Business Intelligence

1st Assignment

MASTER'S DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Group 23

Andreia Bastos, m20210604

João Silva, m20211014

Pauline Richard, m20211019

Sarra Jebali, m20210765

April 2022

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Index

1. Introduction

Digital innovations have drastically transformed our daily lives, with one of the most profound effects being noticed in the corporate sphere. Data-driven tools and tactics are now available to companies and represent a major change for them to understand more about their customers and themselves than ever before.

In that context, we have been called by a major technology retail company to provide them with a solution that will give them a daily view of their product sales and cost in all of their Portuguese online stores. Indeed, since retail businesses manage an enormous amount of information – from supplier data to customer buying behavior, employee information to inventory details – every interaction and data point offers an opportunity to make their retail business more successful. By utilizing BI effectively, our goal is to help the company have access to more actionable data, gain valuable insights into market trends, and enable a more strategically oriented decision-making approach.

For today's mission, we will start by explaining the Business context and Business needs of the company. Then, related to those requirements, we will build step by step the Data Warehouse, and explain the process of data integration.

2. Business Context and Understanding

2.1. Business Context

Before building a data warehouse, it is important to have an overview of the business's main operations, looking at key figures and numbers. Understanding the retail company business context will help us in building better and more relevant tools, for better insights and recommendations.

First, we decided to look at the sales performance of the business. We started by displaying an overview of the monthly sales during 2020. We could see, from the Area graph in Figure 1, that the business is impacted by high seasonality. March and July represent the months with the lowest profit whereas January, August, and December exhibit the highest sales volume.

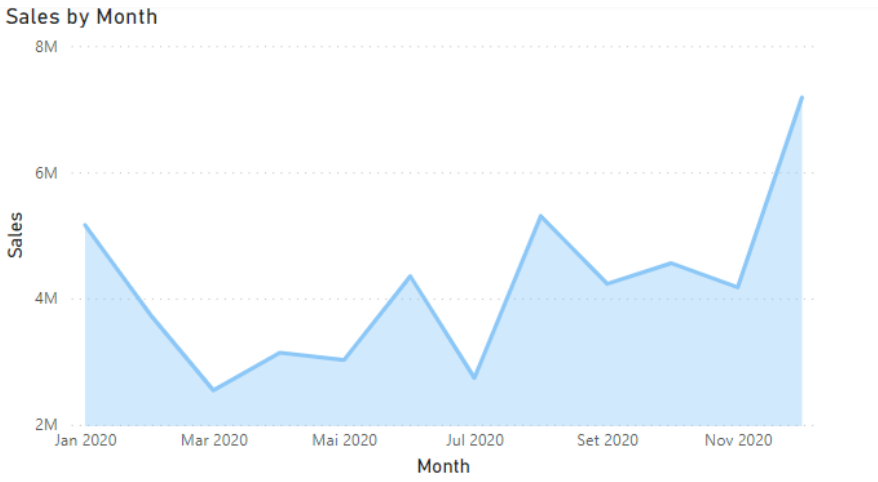

Figure 1: Sales by Month for the year 2020

Then, we analyzed the performance of each of their 5 stores, using a stacked bar chart, in Figure 2. As we can see, Lisbon has the biggest sales volume, which can be expected since it has an additional store (3 are in Lisbon and 2 in Porto).

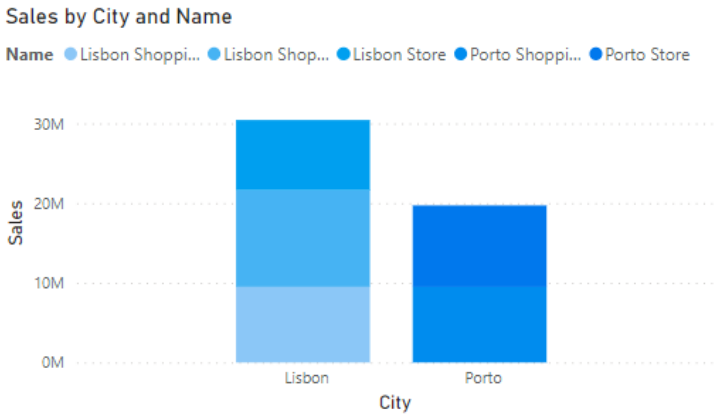

Figure 2: Sales volume of each store per City.

