POLITECNICO DI TORINO

Master's Degree in Data Science & Engineering

Master's Degree Thesis

Design and implementation of a Business Intelligence pipeline for Poste Italiane

Supervisor

prof. Eliana PASTOR

Candidate

Alessio VACCA

2022/2023

Summary

The growing amount of data generated by companies in today's digital age has led to an increased demand for tools and techniques that can help organizations make informed decisions based on data-driven insights. For this reason, more and more companies are deciding to rely on consulting firms to help them create and manage systems that can measure and understand business performance and make them available to their users. In this context, Business Intelligence (BI) and data analytics have become crucial for organizations that want to gain a competitive advantage in their industry, enabling them to extract valuable insights from data and use them to optimize their operations, improve customer satisfaction, and drive growth. This thesis project arises from the collaboration between Advant s.r.l and Poste Italiane S.p.A. The first is a consulting IT company, mostly dedicated to data analysis and Business Intelligence, the latter is a well-known Italian public company, which is responsible for managing postal, banking, financial, logistics, and telecommunication services.

In that respect, I have been involved in the design and implementation of a BI system that supports decision-making and strategic and monitoring activities for its users within the sales force departments, in order to provide evidence to the salespeople operating in the territory. In particular, the project involved the analysis of the revenues and volumes from auxiliary services related to parcel products sold. The entire development pipeline is based on cloud technologies from Microsoft's Azure suite. It includes a back-end part consisting of a data mart model built on top of the provided data source, after an initial phase of functional and user requirements analysis. Subsequently, an ad hoc ETL process was designed to automatically feed the target data storage, so as to update the dashboards containing the required reports and make them available to the end users. My contribution was then integrated into a larger project, with which it shares some technical features.

Acknowledgements

I want to express my gratitude to my company for the opportunity they have offered me, their warm reception, and support. I also extend my gratefulness to my academic supervisor, Eliana Pastor, for her guidance. My family deserves a huge thank you for providing all the support and encouragement that enabled me to reach this milestone. Lastly, I would like to extend my thanks to all the remarkable people I have had the privilege to meet and share these great years with.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the past decades, there has been an explosion in the amount of data generated and consumed by individuals and organizations, due to the widespread adoption of digital technologies. This has led to an increased demand for tools and techniques that can assist organizations in making informed data-driven decisions. For this reason, more and more companies are turning to consultancy firms to help them create and manage systems that can measure and understand business performance and, ultimately, make it available to users.
In this context, the landscape of Business Intelligence (BI) and data analytics systems have seen significant growth, becoming crucial for organizations to gain a competitive advantage in their industry by enabling them to extract precious insights from their data and use them to optimize their operational efficiency, improve customer satisfaction, and drive growth. In fact, more and more companies offer a range of solutions to meet the diverse needs of their customers and users, including data integration, data visualization, analytics, and reporting.

Although the effectiveness of using business intelligence as a means of improving business performance is something of a factual matter, the same cannot be said for the methodology and approach to creating a BI system.
Indeed, BI is not a one-time project or a fixed set of tools and technologies; instead, it is continually evolving in tandem with the changing needs of businesses and the latest advancements in technology. In this respect, data is moving to the cloud faster than ever before due to its scalability and cost-effectiveness. This is prompting organizations to rethink their data strategy and switch from on-premise

solutions to cloud-based environments, in order to keep up with market needs. Among the various companies that have made BI an indispensable decision-making fulcrum is that of Poste Italiane, which increasingly relies on data analysis and business intelligence to enhance its investments and make business processes more efficient.
Among the diverse applications, Big Data analysis allows, for example, to profile customers so as to improve and increase sales performance. Precisely with regard to the latter application, the work carried out within Advant S.r.l. and described in the following paper focuses on the implementation of a BI system that supports the decision-making, strategic and monitoring activities of the sales force by business area, and sales portfolios in the various territorial areas from the MIPA[1] products. The ultimate goal then is to enable managers to achieve the results set by management and to ensure management is oriented toward improvement continuous.
The system assembled leverages some of the Microsoft products, in particular SQL Server Database, SQL Server Analysis Services, Azure Data Factory, and Power BI, and it is fully automated and cloud-based. For organizations invested in Microsoft products, Azure offers seamless integration, enabling hybrid cloud deployments and smooth migration. It is a preferred choice for businesses with on-premises infrastructure, offering robust hybrid cloud computing services capabilities. The data sources provided are used to feed the data warehouse, designed to optimally support the required analysis. Afterward, the data mart is passed to an instance of analysis services in order to create the tabular model with the required measures and apply the row-level security.
The tabular model itself is therefore provided to the reporting software to produce the customer's expected results, and eventually make it available to the end users, who benefit the latter via an online dashboard. The entire pipeline has been orchestrated through an ad-hoc ETL process able to manage and guarantee consistency, scalability, robustness, and security all through the activities, from the data integration to the final reports.
The solution was developed in the various development environments, and then deployed in the test and production environments. The versioning of the system release follows the practice of continuous integration and continuous delivery (CI/CD). This is particularly important because evolutions and updates can be introduced in the future to improve and expand the current system.
This thesis aims to demonstrate the system's implementation by providing a detailed description of the processes and tools used to achieve the desired outcome while

[1] It is a business division in Poste Italiane company and stands for "Medium-sized Business and Public Administration"

adhering to the specified constraints.

Thesis Content

This thesis work consists of six chapters. It starts with a review of the main concepts of business intelligence and cloud analytics to describe the entire design process of the data pipeline, which led from the gathering of requirements, the creation of the data warehouse, the ETL orchestration, and the subsequent reporting.

- The **second chapter** consists of a review of the Business Intelligence fundamentals, emphasizing its role and necessity in today's business as well as the several components of a typical BI process, such as *Data Warehouses*, *Extract, Transformation* & *Load* process, and reporting. Special attention is also given to cloud-based services since the entire solution is fully deployed in the cloud, highlighting their advantages.

- The **third chapter** introduces the case study that is the subject of this thesis, describing the two companies involved in the project: Poste Italiane, the client company, and Advant S.R.L, the consulting firm hired to carry out the project. In addition, the objective of the study and the project context will be explained.

- The **fourth chapter** describes the requirements outlined by the client and the subsequent design of the new Data Marts to efficiently support the required analyses.

- The **fifth chapter** provides an overview of the developed system architecture, focusing on each stage of the pipeline and the software used to implement it. This includes a description of how the data were extracted and loaded into the staging area; the implementation of the data warehouse area; and how the entire solution was orchestrated and automated until the required results were achieved.

- The **sixth chapter** shows the result of the whole process, namely the dashboards accessible to end users.

Chapter 2

The Business Intelligence Fundamentals

2.1 Business Intelligence

As data analysis has become one of the most valuable activities today, it is more important than ever for companies to understand how to extract every drop of value from the myriad of digital information available at their fingertips. To realize the full potential of their data, companies need to invest in the right tools and processes to ensure that everything goes as planned. By gaining the ability to understand which data sets are relevant to particular goals, strategies, and initiatives, they can identify trends or patterns that help them make significant improvements in several key areas of the organization. This concept is known as Business Intelligence (BI) and is becoming more prevalent every year in all sectors.

BI therefore represents a set of tools and processes that enable companies to manage data from acquisition to reporting, providing a comprehensive (real-time), historical, and predictive view of structured data concerning all departments in an organization. Thus, companies no longer have to rely on decisions based on personal insights or experiences, but instead can make informed decisions based on their data, greatly increasing operational understanding and improving overall performance by fostering continuous growth. This is accomplished by simplifying access to advanced analytic technologies, including data warehouses, analytics, and visualization, to identify and analyze essential business data. BI is also valuable to businesses because of its self-service nature, enabling users with or without advanced analytical skills to explore data by providing automated capabilities for data collection and monitoring. However, successful BI implementation requires the understanding and execution of numerous processes.

2.1.1 Decision Support Systems

The great and rapid development of technology and information technology in the last decades has moved business activities in every sector toward a "smart," data-driven approach. This has increasingly shifted the focus to the customer and his or her needs and requirements. Therefore, information becomes a strategic resource to support decision-making and is considered a real business asset for the companies. It is in this context that Decision Support Systems (DSS) are placed, software designed to support semi-structured and unstructured decision-making, able to quickly deliver helpful information for decision-making processes in a variety of ways. This enables users to increase the effectiveness of their analyses and take full control of their activities, as it provides support to all those who need to make quick, strategic, and well-informed decisions in the face of problems that cannot be solved by operational research models. [1] The essential aspects of a DSS are:

- **Ease of Use**: should be designed to be accessible and usable by a wide range of users, regardless of their technical or professional knowledge. This means that the user interface should be intuitive and user-friendly and users should be able to easily access the data and functionality of the system.

- **Interactive Environment**: is characterized by an interactive environment that lets users explore data, perform analysis, and make decisions dynamically. This interactive environment can include data visualization tools, customizable control panels, and the ability to query data in real time. Interactivity is fundamental because it allows users to adapt to changing decision-making needs.

- **Decision Process Support**: The system should be designed to assist users at different stages of the decision-making process, from gathering information to generating alternatives and evaluating expected outcomes. This may include providing relevant data, automating complex calculations, and presenting alternative scenarios.

- **Data Analysis**: should be able to use analytical models and advanced algorithms to extract meaningful information from data. This includes the ability to perform predictive analysis, simulate scenarios, and identify trends or anomalies in the data. Effectiveness in using these models helps to improve the accuracy of predictions and recommendations provided by the system.

DSS have very different characteristics, but it is useful to classify them into two main types, **data-driven** and **model-driven**. The former is an outgrowth of the early proposals for decision support systems made in the late 1970s, ideal for making structured or semi-structured decisions, and their value depends on

the quality of the model used. The simplest solutions use spreadsheets for "what if" analysis, while more sophisticated models are used in fields such as operations research and artificial intelligence.
The latter enables users to extract data from large databases, often found in enterprise data warehouses, and synthesize it into a useful and easily interpretable format to help managers evaluate the performance of business processes and make strategic decisions. Online analytical processing (OLAP) and data mining can then be used to analyze data. Data mining provides insights into business data that cannot be obtained with OLAP, finding hidden patterns and relationships in large databases and inferring rules to predict future behavior

2.2 Data Warehouse

2.2.1 Why Data Warehouse?

Information is an increasingly valuable commodity, necessary to effectively plan and control business realities, it is the raw material that is processed by information systems. Unfortunately, the equation: *data = information* is not always correct: often, in fact, the availability of too much data makes it difficult, if not impossible, to extract the really important information. The phenomenon of data warehousing stems precisely from the enormous accumulation of data over the last decade and the pressing demand to actively use this data for purposes beyond routine, everyday processing.
A typical scenario is that of a large company with numerous subsidiaries, whose managers wish to quantify and evaluate the contribution of each of them to the company's overall business performance. Since elementary data on the activities performed are available in the company database, one possible approach is to formulate an ad hoc query that performs the necessary calculations on the data (usually SQL aggregations). However, this route is difficult to take, as it consumes unnecessary time and resources and, at the same time, does not always produce the desired result. Among other things, mixing this type of 'analytical' query with routine 'transactional' queries leads to inevitable slowdowns that make users of both categories dissatisfied.
The idea behind Data Warehousing is therefore to separate the On-Line Analytical Processing (**OLAP**) from the On-Line Transactional Processing (**OLTP**) by building a new information collector that integrates elementary data from a variety of sources, organises them in an appropriate form, and then makes them available for analysis and evaluation purposes for planning and decision-making. There are several areas where Data Warehousing technologies are being successfully deployed:

- **Commerce**: sales and claims analysis, shipment and inventory control, relationship care with customers, which is the area of interest of this work;
- **Manufacturing**: production cost control, supplier and order support;
- **Financial services**: risk and credit card analysis, fraud detection;
- **Transportation**: fleet management;
- **Telecommunications**: call flow and customer profile analysis;
- **Health care**: analysis of admissions and discharges, accounting by cost centers.

A common feature of all these fields is the need for archiving and querying tools that make it possible to easily and quickly obtain, from the enormous amount of data stored in databases or made available on the Internet, summary information that permits the evaluation of a phenomenon, the discovery of significant correlations and, ultimately, the acquisition of knowledge useful as decision support.

2.2.2 What is a Data Warehouse?

The most widely used definition of a data warehouse is given by U.S. computer scientist William H. Inmon, who states that *"A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management decision-making process."* [2]. The main function of the DW is decision support, so it must be designed specifically to answer business questions.
It is defined to be *subject-oriented* because it focuses on the concepts of interest to the company, such as customers, products, sales, and orders. Conversely, operational databases are organized around the transactions originating from the different applications of the business domain.
The emphasis on *integration* and consistency aspects are important since DW relies on multiple heterogeneous data sources: data extracted from the production environment, and then originally stored in enterprise databases or, even, from information systems outside the company. Of all these data the DW strives to return a uniform and consistent view.
Broadly speaking, it can be said that the construction of a data warehousing system does not involve the inclusion of new information but rather the reorganization of existing information. While operational data cover a time span that is usually rather limited since most transactions involve the most recent data, the DWH must allow for analyses that span the perspective of several years, thus maintaining the history so as to have a complete view of temporal evolution.
For this reason, the DW is updated at regular intervals, from the operational data,

and is continuously growing (*time-variant*). Precisely because of the fact that, in principle, data are never deleted data from the DW and that updates are typically performed when it is offline, it can basically be regarded as a read-only, nonvolatile database.

2.2.3 OLAP vs OLTP

The characteristic of a DWH to be a read-only database of unified heterogeneous data, together with the need of users to shrink the response time to analysis queries has important consequences at several levels.
Initially, this has a significant impact on the technologies adopted by specialized database management systems for data warehousing, eliminating the need for sophisticated transaction management techniques typical of operational applications. In addition, the fact that they operate on a read-only basis profoundly differentiates the logical design solutions for DWHs from those used for operational databases: perhaps the most obvious aspect in relational implementations is that the practice of table normalization is abandoned in favor of partial denormalization, aimed at improving performance.
Further and fundamental differences between operational and DW databases are related to the types of queries. For the former, queries perform transactions that typically read and write a small number of records from several tables linked by simple relationships. This type of processing is commonly referred to as OLTP.
In contrast, the type of processing for which DWs are born is called OLAP and is characterized by dynamic, multidimensional analysis that requires scanning a huge amount of records to compute a set of summary numerical data that quantifies the company's performance. It is important to note that while in OLTP systems the substantial core of the workload is frozen within the application programs, and only occasional queries are launched extemporaneous or extraordinary maintenance on the data, in a DW, interactivity, is an indispensable of the analysis sessions and causes the actual workload to vary continuously over time.
The peculiarities of OLAP queries mean that the data in the DW are normally represented in multidimensional form. The basic idea is to view data as points in a space whose dimensions correspond to as many possible dimensions of analysis; each point, representative of an event that occurred in the enterprise, is described by a set of measures of interest to decision-making. The main differences between operational databases and data warehouses are summarized in Table 2.1.

2.2.4 Architecture of a Data-warehouse

In 1999 Kelly[3] defined what are the indispensable architectural features for a Data Warehousing system. In particular, the key components are:

Table 2.1: OLAP vs OLTP

Features	**Operational databases**	**Data Warehouses**
Users	Thousands	Hundreds
Workload	Predefined Transactions	Ad Hoc Analysis Queries
Access to	Hundreds of records, reading and writing	Millions of records, mostly reading
Purpose	Depends on application	Decision support
Data	Elementary data, both numeric and alphanumeric	Summary data, mostly numeric
Data integration	By application	By subject
Quality	In terms of integrity	In terms of consistency
Temporal coverage	Current data	Historical data
Updates	Continuous	Periodic
Model	Normalized	Denormalized and multidimensional model
Optimization	For OLTP accesses on a fraction of the database	OLAP accesses on a large part of the database.
Developments	Cascading	Iterative

- **Separation**: analytical and transactional processing should be kept as much as possible separated.

- **Scalability**: the hardware and software architecture must be able to be easily scaled in the face of growth over time in the volumes of data to be managed and processed and the number of users to be satisfied.

- **Extensibility**: it must be possible to accommodate new applications and technologies without redesigning the system entirely.

- **Security**: access control is essential because of the strategic nature of the data stored in the system.

- **Administrability**: the complexity of the administration activity should not be excessive.

Data warehouse architectures

There are three types of architectures that can be adopted to implement a data warehouse process, depending on the number of data layers used: a *single-tier architecture*, a *two-tier architecture*, and a *three-tier architecture*. The goal of the single-tier architecture is the minimization of the stored data, achieved by eliminating redundancies. In practice the DW is virtual, that is, implemented as a multidimensional view of operational data generated by a special *middleware*. This type of solution has several weaknesses; it does not meet the requirement of separating OLAP analytical processing from OLTP transactional processing, rendering the queries on the operational data. It becomes impossible to express a higher level of historicization than the sources. For these reasons, it is actually almost never used in reality, only in contexts where the analysis needs are particularly limited and the volume of data to be examined is very large can be used.

