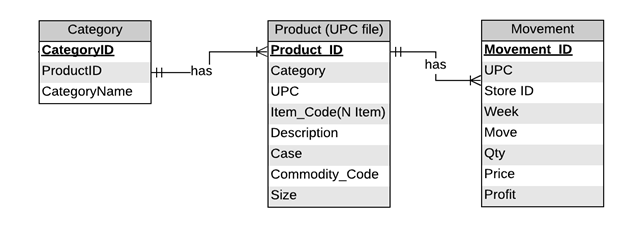
**4. What is the trend of candies’ sale during Halloween year by year?**

**Pivot:**

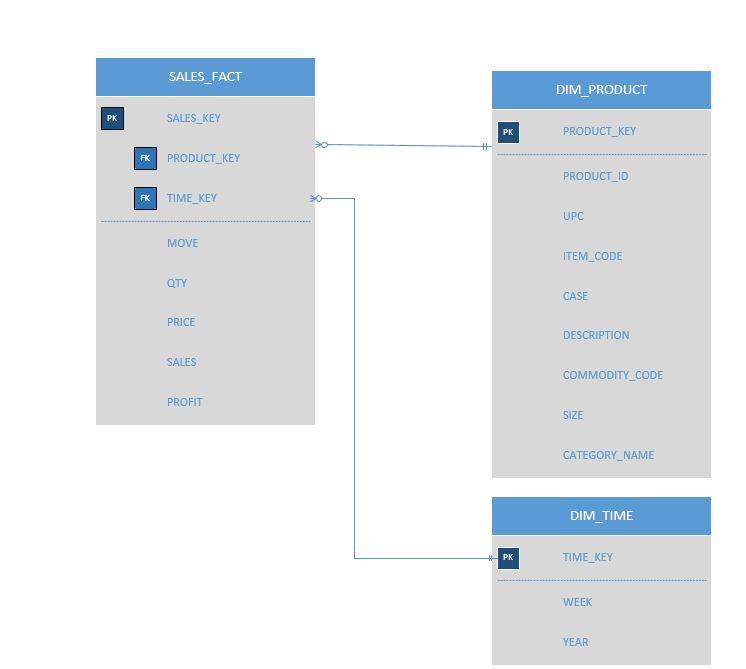


**Schema Justification**: Graph shows the trend of candies during Halloween season from 89 to 96. It can help to predict sales of next year which could range between 4000-5000. This trend can be validated with data from years to come.

**ERD:**



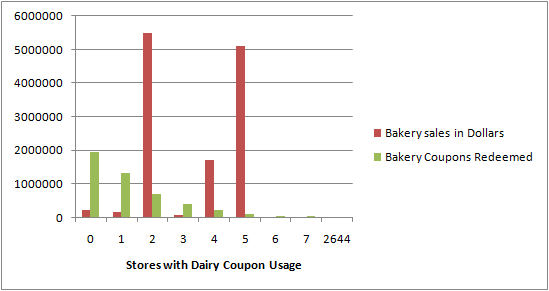
**Dimensional Model:**



**Schema Justification:** To find the sales of candies again uses SALES\_FACT which is linked to DIM\_TIME to find Halloween weeks and DIM\_PRODUCT to find select only candies products.

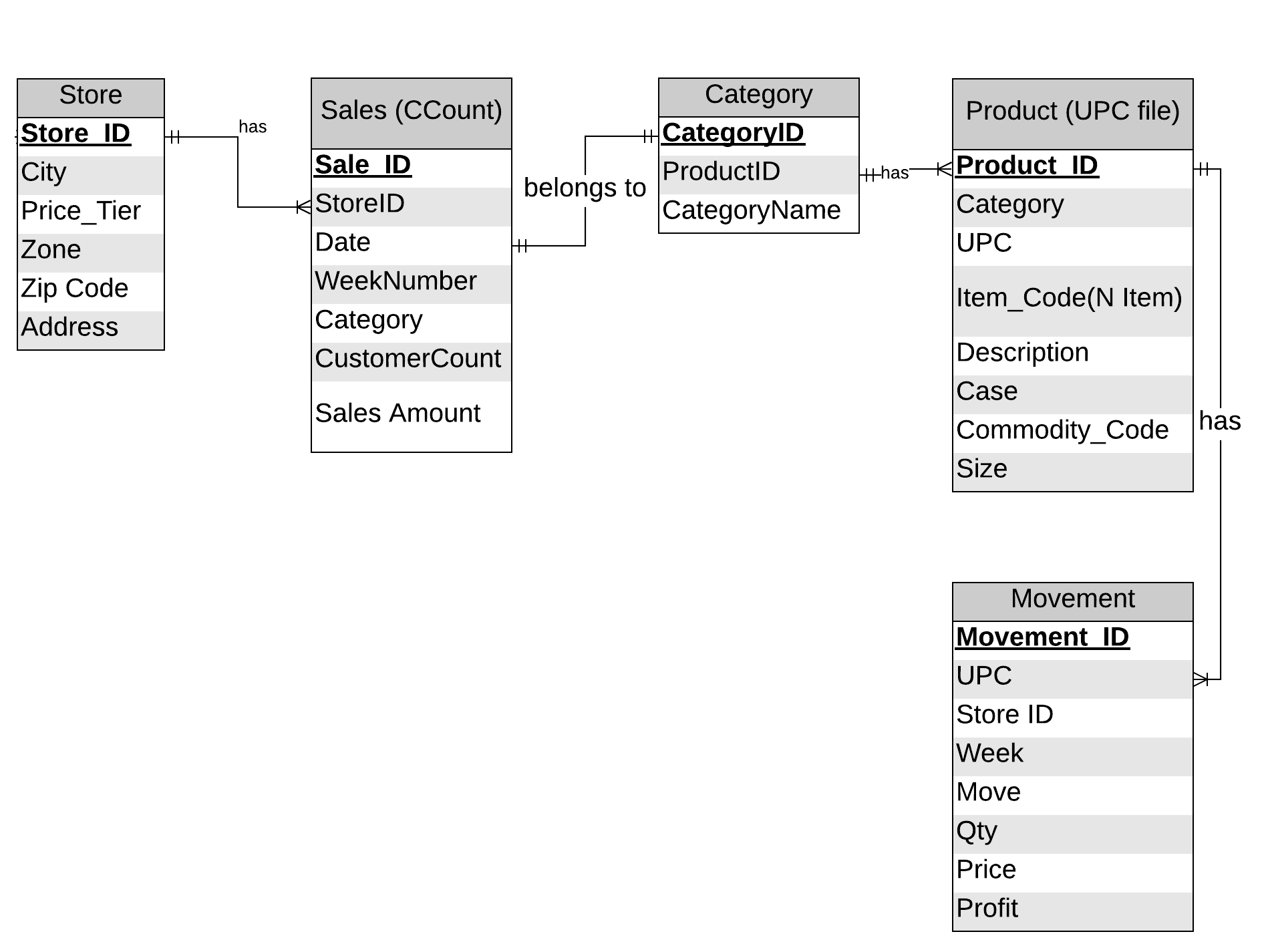
**5. Find out the relation between Bakery sales and bakery coupons redeemed for all stores?**

**Pivot**

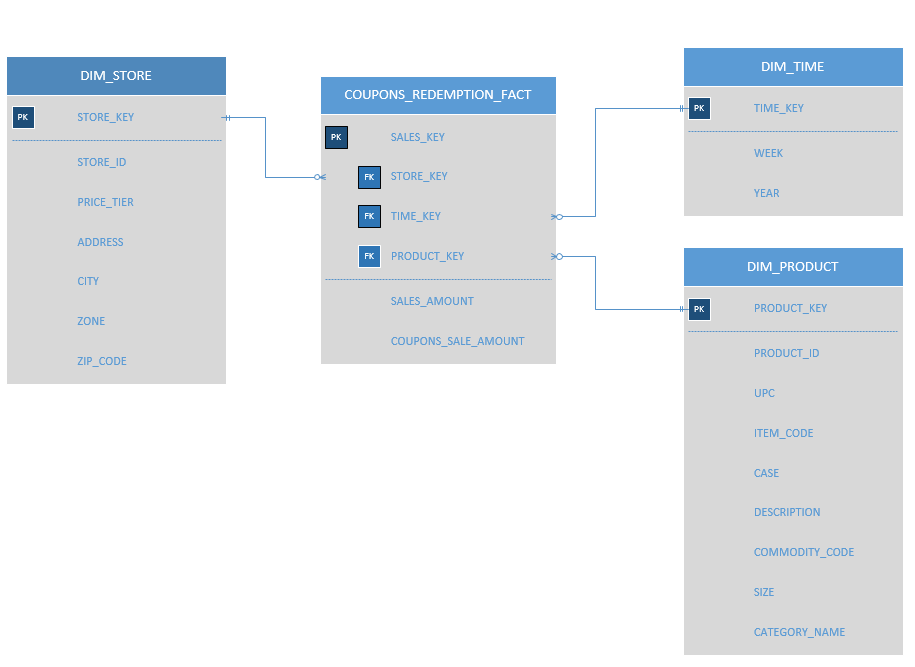


**Schema Justification:** In this business question, we looked at the data given in the CCount table for bakery sales. We wanted to evaluate this amount store wise to see if there is any relevance of the coupons redeemed for bakery items. We noticed that for certain stores such as 0, 1, 2 as shown above, the coupons redemption was high. In contrast, the sale of bakery items was very low. If we see the sale for store 2, 4 and 5, we notice that the bakery sales were extremely high. In this case, the coupons redeemed was not so substantial. This analysis can be further explored to evaluate the difference. One of the strategies by DFF can be to give out more options in coupons or discounts to attract customers in store 2, 4, 5 to eventually increase the sales exponentially. DFF can also try to understand the trend of sales for coupons in stores 0, 1, 2 and strategize to increase the bakery sales in those stores.

