Disseny de fluxos ETL amb eines dbt i Snowflake

Marc Sedó Galimany

Abstract—This final degree project addresses the challenge of identifying the causes of package losses or delays in a shipping firm. The objective is to develop a comprehensive solution by leveraging data engineering techniques and implementing an ELT (Extract, Load, Transform) process to create a dimensional model. The project will utilize dbt (Data Build Tool) and Snowflake to build a well-structured star schema that enables efficient analysis of shipping data. By visualizing and analyzing this data, valuable insights will be extracted to inform the company's operations. Furthermore, the project will employ predictive modeling techniques to forecast package statuses, enhancing the company's ability to make data-driven decisions and improve overall operations.

Index Terms— Data modeling, dimensional model, star schema, dbt (Data Build Tool), Snowflake, Data warehousing, ETL (Extract, Transform, Load), Business Intelligence, External data enrichment, Fact table, Dimension table.

1 Introduction – Business case

Nowadays, people are starting to use shipping companies a lot more in order to receive their packages instead of going to pick them up in person. In fact, the number of parcels sent has been booming in recent years, an amount that has already gone from forty-three billion in 2014 to 159 billion in 2021 [1]. In this report, we will talk about how to solve some of the most frequent problems that these types of companies suffer from.

The company in question is a shipping firm that needs to analyze shipping data in order to make data-driven decisions about its operations. The data is stored in a single table that contains all of the shipping information, including shipment date, orders cost, shipment location, and other relevant information.

They operate globally and use containers to transport goods across oceans. They also have a network of trucks and vans that are used for local transportation and last-mile deliveries. The company is interested in analyzing delivery times, transportation costs, and shipment volumes.

The company currently uses spreadsheets to access and analyze their shipping data in which they manually input data and use pivot tables to create reports and visualize their data.

Their main problem is the loss and delay of packages, and it is for this reason that they want to investigate what are the reasons or the conditions under which these delays or losses occur.

In order to achieve this, they do not have enough with the only table they currently have, but they need a dimensional data model with which they can make visual reports in order to display the data properly and thus discover what the conditions are with which these losses occur.

The entities we know from this initial table, and which can therefore help us to solve the problem are customers, carriers, products, geography, date, and delivery.

2 State of the art

ELT (Extract, Load, Transform) is essential for transforming a single table into a dimensional model. It enables efficient handling of large data volumes and accommodates complex transformations and enrichment. ELT supports an iterative development approach, facilitating quick exploration and analysis while refining the dimensional model over time. It provides flexibility and agility by allowing storage of raw data in various formats, adapting to changing business needs.

Currently there are several tools that can be used for the purpose of building a dimensional model using an ETL tool and later used to build reports.

I will be using dbt [2] as the ETL tool, which is an open-source command-line tool for modern data warehouses that helps data analysts and engineers build, document, and maintain data transformation pipelines in a structured and modular way.

- *Contact email: 1564815@uab.cat*
- *Work tutor: Antonio Espinosa Morales (Àrea d'Arquitectura i de Tecnologia de Computadors)*
- *Course: 2022/23*
- *Degree: Enginyeria de Dades*

dbt helps enforce best practices such as modularization, version control, testing, and documentation, making it easier to maintain and collaborate on data transformation pipelines over time.

Another ETL tool is Apache Airflow [3], and the choice between them depends on your specific needs and preferences. If you need to focus on data transformation and want a more structured approach, dbt might be a good choice. If you need a more flexible and general-purpose platform for managing workflows, Apache Airflow might be a better fit.

Then, I will be using Snowflake [4][5] as data warehouse, which is a cloud-based data warehousing platform that allows organizations to store, process, and analyze large amounts of structured and semi-structured data. It is built for the cloud and designed to be highly scalable, flexible, and easy to use.
Snowflake separates storage and compute, which means that users can easily scale computing resources up or down based on their needs without having to worry about managing infrastructure.

An alternative to Snowflake is Google BigQuery [6]. Snowflake integrates well with a wide range of data sources and tools, including popular BI and ETL tools. BigQuery is also highly integrable with many third-party tools, but its integration options may not be as extensive as Snowflake.

Finally, for the reports, I will be using Power BI [7], which is a business analytics service provided by Microsoft that allows users to visualize and analyze data from various sources. It enables users to connect to a wide variety of data sources, including spreadsheets, databases, cloud-based and on-premises data sources, and web-based sources.
With Power BI, users can create interactive visualizations, reports, and dashboards to help them understand their data and communicate insights. The platform provides a range of data visualization tools, including charts, graphs, maps, and tables, as well as powerful data modeling capabilities.

The popular alternative to Power BI is Tableau [8]. Power BI is considered more user-friendly and easier to learn for beginners, while Tableau has a steeper learning curve but offers more advanced features and customization options. Also, Power BI has a stronger connection to Microsoft data sources, while Tableau has broader data source connectivity, including native connectors for cloud databases and web data connectors.

3 OBJECTIVES

The main objective, as explained in the introduction, is to investigate what are the reasons or the conditions under which these delays or losses occur.
At the same time, the aim of my project is to test the concept of integrating a data transformation tool with a data warehouse for subsequent visualization.

The proposal to achieve the objective is to build a Data Engineering plan, using an ETL in order to transform the company's current data, which is a simple spreadsheet, into a dimensional model, stored in a data warehouse, where any series of queries can be made more efficiently and extract all the information the company needs [9].

Once the model is generated, it is proposed to create an interactive dashboard. In this, the most significant graphics will be created that provide information on the state of the packages and thus, thanks to the visualizations, the company will be able to know the conditions under which package losses occur and also the reasons why packages are delayed.

This dashboard will also contain forecasting in order to predict at which times losses and delays are most likely to occur in the immediate future.

