Datawarehouse Design and Implementation for Dominick Fine Food

Section Number: 601

Group Number: 11

Jain, Nilesh

Porwal, Priyanshu

Sachdeva, Aakash

INDEX

Introduction:

Dominick's Finer Food (DFF), a retail grocery store chain in the Chicago area has a large business with more than 100 stores dealing in about 3500 products across 30 categories. DFF needs to holistically analyze various aspects in designing strategy to assure its business operates optimally. Our project aims to support DFF in strategic decision making and help in implementation planning by leveraging the insights generated from holistically analyzing DFF's data.

Data warehousing is the area of technology that helps organizations leverage their huge amount of data in different dimensions and angles to draw meaningful trends and insights that can give businesses impetus and an edge in today's competitive environment. The data from various operational sources are combined together in a timely fashion in a structure to help different subject areas of a business to analyze various business metrics in respect to the many different variables and dimensions affecting its performance. This helps businesses fine tune the metrics and attributes to create a strategy that optimizes the outcomes.

Data warehousing requires huge resources and expertise; hence it is a complex area. The process to integrate different sources in a timely fashion, and in a structure that will serve the dynamically changing business requirement is a challenging task that comes under data warehousing.

The project aims to implement a data warehouse for Dominick's Finer Food (DFF). The data that we have is historic data and serves as a platform for this academic project. This is provided by the James M. Kilts Center, University of Chicago Booth School of Business.

The particular data set has a huge amount of data that can be used to understand the performance of various key metrics based on various dimensions useful for the retail industry and moreover, a wide range of other attributes that makes it challenging to align and process in a way to get optimistic and reliable inferences.

We shed light on our understanding of the particular business domain – 'marketing aspect of retail industry'. Which will help us guide Dominick to get the most useful insights DFF would be interested in, to give its businesses a push and more profit. The report explains the data files made available to us for building the data warehouse. We will be analyzing this data set to find trends and insights that can help DFF drive solutions and strategy to increase its overall profit.

Domain Knowledge:

This report targets to study the trends and establish insights for 'the marketing aspects of the retail world'. The retail network is one of the most important *pieces* of the entire commerce and industry chain. The volume of variables, their behavior, and nature are complex, making it a *complicated* domain to analyze various optimal trends and insights. We consider the Dominick's Dataset for our study. The data is a result of the partnership between the Chicago Booth and Dominick's Finer Foods for store-level research into shelf management and pricing. This data is unique for the breadth of its coverage and the information available on retail margins.[1]

Before studying the data and analyzing the trends for Dominick, we dive deep into the domain of marketing aspects of the retail world to understand the retail business, consumer behavior, and business expectations. Consumer behavior and buying habits form a major study for understanding success in this domain. This depends on multiple variables and in an intricate pattern of dependencies. Apart from consumer buying, factors like inventory management and pricing play a major role in driving profit for the retail store and are the ultimate goal of businesses. *A great deal* of research has been carried in the field to understand the methodology of analysis and the impacts of variables on consumer behavior.

Huang, Tao, Robert Fildes, and Didier Soopramanien give a direction by evidence support, to incorporate competitive variables like prices and promotions. This, they prove, will largely *benefit* inventory management and increase profit. The authors point out how using promotional data as ad-hoc information can be inefficient at the same time highlighting the complexities in newer machine learning models. The author suggests a two-step approach to efficiently forecast retail sales using competitive information.[2]

The marketing aspects of the retail industry is highly influenced by the pricing of commodities. The response of consumers based on prices and the rigidity in its movement is often a detailed subject of study in the domain. The world of data warehousing allows us to include the effects of complex variables like package sizing, promotions, and other demographic details in deciphering the pricing model and optimizing profit margins. In this context, Levy, Daniel, Avichai Snir, Alex Gotler, and Haipeng (Allan) Chen point out an interesting insight into products having prices ending with '9'. Prices ending with 9 show rigidity in movement compared to other commodities in the product catalog. Apart from the asymmetric rigidity more insights on consumer behavior related to this pricing were highlighted. Consumers use 9-endings as a signal for low prices, influencing their perceptions, and hence this strategy is also used while increasing the prices of any commodity.[3]

One of the most important papers on the subject is written Jami, Ata, and Himanshu Mishra - "Downsizing and Supersizing: How Changes in Product Attributes Influence Consumer Preferences". They delineate how changes in various product attributes impact and influence the judgment and behavior of the consumers with different permutations and combinations of various attributes. Attributes changes are inevitable, like changing package size to accommodate profits, combine products and sell them as 'bonus buy' etc. The impact of changes in these combinations of different attributes plays a major role in customer judgment and buying behavior among different demographics. This research therefore also stands as a promoter of data warehousing to help understand the complex association of different attributes. [4] Changes in product demand based on the season are a very important factor, maximizing profits by providing the best deals and incentives or price changes during this period serves as a vector to optimize the supply chain and gain maximum profit. In their paper, "Why Does the Average Price of Tuna Fall during Lent?" (2005) [5], Nevo, Aviv, and Hatzitaskos, Konstantinos talk about different commodities and their performance on the shelf during different seasons. It stands as an important study in helping to increase profits.

In our report, we identify 10 such business questions, that can help uncover insights that will help Dominick Finer Food (DFF) optimize key metrics like inventory, sales, price,and gains in

profit. The 10 questions will focus on different dimensions like season, package sizing, demographics, etc. affect the decided metrics. These analysis on a large data set will help give reliable trends to draft future strategies in support to increase business and profit.

Understanding the Data:

In this section we understand the source of the data and its structure.

The Data Source:
The data used for the study is provided by James M. Kilts Center, University of Chicago Booth School of Business.The Dominick's database covers store-level scanner data collected at Dominick's Finer Foods over a period of more than seven years. The data is made available in SAS format and is converted into CSV files by Jens Mehrhoff. Variables have been modified and truncated in the process but yield the same end results. [6]

Data Description:

The data is vast and unique in its breadth of coverage. The database covers approximately 9 years of data throughout all stores in the 100–store chain. It has details on more than 3500 universal product codes (UPCs) spanning across over25 categories. Along with this there are several supporting files that provide demographic, seasonal and store location data to help support the different metrics with more attributes.

The Data is divided in two types of files: category-specific files and general files.

1) General Files:

The general files contain information across all the categories and about the overall performance of products and holistic attributes across the categories. There are two files in this category: The customer count file and store-specific demographics

a. Customer Count File: The file contains daily information of store level information like number of customers visited, purchased, total dollar sales, coupons redeemed etc. across more than 25 categories of products.
b. Store Specific Demographics: The file contains store specific demographic data. The data has been originally provided by the U.S. government census and helps create demographic profile for each DFF store

2) Category Specific Files:

a. UPC File: There are different files for each 29 category containing one record for each UPC. These files are named as UPCXXX where XXX stands for the specific category. The file provides us with information regarding identifier codes, product name, size etc. regarding a particular UPC
b. Movement Files: The movement files are divided at category level like the UPC files and include weekly sales data for each UPC in each store for over 5

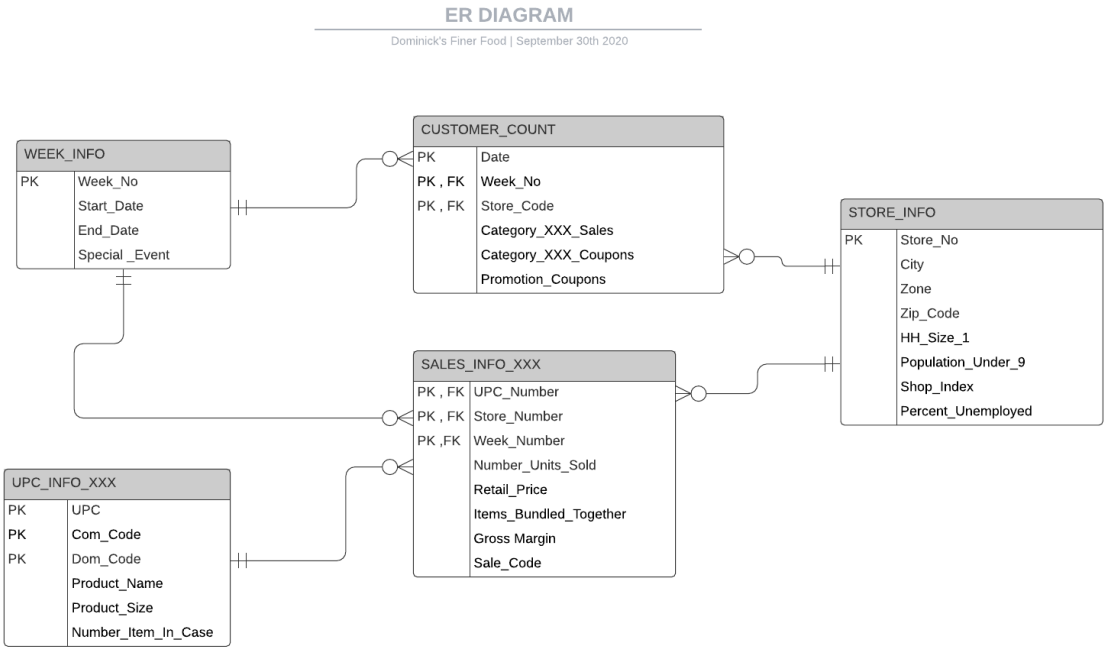
years. The attributes included in this file are price, unit sold, profit margin, deal code, etc. The files are sorted by UPC, store, week.

Other important data:

1) Dominick Store locations: This file has all store data like the city, zonal information, the price tier and the address information
2) Map of Dominick stores: Above information is supplemented by map of Dominick Stores
3) Week Decode table: This table contains includes information of the week like start date, end date, special events marked on those days , mapped to a week number used in the other files mentioned above.

Below is an ER Diagram to help us understand the fields of the data files and how they are related to each other.

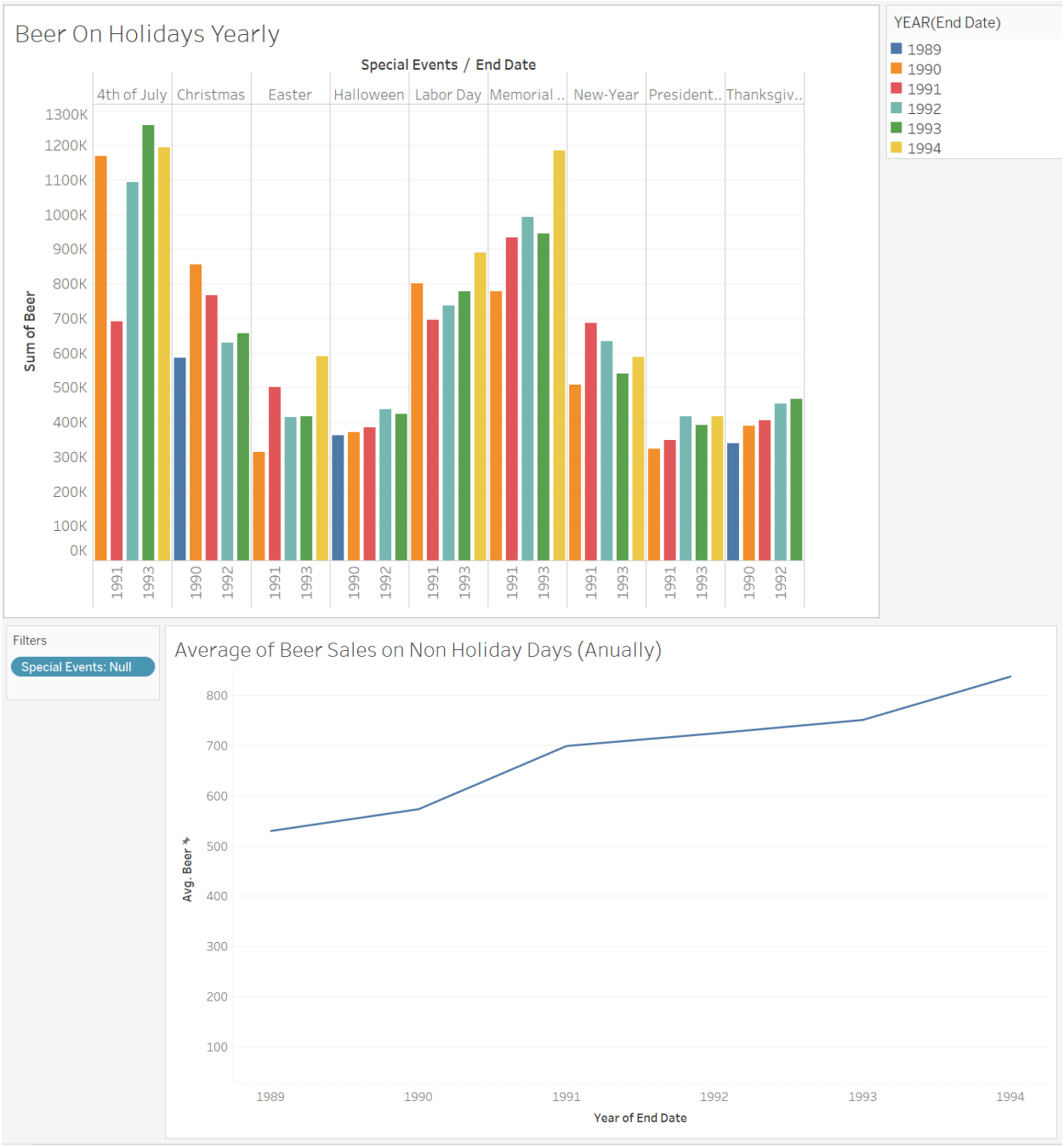
ER Diagram:



1) Category_XXX_Sales and Category_XXX_Coupons in the CUSTOMER_COUNT table represent different columns for different category
2) There are separate table for different category for UPC_INFO_XXX and SALES_INFO_XXX
3) The movement files mentioned above are named as SALES_INFO_XXX tables

Business Questions:

1. Holiday season is a peak time for sales, determine the sales of *Beer (or some other product)* during these times.



Justification: Retail stores do target specific seasons for most of their sales and revenue generation and it will be beneficial to get some analysis on the holiday season. The holiday season will include all the major holidays in the United States like Thanksgiving, Halloween, Christmas, New Year, Easter, Memorial Day, and so on. There should be more consumption of

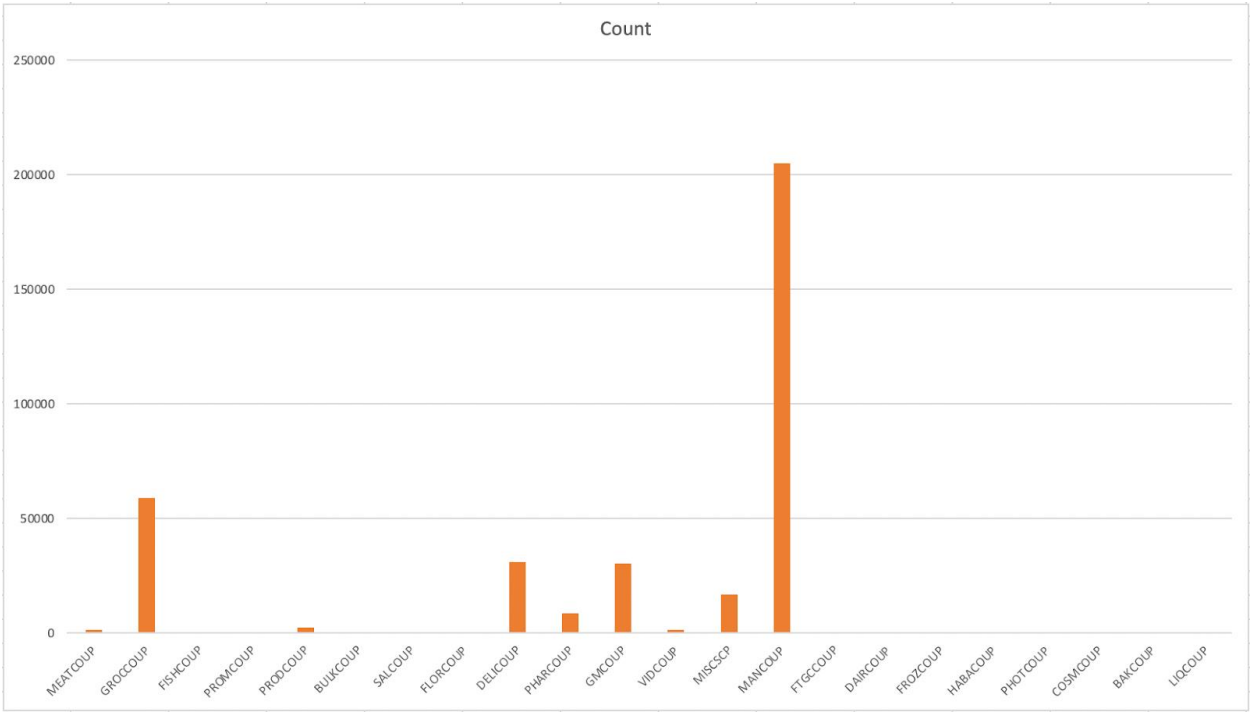
beer during these times which directly implies that the sales of beer should increase. Our analysis of average sales on non-holiday days (annually) and that on holiday days shows how some holidays have more beer sales while others have less. This analysis will provide great insights on the purchase of beer during the holiday season and how the retail stores can be best prepared for such a peak.

Data Snapshot:

STORE	DATE	DATE	WEEK	BEER
2	881212	12/12/1988	-39	1.99
2	890106	1/6/1989	-35	0.42
2	890109	1/9/1989	-35	0.5
2	890115	1/15/1989	-34	0.5
2	890129	1/29/1989	-32	3.99
2	890222	2/22/1989	-29	0.54
2	890306	3/6/1989	-27	2.99
2	890310	3/10/1989	-26	1.98
2	890501	5/1/1989	-19	1.27
2	890528	5/28/1989	-15	1.05
2	890707	7/7/1989	-9	1.5
2	890818	8/18/1989	-3	7.63
2	890906	9/6/1989	-1	1.49
2	890912	9/12/1989	0	-0.98
2	900326	3/26/1990	28	2.27
2	900620	6/20/1990	40	0.75
2	900927	9/27/1990	55	0.99
2	901210	12/10/1990	65	1
2	901219	12/19/1990	66	3
2	901223	12/23/1990	67	37.18

Week	Start Date	End Date	Special Events
1	9/14/1989	9/20/1989	
2	9/21/1989	9/27/1989	
3	9/28/1989	10/4/1989	
4	10/5/1989	10/11/1989	
5	10/12/1989	10/18/1989	
6	10/19/1989	10/25/1989	
7	10/26/1989	11/1/1989	Halloween
8	11/2/1989	11/8/1989	
9	11/9/1989	11/15/1989	
10	11/16/1989	11/22/1989	
11	11/23/1989	11/29/1989	Thanksgiving
12	11/30/1989	12/6/1989	
13	12/7/1989	12/13/1989	
14	12/14/1989	12/20/1989	
15	12/21/1989	12/27/1989	Christmas
16	12/28/1989	1/3/1990	New-Year
17	1/4/1990	1/10/1990	
18	1/11/1990	1/17/1990	
19	1/18/1990	1/24/1990	
20	1/25/1990	1/31/1990	
21	2/1/1990	2/7/1990	
22	2/8/1990	2/14/1990	
23	2/15/1990	2/21/1990	Presidents Day
24	2/22/1990	2/28/1990	

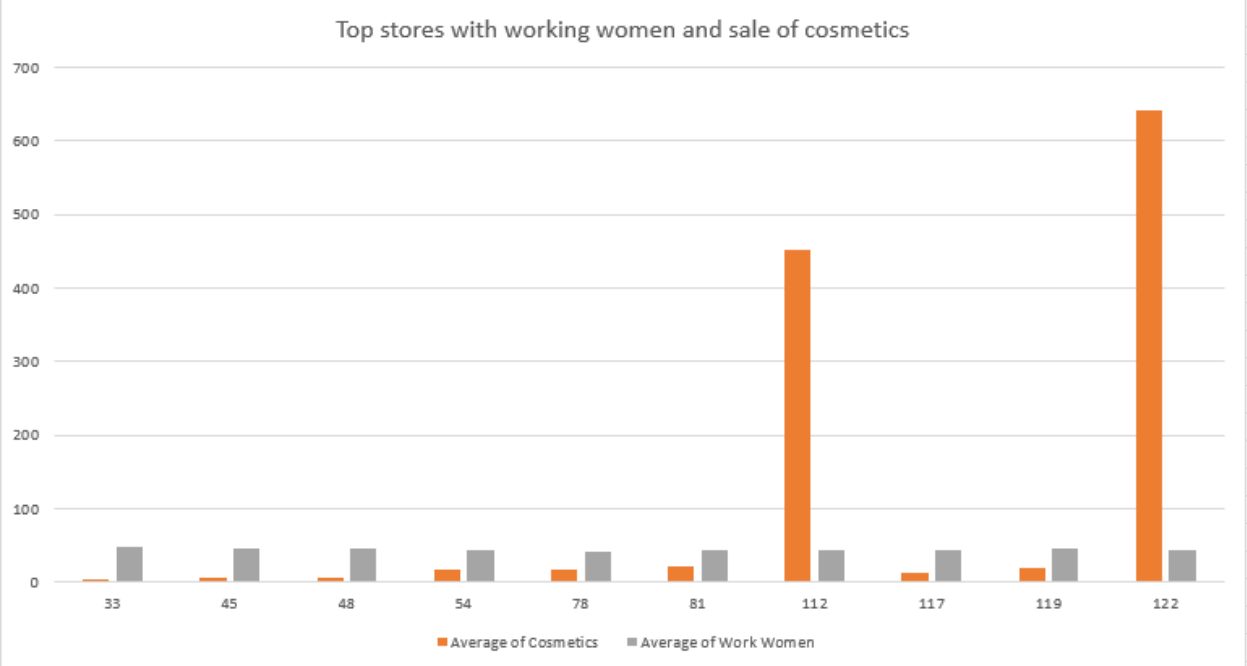
2. Compare the coupons that are most used by consumers in the stores.



Coupon	Count
MEATCOUP	1284.29
GROCCOUP	58940.69
FISHCOUP	14
PROMCOUP	14
PRODCOUP	2430
BULKCOUP	0
SALCOUP	1
FLORCOUP	250
DELICOUP	30986
PHARCOUP	8530
GMCOUP	30460
VIDCOUP	1263
MISCSCP	16739
MANCOUP	204926
FTGCCOUP	0
DAIRCOUP	0
FROZCOUP	0
HABACOUP	0
PHOTCOUP	0
COSMCOUP	0
BAKCOUP	0
LIQCOUP	0

Justification: As the retail stores provide coupons to customers for their purchases, it will be a good way to analyze whether these coupons are used and by how much. Comparison of the coupons can also give deeper insights on the ones that are most used and the ones that are least used. The managers can look at the answers from this analysis to decide on the further use of particular coupons and to what extent they can be used. They can also use these as promotions and attract the customers to generate greater revenues.

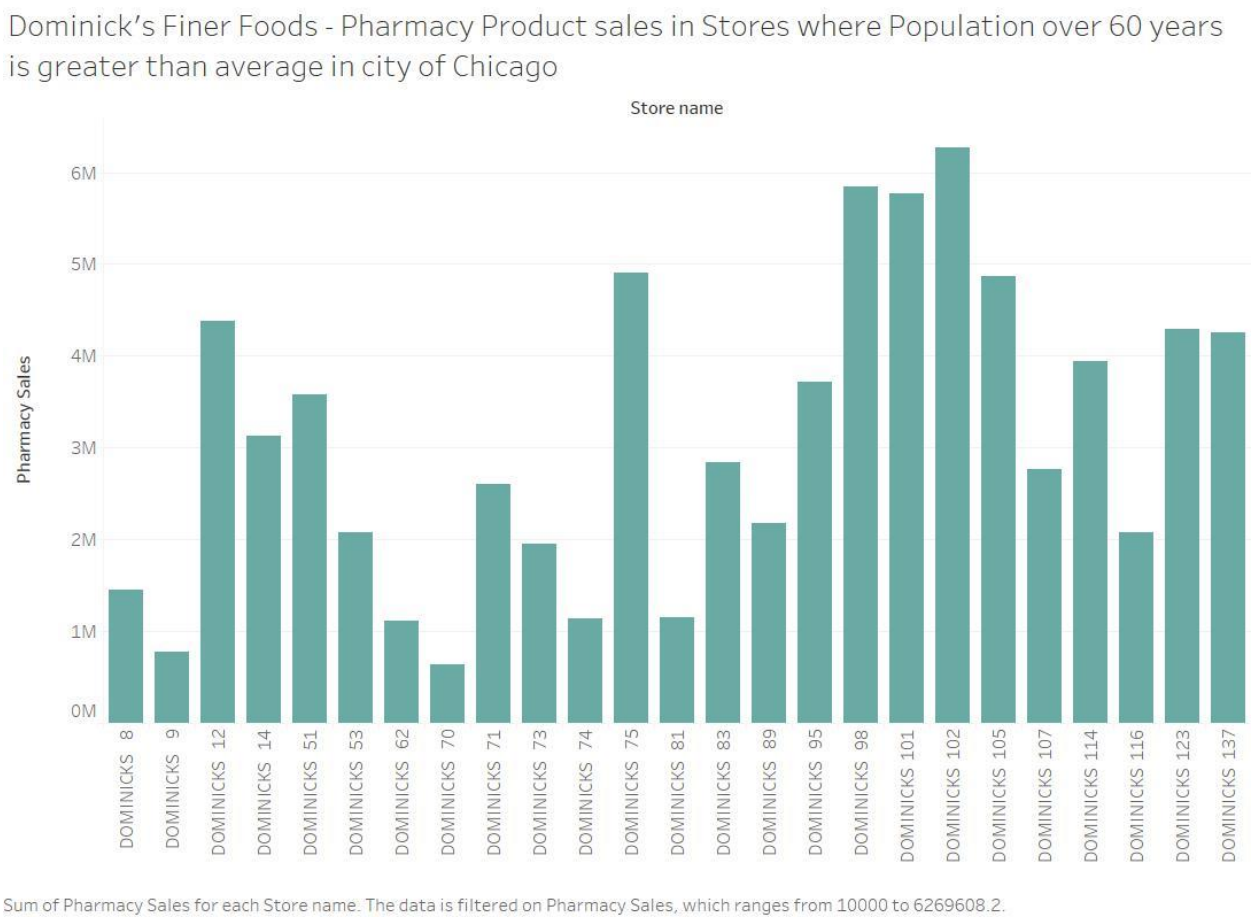
3. Which ten stores have the most population of working women and what is their sale of cosmetic products?



Row Labels	Average of Cosmetics	Average of Work Women
33	4.38	47.23
45	6.71	46.41
48	6.39	45.97
54	16.27	43.67
78	16.47	42.4
81	21.7	42.95
112	451.15	44.01
117	13.81	43.91
119	18.34	46.22
122	641.26	42.59

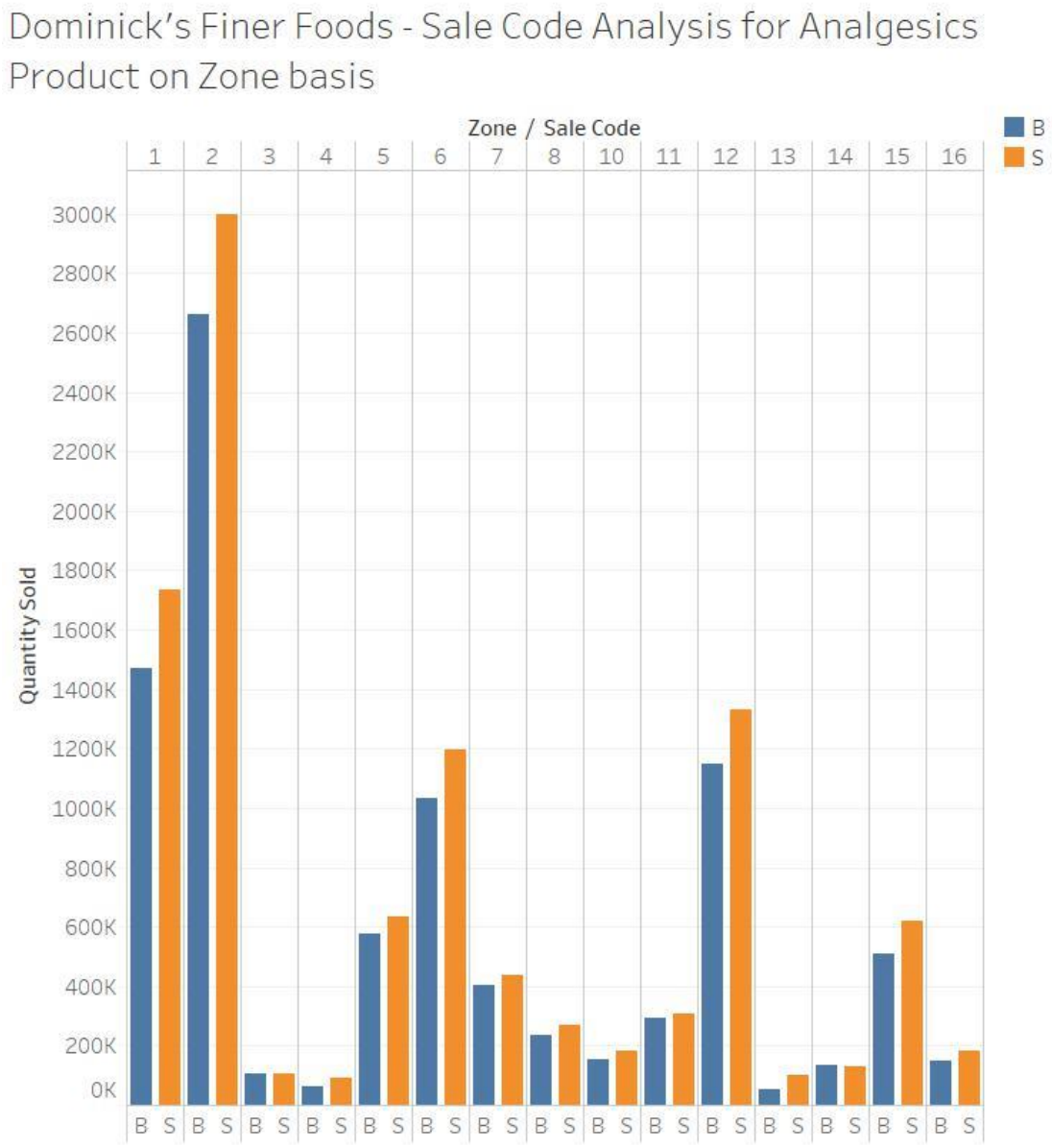
Justification: It is of most interest to the retail stores to determine the top performing stores in a particular area, so that they can sell it more to earn greater profits. Along with this, the retail stores have a target customer set and they can use to analyse the product sales of cosmetic products. This ensures two things, one is more sales of the product and in turn more revenue for the retail store, and the second is customer satisfaction. Thus, analysis of the top ten stores helps in decisions in support of the market. Comparison of the sales of cosmetic products is also justifiable as it helps the managers take decisions on increasing their inventory or increasing their price to never run out of the most in demand products and also increase sales.

4. What are the sales across different stores for pharmacy products having a population of 60 years?



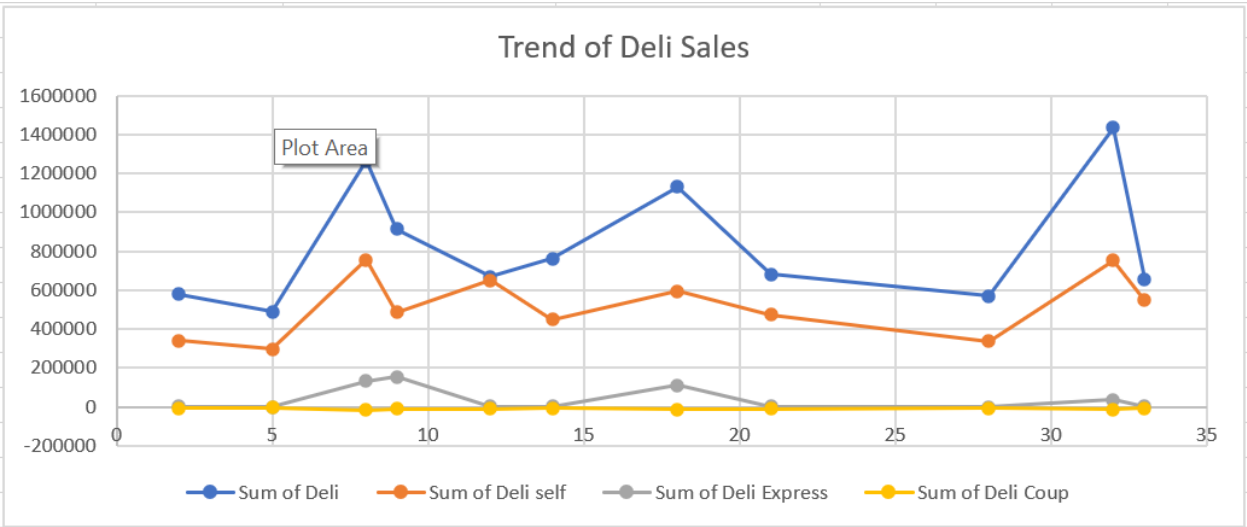
Justification: To boost the sale of a specific product category, it is critical to analyze its sales in a particular demographic. More importantly, it will provide greater value if we can target the age group of consumers and analyze their purchase. It is true that all age groups need pharmacy products from time to time, but people over 60 years may use these products more often and hence this analysis will give the managers good decision-making opportunities. In case the sales are low, the retail store can provide offers for certain age groups and if the opposite is true, there are possibilities of increasing the price or running schemes to attract the consumers more.

5. How do bonus buy and simple price reduction for products sold on promotion compare for *analgesics (or other product category)*?



Justification: The retail stores run promotion from time to time to sell more products and two promotions were Bonus buy and Simple price reduction. It will be beneficial to learn about the impact of these promotions and compare them. The comparison of these two promotions will help determine which promotion is more successful and should be continued further. It is also possible that both the promotions match and increase the revenues equally well, in that case it is a good idea to know the quantity being sold through each promotion.

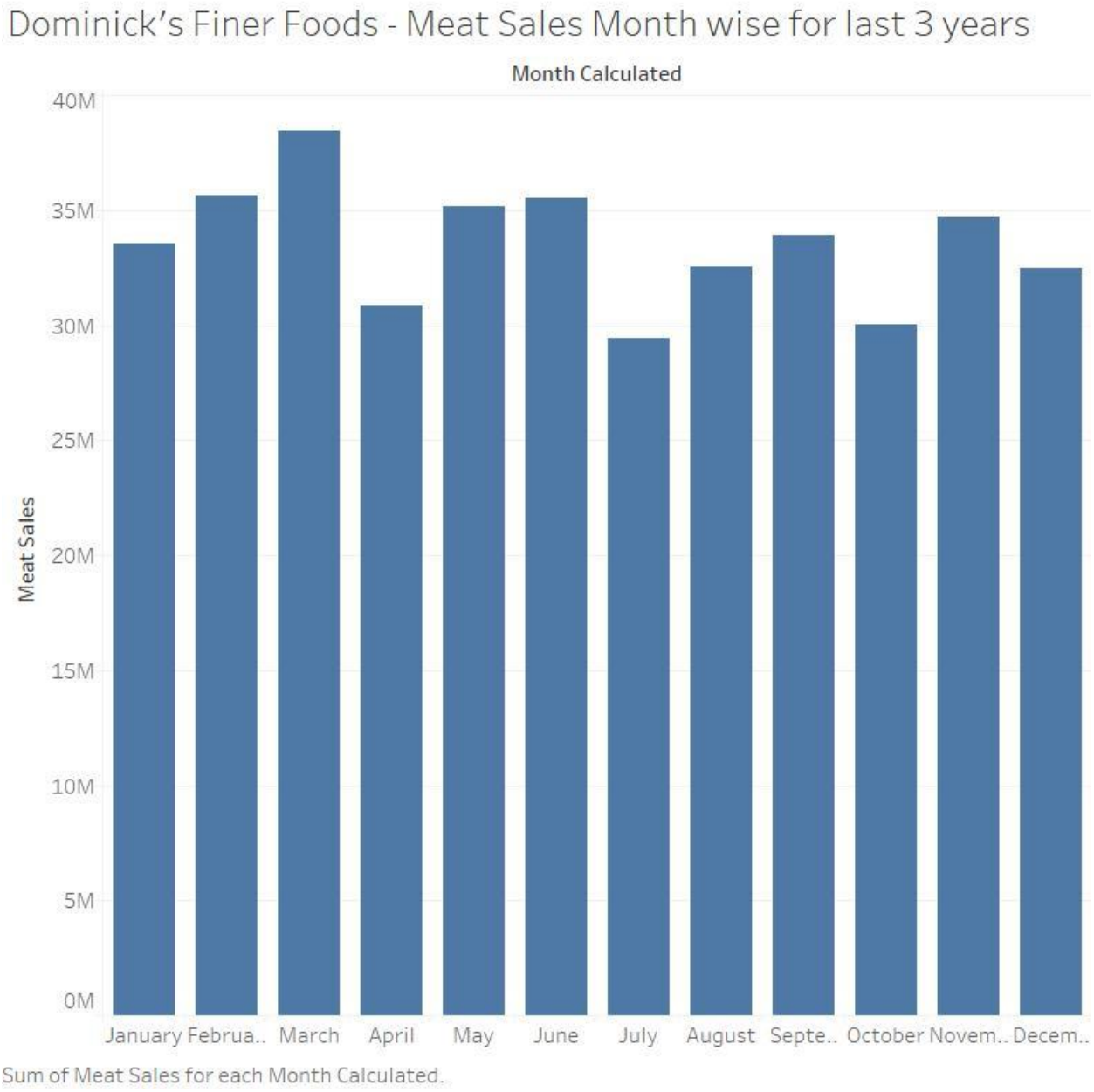
6. Compare Deli Sales, Deli Coupons, Deli self-service sales, and Deli Express and show their trends for the different stores in Chicago during a year.



Row Labels	Sum of Deli	Sum of Deli self	Sum of Deli Express	Sum of Deli Coup
2	578276.03	339260.39	463.87	-6692.63
5	488012.69	296548.29	0	-4371.3
8	1265516.11	755014.41	131697.9	-16626.93
9	914038.83	485297.26	155240	-11033.28
12	670210.39	651104.44	1363.08	-11072.75
14	763301.43	448620.34	173.16	-8481.77
18	1130445.02	593923.92	111401.42	-13018.77
21	680841.33	473501.44	640.65	-10533.9
28	569366.39	336684.1	75.44	-6536.82
32	1433731.01	752832.44	34739.58	-14026.5
33	656589.8	546393.82	408.06	-7127.75

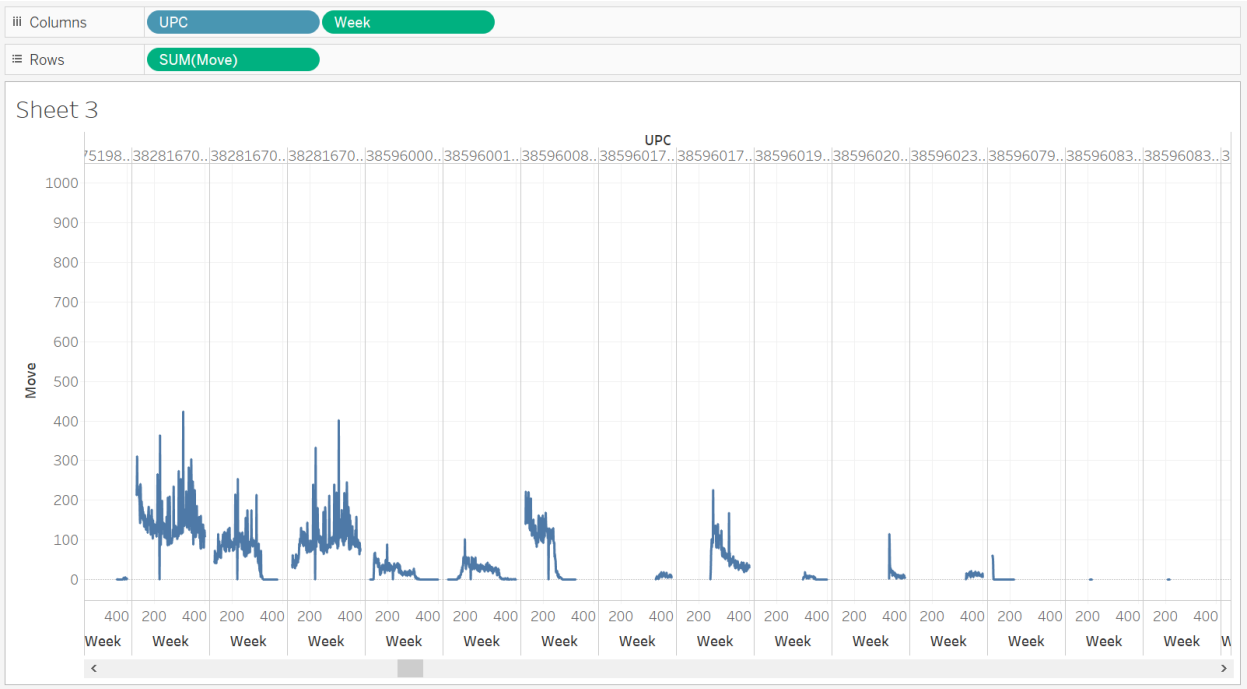
Justification: The analysis of Deli sales through their trends will help in identifying the sales of products across different stores. This is useful for decision making during inventory management and gives insight on the sub-categories. The market strategy can be planned using the comparison of Deli Sales, Deli Coupons, Deli self-service sales, and Deli Express. This can also be useful for understanding and best planning the coupons and their promotion.

7. Determine the month during which the meat sales were highest and lowest for the last 3 years?



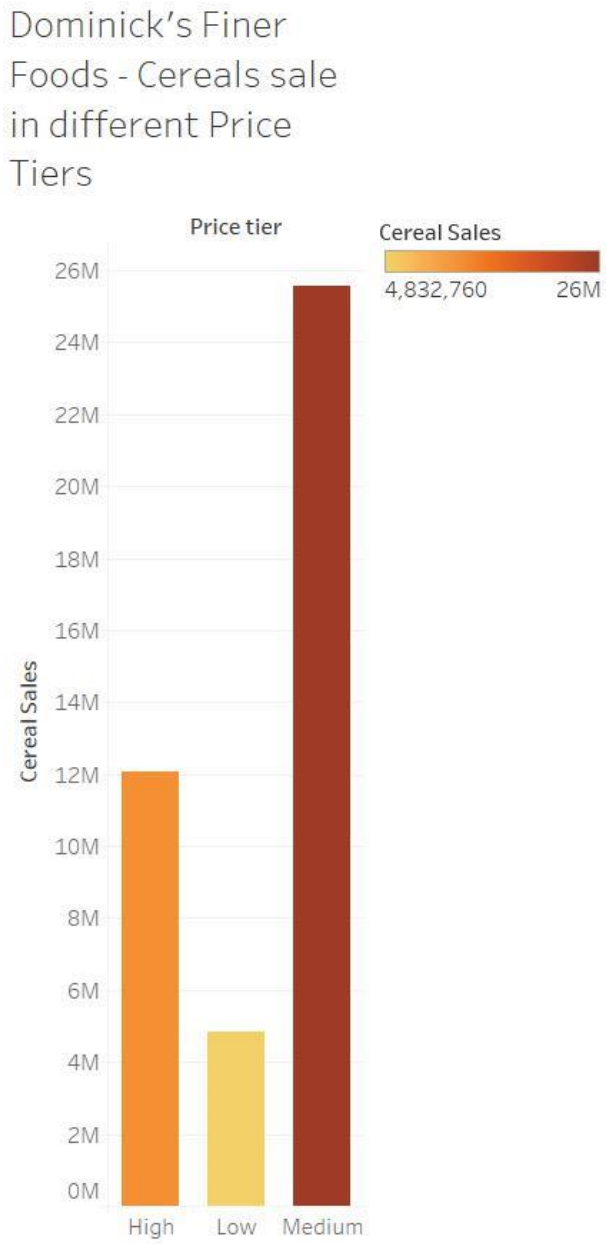
Justification: The given business question is similar to the question where we try to find the pattern in the sales of a product during the festival season. Finding the trends in the meat sales by analyzing the past 3 years data can facilitate the managers in efficient inventory planning and management. It can also help them to understand the root cause of the low sales and what additional steps like promotional campaigns can be taken to ensure increase in the sales.

8. Determine the bath soap product which has a decreasing sales trend across different zones?



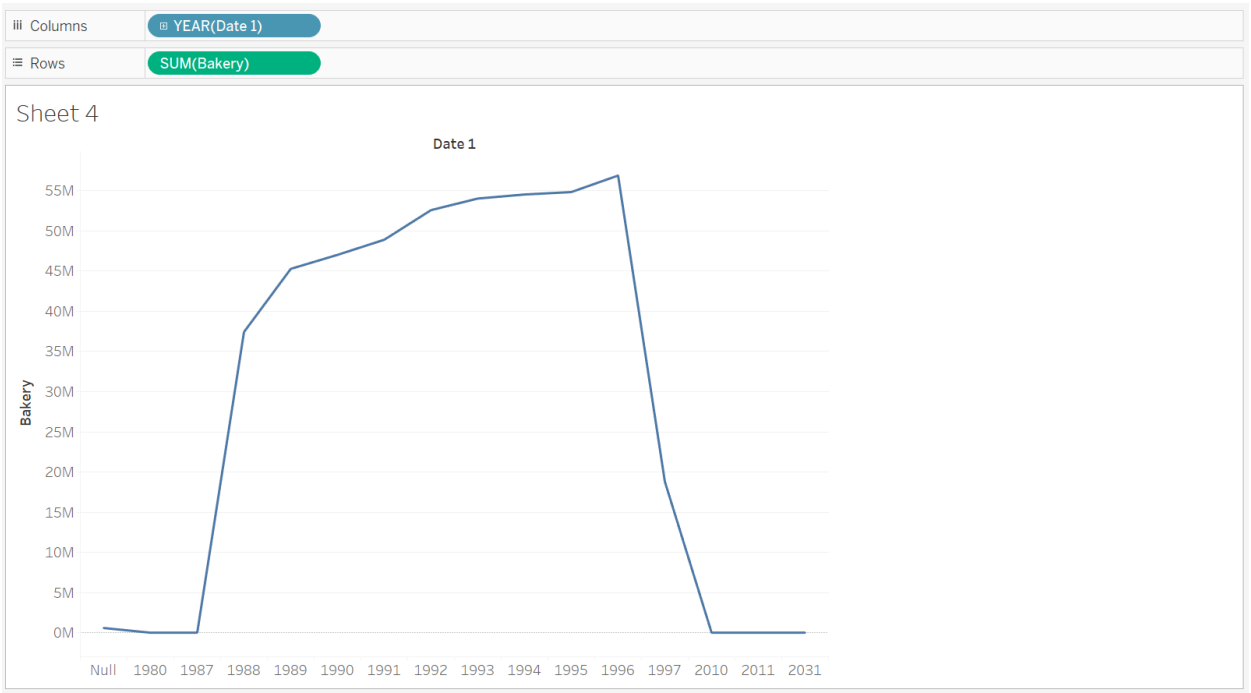
Justification: The given business question can act as a model for analyzing the sales trend of a particular product across different zones. It will help the managers to understand the sales pattern of various products and which products need special attention. Coming out with special offers and marketing campaigns for these products will help to increase the overall sales. Also, understanding the reasons for the lackluster sales of the products can help to solve the challenges.

9. What is the cereal sales in a store in different price tiers over the whole period of time?



Justification: Dominick Fast Food adopts a multi-tier pricing strategy for products. The given business question will help to analyze the sales of cereals in different price tiers and to understand which price tier is suitable for a certain store. Understanding the sales patterns based on different pricing tier and stores demographic will further help managers to analyze the fast-moving products based on the pricing.

10. How is the growth of Bakery from the year 1990 to 1996?



Justification: To understand the consumer base for a certain product category, a retail industry needs a continuous study of products and their categories based on regions. It is therefore obvious to evaluate the sales of the bakery category and it has been rising in recent years to introduce strategies to access more goods and supplies at any time in any shop. DFF needs to analyze the growth of the bakery category from 1990 to 1996 in order to understand the trends and then strategize accordingly to fully harness the potential of this category.

1. LOGICAL DESIGN

1.1 Requirement Prioritization

The questions identified above will help the different departments of DFF like marketing and research, sales and finances etc. Based on the most important insights required by DFF and its departments, 5 key questions have been prioritized, which are as follows:

2. Compare the coupons that are most used by consumers in the stores. [Marketing & Promotions]
3. Which ten stores have the most population of working women and what is their sale of cosmetic products? [Sales]
5. How does bonus buy and simple price reduction for products sold on promotion compare for beer (or other product category)? [Sales, Marketing & Promotions]
6. Compare Deli Sales, Deli Coupons, Deli self-service sales, and Deli Express and show their trends for the different stores in Chicago during a year. [Sales, Marketing and promotions]
7. Determine the month during which the meat sales were highest and lowest for the last 3 years? [Sales]

To address these needs of providing insights for strategic decision making to different departments which could also benefit DFF as whole, we choose to create data warehouse using kimball's approach of making conformable data marts. In the sections below we present logical design plan to implement this data warehouse.

1.2 Information Package:

The above requirements can be summarized into the below information package, which can help us in further designing our conformable data marts

Time	Store	Product Category	Coupon Category
Month	Store Name	Category	Category
Year	City		
	% Working Women		
Facts: Coupons Redeemed, Sales of products, promotions			

1.3 Data Mart Matrix

	Time Dim	Store Dim	ProductCategory Dim	CouponCategory Dim
Promotions	x	x		x
Sales	x	x	x	

1.4 Conceptual Design

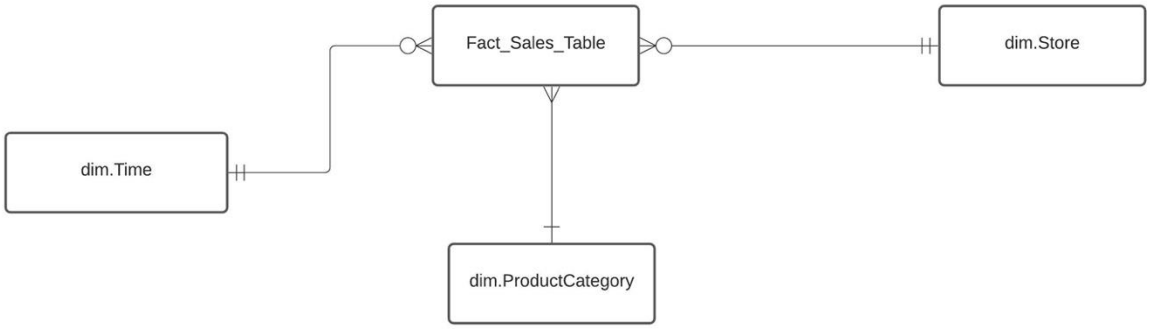


Figure 1: Conceptual Schema for Sales Data Mart

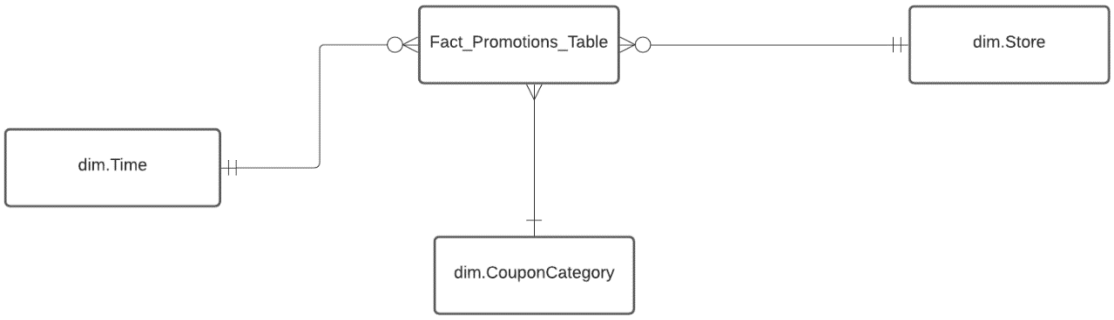
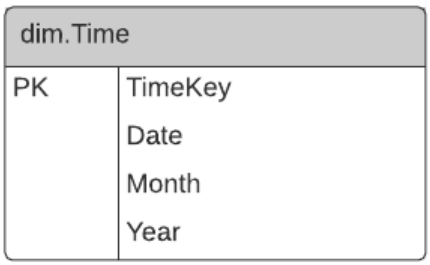


Figure 2: Conceptual Schema for Promotions Data Mart

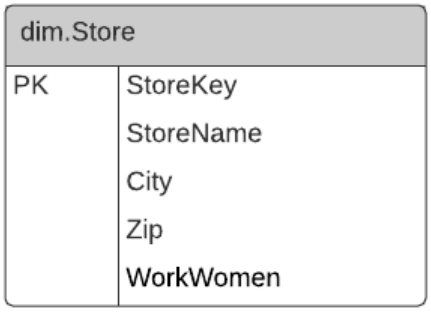
1.4 Dimension Table Details:

1. Time (dim.Time):
   This table will help to provide the time dimension to our facts. The granularity of the dimension depicted via the hierarchy and are the attributes of the dimension table. We store the data at daily granular level to facilitate fine analysis which can be aggregated up to a year's period.



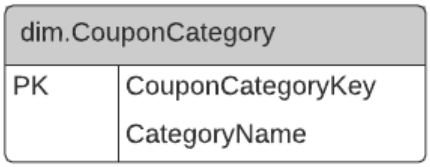
   Below is the detailed description of various attributes of time dimension table:
   TimeKey: This is a surrogate key and serves as an unique identifier for the time dimension
   Date, Month and Year serve as the hierarchical attribute of the table to help fine tune analysis.

2. Store (dim.Store):
   The store dimension table is used to provide all the store related information for DFF . It contains information that is used for reference by the Sales Fact table and promotions fact table.



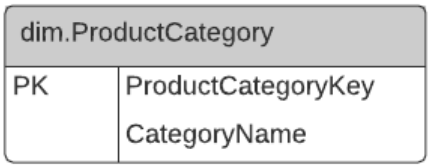
   Below is the detailed description of various attributes of time dimension table:
   StoreKey: This is a unique identifier by the store dimension table which is a surrogate key and is generated as the row number.
   StoreName: This is the unique store name provided by DFF to each of its store in the Chicago are
   City: This attribute displays the city the store is located in.
   Zip: This attribute will provide us with the zip code of the store
   WorkWoman: This attribute provides us with an important insight about the percent of working woman in the region where the store is located

3. Coupon Category (dim.CouponCategory):
   This dimension table provides us with the category of coupons that were issued by the DFF



   Below is the detailed description of various attributes of time dimension table:
   CouponCategoryKey: This is the surrogate key that is generated to represent as an unique identifier to each category
   CategoryName: This attribute identifies the name of the category for which coupon was issued

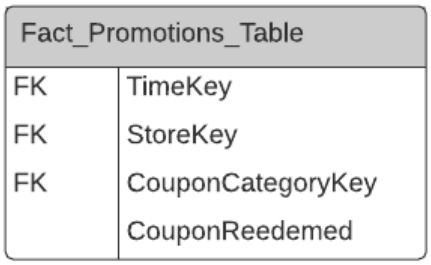
4. Product Category (dim.ProductCategory)
   This dimension table provides us with the category of that products being sold by the DFF



   Below is the detailed description of various attributes of time dimension table:
   ProductCategoryKey: This is the surrogate key that is generated to represent as an unique identifier to each product category
   CategoryName: This attribute identifies the name of the product category

1.5 Fact Table Details:

1. Promtions Fact Table
   This fact table contains the data of the number of coupons redeemed across each category at store level on a daily basis.

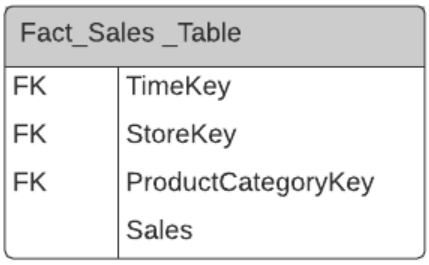


   As mentioned above the granularity of this table will be daily data for each store. This is done to facilitate fine analysis.
   The structure of the table depicts a single row of coupons redeemed for a single day at a single store for a single category. These attributes form the columns of the table as shown above and their definition and sources are provided below:
   The three foreign keys coming from the dimTime, dimStore and dimCouponCategory will serve as the composite primary key for Promtions Fact Table

   Fact Table Keys:
   TimeKey: This is a unique identifier used for time and it is the row number that is generated as a surrogate key in dimTime. It serves as a foreign key for this fact table.
   StoreKey: This is a unique identifier by the store dimension table which is a surrogate key and is generated as the row number. It serves as a foreign key for this fact table.
   CouponCategoryKey: This is a unique identifier of the category dimension table which is a surrogate key and is generated as the row number

   Fact Table Measures:
   CouponsRedeemed: This indicates the number of coupons redeemed for each category across days and stores.

2. Sales Fact Table:
   This fact table contains the data of the sales (in $) of each category at store level on a daily basis.



As mentioned above the granularity of this table will be daily data for each store. This is done to facilitate fine analysis.
The structure of the table depicts a single row of dollar sales for a single day at a single store for a single category. These attributes form the columns of the table as shown above and their definition and sources are provided below:
The three foreign keys coming from the dimTime, dimStore and dimProductCategory will serve as the composite primary key for Sales Fact Table

Fact Table Keys:
TimeKey: This is a unique identifier used for time and it is the row number that is generated as a surrogate key in dimTime. It serves as a foreign key for this fact table.
StoreKey: This is a unique identifier by the store dimension table which is a surrogate key and is generated as the row number. It serves as a foreign key for this fact table.
ProductCategoryKey: This is a unique identifier of the category dimension table which is a surrogate key and is generated as the row number

Fact Table Measures:
Sales: This indicates whether the product was sold on what kind of promotion

1.6 Star Schema:

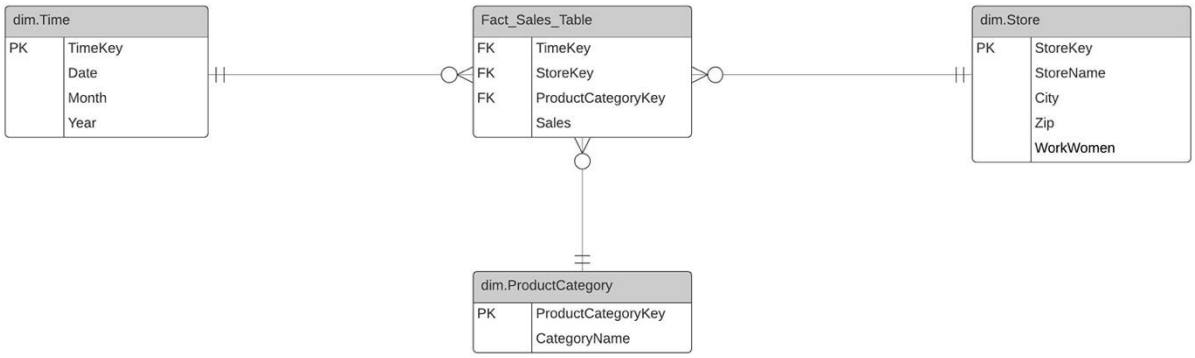


Figure 3: Star Schema for Sales Data Mart

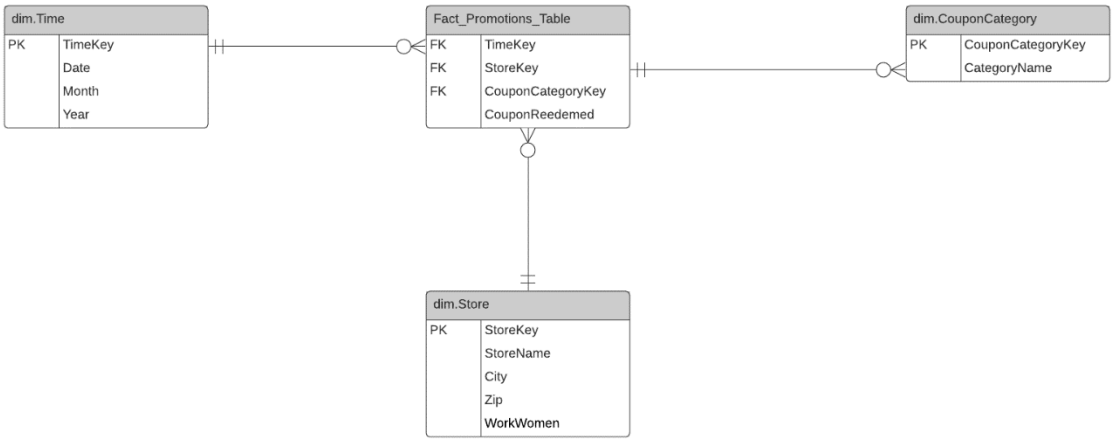


Figure 4: Star Schema for Promotions Data Mart

1.7 Business Question Justification at Data Mart Level

1. Compare the coupons that are most used by consumers in the stores.

This question concerns the marketing and promotional team to help them understand how many coupons were used and which coupons drive was successful. The Promotions data mart serve as a conformable data mart for this particular requirement. The FactPromotionTable provides with important data of number of coupons redeemed and along with the Dim.CouponCategory dimension table. The dim.Time and dim.Store dimensions can be further used to refine the analysis and show in more details analysis

2. Which ten stores have the most population of working women and what is their sale of cosmetic products?

This business concern helps to define consumers' profiles even more effectively, in order to exploit the target market with advertising strategies at each shop level. The Sale Data Mart can be used to answer this business question that can be of great help for the sales team. The dim.Sales dimension will provide the information regarding the percentage of working woman in the store locality and dim.ProductCategory will help to identify the cosmetics category. Various other demographic information can be similarly used to answer other business queries with this data mart and can be easily update with data as per the business requirements

3. How do bonus buy and simple price reduction for products sold on promotion compare for a product category?

This business question will help the marketing and promotion team to address study the success of store level promotional activity. The business questions will be solved across the two data marts providing us with the information on corresponding sales detail in Sales Data Mart for a particular promotional activity of a category that will be derived from product and coupon category dimensions

4. Compare Deli Sales, Deli Coupons, Deli self-service sales, and Deli Express and show their trends for the different stores in Chicago during a year.

