Graphical user interface

Description automatically generated with low confidence

Integração e Processamento Analítico de Informação Project

**Stage 2: Dimensional Modelling**

**Group 1**

Tiago Rodrigues nº49593

Ivo Oliveira nº50301

Martim Silva nº51304

Alexandre Sobreira nº59451

2022/2023

Table of Contents

[1 – Introduction 2](#_Toc132217046)

[2 – Fix problems detected in the previous stage 3](#_Toc132217047)

[3 – Declare the grain and type of the fact table 3](#_Toc132217048)

[4 – Model business dimensions, including data hierarchies 4](#_Toc132217049)

[4.1 - Facts Table 4](#_Toc132217050)

[4.2 - Product Dimension 5](#_Toc132217051)

[4.3 - Date Dimension (Role-Playing – Order & Ship) 6](#_Toc132217052)

[4.4 - Holiday Dimension (Outrigger - Connected to Date Dimension) 7](#_Toc132217053)

[4.5 - Seller Dimension 8](#_Toc132217054)

[4.6 - Customer Dimension 9](#_Toc132217055)

[4.7 - GDP Dimension (Outrigger - Connected to Customer Dimension) 10](#_Toc132217056)

[4.8 - Order Information Dimension 11](#_Toc132217057)

[5 – Identify numerical measurements in the fact table(s) 11](#_Toc132217058)

[6 – Draw the data warehouse star diagram 11](#_Toc132217059)

# 1 – Introduction

As we move into this next stage, we have an opportunity to build upon the work done in the previous stage by creating a multidimensional model that is well-suited for a data warehouse. The focus here will be on identifying and addressing any issues that were detected in the previous stage, as well as declaring the grain and type of the fact table(s), modelling business dimensions, and identifying numerical measurements that will be included in the fact table(s). Additionally, we will be creating a data warehouse star diagram that will provide a clear visual representation of our data model. As we undertake these tasks, our goal will be to create a comprehensive, efficient, and effective data warehouse that will serve the needs of our organization.

# 2 – Fix problems detected in the previous stage

Irregularities with the original formatting of the data, when taken from each source in stage 1, were treated in this stage. Different representations for data with the same meanings were brought down to a unified standard such as the object type representations of dates for orders and holidays. All columns referencing an ID were changed to hold values in a string format, this was done also for other columns where it would make sense to have their values be strings like City, Product Name, etc. Columns with a high count of missing values that did not contribute in major ways to our business process were promptly discarded. Initially columns with Boolean values in string formats like “Yes” or “No” were to be replaced with actual Boolean True/False values, however, this will be changed to remain in string format but be more descriptive of the character of the data the value is about, as an example we change the values of a column “Returned” from “Yes” or “No” not to True or False but to “Returned” and “Not Returned” to be adequately elucidative for subsequent analyses in stage 3. For this stage, we also changed the average GDP per state column associations with the states themselves to allow the creation of a primary key that would describe both a state and its average GDP together. Column names and String values with characters with extra unnecessary accents that can cause file encoding format issues were standardized by using the “no accent” versions of the original strings such as the column names “Valentine’s Day” or “New Year’s Eve”. All the above pre-processing was done using Python 3.

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

To define the specifics of the star schema to be used for the building process of the data warehouse it is also necessary to clearly define what is going to constitute the contents of each of the tables that follow suit.

Under a star schema representation, our data will be separated in a fact table where a row in this table stores (and details as much as needed for posterior analyses) each recording or fact of the same type of business event that the would-be operational system would perform (the grain) and several dimension tables that are connected using unique identifying values (respective primary keys) to the fact table (holding foreign keys to these dimensional tables) and each of these holds specific information about a part of the business and for that part alone.

Our delineation of what constitutes the grain in this project as we have a single fact table, and the chosen business process of focus is on product profit will then be that of a row that represents a **purchase** of some **quantity** of a **single product** belonging to a type of **category** and **subcategory** that belongs to a **specific order** made on a **specific ordering date** that was made with a **priority** of low, medium or high degree. This purchase was shipped with a **shipping cost** at a specific monetary value amount associated with a **shipping mode** that included first class, second class, standard class or same day options with a specific **shipping date** and a **discount value** on the product ranging between 0 (inclusive) and 1 (exclusive). It was conducted by a **single named customer** belonging to a corporate or consumer **segment** to the superstore from a specific **market** done from a specific **region**, **country, state and city** which was **sold** at a specific monetary value amount and incurred in a **profit** of another specific monetary value amount unless it was a product that belonged to an order made from a **certain region** that was later **returned** to the superstore.

