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

Rijoe F Gowra
232003187
rijoe_g@tamu.edu

Harsha Narendra Kulkarni
431000220
harsha-kulkarni@tamu.edu

Yash Katariya
530005734
yashkatariya@tamu.edu

Sujay Manjunath
830007134
sujaymanjunath@tamu.edu

Introduction

The project aims to develop a data warehouse to house data obtained on a retail store chain called Dominick's Finer Foods (DFF) to enable complex data analysis in support of impactful business decision-making for DFF.

Dominick's Finer Foods is a Chicago-based retail store chain with over 100 branches. It is the second-largest supermarket operation in Chicago. Many of its outlets combine food stores and drug stores and contain floral sections, in-store cafes, and extended produce sections among others.

DFF partnered with the Chicago Booth School of Business to conduct store-level research into pricing and shelf management. The data utilized for this report and the project has been collected during this 5-year research collaboration.

This report will detail our understanding of the data, the business domain associated with the data, and an evidence-based analysis of the data. The section following this brief introduction aims to provide the reader with a comprehensive understanding of the dataset. We will begin with a high-level description of the data which will necessarily include an exploration of metadata followed by an in-depth examination using snapshots of actual data, ERDs, and charts as and when needed.

The next section focuses on our understanding of the business domain of the data. We shall begin by exploring the objectives of the business domain, describing the metrics used to measure performance, domain concepts as they relate to understanding DFF's data, and how all of these topics may be pieced together to inform our analysis.

Finally, from our understanding of the data and the business domain, we will detail a set of important business questions that we may ask of the data to derive answers that can provide guidance when making business decisions. We will prioritize these questions in the order of their importance by making considerations such as their impact on business objectives and their relevance to the most pressing problems facing DFF if any. A rationale will be provided for why each of these questions has been included in our analysis. Each question will be followed by evidence from the data to support their validity if need be. Some of the problems that are faced by DFF are inventory management, pricing strategies, and efficiently planned store locations. Inventory management includes making sure most SKUs are available at majority store locations and pricing at these stores is done in a way that attracts customers.

Examination of the Data Set

Overview of the Dataset

The data set contains scanner-level data collected at Dominick's finer foods over a period of about 10 years beginning from 1988 to 1997. It pertains to 25 different product categories throughout all stores of the chain and covers sales information on more than 3,500 UPCs.

The data set's files can be categorized into two types: general and category-specific files. General files contain information relating to all product categories in this data set.

The following files can be found in this data set:

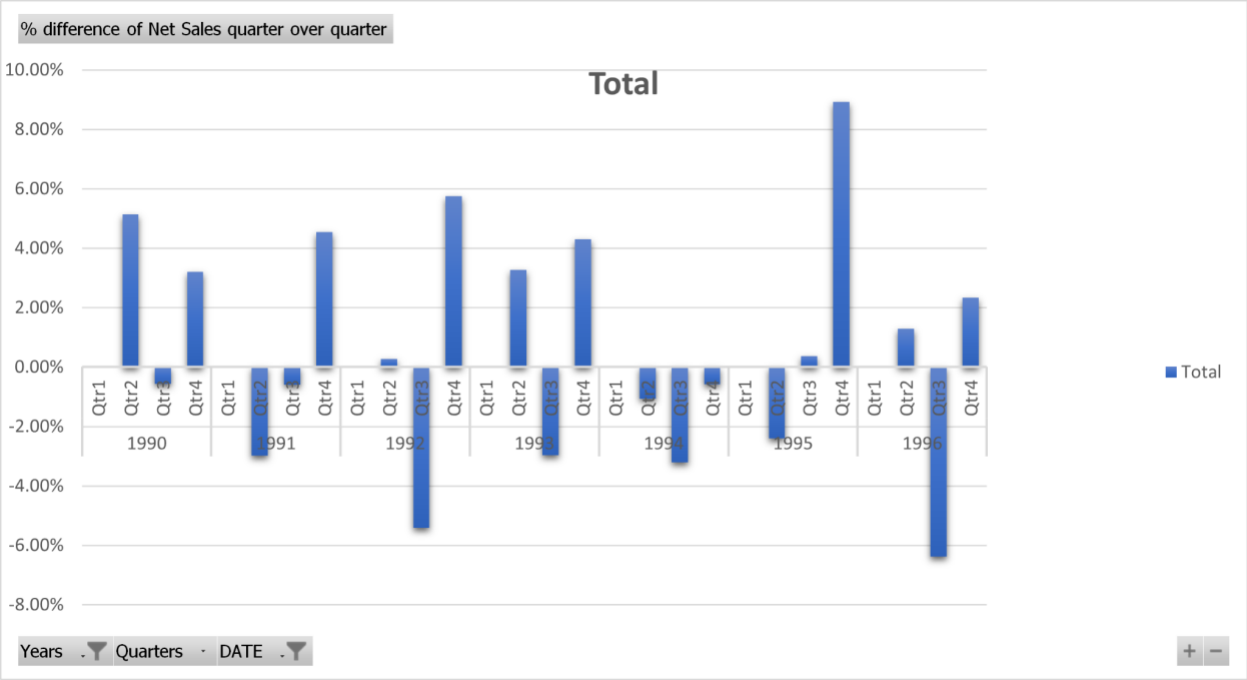
- Customer count file (general): It contains information on store traffic, sales by DFF-defined departments, and coupon usage by DFF-defined departments. Each record corresponds to a daily observation at a particular store.
- Demographics file (general): This file contains store-specific demographic data. Census data for the Chicago metropolitan area has been processed to generate demographic profiles for each of the stores.
- UPC files (category-specific): Each file contains descriptions of UPCs within a particular category. The name convention used for the files follows UPCXXX, where XXX is the three-letter acronym for the product category.
- Movement files (category-specific): These files contain weekly sales data for each UPC in a category at the store level.

Business Questions and their substantials and explanation

The following are the business questions that we have come up with to define DFF's business -

1. What is the percentage change in net sales quarter over quarter for the period 1990-96? (Implementing)

Years and Quarters	% difference of Net Sales quarter over quarter
1990	
Qtr1	
Qtr2	5.13%
Qtr3	-0.56%
Qtr4	3.20%
1991	
Qtr1	
Qtr2	-2.98%
Qtr3	-0.59%
Qtr4	4.54%
1992	
Qtr1	
Qtr2	0.27%
Qtr3	-5.41%
Qtr4	5.75%
1993	
Qtr1	
Qtr2	3.27%
Qtr3	-2.96%
Qtr4	4.30%
1994	
Qtr1	
Qtr2	-1.07%
Qtr3	-3.20%
Qtr4	-0.59%
1995	
Qtr1	
Qtr2	-2.41%
Qtr3	0.38%
Qtr4	8.92%
1996	
Qtr1	
Qtr2	1.29%
Qtr3	-6.39%
Qtr4	2.33%



Sales revenue is an important business metric that all businesses use to measure performance. In DFF's case, the sales revenues (column 'Net Sales') for the years from 1990 to 1966 have been obtained by subtracting/adding the coupon total as appropriate. Most companies use quarterly metrics for various reasons including to allow for comparison with competitors and to allow for setting and tracking goals among others. In order to measure how a metric like sales revenue is changing over the course of a fiscal year, the percentage difference between quarterly sales as compared to the previous year is a useful measure.

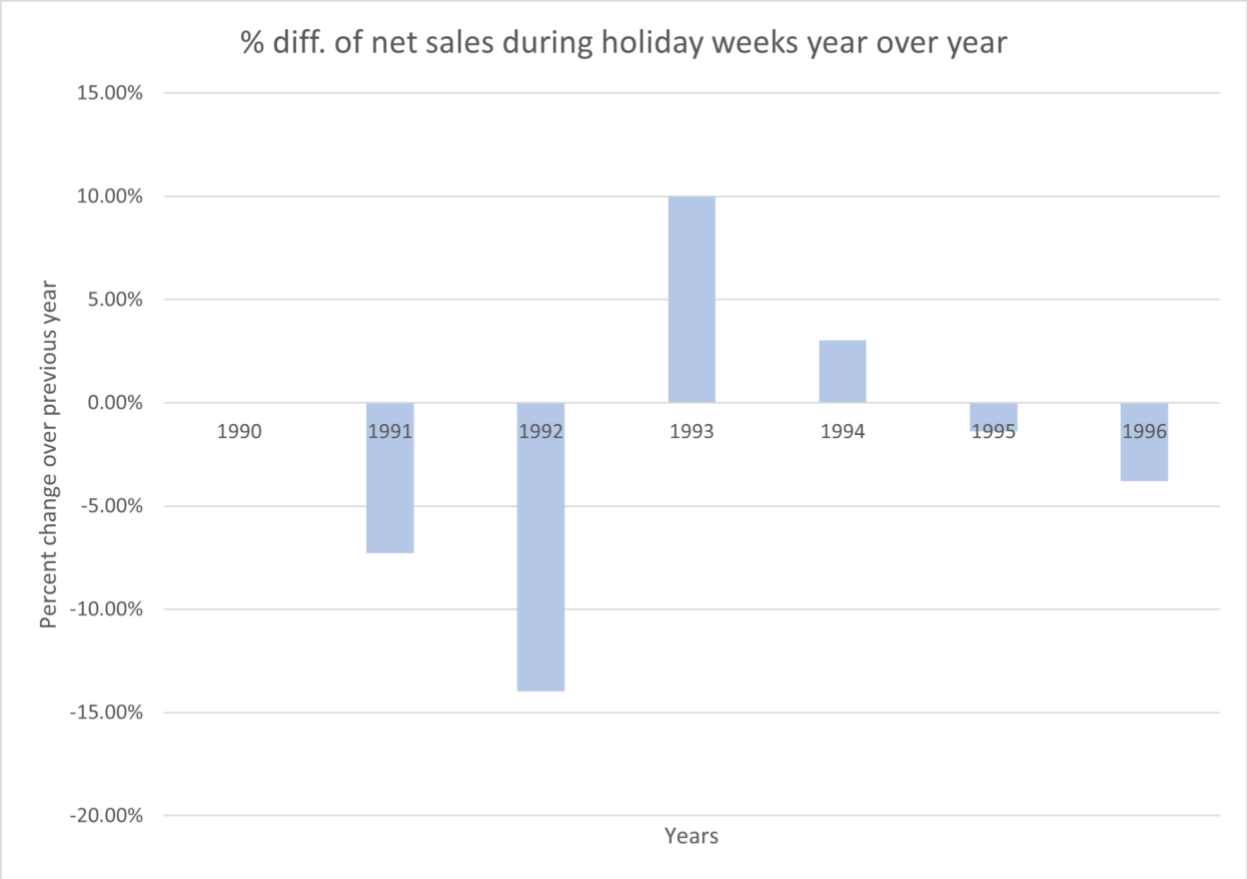
2. List the top ten stores by net sales for the period from 1990 to 1997 and their customer count.

Store	Top 10 stores by Net Sales from 1990-96	Customer Count Total from 1990-96
32	$194,314,787	9866760
102	$228,778,522	9392994
109	$244,601,818	7747393
112	$202,080,728	8292442
121	$199,886,945	8068380
122	$226,000,980	8964231
126	$219,591,627	8608117
128	$201,817,010	9402378
132	$205,630,721	8839808
137	$220,703,277	6898410
Grand Total	**$2,143,406,414**	**86080913**

An executive examining the sales numbers of DFF might want to know the top ten stores by sales revenue. After that he/she might want to know the customer count for these ten stores to get a feel for how sales numbers relate to customer count. A pictorial description of the numbers as shown above can make it easier to grasp the relationship between these two numbers. It can be observed from the graph above that there seems to be a significant relationship between sales figures and customer count for many of those stores but there are also stores that have done exceptionally well despite relatively less customer count and vice versa.

3. What is the percentage change in net sales for all holiday weeks year over year from 1990-96? (Implementing)

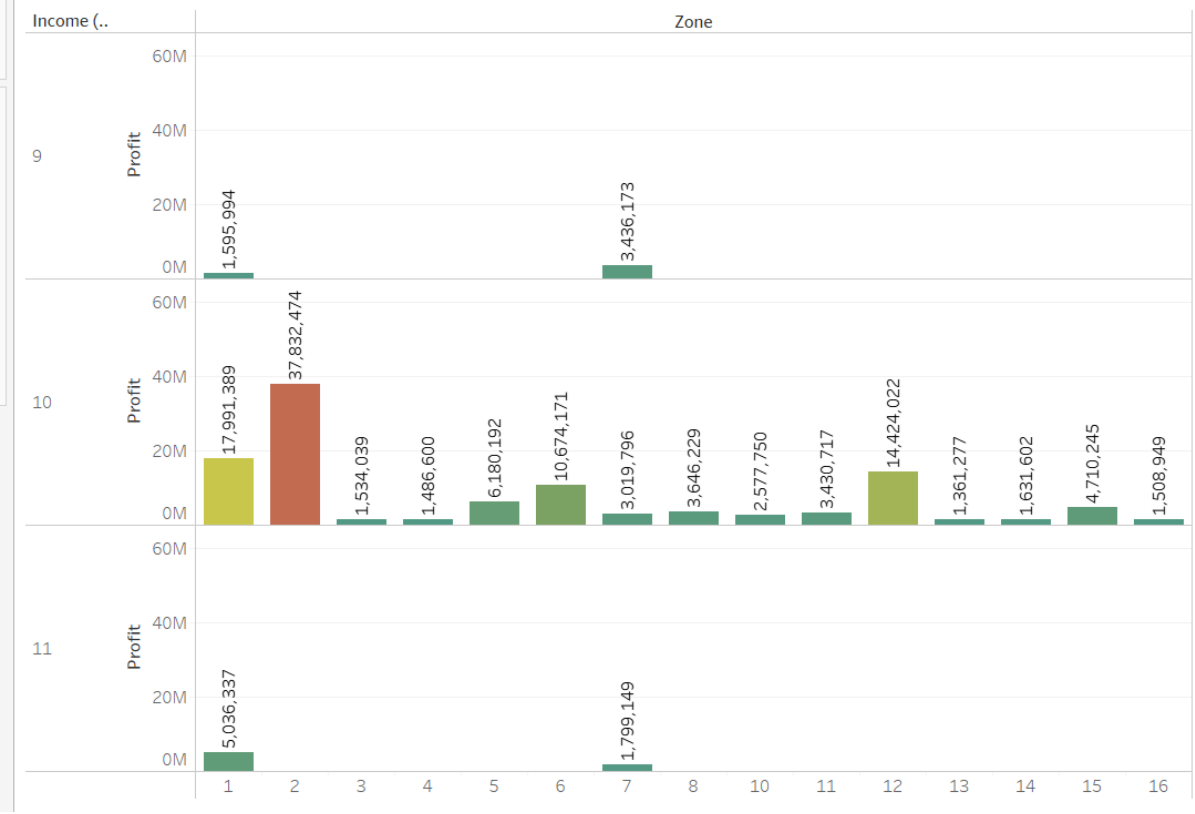
Years	% diff. of net sales during holiday weeks year over year
1990	
1991	-7.27%
1992	-13.96%
1993	9.97%
1994	3.04%
1995	-1.39%
1996	-3.80%



Holidays are important periods for retail businesses like DFF due to their disproportionately high contribution to sales. As such, it is important for businesses to monitor their sales figures during these times to inform decision-making. Analysis of these numbers may begin with a simple description of total sales over the course of a period of interest. The next phase of analysis may include examining how these figures have changed over the course of that period. Percentage change is a very useful measure of how a total has changed over a period. The percentage change of total sales during holiday periods year over year can be used to gauge performance over the

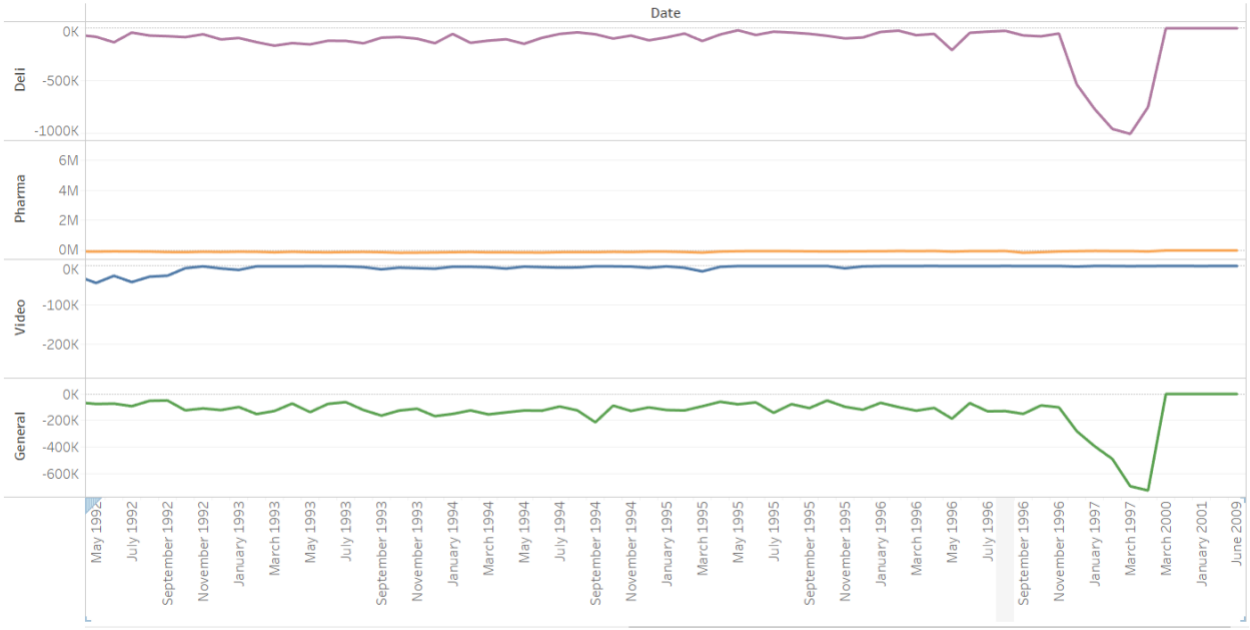
course of a period which in this case is from 1990-96. Observing that the performance has dipped or increased, the analyst can then further dig into the reasons for why that might be.

4. Which zone generated maximum profits, and what were the income ranges of the people in these zone markets? (Implementing)



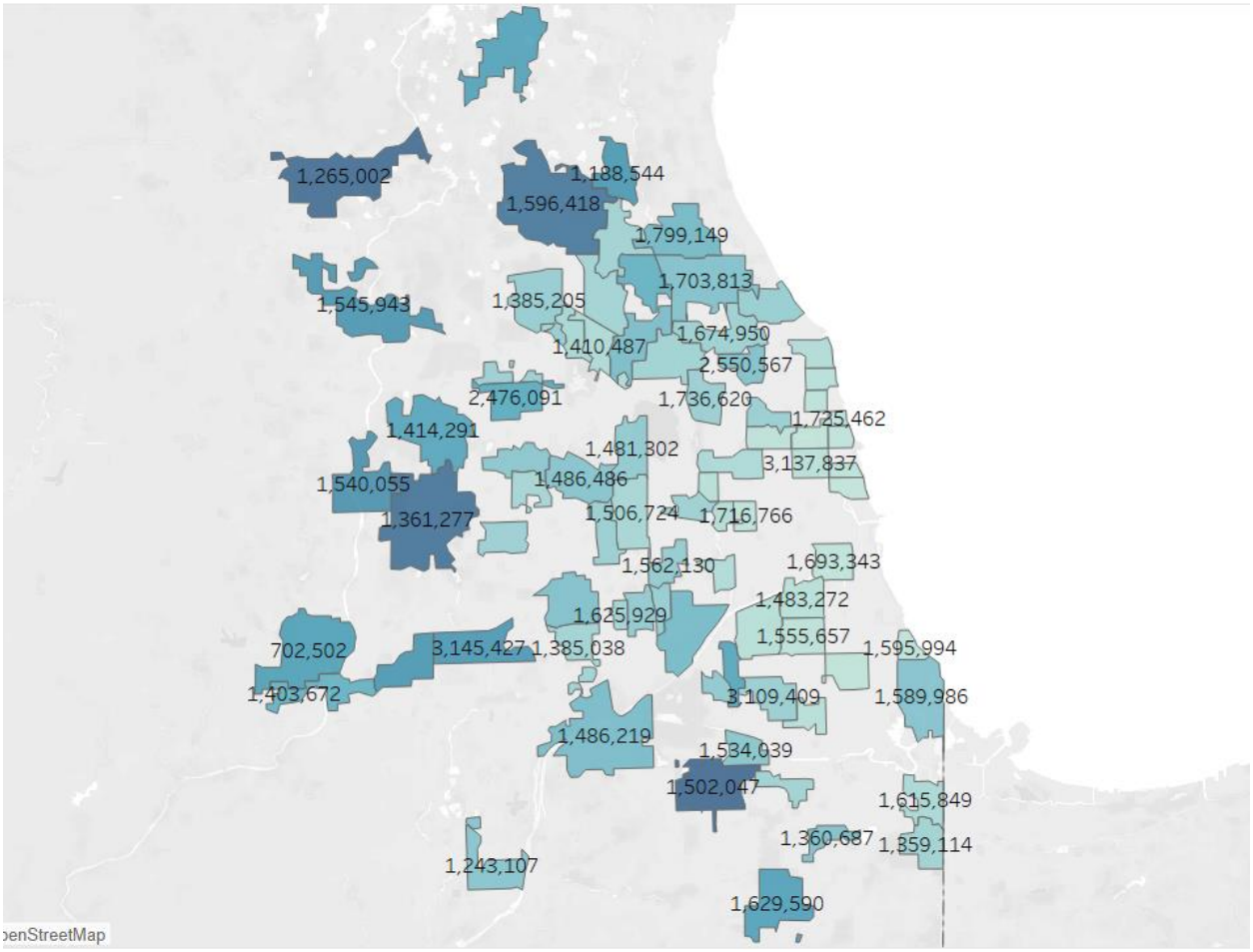
It is important to know which geographic regions are generating the most profits and the range of household incomes that are generating these profits. This analysis shows that maximum profits are generated in the zone and particularly in the household having an average income of 10-11k. This segment of the target market should be the company's focus and should make deals, and products that suit this segment.

5. Which category of coupons reduced profits? (Implementing)



Coupons and offers lead to lower profits. There is a substantial chance that the pricing strategy of the company could go wrong if the promotions aren't effectively applied. The above analysis shows the major category coupons that are lowering the profits of DFF. Coupons can be given during the holiday season to boost sales of gift category products and not necessarily on other items to increase profits.

6. Which regions are densely populated and what are the profits generated by these regions? (Implementing)

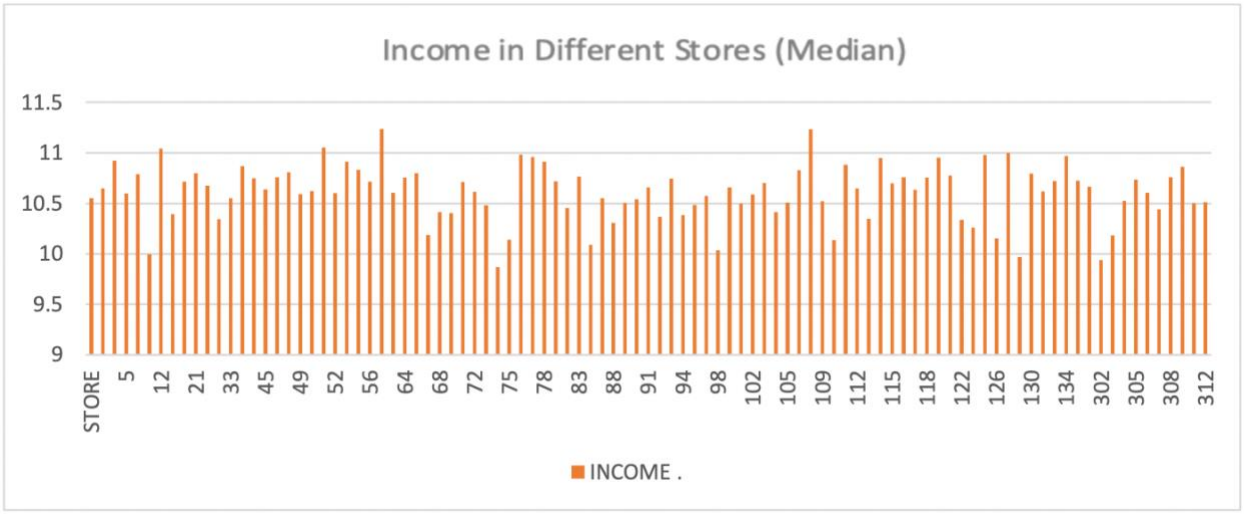


The inventory distribution is largely dependent on the population of the region. Knowing the density of the population is essential to understand shopping behavior and inventory stocking. Here the largely populated regions are of a darker shade. The numbers in these regions are the profits generated in that region. The profits generated are non-uniformly distributed and do not have a dependence on density. Hence density of the region may not be considered when generating strategies for increasing profits.

7. Which store saw the most number of customers over the course of 7 days?

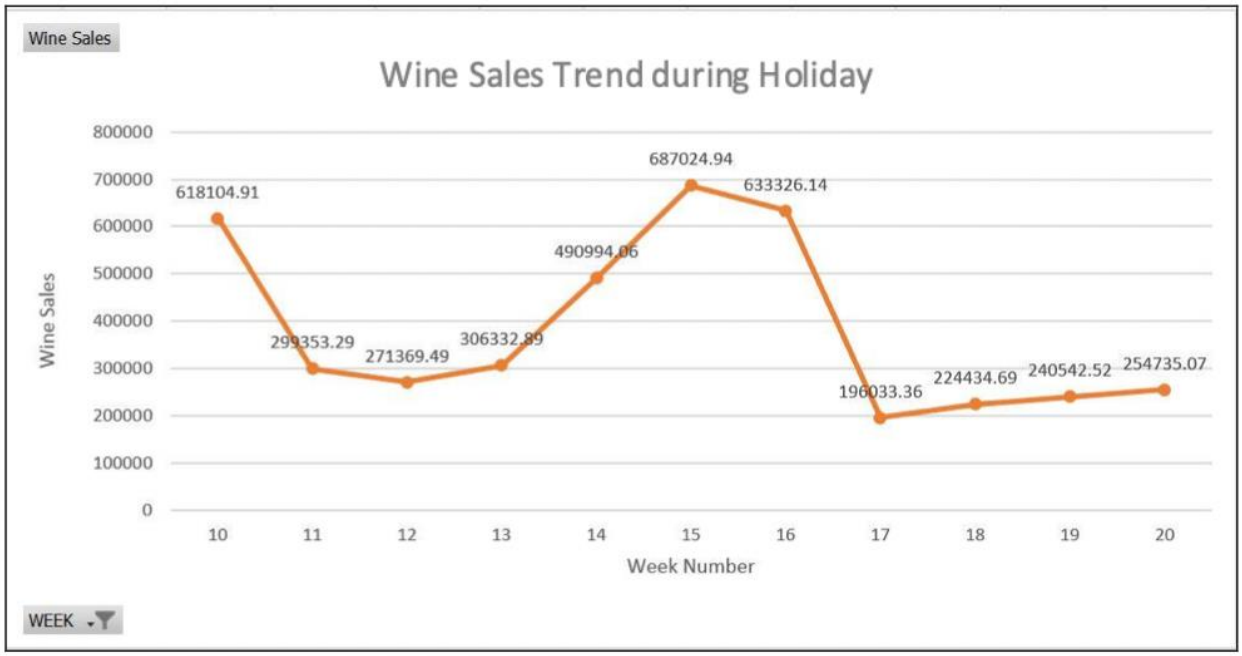
A location with heavy traffic could be a lucrative prospect for advertising. In order to strategically advertise or market a new product to increase sales, the busiest store will be the best location and for that reason, analyzing the store's weekly customer count could provide valuable insights for decision-making.

8. What retailer is located in the neighborhood with the greatest average salary?



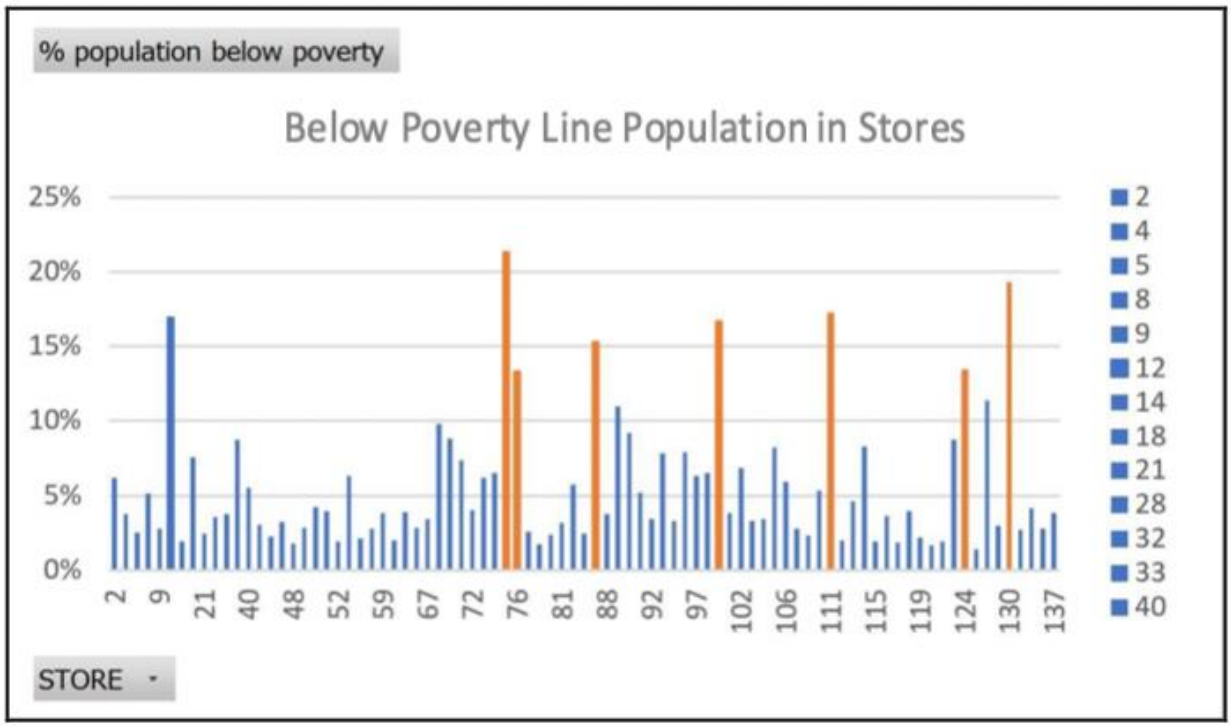
It is very much essential to know where the store is making much profit. And we know that most of the profits are made in a wealthy district where the median income is very high. Using this data we can then allocate more talented and skillful employees to such regions so as to get better sales which will, in turn, lead to increased profits.

9. How did the sale of wine change during the holiday season?



Holidays are a major time for the sale of goods. Businesses observe a significant rise in sales during festivals/holidays. In the data provided, wine saw a drastic change in sales during Christmas(i.e. during weeks 13-17). DFF will be able to discover all the goods that typically see seasonal sales growth thanks to such a study. Increased sales call for more inventory, and better shelf management techniques have to be adopted, which is one of the biggest worries of businesses.

10. Which store is most preferred by people in the low-income groups?



Analyzing the information as to which store is preferred by wealthy people and which store is preferred by low-income groups will aid DFF to distribute the items catered to that particular group of people. In this case, low-income group people usually preferred to buy non-luxury goods.

Business Questions

The 5 business questions that are most important for DFF's business and being implemented are

BQ1.: What is the percentage change in net sales quarter over quarter for the period 1990-96?
BQ3.: What is the percentage change in net sales for all holiday weeks year over year from 1990-96?
BQ4.: Which zone generated maximum profits, and what were the income ranges of the people in these zone markets?
BQ5.: Which category of coupons reduced profits?
BQ6.: Which regions are densely populated and what are the profits generated by these regions?

Independent Data Marts design using Kimball's approach

Kimball's Methodology for DFF Data Warehouse Design:

Step 1: Requirement Analysis and Prioritization

The final five business queries most crucial to our company's operations have been determined. These are what they are:

- What is the percentage change in net sales quarter over quarter for the period 1990-96?
- What is the percentage change in net sales for all holiday weeks year over year from 1990-96?
- Which zone generated maximum profits, and what income ranges of the people in these zone markets?
- Which category of coupons reduced profits?
- Which regions are densely populated, and what are the profits generated by these regions?

Step 2: Define Dimensions

By developing the dimension tables below, we have provided answers to the following business queries:

- Time
- Store
- Product
- Coupon

Step 3: Define Data Marts

We are developing two data marts to answer our five business questions:

- Sales
- Profits

Step 4: Build Matrix

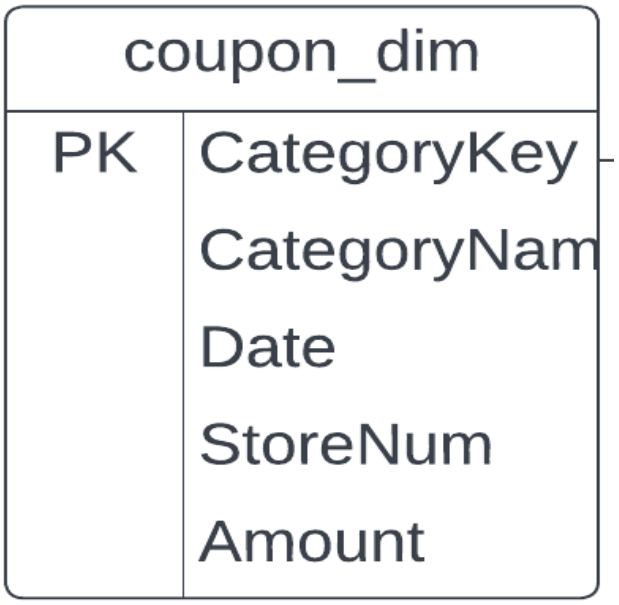
		Dimensions			
		store	time	product	coupon
Datamarts	sales	Y	Y	Y	
	profits	Y	Y	Y	Y

Step 5: Build Dimension Tables

We would generate the following dimension tables for our dimensional model. The importance of each attribute's matching properties has also been explained.

coupon_dim:

The coupon dimension table contains information regarding coupons that can be used at the DFF.



The following thoroughly explains the several attributes of the coupon dimension table:

- **CategoryKey:** The Dominick's Finer Foods coupons are uniquely identified in each row of this dimension table by this surrogate key. The fact table in the Data Mart uses this field as a foreign essential reference.
- **CategoryName:** This field refers to the different types of valid coupons at DFF.
- **Date**: The date at which the coupon was used
- **StoreNum**: The stored number where the coupon was used
- **Amount:** This field refers to the different amounts that several coupons are available at DFF.

product_dim :

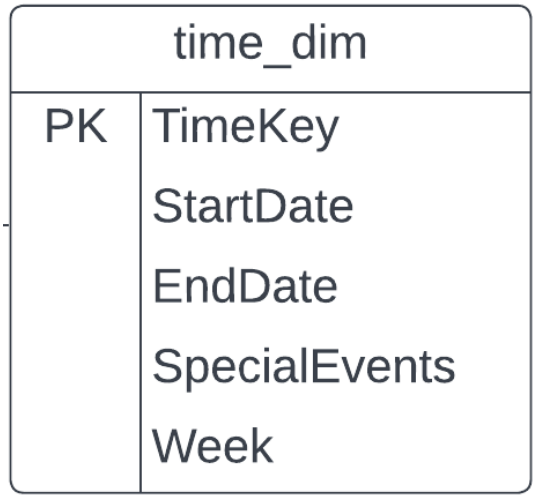
The specifics of each product that is offered in each of Dominick's Finer Foods' locations will be listed in the table called "product_dim" (DFF). The table contains information about

The following. It thoroughly explains the several attributes of the product dimension table:

- **ProductID:** The Dominick's Finer Foods products are uniquely identified in each row of this dimension table by this surrogate key. The fact table in the Data Mart uses this field as a foreign key reference.
- **UPC:** Includes each product's UPC number. The dataset's Dominick's Manual states that the manufacturer is identified by the first three digits of the UPC number, while the last five numbers identify the product.
- **Description:** Refers to the details provided in the product descriptions found in DFF retail locations.
- **Size:** This field refers to the details of the quantity/measurement of the products at DFF.
- **SKU:** This field contains the data about the stock-keeping unit required for inventory management at DFF.
- **RetailPrice**: The price at which the product is sold
- **Week**: The week number of the sale
- **Profit**: The profit that was generated from the sale of the product

time_dim:

This table will assist in giving our fact table information a time dimension. The properties of the dimension table are the granularity of the dimension. To enable fine analysis that may be summed up to a year's duration, we keep the data at a daily granular level.



The following provides a thorough explanation of the several attributes of the time dimension table:

- **TimeKey:** The Dominick's Finer Foods time dimension is uniquely identified in each row of this dimension table by this surrogate key. The fact table in the Data Mart uses this field as a foreign key reference.

- **StartDate**: The first date of the week
- **EndDate**: The last date of the week
- **SpecialEvents**: Any event that occurred in this interval
- **Week**: Week number of the date

store_dim:

Each Dominick's Finer Foods store's unique information will be contained in the "store_dim" (DFF). To examine the dataset based on the locations of the stores, Dominick's Finer Food (DFF) stores' geographic information is provided in this table.

The following provides a thorough explanation of the several attributes of the store dimension table:

- **StoreID:** The Dominick's Finer Foods store locations are uniquely identified in each row of this dimension table by the surrogate key. The fact table in the Data Mart uses this field as a foreign key reference.
- **StoreNum:** Each DFF store is uniquely identified by this field. This field's information is taken directly from the DFF data set's Store table.
- **Density:** Refers to the details provided on the density of customer population for the DFF stores.
- **Zone:** This field relates to the geographic region that the stores are located. We get zone-based insights from aggregation based on this field.
- **City:** This field is used to specify the name of the city where the stores are situated.
- **Zip:** Each DFF store's zip or postal code is listed in this field.

Step 6: Build Fact Tables

We would require information from the Dominick's Finer Food (DFF) databases tables for Fact Tables. Two data marts would need to be established to respond properly to business inquiries. As stated below, there will be one fact table for each data mart.

Fact Table 1: sales_fact

This fact table will give us the data regarding the store-level sales of products based on the quantity and move on each day.

Dimensions: store_dim, time_dim, product_dim

Keys:

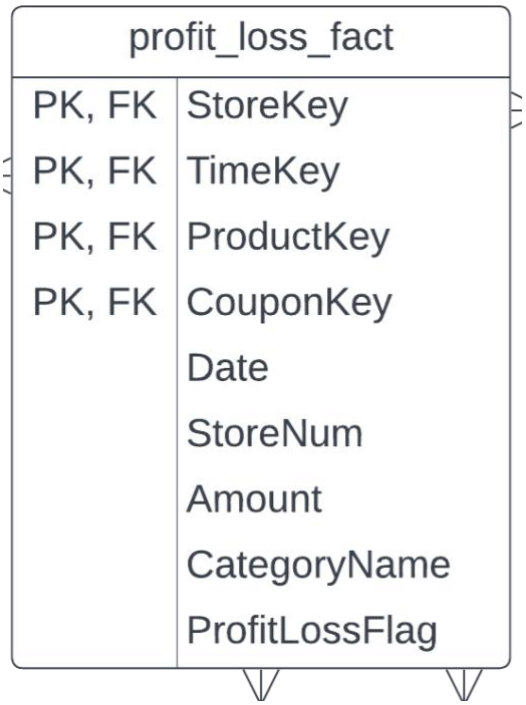
StoreID: Referenced foreign key ‘’store_id’ in ‘store_dim’
TimeKeyReferenced foreign key ‘’date’ in ‘time_dim’

Facts:

Quantity: Refers to the number of actual items (units) sold at the DFF
SpecialEvents: Any special event that occurred
TotalSales: Refers to the amount of sales made by selling the products at DFF

Fact Table 2: profit_loss_fact

This fact table contains the information on whether the sales made profit or loss at the DFF on store-level.



Dimensions: store_dim, time_dim, product_dim, coupon_dim

Keys :

CouponKey: Referenced foreign key ‘’coupon_type’ in ‘coupon_dim’

StoreKey: Referenced foreign key ‘’store_id’ in ‘store_dim’
ProductKey: Referenced foreign key ‘’product_id’ in ‘product_dim’
Timekey: Referenced foreign key ‘’date’ in ‘time_dim’

Facts :

Amount: Refers to the amount of sales made by selling the products at DFF

ProfitLossFlag: Flag set to describe whether the sale incurred profit or loss

Kimball's Matrix for Datamarts

		Dimensions			
		store	time	product	coupon
Datamarts	sales	Y	Y	Y	
	profits	Y	Y	Y	Y

Logical Design

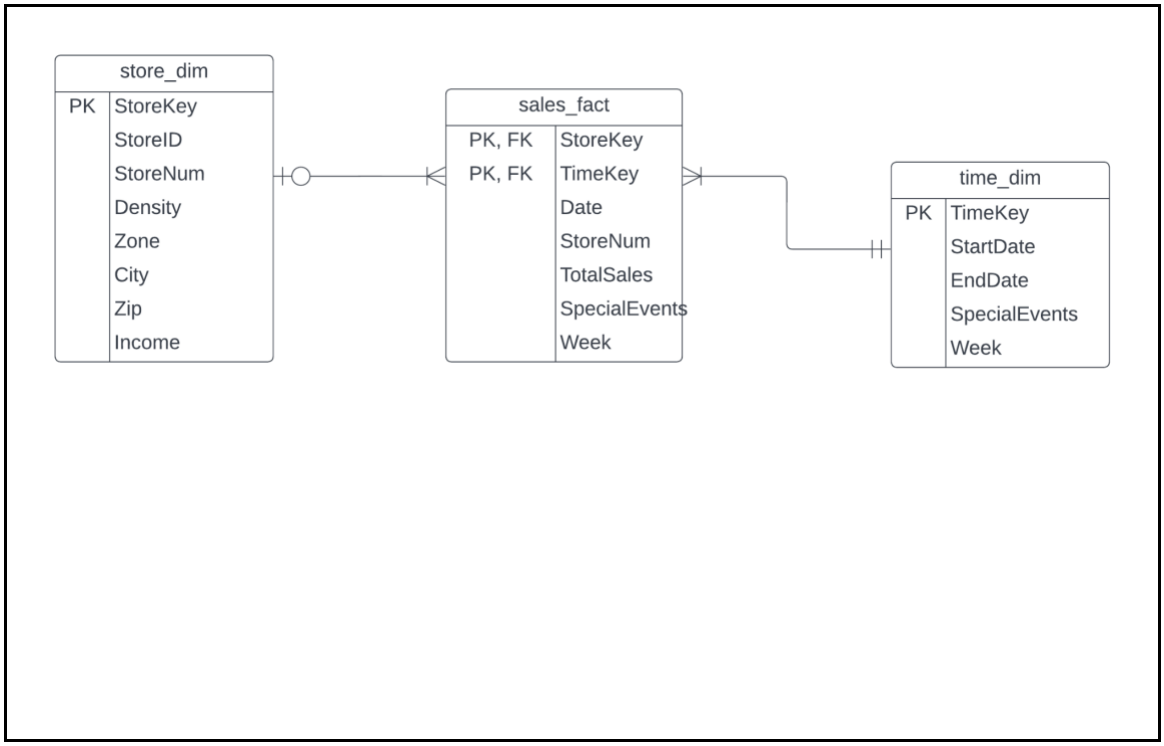


Image: Star schema for Sales

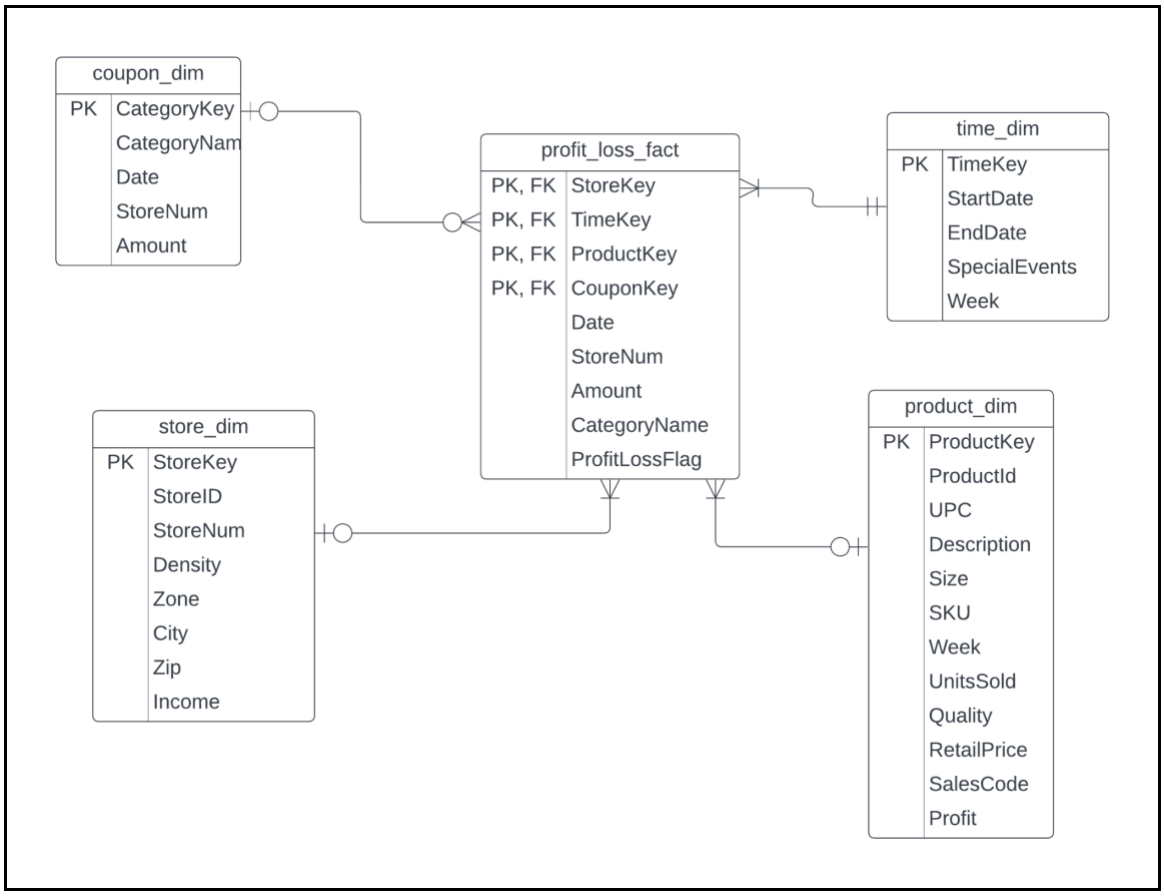


Image: Star schema for Profits

Data Cleaning and Integration

Data Quality

Data must be of usable quality before it is used for reporting purposes. The data quality checks can be done before the data is brought into the data warehousing platform or while performing the ETL process. This clean data will be used in creating fact and dimension tables. For this project, we have performed cleansing before loading data into the fact and dimension tables.

The different steps that were followed to clean data are -

Different formats of data source
The DFF dataset has data stored in different formats such as csv files and text files. Integrating the data files in a single format and loading it to the staging area was one of the cleansing activities completed as part of this project.

Removing null values

There are certain fields in the dataset that contain null values or invalid values. Filtering these records while loading data from staging tables to dimension and fact tables was performed. For example, the store numbers cannot be null or empty.