In contrast, the two-tier architecture, so named to highlight the separation between the source layer and the DW layer itself, consists of four overall distinct layers, describing successive stages of the data flow. These layers are:

- **Source level**: the DW uses heterogeneous data sources extracted from the production environment and, therefore, originally stored in enterprise, relational, or legacy databases, or from information systems outside the enterprise.

- **Data staging**: data stored in sources need to be extracted, cleaned up (to remove inconsistencies and complete any missing parts), and integrated (to merge heterogeneous sources according to a common pattern). So-called ETL (Extraction, Transformation, and Loading) tools enable the integration of heterogeneous patterns, as well as the extraction, transformation, cleaning, validation, filtering, and loading of data from sources into the DW.

- **Warehouse Level**: information is collected in a single, logically centralized "container" (the DW itself). It can be directly accessed, but also used as a source for building data marts; the latter can be viewed as small local DWs that replicate, and possibly further synthesize, the portion of the primary DW of interest to a particular area.

- **Analysis Layer**: allows efficient and flexible consultation of integrated data for writing reports as well as for analysis and simulation activities.

This solution separates the operational database management system from the data warehouse and decision support system, as well as operational applications from business intelligence applications, so that business analysis does not interfere with and degrade the performance of operational applications.

The last type of architecture, the three-tier one, adds an additional layer to the two-tier architecture: the *reconciliation data layer*. It is also called the Operational Data Store (ODS) and materializes the operational data obtained downstream of the integration and cleansing process of the source data, resulting in integrated, consistent, correct, volatile, current, and detailed data. In fact, the DW is no longer fed directly from the sources, but rather from the reconciled data. The main advantage of this additional layer is that it creates a common data model and reference for the entire enterprise, while at the same time introducing a clear separation between the issues involved in extracting and integrating data from the sources and those inherent in feeding the DW. This represents the most widely used approach today and is precisely the one that will be employed within this project. The three components that give the name to this architecture are the Bottom Tier, the Middle Tier, and the Top Tier, as can be seen in figure 2.1.

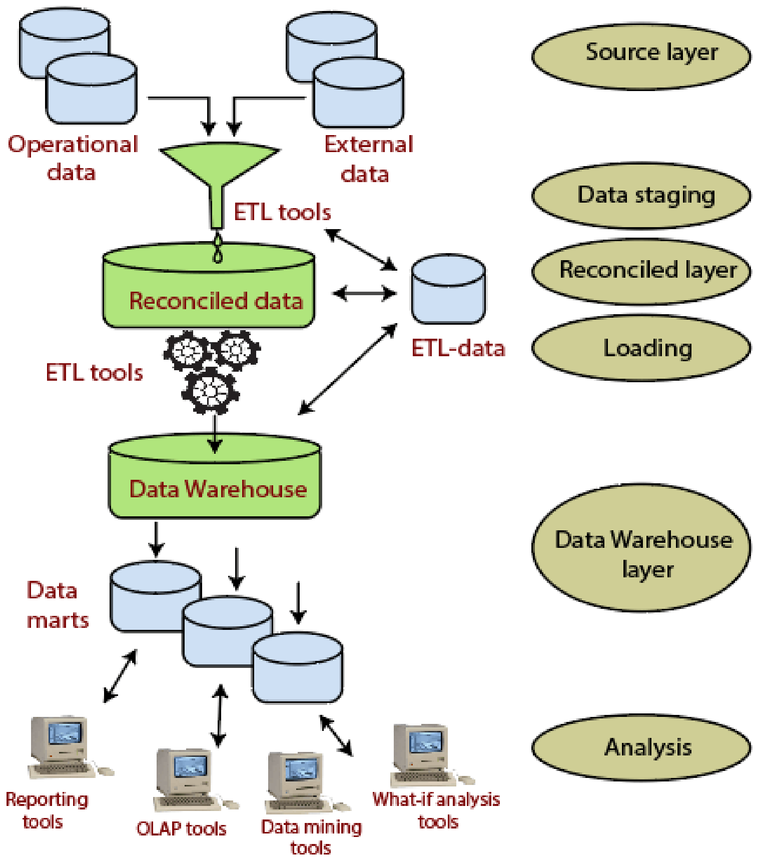


Figure 2.1: Three-tier Architecture of a Data Warehouse

The **bottom tier** consists of the Data Warehouse server, which is typically a Relational Database Management System (RDBMS). In this layer data from operational databases and external sources are extracted (using APIs[1] called gateways) exploiting Extract, Transform & Load (ETL) tools, cleaned, transformed, and loaded into the reconciled data layer and in the DW.

The **middle tier** consists of an OLAP server, useful for fast querying on top of the data warehouse's data. This can be implemented using either a Relational OLAP (ROLAP) model or a multidimensional OLAP (MOLAP) model.

The **top tier** contains front-end tools for displaying results provided by OLAP, as well as additional tools for data mining of the OLAP-generated data. This layer is the most important as it is the one in which the data are first presented to the end user, and its correctness depends on the accuracy and attention devoted to the previous layers.

2.3 Extract, Transformation & Load

2.3.1 What is ETL?

The Extract, Transform, and Load (ETL) process is fundamental to data integration in modern data engineering workflows. It involves extracting data from various sources, transforming it to fit the desired structure or format, that is optimally suited for analytical purposes, and loading it into a target repository (Data Warehouse) for analysis and reporting. By leveraging this process, the organization can systematically prepare individual raw data sets, thereby producing deeper and more actionable insights. Organizations today are faced with an ever-widening range of data, including structured and unstructured information from a variety of sources. These sources include:

- Customer data from systems such as online payment platforms and customer relationship management (CRM) systems.
- Operational and inventory data from supplier systems.
- Sensor data derived from the wide range of Internet of Things (IoT) devices.
- Marketing-related data including information gathered from social media platforms and customer feedback mechanisms.

[1] Application Programming Interfaces, are sets of rules and protocols that let different software applications communicate with each other, enabling them to exchange data and functionality seamlessly.

- Employee-related data extracted from internal human resource systems.

The use of ETL technology is an essential step in improving the business intelligence and data analysis process, making it more reliable, accurate, detailed, and efficient. It provides a **robust historical context** to data within an organization, allowing older data and newer information to be viewed together, offering a long-term view of the data. In fact, enterprises can integrate historical data, from older systems, with data from new platforms and applications.
It also guarantees a **unified view of data**, which is crucial for performing in-depth analysis and generating detailed reports. Managing numerous datasets requires time and coordination, with the risk of inefficiencies and delays. ETL combines databases and data in various forms into a single coherent and consistent view. This data integration process not only improves data quality, but also reduces the time required to move, categorize, and standardize the data. The result is a simplification in analyzing, visualizing, and understanding voluminous data sets.
It ensures **reliable data analysis** while meeting regulations and compliance standards. ETL tools can be integrated with data quality solutions to profile, verify, and cleanse data, ensuring its reliability.
ETL also **automates** repeatable data processing tasks, thereby optimizing the efficiency of analysis. Tasks like data migration can be automated and configured to integrate data changes on a periodic basis or at run time. As a result, data engineers can focus more on innovation and reduce time spent on tedious tasks such as data transfer and formatting. For most organizations using ETL, the process is automated, well-defined, continuous, and batch-based. Typically, it occurs during non-business hours, when traffic on the source systems and data warehouse is at a minimum.

This paragraph delves into each phase of the ETL process, of which a diagram architecture is shown in figure 2.2, highlighting their significance and importance in achieving accurate, consistent, and usable data for analysis and decision-making.

2.3.2 Data extraction

Data extraction represents the first phase of every ETL process, where data is gathered from various source systems, databases, or external data providers, setting the foundation for the subsequent transformations and analyses.
Typically, extracted data are temporarily stored in a special area called the staging area, which serves as a temporary storage space where data extracted from different sources are temporarily deposited before being further processed and loaded into the target system. This step is important because it clearly separates data acquisition from data processing, ensuring that raw, unprocessed data does not immediately affect the target system.

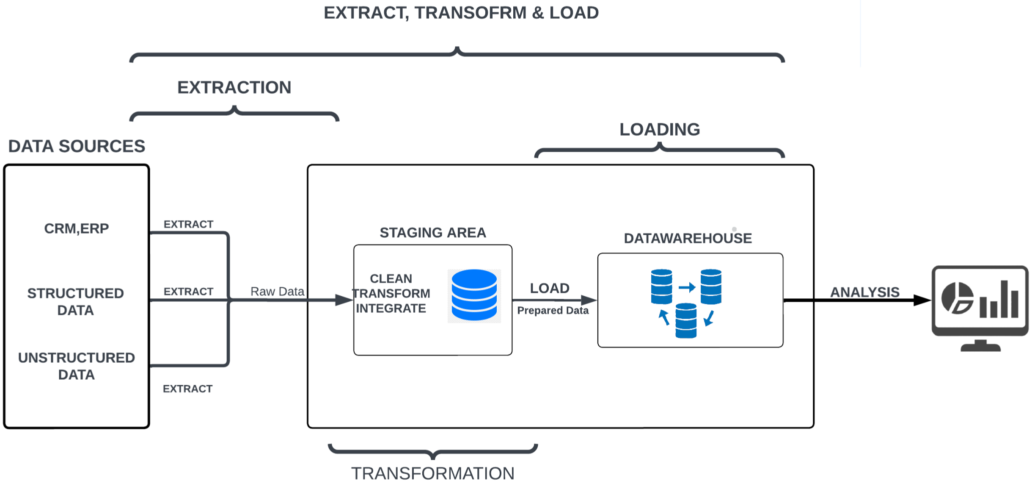


Figure 2.2: Extract, Transform & Load process

The choice of the extraction method may depend on several factors such as data volume, frequency of updates, available resources, etc... These are the most used extraction methods in the ETL process:

- **Trigger-Based method**: The source system sends a communication when a data item undergoes a change. Then, the extract operation is performed only for the portion of the data affected by the change. Most databases and web applications offer update notification capabilities to facilitate this type of data integration.
- **Incremental Extraction**: Some data sources do not offer update notification mechanisms but allow for the identification and extraction of data that has been changed in a specific time interval. In this case, the system periodically checks for changes, such as once a week, a month, or at the end of a campaign. Only the portion of the data that has been changed is then extracted, avoiding the need to transfer the entire dataset again.
- **Full Extraction**: In some situations, source systems cannot independently detect data changes or send notifications about them. Therefore, the only solution is to perform a full reload of all data. This method requires keeping a copy of the last extraction to identify new or changed records. Since this approach involves transferring large volumes of data, it is advisable to use it only for small tables.

In order to highlight the pros and cons of each extraction method, the table 2.2 below compares each method to show the advantages and disadvantages, to help make informed decisions during the ETL process.

Table 2.2: Comparing extraction methods

Factors	**Trigger-Based Extraction**	**Full Extraction**	**Incremental Extraction**
Data Volume	Suitable for smaller volumes of data that change frequently	Suitable for small to large volumes but may be resource-intensive for very large datasets	Suitable for large volumes, especially when only a portion of data changes
Frequency of Updates	Ideal for frequent data changes, capturing changes in real-time or near-real-time	Not efficient for frequent updates, as it requires scanning the entire dataset	Efficient for frequent changes, capturing only the changes
Data Source Availability	Requires real-time or near-real-time access to source data	Doesn't require real-time access to source data	Doesn't require real-time access to source data
Latency Requirements	Provides low latency for data freshness	May have higher latency due to full data extraction	Provides moderate latency for data freshness
Resource Constraints	Requires additional resources for monitoring and triggers	May be resource-intensive, especially for very large datasets	Resource-efficient, especially for large datasets with frequent changes
Data Quality	Dependent on the quality of source data; real-time validation possible	Dependent on the quality of source data; separate data cleaning often needed	Dependent on the quality of source data; real-time validation possible
Data Retention Policies	Supports historical data retention with ease	Supports historical data retention but may require archiving	Supports historical data retention with ease
Cost Considerations	May involve higher tool and resource costs	May involve lower tool costs but higher resource costs	May involve moderate tool and resource costs
Historical Data	Maintains historical data efficiently	May overwrite historical data; archiving may be required	Maintains historical data efficiently
Data Governance	Supports real-time data governance and validation	Supports data governance and validation but with latency	Supports real-time data governance and validation

2.3.3 Data transformation

After the data has been extracted and loaded in raw form into the staging area, it needs to undergo a rigorous cleaning and transformation process.
This process is crucial as it must ensure that the data, from various sources, conform

to a uniform standard and undergo a transformation process that makes them suitable for the analyses to be performed, before being stored in the main data warehouse.

The first step in **data cleaning** is the removal of erroneous or inconsistent Data. Indeed, during extraction from source systems, it is common for errors, null values, missing values, or data with invalid formats to occur.
These problems may be due to data entry errors, different field formats, or evolving business practices, therefore is important to identify and correct them to ensure that the data are accurate and reliable. The presence of duplicate data can also be problematic, so proper management of them is necessary to maintain data integrity and consistency and to avoid biased results in analysis.
During cleaning, it is fundamental to perform validity checks on the data, such as applying business rules and standards to verify that the data meets these criteria. For example, you can check that dates are in the correct format, that numeric values are within expected limits, or that foreign keys are valid. It is also important to note that, in some cases, data cleansing may include the identification and management of outliers, i.e., outliers that could adversely affect analyses if not handled properly.
Finally, you may have to use normalization techniques to bring data, which come from sources with different structures to a uniform form. This may involve converting units of measure, homogenizing product or place names, and formatting the data consistently so that the data are comparable and ready for analysis.
Data transformation is where raw data is manipulated, enriched, and structured to align with the desired format for analysis and reporting. This phase includes operations such as data aggregation, joining, filtering, and calculations. Data often originate at a very low granular level, such as individual transactions or daily records.
Aggregation involves combining these fine-grained data into coarser summaries with a higher level of granularity. In addition, in many cases, data come from multiple sources or tables, and it is essential to consolidate this disparate information.
Merging data involves merging data sets based on common keys or attributes, creating relationships that enable comprehensive analysis. However, not all data may be relevant to a particular analysis, so it becomes necessary to eliminate noise in the data and focus on the most important information.
Data filtering comes to the rescue, which involves selecting specific data points or records that meet predefined criteria.
Data transformation often requires performing **calculations** on existing data to derive new attributes or metrics. These calculations can be simple, such as calculating the total cost of items sold, or more complex, with statistical modeling or machine learning algorithms leading to further insights and supporting more advanced analyses.

Extracted data are not always sufficient to fulfill the required analysis, so it is possible to **enrich the data**, that is, augment the existing data with additional information from external sources so as to improve the context and depth of the data, enabling more comprehensive analysis.
The transformation phase significantly impacts the final analysis, as it directly affects the accuracy and relevance of the derived insights.

2.3.4 Data Loading

Once the data has been loaded and transformed the next step is to load it into the target system, which is often a data warehouse, but also data marts, data lakes, or specific databases optimized for certain types of analysis.
Loading strategies may include **bulk load** or **incremental load**, depending on the use case.

In the first approach, all the prepared data is loaded into the target system at once. This is efficient only for the initial data population but may not be suitable for real-time updates. In addition, with such an overwhelming amount of data getting moved at once, it is much easier for data to get lost within the big move. After the initial load, the data warehouse needs to be maintained and updated. This can be done with an incremental load approach, which involves loading only the data that has changed or is new since the last load. Another option used more rarely, is to consider a refresh, which consists of deleting and replacing all the data in the storage at specified intervals.

2.3.5 ETL vs ELT

The most obvious difference between ETL and ELT (Extraction, Load & Transform) is the order of operations. While the former transforms data before loading it into the target system, ELT copies or exports data from source locations, but instead of loading it into a staging area for transformation, it loads the raw data directly into the target data store to process as needed. Although both processes leverage a variety of data stores, such as databases, data warehouses, and data lakes, each process has advantages and disadvantages.
ELT is particularly useful for large volumes of unstructured data since loading can be done directly from the source. ELT can be ideal for big data management because it does not require advance planning for data extraction and storage, which can be done after the data has been stored [4].
The ETL process, on the other hand, requires more up-front definition; in fact, analysts must be involved from the beginning to define data types, structures, and relationships, as well as it is necessary to build business rules for data transformations.

This work can usually depend on the data requirements for a particular type of analysis, which will determine the level of synthesis the data must have. Although ELT has become increasingly popular with the adoption of cloud infrastructure, which gives target databases the processing power needed for transformations, it has its drawbacks in that it is a newer process, which means that best practices are still being defined.
The table 2.3 below summarizes the differences between the two approaches, leading to the conclusion that the best to use depends on the use case.

