NOVA IMS Information Management School

MGI

Master Degree Program in
Information Management

Design of an Independent Data Mart for the Asset Management Area in the Airport Industry

Marco Antonio Morán Velásquez

Project Work

presented as partial requirement for obtaining the Master Degree in Information Management

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

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

DESIGN OF A DATA MART FOR THE ASSET MANAGEMENT AREA IN THE AIRPORT INDUSTRY

by

Marco Antonio Morán Velásquez

Project Work presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Knowledge Management and Business Intelligence.

Supervised by

Supervisor: Bruno Jardim, PhD, NOVA Information Management School

July 2024

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Marco Moran

Lisboa, July 2024

ABSTRACT

Airport infrastructure and assets in the aviation industry require meticulous control and analysis due to their continuous operations, with the Asset Management department being responsible for this task. Robust asset management strategies are essential to maintain efficient operations. This study aims to enhance the Asset Management department's capabilities by delivering pivotal information to the team to support data-driven decision-making through the development of a structured data repository, or Data Mart. Currently, the Asset Management area uses an EAM system that provides asset maintenance and operation information in a slow and static manner. To address this, establishing a Business Intelligence or Decision support system environment is proposed, beginning with the creation of a Data Mart. This Data Mart will aggregate and structure data extracted from the EAM system into an OLAP database, enabling comprehensive analysis and presenting actionable insights through interactive dashboards. The primary objectives are to design an efficient data flow, integrate the OLTP system into the Data Mart, establish ETL processes, and deliver the solution to internal clients for enhanced data analysis. This approach aims to optimize decision-making processes, which ultimately will impact the improvement of the performance and lifecycle of airport assets.

KEYWORDS

Asset management; Data Mart; OLAP data base; ETL processes; Structured data

Sustainable Development Goals (SDG):

8 TRABALHO DIGNO E CRESCIMENTO ECONÓMICO

9 INDÚSTRIA, INOVAÇÃO E INFRAESTRUTURAS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

OLTP	On-Line transaction processing
OLAP	On-line analytical processing
EAM	Enterprise Asset Management
ETL	Extract, Transform & Load
OLAP	On-line analytical processing
SSIS	SQL Server Integration Services
SQL	Structures Query language
SSMS	SQL Server Management Studio
KPI	Key Performance Indicator
OLE DB	Object Linking and Embedding Database
EDW	Enterprise Data Warehouse

1. INTRODUCTION

1.1. MOTIVATION

The aviation industry encompasses various sectors, and one of its pivotal components is airport infrastructure, because of its relentless and uninterrupted operations every aspect within airports demands meticulous control and analysis, as disruptions can significantly impact the fast-paced flow of activities.

Airport's primary function is offering services for distinct clients as airlines, providing them the physical facilities for arrivals and passenger connections. These passengers themselves are considered clients of the airports as well, utilizing the infrastructure, while other enterprises operate within airport premises, offering a spectrum of products and services.

To ensure that the airports can offer their services to their clientele, robust asset management strategies are essential. There is one specifical area that has the goal of take care of the assets within the company such as boarding bridges, the baggage handling system and other equipment, this area is called “Asset Management” which means operating a group of assets over the whole technical life cycle guaranteeing a suitable return and ensuring defined service and security standards. (Schneider et al., 2006).

To enhance the endeavors of the Asset Management area, such as, for example, creating strategies for the maintenance of the assets, comprehensive data analysis is paramount. Therefore, the creation of a structured repository for transactional data generated within the Enterprise Asset Management (EAM) system has become imperative. This repository will be an invaluable resource for users, engineers, and analysts within the Asset Management area, enabling them to make well-informed, data-driven decisions effectively.

1.1.1 CREATION OF A DATA MART

As a member of the asset management team within Portugal's leading airports management company, it's clear that there's a lot of data generated about asset maintenance and operation. However, there's currently a lack of complete analysis to optimize usage and improve decision-making. The area currently employs an EAM system called Maximo, but it captures and analyzes asset’s information slowly and statically.

Creating a decision support system will enhance decision-making, thereby optimizing overall operations. The next phase involves establishing a Business Intelligence environment for the area, beginning with data sourced from the EAM system. This data will then be extracted to an OLAP database, enabling the correlation of data for analysis. Finally, this collected information will be presented as actionable insights through interactive dashboards.

There is a need for a database and the first step would be defining the boundaries for data extraction becomes essential. Based on the Kimball methodology a data mart design considers specific business questions, which, in this study's context, represent the research gap. These questions serve as a guide to determine the information that should be capable of addressing:

1. How good or bad is the performance of the assets?

2. What are the costs related to the maintenance of the assets?
3. Does every asset have a life cycle analysis?
4. When to choose between replacement and rehabilitation of an asset?

1.2. Objectives and Methodology

This research aims to design an efficient data flow for managing the Asset Management department's data. The objectives are as follows:

1. Integrate an OLTP system into a data mart.
2. Establish the ETL processes.
3. Deliver this solution to internal clients for further data analysis.

The methodological process for this research involves: defining the business questions through the Goal, Question, Metric approach (Basili et al., 1994), designing the data mart to simplify the OLTP's relational structure into a straightforward database (Moody, 2000), developing the new relational model based on a star schema (Sharda et al., 2018), the Kimball model is used as well and finally deliver the information to the team members through dashboards.

A comparison of the status of the business intelligences developments against the results aimed in this research is shown in the figures 1 & 2.

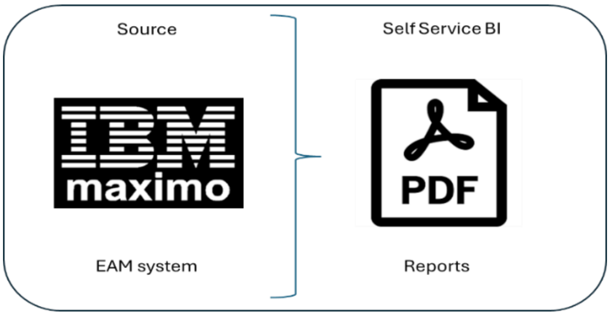


Figure 1 - BI Diagram "as is"

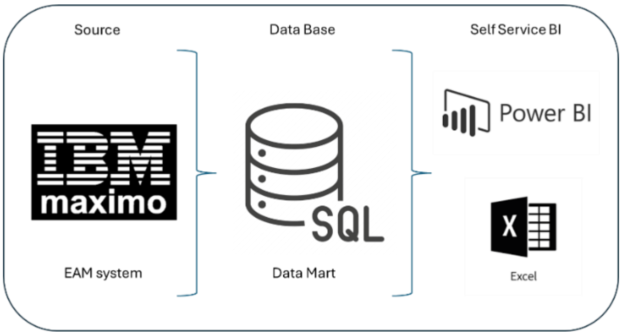


Figure 2 - BI Diagram "to be"

In the next chapters, the study delves into the existing developments regarding the database creation for the asset management area. It will cover the database design, including all the critical decisions necessary for developing the data repository, and provide a step-by-step guide on the development process using the selected ETL software. Chapter 5 explains the results and details of the data products developed for the Self-service BI.

2. LITERATURE REVIEW

In this section the reader will find relevant information that serves as a valuable resource offering insights and guidelines that can be used as a base methodology for developing and designing a database. The literature review acts as a cornerstone, providing a comprehensive understanding of the key considerations and approaches in the process of Database design. Furthermore, the section delves into the explanation of selected techniques, chosen for their alignment with the database's requirements, to craft an applicable data repository.

