

**DESIGN AND IMPLEMENTATION OF A DATA WAREHOUSE**

Joint Consulting Project

**Report 2**

**Team Members**

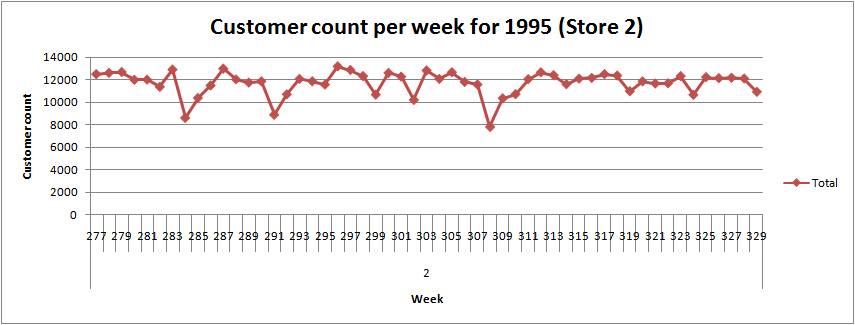
Naresh Choudhary

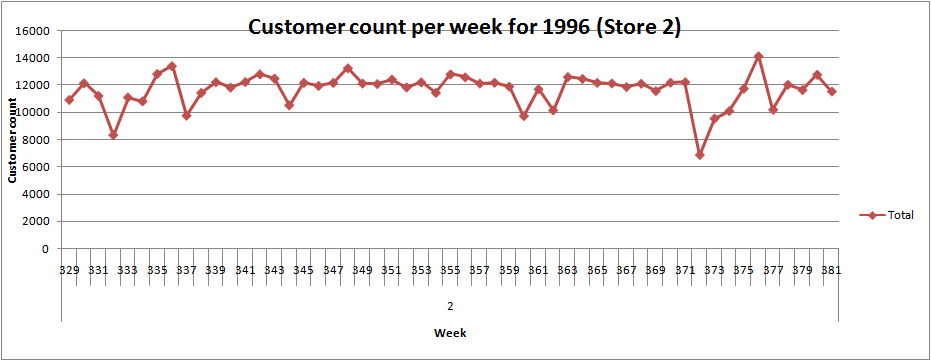
Pallavi Tiwari

Chetan Joshi

1. **Business Questions and their Dimension models**
2. **What is weekly trend of customer in traffic for store 2 between 1995 and 1996?**

