Dominick's Fine Foods

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

COMPREHENSIVE REPORT

Rahul Vincent Gerald
Tushara Chigicherla Kamalakar
Vani Shivanand Kambi

Email: vanisk@tamu.edu
SECTION 602; GROUP2

Login Details

Staging Database: 602_Group2_StagingDB

Credentials: Username = vi5275; Password = Mays5275

Data Warehouse Database: 602_Group2_DataWarehouseDB

Credentials: Username = vi5275; Password = Mays5275

Server URL for reports: http://infodata16.mbs.tamu.edu/ReportServer

Table of Contents

Introduction

The retail industry is a quickly transforming, rapidly evolving and a competitive industry. In order to be the top players in the industry, companies must analyze the changing market dynamics and adapt to the changing trends. These companies generate a lot of data pertaining to sales, customer specific data, etc. All of this data can be utilized to analyze trends and help the companies change their operational or market strategy to improve business. This is where [1] data warehousing and analytics come into picture. Companies can utilize data warehousing and analytics to determine trends specific to their business, which would help them enhance their business and better serve their customers. There are several benefits [1] in making use of data warehousing for improving business strategy and service. We look at Dominick's Finer Foods and see how the business can benefit by data warehousing services.

The Dominick's Finer Foods was a chain of grocery stores located at Chicago. It was founded in 1918 and by 1998 the chain had established 116 stores. The chain had 83 store locations and was considered one of the best grocery stores in the Chicago region. It however closed its doors on December 28th, 2013 citing lack of sales and performance as the reasons. The store consisted of products covering spanning over multiple categories and brands. The data set used in this project has been taken from [2].

The data set consists of historical store-level data about sales and retail margins for 5 years (1989 to 1994). We made use of the various data consisting of products sold, store demographics, customer details, etc. Our group utilizes this data to help DFF answer a few questions pertaining to their business, thus helping them improve their profits and customer reach. By answering the chosen business questions, we come up with facts and figures that can be used by DFF to make crucial decisions that directly impact success of their business. Through this project, we showcase how data warehousing can be used to help companies analyze their business over several dimensions. We made use of their store level data to set up a data warehouse and help them analyze it by answering important business questions.

The major problems and tasks with DFF data set has that need to be overcome and implemented through this project are:

i. Not all the data in the files are in the right format, the data needs to first be cleaned before any analysis can be made on it. Example: a few dates are not in the same format or are missing, store codes are missing in the demographics file, etc.
ii. The data in some files are redundant which can be ignored in the analysis phase. We only loaded the data that was relevant to our business questions. For instance – we only used a few files from UPC and Movement.

iii. There are a few missing entries in a few files which could lead to incorrect analysis, these missing entries were taken care of during the analysis phase. We deleted entries that were violating referential integrity constraint.
iv. There are incomprehensible values in a few files which need to be ignored while making our analysis. For instance – Store ID and data have junk values in CCount and demographics files, we ignored entries with such values.
v. The sales have missing values which if not ignored will provide incorrect values in the reports generated.
vi. Integrating the data required to answer the business questions into a single server was a major task. This was accomplished with Visual Studio and SQL Server Management Studio.
vii. Querying for the exact data required to answer a business question in a report was also one of the major tasks in the project. This was accomplished with Report Builder, SSRS and SSAS.

Domain Understanding

Dominick is a retail chain in the Chicago area. The given data set has all the store level scanner data through years [2]. The data contains information about all of the different categories in the project. In this report we look at the data in detail – CSV, text and HTML files and the metadata – which gives more information about the data contained in movement, customer count, demographics and UPC – how the data is stored, the attributes in each of the tables. Understanding the business processes using the above data and creating Entity Relationship Diagrams (ERDs) corresponding to our understanding. We refine and analyze the data to assess the business. We examine demands for various kinds of products and come up with relevant business questions that help Dominick Fine Foods (DFF).

There is always challenges with examining such (retail) data [3] [4]. Firstly, the amount of data involved in retail businesses is huge. It is not just about the size of the data, the accuracy of the data. Think about all the fluff (unwanted data or noise) that could be in the data. It is important that we know which data matters, and which do not matter. The way the data becomes relevant to the business. Customers can be unpredictable, and we have no way to know what data they would want to query for. Hence the way the data is collected and stored is critical. It is a deciding factor for what data can be analyzed and queried for. Collecting and summarizing data from various business processes in a meaningful way is an essential task. For example, to find most sold product in a given time period becomes difficult if the data about the time dimension and sales dimension are not available or are not available in the required format.

The next challenge commonly faced by any industry today is data security. All data be it retail, or marketing needs to comply with the General Data Protection Regulation (GDPR). The challenge here is with the way the data is stored. If there are not enough security regulations in place, it would be a cake walk for attackers to gain access to confidential data. They (attackers) can use the data thus gained for unauthorized purposes. For example, private customer information can be accessed. A competitor in the field can gain access to the sales details and plan to take down DFF. The repercussions would definitely be severe. All these means that we need to keep the data secure, this can be achieved by regular security audits, policies and regulations. This is an important hurdle for any business.

Next comes the use of technology to gather data about say – forecast sales, actual sales, customer demands, etc. All this data needs to be gathered carefully. Human intervention might be necessary at times, however when using a software to track these metrics, it is important that there is some set rubric that calculates this with minimum error. If not, it becomes an overhead on the sales personnel which may induce unwanted human error and delay the data gathering process making it totally irrelevant. The veracity of the data could be in danger here.

Next comes drawing the right conclusions from the available data. A data analytics tool helps us with the extraction, transformation and loading of data. This task is commonly thought to be as easy and doable in a short period of time, however it all depends on the type of data available to us and the also the kind of analysis we want to perform on the data. The trends or patterns that would help the business. The time taken by this process is most commonly underestimated. The act of drawing insights and acting on the derived analysis would need to go through a lot of approval cycles which means lost time. This might cause the entire thing to be useless, time lost cannot be regained. When the required actions are not taken at the right times, the business might not see the required result and moreover the whole act can negatively affect the business.

The other most important factor that can sometimes be a challenge is earning the trust of the customers to let businesses capture their data – demographics, sale interests, product affinity etc. The customers would need to believe in the business to not to misuse their data. It is important to obtain customer consent before data collection and provide them the assurance of data privacy.

The challenges relevant to Dominick's business include but is not limited to –

a. Cleansing and consolidating data from disparate sources. Taking into account the mere size of the data.

b. The integrity, security and compliance of the collected data.

c. Use of technology and third-party resources to collect and analyze data. Also, to ensure the veracity of the data.

d. Drawing relevant conclusions from the available data. Not underestimating the time this process takes. Visualization with respect to data exploration, decision making and communication.

e. Checking whether there is customer consent.

These are the few challenges that our team found relevant to Dominick data. However, the main challenges were with processing the data –keeping relevant or eliminating irrelevant data from the dataset for better visualization. Deciding on which data was most relevant or irrelevant was also crucial.

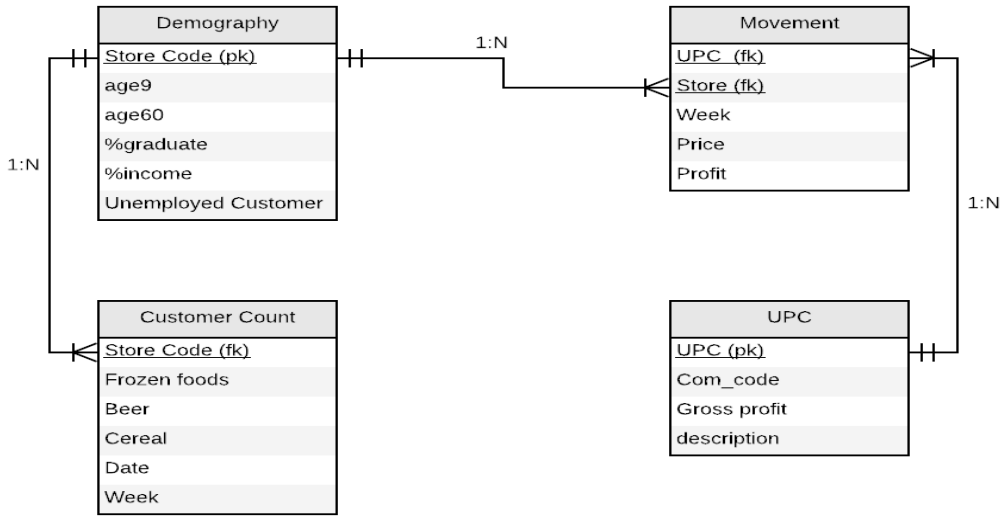
Data Description

Details about DFF Data

The DFF data set is primarily made up of sales data of more than 3500 products over all store locations. This data is from over 100 stores spread across the country. The data is collected from daily transactions at these stores. Data about customers and their preferences is also captured. Most of the data is redundant and contains lot of junk values, it must be cleaned before proceeding with the analysis. The size of the data set is around 4.76 Gigabytes. At a high level the data constitutes two kinds of files – general and category specific. The general files contain customer specific information and store demographics information. The category specific files contain UPC and Movement data.

There are four major files, each consisting of data about a different entity. These files in the data set are as follows:

1. **Customer Count Files** – contains data about coupon usage and store traffic at each store location for each day. It gives information about the number of customers visiting the store and buying a product. This is important in the analysis and preparing reports regarding the total number of customers visiting a store and also the total coupons redeemed. Customer count file also has data about the product and the price paid for that product in a given sale.
2. **Demographics** – this file contains data about the location of each store as well as statistics about the surrounding population. Example: percentage of citizens above the age of 60. This data is valuable in building reports as to what type of people visited a store in a region and comparing the results with another region. Demographic data can be used to answer business questions regarding a target audience and how to increase sales for a given type of population or audience.
3. **UPC Files** – there many categories of products sold at DFF. Each product under a given category is stored in a UPC file with XXX denoting the category of the products. DFF also uses two of its own variables to track products. It uses a variable called commodity code which can be present and unique to a single file (ie. Unique to a category). It also uses a variable called item code which can be taken as a variable to track newer versions of a product in a given category.
4. **Movement** – this file contains sales data for each UPC (product) for a given week. It gives information about the units sold, profit margin, price etc. The data can be used to get information regarding how many products were sold in a specific week for a particular store. This is helpful for higher management to compare which store is doing better than the rest.



We decided to use the following ERD to construct a staging database using which many transformation operations will be performed on the data before loading it into data warehouse.

Metadata

Metadata for all the OLTP Files from source

1) Customer Count Files:

The customer count file contains information regarding in-store traffic. The data contained in these files on a daily basis is store specific. The number of customers visiting the store and purchasing something. This file also contains the total dollar sales and total coupons redeemed details, by DFF defined department. The details of the CCOUNT file variables are described below:

CCOUNT			
Variable	**Description**	**Type**	**Length**
DATE	Date of the Observation	Character	6
Week	Week Number	Numeric	8
Store	Store Code	Numeric	8

BAKCOUP	Bakery Coupons Redeemed	Numeric	8
BAKERY	Bakery Sales in Dollars	Numeric	8
BEER	Beer Sales in Dollars	Numeric	8
BOTTLE	Bottle Sales in Dollars	Numeric	8
BULK	Bulk Sales in Dollars	Numeric	8
BULKCOUP	Bulk Coupons Redeemed	Numeric	8
CAMERA	Camera Sales in Dollars	Numeric	8
CHEESE	Cheese Sales in Dollars	Numeric	8
CONVFOOD	Conventional Foods Sales in Dollars	Numeric	8
COSMCOUP	Cosmetics Coupons Redeemed	Numeric	8
COSMETIC	Cosmetics Sales in Dollars	Numeric	8
CUSTCOUN	Customer Count	Numeric	8
DAIRCOUP	Dairy Coupons Redeemed	Numeric	8
DAIRY	Dairy Sales in Dollars	Numeric	8
DELI	Deli Sales in Dollars	Numeric	8
DELICOUP	Deli Coupons Redeemed	Numeric	8
DELIEXPR	Deli Express Sales in Dollars	Numeric	8
DELISELF	Deli Self Service Sales in Dollars	Numeric	8

FISH	Fish Sales in Dollars	Numeric	8
FISHCOUP	Fish Coupons Redeemed	Numeric	8
FLORAL	Floral Sales in Dollars	Numeric	8
FLORCOUP	Floral Coupons Redeemed	Numeric	8
FROZCOUP	Frozen Items Coupons Redeemed	Numeric	8
FROZEN	Frozen Items Sales	Numeric	8
FTGCCOUP	Food-to-Go Coupons Redeemed	Numeric	8
FTGCHIN	Food-to-Go Chinese Sales in Dollars	Numeric	8
FTGICOUP	Food-to-Go Coupons Redeemed	Numeric	8
FTGITAL	Food-to-Go Italian Sales in Dollars	Numeric	8
GM	General Merchandise Sales in Dollars	Numeric	8
GMCOUP	General Coupons Redeemed	Numeric	8
GROCCOUP	Grocery Coupons Redeemed	Numeric	8
GROCERY	Grocery Sales in Dollars	Numeric	8
HABA	Health and Beauty Aids Sales in Dollars	Numeric	8
HABACOUP	Health and Beauty Aids Coupons Redeemed	Numeric	8
JEWELRY	Jewelry Sales in Dollars	Numeric	8

LIQCOUP	Liquor Coupons Redeemed	Numeric	8
MANCOUP	Manufacturer Coupons Redeemed	Numeric	8
MEAT	Meat Sales in Dollars	Numeric	8
MEATCOUP	Meat Coupons Redeemed	Numeric	8
MEATFROZ	Meat-Frozen Sales in Dollars	Numeric	8
MISCSCP	Misc. Coupons Redeemed	Numeric	8
MVPCLUB	MVP	Numeric	8
PHARCOUP	Pharmacy Coupons Redeemed	Numeric	8
PHARMACY	Pharmacy Sales in Dollars	Numeric	8
PHOTCOUP	Photo Coupons Redeemed	Numeric	8
PHOTOFIN	Photo	Numeric	8
PRODCOUP	Produce Coupons Redeemed	Numeric	8
PRODUCE	Produce Sales in Dollars	Numeric	8
PROMCOUP	Promotion Coupons Redeemed	Numeric	8
PROMO	Promotion Sales in Dollars	Numeric	8
SALADBAR	Salad Bar Sales in Dollars	Numeric	8
SALCOUP	Salad Coupons Redeemed	Numeric	8

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

2) Store Specific Demographics:

The demographics file consists of store-specific demographic data. The data originally comes from U.S. government (1990) census data for the Chicago metropolitan area. Market Metrics processed this data to generate demographic profiles for each of the DFF stores. The table below gives the descriptions for all the files in the demographics database. The demographics file contains the following variables:

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
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)

3) Category Descriptions:

Category descriptions include the list of all UPC's and their description and size for a particular category. For example, under the analgesics category we can have pain relievers can be aspirin, ibuprofen, or acetaminophen base.

4) Week's Decode Table:

This data is part of Dominick's manual and Codebook document. These files consist of information on the codes that correspond to the week when sales data was recorded. This table is particularly useful in comparison of sales trends between holiday and non-holiday weeks. The data is this table could also be of essence in evaluating monthly, quarterly or semi-annual sales reports.

5) UPC Files:

The UPC files contains information of UPC belonging to each category. The files are named upcxxx, where xxx is the three-letter acronym for the category. Information extracted can be Product Name, Size, UPC Number etc. The variables in this file are described below:

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

6) Movement Data by UPC:

These files contain the information that are crucial in answering business questions pertaining to holiday sales, demographic-level sales etc. These files contain weekly sales data of the products belonging to different categories. The files are named wxxx where xxx is the three- letter acronym for the category. The variable details for this file are as follows:

Movement of Data by UPC

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

Business Questions

1. **Determine average gross profit margin of frozen products across stores and verify which stores are below average and plot the same?**

Justification: Finding Gross Profit any product in any retail industry is crucial for the executives of the company to understand the benefits of marketing and/or selling any category of products. As the company grows, the inventory and accordingly the number of categories of products grow. This is a result of continuously growing industry. To ensure a constant growth, executives make certain decisions in regard to products and their categories periodically to make sure the company makes profit. Determining average gross profit margin of frozen products across stores will help Dominick's analysis on company's current statistics on the profit made by selling the products in a category and how it differs from store to store. Hence this is a highest priority business question as frozen goods demand comparatively higher investments and storage facilities.

2. **What is the average sales for beer in the last 3 years?**

Justification: Depending on the seasons and the time of the year, certain products have additional sales and hence a profit. It is important to make a decision to promote correct products at any period of the year so as to attract the right customer base and thus fulfilling their needs without falling short of supplies. For instance, summer is the peak period for beer and other beverages and so are holidays and celebrated days in a year. Thus, it is recommended for the company to analyze their beer consumption across various months to increase the availability of this category across stores to ensure a perfect sales record.

3. **What is the growth of Bakery from the year 1990 to 1996?**

Justification: A retail industry needs a constant analysis of products and their categories based on regions to understand the customer base for a certain product category. America is known for its bakery food items and their delicious temptations that any customer can never avoid. Hence it is apparent to analyze the bakery category sales and it's growth for the past years to deploy strategies to avail more products and supplies in any store at any period of time. In order to understand this growth DFF needs a growth analysis of bakery product category for a certain period of time, in this case from 1990 to 1996, to contribute towards understanding the sales of the bakery products. This will help the company extract the growth to forecast the demands in the industry within a 5 year period and thus to check if the demand is shrinking or rising in those 5 years and if the product needs to be marketed more into the stores for additional sales using more marketing strategies.

4. What is the average sale per store for cereal for the past year?

Justification: Cereal is considered to be the go to morning breakfast item in most of the American households. This product can even be considered the basic product to be purchased by any customer. Basic product categories help keep the company's sales and revenue at a reasonably constant sales average. Thus inspecting sales for such basic products every year will help the company track the irregularities so that proper measures can be enforced to build marketing strategies to promote such products and provide sufficient inventory management.

5. What is the store count in US?

Justification: Every company starts small, but over the period of time, expansion can be hectic for the management to keep track of every store in every location. To know the number of stores in a country at any point in time is important for the CEO to enhance their productivity. Basic question like this carries medium weightage but higher priority for a growing industry to keep track of changes to its expansion schemes.

6. How many graduates and above are customers at any store?

Justification: Every store in every region has its own dominant customer base. For instance, a small town like college station will have majority of its customers as students. And hence understanding the customer is the key to any service-based and product-based industry. This will provide the company with enough knowledge on the needs of the customers and the products they regularly prefer. Knowing the average educational status of the customers will help the company promote relevant product categories.

7. Which of the top 5 stores have customers with highest % income?

Justification: For a largely spread out company, it is imperative for the higher management and sales specialists to understand the customer base for each and every store. Every region is different in a country and so are the needs of the customers in those regions. Economy is another key factor to understanding the average expenses of customers in a region. For instance, College Station has majority of its customer base as students and students who earn comparatively less and are bound to student loans, prefer affordable and quality products. In order to build the trust and foundation of a customer dependent industry, it is important to know the customer's economic conditions and accordingly promote the product categories that they can afford and are satisfied with.

8. Top 5 products sold in the last year?

Justification: Any retail industry will contain a large list of inventory items and products. Finding the right vendors for a product to ensure customer satisfaction with the product is an important task for the company executives. In order to determine which product is in more demand and requires more vendor support or inventory management assistance, it is crucial to analyze the top products that were sold in the last year so as to implement the necessary precautions and actions in place to ensure that those products are never short in supplies and also have good vendor support to provide variety. Not every product carries equal importance in the stores and based on this fact, determining the products which have been popularly purchased will help the company invest in similar categories to attain higher profit margins.

9. Revenue for last year?

Justification: Revenue management is an important part of the job efficiently carried out by sales specialists and auditors which is then scrutinized by the executives to understand the current worth of the company as compared to competitors in the market. Every company relies on revenue to device future tactics for the company. It is a primary requirement to keep the revenue in check so that the company doesn't go bankrupt.

10. Which stores/cities have highest poverty %?
Justification: Every region is different in a country and so are the needs of the customers in those regions. Economy is another key factor to understanding the average expenses of customers in a region. For instance, College Station has majority of its customer base as students and students who earn comparatively less and are bound to student loans, prefer affordable and quality products. Customers have different interests and affordability at any location. To make sure every customer gets what they want always, it's important to know the economical standards at the location of the stores to make the right products available at all times. This will help build the trust between the customer and the company as well as the brand name.

11. Top 5 stores with highest unemployed customers?
Justification: Every region is different in a country and so are the needs of the customers in those regions. Economy is another key factor to understanding the average expenses of customers in a region. For instance, College Station has majority of it's customer base as students and students who earn comparatively less and are bound to student loans, prefer affordable and quality products. Products that are purchased by categorically different customers will help the company in carrying out effective inventory management and hence implementing equivalent promotions for marketing the necessary products.

12. Average graduate customers per store of all the stores?
Justification: Every store in every region has its own dominant customer base. For instance, a small town like college station will have majority of its customers as students. Now based on the hierarchy of education can cause different opnions towards a product category and thus changing the demands for the same. And hence understanding the customer is the key to the success of any retail industry. This will provide the company with enough knowledge on the needs of the graduate customers and the products they regularly choose to buy. Knowing the average educational status of the customers will help the company promote relevant product categories and increase the sales for the same which in turn will affect the gross profit.

Off these twelve questions, questions 1, 2, 3, 7 and 8 were chosen for us to work on.

Dimensional Modeling

Dimension models facilitate answering business questions based on querying and analyzing the data. It is important that we recognize what is important for the analysis and what can be discarded, what data plays a major role in answering the BQ and how to simplify this process. All of these can be accomplished with dimensional models. They give us the logical structure for the data warehouse, which also helps the customer in understanding the underlying process easily. A dimensional model should provide an efficient way to access data, must be optimized for queries and analysis, and must show how dimensional tables interact with fact tables and must allow drill down or roll up along dimensional hierarchies. Dimensional models capture all the critical metrics along multiple dimensions and is easy to understand for the business users.

The following are the major tasks involved in dimensional modeling phase:

a. Selecting the process – selecting the entities based on the requirements we gathered in report1. These are the first structures that need to be designed.
b. Choosing the grain – we need to keep in mind the level of detail that will be needed not just for now but in future as well. So, we had to plan ahead and include the most appropriate granularity for the data.
c. Identifying and conforming dimensions – based on the business questions we chose suitable dimensions that would help us fetch the desired results. These are the dimensions for the entities chosen in the first task. We also need to ensure that data in each of these dimensions are conformed to each other.
d. Choosing the facts – to the measurement metric of interest. For instance – total sales for a particular product.
e. Choosing the duration of the database – this is based on how much historical data we want to examine. That is how far in time will we be looking into to measure the metrics and generate reports.

Dimensional models consist of structures for i) measurement metrics ii) dimensions iii) dimensional attributes. All of these will be represented as entities in our logical model. The measurement metrics or units are nothing but facts we are interested in. Hence, they belong in the fact tables. So, each measurement is an attribute in the fact table. Next, we need to look at the business dimensions that are of significant interest. The dimensions to be chosen, depend on the type of analysis we need to make from the facts. For instance – if we are interested in the total sales, this will be our metric in the fact table, the dimensions we would be interested in would be time – as in daily, weekly, monthly, quarterly, annually etc., sales; product– detergent, food, furniture, etc., These would constitute the business dimensions. Next, we will have queries where the metrics in the fact table will be analyzed along the dimensions in the dimension tables. The fact table is linked to the dimension tables with foreign keys, these can be used as a concatenated primary key or combine the dimension keys to form a compound primary key for the fact table or we can have a generated primary key independent of the dimension keys. In our model we make use of independent keys in fact tables.

When we design the logical model with all of this in mind, we find that the fact table would be in the middle, surrounded by multiple dimension tables. This would resemble a star; hence such

schemas are called STAR schemas. Star schemas can be easily understood by the users and designers alike, it optimizes the navigation through the database – navigation would be simple and easy even for a complex query, it is highly suitable for query processing and analysis because it facilitates easy drill down and roll up for queries.

It is essential for DFF to have a data warehouse in place so that reports can be generated which answer business questions. To construct a data warehouse, Kimball's rules for dimensional modelling can be used to build an efficient warehouse. We followed Kimball's methodology for constructing the dimensional model for our data warehouse. The Kimball rules for a good dimensional model mapped to DFF data are as follows:

Rule 1 – Atomic data in dimensional structures

When data is stored in a dimensional model care should be taken to store the data at its lowest level, the data granularity should be maintained. This is because if summarized data is stored in the data model users will meet a dead end and will not be able to dig deeper and find answers to business questions. By storing Dominick data at its lowest level of granularity it becomes easier to go deeper into the data set and generate reports such as sales over a given area during different times (weekly, monthly, yearly).

Rule 2 – Dimensional models should be structured around business process

When structuring a dimensional model, the business process for which it is being made should be kept in mind. For instance, the fact table used should primarily contain data related to the business process in question. If the Dominick fine foods BQs are related to sales data over a period of time across its branches at Chicago, then the fact table should have fact related to the sales process, like total sales, total profit etc. Here, the business process in question is sales and the model should be structured around this.

Rule 3 – Fact table should have an associated date dimension table

Every fact table must have a foreign key related to a date dimension table in order to keep track of the timestamp. The date dimension allows users to filter out data according to a date, for example; daily, weekly, monthly or yearly. This rule will play an important role in Dominick data as the date dimension can be used to generate reports on the number of coupons which were redeemed at each store over a period of time.

Rule 4 – All facts in a fact table must be at the same level

A fact table contains measures which are used to access and answer business questions, if they are not at the same level it will make it complex for the BI team to create reports. The measures should be in any one of the following levels – transaction, periodic or accumulating. The measures used here would be total sales, total profit etc. If they are not at the same level it will make it complex for the BI team to create reports based on sales of Dominick fine foods over the Chicago area.

Rule 5 – Resolve the many-to-many relationships in fact tables.

When there is a many-to-many relationship between entities, this relationship should be split. The foreign key will be placed in the fact table and a relationship is formed between the fact table and its respective dimension table. Example: one employee can work on may projects and one project can have many employees working on it. In the Dominick data set it is clear that there is a many-to-many relationship between products and the demography (a tore at one location can have many products and a product can be present in many stores). It is important to split this relationship by keeping the foreign key n the fact table and forming a relation with the dimension table of demography and products (UPC).

Rule 6 - Resolve the many-to-one relationships in fact tables.

Similar to rule 5, many-to-one relationships can be resolved by placing the foreign key in the fact table. However, this should be sparingly used. Similar to rule 5, the many-to-one relationships between customer count files and demography can be resolved by placing the foreign key in the fact table. This allows the generation of reports with respect to customer count and demography.

Rule 7 – Filter domain values should be stored in dimension tables.

The dimension tables should not only contain coded data based on the dimension; it should consist of descriptive data too. This makes it easy for the BI team to prepare reports which are understandable and readable. The dimension tables should contain descriptive data about the dimension it is in. For example: the product dimension for Dominick data should contain a description about the product this will make it easy for the BI team to prepare reports regarding product distribution across the different stores which are understandable and readable.

Rule 8 – Surrogate keys should be used in dimension tables.

It is important to uniquely identify the records in the dimension table; hence surrogate keys are used which uniquely identify each row of the dimension table. It is important to use surrogate keys; in Dominick data each dimension table should have a surrogate key so that it will be easy to identify each record from the dimension table. Example: each demographic location from the demography dimension can be uniquely identified.

Rule 9 – Use conformed dimensions across the enterprise.

By using a consistent set of dimensions across the entire enterprise helps making ETL processes execute faster and more efficiently. For the Dominick data, if we use a conformed dimension such as product across the entire enterprise, efficient reports can be generated with respect to the movement of each product across a geographical area for a given time period.

Rule 10 – Requirements should be balanced continuously

Business processes change continuously, and this also makes the business questions asked change. To keep up with this change it is important to change and update the dimension tables regularly. As Dominick fine foods is a large chain spread across the US, it is important to keep up with the ever changing retail industry and update the requirements of the business process. This is one of the most important rules to follow as the business goal will change over a period of time and it is important to adapt to this change.

Designing data marts using Kimball's approach:

Based on the source files and the chosen BQs, these are the dimension and fact matrix that we used to form dimension and fact tables for our data warehouse.

Staging Area Tables	Dimension Tables				Fact Tables			
	dimProduct	dimStore	dimDate	factCategorySales	factProfitMargin	factBakerySales	factHighestIncome	factTopProducts
DEMO$		X		X	X	X	X	X
MOVEMET$	X			X	X			
CCOUNT$		X				X	X	X
UPC$	X			X	X			
DATE			X	X	X	X	X	X

Dimension tables

1. Product dimension: dimProduct

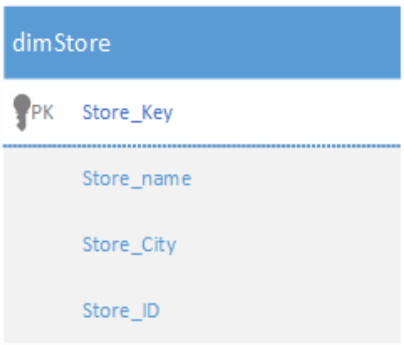
The attributes of the product dimension are summarized as follows:
Product Key: A unique identifier of the product dimension table, that is an autogenerated key.
UPC_ID: The last five digits of this number identifies the product; the remaining digits identify the manufacturer.
Product_Name: Describes the name of the product
Product_Category: Specifies the category that a product belongs to.

The corresponding physical model for dimProduct is

dimProduct

Column Name	Data Type	Nulls Allowed	Comments
Product Key	Numeric	No	Used as a primary key for the product dimension table
UPC_Number	Numeric	No	Used as a primary key to identify a unique product
Product_Name	String	No	A name given to an individual product, need not be unique
Product Category	String	No	The category to which a product belongs to

2. Store dimension: dimStore



The attributes of the store dimension are summarized as follows:
Store Key: A unique identifier for the store dimension table, this is an autogenerated key.
Store_name: This describes the name of the store.
Store_City: This identifies the city where the store is located in.
Store_ID: This is a unique number that is associated with each store under DFF.

The corresponding physical model for dimStore is:

dimStore

Column Name	Data Type	Nulls Allowed	Comments
Store Key	Numeric	No	Used as a primary key for the store dimension table, to uniquely identify each store in the data mart
Store Name	String	No	Gives information about the stores name
Store City	String	No	Demographic information about the city name where the store is
Store Number	Numeric	No	The key used to uniquely identify the store in the source table

3. Time dimension: dimTime

The attributes of the time dimension can be summarized as:
Time Key: A unique identifier for the time dimension table, this is an autogenerated key.
Week: This specifies the week number.
Month: This specifies the month number, with values between 1-12.
Year: This specifies the year.

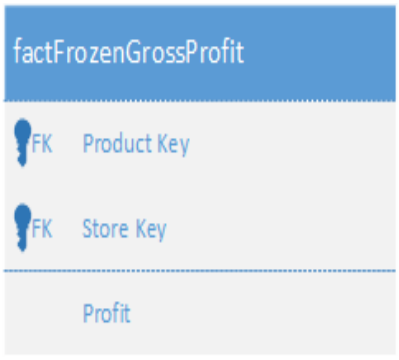
The corresponding physical model for dimTime is:

dimTime

Column Name	Data Type	Nulls Allowed	Comments
Week	Date	No	The week details of a given time
Month	Date	No	The week details of a given time duration
Year	Date	No	The year details of a given time duration

Fact Tables

1. FactFrozenGrossProfit



The description of the attributes in factFrozenGrossProfit can be summarized as:
Fact table Keys -
Product Key: A unique identifier for each of the products, this is a surrogate key from the product dimension - dimProduct.

Store Key: A unique identifier for each of the stores under DFF. This is a surrogate key from the store dimension – dimStore.

Fact table Measures -
Profit: This is a fact table measure that gives the gross profit value for products under frozen category.

The corresponding physical model for factFrozenGrossProfit fact table is:

factFrozenGrossProfit

Column Name	Data Type	Nulls Allowed	Comments
Product Key	Numeric	No	Foreign key, used to reference a row in the product dimension table
Store Key	Numeric	No	Foreign key, used to reference a row in the store dimension table
Profit	Numeric	No	The measure which is used to calculate the gross profit of a given product

2. factBeerSales

The description of the attributes in factBeerSales can be summarized as:

Fact table Keys -
Product Key: A unique identifier for each of the products, this is a surrogate key from the product dimension - dimProduct.
Store Key: A unique identifier for each of the stores under DFF. This is a surrogate key from the store dimension – dimStore.
Time Key: A unique identifier for time. This is a surrogate key from the time dimension – dimTime.

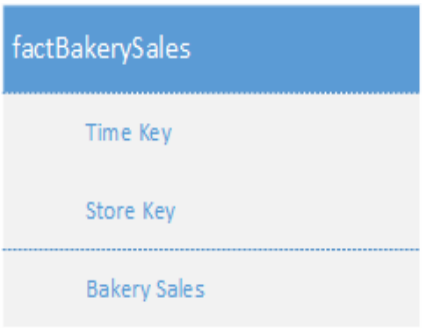
Fact table Measures -
Beer Sales: This is a fact table measure that gives the amount of sales for products under Beer category across stores and different time period.

The corresponding physical model for factBeerSales fact table is:

factBeerSales:

Column Name	Data Type	Nulls Allowed	Comments
Product Key	Numeric	No	Foreign key, used to reference a row in the product dimension table
Store Key	Numeric	No	Foreign key, used to reference a row in the store dimension table
Time Key	Numeric	No	Foreign key, used to reference a date (week, month or year) from the time dimension table
Beer Sales	Numeric	No	The measure which is used to calculate the sales of a products under a particular category

3. factBakerySales



The description of the attributes in factBakerySales can be summarized as:

Fact table Keys -
Time Key: A unique identifier for time. This is a surrogate key from the time dimension – dimTime.
Store Key: A unique identifier for each of the stores under DFF. This is a surrogate key from the store dimension – dimStore.

Fact table Measures -
Bakery Sales: This is a fact table measure that gives the amount of sales for products under Bakery category across stores and different time periods
.

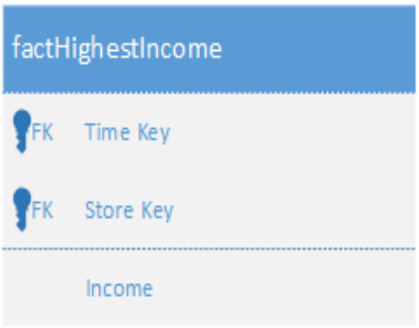
The physical model corresponding to factBakerySales fact table is:

factBakerySales:

Column Name	Data Type	Nulls Allowed	Comments
Product Key	Numeric	No	Foreign key, used to reference a row in the product dimension table
Store Key	Numeric	No	Foreign key, used to reference a row in the store dimension table

Time Key	Numeric	No	Foreign key, used to reference a date (week, month or year) from the time dimension table
Bakery Sales	Numeric	No	The measure which is used to calculate the sales of a products under a particular category

4. factHighestIncome



The description of the attributes in factHighestIncome can be summarized as:

Fact table Keys -
Time Key: A unique identifier for time. This is a surrogate key from the time dimension – dimTime.
Store Key: A unique identifier for each of the stores under DFF. This is a surrogate key from the store dimension – dimStore.

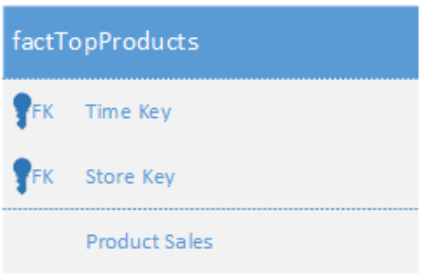
Fact table Measures -
Income: This is a fact table measure that gives the income of customers across stores for different time periods.

The table corresponding to factHighestIncome fact table is:

factHighestIncome:

Column Name	**Data Type**	**Nulls Allowed**	**Comments**
Store Key	Numeric	No	Foreign key, used to reference a row in the store dimension table
Time Key	Numeric	No	Foreign key, used to reference a date (week, month or year) from the time dimension table
Income	Numeric	No	The measure which is used to calculate the which store received the highest income

5. factTopProducts



The description of the attributes in factTopProducts can be summarized as:

Fact table Keys -
Time Key: A unique identifier for time. This is a surrogate key from the time dimension – dimTime.
Store Key: A unique identifier for each of the stores under DFF. This is a surrogate key from the store dimension – dimStore.

Fact table Measures -
Product Sales: This is a fact table measure that gives the amount of sales for different product categories across stores and different time periods.

The physical model corresponding to factTopProducts fact table is:

factTopProducts:

Column Name	**Data Type**	**Nulls Allowed**	**Comments**
Product Key	Numeric	No	Foreign key, used to reference a row in the product dimension table
Time Key	Numeric	No	Foreign key, used to reference a date (week, month or year) from the time dimension table
Product Sales	Numeric	No	The measure which is used which product in a given category had maximum sales

Star schemas corresponding to Business Questions:

Q1. Determine average gross profit margin of all the frozen products across stores and verify which stores are below average and plot the same.

The gross profit margin for any business is a fundamental measure in determining their economical or financial standing. Analysis of this metric will be critical in determining the future steps for either maintaining the financial gains or improving their profit margins. This question is designed to aid DFF business in understanding their current economic standing in the market for the sale of frozen products across their stores. The ability to look at gross profit over various time periods is crucial to understand the variations in market and customer needs. The fact table Figure-1 - factFrozenGrossProfit contains gross profit as a measurement metric, with dimProduct and dimStore as dimensions. We have the product dimension because we want the business to be able to look at profits for not just frozen products but other product categories as well. The time dimension allows us to look at the profit distribution across various times – week, month, and year.

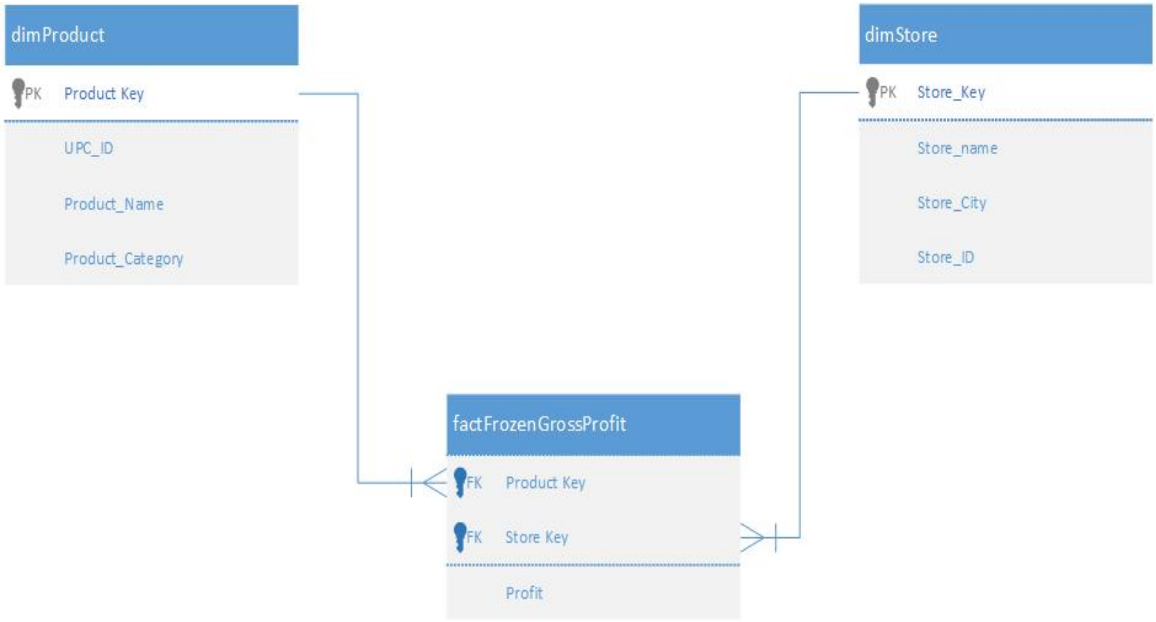


Figure 1 Star Schema 1

The mapping table corresponding to this question makes use of the following data:
These files from Movement and UPC excel sheets are used –
UPC - UPCFRD, UPCFRE, UPCFRJ
Movement - WFRD, WFRE, WFRJ

We make use of only these files because we are interested only in frozen products. However our logical model facilitates the measure for any category of products. We do this by having category type in the product dimension.

Q2. What is the average sales for beer in the last 3 years?

For any business, monthly sales is an important aspect to look at. This figure can be used to extrapolate the sales for the coming months and make necessary changes to the business. The sales metric also helps in understanding if a given category of products is being sold in the market or not and if sold how much of it is being sold. This business question aims at giving a report of sales of beer over the last three years at DFF. This will help DFF in analyzing which month had the highest sales for beer, giving them an understanding as to which months to target to sell more beer and in which month's customers don't buy beer. This helps in managing the product itinerary based on the customer demand.

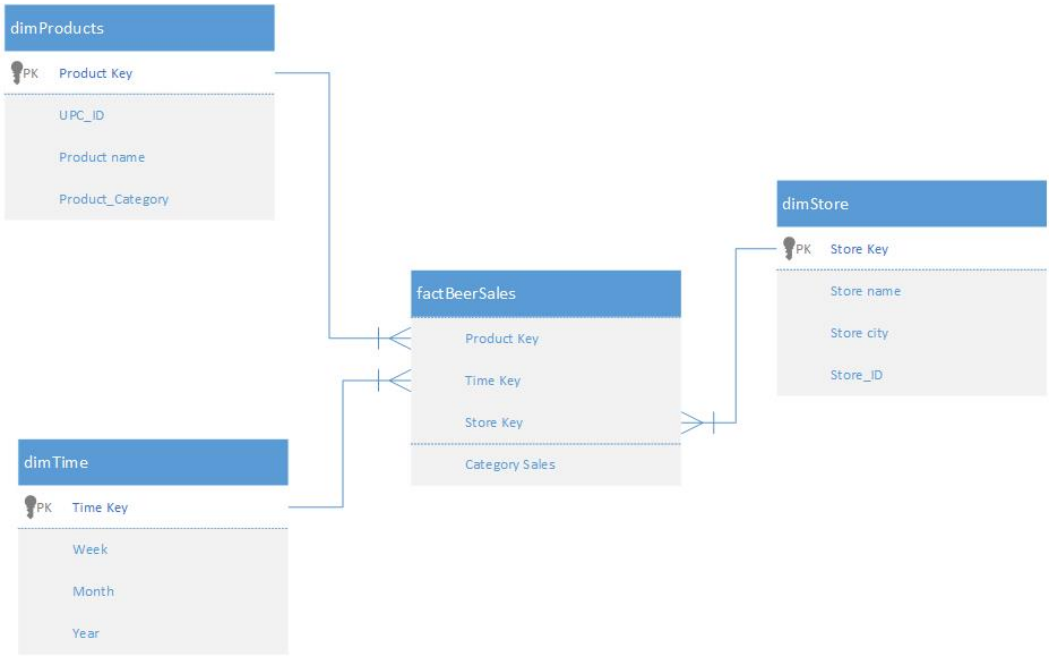


Figure 2 Star Schema 2

The fact table in Figure 2 – factBeerSales has the metric category sales which gives the amount of sales for products. We can look at the results for a particular category of products using the product category attribute in the product dimension - dimProduct. This metric can be reported across different time periods with the help of the time dimension – dimTime, across various stores with the store dimension - dimStore.

The mapping table corresponding to this question makes use of the these files from Movement and UPC excel sheets –
UPC – UPCBER
Movement – WBER

We make use of only these files because we are interested only in Beer category. However our logical model facilitates the measure for any category of products. We do this by having category type in the product dimension.

Q3. What is the growth of Bakery from the year 1990 to 1996?

Growth here is for the increase in sales of a particular product in DFF. This again is an important metric for businesses to consider. This would help the businesses keep track of increase or decrease in sales for a particular product category and make decisions regarding methods that would help increase the sale or remove the product from the market. This business question looks at the growth of sales in bakery products in particular. In this fast-food oriented generation, home cooking is declining and ready to eat or takeout meals are on the increase. Bakery products are part of the ready to eat food category and monitoring the sales in that direction would definitely be fruitful for DFF business.

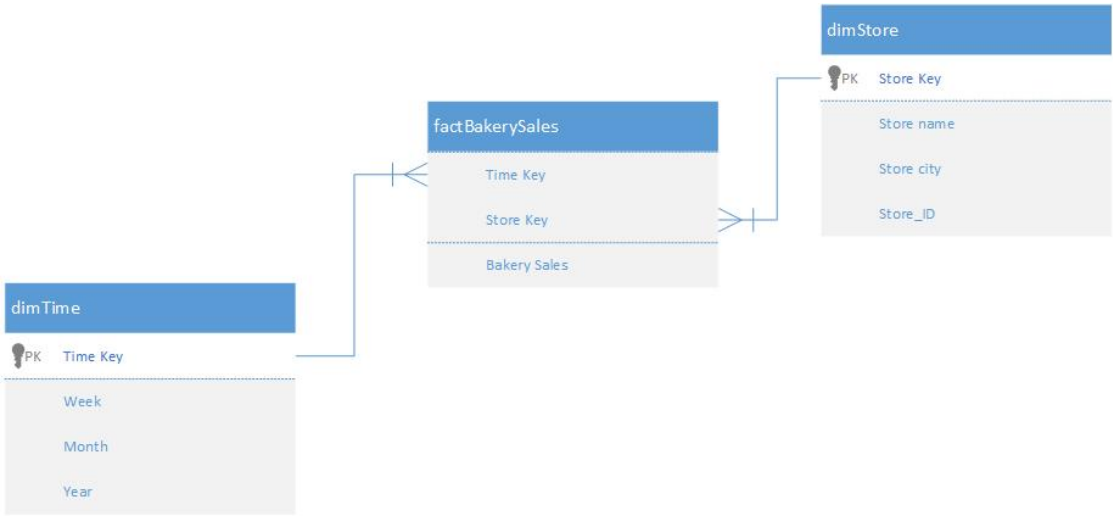


Figure 3 Star Schema 3

We make use of a fact table similar to that in Figure 3, except that the category of food we are interested in here is bakery products. The fact table in Figure 3 – factBakerySales has the metric Category sales which is the amount of sales of bakery products measured over a period of time. The time period we are interested in here is 1990-1996. The dimProduct dimension tables helps us isolate the bakery products, dimTime dimension table helps us isolate the data for time period 1990-1996, while dimStore dimension table helps us isolate the data for a particular DFF store.

The corresponding mapping table is given in Table 3. The category of products we will be interested in is Beer, taken from the CCount files. We make use of only this files because we are interested only in bakery product. However our logical model facilitates the measure for any category of products. We do this by having category type in the product dimension.

Q4. Which of the top 5 stores have customers with highest % income?

Knowledge about the financial standing of targeted customers is always good for any business. This information helps in identifying the probable products that the customers can afford and would want to buy. Also eliminating products that these customers would never or hardly buy. This would save the business thousands of dollars – investment in the right set of products, increase

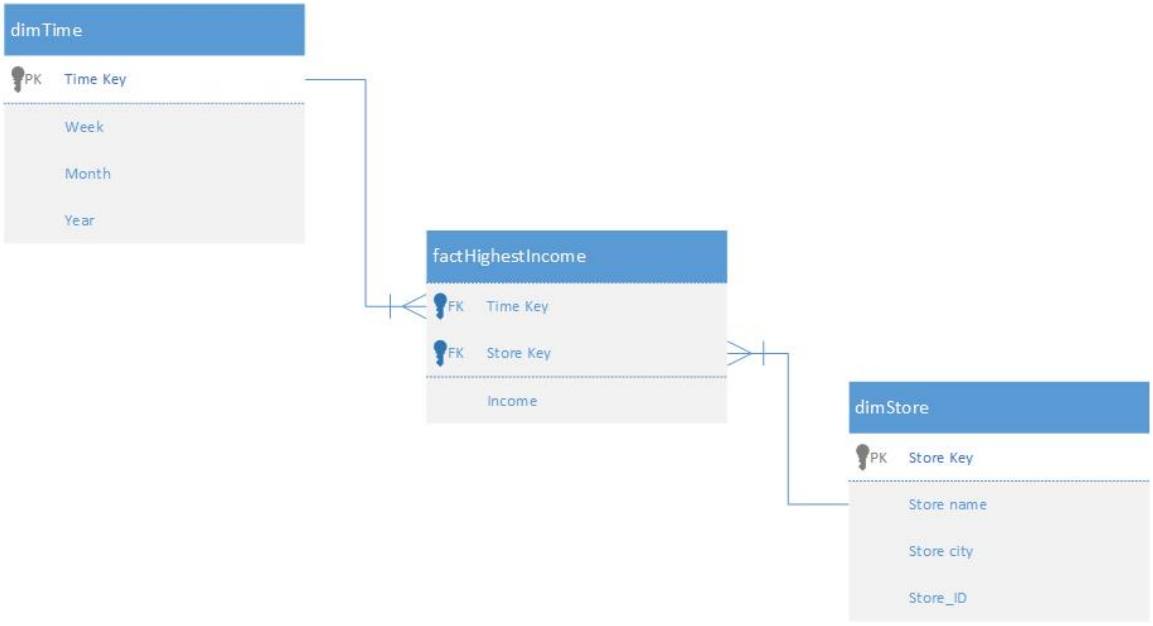


Figure 4 Star Schema 4

in sales because we have correctly identified the price range of products that customers will be inclined to purchase. With this business question we aim at producing a report that identify the top five stores which are frequented by customers with high income percentage. This would help DFF in identifying the price range of these shoppers and cater the products that these customers would like to indulge in. Thus, increasing the likelihood of sales and helping the business.

The fact table in Figure 4 – factHighestIncome has the metric highest income that gives us the highest income in every store. We isolate the top five stores with highest income. We have a dimension table for store – dimStore to facilitate filtering the results for each store and a dimension table for time – dimTime to facilitate filtering the results for a period of time. With this we can obtain the desired result, draw inferences correspondingly and apply necessary measures to improve business.

The mapping table corresponding to Figure 4 is given in Table 4. Here we use the demographics file as the source file for store dimension table and correspondingly extract the income of customers. The dimension table dimTime has week, month, year hierarchies. The dimension table dimStore has store name, city and number as hierarchies. With these dimensions we can get reports over any time period and for any city.

Q5. Top 5 products sold in the last year?

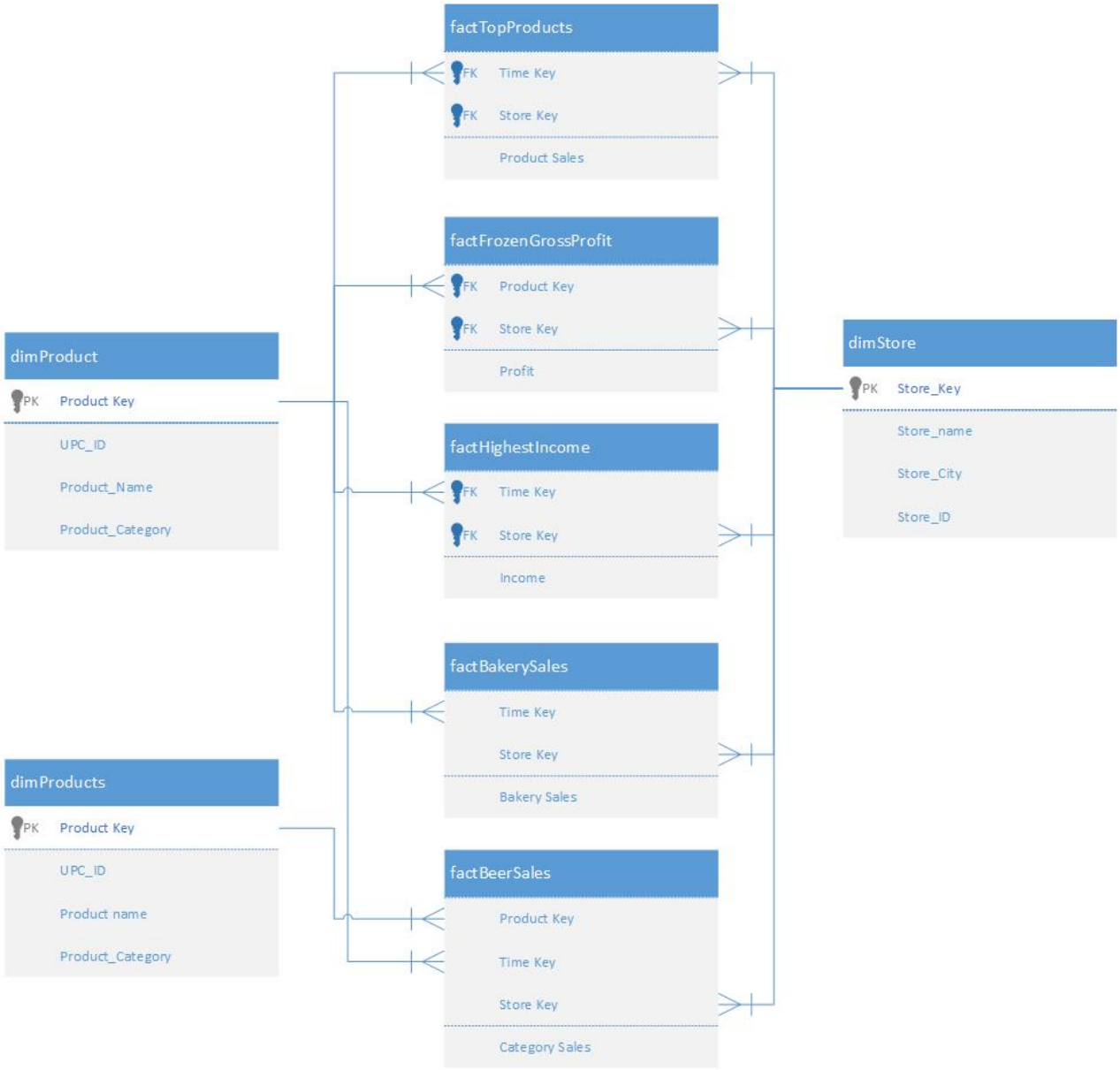
It is always good to know which products are being sold the most. This helps in promoting the lesser sold products more and stocking up the highly sold products. This business question aims at providing DFF with the information regarding the top five best sold products. This information can aid DFF in making decisions regarding product promotion, stock increase based on demand, price modulation to increase profits, etc. Observe the effects of these measures and decide on continuing or changing the marketing approach for products to improve sales.

Figure 5 Star Schema 5

The dimension model for this BQ is given in Figure 5. We have a fact table – factTopProducts that has the metric Product Sales. This gives us the list of products that are sold the most across all stores in DFF. The dimension table dimProducts gives us the product names and categories, whereas the dimension table dimTime is the time dimension which allows us to filter the results over different time periods – a week, month or year.

The mapping table corresponding to Figure 5 is given in Table 5. Here we use the Movement and CCount files as the source file for product dimension table and correspondingly extract the top five products sold. The dimension table dimTime has week, month, and year hierarchies. The dimension table dimStore has store name, city and number as hierarchies. With these dimensions we can get reports over any time period and across different stores and cities.

