VIETNAMESE-GERMAN UNIVERSITY

Current Topic in Computer Science

Group 4 Report

Promotion and clustering

Instructor:	Dr. Phan Trọng Nhân
Students:	Trần Kim Hoàn (18810)
	Nguyễn Quốc Huy (17409)
	Nguyễn Khắc Hoàng (18230)
	Lê Vân Khánh (15872)

JANUARY 2024

Member list & Workload

Our group has 4 members and we together decided to split the total workload into proper parts equally. Each member is responsible for the work as below:

No.	Full Name	ID	Percentage of contribution
1	Trần Kim Hoàn	18810	100%
2	Nguyễn Quốc Huy	17409	100%
3	Nguyễn Khắc Hoàng	18230	100%
4	Lê Vân Khánh	15872	20%

- Trần Kim Hoàn
  - Analysis data source
  - Draw star schema diagram
  - Design data warehouse
  - Apply data mining method
  - Design BI dashboard
  - Write and finalize report
- Nguyễn Quốc Huy
  - Analysis data source
  - Design data warehouse
  - Implement ETL pipeline with incremental loading and (near) real-time
  - Apply data mining method
  - Write report
- Nguyễn Khắc Hoàng
  - Analysis data source
  - Design data warehouse
  - Draw star schema diagram
  - Apply data mining method and finalize
  - Design BI dashboard and finalize visualization
  - Write report
- Lê Vân Khánh
  - Design slides for presentation

Table of contents

I. Introduction

In the rapidly evolving business environment, Business Intelligence (BI) and Analytics have emerged as important tools to gain competitive advantage. This report explores the application of BI and Analytics in the context of CompanyX, a retailer specializing in bicycles, bicycle components, accessories, and bicycle clothing. The focus is on the use of advertising strategies and clustering techniques to improve business efficiency.

II. Topic investigation and Data source analysis

1. Topic investigation

Clustering-Based Recommender Systems

In their paper, "Review of Clustering-Based Recommender Systems," Beregovskaya and Koroteev provide a comprehensive overview of modern approaches to recommender system design using clustering. They highlight how clustering can address several known issues in recommendation systems, such as increasing the diversity, consistency, and reliability of recommendations; the data sparsity of user-preference matrices; and changes in user preferences over time. The authors discuss various clustering algorithms, including K-means, hierarchical clustering, and DBSCAN, and their applications in recommender systems. They also present a comparative analysis of these algorithms, discussing their strengths and weaknesses. This paper provides valuable insights for researchers and practitioners in the field, suggesting that clustering can significantly improve the performance of recommender systems by addressing some of their inherent challenges.

In the context of our project, these techniques can be used to segregate our data into distinct performance-based clusters. Each cluster can be formed based on total sales and total quantities, which will represent the performance of the products. The clustering process can help identify common characteristics and trends of the territories, products, quarters, and years within each cluster. These insights can then be used to develop strategic recommendations for applying special offers to specific territories, products, quarters, and years, thereby enhancing the strategic decision-making process at CompanyX.

User Profile Clustering in Collaborative Filtering

Braak et al. in their paper "Improving the Performance of Collaborative Filtering Recommender Systems through User Profile Clustering" discuss how user profile clustering can improve the performance of collaborative filtering recommender systems. The authors propose a method that clusters users based on their profiles, which are then used to generate more accurate recommendations. The authors argue that this approach can address the data sparsity problem in collaborative filtering and improve the diversity of recommendations.

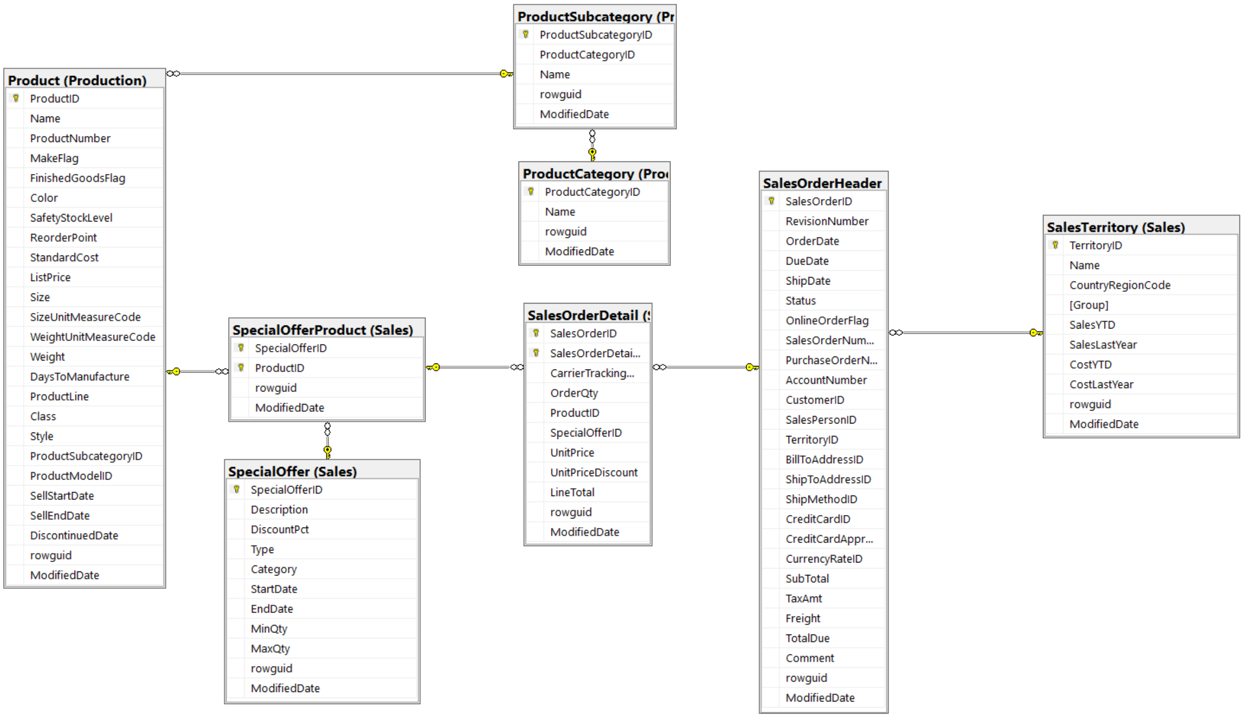
In the context of our project, each cluster can be considered as a "user profile". These profiles can be thoroughly examined to identify common characteristics and trends. This approach can address the data sparsity problem, which is a common issue in many data-driven projects. By understanding the characteristics of each cluster (user profile), you can develop more accurate and effective strategies for applying special offers. This will not only improve the performance of your recommender system but also contribute to the overall sales strategy of CompanyX.

2. Data source analysis

An analysis of data in the database of CompanyX is conducted. After the investigation, not all of the tables and their information are crucial and necessary for the data warehouse and analysis stage. However, there are some useful tables that can be added to the data warehouse and some useful information related to promotion:

- Production.Product provides a table containing the name, CategoryID, SubCategoryID, and other relevant information about the products in the CompanyX.
- Production.ProductCategory provides a table containing the main categories of the products and SubCategoryID.
- Production.ProductionSubCategory provides a table containing information about the subcategories of the product.
- Sales.SpecialOffer provides the information about the type of the promotion. In other words, it provides the name and description of different types of special offers of the CompanyX
- Sales.SpecialOfferProduct describes the relationship between a product and its special offer.
- Sales.SalesTerritory provides information about the countries or their locations.
- Sales.SalesOrderHeader gives the general information about the receipt of the customer including OrderDate, SubTotal, …
- Sales.SalesOrderDetail gives the specific information about the receipt including the OrderQty, ProductID, SpecialOfferID, …

Below is the original diagram for those tables:


ProductSubcategory (Pr
ProductSubcategoryID
ProductCategoryID
Name
rowguid
ModifiedDate

Product (Production)
ProductID
Name
ProductNumber
MakeFlag
FinishedGoodsFlag
Color
SafetyStockLevel
ReorderPoint
StandardCost
ListPrice
Size
SizeUnitMeasureCode
WeightUnitMeasureCode
Weight
DaysToManufacture
ProductLine
Class
Style
ProductSubcategoryID
ProductModelID
SellStartDate
SellEndDate
DiscontinuedDate
rowguid
ModifiedDate

ProductCategory (Pro
ProductCategoryID
Name
rowguid
ModifiedDate

SalesOrderHeader
SalesOrderID
RevisionNumber
OrderDate
DueDate
ShipDate
Status
OnlineOrderFlag
SalesOrderNum...
PurchaseOrderN...
AccountNumber
CustomerID
SalesPersonID
TerritoryID
BillToAddressID
ShipToAddressID
ShipMethodID
CreditCardID
CreditCardAppr...
CurrencyRateID
SubTotal
TaxAmt
Freight
TotalDue
Comment
rowguid
ModifiedDate

SalesTerritory (Sales)
TerritoryID
Name
CountryRegionCode
[Group]
SalesYTD
SalesLastYear
CostYTD
CostLastYear
rowguid
ModifiedDate

SpecialOfferProduct (Sales)
SpecialOfferID
ProductID
rowguid
ModifiedDate

SalesOrderDetail (S
SalesOrderID
SalesOrderDetai...
CarrierTracking...
OrderQty
ProductID
SpecialOfferID
UnitPrice
UnitPriceDiscount
LineTotal
rowguid
ModifiedDate

SpecialOffer (Sales)
SpecialOfferID
Description
DiscountPct
Type
Category
StartDate
EndDate
MinQty
MaxQty
rowguid
ModifiedDate

III. Problem Statement

Company Background: CompanyX is an enterprise specializing in the sale of bicycles, bicycle components, accessories, and bicycle clothing.

Problem: CompanyX, operating in the competitive bicycle and accessories market, faces challenges in devising effective promotional strategies for different products, territories, and time periods. This struggle, stemming from a lack of systematic analysis, impacts their sales and market presence.

Solution: The proposed solution is to develop a business intelligence system. This system will analyze the effectiveness of promotions, thereby providing valuable insights that can enhance strategic planning and decision-making.

Problem Statement: The lack of effective promotional strategies is hindering CompanyX's sales growth and market presence. The proposed business intelligence system aims to rectify this by analyzing the impact of promotions on sales, thereby aiding in the formulation of more effective strategies.

IV. Project objectives

Our project aims to gather and preprocess data related to quarters, years, territories, special offers, and product names from CompanyX, along with their corresponding total sales and total quantities. This data will then be subjected to a clustering analysis to segregate it into distinct performance-based clusters. The performance will be determined based on the total sales and total quantities associated with the product. Each cluster will be thoroughly examined to identify common characteristics and trends of the territories, products, quarters, and years. Based on these insights, strategic recommendations will be developed, suggesting which special offers could be applied to specific territories, products, quarters, and years. The effectiveness of these strategies will be continually assessed and refined as necessary, ensuring optimal results and contributing to the strategic decision-making process at CompanyX.

VI. Key definitions

Low-Performance Cluster: This could represent the cluster with the lowest total quantities and lowest total sales.

Moderate Performance Cluster: This could be the cluster with moderate total quantities and sales, performing better than the low-performance cluster but not as well as the high-performance clusters.

High Performance Cluster: This could denote the cluster with high total quantities and sales, but not the highest.

Top Performance Cluster: This could represent the cluster with the highest total quantities and highest total sales, indicating the group that contributes the most.

VII. Methodology

1. Data Warehouse

The data warehouse is designed based on the Top-Down Design Approach. The purpose of this data warehouse is to analyze which kind of special offer is suitable for each product, quarter, and country to bring out the most revenue for the company.

Based on the data source analysis, the star schema for the data warehouse contains four dimensions related to four pieces of information: Date, Product Category, Special Offer, and Territory. In the center, there will be a sales fact table comprising metrics of Total Quantity, Total Sales, and Total Discount. Below is the Star Schema of SalesFact:

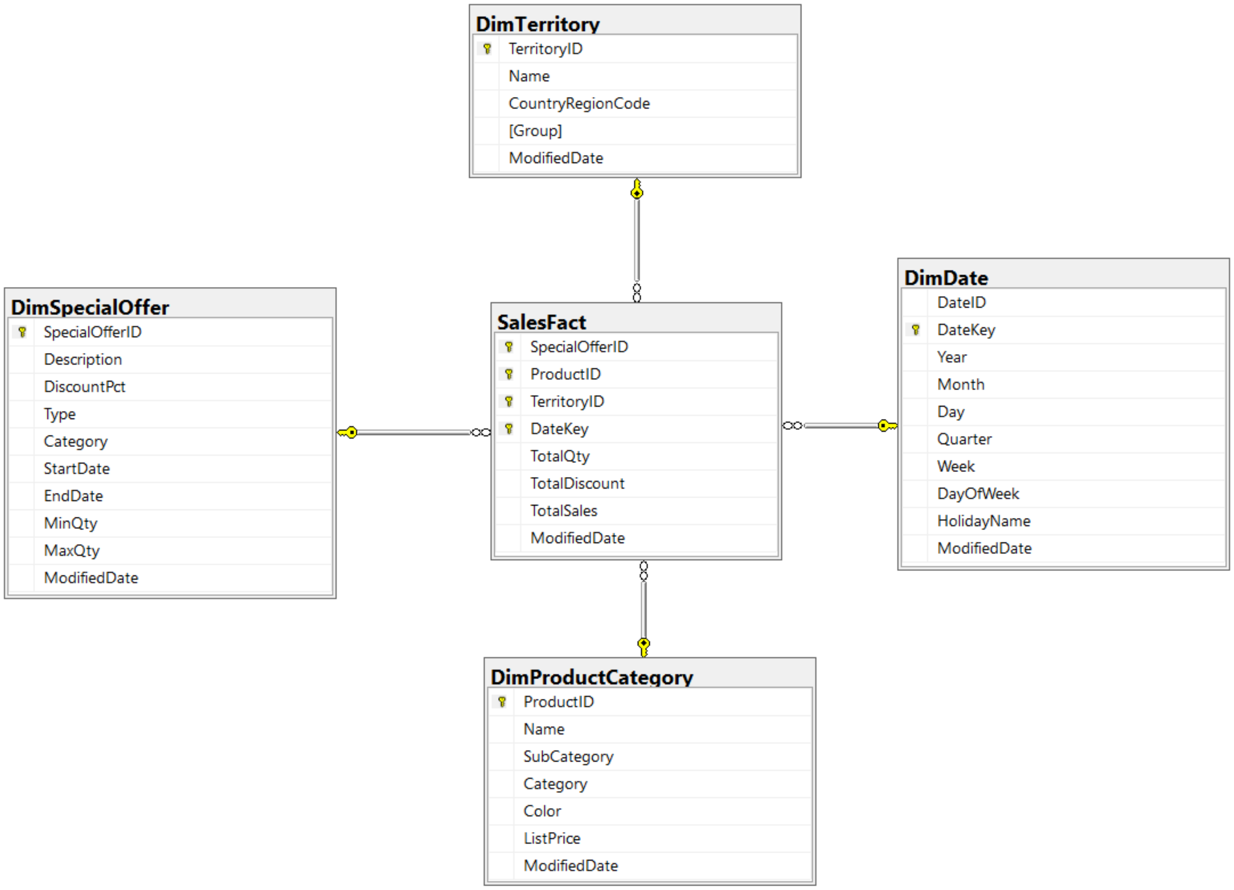


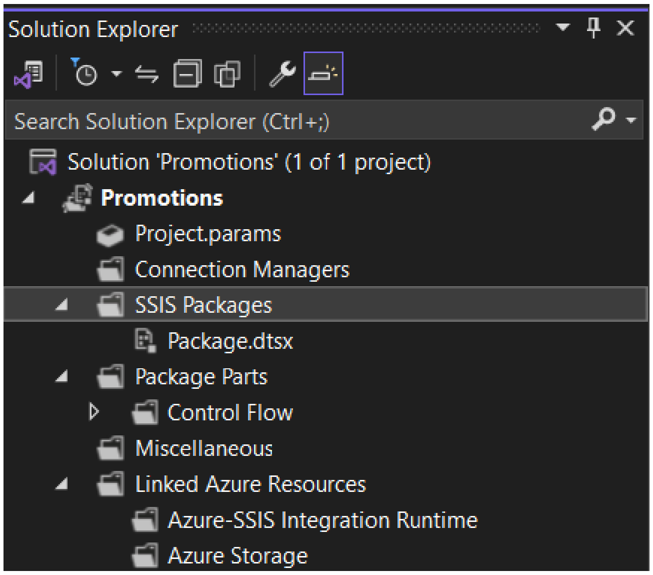
Figure: Star Schema of SalesFact

Further information about the dimensions and sales fact table will be provided as follows:

- Dim_Date dimension contains the primary key named DateKey of the Date table created by hand in the database and other attributes related to the time such as Year, Quarter,… Moreover, there is a surrogate key named DateID.
- Dim_Territory Dimension contains the primary key named TerritoryID of the Sales.SalesTerritory table and other attributes related to the identification of the territory such as Name, Country Region Code, …
- Dim_SpecialOffer dimension contains the primary key names SpecialOfferID of the Sales.SpecialOffer and other attributes related to the promotion offered by the company such as Description and the active period, …
- Dim_ProductCategory dimension contains the combined information of three tables: Production.Product, Production.ProductSubCategory and Production.ProductCategory. The primary key of this dimension is the primary key of the Production.Product table, which is ProductID. This dimension contains the product name, subcategories, categories and other information about the product.
- SalesFact table contains the primary keys of all dimensions in the star schema. The metric of the fact table including Total Quantity, Total Discount and Total Sales will be computed by the information from the table Sales.SalesOrderHeader, Sales.SalesOrderDetail, dbo.DateData and Production.Product.

2. ETL Pipeline

The data warehouse uses the ETL pipeline to extract, transform and load the data from the database of the CompanyX to the data warehouse DWHPromotion. Specifically, the ETL pipeline for the data warehouse comprises a SSIS package named: Package.dtsx.

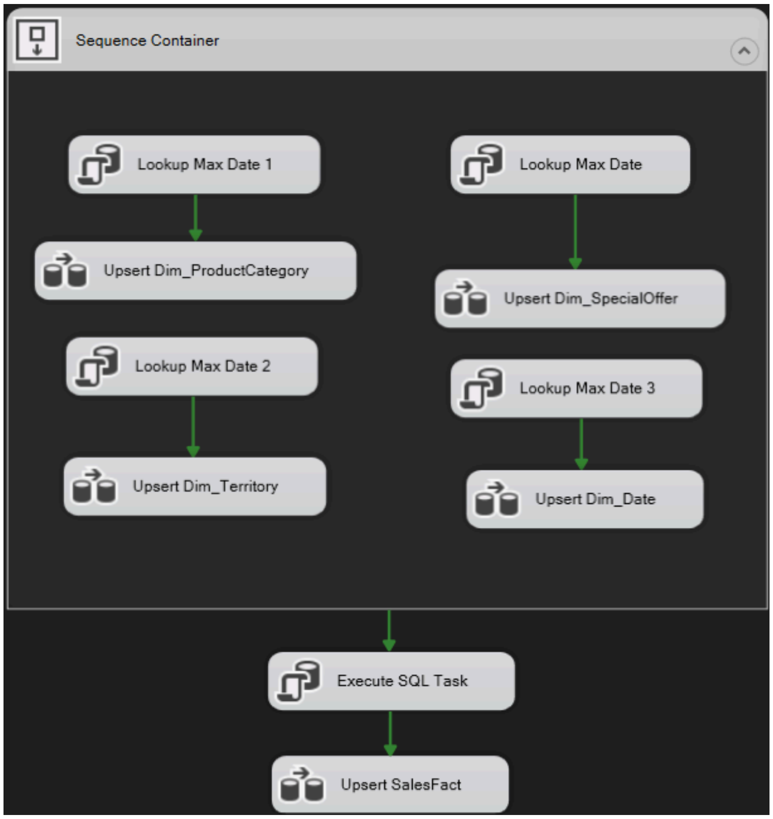


Solution Explorer

Search Solution Explorer (Ctrl+;)

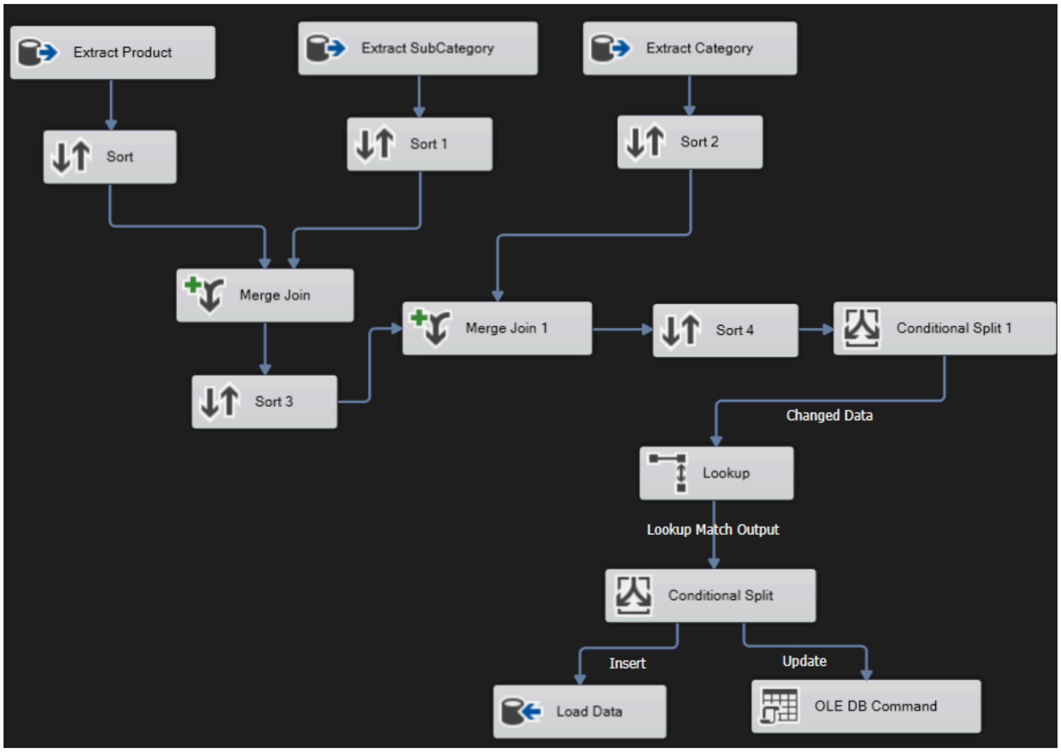
- Solution 'Promotions' (1 of 1 project)
  - **Promotions**
    - Project.params
    - Connection Managers
    - SSIS Packages
      - Package.dtsx
    - Package Parts
      - Control Flow
    - Miscellaneous
    - Linked Azure Resources
      - Azure-SSIS Integration Runtime
      - Azure Storage

Based on the designed star schema of the data warehouse, there will be a control flow consisting of five data flows for four dimensions: DimDate, DimTerritory, DimSpecialOffer, DimProductCategory, and one fact table: SalesFact.



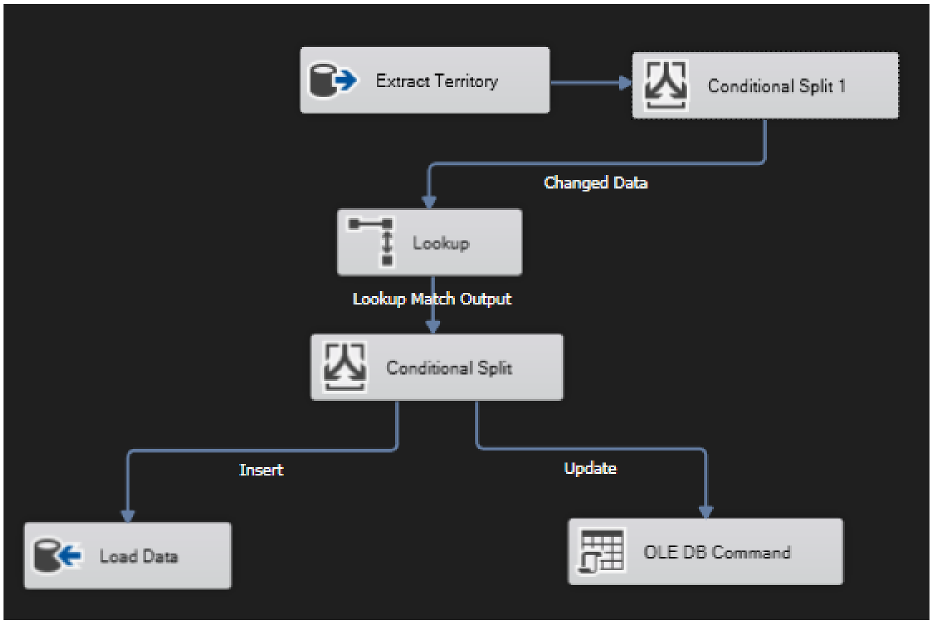
As can be seen from the architecture above, the upsert data flow of the four dimensions will be put into a sequence container to complete first. After the work for the sequence container is done successfully, it will run the data flow for the SalesFact table. Moreover, each data flow will have a corresponding Execute SQL Task to select the largest date in the table to perform a comparison to update or insert specific rows. This implementation will guarantee the incremental loading feature of the ETL pipeline.

Let’s look into each data flow for more information about the implementation and the ETL pipeline:



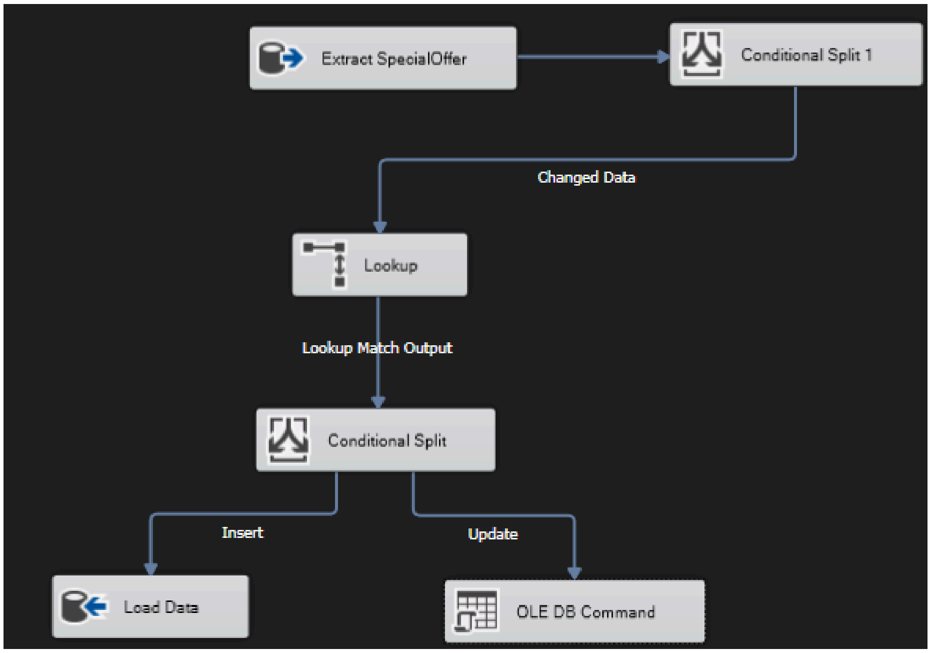
The Product Category Dimension will have the information needed for the data warehouse, which will be combined, extracted and transformed from three tables: Production.Product, Production.ProductSubCategory and Production.ProductCategory before loading the data into the Product Category Dimension. After the ETL process, the Conditional Split 1 will look up the largest date in the ProductCategory Dimension table. If the modified date in each row in the Production.Product table is not larger than the largest modified date, it will do nothing. Otherwise, it will be passed through to the second conditional split. This one will check whether this row already existed or not. If it already existed, it will update the changed rows. Otherwise, it will insert the changed row into the dimension table.

Similarly, the other dimension can be done with the same implementation. The next dimension will be looked into is the Territory Dimension.