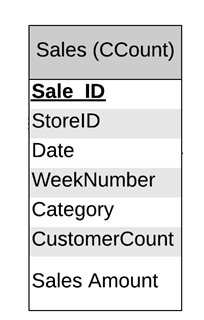
**Pivot:**

****

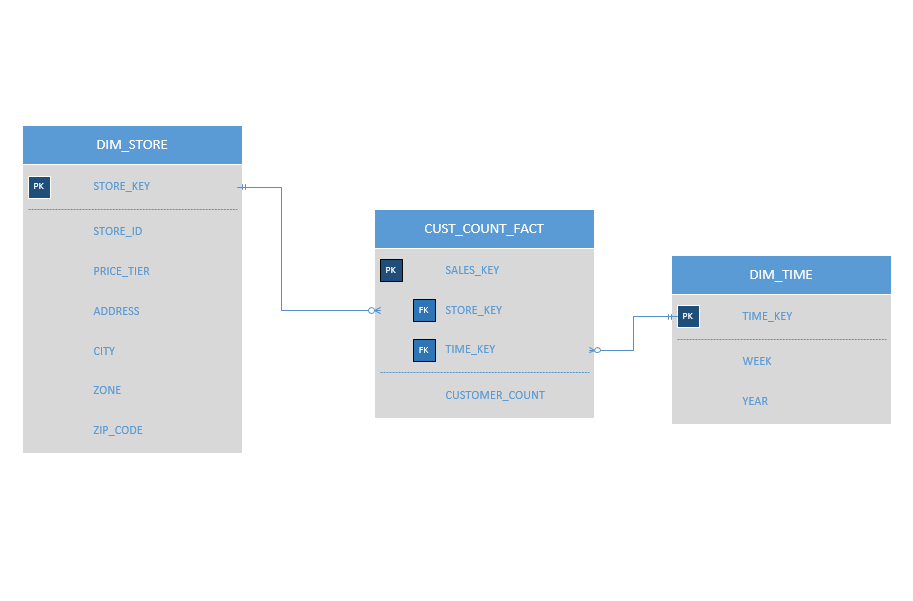
****

**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:**

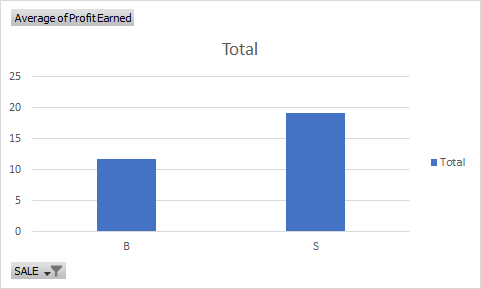


**Schema Justification:** To calculate customer count on a specific store on weekly basis, we need two dimensions namely *DIM\_TIME* and *DIM\_STORE*. Time dimension will aggregate the data on weeks and year. Similarly store dimension will help to find customer count on different stores.

*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)*

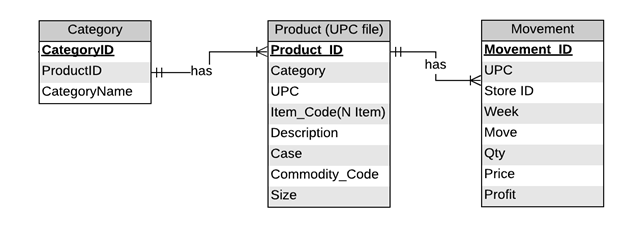
**2. In category of frozen products, which method of discount (Coupon or price reduction) gives more profit?**

**Pivot:**

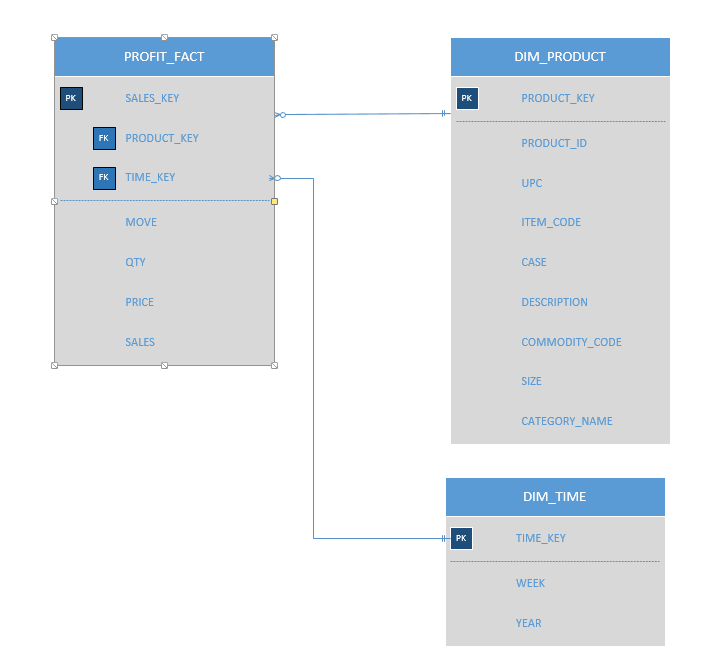


**Schema Justification**: Due to lower shelf life of frozen foods they are often sold on discounted rates to avoid loss. Analysis of profit earned in category frozen entree shows that when simple price reduction is offered, sale and profit is more. Dominick can consider offering simple price reduction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