Finally, a model will be created using the python sklearn library to predict the status of a package (delayed, lost or shipped) and a study will be made of the reasons that cause this status to change.

The data architecture described above is represented as follows (Figure 1).

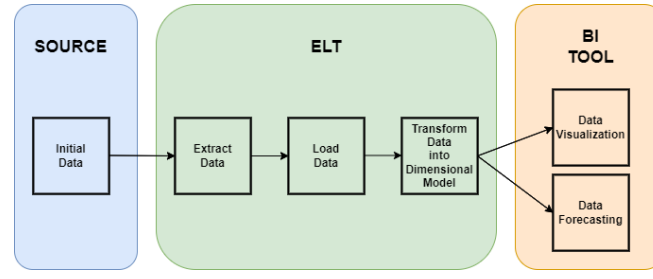


Fig 1. Data Architecture

For this process I will be using dbt (Data Build Tool) as an ETL, Snowflake as a data warehouse and Power BI as a visualization tool.

Using dbt, I can write data transformations to convert the single table into a star schema model, which includes a central fact table and multiple dimension tables. The data transformations are run within Snowflake, taking advantage of its performance and scalability, and the resulting star schema is stored in Snowflake for analysis and visualizations.
Finally, using Power BI, I can build the interactive dashboard with also some forecasting charts.

In order to reach the final product, the structure shown in the following image will be built (Figure 2).

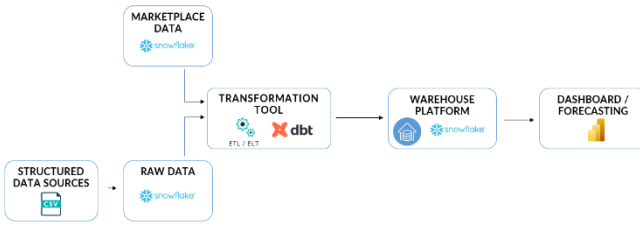


Fig 2. Solution proposal

Some specific requirements for the model are delivery dates so the model should include a date dimension table to allow the company to track delivery dates; location so the model should also include a geography dimension table to analyze shipment volumes by country or region; finally, the model should be flexible enough to incorporate new data sources and external information to enrich the model.

4 Methodology & Planning

The methodology for this project has been determined based on the fact that it is a cyclic process where the different tasks will be repeated to increase the complexity of the model and to incorporate improvements.

This is an agile methodology based on sprints. Prior to the cycle, learning tasks of both dbt and snowflake and also their joint operation will be completed.

Once they are established, the cycle will begin and will be repeated throughout the months every 2 or 3 weeks depending on the difficulty associated with every sprint.
The tasks that make up the cycle are:

- Model design
- Data transformation, using dbt
- Load the generated tables into snowflake
- Creation of the report/forecast using the model

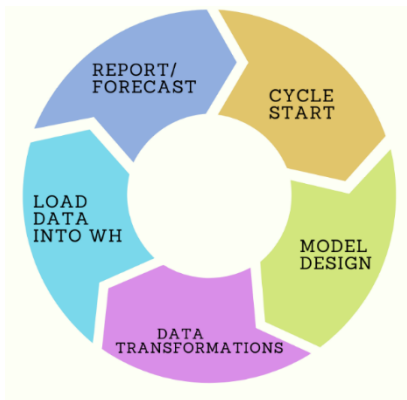


Fig 3. Project Planning

For communication with the tutor and for the correct monitoring of the project, the communication tools Outlook and Microsoft Teams will be used.

5 Development

As mentioned in the previous section, in order to achieve the proposed objective, a series of tasks will be carried out that will be repeated cyclically in order to increase the complexity of the solution.

Below are the tasks that have been done so far and also the next steps are shown.

5.1 Model Design

After the preliminary tasks of learning how dbt and Snowflake work, the first step is to design the dimensional model in order to use it as a guide for the transformations that will be made later.

As mentioned in the introduction, initially there is a single table with all the shipment information. That is why it is necessary to see what are the dimensions that can make up the final model.
That is why the initial dimensional model is as follows:

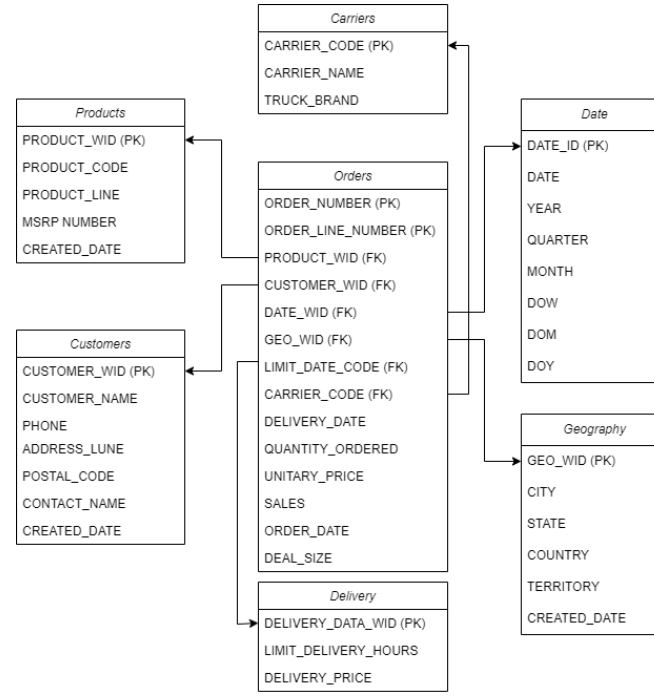


Fig 4. Dimensional model design

As can be seen in Figure 4, the model consists of a single central fact table, together with the different dimensions:

- Products: which contains information about the different products, such as the product_code.
- Customers: contains customer information such as customer_name or phone.