Table 2.3: ETL vs ELT

Aspect	**ETL (Extract, Transform, Load)**	**ELT (Extract, Load, Transform)**
Data Transformation	Structured transformation in staging area	Transformation occurs within the target data store
Control & Governance	Greater control over data quality and governance	Limited control pre-load; governance post-load
Raw Data Availability	Transformed data loaded; raw data not immediately available	Raw data available immediately in the target system
Scalability	Resource-intensive transformations; scalability challenges with high data volumes	Scalable, can leverage powerful target databases or cloud infrastructure
Flexibility	Less flexible, detailed planning required upfront	More flexible, suitable for evolving data requirements
Suitable for Big Data	Challenges with big data due to transformations	Ideal for big data, as it supports loading first and transformation later
Complexity	Typically simpler target data stores	Requires advanced processing capabilities in target systems
Best Use Cases	Traditional data warehousing, historical data integration	Big data analytics, cloud-based data solutions

2.4 Data visualization

Data by itself has no concrete value and no meaning, and only when it can be interpreted does it turn into knowledge. Data visualization (DV) plays a central role in the business intelligence process, providing an effective way to transform complex data into clear and meaningful information that can be easily interpreted by the human eyes [5]. It can be thought of as the mapping of data into graphical elements, which determines how the attributes of these elements change based on the underlying data.
DV is designed to explore and analyze data through a visual approach, providing a concise view of business activity. In addition, it makes it possible to identify areas for improvement in the business, understand the key factors affecting certain trends, and even make forecasts. There are several visualization techniques, the most common of which include basic representation of data through charts, maps, and key performance indicators (KPIs[2]). Another technique is the use of dashboards, which provide an overall view through cohesive individual visualizations linked by a

[2] KPIs are measurable values that help organizations assess and track their progress toward achieving specific business objectives and goals.

specific theme. A dashboard is a kind of "cockpit" that collects structured graphical elements such as tables, bar charts, diagrams, and maps, which make complex information easy to understand at a glance. Charts included in dashboards are usually categorized according to their objectives or the visual features they present. These visualizations not only simplify data interpretation but also let users explore details, see data trends over time, spot anomalies, and make informed decisions. Dashboards can be accessible via a web browser and, with current technology, can be automated and run on a predetermined schedule, resulting in up-to-date, ready-to-use data.

2.5 Cloud-based analytics

2.5.1 Cloud Computing

In the past two decades, the computing world has experienced an unprecedented revolution thanks to the rise of cloud computing. This innovative technology has radically transformed the way businesses manage their data, computing resources, and digital services.

Cloud computing is a computing service delivery model that provides access to computing resources and applications via the Internet.

Instead of hosting data and applications on local or on-site servers, everything is hosted on remote servers and managed by cloud service providers. Users can access these resources on-demand, via any Internet-connected device, making the use of computing resources more flexible and accessible. [6]

The main enabling technology for cloud computing is the **abstraction**, needed to separate a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks.

The goal of cloud computing is thus to enable users to benefit from all the technologies provided by the cloud service, without the need for deep knowledge or experience with each of them.

Nowadays, the companies that have most imposed their dominance in this area are three big tech giants, who were the first to take advantage of the wave of the cloud era, imposing their cloud computing styles based on the underlying resource abstraction technologies:

- **Amazon**'s cloud computing is based on server virtualization technology and was released during 2006 – 2007, under the name Amazon Web Service™ (AWS)[7].
- **Google**'s cloud computing is based on technique-specific sandbox. The platform, which is called Google App Engine™ (GAE), was released to the

public as a kind of Platform as a Service (PaaS) cloud computing service in 2008.

- **Microsoft Azure™**[8] was released in Oct. 2008, which uses Windows Azure Hypervisor (WAH) as the underlying cloud infrastructure and .NET as the application container.

2.5.2 Benefits of Cloud Computing

There are several reasons why the adoption of cloud-based analytics is growing steadily and is the result of several significant advantages over on-premises solutions. First, **scalability** is a crucial advantage. The cloud allows companies to easily adapt computing resources to their specific needs. They can increase or decrease the use of resources without having to invest in new hardware infrastructure, which means greater control over costs and unprecedented flexibility.
Likewise, **accessibility** is critical in an increasingly mobile and remote world. Thanks to the cloud, data, and applications are accessible from anywhere and from any device with an Internet connection. This accessibility greatly improves business productivity and responsiveness.
Another crucial aspect is **cost reduction**. Cloud adoption can reduce operational costs significantly, so companies can avoid purchasing, managing, and maintaining expensive hardware and server infrastructure, saving financial resources significantly.
Agility is another key benefit offered by the cloud. Companies can respond quickly to changing market needs, and deploy new applications and services faster and with less effort, enabling them to remain competitive and responsive.
Security is a top concern for many companies, and this is where the cloud can make a difference. Many cloud service providers offer advanced security solutions, enabling companies to better protect their data than they could on-premises.
In addition, the cloud fosters **collaboration**. Companies can easily share data and resources among teams and business partners in real time, facilitating collaboration and improving overall operational efficiency.
Finally, automatic updates are another benefit. Cloud service providers take care of update and patch management, ensuring that companies have constant access to the latest versions of applications. This greatly simplifies the maintenance and updating of IT resources, enabling companies to stay on the cutting edge of technology.

2.5.3 Types of clouds

Cloud computing offers different deployment models to meet specific business needs, each with its own advantages and disadvantages. In fact, several factors, such as solution functionality, cost, organizational and integration aspects, as well as security, influence the decision of enterprises and organizations to choose the right type of cloud for them.

Public

Public clouds are delivered and managed by third-party cloud service providers and can be offered on a paid subscription basis or for free. They are inexpensive and easily scalable, making them particularly attractive to startups and small and medium-sized businesses that want to avoid high upfront infrastructure costs. Public clouds also offer a wide range of services, from simple storage to advanced machine learning. However, significant data security and privacy concerns arise as resources are shared among different clients on the same cloud infrastructure. Organizations that handle sensitive data or subject to strict regulatory requirements may consider public clouds less suitable.

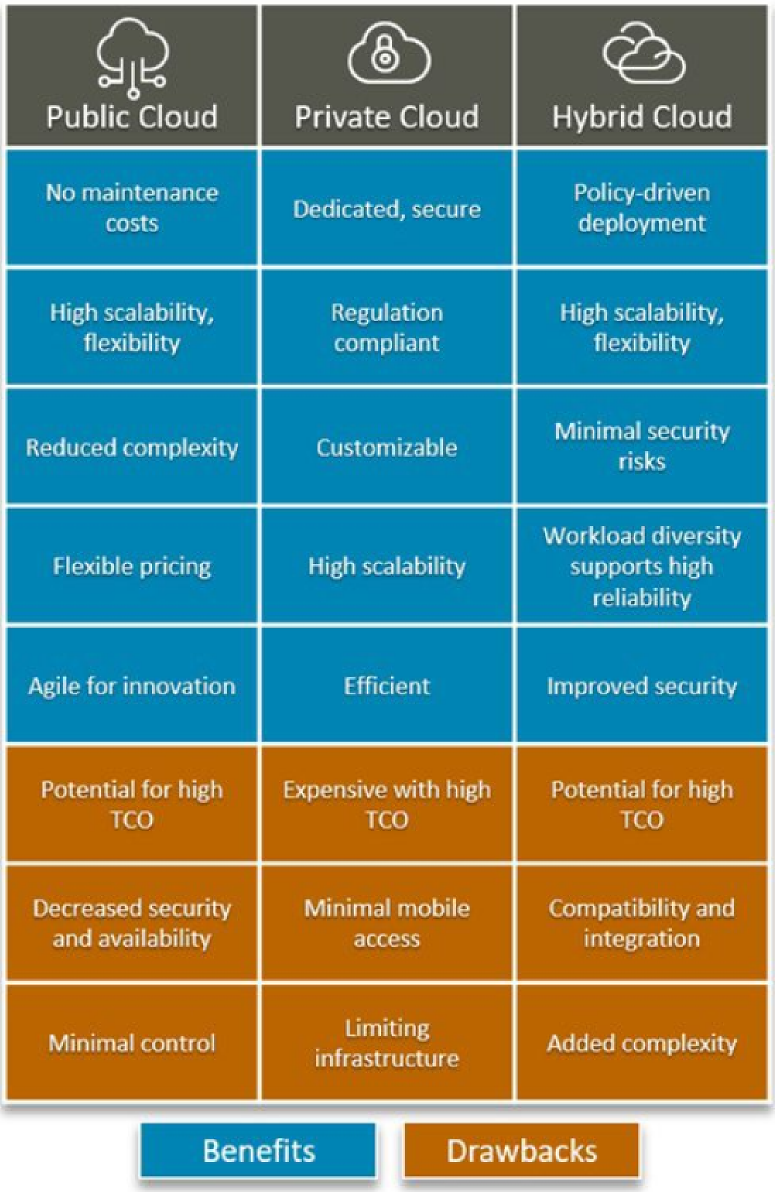
Private

In contrast, private clouds are dedicated to a single organization and can be hosted either in-house or by a third-party provider. This configuration provides greater control, security, and customization options. Industries with strict regulatory frameworks, such as healthcare and the financial sector, often prefer private clouds to ensure compliance. However, private clouds carry higher upfront costs for implementation and ongoing maintenance, which may discourage smaller enterprises.

Hybrid

Hybrid clouds offer the best of both worlds, combining elements of public and private clouds, and their approaches are widespread because almost no one today relies entirely on a single public cloud. Hybrid solutions let you to migrate and manage workloads between these various cloud environments, giving businesses the flexibility to deal with constantly changing workloads based on your specific needs. This configuration is particularly beneficial for organizations that manage both sensitive and non-sensitive data. However, managing a hybrid cloud environment can be complex and requires expertise in both public and private clouds.

In summary, choosing the right cloud model depends on a number of factors, including the sensitivity of the data being managed, budget constraints, scalability requirements, and regulatory compliance. Many forward-looking organizations adopt a multi-cloud strategy, leveraging various cloud models for different areas of their operations to optimize efficiency and convenience. A comparison of the above-mentioned cloud types is shown in figure 2.3.



Public Cloud	Private Cloud	Hybrid Cloud
No maintenance costs	Dedicated, secure	Policy-driven deployment
High scalability, flexibility	Regulation compliant	High scalability, flexibility
Reduced complexity	Customizable	Minimal security risks
Flexible pricing	High scalability	Workload diversity supports high reliability
Agile for innovation	Efficient	Improved security
Potential for high TCO	Expensive with high TCO	Potential for high TCO
Decreased security and availability	Minimal mobile access	Compatibility and integration
Minimal control	Limiting infrastructure	Added complexity

Benefits

Drawbacks

Figure 2.3: Cloud types Comparison

2.5.4 Cloud Models

On-premise IT infrastructure represents the greatest level of responsibility for the user and manager. When the hardware and software are all on-premise, it is up to you and your team to manage, upgrade, and replace each component as needed. Cloud computing permits you to outsource the management of one, several, or all components of your infrastructure to a third party. In fact, the term "As-a-service" generally refers to a cloud computing service provided by a third party, and there are three types, depending on how much on-premise infrastructure is left to manage. In particular:

- **IaaS (Infrastructure as a Service)**: this model offers companies the ability to lease virtualized IT infrastructure such as servers, storage, networks and virtual machines, avoiding the need to physically own and manage them. IaaS offers a high level of flexibility and control, allowing organizations to create and customize their own IT environment. This flexibility is especially beneficial for developers and IT professionals who need a platform on which to run highly customized applications or services.

- **PaaS (Platform as a Service)**: the PaaS model goes beyond infrastructure, providing a complete environment for application development and deployment. This approach provides tools and services for creating, testing, and releasing software without the need to manage the underlying infrastructure. PaaS is ideal for developers who wish to focus solely on writing code without having to worry about infrastructure management. This platform greatly accelerates application development and deployment, saving companies valuable time.

- **SaaS (Software as a Service)**: in this model, software is delivered as a service over the Internet, allowing users to access the application through a simple Web browser without the need to install or manage the software locally. SaaS is used for a wide range of applications, including email solutions, customer relationship management (CRM), project management, and more. This model is particularly suitable for companies that wish to avoid the complexity associated with software installation and maintenance. SaaS offers convenience and accessibility, although it may limit customization options compared to local or PaaS solutions.

The choice among these models depends on the specific needs of the organization and the applications or services it intends to use. Each model offers a different degree of control, customization, and complexity (summarized in the image 2.4 below), allowing companies to select the option that best suits their operations and strategic goals.

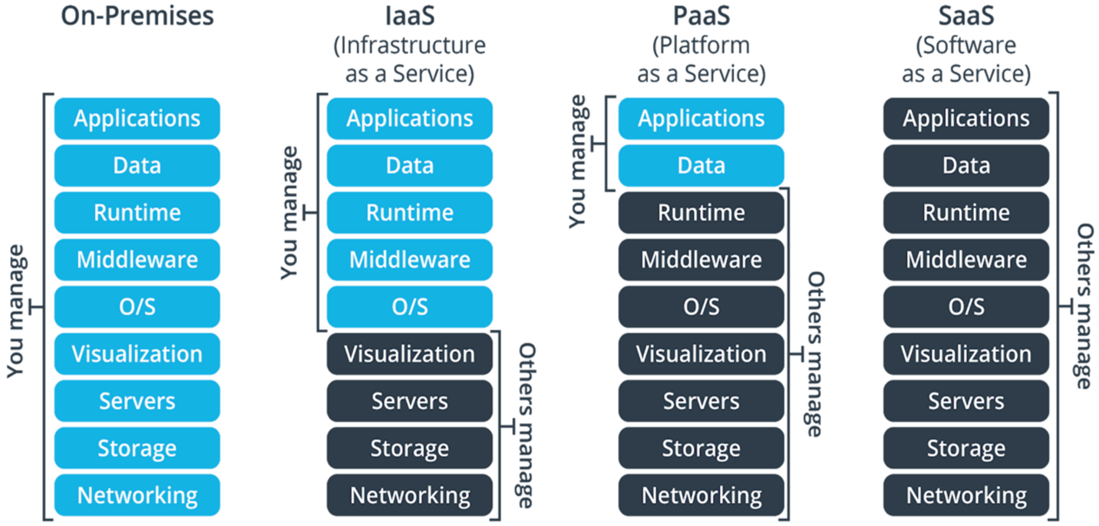


Figure 2.4: IaaS vs. PaaS vs. SaaS

2.5.5 Cloud Business Intelligence

Cloud Business Intelligence (BI) is the natural evolution of traditional business intelligence solutions pushed to cloud computing. This discipline combines advanced data analysis and BI technologies with the scalability, flexibility and accessibility offered by the cloud. The main reason behind the adoption of Cloud BI is the growing need for companies to obtain critical information quickly and efficiently, without having to deal with costly local deployments or specialized hardware.

Among the benefits of Cloud BI is universal accessibility, as it let users to access data and dashboards from any location and device connected to the Internet. In addition, scalability is a key factor, as resources can be easily adapted according to business needs, avoiding over-investment in hardware infrastructure. Automatic maintenance and updates are provided by cloud service providers, simplifying management and ensuring access to the latest features.

However, there are some challenges associated with Cloud BI, such as data security and dependence on cloud service providers. Companies need to take strict measures to protect sensitive data and carefully consider regulatory compliance. In addition, the use of Cloud BI can result in recurring operational costs, although these are often lower than the initial investment costs for local infrastructure. Overall, Cloud BI represents a significant step toward optimizing enterprise data analysis, but it requires careful planning and management to maximize its benefits.

Chapter 3

Project under analysis: sales force monitoring

This chapter introduces the context of the project analysis, providing an introduction to the companies involved, in order to give a complete and comprehensive picture of the "reference environment." In addition, a small overview of the project performed and what my contribution was is given. The configurations of the project environments and the nonfunctional requirements requested by the client are then described. The client is represented by Poste Italiane S.p.A., while the contractor is represented by Advant S.r.l.

3.1 Context Analysis

3.1.1 Contractor: Advant S.r.l

Advant S.r.l[9] is a professional ICT consulting company born in Rome in 2014 from Proge-Software S.r.l., a leading company in the IT market, to expand the range of services and the offer of technological solutions available to customers. Advant, through the most advanced technological tools and the application of the most recent business management models, offers Information Technology solutions and services to medium and large businesses and the public administration. The main areas of competence of the company are:

- **Data & AI**: turn data into knowledge by analyzing, modeling, and returning results in interactive synoptic dashboards. Design and implementation of complete business intelligence systems of medium and large size, in the different business areas as well as creation of predictive systems based on Artificial Intelligence.