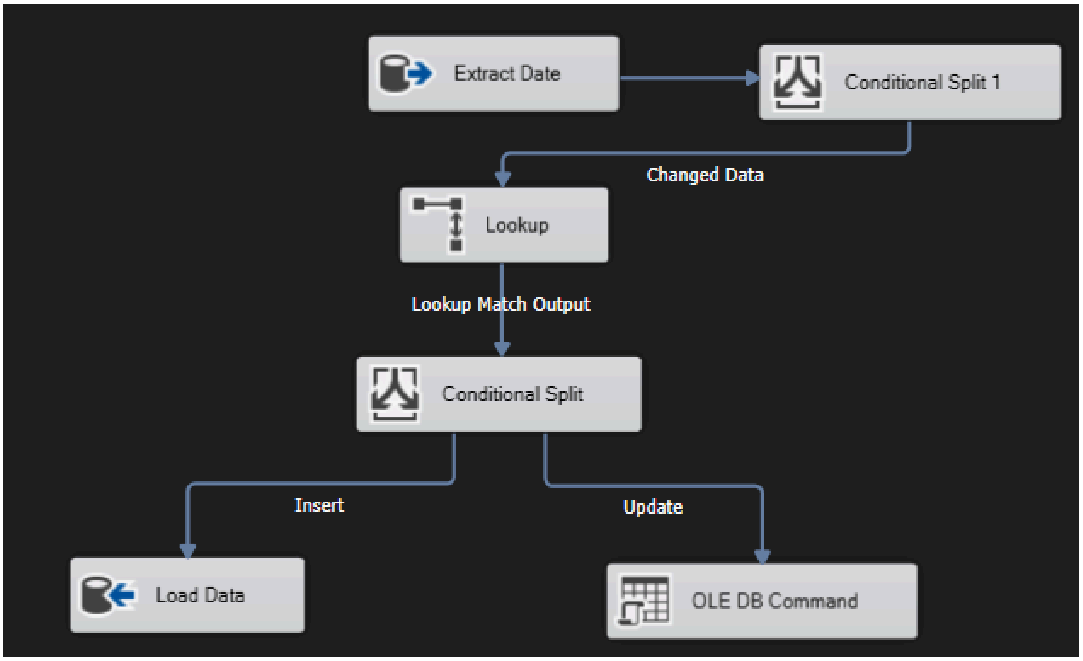
This Dimension will extract the data from the Sales.SalesTerritory table in the database of CompanyX. After transforming the data, it will look up the modified date in the Sales.SalesTerritory table. If the modified date of the changed row is not smaller than the largest modified date in the dimension table, it will be passed through the second conditional split. At this step, it will insert a new row into the table if the changed row does not exist. Otherwise, it will update only the changed row.

The next one is the Special Offer Dimension:



The same operation will be done with this dimension table. After extracting the data from the Sales.SpecialOffer, it will check the modified date in each row. If the modified date of each row in the Sales.SpecialOffer table is not smaller than the largest modified date in the DimSpecialOffer, it will be passed through to the second conditional split. At this step, it will check whether the row already exists or not to decide the next step, which is insert or update. It will insert a new row into the table if the changed row does not exist. Otherwise, it will update only the changed row.

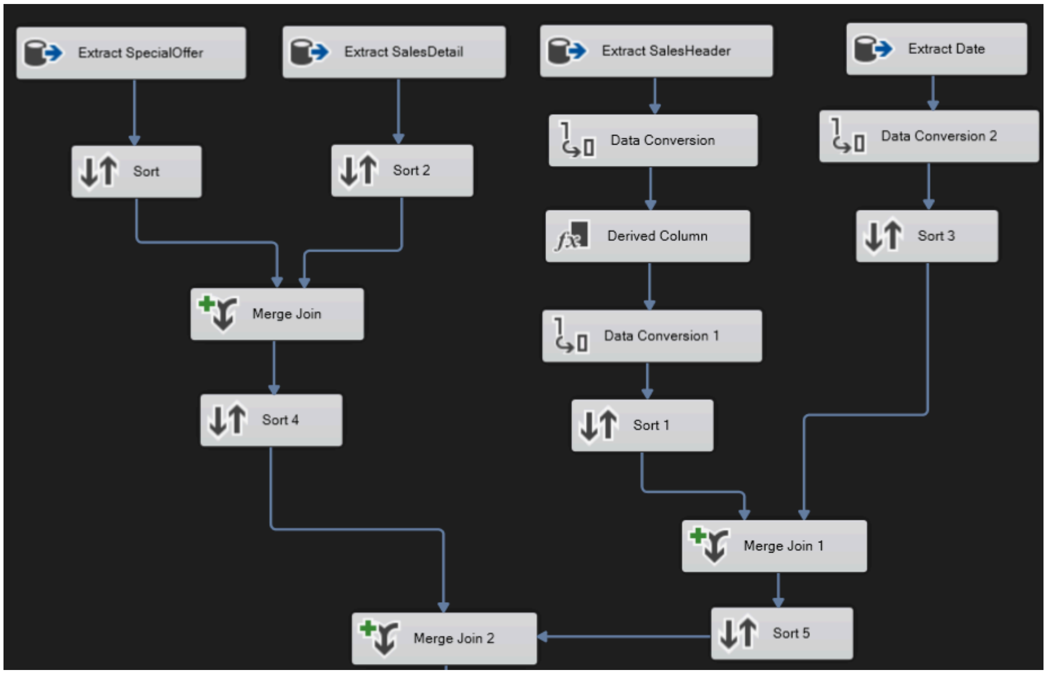
The last dimension of the star schema is Date Dimension.



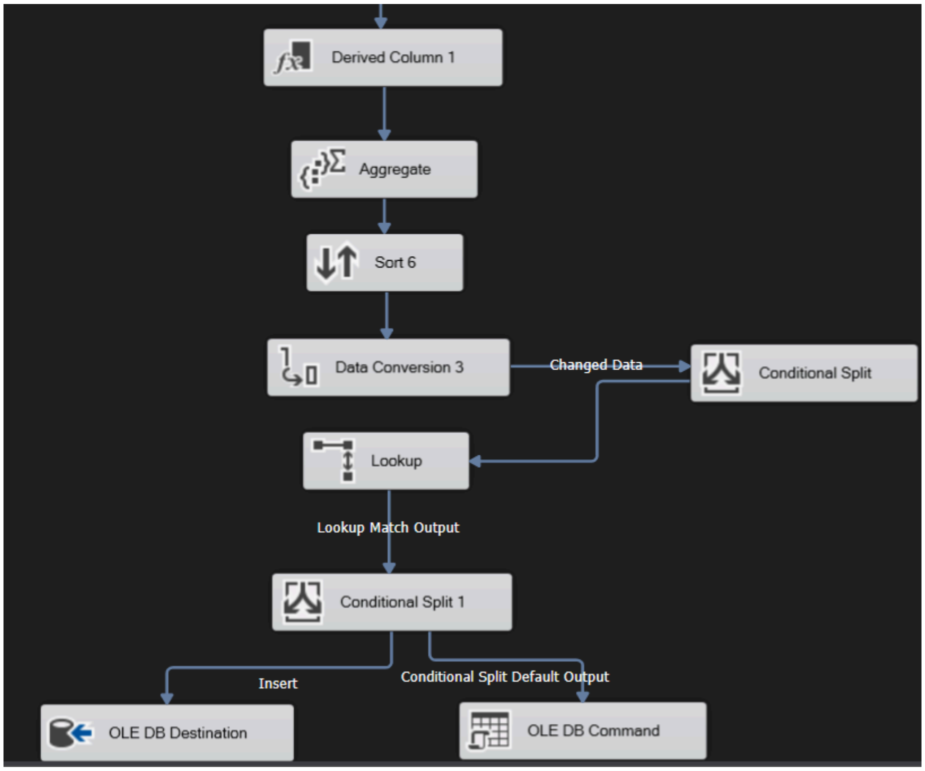
This dimension is especially important for the data warehouse. To create this dimension, it is crucial to create a table by using the command in SQL server and populate the table by hand. This dimension only contains the information from 2011 to the end of 2014. This dimension was created to manage and control all the transactions of the company to check the revenue of the company.

With the same method as other dimensions, it will extract the information from the Date table in the CompanyX and make a comparison between the modified date in the Date table and the Date Dimension table. If the modified date in the dimension table is not larger than the modified date in the Date table. It will come to the second conditional split. At this stage, it will decide to insert if the row does not exist. Otherwise, it will update the existing row in the dimension table.

The final table of the star schema in the data warehouse is the SalesFact table.



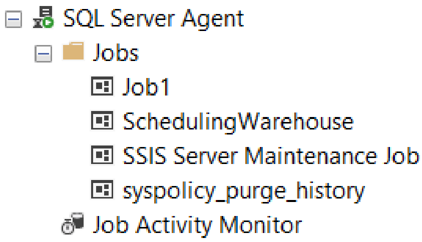
The Fact table consists of the information extracted from four tables in the data warehouse: Sales.SpecialOffer, Sales.SalesOrderDetail, Sales.SalesOrderHeader and dbo.DateData. After the transformation and merge of data from the source tables, only the necessary information for the fact table will be selected. In addition, the primary keys in the four dimensions and other data needed to calculate the metrics will be selected to bring into the fact table. The next step is to build the metric for the SalesFact table.



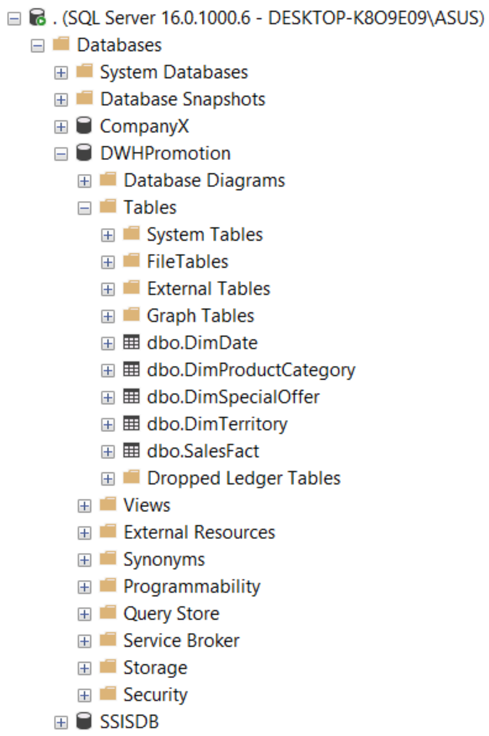
A new column will be created to calculate the money discounted for each row of the merged table. After that, the table will be grouped by five important information: primary keys of four dimensions and ModifiedDate column and the metric of the table including Total Quantity, Total Discount and Total Sales for each row of the SalesFact table. The data conversion is necessary to match the output to the destination table. After these steps, it will make a comparison between the modified date in the Sales.SalesOrderDetail and modified date in the SalesFact table. The reason why the modified date in the SalesOrderDetail table is chosen is it affects the metric in the fact table the most.

If the modified date in the Sales.SalesOrderDetail is not smaller than the largest modified date in the fact table, the output will come to the second conditional split. At this stage, it will decide to insert if the row does not exist. Otherwise, it will update the existing row in the dimension table.

After the successful implementation of ETL pipeline for the data warehouse, it is vital to make the ETL near real-time for the industry. A new job in the SQL Server Agent will be created to guarantee the near real-time feature of the ETL pipeline. The recurrent job named SchedulingWarehouse uses the ETL package.dtsx will be executed daily for every 10 seconds.



A new database named DWHPromotion will be created to act as a data warehouse for CompanyX



```
⊟ . (SQL Server 16.0.1000.6 - DESKTOP-K8O9E09\ASUS)
  ⊟ Databases
    ⊞ System Databases
    ⊞ Database Snapshots
    ⊞ CompanyX
    ⊟ DWHPromotion
      ⊞ Database Diagrams
      ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.DimDate
        ⊞ dbo.DimProductCategory
        ⊞ dbo.DimSpecialOffer
        ⊞ dbo.DimTerritory
        ⊞ dbo.SalesFact
        ⊞ Dropped Ledger Tables
      ⊞ Views
      ⊞ External Resources
      ⊞ Synonyms
      ⊞ Programmability
      ⊞ Query Store
      ⊞ Service Broker
      ⊞ Storage
      ⊞ Security
    ⊞ SSISDB
```

As can be seen from the database, all of the tables in the star schema will be loaded into this data warehouse. Moreover, the information of all tables in the data warehouse will be automatically updated by using the implemented ETL pipeline after 10 seconds every day.

3. Analysis

This analysis outlines the crucial steps taken in the data preparation, mining, and transformation stages. In the pursuit of extracting insights from intricate sales data, this analysis meticulously navigates through three distinct yet interwoven stages.

Beginning with the Data Pre-processing phase, this initial segment is devoted to the methodical preparation of the dataset. Transitioning seamlessly into the second phase, the data mining stage delves into the depths of exploratory analysis. The final phase centers on preparing data for Power BI, where dimension tables are meticulously crafted for seamless integration into the powerful visualization platform.

3.1. Data Pre-processing

a. Library import and File loading

The initial phase of this analysis encompasses the essential setup, commencing with the importation of crucial libraries instrumental in data analysis. Libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn are incorporated to facilitate sophisticated data manipulation, numerical computing, visualization, and clustering techniques. Additionally, the code establishes a connection to the Data Warehouse (DWHPromotion) through pyodbc library.

```
#import libraries for analyzing
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import cluster
import warnings
warnings.filterwarnings("ignore")

#Connect to Data Warehouse
connect = pyodbc.connect(
    Trusted_Connection='Yes',
    Driver='{ODBC Driver 17 for SQL Server}',
    Server='.',
    Database='DWHPromotion'
)
```

Import libraries and connect to data warehouse

Utilizing SQL queries, the code retrieves specific data from different tables within the Data Warehouse. The 'SalesFact' table, 'TerritoryDim', 'DateDim', 'ProductCategoryDim', and 'SpecialOfferDim' tables are individually retrieved using SQL queries 'select * from' corresponding tables in the Data Warehouse, read through the 'pd.read_sql_query' function in Pandas, and stored in respective dataframes. This process lays the foundation for subsequent analysis by acquiring comprehensive datasets encompassing sales records, territorial information, temporal attributes, product categories, and details of special offers, enabling thorough exploration and analysis of the data within the Python environment.

```
#import sales fact table file
SalesFact = pd.read_sql_query('select * from SalesFact', connect)
#import territory dimmension file
TerritoryDim = pd.read_sql_query('select * from SalesFact', connect)
#import date dimmension file
```

```
DateDim = pd.read_sql_query('select * from DimDate', connect)
#import Product Catergory dimmension file
ProductCategoryDim = pd.read_sql_query('select * from DimProductCategory', connect)
#import Special Offer dimmension file
SpecialOfferDim = pd.read_sql_query('select * from DimSpecialOffer', connect)
```