Removing special characters
There are certain special characters present in the dataset such as '~' that make the data invalid. The transformations to remove these special characters are done in ETL process.

Data type conversion
While loading data into the staging area all the data is loaded in string format, but to perform transformation and calculations the data types are cast and converted while loading data into dimension and fact tables. Some of the most common conversions are amounts that are converted into floats and decimals.

Removing empty values
Certain records in the dataset contain fields that have no values or have ''. Such records are being filtered while loading data into dimension and fact tables.

ETL Plan

The ETL Development Plan lays out the method for loading data into the data warehouse based on the prior reports created.

Steps involved in the development plan :

- Determination of target data
- Determination of source data
- Mapping tables for staging and data mart loads
- Comprehensive data extraction rules
- Data transformation and cleansing rules
- Plan for aggregate tables
- Procedures for data extraction and loading
    - ETL for dimension tables
    - ETL for fact tables

Determination of target data

In the Data Warehouse section below, our suggested dimensional model has four dimension tables with two fact definitions.

Dimension Tables

Dimension: Coupon		
DW Target Table	DW Target Column	Target Datatype
DW-dbo.coupon_dim	CategoryKey	int
	CategoryName	nvarchar
	Date	date
	StoreNum	nvarchar
	Amount	nvarchar

Dimension : Product		
DW Target Table	DW Target Column	Target Datatype
DW-dbo.product_dim	ProductKey	int
	ProductId	nvarchar
	upc	nvarchar
	description	nvarchar
	size	nvarchar
	SKU	nvarchar
	week	int
	UnitsSold	int
	Quality	int

	RetailPrice	decimal
	SalesCode	nvarchar
	Profit	int

Dimension: Time		
DW Target Table	DW Target Column	Target Datatype
DW-dbo.time_dim	TimeKey	int
	StartDate	date
	EndDate	date
	SpecialEvents	varchar
	week	nvarchar

Dimension: Store		
DW Target Table	DW Target Column	Target Datatype
DW-dbo.store_dim	StoreKey	int
	StoreID	nvarchar
	StoreNum	nvarchar
	Density	nvarchar
	Income	numeric
	Zone	int

	City	nvarchar
	Zip	bigint

Fact Tables :

Fact: Sales		
DW Target Table	DW Target Column	Target Datatype
DW-dbo.sales_fact	StoreKey	int
	TimeKey	int
	Date	date
	StoreNum	varchar
	TotalSales	decimal
	SpecialEvents	varchar
	Week	varchar

Fact : ProfitLoss		
DW Target Table	DW Target Column	Target Datatype
DW-dbo.profit_loss_fact	StoreKey	int
	CategoryKey	int
	TimeKey	int
	ProductKey	int

	StoreNum	varchar
	Date	date
	Amount	varchar
	CategoryName	varchar
	ProfitLossFlag	varchar

Determination of Source Data

Source data for above-designated schema come from Dominick's FF data files CCount.csv, Demo.csv, movement and UPC files.

Mapping tables for staging and data mart loads

a. From source to staging

Source File Name	Source File Attributes	Staging Area Table Name	Staging Area Table Attributes	Mapping / Transformation Function
Demo.csv	NAME	dbo.DEMO	NAME	Load directly from the source
	CITY		CITY	
	ZIP		ZIP	
	STORE		STORE	
CCount.csv	STORE	dbo.CCount	STORE	Load directly from the source
	DATE		DATE	
	DAY		DAY	
	PRODUCT_NAMES		PRODUCT_NAMES	

	CATEGORY_COUPONS		CATEGORY_COUPONS	
	WEEK		WEEK	
Movement files	STORE	dbo.movement	STORE	Load directly from the source
	UPC		UPC	
	WEEK		WEEK	
	PROFIT		PROFIT	
	SALE		SALE	
	CATEGORY		CATEGORY	
UPC Files	UPC_ID	dbo.UPC	UPC_ID	Load directly from the source
	NITEM		NITEM	
	CAT_DESC		CAT_DESC	
	DESCRIP		DESCRIP	
	SIZE		SIZE	
	CASE		CASE	

b. From staging to presentation

Dimension - Store

Target Table	Target Column	Target Data Type	Source Table	Source Column	Mapping / Transformation Function
DW-	StoreKey	int	dbo.DEMO	STORE	Surrogate key

dbo.store_dim	StoreID	nvarchar		STORE	Copy
	StoreNum	nvarchar		STORE	Copy
	Density	nvarchar		DENSITY	Copy
	Income	numeric		INCOME	Copy
	Zone	int		ZONE	Datatype Conversion
	City	nvarchar		CITY	Cleanse
	Zip	bigint		ZIP	Copy

Dimension - Time

Target Table	Target Column	Target Data Type	Source Table	Source Column	Mapping / Transformation Function
DW-dbo.time_dim	TimeKey	int	dbo.time	TimeKey	Surrogate key
	StartDate	date		date	Datatype Conversion
	EndDate	date		date	Datatype Conversion
	SpecialEvents	varchar		SpecialEvents	copy
	week	nvarchar		WeekNo	copy

Dimension - Coupon

Target Table	Target Column	Target Data Type	Source Table	Source Column	Mapping / Transformation Function
DW-dbo.coupon_dim	CategoryKey	int	dbo.CCOUNT	CategoryKey	Surrogate key
	CategoryName	nvarchar		CategoryName	copy
	Date	date		Date	Datatype Conversion
	StoreNum	nvarchar		STORE	copy
	Amount	nvarchar		Amount	Aggregationnn

Dimension - Product

Target Table	Target Column	Target Data Type	Source Table	Source Column	Mapping / Transformation Function
DW-dbo.product_dim	ProductKey	int	dbo.CCOUNT	ProductKey	Surrogate key
	ProductId	nvarchar		ProductId	Surrogate key
	upc	nvarchar		upc	Clease
	description	nvarchar		description	Cleanse
	size	nvarchar		size	Cleanse

	SKU	nvarchar		SKU	Cleanse
	week	int		week	Cleanse & Datatype Conversion
	UnitsSold	int		UnitsSold	Cleanse & Datatype Conversion
	Quantity	int		Quantity	Cleanse & Datatype Conversion
	RetailPrice	decimal		RetailPrice	Datatype Conversion
	SalesCode	nvarchar		SalesCode	Cleanse
	Profit	int		Profit	Datatype Conversion

Dimension - sales_fact

Target Table	Target Column	Target Data Type	Source Table	Source Column	Mapping / Transformation Function
DW-dbo.sales_fact	StoreKey	int	The source for the fact table is the 3 dimension tables that we have defined: Store Time	StoreKey	Foreign key corresponding to primary key store_id of store_dim dimension
	TimeKey	int		TimeKey	Foreign key corresponding to primary key product_id of

					product_dim dimension
	Date	date		Date	copy
	StoreNum	varchar		StoreNum	copy
	TotalSales	decimal		TotalSales	copy
	SpecialEvents	varchar		SpecialEvents	copy
	Week	varchar		Week	copy

Dimension - profit_loss

Target Table	Target Column	Target Data Type	Source Table	Source Column	Mapping / Transformation Function
profit_loss_fact	StoreKey	int	The source for the fact table are the 4 dimension tables that we have defined : Store Coupon Time Product	StoreKey	Surrogate key
	CategoryKey	int		CategoryKey	copy
	TimeKey	int		TimeKey	copy
	ProductKey	int		ProductKey	copy
	StoreNum	varchar		StoreNum	copy
	Date	date		Date	copy
	Amount	varchar		Amount	copy
	CategoryName	varchar		CategoryName	copy
	ProfitLossFlag	varchar		ProfitLossFlag	Derived

					based on Profit value

Data extraction rules

Data has been taken out of the given CSV files and the DFF data handbook. The following extraction guidelines are used:

- The DATE function is used to convert the original DATE field's integer date values into the Date format.
- The sales statistics from the several columns, which reflect the DFF-defined departments, are combined to form "TOTAL SALES."
- All of the coupon columns from the original table are combined to form the "COUPON TOTAL."
- The datatype of store IDs, Week, and product category coupon values have been changed from varchar to int.
- Junk data with missing MMIDs was cleaned during the data extraction process for the demo table.
- Demo.csv is being used to extract the DEMO table. During the extraction procedure, the rows with non-numeric Store IDs were removed.

Data Transformation

Data transformation and cleaning must be done after the data has been extracted and placed in the staging area. Once the data has been imported into the staging area, it must be transformed and cleaned to preserve its consistency and integrity. The data is loaded into the data mart for effective processing after cleaning it. The steps we took for a few tables in transformation and cleansing are below:

Sales Calculation: Using the transformations listed below, the sales amount for each row in the staging area's Movement table has been determined.

*Sales_in_dollars = (Unit_Price*Move/Quantity)*

Quotation Marks Removal: For data fields, we eliminated the quote mark, and we eliminated dates that were not numeric for customer counts.

Data Type Conversion (Casting): The data types were transformed before the data were loaded into the tables.

Surrogate Keys Creation: To map all of the dimension tables to the fact tables, substitute keys are required. These substitute keys serve as distinctive identifiers as well. The following are the Surrogate Keys:

StoreKey: Store dimension table unique identifier
CategoryKey: Coupon dimension table unique identifier
TimeKey: Time dimension table unique identifier
ProductKey: Product dimension table unique identifier

Special Symbols Elimination: Data cleaning will be aided by the removal of some special characters that are present in the dataset but do not make sense. As an illustration, the removal of the quotes from the varchar columns in the Demo table will follow the prescribed structure. The effectiveness of data processing will enhance the procedure as a whole and maintain its standard structure.

Irrelevant Data Removal: Some characteristics have been eliminated since they are not required to answer the business questions we have selected for this case study.

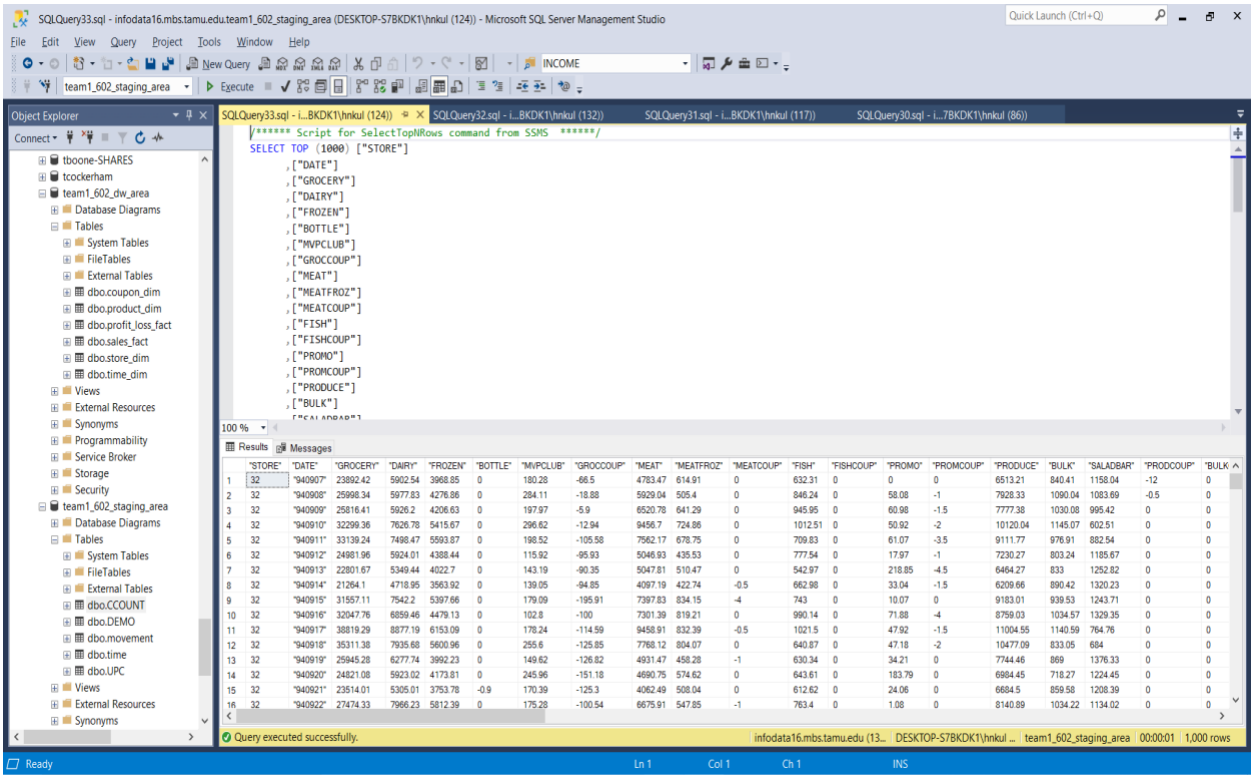
Null Value Removal: To ensure consistency throughout the data, several crucial columns that contained null values needed to be deleted. Because data in stores, for instance, cannot be null, these entries must be deleted.

Eliminating Negative Values: There are several negative values that must only contain positive values, much as the null values. These principles, much like the week's value, cannot ever be accepted in a bad situation. It is necessary to delete these kinds of records as a result. To ensure that these kinds of irrelevant variables do not affect the findings from reporting from the data warehouse, it is crucial to eliminate them.

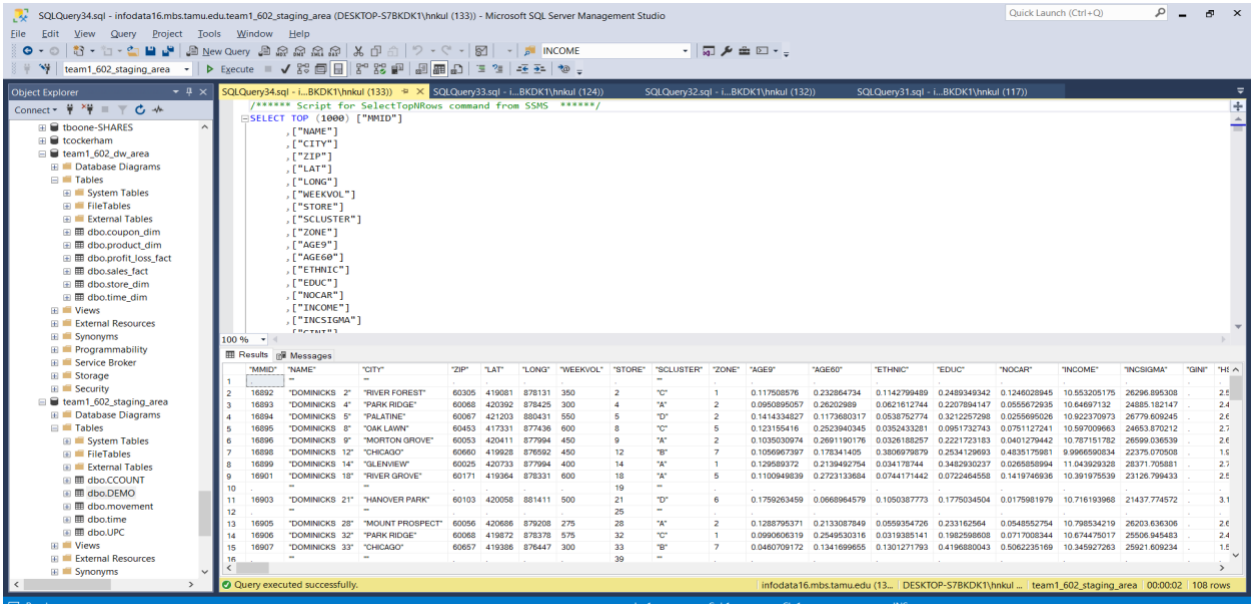
Formulaic Calculations: To match the level of detail required by the tables we have chosen to use to address our business questions, some attributes have been created as a result of applying formulas and doing calculations on the existing attributes and records.

Screen Grabs <u>Before</u> Transformation & Cleansing:

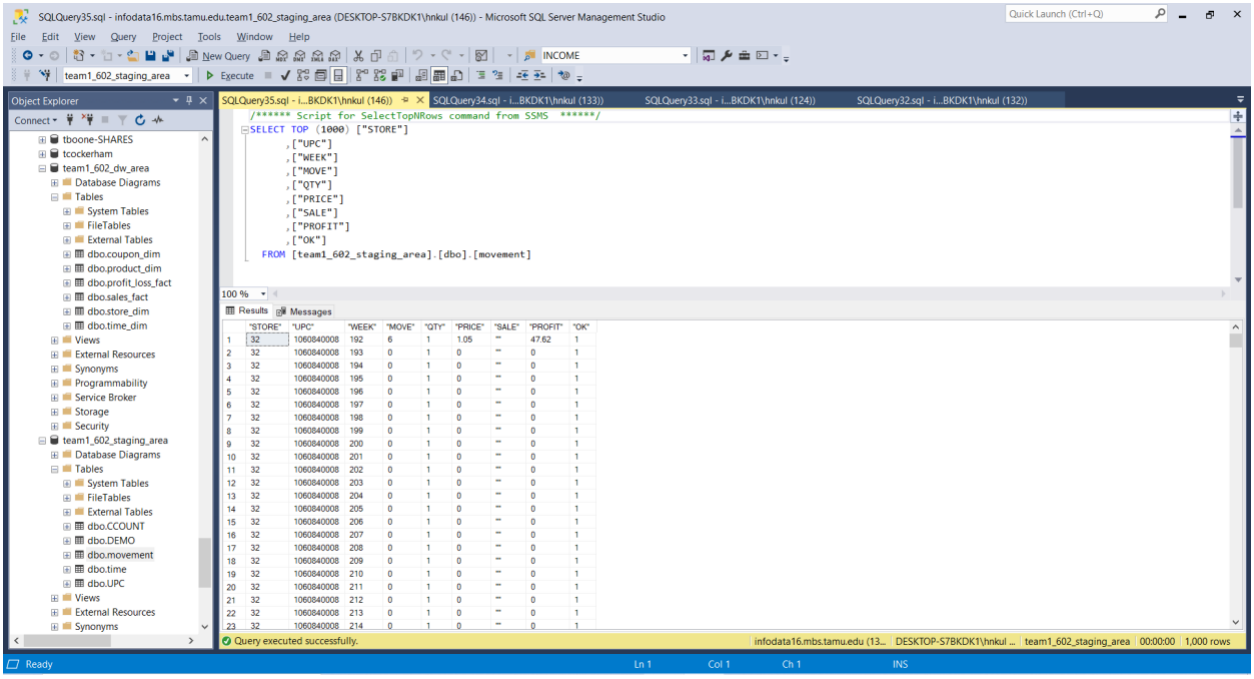
CCount.csv



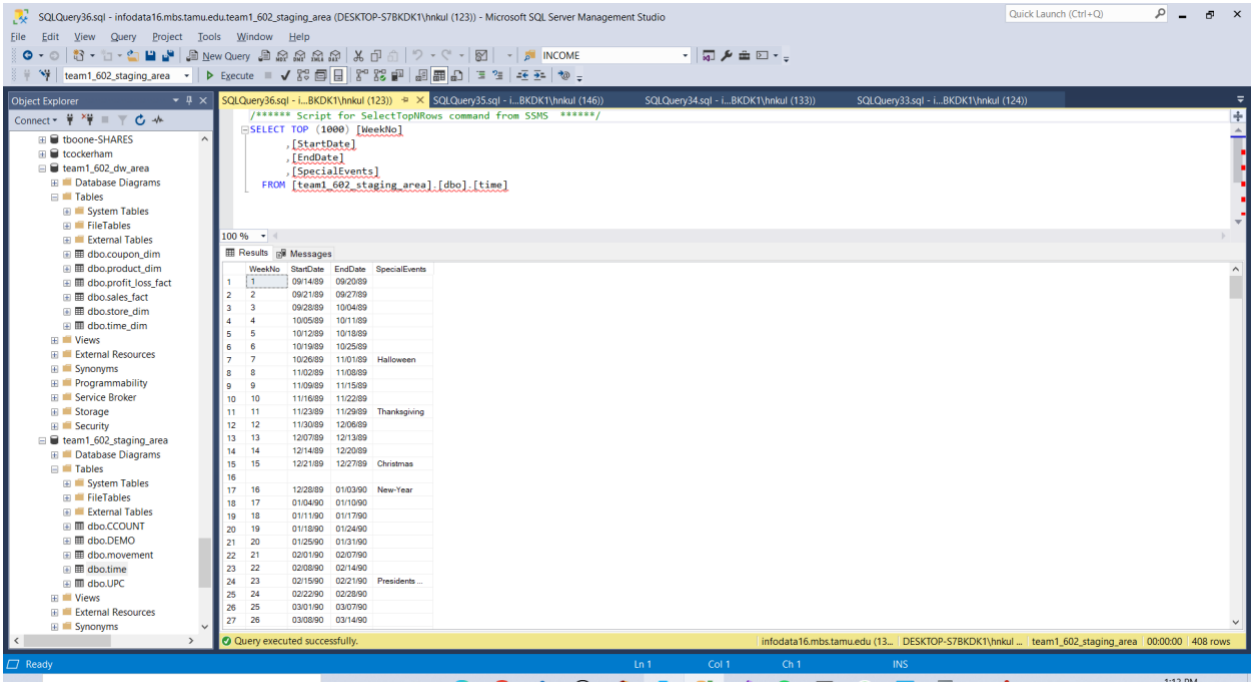
DEMO.csv



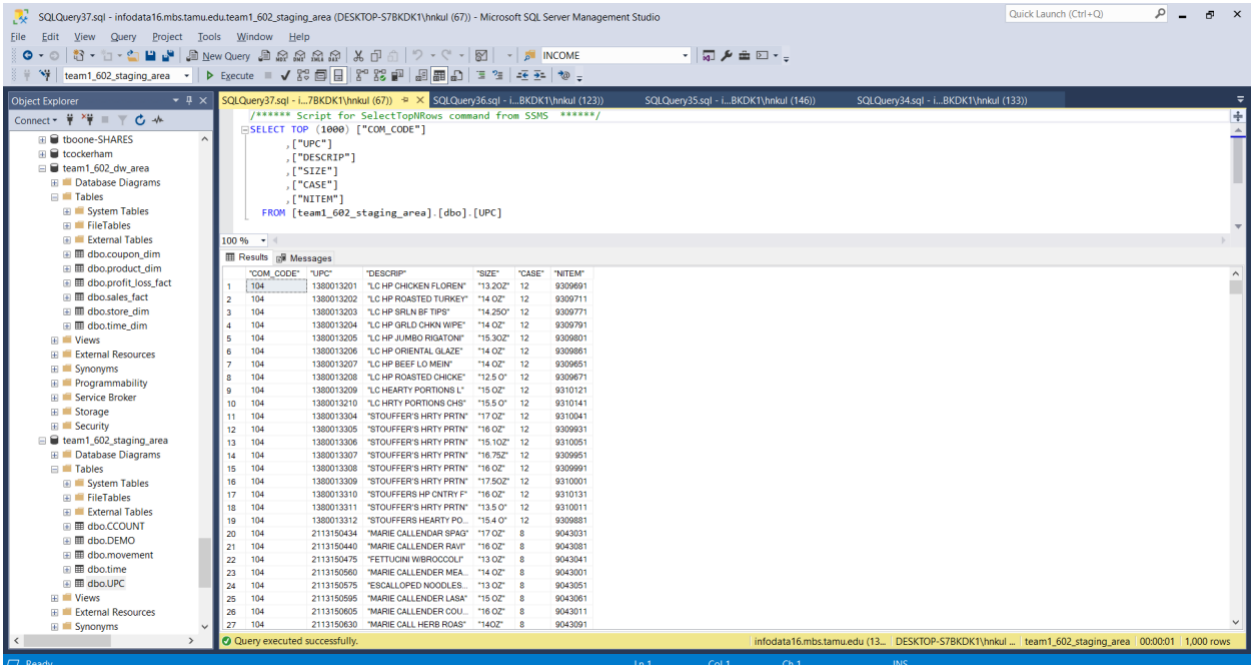
Movement files



Time.csv

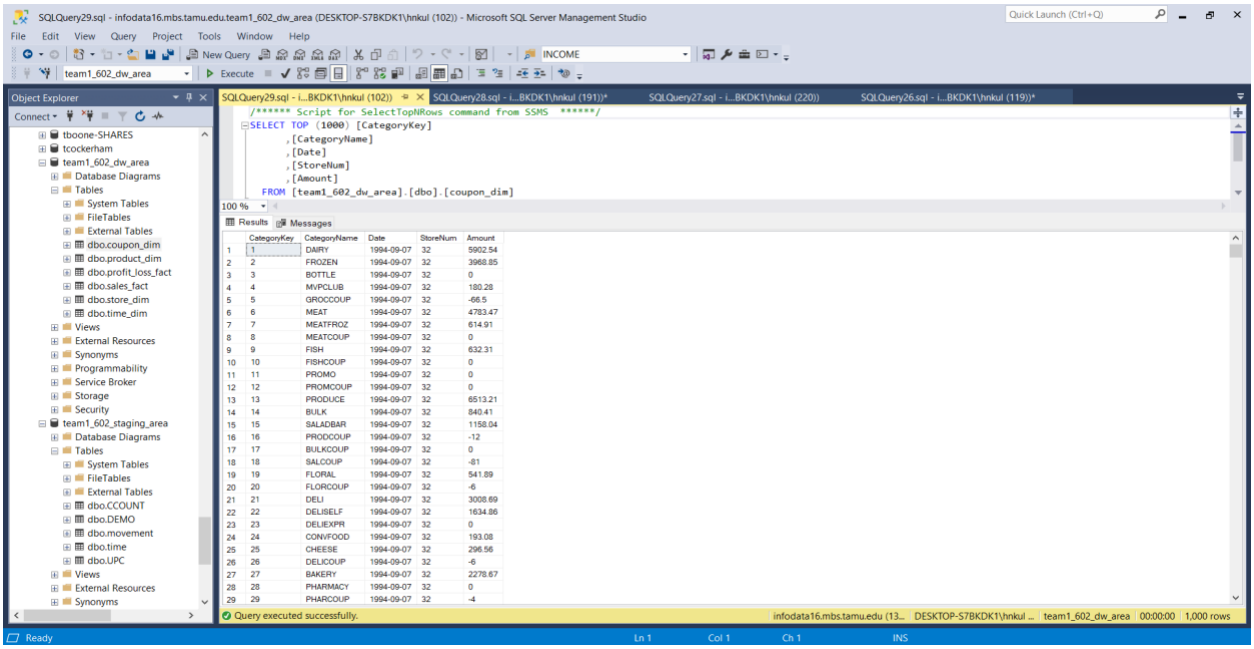


UPC files

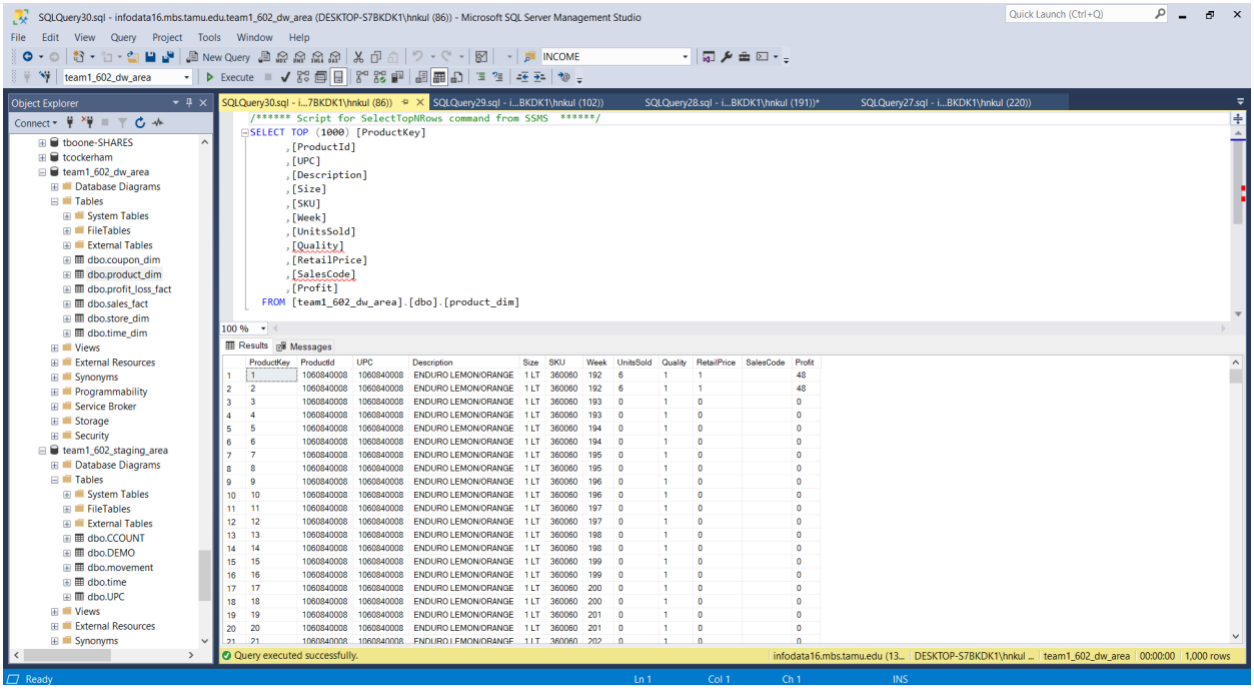


Screen Grabs after Transformation & Cleansing:

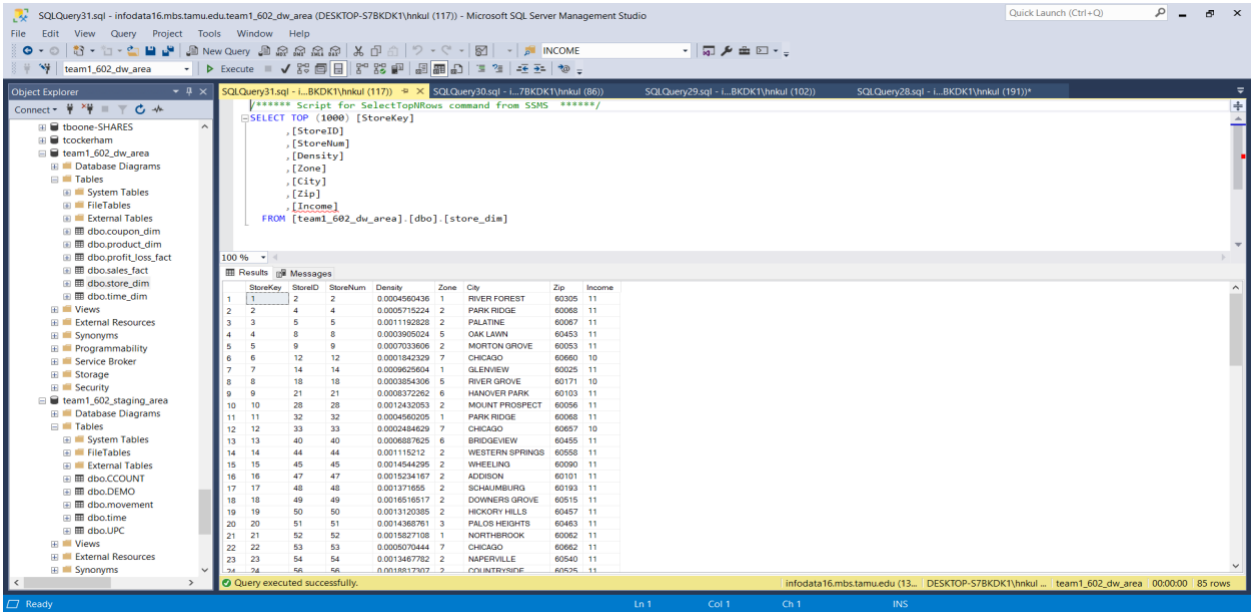
Coupon Dimension



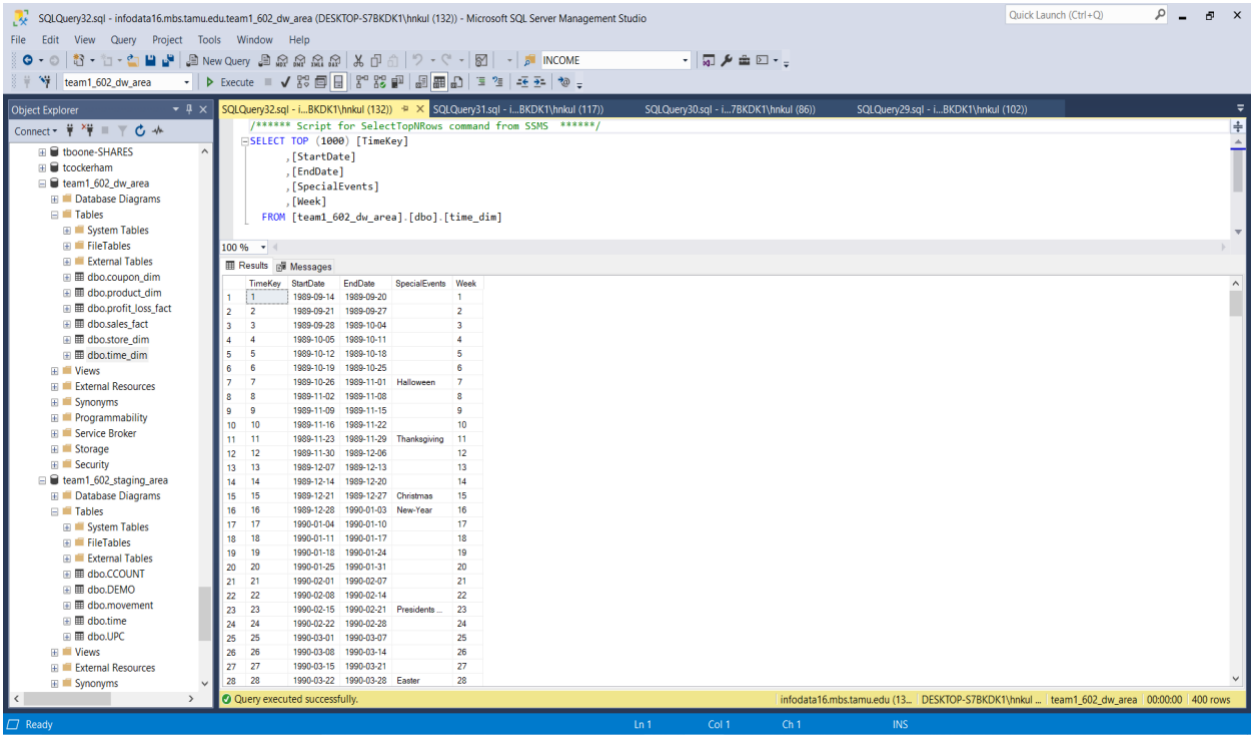
Product Dimension



Store Dimension



Time Dimension



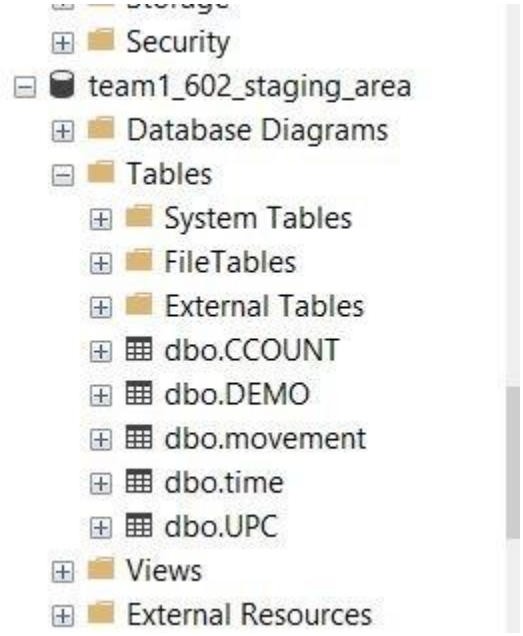
Plan for Aggregate tables

The users would be able to move up and down whichever degree of aggregation they wanted and get the precise information for their report if the data were stored at the lowest granularity level possible. It might, however, affect the timing and effectiveness. Therefore, we must examine the granularity level we want to utilize for data storage and put ourselves in the position of a data architect.

The speed and effectiveness of data warehousing would both be greatly enhanced by data aggregation. Prior to getting started, it's important to think about our business needs and determine which aggregation plan will work best for us. In our case, not all the business questions need deep granularity, but for those who do need the deep level of granularity, we have precalculated the result, and in other such cases, we keep the data at the granular level that fits the requirement in question.

Organization of Data staging area

The team1_602_staging_area database has been used to store the data that was taken from the data source. Only the necessary columns from the original Excel files were chosen. Data from the staging area will undergo additional transformations for data marts. The following are screenshots of the various tables in the staging area:



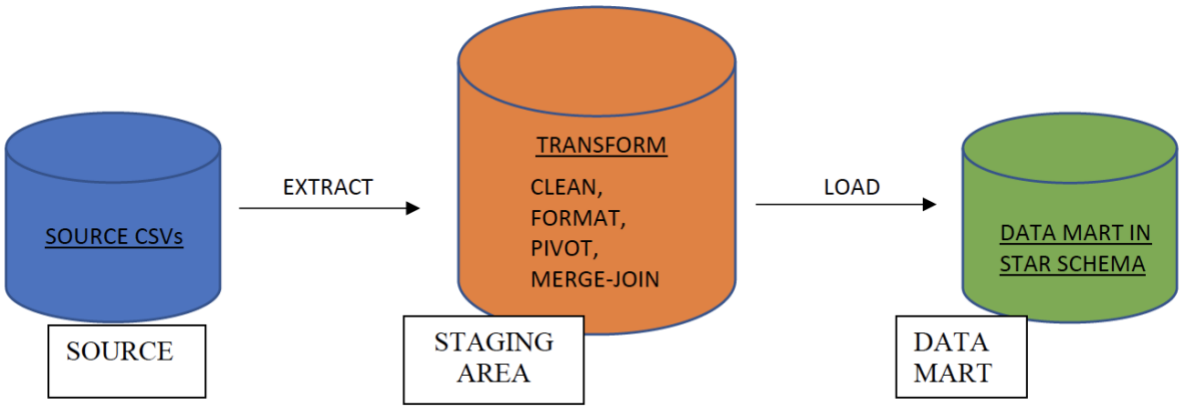
Procedure for Data Extraction and Loading

The source data files are in CSV format which can be considered to be flat files. These flat files have been imported into the staging database as is and into tables with the exact same names as the files. The Import/Export wizard from SSIS has been used to extract and load the data. During extraction, several columns have been excluded and these actions have been documented in an earlier section of this report under “Extraction rules”. The Import/export wizard in SSIS proved handy by letting us choose how and what data gets extracted in a simple point-and-click fashion rather than having to use complex SQL code. It allowed us to select the flat files as the source and the staging database as the destination. The import/export wizard automatically created a data flow task from source to destination in the case of the extraction. The package that was created using the wizard was then executed to implement the transfer. Once the transfer was complete we went

back to SSMS to check if the data had been transferred over to the staging database exactly the way we wanted it.

The Loading process also involved using the import/export wizard. But this time the tables that had been brought into the staging database were the source and the final tables that we had planned on storing in the presentation server were the destination. After the transformations were completed on the staging tables, the data was then loaded into the presentation server. The surrogate keys for the dimension tables and the fact tables were created right before loading. To be precise, they were included in the destination table creation queries. Once the tables were created it was just a matter of selecting them as the destination in the SSIS import/export wizard. The packages were created and executed to complete the transfer.

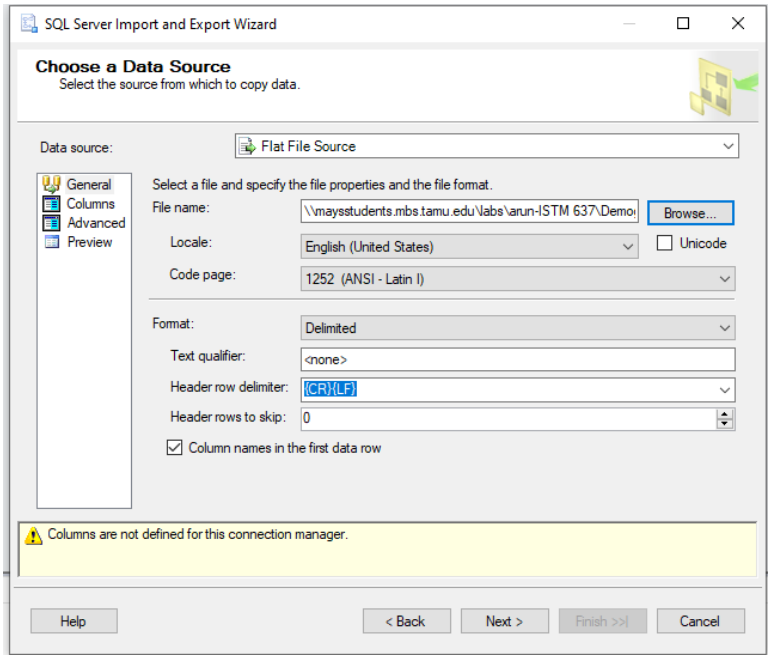
Implementation of ETL

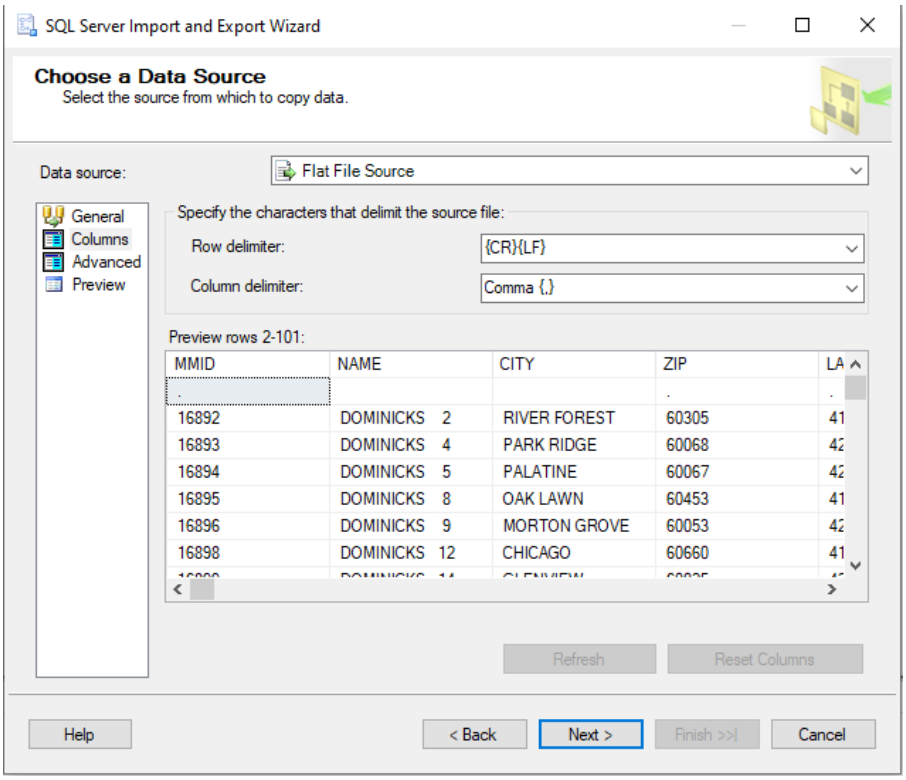


The ETL strategy for dimension tables are discussed in this section, along with flow diagrams. In the section on ETL implementation, their implementations are demonstrated in detail, together with package execution.