The star family for these five business questions is as follows –



The mapping table for the star schema -

	Staging Table	Source Column Mapping	Dimension or Fact Tables	Destination Column Mapping	Transformation
1	[UPC_Staging]	Auto	[dimProduct]	[Product_Key]	The UPC number and product key should be of type bigint and is autogenerated
		[UPC_Number]		[UPC_Number]	
		[Product_Name]		[Product_Name]	
		[Product_Category]		[Category]	
2	[Demographics_Staging]	Auto	dimStore	[Store_Key]	The store key and store number have to be of type int and is autogenerated.
		[Store_Number]		[Store_Number]	
		[Store_Name]		[Store_Name]	
		[Store_City]		[Store_City]	
3	[Date_Staging]	Auto	[dimTime]	[Time_Key]	The date key and date number have to be of type int
		year([StartDate])		[Year]	
		month([StartDate])		[Month]	
		[week]		[week]	
5	[dimTime] & [dimProduct] & [dimStore] & [Movement_Staging]	t.[Time_Key]	[factFrozenGrossProfit]	[Time_Key]	
		p.[Product_Key]		[Product_Key]	
		s.[Store_Key]		[Store_Key]	
		[Gross_Profit]		[Profit]	Profit is of type float
	[Movement_Staging] & [dimTime] & [dimStore] & [dimProduct]	t.[Time_Key]		[Time_Key]	
		p.[Product_Key]		[Product_Key]	
		s.[Store_Key]		[Store_Key]	
		stc.[SalesAmount]		Bakery_Sales]	Sales is of type float
	[CCOUNT_Staging] & [dimTime] & [dimStore] & [dimProduct]	t.[Time_Key]		[Time_Key]	
		p.[Product_Key]		[Product_Key]	
		s.[Store_Key]		[Store_Key]	
		stc.[SalesAmount]		[Sales]	Sales is of type float
	[Demographics_Staging] & [dimTime] & [dimStore]	t.[Time_Key]		[Time_Key]	
		s.[Store_key]		[Store_Key]	
		i.[HighestIncome]		[Income]	The highest income must be of type float
	[Ccount_Staging] & [dimTime] & [dimStore]	t. [Time_key]		[Time_key]	
		s. [Store_key]		[Store_key]	
		st. [Product Sales]		[Product_Sales]	Sales is of type float

Data Quality

Data quality is fundamental and critical for a data warehouse. This is because strategic decisions are made based on the analysis carried on over this data. If the data is dirty, it would result in data contamination, incorrect analysis, disastrous decisions, etc. Poor data quality is one of the biggest challenges faced by organizations across the world. It is not any different with DFF data. We encountered a number of data quality problems with the DFF data. These are summarized below:

Referential Integrity Problem:

Referential integrity was difficult to figure out as there were values that were null even in keys which could be assumed to be primary key. Many of the dependant tables had values that did not exist in the parent table. e.g.:- Store number was repeated and had inconsistent values.
Format. Store IDs in movement data did not exist in Demographics data.

Values did not follow consistent formatting standards. E.g.: Date attribute had a format as XXXXXX instead of XX/XX/XX

Data Standard

Though data elements were explained well in DFF catalog, looking at the data it was difficult to understand what it represented. E.g.: Coupons had decimal values, so shall a user assume it is coupon's dollar value or it is dirty data.

Consistency

Data was explained well in the catalog and hence the data had same meanings across the systems. However, amongst files there was an overlap with the data values which created misrepresentation of data.

Completeness

After formulating the business questions, it was found that a lot of data was not present. For example, the movement files were not available for all the UPCs.

Accuracy

Accuracy was questionable as data had null values where value was expected, negative values for Sales data and decimal value for coupon.

Validity

Data values did not fall within acceptable ranges defined by the business. Store number have out of range values and hence have invalid values.

Fit for purpose

The information contains a lot of sales data and customer data and hence is valuable to the business.

Data Extraction Rules

ETL stands for Extract-Transform-Load. ETL rules specify how the data is extracted from the source, transformed and loaded into the data warehouse.

The *Extract* step covers the data extraction from the source system. The main objective of extract is to retrieve all the required data from the source system with as little resources as possible. The extract step should be designed in a way that it does not negatively affect the source system in terms of performance, response time, etc. We only extract the data that is relevant to our business questions. This reduces the overhead of extracting all of the data and hence increases performance efficiency.

The cleaning step is one of the most important as it ensures the quality of the data in the data warehouse. Cleaning should perform basic data unification rules, such as making identifiers unique, converting NULL values into standardized not available/not provided value, convert phone numbers to a standardized form and validate address fields against each other (State/Country, City/State, City/ZIP code, City/Street). We excluded the NULL and junk value tuples from the source tables. Including such values would give rise to problems later on while transforming and performing aggregate functions.

Transform involves a set of rules to transform data from source to the target. This involves converting the data into conformed dimensions using the same units so that they can later be joined. This also involves joining the data from numerous sources, generating aggregates, surrogate keys, sorting and deriving calculated values. We had to transform the date attribute in CCount from string type to DBMS date type. We did this using derived attributes in SSIS.

It is imperative that the *Load* is performed correctly and with as little resources as possible. In order to make the load process efficient, it is helpful to disable any constraints and indexes before the load and enable them back only after the load completes. The referential integrity needs to be maintained by ETL tool to ensure consistency. However since we had data relevant to our BQs we did not disable any constraints prior to load.

Once the data is loaded into the staging area called as “602_Group2_Staging”, the data is then transformed as per the rules stated in the data mapping and transformation area. The data is then extracted into the data warehousing area which consists of the data marts made up of the dimension and fact tables.

Mapping and Transformation:

One of the most important aspects of ETL is data mapping and transforming the data. In data mapping the source tables, in this case excel source files, are mapped with the table created in the staging area and then accordingly mapped with the tables in the data warehousing area or with the data marts. Data transformation consists of methods of converting the data from one format to another based on certain rules in order to keep the data consistent throughout the process.

The mapping from source files to staging tables is as below:

	Source	Source Column Mapping	Staging Table	Destination Column Mapping	Transformation
1	CCount Excel	Store_ID	[Ccount_Staging]	[Store_ID]	Change from String to Int FK references [Demo_Staging]
		Remaining Coulmns		Auto_Mapped	
2	Demographics Excel	[Store_ID]	[Demo_Staging]	[Store_ID]	Change from String to Int
		Remaining Coulmns		Auto_Mapped	
3	WFRD	UPC_ID	[Movement_Staging]	[UPC_ID]	Change from String to Int
	WFRE	Store_ID		[Store_ID]	Change from String to Int FK references [Demo_Staging]
	WFRJ	Remaining Coulmns		Auto_Mapped	
	WBER				
4	UPCFRD	[UPC_ID]	[UPC_Staging]	[UPC_ID]	Change from String to Int FK references [UPC_Staging]
	UPCFRE	Remaining Coulmns		Auto_Mapped	
	UPCFRJ				

The mapping from staging tables to dimension and fact tables is as below:

	Staging Table	Source Column Mapping	Dimension or Fact Tables	Destination Column Mapping	Transformation
1	[UPC_Staging]	Auto	[dimProduct]	[Product_Key]	The UPC number and product key should be of type bigint and is autogenerated
		[UPC_Number]		[UPC_Number]	
		[Product_Name]		[Product_Name]	
		[Product_Category]		[Category]	
2	[Demographics_Staging]	Auto	dimStore	[Store_Key]	The store key and store number have to be of type int and is autogenerated.
		[Store_Number]		[Store_Number]	
		[Store_Name]		[Store_Name]	
		[Store_City]		[Store_City]	
3	[Date_Staging]	Auto	[dimTime]	[Time_Key]	The date key and date number have to be of type int
		year([StartDate])		[Year]	
		month([StartDate])		[Month]	
		[week]		[week]	
5	[dimTime] & [dimProduct] & [dimStore] & [Movement_Staging]	t.[Time_Key]	[factFrozenGrossProfit]	[Time_Key]	
		p.[Product_Key]		[Product_Key]	
		s.[Store_Key]		[Store_Key]	
		[Gross_Profit]		[Profit]	Profit is of type float
	[Movement_Staging] & [dimTime] & [dimStore] & [dimProduct]	t.[Time_Key]		[Time_Key]	
		p.[Product_Key]		[Product_Key]	
		s.[Store_Key]		[Store_Key]	
		stc.[SalesAmount]		Bakery_Sales]	Sales is of type float
	[CCOUNT_Staging] & [dimTime] & [dimStore] & [dimProduct]	t.[Time_Key]		[Time_Key]	
		p.[Product_Key]		[Product_Key]	
		s.[Store_Key]		[Store_Key]	
		stc.[SalesAmount]		[Sales]	Sales is of type float
	[Demographics_Staging] & [dimTime] & [dimStore]	t.[Time_Key]		[Time_Key]	
		s.[Store_key]		[Store_Key]	
		i.[HighestIncome]		[Income]	The highest income must be of type float
	[Ccount_Staging] & [dimTime] & [dimStore]	t. [Time_key]		[Time_key]	
		s. [Store_key]		[Store_key]	
		st. [Product Sales]		[Product_Sales]	Sales is of type float

Determining data transformation and cleansing rules

Data transformation and cleaning process follows the Data Extraction process. The input to this process is the extracted data that is present in the staging area. Transformation and cleaning rules have been applied to this data to create a data that is clean and consistent throughout. The clean data can then be loaded into the DW area so as to create the data marts and get the appropriate results.

The general transformation and cleaning rules that are applied to the data are as follows:

i. Removal of NULL values

Null values that were existing in the data and that were created while extracting data into tables will be deleted.

ii. Removal of dirty/junk data

Attributes that are not part of the answers to the business questions will be removed. For instance, attributes other than Store, Beer, Fish, Camera and Week in the Customer Count files have been eliminated. Also, the blank records, rows with just a '.' (Dot) and other non-meaningful values such as unnecessary negatives will be removed. The records where the value for the sales of various departments are '.' will be removed.

iii. Data conversion

The data was extracted from the source files are stored as attributes of type varchar (that is, as string) in the staging area. These have been converted into their respective data types such as int, float and date. Date type field was stored as varchar. The date has been split using functionalities in SSIS and then stored as Year and Month attributes.

iv. Creation of surrogate keys

Surrogate keys have been created for all the dimension tables and fact tables before loading the data in the data warehouse.

v. Derived attributes

Derived attributes exist in two dimensions, described as below:

- Year and month in the Time dimension, obtained from the functions YEAR (date) and MONTH (date) respectively.

SSIS Functions

The SSIS functions that aided during the transformation and cleaning process include:

1. **LOOKUP**: It joins additional columns to the data flow by looking up values in a table.

2. **DATA CONVERSION**: It converts data from one data type to another.

3. **DERIVED COLUMNS**: It creates new column values by applying expressions to input columns.

4. **AGGREGATE**: It aggregates data with functions such as count and sum.

Q1. Determine average gross profit margin of all the frozen products across stores and verify which stores are below average and plot the same.

For this question, data is extracted from the date, UPC and demo tables of the staging area. From the date table, date number, date, week number, year and day are mapped into the date dimension. From the UPC staging table, UPC ID, product name and category of only frozen foods are mapped into the product dimension. Finally, the demographic information such as store name, city and number are mapped to the store dimension. The fact table is mapped by creating a foreign key relation with the dimension tables of product, store and date. The gross profit is extracted and mapped from the UPC table of the staging area.

Q2. Which month had the highest sales for beer during the last 3 years?

For this question, data is extracted from the date, UPC and demo tables of the staging area. From the date table, date number, date, week number, year and day are mapped into the date dimension. From the UPC table, UPC number, product name and of only beer category are mapped into the product dimension. Finally, the demographic information such as store name, city and number are mapped to the store dimension. The fact table is mapped by creating a foreign key relation with the dimension tables of product, store and date. The sales for Beer is extracted and mapped from the Movement table of the staging area.

Q3. What is the growth of Bakery from the year 1990 to 1996?

For this question, data is extracted from the date, CCount and demo tables of the staging area. From the date table, date number, date, week number, year and day are mapped into the date dimension. From the CCount table, UPC number, product name and of only bakery category are mapped into the product dimension. Finally, the demographic information such as store name, city and number are mapped to the store dimension. The fact table is mapped by creating a foreign key relation with the dimension tables of product, store and date. The growth calculated from the sales of bakery items and is extracted and mapped from the CCount table of the staging area.

Q4. Which of the top 5 stores have customers with highest % income?

For this question, data is extracted from the date, CCount and Demo tables of the staging area. From the date table, date number, date, week number, year and day are mapped into the date dimension. Finally, the demographic information such as store name, city and number are mapped to the store dimension. The fact table is mapped by creating a foreign key relation with the dimension tables of product, store and date. The top 5 stores with highest % income are taken from the demo table and are extracted and mapped from the demo table of the staging area.

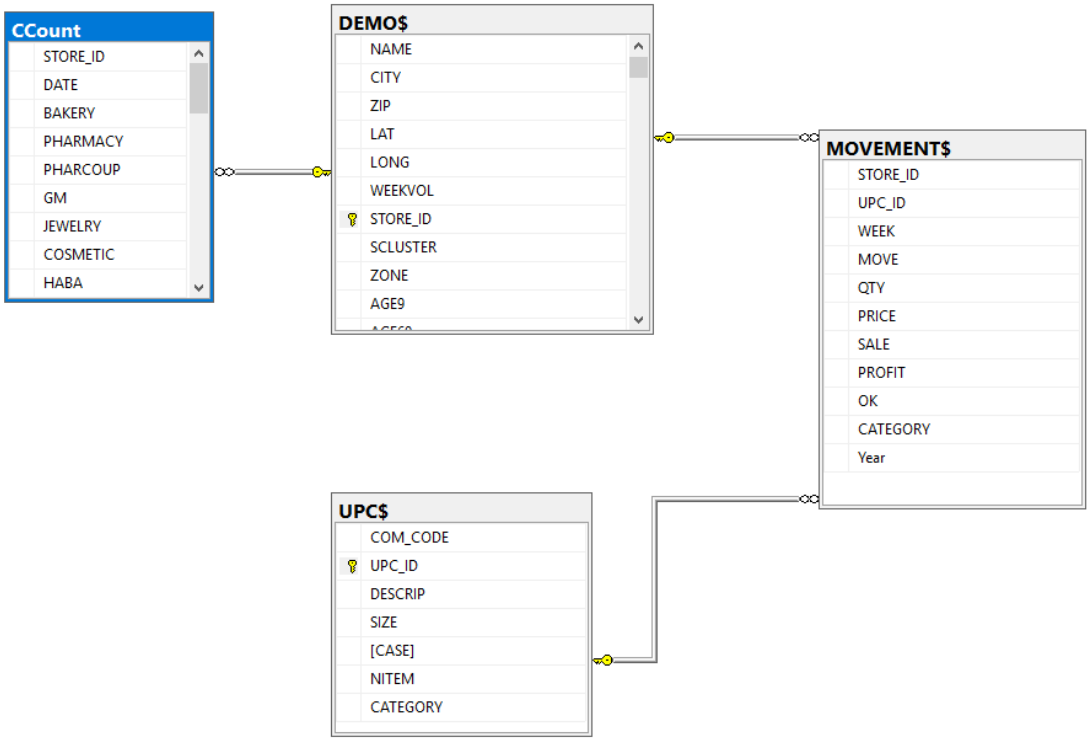
Q5. Top 5 products sold in the last year?

For this question, data is extracted from the date, CCount and demo tables of the staging area. From the date table, date number, date, week number, year and day are mapped into the date dimension. Finally, the demographic information such as store name, city and number are mapped to the store dimension. The fact table is mapped by creating a foreign key relation with the dimension tables of product, store and date. The top 5 product categories sold are taken from the CCount table and is extracted and mapped from the demo table of the staging area.

Organization of Staging Area

Data Staging area consists of tables loaded directly from source files via SSIS. All the files are loaded into staging DB with a bit of data manipulation and attribute type alteration based on the required relationships between tables. Data transformation is carried out in the staging area during extraction from the source excel files. Tables are created on the fly using SSIS import and export wizard where data source is chosen, and the destination is staging DB. During the data transfer from source to destination, table is auto-created with a flexibility to rename or change types of the destination attributes. This is a sophisticated way to map source and destination files and reduce additional hassle of creating the tables. We have followed this method while creating and designing all the staging DB tables. Staging Database consists of DEMO$, MOVEMENT$, UPC$ and CCount$ tables. Foreign keys are constructed during the SSIS import and export process by appending additional SQL query lines to CREATE table query (Auto generated by the SSIS package). We also have used complex data transformation for date column to (mmyydd) date format, which was originally in text form (yymmdd) in CCount file.

Structure of tables in Staging Area after extraction and loading from the source files –



Data Extraction Procedure

Source Identification:

Source for the data required in the staging area are in Excel/ .CSV format. As the source contains historic data, we will proceed with one-time data load using these excel and CSV files into the staging DB. There are mainly 4 data files that will be loaded in the staging DB as explained earlier i.e. Demographics, Movement, UPCxxx (xxx represents category codes), CCount and Date.

Method of Extraction:

Extraction from these excel and CSV files will be carried out via SSIS import & Export packages in Microsoft Visual Studio. Since CCount and Movement files are considerably large and are time consuming, Visual Studio is the most convenient and efficient way to extract data from these files and store into the staging environment in a single flow of steps. This method via SSIS helps us avoid table creation at staging DB before loading the data, as there is a more convenient way of creating the table on the fly right before importing the data into the tables. Such useful features save us tremendous amount of efforts in extracting historic data from source files.

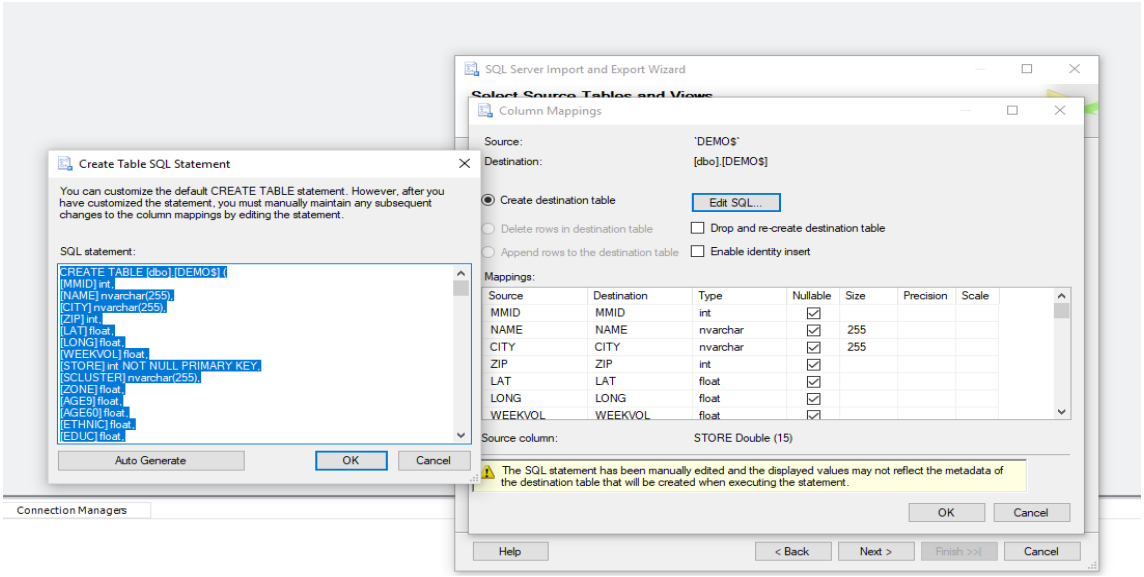
Extraction frequency: One-time extraction.

Extraction Procedure:

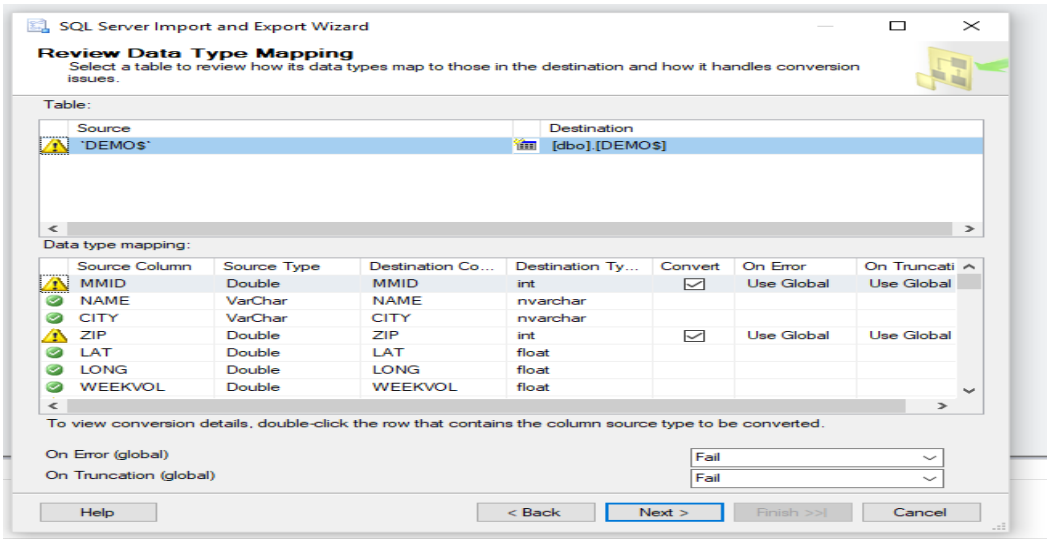
Detailed process for all the extractions and loadings using SSIS are as follows:

i. Select Source file as Demographics.xlsx, select 602_group2_staging_db as the destination database.

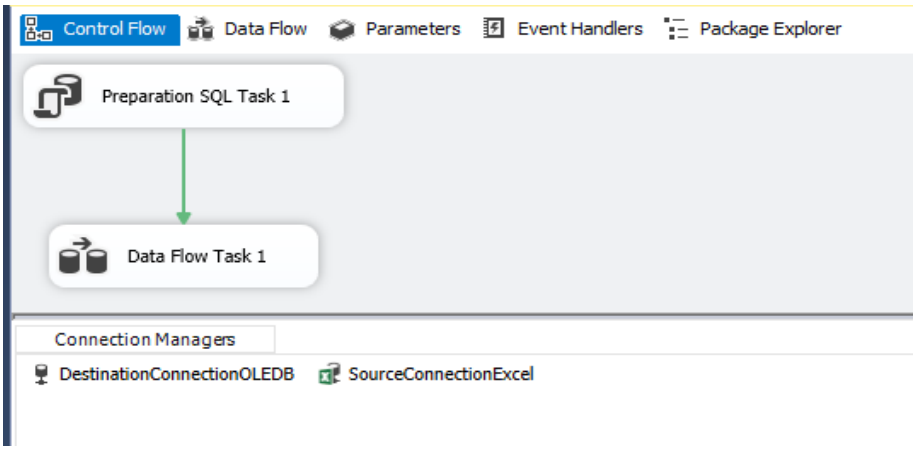
ii. Next, we proceed to map the columns and change the attribute type of Store_ID from varchar to int. Since the table does not already exist in the database, the SSIS package automatically creates the table. We append the SQL query to make Store_ID as primary key.



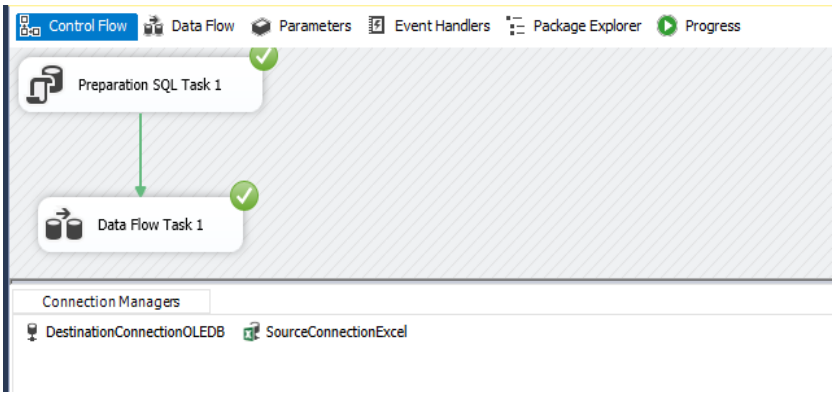
iii. Next, we make sure the columns from source are mapped correctly to the destination table.



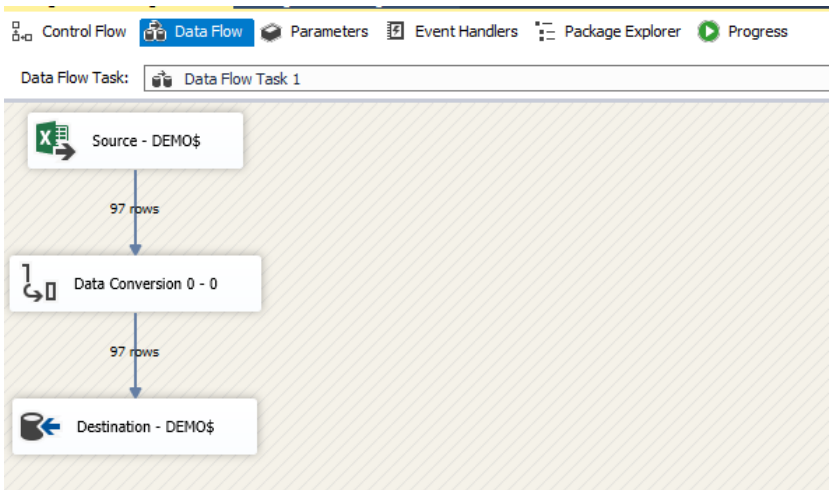
iv. The package will now have the following tasks in control flow:



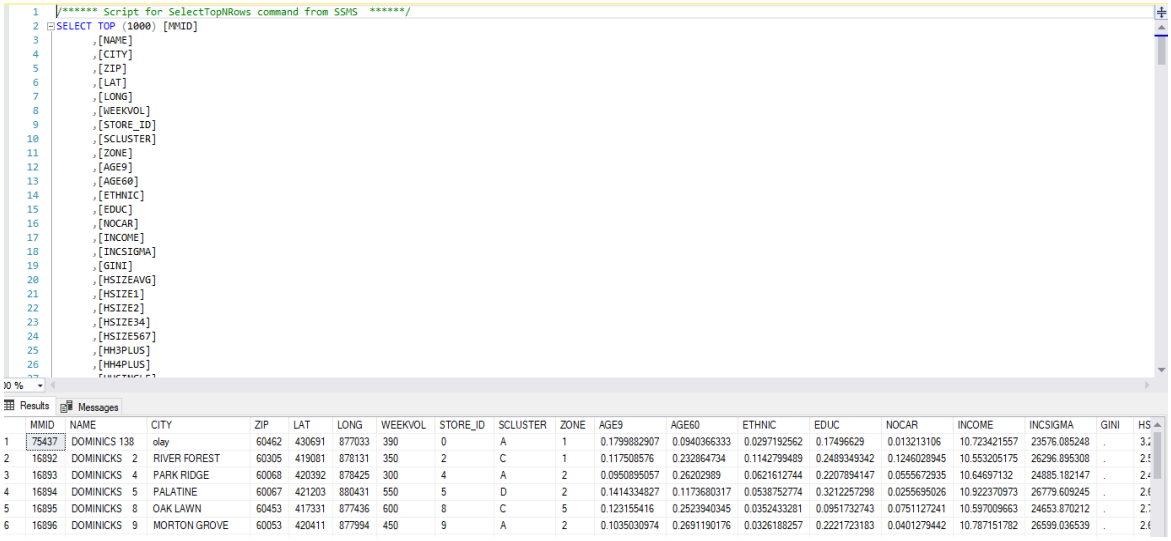
v. Next, we execute the package to reflect changes in the staging database. On successful execution we see these results



vi. The data flow tab looks like this



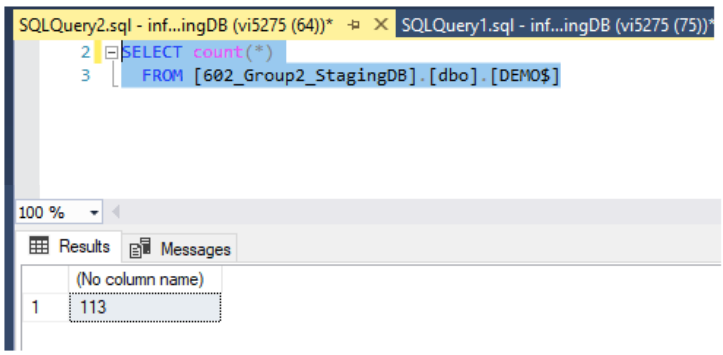
vii. We can now verify the creation of the table and check the contents of the demo table in the staging area using SQL Server Management Studio.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [MMID]
      ,[NAME]
      ,[CITY]
      ,[ZIP]
      ,[LAT]
      ,[LONG]
      ,[WEEKVOL]
      ,[STORE_ID]
      ,[SCLUSTER]
      ,[ZONE]
      ,[AGE9]
      ,[AGE60]
      ,[ETHNIC]
      ,[EDUC]
      ,[NOCAR]
      ,[INCOME]
      ,[INCSIGMA]
      ,[GINI]
      ,[HSIZEAVG]
      ,[HSIZE1]
      ,[HSIZE2]
      ,[HSIZE34]
      ,[HSIZE567]
      ,[HH3PLUS]
      ,[HH4PLUS]
```

Results | Messages

	MMID	NAME	CITY	ZIP	LAT	LONG	WEEKVOL	STORE_ID	SCLUSTER	ZONE	AGE9	AGE60	ETHNIC	EDUC	NOCAR	INCOME	INCSIGMA	GINI	HS
1	75437	DOMINICS 138	olay	60462	430691	877033	390	0	A	1	0.1799882907	0.0940366333	0.0297192562	0.17496629	0.013213106	10.723421557	23576.085248	.	3.2
2	16892	DOMINICKS 2	RIVER FOREST	60305	419081	878131	350	2	C	1	0.117508576	0.23284734	0.1142799489	0.2489349342	0.1246028945	10.553205175	26296.895308	.	2.5
3	16893	DOMINICKS 4	PARK RIDGE	60068	420392	878425	300	4	A	2	0.0950895057	0.26202989	0.0621612744	0.2207894147	0.0555672935	10.64697132	24885.182147	.	2.4
4	16894	DOMINICKS 5	PALATINE	60067	421203	880431	550	5	D	2	0.1414334827	0.1173680317	0.0538752774	0.3212257298	0.0255695026	10.922370973	26779.609245	.	2.6
5	16895	DOMINICKS 8	OAK LAWN	60453	417331	877436	600	8	C	5	0.123155416	0.2523940345	0.0352433281	0.0951732743	0.0751127241	10.597009663	24653.870212	.	2.7
6	16896	DOMINICKS 9	MORTON GROVE	60053	420411	877994	450	9	A	2	0.1035030974	0.2691190176	0.0326188257	0.2221723183	0.0401279442	10.787151782	26599.036539	.	2.6



SQLQuery2.sql - inf...ingDB (vi5275 (64))* | SQLQuery1.sql - inf...ingDB (vi5275 (75))*

```
SELECT count(*)
  FROM [602_Group2_StagingDB].[dbo].[DEMO$]
```

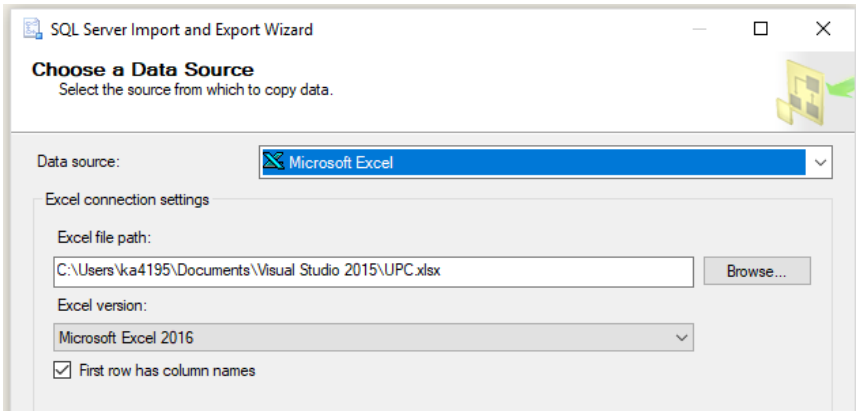
100 %

Results | Messages

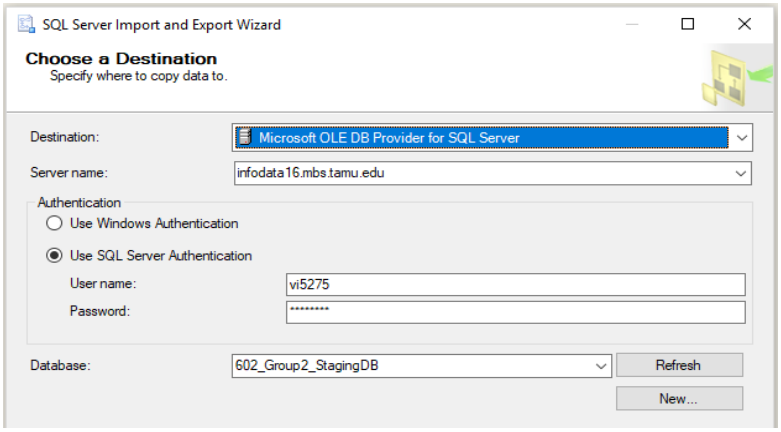
	(No column name)
1	113

Procedure for extraction and loading of UPC file from excel source files to staging DB:

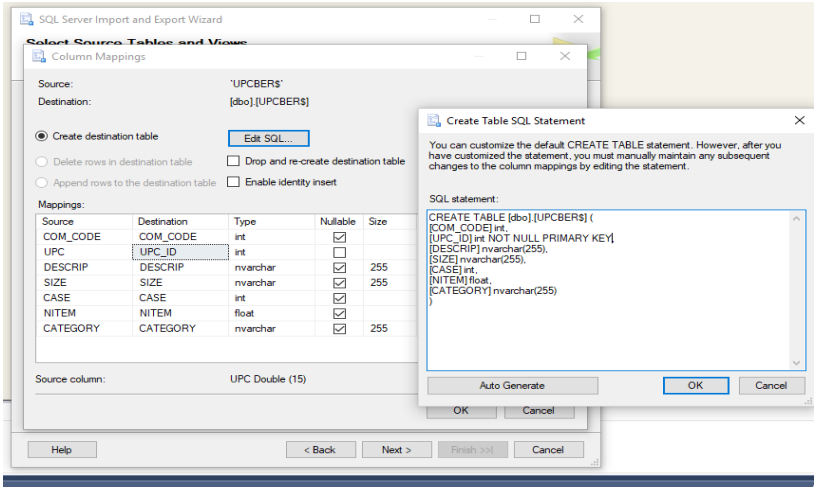
i. Specify source



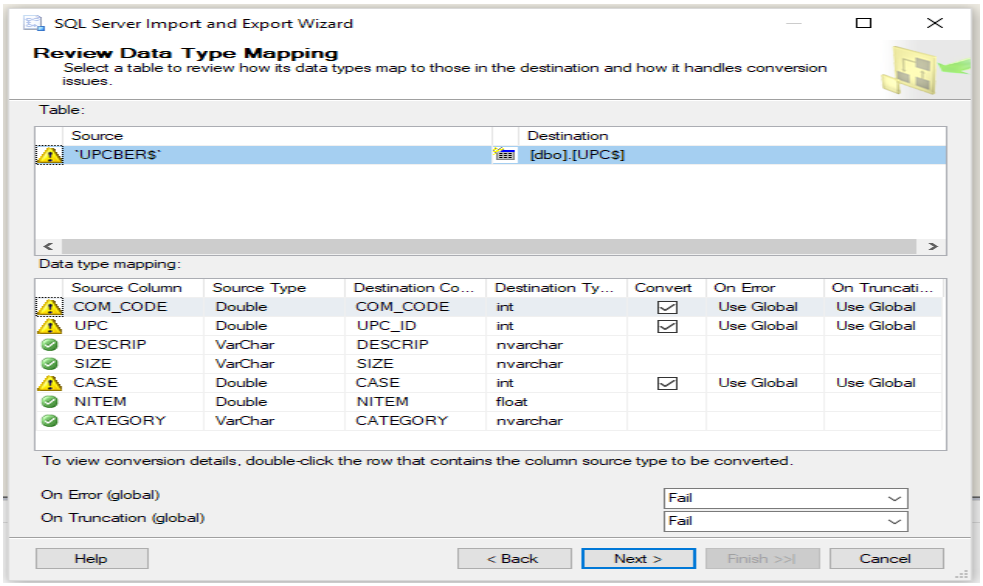
ii. Specify destination



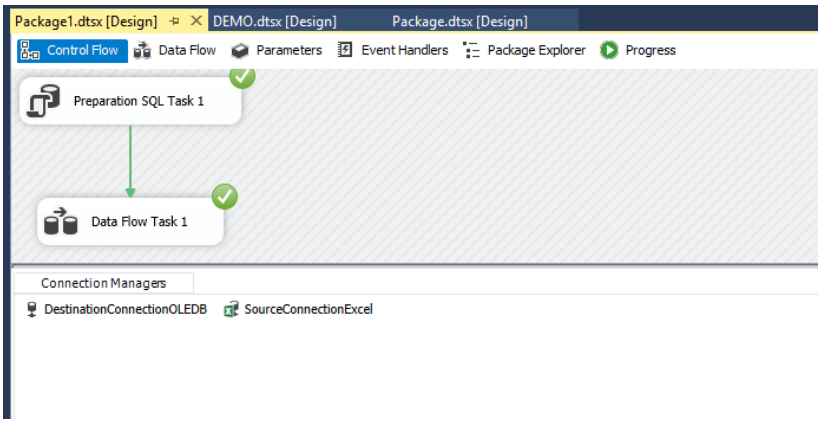
iii. Alter the create table query to include primary key UPC_ID as bigint

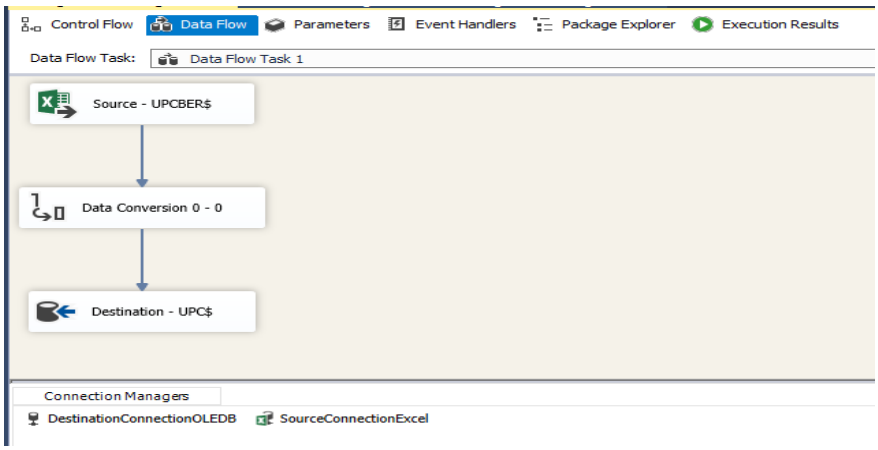


iv. Make sure the mappings from the source match the destination columns

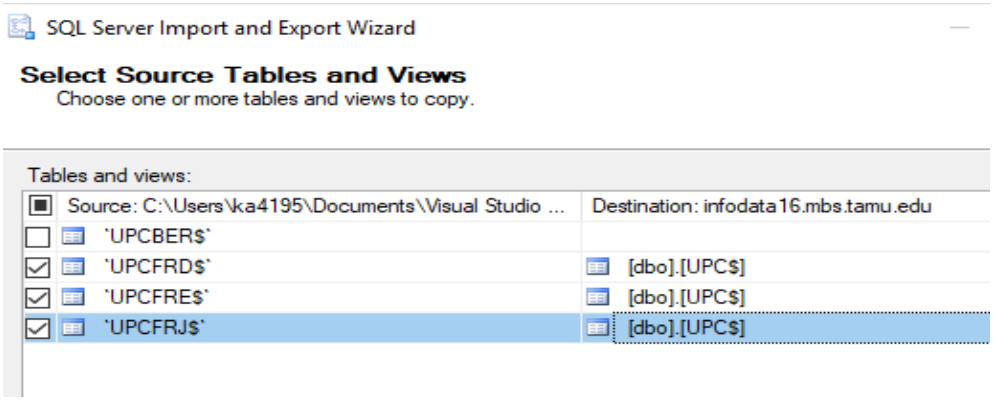


v. Execute the SSIS package.

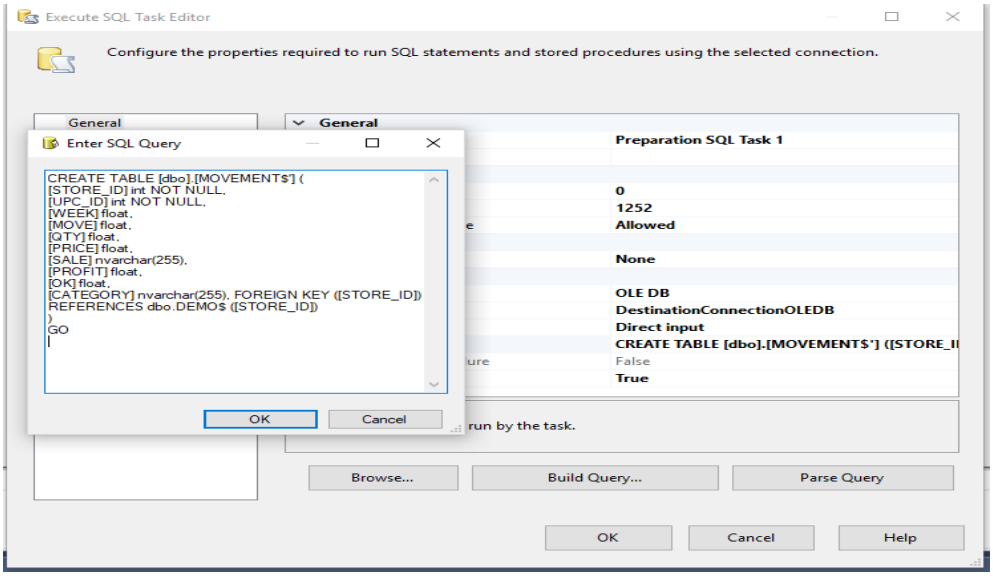




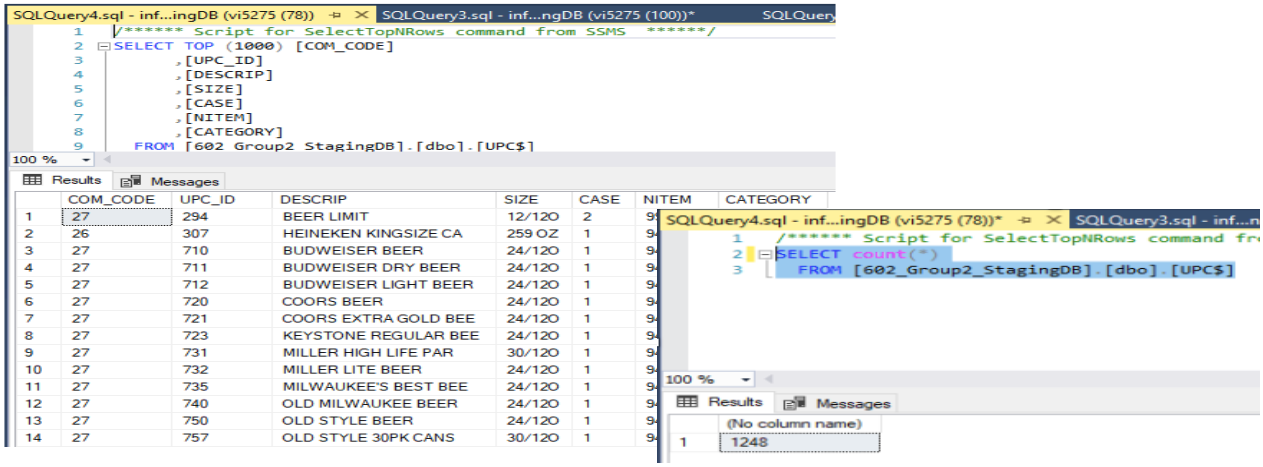
vi. Similarly, extract and load the other UPC files – UPCFRD, UPCFRE and UPCFRJ.



vii. Defining relationship constraints while creating the table by editing SQL query



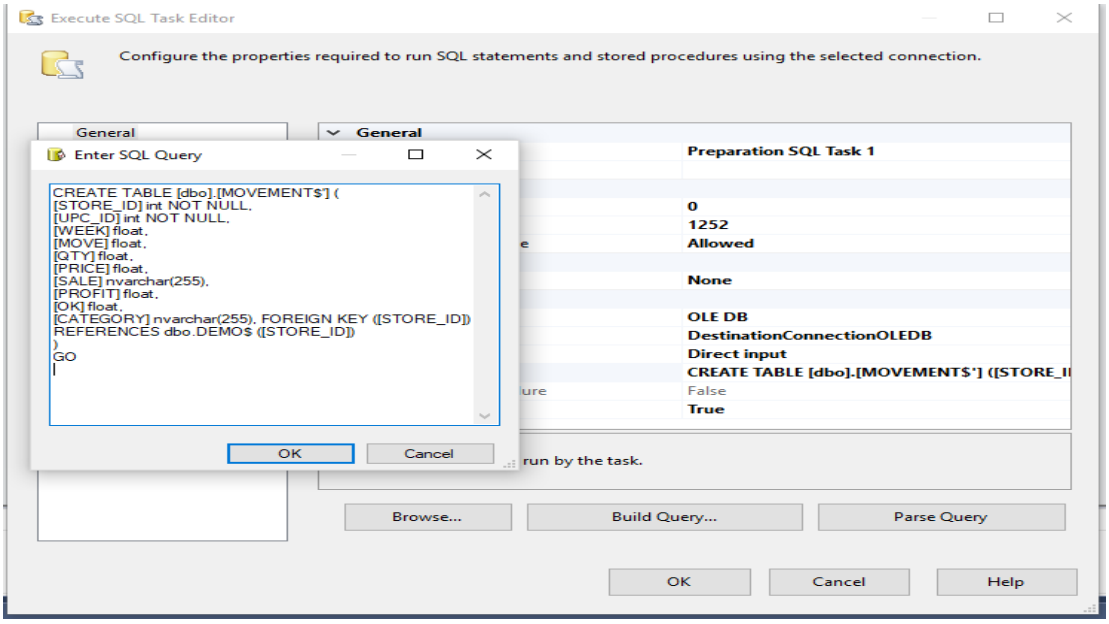
viii. After executing the package, we can verify the table creation and contents for UPC table in the staging area



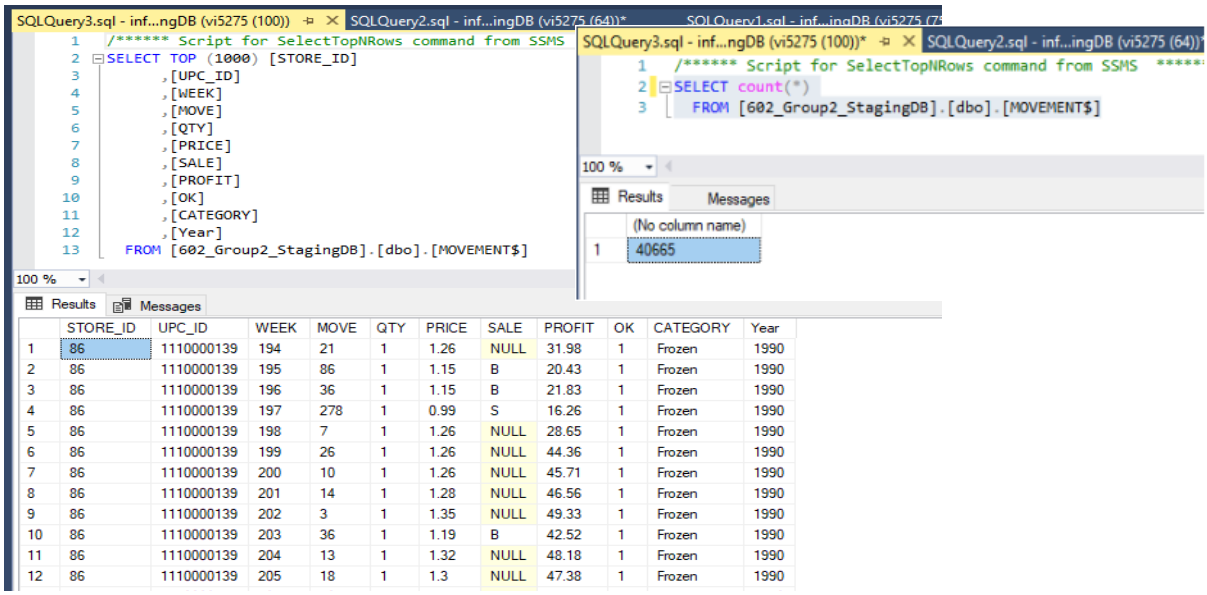
Source to Staging area process for Movement:

i. Select Movement excel file as the source and select 602_Group2_StagingDB as the destination database.

ii. Next, edit column mappings and change the attribute type for Store_ID to int and UPC_ID to bigint. Also, ensure to add referential integrity constraints referencing Demo and UPC table.



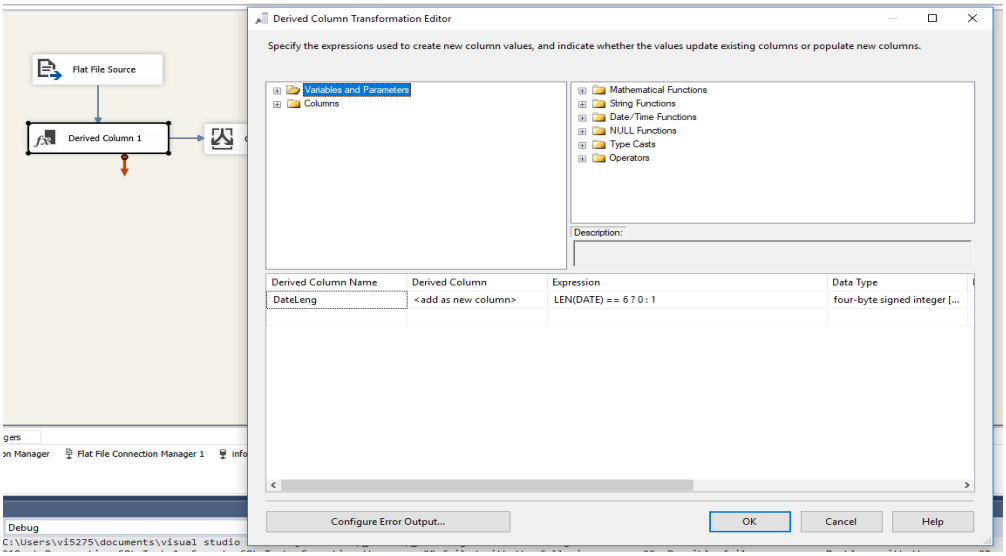
iii. Ensure correct mappings from source to destination table. And execute the package. We can verify the creation of movement table and its content in SQL Server Management Studio.



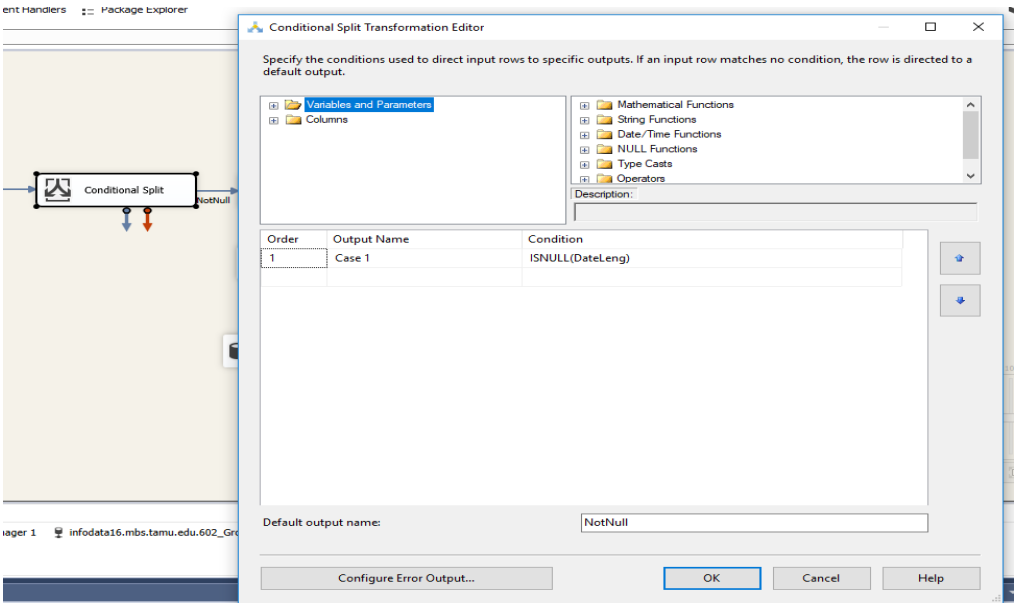
Source to Staging area process for CCount:

Note: CCount file can not be directly imported into staging area because Date attribute in the file needs further transformation. The necessary transformations are carried out using SSIS package – data flow. Process is as shown below.
SSIS Functions used for transformation: Derived column, Conditional Split

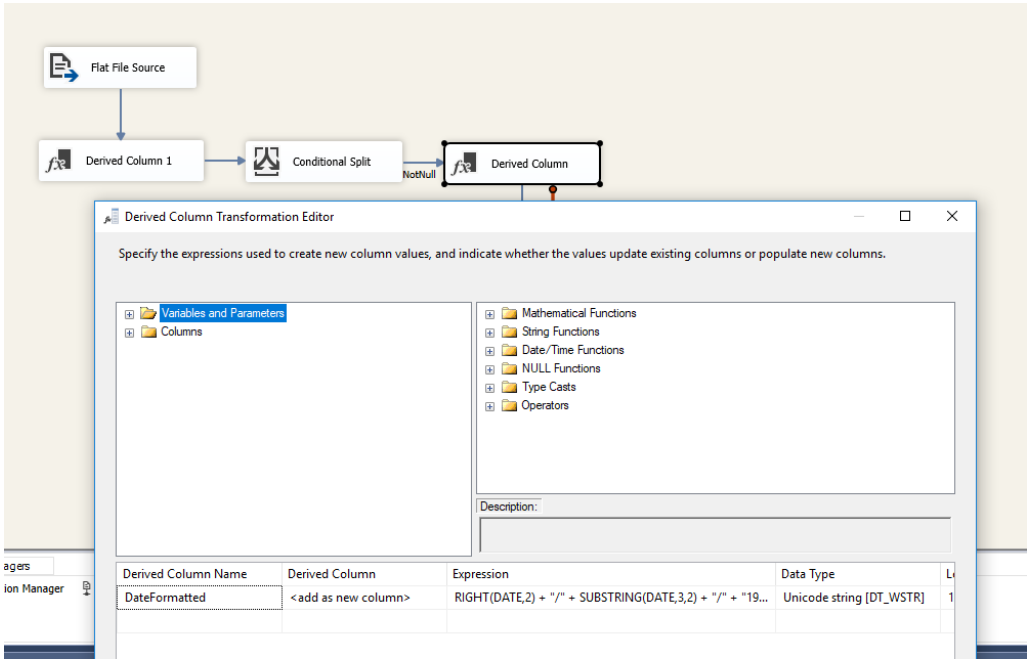
i. Convert CCount excel file to a CSV file. This conversion is needed because the date field in CCount file is a string and SSIS cannot translate this to DBDate format.
ii. Open Data flow tab - add CCount CSV as source. Add a derived column for date length.



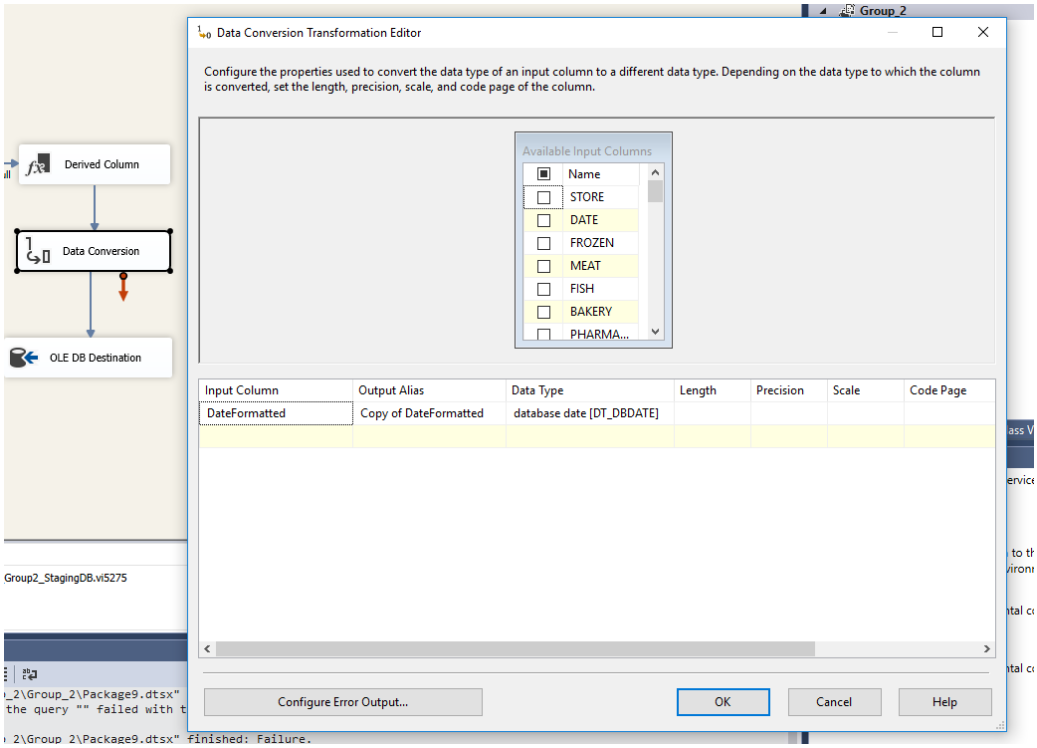
iii. Next, add conditional split based on null values in date length.



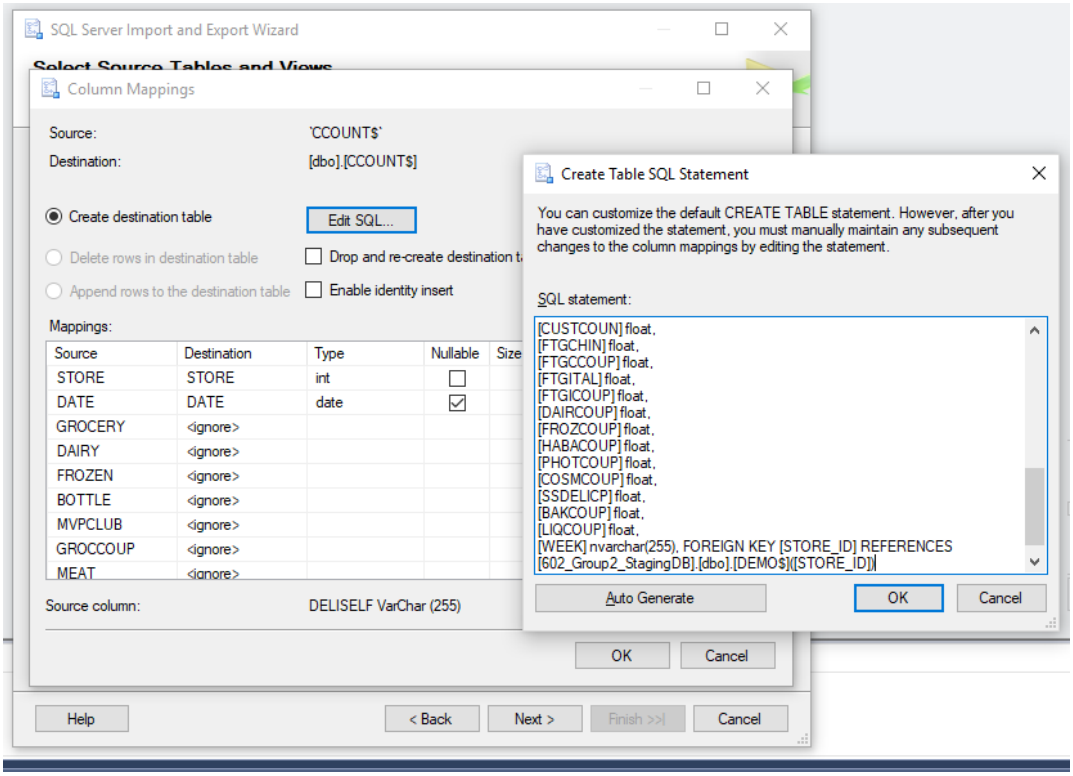
iv. Now, add a derived column for date formatted. Format the input by specifying the expression that will convert the string to mmddyy format.



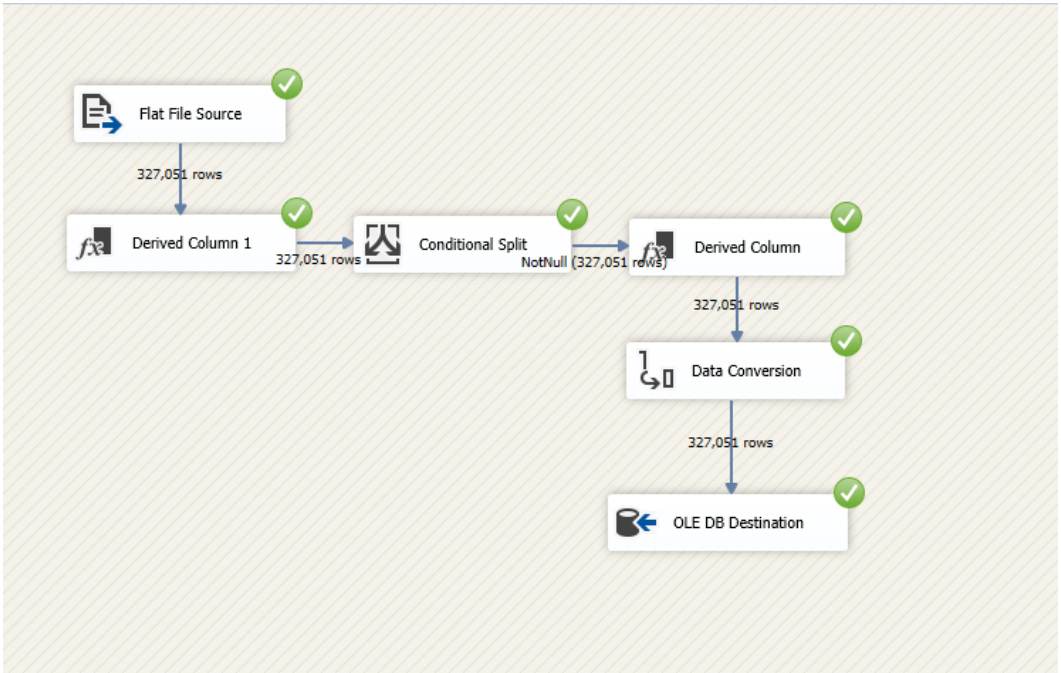
v. Add a data conversion task, that will make sure that the date attribute is in the format accepted by SQL Server



vi. Add OLE DB Destination to create the CCount table with appropriate foreign keys.



vii. Execute the package and verify creation of table in the staging database.



Successful execution can be verified in staging area

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [STORE_ID]
      ,[DATE]
      ,[BAKERY]
      ,[PHARMACY]
      ,[PHARCOUP]
      ,[GM]
      ,[JEWELRY]
      ,[COSMETIC]
      ,[HABA]
      ,[GMCOUP]
      ,[CAMERA]
      ,[PHOTOFIN]
      ,[VIDEO]
      ,[VIDEOREN]
      ,[VIDCOUP]
      ,[BEER]
      ,[WINE]
      ,[SPIRITS]
      ,[MISCSCP]
      ,[MANCOUP]
      ,[CUSTCOUN]
```

