Graphical user interface

Description automatically generated with low confidence

Integração e Processamento Analítico de Informação Project

**Stage 3: ETL system and reports**

**Group 1**

Tiago Rodrigues nº49593

Ivo Oliveira nº50301

Martim Silva nº51304

Alexandre Sobreira nº59451

2022/2023

**Table of Contents**

[- Introduction 4](#_Toc136118066)

[1 - First Stage 5](#_Toc136118067)

[1.1 - Identify the data sources used 5](#_Toc136118068)

[1.1.1 - Global Superstore Orders 2016 5](#_Toc136118069)

[1.1.2 - Global Superstore Returns 2016 9](#_Toc136118070)

[1.1.3 - Superstore Dataset 10](#_Toc136118071)

[1.1.4 - US Holiday Dates 13](#_Toc136118072)

[1.1.5 - Highest GDP Counties in the USA 13](#_Toc136118073)

[1.2 - Analyse values and errors in the data fields of each source 15](#_Toc136118074)

[1.2.1 - Orders table 16](#_Toc136118075)

[1.2.2 - Customers table 16](#_Toc136118076)

[1.2.3 - Returns table 16](#_Toc136118077)

[1.2.4 - Sellers table 16](#_Toc136118078)

[1.2.5 - Holiday USA table 16](#_Toc136118079)

[1.2.6 - USA GDP table 16](#_Toc136118080)

[1.3 - Draw a diagram with links between data sources 17](#_Toc136118081)

[1.4 - Describe a business process to take advantage of data 17](#_Toc136118082)

[1.5 - Define three analytical questions for the business process 18](#_Toc136118083)

[1.5.1 - Sellers Analysis 18](#_Toc136118084)

[1.5.2 - Customer analysis 18](#_Toc136118085)

[1.5.3 - Product analysis 19](#_Toc136118086)

[1.6 – Conclusion of the First Stage 19](#_Toc136118087)

[2 - Second Stage 20](#_Toc136118088)

[2.1 – Fix problems detected in the previous stage 20](#_Toc136118089)

[2.2 – Declare the grain and type of the fact table 21](#_Toc136118090)

[2.3 – Model business dimensions, including data hierarchies 22](#_Toc136118091)

[2.3.1 - Facts Table 23](#_Toc136118092)

[2.3.2 - Product Dimension 23](#_Toc136118093)

[2.3.3 - Date Dimension (Role-Playing – Order & Ship) 24](#_Toc136118094)

[2.3.4 - Holiday Dimension (Outrigger - Connected to Date Dimension) 26](#_Toc136118095)

[2.3.5 - Seller Dimension 26](#_Toc136118096)

[2.3.6 - Customer Dimension 27](#_Toc136118097)

[2.3.7 - GDP Dimension (Outrigger - Connected to Customer Dimension) 28](#_Toc136118098)

[2.3.8 - Order Information Dimension 29](#_Toc136118099)

[2.4 – Identify numerical measurements in the fact table(s) 30](#_Toc136118100)

[2.5 – Draw the data warehouse star diagram 35](#_Toc136118101)

[2.6 - Conclusion of the Second Stage 37](#_Toc136118102)

[3 - Third Stage 38](#_Toc136118103)

[3.1 - Fix problems detected in the previous stage 38](#_Toc136118104)

[3.2 – Staging Area 39](#_Toc136118105)

[3.2.1 - Extraction 40](#_Toc136118106)

[3.2.2 - Transformation 40](#_Toc136118107)

[3.2.3 - Loading 44](#_Toc136118108)

[3.2.4 - Draw a diagram with data flows and ETL system programs 45](#_Toc136118109)

[3.2.5 - Show dimensions and fact table of the implemented data cube 47](#_Toc136118110)

[3.3 - Presentation Area 49](#_Toc136118111)

[3.4 - Produce annotated reports that answer the analytical questions (pp.86) 50](#_Toc136118112)

[3.4.1 - Sellers Analysis 50](#_Toc136118113)

[3.4.2 - Customer Analysis 55](#_Toc136118114)

[3.4.3 - Product Analysis 58](#_Toc136118115)

[3.5 - Conclusion of the Third Phase 62](#_Toc136118116)

[3.5.1 - ETL system 62](#_Toc136118117)

[3.5.2 - Analytical questions 62](#_Toc136118118)

[3.5.2.1 - Seller 62](#_Toc136118119)

[3.5.2.2 - Customer 62](#_Toc136118120)

[3.5.2.3 - Product 63](#_Toc136118121)

Introduction

The IPAI project is an extensive endeavour focused on modelling and constructing a data warehouse that effectively integrates data from diverse sources within a specific time frame. Its primary objective is to develop a robust platform that facilitates informed decision-making processes. The project involves comprehensive research and consultation of open data sources, with a specific emphasis on selecting a business process. It is divided into three distinct stages, each crucial for the successful construction, development, and analysis of the data warehouse.

The first stage revolves around meticulous data analysis, encompassing the identification of relevant data sources associated with the chosen business process. This phase entails analysing the data and establishing links between different sources, visualized through a diagram. Its key objectives include identifying the data sources utilized, analysing values, and detecting errors in the data fields of each source, drawing a comprehensive diagram illustrating the connections between data sources, describing a business process to leverage the available data, and formulating three analytical questions pertinent to the business process. This stage serves as the foundation for subsequent phases, ensuring the identification of relevant data sources, assessing data quality, understanding data relationships, and establishing a framework for informed decision-making and successful project implementation.

The second stage focuses on dimensional modelling, a critical step in creating a multidimensional model tailored for a data warehouse. The primary objectives of this stage encompass declaring the granularity and type of fact table(s), modelling business dimensions that include data hierarchies, identifying numerical measurements within the fact table(s), and constructing a data warehouse star diagram. By prioritizing dimensional modelling, the project team establishes a solid groundwork for subsequent stages, including ETL system development and analytical reporting. This approach ensures that the data warehouse is designed to support efficient data retrieval, analysis, and decision-making. The dimensional model enables users to explore data from various perspectives, comprehend relationships between data elements, and derive valuable insights from the data stored within the warehouse.

Lastly, the third stage concentrates on developing the ETL system and generating analytical reports to address the questions identified in the first stage. This phase necessitates the implementation of automation processes for extracting data from repositories, transforming it within the data staging area, and loading the treated data into the data presentation area. The principal objectives of this stage involve developing and testing the programs comprising the ETL system, describing the responsibilities, inputs, and outputs of each program or process, creating a diagram depicting data flows and ETL system programs, showcasing the dimensions and fact table of the implemented data cube, and producing annotated reports that effectively answer the analytical questions. This final stage ensures efficient data processing, accurate data representation, and insightful reporting, empowering stakeholders to access relevant information promptly, make informed decisions, and achieve the project's goals.

Throughout the subsequent stages, the project team will address and resolve several identified issues between stages, which will be duly documented in this report.

# 1 - First Stage

The first stage of the data analysis process is focused on meticulous data analysis, with a primary emphasis on identifying relevant data sources associated with the chosen business process. This phase entails a comprehensive examination of the data, establishing connections between different sources, and visualizing these relationships through a diagram.

Firstly, we systematically identify the data sources utilized in the business process. This involves gaining an understanding of the nature of the data and its origin, ensuring we have a comprehensive view of the available information.

Next, we conduct a thorough analysis of the data fields within each source. By scrutinizing the values present, we aim to detect any errors or inconsistencies that may impact the accuracy and reliability of the data. This step is crucial for ensuring the integrity of the information we will be working with.

As we proceed, we construct a comprehensive diagram that illustrates the connections between the various data sources. This visual representation enables us to gain insights into the interdependencies and relationships between the data elements, facilitating a deeper understanding of how they interact.

Additionally, we provide a clear and concise description of the business process that will leverage the available data. This step ensures that we have a contextual understanding of how the data will be utilized, guiding our subsequent analysis, and aligning our efforts with the specific objectives of the business process.

Lastly, we formulate three analytical questions that are directly relevant to the business process. These questions serve as guiding principles for the upcoming stages of the analysis. By addressing these questions, we can derive meaningful insights and develop actionable recommendations that contribute to the success of the project.

## 1.1 - Identify the data sources used

This project will aim to aid a big e-commerce chain in improving its business by analysing various datasets that are related directly or indirectly to it. With this goal in mind, four data sets will be analysed together. The four datasets are Global Superstore Orders 2016; Global Superstore Returns 2016; Superstore Dataset; US Holiday Dates (2004-2021); Highest GDP Counties in the USA. All the analysis in this step was done using Python 3 with the aid of packages such as pandas, NumPy, seaborn and matplotlib. The four datasets will be described in the following steps.

### 1.1.1 - Global Superstore Orders 2016

This dataset can be found in the open source [data.world](https://data.world/tableauhelp/superstore-data-sets). It has a total of 8.08 MB in xlsx format, containing two tables, Orders and People, which we will call Sellers. This dataset includes information regarding the sales of the large-scale e-commerce chain.

#### 1.1.1.1 - Orders

The initial dataset table is called "Orders" and it serves as the primary source of information for this project. The table includes 51290 rows and 24 columns, which are outlined below.

Table 1 Orders table column number (#), name, description, and data type

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **Description** | **Data type** |
| **1** | **Row ID** | Unique ID for each row. | Int64 |
| **2** | **Order ID** | Unique Order ID for each Customer. | Object |
| **3** | **Order Date** | Order Date of the product. | Datetime64[ns] |
| **4** | **Ship Date** | Shipping Date of the Product. | Datetime64[ns] |
| **5** | **Ship Mode** | Shipping Mode specified by the customer. | Object |
| **6** | **Customer ID** | Unique ID to identify each customer. | Object |
| **7** | **Customer Name** | Name of the Customer. | Object |
| **8** | **Segment** | The segment where the customer belongs. | Object |
| **9** | **Postal Code** | Postal Code of every Customer | Float64 |
| **10** | **City** | City of residence of the seller. | Object |
| **11** | **State** | State of residence of the seller. | Object |
| **12** | **Country** | Country of residence of the seller. | Object |
| **13** | **Region** | The region where the seller belongs. | Object |
| **14** | **Market** | Global location by Market | Object |
| **15** | **Product ID** | Unique ID of the Product | Object |
| **16** | **Category** | The Category of the product ordered. | Object |
| **17** | **Product Name** | Name of the Product | Object |
| **18** | **Sales** | Sales of the Product. | Float64 |
| **19** | **Quantity** | Quantity of the Product. | Int64 |
| **20** | **Discount** | Discount provided. | Float64 |
| **21** | **Profit** | Profit/Loss incurred. | Float64 |
| **23** | **Shipping Cost** | Cost per shipment | Float64 |
| **24** | **Order Priority** | Level of priority for the order | Object |

Table 1 shows that the Orders table comprises various data types, including order and shipment dates, customer and seller information (e.g., region), and product details such as category and profit.

For clarity, the variables in the Orders table were divided into categorical and numerical categories to aid in their description, which can be seen below.

Table 2 Orders table categorical variables description with a column name, number of unique instances and example of content.

|  |  |  |
| --- | --- | --- |
| **Column** | **Unique** | **Example** |
| **Order ID** | 25728 | CA-2015-SV20365140-42268 |
| **Order Date** | 1430 | 2015-12-31 0:00 |
| **Ship Date** | 1464 | 2016-01-07 0:00 |
| **Ship Mode** | 4 | Standard Class |
| **Customer ID** | 17415 | SV-203651406 |
| **Customer Name** | 796 | Muhammed Yedwab |
| **Segment** | 3 | Consumer |
| **Country** | 165 | United States |
| **City** | 3650 | New York City |
| **State** | 1102 | California |
| **Region** | 23 | Western Europe |
| **Product ID** | 3788 | OFF-FA-6129 |
| **Category** | 3 | Office Supplies |
| **Sub-Category** | 17 | Binders |
| **Product Name** | 3788 | Staples |
| **Market** | 5 | Asia Pacific |
| **Order Priority** | 4 | Medium |

Table 3 Orders table numerical variables mean, standard deviation (STD), minimum and maximum values and quantiles (25%, 50%, 75%)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Sales ($)** | **Quantity** | **Discount (%)** | **Profit** | **Shipping Cost ($)** |
| **Mean** | 246.491 | 3.477 | 0.143 | 28.611 | 26.479 |
| **STD** | 487.565 | 2.279 | 0.212 | 174.341 | 57.251 |
| **Min** | 0.444 | 1 | 0 | -6599.978 | 1.002 |
| **25%** | 30.759 | 2 | 0 | 0 | 2.61 |
| **50%** | 85.053 | 3 | 0 | 9.24 | 7.79 |
| **75%** | 251.053 | 5 | 0.2 | 36.81 | 24.45 |
| **Max** | 22638.48 | 14 | 0.85 | 8399.976 | 933.57 |

To gain a better understanding of the data in the Orders table, several plots were generated to examine the number of orders according to various factors, including Ship Mode, Segment, Region, Category, and Sub-Category. These plots are presented below.

Chart, bar chart

Description automatically generated

Figure 1 Frequency of Orders by Shipping Modes

Chart, bar chart

Description automatically generated

Figure 2 Frequency of Orders by Segment

Chart, bar chart

Description automatically generated

Figure 3 Frequency of Orders by Region

Chart, bar chart

Description automatically generated

Figure 4 Frequency of Orders by Category

Chart, bar chart

Description automatically generated

Figure 5 Frequency of Orders by Sub-Category

The above plots offer valuable insights. Firstly, it's evident that Standard Class Shipping is the most preferred mode of shipment amongst customers. Secondly, the majority of the customers belong to the Consumer Segment. Thirdly, the Central Asia region accounts for the least number of orders, while Western Europe dominates the sales. Moreover, the sales data indicate that Office Supplies is the most popular product category. Lastly, it's interesting to note that Blinders and Paper emerge as clear leaders in sales among customers.

#### 1.1.1.2 - Sellers

The second table of this dataset is the table Sellers. This table contains 24 rows and two columns, which are presented below.

Table 4 Sellers table column number (#), name, description and data type

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **Description** | **Data type** |
| **1** | **Person** | Name of the Seller | Object |
| **2** | **Region** | The region where the seller belongs. | Object |

As it is possible to observe in Table 4, the table Sellers contains information regarding the name of the seller and the corresponding location (i.e., region)

To better understand the Sellers table variables, a division was made between categorical and numerical variables so they could be easier described, which can be seen below.

Table 5 Sellers table categorical variables description with a column name, number of unique instances, and example of content.

|  |  |  |
| --- | --- | --- |
| **Column** | **Unique** | **Example** |
| **Person** | 24 | Marilène Rousseau |
| **Region** | 24 | Caribbean |

Since no numerical variables were present in this table, creating a plot with the provided information would not yield much insight.

### 1.1.2 - Global Superstore Returns 2016

This dataset can also be found in the open source [data.world](https://data.world/tableauhelp/superstore-data-sets). It has a total of 45.03 KB in CSV format, containing one table, Returns. This dataset also contains information regarding the sales of the large-scale e-commerce chain.

#### 1.1.2.1 - Returns

The table Returns displays 1079 rows and three columns, which are presented below.

Table 6 Returns table column number (#), name, description and data type.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **Description** | **Data type** |
| **1** | **Returned** | Boolean that refers if the order was or was not returned | Object |
| **2** | **Order ID** | Unique Order ID for each Customer. | Object |
| **3** | **Region** | The region where the seller belongs | Object |

In Table 6 Returns contains information that shows if a certain order was returned or not, as well as the region of the seller of that order.

For a better understanding of the Returns table variables, a division was made between categorical and numerical variables so they could be easier described, which can be seen below.

Table 7 Returns table categorical variables description with a column name, number of unique instances, and example of content.

|  |  |  |
| --- | --- | --- |
| **Column** | **Unique** | **Example** |
| **Returned** | 1 | Yes |
| **Order ID** | 1079 | CA-2012-SA20830140-41210 |
| **Region** | 24 | Western Europe |

Since there were no numerical variables present in this table, creating a plot with the provided information would not yield much insight.

### 1.1.3 - Superstore Dataset

This dataset can be found on the open-source Kaggle website ([Superstore Dataset | Kaggle](https://www.kaggle.com/datasets/vivek468/superstore-dataset-final)). It has a total of 2.29 MB in CSV format, containing one table, which we will call customers\_USA. This dataset also contains information regarding the sales of the large-scale e-commerce chain but now with a focus on the United States of America-based customers.

The table custumers\_USA displays 51290 rows and 24 columns, which are presented below.

