# Design and Implementation of a Data Warehouse for a Retail Store with Store-level Data

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

# Table of Contents

**Section 1: Introduction....3**
**1.1 Details about the Data....4**
1.1.1 General Files....4
Customer Count (CCOUNT.csv)....4
Store-Level Demographics (DEMO.csv)....4
Dominick Store Locations....4
Week's Decode Table....4
1.1.2 Category-Specific Files....5
Movement Data (WXXX.csv)....5
UPC Data (UPCXXX.csv)....5
1.2 Metadata for all Source files....5
1.3 Entity Relationship Diagram....9
**Section 2: Subject Area Understanding....11**
**2.1 Business Questions....12**
2.2 Selected Business Questions....24
**Section 3: Independent Data Marts Design using Kimball's Methodology....25**
3.1 Steps to follow under Kimball's Methodology....25
3.2 Why is it important to follow Kimball's Methodology?....25
3.3 Dimension Tables Design....26
3.4 Fact Tables Design....27
Dimension Matrix....27
3.5 Data Warehouse Schema....28
Star Schema....28
Sales Data Mart....28
Customer Data Mart....28
3.6 Schema Justification for Business Questions....29
3.7 Mapping Tables....30
3.8 Physical Design....33
Standardization....33
Aggregation....33
Partitioning....33
Clustering....33
Indexing....34
Data Structure for Storage....34
**Section 4: Data Cleaning and Integration....35**
4.1 Determine the target data....35
4.2 Determine the Data Sources....36
4.3 Mapping Tables in Excel....37

Source to Staging Area....37
Staging to DW Mapping....38
4.4 Data Extraction Rules....38
4.5 Data Transformation and Cleaning Rules....39
4.6 Plan for aggregate tables....39
4.7 Organization of data staging area....39
4.8 Procedure for data extraction and loading....40
4.9 ETL for Dimension Table....40
4.10 ETL for Fact Table....40
4.11 Data Quality Issues....41
4.12 ETL for Staging Area....41
ETL for CCOUNT to Staging table Cust_Count....42
ETL for Demo.csv to Staging Area....46
ETL for Weekly_Decode to Staging....48
ETL for WGRO_csv and WBER_csv (Movement Files) to Staging....50
4.13 List of Temporary Tables....54
4.14 Data Granularity at Independent Data Mart Level....54
4.15 ETL for Dimension Tables....54
DIM_PRODUCT....54
DIM_STORE_DEMO....56
DIM_TIME....58
4.16 ETL for Fact Tables....60
FACT_CUSTOMER....60
FACT_SALES....62
4.17 Queries used in the process....64
**Section 5: BI Reporting....66**
5.1 Reporting Plan....66
5.1.1 Determine all the target reports that satisfy the business questions and why....66
5.1.2 Determine all the mappings from the independent data marts to the report attributes....70
5.1.3 Describe All Report Templates....72
5.2 Reporting Implementation....73
5.2.1 Report using SSRS....73
5.2.2 Report using PowerBI - 1....78
5.2.3 Report using PowerBI - 2....81
5.2.4 Cube creation using SSAS and Reporting on SSRS....84
5.2.5 SSAS....93
**References....100**

# Section 1: Introduction

A Data Warehouse solution is a centralized data repository tailored to an organization's analytical and reporting needs. It gathers, cleans, and organizes data from numerous sources, bringing information from many departments together in a single, easily accessible area. Data warehouses store historical data, allowing firms to discover trends, patterns, and previous performance, all of which are important for strategic decision-making. Data is frequently modified in order to accommodate analytical inquiries, and specialized tools and query languages make data retrieval and analysis easier.

Retail operations create a large amount of data, such as sales transactions, inventory management, customer interactions, and marketing activities. The data warehouse centralizes this information, facilitating data administration and analysis. Historical data preservation allows merchants to track sales patterns, consumer behavior, and product performance over time, aiding in inventory planning and marketing tactics. Cross-functional analysis identifies links between sales, marketing activities, and inventory management, allowing for greater operational efficiency. Overall, a data warehouse enables retail companies to use data-driven insights, improve decision-making, and enhance the consumer experience.

The goal of this project is to design and build a data warehouse for Dominick's Fine Food (DFF), a retail shop chain with many locations in the Chicago region. This project seeks to capitalize on the large and unique information collected by Chicago Booth to build a powerful data warehouse that will allow DFF to obtain significant business insights, enhance shelf management and pricing strategies, and make data-driven choices throughout its retail operations.

The data warehouse will be designed in such a way that it addresses the problems a retail store chain like Dominick’s Finer Foods might face in its journey to grow and become successful in the Chicago market. This report details the steps involved in creating a data warehouse that will answer business questions that assist DFF’s management in making data-driven decisions.

The problems a retail store like DFF might face can range from profit generation, inventory management, supply chain streamlining, customer satisfaction, targeted marketing, and so on. In this report, our primary focus is directed towards two crucial areas, namely profit generation and the enhancement of marketing strategies. These two aspects are inherently interconnected and exert a significant influence on the overall success and sustainability of the retail business. Profit generation serves as the lifeblood of any enterprise, while effective marketing strategies are instrumental in driving revenue and ensuring that the business remains competitive and relevant in a dynamic market environment.

For example, the report contains the trend of weekly aggregations of beer sales over the course of one full year. This information allows the management of DFF to understand the seasonality of beer sales across all stores and assist with inventory planning and forecasting demand for future years. This information can also be computed on a store-wise basis to understand demand at each store and hence planning can be done in a more organized and scientific manner. Similarly, each business question that this report discusses and the answers the data warehouse assists in developing will have a significant impact on the decisions the business will have to make to solve problems.

## 1.1 Details about the Data

During the span of 1989 to 1997, Chicago Booth and Dominick's Finer Foods teamed up to embark on a joint venture in the realm of store-level research, particularly focusing on shelf management and pricing. They carried out randomized experiments across more than 25 diverse product categories within all 100 stores of this retail chain. This collaborative effort resulted in the collection of retail data that has since been used in multiple research initiatives in the field of marketing and sales. In this project, we will be using this data to build a data warehouse that can answer a wide range of questions about various trends across different stores managed by Dominick's Finer Foods.

The data can be broadly divided into general and category-specific files. The general files have information about all categories whereas the rest have information split for each category.

### 1.1.1 General Files

#### Customer Count (CCOUNT.csv)

The customer count file contains details regarding the number of visitors to the store and their purchases on a daily basis, specific to each store location [4]. Additionally, this file includes data on the total dollar sales and the number of coupons redeemed, categorized by department as defined by DFF. These statistics are collected daily from the receipts generated by the cash registers and scanners. This table has statistics about sales for each category on a daily basis. The granularity for this file is at a daily level.

#### Store-Level Demographics (DEMO.csv)

The demographics file contains detailed information based on the US Government census. This file contains information regarding different age categories, income levels, ethnicity, family members, and the distribution of retired and working people for each of the stores.

#### Dominick Store Locations

This table contains information about each store. It stores the store number, city, price tier, zone, and address data for each store.

#### Week's Decode Table

This table stores the start and end date of each week's number that is being used in other tables in the system such as the CCOUNT and Movement tables. This allows us to decipher which week corresponds to which date and use that to analyze trends better.

### 1.1.2 Category-Specific Files

#### Movement Data (WXXX.csv)

The movement files house sales data for each UPC within a specific category, organized by store and presented on a weekly basis. These files follow a naming convention where they are labeled as "wxxx," with "xxx" representing the three-letter acronym for the respective category. The granularity level of sales in this table is weekly. This table has the price, quantity, and movement information which can be used to calculate the total sales in dollars for a particular UPC. It also stores profit information which can be an important metric for various kinds of analysis.

#### UPC Data (UPCXXX.csv)

The UPC files provide descriptions for every UPC within a given category. These files adhere to a naming convention where they are denoted as "upcxxx," with "xxx" being the three-letter acronym that corresponds to the specific category. This table has the UPC code for each product, the item code that DFF uses internally, and a description for each product. We can also distinguish between the status of each product to see if it is discontinued or if it is available only in Combo stores and so on.

## 1.2 Metadata for all Source files

Customer Count

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

| Variable | Description | Type | Length |
|---|---|---|---|
| DELI | Deli Sales in Dollars | Numeric | 8 |
| DELICOUP | Deli Coupons Redeemed | Numeric | 8 |
| DELIEXPR | Deli Express Sales in Dollars | Numeric | 8 |
| DELISELF | Deli Self Service Sales in Dollars | Numeric | 8 |
| FISH | Fish Sales in Dollars | Numeric | 8 |
| FISHCOUP | Fish Coupons Redeemed | Numeric | 8 |
| FLORAL | Floral Sales in Dollars | Numeric | 8 |
| FLORCOUP | Floral Coupons Redeemed | Numeric | 8 |
| FROZCOUP | Frozen Items Coupons Redeemed | Numeric | 8 |
| FROZEN | Frozen Items Sales | Numeric | 8 |
| FTGCCOUP | Food-to-Go Coupons Redeemed | Numeric | 8 |
| FTGCHIN | Food-to-Go Chinese Sales in Dollars | Numeric | 8 |
| FTGICOUP | Food-to-Go Coupons Redeemed | Numeric | 8 |
| FTGITAL | Food-to-Go Italian Sales in Dollars | Numeric | 8 |
| GM | General Merchandise Sales in Dollars | Numeric | 8 |
| GMCOUP | General Coupons Redeemed | Numeric | 8 |
| GROCCOUP | Grocery Coupons Redeemed | Numeric | 8 |
| GROCERY | Grocery Sales in Dollars | Numeric | 8 |
| HABA | Health and Beauty Aids Sales in Dollars | Numeric | 8 |
| HABACOUP | Health and Beauty Aids Coupons Redeemed | Numeric | 8 |
| JEWELRY | Jewelry Sales in Dollars | Numeric | 8 |
| LIQCOUP | Liquor Coupons Redeemed | Numeric | 8 |
| MANCOUP | Manufacturer Coupons Redeemed | Numeric | 8 |
| MEAT | Meat Sales in Dollars | Numeric | 8 |
| MEATCOUP | Meat Coupons Redeemed | Numeric | 8 |
| MEATFROZ | Meat-Frozen Sales in Dollars | Numeric | 8 |
| MISCSCP | Misc. Coupons Redeemed | Numeric | 8 |
| MVPCLUB | MVP | Numeric | 8 |
| PHARCOUP | Pharmacy Coupons Redeemed | Numeric | 8 |
| PHARMACY | Pharmacy Sales in Dollars | Numeric | 8 |
| PHOTCOUP | Photo Coupons Redeemed | Numeric | 8 |
| PHOTOFIN | Photo | Numeric | 8 |
| PRODCOUP | Produce Coupons Redeemed | Numeric | 8 |
| PRODUCE | Produce Sales in Dollars | Numeric | 8 |
| PROMCOUP | Promotion Coupons Redeemed | Numeric | 8 |
| PROMO | Promotion Sales in Dollars | Numeric | 8 |
| SALADBAR | Salad Bar Sales in Dollars | Numeric | 8 |
| SALCOUP | Salad Coupons Redeemed | Numeric | 8 |

| Variable | Description | Type | Length |
|---|---|---|---|
| SPIRITS | Spirits Sales in Dollars | Numeric | 8 |
| SSDELICP | Self Service Deli Sales in Dollars | Numeric | 8 |
| VIDCOUP | Video Coupons Redeemed | Numeric | 8 |
| VIDEO | Video Sales in Dollars | Numeric | 8 |
| VIDEOREN | Video Rentals (Dollar Amounts) | Numeric | 8 |
| WINE | Wine Sales in Dollars | Numeric | 8 |

Demographics

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

| Variable Name | Description |
|---|---|
| nwrkch5 | % of non-working women with children under 5 |
| nwrkch17 | % of non-working women with children 6 - 17 |
| wrkch | % of working women with children |
| nwrkch | % of non-working women with children |
| wrkwch | % of working women with children under 5 |
| wrkwnch | % of working women with no children |
| telephn | % of households with telephones |
| mortgage | % of households with mortgages |
| nwhite | % of population that is non-white |
| poverty | % of population with income under $15,000 |
| shopcons | % of Constrained Shoppers |
| shophurr | % of Hurried Shoppers |
| shopavid | % of Avid Shoppers |
| shopstr | % of Shopping Stranges |
| shopunft | % of Unfettered Shoppers |
| shopbird | % of Shopper Birds |
| shopindx | Ability to Shop (Car and Single Family House) |
| shpindx | Ability to Shop (Car and Single Family House) |

UPC

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

Movement

| Variable | Description | Type | Length |
|---|---|---|---|
| upc | UPC Number | Numeric | 8 |
| store | Store Number | numeric | 3 |
| week | Week Number | Numeric | 3 |
| move | Number of unit sold | Numeric | 8 |

| | | | |
|---|---|---|---|
| price | Retail Price | Numeric | 8 |
| qty | Number of item bundled together | Numeric | 3 |
| profit | Gross margin | Numeric | 8 |
| sale | Sale code (B,C,S) | Character | 8 |
| ok | 1 for valid data, 0 for trash | Numeric | 3 |

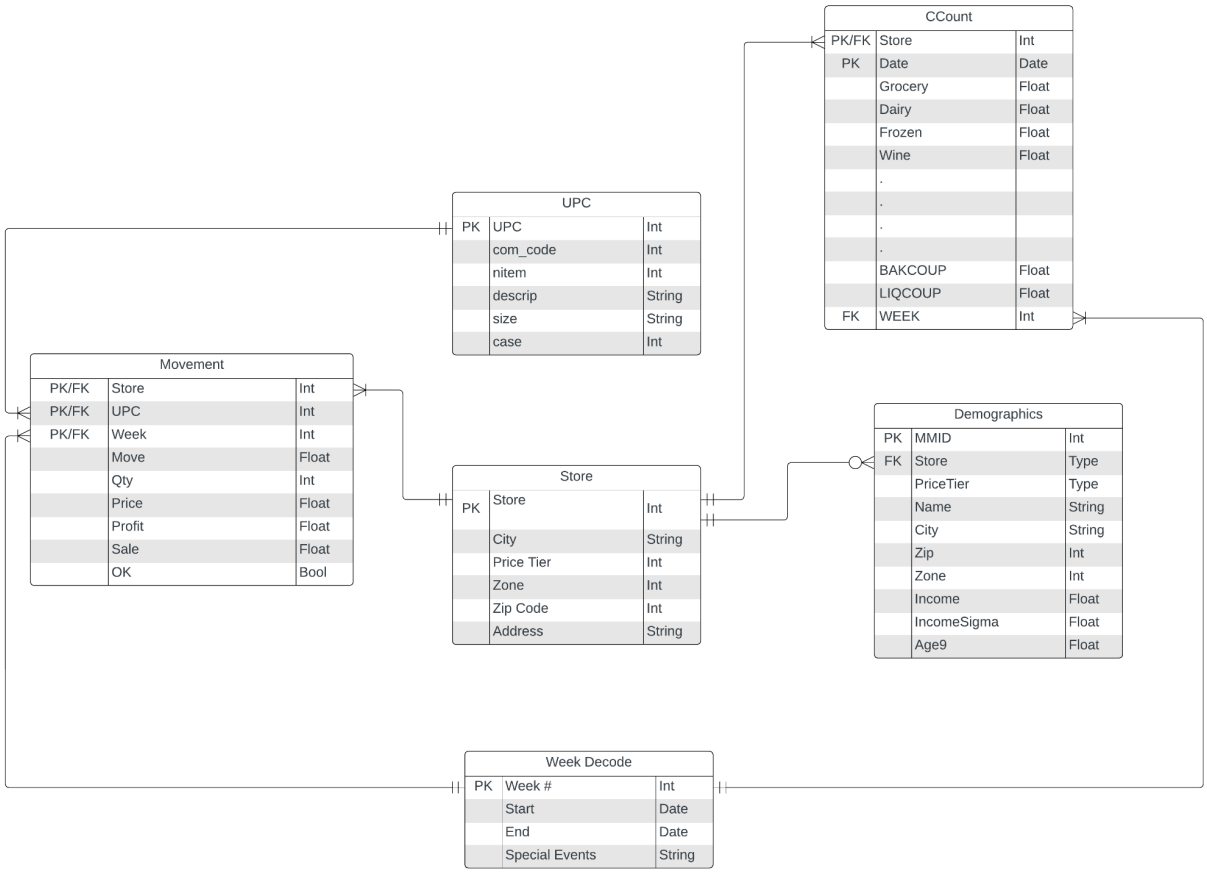
Store

| Variable | Description | Type | Length |
|---|---|---|---|
| Store | Store code | Numeric | 3 |
| City | City Name | Character | 20 |
| Price Tier | Price segmentation | Character | 10 |
| Zone | Store location zone | Numeric | 6 |
| Zip code | Postal code of store | Numeric | 8 |
| Address | Physical address | Character | 24 |

Week decode

| Variable | Description | Type | Length |
|---|---|---|---|
| Week # | Week number | Numeric | 3 |
| Start | Start Date of week | Date | 20 |
| End | End Date of week | Date | 20 |
| Special Events | Holiday information | Character | 20 |

## 1.3 Entity Relationship Diagram


CCount
PK/FK Store Int
PK Date Date
Grocery Float
Dairy Float
Frozen Float
Wine Float
.
.
.
.
BAKCOUP Float
LIQCOUP Float
FK WEEK Int
UPC
PK UPC Int
com_code Int
nitem Int
descrip String
size String
case Int
Movement
PK/FK Store Int
PK/FK UPC Int
PK/FK Week Int
Move Float
Qty Int
Price Float
Profit Float
Sale Float
OK Bool
Store
PK Store Int
City String
Price Tier Int
Zone Int
Zip Code Int
Address String
Demographics
PK MMID Int
FK Store Type
PriceTier Type
Name String
City String
Zip Int
Zone Int
Income Float
IncomeSigma Float
Age9 Float
Week Decode
PK Week # Int
Start Date
End Date
Special Events String

# Section 2: Subject Area Understanding

There were multiple resources at our disposal to help us understand more about the business and operations of big retail chains. We picked three articles and research papers that gave us a better understanding of the retail industry and the challenges faced by this industry historically.

The article titled “The Geography of Brands” [1] discusses a new approach to studying the performance of nationally distributed brands in the consumer packaged goods (CPG) industry. This is especially relevant to this report because Dominick's Fine Food also happens to be a big brand with multiple stores. The study conducted by University of Chicago Graduate School of Business professors, Sanjay K. Dhar, Jean-Pierre Dubé, and Bart J. Bronnenberg of UCLA, found significant geographic patterns in the performance of national CPG brands. The research challenges the concept of a "national brand" by highlighting the substantial variation in market shares, perceived quality, and local brand-share leaders across different geographic markets.

The authors point out that studying variations in market shares across different markets can be more informative about the long-term effects of marketing investments, advertising, and distribution. This approach may impact strategic marketing decisions regarding product positioning and branding.

In the research paper titled “Quality-Tier Competition Optimal Pricing” [2], the author, Sivakumar discusses the implications of competing based on the quality tier of a product. They have developed a conceptual framework to explore how to make optimal pricing decisions based on the quality level of a brand. This article explains how a comparative analysis can be conducted between two different products in terms of their pricing. It also delves into the significance of this framework and the steps required to implement it. This will be beneficial in devising pricing strategies for items in DFF, especially in the face of competition, utilizing the data warehouse that is planned for the upcoming stages of the project.

Additionally, the author provides insights into a customer decision model, which elucidates the potential decisions a customer might make when comparing the price and brand category of two competing products. This concept is valuable in determining the appropriate pricing for each product, considering the utility derived from it. The framework outlined in this paper suggests an empirical distribution of possible prices for a product. It also outlines the profits attainable at each price point within the distribution and the managerial implications of these findings. Overall, the paper establishes a comprehensive connection between academic research and practical marketing approaches for pricing a product.

In a study published in the European Journal of Operational Research, titled "The Value of Competitive Information in Forecasting FMCG Retail Product Sales and the Variable Selection Problem," [3] researchers made use of the Dominick's data set to examine the utility of competitive information in predicting sales at the UPC (Universal Product Code) level. The researchers noted that the UPC level has a significantly larger number of competing products compared to the brand level. They illustrated this point with the example of soft drinks, which can encompass hundreds of UPC items tailored to various customer preferences, all falling under just a few brand categories (as mentioned in Huang et al., 2014, "The Value of Competitive Information in Forecasting FMCG Retail Product Sales and the Variable Selection Problem").

Furthermore, the researchers emphasized the presence of numerous competitive explanatory variables, making it crucial to identify the most pertinent ones for accurate sales forecasting. The Dominick's data set contains thousands of UPC records and accurately predicting sales at the UPC level for inventory management poses a formidable challenge due to the extensive UPC information in the dataset and the multitude of competitive variables that can be generated when examining historical prices and promotions.

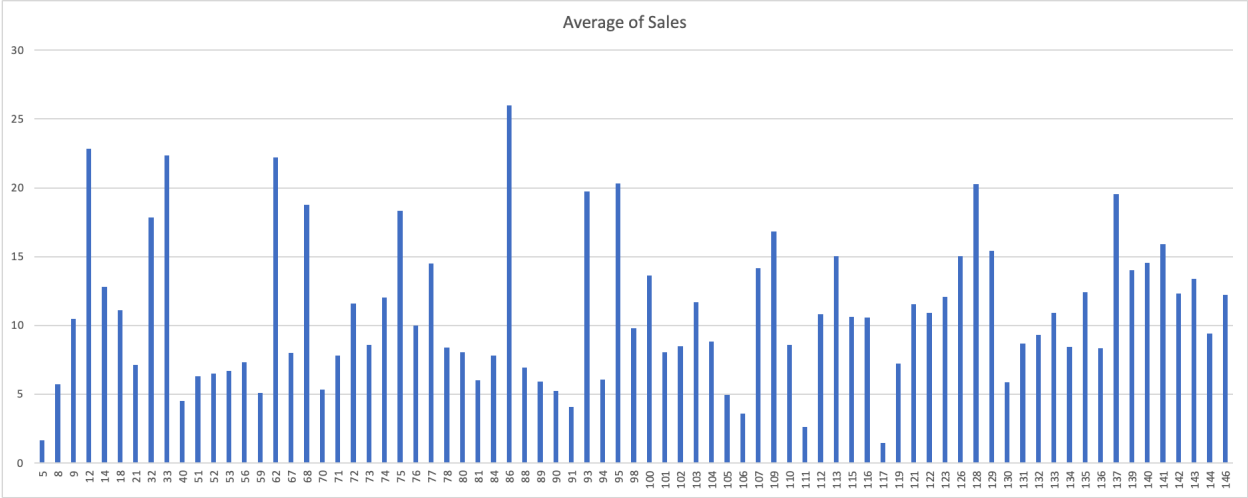
## 2.1 Business Questions

The business questions below have been ordered from highest priority to lowest priority. Priority was determined based on the impact the answers to these questions would have on the growth of the business. We used the justification of each business question to assess the prioritization of the questions and ordered them accordingly.

**BQ 1**: What are the average sales for each store in the years X and Y?

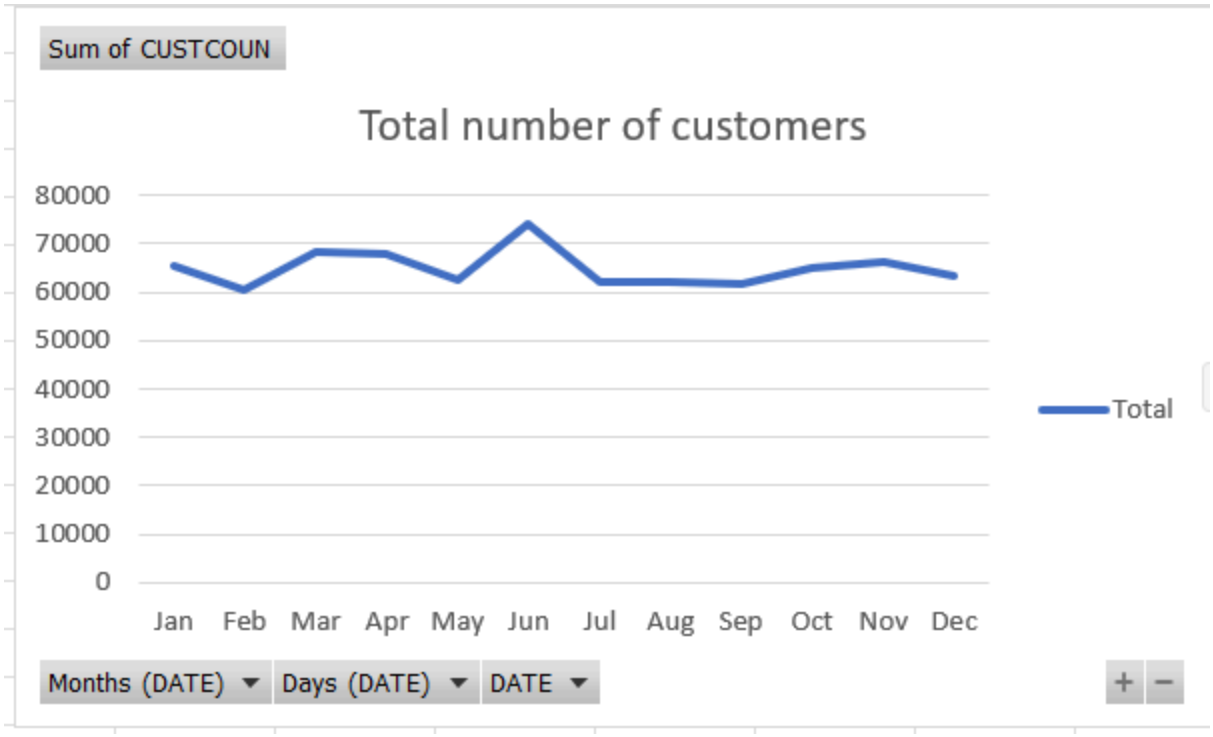
| Store | Average of Sales | | Store | Average of Sales | | Store | Average of Sales |
|---|---|---|---|---|---|---|---|
| 5 | 1.671466194 | | 78 | 8.389957677 | | 115 | 10.59998623 |
| 8 | 5.709716834 | | 80 | 8.038079918 | | 116 | 10.56769416 |
| 9 | 10.47151529 | | 81 | 5.999620056 | | 117 | 1.482655119 |
| 12 | 22.82643122 | | 84 | 7.824191693 | | 119 | 7.232052029 |
| 14 | 12.78191434 | | 86 | 25.9680367 | | 121 | 11.56624113 |
| 18 | 11.11479803 | | 88 | 6.917887749 | | 122 | 10.91823193 |
| 21 | 7.117528348 | | 89 | 5.941306931 | | 123 | 12.10117039 |
| 32 | 17.82888216 | | 90 | 5.240064191 | | 126 | 15.03043474 |
| 33 | 22.34290594 | | 91 | 4.079924242 | | 128 | 20.28484195 |
| 40 | 4.493893838 | | 93 | 19.72789069 | | 129 | 15.42187107 |
| 51 | 6.295399338 | | 94 | 6.057303993 | | 130 | 5.872650965 |
| 52 | 6.489087915 | | 95 | 20.3026195 | | 131 | 8.690279839 |
| 53 | 6.684047575 | | 98 | 9.822874482 | | 132 | 9.334305021 |
| 56 | 7.344777328 | | 100 | 13.64927841 | | 133 | 10.90539255 |
| 59 | 5.076799114 | | 101 | 8.07602269 | | 134 | 8.439238156 |
| 62 | 22.19245067 | | 102 | 8.511162501 | | 135 | 12.39736882 |
| 67 | 7.990439167 | | 103 | 11.66662408 | | 136 | 8.331050477 |

| Store | Average of Sales | | Store | Average of Sales | | Store | Average of Sales |
|---|---|---|---|---|---|---|---|
| 68 | 18.7556074 | | 104 | 8.829718543 | | 137 | 19.56360024 |
| 70 | 5.322666379 | | 105 | 4.947358102 | | 139 | 14.03721928 |
| 71 | 7.795954245 | | 106 | 3.594035964 | | 140 | 14.52577453 |
| 72 | 11.60363317 | | 107 | 14.16141709 | | 141 | 15.92964306 |
| 73 | 8.584834787 | | 109 | 16.81393594 | | 142 | 12.30326792 |
| 74 | 12.02487398 | | 110 | 8.60617394 | | 143 | 13.37139762 |
| 75 | 18.34619823 | | 111 | 2.651397195 | | 144 | 9.403635881 |
| 76 | 10.00485321 | | 112 | 10.79672421 | | 146 | 12.24319592 |
| 77 | 14.48527915 | | 113 | 15.04097668 | | **Grand Total** | **11.11843193** |




**Justification**: This data gives an overall view of the weekly average of sales taking place in different stores. This allows the management to analyze how much a store sells on average for a given week in all departments. The management can then use this information to make decisions on marketing spend for each store based on the revenue each store generates. They may choose to spend more on underperforming stores or on the contrary spend more on stores that do well to further encourage customers to buy more products. Hence, it is important to track the performance of each store.

**BQ 2:** What does total footfall for Store X look like throughout a particular year?

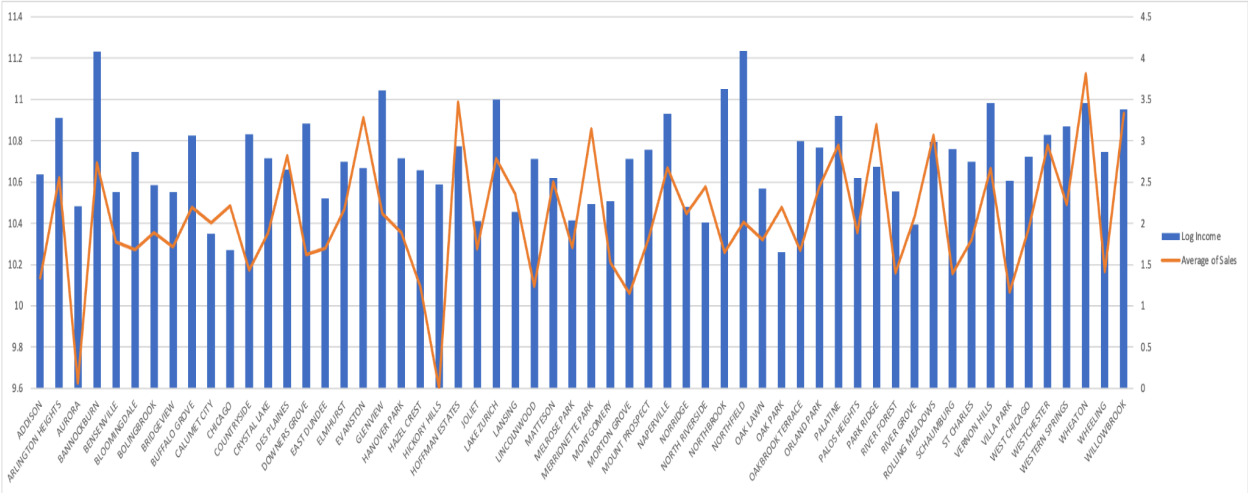


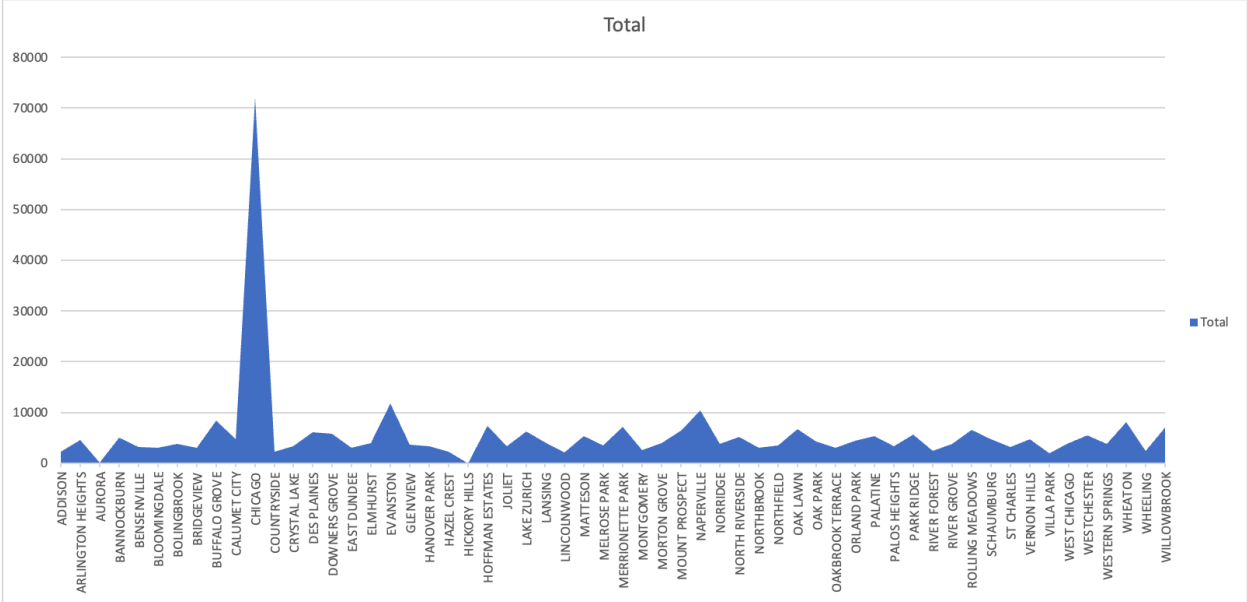

**Justification**: This data will be useful to a store manager wanting to know how many customers that store attracts throughout the year. With just this example, we see that there is a sharp increase in customers for the month of June. This insight could help the manager make business decisions. One example of such a decision could be to increase the number of employees working in that store to cater to the increasing demand. This information could also be used by managers in corporate offices to assess the traffic for each of their stores.

**BQ 3:** What is the city-wise trend of sales of grooming products in the year X? Compare this with the logarithmic income median of each city.

| City | Log Income | Average of Sales | Sum of Quantity of products sold | | City | Log Income | Average of Sales | Sum of Quantity of products sold |
|---|---|---|---|---|---|---|---|---|
| ADDISON | 10.63532646 | 1.339458151 | 2198 | | MERRIONETTE PARK | 10.49385422 | 3.144140394 | 7097 |
| ARLINGTON HEIGHTS | 10.90950929 | 2.549148751 | 4505 | | MONTGOMERY | 10.50684678 | 1.529683022 | 2636 |
| AURORA | 10.48374595 | 0.062232804 | 96 | | MORTON GROVE | 10.71126384 | 1.149785389 | 3921 |
| BANNOCKBURN | 11.23319834 | 2.733916026 | 5032 | | MOUNT PROSPECT | 10.75687382 | 1.80808936 | 6405 |
| BENSENVILLE | 10.54980515 | 1.777237388 | 3192 | | NAPERVILLE | 10.93117771 | 2.67013779 | 10319 |
| BLOOMINGDALE | 10.7445158 | 1.682449643 | 2987 | | NORRIDGE | 10.48001664 | 2.11759143 | 3823 |
| BOLINGBROOK | 10.58524512 | 1.883061112 | 3858 | | NORTH RIVERSIDE | 10.40483843 | 2.444347349 | 5119 |

| City | Log Income | Average of Sales | Sum of Quantity of products sold | | City | Log Income | Average of Sales | Sum of Quantity of products sold |
|---|---|---|---|---|---|---|---|---|
| BRIDGEVIEW | 10.55025042 | 1.715358249 | 3094 | | NORTHBROOK | 11.05101698 | 1.644073176 | 3065 |
| BUFFALO GROVE | 10.82521987 | 2.19405442 | 8334 | | NORTHFIELD | 11.23619652 | 2.013702062 | 3437 |
| CALUMET CITY | 10.34794899 | 2.002440415 | 4720 | | OAK LAWN | 10.56840671 | 1.798839188 | 6651 |
| CHICAGO | 10.2700928 | 2.212854234 | 72061 | | OAK PARK | 10.25895677 | 2.189865929 | 4200 |
| COUNTRYSIDE | 10.83182542 | 1.429181861 | 2308 | | OAKBROOK TERRACE | 10.79695991 | 1.670297964 | 3028 |
| CRYSTAL LAKE | 10.7150398 | 1.889626128 | 3349 | | ORLAND PARK | 10.76561773 | 2.440368072 | 4482 |
| DES PLAINES | 10.65993812 | 2.820212477 | 6065 | | PALATINE | 10.92237097 | 2.944204738 | 5365 |
| DOWNERS GROVE | 10.88355095 | 1.618227552 | 5763 | | PALOS HEIGHTS | 10.62083942 | 1.885817204 | 3322 |
| EAST DUNDEE | 10.52085999 | 1.693421498 | 3042 | | PARK RIDGE | 10.67447502 | 3.194377664 | 5590 |
| ELMHURST | 10.69711335 | 2.171258037 | 3888 | | RIVER FOREST | 10.55320517 | 1.394888109 | 2392 |
| EVANSTON | 10.66609936 | 3.284738313 | 11842 | | RIVER GROVE | 10.39197554 | 2.076005041 | 3780 |
| GLENVIEW | 11.04392933 | 2.115009993 | 3589 | | ROLLING MEADOWS | 10.79353697 | 3.062475517 | 6505 |
| HANOVER PARK | 10.71619397 | 1.886909621 | 3413 | | SCHAUMBURG | 10.75846583 | 1.390540631 | 4755 |
| HAZEL CREST | 10.65780039 | 1.22769265 | 2225 | | ST CHARLES | 10.69885282 | 1.798096677 | 3228 |
| HICKORY HILLS | 10.58930796 | 0 | 0 | | VERNON HILLS | 10.98312087 | 2.667272351 | 4763 |
| HOFFMAN ESTATES | 10.77505285 | 3.470579959 | 7331 | | VILLA PARK | 10.60581744 | 1.165966463 | 1946 |
| JOLIET | 10.41235125 | 1.693093056 | 3279 | | WEST CHICAGO | 10.72251821 | 1.941188584 | 3958 |
| LAKE ZURICH | 10.99986401 | 2.78768097 | 6241 | | WESTCHESTER | 10.82713118 | 2.946960961 | 5414 |
| LANSING | 10.45607873 | 2.358090909 | 4082 | | WESTERN SPRINGS | 10.86915875 | 2.223957697 | 3846 |
| LINCOLNWOOD | 10.71219307 | 1.23119195 | 2086 | | WHEATON | 10.98087621 | 3.811051514 | 8070 |
| MATTESON | 10.61915426 | 2.501073684 | 5356 | | WHEELING | 10.74537796 | 1.416061446 | 2389 |
| MELROSE PARK | 10.41439317 | 1.703564949 | 3491 | | WILLOWBROOK | 10.94992897 | 3.33384829 | 6962 |




**Justification**: This data provides information about sales of all grooming products across different cities in which DFF has stores. It also brings in the median income information for the people living in that particular city. This allows us to compare and understand the existence of any correlation between the income of a demographic and their spending habits since grooming products are usually not considered a necessity. Analyzing this information allows the management to direct stock and inventory management accordingly and also adjust marketing strategies according to city-wise spending patterns in an attempt to drive sales of grooming products.

**BQ 4:** What was the most selling category of product throughout a particular year?

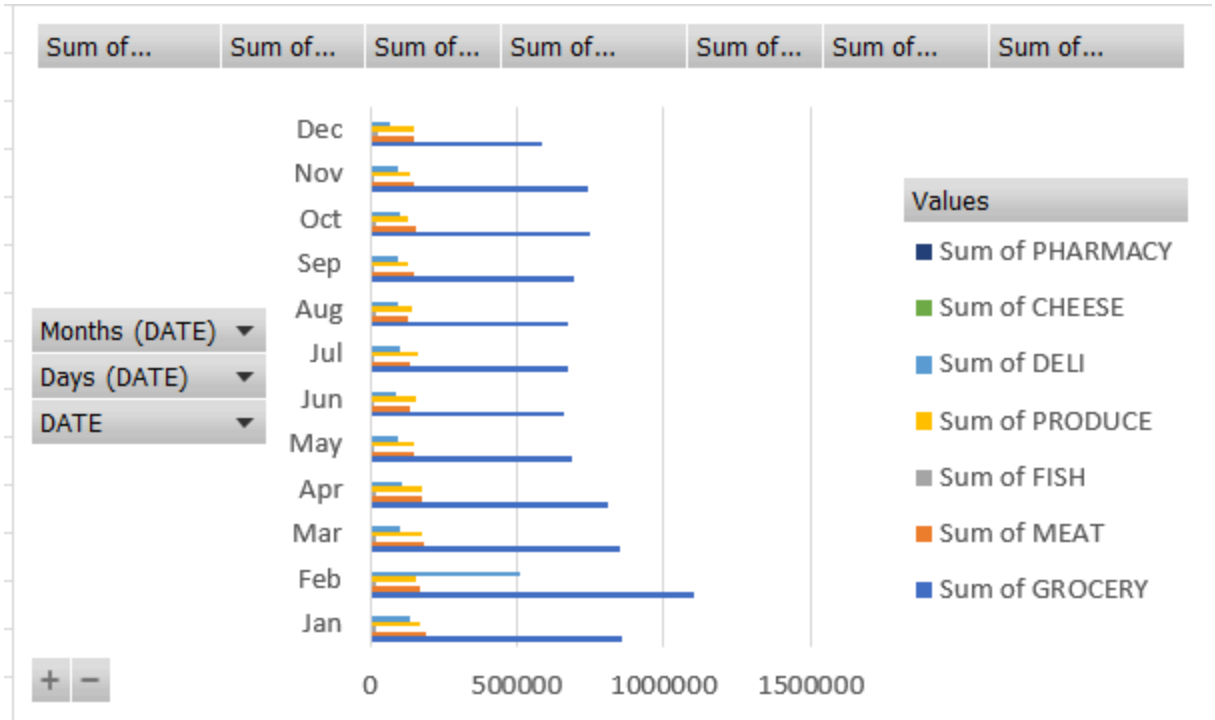


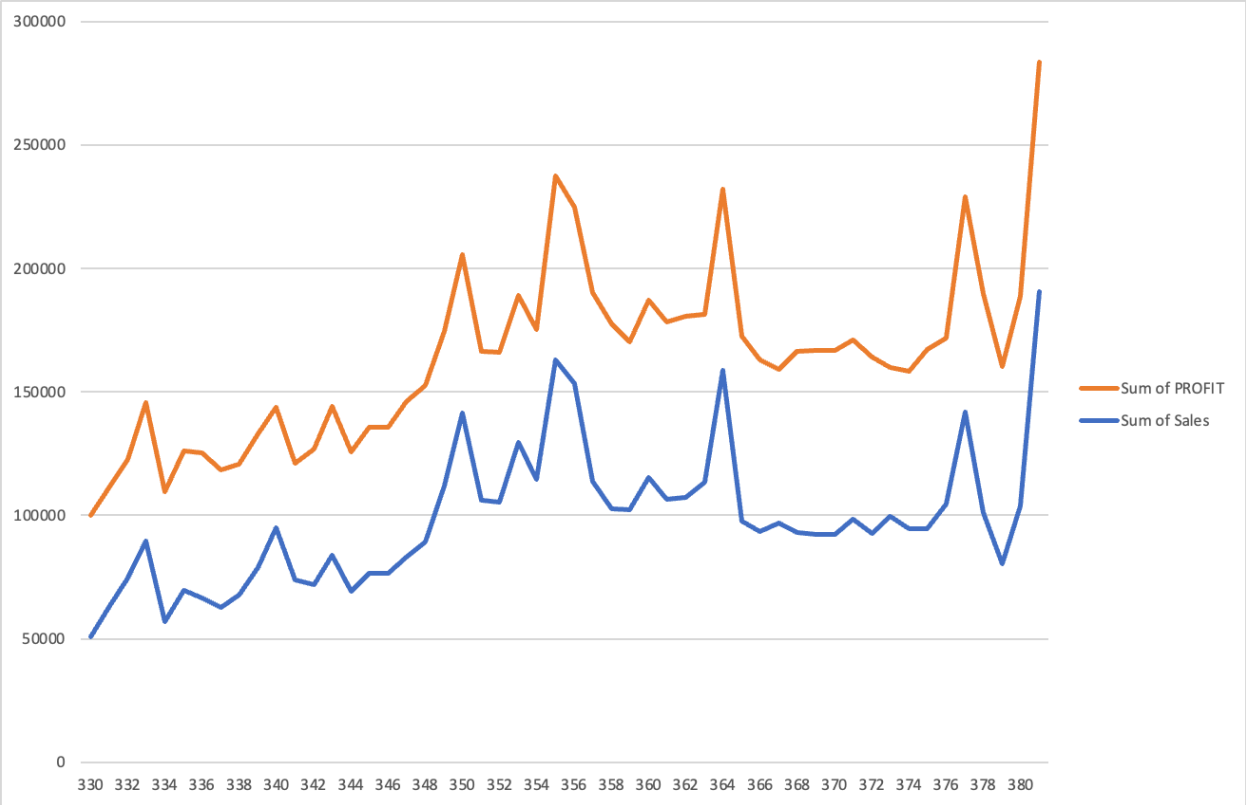

*Fig 1.2 Sales figures for major categories for each month*

**Justification**: This question helps us find out which category of products customers buy the most. This information could be used by the operations manager to make sure the inventory for that category of product is always stocked up. With the example shown in Fig 1.2, we see that grocery items sell the most throughout the year.