Import data from data warehouse to data frame

b. Initial Data cleaning

In refining the data for analysis, the code systematically removes the 'ModifiedDate' column from each dimension and the sales dataframe, recognizing that this attribute doesn't contribute significantly to the analytical process. The repeated application of 'drop' functions across 'Sales_df', 'Territory_df', 'Date_df', 'Product_df', and 'SpecialOffer_df' ensures a standardized dataset across dimensions, eliminating non-contributory columns to streamline subsequent analytical processes.

```
        Sales_df.drop(['ModifiedDate'], axis=1, inplace=True)
        Sales_df
        Territory_df.drop(['ModifiedDate'], axis=1, inplace=True)
        Territory_df
        Date_df.drop(['ModifiedDate'], axis=1, inplace=True)
        Date_df
        Product_df.drop(['ModifiedDate'], axis=1, inplace=True)
        Product_df
        SpecialOffer_df.drop(['ModifiedDate'], axis=1, inplace=True)
        SpecialOffer_df
```

Drop Modified Date column out of Sales Fact and Territory, Date, Product Dimension

To ensure data integrity and accuracy, null rows within the 'Sales_df' dataframe were dropped. This step, executed through the 'dropna()' function, strategically removes incomplete or missing data points within the sales records.

```
        #drop null row in sales fact data frame
        Sales_df =  Sales_df.dropna()
        Sales_df
```

Drop null rows out of Sales Fact

Furthermore, within the sales dataframe ('Sales_df'), transactions associated with 'SpecialOfferID' equal to 1 were excluded from the analysis. The rationale behind this exclusion lies in the specific identification of 'SpecialOfferID' 1, which signifies transactions without any associated promotion or discount. Thus, the code filters out these entries to concentrate solely on transactions that involve promotional offers, providing a focused analysis of sales data affected by promotional discounts.

```
#drop special offer 1 in Sales fact data frame
Sales_df =  Sales_df[Sales_df['SpecialOfferID'] != 1]
Sales_df
```

Drop rows with SpecialOfferID equivalent to 1 out of Sales Fact

c. Dataframe merging

The sales facts were merged with territory details ('Territory_df'), resulting in the creation of 'Territory_Sales_df'. This amalgamation aimed to enrich the dataset, providing comprehensive insights into sales across diverse territories.

```
# Merge Territory dimmension dataframe with sale facts dataframe
Territory_Sales_df = pd.merge(Sales_df,Territory_df)
Territory_Sales_df
```

Merging Territory dimension with Sale Fact dataframe

The sales data was combined with date-related information from 'Date_df', forming 'Date_Sales_df'. This fusion facilitated a detailed examination of sales trends concerning various temporal aspects.

```
# Merge Date dimmension dataframe with sale facts dataframe
Date_Sales_df = pd.merge(Sales_df,Date_df)
Date_Sales_df
```

Merging Date dimension with Sale Fact dataframe

The sales information was merged with product details ('Product_df'), resulting in 'Product_Sales_df'. This dataset's purpose was to investigate sales patterns associated with different product categories.

```
# Merge Product dimmension dataframe with sale facts dataframe
Product_Sales_df = pd.merge(Sales_df,Product_df)
Product_Sales_df
```

Merging Product dimension with Sale Fact dataframe

d. Attribute Selection for Analysis

Attributes like 'SpecialOfferID', 'TerritoryID', 'Name', 'TotalQty', and 'TotalSales' were extracted from 'Territory_Sales_df'. This extraction allowed a focused exploration of sales patterns within different territories, ensuring data accuracy by removing null values.

```
# data frame for analyzing Territory dim
df1 = pd.DataFrame()

#choose attributes for analyzing in Territory dim
df1['SpecialOfferID'] =  Territory_Sales_df['SpecialOfferID']
df1['TerritoryID'] = Territory_Sales_df['TerritoryID']
df1['Name'] = Territory_Sales_df['Name']
df1['TotalQty'] = Territory_Sales_df['TotalQty']
df1['TotalSales'] = Territory_Sales_df['TotalSales']
df1 =  df1.dropna()
df1
```

Storing attributes for analyzing from Territory Dimension into df1

Similar to the Territory dimension, essential attributes—'SpecialOfferID', 'Year', 'Quarter', 'Month', 'DayOfWeek', 'TotalQty', and 'TotalSales'—were picked from 'Date_Sales_df'. This subset enabled analysis of sales trends over time, also ensuring data completeness by eliminating null entries.

```
# data frame for analyzing Date dim
df2 = pd.DataFrame()

#choose attributes for analyzing in Date dim
df2['SpecialOfferID'] =  Date_Sales_df['SpecialOfferID']
df2['Year'] = Date_Sales_df['Year']
df2['Quarter'] = Date_Sales_df['Quarter']
df2['Month'] = Date_Sales_df['Month']
df2['DayOfWeek'] = Date_Sales_df['DayOfWeek']
df2['TotalQty'] = Date_Sales_df['TotalQty']
df2['TotalSales'] = Date_Sales_df['TotalSales']
df2 =  df2.dropna()
df2
```

Storing attributes for analyzing from Date Dimension into df2

Following the same approach, critical attributes—'SpecialOfferID', 'ProductID', 'SubCategory', 'TotalQty', and 'TotalSales'—were extracted from 'Product_Sales_df'. This extraction aimed to explore sales behavior across diverse product categories, maintaining data accuracy by removing null rows.

```
# data frame for analyzing Product dim
df3 = pd.DataFrame()

#choose attributes for analyzing in Product dim
df3['SpecialOfferID'] =  Product_Sales_df['SpecialOfferID']
df3['ProductID'] = Product_Sales_df['ProductID']
df3['SubCategory'] = Product_Sales_df['SubCategory']
df3['TotalQty'] = Product_Sales_df['TotalQty']
df3['TotalSales'] = Product_Sales_df['TotalSales']
df3 =  df3.dropna()
df3
```

Storing attributes for analyzing from Product Dimension into df3

3.2. Data mining

a. Grouping data

For a comprehensive analysis, the 'TotalQty' and 'TotalSales' attributes within the Territory dimension were aggregated. This resulted in the creation of a new dataframe, 'TerritoryCluster', formed by grouping attributes like 'SpecialOfferID', 'TerritoryID', and 'Name' from 'df1'. The sums of 'TotalQty' and 'TotalSales' were calculated for each distinct attribute combination.

```
#Create dataframe contain the sum of TotalQty and TotalSales group by attributes of Territory Dimmension
TerritoryCluster = df1.groupby(['SpecialOfferID','TerritoryID','Name'], as_index=False).agg({
   'TotalQty': 'sum',
   'TotalSales': 'sum',
})
TerritoryCluster=TerritoryCluster.sort_values(['SpecialOfferID',
'TerritoryID','Name']  , ascending=True)
TerritoryCluster.reset_index(drop=True, inplace=True)
TerritoryCluster
```

Group attributes for clustering Territory Dimension

Following a similar approach, the Date dimension underwent data summarization. 'DateCluster' was created by aggregating 'TotalQty' and 'TotalSales' based on attributes like 'SpecialOfferID', 'DayOfWeek', 'Month', 'Quarter', and 'Year' from 'df2'.

```
#Create dataframe contain the sum of TotalQty and TotalSales group by attributes of Date Dimmension
DateCluster=df2.groupby(['SpecialOfferID','DayOfWeek','Month','Quarter','Year'],
as_index=False).agg({
   'TotalQty': 'sum',
   'TotalSales': 'sum',
})
DateCluster=DateCluster.sort_values(['SpecialOfferID','Year','Quarter',
'Month','DayOfWeek',]  , ascending=True)
DateCluster.reset_index(drop=True, inplace=True)
DateCluster
```

Group attributes for clustering Date Dimension

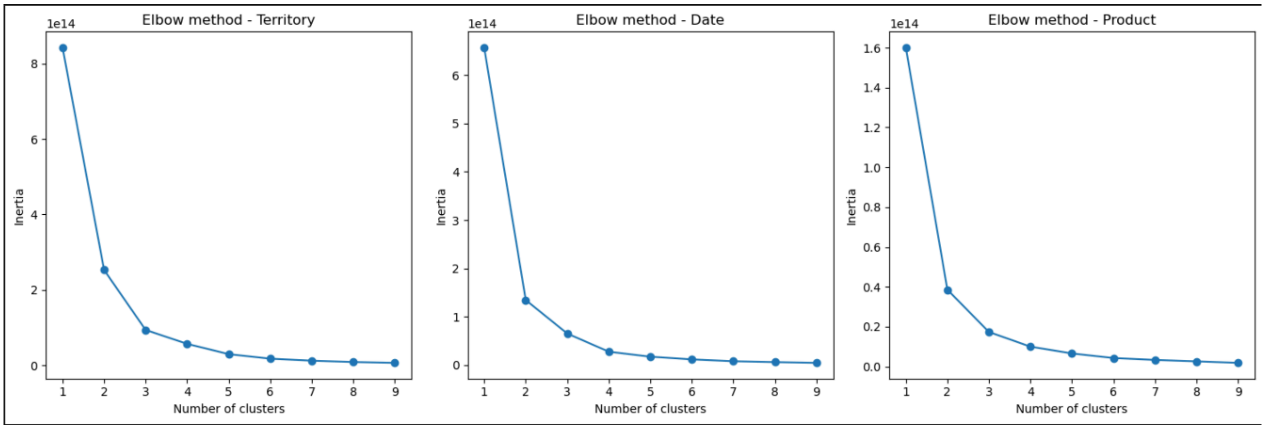
In line with the previous dimensions, the Product dimension's 'TotalQty' and 'TotalSales' were summarized. The 'ProductCluster' dataframe was constructed by aggregating these attributes, grouped by 'SpecialOfferID', 'ProductID', and 'SubCategory' from 'df3'. This aggregation facilitated an exploration of sales patterns related to various product categories.

```
#Create dataframe contain the sum of TotalQty and TotalSales group by attributes of Product Dimmension
ProductCluster = df3.groupby(['SpecialOfferID','ProductID','SubCategory'], as_index=False).agg({
    'TotalQty': 'sum',
    'TotalSales': 'sum',
})
ProductCluster=ProductCluster.sort_values(['SpecialOfferID','ProductID',
'SubCategory'] , ascending=True)
ProductCluster.reset_index(drop=True, inplace=True)
ProductCluster
```

Group attributes for clustering Product Dimension

b. Elbow method

The Elbow Method visualization was generated using Matplotlib's subplots, plotting the inertia values against varying numbers of clusters (ranging from 1 to 9). This facilitated the identification of the ideal cluster count by observing the point where the inertia starts to decrease at a slower rate, hence resembling an 'elbow' in the plot. Each subplot was labeled to denote its respective dimension, aiding in easy identification and interpretation of the optimal cluster count.



Elbow Method visualization in Python

c. KMeans clustering

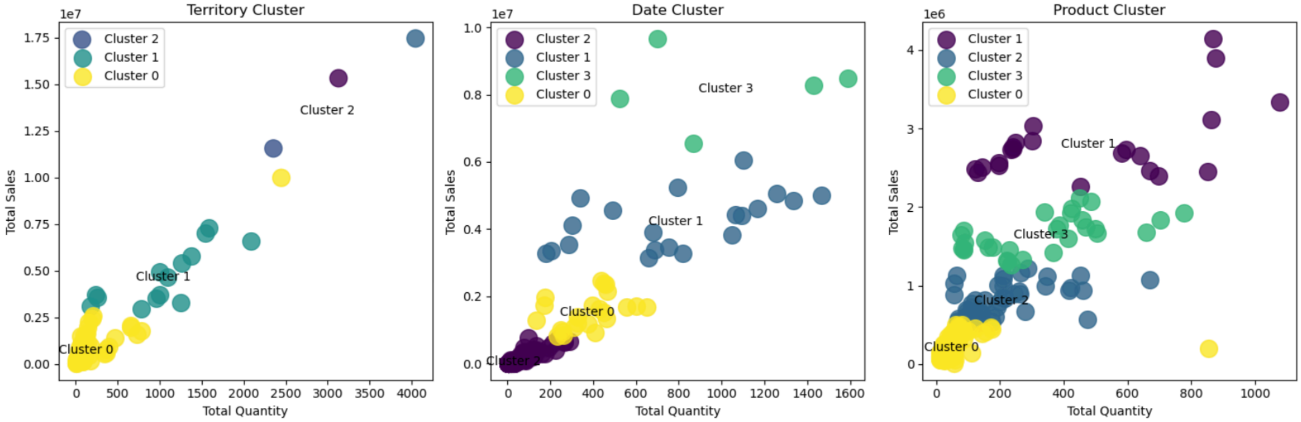
During the clustering phase, 'X1', 'X2', and 'X3' represent aggregated datasets derived from 'TerritoryCluster', 'DateCluster', and 'ProductCluster', combining 'TotalQty' and 'TotalSales' attributes for comprehensive sales analysis. The choice of 'k_means_3' and 'k_means_4' corresponds to the optimal number of clusters determined via the Elbow Method. 'k_means_3' signifies three clusters identified for Territory analysis, while 'k_means_4' denotes models configured with four clusters for Date and Product analysis. This approach enabled the identification of specific groupings within each dimension, facilitating deeper analytical insights into sales patterns across diverse dimensions.

```
#Select the TotalQty and TotalSales column as a criteria for clustering
X1= TerritoryCluster.iloc[:, -2:].values
X2= DateCluster.iloc[:, -2:].values
X3= ProductCluster.iloc[:, -2:].values

#Define number of cluster and put criteria data for clustering using Kmean model
k_means_3= KMeans(n_clusters = 3, init = 'k-means++',  random_state=42) # model use for
Territory
y1 = k_means_3.fit_predict(X1)

k_means_4= KMeans(n_clusters = 4, init = 'k-means++',  random_state=42) # model use for
Date and Product
y2 = k_means_4.fit_predict(X2)
y3 = k_means_4.fit_predict(X3)
```

K-mean clustering application for three dimensions

d. Python visualization

The clustering results were visually represented using scatter plots, where each dimension's clusters ('Territory Cluster', 'Date Cluster', 'Product Cluster') were visualized based on 'TotalQty' against 'TotalSales'. Each cluster was distinguished by unique colors, with centroids marked and labeled accordingly, providing a clear visual delineation of distinct clusters within each dimension.