**ERD:**

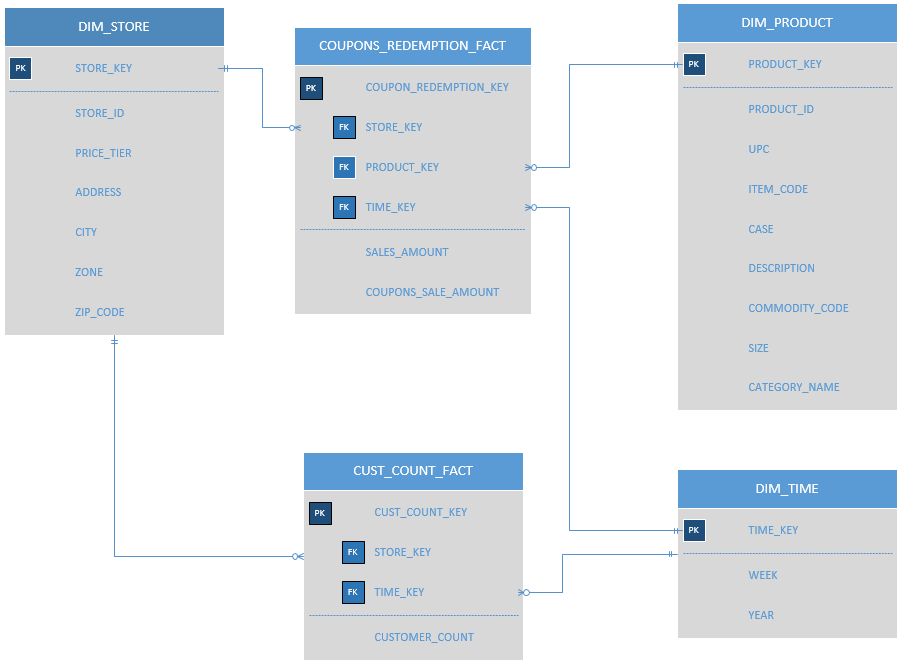
****

**Dimensional Model:**



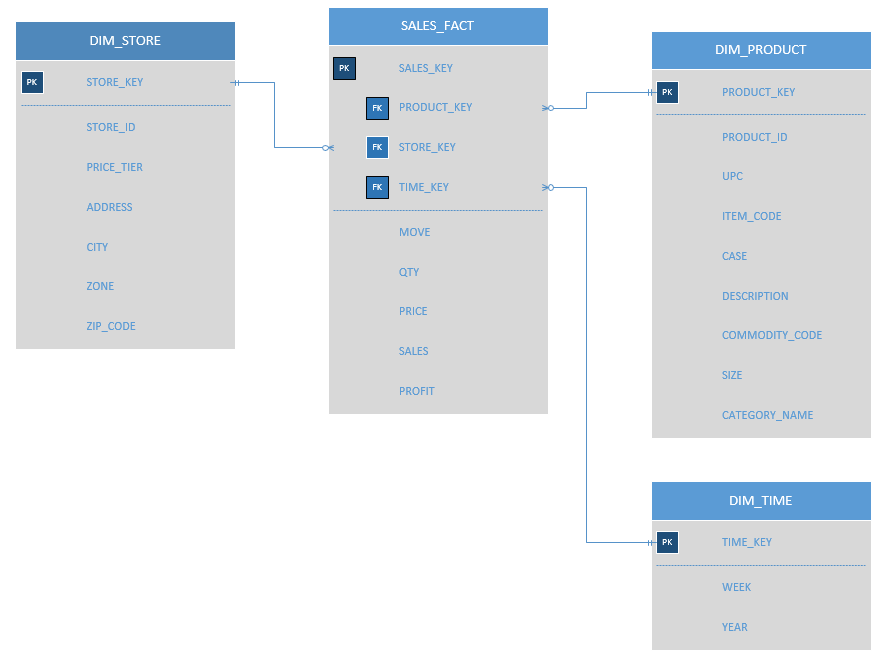
**Schema Justification:** To find the relation between bakery sales and coupons redeemed for all stores, we propose a schema which consists of the COUPONS\_REDEMPTION\_FACT fact table. This table stores the SALES\_AMOUNT for regular sales and COUPONS\_SALE\_AMOUNT for the coupons redeemed sales amount from the CCount table data. The DIM\_STORE table stores the store details and the DIM\_PRODUCT stores the product details along with the CATEGORY\_NAME. The DIM\_TIME dimension table stores the week at the lowest granularity, moving up to year.

**1. Coupon sales & Customer Count Data Mart:**



**Schema Justification:** The first data mart is created using the models created for Q1 & Q5. We combined the models to have two fact tables, one for the Customer Count related facts and the other for Sales amount related to coupons. Rationale behind keeping two fact table is that one question demands fact measured at customer level while other business question demands facts at store level. The other dimensions Time is constant.

1. **Sales & Profit Data Mart:**



**Schema Justification:** The remaining questions Q2, Q3, Q4 require us to calculate sales or profits for products or for a category. In essence, we need a sales fact table in all three questions. Since granularity and other factors were same we merged all 3 fact table to create our second data mart with the fact table as SALES\_FACT and related dimension tables: DIM\_STORE, DIM\_PRODUCT & DIM\_TIME.

**A. MAPPING TABLE:**

**1. Dimension tables:**

**STORE**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DW Target Table** | **DW Target Attribute** | **Target Datatype** | **Source Table** | **Source Column** | **Mapping Function** | **Others** |
| DIM\_STORE | STORE\_KEY | Numeric |  |  |  | Surrogate key |
| DIM\_STORE | STORE\_ID | Numeric | Dominick’s dataset-Store | STOREID |  |  |
| DIM\_STORE | PRICE\_TIER | String | Dominick’s dataset-Store | TIER |  |  |
| DIM\_STORE | ADDRESS | String | Dominick’s dataset-Store | ADDRESS |  |  |
| DIM\_STORE | CITY | String |  | CITY |  |  |
| DIM\_STORE | ZONE | String |  | ZONE |  |  |
| DIM\_STORE | ZIP\_CODE | Numeric |  | ZIP\_CODE |  |  |

**PRODUCT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DW Target Table** | **DW Target Attribute** | **Target Datatype** | **Source Table** | **Source Column** | **Mapping Function** | **Others** |
| DIM\_PRODUCT | PRODUCT\_KEY | Numeric |  |  |  | Surrogate key |
| DIM\_PRODUCT | PRODUCT\_ID | Numeric |  | Generated in ERD |  | Primary key |
| DIM\_PRODUCT | UPC | String | Dominick’s dataset-Store | UPC |  |  |
| DIM\_PRODUCT | ITEM\_CODE | String | Dominick’s dataset-Store | ITEM\_CODE |  |  |
| DIM\_PRODUCT | CASE | Numeric | Dominick’s dataset-Store | CASE |  |  |
| DIM\_PRODUCT | ZONE | String | Dominick’s dataset-Store | ZONE |  |  |
| DIM\_PRODUCT | DESCRIPTION | String | Dominick’s dataset-Store | DESCRIPTION |  |  |
| DIM\_PRODUCT | COMMODITY\_CODE | String | Dominick’s dataset-Store | COMMODITY\_CODE |  |  |
| DIM\_PRODUCT | SIZE | String | Dominick’s dataset-Store | SIZE |  |  |
| DIM\_PRODUCT | CATEGORY\_NAME | String | Dominick’s dataset-Store | CATEGORY\_NAME |  |  |

**TIME:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DW Target Table** | **DW Target Attribute** | **Target Datatype** | **Source Table** | **Source Column** | **Mapping Function** | **Others** |
| DIM\_TIME | TIME\_KEY | Numeric |  |  |  | Surrogate key |
| DIM\_TIME | WEEK | Numeric | Dominick’s dataset-Store | WEEK\_NUMBER |  |  |
| DIM\_TIME | YEAR | Numeric | Dominick’s dataset-Store | YEAR |  |  |

**SALES\_FACT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DW Target Table** | **DW Target Attribute** | **Target Datatype** | **Source Table** | **Source Column** | **Mapping Function** | **Others** |
| SALES\_FACT | SALES\_KEY | Numeric |  |  |  |  |
| SALES\_FACT | PRODUCT\_KEY | Numeric | DIM\_PRODUCT | PRODUCT\_KEY |  |  |
| SALES\_FACT | STORE\_KEY | Numeric | DIM\_STORE | STORE\_KEY |  |  |
| SALES\_FACT | MOVE | Numeric | Dominick’s dataset-Store | MOVE | sum(move) from movement group by week |  |
| SALES\_FACT | QTY | Numeric | Dominick’s dataset-Store | QTY | sum(qty) from movement group by week |  |
| SALES\_FACT | PRICE | Numeric | Dominick’s dataset-Store | PRICE | sum(price) from movement group by week |  |
| SALES\_FACT | SALES | Numeric | Dominick’s dataset-Store | SALES | sum(sales) group by week |  |

**COUPONS\_REDEMPTION\_FACT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DW Target Table** | **DW Target Attribute** | **Target Datatype** | **Source Table** | **Source Column** | **Mapping Function** | **Others** |
| COUPONS\_REDEEMPTION\_FACT | COUPON\_REDEMPTION\_KEY | Numeric |  |  |  | Primary key |
| COUPONS\_REDEEMPTION\_FACT | PRODUCT\_KEY | Numeric | DIM\_PRODUCT | PRODUCT\_KEY |  |  |
| COUPONS\_REDEEMPTION\_FACT | STORE\_KEY | Numeric | DIM\_STORE | STORE\_KEY |  |  |
| COUPONS\_REDEEMPTION\_FACT | TIME\_KEY | Numeric | DIM\_TIME | TIME\_KEY |  |  |
| COUPONS\_REDEEMPTION\_FACT | SALES\_AMOUNT | Numeric | Dominick’s dataset-Store | SALES\_AMOUNT | Sum (sales\_amount) from sales\_table/ccount group by week |  |
| COUPONS\_REDEEMPTION\_FACT | COUPON\_SALE\_AMOUNT | Numeric | Dominick’s dataset-Store | SALES\_AMOUNT | Sum (sales\_amount\_ from sales\_table/ccount where category group by week |  |

**CUST\_COUNT\_FACT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DW Target Table** | **DW Target Attribute** | **Target Datatype** | **Source Table** | **Source Column** | **Mapping Function** | **Others** |
| CUST\_COUNT\_FACT | CUST\_COUNT\_KEY | Numeric |  |  |  | Primary key |
| CUST\_COUNT\_FACT | STORE\_KEY | Numeric | DIM\_STORE | STORE\_KEY |  |  |
| CUST\_COUNT\_FACT | TIME\_KEY | Numeric | DIM\_TIME | TIME\_KEY |  |  |
| CUST\_COUNT\_FACT | CUSTOMER\_COUNT | Numeric | SALES/ MOVEMENT | SALES |  |  |

1. **ETL Development Plan and Implementation**
   * 1. **Target data**

**a. Plan**

Our solution for the data warehouse consists of 3 fact tables and 3 dimension tables.

The target data is shown below:

Dimension Tables:

|  |  |  |
| --- | --- | --- |
| **Target table** | **Target Column** | **Target Column Datatype** |
| 602\_Group6\_Warehouse.DIM\_STORE | STORE\_KEY | INT |
| STORE\_ID | INT |
| PRICE\_TIER | VARCHAR |
| ADDRESS | VARCHAR |
| CITY | VARCHAR |
| ZONE | VARCHAR |
| ZIP\_CODE | VARCHAR |

|  |  |  |
| --- | --- | --- |
| **Target table** | **Target Column** | **Target Column Datatype** |
| 602\_Group6\_Warehouse.DIM\_WEEK | WEEK\_KEY | INT |
| WEEK\_NUMBER | INT |
| START | DATE |
| END | DATE |
| SPECIAL\_EVENTS | VARCHAR |

|  |  |  |
| --- | --- | --- |
| **Target table** | **Target Column** | **Target Column Datatype** |
| 602\_Group6\_Warehouse.DIM\_PRODUCT | PRODUCT\_KEY | INT |
| PRODUCT | INT |
| ITEM\_CODE | INT |
| UPC | INT |
| SIZE | VARCHAR |
| CASE | INT |
| CATEGORY | VARCHAR |
| DESCRIPTION | VARCHAR |
| COMMODITY\_CODE | INT |