- Date: Contains date information, in order to facilitate a future visualization of the data.

- Geography: contains information about the city, along with the state, country, and territory to which it belongs.

- Delivery: contains information about the type of delivery that has been requested.
  As can be seen, this dimension was not present initially. It has been incorporated in order to differentiate the different types of order delivery and thus be able to draw more precise conclusions.

- Carriers: it contains information about the carriers and also about the brand of vehicle used.

5.2 DATA TRANSFORMATIONS

Once the dimensional model is designed, the transformations can begin in order to generate the model established from the initial table.
As discussed above, the transformations will be done in the ELT tool, dbt.

In order to generate the model, three essential steps will be considered: seeds, staging, and tables.
Once all the model tables have been generated, a series of tests will be generated to check and guarantee the quality of the final data and the relationship between the model tables.

5.2.1 SEED

The step before starting the data transformations is the creation of a seed. Seeds, in the context of dbt, are CSV files that are part of the dbt project and can be loaded into the data warehouse, specifically in our case, Snowflake.

A seed acts as the initial dataset or starting point for the data transformation process. It serves as a foundational source of data that can be leveraged by dbt to build upon and apply transformations. In our scenario, we have a single table stored in CSV format, which can be conveniently configured as a seed in dbt.

In order to do this, you just need to copy the data in CSV form to the seed directory of the dbt project.

5.2.2 STAGING

Staging [10] is a critical step in the data transformation process, where raw data is transformed into a format that is optimized for analysis.
In dbt, staging plays a key role in ensuring that data is clean, consistent, and ready for analysis. dbt provides a flexible and scalable platform for defining, executing, and managing data transformations in a repeatable and reliable manner.

In our case, we will use staging to make the relevant transformations and prepare the data for the creation of the model tables.

```
SUBSTR(orderdate, 1, LENGTH(orderdate) - 5) as DATE_COLUMN,
TO_DATE(DATE_COLUMN) as DATE_COLUMN_,
TO_CHAR(DATE_COLUMN_, 'DDMMYYYY') as DATE_WID,
status_d as STATUS,
productline as PRODUCT_LINE,
msrp as MSRP,
productcode as PRODUCT_CODE,
customername as CUSTOMER_NAME,
phone as PHONE,
addressline1 as ADDRESS_LINE,
city as CITY,
state as STATE,
postalcode as POSTAL_CODE,
country as COUNTRY,
territory as TERRITORY,
concat(contactfirstname,' ',contactlastname) as CONTACT_NAME,
```

Fig 5. Staging layer

In the previous figure (Fig 5), the staging layer I created in dbt is shown. As previously mentioned, it is at this moment that the necessary transformations are made prior to the creation of the final tables of the model.

Apart from changing the name of the columns so that they all have the same style, other transformations are made as may be the case of the 'date' column, which is changed in the format by subsequent steps and also in the case of the first and second name of the customers, which are concatenated.

5.2.3 MODEL TABLES

Once we have the staging, we can now create the tables for the dimensional model.
As mentioned in the model design section, this consists of a central fact table and several dimension tables.
That is why the creation of the two types of tables will be explained separately below.

5.2.3.1 DIMENSIONAL TABLES

To create the dimensional tables, we will have to make SQL queries to the staging level with the relevant information.

Following the business case, we need to create the customers' dimension table. In order to do this, we will make a SQL query as seen in Figure 6.

```
SELECT
    seq_customers.nextval as CUSTOMER_WID,
    A.CUSTOMER_NAME,
    A.PHONE,
    A.ADDRESS_LINE,
    A.POSTAL_CODE,
    A.CONTACT_NAME,
    current_timestamp() as CREATED_DATE
   FROM
(select


    CUSTOMER_NAME,
    PHONE,
    ADDRESS_LINE,
    POSTAL_CODE,
    CONTACT_NAME

from {{ref('stg_orders')}}

group by CUSTOMER_NAME, PHONE, ADDRESS_LINE, POSTAL_CODE, CONTACT_NAME) A
```

Fig 6. Dimensional table query

As you can see, we are using snowflake sequences to create the WIDs of the dimension tables in order to connect the tables within the data warehouse.

5.2.3.2 FACT TABLE

Once we have all the dimension tables, the fact table must be configured, which we will create with an SQL query as seen in figure 7.

In addition, the relevant joins must be created in order to connect the fact table with the different dimension tables. A left join, also known as a left outer join, is a type of join operation that combines rows from two tables based on a related column and includes all the rows from the left table regardless of whether there is a match in the right table.

```
select

    ORDER_NUMBER,
    ORDER_LINE_NUMBER,
    PROD.PRODUCT_WID,
    CUST.CUSTOMER_WID,
    GEO.GEO_WID,
    DATE_WID,
    QUANTITY_ORDERED,
    PRICE_EACH AS "UNITARY_PRICE",
    SALES,
    STATUS,
    ORDER_DATE,
    DELIVERY_DATE,
    DEAL_SIZE,
    LIMIT_DATE_CODE,
    CARRIER_CODE

from {{ref('staging_orders')}} ORD


LEFT JOIN dim_products PROD ON ORD.PRODUCT_CODE = PROD.PRODUCT_CODE
LEFT JOIN dim_customers CUST ON ORD.CUSTOMER_NAME = CUST.CUSTOMER_NAME
LEFT JOIN dim_geography GEO ON ORD.CITY = GEO.CITY AND ORD.COUNTRY = GEO.COUNTRY
LEFT JOIN dim_delivery DEL on ORD.LIMIT_DATE_CODE = DEL.LIMIT_CODE
```

Fig 7. Fact table query

Once we have all the star model tables created, we will need to run the 'dbt run' command in order to load the tables into the data warehouse.