Scatter plots clustering representation for each dimension

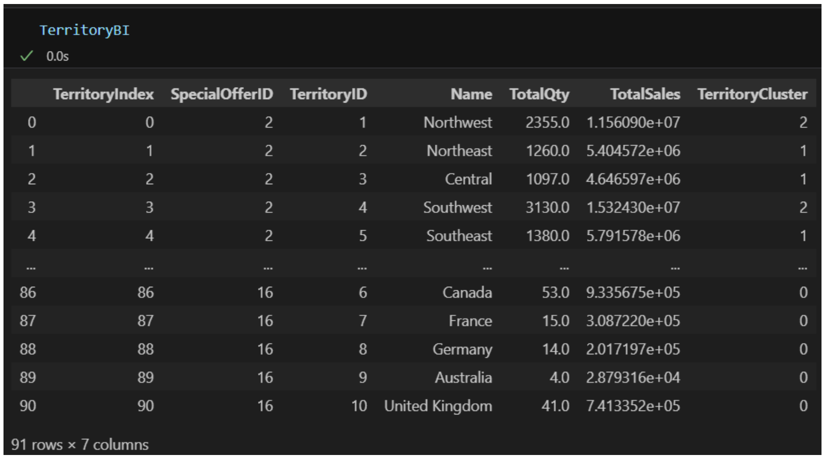
3.3. Preparing data for PowerBI

In preparation for Power BI integration, the data underwent refinement and restructuring across various stages. The creation of the PromotionSale dataframe involved merging 'Sales_df' with 'Date_df', removing specific columns ('DateKey', 'TotalDiscount', 'DateID', 'Day', 'Week', 'HolidayName'), and reordering columns to extract relevant attributes ('SpecialOfferID', 'DayOfWeek', 'Month', 'Quarter', 'Year', 'ProductID', 'TerritoryID', 'TotalQty', 'TotalSales'). Excluding rows with 'SpecialOfferID' equal to 1 ensured analysis focused solely on transactions involving promotions.

```
# Perpare PromotionSale for fact table in Power BI
PromotionSale = pd.merge(Sales_df,Date_df)
columns_to_drop = ['DateKey', 'TotalDiscount','DateID','Day','Week','HolidayName']  # List of column names to drop
PromotionSale = PromotionSale.drop(columns=columns_to_drop)
new_column_order= ['SpecialOfferID','DayOfWeek','Month','Quarter','Year','ProductID', 'TerritoryID','TotalQty','TotalSales']
PromotionSale = PromotionSale[new_column_order]
PromotionSale =  PromotionSale[PromotionSale['SpecialOfferID'] != 1]
PromotionSale.dropna
PromotionSale
```

Prepare necessary data for Power BI

Following this, dimension tables ('TerritoryBI', 'DateBI', 'ProductBI') were derived from respective clustered datasets ('TerritoryCluster', 'DateCluster', 'ProductCluster'). These tables were augmented with index columns ('TerritoryIndex', 'DateIndex', 'ProductIndex') and renamed ('TerritoryCluster', 'DateCluster', 'ProductCluster') to facilitate their integration into Power BI.

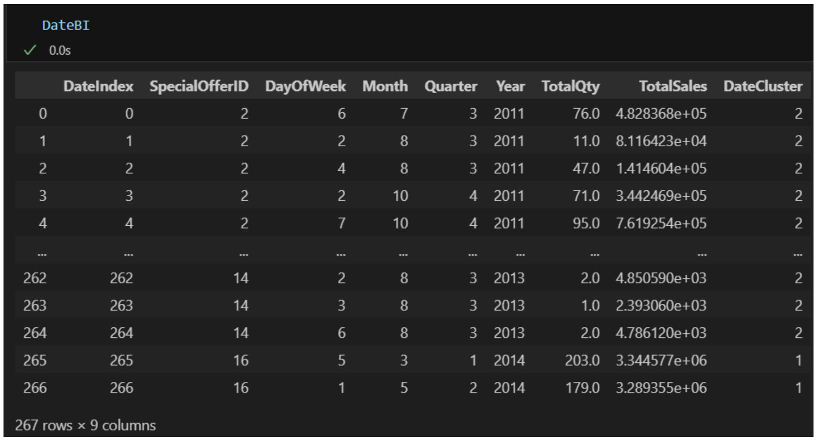


```
TerritoryBI
✓ 0.0s
```

	TerritoryIndex	SpecialOfferID	TerritoryID	Name	TotalQty	TotalSales	TerritoryCluster
0	0	2	1	Northwest	2355.0	1.156090e+07	2
1	1	2	2	Northeast	1260.0	5.404572e+06	1
2	2	2	3	Central	1097.0	4.646597e+06	1
3	3	2	4	Southwest	3130.0	1.532430e+07	2
4	4	2	5	Southeast	1380.0	5.791578e+06	1
...	...	...	...	...	...	...	...
86	86	16	6	Canada	53.0	9.335675e+05	0
87	87	16	7	France	15.0	3.087220e+05	0
88	88	16	8	Germany	14.0	2.017197e+05	0
89	89	16	9	Australia	4.0	2.879316e+04	0
90	90	16	10	United Kingdom	41.0	7.413352e+05	0

91 rows × 7 columns

Complete Territory Dimension clustering dataframe for Power BI

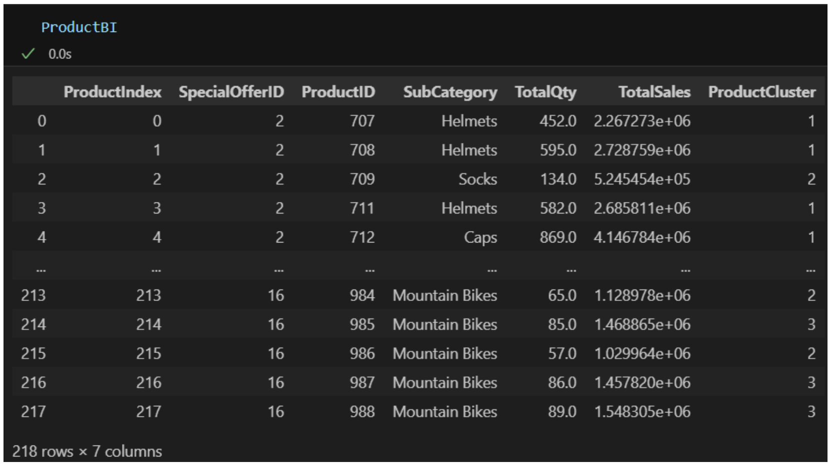


```
DateBI
✓ 0.0s
```

	DateIndex	SpecialOfferID	DayOfWeek	Month	Quarter	Year	TotalQty	TotalSales	DateCluster
0	0	2	6	7	3	2011	76.0	4.828368e+05	2
1	1	2	2	8	3	2011	11.0	8.116423e+04	2
2	2	2	4	8	3	2011	47.0	1.414604e+05	2
3	3	2	2	10	4	2011	71.0	3.442469e+05	2
4	4	2	7	10	4	2011	95.0	7.619254e+05	2
...	...	...	...	...	...	...	...	...	...
262	262	14	2	8	3	2013	2.0	4.850590e+03	2
263	263	14	3	8	3	2013	1.0	2.393060e+03	2
264	264	14	6	8	3	2013	2.0	4.786120e+03	2
265	265	16	5	3	1	2014	203.0	3.344577e+06	1
266	266	16	1	5	2	2014	179.0	3.289355e+06	1

267 rows × 9 columns

Complete Date Dimension clustering dataframe for Power BI



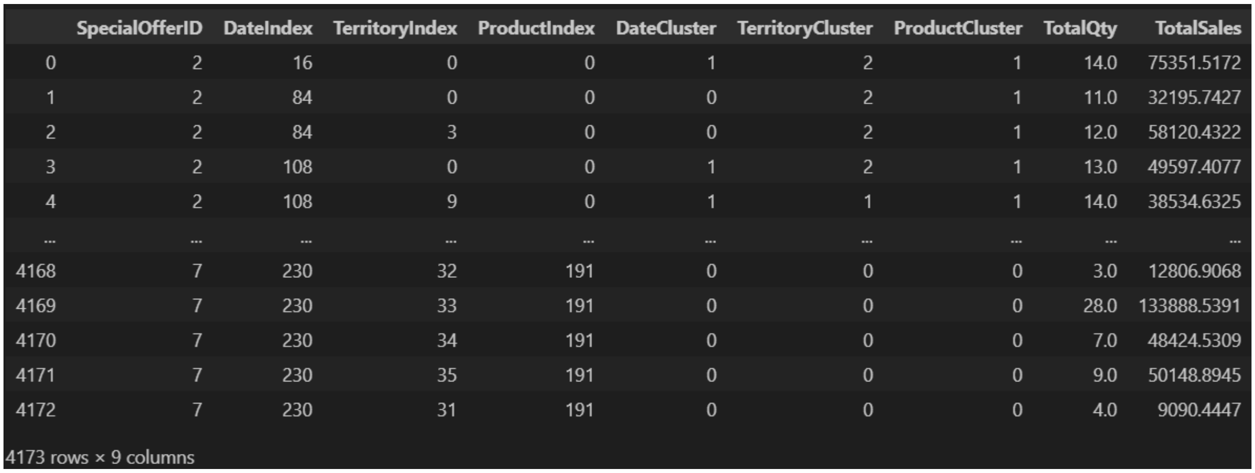
ProductBI
✓ 0.0s

	ProductIndex	SpecialOfferID	ProductID	SubCategory	TotalQty	TotalSales	ProductCluster
0	0	2	707	Helmets	452.0	2.267273e+06	1
1	1	2	708	Helmets	595.0	2.728759e+06	1
2	2	2	709	Socks	134.0	5.245454e+05	2
3	3	2	711	Helmets	582.0	2.685811e+06	1
4	4	2	712	Caps	869.0	4.146784e+06	1
...	...	...	...	...	...	...	...
213	213	16	984	Mountain Bikes	65.0	1.128978e+06	2
214	214	16	985	Mountain Bikes	85.0	1.468865e+06	3
215	215	16	986	Mountain Bikes	57.0	1.029964e+06	2
216	216	16	987	Mountain Bikes	86.0	1.457820e+06	3
217	217	16	988	Mountain Bikes	89.0	1.548305e+06	3

218 rows × 7 columns

Complete Product Dimension clustering dataframe for Power BI

Furthermore, the 'FactBI' table was created by merging 'PromotionSale' with the dimension tables, dropping irrelevant columns ('DayOfWeek', 'Month', 'Quarter', 'Year', 'ProductID', 'TerritoryID', 'Name', 'SubCategory'). This optimized 'FactBI' contained essential attributes ('SpecialOfferID', 'DateIndex', 'TerritoryIndex', 'ProductIndex', 'DateCluster', 'TerritoryCluster', 'ProductCluster', 'TotalQty', 'TotalSales') crucial for meaningful visualizations and deeper sales data analysis within Power BI.



	SpecialOfferID	DateIndex	TerritoryIndex	ProductIndex	DateCluster	TerritoryCluster	ProductCluster	TotalQty	TotalSales
0	2	16	0	0	1	2	1	14.0	75351.5172
1	2	84	0	0	0	2	1	11.0	32195.7427
2	2	84	3	0	0	2	1	12.0	58120.4322
3	2	108	0	0	1	2	1	13.0	49597.4077
4	2	108	9	0	1	1	1	14.0	38534.6325
...	...	...	...	...	...	...	...	...	...
4168	7	230	32	191	0	0	0	3.0	12806.9068
4169	7	230	33	191	0	0	0	28.0	133888.5391
4170	7	230	34	191	0	0	0	7.0	48424.5309
4171	7	230	35	191	0	0	0	9.0	50148.8945
4172	7	230	31	191	0	0	0	4.0	9090.4447

4173 rows × 9 columns

Complete Sale Fact dataframe for Power BI

Subsequently, these tables were saved into the data warehouse with the table named TerritoryBI, DateBI, ProductBI for seamless use within Power BI. Finally, the 'FactBI' table also was saved into the data warehouse with the table named FactBI for Power BI utilization.

```
import sqlalchemy as sa
```

```
    connection_string = (
        'Driver=ODBC Driver 17 for SQL Server;'
        'Server=.;'
        'Database=DWHPromotion;'
        'Trusted_Connection=yes;'
    )

    connection_url = sa.engine.URL.create(
        "mssql+pyodbc",
        query=dict(odbc_connect=connection_string)
    )
    engine = sa.create_engine(connection_url, fast_executemany=True)

    # Deleting existing data in SQL Table:-
    with engine.begin() as conn:
        conn.exec_driver_sql("DELETE FROM ProductBI")

    # upload the DataFrame
    ProductBI.to_sql("ProductBI", engine, if_exists="append", index=False)
```

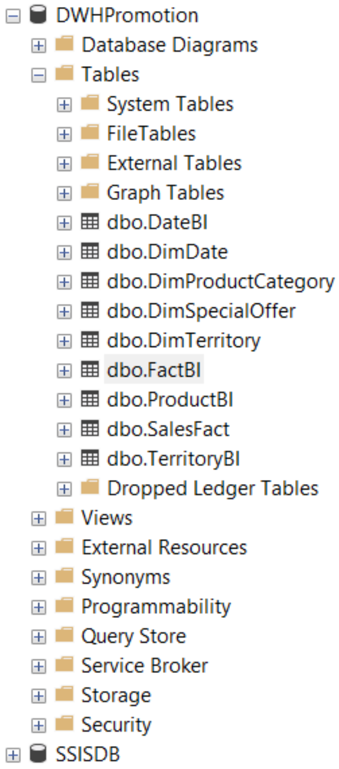
```
# Deleting existing data in SQL Table:-
with engine.begin() as conn:
    conn.exec_driver_sql("DELETE FROM TerritoryBI")

# upload the DataFrame
TerritoryBI.to_sql("TerritoryBI", engine, if_exists="append", index=False)
```

```
# Deleting existing data in SQL Table:-
with engine.begin() as conn:
    conn.exec_driver_sql("DELETE FROM DateBI")
```

```
# upload the DataFrame
DateBI.to_sql("DateBI", engine, if_exists="append", index=False)
```

```
# Deleting existing data in SQL Table:-
with engine.begin() as conn:
    conn.exec_driver_sql("DELETE FROM FactBI")

# upload the DataFrame
FactBI.to_sql("FactBI", engine, if_exists="append", index=False)
```