In the second part, we looked at the profile of their customers. First, two very important characteristics to analyze were: the market (where they are coming from?) and the segment (who is buying?). Looking at the doughnut charts in Figure 3, we could see that the distribution of clients per segment is almost equal, with the Private Segment representing about 50.59% and the Company one, 49.41%. However, in Figure 4, we can see customers are divided into two distinct markets: Japan and USA. The distribution shows a slightly bigger difference, with Japan doing around 54% of the total purchases.

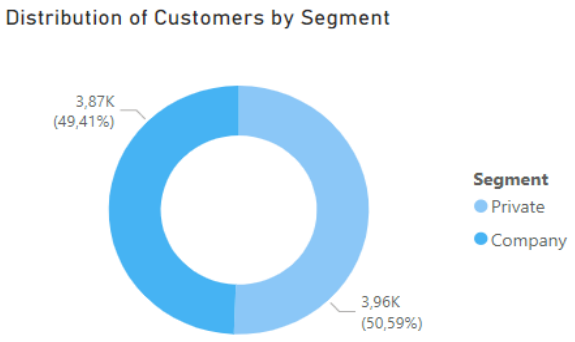

Figure 3: Distribution per Segment

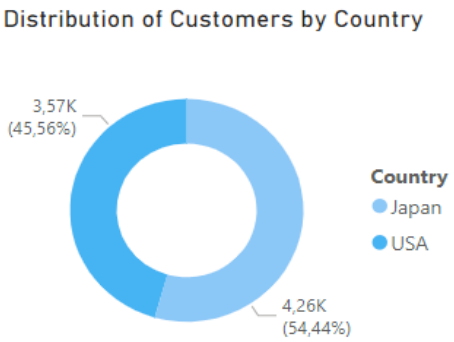

Figure 4: Distribution per Market

To complement those analyses, we looked geographically at who were the 6 top states that were buying the largest quantity of products. The results can be seen in Figure 5, where we can observe that 5 out of the 6 states are from Japan, which we can conclude is the major market for the retail company.

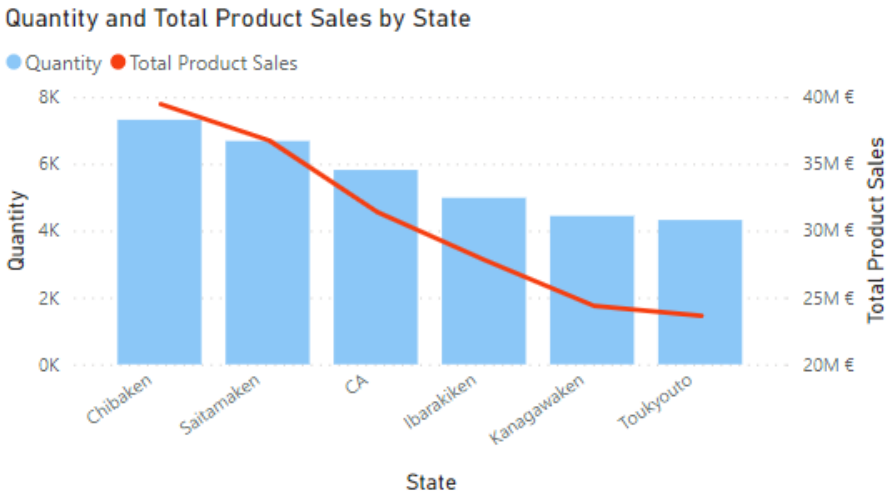

Figure 5: Distribution per Market

Finally, using combo charts, we analyzed the different types of families of products that the company is selling. From Figure 6, we can first conclude that all products are tech products coming from the same brand: Apple. We can also conclude that their Top Sales in terms of quantity are the Apple watches, however, in terms of value, they are getting higher revenue from selling the iPhone. Apple Tv, HomePod, and the Apple Keyboard are the three less profitable products, which can be explained by the difference in the unit price (for the apple keyboard) or the quantity sold (Apple TV).