5.2.4 DATA LINAGE

After executing the mentioned steps, the data lineage that is generated in the same dbt tool, is as follows:

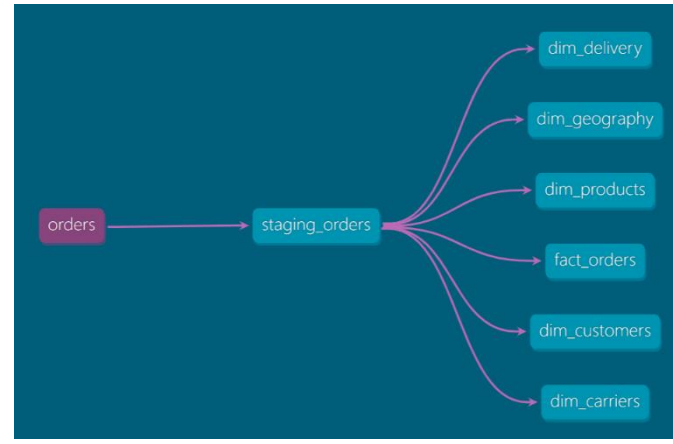


Fig 8. Data transformations lineage

As can be seen in Figure 8, the steps mentioned above are followed where the data flow starts with the seed called 'orders' and ends with the creation of the different tables, passing through the staging layer called 'staging_orders'.

Finally, it should be noted that all the transformations are done using SQL together with a proprietary language of dbt called jinja [11].
All dbt project code, including transformations, is stored in a GitHub repository that can be accessed via the following link:
https://github.com/MarcSedo01/ELT_PIPELINES.git

5.3 LOAD DATA INTO WAREHOUSE

Once we have the different tables of the model created with the transformation tool, dbt, the next task is to load the data into the data warehouse, in this case, Snowflake.

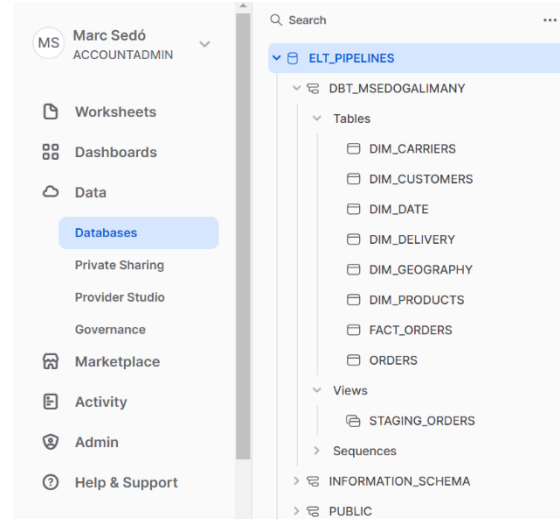


Fig 9. Data stored in Snowflake

As you can see in figure 9, the staging is saved as a view, so it is better to use materialized views for the staging layer.

Materialized views are database objects that store the results of a pre-defined query or a set of queries. They are used to improve query performance by pre-computing and caching the results of complex or frequently executed queries.

There are a few benefits to using materialized views for your staging layer:

- Performance: Materialized views are pre-computed and stored in a table, so they can be queried more efficiently than regular views.
- Simplicity: Materialized views allow you to treat your staging layer like a regular table, which can make your dbt models and transformations simpler and easier to understand.
- Flexibility: Materialized views can be indexed and queried just like regular tables, so you can use them in a variety of contexts and scenarios.

Finally, it should be noted that loading data into the data warehouse is a necessary step prior to the creation of reports/forecasting, as it is from the warehouse where the data will be taken for the visualizations.

5.4 REPORT & FORECASTING

Once we have the model loaded into the data warehouse, Snowflake, we can move on to the last step of planning, which refers to the creation of visualizations of the data and the forecasting of the most significant KPIs.

For this part, I am using Power BI, which is a business intelligence and data visualization tool developed by Microsoft that allows users to connect to various data sources, transform and shape the data, and create interactive reports, dashboards, and visualizations.
Since we have the model in Snowflake, first the connection to the warehouse will have to be created in order to import the data into Power BI and thus be able to use it in the dashboards.
Once we have the data imported, we need to check that the model has been imported correctly. To do this, we can look in the 'Model' part of Power BI and so we can see our model.

As you can see in figure 10, we check that the model has been imported correctly. We can check this by seeing how the relationships between the fact table and the dimension tables are correct. We can also see how the relationships are 'one to many', which means that a value in the dimension table corresponds to many values in the fact table.

The choice of a one-to-many relationship between a fact table and dimension table in data modeling is driven by several factors.

Firstly, it allows for the appropriate granularity and aggregation of data. Fact tables contain quantitative measurements, or facts, captured at a detailed level, while dimension tables provide descriptive context. By linking the fact table to multiple dimensions, analysts can easily aggregate facts by different dimensions, facilitating comprehensive and insightful data analysis.

Secondly, this design simplifies analysis by enabling users to explore data from different perspectives. With multiple dimensions, users can drill down, slice, and dice data, gaining valuable insights into various aspects of the business.

Finally, a one-to-many relationship supports scalability and extensibility. As fact tables grow over time, new dimensions can be added without modifying the existing structure, ensuring the data model can adapt to evolving analytical needs.