The business questions help us identify the sales and various other trends on the Deli sales. This question can be addressed by the sales data marts for sales and promotions data mart for coupons trends.

5. Determine the month during which the meat sales were highest and lowest for the last 3 years?

This is a business question which can arise most frequently and be of great help for managers and strategic decision makers. The sales data mart can be used to answer this question. To get the result we will be using the Fact_Sales_Table along with dim.ProductCategory and dim.Time dimensions to slice the required data by the managers.

1.8 Data Mapping Table

Time Dimension Table	**DW Dimensional Attribute**	**Source Table**	**Source Table Attribute**	**Mapping function**	**Generated Key**
	TimeKey				Surrogate Key
	Date	CCOUNT	Date		
	Month	CCOUNT	Date	It will be derived from the date attribute	
	Year	CCOUNT	Date	It will be derived from the date attribute	
Store Dimension Table	**DW Dimensional Attribute**	**Source Table**	**Source Table Attribute**	**Mapping function**	
	StoreKey				Surrogate Key
	StoreName	DEMO	Name		
	City	DEMO	City		
	Zip	DEMO	Zip		
	WorkWomen	DEMO	WorkWom		
Product Category Dimension Table	**DW Dimensional Attribute**	**Source Table**	**Source Table Attribute**	**Mapping function**	
	ProductCategoryKey				Surrogate Key
	Category Name	CCOUNT		Categories available in the CCOUNT table	
Coupon Category Dimension	**DW Dimensional Attribute**	**Source Table**	**Source Table Attribute**	**Mapping function**	
	CouponCategoryKey				Surrogate Key
	Category Name	CCOUNT		Categories available in the CCOUNT table	
Promotions Fact tables	**DW Dimensional Attribute**	**Source Table**	**Source Table Attribute**	**Mapping function**	
	TimeKey				Surrogate Key
	StoreKey				
	CouponCategoryKey				
	CouponReedemed	CCOUNT		Coupons available in the CCOUNT table	

Sales Fact tables	DW Dimensional Attribute	Source Table	Source Table Attribute	Mapping function	
	TimeKey				Surrogate Key
	StoreKey				Surrogate Key
	CouponCategoryKey				Surrogate Key
	Sales	CCOUNT		Sales available in the CCOUNT table	

2 Physical Design Plans:

2.1 Standardization Plan:

This is the first and basic step of the physical design process for the data warehouse. To maintain consistency and make it easier for users, it is necessary to establish standards. An undefined naming convention for data warehouses can increase complexities and minimize usability [1]. The importance of having a clearly defined standards is further strengthened in the case of data warehouse since multiple users uses the object names of the data warehouse for multiple use cases and hence clearly defined standards can certainly make the job a lot easier. Below are the basic guidelines for establishing a data warehouse standard:

- A clear and consistent naming convention should be followed for naming the fact tables, dimension tables and corresponding attributes
- The data stored in the table should, without exception, be according to the specified data type
- There should not be any null values in the mandatory fields
- While using a primary key for different instances in a table, its uniqueness should be maintained

Standards to be followed in the DFF Project:

1) Naming of Database Object:

Database Objects	Standard
Data Mart	Department_Data_Mart This will be used to identify the conformable data marts created using kimball's approach (eg: Sales_Data_Mart)
Fact Table	Fact_*Department*_Table upper camel casing convention must be followed (eg: Fact_Sales_Tables)
Dimension Table	dim.*DimensionName* The *DimensionName* will be given in upper camel case (eg: dim.Time, dim.CouponCategory)
Column name	AttributeName This will be in upper camel case. If the column is a unique identifier or part of group of unique identifier columns, it must end with 'Key' (eg: TimeKey, Year)

2) Source Code Files
All the source related files and documents will be having the convention defined by the respective OLTP systems, any files generated as source file specifically for data warehouse purpose will be named as Source_SourceObject

3) Staging Files
The files in staging area will be named using the below convention:
Stage_object_step_id_step_name

Breaking down the different parts of staging file and table names:
a. *Stage*: This represents the file and table is of staging area
b. *Object*: This will be the name of the datawarehouse object the staging file will be moving into like the data mart table.
c. *Step_id*: This represents the step used in transformation chain to generate the file
d. *Step_name*: This represents a logical name for the step used to generate the file

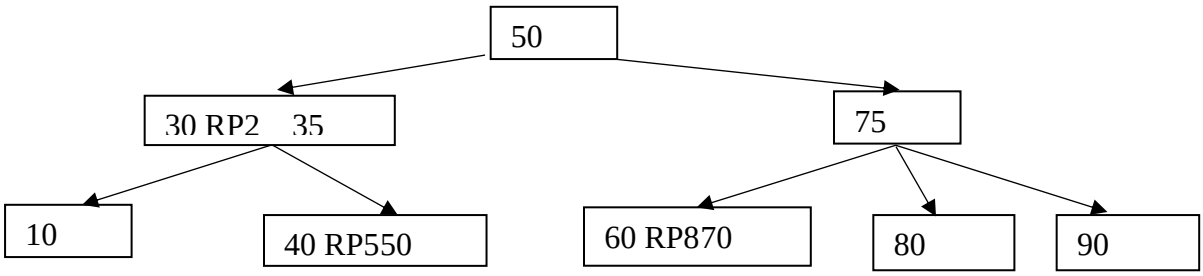
2.2 Aggregate Plan:

The data stored in the tables is at atomic level to provide flexibility for queries and being able to adapt the new demands of analysis and querying. But from business perspective most of the time the analysis is done on the summarized data. Thus, if we store summarized data as well, it enhances the performance and reduces the time to run queries on the datasets [2]. As per the business requirements and after analyzing the hierarchies of the table, the aggregation will be done for two instances

1. The sale data in FACT_SALES_TABLE will be aggregated over the time dimension to reflect the annual sales across different stores and category by utilizing the cube structure. This as per requirement gathering would be very important summarization in for analyzing annual sales trend across stores.
2. Similarly, coupons data in FACT_PROMOTIONS_TABLE will be aggregated over store and category dimensions at monthly time level to get desired insights into promotional activities. The aggregation plan mentioned here, utilizes the cube design trait of the star schema to provide summarized results that can help optimize the performance of analysis query.

2.3 Indexing Plan:

The primary purpose of online analytical processing is to analyze data through queries as per the business requirements. The voluminous data is regularly used for validation and analysis therefore it is important to have a convenient process to fetch the data. The efficacy of such process can be achieved through indexing [1]. Many things like indexing key, number of indexes, data to be indexed etc. needs to keep in mind while deciding on how to index the fact and dimension tables. For the purpose of indexing the fact and dimension tables of our data warehouse, we would be using the concept of B trees (a balanced binary tree). It will help in multi-level indexing as well as improvement in the performance of queries and ensure fast retrieval. We can take the example of product dimension table, then the b-tree indexing will look like below depending on the ProductCategoryKey and record pointer (RP).



2.4 Storage Plan:

Data for the Dominick Fine Foods has been categorized into different tables like CCOUNT, MOVEMENT, STORE and UPC. The data needs to be loaded into the data warehouse and it is in the CSV format. Data staging area will be used to perform this task. Here, the data cleansing also needs to be carried out after which the data can be moved to warehouse database [3]. It is very important to get an estimate about the size of the staging area for the data.

Table	**Number of Rows**	**Anticipated yearly increase**
Time Dimension Table	327046	35000
Store Dimension Table	100	N/A
Product Dimension Table	3347	500
Sales Fact Table	330000	36000
Product Fact Table	430000	36510

As per the above data, we currently do not need to partition the data as the storage size would suffice for storing current designed data marts. But as the size increases in near future, partitioning of the database will be needed. **Horizontal partitioning** will be based on the index keys specifically the Timekey.

2.5 Physical Model:

Table/Attribute Name	Data Type	Nulls Allowed	Comments
TIME DIMESNION TABLE			
TimeKey (Row Number)	Integer	No	Primary key of time dimension table
Date	Datetime	No	
Month	Char (30)	No	It will be derived from the date attribute
Year	Integer	No	It will be derived from the date attribute
STORE DIMENSION TABLE			
StoreKey (Row Number)	Integer	No	Primary Key of the store dimension table
StoreName	Varchar (50)	No	Name of the store of Dominick fine foods
City	Varchar (50)	No	City of the DFF store
Zip	Integer	No	Zip code of the DFF store
WorkWomen	Float	No	Percentage of the working women with full time jobs

PRODUCT CATEGORY DIMENSION TABLE			
ProductCategoryKey	Integer	No	Primary Key of the product category dimension table
Category Name	Varchar (50)	No	The category to which the product belongs to
COUPON CATEGORY DIMENSION TABLE			
Coupon Category Key	Integer	No	Primary Key of the coupon category dimension table
Category Name	Varchar (50)	No	The category to which the coupon belongs to
SALES FACT TABLE			
TimeKey	Integer	No	Partial primary key and foreign key referencing time dimension table
StoreKey	Integer	No	Partial primary key and foreign key referencing store dimension table
ProductCategoryKey	Integer	No	Partial primary key and foreign key referencing product category dimension table

Sales	Integer	No	Sales amount of the categories
PROMOTIONS FACT TABLE			
TimeKey	Integer	No	Partial primary key and foreign key referencing time dimension table
StoreKey	Integer	No	Partial primary key and foreign key referencing store dimension table
CouponCategoryKey	Integer	No	Partial primary key and foreign key referencing coupon category dimension table
CouponReedemed	Integer	No	Number of coupons of a redeemed

Data Sources:

Data Store	Source Files
CUSTOMER COUNT	CCOUNT.csv
MOVEMENT	WXXX.csv files
STORE	Used Demo.csv
UPC	UPCXXX.csv files

Data sources:

Data Sources	Source Files
CUSTOMER COUNT	CCOUNT.csv
MOVEMENT	WXXX.csv files
STORE	Used Demo.csv
WEEK	Used Week's Decode Table
UPC	UPCXXX.csv files

Target Tables:

Target Table Name	Columns	Data type
dimStore	StoreKey	varchar(50)
	StoreName	varchar(50)
	City	varchar(50)
	Zip	varchar(50)
	WorkWomen	varchar(50)

Target Table Name	Columns	Data type
dimTime	TimeKey	int
	Month	int
	Year	int
	Date	date

Target Table Name	Columns	Data type
dimProduct	ProductCategoryKey	int
	CategoryName	varchar(50)

Target Table Name	Columns	Data type
dimCoupon	CouponCategoryKey	int
	CategoryName	varchar(50)

Target Table Name	Columns	Data type
Promotions Fact Tables	CouponCategoryKey	int
	TimeKey	int
	StoreKey	varchar(50)
	CouponRedeemed	numeric(38,0)

Target Table Name	Columns	Data type
Sales Fact Tables	TimeKey	int
	StoreKey	varchar(50)
	ProductCategoryKey	int
	Sales	numeric(38,0)
	Sale_Promotions	varchar(50)

Data Mapping tables:

Data sources to data staging:

Source File Name	Source File Attributes	Staging Area Table Name	Staging Area Table Attributes
Demo.csv	NAME	STG_DEMO_1_EXTRACT	NAME
	CITY		CITY
	ZIP		ZIP
	WORKWOM		WORKWOM
	STORE		STORE
WBER.csv	STORE	STG_Beer_Movement_1_EXTRACT	STORE
	UPC		UPC
	WEEK		WEEK
	PROFIT		PROFIT
	SALE		SALE
	Category		Category
CCount.csv	STORE	STG_CCount_1_Extract	STORE
	DATE		DATE
	DAY		DAY
	PRODUCT_NAMES (34)		PRODUCT_NAMES (34)
	CATEGORY_COUPONS (18)		CATEGORY_COUPONS (18)
	WEEK		WEEK
			DERIVED_DATE
			MONTH_OF_RECORD
			YEAR_OF_RECORD
			DATE_FORMATTED

Staging area to data warehouse:

Source File Name	Source File Attributes	Dimension/Fact Table Name	Dimension/Fact Table Attributes	Mapping Function
STG_dim.Store	StoreKey	dim.Store	StoreKey	Surrogate Key
	StoreName		StoreName	
	City		City	
	Zip		Zip	
	WorkWomen		WorkWomen	
STG_dim.Time	TimeKey	dim.Time	TimeKey	Surrogate Key

	Month		Month	
	Year		Year	
	Date		Date	
STG_dim.CouponCategory	CouponCategoryKey	dim.CouponCategory	CouponCategoryKey	Surrogate Key
	CategoryName		CategoryName	
STG_dim.ProductCategory	ProductCategoryKey	dim.ProductCategory	ProductCategoryKey	Surrogate Key
	CategoryName		CategoryName	
STG_Fact_Sales_Table	TimeKey	Fact_Sales_Table	TimeKey	Surrogate Key
	StoreKey		SoreKey	
	ProductCategoryKey		ProductCategoryKey	
	Sales		Sales	
	SalesPromotions		SalesPromotions	
STG_Fact_Promotiomns_Table	CouponCategoryKey	Fact_Promotiomns_Table	CouponCategoryKey	Surrogate Key
	TimeKey		TimeKey	
	StoreKey		StoreKey	
	CouponReedeemed		CouponRedeemed	

Data extraction rules:

Data extraction is the first step in the implementation of data warehouse for the Dominick Fine Food (DFF). Data extraction was performed using the csv files that were provided to us through the Dominick Dataset. For cleaning purposes, these files were first loaded into the staging area. A pivotal task in establishing the data warehouse was the efficient extraction of data from these archives. Extraction process was carried out with due diligence to ensure that there are no aberrations in it.

Data extraction plan was charted out keeping the balance of efficiency and data integrity in mind. We extracted only the relevant columns from each csv file which helped to reduce table size in the staging area.

Dominick Fine Food (DFF) data manual was referred to understand and gain insight about the data. The data from various csv files were extracted into the tables of the data staging area named 601_Group11_STG. We used CCount, Demo and Movement files to extract the data and create the corresponding tables in the database in the Staging area named 601_Group11_STG.

Data transformation and cleansing rules:

Following the data extraction into the staging area, we must perform the process of transformation and cleansing of data. This transforms the extracted data into the clean tables that will be useful in creating the dimension and fact tables. The regulations regulating data guarantees the data is reliable and error free. To ensure efficient query execution, the cleaned data should be loaded into the data staging area.

1) Removal of null values – Some of the important columns had null values and need to be removed so that all the data is consistent. The values for example, in stores cannot be null and thus these records need to be removed.

2) Removal of negative value - Similar to the null values, there are certain values containing negative values that need to be positive only. These values like that of week, can never be accepted in the negative scenario. Thus, these types of records need to be removed. It is critical to remove these types of irrelevant values so that they do not impact the results that will be gained through reporting from the data warehouse.

3) Removal Special Character- There are some special characters in the data that do not make any sense and their elimination will help clean the data. For example, the varchar columns in the Demo table have quotes and their removal will be according to the standard format. The data processing and its efficiency will improve the overall process and keep it in standard format.

4) Data Conversion - There are some values that need to be in the fact table and should be additive in nature. Thus, the data type of these values should be numeric and hence need to be converted into the appropriate data type. This data conversion, for example, is done on the sales of products in the CCount table and also the coupon categories for products.

5) Removing Unwanted Data - There are certain attributes that need not be present to answer our specific business questions and have been removed.

6) Creating fields using formula - Some of the attributes such as month and year need to be extracted from the date column to be in correspondence with the granularity of the tables we have decided to answer our business questions. Thus, these fields have been created using the formula month(date) and year(date).

7) Creation of Surrogate keys - All the dimension tables need to have surrogate keys to map them to the fact tables. These surrogate keys also act as unique identifiers.
The Surrogate Keys are as follows:

▪ TimeKey: Unique identifier for the time dimension table i.e. dim.Time

▪ StoreKey: Unique identifier for the store dimension table i.e. dim.Store

▪ ProductCategoryKey: Unique identifier for the product category dimension table i.e. dim.ProductCategory

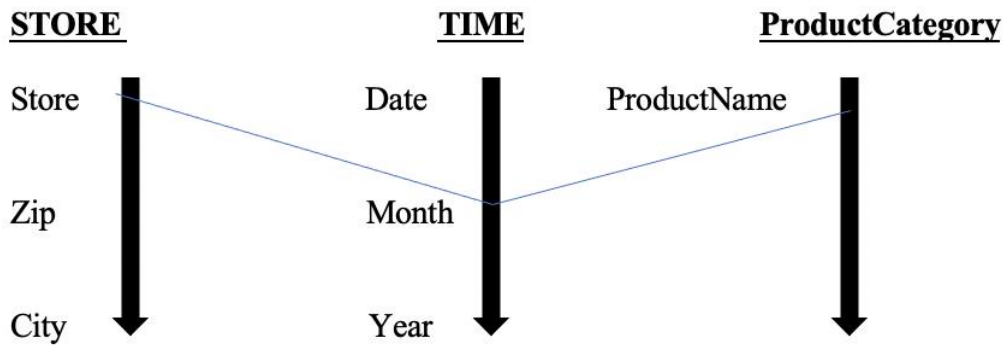
▪ CouponCategoryKey: Unique identifier for the coupon category dimension table i.e. dim.CouponCategory

Plan for aggregate tables:

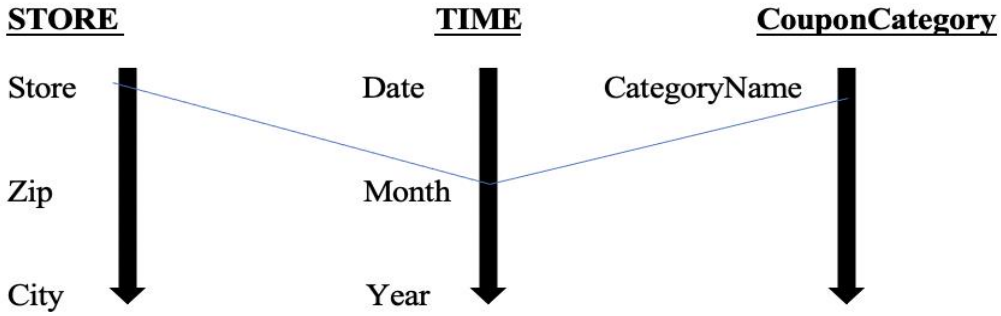
Based on the business requirements, many times there is a need for the summarization of data. After giving due diligence to the performance and requirements we decided to come up with the aggregate plan for two cases:

Target Table Name	Columns	Data type
dim.Time.Aggregated	MonthKey	int
	Month	int
	Year	int
Fact_Aggregated_Sales_Table	MonthKey	int
	StoreKey	varchar(50)
	ProductCategoryKey	int
	Sales	numeric(38,0)
	Sales_Promotions	bigint
Fact_Aggregated_Promotions_Table`	CouponCategoryKey	int
	MonthKey	int
	StoreKey	varchar(50)
	CouponRedeemed	numeric(38)

i. For the Sales fact table, we opted for one-way aggregation in which one fact table row will represent sales per month for each store and for each product



ii. For the Promotions fact table, we opted for one-way aggregation in which one fact table row will represent sales of coupon per month for each store and for each coupon



SSIS Functions:

The SSIS functions used during transformation and cleansing is as follows:

1) DATA CONVERSION: This function helps translate data to the desired data type, allowing us to ensure accuracy of the data type in all tables.

2) AGGREGATE: Aggregation function was used to aggregate the data based on the business need.

3) UNPIVOT: Denormalized data was made more feasible by converting it into a normalized version. We have used this function for creating Product and Coupon Category staging tables.

4) MERGE JOIN: A Merge join was used to join the datasets which were sorted.

5) LOOKUP: In order to join the additional columns in the dataflow, the lookup function was used by referencing the values in the dimension table.

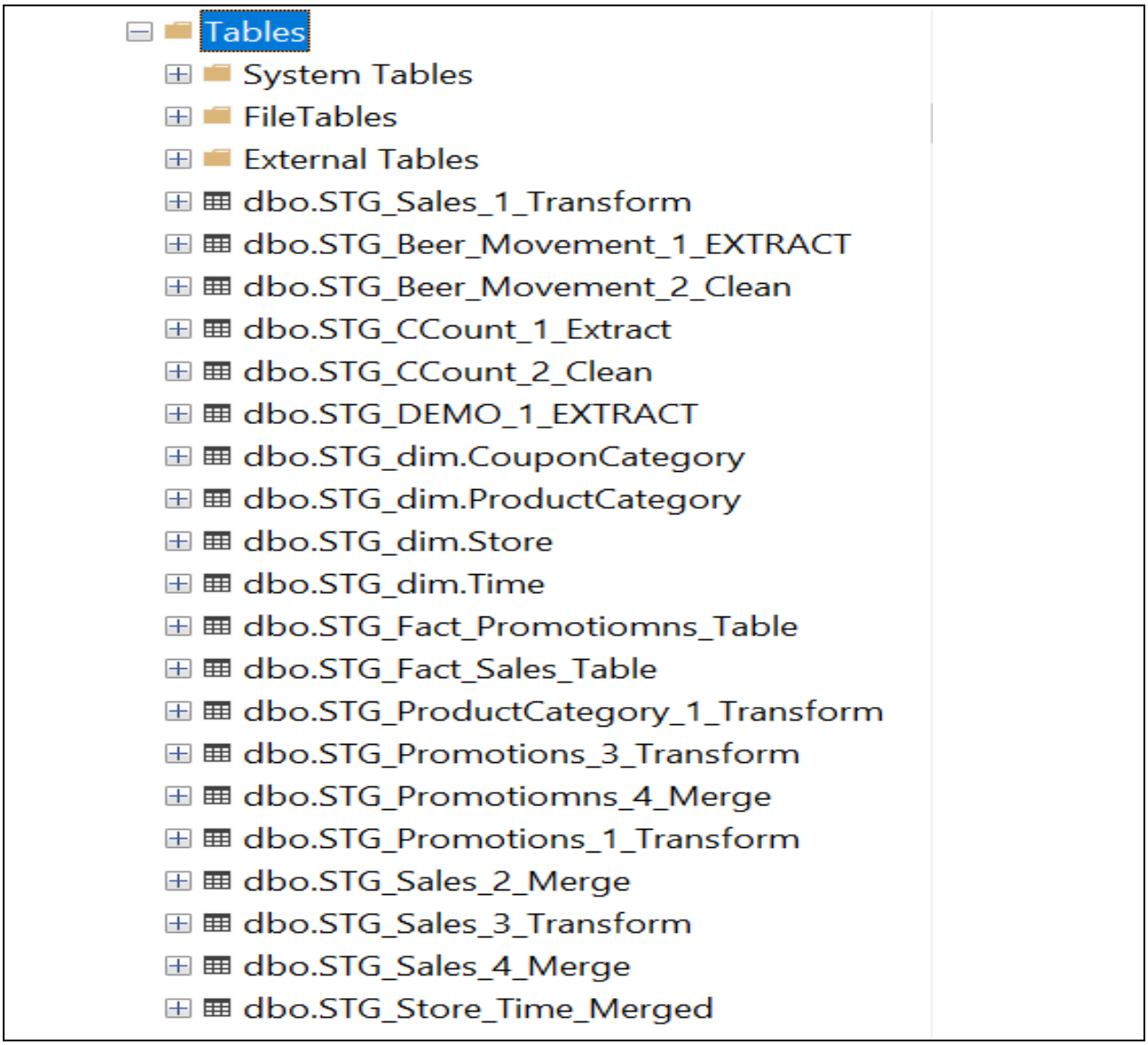
Organization of data staging area:

Based on the naming standards, tables have been created at every logical step in the staging database and are given below in the table:

Staging Area Table Name	Staging Area Table Attributes
STG_DEMO_1_EXTRACT	NAME
	CITY
	ZIP
	WORKWOM
	STORE
STG_Beer_Movement_1_EXTRACT	STORE
	UPC
	WEEK
	PROFIT
	SALE
	Category
STG_CCount_1_Extract	STORE
	DATE
	DAY
	PRODUCT_NAMES (34)
	CATEGORY_COUPONS (18)
	WEEK
	DERIVED_DATE
	MONTH_OF_RECORD

	YEAR_OF_RECORD
	DATE_FORMATTED
STG_Sales_1_Transform	STORE
	PRODUCT_NAMES (34)
	MONTH_OF_RECORD
	YEAR_OF_RECORD
	DATE_FORMATTED
	WEEK
STG_ProductCategory_1_Transform	CATEGORYNAME
STG_Promotions_1_Transform	STORE
	MONTH_OF_RECORD
	YEAR_OF_RECORD
	DATE_FORMATTED
	CATEGORY_COUPONS (18)
STG_Promotions_3_Transform	CATEGORYNAME
	COUPONREDEEMED
	STORE
	MONTH_OF_RECORD
	YEAR_OF_RECORD
	DATE_FORMATTED
STG_Sales_3_Transform	PRODUCTCATEGORYNAME
	STORE
	DATE_FORMATTED
	SALE_PROMOTIONS
	SALES
STG_Beer_Movement_2_Clean	STORE
	WEEK
	SALE
STG_CCount_2_Clean	STORE
	DATE
	DERIVED_DATE
	DAY
	PRODUCT_NAMES (34)
	CATEGORY_COUPONS (18)
	WEEK
	DERIVED_DATE
	MONTH_OF_RECORD
	YEAR_OF_RECORD
	DATE_FORMATTED
STG_Sales_2_Merge	STORE
	PRODUCT_NAMES (34)

	MONTH_OF_RECORD
	YEAR_OF_RECORD
	DATE_FORMATTED
	SALE_PROMOTIONS
STG_Sales_4_Merge	PRODUCTCATEGORYNAME
	TIMEKEY
	STORE
	SALE_PROMOTIONS
	SALES
STG_Promotions_4_Merge	CATEGORYNAME
	COUPONREDEEMED
	STORE
	TIMEKEY
STG_Store_Time_Merged	STOREKEY
	TIMEKEY
	MONTH
	YEAR
	DATE



Staging Database

Procedures for extraction and loading:

Data Extraction Procedures:

The following tables are created in the data staging area

a) CCount

Source: Ccount.csv

Destination: STG_CCount_1_Extract, STG_CCount_2_Clean

i. The SSIS package is used to extract the CCount.csv to STG_CCount_1_Extract
ii. The data was cleaned through transformation package and stored in STG_CCount_2_Clean
iii. The columns that were not required were removed
iv. columns MONTH_OF_RECORD, YEAR_OF_RECORD, and DATE_FORMATTED were created
v. Some of the columns were unpivoted to form ProductCategory and CouponCategory specific tables

b) Movement

Source: DONE-WBER.csv

Destination: STG_Beer_Movement_1_EXTRACT, STG_Beer_Movement_2_Clean

i. The movement file for beer was extracted
ii. The columns that were not needed were removed and further the table was cleaned using SSIS packages and the queries in Execution task to a clean table STG_Beer_Movement_2_Clean
iii. The sale column had promotions which were mapped as B - 0, S - 1, and C - 2 to help in analysis in the future

c) Demographics

Source: Demo.csv

Destination: STG_DEMO_1_EXTRACT

i. The Demo.csv is extracted to STG_DEMO_1_EXTRACT with the required columns
ii. The unnecessary columns are dropped

Data Loading Procedures:

The following tables are created in the warehouse area

a) dim.Store

Source: STG_dim.Store
Destination: dim.Store

i. dim.Store is created in the warehouse using the STG_dim.Store table which was created in the Staging area

b) dim.Time

Source: STG_dim.Time
Destination: dim.Time

i. dim.Time is created in the warehouse using the STG_dim.Time table which was created in the Staging area

c) dim.Time.Aggregated

Source: STG_dim.Time
Destination: dim.Time.Aggregated

i. dim.Time is created in the warehouse using the STG_dim.Time table which was created in the Staging area
ii. The aggregation was done over month column of the STG_dim.Time table

d) dim.ProductCategory

Source: STG_dim.ProductCategory
Destination: dim.ProductCategory

i. dim.ProductCategory is created in the warehouse using the STG_dim.ProductCategory table which was created in the Staging area

e) dim.CouponCategory

Source: STG_dim.CouponCategory
Destination: dim.CouponCategory

i. dim.CouponCategory is created in the warehouse using the STG_dim.CouponCategory table which was created in the Staging area

f) Fact_Promotions_Table

Source: STG_Fact_Promotions_Table
Destination: Fact_Promotions_Table

i. CouponCategoryKey, TimeKey and StoreKey are the Surrogate keys which are derived from CouponCategory dimension, Time dimension and Demo table.

g) Fact_Aggregated_Promotions

Source: STG_Fact_Promotions_Table
Destination: Fact_Aggregated_Promotions

i. CouponCategoryKey, TimeKey and StoreKey are the Surrogate keys which are derived from CouponCategory dimension, Time dimension and Demo table.
ii. The aggregation was done over month column of the STG_Fact_Promotions_Table

h) Fact_Sales_Table

Source: STG_Fact_Sales_Table
Destination: Fact_Sales_Table

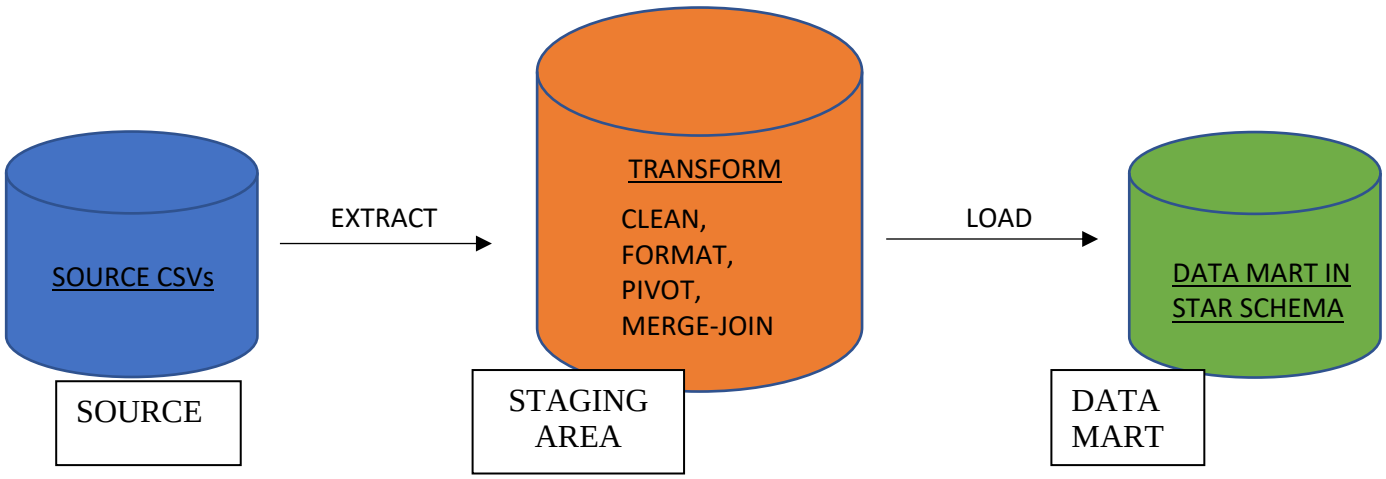
i. ProductCategoryKey, TimeKey and StoreKey are the Surrogate keys which are derived from ProductCategory dimension, Time dimension and Demo table.

i) Fact_Aggregated_Sales

Source: STG_Fact_Sales_Table
Destination: Fact_Aggregated_Sales

i. ProductCategoryKey, TimeKey and StoreKey are the Surrogate keys which are derived from ProductCategory dimension, Time dimension and Demo table.
ii. The aggregation was done over month column of the STG_Fact_Sales_Table

ETL Plan:

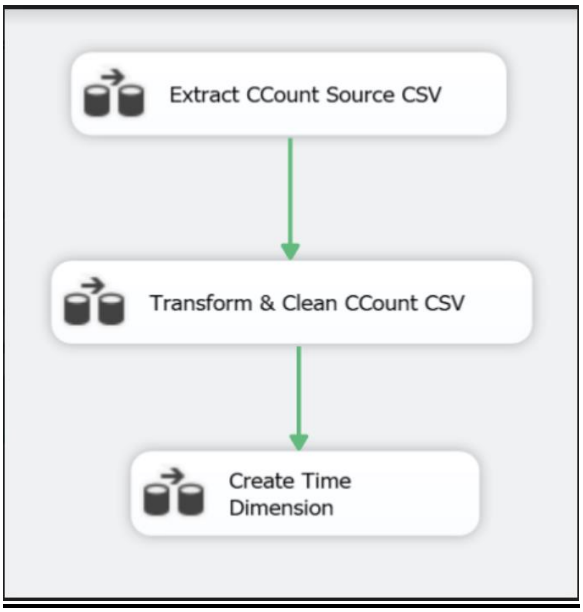


ETL for dimension tables:

This section describes the ETL plan for dimension tables with the flow diagrams. Their implementations are shown in detail with the package execution in the ETL implementation part.

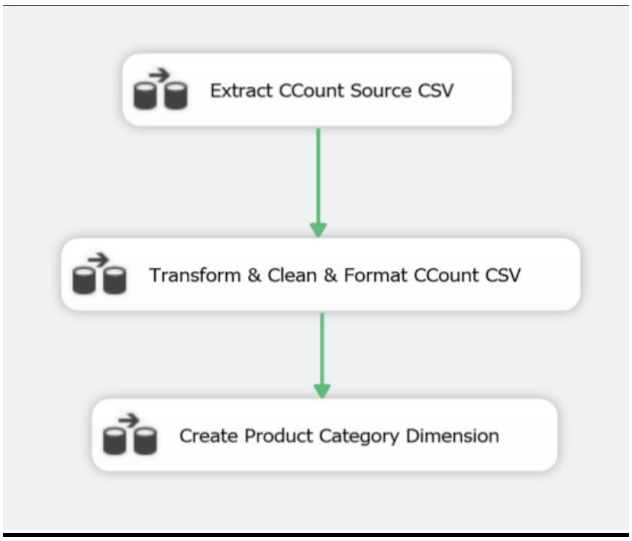
Time Dimension:

The time dimension contains TimeKey, Date, Month, and Year which were extracted through the transformation process. The date was formatted in the right way and some functions such as Month() and Year() were performed to finally get the time dimension.



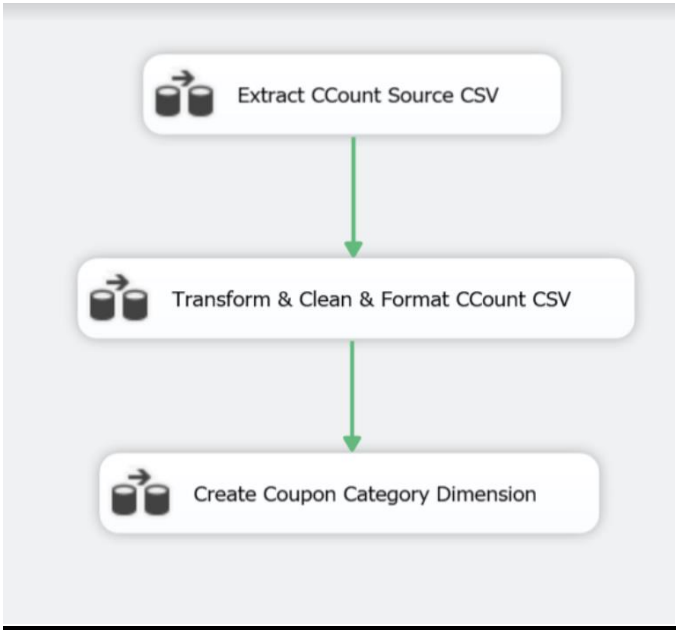
ProductCategory Dimension:

The ProductCategory dimension contains the columns ProductCategoryKey and CategoryName. The surrogate key ProductCategoryKey was created and the CategoryName was extracted from the column names of the CCount and unpivoted to get the final dimension table.



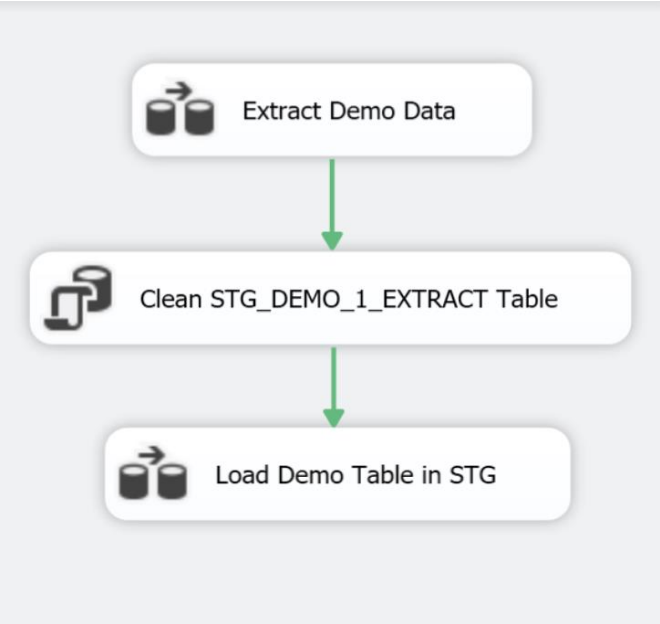
CouponCategory Dimension:

The CouponCategory dimension contains the columns CouponCategoryKey and CategoryName. The surrogate key CouponCategoryKey was created and the CategoryName was extracted from the column names of the CCount and unpivoted to get the final dimension table.



Store Dimension:

The store dimension contains StoreKey, StoreName, City, Zip, and WorkWomen. The data is collected from the demo table and cleaned to have only the columns required before creating the final dimension table along with the surrogate key StoreKey.

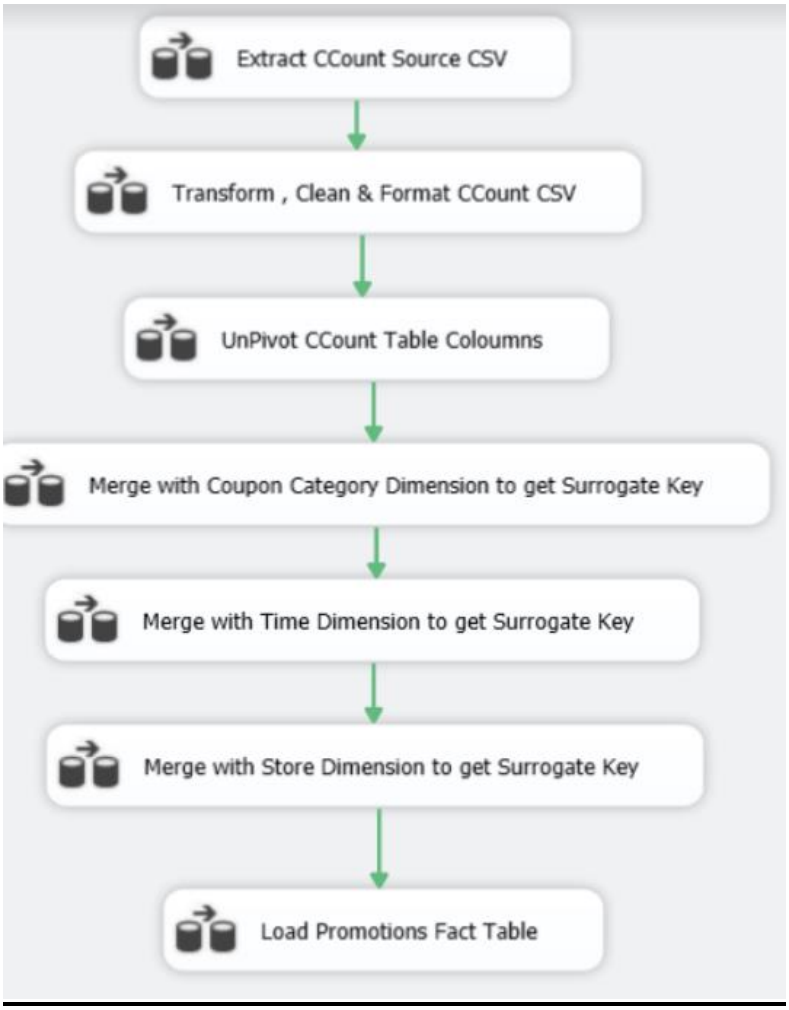


ETL for fact tables:

This section describes the ETL plan for fact tables with the flow diagrams. Their implementations are shown in detail with the package execution in the ETL implementation part.

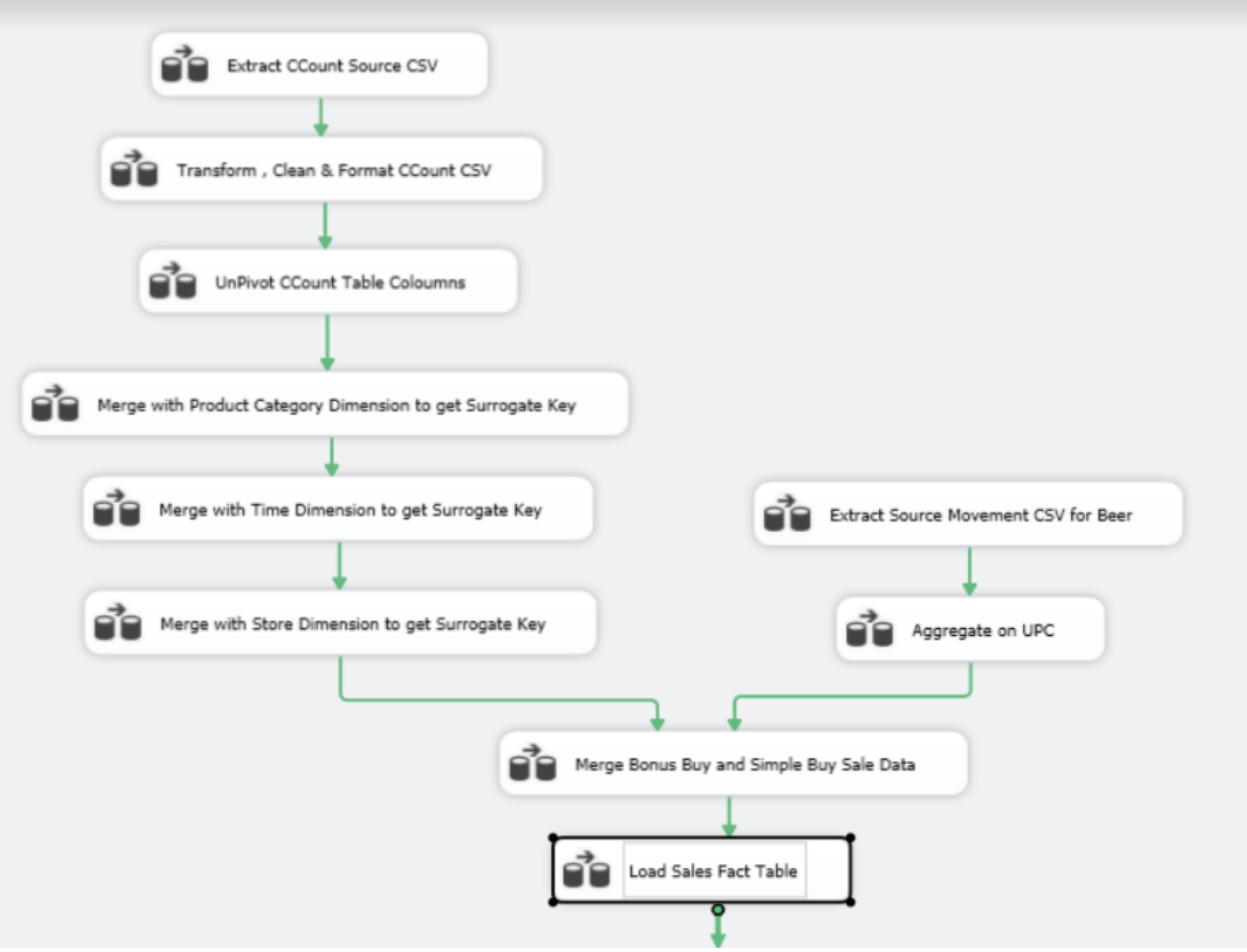
Promotions Fact Table:

The promotions fact table has columns TimeKey, StoreKey, CouponCategoryKey as foreign keys and the CouponRedeemed. The cleaned tables are used to extract data and merge with dimension tables to get the right data in correspondence to the surrogate keys.



Sales Fact Table:

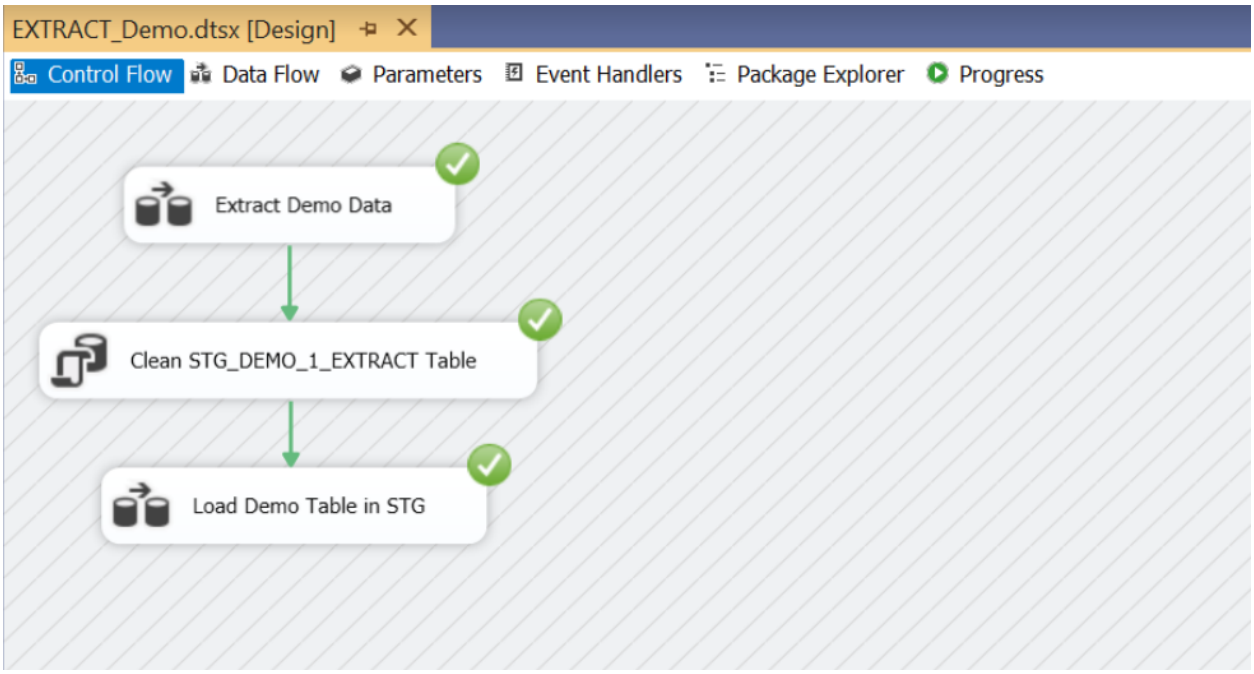
The promotions fact table has columns TimeKey, StoreKey, ProductCategoryKey as foreign keys and the Sales and SalePromotions. The cleaned tables are used to extract data and merge with dimension tables to get the right data in correspondence to the surrogate keys.



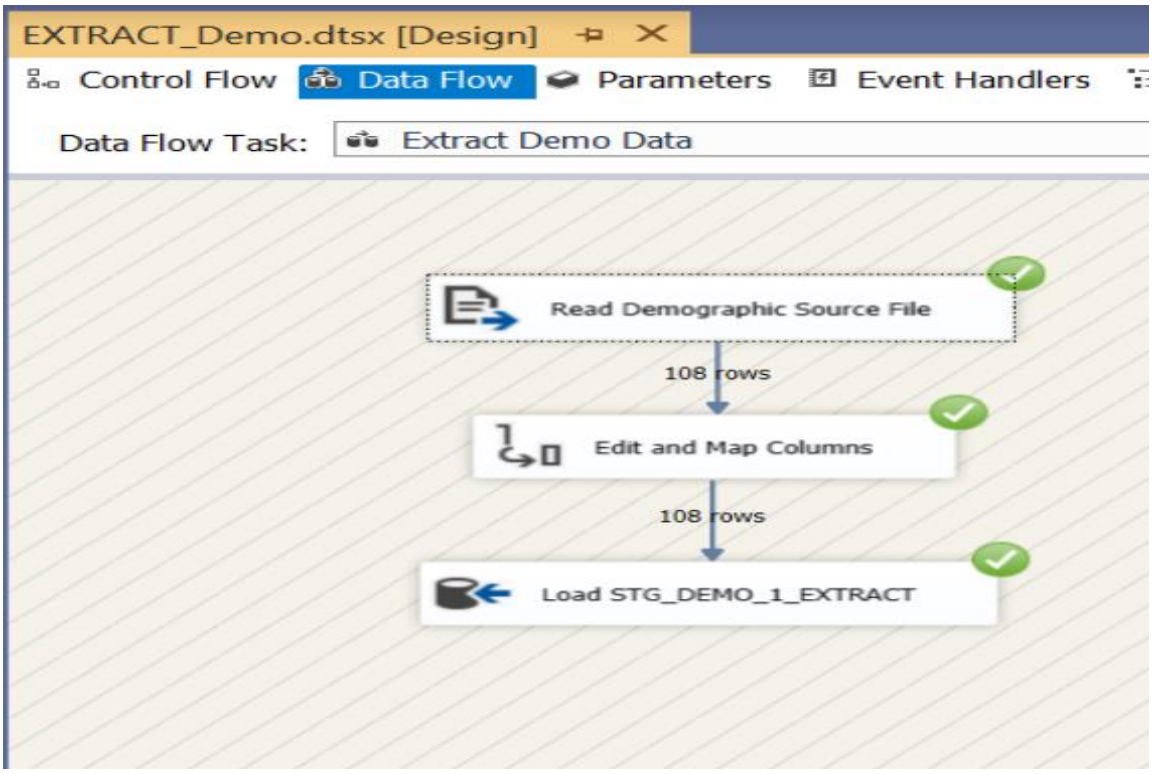
ETL Implementations:

The ETL implementation is divided into the steps of extraction of source data, cleaning and transformation of the tables, and their loading into the data marts.

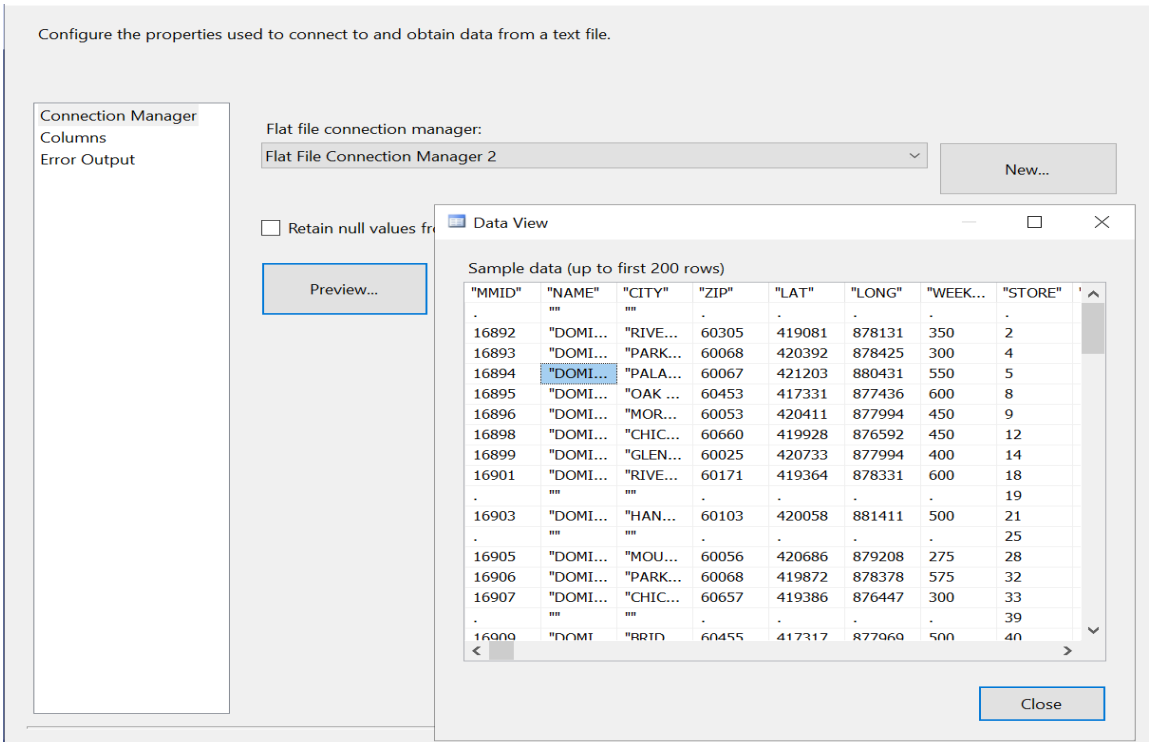
ETL for Store Dimension:

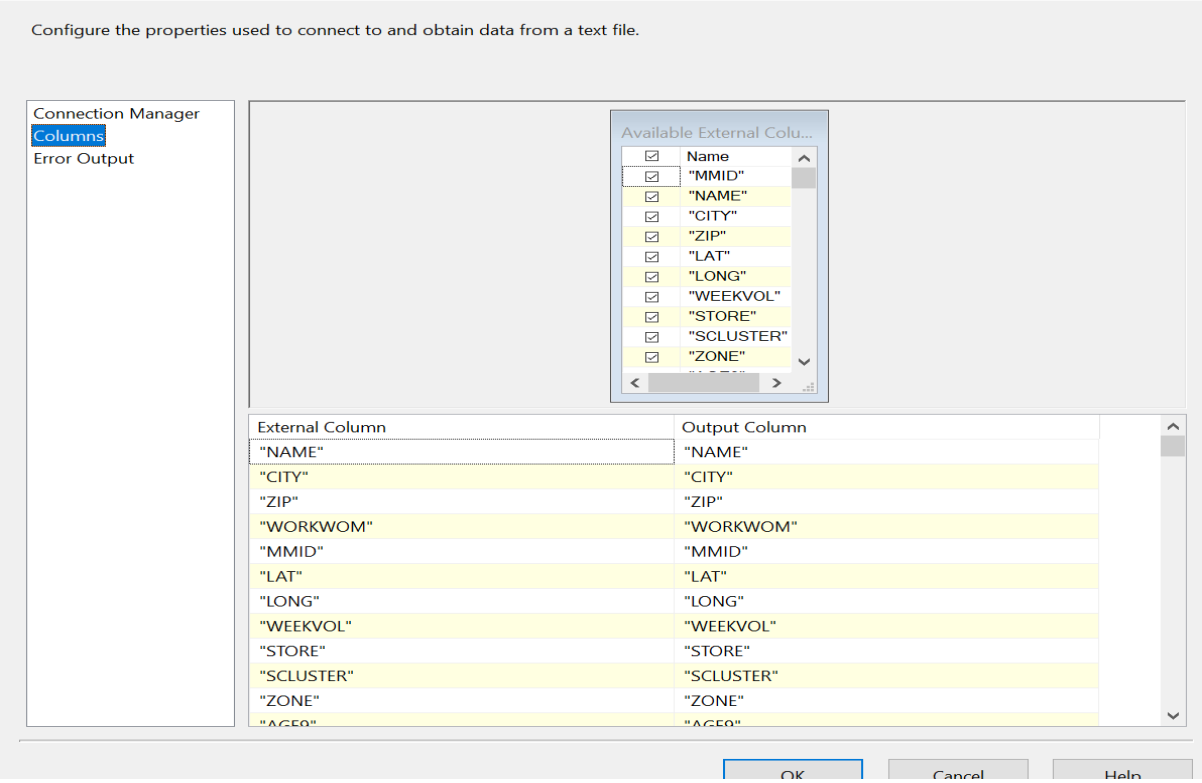


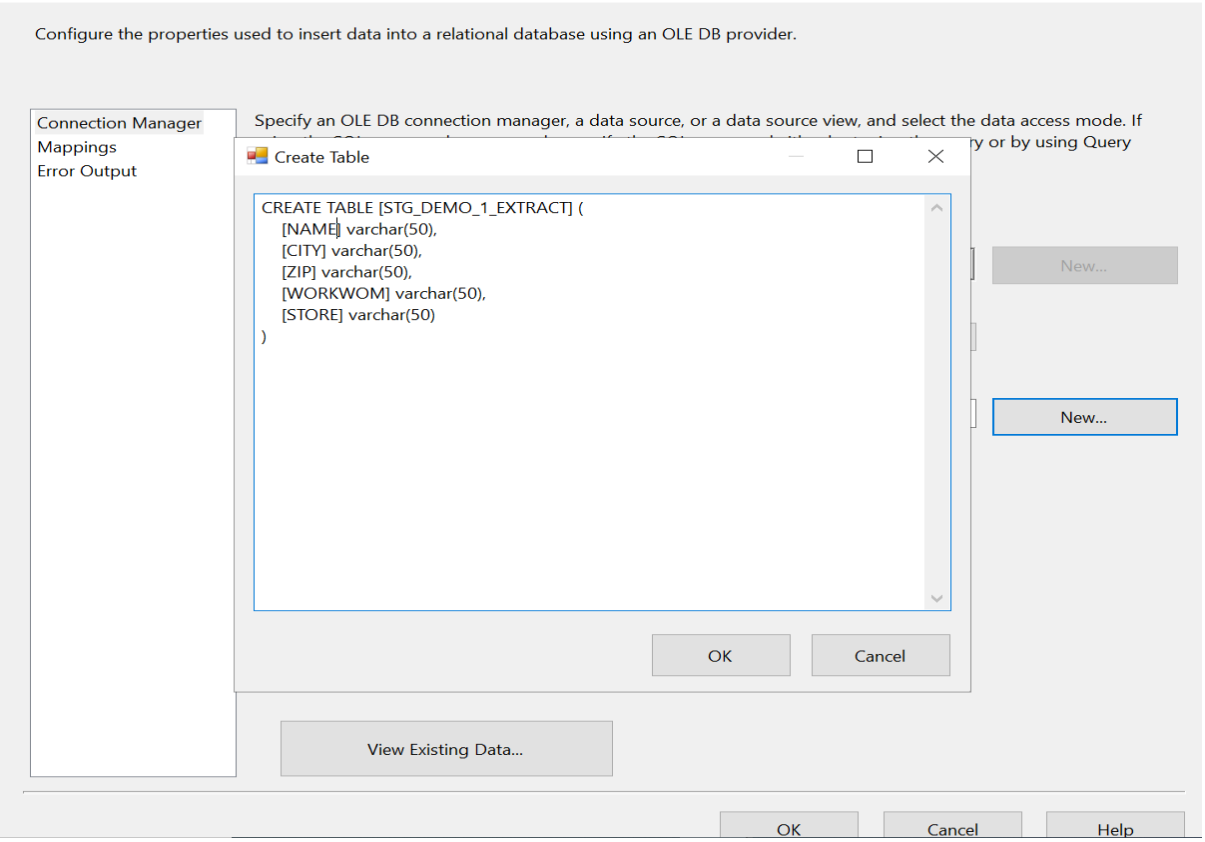
Extraction of Demo data to Demo table

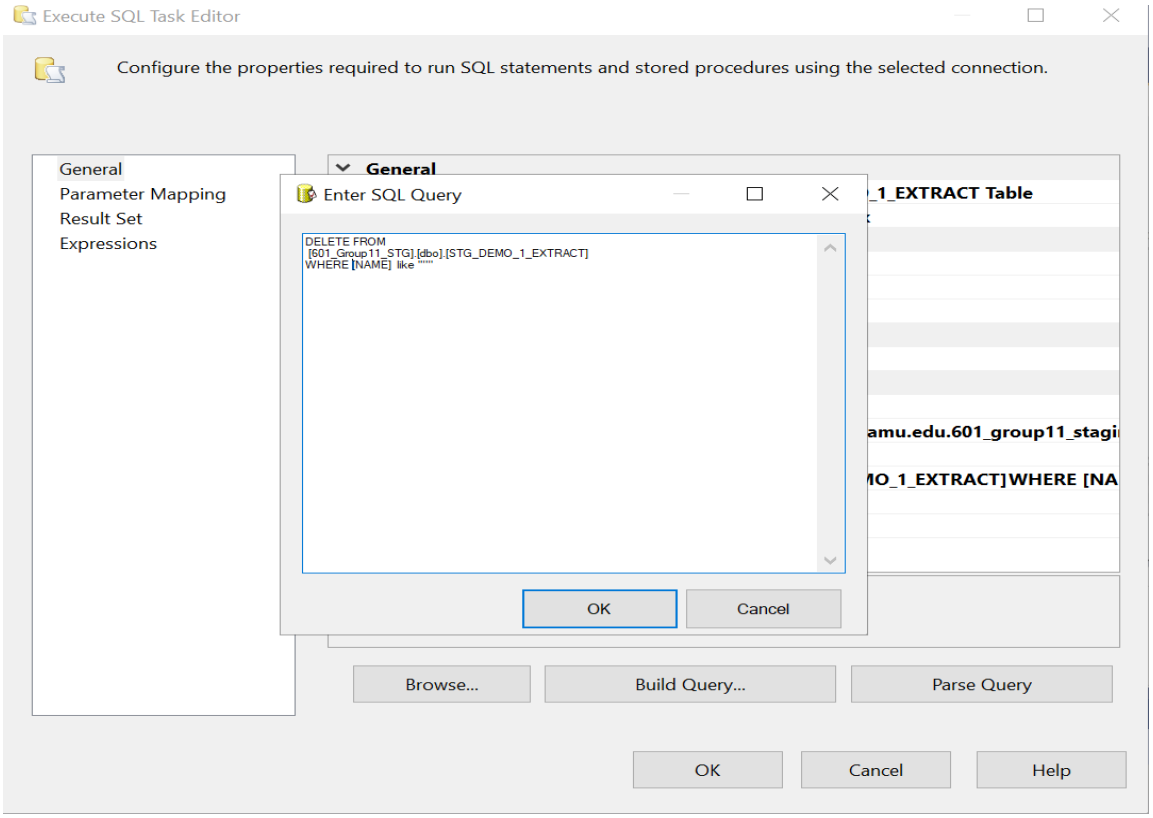


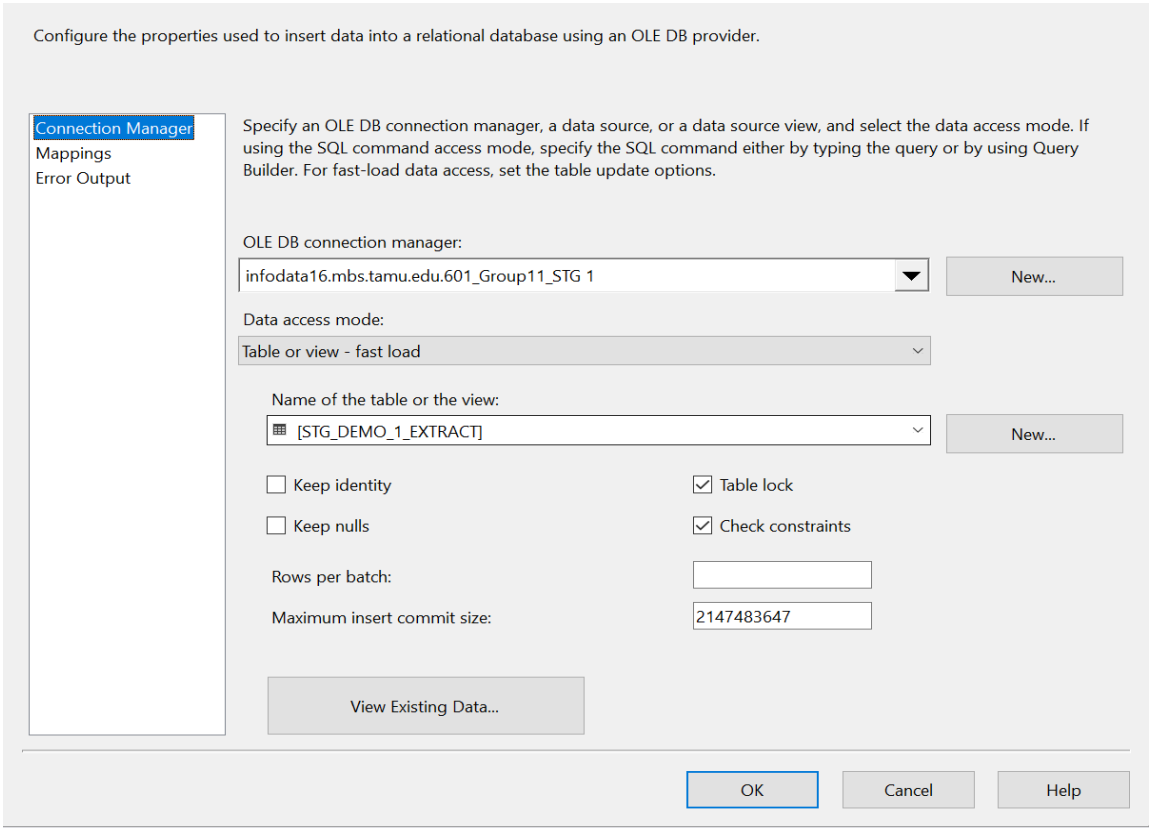
Data Flow for Extract Demo Data

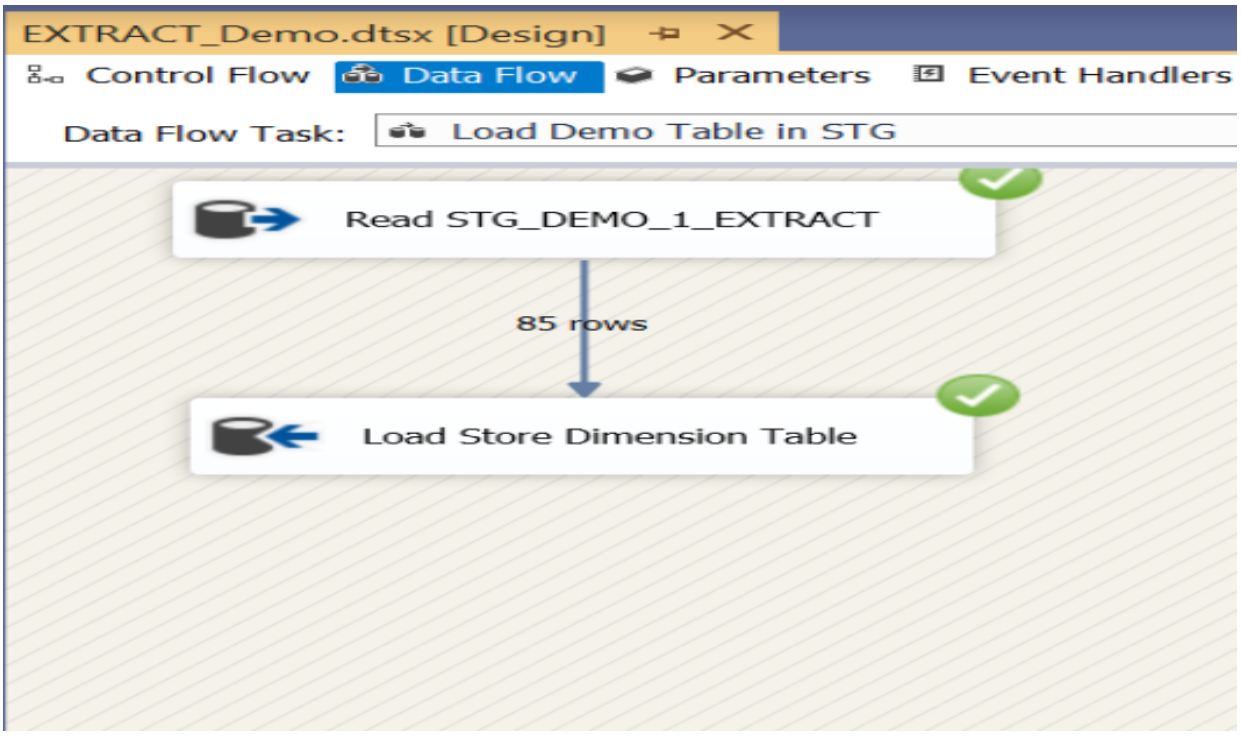



Configure the properties used to connect to and obtain data from a text file.
Connection Manager
Columns
Error Output
Available External Colu...
Name
"MMID"
"NAME"
"CITY"
"ZIP"
"LAT"
"LONG"
"WEEKVOL"
"STORE"
"SCLUSTER"
"ZONE"
External Column
Output Column
"NAME"
"CITY"
"ZIP"
"WORKWOM"
"MMID"
"LAT"
"LONG"
"WEEKVOL"
"STORE"
"SCLUSTER"
"ZONE"
OK
Cancel
Help

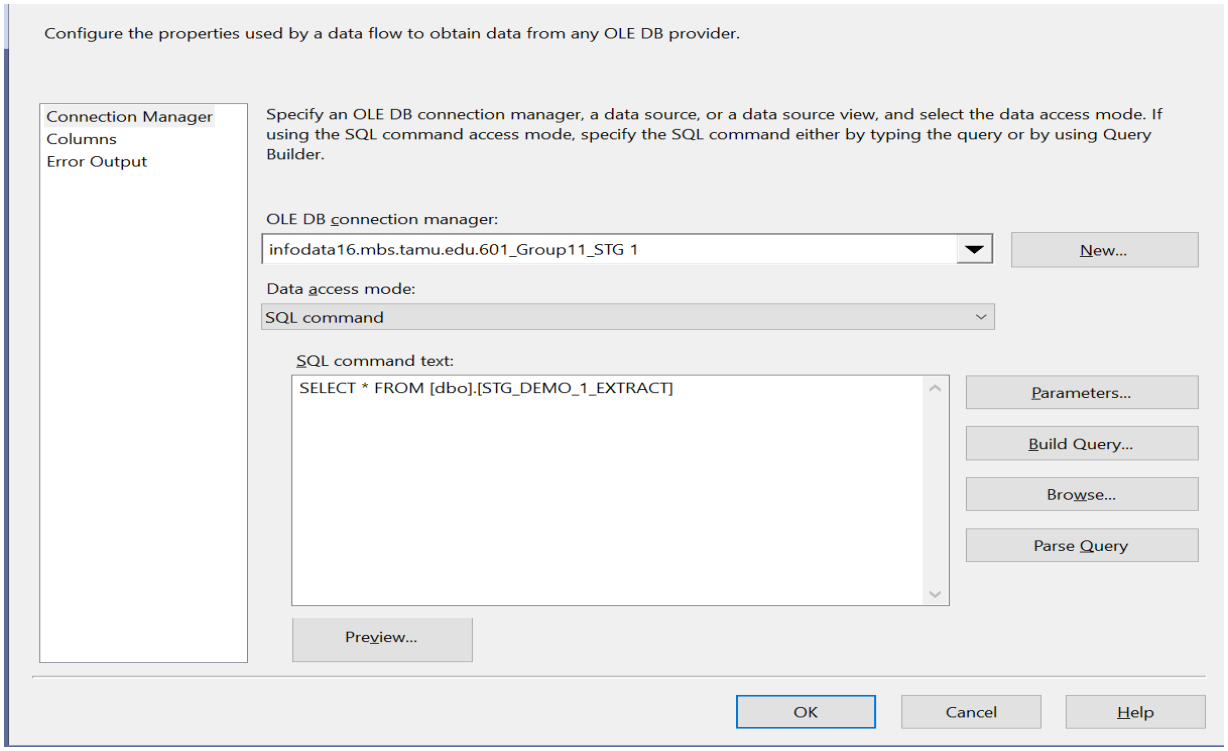

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
or by using Query
Create Table
CREATE TABLE [STG_DEMO_1_EXTRACT] (
[NAME] varchar(50),
[CITY] varchar(50),
[ZIP] varchar(50),
[WORKWOM] varchar(50),
[STORE] varchar(50)
)
OK
Cancel
New...
New...
View Existing Data...
OK
Cancel
Help

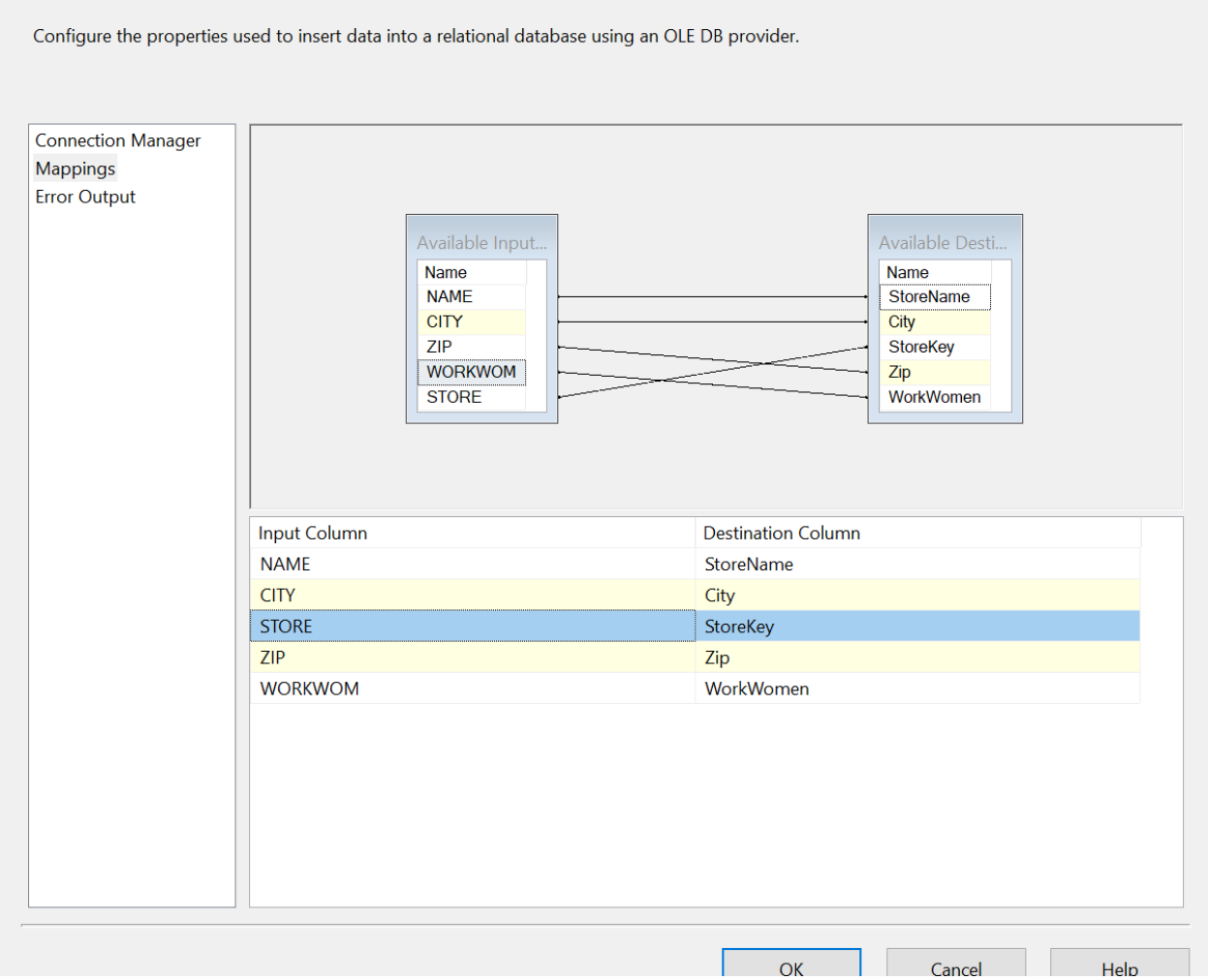

Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
General
Enter SQL Query
DELETE FROM
[601_Group11_STG].[dbo].[STG_DEMO_1_EXTRACT]
WHERE [NAME] like '%'
OK
Cancel
Browse...
Build Query...
Parse Query
OK
Cancel
Help

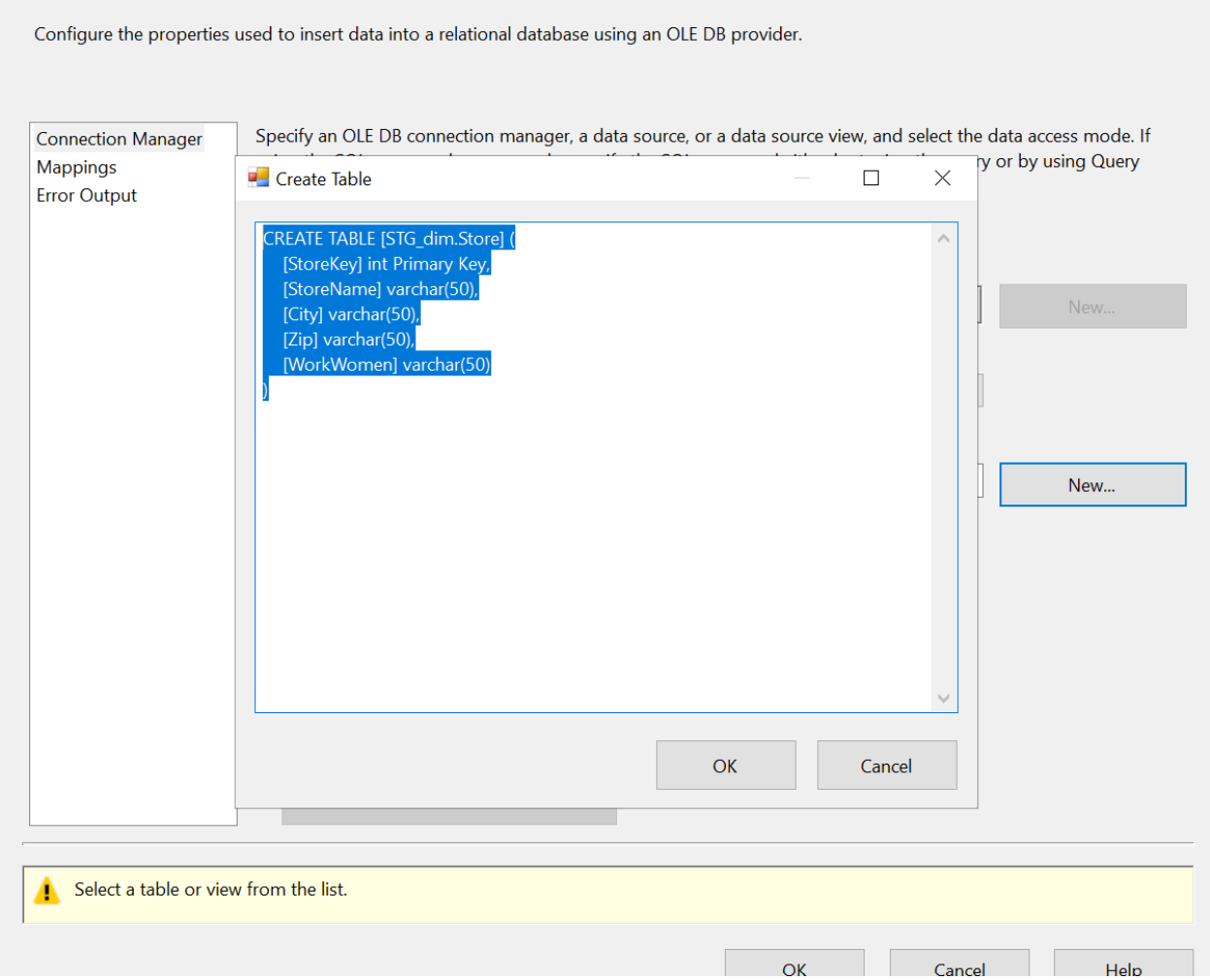

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG 1
New...
Data access mode:
Table or view - fast load
Name of the table or the view:
[STG_DEMO_1_EXTRACT]
New...
Keep identity
Table lock
Keep nulls
Check constraints
Rows per batch:
Maximum insert commit size:
2147483647
View Existing Data...
OK
Cancel
Help



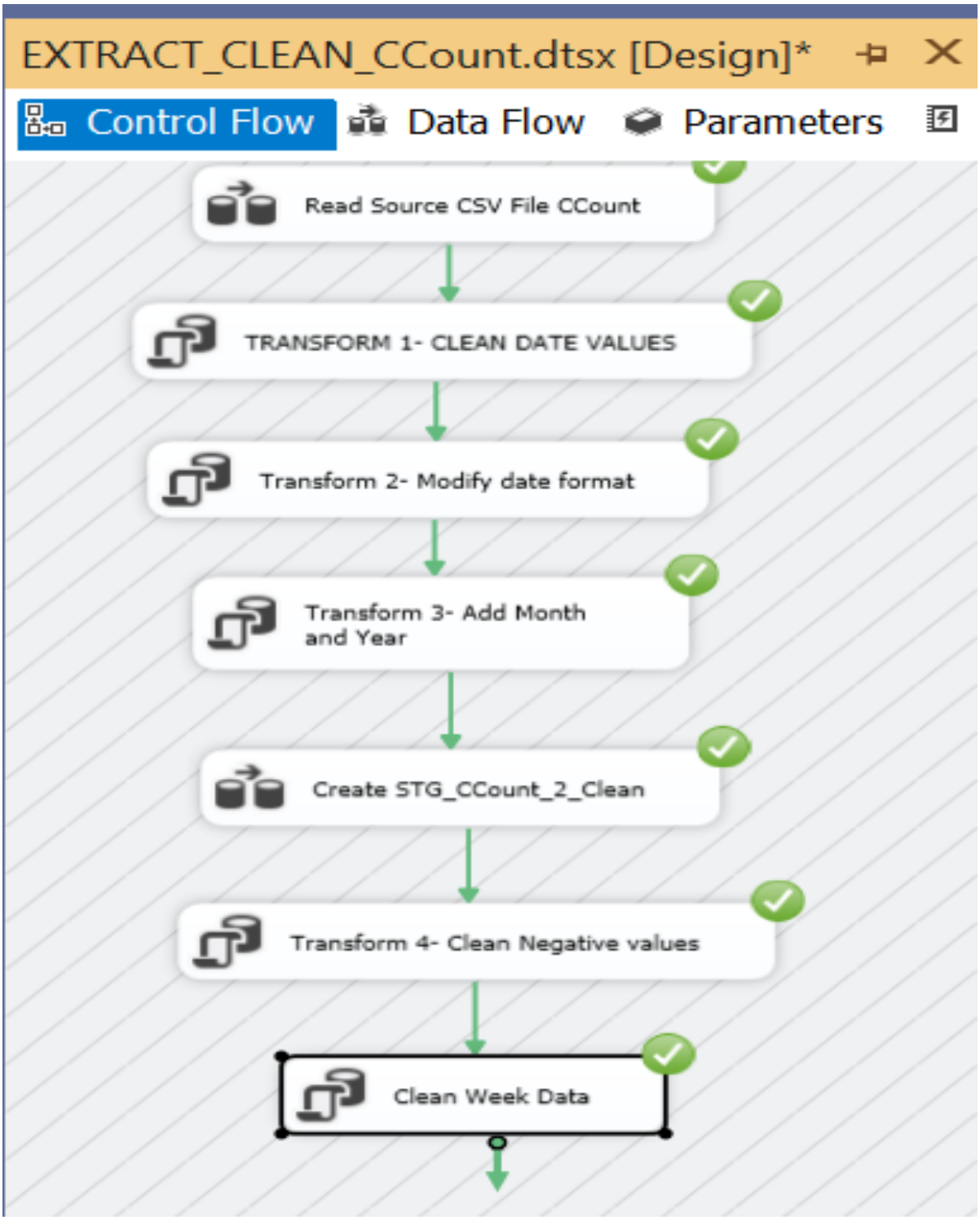
Data Flow for loading Store dimension table



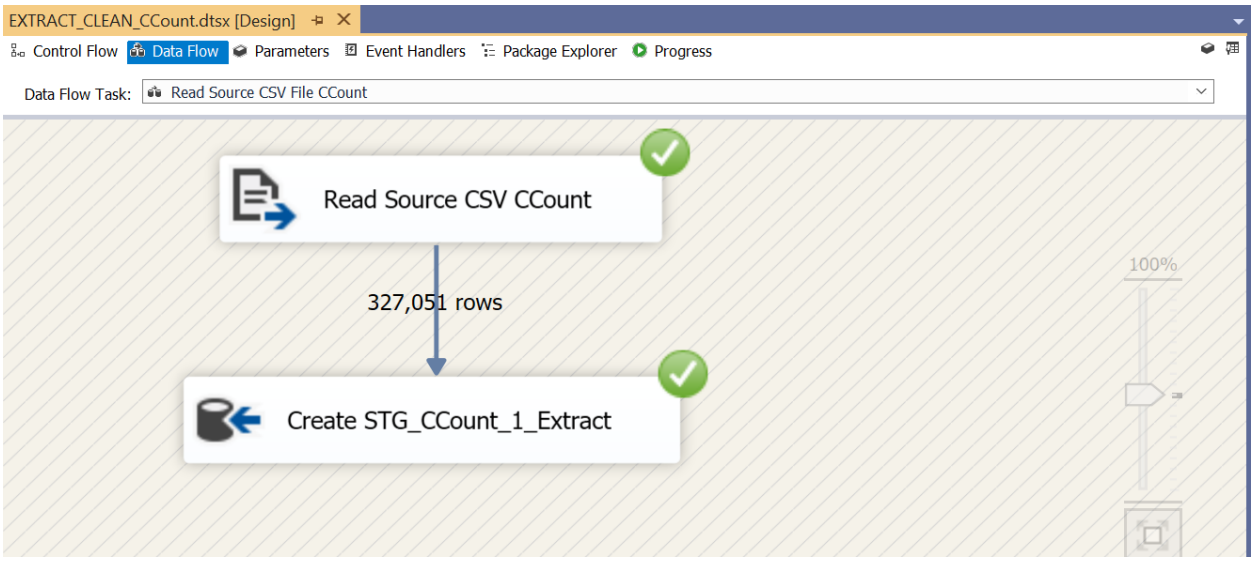

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Available Input...
Name
NAME
CITY
ZIP
WORKWOM
STORE
Available Desti...
Name
StoreName
City
StoreKey
Zip
WorkWomen
Input Column
Destination Column
NAME
StoreName
CITY
City
STORE
StoreKey
ZIP
Zip
WORKWOM
WorkWomen
OK
Cancel
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
y or by using Query
Create Table
CREATE TABLE [STG_dim.Store] (
[StoreKey] int Primary Key,
[StoreName] varchar(50),
[City] varchar(50),
[Zip] varchar(50),
[WorkWomen] varchar(50)
)
OK
Cancel
New...
New...
Select a table or view from the list.
OK
Cancel
Help

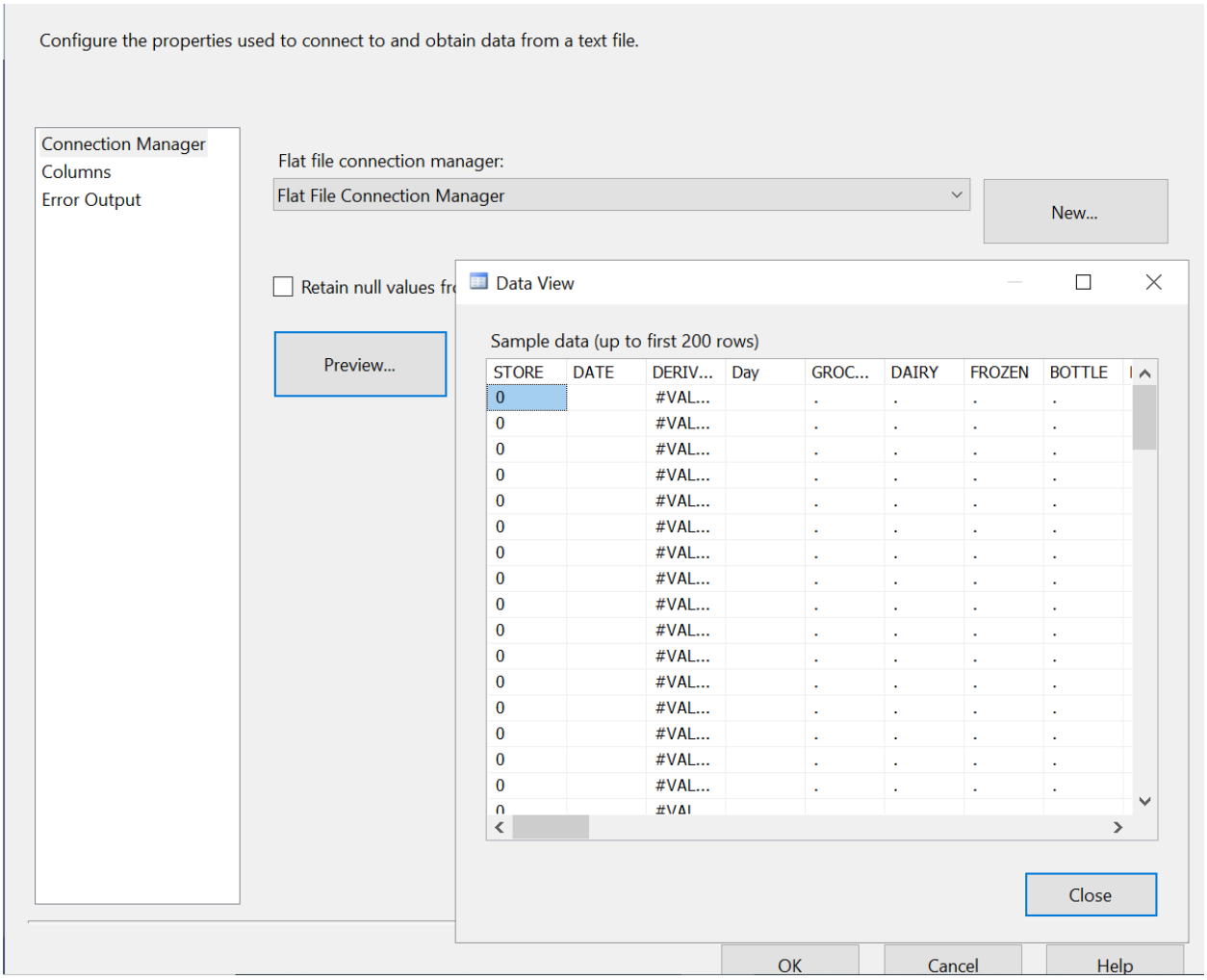
Extraction and Transformation of CCount:

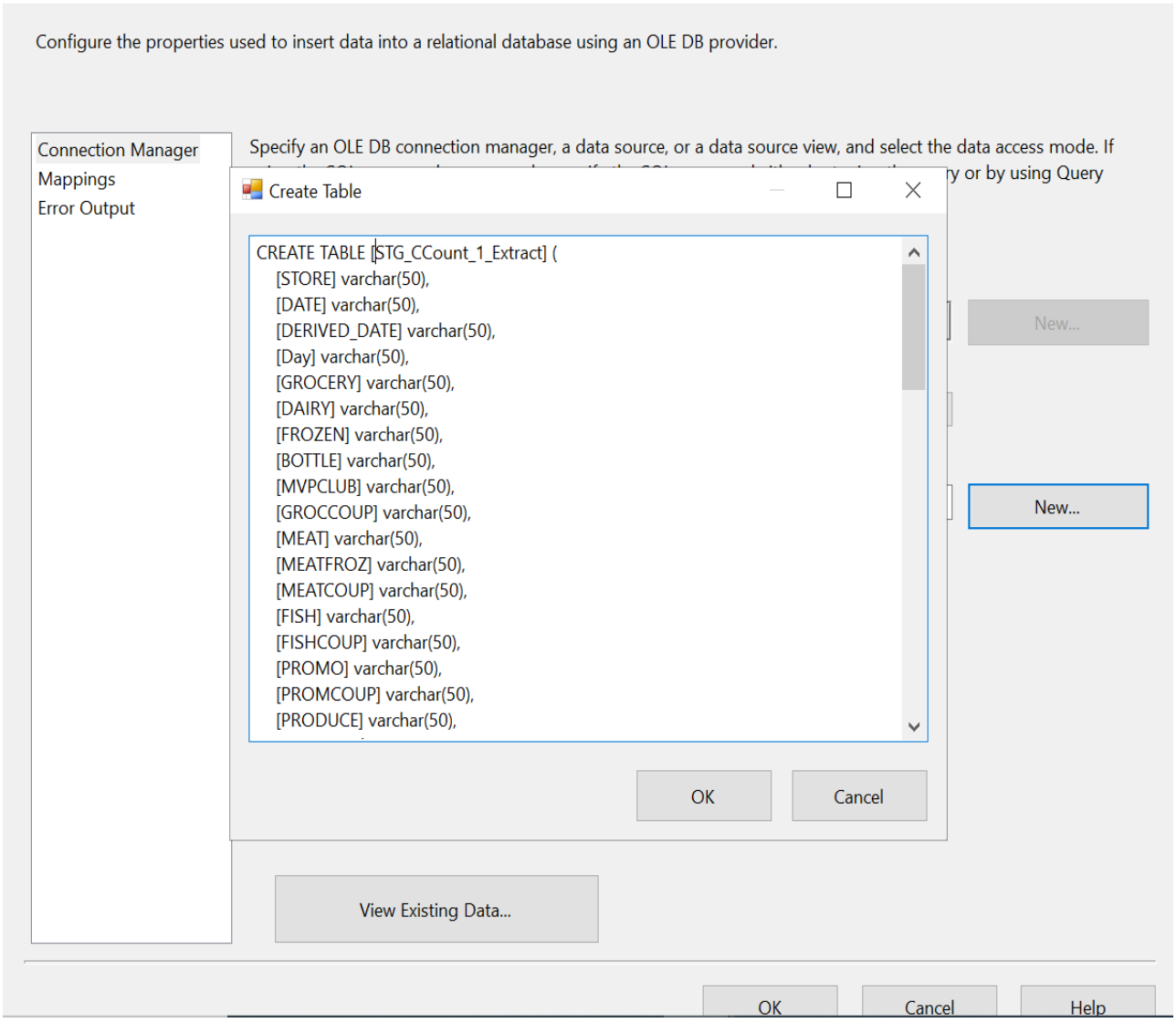


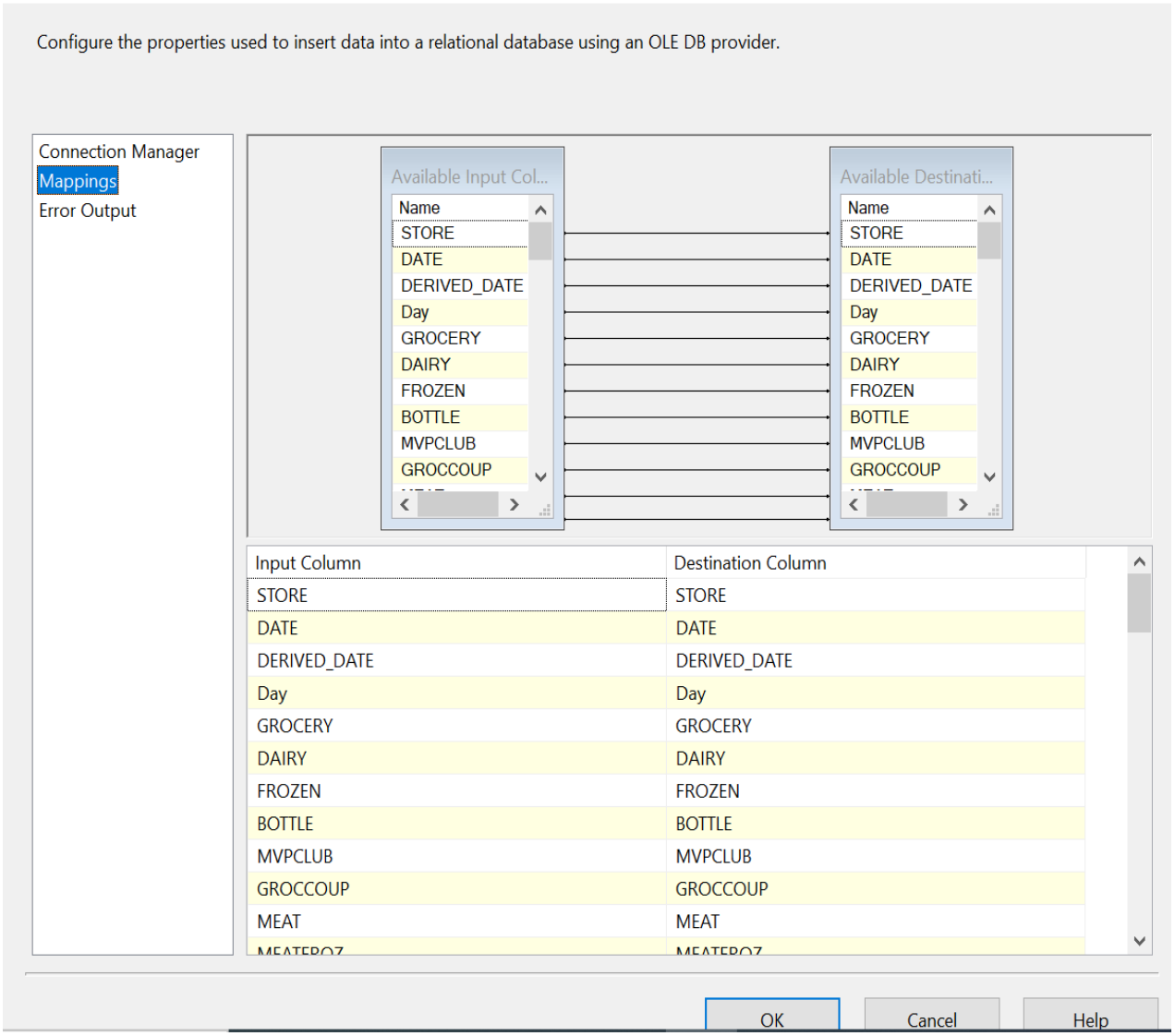
Control flow for Extraction and Transformation Process for CCount

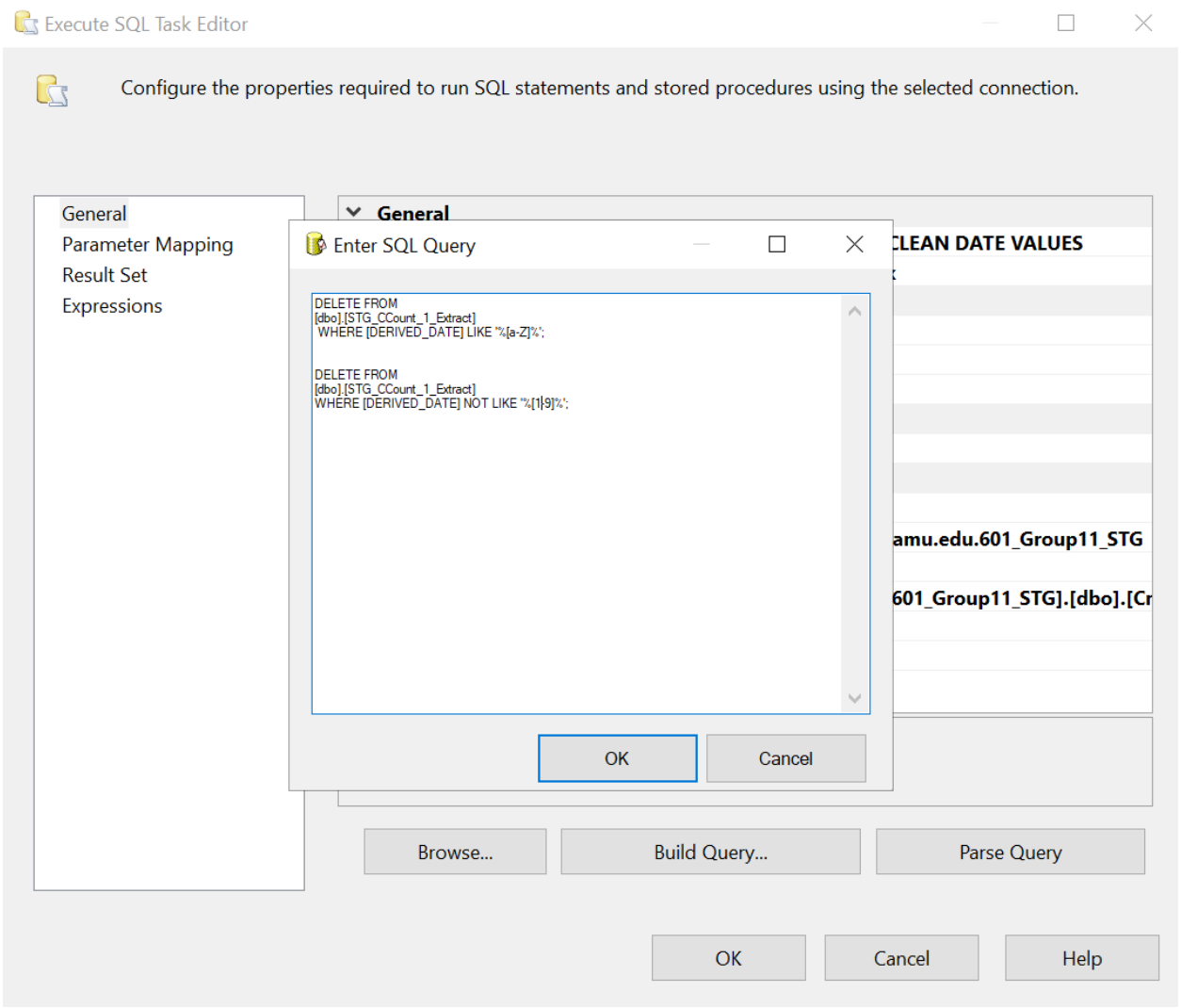


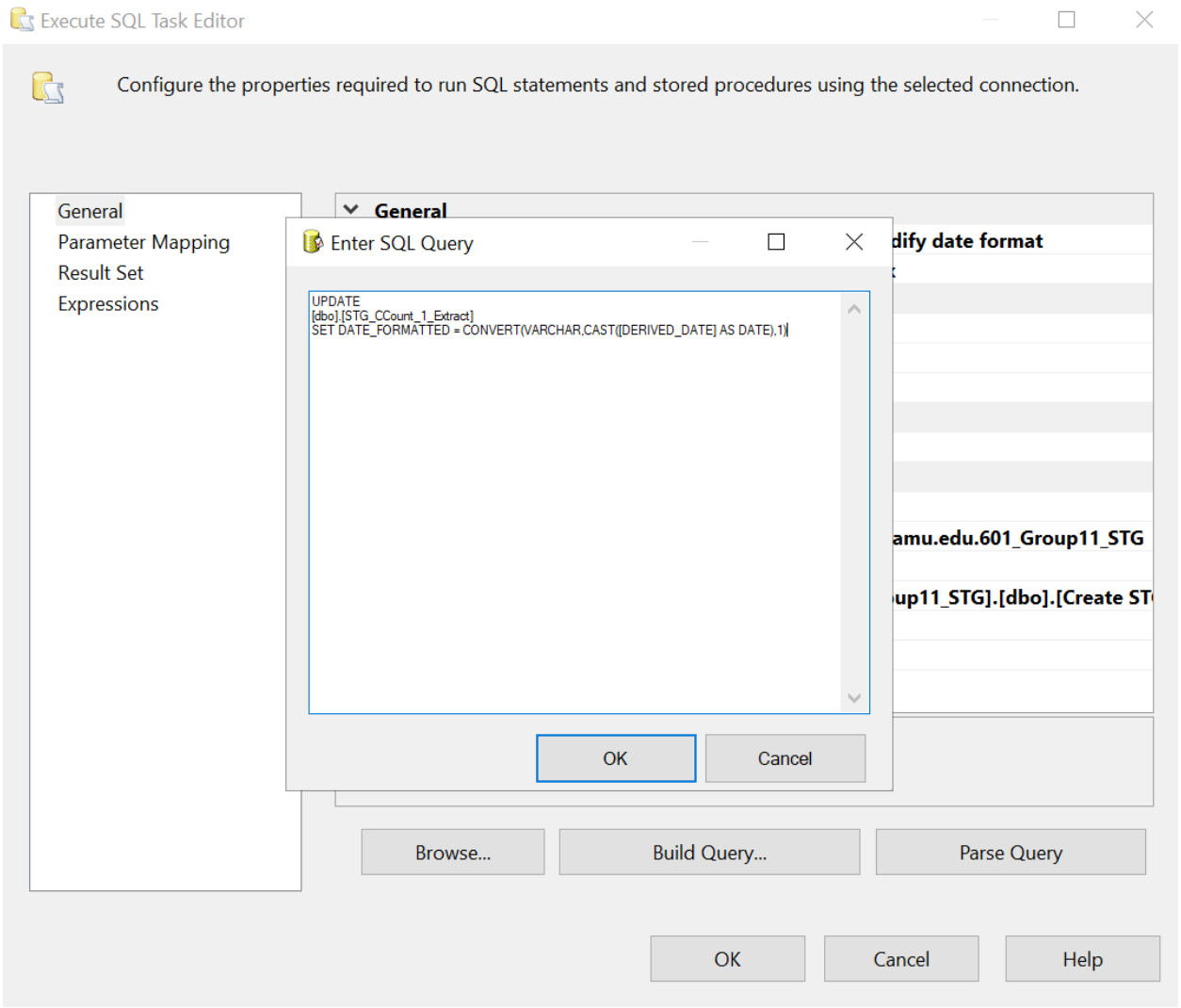
Data flow to read source CCount csv file

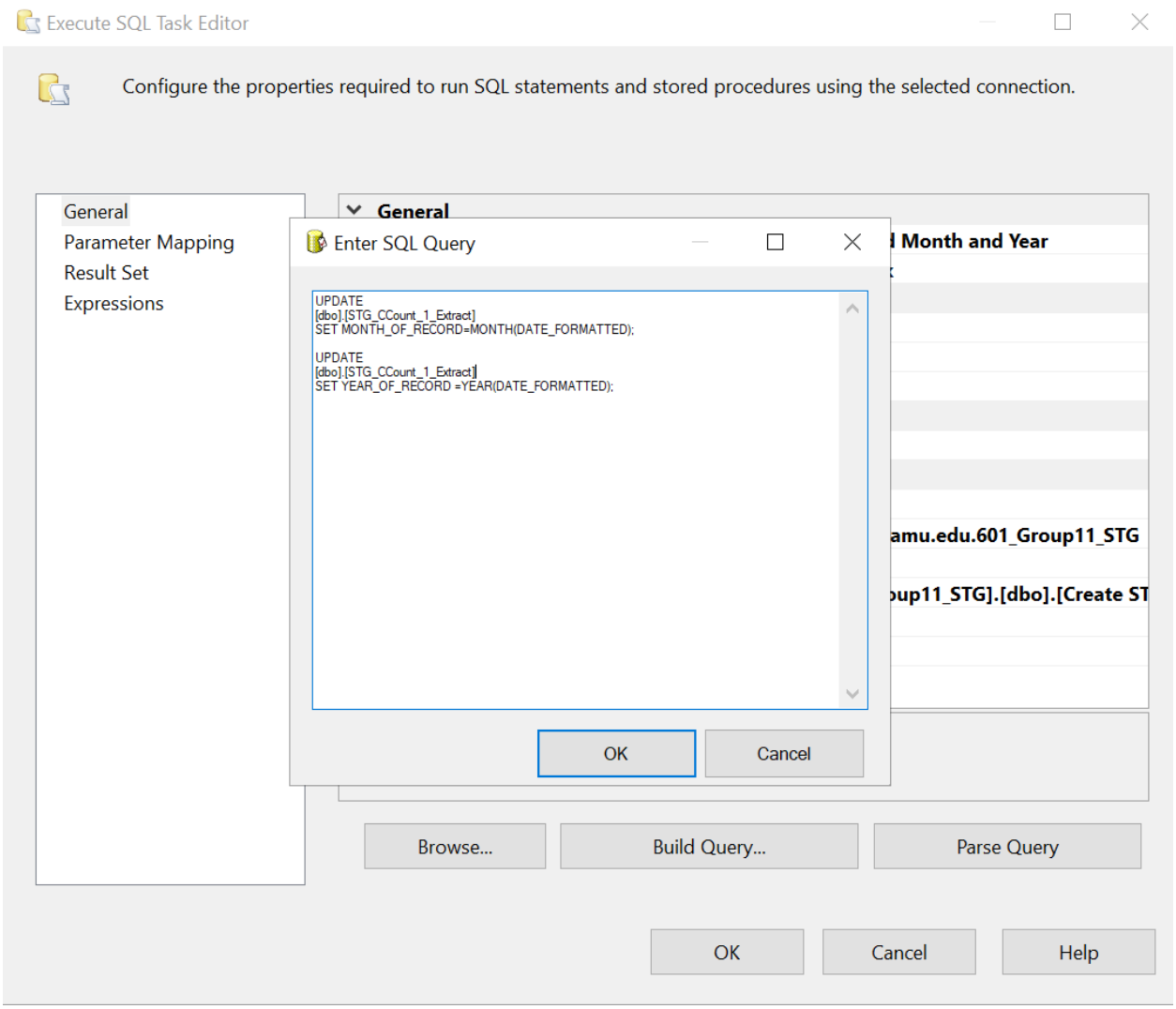


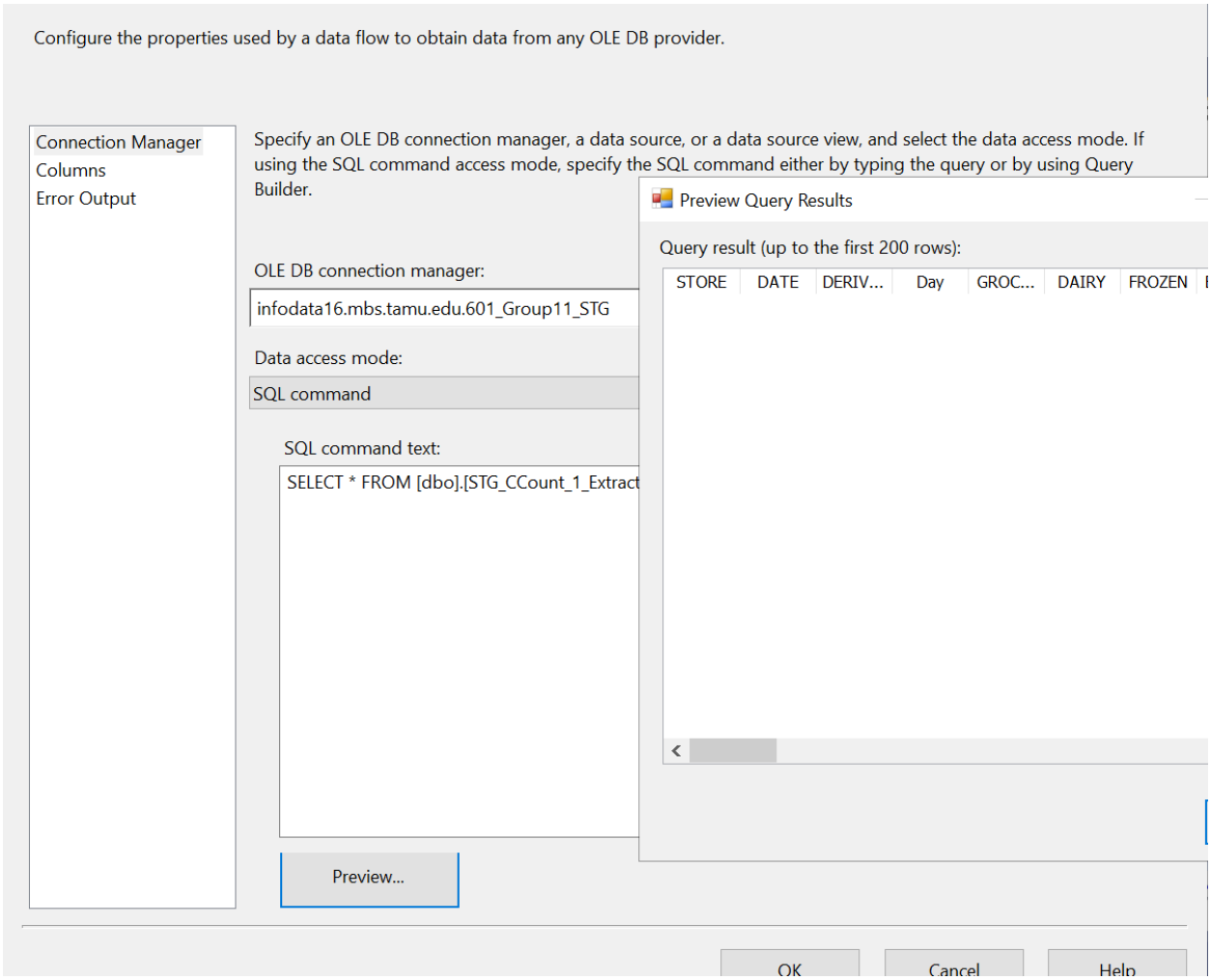

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
y or by using Query
Create Table
CREATE TABLE [STG_CCount_1_Extract] (
[STORE] varchar(50),
[DATE] varchar(50),
[DERIVED_DATE] varchar(50),
[Day] varchar(50),
[GROCERY] varchar(50),
[DAIRY] varchar(50),
[FROZEN] varchar(50),
[BOTTLE] varchar(50),
[MVPCLUB] varchar(50),
[GROCCOUP] varchar(50),
[MEAT] varchar(50),
[MEATFROZ] varchar(50),
[MEATCOUP] varchar(50),
[FISH] varchar(50),
[FISHCOUP] varchar(50),
[PROMO] varchar(50),
[PROMCOUP] varchar(50),
[PRODUCE] varchar(50),
OK
Cancel
New...
New...
View Existing Data...
OK
Cancel
Help

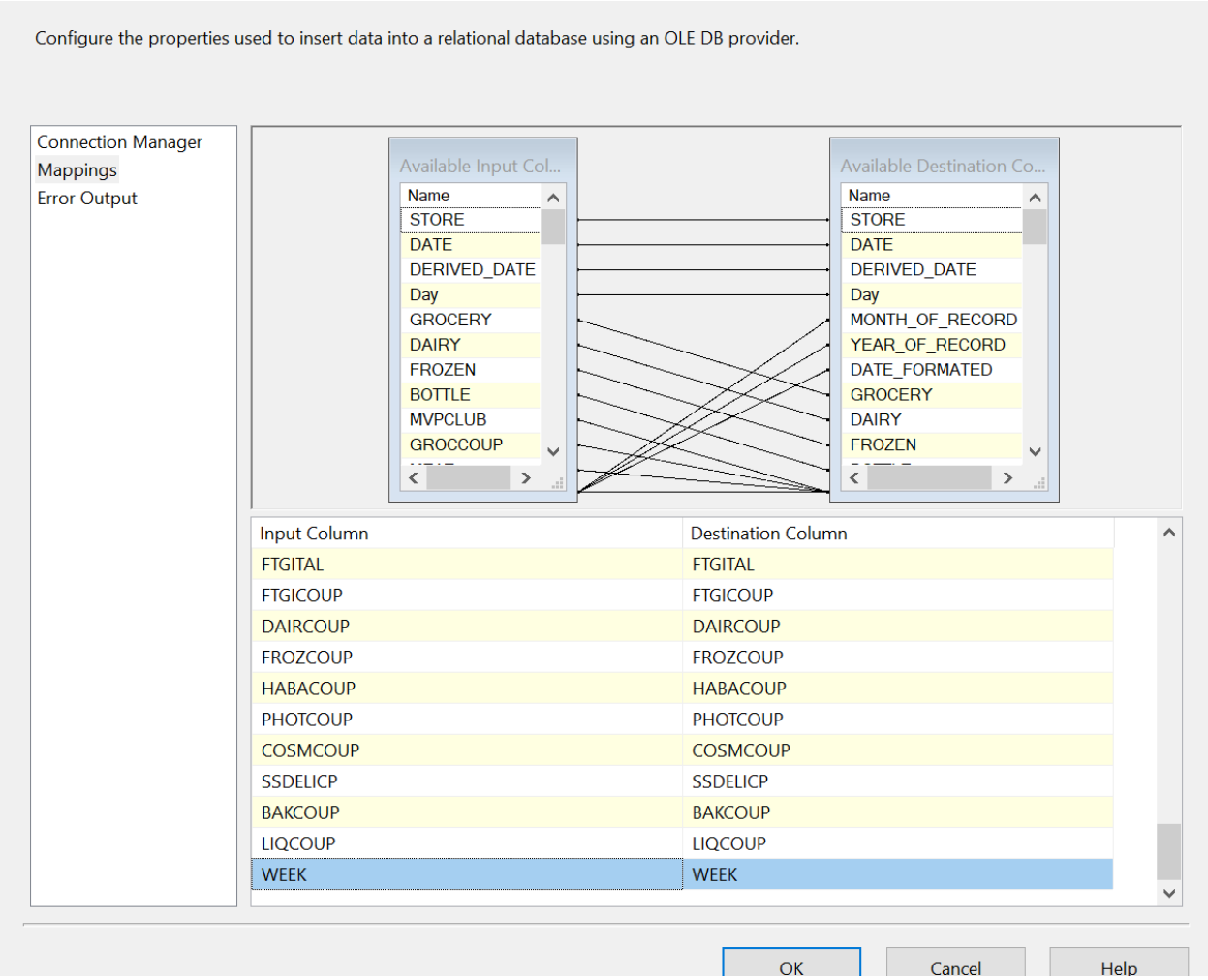

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Available Input Col...
Name
STORE
DATE
DERIVED_DATE
Day
GROCERY
DAIRY
FROZEN
BOTTLE
MVPCLUB
GROCCOUP
Available Destinati...
Name
STORE
DATE
DERIVED_DATE
Day
GROCERY
DAIRY
FROZEN
BOTTLE
MVPCLUB
GROCCOUP
Input Column
Destination Column
STORE
STORE
DATE
DATE
DERIVED_DATE
DERIVED_DATE
Day
Day
GROCERY
GROCERY
DAIRY
DAIRY
FROZEN
FROZEN
BOTTLE
BOTTLE
MVPCLUB
MVPCLUB
GROCCOUP
GROCCOUP
MEAT
MEAT
OK
Cancel
Help

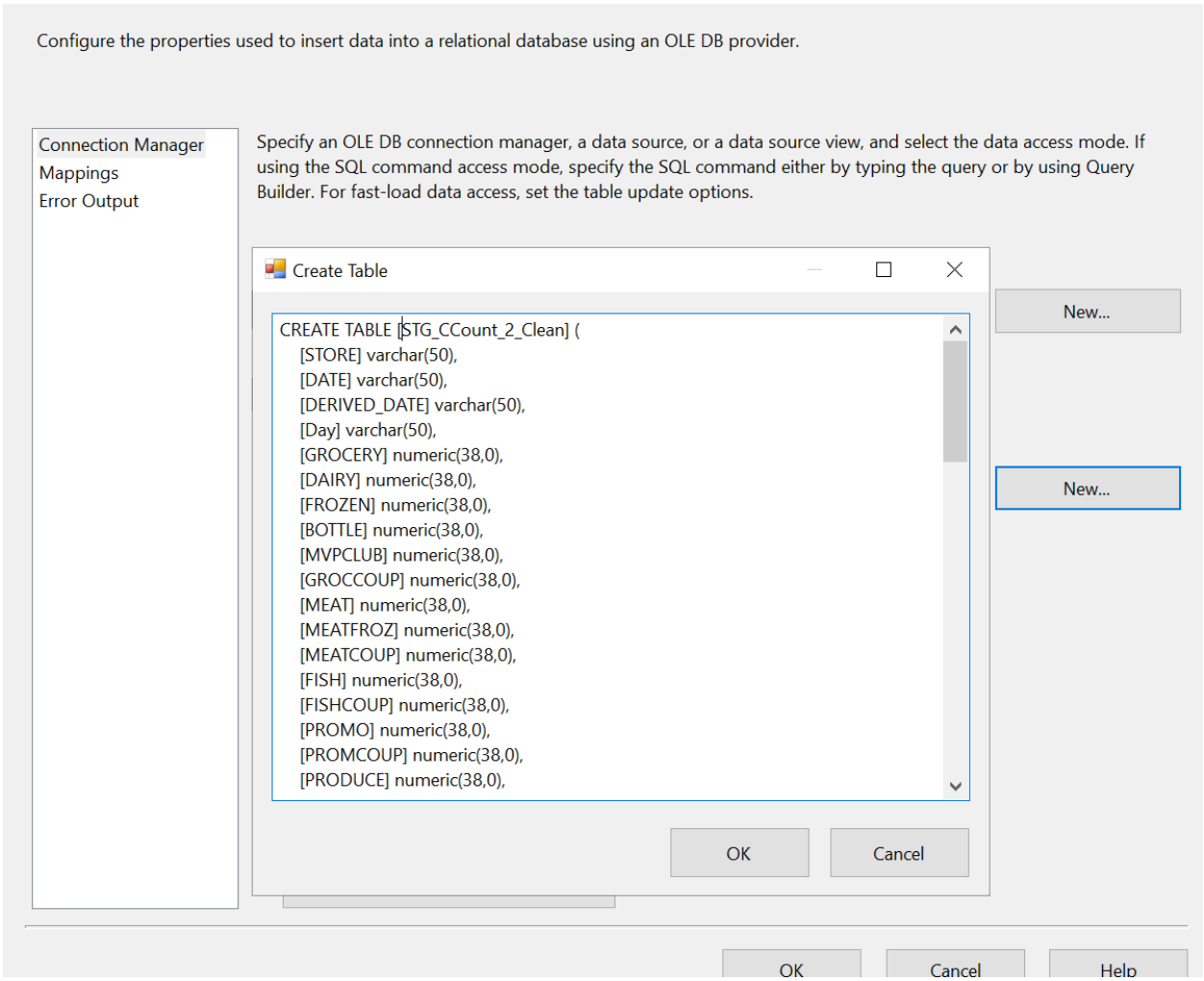

Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
General
LEAN DATE VALUES
amu.edu.601_Group11_STG
601_Group11_STG].[dbo].[Cr
Enter SQL Query
DELETE FROM
[dbo].[STG_CCount_1_Extract]
WHERE [DERIVED_DATE] LIKE '%[a-Z]%';
DELETE FROM
[dbo].[STG_CCount_1_Extract]
WHERE [DERIVED_DATE] NOT LIKE '%[1|9]%';
OK
Cancel
Browse...
Build Query...
Parse Query
OK
Cancel
Help