Table 8 customers\_USA table column number (#), name, description, and data type

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **Description** | **Data type** |
| **1** | **Row ID** | Unique ID for each row. | Int64 |
| **2** | **Order ID** | Unique Order ID for each Customer. | Object |
| **3** | **Order Date** | Order Date of the product. | Object |
| **4** | **Ship Date** | Shipping Date of the Product. | Object |
| **5** | **Ship Mode** | Shipping Mode specified by the customer. | Object |
| **6** | **Customer ID** | Unique ID to identify each customer. | Object |
| **7** | **Customer Name** | Name of the Customer. | Object |
| **8** | **Segment** | The segment where the customer belongs. | Object |
| **9** | **Country** | Country of residence of the customer. | Object |
| **10** | **City** | City of residence of the customer. | Object |
| **11** | **State** | State of residence of the customer. | Object |
| **12** | **Postal Code** | Postal Code of every Customer | Int64 |
| **13** | **Region** | The region where the customer belongs. | Object |
| **14** | **Product ID** | Unique ID of the Product. | Object |
| **15** | **Category** | Category of the product ordered. | Object |
| **16** | **Sub-Category** | The sub-Category of the product ordered. | Object |
| **17** | **Product Name** | Name of the Product | Object |
| **18** | **Sales** | Sales of the Product. | Float64 |
| **19** | **Quantity** | Quantity of the Product. | Int64 |
| **20** | **Discount** | Discount provided. | Float64 |
| **21** | **Profit** | Profit/Loss incurred. | Float64 |

As it is possible to observe in Table 8, the table customers\_USA contains information similar to the Orders table, which goes from details about the dates of the orders and shipments, products categories and profits, with the difference that this table doesn't contain information regarding the seller, being its focus on the customer (USA based), providing more information, such as region, city, state, name, etc.

To better understand the customers\_USA table variables, a division was made between categorical and numerical variables so they could be easier described, which can be seen below.

Table 9 customers\_USA table categorical variables description with a column name, number of unique instances, and example of content.

|  |  |  |
| --- | --- | --- |
| **Column** | **Unique** | **Example** |
| **Order ID** | 5009 | CA-2017-100111 |
| **Order Date** | 1237 | 9/5/2016 |
| **Ship Date** | 1334 | 12/16/2015 |
| **Ship Mode** | 4 | Standard Class |
| **Customer ID** | 793 | WB-21850 |
| **Customer Name** | 793 | William Brown |
| **Segment** | 3 | Consumer |
| **Country** | 1 | United States |
| **City** | 531 | New York City |
| **State** | 49 | California |
| **Region** | 4 | West |
| **Product ID** | 1862 | OFF-PA-10001970 |
| **Category** | 3 | Office Supplies |
| **Sub-category** | 17 | Binders |
| **Product Name** | 1850 | Staple envelope |
| **Postal Code** | 631 | 42420 |

Table 10 customers\_USA table numerical variables mean, standard deviation (STD), minimum and maximum values and quantiles (25%, 50%, 75%)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sales ($)** | **Quantity** | **Discount (%)** | **Profit ($)** |
| **Mean** | 229.858 | 3.789 | 0.156 | 28.657 |
| **STD** | 623.245 | 2.225 | 0.206 | 234.26 |
| **Min** | 0.444 | 1 | 0 | -6599.978 |
| **25%** | 17.28 | 2 | 0 | 1.729 |
| **50%** | 54.49 | 3 | 0.2 | 8.667 |
| **75%** | 209.94 | 5 | 0.2 | 29.364 |
| **Max** | 22638.48 | 14 | 0.8 | 8399.976 |

Some plots were made to better understand the shape of the data present in the customers\_USA table. The number of customers was analyzed by State, City and Region. The various plots can be seen below.

Map

Description automatically generated

Figure 6 Number of Customers by State Map

Chart

Description automatically generated

Figure 7 Number of Customers by State (Top 25)

Chart

Description automatically generated

Figure 8 Number of Customers by City (Top 25)

Chart, bar chart

Description automatically generated

Figure 9 Number of Customers by Region

Regarding these plots, three orders of insights can be taken. The three most popular states among the customers are California, New York and Texas, and the least is Connecticut. The most popular city among the customers is New York City, and the 25th is Rochester. And most popular region among customers is the West region, and the least popular is the South region.

### 1.1.4 - US Holiday Dates

This dataset can be found on the open-source Kaggle website ([US Holiday Dates (2004 - 2021) | Kaggle](https://www.kaggle.com/datasets/donnetew/us-holiday-dates-2004-2021)). It has a total of 15.7 kB in CSV format, containing one table, which we will call holiday\_USA, with information regarding the dates of US holidays.

The table holiday\_USA displays 342 rows and six columns, presented below.

Table 11 holiday\_USA table column number (#), name, description and data type

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **Description** | **Data type** |
| **1** | **Date** | Date corresponding to holiday | Object |
| **2** | **Holiday** | Holiday designation | Object |
| **3** | **WeekDay** | Day of the week when the holiday occurs | Object |
| **4** | **Month** | The month when the holiday occurs | Int64 |
| **5** | **Day** | Day of the month when the holiday occurs | Int64 |
| **6** | **Year** | Year of reference [2004-2021] | Int64 |

As it is possible to observe in Table 10, table holiday\_USA contains a list of holidays that includes 18 years of US Holidays dated between 2004 and 2021. Each record has a Date, Holiday, Weekday, Month, Day and Year.

For a better understanding of the holiday\_USA table variables, a division was made between categorical and numerical variables so they could be easier described, which can be seen below.

Table 12 holiday\_USA table categorical variables description with a column name, number of unique instances and example of content.

|  |  |  |
| --- | --- | --- |
| **Column** | **Unique** | **Example** |
| **Date** | 336 | 2007-04-08 |
| **Holiday** | 18 | Labour Day Weekend |
| **WeekDay** | 7 | Monday |

Since there were no numerical variables present in this table, creating a plot with the provided information would not yield much insight.

### 1.1.5 - Highest GDP Counties in the USA

This dataset can also be found on the open-source Kaggle website ([Highest GDP Counties in USA | Kaggle](https://www.kaggle.com/datasets/anasmahmood000/highest-gdp-counties-in-usa)). It has a total of 3.65 MB in CSV format, containing one table, which we will call GDP\_USA. This dataset also includes data on the GDP of counties in the United States

The table GDP\_USA displays 55501 rows and eight columns, presented below.

Table 13 GDP\_USA table column number (#), name, description and data type.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Column** | **Description** | **Data Type** |
| **1** | **index** |  | Int64 |
| **2** | **Year** | Year of reference | Int64 |
| **3** | **Region** | The region of reference for the GDP | Object |
| **4** | **SUB\_REGION** | The sub-region of reference for the GDP | Object |
| **5** | **State** | The state of reference for the GDP | Object |
| **6** | **County** | The county of reference for the GDP | Object |
| **7** | **GDP(Chained $)** | GDP value in $ | Float64 |

As it is possible to observe in Table 12, the table GDP\_USA contains data on the GDP in dollars, of counties in the United States, organized by year, region, sub-region, and county.

To better understand the GDP\_USA table variables, a division was made between categorical and numerical variables so they could be easier described, which can be seen below.

Table 14 GDP\_USA table categorical variables description with a column name, number of unique instances and example of content.

|  |  |  |
| --- | --- | --- |
| **Column** | **Unique** | **Example** |
| **Year** | 18 | 2001 |
| **Region** | 8 | Southeast |
| **SUB\_REGION** | 9 | West North Central |
| **State** | 51 | Texas |
| **County** | 1801 | Washington |

Table 15 GDP\_USA table numerical variables mean, standard deviation (STD), minimum and maximum values and quantiles (25%, 50%, 75%).

|  |  |
| --- | --- |
|  | **GDP (Chained $)** |
| **Mean** | 5.05E+09 |
| **STD** | 2.15E+10 |
| **Min** | 9.95E+06 |
| **25%** | 3.37E+08 |
| **50%** | 8.61E+08 |
| **75%** | 2.49E+09 |
| **Max** | 7.11E+11 |

To better understand the shape of the data present in the GDP\_USA table, some plots were done. The GDP (Chained $) was analyzed by State and Region. The various plots can be seen below.

Chart

Description automatically generated

Figure 10 GDP by State (Top 25)

Chart, bar chart

Description automatically generated

Figure 11 GDP by Region

It is possible to observe that the state with the highest GDP is the state of California, and the 25th is Alaska. In terms of regions, the one with the highest GDP is the Southeast, and the one with the lowest is the Rocky Mountains.

## 1.2 - Analyse values and errors in the data fields of each source

Data standardisation is a critical step in data analysis and pre-processing (Gal & Rubinfeld, 2018). It involves bringing different data sources into a common format, allowing for meaningful comparisons and accurate analysis (Aggarwal, 2015). A critical aspect of data standardisation is analysing each feature's values and errors of all the used data sources. This involves carefully examining the data to identify any inconsistencies or anomalies and ensuring that the data is reliable and accurate (Gal & Rubinfeld, 2018). By standardising the data, organisations can make better-informed decisions based on consistent, reliable data, leading to improved efficiency and better outcomes (Gal & Rubinfeld, 2018).

Performing the analysis of each feature's values and errors of all the used data sources is a crucial step in data standardisation, and this can be achieved using programming languages such as Python. Python is a widely used language for data analysis, with numerous libraries and tools available that make it an ideal choice for standardising data.

In this context, the analysis of each feature's values and errors will cover all four tables, namely the orders, returns, sellers, customers\_USA, holiday\_USA and GDP\_USA tables. These tables contain different types of data, including transactional data, customer data, and sales data. The analysis will involve checking for any missing or inconsistent data, identifying any outliers or anomalies, and addressing any issues that may affect the data quality.

Using Python for this analysis makes it possible to automate the process and perform it more efficiently. This ensures that the standardisation of data is not only accurate but also timely. Additionally, using Python makes it easier to visualise and communicate the results of the analysis, allowing for better decision-making based on reliable data.

In some cases, a reoccurring situation that occurs in the tables is that some features are mistakenly represented as object data types when they should be represented as a different kind of data type. Object data types can be used for a variety of data, including text and alphanumeric characters. However, in certain scenarios, they may not be the most efficient or appropriate data type to use.

Using the wrong data type can result in a range of issues, including increased memory usage, reduced processing speed, and decreased accuracy (Tabatabaei et al., 2021). For example, if a feature such as a date is represented as an object data type instead of a datetime data type, it may be challenging to perform certain types of analysis on the data, such as sorting or filtering by date.

### 1.2.1 - Orders table

With the goal of standardising the data, a thorough analysis was performed on the table orders, which revealed several data type inconsistencies. First, some features that were meant to be dates were represented as object data types, prompting a change to the appropriate date format. Similarly, all information regarding the IDs (Order ID and Customer ID) was classified as an object data type, but to avoid potential issues stemming from mixed numerical and alphabetical characters, it was converted to a string data type. The same process was applied to other features like the City, State, Country, Region, Market, Product Name

Furthermore, there are other columns in the 'orders' table, such as Segment, Category, Sub-Category, and Order Priority, that are currently represented as an object data type but could benefit from a change to a category data type. This type of conversion can offer performance and memory benefits, as well as make it easier to perform certain types of analysis on the data.

Lastly, it was found that the postal code column had more than 50% of missing data. Since this column might not be used for the analysis, it was discarded from the dataset. The reordering columns had a proper data type classification.

### 1.2.2 - Customers table

Since the customer's table has the same kind of data type inconsistencies as the orders table, the same modifications were performed to standardise the data.

### 1.2.3 - Returns table

To improve performance and consistency, certain modifications were made to the returns table. Specifically, the "Returned" column, which previously consisted of "Yes" or "No" values, was replaced with a Boolean data type. Similarly, the Order ID in the returns table was changed to a string data type, following the same reasoning applied to the orders table. As for the Region column, instead of it being an object data type it was changed to a string data type.

### 1.2.4 - Sellers table

In the sellers table, the Person and Region columns were initially of the object data type. They were transformed to string data type to ensure consistency with the rest of the data.

### 1.2.5 - Holiday USA table

Like the previous tables, it was essential to modify the data types of certain columns in this table to ensure consistency with the rest of the data. The Date column was converted to a date data type to facilitate date-based computations. Additionally, the Holiday and WeekDay columns, which were initially of object data type, were converted to string type to ensure consistency across all columns.

To work with a subset of the data that is consistent with the orders dataset, only the rows corresponding to the years 2012 to 2015 were selected. Additionally, the Date column was in a different date format than the one used in the orders dataset. Therefore, the date format was converted from Y-M-D to D-M-Y to ensure consistency across all columns.

### 1.2.6 - USA GDP table

Several modifications were made to the GDP table to ensure consistency with the orders dataset and enable better data analysis. Firstly, the unnecessary columns, namely Region, SUB\_REGION, and County, were removed as they do not provide any useful information for the analysis.

Furthermore, it was observed that the orders dataset did not contain data for the states of Alaska and Hawaii, while the GDP table did. To orderstain consistency between the datasets, the data for Alaska and Hawaii was removed from the GDP table.

To gain a better understanding of the data associated with counties, the data was grouped by state. Two new columns were created to provide insights into the GDP of each state: Avg\_GDP column containing the average GDP of each state, and Total\_GDP column containing the sum of the GDP of all counties within each state.

Finally, the State column was converted to a string data type to ensure consistency with the rest of the data.

## 1.3 - Draw a diagram with links between data sources

Once all the necessary information was collected, the next step was to create a diagram that would effectively illustrate the relationships between the data. The figure below presents the resulting diagram, which displays all the relevant tables, their corresponding variables, and the connections between the tables that are established through these variables.

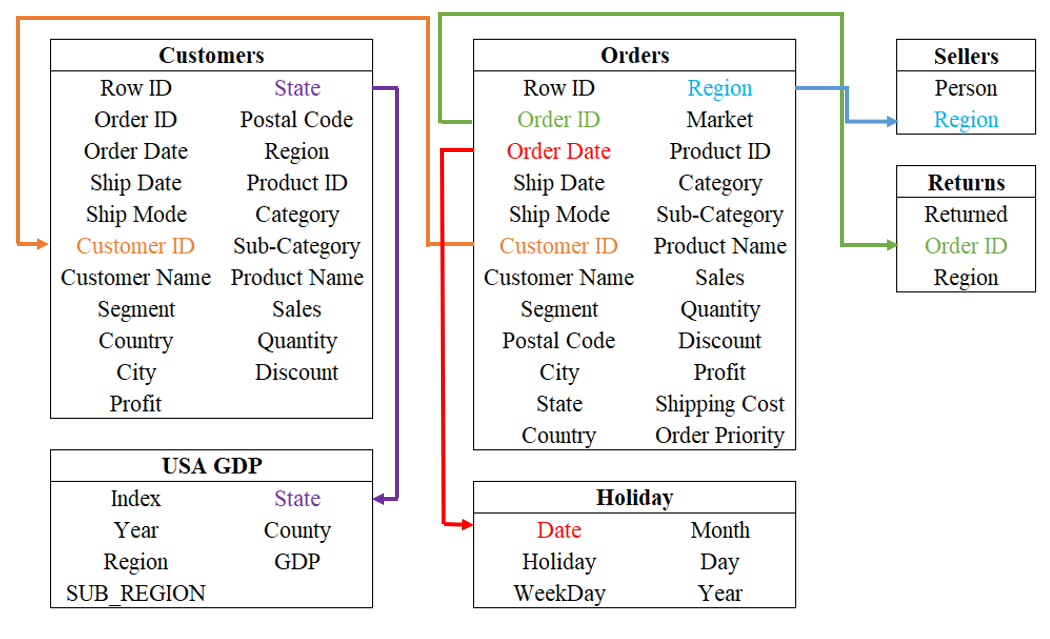


Figure 12 Diagram with all tables chosen for this project as well as their connections.

## 1.4 - Describe a business process to take advantage of data

The datasets presented simulate a typical sales process that is common to many businesses. They contain valuable information on product sales, customer and store demographics, orders and shipping, product returns, and promotional activity. By analysing this data, businesses can gain insights into their performance over time and across different locations and identify areas for improvement. As an example, an analyst could use the dataset to:

* Analyse sales trends over time and identify top-selling and most profitable products or categories. These values can also be examined across different locations to identify underperforming areas and determine which items to promote in each.
* Identify the customer segments that generate the most revenue, their geographical locations, preferred products, and how promotions influence their purchasing behaviour. This information can be used to adjust marketing strategies and optimise promotions to increase revenue and customer satisfaction.
* Evaluate shipping performance across different stores and identify areas for improvement in delivery times and shipping costs. Analyse if these factors are linked to low sales or high refund counts for other products.
* Investigate the root causes of poor delivery times and high shipping costs, such as the distance from the store to the customer, the type of product, order priority, or delivery mode. This information can be used to improve delivery times and lower shipping costs, thereby enhancing customer trust and satisfaction.
* Utilising the orders information, we can identify which products are often purchased together and determine how marketing strategies on certain products affect the sales of others. This information can be used to optimise inventory and strategize which promotions would be most effective, while also identifying products that are not influenced by promotions and adjusting marketing accordingly. Additionally, we can determine which products have seasonal sales patterns and target them during the most profitable timeframes.
* We can also analyse trends in customer loyalty across different locations by examining repeat purchases and identifying what top-performing sellers do differently. This information can be used to improve customer trust and satisfaction and inform strategies for improving sales in underperforming locations.

Given the project's scope, this analysis will focus on one critical business process: product profitability.