**BQ 5**: How do the week-wise profit and dollar sales vary for Beer in the year X?

| Week | Sum of Sales | Sum of PROFIT | | Week | Sum of Sales | Sum of PROFIT |
|---|---|---|---|---|---|---|
| 330 | 50702.38 | 49288.65 | | 357 | 113642.84 | 76717.93 |
| 331 | 62798.91 | 48368.02 | | 358 | 102560.92 | 74843.29 |
| 332 | 74538.63 | 47907.64 | | 359 | 102142.2 | 68236.78 |
| 333 | 89736.81 | 55806.43 | | 360 | 115310.31 | 71940.03 |
| 334 | 57007.68 | 52596.7 | | 361 | 106533.03 | 71649.18 |
| 335 | 69654.53 | 56400.55 | | 362 | 107380.62 | 73415.71 |
| 336 | 66345.18 | 59092.54 | | 363 | 113307.72 | 68243.26 |
| 337 | 62701.9 | 55869.21 | | 364 | 158837.58 | 73172.77 |
| 338 | 67646.66 | 53214.07 | | 365 | 97654.84 | 75009.19 |

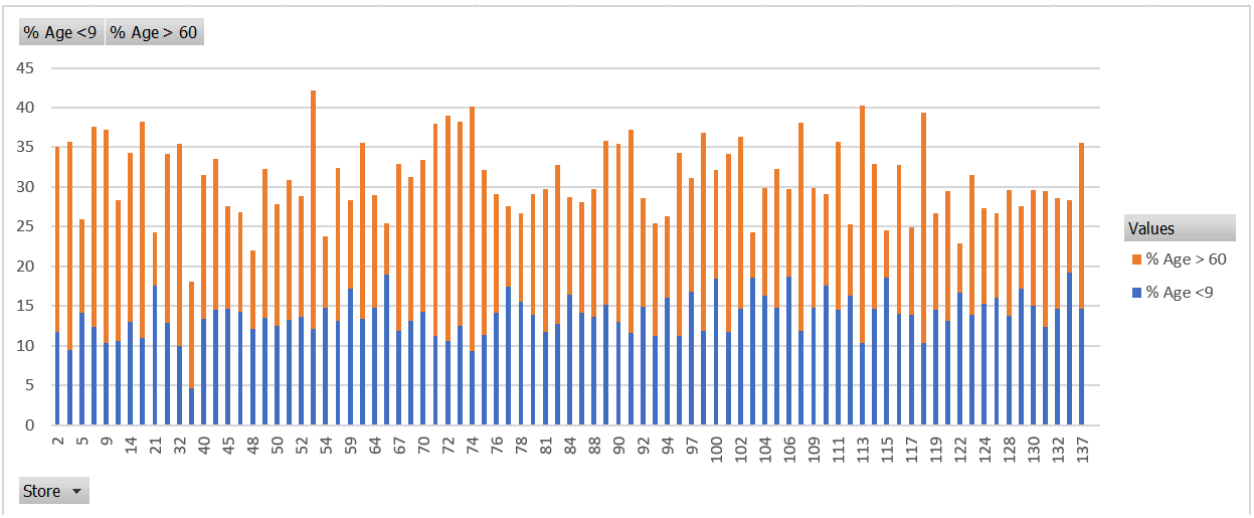
| Week | Sum of Sales | Sum of PROFIT | | Week | Sum of Sales | Sum of PROFIT |
|---|---|---|---|---|---|---|
| 339 | 78673.96 | 54262.08 | | 366 | 93372.2 | 69687.13 |
| 340 | 94975.2 | 48677.3 | | 367 | 96728.49 | 62310.27 |
| 341 | 74009.78 | 47009.25 | | 368 | 93023.77 | 73350.67 |
| 342 | 71821.97 | 55113.18 | | 369 | 92332.2 | 74370.11 |
| 343 | 83655.89 | 60585.67 | | 370 | 92332.2 | 74370.11 |
| 344 | 69227.9 | 56514.97 | | 371 | 98262.12 | 72844.25 |
| 345 | 76652.19 | 58892.79 | | 372 | 92704.57 | 71393.83 |
| 346 | 76496.15 | 59349.97 | | 373 | 99524.61 | 60567.15 |
| 347 | 83209.08 | 62736.47 | | 374 | 94705.15 | 63630.89 |
| 348 | 89269.21 | 63241.02 | | 375 | 94637.36 | 72577.79 |
| 349 | 111961.06 | 62678.12 | | 376 | 104429.83 | 67467.48 |
| 350 | 141311.43 | 64401.42 | | 377 | 141713.83 | 87370.7 |
| 351 | 106105.8 | 60206.5 | | 378 | 101195.76 | 88550.73 |
| 352 | 105245.56 | 60668.82 | | 379 | 80480.34 | 79729.66 |
| 353 | 129385.34 | 59780.25 | | 380 | 103784.57 | 85031.82 |
| 354 | 114734.32 | 60522.61 | | 381 | 190691.12 | 92829.59 |
| 355 | 163152.9 | 74347.86 | | | | |
| 356 | 153340.87 | 71507.39 | | **Grand Total** | **5111649.47** | **3408349.8** |




**Justification**: This data will be useful to a store manager wanting to know how the sales of beer vary throughout the year. Products like Beer tend to have seasonality attached to its sale and it would be helpful to have historical data prove this. For example, from the graph above, we can see that sales go up drastically towards the end of the year which is a festive period. This will help the organization prepare for higher demand for beer during this period. We may also want to have this data store-wise so that store managers can filter accordingly and make decisions based on their store.

**BQ 6:** For each store what is the percentage of customers below the age of 9 and above 60 years?

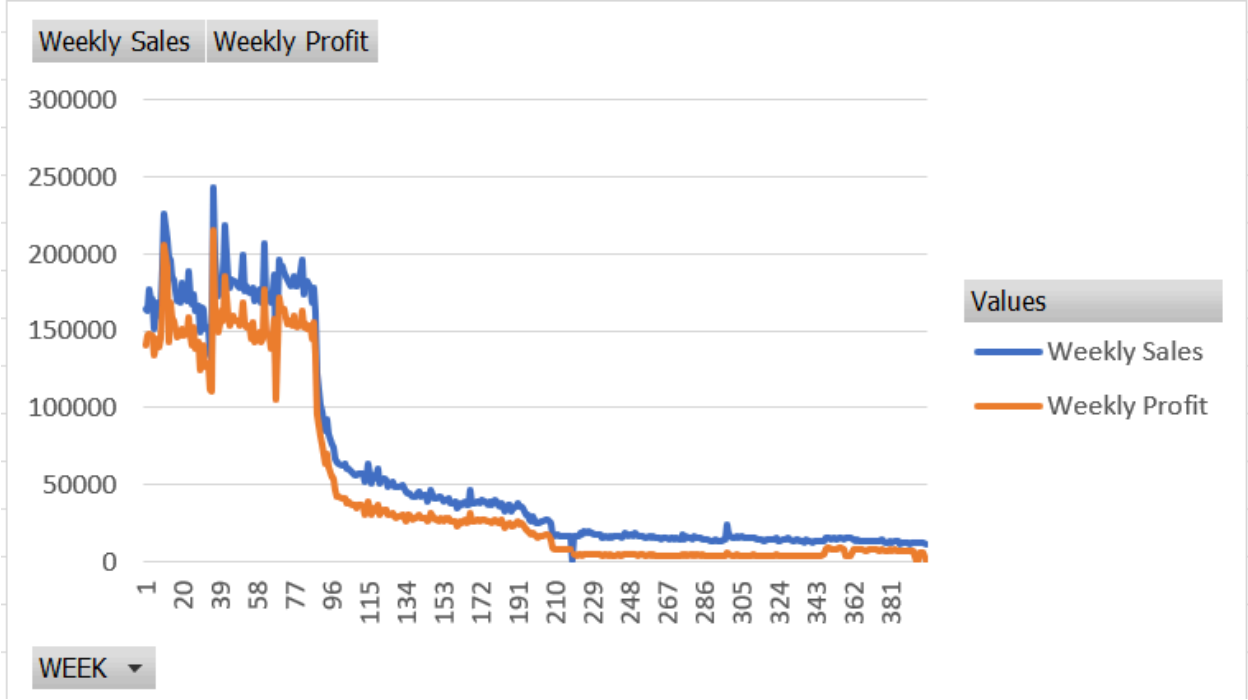
| Row Labels | % Age <9 | % Age > 60 |
|---|---|---|
| 2 | 11.7508576 | 23.2864734 |
| 4 | 9.50895057 | 26.202989 |
| 5 | 14.14334827 | 11.73680317 |
| 8 | 12.3155416 | 25.23940345 |
| 9 | 10.35030974 | 26.91190176 |
| 12 | 10.56967397 | 17.8341405 |
| 14 | 12.9589372 | 21.39492754 |
| 18 | 11.00949839 | 27.23133684 |
| 21 | 17.59263459 | 6.68964579 |
| 28 | 12.88795371 | 21.33087849 |
| 32 | 9.90606319 | 25.49530316 |
| 33 | 4.60709172 | 13.41699655 |
| 40 | 13.36846485 | 18.18518005 |
| 44 | 14.48834853 | 19.09827761 |
| 45 | 14.67187625 | 12.88573479 |
| 47 | 14.29616817 | 12.5798297 |
| 48 | 12.17670803 | 9.79219614 |
| 49 | 13.48777349 | 18.74731875 |
| 50 | 12.44204019 | 15.33573759 |
| 51 | 13.2472108 | 17.61597181 |
| 52 | 13.660619 | 15.22411953 |
| 53 | 12.08391392 | 30.02786809 |
| 54 | 14.79145854 | 9.02222777 |




**Justification:** The data above provides information about the demographics of the customers for each store. We look at the store-level information for customers below the age of 9 and above 60 years of age. This data will help the management to manage inventory levels, provide deals catering to the needs of the customers based on their age groups.

**BQ 7:** What is the weekly trend of sales for cigarettes across all stores?

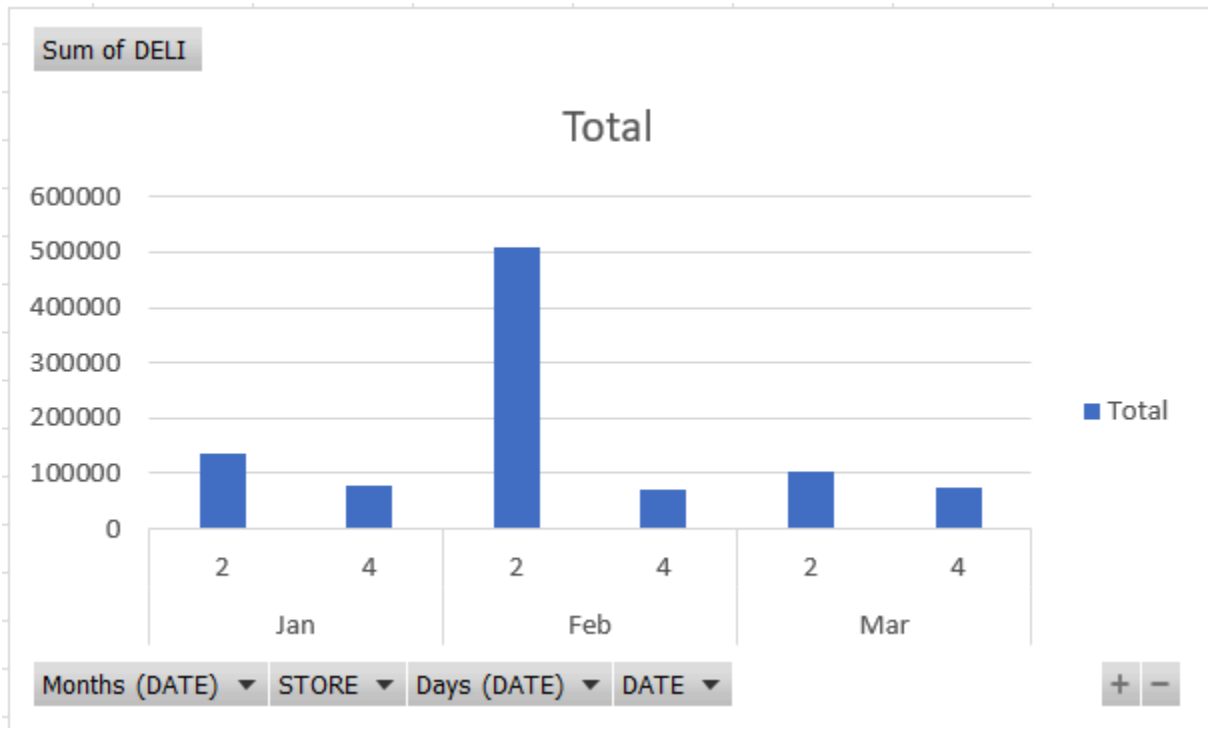
| Weeks | Weekly Sales | Weekly Profit |
|---|---|---|
| 1 | 164083.59 | 140959.8504 |
| 2 | 163224.94 | 147403.4517 |
| 3 | 177179.71 | 148504.3703 |
| 4 | 170255.15 | 147181.4259 |
| 5 | 170264.02 | 146912.5309 |
| 6 | 150833.73 | 133991.5145 |
| 7 | 167851.5 | 144964.25 |
| 8 | 159799.26 | 139321.5603 |
| 9 | 164314.94 | 144844.8331 |
| 10 | 190020.26 | 163645.5563 |
| 11 | 226202.05 | 206028.0828 |
| 12 | 210284.49 | 191601.2067 |
| 13 | 197183.37 | 142487.4576 |
| 14 | 196164.79 | 168147.915 |
| 15 | 184366.11 | 157720.1953 |
| 16 | 183544.26 | 156611.0092 |
| 17 | 169111.35 | 146308.8813 |




**Justification:**

The data provides information about the weekly trends in sales and profits across all DFF stores. With the introduction of alternate products, and the falling number of cigarette sales in the US [5] we went on to look for the sales of cigarettes across every DFF store and it points out the decline in sales and profits over the weeks. Management may handle this by using techniques like targeted promotions, inventory modifications, cost-cutting measures, and consumer engagement efforts.

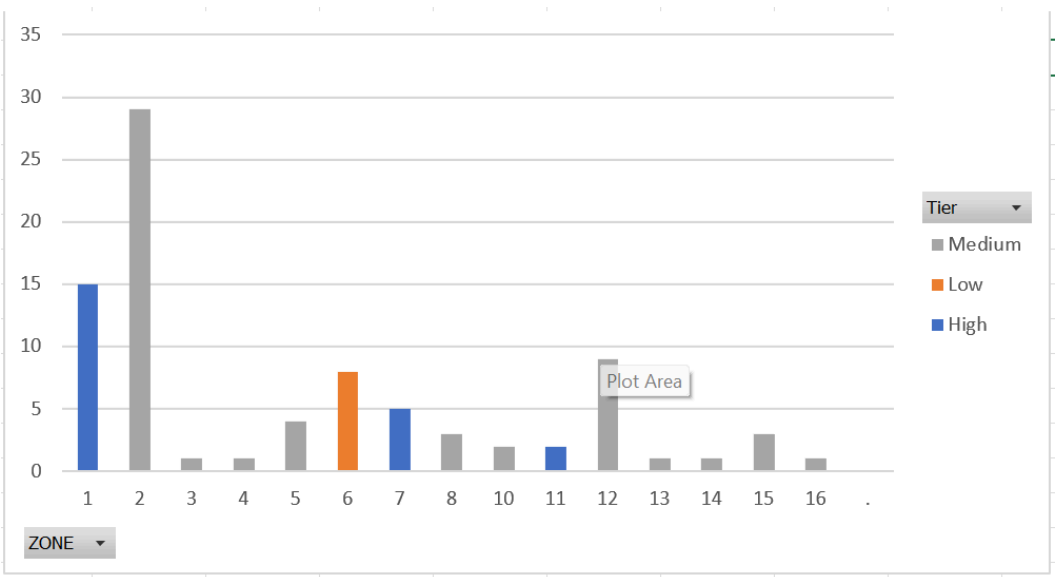
**BQ 8:** How do Store X and Store Y compare in deli sales for the first quarter of year?




**Justification:** A corporate office manager would want to know how certain stores perform in comparison to a particular category of product. In this question specifically, the manager wants to know the comparison for sales from the deli section between store 2 and store 4 for the first quarter of the year.

**BQ 9**: What is the distribution of stores based on tier and zones?

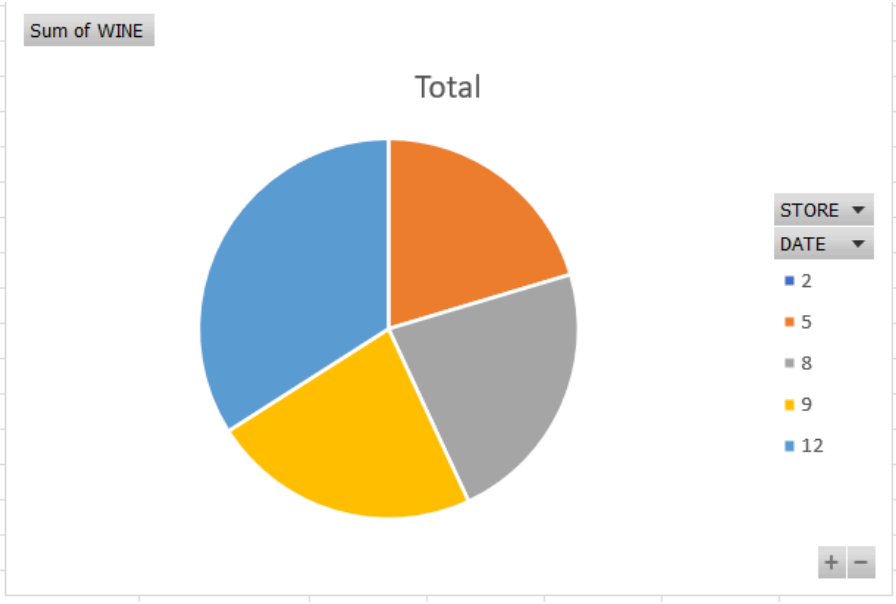
| Store Number | Column Labels | | | |
|---|---|---|---|---|
| Row Labels | High | Low | Medium | Grand Total |
| 1 | 15 | | | 15 |
| 2 | | | 29 | 29 |
| 3 | | | 1 | 1 |
| 4 | | | 1 | 1 |
| 5 | | | 4 | 4 |
| 6 | | 8 | | 8 |
| 7 | 5 | | | 5 |
| 8 | | | 3 | 3 |
| 10 | | | 2 | 2 |
| 11 | 2 | | | 2 |
| 12 | | | 9 | 9 |
| 13 | | | 1 | 1 |
| 14 | | | 1 | 1 |
| 15 | | | 3 | 3 |
| 16 | | | 1 | 1 |
| . | | | | |
| **Grand Total** | **22** | **8** | **55** | **85** |




**Justification:** The data tells us about the distribution of DFF stores across the 16 zones and also what tier stores are situated in each of these zones. The data helps the management make strategic decisions in terms of investments. Understanding store distribution across price tiers also aids in improving product placement and inventory management. By examining this data, management may make decisions regarding extending or consolidating stores in various zones, as well as adapting product offers to different customer needs, increasing profitability and development possibilities.

**BQ 10:** What were the wine sales for different stores in a given month?

| Row Labels | Sum of WINE |
|---|---|
| ⊞ 2 | 0 |
| ⊞ 5 | 26765.45 |
| ⊞ 8 | 29773.71 |
| ⊞ 9 | 30054.49 |
| ⊞ 12 | 44666.66 |
| Grand Total | 131260.31 |




**Justification:** This question gives us information about how much total sales of wine occurred for each store in the month of December. From this example, we can see that store 12 sold the most amount of wine. This information can then be used to drive business decisions. One example of such a decision could be to run marketing campaigns and offer discounts to further increase the purchase of wine from store 12.

## 2.2 Selected Business Questions

By integrating data from diverse sources, assuring data quality, and providing a single repository for historical data, data warehousing assists in solving business issues. This cleaned and sorted data is arranged into dimensional models for simple access and effective querying. Businesses may swiftly extract important information, assess trends, find patterns, and gain insights into their operations by using well-defined schemas. The capacity to generate multidimensional perspectives and access historical data enables businesses to make educated decisions and adjust to changing market conditions.

The following business questions were selected by DFF for our data warehouse project:

- What does total footfall for Store X look like throughout a particular year?
- What is the city-wise trend of sales of grooming products in the year X? Compare this with the logarithmic income median of each city.
- How do the week-wise profit and dollar sales vary for Beer in the year X?
- What is the weekly trend of sales for cigarettes across all stores?
- What were the wine sales for different stores in a given month?

# Section 3: Independent Data Marts Design using Kimball's Methodology

Ralph Kimball invented the Kimball technique, a commonly used methodology for developing and implementing data warehouses. It stresses the utility of data warehousing by concentrating on the demands of end users. Ralph Kimball's approach to data warehousing entails the development of independent and conformable data marts. These data marts are sections of the main data warehouse that cater to certain business dimensions.

## 3.1 Steps to follow under Kimball's Methodology

The following are the steps to implementing Kimball's technique for creating separate conformable data marts:

- **Identify a business process:** We begin by understanding the unique business requirements of the business unit for which a data mart is being created. Collaborate closely with end users to establish their needs and data analysis objectives. In our case, we focus on the sales and customer demographics at a store level to answer the business questions in focus.
- **Decide the Grain:** The data mart's granularity, or "grain," must be defined. This entails defining the amount of detail in which data will be kept. In a sales data mart, for example, the grain may be at the daily, weekly, or monthly level, depending on the needs of the organization. Choosing the appropriate grain is critical for striking a balance between detail and performance.
- **Conform Dimensions:** Ensure that dimension tables are conformed such that attributes and meaning are consistent across all data marts.
- **Decide the Facts:** Determine the essential measurements that will be included in the fact table based on the business requirements. Measures are often numerical in nature and indicate performance or quantitative data points such as revenue from sales, units sold, or customer counts.
- **Design and Built Data Marts:** Create a dimensional model for the chosen subject area, create the ETL procedures for extracting, transforming, and loading data, and build the data mart. Ensure data quality and incorporate any essential business logic.
- **Choosing Duration of Database:** Consider the effects of your data retention decision on data volume and storage needs. Keeping a large amount of historical data can drastically raise storage requirements and expenditures.
- **Populate the Data Marts:** Load the data mart with appropriate data depending on company needs. During the ETL process, ensure data quality, consistency, and correctness.

## 3.2 Why is it important to follow Kimball's Methodology?

Kimball's methodology streamlines the process for data warehouse architects, offering a systematic approach to design, development, and implementation. This framework ensures an efficient and coherent strategy for crafting data warehouses. More specifically, here are a few reasons why one might choose to follow Kimball's methodology for creating a data warehouse -

**Business Alignment**: Kimball's methodology focuses on business processes while coming up with the design of a warehouse. Hence, the data warehouse is built around business needs and it ensures that the warehouse assists in solving the questions posed by the business.

**Incremental Development**: Independent data marts allow for an incremental and iterative approach to data warehouse development. Instead of waiting for a comprehensive enterprise-wide solution, businesses can prioritize and deliver value more rapidly, one data mart at a time.

**Reduced Complexity**: Each data mart focuses on a particular subject area or business process. This split approach simplifies design, development, and implementation.

**Risk Mitigation**: Developing a comprehensive data warehouse is a massive and complex process. By breaking the process down into smaller chunks, the risk associated with each phase is limited.

## 3.3 Dimension Tables Design

In a data warehouse, a dimensional table holds descriptive qualities such as time, location, or product categories that offer context for quantitative values included in fact tables. Dimensional tables are used to facilitate effective data processing and reporting. These tables' attributes represent categorized or textual information, allowing for multi-dimensional analysis. Data warehouses may develop organized schemas like star or snowflake schemas by linking these dimension tables to fact tables via foreign keys, simplifying difficult queries and improving the entire data analysis process.

| **Dimension Table** | **Purpose** | **Fields** |
|---|---|---|
| DIM_STORE_DEMO | To store details about different stores, their location, and the income of people in those locations | Store_Key - Surrogate key that identifies the store |
| | | Store_Number: Number of the store |
| | | City: Name of city where the store is located |
| | | Log_Income: Log of the income of people in that city |
| DIM_TIME | To capture data about time like day, month and year | Date_Key: Surrogate key which uniquely identifies the date |
| | | Week_Number: Week of the date |
| | | Month: Month of the date |

| | | |
|---|---|---|
| | | Year: Year of the date |
| DIM_PRODUCT | To capture details about each the product | UPC_Key: Surrogate key for each unique product |
| | | UPC_Number: Code for the product |
| | | Category: Category of the product |

## 3.4 Fact Tables Design

In a data warehouse, a fact table comprises numerical data that reflects key performance indicators, metrics, or measurements. It saves information about individual events or transactions, such as sales income, unit sales, or customer orders. Fact tables serve to help data analysis and business intelligence by providing a numerical foundation for decision-making. To enable multi-dimensional analysis, fact tables are linked to dimensional tables using foreign keys, allowing enterprises to get insights from their data.

| **Fact Table** | **Purpose** |
|---|---|
| FACT_SALES | To answer questions about sales of different products |
| FACT_CUSTOMER | To answer questions about customers visiting the stores |

### Dimension Matrix

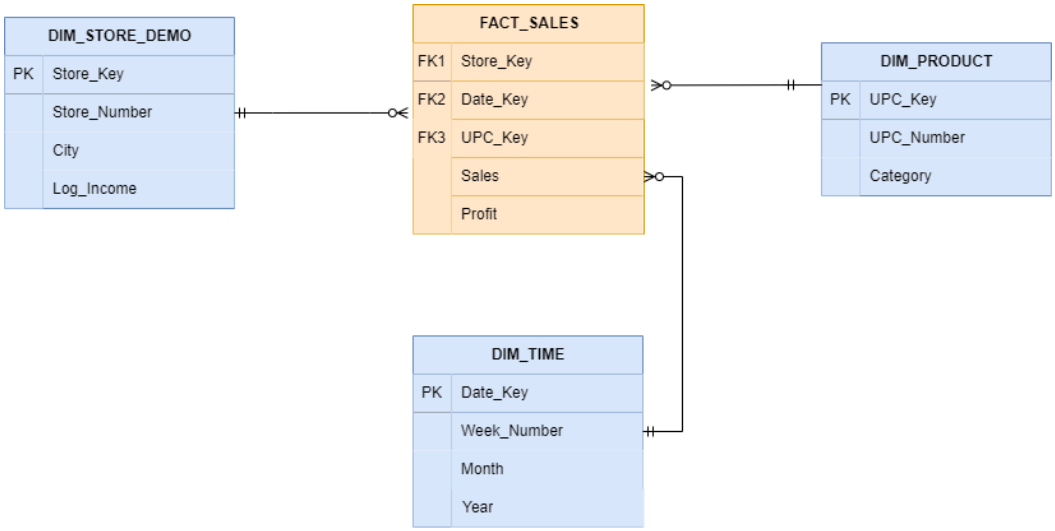
| **Data Marts** | **Dimension** | | |
|---|---|---|---|
| | DIM_PRODUCT | DIM_DATE | DIM_STORE_DEMO |
| SALES DATA MART | ✅ | ✅ | ✅ |
| CUSTOMER DATA MART | | ✅ | ✅ |

# 3.5 Data Warehouse Schema

## Star Schema

A star schema is a data warehouse paradigm in which data is organized into a core fact table surrounded by numerous related dimension tables. The fact table contains quantitative data, such as sales statistics or income, whereas the dimension tables provide descriptive features, such as time, location, and product information. Because it enables easy browsing through several dimensions for insights, this design streamlines data analysis and reporting. Star schemas are an effective format for complicated queries and are widely used in data warehousing to facilitate corporate intelligence and decision-making.
In our design, we have two data marts i.e. Fact sales data mart and fact customer data mart which are used to analyze sales distribution for products across stores and timelines.

## Sales Data Mart




## Customer Data Mart

## 3.6 Schema Justification for Business Questions

**BQ2**: What does total footfall for Store X look like throughout a particular year?

This data will be useful to a store manager wanting to know how many customers that store attracts throughout the year. With just this example, we see that there is a sharp increase in customers for the month of June. This insight could help the manager make business decisions.

*FACT_CUSTOMER* along with the two Dimension tables viz, *DIM_TIME* and *DIM_STORE_DEMO* will be used to answer this question. Store_Key column will give us the store we are trying to answer the question for. The aggregated value of Customer_Count and the year obtained from the *DIM_DATE* will help us answer this question.

**BQ3**: What is the city-wise trend of sales of grooming products in the year X? Compare this with the logarithmic income median of each city.

This data provides information about sales of all grooming products across different cities in which DFF has stores. It also brings in the median income information for the people living in that particular city.

*FACT_SALES* together with *DIM_DATE*, *DIM_STORE_DEMO*, and *DIM_PRODUCT* will be used to answer this question. UPC_Key column from the *FACT_SALES* will be joined with UPC_Key from *DIM_PRODUCT*. Columns Log_Income, City from *DIM_STORE_DEMO*, and Year from *DIM_DATE* will also be used to answer this question.

**BQ 5**: How do the week-wise profit and dollar sales vary for Beer in the year X?

This data will be useful to a store manager wanting to know how the sales of beer vary throughout the year. Products like Beer tend to have seasonality attached to its sale and it would be helpful to have historical data prove this.

The query can be resolved by referencing *FACT_SALES* for beer sales and profit data, consulting DIM_PRODUCT for product specifics, utilizing *DIM_STORE_DEMO* for store details, and referring to DIM_TIME for relevant week-specific information.

**BQ 7:** What is the weekly trend of sales for cigarettes across all stores?

The data provides information about the weekly trends in sales and profits across all DFF stores. Management may handle this by using techniques like targeted promotions, inventory modifications, cost-cutting measures, and consumer engagement efforts.

The question can be answered using *FACT_SALES* to get the cigarette sales and profit figures, *DIM_PRODUCT* to get product information, *DIM_STORE_DEMO* to get store information, *DIM_TIME* to fetch data pertaining to week number.

**BQ 10:** What were the wine sales for different stores in a given month?
This question gives us information about how much total sales of wine occurred for each store in the month of December. This information can then be used to drive business decisions.

Like BQ1, to answer this question we will need *FACT_CUSTOMER* along with the two Dimension tables viz, *DIM_TIME* and *DIM_STORE_DEMO.* Wine_Sales for a given store can be found in the *FACT_CUSTOMER* table. This information for the month, retrieved from *DIM_TIME,* will answer the question.

## 3.7 Mapping Tables

Mapping of fields between Source tables and Staging tables -

| **Source File Name** | **Source Data Field** | **Mapping** | **Staging Table Type** | **Staging Table Name** | **Staging Table Attribute** |
|---|---|---|---|---|---|
| DONE-XXXX.csv (Movement Tables) | STORE | Direct Copy | Relation | Movement | Store |
| | UPC | | | | Product_Code |
| | WEEK | | | | Week |
| | QTY | | | | Quantity |
| | PRICE | | | | Price |
| | MOVE | | | | Sale |
| | PROFIT | | | | Profit |
| | OK | | | | OK |
| | File Name (Beer/Grooming) | Use the file name to determine product category | | | Product_Category |
| Demo.csv (Demographic table) | STORE | Direct Copy | Relation | Demographics | Store |
| | CITY | | | | City |
| | INCOME | | | | Income |
| Week's Decode Table | Week # | Direct Copy | Relation | Week_Decode | Week_No |
| | Start | | | | Start |
| | End | | | | End |
| Ccount.csv | Date | Direct Copy | Relation | Cust_Count | Date |
| | Store | | | | Store |

| Source File Name | Source Data Field | Mapping | Staging Table Type | Staging Table Name | Staging Table Attribute |
|---|---|---|---|---|---|
| | CUSTCOUN | | | | Count_of_Customer |
| | Wine | | | | Wine_Sale |

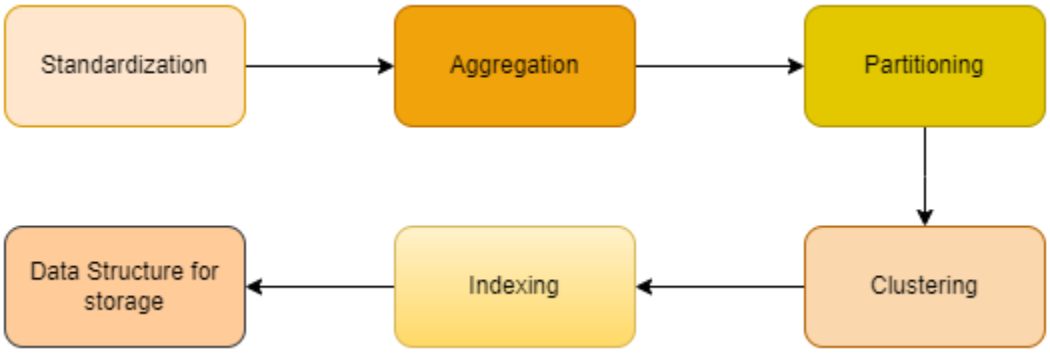
Mapping of fields from Staging tables to Data mart (Fact/Dimension) tables -

| Staging Table Name | Staging Table Attribute | Mapping | Data Mart Table Type | Data Mart Table Name | Data Mart Table Attribute |
|---|---|---|---|---|---|
| Movement | Store | Direct Copy | Fact Table | Fact_Sales | Store_Key |
| | Product_Code | Surrogate Key | | | UPC_Key |
| | Week | Surrogate Key | | | Date_Key |
| | Quantity | Used to calculate Sales = Move * Price / Quantity | | | Sales |
| | Price | | | | |
| | Move | | | | |
| | Profit | Direct Copy | | | Profit |
| | OK | Used to filter data, does not go into Fact table | | | N/A |
| Demographics | | Surrogate Key | Dimension Table | Dim_Store_Demo | Store_Key |
| | Store | Direct Copy | | | Store_Number |
| | City | Direct Copy | | | City |
| | Income | Direct Copy | | | Log_Income |
| Week_Decode | | Surrogate Key | Dimension Table | Dim_Time | Date_Key |
| | Week_No | Direct Copy | | | Week_Number |

| Staging Table Name | Staging Table Attribute | Mapping | Data Mart Table Type | Data Mart Table Name | Data Mart Table Attribute |
|---|---|---|---|---|---|
| | Start | Calculate Month number based on start and end date | | | Month |
| | End | Calculate year from start and end date | | | Year |
| Cust_Count | Date | Surrogate Key | Fact Table | Fact_Customer | Date_Key |
| | Store | Surrogate Key | | | Store_Key |
| | Count_of_Customer | Customer count aggregated at a weekly basis | | | Customer_Count |
| | Wine_Sale | Wine sales summed on a weekly basis | | | Wine_Sales |
| Movement | | Surrogate Key | Dimension Table | Dim_Product | UPC_Key |
| | Product_Code | Direct Copy | | | UPC_Number |
| | Product_Category | Direct Copy | | | Category |

## 3.8 Physical Design

In preparation for the future growth of your Data Mart for DFF , it is essential to develop a robust physical design plan that ensures efficient data management and scalability.




### Standardization

At the core of our data management strategy, we will proactively develop and enforce rigorous data standards. This includes the establishment of clear naming conventions for all data elements, ensuring that data is consistently labeled, and making it more intuitive for users to understand and query.

### Aggregation

We will put in place a data aggregation strategy that uses routine ETL (Extract, Transform, Load) procedures to compile and summarize data from diverse sources in order to meet the changing needs of DFF. This will speed up insights, improve query performance, and lower query complexity. For reporting and analytics purposes, aggregated data will be kept in specific tables so as not to overburden the system.

### Partitioning

Intelligent warehouse partitioning is an important aspect of developing the physical architectural plan for a data warehouse. Partitioning, which improves query efficiency, is just the partition of tables into smaller, easier-to-manage sections. The query needs to handle millions of rows of data in order to filter out the result, even though the amount of data to be obtained is not that large. For this reason, splitting data appropriately is essential.

### Clustering

There is a substantial correlation among several of the query results. In some cases, things are made easier by grouping related facts together. This will enable a more efficient data access procedure, which will enhance query performance. The general technique that allows us to achieve this is clustering.

## Indexing

We'll put in place a thorough indexing plan to maximize query performance and provide fast access to data. As part of this approach, relevant indexes on frequently searched columns will be created, using both clustered and non-clustered indexes as necessary. To guarantee peak performance even with increasing data volumes, routine index maintenance will be planned.

## Data Structure for Storage

After the stages have been successfully accomplished, it becomes imperative to make a thoughtful selection of the data storage structure. This step is pivotal for ensuring the efficient execution of data retrieval processes. Key questions that must be addressed include: What factors should be taken into account when deciding on temporary data storage solutions? How should we configure storage for the staging area? What considerations are pertinent when determining the storage approach for application data? These are among the critical queries that must be resolved in this phase.