**ERD:**



**Dimensional Model:**



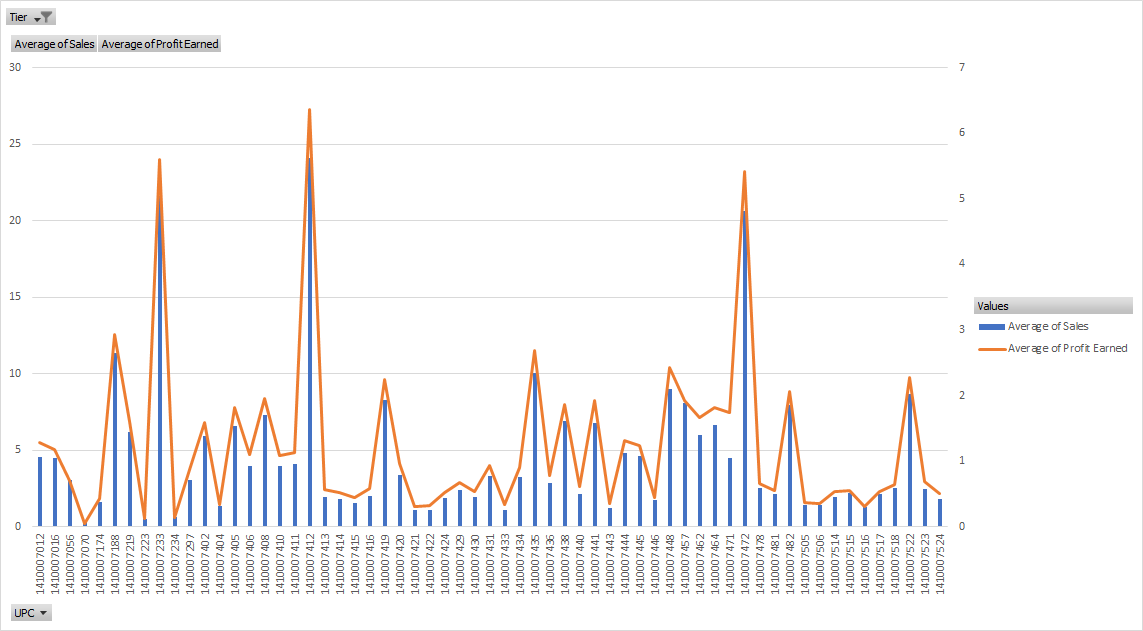
*1. Q1 Dimensional Model*

**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.

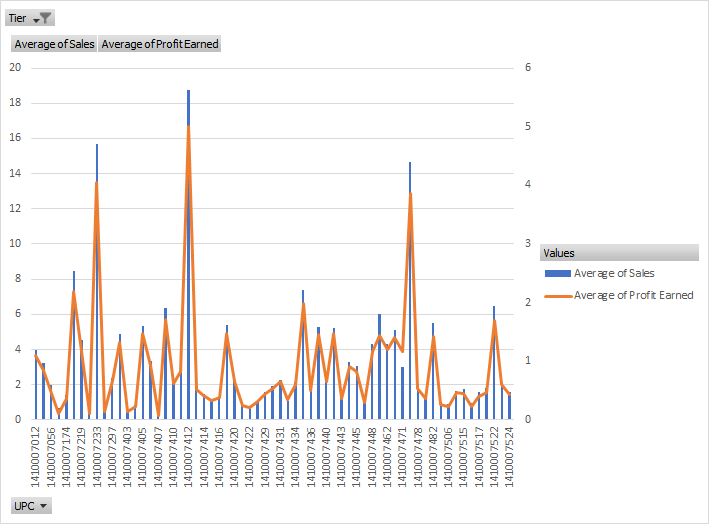
Data in PROFIT\_FACT table comes from movement table/movement file. Since ERD movement table (or movement file from Dominick) records data on a daily basis, sales will be aggregated week wise.