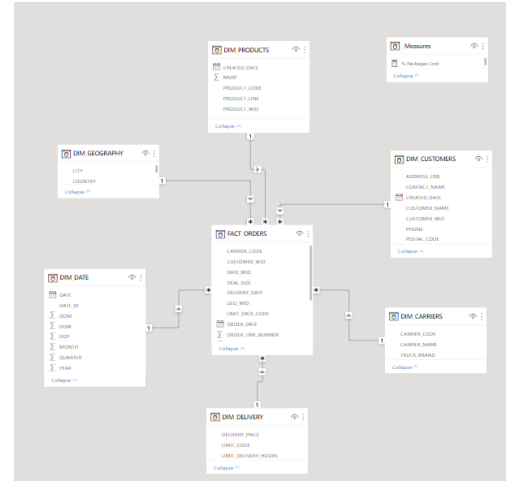


Fig 10. Dimensional Model

Once we have checked that we have the model set up correctly, we can generate the graphs and predictions.

Following our use case, where we want to know under what conditions packages are lost or delayed, a dashboard is first generated that contains four different graphics (Appendix 1).

It should be mentioned beforehand that this dashboard can be filtered for items sent, lost, or delayed. Also, since it is a Power BI dashboard, when you click on any value in a graphic, it will filter the entire dashboard considering the clicked value.

As for the graphics, first of all, there is a pie chart showing the total percentage of shipped, delayed, and lost packages to have a global view of the situation.

Secondly, a table with all the existing orders is displayed, together with a series of information relating to the order such as the carrier's name, product line, among others.
This table helps us when we filter lost or delayed packages, which are those orders that have caused problems.

Thirdly, a bar graph is shown showing all the packages transported by the different carriers.
This graphic is used to see, when the dashboard is filtered, which carriers have lost the most packages or have been delayed the most.

Finally, a treemap has been created to see, on the one hand, the product type that is lost the most, and on the other hand, the customers who suffer the most delays.

Apart from the graphics discussed above, two maps have been created. They give us information about lost packages around the world, and also in which countries package delays occur (Appendix 2).

Once we have seen the different graphs, let's now move on to the forecast part. Following the use case, the information we are interested in predicting are lost and delayed packages.

It is for this reason that, using the forecast functionality of Power BI, you can create predictions of these two KPIs (Appendix 3).

Finally, in order to see what the reasons are why the status of a package can change between shipped, delayed, and lost, I proposed a model using the sklearn python library [13].

For the model, the status column of the fact table has been used as target and the numerical variables of the same fact table have been used as parameters.

Since there are a total of 2000 records in the dimensional model, 80% have been used for training and the other 20% for testing. The result obtained was 0.80 accuracy, which is not very high but sufficient to draw some conclusions.

One of the main points for which the decision tree has been used is that it can effectively capture non-linear relationships between numeric features and a categorical target. Unlike linear models, decision trees are capable of modeling complex interactions and nonlinear patterns without imposing any assumptions on the relationship between the features and the target.

Once the model has been trained, I have studied the importance of the different parameters to see which ones have the most influence on the model's result.

This is why those parameters with a higher value of importance are the ones that influence the status of packages the most and can be the reasons that produce loss or delays of packages.

6 RESULTS

Once we have finished the development part, it is time to extract that most important information as results.

In our use case, we wanted to know the conditions under which package losses or delays occurred in order to avoid them.

In order to find out, we have to pay attention to the graphics generated in the last part of the development.
That is why below we'll see what information each one gives us and if it really helps us with the initial problem.

First of all, we have the dashboard made up of four different graphics. As discussed during development, the dashboard can be filtered considering whether packages have been sent, lost, or delayed. Taking advantage of this Power BI functionality, we filter the dashboard by delayed packages and then by lost packages.

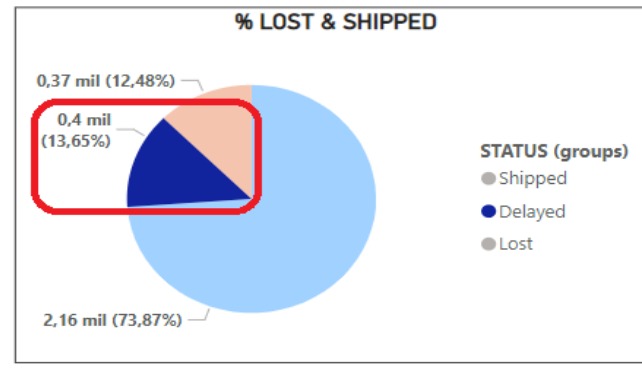


Fig 11. Delayed packages %

In the previous image (Figure 11) you can see how the delayed packages are part of 13.65% of the total packages.

Even so, we can see in Figure 12, how all the carriers have a slight percentage of delayed packages, which tells us that the problem does not lie with the carriers but with some external fact, which will be tried to extract with the rest of the graphs.

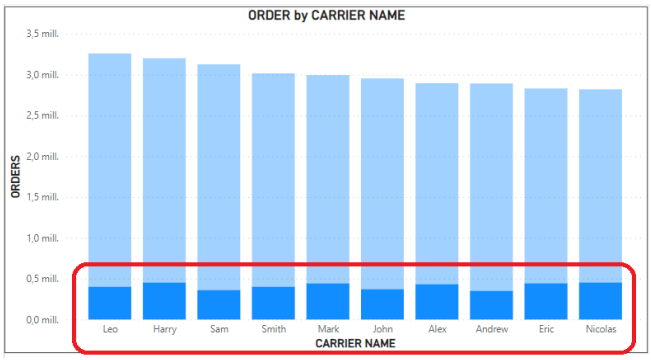


Fig 12. Delayed packages by Carrier

To finish extracting all information from the initial dashboard, let's now see the information on lost packages.