# Section 4: Data Cleaning and Integration

## 4.1 Determine the target data

We have a total of 3 dimension tables and 2 fact tables as our target tables as we create the data warehouse. Given below are the table structures of each of these facts and dimensions.

| Target Table | Columns | Data Type |
|---|---|---|
| ISTM637_603_Gr2_DW<br>DIM_STORE_DEMO | Store_Key | Int |
| | Store_Number | Int |
| | City | Varchar(50) |
| | Log_Income | Int |

| Target Table | Columns | Data Type |
|---|---|---|
| ISTM637_603_Gr2_DW<br>DIM_PRODUCT | UPC_Key | Varchar(50) |
| | UPC_Numberr | Varchar(50) |
| | Category | Varchar(50) |

| Target Table | Columns | Data Type |
|---|---|---|
| ISTM637_603_Gr2_DW<br>DIM_TIME | Date_Key | Int |
| | Week_Number | Int |
| | Month_Number | Int |
| | Year_Number | Int |

| Target Table | Columns | Data Type |
|---|---|---|
| ISTM637_603_Gr2_DW<br>FACT_SALES | Store_Key | Int |
| | Date_Key | Int |
| | UPC_Key | Int |
| | Sales | Int |
| | Profit | Int |

| Target Table | Columns | Data Type |
|---|---|---|
| ISTM637_603_Gr2_DW<br><br>FACT_CUSTOMER | Store_Key | Int |
| | Date_Key | Int |
| | Customer_Count | Int |
| | Wine_Sales | Int |

## 4.2 Determine the Data Sources

All the data is retrieved from CSV files that are provided by DFF. Data from these CSV files were initially loaded into the staging area and then transformed and loaded into dimension and fact tables as needed. Given below is a list of all data sources used for this data warehouse -

| Data Source | Data Format | Description of data |
|---|---|---|
| CCount | Comma Separated Values (CSV) | The customer count file contains details regarding the number of visitors to the store and their purchases on a daily basis, specific to each store location |
| DEMO | Comma Separated Values (CSV) | The demographics file contains detailed information based on the US Government census. |
| Weeks Decode | PDF converted to CSV for loading purposes | This table stores the start and end date of each week's number that is being used in other tables in the system. |
| Movement | Comma Separated Values (CSV) | The movement files house sales data for each UPC within a specific category, organized by store and presented on a weekly basis. |
| UPC | Comma Separated Values (CSV) | The UPC files provide descriptions for every UPC within a given category. |

# 4.3 Mapping Tables in Excel

## Source to Staging Area

| Source File Name | Source Data Field | Mapping | Staging Table Type | Staging Table Name | Staging Table Attribute |
|---|---|---|---|---|---|
| DONE-XXXX.csv (Movement Tables) | STORE | Direct Copy | Relation | Movement | Store |
| | UPC | | | | Product_Code |
| | WEEK | | | | Week |
| | QTY | | | | Quantity |
| | PRICE | | | | Price |
| | MOVE | | | | Sale |
| | PROFIT | | | | Profit |
| | OK | | | | OK |
| | File Name (Beer/Grooming) | Use the file name to determine product category | | | Product_Category |
| Demo.csv (Demographic table) | STORE | Direct Copy | Relation | Demographics | Store |
| | CITY | | | | City |
| | INCOME | | | | Income |
| Week's Decode Table | Week # | Direct Copy | Relation | Week_Decode | Week_No |
| | Start | | | | Start |
| | End | | | | End |
| Ccount.csv | Date | Direct Copy | Relation | Cust_Count | Date |
| | Store | | | | Store |
| | CUSTCOUN | | | | Count_of_Customer |
| | Wine | | | | Wine_Sale |

## Staging to DW Mapping

| Staging Table Name | Staging Table Attribute | Mapping | Data Mart Table Type | Data Mart Table Name | Data Mart Table Attribute |
|---|---|---|---|---|---|
| Movement | Store | Direct Copy | Fact Table | Fact_Sales | Store_Key |
| | Product_Code | Surrogate Key | | | UPC_Key |
| | Week | Surrogate Key | | | Date_Key |
| | Quantity | Used to calculate Sales = Move * Price / Quantity | | | Sales |
| | Price | | | | |
| | Move | | | | |
| | Profit | Direct Copy | | | Profit |
| | OK | Used to filter data, does not go into Fact table | | | N/A |
| Demographics | | Surrogate Key | Dimension Table | Dim_Store_Demo | Store_Key |
| | Store | Direct Copy | | | Store_Number |
| | City | Direct Copy | | | City |
| | Income | Direct Copy | | | Log_Income |
| Week_Decode | | Surrogate Key | Dimension Table | Dim_Time | Date_Key |
| | Week_No | Direct Copy | | | Week_Number |
| | Start | Calculate Month number based on start and end date | | | Month |
| | End | Calculate year from start and end date | | | Year |
| Cust_Count | Date | Surrogate Key | Fact Table | Fact_Customer | Date_Key |
| | Store | Surrogate Key | | | Store_Key |
| | Count_of_Customer | Customer count aggregated at a weekly basis | | | Customer_Count |
| | Wine_Sale | Wine sales summed on a weekly basis | | | Wine_Sales |
| Movement | | Surrogate Key | Dimension Table | Dim_Product | UPC_Key |
| | Product_Code | Direct Copy | | | UPC_Number |
| | Product_Category | Direct Copy | | | Category |

# 4.4 Data Extraction Rules

The first and most important step in every data-related activity is data extraction. The formats in which the data are stored can vary when they are pulled from different sources. Excel, CSV files, and databases are common sources. The data used in this project is mostly in CSV format. Our objective is to transfer data from CSV files to the SQL server so that it can undergo various transformations.

When working with source data, we need to exercise caution because any modifications made during extraction may result in decreased performance. We cautiously handled any modifications to the data format, null values, and other aspects. The source contains OLTP data that contains the fine foods' everyday business transactions, necessitating a comprehension of data and metadata. Given below are a few rules we followed while extracting data -

- All fields were pulled into the staging area. Columns were omitted only during transformation.
- All files were converted to CSV files before loading into the staging area.
- Number of records were validated before and after loading.
- All data extraction was done using SSIS (Import/Export Wizard).
- Text qualifier of “ was removed using the Import/Export wherever applicable.

## 4.5 Data Transformation and Cleaning Rules

The data must then be transformed after it has been extracted. While loading can occasionally be completed prior to transformation, in our project, data transformation occurs after extraction and cleaning. The majority of the data needs to be cleaned; after this is done, the data is of higher quality and is free of errors. If cleaning is neglected, the data may still contain null values, redundancies, and other issues that eventually affect how our data is modeled and the conclusions we get from it. Even though they might not be in the source data, we might add pertinent additional columns or necessary information as we go along (like surrogate key columns).

Given below are a few rules followed while performing transformations on the extracted data -

- Removing NULL values: The first step to clean the data was to remove rows that had garbage values/nulls. Each table had a unique field that we used to filter out for nulls.
- All unnecessary columns were removed before any transformation was performed
- Data Type Conversion/ Casting: All numeric fields were converted/casted during transformation so that aggregations can be performed.
- Any temporary tables created were dropped once the transformation was done
- Derived columns were calculated in the staging area

## 4.6 Plan for aggregate tables

In our ETL implementation for the FACT_CUSTOMER table, we had to do some data aggregation to make sure that the target table is at the right granularity to answer our business questions. More particularly, the Wine_Sales and Customer_Count values were summed up to be aggregated at WEEK level from DAY level. None of the other tables required any aggregations.

## 4.7 Organization of data staging area

The data staging area is the space that acts as a temporary buffer between the data warehouse and the source. The staging area's function is to house data that is received from different sources (Excel, CSV, etc.) and utilized for subsequent procedures. From this staging area, the tables undergo a variety of ETL operations to form the dimension and fact tables. The staging area for our data warehouse had the following tables -

- Dbo.cust_count - This table contains data from the CCount table. All unnecessary records and columns were removed from the table within the staging area.
- Dbo.Demographics - This table contains data coming in from the demographics table. This is again the final table before using it to load fact and dimension tables. All transformations have been explained in section 6.
- Dbo.Movement - This table contains data unioned from both Grooming and Beer movement tables.

- Dbo.Week_decode - This table contains data directly imported from the week's decode table provided by DFF.

## 4.8 Procedure for data extraction and loading

1. Identify the precise information required to address our business inquiries.
2. Determine the data sources from which the necessary data can be sourced.
3. Fill the SQL server's staging area, ISTM637_603_Gr2_Staging, with the data that was loaded from the source.
4. For the purpose of initial cleaning, redundancy checking, and transformation, create temporary staging tables. Then, load the dated data into the final staging table. Once data is clean, remove all temporary tables from the staging area.
5. Use SSIS to load the data into the data mart's dimension tables. While loading the data, auto-increment the surrogate key values to generate the keys.
6. Once the dimensions are loaded, join these dimensions with the staging tables to be able to retrieve surrogate keys to populate the fact tables. Retrieve the measures from the staging tables, perform aggregations as needed.
7. Fact and dimension tables are now ready to use.

## 4.9 ETL for Dimension Table

All tables are first loaded into the staging area before transformations are performed. The clean data is then moved into the data warehouse to populate the dimension tables. Each row in the dimension table corresponds to a unique record. For example, dim_store_demo has one row for each store and has details about the demographics of the store that is needed for report generation. In the create table query for the dimension tables, an identity function is used to auto generate the surrogate keys that act as a unique identifier for each row in the table. Hence, for each new insert into the table, a new key value is generated which is auto-incremented based on the previous value.

For our data warehouse, we have the following dimension tables as mentioned in the mapping tables in earlier sections -

- Dim_Time
- Dim_store_demo
- Dim_Product

Here, dim_time is used to track each week within the data, dim_store_demo has information about the demographics of each store that is required within the report and dim_product stores the information about each product. It has the UPC number and the category it belongs to. This information in conjunction with the measures in fact tables will assist in creating dashboards and reports.

## 4.10 ETL for Fact Table

The ETL plan for fact tables is very similar to the plan for dimension tables. However, the way we populate fact tables is different. Fact tables generally track different measures for a given combination of dimensions. Hence, in order to store these measures, we first need to calculate them. This can be done during the transformation phase of ETL.

Once all dimension tables have been loaded, we use the surrogate keys from each of the dimension tables and add them into the fact table. The dimension tables are joined with the tables in the staging area and the keys retrieved from the dimension tables are used to populate the surrogate keys in the fact tables. The value for each measure is calculated in the staging area and this is then loaded into the fact tables using the same join conditions.

In our data warehouse, we have the following fact tables -

- Fact_Sales
- Fact_Customer

Fact_Sales tracks the sales of beer and grooming products on a weekly basis whereas fact_customer has customer count and wine sales for each store on a monthly basis.

## 4.11 Data Quality Issues

Even though we have all of the data from Dominick's Fine Foods dataset, there are still inconsistencies and the data does not adhere to quality requirements. Therefore, in order to transfer this data into a format that can be used in the data warehouse, cleaning and transformation must be done to it. We began by eliminating coupon fields and other fields that were not needed in our data warehouse implementation. Negative or null values were eliminated as well. The sales data was recovered into the fields that uniquely identify attributes and data types.

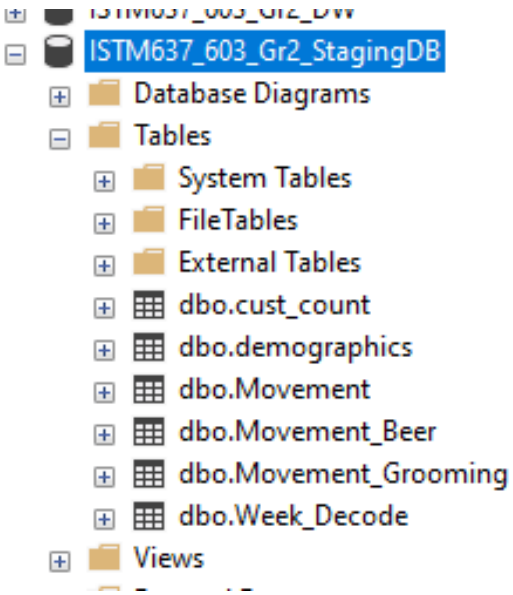
The dimension and fact tables are created using the following steps:

- Data is retrieved from the flat files (CSV, pdf) and moved to the staging area thus completing the extraction step
- The data is then transformed into formats suitable for the dimension and fact tables in the staging area
- The data is then mapped from the source staging area table to the the dimension and fact tables in the data warehouse

Given below are the results of implementing the ETL plans for all dimension and fact tables.

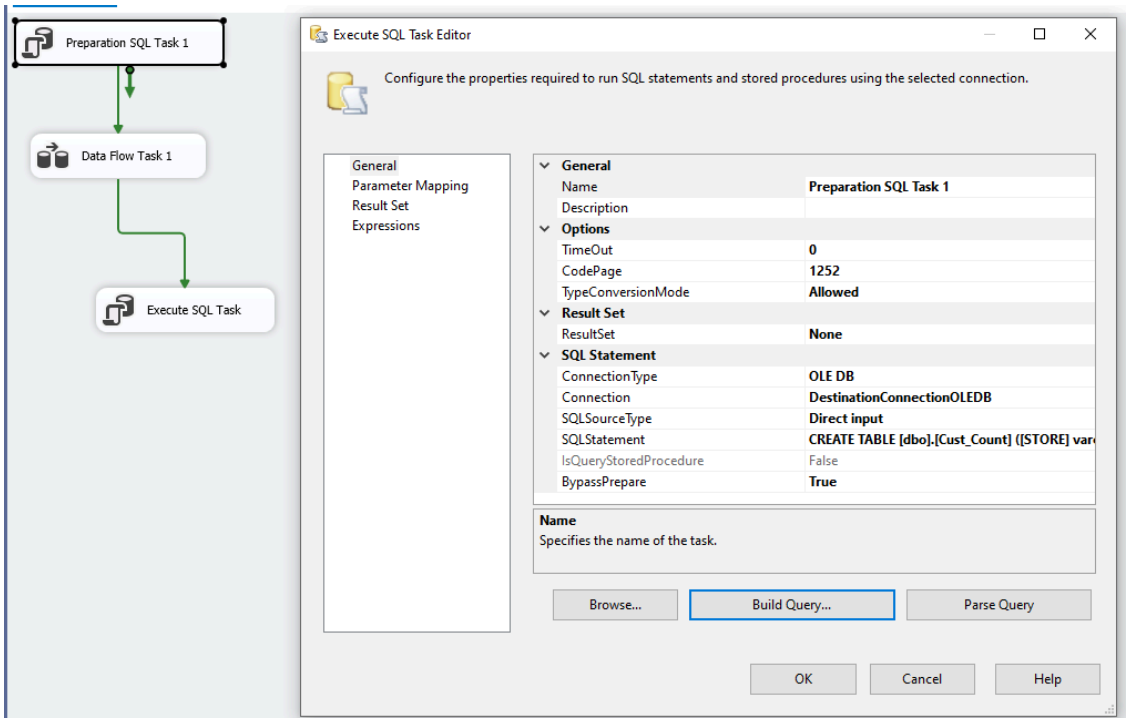
## 4.12 ETL for Staging Area

**Staging Area Tables in the Staging DB**

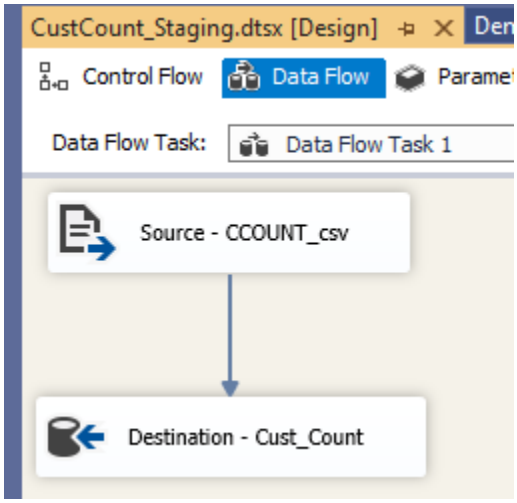



## ETL for CCOUNT to Staging table Cust_Count

We first created the cust_count temp table in the staging table in the ISTM_637_603_Gr2_StagingDB using an SSIS SQL task. We imported all the columns from CCOUNT_csv to the Cust_count staging table.

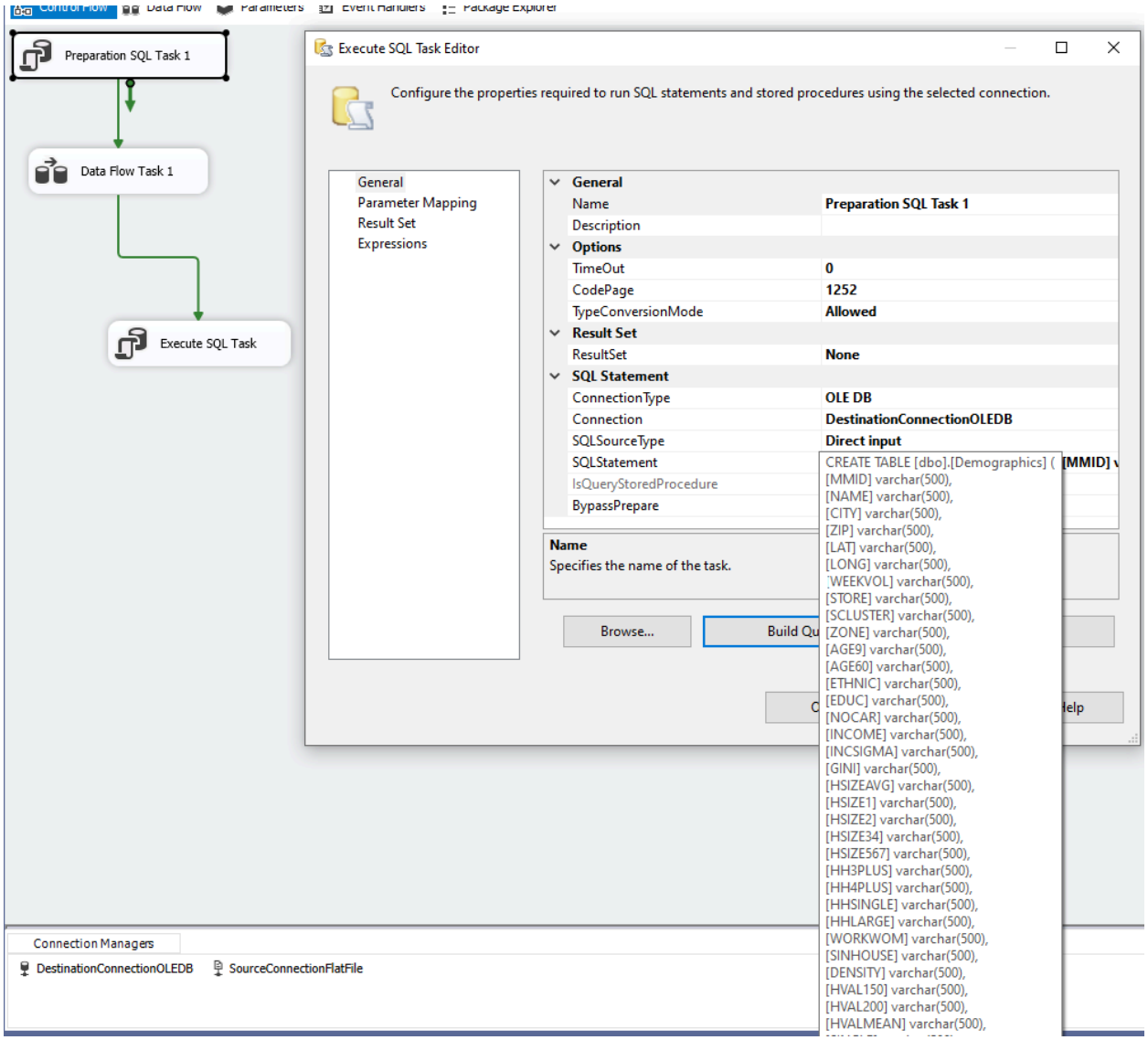


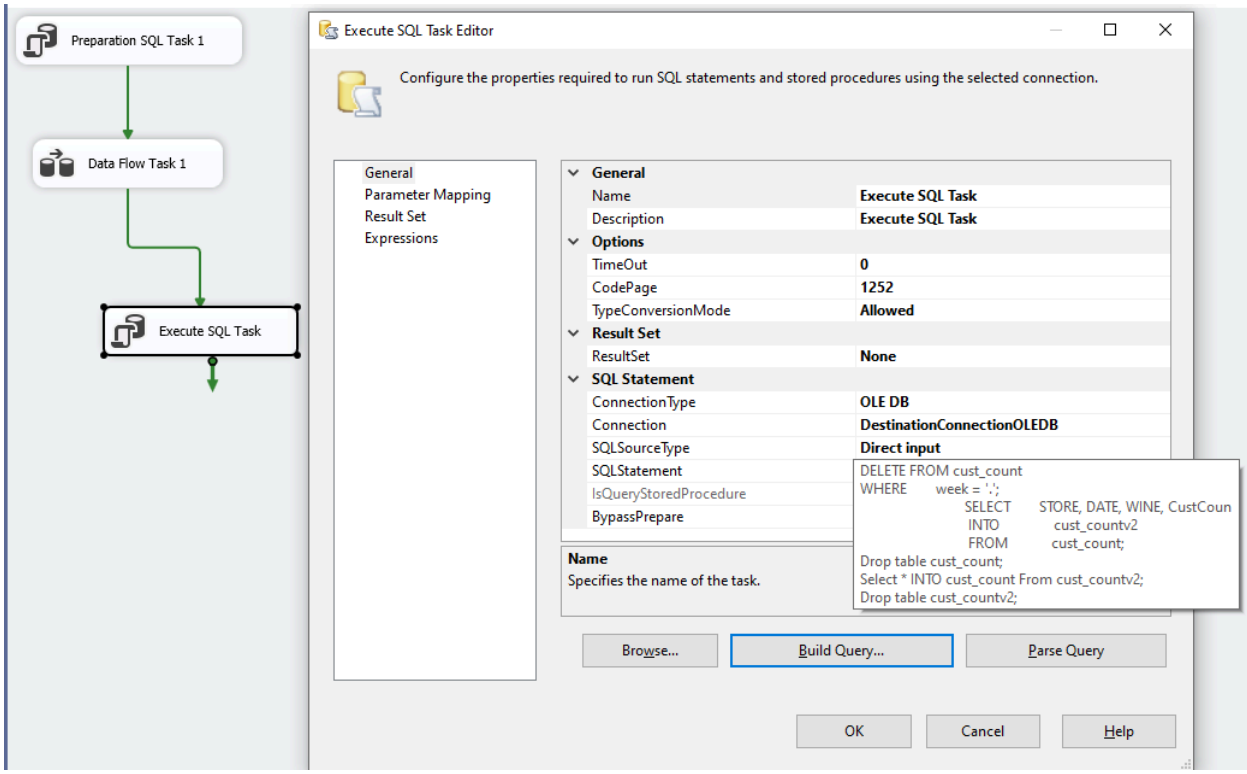
The data is extracted from the source flat file to the destination staging table as can be seen in the data flow diagram below. The source file is ccount_csv and destination is Cust_Count



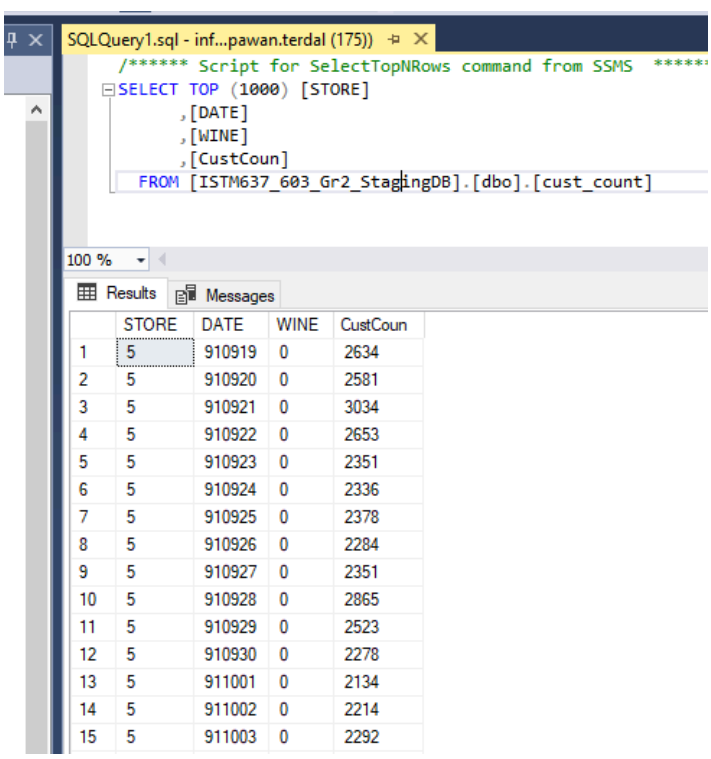

**Transformation**

We performed transformations while removing empty records from the temp table and selecting the required columns of store, data, wine and custcoun to be loaded into the Cust_Count table in the ISTM_637_603_Gr2_StagingDB and then dropped the temporary cust_countv2 table from the staging database.


Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Preparation SQL Task 1
Data Flow Task 1
Execute SQL Task
Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
General
Name
Preparation SQL Task 1
Description
Options
TimeOut
0
CodePage
1252
TypeConversionMode
Allowed
Result Set
ResultSet
None
SQL Statement
ConnectionType
OLE DB
Connection
DestinationConnectionOLEDB
SQLSourceType
Direct input
SQLStatement
IsQueryStoredProcedure
BypassPrepare
Name
Specifies the name of the task.
Browse...
Build Qu
Help
CREATE TABLE [dbo].[Demographics] (
[MMID] varchar(500),
[NAME] varchar(500),
[CITY] varchar(500),
[ZIP] varchar(500),
[LAT] varchar(500),
[LONG] varchar(500),
[WEEKVOL] varchar(500),
[STORE] varchar(500),
[SCLUSTER] varchar(500),
[ZONE] varchar(500),
[AGE9] varchar(500),
[AGE60] varchar(500),
[ETHNIC] varchar(500),
[EDUC] varchar(500),
[NOCAR] varchar(500),
[INCOME] varchar(500),
[INCSIGMA] varchar(500),
[GINI] varchar(500),
[HSIZEAVG] varchar(500),
[HSIZE1] varchar(500),
[HSIZE2] varchar(500),
[HSIZE34] varchar(500),
[HSIZE567] varchar(500),
[HH3PLUS] varchar(500),
[HH4PLUS] varchar(500),
[HHSINGLE] varchar(500),
[HHLARGE] varchar(500),
[WORKWOM] varchar(500),
[SINHOUSE] varchar(500),
[DENSITY] varchar(500),
[HVAL150] varchar(500),
[HVAL200] varchar(500),
[HVALMEAN] varchar(500),
Connection Managers
DestinationConnectionOLEDB
SourceConnectionFlatFile




## Cust_count table in the Staging Database

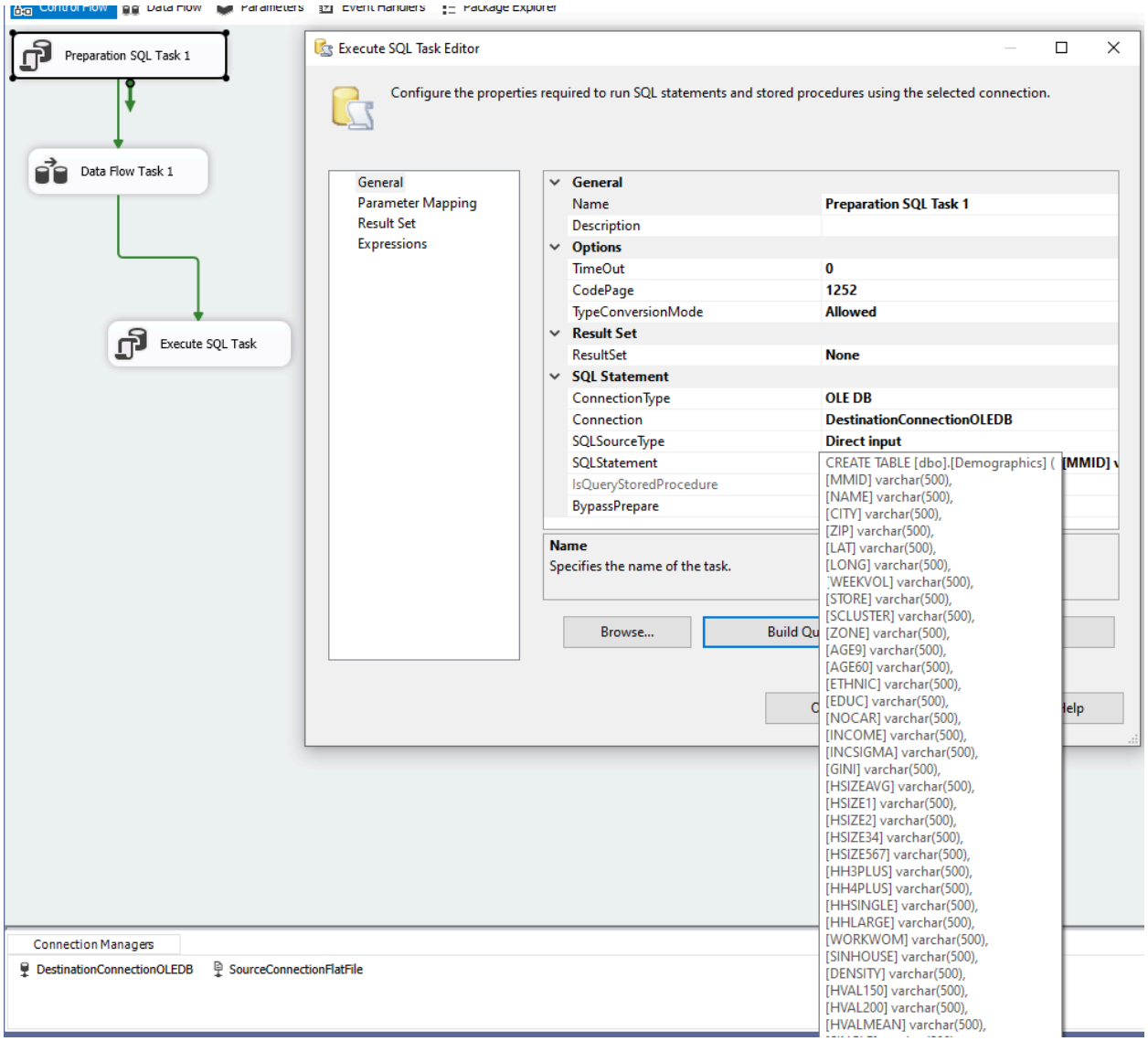



| | STORE | DATE | WINE | CustCoun |
|---|---|---|---|---|
| 1 | 5 | 910919 | 0 | 2634 |
| 2 | 5 | 910920 | 0 | 2581 |
| 3 | 5 | 910921 | 0 | 3034 |
| 4 | 5 | 910922 | 0 | 2653 |
| 5 | 5 | 910923 | 0 | 2351 |
| 6 | 5 | 910924 | 0 | 2336 |
| 7 | 5 | 910925 | 0 | 2378 |
| 8 | 5 | 910926 | 0 | 2284 |
| 9 | 5 | 910927 | 0 | 2351 |
| 10 | 5 | 910928 | 0 | 2865 |
| 11 | 5 | 910929 | 0 | 2523 |
| 12 | 5 | 910930 | 0 | 2278 |
| 13 | 5 | 911001 | 0 | 2134 |
| 14 | 5 | 911002 | 0 | 2214 |
| 15 | 5 | 911003 | 0 | 2292 |

## ETL for Demo.csv to Staging Area

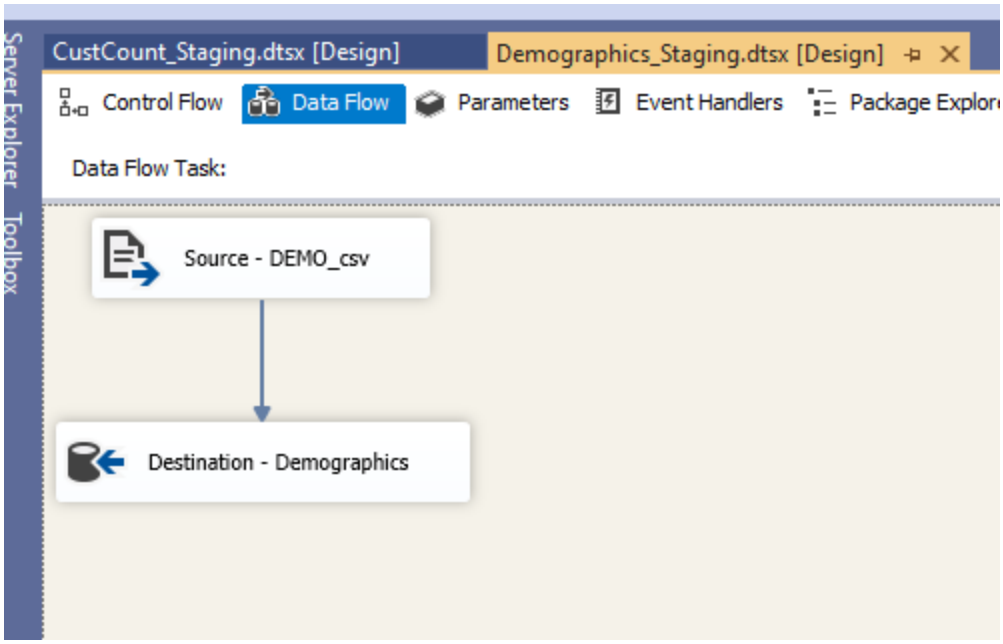
We moved the demo.csv file to the staging area in the ISTM_637_603_Gr2_StagingDB database on the SQL Server. For this we first created a temporary demographics table in the staging area and loaded the entire flat file into the temp table to perform transformations on it.

### Temp Table Creation



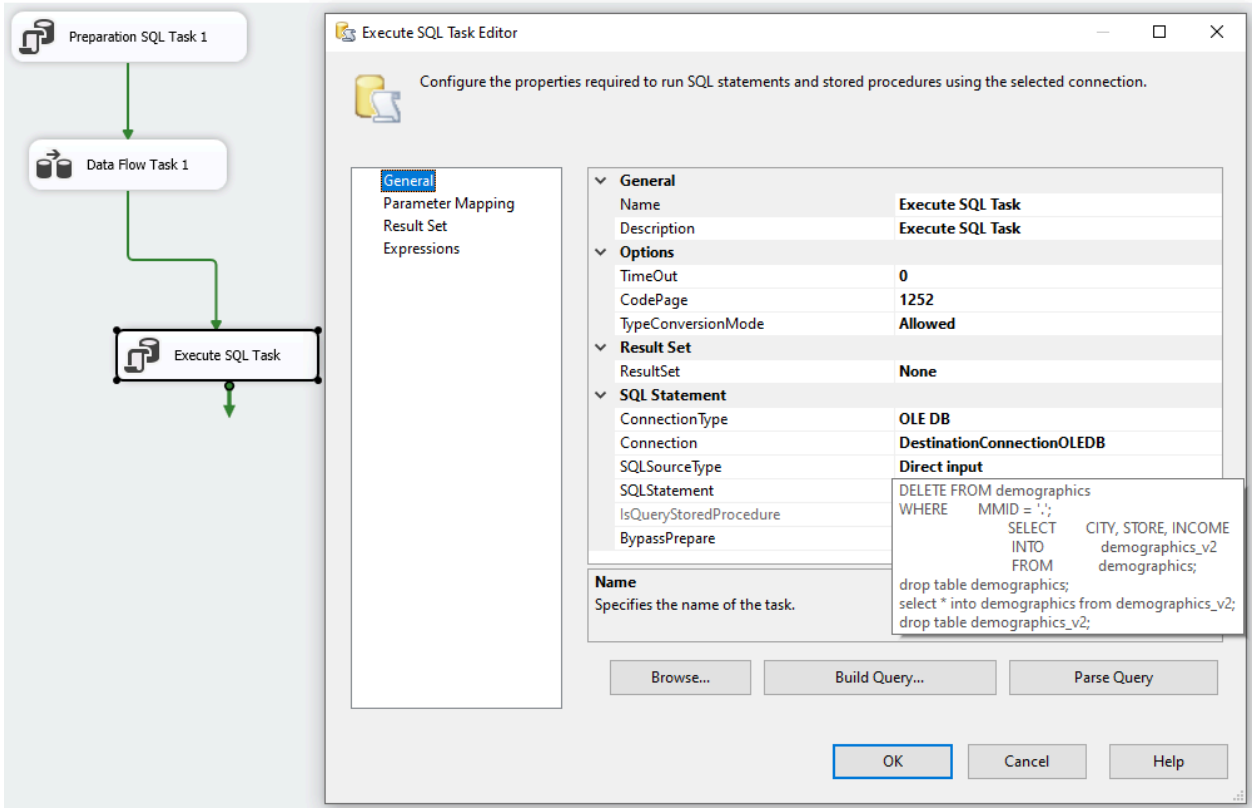

### Data Flow

The diagram below shows the flow of data from demo_csv to temporary demographics table in the staging area




**Transformation**

While loading the data into the demographics table we eliminated the blank records and selected only the required columns of city, store and income into the Demographics table in the staging area and dropped the temporary demographics_v2.



**Demographics Table in the Staging DB**

SQLQuery2.sql - inf...pawan.terdal (234)) | SQLQuery1.sql - inf...pawan.terdal (175)

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [CITY]
      ,[STORE]
      ,[INCOME]
  FROM [ISTM637_603_Gr2_StagingDB].[dbo].[demographics]
```

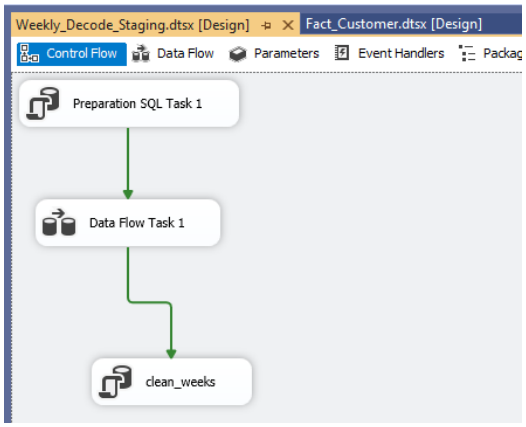
100 %

Results | Messages

| | CITY | STORE | INCOME |
|---|---|---|---|
| 1 | VERNON HILLS | 77 | 10.983120875 |
| 2 | DOWNERS GROVE | 78 | 10.959174943 |
| 3 | ARLINGTON HEIGHTS | 80 | 10.909509293 |
| 4 | MOUNT PROSPECT | 81 | 10.719935849 |
| 5 | CHICAGO | 111 | 10.138282957 |
| 6 | BUFFALO GROVE | 112 | 10.882659182 |
| 7 | CHICAGO | 113 | 10.647660492 |
| 8 | CALUMET CITY | 114 | 10.34794899 |
| 9 | NAPERVILLE | 115 | 10.947485248 |
| 10 | ELMHURST | 116 | 10.697113352 |
| 11 | SCHAUMBURG | 117 | 10.760962027 |

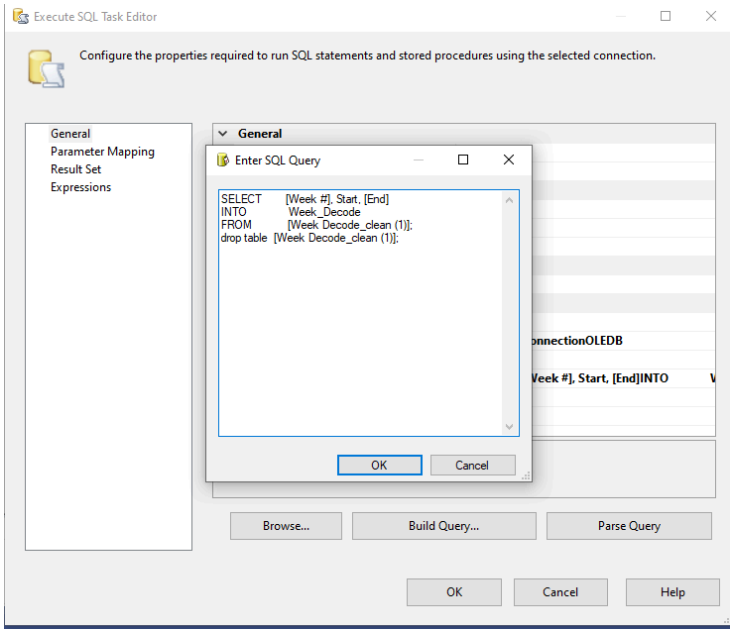
## ETL for Weekly_Decode to Staging

**Control Flow**

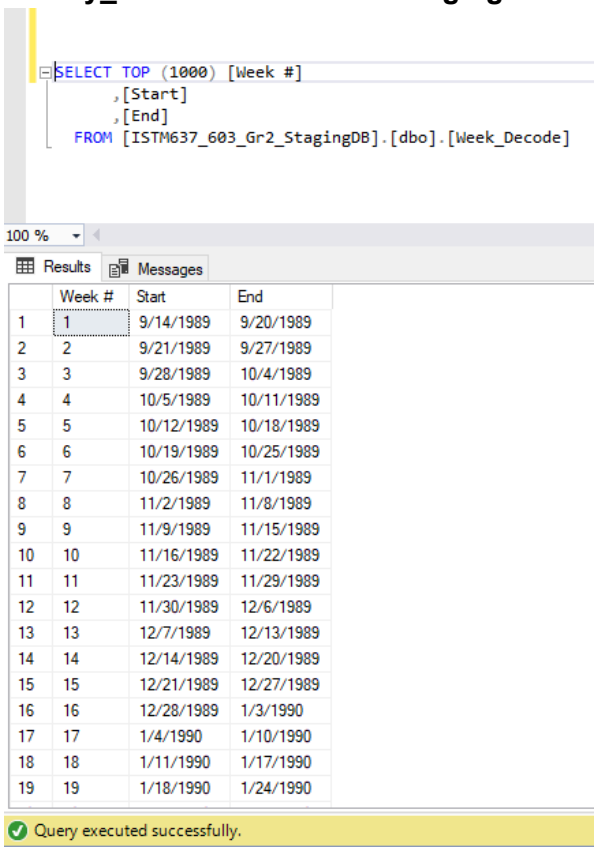



**Transformation**

While moving the data from the Weekly Decode csv file to the Week_Decode table in the staging area we only selected the week#, start and end columns into the week_decode table and drop the temporary week_decode_clean table.




## Weekly_Decode Table in the staging DB