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

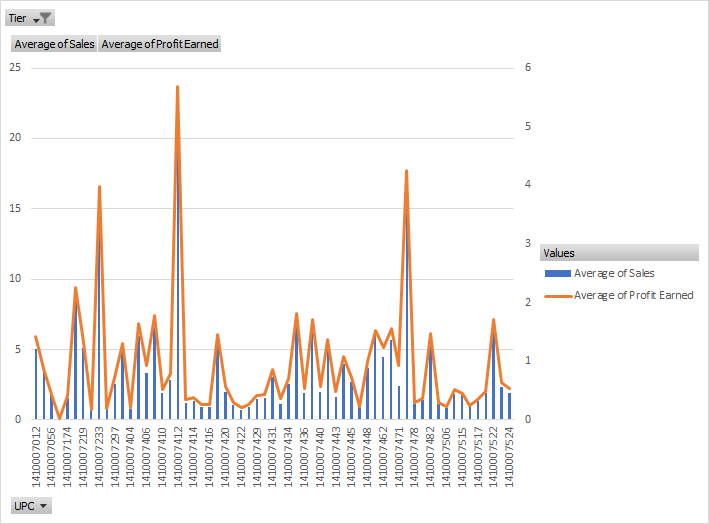
**Pivot (High tier stores):**



**Pivot (Low tier stores):**

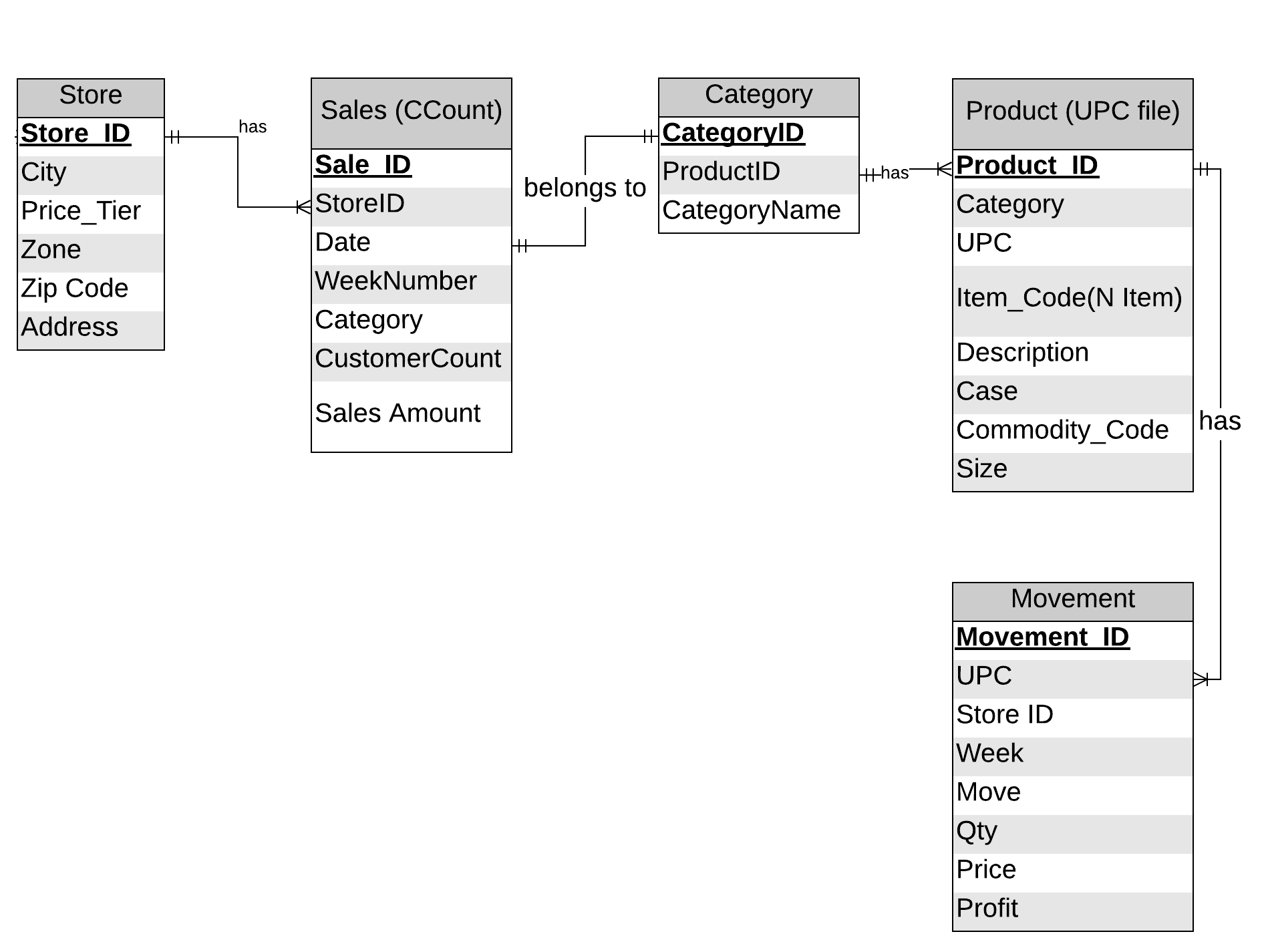
****

**Pivot (Medium tier stores):**

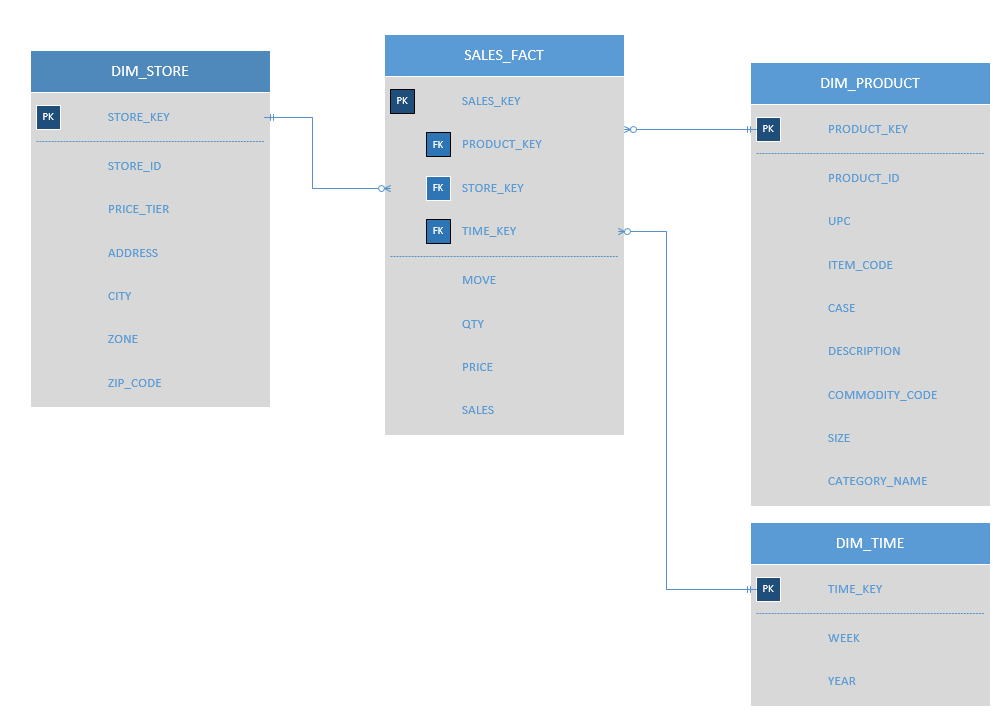
****

**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:**

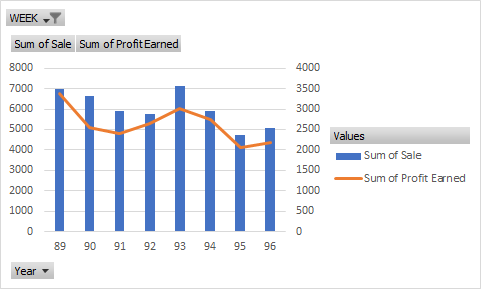


*2. Q2 Dimensional Model*

**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.

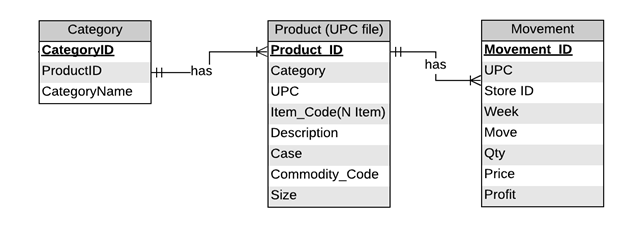
**4. What is the trend of candies’ sale during Halloween year by year?**