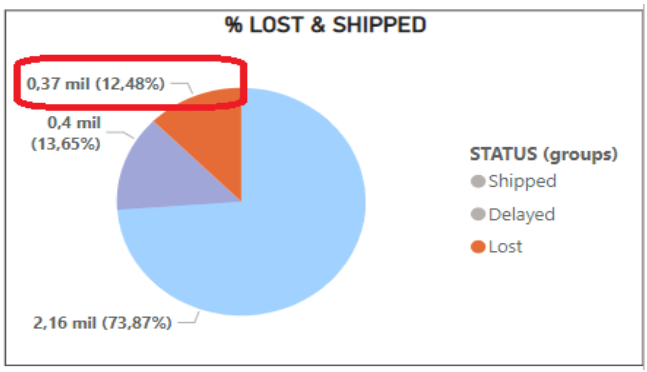


Fig 13. Lost packages %

We can see in the previous image (Figure 13) how the percentage of lost packages is 12.48% of the total.

On the other hand, looking at the following image (Figure 14), we can see how there are three carriers that are to blame for all lost packages.

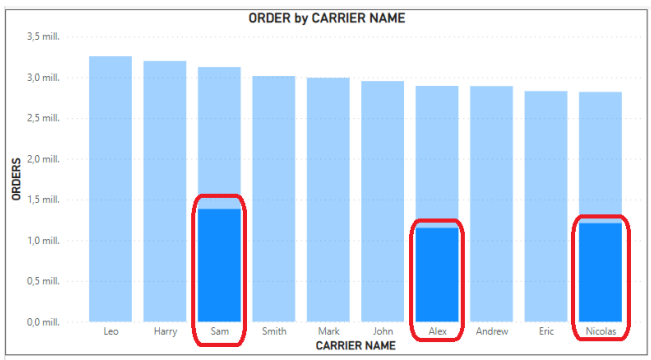


Fig 14. Lost packages by Carrier

Before drawing conclusions, let's look at the maps, also generated with Power BI, which show the places where packages have been lost or delayed.

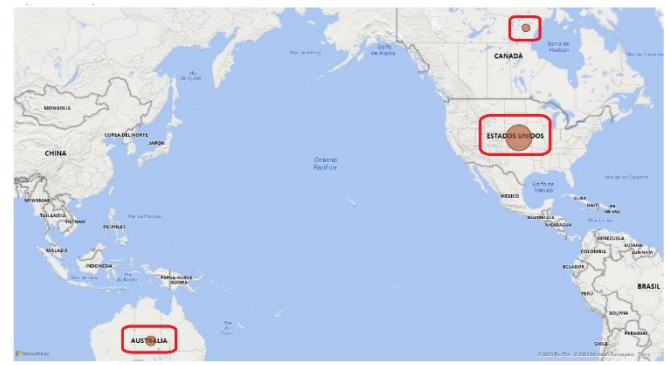


Fig 15. Delayed packages Map

In the previous map (Figure 15) we can see how there are only three countries where packages are lost, which are: USA, Canada, and Australia.

As for the map that gives us information about lost packages, we can see there are some countries such as the United States and the central part of Europe where many packages are lost.

However, they are not the only ones, as there are many other countries, as shown on the map (Figure 16). This is why the blame for the loss of packages lies with the three carriers involved, since in this case it does not involve the affected country as was the case with delayed packages.

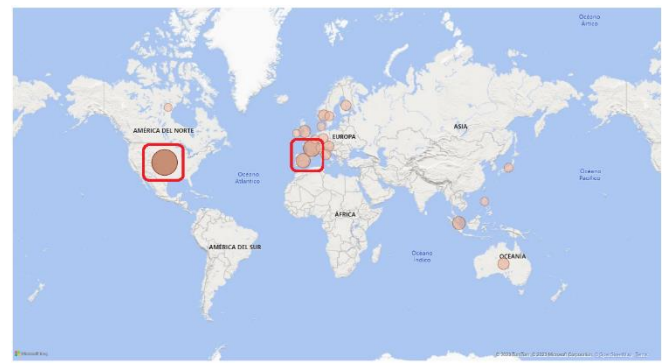
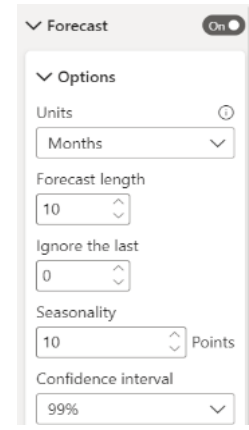


Fig 16. Lost packages Map

Once the results have been explained, let's now move on to the forecast part. Below are the short/medium term predictions of the number of lost and delayed packages separately.



When using Power BI's forecast functionality to create the predictions, different parameters have been considered [12]:

First of all, we have the Forecast Units which is the most important setting of the forecast because all other parameters will be based in these units.

Then, there is the Forecast length is the number of future periods that you want to forecast. In this case, I want the following 10 month.

After, with the Seasonality the forecast model will try and adjust for these highs and lows but needs to know how many periods or months to look back for a full sales cycle.

Finally, Power BI will generate a confidence band and will not give you a specific number for a forecast. Because it does not know the future there are many different outcomes that could occur. Lower confidence intervals will create a narrower band and higher confidence intervals will create a wider band.

It's saying that based on historic data, it's 99% sure that future sales will fall within a specific range.

So first, in Figure 17 you can see the prediction of delayed packages. You can see how there is a peak in the graph where it is predicted that in July there will be an increase in the delay of packages.