```
SELECT TOP (1000) [Week #]
      ,[Start]
      ,[End]
  FROM [ISTM637_603_Gr2_StagingDB].[dbo].[Week_Decode]
```

100 %

Results | Messages

| | Week # | Start | End |
|---|---|---|---|
| 1 | 1 | 9/14/1989 | 9/20/1989 |
| 2 | 2 | 9/21/1989 | 9/27/1989 |
| 3 | 3 | 9/28/1989 | 10/4/1989 |
| 4 | 4 | 10/5/1989 | 10/11/1989 |
| 5 | 5 | 10/12/1989 | 10/18/1989 |
| 6 | 6 | 10/19/1989 | 10/25/1989 |
| 7 | 7 | 10/26/1989 | 11/1/1989 |
| 8 | 8 | 11/2/1989 | 11/8/1989 |
| 9 | 9 | 11/9/1989 | 11/15/1989 |
| 10 | 10 | 11/16/1989 | 11/22/1989 |
| 11 | 11 | 11/23/1989 | 11/29/1989 |
| 12 | 12 | 11/30/1989 | 12/6/1989 |
| 13 | 13 | 12/7/1989 | 12/13/1989 |
| 14 | 14 | 12/14/1989 | 12/20/1989 |
| 15 | 15 | 12/21/1989 | 12/27/1989 |
| 16 | 16 | 12/28/1989 | 1/3/1990 |
| 17 | 17 | 1/4/1990 | 1/10/1990 |
| 18 | 18 | 1/11/1990 | 1/17/1990 |
| 19 | 19 | 1/18/1990 | 1/24/1990 |

Query executed successfully.

## ETL for WGRO_csv and WBER_csv (Movement Files) to Staging

**Control Flow**

We created three separate data flows, one for moving the data from wgro_csv i.e movement file for grooming products, one for moving the data from wber_csv i.e movement file for beer and one for moving the data from wcig_csv i.e movement file for cigarettes into three separate temporary tables in the staging area.

These three temporary tables were then transformed into one common table Movement as shown in the subsequent steps.

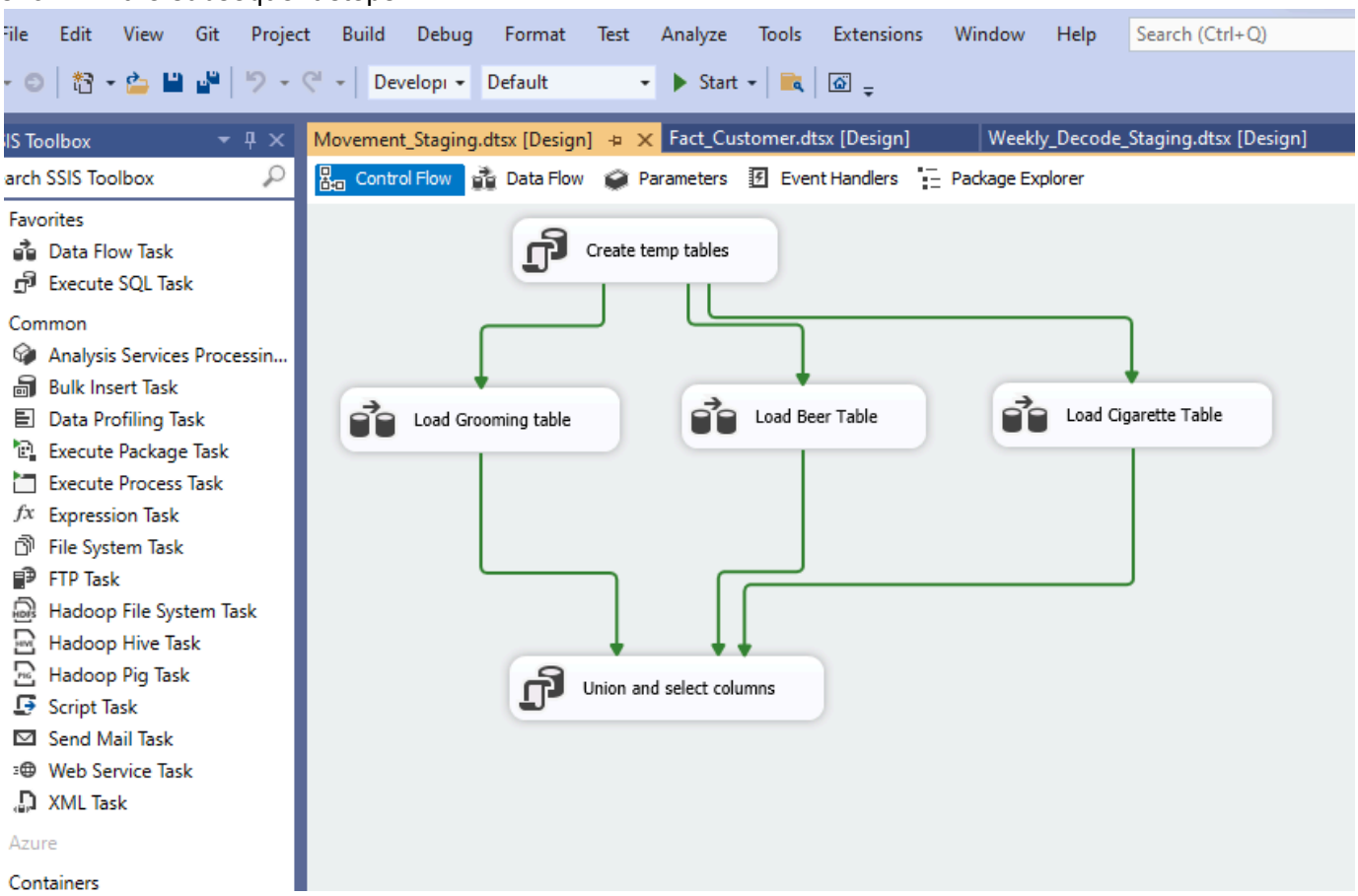
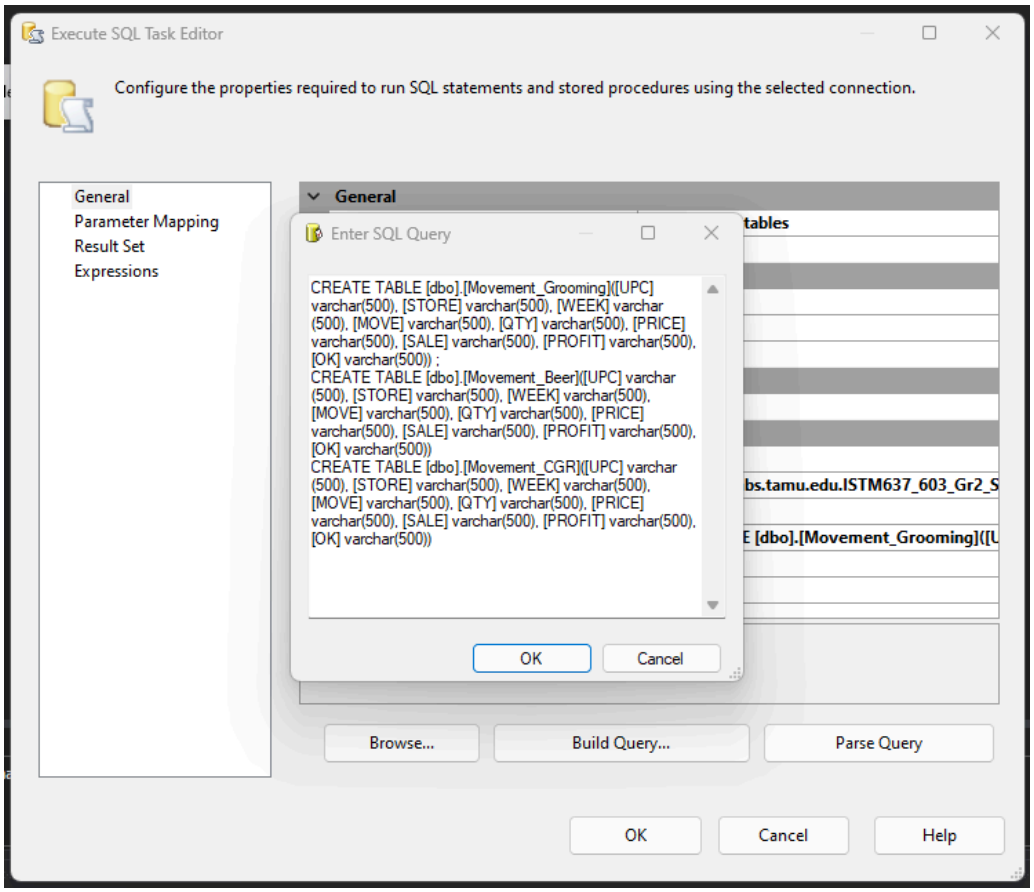
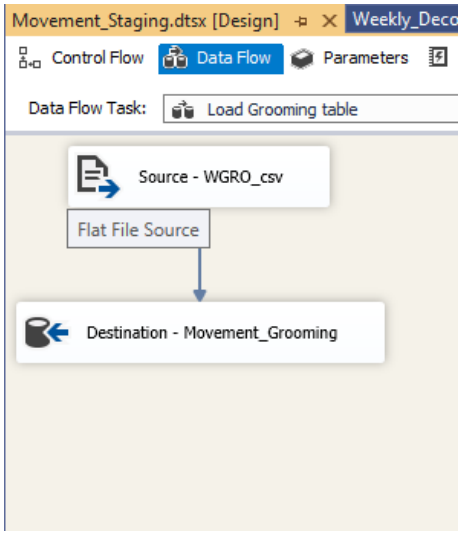


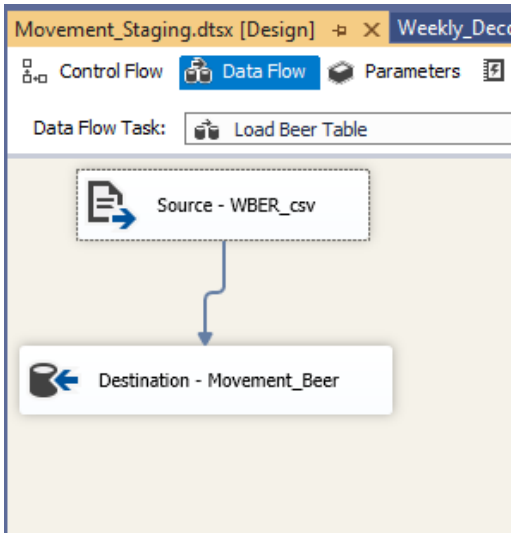

**Table Creation**

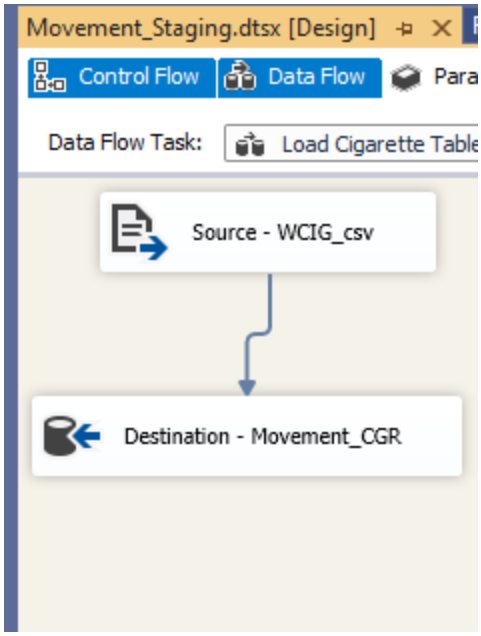
We created three tables - one for grooming products, one for beer and and the other for cigarettes and moved all of the columns from the flat files to the temporary tables.


Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
General
Enter SQL Query
CREATE TABLE [dbo].[Movement_Grooming]([UPC] varchar(500), [STORE] varchar(500), [WEEK] varchar (500), [MOVE] varchar(500), [QTY] varchar(500), [PRICE] varchar(500), [SALE] varchar(500), [PROFIT] varchar(500), [OK] varchar(500)) ;
CREATE TABLE [dbo].[Movement_Beer]([UPC] varchar (500), [STORE] varchar(500), [WEEK] varchar(500), [MOVE] varchar(500), [QTY] varchar(500), [PRICE] varchar(500), [SALE] varchar(500), [PROFIT] varchar(500), [OK] varchar(500))
CREATE TABLE [dbo].[Movement_CGR]([UPC] varchar (500), [STORE] varchar(500), [WEEK] varchar(500), [MOVE] varchar(500), [QTY] varchar(500), [PRICE] varchar(500), [SALE] varchar(500), [PROFIT] varchar(500), [OK] varchar(500))
OK
Cancel
Browse...
Build Query...
Parse Query
OK
Cancel
Help

## Data Flow:

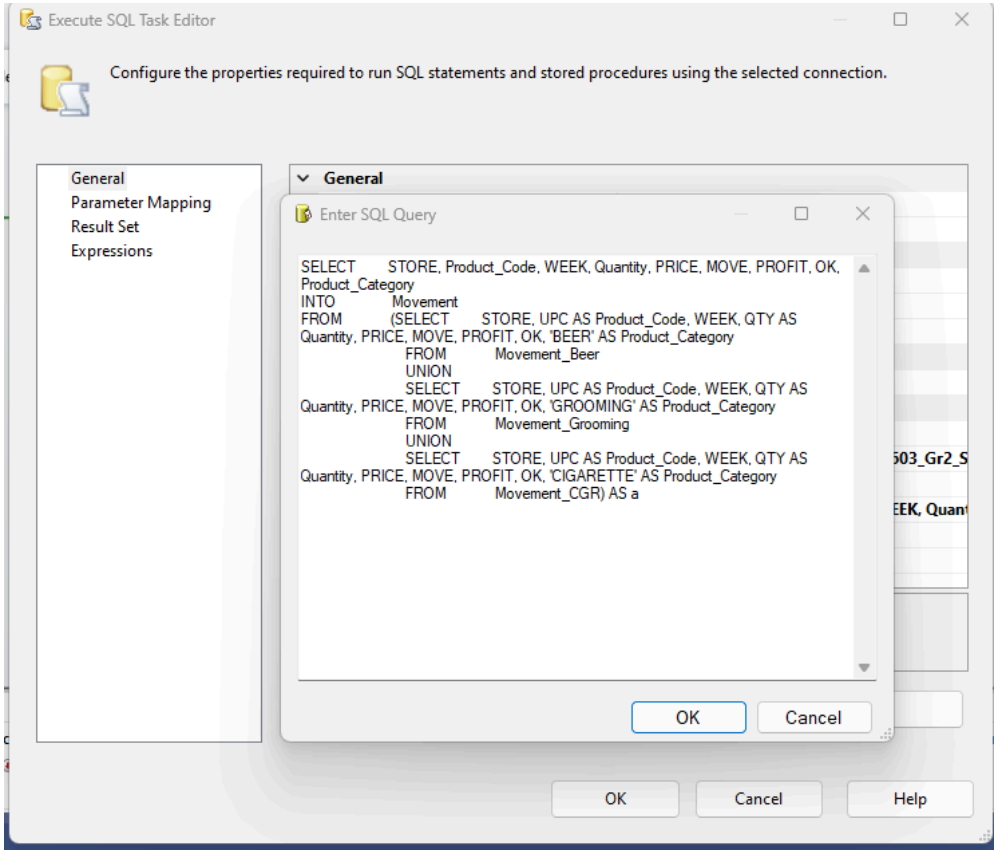

Movement_Staging.dtsx [Design]
Weekly_Deco
Control Flow
Data Flow
Parameters
Data Flow Task:
Load Grooming table
Source - WGRO_csv
Flat File Source
Destination - Movement_Grooming


Movement_Staging.dtsx [Design]
Weekly_Deco
Control Flow
Data Flow
Parameters
Data Flow Task:
Load Beer Table
Source - WBER_csv
Destination - Movement_Beer

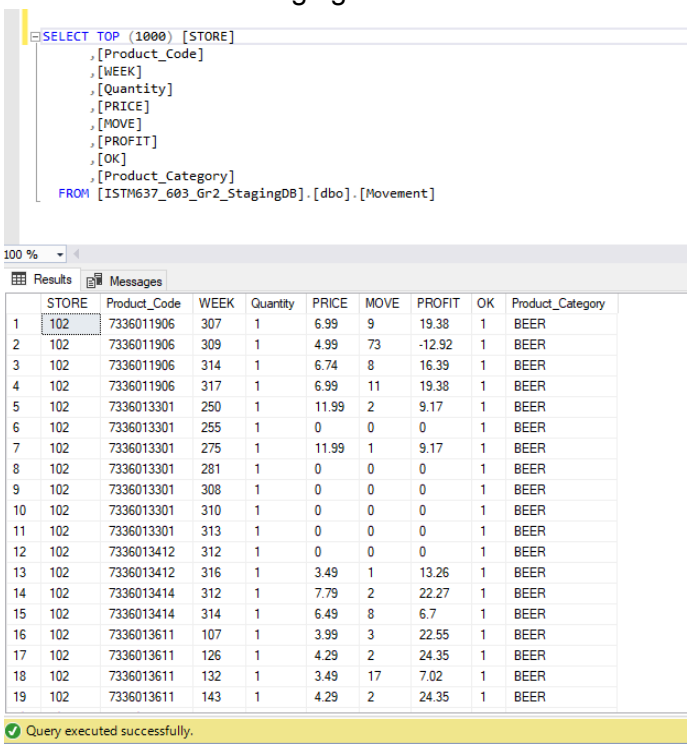



**Transformations**

We created a subquery with a union operator to merge the results from the three temporary tables for grooming, beer and cigarette products and while doing so we added a product category column to help us identify which record belonged to the beer, cigarette and grooming categories. Furthermore, we moved this subquery result into the Movement table in the staging area and dropped the temporary table.




Movement table in Staging DB



```
SELECT TOP (1000) [STORE]
      ,[Product_Code]
      ,[WEEK]
      ,[Quantity]
      ,[PRICE]
      ,[MOVE]
      ,[PROFIT]
      ,[OK]
      ,[Product_Category]
  FROM [ISTM637_603_Gr2_StagingDB].[dbo].[Movement]
```

100 %

Results | Messages

| | STORE | Product_Code | WEEK | Quantity | PRICE | MOVE | PROFIT | OK | Product_Category |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 102 | 7336011906 | 307 | 1 | 6.99 | 9 | 19.38 | 1 | BEER |
| 2 | 102 | 7336011906 | 309 | 1 | 4.99 | 73 | -12.92 | 1 | BEER |
| 3 | 102 | 7336011906 | 314 | 1 | 6.74 | 8 | 16.39 | 1 | BEER |
| 4 | 102 | 7336011906 | 317 | 1 | 6.99 | 11 | 19.38 | 1 | BEER |
| 5 | 102 | 7336013301 | 250 | 1 | 11.99 | 2 | 9.17 | 1 | BEER |
| 6 | 102 | 7336013301 | 255 | 1 | 0 | 0 | 0 | 1 | BEER |
| 7 | 102 | 7336013301 | 275 | 1 | 11.99 | 1 | 9.17 | 1 | BEER |
| 8 | 102 | 7336013301 | 281 | 1 | 0 | 0 | 0 | 1 | BEER |
| 9 | 102 | 7336013301 | 308 | 1 | 0 | 0 | 0 | 1 | BEER |
| 10 | 102 | 7336013301 | 310 | 1 | 0 | 0 | 0 | 1 | BEER |
| 11 | 102 | 7336013301 | 313 | 1 | 0 | 0 | 0 | 1 | BEER |
| 12 | 102 | 7336013412 | 312 | 1 | 0 | 0 | 0 | 1 | BEER |
| 13 | 102 | 7336013412 | 316 | 1 | 3.49 | 1 | 13.26 | 1 | BEER |
| 14 | 102 | 7336013414 | 312 | 1 | 7.79 | 2 | 22.27 | 1 | BEER |
| 15 | 102 | 7336013414 | 314 | 1 | 6.49 | 8 | 6.7 | 1 | BEER |
| 16 | 102 | 7336013611 | 107 | 1 | 3.99 | 3 | 22.55 | 1 | BEER |
| 17 | 102 | 7336013611 | 126 | 1 | 4.29 | 2 | 24.35 | 1 | BEER |
| 18 | 102 | 7336013611 | 132 | 1 | 3.49 | 17 | 7.02 | 1 | BEER |
| 19 | 102 | 7336013611 | 143 | 1 | 4.29 | 2 | 24.35 | 1 | BEER |

Query executed successfully.

## 4.13 List of Temporary Tables

The following are the temporary tables in the staging area:

- Cust_countv2
- Demographics_v2
- Week Decode_clean
- Movement_grooming
- Movement_Beer

All of the temporary tables were dropped from the staging area after the data was transformed, cleaned and finally loaded into final staging area tables.

## 4.14 Data Granularity at Independent Data Mart Level

For our implementation, we needed data mainly at two granular levels - week and month. The DIM_TIME dimension table stores the dates on a weekly basis. Each row in this table represents a week with a week number. The other granular level mentioned earlier is used in the FACT_SALES table where the sale amount of beer and grooming product is stored at a weekly level. Finally, in the FACT_CUSTOMER table, the wine sales is also aggregated at the same granular level. To answer one of the questions that asks for monthly sales of wine, we will have to aggregate the sales values from week level to month level.

100 %

Results | Messages

| | Week # | Start | End |
|---|---|---|---|
| 1 | 1 | 9/14/1989 | 9/20/1989 |
| 2 | 2 | 9/21/1989 | 9/27/1989 |
| 3 | 3 | 9/28/1989 | 10/4/1989 |
| 4 | 4 | 10/5/1989 | 10/11/1989 |
| 5 | 5 | 10/12/1989 | 10/18/1989 |
| 6 | 6 | 10/19/1989 | 10/25/1989 |
| 7 | 7 | 10/26/1989 | 11/1/1989 |
| 8 | 8 | 11/2/1989 | 11/8/1989 |
| 9 | 9 | 11/9/1989 | 11/15/1989 |
| 10 | 10 | 11/16/1989 | 11/22/1989 |
| 11 | 11 | 11/23/1989 | 11/29/1989 |
| 12 | 12 | 11/30/1989 | 12/6/1989 |
| 13 | 13 | 12/7/1989 | 12/13/1989 |
| 14 | 14 | 12/14/1989 | 12/20/1989 |
| 15 | 15 | 12/21/1989 | 12/27/1989 |
| 16 | 16 | 12/28/1989 | 1/3/1990 |
| 17 | 17 | 1/4/1990 | 1/10/1990 |
| 18 | 18 | 1/11/1990 | 1/17/1990 |
| 19 | 19 | 1/18/1990 | 1/24/1990 |

Query executed successfully.

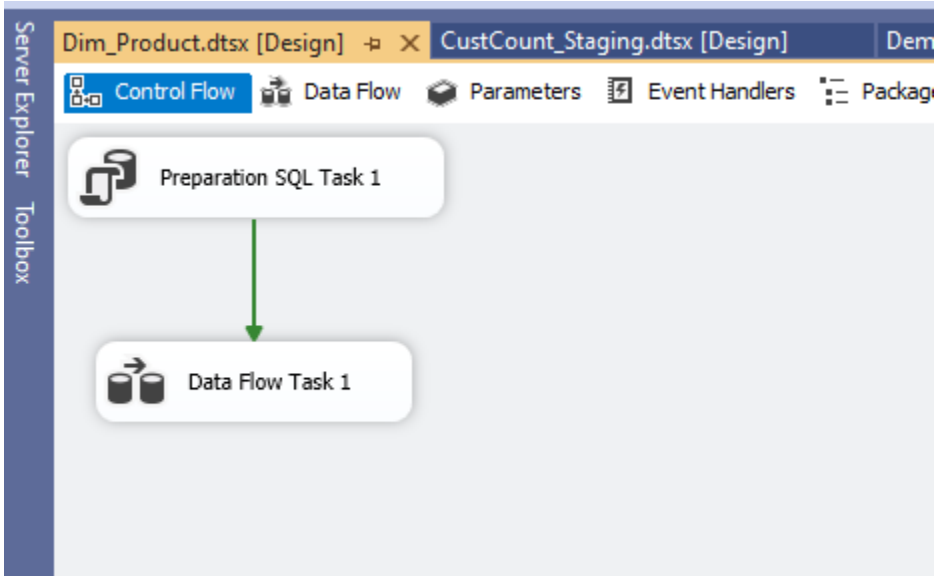
## 4.15 ETL for Dimension Tables

### DIM_PRODUCT

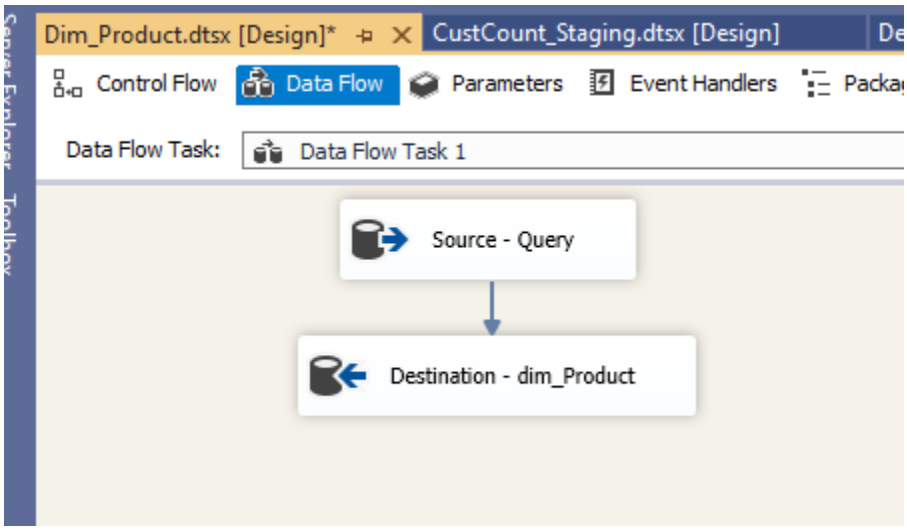
For the ETL implementation of the DIM_Product table, we use the following control flow and data flow in the SSIS tool. The mapping from staging table to dimension table was direct in this

case and no transformation was needed. This is because the staging table already had the transformed and cleaned data.

**Control Flow**




**Data Flow**




**Dim_Product Dimension table:**

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [UPC_Key]
      ,[UPC_Number]
      ,[Category]
  FROM [ISTM637_603_Gr2_DW].[dbo].[dim_Product]
```

100 %

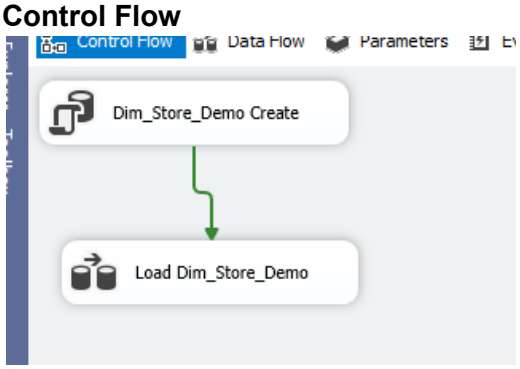
Results | Messages

| | UPC_Key | UPC_Number | Category |
|---|---|---|---|
| 1 | 1 | 1204400006 | GROOMING |
| 2 | 2 | 1204412800 | GROOMING |
| 3 | 3 | 1204430225 | GROOMING |
| 4 | 4 | 1204433670 | GROOMING |
| 5 | 5 | 1204437180 | GROOMING |
| 6 | 6 | 1204438000 | GROOMING |
| 7 | 7 | 1204438800 | GROOMING |
| 8 | 8 | 1254700354 | GROOMING |
| 9 | 9 | 1254700400 | GROOMING |
| 10 | 10 | 1254700510 | GROOMING |
| 11 | 11 | 1254700727 | GROOMING |
| 12 | 12 | 1254700852 | GROOMING |
| 13 | 13 | 1820000008 | BEER |
| 14 | 14 | 1820000031 | BEER |
| 15 | 15 | 1820000106 | BEER |

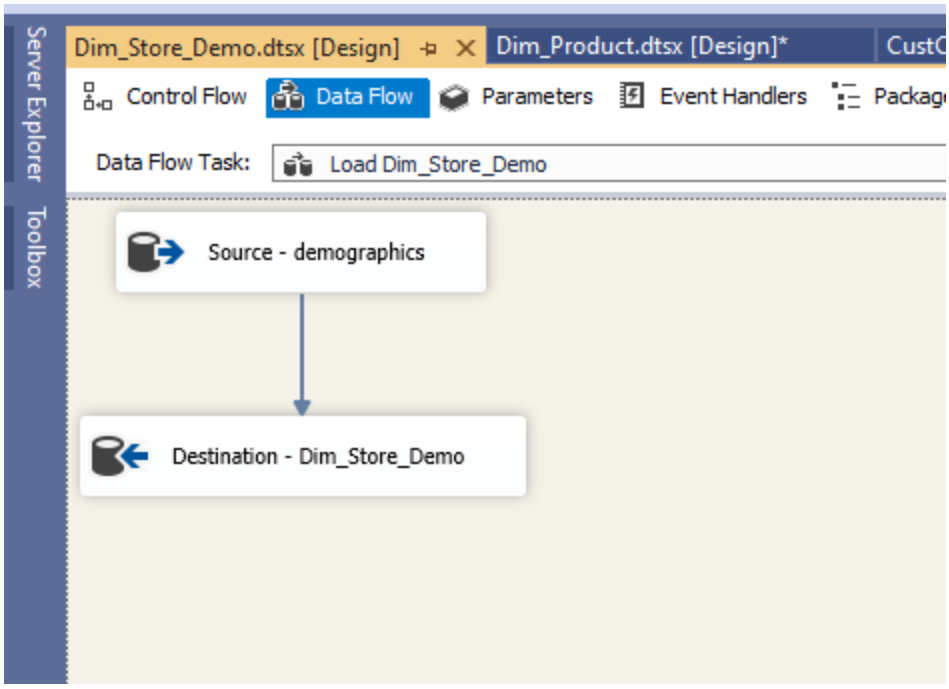
## DIM_STORE_DEMO

For the ETL implementation of the DIM_STORE_DEMO table, we use the following control flow and data flow in the SSIS tool. The mapping from staging table to dimension table was direct in this case and no transformation was needed. This is because the staging table already had the transformed and cleaned data.

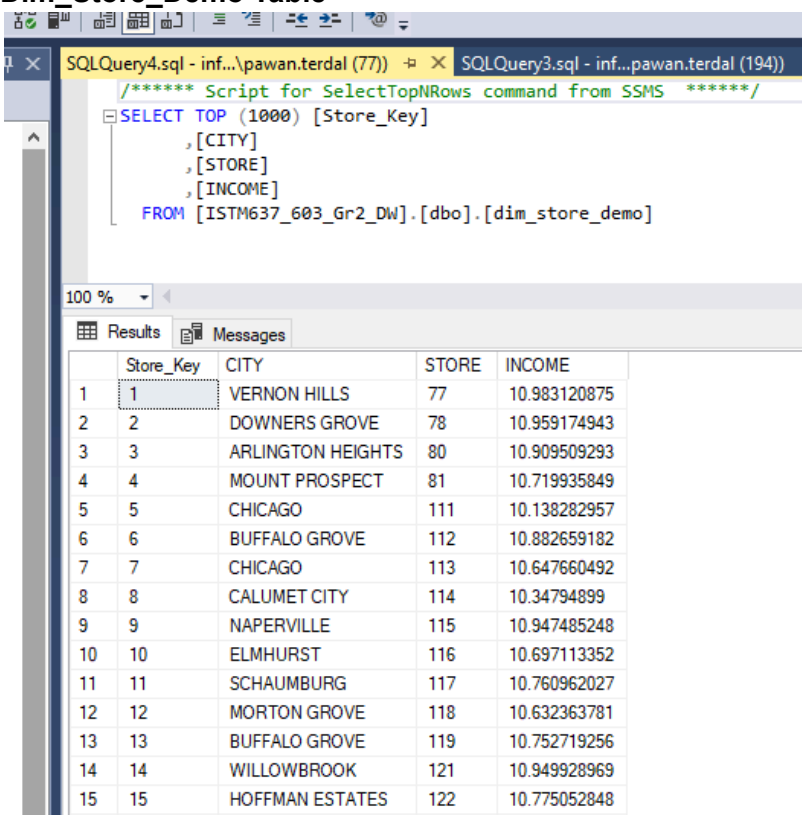
**Control Flow**




**Data Flow**




**Dim_Store_Demo Table**

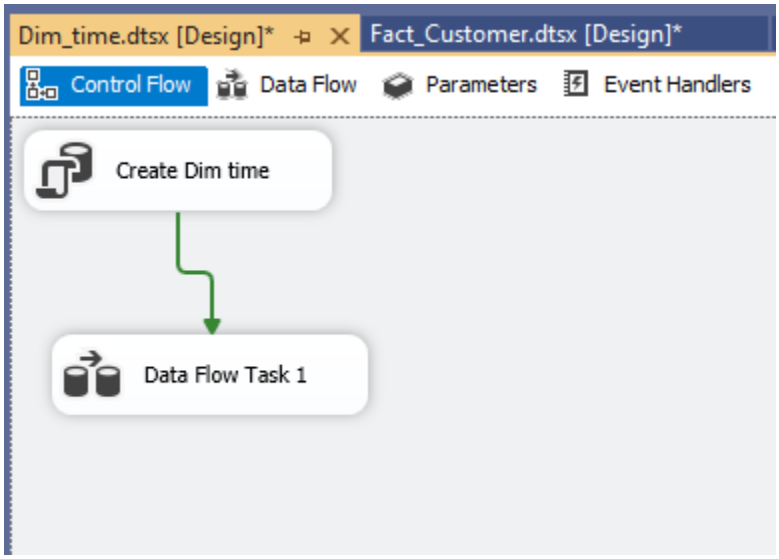



| | Store_Key | CITY | STORE | INCOME |
|---|---|---|---|---|
| 1 | 1 | VERNON HILLS | 77 | 10.983120875 |
| 2 | 2 | DOWNERS GROVE | 78 | 10.959174943 |
| 3 | 3 | ARLINGTON HEIGHTS | 80 | 10.909509293 |
| 4 | 4 | MOUNT PROSPECT | 81 | 10.719935849 |
| 5 | 5 | CHICAGO | 111 | 10.138282957 |
| 6 | 6 | BUFFALO GROVE | 112 | 10.882659182 |
| 7 | 7 | CHICAGO | 113 | 10.647660492 |
| 8 | 8 | CALUMET CITY | 114 | 10.34794899 |
| 9 | 9 | NAPERVILLE | 115 | 10.947485248 |
| 10 | 10 | ELMHURST | 116 | 10.697113352 |
| 11 | 11 | SCHAUMBURG | 117 | 10.760962027 |
| 12 | 12 | MORTON GROVE | 118 | 10.632363781 |
| 13 | 13 | BUFFALO GROVE | 119 | 10.752719256 |
| 14 | 14 | WILLOWBROOK | 121 | 10.949928969 |
| 15 | 15 | HOFFMAN ESTATES | 122 | 10.775052848 |

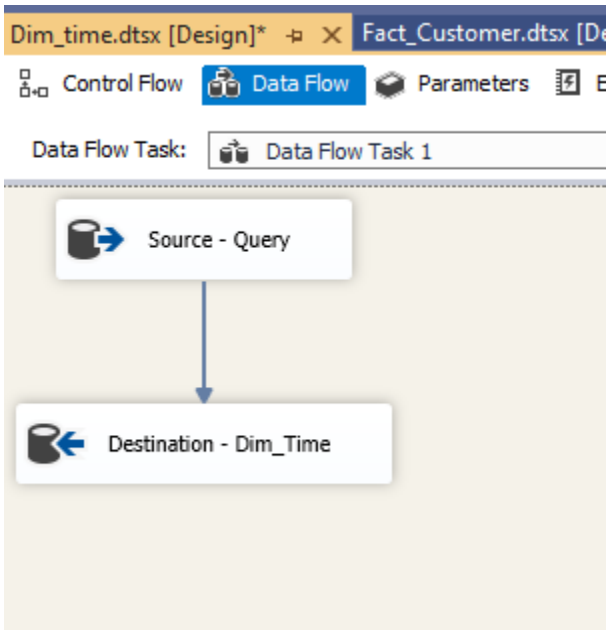
## DIM_TIME

For the ETL implementation of the DIM_TIME table, we use the following control flow and data flow in the SSIS tool. The mapping from staging table to dimension table involved some transformation before the data was put into the destination. This transformation was needed to make sure the WEEK, MONTH and YEAR numbers are extracted correctly from the date. The query for this extraction is shown below under the transformation section.

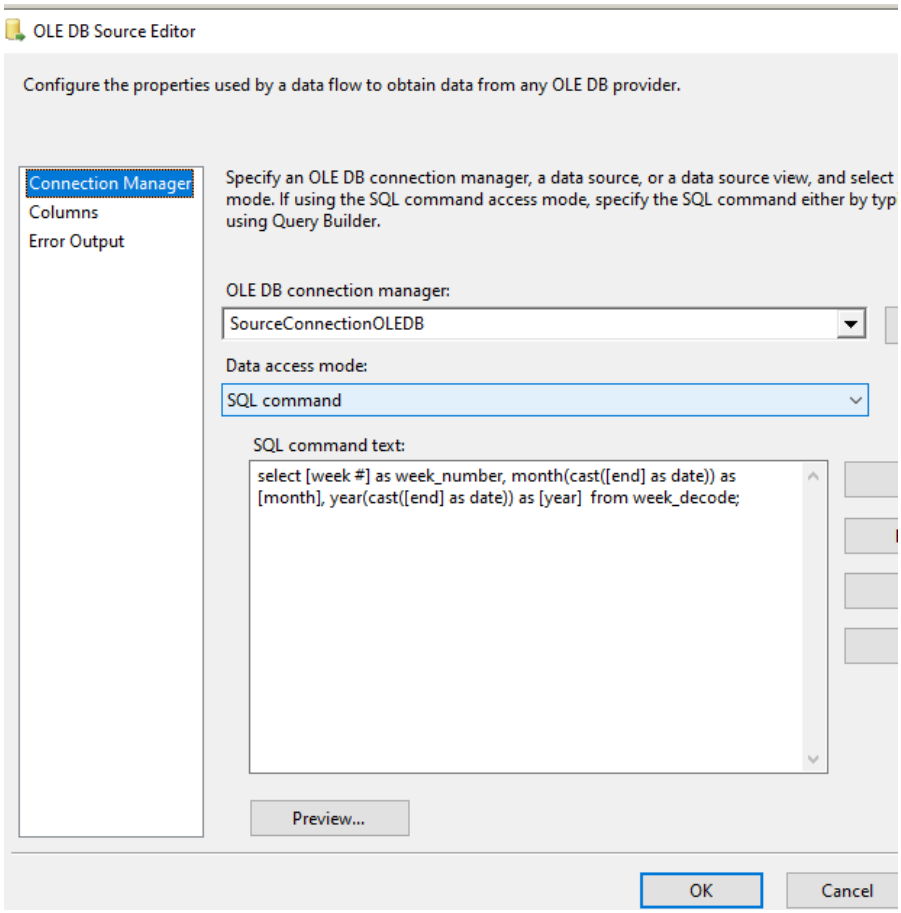
**Control Flow**



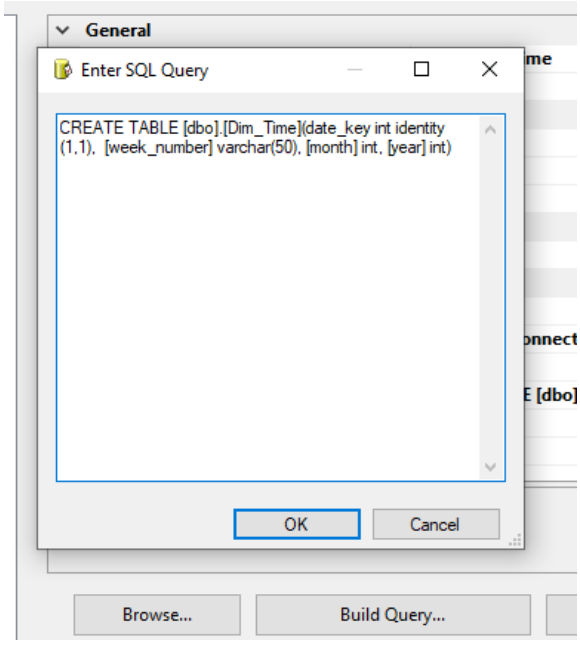

**Data Flow**




**Transformation**




## Dim_Time Table Creation

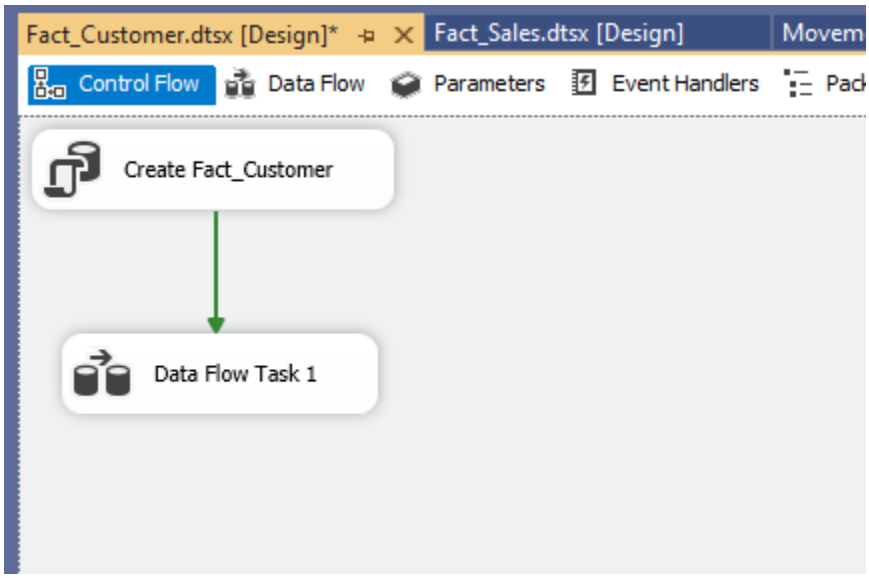


# 4.16 ETL for Fact Tables

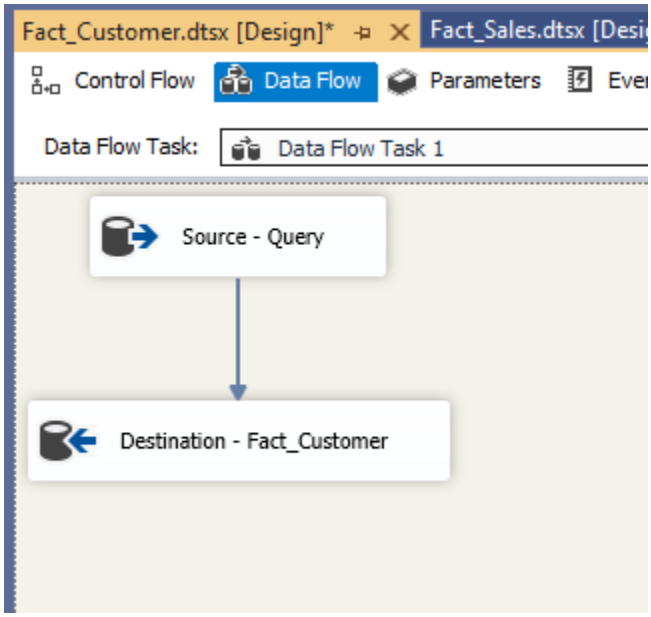
## FACT_CUSTOMER

**Control Flow**

Moving the data from tables in the staging area into the fact_customer table in the data warehouse.