**Pivot:**

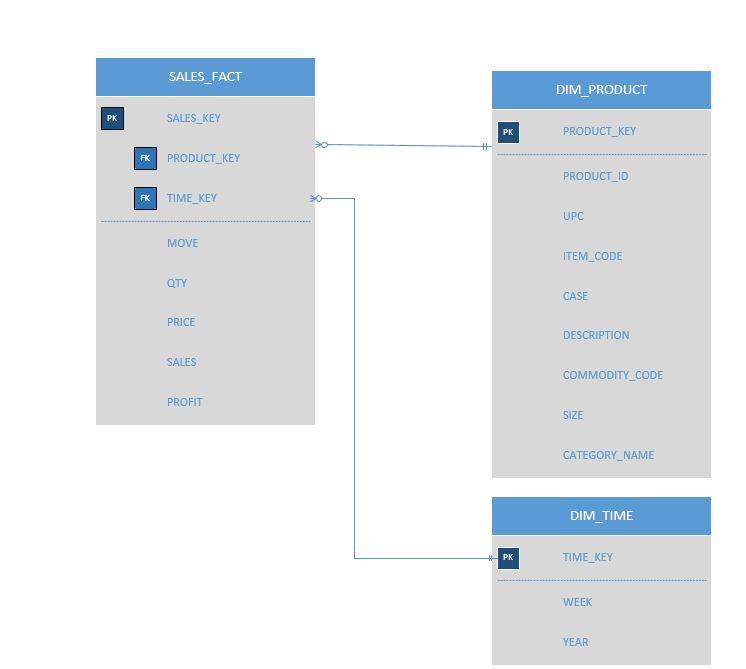


**Schema Justification**: Graph shows the trend of candies during Halloween season from 89 to 96. It can help to predict sales of next year which could range between 4000-5000. This trend can be validated with data from years to come.

**ERD:**



**Dimensional Model:**

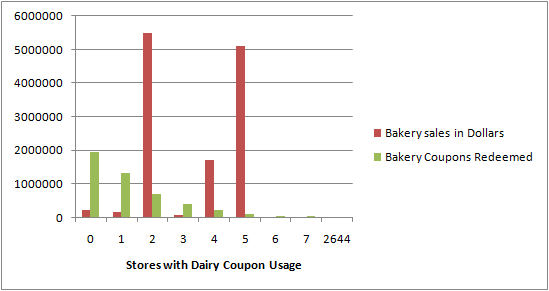


*3. Q3 Dimensional Model*

**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.

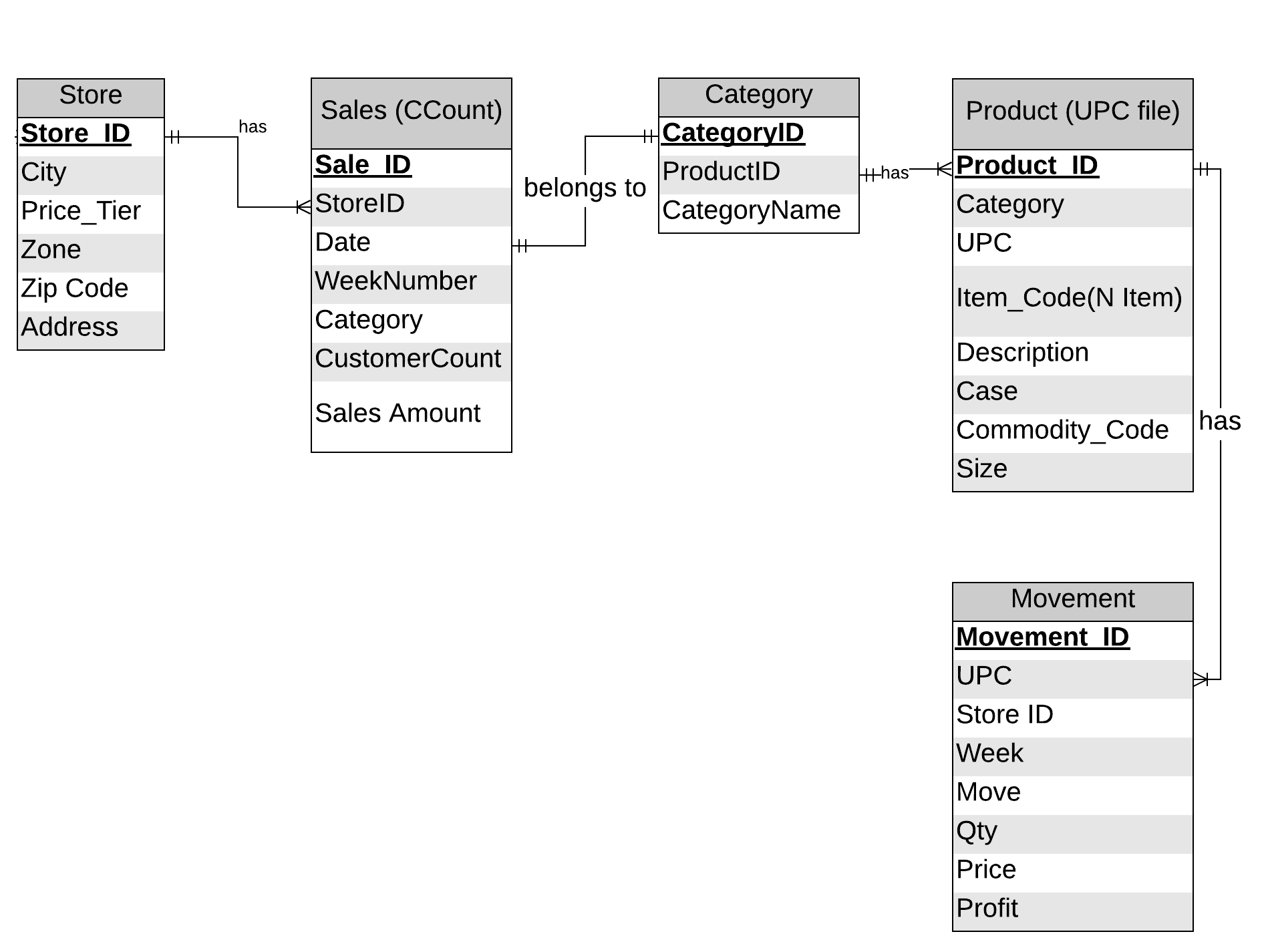
**5. Find out the relation between Bakery sales and bakery coupons redeemed for all stores?**

