# ETL Development Plan

Based upon the previous reports developed, ETL Development Plan is designed to outline the roadmap for the data load process into the data warehouse.

The proposed plan is presented as:

1. Determine the target data
2. Determine the source data
3. Mapping tables for staging and data mart loads
4. Comprehensive data extraction rules
5. Data staging area and screen shots
6. Data transformation and cleansing rules
7. Plan for aggregate tables
8. Procedures for data extraction and loading
   1. ETL for dimension tables
   2. ETL for fact tables

The implementation is shown as:

1. Mapping definition describing the source to end table for all dimension and fact tables
2. SQL statements used for the ETL operations
3. Before and after table screen shots

## Determine the target data

Our proposed dimensional model contains 5 dimension tables and 2 fact tables. Please find the definitions for each of them at Data Warehouse area below.

**Dimensional Tables**

|  |  |  |
| --- | --- | --- |
| **Dimension: Product** | | |
| **DW Target Table** | **DW Target Column** | **Target Datatype** |
| 601-Group11-DW-Area.PRODUCT\_DIM | PRODUCT\_ID | int |
| UPC\_CODE | varchar |
| PRODUCT\_CATEGORY | varchar |
| ITEM\_CODE | varchar |

|  |  |  |
| --- | --- | --- |
| **Dimension: Demographic** | | |
| **DW Target Table** | **DW Target Column** | **Target Datatype** |
| 601-Group11-DW-Area.DEMOGRAPHIC\_DIM | DEMO\_ID | int |
| INCOME\_LEVEL | varchar |
| POOR\_% | float |
| BELOW\_9\_% | float |
| ABOVE\_60\_% | float |

|  |  |  |
| --- | --- | --- |
| **Dimension: Coupon** | | |
| **DW Target Table** | **DW Target Column** | **Target Datatype** |
| 601-Group11-DW-Area.COUPON\_DIM | COUPON\_ID | int |
| COUPON\_REDEEMED | int |

|  |  |  |
| --- | --- | --- |
| **Dimension: Store** | | |
| **DW Target Table** | **DW Target Column** | **Target Datatype** |
| 601-Group11-DW-Area.STORE\_DIM | STORE\_ID | int |
| STORE\_NUMBER | int |
| PRICE\_TIER | varchar |
| ZONE | int |

|  |  |  |
| --- | --- | --- |
| **Dimension: Time** | | |
| **DW Target Table** | **DW Target Column** | **Target Datatype** |
| 601-Group11-DW-Area.TIME\_DIM | TIME\_ID | int |
| WEEK | int |
| MONTH | int |
| YEAR | int |
| EVENT | varchar |

**Fact Tables**

|  |  |  |
| --- | --- | --- |
| **Fact: Store Visits** | | |
| **DW Target Table** | **DW Target Column** | **Target Datatype** |
| 601-Group11-DW-Area.STORE\_VISITS\_FACT | DEMO\_ID | int |
| COUPON\_ID | int |
| STORE\_ID | int |
| TIME\_ID | int |
| CUSTOMER\_COUNT | int |

|  |  |  |
| --- | --- | --- |
| **Fact: Sales** | | |
| **DW Target Table** | **DW Target Column** | **Target Datatype** |
| 601-Group11-DW-Area.  SALES\_FACT | STORE\_ID | int |
| TIME\_ID | int |
| PRODUCT\_ID | int |
| MOVEMENT | int |
| QUANTITY | int |
| SALES\_AMOUNT | float |

## Determine Source data

Source data to the above designed schema are from CCOUNT.csv, DEMO.csv, MOVEMENT and UPC file that are available with the Dominick’s FF data.

## Mapping tables for staging and data mart loads.

Please find the mapping tables for staging and data mart loads below.

1. **Mapping of data marts and their sources.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dimension: Product** | | | | | | |
| **DW Target Table** | **DW Target Column** | **Target Data type** | **Source System/Table** | **Source Column** | **Transformation/Business Rule** | **Error Handling Rules** |
| 601-Group11-DW-Area.PRODUCT\_DIM | PRODUCT\_ID | int |  | surrogate key |  |  |
| UPC\_CODE | varchar | UPCXXX.csv | upc |  |  |
| PRODUCT\_CATEGORY | varchar | UPCXXX.csv |  | XXX in source file name corresponds to the category code. e.g. If source csv file is named 'upcana', PRODUCT\_CATEGORY will be 'ana' |  |
| ITEM\_CODE | varchar | UPCXXX.csv | nitem |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Dimension: Demographic** | | | | | | |
| **DW Target Table** | **DW Target Column** | **Target Data type** | **Source System/Table** | **Source Column** | **Transformation/Business Rule** | **Error Handling Rules** |
| 601-Group11-DW-Area.DEMOGRAPHIC\_DIM | DEMO\_ID | int |  | surrogate key |  |  |
| INCOME\_LEVEL | varchar | DEMO.csv | income | Income in source column is log of median income (M.I.), which ranges from 9.87 to 11.24. e9.87=$19,341 and e11.24 = $76,114. We define 3 levels for INCOME\_LEVEL:  1. *Low*: <10.3 (M.I. less than $30,000 ) 2. *Medium*: 10.3 to 11.0(M.I. between $60,000) 3. *High*: >11.0 (M.I. greater than $60,000) | If income field is null in source table, update INCOME\_LEVEL with NA |
| POOR\_% | float | DEMO.csv | poverty |  |  |
| BELOW\_9\_% | float | DEMO.csv | age9 |  |  |
| ABOVE\_60\_% | float | DEMO.csv | age60 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Dimension: Coupon** | | | | | | |
| **DW Target Table** | **DW Target Column** | **Target Data type** | **Source System/Table** | **Source Column** | **Transformation/Business Rule** | **Error Handling Rules** |
| 601-Group11-DW-Area.COUPON\_DIM | COUPON\_ID | int |  | surrogate key |  |  |
| COUPON\_REDEEMED | int | ccount |  | Sum up coupons redeemed across all product categories. i.e. COUPON\_REDEEMED = BAKCOUP+BULKCOUP+COSMCOUP+FISHCOUP+… | for any missing coupons redeemed value across a product category, assume the value to be 0 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Dimension: Store** | | | | | | |
| **DW Target Table** | **DW Target Column** | **Target Data type** | **Source System/Table** | **Source Column** | **Transformation/Business Rule** | **Error Handling Rules** |
| 601-Group11-DW-Area.STORE\_DIM | STORE\_ID | int |  | Surrogate key |  |  |
| STORE\_NUMBER | int | Dominick’s Stores | Store |  |  |
| PRICE\_TIER | varchar | Dominick’s Stores | Price Tier |  |  |
| ZONE | int | Dominick’s Stores | Zone |  | If a row doesn’t have a value for the price tier, set the target as ‘0’ |
| CITY | varchar | Dominick’s Stores | City |  |  |
| ZIPCODE | int | Dominick’s Storess | Zip Code |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Dimension: Time** | | | | | | |
| **DW Target Table** | **DW Target Column** | **Target Data type** | **Source System/Table** | **Source Column** | **Transformation/Business Rule** | **Error Handling Rules** |
| 601-Group11-DW-Area.TIME\_DIM | TIME\_ID | int | Surrogate key |  |  |  |
| WEEK | int | Week’s Decode Table | Week # | Populated from 1 to 400 |  |
| MONTH | int | Week’s Decode Table | Start | The start column is of format MM/DD/YY. Split it to get MM |  |
| YEAR | int | Week’s Decode Table | Start | The start column is of format MM/DD/YY. Split it to get YY |  |
| EVENT | varrchar |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Fact: Store Visits** | | | | | | |
| **DW Target Table** | **DW Target Column** | **Target Data type** | **Source System/Table** | **Source Column** | **Transformation/Business Rule** | **Error Handling Rules** |
| 601-Group11-DW-Area.STORE\_VISITS\_FACT | DEMO\_ID | int | DEMOGRAPHIC\_DIM | DEMO\_ID | Foreign key corresponding to primary key DEMO\_ID of DEMOGRAPHIC\_DIM dimension |  |
| COUPON\_ID | int | COUPON\_DIM | DEMO\_ID | Foreign key corresponding to primary key COUPON\_ID of COUPON\_DIM dimension |  |
| STORE\_ID | int | STORE\_DIM | STORE\_ID | Foreign key corresponding to primary key STORE\_ID of STORE\_DIM dimension |  |
| TIME\_ID | int | TIME\_DIM | TIME\_ID | Foreign key corresponding to primary key TIME\_ID of TIME\_DIM dimension |  |
| CUSTOMER\_COUNT | int | ccount | CUSTCOUN |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Fact: Sales** | | | | | | |
| **DW Target Table** | **DW Target Column** | **Target Data type** | **Source System/Table** | **Source Column** | **Transformation/Business Rule** | **Error Handling Rules** |
| 601-Group11-DW-Area.SALES\_FACT | STORE\_ID | int | STORE\_DIM | STORE\_ID | Foreign key corresponding to primary key STORE\_ID of STORE\_DIM dimension |  |
| TIME\_ID | int | TIME\_DIM | TIME\_ID | Foreign key corresponding to primary key TIME\_ID of TIME\_DIM dimension |  |
| PRODUCT\_ID | int | PRODUCT\_DIM | PRODUCT\_ID | Foreign key corresponding to primary key PRODUCT\_ID of PRODUCT\_DIM dimension |  |
| MOVEMENT | int | UPCXXX.csv | move |  |  |
| QUANTITY | int | UPCXXX.csv | qty |  |  |
| SALES\_AMOUNT | float | UPCXXX.csv |  | SALES\_AMOUNT = Price \* Move / Qty for a given row |  |