100 %

Results | Messages

	STORE_ID	DATE	BAKERY	PHARMACY	PHARCOUP	GM	JEWELRY	COSMETIC	HABA	GMCOUP	CAMERA	PHOTOFIN	VIDEO	VIDEOREN	VIDCOUP	BEER	WINE	SPIRITS	MISCSCP	MANCOUP	CUSTCOUN	FTC
1	0	930216	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2	0	930423	1893	1423	0	15254	0	0	0	0	0	0	253	542	0	0	0	0	0	0	4192	0
3	0	930607	921	0	0	2179	0	0	0	0	0	0	0	0	0	1003	0	0	0	0	1652	0
4	2	880101	1144.7	0	0	1884.91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1505	0
5	2	880102	2152.96	0	0	4253.57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2325	0
6	2	880103	1479.11	0	0	3818.29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2154	0
7	2	880104	953	0	0	2657	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1548	0
8	2	880105	1067.82	0	0	2406.37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1590	0
9	2	880106	1407.65	0	0	2891.34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1721	0
10	2	880107	1648.65	0	0	3348.72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2090	0

SQLQuery1.sql - inf...ingDB (vi5275 (75))* | INFODATA16.602_Gr...ingDB - Diagram_0

```
SELECT count(*)
  FROM [602_Group2_StagingDB].[dbo].[CCount]
```

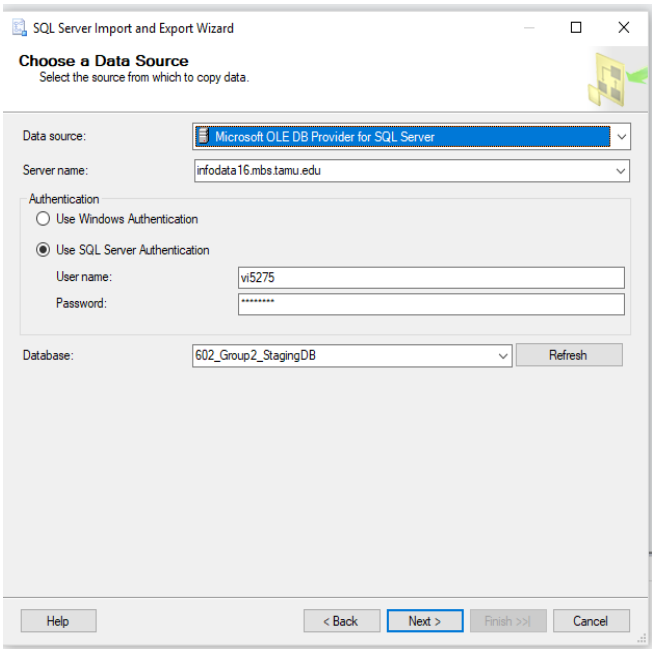
100 %

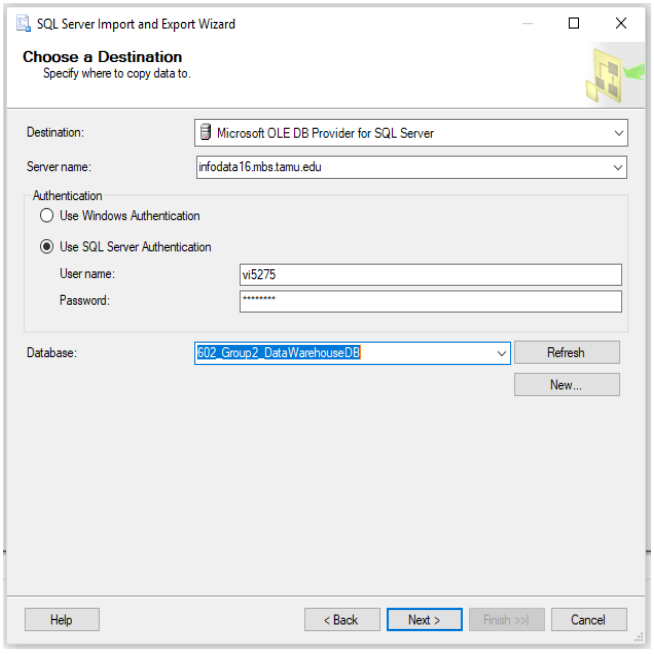
Results | Messages

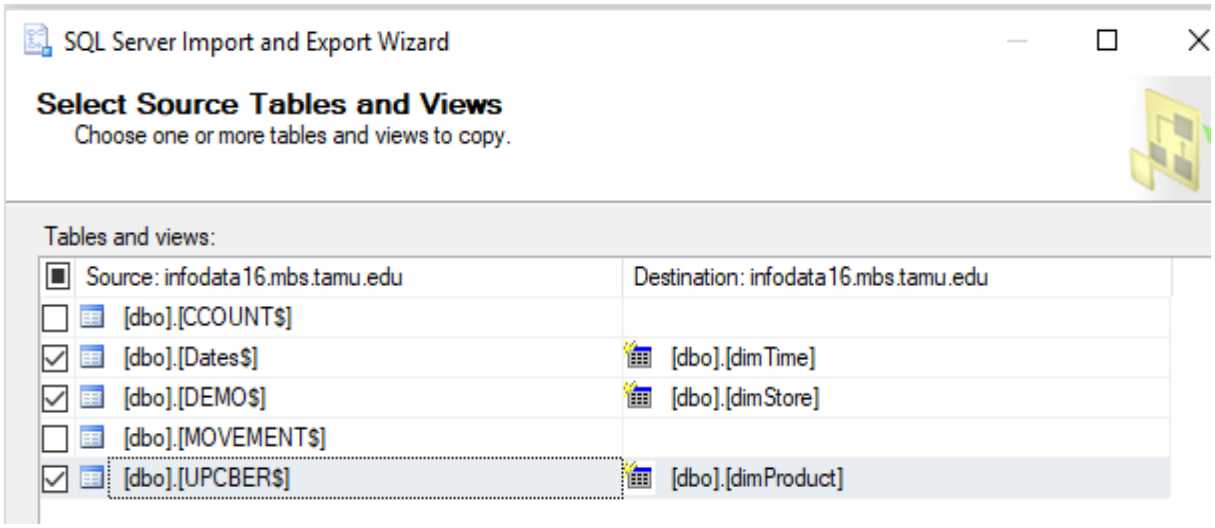
	(No column name)
1	310757

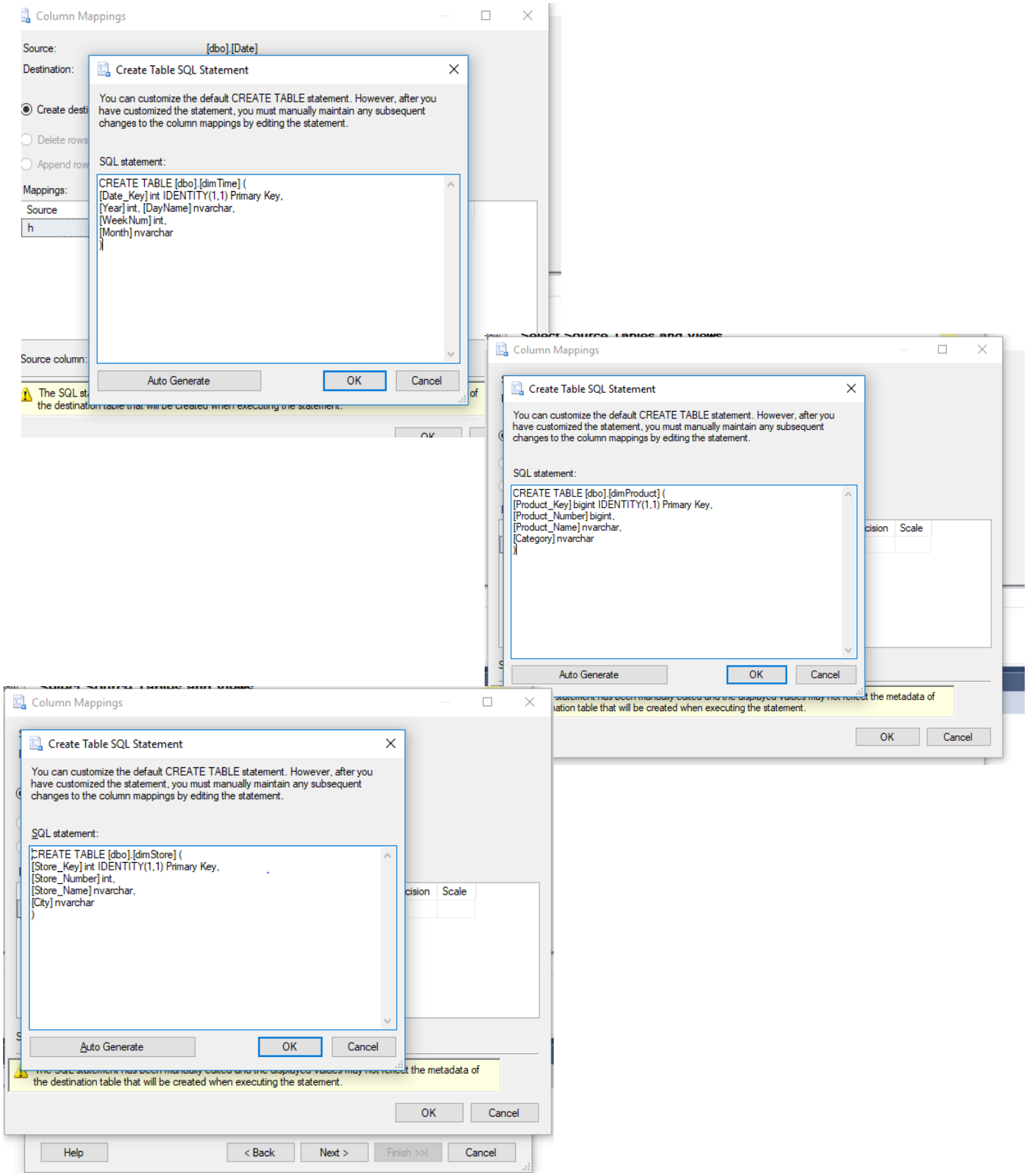
Data loading of Dimension Tables from staging DB area:
Each table extracted from source file contains data at granular level and is hence moved as-is into the staging as well as data warehouse as no further reduction in data level is necessary.

Loading Date, Demo and UPC tables into the data warehouse DB as dimTime, dimStore, dimProduct tables -








Column Mappings
Source:
[dbo].[Date]
Destination:
Create Table SQL Statement
You can customize the default CREATE TABLE statement. However, after you have customized the statement, you must manually maintain any subsequent changes to the column mappings by editing the statement.
SQL statement:
CREATE TABLE [dbo].[dimTime] (
[Date_Key] int IDENTITY(1,1) Primary Key,
[Year] int, [DayName] nvarchar,
[WeekNum] int,
[Month] nvarchar
)
Auto Generate
OK
Cancel
Mappings:
Source
Source column:
Column Mappings
Create Table SQL Statement
You can customize the default CREATE TABLE statement. However, after you have customized the statement, you must manually maintain any subsequent changes to the column mappings by editing the statement.
SQL statement:
CREATE TABLE [dbo].[dimProduct] (
[Product_Key] bigint IDENTITY(1,1) Primary Key,
[Product_Number] bigint,
[Product_Name] nvarchar,
[Category] nvarchar
)
Auto Generate
OK
Cancel
Scale
OK
Cancel
Column Mappings
Create Table SQL Statement
You can customize the default CREATE TABLE statement. However, after you have customized the statement, you must manually maintain any subsequent changes to the column mappings by editing the statement.
SQL statement:
CREATE TABLE [dbo].[dimStore] (
[Store_Key] int IDENTITY(1,1) Primary Key,
[Store_Number] int,
[Store_Name] nvarchar,
[City] nvarchar
)
Auto Generate
OK
Cancel
Scale
the destination table that will be created when executing the statement.
OK
Cancel
Help
< Back
Next >
Finish >>|
Cancel

Dimension table outputs post data loading process

dimProduct

SQLQuery5.sql - inf...seDB (vi5275 (111)) | INFODATA16.602_Gr...useDB - Diagram_0*

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Product_Key]
      ,[UPC_ID]
      ,[Product_Name]
      ,[Product_Category]
  FROM [602_Group2_DataWarehouseDB].[dbo].[dimProduct]
```

100 %

Results | Messages

	Product_Key	UPC_ID	Product_Name	Product_Category
1	1	294	BEER LIMIT	Beer
2	2	307	HEINEKEN KINGSIZE CA	Beer
3	3	710	BUDWEISER BEER	Beer
4	4	711	BUDWEISER DRY BEER	Beer
5	5	712	BUDWEISER LIGHT BEER	Beer
6	6	720	COORS BEER	Beer
7	7	721	COORS EXTRA GOLD BEE	Beer
8	8	723	KEYSTONE REGULAR BEE	Beer
9	9	731	MILLER HIGH LIFE PAR	Beer
10	10	732	MILLER LITE BEER	Beer
11	11	735	MILWAUKEE'S BEST BEE	Beer

SQLQuery11.sql - in...useDB (vi5275 (69))* | SQLQuery10.sql - in...seDB (

```
/****** Script for SelectTopNRows command from SSMS  *
SELECT count(*)
  FROM [602_Group2_DataWarehouseDB].[dbo].[dimProduct]
```

100 %

Results | Messages

	(No column name)
1	1248

dimStore

SQLQuery6.sql - inf...seDB (vi5275 (169))* | SQLQuery5.sql - inf...seDB (vi5275 (111))*

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT count(*)
  FROM [602_Group2_DataWarehouseDB].[dbo].[dimStore]
```

100 %

Results | Messages

	(No column name)
1	113

SQLQuery6.sql - inf...seDB (vi5275 (169)) | SQLQuery5.sql - inf...seDB (vi5275 (111))*

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Store_Key]
      ,[STORE_ID]
      ,[NAME]
      ,[CITY]
  FROM [602_Group2_DataWarehouseDB].[dbo].[dimStore]
```

100 %

Results | Messages

	Store_Key	STORE_ID	NAME	CITY
1	1	0	DOMINICS 138	olay
2	2	2	DOMINICKS 2	RIVER FOREST
3	3	4	DOMINICKS 4	PARK RIDGE
4	4	5	DOMINICKS 5	PALATINE
5	5	8	DOMINICKS 8	OAK LAWN
6	6	9	DOMINICKS 9	MORTON GROVE
7	7	12	DOMINICKS 12	CHICAGO
8	8	14	DOMINICKS 14	GLENVIEW
9	9	18	DOMINICKS 18	RIVER GROVE
10	10	19	NULL	NULL

dimTime

SQLQuery7.sql - inf...seDB (vi5275 (193)) | SQLQuery6.sql - inf...seDB (vi5275 (169))* | SQLQuery5.sql - inf...seDB (vi5275 (111))* | INFODATA16.602_Gr...useDB -

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Time_Key]
      ,[DateNum]
      ,[Date]
      ,[YearMonthNum]
      ,[MonthNum]
      ,[MonthName]
      ,[WeekNum]
      ,[DayNumOfYear]
      ,[DayNumOfMonth]
      ,[DayNumOfWeek]
      ,[DayName]
      ,[Year]
  FROM [602_Group2_DataWarehouseDB].[dbo].[dimTime]
```

100 %

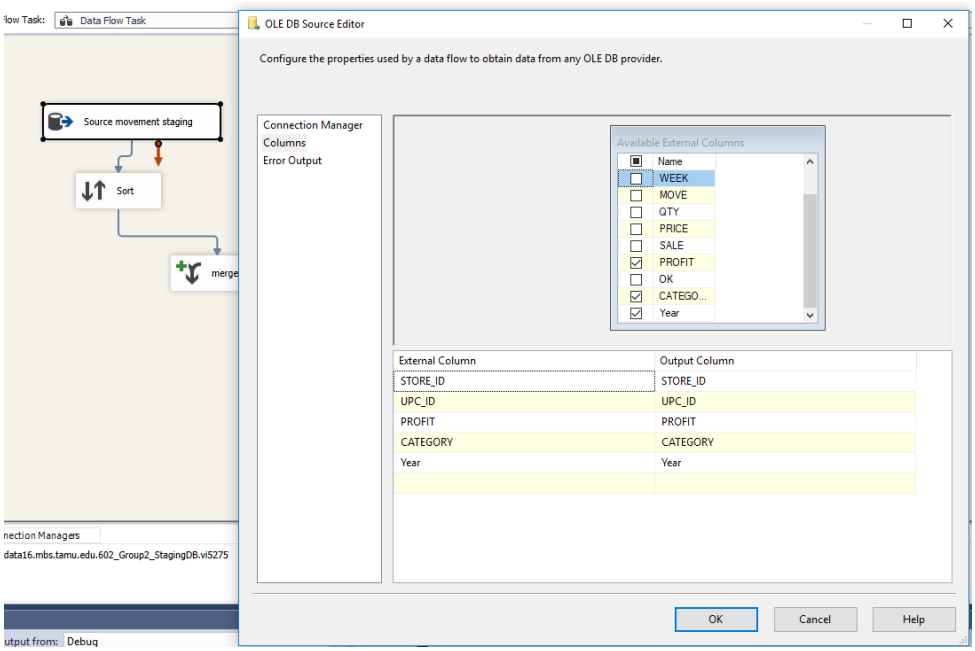
Results | Messages

	Time_Key	DateNum	Date	YearMonthNum	MonthNum	MonthName	WeekNum	DayNumOfYear	DayNumOfMonth	DayNumOfWeek	DayName	Year
1	1	19910101	1991-01-01	199101	1	January	1	1	1	3	Tuesday	1991
2	2	19910102	1991-01-02	199101	1	January	1	2	2	4	Wednesday	1991
3	3	19910103	1991-01-03	199101	1	January	1	3	3	5	Thursday	1991
4	4	19910104	1991-01-04	199101	1	January	1	4	4	6	Friday	1991
5	5	19910105	1991-01-05	199101	1	January	1	5	5	7	Saturday	1991
6	6	19910106	1991-01-06	199101	1	January	1	6	6	1	Sunday	1991
7	7	19910107	1991-01-07	199101	1	January	2	7	7	2	Monday	1991
8	8	19910108	1991-01-08	199101	1	January	2	8	8	3	Tuesday	1991
9	9	19910109	1991-01-09	199101	1	January	2	9	9	4	Wednesday	1991
10	10	19910110	1991-01-10	199101	1	January	2	10	10	5	Thursday	1991
11	11	19910111	1991-01-11	199101	1	January	2	11	11	6	Friday	1991
12	12	19910112	1991-01-12	199101	1	January	2	12	12	7	Saturday	1991
13	13	19910113	1991-01-13	199101	1	January	2	13	13	1	Sunday	1991
14	14	19910114	1991-01-14	199101	1	January	3	14	14	2	Monday	1991

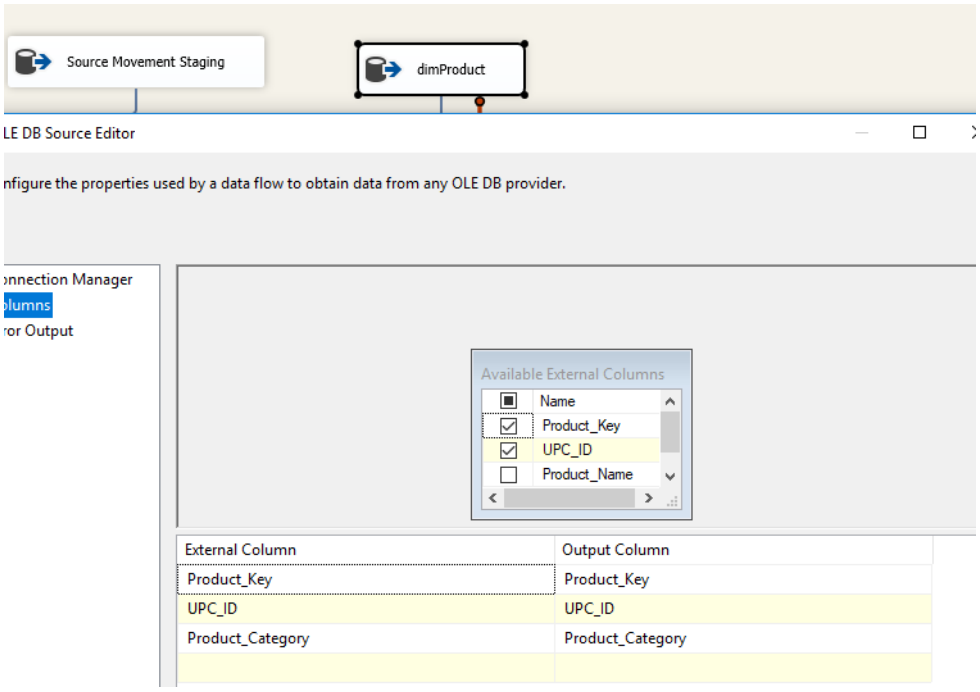
Following are the steps followed in SSIS for each Fact table

SSIS data flow task for creating FactfrozenGrossProfit table and factBeerSales table: it is done in one task as both fact tables have similar structure:

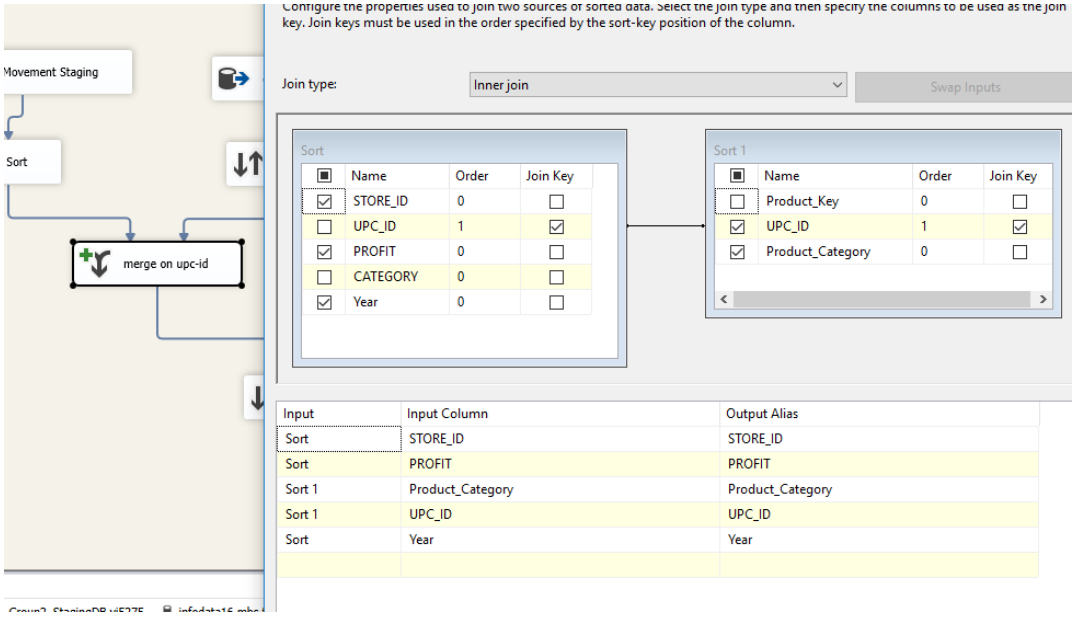
i. Add movement table from staging area as OLE DB source and sort on Store_ID. Select Store_ID, UPC_ID, Profit, Category and Year columns.



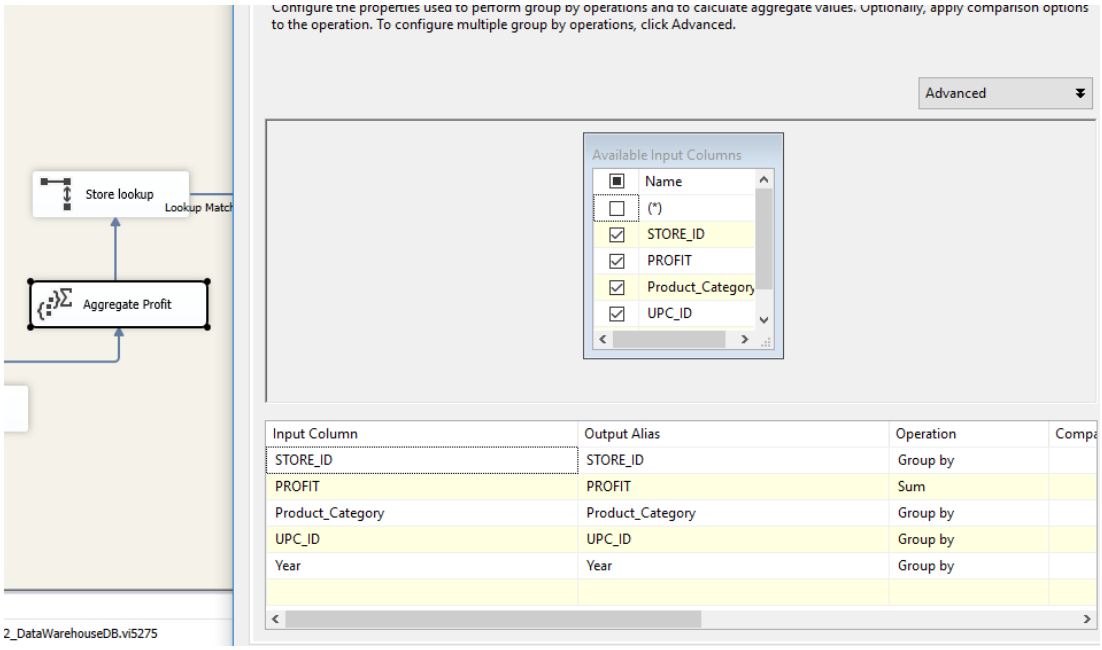
ii. Add dimProduct table as the OLE DB source from Datawarehouse DB. Select Product Key, UPC_ID, Product_Category columns. And Sort on Store_ID.



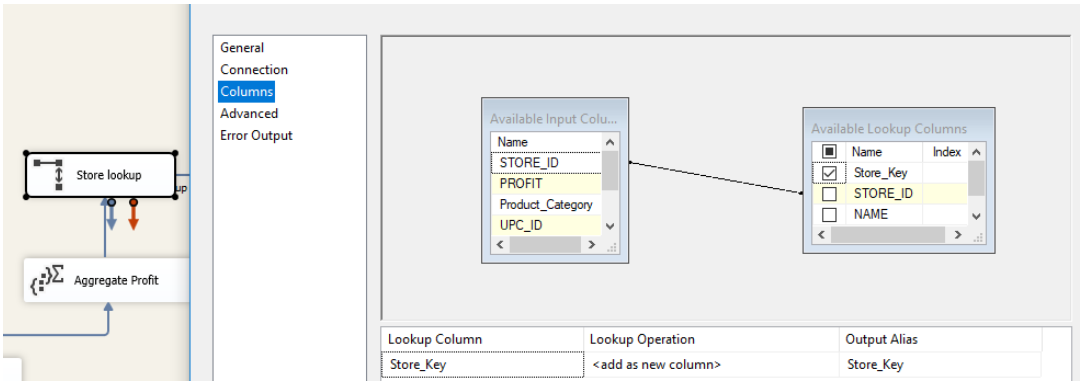
iii. Next, Merge join these tables on Store_ID. Select Store_ID, Profit, Year from movement table and select UPC_ID, Product_Category from dimProduct table.



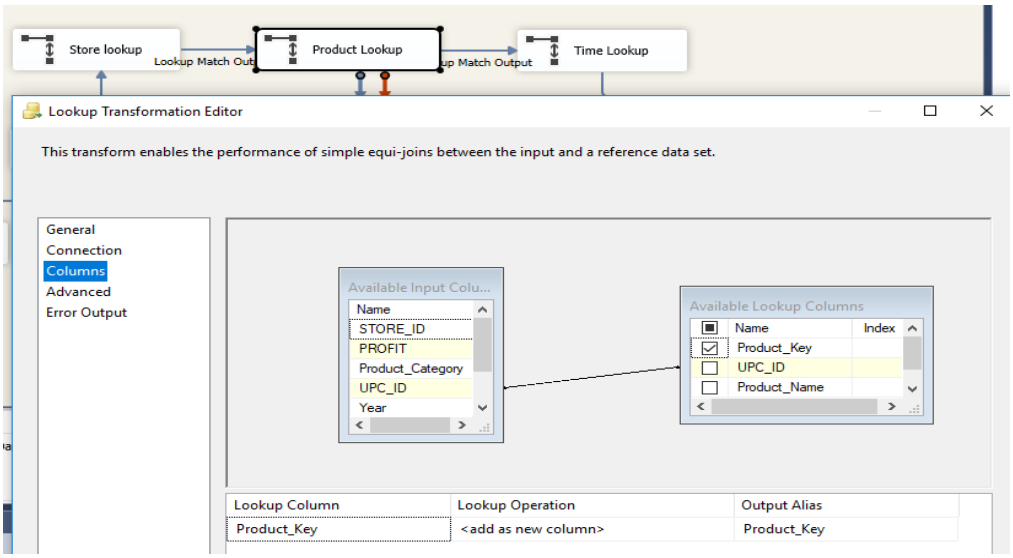
iv. Add Aggregate function and choose sum for profit and group_by the other attributes.



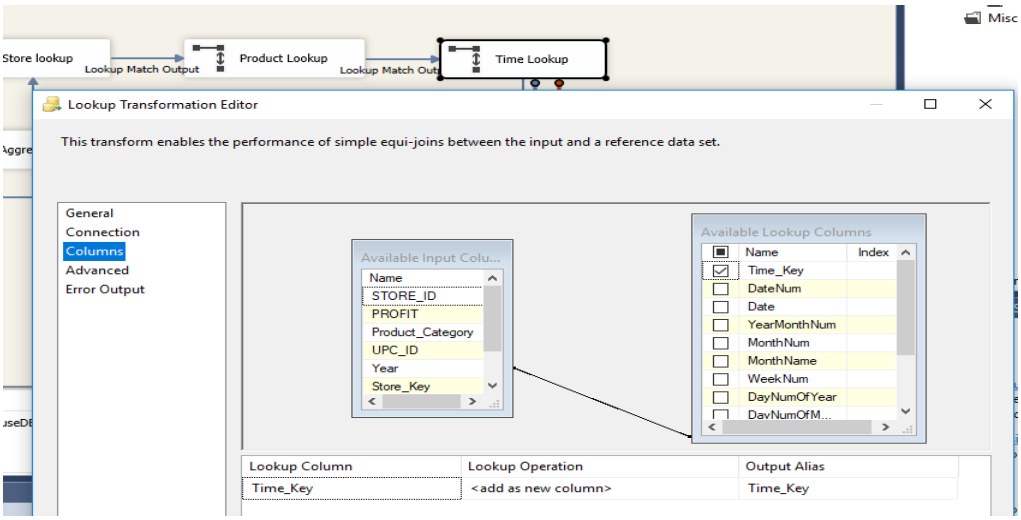
v. Next, lookup dimStore, join on Store_ID and select Store_Key



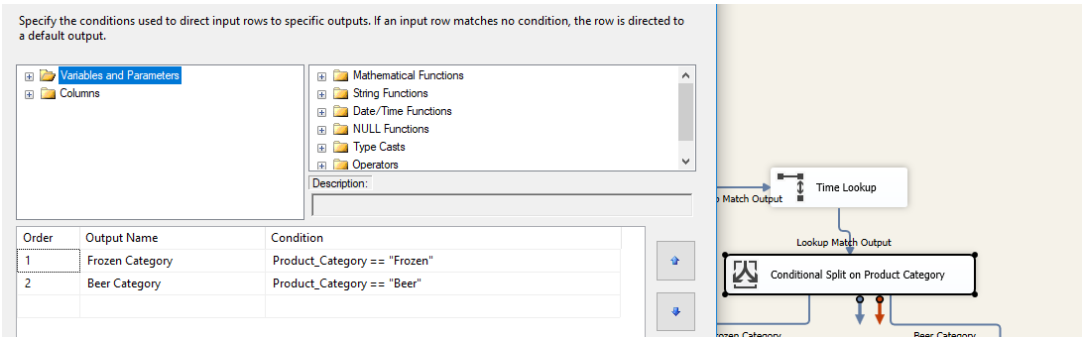
vi. Lookup dimProduct, join on UPC_ID and select Product_Key



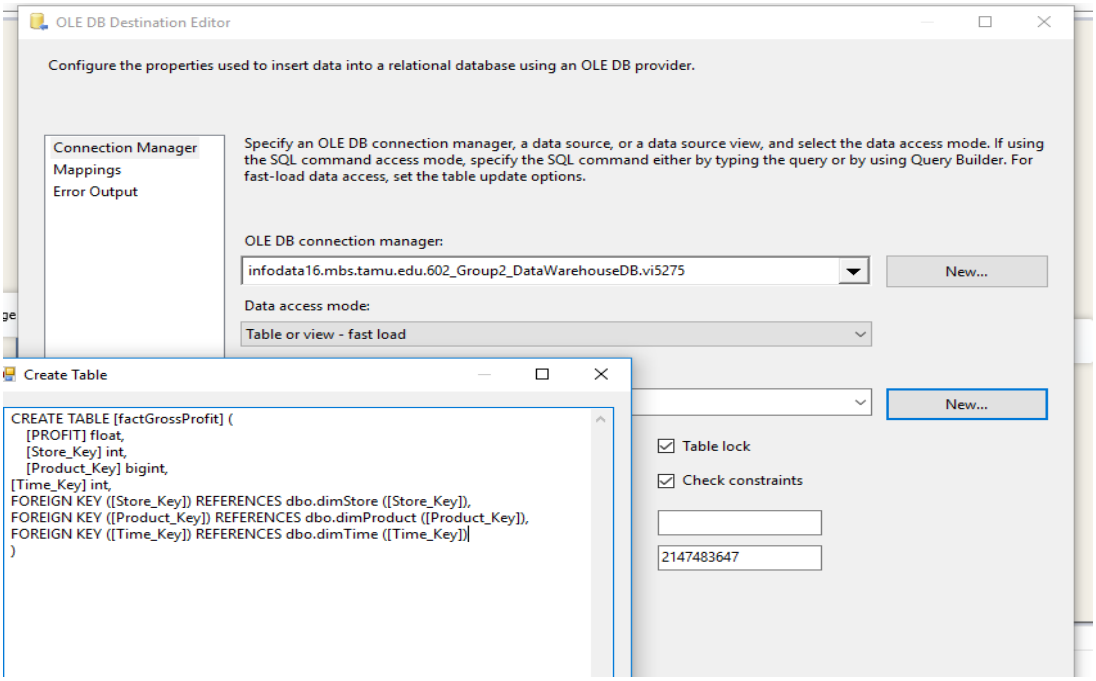
vii. Lookup dimTime, join on year and select Time_Key.



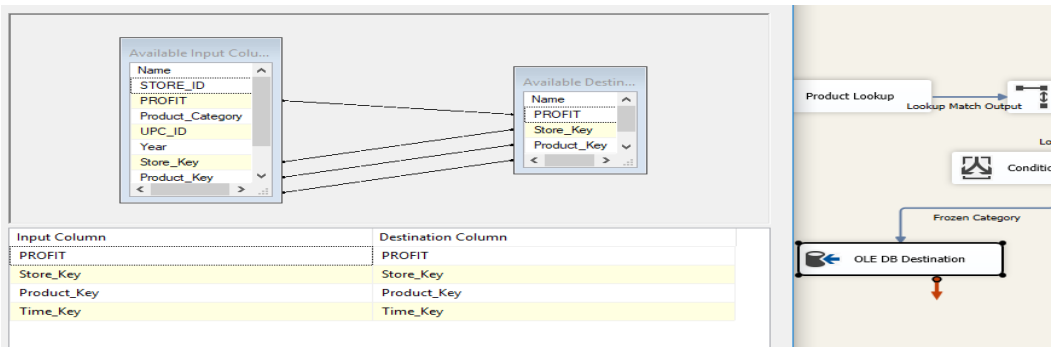
viii. Use conditional split on the previous output based on Product_Category. Set 1 will have frozen products and set 2 will have products belonging to category beer.



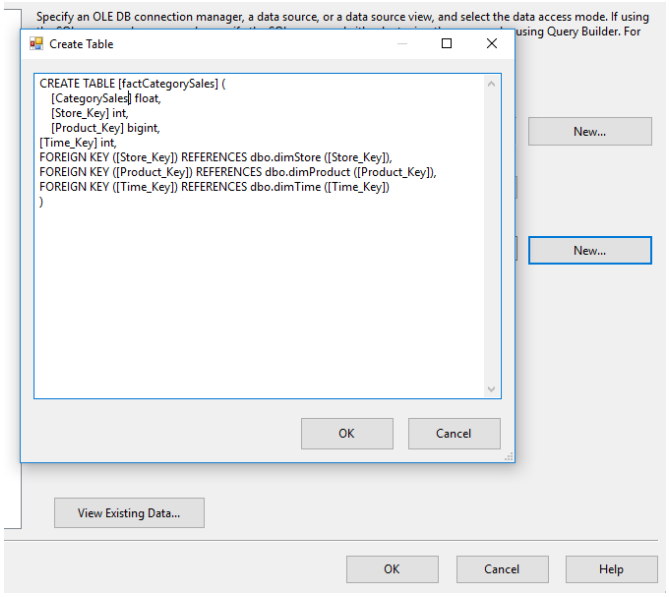
ix. Add OLE DB source, connect to data warehouse DB and create factFrozenGrossProfit table. This will contain gross profit of frozen products.



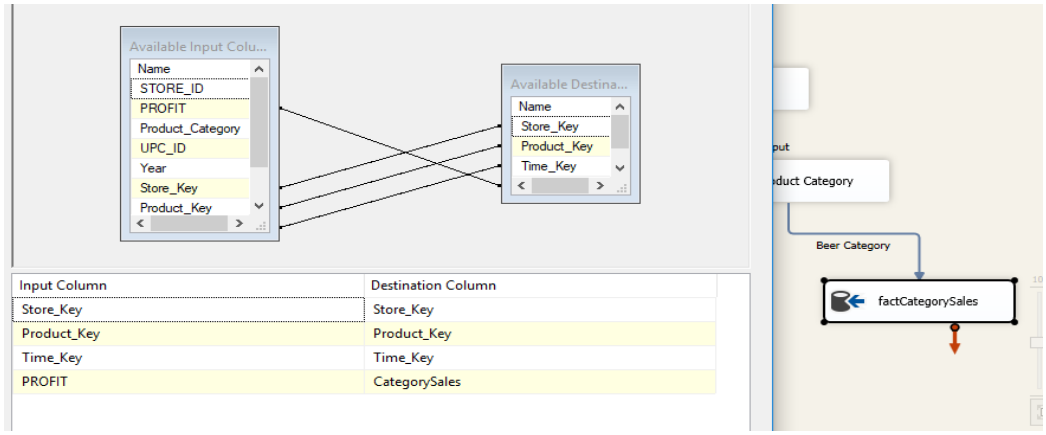
Make sure all the attributes have correct data types and add all the relevant referential keys. This step will create the fact table in 602_Group2_DatawarehouseDB.



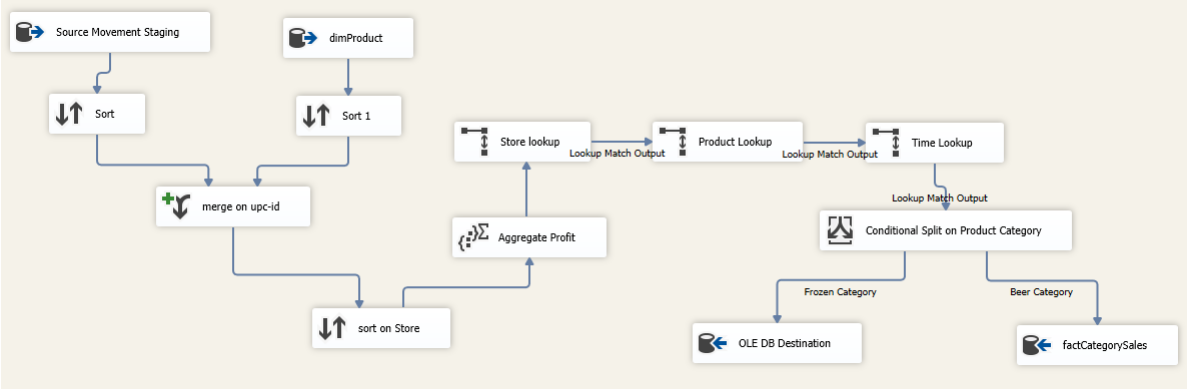
x. Add another OLE DB Source, connect to data warehouse DB and create factBeerSales.



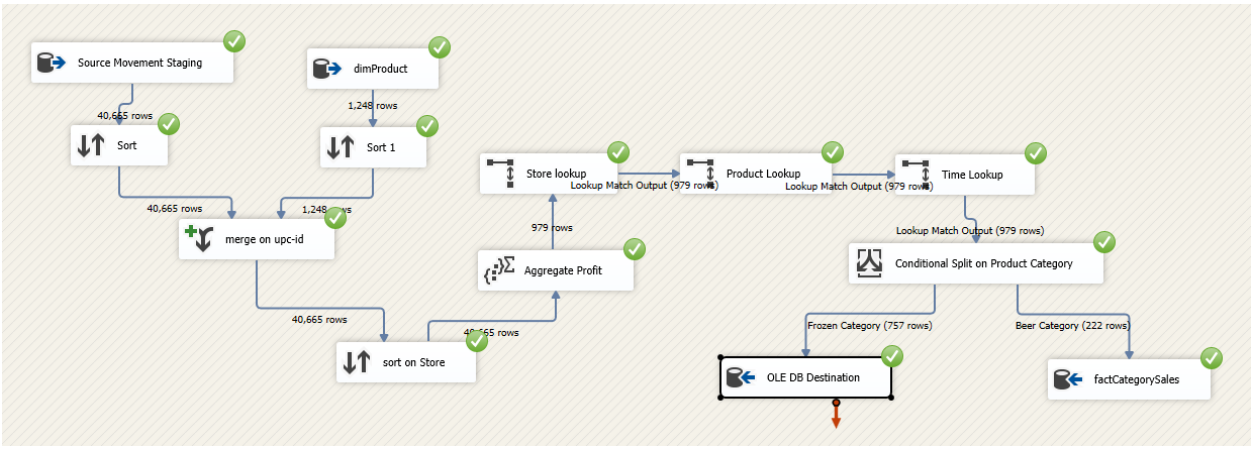
xi. Edit SQL to alter the query to contain all the necessary referential keys. This step will create factBeerSales in 602_Group2_DatawarehouseDB. This contains the sales for Beer. Make sure that the mappings from the output to fact table are correct.



ix. The package will now look like this



x. Execute the package



We can verify that factFrozenGrossProfit and factBeerSales will be created and populated will relevant data.

factFrozenGrossProfit and factBeerSales

SQLQuery8.sql - inf...seDB (vi5275 (133)) | SQLQuery7.sql - inf...seDB (vi5275 (193))*

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [PROFIT]
      ,[Store_Key]
      ,[Product_Key]
      ,[Time_Key]
  FROM [602_Group2_DataWarehouseDB].[dbo].[factFrozenGrossProfit]
```

100 %

Results | Messages

	PROFIT	Store_Key	Product_Key	Time_Key
1	1589.16	17	22	17899
2	1405.31	17	22	17900
3	1993.17	17	22	17901
4	2158.27	17	22	17902
5	1480.34	17	22	1
6	1093.73	17	22	366
7	1140.62	17	22	732
8	743.43	17	22	1097
9	711.96	60	26	366
10	1274.09	60	26	732
11	908.18	60	26	1097
12	1826.97	73	22	17899
13	1612.88	73	22	17900
14	2385.3	73	22	17901
15	2318.77	73	22	17902
16	1918.06	73	22	1
17	1563.3	73	22	366
18	1508.89	73	22	732
19	976.43	73	22	1097
20	508.83	40	26	366
21	1050.4	40	26	732
22	891.18	40	26	1097

```
SELECT TOP (1000) [CategorySales]
      ,[Store_Key]
      ,[Product_Key]
      ,[Time_Key]
  FROM [602_Group2_DataWarehouseDB].[dbo].[factBeerSales]
```

100 %

Results | Messages

	CategorySales	Store_Key	Product_Key	Time_Key
1	-50.81	21	1	366
2	0	21	1	732
3	0	21	1	1097
4	-3.45	91	13	366
5	10.02	42	1	366
6	0	42	1	732
7	0	42	1	1097
8	-27.46	49	1	366
9	0	49	1	732
10	0	49	1	1097
11	-61.61	56	1	366
12	0	56	1	732
13	0	56	1	1097
14	10.26	69	15	366
15	-8	9	4	366
16	-14.37	80	10	366
17	0	69	16	17900

Query executed successfully.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT count(*)
  FROM [602_Group2_DataWarehouseDB].[dbo].[factFrozenGrossProfit]
```

100 %

Results | Messages

	(No column name)
1	757

```
/****** Script for SelectTopNRows command from SSMS  ****
SELECT count(*)
  FROM [602_Group2_DataWarehouseDB].[dbo].[factBeerSales]
```

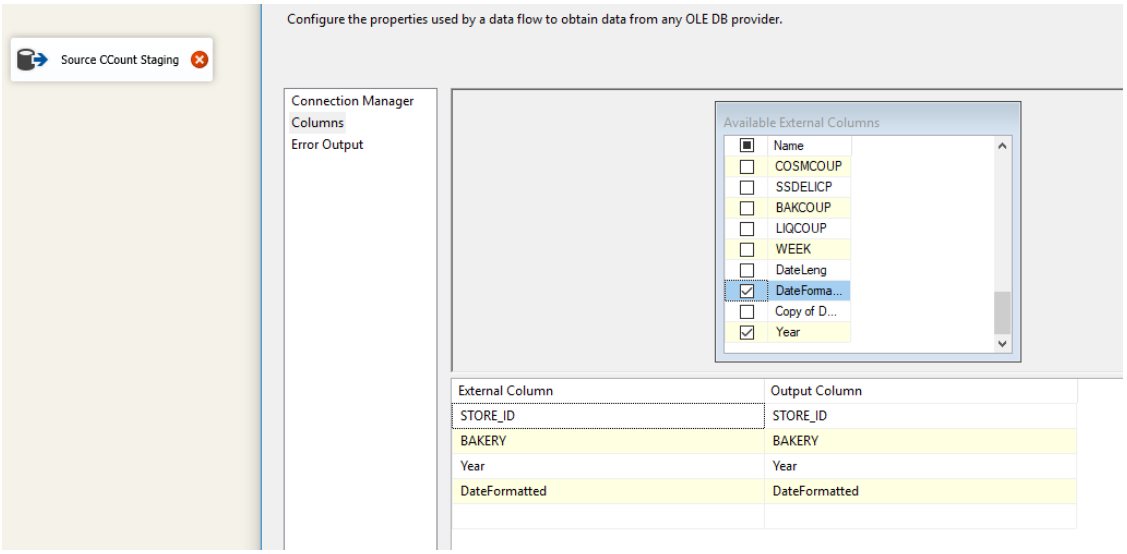
100 %

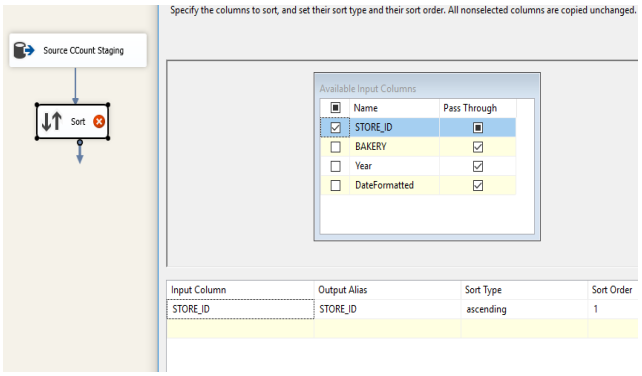
Results | Messages

	(No column name)
1	222

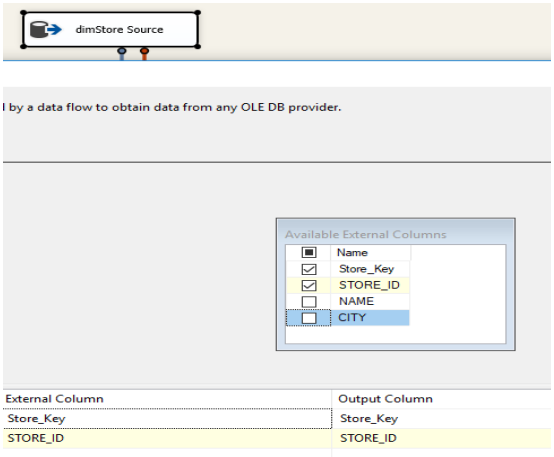
SSIS Data flow for creating the fact table factBakerySales:

i. Add an OLE DB Source in the data flow tab. Connect to staging area 602_Group2_StagingDB and select CCount table. We only selected necessary columns – Year, Bakery, Store_ID, DateFormatted. After this, sort on Store_ID.