**Pivot**

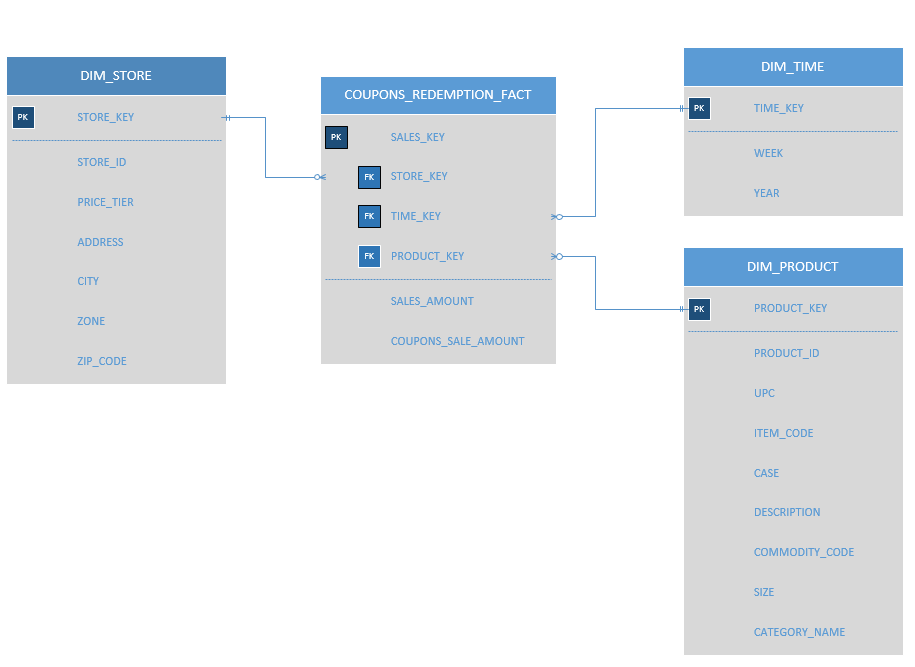


**Schema Justification:** In this business question, we looked at the data given in the CCount table for bakery sales. We wanted to evaluate this amount store wise to see if there is any relevance of the coupons redeemed for bakery items. We noticed that for certain stores such as 0, 1, 2 as shown above, 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:**



*4. Q4 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.