1. **Mapping of data from .csv data sources to staging area**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staging Table: dbo.STORE** | | | | | |
| **File used to load** | **File Column** | **Data type** | **Staging Table** | **Staging Column** | **Staging Column Data type** |
| Dominick’s Stores | Store | varchar | 601-Group11-Staging-Area.STORE | STORE\_NUMBER | int |
| Price Tier | varchar | PRICE\_TIER | int |
| Zone | varchar | ZONE | varchar |
| City | varchar | CITY | varchar |
| Zip Code | varchar | ZIP\_CODE | varchar |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staging Table: dbo.TIME** | | | | | |
| **File used to load** | **File Column** | **Data type** | **Staging Table** | **Staging Column** | **Staging Column Data type** |
| Weeks Decode Table | Week | varchar | 601-Group11-Staging-Area.TIME | WEEK | int |
| -derived column- | varchar | MONTH | int |
| -derived column- | varchar | YEAR | int |
|  | Event | varchar |  | EVENT | varchar |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staging Table: dbo.PRODUCT** | | | | | |
| **File used to load** | **File Column** | **Data type** | **Staging Table** | **Staging Column** | **Staging Column Data type** |
| UPCXXX.csv | UPC Code | varchar | 601-Group11-Staging-Area.PRODUCT | UPC\_CODE | int |
| Item Code | varchar | ITEM\_CODE | varchar |
| Product Category | varchar | PRODUCT\_CATEGORY | varchar |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staging Table: dbo.COUPON** | | | | | |
| **File used to load** | **File Column** | **Data type** | **Staging Table** | **Staging Column** | **Staging Column Data type** |
| CCOUNT.csv | Coupons Redeemed | varchar | 601-Group11-Staging-Area.COUPON | COUPON\_REDEEMED | int |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staging Table: dbo.DEMOGRAPHIC** | | | | | |
| **File used to load** | **File Column** | **Data type** | **Staging Table** | **Staging Column** | **Staging Column Data type** |
| Demo.csv | Age9 | varchar | 601-Group11-Staging-Area.DEMOGRAPHIC | BELOW\_9\_% | float |
| Age60 | varchar | ABOVE\_60\_% | float |
| income | varchar | INCOME\_LEVEL | varchar |
|  | poverty | varchar |  | POOR\_% | float |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staging Table: dbo.Movement** | | | | | |
| **File used to load** | **File Column** | **Data type** | **Staging Table** | **Staging Column** | **Staging Column Data type** |
| Movements Table | No of Units Sold | varchar | 601-Group11-Staging-Area.MOVEMENT | MOVEMENT | int |
| Price | varchar | UNIT\_PRICE | float |
| Quantity | varchar | QUANTITY | int |
|  | -derived column- | varchar |  | SALES\_IN\_DOLLARS | float |
|  | Week | varchar |  | WEEK | int |
|  | Store Number | varchar |  | STORE\_NUMBER | int |
|  | UPC | varchar |  | UPC\_CODE | int |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staging Table: dbo.CCOUNT** | | | | | |
| **File used to load** | **File Column** | **Data type** | **Staging Table** | **Staging Column** | **Staging Column Data type** |
| CCount table | Date | varchar | 601-Group11-Staging-Area.CCOUNT | DATE | varchar |
| Week | varchar | WEEK | int |
| Cust Count | varchar | CUST\_COUNT | int |
|  | Store | varchar |  | STORE\_NUMBER | int |

1. **Mapping of data from staging area to data marts.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DW Table: dbo.STORE\_DIM** | | | | | |
| **Staging Table** | **Staging Column** | **Staging Column Data type** | **Production Table** | **Production Column** | **Production Column Data type** |
| 601-Group11-Staging-Area.STORE\_DIM | STORE\_NUMBER | int | 601-Group11-DW-Area.STORE\_DIM | STORE\_NUMBER | int |
| PRICE\_TIER | int | PRICE\_TIER | int |
| ZONE | varchar | ZONE | varchar |
| CITY | varchar | CITY | varchar |
| ZIP\_CODE | varchar | ZIP\_CODE | varchar |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DW Table: dbo.PRODUCT\_DIM** | | | | | |
| **Staging Table** | **Staging Column** | **Staging Column Data type** | **Production Table** | **Production Column** | **Production Column Data type** |
| 601-Group11-Staging-Area.PRODUCT\_DIM | UPC\_CODE | int | 601-Group11-DW-Area.PRODUCT\_DIM | UPC\_CODE | int |
| ITEM\_CODE | varchar | ITEM\_CODE | varchar |
| PRODUCT\_CATEGORY | varchar | PRODUCT\_CATEGORY | varchar |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DW Table: dbo.TIME\_DIM** | | | | | |
| **Staging Table** | **Staging Column** | **Staging Column Data type** | **Production Table** | **Production Column** | **Production Column Data type** |
| 601-Group11-Staging-Area.TIME\_DIM | WEEK | int | 601-Group11-DW-Area.TIME\_DIM | WEEK | int |
| MONTH | int | MONTH | int |
| YEAR | int | YEAR | int |
| EVENT | varchar | EVENT | varchar |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DW Table: dbo.COUPON\_DIM** | | | | | |
| **Staging Table** | **Staging Column** | **Staging Column Data type** | **Production Table** | **Production Column** | **Production Column Data type** |
| 601-Group11-Staging-Area.COUPON\_DIM | COUPON\_REDEEMED | int | 601-Group11-DW-Area.COUPON\_DIM | COUPON\_REDEEMED | int |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DW Table: dbo.DEMOGRAPHIC\_DIM** | | | | | |
| **Staging Table** | **Staging Column** | **Staging Column Data type** | **Production Table** | **Production Column** | **Production Column Data type** |
| 601-Group11-Staging-Area.DEMOGRAPHIC\_DIM | BELOW\_9\_% | float | 601-Group11-DW-Area.DEMOGRAPHIC\_DIM | BELOW\_9\_% | float |
| ABOVE\_60\_% | float | ABOVE\_60\_% | float |
| INCOME\_LEVEL | varchar | INCOME\_LEVEL | varchar |
| POOR\_% | float | POOR\_% | float |