- **Corporate Performance Management**: development of strategic planning, budgeting, reporting, and business performance analysis projects.
- **Business Solution Development**: Implementation of specific business solutions based on client's needs, directing choices toward the use of Cloud services in line with modern low-code/no-code paradigms.

Thanks to the quality of its work over the years, the company has succeeded in attracting more and more new clients and retaining many of them. Among the big names with whom the company has had the pleasure of working certainly stand out *Pfizer*, *Toyota Financial Services Italia*, *Lottomatica*, *RCI*, the aforementioned *Poste Italiane* and many others.

3.1.2 Customer: Poste Italiane S.p.A

Poste Italiane S.p.A. has emerged as the nation's premier postal service provider in Italy, boasting a rich heritage, widespread infrastructure, and a varied range of services. Precisely, the company's core business spans from mail and parcel delivery to financial and banking operations, insurance solutions, payment services, telecommunications, and more. As a company deeply entrenched in the fabric of Italian society, has continually evolved to meet the dynamic needs of its customers in an era of digital transformation and global connectivity. The company has strategically expanded into the financial sector, leveraging its extensive customer base and nationwide presence to offer a range of banking services, including savings accounts, loans, investments, and insurance products. This diversification has not only strengthened Poste Italiane's revenue streams but has also let it to foster deeper customer relationships by providing integrated financial solutions conveniently accessible through its vast network of branches and post offices. Through the use of technology and innovative platforms, the company has developed an online presence that enables customers to access a wide array of services.
Over the past few years, Poste Italiane has focused on Microsoft technology platforms to improve productivity and collaboration, as well as renewing the post office network and ecosystem of services offered for the benefit of the market, becoming a virtuous example in the Italian digital transformation landscape.
Thanks to Microsoft Azure, has evolved its IT Infrastructure making it more scalable and secure to ensure greater agility and speed in innovation processes[10]. For organizations invested in Microsoft products, Azure offers seamless integration, enabling hybrid cloud deployments and smooth migration. Thus, it is a preferred choice for businesses with on-premises infrastructure, offering robust hybrid cloud computing services capabilities. The objective is to enrich Poste Italiane's CRM (Customer Relationship Management), which represents the strategy and a set of business practices designed to manage and improve customer relationships, with

synthetic indices of potential on small business and e-commerce merchants to improve business leads for the sales force and sales channels.
The digital transformation undertaken by Poste Italiane in recent years has involved not only its offerings but also its distribution model, which, through an omnichannel strategy, ensures that the company can deliver services in ways that are consistent with the changing needs of customers. [11]

The Group's integrated multichannel platform provides for customer preservation and service delivery through 3 channels:

- **The proprietary physical network**: it consists of Post Offices, the commercial articulation specialized on business customers, and the logistics network for mail and parcel delivery;
- **A digital infrastructure and remote points of contact**: consisting of all the Group's digital properties and contact center, capable of serving the entire national population;
- **The third-party physical network**: consisting of about 62 thousand points, the result of commercial partnership agreements for the marketing of the Group's products and services.

The Post Office network is governed by the Private Market business function organized into Territorial Macro Areas, Branches, and Post Offices covering the whole country. In figure 3.1 it is possible to observe their distribution

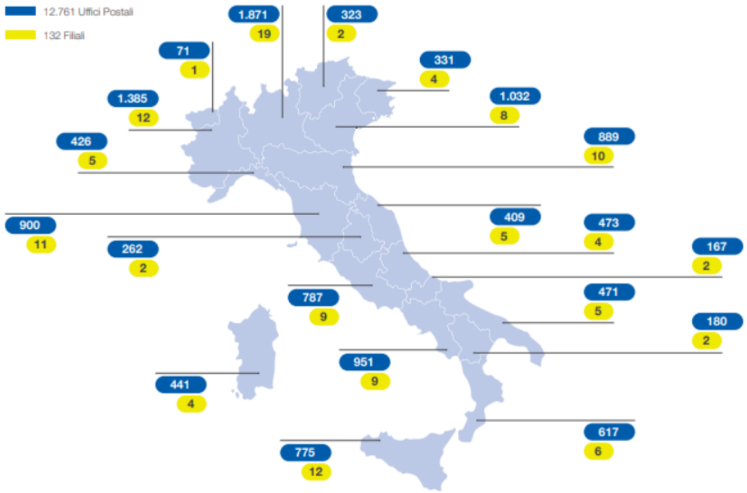


Figure 3.1: Post offices distributions

The commercial supervision and sale of the Group's products and services to businesses and Central and Local Public Administration is guaranteed by Poste

Italiane's Business and Public Administration Market function. The sales force organization ensures territorial presidium focused by product segment, through:

- **3 Macro Sales Areas** (North West Lombardy, North Central and North East, South Central), with exclusive commercial responsibility for mail and communication revenues and commercial support to the specialist sales force 3.2;
- **2 Sales Areas** dedicated to the commercial management of logistics and parcel products/services and specialized by industry:
  - **Industry 1**: Health & Beauty, Electronics and Information Technology, Other Sectors;
  - **Industry 2**: Food & Grocery & Pet, Homeliving & Fashion, Platform & Solution Players;
- **1 Sales Area** specialized on offering financial and insurance products on large business customers and Public Administration;
- **1 Commercial Area** for Central and Local Public Administration related to Regions and Metropolitan Cities;
- **1 Commercial Area** dedicated to the commercial management of partnerships with third-party networks and remote sales. The remote offering service allows advisors to send a savings or investment proposal to the client's private area; the client can consult the proposal and decide whether to finalize it directly online or in the Post Office.

3.2 Project Overview

The project in question was born as an evolutive of the BI system for the analysis of sales force data already in use by the client, built by my company in previous years. The objective is to give evidence to sellers operating in the Italian territory, by monitoring the performance of services and products sold on the business segments of interest, in terms of volumes and revenues.

The technologies used belong to Microsoft's Azure stack such as Azure SQL database, SQL Server Integration Service (SSIS), Azure Data Factory (ADF), SQL Server Analysis Services (SSAS), and PowerBI.

The system monitors the sales force in the MIPA sector (medium-sized enterprises and public administration) that allows the sales organization to detect sales, earnings, volumes, and performance segmented by area of responsibility.

The aforementioned evolution includes the addition of a new data mart to the main

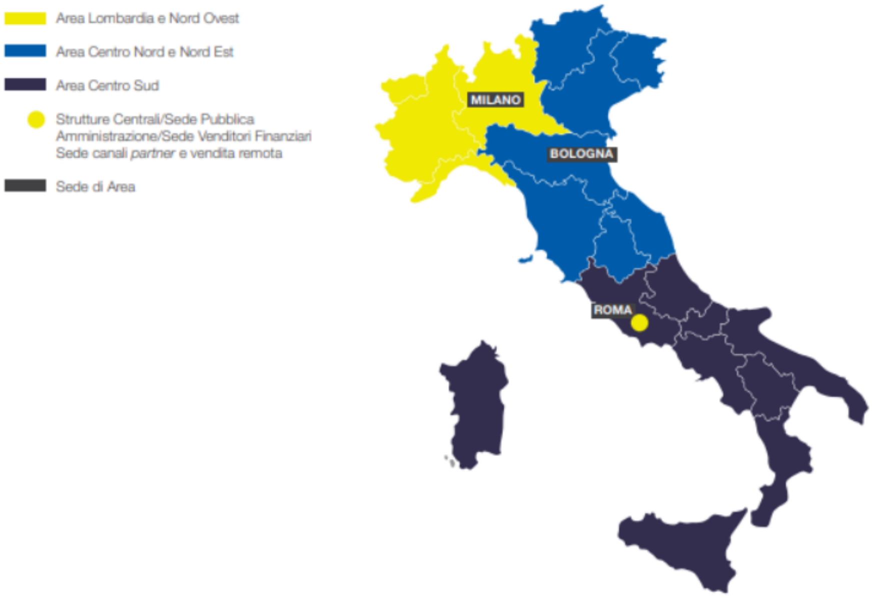

Area Lombardia e Nord Ovest
Area Centro Nord e Nord Est
Area Centro Sud
Strutture Centrali/Sede Pubblica Amministrazione/Sede Venditori Finanziari Sede canali *partner* e vendita remota
Sede di Area
MILANO
BOLOGNA
ROMA

Figure 3.2: Macro areas enterprises and public administration

sales force data warehouse, with new analytics defined by KPIs of customer interest. Specifically, it is the monitoring of revenues and volumes derived from the sale of auxiliary services in the parcel business segment. My contribution was to develop the entire data mart, starting from the requirements gathering with the client to the design and implementation of the solution. SQL code was developed in the form of procedures to extract, clean, transform, and organize the data, as well as to manage the process log, error history, and flow logic. The implementation of the data mart ETL flow and its integration within the sales force parent flow, through the creation of data pipelines in the ADF. The creation of the tabular model in SSAS based on the table in the warehouse and its enrichment through analysis measures. The creation of the ad-hoc reports in PowerBI.

3.2.1 Preliminary configurations

Environments configuration

In Azure, a "tenant" is an isolated and dedicated instance of the Microsoft Azure cloud environment, representing a specific organization or customer using Azure services to manage cloud resources and applications. Within an Azure tenant, resources can be organized and managed using a concept called a resource group. A resource group is a logical collection of related resources that are part of a specific application or project, allowing resources to be organized in a consistent manner, simplifying resource management, deployment, and monitoring.

The resource groups used can be considered as separate, parallel entities. Each group contains resources such as Azure Databases, Analysis Services, Virtual Machines, and Data Factory. However, each group is associated with a specific server based on its destination, which can be a development, test, or production environment.

Resource groups are a key element in creating the environments necessary for solution development. The entire process begins with the solution design and development phase, which takes place in specific environments dedicated to development. Once the developments are completed and the solution is ready for further testing, it is replicated in the test environments. In the test environments, those in charge perform a series of thorough tests to assess the functionality of the system and ensure that it meets all predetermined requirements. This phase is crucial to detect any bugs or errors and ensure that the solution works reliably. The strength of this division of environments lies in the fact that it permits for continuous development in separate environments, avoiding interference between development and the testing and production phases. Finally, once the solution has been sufficiently tested and is ready for use, it is implemented in the production environment. This step makes it available for use by end users, thus completing the solution development and deployment cycle.

Security

To ensure maximum protection of data and resources in the various environments sophisticated security systems are implemented. In each environment, whether intended for development, testing, or production, the level of access is carefully managed through the creation of user accounts specific to the needs of each team member.
Access to these environments is exclusively through the use of virtual private networks (VPNs[1]), which add an additional layer of protection. These VPNs are configured to ensure that only authorized and pre-authenticated personnel can access the data and resources contained in each environment. Further, strict multi-factor authentication (MFA[2]) procedures are also in place to ensure that the identity of anyone seeking access is reliably verified.

3.2.2 Non-functional requirements

Non-functional requirements, also known as "quality requirements" or "performance requirements," are critical aspects that define characteristics of the system that go beyond its basic functionality. These requirements are primarily concerned with the qualities, performance and constraints of the system, rather than the specific functionality that the system must perform.

Preview and target dashboards

The customer expressed the need for the creation of two types of final reports, preview and target ones. The ETL must be scheduled to run once a week, so typically on Monday morning, they go to see the performance of the past week. Before the updated reports reach the end-users, the data must be validated by some kind of users called controllers. This is why they check the quality of the data and whether it is in line with their expectations in the preview reports. When they are certain that the data is correct, they perform a rollover to the target environment, via a button configured on their website that triggers the appropriate Data Factory's pipeline. This involves creating two identical tabular models as well as two reports for the data mart and updating them within the Azure Data Factory. Through the ETL procedure the model in preview and the reports in preview will

[1] VPNs are secure and encrypted connections that allow users to access the internet and network resources while safeguarding their privacy and data from potential eavesdropping or cyber threats.

[2] MFA is a security process that requires users to provide two or more separate forms of identification to verify their identity,

be updated, then it will be the controllers' burden to trigger the creation of the reports in target for the end users.

Row-level security

An additional requirement concerns the profiling of data via row-level security (RLS). RLS is a security feature in database management systems that restricts access to specific rows of data within a table based on defined security rules or filters. This helps ensure that sensitive data, such as financial data or customer information, can only be viewed and interacted with by users authorized to access it. This is crucial when dealing with critical business information that should not be disclosed to just anyone, and it allows the information displayed to be tailored to the user's role or responsibility, ensuring that they see only data relevant to their activities. The filter rule follows the following hierarchy between salespeople, coordinators, and managers:

1. Each seller can see the data of their sales portfolio.
2. Each coordinator can see the data of his sellers.
3. Each manager can see the data of his coordinators.

The client provides a table that contains, for each user account that is going to use the system, information regarding the area of responsibility, the name of the vendor, and its coordinators and managers. The information provided is then used to filter the size of the sales network through the implementation of filters within the tabular model.

Chapter 4

Data-mart Design

4.1 Requirements Analysis

Gathering requirements with the customer in a business intelligence project is the first step in creating a solution that fully meets needs and objectives. This is essential for understanding the data needed to make informed decisions; it also helps to clearly define the scope of the project and avoid costly changes and late revisions.

The need for this new data mart arose to give Poste CEP segment (Courier and Parcels) vendors evidence of the auxiliary services they are selling concerning customers, territories, and services. It represents the monitoring of their active customer base in relation to these services sold on revenues and volumes. How these customers relate to sales portfolios, territorial areas, and services sold.

During the various meetings, the client expressed the need for the analysis to consider data from January 1, 2021, thus providing a complete view of the previous two years. In addition, several business questions were defined that describe the focus of the analysis and are shown in Table 4.1. Each business question has been discussed and analyzed to identify the preliminary dimensions, attributes, and measures of interest.

Table 4.1: Business questions

N° Question	Business Question
1	Total revenues, revenues monthly trend and revenues by semester
2	Total Volumes, volumes monthly trend and volumes by semester
3	Top 5 clients per earning, volume and service
4	How top 5 clients earnings and volume are related to portfolio type and sales network
5	Earning and volumes per service and auxiliary service
6	Overview for vendors on the customers they manage

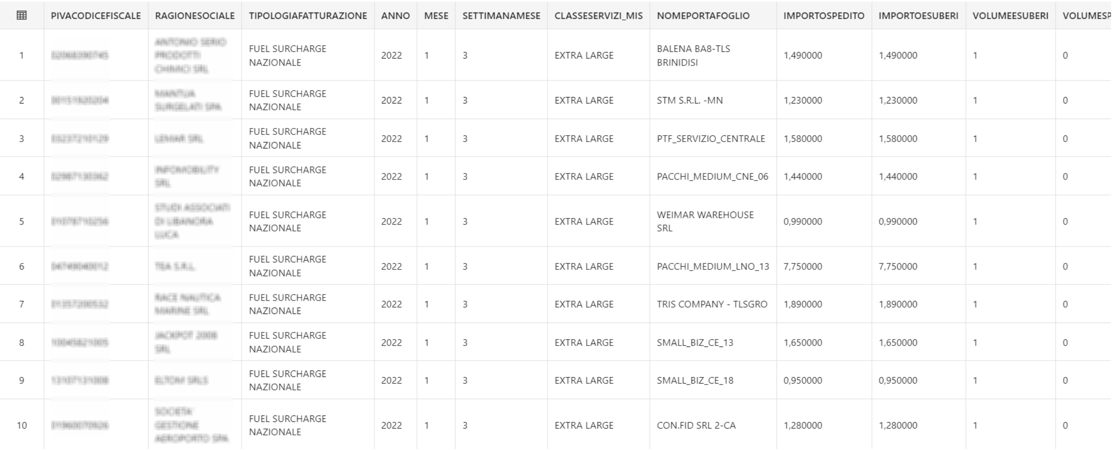
During the requirements gathering phase, the customer, provided an extraction of the data source so as to help in the design phase. A snapshot of how the source looks like is in figure 4.1, and this presents information and measures of interest such as:

- **VAT number**: the VAT number allows to associate the transaction referred to the customer with the dimension dedicated to him, thus making it possible to trace the sales portfolios to which he belongs.
- **Billing type**: this corresponds to the auxiliary service associated with the sale.
- **Service class**: this represents the class of service to which the auxiliary service belongs.
- **Year, month, and week of the month**: the data from the extracted transactions have a weekly granularity, so there is no daily transaction detail. The information reported in the source is the year, the month, and the week in the month.
- **Surplus volume**: refers to the quantity of the individual auxiliary service sold, without regard to the service to which it belongs.
- **Shipped volume**: refers to the total of the dispatch base, which we do not have as a detail but which also consists of the supplementary volume.
- **Surplus amount**: refers to the quantity of the auxiliary services only, without regard to the service to which it belongs.
- **Shipped amount**: refers to the total amount of the shipment.
- **Portfolio**: represents the sales portfolio to which the transaction belongs and is useful for tracing the seller's areas of expertise.