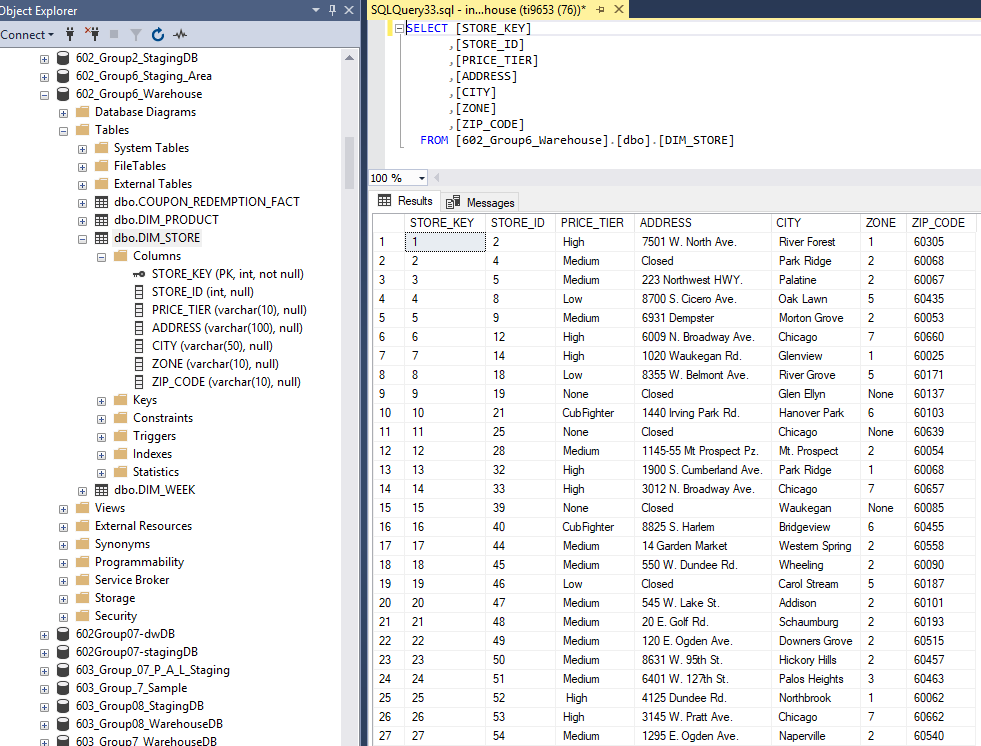
Fact Tables:

|  |  |  |
| --- | --- | --- |
| **Target table** | **Target Column** | **Target Column Datatype** |
| 602\_Group6\_Warehouse.SALES\_MOVE\_FACT | STORE\_KEY | INT |
| STORE\_ID | INT |
| PRICE\_TIER | VARCHAR |
| ADDRESS | VARCHAR |
|  | CITY | VARCHAR |
|  | ZONE | VARCHAR |
|  | ZIP\_CODE | VARCHAR |

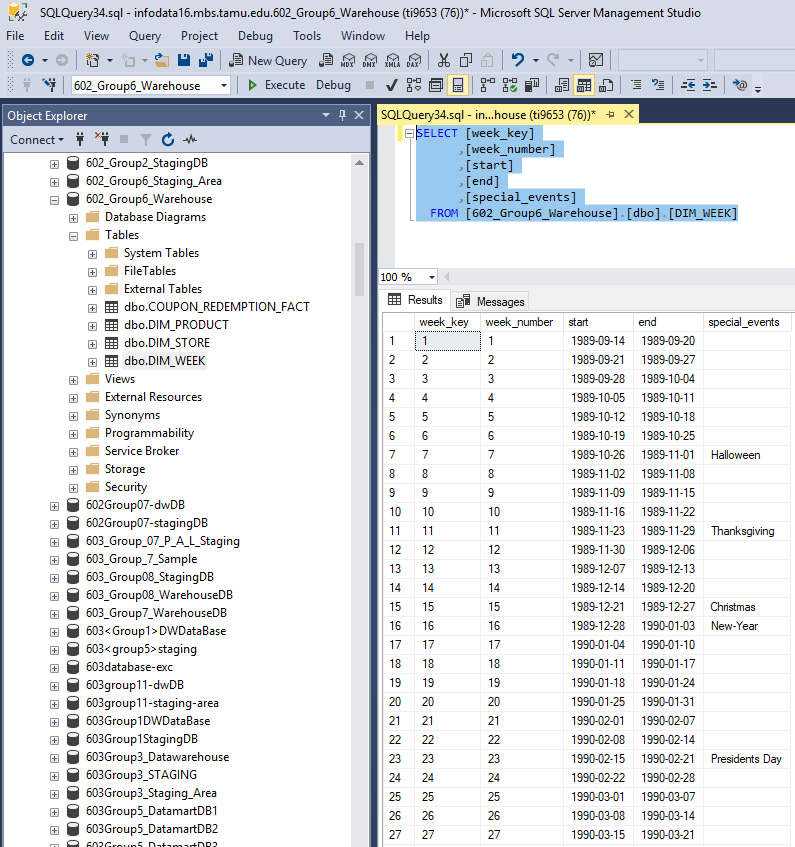
|  |  |  |
| --- | --- | --- |
| **Target table** | **Target Column** | **Target Column Datatype** |
| 602\_Group6\_Warehouse.CUST\_COUNT\_FACT | STORE\_KEY | INT |
| STORE\_ID | INT |
| PRICE\_TIER | VARCHAR |
| ADDRESS | VARCHAR |
|  | CITY | VARCHAR |
|  | ZONE | VARCHAR |
|  | ZIP\_CODE | VARCHAR |

|  |  |  |
| --- | --- | --- |
| **Target table** | **Target Column** | **Target Column Datatype** |
| 602\_Group6\_Warehouse.COUPON\_REDEMPTION\_FACT | COUPON\_REDEMPTION\_KEY | INT |
| STORE\_KEY | INT |
| WEEK\_KEY | INT |
| PRODUCT\_KEY | INT |
|  | SALES\_AMOUNT | FLOAT |

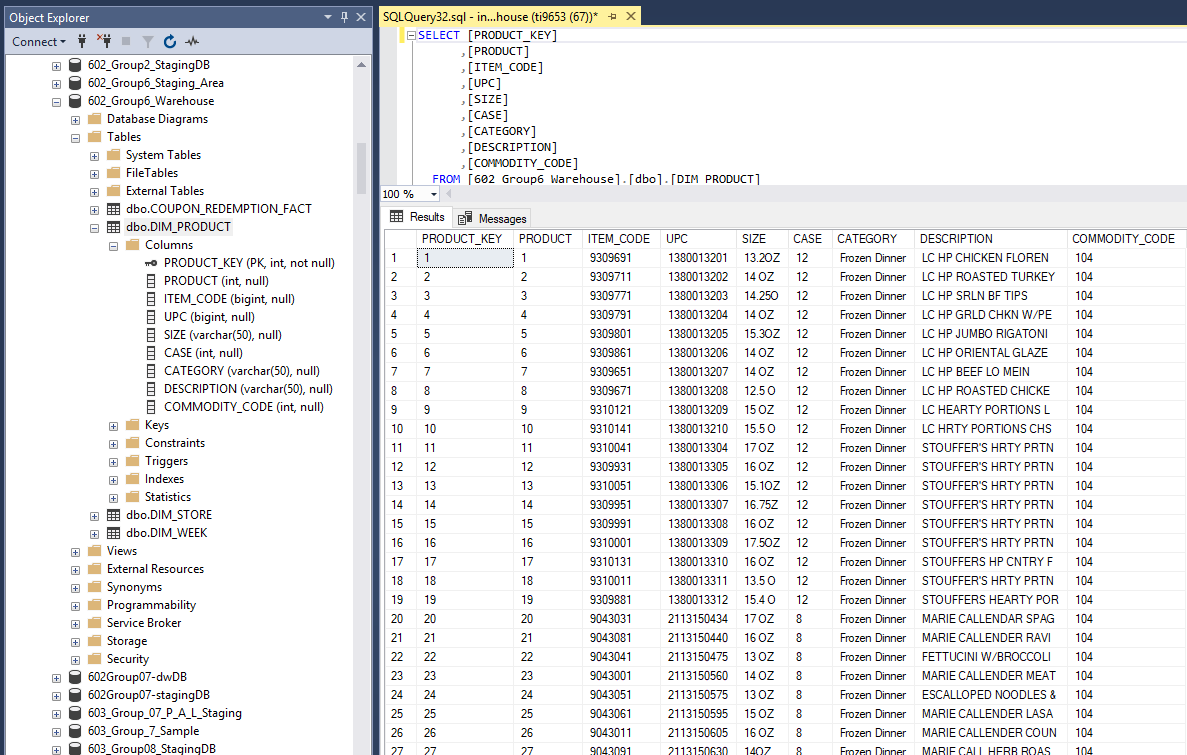
DIM\_STORE

****

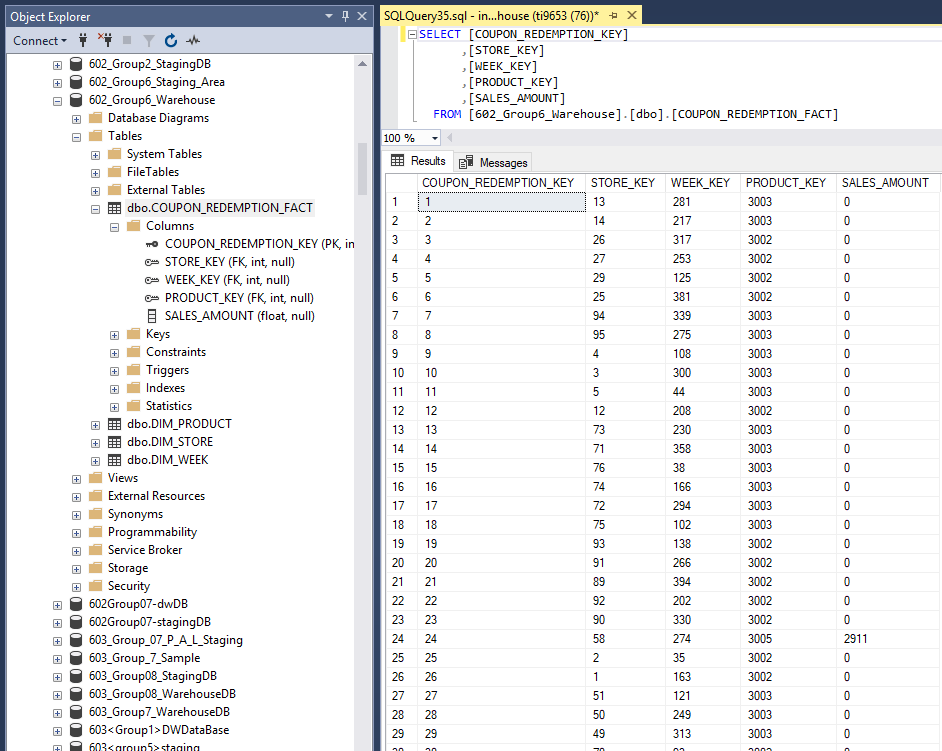
DIM\_WEEK

****

DIM\_PRODUCT

****

COUPON\_REDEMPTION\_FACT

****

* + 1. **Data Sources**

The data sources are the files given in the Dominick data set. For the business questions that we need to focus on, we used the CSV files provided which are CCount, Movement, UPC and Store files.

For one business question, we require data of customer count. This data is available in the CCount file. Another question required us to import data from CCount because we needed to find the sales data for coupons redemption for a category. We will use the columns needed for these questions from CCount file. For three questions, we need the sales and profit, so we will make use of the Movement and UPC files for this. Additionally, we will use the Store table given in the Dominick data manual to create a store dimension as we have questions which use the store data.