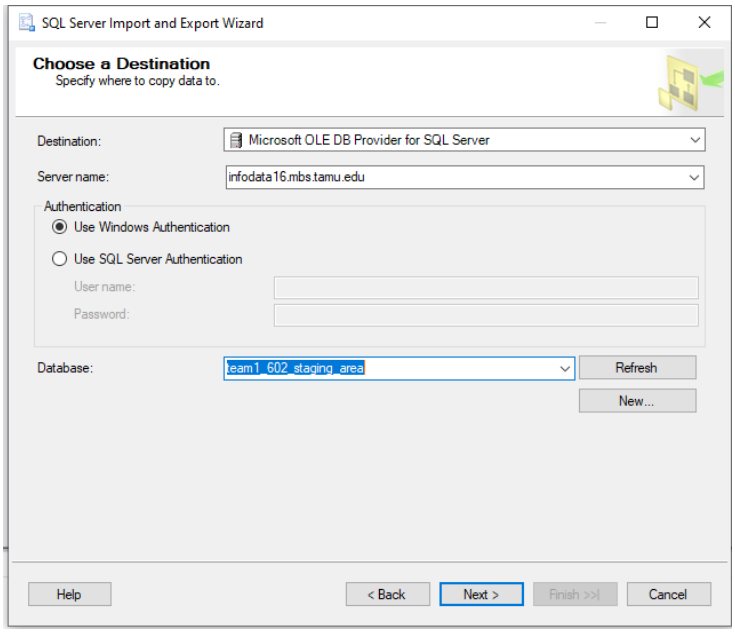
ETL for CCount.csv file

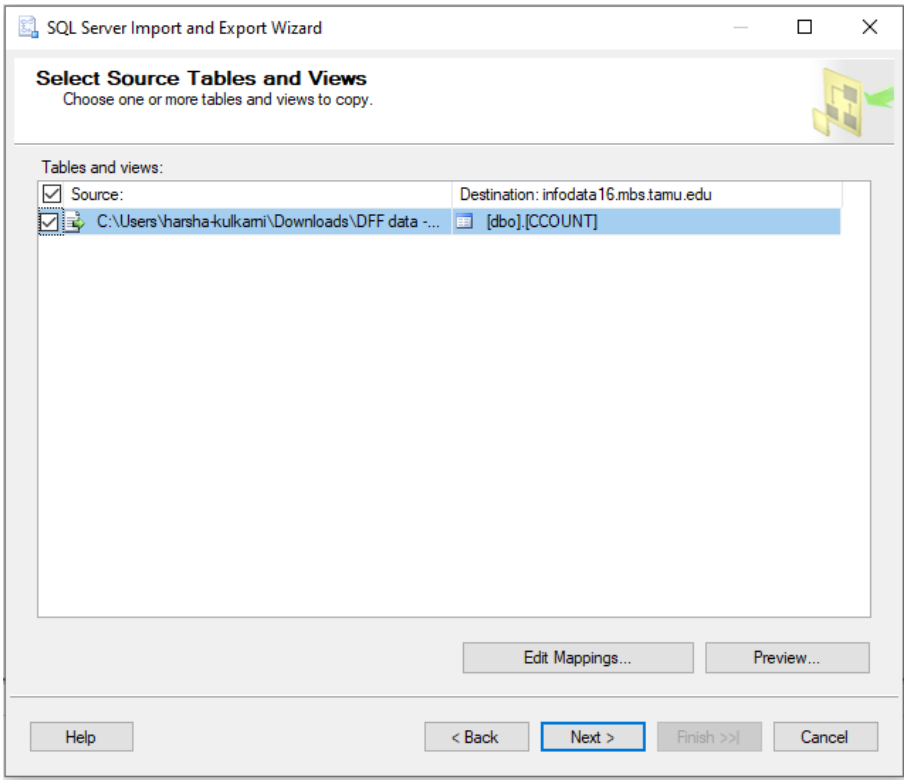
Step 1: Select data source



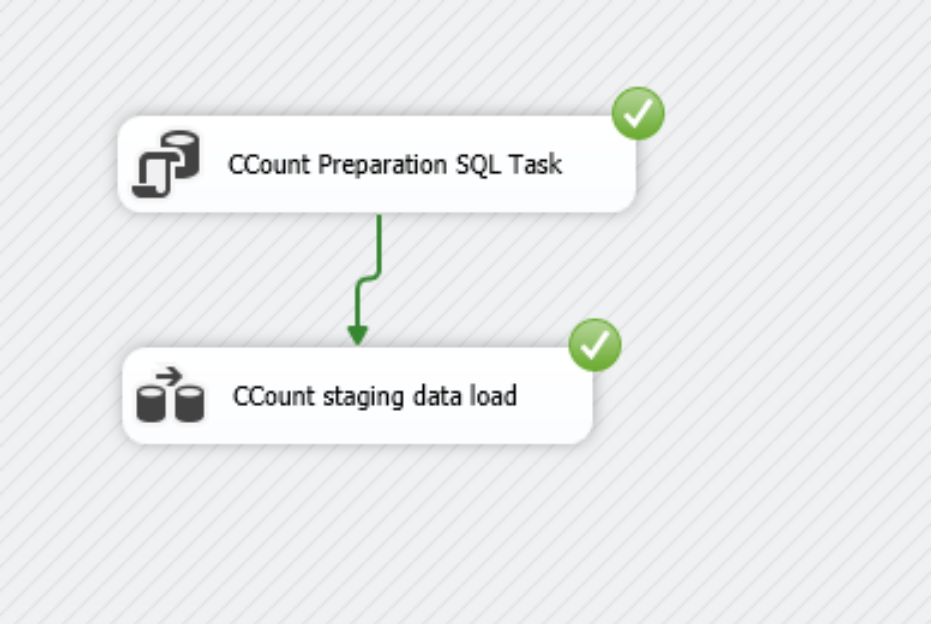


Step 2: Select destination (staging area in our case)





Step 3 : Execute Package

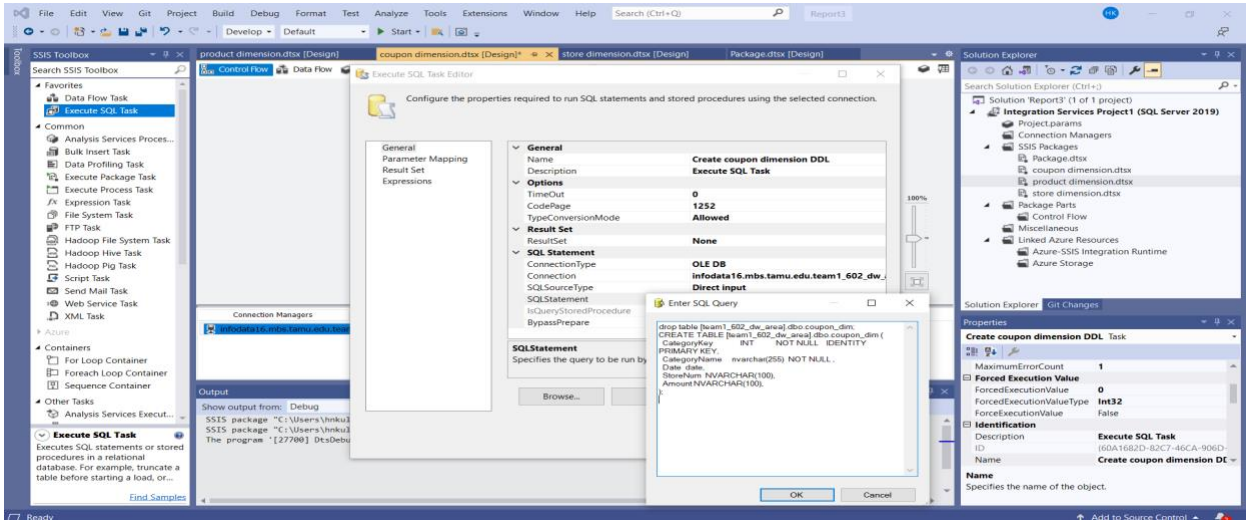


ETL for Dimension tables

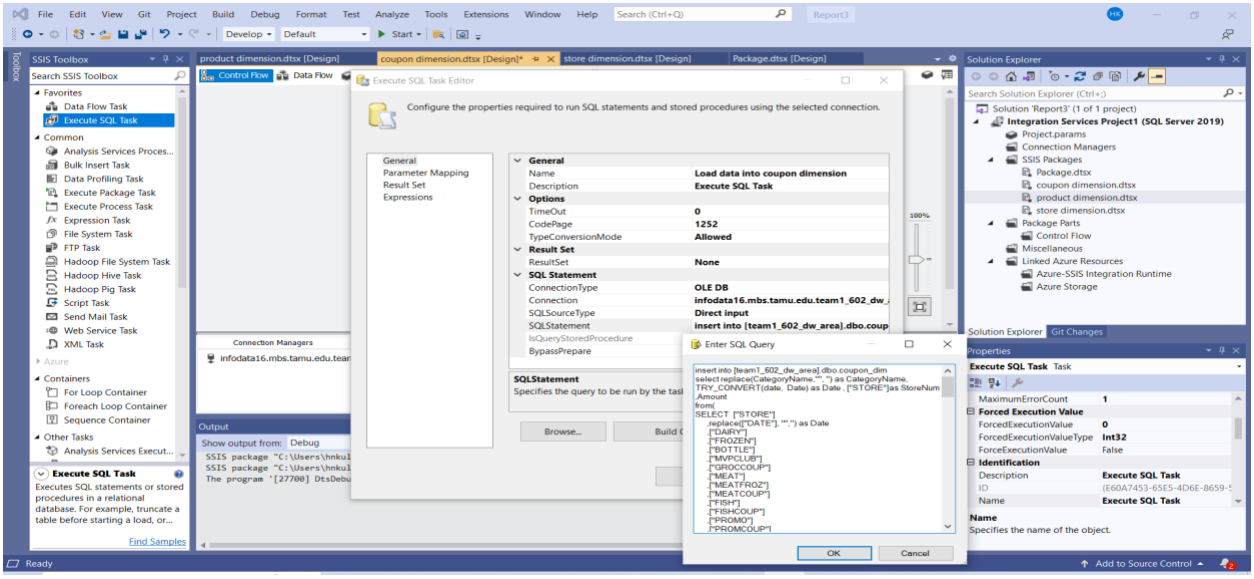
Coupon Dimension

The coupon dimension table contains columns CategoryKey(which is the primary key for the table) CategoryName, Date, StoreNum, and Amount which are obtained for undergoing the transformation process.

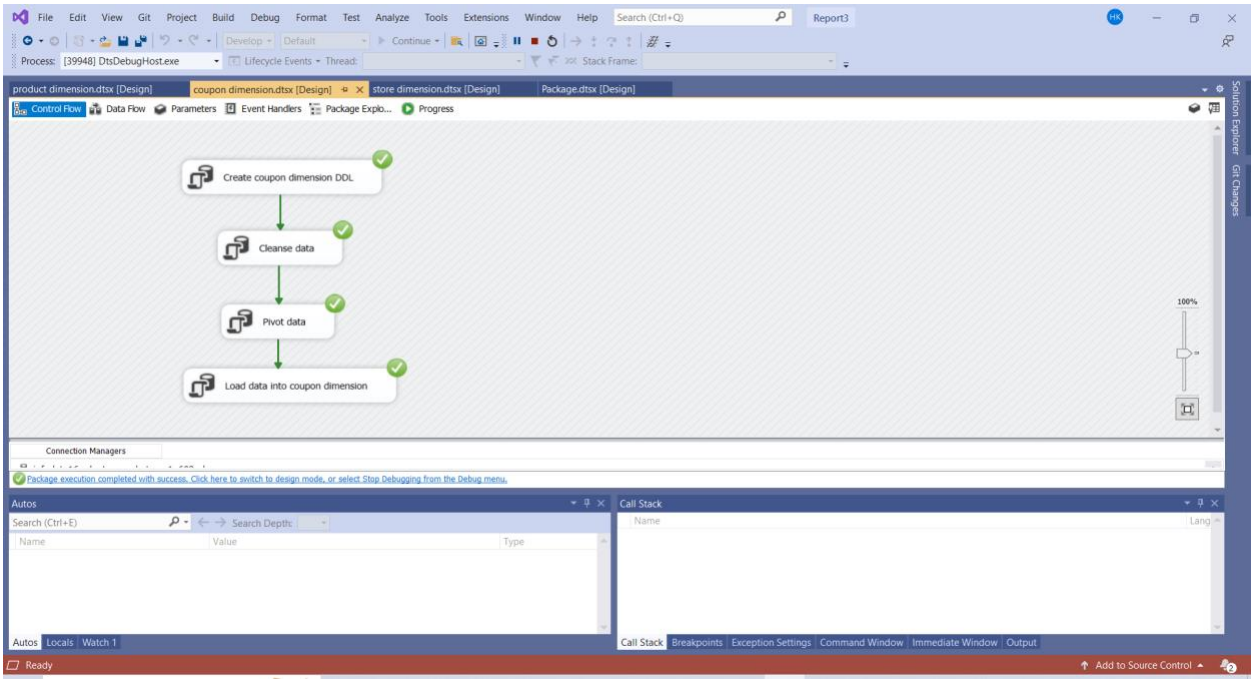
Creation:



Cleaning and loading:

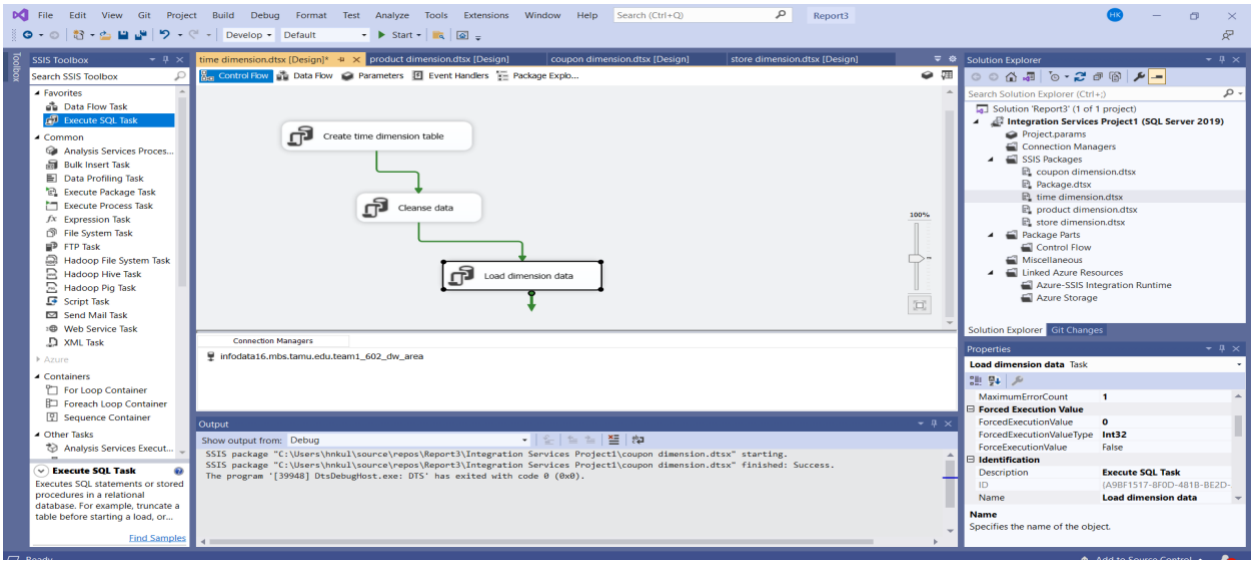


Execution:

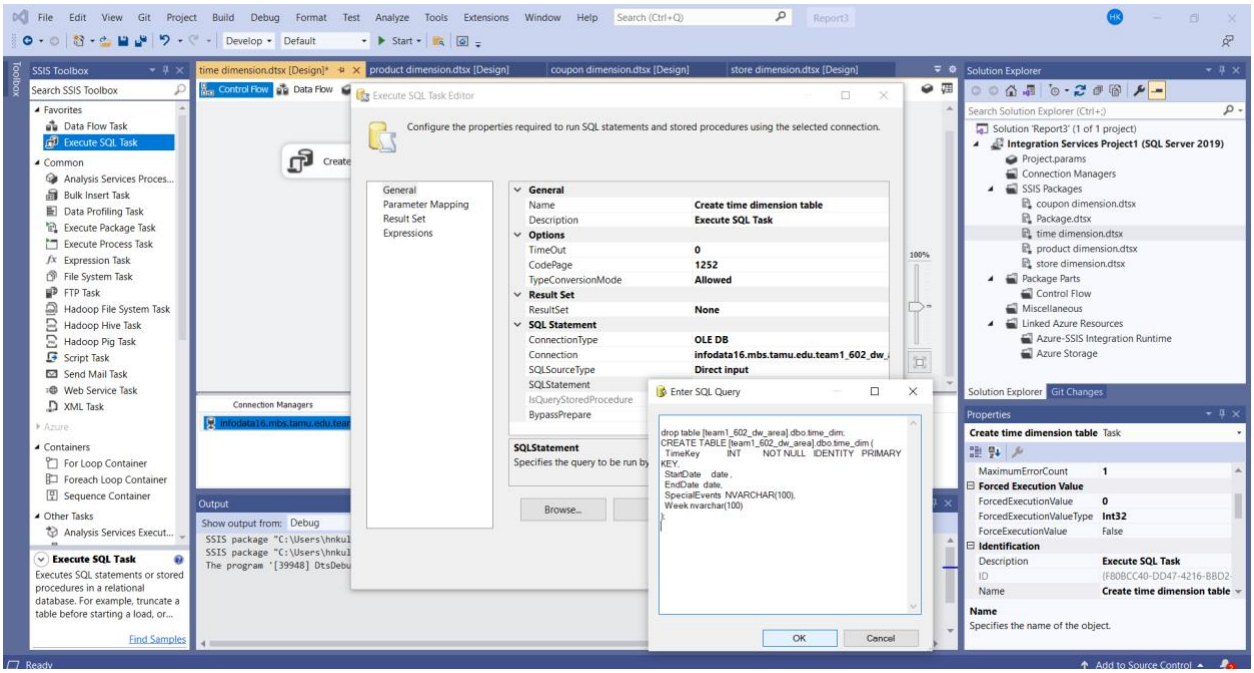


Time Dimension

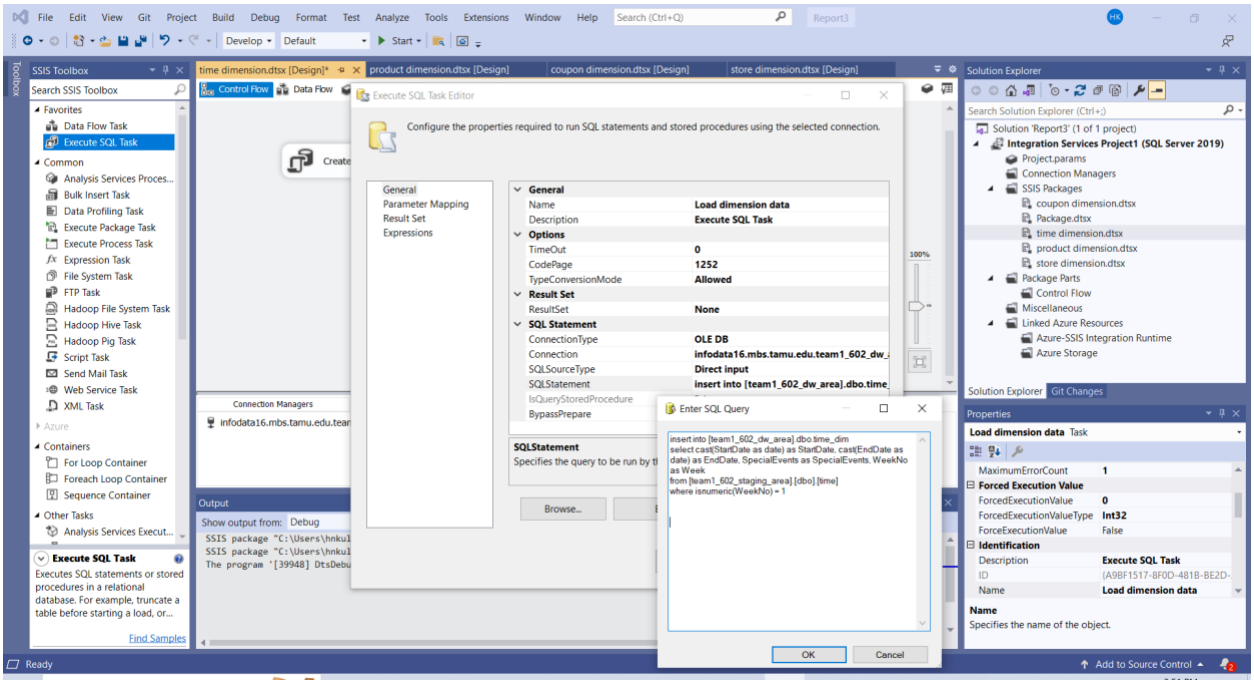
Time-dim is very much like store_dim data. The source for this would be DFF's cookbook. It contains columns date, year, month, week, and quarter which are obtained after doing the transformation.



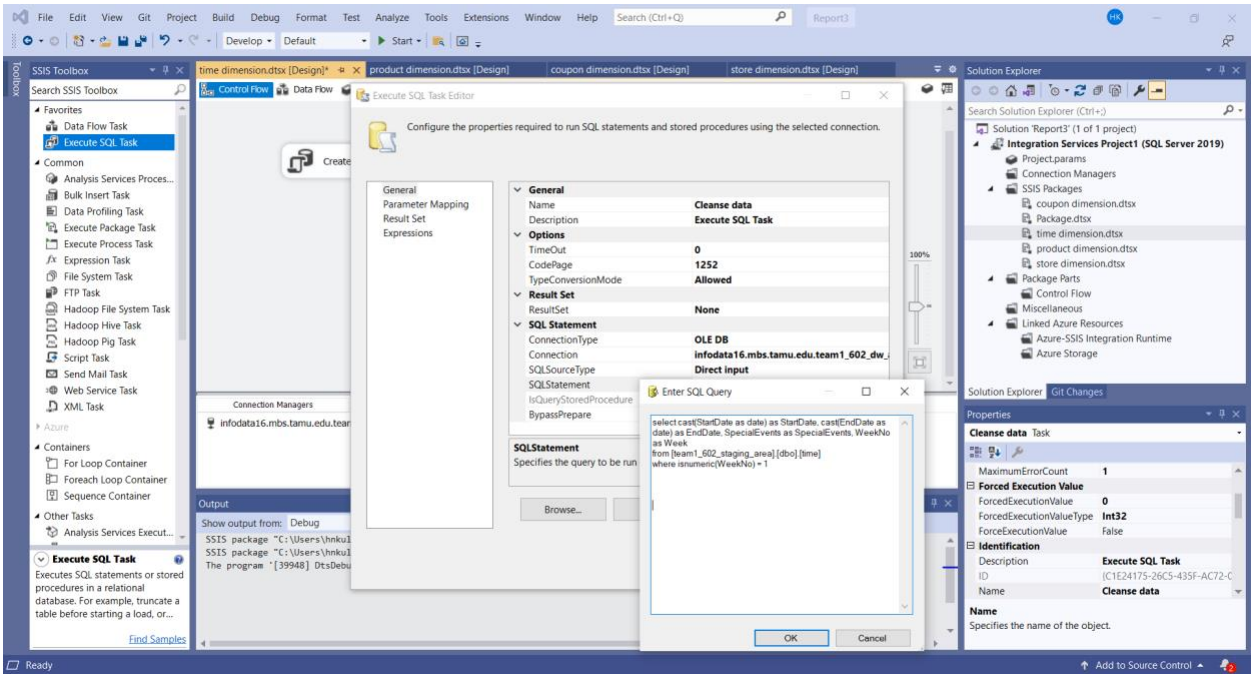
Creation :



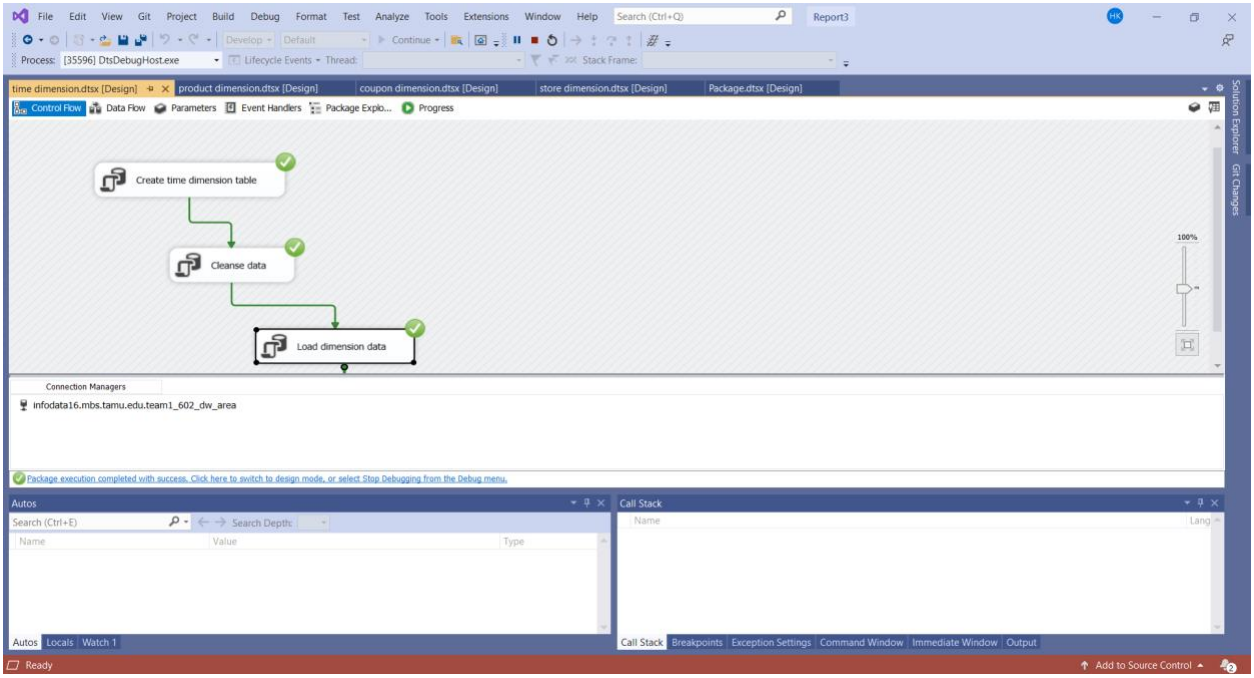
Loading:



Cleaning :

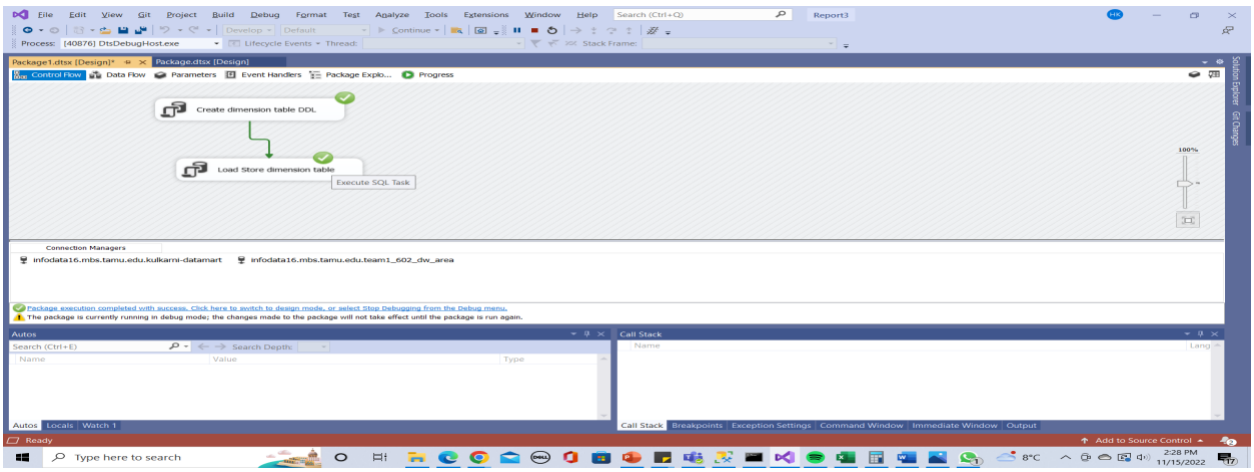


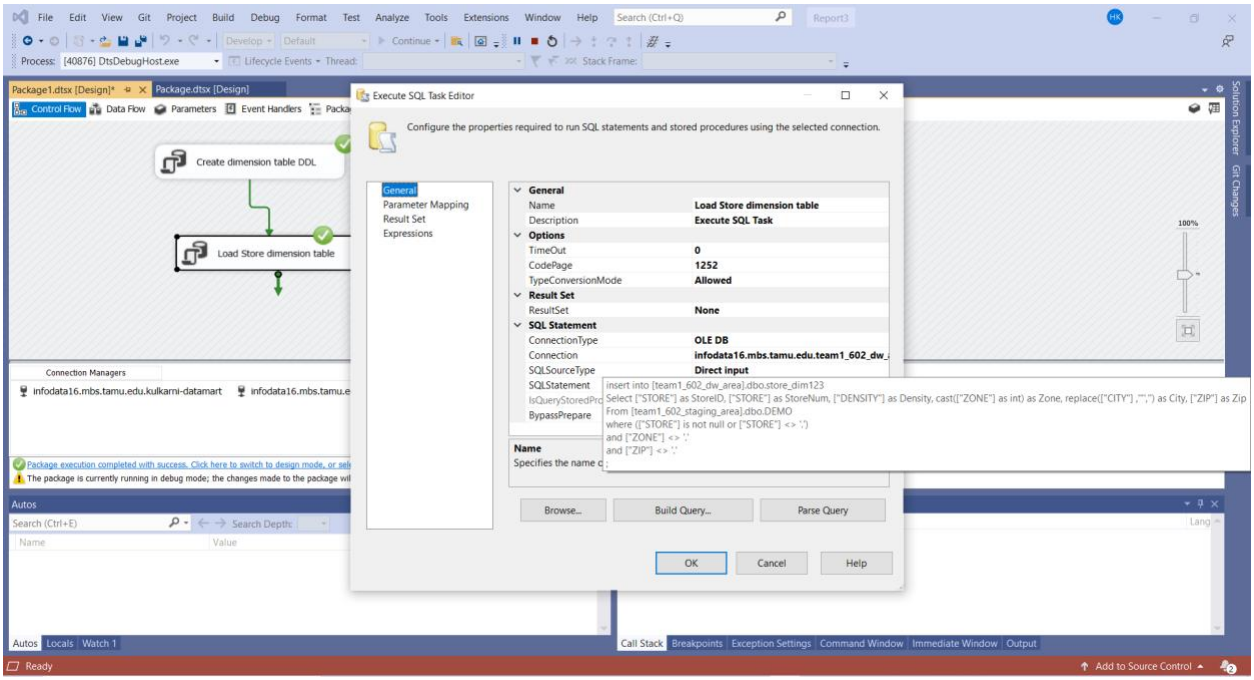
Execution:



Dimension Store

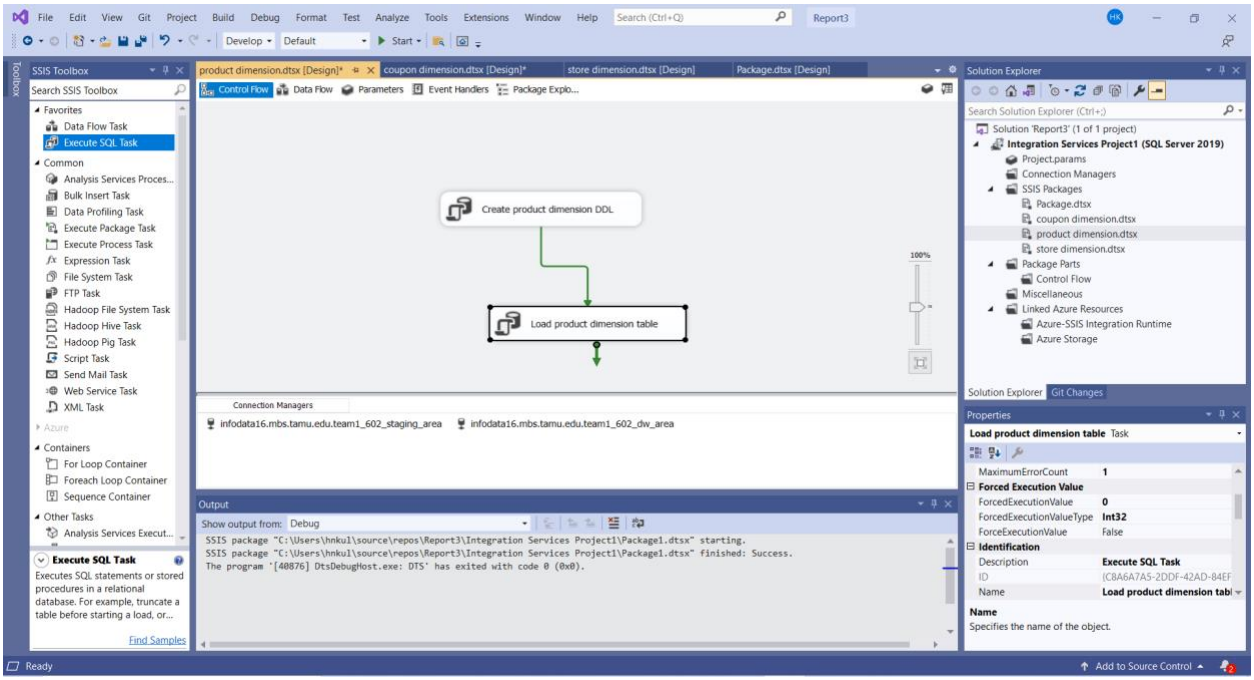
Raw data from DFF's cookbook serves as the source data for Dim Store. It has already been cleaned. Additionally, we modified the data types of various columns, including StoreID and ZipCode. The store key, the principal key, is an auto-increment key.



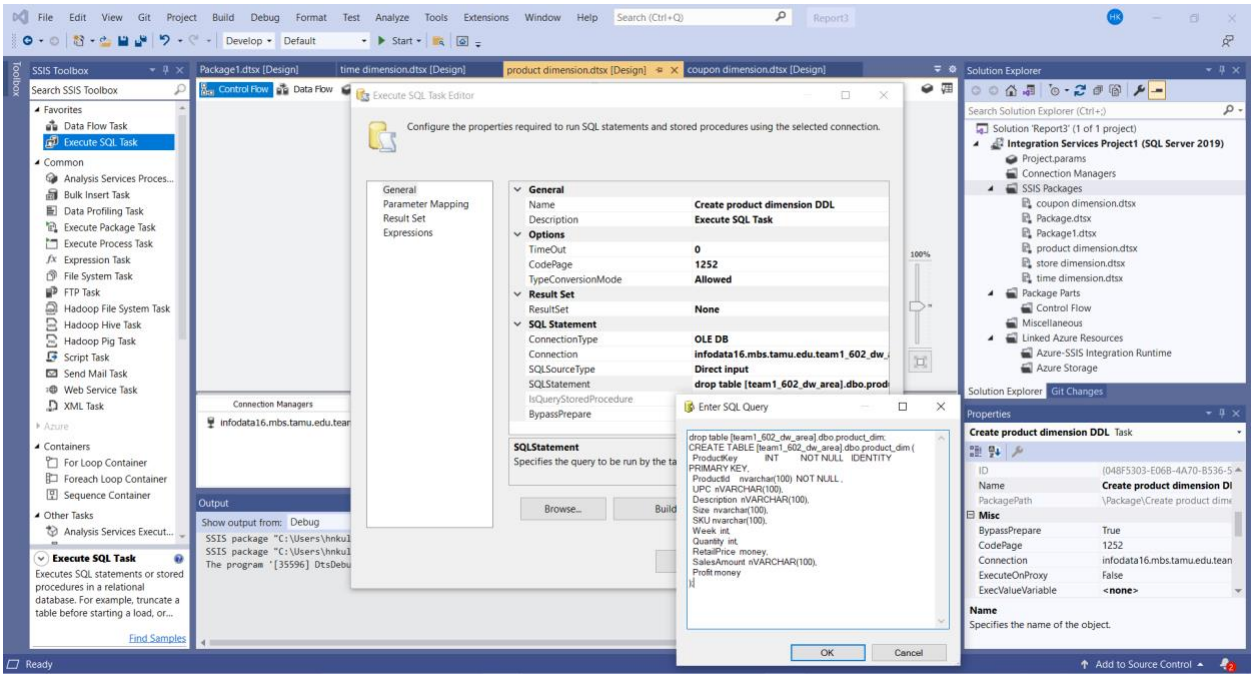


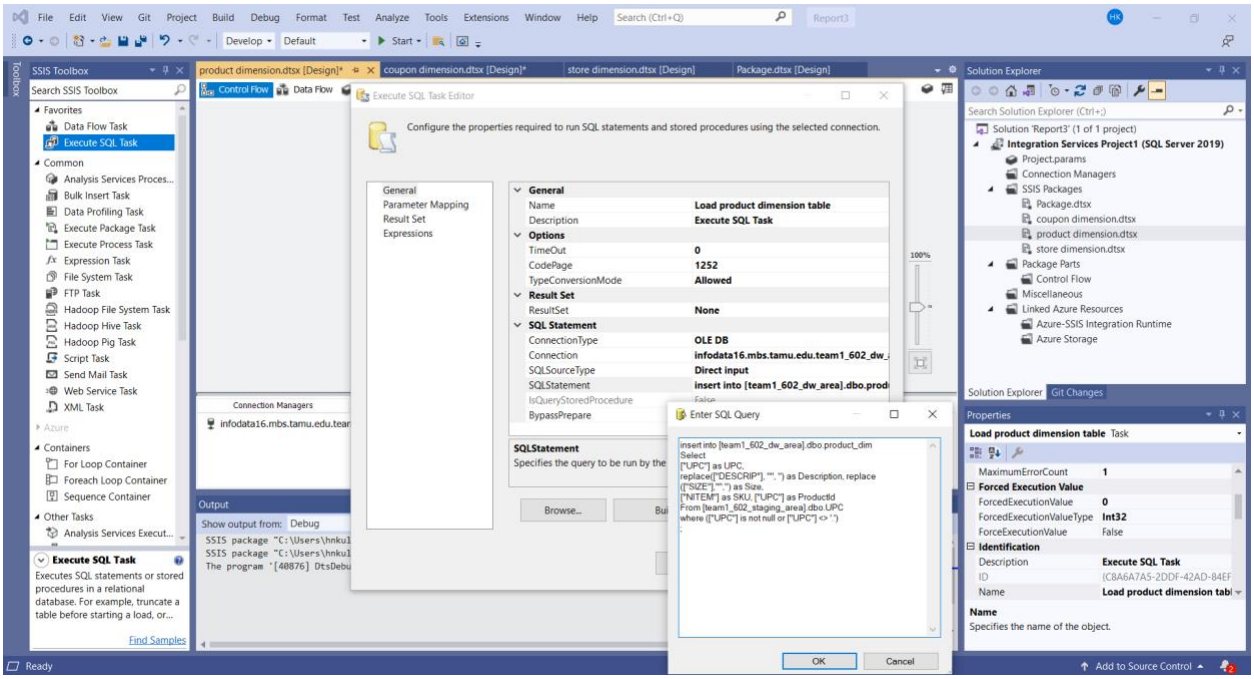
Product Dimension

Product dimension tables contain columns ProductKey(which is the primary key), ProductID, UPC, Description, Size, SKU, Week, UnitsSold, Quality, RetailPrice, SalesCode and Profit.

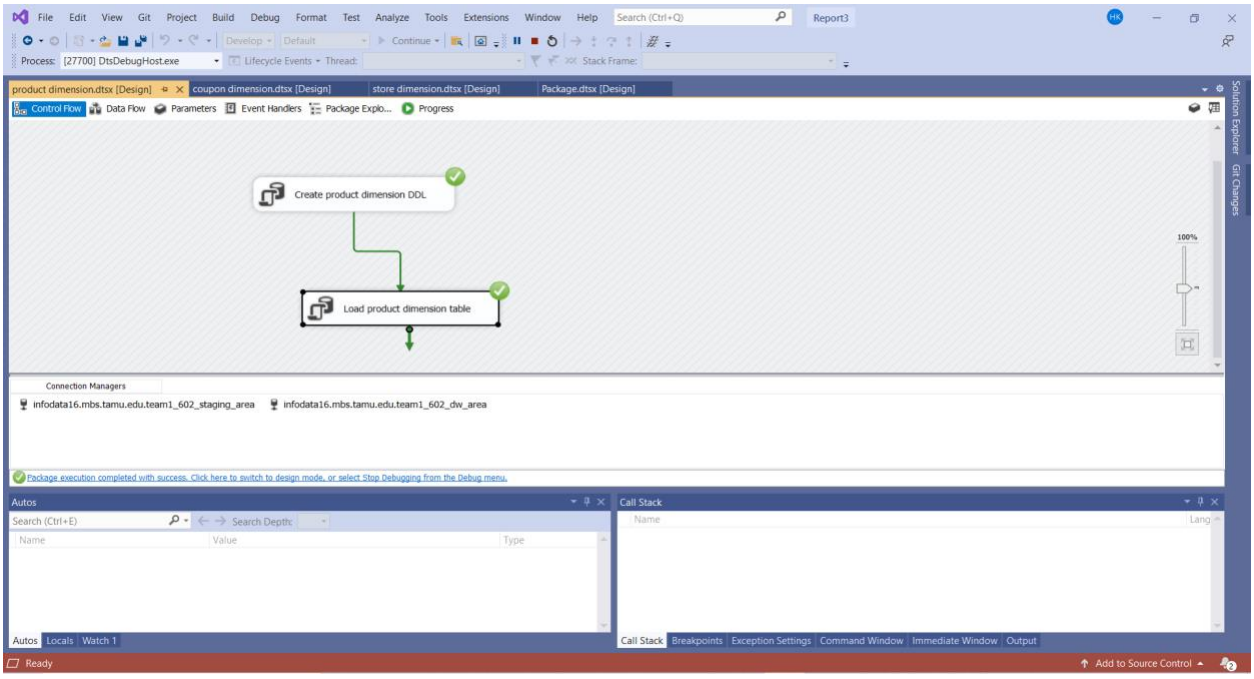


Creation:




Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
Enter SQL Query
OK
Cancel

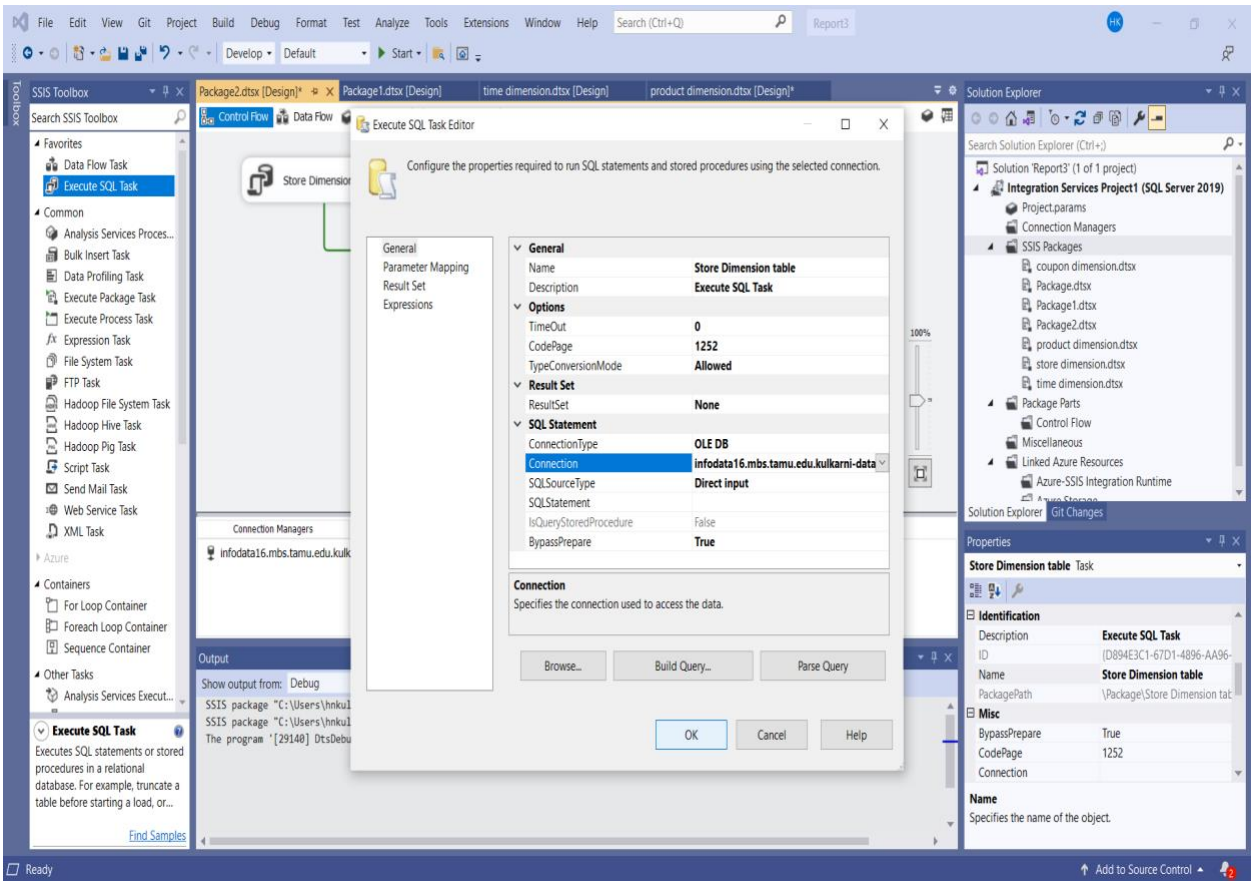
Execution:

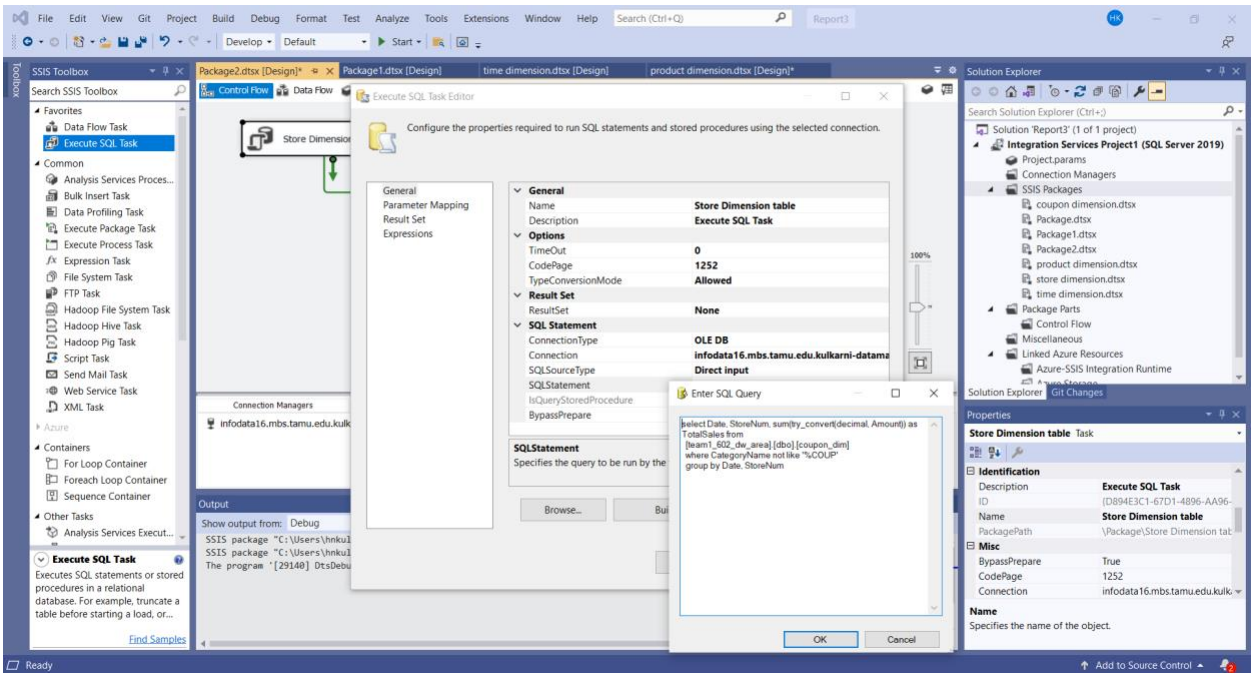

Create product dimension DDL
Load product dimension table
Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

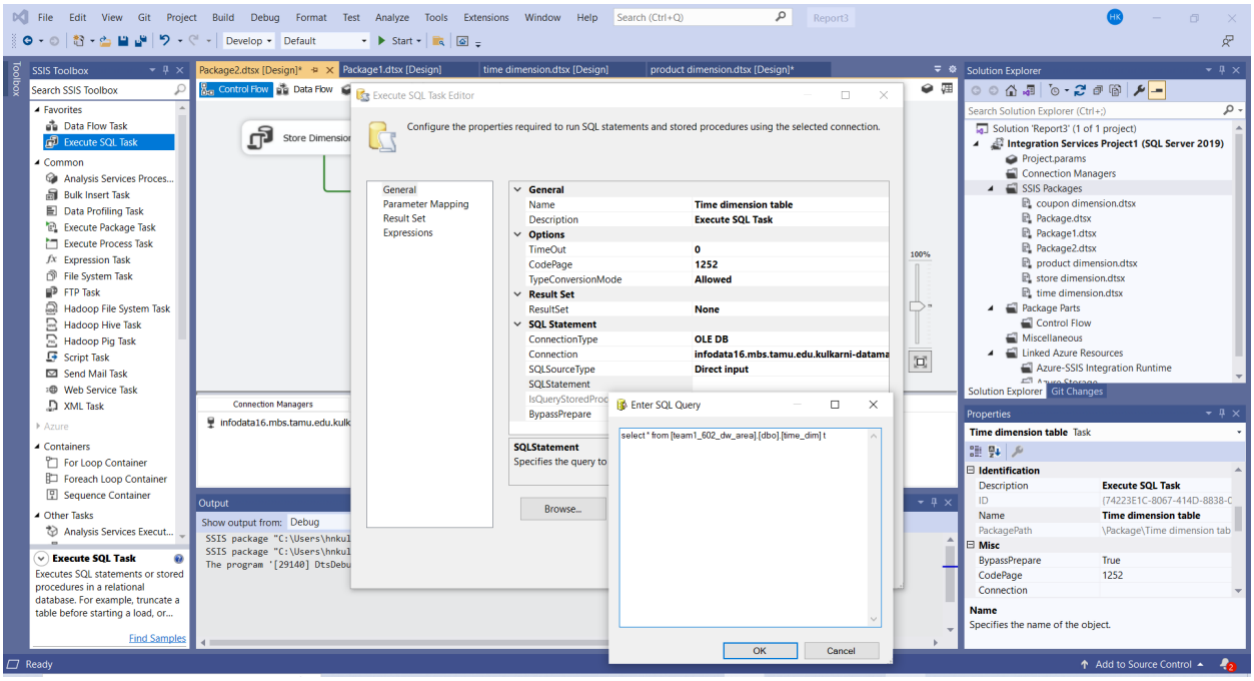
ETL for fact tables

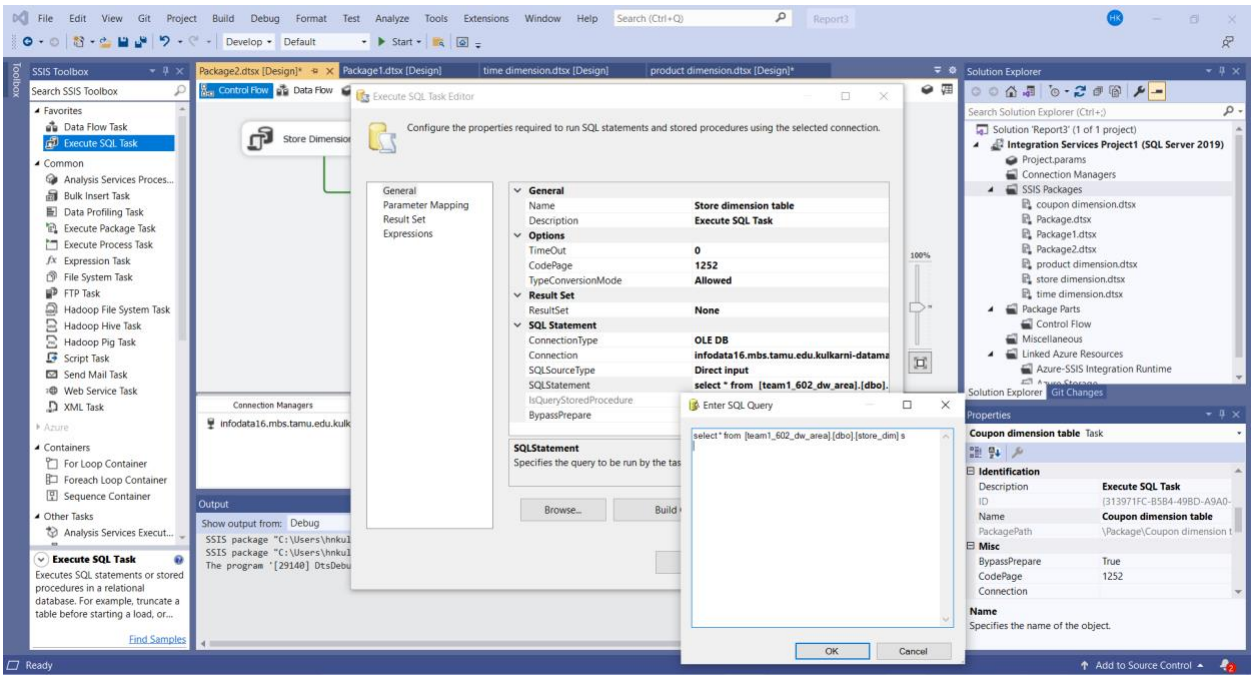
sales_fact

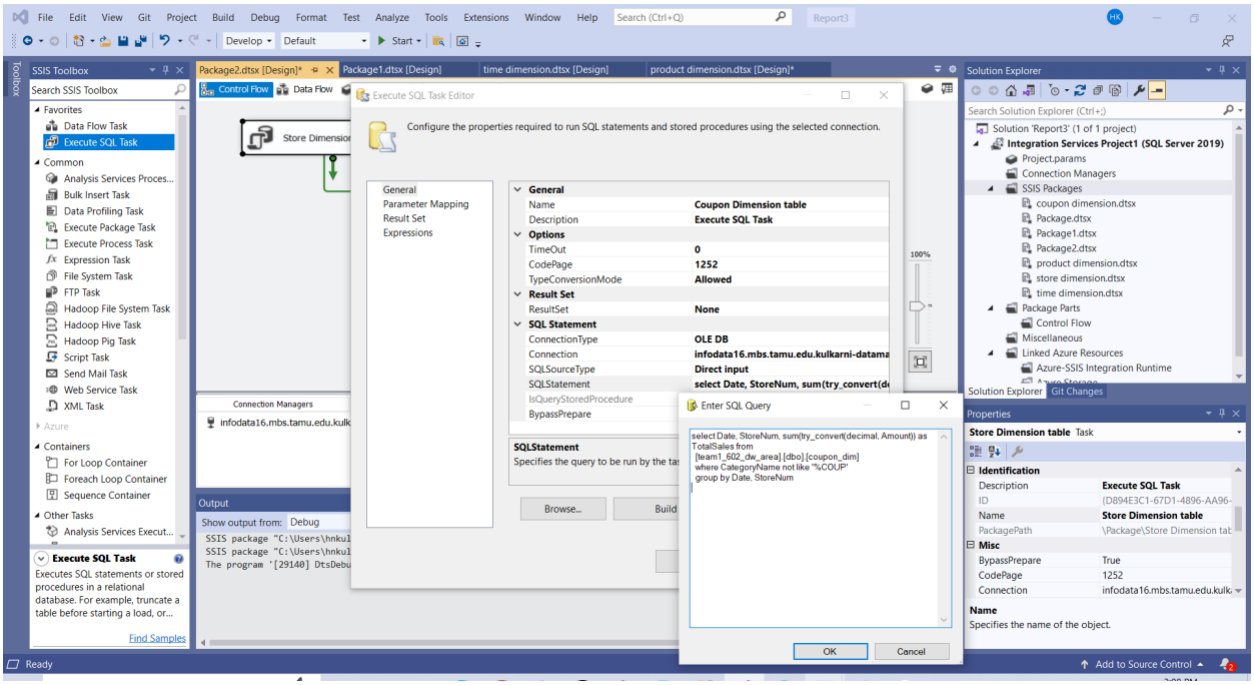
The Sales fact table's source data comes from the Movement table. The value produced from the Price, Quantity, and Movement variables in the Movement table is SALES AMOUNT. This table underwent cleaning and was maintained in the staging area. On this table, a lookup transformation was done. Before loading the data in the SALES FACT database. The below screenshots describe the queries used to produce and populate the sales_fact table.