It is important now to be able, from the requirements gathered and the extraction provided, to design a data mart capable of optimally supporting the required analyses.

4.2 Conceptual Design

The first step in developing the data mart consists of the conceptual design of the solution, by defining 'facts' and 'dimensions'. This is useful for reasoning about the characteristics of the data at an abstract level, independent of implementation concerns.



▦	PIVACODICEFISCALE	RAGIONESOCIALE	TIPOLOGIAFATTURAZIONE	ANNO	MESE	SETTIMANAMESE	CLASSESERVIZI_MIS	NOMEPORTAFOGLIO	IMPORTOSPEDITO	IMPORTOESUBERI	VOLUMEESUBERI	VOLUMESP
1	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	BALENA BA8-TLS BRINIDISI	1,490000	1,490000	1	0
2	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	STM S.R.L. -MN	1,230000	1,230000	1	0
3	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	PTF_SERVIZIO_CENTRALE	1,580000	1,580000	1	0
4	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	PACCHI_MEDIUM_CNE_06	1,440000	1,440000	1	0
5	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	WEIMAR WAREHOUSE SRL	0,990000	0,990000	1	0
6	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	PACCHI_MEDIUM_LNO_13	7,750000	7,750000	1	0
7	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	TRIS COMPANY - TLSGRO	1,890000	1,890000	1	0
8	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	SMALL_BIZ_CE_13	1,650000	1,650000	1	0
9	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	SMALL_BIZ_CE_18	0,950000	0,950000	1	0
10	[redacted]	[redacted]	FUEL SURCHARGE NAZIONALE	2022	1	3	EXTRA LARGE	CON.FID SRL 2-CA	1,280000	1,280000	1	0

Figure 4.1: Source data extraction

Facts represent the collection of observations that express the target variables or measures of the analysis of our Business Intelligence system. In our case, the fact table will express the quantities of revenues and volumes based on the auxiliary services sold and will be enriched later in the development of numerical measures (useful for speeding up analysis).
A dimension represents contextual information in which a performance measurement of the business process of interest is captured. Dimensions describe discrete domains, usually organized in levels of aggregation, and represent a perspective against which to perform the analysis. In general, are described by a set of attributes used to qualify, categorize, or summarize facts in reports.
Bearing in mind that the granularity of the fact is weekly, these are the dimensions identified:

- **Calendar**: classic dimension which allows contextualization of the timing of each event (contains dates for the last 3 years).
- **Product**: specifies the technical characteristics of all products that can be sold.
- **Client**: serves to classify each customer on the basis of the product they buy and therefore the type of portfolio they invest in.
- **Sales Network**: associates each portfolio with a specific coordinator, manager, and salesperson. The row-level security filters are applied over this dimension.

In table 4.2 can be observed the dimensions and the measures identified to answer the business questions.

Table 4.2: Preliminary dimensions and measures

Business Question	Dimensions	Measures
1	Calendar(year, semester, month)	Total earnings, Total surplus earnings, Total shipped earnings
2	Calendar(year, semester, month)	Total volume, Total surplus volume, Total shipped volume
3	Client(name), Product(service, auxiliary service)	Total earnings, Total volume
4	Client(name), Sales Network(portfolio, area)	Total earnings, Total volume
5	Product(service, auxiliary service)	Total earnings, Total volume
6	Client(name), Product(service, auxiliary service), Sales Network(portfolio, area)	Total earnings, Total volume, Total surplus earnings, Total shipped earnings, Total surplus volume, Totale shipped volume

The initial conceptual model in figure 4.2 shows the fact table with the measures, from which the links to the dimensions, characterized by a list of attributes, some of which belong to dimensional hierarchies:

- **Calendar**: Day → Week → Month → Semester → Year
- **Sales Network**: Portfolio type → Portfolio → Area
  Salesman → Coordinator → Manager

- **Product**: Auxiliary Service → Service

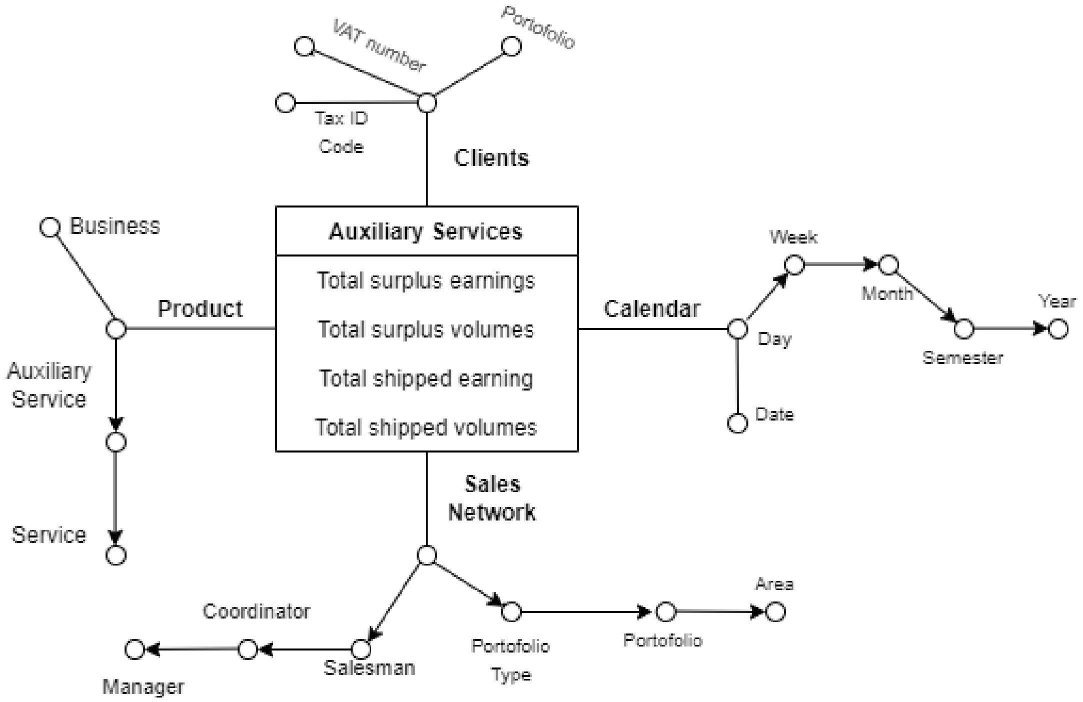


Figure 4.2: Data mart initial conceptual design

4.3 Logical Design

The objective of this step is to move from the conceptual design created earlier toward the logical structures used for storage within a relational DBMS. In designing this step, a ROLAP solution was chosen. ROLAP, which stands for 'Relational Online Analytical Processing', implements the solution through the use of the relational model: data is collected in columns and rows and information is retrieved on demand through user-submitted queries.
When designing a logical schema, the most important thing to decide is the type of schema that best fits the data mart. In the case of our data mart, considering that the data come from a single source and the required analyses can be fulfilled by a single fact table, the most suitable schema is the star schema.
A star schema has a single fact table in the center, containing the business 'facts', which is linked to the dimension tables via foreign keys. These foreign keys are used to establish relationships between the data in the fact table and the related details

in the dimension tables. This structure greatly simplifies complex data aggregation and analysis queries, ensuring high performance as the data is highly denormalized. This means that within the fact table, much of the information is duplicated or made redundant. While resulting in a slight increase in the amount of data stored, denormalization greatly simplifies data query and aggregation operations, reducing the need to join complex tables to retrieve information. It also permits a certain ease of understanding and design, which facilitates the creation of reports and analyses. The flexibility to add new dimensions or measures without changing the fundamental structure is also something to take into account.

The type of relation between dimensions and the fact table is *one-to-many*: a single row in a dimension table can be associated with multiple rows in the fact table. For instance, a single customer from the client dimension may have multiple transactions or sales records in the fact table. It was then necessary to introduce a surrogate key as the primary key for each dimension table and the corresponding foreign key in the fact tables. A surrogate key is generally a numeric attribute automatically generated by the database that speeds up the calculation of queries and prevents the inadvertent introduction of duplicate records within dimensions. The logic schema for the data mart developed can be observed in figure 4.3, which also includes an RLS table needed to profile the users depending on the sales area of their competence.

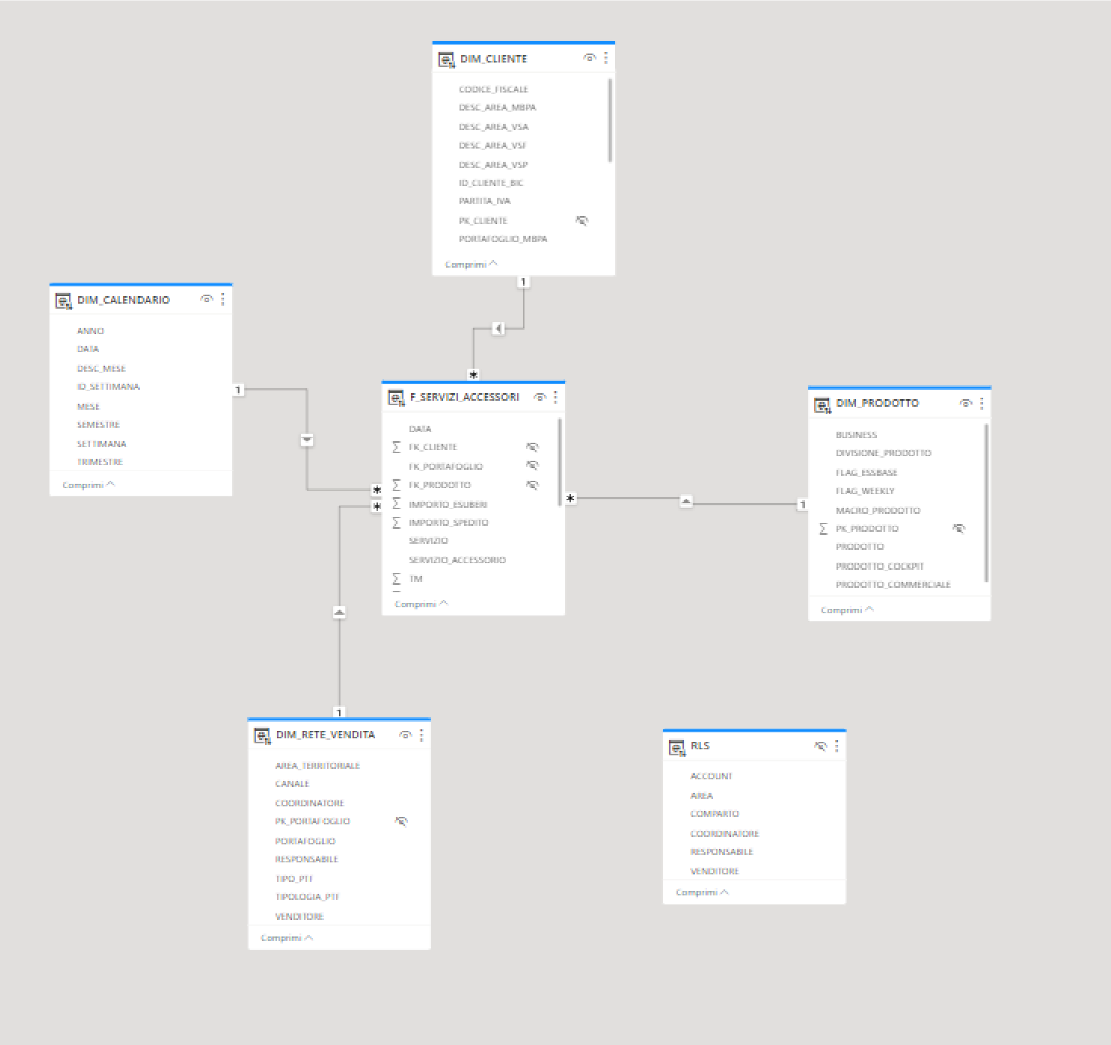


Figure 4.3: Data mart logic design

Chapter 5

Developments

The upcoming chapter focuses on the development environments and software tools utilized in designing the data warehouse and BI system, with a specific emphasis on Microsoft Cloud Stack. These technologies are highly compatible with the overall BI objectives and form the foundations for this project, leveraging its capabilities for ETL, orchestration, and reporting tasks. They provide a comprehensive suite of tools and functionalities that synergistically work together, resulting in reduced integration efforts across the different stages.

5.1 Solution Architecture

The proposed architecture represents a classic end-to-end business intelligence process, which starts from raw data, extracted from external sources, and leads to accurate data processing until end users can benefit from their insights in the form of reports within interactive dashboards in a web portal. Four technologies provided by Microsoft Azure, Microsoft's cloud computing platform are used to achieve this:

- **Azure SQL Database**: cloud database hosting the staging area and data warehouses;

- **SQL Server Analysis Services**: analytical data engine for the creation of the tabular model and analysis measures;

- **Azure Data Factory**: cloud-based data integration service, let the entire process to be orchestrated;

- **Power BI**: provides interactive data visualization and enables the creation of reports;

As can be seen in the image below 5.1, these 4 technologies also constitute the 4 macro-blocks of the architecture.

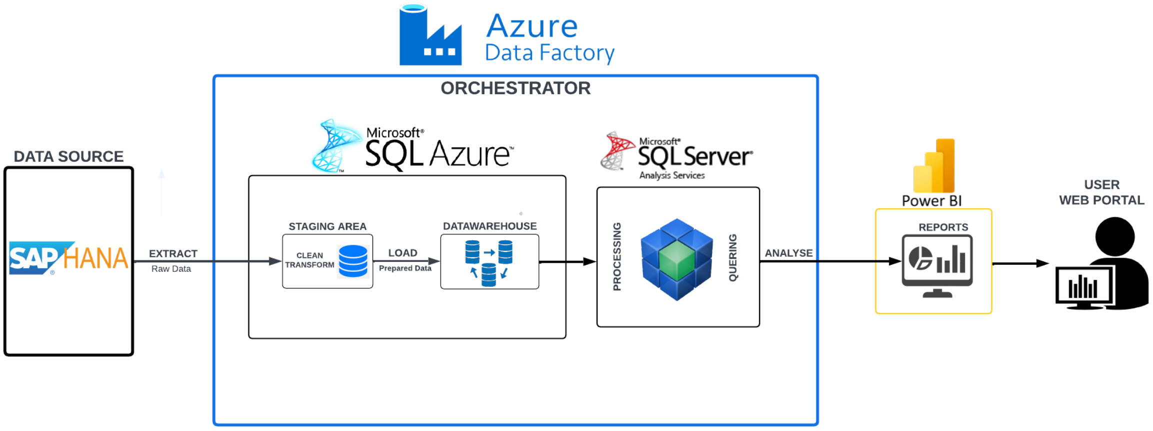


Figure 5.1: Architecture of the developed BI Pipeline

5.1.1 Azure SQL database

Azure SQL Database is a PaaS deployment option of Azure SQL that abstracts both the operating system and the SQL Server instance. As a result, the responsibility for managing the hardware, the operating system, and the SQL Server instance lies with the service provider, in this case, Microsoft Azure.
It is a fully managed relational database solution that offers users the ability to create, manage, and scale SQL databases in a cloud environment and primarily supports the Microsoft SQL Server database engine.
This deployment option makes it possible to quickly obtain a database and immediately start developing applications. Azure SQL Database is also the only deployment option that supports scenarios requiring unlimited database storage space (100TB+ **Hyperscale**) and automatic scalability for unpredictable workloads (**Serverless**) [12].
It constitutes the first block of the architecture in chronological order of development. Here, within the customer's development environment, the necessary data structures for the creation of the staging area and data mart were created in the sales force database. In addition, the stored procedures for transforming the data and loading them into the data warehouse were written, as well as the logic tables and procedures needed to orchestrate the ETL process.