Overall, these datasets can be used to gain insights into the sales process of a business and identify opportunities for optimisation and growth.

## 1.5 - Define three analytical questions for the business process

According to these datasets, it would be beneficial to understand which products, regions, categories, and customer segments this chain should target or avoid as well as the best and worst performing sellers. For that, three analytical questions aligned with the orders of three components that make up our business process (product profit) were created.

### 1.5.1 - Sellers Analysis

There are various questions relative to the seller that can be asked which are aligned with our business process (product profit).

Which seller generates the highest / lowest profit? This can be considered the main question, from which we can start decomposing. Who is the seller with the highest or lowest profit? Other variables can be analysed to help answer this question, for example: Do the customers of the best seller belong to a state/city/region with high GDP values? The inverse can be asked for the worst seller as well. What is the most frequent customer segment and product category of the highest and lowest seller? Do discounts influence the seller's profit? All these questions can give various insights into the influences or at least what correlates with the highest and lowest sellers.

### 1.5.2 - Customer analysis

Which customers are most valuable and profitable? Which ones keep returning to the same store(seller)? Do customers increase confidence in the stores with time? For example, do customers increase the frequency and or amount spent on a particular store(seller)?

What is the percentage of customers, out of all customers of the superstore, that returned a product? What was the shipment mode that customers preferred the least for every region the superstore operates in? Are most of the orders of products of the "Furniture" category made by customers with the consumer or corporate segment? Does the customer importance relate to how often they ask for critical priority orders? And being more specific: How many customers that ordered products from the Europe market conducted critical priority orders in the year 2015?

These questions offer decision-makers greater insight into customer behaviour which can be critical to improving the business process of the superstore. Information such as frequency and distribution of the orders and returns, the number of individuals who were always satisfied with their purchases and the individuals who had issues with what they ordered on multiple occasions. Putting these instances under a microscope and studying the reason for them can be very benefitting for the superstore.

### 1.5.3 - Product analysis

What was the season of the year when orders for "GlobeWeis Clasp Envelopes" were made in higher amounts? Does this or any other product orders relate to the occurrence of holidays? What were the top 3 categories and respective sub-categories of products that generated the most profit for the superstore? For these three categories, what was the sub-category that had products with the highest discounts? Was there any crossover between the sub-categories for this answer and the sub-categories that generated the highest profit asked before? What is the average price of shipping that made customers not buy the product?

These questions, more centred around the products being sold, allow the decision makers to see clearly at what moments in a year a product is in higher demand and if they would benefit more from increased discount percentages at those times. What categories and sub-categories are the most profitable for the company is also indispensable information for the people tasked with deciding prices and quantities of future resupply orders. To be able to single out less profitable products is of worth to the superstore as studies can be made as to why these products are not performing as well as the rest and if needed discontinuation from selling them all together can now be supported from this analysis made available from being able to answer questions such as the ones asked here.

## 1.6 – Conclusion of the First Stage

In conclusion, the first stage of the project focused on identifying the data sources used and analysing the values and errors in each data field of the selected datasets. Four main datasets were analysed: Global Superstore Orders 2016, Global Superstore Returns 2016, Superstore Dataset (customers\_USA), and US Holiday Dates (2004-2021), along with the Highest GDP Counties in the USA dataset.

The analysis of each dataset involved examining the data types, identifying inconsistencies or errors, and standardizing the data to ensure accuracy and consistency. Python programming language and various libraries such as pandas, NumPy, seaborn, and matplotlib were utilized for data analysis and processing.

The Orders dataset contained information about sales, while the Returns dataset provided details on product returns. The Superstore Dataset focused on sales in the United States, and the US Holiday Dates dataset provided information about holidays. The Highest GDP Counties dataset offered data on the GDP of counties in the USA.

During the analysis, inconsistencies in data types were identified, such as object data types being used for features that could benefit from different data types. Modifications were made to standardize the data, including converting object data types to appropriate data types such as dates, strings, Booleans, and categories. Missing data and unnecessary columns were also addressed.

A diagram was created to visualize the relationships between the data sources and their variables, highlighting the connections established through these variables.

Furthermore, a business process was described to take advantage of the data, focusing on product profitability. Analytical questions were defined for the business process, covering areas such as seller analysis, customer analysis, and product analysis. These questions aimed to identify the highest/lowest profit-generating sellers, valuable and returning customers, and profitable product categories.

# 2 - Second Stage

The second stage of the data analysis process is dedicated to dimensional modelling, a crucial step in creating a specialized multidimensional model tailored for a data warehouse. This stage holds significant importance as it lays the groundwork for subsequent phases, such as ETL system development and analytical reporting.

Firstly, we declare the granularity and type of the fact table(s). The fact table represents the core of the dimensional model and contains the numerical measurements that we aim to analyse. Determining the appropriate granularity and type ensures that the fact table aligns with the specific needs of the business process.

Next, we focus on modelling business dimensions. These dimensions provide context and perspectives to the data stored in the fact table. They include hierarchies that define the relationships and levels of detail within the data. By accurately modelling the business dimensions, we enable users to explore the data from different angles and comprehend the intricate relationships between various data elements.

Additionally, we identify the numerical measurements within the fact table(s). These measurements represent the key metrics that are of interest for analysis. By pinpointing these measurements, we ensure that the dimensional model encompasses the necessary data to support insightful analysis and decision-making.

Furthermore, we construct a data warehouse star diagram. This diagram visually represents the relationships between the fact table(s) and the associated dimensions. The star schema design, characterized by a central fact table connected to multiple dimension tables, facilitates efficient data retrieval, analysis, and reporting.

By prioritizing dimensional modelling in this stage, we establish a solid foundation for the subsequent phases of the project. The dimensional model allows users to explore data from various perspectives, comprehend relationships between data elements, and derive valuable insights from the data stored within the data warehouse.

With an effective dimensional model in place, the data warehouse is designed to support efficient data retrieval, analysis, and decision-making. It empowers users to navigate and explore the data with ease, unlocking the full potential of the stored information and facilitating informed business decisions.

## 2.1 – Fix problems detected in the previous stage

In this stage, we focused on addressing irregularities and standardizing the formatting of the data that was collected from each source in the previous stage. Our goal was to bring different representations of data with the same meaning to a unified standard.

To achieve this, we implemented various transformations. For example, we standardized the representation of dates for orders and holidays, ensuring that they followed a consistent format. We also modified columns referencing an ID to hold values in a string format. Similarly, we converted other columns, such as City and Product Name, to strings when it made sense to do so.

Columns with a high number of missing values that did not significantly contribute to our business process were promptly discarded. This allowed us to streamline the dataset and focus on the most relevant information.

Initially, we had planned to replace Boolean values in string format (e.g., "Yes" or "No") with actual Boolean True/False values. However, we decided to retain the string format while making it more descriptive of the character of the data. For example, instead of using True/False, we changed the values of a column named "Returned" to "Returned" and "Not Returned," providing clearer context for subsequent analyses in the next stage.

To facilitate consistency, we standardized column names and string values by removing unnecessary accents that could potentially cause file encoding format issues. We achieved this by using the "no accent" versions of the original strings. For instance, column names like "Valentine's Day" or "New Year's Eve" were modified to use simplified versions without accents.

In the orders table, there is a column named "Discount". For analysis purposes, we decided not to include it when defining the grain or conducting the analysis. This is because the "Discount" variable represents a percentage value that varies based on the product, quantity purchased, and other factors. Including this column in the fact table would not make mathematical sense and could potentially distort the accuracy of the analysis.

All the data pre-processing tasks were performed using Python 3, ensuring efficient and effective data manipulation.

## 2.2 – Declare the grain and type of the fact table

To ensure the successful construction of the data warehouse, it is essential to define the specifics of the star schema and establish the contents of each table that will be utilized.

In a star schema representation, the data is divided into a fact table and several dimension tables. The fact table contains rows that store detailed information about each instance or fact of a specific type of business event, also known as the grain. The dimension tables are connected to the fact table through unique identifying values, serving as primary keys in the dimensional tables. Each dimension table holds specific information about a particular aspect of the business.

In the case of the e-commerce store sales, it is evident that the facts table will be of the transactional type. This type of facts table is designed to capture, and store detailed transactional data related to specific business processes or events. In this context, the facts table will record individual transactions for each customer purchase. It will include important details such as the customer, items purchased, quantities, prices, and other relevant transaction-specific information.

By utilizing a transactional facts table, the data warehouse can effectively capture and store the granular details of each transaction. This level of detail enables in-depth analysis and reporting, offering a comprehensive view of the sales process. It serves as the foundation for various types of analysis, including sales performance evaluation, customer behaviour understanding, and product trend analysis.

Furthermore, when developing the dimensional model, it is preferable to focus on the most atomic information captured by the business process. Atomic data refers to the most detailed information that cannot be further subdivided.

For this project, the grain definition in the chosen business process, which revolves around product profit, will be as follows: A row in the fact table represents a purchase of a specific quantity of a single product belonging to a particular category and subcategory. This purchase is associated with a specific order placed on a particular ordering date, prioritized as low, medium, or high. The purchase includes shipping, with a specific shipping cost and mode (e.g., first class, second class, standard class, or same day) and a specific shipping date. The purchase is made by a named customer belonging to either the corporate or consumer segment of the superstore. It originates from a specific market within a particular region, country, state, and city. The product is sold at a specific monetary value and incurs a specific profit unless it is part of an order from a specific region that is later returned to the superstore.

By defining the grain in this manner, we ensure that the facts table captures the necessary level of detail for analysis and reporting, enabling a comprehensive understanding of the business process and its associated profitability.

## 2.3 – Model business dimensions, including data hierarchies

In the world of business intelligence and data analytics, modelling business dimensions is a crucial step in developing an effective data architecture. Business dimensions are the various categories that describe an organization's operations, customers, products, and other key aspects. By modelling these dimensions, businesses can gain insights into their operations and make informed decisions based on data.

One key element of modelling business dimensions is the use of dimensional tables. These tables provide a standardized way to organize data related to a particular business dimension, such as products, customers, or time. Each dimension table typically includes a set of attributes that define the dimension, such as product name, customer ID, or date.

Modelling business dimensions involves creating data hierarchies, which illustrate the relationships between different levels of data within a given dimension. For instance, a product hierarchy may include product type, brand, and model categories. In terms of the data dimension, there is a many-to-one relationship between attributes (e.g., many months make up a year). Hierarchies can have either fixed or variable depth, with fixed-depth hierarchies containing values for all levels and variable-depth hierarchies potentially having unfilled levels. When designing tables with hierarchies, it's important to sort them in descending order of granularity (Year > Month > Day) to allow for analysis at various levels of granularity. Redundancy of data in dimension tables enables businesses to analyse data at different levels and obtain deeper insights into their operations, resulting in better-informed decisions.

In addition to modelling business dimensions and creating data hierarchies, another important aspect of designing a data architecture is optimizing the size and performance of fact tables. In some cases, it may be necessary to split a large fact table into multiple smaller ones to improve query efficiency and reduce storage requirements.

Based on the characteristics of the data used in this project, it can be categorized as multidimensional, which can be stored in three different ways: ROLAP, MOLAP, and HOLAP. For this project, a ROLAP system will be utilized, which typically involves a fact table as the main table. The fact table connects the business measures to the dimensions, each of which has its own table called a dimension table. The details of this system will be described below.

The next part of this report will have an analysis of the fact table and each of the dimensional tables used for the data under study. From these tables, it will be possible to see each of the features inside the fact table and each of the dimensional tables.

### 2.3.1 - Facts Table

Given the information about the grain referred on the previous point, the Facts Table will have six dimensions: Product Dimension; Customer Dimension; Order Information Dimension; Seller Dimension; Date Dimension (Role-Playing Dimension for Order Date and Shipment Date). The Facts Table will also contain the following additive measures: Sales; Quantity; Profit; Shipping Cost. Lastly is important to refer that a degenerate dimension was created (Transaction Key) which in certain form represents the grain, each purchase that was executed, which is aligned with the type of the Facts Table (Transaction) where the grain is one row per transaction. Degenerate dimensions are very common when the grain of a fact table represents a single transaction or transaction line item because the degenerate dimension represents the unique identifier of the parent (Kimball & Ross, 2011).

Table 16 Facts Table description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| Transaction Key (DD) | Unique Key - Degenerate Dimension | NUMERIC | 1 |
| Product Key (FK) | Foreign Key | NUMERIC | 1 |
| Customer Key (FK) | Foreign Key | NUMERIC | 1293 |
| Order Key (FK) | Foreign Key | NUMERIC | 1 |
| Order Date Key (FK) | Foreign Key | NUMERIC | 264 |
| Ship Date Key (FK) | Foreign Key | NUMERIC | 276 |
| Seller Key (FK) | Foreign Key | NUMERIC | 1 |
| Sales | Sales in $ of a certain purchase | NUMERIC | 82,674 |
| Quantity | Quantity bought of a certain product in a certain purchase | NUMERIC | 2 |
| Profit | Profit obtained in $ from a certain purchase | NUMERIC | -157,086 |
| Shipping Cost | Cost in $ of a certain shipment of a certain order | NUMERIC | 5,69 |

Given the above description, the following table contains the five first rows of the Facts Table.

Table 17 Five first rows of the Facts Table

A picture containing text, road, monitor, screen

Description automatically generated

### 2.3.2 - Product Dimension

The product dimension table contains valuable information about the products sold by the organization, including product names, categories, and sub-categories. In addition, the primary key for this dimension is the Product Key, providing a unique identifier for each product sold according to its Product ID. These attributes allow businesses to analyse product sales and trends across different categories, brands, and manufacturers.

The product name, category, and sub-category are the data hierarchies of this table. The data hierarchy goes in descending order, category, sub-category, and product name. The category attribute provides high-level information about the product, while the sub-category attribute narrows down to a more specific type of product. Finally, the product name attribute provides the individual name of each product. Using this data hierarchy, businesses can better understand their product sales and identify which categories, sub-categories, and individual products are performing well or poorly.

Table 18 Product Dimension description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| Product Key (PK) | Primary Key | NUMERIC | 1 |
| Product ID | Unique ID of the Product | NUMERIC | OFF-FA-6129 |
| Category | Category of a given purchased product | VARCHAR | Office Supplies |
| Subcategory | Sub-Category of a given purchased product | VARCHAR | Storage |
| Product Name | Name of the product | VARCHAR | Fellowes File Cart, Industrial |

To enrich the Product Dimension description the following table contains its first five first rows.

Table 19 Five first rows of the Product Dimension Table

Text

Description automatically generated

### 2.3.3 - Date Dimension (Role-Playing – Order & Ship)

The Date Dimension can be seen as a fundamental dimension of every data warehouse, enabling a longitudinal analysis of the business process, and therefore uncovering possible trends in the data. The Dimension Date can be seen as ubiquitous in data warehouses, being essential for historical data analysis.

A table with multiple valid relationships between itself and another table is known as a role-playing dimension. This is most seen in dimensions such as Time /Date. The Facts Table has 2 relationships to the Dimension Date on the Order Date Key and Ship Date Key. One possibility to operationalize this could be to have one physical table with all dates and obtain multiple logical tables using synonyms or SQL views given that views allow attribute names to be role specific.

Another important aspect to refer to in this Dimension is the presence of hierarchies which play an important role in future navigation in the data cube (e.g., drill-down and roll-up operations) and to pre-calculate aggregate values for each hierarchical level. Starting from the broader attribute, the Year, followed by the Season, Semester, Month, Week of the Month, Day of the Month and Day of the Week. This can be considered a hierarchy of fixed depth given all levels of the hierarchy always have values.

Lastly, to avoid further increase of this dimension size, a Foreign Key with the name of Holiday Key has been created as well as another attribute called Holiday Indicator which informs if the Date of Order or Shipment was a holiday or not. The Holiday Key will be connected to the Dimension Holiday which will be an Outrigger.

Table 20 Date Dimension description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| DateKey (PK) | Primary Key | NUMERIC | 1 |
| Full Date Description | Full Date of a certain Order (year/month/day) | TIMESTAMP | 01/01/2012 |
| Year | Year of a certain Order/shipment | NUMERIC | 2012 |
| Season | Season of the Year of a certain Order/shipment | VARCHAR | Winter |
| Semester | Semester of a certain Order/shipment | NUMERIC | 1 |
| Month Number Year | Number of Months of a certain Order/shipment | NUMERIC | 1 |
| Week Number Year | Number of Week in Year of o certain Order/shipment | NUMERIC | 1 |
| Day Number Month | Number of Day in Month of a certain Order/shipment | NUMERIC | 1 |
| Day Number Week | Number of Day in Week of a certain Order/shipment | NUMERIC | 5 |
| Day Name Week | Name of Day of the Week of a certain Order/shipment | VARCHAR | Sunday |
| Holiday Key (FK) | Foreign Key | NUMERIC | 12 |
| Holiday Indicator | Indication of if it was a Holiday or not | VARCHAR | Holiday |
| Weekend Indicator | Indication of if it was a Weekend or not | VARCHAR | Weekend |

To better illustrate how the role-playing will be expressed, below can be seen the five first rows for the Date Dimension taking the form of the Order Dates first, and second, taking the form of the Shipment Dates.