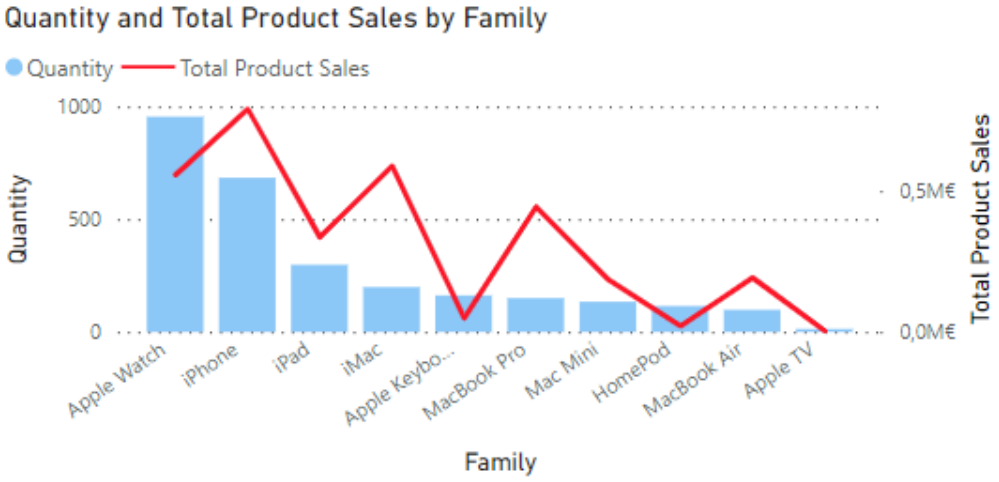

Figure 6: Total Quantity and Sales per Family of product

2.2. Business Needs

With the recent expansion of online business in the last few decades, the technology retail company is under pressure to do things faster and better and meet its sales targets. To fulfill this purpose, it wishes to improve its market operation for the Portugal online stores by getting a better

overview of its daily products' sales and costs. Indeed, having the right information at the right time will help the company to do more data-driven decisions and reach its goals. Understanding and predicting demands will also lead to better productivity and cost savings since the stock level and human capital can be scheduled more efficiently.

We took into account the several requirements and needs of the three departments: Commercial, Logistics, and Sales, and we summarized them into pinpoints that we will answer using PowerBI software:

Commercial

1. What are the predicting variation of the Costs and gross margin volume? with and without tax.
2. How the Sales value are geographically distributed?
3. What is the evolution of sales volume per year, month, and week?
4. Does the key business objective of increasing the target sales by 25% fulfilled? looking at the KPIs
5. What are the predictions of sales per family of products for the next months?

Sales

1. How well each of the 5 stores is performing, in terms of sales volume, and compared to the target?
2. What is the sales performance by product compared to its target value?

Logistics

1. How long does it take to deliver an order? (Status: delivered)
2. How many orders are still in progress? (Status: not Delivery)
3. How much each customer is generating sales?

3. Data Integration Process

3.1. Loading the Source data

The data was gathered via excel files, comprehending about sixteen spreadsheets described below.

Store: This file has information about the five Portuguese online stores, including the name, full address, phone number, and email;

Product: The second sheet gives details about the products: their family (iPhone, Apple Watch, etc), release date, and date when they were discontinued (if there is one);

Target: The target contains data about each store's sales per month;

FamilyImages: Here we have each family (iPhone, Apple Watch, etc) associated with a link to their image;

Customer: This file contains detailed information about each customer and their customer identification, including first and last name, customer segment (private or company), phone number, and address.

Invoices: This folder contains five sheets, one per store. In the files, we can find information regarding each store's orders: the date when it was performed and delivered (if they have been processed), the ID of the customer responsible for them, and the order number.

InvoiceDetails: Similar to the previous one, it also contains five files, one per store. Even though it is also about the orders, here we have information about their composition: the order number, what products were bought, their price per unit and the tax rate applied to that product, and the delivery cost;

LocationDetail: In here we have a public database with information about locations in the USA and Japan.