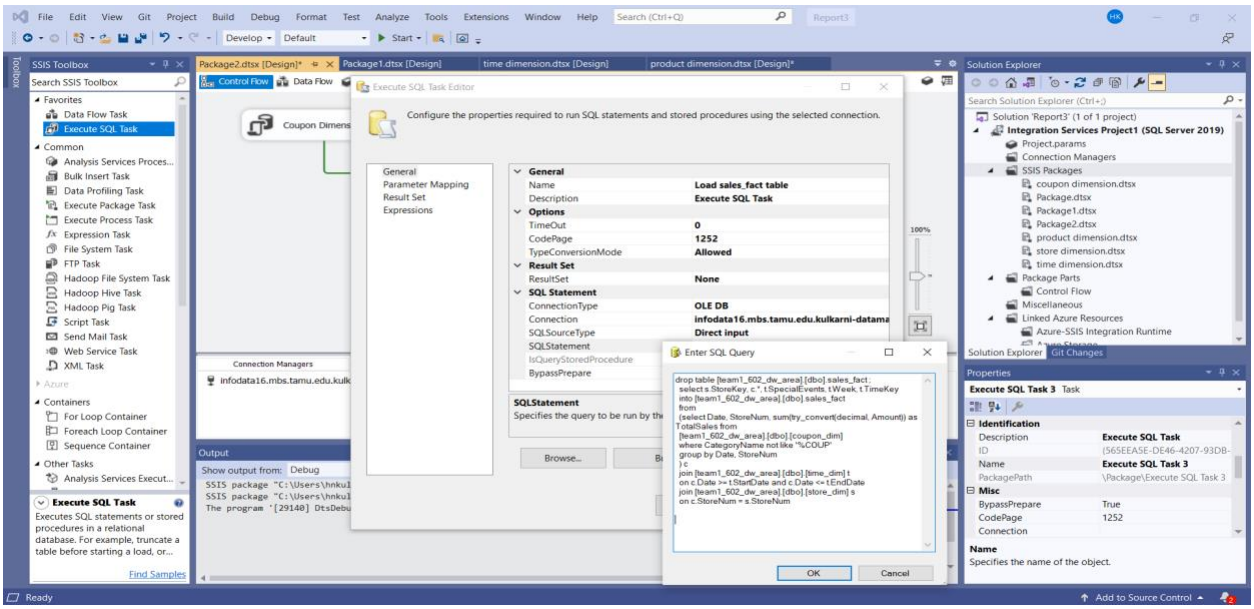



Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
Enter SQL Query
select Date, StoreNum, sum(try_convert(decimal, Amount)) as TotalSales from [team1_602_dw_area].[dbo].[coupon_dim] where CategoryName not like '%COUP' group by Date, StoreNum
OK
Cancel
Store Dimension table
Execute SQL Task

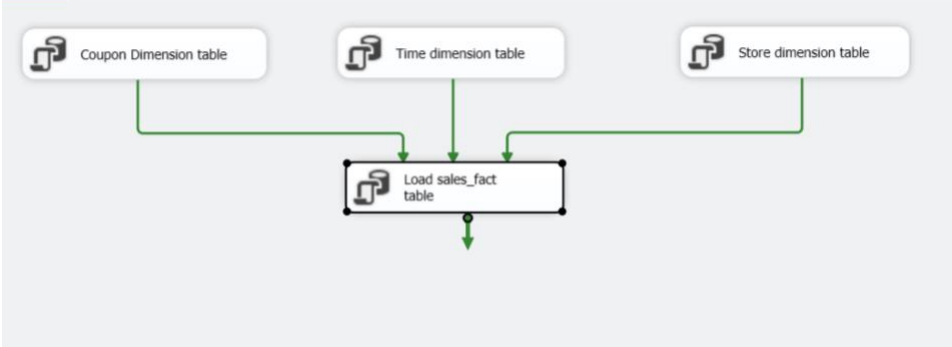

Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
Enter SQL Query
select * from [team1_602_dw_area].[dbo].[time_dim] t
OK
Cancel
Time dimension table
Execute SQL Task


Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
Name
Store dimension table
Description
Execute SQL Task
Options
TimeOut
0
CodePage
1252
TypeConversionMode
Allowed
Result Set
ResultSet
None
SQL Statement
ConnectionType
OLE DB
Connection
infodata16.mbs.tamu.edu.kulkarni-datama
SQLSourceType
Direct input
SQLStatement
select * from [team1_602_dw_area].[dbo].
IsQueryStoredProcedure
BypassPrepare
SQLStatement
Specifies the query to be run by the tas
Browse...
Enter SQL Query
select * from [team1_602_dw_area].[dbo].[store_dim] s
OK
Cancel
Connection Managers
infodata16.mbs.tamu.edu.kulk
Output
Show output from: Debug
SSIS package "C:\Users\hnkul
SSIS package "C:\Users\hnkul
The program '[29140] DtsDebu
Solution Explorer
Solution 'Report3' (1 of 1 project)
Integration Services Project1 (SQL Server 2019)
Project.params
Connection Managers
SSIS Packages
coupon dimension.dtsx
Package.dtsx
Package1.dtsx
Package2.dtsx
product dimension.dtsx
store dimension.dtsx
time dimension.dtsx
Package Parts
Control Flow
Miscellaneous
Linked Azure Resources
Azure-SSIS Integration Runtime
Properties
Coupon dimension table Task
Identification
Description
Execute SQL Task
ID
{313971FC-B5B4-49BD-A9A0-
Name
Coupon dimension table
PackagePath
\Package\Coupon dimension t
Misc
BypassPrepare
True
CodePage
1252
Connection
Name
Specifies the name of the object.
SSIS Toolbox
Execute SQL Task
Executes SQL statements or stored procedures in a relational database. For example, truncate a table before starting a load, or...
Find Samples
Ready
Add to Source Control

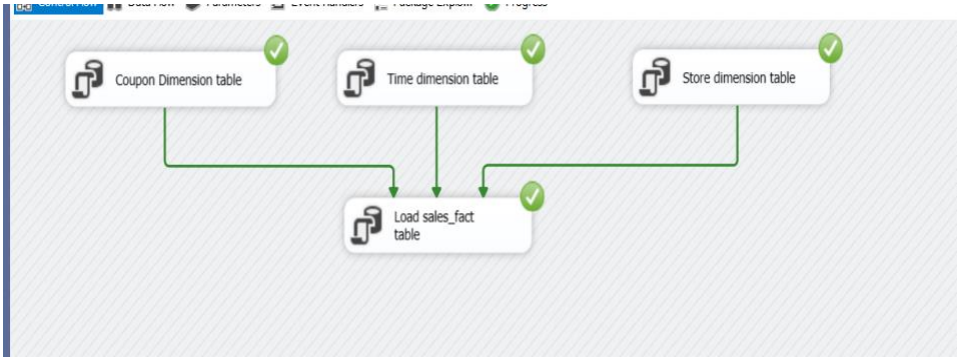

Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
Name
Coupon Dimension table
Description
Execute SQL Task
Options
TimeOut
0
CodePage
1252
TypeConversionMode
Allowed
Result Set
ResultSet
None
SQL Statement
ConnectionType
OLE DB
Connection
infodata16.mbs.tamu.edu.kulkarni-datama
SQLSourceType
Direct input
SQLStatement
select Date, StoreNum, sum(try_convert(d
IsQueryStoredProcedure
BypassPrepare
SQLStatement
Specifies the query to be run by the ta
Browse...
Enter SQL Query
select Date, StoreNum, sum(try_convert(decimal, Amount)) as TotalSales from [team1_602_dw_area].[dbo].[coupon_dim] where CategoryName not like '%COUP' group by Date, StoreNum
OK
Cancel
Connection Managers
infodata16.mbs.tamu.edu.kulk
Output
Show output from: Debug
SSIS package "C:\Users\hnkul
SSIS package "C:\Users\hnkul
The program '[29140] DtsDebu
Solution Explorer
Solution 'Report3' (1 of 1 project)
Integration Services Project1 (SQL Server 2019)
Project.params
Connection Managers
SSIS Packages
coupon dimension.dtsx
Package.dtsx
Package1.dtsx
Package2.dtsx
product dimension.dtsx
store dimension.dtsx
time dimension.dtsx
Package Parts
Control Flow
Miscellaneous
Linked Azure Resources
Azure-SSIS Integration Runtime
Properties
Store Dimension table Task
Identification
Description
Execute SQL Task
ID
{D894E3C1-67D1-4896-AA96-
Name
Store Dimension table
PackagePath
\Package\Store Dimension tat
Misc
BypassPrepare
True
CodePage
1252
Connection
infodata16.mbs.tamu.edu.kulk
Name
Specifies the name of the object.
SSIS Toolbox
Execute SQL Task
Executes SQL statements or stored procedures in a relational database. For example, truncate a table before starting a load, or...
Find Samples
Ready
Add to Source Control



-The profit_loss fact table is derived from the respective dimension tables.

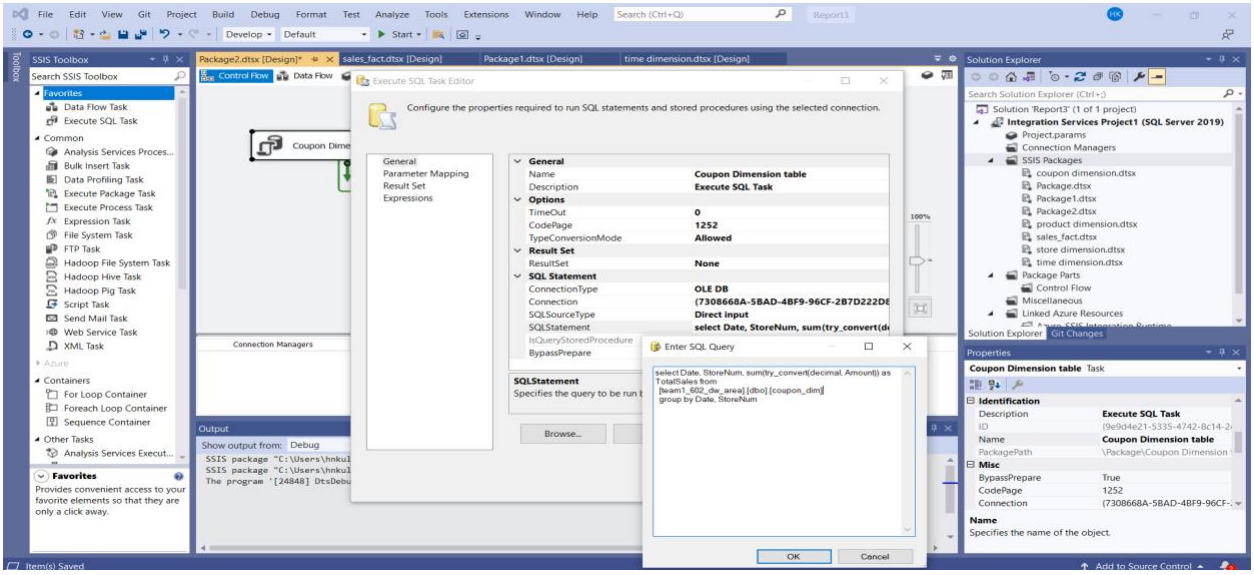


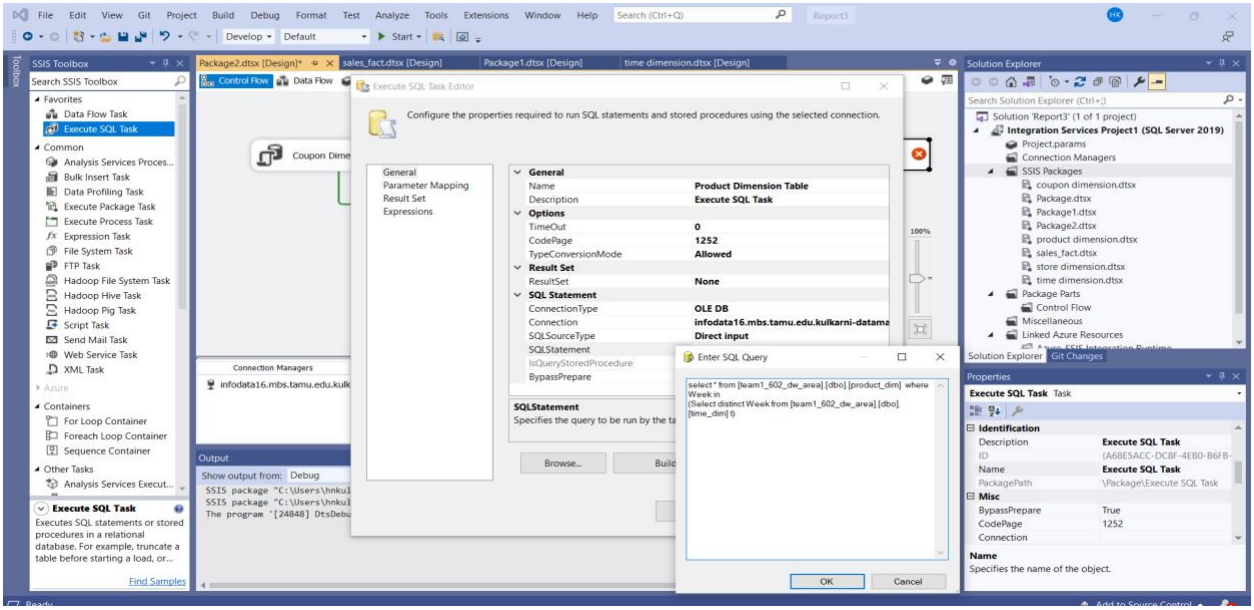
-The result we get after successfully executing the package to load the sales_fact can be seen in the below image.

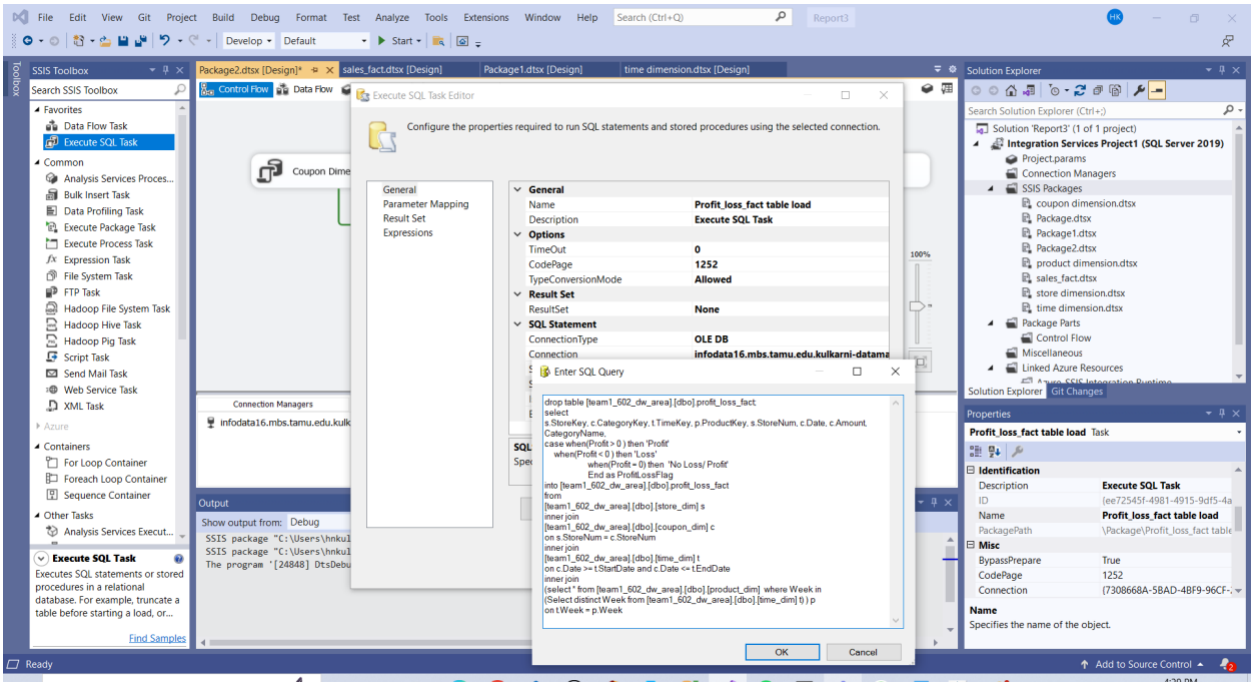


Profit_loss

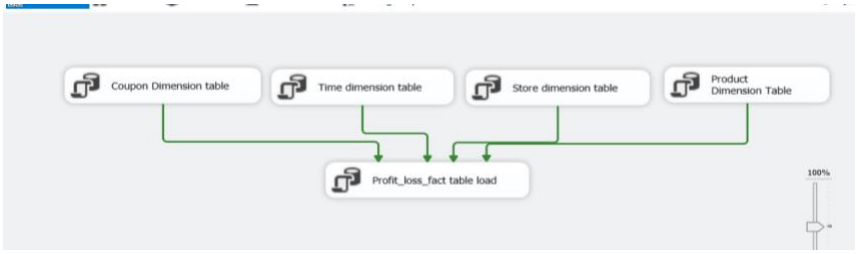
Profit_loss fact table has been derived from the 4 dimension tables that we created from the data extracted, cleaned, and transformed in the prior stages of building a data warehouse and performing ETL. The below screenshots provide a comprehensive view of queries used to create and populate the fact table from the aforementioned dimension tables (Product, Time, Coupon, Store).



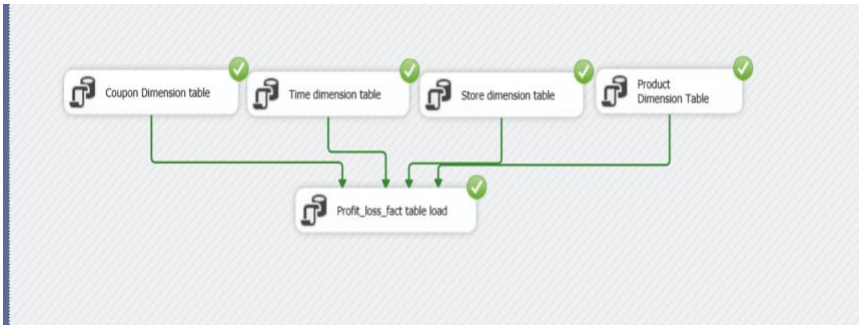




-The profit_loss fact table is derived from the respective dimension tables.



-The result we get after successfully executing the package to load the profit_loss_fact can be seen in the below image.



Data Granularity at Independent Data Mart level

The degree of specificity (or the ability to move up and down the data level) contained in the data mart based on the user's requirements is known as data granularity. The usage of summary statistics, such as monthly or yearly trends, is an option if consumers don't need to go further into the specifics. The consumers should be given access to deeper levels of data, such as weekly or daily trends, if they want to dig into additional specifics. In our case, to address the business questions, we have already kept the data at the very granular level in each data mart.

SQL Statements for ETL (SSIS)

These SQLs are executed as “Execute SQL Tasks” in **SSIS.**

Creating and inserting into store_dim

```
Creating store_dim

drop table [team1_602_dw_area].dbo.store_dim;
CREATE TABLE [team1_602_dw_area].dbo.store_dim (
  StoreKey              INT           NOT NULL    IDENTITY    PRIMARY KEY,
  StoreID     nvarchar(100)  NOT NULL ,
  StoreNum  nVARCHAR(100),
  Density  nVARCHAR(100),
  Zone  int,
  City nvarchar(100),
  Zip bigint,
Income numeric
);

insert into [team1_602_dw_area].dbo.store_dim
Select ["STORE"] as StoreID, ["STORE"] as StoreNum,
["DENSITY"] as Density,
cast(["ZONE"] as int) as Zone,
replace(["CITY"] ,'"','') as City,
["ZIP"] as Zip,
[“INCOME”] as Income
From [team1_602_staging_area].dbo.DEMO
where (["STORE"] is not null or ["STORE"] <> '.')
and ["ZONE"] <> '.'
and ["ZIP"] <> '.'
;

select * from [team1_602_dw_area].dbo.store_dim
```

Creating and inserting into product_dim

```
Creating product dim

drop table [team1_602_dw_area].dbo.product_dim;
CREATE TABLE [team1_602_dw_area].dbo.product_dim (
  ProductKey              INT           NOT NULL    IDENTITY    PRIMARY KEY,
  ProductId     nvarchar(100)  NOT NULL ,
  UPC  nVARCHAR(100),
  Description  nVARCHAR(100),
  Size  nvarchar(100),
  SKU nvarchar(100),
Week Int,
UnitsSold Int,
Quality Int,
RetailPrice decimal,
SalesCode nvarchar(100),
Profit Int,

);
```

```
drop table [team1_602_dw_area].dbo.product_dim;
Insert into  [team1_602_dw_area].dbo.product_dim
 select ProductId, mov.UPC, Description, Size
, SKU
, try_convert(int, Week) as Week
, try_convert(int, UnitsSold ) as UnitsSold
, try_convert (int, Quantity) as Quantity
, try_cast(RetailPrice as decimal) as RetailPrice
, SalesCode
, try_cast(Profit as decimal)  as Profit
from
(select
replace(["UPC"],',','') as UPC
, replace(["WEEK"],',','') as Week
, replace(["MOVE"], ',','') as UnitsSold
, replace(["QTY"],',','') as Quantity
, ["PRICE"] as RetailPrice
, replace(["SALE"],'"','') as SalesCode
, ["PROFIT"] as Profit
from [team1_602_staging_area].dbo.movement
) mov
inner join
(Select ["UPC"] as UPC, replace(["DESCRIP"], '"', '') as Description, replace(["SIZE"],'"','') as Size, ["NITEM"] as SKU,
From [team1_602_staging_area].dbo.UPC
where (["UPC"] is not null or ["UPC"] <> '.')) as p
on mov.UPC = p.UPC
```

Creating and inserting into coupon_dim

```
Creating coupon dim

drop table [team1_602_dw_area].dbo.coupon_dim;
CREATE TABLE [team1_602_dw_area].dbo.coupon_dim (
  CategoryKey             INT           NOT NULL    IDENTITY    PRIMARY KEY,
  CategoryName     nvarchar(255)  NOT NULL ,
  Date  date,
  StoreNum  NVARCHAR(100),
  Amount NVARCHAR(100),
);
```

```
insert into [team1_602_dw_area].dbo.coupon_dim
select replace(CategoryName,'"', '') as CategoryName, TRY_CONVERT(date,  Date) as Date , ["STORE"]as StoreNum ,Amount
from(
SELECT  ["STORE"]
      ,replace(["DATE"], '"','') as Date
      ,["DAIRY"] ,["FROZEN"],["BOTTLE"],["MVPCLUB"],["GROCCOUP"],["MEAT"]
      ,["MEATFROZ"],["MEATCOUP"],["FISH"],["FISHCOUP"],["PROMO"],["PROMCOUP"],["PRODUCE"],["BULK"],["SALADBAR"]
      ,["PRODCOUP"],["BULKCOUP"],["SALCOUP"],["FLORAL"],["FLORCOUP"],["DELI"],["DELISELF"]
      ,["DELIEXPR"],["CONVFOOD"],["CHEESE"],["DELICOUP"],["BAKERY"],["PHARMACY"],["PHARCOUP"] ,["GM"]
      ,["JEWELRY"],["COSMETIC"],["HABA"],["GMCOUP"],["CAMERA"],["PHOTOFIN"],["VIDEO"],["VIDEOREN"]
      ,["VIDCOUP"],["BEER"],["WINE"],["SPIRITS"],["MISCSCP"],["MANCOUP"]
      ,["CUSTCOUN"],["FTGCHIN"],["FTGCCOUP"],["FTGITAL"],["FTGICOUP"],["DAIRCOUP"],["FROZCOUP"]
      ,["HABACOUP"],["PHOTCOUP"],["COSMCOUP"],["SSDELICP"],["BAKCOUP"],["LIQCOUP"]
      from [team1_602_staging_area].[dbo].[CCOUNT]
      where ISNUMERIC(replace(["STORE"],'"',''')) = 1
      and ISNUMERIC(replace(["DATE"],'"',''')) = 1
      )p
      unpivot (Amount for CategoryName in (["DAIRY"]
      ,["FROZEN"],["BOTTLE"],["MVPCLUB"],["GROCCOUP"],["MEAT"],["MEATFROZ"]
      ,["MEATCOUP"],["FISH"],["FISHCOUP"],["PROMO"],["PROMCOUP"]
      ,["PRODUCE"],["BULK"],["SALADBAR"],["PRODCOUP"],["BULKCOUP"]
      ,["SALCOUP"],["FLORAL"],["FLORCOUP"],["DELI"],["DELISELF"]
      ,["DELIEXPR"],["CONVFOOD"],["CHEESE"],["DELICOUP"],["BAKERY"]
      ,["PHARMACY"],["PHARCOUP"],["GM"],["JEWELRY"],["COSMETIC"]
      ,["HABA"],["GMCOUP"],["CAMERA"],["PHOTOFIN"],["VIDEO"],["VIDEOREN"]
      ,["VIDCOUP"],["BEER"],["WINE"],["SPIRITS"],["MISCSCP"],["MANCOUP"]
      ,["CUSTCOUN"],["FTGCHIN"],["FTGCCOUP"],["FTGITAL"]
      ,["FTGICOUP"],["DAIRCOUP"],["FROZCOUP"],["HABACOUP"],["PHOTCOUP"]
      ,["COSMCOUP"],["SSDELICP"],["BAKCOUP"]
      ,["LIQCOUP"]) ) as unpvt
```

Creating and inserting into time_dim

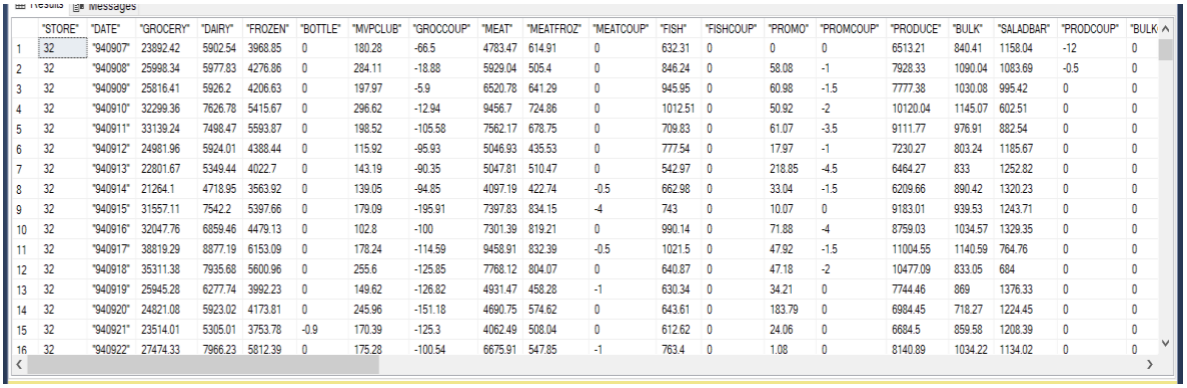
```
Creating time dimensions

drop table [team1_602_dw_area].dbo.time_dim;

CREATE TABLE [team1_602_dw_area].dbo.time_dim (
  TimeKey              INT           NOT NULL    IDENTITY    PRIMARY KEY,
  StartDate     date ,
  EndDate  date,
  SpecialEvents  NVARCHAR(100),
  Week nvarchar(100)
);

insert into [team1_602_dw_area].dbo.time_dim
select cast(StartDate as date) as StartDate,
cast(EndDate as date) as EndDate,
SpecialEvents as SpecialEvents,
WeekNo as Week
from [team1_602_staging_area].[dbo].[time]
where isnumeric(WeekNo) = 1
```

Creating sales_fact table and profit_loss fact table

```
Create sales fact table

drop table [team1_602_dw_area].[dbo].sales_fact ;
select s.StoreKey, c.*, t.SpecialEvents, t.Week, t.TimeKey
into [team1_602_dw_area].[dbo].sales_fact
from
(select Date, StoreNum, sum(try_convert(decimal, Amount)) as TotalSales from
[team1_602_dw_area].[dbo].[coupon_dim]
where CategoryName not like '%COUP'
group by Date, StoreNum
) c
join [team1_602_dw_area].[dbo].[time_dim] t
on c.Date >= t.StartDate and c.Date <= t.EndDate
join [team1_602_dw_area].[dbo].[store_dim] s
on c.StoreNum = s.StoreNum



select s.StoreKey, c.CategoryKey, t.TimeKey, p.ProductKey, s.StoreNum, c.Date, c.Amount, CategoryName
into [team1_602_dw_area].[dbo].profit_loss_fact
from
[team1_602_dw_area].[dbo].[store_dim] s
inner join
[team1_602_dw_area].[dbo].[coupon_dim] c
on s.StoreNum = c.StoreNum
inner join
[team1_602_dw_area].[dbo].[time_dim] t
on c.Date >= t.StartDate and c.Date <= t.EndDate
inner join
[team1_602_dw_area].[dbo].[product_dim] p
on t.Week = p.Week
```

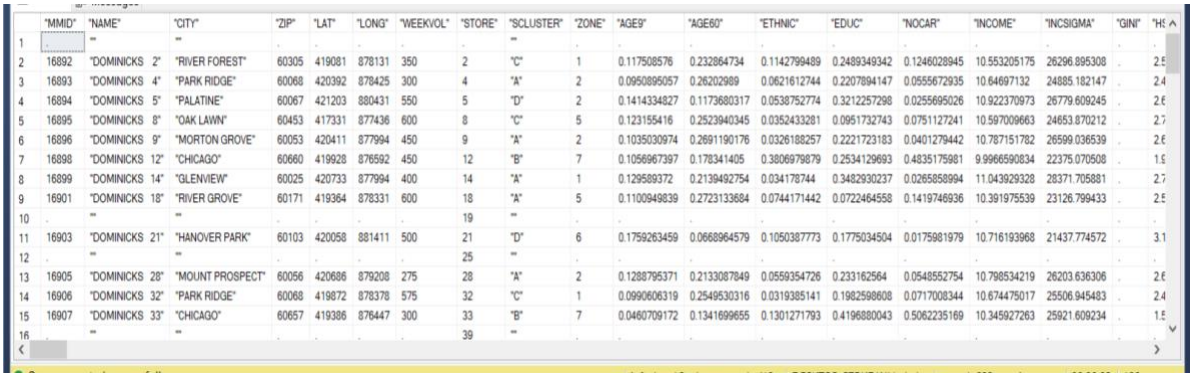
Before ETL :

- **CCount.csv**



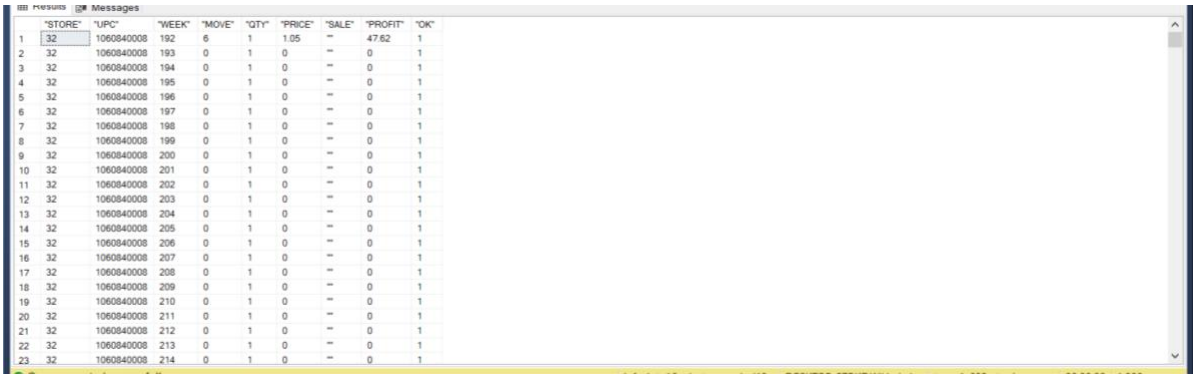
	"STORE"	"DATE"	"GROCERY"	"DAIRY"	"FROZEN"	"BOTTLE"	"MVPCLUB"	"GROCCOUP"	"MEAT"	"MEATFROZ"	"MEATCOUP"	"FISH"	"FISHCOUP"	"PROMO"	"PROMCOUP"	"PRODUCE"	"BULK"	"SALADBAR"	"PRODCOUP"	"BULK
1	32	"940907"	23892.42	5902.54	3968.85	0	180.28	-66.5	4783.47	614.91	0	632.31	0	0	0	6513.21	840.41	1158.04	-12	0
2	32	"940908"	25998.34	5977.83	4276.86	0	284.11	-18.88	5929.04	505.4	0	846.24	0	58.08	-1	7928.33	1090.04	1083.69	-0.5	0
3	32	"940909"	25816.41	5926.2	4206.63	0	197.97	-5.9	6520.78	641.29	0	945.95	0	60.98	-1.5	7777.38	1030.08	995.42	0	0
4	32	"940910"	32299.36	7626.78	5415.67	0	296.62	-12.94	9456.7	724.86	0	1012.51	0	50.92	-2	10120.04	1145.07	602.51	0	0
5	32	"940911"	33139.24	7498.47	5593.87	0	198.52	-105.58	7562.17	678.75	0	709.83	0	61.07	-3.5	9111.77	976.91	882.54	0	0
6	32	"940912"	24981.96	5924.01	4388.44	0	115.92	-95.93	5046.93	435.53	0	777.54	0	17.97	-1	7230.27	803.24	1185.67	0	0
7	32	"940913"	22801.67	5349.44	4022.7	0	143.19	-90.35	5047.81	510.47	0	542.97	0	218.85	-4.5	6464.27	833	1252.82	0	0
8	32	"940914"	21264.1	4718.95	3563.92	0	139.05	-94.85	4097.19	422.74	-0.5	662.98	0	33.04	-1.5	6209.66	890.42	1320.23	0	0
9	32	"940915"	31557.11	7542.2	5397.66	0	179.09	-195.91	7397.83	834.15	-4	743	0	10.07	0	9183.01	939.53	1243.71	0	0
10	32	"940916"	32047.76	6859.46	4479.13	0	102.8	-100	7301.39	819.21	0	990.14	0	71.88	-4	8759.03	1034.57	1329.35	0	0
11	32	"940917"	38819.29	8877.19	6153.09	0	178.24	-114.59	9458.91	832.39	-0.5	1021.5	0	47.92	-1.5	11004.55	1140.59	764.76	0	0
12	32	"940918"	35311.38	7935.68	5600.96	0	255.6	-125.85	7768.12	804.07	0	640.87	0	47.18	-2	10477.09	833.05	684	0	0
13	32	"940919"	25945.28	6277.74	3992.23	0	149.62	-126.82	4931.47	458.28	-1	630.34	0	34.21	0	7744.46	869	1376.33	0	0
14	32	"940920"	24821.08	5923.02	4173.81	0	245.96	-151.18	4690.75	574.62	0	643.61	0	183.79	0	6984.45	718.27	1224.45	0	0
15	32	"940921"	23514.01	5305.01	3753.78	-0.9	170.39	-125.3	4062.49	508.04	0	612.62	0	24.06	0	6684.5	859.58	1208.39	0	0
16	32	"940922"	27474.33	7966.23	5812.39	0	175.28	-100.54	6675.91	547.85	-1	763.4	0	1.08	0	8140.89	1034.22	1134.02	0	0

-**Demo.csv**



	"MMID"	"NAME"	"CITY"	"ZIP"	"LAT"	"LONG"	"WEEKVOL"	"STORE"	"SCLUSTER"	"ZONE"	"AGE9"	"AGE60"	"ETHNIC"	"EDUC"	"NOCAR"	"INCOME"	"INCSIGMA"	"GINI"	"HS
1		""	""	.	.	.	.	.	""	.	.	.	.	.	.	.	.	.	.
2	16892	"DOMINICKS 2"	"RIVER FOREST"	60305	419081	878131	350	2	"C"	1	0.117508576	0.232864734	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
7	16898	"DOMINICKS 12"	"CHICAGO"	60660	419928	876592	450	12	"B"	7	0.1056967397	0.178341405	0.3806979879	0.2534129693	0.4835175981	9.9966590834	22375.070508	.	1.9
8	16899	"DOMINICKS 14"	"GLENVIEW"	60025	420733	877994	400	14	"A"	1	0.129589372	0.2139492754	0.034178744	0.3482930237	0.0265858994	11.043929328	28371.705881	.	2.7
9	16901	"DOMINICKS 18"	"RIVER GROVE"	60171	419364	878331	600	18	"A"	5	0.1100949839	0.2723133684	0.0744171442	0.0722464558	0.1419746936	10.391975539	23126.799433	.	2.5
10	.	""	""	.	.	.	.	19	""	.	.	.	.	.	.	.	.	.	.
11	16903	"DOMINICKS 21"	"HANOVER PARK"	60103	420058	881411	500	21	"D"	6	0.1759263459	0.0668964579	0.1050387773	0.1775034504	0.0175981979	10.716193968	21437.774572	.	3.1
12	.	""	""	.	.	.	.	25	""	.	.	.	.	.	.	.	.	.	.
13	16905	"DOMINICKS 28"	"MOUNT PROSPECT"	60056	420686	879208	275	28	"A"	2	0.1288795371	0.2133087849	0.0559354726	0.233162564	0.0548552754	10.798534219	26203.636306	.	2.6
14	16906	"DOMINICKS 32"	"PARK RIDGE"	60068	419872	878378	575	32	"C"	1	0.0990606319	0.2549530316	0.0319385141	0.1982598608	0.0717008344	10.674475017	25506.945483	.	2.4
15	16907	"DOMINICKS 33"	"CHICAGO"	60657	419386	876447	300	33	"B"	7	0.0460709172	0.1341699655	0.1301271793	0.4196880043	0.5062235169	10.345927263	25921.609234	.	1.5
16	.	""	""	.	.	.	.	39	""	.	.	.	.	.	.	.	.	.	.