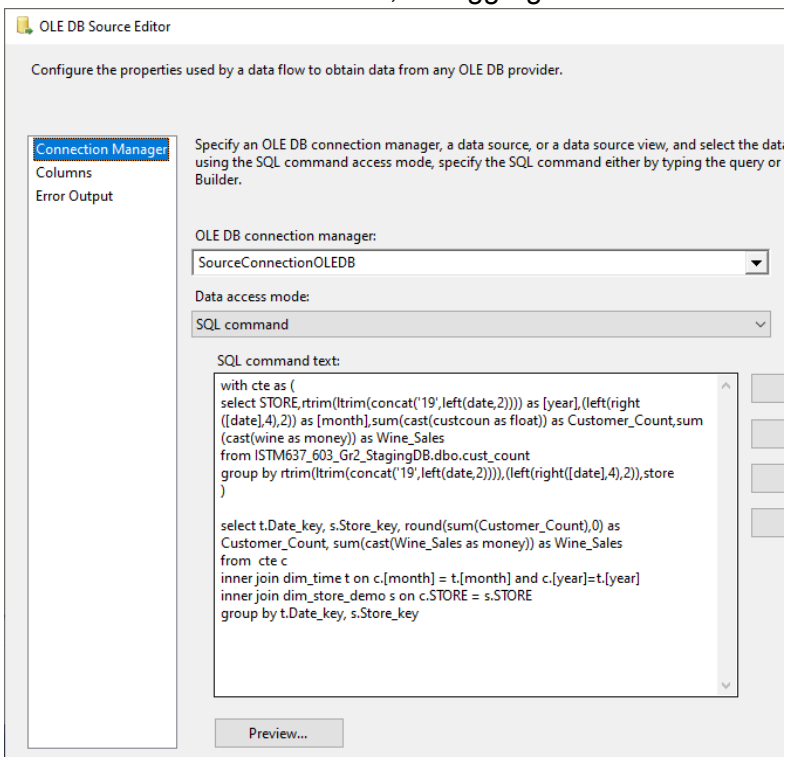


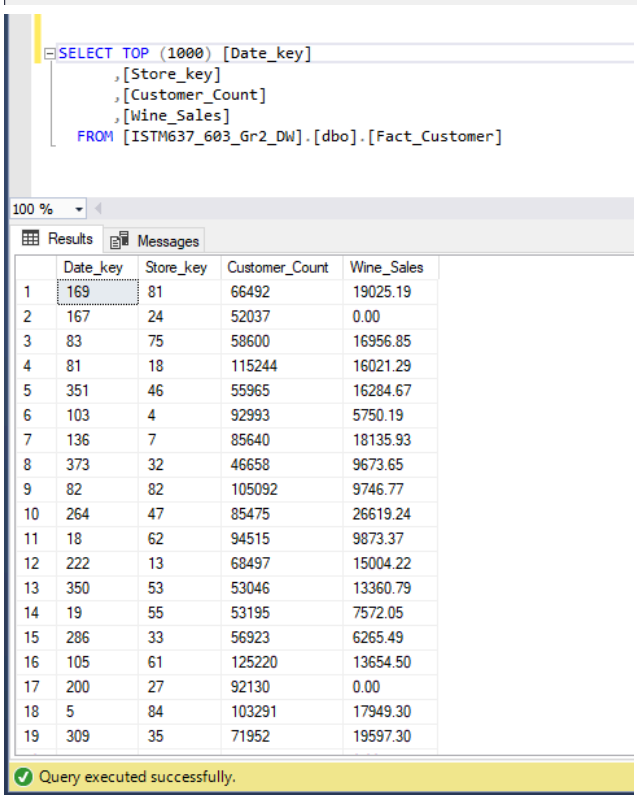

**Data Flow**




While moving the data from the staging area to the fact_customer table we performed transformations to make the columns appropriate for the columns in the fact table. For this, we

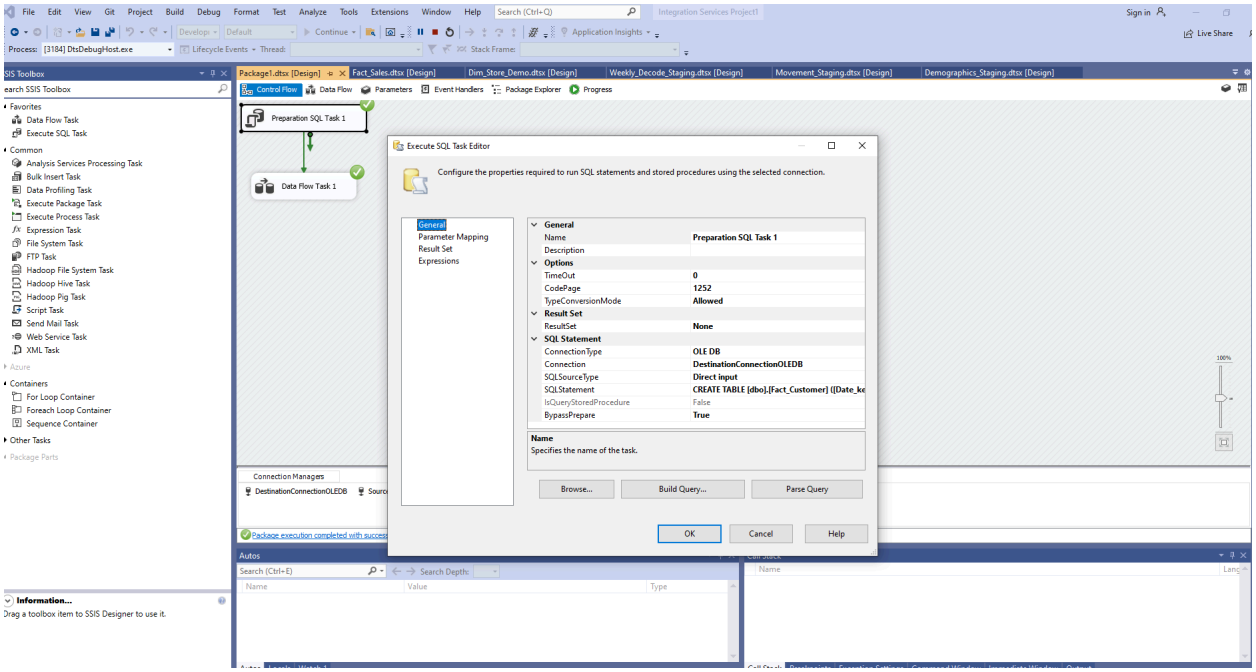
removed leading and trailing spaces from the date column and converted the data type for the custcoun column.Furthermore, we aggregated the wine sales on a monthly level.




| | Date_key | Store_key | Customer_Count | Wine_Sales |
|---|---|---|---|---|
| 1 | 169 | 81 | 66492 | 19025.19 |
| 2 | 167 | 24 | 52037 | 0.00 |
| 3 | 83 | 75 | 58600 | 16956.85 |
| 4 | 81 | 18 | 115244 | 16021.29 |
| 5 | 351 | 46 | 55965 | 16284.67 |
| 6 | 103 | 4 | 92993 | 5750.19 |
| 7 | 136 | 7 | 85640 | 18135.93 |
| 8 | 373 | 32 | 46658 | 9673.65 |
| 9 | 82 | 82 | 105092 | 9746.77 |
| 10 | 264 | 47 | 85475 | 26619.24 |
| 11 | 18 | 62 | 94515 | 9873.37 |
| 12 | 222 | 13 | 68497 | 15004.22 |
| 13 | 350 | 53 | 53046 | 13360.79 |
| 14 | 19 | 55 | 53195 | 7572.05 |
| 15 | 286 | 33 | 56923 | 6265.49 |
| 16 | 105 | 61 | 125220 | 13654.50 |
| 17 | 200 | 27 | 92130 | 0.00 |
| 18 | 5 | 84 | 103291 | 17949.30 |
| 19 | 309 | 35 | 71952 | 19597.30 |

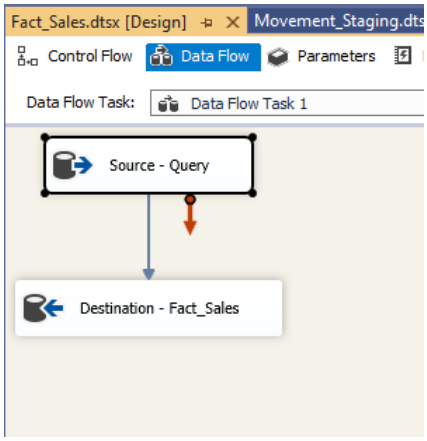
Control flow after Execution



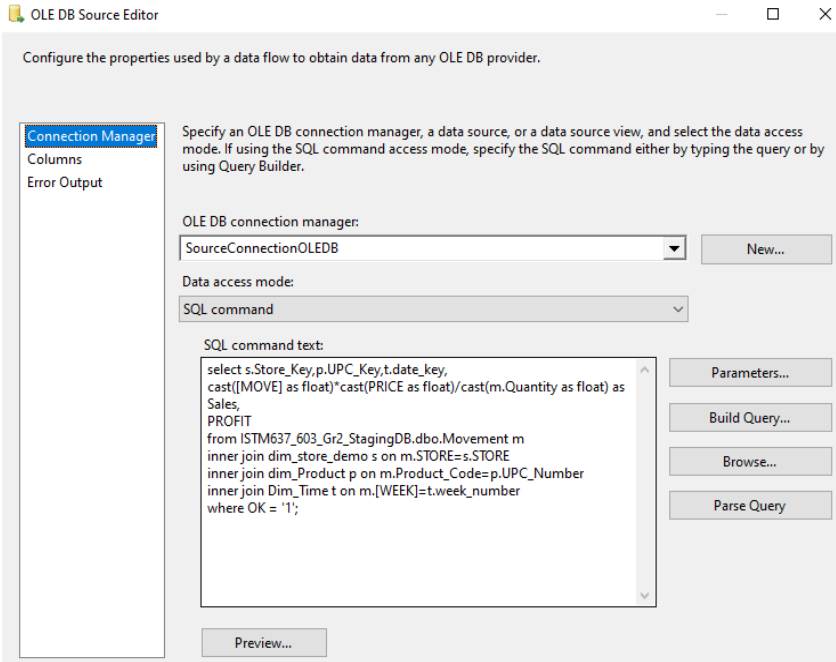

## FACT_SALES

While loading the data into the FACT_SALES table, we had to cast the sales data to reflect numerical value. The sales value was calculated on the fly while loading by using the formula sales = move*price/quantity. The different columns in these tables came from the respective dimension tables which were also used while joining using the surrogate keys.

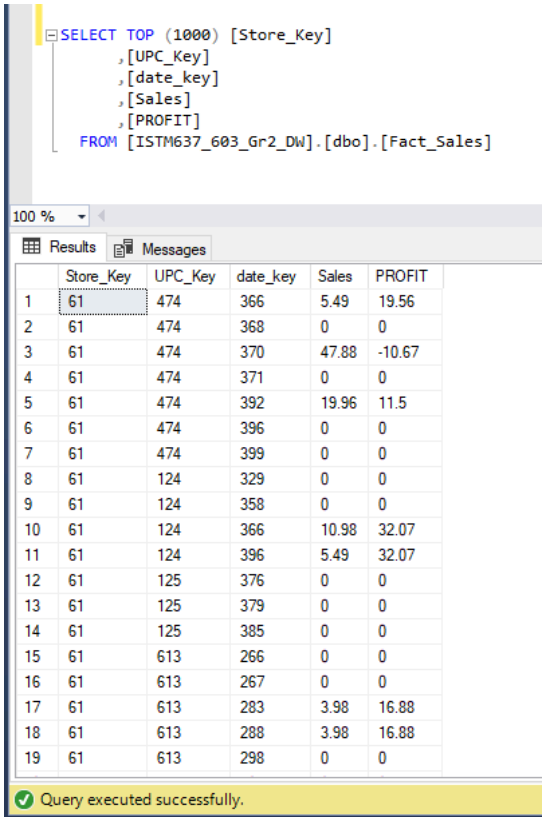
### Data Flow




### Transformation




## FACT_SALES Table



```
SELECT TOP (1000) [Store_Key]
      ,[UPC_Key]
      ,[date_key]
      ,[Sales]
      ,[PROFIT]
  FROM [ISTM637_603_Gr2_DW].[dbo].[Fact_Sales]
```

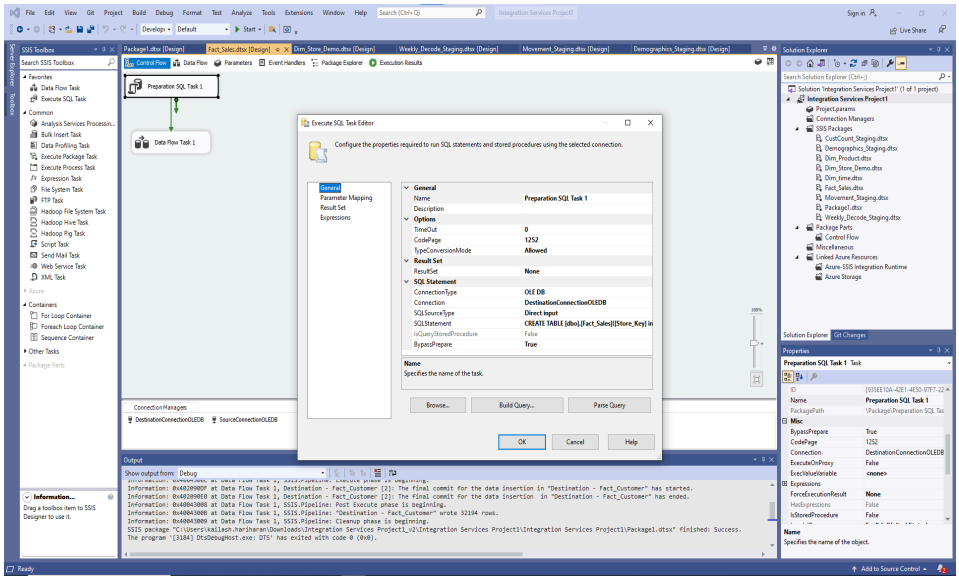
100 %

Results | Messages

| | Store_Key | UPC_Key | date_key | Sales | PROFIT |
|---|---|---|---|---|---|
| 1 | 61 | 474 | 366 | 5.49 | 19.56 |
| 2 | 61 | 474 | 368 | 0 | 0 |
| 3 | 61 | 474 | 370 | 47.88 | -10.67 |
| 4 | 61 | 474 | 371 | 0 | 0 |
| 5 | 61 | 474 | 392 | 19.96 | 11.5 |
| 6 | 61 | 474 | 396 | 0 | 0 |
| 7 | 61 | 474 | 399 | 0 | 0 |
| 8 | 61 | 124 | 329 | 0 | 0 |
| 9 | 61 | 124 | 358 | 0 | 0 |
| 10 | 61 | 124 | 366 | 10.98 | 32.07 |
| 11 | 61 | 124 | 396 | 5.49 | 32.07 |
| 12 | 61 | 125 | 376 | 0 | 0 |
| 13 | 61 | 125 | 379 | 0 | 0 |
| 14 | 61 | 125 | 385 | 0 | 0 |
| 15 | 61 | 613 | 266 | 0 | 0 |
| 16 | 61 | 613 | 267 | 0 | 0 |
| 17 | 61 | 613 | 283 | 3.98 | 16.88 |
| 18 | 61 | 613 | 288 | 3.98 | 16.88 |
| 19 | 61 | 613 | 298 | 0 | 0 |

Query executed successfully.

## Control Flow




## 4.17 Queries used in the process

Below are some of the important queries that we used during the process of ETL.

```
DELETE FROM demographics
WHERE       MMID = '.';

SELECT      CITY, STORE, INCOME
INTO         demographics_v2
FROM        demographics;

DROP TABLE demographics;

SELECT      *
INTO         demographics
FROM        demographics_v2; DROP TABLE demographics_v2;


DELETE FROM cust_count
WHERE       week = '.';

SELECT      STORE, DATE, WINE, CustCoun
INTO         cust_countv2
FROM        cust_count; DROP TABLE cust_count;

SELECT      *
INTO         cust_count
FROM        cust_countv2; DROP TABLE cust_countv2;
```

```
SELECT      [Week #], Start, [End]
INTO        Week_Decode
FROM        [Week Decode_clean (1)]; DROP TABLE [Week Decode_clean (1)];

CREATE TABLE [dbo].[dim_store_demo](Store_Key int IDENTITY (1, 1), [CITY] varchar(500),
[STORE] varchar(500), [INCOME] varchar(500));

with cte as (
select STORE,rtrim(ltrim(concat('19',left(date,2)))) as [year],(left(right([date],4),2)) as
[month],sum(cast(custcoun as float)) as Customer_Count,sum(cast(wine as money)) as
Wine_Sales
from ISTM637_603_Gr2_StagingDB.dbo.cust_count
group by rtrim(ltrim(concat('19',left(date,2)))),(left(right([date],4),2)),store
)
select t.Date_key, s.Store_key, round(sum(Customer_Count),0) as Customer_Count,
sum(cast(Wine_Sales as money)) as Wine_Sales
from  cte c
inner join dim_time t on c.[month] = t.[month] and c.[year]=t.[year]
inner join dim_store_demo s on c.STORE = s.STORE
group by t.Date_key, s.Store_key

select s.Store_Key,p.UPC_Key,t.date_key,
cast([MOVE] as float)*cast(PRICE as float)/cast(m.Quantity as float) as Sales,
PROFIT
from ISTM637_603_Gr2_StagingDB.dbo.Movement m
inner join dim_store_demo s on m.STORE=s.STORE
inner join dim_Product p on m.Product_Code=p.UPC_Number
inner join Dim_Time t on m.[WEEK]=t.week_number
where OK = '1'

SELECT DISTINCT Product_Code AS UPC_Number, Product_Category AS Category
FROM        Movement;

CREATE TABLE [dbo].[dim_store_demo](Store_Key int IDENTITY (1, 1), [CITY] varchar(500),
[STORE] varchar(500), [INCOME] varchar(500));
```

# Section 5: BI Reporting

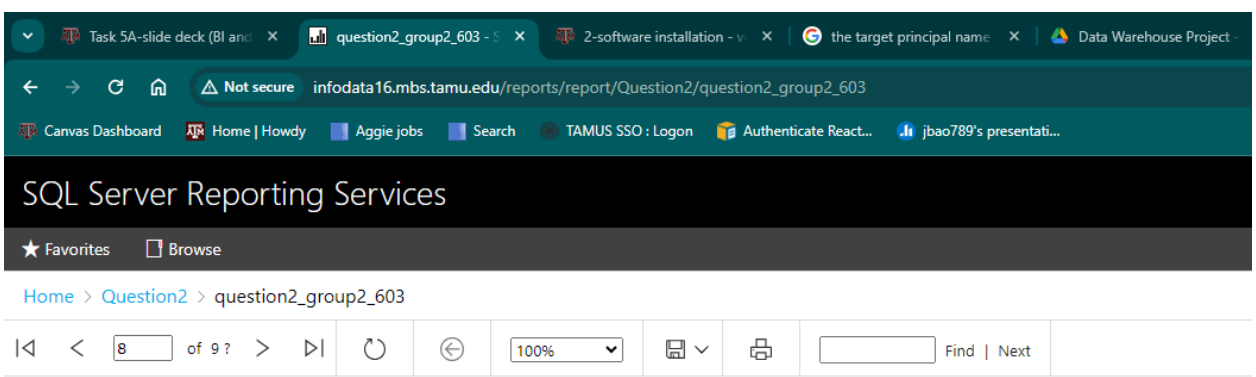
## 5.1 Reporting Plan

| Business Questions | Reporting Tool |
|---|---|
| 1 | SSRS |
| 2 | Power BI |
| 3 | Power BI |
| 4 | SSAS+SSRS |
| 5 | SSAS |

### 5.1.1 Determine all the target reports that satisfy the business questions and why

**BQ1: What does total footfall for Store X look like throughout a particular year?**

**SSRS**



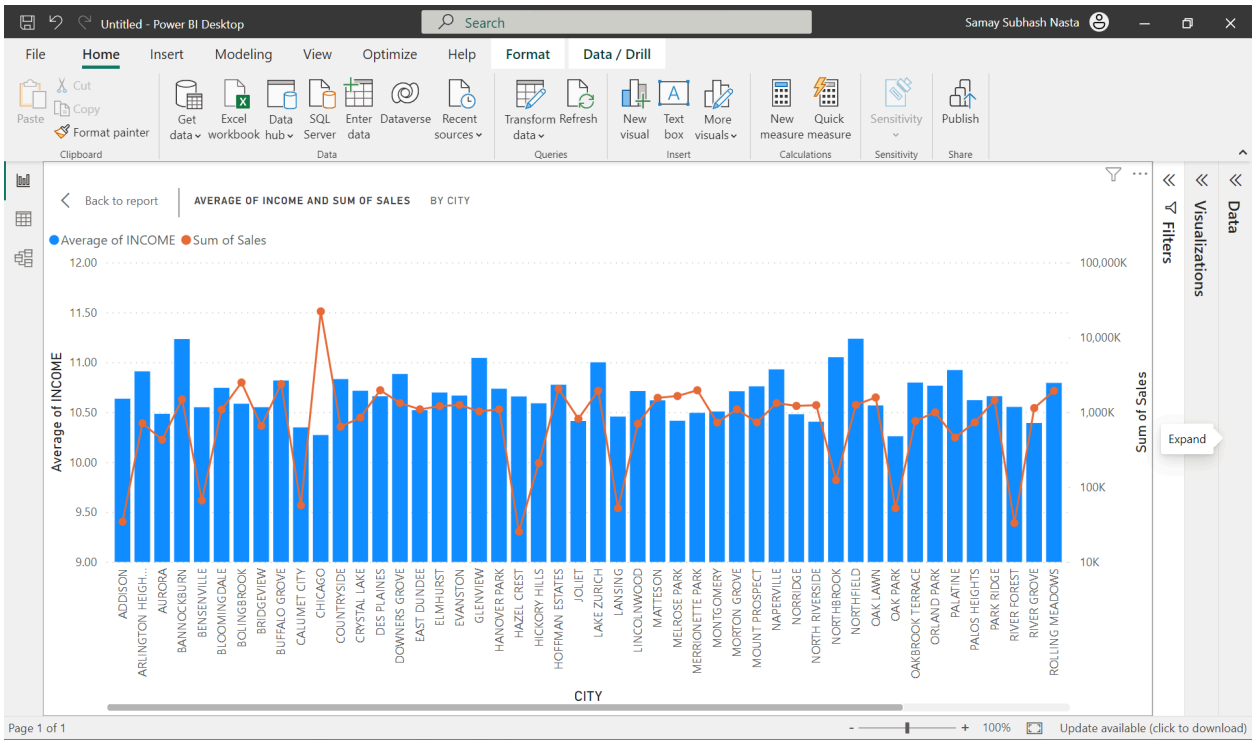

107

| | 1989 | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | 411615 | 439870 | 461180 | 340744 | 308236 | 369076 | 434805 | 472985 |
| 2 | | | 306680 | 331300 | 341636 | 305396 | 243448 | 311724 | 338828 | 354992 |
| 3 | | | 355088 | 384456 | 378500 | 432115 | 391460 | 480450 | 377392 | 414676 |
| 4 | | | 356424 | 349164 | 453395 | 347040 | 338476 | 334716 | 358460 | 496915 |
| 5 | | | 471795 | 486130 | 402976 | 372640 | 464464 | 492800 | 493005 | |
| 6 | | | 376124 | 370528 | 376588 | 448210 | 498510 | 365940 | 413440 | |
| 7 | | | 394732 | 496400 | 499895 | 384992 | 423912 | 394332 | 534650 | |
| 8 | | | 481695 | 401764 | 383652 | 370948 | 441750 | 432520 | 432648 | |
| 9 | | 173712 | 366776 | 379192 | 453055 | 440740 | 402056 | 384520 | 415732 | |
| 10 | | 352100 | 463140 | 492495 | 389600 | 352800 | 336564 | 381108 | 485350 | |
| 11 | | 424770 | 358468 | 370972 | 343900 | 306908 | 417950 | 439750 | 373284 | |
| 12 | | 358104 | 392772 | 369404 | 449135 | 433240 | 347916 | 386408 | 409732 | |

The answer to this question was done using the SSRS tool. The above screenshot shows the report generated that was deployed on the server. Each page of this report gives the customer count for all the months for a particular store. For the screenshot in question, it tells us what the customer count for store 107 was for each month across different years, precisely answering the question in the process. The report can be found here - [http://infodata16.mbs.tamu.edu/ReportServer/Pages/ReportViewer.aspx?%2fQuestion2%2fquestion2_group2_603&rs:Command=Render](http://infodata16.mbs.tamu.edu/ReportServer/Pages/ReportViewer.aspx?%2fQuestion2%2fquestion2_group2_603&rs:Command=Render)

**BQ2: What is the city-wise trend of sales of grooming products in the year X? Compare this with the logarithmic income median of each city.**

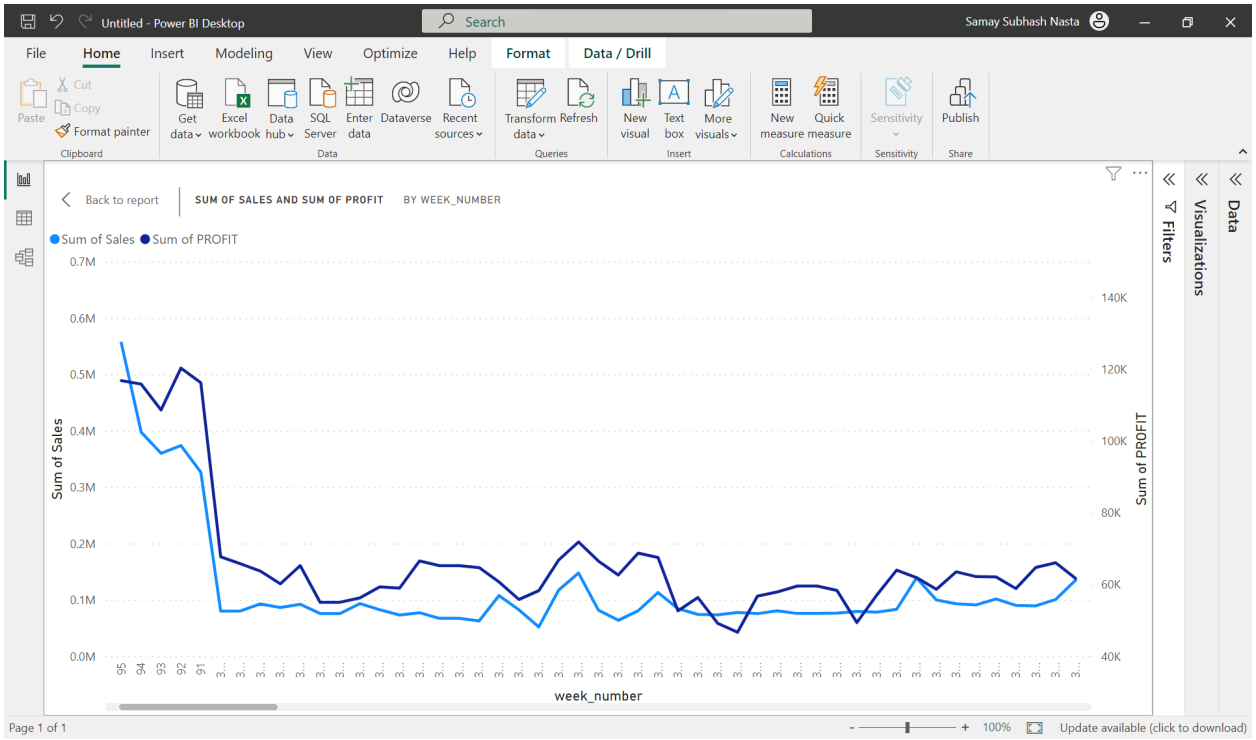
**Report using PowerBI**




Analyzing the city-wise trend of sales of grooming products in a particular year in comparison with the log income median of the city can help in identifying the relation between people's decision in purchasing grooming products based on their income, this will further help Dominic's Fine food in strategically managing their inventory and shelf space. To answer this business question, DIM_STORE_DEMO, DIM_PRODUCT, DIM_TIME and FACT_SALES of the Sales Datamart are used. The data from DIM_STORE_DEMO is used to get city information, DIM_TIME is used to aggregate data on weekly level, DIM_PRODUCT to filter the products and get data for grooming products and FACT_SALES to get sum of sales on a weekly level. Representing the data in a line and bar chart helps in understanding the relation between the log of average income and the sum of sales of grooming products.

**BQ3: How do the week-wise profit and dollar sales vary for Beer in the year X?**

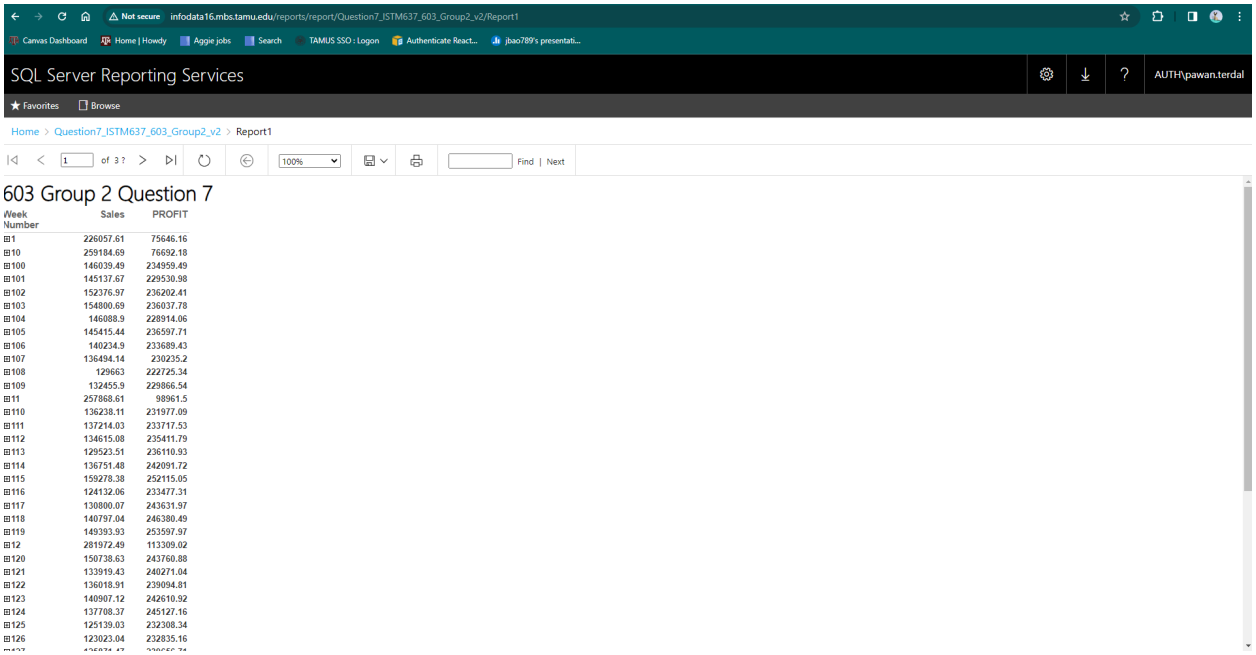
**Report Using PowerBI**




Analyzing the week-wise profit and sales for beer for a particular year will help the management understand the trend of sales and plan their inventory and shelf space accordingly. To answer the business question we will need to use DIM_TIME, DIM_PRODUCT and FACT_SALES tables. The data from DIM_TIME is used to aggregate data on weekly level, DIM_PRODUCT to filter the products and get data for Beer products and FACT_SALES to get the sum of sales and profit on a weekly level. Representing the data in a line chart helps in understanding the trend of beer sales in a particular year.

**BQ4: What is the weekly trend of sales for cigarettes across all stores?**

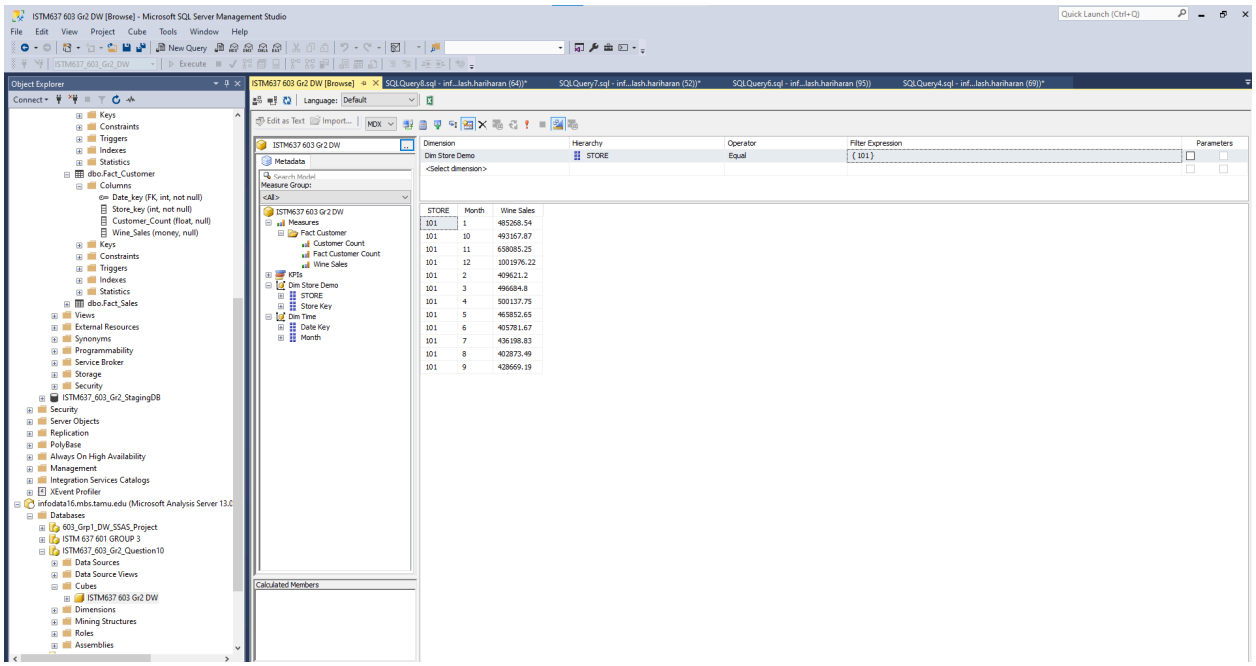
**SSAS + SSRS**




| Week Number | Sales | PROFIT |
|---|---|---|
| 1 | 226057.61 | 75646.16 |
| 10 | 259184.69 | 76692.18 |
| 100 | 146039.49 | 234959.49 |
| 101 | 145137.67 | 229530.98 |
| 102 | 152376.97 | 236202.41 |
| 103 | 154800.69 | 236037.78 |
| 104 | 146088.9 | 228914.06 |
| 105 | 145415.44 | 236597.71 |
| 106 | 140234.9 | 233689.43 |
| 107 | 136494.14 | 230235.2 |
| 108 | 129663 | 222725.34 |
| 109 | 132455.9 | 229866.54 |
| 11 | 257868.61 | 98961.5 |
| 110 | 136238.11 | 231977.09 |
| 111 | 137214.03 | 233717.53 |
| 112 | 134615.08 | 235411.79 |
| 113 | 129523.51 | 236110.93 |
| 114 | 136751.48 | 242091.72 |
| 115 | 159278.38 | 252115.05 |
| 116 | 124132.06 | 233477.31 |
| 117 | 130800.07 | 243631.97 |
| 118 | 140797.04 | 246380.49 |
| 119 | 149393.93 | 253597.97 |
| 12 | 281972.49 | 113309.02 |
| 120 | 150738.63 | 243760.88 |
| 121 | 133919.43 | 240271.04 |
| 122 | 136018.91 | 239094.81 |
| 123 | 140907.12 | 242610.92 |
| 124 | 137708.37 | 245127.16 |
| 125 | 125139.03 | 232308.34 |
| 126 | 123023.04 | 232835.16 |

Was done using SSAS+SSRS. This report accurately answers the business question of weekly trends of cigarette sales for different stores. The screenshot above shows sales and profit figures for cigarettes for each week determined by the week number.
[http://infodata16.mbs.tamu.edu/ReportServer/Pages/ReportViewer.aspx?%2fQuestion7_ISTM637_603_Group2_v2%2fReport1&rs:Command=Render](http://infodata16.mbs.tamu.edu/ReportServer/Pages/ReportViewer.aspx?%2fQuestion7_ISTM637_603_Group2_v2%2fReport1&rs:Command=Render)

**BQ5: What were the wine sales for different stores in a given month?**
**SSAS**




Tool used to generate this report was **SSAS**. As evidenced by the screenshot above, this report helps us answer the business question of wine sales for different stores in a given month. This shows the wine sales for store 101 for different months. The report can be changed to show the sales figures for different stores very easily by just changing the store number.

## 5.1.2 Determine all the mappings from the independent data marts to the report attributes

**BQ1: What does total footfall for Store X look like throughout a particular year?**

| Attribute Name | Attribute's Dimension/Fact table | Filters | Corresponding Report Attribute |
|---|---|---|---|
| Store_Number | Dim_Store_Demo | Change page to see different stores | Store |
| Year | Dim_Time | All years | Year |
| Month | Dim_Time | | Month |
| Customer_Count | Fact_Customer | Sum | Values |

**BQ2: What is the city-wise trend of sales of grooming products in the year X? Compare this with the logarithmic income median of each city.**

| Attribute Name | Attribute's Dimension/Fact table | Transformations | Corresponding Report Attribute |
|---|---|---|---|
| City | Dim_Store_Demo | | City |
| Year | Dim_Time | Filter for one year | Year |
| Month | Dim_Time | | Month |
| Log_Income | Dim_Store_Demo | Average | Income |
| Sales | Fact_Sales | Sum | Sales |
| Product_category | Dim_Product | Filter for Grooming products | N/A |

**BQ3: How do the week-wise profit and dollar sales vary for Beer in the year X?**

| Attribute Name | Attribute's Dimension/Fact table | Transformations | Corresponding Report Attribute |
|---|---|---|---|
| Year | Dim_Time | Filter for one year | Year |
| Week | Dim_Time | | Week |
| Sales | Fact_Sales | Sum | Sales |
| Profit | Fact_Sales | Sum | Profit |
| Product_category | Dim_Product | Filter for Beer | N/A |

**BQ4: What is the weekly trend of sales for cigarettes across all stores?**

| Attribute Name | Attribute's Dimension/Fact table | Transformations | Corresponding Report Attribute |
|---|---|---|---|
| Week | Dim_Time | | Week |
| Sales | Fact_Sales | Sum | Sales |
| Profit | Fact_Sales | Sum | Profit |

| | | | |
|---|---|---|---|
| Product_category | Dim_Product | Filter for Cigarette | N/A |

**BQ5: What were the wine sales for different stores in a given month?**

| **Attribute Name** | **Attribute's Dimension/Fact table** | **Transformations** | **Corresponding Report Attribute** |
|---|---|---|---|
| Store | Dim_Store_Demo | Filter for a given store | Store |
| Month | Dim_Time | | Month |
| Wine_Sales | Fact_Customer | Sum | Sales |

### 5.1.3 Describe All Report Templates

Reports Using SSRS

SQL Server Reporting Services has a user-friendly design interface, supports a number of data sources, and has extensive formatting and visualization features. With SSRS, one can plan and automate report production, assuring timely and accurate information delivery. Its role-based security features boost data security. Furthermore, SSRS provides parameterized reports, which allow users to tailor outputs. Overall, SSRS is a comprehensive tool for creating, maintaining, and distributing reports inside an organization [10]. Hence, we have used SSRS to answer questions 1 and 4.

Cubes from SSAS

SQL Server Analysis Services Cubes are used when businesses need multidimensional analysis for complicated business scenarios. Cubes aggregate and pre-calculate data, boosting query efficiency for analytical queries. They are useful for managing enormous datasets and performing complex computations. SSAS delivers a uniform and centralized data model, which improves consistency across reports and dashboards. Cubes enable ad-hoc querying and dynamic drill-downs, providing users with interactive analytics. Overall, SSAS cubes are favored when enterprises want complete multidimensional analysis, improved performance, and sophisticated analytics capabilities in their business intelligence platforms[11]. Hence used SSAS to answer business questions 4 and 5.

PowerBI Report

PowerBI reports are suitable for organizations looking for interactive and visually appealing data analysis and reporting. It is a user-friendly interface for producing dynamic reports and

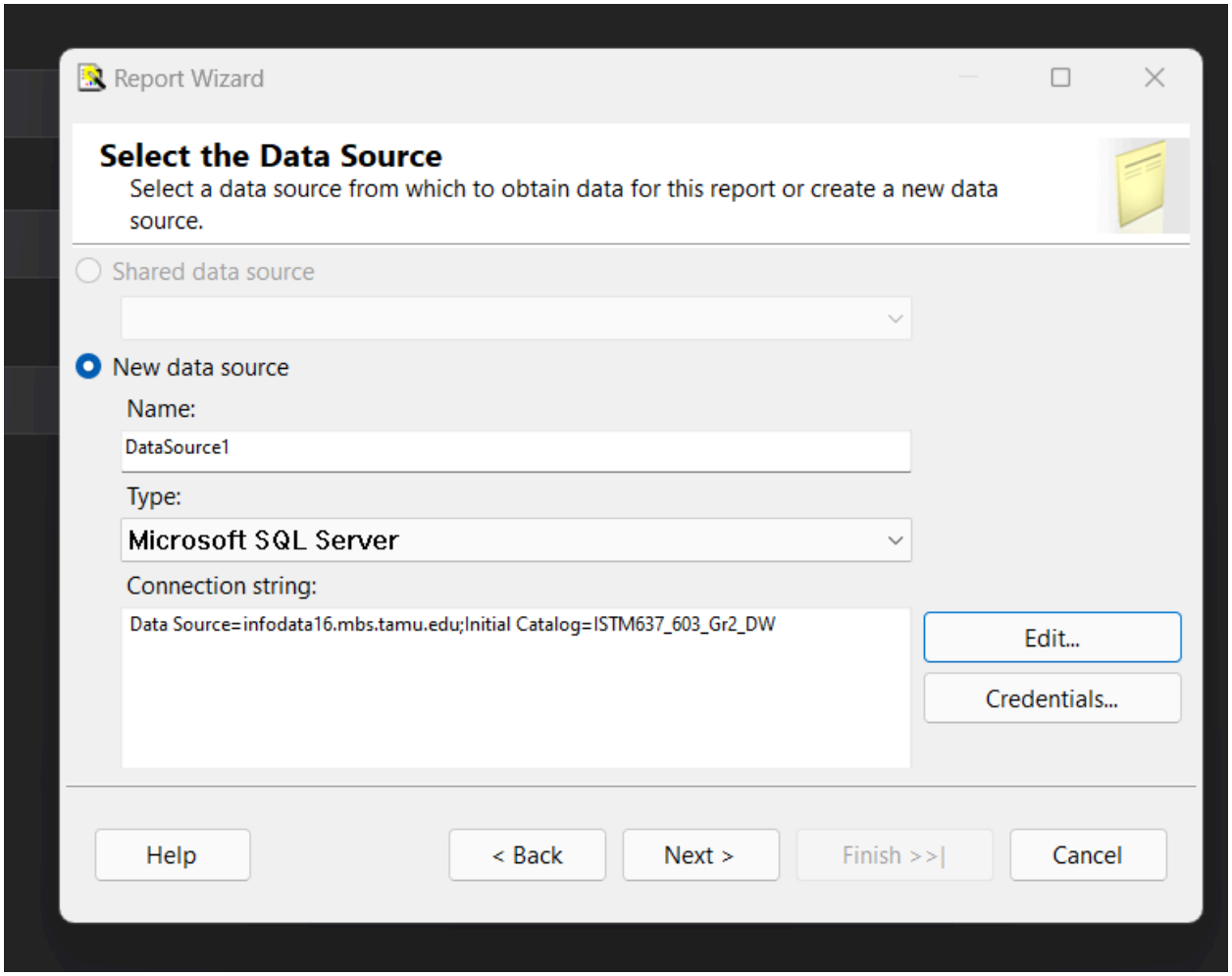
dashboards. It is appropriate for circumstances in which real-time or near-real-time data insights are critical. PowerBI integrates to numerous data sources, allowing for smooth integration with a wide range of information. It is especially useful for self-service business intelligence, allowing users to produce reports without requiring substantial technical knowledge. Power BI's sharing and collaboration features make it ideal for team-based data investigation. Overall, Power BI is suggested when businesses value intuitive, interactive, and collaborative data visualization for decision-making processes. PowerBI reports were used to answer the questions 2 & 3.