* + 1. Prepare and write data mappings for data elements from sources in Excel to staging and then data mapping from staging to data warehouse (include all transformations);

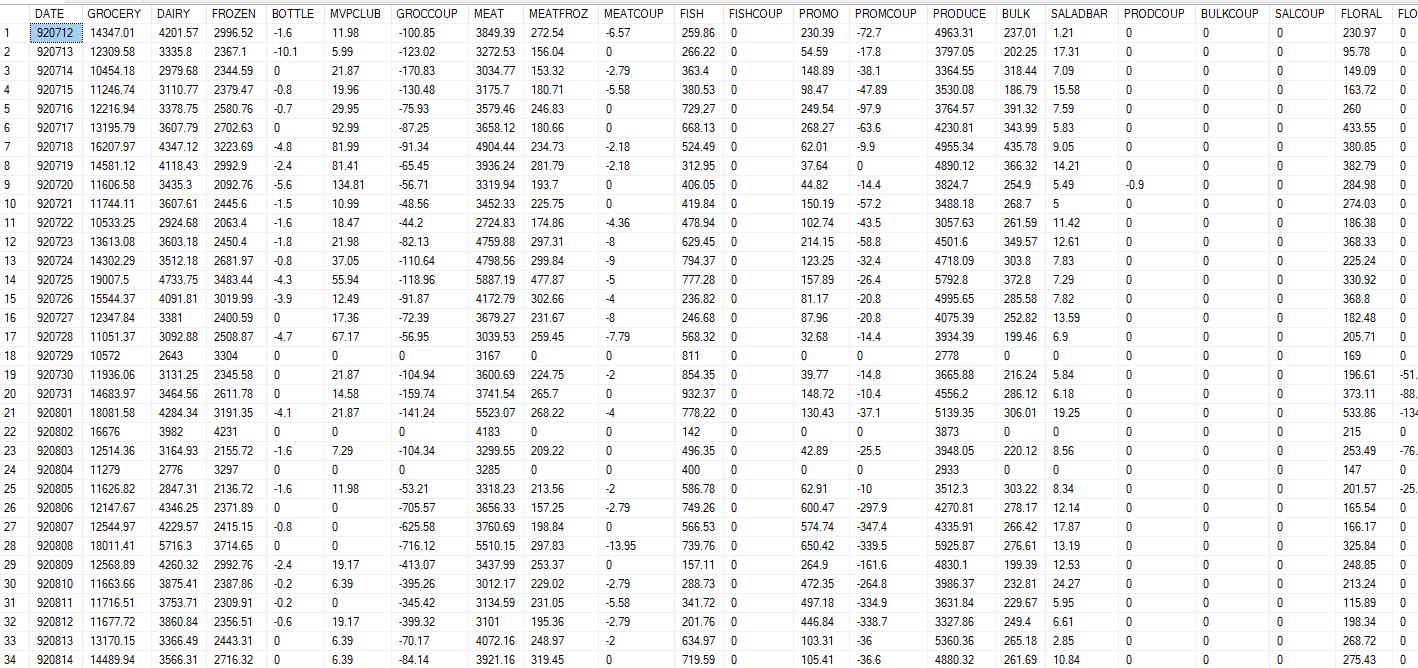
NARESH

* + 1. **Data Extraction rules**

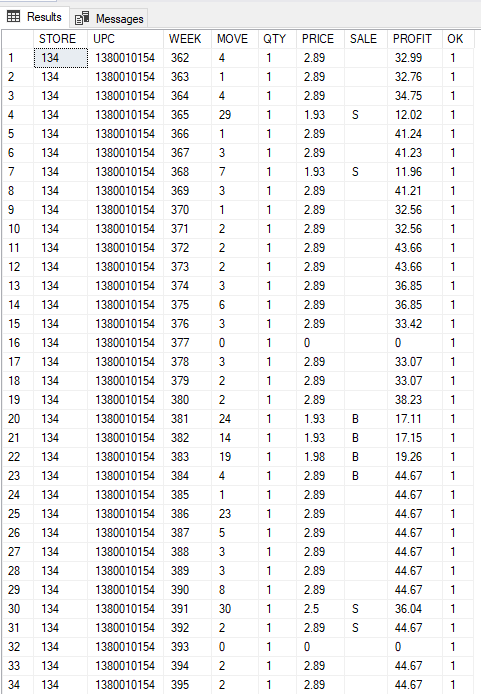
The extraction process will involve loading data from the source files to the staging area. Since we are not going to take care of updates to the data, we are going to use the Full Extraction process. We will take the CCount file and directly import it to the staging area as CCOUNT. For the UPC and Movement files, we will take the files we need for our questions and load them into the staging area. For example, for frozen entrees we will load the UPC file into a table called *product\_frozenentree* and movement table for frozen entrée products into the *move\_frozenentree* table into the staging area. After this, we will combine all such *product\_xxx* tables into a final PRODUCT table and all such *move\_xxx* tables into final SALES table. The extraction of all the store details will be stored in the STORE table. The week and date table provided to us will be loaded into the week table in the staging area.

Additionally, while exporting data from CSV we made sure that the column names are not in “ ”.

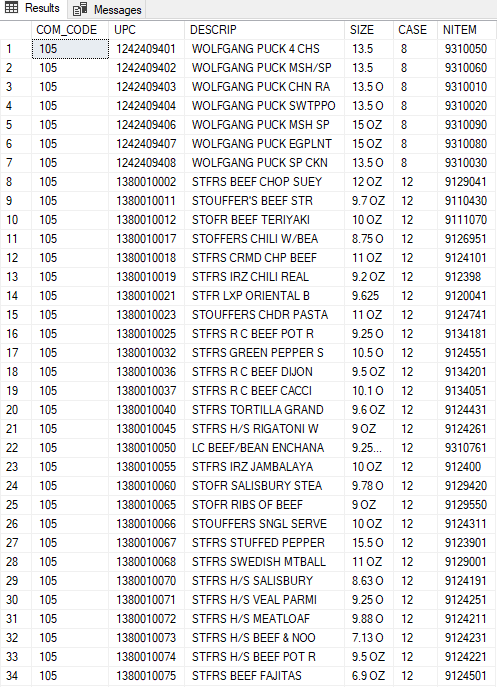
CCount



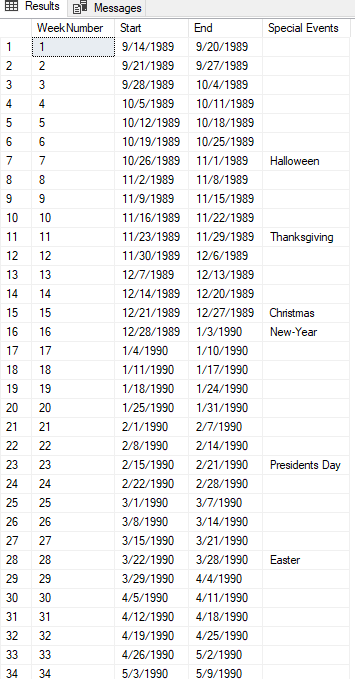
move\_frozenentree



product\_frozenentree



WeekTable



Store



* + 1. **Data Transformation and Cleansing rules**

**CCount data**

1. For CCount, the data we need the data in the format as shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sales ID | StoreID | Date | Category | Sales | Customer Count |
| 1 | 47 | 880101 | GROCERY | 14900 | 1546 |
| 2 | 47 | 880101 | DAIRY | 3321 | 1546 |

For this purpose, we need to apply transformation to this CCount. TODO

**Store data**

1. The store table after extraction to staging area had all the data types as varchar(255)/nvarchar(255). In order to create the dimension table from this, we will convert the value to appropriate datatypes.

2. The Store number will be converted to integer value to suit our needs

3. The other columns were kept as varchar and their sizes will be changed from 255 to needed number

4. Since we did not want to lose any data from this master table, we will add a value ‘None’ to all the column values which were NULL or empty strings. For example, we need the price tier column for one question, so we will change the value to ‘None’ wherever this value was NULL

**Week data**

1. For the week data taken from the Dominick’s data, after extraction into the staging area, we will need to a new week 0 was added into the week table as there is existing data in the sales and movement data

**Product data**

1. After loading the data for the products we need into the staging area, we will use SSIS to integrate all the tables and use derived column to add a new column

2. This new column is the ‘Category’ column we need to distinguish the products into a unique category

3. After this, we will store it into the Product table in the staging area

**Movement data**

1. After loading the data for the Movement we need into the staging area, we will use SSIS to integrate all the tables

2. After this, we will store it into a temporary Move table in the staging area

* + 1. **Aggregate tables**

Our aggregation plans included aggregation in the fact tables as the lowest granularity is week. Data is given date wise for Movement, UPC and CCount. So to bring it to the required granularity level, we will aggregate the data store wise and week wise in our fact tables.

* + 1. **Organization of data staging area**

1. We will load the raw data into tables in the staging area as it is and create tables such as

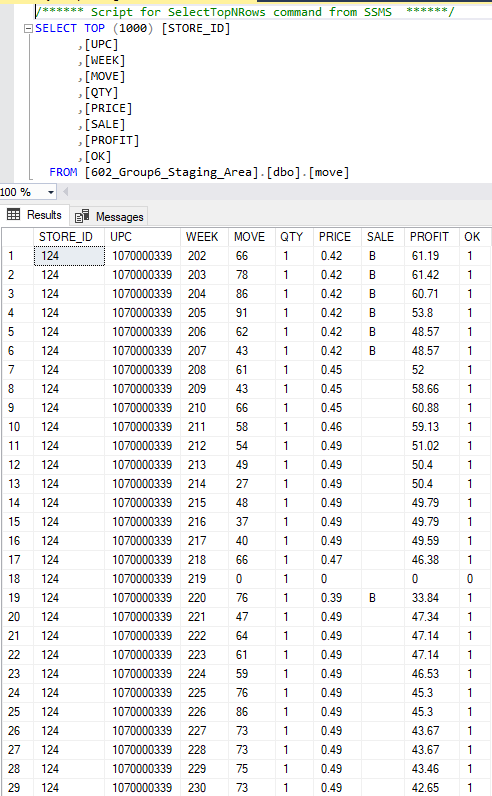
* dbo.CCOUNT
* dbo.move\_frozendinner
* dbo.move\_frozenentree
* dbo.move\_frozenjuice
* dbo.move\_cookie
* dbo.move\_candy
* dbo.product\_frozendinner
* dbo.product\_frozenentree
* dbo. product\_frozenjuice
* dbo. product\_cookie
* dbo. product\_candy

2. After this we will clean all the data, apply transformations, integrate the product tables into one product table, and integrate movement tables into one table to create the sales table

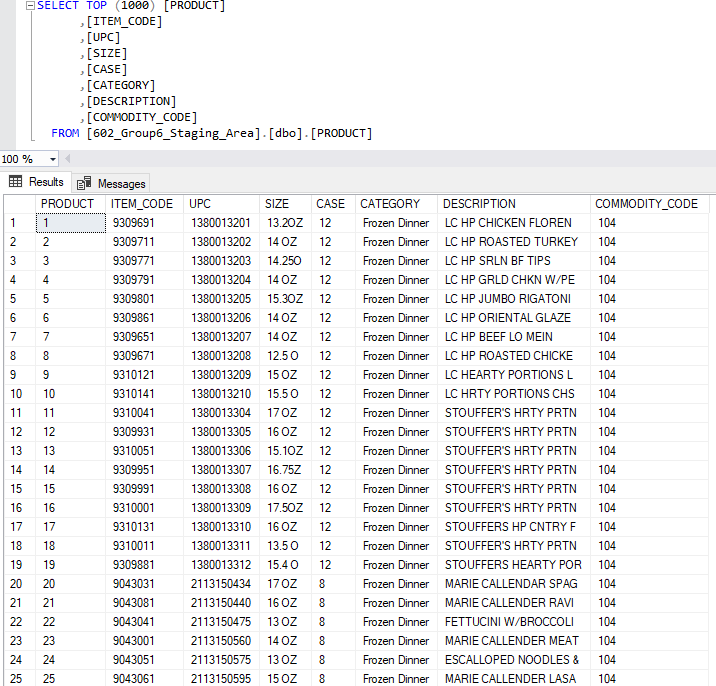
3. In between the processes, we will be using temporary tables to apply transformations and at the end we will drop these temporary tables to clean the staging area

Integrated tables:

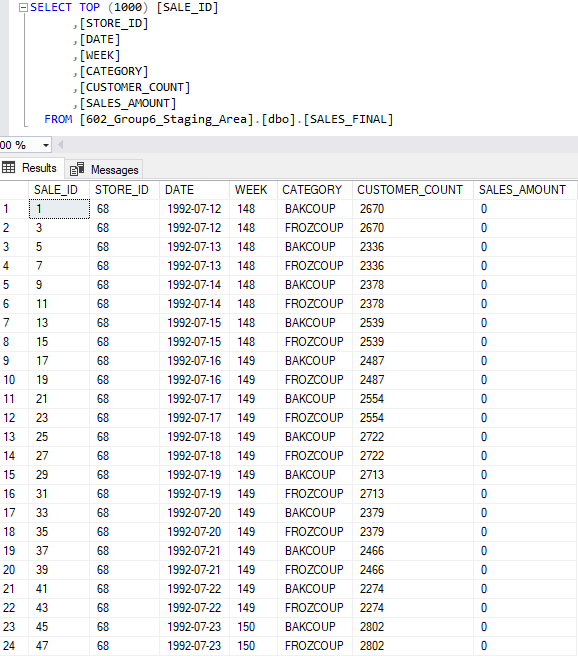
Move



Product

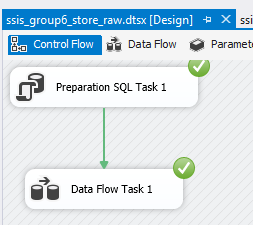
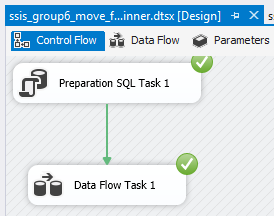


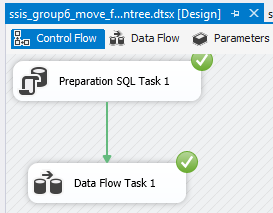
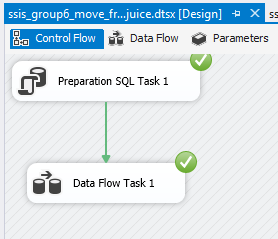
Sales\_Final

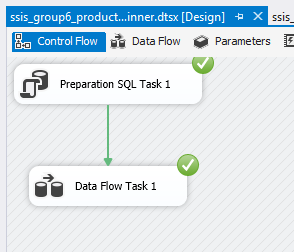
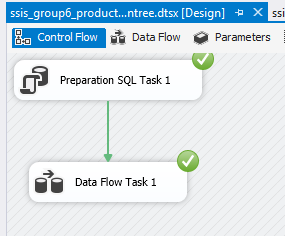


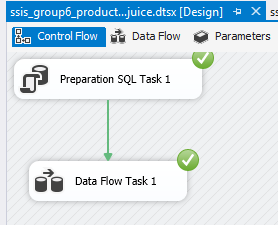
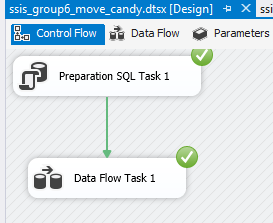
* + 1. **Procedures for all data extractions and loadings**

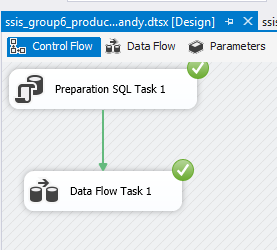
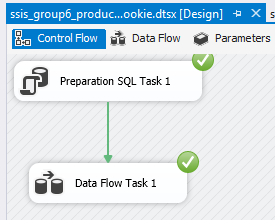
The sources for this project are the csv files given which are UPC, Movement, Ccount, Demographics. Additionally, we have data for Store and Date dimension. The data extractions and load is shown in the SSIS package execution below.

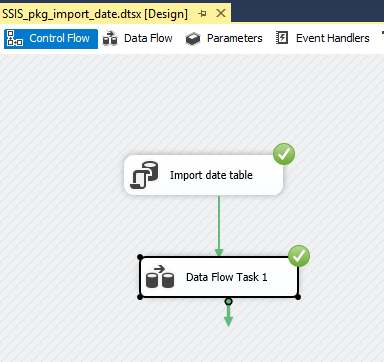
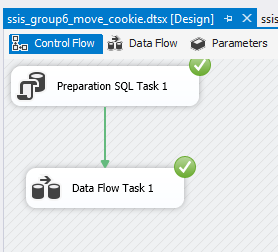
 

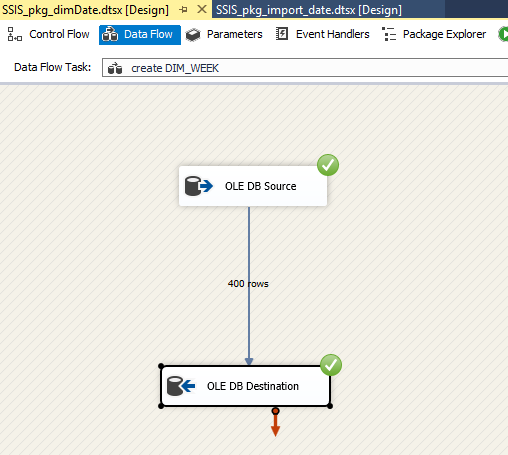
 



* + 1. **ETL for dimension tables**

**1. DIM\_WEEK**

The week dimension is simply created from the WeekTable in the staging area.



CREATE TABLE [DIM\_WEEK]

(

   [WeekNumber] int,

   [Start] date,

   [End] date,

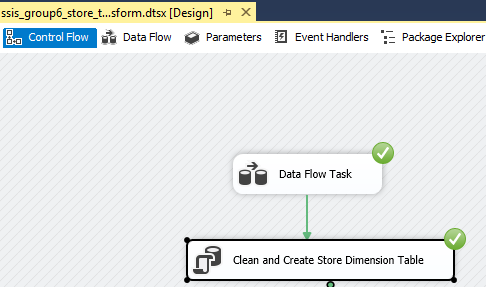
   [Special Events] varchar(50)

)

**SOMETHING MISSING HERE**

**2. DIM\_STORE**

The store dimension is created from the table loaded ‘Store’ into the staging area. Transformations were applied to it before creating the dimension as mentioned in the transformation and cleaning phase above.



UPDATE [STORE]

SET [Price Tier]='None'

WHERE [Price Tier] IS NULL OR [Price Tier] ='';

UPDATE [STORE]

SET [Zone]='None'

WHERE [Zone] IS NULL OR [Zone] ='';

CREATE TABLE DIM\_STORE

(

STORE\_KEY INT NOT NULL PRIMARY KEY IDENTITY(1,1),

STORE\_ID INT NOT NULL,

PRICE\_TIER VARCHAR(10) NOT NULL,

ADDRESS  VARCHAR(100) ,

CITY VARCHAR(50) NOT NULL,

ZONE VARCHAR(10) ,

ZIP\_CODE VARCHAR(10)

);

INSERT INTO DIM\_STORE(STORE\_ID ,PRICE\_TIER ,ADDRESS ,CITY ,ZONE ,ZIP\_CODE )

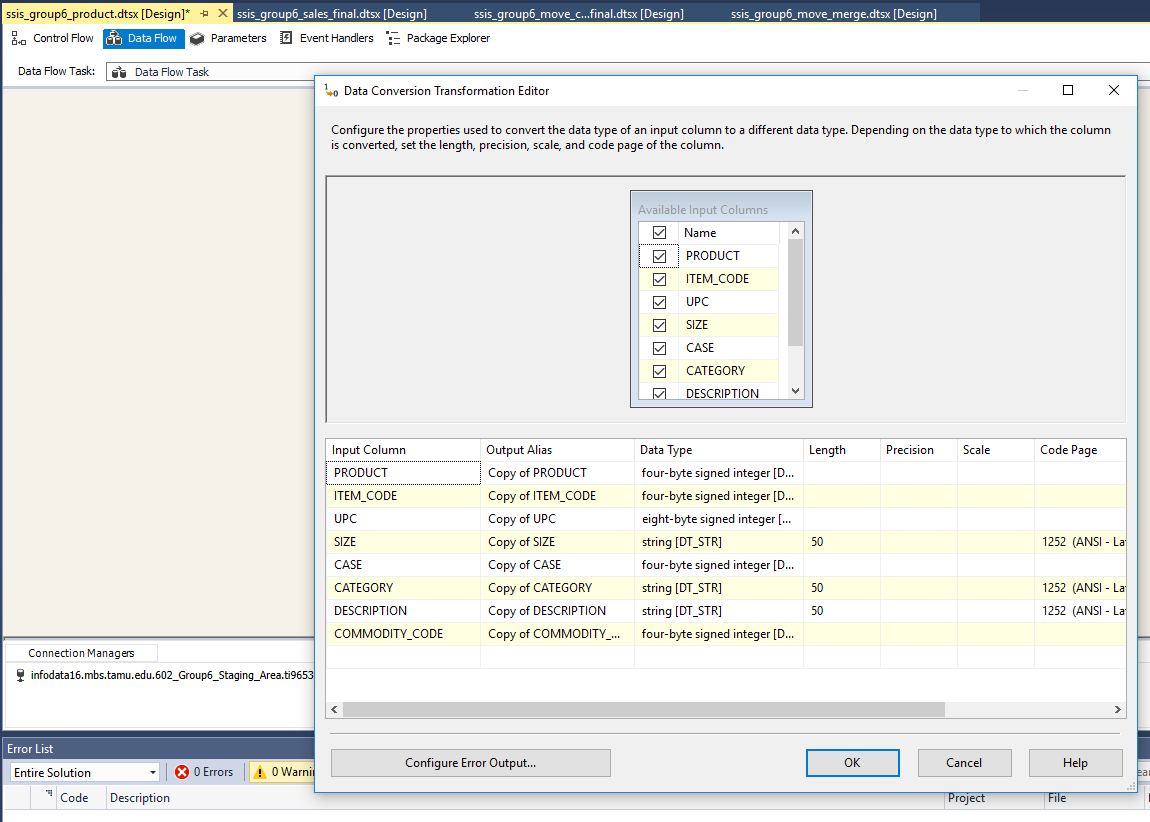
SELECT [Store],[Price Tier], [Address],[City], [Zone],

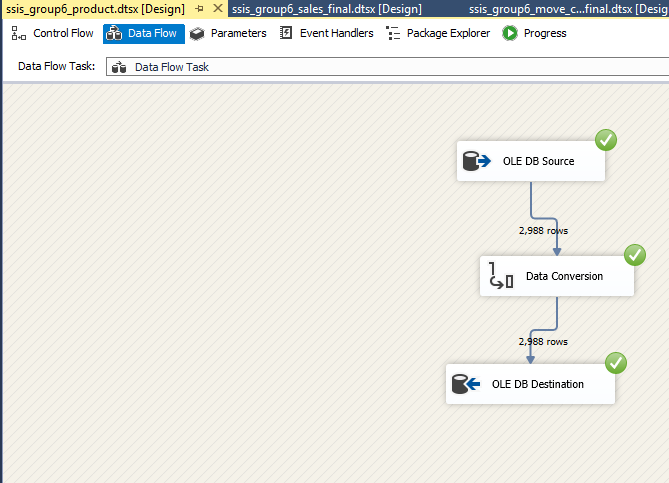
[Zip Code]

FROM STORE

1. **DIM\_PRODUCT**

The product dimension is created from the integrated Product table in the staging area. The data types and length of the columns were changed in the SSIS as shown below.