3.2. Data Preprocessing

Before going any further, once the data were loaded, we had to preprocess each file so we will keep only the relevant data for our analysis. Here is a summary of the actions that we performed per file:

Customer
Our preprocessing approach for this table was to first remove the 2 null top rows and change the type of columns *Customer_id* to int64, and *Customer_Lname, Customer_Segment, Zip*, and *Phone* to text.
Moving on, we merged the columns *Customer_Fname* and *Customer_Lname* to form *Custumer_Fullname*, and extracted all the elements in the column Zip before the first comma to get the format zipcode-city in order to merge with the 'Location' table using a left-join.

We finally removed the *Phone* column as we thought it would not be relevant to meet any of our business requirements.

Location
Here we repeated the process of extracting the city before the first comma in the *City* column, we then split the column *State/Region/District/Country* by '/' in order to get all these details in different columns, and finally concatenated the *City* and *Zip* columns in order to perform the 'Customer' table merge.

Conversion EUR/USD
This is a simple table that only provides information on the conversion of Euros to US Dollars, so our main goal here was to change the badly presented date format, in order to connect it with the 'Sales' table. To do this we extracted the last 2 digits of the original format to obtain the month, so for example, 202001 becomes 2020, a new column is created with the value 01 and we merge these two columns by adding a "/" in order to convert it to the correct date format.

FamilyImages
This table only contains images related to its products and the data was all consistent except for its format, which led us to transpose the table and eliminate the resulting extra rows and columns.

Product
Started by changing the type of all the columns to the types we assumed to be the more adequate. Filled the null values in *Family* using the FillUp function, renamed the *Model Code* column to *ProductID,* and merged the table with 'FamilyImages**'** by the *Family* column. We ended up deleting the release_day and discontinuous columns as we figured these 2 columns will not be relevant for our analysis.

Store
When dealing with the store table we start by transposing this table since we considered that it would be a more favorable display than the original one. We then changed the types of the columns adequately and finally removed the bottom row as it contained unformatted and dirty data.

Invoices and InvoiceDetails
It was clear from the beginning that the information contained in both Invoices and InvoiceDetails folders were somehow going to converge into a fact table, however, our challenge here was to obtain a general table in which we could concentrate all the information regarding the clients' invoices.

To do this we applied two functions, *get_details* and *get_header* that created 'Invoice_Details' and 'Invoice_Header' respectively by concatenating all the information in each folder(originally contained one table per store).

For the final 'Invoice_Details' table, we had some minor incoherences that we had to deal with, such as some decimal plates in the *TAX_RATE* column. We also had to change some column types according to their accurate representation like in previous data transformation tasks along with the project and changed how the *UNIT_PRICE* and *DELIVERY_COST* information were displayed by removing the prefix "EUR".

Regarding 'Invoice_Header' the process was very similar regarding the change in column types. One interesting incoherence we found in the *ORDER_DATE* and *DELIVERY_DATE* columns was a record for the 29/02/21, that would be impossible so we just replaced this with 28/02. We ended up also removing 4 extra columns with null values but decided to keep the null values in the delivery_date column as they could only be a representation that the order was not delivered yet.

Sales
This table was obtained through the merging of 'Invoice_Details' and 'Invoice_Header' by *ORDER_NUMBER*. Besides doing the correct type changes, we also added a new column, *Start of Month*, which was our solution to merge to our fact table the information stored in the 'Conversion EUR USD' table.

Target
The approach on this table was relatively simple as we only had to change types, remove null values from all its columns, and finally merged the conversion rates from the 'Conversion EUR USD' table.

4. Building our Dimensional Model

For designing our Data Warehouse, we decided to follow the Kimball Methodology. Introducing the Start Schema approach in 1996, the first principle is to look at the requirements and needs of the business (done in part 2.2) and identify the fact table (or facts) and its attributes.