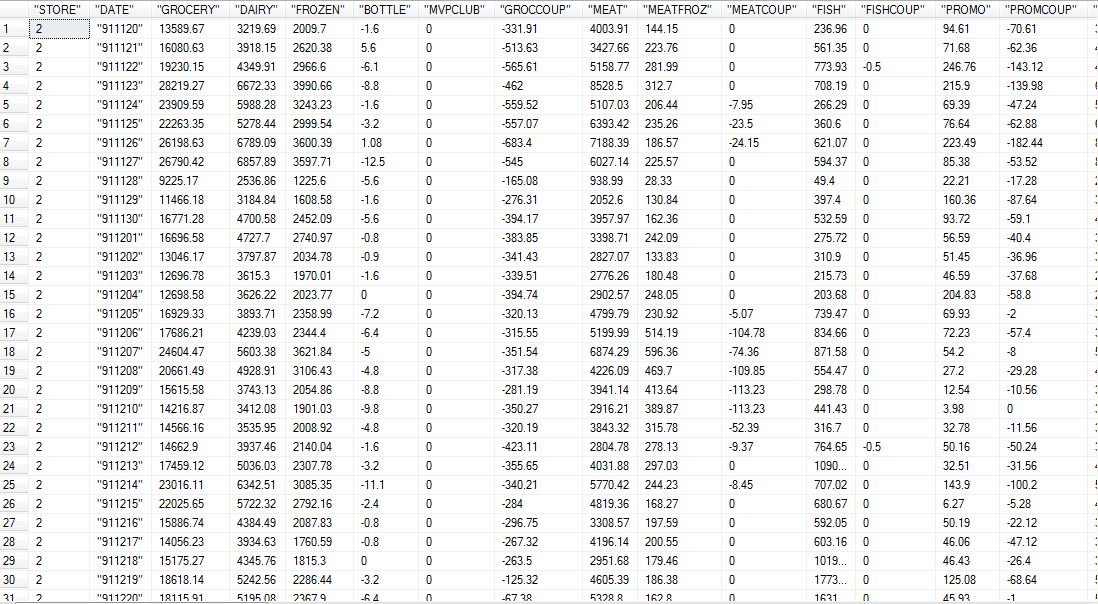
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DW Table: dbo.STORE\_VISITS\_FACT** | | | | | |
| **Staging Table** | **Staging Column** | **Staging Column Data type** | **Production Table** | **Production Column** | **Production Column Data type** |
| 601-Group11-Staging-Area.CCOUNT | CUST\_COUNT | int | 601-Group11-DW-Area.STORE\_VISITS\_FACT | CUSTOMER\_COUNT | int |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DW Table: dbo.SALES\_FACT** | | | | | |
| **Staging Table** | **Staging Column** | **Staging Column Data type** | **Production Table** | **Production Column** | **Production Column Data type** |
| 601-Group11-Staging-Area. MOVEMENT | MOVEMENT | int | 601-Group11-DW-Area.SALES\_FACT | MOVEMENT | int |
| QUANTITY | int | QUANTITY | int |
| SALES\_IN\_DOLLARS | float | SALES\_AMOUNT | float |

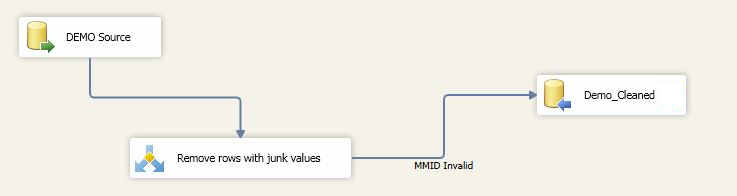
## Comprehensive Data Extraction Rules

Data has been extracted from the provided CSV files and data manual of DFF. The extraction rules used in the report are as follows.

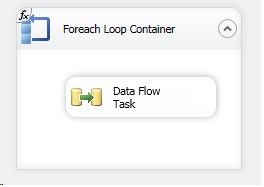
* The ccount.csv has been used as a source to extract data for COUPON dimension. The coupon redeemed columns across various product categories will be used to COUPON\_DIM. So, in order to handle missing value in coupon redeemed columns, we replaced all the missing coupon redeemed values with 0.
* We truncated data with invalid store numbers from the CCOUNT table.
* Additionally, week number will be used to analyze chronological trends, so we removed the junk rows with missing week numbers during extraction process.
  + After extraction with above rules from ccount.csv, CCOUNT table looks as follows.



* The datatype of store IDs, Week and product category coupon redeemed values have been changed from varchar to int during extraction.
* The DEMO table is being extracted from demo.csv. The rows with non-numeric Store IDs have been cleaned during extraction process.
* While extracting data for demo table, junk data with missing MMIDs, which uniquely identify each demo row, have been cleaned.



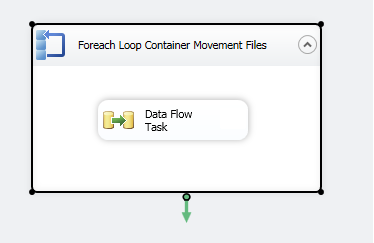
* The columns below\_9\_%, above\_60\_% and poor\_% have been converted from varchar to numeric.
* The data in UPC table is being extracted using Foreach Container for all UPCXXX.csv files and then combined into final table.



* In the cleaned UPC table we are mapping each row with product category. E.g. for data extracted from the file UPCANA.csv, product category is “ANA”. We are using following extraction expression to extract product category from file name.

*SUBSTRING(@[User::fileName],LEN(@[User::fileName]) - FINDSTRING(REVERSE(@[User::fileName]),"\\",1) + 2,LEN(RIGHT(@[User::fileName],FINDSTRING(REVERSE(@[User::fileName]),"\\",1) - 1)) - FINDSTRING(REVERSE(@[User::fileName]),".",1))*

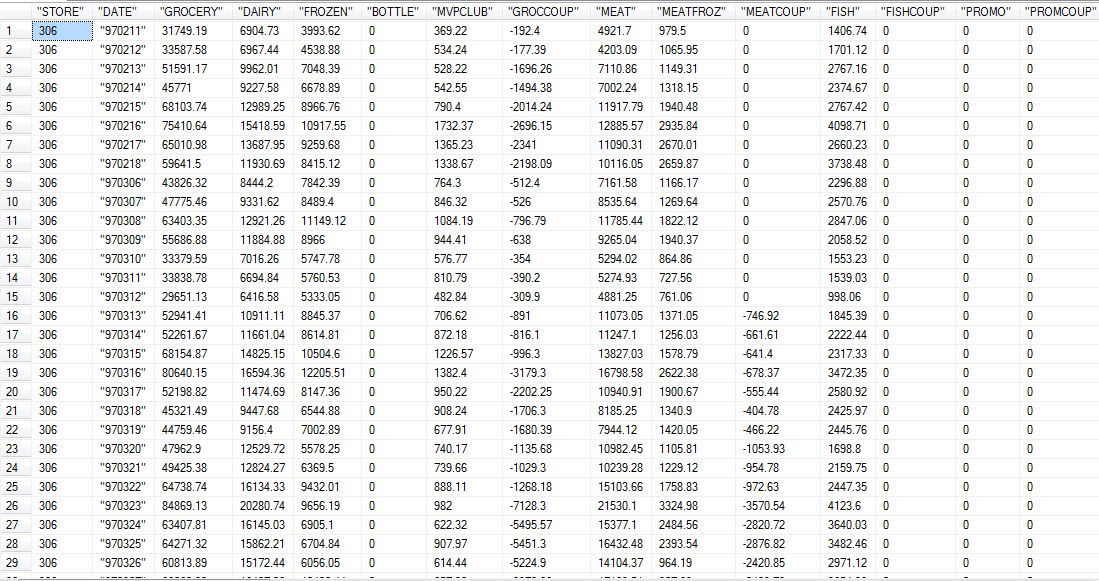
* The data in Movement table is being extracted using Foreach Container for all movement csv files and then combined into final Movement table.



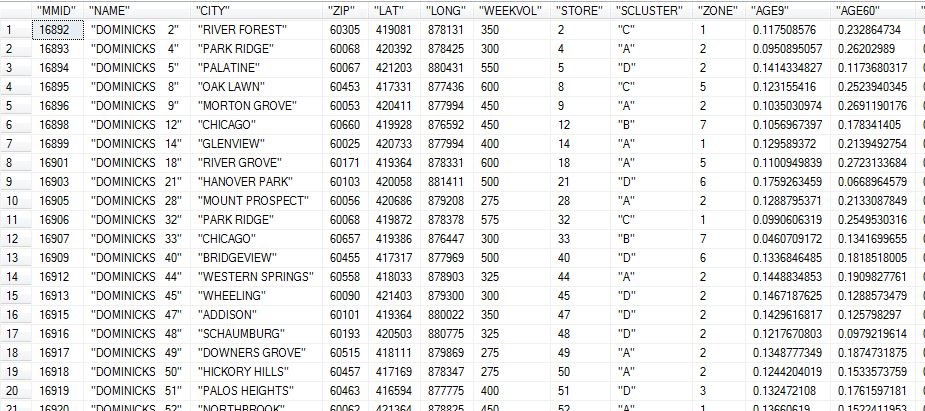
## Screenshots of Data Staging area

The extracted data from the data sources have been stored in the 601-Group11-Staging-Area database. The data from staging area will further be transformed for data marts. The screenshots of different tables in staging area are as follows:

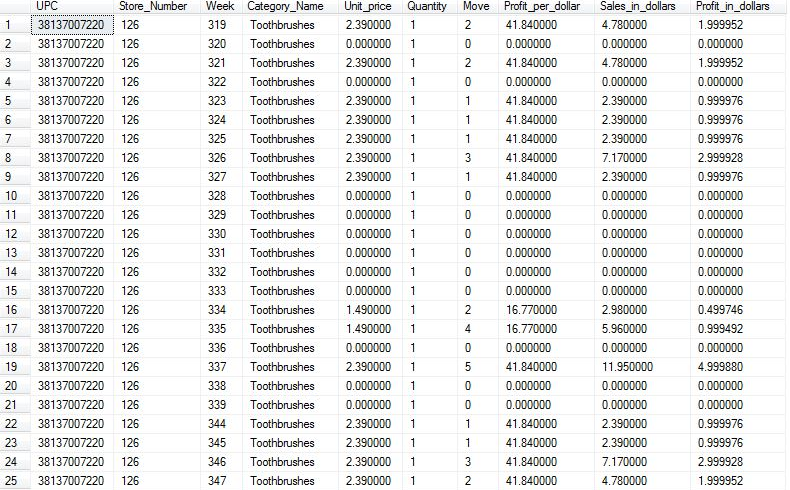
**Table: 601-Group11-Staging-Area.CCOUNT**