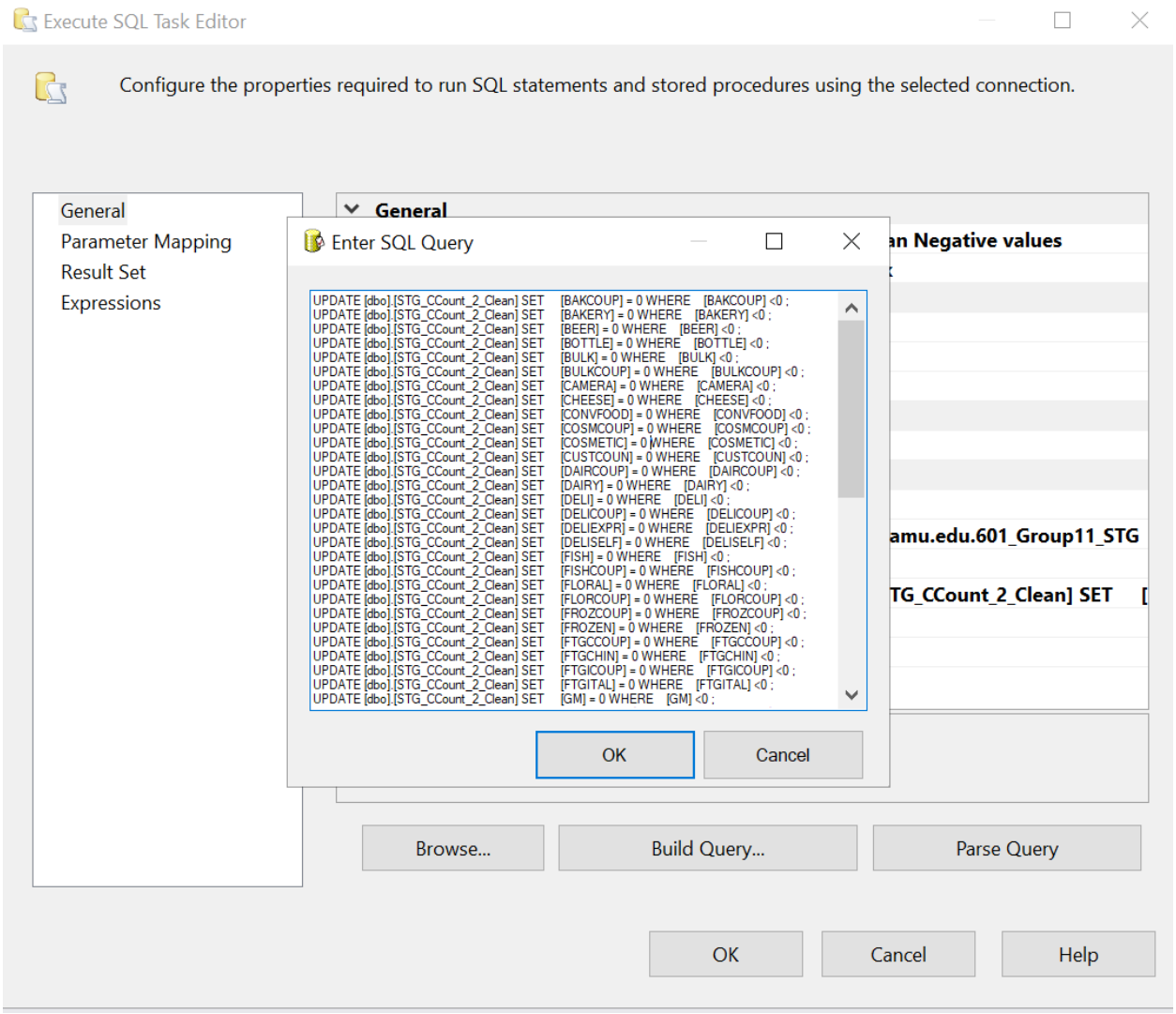

Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
General
Enter SQL Query
UPDATE
[dbo].[STG_CCount_1_Extract]
SET DATE_FORMATTED = CONVERT(VARCHAR,CAST([DERIVED_DATE] AS DATE),1)
OK
Cancel
dify date format
amu.edu.601_Group11_STG
up11_STG].[dbo].[Create ST
Browse...
Build Query...
Parse Query
OK
Cancel
Help

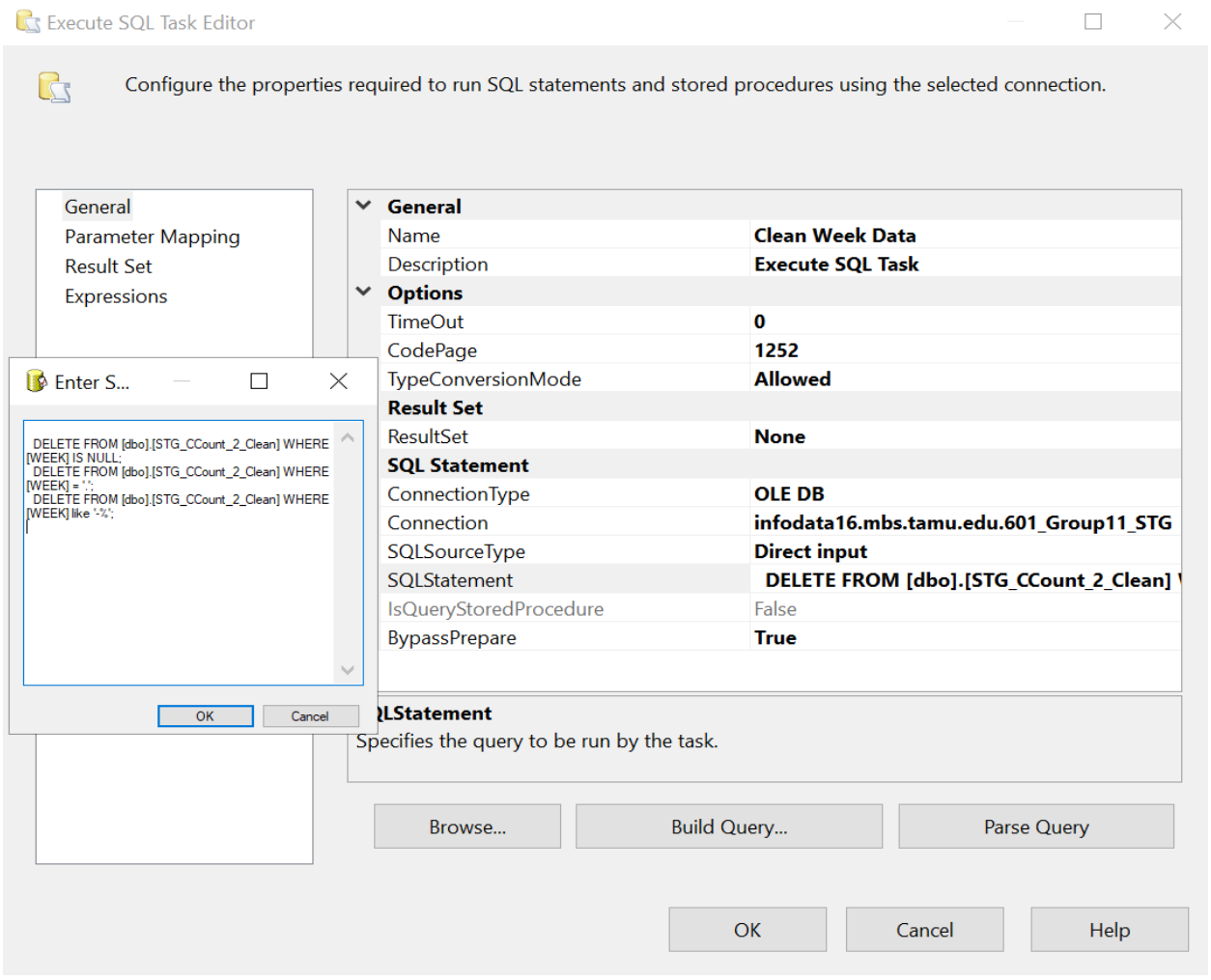

Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
General
Month and Year
amu.edu.601_Group11_STG
oup11_STG].[dbo].[Create ST
Enter SQL Query
UPDATE
[dbo].[STG_CCount_1_Extract]
SET MONTH_OF_RECORD=MONTH(DATE_FORMATTED);
UPDATE
[dbo].[STG_CCount_1_Extract]
SET YEAR_OF_RECORD =YEAR(DATE_FORMATTED);
OK
Cancel
Browse...
Build Query...
Parse Query
OK
Cancel
Help

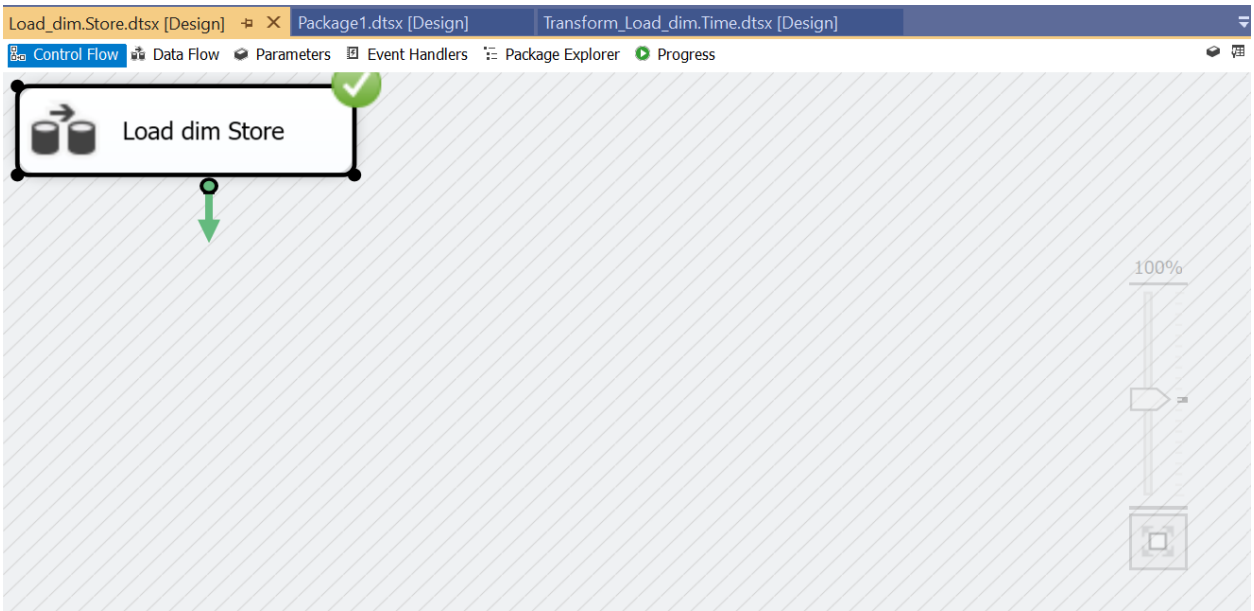

Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
Data access mode:
SQL command
SQL command text:
SELECT * FROM [dbo].[STG_CCount_1_Extract
Preview...
Preview Query Results
Query result (up to the first 200 rows):
STORE
DATE
DERIV...
Day
GROC...
DAIRY
FROZEN
OK
Cancel
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Available Input Col...
Name
STORE
DATE
DERIVED_DATE
Day
GROCERY
DAIRY
FROZEN
BOTTLE
MVPCLUB
GROCCOUP
Available Destination Co...
Name
STORE
DATE
DERIVED_DATE
Day
MONTH_OF_RECORD
YEAR_OF_RECORD
DATE_FORMATED
GROCERY
DAIRY
FROZEN
Input Column
Destination Column
FTGITAL
FTGICOUP
DAIRCOUP
FROZCOUP
HABACOUP
PHOTCOUP
COSMCOUP
SSDELICP
BAKCOUP
LIQCOUP
WEEK
OK
Cancel
Help

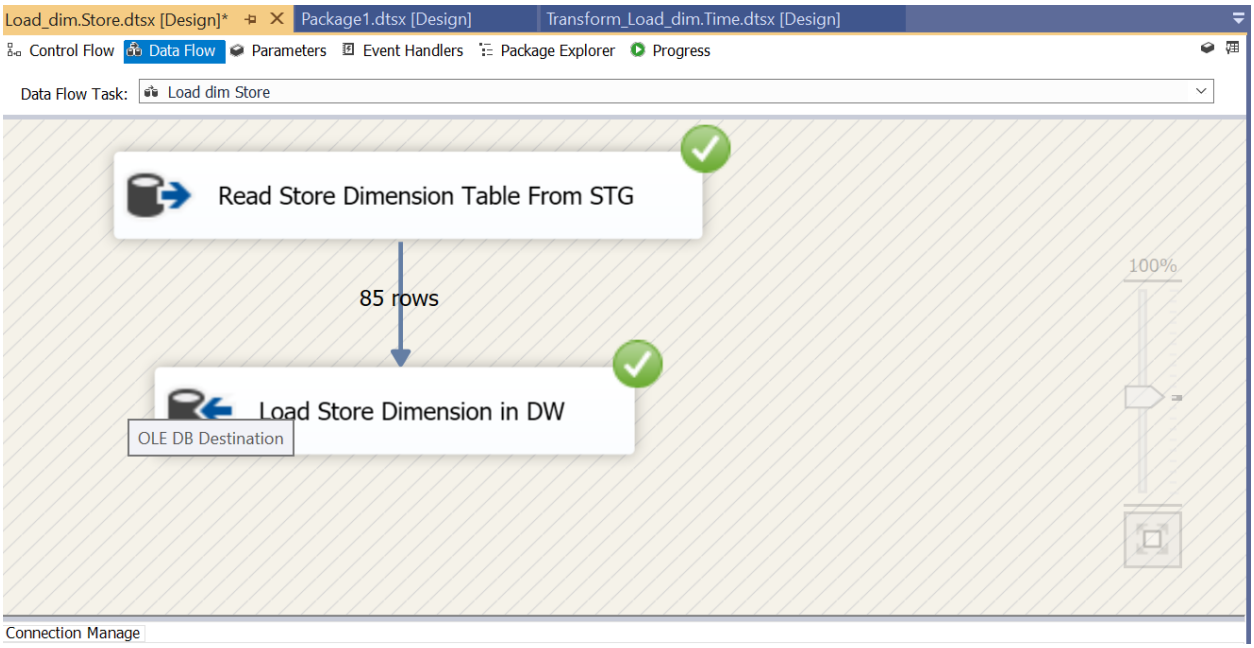

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
Create Table
CREATE TABLE [STG_CCount_2_Clean] (
[STORE] varchar(50),
[DATE] varchar(50),
[DERIVED_DATE] varchar(50),
[Day] varchar(50),
[GROCERY] numeric(38,0),
[DAIRY] numeric(38,0),
[FROZEN] numeric(38,0),
[BOTTLE] numeric(38,0),
[MVPCLUB] numeric(38,0),
[GROCCOUP] numeric(38,0),
[MEAT] numeric(38,0),
[MEATFROZ] numeric(38,0),
[MEATCOUP] numeric(38,0),
[FISH] numeric(38,0),
[FISHCOUP] numeric(38,0),
[PROMO] numeric(38,0),
[PROMCOUP] numeric(38,0),
[PRODUCE] numeric(38,0),
OK
Cancel
New...
New...
OK
Cancel
Help


Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
General
Parameter Mapping
Result Set
Expressions
General
Enter SQL Query
UPDATE [dbo].[STG_CCount_2_Clean] SET [BAKCOUP] = 0 WHERE [BAKCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [BAKERY] = 0 WHERE [BAKERY] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [BEER] = 0 WHERE [BEER] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [BOTTLE] = 0 WHERE [BOTTLE] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [BULK] = 0 WHERE [BULK] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [BULKCOUP] = 0 WHERE [BULKCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [CAMERA] = 0 WHERE [CAMERA] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [CHEESE] = 0 WHERE [CHEESE] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [CONVFOOD] = 0 WHERE [CONVFOOD] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [COSMCOUP] = 0 WHERE [COSMCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [COSMETIC] = 0 WHERE [COSMETIC] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [CUSTCOUN] = 0 WHERE [CUSTCOUN] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [DAIRCOUP] = 0 WHERE [DAIRCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [DAIRY] = 0 WHERE [DAIRY] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [DELI] = 0 WHERE [DELI] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [DELICOUP] = 0 WHERE [DELICOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [DELIEXPR] = 0 WHERE [DELIEXPR] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [DELISELF] = 0 WHERE [DELISELF] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FISH] = 0 WHERE [FISH] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FISHCOUP] = 0 WHERE [FISHCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FLORAL] = 0 WHERE [FLORAL] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FLORCOUP] = 0 WHERE [FLORCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FROZCOUP] = 0 WHERE [FROZCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FROZEN] = 0 WHERE [FROZEN] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FTGCCOUP] = 0 WHERE [FTGCCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FTGCHIN] = 0 WHERE [FTGCHIN] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FTGICOUP] = 0 WHERE [FTGICOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [FTGITAL] = 0 WHERE [FTGITAL] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET [GM] = 0 WHERE [GM] <0 ;
OK
Cancel
an Negative values
amu.edu.601_Group11_STG
TG_CCount_2_Clean] SET
Browse...
Build Query...
Parse Query
OK
Cancel
Help

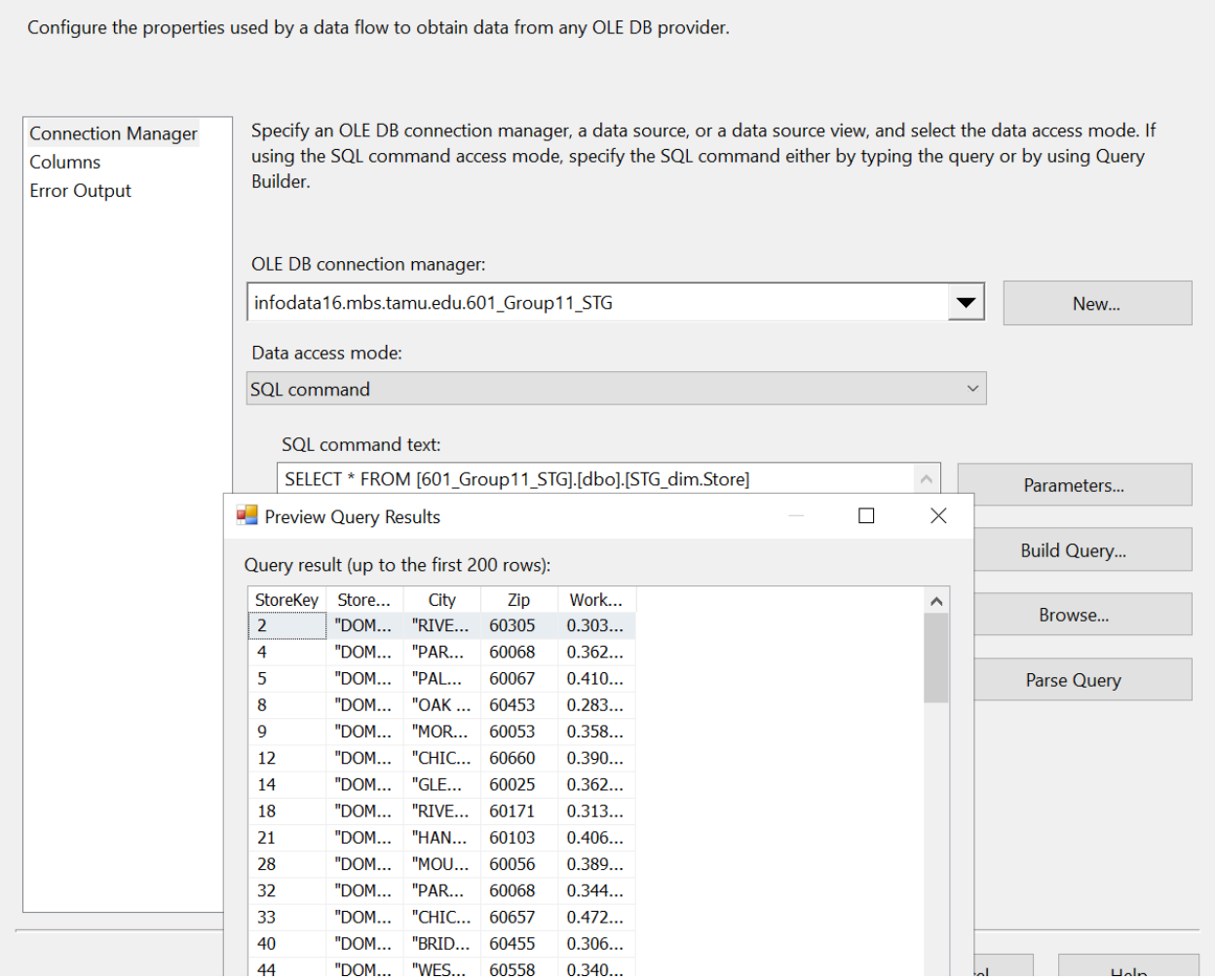


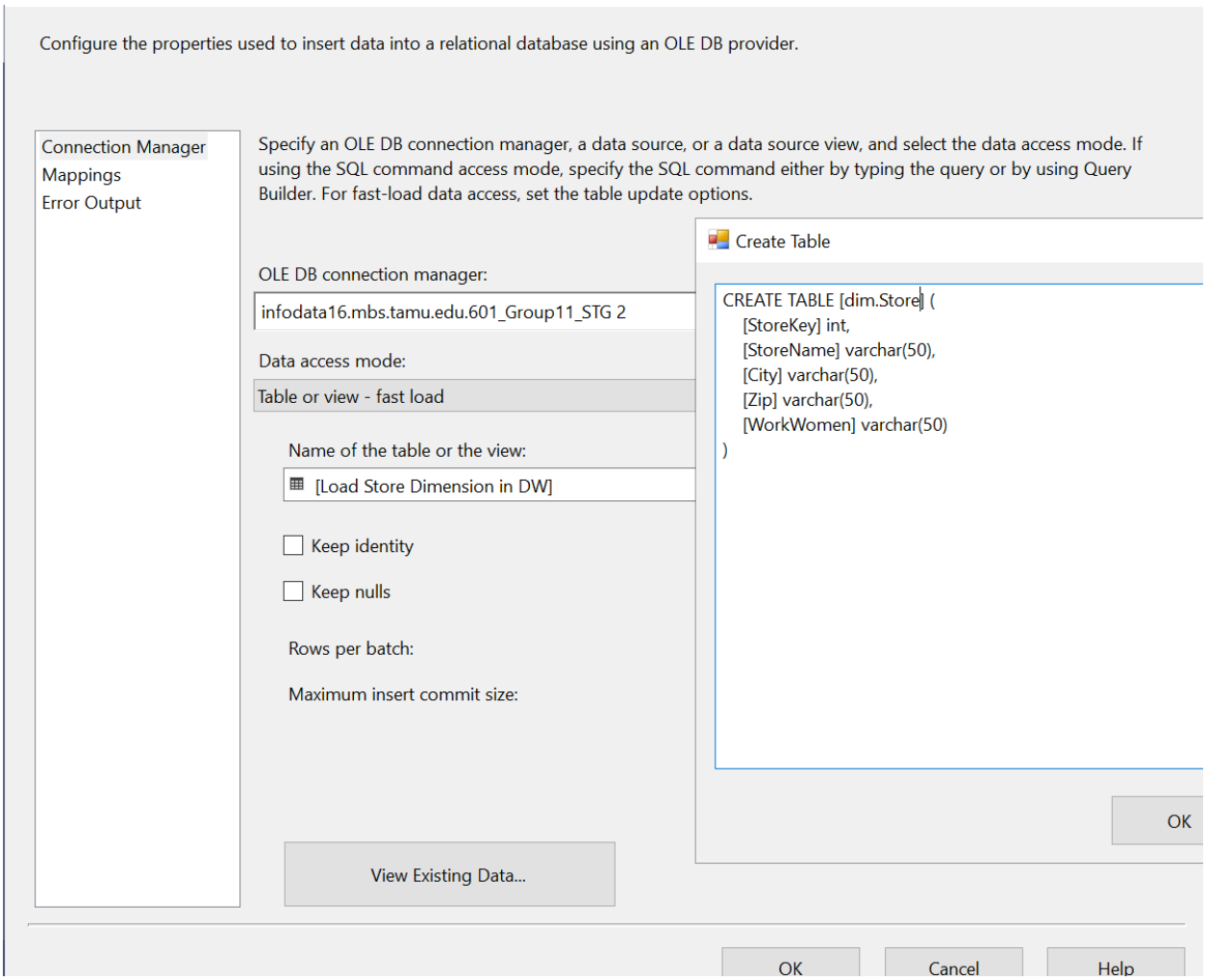


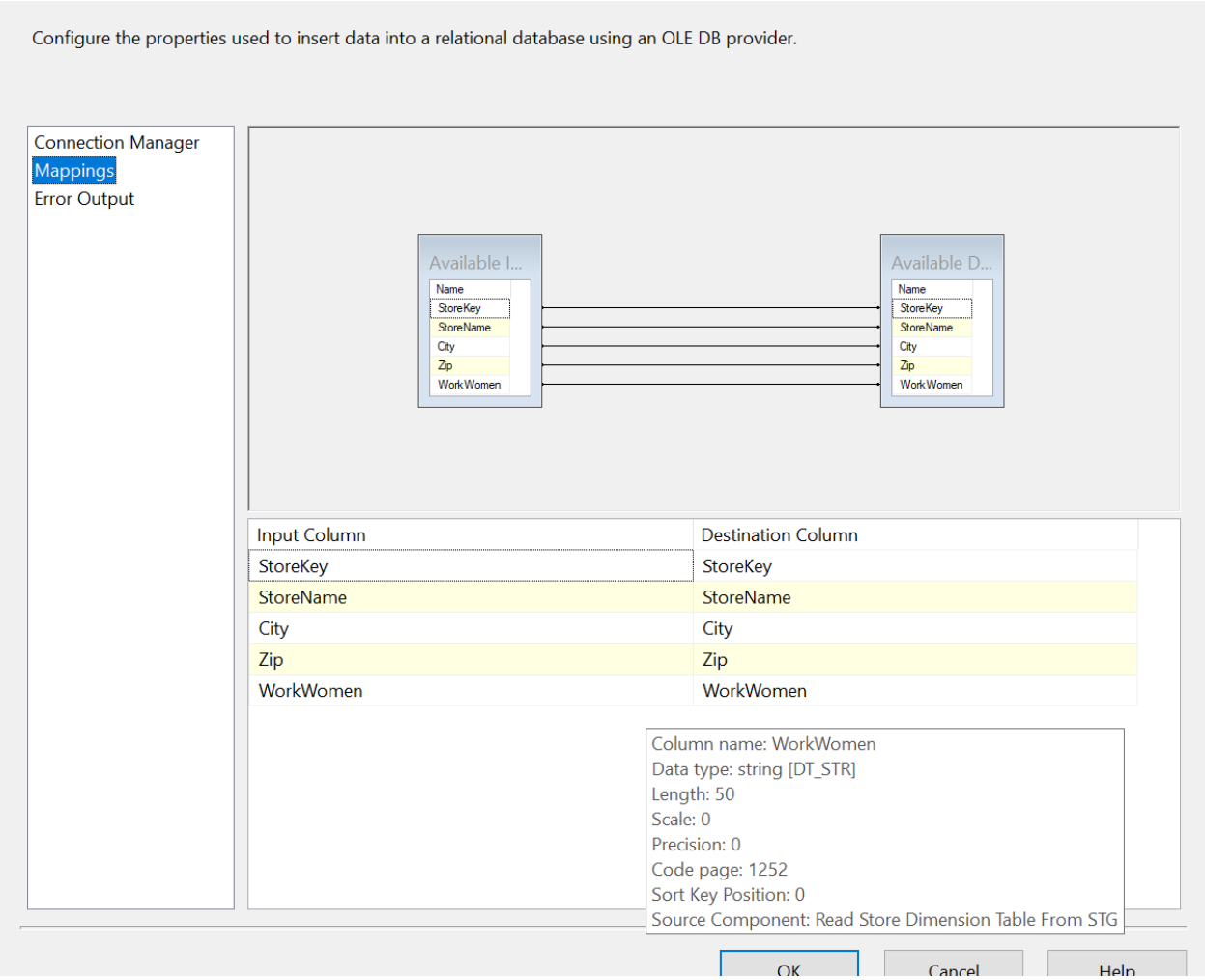
Control Flow for loading Store Dimension



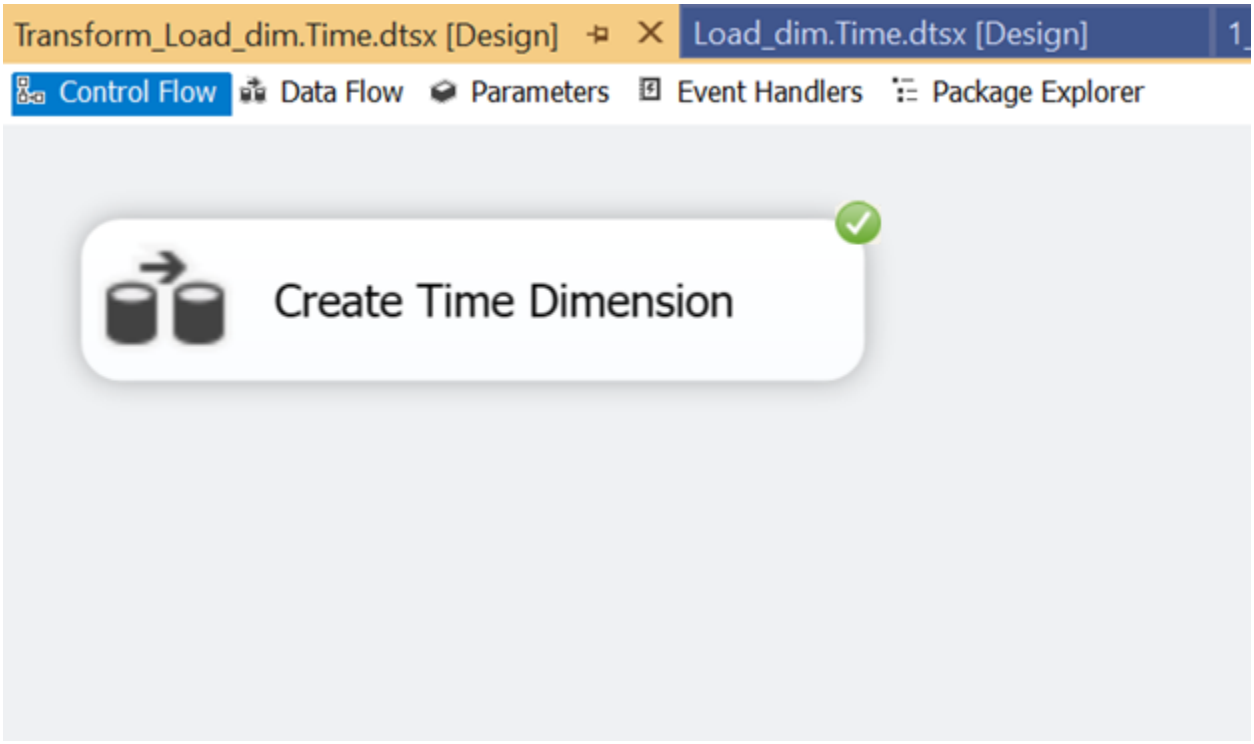
Data Flow to load dim Store


Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
New...
Data access mode:
SQL command
SQL command text:
SELECT * FROM [601_Group11_STG].[dbo].[STG_dim.Store]
Parameters...
Build Query...
Browse...
Parse Query
Preview Query Results
Query result (up to the first 200 rows):
StoreKey Store... City Zip Work...
2 "DOM... "RIVE... 60305 0.303...
4 "DOM... "PAR... 60068 0.362...
5 "DOM... "PAL... 60067 0.410...
8 "DOM... "OAK ... 60453 0.283...
9 "DOM... "MOR... 60053 0.358...
12 "DOM... "CHIC... 60660 0.390...
14 "DOM... "GLE... 60025 0.362...
18 "DOM... "RIVE... 60171 0.313...
21 "DOM... "HAN... 60103 0.406...
28 "DOM... "MOU... 60056 0.389...
32 "DOM... "PAR... 60068 0.344...
33 "DOM... "CHIC... 60657 0.472...
40 "DOM... "BRID... 60455 0.306...
44 "DOM... "WES... 60558 0.340...

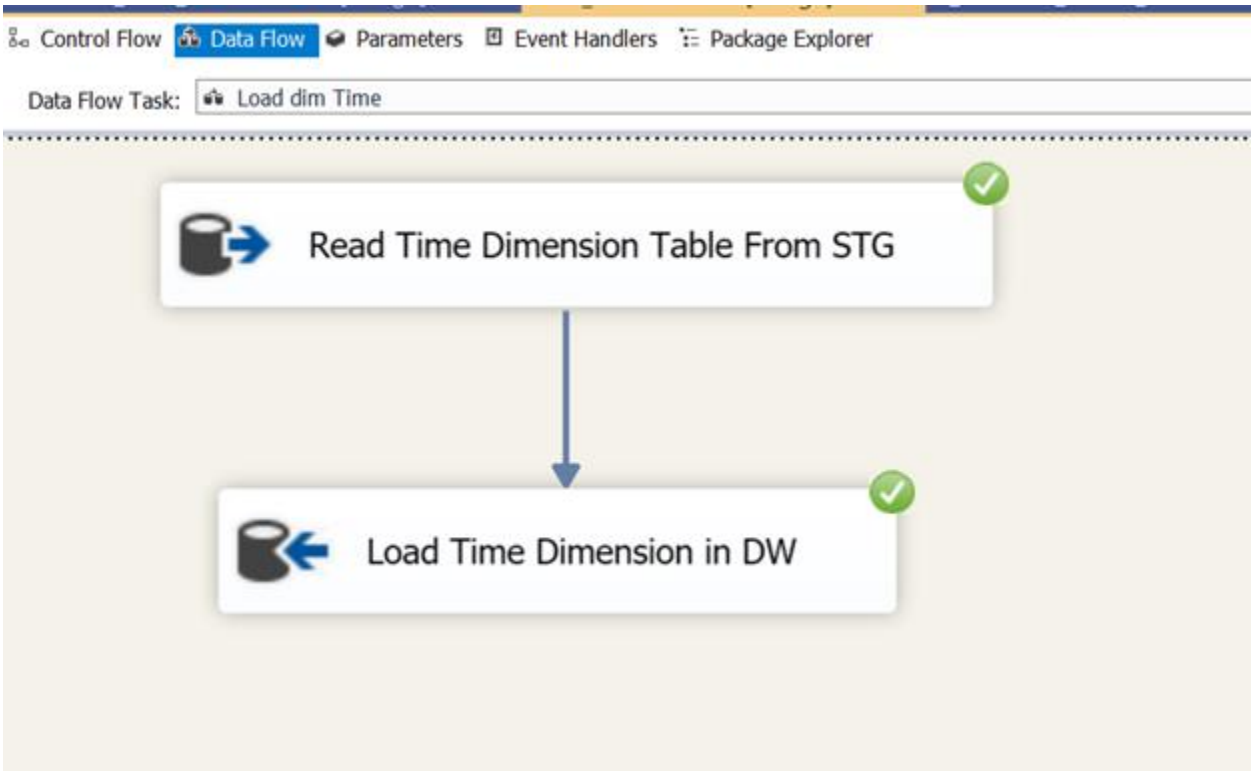

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG 2
Data access mode:
Table or view - fast load
Name of the table or the view:
[Load Store Dimension in DW]
Keep identity
Keep nulls
Rows per batch:
Maximum insert commit size:
View Existing Data...
Create Table
CREATE TABLE [dim.Store] (
[StoreKey] int,
[StoreName] varchar(50),
[City] varchar(50),
[Zip] varchar(50),
[WorkWomen] varchar(50)
)
OK
OK
Cancel
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Available I...
Name
StoreKey
StoreName
City
Zip
WorkWomen
Available D...
Name
StoreKey
StoreName
City
Zip
WorkWomen
Input Column
Destination Column
StoreKey
StoreKey
StoreName
StoreName
City
City
Zip
Zip
WorkWomen
WorkWomen
Column name: WorkWomen
Data type: string [DT_STR]
Length: 50
Scale: 0
Precision: 0
Code page: 1252
Sort Key Position: 0
Source Component: Read Store Dimension Table From STG
OK
Cancel
Help

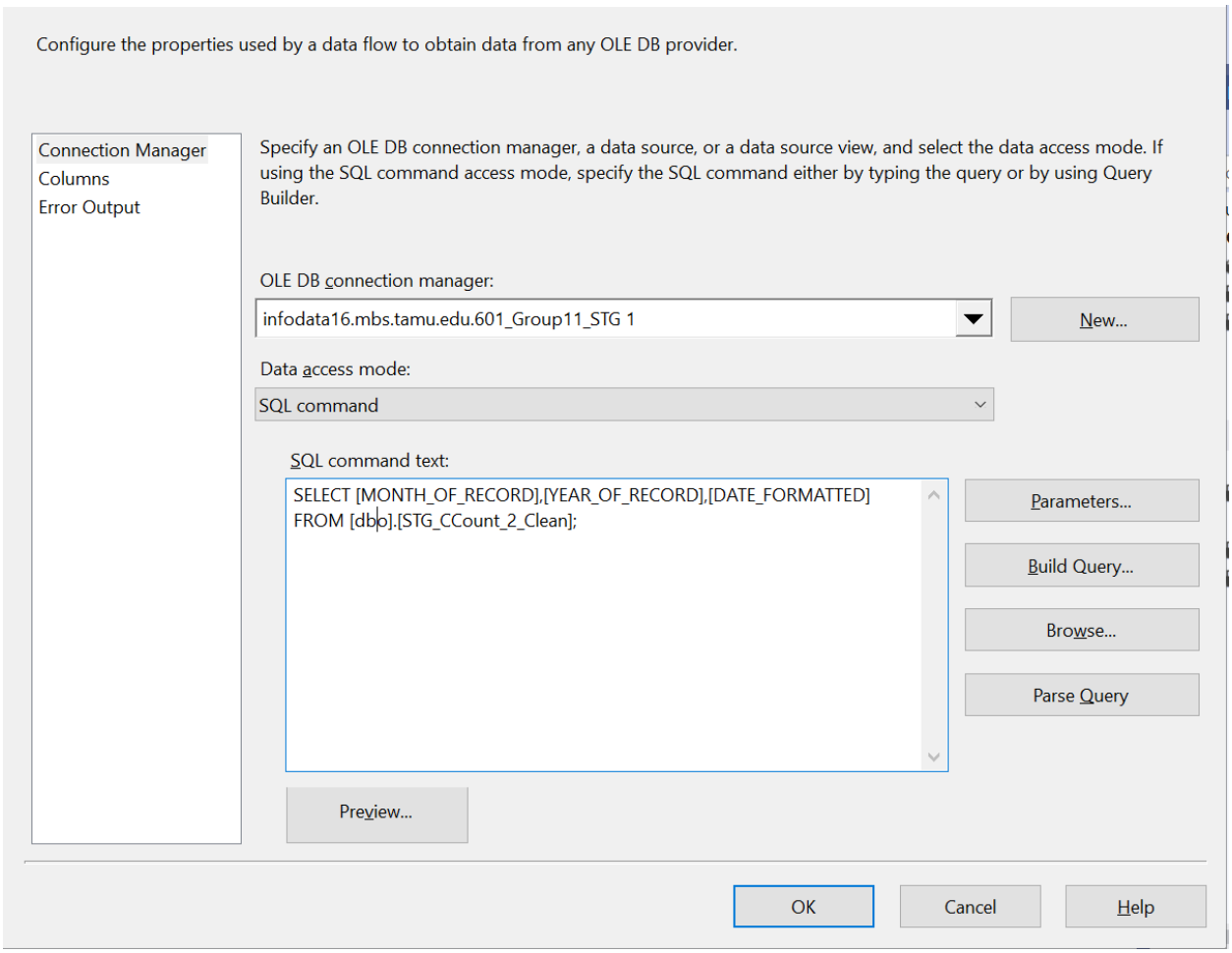
ETL for Time Dimension:

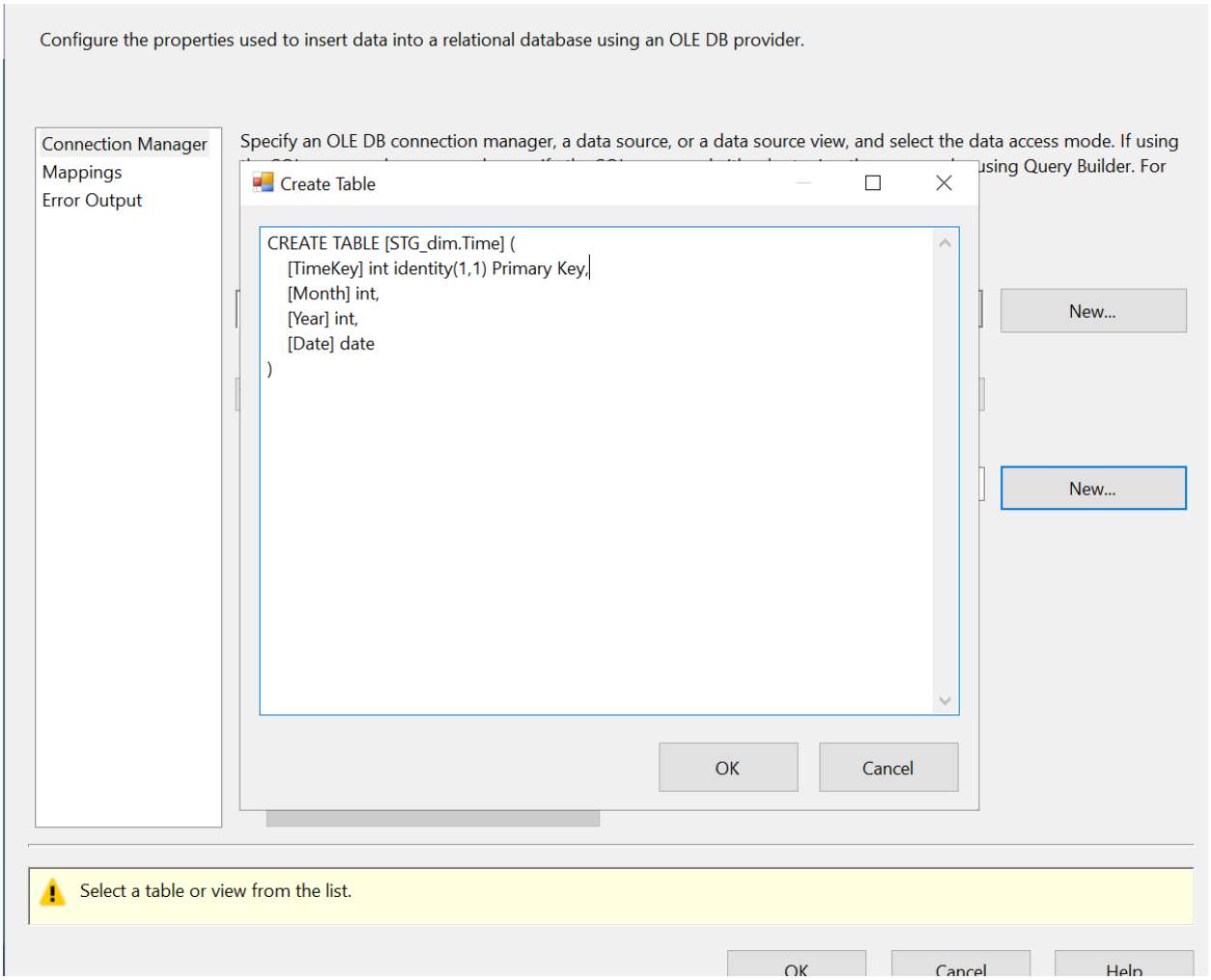


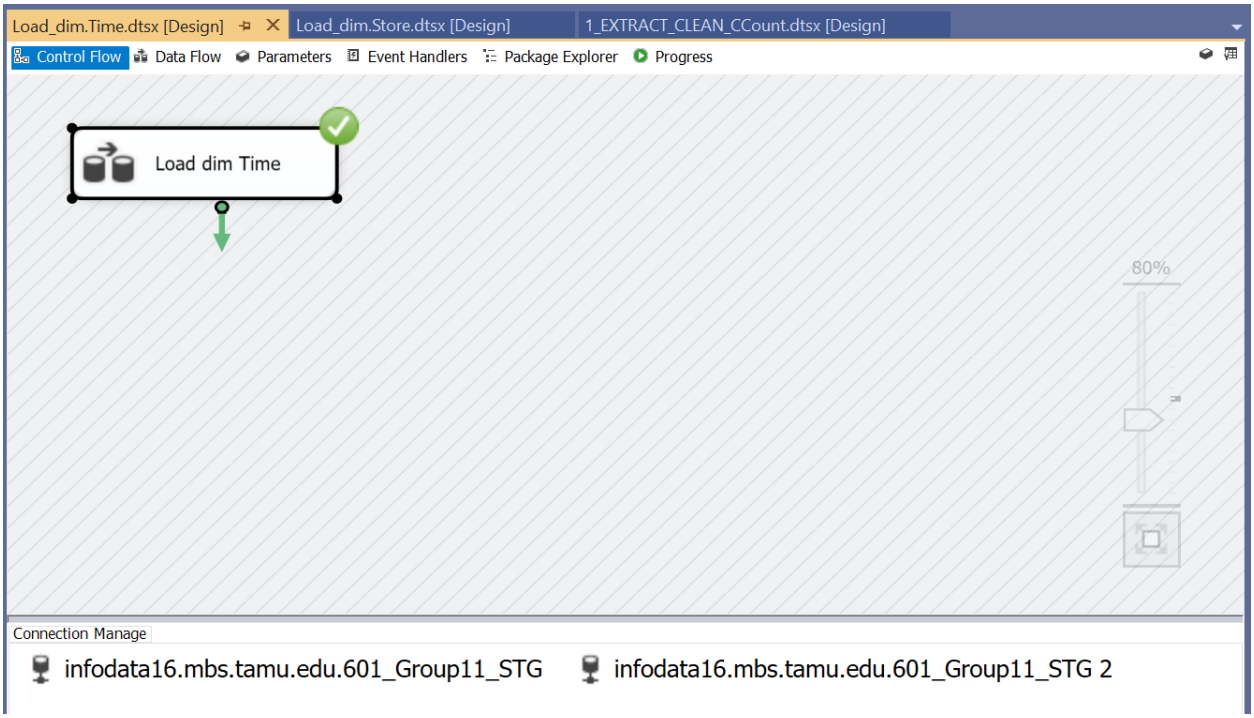
Control Flow for Time Dimension creation



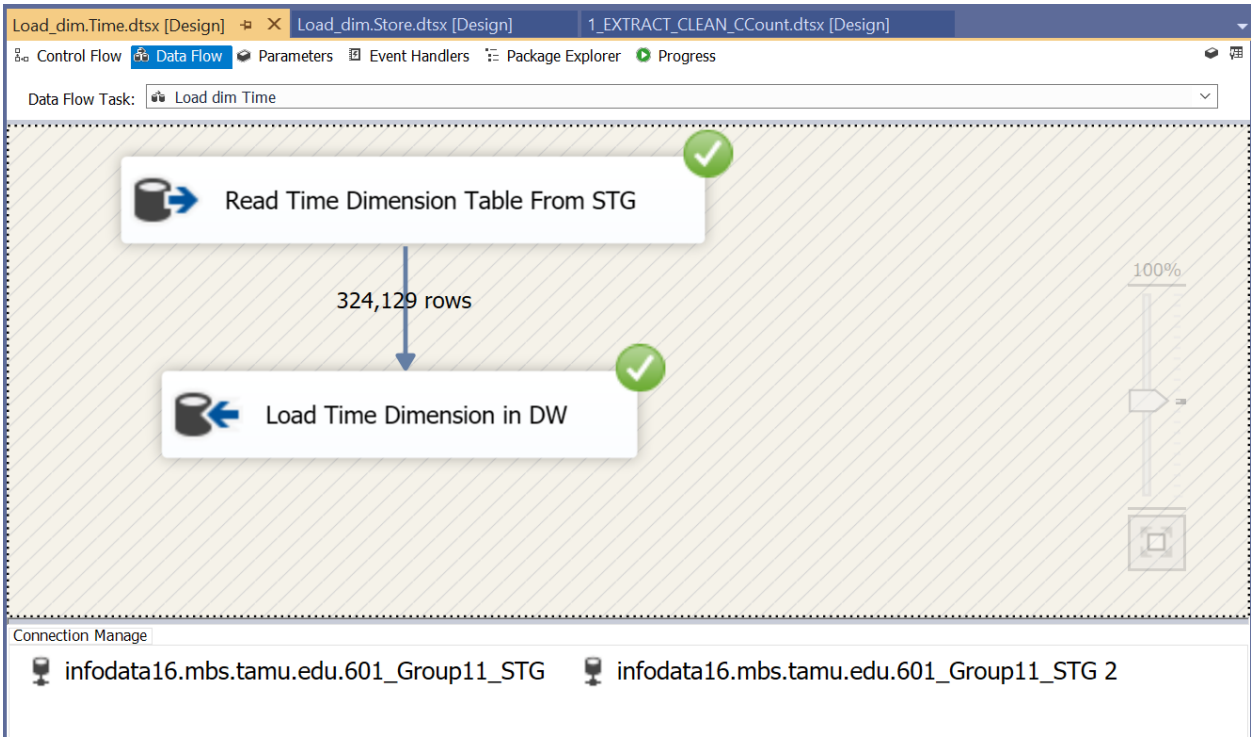
Data Flow for loading Time Dimension


Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG 1
New...
Data access mode:
SQL command
SQL command text:
SELECT [MONTH_OF_RECORD],[YEAR_OF_RECORD],[DATE_FORMATTED]
FROM [dbo].[STG_CCount_2_Clean];
Parameters...
Build Query...
Browse...
Parse Query
Preview...
OK
Cancel
Help

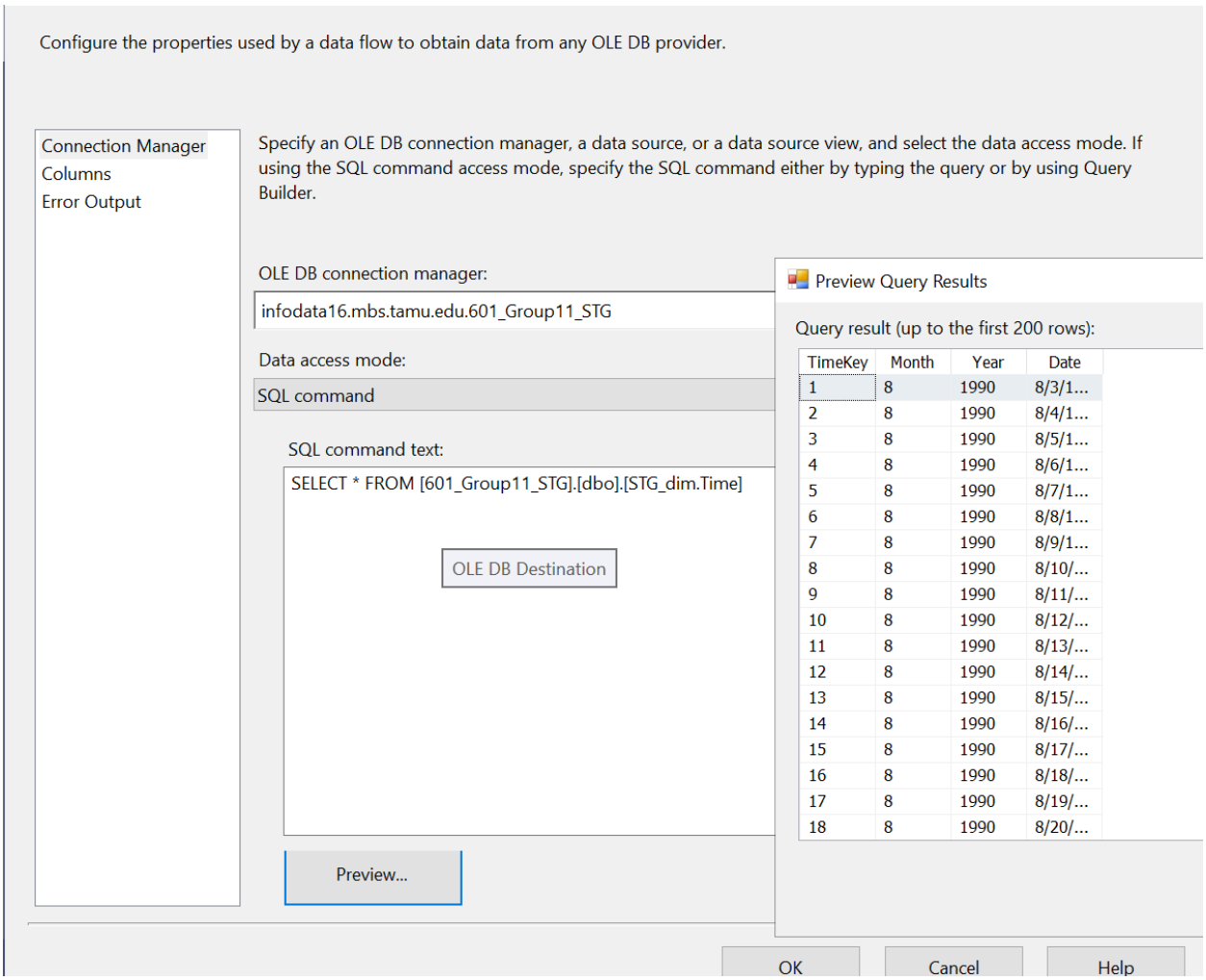

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using
using Query Builder. For
Create Table
CREATE TABLE [STG_dim.Time] (
[TimeKey] int identity(1,1) Primary Key,
[Month] int,
[Year] int,
[Date] date
)
OK
Cancel
New...
New...
Select a table or view from the list.
OK
Cancel
Help

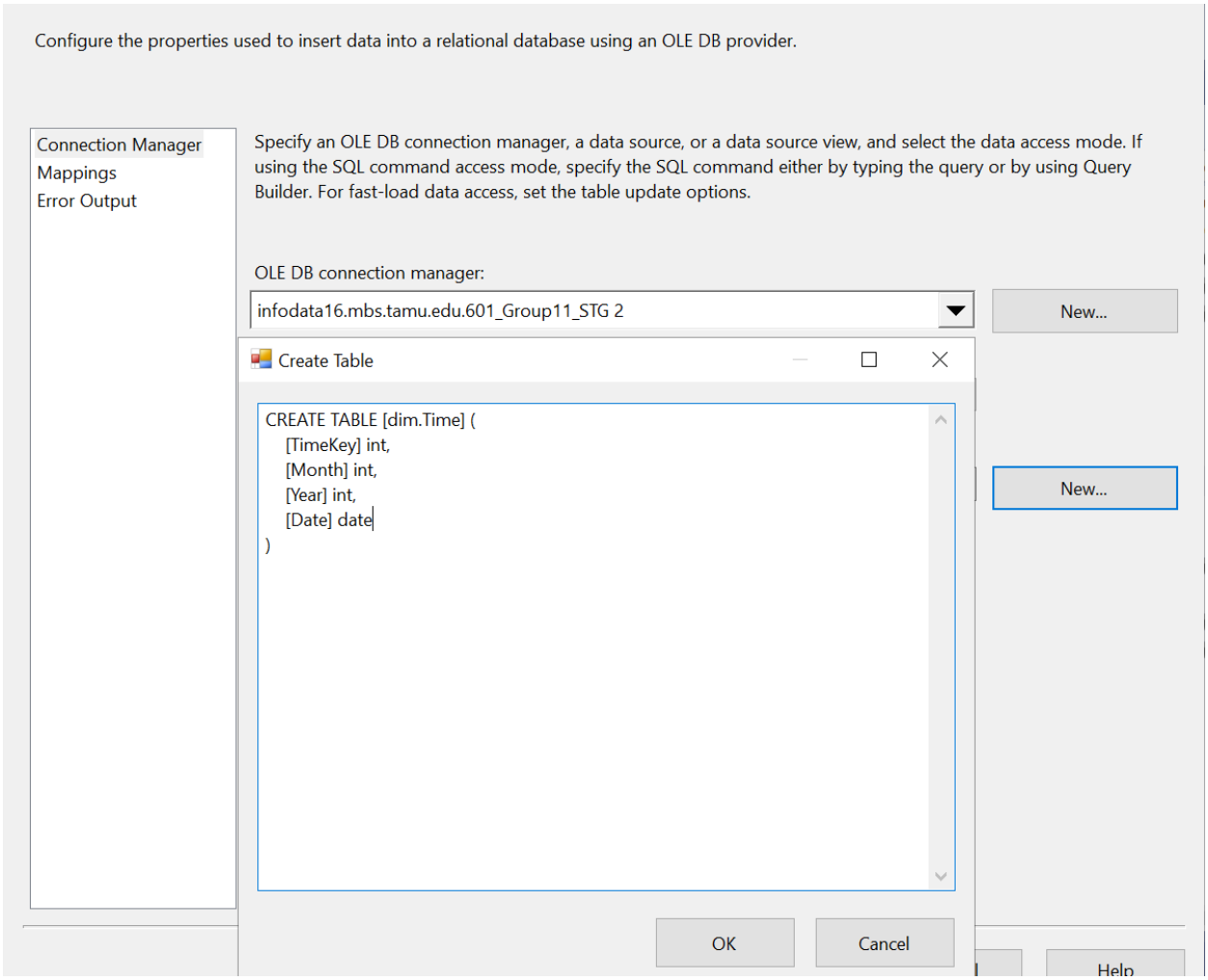


Control Flow for loading Time Dimension



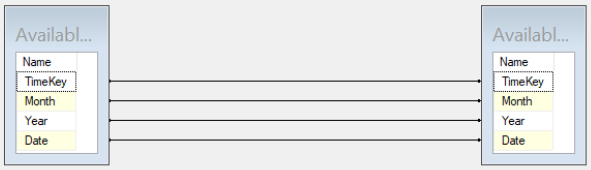
Data Flow for load dim Time


Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
Data access mode:
SQL command
SQL command text:
SELECT * FROM [601_Group11_STG].[dbo].[STG_dim.Time]
OLE DB Destination
Preview...
Preview Query Results
Query result (up to the first 200 rows):
TimeKey Month Year Date
1 8 1990 8/3/1...
2 8 1990 8/4/1...
3 8 1990 8/5/1...
4 8 1990 8/6/1...
5 8 1990 8/7/1...
6 8 1990 8/8/1...
7 8 1990 8/9/1...
8 8 1990 8/10/...
9 8 1990 8/11/...
10 8 1990 8/12/...
11 8 1990 8/13/...
12 8 1990 8/14/...
13 8 1990 8/15/...
14 8 1990 8/16/...
15 8 1990 8/17/...
16 8 1990 8/18/...
17 8 1990 8/19/...
18 8 1990 8/20/...
OK
Cancel
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG 2
New...
Create Table
CREATE TABLE [dim.Time] (
[TimeKey] int,
[Month] int,
[Year] int,
[Date] date
)
New...
OK
Cancel
Help

Configure the properties used to insert data into a relational database using an OLE DB provider.