2.1. INTEGRATION OF DATA FOR ASSET MANAGEMENT

One of the objectives of asset management is to establish a decision support system based on accurate data. This data serves as the foundation for generating meaningful information and enabling effective and efficient decision-making processes (Pantelias, 2005).

Data integration is essential for transforming data into insightful information that supports decision-making at any level of the asset management area.

There are two data integration alternatives (Pantelias, 2005): fused database, this strategy involves creating a single database that contains all integrated data and the second one is the interoperable databases, this approach links different databases together where the integration of data is achieved using queries, providing a view of the linked data.

The study will be looking for a way to recollect and integrate the data into an OLAP (On-Line Analytical Processing) database, meaning that this tool will be developed to allow the multidimensional analysis of the historical data using a OLTP system, where the transactional data is generated, as Garani et al. (2023) states.

When integrating data into a repository, it is important to consider the frequency of data creation to decide the appropriate timing for updating the database. In the case developed by Xiaofeng He et al (2005) a "Power Data Warehouse" was created, specializing in extracting and hosting real-time data. This approach is useful in fast-paced maintenance environments where informed decisions need to be made quickly to avoid operational disruptions. The choice of data extraction and storage methods depends on the specific needs of each area and industry

2.2. THE NEED FOR A DATABASE IN A MAINTENANCE RELATED AREA

To establish a data-driven strategy for asset maintenance and management, it is crucial to gather comprehensive and accurate data related to transactions such as interventions, costs, and asset functionality. This data provides insights into the actual operational conditions of the assets (Duarte et al., 2013).

The database designed for the maintenance area caters to three distinct user categories (Duarte et al., 2013): Risk & reliability Analysts who are in charge of the analysis and prediction of the asset's reliability, maintenance engineers: Measure and optimize the maintenance performance and the component designers who Analyze and optimize the performance of asset components.

The ultimate objective of developing this maintenance database is to transform it into a central component of a decision support management system (Duarte et al., 2013).

2.3. ADVANTAGES OF THE DESIGN AND IMPLEMENTATION OF A DATA MART

In today's companies, large quantities of data are being generated, yet in many cases, this data is not being utilized effectively. To capitalize on the valuable resource that data represents within companies and teams, it is necessary to structure the data and develop a storage solution, as well as actively engage with the data (Jaleel & Abbas, 2020). While data is already being stored in the systems that generate it, accessing and extracting this data can be challenging. It may lack precision, and users often spend significant amounts of time querying the data for on-demand analysis and insights generation. Therefore, the final advantage of implementing a dedicated data mart to an area is to improve and facilitate the decision-making process (Jaleel & Abbas, 2020).

There are also more elaborate decision support systems based on the asset management area. For example, as explained by Osladil & Kozubik (2015) a "smart" asset management system should leverage information on predictive maintenance. This can be achieved by creating a data flow where first, an ETL process extracts and structures data from different sources. Next, a data warehouse is created to store clean data. Following this, a reporting layer is developed for basic analysis and data marts creation if needed. Finally, a complete analytical layer is built, where advanced analytics and predictive models are trained. The results are stored in the data warehouse as well, enabling the asset management team to make data-driven decisions based on predictive and descriptive information.

2.4. DATA WAREHOUSING

As Turban (2015) said, data warehousing is a process or activity that results in data products that provides decision support capability, allows access to business information and generation of insights. As data warehousing is a complete discipline and data warehouse is a repository, there could be distinct types of Data Warehouses, depending on the use and application: data marts, operational data stores and enterprise data warehouses.

2.4.1. Data Warehouse

The data warehouse "is a repository of current and historical data created to support decision making. The stored data is usually structured to be available for analytical processing activities" (Turban, 2015). Meaning that, the data being uploaded to the repository should be structured in a related form between the entities, so the generation of information and insights from the data is possible.

2.4.1.1. Characteristics of a Data Warehouse

Based on what Turban (2015) explains a data warehouse's characteristics should be:

Subject Oriented: The data is organized by a detailed subject, which enables users to determine the performance of the business and a more comprehensive view of the organization.
Integrated: The data warehouse must join and place data from different sources into a one consistent format that enables relationships between entities.
Time Variant: Time or date is the one important dimension that all data warehouses must support, this dimension allows the detections of trends, deviations and long-term relationships for forecasting and comparisons, guiding to an improved decision making.
Nonvolatile: When the data entered the Data Warehouse, users can not change the data and the obsolete data should be discarded.

2.4.1.2. Data Marts

"A Data Mart is a repository that focuses on a particular subject or area withing the organization" (Turban, 2015). Usually, this type of data warehouse is smaller but still needs to develop the entire process of data warehousing.

Types of Data Marts:

Dependent Data Marts: "Is a subset of data that is created from the Enterprise Data Warehouse" (Turban, 2015), Which means that an Enterprise data warehouse must be constructed first. This type of Data Mart supports the concept of a single enterprise-wide data model and takes the advantage from it providing quality data to the decision makers.

Independent Data Marts: Is a small warehouse designed for the necessities of a business unit or an area within and organization, the main difference with the dependent type is that, the source of data is not a EDW (Turban, 2015).

2.5. Data Warehousing Architecture

The data warehousing architecture helps to design the data repository based on the scalability, performance, volume of data and the availability of the data for the different areas of the organizations that the project needs to have, also it evaluates the hardware needs for the development. These factors are the first part of the technical approach that is needed when designing a database.

There are many basic information system architectures used for data warehousing, usually called N-tier architectures. These types of multi-tiered architectures are used to serve the needs of a large-scale and performance demanding data warehouses (Turban, 2015).

In a 3-tier architecture, the first tier is where the operational system contains the data and the software for the data acquisition, the data warehouse is a complete tier and finally the third-tier has the BI engine where the users have access to the data (Turban, 2015).

The 2-tier architecture, Joins the database server and the BI engine in just one tier. Which could be a potential problem for the performance when working with large data warehouses, but this architecture is more economical for development (Turban, 2015).

In the case explained by Campos et al (2017) a 3-layer architecture was developed for an asset management area. The first layer involves the ETL process, which extracts data from different sources, utilizing big data technologies to handle the volume, velocity, and variety of structured and unstructured data. In the second layer, the loaded data is analyzed by data scientists, turning this tier into an analytical layer through data mining and big data analytics technologies. The final layer is the visualization layer, where the asset management team members can understand the information and generate insights from the visualizations.

2.5.1. Alternative Data Warehousing Architectures

In a macro perspective of data warehousing architecture, the design is usually categorized into an enterprise-wide data warehouse (EDW) or a data mart (DM). There are some alternative architecture designs that do not strictly adhere to either EDW or DM principles (Turban, 2015):

Independent Data Marts: Data marts are designed to operate independently, because they respond to the individual needs of each area or business units. As the data definitions could be different from each one, it would be difficult to analyze the data across the organization. This architecture is the simplest and most economical one to develop and apply.
Data Mart Bus Architecture: This architecture joins the different Data marts that exist in every business unit via middleware. The idea would be consistency in the data, so the users can make complex queries, but the performance could be compromised.
Hub-and-spoke architecture: In this case the architecture refers to the creation of an Enterprise-wide data warehouse (EDW) and then dependent data marts to deliver specific information to every business unit. This architecture is focused on building a scalable and maintainable infrastructure, also easy to customize for the users. The hub-and-spoke architecture may lead to redundancy and latency.
Centralized data warehouse: This architecture is like Hub-and-spoke but there is no dependent Data Marts, meaning that there is just one Enterprise-wide data warehouse (EDW) where users from all the business units can access the data. This approach reduces the operative work of the ETL and could help to have a holistic view of the organization when it is well developed (Turban, 2015).