**Table: 601-Group11-Staging-Area.DEMO**



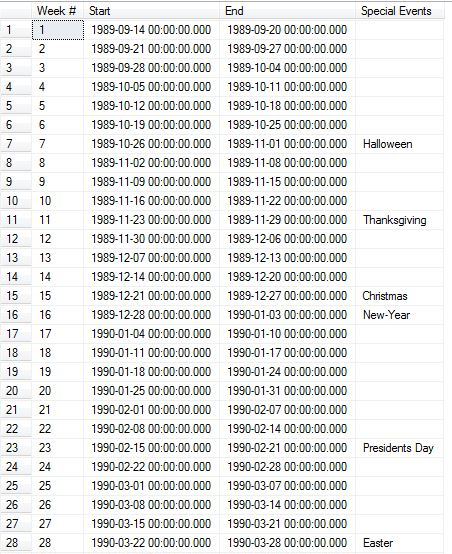
**Table: 601-Group11-Staging-Area.UPC**



**Table: 601-Group11-Staging-Area.STORES**



**Table: 601-Group11-Staging-Area.TIME**

****

## Data Transformation and Cleansing Rules

1. **Transformation of INCOME in CCOUNT to INCOME\_LEVEL**

Income column on CCOUNT tables stores log of median income (M.I.), which ranges from 9.87 to 11.24. e9.87=$19,341 and e11.24 = $76,114. We define 3 levels for INCOME\_LEVEL:   
 *Low*: <10.3 (M.I. less than $30,000 )  
 *Medium*: 10.3 to 11.0(M.I. between $60,000)  
 *High*: >11.0 (M.I. greater than $60,000)

We have used following expression in SSIS to generate the income levels from income values in the DEMO table.

*INCOME\_LEVEL = (INCOME > 11) ? "High" : (INCOME < 10.3 ? "Low" : "Medium")*

1. **Calculation of Sales Amount value from Price, Movement and Quality columns**

The sales amount for each row in Movement table in staging area has been calculated using following transformation.

*Sales\_in\_dollars = (Unit\_Price\*Move/Quantity)*

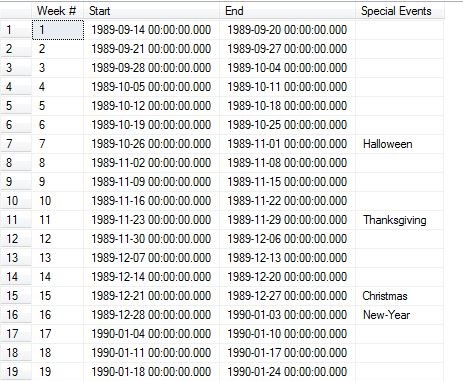
1. **Deriving Product\_Category from UPCXXX.csv filepath**

We derived Product\_Category from path of each UPCXXX.cvs file. For example, if full path a upc file is C:/folder/UPCANA.csv, we extracted the filename i.e. UPCANA.csv and then we further extracted product category as “ANA”. The expression used in derived column SSIS component is as follows:

*SUBSTRING(@[User::fileName],LEN(@[User::fileName]) - FINDSTRING(REVERSE(@[User::fileName]),"\\",1) + 2,LEN(RIGHT(@[User::fileName],FINDSTRING(REVERSE(@[User::fileName]),"\\",1) - 1)) - FINDSTRING(REVERSE(@[User::fileName]),".",1))*

1. **Transformation of Time data into Week, Month and Year details**

We used Week’s Decode Table for this transformation.



For this transformation, we converted start date for each week from varchar to datetime datatype and then derived Month and Year value from the data using following SSIS expressions.

*YEAR = YEAR([Datatype convert for Week].Start)*

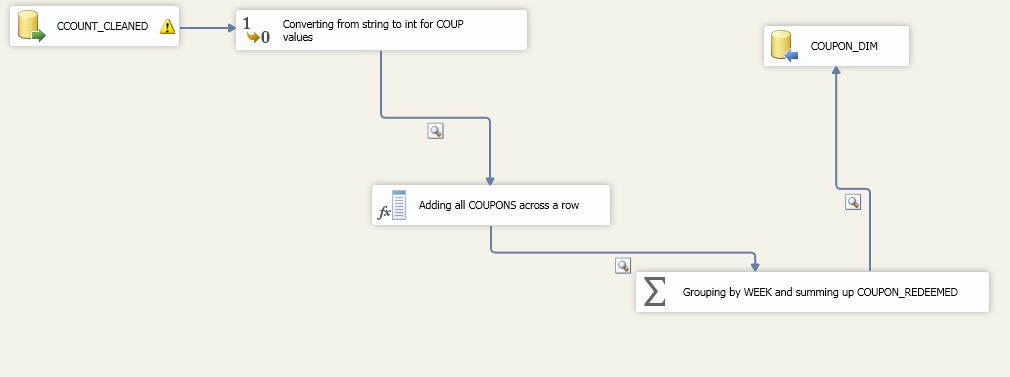
*MONTH = MONTH([Datatype convert for Week].Start)*

## Plan for Aggregation

Aggregation is done in Coupon dimension table, Sales fact table and Store Visits fact table for answering the business questions and to fix the granularity. Please find the explanation for each aggregation below.

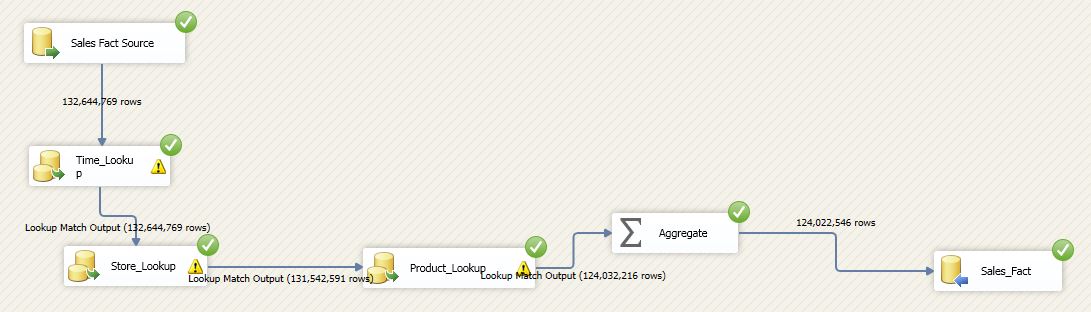
1. **Aggregation in Coupon dimension table**.

Coupon redeemed data is loaded from CCount table and the data is aggregated on weekly basis to keep the time granularity of the Coupon redeemed sales at Weekly level. As shown in figure below Coupon redeemed attribute in Coupon dimension table is calculated by summing up the Coupon redeemed value on a weekly basis. Hence the surrogate keys generated for Coupon dimension in in one-to-one mapping with week number.



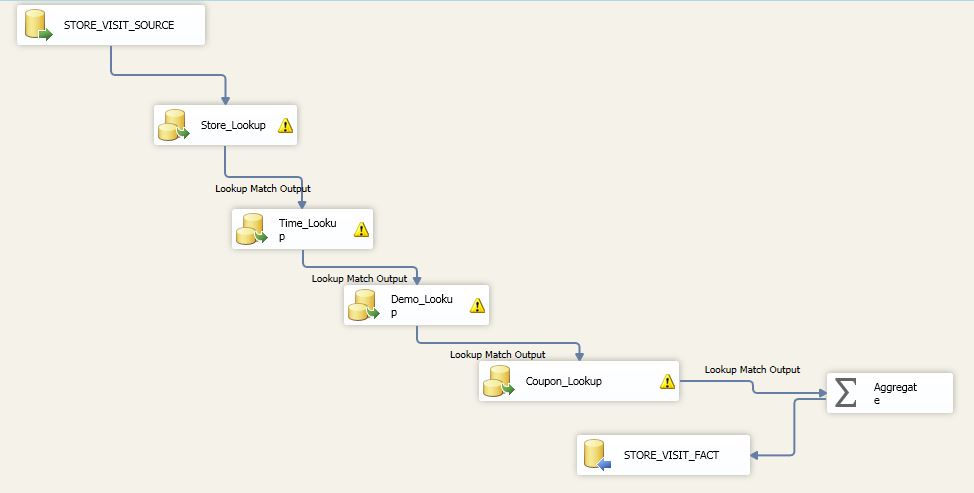
1. **Aggregation in SALES\_FACT table.**

Source data for Sales fact table is Movement table .lookup transformation is performed with cleaned Movement table from data staging area on Time, Store and Product dimension tables on the matching of attributes Week Number, Store Number and UPC Code with respective tables. Aggregation of the MOVEMENT, QUANTITY and SALES\_AMOUNT in the resultant table is aggregated on the basis of Store\_ID, Time\_ID and Product\_ID. Store\_ID, Time\_ID and Product\_ID are mapped to Store Number, Week Number and UPC\_Code respectively. So the aggregation ensures the granularity in the above levels for each dimensions. We are concerned mainly about the Time granularity for drilling down and rolling up in later phases of the project and the time granularity will be on Week level. This can be rolled up to Month and Year.