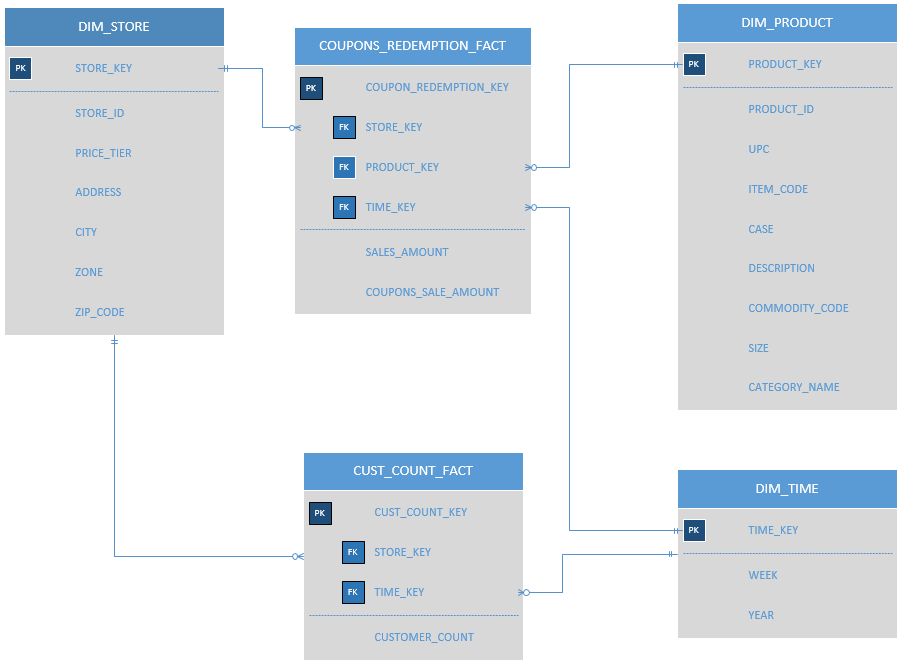
**Kimball’s Matrix for Data Marts:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. **Data Warehouse Schema**

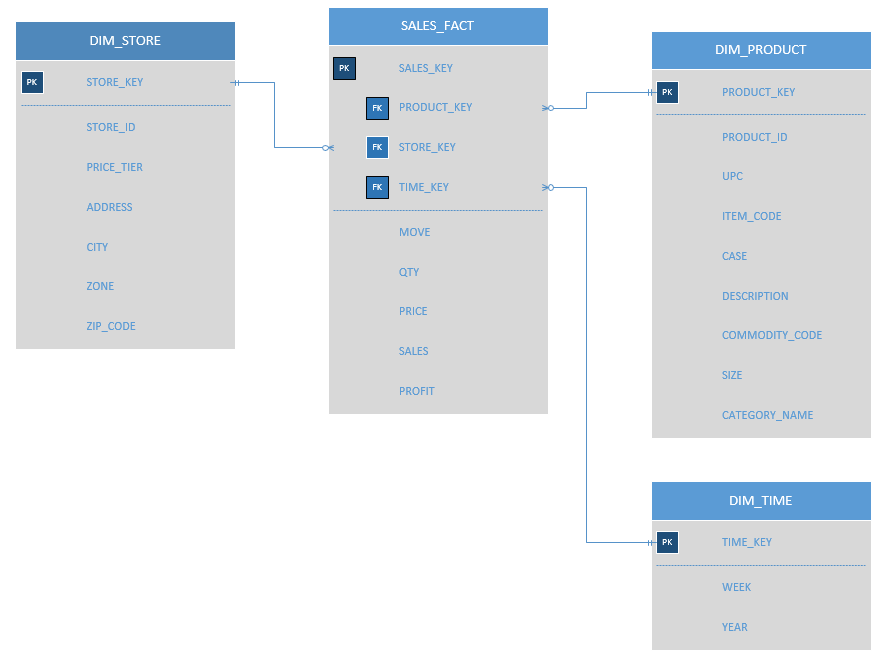
**1. Coupon sales & Customer Count Data Mart:**



1. *Star Schema for 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:**



1. *Star Schema for 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

1. **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. **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