CREATE TABLE [DIM\_PRODUCT]

(

[PRODUCT\_KEY] int IDENTITY(1,1) PRIMARY KEY,

[PRODUCT] int,

[ITEM\_CODE] bigint,

[UPC] bigint,

[SIZE] varchar(50),

[CASE] int,

[CATEGORY] varchar(50),

[DESCRIPTION] varchar(50),

[COMMODITY\_CODE] int

)

* + 1. **ETL for fact tables**

**10.1 CUSTOMER\_COUNT\_FACT TABLE**

10.1.1 Cleaning the required columns by only taking numerical values in table SALES\_CCF\_TEMP1

CREATE TABLE SALES\_CCF\_TEMP1 (

[STORE\_ID] VARCHAR(50),

[WEEK] VARCHAR(50),

[CUSTOMER\_COUNT] VARCHAR(50),

)

INSERT INTO SALES\_CCF\_TEMP1

(store\_id, [week], customer\_count)

SELECT store, [week], CUSTCOUN

FROM [DBO].[CCOUNT] WHERE [WEEK] NOT LIKE '%[^0-9]%' AND [CUSTCOUN] NOT LIKE '%[^0-9]%'

AND STORE NOT LIKE '%[^0-9]%'

10.1.2. Further cleaned table by converting values in integer and removing wrong values in the table SALES\_CCF\_TEMP2

CREATE TABLE SALES\_CCF\_TEMP2 (

[STORE\_ID] INT,

[WEEK] INT,

[CUSTOMER\_COUNT] INT,

)

INSERT INTO SALES\_CCF\_TEMP2

( store\_id, [week], customer\_count)

SELECT CAST(store\_id AS INT), CAST([week] AS INT), CAST(customer\_count AS FLOAT)

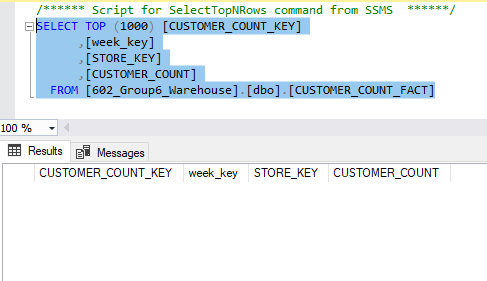
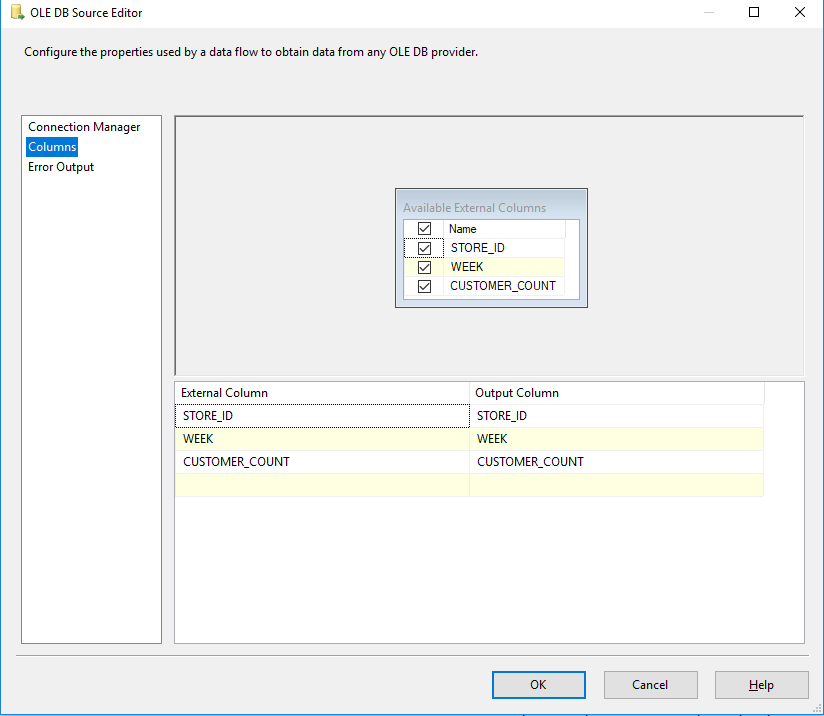
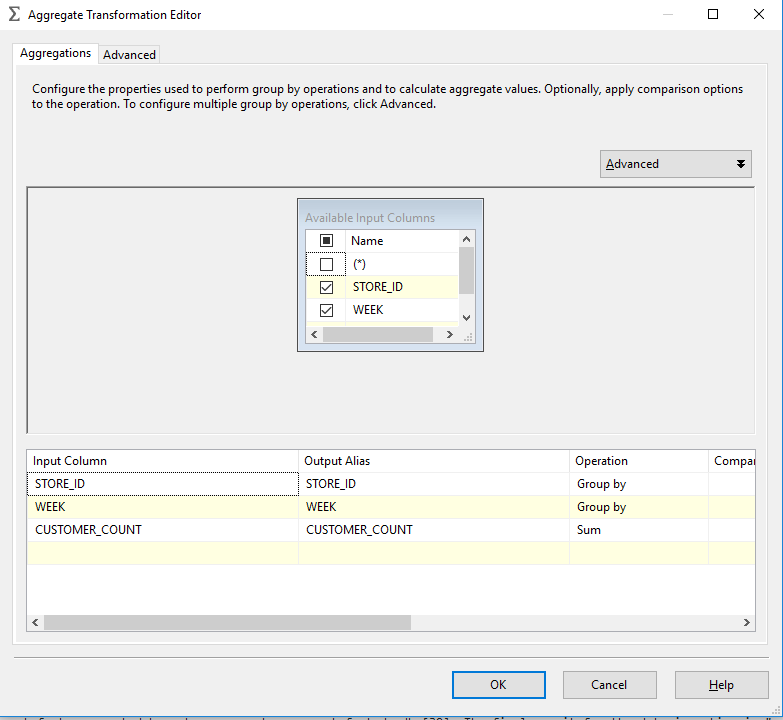
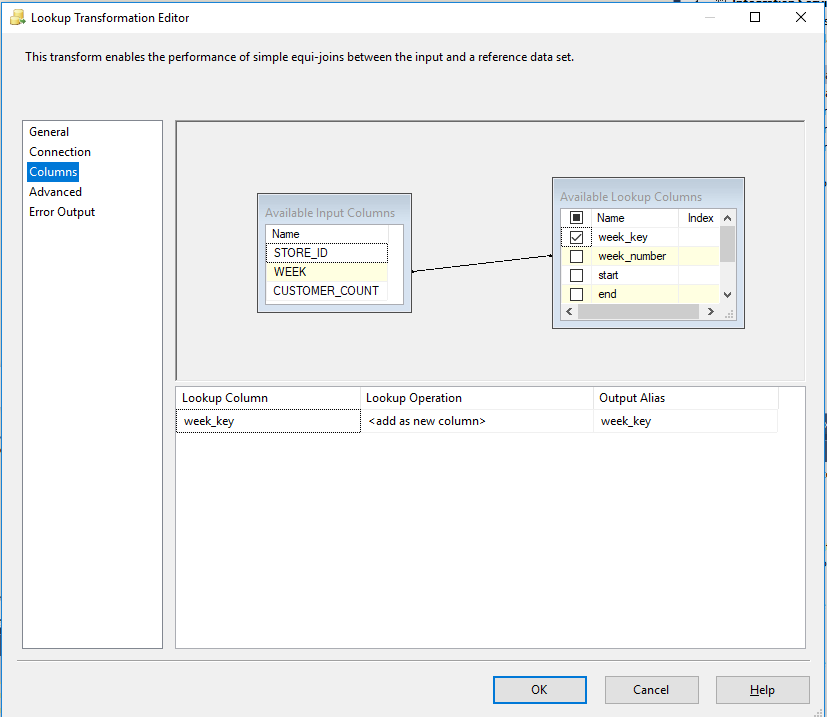
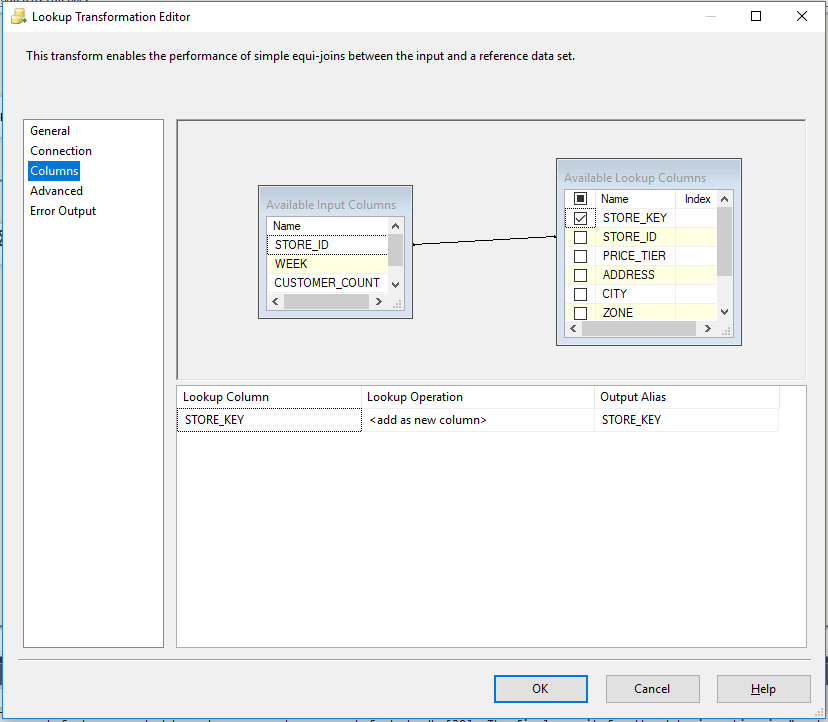
FROM [DBO].[SALES\_CCF\_TEMP1] WHERE cast(STORE\_ID as INT) <140

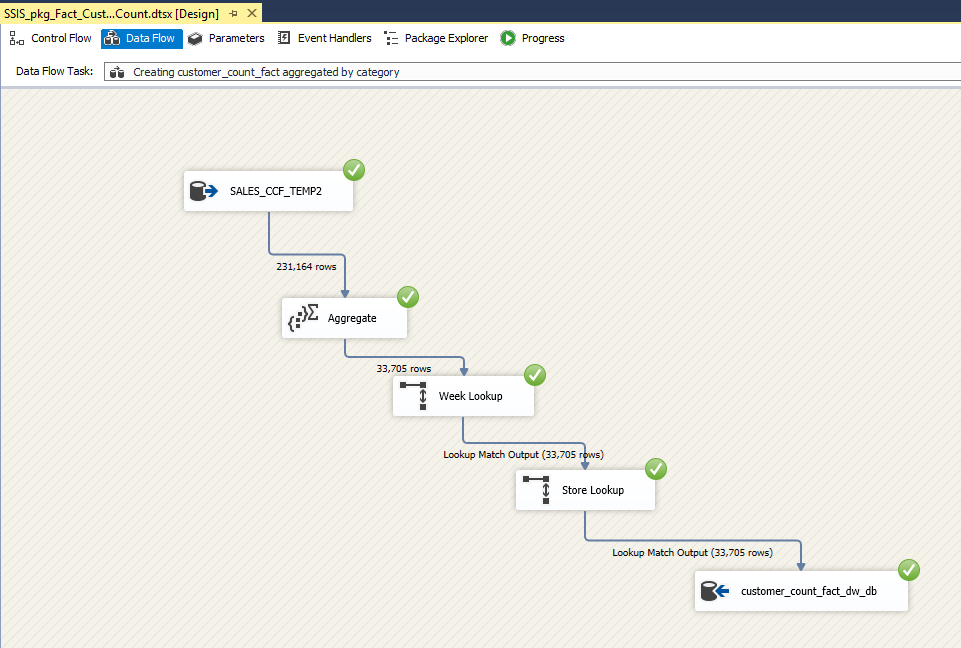
AND STORE\_ID IS NOT NULL

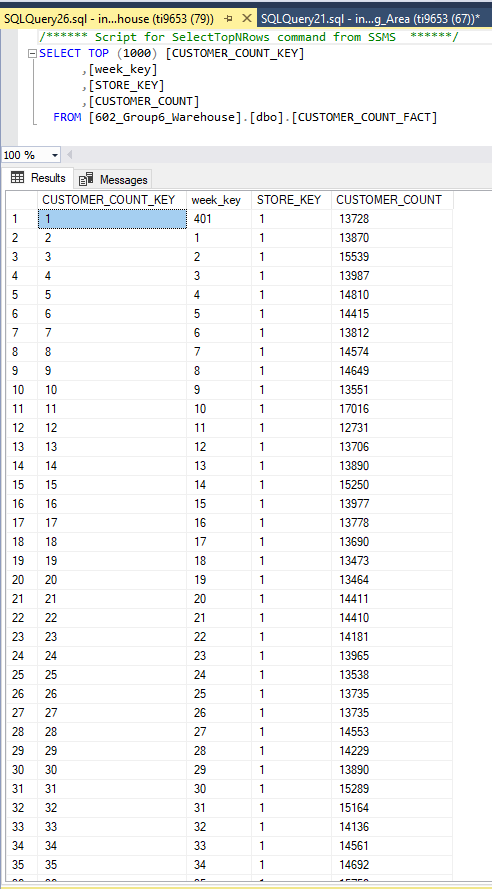
AND WEEK IS NOT NULL

AND CUSTOMER\_COUNT IS NOT NULL

AND STORE\_ID NOT IN ('135','1','0');