5.1.2 Azure Analysis Services

Azure Analysis Services is a fully managed distributed platform as a service (PaaS) that provides enterprise-grade data models in the cloud and is based on the SQL Server Analysis Services engine [13]. The latter is an analytical engine that provides enterprise-level semantic data models for reports and client applications.
It supports tabular models, i.e. a data model that focuses on creating an in-memory data structure designed to analyze and aggregate data efficiently, at all levels of compatibility.
These guarantee high performance when querying data, much faster than in traditional relational databases. here are some of the key points of a table model in Analysis Services:

- **In-memory table**: means that data is loaded into memory even before a data request is made, to enable fast queries and interactive analysis. This in-memory approach is designed to significantly improve performance compared to traditional databases.
- **Highly compressed**: the high degree of data compression, achieved through the *VertiPaq engine*, results in an in-memory data model up to 10 times smaller than that on disk. Data is compressed efficiently to reduce memory usage, allowing large amounts of data to be loaded and retrieved more quickly and efficiently.
- **Detail-level rows**: rows of data are stored at the lowest level of detail or granularity. This means that each individual row of data is available for analysis, allowing the data to be aggregated in many ways to answer different analytical questions.
- **Vertically stored columns**: data columns are stored vertically, instead of horizontally as in traditional relational databases, thus allowing better compression and selective access to the columns needed for queries.
- **Includes measurements, calculations, and more**: measurements and calculated columns can be included, allowing analytical calculations to be performed on the data without having to physically store it, making the model more flexible and adaptable to analytical needs.

It constitutes the second block of the architecture in chronological order of development. Once the data has been processed and loaded into the target data mart, the tabular model is created in the on-premise version of SQL Server Analysis Services. Here, it is possible to connect directly to the Azure SQL Server containing the database and tables, perform further transformations on them, and write the DAX (Data Analysis Expressions) measures required for the reports. The measures

written are the sum of redundant revenue, the sum of shipped revenue, the sum of redundant volume, the sum of shipped volume, and others.
In addition, two further tables are added to the model. The first is the product table, which is shared with the rest of the sales force project and is useful for quick filtering according to the product being displayed; it is therefore taken from the database and added to the model.
The second is the RLS table, which is necessary in order to be able to subsequently filter the reports on a row level so that each user sees the data pertaining to him or her. The table contains information such as account, area of responsibility, manager, coordinator, supplier, and department. The value of the tabular model is that you can directly connect to it from the Power BI service (the cloud repository of the reports). Create your reports up there, and it's a live connection that automatically refreshes whenever you update your model on the back end. As soon as the tabular model is ready, it is deployed to the development instance on Azure Analysis Services and security roles are applied here, which filter the data against the RLS table and the areas of responsibility using filters in DAX. When this is done, the model on Azure Analysis Services can be easily hooked up by the reporting tool, PowerBI.

5.1.3 Azure Data Factory

Azure Data Factory is a serverless cloud-based data integration service that enables the creation of data-driven workflows in the cloud that orchestrate and automate the movement and transformation of data at scale [14]. It contains a number of interconnected systems that provide a complete end-to-end platform, and covers the following aspects:

- **Ingestion**: the system is responsible for ingesting data from various sources and includes activities such as extracting, moving, and copying data from source systems to the ADF. This is possible thanks to more than 100 native connectors.

- **Control Flow**: orchestrates the workflow of data processing activities across pipelines. It defines the logical sequence of activities, dependencies, and conditions for the transformation and movement of data within the data factory.

- **Data Flow**: defines how data is transformed from the source format into a format suitable for analysis and reporting and may include transformations, aggregations, and other data manipulation operations.

- **Monitoring**: provides information on the status and performance of data flows by tracking pipeline execution, allowing problems to be identified and resolved and performance to be optimized.

There are 6 concepts that constitute the main components of each project:

- **Pipelines**: constitute the fundamental elements of ADF and define the flow of data and activities from source to destination. A pipeline represents a logical grouping of several activities that are executed in sequence or in parallel.
- **Activities**: represent the individual data processing or control tasks within a pipeline. There are different types of activities, including data movement activities (e.g. the data copy activity), data transformation activities (e.g. the data flow activity), and control flow activities for conditional logic and branching.
- **Datasets**: are representations of data structures and may represent data sources, destinations, or intermediate data storage. These define the format, schema, and location of the data used by the activities in a pipeline.
- **Linked Services**: these are used within data sets and activities to establish data connections and represent connections to external data stores or computing resources. They provide the information necessary for ADF to connect and interact with data sources and destinations.
- **Data flows**: let ETL processes to be designed and executed to transform data from source to destination.
- **Integration runtimes**: are processing environments, that serve as the intermediary connecting activities and linked services, creating the environment in which data integration tasks and data flows are executed. There are several integration runtimes supported by ADF and their choice depends on factors such as data location and processing requirements. Every data factory instance comes with its built-in integration runtime, known as *AutoResolveIntegrationRuntime*. Additionally, there is another type of integration runtime called *Self-Hosted Integration Runtime*, which is designed to ensure secure access to private networks. This is particularly useful for accessing data sources situated within an Azure virtual network, such as file systems, SQL server instances, and ODBC-compliant database services.

The diagram 5.2 shows the relationships between pipelines, activities, datasets, and related services.

Data Factory constitutes the third block of the architecture in chronological order of development. After the tabular model has been successfully built, it is necessary to orchestrate the entire data pipeline that starts from the sources and lands within the tabular model. The protagonist of this phase is the Data Factory,

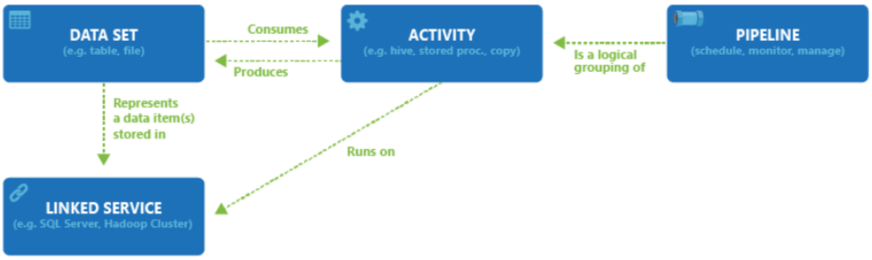

DATA SET
(e.g. table, file)
Consumes
Produces
ACTIVITY
(e.g. hive, stored proc., copy)
Is a logical grouping of
PIPELINE
(schedule, monitor, manage)
Represents a data item(s) stored in
Runs on
LINKED SERVICE
(e.g. SQL Server, Hadoop Cluster)

Figure 5.2

which automates the ETL phase, from the row data to the prepared data within the tabular model. Thanks to the creation of numerous pipelines called up within each other, it was possible to manage the creation of the staging area for each data mart, as well as the loading onto the data warehouse and the updating of the tabular data model in a modular manner. This made it possible to orchestrate the entire ETL process for the data warehouse and to automate its flow on a scheduled basis.

5.1.4 Power BI

Power BI is a unified and scalable platform for enterprise and self-service business intelligence, reporting, and data visualization. It provides interactive data visualizations with a graphical user interface, enabling users to create highly advanced reports and dashboards using DAX language [15]. The latter is a key component of the platform as it allows the creation of calculated columns, customized measures, complex aggregations, temporal analyses, and more. PowerBI constitutes the last piece of our architecture. Being able to hook directly into the tabular model published on an instance of Analysis Services, it is possible to build visualizations and reports on top of the model. The visualizations reflect the customer's needs that emerged during the requirements analysis, and their creation is facilitated thanks to the calculated measures that have been included in the tabular model. In addition to this, Power BI offers other key functions available to users:

- Input data can come from a variety of data sources, including basic Excel spreadsheets, CSV files, databases (SQL Server, MySQL, Oracle, etc....), and cloud-based and on-premise applications.

- It is optimized for handling huge amounts of data; for example, it can process datasets containing more than 100 million rows.

- It contains built-in machine learning functions that can be used to analyze data and help users identify important trends and make more informed predictions.

- It offers an intuitive and easy-to-use interface that makes it easy to navigate complex spreadsheets.

- It features an extreme level of customization that lets users create dashboards to access the data they need quickly and in the format that best suits them.

Power BI consists of two main components: Power BI Desktop and Power BI Service, each with its own features and functionalities.

PowerBI Desktop

Power BI Desktop is a free application that can be installed on a local PC and allows users to connect, transform and visualize data from a variety of sources, while also being able to integrate these sources into a single tabular data model. The creation of this model often requires the transformation of the data itself, a process facilitated by the integration of Power Query Editor in Power BI Desktop. This editor is a tool for importing, transforming, and cleaning up data. It offers a wide range of operations, such as editing data types, removing columns, splitting columns according to custom rules, aggregations, and combining data from different tables based on common fields, among many other options. It is often used for the data preparation phase, after which, the data model is ready to be used to create visualizations and summaries of the information contained within it.
The Power BI Desktop interface is extremely intuitive and lets users to drag and drop fields of interest directly onto the model, allowing them to easily create customized graphical representations that meet their specific needs. Finally, once the report is complete, it can be published on Power BI Service, making it accessible to anyone in the organization with a Power BI license.
Thus, the desktop version is the one used to create the reports, and once they are concluded they are published to the cloud in "Power BI Services," the SaaS version of PowerBI, in the form of interactive dashboards.

PowerBI Service

Power BI Service is Power BI's dynamic cloud-based platform, which opens the door to advanced data analysis and sharing. Here, users can explore and interact with high-quality dashboards and reports created through the Desktop version [16]. Although modeling capabilities may be limited compared to its desktop counterpart, Power BI Service offers an environment without memory and speed issues. This means users can create new reports and share information efficiently without worrying about traditional restrictions related to local resources. It is a place where data come to life and become collaborative decision-making tools, unifying the power of analytics and the flexibility of the cloud. In turn, these dashboards published on the cloud are linked to other dashboards on the client's web portal, from which end users can finally use the reports.

[1] The Venn Diagram has been taken from the official Microsoft documentation: https://learn.microsoft.com/en-us/power-bi/fundamentals/service-service-vs-desktop

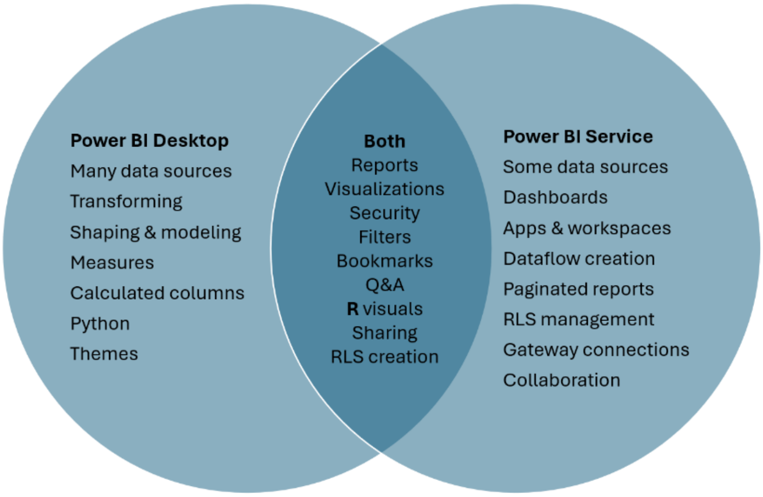


Figure 5.3: Feature comparison between Power BI Desktop and Power BI Service [1]

5.2 ETL Design

This section will show in detail the different activities implemented in the Data Factory to execute the ETL procedure, as well as a description of the stored procedures written in SQL Server. A stored procedure represents a database object that encapsulates one or more SQL statements in a single block of code. This can be executed by calling the stored procedure when necessary, thus favoring code reuse and performance, as they are generally precompiled and cached by the server.

5.2.1 Data Extraction & Staging Area

Extracting data from the source is the first step in the data flow. Considering that the customer's request is to have weekly data updates, the extraction technique used at the beginning of the ETL process is full extraction. In fact, since there is no need to have the data updated frequently, full extraction is preferred over incremental extraction. This is because the former provides greater simplicity and reliability while saving time and resources. The staging tables, as well as the data mart tables, are therefore emptied and repopulated every week. The image 5.4 shows the pipeline implemented in the Data Factory to move data from the source to the staging area of the development database.

The first activity is the execution of a Stored Procedure in the database that keeps track of the steps in the process. This passes the ID of the pipeline on the DF and the type of operation as parameters and writes information such as the start and end date, operation name, status as 'work in progress', pipeline ID, and data mart name to the log table. Log writing is present before the start of each process and is useful when debugging the system.

The second activity is a copy activity, which is used to move data from the source to the staging area. The data source is represented by a table in the SAP HANA database of the customer. In order to get access to this table, a SAP HANA linked service is created, suitable for the ingestion of data from that type of database. It runs on the Azure Data Factory self-hosted integration runtime. This is a snippet of code in JSON that creates a link between the HANA environment running in its virtual network and the ADF running in the cloud in a call service. The ADF self-hosted integration runtime needs to run in a Windows Virtual Machine(VM[2]) and needs the SAP HANA OBDC driver installed in it. The VM must be in the same virtual network with SAP and able to connect through the driver.

After the linked service has been configured, a dataset of type SAP HANA is created and linked to the target table. The dataset is then passed as a data source

[2] A Virtual Machine is a software-based emulation of a physical computer that permits multiple operating systems and applications to run on a single physical hardware host.

within the copy activity, allowing an extraction query to be executed in the SAP database. The target table is set to be the staging area table created in the database to host the raw data. This exploits a linked service with the Azure SQL Database to be created, as well as a dataset linked to the staging area table of the data mart. After the data source and destination have been set, the mapping between the columns of one and the other must be defined, necessary for correct data entry in the destination. In addition, a pre-copy script is executed that empties the destination table before it receives the data from the source so that the data are always consistent.
In the event of a positive extraction result, the 'LOG OK' procedure is executed, which updates with the status 'OK' the row associated with the operation in the log table, contrary it is updated with the status 'KO'.

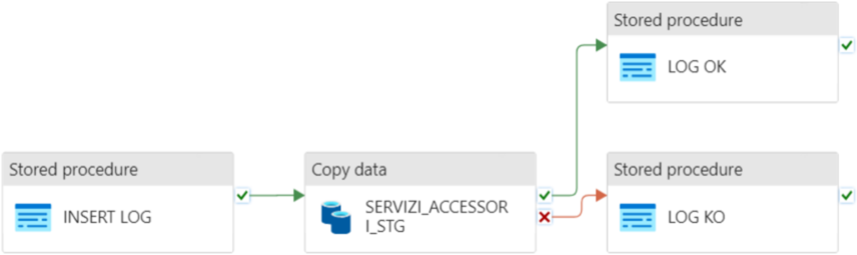


Figure 5.4: Data mart staging area

5.2.2 Data Transformation & Loading

The Operating Data Store (ODS) area, in a data warehousing process, represents the area where transformations, cleansing and aggregations are performed on the data arriving as input from the staging area. This part of the process is the most resource-intensive, as it includes a number of operations such as: **data quality control**, in which only the fields of interest are selected; **normalizations**, for which data or unit conversions are performed; **aggregations**, as data may arrive with a different level of detail, so in these cases it is necessary to perform aggregations to make them uniform; **filtering of outliers**; and **simplification of the data**, such as changing column names so that they are more intuitive.
For the creation of the data mart, a pipeline was created containing 3 activities that execute the stored procedures for the creation of the dimensions Calendar, Client, Product, and Sales Network. This activity uses the linked service to connect to the Azure SQL Database and execute the procedure specified in its parameters. The fact table is then populated via another stored procedure on the success of these activities. The stored procedures all share the same structure:

- Creates a new log of the current operation and sets it as 'WIP'.
- Starts a try block where the table undergoing the operation is emptied. This is to ensure that the data in the table is always the latest available.
- Run the query that extracts the data from the source table in the staging area to the target table in the data mart, making the necessary data transformations.
- Update the operation in the log with 'OK' status, and if any error occurs the catch block will set the operation in the log as 'KO' and save the error message.

Regarding dimensions, they are all shared with the main data mart, that of the sales force, as they act on the same MIPA market segmentation. Therefore, for the creation of the calendar, customers and sales network, it was sufficient to select the necessary columns from the same dimensions of the main flow and perform some transformations to align them with the current data mart. In each dimension, a natural key has been identified that goes to uniquely identify each record in the tables, which cannot, therefore, contain duplicates. In the case of clients, the natural key is represented by the unique business name ID for each customer. Similarly, for products, it is the product code, and for the calendar, it's the daily date, and so forth. For the creation of the fact table, we started from the staging table populated by SAP HANA, performing operations called 'left join' on the different dimensions. The "left join" is one of the most common operations in SQL and is used to combine data from two tables based on a link key, preserving all records from the left table (the fact table) and only matching records from the right table (the dimension table). This is convenient compared to a full join as it would lead to a large sparsity of data as the transactions do not necessarily present all the information in the various dimensions. The first approach is therefore preferred, whereby the dimensions are only matched for rows within the fact table. In detail, are performed two left joins to the same customer dimension, but using two different link keys. Once using the field "tax code" and another time using the field "VAT number". The aim is to obtain complete and accurate customer data, considering that some customers may be identified by their tax code, while others by their VAT number. Then a left join is performed to the dimension 'sales network' using the name of the sales portfolio as a key. This information can be obtained via the previous join to the customer dimension, as the latter contains the name of the sales portfolio associated with the customer. This step can be useful for linking information on sales areas to customer data. Finally, a left join is performed to the 'product' dimension based on the 'sales product' field. This can be useful for linking information on products sold to the data in the facts table. All these operations enable the fact table to be enriched with additional data from different dimensions to create a complete dataset for analysis. Several strategies were adopted to ensure the integrity and usability of the data. The first is the elimination of null values, with a filtering process in which rows with revenue and volume amounts of zero are excluded. In addition, work has been done to standardize the data through replacement and merging operations. One example is the creation of a new date column, which was originally absent in the source data. To generate this column, the year, month, and week were used, assigning each row the date corresponding to the first day of that week in that month and year. This column is critically important, as it allows each event to be contextualized in relation to all the other dimensions involved in the analysis.