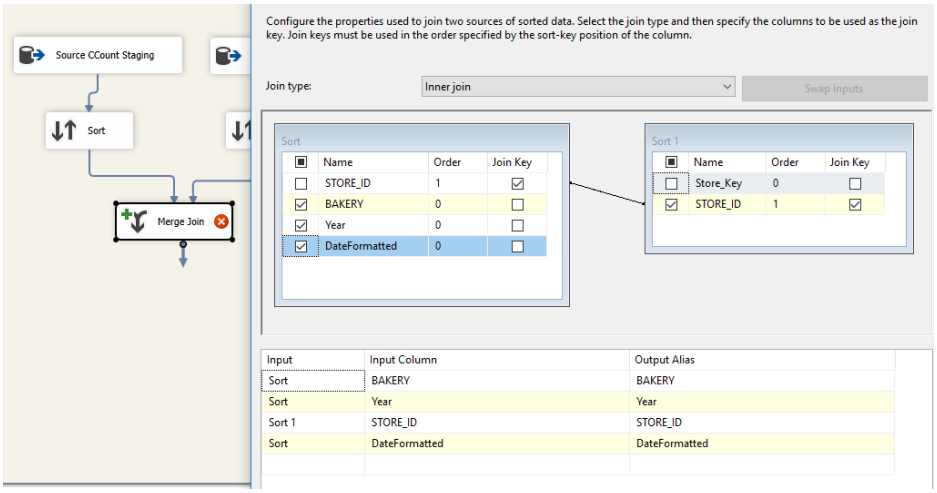




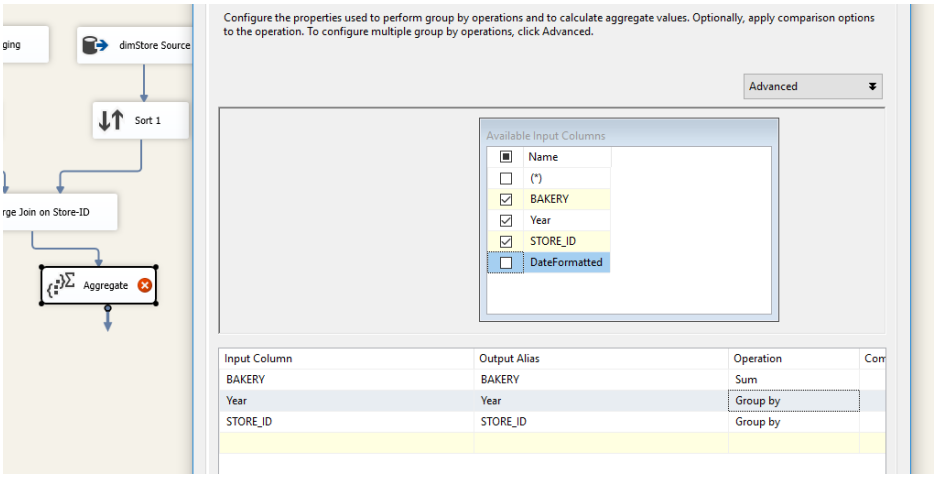
ii. Add dimStore table from data warehouse table 602_Group2_DatawarehouseDB. Select Store_Key and Store_ID and sort on Store_ID



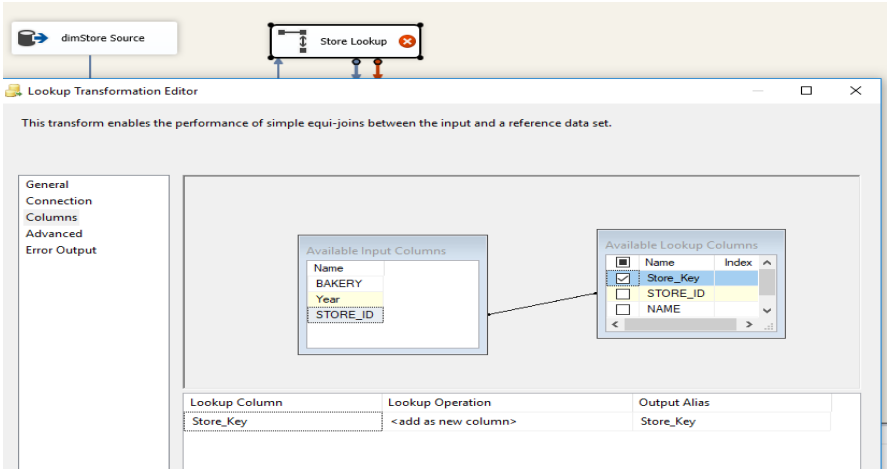
iii. Next, merge the sorted outputs by joining on Store_ID. Select Bakery, Year, DateFormatted from CCount table and select Store_ID from dimStore.



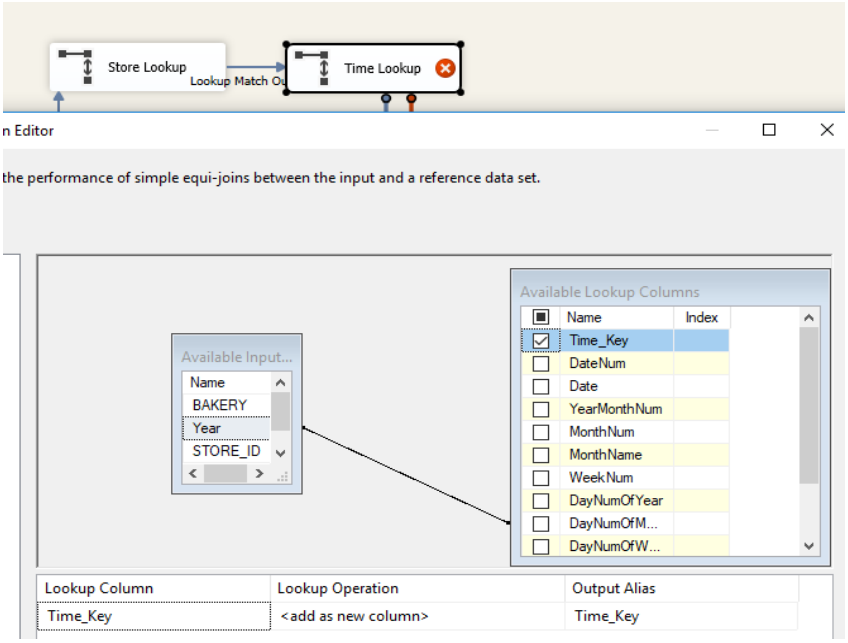
iv. Add an aggregate task, select sum function for bakery and group_by clause for the other attributes.



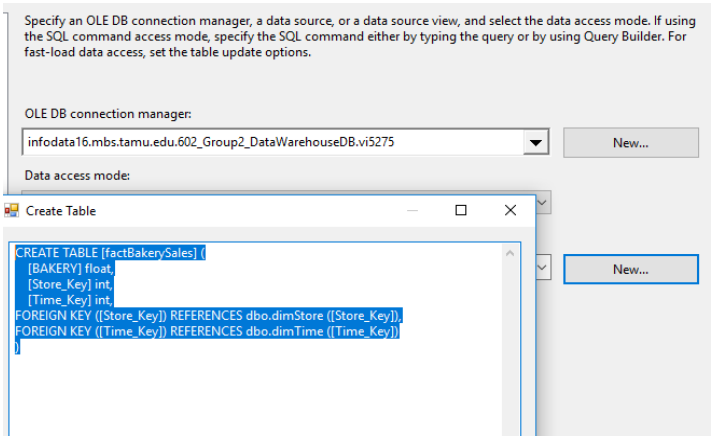
v. Lookup on dimStore; join on Store_ID and select Store_Key

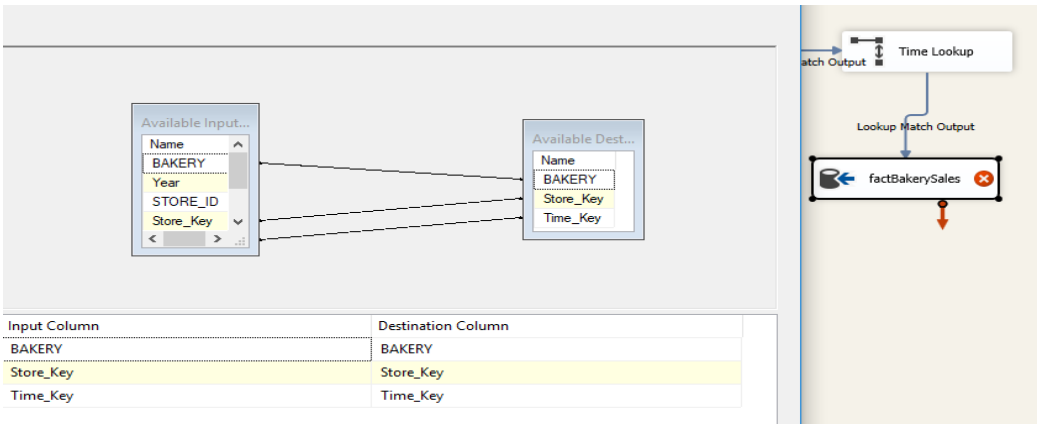


vi. Next, lookup on dimTime; join on Year and select Time_Key

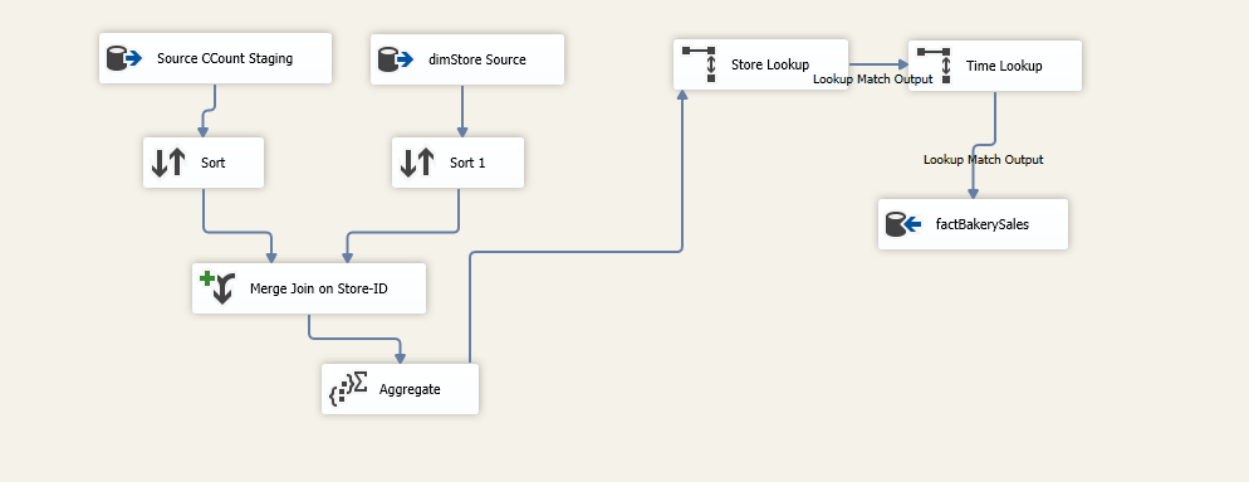


vii. Add OLE DB Destination to create factBakerySales table. Connect to 602_Group2_DatawarehouseDB, create new table called factBakerySales with relevant attributes and foreign keys. Also make sure to map the correct columns from source to destination.

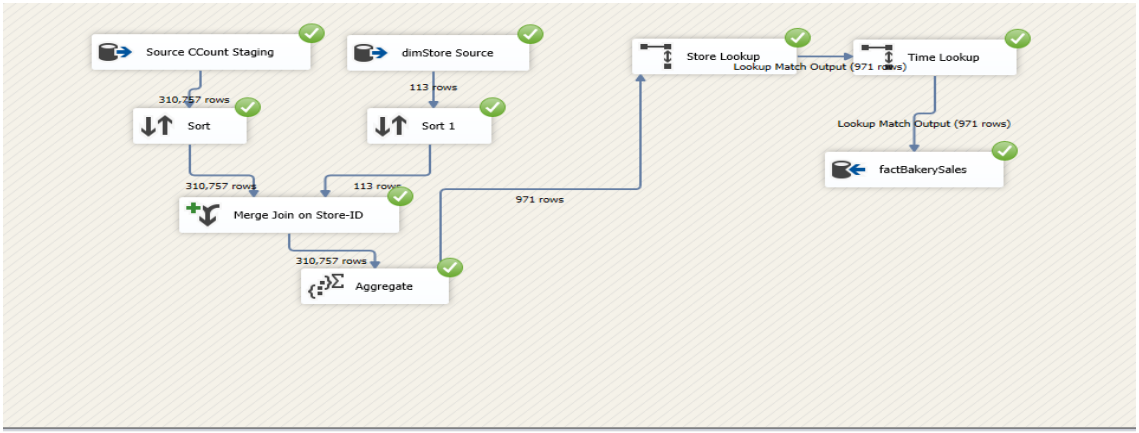


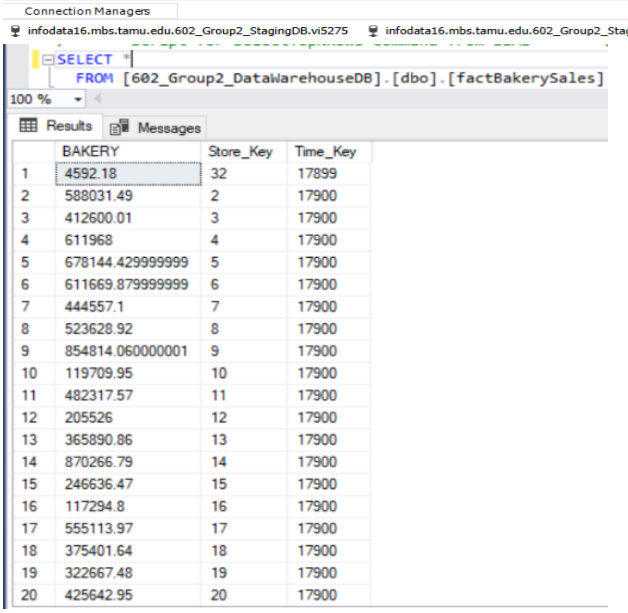


viii. The package now looks like this -



ix. Execute the package and verify the creation of table in SQL Server Management Studio.





Connection Managers

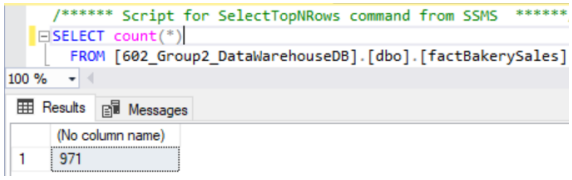
infodata16.mbs.tamu.edu.602_Group2_StagingDB.vi5275 | infodata16.mbs.tamu.edu.602_Group2_StagingDB.vi5275 1 | infodata16.mbs.tamu.edu.602_Group2_DataWarehouseDB.vi5275

```
SELECT *
  FROM [602_Group2_DataWarehouseDB].[dbo].[factBakerySales]
```

100 %

Results | Messages

	BAKERY	Store_Key	Time_Key
1	4592.18	32	17899
2	588031.49	2	17900
3	412600.01	3	17900
4	611968	4	17900
5	678144.429999999	5	17900
6	611669.879999999	6	17900
7	444557.1	7	17900
8	523628.92	8	17900
9	854814.060000001	9	17900
10	119709.95	10	17900
11	482317.57	11	17900
12	205526	12	17900
13	365890.86	13	17900
14	870266.79	14	17900
15	246636.47	15	17900
16	117294.8	16	17900
17	555113.97	17	17900
18	375401.64	18	17900
19	322667.48	19	17900
20	425642.95	20	17900



```
/****** Script for SelectTopNRows command from SSMS  ******
SELECT count(*)
  FROM [602_Group2_DataWarehouseDB].[dbo].[factBakerySales]
```

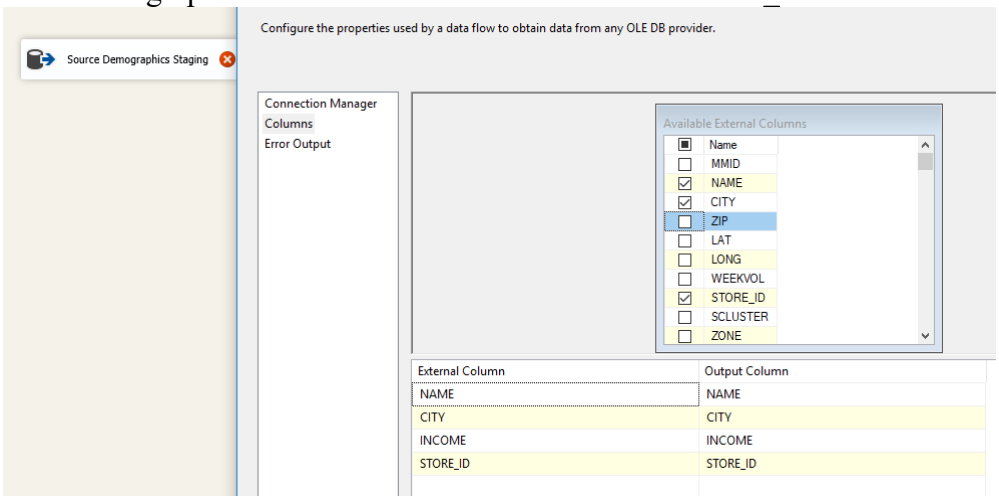
100 %

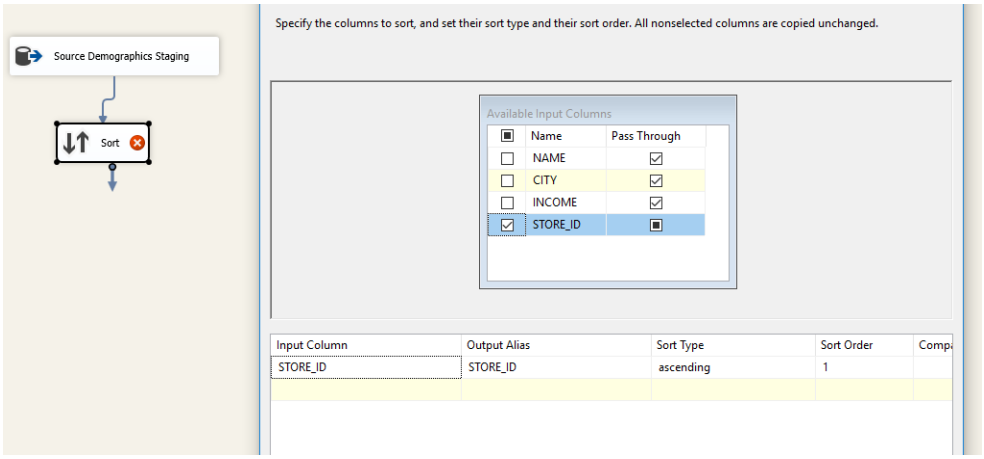
Results | Messages

	(No column name)
1	971

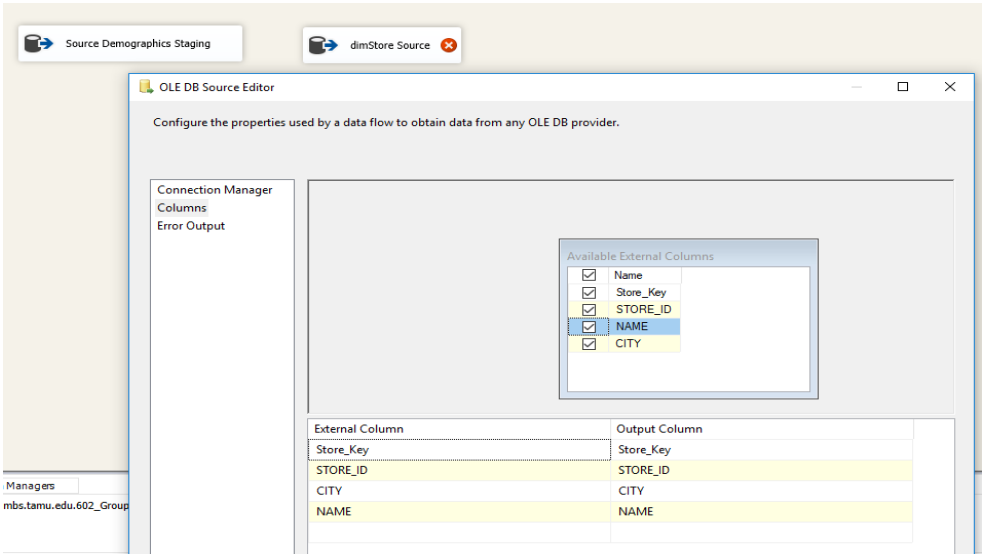
SSIS data flow for creating factHighestIncome:

i. Add Demographics table as OLE DB Source. Sort on Store_ID.

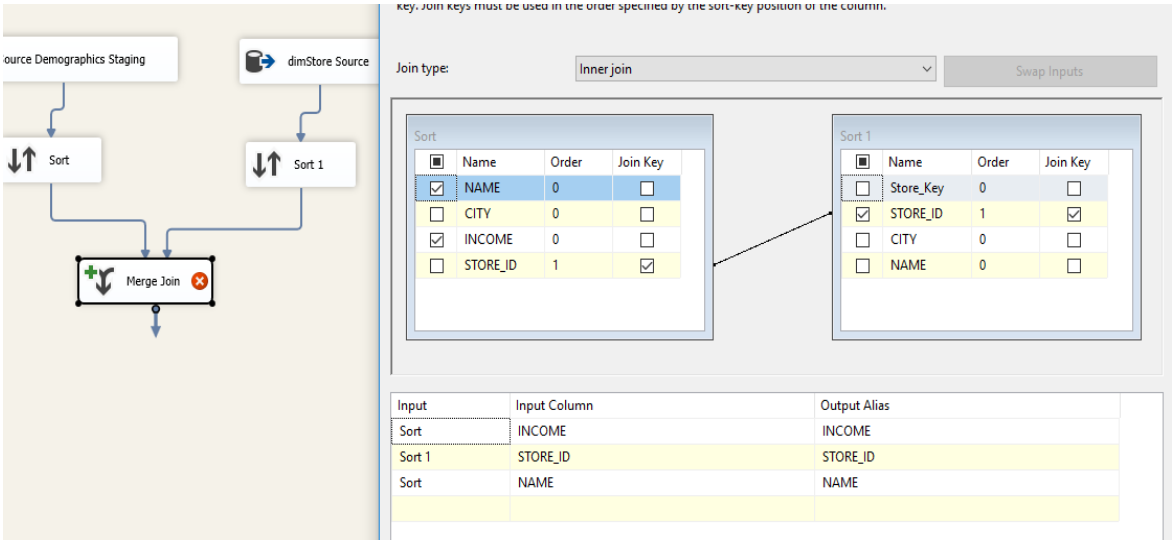




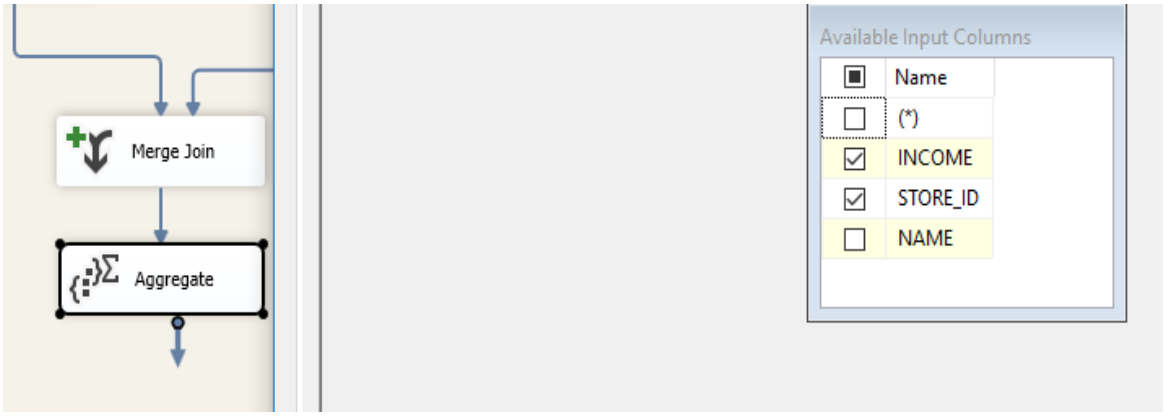
ii. Add dimStore as another OLE DB Source, and sort on Store_ID.



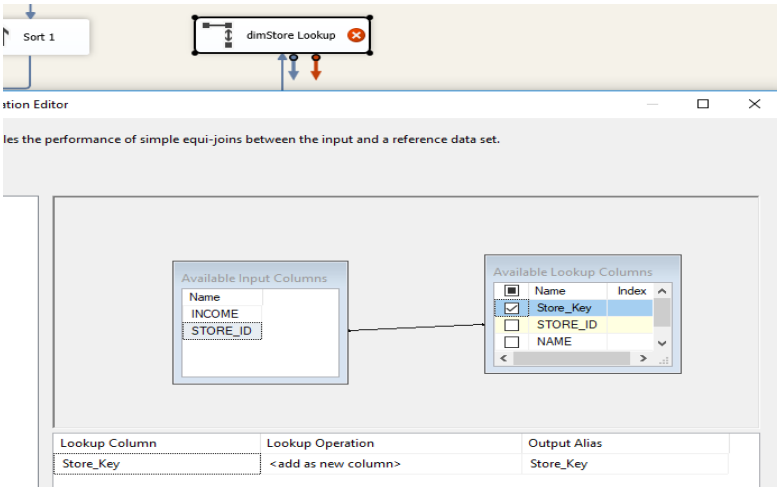
iii. Add a merge join task; merge the previous inputs on Store_ID. Select Name, Income, and Store_ID as follows-



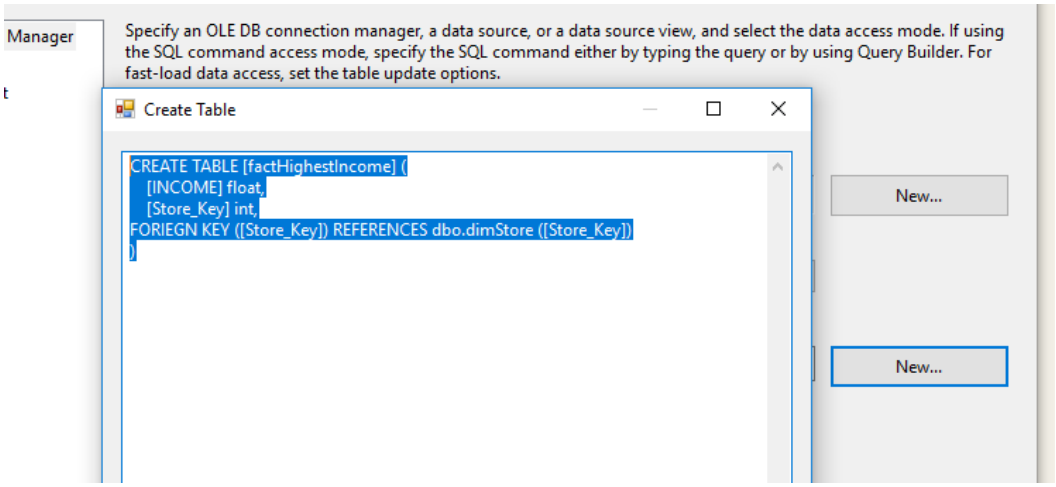
iv. Aggregate the merged output. Use sum function on income and group by clause on Store_ID.

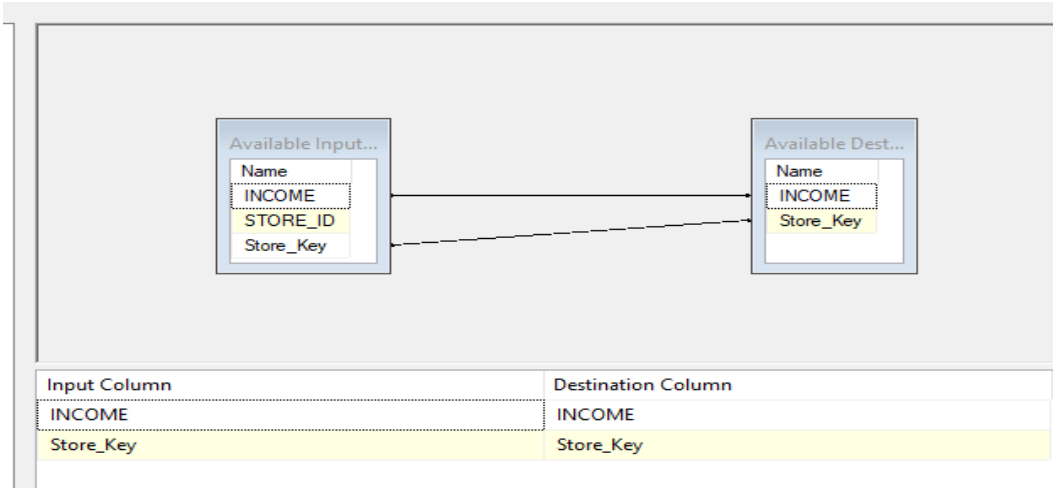


v. Lookup on dimStore. Join on Store_ID and select Store_Key.

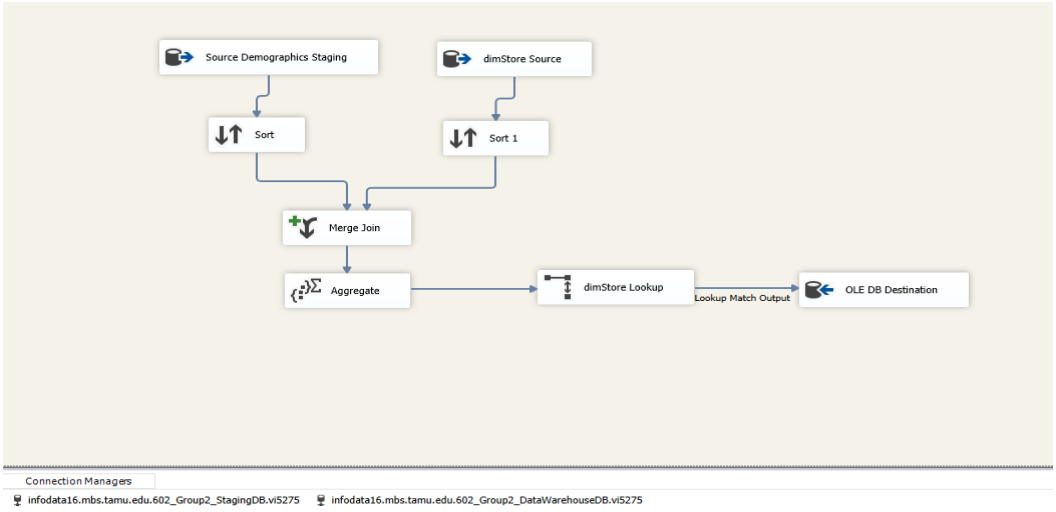


vi. Add an OLE DB Destination to create factHighestIncome table. Connect to data warehouse DB 602_Group2_DatawarehouseDB. Create a new fact table using SQL query. Select relevant columns and ensure the column mappings are correct.

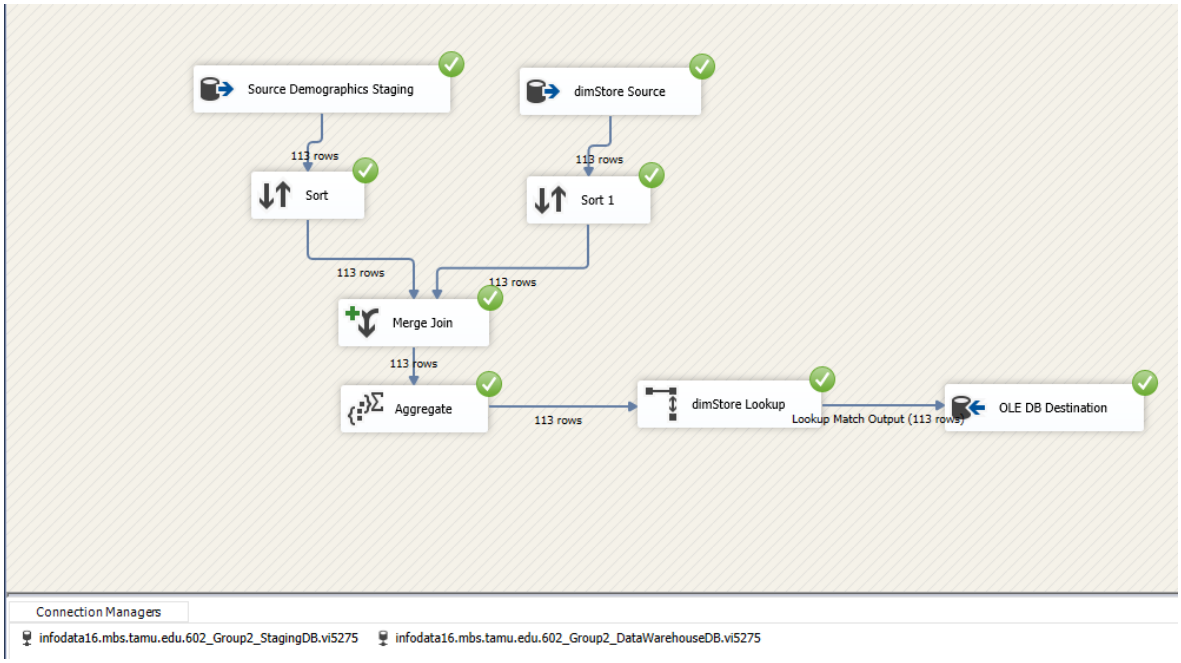




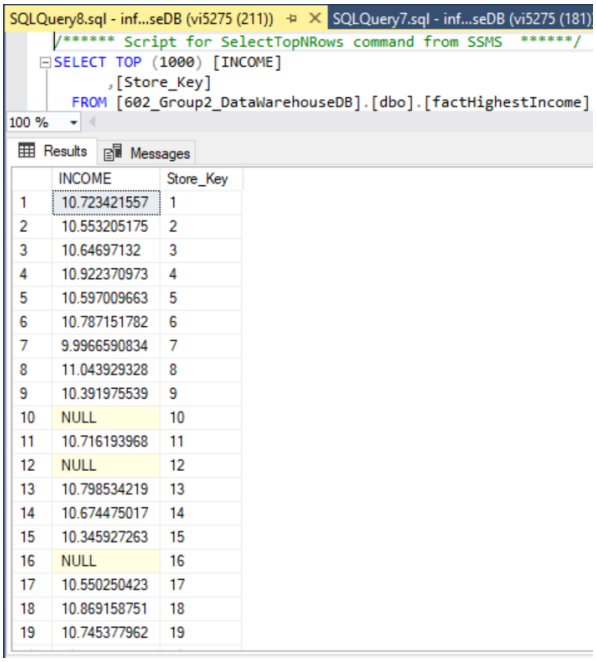
vii. The package now looks like this –



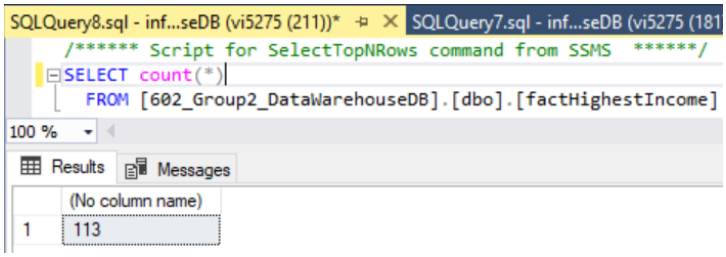
viii. Execute the package and verify the creation of table in the datawarehouse database.



Output of factHighestIncome:



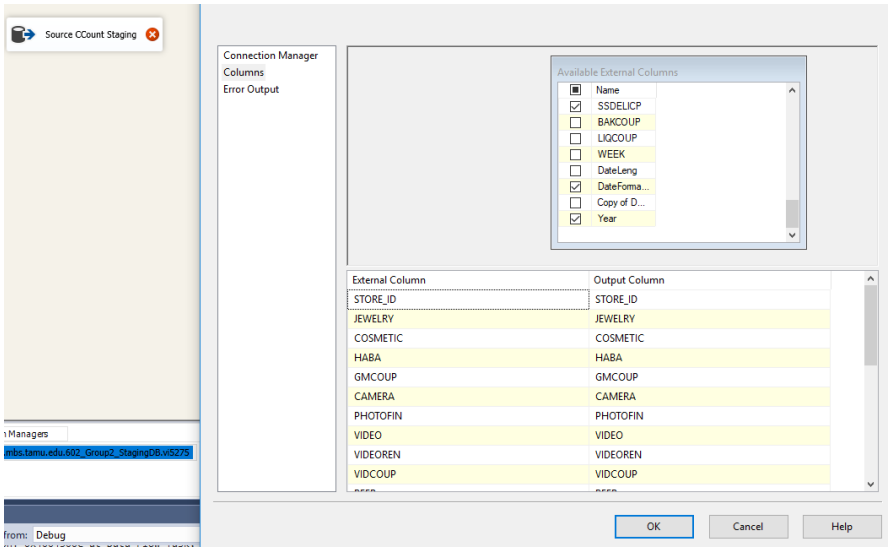
	INCOME	Store_Key
1	10.723421557	1
2	10.553205175	2
3	10.64697132	3
4	10.922370973	4
5	10.597009663	5
6	10.787151782	6
7	9.9966590834	7
8	11.043929328	8
9	10.391975539	9
10	NULL	10
11	10.716193968	11
12	NULL	12
13	10.798534219	13
14	10.674475017	14
15	10.345927263	15
16	NULL	16
17	10.550250423	17
18	10.869158751	18
19	10.745377962	19



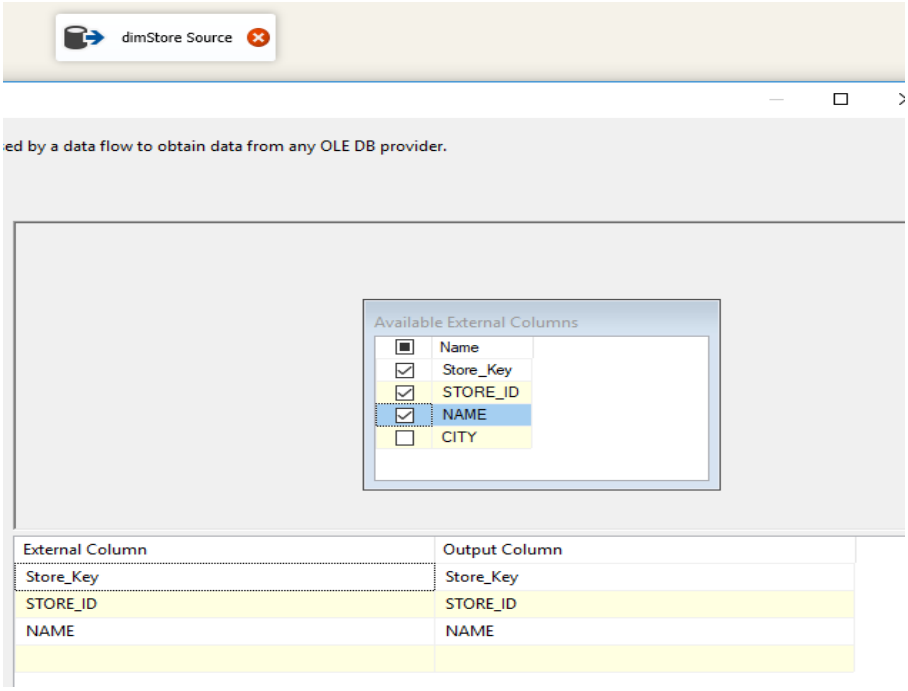
	(No column name)
1	113

SSIS data flow for creating the fact table factTopProducts:

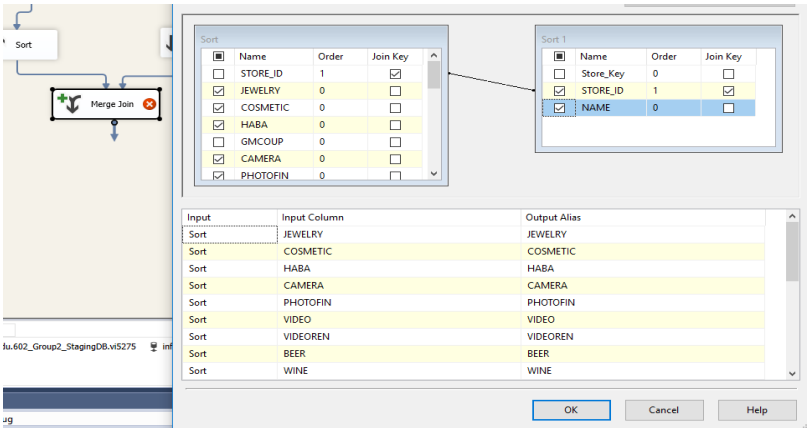
i. Add CCount table from Staging DB as OLE DB Source. Select all product categories. Next, sort this on Store_ID.



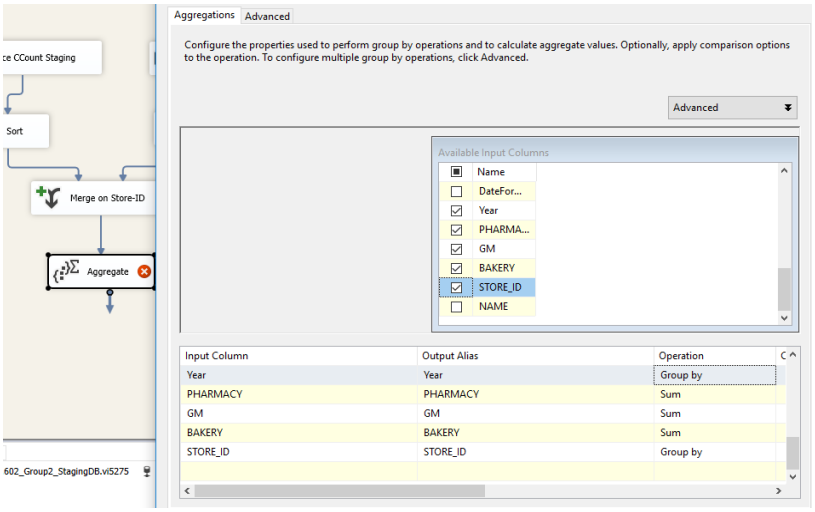
ii. Add dimStore from the data warehouse DB. Select Store_Key, Store_ID, Name. Sort this on Store_ID.



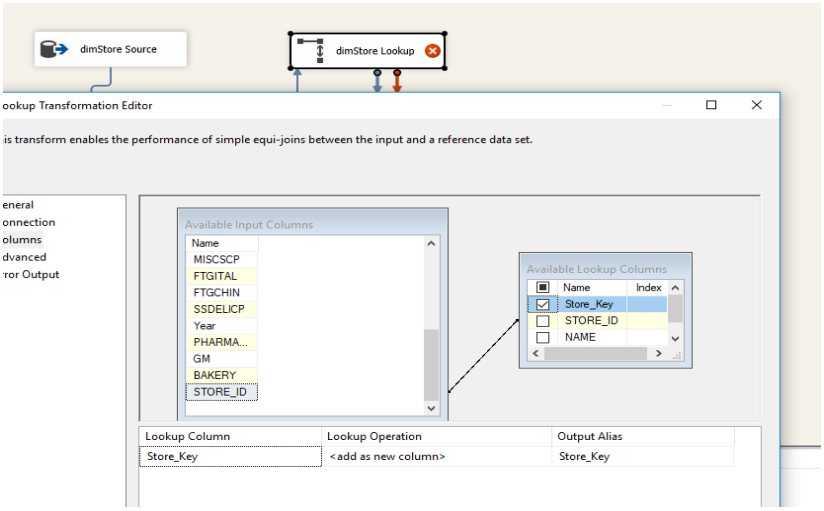
iii. Add merge join task. Join the sorted outputs on Store_ID. Select all product categories, year and Store_ID.



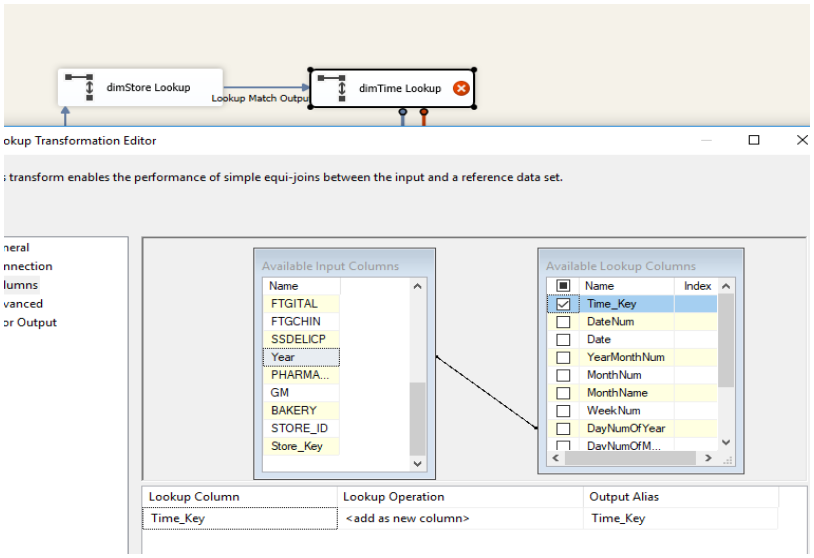
iv. Aggregate the output from merge join. Use sum function for all the product categories and group by year and Store_ID.



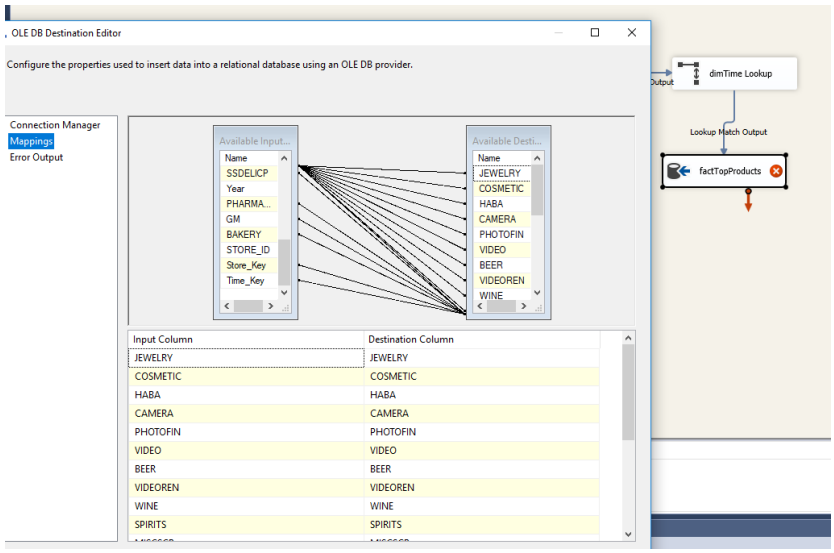
v. Lookup on dimStore. Join on Store_ID and select Store_Key.



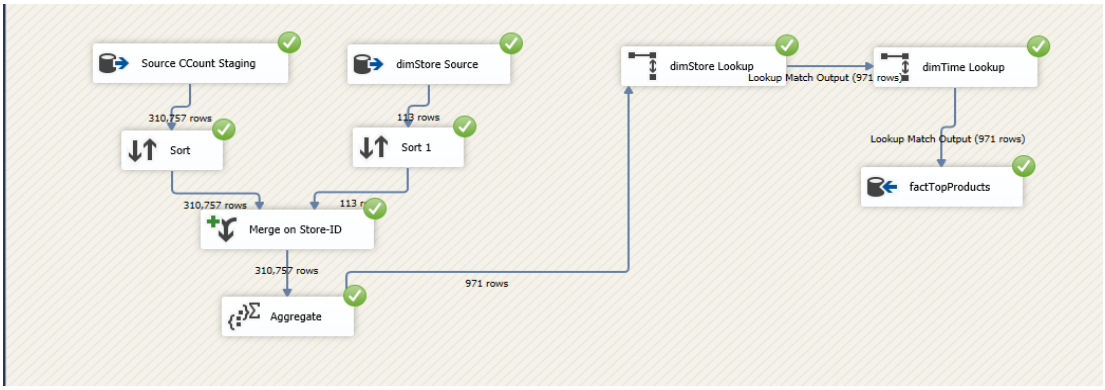
vi. Lookup on dimTime. Join on Year and select Time_Key.



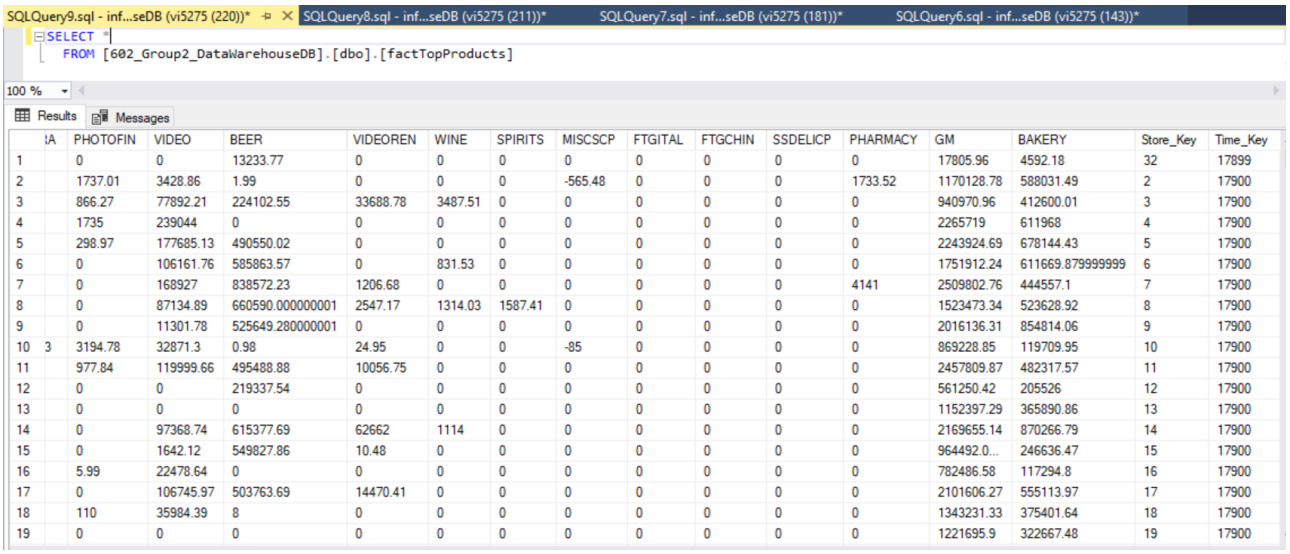
vii. Add OLE DB Destination to create factTopProducts. Connect to data warehouse DB 602_Group2_DatawarehouseDB and create new fact table – factTopProducts. Ensure correct column mappings and that there is correct foreign key references to dimStore and dimTime.



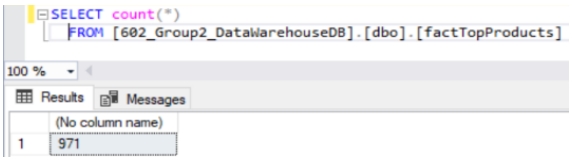
viii. Execute the package



Data output for factTopProducts:



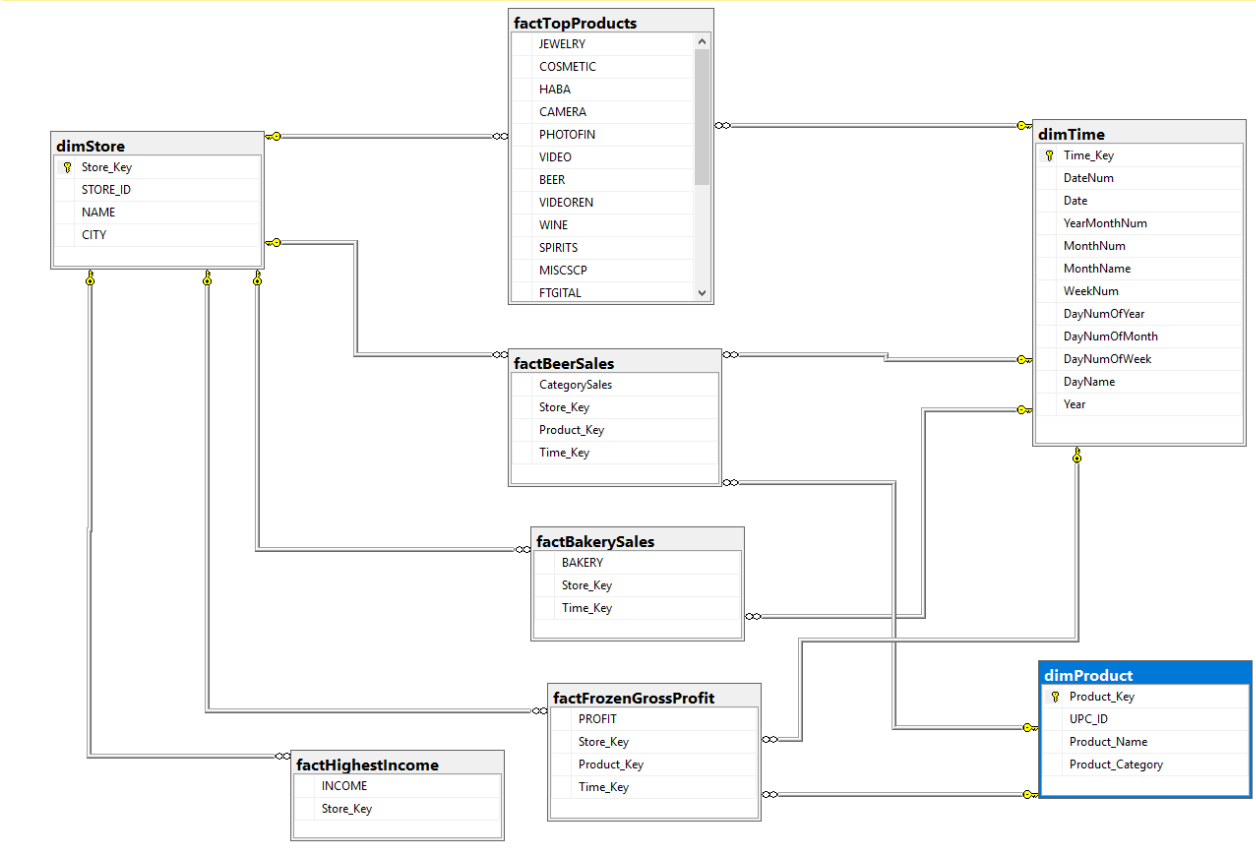
	!A	PHOTOFIN	VIDEO	BEER	VIDEOREN	WINE	SPIRITS	MISCSCP	FTGITAL	FTGCHIN	SSDELICP	PHARMACY	GM	BAKERY	Store_Key	Time_Key
1		0	0	13233.77	0	0	0	0	0	0	0	0	17805.96	4592.18	32	17899
2		1737.01	3428.86	1.99	0	0	0	-565.48	0	0	0	1733.52	1170128.78	588031.49	2	17900
3		866.27	77892.21	224102.55	33688.78	3487.51	0	0	0	0	0	0	940970.96	412600.01	3	17900
4		1735	239044	0	0	0	0	0	0	0	0	0	2265719	611968	4	17900
5		298.97	177685.13	490550.02	0	0	0	0	0	0	0	0	2243924.69	678144.43	5	17900
6		0	106161.76	585863.57	0	831.53	0	0	0	0	0	0	1751912.24	611669.879999999	6	17900
7		0	168927	838572.23	1206.68	0	0	0	0	0	0	4141	2509802.76	444557.1	7	17900
8		0	87134.89	660590.000000001	2547.17	1314.03	1587.41	0	0	0	0	0	1523473.34	523628.92	8	17900
9		0	11301.78	525649.280000001	0	0	0	0	0	0	0	0	2016136.31	854814.06	9	17900
10	3	3194.78	32871.3	0.98	24.95	0	0	-85	0	0	0	0	869228.85	119709.95	10	17900
11		977.84	119999.66	495488.88	10056.75	0	0	0	0	0	0	0	2457809.87	482317.57	11	17900
12		0	0	219337.54	0	0	0	0	0	0	0	0	561250.42	205526	12	17900
13		0	0	0	0	0	0	0	0	0	0	0	1152397.29	365890.86	13	17900
14		0	97368.74	615377.69	62662	1114	0	0	0	0	0	0	2169655.14	870266.79	14	17900
15		0	1642.12	549827.86	10.48	0	0	0	0	0	0	0	964492.0...	246636.47	15	17900
16		5.99	22478.64	0	0	0	0	0	0	0	0	0	782486.58	117294.8	16	17900
17		0	106745.97	503763.69	14470.41	0	0	0	0	0	0	0	2101606.27	555113.97	17	17900
18		110	35984.39	8	0	0	0	0	0	0	0	0	1343231.33	375401.64	18	17900
19		0	0	0	0	0	0	0	0	0	0	0	1221695.9	322667.48	19	17900



	(No column name)
1	971

Database Diagram of the Data warehouse

The data warehouse now looks like the following diagram. This explains the structure and relationships between dimension and fact tables in the data warehouse.