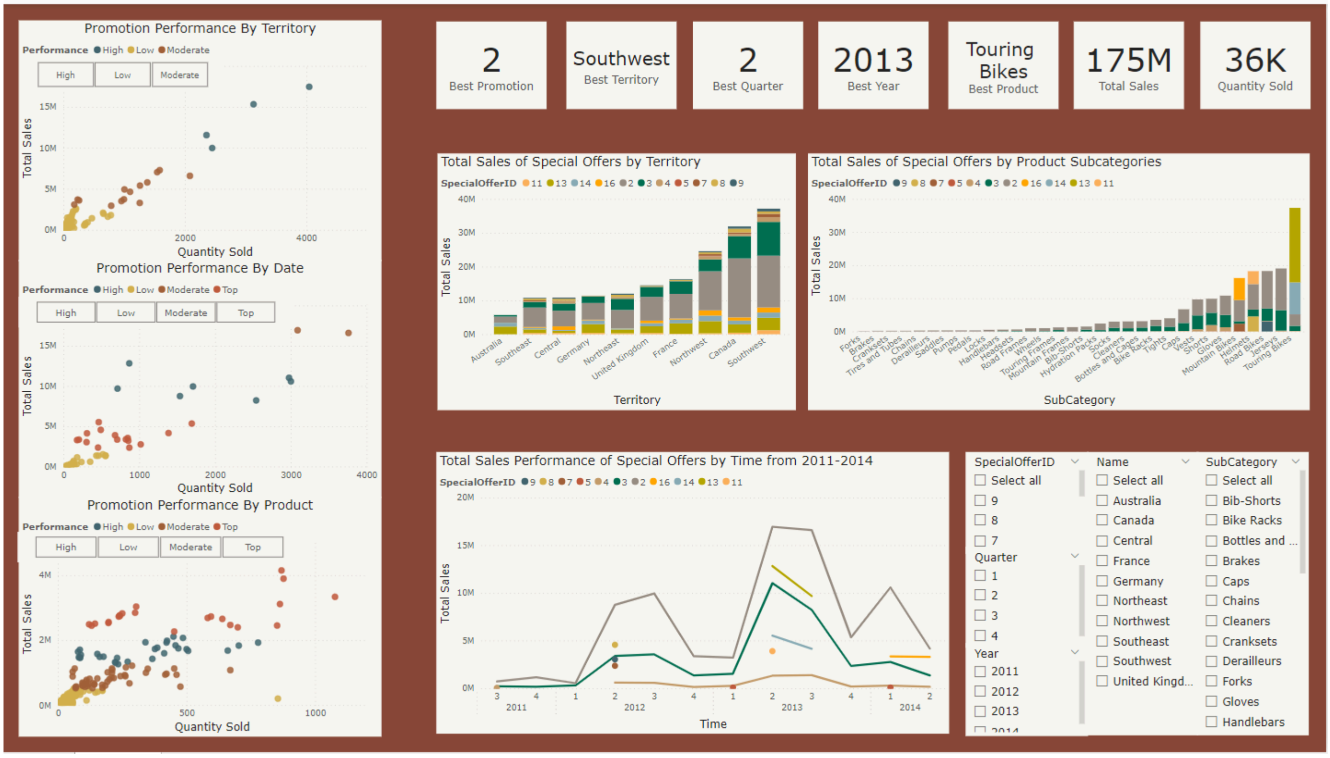
VIII. BI System and Visualization



```
⊟ DWHPromotion
  ⊞ Database Diagrams
  ⊟ Tables
    ⊞ System Tables
    ⊞ FileTables
    ⊞ External Tables
    ⊞ Graph Tables
    ⊞ dbo.DateBI
    ⊞ dbo.DimDate
    ⊞ dbo.DimProductCategory
    ⊞ dbo.DimSpecialOffer
    ⊞ dbo.DimTerritory
    ⊞ dbo.FactBI
    ⊞ dbo.ProductBI
    ⊞ dbo.SalesFact
    ⊞ dbo.TerritoryBI
    ⊞ Dropped Ledger Tables
  ⊞ Views
  ⊞ External Resources
  ⊞ Synonyms
  ⊞ Programmability
  ⊞ Query Store
  ⊞ Service Broker
  ⊞ Storage
  ⊞ Security
⊞ SSISDB
```

1. BI system

The Power BI dashboard presented here serves as a robust tool enabling users to comprehend intricate information effortlessly. By navigating this dashboard, end users can pinpoint the most effective promotional combinations across territories, dates, and products, optimizing sales and quantities sold. Through its intuitive visualizations, this dashboard encapsulates pivotal insights, facilitating informed decision-making processes. It offers a comprehensive view of current sales trends across various dimensions, empowering users to gauge market behaviors and make strategic decisions. With multiple filters and detailed insights, this visualization aids users in determining optimal promotion durations, timing, and product-specific strategies, empowering precise decision-making for promotions across different territories and times of the year.


2 Best Promotion
Southwest Best Territory
2 Best Quarter
2013 Best Year
Touring Bikes Best Product
175M Total Sales
36K Quantity Sold

Promotion Performance By Territory
Performance ● High ● Low ● Moderate
High
Low
Moderate
Total Sales
Quantity Sold
Promotion Performance By Date
Performance ● High ● Low ● Moderate ● Top
Top
Promotion Performance By Product
Total Sales of Special Offers by Territory
SpecialOfferID ● 11 ● 13 ● 14 ● 16 ● 2 ● 3 ● 4 ● 5 ● 7 ● 8 ● 9
Territory
Australia
Southeast
Central
Germany
Northeast
United Kingdom
France
Northwest
Canada
Southwest
Total Sales of Special Offers by Product Subcategories
SubCategory
Total Sales Performance of Special Offers by Time from 2011-2014
Time
2011
2012
2013
2014
SpecialOfferID
Quarter
Year
Name
SubCategory
Select all

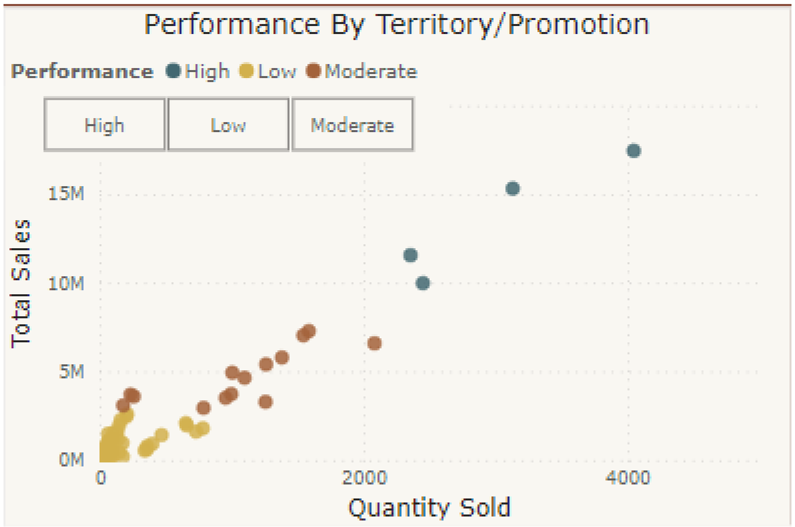
Power BI dashboard visualization

2. Visualization

a. Scatter plots

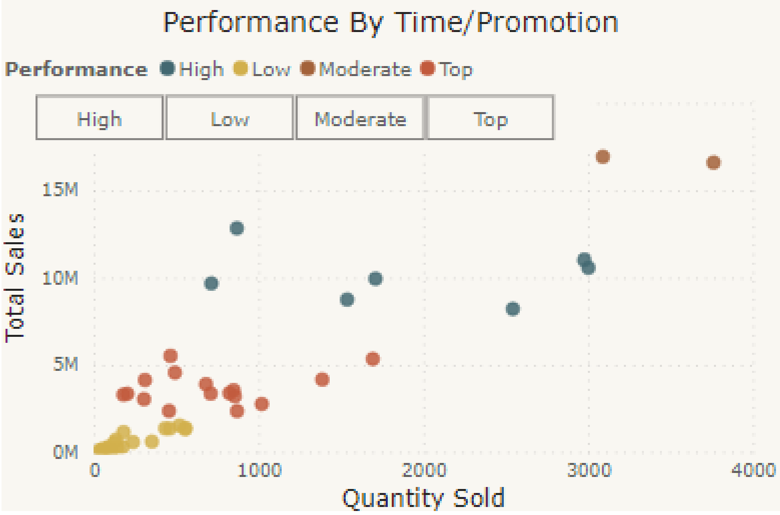
Scatter plots were chosen as they present a clear and intuitive display of multivariate data. With 'TotalQty' and 'TotalSales' plotted on X and Y axes, respectively, these plots showcase clusters, allowing for easy identification of patterns and relationships. Their simplicity enables quick recognition of trends, outliers, and cluster distributions, making them an ideal choice for interpreting clustering results efficiently.

The "Performance by Territory/Prom" scatter plot illustrates the relationship between sales quantity ('TotalQty') and total sales value ('TotalSales') across different territories, segmented by performance levels—Low, Moderate, or High—based on the 'TerritoryCluster' legend. This visualization facilitates the comparison of territories and their respective sales performance, considering the impact of promotions delimited by 'SpecialOfferID.' Users can discern territories exhibiting varying sales quantities and total sales values, helping identify successful promotion strategies deployed in different regions.



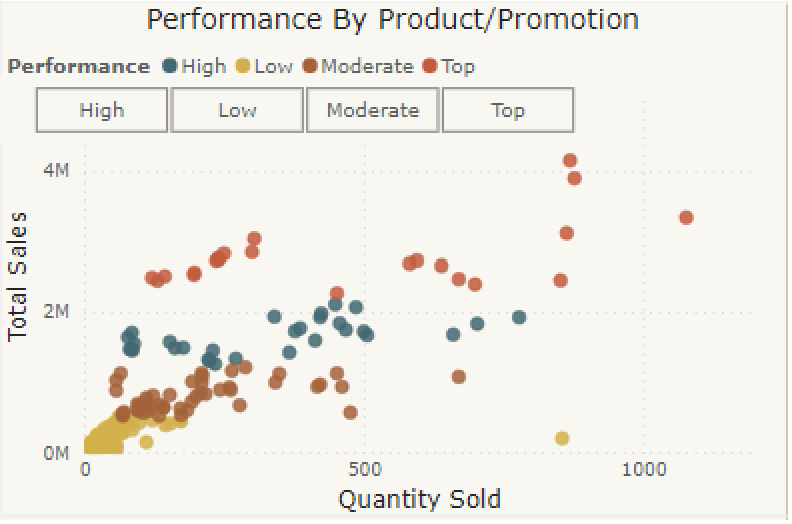
Scatter plot clustering based on Territory and Promotion

Similarly to the previous scatter plot, the "Performance by Time/Promotion" scatter plot demonstrates the relationship between sales quantity ('TotalQty') and total sales value ('TotalSales') across different periods—divided by quarters or years. It incorporates performance segments—Low, Moderate, High, or Top—highlighted by the 'TimeCluster' legend. This visualization enables users to assess the sales trends and performance across various time segments, considering the impact of different 'SpecialOfferID' promotions. Users can identify time periods with varying sales quantities and total sales values, aiding in recognizing successful promotional strategies deployed during specific quarters or years.



Scatter plot clustering based on Time and Promotion

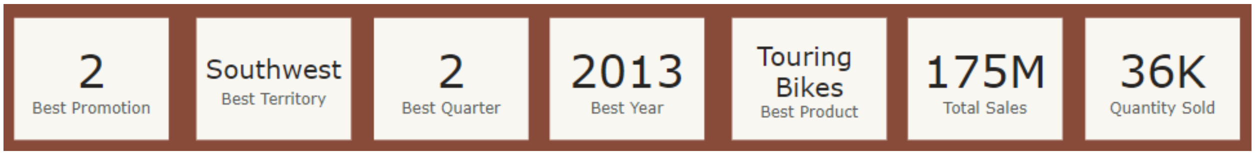
The scatter plot depicting "Performance by Product/Promotion" showcases the correlation between sales quantity ('TotalQty') and total sales value ('TotalSales') across distinct product categories, identified by 'ProductID' and 'SubCategory'. The plot segments products into performance categories—Low, Moderate, High, or Top—based on the 'ProductCluster' legend. Similar to the previous visualizations, this graphic allows users to discern sales patterns and performance within various product categories, considering different 'SpecialOfferID' promotions.



Scatter plot clustering based on Product and Promotion

b. Cards

Cards in Power BI serve as concise, information-rich elements that quickly highlight essential metrics and insights. These cards offer an at-a-glance view of crucial metrics, streamlining decision-making processes.



2
Best Promotion

Southwest
Best Territory

2
Best Quarter

2013
Best Year

Touring Bikes
Best Product

175M
Total Sales

36K
Quantity Sold

Cards highlighting key information

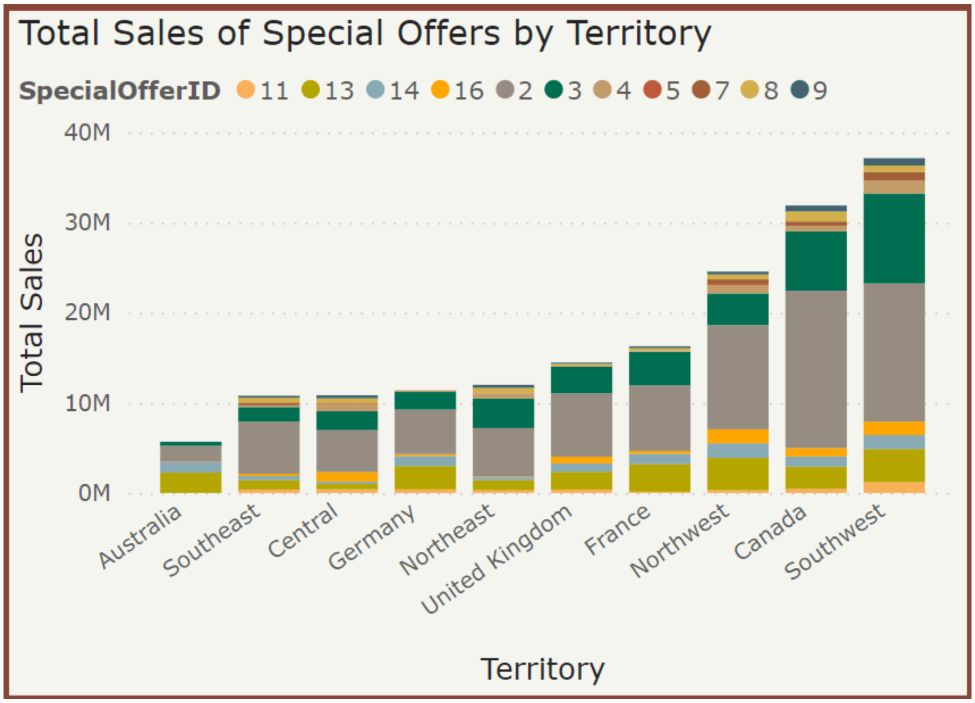
In the presented dashboard, these cards encapsulate critical information aiding end users in making swift, informed decisions:

- **Best Promotion:** A card showcasing the promotion that generates the highest profit, aggregating 'TotalSales' data across different promotions. This card helps identify the most lucrative promotional strategy.
- **Best Territory:** Highlighting the territory with the highest sales success leveraging promotional strategies. It encapsulates 'TotalSales' data attributed to distinct territories, emphasizing the most successful regions.
- **Best Quarter:** A card displaying the quarter of the year with the highest total sales attributed to promotions. This offers insight into seasonal trends or successful quarters for promotional campaigns.
- **Best Year:** Showcasing the year with the highest aggregated sales due to promotions. This highlights the overall success of promotions across the year.
- **Best Product:** Highlighting product category with the highest promotional sales, reflecting on the most successful product promotions.
- **Total Sales:** Displaying the cumulative sales generated through various promotional efforts, offering a comprehensive view of the overall promotional campaign effectiveness in driving revenue.
- **Quantity Sold:** A card summarizing the overall quantity of products sold under promotional strategies, providing an understanding of the collective promotional impact on sales volume.

c. Stack column charts

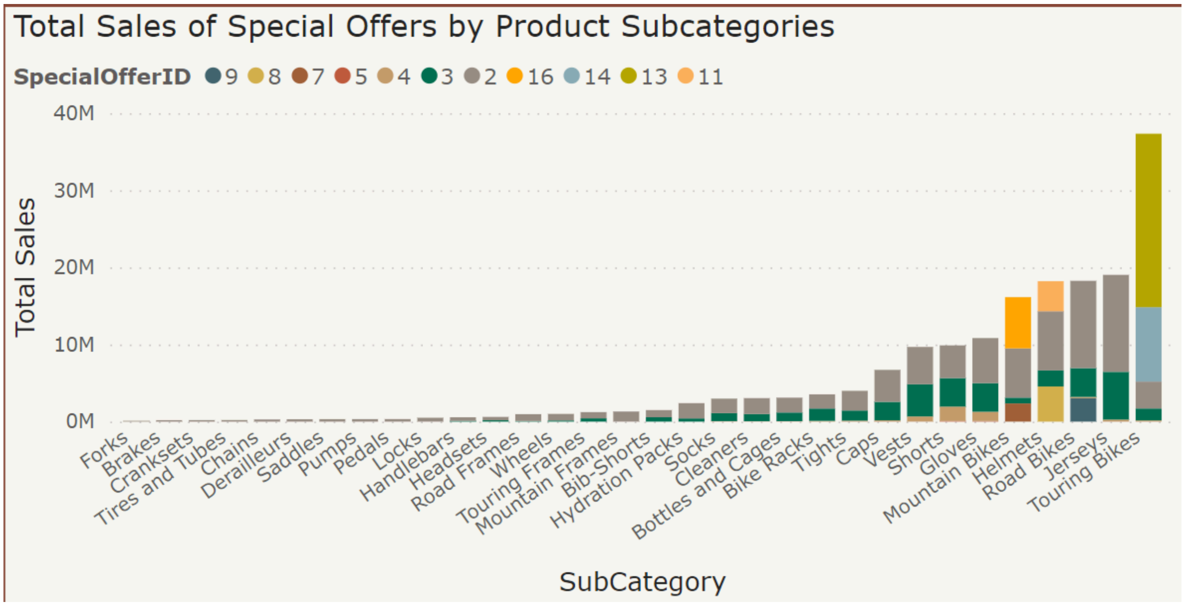
Stacked column charts in Power BI help users by showing multiple categories or data segments stacked within columns. They make it easy to compare different categories at once, see the contribution of each part to the whole, and track trends over time. This visual representation highlights proportions and helps users quickly grasp the distribution and changes in their data.

The "Total Sales of Special Offers by Territory" stacked column chart provides users with a comprehensive view of the sales performance across different territories concerning various special offers. It enables users to discern how each territory contributes to the total sales attributed to different promotional strategies. By visualizing the stacked columns representing individual special offers within each territory, users can identify which territories exhibit higher sales figures and the specific promotions that contribute significantly to those sales



Stack column charts of total sales on territories and promotions

On the other hand, the "Total Sales of Special Offers by Product Subcategories" stacked column chart presents a breakdown of sales performance across product subcategories concerning different special offers. This visualization allows users to discern the distribution of sales among various product categories under different promotional strategies. By showcasing stacked columns representing total sales for each product subcategory within different special offers, users gain insights into which product segments generate the most sales under specific promotions.

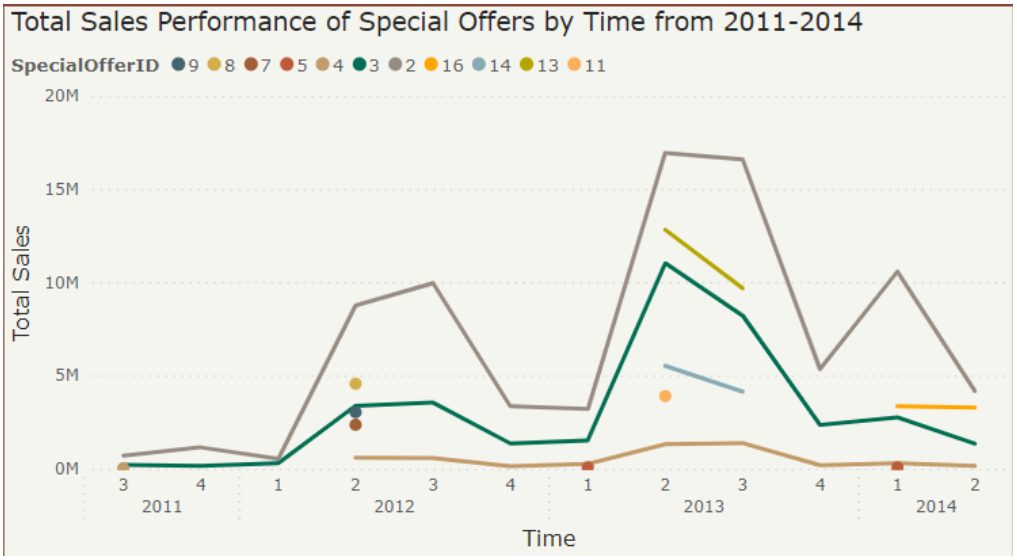


Stack column charts of total sales on territories and promotions

d. Line charts

Line charts serve as valuable tools for visualizing trends, patterns, and changes in data over a continuous period. They are particularly effective in showcasing variations or trends in numerical data across different intervals, making them a preferred choice for depicting time-series data.

Line chart is instrumental in showcasing the trends and fluctuations in sales volumes attributed to different special offers over the years 2011 to 2014. Users can observe how each special offer performs in terms of sales over specific quarters and years, facilitating the identification of seasonal trends or variations in sales patterns. Additionally, it enables users to compare the performance of various special offers against each other over different time intervals, aiding in understanding the effectiveness and consistency of each promotion strategy throughout the specified time frame.

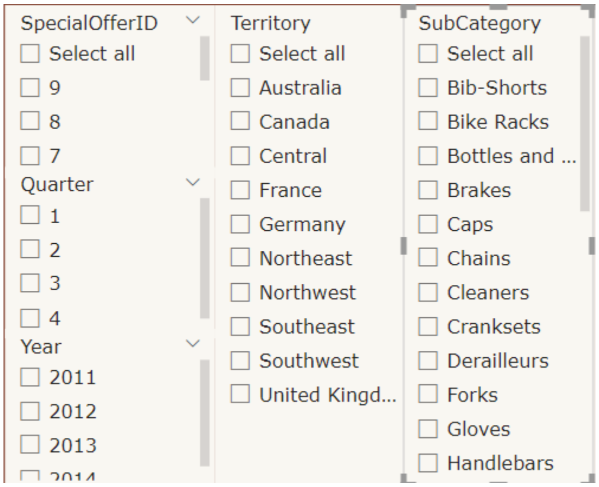


Line charts illustration for promotions sales performance over the years

e. Slicers

Slicers serve as interactive filters in data visualization tools, allowing users to dynamically control and manipulate displayed data. These tools provide a user-friendly interface with options to select and filter data based on specific criteria or categories. By interacting with slicers, users can swiftly isolate subsets of data, focus on particular segments, or refine visualizations based on their preferences.

Through the diverse range of slicers provided, end users can actively participate in strategic decision-making concerning promotions. These interactive filters empower users to discern and fine-tune essential aspects of promotional campaigns.



Slicers for filtering

By selecting specific 'SpecialOfferID' slicer options, users can delineate and evaluate the performance of individual promotions, identifying which ones to initiate, extend, or conclude.

Moreover, the 'Quarter' and 'Years' slicers facilitate precise temporal control, allowing users to pinpoint the optimal timing for promotions. Users can strategically schedule promotions during specific quarters or years, leveraging historical data trends and performance indicators for informed decision-making.

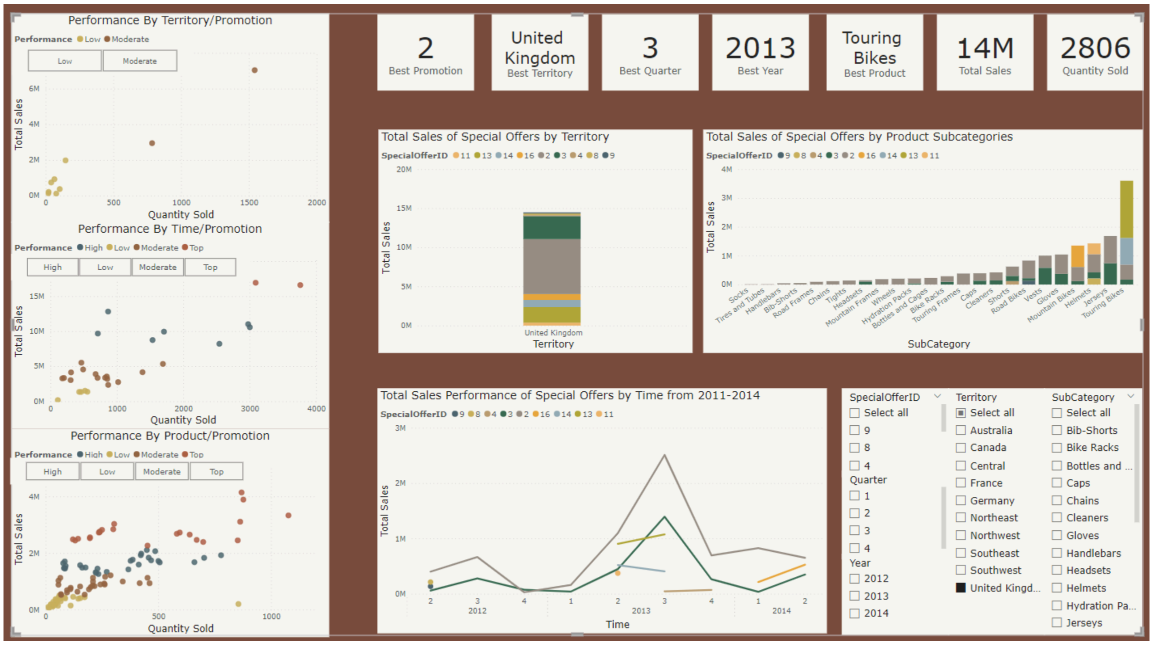
The 'Territory' slicer serves as a geographic filter, enabling users to focus on particular regions or territories, aiding in identifying high-performing areas or targeting regions for improved promotional strategies.

Additionally, the 'Product categories' slicer allows users to refine their analysis by product category, discerning which product segments exhibit potential for increased sales through tailored promotional efforts.

Overall, these slicers collectively empower end users to customize and optimize promotional strategies based on granular insights derived from filtered data subsets.

IX. Testcase

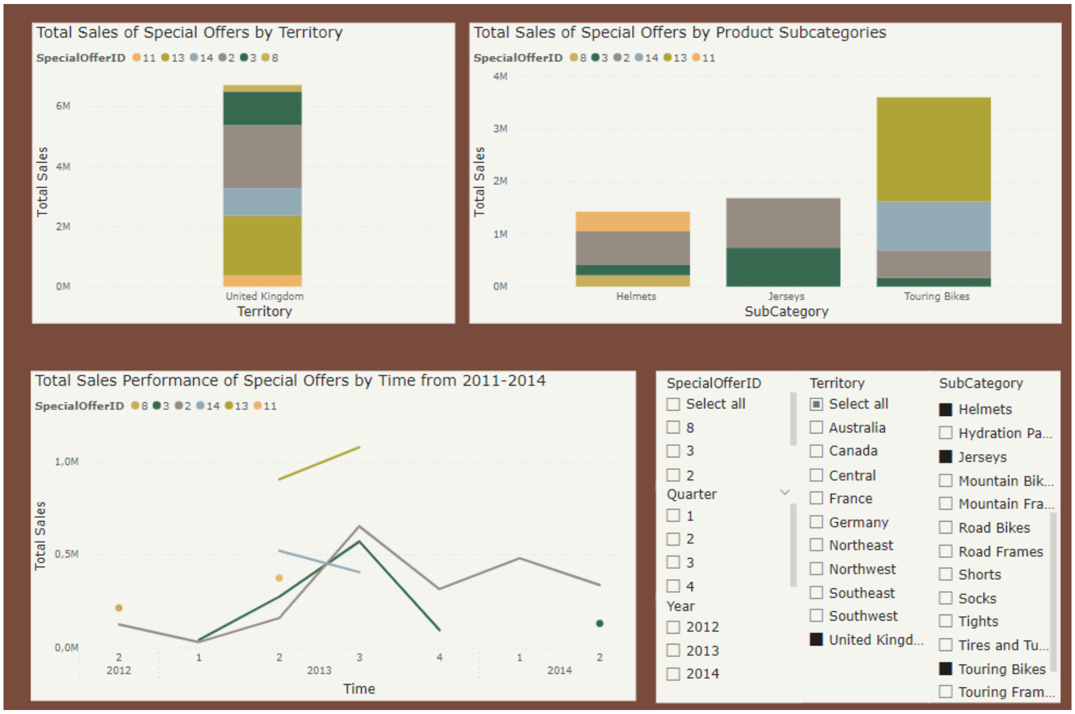
The company is looking for effective promotional strategies for the United Kingdom territory by leveraging a BI system to recommend special offers on specific products during specific time periods to optimize performance in the United Kingdom.