-**Movement.csv**

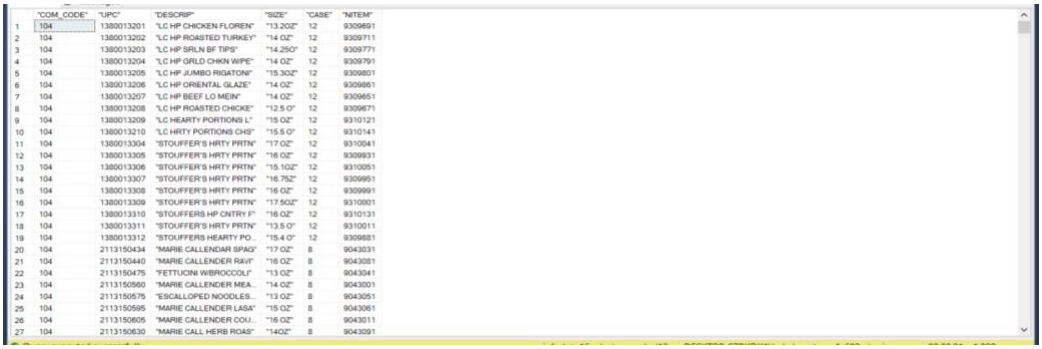


	"STORE"	"UPC"	"WEEK"	"MOVE"	"QTY"	"PRICE"	"SALE"	"PROFIT"	"OK"
1	32	1060840008	192	6	1	1.05	""	47.62	1
2	32	1060840008	193	0	1	0	""	0	1
3	32	1060840008	194	0	1	0	""	0	1
4	32	1060840008	195	0	1	0	""	0	1
5	32	1060840008	196	0	1	0	""	0	1
6	32	1060840008	197	0	1	0	""	0	1
7	32	1060840008	198	0	1	0	""	0	1
8	32	1060840008	199	0	1	0	""	0	1
9	32	1060840008	200	0	1	0	""	0	1
10	32	1060840008	201	0	1	0	""	0	1
11	32	1060840008	202	0	1	0	""	0	1
12	32	1060840008	203	0	1	0	""	0	1
13	32	1060840008	204	0	1	0	""	0	1
14	32	1060840008	205	0	1	0	""	0	1
15	32	1060840008	206	0	1	0	""	0	1
16	32	1060840008	207	0	1	0	""	0	1
17	32	1060840008	208	0	1	0	""	0	1
18	32	1060840008	209	0	1	0	""	0	1
19	32	1060840008	210	0	1	0	""	0	1
20	32	1060840008	211	0	1	0	""	0	1
21	32	1060840008	212	0	1	0	""	0	1
22	32	1060840008	213	0	1	0	""	0	1
23	32	1060840008	214	0	1	0	""	0	1

-Time.csv

	WeekNo	StartDate	EndDate	SpecialEvents
1	1	09/14/89	09/20/89	
2	2	09/21/89	09/27/89	
3	3	09/28/89	10/04/89	
4	4	10/05/89	10/11/89	
5	5	10/12/89	10/18/89	
6	6	10/19/89	10/25/89	
7	7	10/26/89	11/01/89	Halloween
8	8	11/02/89	11/08/89	
9	9	11/09/89	11/15/89	
10	10	11/16/89	11/22/89	
11	11	11/23/89	11/29/89	Thanksgiving
12	12	11/30/89	12/06/89	
13	13	12/07/89	12/13/89	
14	14	12/14/89	12/20/89	
15	15	12/21/89	12/27/89	Christmas
16				
17	16	12/28/89	01/03/90	New-Year
18	17	01/04/90	01/10/90	
19	18	01/11/90	01/17/90	
20	19	01/18/90	01/24/90	
21	20	01/25/90	01/31/90	
22	21	02/01/90	02/07/90	
23	22	02/08/90	02/14/90	
24	23	02/15/90	02/21/90	Presidents ...
25	24	02/22/90	02/28/90	
26	25	03/01/90	03/07/90	
27	26	03/08/90	03/14/90	

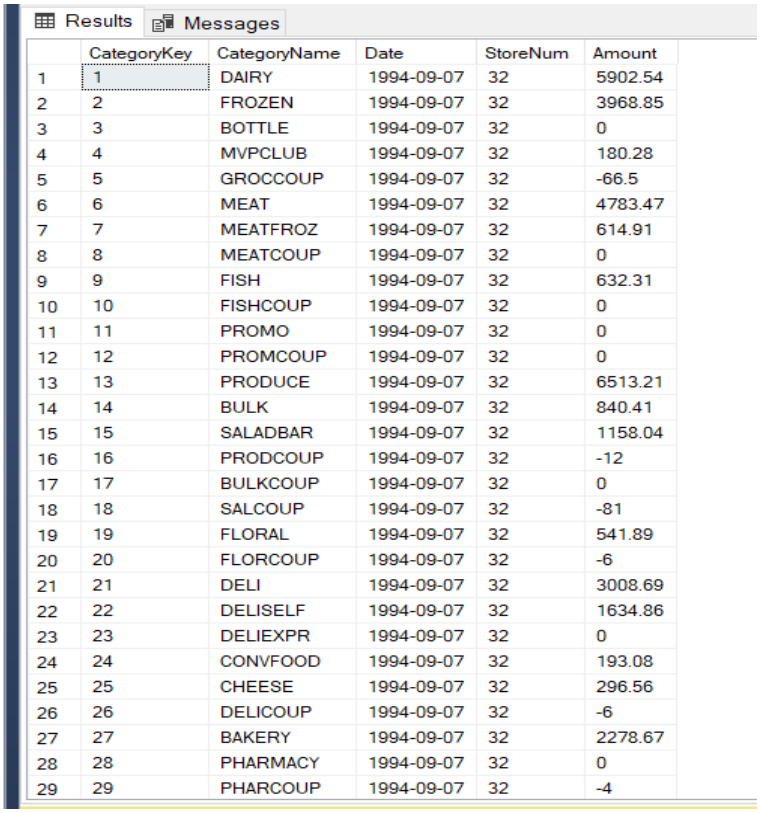
-UPC.csv



	"COM_CODE"	"UPC"	"DESCRIP"	"SIZE"	"CASE"	"NITEM"
1	104	1380013201	"LC HP CHICKEN FLOREN"	"13.2OZ"	12	9309691
2	104	1380013202	"LC HP ROASTED TURKEY"	"14 OZ"	12	9309711
3	104	1380013203	"LC HP SRLN BF TIPS"	"14.25O"	12	9309771
4	104	1380013204	"LC HP GRLD CHKN WIPE"	"14 OZ"	12	9309791
5	104	1380013205	"LC HP JUMBO RIGATONI"	"15.3OZ"	12	9309801
6	104	1380013206	"LC HP ORIENTAL GLAZE"	"14 OZ"	12	9309861
7	104	1380013207	"LC HP BEEF LO MEIN"	"14 OZ"	12	9309651
8	104	1380013208	"LC HP ROASTED CHICKE"	"12.5 O"	12	9309671
9	104	1380013209	"LC HEARTY PORTIONS L"	"15 OZ"	12	9310121
10	104	1380013210	"LC HRTY PORTIONS CHS"	"15.5 O"	12	9310141
11	104	1380013304	"STOUFFER'S HRTY PRTN"	"17 OZ"	12	9310041
12	104	1380013305	"STOUFFER'S HRTY PRTN"	"16 OZ"	12	9309931
13	104	1380013306	"STOUFFER'S HRTY PRTN"	"15.1OZ"	12	9310051
14	104	1380013307	"STOUFFER'S HRTY PRTN"	"16.75Z"	12	9309951
15	104	1380013308	"STOUFFER'S HRTY PRTN"	"16 OZ"	12	9309991
16	104	1380013309	"STOUFFER'S HRTY PRTN"	"17.5OZ"	12	9310001
17	104	1380013310	"STOUFFERS HP CNTRY F"	"16 OZ"	12	9310131
18	104	1380013311	"STOUFFER'S HRTY PRTN"	"13.5 O"	12	9310011
19	104	1380013312	"STOUFFERS HEARTY PO..."	"15.4 O"	12	9309881
20	104	2113150434	"MARIE CALLENDAR SPAG"	"17 OZ"	8	9043031
21	104	2113150440	"MARIE CALLENDER RAVI"	"16 OZ"	8	9043081
22	104	2113150475	"FETTUCINI W/BROCCOLI"	"13 OZ"	8	9043041
23	104	2113150560	"MARIE CALLENDER MEA..."	"14 OZ"	8	9043001
24	104	2113150575	"ESCALLOPED NOODLES..."	"13 OZ"	8	9043051
25	104	2113150595	"MARIE CALLENDER LASA"	"15 OZ"	8	9043061
26	104	2113150605	"MARIE CALLENDER COU..."	"16 OZ"	8	9043011
27	104	2113150630	"MARIE CALL HERB ROAS"	"14OZ"	8	9043091

After ETL :

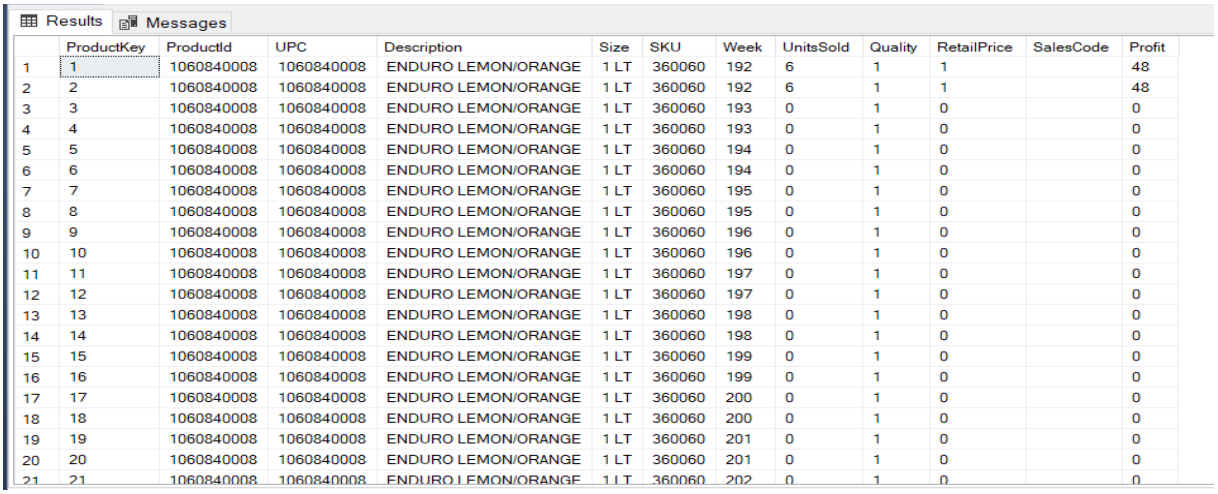
- **coupon_dim**



Results | Messages

	CategoryKey	CategoryName	Date	StoreNum	Amount
1	1	DAIRY	1994-09-07	32	5902.54
2	2	FROZEN	1994-09-07	32	3968.85
3	3	BOTTLE	1994-09-07	32	0
4	4	MVPCLUB	1994-09-07	32	180.28
5	5	GROCCOUP	1994-09-07	32	-66.5
6	6	MEAT	1994-09-07	32	4783.47
7	7	MEATFROZ	1994-09-07	32	614.91
8	8	MEATCOUP	1994-09-07	32	0
9	9	FISH	1994-09-07	32	632.31
10	10	FISHCOUP	1994-09-07	32	0
11	11	PROMO	1994-09-07	32	0
12	12	PROMCOUP	1994-09-07	32	0
13	13	PRODUCE	1994-09-07	32	6513.21
14	14	BULK	1994-09-07	32	840.41
15	15	SALADBAR	1994-09-07	32	1158.04
16	16	PRODCOUP	1994-09-07	32	-12
17	17	BULKCOUP	1994-09-07	32	0
18	18	SALCOUP	1994-09-07	32	-81
19	19	FLORAL	1994-09-07	32	541.89
20	20	FLORCOUP	1994-09-07	32	-6
21	21	DELI	1994-09-07	32	3008.69
22	22	DELISELF	1994-09-07	32	1634.86
23	23	DELIEXPR	1994-09-07	32	0
24	24	CONVFOOD	1994-09-07	32	193.08
25	25	CHEESE	1994-09-07	32	296.56
26	26	DELICOUP	1994-09-07	32	-6
27	27	BAKERY	1994-09-07	32	2278.67
28	28	PHARMACY	1994-09-07	32	0
29	29	PHARCOUP	1994-09-07	32	-4

- **product_dim**



Results | Messages

	ProductKey	ProductId	UPC	Description	Size	SKU	Week	UnitsSold	Quality	RetailPrice	SalesCode	Profit
1	1	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	192	6	1	1		48
2	2	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	192	6	1	1		48
3	3	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	193	0	1	0		0
4	4	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	193	0	1	0		0
5	5	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	194	0	1	0		0
6	6	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	194	0	1	0		0
7	7	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	195	0	1	0		0
8	8	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	195	0	1	0		0
9	9	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	196	0	1	0		0
10	10	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	196	0	1	0		0
11	11	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	197	0	1	0		0
12	12	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	197	0	1	0		0
13	13	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	198	0	1	0		0
14	14	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	198	0	1	0		0
15	15	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	199	0	1	0		0
16	16	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	199	0	1	0		0
17	17	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	200	0	1	0		0
18	18	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	200	0	1	0		0
19	19	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	201	0	1	0		0
20	20	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	201	0	1	0		0
21	21	1060840008	1060840008	ENDURO LEMON/ORANGE	1 LT	360060	202	0	1	0		0

- **store_dim**

Results Messages

	StoreKey	StoreID	StoreNum	Density	Zone	City	Zip	Income
1	1	2	2	0.0004560436	1	RIVER FOREST	60305	11
2	2	4	4	0.0005715224	2	PARK RIDGE	60068	11
3	3	5	5	0.0011192828	2	PALATINE	60067	11
4	4	8	8	0.0003905024	5	OAK LAWN	60453	11
5	5	9	9	0.0007033606	2	MORTON GROVE	60053	11
6	6	12	12	0.0001842329	7	CHICAGO	60660	10
7	7	14	14	0.0009625604	1	GLENVIEW	60025	11
8	8	18	18	0.0003854306	5	RIVER GROVE	60171	10
9	9	21	21	0.0008372262	6	HANOVER PARK	60103	11
10	10	28	28	0.0012432053	2	MOUNT PROSPECT	60056	11
11	11	32	32	0.0004560205	1	PARK RIDGE	60068	11
12	12	33	33	0.0002484629	7	CHICAGO	60657	10
13	13	40	40	0.0006887625	6	BRIDGEVIEW	60455	11
14	14	44	44	0.001115212	2	WESTERN SPRINGS	60558	11
15	15	45	45	0.0014544295	2	WHEELING	60090	11
16	16	47	47	0.0015234167	2	ADDISON	60101	11
17	17	48	48	0.001371655	2	SCHAUMBURG	60193	11
18	18	49	49	0.0016516517	2	DOWNERS GROVE	60515	11
19	19	50	50	0.0013120385	2	HICKORY HILLS	60457	11
20	20	51	51	0.0014368761	3	PALOS HEIGHTS	60463	11
21	21	52	52	0.0015827108	1	NORTHBROOK	60062	11
22	22	53	53	0.0005070444	7	CHICAGO	60662	11
23	23	54	54	0.0013467782	2	NAPERVILLE	60540	11
24	24	56	56	0.0018817307	2	COUNTRYSIDE	60525	11

- **time_dim**

	TimeKey	StartDate	EndDate	SpecialEvents	Week
1	1	1989-09-14	1989-09-20		1
2	2	1989-09-21	1989-09-27		2
3	3	1989-09-28	1989-10-04		3
4	4	1989-10-05	1989-10-11		4
5	5	1989-10-12	1989-10-18		5
6	6	1989-10-19	1989-10-25		6
7	7	1989-10-26	1989-11-01	Halloween	7
8	8	1989-11-02	1989-11-08		8
9	9	1989-11-09	1989-11-15		9
10	10	1989-11-16	1989-11-22		10
11	11	1989-11-23	1989-11-29	Thanksgiving	11
12	12	1989-11-30	1989-12-06		12
13	13	1989-12-07	1989-12-13		13
14	14	1989-12-14	1989-12-20		14
15	15	1989-12-21	1989-12-27	Christmas	15
16	16	1989-12-28	1990-01-03	New-Year	16
17	17	1990-01-04	1990-01-10		17
18	18	1990-01-11	1990-01-17		18
19	19	1990-01-18	1990-01-24		19
20	20	1990-01-25	1990-01-31		20
21	21	1990-02-01	1990-02-07		21
22	22	1990-02-08	1990-02-14		22
23	23	1990-02-15	1990-02-21	Presidents ...	23
24	24	1990-02-22	1990-02-28		24
25	25	1990-03-01	1990-03-07		25
26	26	1990-03-08	1990-03-14		26
27	27	1990-03-15	1990-03-21		27
28	28	1990-03-22	1990-03-28	Easter	28

Fact tables

- **profit_loss_fact**

	StoreKey	CategoryKey	TimeKey	ProductKey	StoreNum	Date	Amount	CategoryName	ProfitLossFlag
190	12	10641	288	16621691	33	1995-03-17	0	VIDCOUP	Profit
191	12	10641	288	17435909	33	1995-03-17	0	VIDCOUP	Profit
192	12	10641	288	16618047	33	1995-03-17	0	VIDCOUP	No Loss/ Profit
193	12	10641	288	16617547	33	1995-03-17	0	VIDCOUP	No Loss/ Profit
194	12	10641	288	18339006	33	1995-03-17	0	VIDCOUP	Profit
195	12	10641	288	15797685	33	1995-03-17	0	VIDCOUP	No Loss/ Profit
196	12	76007	322	12217242	33	1995-11-10	-13.96	DELICOUP	Profit
197	12	76007	322	14900113	33	1995-11-10	-13.96	DELICOUP	No Loss/ Profit
198	12	76007	322	14899499	33	1995-11-10	-13.96	DELICOUP	No Loss/ Profit
199	12	76007	322	8545072	33	1995-11-10	-13.96	DELICOUP	Profit
200	12	76007	322	8541976	33	1995-11-10	-13.96	DELICOUP	No Loss/ Profit
201	12	76007	322	13106738	33	1995-11-10	-13.96	DELICOUP	Profit
202	12	76007	322	8540767	33	1995-11-10	-13.96	DELICOUP	Profit
203	12	76007	322	14023306	33	1995-11-10	-13.96	DELICOUP	Profit
204	12	76007	322	11383182	33	1995-11-10	-13.96	DELICOUP	No Loss/ Profit
205	11	125	261	3101723	32	1994-09-09	60.98	PROMO	Profit
206	11	125	261	3062045	32	1994-09-09	60.98	PROMO	Profit
207	11	125	261	3059687	32	1994-09-09	60.98	PROMO	Profit
208	11	125	261	3078061	32	1994-09-09	60.98	PROMO	Profit
209	11	125	261	3076489	32	1994-09-09	60.98	PROMO	Profit
210	11	125	261	2930251	32	1994-09-09	60.98	PROMO	Profit
211	11	125	261	3074131	32	1994-09-09	60.98	PROMO	Profit
212	11	125	261	2996073	32	1994-09-09	60.98	PROMO	Profit
213	11	125	261	2992619	32	1994-09-09	60.98	PROMO	Profit

- **sales_fact**

	StoreKey	Date	StoreNum	TotalSales	SpecialEvents	Week	TimeKey
539	69	1989-10-29	115	58109	Halloween	7	7
540	67	1989-10-29	113	44959	Halloween	7	7
541	25	1989-10-29	59	37829	Halloween	7	7
542	35	1989-10-29	74	48351	Halloween	7	7
543	30	1989-10-29	68	31017	Halloween	7	7
544	23	1989-10-29	54	29501	Halloween	7	7
545	38	1989-10-29	77	37271	Halloween	7	7
546	33	1989-10-29	72	27903	Halloween	7	7
547	46	1989-10-29	89	32313	Halloween	7	7
548	70	1989-10-29	116	32046	Halloween	7	7
549	37	1989-10-29	76	52165	Halloween	7	7
550	22	1989-10-29	53	24478	Halloween	7	7
551	7	1989-10-29	14	26262	Halloween	7	7
552	31	1989-10-29	70	45766	Halloween	7	7
553	1	1989-10-29	2	31654	Halloween	7	7
554	60	1989-10-29	105	66667	Halloween	7	7
555	34	1989-10-29	73	49232	Halloween	7	7
556	27	1989-10-29	64	25492	Halloween	7	7
557	15	1989-10-29	45	30850	Halloween	7	7
558	9	1989-10-29	21	37390	Halloween	7	7
559	64	1989-10-29	110	50923	Halloween	7	7
560	58	1989-10-29	103	50128	Halloween	7	7
561	79	1989-10-29	128	74649	Halloween	7	7
562	75	1989-10-29	122	85349	Halloween	7	7
563	52	1989-10-29	95	26503	Halloween	7	7
564	59	1989-10-29	104	40414	Halloween	7	7
565	82	1989-10-29	131	63179	Halloween	7	7

BI Reporting

Reporting Plan

Every corporation needs to be exceptionally proficient in business intelligence. Business intelligence is the collection of tactics and procedures used by an organization to look for trends in its data using reports, infographics, data analysis, and principles from data warehousing. It is crucially important for organizations to examine how their operations have fared over time in the past, and present, and even estimate how they may fare in the near future. Organizations can use BI technologies to help them make decisions based on analyses, reports, and visualizations that are provided by these tools.

Since Dominick's Finer Foods heavily relies on data to understand how their operations have produced or not produced business results and to continue improving their strategies and procedures in so as to bring in more client base, visualizations and reports are extremely important in this specific scenario for DFF to evaluate their information to figure any particular trends or to forecasting potential business measures. Reports will assist DFF in realizing their current position in the business world and in understanding the facts more thoroughly with the use of graphics for each business topic.

SSAS, SSRS, and Tableau were used to create reports and visualizations for the data related to the appropriate Business Questions. The implementation tool for BI utilized for the questions is summarized in the table below.

Question #	Business Questions	DW BI Tool
1	**What is the percentage change in net sales quarter over quarter for the period 1990-96?**	Tableau
2	**What is the percentage change in net sales for all holiday weeks year over year from 1990-96?**	Tableau
3	**Which zone generated maximum profits, and what were the income ranges of the people in these zone markets?**	SSRS
4	**Which category of coupons reduced profits?**	SSAS
5	**Which regions are densely populated, and what are the profits generated by these regions?**	SSAS + SSRS

a. Determining all target reports that satisfy business questions

BQ1. : What is the percentage change in net sales quarter over quarter for the period 1990-96?

Report will be generated using Tableau. To generate this report, the Time dimension and the Profit_Loss_fact table from the data warehouse have to be used. The report will contain a Date field which will be derived from the StaeDate and the EndDate fields from the Time Dimension table. The sales figures which are at the week level of granularity will have to be aggregated to the quarter level to generate the desired report. The sales figures will be derived from the Amount field of the Profit_Loss_fact table.

BQ3. : What is the percentage change in net sales for all holiday weeks year over year from 1990-96?

The report will be generated using Tableau. To generate this report, the Time Dimension and the Profit_Loss_Fact table from the data warehouse have to be used. The report will contain a Date field which will be selected using the StartDate, EndDate, and SpecialEvents fields from the Time Dimension. The weekly sales figures will have to be filtered for special events and then aggregated to the year level for this report. The sales figures will be derived from the Amount field of the Profit_Loss_Fact table.

BQ4. : Which zone generated maximum profits, and what were the income ranges of the people in these zone markets?

Report generated from SSRS alone.

To visualize this report we used sales_fact table and store dimension table. As we required income ranges and zone values, these were retrieved from the store dimension table whereas the actual profit amount was fetched from the sales fact table. The joins between the tables were performed using surrogate keys. Here we used only SSRS to generate this report. The report focuses on the top 3 zoes and income ranges that have maximum profits.

BQ5. : Which category of coupons reduced profits?

Report generated from SSAS alone.

This report was built using the profit_loss_fact fact table and the coupon and store dimension table. Cubes were built using SSAS. The coupon dimension table was used to calculate the profits that were generated for the different coupon categories. The joins were performed on the surrogate keys and the aggregate on the amount values.

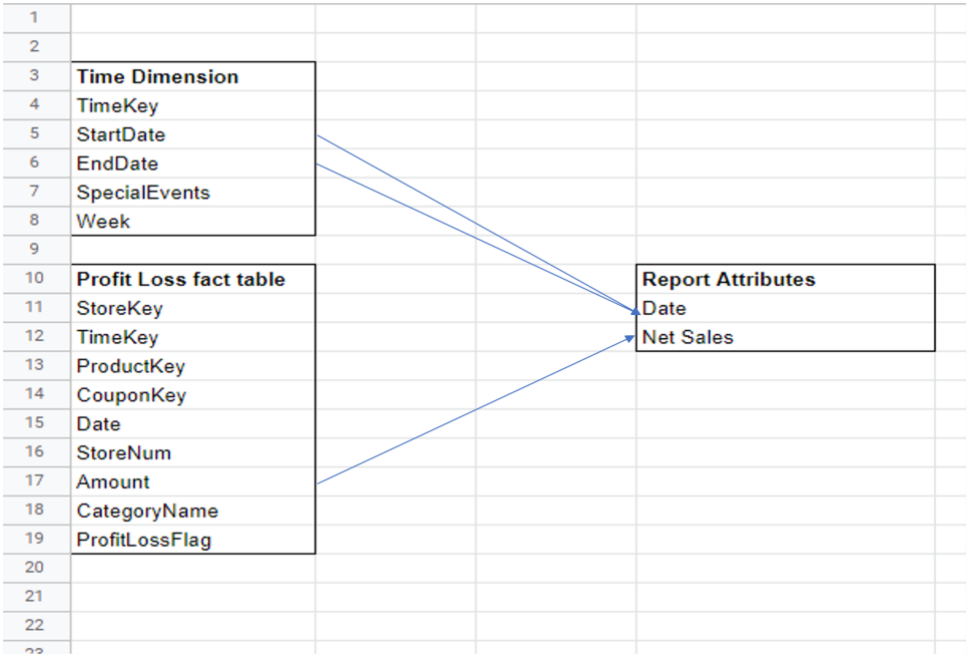
BQ6. : Which regions are densely populated, and what are the profits generated by these regions?

Report generated from SSAS and SSRS.

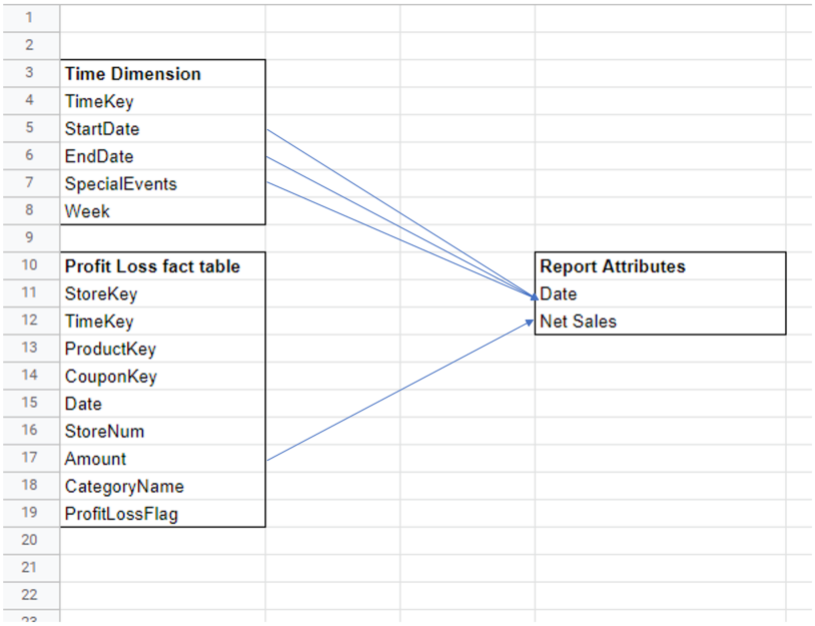
The store dimension table and the profit_loss fact table were used to construct this report. SSAS was utilized to create cubes. The store dimension was used to calculate the profits . Inner join was performed on surrogate keys and aggregate.

b. Mappings from the tables in the Independent data marts to the report attributes

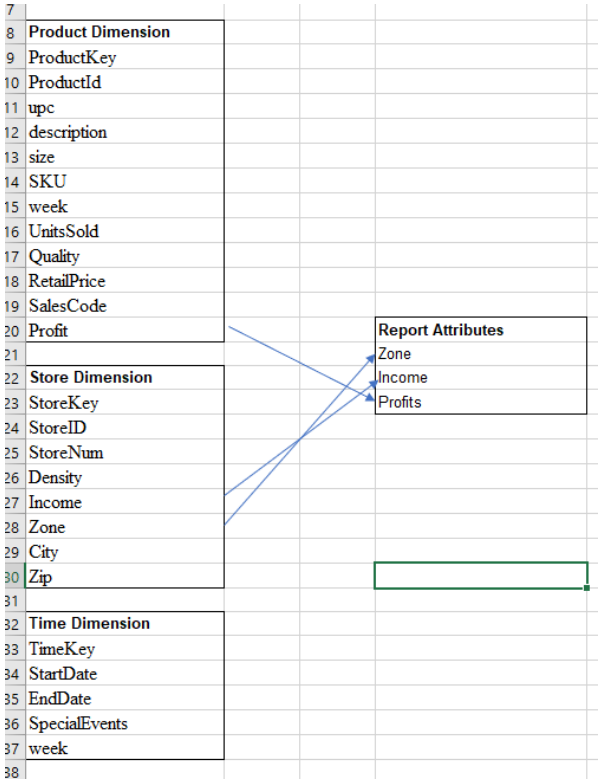
BQ1. : What is the percentage change in net sales quarter over quarter for the period 1990-96?



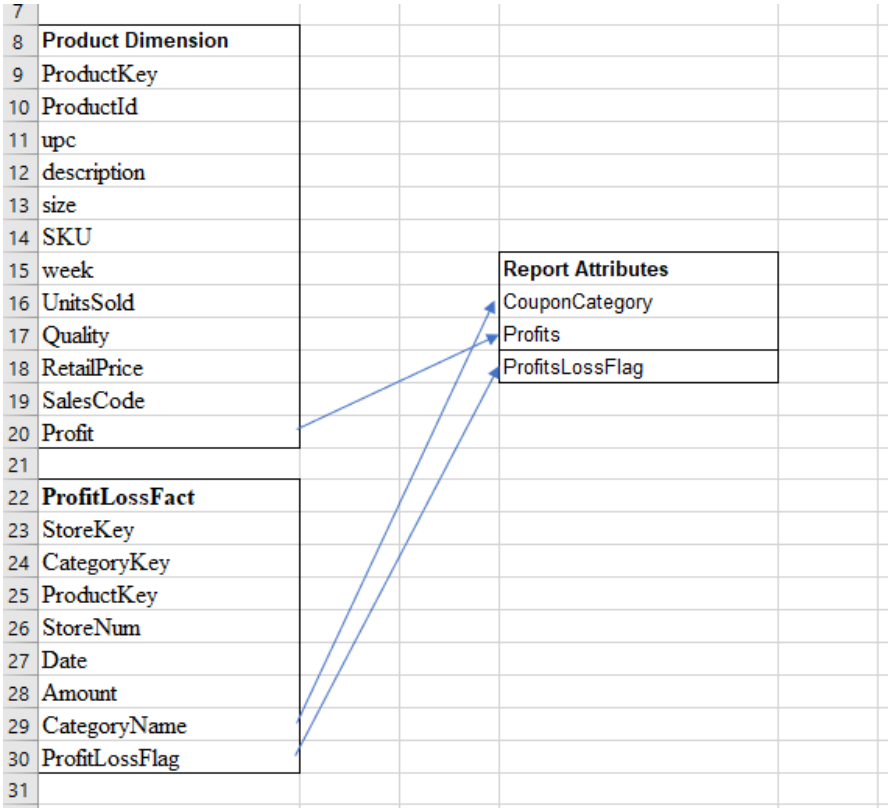
BQ3. : What is the percentage change in net sales for all holiday weeks year over year from 1990-96?



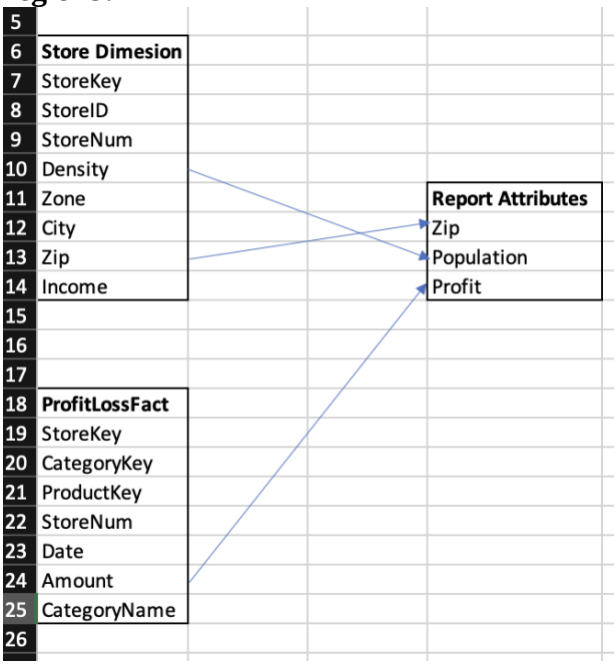
BQ4. : Which zone generated maximum profits, and what were the income ranges of the people in these zone markets?



BQ5.: Which category of coupons reduced profits?



BQ6. : Which regions are densely populated, and what are the profits generated by these regions?



c. Report Templates description

BQ1. : What is the percentage change in net sales quarter over quarter for the period 1990-96?

Years and Quarters		Percentage Diff. of Net Sales Q-Q
1990		
	Q1	x%
	Q2	y%
	Q3	z%
	Q4	.....
1991		
.....	....	.....
.....	....	.....
.....	....	.....

The base table for the visualization that will be created using Tableau will contain two columns as described above. The field on the left lists the years and quarters. The field on the right will contain a percentage difference of the Net Sales amounts for the current quarter as compared to the previous quarter.

BQ3. : What is the percentage change in net sales for all holiday weeks year over year from 1990-96?

Years	Percentage Diff. of Net Sales for holiday weeks Y-Y
1990	x%
1991	y%
1992	z%
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....

The base table for the visualization that will be created using Tableau can be seen above. It will contain two columns. The field on the left will contain the years. The field on the right will contain a percentage difference of the Net Sales amounts for just the holiday weeks aggregated by year. The percentage difference is for the amount corresponding to the current year as compared to the previous year.

BQ4. : Which zone generated maximum profits, and what were the income ranges of the people in these zone markets?

Zone	Income	Profit
1	x	x
2	y	y
3	z	z
4	..	
5		

This report is built using SSRS and shows the profit and income ranges of the different zones. The income and profit values are aggregated at the zone level. These values will help in determining which zone should be given greater focus.

BQ5. : Which category of coupons reduced profits?

CategoryName	Profit	ProfitLossFlag
x	xx.yy	Y/N
y	xx.yy	Y/N
z	xx.yy	Y/N

The report for this business question will be created using SSAS. It shows the different categories of coupons that were issued by DFF and the total increase and decrease in profits due to these coupons. There is an additional column to denote if the coupon resulted in profit or loss.

BQ6. : Which regions are densely populated, and what are the profits generated by these regions?

Zip	Population	Profit
1	x	x
2	y	y
3	z	z
4	..	...
5		
...		

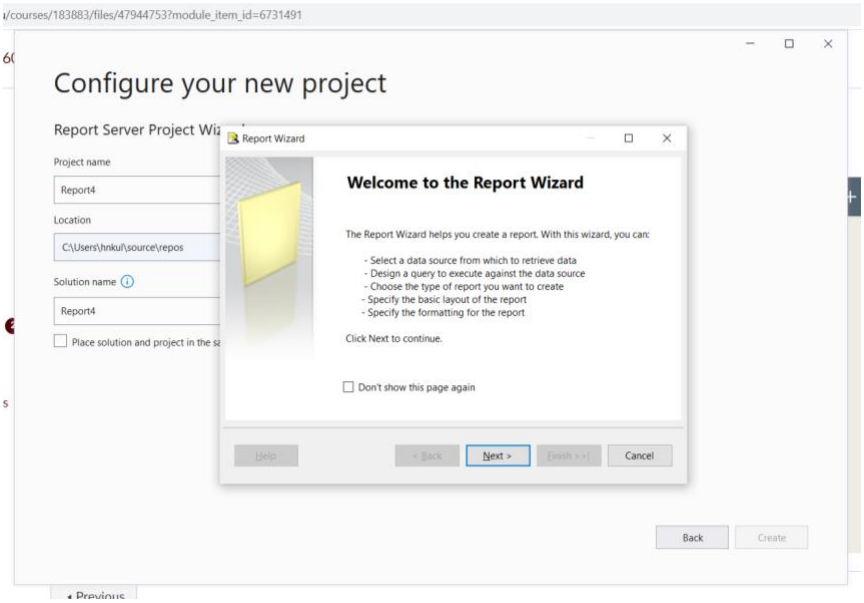
The report for this business question will be created using SSAS and SSRS. It shows the population of different regions(zip code) and the profits generated by these regions.

Reporting implementation

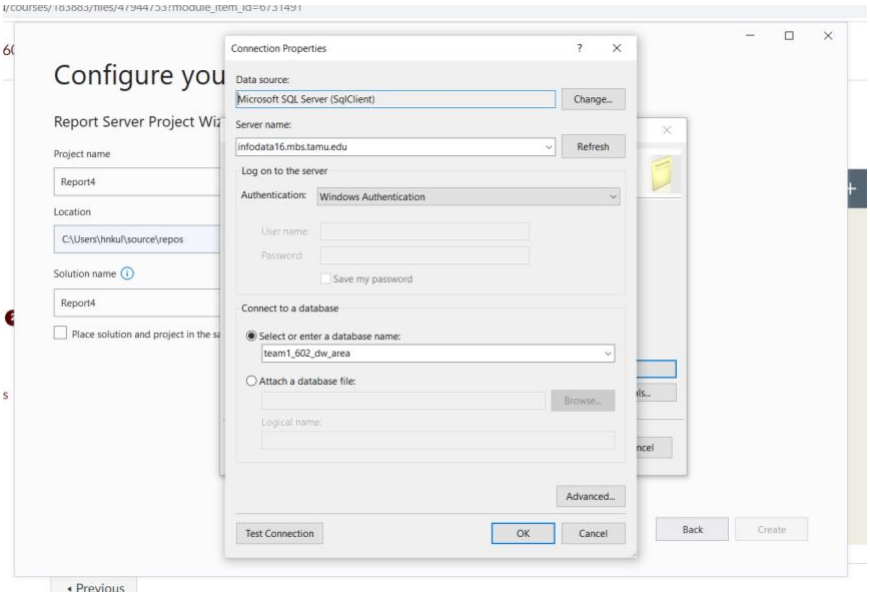
a. Reports using SSRS

BQ4 - Which zone generated maximum profits, and what were the income ranges of the people in these zone markets?

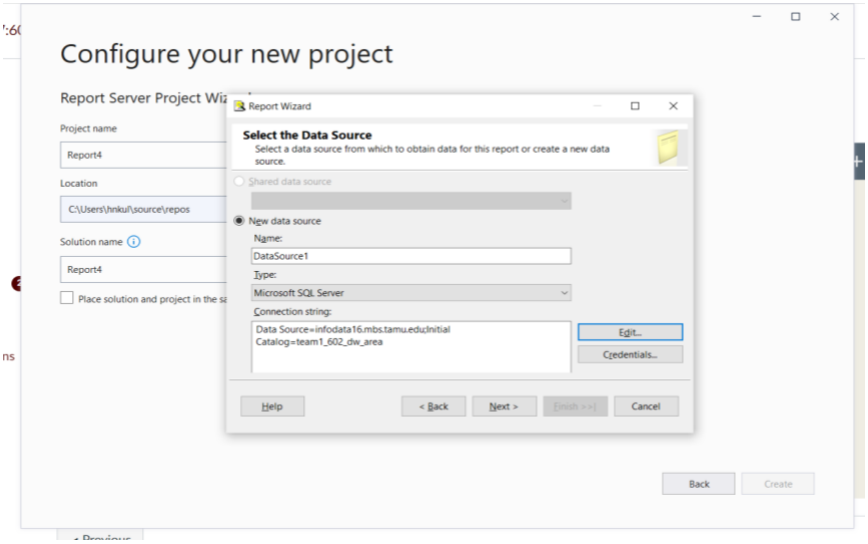
-We open a report wizard and follow the steps to select the data source and go about developing the report.



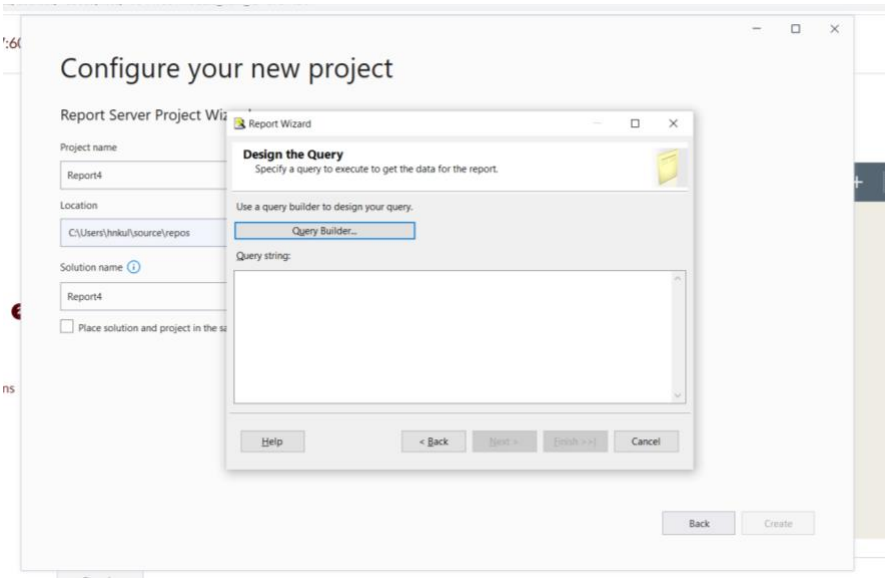
-We define the data source and server name and select the database.



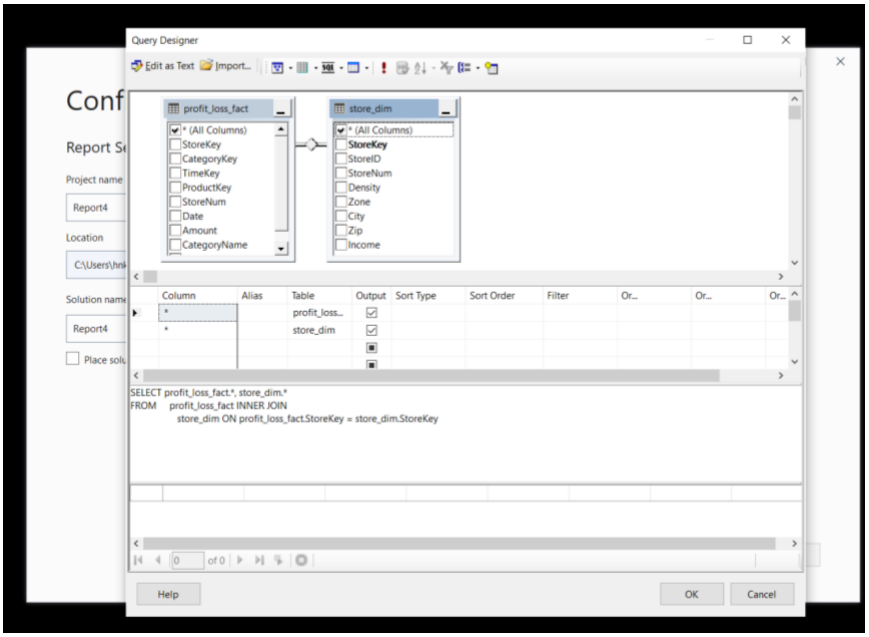
- Here we select the data source and insert the connection string.



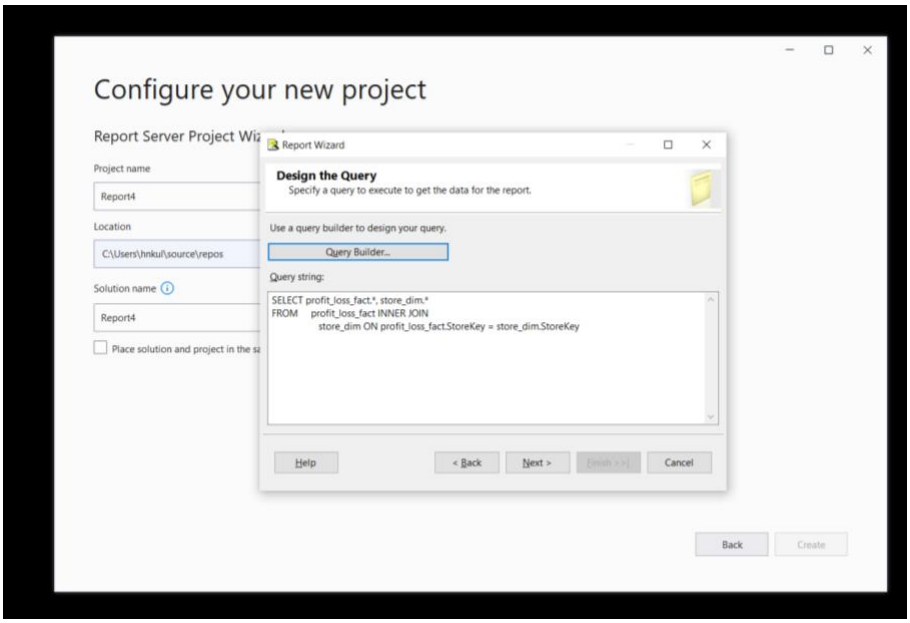
-We now design the query using a query builder.



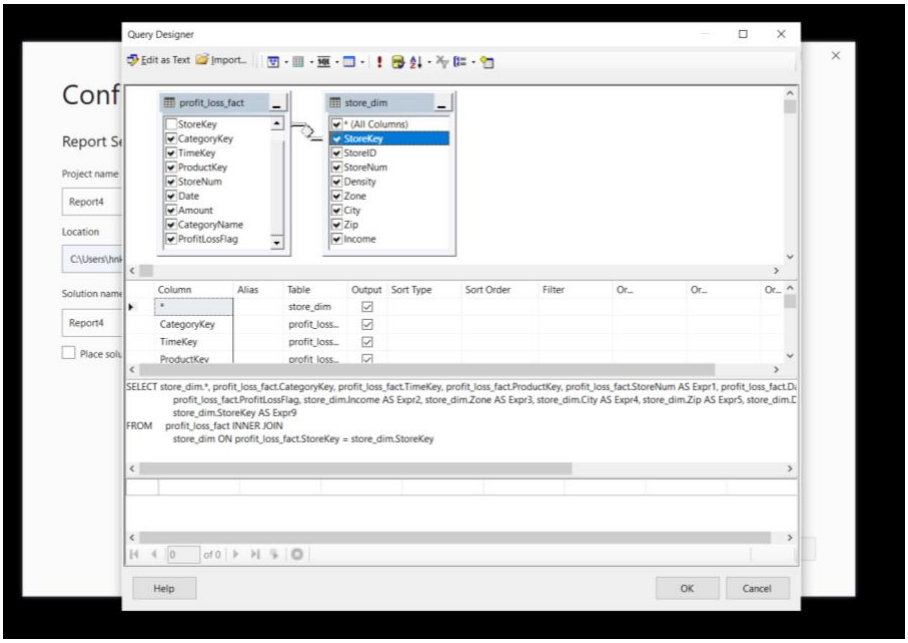
-Using the query designer, we get the following query :



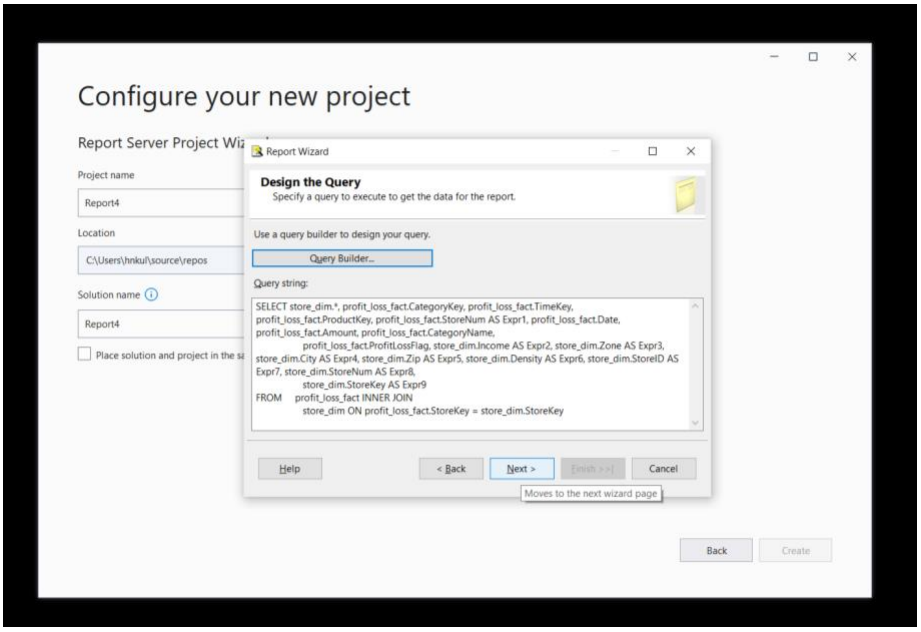
-Now we have the query in the query builder for further execution.



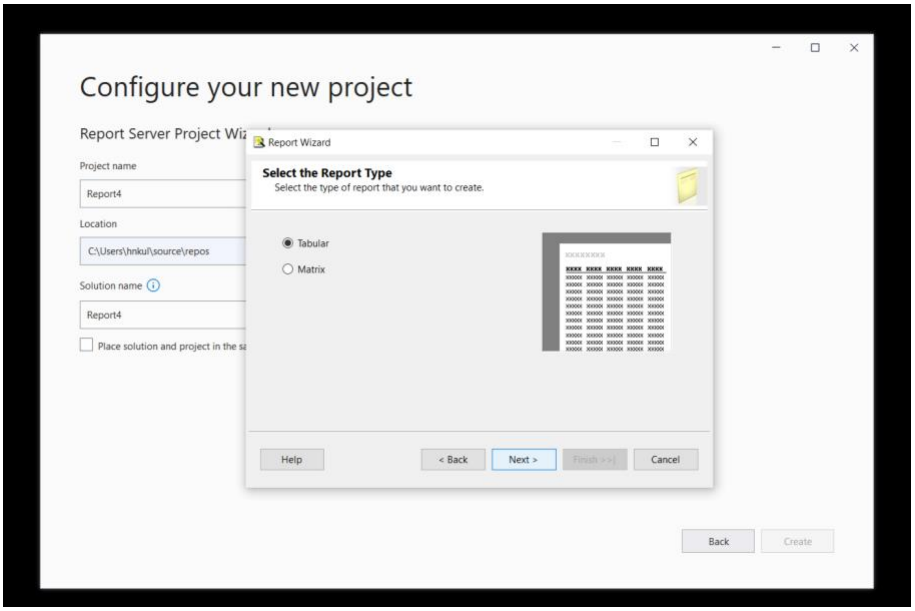
-As in the query, we have used the INNER Join on the store dimension table and profit_loss fact table.