## 5.2 Reporting Implementation

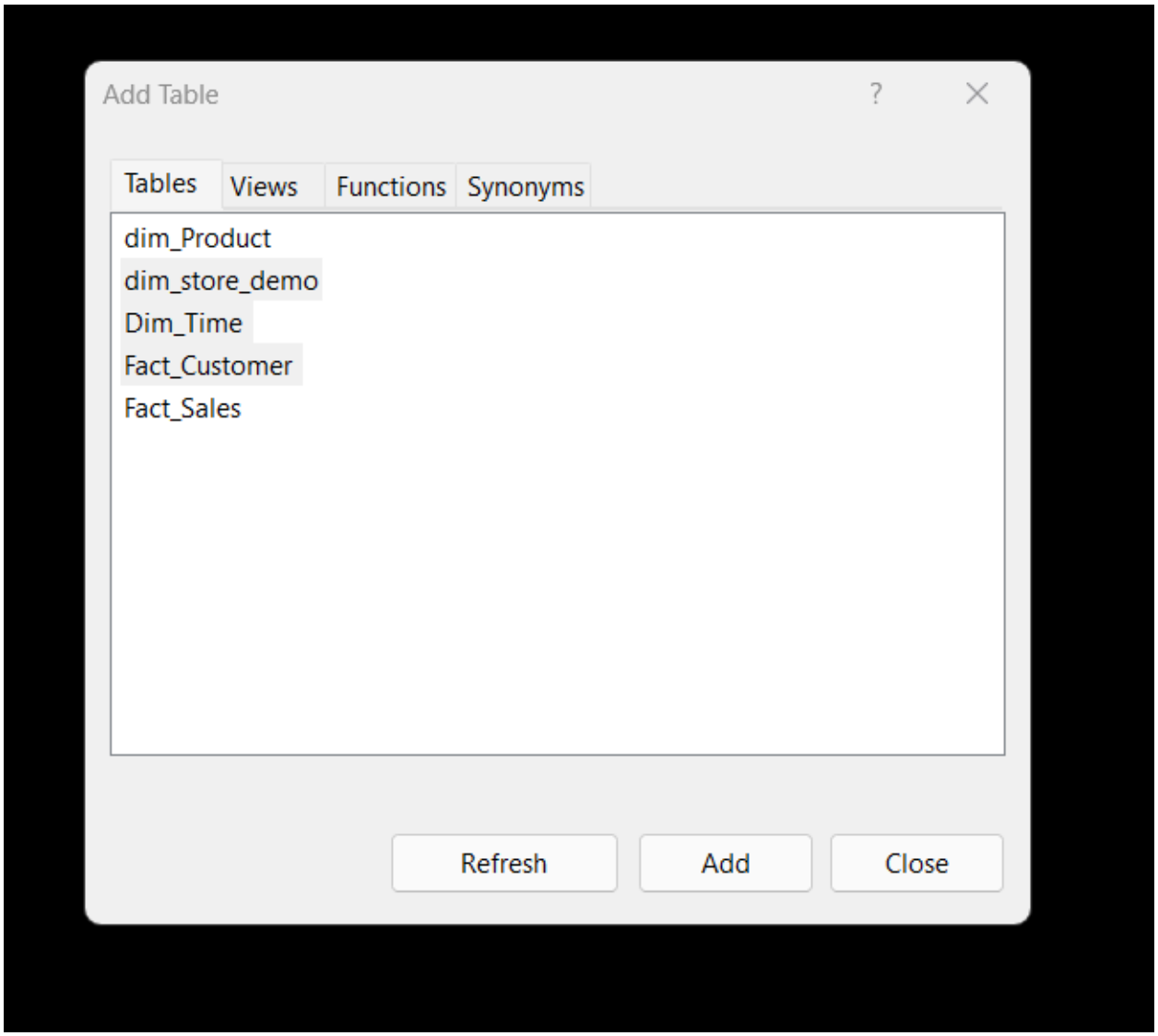
BQ1: **What does total footfall for Store X look like throughout a particular year?**

### 5.2.1 Report using SSRS

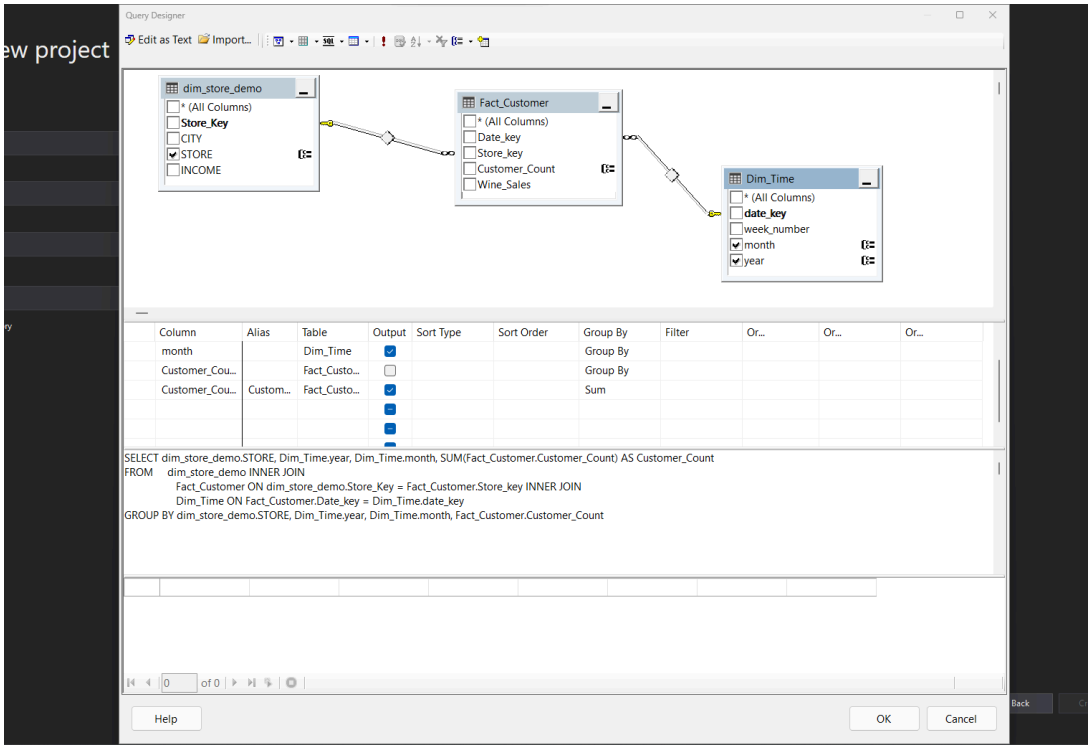
Choose the data source. It is the DW database that we have created under our group name.



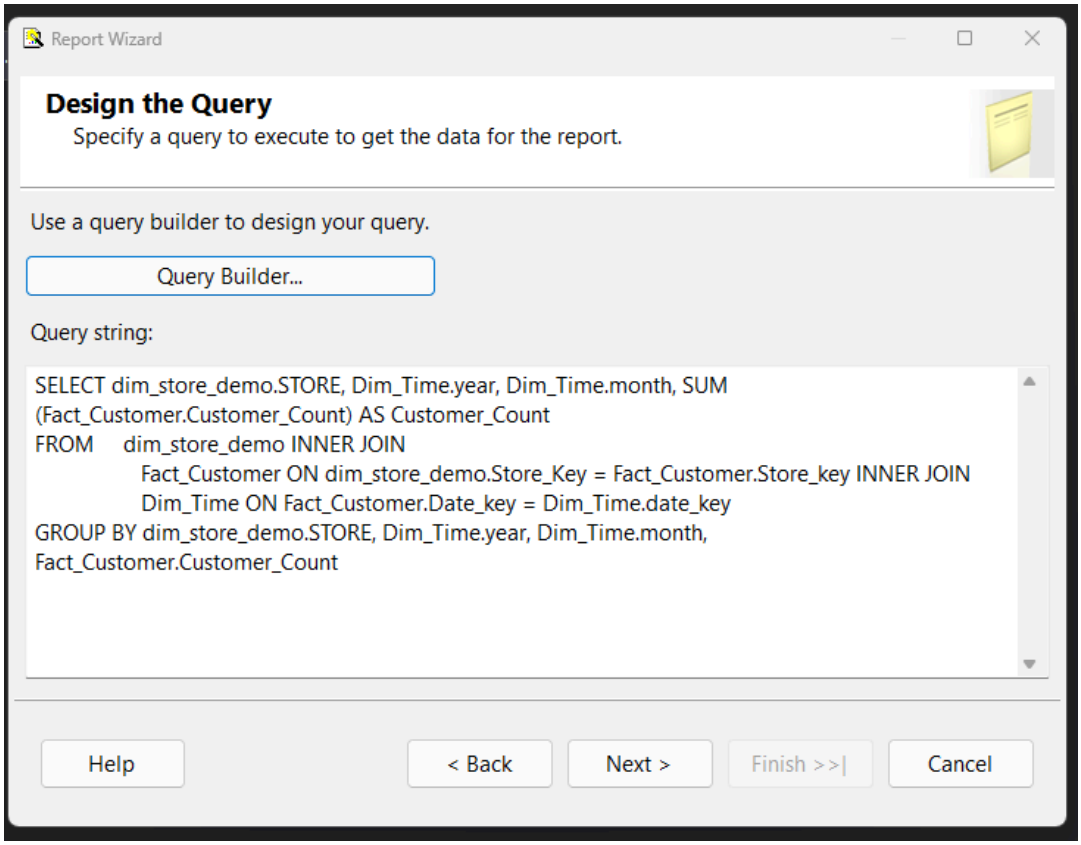
Choose the Dimension tables and Fact tables needed to answer the question



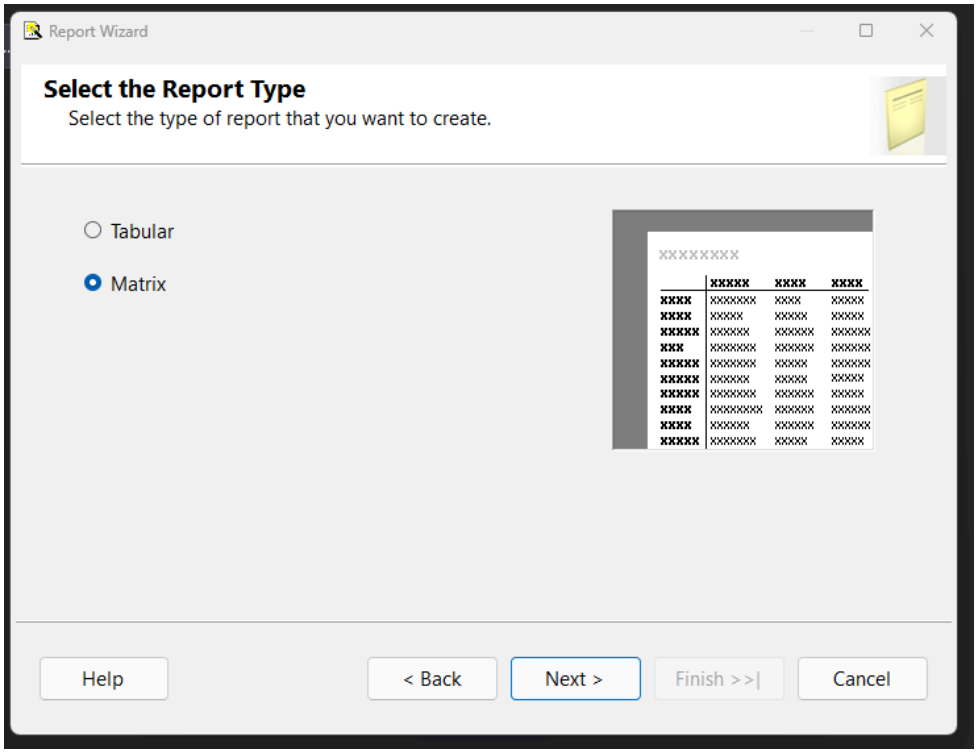
Select the fields from the columns that are relevant to the business question



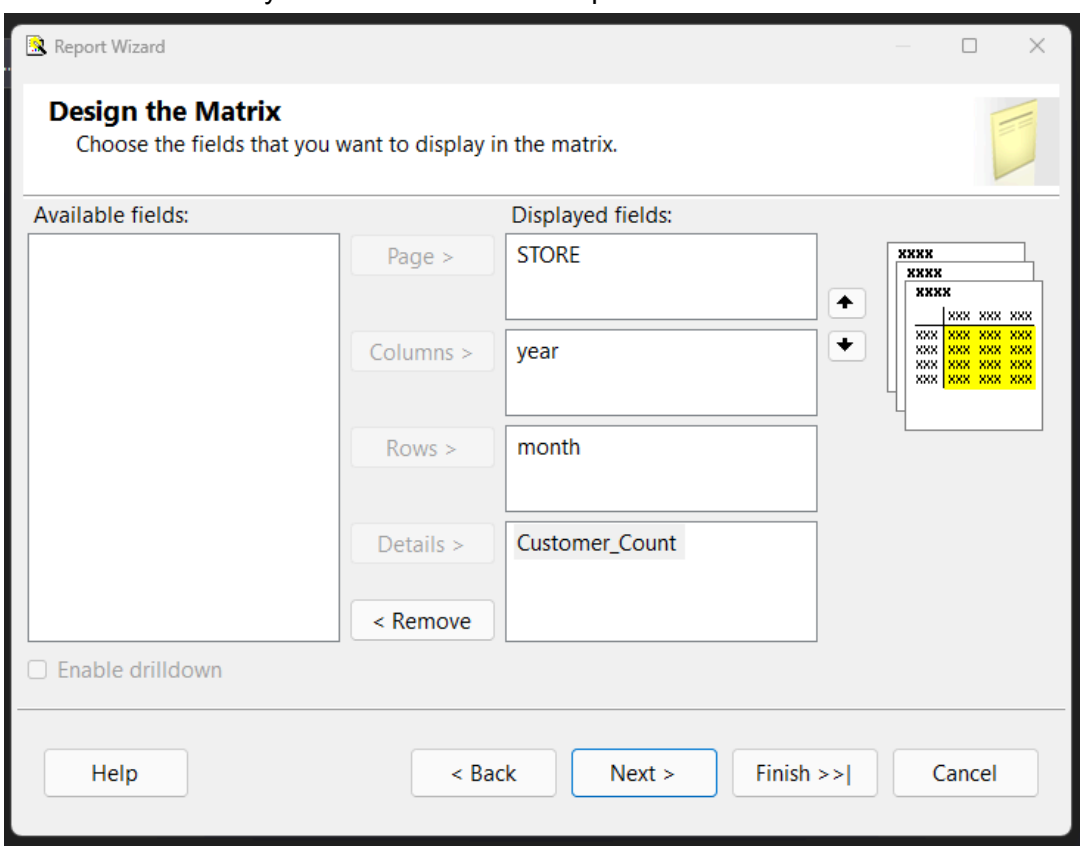

Modify the generated query if needed

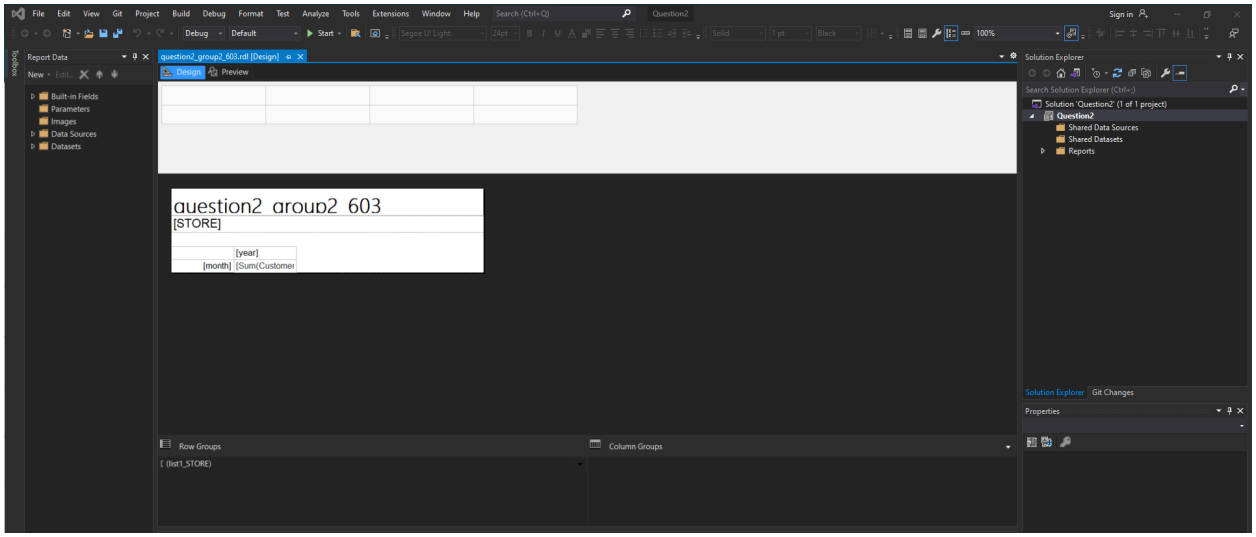


Choose the structure of report wanted

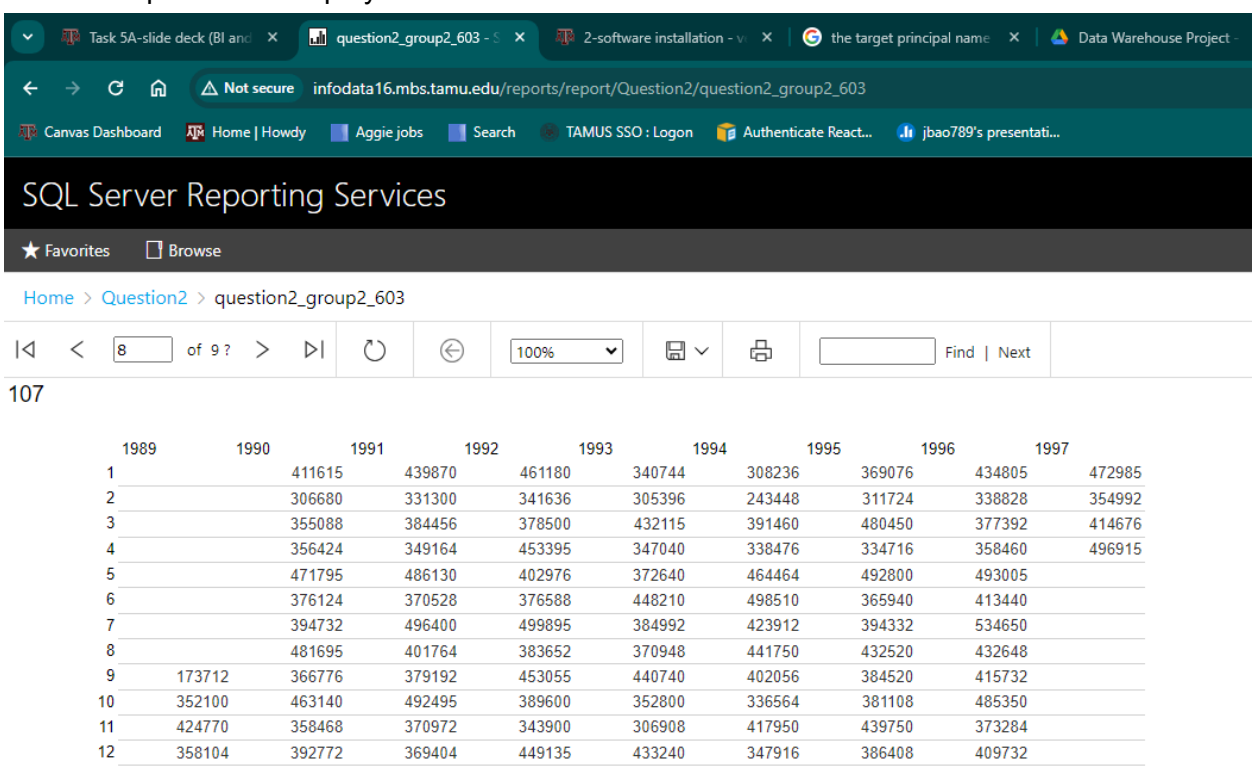



Choose the hierarchy of fields to match the report structure






The final report that is deployed on the server



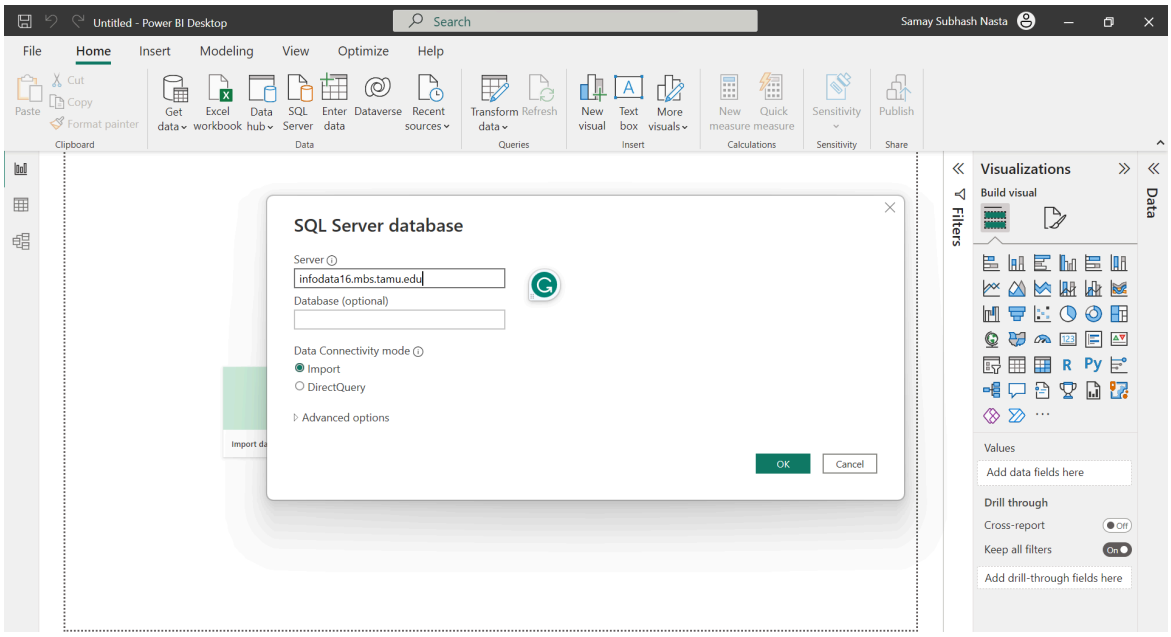

107

| | 1989 | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | 411615 | 439870 | 461180 | 340744 | 308236 | 369076 | 434805 | 472985 |
| 2 | | | 306680 | 331300 | 341636 | 305396 | 243448 | 311724 | 338828 | 354992 |
| 3 | | | 355088 | 384456 | 378500 | 432115 | 391460 | 480450 | 377392 | 414676 |
| 4 | | | 356424 | 349164 | 453395 | 347040 | 338476 | 334716 | 358460 | 496915 |
| 5 | | | 471795 | 486130 | 402976 | 372640 | 464464 | 492800 | 493005 | |
| 6 | | | 376124 | 370528 | 376588 | 448210 | 498510 | 365940 | 413440 | |
| 7 | | | 394732 | 496400 | 499895 | 384992 | 423912 | 394332 | 534650 | |
| 8 | | | 481695 | 401764 | 383652 | 370948 | 441750 | 432520 | 432648 | |
| 9 | | 173712 | 366776 | 379192 | 453055 | 440740 | 402056 | 384520 | 415732 | |
| 10 | | 352100 | 463140 | 492495 | 389600 | 352800 | 336564 | 381108 | 485350 | |
| 11 | | 424770 | 358468 | 370972 | 343900 | 306908 | 417950 | 439750 | 373284 | |
| 12 | | 358104 | 392772 | 369404 | 449135 | 433240 | 347916 | 386408 | 409732 | |

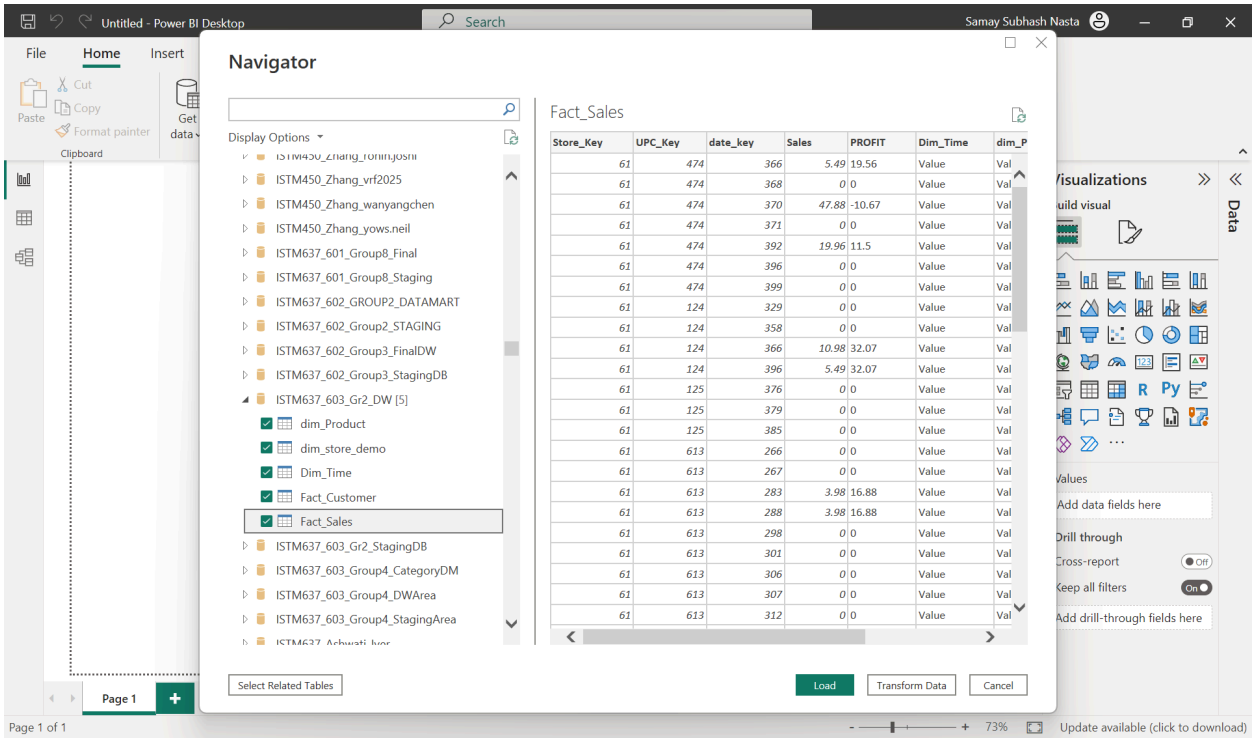
## 5.2.2 Report using PowerBI - 1

**BQ2: What is the city-wise trend of sales of grooming products in the year X? Compare this with the logarithmic income median of each city.**

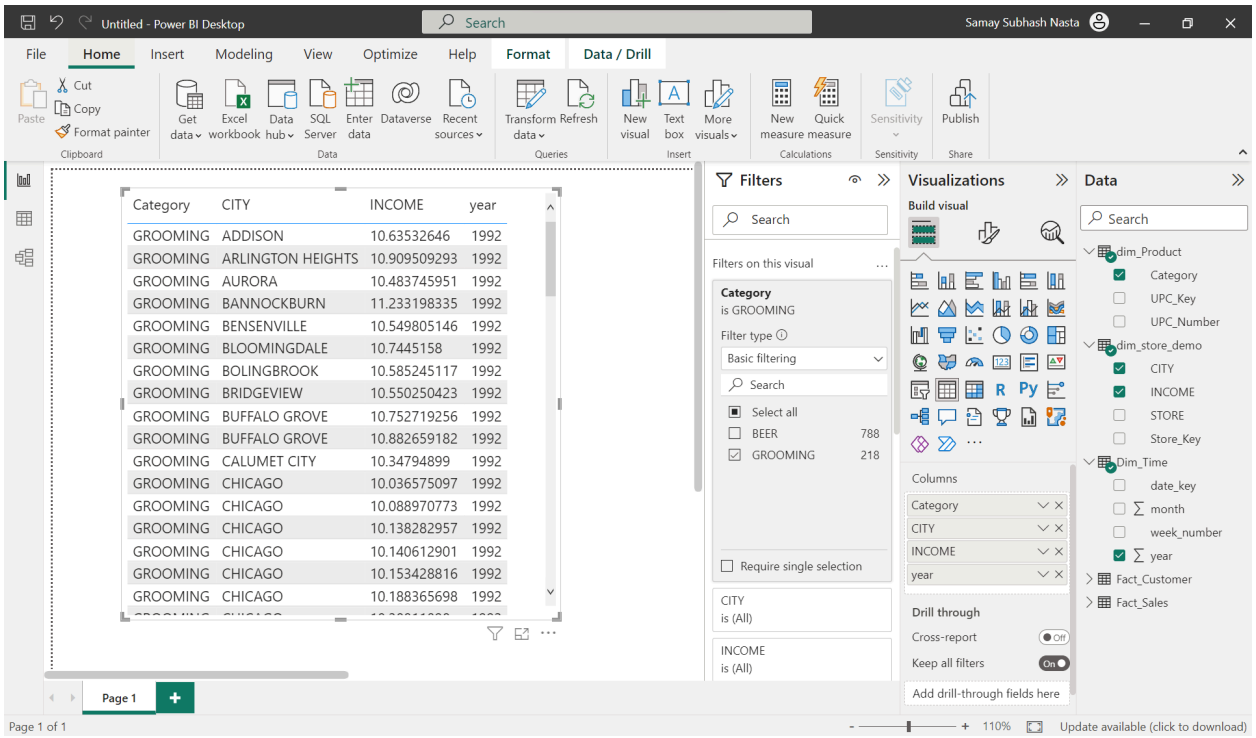
Connect the Power BI tool to SQL server to fetch the data needed



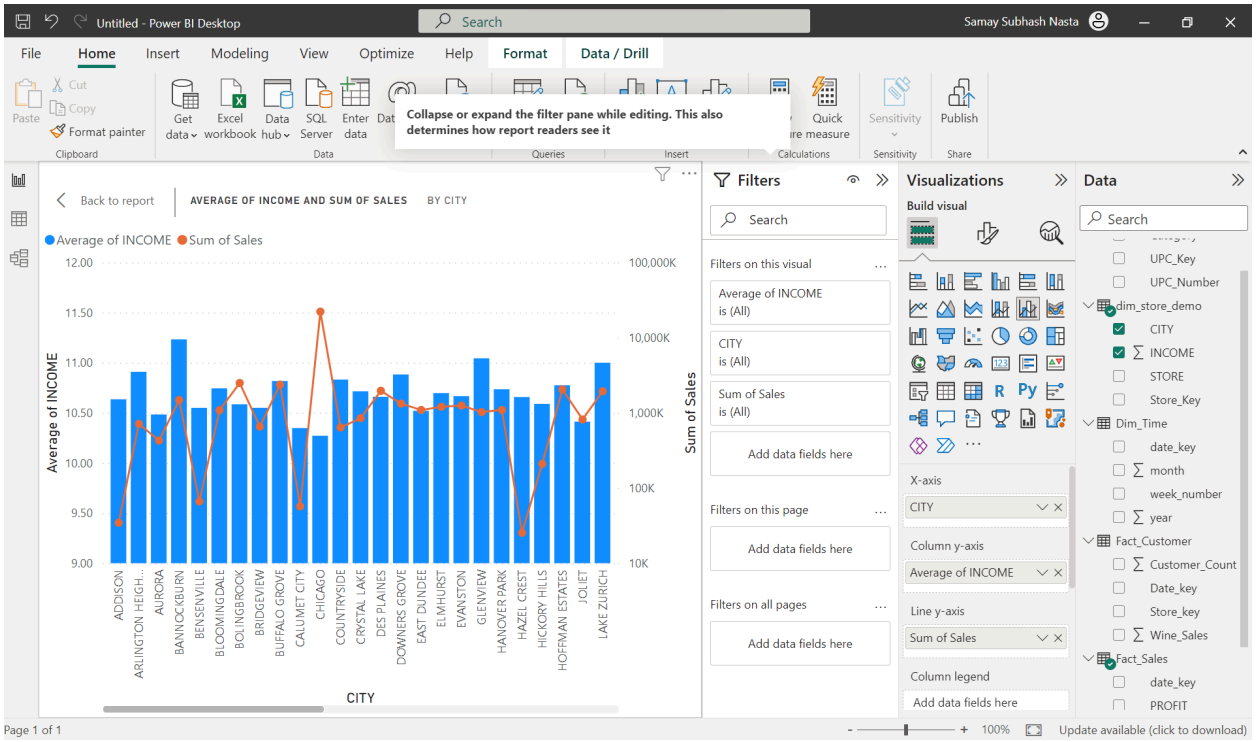

Select the appropriate tables needed to answer the question



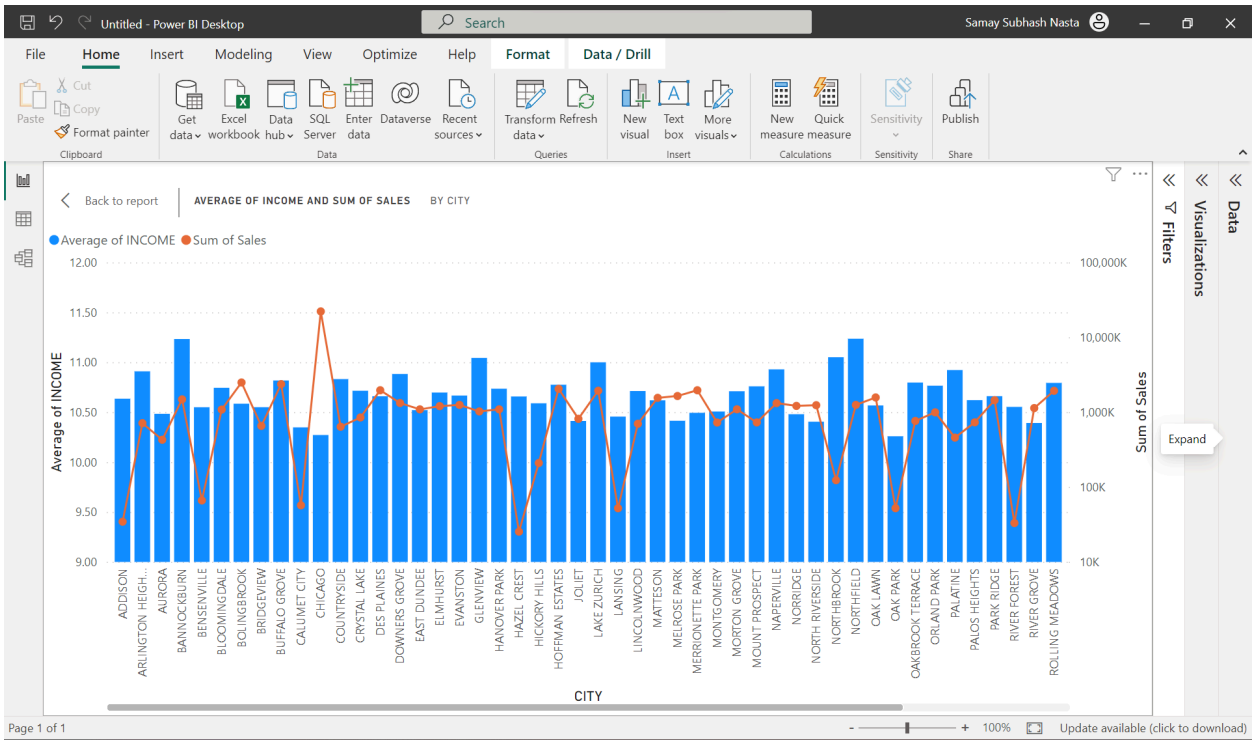

Drag the fields that go into X and Y axes respectively




Changed the calculation for the income to show average values and sum of sales.



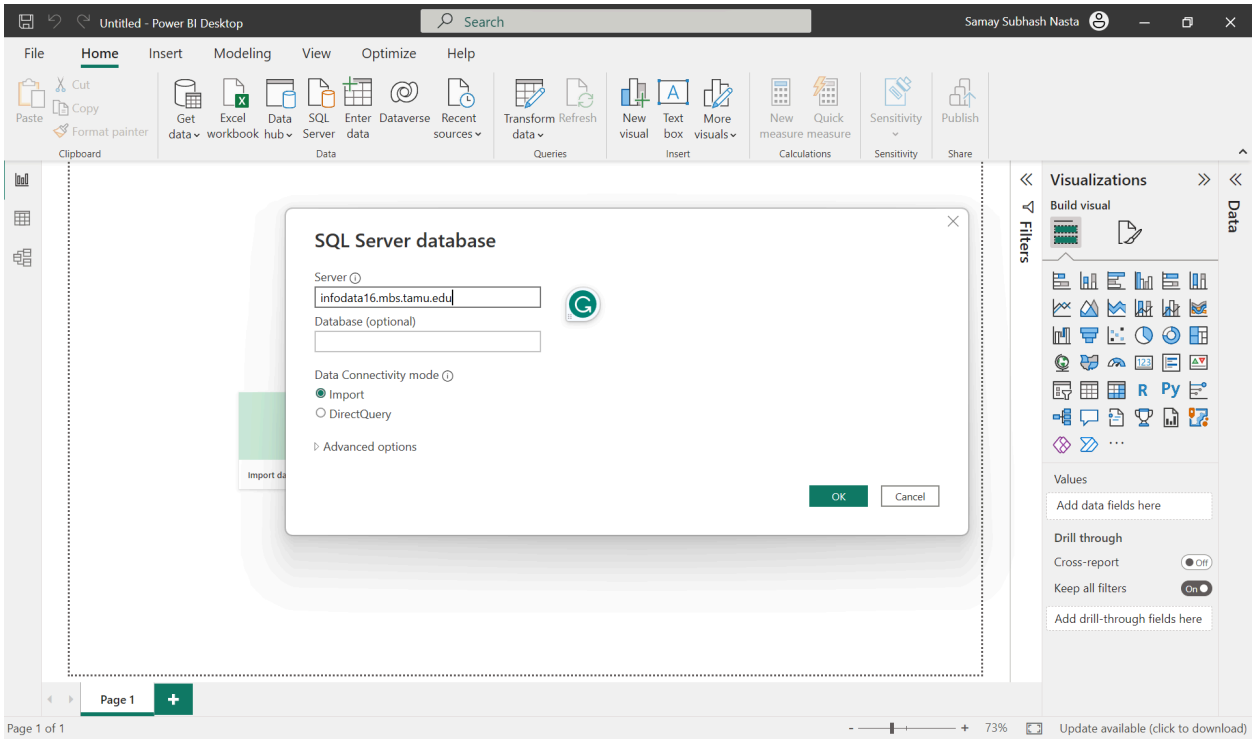

Modify colors if needed. Adjust the scale of X and Y axes to fit the structure. Finally, display the chart.