# 4 – 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.

Another important aspect of modelling business dimensions is the creation of data hierarchies. Data hierarchies represent the relationships between different levels of data within a given dimension. For example, a product hierarchy might include categories such as product type, brand, and model. By defining these hierarchies, businesses can analyse data at different levels of granularity, providing deeper insights into their operations.

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.

## 4.1 - 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; Discount; 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | The quantity bought of a certain product in a certain purchase | NUMERIC | 2 |
| Discount | Discount applied to a certain purchase | NUMERIC | 0,7 |
| Profit | Profit obtain in $ from a certain purchase | NUMERIC | -157,086 |
| Shipping Cost | Cost in $ of a certain shipment of a certain order | NUMERIC | 5,69 |

Table 1 Facts Table description with Column Name (Field) Column description(Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

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

A picture containing text, monitor, screen, black

Description automatically generated

Table 2 Five first rows of the Facts Table

## 4.2 - 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 analyze 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.

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

Table 3 Product Dimension description with Column Name (Field) Column description(Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

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

Text

Description automatically generated

Table 4 Five first rows of the Product Dimension Table

## 4.3 - 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 commonly 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-dow and roll-up operations) and also 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Weeks in Year of o certain Order/shipment | NUMERIC | 1 |
| Day Number Month | Number of Days in a Month of a certain Order/shipment | NUMERIC | 1 |
| Day Number Week | Number of Days 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 |

Table 5 Date Dimension description with Column Name (Field) Column description(Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

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.

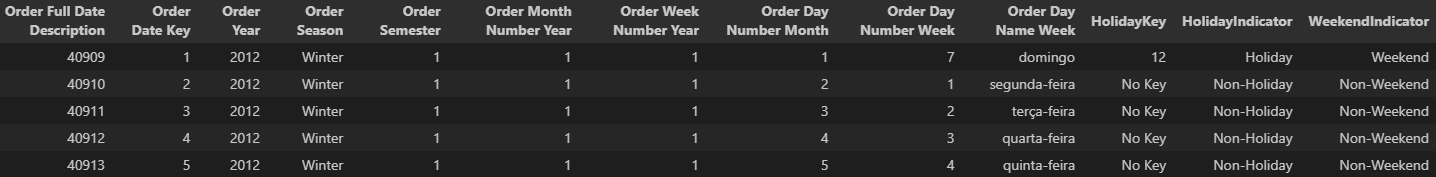


Table 6 Five first rows of the Date Dimension Table for Order Date

Graphical user interface, application

Description automatically generated

Table 7 Five first rows of the Date Dimension Table for Shipment Date

## 4.4 - 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 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

Table 8 Holiday Dimension description with Column Name (Field) Column description(Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

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

A picture containing text, road, screenshot

Description automatically generated

Table 9 Five first rows of the Holiday Dimension Table

## 4.5 - 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Data type** | **Example** |
| Seller Key (PK) | Primary Key | NUMERIC | 1 |
| Seller Name | Name of the Seller | VARCHAR | Kaoru Xun |
| Seller Market | The 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 |

Table 10 Seller Dimension description with Column Name (Field) Column description(Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

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

Graphical user interface, application

Description automatically generated

Table 11 Five first rows of the Seller Dimension Table

## 4.6 - 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 analyzing 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 analyzing 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | The segment to which 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 |

Table 12 Customer Dimension description with Column Name (Field) Column description(Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

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

A black screen with white text

Description automatically generated with low confidence

Table 13 Five first rows of the Seller Dimension Table

## 4.7 - 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 takes into account 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 $ per state between the years 2012 and 2015.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

Table 14 GDP Dimension description with Column Name (Field) Column description(Description), PostgreSQL datatype (Data Type) and an example of a value (Example).

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

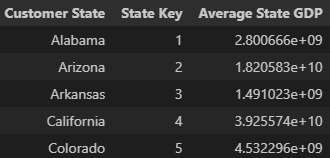


Table 15 Five first rows of the GDP Dimension Table

## 4.8 - 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

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

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

A black screen with white text

Description automatically generated with medium confidence

Table 17 Five first rows of the order Information Dimension Table

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

# 6 – Draw the data warehouse star diagram

The most popular data structure (or data modelling technique) used in data warehouses (DWs) is the star schema (SAMSTAR: A Semi-Automated Lexical Method for Generating Star Schemas from an Entity-Relationship Diagram), 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) (The Translation of Star Schema into Entity-Relationship Diagrams Michael Krippendorf and Il-Yeol Song).