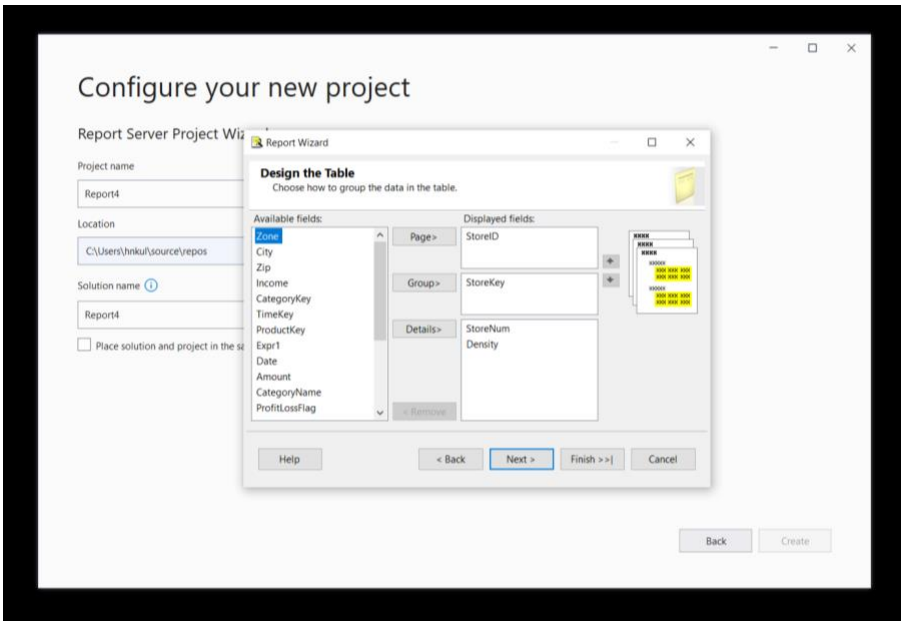
-We design the query in the query builder.



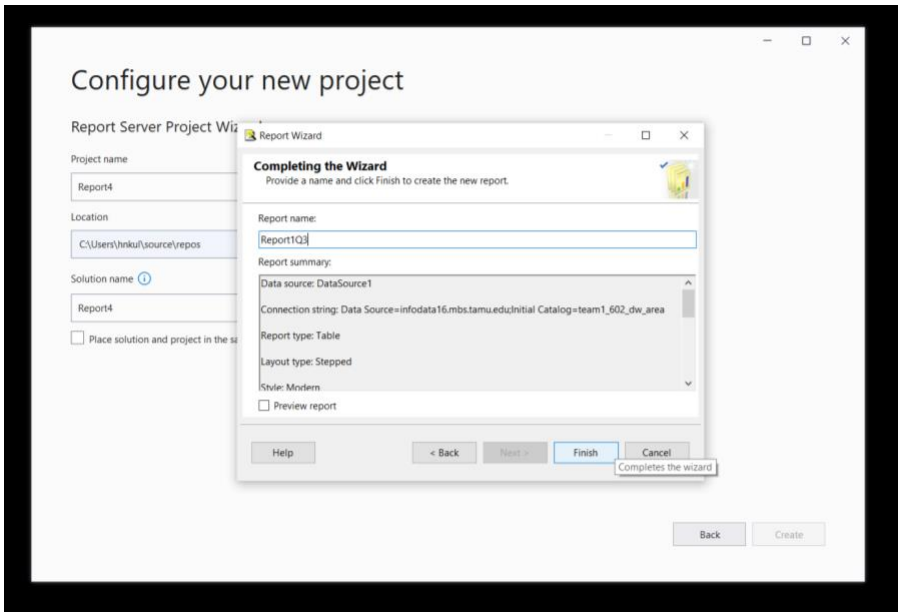
-We select the report type as ‘Tabular’



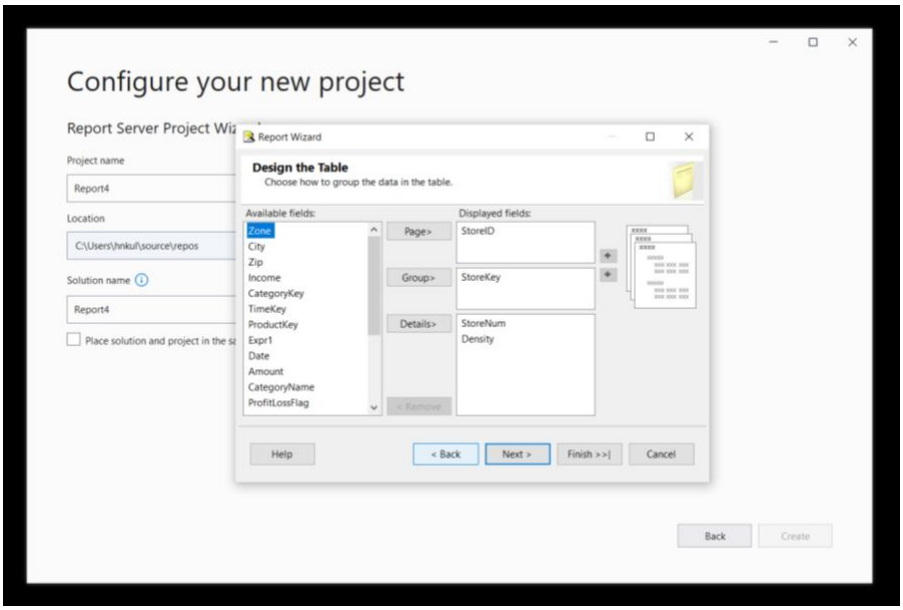
-In the following prompt, the data is grouped in the table.

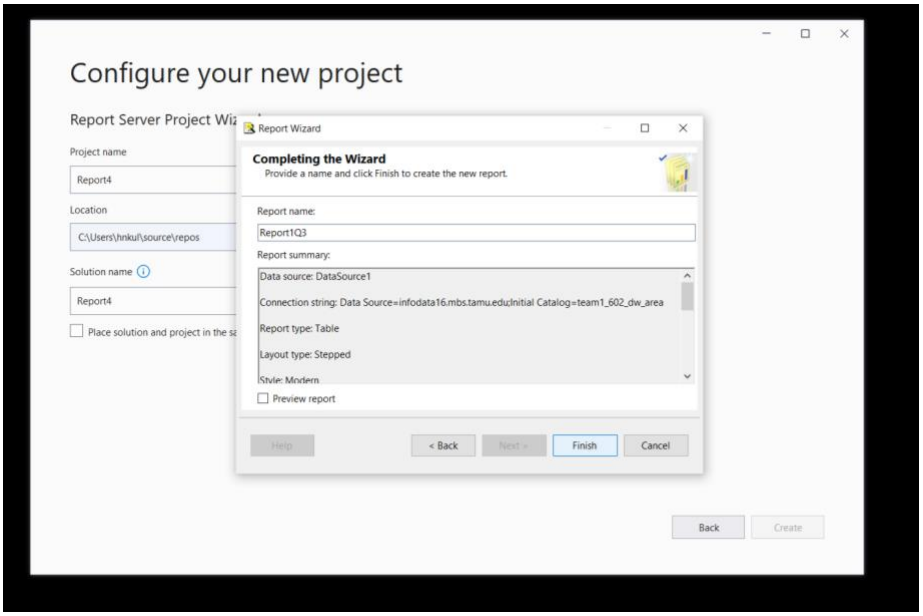


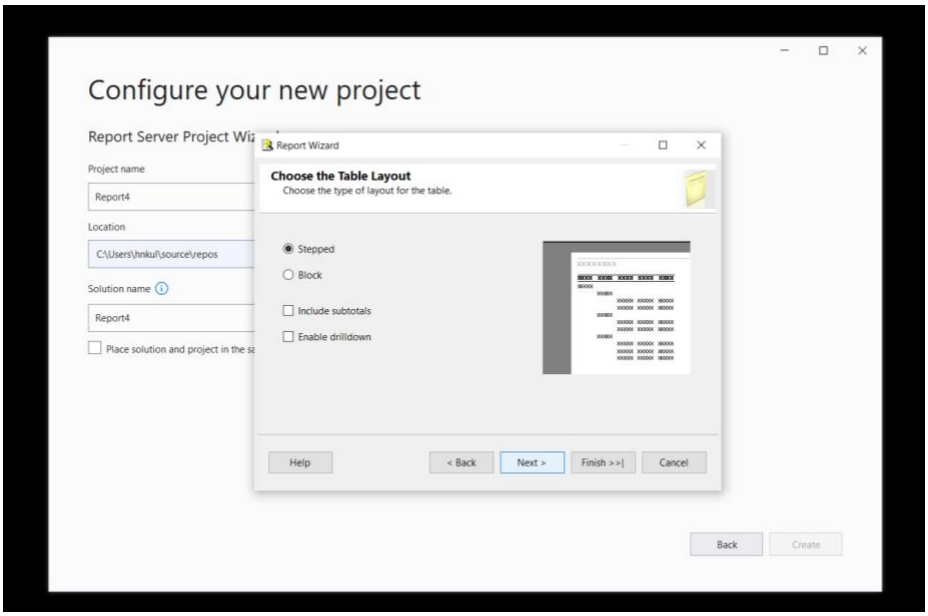
-Here, we name the report, and provide the report summary

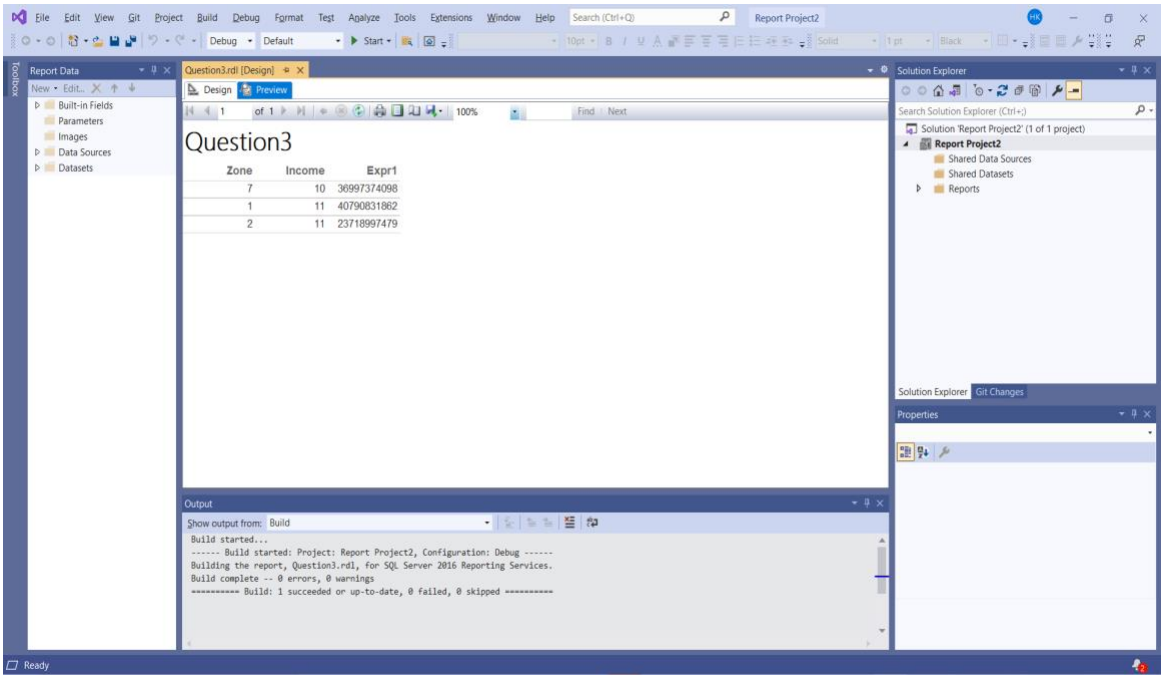


-Table design can be used to group the data.

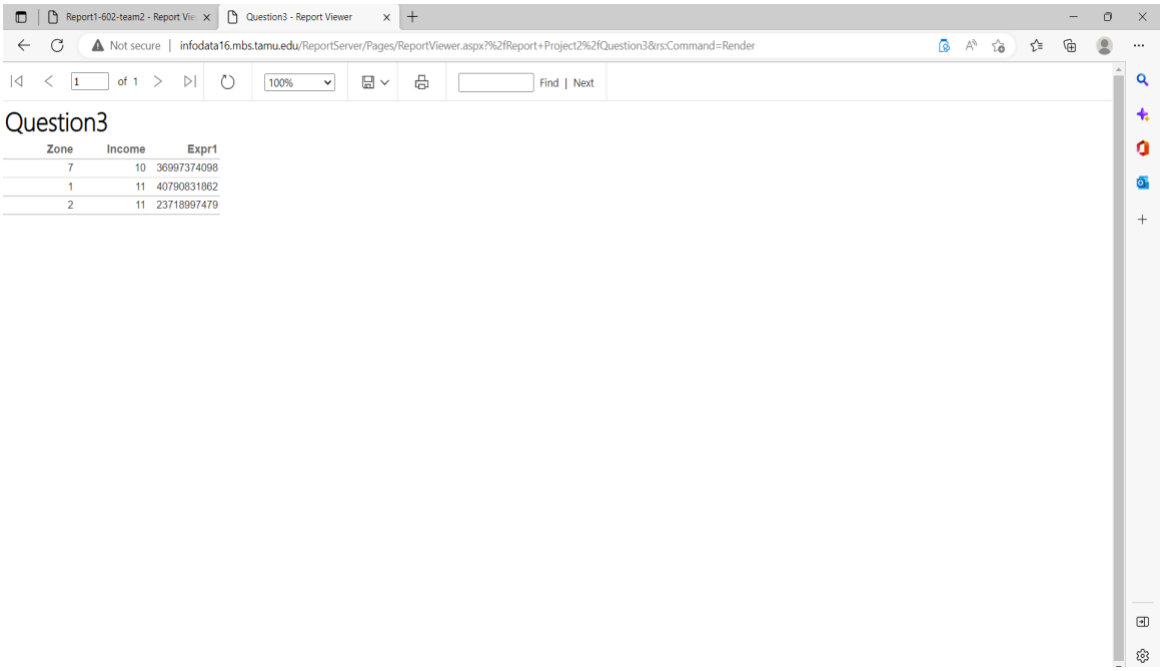





Configure your new project
Report Server Project Wiz
Project name
Report4
Location
C:\Users\hnkul\source\repos
Solution name
Report4
Place solution and project in the sa
Report Wizard
Choose the Table Layout
Choose the type of layout for the table.
Stepped
Block
Include subtotals
Enable drilldown
Help
< Back
Next >
Finish >>|
Cancel
Back
Create


File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help
Report Project2
Debug
Default
Start
Report Data
New
Edit...
Built-in Fields
Parameters
Images
Data Sources
Datasets
Question3.rdl [Design]
Design
Preview
Question3
Zone Income Expr1
7 10 36997374098
1 11 40790831862
2 11 23718997479
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Report Project2' (1 of 1 project)
Report Project2
Shared Data Sources
Shared Datasets
Reports
Solution Explorer
Git Changes
Properties
Output
Show output from: Build
Build started...
------ Build started: Project: Report Project2, Configuration: Debug ------
Building the report, Question3.rdl, for SQL Server 2016 Reporting Services.
Build complete -- 0 errors, 0 warnings
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
Ready

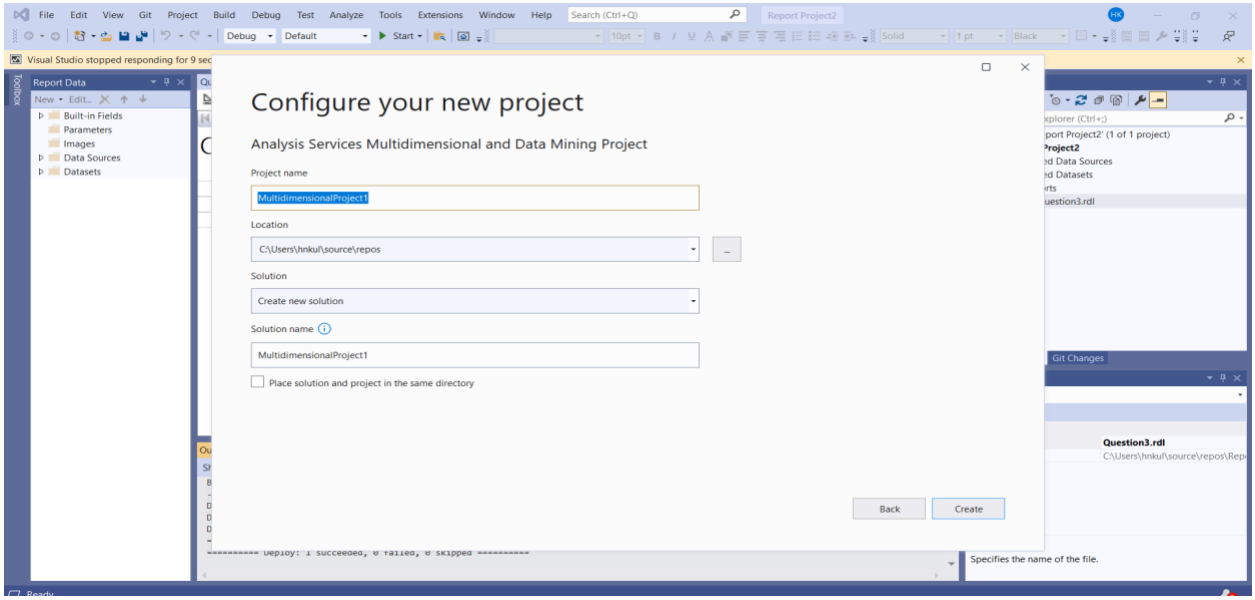
-Here’s the BI reporting or visualization of the report



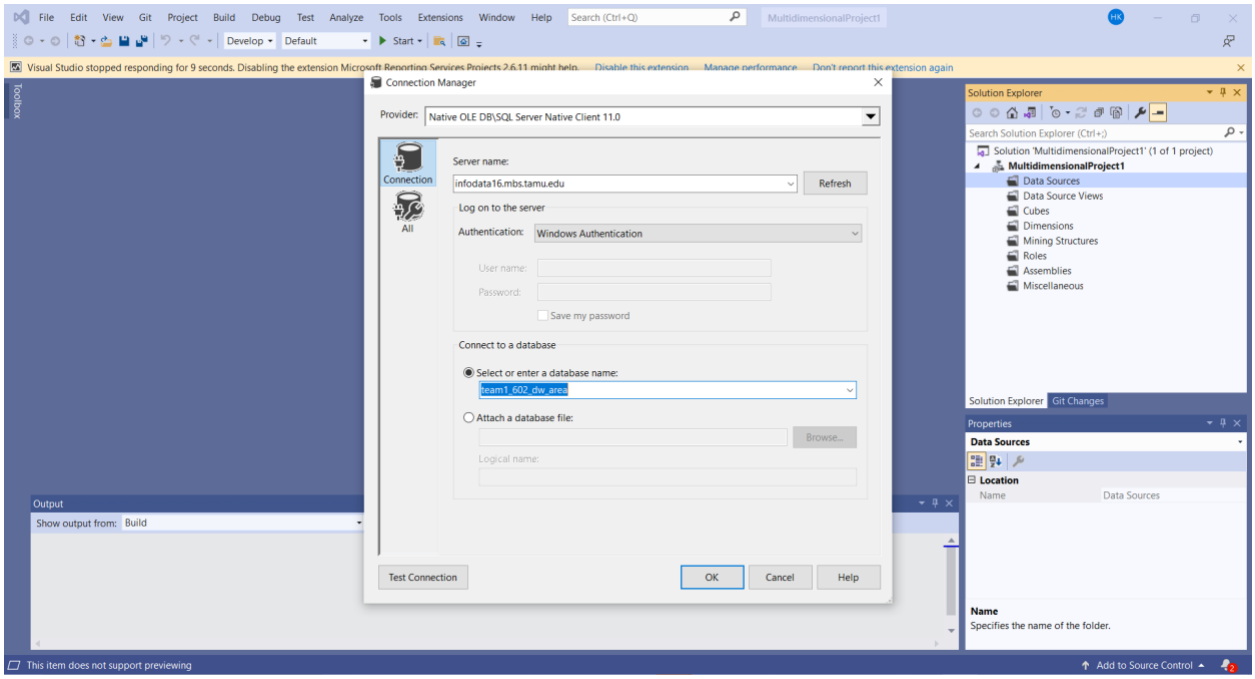
b. Reports using SSAS

BQ5 - Which category of coupons reduced profits?

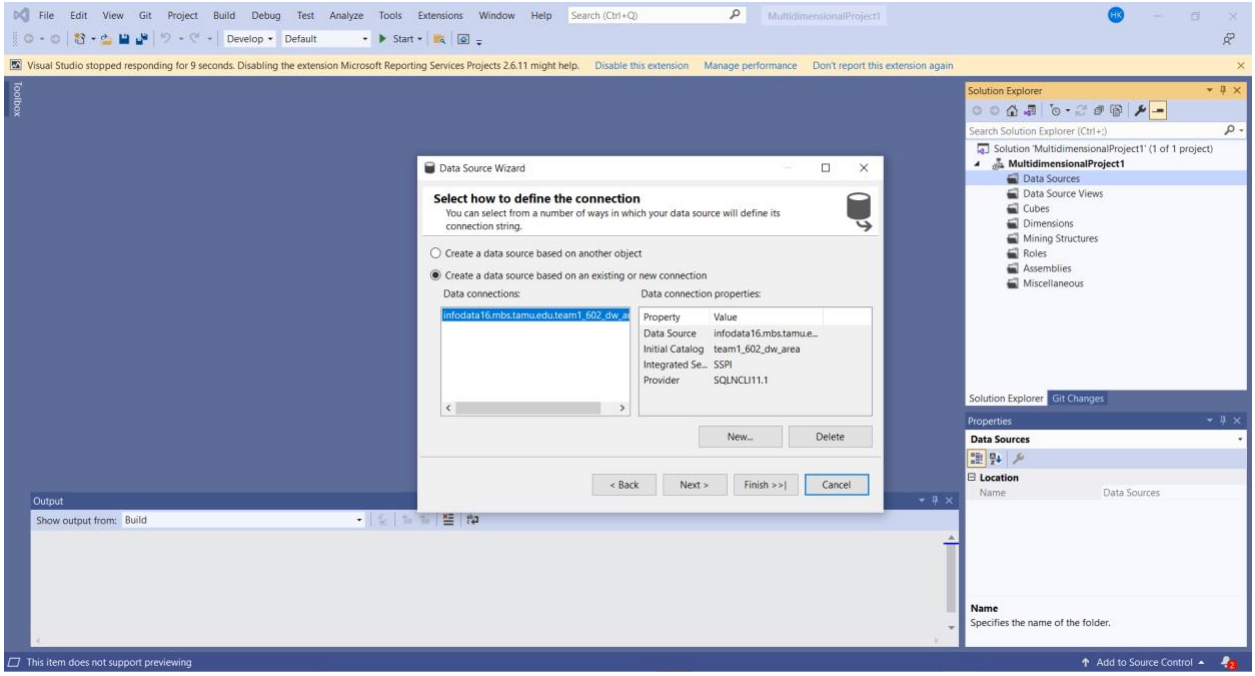
-Configure the project



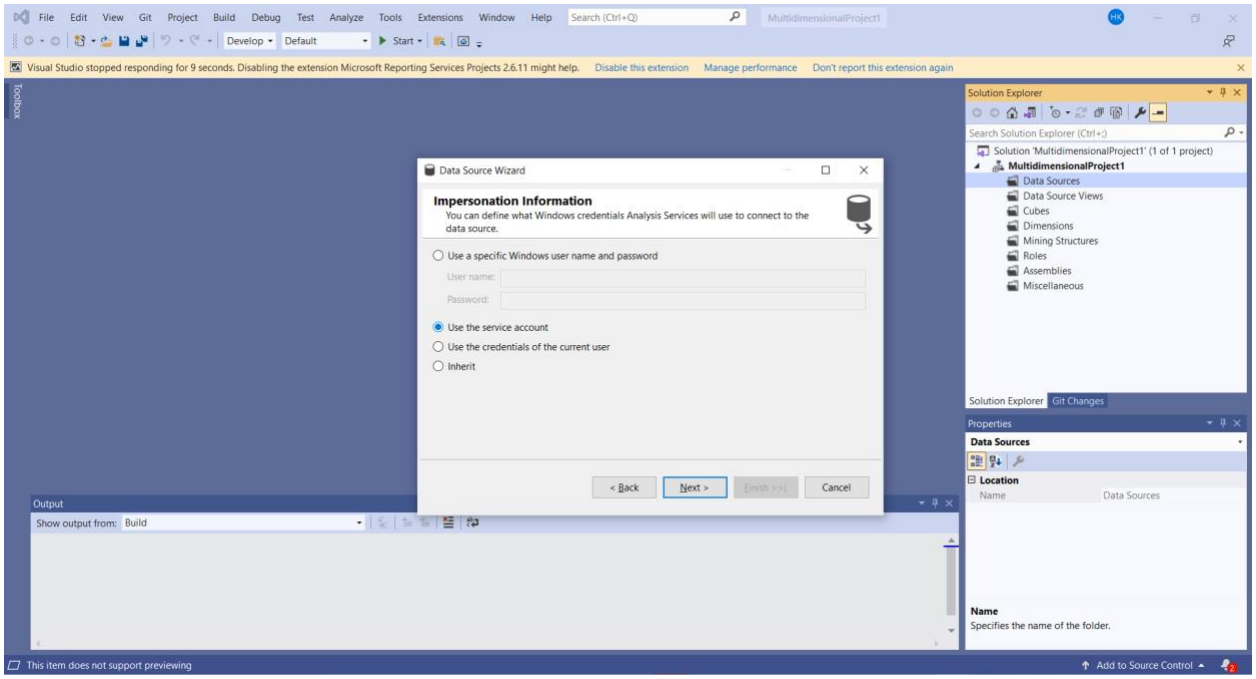
-Mentioned the server name and entered the database.



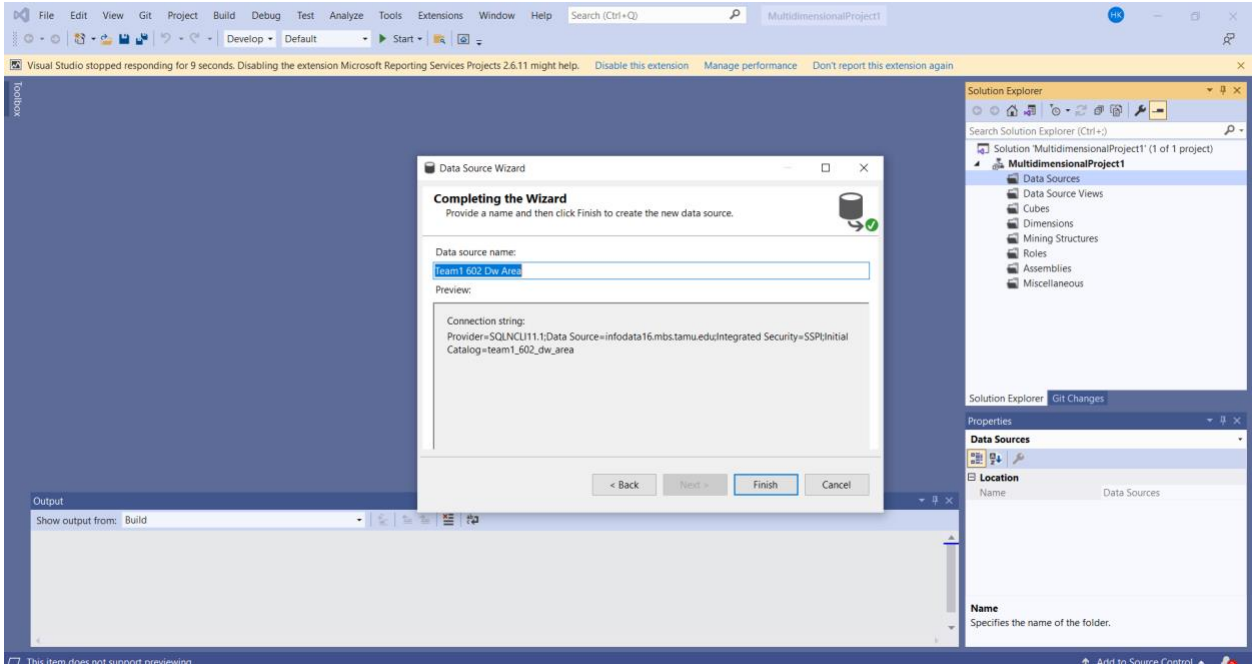
- Select how to define the connection

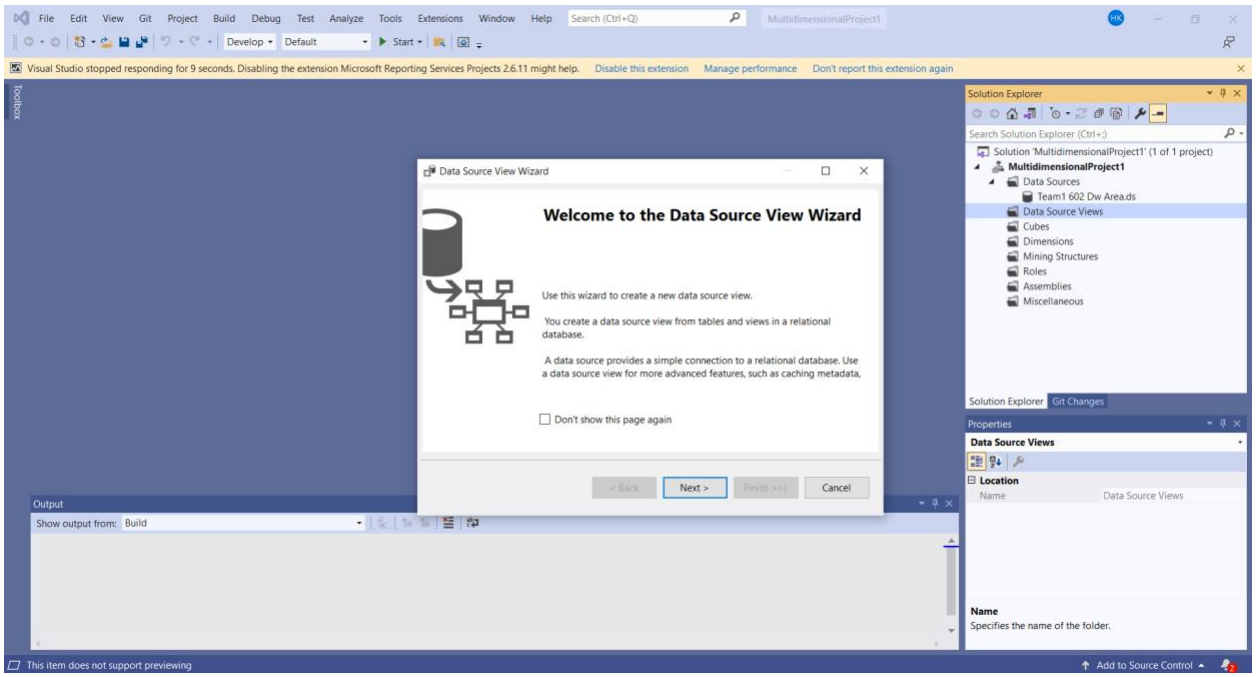


-Selecting the service account to define the credentials analysis service that will be connected.

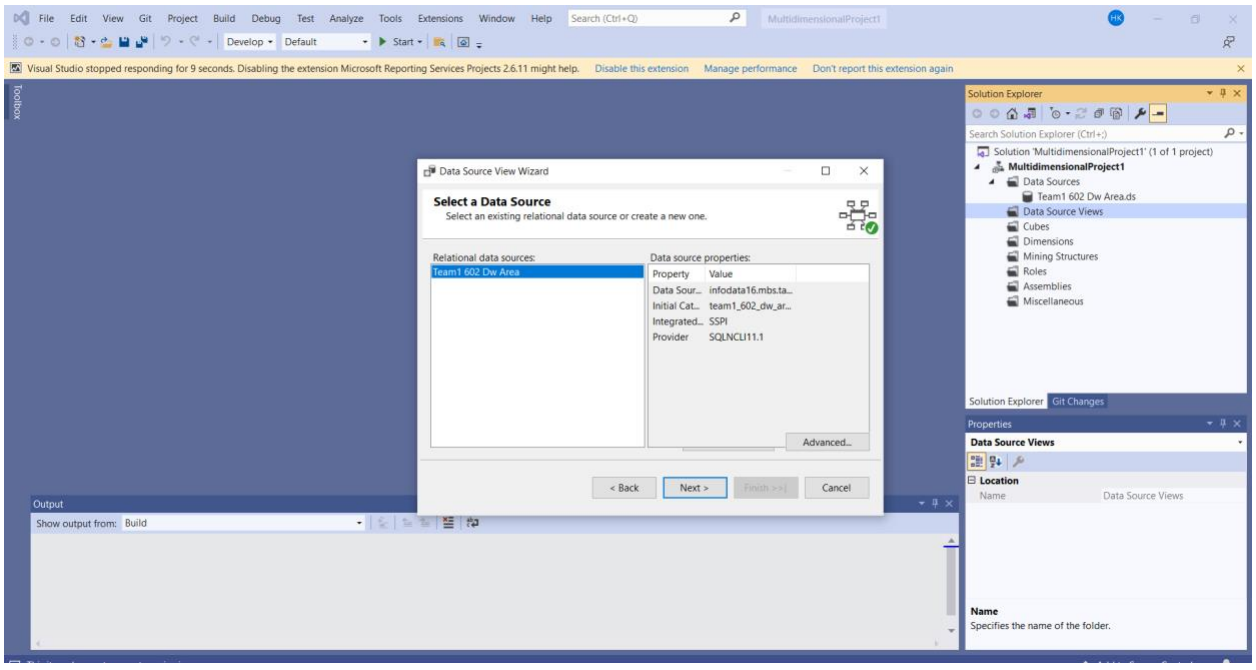


- Enter the data source name

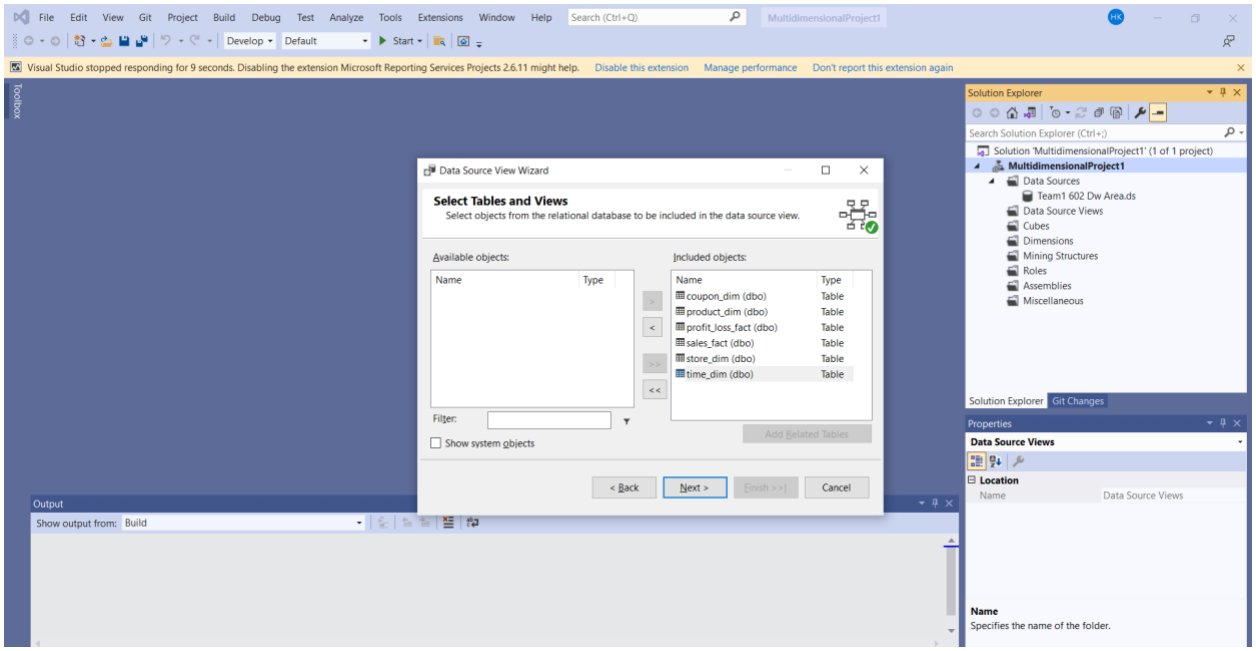




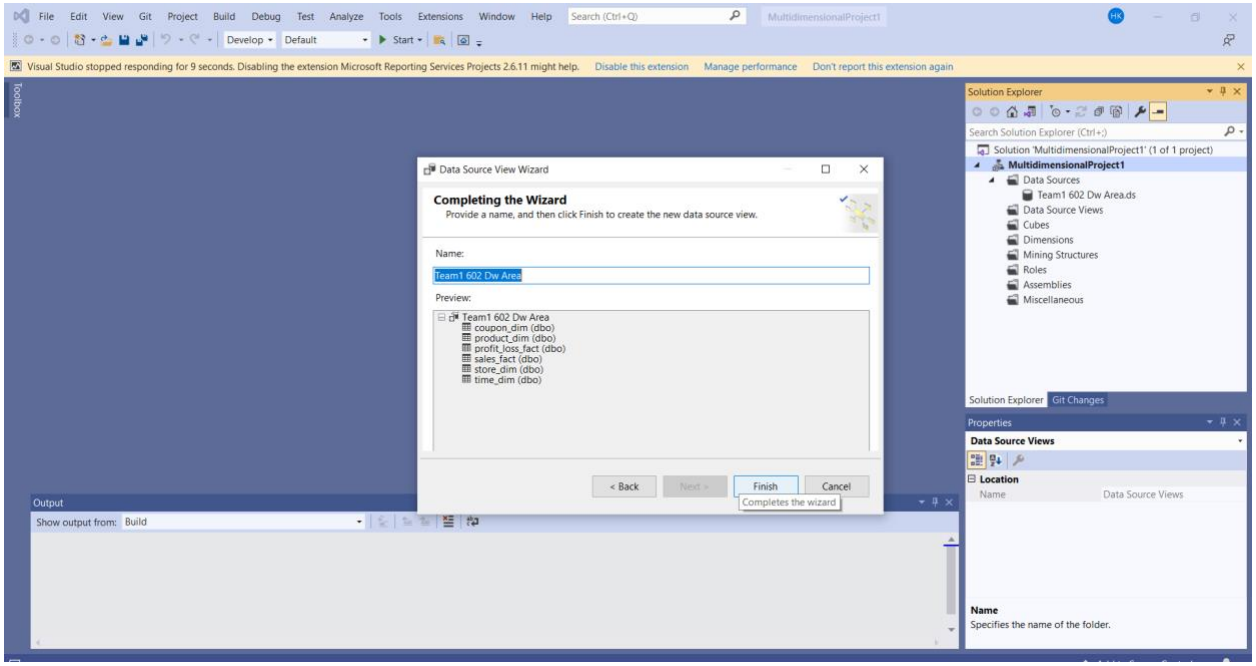
- Select Data Source



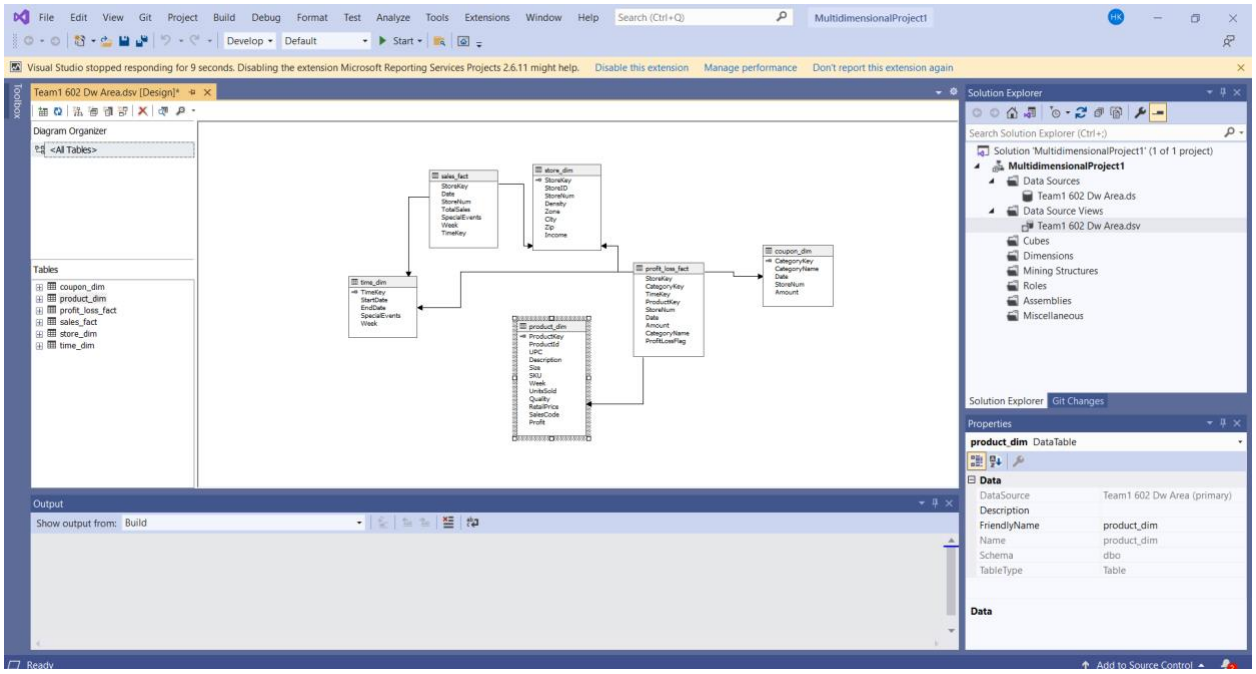
- Select tables and views



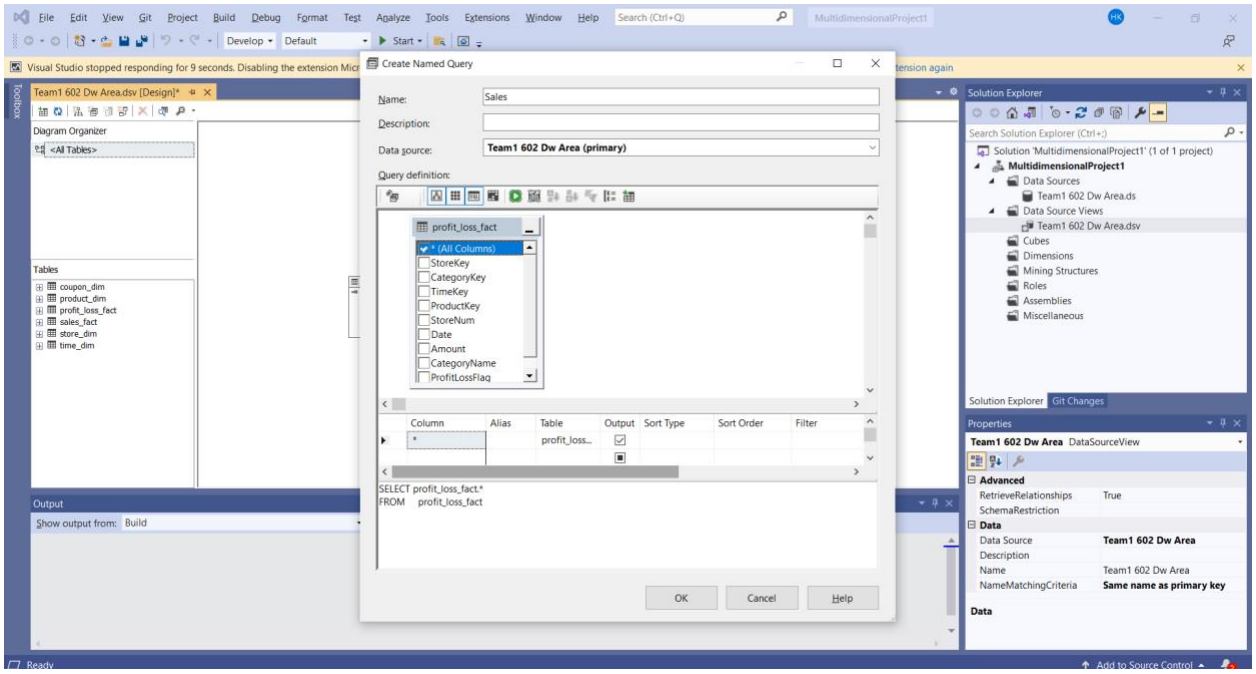
- Give a name to DW area



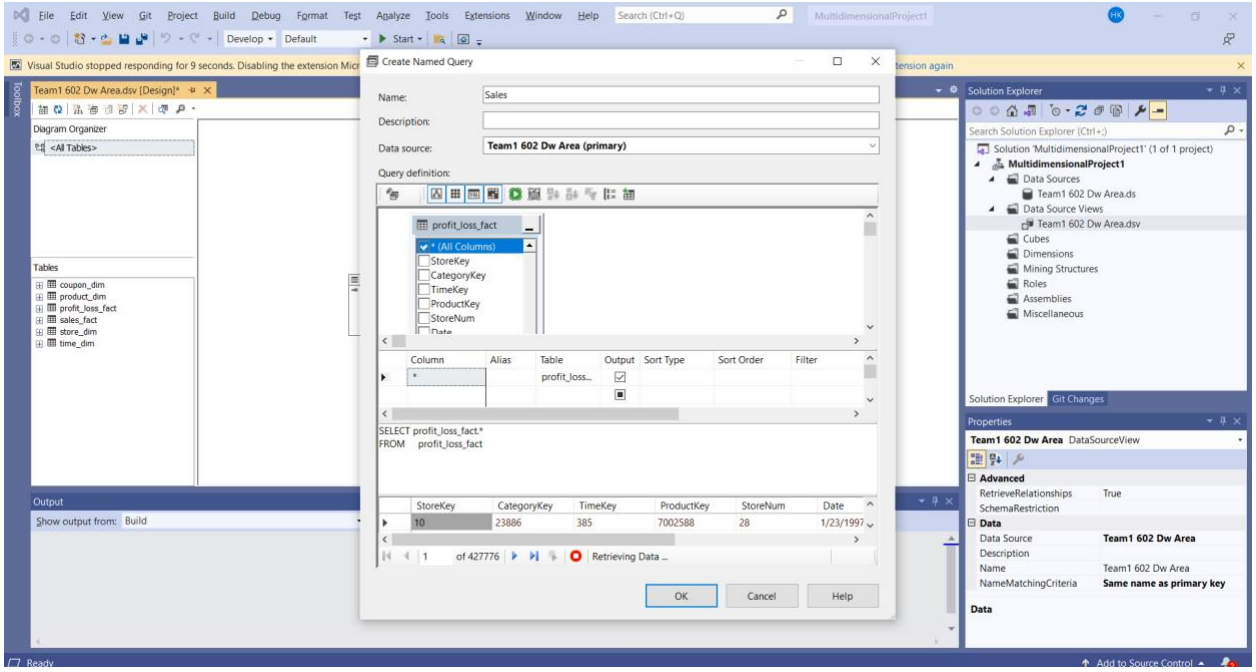
-Below defines the mapping of the dimension tables with the fact table.