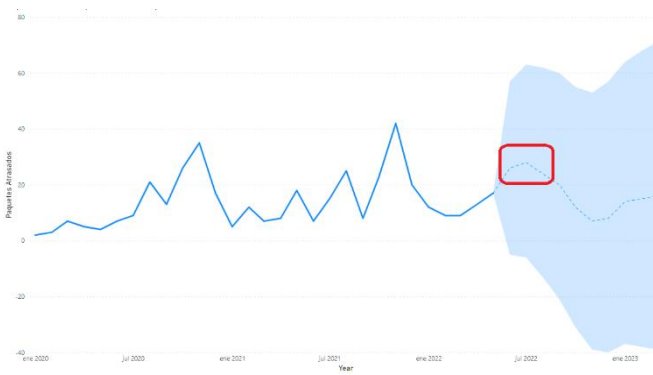


Fig 17. Delayed Packages Prediction

Fig 18. Lost Packages Prediction

Finally, in the above image (Figure 18), you can see the lost packages' prediction using the same parameters, commented before.

As you can see, there is a peak in the graph where it is predicted that in July there will be an increase in the delay of packages, and then in September a smaller one with less loses than the other peak but more than normal.

Regarding the prediction using the decision tree classifier, based on this, it has been studied which are the characteristics that most influence the final status and the result obtained are as follows (Fig 19):

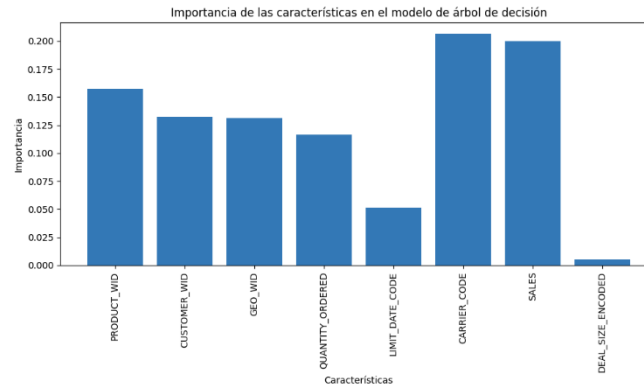


Fig 19. Characteristics' influence

Thanks to this proposal, it is possible to see which are the parameters, which correspond to characteristics of the fact table, which influence more on the status of the packages.
As can be seen in the figure 19, the influence value of carrier_code is the highest when determining the status of a package, while in the case of deal_size the influence is almost zero.

These characteristics are the ones that must be considered when generating the visualizations that have been discussed previously, as may be in the case of carriers, since these are the reasons why packages are lost or delayed .

It should be mentioned that this is a proposal, which is to say, a first approach to what could be a model in order to determine the importance of the different characteristics.

7 CONCLUSIONS

During this work I managed to extract and load data into the data warehouse, perform transformations and visualize the data, so the integration of a data transformation tool, dbt, with a data warehouse like Snowflake was achieved.

Apart from that, conclusions can also be drawn from the tools used in this project.

One of the key benefits of integrating dbt and Snowflake is the ability to leverage Snowflake's scalability and performance to run data transformations at scale. Snowflake's unique architecture allows organizations to store and analyze petabyte-scale data in a way that is both fast and cost-effective. With dbt, data transformations can be written and executed within Snowflake, taking advantage of its performance and scalability.

In addition to these benefits, dbt and Snowflake also offer strong security and compliance features, making it possible for organizations to manage sensitive data with confidence. Snowflake provides full isolation of workloads and a highly secure architecture, while dbt provides version control and access control features that allow data teams to manage access to data transformations.

Finally, both snowflake and dbt are cloud-based platforms, so no prior installation is necessary to use the two tools together. It should also be added that in the case of GitHub no installation is necessary either.

About the use case, the following conclusions can be drawn.
The problem of package loss is not caused by any specific country, but by three of the company's carriers that lose 100% of the packages. In addition, thanks to the forecast it has been predicted that the months of July and September will produce more lost packages than usual.

Regarding delayed packages, there are three countries where all package delays occur, which are the USA, Canada, and Australia. This fact may be strongly related to the fact that these three countries have many restrictions on import packages at customs and this may be the reason that delays the packages.
Finally, using the forecast, it has been predicted that during the month of July there will be more package delays than usual. Also note that during the month of November the package delay will be reduced.

On the other hand, after observing the importance of the characteristics of the model, we can conclude that the characteristic that has the most influence on the state of a package is carrier code followed by sales. We can also say that the deal size is the characteristic with the least influence on the final result.

Finally, comment on the next steps in the future that could be taken to expand the complexity of the project.
During this case, some of the functionalities of dbt have been used. However, there are some others with which the project could be enriched, such as the dbt macros.

dbt macros [11] are pre-written pieces of code that perform specific transformations on the data. They allow you to reuse commonly used operations and can be easily incorporated into the dbt models, saving time and effort. Macros can be used for a variety of purposes, such as data cleaning, data enrichment, and data validation.

By using dbt macros, you can improve the efficiency and consistency of data transformations, and you can also simplify the process of maintaining your data pipeline over time, and it's for this reason that it would be one of the main next steps to take.

BIBLIOGRAPHY

[1] The Parcel Shipping Boom Continues [Consulted: March 2023]. Available on the Internet: https://www.statista.com/chart/10922/parcel-shipping-volume-and-parcel-spend-in-selected-countries/

[2] What is dbt? [Consulted: March 2023]. Available on the Internet: https://docs.getdbt.com/docs/introduction

[3] Apache Airflow [Consulted: March 2023]. Available on the Internet: https://airflow.apache.org/

[4] Hands on essentials. [Consulted: March 2023]. Available on the Internet: https://www.snowflake.com/snowflake-essentials-training/

[5] Accelerating data teams with dbt cloud & snowflake. [Consulted: March 2023]. Available on the Internet: https://quickstarts.snowflake.com/guide/data_teams_with_dbt_cloud/#0

[6] Google Cloud BigQuery [Consulted: March 2023]. Available on the Internet: https://cloud.google.com/?hl=es