Table 21 Five first rows of the Date Dimension Table for Order Date

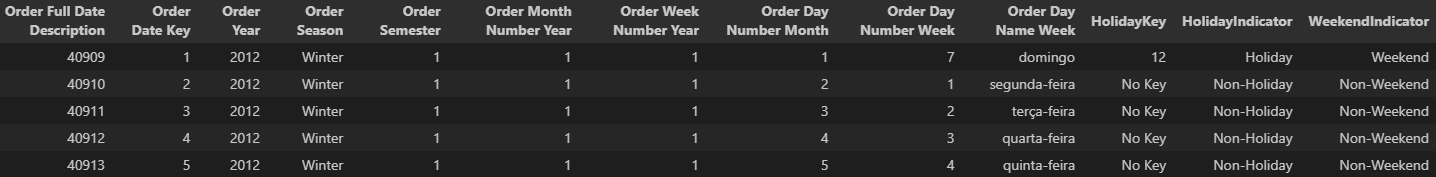


Table 22 Five first rows of the Date Dimension Table for Shipment Date

Graphical user interface, application

Description automatically generated

### 2.3.4 - Holiday Dimension (Outrigger - Connected to Date Dimension)

The Holiday Dimension captures the dates of United States of America holidays as attributes that change infrequently. An outrigger was created by snowflaking the "monster" Date Dimension, and the connection between the Holiday Dimension and the Date Dimension is established through the Holiday Key.

In addition to the full date of the holiday, this dimension includes the holiday name and a fixed type of hierarchy comprising the Year, Month, Day of the Month, and Day of the Week of the holiday. This table is expected to facilitate the identification of consumer behaviour patterns related to holidays if they exist. The date range covers the years 2012 to 2015.

Table 23 Holiday Dimension description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| Holiday Key (PK) | Primary Key | NUMERIC | 1 |
| Full Holiday Date | Full Date of the Holiday (year/month/day) | TIMESTAMP | 04/07/2004 |
| Holiday Name | Name of Holiday | VARCHAR | 4th of July |
| Year Holiday | Year of Holiday | NUMERIC | 2012 |
| Month Holiday | Month of Holiday | NUMERIC | 7 |
| Day Month Holiday | Day of the Month of Holiday | NUMERIC | 4 |
| Day Week Holiday | Day of the Week of Holiday | VARCHAR | Wednesday |

Given the above description, the following table contains the five first rows of the Holiday Dimension Table.

Table 24 Five first rows of the Holiday Dimension Table

A picture containing text, road, screenshot

Description automatically generated

### 2.3.5 - Seller Dimension

The Seller Dimension provides information about the sellers in the present e-commerce chain. In addition to the seller's name, this dimension includes location attributes that enable the creation of a hierarchy, such as Seller Market, Seller Region, Seller Country, Seller State, and Seller City.

The Seller Key serves as a unique identifier for a specific seller in a particular city. Therefore, it is possible to have multiple keys for the same seller but with different cities. This attribute highlights an important aspect of the Seller Dimension, namely that it is a Slowly Changing Dimension (SCD). Specifically, the Seller Dimension can be considered a Type 2 SCD since it creates a new line of data whenever there is an update to the localization of the seller. By including this context, we gain a better understanding of how the data in the Seller Dimension is managed and how the Seller Key attribute relates to other attributes in the dimension.

Table 25 Dimension description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| Seller Key (PK) | Primary Key | NUMERIC | 1 |
| Seller Name | Name of the Seller | VARCHAR | Kaoru Xun |
| Seller Market | Market of the Seller | VARCHAR | Asia Pacific |
| Seller Region | Region of the Seller | VARCHAR | Western Asia |
| Seller Country | Country of the Seller | VARCHAR | United Arab Emirates |
| Seller State | State of the Seller | VARCHAR | Ajman |
| Seller City | City of the Seller | VARCHAR | Ajman |

To enrich the Product Dimension description the following table contains its first five first rows.

Table 26 Five first rows of the Seller Dimension Table

Graphical user interface, application

Description automatically generated

### 2.3.6 - Customer Dimension

The customer dimension table provides valuable information about the organization's customers, including their names, segments, states, regions, and city postal codes. The primary key for this dimension is the Customer Key, which serves as a unique identifier for each customer in a specific postal code. This means that one customer can have multiple postal codes associated with different places where orders occurred. This attribute highlights that the customer dimension table is a Slowly Changing Dimension (SCD) that follows the Type 2 methodology, where changes in customer localization result in the creation of a new row of data. By analysing these attributes, businesses can gain insights into customer behaviour and preferences, identify opportunities for targeted marketing, and develop customer retention strategies.

For the customer dimension table, the hierarchies are the ones with the segment and customer name. The segment attribute allows businesses to group customers based on their demographic or behavioural characteristics, while the customer’s name attribute provides a unique identifier for each customer. By analysing these hierarchies, businesses can gain insights into customer behaviour and preferences, identify profitable customer segments, and target their marketing efforts more effectively.

Another important hierarchy for the customer dimension table is the State, Region, City and Postal Code. This hierarchy provides businesses with valuable insights into the geographic distribution of their customer base. By analysing this hierarchy, businesses can identify trends and patterns in customer behaviour across different regions, target their marketing efforts more effectively, and optimize their supply chain and logistics operations. Moreover, businesses can identify areas with high customer concentration, prioritize them for expansion, and allocate resources accordingly.

Table 27 Customer Dimension description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| Customer Key (PK) | Primary Key | NUMERIC | 1 |
| Customer ID | Unique ID to identify each customer. | VARCHAR | AA-10315 |
| Customer Name | Name of the customer | VARCHAR | Alex Avila |
| Customer Segment | Segment where the customer belongs | VARCHAR | Consumer |
| Customer State | State of residence of the customer | VARCHAR | Minnesota |
| State Key (FK) | Primary Key | NUMERIC | 22 |
| Customer Region | Region of residence of the customer | VARCHAR | Central |
| Customer City | City of residence of the customer | VARCHAR | Minneapolis |
| Customer Postal Code | Postal Code of the customer | VARCHAR | 55407 |

Given the above description, the following table contains the five first rows of the Customers Dimension Table.

Table 28 Five first rows of the Cutomer Dimension Table

A black screen with white text

Description automatically generated with low confidence

### 2.3.7 - GDP Dimension (Outrigger - Connected to Customer Dimension)

In the data architecture for this business intelligence system, the customer dimension plays a critical role in understanding customer behaviour and preferences. To further enhance the usefulness of this dimension, an outrigger dimension was created that considers the Gross Domestic Product (GDP) of the state where the customer lives.

The decision to create this outrigger was based on the fact that whenever the analysis is performed on GDP, it is likely to be associated with the customer dimension. By incorporating GDP data into the customer dimension, businesses can gain deeper insights into customer behaviour and preferences concerning economic factors, such as spending power and purchasing habits. The Outrigger avoids repeating the GDP data for customers in the same State.

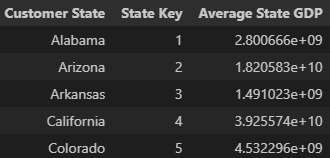
The primary key for this sub-dimension table is the State Key, which provides a unique identifier for each state where the organization's customers reside. Key attributes in this table include the state name and the average GDP of the state. The Average State GDP can be seen as a Semi-additive aggregate function that contains the average of the GDP values in US $ per state according to each state's county GDP value, between the years 2012 and 2015. This measure was used instead of the county itself to be in conformity with the rest of the other facts and dimensions tables.

Table 29 GDP Dimension description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| State Key (FK) | Foreign Key | NUMERIC | 1 |
| Customer State | The state of reference for the GDP | VARCHAR | Alabama |
| Average State GDP | AVG GDP value in $ per State between 2012 and 2015 | NUMERIC | 2800665869,403 |

To enrich the GDP Dimension description the following table contains its first five first rows.

Table 30 Five first rows of the GDP Dimension Table



### 2.3.8 - Order Information Dimension

The Order Information Dimension was designed to provide a centralized location for information about orders, given their high availability in our data sources. This dimension includes key attributes such as the Returned Indicator, which indicates whether an order was returned to the seller, as well as Ship Mode, Order Priority, and Order ID. The Order Key serves as the primary identifier for each unique order based on its ID.

By creating this dimension, we aim to streamline the analysis of orders and extract valuable insights for the business process. For instance, we can determine if certain products are frequently returned, or if high-priority orders are associated with specific products.

Table 31 Order Information Dimension description with Column Name (Field) Column description (Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| Order Key (PK) | Primary Key | NUMERIC | 1 |
| Order ID | Unique ID to identify each order. | VARCHAR | AE-2012-PO8865138-41184 |
| Returned Indicator | Indication of if the Order was Returned or not | VARCHAR | Not Returned |
| Ship Mode | Shipment Mode of the Order | VARCHAR | Standard Class |
| Order Priority | Level of Priority of the Order | VARCHAR | Medium |

Given the above description, the following table contains the five first rows of the Order Information Dimension Table.

Table 32 Five first rows of the order Information Dimension Table

A black screen with white text

Description automatically generated with medium confidence

## 2.4 – Identify numerical measurements in the fact table(s)

Numerical measurements, such as product profit, can be considered as performance indicators for a business. These measurement data are ultimately stored in a single data mart to aid in the decision-making process of the company. In this context, a "fact" represents a business measure, and numeric and additive facts are particularly useful.

The additivity of facts is crucial, as the data warehouse application will rarely retrieve a single fact table row (Kimball & Ross, 2011). Therefore, all numerical measurements in the fact table should be additive and have no empty values, which is recommended to prevent overwhelming the fact table (Kimball & Ross, 2011).

There are 4 additive measures in the facts table, the Profit in dollars, the Sales in dollars, the Quantity of products sold and the Shipping Cost of each sale in dollars.

Table 33 Information about the distributions of the numerical measures in the fact table such as margin values, quantiles, mean and standard deviation.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Description** | **Min** | **25%** | **50%** | **75%** | **Max** | **AVG** | **SD** | **Hist** |
| Sales | Value of sale in dollars | 0,444 | 30,759 | 85,053 | 251,053 | 22.638,48 | 246,491 | 487,565 |  |
| Quantity | Quantity of the product bought | 1 | 200% | 300% | 500% | 14 | 3,477 | 2,279 |  |
| Profit | Profit in dollars | -6.599,978 | 0 | 9,24 | 36,81 | 8.399,976 | 28,611 | 174,341 |  |
| Shipping Cost | Cost of shipping item(s) | 1,002 | 2,61 | 7,79 | 24,45 | 933,57 | 26,479 | 57,251 |  |

To gain a deeper understanding of the behavior of numerical measurements, we created several plots. Figure 13, Figure 14, Figure 15 present Quarterly evolution of Sales of the products. illustrate the temporal evolution of Sales, Profit, and Quantity of items sold over the four-year analysis period.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 13 Quarterly evolution of Sales of the products.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 14 Quarterly evolution Profit of products.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 15 Quarterly evolution quantity of products sold.

Upon observing the figures above, it becomes evident that the three measures (Sales, Profit, and Quantity of items sold) are highly correlated. Selling more products leads to higher sales, which, in turn, leads to higher profits, and vice versa.

The gradual increase of all three measures over time indicates an improvement in business productivity. However, upon analyzing each year individually, it becomes apparent that there is a gradual increase in all three measures at the beginning and throughout the year, followed by a drastic decrease in the first quarter of the new year. This may be due to the end-of-year holidays, including Halloween, Thanksgiving, Christmas, and New Year's, which can drain consumers' finances and make them more cautious with their spending at the beginning of the year.

In addition, we also conducted an analysis of the same measures using a monthly granularity, as presented in Figure 16, Figure 17, Figure 18.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 16 Average Sales of products sold in each month of the year.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 17 Average profit of products sold in each month of the year.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 18 Average Quantity of products sold in each month of the year.

The figures above clearly demonstrate the strong correlation between the three measures. Notably, February and July appear to be the worst months for the business. February's lack of profitability may be attributed to the fact that it typically has two to three fewer days than other months, which can significantly impact sales. In contrast, the drastic decline in July may be due to the nature of the products sold (office supplies), as many people take summer vacations and do not require such items.

The slight variations in sequential monthly values can be attributed to differences in the number of days in each month. Overall, profitability increases steadily throughout the year, with the start of the school year in the last five months potentially contributing to higher demand for office supplies.

We also conducted an analysis of the same measures using a daily granularity by month instead of monthly, as presented in Figure 19, Figure 20, Figure 21.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 19 Average Sales of products sold on each day of the month.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 20 Average Profit of products sold on each day of the month.

Uma imagem com gráfico

Descrição gerada automaticamente

Figure 21 Average Quantity of products sold on each day of the month.

As is possible to see in the above figures, Sales, Profit and Quantity of items sold vary day to day but not a whole lot, so it’s hard to see if any behavior is characteristic of a certain day of the month.

Normally the 9th and 23rd day of the month has the lowest business, at this point, we can’t hypothesize why. Seems like the most frequent last days of the month (28th,29th,30th) represent a steady decline in profitability, perhaps because the consumers have less money to spend since most people receive their paychecks at the beginning of the month. But that should probably influence the Sales on the 1st day of the month, and it doesn’t seem to be the case.

So is possible to propose the hypothesis that perhaps consumer behavior doesn’t change that much depending on the day of the month, but if it does, it must happen in specific months since we can’t detect a pattern with the average values.

We opted not to include any plots depicting the Shipping Cost of the Sale measure because it is a complex metric that depends on several variables, such as the product and the quantity sold, the distance between the vendor and the customer, and time itself. Moreover, the shipping fees charged in 2012 may differ from those applied in 2015, further complicating the analysis. Thus, it was challenging to establish any conclusive relationship between these variables and the Shipping Cost in a 2D plot.

While understanding how Shipping Cost affects business profitability is essential, we focused on showcasing simple examples of how additive measures vary over time to identify evident patterns initially. Further analysis can be conducted to explore the relationship between Shipping Cost and other metrics.

## 2.5 – Draw the data warehouse star diagram

The most popular data structure (or data modelling technique) used in data warehouses (DWs) is the star schema (Song et al., 2007), because of its logical construction of table structures, specifically to facilitate the execution of high-volume and intricate queries commonly referred to as online analytical processing (OLAP) (Krippendorf & Song, 1997).

In the star schema, the data is organized into a central fact table and surrounding it dimension tables (Sidi et al., 2016), creating a structure that resembles a star shape. The central facts table typically not only has quantitative measures of the data (for example sales and/or revenue) but also stores the foreign keys which will connect the dimension tables to the centre facts table (Weininger, 2002). The dimension tables contain the key which is used to connect to the central fact table and contain descriptive data that provide context to the measures in the fact table (for example periods, geography, and product categories) (Rowen et al., 2001). These attributes are the ones used for a more in-depth or more generalized view of the data, using methods like slice and dice, drill-down and roll-up.

An advantage that the star schema has is that it enables fast and efficient querying and analysis of large datasets (Sidi et al., 2016). It also provides a simple and intuitive way of organizing data, making it easier to understand and explain to other people. The star schema designed for the chosen dataset contains the following information:

1. **Facts Table:** This table contains measures such as Sales, Quantity, Discount, Profit, and Shipping Cost. It also includes foreign keys that establish connections with the dimension tables and the Transaction Key.
2. **Customer Dimension:** This dimension table incorporates attributes like Customer Key, Customer ID, Customer Name, Customer Segment, Customer State, Customer Region, Customer City, State Key, and Customer Postal Code.
3. **GDP Dimension** (Outrigger): This dimension table encompasses attributes such as State Key, Customer State, and Average Region GDP.
4. **Product Dimension:** This dimension table comprises attributes such as Product Key, Product ID, Product Name, Category, and Subcategory.
5. **Order Date Dimension:** This dimension table encompasses attributes such as Order Date Key, Order Full Date Description, Order Year, Order Season, Order Semester, Order Month Number Year, Order Week Number Year, Order Day Number Month, Order Day Number Week, Order Day Name Week, Holiday Key, and Weekend Indicator.
6. **Ship Date Dimension:** This dimension table incorporates attributes like Ship Date Key, Ship Full Date Description, Ship Year, Ship Season, Ship Semester, Ship Month Number Year, Ship Week Number Year, Ship Day Number Month, Ship Day Number Week, Ship Day Name Week, Holiday Key, and Weekend Indicator.
7. **Holiday Dimension** (Outrigger): This dimension table includes attributes such as Holiday Key, Full Holiday Date, Holiday Name, Year Holiday, Month Holiday, Day Month Holiday, and Day Week Holiday.
8. **Order Information Dimension:** This dimension table comprises attributes such as Order Key, Order ID, Returned Indicator, Ship Mode, and Order Priority.
9. **Seller Dimension:** This dimension table encompasses attributes like Seller Key, Seller Name, Seller Market, Seller Region, Seller Country, Seller State, and Seller City.