5.2.3 Refresh of tabular model

Right after the pre-processed data is loaded into the data mart, it becomes necessary to initiate a data refresh within the tabular model directly linked to the tables in the Azure SQL database. This step is crucial as refreshing it guarantees that the data remains as accurate and current as possible, given that PowerBI's reports directly query the data directly from this tabular model. The image below in figure 5.5 shows the pipeline with which it has been implemented.
Firstly, it is needed to give Azure Data Factory access to the Azure Analysis Services (AAS) server, in order to perform these operations using managed service identities. This can be realized by setting the Data Factory instance as a server administrator in the AAS, a role used to grant server-level security privileges to a user or group of users. To achieve this, you have to set the security user in the AAS in the following format: "app:*TenantId*@*ApplicationId*". Both of these IDs are specific to the Data Factory instance. This configuration facilitates seamless communication between the two systems, ensuring the prevention of security errors. Within the pipeline in the Data Factory, through the use of Web activity, it was possible to make a Post call via Rest API to the table model on the Analysis Service endpoint, ordering a data update. In fact, using any programming language that supports REST calls, it can be performed asynchronous data update operations on Azure Analysis Services table models. The REST API endpoint is set to:

```
@concat(’https://’, pipeline().parameters.Region, ’.asazure.windows.net/’,
        servers/pipeline().parameters.ServerName, ’/models/’,
        pipeline().parameters.ModelName, ’/refreshes’)
```

The parameters set needed are the region of use, in this case, West Europe, the name of the server and the name of the tabular model. The parametrization makes the pipeline reusable with different models. The body of the Post call instead is set to:

```
    @concat(
'{
    "Type": "',pipeline().parameters.RefreshType,'",
    "CommitMode": "transactional",
    "MaxParallelism":10,
    "RetryCount": 2,
    }'
)
```

The parameter *RefreshType* is configured as *Full* indicating that data refresh and recalculation of all dependents will be performed for all partitions within the specified partition, table, or database. Managed identities (MSI) are used to access Azure Analysis Services Rest API. Since the method is asynchronous, when you trigger a refresh of a model, you get the response of the REST API call, but not the final status of the refresh. Thus, the result of the update operation is evaluated and the pipeline waits for a positive state, by iterating on the status variable until it is positive, before proceeding to write the log.

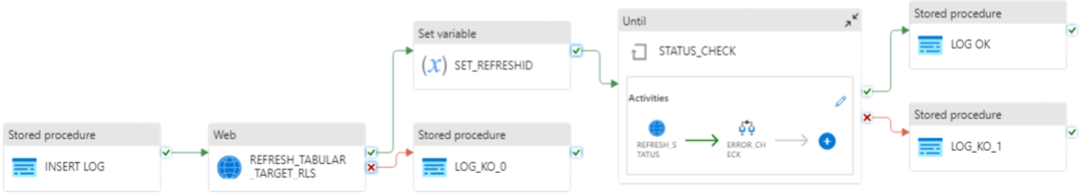


Figure 5.5: Refresh of tabular model

5.2.4 Data Mart ETL

Figure 5.6 shows the pipeline designated to coordinate the ETL process of the data mart to perform both the creation of the tables in the data warehouse and the refresh of the tabular model. The order of execution is based on the outcome of 'traffic lights' set to coordinate the main ETL flow together with the auxiliary data marts. These traffic lights are nothing more than logs saved within the database, which are checked within pipelines before certain activities in order to understand whether they can start or not. It's indeed crucial that the main ETL flow is successfully concluded before the ETL flows of the various data marts begin, as many of them create their own dimensions based on the main ones.
On the check of a positive outcome of the main 'traffic light', through a LookUp activity, the two pipelines described in 5.2.2 and 5.2.3 are executed, thus loading the data in the warehouse and refreshing the tabular model. At the end of the two processes, the 'traffic light' of the data mart is updated with the 'OK' status.

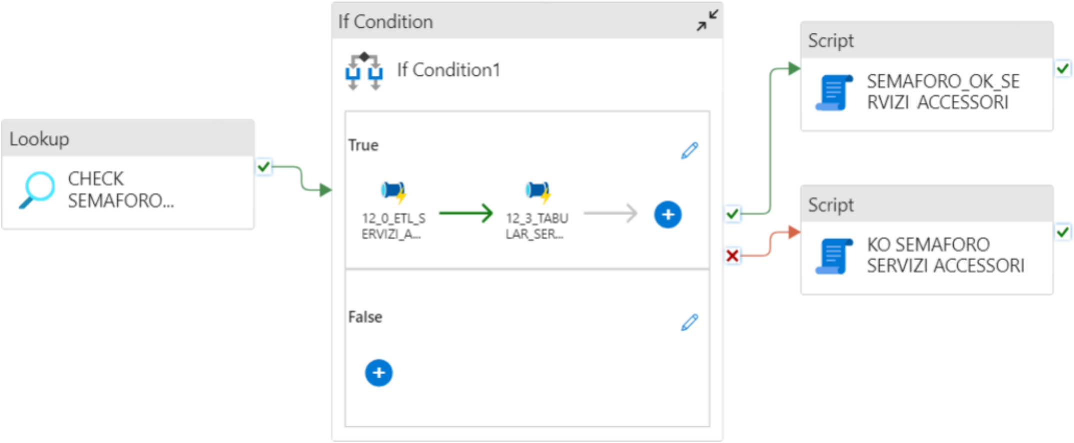


Figure 5.6: Data mart ETL

5.2.5 Data Warehouse ETL

Figure 5.7 shows how the pipeline was set up to orchestrate the ETL of the entire project. First of all, a pre-staging SSIS package is run to configure some initial settings before the start of the process and upload the staging areas of the main flow. On the success of the latter, a procedure is executed to scale up the database. Leveraging the pay-as-you-go model, we are able to increase the number of Database Transaction Units (DTU) of the Azure SQL database up to 8 times, passing from an S2 to an S6 computational level. This is necessary to speed up the ETL process as there are many data extractions and transformations acting in parallel, requiring high computational capacity.
As soon as the database has been scaled the staging area pipelines of the data marts are executed. Once the semaphore is successful, move on to the creation of tables in the data warehouse for the main flow, which is necessary as some data marts base dimensions on its tables. On the success of this and via the semaphore control, the ETL pipeline for each data mart is executed.
In turn, the pipeline just described is incorporated into a further one, which is needed to downscale the database at the end of the process or in the event of any errors within it, thus also updating the error history in the latter case.

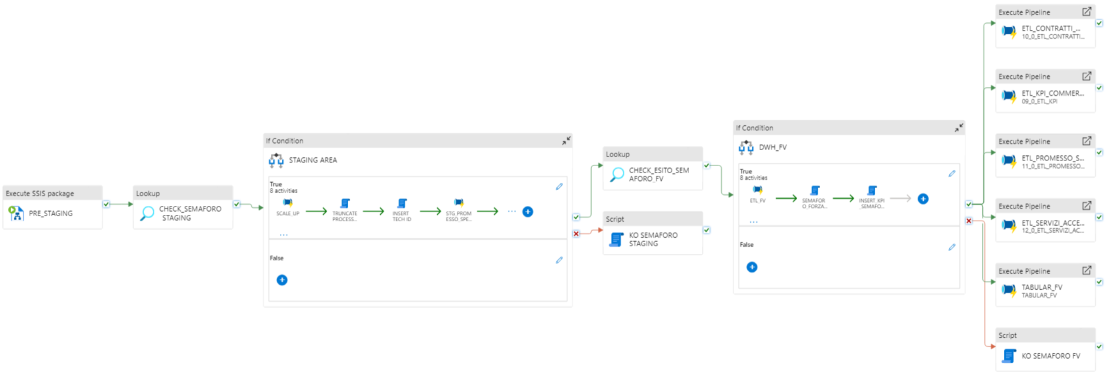


Figure 5.7: Data Warehouse ETL

5.3 Project Delivery

Upon completing the solution in various development environments, the subsequent phase involves transitioning to the testing environments. This entails replicating the tables, functions, and stored procedures from the development server to the test server. The same process applies to the ETL flow within the Data Factory and the tabular model within Analysis Services. To manage version control for this solution, we make use of GitLab, a web platform that specializes in managing Git repositories. This tool permits us to track changes, coordinate work among team members, and ensure accurate version control.

We begin by creating a directory that houses all the files associated with each piece of software in each data mart. For example, for the Data Factory part, we store the '*arm_template.json*' file, a JSON document that defines the infrastructure and configuration of the project. It includes associated resources, such as linked services, datasets, pipelines, and triggers, in a structured and reproducible manner. For SSAS and SSIS, we store '*.sln*' files that contain the relevant projects. Regarding Azure SQL Server, we store all the necessary tables (both staging and target), functions, and stored procedures. Concerning visualizations, we store the '*.pbix*' files that contain the reports. After setting up the new solution, we initiate the process by establishing a dedicated branch. We then transfer the local repository copy, complete with all the relevant files, to this newly created branch. Following this, we commit the changes to document our modifications, and subsequently, we submit a pull request, designating the newly created branch as the primary source. In a second step, through Jenkins, a DevOps software of fundamental importance for building, testing, and releasing the project, we pull the solution from the GitLab repository. Jenkins enables continuous integration and continuous delivery, greatly easing the deployment process. By creating a custom workflow on Jenkins and configuring the necessary parameters, we automate the process of deploying the solution from the GitLab repository to the testing or production environments. The solution, once deployed to the production environment, becomes readily available to end users, following a weekly schedule. Typically, during the weekend, we perform the ETL process, which involves filling the data warehouse tables with updated data after emptying them. This is handled efficiently by creating specific scheduled triggers within Azure Data Factory, ensuring an efficient and continuous workflow.

Chapter 6

Dashboards

The following chapter will describe the entire reporting part of this project, with the respective dashboards, implemented entirely through PowerBI Desktop, and then deployed to the cloud through Power BI Service. Reporting is a key part of the entire process implemented, as it is the final product that is delivered to the client, which will be used to support decision-making and marketing analysis. The client specified the need for a set of filters within each dashboard that can be combined together interactively to gain different insights from the data. Specifically, the filters are as follows:

- **Area**: refers to the areas dedicated to the commercial management of logistics and parcel and specialized products/services by industry. These are the same as those described in section 3.1.2 and differ based on the category to which the products sold belong.
- **Portfolio**: specifies the name of the sales portfolio to which the product belongs.
- **Portfolio type**: specifies, within each industry, to which category the product sold belongs (e.g. Health & Beauty or Electronics and Informatic) and the parcel typology (e.g. Large, Top).
- **Coordinator**: this filter is useful for managers viewing reports to be able to monitor the sales performance of salespeople under each coordinator.
- **Business**: used to distinguish between the various business segments of the Post Office, however in the case of the following data mart the segment is always CEP, as the data only cover courier and parcel services.
- **Service**: explicates the shipping service to which the auxiliary service is connected (e.g. Extra Large, Poste Delivery Business).

- **Auxiliary service**: this filter is used to select a particular ancillary service that you want to monitor. Some of these may be delivery to inconvenient postcodes, scheduled delivery, overweight, and so on.
- **Time**: is used to select the desired time interval of the data.

On the top of the dashboard are the filters described above. These are created using the visual object called data filter, and in turn, the data entered in the filters are filtered out so as not to report options for which the data have null values and which have the value of the 'row count' measure greater than zero. This makes it possible to select options from the filters that actually have data below them. Then these filter objects are assigned to the various dashboard views so that the data within them change as the filters change. The dashboards shown below have January 1, 2023, as the lower limit of the time filter, so as to show the situation in the current year. For this reason, the time axis stops at October, as later data are not yet present.

6.0.1 Dashboard: Earnings

Figure 6.1 shows the first dashboard requested by the client. It provides a comprehensive overview of earnings generated by the services over time. The first visual element, located at the top left, is a data card designed to display a single value. In this instance, it presents the measure used to calculate the total earnings from all services. Next, there's a line plot that utilizes the same measure to illustrate the month-by-month earnings trend. Finally, at the bottom, there's a horizontal bar chart that facilitates the comparison of revenue sums between the first and second halves of the year. Given that the data extends up to September 2023, it's evident that earnings are significantly higher in the first half of the year.

6.0.2 Volumes

The second dashboard shown in Figure 6.2 faithfully emulates the style and visual elements of the first one, but it focuses on volumes rather than revenues. This is crucial for the customer's ability to track volumes, as both costs and revenues can be influenced by them. The visualizations within this dashboard illustrate the total volume shipped, the volume trends over the months, and a comparison of volumes between the first and second semester months.