In the star schema, the data is organized into a central fact table and surrounding it dimension tables (Star Schema Advantages on Data Warehouse: Using Bitmap Index and Partitioned Fact Tables), 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 (Efficient Execution of Joins in a Star Schema). The dimension tables contain the key which is used to connect to the central fact table and also contains descriptive data that provide context to the measures in the fact table (for example time periods, geography and product categories) (An Analysis of Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional Modeling). 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 (Star Schema Advantages on Data Warehouse: Using Bitmap Index and Partitioned Fact Tables). 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:

* Facts Table: This table includes measures such as Sales, Quantity, Discount, Profit, and Shipping Cost, along with foreign keys to link the dimension tables and the Transaction Key.
* Customer Dimension: This dimension table includes attributes such as Customer Key, Customer ID, Customer Name, Customer Segment, Customer State, Customer Region, Customer City, State Key, and Customer Postal Code.
* GDP Dimension (Outrigger): This dimension table includes attributes such as State Key, Customer State, and Average Region GDP.
* Product Dimension: This dimension table includes attributes such as Product Key, Product ID, Product Name, Category, and Sub Category.
* Order Date Dimension: This dimension table includes 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
* Ship Date Dimension: This dimension table includes attributes such as 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
* 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.
* Order Information Dimension: This dimension table includes attributes such as Order Key, Order ID, Returned Indicator, Ship Mode, and Order Priority.
* Seller Dimension: This dimension table includes attributes such as 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.

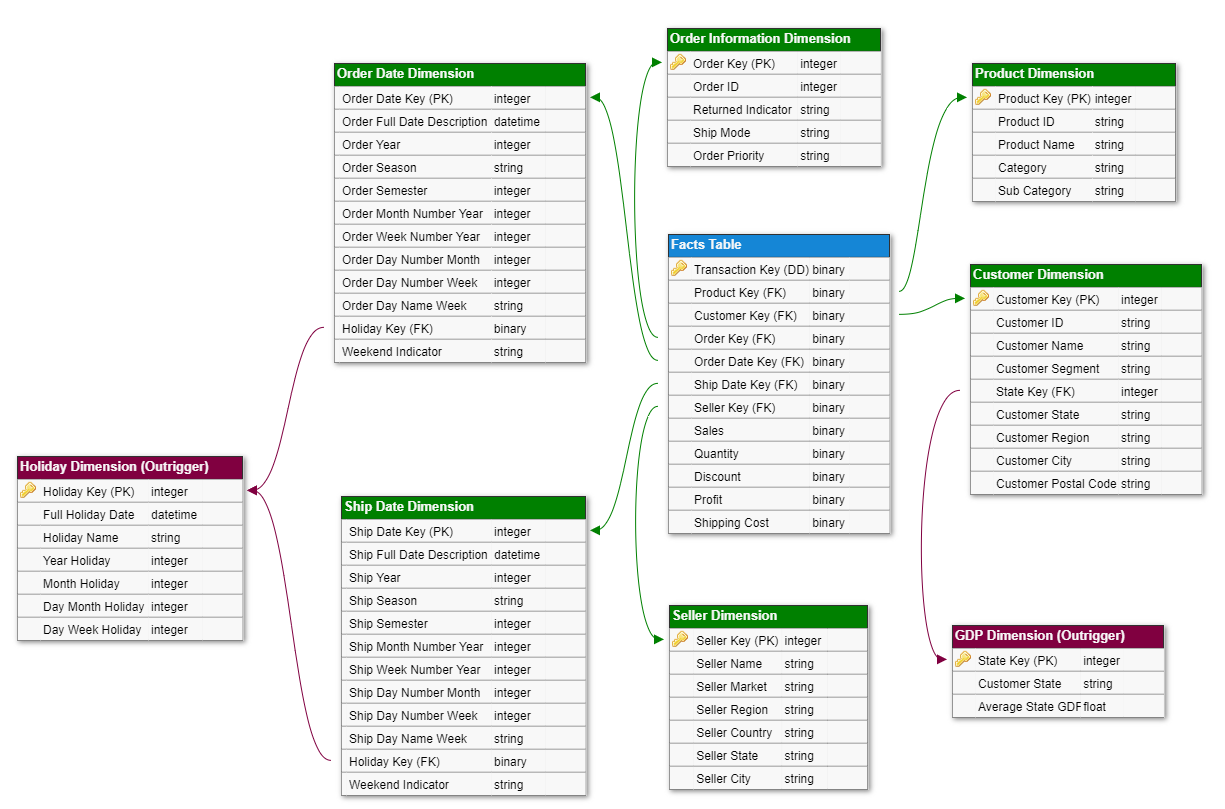


Figure 1 Star Schema containing the Facts Table (blue) at the Center with the Dimension Tables (green) surrounding it with their associated Outriggers (purple)