2.6. Extraction Transformation and Load (ETL)

The ETL process is the technical aspect that describes the activities to be undertaken in data warehousing. This process is responsible for transporting data from various sources to a target database, organizing it in a specific structure to establish relationships between entities within the target database.
The extraction activity involves accessing data from relevant sources, which can take various forms such as OLTP databases, files, spreadsheets, etc. The transformation activity is the process of structuring and formatting this data, making it suitable for placement in the database. Loading occurs when this organized data is inserted into the data warehouse (Turban, 2015).

Another aspect of the ETL process is its scheduling, as noted by De Assis Velila (2023) The ETL typically transfers data from the source to the data warehouse periodically. This cycle can occur daily, monthly, yearly, or even multiple times per day, depending on the velocity of data creation and analysis. During the ETL process, the OLAP system should be offline. Typically, the time window for executing the ETL is selected to minimize disruptions, ensuring that users do not access or query the database during this period.

2.7. Data Warehouse Development

Establishing a data warehouse within a company or department is a significant activity that can transform the entire decision support system. It empowers users with information, enabling them to make more informed decisions. The implementation of this information management technology brings about numerous benefits, including digital transformation, ultimately leading to the organization becoming data driven. But it also brings some challenges to the development of the Data Base, therefore the approach and the representation of the data are important matters that need to be analyzed during the design process.

2.7.1. Data Warehouses Development Approaches

Inmon model, edw approach: This approach is based on a top-down development, using a stablished Enterprise-Wide data warehouse, which provides a consistent understanding of the whole organization and keeps the idea of one only truth (Turban, 2015).

Kimball model, data mart approach: This type of strategy is usually called “Plan big, build small”, this approach consists of creating scaled-down data warehouses known as data marts which are subject-oriented or department-oriented repositories of data (Turban, 2015).

Conceptual modeling, As explained by Mathew et al (2009) this modeling approach has been used in various cases for implementing a data repository in the asset management domain. This type of design is applied when the database needs to store crucial data for each subarea within asset management. It requires a comprehensive understanding of the different norms, measures, and variables involved. Ultimately, this approach can be effective but needs the development of a complete data warehouse for the asset management area, potentially accompanied by data marts to facilitate analytics for each subarea. This kind of modeling demands significant effort from the development team and extensive collaboration with the subareas, making it both time-consuming and costly. However, it offers a high level of detail in terms of stored information.

2.7.2. Representation of Data in Data Warehouse

There could be different architectures of the data warehouse, but the design of the data representation in the data warehouse has always been based on the dimensional modeling concept (Turban, 2015).

Dimensional modeling: This is a relational system designed to facilitate high-volume query access. The data warehouse should be structured in a manner that optimizes data storage and accommodates multidimensional queries. Typically, the star and snowflake schemas are employed as the methods to implement dimensional modeling within the repository (Turban, 2015).

As Stated by Garani (2023), a dimensional database design offers advantages such as versatility, ease of use, and improved efficiency in querying the database, particularly when creating visual reports with BI tools.

2.7.2.1. Star Schema

This schema is the most used and the simplest form of dimensional modeling. It consists of a central table called fact table, which is connected to dimension tables linked by foreign keys (Turban, 2015). The fact table contains all the performance measures, addressing what the data warehouse supports for decision analysis about the transactions. The fact table is surrounded by the dimension tables, dimension tables contain the attributes describing records within the fact table. The hierarchies in the dimension tables show how the measures can be analyzed, aggregated, or summarized. The dimension tables must have a one-to-many relationship with the foreign key rows in the fact table (Turban, 2015).

2.8. From Entity Relationship Models to Dimensional Models

As the study focusses on extracting and loading the data from a source that is a database with structured data generated in a OLTP system, it is needed to use a methodology to extract the relevant metrics that were discovered. The models that support an OLTP system works in a different way, because of its representation, normalization of data or other reasons, than the one that we aim to develop. Based on this methodology the study will be able to translate the data found in the source into the data that is needed for the data base designed.

The data models employed for operational systems (OLTP) are designed for transaction processing environments. Typically, these models minimize redundancy to optimize the updating. However, the challenge with relational models in OLTP systems is their complexity, making it difficult for decision-makers to comprehend how to relate to the entities. Querying information from these databases becomes challenging, often requiring reliance on technical specialists to write queries for decision-makers (Moody, 2000). To develop a dimensional model from an entity-relationship model, a specific method will be outlined below in the next sub chapter.

2.8.1. Classify Entities

The first step to create a dimensional model is to classify the entities that we have in the model that we are going to use as a source, the classification of entities is (Moody, 2000):

Transaction entities: this is the data that contains the details of the events that happen in the business, and this data should be measurements or quantities. These kinds of entities are the most important ones in a data warehouse and are the base for the fact tables in a star schema.

Component entities: these entities are the ones that are related to a transaction entity via one-to-many relationship and describe the characteristics or details of each business transaction. Normally the component entities help to answer “Who”, “Where”, “When”, “Where”. A key component entity is the time, which answers the question when, this entity allows the time historical analysis of the transactions. These entities are the basis for the construction of the dimension tables.

Classification: entities in the OLTP model are related to a component entity by one-to-many relationships, so they are dependent on a component entity. Classification Entities represent hierarchies which could be joined with component entities and then collapsed to form dimension tables in a star schema.

2.8.2. Identify Hierarchies

The understanding of the hierarchies in dimensional model is important, because most of the dimension tables in a star schema have embedded hierarchies, a hierarchy in an entity relational model is a sequence of entities joined together via one-to-many relationships in the same direction (Moody, 2000).

The hierarchies could be, “maximal” if it cannot be extended up or down by including another entity, In the entity relationship model an entity is “minimal” or “leaf” if it is at the bottom of a *maximal hierarchy* and “maximal” or “root” if it is at the top of a *maximal hierarchy* (Moody, 2000).

2.8.3. Produce Dimensional Models

The basis for creating dimensional model from an entity model are tow operators (Moody, 2000):

Collapse hierarchy: Normalization is a standard practice in an entity-relationship model due to the construction of hierarchies in different entities connected through one-to-many relationships. The concept behind the "Collapse Hierarchy" operator is to denormalize these entities by consolidating them into a single entity within the dimensional model, typically transforming it into a dimension table.

Aggregation: This operator is employed when there is a necessity to form a new entity in the dimensional model derived from a transactional entity but with aggregated data. Meaning that in the new dimensional model, a fact table could be created, but the granularity of the information could differ. Hence, an aggregation to the measures is applied to transform them into the new table.

2.9. DIMENSIONAL DESIGN PROCESS

To design a dimensional model for a database, it is important to make four key decisions. “select the business process, declare the grain, identify the dimensions and identify the fact” (Kimball & Ross, 2013).

2.9.1. Business Processes

The business processes encompass the activities carried out by the organization. These transactions generate the measures or metrics stored in the repository for analysis. Typically, a specific business process is translated into a fact table. The chosen process would help to define a design target and it will let the grain, fact table and dimension tables to be declared in an easier way (Kimball & Ross, 2013).

2.9.2. Grain

Declaring the granularity is a crucial step in the dimensional model design. It defines what each row in the fact table represents, specifying the depth level of the data stored in the dimensional model. The granularity should be consistent across both fact and dimension tables. Given its importance, it's recommended to declare the granularity before selecting dimensions or facts in the design process. (Kimball & Ross, 2013)

2.9.3. Dimension Table Techniques

Dimension table structure: Each dimension table should have a unique primary key column, ensuring the uniqueness of each row without repetition. This primary key is then integrated as a foreign key in the related fact table, establishing a one-to-many relationship, as mentioned earlier. The context of the row on the transaction within the fact table describes its attributes. Dimension tables are denormalized, and their attributes are subject to constraints and grouping specifications that allows users to aggregate data for analysis (Kimball & Ross, 2013).