SQL Queries

1. CREATE TABLE [dbo].[dimTime] (
   [Date_Key] int IDENTITY(1,1) Primary Key,
   [Year] int,
   [DayName] nvarchar,
   [WeekNum] int,
   [Month] nvarchar
   );

   ALTER TABLE dimTime ADD Year AS ((YearMonthNum / 100));

2. CREATE TABLE [dbo].[dimProduct] (
   [Product_Key] bigint IDENTITY(1,1) Primary Key,
   [Product_Number] bigint,
   [Product_Name] nvarchar,
   [Category] nvarchar
   );

3. CREATE TABLE [dbo].[dimStore] (
   [Store_Key] int IDENTITY(1,1) Primary Key,
   [Store_Number] int,
   [Store_Name] nvarchar,
   [City] nvarchar
)

4. CREATE TABLE [factFrozenGrossProfit] (
   [Profit] float,
   [Store_ID] int,
   [Product_Key] bigint,
   [Time_Key] int,
   FOREIGN KEY ([Store_Key]) REFERENCES dbo.dimStore ([Store_Key]),
   FOREIGN KEY ([Product_Key]) REFERENCES dbo.dimProduct ([Product_Key]),
   FOREIGN KEY ([Time_Key]) REFERENCES dbo.dimTime ([Time_Key]),
   );

5. CREATE TABLE [factBeerSales](
   [Sales] float,
   [Store_ID] int,
   [Product_Key] bigint,
   [Time_Key] int,
   FOREIGN KEY ([Store_Key]) REFERENCES dbo.dimStore ([Store_Key]),
   FOREIGN KEY ([Product_Key]) REFERENCES dbo.dimProduct ([Product_Key]),
   FOREIGN KEY ([Time_Key]) REFERENCES dbo.dimTime ([Time_Key]),
   );

6. CREATE TABLE [factBakerySales](
   [Bakery] float,
   [Store_Key] int,
   [Time_Key] int,
   FOREIGN KEY ([Store_Key]) REFERENCES dbo.dimStore ([Store_Key]),
   FOREIGN KEY ([Time_Key]) REFERENCES dbo.dimTime ([Time_Key]),
   );

7. CREATE TABLE [factHighestIncome](
   [Income] float,
   [Store_Key] int,
   FOREIGN KEY ([Store_Key]) REFERENCES dbo.dimStore ([Store_Key]),
   );

BI Reporting

Reporting Plan

Business Intelligence reporting (BI Reporting) is the process of providing information in the form of reports to end-users, companies, organizations or Decision Support (DSS) Applications. The results of BI reporting are generally in the form of actionable results that help the organization in making tactical and strategic decisions. This is a technology driven process of presenting results from which users can understand the current standing of their business or organization and identify areas of improvement.

BI encompasses a variety of tools, methodologies that enable businesses to analyze data collected from disparate sources, run queries against this data to create reports, dashboards and analyze the reports from a data visualization tool.

Benefits of Reporting

Reports make understanding the results of complex analysis easy for the end users. Reports enable collecting and presenting data (including historical data) that is ready to be analyzed and tracked over time. Reports empower the end-users or decision makers with the knowledge that will enable them to gain expertise in their business domain.

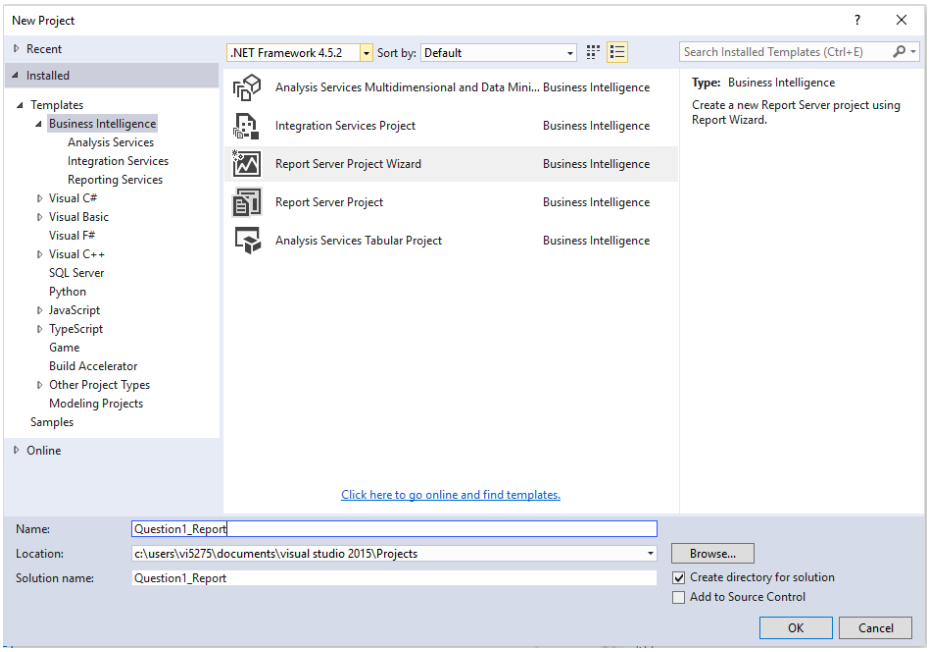
For this project, we use SSRS, Report Builder 3.0, SSAS and SSRS over SSAS to build reports for the chosen business questions.

The following is the list of tools we made use of in order to build reports for the business questions.

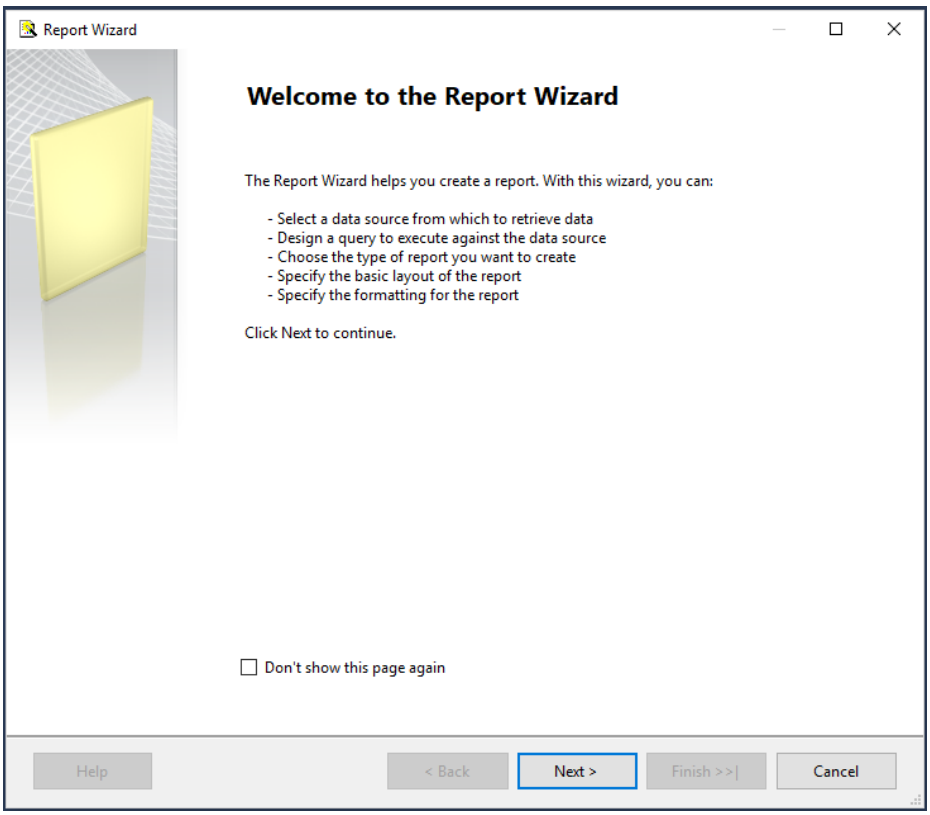
Reporting Tool	Question
SSRS	1
Report Builder	2
SSAS	3
SSRS	4
SSRS over SSAS	5

Q1. Determine average gross profit margin of frozen products across stores and verify which stores are below average and plot the same?

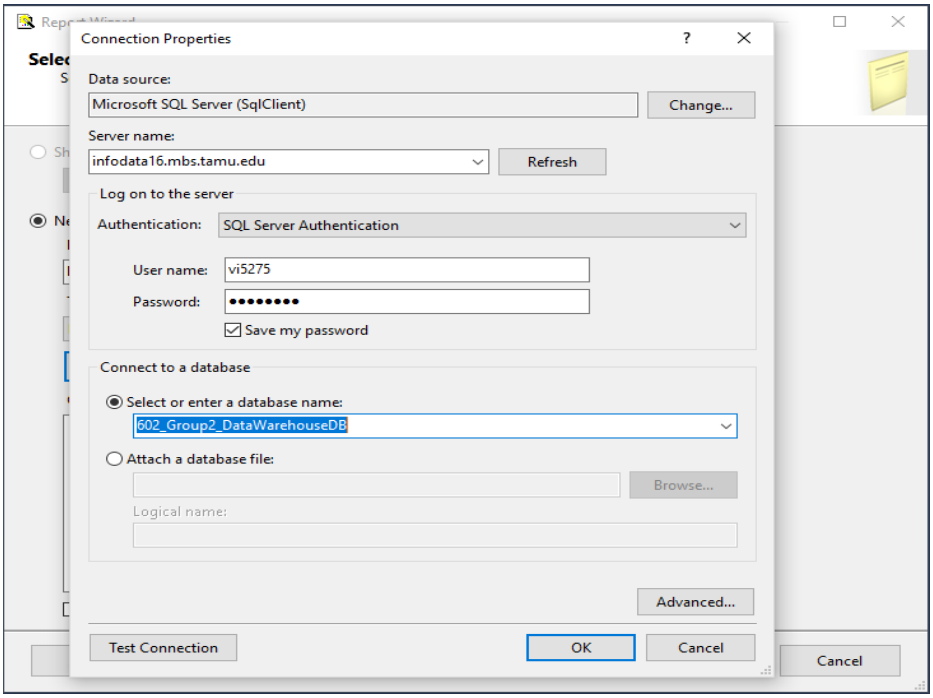
i. Create a new SSRS project using - report server project wizard



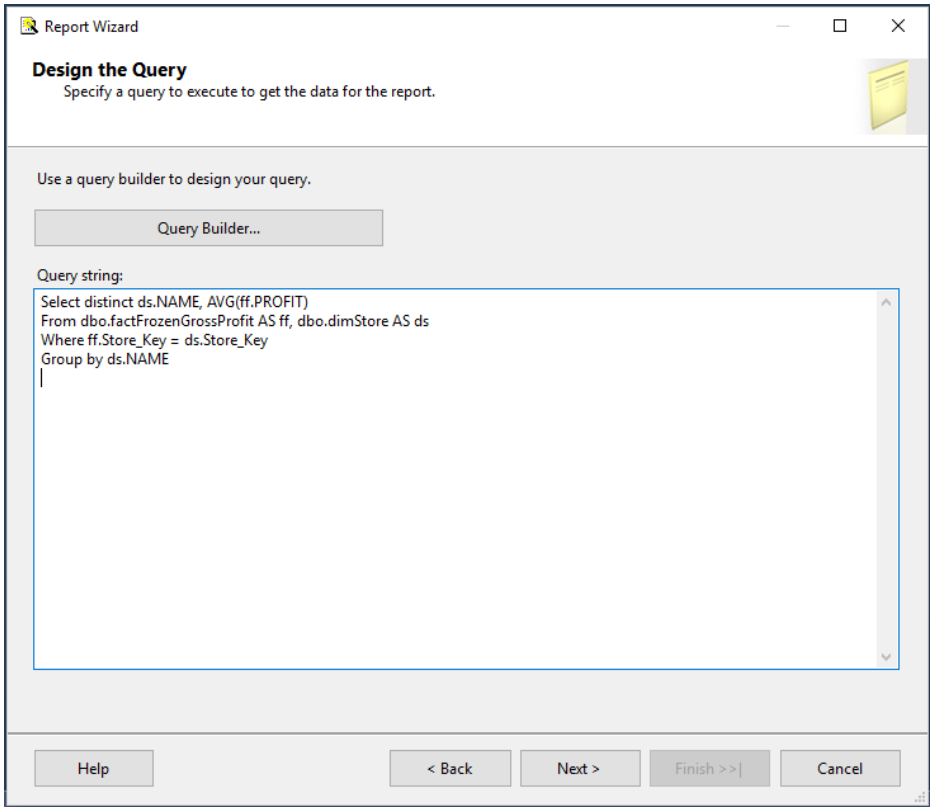
ii. Click Next on the report wizard



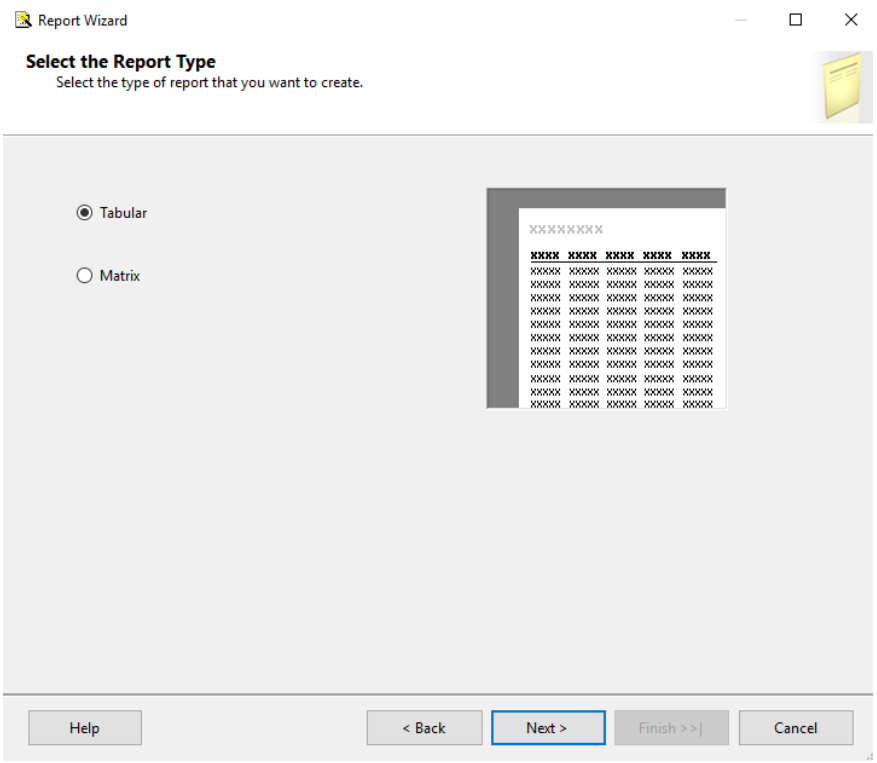
iii. Enter the server name as infodata16.mbs.tamu.edu, and enter the credentials followed by the database name.



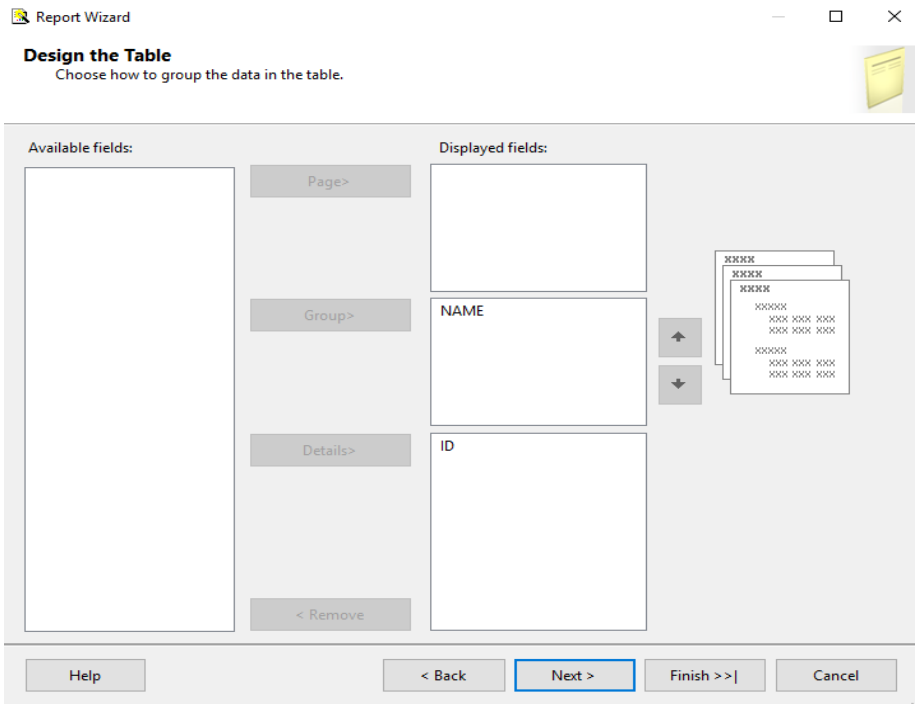
iv. Create the query using design query



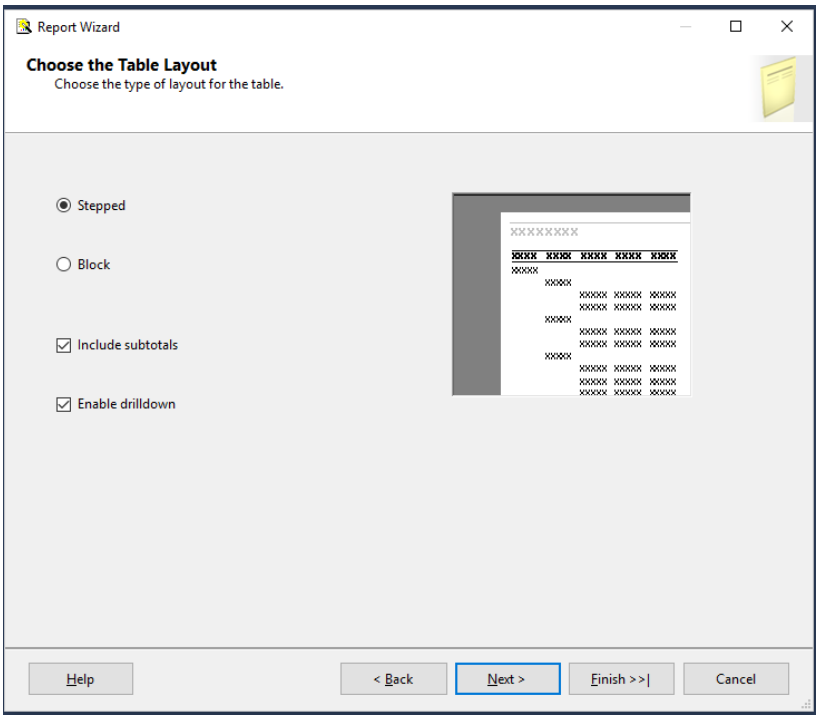
v.



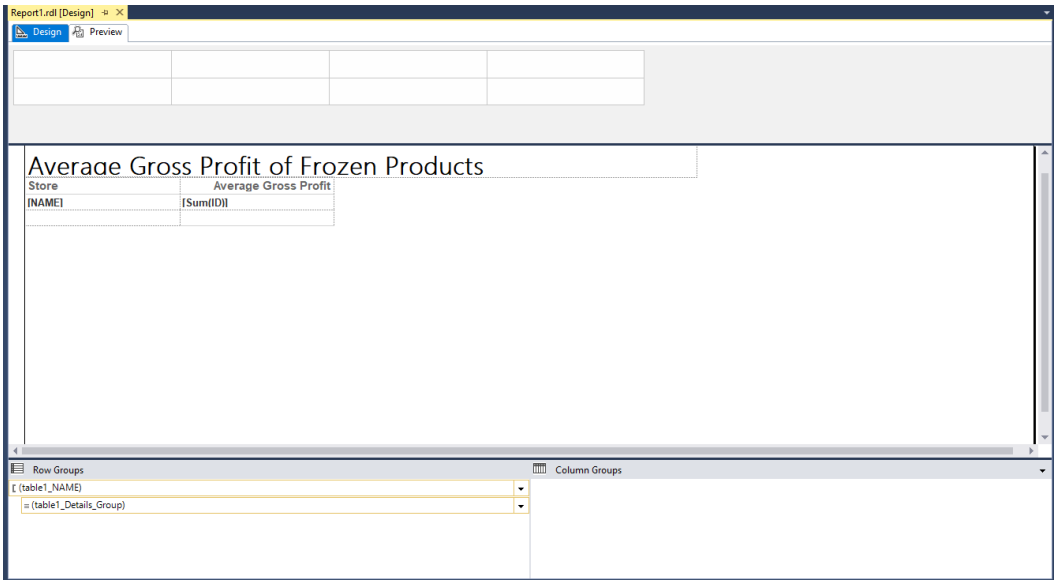
vi. Group by store name and aggregate the profit by average



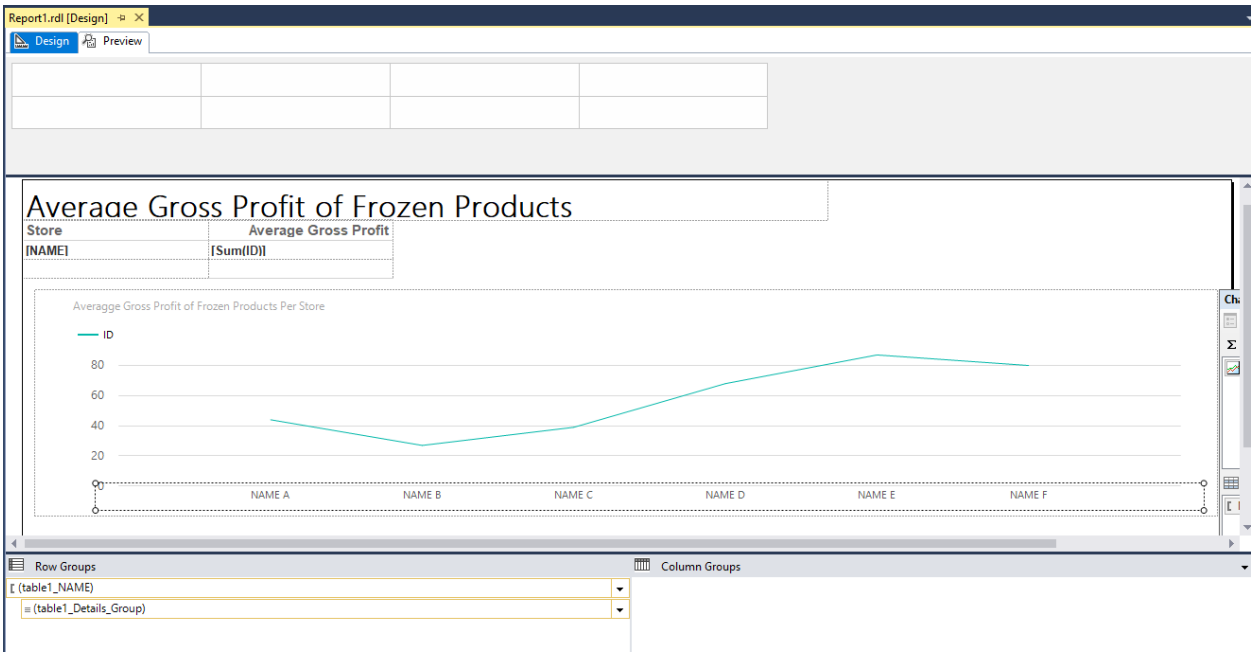
vii.


Report Wizard
Choose the Table Layout
Choose the type of layout for the table.
Stepped
Block
Include subtotals
Enable drilldown
Help
< Back
Next >
Finish >>|
Cancel

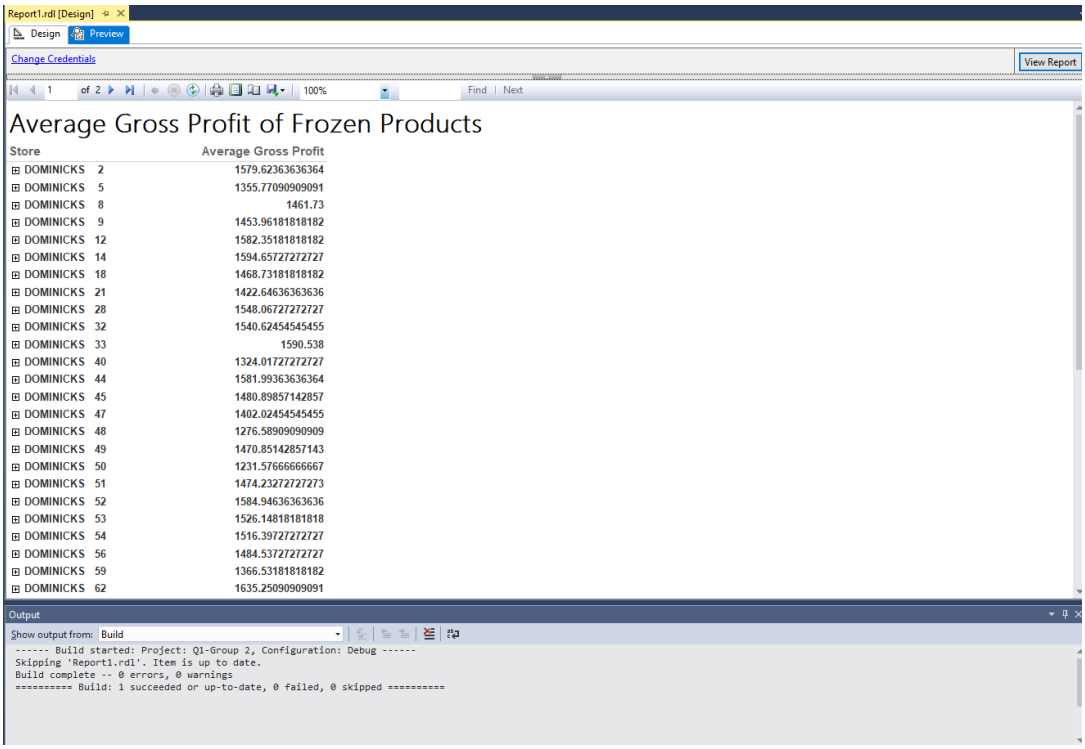
viii.


Report1.rdl [Design]
Design
Preview
Average Gross Profit of Frozen Products
Store
Average Gross Profit
[NAME]
[Sum(ID)]
Row Groups
Column Groups
[(table1_NAME)
= (table1_Details_Group)

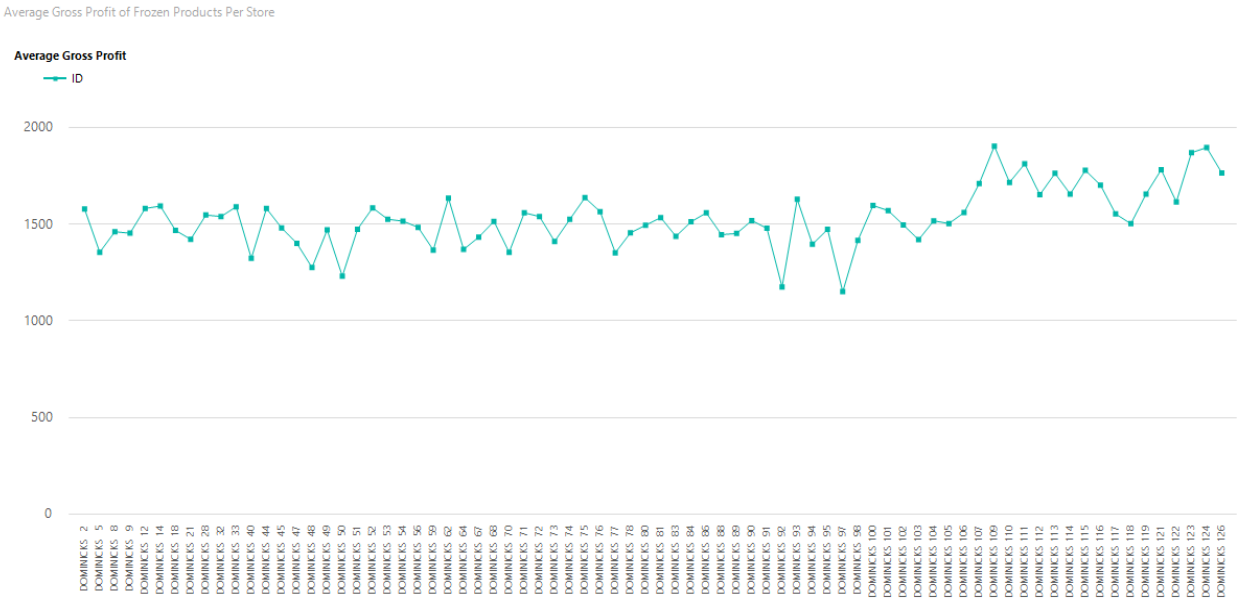
ix. The report is created as follows:



x.



xi. Final Output, when we run this report:

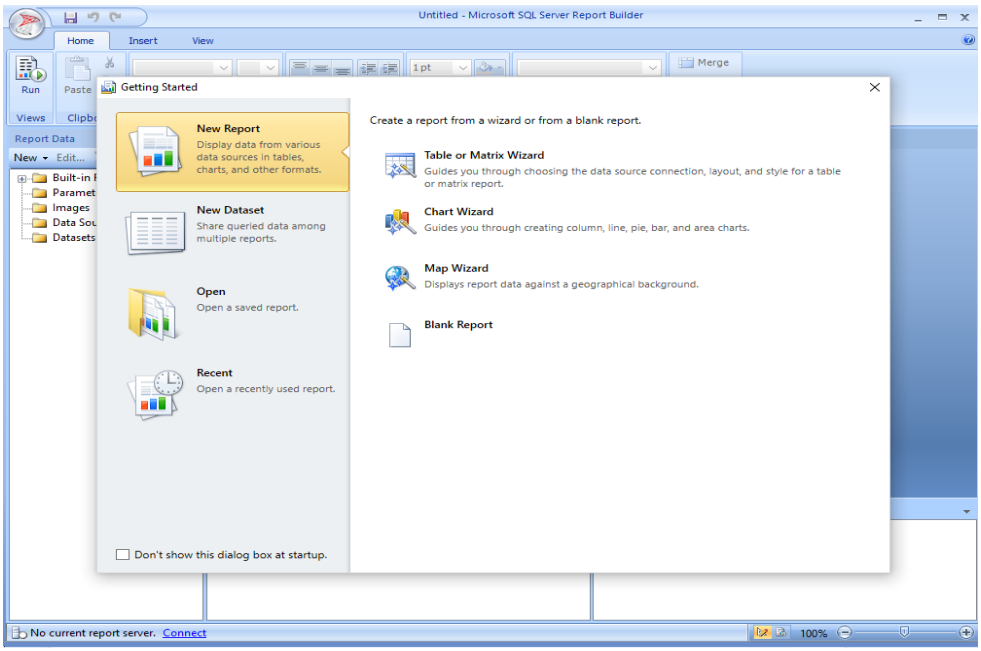


Conclusion:

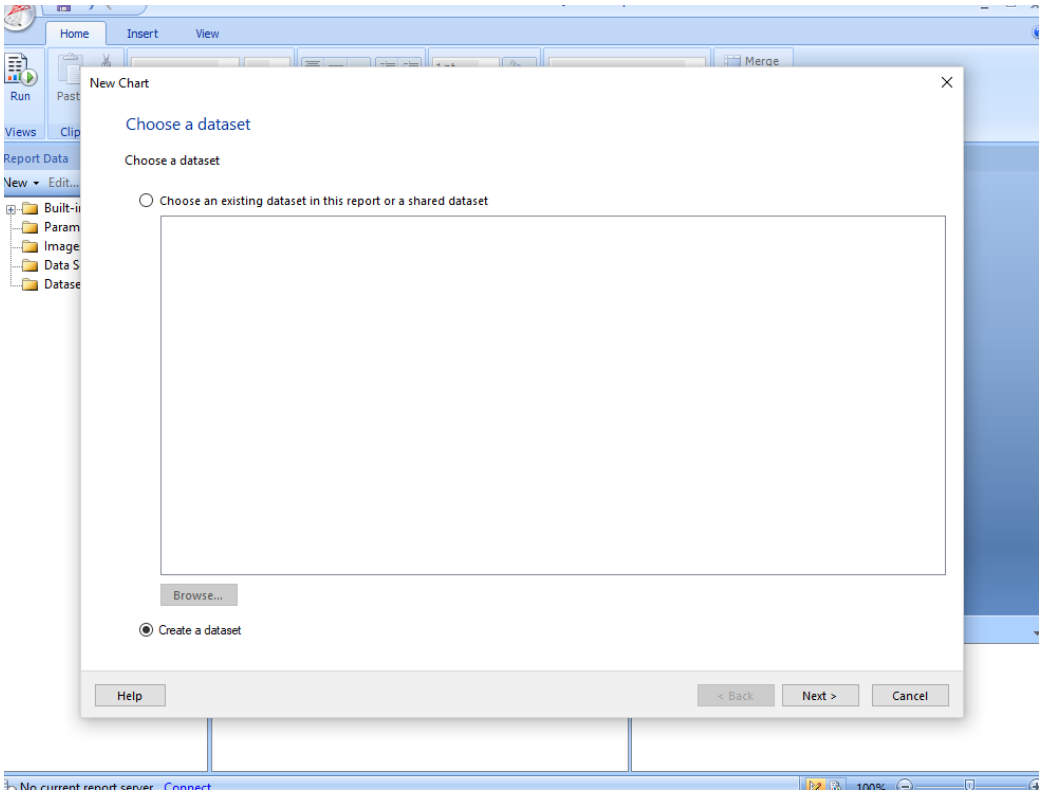
Through this report we can identify as to which stores made a good return on profit by selling frozen products. In the US frozen products have become a go to option for many people. Through this analysis DFF can target those stores where the gross profit is less and aim at trying new initiatives to increase the profit. The stores where the profit is more can be looked at as an example for the stores with lesser profit to follow and improve their sales.

Q2. What is the average sales for beer in the last 3 years?

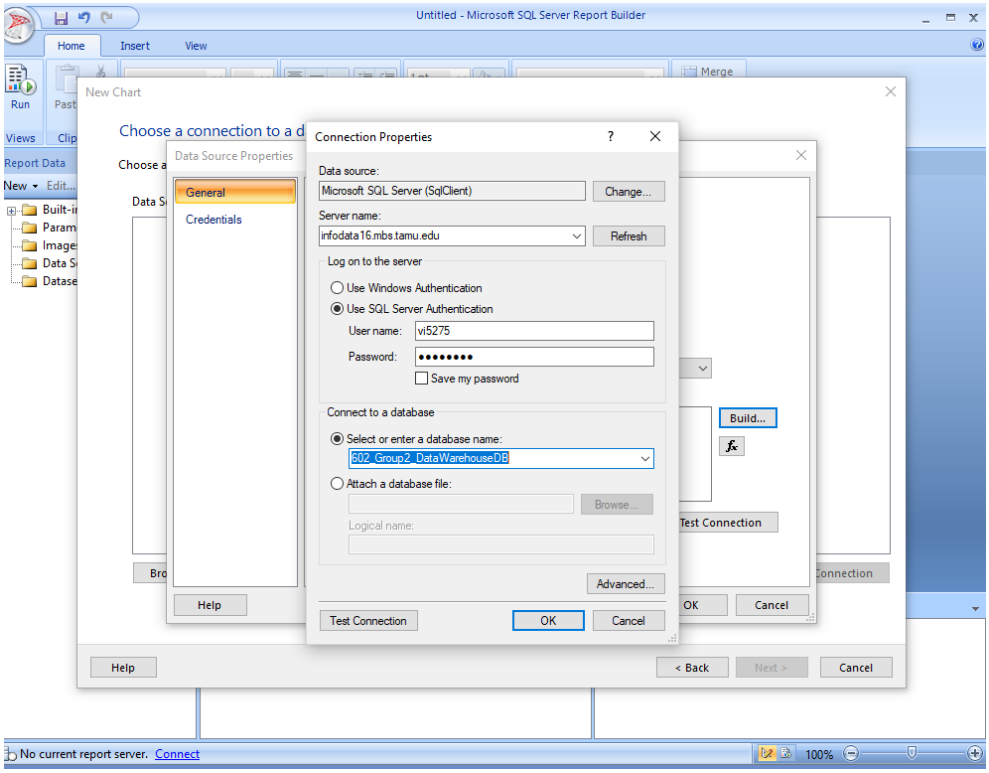
i. Create a new project using report builder - choose chart wizard



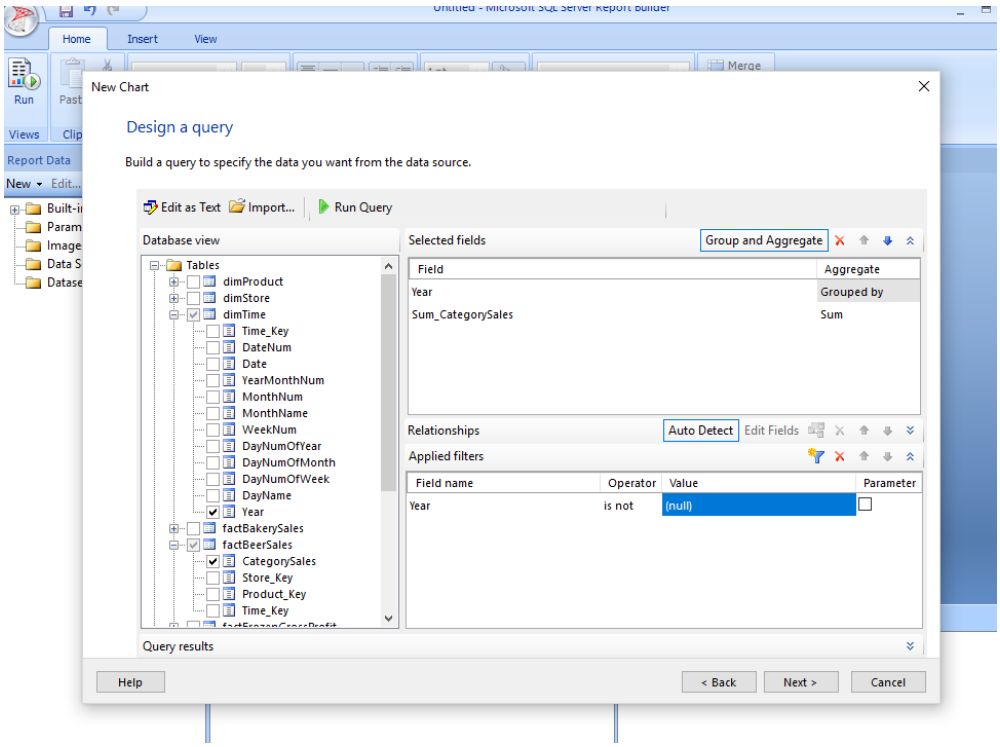
ii. Create a new database connection



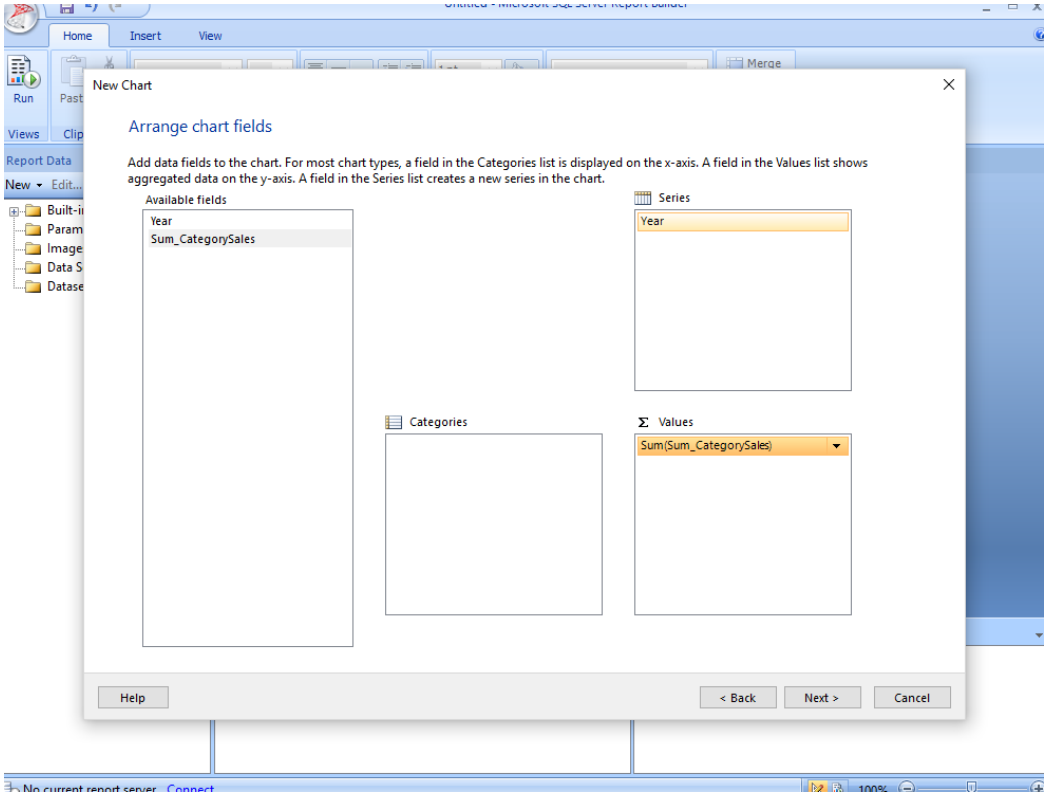
iii. Enter the server name - imfodata16.mbs.tamu.edu, enter credentials and select the data warehouse database



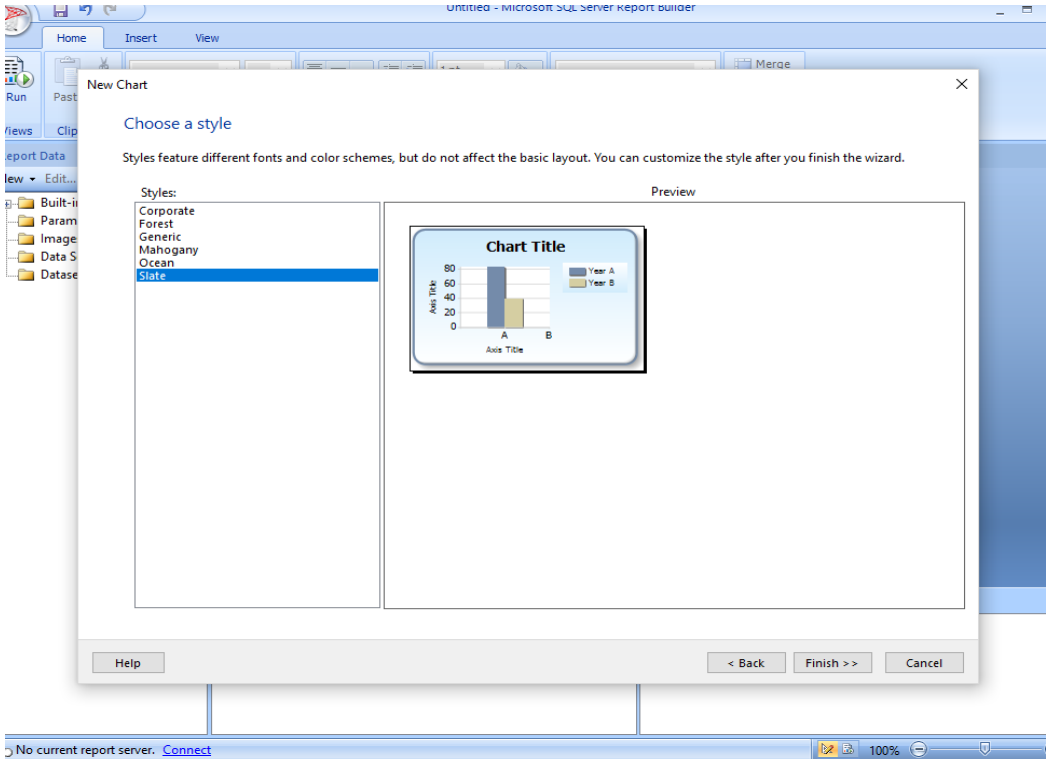
iv. Create the query using query designer



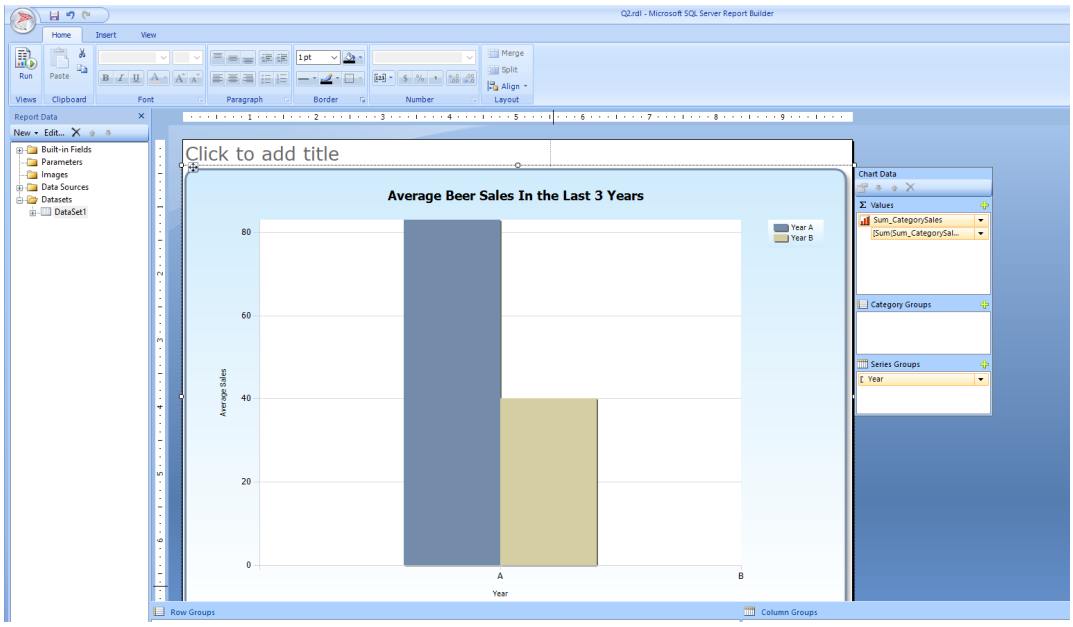
v. Aggregate the sales of beer by the year

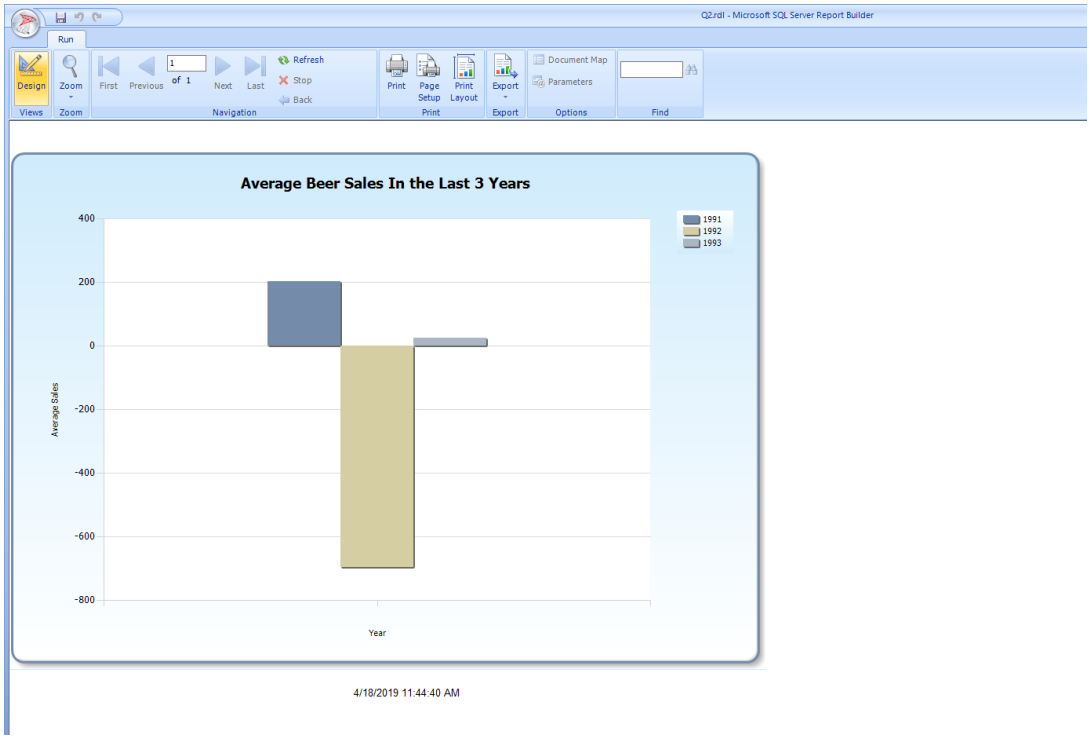


vi. Choose the chart type

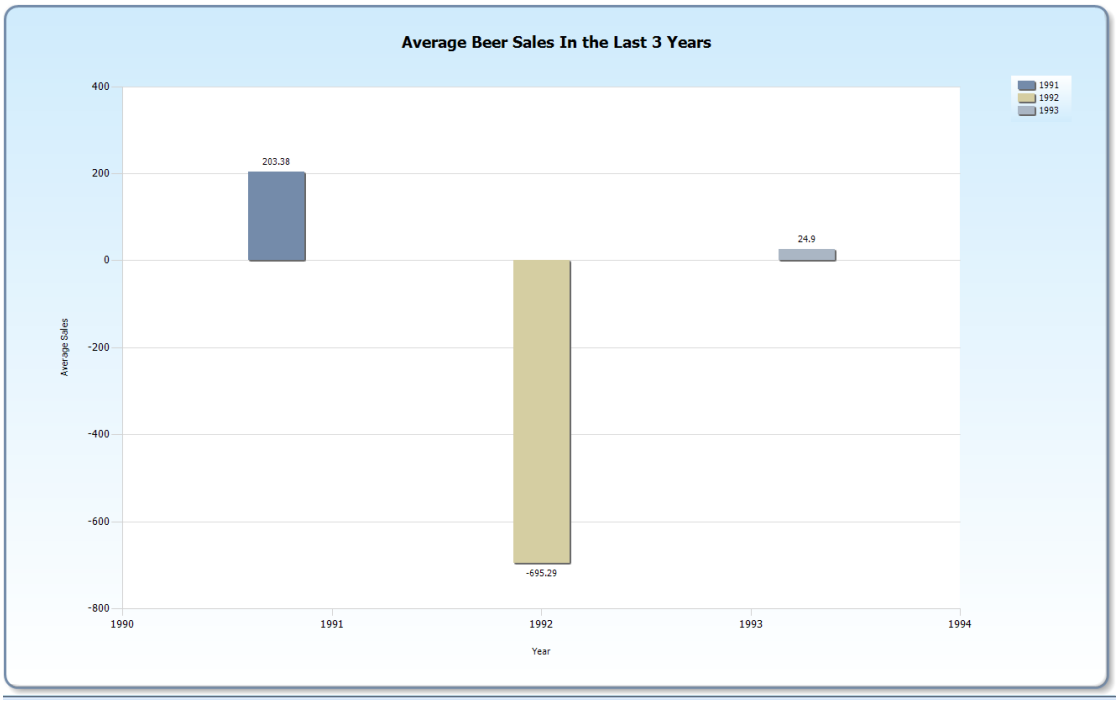


vii. The report is created as follows:





viii. This is the final report output:



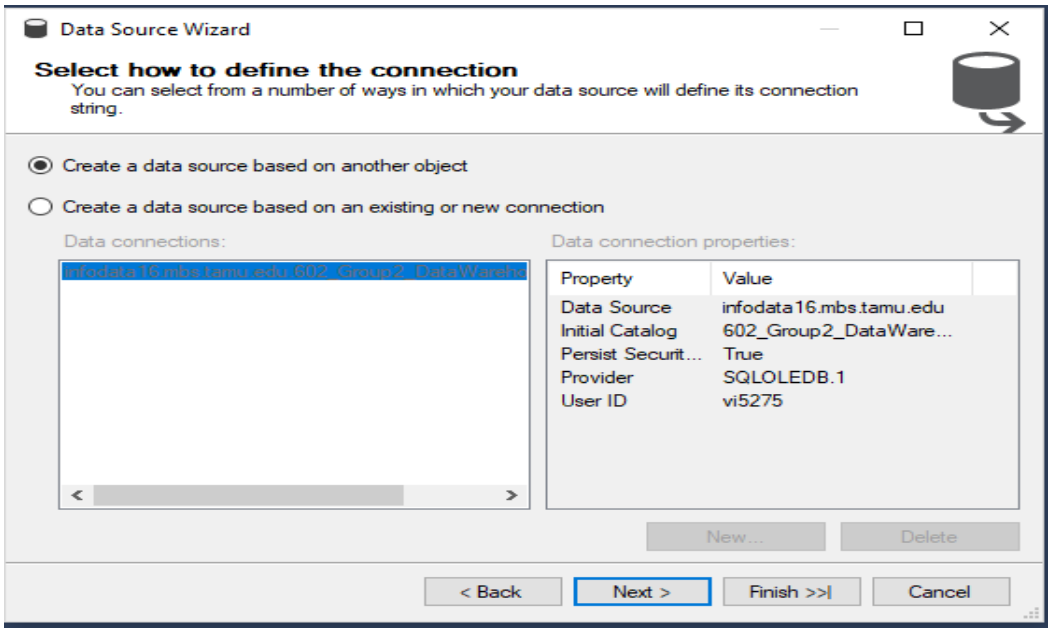
Conclusion:

Beer consumption in the US has been described as one of the largest across the world. This report aims to show the average beer sales across the last three years. We can see a huge drop in the sale of beer during 1992 which was preceded by a relatively high average from 1990 to 1991. This is followed by a rise again in the average sale of beer. This could mean that during 1992 DFF concentrated on the sales of other products or perhaps DFF improved their marketing for beer products due to which there was a rise again in the sale of beer.

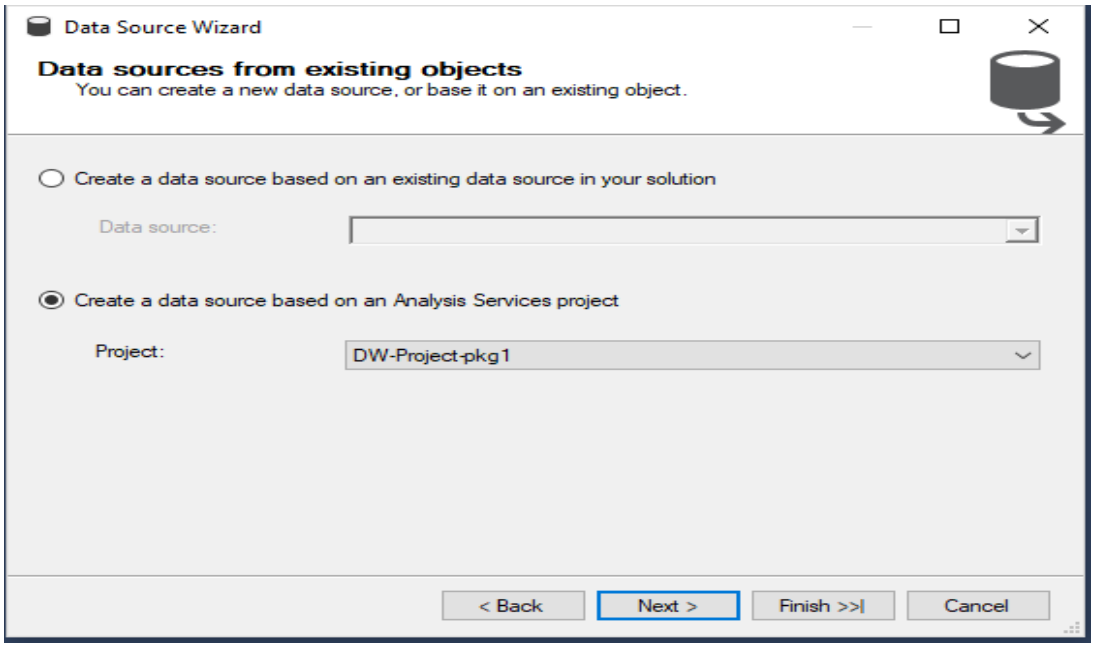
Q3. What is the growth of Bakery from the year 1990 to 1996?