* + 1. Data before running SSIS commands  
       
    2. Fetching store\_id, week and customer count from SALES\_CCF\_TEMP2 TABLE  
       
    3. Aggregating data on store\_id and week and by summing the CUSTOMER\_COUNT  
       
    4. Next, performing look up by DIM\_WEEK (on week & WEEK\_NUMBER) and selecting week\_key  
       
    5. Performing look up by DIM\_STORE (on store\_id and store\_id) and selection STORE\_KEY  
       
    6. Final execution of “creating\_customer\_count\_fact”



* + 1. Result after running above SSIS  
       
    2. **List of all temporary tables in the Staging area**

1. 602\_Group6\_Staging\_Area. CCOUNT
2. 602\_Group6\_Staging\_Area.move
3. 602\_Group6\_Staging\_Area.move\_candy
4. 602\_Group6\_Staging\_Area.move\_cookie
5. 602\_Group6\_Staging\_Area.move\_frozendinner
6. 602\_Group6\_Staging\_Area.move\_frozenentree
7. 602\_Group6\_Staging\_Area.move\_frozenjuice
8. 602\_Group6\_Staging\_Area.move\_temp
9. 602\_Group6\_Staging\_Area.PRODUCT
10. 602\_Group6\_Staging\_Area.product\_candy
11. 602\_Group6\_Staging\_Area.product\_cookie
12. 602\_Group6\_Staging\_Area.product\_frozendinner
13. 602\_Group6\_Staging\_Area.product\_frozenentree
14. 602\_Group6\_Staging\_Area.product\_frozenjuice
15. 602\_Group6\_Staging\_Area.SALES
16. 602\_Group6\_Staging\_Area.SALES\_FINAL
17. 602\_Group6\_Staging\_Area.store
18. 602\_Group6\_Staging\_Area.WEEKTABLE

**6. Physical Design**

**1. Aggregation Plan**

We are storing the data at the lowest granularity level of time dimension i.e. week. Of the several types of aggregation, the most common is called a roll-up aggregation. In this case, we will take the customer count and daily sales data given in the Customer Count table and roll it up for weekly sales table. Additionally, we can use roll up for this same data to get the monthly and yearly sales tables. For the sales and profit data, we can roll up the sales amounts from weekly basis to monthly and yearly sales tables. These types of summaries are easily computed from the base data warehouse by using the SQL SUM operator.

Essentially, we can either aggregate at runtime or pre-aggregate the data offline, thus making the totals available without real-time computation. The option of calculation aggregate at runtime will provide with real time data but it may affect the performance. Hence, we have come up with the alternative to write to write SQL to pre-aggregate data according to the dimensions that end users will frequently want to see.

**2. Indexing Plan**

The following columns will be used as index for the corresponding tables. Binary tree is being used to enhance performance of the indexes. Primary key is considered index for all tables.

|  |  |  |
| --- | --- | --- |
| **Indexes** | | |
| **Table** | **Index** | **Indexing schema** |
| DIM\_STORE | STORE\_KEY,STORE\_ID | Binary Tree |
| DIM\_PRODUCT | PRODUCT\_KEY,PRODUCT\_ID |
| DIM\_DATE | TIME\_KEY, WEEK, YEAR |
| COUPON\_REDEMPTION\_FACT | COUPON\_REDEMPTION\_KEY |
| CUST\_COUNT\_FACT | CUST\_COUNT\_KEY |
| SALES\_FACT | SALES\_KEY |

**3. Data Standardization plan**

The following naming standards are to be followed for the warehouse objects:

|  |  |
| --- | --- |
| **Naming Standards** | |
| Fact tables | Ending with "\_FACT" |
| Dimension tables | Starts with "DIM\_" |
| Staging tables | Starts with "STAG\_" |
| Aggregated tables | Starts with "AGGR" |

The following standards for the name length will be followed along with the general guidelines for the naming conventions in Microsoft SQL Server databases:

|  |  |
| --- | --- |
| **Name Length** | |
| Table | 30 |
| Column Headers | 20 |
| Keys | 20 |

**4. Storage plan**

The table below mentions are rough estimate for table sizes with the available data with us. This size is going to vary with cleanup and addition/deletion of tuples. The estimation is done by taking size of data types and multiplying by the data available with the number of rows and columns.

|  |  |
| --- | --- |
| **Warehouse Table Size( in Megabytes)** | |
| DIM\_STORE | ~ 0.30  (3 numeric,3 string, 140 rows) |
| DIM\_PRODUCT | ~ 6  (3 numeric, 6 string, 18k rows) |
| DIM\_TIME | ~ 0.04  (3 numeric, 1 string, 400 rows) |
| COUPONS\_REDEMPTION\_FACT | ~ 10  (6 numeric, 350k rows, divide by 7\*) |
| CUST\_COUNT\_FACT | ~ 7  (4 numeric, 350k rows, divide by 7\*) |
| SALES\_FACT | ~ 2000  (7 numeric, 90M rows) |

Estimated table growth every year: 10 %

\*to get weekly data

|  |  |
| --- | --- |
| **Additional Table Size** | |
| Temp area | ~ 500 MB ( auto grow by 128 MB) |
| Staging area | ~ 2000 MB |
| Indexes | ~ 500 MB |
| OLAP system files | ~ 1500 MB |
| Application | ~ 2000 MB |

Estimated growth every year: 15 %

**5. Partitioning Plan**

The tables to be partitioned:

* DIM\_PRODUCT: Since it has many columns and huge data, this dimension table has to be partitioned vertically.
* SALES\_FACT: Since this table contains millions of rows, this table should be horizontally partitioned.

Corresponding queries need to be optimized for recognizing the partitions

**References**:

[1] <https://www.forbes.com/sites/lauraheller/2014/01/31/this-is-what-a-failed-supermarket-looks-like/#4677ddb54866>

[2] <https://en.wikipedia.org/wiki/Retail>

[3] <https://www.marketingdonut.co.uk/customer-care/the-five-principles-of-retail>

<https://rsmus.com/what-we-do/industries/consumer-products/retail/top-trends-and-issues-for-retail-in-2019.html>

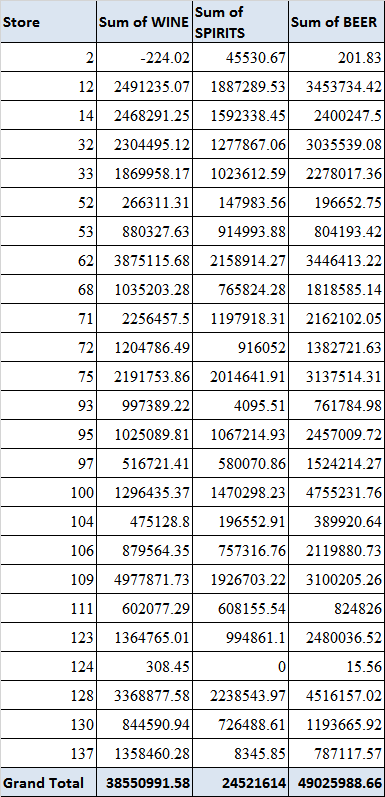
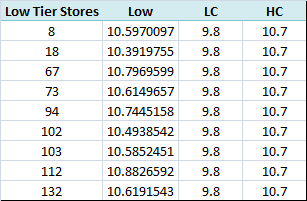
[4] https://www.inc.com/guides/2010/06/defining-your-target-market.html

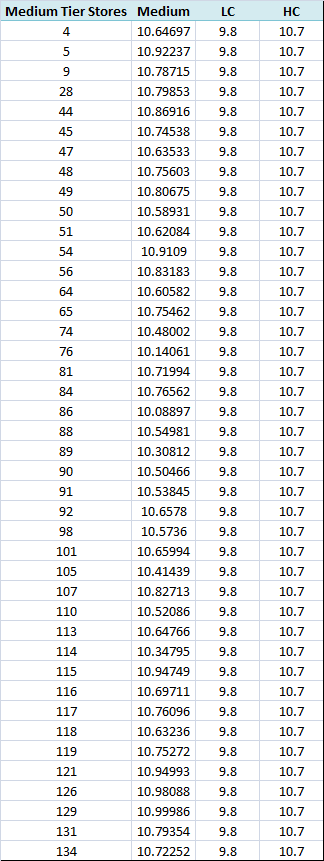
[5] <https://extension.psu.edu/understanding-your-customers-how-demographics-and-psychographics-can-help>

[6] <https://www.forbes.com/sites/rogerdooley/2014/01/28/h-e-b/#48e78c8632b8>

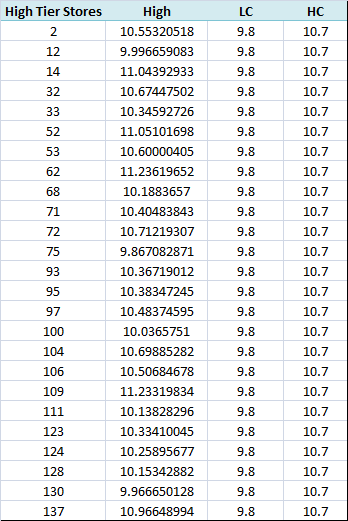
[7] <https://pdfs.semanticscholar.org/ed03/e6a1f4fb6fe0b185938e45ef69dc64b4bb69.pdf>)

**Appendix**

* 1. 
  2. 