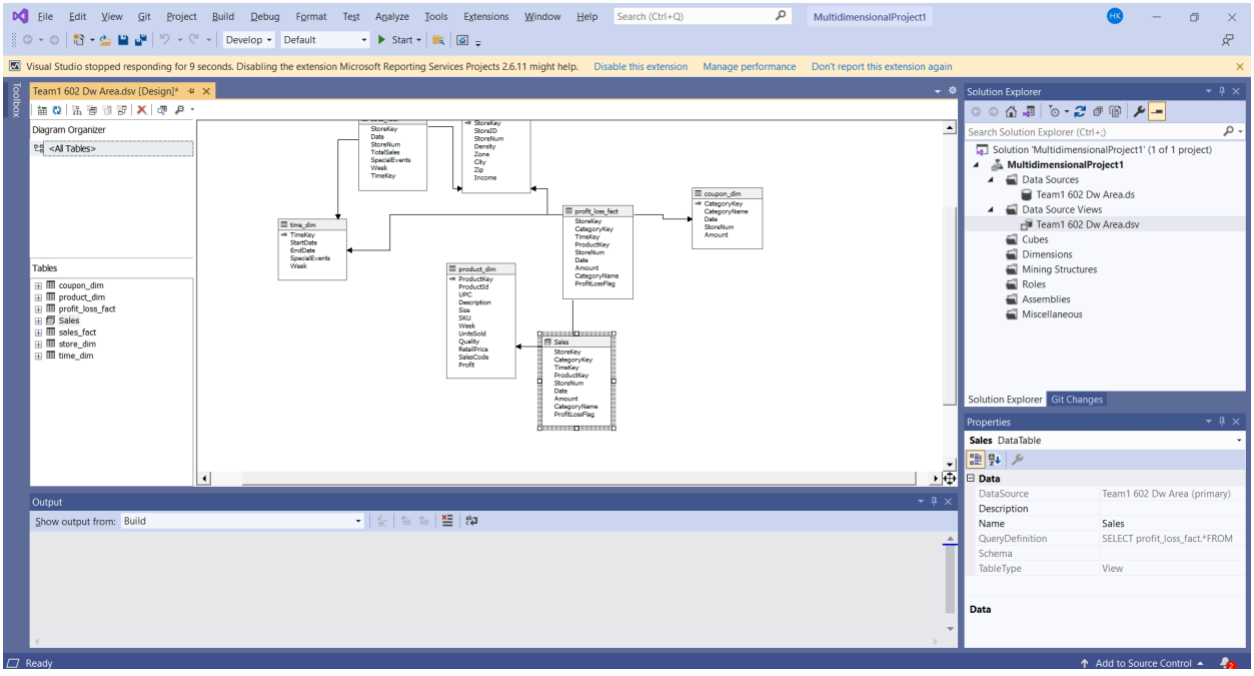
- Creating Named query

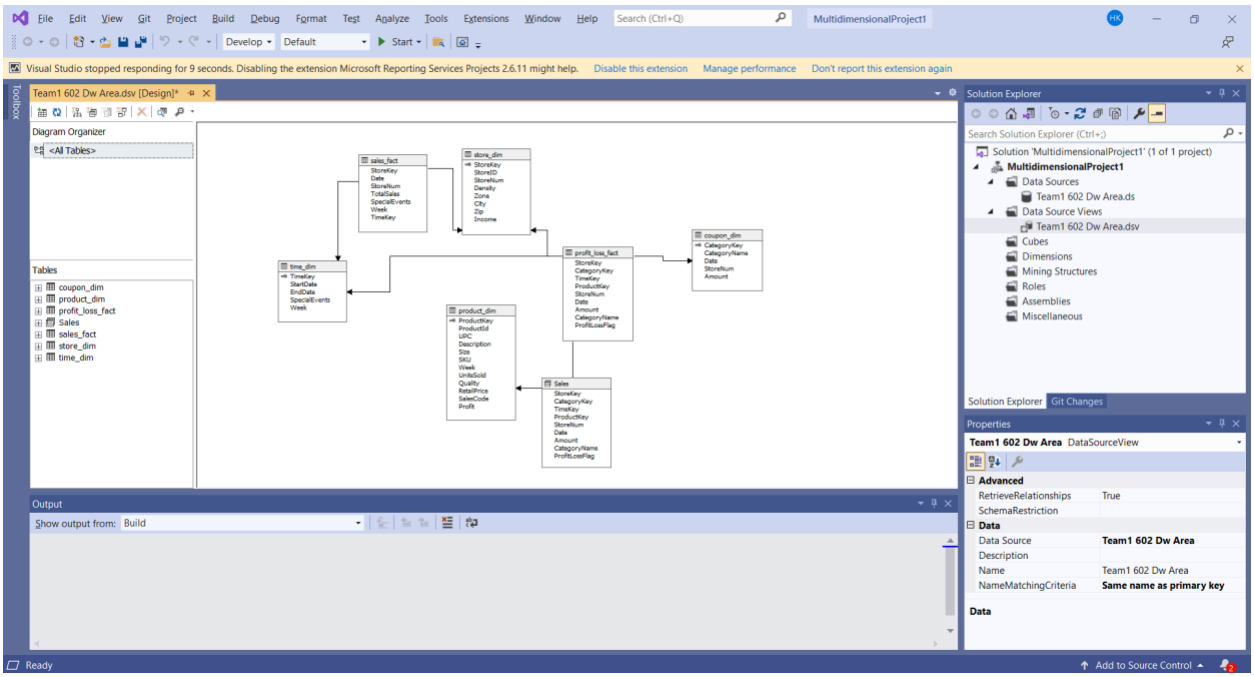


- Select the columns required

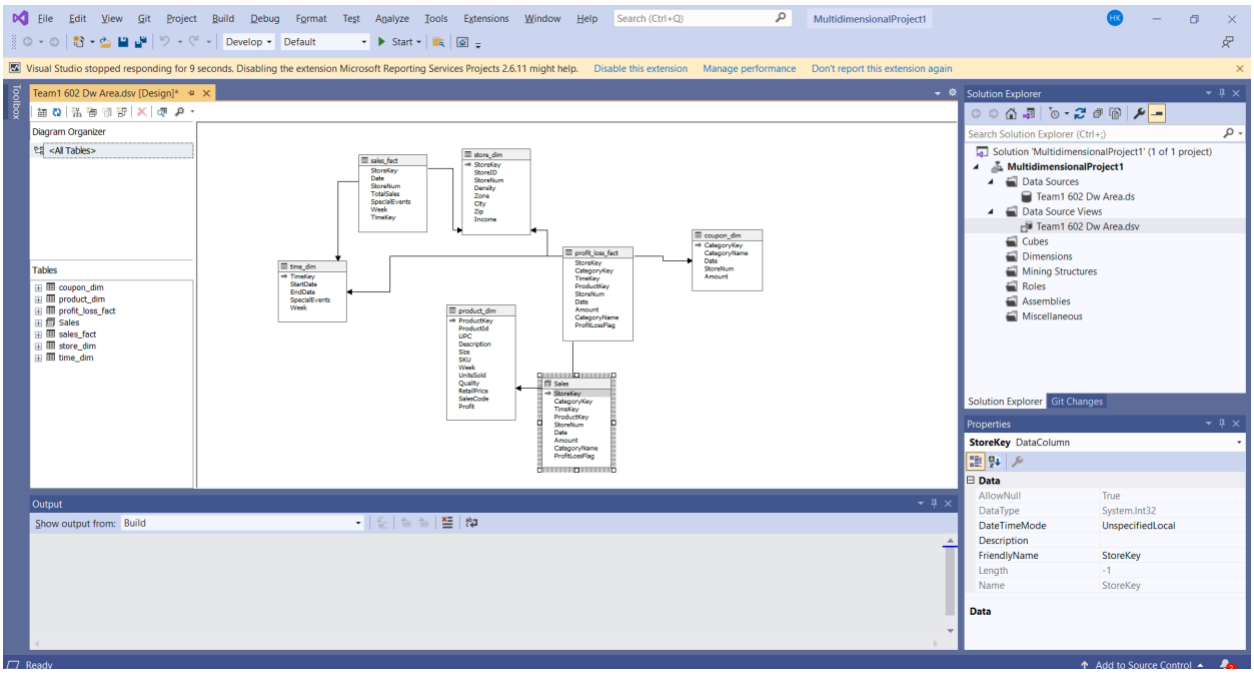


- Output representation

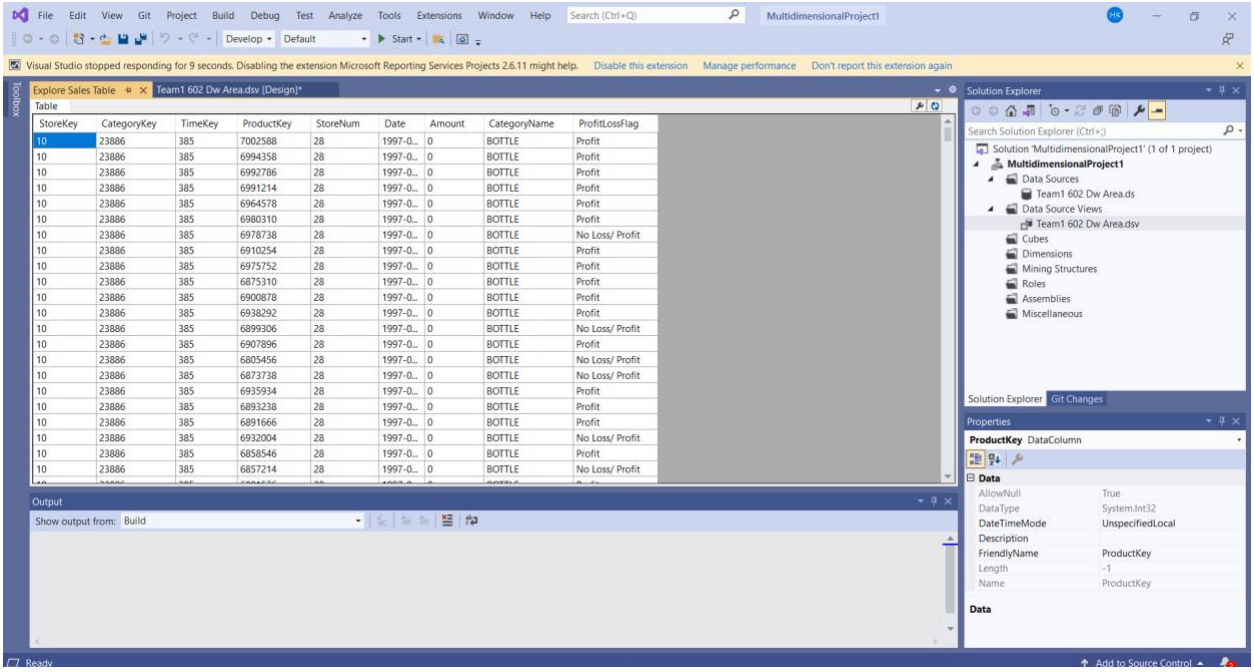




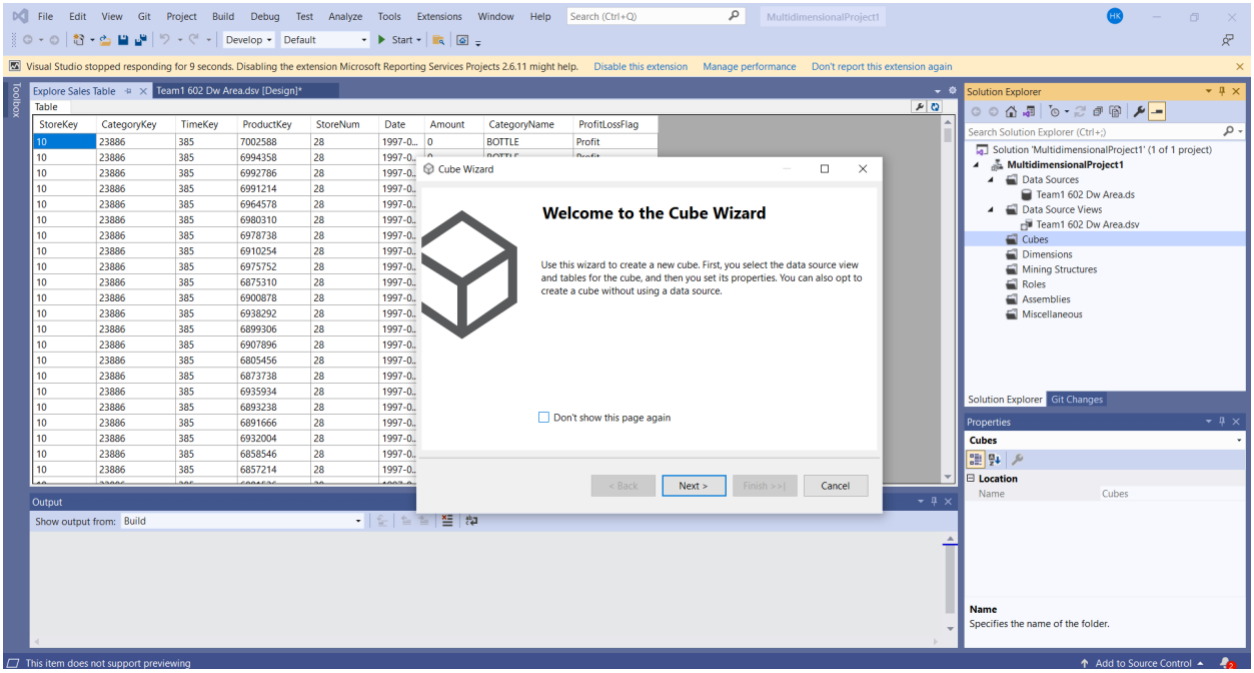
--Below defines the mapping of the dimension tables with the fact table.



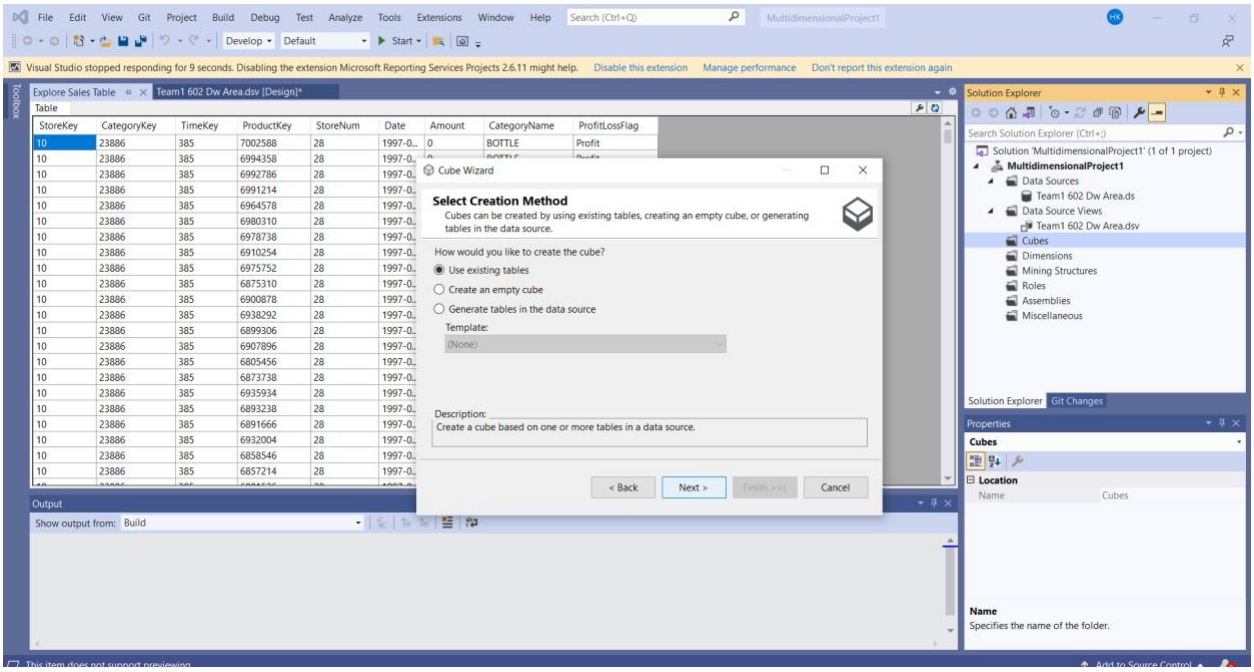
-Running visual studio to browse through the table.

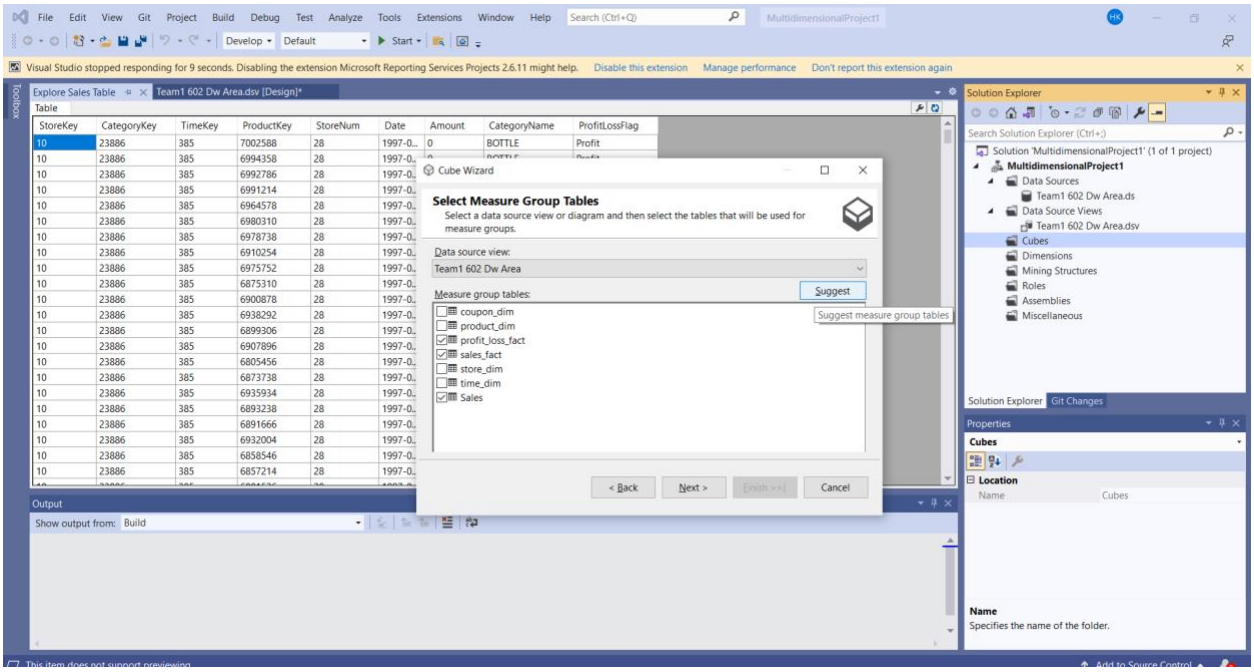


-We start the cube creation for the report

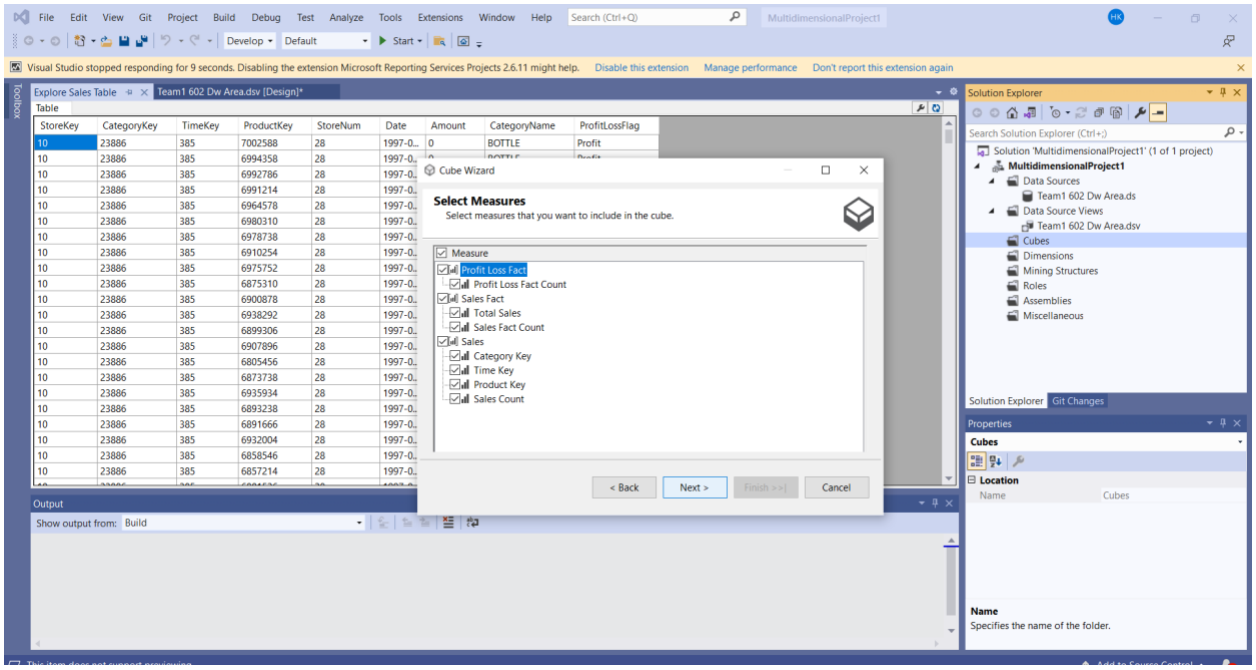


- Select Create using existing tables

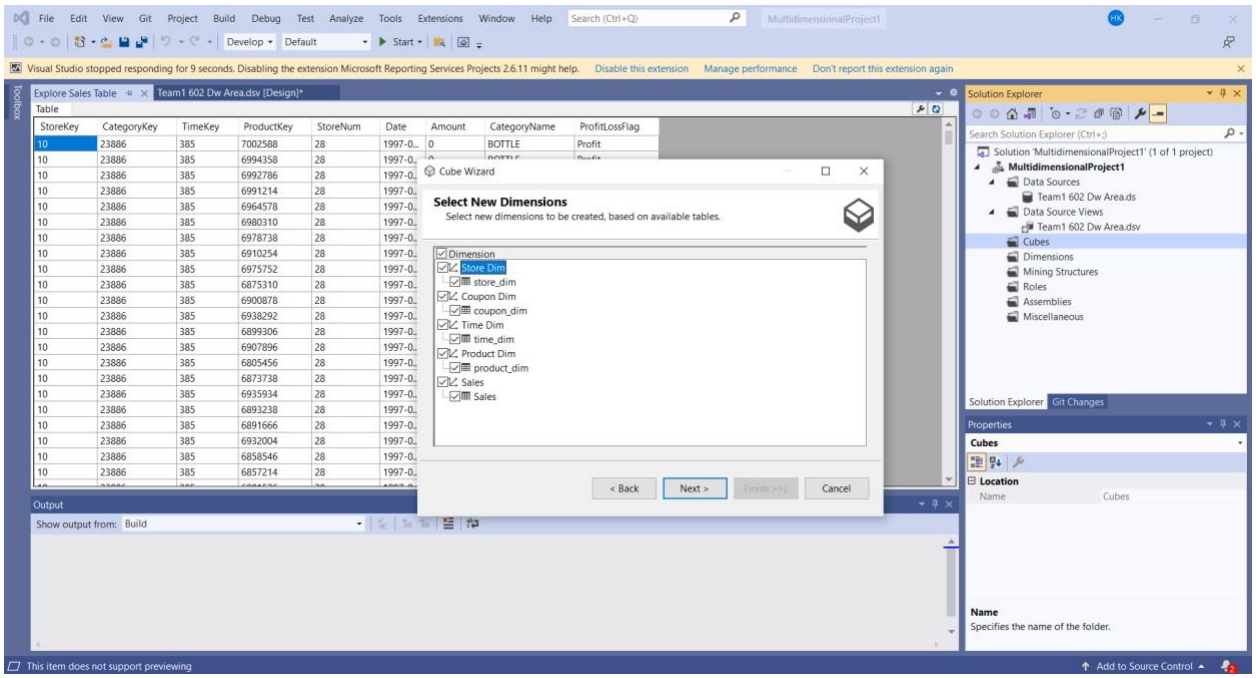




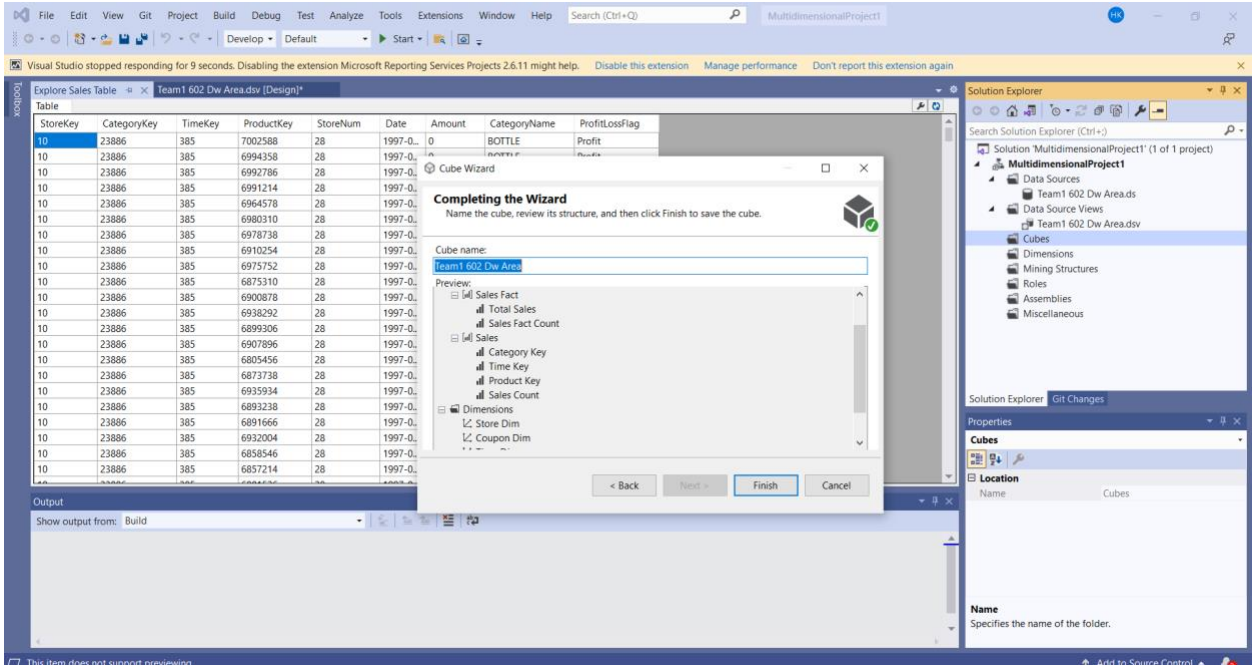
- Select profit loss fact table



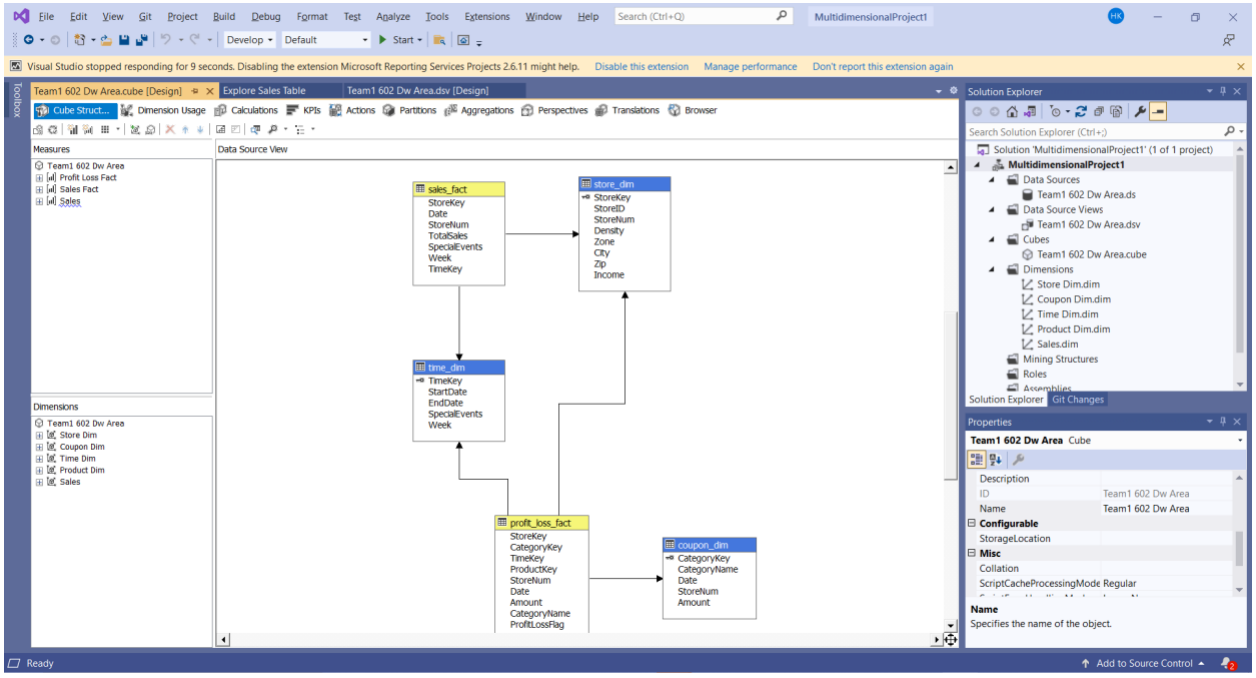
- Select store dimension table



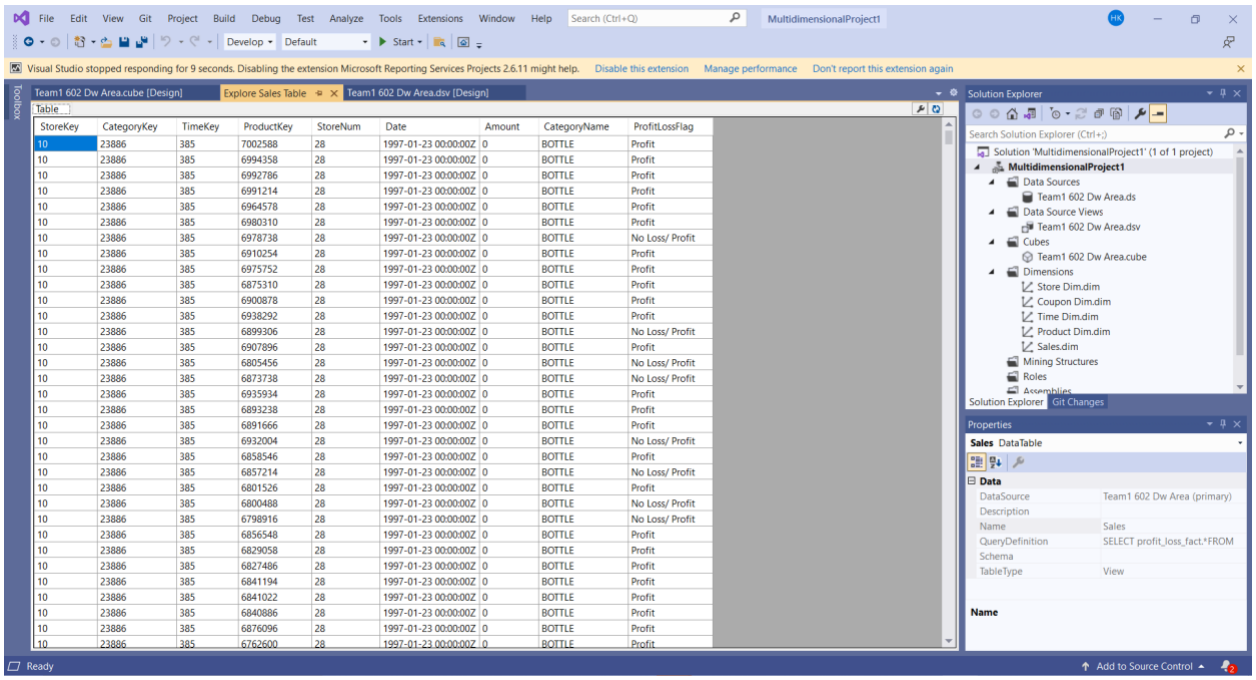
-The wizard setup has been completed, cube structure can be seen below :

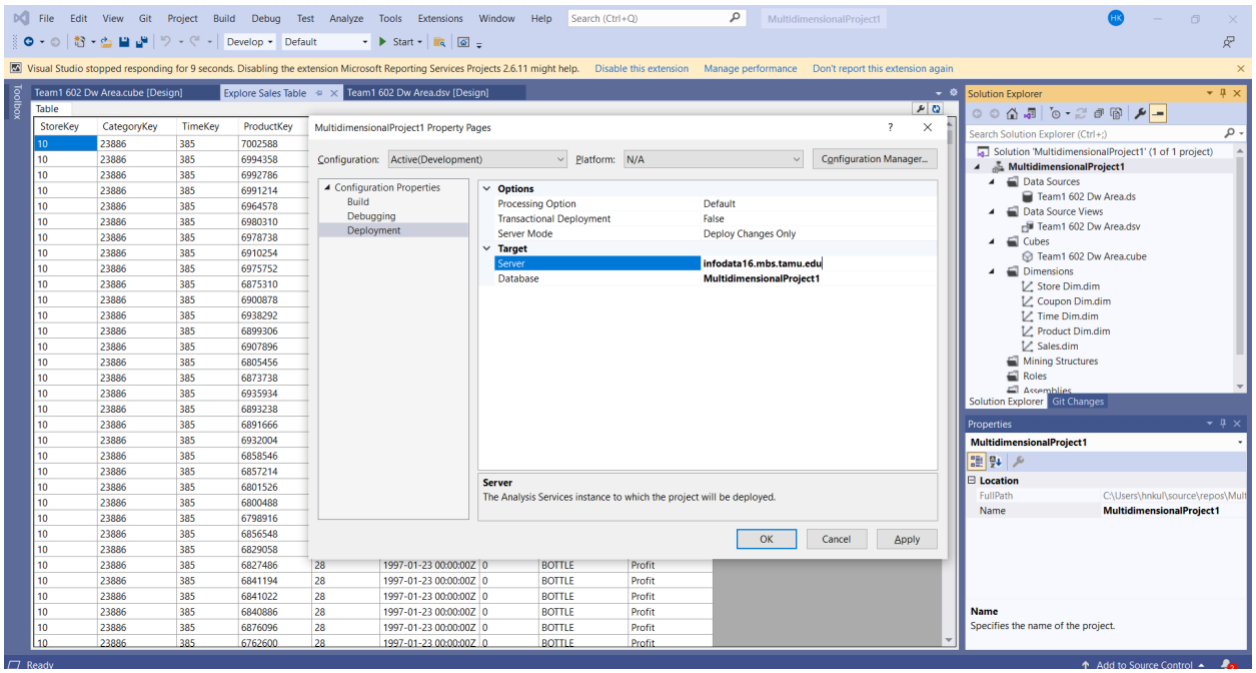


- Cube Design (With Dimension and Fact Tables)

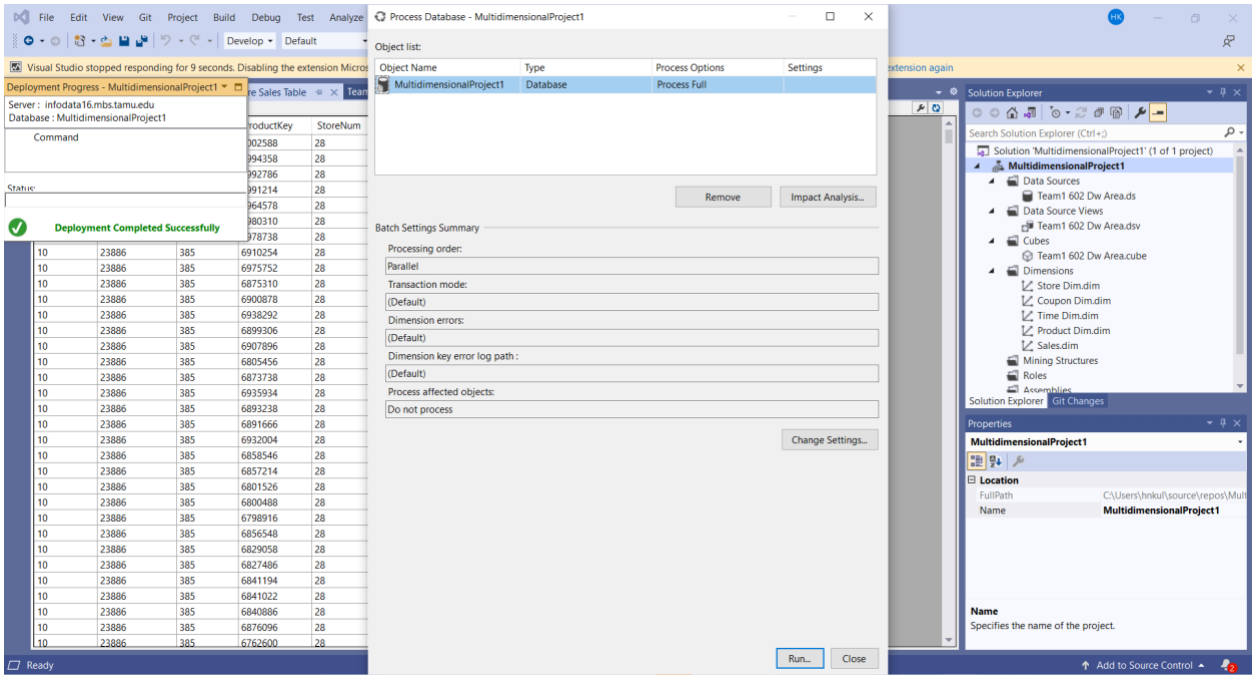


-Rendering in Visual Studio and Browsing through the tables.

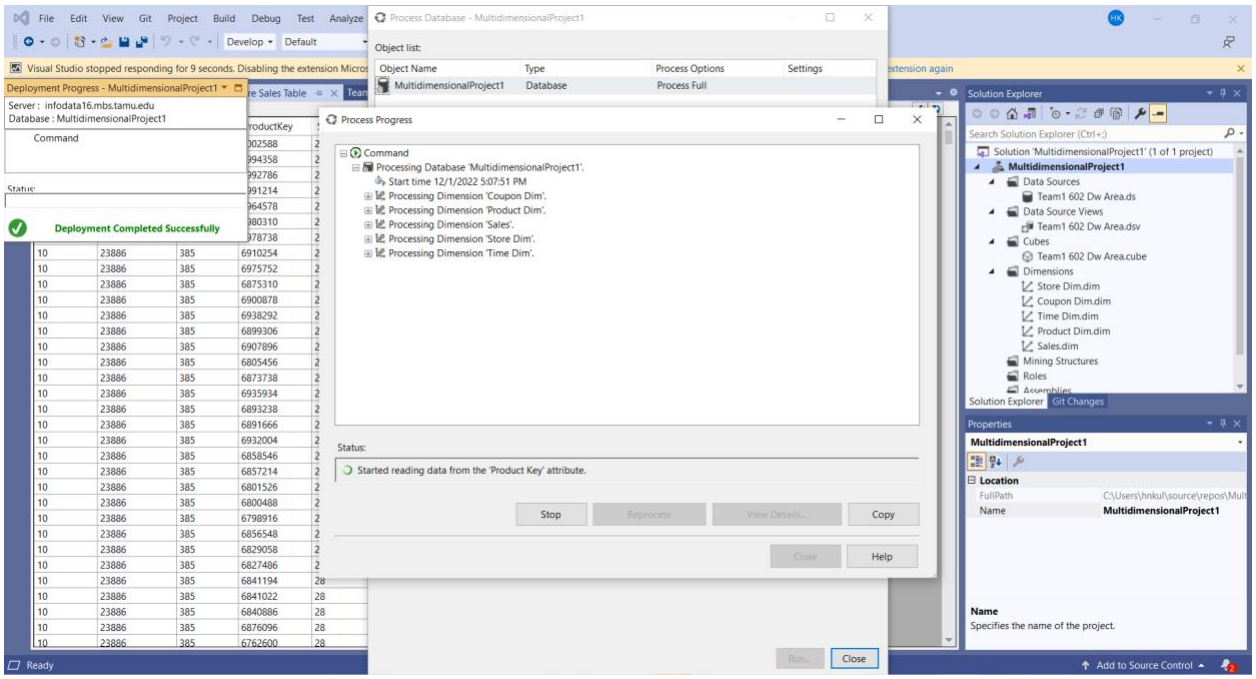




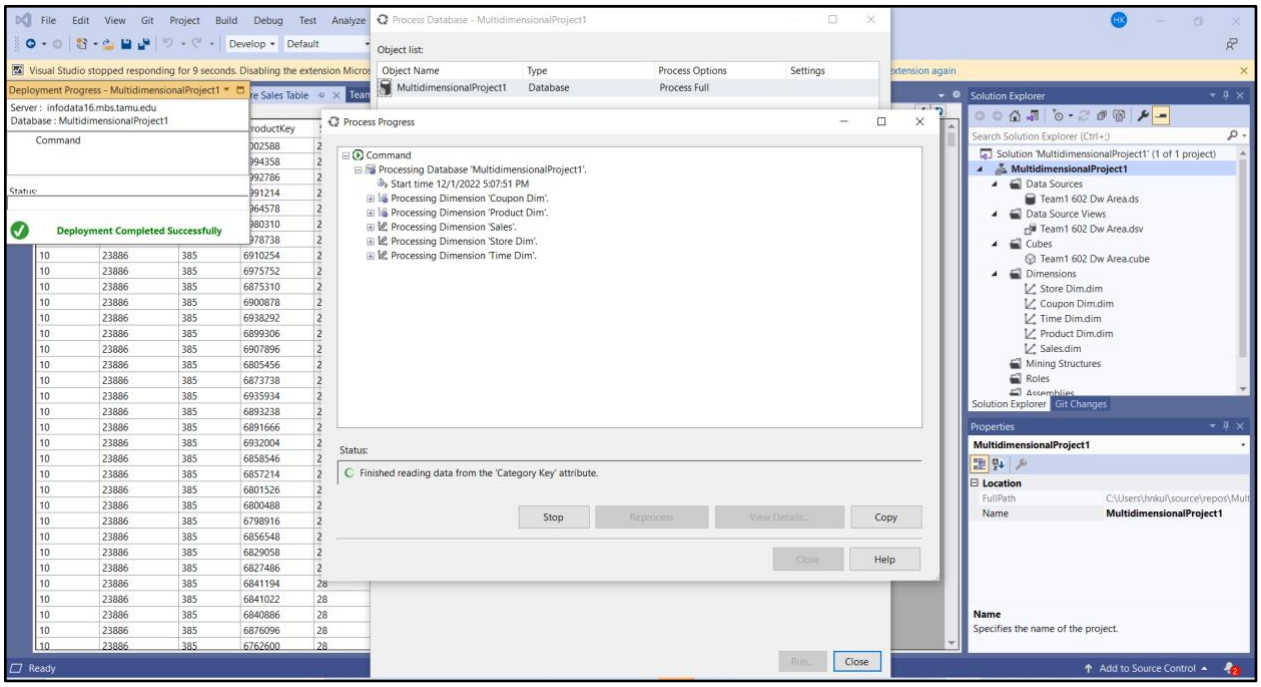
-Setting the batch summary, and processing the database.



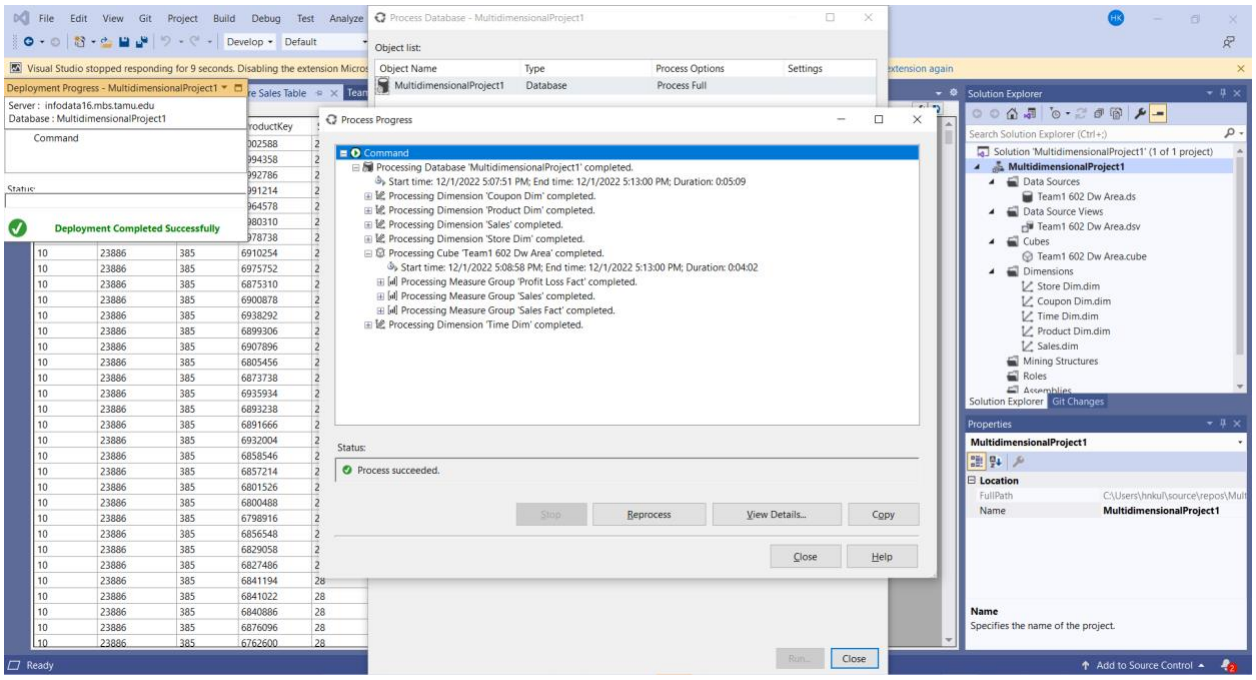
-Reading database process in progress, reading data from the 'Product Key' attribute.

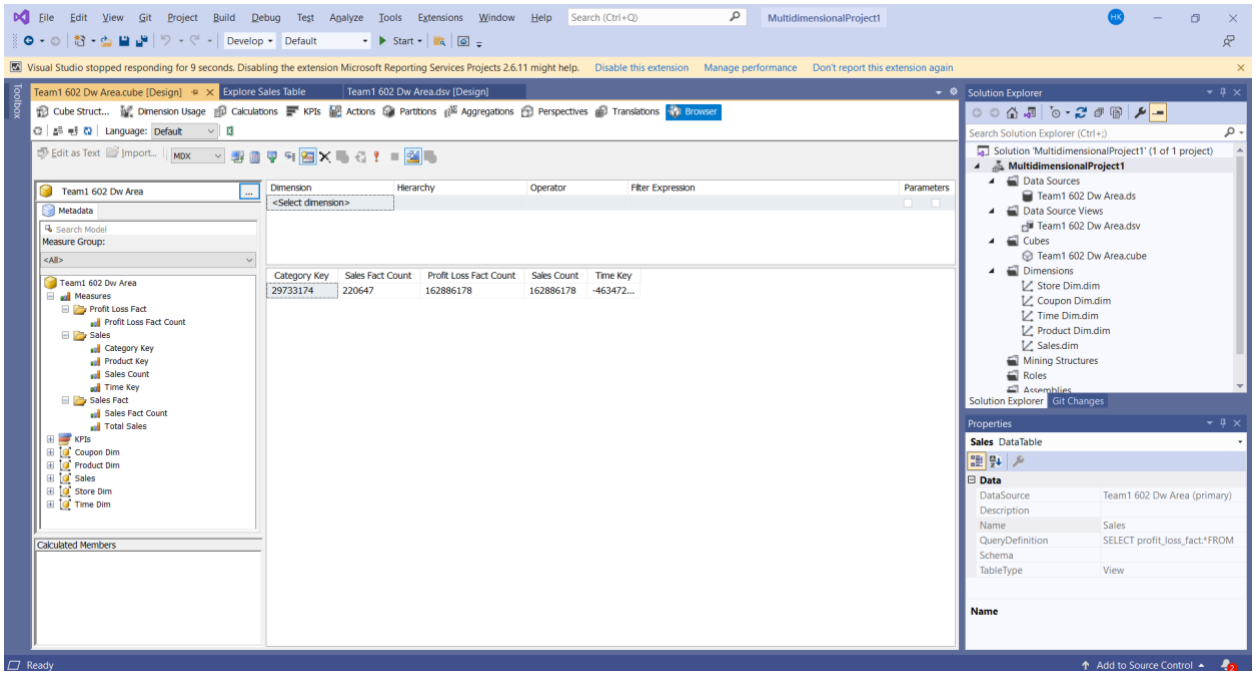


-Reading database process in progress, reading data from the Category Key' attribute.