Dimension surrogate keys: In the context of relational models, dimension tables feature a unique primary key column. This primary key cannot be the operational system key, which is usually known as *business key*. To maintain control over these primary keys, which may come from various sources and may be poorly managed, anonymous integer primary keys needs to be generated for each dimension. The dimension ‘date’ could be exempt of this rule because this dimension is highly predictable, and it is easier to find a meaningful primary key (Kimball & Ross, 2013).

Slowly changing dimensions: As it is common to have attributes in the dimensions tables that could change, they would need diferent change tracking types. Type 0 retain original, type 1 overwrites, type 2 add new row and type 3 add new attribute (Kimball & Ross, 2013).

2.9.4. Fact Table Techniques

Structure of the fact table: Within the data warehouse, the fact table serves as the repository for all numeric measures resulting from an operational event. This transaction needs to be aligned with the declared grain during the model design. The fact table consistently includes foreign keys, using the surrogate keys, for each associated dimension. Given that the fact table contains all the measures and metrics of the transactions, these tables become the focal point for computations and aggregations when querying the database (Kimball & Ross, 2013).

3. METHODOLOGY

3.1. FINDING THE BUSINESS QUESTIONS

To initiate the process of designing the independent DataMart, it is essential to delve into the asset management area's strategy. This will enable us to formulate the business questions that will serve as a guide for the extraction of the important data from the OLTP system.

This segment of the research is based on the "Goal Question Metric" methodology (Basili et al., 1994), which allows us to establish a measurement system for a specific set of issues. It is divided into three distinct levels: conceptual, quantitative, and operational levels which correspond to goals, questions, and metrics, respectively.

Adopting this methodology is particularly relevant as the Independent DataMart is intended for use and design within the asset management area. This approach facilitates the exploration of information at different depths to understand the area's objectives. The business questions that need to be addressed to achieve the objectives, and the data that must be stored to answer these questions.

The knowledge of the strategy, the required information, and the business questions pertinent to the Asset Management team, are held by the people working within the department. To extract this valuable information a meeting was developed with the asset management team, which is composed of a manager, coordinator, three specialists and two technicians, these team members have experience in maintenance, asset management and civil engineering.

The meeting was structured into three parts, each corresponding to one of the three levels to systematically generate and categorize the ideas created by the team. To facilitate the idea generation process, we employed various design thinking techniques. These methods not only aided in idea generation but also kept the participants engaged in the activities.

3.1.1. Goal

The first part of the meeting aimed at defining the goals pursued by the Asset Management area. We employed the design thinking methodology known as 'Silent Brainstorming or brainwriting'. As explained by Litcanu et al (2015) In this approach, participants silently generated ideas and subsequently shared them with the team, explaining the logic behind their choices. To identify the most significant ideas, team members then voted for the most important ones. In table 1 the results from this initial activity are shown.

Table 1 - Discovered Goals

Goals	votes
Assets criticality matrix	7
Ensure data quality	6
Concept standardizations between business units	4
Problem-Cause-Action methodology for assets	3
Asset optimization, increase performance with cost reduction	3
definition of long-term investment needs	2
Connection between Maximo-SIG-APMS	2

Goals	votes
Data extraction optimization of the friction process	1
Asset intervention planning	1
Creation of an alert system	1
Assist the executive committee (COMEXE) in decision-making	1
Prevention of future asset interventions with a ML model	
Creation of an Asset inventory	
Planning & estimation of asset operational time	
Recognition by the business units of the impacts of GA on their operations	

Based on the votes and the impact that the goal could have in the asset management area 6 specific goals were chosen to be analyzed in the quantitative level of the methodology, table 2 show the selected goals.

Table 2 - Selected goals

Goals
Assets criticality matrix
Ensure data quality
Concepts standardization between business units
Asset optimization, increase performance with cost reduction
Asset intervention planning
Planning & estimation of asset operational time

3.1.2. Question

The second phase of the meeting was dedicated to advancing to the next level of information, the quantitative aspect, which refers to questions. Once we had identified the most critical goals for the Asset Management area, the team had to generate questions related to those specific goals. To facilitate this, we employed the 'Brainwriting' technique, but with a variation.

In this exercise, each team member was tasked with contributing two question ideas for every goal. Then, these questions were shared with the entire team. Following this, each team member read the questions generated and was asked to develop additional ones. This process was repeated three rounds to ensure a substantial pool of questions, resulting in 32 ideas.

Finally, the team collectively assessed and voted for the most significant questions aligned with each goal. In table 3 the reader can find the most important questions by goal:

Table 3 - Discovered Questions

Goal	Question
Asset intervention planning	The asset has a maintenance plan?
	The Asset has a replacement plan?
	When do we need to make a big rehabilitation or replace the asset?
Asset optimization, increase performance with cost reduction	How good is the performance of the assets?
	What are the costs related to the maintenance of the assets
	The asset has a life cycle analysis?
Concepts standardization between business units	How to convey this need for standardization to the UN?
	How to define the standardization of the nomenclature
	What are the advantages of this standardization?
	What concepts do we want to standardize? How can we identify the most important ones?
Ensure data quality	Which are the essential data that we need as reliable ones?
	The KPI´s must be defined by whom?
	Which are the main KPIs (infrastructures, locations, assets) that we need to monitor?
	How can we analyze the Ots information, by site, Asset, Labor
Assets criticality matrix	How to assess the condition or state of an asset?
	Who needs to be consulted for the evaluation of criticality?
	What is the admissible risk for the assets?
Planning & estimation of asset operational time	What is the operational time of each asset?
	What is the downtime for each asset
	What are the time constraints for each airport/asset type

3.1.3. Metric

To complete the methodological approach, the last dynamic was used to find the metrics that would help to answer the defined and most important questions. As there were many business questions that would need a metric, the team was divided into small groups to brainstorm and identify the measurements for the assigned goals. As the team members have expertise in asset management and maintenance, the results proved meaningful. However, there was one challenge. Some of the goals represent processes that are still under development and data on these matters is not yet available. In table 4 the metrics by question are shown.

Table 4 - Discovered Metrics

Goal	Question	Metric
Asset intervention planning	The asset has a maintenance plan?	Count of future Work orders for an asset.
	The Asset has a replacement plan?	It would be based on the recommended lifetime.
	When do we need to make a big rehabilitation or replace the asset?	It is a decision based on the KPIs in the life cycle report like costs and time.