1. **Aggregation in STORE\_VISITS\_FACT table.**

Source data for the STORE\_VISITS\_FACT table is the CCount data. Lookup transformation was performed with cleaned CCount table from the staging area on STORE\_DIM, TIME\_DIM, DEMOGRAPHIC\_DIM and COUPON\_DIM tables in the same order as shown in the below figure. Attributes matched for lookup in STORE\_DIM, TIME\_DIM, DEMOGRAPHIC\_DIM and COUPON\_DIM tables are STORE\_ID, TIME\_ID,DEMO\_ID and COUPON\_ID respectively. Time granularity for the CUSTOMER\_COUNT in the STORE\_VISITS\_FACT is WEEK .This can be rolled up to Month and Year level.



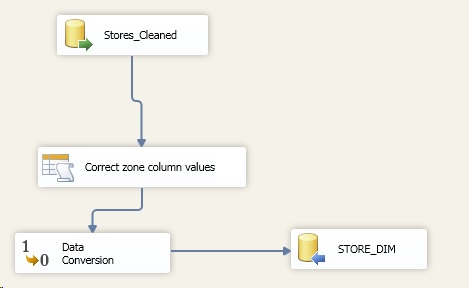
## Step by step procedure for data extraction and loading

Main source of data for this project is main from 4 .csv files- CCount, Demographics, UPC and Movement. This data was directly loaded into data staging area of the warehouse directly and data cleaning operations were performed as explained in this report. This cleaned data was further transformed using different ETL processes as explained below for the 5 final dimension(STORE\_DIM, PRODUCT\_DIM,TIME\_DIM, COUPON\_DIM, DEMOGRAPHICS\_DIM) tables and 2(SALES\_FACT, STORE\_VISITS\_FACT) fact tables.

1. **ETL for DIMENSION tables**

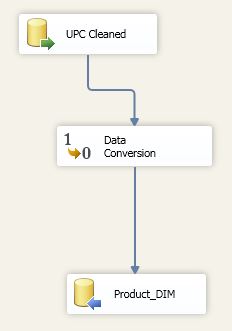
**STORE\_DIM**

Source data for the STORE\_DIM is Dominick’s Stores data table. This data was cleaned and ETL was performed on it as shown in the below diagram. Few Column’s had no Zone value in Dominick Stores table and then data conversion was performed on it from String to Integer. This transformed data was stored in STORE\_DIM table.



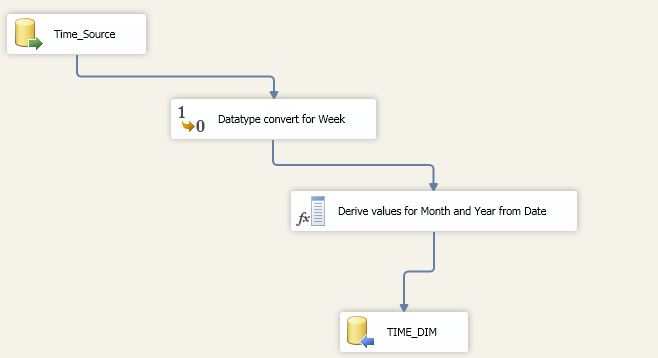
**PRODUCT\_DIM**

Product Dimension tables took source data from UPC tables in Dominick’s data. Each product has its own UPC tables and all tables were merged together. This table was cleaned as explained previously. On the cleaned UPC data, data conversion was performed for ITEM\_Code to convert from Varchar to Integer as shown below.



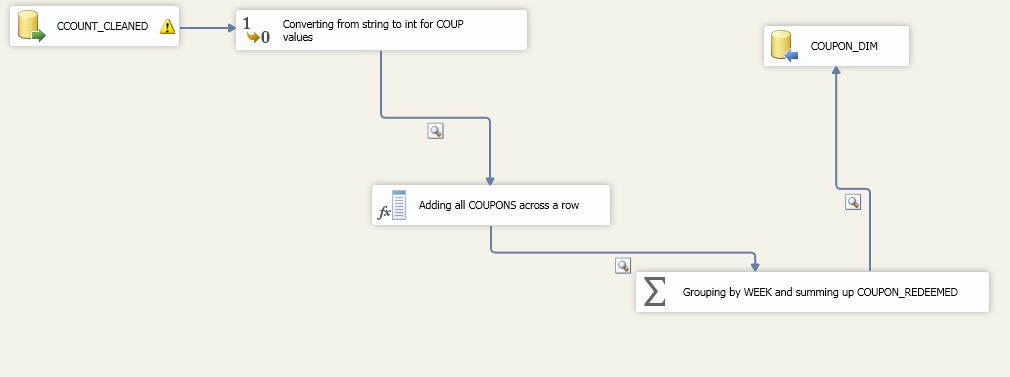
**TIME\_DIM**

Weeks Decode table from Dominick’s manual acts as the source table for the TIME\_DIM table. Initially the data type for Week is Varchar and it is converted to Integer. Thereafter the Month and Year attributes in TIME\_DIM was derived from the date as per the Transformation described previously. This final transformed table is stored in TIME\_DIM.

****

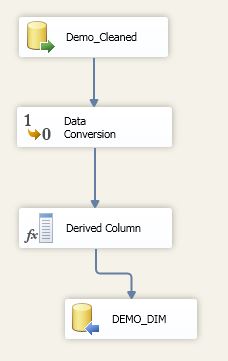
**COUPON\_DIM**

Coupon redeemed values for the COUPON\_DIM was sourced from CCount table. ETL is performed on the cleaned CCount table in data staging area. First of all data conversion from varchar to Integer was performed on Coupon redeemed values. Thereafter is performed on the Coupon redeemed values on weekly basis to get the weekly coupon redeemed values every week. This transformed table is kept COUPON\_DIM table.

****

**DEMOGRAPHICS\_DIM**

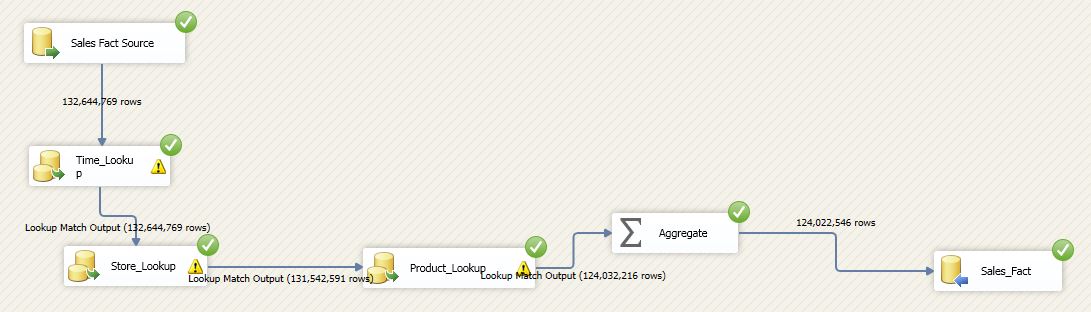
Demographics table from Dominick’s is the source table for DEMOGRAPHICS\_DIM. The cleaned data is kept in data staging area and ETL transformation performed on the same. Firstly the values BELOW\_9\_% and ABOVE\_60\_% undergoes data type conversion from varchar to float. There after derived column INCOME\_LEVEL is set as per the range of log of median income. Business rule for the same is explained in the mapping table. This will be the second stage of ETL transformation and the resultant table after this step is stored in DEMO\_DIM.