Using these dimension and fact tables, the following star schema was designed to represent the data effectively.

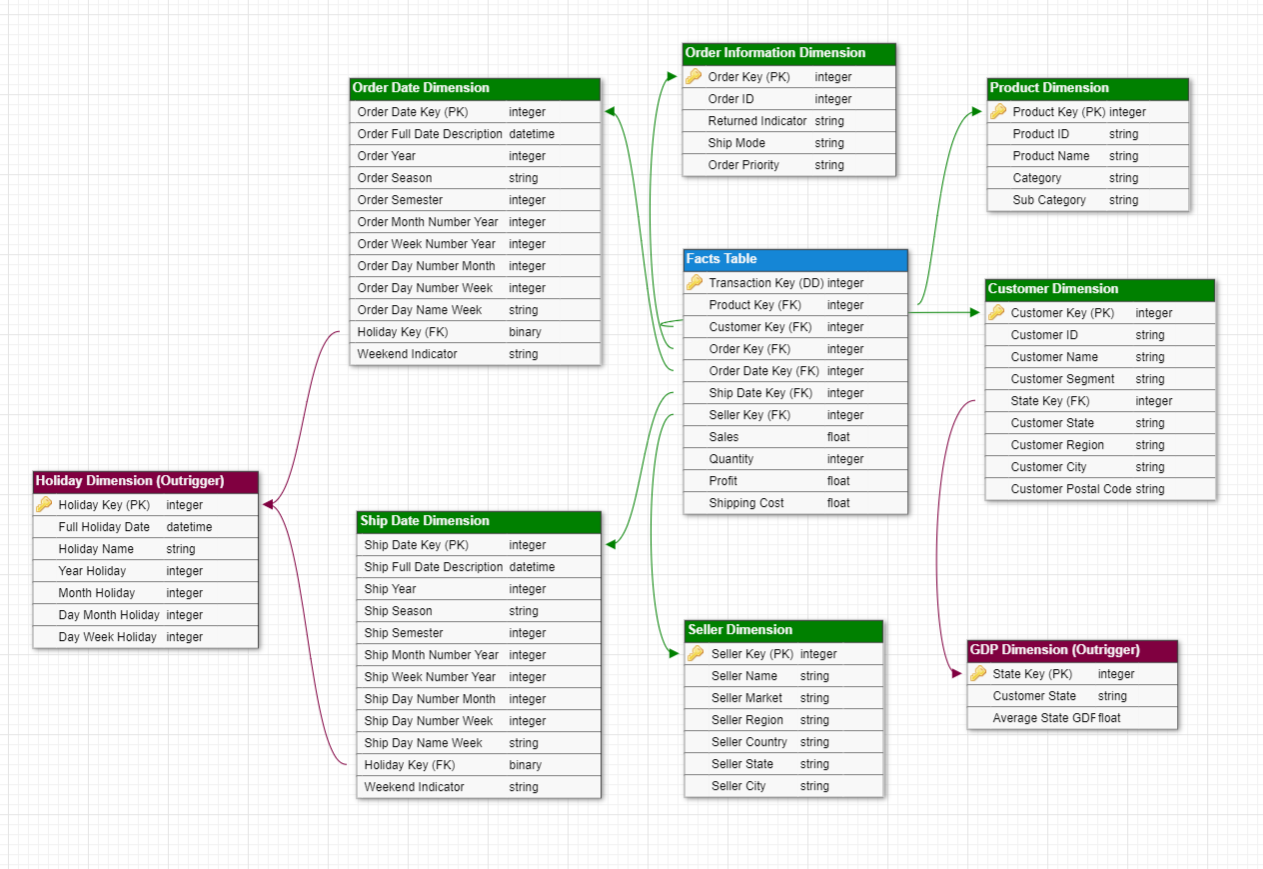


Figure 22 Star Schema containing the Facts Table (blue) at the Centre with the Dimension Tables (green) surrounding it with their associated Outriggers (purple)

## 2.6 - Conclusion of the Second Stage

In conclusion, the second stage of the data analysis process focuses on dimensional modelling, which is a crucial step in creating a specialized multidimensional model for a data warehouse. This stage involves determining the granularity and type of the fact table(s), modelling business dimensions, identifying numerical measurements, and constructing a data warehouse star diagram.

During this stage, efforts were made to address irregularities and standardize the formatting of the collected data from various sources. Transformations were implemented to bring different representations of data with the same meaning to a unified standard. Data pre-processing tasks included standardizing dates, modifying column types, discarding irrelevant columns, and standardizing column names and string values.

The grain and type of the fact table were declared to ensure the successful construction of the data warehouse. The fact table was identified as transactional, capturing detailed transactional data related to specific business processes. The grain of the fact table represented a purchase of a specific quantity of a single product belonging to a particular category and subcategory, associated with various details such as order information, customer information, seller information, and dates.

Modelling business dimensions played a crucial role in the second stage. Business dimensions, such as product, date, holiday, seller, and customer dimensions, were modelled to provide context and perspectives to the data stored in the fact table. Hierarchies within these dimensions were created to define the relationships and levels of detail within the data, enabling users to explore the data from different angles.

The analysis of the fact table and each dimensional table highlighted the attributes and hierarchies present in each table. The fact table contained additive measures such as sales, quantity, profit, and shipping cost, while the dimensional tables provided detailed information about products, dates, holidays, sellers, and customers.

By prioritizing dimensional modelling in the second stage, a solid foundation was established for subsequent phases of the project. The dimensional model allows users to explore data from various perspectives, comprehend relationships between data elements, and derive valuable insights from the data stored within the data warehouse. With an effective dimensional model in place, the data warehouse is designed to support efficient data retrieval, analysis, and decision-making, empowering users to navigate and explore the data with ease.

# 3 - Third Stage

The third stage of our project aims to optimize the staging and presentation areas, enhancing the visual and experiential aspects of our data warehouse. Our primary objective in this stage is to develop the Extract, Transform, Load (ETL) system and generate analytical reports to address the questions identified in stage one.

This stage is divided into two main branches: the staging area and the presentation area. In the staging area, our focus is on creating and developing the ETL system. This involves designing and implementing the processes to extract data from various sources, transforming it into a suitable format, and loading it into the database for analysis.

In the presentation area, we utilize the data that has been successfully loaded into the database. Leveraging the analytical power of tools like Power BI and SQL, we aim to answer the business questions that were initially proposed. By applying data visualization techniques, we will present the findings in a clear and informative manner, facilitating data-driven decision-making.

Throughout this stage, we will ensure a seamless integration between the staging and presentation areas, enabling a streamlined flow of data from extraction to analysis and reporting.

## 3.1 - Fix problems detected in the previous stage

During the previous stage of our analysis, we encountered several issues that demanded our attention and resolution. These concerns were identified through extensive discussions and in-depth analysis, and we took the necessary measures to effectively address them.

Initially, in the GDP Dimension, our approach involved calculating the average GDP from 2012 to 2015. However, after careful consideration and deliberation, we recognized that determining the average GDP for each individual year would yield more detailed and insightful analysis. This adjustment allows us to gain a comprehensive understanding of annual variations and trends in GDP, enabling more accurate decision-making.

We also came across a noteworthy attribute within the customer dimension table. It became apparent that a customer could have multiple postal codes associated with different places where orders occurred. This realization indicated that the customer dimension table is a Slowly Changing Dimension (SCD) that follows the Type 2 methodology. As a result, changes in customer localization led to the creation of new rows of data. To address this, we introduced a new column in the Customer Dimension, labelled "Status," which denotes the most recent postal code and identifies the older ones. To achieve this, we utilized the Order Date associated with the initial customer’s table. If the last order of a given customer exists, the "Status" column would be marked as "Updated," while otherwise, it would be considered "Outdated."

From these modifications it was necessary to apply changes to the star schema, which forms the basis of our data model. By reconfiguring the star schema to accommodate the revised GDP dimension and implementing other structural changes, we ensured the integrity and accuracy of our data representation. The resulting star schema will be presented during this stage when we talk about the data cube.

In addition, we rectified an omission from the previous stages of the report. It was noted that the conclusions of each stage were not included, so in this final report, we have provided comprehensive conclusions for each stage, encapsulating the key findings and outcomes.

Lastly, we realized that we neglected to mention the type of facts table utilized in our analysis. To rectify this oversight, we have included a detailed description of the facts table type in the preceding stage, offering a comprehensive overview of its characteristics and purpose.

By addressing these identified issues and implementing the necessary adjustments, we have enhanced the accuracy and depth of our analysis. These corrective measures contribute to a more comprehensive understanding of the data and enable us to derive more meaningful insights from our analysis.

## 3.2 – Staging Area

In the staging area, we will provide a detailed explanation of the step-by-step process involved in creating the ETL (Extract, Transform, Load) system. Throughout this section, we will explore the key components and techniques employed during the ETL system creation. Furthermore, we will highlight any considerations, challenges, or best practices encountered during the development of the ETL system.

The development and testing phase holds significant importance in the overall ETL (Extract, Transform, Load) system. During this phase, the programs that make up the ETL system are created and rigorously tested. Python3 serves as the programming language of choice for constructing the ETL system, while PostgreSQL acts as the underlying database, responsible for the storage and management of the extracted and transformed data.

Within the ETL system, each program/process has specific responsibilities, inputs, and outputs. Let us explore these aspects in detail for a comprehensive understanding of their roles within the larger framework of the ETL system.

1. Extraction Program/Process:
   * Responsibilities: The extraction program/process is responsible for retrieving data from various sources such as databases, APIs, files, or web scraping.
   * Inputs: It takes inputs in the form of connection details, query parameters, and data source specifications.
   * Outputs: The extraction process generates raw data in its original format, which is typically unprocessed and unstructured.
2. Transformation Program/Process:
   * Responsibilities: The transformation program/process is responsible for converting and reshaping the extracted data into a consistent and usable format.
   * Inputs: It takes the raw data obtained from the extraction phase as input.
   * Outputs: The transformation process produces transformed data that is cleansed, standardized, and enriched. This data is often organized into tables or data structures suitable for analysis and further processing.
3. Loading Program/Process:
   * Responsibilities: The loading program/process handles the task of storing the transformed data into the target database or data warehouse.
   * Inputs: It takes the transformed data as input.
   * Outputs: The loading process populates the target database or data warehouse with the transformed data, ensuring proper indexing and schema management.

By understanding the unique responsibilities, inputs, and outputs of each program/process within the ETL system, we can effectively orchestrate the extraction, transformation, and loading of data, guaranteeing a robust and reliable foundation for further analysis and decision-making.

### 3.2.1 - Extraction

The extraction phase serves as a critical and foundational step in the ETL (Extract, Transform, Load) system, laying the groundwork for retrieving data from operational systems and preparing it for further processing. This phase involves the utilization of diverse tools and techniques to effectively extract the required data from the source systems.

In this particular project, instead of extracting data directly from operational systems, the data was obtained from the open-source Kaggle website. To streamline the extraction process, we leveraged the widely adopted data manipulation library, Pandas. Pandas played a pivotal role in seamlessly reading and transforming the extracted data into structured DataFrames. By employing Pandas, we ensured that the data was appropriately organized and primed for subsequent transformations, ready to be loaded into the designated target destination.

### 3.2.2 - Transformation

The transformation phase, which follows the data extraction from the operational systems, is a crucial stage within the ETL (Extract, Transform, Load) system. In this phase, the extracted data undergoes a comprehensive transformation process in the data staging area. This process involves not only error identification and rectification but also the necessary adjustments to enable the creation of dimensions and facts tables.

To accomplish these transformations, Pandas proved to be an invaluable tool. It was employed extensively during this phase, providing capabilities for data writing, DataFrame manipulation, merging, and joining. A total of 5 CSV files and 1 XLSX file, extracted from the operational systems, served as the foundation for the required manipulations.

The transformation process encompasses a range of operations applied to the operational system tables. These operations include:

1. Data cleaning and validation: This involves identifying and correcting any errors or inconsistencies within the extracted data. It ensures data quality and reliability.
2. Data enrichment: Additional attributes or calculated values are incorporated into the dataset to enhance its analytical value and provide a broader context for analysis.
3. Data aggregation and summarization: Data is aggregated and summarized to create dimensions and facts tables, facilitating more efficient analysis and reporting.
4. Data merging and joining: Multiple datasets are combined based on common fields to create a unified and comprehensive dataset, enabling cross-referencing and analysis across different sources.

By executing these transformation operations, the data undergoes significant refinement, enabling it to be effectively utilized in subsequent stages of the ETL process. The resulting transformed dataset provides a structured and consolidated view, incorporating the necessary adjustments and enhancements required for insightful analysis and data-driven decision-making.

In the following discussion, we will explore the initial state of the different tables and present a comprehensive, step-by-step process of transforming them into their corresponding dimensions and facts tables.

#### 3.2.2.1 - Initial Datasets

##### Table Orders

The transformations applied to the table "Orders" primarily involved changes to the data types. Nominal variables such as Customer name and City were converted to strings. Variables containing values, which were all whole numbers in this case, were converted to integers. Categorical variables were transformed into the category data type. Lastly, the dates were converted to the very same DateTime format. To prevent conformity issues from spawning later the City and State columns had their values checked for presence of characters that were not present in all tables – quote and double quote marks were removed (ex: 'Ajman -> Ajman).

##### Table Customers\_USA

The Customers\_USA dataset comprises customer information for those who have made purchases in our e-commerce shop. It is worth noting that most of the column data types are currently defined as "object" data types. However, to enhance the performance of SQL searches, it is advisable to convert these data types to more appropriate alternatives. Therefore, columns containing integers, strings, dates, and categories have been appropriately modified.

##### Table returns

The table “returns” was checked similarly to Orders for the values in each column being in the string format.

##### Table sellers

As it only contains names of the sellers and what regions they operate in, both Nominal variables this table had these values be checked for type string and the column name of “Person” be more indicative of the values they represent so it was changed to “Name”.

##### Table GDP\_USA

To facilitate a wide range of intriguing queries utilizing the GDP of the USA, we obtained the GDP\_USA table. This comprehensive table captures the GDP per county of the USA from 2001 to 2018. To maximize the potential of this dataset and integrate it effectively with the e-commerce dataset, several pre-processing steps are necessary. Initially, we transformed the data types from 'object' to their respective specific data types. For instance, the 'Year' column was converted to a date data type, while the remaining columns were transformed into category data types, except for the 'GDP' column which retains the GDP values.

##### Table Holiday\_USA

The table "Holiday\_USA" serves as an outrigger for the Date Dimensions, encompassing all holidays along with their corresponding keys. However, to ensure that the foreign key "Holiday\_Key" in the merged Holiday and Date Dimensions does not have any missing values, a new row needs to be added. This new row represents a No Holiday Date and has a corresponding key. Figure 23 belows provides a visual representation of this additional row.

A screen shot of a computer code

Description automatically generated with low confidence

Figure 23 New Row added to holiday\_USA table to represent the absence of Holiday

Furthermore, several other transformations were applied to this table. Firstly, only the years relevant to the subsequent analysis (2012 to 2015) were retained. Additionally, the Date column was converted to the datetime format. Lastly, any holiday names that contained writing errors or typos were corrected.

#### 3.2.2.2 - Construction of the dimensional model

##### Holiday Dimension

To begin with, a column was generated to capture the Holiday Date from the holiday\_USA table. This column serves as the basis for the foreign key that will establish a connection to the Date dimensions later. Additionally, columns were created to store the holiday name, year, month, day of the month, and day of the week (as strings).

Finally, the holiday key associated with the absence of a holiday was replaced with "999" to simplify its identification.

##### Shipment Date and Order Dimensions

The same procedure was applied to both the ship date dimension and the order date dimension, as the only difference between them was whether it represented the shipment or order date. The process began by creating a column containing the unique dates of orders or shipments present in the Orders table.

Subsequently, various columns were generated to capture information related to these unique dates. This included details such as the season, semester, month/week number in the year, number of days in the month/week, and the name of the day in the week. Additionally, two descriptive indicators were added to identify the presence or absence of weekends and holidays. To accomplish the holiday indicator, the date dimensions were joined with the holiday dimension based on the Dates column. This resulted in the incorporation of the holiday key, which now served as a foreign key in the date dimensions. If the holiday key corresponded to the presence of a holiday in the holiday dimension, the holiday indicator in the date dimensions would reflect this. Conversely, if no holiday were found, the absence of a holiday would be indicated.

Finally, the foreign key was created as the Ship/Order date key. This decision was made to address potential conflicts and avoid the creation of duplicate keys. By creating the foreign key after the join with the holiday dimension, any conflicts were resolved, ensuring a more accurate and streamlined process.

##### Product Dimension

The Product Dimension was initially created by extracting columns for Product ID, Name, Category, and Sub-category from the Orders table. Once all these columns were included in the dimension, duplicate entries were removed, resulting in only unique products based on their product ID. Subsequently, the primary key was established using the unique Product IDs.