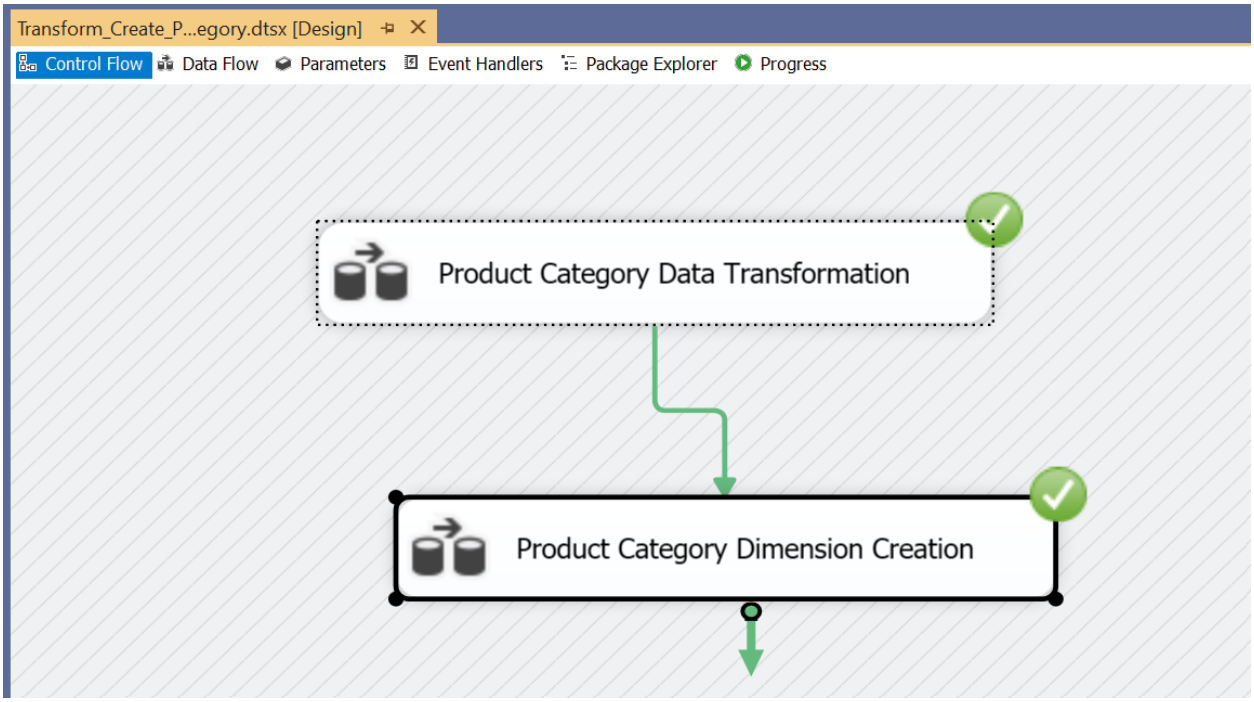
Connection Manager
Mappings
Error Output



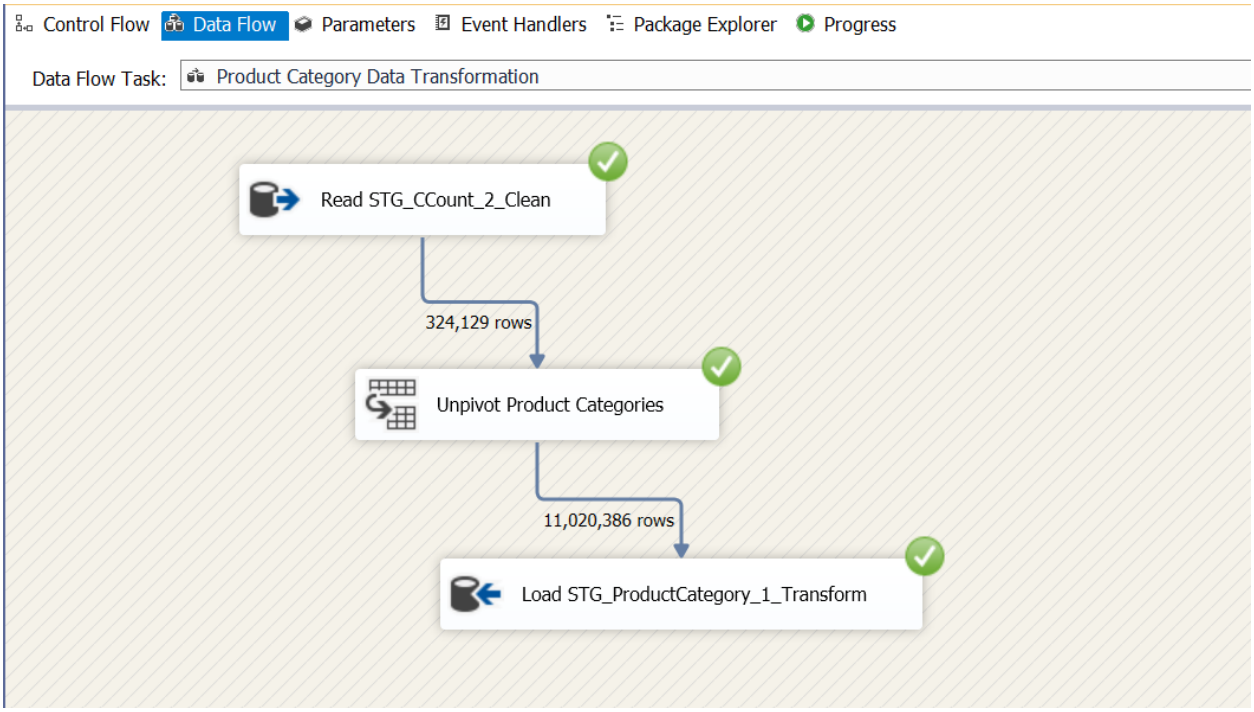
Input Column	Destination Column
TimeKey	TimeKey
Month	Month
Year	Year
Date	Date

OK Cancel Help

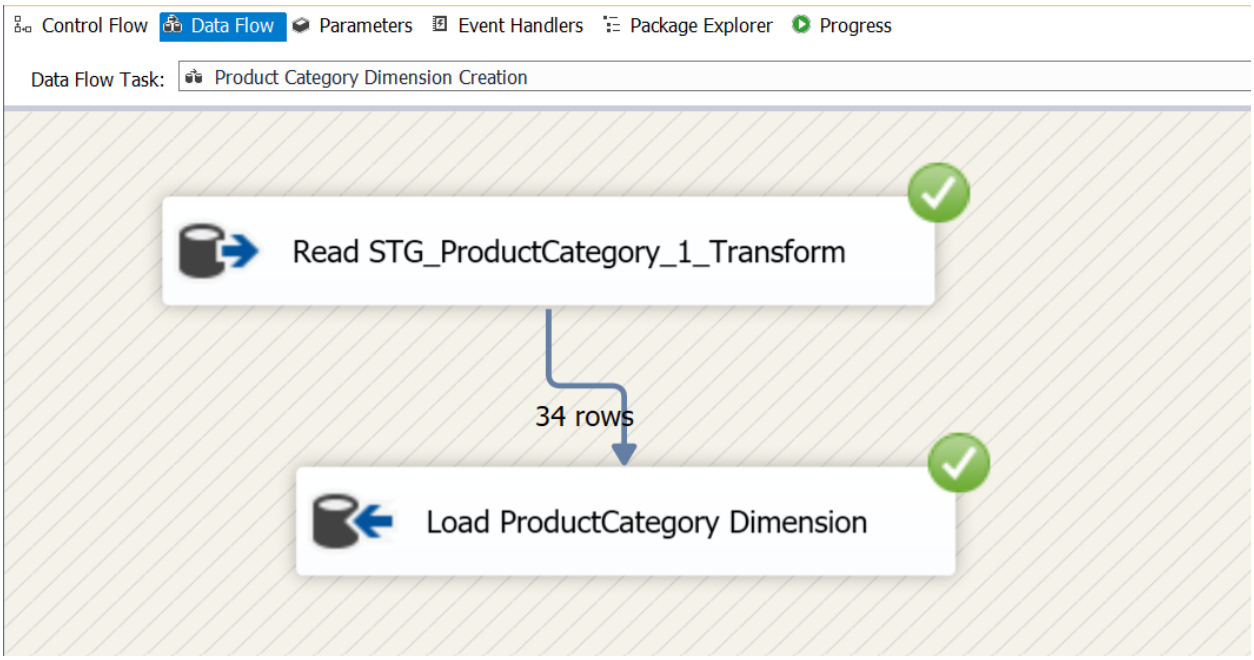
ETL for ProductCategory Dimension:



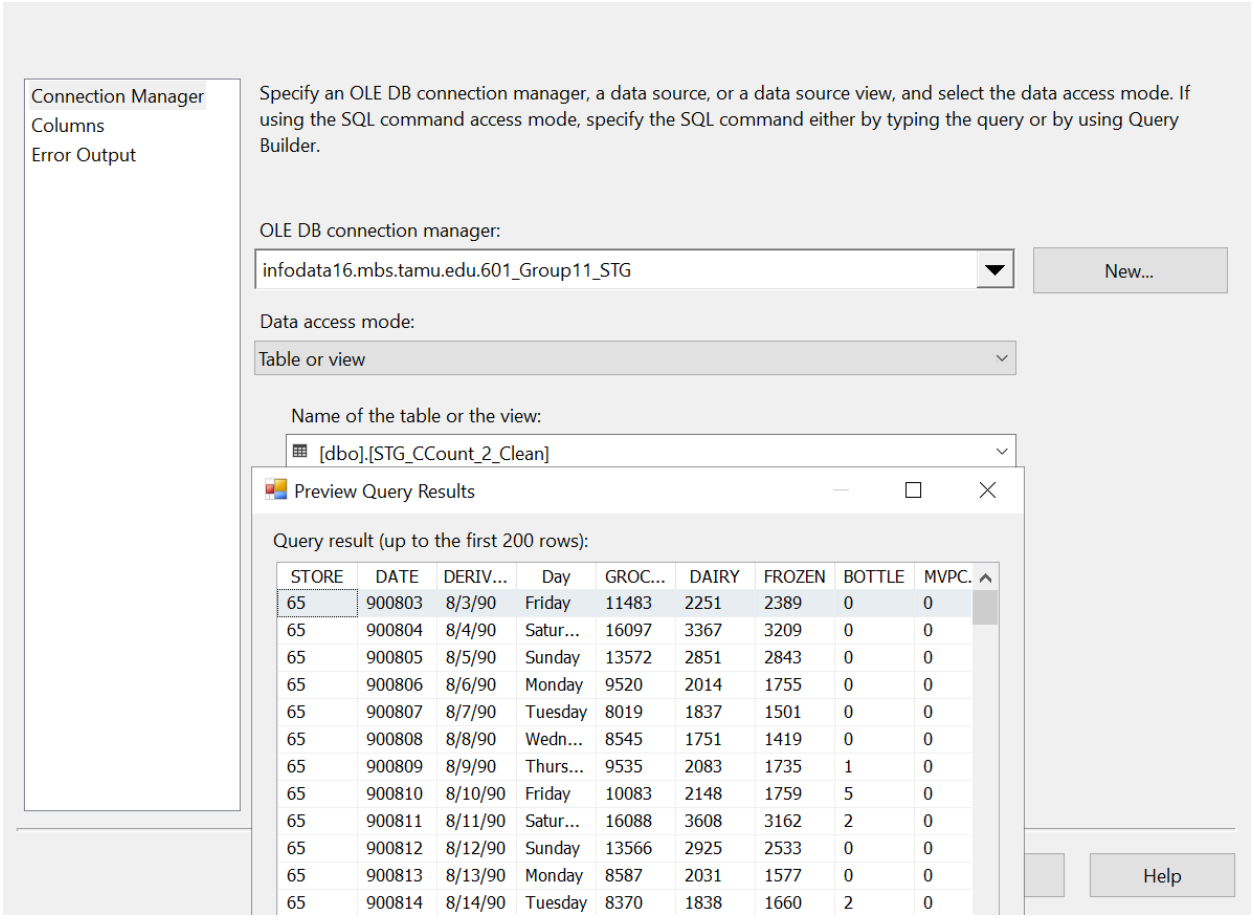
Control Flow for ProductCategory Dimension



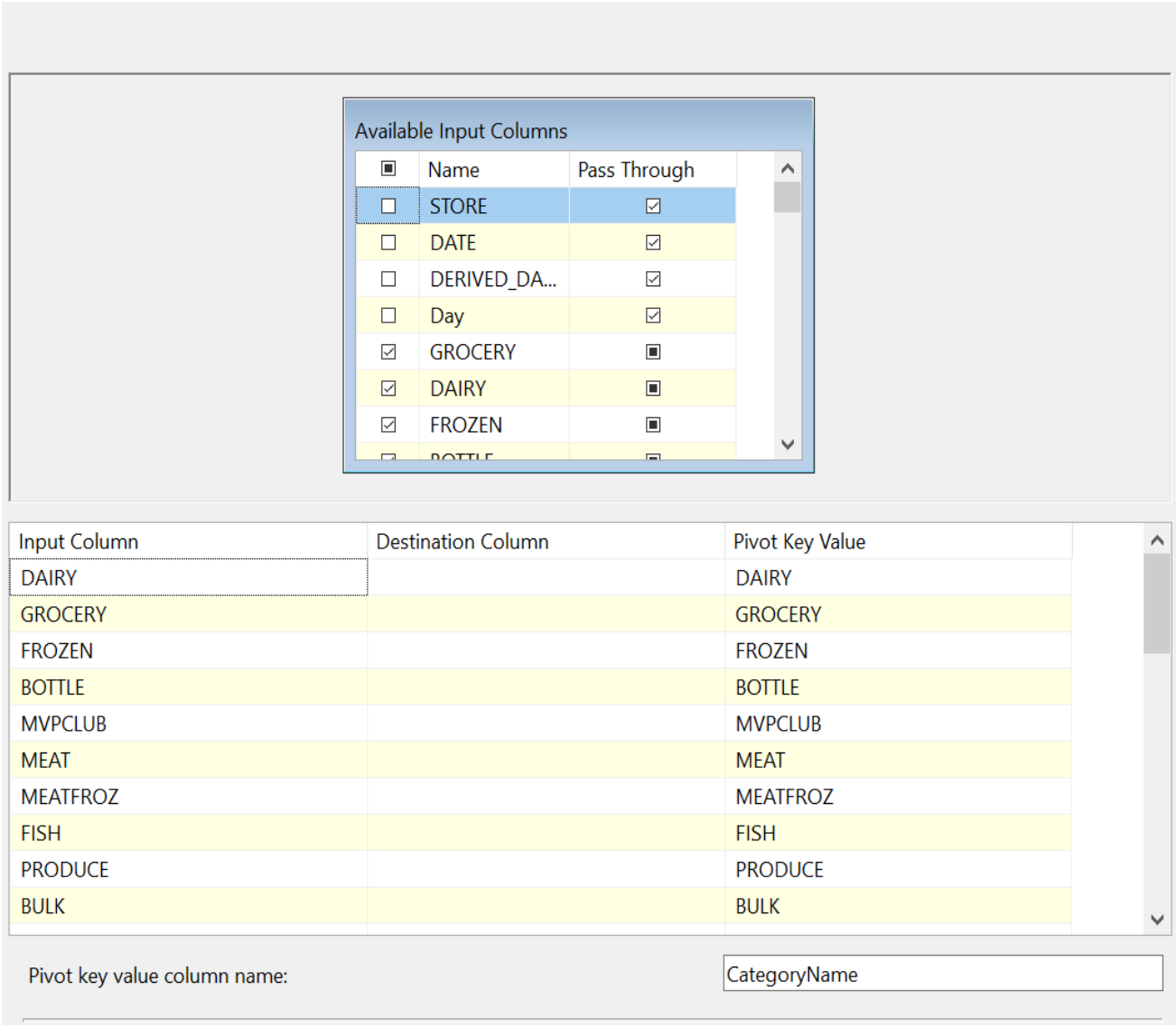
Data Flow for Product Category Data Transformation

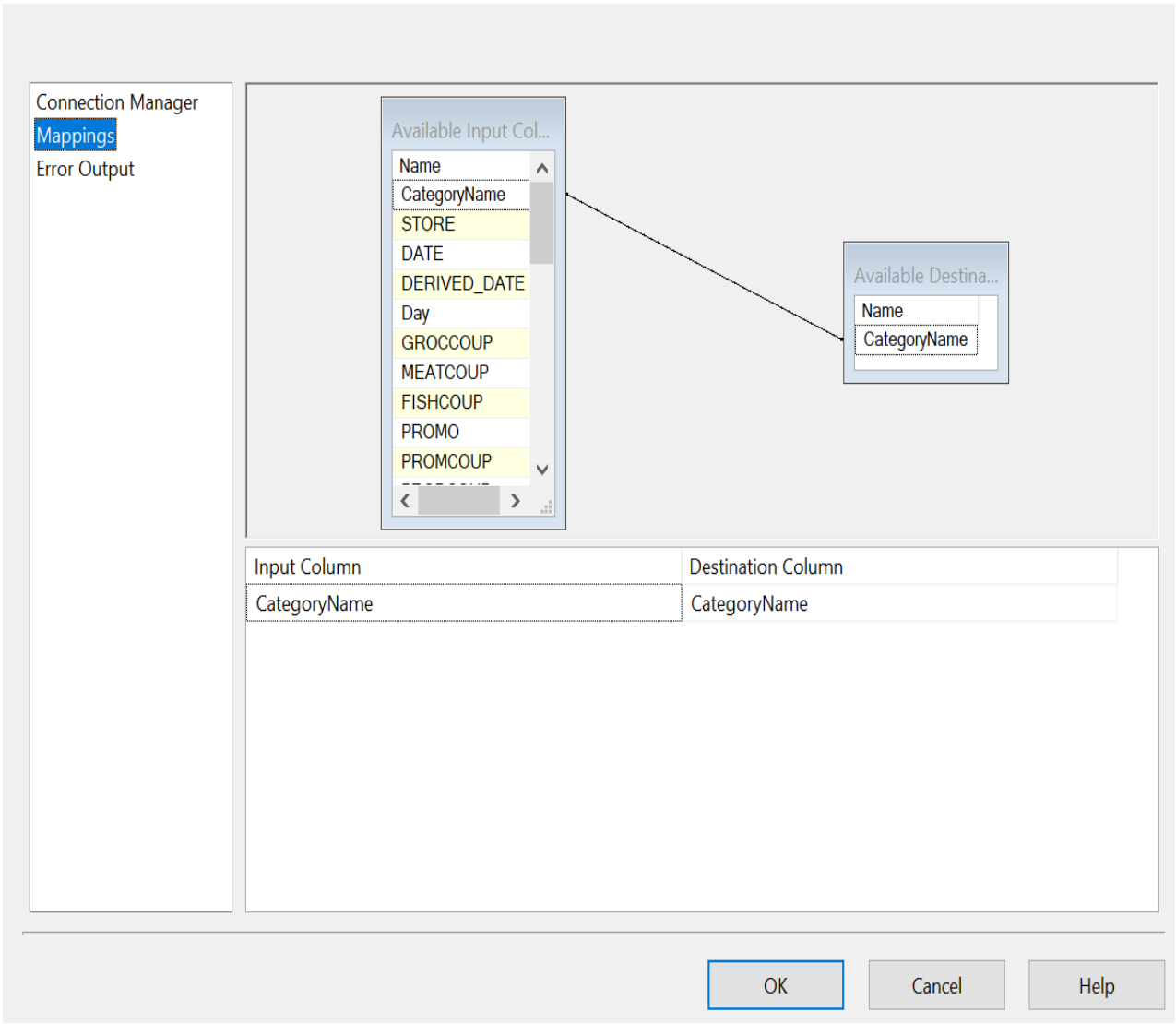


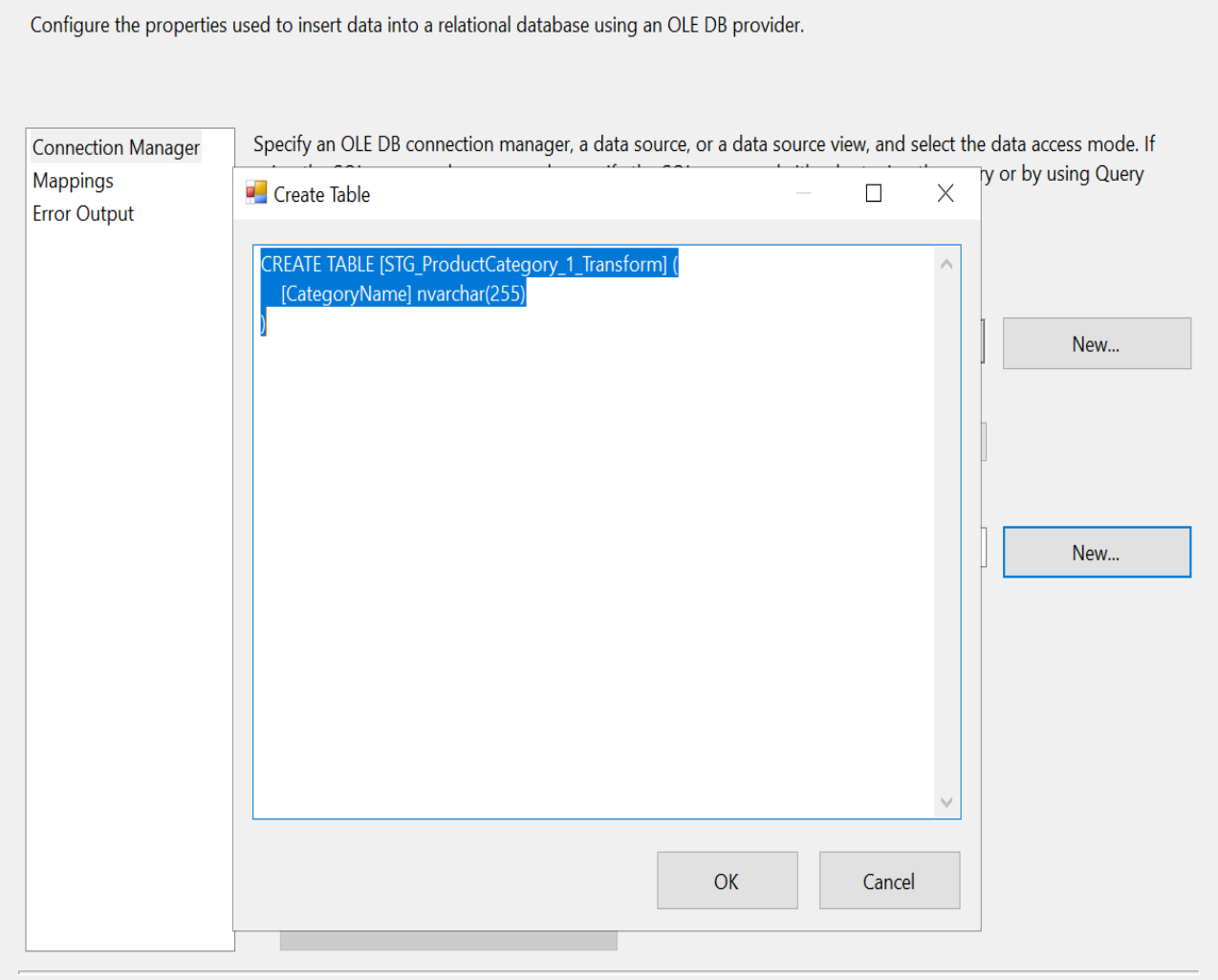
Data Flow for Product Category Dimension Creation

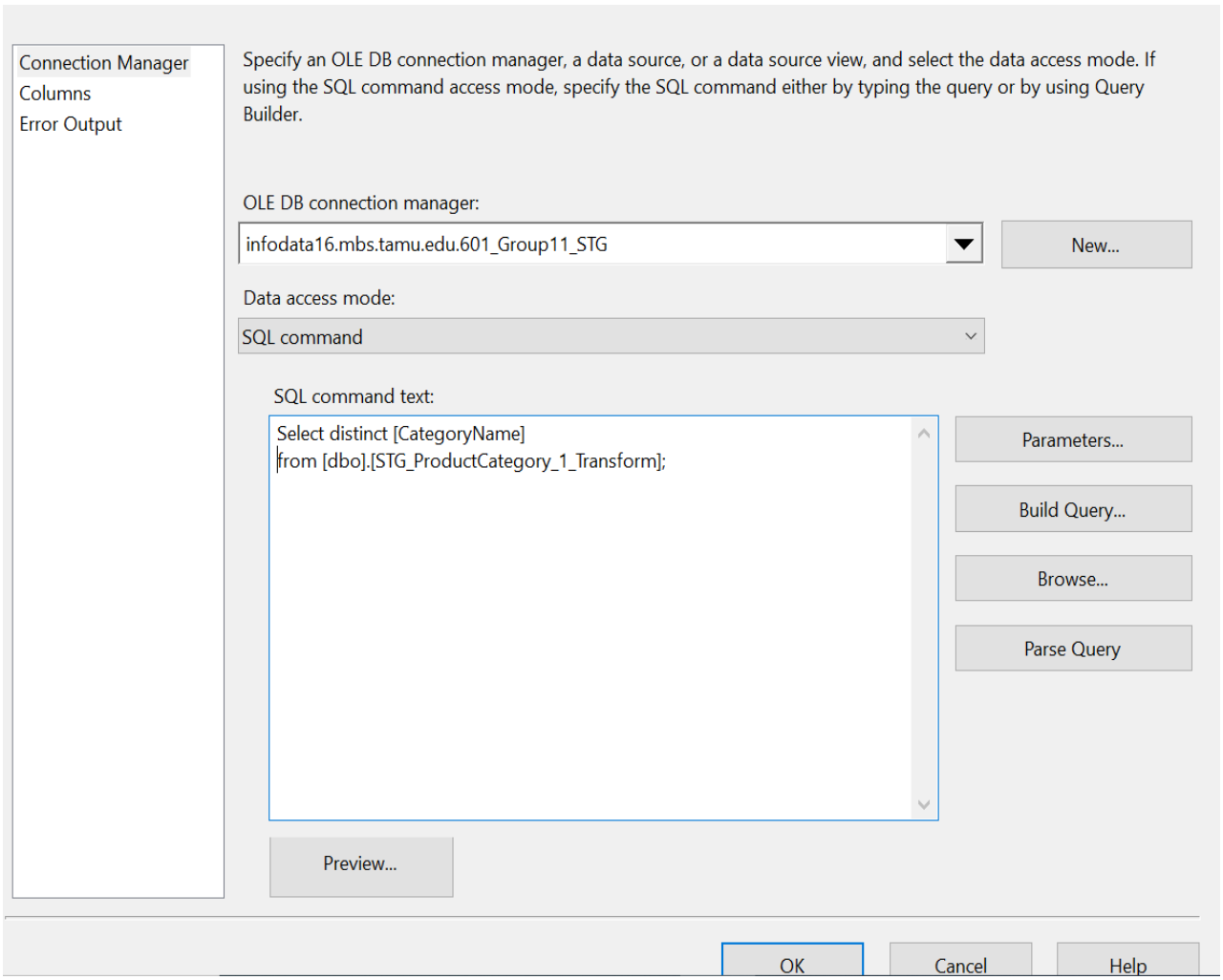


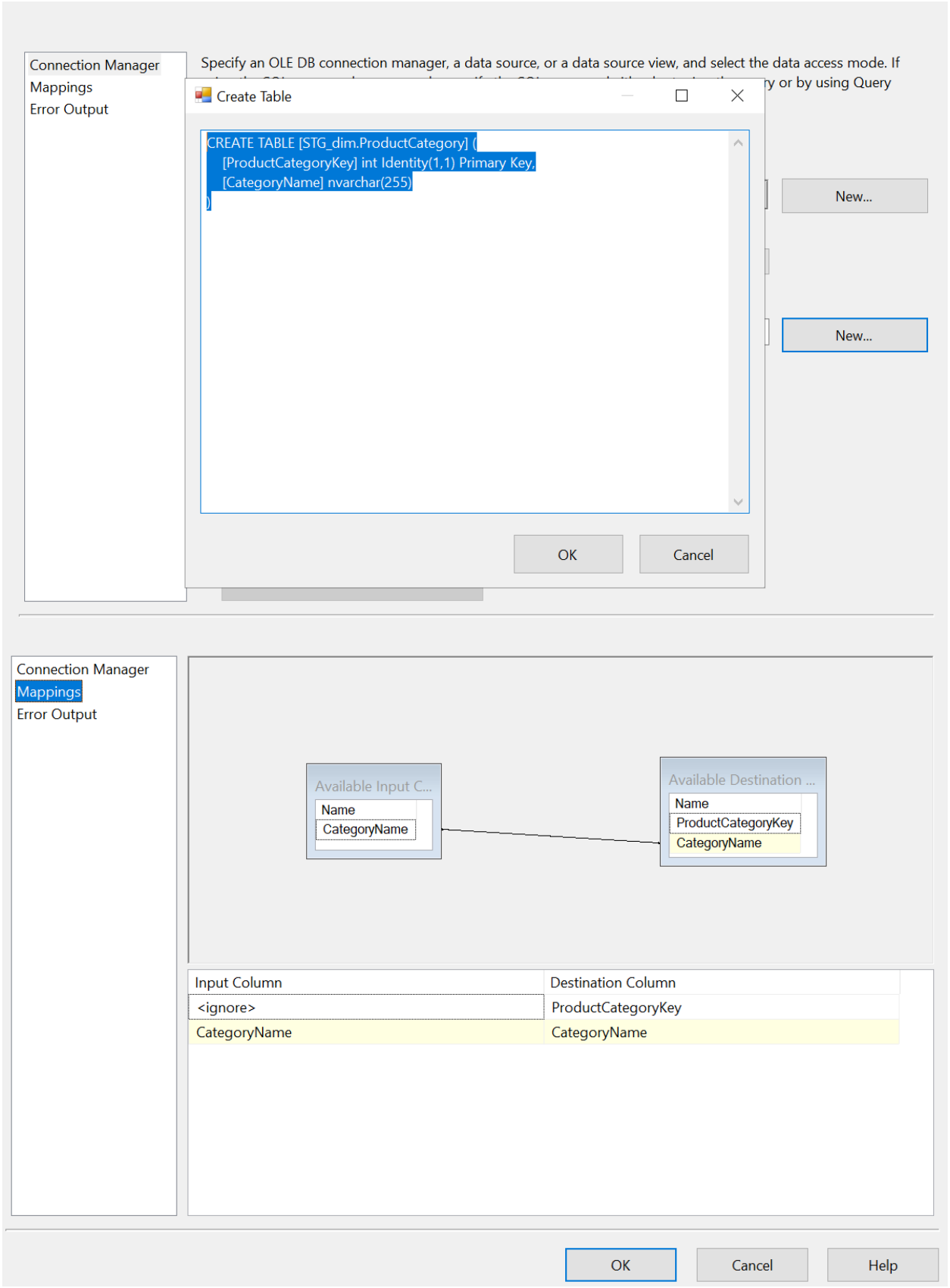
STORE	DATE	DERIV...	Day	GROC...	DAIRY	FROZEN	BOTTLE	MVPC.
65	900803	8/3/90	Friday	11483	2251	2389	0	0
65	900804	8/4/90	Satur...	16097	3367	3209	0	0
65	900805	8/5/90	Sunday	13572	2851	2843	0	0
65	900806	8/6/90	Monday	9520	2014	1755	0	0
65	900807	8/7/90	Tuesday	8019	1837	1501	0	0
65	900808	8/8/90	Wedn...	8545	1751	1419	0	0
65	900809	8/9/90	Thurs...	9535	2083	1735	1	0
65	900810	8/10/90	Friday	10083	2148	1759	5	0
65	900811	8/11/90	Satur...	16088	3608	3162	2	0
65	900812	8/12/90	Sunday	13566	2925	2533	0	0
65	900813	8/13/90	Monday	8587	2031	1577	0	0
65	900814	8/14/90	Tuesday	8370	1838	1660	2	0

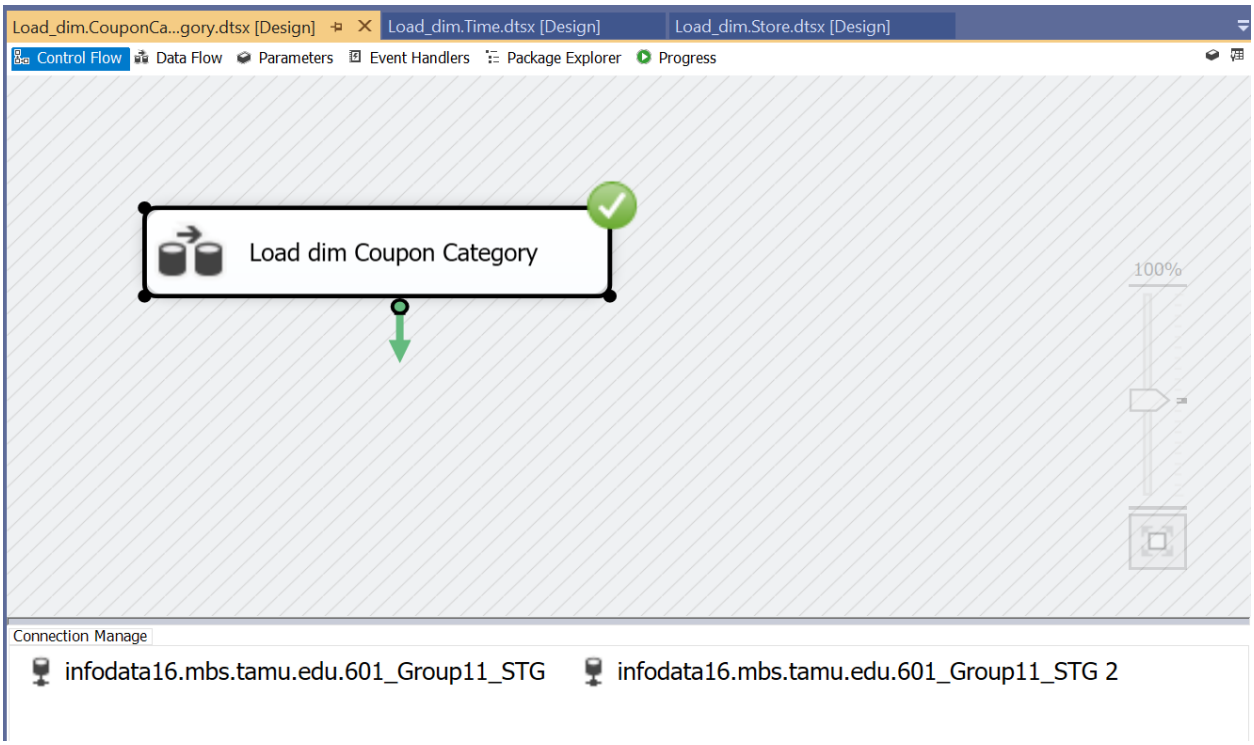

Available Input Columns
Name
Pass Through
STORE
DATE
DERIVED_DA...
Day
GROCERY
DAIRY
FROZEN
Input Column
Destination Column
Pivot Key Value
DAIRY
GROCERY
FROZEN
BOTTLE
MVPCLUB
MEAT
MEATFROZ
FISH
PRODUCE
BULK
Pivot key value column name:
CategoryName


Connection Manager
Mappings
Error Output
Available Input Col...
Name
CategoryName
STORE
DATE
DERIVED_DATE
Day
GROCCOUP
MEATCOUP
FISHCOUP
PROMO
PROMCOUP
Available Destina...
Name
CategoryName
Input Column
Destination Column
CategoryName
CategoryName
OK
Cancel
Help

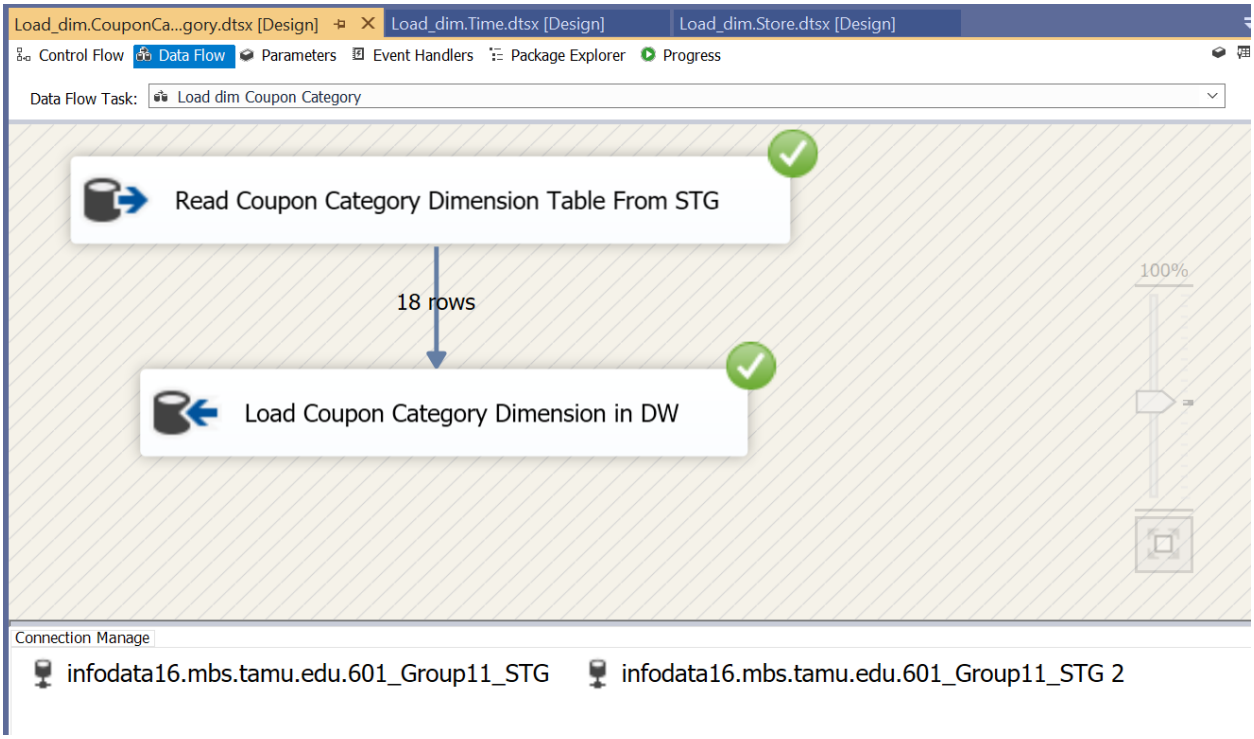

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
y or by using Query
Create Table
CREATE TABLE [STG_ProductCategory_1_Transform] (
[CategoryName] nvarchar(255)
)
New...
New...
OK
Cancel


Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
New...
Data access mode:
SQL command
SQL command text:
Select distinct [CategoryName]
from [dbo].[STG_ProductCategory_1_Transform];
Parameters...
Build Query...
Browse...
Parse Query
Preview...
OK
Cancel
Help

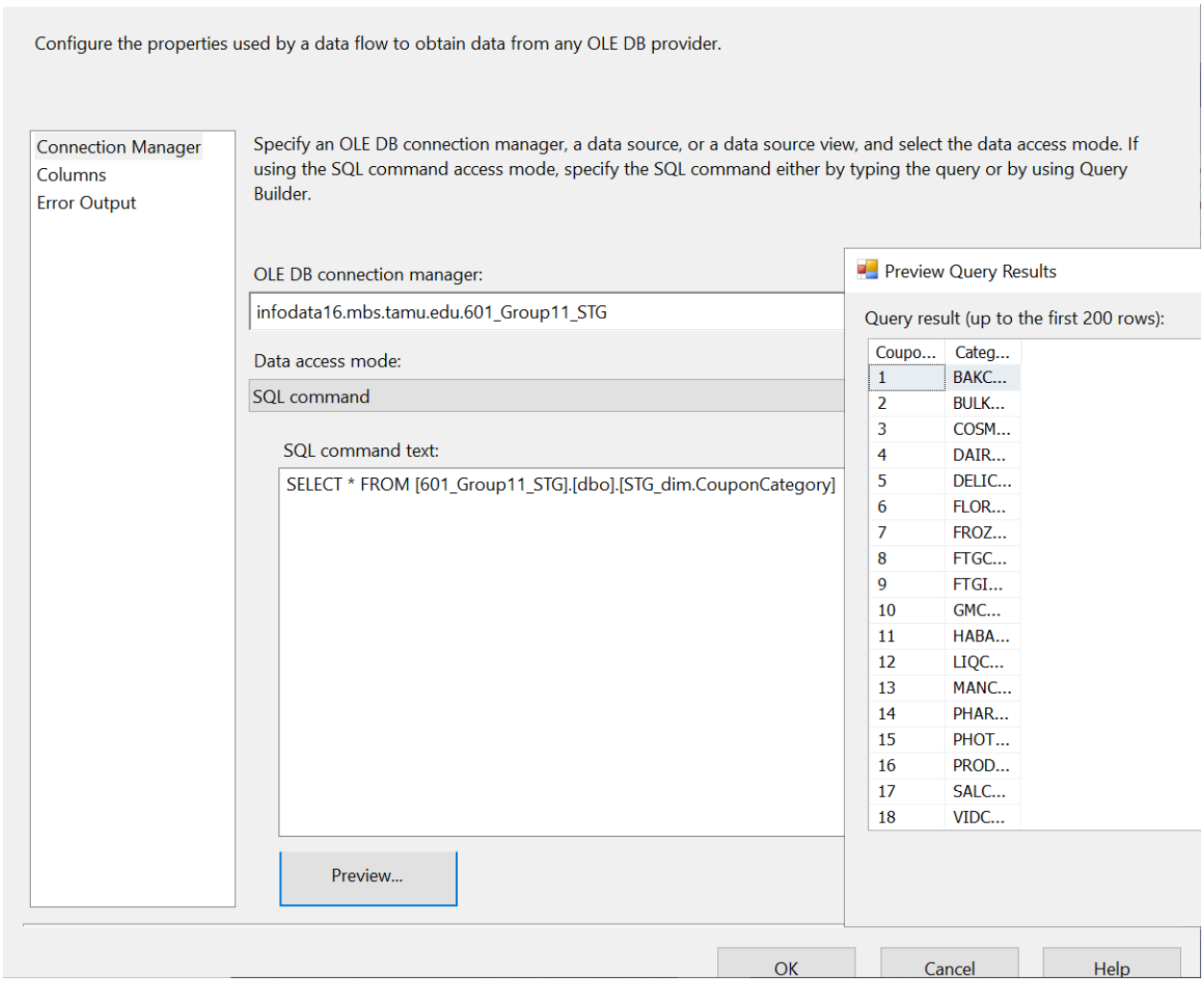

Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
y or by using Query
Create Table
CREATE TABLE [STG_dim.ProductCategory] (
[ProductCategoryKey] int Identity(1,1) Primary Key,
[CategoryName] nvarchar(255)
)
OK
Cancel
New...
New...
Connection Manager
Mappings
Error Output
Available Input C...
Name
CategoryName
Available Destination ...
Name
ProductCategoryKey
CategoryName
Input Column
Destination Column
<ignore>
ProductCategoryKey
CategoryName
CategoryName
OK
Cancel
Help

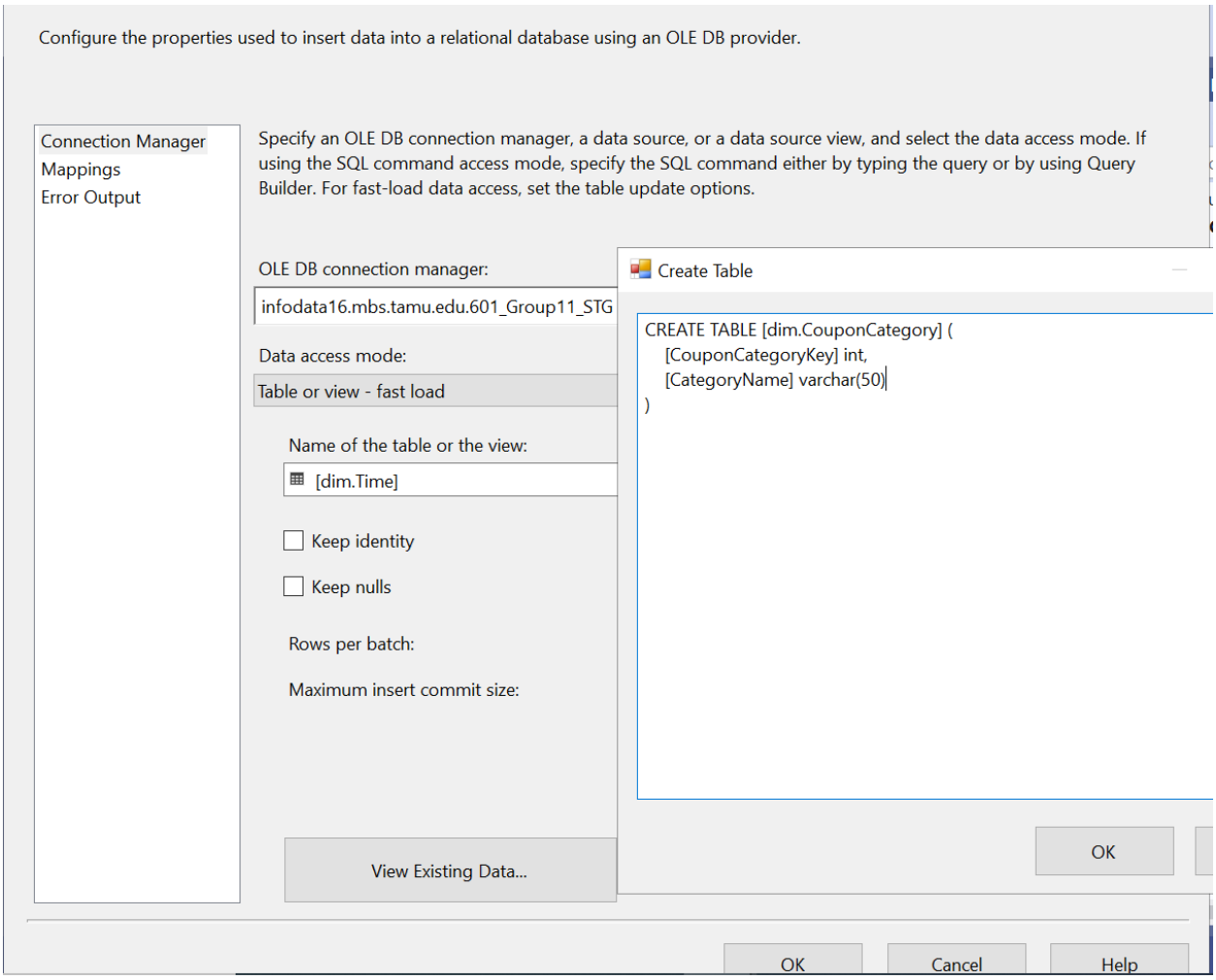


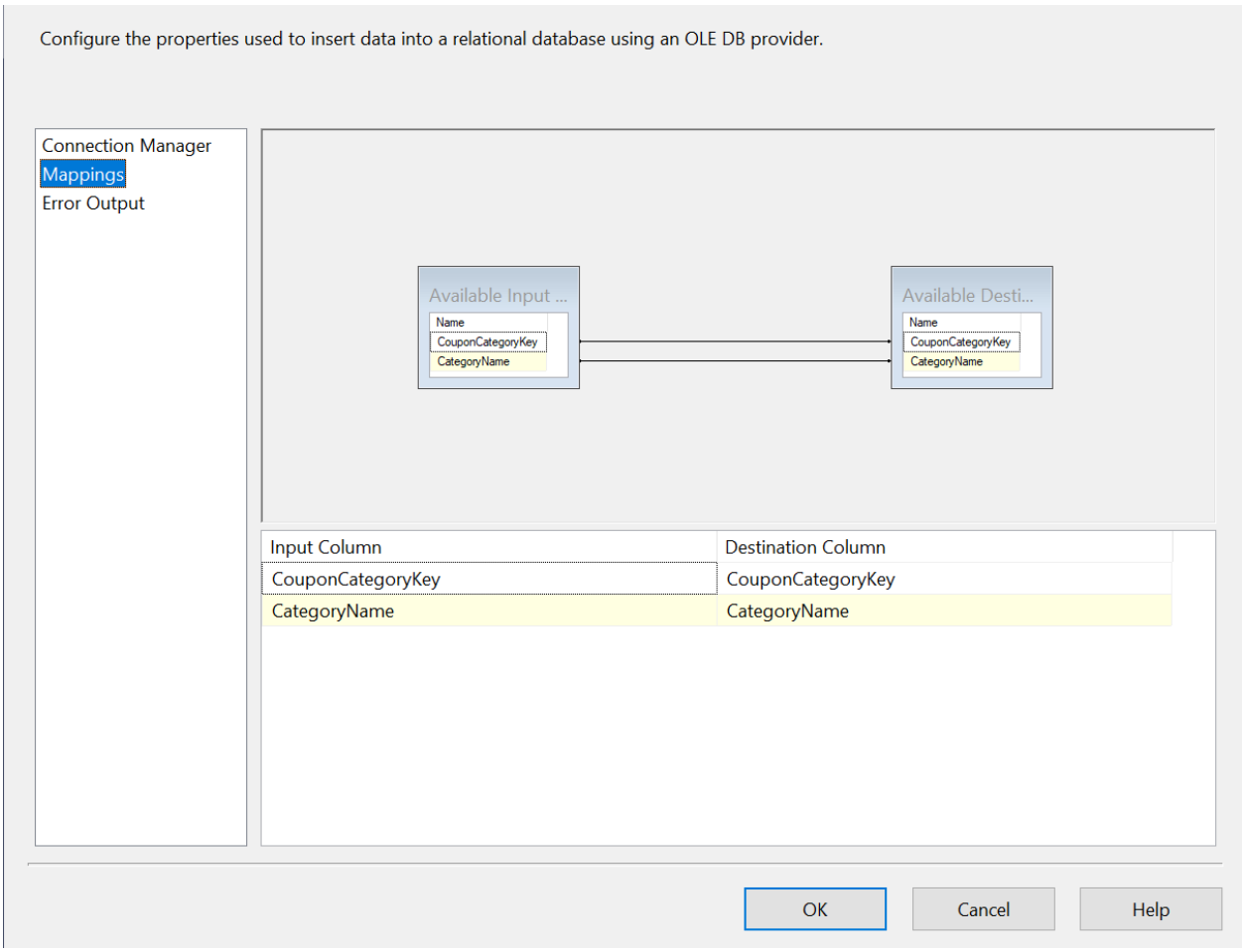
Control Flow for loading Coupon Category Dimension



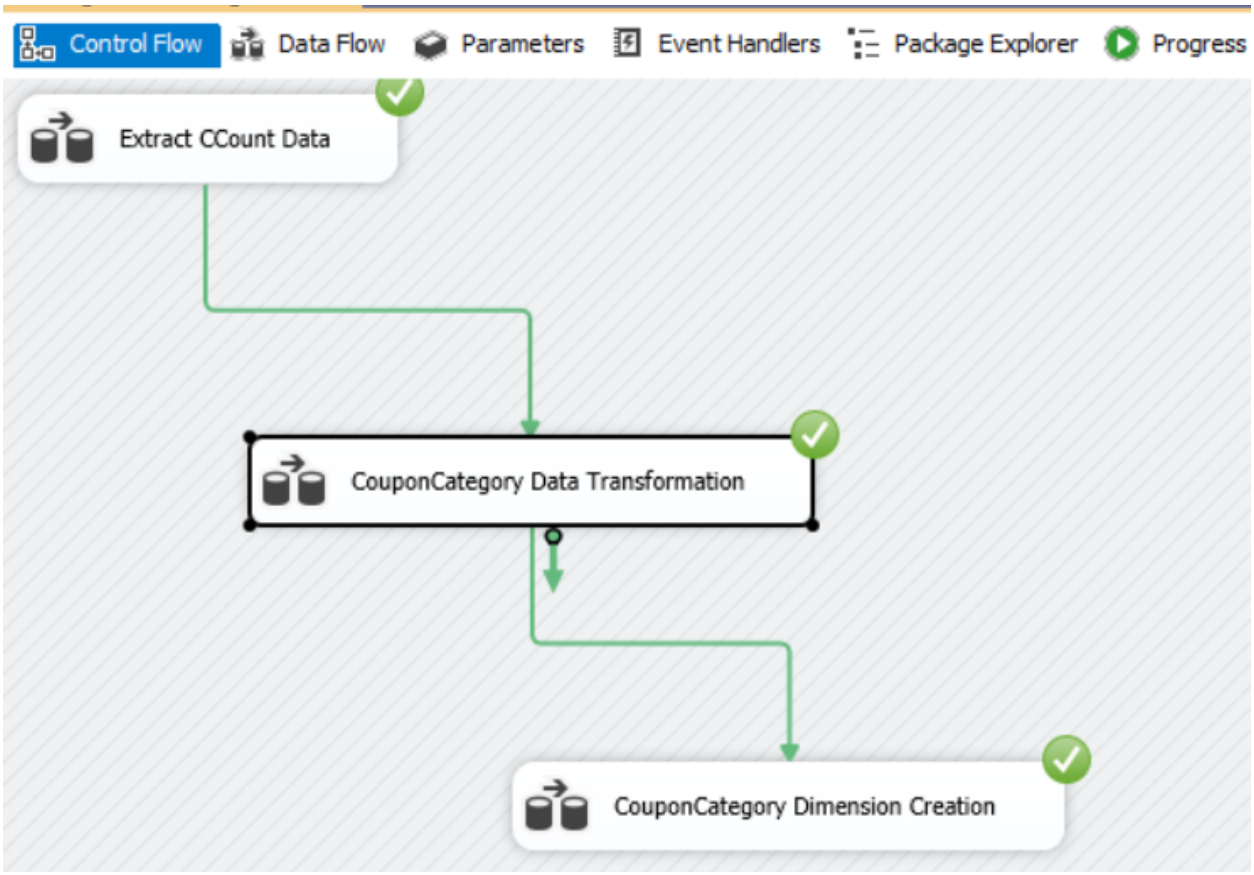
Data Flow for load dim Coupon Category


Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
Data access mode:
SQL command
SQL command text:
SELECT * FROM [601_Group11_STG].[dbo].[STG_dim.CouponCategory]
Preview...
Preview Query Results
Query result (up to the first 200 rows):
Coupo... Categ...
1 BAKC...
2 BULK...
3 COSM...
4 DAIR...
5 DELIC...
6 FLOR...
7 FROZ...
8 FTGC...
9 FTGI...
10 GMC...
11 HABA...
12 LIQC...
13 MANC...
14 PHAR...
15 PHOT...
16 PROD...
17 SALC...
18 VIDC...
OK
Cancel
Help

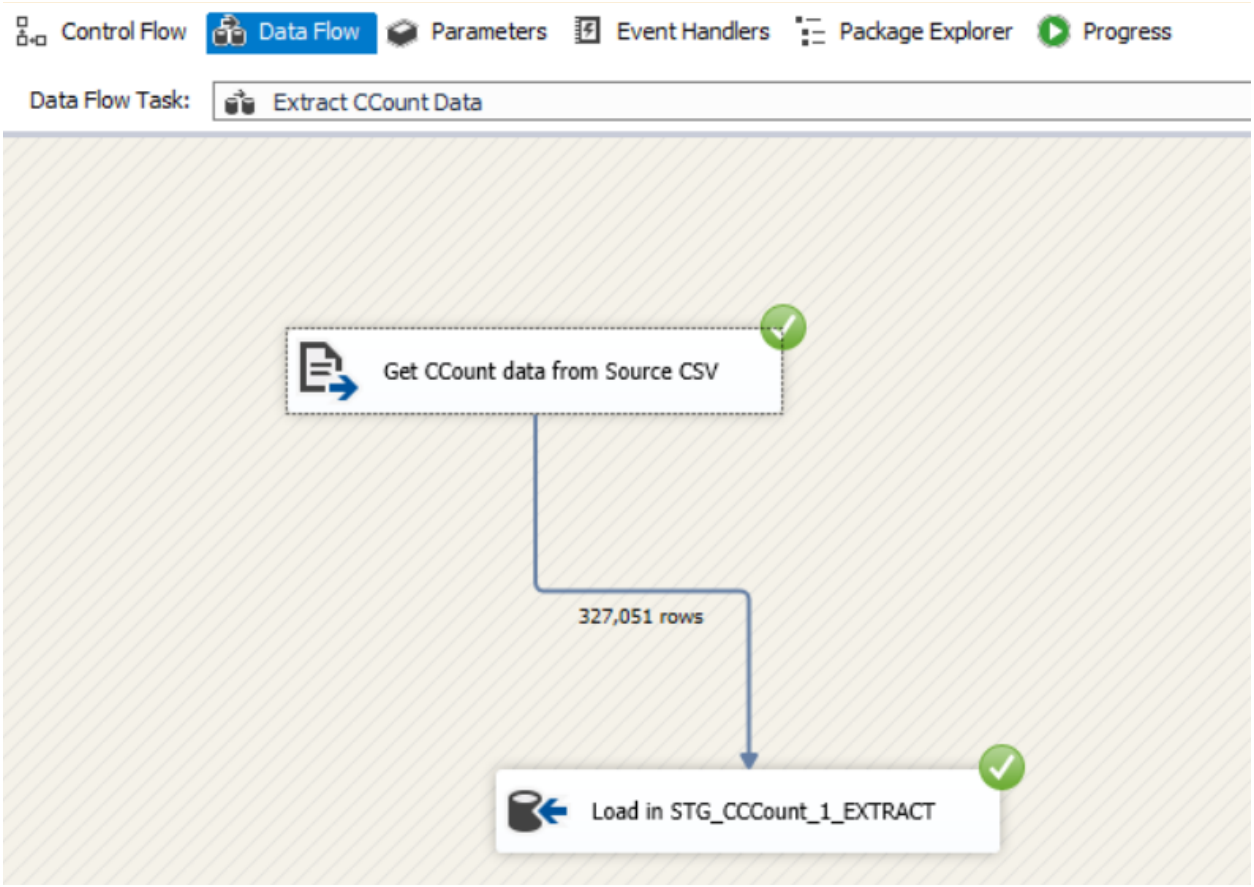

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
Data access mode:
Table or view - fast load
Name of the table or the view:
[dim.Time]
Keep identity
Keep nulls
Rows per batch:
Maximum insert commit size:
View Existing Data...
Create Table
CREATE TABLE [dim.CouponCategory] (
[CouponCategoryKey] int,
[CategoryName] varchar(50)
)
OK
OK
Cancel
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Available Input ...
Name
CouponCategoryKey
CategoryName
Available Desti...
Name
CouponCategoryKey
CategoryName
Input Column
Destination Column
CouponCategoryKey
CouponCategoryKey
CategoryName
CategoryName
OK
Cancel
Help

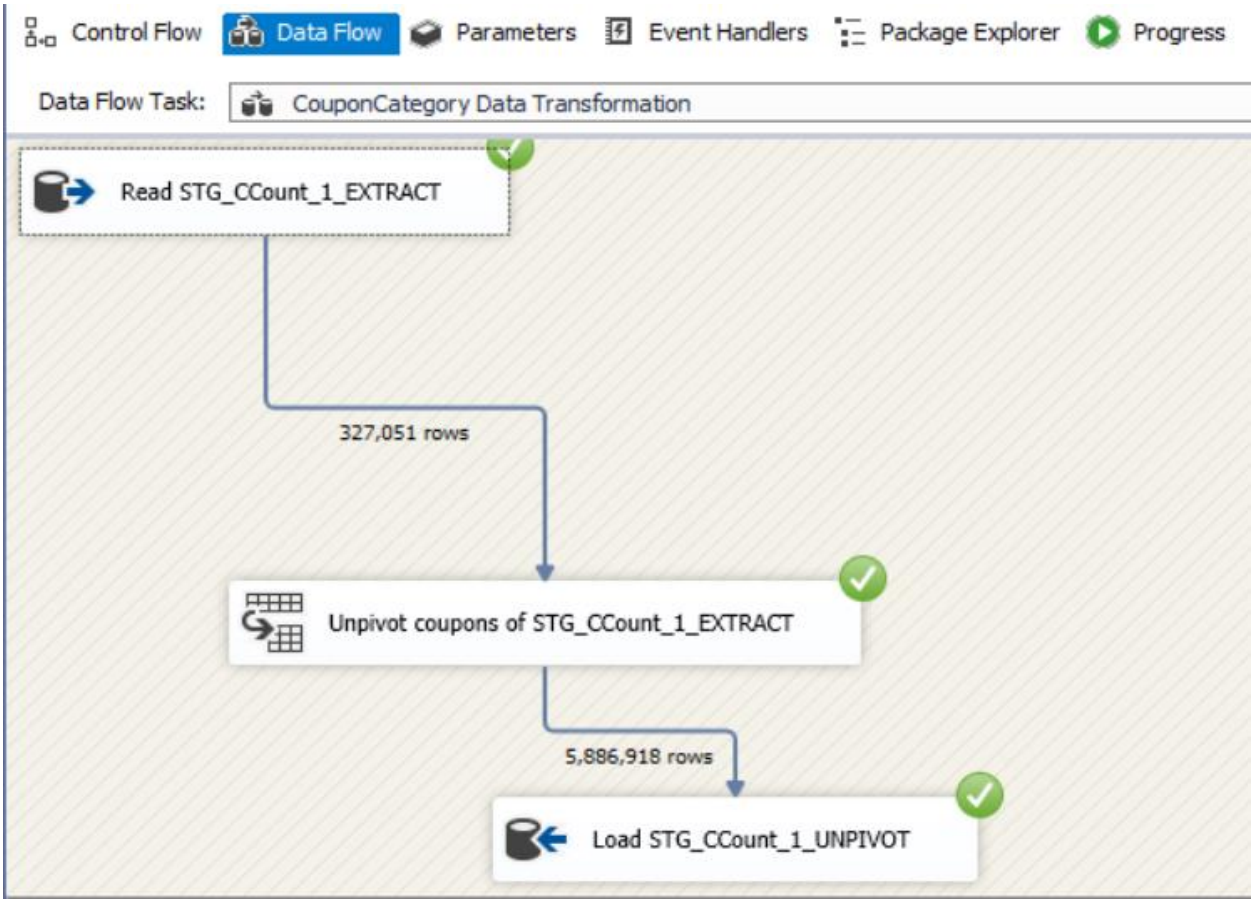
ETL for CouponCategory Dimension:



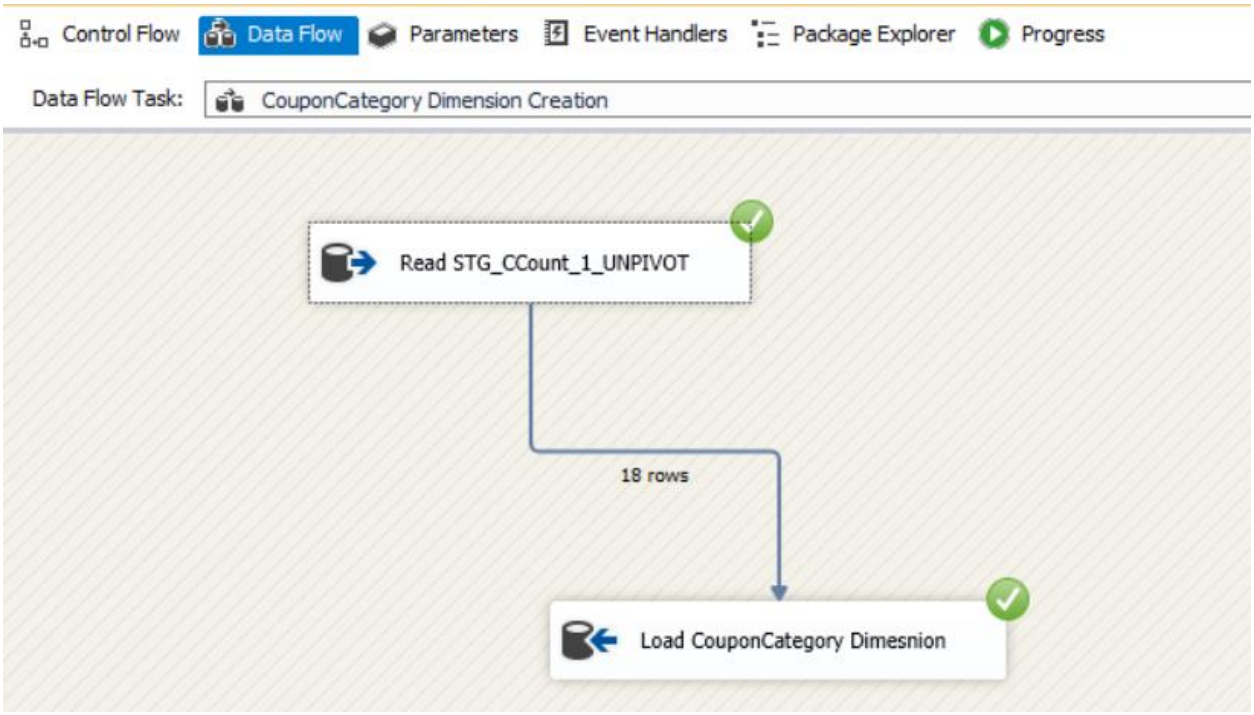
Control Flow for Coupon Category Dimension Creation



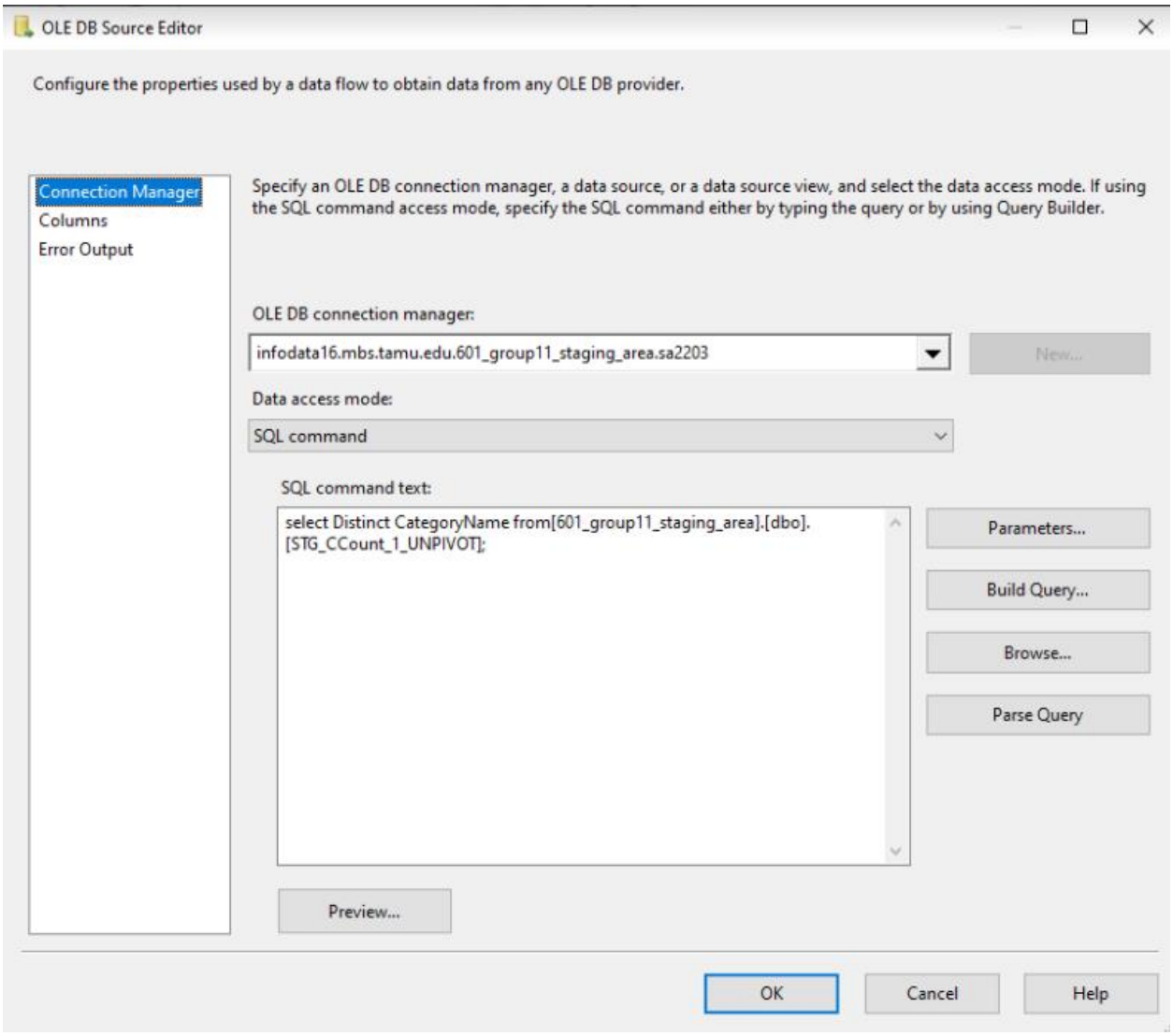
Data Flow for Extract CCount Data

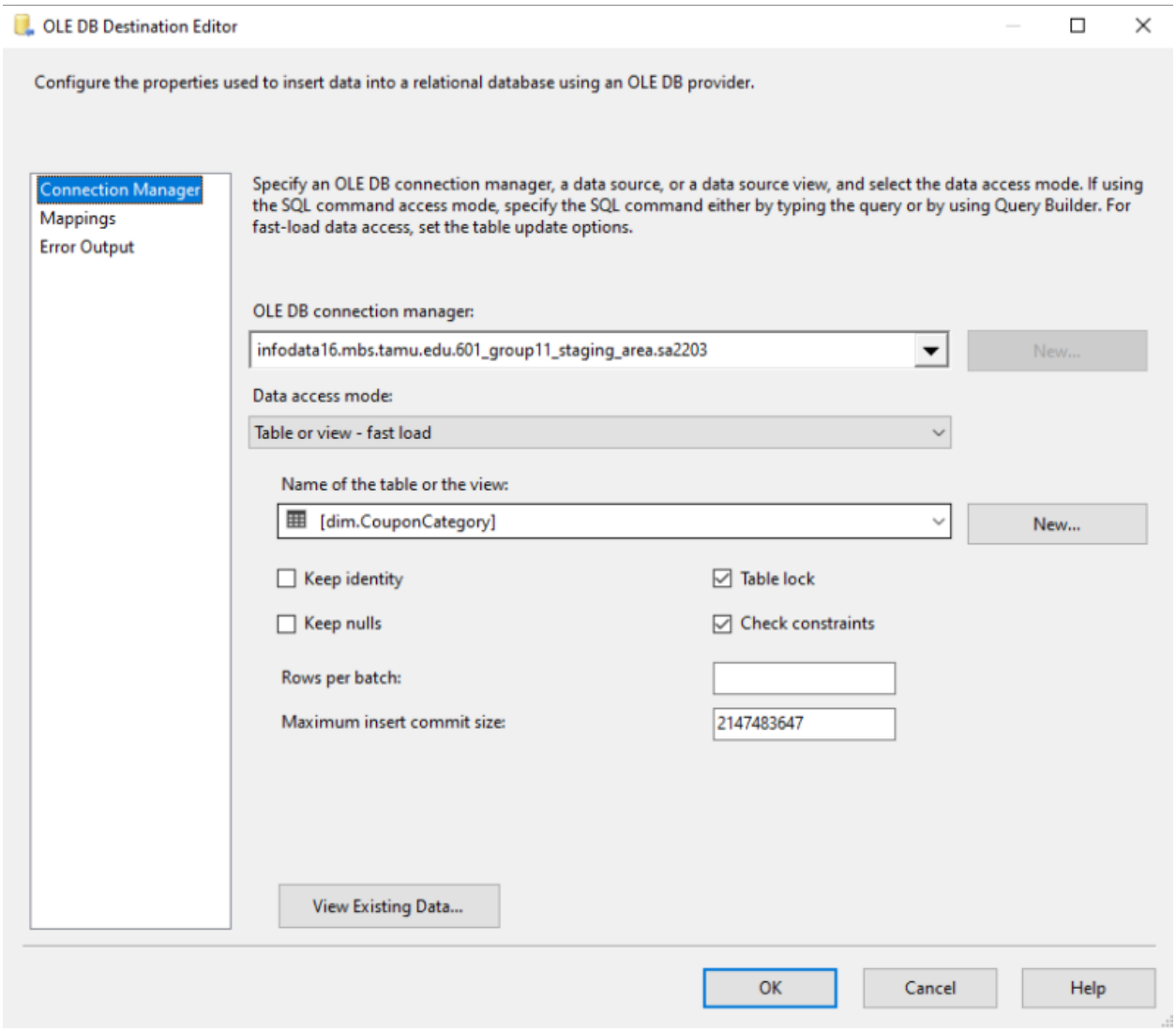


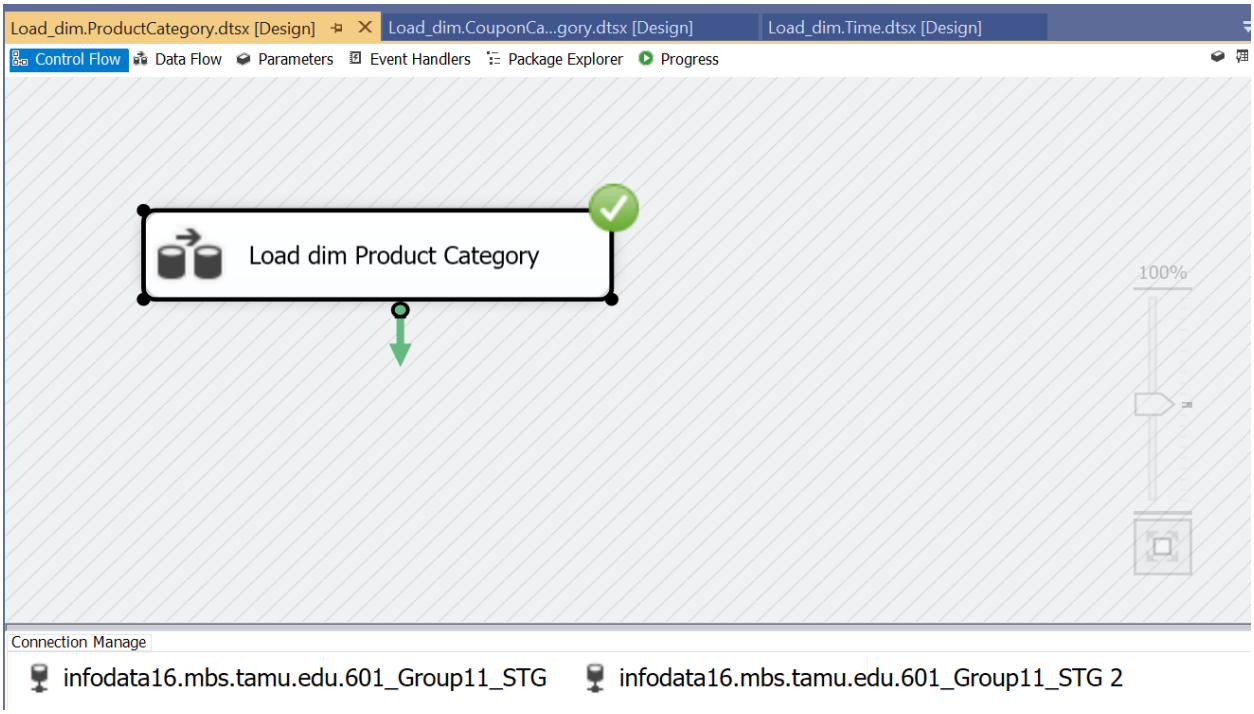
Data Flow for CouponCategory Data Transformation



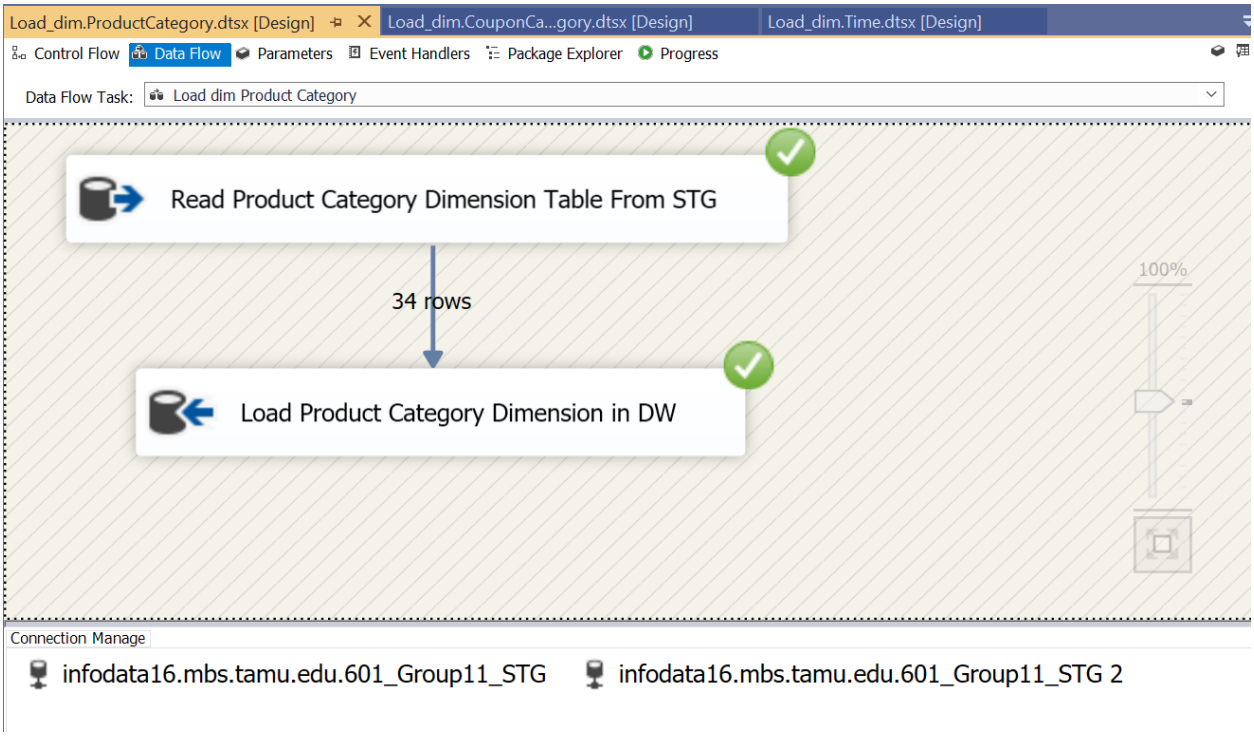
Data Flow for CouponCategory Dimension Creation


OLE DB Source Editor
Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_group11_staging_area.sa2203
New...
Data access mode:
SQL command
SQL command text:
select Distinct CategoryName from[601_group11_staging_area].[dbo].[STG_CCount_1_UNPIVOT];
Parameters...
Build Query...
Browse...
Parse Query
Preview...
OK
Cancel
Help

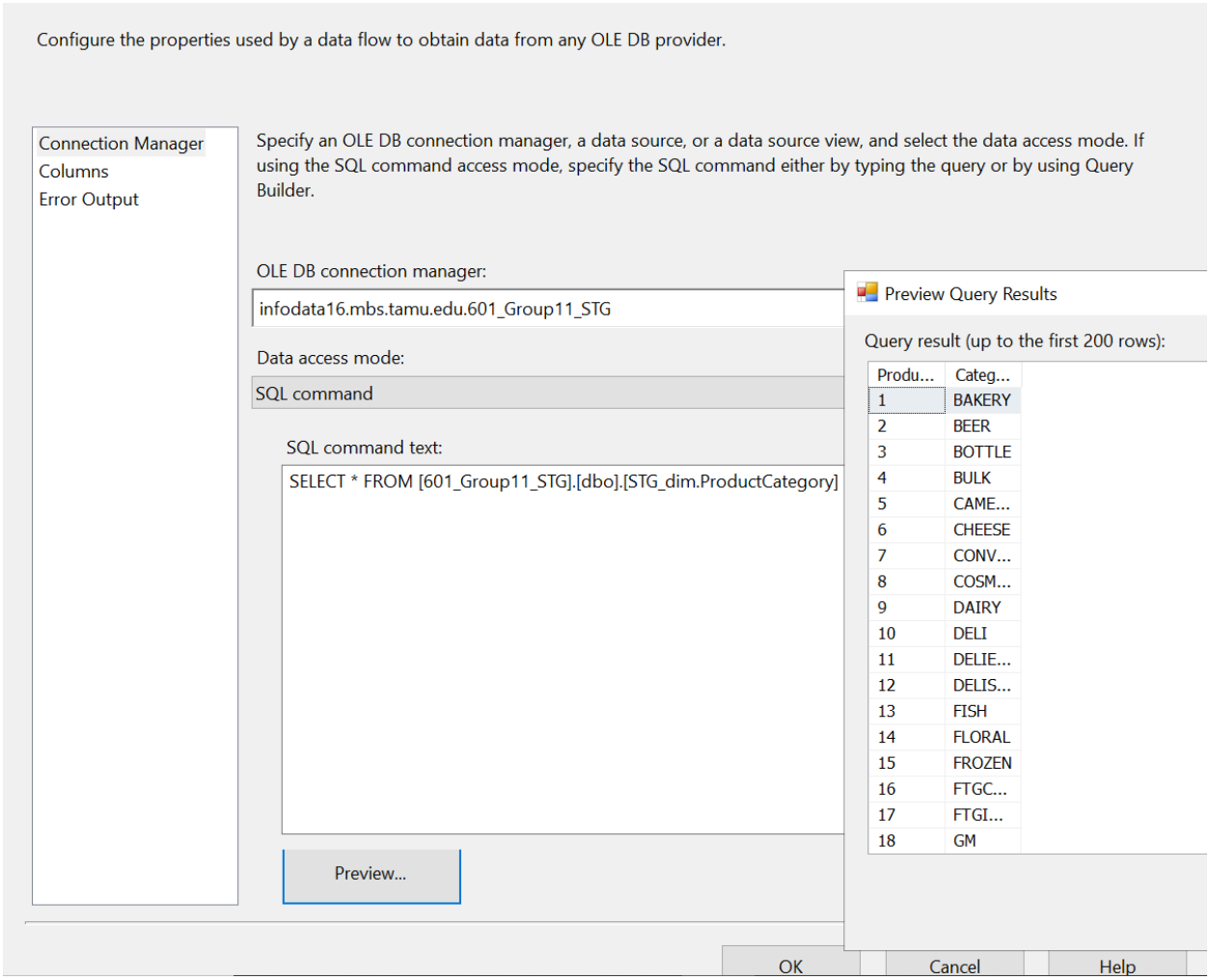

OLE DB Destination Editor
Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_group11_staging_area.sa2203
New...
Data access mode:
Table or view - fast load
Name of the table or the view:
[dim.CouponCategory]
New...
Keep identity
Table lock
Keep nulls
Check constraints
Rows per batch:
Maximum insert commit size:
2147483647
View Existing Data...
OK
Cancel
Help

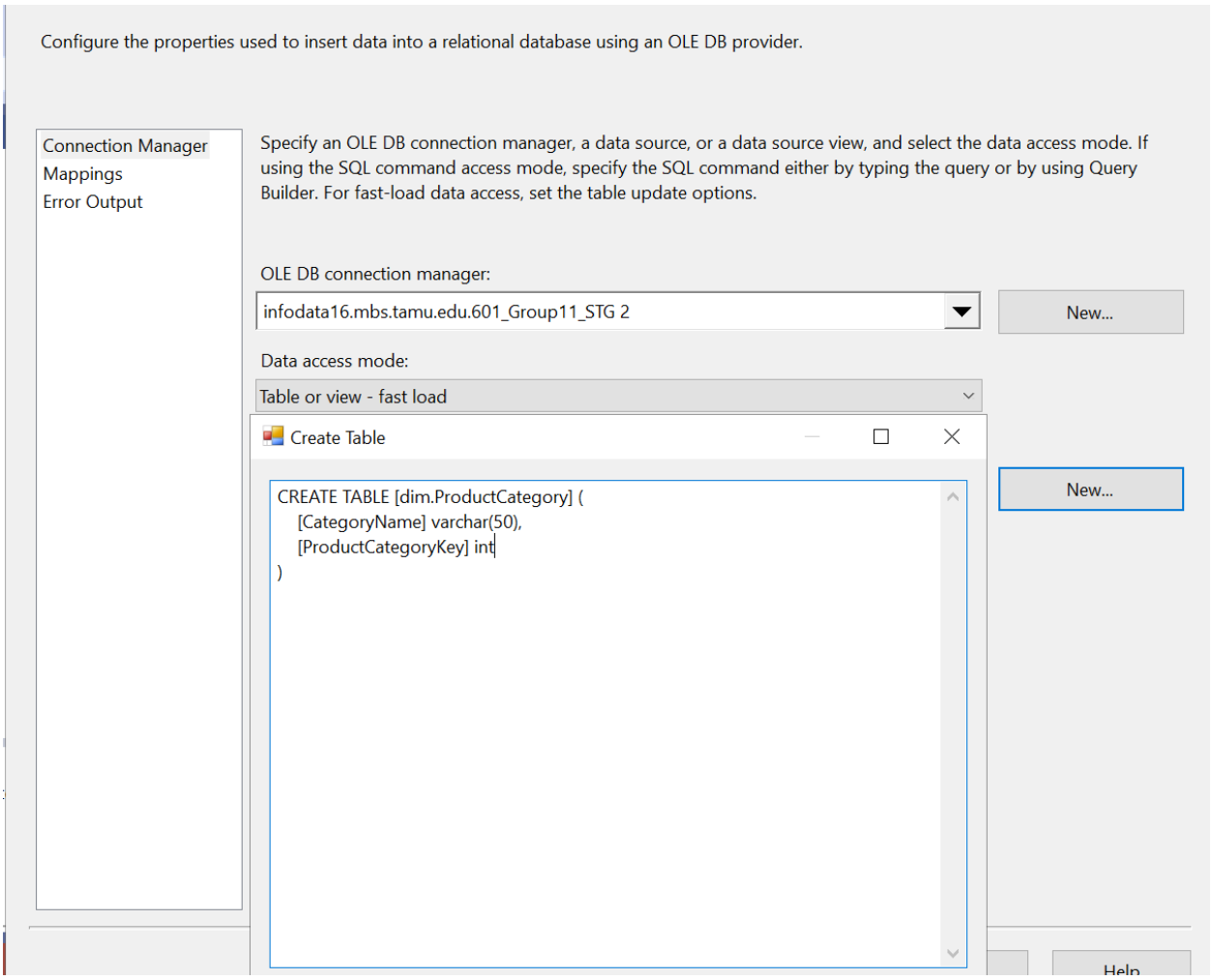


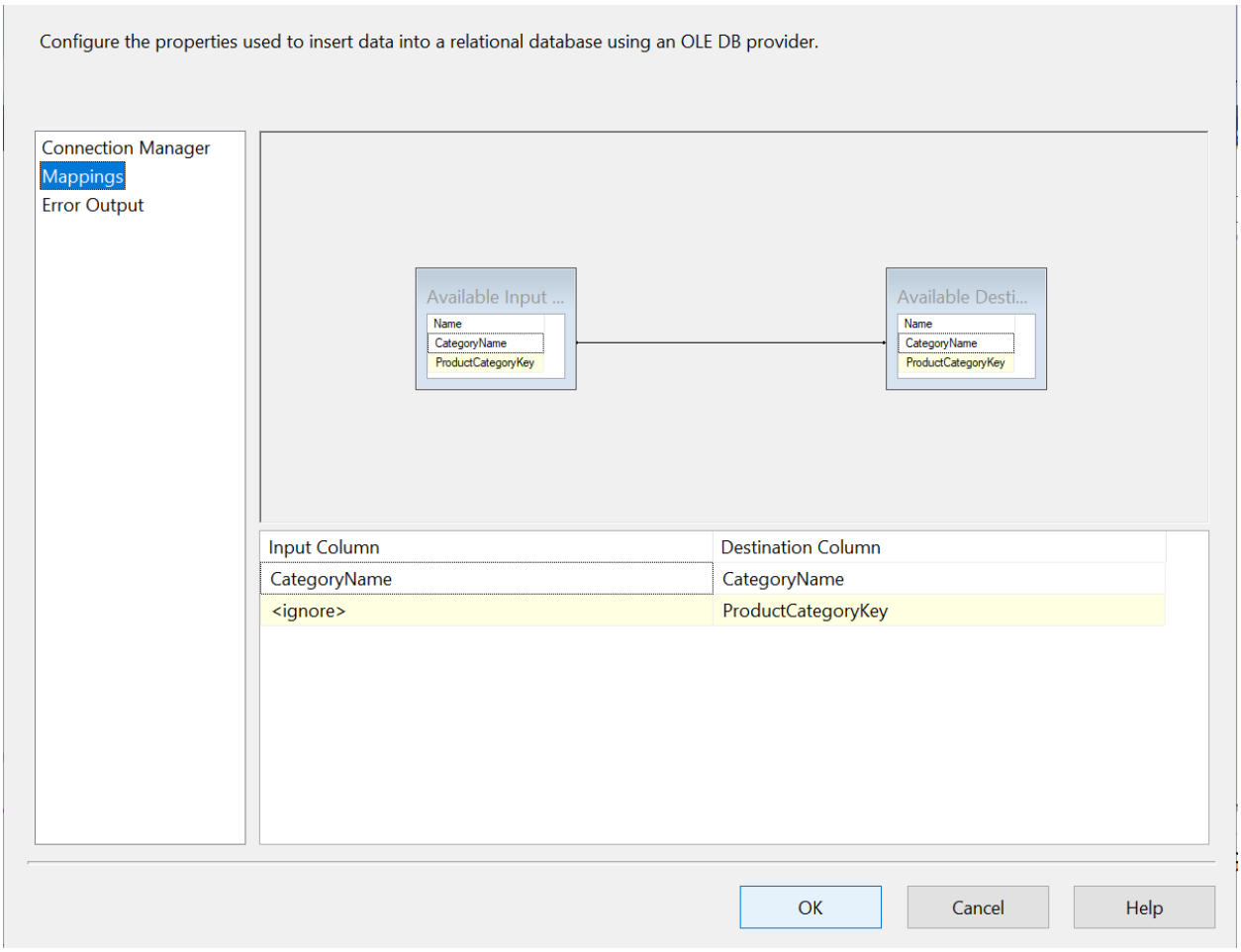
Control flow for loading dim Product Category



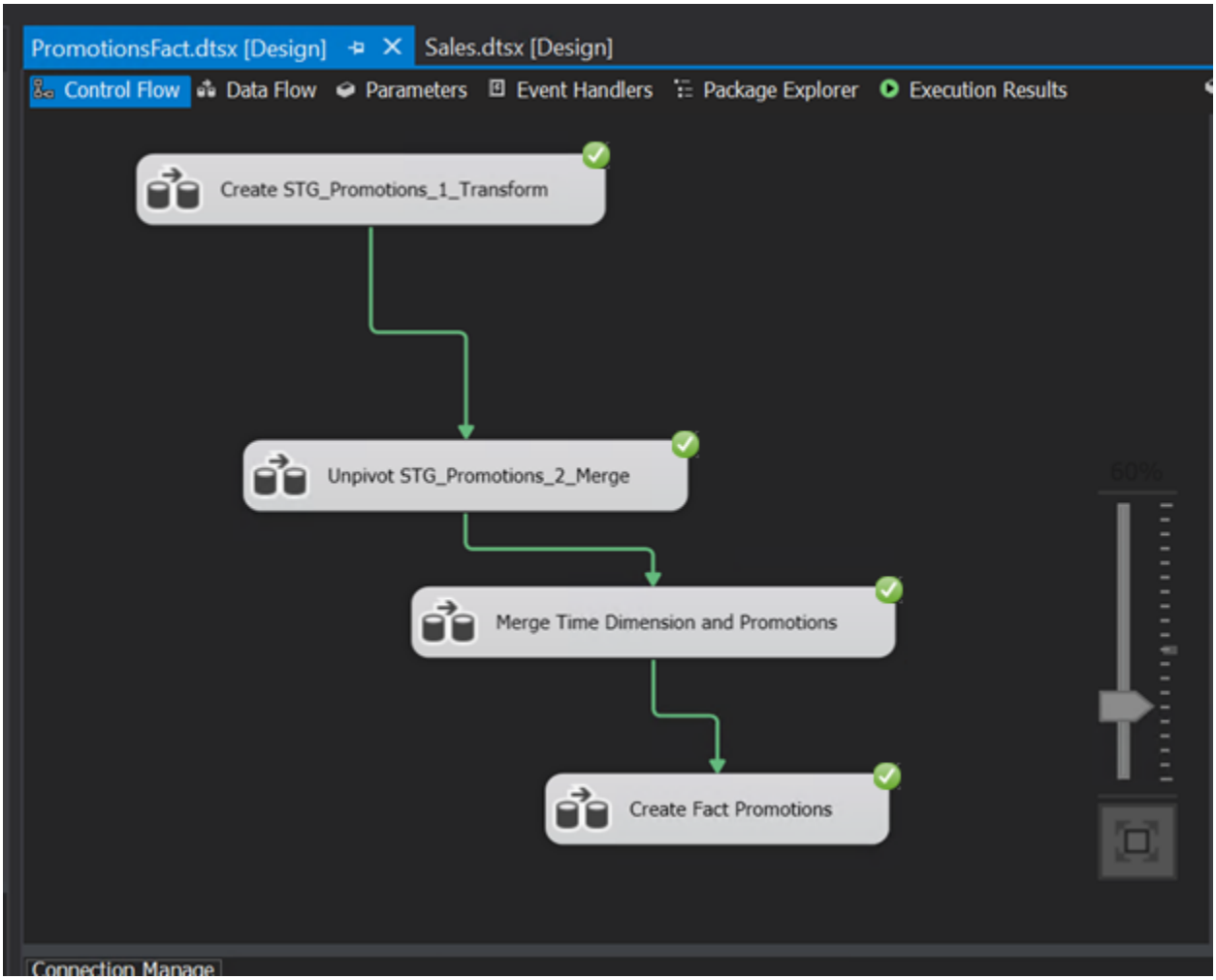
Data Flow to load dim Product Category


Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
Data access mode:
SQL command
SQL command text:
SELECT * FROM [601_Group11_STG].[dbo].[STG_dim.ProductCategory]
Preview...
Preview Query Results
Query result (up to the first 200 rows):
Produ... Categ...
1 BAKERY
2 BEER
3 BOTTLE
4 BULK
5 CAME...
6 CHEESE
7 CONV...
8 COSM...
9 DAIRY
10 DELI
11 DELIE...
12 DELIS...
13 FISH
14 FLORAL
15 FROZEN
16 FTGC...
17 FTGI...
18 GM
OK
Cancel
Help

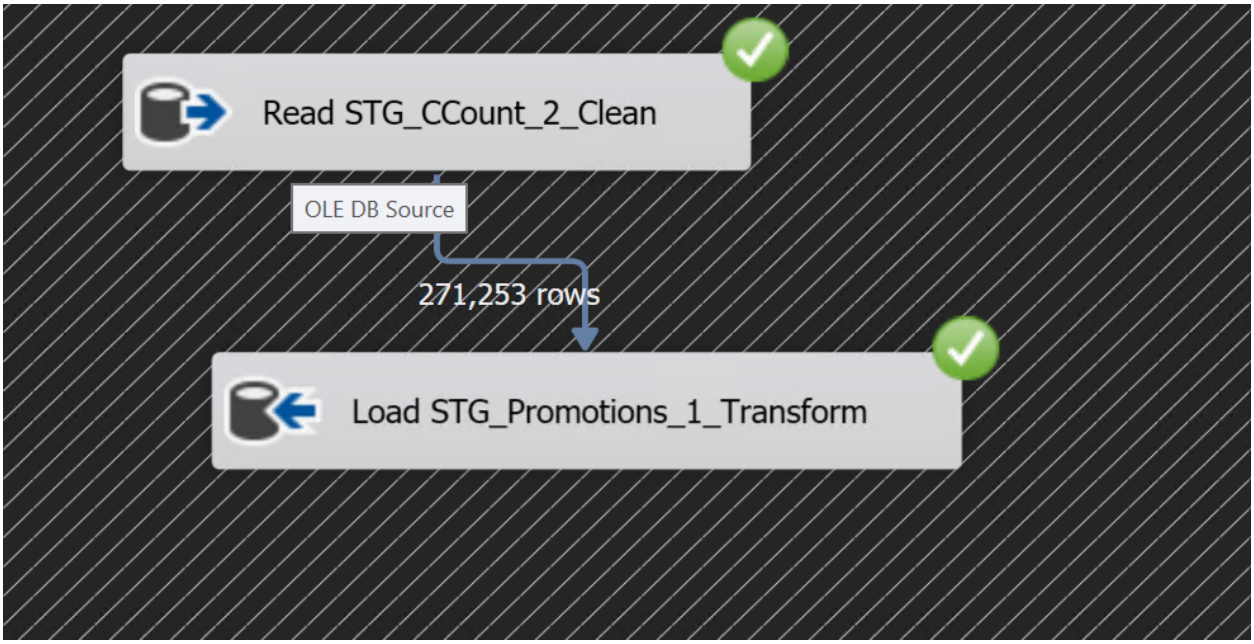

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG 2
New...
Data access mode:
Table or view - fast load
Create Table
CREATE TABLE [dim.ProductCategory] (
[CategoryName] varchar(50),
[ProductCategoryKey] int
)
New...
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Available Input ...
Name
CategoryName
ProductCategoryKey
Available Desti...
Name
CategoryName
ProductCategoryKey
Input Column
Destination Column
CategoryName
CategoryName
<ignore>
ProductCategoryKey
OK
Cancel
Help

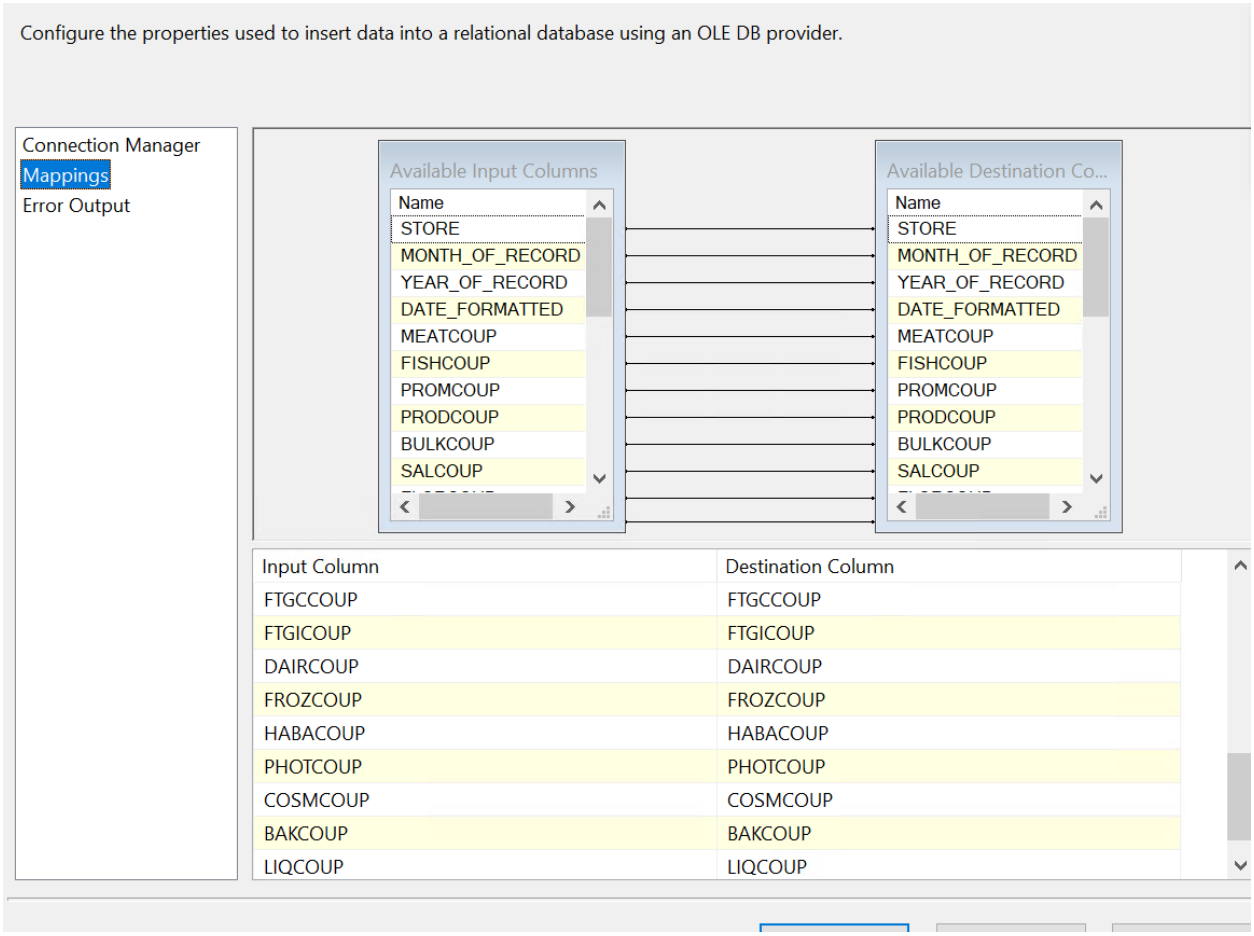
ETL For Promotions Fact Table:

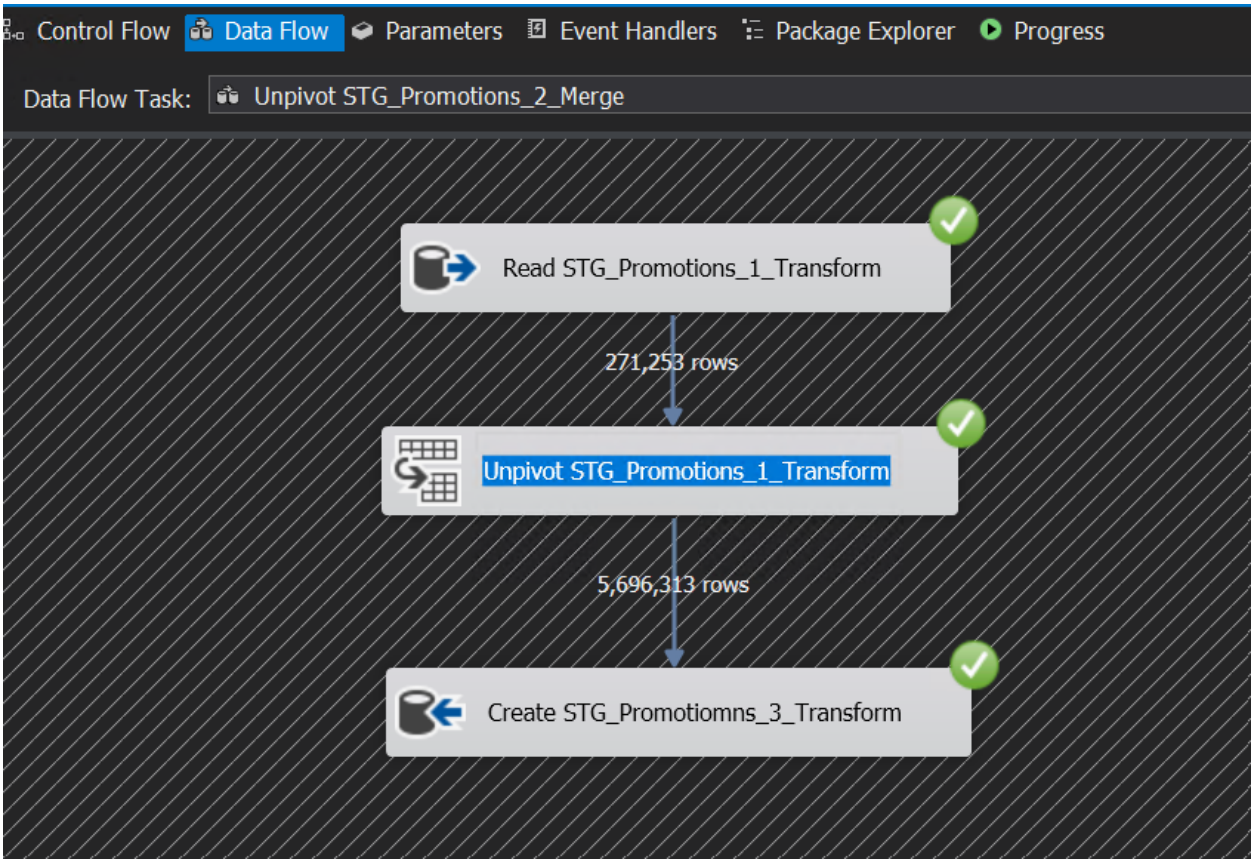


Control Flow for Promotions Fact Table

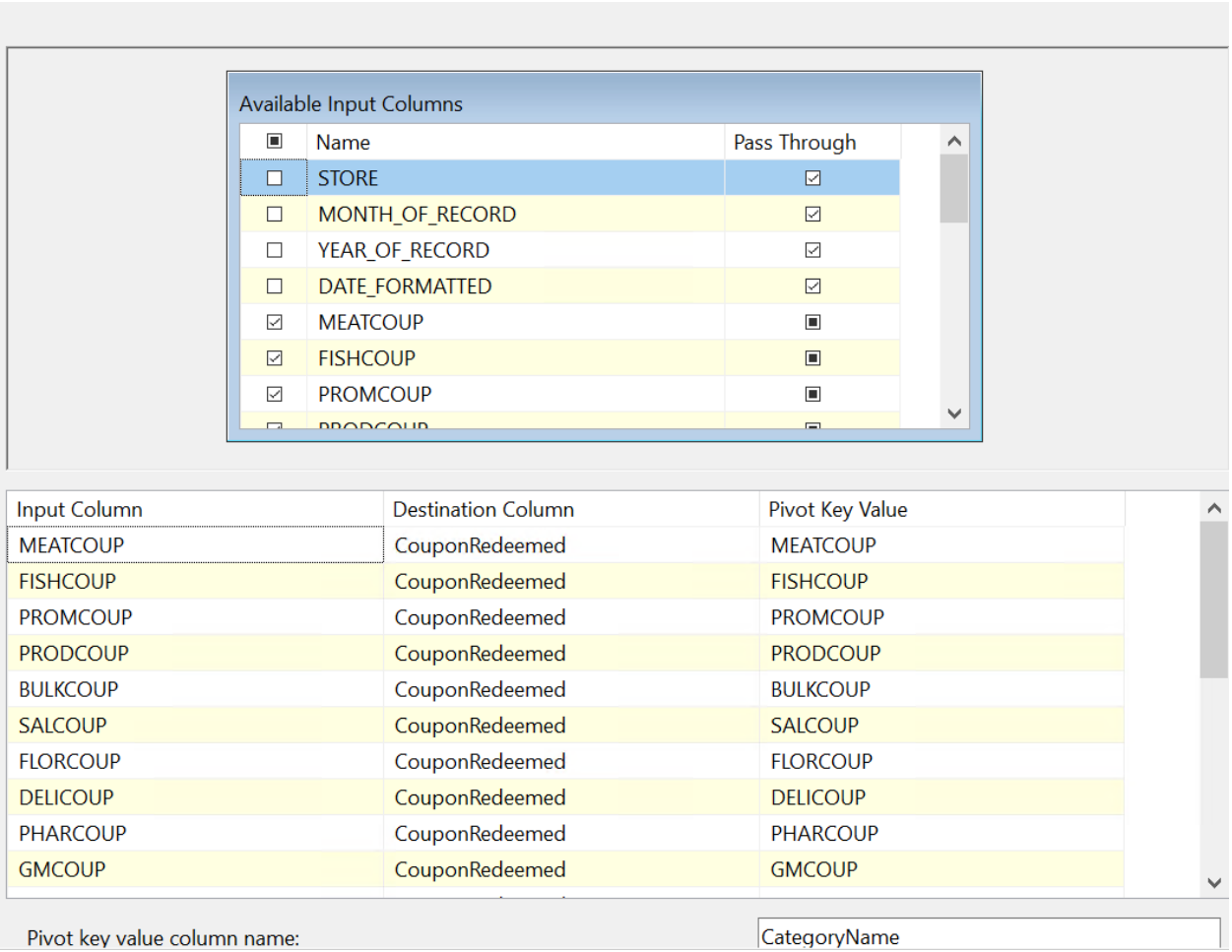


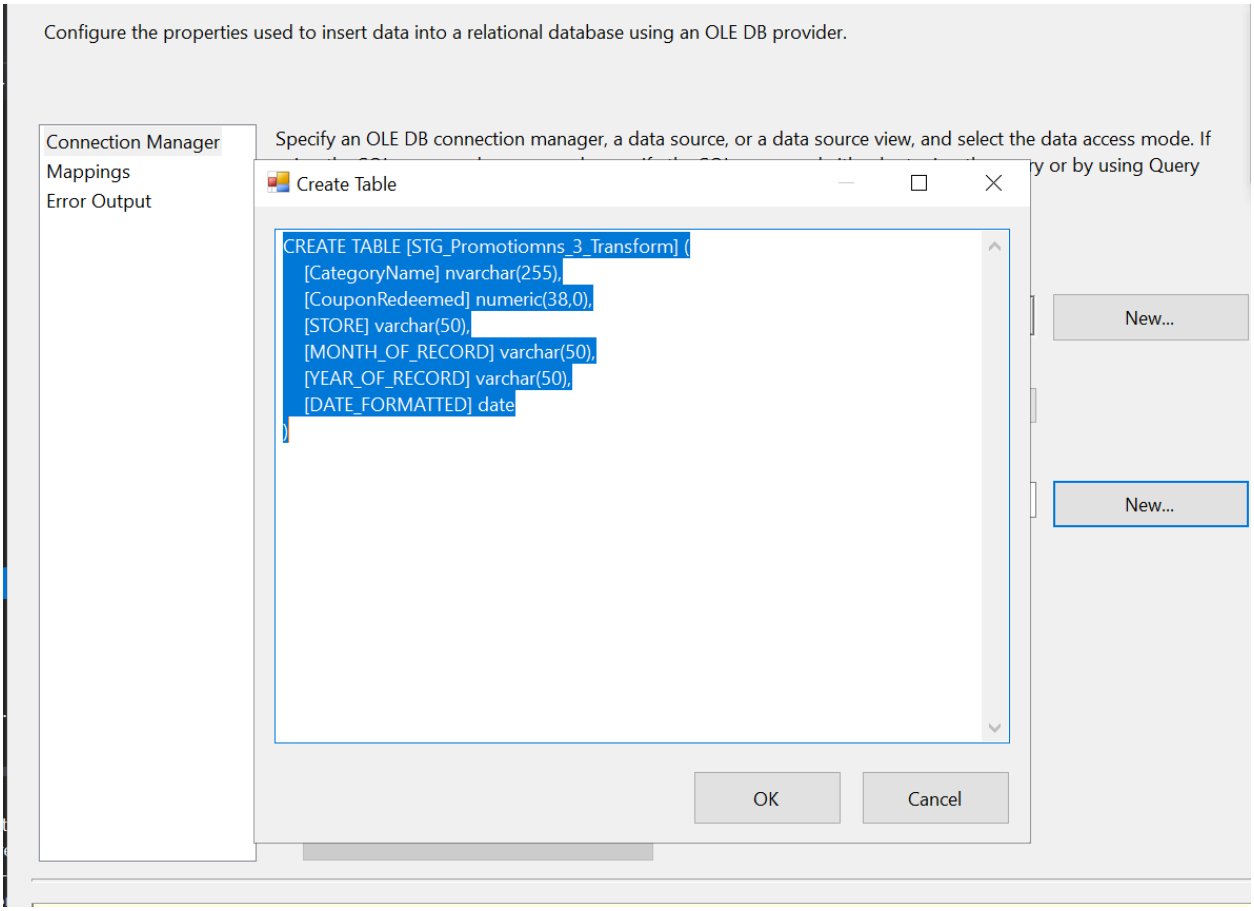
Data Flow for Transform table creation

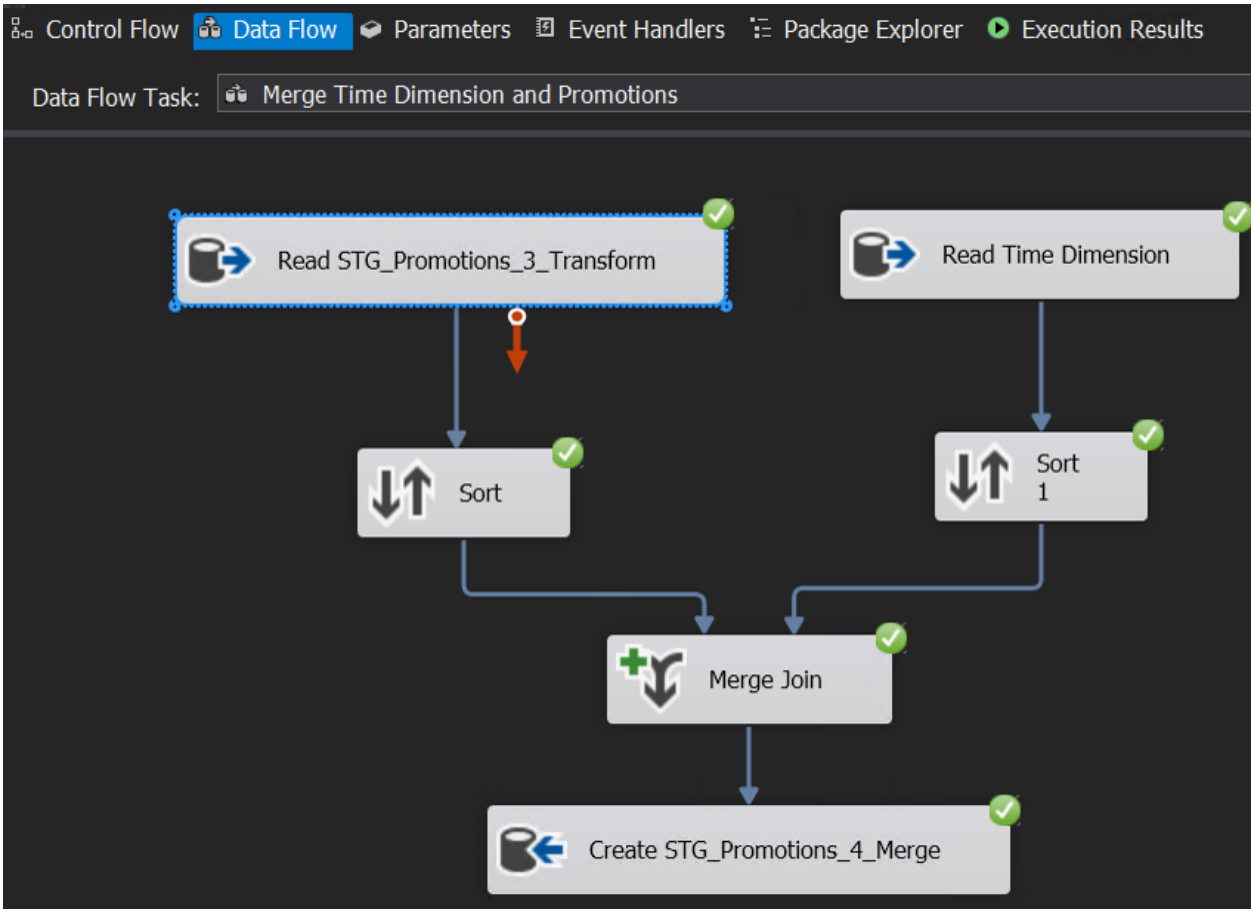




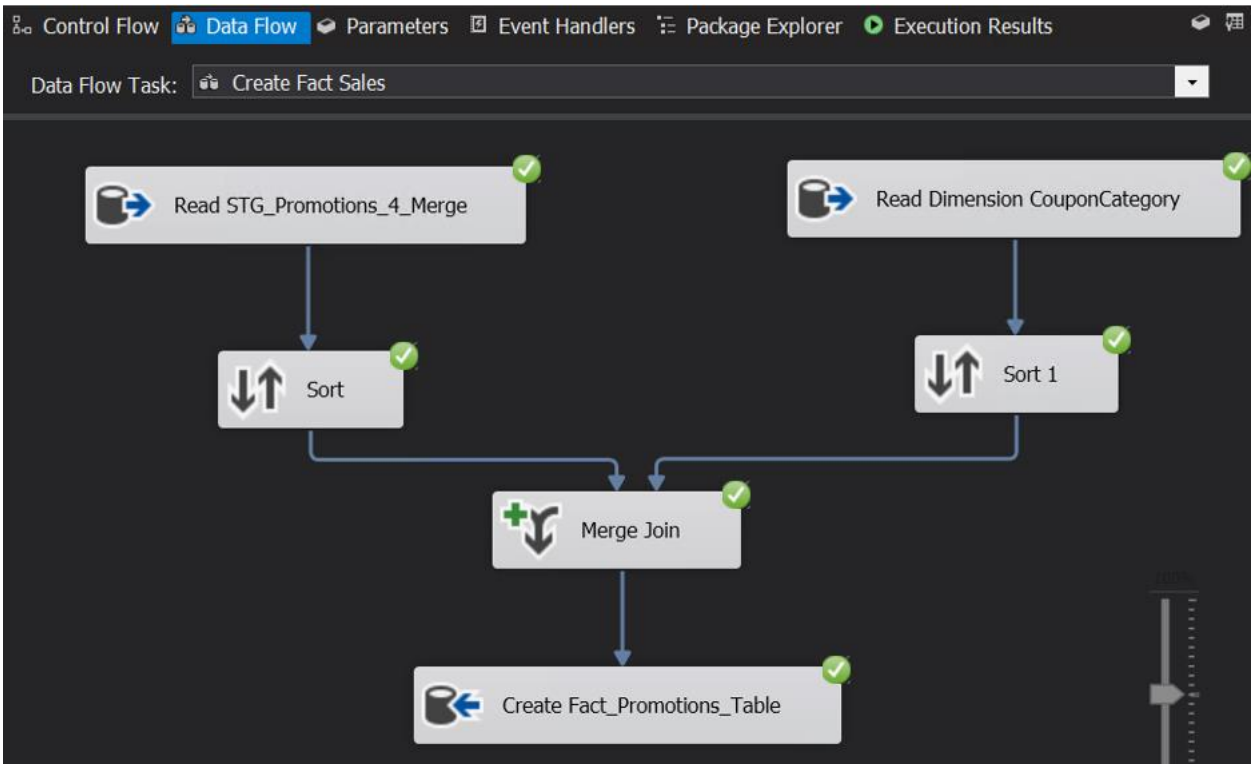
Data Flow for Unpivot


Available Input Columns
Name
Pass Through
STORE
MONTH_OF_RECORD
YEAR_OF_RECORD
DATE_FORMATTED
MEATCOUP
FISHCOUP
PROMCOUP
Input Column
Destination Column
Pivot Key Value
MEATCOUP
CouponRedeemed
MEATCOUP
FISHCOUP
CouponRedeemed
FISHCOUP
PROMCOUP
CouponRedeemed
PROMCOUP
PRODCOUP
CouponRedeemed
PRODCOUP
BULKCOUP
CouponRedeemed
BULKCOUP
SALCOUP
CouponRedeemed
SALCOUP
FLORCOUP
CouponRedeemed
FLORCOUP
DELICOUP
CouponRedeemed
DELICOUP
PHARCOUP
CouponRedeemed
PHARCOUP
GMCOUP
CouponRedeemed
GMCOUP
Pivot key value column name:
CategoryName


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
y or by using Query
Create Table
CREATE TABLE [STG_Promotiomns_3_Transform] (
[CategoryName] nvarchar(255),
[CouponRedeemed] numeric(38,0),
[STORE] varchar(50),
[MONTH_OF_RECORD] varchar(50),
[YEAR_OF_RECORD] varchar(50),
[DATE_FORMATTED] date
)
New...
New...
OK
Cancel