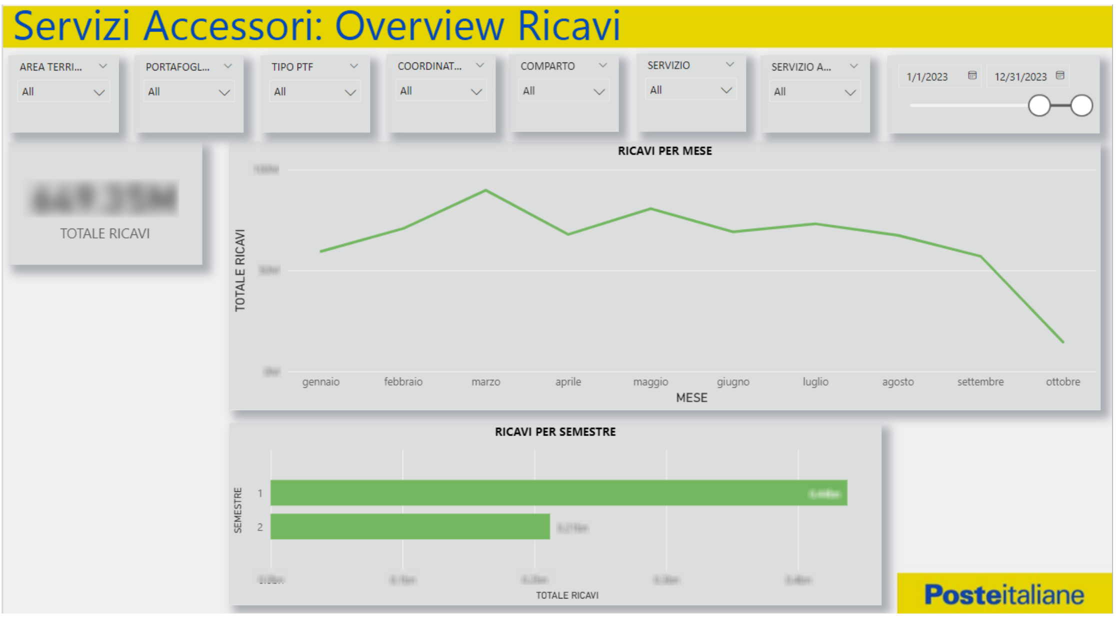


Figure 6.1: Dashboard 1: Earnings Overview

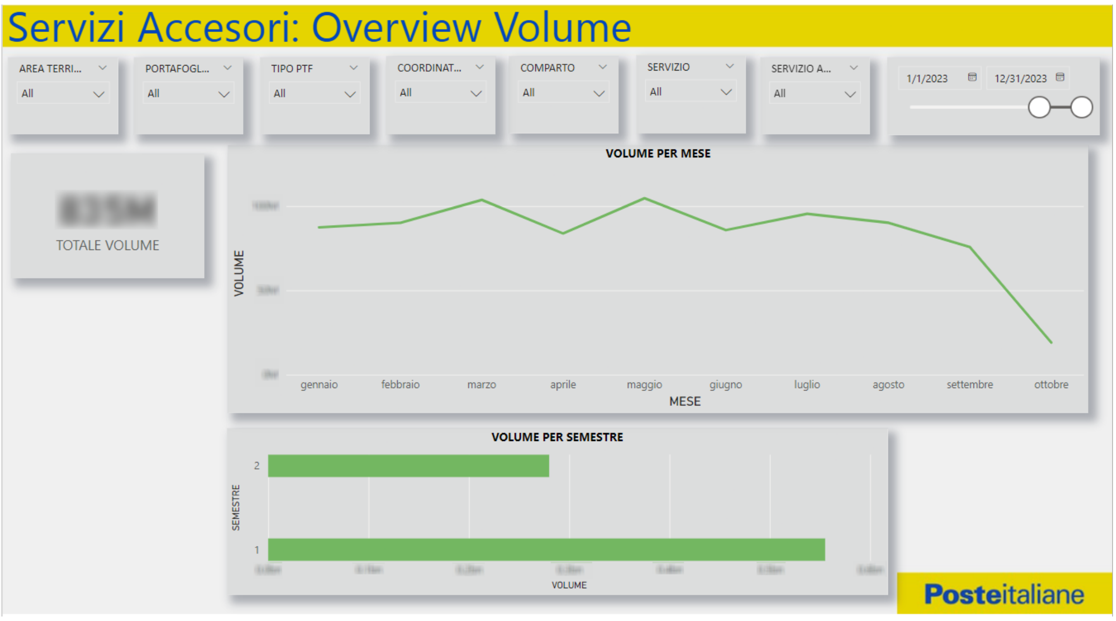
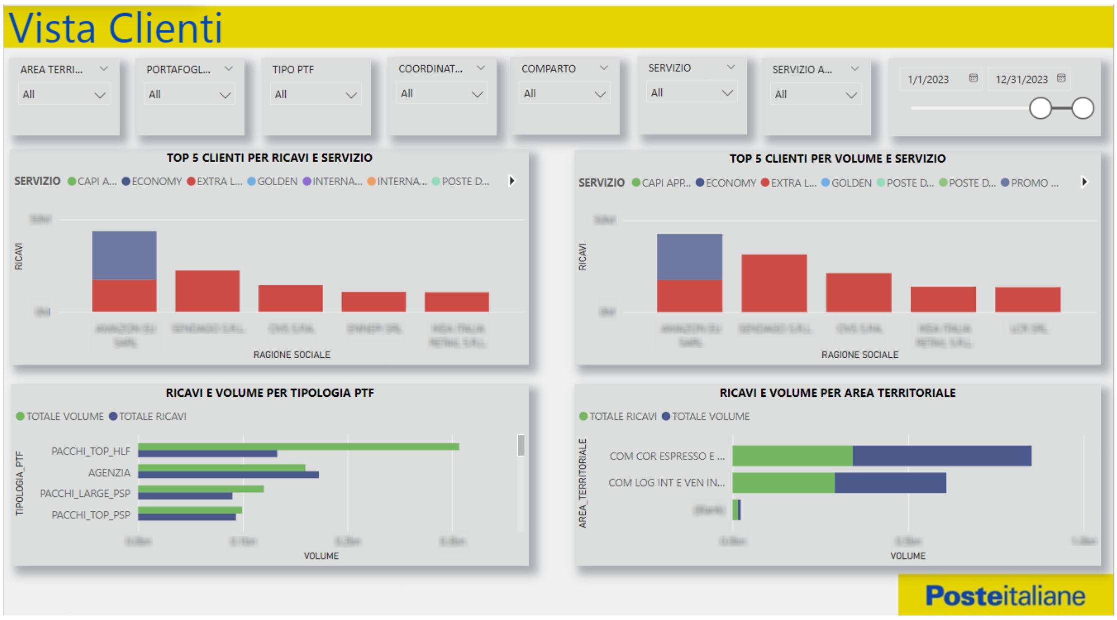


Figure 6.2: Dashboard 2: Volumes Overview

6.0.3 Clients

The dashboard in figure 6.3 is useful for managers to monitor what are the top customers. The chart in the upper left-hand corner consists of stacked vertical bars showing each customer's total revenue. In addition, the legend associated with this chart provides information about the services by highlighting the percentages of revenue attributed to each service through different colors within the bar. The same type of graph is used in the top right chart that aims to identify the top 5 customers, shifting the focus to volumes. On the bottom left, there is a horizontally grouped bar chart that shows the total volumes and revenues for each service category. The last chart on the bottom right is a horizontal stacked bar and shows total revenue and volume by area.



Vista Clienti

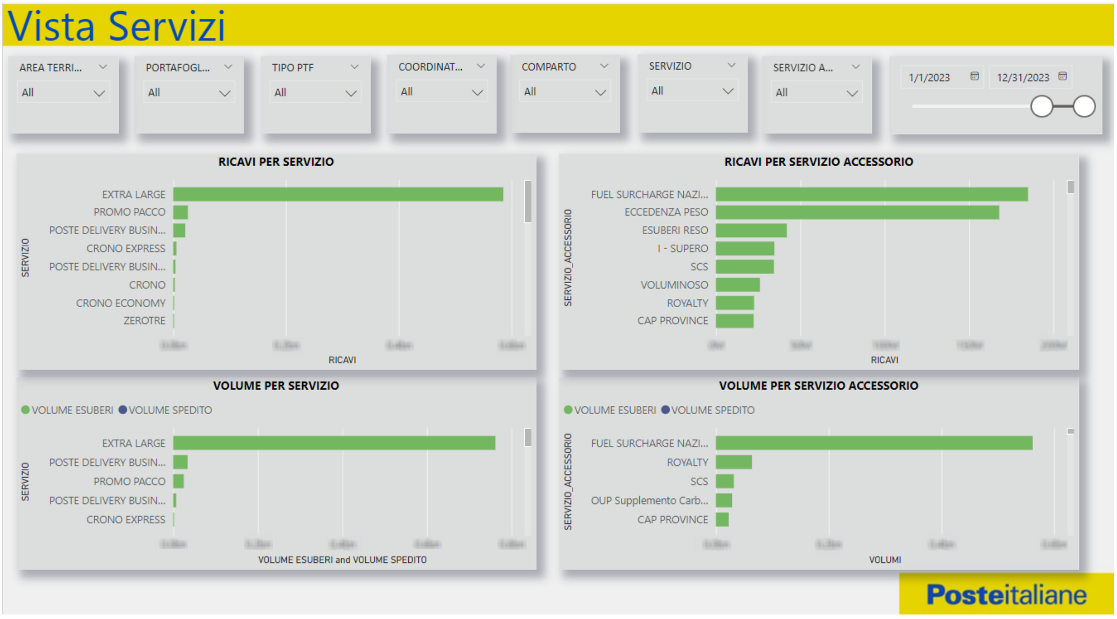
AREA TERRI... All
PORTAFOGL... All
TIPO PTF All
COORDINAT... All
COMPARTO All
SERVIZIO All
SERVIZIO A... All
1/1/2023 12/31/2023

TOP 5 CLIENTI PER RICAVI E SERVIZIO
SERVIZIO ● CAPI A... ● ECONOMY ● EXTRA L... ● GOLDEN ● INTERNA... ● INTERNA... ● POSTE D...
RICAVI
RAGIONE SOCIALE
TOP 5 CLIENTI PER VOLUME E SERVIZIO
SERVIZIO ● CAPI APP... ● ECONOMY ● EXTRA L... ● GOLDEN ● POSTE D... ● POSTE D... ● PROMO ...
RICAVI
RAGIONE SOCIALE
RICAVI E VOLUME PER TIPOLOGIA PTF
● TOTALE VOLUME ● TOTALE RICAVI
TIPOLOGIA_PTF
PACCHI_TOP_HLF
AGENZIA
PACCHI_LARGE_PSP
PACCHI_TOP_PSP
VOLUME
RICAVI E VOLUME PER AREA TERRITORIALE
● TOTALE RICAVI ● TOTALE VOLUME
AREA_TERRITORIALE
COM COR ESPRESSO E ...
COM LOG INT E VEN IN...
VOLUME
Posteitaliane

Figure 6.3: Dashboard 3: Clients

6.0.4 Services

This dashboard in figure 6.4 is useful for managers to compare revenues and volumes between the service and the associated auxiliary services. The first graph in the upper left shows, in order, for which services you have the highest revenues. The next one shows, in order, for which auxiliary services have the highest revenues. Bottom left, in order, for which services you have the highest volumes. The last graph shows, in order, which auxiliary services have the highest volumes.



Vista Servizi

AREA TERRI... All

PORTAFOGL... All

TIPO PTF All

COORDINAT... All

COMPARTO All

SERVIZIO All

SERVIZIO A... All

1/1/2023 12/31/2023

Figure 6.4: Dashboard 4: Services

6.0.5 Earnings Volumes Table

This last dashboard in figure 6.5 does not contain any graphs, rather there is a table-type view that allows users to inspect thoroughly all the information about sales referred to any particular customers in detail. Indeed, each row contains information such as the client, region, portfolio type, service, and auxiliary services. On the other hand, the metrics consist of surplus revenue, shipped revenue, surplus volume, and shipped volume, as well as the overall total revenue and total volume.

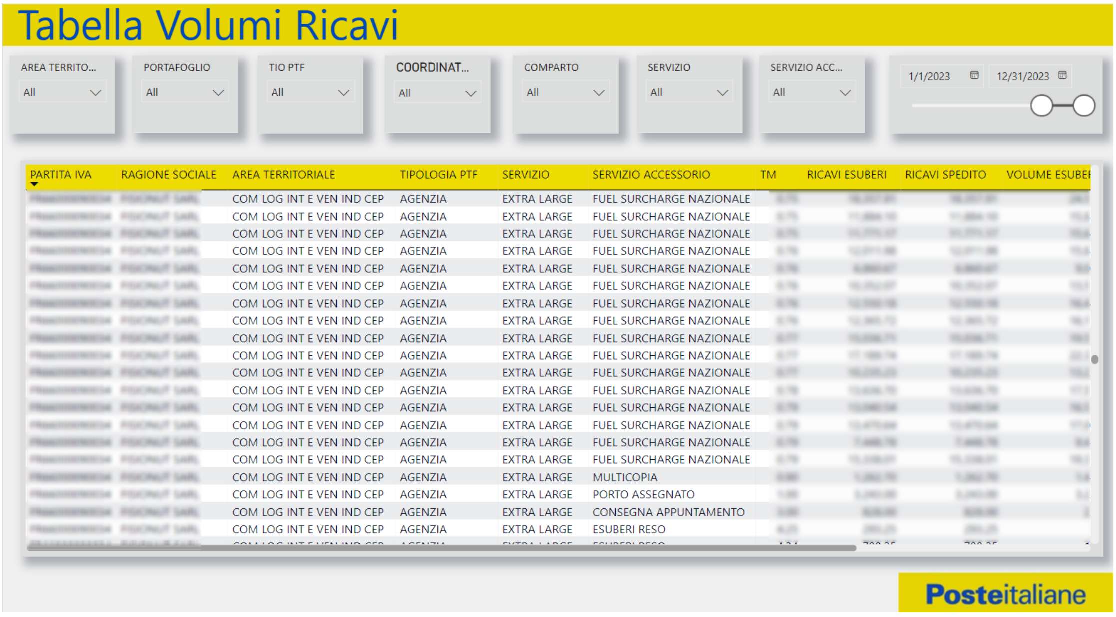


Tabella Volumi Ricavi

AREA TERRITO... All

PORTAFOGLIO All

TIO PTF All

COORDINAT... All

COMPARTO All

SERVIZIO All

SERVIZIO ACC... All

1/1/2023 12/31/2023

PARTITA IVA	RAGIONE SOCIALE	AREA TERRITORIALE	TIPOLOGIA PTF	SERVIZIO	SERVIZIO ACCESSORIO	TM	RICAVI ESUBERI	RICAVI SPEDITO	VOLUME ESUBEI
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	FUEL SURCHARGE NAZIONALE	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	MULTICOPIA	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	PORTO ASSEGNATO	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	CONSEGNA APPUNTAMENTO	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	COM LOG INT E VEN IND CEP	AGENZIA	EXTRA LARGE	ESUBERI RESO	[redacted]	[redacted]	[redacted]	

Posteitaliane

Figure 6.5: Dashboard 5: Earnings Volumes Table

Chapter 7

Conclusions

The increase in data production, combined with technological progress and increasing digitization, will make analytical tools a key factor in business competition and a powerful means of supporting organizations' strategic choices. Similarly, the company Poste Italiane, which is strongly linked to the Italian social fabric, has been undertaking a real technological transaction for many years now, focusing more and more on digital innovation and being data-driven. Surrounding itself with consultants and technologies capable of extrapolating insights from useful information to verify the company's performance against set objectives, efficiency in resource allocation, and process optimization. In this context my thesis project was carried out, thanks to my company I was able to participate in the creation of a Business Intelligence system aimed at analyzing the sales of auxiliary services belonging to the courier and parcel services of Poste Italiane.

The focus of the work was not so much on the type of analysis performed and the interpretation of the results, but rather on the approach to the creation of a system capable of sustaining itself and creating value from the data stored on the client's systems. In fact, the interpretation of the results is left to the end-users, i.e. managers and vendors, who are able to draw data-driven insights from them. The solution was meticulously built by adhering to a software lifecycle methodology, which begins with the design phase in development environments and culminates with implementation in production after successful testing. The choice of the technologies exploited was constrained by the integration of the system within a larger, pre-existing BI data flow, thus leading to the inheritance of the tools used. From the point of view of the implementation, the system combines a high level of versatility both in immediate use and in view of the future, exploiting the benefits of the cloud services offered by Microsoft, so as to be able to accommodate modifications or enrichments in the best possible way.

The project adhered to the traditional process of developing such a system, starting with the collection of customer requirements. This initial phase is essential for a complete alignment with the client's needs and objectives, thus avoiding costly modifications and late revisions. We then proceed to map the information obtained with a relational data structure and the consequent creation of the data mart, hosted within the Azure SQL Database, which was constructed in order to best support the required analyses. The entire process was then orchestrated through the use of Azure Data Factory, which, thanks to its many integrations with various services in the data sphere, made it possible to automate the ETL process by extracting data from the source systems and transforming it into the target destination. The ETL was scheduled weekly, thus bringing new data into the database and updating reports published on the cloud. This approach allows the end user to benefit from weekly performance analysis for the preceding week while also providing an overview of performance spanning multiple years. Regarding possible further developments of the system, it is likely that the client may request changes or improvements to the solution itself, such as the addition of new KPIs of interest, in the future. Looking instead at other possible enrichments of the system, one can certainly come from the exploitation of artificial intelligence. Through the use of its algorithms on business data, the dashboards may provide increasingly accurate answers as to which is the right choice to make, assisting users in their data-driven decisions.

Bibliography

[1] Antonio Albano. «Decision support databases essentials». In: (2013) (cit. on p. 5).

[2] William H Inmon. «What is a data warehouse». In: *Prism Tech Topic* 1.1 (1995), pp. 1–5 (cit. on p. 7).

[3] Il-Yeol Song and Kelly LeVan-Shultz. «Data Warehouse Design for E-Commerce Environments». In: *Advances in Conceptual Modeling: ER '99 Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling, Paris, France, November 15-18, 1999, Proceedings.* Ed. by Peter P. Chen, David W. Embley, Jacques Kouloumdjian, Stephen W. Liddle, and John F. Roddick. Vol. 1727. Lecture Notes in Computer Science. Springer, 1999, pp. 374–387. DOI: `10.1007/3-540-48054-4\_30`. URL: `https://doi.org/10.1007/3-540-48054-4%5C_30` (cit. on p. 8).

[4] URL: `https://www.ibm.com/it-it/topics/etl` (cit. on p. 17).

[5] S M Kumar and Meena Belwal. «Performance dashboard: Cutting-edge business intelligence and data visualization». In: *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. 2017, pp. 1201–1207. DOI: `10.1109/SmartTechCon.2017.8358558` (cit. on p. 18).

[6] Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. «Cloud computing: An overview». In: *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings 1*. Springer. 2009, pp. 626–631 (cit. on p. 19).

[7] URL: `http://aws.amazon.com` (cit. on p. 19).

[8] URL: `http://www.microsoft.com/azure/` (cit. on p. 20).

[9] URL: `https://advant.it/` (cit. on p. 26).

[10] URL: `https://www.posteitaliane.it/it/comunicati/posteitalianeemic-1476517871030.html` (cit. on p. 27).

[11] URL: `https://www.posteitaliane.it/it/strategia-omnicanale.html` (cit. on p. 28).

[12] URL: https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql (cit. on p. 42).

[13] URL: https://learn.microsoft.com/en-us/analysis-services/ssas-overview?view=asallproducts-allversions (cit. on p. 43).

[14] URL: https://learn.microsoft.com/en-us/azure/data-factory/introduction (cit. on p. 44).

[15] URL: https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview (cit. on p. 47).

[16] URL: https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-service-overview (cit. on p. 48).