Goal	Question	Metric
Asset optimization, increase performance with cost reduction	How good is the performance of the assets?	The Question are answered with the measures and metrics that we have in the life cycle analysis report: Interventions, KPIs, Labor & Costs.
	What are the costs related to the maintenance of the assets	
	The asset has a life cycle analysis?	
Concepts standardization between business units	How to convey this need for standardization to the UN?	# assets enter right and wrong, # of assets with standardized nomenclature, qualify or get feedback of the nomenclature in a specific scale to measure.
	How to define the standardization of the nomenclature	# of people that are using the nomenclature, # of types of nomenclature, how fast is it to find and asset based on that nomenclature.
	What are the advantages of this standardization?	faster registration of the WO, a smaller number of classifications, different # of maintenance plans for same type of assets in the airports, variation of the # of registered WO for same asset type.
	What concepts do we want to standardize? How can we identify the most important ones?	Regarding the asset data, how to create descriptions, locations and codes to identify the assets.
Ensure data quality	Which are the essential data that we need as reliable ones?	Asset: Site, ID, RQSA, Installation date, expiration date, rotable, CC, labor, Work Orders.
	The KPI´s must be defined by whom?	the Area of asset management manages the information that is required for the operation of maintenance, the engineers within the team with the operational team must define the Kpis.
	Which are the main KPIs (infrastructures, locations, assets) that we need to monitor?	Performance of the assets during their life cycle and correlated with the users of the airports.
	How can we analyze the Ots information, by site, Asset, Labor	Developing the historical records of this data as dimensions for a relational model.
Assets criticality matrix	How to assess the condition or state of an asset?	The condition is measured on a specific scale, that is created based on the know-how of the asset functionality and maintenance information.
	Who needs to be consulted for the evaluation of criticality?	Segurança/RH/SST Operação/Manutenção Ambiente/Manutenção for each of the consulted areas there would be a specific metric, but it's also based on the know-how of the people that works withing each team.
	What is the admissible risk for the assets?	Depending on each area that analyses the risk/criticality there would be a definition, it could change on every type of asset.
Planning & estimation of asset operational time	What is the operational time of each asset?	Most of the assets are measured as they are operating 24 hours a day, but depending on the airport this could change.
	What is the downtime for each asset	It's a KPI or metric that Maximo has the information based on the work orders maintenance time information.
	What are the time constraints for each airport/asset type	The quantity of time depends on the operation of the airport and type of asset.

While the brainstorming activity generated valuable insights, a significant challenge emerged due to data limitations. Consequently, this study will concentrate on two specific goals: asset intervention planning and asset optimization. These two objectives have a fundamental significance, given their substantial impact on the operational efficiency and the availability of pertinent data within the Enterprise Asset Management (EAM) system, therefore, the chosen business questions that will guide the development of the database and will be answered with the resulting model are shown in table 5.

Table 5 - Final Business Questions

Goal	Question	Metric
Asset intervention planning	**When to choose between replacement and rehabilitation of the asset?**	It is a decision based on the KPIs in the life cycle report like costs and time.
Asset optimization, increase performance with cost reduction	**How good is the performance of the assets?**	The Question are answered with the measures and metrics that we have in the life cycle analysis report: Interventions, KPIs, Labor & Costs.
	What are the costs related to the maintenance of the assets	
	The asset has a life cycle analysis?	

A key source of metrics for these two targeted goals is the Life Cycle Analysis report. Therefore, comprehending the document is essential to extract accurate data from the EAM's database. The upcoming chapter will present the report to provide a better understanding of its content.

3.1.4. Life Cycle Analysis Report

Our focus will be on generating insights from the 'Life Cycle Analysis' report and its Key Performance Indicators (KPIs). However, it is important to note that this report is static, requiring users to manually download a PDF report for an individual asset or asset groups. The staff has not been taking advantage of the information in the report because it is time-consuming and is difficult to comprehend the information effectively. Moreover, since the EAM system is not designed to be a reporting system, the development of its reports is limited, despite the valuable information they include. The report has 4 sections that explains the condition and performance of the Assets. *Characterization*, in this section, essential asset information is provided to the user. This includes details such as brand, model, and serial number as shown in figure 3.


MAPA 022.01 CICLO DE VIDA DO ATIVO

Ativo: RCC.CAPN.R1 - Regulador de Corrente Constante CAP Nascente - R1
Classificação: Reguladores de Corrente Constante
Localização: 065.01.00.01.AT65.1.6 - Reguladores de corrente constante
Setor Responsável: AFRMELE
Centro Custo: 535162

1. Caraterização

Marca: ADB	Modelo: CRE	Nº Série: 00022012/3
Nº Imobilizado:	Data Aquisição / Instalação: n.a.	Idade: n.d.
Preço Aquisição: n.a.	Data Estimada Fim de Vida: n.d.	Período Estimado de Vida Útil: n.a.
Contrato de Manutenção: ☐	Prestador Atual:	

Figure 3 - Header and characterization of the life cycle analysis report

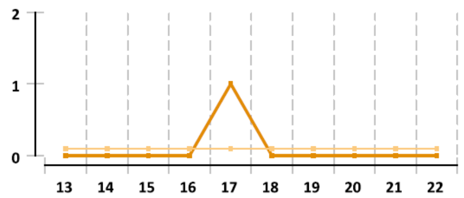
Technical Analysis, this section offers analysts the chance to extract insights from the data's more technical aspects. It is divided into two distinct parts “technical analysis” in figure 4 and technical indicators in figure 5.

2. Análise Técnica

2.1 Intervenções Técnicas

Ano	Incidências (Qt.)	Ordens Trabalho (Qt.)						
		MC	MP	MC/(MC+MP)	ST Alter.	ST Benef.	ST Outras	Total
2013	0	0	6	0,00%	0	0	0	6
2014	0	0	11	0,00%	0	0	1	12
2015	0	0	10	0,00%	0	0	1	11
2016	0	0	11	0,00%	0	0	1	12
2017	1	0	9	0,00%	0	0	1	10
2018	0	1	12	7,69%	0	0	0	13
2019	0	0	11	0,00%	0	0	0	11
2020	0	2	9	18,18%	0	2	0	13
2021	0	1	11	8,33%	0	2	1	15
2022	0	1	10	9,09%	0	0	0	11
Média	**0,1**	**0,5**	**10,0**	**4,76%**	**0,0**	**0,4**	**0,5**	**11,4**

2.1.1 Incidências



2.1.2 Ordens de Trabalho MC

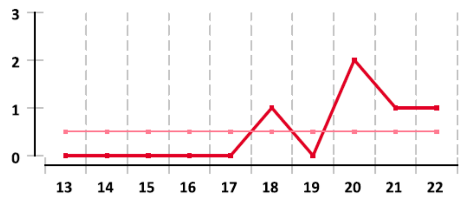
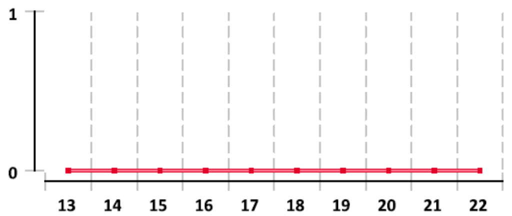


Figure 4 - Technical analysis of the life cycle analysis report

2.2 Indicadores Técnicos

Ano	Avarias (Qt.)	Disp. Téc. (%)	MTBF (h:min)	MRT (h:min)	MMT (h:min)	MTTR (h:min)	Taxa Avarias (%)	Fiabilidade (%)
2013	0	100,0000		00:00	119:48	00:00	0,0000	100,0000
2014	0	100,0000		00:00	28:58	00:00	0,0000	100,0000
2015	0	100,0000		00:00	87:32	00:00	0,0000	100,0000
2016	0	100,0000		00:00	118:26	00:00	0,0000	100,0000
2017	0	100,0000		00:00	128:34	00:00	0,0000	100,0000
2018	0	100,0000		03:00	49:09	00:00	0,0000	100,0000
2019	0	100,0000		00:00	50:18	00:00	0,0000	100,0000
2020	0	100,0000		04:00	38:14	00:00	0,0000	100,0000
2021	0	100,0000		00:30	65:05	00:00	0,0000	100,0000
2022	0	100,0000		00:50	70:29	00:00	0,0000	100,0000
Global	**0**	**100,0000**		**02:28**	**71:50**	**00:00**	**0,0000**	**100,0000**

2.2.1 Avarias



2.2.2 Disponibilidade Técnica

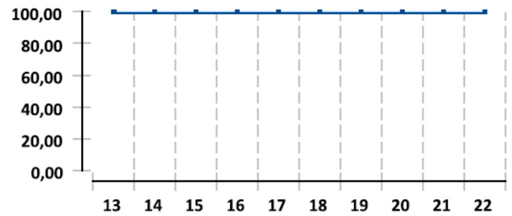


Figure 5 - Technical indicators of the life cycle analysis report

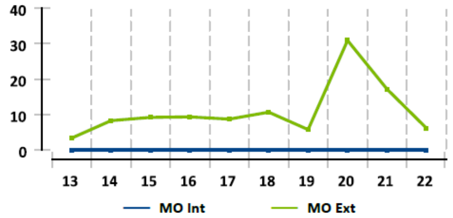
Organizational Analysis, in this section of the report, the user can find information related to labor in man-hours, which represents the amount of time technicians spent on asset maintenance as shown in figure 6.