****

1. **ETL for FACT tables**

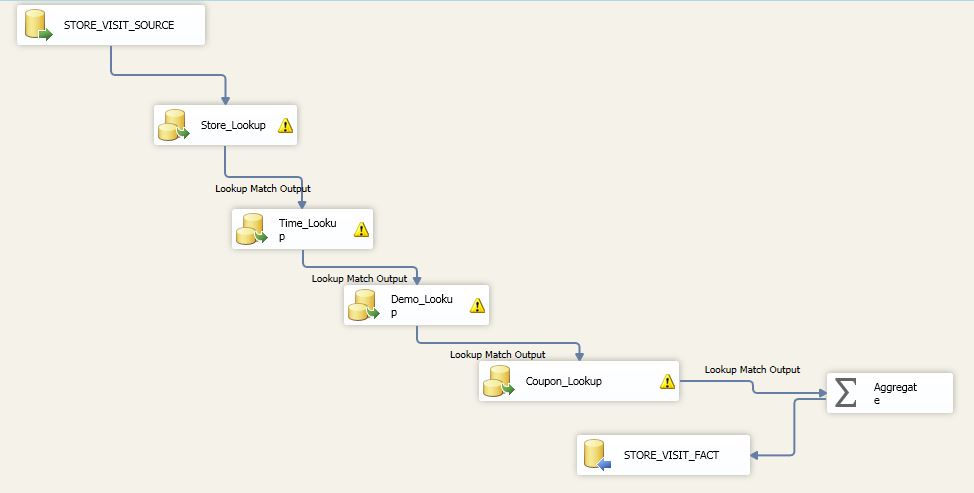
**SALES\_FACT**

Movement table is the source data for Sales fact table. SALES\_AMOUNT in Movement table is the value derived from Price, Quantity and Movement values in Movement table. Cleaning was performed on this table and kept in Staging area. Lookup transformation was performed on this table this table as source on TIME\_DIM, STORE\_DIM, PRODUCT\_DIM and aggregation was performed as explained previously before loading the data in SALES\_FACT table.



**STORE\_VISITS\_FACT**

Source data for STORE\_VISITS\_FACT table is CCount table. Cleaned CCount data in staging area undergoes Lookup transformation with STORE\_DIM, TIME\_DIM, DEMO\_DIM and COUPON \_DIM. Finally aggregation is performed as explained previously and final data is loaded in STORE\_VISITS\_FACT table.



## Mapping definition describing the source to end table for all dimension and fact tables

End to end mapping of elements from source to destination data marts

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dimension: Product** | | | | | | | | |
| **Source File** | **Column Name** | **Datatype** | **Staging Table** | **Staging Column** | **Staging Column Datatype** | **Production Table** | **Production Column** | **Production Column Datatype** |
| UPC.csv | UPC\_CODE | varchar | UPC-Cleaned | UPC | varchar | PRODUCT\_DIM | UPC\_CODE | int |
| PRODUCT\_CATEGORY | varchar | PROD\_CATEGORY | varchar | PRODUCT\_CATEOGRY | varchar |
| ITEM\_CODE | varchar | NITEM | varchar | ITEM\_CODE | int |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dimension: Store** | | | | | | | | |
| **Source File** | **Column Name** | **Datatype** | **Staging Table** | **Staging Column** | **Staging Column Datatype** | **Production Table** | **Production Column** | **Production Column Datatype** |
| Demo.csv | STORE | varchar | Stores-Cleaned | STORE | int | STORE\_DIM | STORE\_NUMBER | int |
| PRICE TIER | varchar | PRICE TIER | varchar | PRICE\_TIER | varchar |
| ZONE | varchar | ZONE | varchar | ZONE | int |
| CITY | varchar | CITY | varchar | CITY | varchar |
| ZIPCODE | varchar | ZIPCODE | varchar | ZIPCODE | varchar |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dimension: Time** | | | | | | | | |
| **Source File** | **Column Name** | **Datatype** | **Staging Table** | **Staging Column** | **Staging Column Datatype** | **Production Table** | **Production Column** | **Production Column Datatype** |
| Dominick's Stores | Week # | varchar | Time-Cleaned | Week # | int | TIME\_DIM | WEEK | int |
| Start | varchar | Start | Datetime | MONTH | Calculated Field |
| End | varchar | End | Datetime | YEAR | Calculated Field |
| Special Events | varchar | Special Events | varchar | SPECIAL EVENTS | varchar |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dimension: Demographic** | | | | | | | | |
| **Source File** | **Column Name** | **Datatype** | **Staging Table** | **Staging Column** | **Staging Column Datatype** | **Production Table** | **Production Column** | **Production Column Datatype** |
| Demo.csv | Age9 | varchar | DEMO-Cleaned | BELOW\_9\_% | varchar | DEMOGRAPHIC\_DIM | BELOW\_9\_% | decimal |
| age60 | varchar | ABOVE\_60\_% | varchar | ABOVE\_60\_% | decimal |
| poverty | varchar | POOR\_% | varchar | POOR\_% | decimal |
| income | varchar | INCOME\_LEVEL | decimal | INCOME\_LEVEL | Calculated Value |
| Store | varchar | STORE\_NUM | int | STORE\_NUM | int |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dimension: Coupon** | | | | | | | | |
| **Source File** | **Column Name** | **Datatype** | **Staging Table** | **Staging Column** | **Staging Column Datatype** | **Production Table** | **Production Column** | **Production Column Datatype** |
| CCOUNT.csv | Coupon Redeemed | Coupon value across the different products- Calculated field | CCOUNT-Cleaned | COUPON\_REDEEMED | decimal | COUPON\_DIM | COUPON\_REDEEMED | varchar |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fact Table: Sales** | | | | | | | | |
| **Source File** | **Column Name** | **Datatype** | **Staging Table** | **Staging Column** | **Staging Column Datatype** | **Production Table** | **Production Column** | **Production Column Datatype** |
| Keys are auto incremented values | | | PRODUCT\_DIM | PRODUCT\_ID | int | SALES\_FACT | PRODUCT\_ID | int |
| STORE\_DIM | STORE\_ID | int | STORE\_ID | int |
| TIME\_DIM | TIME\_ID | int | TIME\_ID | int |
| UPC.csv | move | varchar | UPC-Cleaned | movement | int | MOVEMENT | int |
| UPC.csv | qty | varchar | UPC-Cleaned | quantity | int | QUANTITY | decimal |
| UPC.csv | Sales | Calculated value | UPC-Cleaned | Sales amount | decimal | SALES\_AMOUNT | decimal |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fact Table: Store visits** | | | | | | | | |
| **Source File** | **Column Name** | **Datatype** | **Staging Table** | **Staging Column** | **Staging Column Datatype** | **Production Table** | **Production Column** | **Production Column Datatype** |
| Keys are auto incremented values | | | DEMOGRAPHIC\_DIM | DEMO\_ID | int | STORE\_VISITS\_FACT | DEMO\_ID | int |
| COUPON\_DIM | COUPON\_ID | int | COUPON\_ID | int |
| STORE\_DIM | STORE\_ID | int | STORE\_ID | int |
| TIME\_DIM | TIME\_ID | int | TIME\_ID | int |
| CCOUNT.csv | CUSTCOUN | varchar | CCOUNT-Cleaned | Customer Count | int | COUSTOMER\_COUNT | int |

## SQL statements used for the ETL operations

Below are the SQL scripts used to create the destination tables:

CREATE TABLE [dbo].[COUPON\_DIM](

[COUPON\_ID] [int] NOT NULL,

[COUPON\_REDEEMED] [real] NULL,

PRIMARY KEY [COUPON\_ID]

)

CREATE TABLE [dbo].[DEMOGRAPHIC\_DIM](

[DEMO\_ID] [int] IDENTITY(1,1) NOT NULL,

[BELOW\_9\_%] [numeric](10, 2) NULL,

[ABOVE\_60\_%] [numeric](10, 2) NULL,

[POOR\_%] [numeric](10, 2) NULL,

[INCOME\_LEVEL] [varchar](50) NULL,

[STORE\_NUM] [int] NOT NULL,

PRIMARY KEY [DEMO\_ID]

)

CREATE TABLE [dbo].[PRODUCT\_DIM](

[PRODUCT\_ID] [int] IDENTITY(1,1) NOT NULL,

[UPC\_CODE] [varchar](50) NULL,

[PRODUCT\_CATEGORY] [varchar](50) NULL,

[ITEM\_CODE] [varchar](50) NULL,

PRIMARY KEY [PRODUCT\_ID]

)

CREATE TABLE [dbo].[STORE\_DIM](

[STORE\_ID] [int] IDENTITY(1,1) NOT NULL,

[STORE\_NUMBER] [int] NULL,

[PRICE\_TIER] [varchar](50) NULL,

[ZONE] [int] NULL,

[CITY] [varchar](50) NULL,

[ZIPCODE] [varchar](50) NULL,

PRIMARY KEY [STORE\_ID]