1.4



1.5

|  |  |
| --- | --- |
| Promotion Type | Average of Profit Earned |
| B | 11.78158661 |
| S | 19.06941434 |
| Grand Total | 14.04792915 |

1.6

|  |  |  |
| --- | --- | --- |
| Year | Sum of Sale | Sum of Profit Earned |
| 89 | 6999.22 | 3376.3228 |
| 90 | 6651.263 | 2535.38212 |
| 91 | 5901.45 | 2404.6478 |
| 92 | 5754.04 | 2654.70915 |
| 93 | 7104.81 | 3005.81068 |
| 94 | 5910.68 | 2750.00918 |
| 95 | 4712.39 | 2063.50582 |
| 96 | 5086.38 | 2192.1153 |
| Grand Total | 48120.23 | 20982.5028 |

1.7

|  |  |  |  |
| --- | --- | --- | --- |
| STORE | NOCAR | Sum of Bulk | Zip |
| 2 | 0.124603 | 429527.34 | 60305 |
| 4 | 0.055567 | 204587.75 | 60068 |
| 5 | 0.02557 | 2892369.13 | 60067 |
| 8 | 0.075113 | 3237831.36 | 60453 |
| 9 | 0.040128 | 2981199.02 | 60053 |
| 12 | 0.483518 | 2719576.21 | 60660 |
| 14 | 0.026586 | 1043915.44 | 60025 |
| 18 | 0.141975 | 2764367.79 | 60171 |
| 21 | 0.017598 | 2158103.24 | 60103 |
| 28 | 0.054855 | 1183723.9 | 60056 |
| 32 | 0.071701 | 3681720.42 | 60068 |
| 33 | 0.506224 | 288046.5 | 60657 |
| 40 | 0.04633 | 2376585.12 | 60455 |
| 44 | 0.040766 | 2399007.66 | 60558 |
| 45 | 0.020232 | 978547.28 | 60090 |
| 47 | 0.021297 | 1686834.91 | 60101 |
| 48 | 0.021209 | 363853.96 | 60193 |
| 49 | 0.054382 | 610211.83 | 60515 |
| 50 | 0.036434 | 981863.92 | 60457 |
| 51 | 0.025436 | 2272079.47 | 60463 |
| 52 | 0.0149 | 2948795.99 | 60062 |
| 53 | 0.145363 | 558740.95 | 60662 |
| 54 | 0.02084 | 661770.39 | 60540 |
| 56 | 0.031997 | 2151294.25 | 60525 |
| 59 | 0.031318 | 2323421.5 | 60014 |
| 62 | 0.026528 | 770042.09 | 60093 |
| 64 | 0.043621 | 451588.23 | 60181 |
| 65 | 0.015285 | 403082.49 | 60103 |
| 67 | 0.045029 | 2190976.5 | 60521 |
| 68 | 0.305256 | 489008.12 | 60625 |
| 70 | 0.081126 | 3151661.59 | 60435 |
| 71 | 0.153432 | 2102623.47 | 60546 |
| 72 | 0.067621 | 988764.55 | 60646 |
| 73 | 0.133319 | 3095726.85 | 60629 |
| 74 | 0.143645 | 3344568.01 | 60634 |
| 75 | 0.550655 | 390182.56 | 60640 |
| 76 | 0.347964 | 2667431.82 | 60618 |
| 77 | 0.016868 | 2192052.98 | 60061 |
| 78 | 0.01387 | 2750202.77 | 60516 |
| 80 | 0.030875 | 3110142.36 | 60005 |
| 81 | 0.033181 | 2906096.69 | 60056 |
| 83 | 0.043726 | 3295128.98 | 60438 |
| 84 | 0.01332 | 2089748.49 | 60462 |
| 86 | 0.353421 | 3026260.71 | 60618 |
| 88 | 0.049691 | 2010013.5 | 60106 |
| 89 | 0.284283 | 1253452.23 | 60632 |
| 90 | 0.180875 | 2085018.05 | 60617 |
| 91 | 0.072177 | 1698714.88 | 60453 |
| 92 | 0.027058 | 2112668.14 | 60429 |
| 93 | 0.238513 | 1400036.6 | 60202 |
| 94 | 0.012373 | 496998.87 | 60108 |
| 95 | 0.25188 | 1539443.4 | 60634 |
| 97 | 0.084652 | 1523557.9 | 60506 |
| 98 | 0.149392 | 3663537.64 | 60638 |
| 100 | 0.36565 | 3326345.95 | 60608 |
| 101 | 0.06103 | 3094720.3 | 60016 |
| 102 | 0.114071 | 4172533.81 | 60655 |
| 103 | 0.015063 | 1906286.39 | 60439 |
| 104 | 0.034121 | 3108703.1 | 60174 |
| 105 | 0.092157 | 2107330.91 | 60160 |
| 106 | 0.05823 | 1968248.43 | 60538 |
| 107 | 0.042052 | 3273847.15 | 60154 |
| 109 | 0.030239 | 4252568.86 | 60015 |
| 110 | 0.056954 | 2700349.34 | 60118 |
| 111 | 0.334262 | 591340.05 | 60620 |
| 112 | 0.013949 | 2689874.58 | 60090 |
| 113 | 0.144444 | 3467796.24 | 60646 |
| 114 | 0.085277 | 2634598.13 | 60409 |
| 115 | 0.018764 | 2390132.88 | 60540 |
| 116 | 0.060982 | 2549918.33 | 60126 |
| 117 | 0.018772 | 1617513.59 | 60193 |
| 118 | 0.081784 | 1610655.9 | 60053 |
| 119 | 0.015622 | 1670426.1 | 60089 |
| 121 | 0.023876 | 3192807.64 | 60514 |
| 122 | 0.019283 | 3617500.14 | 60194 |
| 123 | 0.241944 | 2859046.69 | 60630 |
| 124 | 0.29272 | 3096226.07 | 60302 |
| 126 | 0.02345 | 2983417.63 | 60187 |
| 128 | 0.340148 | 2777635.09 | 60645 |
| 129 | 0.024509 | 2168587.81 | 60047 |
| 130 | 0.407352 | 2055944.68 | 60649 |
| 131 | 0.037447 | 2455545.95 | 60008 |
| 132 | 0.032774 | 2761699.55 | 60443 |
| 134 | 0.021372 | 1256478.9 | 60185 |
| 137 | 0.104094 | 638306.03 | 60201 |
| 301 | 0.013213 | 3833573.14 | 60462 |
| 302 | 0.036081 | 3115662.18 | 60139 |
| 303 | 0.275086 | 3090437.25 | 60650 |
| 304 | 0.411329 | 3325924.31 | 60614 |
| 305 | 0.051017 | 2309175.39 | 60050 |
| 306 | 0.064158 | 3057769.01 | 60648 |
| 307 | 0.039286 | 2322793.29 | 60073 |
| 308 | 0.113115 | 1745790.84 | 60160 |
| 309 | 0.047555 | 2764393.03 | 60504 |
| 310 | 0.030871 | 2145682.99 | 60004 |
| 312 | 0.063657 | 2230005.83 | 60445 |
| 315 | 0.049952 | 2839280.8 | 60455 |

1.8

|  |  |
| --- | --- |
| Year & Tiers | Sum of MEAT |
| 89 | 153880949.1 |
| High | 49851107.74 |
| Low | 25324790.45 |
| Medium | 78705050.92 |
| 90 | 154609983.4 |
| High | 50169964.99 |
| Low | 25323391.25 |
| Medium | 79116627.11 |
| 91 | 151547740.1 |
| High | 50898534.76 |
| Low | 23506119.49 |
| Medium | 77143085.81 |
| 92 | 141159609.7 |
| High | 47582555.42 |
| Low | 20635103.03 |
| Medium | 72941951.23 |
| 93 | 132774657.7 |
| High | 45533074.56 |
| Low | 18825978.34 |
| Medium | 68415604.8 |
| 94 | 127200794.3 |
| High | 43113963.68 |
| Low | 17768582.22 |
| Medium | 66318248.36 |
| 95 | 120168615.3 |
| High | 40168590.55 |
| Low | 18349791.36 |
| Medium | 61650233.39 |
| 96 | 119899745.8 |
| High | 40344226.99 |
| Low | 18457280.75 |
| Medium | 61098238.07 |
| 97 | 39132831.16 |
| High | 13483313.47 |
| Low | 6278639.57 |
| Medium | 19370878.12 |
| Grand Total | 1140374926 |

1.9

|  |  |  |
| --- | --- | --- |
| Bundle | Average of Sales | Average of Profit Earned |
| 1 | 65.27789 | 8.260493 |
| 2 | 119.4916 | 12.00605 |
| 3 | 528.9431 | 163.9724 |
| 4 | 172.9482 | -3.41767 |
| 6 | 328.9628 | 7.251246 |
| 7 | 50.37261 | 10.03501 |
| Grand Total | 65.90712 | 8.335087 |

1.10

|  |  |  |
| --- | --- | --- |
| UPC | Average of Sales | Average of Profit Earned |
| 1410007012 | 4.384428 | 1.226138 |
| 1410007016 | 3.633641 | 0.945096 |
| 1410007056 | 2.311949 | 0.534848 |
| 1410007070 | 0.608514 | 0.102474 |
| 1410007174 | 1.489408 | 0.389054 |
| 1410007188 | 9.442057 | 2.432816 |
| 1410007219 | 5.11865 | 1.31273 |
| 1410007223 | 0.547326 | 0.134451 |
| 1410007233 | 17.60341 | 4.552681 |
| 1410007234 | 0.637829 | 0.156388 |
| 1410007297 | 2.63538 | 0.740889 |
| 1410007402 | 5.18476 | 1.395821 |
| 1410007403 | 0.429607 | 0.137603 |
| 1410007404 | 0.997156 | 0.256517 |
| 1410007405 | 5.7673 | 1.594558 |
| 1410007406 | 3.518518 | 0.982424 |
| 1410007407 | 0.221964 | 0.071095 |
| 1410007408 | 6.707648 | 1.791357 |
| 1410007410 | 2.685742 | 0.741502 |
| 1410007411 | 3.270521 | 0.909736 |
| 1410007412 | 20.62517 | 5.474156 |
| 1410007413 | 1.702379 | 0.491922 |
| 1410007414 | 1.536171 | 0.439979 |
| 1410007415 | 1.207981 | 0.346171 |
| 1410007416 | 1.480713 | 0.425516 |
| 1410007419 | 6.257317 | 1.697188 |
| 1410007420 | 2.574415 | 0.738901 |
| 1410007421 | 0.949296 | 0.274452 |
| 1410007422 | 0.853159 | 0.249834 |
| 1410007424 | 1.313253 | 0.376025 |
| 1410007429 | 1.82841 | 0.517323 |
| 1410007430 | 1.855331 | 0.519711 |
| 1410007431 | 2.662865 | 0.75069 |
| 1410007433 | 1.044559 | 0.331283 |
| 1410007434 | 2.546091 | 0.707441 |
| 1410007435 | 8.050003 | 2.158158 |
| 1410007436 | 2.117222 | 0.591408 |
| 1410007438 | 5.880256 | 1.599169 |
| 1410007440 | 2.179403 | 0.622918 |
| 1410007441 | 6.041874 | 1.698873 |
| 1410007443 | 1.248802 | 0.365389 |
| 1410007444 | 3.793344 | 1.036033 |
| 1410007445 | 3.415281 | 0.922931 |
| 1410007446 | 1.289892 | 0.338155 |
| 1410007448 | 5.616146 | 1.527805 |
| 1410007457 | 6.681319 | 1.591235 |
| 1410007462 | 4.813176 | 1.338268 |
| 1410007464 | 5.59313 | 1.537961 |
| 1410007471 | 3.483946 | 1.346603 |
| 1410007472 | 16.72433 | 4.393285 |
| 1410007478 | 2.025158 | 0.533507 |
| 1410007481 | 1.617164 | 0.419084 |
| 1410007482 | 6.251552 | 1.617076 |
| 1410007505 | 1.143223 | 0.29432 |
| 1410007506 | 1.045518 | 0.26883 |
| 1410007514 | 1.759799 | 0.492661 |
| 1410007515 | 1.902872 | 0.477356 |
| 1410007516 | 1.045954 | 0.255587 |
| 1410007517 | 1.715455 | 0.433174 |
| 1410007518 | 2.036305 | 0.515899 |
| 1410007522 | 7.154801 | 1.874786 |
| 1410007523 | 2.315022 | 0.645995 |
| 141000752 | 1.736384 | 0.484798 |
| Grand Total | **4.5649** | **1.2266** |

1.11

|  |  |  |
| --- | --- | --- |
| Store | Bakery sales in Dollars | Bakery Coupons Redeemed |
| 0 | 219848.2 | 1949513 |
| 1 | 164017 | 1309851 |
| 2 | 5496520 | 682590.2 |
| 3 | 59118.35 | 385019.7 |
| 4 | 1709304 | 223054.4 |
| 5 | 5102597 | 79931.5 |
| 6 | 5403.62 | 37551.05 |
| 7 | 3106.9 | 21653.23 |
| 2644 | 218.95 | 1083.38 |

1.12