3. Análise Organizacional

3.1 Mão-de-Obra (h.H)

Ano	Interna				Externa				Total
	MC	MP	Total	%	MC	MP	Total	%	
2013	0,00	3,34	3,34	100,00	0,00	0,00	0,00	0,00	3,34
2014	0,00	6,44	8,27	100,00	0,00	0,00	0,00	0,00	8,27
2015	0,00	5,19	9,19	100,00	0,00	0,00	0,00	0,00	9,19
2016	0,00	6,32	9,32	100,00	0,00	0,00	0,00	0,00	9,32
2017	0,00	4,20	8,70	100,00	0,00	0,00	0,00	0,00	8,70
2018	3,00	7,66	10,66	100,00	0,00	0,00	0,00	0,00	10,66
2019	0,00	5,74	5,74	100,00	0,00	0,00	0,00	0,00	5,74
2020	6,50	4,42	30,92	100,00	0,00	0,00	0,00	0,00	30,92
2021	1,00	5,68	17,18	100,00	0,00	0,00	0,00	0,00	17,18
2022	1,67	4,50	6,17	100,00	0,00	0,00	0,00	0,00	6,17
Média	**1,22**	**5,35**	**10,95**	**100,00**	**0,00**	**0,00**	**0,00**	**0,00**	**10,95**

3.1.1 Mão-de-Obra Total (h.H)



3.1.2 Mão-de-Obra (Último Ano)

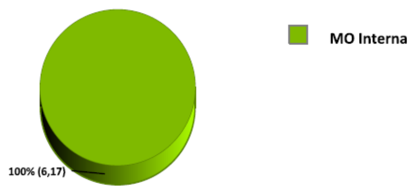


Figure 6 - Organizational analysis of the life cycle analysis report

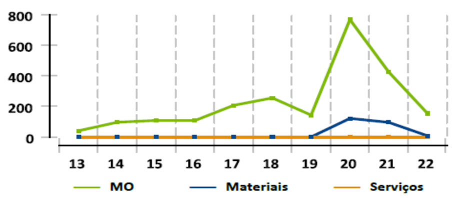
Economic Analysis, this section of the report explains the costs of maintenance activities for the assets. In this case, the values are presented in Euros as shown in figure 7.

4. Análise Económica

4.1 Custos (€)

Ano	Mão-de-Obra	Material	Serviços		Total
	Interna		Contratuais(1)	Pontuais	
2013	40,24	0,00	0,00	0,00	40,24
2014	95,97	0,00	0,00	0,00	95,97
2015	107,67	0,00	0,00	0,00	107,67
2016	106,92	0,00	0,00	0,00	106,92
2017	206,52	0,00	0,00	0,00	206,52
2018	254,77	0,00	0,00	0,00	254,77
2019	142,59	0,00	0,00	0,00	142,59
2020	767,06	121,06	0,00	0,00	888,12
2021	424,74	95,18	0,00	0,00	519,92
2022	153,92	7,57	0,00	0,00	161,49
Total	**2.300,40**	**223,81**	**0,00**	**0,00**	**2.524,21**

4.1.1 Custos Parcelares



4.1.2 Custo Total

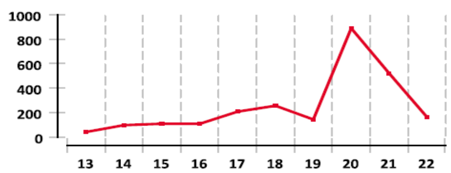


Figure 7 - Economical analysis of the life cycle analysis report

The 'Life Cycle Analysis' report is a valuable tool for analyzing an asset's yearly performance in terms of maintenance activities. However, there is no clarity in the key performance indicators (KPIs) and measures of each section. The visuals presented are not self-explanatory, but the tables included in the report aid in the comprehension.

3.2. DATA DESCRIPTION

3.2.1. Source

The data to be retrieved will be originated from Maximo EAM. Access to the database, where transactions and relevant data are recorded, is essential for utilizing it as a source in the ETL process.

The database utilized in this study is named "MX76REP" residing within the SQL server "AHD-VSQLSTD-SGM\SGM." Access has been granted at a read-only level for the development process. This database operates as a reporting repository, meaning that, despite daily data collection from the EAM, information is loaded only once per day. This contrasts with the production database, where transactions are recorded in real time. This latency in the updating of the data is a manageable matter because the DataMart will update the ETL process one time per day.

Securing access to this server and database ensures retrieval of accurate and up-to-date data, mitigating the risk of any potential impact or issues to the production database during the ETL process.

Within the database, there are 1297 tables and 266 views that contain structured data, created according to a relationship model that differs significantly from the one intended for the data mart. It is essential to note that not all data stored in the EAM's database will be collected for the repository, the metrics crucial for addressing the predefined business questions will be extracted. To design the data mart systematically, the methodology "FROM ENTITY RELATIONSHIP MODELS TO DIMENSIONAL MODELS" will be followed.

3.2.2. Data Mart Hosting

The database is set to reside on a server named “ANS-GESTATIVOS.ANA.PT”, which was provided by the company. This on-premises SQL server will be under the administration of the company's IT team, responsible for managing scalability and maintenance. Additionally, the asset management area has control over the server, allowing them to manage access permissions for the databases within.

Acquiring a server operating 24/7 was an important move for the asset management area, given the numerous data projects within the domain. This server will serve as a central hub for storing all data products generated by the area. The decision to opt for an on-premises server was the result of an analysis by the IT department. They evaluated the volume of data to be stored and processed daily in the database, concluding that an on-premises server was the optimal and fastest solution. In case the capacity threshold is approached, the area will be notified, and IT will scale the server accordingly.

For the development of the DataMart, two distinct databases will be established within the server. The initial database, named "DM_staging_Area" will function as the staging area. This is where extracted data from the source undergoes the necessary transformations to align with the relational model design before being loaded into the DataMart Database. The second database is named "Data_Mart_GA". In both databases, the loaded data will be structured data.

3.3. DATAMART DESIGN

The ultimate objective of this study is to create and implement a repository of structured data. Once the source data has been examined, the ETL process could be able to link and retrieve it. The information to be extracted is defined by the business questions found. Also, with the designated server for the new database identified, it is possible to start the database's design, which will be divided into development approach, database architecture, model representation and the ETL process.

3.3.1. DataMart Development Approach

To enhance and transform the decision support system within the asset management area, an initial step in the digital transformation journey is converting collected data into information. The key to add value to the data lies in contextualizing it. This means that once the data sourced from its origin is organized, structured, and categorized, the team can use this information for in-depth analysis and informed decision-making.

Consequently, there is a need for the asset management area to create a repository of meticulously organized and structured data. The choice of the database development approach is based on the scale of the data to be retrieved. In this study the goal is to store a segment of the extensive data generated across the entire organization, the Data Mart approach or also called "Kimball model approach" is the choice. This approach designs data storage to suit the specific needs and subjects identified in the defined metrics, essentially creating a scaled-down data warehouse.