)

CREATE TABLE [dbo].[TIME\_DIM](

[TIME\_ID] [int] IDENTITY(1,1) NOT NULL,

[WEEK] [int] NULL,

[MONTH] [int] NULL,

[YEAR] [int] NULL,

[EVENTS] [varchar](50) NULL,

PRIMARY KEY [TIME\_ID]

)

CREATE TABLE [STORE\_VISITS\_FACT] (

DEMO\_ID int FOREIGN KEY references DEMOGRAPHIC\_DIM(DEMO\_ID),

COUPON\_ID int FOREIGN KEY references COUPON\_DIM(COUPON\_ID),

STORE\_ID int FOREIGN KEY references STORE\_DIM(STORE\_ID),

TIME\_ID int FOREIGN KEY references TIME\_DIM(TIME\_ID),

COSTOMER\_COUNT int

PRIMARY KEY (DEMO\_ID,COUPON\_ID,STORE\_ID,TIME\_ID)

)

CREATE TABLE [SALES\_FACT] (

PRODUCT\_ID int FOREIGN KEY references PRODUCT\_DIM(PRODUCT\_ID),

STORE\_ID int FOREIGN KEY references STORE\_DIM(STORE\_ID),

TIME\_ID int FOREIGN KEY references TIME\_DIM(TIME\_ID),

MOVEMENT int,

QUANTITY int,

SALES\_AMOUNT numeric,

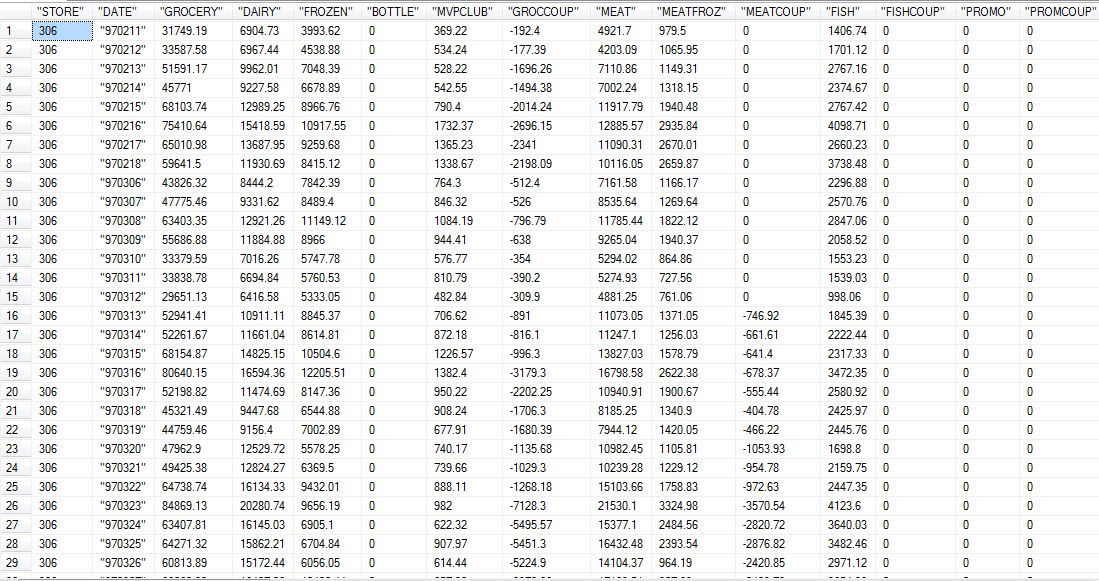
PRIMARY KEY (PRODUCT\_ID,STORE\_ID,TIME\_ID)

)

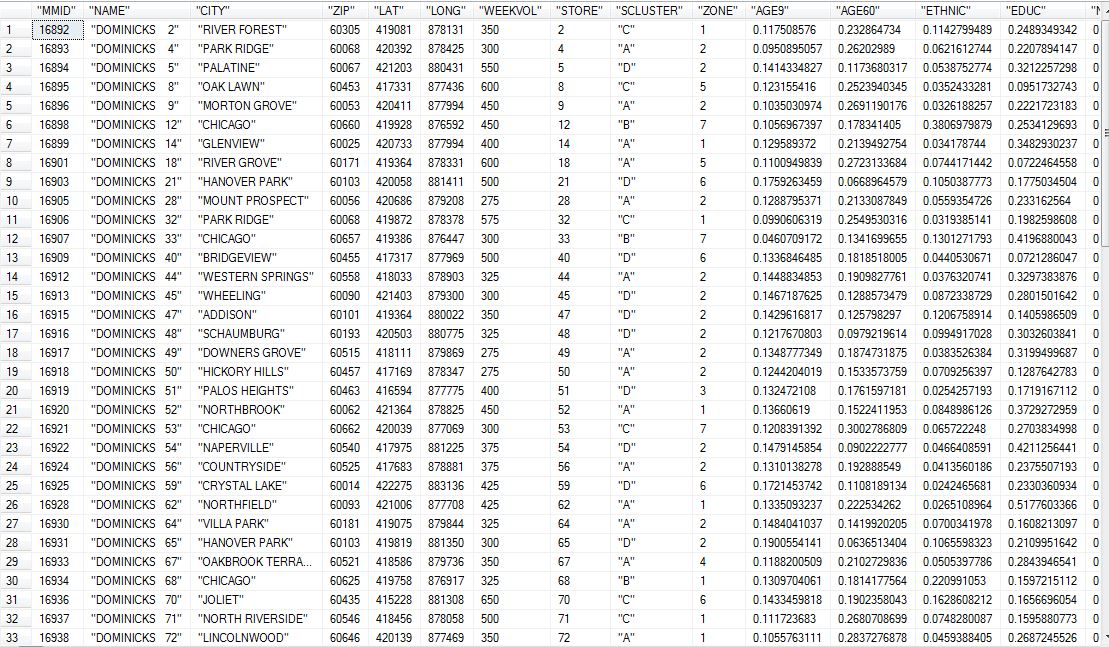
## Staging and Data Mart table screen shots

**Staging tables:**

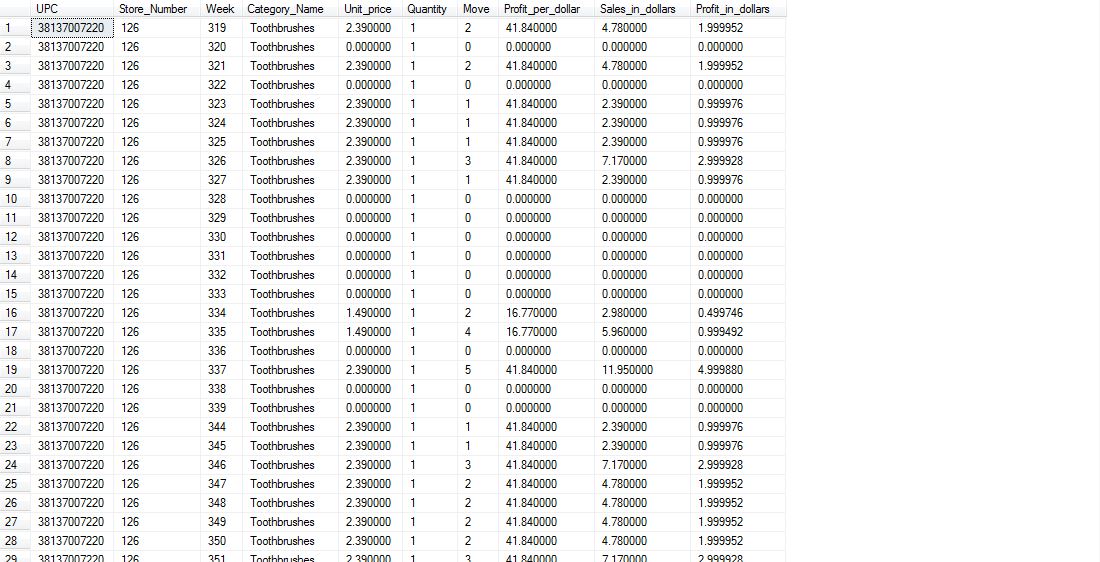
601-Group11-Staging-Area.CCOUNT-Cleaned



601-Group11-Staging-Area.DEMO-Cleaned

****

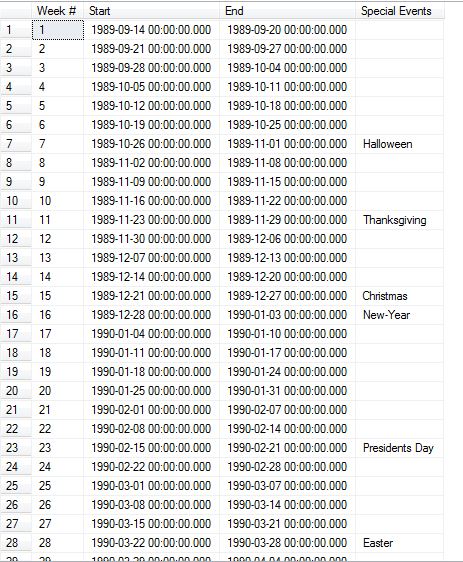
601-Group11-Staging-Area.MOVEMENT-Cleaned