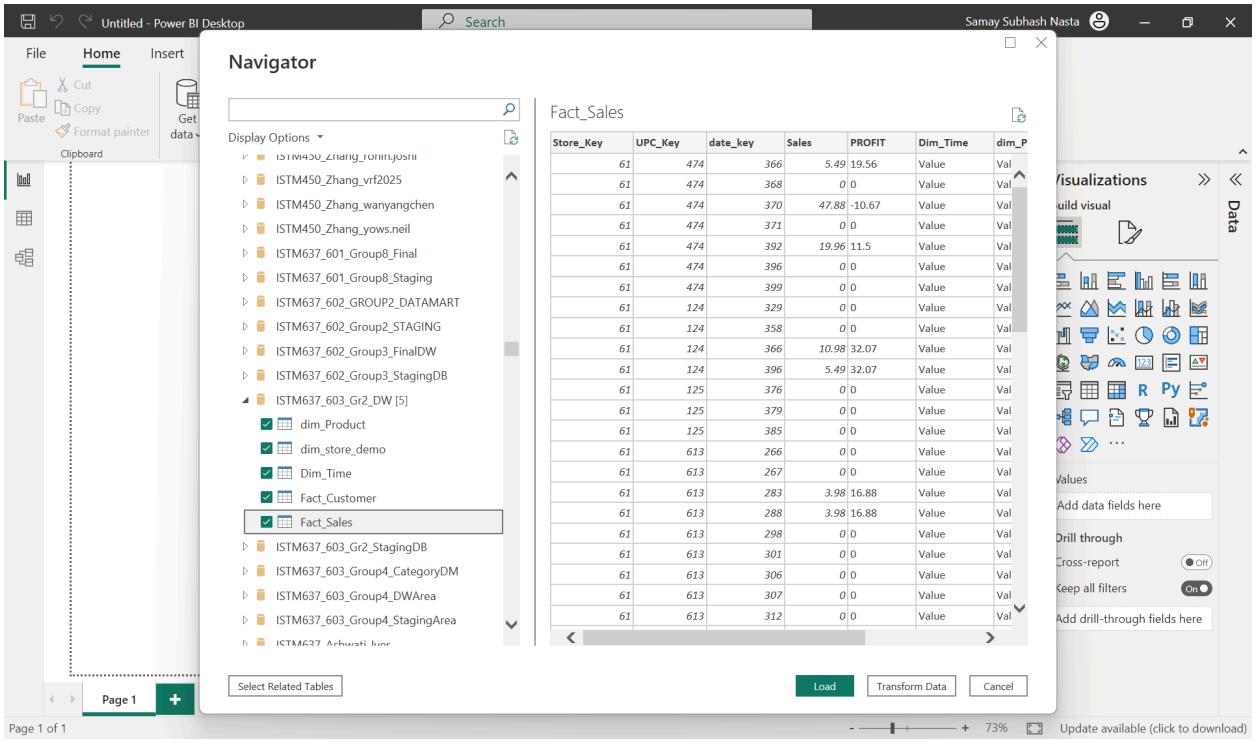
## 5.2.3 Report using PowerBI - 2

**BQ3: What is the city-wise trend of sales of grooming products in the year X? Compare this with the logarithmic income median of each city.**

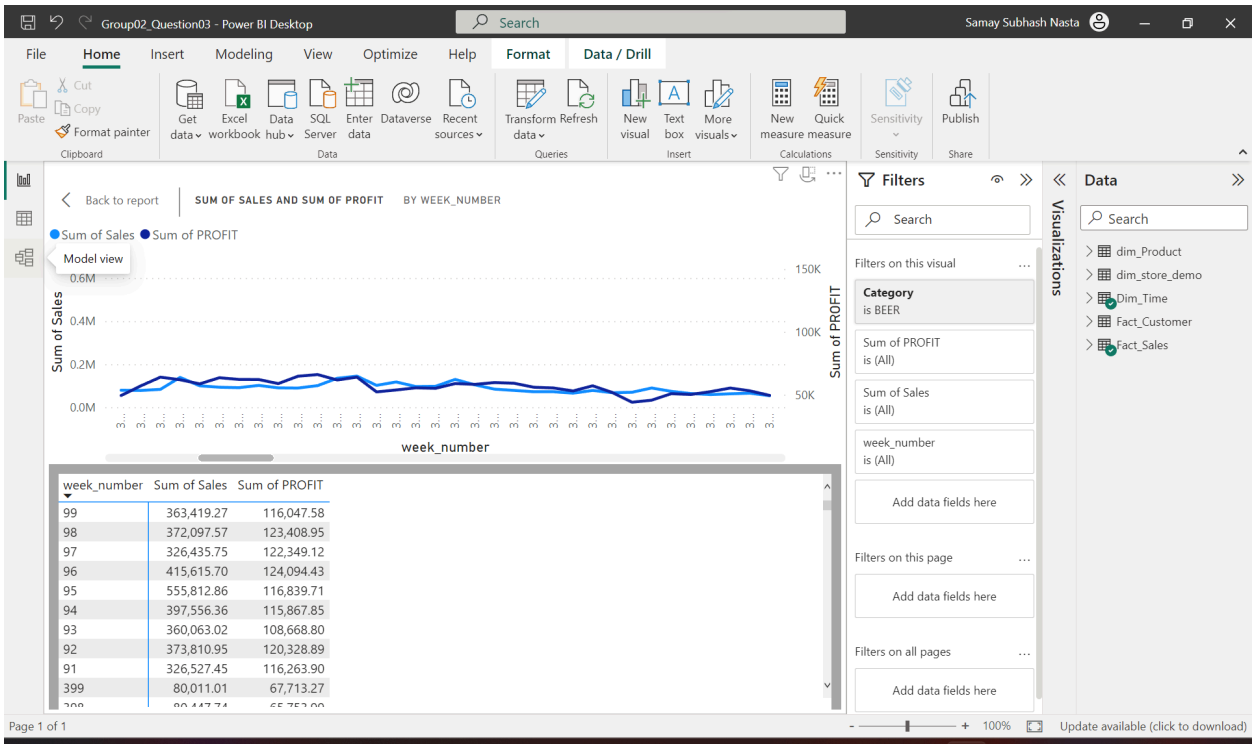
Connect to the data source, which in our case is the SQL server



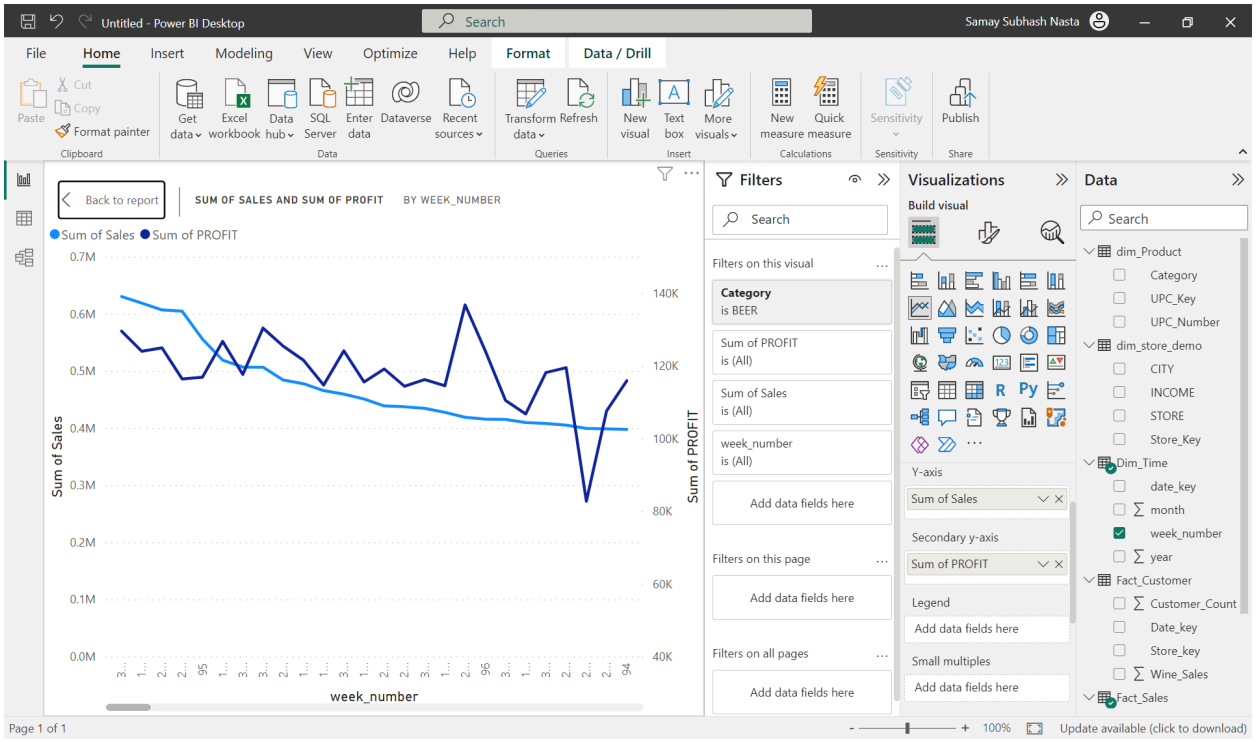
Select the appropriate tables and fields needed to answer the question



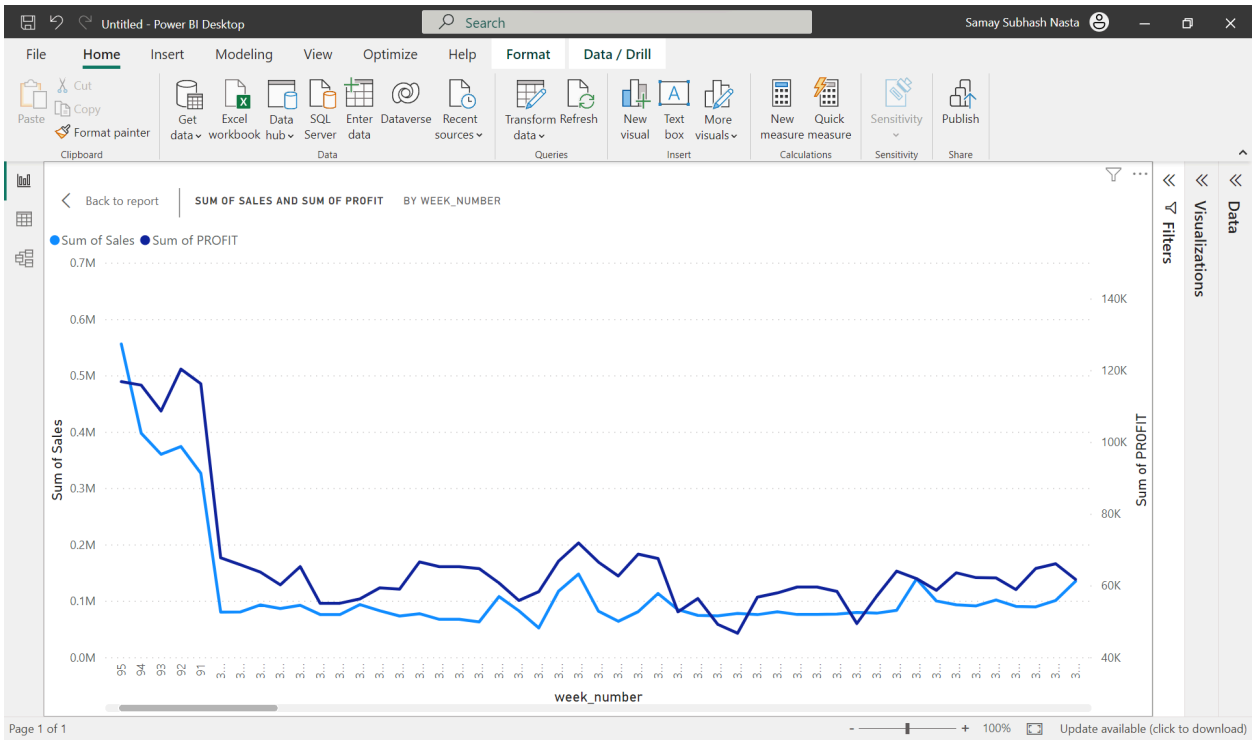

Drag the fields needed to display on the graph



Adjust the scale and other fields as needed



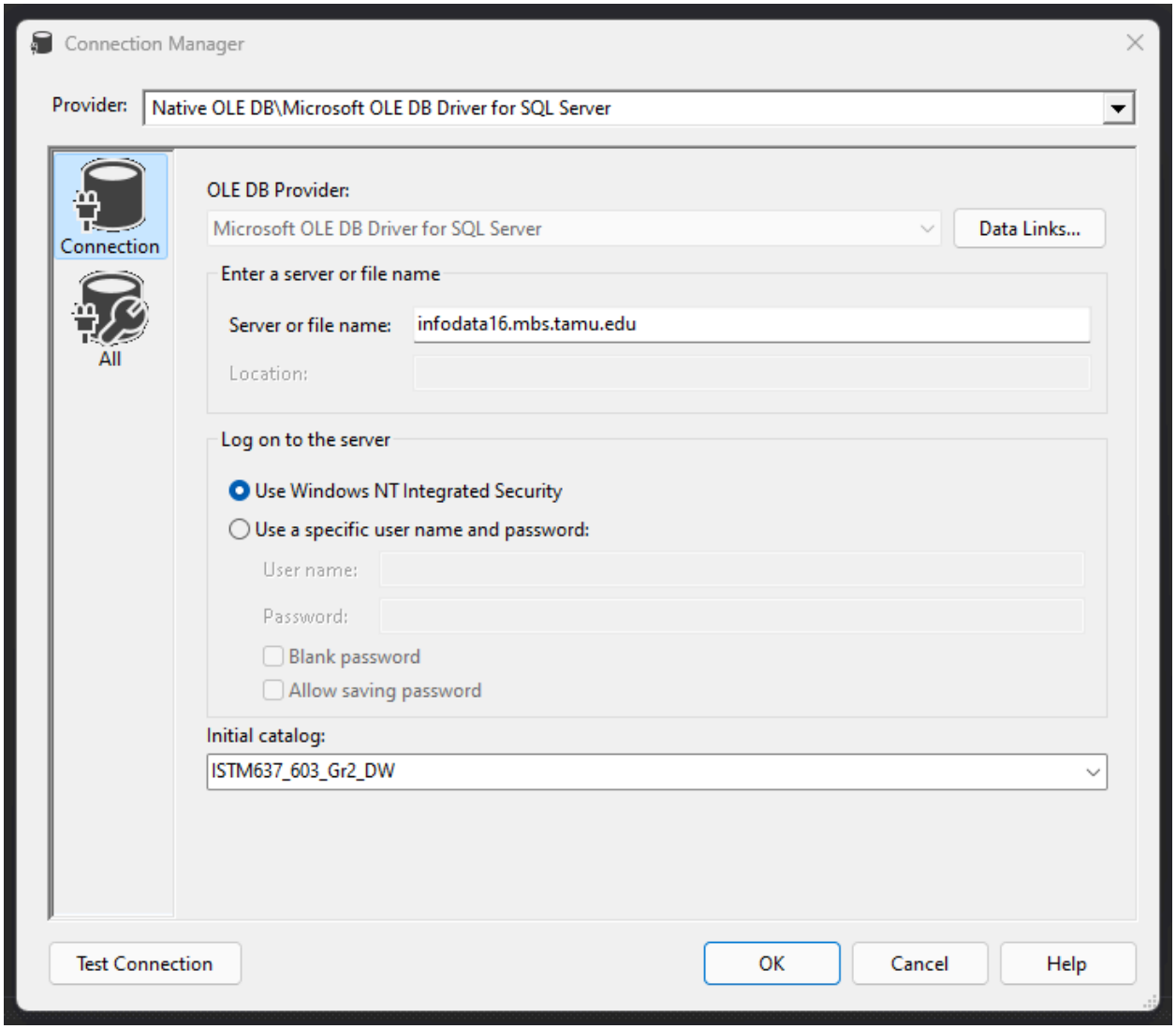

The final reported generated



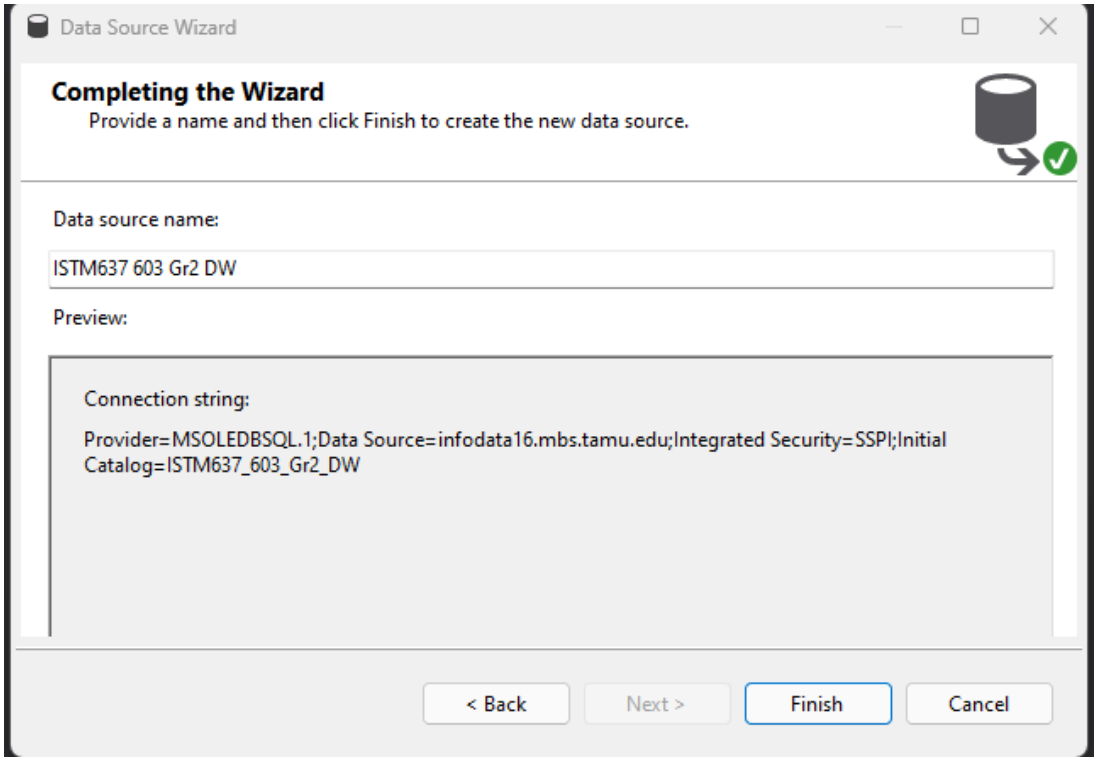
## 5.2.4 Cube creation using SSAS and Reporting on SSRS

**BQ4: What is the weekly trend of sales for cigarettes across all stores?**

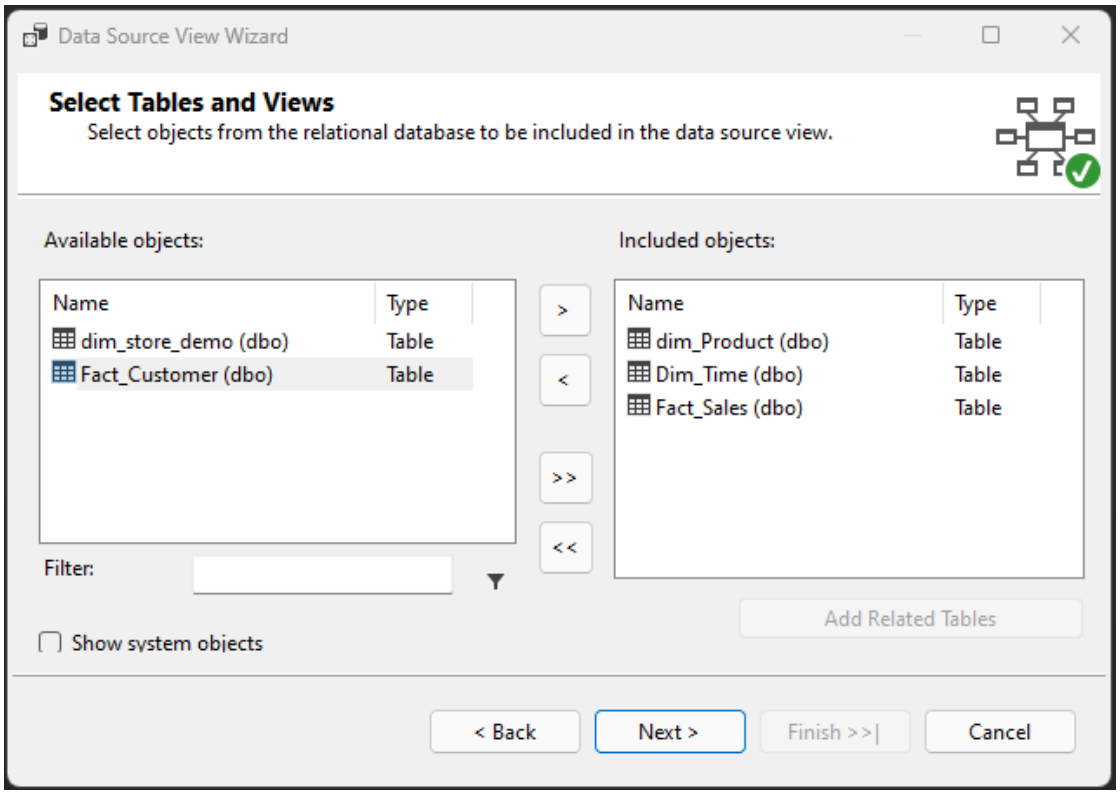
Choose the data source by connecting to SQL server



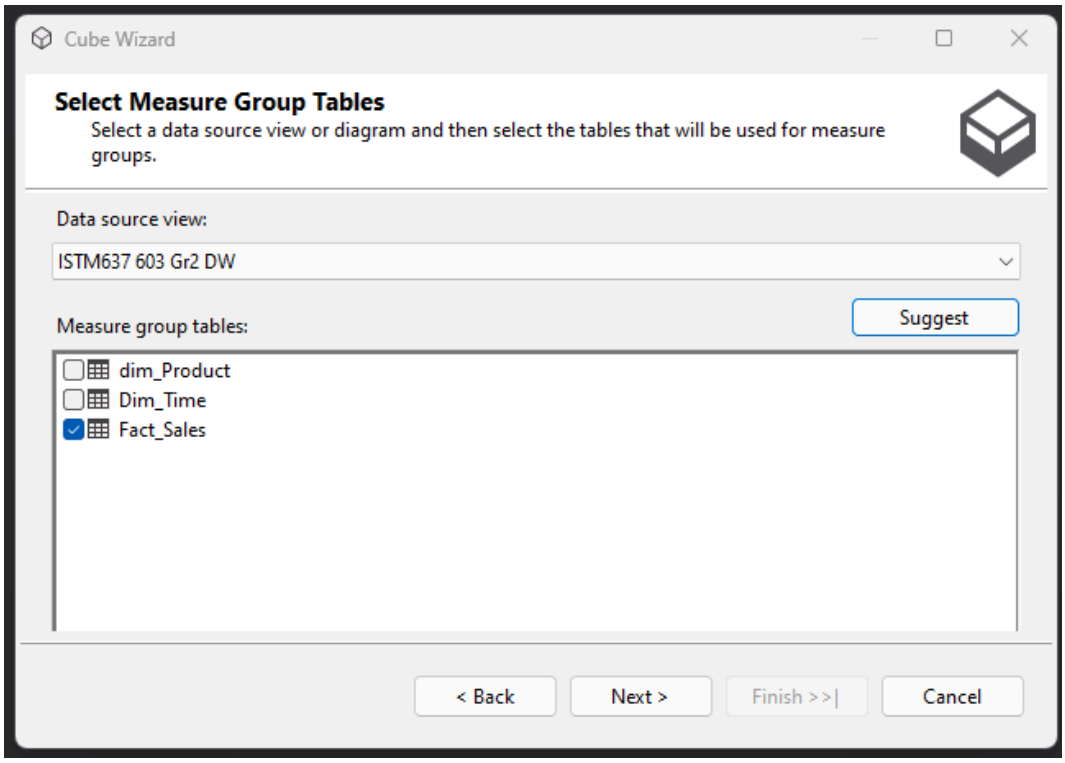
Create Data Source



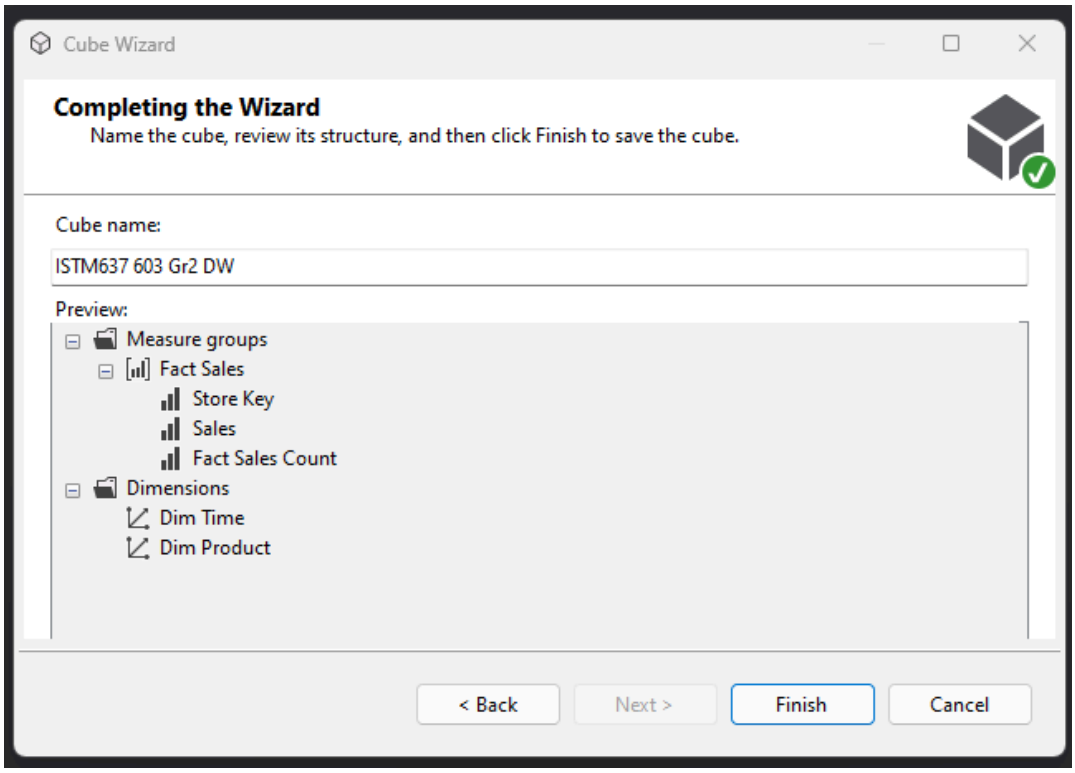

Select the right database and tables needed to answer the question



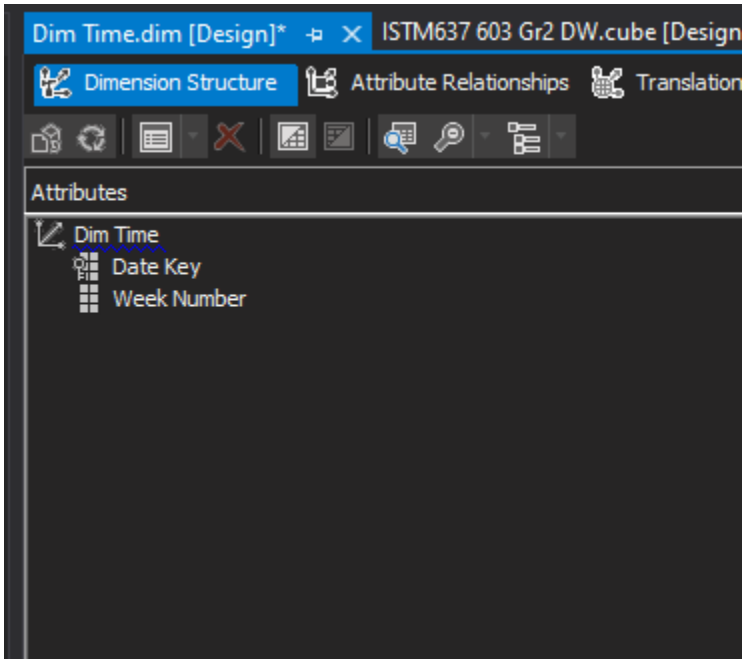
Select Fact table as measure groups

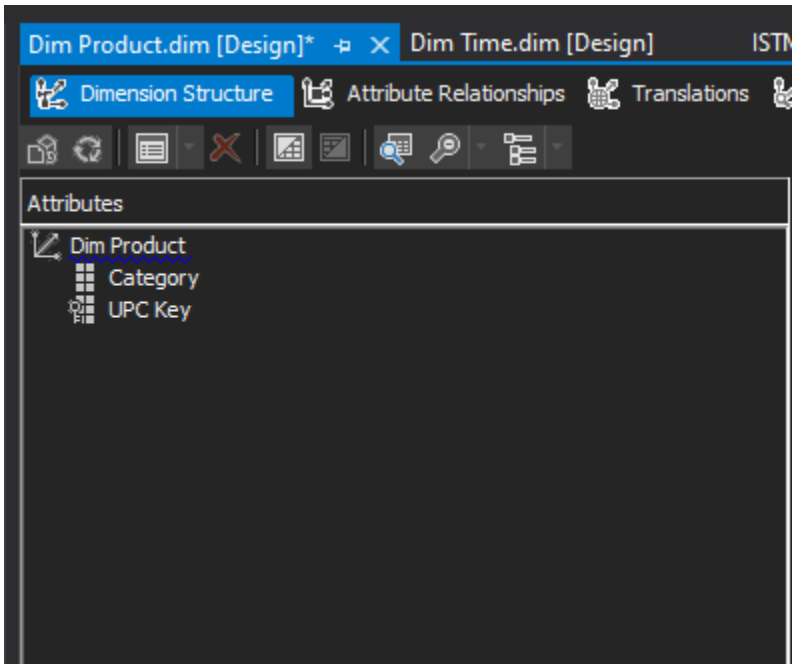



Give an appropriate name to the cube and click finish

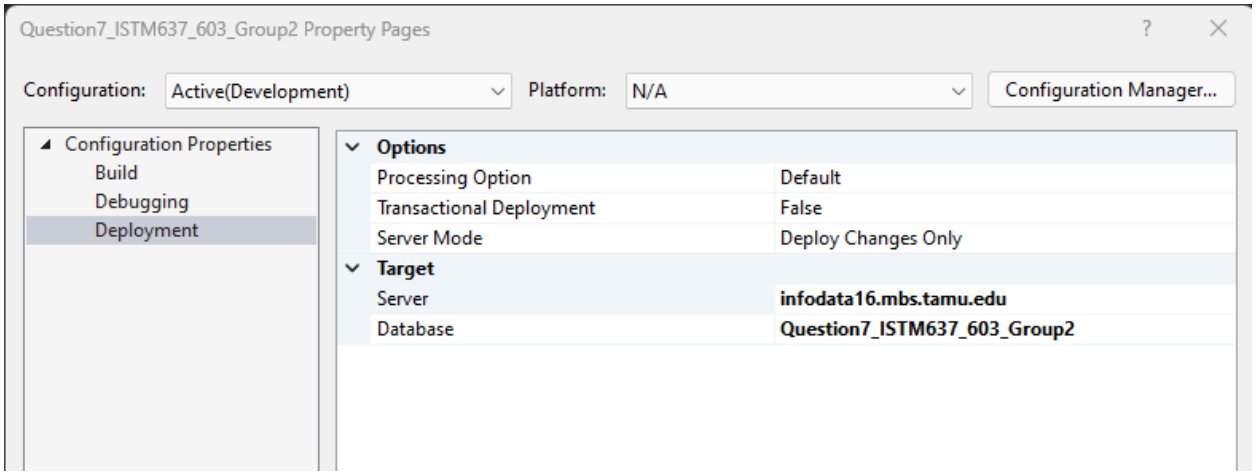


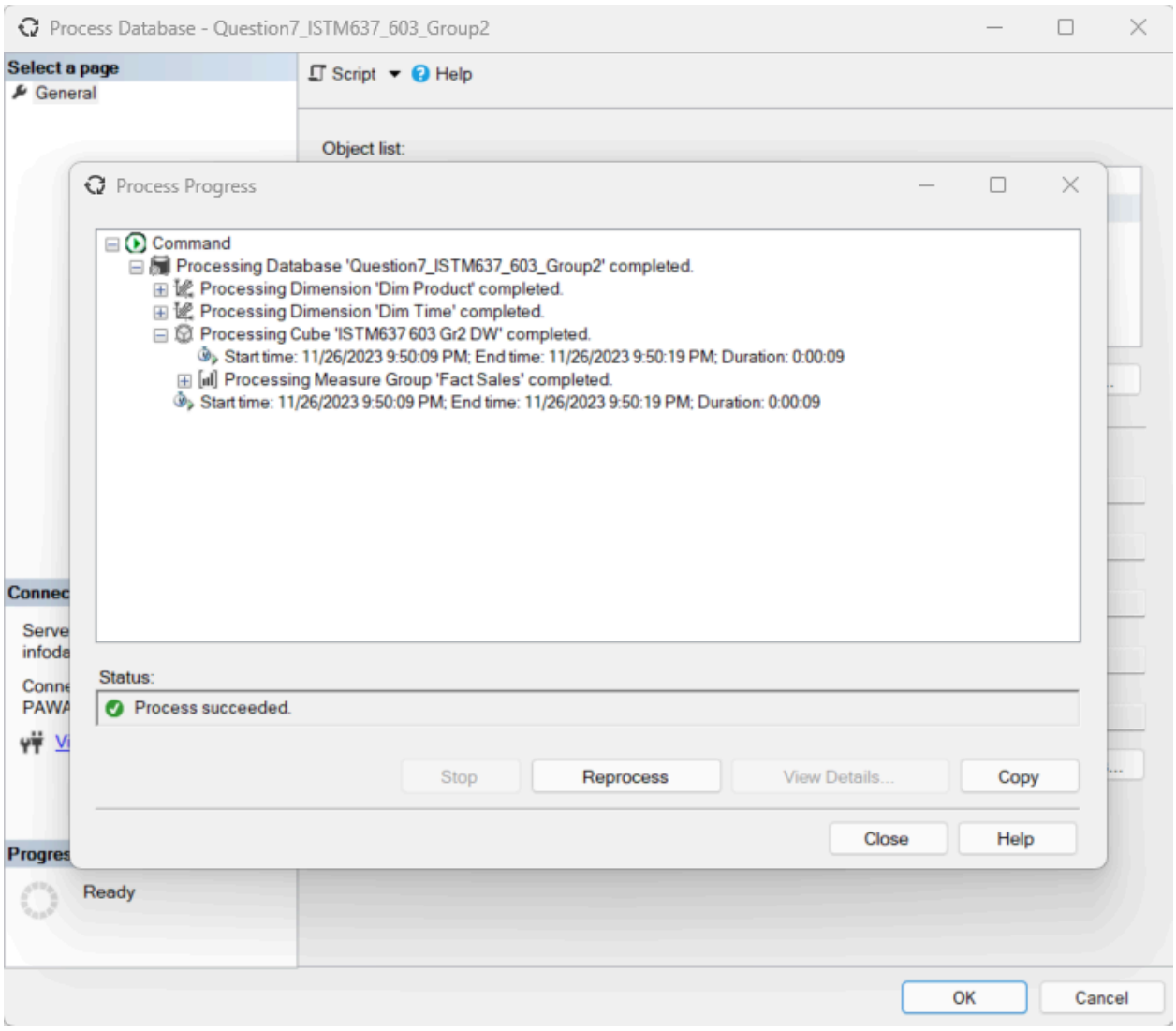
Choose the right attributes from the tables



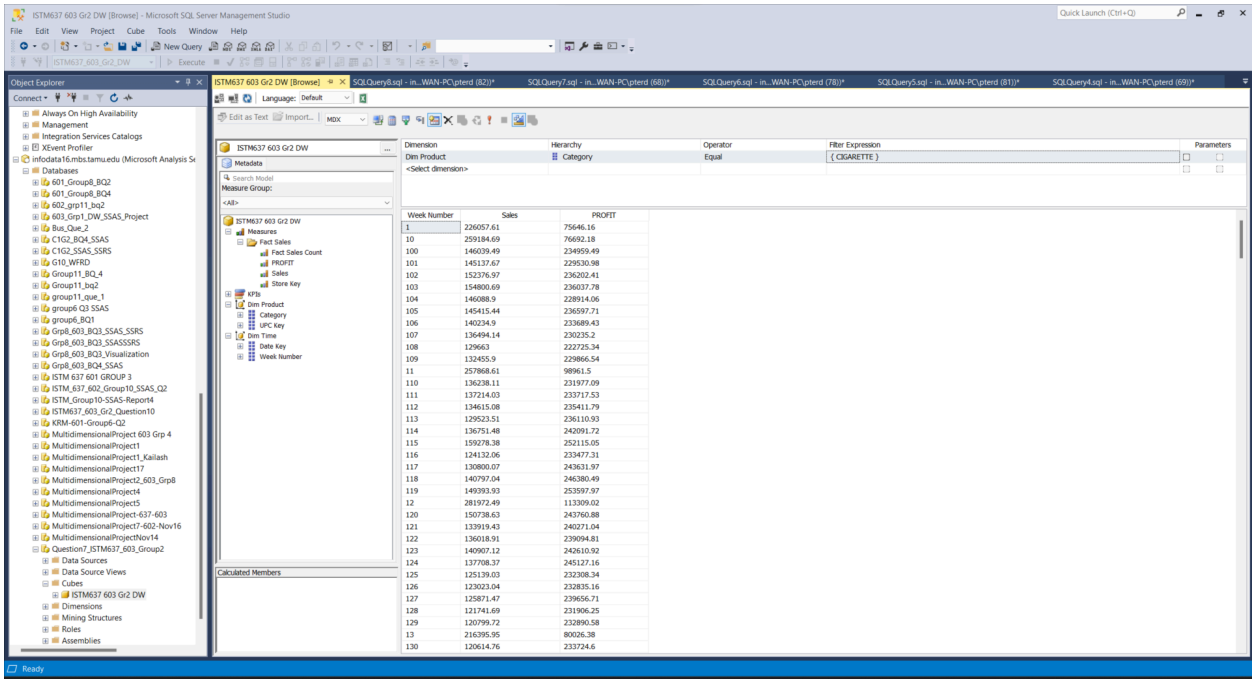



Mention the target server where this SSAS cube should be deployed.