##### Order Information Dimension

The Order Information Dimension was constructed to hold information related to the details relevant to a single unique order of items to the superstore. The associated columns which are all composed of Strings and have been type checked to ensure no inconsistencies later originated from different flat files namely Ship Mod, Order Priority and Order ID from the orders related file while the Returned Indicator was generated by using information from the returns file. The way the returns were stored was by only having records for each order that was returned, if an order ID that belonged to the collection of existing orders did not exist in the return file column for order IDs than no returns were placed for the items of that order.

A join of these two files was done by the order ID column values, since the resulting table would hold each account of an order there was no need to store multiple rows that signalled the same order, as such any duplicate rows with already existing Order IDs were discounted alongside all columns that did not take part in our specification of this dimension in the star schema created in the previous stage. This join done on order ID created “NaN” values for all rows that did not have participation in the returns table through their Order IDs, a final adaptation was done where values of “Yes” were changed to “Returned” and all “NaN’s”, which represented orders that were not returned were replaced by “Not Returned”. A reindexing was done to follow the order of columns of our choice in the previous stage, led by the generation of surrogate key values starting at 1 for each entry by checking on unique Order ID values.

##### Seller Dimension

The Seller Dimension regards all attributes that characterize the seller recorded, these are the seller’s name also what city, state, region, country, and market they operate in. Each seller is responsible for a single market associated with a certain region of the world where the superstore can satisfy orders made. The highest degree of specificity in a row here is the city each seller can work in, as such it is the defining column for generating the surrogate keys like it was done for the Order Information Dimension above. As in the raw flat files there are rows where cities can have quotes or double quotes that could affect in conformity with other fields all these tokens were removed from these geographical centred columns.

To include the name of a seller (in seller file) with the geographical information about a seller (in orders file) it was necessary to join both tables on the common factor of Region, there was an issue where in the orders file the specificity of region for Canada is lesser than it is in the sellers file where the mapping is done with having a representative for Eastern Canada and Western Canada sellers. This was uniformized to only use Canada losing this degree of granularity when creating this dimension regarding sellers that oversee sales in the region of Canada.

##### Customer Dimension

The Customer Dimension plays a vital role in data analysis, providing valuable insights into customer-related information. To begin, we extract specific columns, such as 'Customer ID,' 'Customer Name,' 'Segment,' 'State,' 'Region,' 'City,' and 'Postal Code,' from the customers\_USA dataset. This extracted data forms a new DataFrame called Customer\_Dimension. Subsequently, we create unique numerical identifiers, referred to as 'Customer Key,' for each customer by combining their 'Customer ID' and 'City' information. These identifiers enable us to distinguish and identify individual customers within the dataset effectively.

It is important to note that the Customer Dimension is associated with the GDP Dimension, which serves as an outrigger dimension. Consequently, we add a column to the Customer Dimension, consisting of the key associated with the GDP Dimension. To accomplish this, we perform a merging process based on the common 'State' column found in both dimensions. The resulting Customer Dimension includes the 'State\_Key\_GDP' column from the GDP\_Dimension, which we rename as 'State\_Key\_Customer.' To maintain data integrity, we remove any duplicate rows based on the 'Customer\_Key' column from the Customer\_Dimension.

##### GDP Dimension

To construct the GDP Dimension, several pre-processing steps were executed on the GDP\_USA dataset. Initially, the GDP information was grouped by state instead of using county-specific GDP values since the e-commerce data does not provide county-level information. To enable more insightful analysis, an average of GDP values per state was calculated. Subsequently, a filter was applied to include only the years from 2012 to 2015, which align with the years present in the orders table. As a next step, four new columns were created, each representing the average GDP for a specific year (please note that the values are displayed in billions of dollars). To focus on relevant aspects for analysis, we removed the columns "Region," "SUB\_REGION," and "County." Finally, we added a “State\_Key\_GDP” which consists of a unique identifier for each state.

##### Facts Table

The last step of dimensional modelling involves creating the facts table for the e-commerce store, which serves as a central repository for crucial information about transactions. In our case, the facts table will include various foreign keys connecting to different table dimensions, along with attributes corresponding to the e-commerce shop's profit. These attributes can be found in the orders table, encompassing sales, quantity, profit, and shipping cost. Adding the foreign keys from the dimensions to the facts table requires considering the relationship between each data entry, which involves several pre-processing steps:

1. Firstly, we merge the facts table with the Product Dimension based on the "Product ID" column present in both tables. We then eliminate unnecessary columns, retaining only the "Product Key" column.
2. To ensure data integrity and facilitate time-based analysis, we map the ship date and order date to their respective date keys. This mapping is accomplished by leveraging pre-existing date dimension tables, namely Ship\_Date\_Dimension and Order\_Date\_Dimension. Utilizing the 'map' function, we associate date descriptions with their corresponding date keys.
3. Next, we perform a merge with the Seller Dimension using the 'City' column from the order data and the 'Seller City' column from the Seller Dimension. This enables the association of sellers with their respective cities. The resulting merged DataFrame, 'facts,' removes unnecessary columns from the Seller Dimension while preserving the Seller Key.
4. Similarly, we conduct a mapping for both the Customer and Order Information Dimension. This step enhances the comprehensiveness of the facts table by mapping customer and order information to their respective keys. By utilizing pre-existing dimension tables, namely Customer\_Dimension and Order\_Information\_Dimension, the 'map' function associate's customer names with their corresponding keys and order IDs with their respective order keys. This ensures that the facts table includes keys for each dimension.
5. Finally, we optimize the structure and organization of the facts table by eliminating unnecessary columns.

By following these steps, we obtain the definitive version of the facts table, which can be added to the database, incorporating all the necessary information.

### 3.2.3 - Loading

The third phase of the ETL system focuses on efficiently loading the extracted and transformed data from the Data Warehouse into the data presentation area. To accomplish this, we utilized *Psycopg*, a powerful PostgreSQL database adapter for the Python programming language. This enabled us to establish a seamless connection with the database and effectively load the dimensions and fact table into their respective tables.

Careful consideration must be given when incorporating various tables into a database, especially in our scenario where multiple tables have dependencies. Specifically, the dimensions rely on the outrigger tables, and the facts table is dependent on the dimension tables. Consequently, a systematic approach is essential. We begin by creating the outrigger tables, followed by the dimension tables, and concluding with the facts table.

To effectively create tables and populate them with information in PostgreSQL using Python, it is recommended to follow the following steps, which form the fundamental structure:

* We first check the status of the connection (conn) to the PostgreSQL database. If the status indicates that a transaction is in progress (STATUS\_IN\_TRANSACTION), it performs a rollback to undo any changes made in that transaction. This ensures that the database is in a consistent state before proceeding with further operations.
* We then define an SQL statement that creates a table in the database. Here we can add the several columns of the table and specify the column types and constrains.
* Then we execute a DROP TABLE IF EXISTS statement to remove (if it already exists) the existing table from the database. This ensures a clean slate and avoids conflicts if the table already exists. Another aspect was that when rerunning the code since it will have dependencies the CASCADE statement was used in the drop so that it could also erase the dependencies of the table.
* We then execute SQL statement to create the given table and apply the changes using conn.commit() to make them permanent.

This was the process behind the creation of each of the tables then it is necessary to load the information to the database to do so we performed the following steps:

* We prepare the data to be loaded into the database. The data is converted into a list using .to\_numpy().tolist(). This step transforms the data into a format that can be easily inserted into the database.
* We define an SQL statement that represents the insertion query to be executed. The query uses placeholders %s to indicate the values that will be provided later during execution.
* The code then executes the insertion query using cursor.executemany(). The executemany() method is used to insert multiple rows of data into the database in a single operation.
* Finally, the changes are committed to the database using conn.commit(). This step ensures that the inserted data becomes permanent and is saved in the database.

During the creation of the facts table, an intriguing occurrence unfolded. While loading the data into the database, consistent errors arose, prompting an investigation into their root causes. Two primary factors were identified as responsible for these errors.

Firstly, discrepancies were discovered between the orders and customer\_USA tables. Upon merging these tables, it became apparent that certain values were missing and required correction. Upon careful analysis, it was observed that the customer\_USA table lacked three customer records. Consequently, we excluded the corresponding customer records belonging to 'Tom Zandusky,' 'Cari MacIntyre,' and 'Kai Rey' from the Customer\_Dimension. Given that these names do not appear in the orders table and only three customers are affected, it is a straightforward decision to eliminate the associated rows.

Furthermore, an inconsistency was found between the seller and orders tables. The seller table only provided information about the seller's operating region, which was associated with the orders table. However, several missing values were discovered in the final table due to a more detailed description of Canada as "Western Canada" and "Eastern Canada" in the seller table, while the orders table contained a generic value of "Canada." To address this issue, the values in the seller table were modified to align with the more generic description in the orders table, ensuring a perfect match between the two.

### 3.2.4 - Draw a diagram with data flows and ETL system programs

In this stage, we present a comprehensive diagram that visually depicts the data flows and ETL system programs involved in the extraction, transformation, and loading of data.

The diagram highlights the flow of data from multiple source systems, represented by arrows, into the ETL system. These data sources are specifically associated with the datasets obtained through Kaggle.

Once the datasets are retrieved, they undergo a series of transformations to convert them into well-structured DataFrames. Each DataFrame goes through specific transformations tailored to its requirements.

Next, the transformed DataFrames are used to create dimensional tables and outrigger tables. These tables serve as the foundation for organizing and structuring the data in a way that aligns with the desired analytical framework.

The orders DataFrame plays a crucial role as it serves as the central point where foreign keys are added to establish connections with each of the dimensional tables forming the facts table.

The resulting tables, including the dimensional and outrigger tables, are loaded into a PostgresSQL database. This database serves as the repository for the transformed data, allowing for analysis and providing insights into the previously proposed business questions.

Finally, to answer the business questions that were proposed two main approaches were used one with SQL queries and the other with PowerBI. Both approaches complemented each other in providing comprehensive answers to the business questions. SQL queries allowed for detailed data exploration and analysis, while Power BI facilitated the creation of engaging visual representations of the findings.

The complete diagram, showcasing the data flows and ETL system programs, is provided below (Figure 24) for visual reference. This diagram offers a comprehensive overview of the entire ETL process, facilitating understanding and comprehension of the underlying components and their interactions.

A picture containing text, diagram, plan, technical drawing

Description automatically generated

Figure 24 Complete diagram, showcasing the data flows and ETL system programs

### 3.2.5 - Show dimensions and fact table of the implemented data cube

After completing the pre-processing steps, the resulting data was organized into dimensions and a fact table, which can be effectively implemented in a data cube. A data cube is a multidimensional data structure that allows for efficient and flexible analysis of data from different perspectives. We will now discuss the final dimensions and fact table that were generated as part of the data cube implementation.

As mentioned beforehand the dimensions represent the various aspects or categories along which data can be analysed. They provide context and granularity to the data within the data cube. In our case we identified the following dimensions:

* Customer Dimension: This dimension captures information related to customers, such as customer key, customer ID, customer name, customer segment, customer location, and other relevant attributes.
* Product Dimension: The product dimension contains details about the products offered, including product key, product ID, product name, category, and subcategory.
* Order Date Dimension: This dimension encompasses the temporal aspect of the data, capturing information about the order dates, including order date key, order full date description, order year, order season, order semester, order month number year, order week number year, and other time-related attributes.
* Ship Date Dimension: Like the order date dimension, the ship date dimension focuses on the shipping dates, including ship date key, ship full date description, ship year, ship season, ship semester, ship month number year, ship week number year, and other relevant attributes related to shipping time.
* Seller Dimension: The seller dimension pertains to the seller or vendors involved in the transaction. It includes attributes such as seller key, seller name, seller market, seller region, seller country, seller state, and seller city.
* Holiday Dimension (Outrigger): This dimension captures information related to holiday, including holiday key, full holiday date, holiday name, year holiday, month holiday, day month holiday and day week holiday.

These dimensions collectively provide a comprehensive framework to analyse the data based on different customer attributes, product categories, time-related factors, sellers, and holidays.

The fact table serves as the centrepiece of the data cube, containing the measurable and numerical data that forms the basis for analysis. It establishes the relationship between the dimensions and the numerical measures or facts. In our case, the fact table includes measures such as sales, quantity, profit, and shipping cost. It also includes foreign keys to establish connections with the dimension tables and the transaction key to uniquely identify each transaction.

The resulting star schema can be seen below in Figure 25.

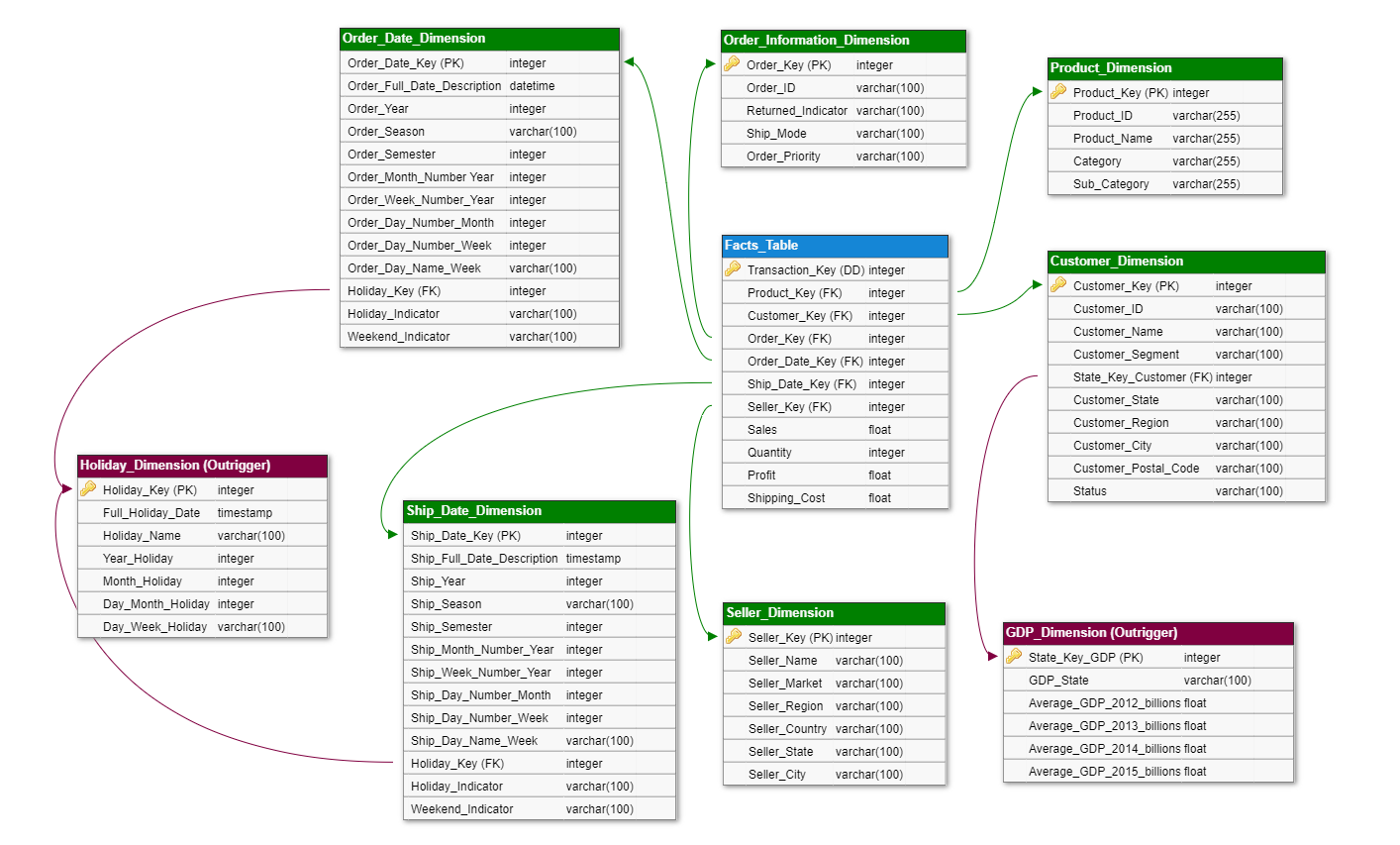


Figure 25 Star Schema

By incorporating the dimensions and the fact table into a data cube structure, we can perform multidimensional analysis, also known as OLAP (Online Analytical Processing). This allows us to analyse the data from various perspectives, drill down into specific dimensions, and derive meaningful insights by aggregating and slicing the data in different ways.

## 3.3 - Presentation Area

Now that the data has been successfully stored in the database, we can embark on a series of analyses that will provide valuable insights and answer the business questions at hand. To facilitate this process, we employed two distinct approaches for retrieving the information: utilizing SQL queries directly within the database and harnessing the capabilities of PowerBI.

By leveraging SQL queries, we can directly access and manipulate the data stored in the database. This method empowers us to extract specific information, perform aggregations, apply filters, and derive meaningful conclusions. SQL offers a robust and flexible toolset for conducting detailed analyses, providing a solid foundation for answering a wide range of business queries.