[7] Power BI [Consulted: March 2023]. Available on the Internet: https://powerbi.microsoft.com/

[8] Tableau [Consulted: March 2023]. Available on the Internet: https://www.tableau.com/

[9] Dimensional Modeling: In a Business Intelligence Environment (2006). [Consulted: March 2023]. Available on the Internet: .https://www.redbooks.ibm.com/redbooks/pdfs/sg247138.pdf

[10] What is a Data Staging Area? [Consulted: April 2023]. Available on the Internet: https://hevodata.com/learn/data-staging-area/

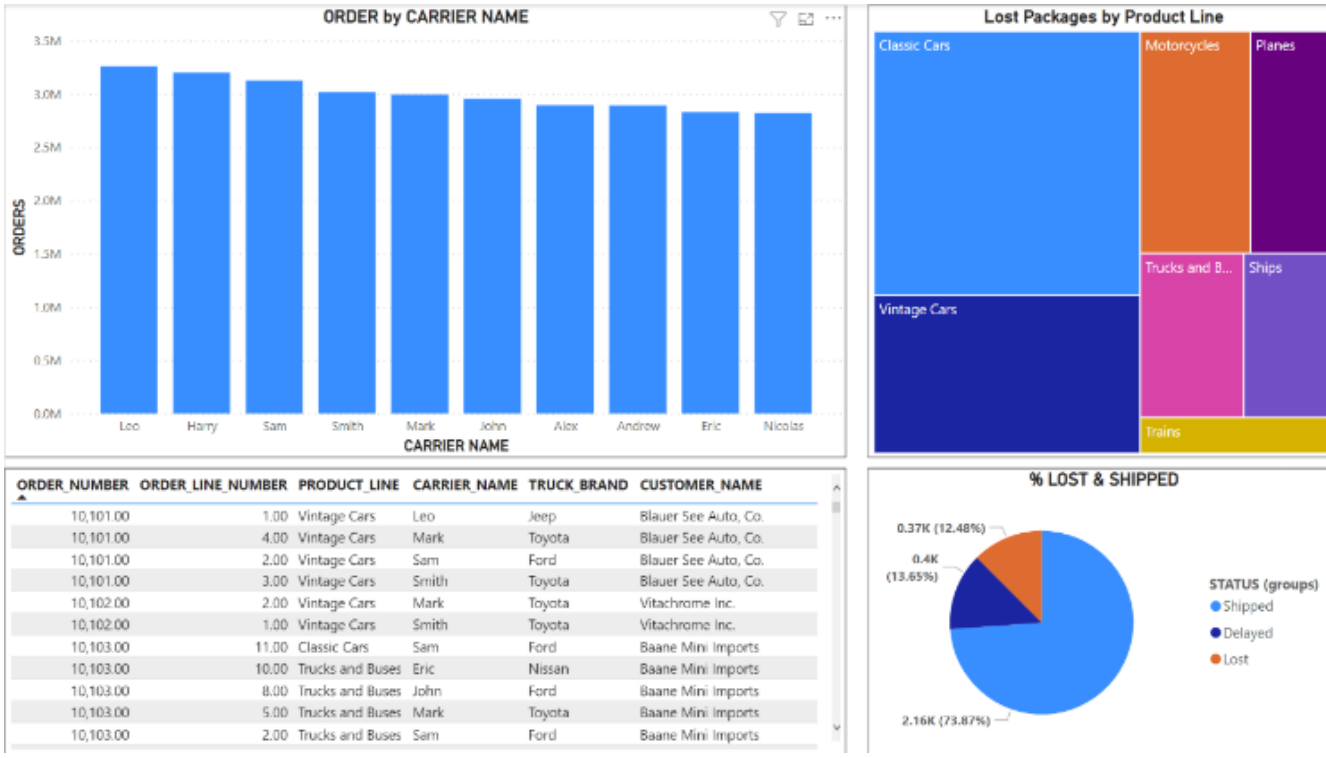
[11] Jinja and macros. [Consulted: April 2023]. Available on the Internet: https://docs.getdbt.com/docs/build/jinja-macros

[12] How to Forecast in Power BI? [Consulted: May 2023]. Available on the Internet: https://www.popautomation.com/post/how-to-forecast-in-power-bi

[13] Sklearn.tree.DecisionTreeClassifier [Consulted: June 2023]. Available on the Internet: https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

APPENDIX

Appendix 1. Power BI Dashboard



ORDER_NUMBER	ORDER_LINE_NUMBER	PRODUCT_LINE	CARRIER_NAME	TRUCK_BRAND	CUSTOMER_NAME
10,101.00	1.00	Vintage Cars	Leo	Jeep	Blauer See Auto, Co.
10,101.00	4.00	Vintage Cars	Mark	Toyota	Blauer See Auto, Co.
10,101.00	2.00	Vintage Cars	Sam	Ford	Blauer See Auto, Co.
10,101.00	3.00	Vintage Cars	Smith	Toyota	Blauer See Auto, Co.
10,102.00	2.00	Vintage Cars	Mark	Toyota	Vitachrome Inc.
10,102.00	1.00	Vintage Cars	Smith	Toyota	Vitachrome Inc.
10,103.00	11.00	Classic Cars	Sam	Ford	Baane Mini Imports
10,103.00	10.00	Trucks and Buses	Eric	Nissan	Baane Mini Imports
10,103.00	8.00	Trucks and Buses	John	Ford	Baane Mini Imports
10,103.00	5.00	Trucks and Buses	Mark	Toyota	Baane Mini Imports
10,103.00	2.00	Trucks and Buses	Sam	Ford	Baane Mini Imports

Appendix 2. Power BI Map

Appendix 3. Power BI Forecast