Going from an Entity Relationships model to a Dimensional Model requires 3 steps, defined in 2002 by Moody and Kortnink:

1. Classify Entities
2. Identify Hierarchies
3. Produce the Dimensional Model

4.1. Classify Entities

Starting with step 1, we have to first classify the entities of our ER Model into different categories. In Figure 7 below, we can see the classification we have done:

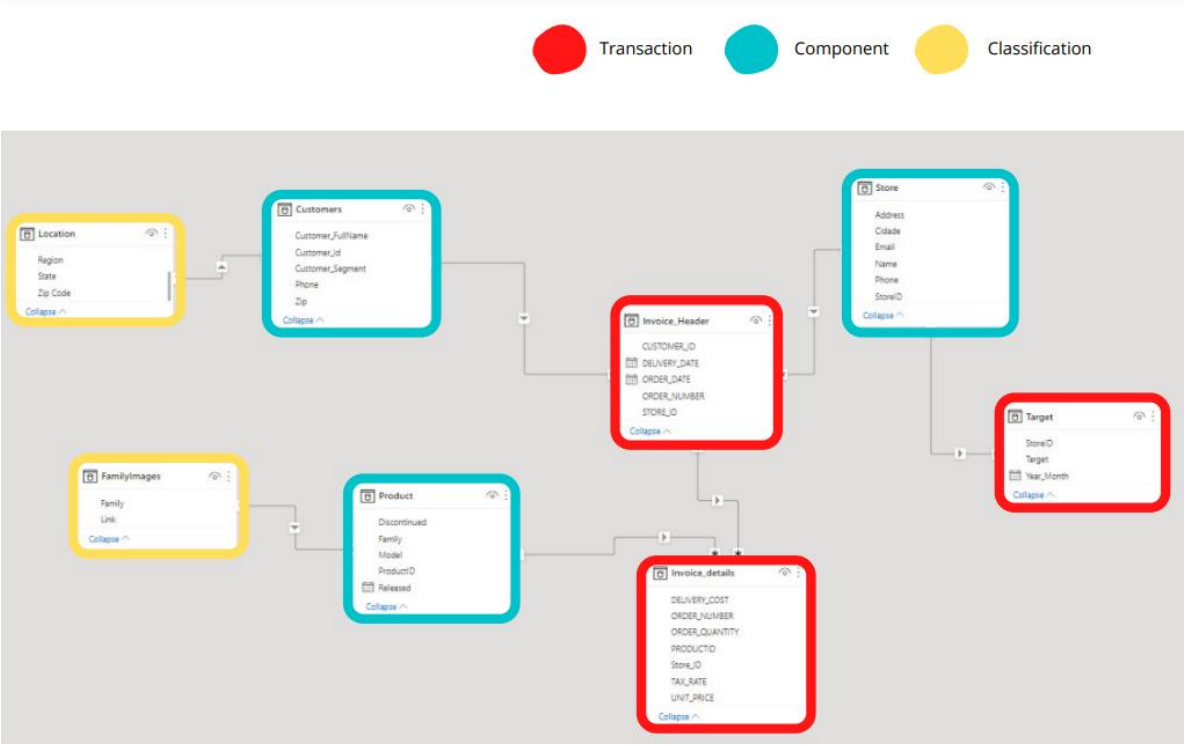

Figure 7: Entity Classification

The Transaction entities (in red in Figure 7) are the tables that contain details about events that happened in a moment m, with measurements. They are useful for decision-makers because they can be quantified and summarized into significant numbers. In this case, we have the invoice header and details that contain all the information about the order made by the client: unit price, order and delivery dates, quantity... etc. We also have the Target that contains the monthly sales goal in Euros.

The Component entities (in blue in Figure 7) are, as implied in their names, the detailed business components of the Transaction entities. They are directly linked using a One-to-Many relationship and will be the base of our Star Schema. Here, we can see that the transactions tables are defined by:

- Customers Table: answer the question of who made the purchase?
- Product Table: answers to what product it has been sold?
- Store Table: answers to where the purchase has been made (out of the 5 stores)?

Finally, we have the Classification entities. They are functionally dependent on the Component Entities, meaning each value of the Component entities is precisely associated with one value from the Classification entities. They are useful to create constraints between two sets of attributes but also contribute to the database's normalization, essential to reduce redundancy and improve its efficiency. In Figure 7, the Classification entities are in yellow, one is Location and is dependent on Customers, and the other one is FamilyImages linked to the Product table.