****

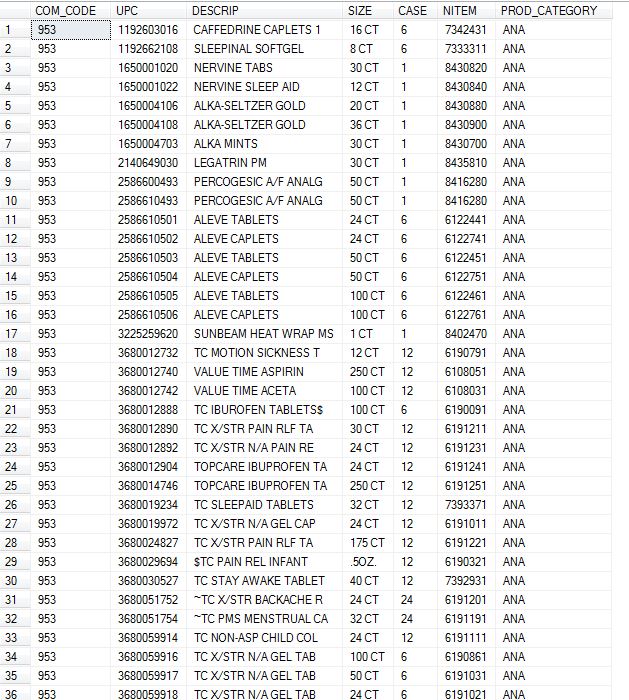
601-Group11-Staging-Area.STORES-Cleaned

****

601-Group11-Staging-Area.TIME-Cleaned

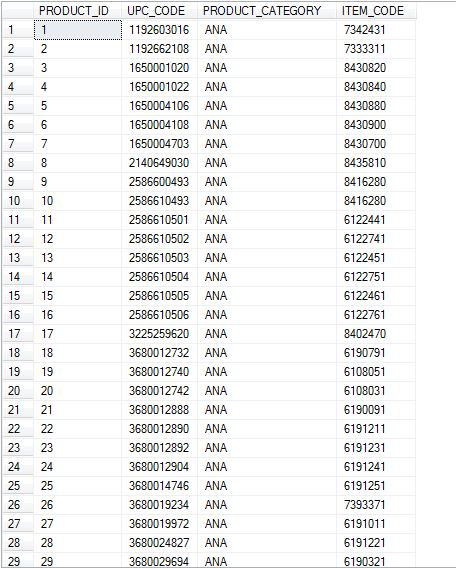


601-Group11-Staging-Area.UPC-Cleaned



**Data Mart tables:**

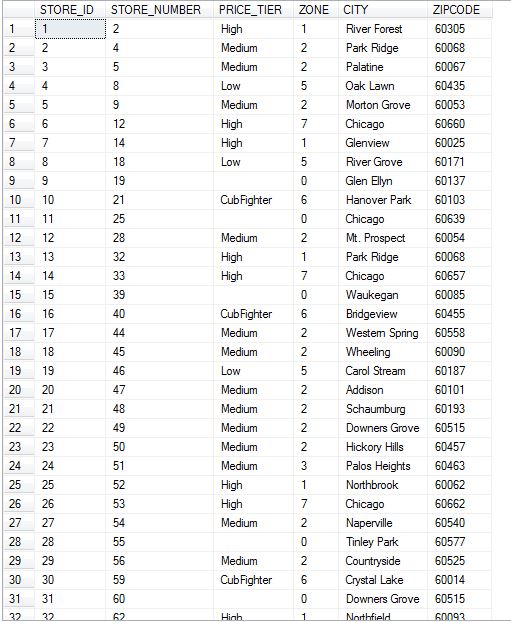
601-Group11-DW-Area.PRODUCT\_DIM



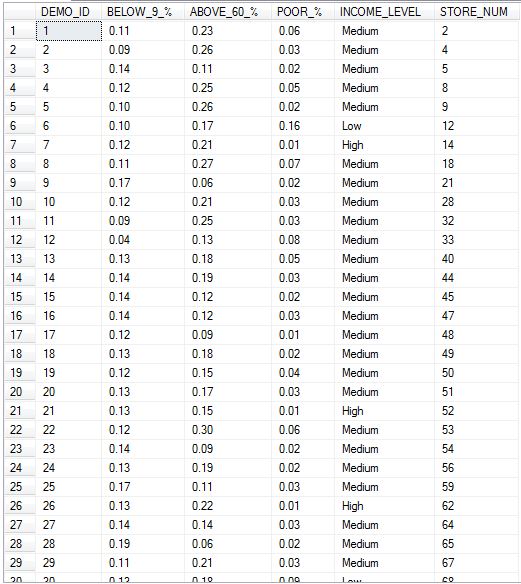
601-Group11-DW-Area.TIME\_DIM



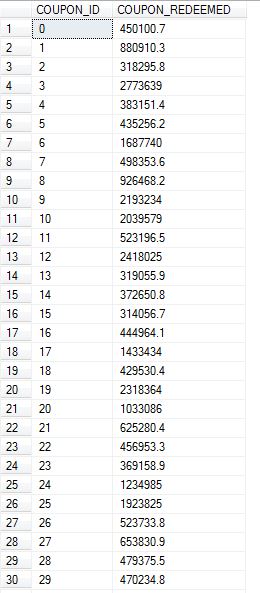
601-Group11-DW-Area.STORE\_DIM

****

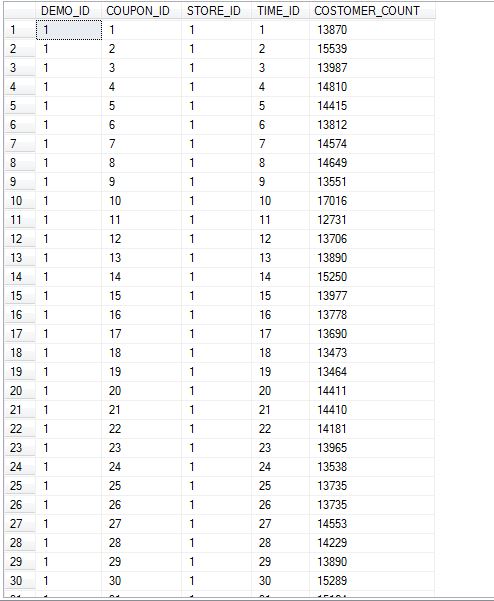
601-Group11-DW-Area.DEMOGRAPHIC\_DIM



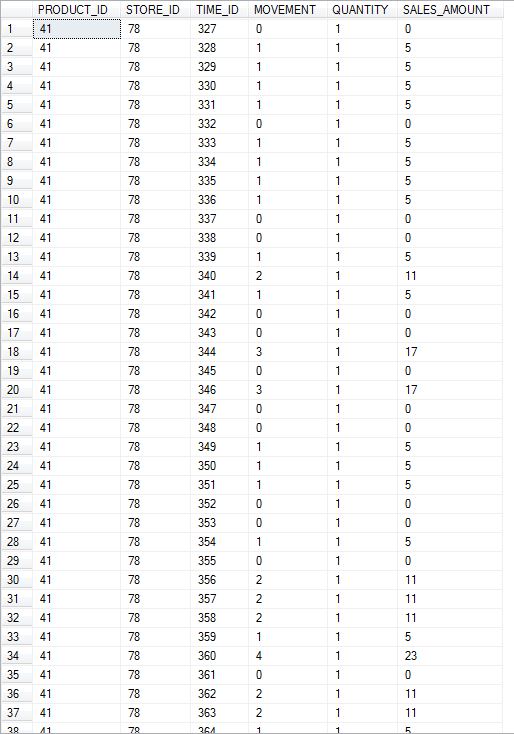
601-Group11-DW-Area.COUPON\_DIM



601-Group11-DW-Area.STORE\_VISITS\_FACT



601-Group11-DW-Area.SALES\_FACT

****