Solution: we adopted SSAS to answer this business question. SSAS template of MS Visual Studio helps us connect to the right data source (Data Warehouse) and deploy the solution in that data source.

i. We start by connecting to the right data source based on the DW created in the previous tasks

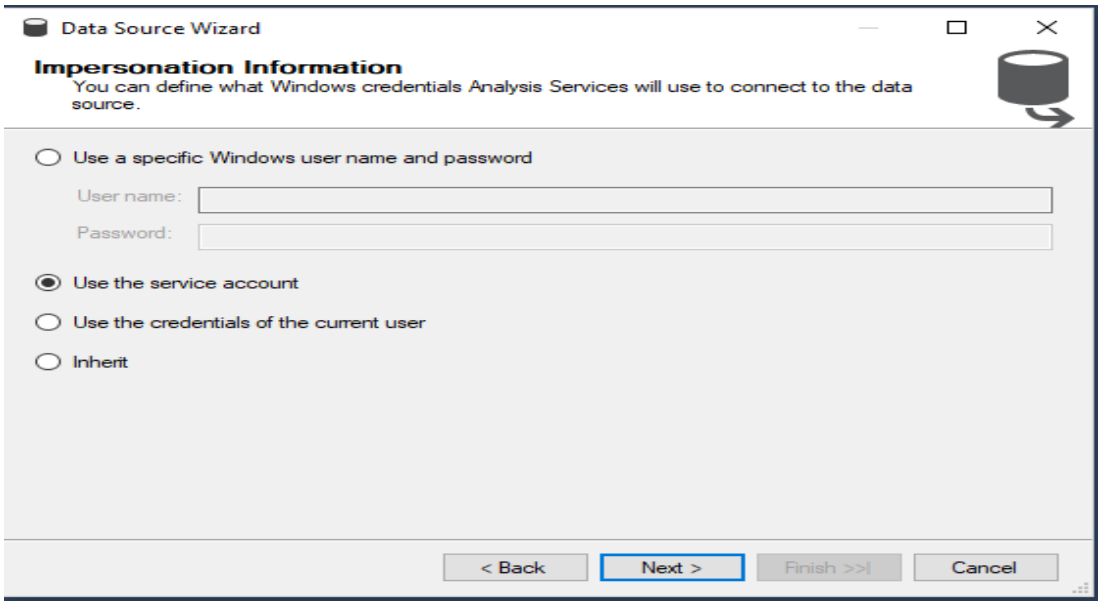


ii. We then create a new data source for analysis services project based on our data

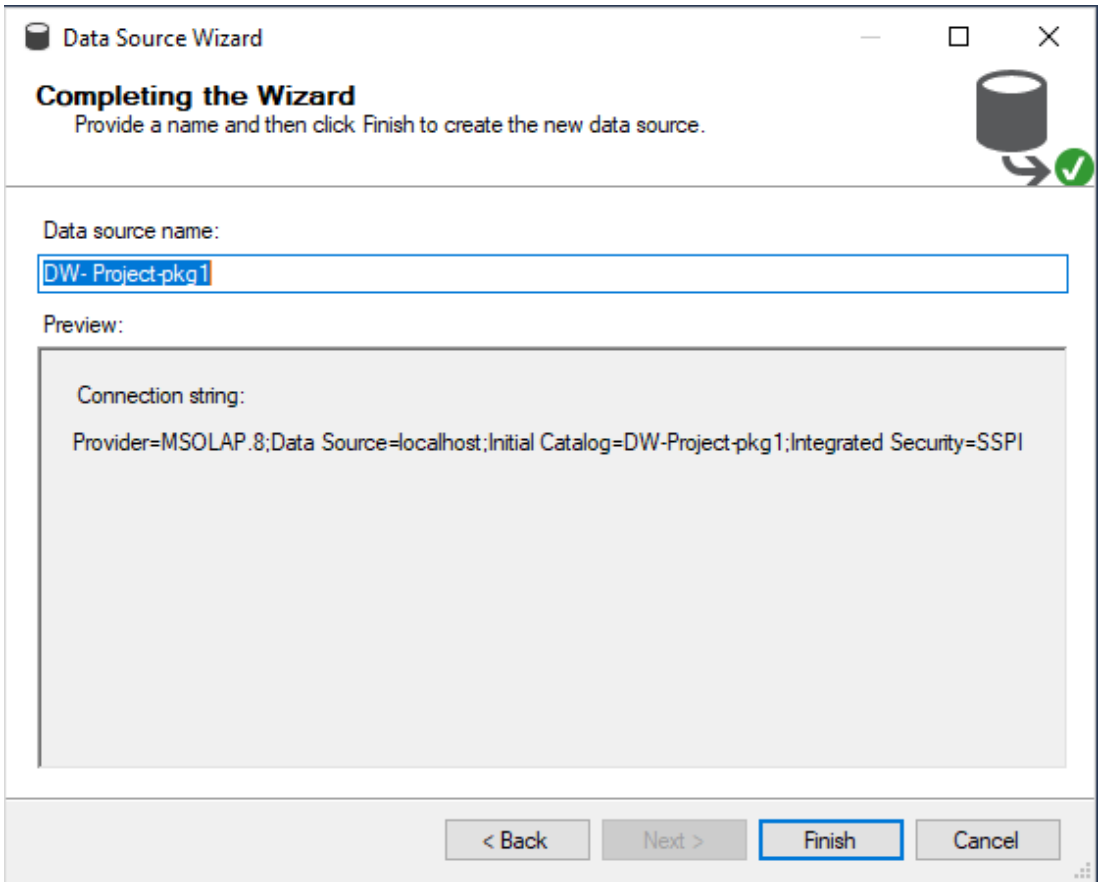


warehouse

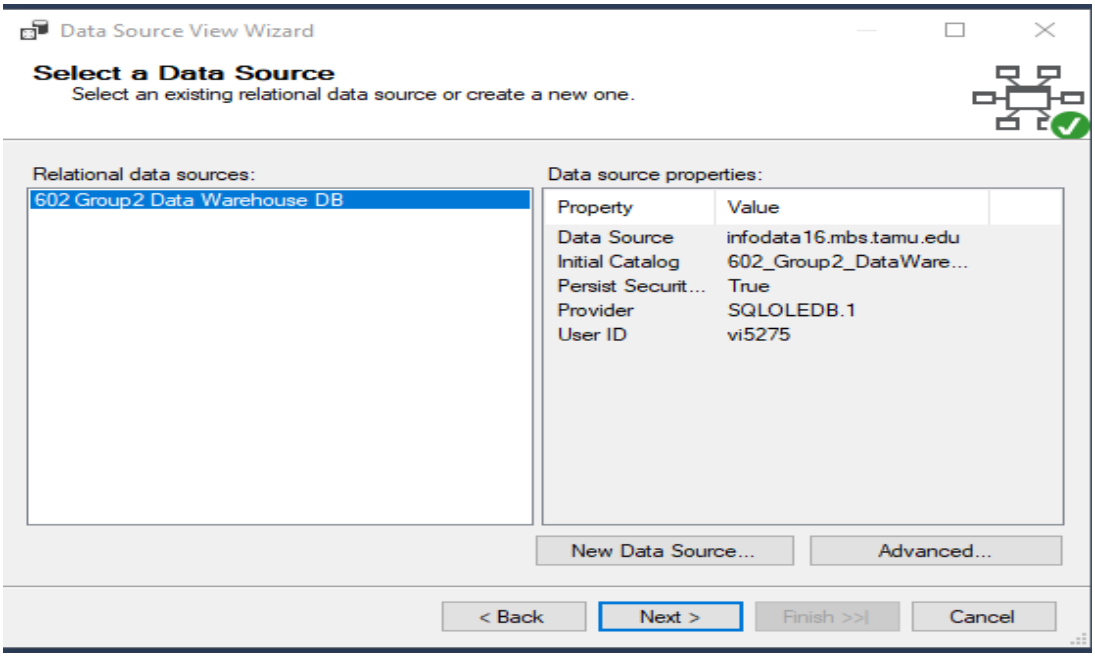
iii. Default Impersonation Information is selected



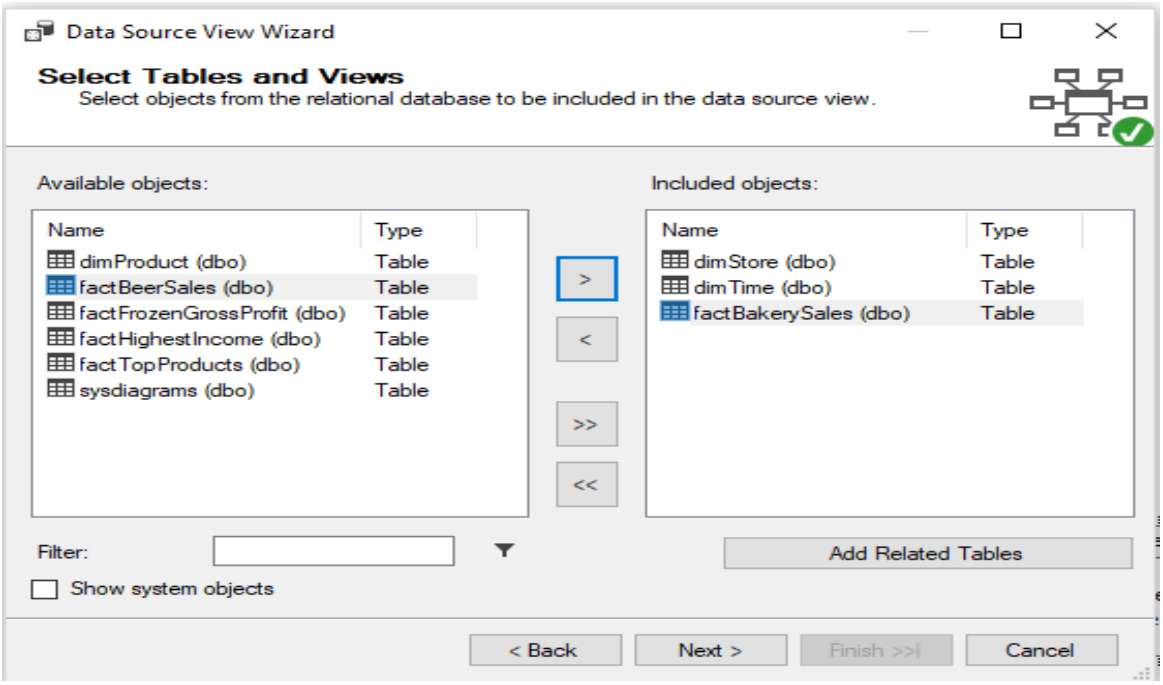
iv. Creating the project is completed by naming it.

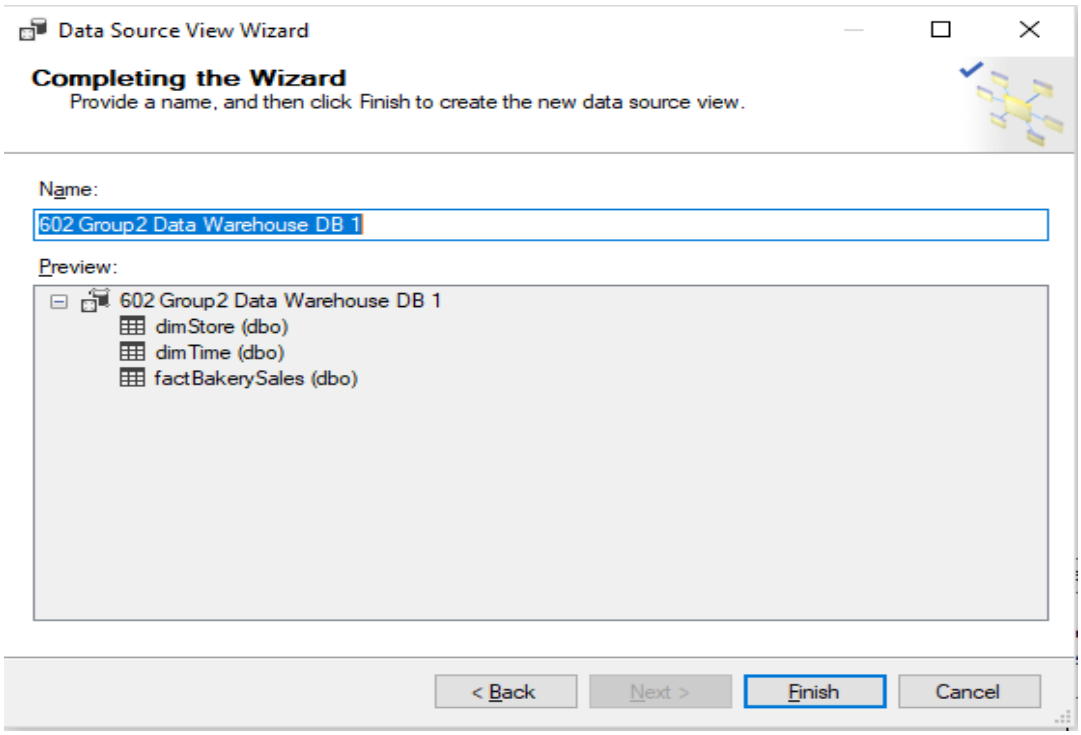


v. Now the new data source view is created to have only necessary objects required by the business questions by selecting the previously created data source connection for the project.

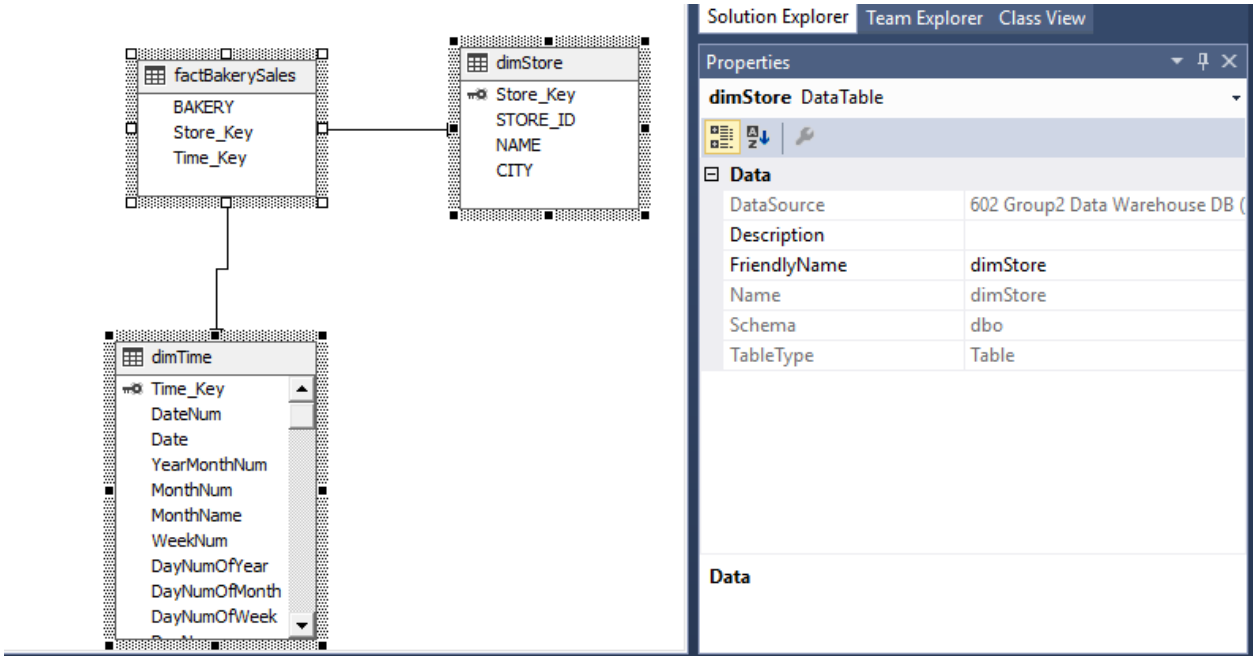


vi. Based on the business question-3, following objects are chosen for the view.

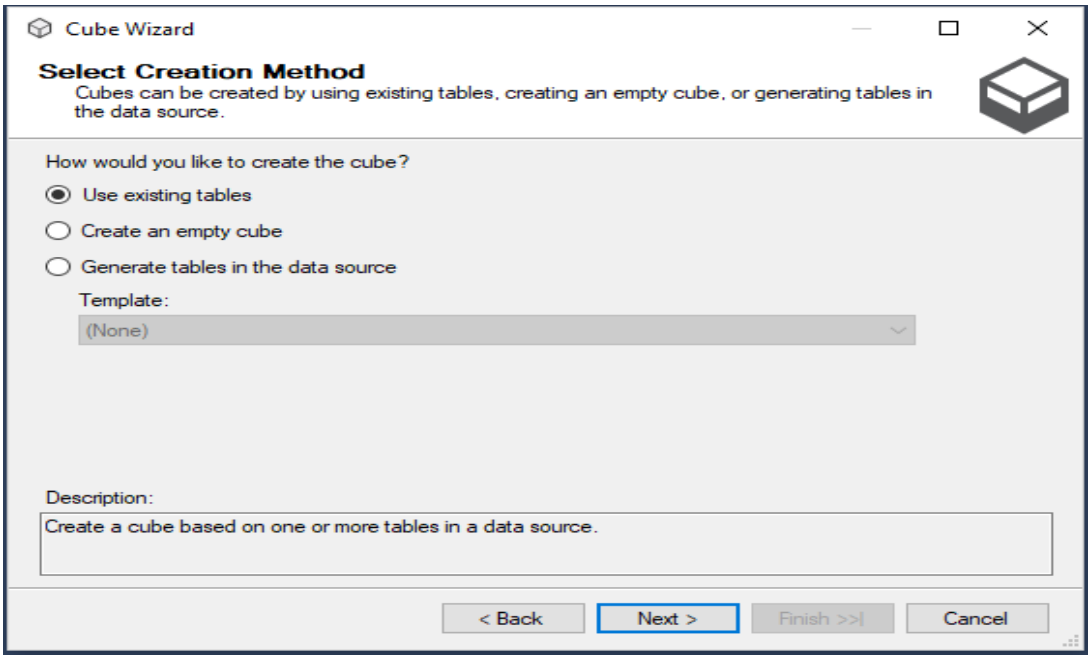




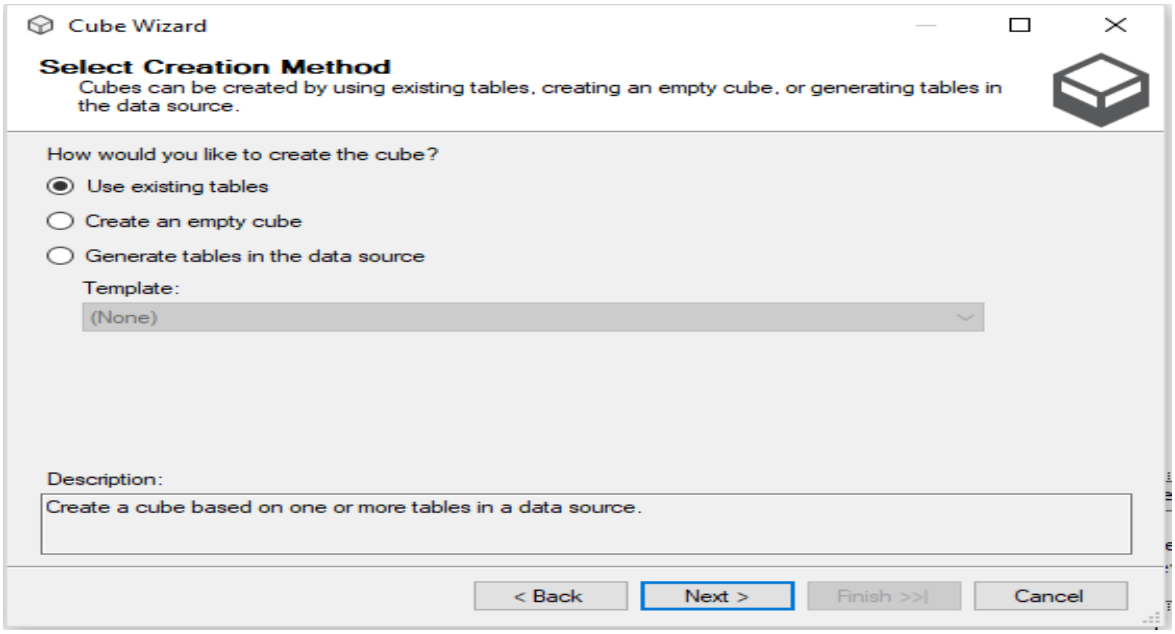
vii. Post creation of the data source view, the DB structure can be verified. This also allows us to manipulate columns in the objects.



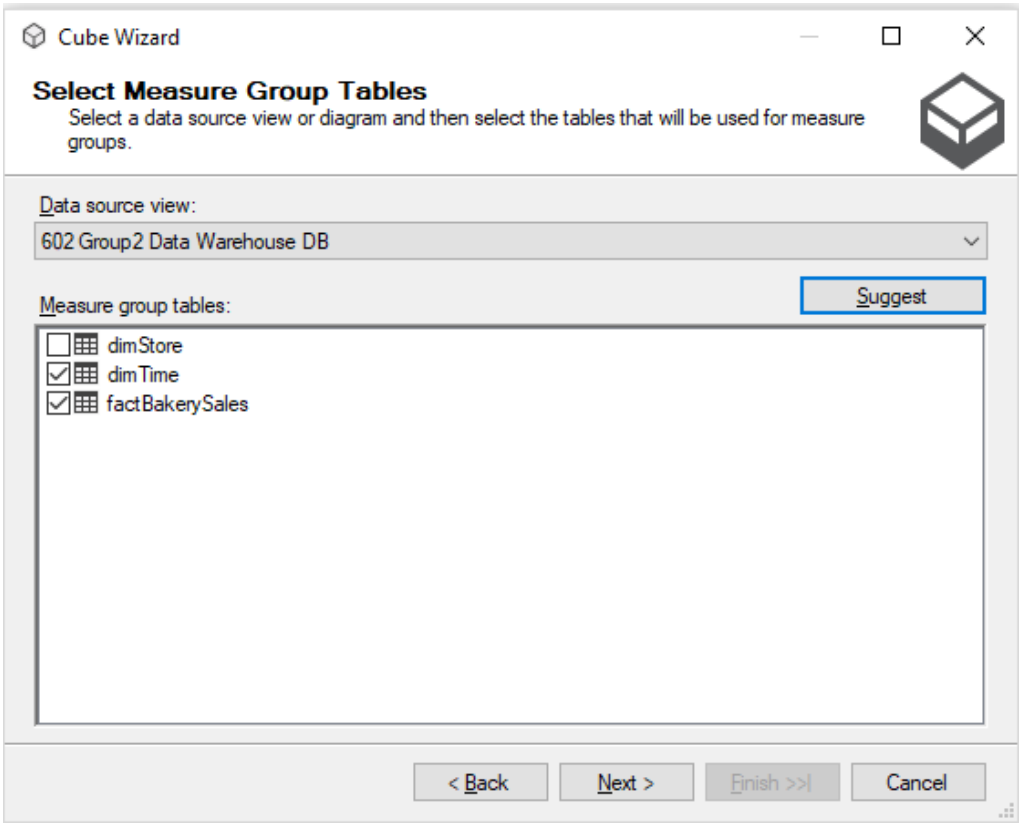
viii. Creating the cube:
Once the data source is created Cube Wizard is accessed under that data source to create a new cube using the existing objects in the data warehouse.



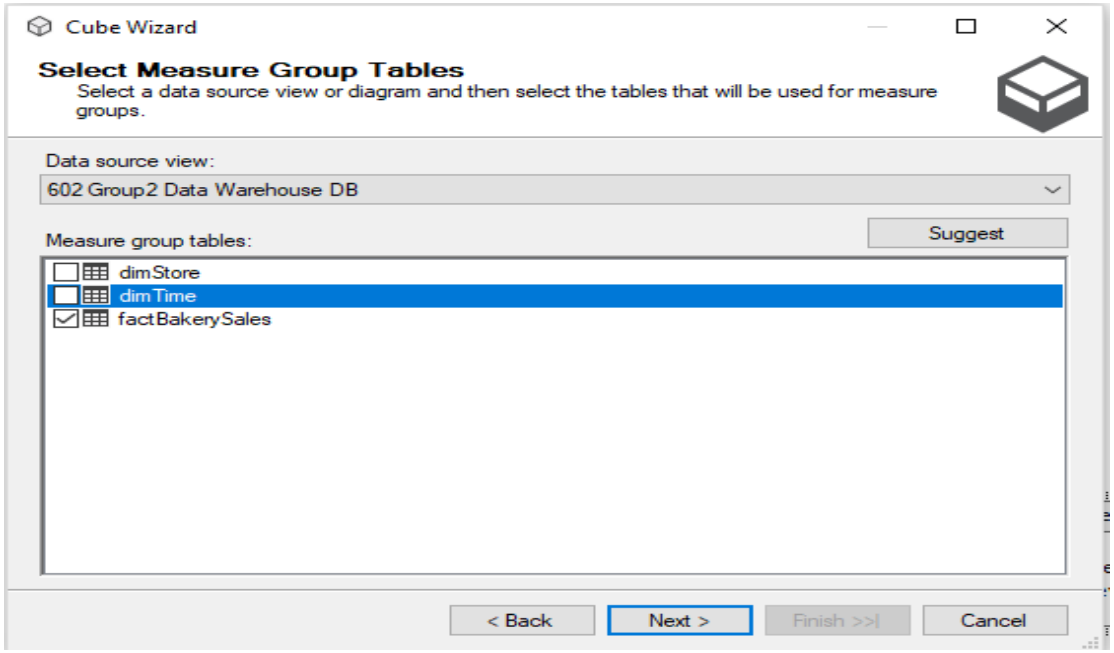
ix. By selecting the existing objects in our data source, we can proceed with creating the cube



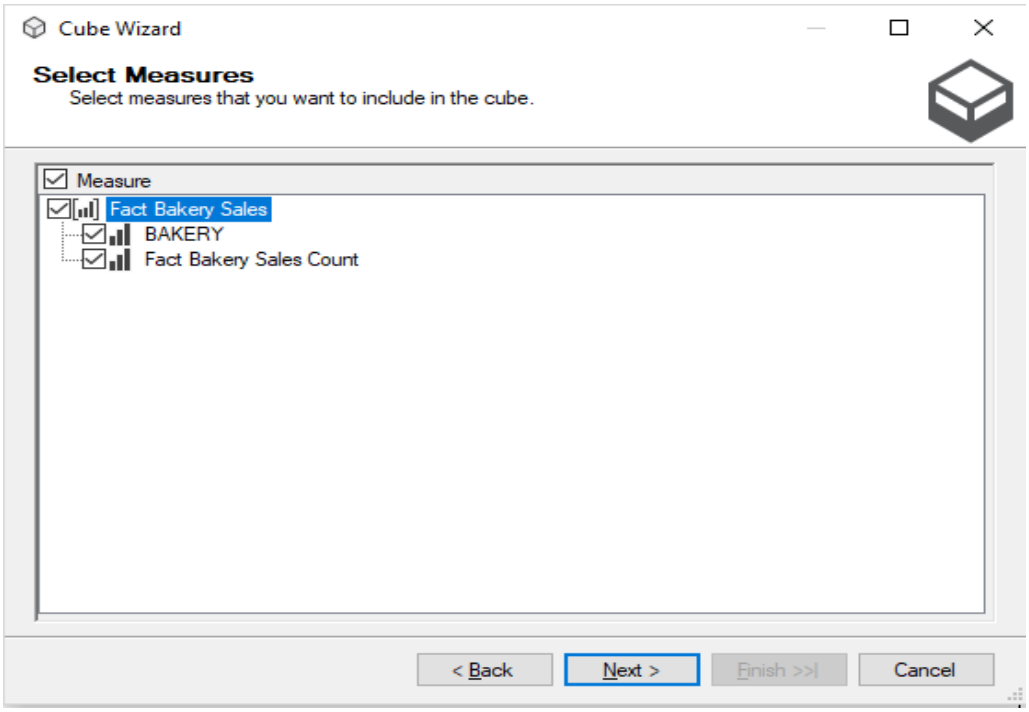
x. The beauty of the wizard is that, it provides us with suggestions for our cube



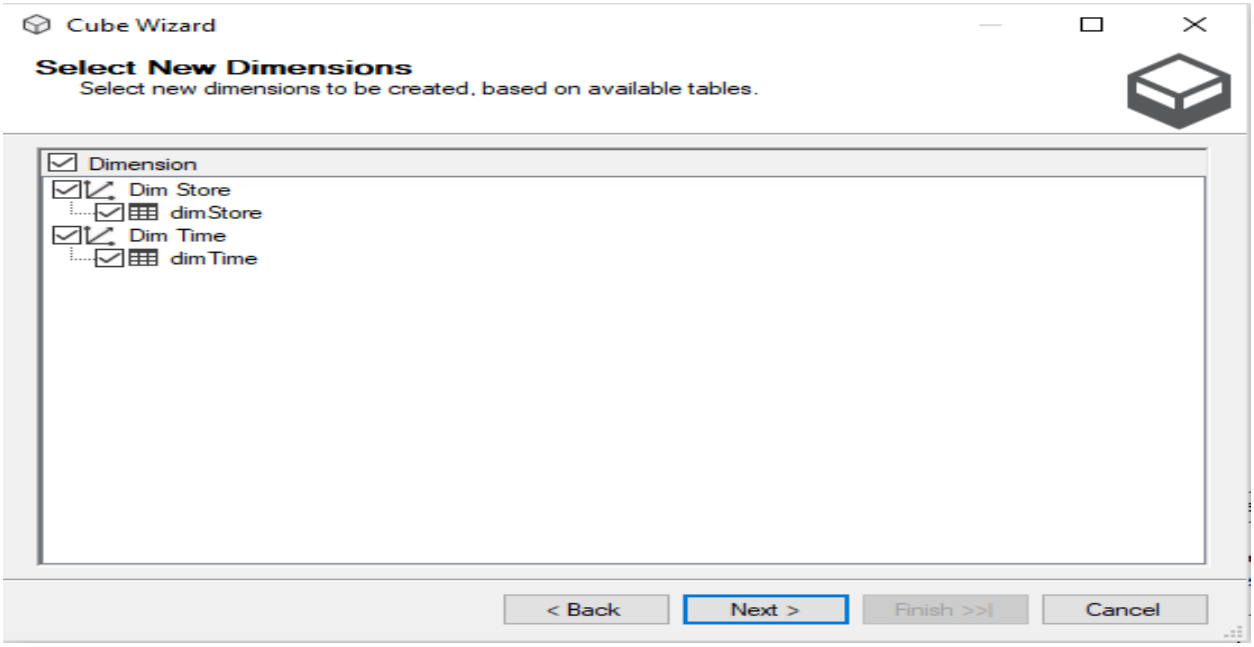
xi. Then we ensure the right measures are chosen for the cube by editing the checklist



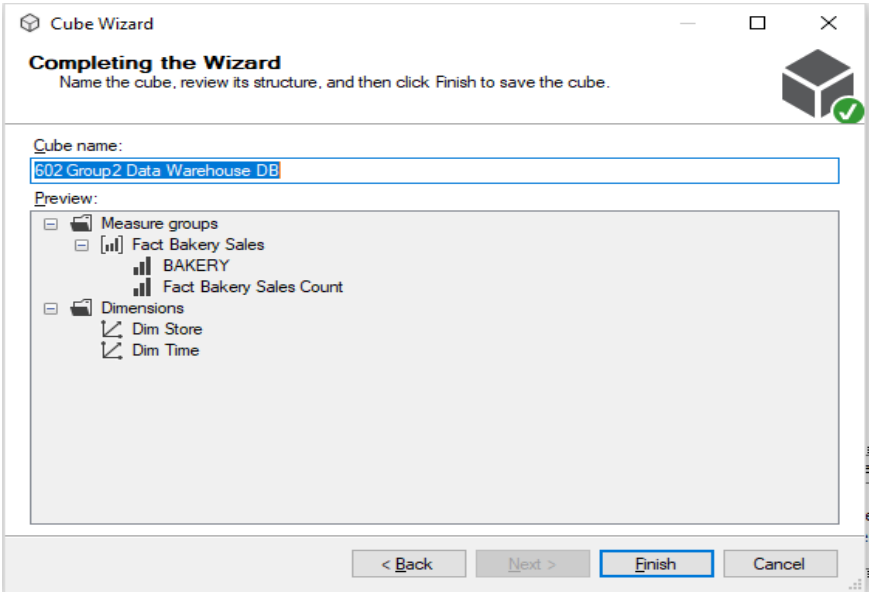
xii. Accordingly the columns are chosen for the measure (fact table)



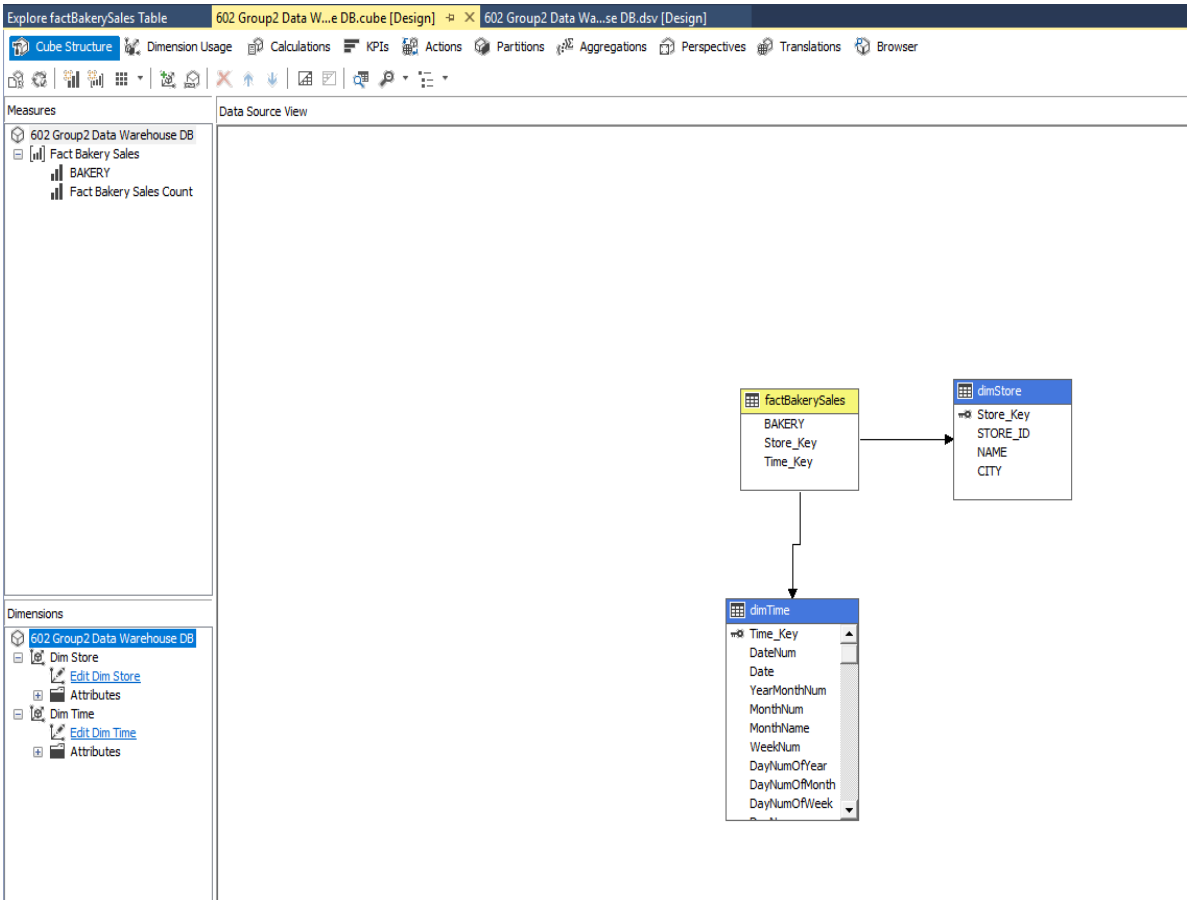
xiii. Next, respective dimension tables are chosen for the cube



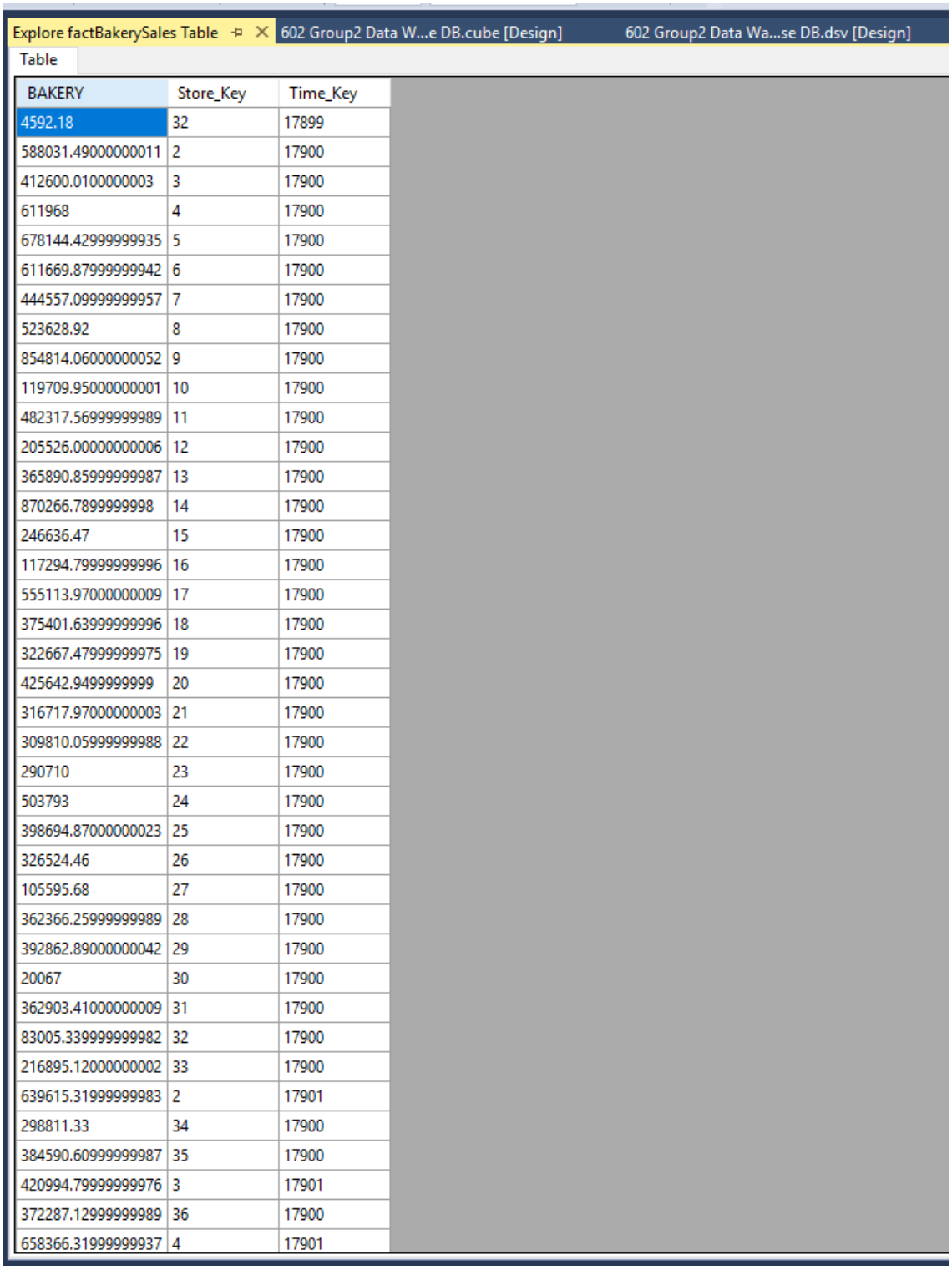
xiv. Cube creation is completed by naming



xv. Once complete, the final cube structure can be verified under Cube Structure tab to ensure the Measures and Dimensions selected, their relationships and the corresponding columns.



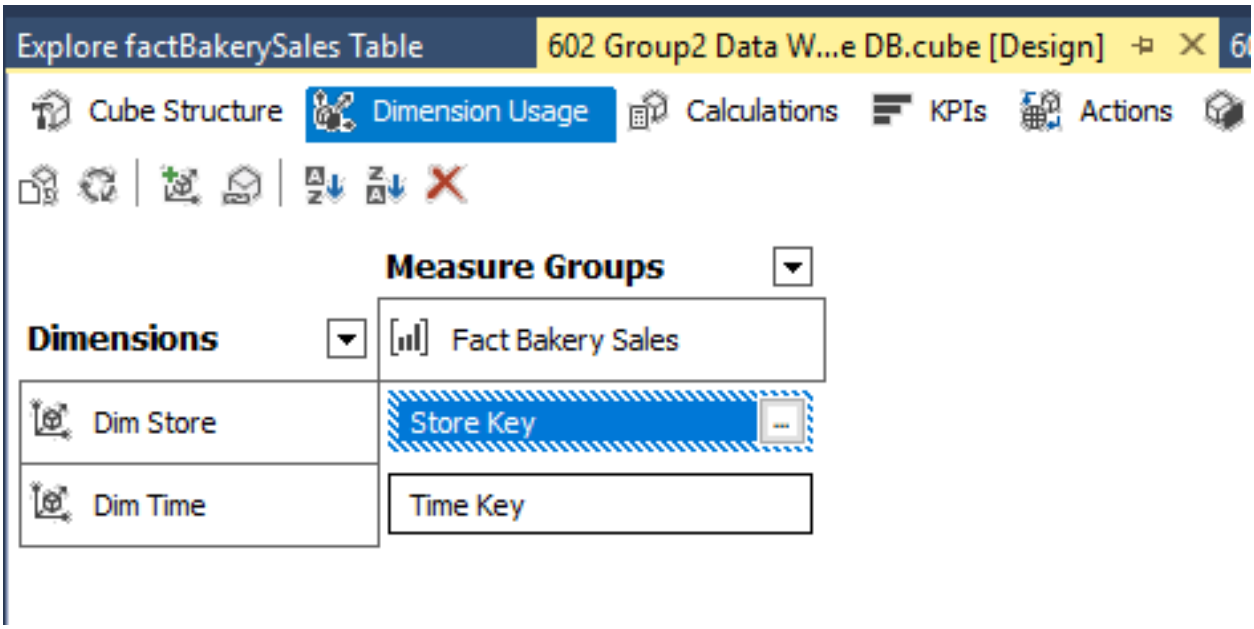
xvi. Explore fact table: Helps us view the data in the fact table residing in the data warehouse. Dimension Usage tab displays which dimensions and measures are being used for the cube and their corresponding keys.



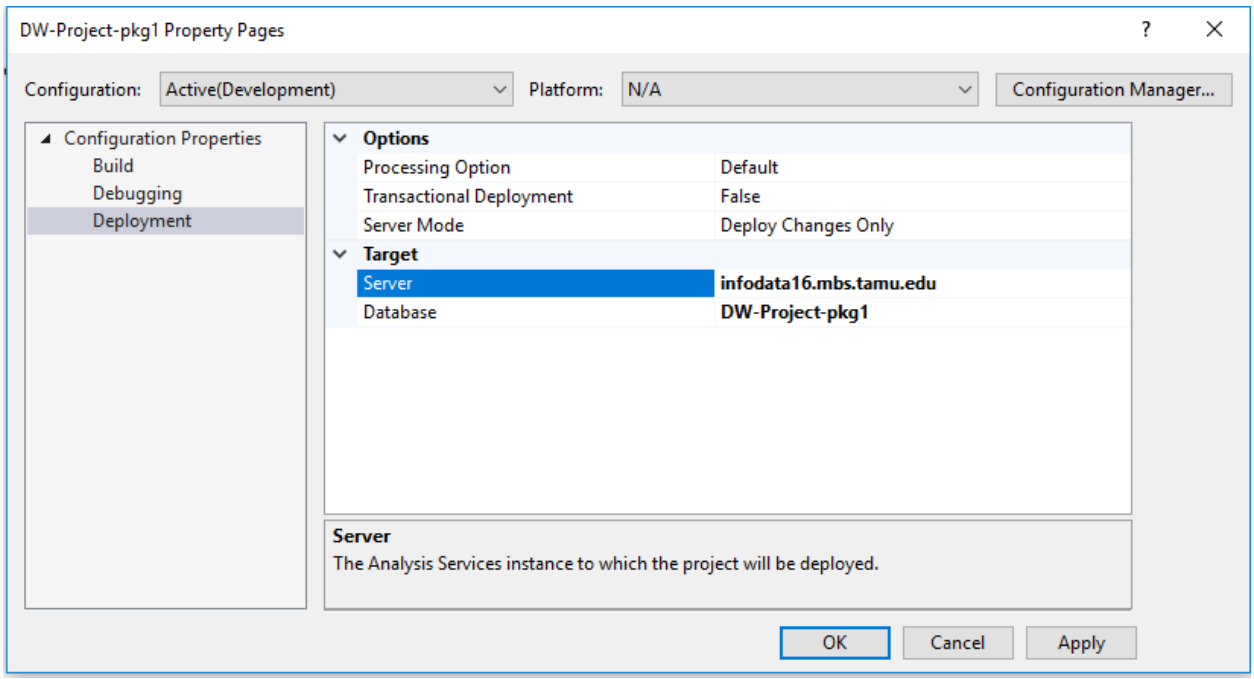
Explore factBakerySales Table | 602 Group2 Data W...e DB.cube [Design] | 602 Group2 Data Wa...se DB.dsv [Design]

Table

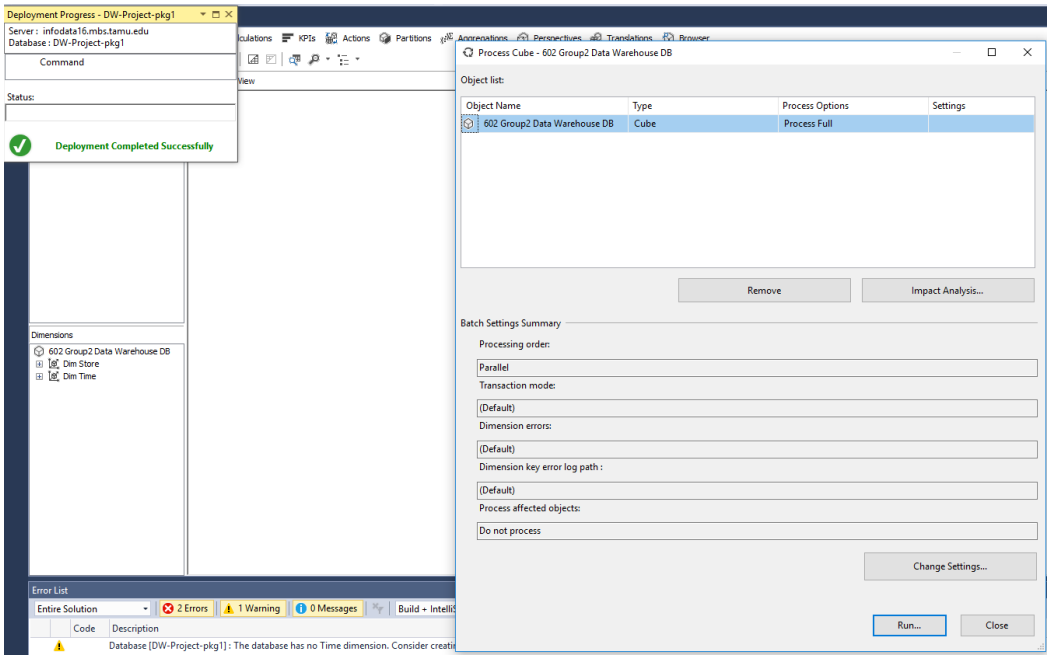
BAKERY	Store_Key	Time_Key
4592.18	32	17899
588031.49000000011	2	17900
412600.0100000003	3	17900
611968	4	17900
678144.42999999935	5	17900
611669.87999999942	6	17900
444557.09999999957	7	17900
523628.92	8	17900
854814.06000000052	9	17900
119709.95000000001	10	17900
482317.56999999989	11	17900
205526.00000000006	12	17900
365890.85999999987	13	17900
870266.7899999998	14	17900
246636.47	15	17900
117294.79999999996	16	17900
555113.97000000009	17	17900
375401.63999999996	18	17900
322667.47999999975	19	17900
425642.9499999999	20	17900
316717.97000000003	21	17900
309810.05999999988	22	17900
290710	23	17900
503793	24	17900
398694.87000000023	25	17900
326524.46	26	17900
105595.68	27	17900
362366.25999999989	28	17900
392862.89000000042	29	17900
20067	30	17900
362903.41000000009	31	17900
83005.339999999982	32	17900
216895.12000000002	33	17900
639615.31999999983	2	17901
298811.33	34	17900
384590.60999999987	35	17900
420994.79999999976	3	17901
372287.12999999989	36	17900
658366.31999999937	4	17901



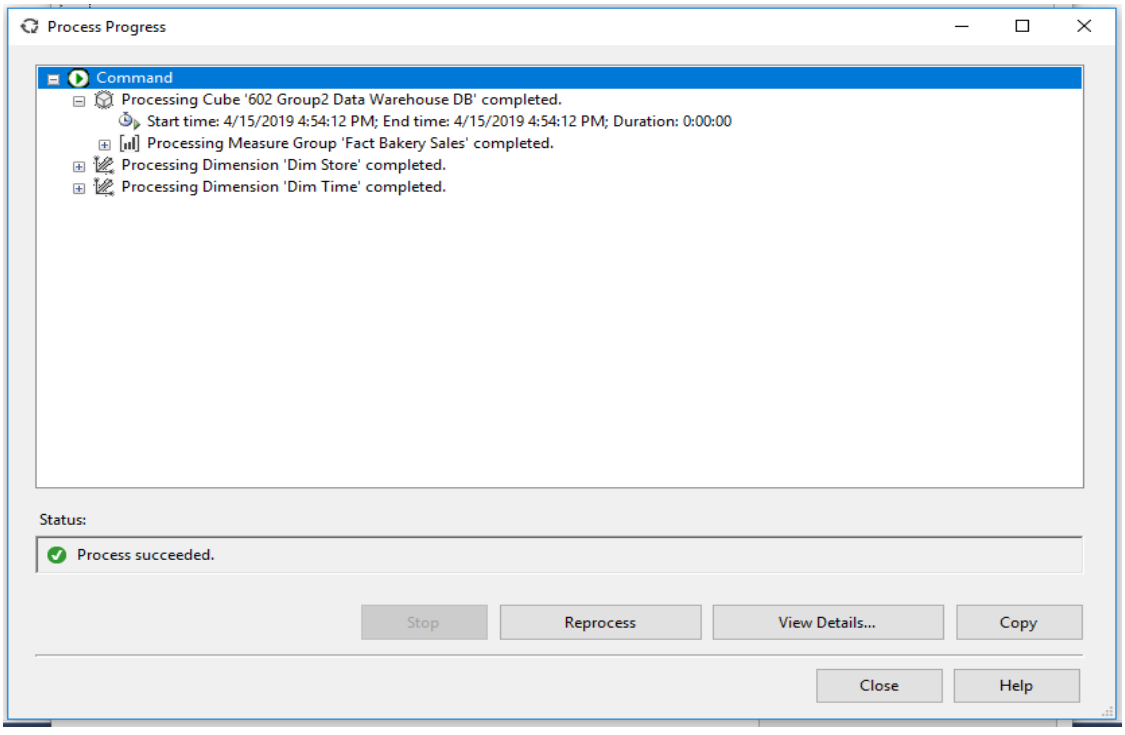
xvii. Once the cube is processed, target server can be manipulated to deploy the cube. We changed it from "localhost" to "infodata16.mbs.tamu.edu"



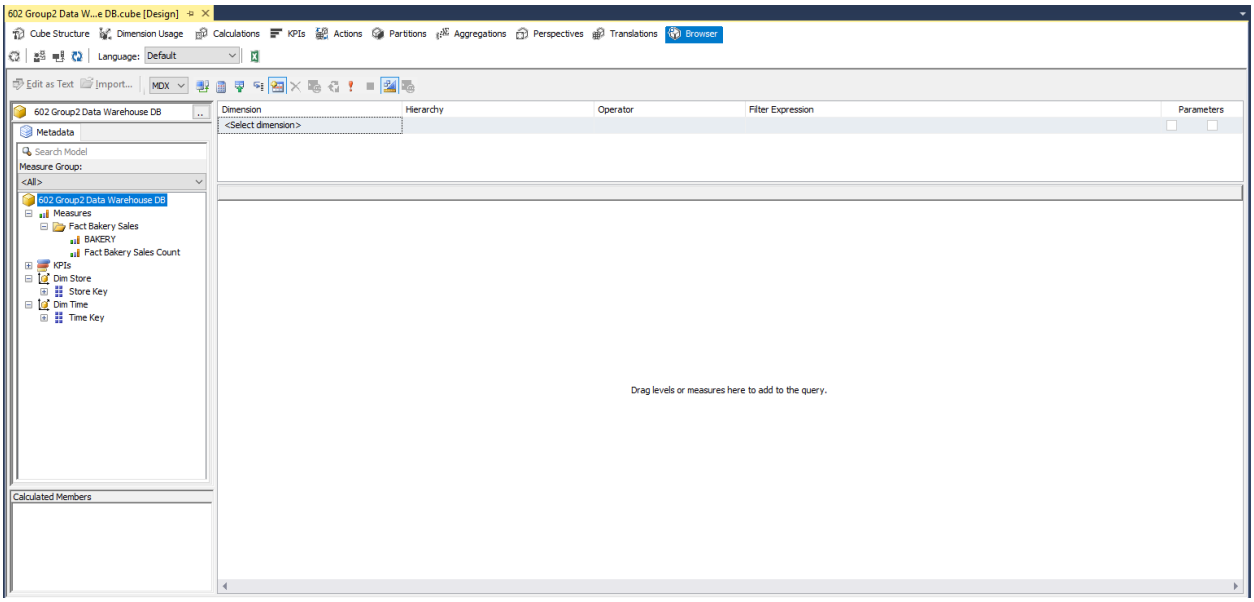
xviii. The cube is then deployed into the data warehouse