4.2. Identify Hierarchies

Dimensions are organized hierarchically, using a parent-child relationship. It allows users to navigate between the different levels and to manipulate data for analysis more easily. Here, for Step 2, the aim was to identify the hierarchical relationships in the ER model. It is represented by any segment of entities linked with 1:n relationships and going in the same direction.

We could identify 4 maximal hierarchies in our data model:
Location > Customers > Invoice_Header > Invoice_details
FamilyImages > Product > Invoice Details
Store > Invoice_Header > Invoice_details
Store > Target

Maximal means that we cannot extend the hierarchy by adding an entity, in any way. Moreover, there are minimal entities that are positioned at the end of maximal hierarchies, and maximal entities (also called "root" entities) that are the ones without any 1:n relationships. Here we have two minimal entities (Invoice_details and Target) and three root entities (Location, FamilyImages, and Store).

4.3. Produce Dimensional Models and Results

The final step consisted of going from the Entity-Relationship Model to a Dimensional one, a Start Schema. We produced it using two different operators.

<u>Dimension tables</u>

Operator 1 is about collapsing the Component entities established above, into lower-level entities (or classification tables) to generate the Dimensions.

In our case, we collapsed the FamilyImages entity into the Product Entity, same for the Location Entity into the Customers entity. In Figure 8, we can see both Product and Customer entities kept their initial attributes, but have now the additional ones of the respective children entities. At the end of this stage, from collapsing hierarchies, we come up with two new dimensions: Product and Customer. The store is one of the other dimensions but by itself, it didn't have any lower-level entity.

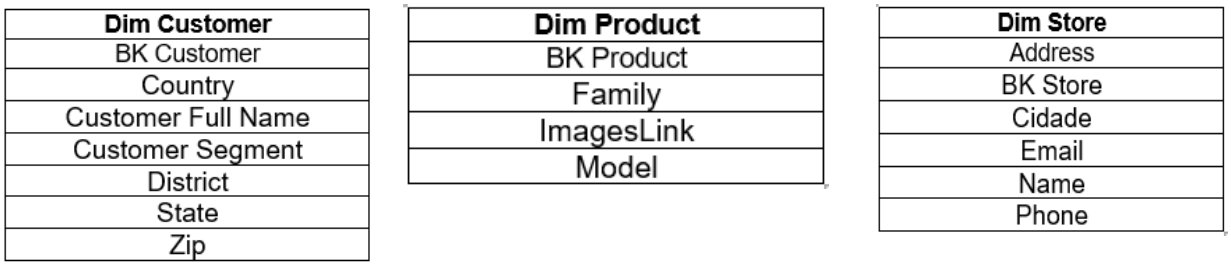

Figure 8: Customer, Product, and Store dimension tables

Figure 9: Date dimension tables

Moreover, we were missing a time dimension. Time historical data is an important part of any analysis, and therefore it is an important component in any data warehouse. For this purpose, we built the dimension table Date (Figure 9).

Fact tables

Each of our transaction entities results in a Fact table. However, we could see a hierarchical relationship exists between Invoice_Header and Invoice_details entities. In this case, we collapsed the children's table (Invoice_details) into Invoice_Header and created the Fact table Sales, so it is possible to 'drill down' between all the different transaction levels. Our second Fact table is Target.

Coming along with building the Fact tables, we had to define the granularity, meaning the lowest level of information that we wanted to be stored. We know that the company wants to *"have a solution providing a daily view of product sales and related costs"*. Following those needs, having 'day' as the lowest level of granularity in the date Dimension was what made more sense, so we could afterward perform the adequate analysis. However, for the Fact Target, 'month' is the lowest level of granularity needed, since the sales objectives are only defined monthly.

Fact Sales
DELIVERY_COST
EUR_USD_Conversion_rate
FK Customer
FK Delivery Date
FK Order Date
FK Product
FK Store
ORDER_NUMBER
ORDER_QUANTITY
TAX_RATE
UNIT_PRICE

Figure 10: Sales Fact table

Fact Target
Conversion_rate_EUR_USD
FK Date
FK Store
Target

Figure 11: Target Fact table

The role of Operator 2 is about aggregation attributes, produced to summarize data gathered and generated from aggregating different numerical attributes of the key dimensions. As we can see in Figure 12, we don't have any aggregated fact.