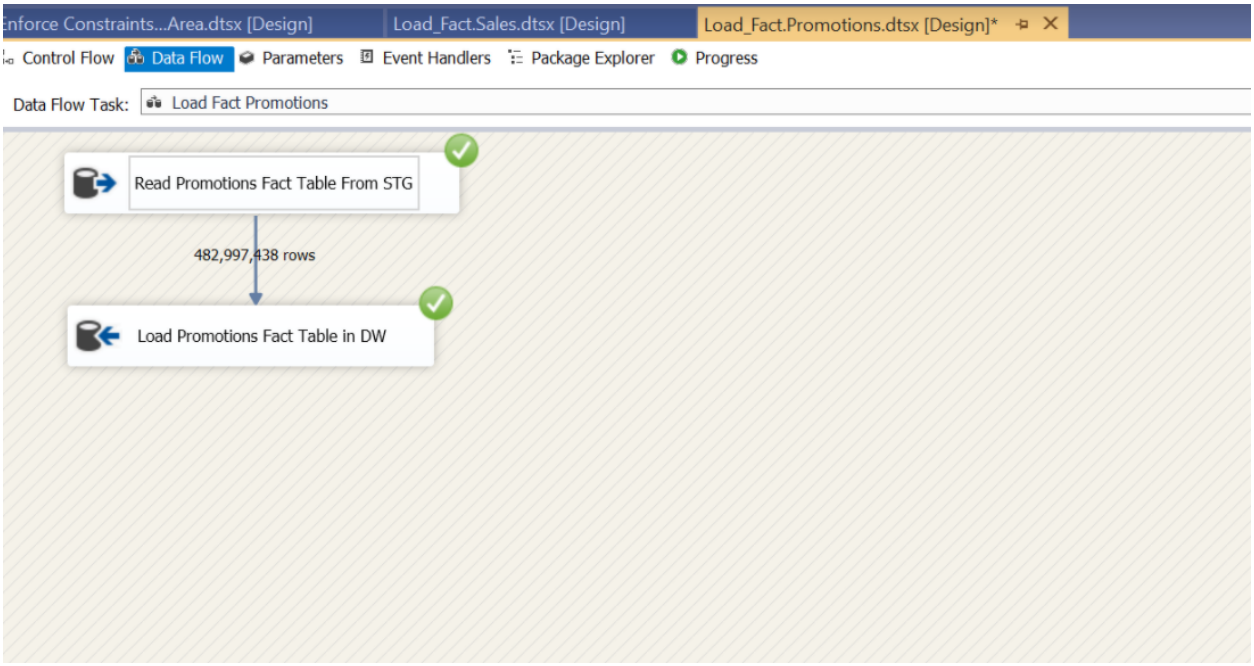


Data Flow for Merge of Promotions and Time Dimension

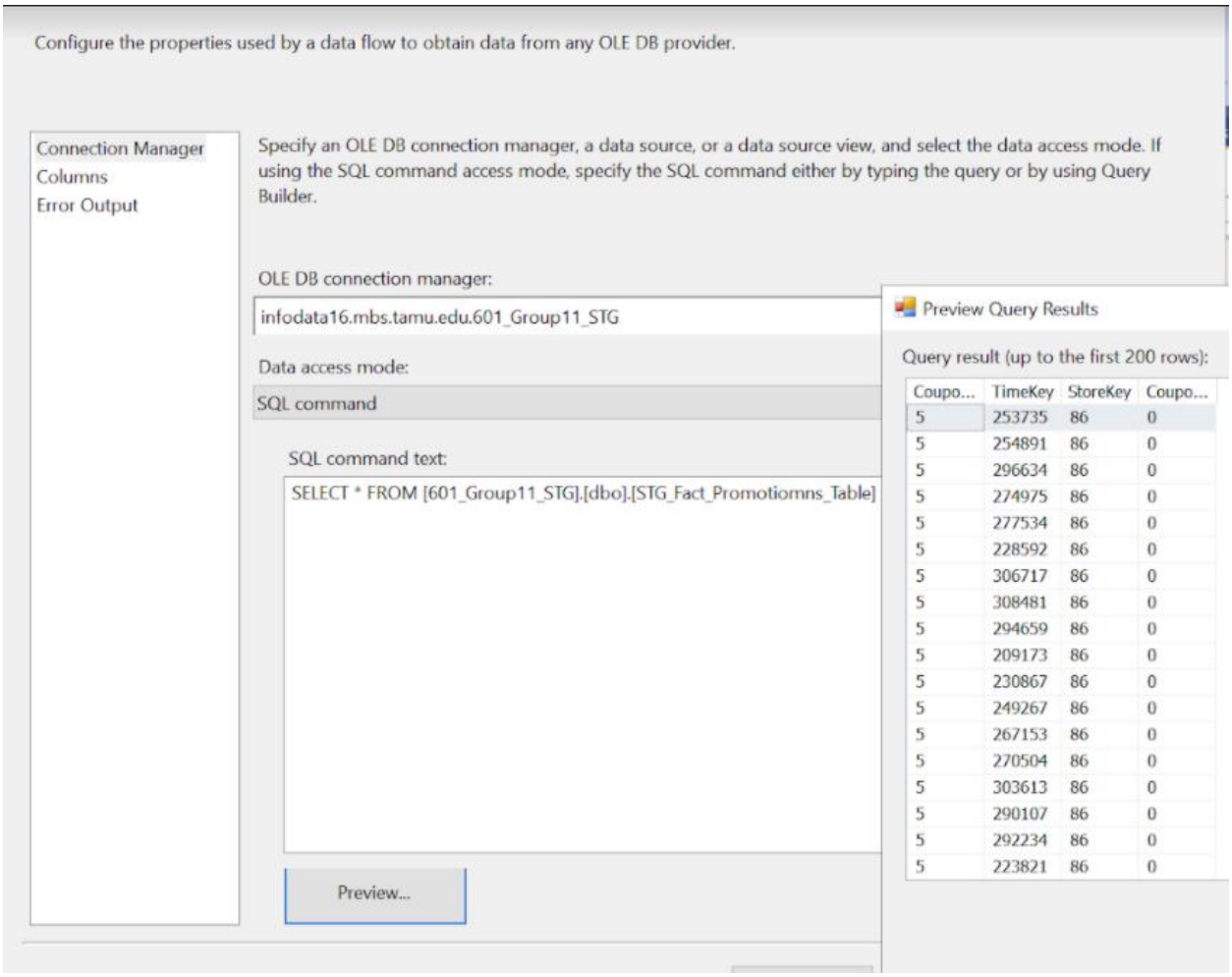


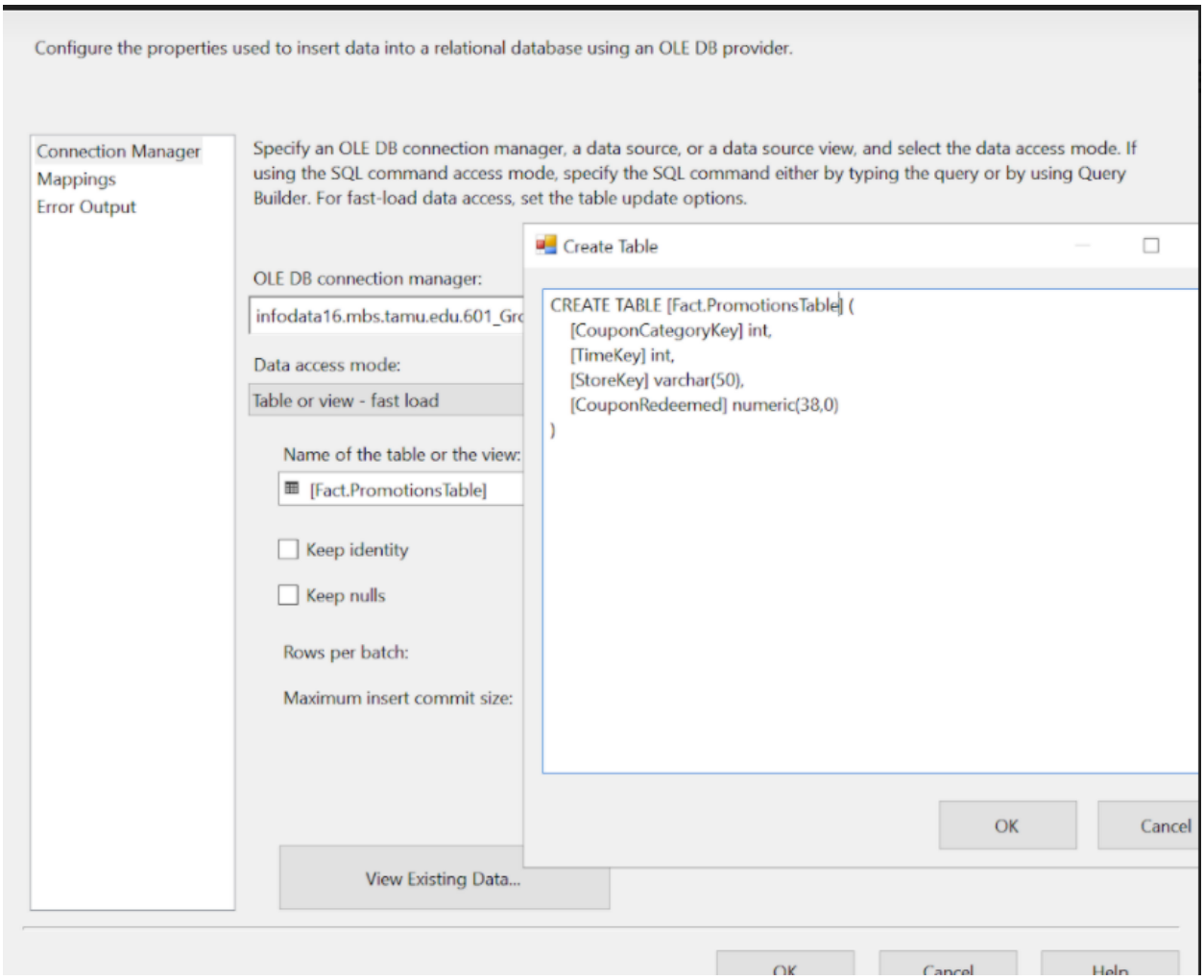
Data Flow for Merge of Promotions and and CouponCategory Dimension

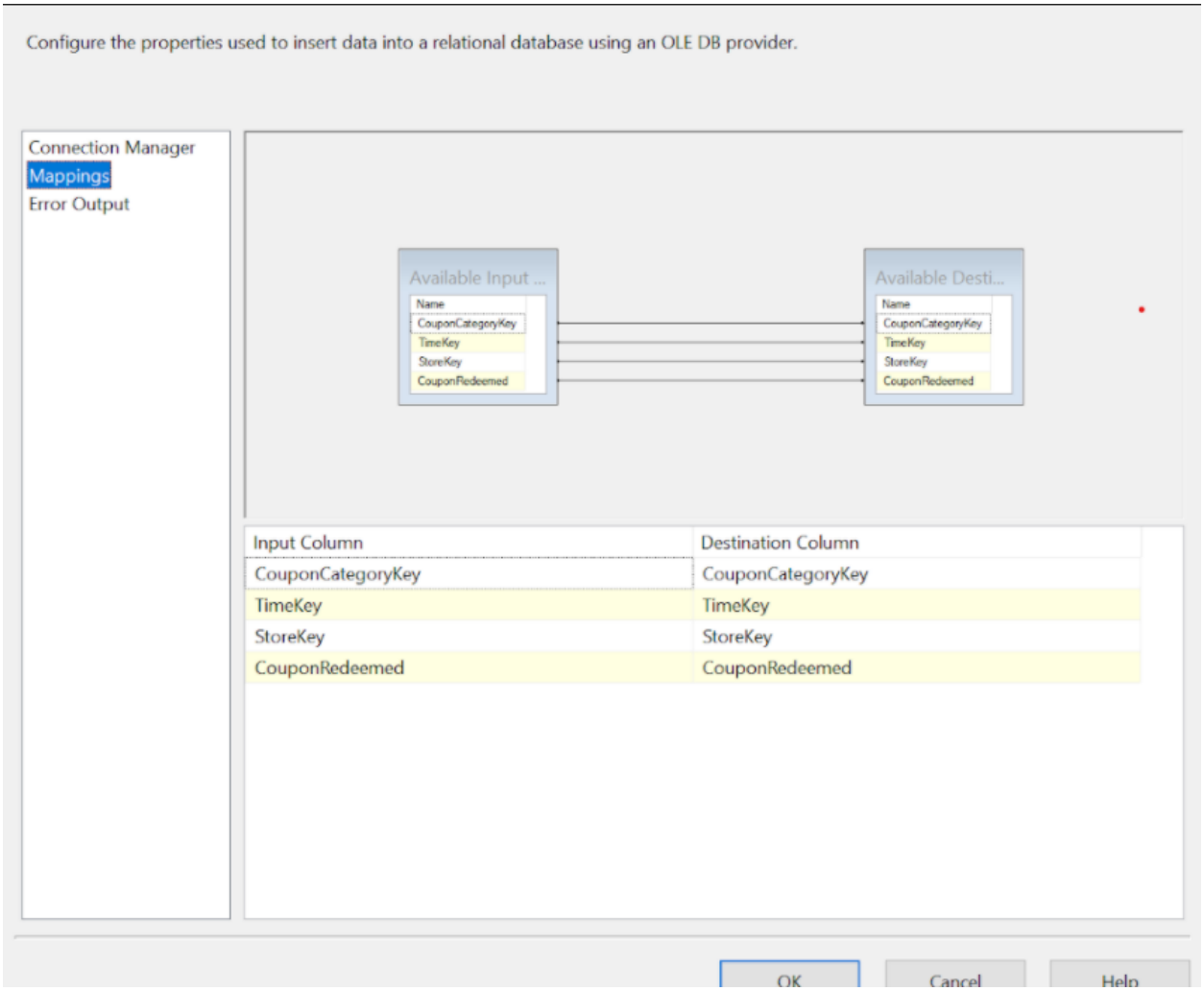
Control Flow for Load of Promotions Fact Table



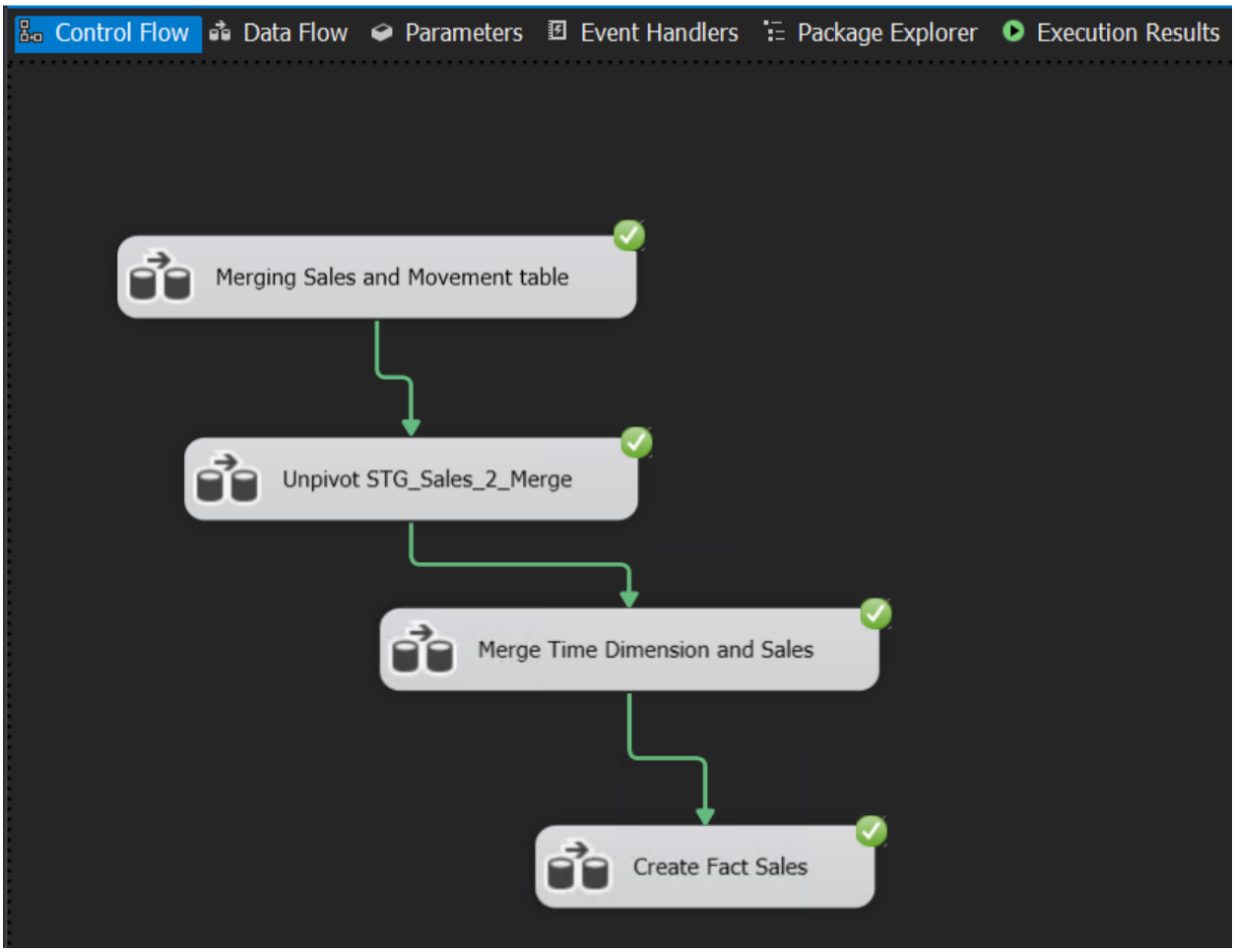
Data Flow for Load Promotions Fact


Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
Data access mode:
SQL command
SQL command text:
SELECT * FROM [601_Group11_STG].[dbo].[STG_Fact_Promotiomns_Table]
Preview...
Preview Query Results
Query result (up to the first 200 rows):
Coupo... TimeKey StoreKey Coupo...
5 253735 86 0
5 254891 86 0
5 296634 86 0
5 274975 86 0
5 277534 86 0
5 228592 86 0
5 306717 86 0
5 308481 86 0
5 294659 86 0
5 209173 86 0
5 230867 86 0
5 249267 86 0
5 267153 86 0
5 270504 86 0
5 303613 86 0
5 290107 86 0
5 292234 86 0
5 223821 86 0

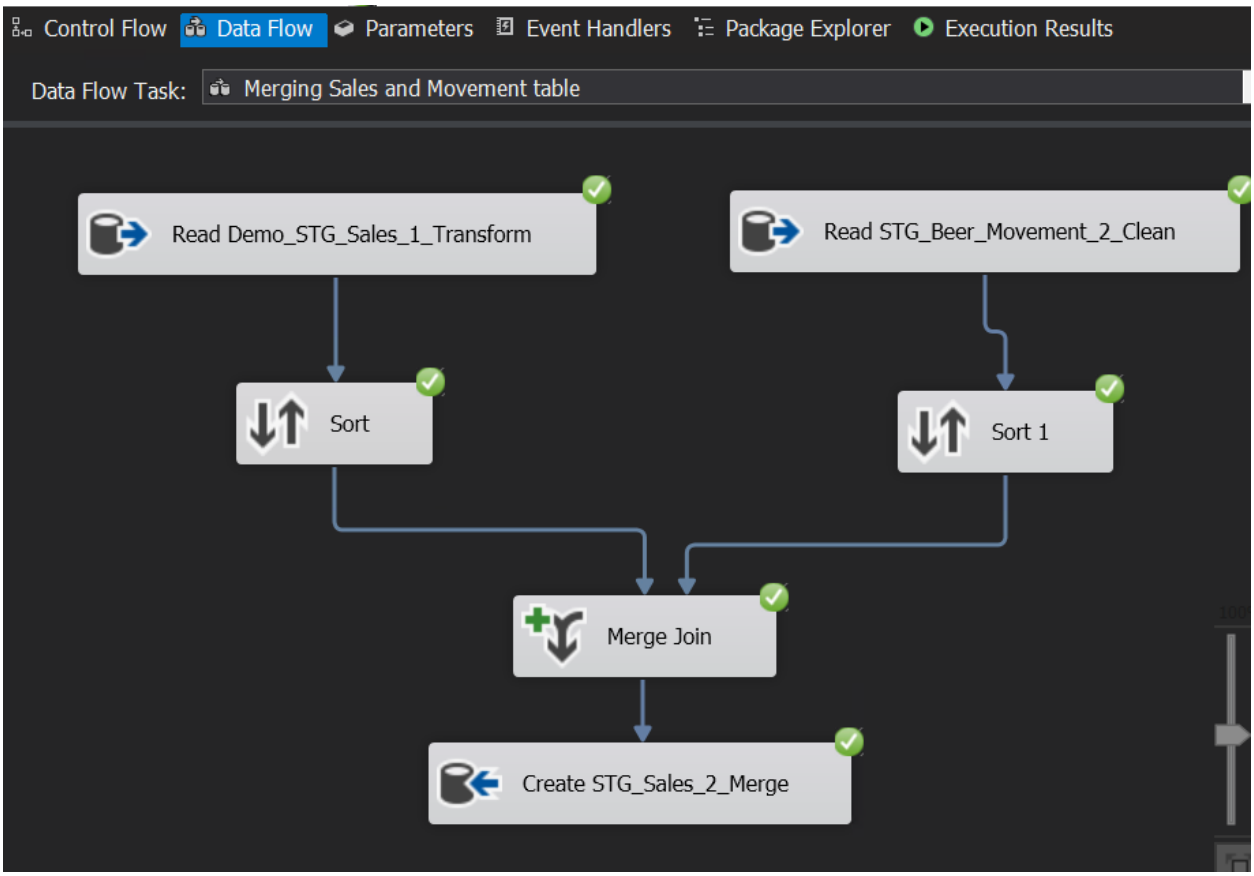

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Gr
Data access mode:
Table or view - fast load
Name of the table or the view:
[Fact.PromotionsTable]
Keep identity
Keep nulls
Rows per batch:
Maximum insert commit size:
View Existing Data...
Create Table
CREATE TABLE [Fact.PromotionsTable] (
[CouponCategoryKey] int,
[TimeKey] int,
[StoreKey] varchar(50),
[CouponRedeemed] numeric(38,0)
)
OK
Cancel
OK
Cancel
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Available Input ...
Name
CouponCategoryKey
TimeKey
StoreKey
CouponRedeemed
Available Desti...
Name
CouponCategoryKey
TimeKey
StoreKey
CouponRedeemed
Input Column
Destination Column
CouponCategoryKey
CouponCategoryKey
TimeKey
TimeKey
StoreKey
StoreKey
CouponRedeemed
CouponRedeemed
OK
Cancel
Help

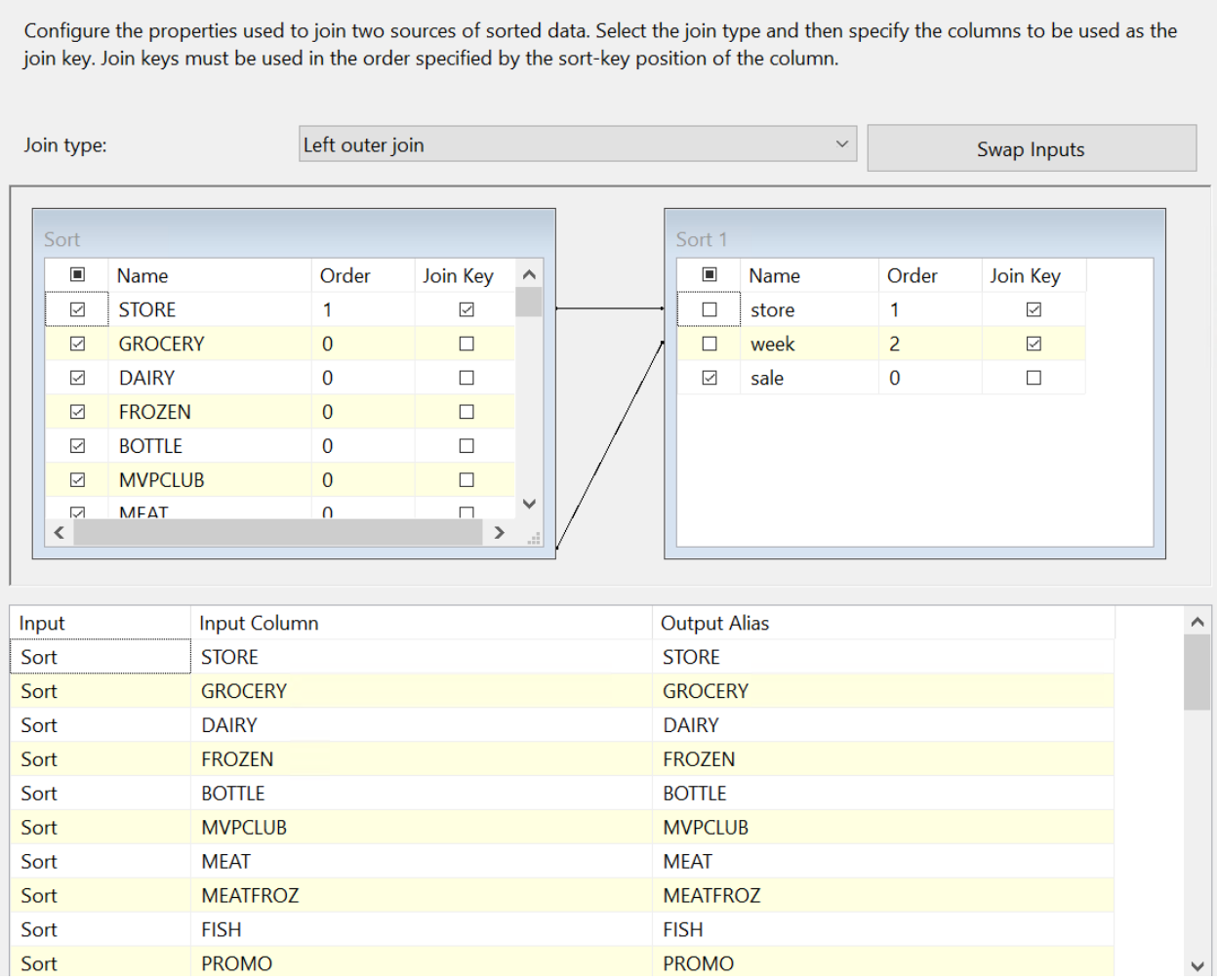
ETL For Sales Fact Table:



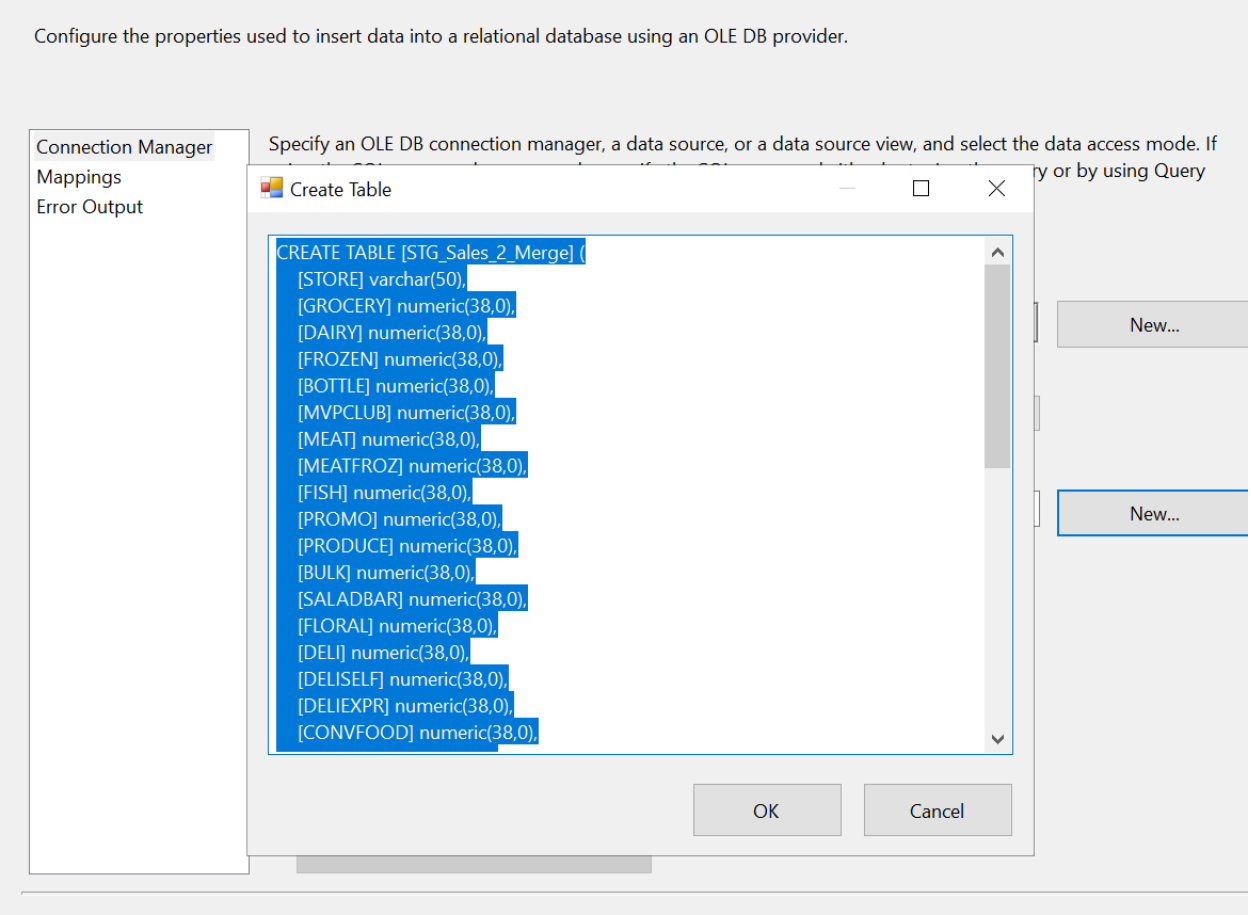
Control Flow for Sales Fact Table

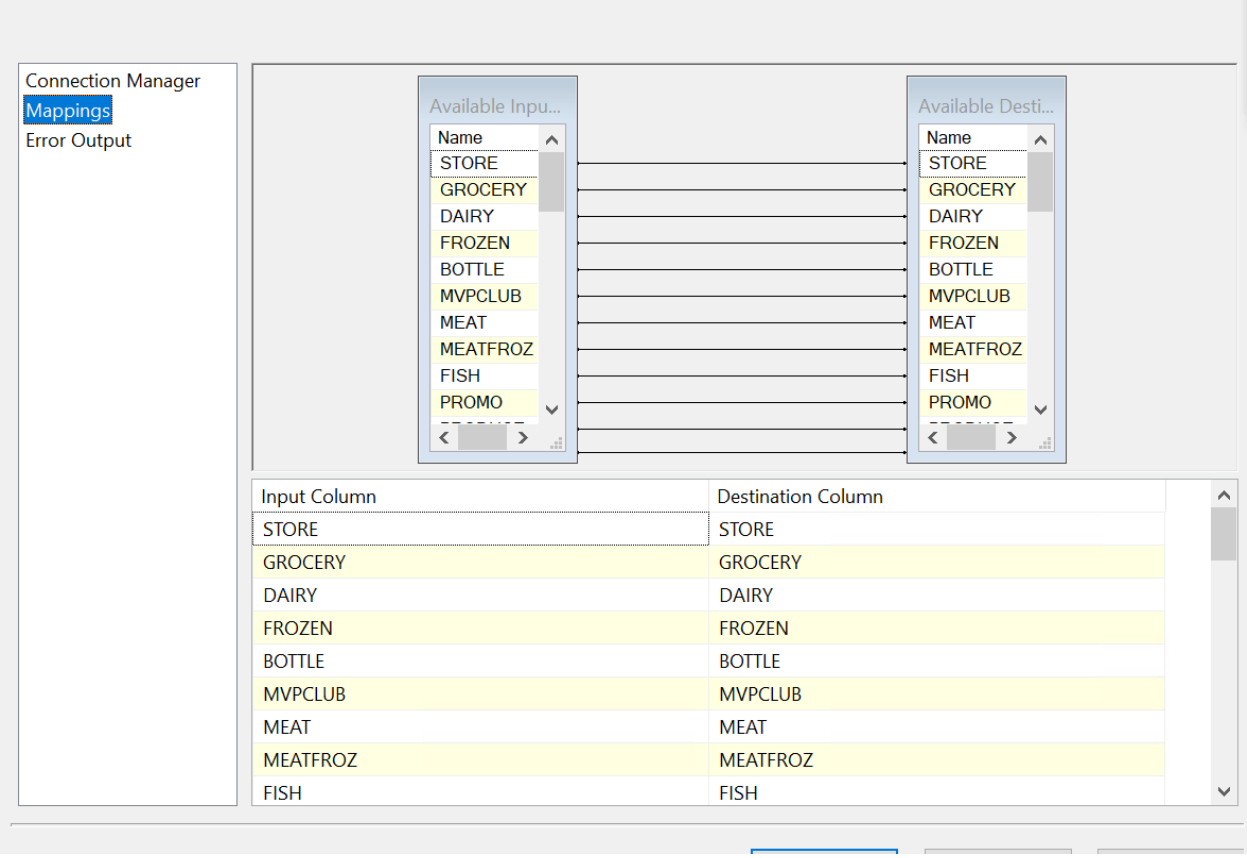


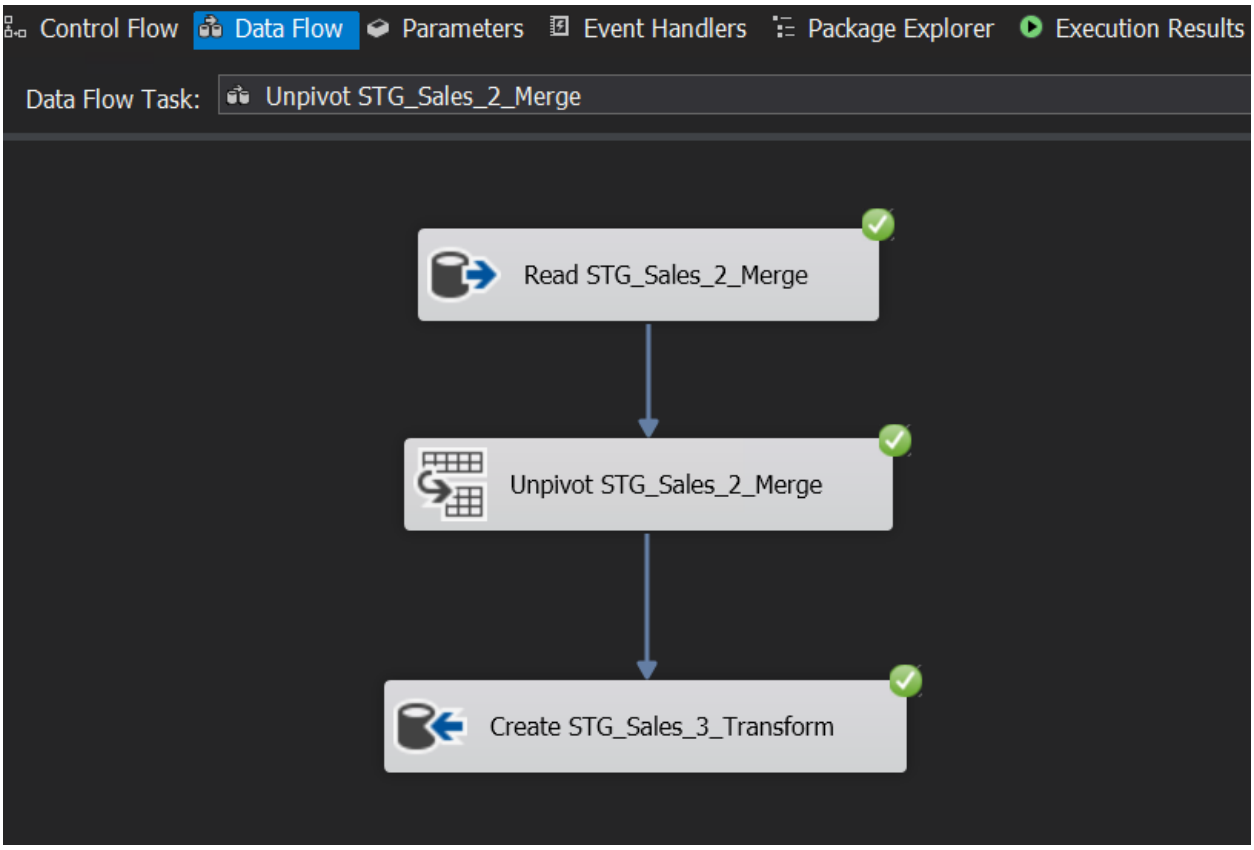
Data Flow for merge of Sales transform and beer movement table


Configure the properties used to join two sources of sorted data. Select the join type and then specify the columns to be used as the join key. Join keys must be used in the order specified by the sort-key position of the column.
Join type:
Left outer join
Swap Inputs
Sort
Name
Order
Join Key
STORE 1
GROCERY 0
DAIRY 0
FROZEN 0
BOTTLE 0
MVPCLUB 0
Sort 1
Name
Order
Join Key
store 1
week 2
sale 0
Input
Input Column
Output Alias
Sort STORE STORE
Sort GROCERY GROCERY
Sort DAIRY DAIRY
Sort FROZEN FROZEN
Sort BOTTLE BOTTLE
Sort MVPCLUB MVPCLUB
Sort MEAT MEAT
Sort MEATFROZ MEATFROZ
Sort FISH FISH
Sort PROMO PROMO

Merge Join

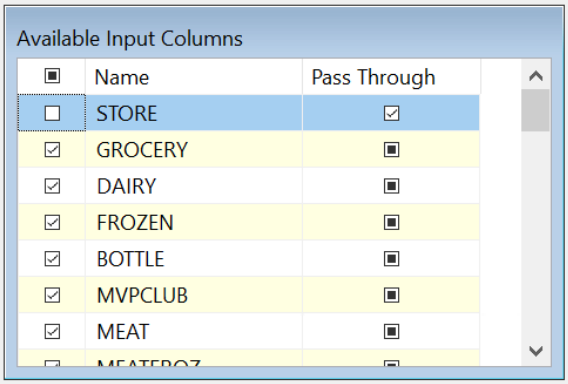

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
ry or by using Query
Create Table
CREATE TABLE [STG_Sales_2_Merge] (
[STORE] varchar(50),
[GROCERY] numeric(38,0),
[DAIRY] numeric(38,0),
[FROZEN] numeric(38,0),
[BOTTLE] numeric(38,0),
[MVPCLUB] numeric(38,0),
[MEAT] numeric(38,0),
[MEATFROZ] numeric(38,0),
[FISH] numeric(38,0),
[PROMO] numeric(38,0),
[PRODUCE] numeric(38,0),
[BULK] numeric(38,0),
[SALADBAR] numeric(38,0),
[FLORAL] numeric(38,0),
[DELI] numeric(38,0),
[DELISELF] numeric(38,0),
[DELIEXPR] numeric(38,0),
[CONVFOOD] numeric(38,0),
OK
Cancel
New...
New...


Connection Manager
Mappings
Error Output
Available Inpu...
Name
STORE
GROCERY
DAIRY
FROZEN
BOTTLE
MVPCLUB
MEAT
MEATFROZ
FISH
PROMO
Available Desti...
Name
STORE
GROCERY
DAIRY
FROZEN
BOTTLE
MVPCLUB
MEAT
MEATFROZ
FISH
PROMO
Input Column
Destination Column
STORE
STORE
GROCERY
GROCERY
DAIRY
DAIRY
FROZEN
FROZEN
BOTTLE
BOTTLE
MVPCLUB
MVPCLUB
MEAT
MEAT
MEATFROZ
MEATFROZ
FISH
FISH



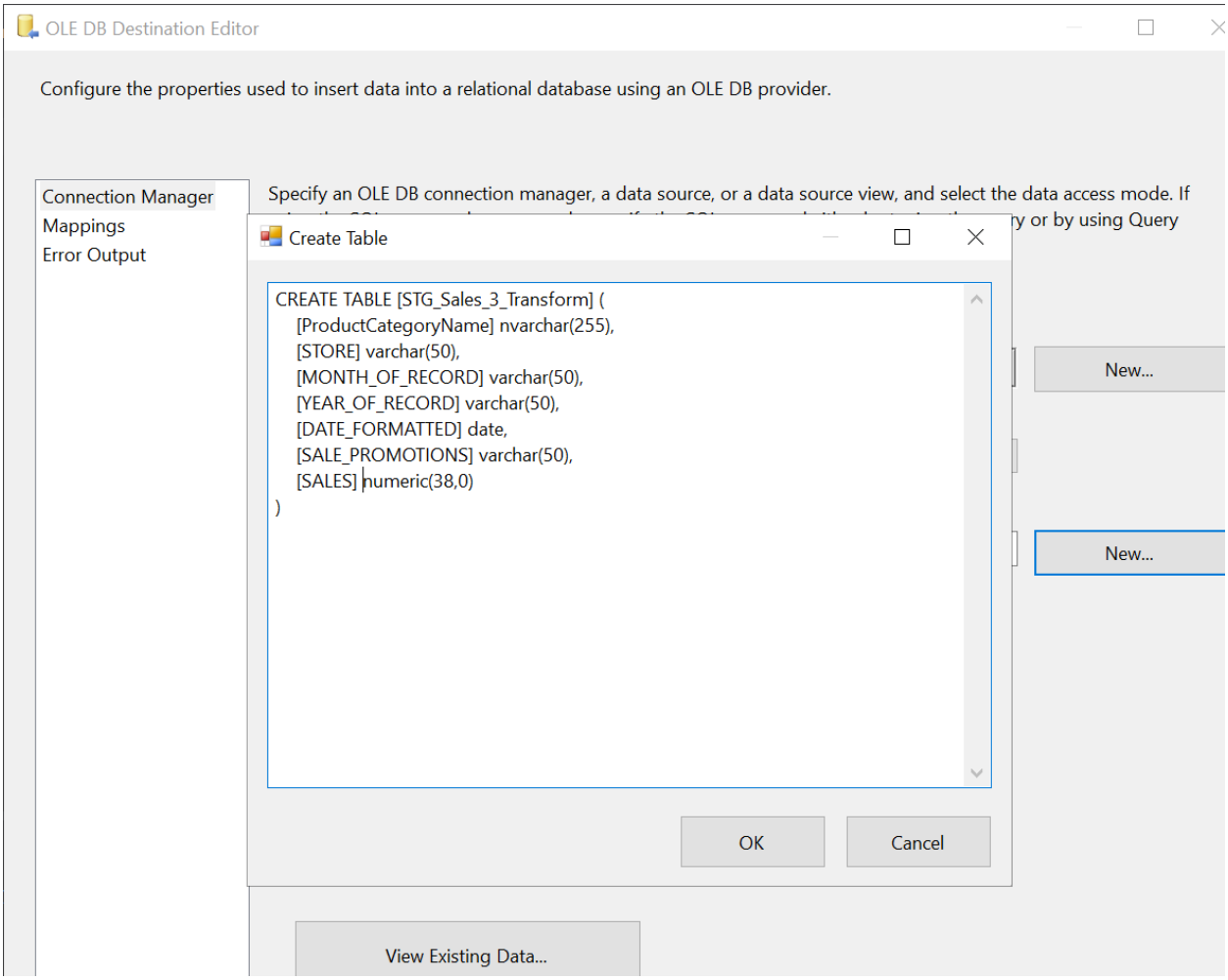
ata Flow for Unpivot

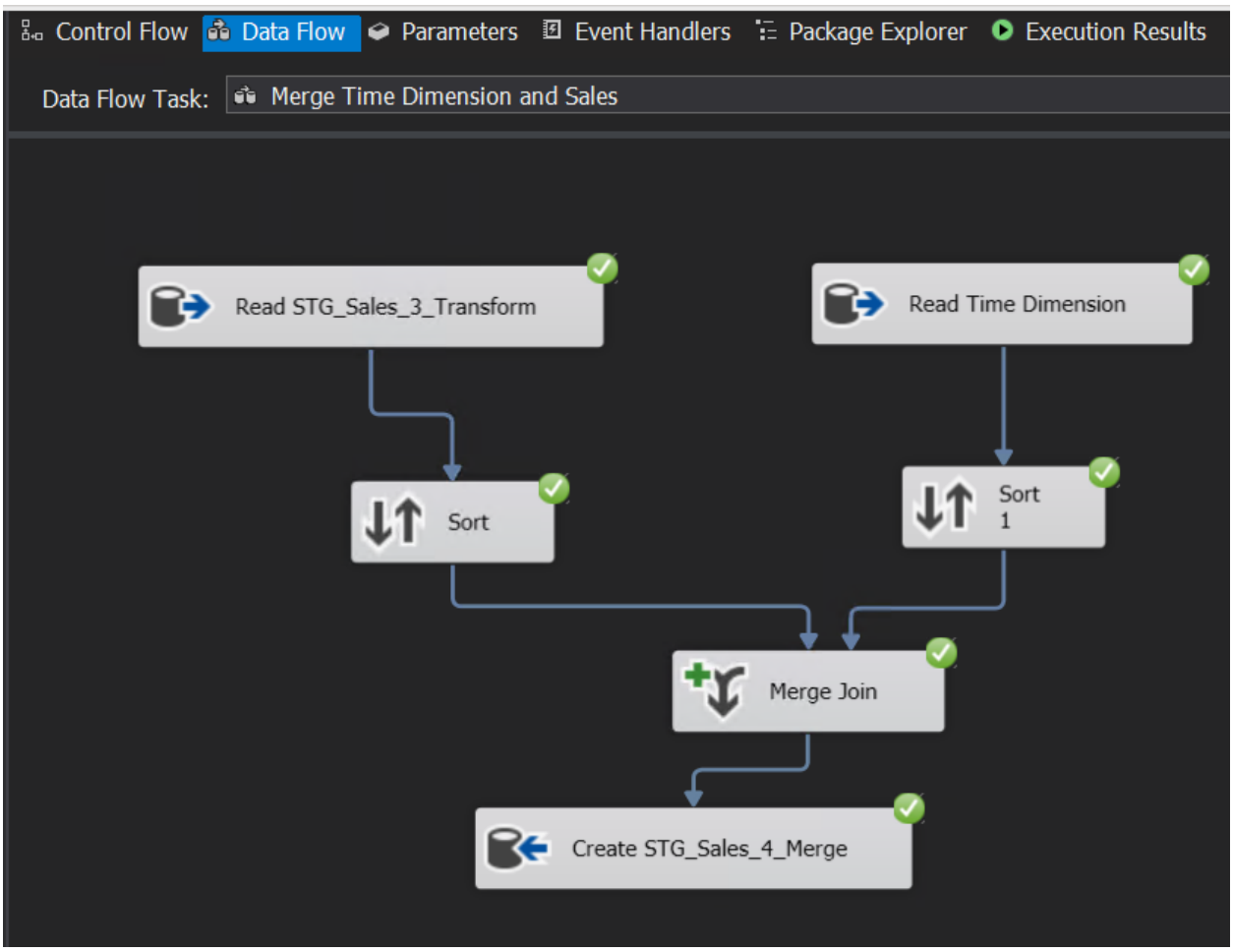
Specify the columns to pivot into rows to make an unnormalized dataset into a more normalized version.



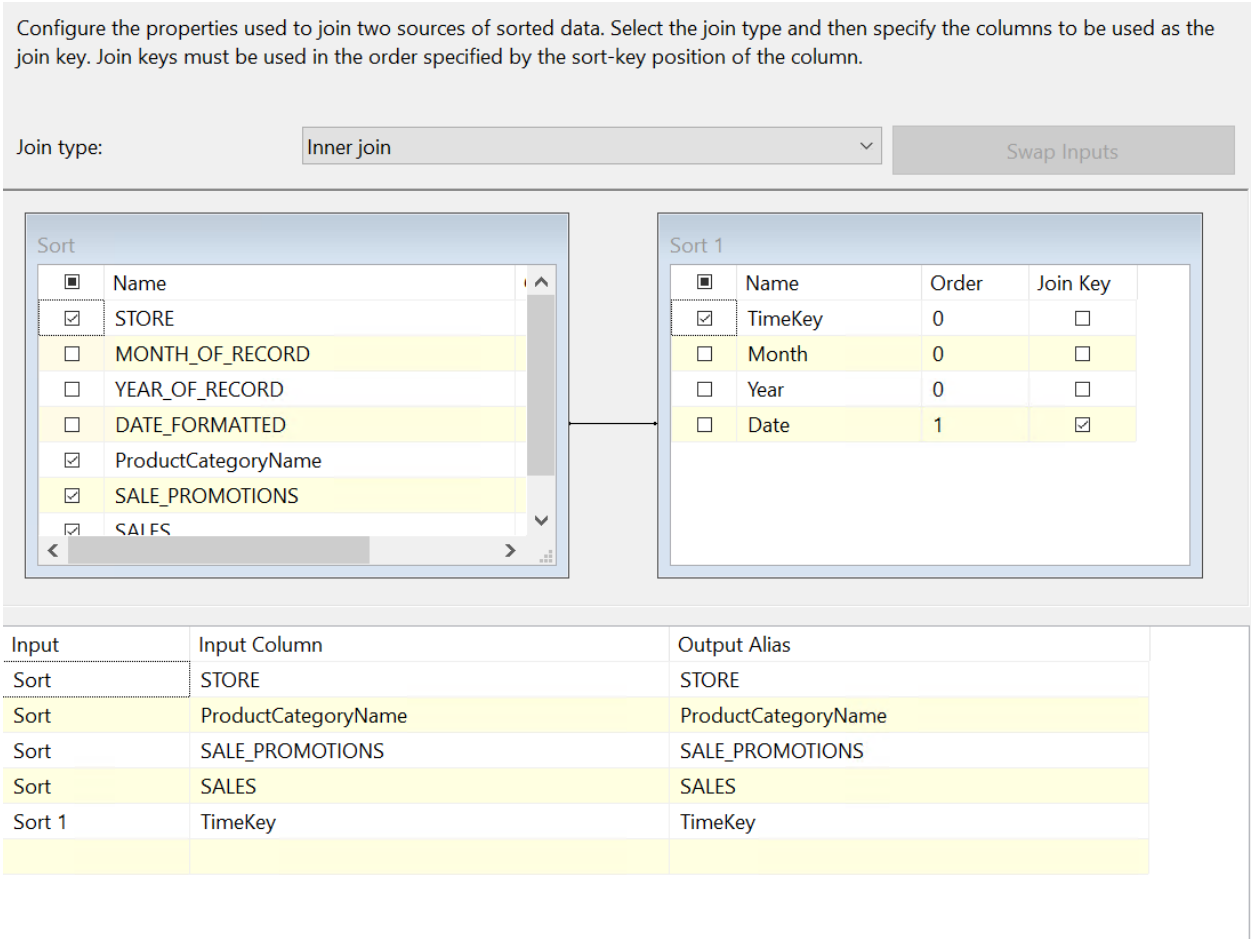
Input Column	Destination Column	Pivot Key Value
GROCERY	Sales	GROCERY
DAIRY	Sales	DAIRY
FROZEN	Sales	FROZEN
BOTTLE	Sales	BOTTLE
MVPCLUB	Sales	MVPCLUB
MEAT	Sales	MEAT
MEATFROZ	Sales	MEATFROZ
FISH	Sales	FISH
PROMO	Sales	PROMO
PRODUCE	Sales	PRODUCE

Pivot key value column name: ProductCategoryName

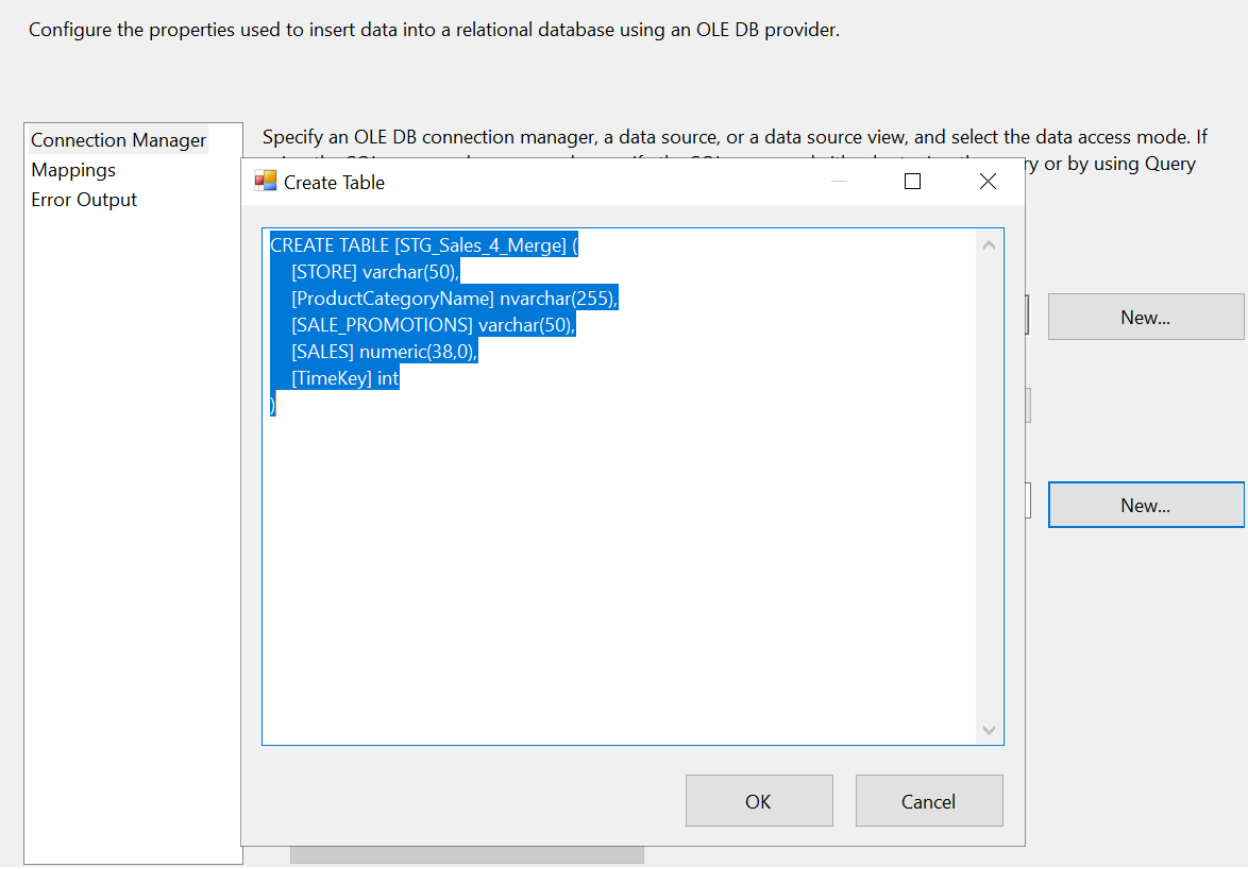

OLE DB Destination Editor
Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
or by using Query
Create Table
CREATE TABLE [STG_Sales_3_Transform] (
[ProductCategoryName] nvarchar(255),
[STORE] varchar(50),
[MONTH_OF_RECORD] varchar(50),
[YEAR_OF_RECORD] varchar(50),
[DATE_FORMATTED] date,
[SALE_PROMOTIONS] varchar(50),
[SALES] numeric(38,0)
)
OK
Cancel
New...
New...
View Existing Data...

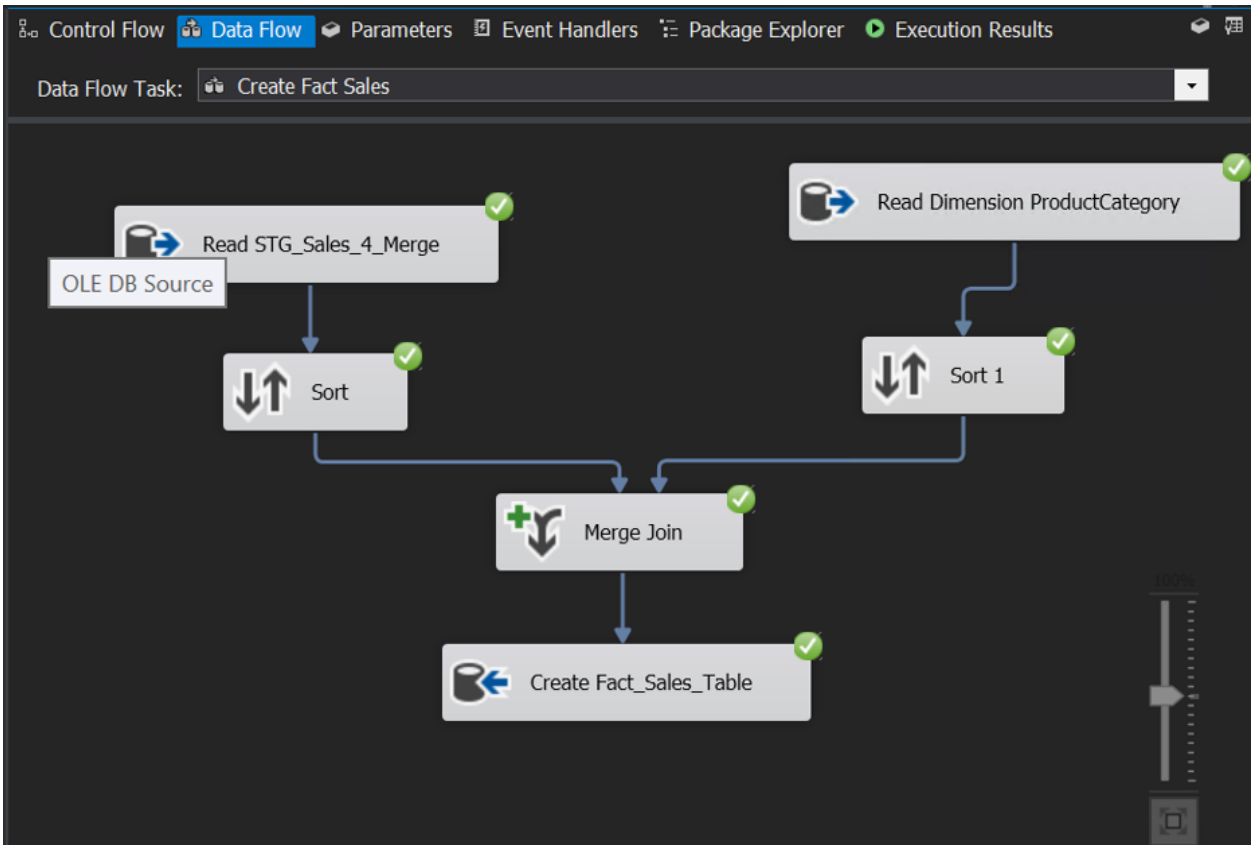


Data Flow for merge of Sales transform and time dimension

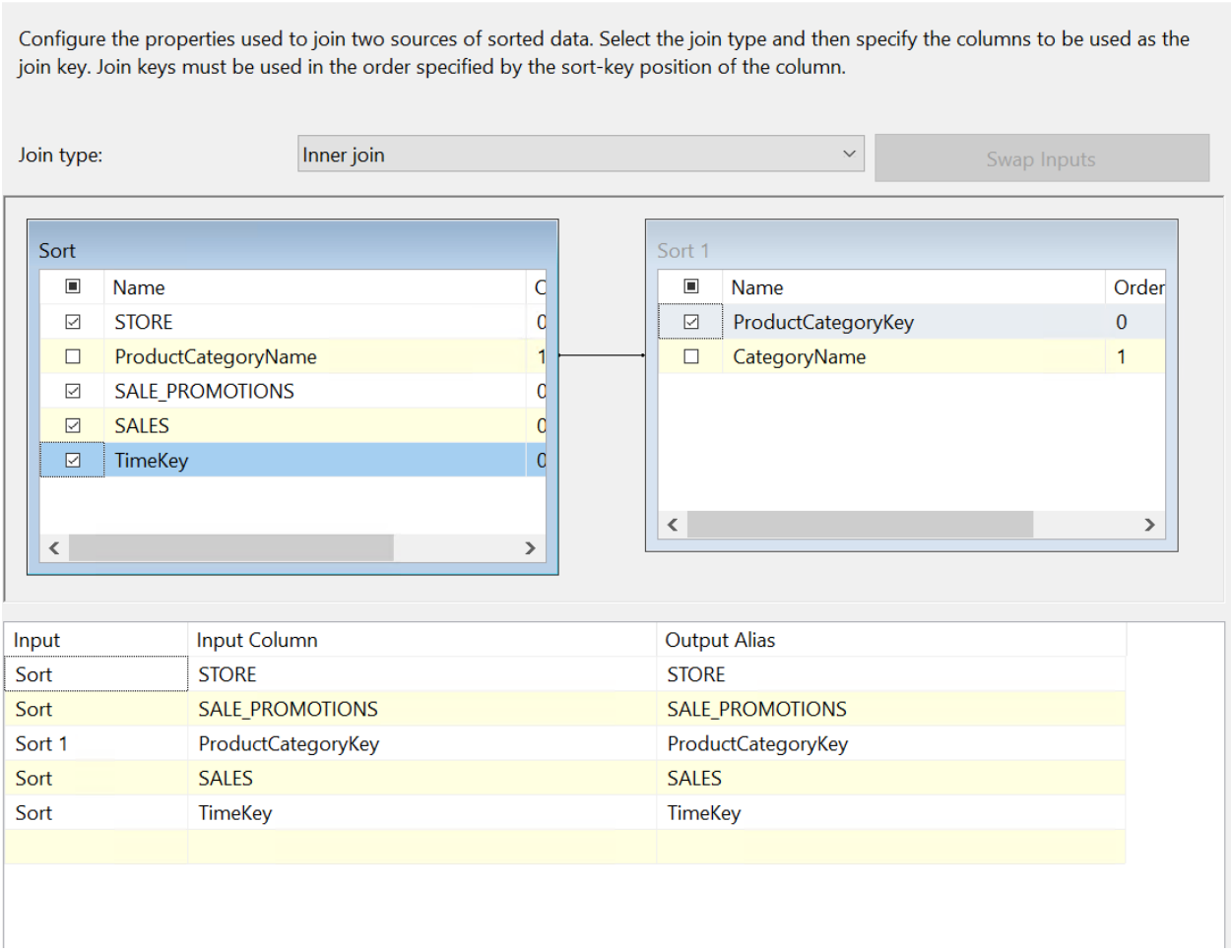

Configure the properties used to join two sources of sorted data. Select the join type and then specify the columns to be used as the join key. Join keys must be used in the order specified by the sort-key position of the column.
Join type:
Inner join
Swap Inputs
Sort
Name
STORE
MONTH_OF_RECORD
YEAR_OF_RECORD
DATE_FORMATTED
ProductCategoryName
SALE_PROMOTIONS
Sort 1
Name
Order
Join Key
TimeKey
0
Month
0
Year
0
Date
1
Input
Input Column
Output Alias
Sort
STORE
STORE
Sort
ProductCategoryName
ProductCategoryName
Sort
SALE_PROMOTIONS
SALE_PROMOTIONS
Sort
SALES
SALES
Sort 1
TimeKey
TimeKey

Merge Join

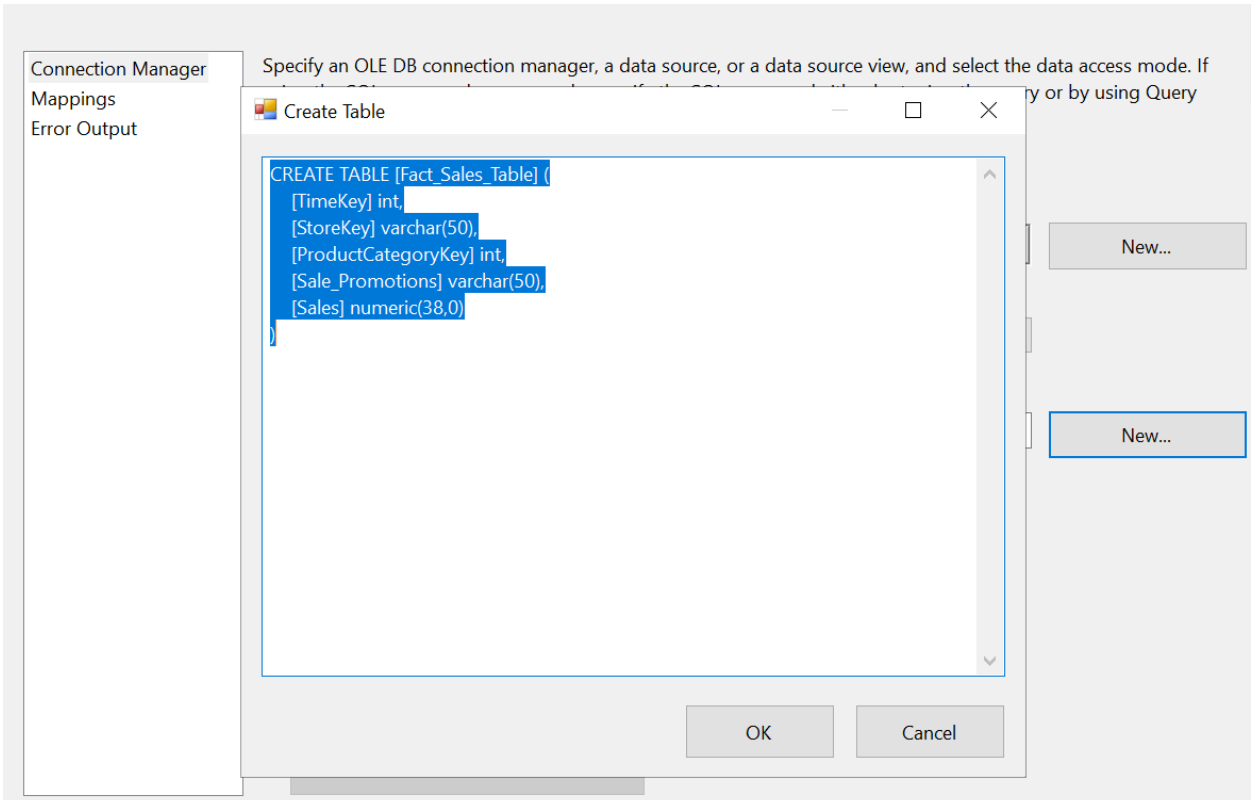

Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
y or by using Query
Create Table
CREATE TABLE [STG_Sales_4_Merge] (
[STORE] varchar(50),
[ProductCategoryName] nvarchar(255),
[SALE_PROMOTIONS] varchar(50),
[SALES] numeric(38,0),
[TimeKey] int
)
OK
Cancel
New...
New...