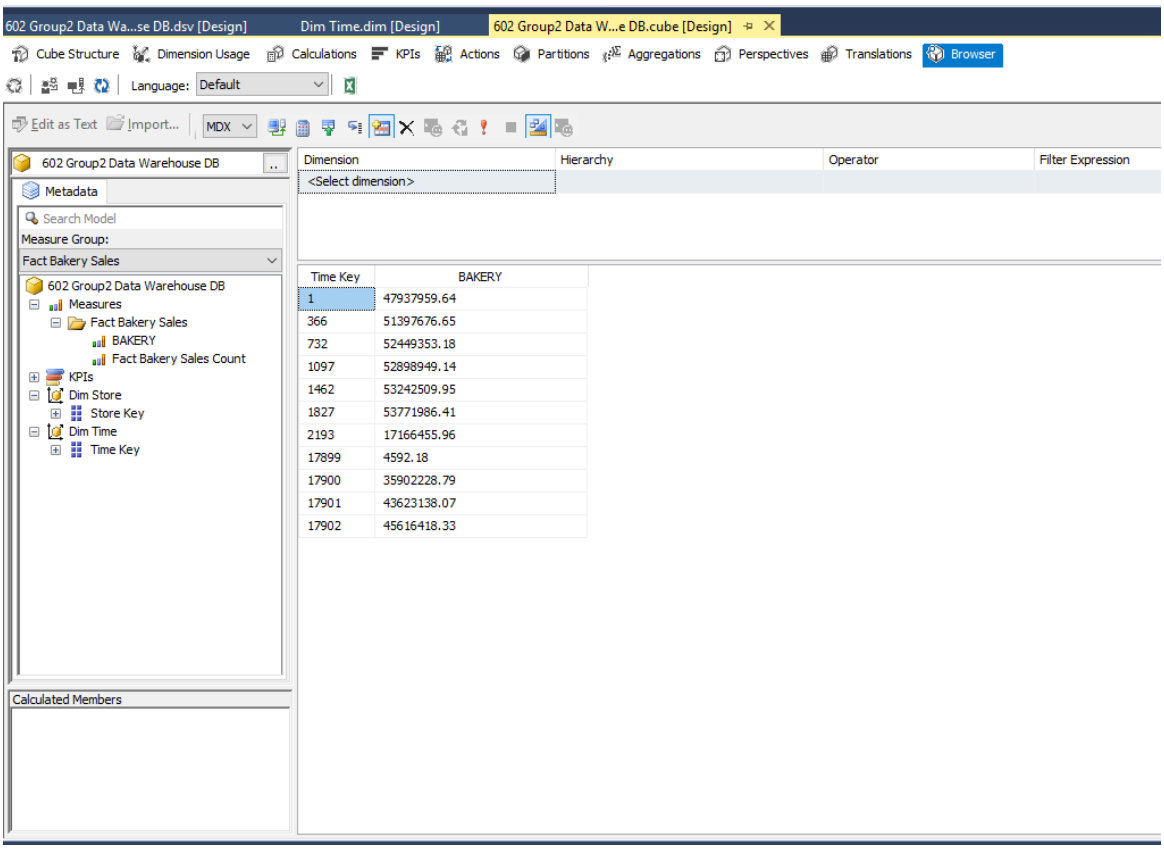


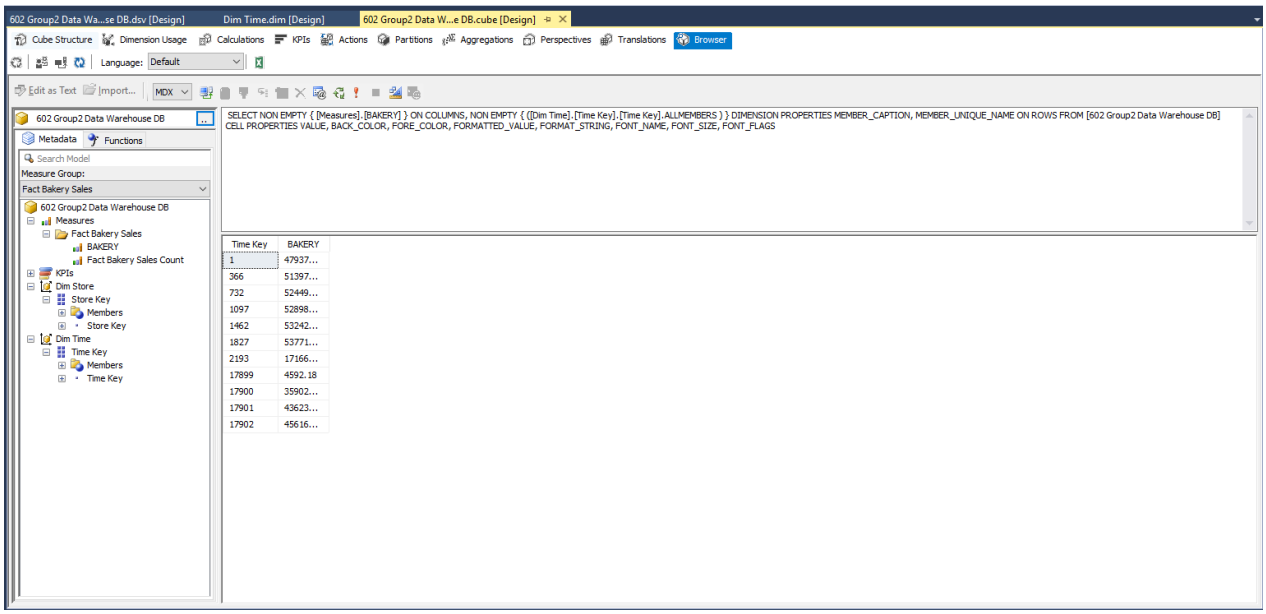
xix. Successful deployment:



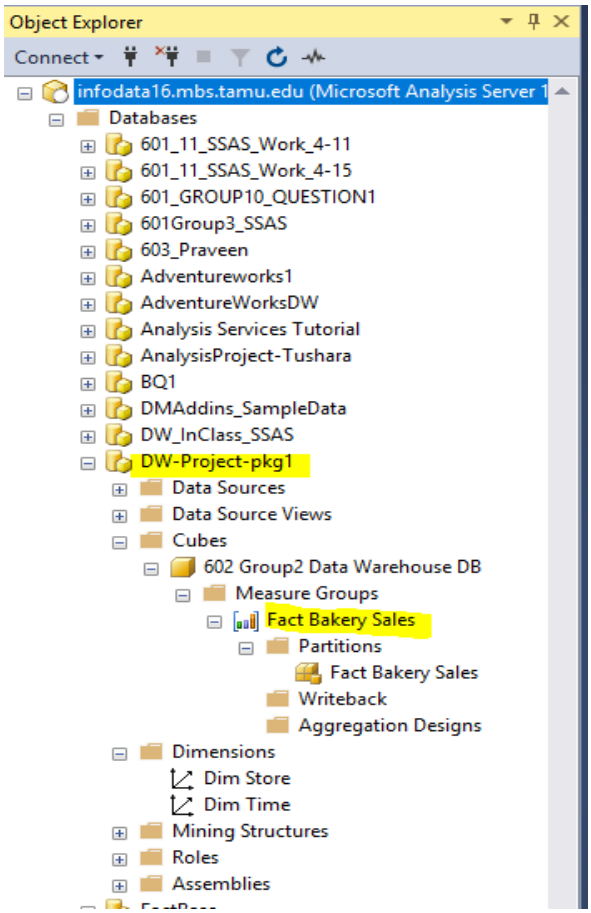
xx. Browser tab helps us utilize the columns from measures and dimensions to build our report



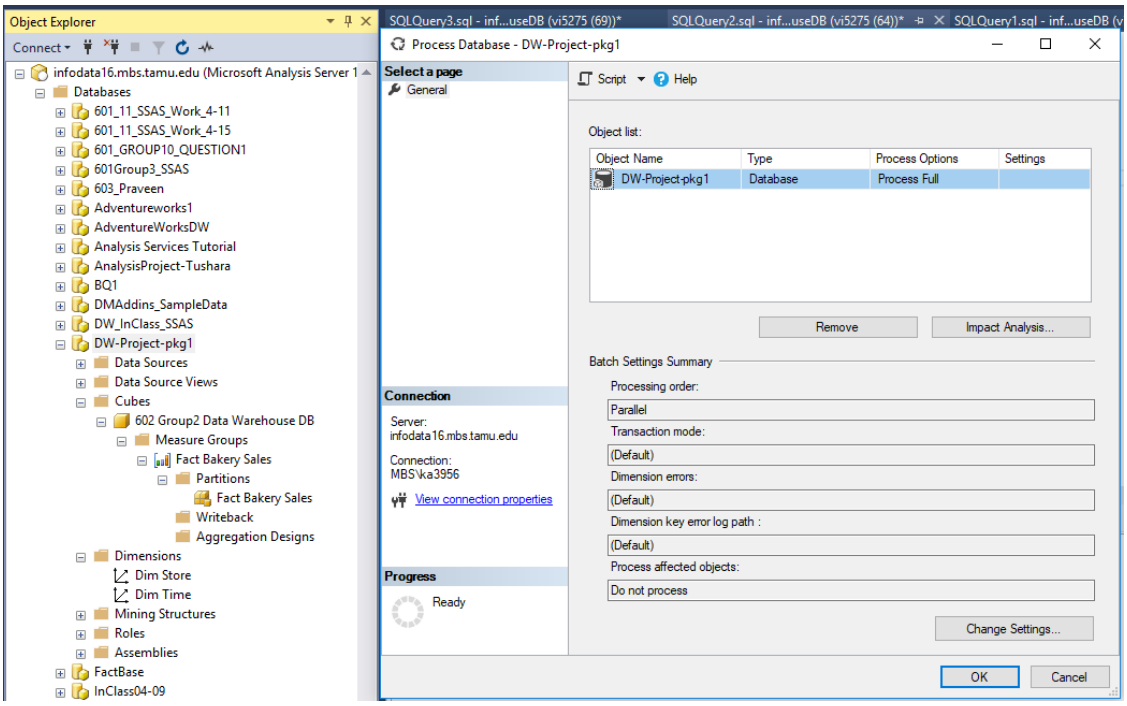


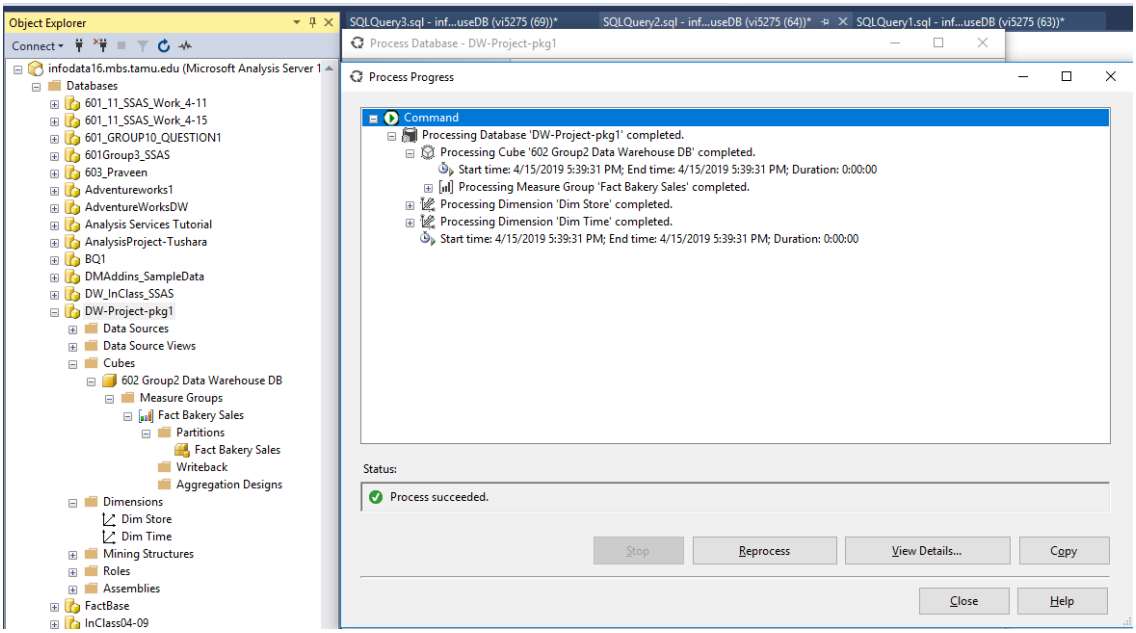


xxi. Once we verify the availability of data in the cube, we can also verify the success of deployment of the cube in data warehouse by connecting to Analysis Server in SQL server management.

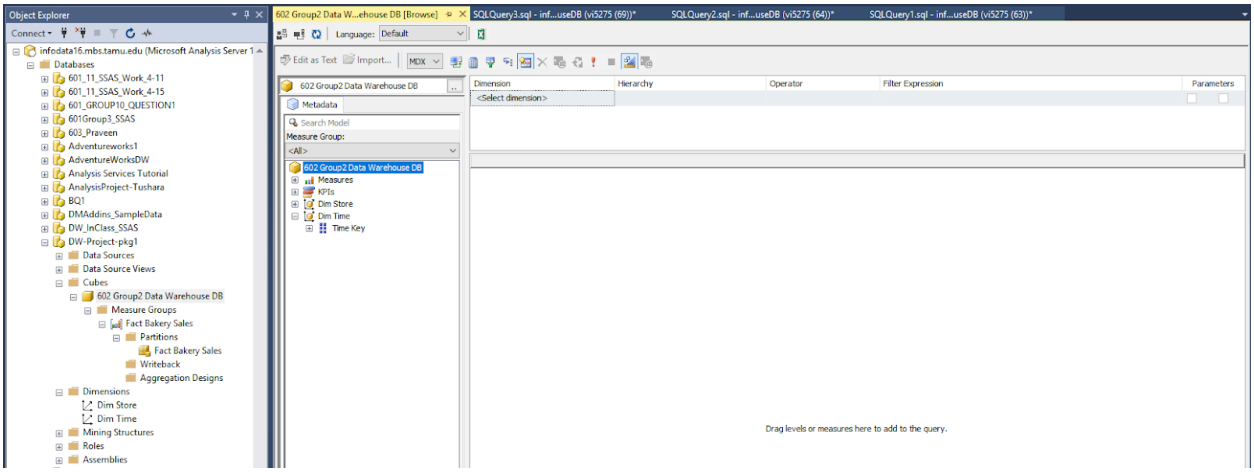


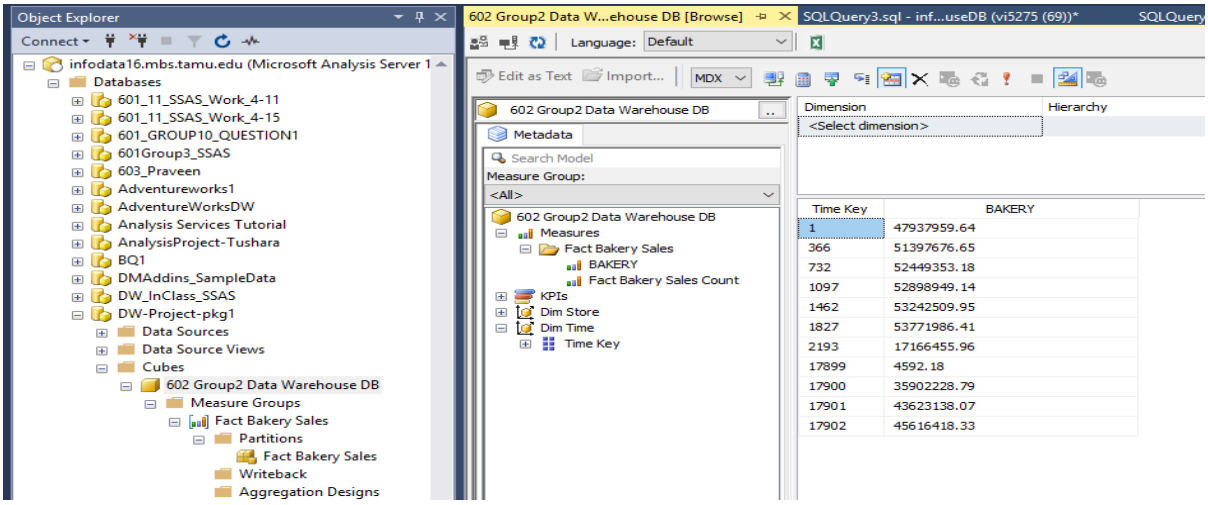
xxii. We then process the cube from data warehouse



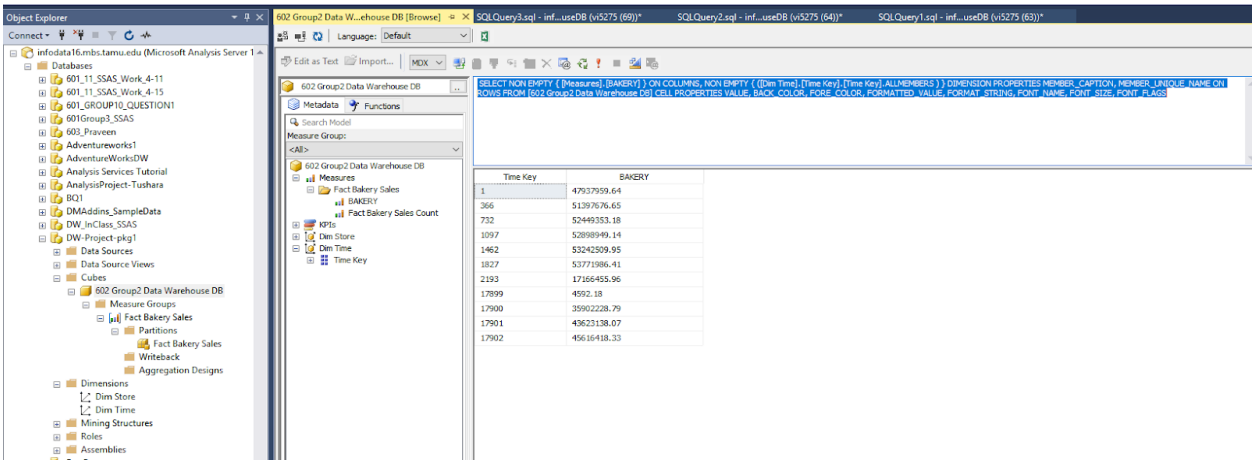


xxiii. Upon successful processing on the cube from both ends, we can access the cube at Analysis server to build our report by dragging and dropping the necessary columns from the measure and the dimensions.





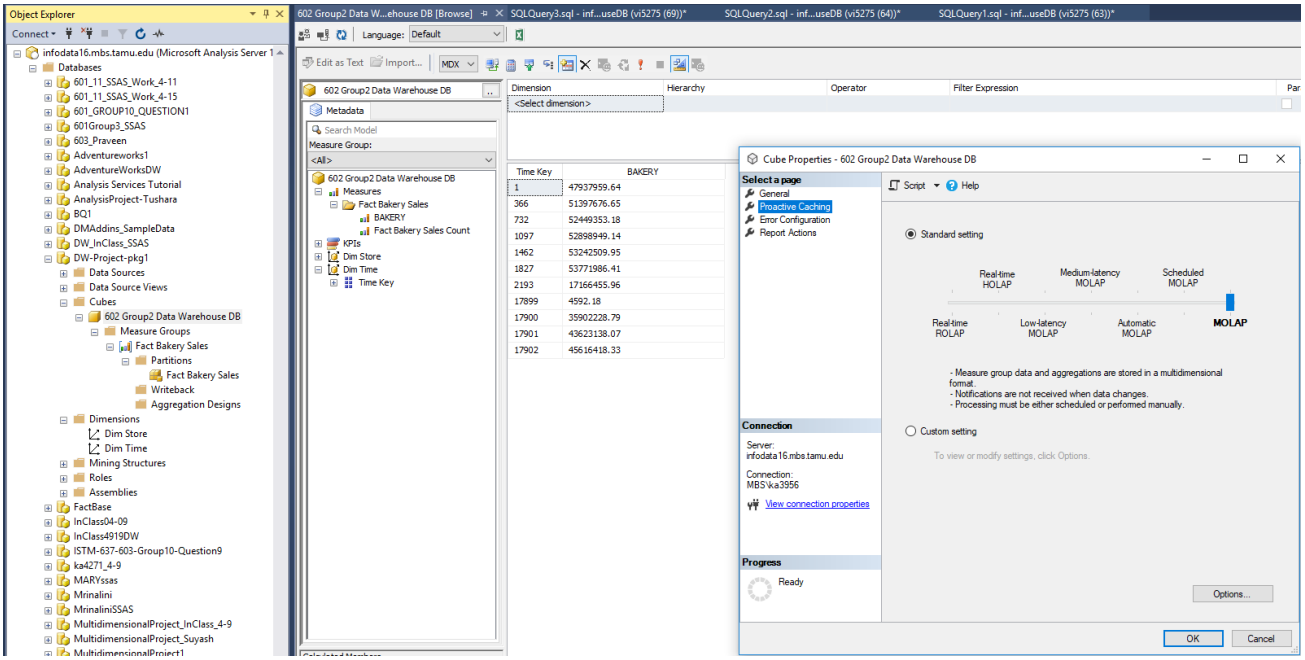
xxiv. The browser provides and automatically generated query for the report.



Auto-Generated SQL Query:

SELECT NON EMPTY { [Measures].[BAKERY] } ON COLUMNS, NON EMPTY { ([Dim Time].[Time Key].[Time Key].ALLMEMBERS) } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS FROM [602 Group2 Data Warehouse DB] CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR, FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS

xxv. Finally, the proactive catalog defines the type of OLAP it belongs to.



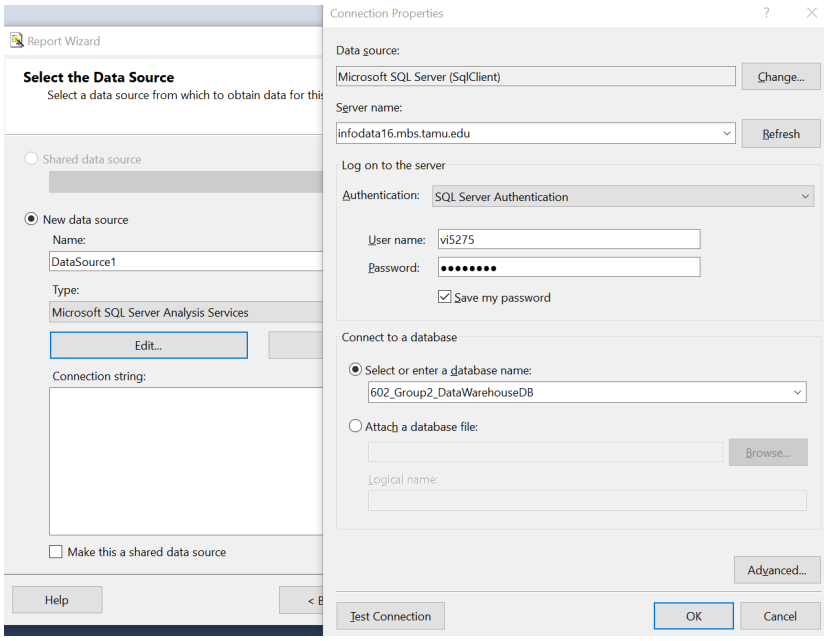
Conclusion:

We have provided solution for business question 3 using SSAS. The report provided above depicts the sales for Bakery category for the period 1990 to 1996. The report can be effectively used to identify the trend of bakery sales for any given period. Likewise, many such sales trends can be calculated for different categories and thus their trends can be evaluated using this process. Based on this trend analysis of bakery sales data of DFF, decisions can be implemented to employ suitable branding and marketing for the products that are most suitable for the customers in those particular stores.

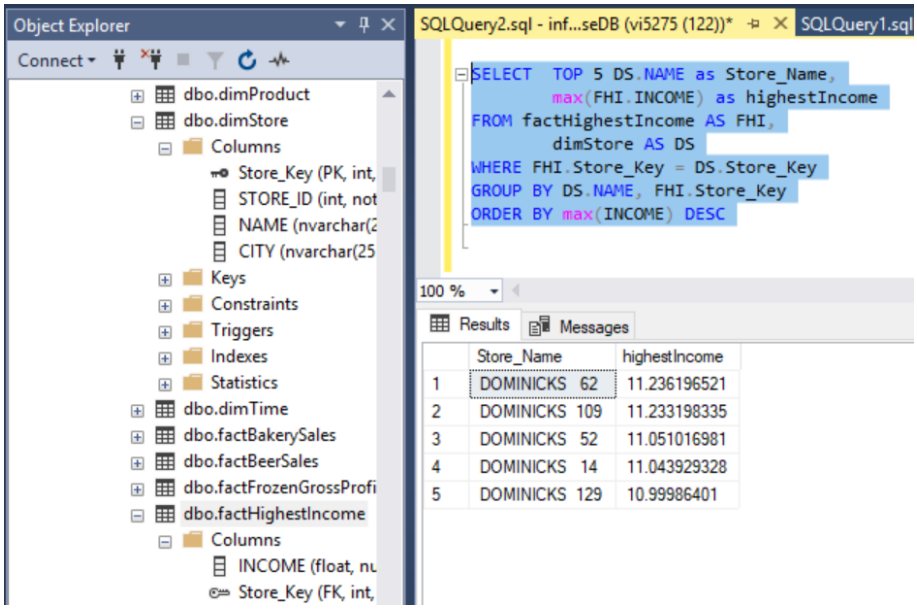
Q4. Which of the top 5 stores have customers with highest % income?

For business question 4 we will use SSRS to build a report to display the top 5 stores which have customers with highest income.

i. We start with launching SSRS report wizard using MS visual studio and then connect to our data warehouse using Edit button. We thus will use our existing data warehouse as our data source to create the report.



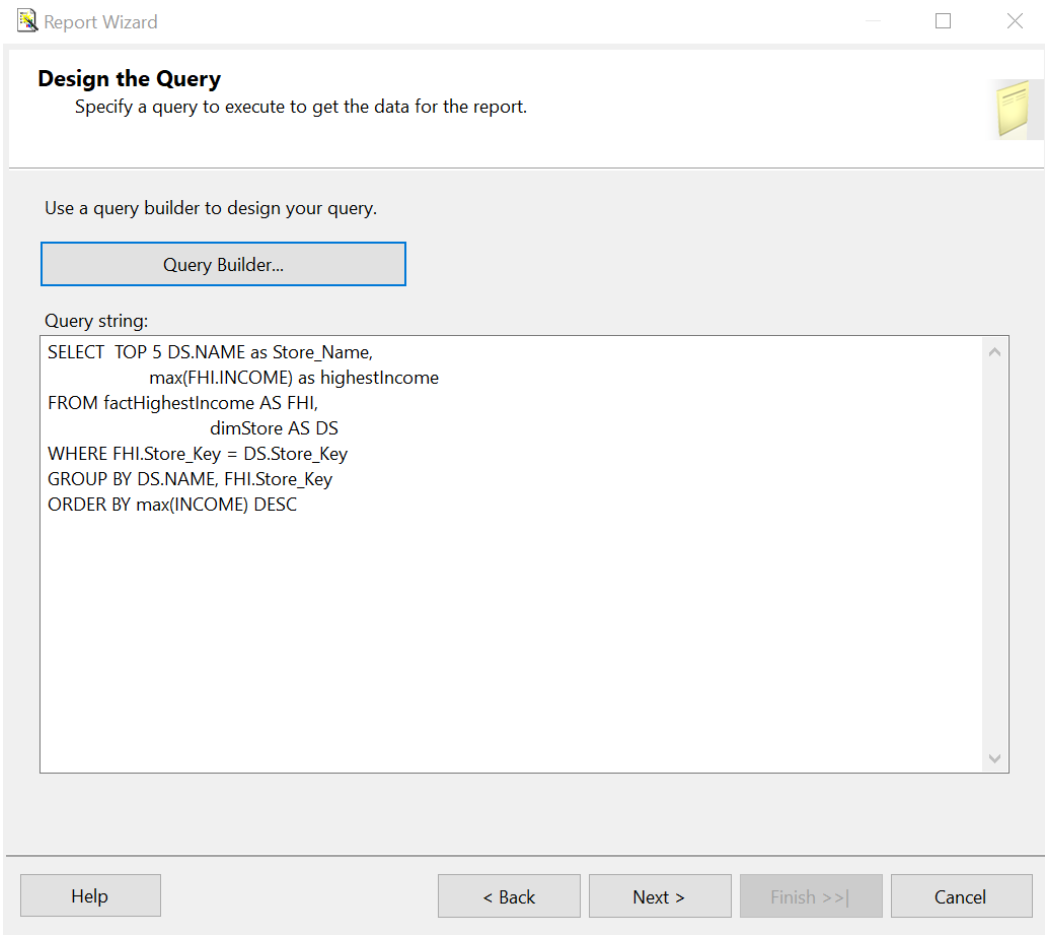
ii. The output we desire can realized by running the following query at the data warehouse directly to realize the desired output as shown which later can be verified using SSRS features.



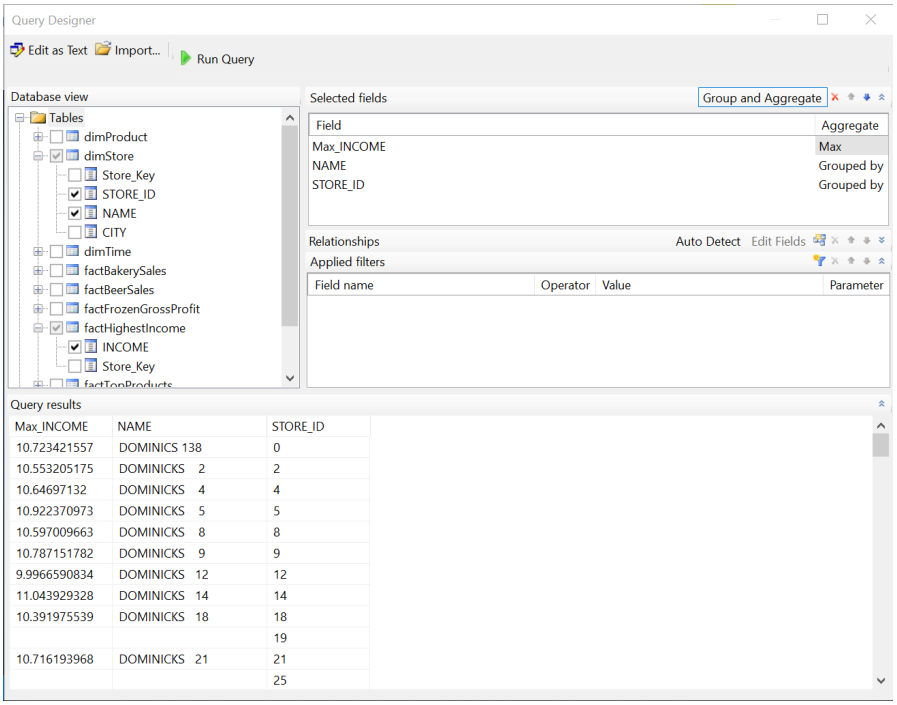
SQL query:

```
SELECT  TOP 5 DS.NAME as Store_Name,
        max(FHI.INCOME) as highestIncome
FROM factHighestIncome AS FHI,
         dimStore AS DS
WHERE FHI.Store_Key = DS.Store_Key
GROUP BY DS.NAME, FHI.Store_Key
ORDER BY max(INCOME) DESC;
```

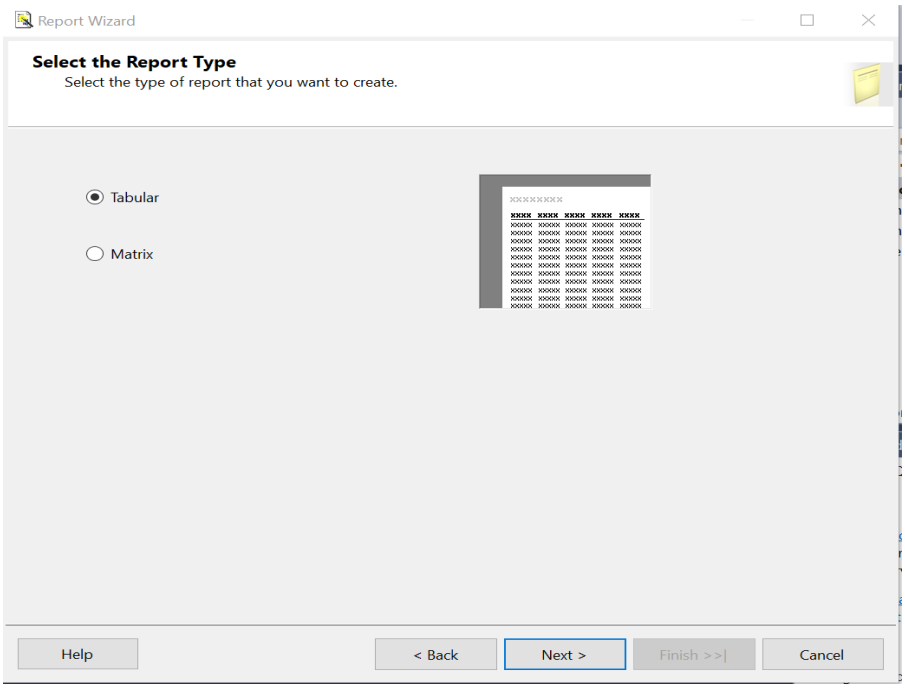
iii. For convenience, same query can be utilized as query string to get appropriate results. Query builder is a quick feature to explore the objects and attributes to form the query on the fly as shown in the next image.



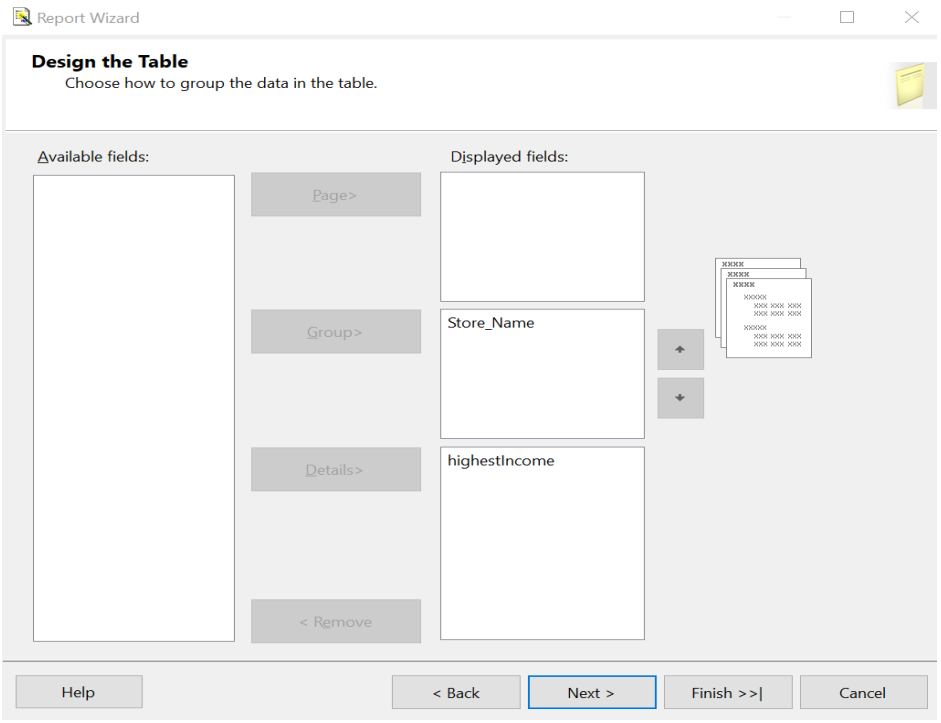
iv. Query designer option is the conveniently provided by the report builder to utilize the dimensions and fact tables from our data warehouse to build a mock-up of the report using aggregate functions and also via drag and drop facility of the tables and attributes. Aggregate and group by functions can be used to form the appropriate query. We will use dimStore, dimTime and factHighestIncome for the query.



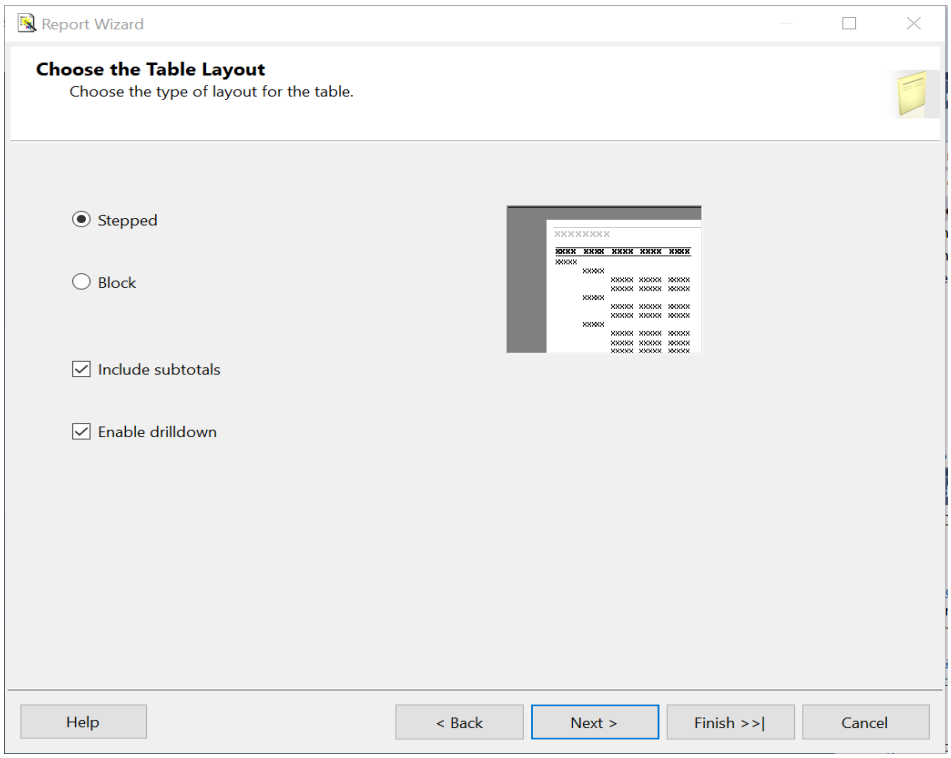
v. The type of the report is selected as Tabular



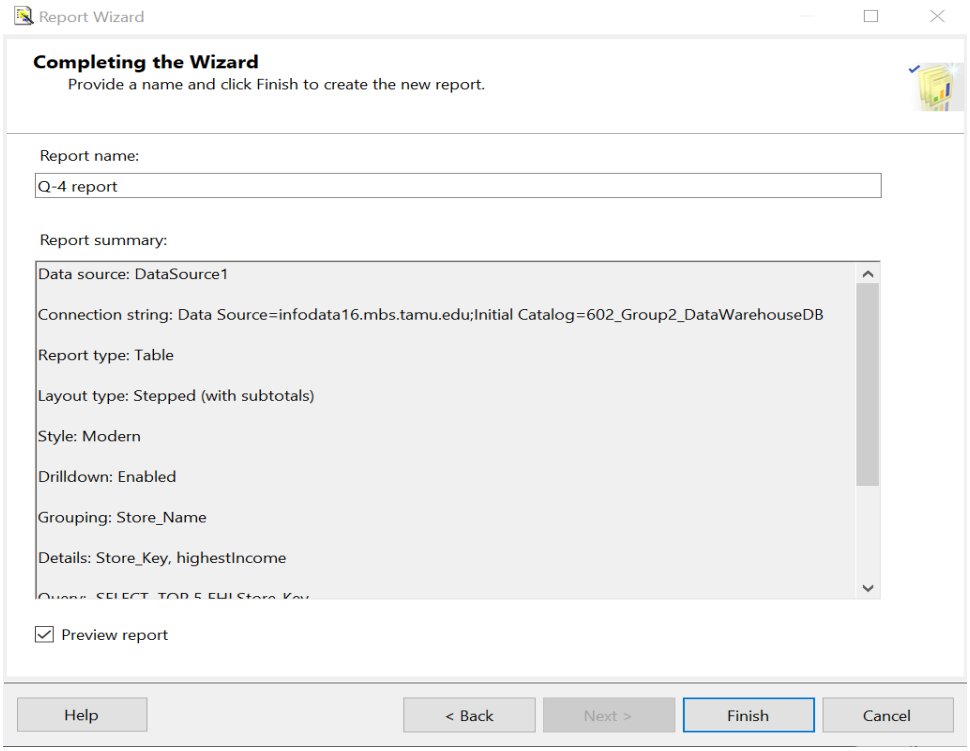
vi. To provide aggregation properties to the output table, fields are arranged in appropriate categories in the design table window. These aggregation can also be modified after building the report or at any time we plan to include or exclude aggregations.



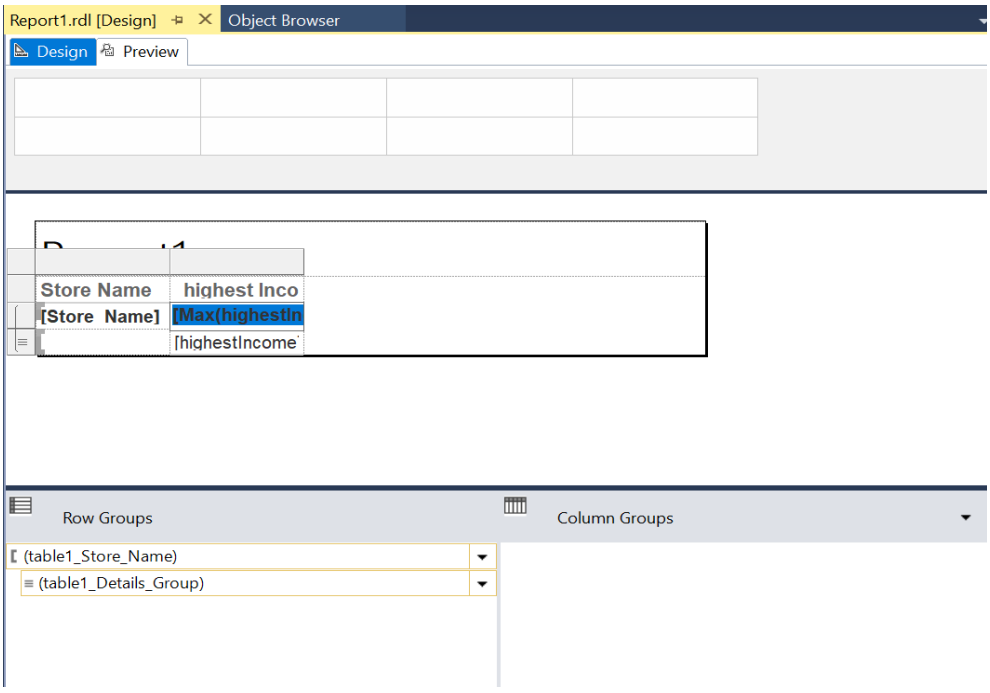
vii. We then finalize the table format necessary



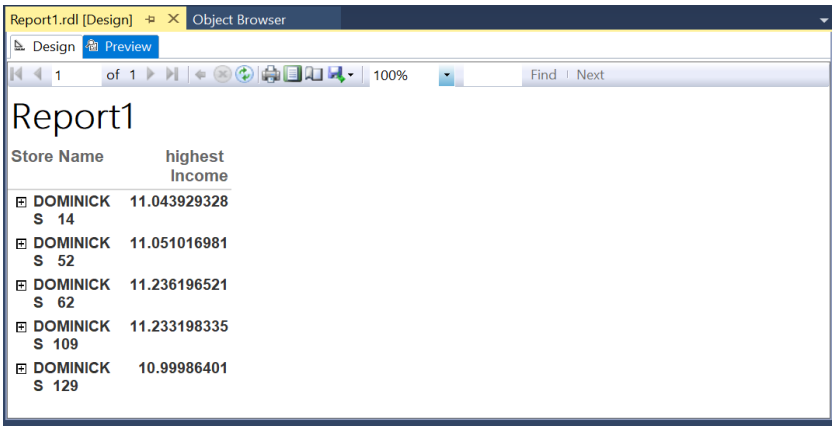
viii. We then name the report and finish the wizard process



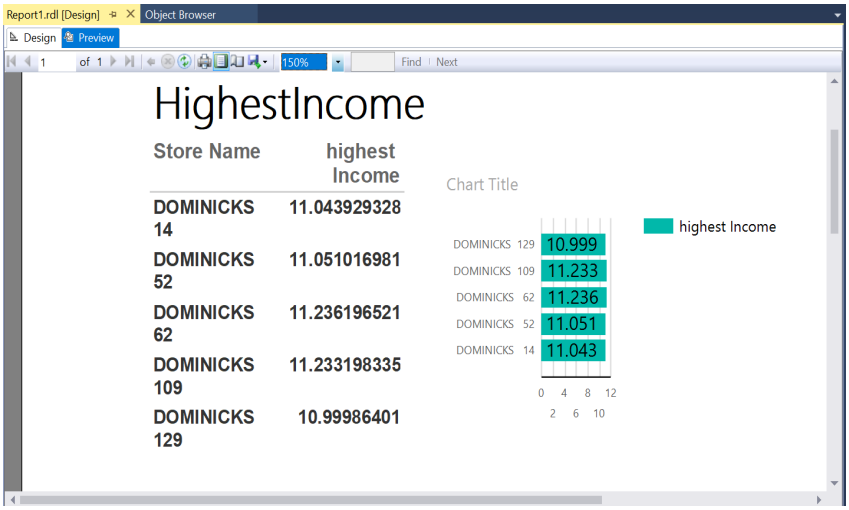
ix. At the end of the report wizard, table containing necessary attributes is created and displayed as follows under design tab



x. This report can be previewed using preview tab and thus display the report as follows



xi. The table has multiple graph and charting options to display the data visually based on the values and number of data outputs. Since the values are so close to each other and due to scaling constrains we chose horizontal bar graph displaying bar values inside as follows. The following preview displays the top 5 stores with highest income% in DFF. We can visually display the 5 records using SSRS.

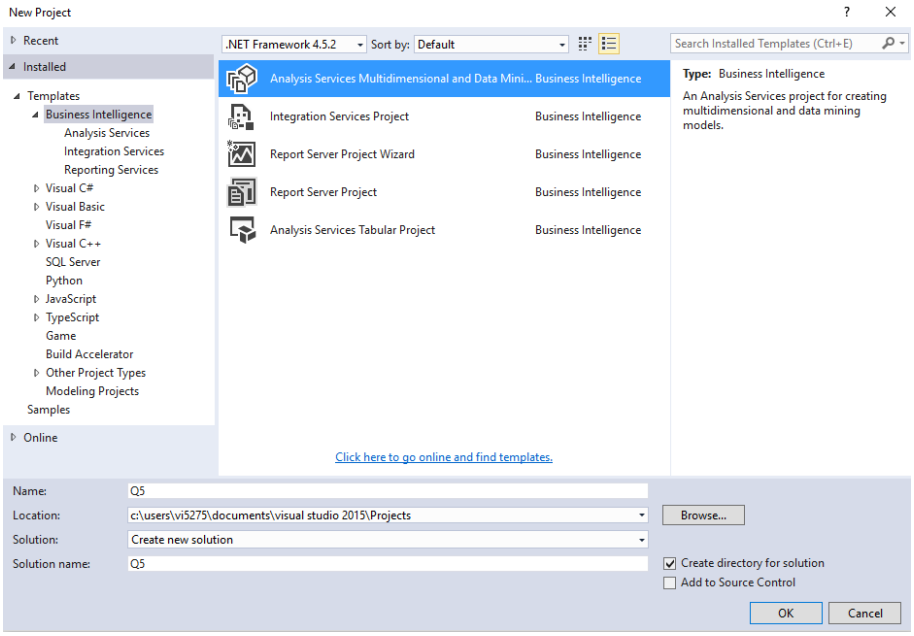


Conclusion:

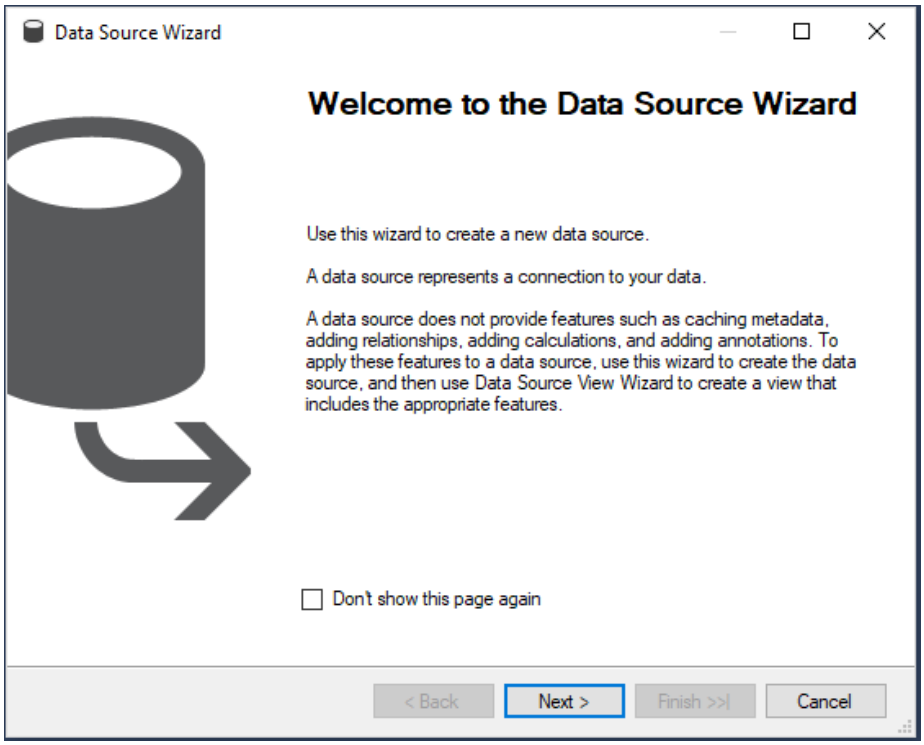
We have provided solution for business question 4 using SSRS. The report and the charts provided above depict the top 5 stores which have highest % income among all of the given stores this was derived primarily from the measures added into the fact table. In the graph has store name on the y-axis and income of the customer with highest % income values on the x-axis. The report can be effectively used to identify the top 5 highest % income of the customers and which stores they belong to. Likewise, many other characteristics of the customers and their trends can be evaluated using this process. Based on this trend analysis of customer information data of DFF, decisions can be implemented to employ suitable branding and marketing for the products that are most suitable for the customers in those particular stores.

Q5. What are the top 5 products sold in the last year?

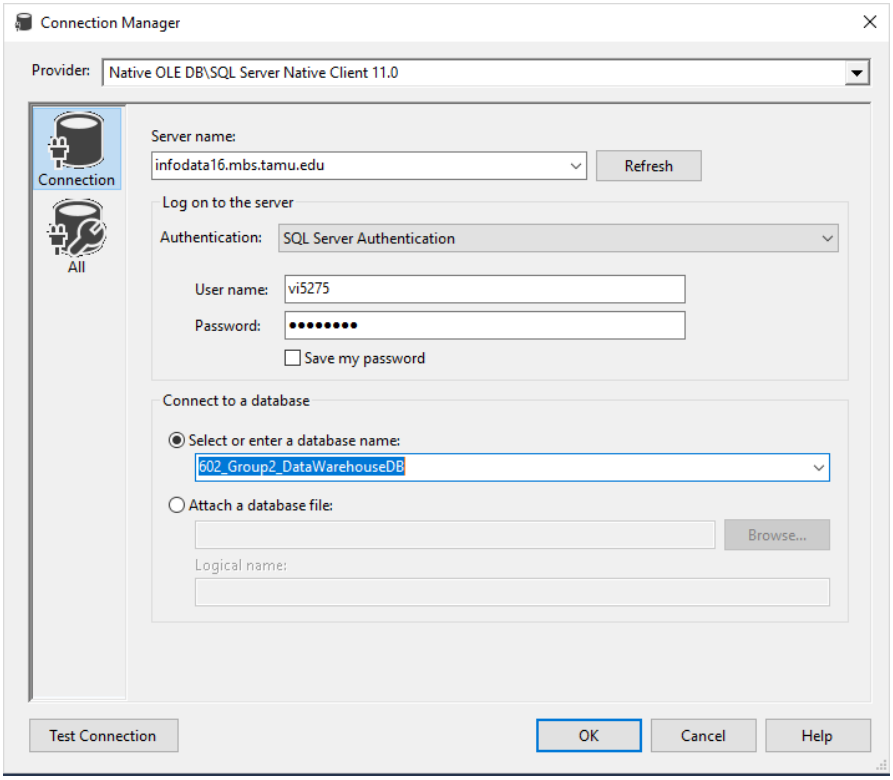
i. First create the data cube by creating a SSAS new project

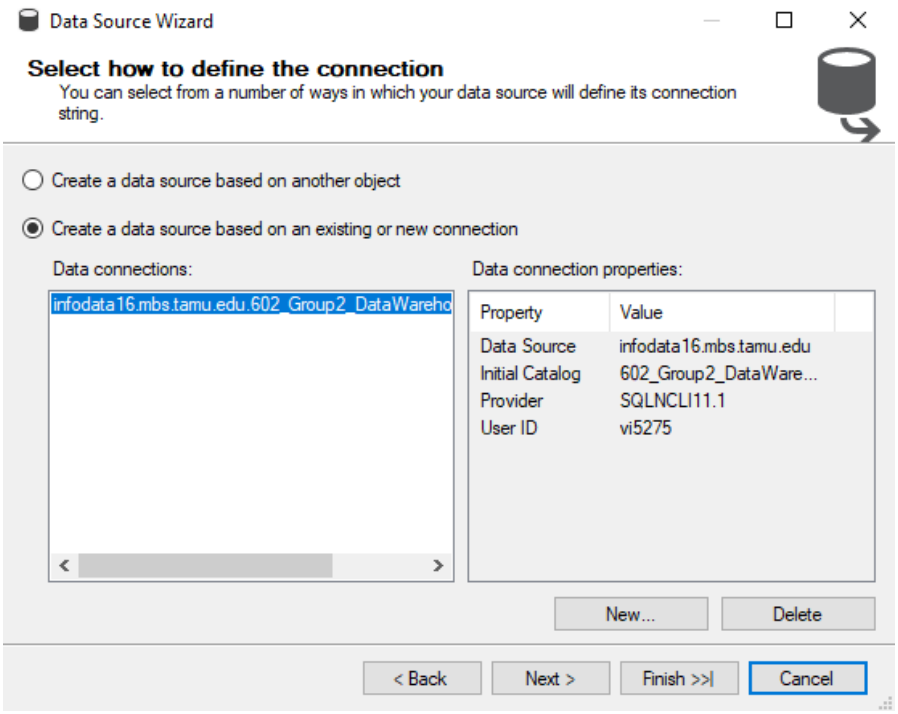


ii. Creating a data source pointing to the data warehouse database.

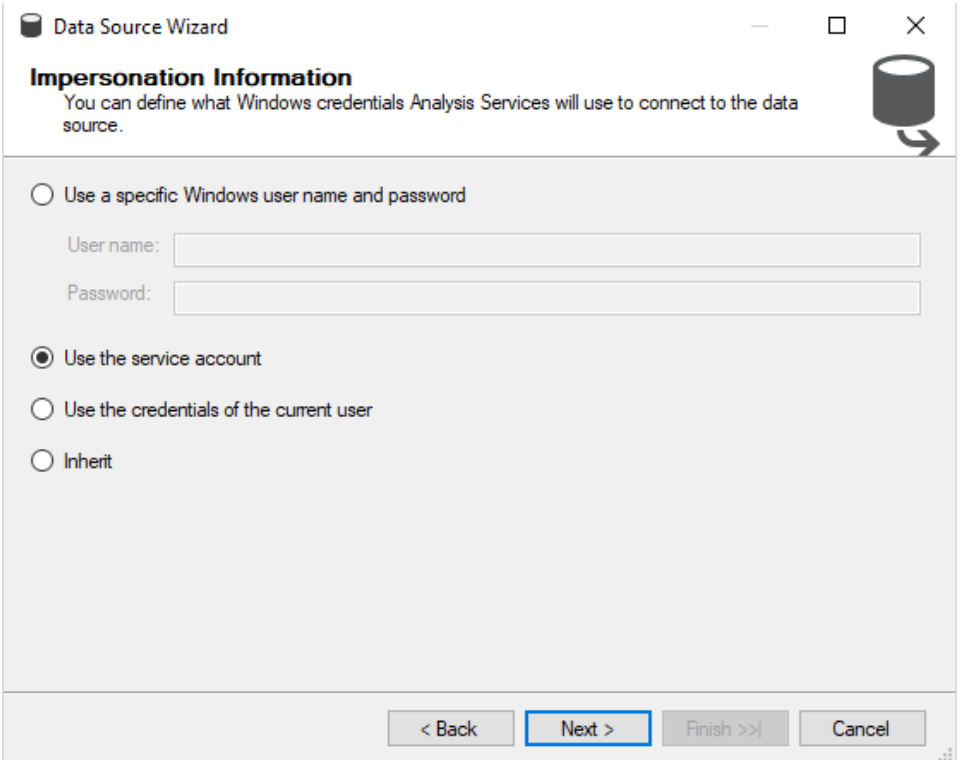


iii. Create the new connection to the server - infodata16.mbs.tamu.edu, also enter the credentials and select the database as the data warehouse.