In addition to SQL, we leveraged the capabilities of PowerBI, a powerful business intelligence tool. With its intuitive visualizations and interactive features, PowerBI enables us to create insightful reports and dashboards. By connecting directly to the database, we can leverage PowerBI's rich set of analytical functions to explore the data from multiple angles, uncover patterns, and gain deeper insights into the business questions at hand. In Figure 1can be seen the star schema which is presented by PowerBI after the data is loaded.

The combined utilization of SQL queries and PowerBI allows us to approach the analysis process from different perspectives, enhancing our ability to derive meaningful insights. This comprehensive approach ensures that we have the necessary tools to address various business questions and make informed decisions based on the analysed data.

When integrating data into the PowerBI application, you have the option to verify the associations made during the creation of the PostgreSQL database. By examining the image below (Figure 26), we can confidently confirm that all the connections between the outrigger, dimensions, and fact tables in the PowerBI application have been accurately established. This alignment is consistent with the connections displayed in the data cube.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 26 Star Scheme showed by PowerBI

## 3.4 - Produce annotated reports that answer the analytical questions (pp.86)

Analytical questions were created in phase one of this project to understand which products, regions, categories, and customer segments this chain should target or avoid as well as the best and worst performing sellers. For that, three analytical questions aligned with business process (product profit) were created which focused on three main aspects: (1) Sellers; (2) Customers; (3) Products.

With the Data warehouse created these three main aspects were approached and the questions were answered using PowerBI which has strengths in up-to-date visualizations and/or queries (postgreSQL).

### 3.4.1 - Sellers Analysis

Regarding the analysis of the sellers four main questions can be made: (1) Which seller generates the highest / lowest profit?; (2) Do the customers of the best seller belong to a state with high GDP values?; (3) What is the most frequent customer segment of the highest and lowest seller?; (4) What are the most frequent product category and sub-category of the highest and lowest seller? Given that the discount was removed from our facts table, the question regarding it was removed.

#### 3.4.1.1 – Which seller generates the highest / lowest profit?

To distinguish easier the best and worst sellers, the top and bottom 5 sellers were selected regarding their sum of profit.

Starting with the SQL queries, they were straightforward involving a SUM of the profits GROUP BY the seller's names. For this one, JOIN was necessary between the facts table and the seller dimension. In the end, the results were ORDER BY the SUM of the profit in descending (DESC) order and LIMIT the output to 5 rows. The same logic was applied to obtain the bottom 5 sellers (worst sellers) with the only change being the ORDER BY ascending (ASC) instead of descending (DESC). The resulting outputs can be seen in Figure 27.

A screenshot of a computer program

Description automatically generated with low confidence

Figure 27 SQL outputs for question 1 of Seller's Analysis showing the Top 5 Sellers (left) and Bottom 5 Sellers (right) in terms of total profit.

With Power BI the procedures were simple, consisting of selecting the sellers and filtering them according to the 5 best or 5 worst in terms of total profit and from here creating the desired visualizations which can be seen below in Figure 28.

A picture containing text, screenshot, line, font

Description automatically generated

Figure 28 PowerBI Horizontal Bar Plots for 5 best sellers(left) and 5 worst sellers (right) in terms of total profit

These first questions show a piece of vital information to the stakeholders which is who are the best and the worst performing employees in terms of generated profit. From here further analysis can proceed to try and answer why these are the best / worst sellers.

#### 3.4.1.2 – Do the customers of the best seller belong to a state with high GDP values?

Given that the best and worst sellers were already detected, to verify if the customers of the best seller belong to states with high GDP two main steps have to be taken: (1) Obtain the count of customers of the best sellers per state and order them to see states with most customers from best sellers; (2) Obtain the states with high GDP values (AVG of 4 years) and see if they match with the states with more customers from the best sellers.

Starting with SQL, to obtain the first step more complex query had to be performed in comparison to the previous question. It consisted of 3 JOINS, involving the facts table, gdp dimension and seller dimension. SELECT the state, and COUNT of the DISTINCT customer\_key only where the seller’s name was equal to the best seller, which was determined on the previous question. After the results were GROUP BY state and ORDER BY the COUNT in descending (DESC) order. For the second step, a simple query was performed where the state and the average of the four years were selected and the results ORDER BY the average GDP in descending (DESC) descending, limiting the results to 5. The outputs of both steps can be seen below in Figure 29.

A screenshot of a computer program

Description automatically generated with low confidence

Figure 29 SQL outputs for question 2 of Seller's Analysis showing the Top 5 states with more customers from best sellers(left) and top 5 states in terms of AVG GDP (right).

When trying to answer this question using PowerBI some difficulties were encountered which were surpassed by creating an auxiliary table which contained the desired information which consisted of three columns regarding the states of the customers that belonged to the best sellers. This auxiliary table was created using three joins with facts table, customer, GDP, and seller dimension, which seemed in SQL language is easily performed but in PowerBI seemed overly complex and in the absence of a more efficient solution this was performed. After this auxiliary table was obtained it was straightforward to obtain the desired graphs which can be seen below in Figure 30.

A picture containing text, plot, line, font

Description automatically generated

Figure 30 PowerBI Vertical Bar Plots for 5 States with Higher AVG GDP(left) and 5 states with the highest number of customers from best sellers (right)

Even if not as straightforward to obtain, the PowerBI plots present the answer in a much more parsimonious way compared to the SQL queries. Is possible to see that there seems to be no relationship between the state average GDP and the presence of the best sellers customers, only the state of California is present in both plots which seems not to be sufficient to take this as a contributor to the seller's success given that the state District of Columbia excels all the other in terms of the AVG GDP and is not represented in the customers states.

#### 3.4.1.3 - What is the most frequent customer segment of the highest and lowest seller?

By trying to determine if there is a pattern regarding the customer segment of the best and worst sellers interesting insights can also be obtained.

To answer this question using SQL, a fairly simple query was performed that encompassed JOIN with facts table and, seller dimension to the customer dimension with a SELECT of the customer segment and a COUNT of the customer names WHERE the seller’s name were or the five best sellers or the five worst. Ending by GROUP BY customer segment. The results obtained from both queries can be seen below in Figure 31.

A picture containing text, screenshot, font, line

Description automatically generated

Figure 31 SQL outputs for question 3 of Seller's Analysis showing the customer segments count for the best seller (left) and worst sellers(right).

For PowerBI, again an auxiliary table was necessary, in this case two, one containing the best sellers and another one containing the worst, plus the other necessary columns such as customer segment and name. After the auxiliary tables were obtained the process was again straightforward. For both the best and worst sellers, two bar plots were created using the customer segment on the X-axis and its count on the Y-axis. The obtained plots can be seen in Figure 32.

A picture containing text, font, plot, line

Description automatically generated

Figure 32 PowerBI Vertical Bar Plots for the Customer Segment Count for the Best Sellers(left) and for the Worst Sellers(right).

Observing the outputs and the plots no visible insights can be taken given that for both the best and worst sellers the customer segment pattern is the same, the only difference in the number of customers, being that the best sellers have a much higher count.

#### 3.4.1.4 - What are the most frequent product categories and sub-categories of the highest and lowest seller?

Another interesting insight that can be taken regarding the analysis of the Sellers is related to the most sold product category and sub-category for the best and worst sellers.

Using SQL to answer this question, a query composed of two JOIN with the fact table and seller dimension to the product dimension, with a SELECT of category/sub-category and count of product names WHERE the seller names were the best or worst sellers. Using a GROUP BY category/sub-category and an ORDER BY the COUNT in descending (DESC) order. The obtained outputs for both categories and sub-categories regarding both best and worst sellers can be seen below in Figure 33.

A screenshot of a computer

Description automatically generated with low confidence

Figure 33 SQL outputs for question 4 of Seller's Analysis showing the Product Category (Top) and Sub-Category (Down) count for the best seller (left) and worst sellers(right).

Regarding the PowerBI usage to answer this question, the same method as in the previous 2 questions was used. In this case, two auxiliary tables (best and worst sellers) were created containing columns regarding the information of interest: category, sub-category and product name. Again, after obtaining these auxiliary tables the process was fairly simple and the resulting plots can be seen below in Figure 34 regarding the product category and in Figure 35 regarding the product sub-category.

A picture containing text, font, plot, line

Description automatically generated

Figure 34 PowerBI Vertical Bar Plots for the Product Category Count for the Best Sellers(left) and for the Worst Sellers(right).

When observing the outputs and plots regarding the product category, is visible that for both best and worst sellers the most frequent category is Office Supplies but there is a difference and that is that for the best sellers the Furniture category has a higher frequency than Technology, even if seemingly small. This visible difference could be tested using a statistical hypotheses test to evaluate if it is a significant difference.

A picture containing screenshot, line, diagram, plot

Description automatically generated

Figure 35 PowerBI Vertical Bar Plots for the Product Sub-Category Count for the Best Sellers(left) and for the Worst Sellers(right).

When observing the outputs and plots regarding the product sub-category, is visible that for both best and worst sellers the most frequent category is the Binders sub-category. There are some interesting differences, mainly if we focus on the top-selling sub-categories. For the best sellers, the Paper sub-category is in the top selling while in the worst sellers, it is in a low rank and the best seller seems not to sell as much art as the worst sellers. This may indicate that the Paper sub-category is a better bet than the Art. Again, some statistical hypotheses testing could be used to verify the significance of these observed differences.

### 3.4.2 - Customer Analysis

Regarding the questions proposed for the Customer analysis, they can be divided in three: (1) Which are the most valuable customers? Do they have hard-fixed habits and are they loyal to a certain store?; (2) Customer behaviour regarding segment and returns; (3) Customer preference of shipment mode and priority of orders.

#### 3.4.2.1 - Which customers are most valuable and profitable?

As we can see the most valuable and profitable Customers are also among the ones that buy in more quantity and the most expensive items. Some examples are “Tamara Chand”, “Raymond Buch” and “Sanjit Chand” which all gave a profit of over 7.5 thousand dollars to the Superstore.



Figure 36 PowerBI Horizontal Bar Plots for the top 10 most profitable Customers, Top 10 Customers that bought more products and made the highest sales.

#### 3.4.2.2 - Do customers keep returning to the same store? Do they increase confidence in the stores with time?

We analysed if the most profitable customers have the habit of always buying from the same sellers, and if that habit increased through time.

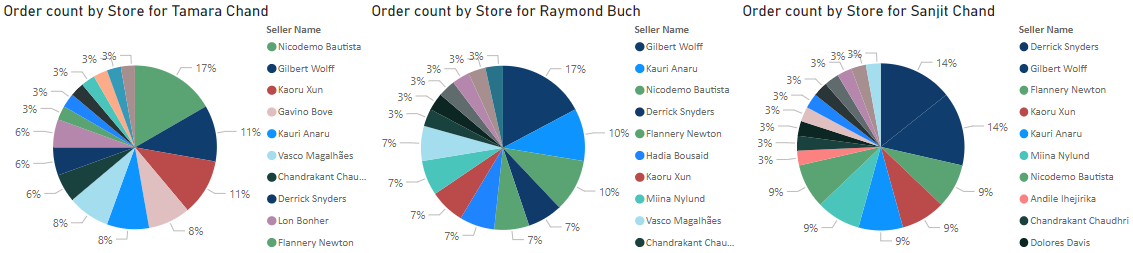


Figure 37 PowerBI Pie Charts of the most frequent sellers for the three most profitable customers of the Superstore.

We can see from these pie charts that the most profitable customers prefer certain sellers over others. Tamara Chand prefers buying from “Nicodemo Bautista”, “Gilbert Wolff” and “Kaoru Xun”. Raymond Buch prefers buying from “Gilbert Wolff”, “Kauri Anaru” and “Nicodemo Bautista”. While Sanjit Chand prefers buying from “Derrick Snyders”, “Gilbert Wolff” and “Flannery Newton”. But just because these are the preferred sellers of the most valuable customers doesn´t mean that the customers’ confidence in the sellers increased throughout time.

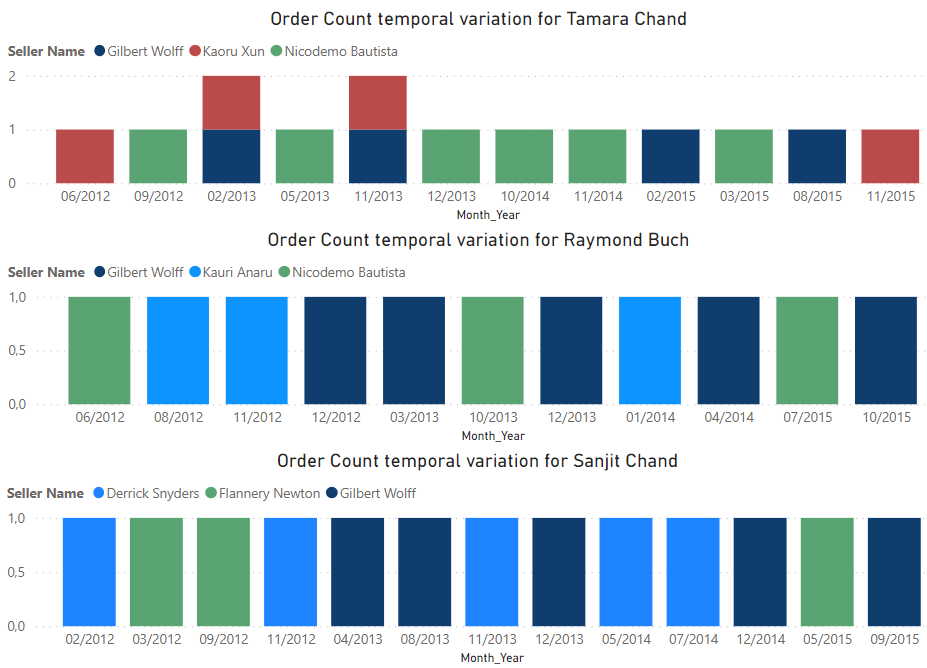


Figure 38 PowerBI Vertical Bar plot of the evolution of Order Count for the most valuable customers’ favourite sellers.

We can see in this plot that the confidence of the most valuable Customers in their favourite sellers didn’t increase with time, the Customers’ behaviour was to place an order randomly when it’s convenient for them and they did not show any significant favouritism in which seller to buy from. Sometimes they made one order with one seller, sometimes with another, it doesn’t seem that the confidence they have on a seller increased with their satisfaction for that specific seller.

#### 3.4.2.3 - What is the percentage of customers, out of all customers of the superstore, that returned a product?

To evaluate the impact of returns throughout time we computed a measure of the Percentage of customers that returned atleast one order and a measure of the Ratio of Returned Orders, and plotted them in function of the Order date, using a Quarter approach.

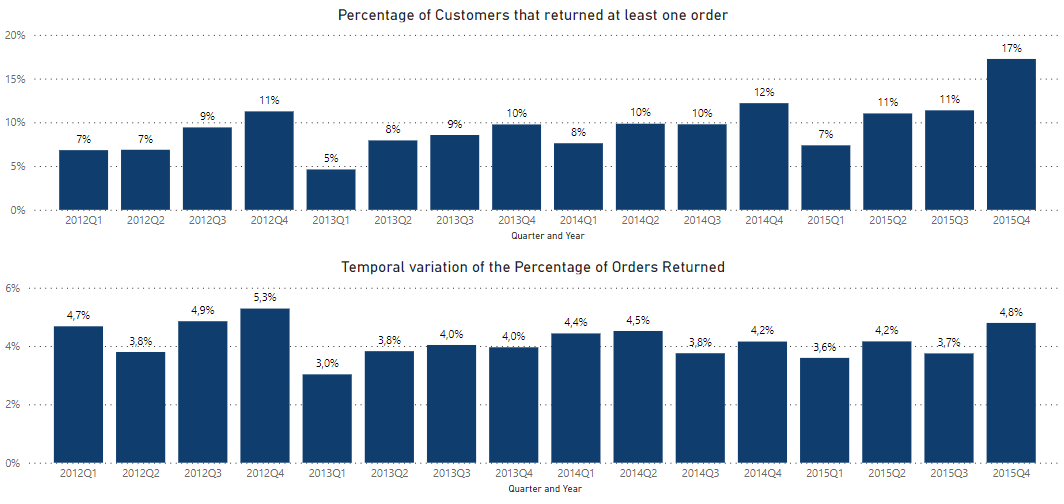


Figure 39 PowerBI Vertical Bar Plots of the ratio of Customers that return orders, and the ratio of returned orders, throughout time.

We can see in the first plot that in general the Ratio of “complaining” customers tends to increase throughout the years, if we look in the same year, the number of “complaining” customers increases in each quarter and decreases when we pass over to the next year. This certainly is because sales increase throughout the year (and years), so in the end of the year there are more orders, so there is more chance of a customer being dissatisfied.

We can observe in the second plot that the ratio of returned orders as been somewhat the same throughout time, having its biggest variation from 2012Q4 to 2013Q1 of 2.3%.

This means that throughout time customer satisfaction has been stable and order quality too.

#### 3.4.2.4 - Are most of the orders of products of the "Furniture" category made by customers with the consumer or corporate segment?