Overview performance of United Kingdom territory by PowerBI

1. Get results from BI Dashboard

According to the PowerBI visualization, which filters for the United Kingdom territory, special offers 2, 3, 4, 8, 9, 11, 12, 14, and 16 were implemented from quarter 2 in 2012 to quarter 2 in 2014. The company achieved 14 million in total sales and sold 2.8K products during this period. Notably, Special Offer ID 2 yielded the highest total sales across all product types, followed by Special Offers ID 13. The top three products in terms of total sales were touring bikes, jerseys, and helmets.


Total Sales of Special Offers by Territory
SpecialOfferID ●11 ●13 ●14 ●2 ●3 ●8
Total Sales
6M
4M
2M
0M
United Kingdom
Territory
Total Sales of Special Offers by Product Subcategories
SpecialOfferID ●8 ●3 ●2 ●14 ●13 ●11
Total Sales
4M
3M
2M
1M
0M
Helmets
Jerseys
Touring Bikes
SubCategory
Total Sales Performance of Special Offers by Time from 2011-2014
SpecialOfferID ●8 ●3 ●2 ●14 ●13 ●11
Total Sales
1,0M
0,5M
0,0M
2
2012
1
2
2013
3
4
1
2014
2
Time

SpecialOfferID
☐ Select all
☐ 8
☐ 3
☐ 2

Quarter
☐ 1
☐ 2
☐ 3
☐ 4

Year
☐ 2012
☐ 2013
☐ 2014

Territory
☒ Select all
☐ Australia
☐ Canada
☐ Central
☐ France
☐ Germany
☐ Northeast
☐ Northwest
☐ Southeast
☐ Southwest
■ United Kingd...

SubCategory
■ Helmets
☐ Hydration Pa...
■ Jerseys
☐ Mountain Bik...
☐ Mountain Fra...
☐ Road Bikes
☐ Road Frames
☐ Shorts
☐ Socks
☐ Tights
☐ Tires and Tu...
■ Touring Bikes
☐ Touring Fram...

BI Dashboard of the United Kingdom territory after filtering helmets, jerseys, and touring bikes

Upon filtering for helmets, jerseys, and touring bikes, it was observed that Special Offer ID 2 generated the highest total sales for helmets and jerseys, while Special Offer ID 13 was most effective for touring bikes. For Special Offers ID 2, the period with the highest total sales was from the first to the third quarter of 2013. For Special Offer ID 13, the period with the highest total sales was from the second to the third quarter of 2013.

2. Strategic recommendations

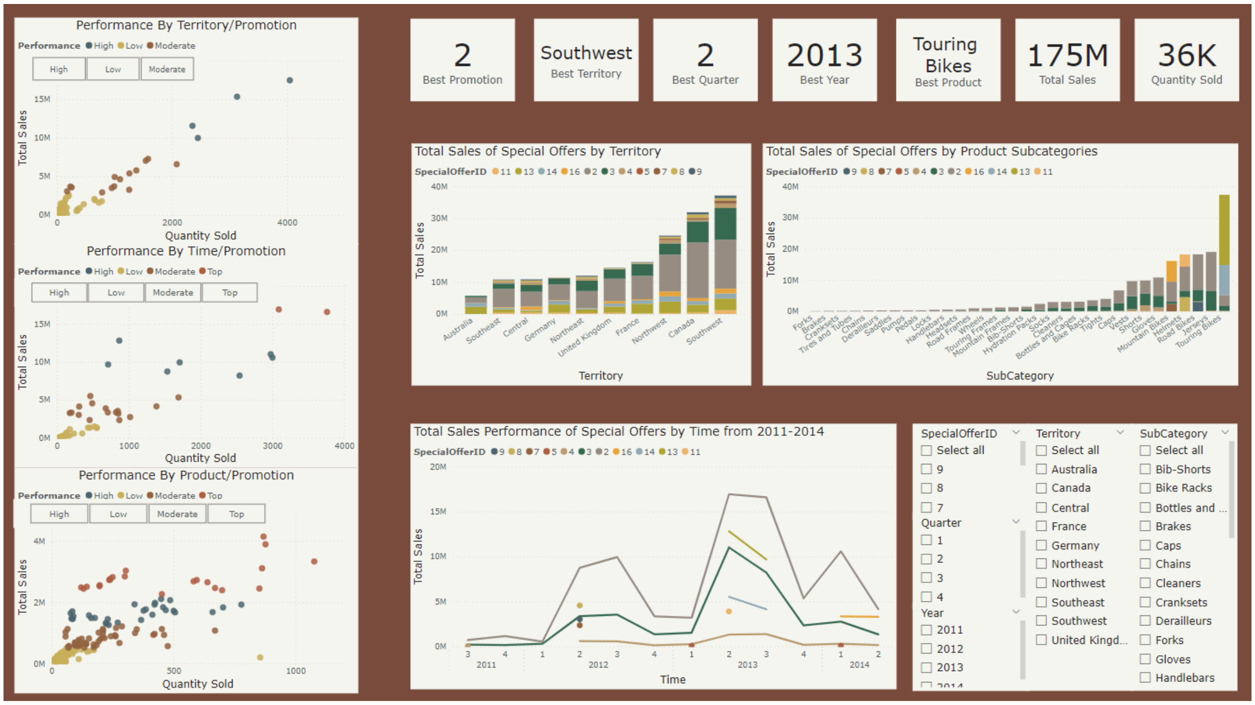
- **Expand Successful Special Offers:** Continue or expand Special Offers 2 for helmets, and jerseys, and Special Offers 13 for touring bikes.
- **Seasonal Sales Strategies:** Given that the highest total sales were achieved from Quarter 1 to 3 in 2013 for Special Offers ID 2, and from Quarter 2 to 3 in 2013 for Special Offer ID 13, it would be beneficial to align promotional strategies with these periods.
- **Focus on Top-Selling Products:** Given the high sales of touring bikes, jerseys, and helmets, these products should be the focus of marketing efforts.

In conclusion, to optimize performance for the United Kingdom territory, it's recommended to continue or expand Special Offers 2 for helmets and jerseys, and Special Offer 13 for touring bikes, align promotional strategies with high-sales periods (Quarter 1 to Quarter 3 in 2013 for

Special Offer 2, and Quarter 2 to Quarter 3 in 2013 for Special Offer 13), and focus marketing efforts on the top-selling products: touring bikes, jerseys, and helmets.

X. Discussion

1. Overview



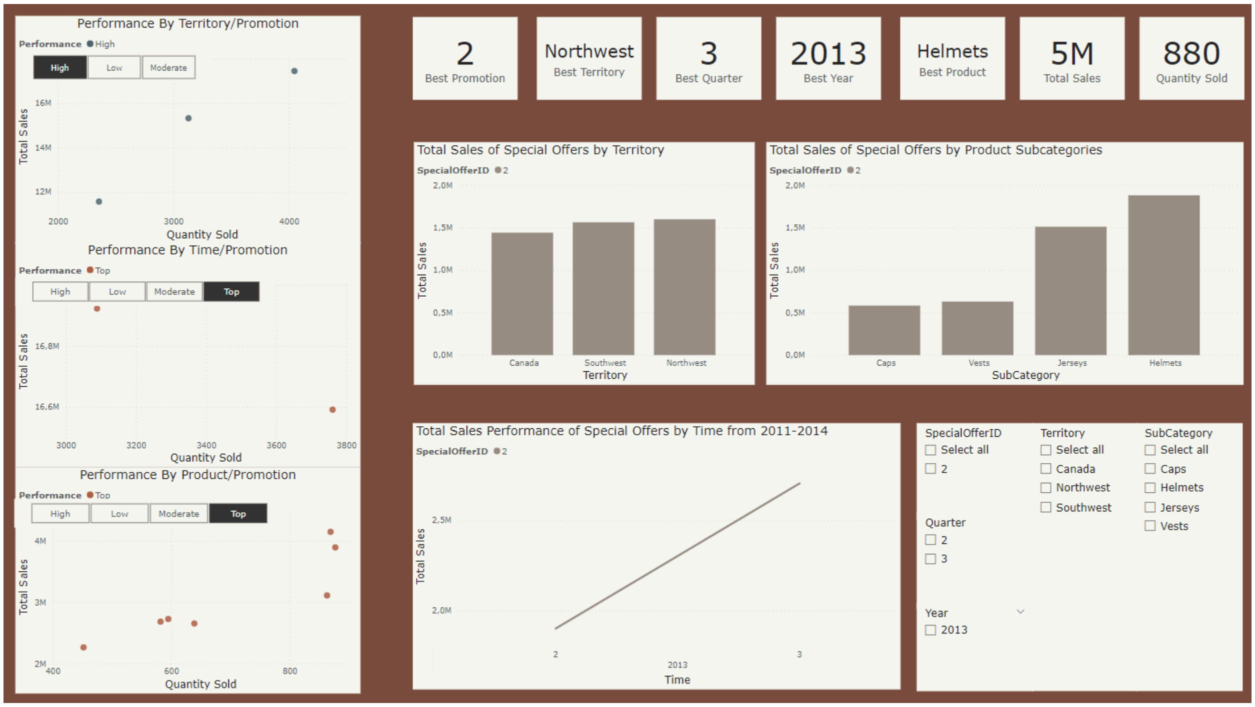
- **CompanyX Performance (2011-2014):** Over four years, from 2011 to 2014, CompanyX achieved total sales of 175.1 million units of currency and sold a total of 36.0K products.
- **Best Performing Special Offer:** The special offer with ID 2 has been identified as the most effective, contributing significantly to the company's sales.
- **Top Selling Territory:** The Southwest Territory recorded the highest total sales and product quantities.
- **Seasonal and Annual Sales Trends:** The data shows that summer and the year 2013 were the most successful times for sales and product numbers.
- **Most Sold Product:** Touring bikes emerged as the most sold product.

2. Top Performance and Strategic Recommendation for CompanyX

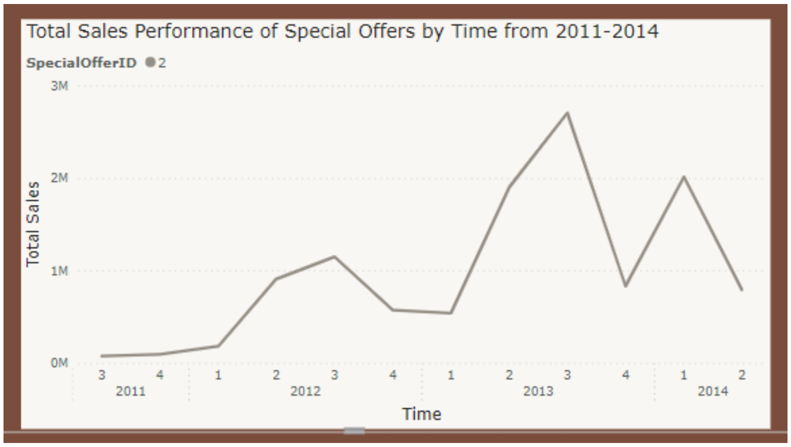
2.1. Get results from BI Dashboard

The visualization of PowerBI reveals that the territories of **Canada, Southwest, and Northwest** are the top performers in terms of total sales and quantities (High-performance cluster of 'Promotion by Territory' scatter chart). In these territories, the products that have the highest total sales and quantities are **caps, vests, jerseys, and helmets** (Top-performance cluster of

‘Promotion by Product’ scatter chart). The special offer that has been applied and shown to be effective is **Special Offer 2**. The period that has the highest total sales and total quantities is **from Quarter 2 to 3 in 2013**. (Top-performance cluster of ‘Promotion by Time’ scatter chart)



Moreover, focusing on the total sales performance of Special Offer 2 for products caps, vests, jerseys, and helmets, it is noticeable that **from Quarter 2 to 3 in both 2012 and 2013**, the total sales witnessed the **same upward trend.**



2.2. Strategic recommendations are proposed for CompanyX

- **Expand Successful Special Offer 2:** Special Offer 2 has been particularly effective in driving sales, especially in the territories of Canada, Southwest, and Northwest. It's recommended to continue or even expand this special offer.
- **Focus on High-Performing Territories and Products:** The territories of Canada, Southwest, and Northwest have shown the highest total sales and quantities, particularly for the products caps, vests, jerseys, and helmets. Tailoring special offers or marketing strategies to these specific territories and products could boost sales.
- **Seasonal Sales Strategies:** The data indicates that the period from Quarter 2 to 3 was particularly successful for sales. Exploring seasonal promotions or sales strategies that align with this period could be beneficial.
- **Monitor Sales Data:** Regularly monitor and analyze sales data to track the performance of products, special offers, and territories. Use these insights to make informed business decisions and adjust strategies as necessary.

In conclusion: To boost performance for companyX, it's recommended to expand Special Offer 2, focus on high-performing territories (Canada, Southwest, Northwest) and products (caps, vests, jerseys, helmets), and implement seasonal strategies targeting Quarter 2 to Quarter 3.

XI. Conclusion

This study has demonstrated the potential of a business intelligence system in enhancing the promotional strategies of CompanyX. By systematically analyzing data related to territories, special offers, and product names, along with their corresponding total sales and total quantities, we were able to segregate the data into distinct performance-based clusters.

These clusters provided valuable insights into the common characteristics and trends of the territories, special offers, and products. Based on these insights, we developed strategic recommendations suggesting which special offers could be applied to specific products and territories.

1. **Achievements:**

- Demonstrated the potential of a business intelligence system in enhancing the promotional strategies of CompanyX.
- Systematically analyzed data related to territories, special offers, and product names, along with their corresponding total sales and total quantities.
- Segregated the data into distinct performance-based clusters.
- Provided valuable insights into the common characteristics and trends of the territories, special offers, and products.
- Developed strategic recommendations suggesting which special offers could be applied to specific products and territories.
- Continually assessed and refined these strategies for optimal results.
- Contributed significantly to the strategic decision-making process at CompanyX.
- Highlighted the importance of leveraging data analytics in formulating effective promotional strategies.
- Contributed to improved sales growth and market presence.

2. **Limitations and Future Research**

The study is based on historical data, and the future performance of the strategies may vary. The effectiveness of the strategies may be influenced by external factors not accounted for in the study. Future research could consider the potential impact of new products or changes in market trends, customer behavior over time, and competitors' actions and strategies.

Preferences

1. Beregovskaya, I., & Koroteev, M. (2021). Review of Clustering-Based Recommender Systems. (PDF) Review of Clustering-Based Recommender Systems (researchgate.net)
2. Braak, P. T., Abdullah, N., & Xu, Y. (2009). Improving the Performance of Collaborative Filtering Recommender Systems through User Profile Clustering. Improving the Performance of Collaborative Filtering Recommender Systems through User Profile Clustering | IEEE Conference Publication | IEEE Xplore