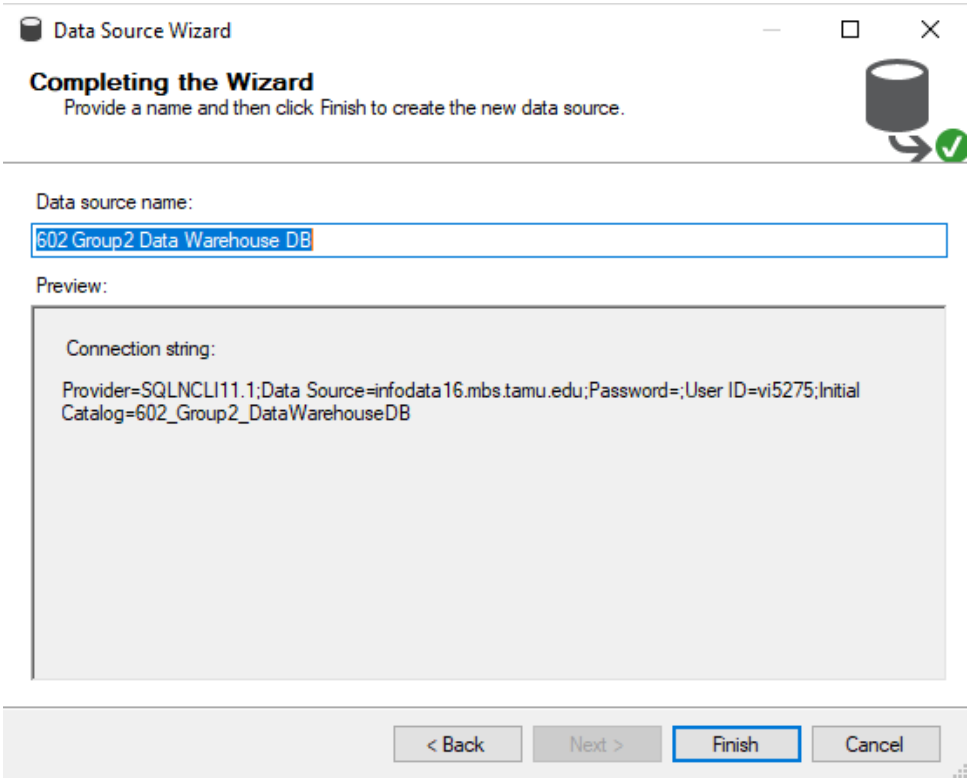




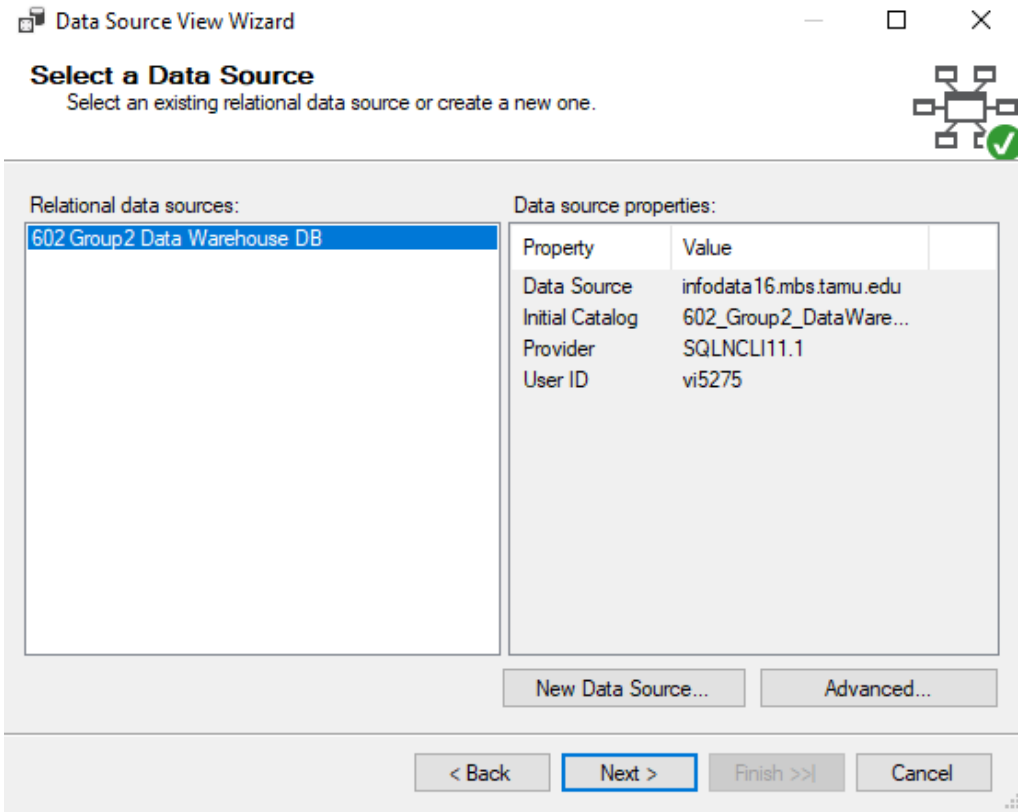
iv. Specifying the impersonation details



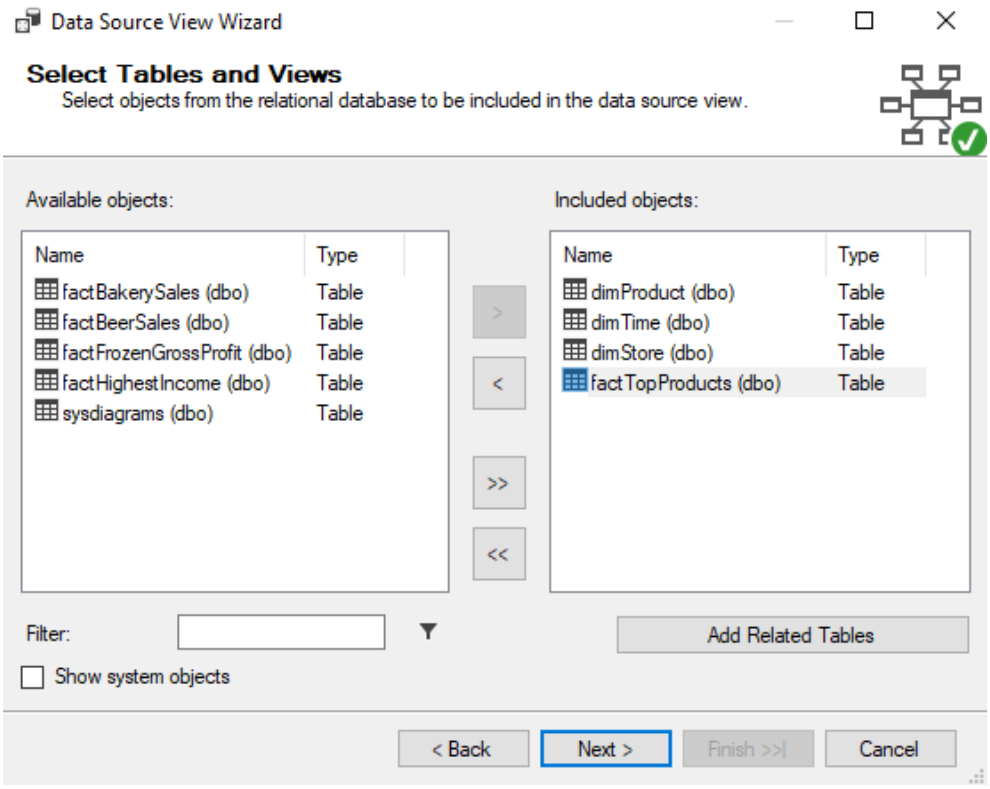
v. Complete the wizard and click finish

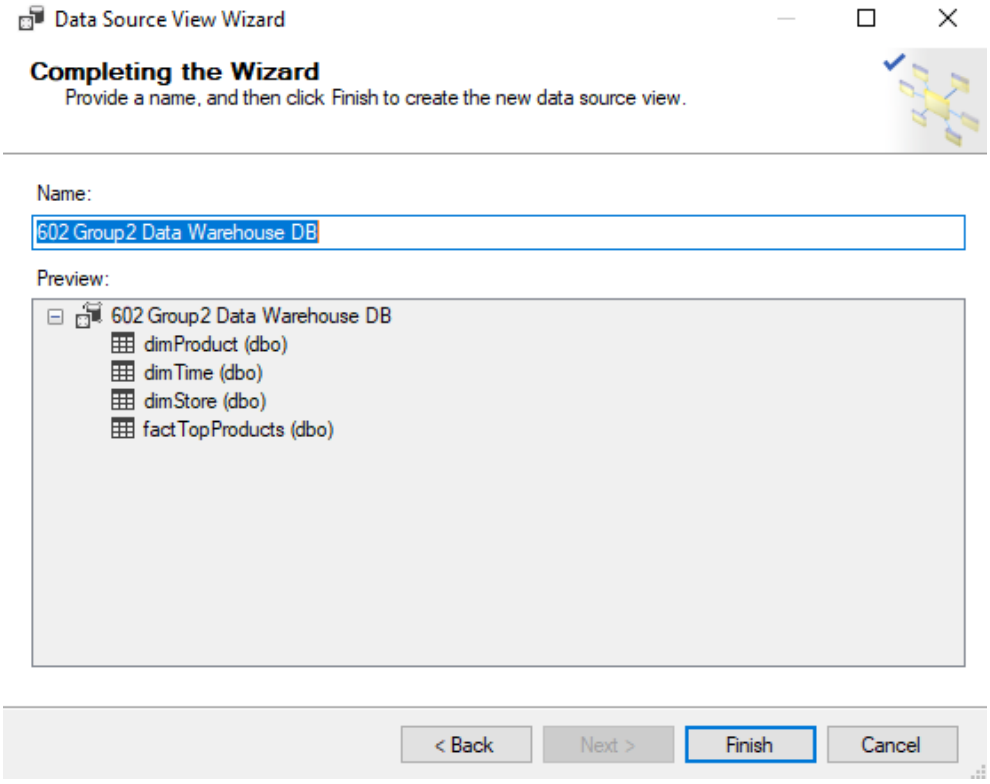


vi. Creating Data Source View

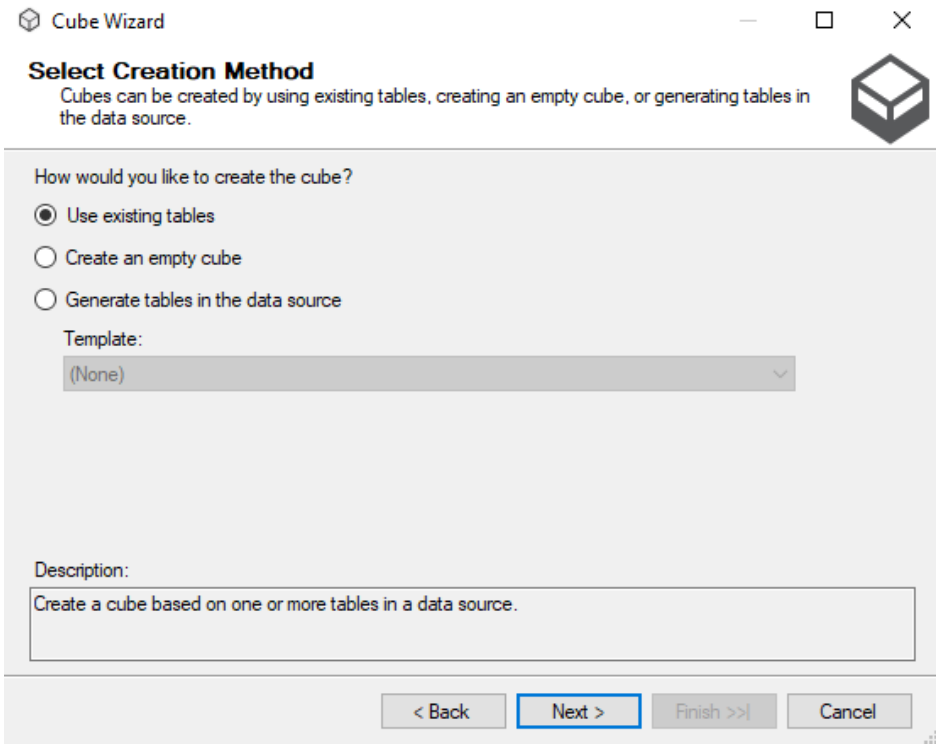


vii. Select the dimension and fact tables needed

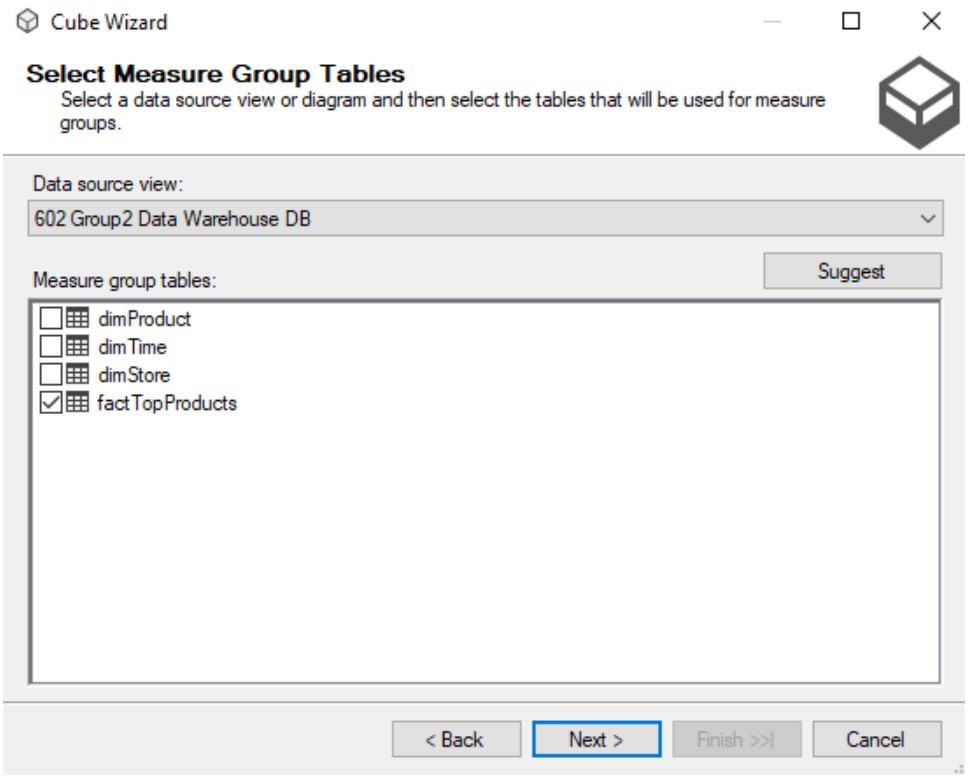




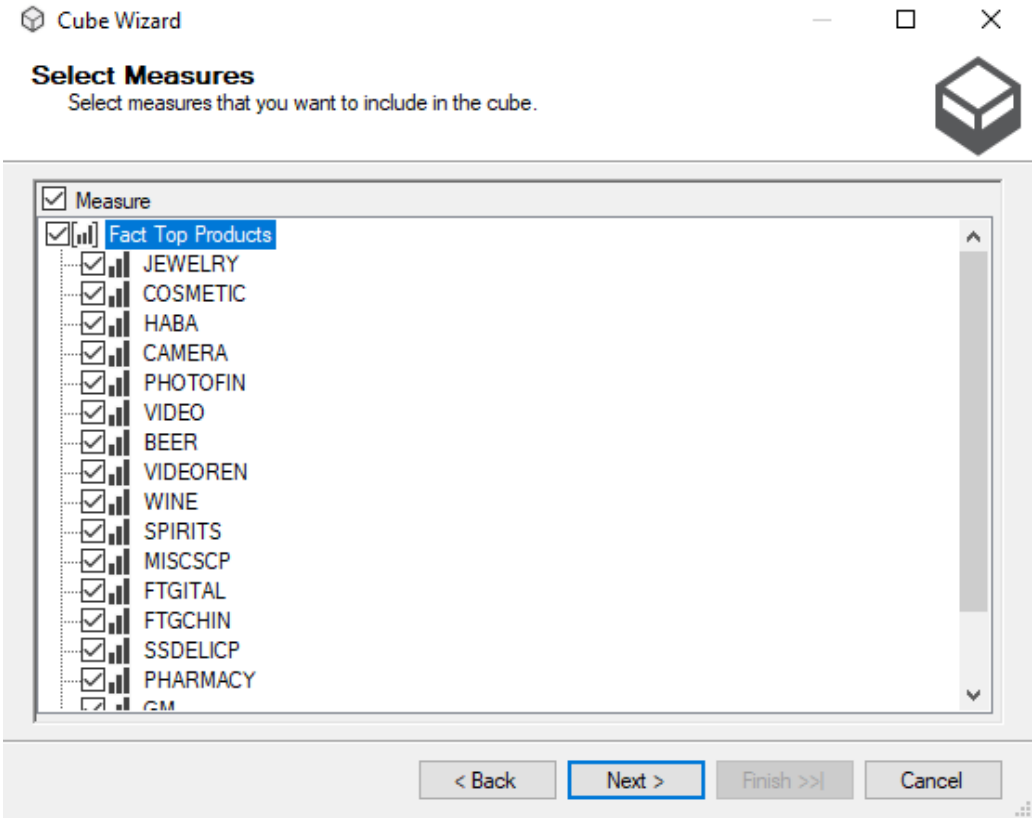
viii. Creating Cube:



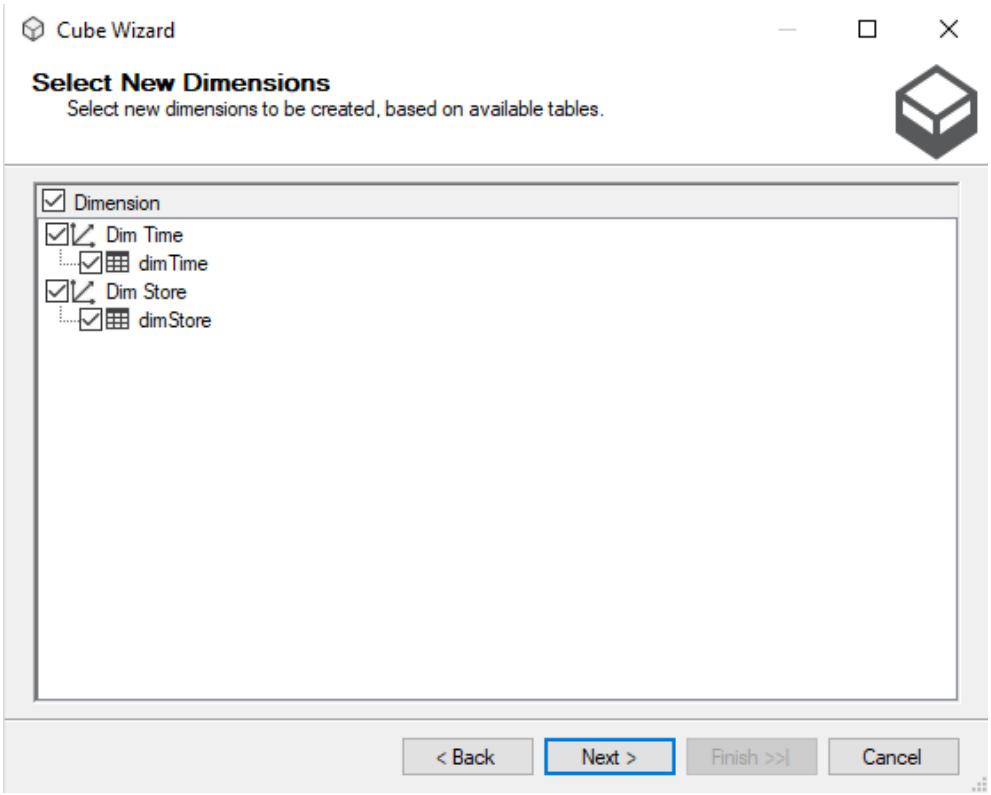
ix. Select the required fact table

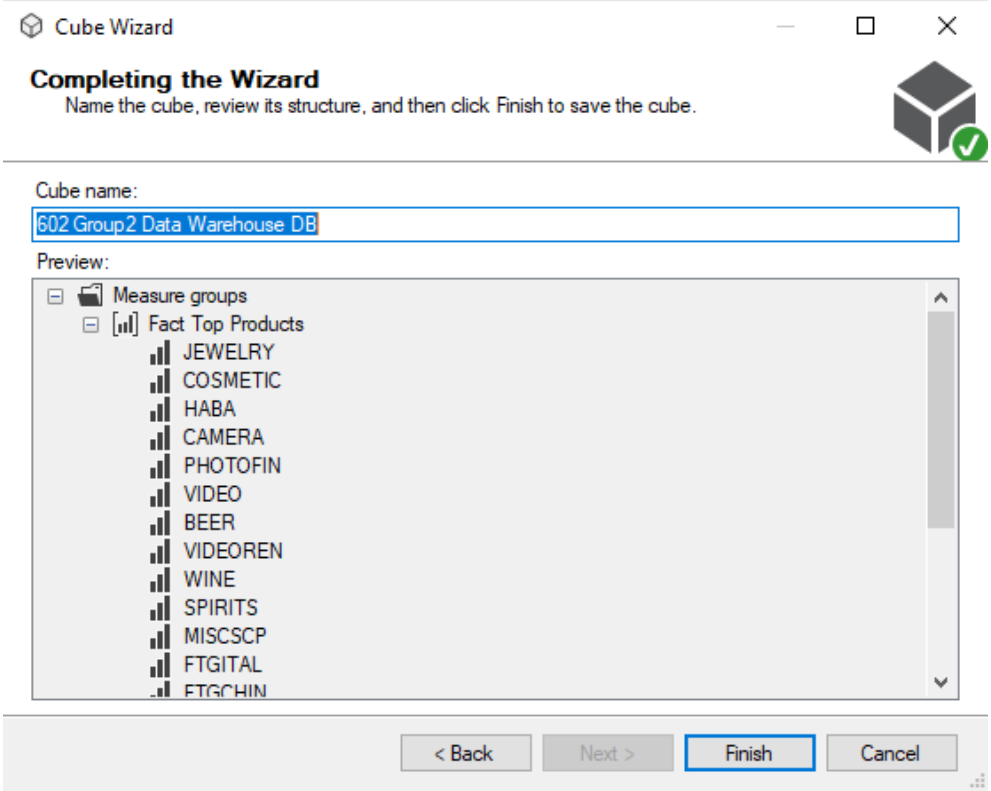


x. Select the appropriate measures:

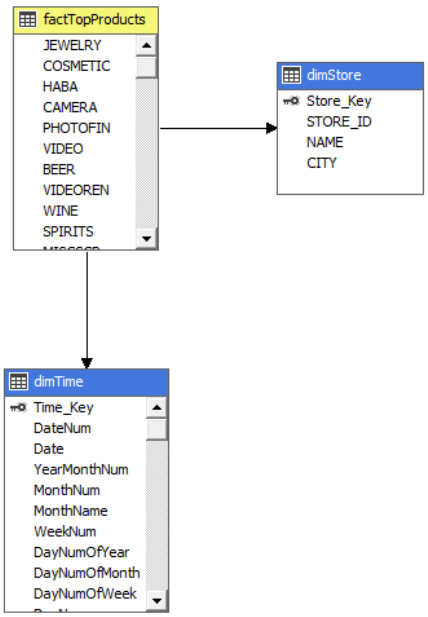


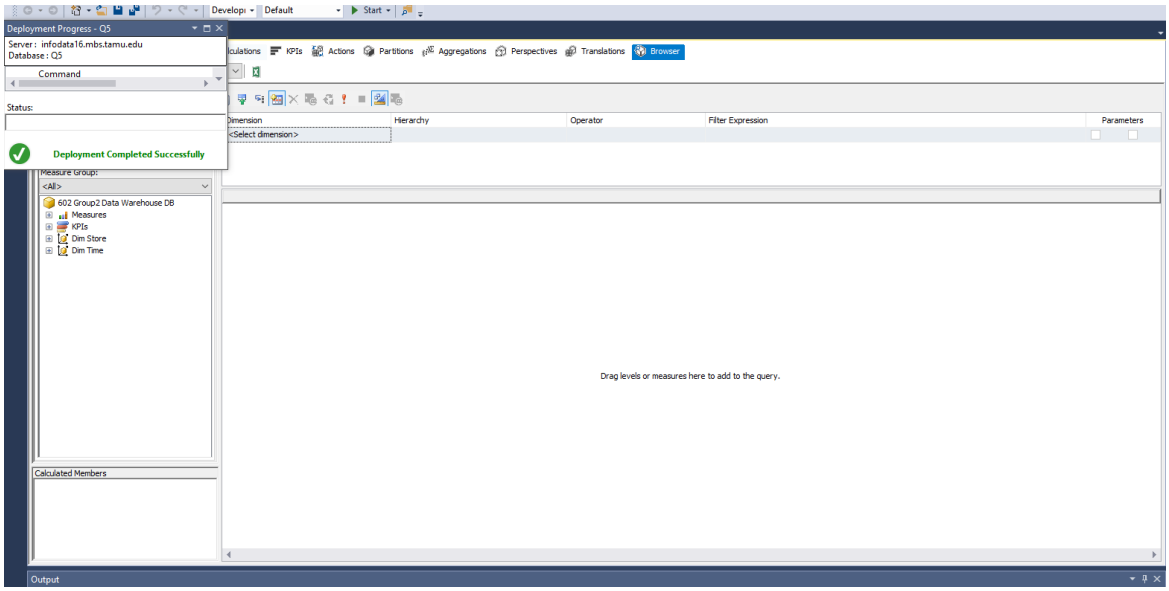
xi. Select the dimensions:



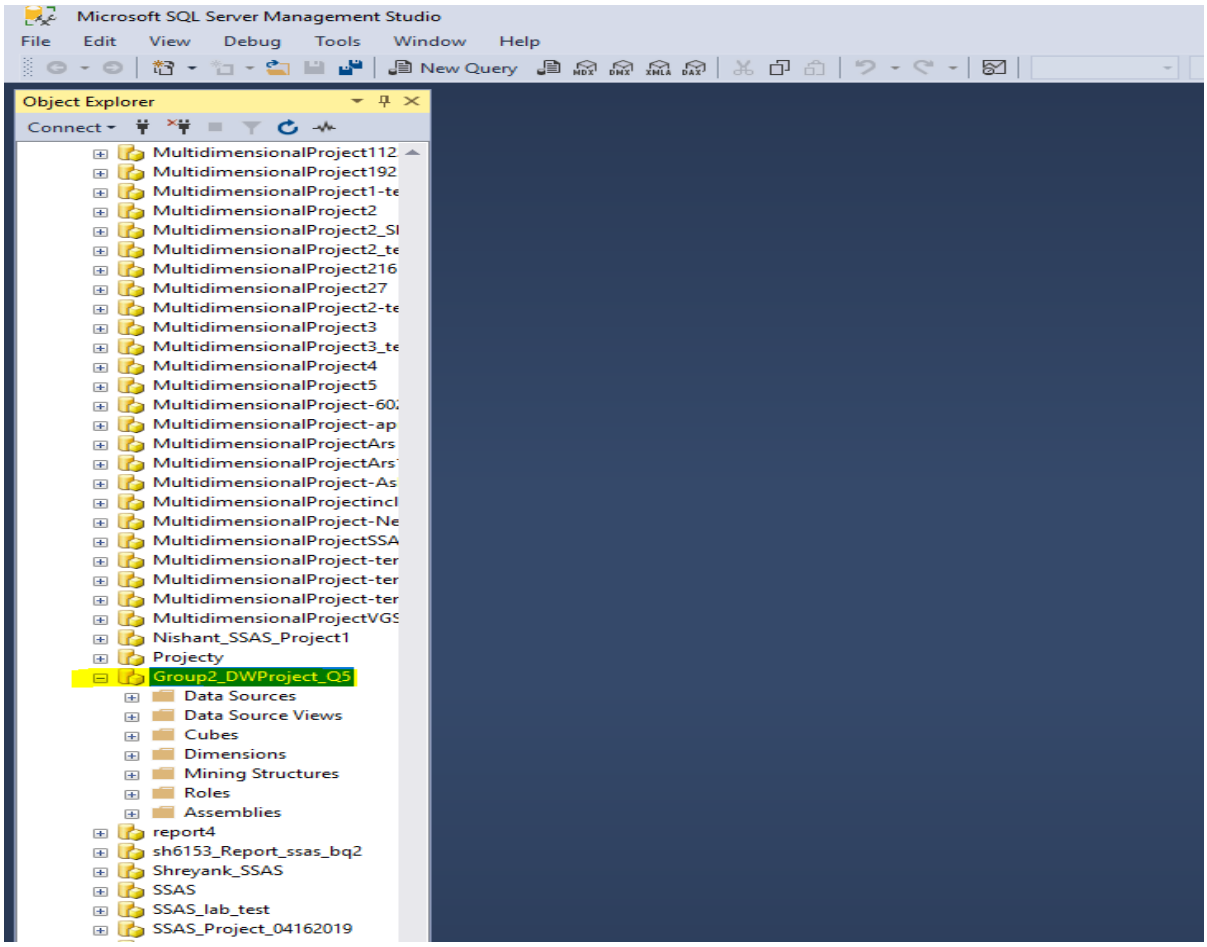


xii. The cube is created with the following relations

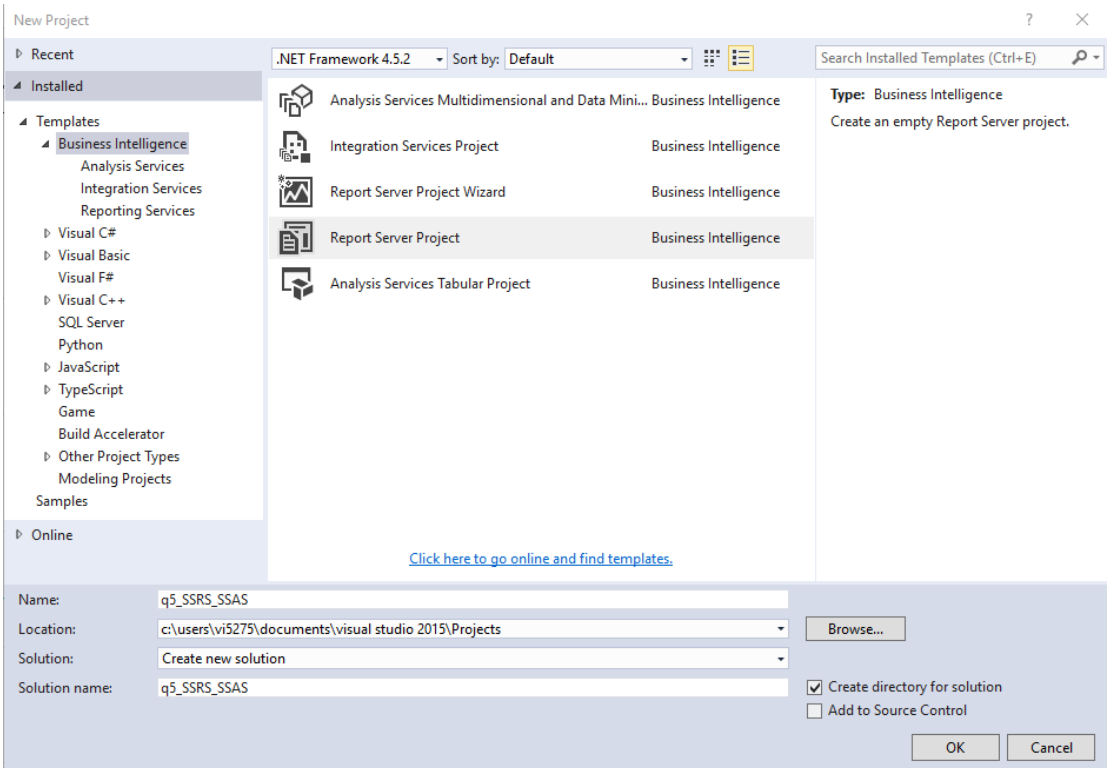




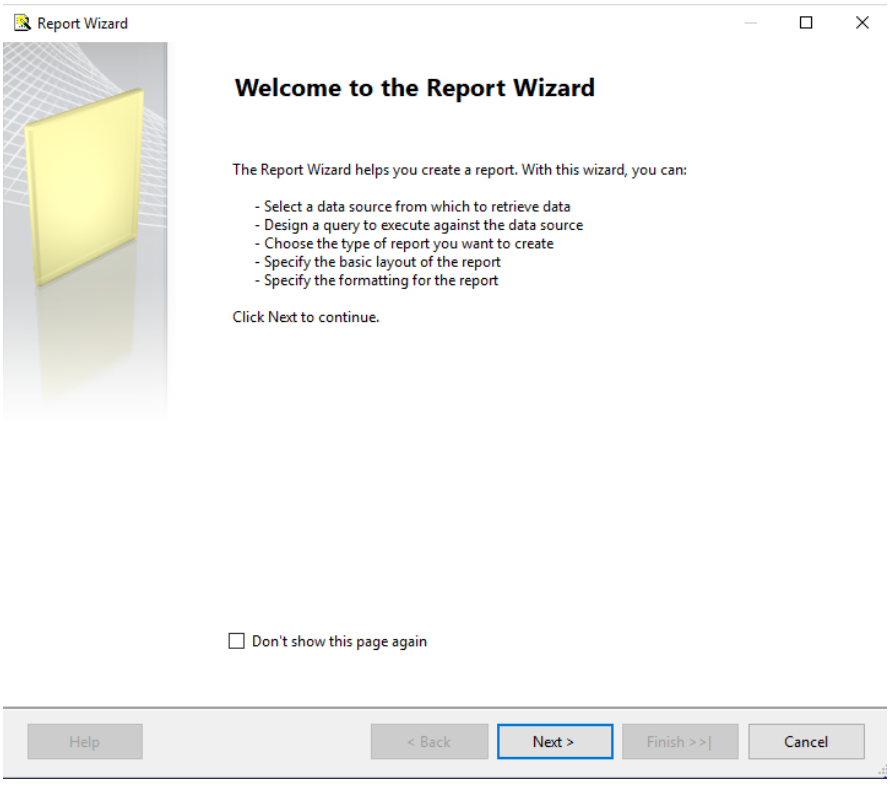
xiii. The cube can be seen in the SQL server analysis



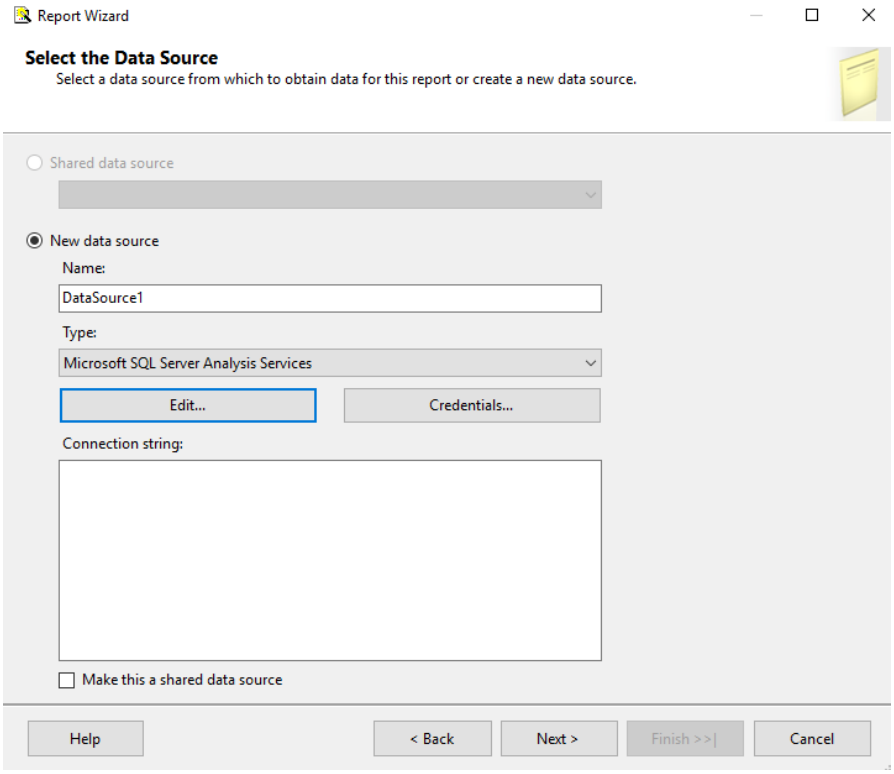
xiv. Now, to create the report - create a new project using Report server Project Wizard

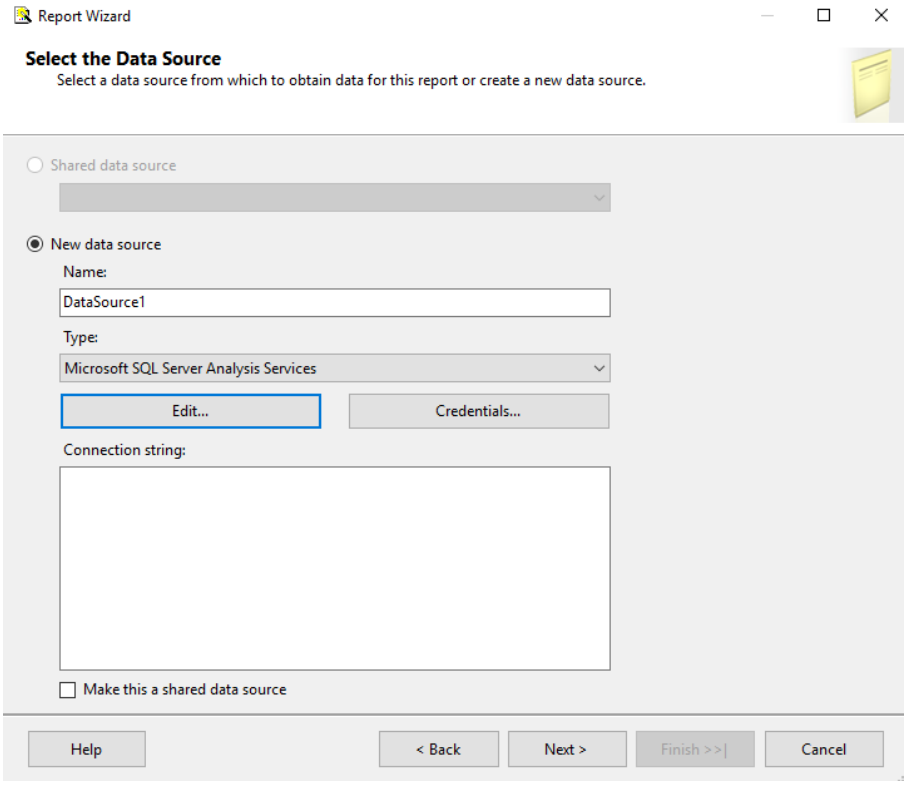


xv. Running SSRS over SSAS

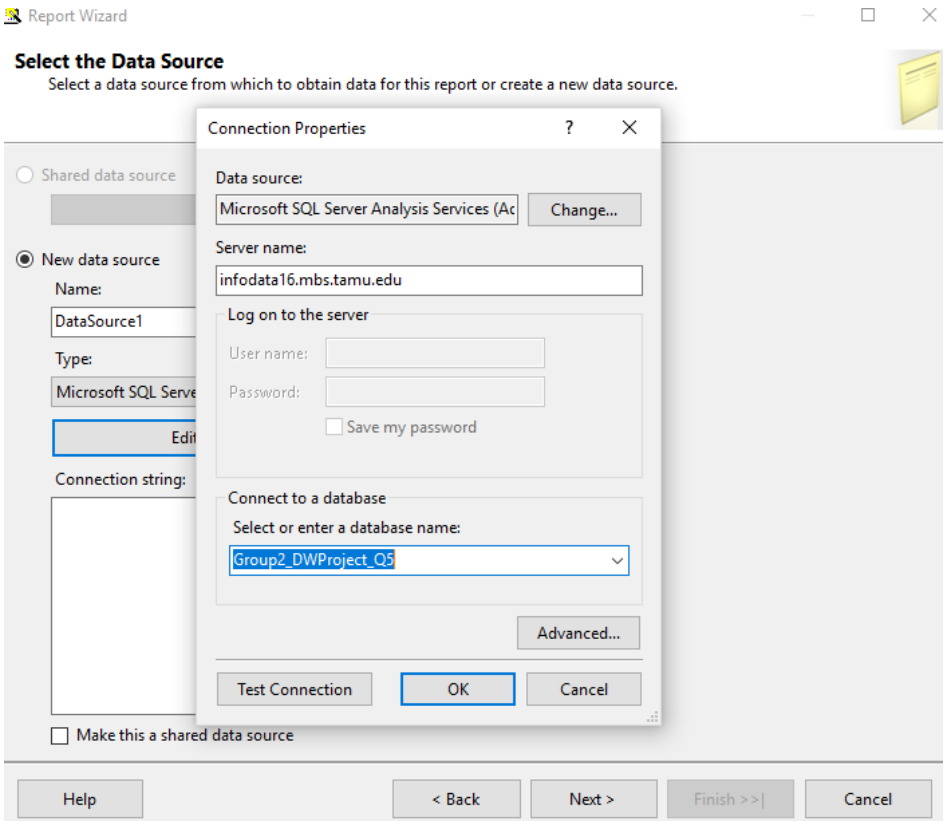


xvi. Create a new database connection

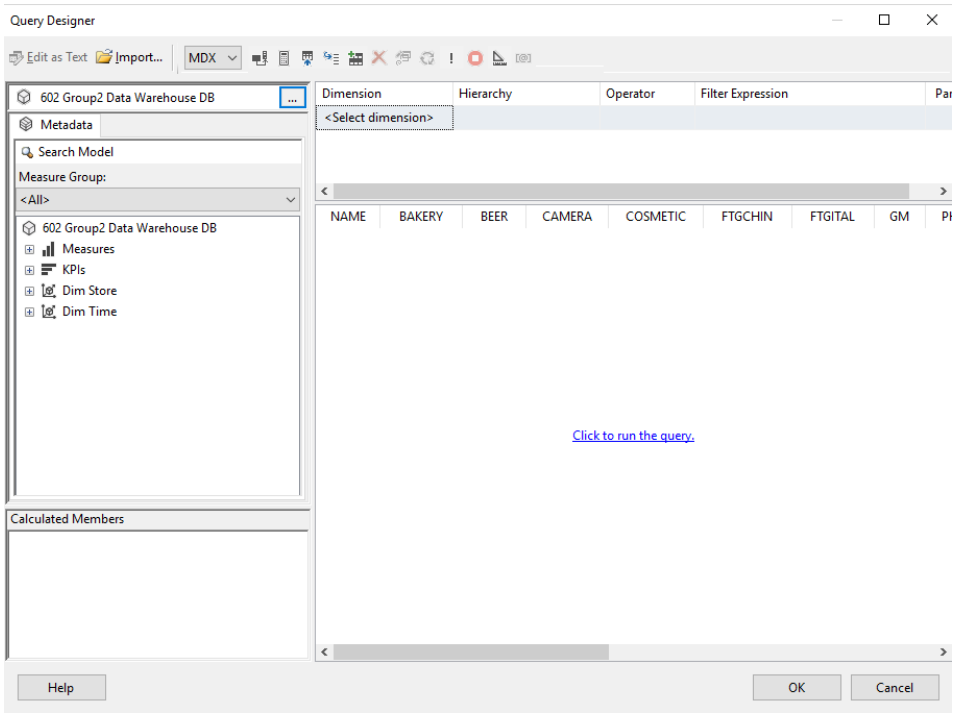


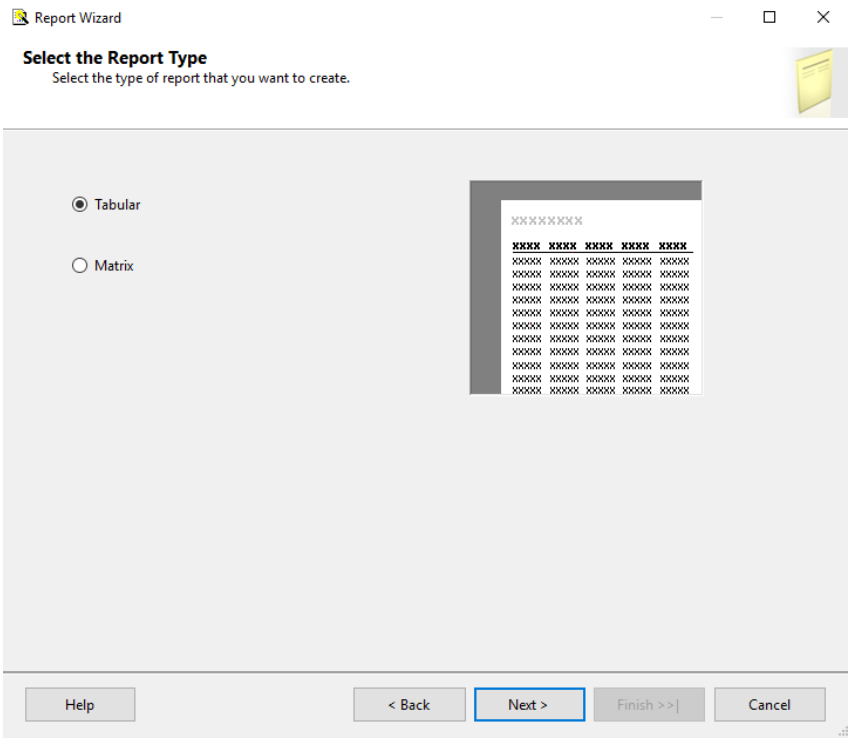


xvii. Provide the server name, credentials and the cube created using SSAS

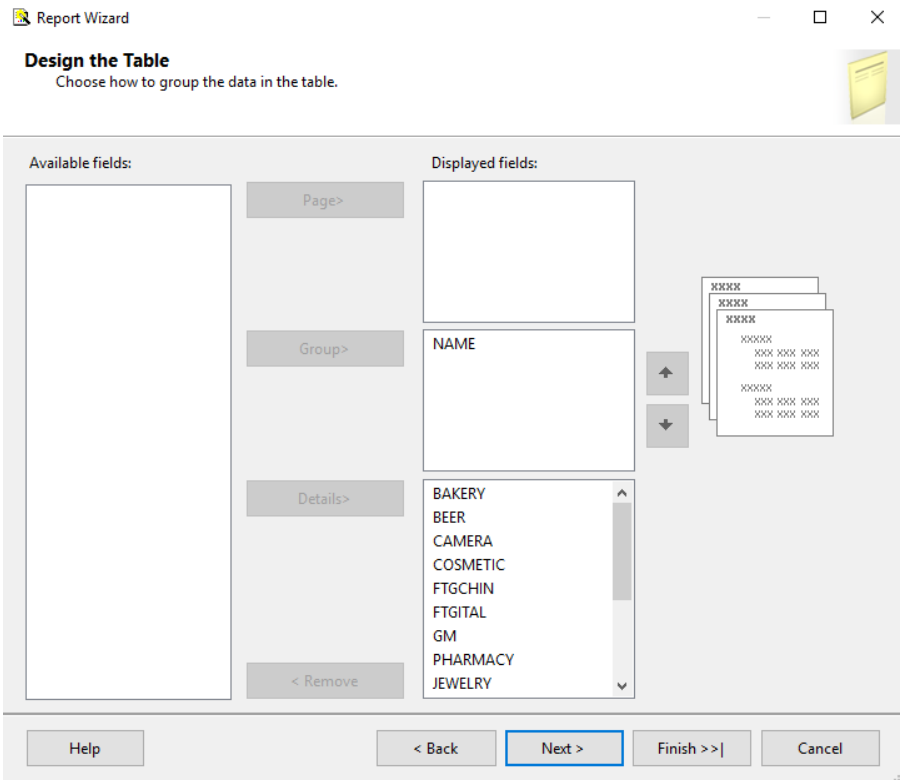


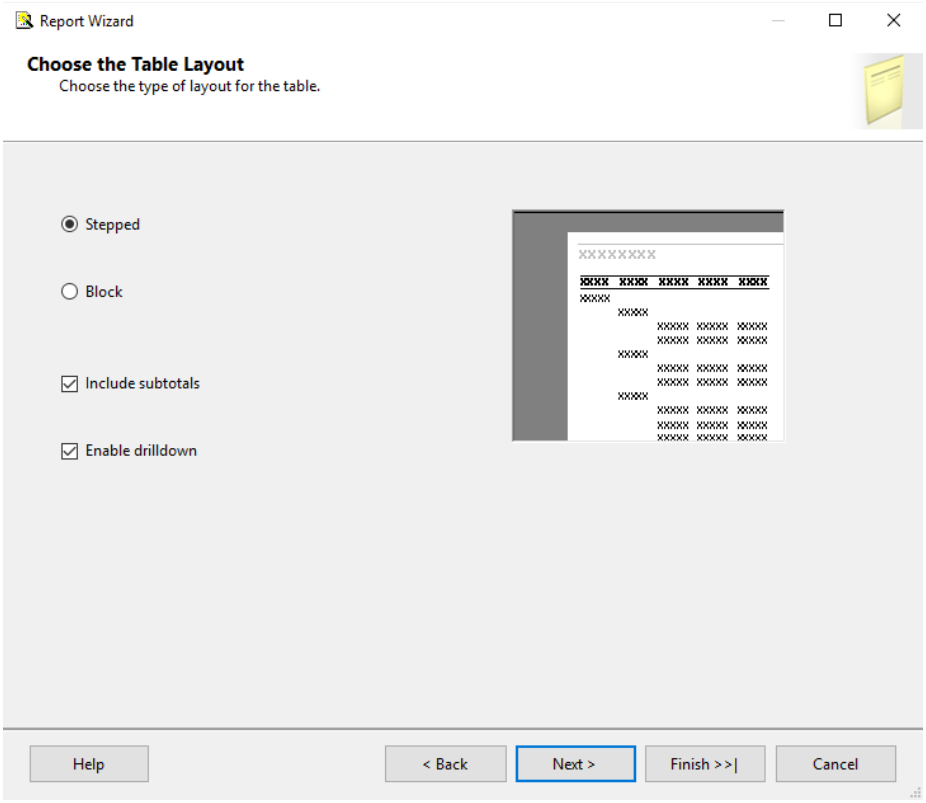
xviii. Create the query using query builder





xix. Aggregate the sales of categories by the stores

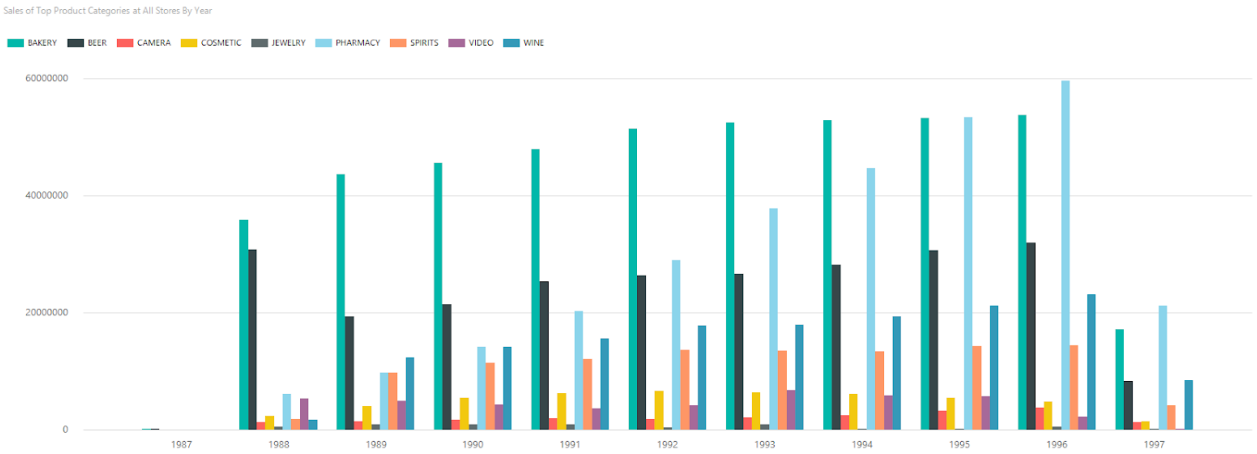




The report is created as follows:

Report3

Year	BAKERY	BEER	CAMERA	COSMETIC	FTGCHIN	FTGITAL	GM	HABA	SSDELICP	JEWELRY	MISCSCP	PHARMACY	PHOTOFIN	SPIRITS	VIDEO
⊞ 1987	4592.18	13233.77	0	0	0	0	17805.96	0	0	0	0	0	0	0	0
⊞ 1988	35902228.79	30791954.52	1308914.52	2376852.77	0	0	143990283.21	10726805.58	0	529527.54	-272065.28	6149591.21	192104.93	1772101.93	5274855.04
⊞ 1989	43623138.07	19390744.59	1426131.52	4065885.86	0	0	141160377.12	46995746.08	0	915778.91	-1166505.31	9697013.06	4598917.52	9775625.7	4884107.95
⊞ 1990	45616418.33	21454041	1646892.03	5449906.45	0	0	145078828.06	53662843.98	0	935182.47	-1068666.52	14205189.02	5909884.11	11478785.25	4279622.72
⊞ 1991	47937959.64	25290716.44	1944852.46	6258590.35	0	0	157052008.92	63993931.49	0	886635.85	-354199.2	20294706.76	6024053.65	12117664.82	3608523.18
⊞ 1992	51397676.65	26415751.76	1775624.1	6675773.64	0	0	150755392.83	76131693.69	0	396785.52	-579510.86	28993957.37	6440243.47	13681351.18	4209510.38
⊞ 1993	52449353.18	26682969.12	2046193.16	6334229.74	0	0	104582664.63	118299591.98	0	885360.54	-759512.509999999	37754399.63	7158058.2	13496662.63	6764598.05
⊞ 1994	52898949.14	28136337.9	2525569.54	6057513.87	0	0	100970945.15	115647903.99	0	133177.97	-916782.12	44727022.2	4817597.69	13356300.05	5856444.97
⊞ 1995	53242509.95	30588118.86	3314592.43	5497140.7	0	0	97628381.27	118314299.92	0	40754.82	-278006.87	53401824.91	6149817.2	14304542.71	5731237.77
⊞ 1996	53771986.41	31971812.09	3768447.4	4789369.41	0	0	98602420.65	119745685.59	0	566183.8	-1037607.27	59623839.71	5164087.98	14420740.22	2184540.29
⊞ 1997	17166455.96	8303666.45	1319274.74	1449456.01	0	0	32058954.85	42527252.37	0	8.7	-1214289.6	21191821.74	1333329.27	4103756.4	1449.29



Conclusion:

Here, we can compare the sales of product categories across all stores for a given year from 1987 to 1997. We can infer which categories contributed the most and which categories had the least contribution. From this DFF can look to improve the marketing strategy for the categories which have low sales or even look to stop selling them and change it with something else.

REFERENCES

[1] https://www.voicendata.com/retail-industry-needs-data-warehousing-analytics/
[2] https://www.chicagobooth.edu/research/kilts/datasets/dominicks
[3] https://datafloq.com/read/6-challenges-big-data-analytics-for-retail/4900
[4] https://www.reltio.com/blog/2016-7-challenges-in-leveraging-big-data-in-retail/
[5] *https://www.slideshare.net/sunitasahu101/dimensional-modeling-53600268*
[6] *http://www.dkms.com/papers/dmerdw.pdf*
[7] *http://www.information-management.com/issues/20000501/2184-1.html*
[8]http://intelligententerprise.informationweek.com/channels/information_management/showArticle.jhtml;jsessionid=DPHDNU4MGRCEHQE1GHPSKHWATMY32JVN?articleID=217700810