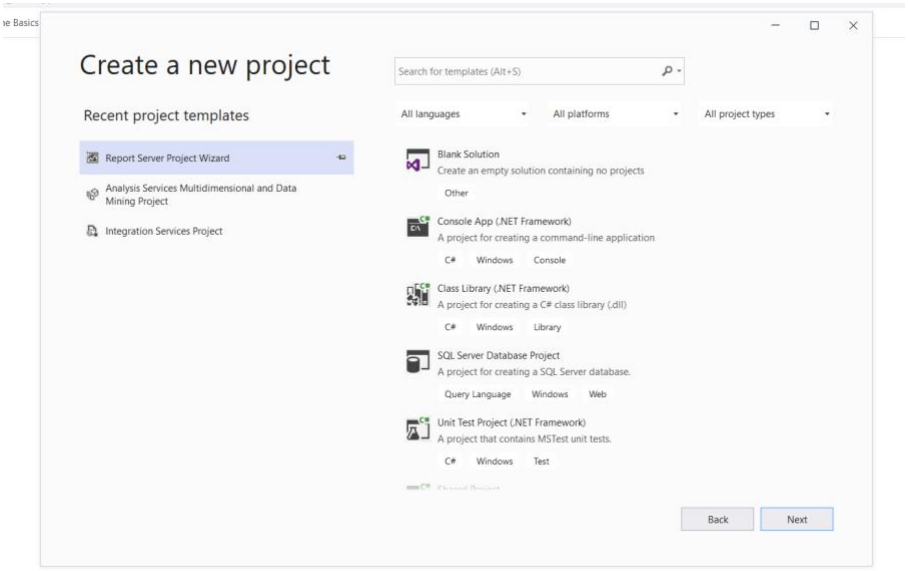
-Process has been successfully completed.

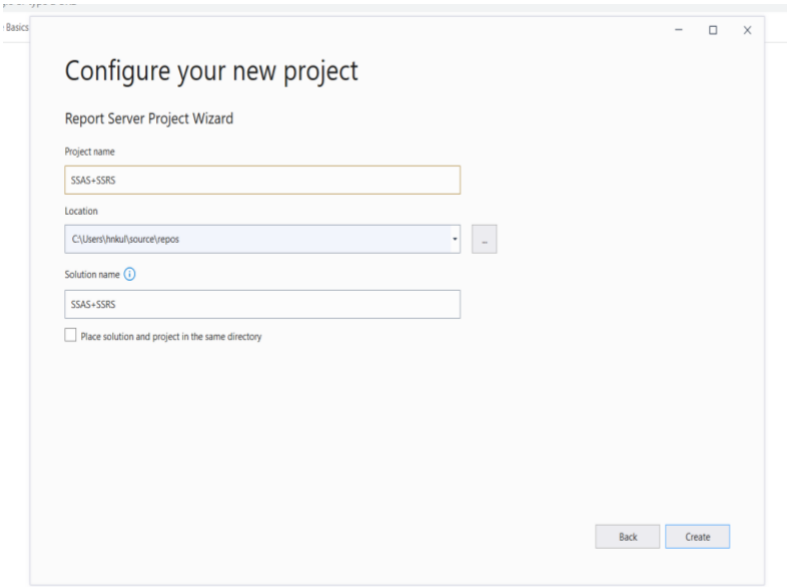



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
MultidimensionalProject1
Develop
Default
Start
Visual Studio stopped responding for 9 seconds. Disabling the extension Microsoft Reporting Services Projects 2.6.11 might help.
Disable this extension
Manage performance
Don't report this extension again
Toolbox
Team1 602 Dw Area.cube [Design]
Explore Sales Table
Team1 602 Dw Area.dsv [Design]
Cube Struct...
Dimension Usage
Calculations
KPIs
Actions
Partitions
Aggregations
Perspectives
Translations
Browser
Language: Default
Edit as Text
Import...
MDX
Team1 602 Dw Area
Metadata
Search Model
Measure Group:
<All>
Team1 602 Dw Area
Measures
Profit Loss Fact
Profit Loss Fact Count
Sales
Category Key
Product Key
Sales Count
Time Key
Sales Fact
Sales Fact Count
Total Sales
KPIs
Coupon Dim
Product Dim
Sales
Store Dim
Time Dim
Calculated Members
Dimension
Hierarchy
Operator
Filter Expression
Parameters
<Select dimension>
Category Key
Sales Fact Count
Profit Loss Fact Count
Sales Count
Time Key
29733174
220647
162886178
162886178
-463472...
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MultidimensionalProject1' (1 of 1 project)
MultidimensionalProject1
Data Sources
Team1 602 Dw Area.ds
Data Source Views
Team1 602 Dw Area.dsv
Cubes
Team1 602 Dw Area.cube
Dimensions
Store Dim.dim
Coupon Dim.dim
Time Dim.dim
Product Dim.dim
Sales.dim
Mining Structures
Roles
Assemblies
Solution Explorer
Git Changes
Properties
Sales DataTable
Data
DataSource
Team1 602 Dw Area (primary)
Description
Name
Sales
QueryDefinition
SELECT profit_loss_fact.*FROM
Schema
TableType
View
Name
Ready
Add to Source Control

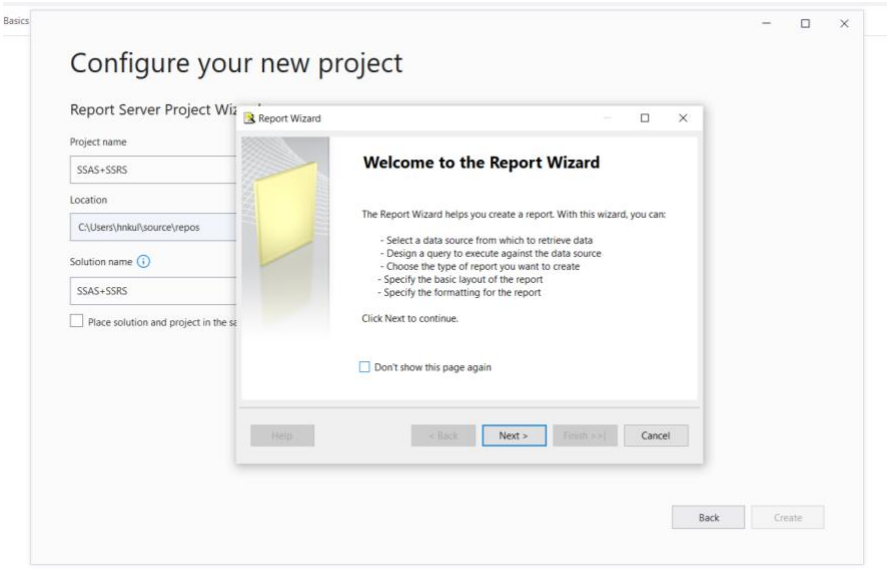
c. Reports using SSAS and SSRS

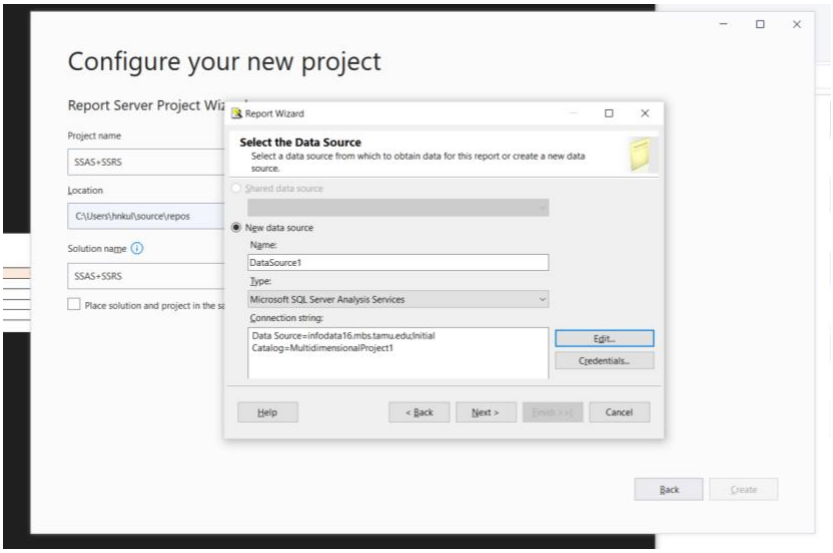
BQ6. : Which regions are densely populated, and what are the profits generated by these regions?

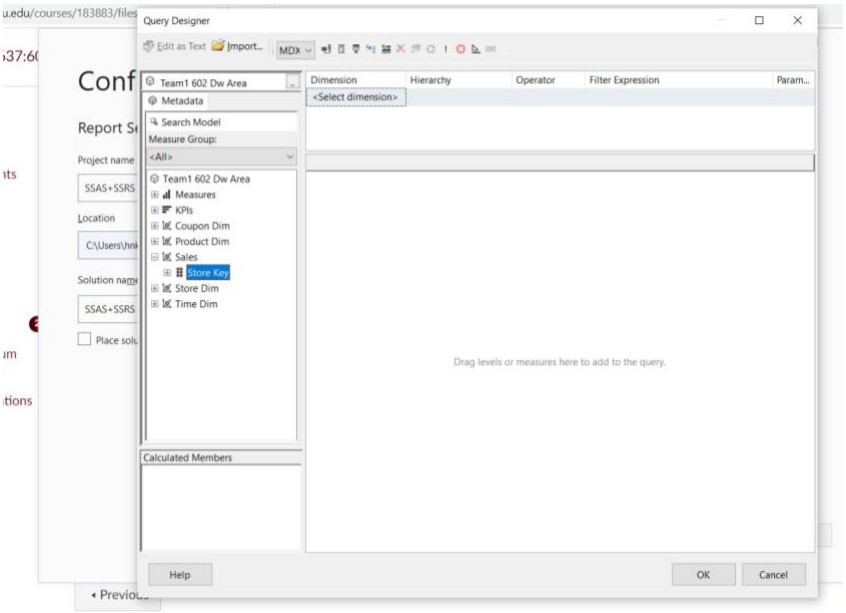


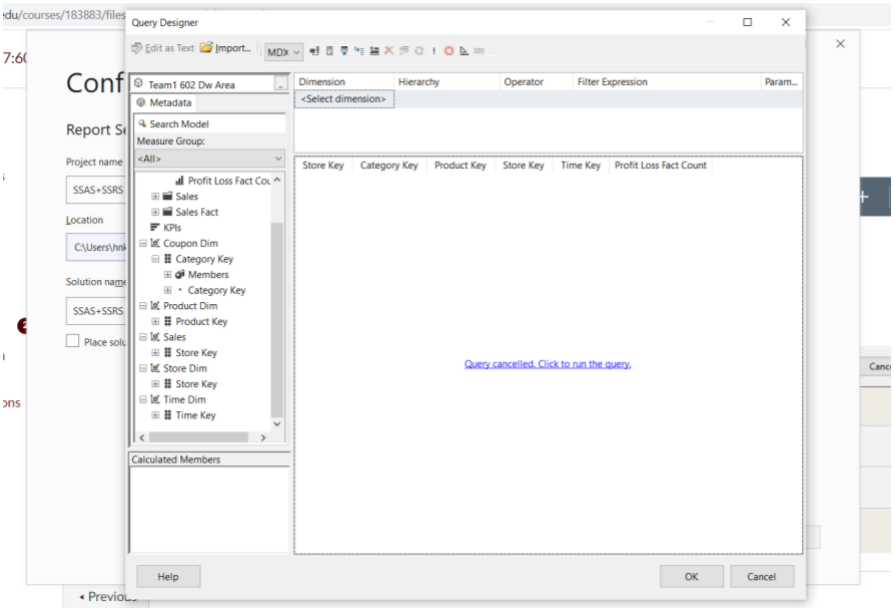


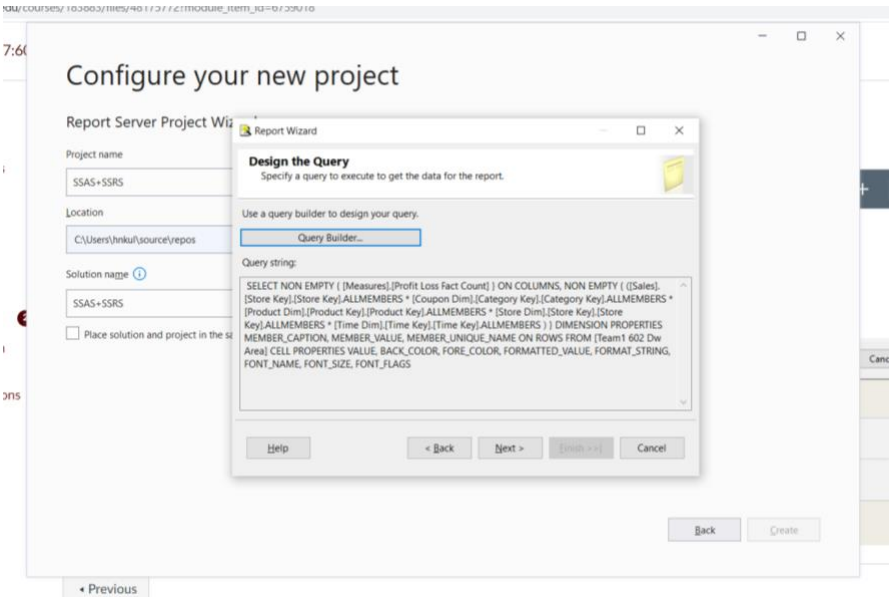
-Configuring new projects using Report Wizard

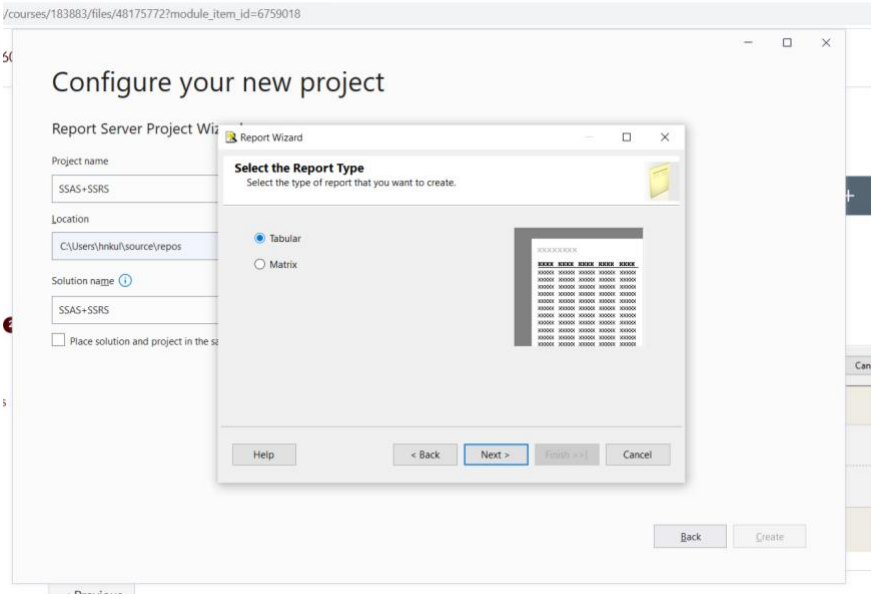


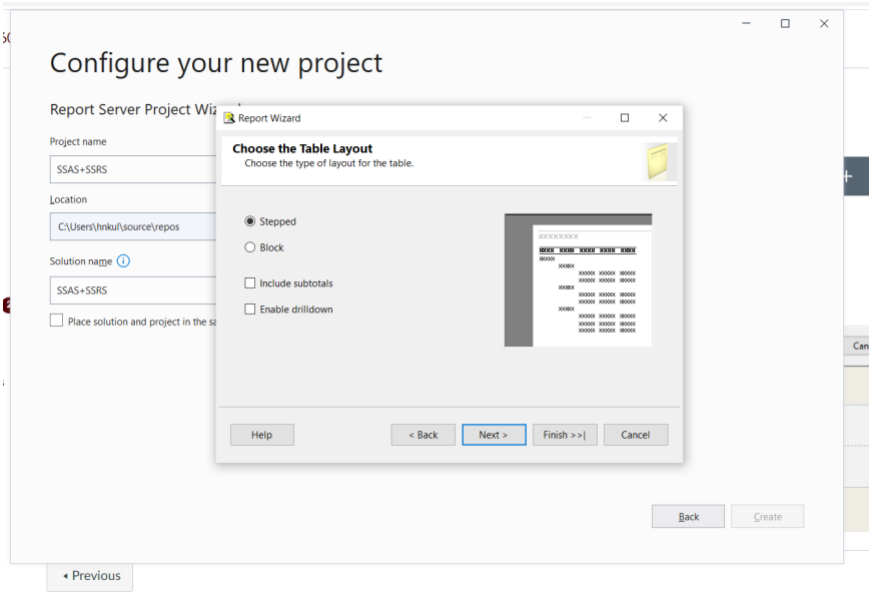



Query Designer
Edit as Text
Import...
MDX
Team1 602 Dw Area
Metadata
Search Model
Measure Group:
<All>
Team1 602 Dw Area
Measures
KPIs
Coupon Dim
Product Dim
Sales
Store Key
Store Dim
Time Dim
Dimension
Hierarchy
Operator
Filter Expression
Param...
<Select dimension>
Drag levels or measures here to add to the query.
Calculated Members
Help
OK
Cancel


Query Designer
Edit as Text
Import...
MDX
Team1 602 Dw Area
Metadata
Search Model
Measure Group:
<All>
Profit Loss Fact Cou
Sales
Sales Fact
KPIs
Coupon Dim
Category Key
Members
Category Key
Product Dim
Product Key
Sales
Store Key
Store Dim
Store Key
Time Dim
Time Key
Dimension
Hierarchy
Operator
Filter Expression
Param...
<Select dimension>
Store Key
Category Key
Product Key
Store Key
Time Key
Profit Loss Fact Count
Query cancelled. Click to run the query.
Calculated Members
Help
OK
Cancel

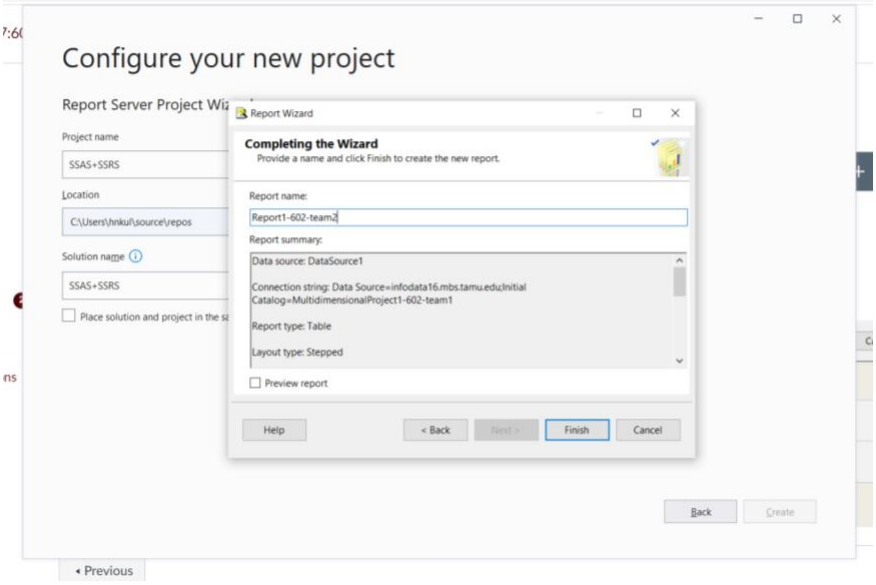

Configure your new project
Report Server Project Wi
Project name
SSAS+SSRS
Location
C:\Users\hnkul\source\repos
Solution name
SSAS+SSRS
Place solution and project in the s
Report Wizard
Design the Query
Specify a query to execute to get the data for the report.
Use a query builder to design your query.
Query Builder...
Query string:
SELECT NON EMPTY { [Measures].[Profit Loss Fact Count] } ON COLUMNS, NON EMPTY { ([Sales].[Store Key].[Store Key].ALLMEMBERS * [Coupon Dim].[Category Key].[Category Key].ALLMEMBERS * [Product Dim].[Product Key].[Product Key].ALLMEMBERS * [Store Dim].[Store Key].[Store Key].ALLMEMBERS * [Time Dim].[Time Key].[Time Key].ALLMEMBERS) } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_VALUE, MEMBER_UNIQUE_NAME ON ROWS FROM [Team1 602 Dw Area] CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR, FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS
Help
< Back
Next >
Cancel
Back
Create
Previous


/courses/183883/files/48175772?module_item_id=6759018
Configure your new project
Report Server Project Wi
Project name
SSAS+SSRS
Location
C:\Users\hnkul\source\repos
Solution name
SSAS+SSRS
Place solution and project in the s
Report Wizard
Select the Report Type
Select the type of report that you want to create.
Tabular
Matrix
Help
< Back
Next >
Cancel
Back
Create

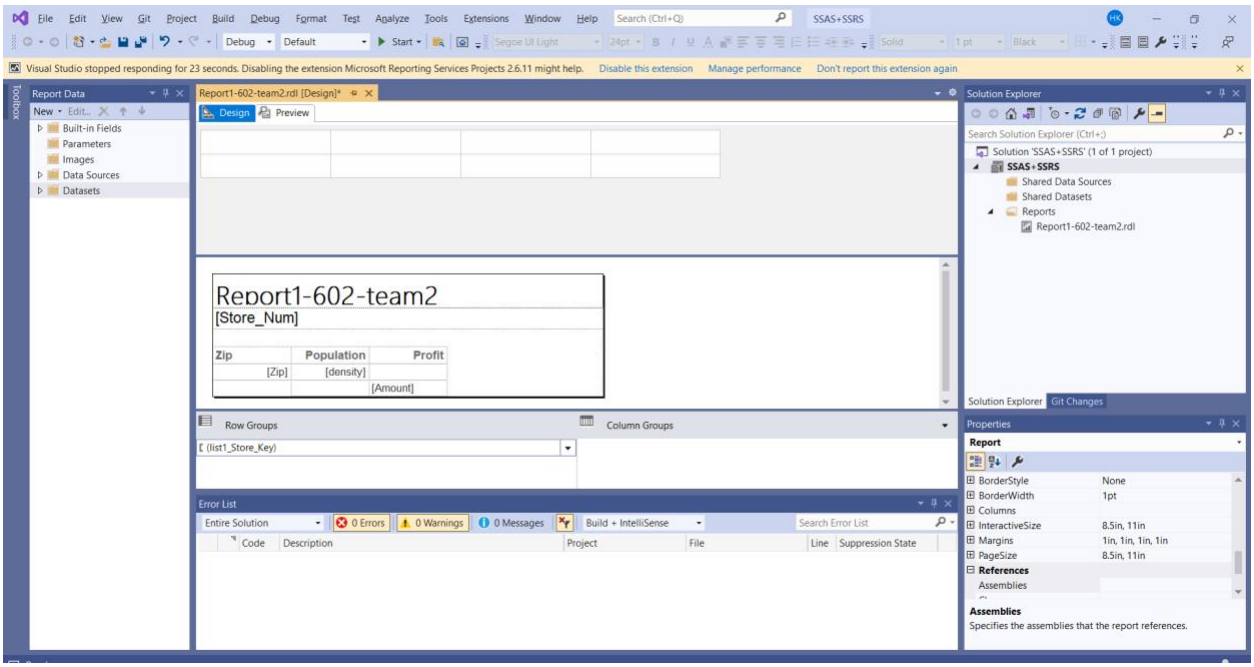


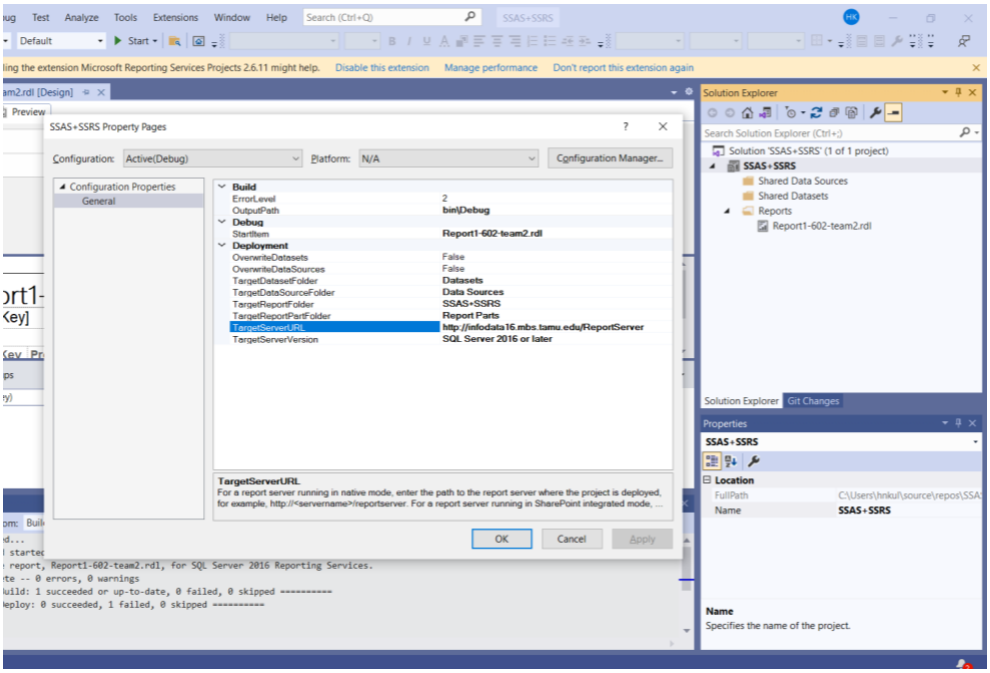
‹ Previous

du/courses/183883/files/48175772?module_item_id=6759018



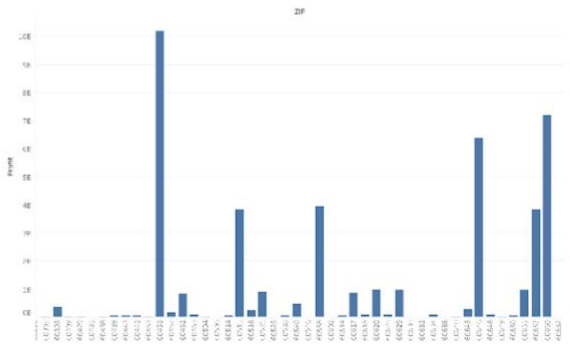
‹ Previous


File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help
SSAS+SSRS
Visual Studio stopped responding for 23 seconds. Disabling the extension Microsoft Reporting Services Projects 2.6.11 might help.
Disable this extension
Manage performance
Don't report this extension again
Report Data
Built-in Fields
Parameters
Images
Data Sources
Datasets
Report1-602-team2.rdl [Design]*
Design
Preview
Report1-602-team2
[Store_Num]
Zip
Population
Profit
[Zip]
[density]
[Amount]
Row Groups
Column Groups
[(list1_Store_Key)
Error List
Entire Solution
0 Errors
0 Warnings
0 Messages
Build + IntelliSense
Search Error List
Code
Description
Project
File
Line
Suppression State
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SSAS+SSRS' (1 of 1 project)
SSAS+SSRS
Shared Data Sources
Shared Datasets
Reports
Report1-602-team2.rdl
Solution Explorer
Git Changes
Properties
Report
BorderStyle
None
BorderWidth
1pt
Columns
InteractiveSize
8.5in, 11in
Margins
1in, 1in, 1in, 1in
PageSize
8.5in, 11in
References
Assemblies
Assemblies
Specifies the assemblies that the report references.


SSAS+SSRS Property Pages
Configuration: Active(Debug)
Platform: N/A
Configuration Manager...
Configuration Properties
General
Build
ErrorLevel
2
OutputPath
bin\Debug
Debug
StartItem
Report1-602-team2.rdl
Deployment
OverwriteDatasets
False
OverwriteDataSources
False
TargetDatasetFolder
Datasets
TargetDataSourceFolder
Data Sources
TargetReportFolder
SSAS+SSRS
TargetReportPartFolder
Report Parts
TargetServerURL
http://infodata16.mbs.tamu.edu/ReportServer
TargetServerVersion
SQL Server 2016 or later
TargetServerURL
For a report server running in native mode, enter the path to the report server where the project is deployed, for example, http://<servername>/reportserver. For a report server running in SharePoint integrated mode, ...
OK
Cancel
Apply
report, Report1-602-team2.rdl, for SQL Server 2016 Reporting Services.
-- 0 errors, 0 warnings
uild: 1 succeeded or up-to-date, 0 failed, 0 skipped
eploy: 0 succeeded, 1 failed, 0 skipped
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SSAS+SSRS' (1 of 1 project)
SSAS+SSRS
Shared Data Sources
Shared Datasets
Reports
Report1-602-team2.rdl
Solution Explorer
Git Changes
Properties
SSAS+SSRS
Location
FullPath
C:\Users\hnkul\source\repos\SSAS
Name
SSAS+SSRS
Name
Specifies the name of the project.

Report1-602-team2

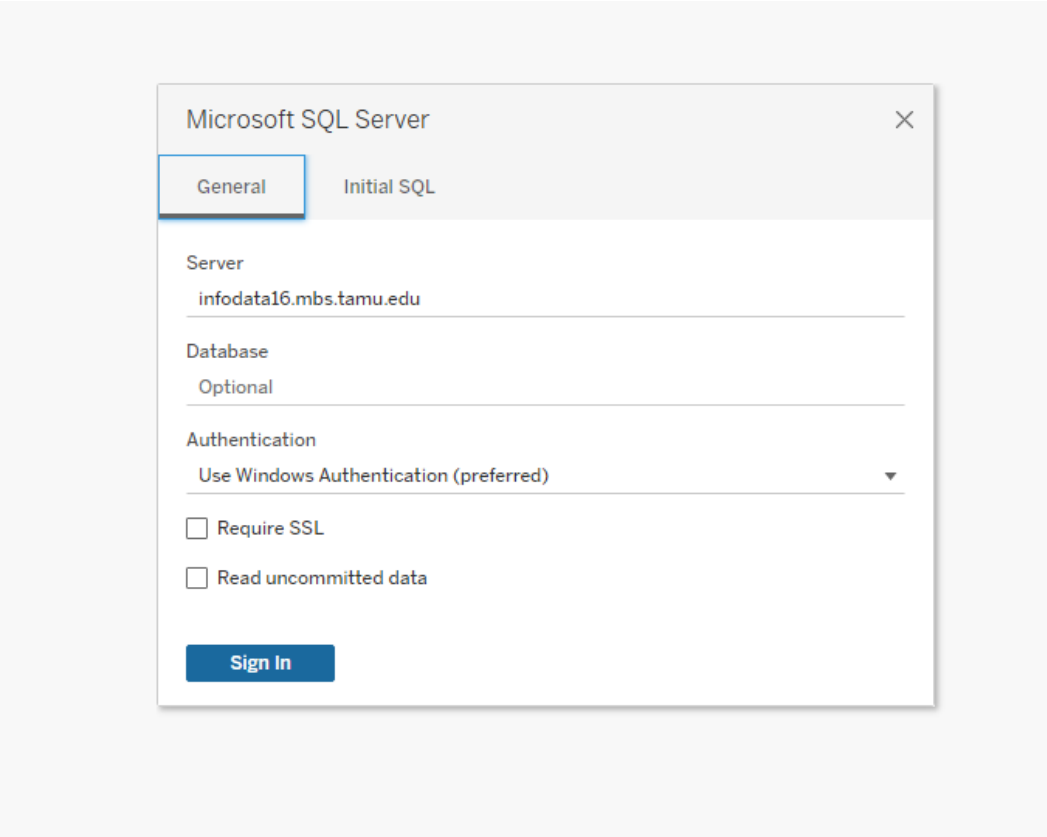
Zip	Population	Profit
60305	0.00045604	
		369973740
60068	0.00057152	
		644847843
60067	0.00111928	
		2355744638
60453	0.0003905	
		9283748
60053	0.00070336	
		3736368264
60660	0.00018423	
		7192731891
60025	0.00096256	
		1927364863
60171	0.00038543	
		8928746738
60103	0.00083723	
		48468293
60056	0.00124321	
		39387474
60068	0.00045602	
		3737438
60657	0.00024846	
		3847484954
60455	0.00068876	
		9289474839



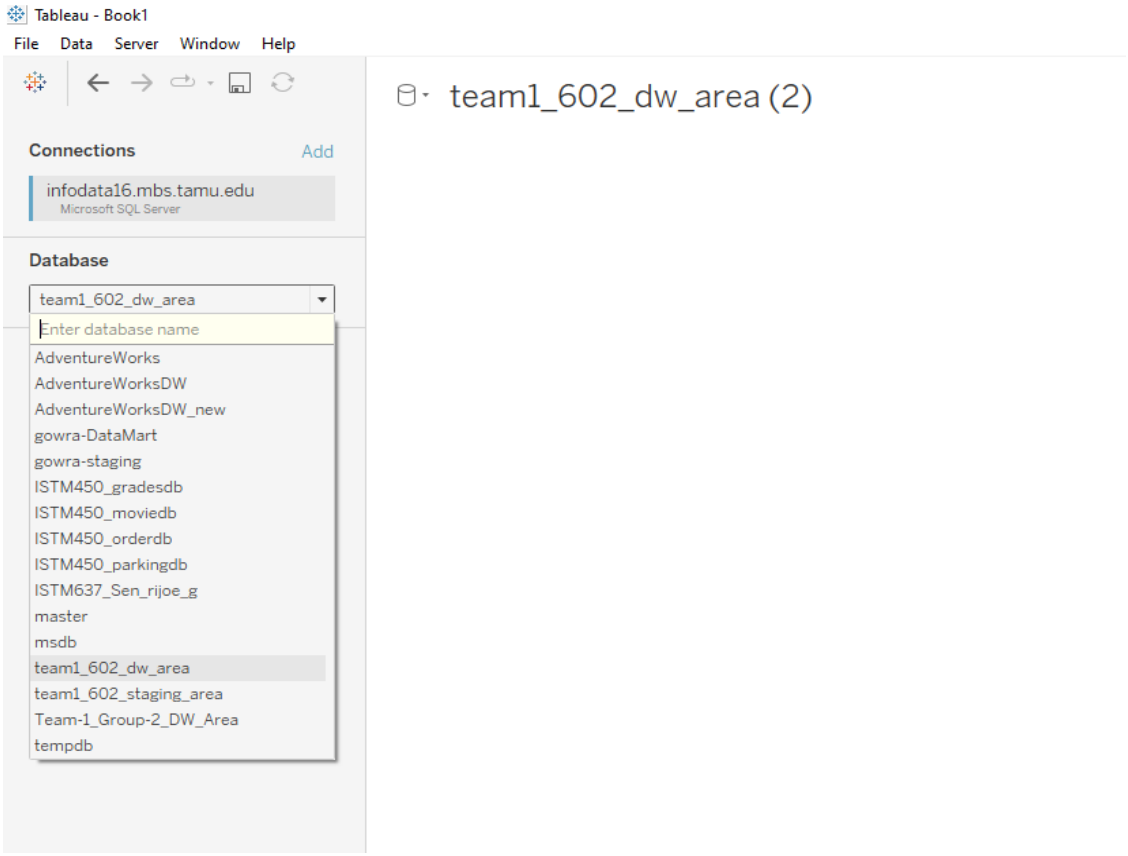
d. Reports using Tableau

BQ1 - What is the percentage change in net sales quarter over quarter for the period 1990-96?

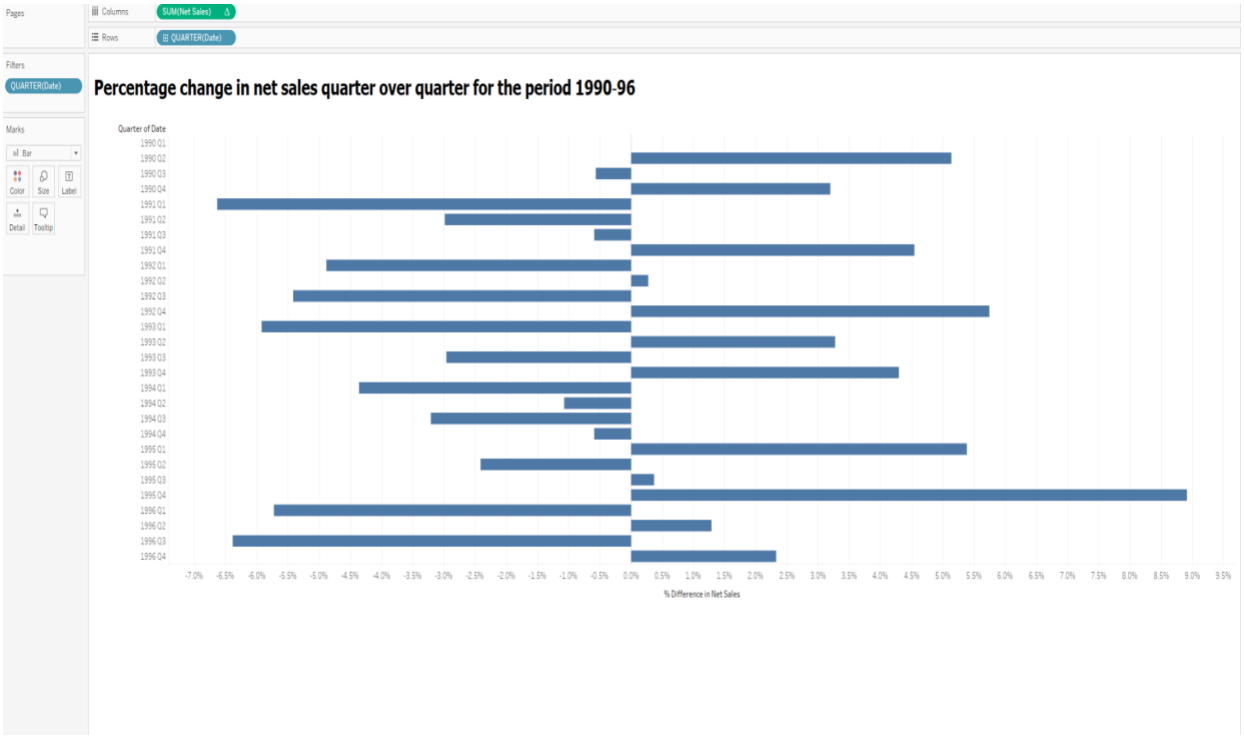
- Connecting MS SQL server to Tableau



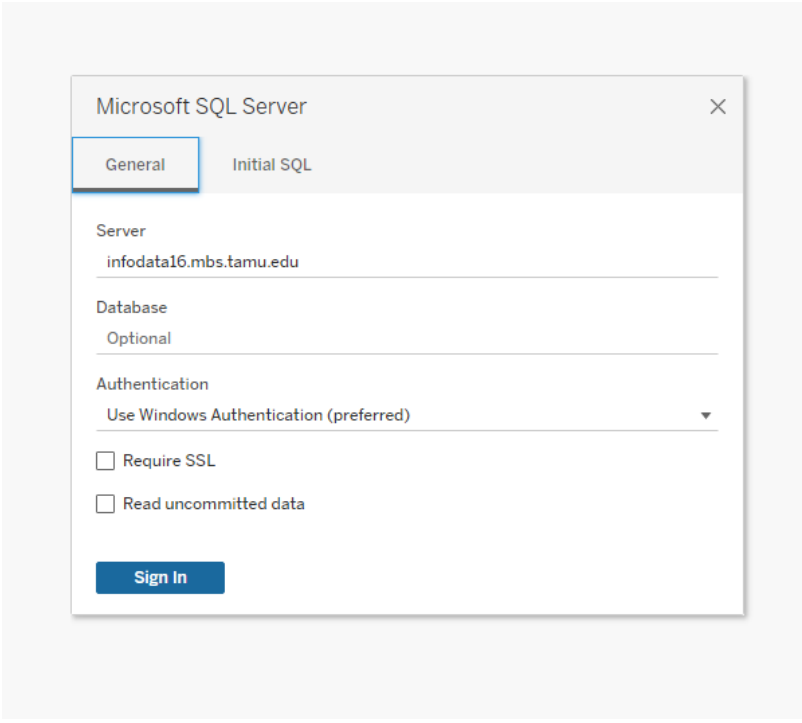
- MS SQL server successfully connected.



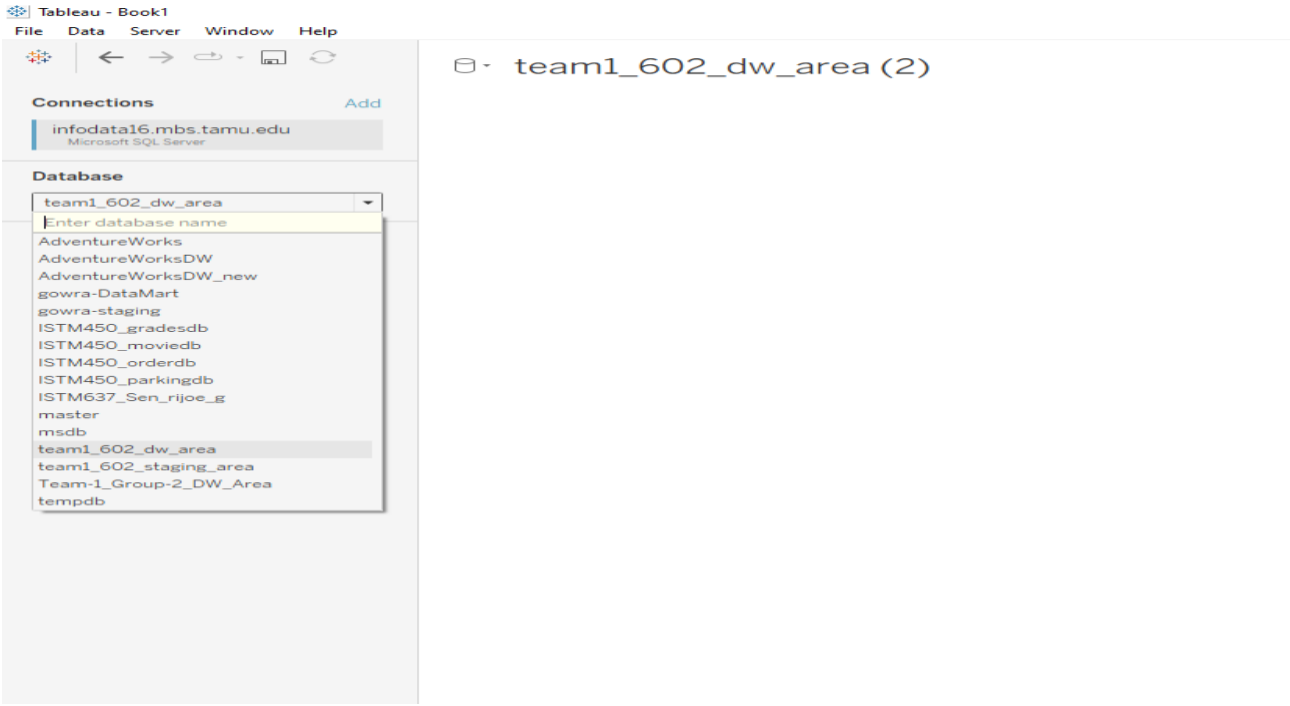
- -



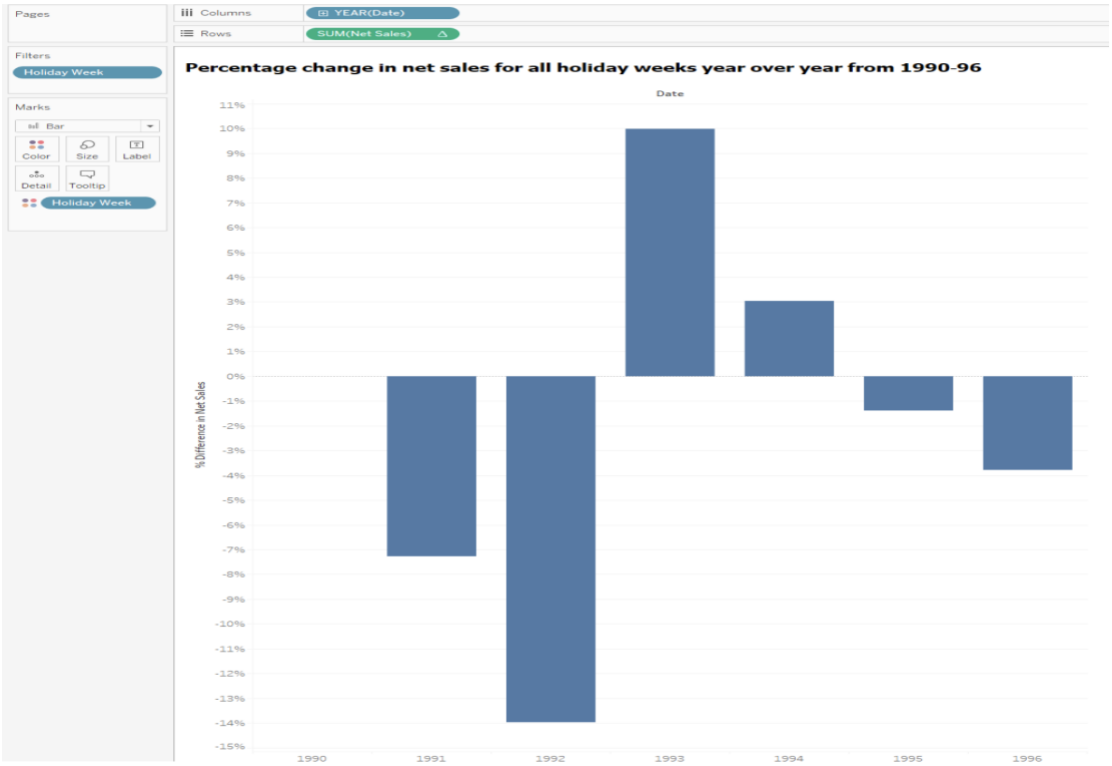
BQ2 - What is the percentage change in net sales for all holiday weeks year over year from 1990-96?



- MS Sql server successfully connected.



- Tableau representation



References:

1. https://en.wikipedia.org/wiki/Dominick%27s
2. https://www.chicagobooth.edu/research/kilts/datasets/dominicks
3. https://www.chicagobooth.edu/-/media/enterprise/centers/kilts/datasets/dominicks-dataset/dominicks-manual-and-codebook_kiltscenter.aspx