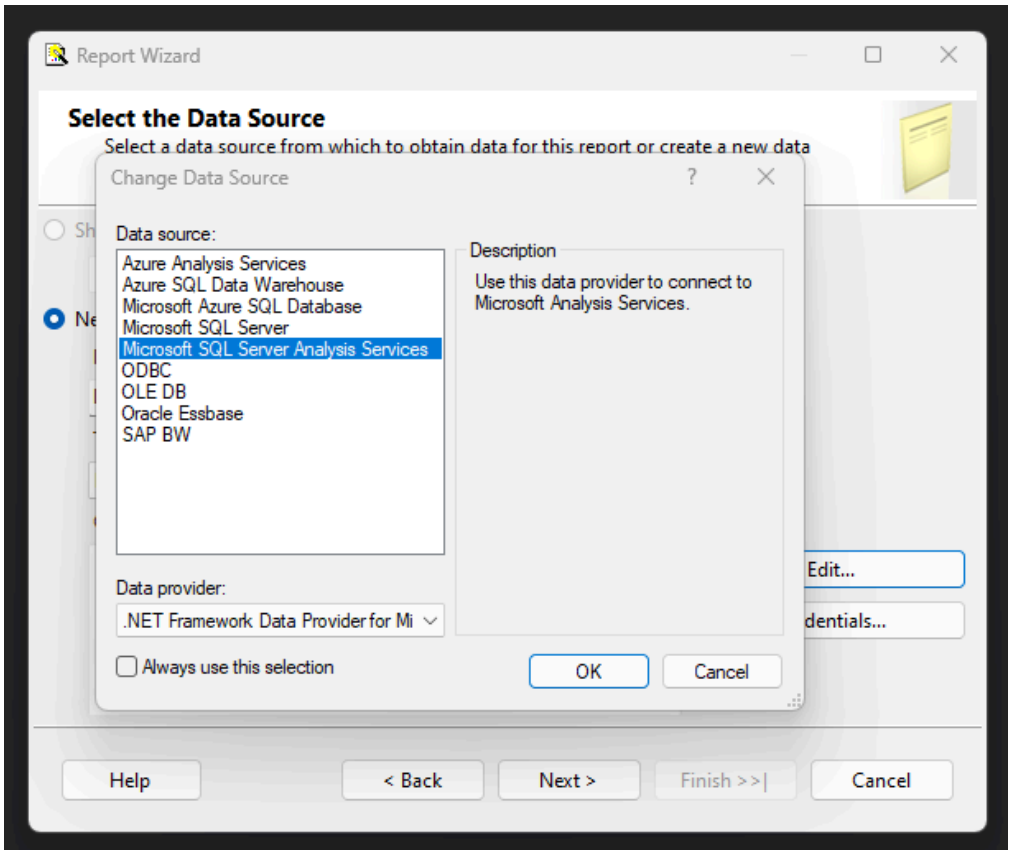


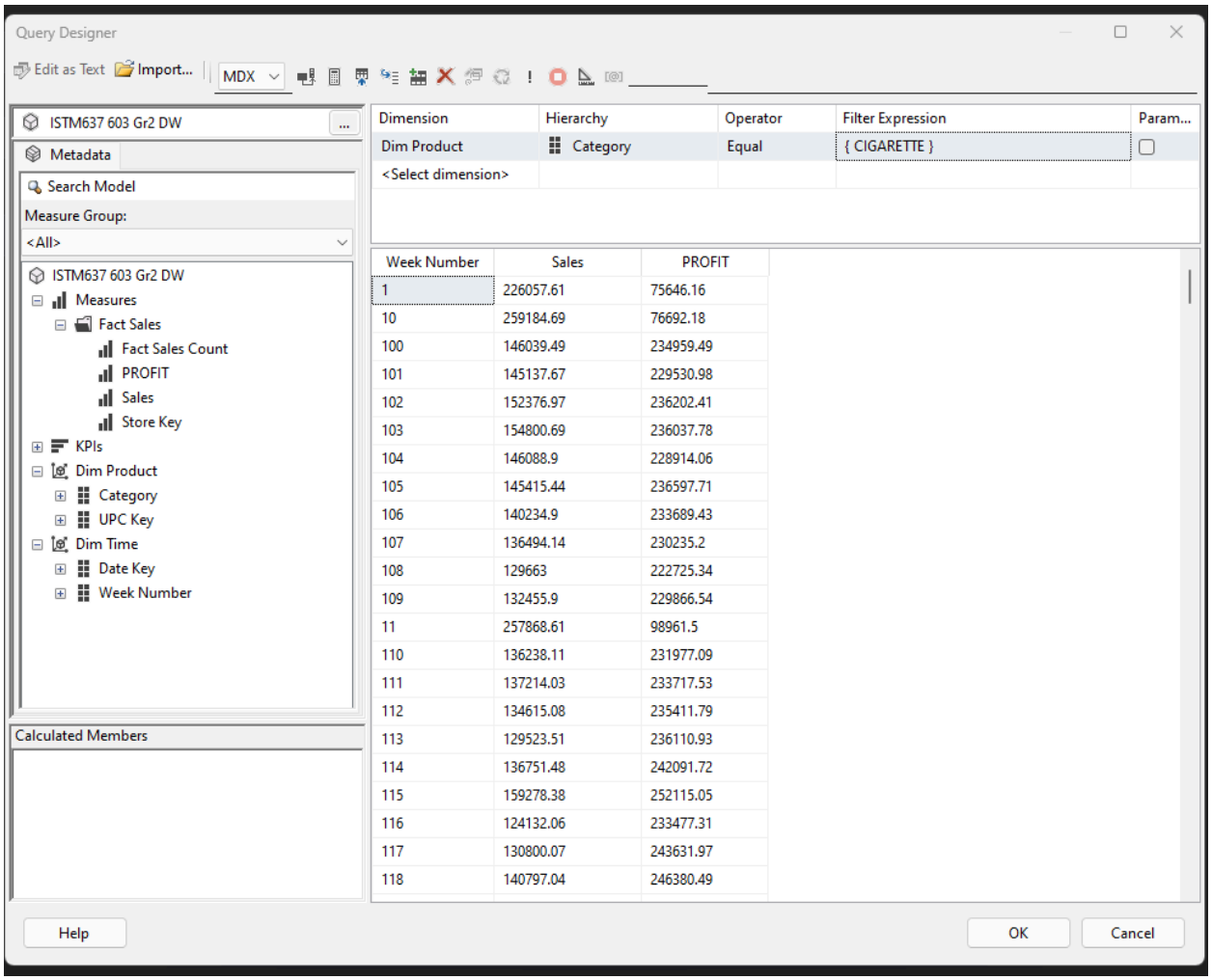
Verify the report generated



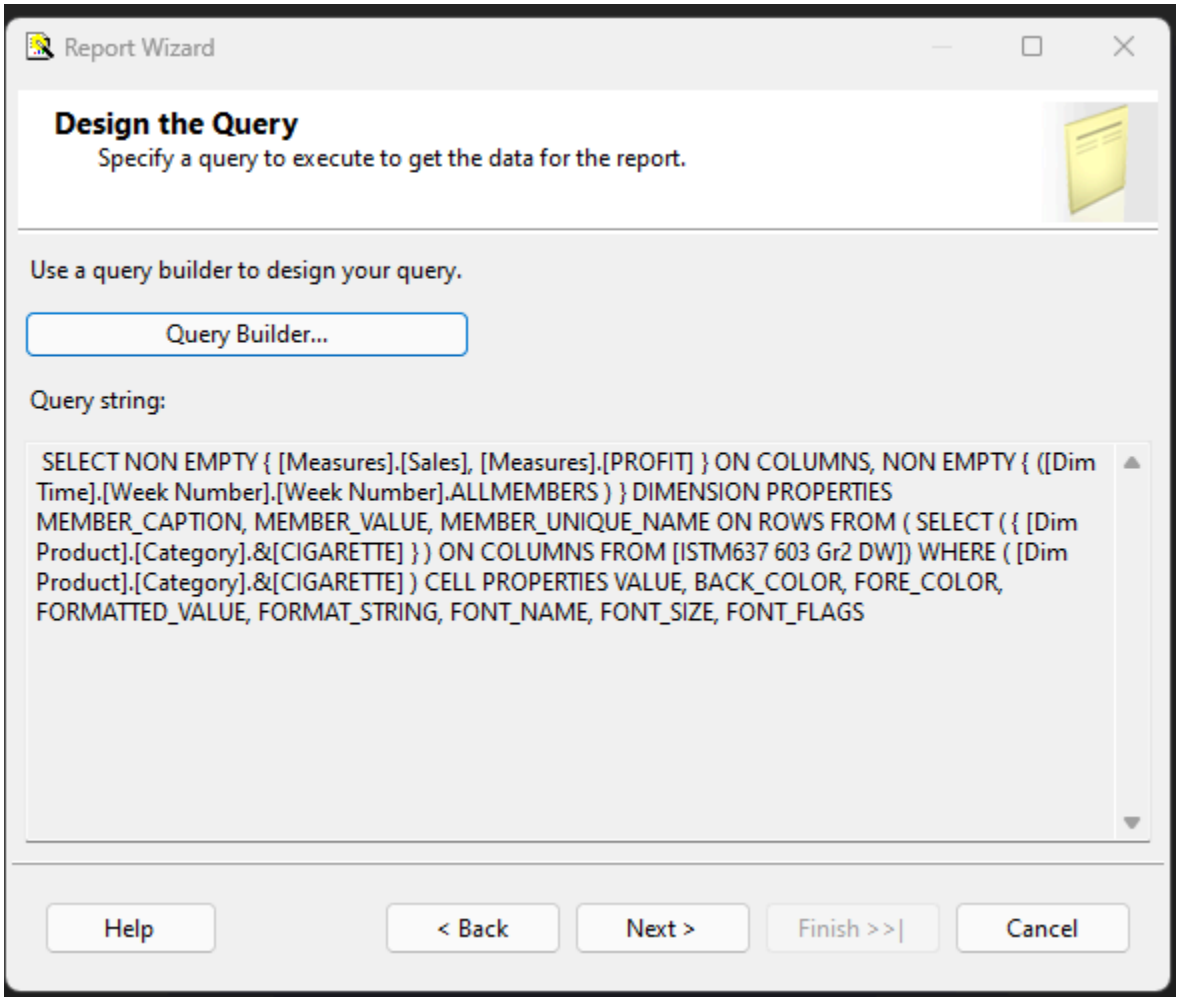
Now move on to SSRS and choose SSAS as data source



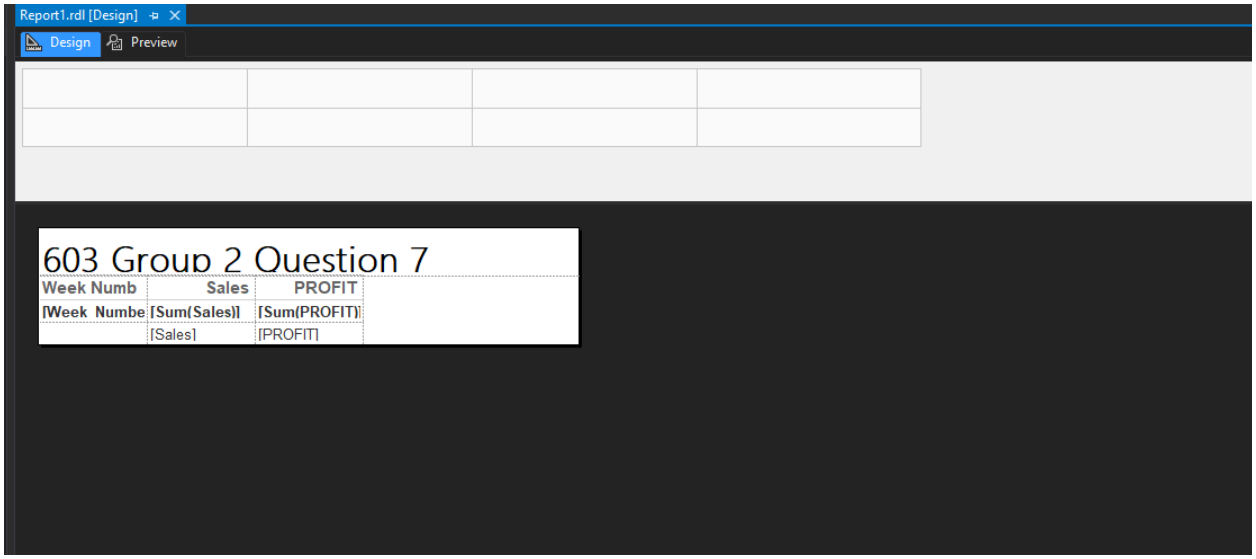
Verify if the source selected is indeed correct



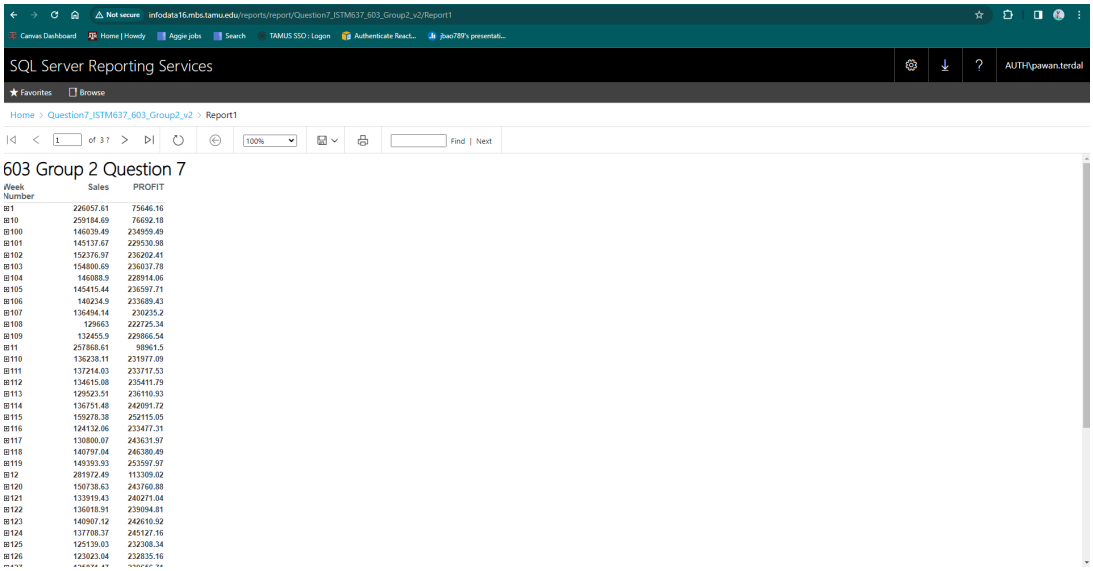
Modify the generated query if needed (Add cigarette filter)




This is the generated report as seen on local machine before deployment



Finally, once the report from SSRS is deployed, this is how it looks



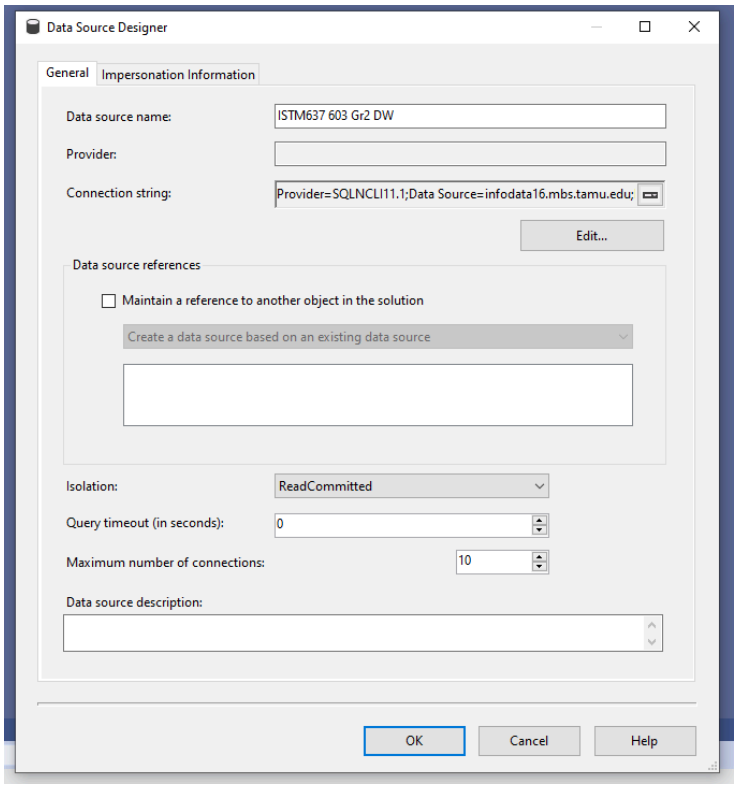

603 Group 2 Question 7

| Week Number | Sales | PROFIT |
|---|---|---|
| B1 | 226057.61 | 75646.16 |
| B10 | 259184.69 | 76692.18 |
| B100 | 146039.49 | 234959.49 |
| B101 | 145137.67 | 229530.98 |
| B102 | 152376.97 | 236202.41 |
| B103 | 154800.69 | 236037.78 |
| B104 | 146088.9 | 228914.06 |
| B105 | 145415.44 | 236597.71 |
| B106 | 140234.9 | 233689.43 |
| B107 | 136494.14 | 230235.2 |
| B108 | 129663 | 222725.34 |
| B109 | 132455.9 | 229866.54 |
| B11 | 257868.61 | 98961.5 |
| B110 | 136238.11 | 231977.09 |
| B111 | 137214.03 | 233717.53 |
| B112 | 134615.08 | 235411.79 |
| B113 | 129523.51 | 236110.93 |
| B114 | 136751.48 | 242091.72 |
| B115 | 159278.38 | 252115.05 |
| B116 | 124132.06 | 233477.31 |
| B117 | 130800.07 | 243631.97 |
| B118 | 140797.04 | 246380.49 |
| B119 | 149393.93 | 253597.97 |
| B12 | 281972.49 | 113309.02 |
| B120 | 150738.63 | 243760.88 |
| B121 | 133919.43 | 240271.04 |
| B122 | 136018.91 | 239094.81 |
| B123 | 140907.12 | 242610.92 |
| B124 | 137708.37 | 245127.16 |
| B125 | 125139.03 | 232308.34 |
| B126 | 123623.04 | 232835.16 |

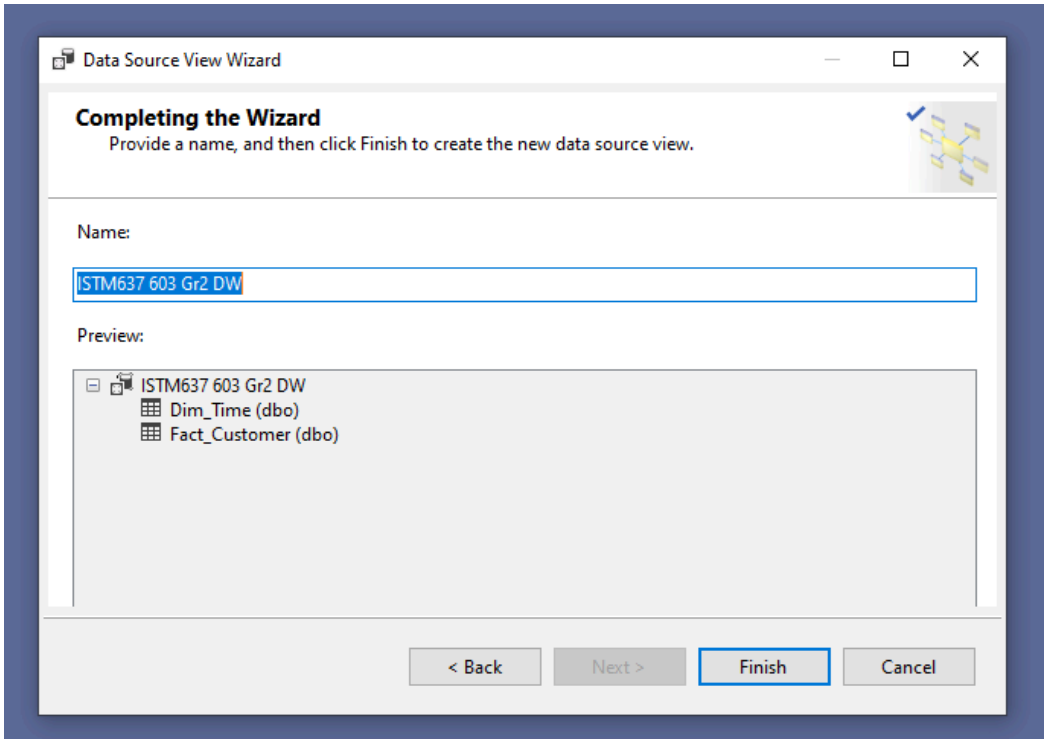
## 5.2.5 SSAS

**BQ5: What were the wine sales for different stores in a given month?**

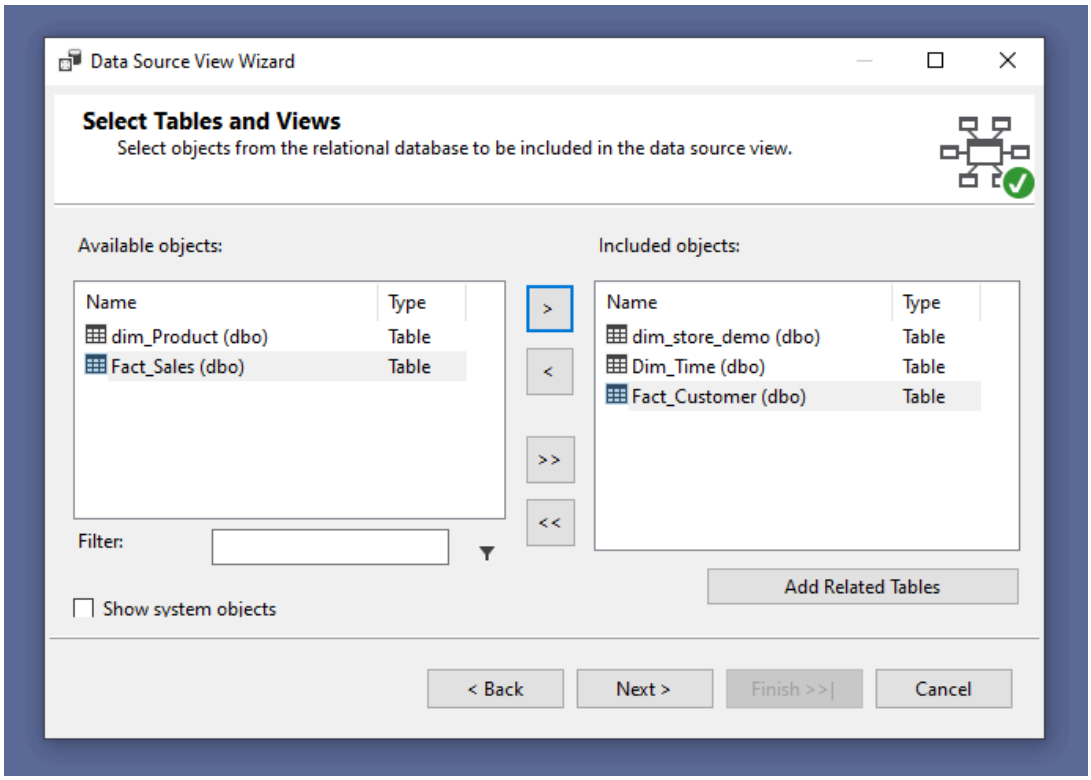
Choose the data source



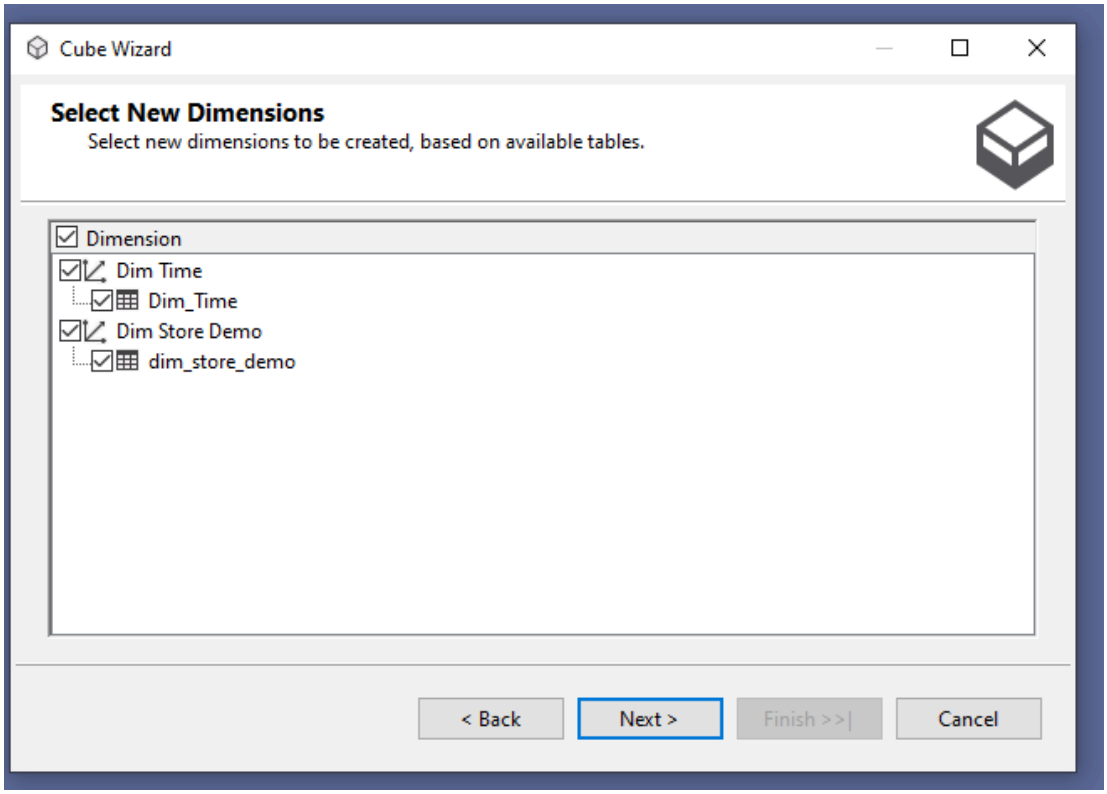
Create Data Source View



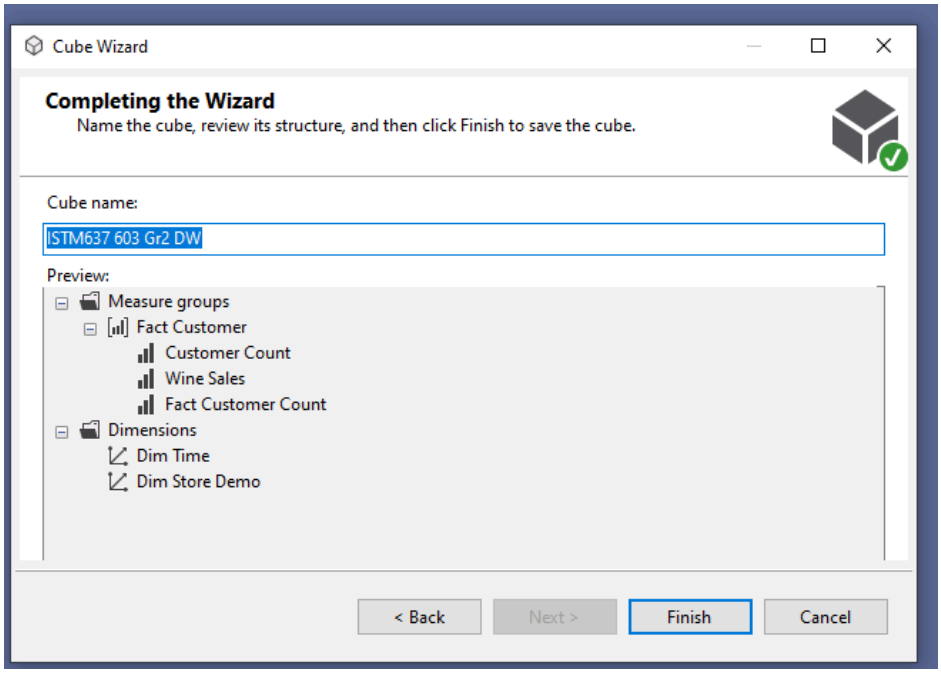

Choose the appropriate tables needed to answer the questions



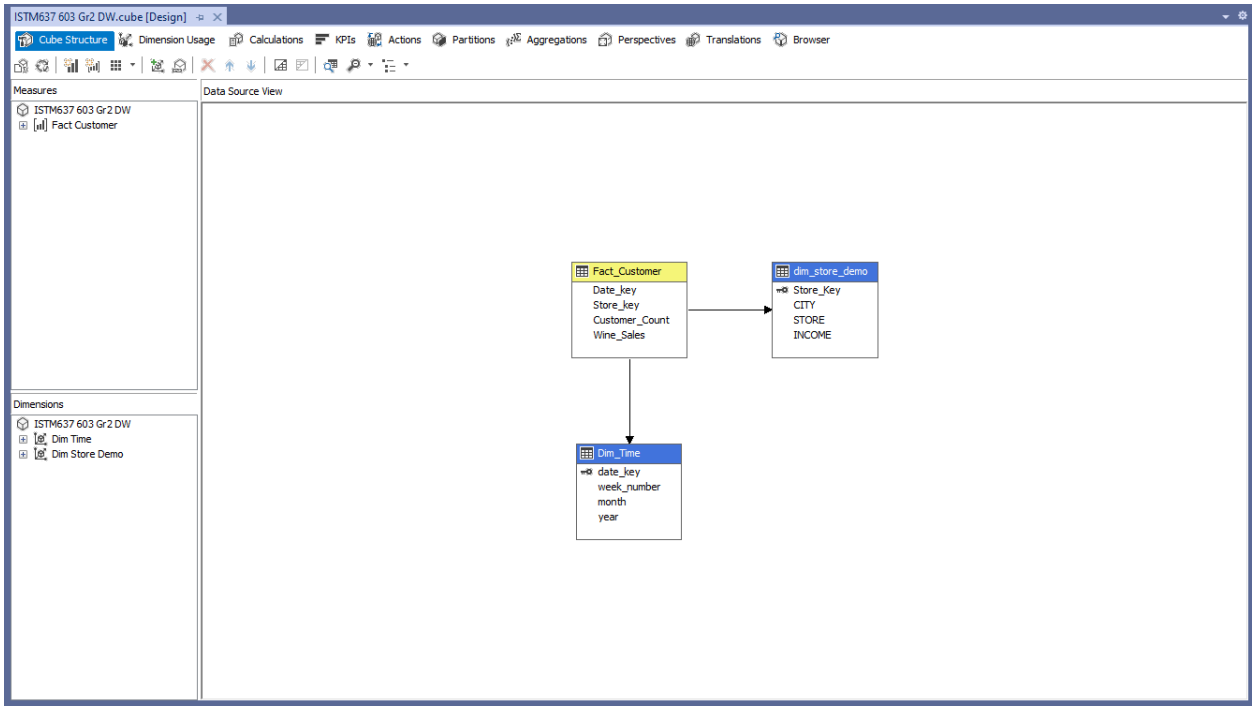

Select the dimensions required for the data mart



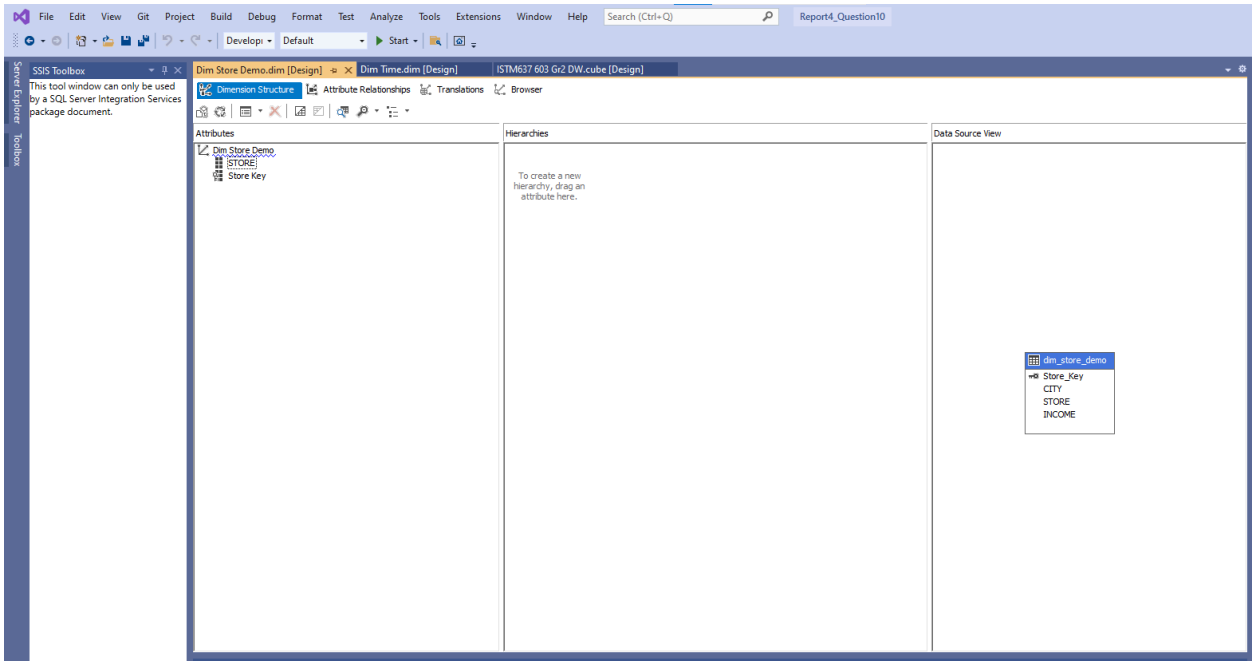

Give the cube an appropriate name and click finish



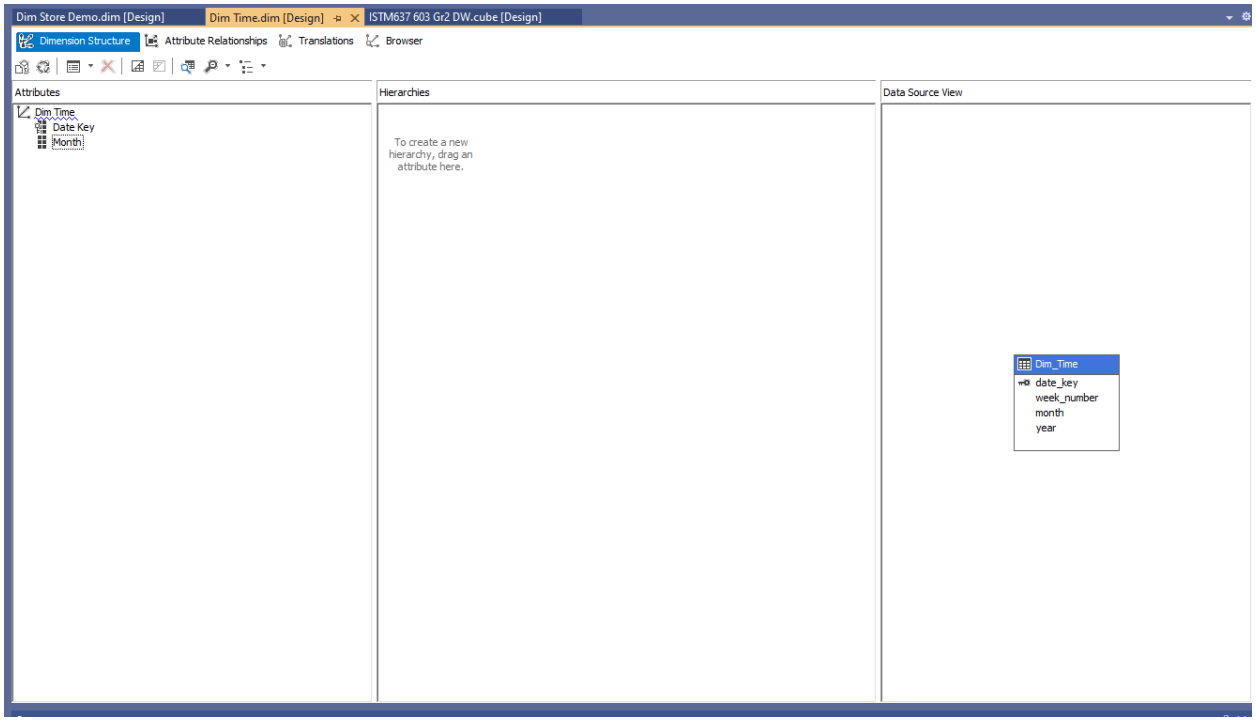

The structure of the cube is as shown below



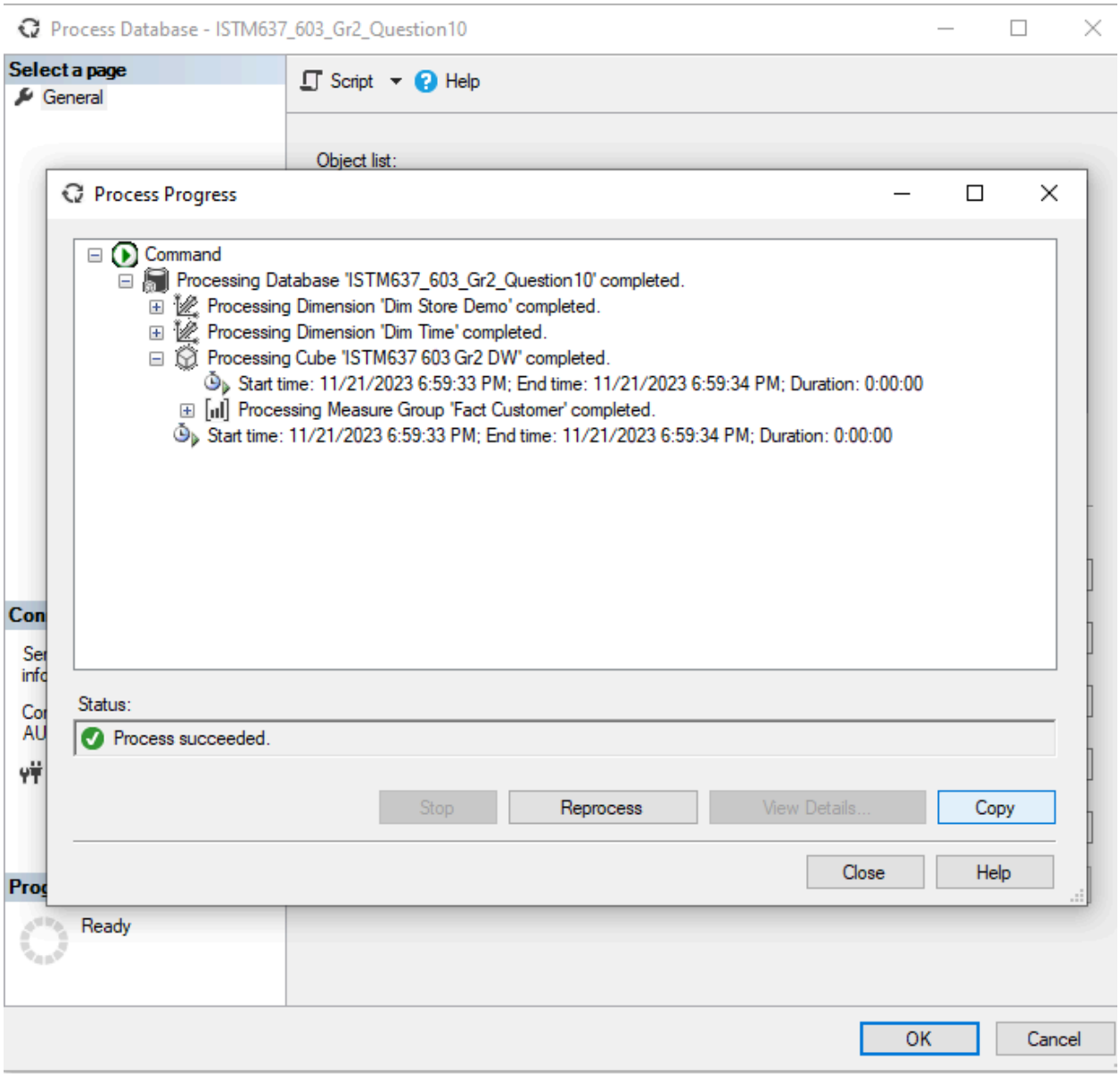

Open the Dim_Store_Demo dimension and drag the fields that are required to be viewed in the report into the Attributes.



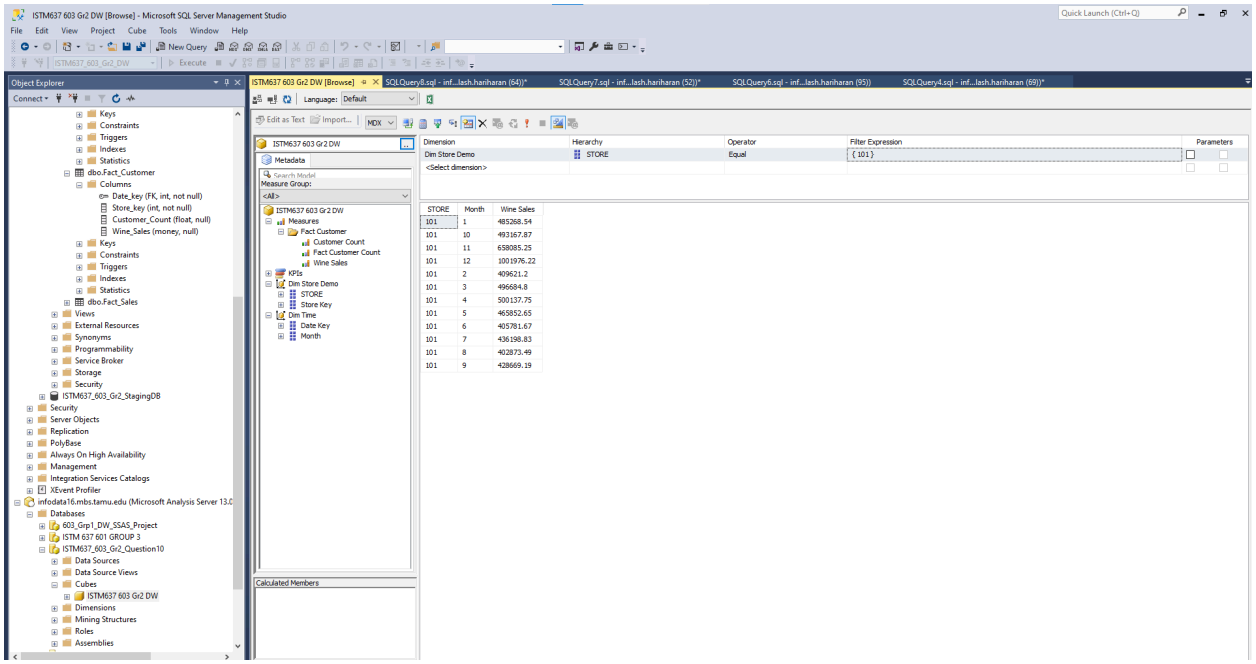
Open the Dim_Time dimension and drag Month into Attributes so that it can be viewed in the report.




Deploy the Cube so that it can be viewed on the server




The cube on the server can be browsed as shown below. A filter can be added to check the details of one store in particular.


ISTM637 603 Gr2 DW [Browse] - Microsoft SQL Server Management Studio
File Edit View Project Cube Tools Window Help
New Query
Execute
Object Explorer
Connect
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fact_Customer
Columns
Date_key (FK, int, not null)
Store_key (int, not null)
Customer_Count (float, null)
Wine_Sales (money, null)
dbo.Fact_Sales
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
ISTM637_603_Gr2_StagingDB
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
XEvent Profiler
infodata16.mbs.tamu.edu (Microsoft Analysis Server 13.0
Databases
603_Grp1_DW_SSAS_Project
ISTM 637 601 GROUP 3
ISTM637_603_Gr2_Question10
Data Sources
Data Source Views
Cubes
ISTM637 603 Gr2 DW
Dimensions
Mining Structures
Roles
Assemblies
ISTM637 603 Gr2 DW [Browse]
SQLQuery8.sql - inf...lash.hariharan (64))*
SQLQuery7.sql - inf...lash.hariharan (52))*
SQLQuery6.sql - inf...lash.hariharan (95))
SQLQuery4.sql - inf...lash.hariharan (69))*
Language: Default
Edit as Text
Import...
MDX
ISTM637 603 Gr2 DW
Metadata
Search Model
Measure Group:
<All>
ISTM637 603 Gr2 DW
Measures
Fact Customer
Customer Count
Fact Customer Count
Wine Sales
KPIs
Dim Store Demo
STORE
Store Key
Dim Time
Date Key
Month
Calculated Members
Dimension
Hierarchy
Operator
Filter Expression
Parameters
Dim Store Demo
STORE
Equal
{ 101 }
<Select dimension>
STORE
Month
Wine Sales
101 1 485268.54
101 10 493167.87
101 11 658085.25
101 12 1001976.22
101 2 409621.2
101 3 496684.8
101 4 500137.75
101 5 465852.65
101 6 405781.67
101 7 436198.83
101 8 402873.49
101 9 428669.19

# References


1. “The Geography of Brands” December 01, 2006 https://www.chicagobooth.edu/review/geography-brands
2. Sivakumar, K., "Quality-Tier Competition and Optimal Pricing," Journal of Business Research 33 (1995): 251-260.https://www.sciencedirect.com/science/article/pii/014829639400074O
3. Tao Huang, Robert Fildes, and Didier Soopramanien. "The Value of Competitive Information in Forecasting FMCG Retail Product Sales and the Variable Selection Problem." European Journal of Operational Research (2014).
4. Dominick’s Data Manual, July 2013
5. https://ourworldindata.org/grapher/sales-of-cigarettes-per-adult-per-day?time=1989..latest&facet=entity
6. Paulraj Ponniah - Data Warehousing Fundamentals for IT Professionals, Second Edition (2010)
7. Naeem, Tehreem. “Data Warehouse Concepts: Kimball vs. Inmon Approach.” *Astera Software*, 27 September 2023 https://www.astera.com/type/blog/data-warehouse-concepts/
8. “Building a Kimball Data Warehouse – SQLServerCentral.” *SQLServerCentral*, 14 June 2023, https://www.sqlservercentral.com/articles/building-a-kimball-data-warehouse
9. https://www.ibm.com/docs/en/ida/9.1?topic=flow-physical-design-considerations
10. What is SQL Server Reporting Services? A definition from WhatIs.com
11. https://learn.microsoft.com/en-us/system-center/scsm/olap-cubes-overview?view=sc-sm-2022