Before finalizing the schema, we created a surrogate key for the dimension Date table and merged it with the fact Sales and Fact Target, in order to have a foreign date key that will replace order_date, delivery_date, and the target_date.

Final Star Schema

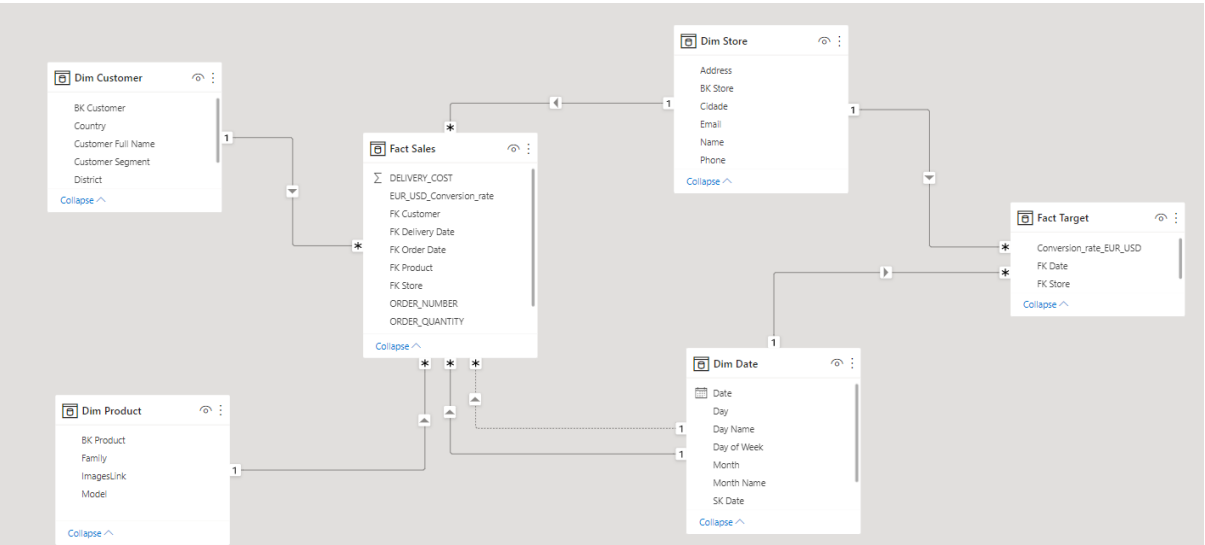

Figure 12: Final Star schema

Our final dimensional model is a combination of a two-star schema, one resulting from the two Invoice transaction entities (Fact Sales) and another one resulting from the Target transaction entity (Fact Target). It has a total of four dimensions and two fact tables. We can also observe that the two fact tables (Target and Sales) are not directly linked to each other, they are only sharing two dimensions (Store and Date). This set of star schemas, combined but not directly linked, is more commonly called a Galaxy schema.
We can notice that the dimension Date has a double connexion with the fact sales. This is Due to the fact that the sales table has 2 important dates in it, order_date and delivery_date. However, only one of those connexions is active at a time.

5. Conclusion

By analyzing the company's needs and challenges we believe that our approach in this initial phase of cleaning data and creating a Dimensional Model met our expectations.

We followed the Kimball methods' best practices when thinking about our design and by already taking into account some company requirements, we managed to create solid grounds for the build of a successful Data Warehouse in the next phase of this project.

6. References

How business intelligence solutions benefit the retail industry. Synoptek. (2022, February 4). Retrieved April 25, 2022

Davidiseminger. (n.d.). *Tutorial: Shape and combine data in power bi desktop - power bi*. Power BI | Microsoft Docs

andyandy1733, & NickWNickW 5. (1969, February 1). *Star schema sales and Goal*. Stack Overflow.

Moody, D., & Kortink, M. (1970, January 1). *[PDF] from enterprise models to dimensional models: A methodology for Data Warehouse and Data Mart Design: Semantic scholar*. undefined