3.3.1.1. Kimball Model

Now that the approach is defined, it's crucial to outline the dimensional design process for this model across its four stages.

Business process, the selected business process for this model revolves around the asset management sub-area that focuses on overseeing the technical assets within airports and their life cycle. In this context, all data generated pertaining to asset usage and interventions is linked to the operational activities of the maintenance area. The entire data is sourced from the Enterprise Asset Management (EAM) system, serving as the foundation for the dimensional model. Additionally, the research conducted with the team to find the business questions helped determine the specific processes and data elements that will be incorporated.

Declare the grain, to declare the grain, the study refers to the transactions recorded in the system, which represent how data is stored. In this case, the transactions consist of work orders that describe the actions performed by the maintenance team on assets. These activities generate various measures, such as the date at a level of day, responsible sector, asset involved, and location of the transaction. Understanding these transactional details defines the granularity of the model, the reader will be able to see that every dimension will follow a hierarchy that allows to aggregate the data for analysis purposes.

Identify the dimensions, the dimensions of the model play a crucial role in describing transactional data. They serve as the foundation for aggregating and explaining the results based on the collected data. Upon declaring the grain, it became evident that defining 'what,' 'when,' 'where,' and 'who' intervened in the transaction was essential. Therefore, the following dimensions were identified to capture these aspects of a work order.

- Dimension Date
- Dimension Asset
- Dimension Sector
- Dimension Location

Identify the fact, as fact tables are the type of tables that allow recording the measures involved in transactions. All values related to the work done by maintenance teams on assets are recorded in transactions called 'work orders'. Additionally, another transaction recorded in this model is the 'service request', which contains information about an initial approach to maintenance work that should be done on an asset. Therefore, the identified fact tables are:

- Fact Work Order
- Fact Service Request

3.3.2. DataMart Architecture

The business questions addressed by the metrics retrieved from the EAM are highly specific and focused on the operational aspects of the maintenance area. The information stored in the DataMart will be accessible to both the asset management team, for analytical purposes, and the maintenance team as well, primarily for informative and actionable insights. The maintenance team will be able to use dashboards and reports to access the information and enhance operational efficiency.

This focus based on the needs of the business unit turns the repository into an independent design, the “Independent Data Mart Architecture” brings several advantages, including cost-effectiveness and independent operational capabilities. However, it has the potential to evolve into a "Data Mart Bus Architecture" if additional databases within the department are developed that could use the information stored in this repository.

3.3.3. Model Representation

The dimensional modeling facilitates easy multidimensional query access for a database, meaning that a dimensional model allows the stored data to be analyzed across the dimensions included in the model. The chosen schema for the independent   is a star schema, characterized by central fact tables containing measures and transactions that are connected to dimension tables that describe the data stored in the fact tables.

3.3.3.1. Fact tables of the independent DataMart

To address the business question effectively the essential transactions that should be stored in the independent DataMart are work orders and service requests.

Fact Work Order, A work order is a digital document created within the EAM system to describe an intervention performed by the maintenance team on an asset. Therefore, this table stores data that outlines the interventions conducted on the assets. It includes information like labor details, costs incurred, intervention duration, measures and KPIs, work order type, work order status and the foreign keys that facilitate the relationship between the model tables. In table 6, all the columns included in the fact work order table can be found.

Table 6 - Fact Work Order

Column Name	**Description**
fk_asset	Asset foreign key
fk_location	Location foreign key
fk_sector	Sector foreign key
fk_date	Date foreign key
bk_workorder	Work Order id number
wo_parent	Work Order parent id number
wo_status	Work Order status
wo_worderid	EAM work order id
wo_histflag	EAM historical work order
wo_statusdate	Last status change date of the work order
wo_oficialdate	Work Order creation date
wo_reportdate	Work Order report date
wo_targstartdate	Planned start date
wo_targcompdate	Planned finish date
wo_actstart	Activity start date
wo_actfinish	Activity finish date
wo_worktype	type of intervention (corrective, preventive, etc)
wo_mptype	Maintenance plan type
wo_description	Work Order description
wo_assetnum	EAM asset id
wo_location	EAM location id
wo_siteid	Airport id
wo_istask	Tasks of the work order
wo_class	Work Order classification
wo_unavailability	type on unavailability of the intervention
wo_downtime	Qty of downtime of the asset in hours
wo_actlabhrs	Qty of labor hours of the intervention
wo_actintlabhrs	Qty of internal labor hours of the intervention
wo_actoutlabhrs	Qty of external labor hours of the intervention
wo_actlabcost	Labor cost in Euros
wo_actintlabcost	Internal labor cost in Euros
wo_actoutlabcost	External labor cost in Euros
wo_actmatcost	Material cost in Euros
wo_actservcost	Service cost in Euros
wo_tta	Time to action in minutes
wo_rt	Repair time in minutes
wo_ttr	Time to repair in minutes
wo_mt	Maintenance time in minutes
wo_fr_symptom	Symptom of the failure
wo_fr_problem	Problem of the failure
wo_fr_cause	Cause of the failure

Column Name	Description
wo_fr_action	Action to solve the failure
wo_inc_id	Service Request id (if related)

Fact Service Request, represent initial communications to the maintenance team regarding asset issues, in this case a team within the maintenance area is contacted to review the problem and assesses whether the situation needs execution of work in a first approach on the asset or if the problem needs to transform the service request into a work order.

Usually, service requests are created when either an internal staff member or a user of the facilities encounters an issue that is not working properly. This issue is then reported to the maintenance department. Many service requests can be resolved through initial assessments or simple inspections. However, there are no specific metrics associated with resolving a service request; instead, they are characterized by descriptions such as request types, priority levels, or task statuses. As mentioned earlier, if the issue requires more extensive attention or has greater complexity, the service request is escalated and transformed into a work order.

Given the nature of the data collected from service requests, the chosen approach for this fact table is to create a factless table, as said by Kimball (2013) A factless fact table can capture events or transactions that lack numerical measures. It serves to record a group of dimensional entities converging at a specific moment. Additionally, factless fact tables can also be used to analyze scenarios where events did not occur.

The data stored in the service request fact table, as depicted in Table 7, is comparatively smaller in comparison to the work order fact table.

Table 7 - Fact Service Request

Column Name	Description
fk_asset	Asset foreign key
fk_location	Location foreign key
fk_sector	Sector foreign key
fk_date	Date foreign key
bk_service_request	Work Order id number
sr_assetnum	EAM asset id
sr_location	EAM location id
sr_request_date	Service Request creation date
sr_2line_date	Date of Service Request moving to diferent sector or activity
sr_register_date	Digital service Request document creation
sr_description	Service Request Description
sr_siteid	Airport id
sr_status	Service Request status
sr_status_date	Last status change date of the Service Request
sr_unnavailability	type on unavailability of the intervention
sr_priority	Classification of priority of the Service Request

Column Name	Description
sr_historical	Service Request is historical
sr_class	Type of service request
sr_assetoff	The asset stopped working during the intervention

3.3.3.2. Dimension tables of the independent DataMart

Dimension Date, this dimension, linked to the fact tables, serves to describe the dates *when* transactions occurred or started. Consequently, the analyses conducted by users utilizing the DataMart as a source will encompass the independent variable of date. The Dimension will be linked to the fact tables by a Surrogate key -Foreign key, the relationship is going to be a one-to-many type.