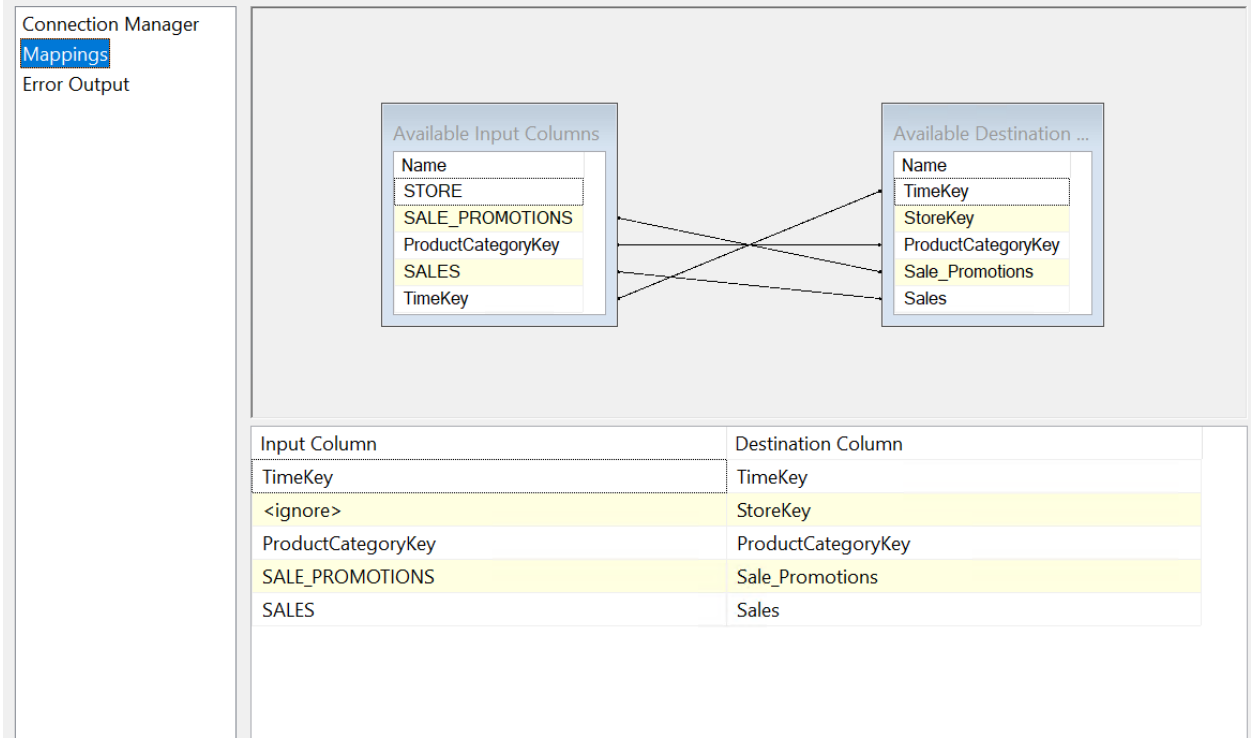


Data Flow for merge of Sales and ProductCategory dimension

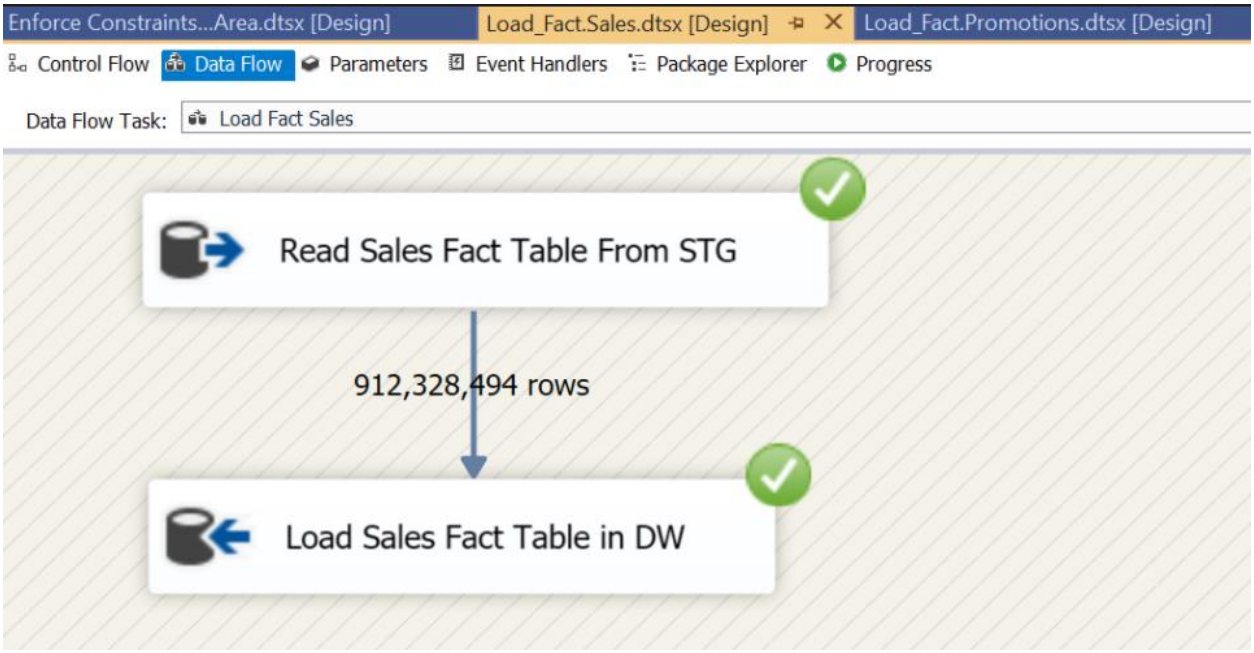

Configure the properties used to join two sources of sorted data. Select the join type and then specify the columns to be used as the join key. Join keys must be used in the order specified by the sort-key position of the column.
Join type:
Inner join
Swap Inputs
Sort
Name
STORE
ProductCategoryName
SALE_PROMOTIONS
SALES
TimeKey
Sort 1
Name
Order
ProductCategoryKey
0
CategoryName
1
Input
Input Column
Output Alias
Sort
STORE
STORE
Sort
SALE_PROMOTIONS
SALE_PROMOTIONS
Sort 1
ProductCategoryKey
ProductCategoryKey
Sort
SALES
SALES
Sort
TimeKey
TimeKey

Merge Join

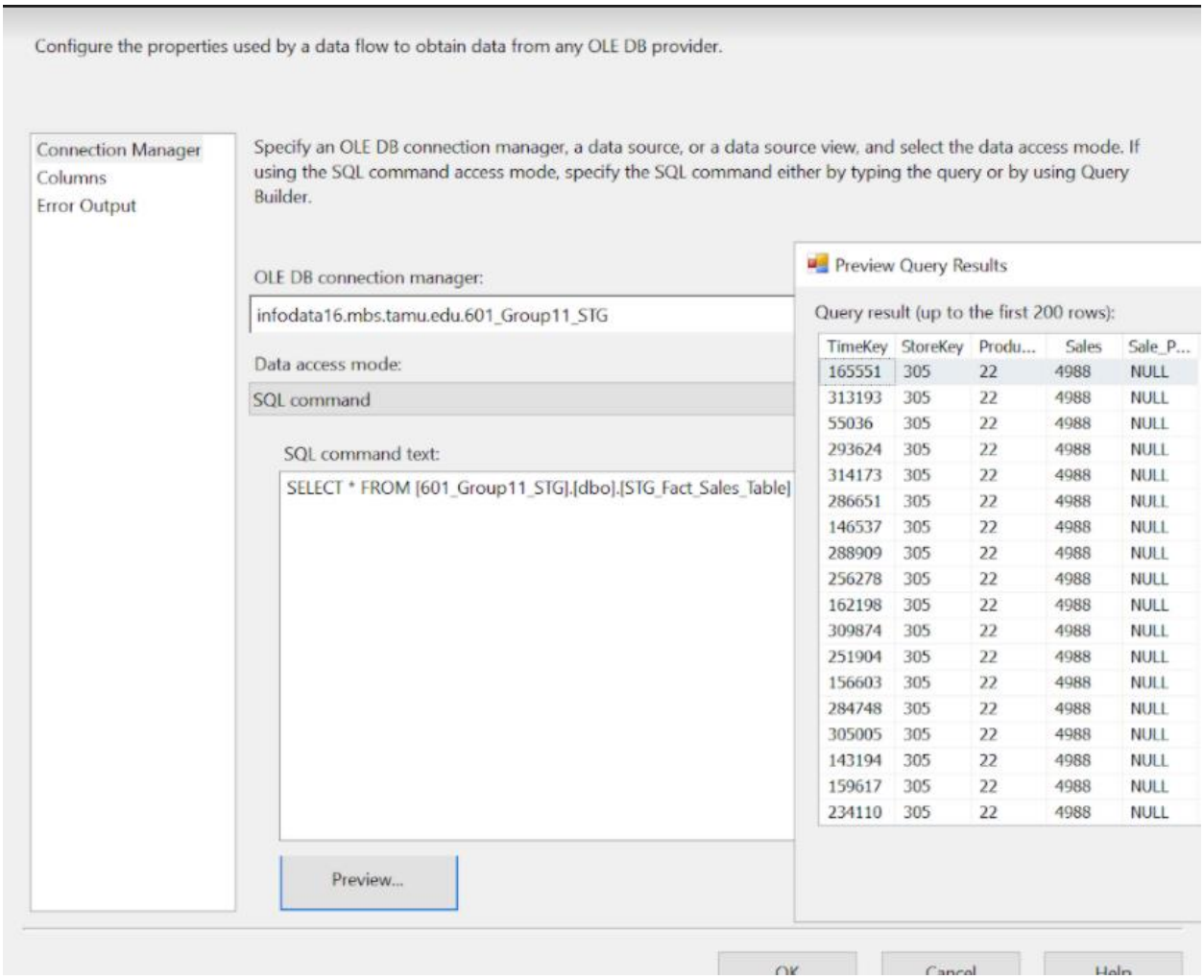

Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If
ry or by using Query
Create Table
CREATE TABLE [Fact_Sales_Table] (
[TimeKey] int,
[StoreKey] varchar(50),
[ProductCategoryKey] int,
[Sale_Promotions] varchar(50),
[Sales] numeric(38,0)
)
OK
Cancel
New...
New...

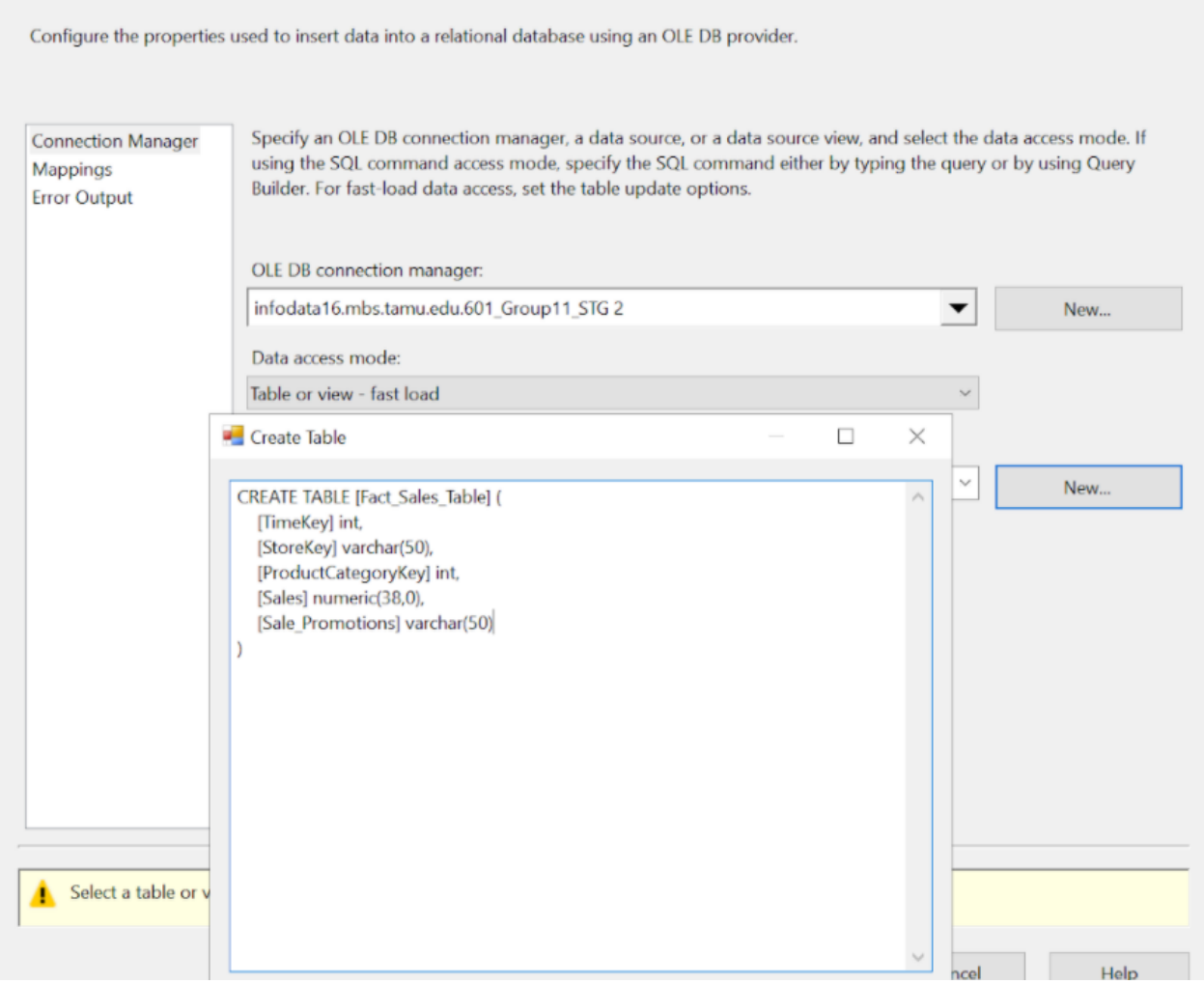

Connection Manager
Mappings
Error Output
Available Input Columns
Name
STORE
SALE_PROMOTIONS
ProductCategoryKey
SALES
TimeKey
Available Destination ...
Name
TimeKey
StoreKey
ProductCategoryKey
Sale_Promotions
Sales
Input Column
Destination Column
TimeKey
TimeKey
<ignore>
StoreKey
ProductCategoryKey
ProductCategoryKey
SALE_PROMOTIONS
Sale_Promotions
SALES
Sales

Control Flow for Sales Fact Table

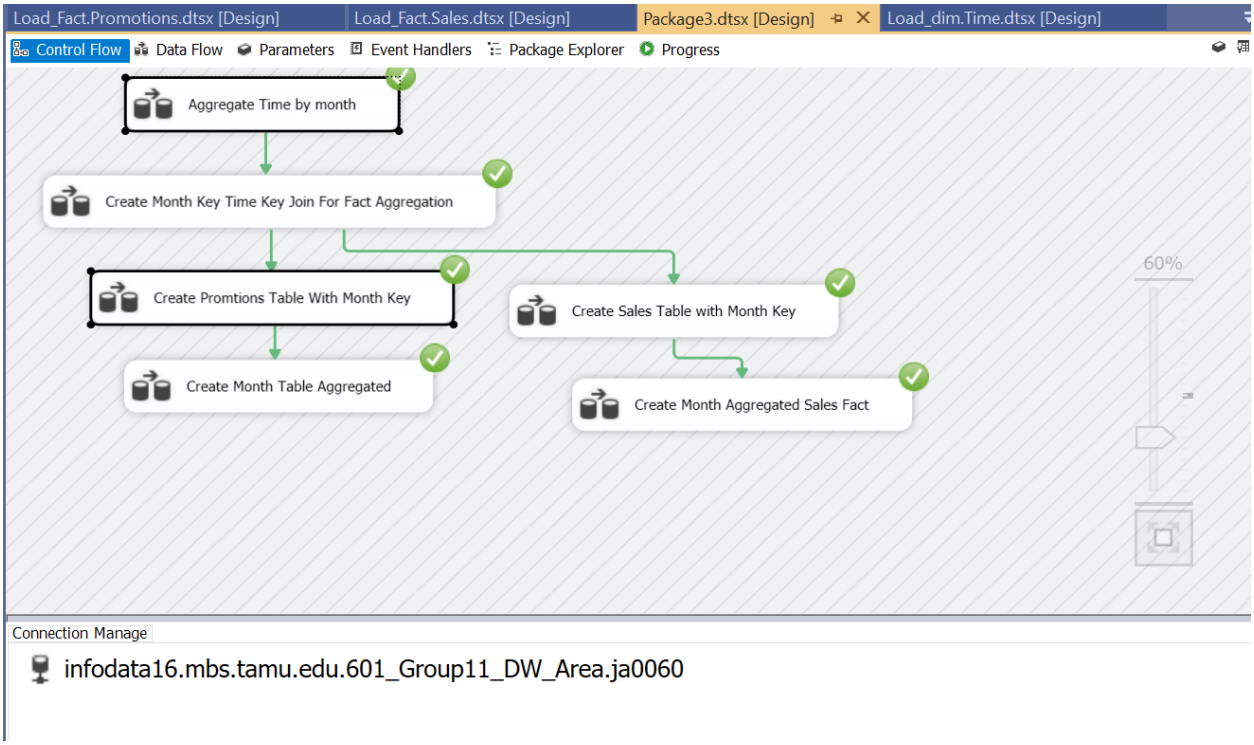


Data Flow for Sales Fact Table

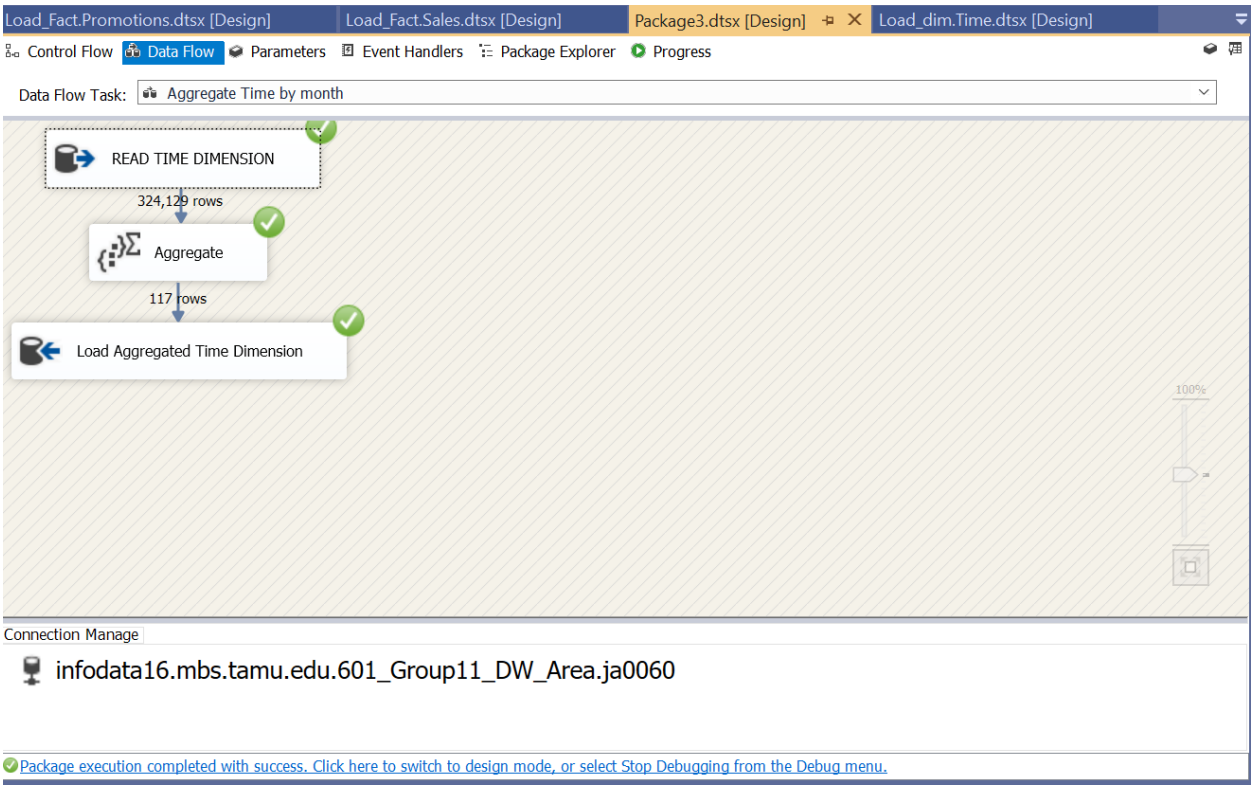

Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG
Data access mode:
SQL command
SQL command text:
SELECT * FROM [601_Group11_STG].[dbo].[STG_Fact_Sales_Table]
Preview...
Preview Query Results
Query result (up to the first 200 rows):
TimeKey StoreKey Produ... Sales Sale_P...
165551 305 22 4988 NULL
313193 305 22 4988 NULL
55036 305 22 4988 NULL
293624 305 22 4988 NULL
314173 305 22 4988 NULL
286651 305 22 4988 NULL
146537 305 22 4988 NULL
288909 305 22 4988 NULL
256278 305 22 4988 NULL
162198 305 22 4988 NULL
309874 305 22 4988 NULL
251904 305 22 4988 NULL
156603 305 22 4988 NULL
284748 305 22 4988 NULL
305005 305 22 4988 NULL
143194 305 22 4988 NULL
159617 305 22 4988 NULL
234110 305 22 4988 NULL
OK
Cancel
Help


Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
infodata16.mbs.tamu.edu.601_Group11_STG 2
New...
Data access mode:
Table or view - fast load
New...
Create Table
CREATE TABLE [Fact_Sales_Table] (
[TimeKey] int,
[StoreKey] varchar(50),
[ProductCategoryKey] int,
[Sales] numeric(38,0),
[Sale_Promotions] varchar(50)
)
Select a table or v
Help

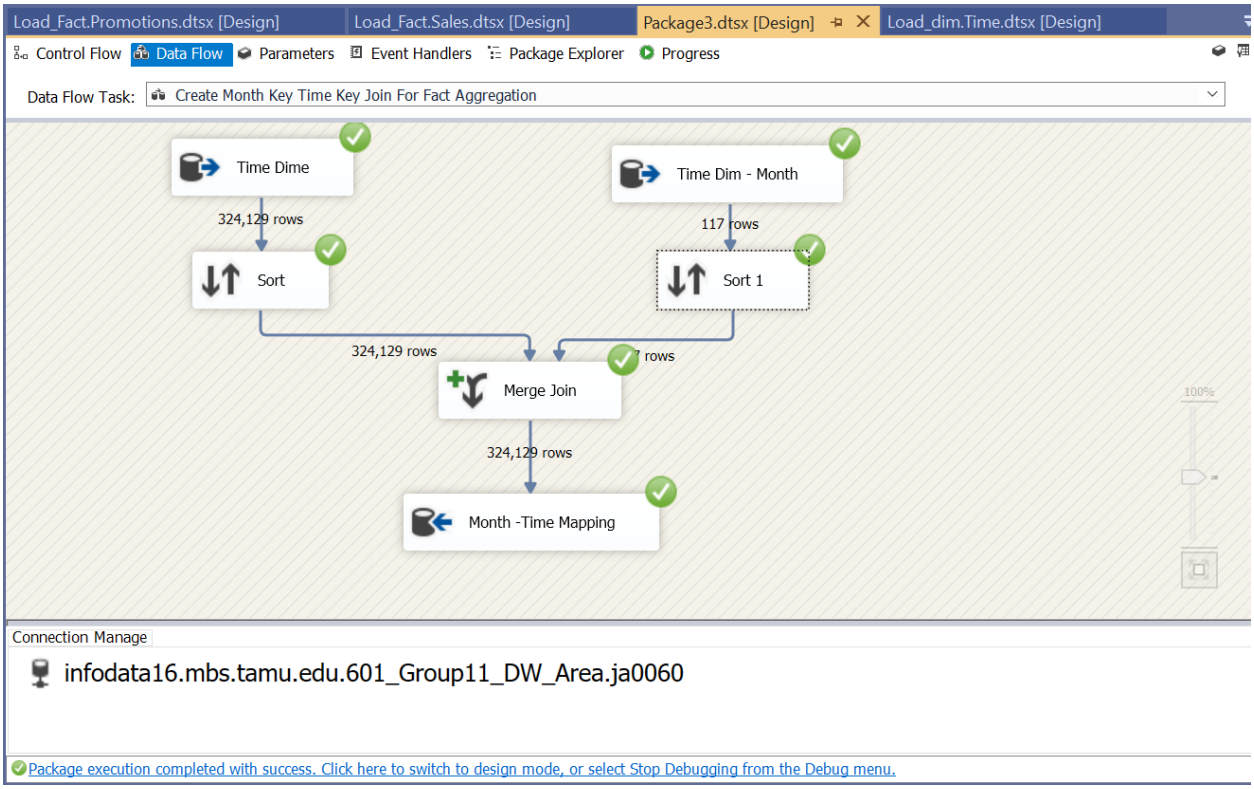
Aggregation of Fact and Dimension tables:



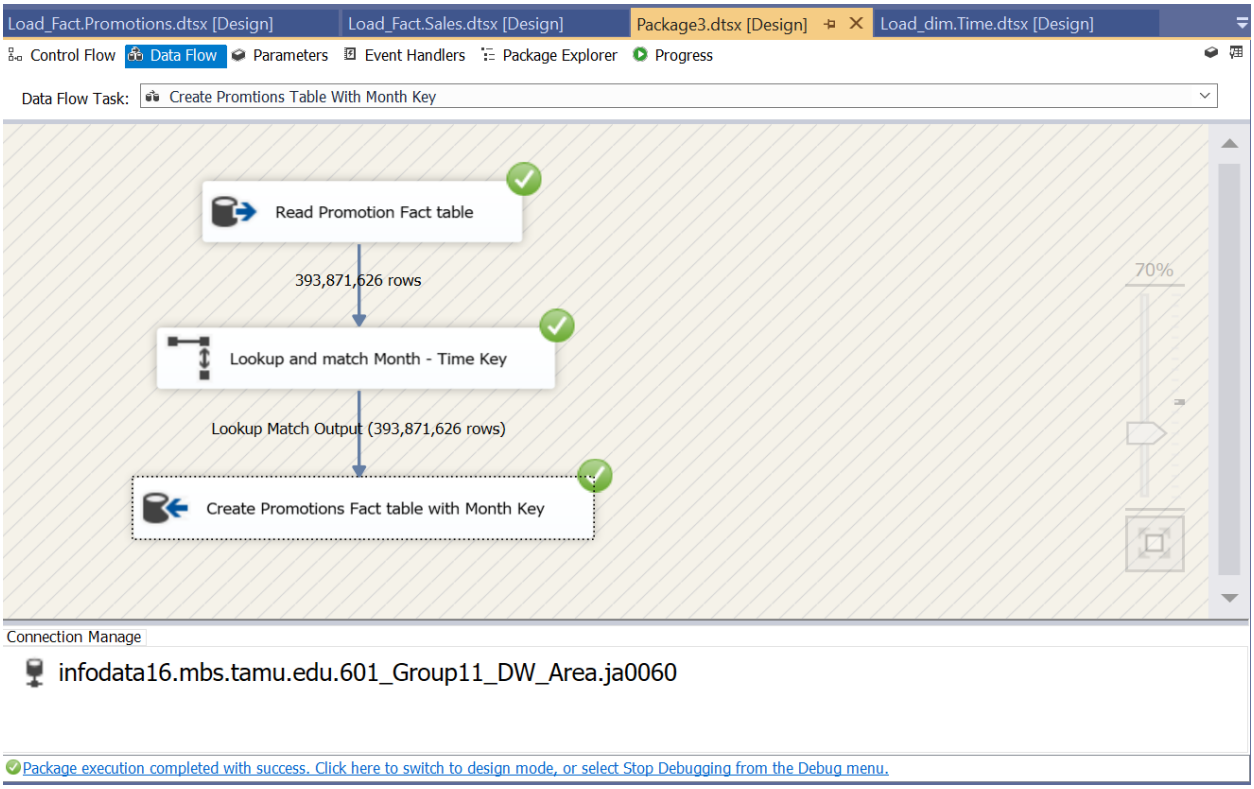
Control Flow for Aggregate Table



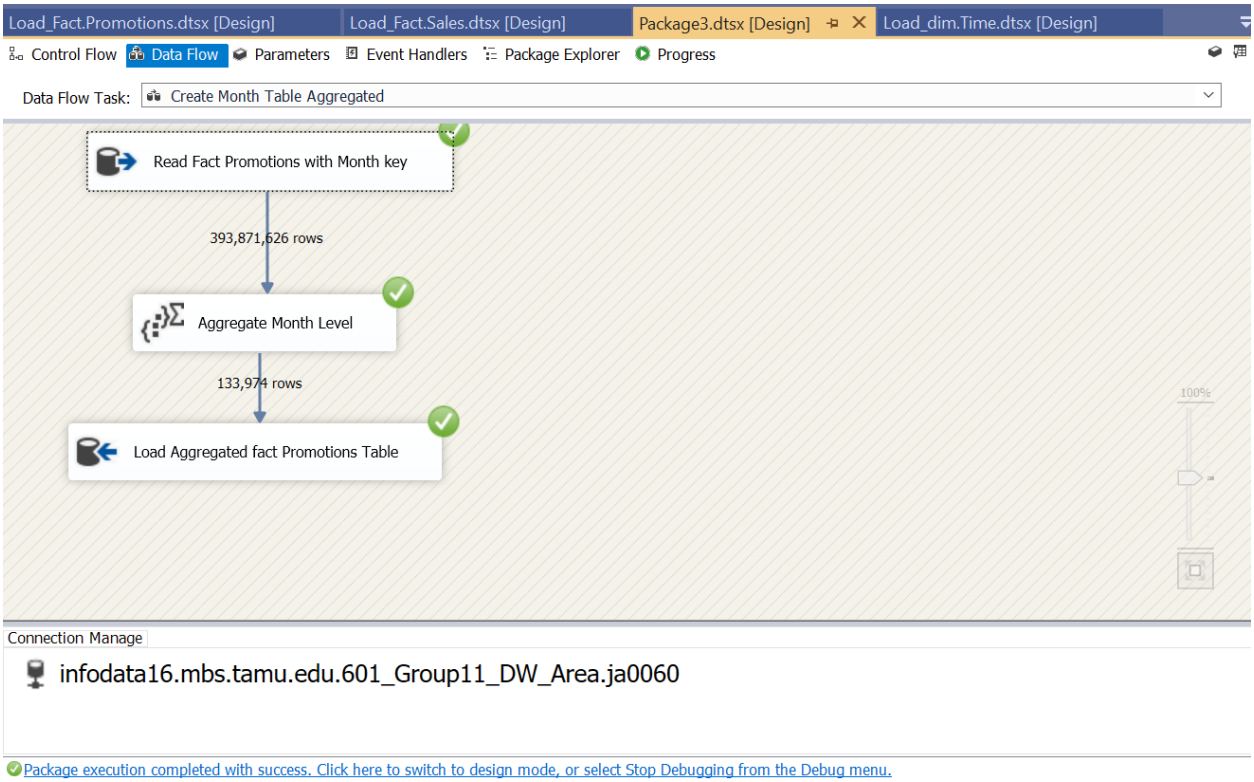
Data Flow for Aggregating the Time Dimension



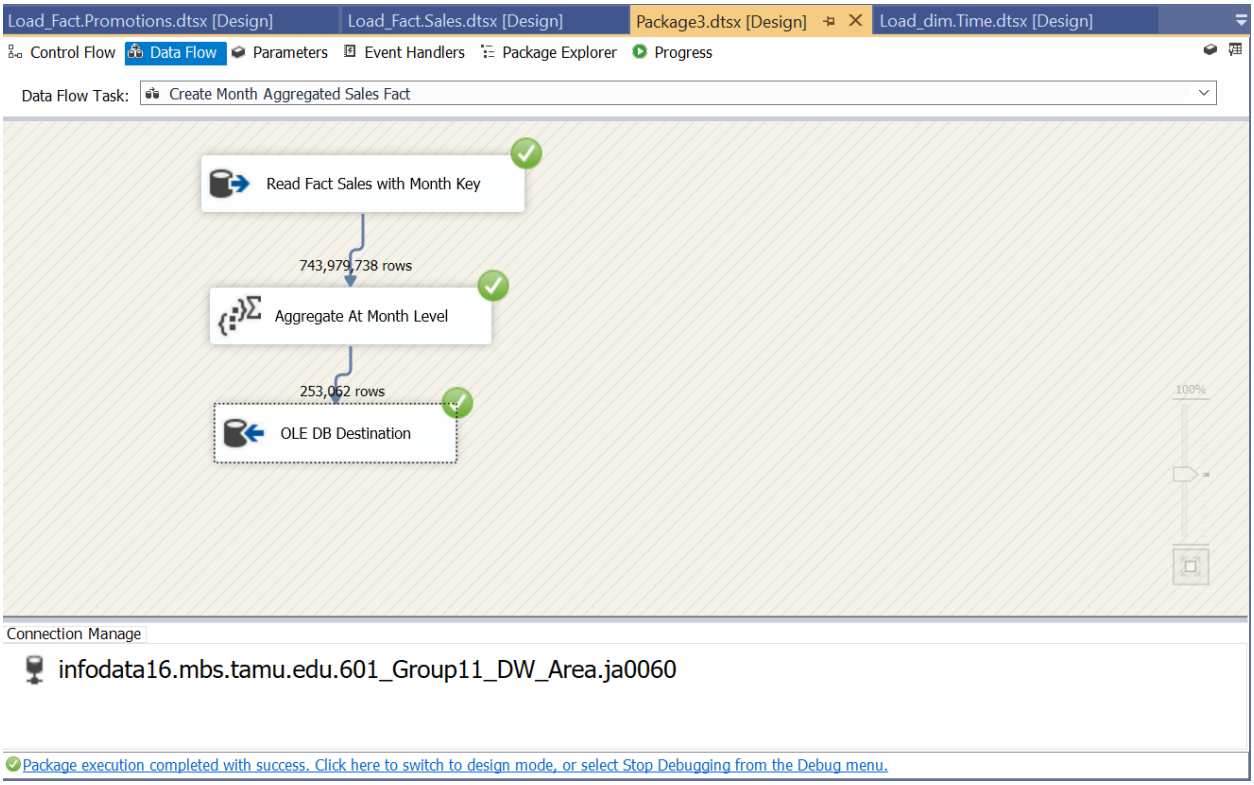
Data Flow for Creating Month Key & Time Key Mapping



Data Flow for adding Surrogate Month Key in Promotions Fact Table



Aggregating Promotion Fact Table with Month Key



Aggregating Sales Fact Table with Month Key

Table Content Snapshots:

```
SELECT TOP (1000) [CouponCategoryKey]
      ,[CategoryName]
  FROM [601_Group11_DW_Area].[dbo].[dim.CouponCategory]
```

110 %

Results | Messages

	CouponCategoryKey	CategoryName
1	1	BAKCOUP
2	2	BULKCOUP
3	3	COSMCOUP
4	4	DAIRCOUP
5	5	DELICOUP
6	6	FLORCOUP
7	7	FROZCOUP
8	8	FTGCCOUP
9	9	FTGICOUP
10	10	GMCOUP
11	11	HABACOUP

```
SELECT TOP (1000) [CategoryName]
      ,[ProductCategoryKey]
  FROM [601_Group11_DW_Area].[dbo].[dim.ProductCategory]
```

10 %

Results | Messages

	CategoryName	ProductCategoryKey
1	BAKERY	1
2	BEER	2
3	BOTTLE	3
4	BULK	4
5	CAMERA	5
6	CHEESE	6
7	CONVFOOD	7
8	COSMETIC	8
9	DAIRY	9
10	DELI	10
11	DELIEXPR	11

```
SELECT TOP (1000) [StoreKey]
      ,[StoreName]
      ,[City]
      ,[Zip]
      ,[WorkWomen]
  FROM [601_Group11_DW_Area].[dbo].[dim.Store]
```

10 %

Results | Messages

	StoreKey	StoreName	City	Zip	WorkWomen
1	100	"DOMINICKS 100"	"CHICAGO"	60608	0.2672339509
2	101	"DOMINICKS 101"	"DES PLAINES"	60016	0.3900651466
3	102	"DOMINICKS 102"	"MERRIONETTE PARK"	60655	0.2899441341
4	103	"DOMINICKS 103"	"BOLINGBROOK"	60439	0.36129428
5	104	"DOMINICKS 104"	"ST CHARLES"	60174	0.405306896
6	105	"DOMINICKS 105"	"MELROSE PARK"	60160	0.3608901188
7	106	"DOMINICKS 106"	"MONTGOMERY"	60538	0.3674829932
8	107	"DOMINICKS 107"	"WESTCHESTER"	60154	0.3112205235
9	109	"DOMINICKS 109"	"BANNOCKBURN"	60015	0.3408066938
10	110	"DOMINICKS 110"	"EAST DUNDEE"	60118	0.4043140865
11	111	"DOMINICKS 111"	"CHICAGO"	60620	0.2885154062
12	112	"DOMINICKS 112"	"BUFFALO GROVE"	60090	0.4401643554

```
SELECT TOP (1000) [TimeKey]
      ,[Month]
      ,[Year]
      ,[Date]
  FROM [601_Group11_DW_Area].[dbo].[dim.Time]
```

110 %

Results | Messages

	TimeKey	Month	Year	Date
1	1	8	1990	1990-08-03
2	2	8	1990	1990-08-04
3	3	8	1990	1990-08-05
4	4	8	1990	1990-08-06
5	5	8	1990	1990-08-07
6	6	8	1990	1990-08-08
7	7	8	1990	1990-08-09
8	8	8	1990	1990-08-10
9	9	8	1990	1990-08-11
10	10	8	1990	1990-08-12
11	11	8	1990	1990-08-13
12	12	8	1990	1990-08-14

```
SELECT TOP (1000) [MonthKey]
      ,[Month]
      ,[Year]
  FROM [601_Group11_DW_Area].[dbo].[dim.Time.Aggregated]
```

10 %

Results | Messages

	MonthKey	Month	Year
1	235	1	1980
2	236	1	1988
3	237	1	1989
4	238	1	1990
5	239	1	1991
6	240	1	1992
7	241	1	1993
8	242	1	1994
9	243	1	1995
10	244	1	1996

```
SELECT TOP (1000) [CouponCategoryKey]
      ,[MonthKey]
      ,[StoreKey]
      ,[CouponRedeemed]
  FROM [601_Group11_DW_Area].[dbo].[Fact_Aggregated_Promotions]
```

110 %

Results | Messages

	CouponCategoryKey	MonthKey	StoreKey	CouponRedeemed
1	14	344	124	0
2	15	280	124	0
3	14	340	134	0
4	14	348	114	0
5	15	284	114	0
6	15	288	104	0
7	14	345	124	0
8	15	281	124	0
9	14	341	134	0
10	14	349	114	0

```
SELECT TOP (1000) [MonthKey]
      ,[StoreKey]
      ,[ProductCategoryKey]
      ,[Sales]
      ,[Sale_Promotions]
  FROM [601_Group11_DW_Area].[dbo].[FACT_AGGREGATED_SAES_TABLE]
```

110 %

Results | Messages

	MonthKey	StoreKey	ProductCategoryKey	Sales	Sale_Promotions
1	251	107	1	5986977	NULL
2	239	137	1	4247757	NULL
3	251	107	2	1167863	404
4	239	137	2	0	NULL
5	251	107	3	505	NULL
6	239	137	3	1616	NULL
7	251	107	4	2733969	NULL
8	239	137	4	1078377	NULL
9	251	107	5	3838	NULL
10	239	137	5	50904	NULL
11	251	107	6	1421979	NULL

```
SELECT TOP (1000) [CouponCategoryKey]
      ,[MonthKey]
      ,[StoreKey]
      ,[CouponRedeemed]
  FROM [601_Group11_DW_Area].[dbo].[Fact_Promotions_Month_Key]
```

110 %

Results | Messages

	CouponCategoryKey	MonthKey	StoreKey	CouponRedeemed
1	6	309	90	0
2	6	309	90	0
3	6	309	90	0
4	6	309	90	0
5	6	309	90	0
6	6	309	90	0
7	6	309	90	0
8	6	309	90	0
9	6	309	90	0
10	6	309	90	0
11	6	309	90	0

```
SELECT TOP (1000) [CouponCategoryKey]
      ,[TimeKey]
      ,[StoreKey]
      ,[CouponRedeemed]
  FROM [601_Group11_DW_Area].[dbo].[Fact_Promotions_Table]
```

110 %

Results | Messages

	CouponCategoryKey	TimeKey	StoreKey	CouponRedeemed
1	15	38866	86	0
2	15	100765	86	0
3	15	117267	86	0
4	15	41385	86	0
5	15	120615	86	0
6	15	135137	86	0
7	15	94218	86	0
8	15	49210	86	0
9	15	138489	86	0
10	15	71489	86	0

```
SELECT TOP (1000) [MonthKey]
      ,[StoreKey]
      ,[ProductCategoryKey]
      ,[Sales]
      ,[Sale_Promotions]
  FROM [601_Group11_DW_Area].[dbo].[FACT_SALES_MONTH_KEY]
```

110 %

Results | Messages

	MonthKey	StoreKey	ProductCategoryKey	Sales	Sale_Promotions
1	290	134	15	2631	NULL
2	290	Click to select the whole column		2631	NULL
3	290	134	15	2631	NULL
4	290	134	15	2631	NULL
5	290	134	15	2631	NULL
6	290	134	15	2631	NULL
7	290	134	15	2631	NULL
8	290	134	15	2631	NULL
9	290	134	15	2631	NULL
10	290	134	15	2631	NULL

```
SELECT TOP (1000) [TimeKey]
      ,[StoreKey]
      ,[ProductCategoryKey]
      ,[Sales]
      ,[Sale_Promotions]
  FROM [601_Group11_DW_Area].[dbo].[Fact_Sales_Table]
```

110 %

Results | Messages

	TimeKey	StoreKey	ProductCategoryKey	Sales	Sale_Promotions
1	154052	129	23	1896	NULL
2	176112	129	23	1896	NULL
3	48023	129	23	1896	NULL
4	179120	129	23	1896	NULL
5	106277	129	23	1896	NULL
6	220331	129	23	1896	NULL
7	197538	129	23	1896	NULL
8	263835	129	23	1896	NULL
9	137302	129	23	1896	NULL
10	199519	129	23	1896	NULL
11	201873	129	23	1896	NULL

```
SELECT TOP (1000) [TimeKey]
      ,[MonthKey]
  FROM [601_Group11_DW_Area].[dbo].[Month_Time_Mapping]
```

110 %

Results | Messages

	TimeKey	MonthKey
1	127499	235
2	17001	236
3	21321	236
4	35795	236
5	40240	236
6	55862	236
7	63579	236
8	73699	236
9	80886	236
10	96285	236
11	102953	236

SQL Queries:

Create table Queries

```
CREATE TABLE [STG_CCount_1_Extract] (
    [STORE] varchar(50),
    [DATE] varchar(50),
    [DERIVED_DATE] varchar(50),
    [Day] varchar(50),
    [GROCERY] varchar(50),
    [DAIRY] varchar(50),
    [FROZEN] varchar(50),
    [BOTTLE] varchar(50),
    [MVPCLUB] varchar(50),
    [GROCCOUP] varchar(50),
    [MEAT] varchar(50),
    [MEATFROZ] varchar(50),
    [MEATCOUP] varchar(50),
    [FISH] varchar(50),
    [FISHCOUP] varchar(50),
    [PROMO] varchar(50),
    [PROMCOUP] varchar(50),
    [PRODUCE] varchar(50),
    [BULK] varchar(50),
    [SALADBAR] varchar(50),
    [PRODCOUP] varchar(50),
    [BULKCOUP] varchar(50),
    [SALCOUP] varchar(50),
    [FLORAL] varchar(50),
    [FLORCOUP] varchar(50),
    [DELI] varchar(50),
    [DELISELF] varchar(50),
    [DELIEXPR] varchar(50),
    [CONVFOOD] varchar(50),
    [CHEESE] varchar(50),
    [DELICOUP] varchar(50),
    [BAKERY] varchar(50),
    [PHARMACY] varchar(50),
    [PHARCOUP] varchar(50),
    [GM] varchar(50),
    [JEWELRY] varchar(50),
    [COSMETIC] varchar(50),
    [HABA] varchar(50),
    [GMCOUP] varchar(50),
    [CAMERA] varchar(50),
```

```
    [PHOTOFIN] varchar(50),
    [VIDEO] varchar(50),
    [VIDEOREN] varchar(50),
    [VIDCOUP] varchar(50),
    [BEER] varchar(50),
    [WINE] varchar(50),
    [SPIRITS] varchar(50),
    [MISCSCP] varchar(50),
    [MANCOUP] varchar(50),
    [CUSTCOUN] varchar(50),
    [FTGCHIN] varchar(50),
    [FTGCCOUP] varchar(50),
    [FTGITAL] varchar(50),
    [FTGICOUP] varchar(50),
    [DAIRCOUP] varchar(50),
    [FROZCOUP] varchar(50),
    [HABACOUP] varchar(50),
    [PHOTCOUP] varchar(50),
    [COSMCOUP] varchar(50),
    [SSDELICP] varchar(50),
    [BAKCOUP] varchar(50),
    [LIQCOUP] varchar(50),
    [WEEK] varchar(50),
    [MONTH_OF_RECORD] varchar(50),
    [YEAR_OF_RECORD] varchar(50),
    [DATE_FORMATTED] date
)

CREATE table [dbo].[STG_Beer_Movement_2_Clean](
        [Store] varchar(50),
        [Week] varchar(50),
        [Sale] varchar(50)
)

CREATE TABLE [STG_dim.CouponCategory] (
    [CouponCategoryKey] INT Not Null Identity(1,1) Primary Key,
    [CategoryName] nvarchar(255)
)

CREATE TABLE [STG_dim.ProductCategory] (
    [ProductCategoryKey] int Identity(1,1) Primary Key,
    [CategoryName] NVARCHAR(255)
)
```

```
CREATE TABLE [STG_dim.Store] (
    [StoreKey] int Primary Key,
    [StoreName] varchar(50),
    [City] varchar(50),
    [Zip] varchar(50),
    [WorkWomen] varchar(50)
)

CREATE TABLE [STG_dim.Time] (
    [TimeKey] int identity(1,1) Primary Key,
    [Month] int,
    [Year] int,
    [Date] date
)

CREATE TABLE [Fact_Promotions_Table] (
    [CouponCategoryKey] int,
    [TimeKey] int,
    [StoreKey] varchar(50),
    [CouponRedeemed] numeric(38,0)
)

CREATE TABLE [STG_Fact_Sales_Table] (
    [TimeKey] int,
    [StoreKey] varchar(50),
    [ProductCategoryKey] int,
    [Sale_Promotions] varchar(50),
    [Sales] numeric(38,0)
)

CREATE TABLE [STG_ProductCategory_1_Transform] (
    [CategoryName] nvarchar(255)
)

CREATE TABLE [STG_Sales_2_Merge] (
    [STORE] varchar(50),
    [GROCERY] numeric(38,0),
    [DAIRY] numeric(38,0),
    [FROZEN] numeric(38,0),
    [BOTTLE] numeric(38,0),
    [MVPCLUB] numeric(38,0),
    [MEAT] numeric(38,0),
    [MEATFROZ] numeric(38,0),
    [FISH] numeric(38,0),
    [PROMO] numeric(38,0),
```

```
    [PRODUCE] numeric(38,0),
    [BULK] numeric(38,0),
    [SALADBAR] numeric(38,0),
    [FLORAL] numeric(38,0),
    [DELI] numeric(38,0),
    [DELISELF] numeric(38,0),
    [DELIEXPR] numeric(38,0),
    [CONVFOOD] numeric(38,0),
    [CHEESE] numeric(38,0),
    [BAKERY] numeric(38,0),
    [PHARMACY] numeric(38,0),
    [GM] numeric(38,0),
    [JEWELRY] numeric(38,0),
    [COSMETIC] numeric(38,0),
    [HABA] numeric(38,0),
    [CAMERA] numeric(38,0),
    [PHOTOFIN] numeric(38,0),
    [VIDEO] numeric(38,0),
    [VIDEOREN] numeric(38,0),
    [BEER] numeric(38,0),
    [WINE] numeric(38,0),
    [SPIRITS] numeric(38,0),
    [MISCSCP] numeric(38,0),
    [CUSTCOUN] numeric(38,0),
    [FTGCHIN] numeric(38,0),
    [FTGITAL] numeric(38,0),
    [SSDELICP] numeric(38,0),
    [MONTH_OF_RECORD] varchar(50),
    [YEAR_OF_RECORD] varchar(50),
    [DATE_FORMATTED] date,
    [WEEK] varchar(50),
    [SALE] varchar(50)
)

CREATE TABLE [STG_Sales_3_Transform] (
    [ProductCategoryName] nvarchar(255),
    [STORE] varchar(50),
    [MONTH_OF_RECORD] varchar(50),
    [YEAR_OF_RECORD] varchar(50),
    [DATE_FORMATTED] date,
    [SALE_PROMOTIONS] varchar(50),
    [SALES] numeric(38,0)
)
```

```
CREATE TABLE [STG_Sales_4_Merge] (
   [STORE] varchar(50),
   [ProductCategoryName] nvarchar(255),
   [SALE_PROMOTIONS] varchar(50),
   [SALES] numeric(38,0),
   [TimeKey] int
)

CREATE TABLE [dim.CouponCategory] (
   [CouponCategoryKey] INT Not Null Identity(1,1) Primary Key,
   [CategoryName] nvarchar(255)
)

CREATE TABLE [dim.ProductCategory] (
   [ProductCategoryKey] int Identity(1,1) Primary Key,
   [CategoryName] NVARCHAR(255)
)

CREATE TABLE [dim.Store] (
   [StoreKey] int Primary Key,
   [StoreName] varchar(50),
   [City] varchar(50),
   [Zip] varchar(50),
   [WorkWomen] varchar(50)
)

CREATE TABLE [dim.Time] (
   [TimeKey] int identity(1,1) Primary Key,
   [Month] int,
   [Year] int,
   [Date] date
)

CREATE TABLE [Fact_Promotions_Table] (
   [CouponCategoryKey] int,
   [TimeKey] int,
   [StoreKey] varchar(50),
   [CouponRedeemed] numeric(38,0)
)

CREATE TABLE [Fact_Sales_Aggregate] (
   [MonthKey] int,
   [StoreKey] varchar(50),
   [ProductCategoryKey] int,
   [Sales] numeric(38,0)
```

```
)

CREATE TABLE [dbo].[Fact_Aggregated_Promotions] (
   [CouponCategoryKey] int,
   [MonthKey] int,
   [StoreKey] varchar(50),
   [CouponRedeemed] numeric(38,0)
)
CREATE TABLE [Fact_Sales_Table] (
   [TimeKey] int,
   [StoreKey] varchar(50),
   [ProductCategoryKey] int,
   [Sale_Promotions] varchar(50),
   [Sales] numeric(38,0)
)
```

Alter Queries for Cleaning , Transformation and other modification:

```
DELETE FROM
[dbo].[STG_CCount_1_Extract]
 WHERE [DERIVED_DATE] LIKE '%[a-Z]%';

DELETE FROM
[dbo].[STG_CCount_1_Extract]
WHERE [DERIVED_DATE] NOT LIKE '%[1-9]%';

UPDATE
[dbo].[STG_CCount_1_Extract]
SET DATE_FORMATTED = CONVERT(VARCHAR,CAST([DERIVED_DATE] AS
DATE),1)

UPDATE
[dbo].[STG_CCount_1_Extract]
SET MONTH_OF_RECORD=MONTH(DATE_FORMATTED);

UPDATE
[dbo].[STG_CCount_1_Extract]
SET YEAR_OF_RECORD =YEAR(DATE_FORMATTED);

UPDATE [dbo].[STG_CCount_2_Clean] SET     [BAKCOUP] = 0 WHERE    [BAKCOUP] <0 ;
UPDATE [dbo].[STG_CCount_2_Clean] SET     [BAKERY] = 0 WHERE    [BAKERY] <0 ;
(Likewise for all coloumns)
```

```
DELETE FROM [dbo].[STG_CCount_2_Clean] WHERE [WEEK] IS NULL;
DELETE FROM [dbo].[STG_CCount_2_Clean] WHERE [WEEK] = '.';
DELETE FROM [dbo].[STG_CCount_2_Clean] WHERE [WEEK] like '-%';

DELETE
[STG_DEMO_1_EXTRACT]
WHERE [NAME]="";
```

Adding Constraints

```
ALTER TABLE [601_Group11_DW_Area].[dbo].[Fact_Promotions_Table]
ADD CONSTRAINT FK_Promotions_Time
FOREIGN KEY(TimeKey) REFERENCES
[601_Group11_DW_Area].[dbo].[dim.Time](TimeKey);

ALTER TABLE [601_Group11_DW_Area].[dbo].[Fact_Sales_Table]
ADD CONSTRAINT FK_Sales_Time
FOREIGN KEY(TimeKey) REFERENCES
[601_Group11_DW_Area].[dbo].[dim.Time](TimeKey);

ALTER TABLE [601_Group11_DW_Area].[dbo].[Fact_Sales_Table]
ADD CONSTRAINT FK_SSK
FOREIGN KEY (StoreKey) REFERENCES
[601_Group11_DW_Area].[dbo].[dim.Store](StoreKey);

ALTER TABLE [601_Group11_DW_Area].[dbo].[Fact_Promotions_Table]
ADD CONSTRAINT FK_PSK
FOREIGN KEY(StoreKey) REFERENCES
[601_Group11_DW_Area].[dbo].[dim.Store](StoreKey);

ALTER TABLE [601_Group11_DW_Area].[dbo].[Fact_Sales_Table]
ADD CONSTRAINT FK_Sales_Category
FOREIGN KEY (ProductCategoryKey) REFERENCES
[601_Group11_DW_Area].[dbo].[dim.ProductCategory](ProductCategoryKey);

ALTER TABLE [601_Group11_DW_Area].[dbo].[Fact_Promotions_Table]
ADD CONSTRAINT FK_Promotions_Category
FOREIGN KEY(CouponCategoryKey) REFERENCES
[601_Group11_DW_Area].[dbo].[dim.CouponCategory](CouponCategoryKey);
```

List of Temp Tables deleted:

Sr. no.	Name of Temp Tables
1	STG_Temp_CCount_Clean_Verify
2	STG_Temp_Demo_Store_Mapping
3	STG_Temp_Movement_Beer_Modeling
4	STG_Temp_Month_Time_Validation
5	STG_Temp_Aggregate_Mapping
6	STG_Temp_ProductCategory_Reduction
7	STG_Temp_CouponCategory_Reduction

References:

[1] https://www.chicagobooth.edu/research/kilts/datasets/dominicks

[2] Huang, Tao, Robert Fildes, and Didier Soopramanien. "The Value of Competitive Information in Forecasting FMCG Retail Product Sales and the Variable Selection Problem." European Journal of Operational Research (2014).

[3] Levy, Daniel, Avichai Snir, Alex Gotler, and Haipeng (Allan) Chen. "Not All Price Endings Are Created Equal: Price Points and Asymmetric Price Rigidity." *Journal of Monetary Economics* 110 (2020): 33-49.

[4] Jami, Ata and Himanshu Mishra. "Downsizing and Supersizing: How Changes in Product Attributes Influence Consumer Preferences." Journal of Behavioral Decision Making (2013).

[5] , "Why Does the Average Price of Tuna Fall during Lent?" (2005) [5], Nevo, Aviv, and Hatzitaskos, Konstantinos
https://ideas.repec.org/p/nbr/nberwo/11572.html

[6] https://www.chicagobooth.edu/-/media/enterprise/centers/kilts/datasets/dominicks-dataset/dominicks-manual-and-codebook_kiltscenter.aspx

[7] Ponniah, P. (2010). Data Warehousing Fundamentals for IT Professionals: A Comprehensive Guide for IT Professionals. Hoboken: John Wiley & Sons.

[8] Levy, Daniel, Haipeng (Allan) Chen, Georg Muller, Shantanu Dutta, and Mark Bergen, "Holiday Price Rigidity and Cost of Price Adjustment," Economica 77 (2010): 172-198.

[9] Huang, Tao, Robert Fildes, and Didier Soopramanien. "The Value of Competitive Information in Forecasting FMCG Retail Product Sales and the Variable Selection Problem." European Journal of Operational Research (2014)

Team Task Sheet

Task	Contributor	Time
Introduction and Domain Knowledge	Jain, Nilesh	8 hours
Data Description	Sachdeva, Aakash	5 hours
ERD	Porwal, Priyanshu	3 hours
Business Question	Jain, Nilesh (3), Sachdeva, Aakash(3), Porwal, Priyanshu (4)	20 hours
Logical Design	Jain, Nilesh	2 hours
Requirement Prioritization	Jain, Nilesh	2 hours
Information Package:	Jain, Nilesh	2 hours
Data Mart Matrix	Jain, Nilesh	2 hours
Conceptual Design	Jain, Nilesh	3 hours
Dimension Table Details	Porwal, Priyanshu	1.5 hours
Fact Table Details:	Porwal, Priyanshu	1.5 hours
Star Schema	Porwal, Priyanshu	2 hours
Business Question Justification at Data Mart Level	Porwal, Priyanshu	2 hours
Data Mapping Table	Porwal, Priyanshu	3 hours
Physical Design Plan	Porwal, Priyanshu	1.5 hours
Standardization Plan	Sachdeva, Aakash	1.5 hours
Aggregate Plan	Sachdeva, Aakash	2 hours
Indexing Plan	Sachdeva, Aakash	2 hours
Storage Plan	Sachdeva, Aakash	2 hours
Physical Model	Sachdeva, Aakash	2 hours

Data Sources	Sachdeva, Aakash	2 hours
ETL Plan	Jain, Nilesh	4 hours
ETL Plan	Porwal, Priyanshu	4 hours
ETL Plan	Sachdeva, Aakash	4 hours
ETL Implementation	Jain, Nilesh	8 hours
ETL Implementation	Porwal, Priyanshu	8 hours
ETL Implementation	Sachdeva, Aakash	8 hours
Aggregation, Documentation	Jain, Nilesh	3 hours
Documentation	Porwal, Priyanshu	3 hours
Documentation	Sachdeva, Aakash	3 hours

Signature:

1. Jain, Nilesh
2. Porwal, Priyanshu
3. Sachdeva, Aakash