This is one of the more specific questions that we developed. In the following figure we count the number of orders that contained furniture and partition the count by Costumer Segment to see which segment buys more furniture.

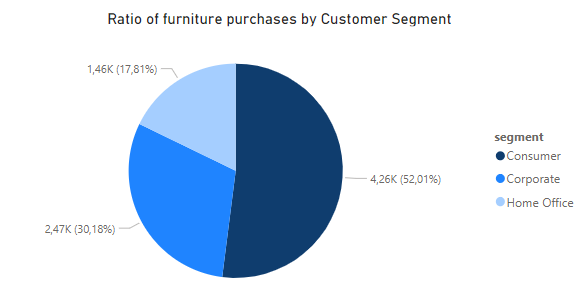


Figure 40 PowerBI Pie Chart of the ratio of furniture purchases by Customer Segment.

We can see that most of the purchases of furniture were made by “Consumer” customers and not “Corporate” ones.

#### 3.4.2.5 - What was the shipment mode that customers preferred the least for every region the superstore operates in?

To obtain this information we plotted the portion of orders done with different shipping modes for all Seller regions.

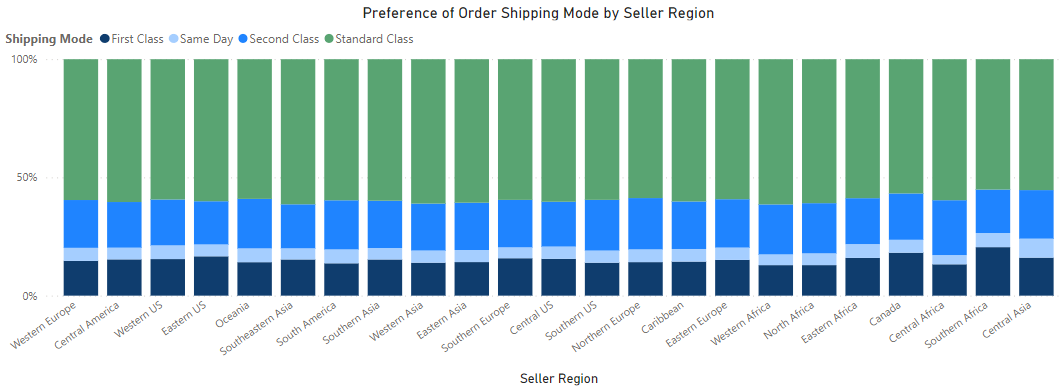


Figure 41 PowerBI 100% Vertical Bar plot for the number of orders placed with different shipping modes in all Seller Regions

We can observe that in all demographics the most popular shipping mode was “Standard”, followed by “Second class”, “First class” and then “Same day”. For every region, the ratio of orders of a certain type of Shipping mode was very similar, so customers don’t change their preferred shipping mode in function of the region of the seller.

*3.4.2.6 - Does the customer importance relate to how often they ask for critical priority orders?*

We selected the 40 most profitable customers to observe if they have similar preferences in Order priority, and if their profitability related to how many times they prefer to order with critical priority.

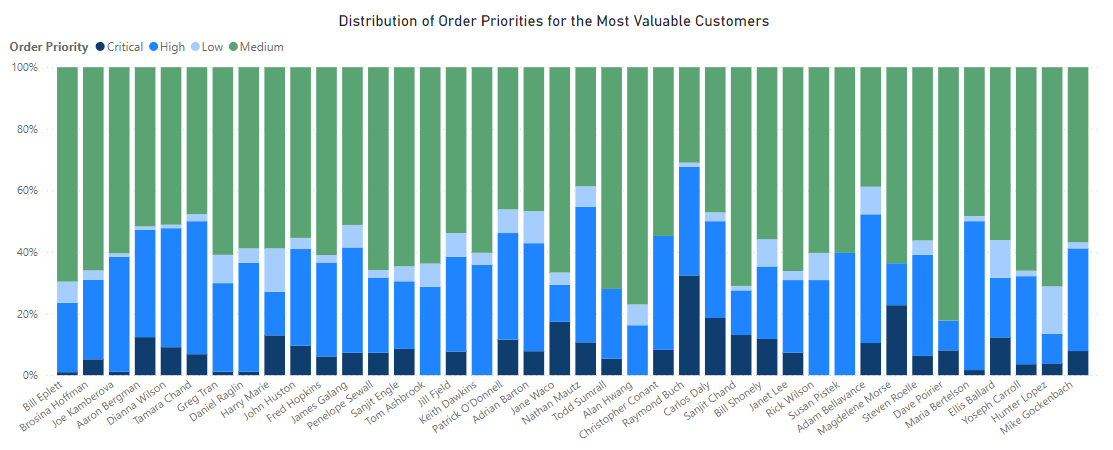


Figure 42 PowerBI 100% Vertical Bar plot for the number of orders placed with different order priorities for the top 40 profitable customers.

We notice some common behaviours in all these customers. The most used priority is “Medium” in most of them, the second most popular is “High”, but then they rarely ask for “Low” and “Critical” priorities.

Apart from that we can see that all these customers have different preferences in priority, and being a valuable customer does not mean that you ask for critical priority orders any more often.

*3.4.2.7 - How many customers that ordered products from the Europe market conducted critical priority orders in the year 2015?*

To answer this question, we plotted a donut chart with the number of customers that made critical priority orders in 2015 for the different Markets.

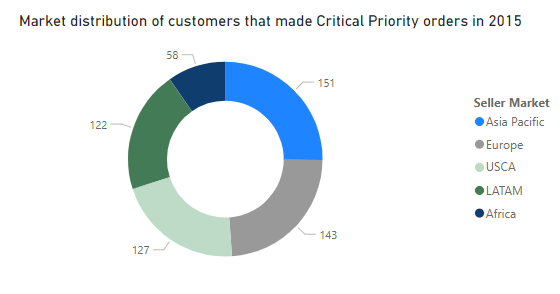


Figure 44 PowerBI Donut chart for the market distribution of customers that made Critical Priority orders in 2015.

We can see that in 2015 there were 143 customers that ordered from the Europe market with critical priority.

### 3.4.3 - Product Analysis

Regarding the analysis of the products three main questions were analysed using only Power BI functionalities: (1) What was the season of the year when orders for products of “GlobeWeis” brand were made in the highest amounts? To follow up on this we investigated on the question: Do the previous results relate to the occurrence of holidays?; (2) What is the category of products that generated the most profit for the superstore? This was followed up by: And what are the respective sub-category of the above categories of products that generated the most profit for the superstore? (3) What are the 6 least sold products? And to finalize the question: What is the average price of the 6 least purchased products? Given that the discount was removed from our facts table, the originally created questions regarding it were removed and replaced with these questions centred around shipping costs.

#### 3.4.3.1 – What was the season of the year when orders for products of “GlobeWeis” brand were made in the highest amounts? Do the previous results relate to the occurrence of holidays?

Knowing what products are more popular at which times in the superstore is relevant to make smart business level decisions that elevate on product profit margins making these questions relevant to find answers to. The first question was generated by having order season information in X axis and the counts of individual products (provided by product key) from the facts table in the Y axis. Using the filtering options of PowerBI only the instances that have product names that start with “GlobeWeis” are used and so the produced chart is achieved.

A picture containing text, screenshot, font, number

Description automatically generated

Figure 45 PowerBI Vertical Bar Plot for the Orders of GlobeWeis line of products per season.

The follow-up question here was if there was any relation to the results with occurrence of holidays. As most products of this brand were ordered in the Autumn season the focus was only on Autumn orders. Using custom measures – a PowerBI tool for analysis – the orders conducted on holidays were obtained alongside the orders conducted on non-holidays (or working days).

A picture containing text, screenshot, rectangle, software

Description automatically generated

Figure 46 PowerBI Vertical Bar Plot for the Orders of GlobeWeis line of products in Autumn made in holidays and in non-holidays.

These plots indicate that it’s not likely that there is a relation between the large number of orders made of products of the “GlobeWeis” brand in the Autumn season and if these orders were made in Holidays or not.

#### 3.4.3.2 – What is the category of products that generated the most profit for the superstore? And what are the respective sub-category of the above categories of products that generated the most profit for the superstore?

This superstore’s range of products are all encompassed within either Technology, Office Supplies or Furniture categories, however discovering the most profitable one can be information of benefit for decision makers to maximise product profit. To generate this plot the axis, take in the product category on Y and the sum of profit on X which is an aggregate measure generated automatically by PowerBI.

A picture containing text, line, font, number

Description automatically generated

Figure 47 PowerBI Horizontal Bar Plot for the most profitable categories of products in the superstore.

As it is observed here products of technology category as those that sell the most in this superstore by far, but one can take this analysis further by drilling down to finding which sub-category out of each one of these categories is the most profitable in their own right which is the follow-up question here. The plots were generated by comparing the sum of profit of the facts table to the subcategories of products table and obtaining only the top instance for each case. To separate by category a slicer was attached to this visual element that changes the produced value in the plot according to category selected giving the most profitable subcategory respective to each category of products.

A picture containing text, screenshot, line, font

Description automatically generated

Figure 48 PowerBI Horizontal Bar Plot for the most profitable sub-category of products in the “Furniture” category.

A picture containing text, line, diagram, plot

Description automatically generated

Figure 49 PowerBI Horizontal Bar Plot for the most profitable sub-category of products in the “Office Supplies” category.

A picture containing text, line, plot, screenshot

Description automatically generated

Figure 50 PowerBI Horizontal Bar Plot for the most profitable sub-category of products in the “Technology” category.

It was noticed that even though the Office Supplies category is more profitable than the Furniture category, bookcases that are considered furniture sell more than appliances that are considered office supplies meaning that there is a large discrepancy in profit generated between bookcases and other furniture products than between appliances and other office supplies products. Also, important to consider the fact that copiers were the most profitable subcategory of all of the superstore.

#### 3.4.3.3 – What are the 6 least sold products? What is the average price of the 6 least purchased products?

Dampening the effect of deficiencies is as important if not more than augmenting existing strengths, this last group of questions focuses on the least sold products by the superstore. Generating the following plot was done by taking product name from the product table as X values and the sum of quantity from the fact table as Y values and finding the bottom 6 values only by using the filter features of PowerBI.

A picture containing text, screenshot, font, number

Description automatically generated

Figure 51 PowerBI Vertical Bar Plot for the least sold products in the superstore.

All these products, throughout the entirety of the considered time window of information this data warehouse holds were sold only once, suggesting that re-ordering more of these may not be a sound idea when the goal is increasing product profit and doing bundles or discounts of these could help clear existing inventory if they still haven’t been all sold.

The follow-up question here investigates the shipping prices of these products and analyses if there is any difference between these products in this respect. The plot was created by creating an auxiliary measure that calculates the average of unique products that were purchased in the lowest amounts and is used in the Y axis while product name stands in the X axis, the filter was again used to single out the least sold products just like in the question above.

A picture containing text, screenshot, font, diagram

Description automatically generated

Figure 52 PowerBI Vertical Bar Plot for the average price of shipping of the least sold products in the superstore.

It is noticeable how the most cumbersome products would have the highest average shipping costs, the Chairs, the Bookcase, and the Scanner all stand higher as it is expected with Pencil Sharpener at the bottom, but “Xerox 20” which is a type of printer, and the “Chromcraft Training Table” would have been expected to place higher alongside the Bookcase and Scanner. This anomalous result invites further analysis which was not conducted but is advised from the inspection of this project.

## 3.5 - Conclusion of the Third Stage

Numerous valuable insights can be drawn from the comprehensive analysis conducted during the Third Phase. This section delves into the key findings of each phase, shedding light on the outcomes achieved. The focal points of this stage encompassed the establishment of the ETL System, and the resolution of the analytical questions formulated during the initial stage.

### 3.5.1 - ETL system

In conclusion, the ETL (Extract, Transform, Load) process that has been developed demonstrates the importance of ensuring data coherence during transformations to ensure error-free loading. By maintaining data coherence throughout the process, we can guarantee the accuracy and reliability of the loaded data.

One key observation is that the loading phase consumes a significant amount of time during the implementation of the ETL process. This is understandable considering the substantial volume of data that needs to be loaded. Efforts should be made to optimize this loading time by employing efficient loading techniques, such as parallel processing or data partitioning, to improve overall efficiency and reduce processing bottlenecks.

### 3.5.2 - Analytical questions

Regarding the analytical questions, based on the analysis performed using both PostgreSQL and PowerBI, some insights are suggested to decision makers.

#### 3.5.2.1 - Seller

Starting with the analysis of the Sellers, the detection of the Top 5 and Worst 5 in terms of total generated profit made it possible to obtain some interesting insights.

First, there doesn’t seem to exist a relation between a customer belonging to a state in the top 5 states in terms of GDPs and being a customer of the top 5 sellers. With this observation, it can be suggested that it probably isn't worth it for the sellers to target the states with the highest GDPs.

Second, the same can be said regarding the customer segment, meaning that there doesn’t seem to exist a relation between the number of clients from each customer segment and being present in the top 5 sellers, given that the patterns of client amount for each customer segment are the same for both best and worst sellers showing that probably isn't a contributory aspect of being a good seller. So probably also isn't worth targeting specific customer segments.

In third place in terms of the most prevalent product categories for both best and worst sellers, there seem to exist some interesting patterns. While for both best and worst sellers the top selling category is Office Supplies, the second and third bestselling isn't the same. For the best sellers, the second-best category is Furniture and for the Worst sellers is Technology. This may lead to a possibility that targeting Furniture may lead to improved sales compared to Furniture. Regarding the subcategories, it was where the most visible differences were detected. Targeting Paper seems to be a better tactic than targeting for example art. This is suggested given that the sub-category Paper is the 4th bestselling for the top sellers while for the worst sellers is in the last places and for the worst sellers Art is placed in 3rd place while for the best sellers is placed in the middle of the line. The significance of these differences could be verified through the usage of statistical hypothesis testing and therefore provide more support for future actions based on these observations.

#### 3.5.2.2 - Customer

From the Customer analysis we learned that the most profitable customers are also the ones that buy more and the most expensive things and even though they have preference on the seller, this preference isn’t built up throughout time as one would expect.

Instead, the most valuable customers buy an item here and there from different sellers, and they don’t progressively buy more and more from each seller.

Even the most valuable customers prefer “Medium” and “High” order priorities as opposed to “Critical” and “Low” priorities, and in general they prefer “Standard” shipping to the alternatives, no matter the Seller Region.

The ratio of order returns hasn’t increased throughout the years, which means customer satisfaction is stable.

The customers that buy more furniture are mostly of the “Consumer” segment.

#### 3.5.2.3 - Product

The focused analysis of the GlobeWeis line of products revealed how most of its sales are during Autumn (145 orders), and that the presence of holidays did not sway the dates of most of the orders. These observations can be useful to do on a product line or category type basis for other products as a form of pattern exploration and provides background information on habits of clients and their preferences of when to conduct orders throughout the year.

Plotting the most profitable categories and subcategories of products ordered from the superstore is also important as it can help shift the focus of decision makers when it comes to restocking, negotiating new contracts with wholesalers, defining discount amounts and times for when to apply them for products on the category and subcategory basis. From our observations the Technology category is the most profitable one, with copiers generating a total of over 259 thousand dollars, in the case of Office Supplies the Appliances stood out almost hitting 141 thousand dollars and Furniture-wise, the bookcases were netting around 161 thousand dollars.

Perusing through the least popular products, all of them having been recorded of selling a single unit each, there are included a bookcase (*Bush Saratoga Collection 5-Shelf, Hanover Cherry \*Special Order*), an electric pencil sharpener (*Boston 1900*), a table (*Chromcraft Training table, Adjustable Height*), a chair (*Global Enterprise Series Seating Low-Back Swivel/Tilt*), a scanner (*Penpower WorldCard Pro Card*), and a printer (*Xerox 20*). Looking into their shipping costs the chair cost 27 dollars, the bookcase cost 13 dollars, the scanner cost 11 dollars and the pencil sharpener, the table and the printer all cost 2 dollars. The decision makers can investigate further on this price fluctuation for the printer and table costing as much as the pencil sharpener as it can show new knowledge relevant to the superstore.

### 3.5.3 - Final Comments

The comprehensive analysis in the Third Stage yielded valuable insights. Key findings include the importance of data coherence in the ETL system and the need to optimize loading time. For sellers, there was no correlation between top states in GDP and top sellers, nor between customer segments and top sellers. Targeting Furniture may improve sales, and differences in subcategories were observed. For the GlobeWeis product line, sales peaked in Autumn, and profitable categories included Technology, Office Supplies (Appliances), and Furniture (bookcases). Investigating low-selling products with varying shipping costs can provide relevant knowledge. Overall, the analysis offers guidance for decision-making in optimizing ETL, understanding sellers, and identifying profitable products.

In summary, the analysis conducted in the Third Stage has provided valuable insights that can inform decision-making across various aspects, including sellers, customers, and products. These findings serve as a foundation for future actions and strategic initiatives, empowering the superstore to enhance profitability, optimize resource allocation, and address market dynamics effectively.