The dimension date is not directly linked to the EAM as a source for the database. Instead, it was designed based on analytical needs and is stored in a separate database within the same server as the DataMart. When additional data needs to be incorporated into this dimension, an ETL process utilizing a SSIS package is already created. Currently, the dates information stored in this dimension extends until the year 2030. The data's granularity within the dimension date is set daily. Although transactions are originally recorded in the OLTP system at date and time granularity, stakeholders within the asset management area concluded that daily-level analysis suffices for decision-making purposes.

The hierarchy of information within this dimension table is organized as year, quarter, month, week and day.

As the dimension date lacks direct connection to the OLTP system, there is no business key associated with it. Instead, a surrogate key is employed for both the staging area and DataMart databases to establish relationships with the fact tables. The columns included in the dimension date are shown in table 8.

Table 8 - Dimension Date

Column Name	Description
sk_date	Date surrogate key
full_date	Complete date
daymonth_number	Month number
dayweek_name	Day name of the week
dayweek_short_name	Short name of the day of the week
weekyear_number	Week number of the year
month_number	Month number
month_name	Month name
quarter_name	Quarte name of the year
year_number	Year number

Dimension Location, Within the location dimension lies information of *where* the transaction occurred. This encompasses a hierarchical and descriptive representation of the physical spaces in the organization where work orders or service requests arose. Like other dimensions employed in this relational model, a business key exists, which will be converted into a surrogate key for relational purposes with the fact tables. The asset management area employs a hierarchical organization system for the locations stored in the EAM's database. This hierarchy ranges from larger establishments such as "Airports" down to the most granular level, which is "Space". Therefore, the hierarchy used for the dimension location is airport, building/area, level/specific area, zone, sub zone and space.

In this scenario, denormalization of hierarchies from different tables within the OLTP system was unnecessary. Most location-related information, including its hierarchies, was consolidated within a single table named "Location". However, one specific piece of information required for analysis was the IATA name of airports, following international aviation conventions. Although the data in the "Location" table was not initially organized to align with the final design of the table, the transformation process facilitated the extraction of pertinent data to conform to the design requirements of the "dimension location" table in the DataMart which could be seen in table 9.

Table 9 - Dimension Location

Column Name	Description
sk_location	Location surrogate key
bk_location	Location business key
location_id	EAM location id
location_description	Location description
location_building_id	building Hierarchy key
location_building_desc	building Hierarchy description
location_level_id	Level Hierarchy key
location_level_desc	Level Hierarchy description
location_zone_id	Zone Hierarchy key
location_zone_desc	Zone Hierarchy description
location_subzone_id	Sub-zone Hierarchy key
location_subzone_desc	Sub-zone Hierarchy description
location_type	Type of location
location_status	Status of the location
location_statusdate	Last status change date of the location
location_site_id	Airport id related to the location
location_site_description	Name of the airport
location_site_iata	IATA code for the airport
location_hierarchy	level of hierarchy of the location
location_sig_max	location is mapped in SIG

The design of this table presents some specific considerations regarding hierarchical information. One notable aspect is that all levels of the hierarchy are included, not just the lowest level of granularity as in any usual design. This inclusion is necessary because the dimension must establish relationships with the fact tables, allowing transactions to be applied to locations at various levels within the hierarchy. To illustrate, consider a scenario where a work order is created for an asset such as a "car". Since this asset cannot be directly associated with a space-level location, transactions involving such assets are typically linked to higher-level locations, such as an airport. Therefore, it becomes imperative to record all locations within the table to relate the dimension table with the fact tables. As depicted in Figure 9, each part of the location hierarchy has an ID and a description, which is the business key divided into the hierarchies.

Another aspect of the design involves the creation of the "location_id" column. Due to the existence of internal norms to create the location´s business keys for airports, certain locations may share the same business key across different airports. Therefore, to ensure accurate tracking and maintain consistency within the relational model, the business key, along with the airport's unique identifier, is stored in the "bk_location" column. This approach enables effective management of relationships between tables in the model without compromising data integrity.

Dimension Sector, the company´s sectors describe specific areas within the organization, enabling users to analyze transactions based on *who* is responsible of the work orders or service requests. Given that these transactions are linked to the maintenance area's operations, sectors represent the most granular level of aggregation for the maintenance teams across the organization.

In this dimension, sectors are organized hierarchically across 8 levels, with the sector representing the most granular level and airports as the highest aggregation. However, due to variations in operational capacity across different airports, not all airports have 8 levels of hierarchy to differentiate sectors. While all sector information was contained in one table of the OLTP system database, the hierarchy construction was stored in another table within the EAM system. Following the technique of "from entity relational models to dimensional models," the denormalization process was executed to consolidate this information, in table 10 the reader can see the information that is included in the dimension sector table.

Table 10 - Dimension Sector

Column Name	**Description**
sk_sector	Sector Surrogate key
bk_sector	Sector business key
sector_description	Sector descirption
sector_level_1	Sector hierarchy level 1
sector_level_2	Sector hierarchy level 2
sector_level_3	Sector hierarchy level 3
sector_level_4	Sector hierarchy level 4
sector_level_5	Sector hierarchy level 5
sector_level_6	Sector hierarchy level 6
sector_level_7	Sector hierarchy level 7
sector_level_8	Sector hierarchy level 8
sector_siteid	Airport id related to the sector
sector_cost_center	Cost center of the sector
sector_child	sector has lower granularity
sector_active	Active status of the sector
sector_inactive	Inactive status of the sector

Dimension Asset, the asset serves as the focal point of analysis within the area, representing the object impacted by the work order or service request. It is also *what* is going to be inspected and repaired in the transaction. The organization has many assets that require proper care and strategy to ensure their proper functioning, that is one of the primary objectives of the asset management area. This dimension is structured according to the various aggregations established by the department for asset management. These predefined groupings of assets provided a framework for establishing the dimension's hierarchy.

The hierarchy from highest level of aggregation to the lowest level of granularity is type, classification, system, collection and asset.

In this dimension, all pertinent information regarding the asset is recorded, including the brand, model, installation date, price, and other relevant data. This table enables analysts to gain a comprehensive understanding of the asset. An important aspect of this dimension is the development of a business key during the design phase. This key is formed by concatenating the original business key of the asset with the ID of the airport where the asset is registered. This is necessary because the process of generating business keys for assets in the OLTP system involves sequential numbering, which can result in duplicate keys across different airports. To ensure uniqueness and avoid conflicts, the original business key of the asset, recorded in the column "asset_numid" in the dimensional model. The columns included in the dimension asset are shown in table 10.

Table 11 - Dimension Asset

Column Name	**Description**
sk_asset	Asset surrogate key
bk_asset	Asset business key
asset_numid	EAM asset id
asset_tag	Asset tag descirption
asset_description	Asset descirption
asset_model	Asset model
asset_brand	Asset brand
asset_serialn	Asset serial number
asset_cost	Asset cost
asset_rqsa	Asset type 'Reglamento Qualidade de Servicio Aeroportuario'
asset_installdate	Asset Installation date
asset_expectedlife	Asset expected life in years
asset_status	Asset status
asset_changedate	Last status change date of the asset
asset_parentid	Asset is child
asset_location	Airport related to the asset
asset_siteid	Airport id
asset_rotatable	Asset with variable location
asset_failureclass	Classification of failures of the asset
asset_collection	Asset collection
asset_system	Asset system

Column Name	Description
asset_classification	Asset classification
asset_type	Asset type
asset_rqsaid	Id of the RQSA classification

Finally, in figure 8, we can see the relationships between the tables of the star schema. The dimensions are connected to the fact tables through surrogate keys created for the DataMart. In the fact tables, these keys generate primary keys, ensuring the uniqueness of each transaction.

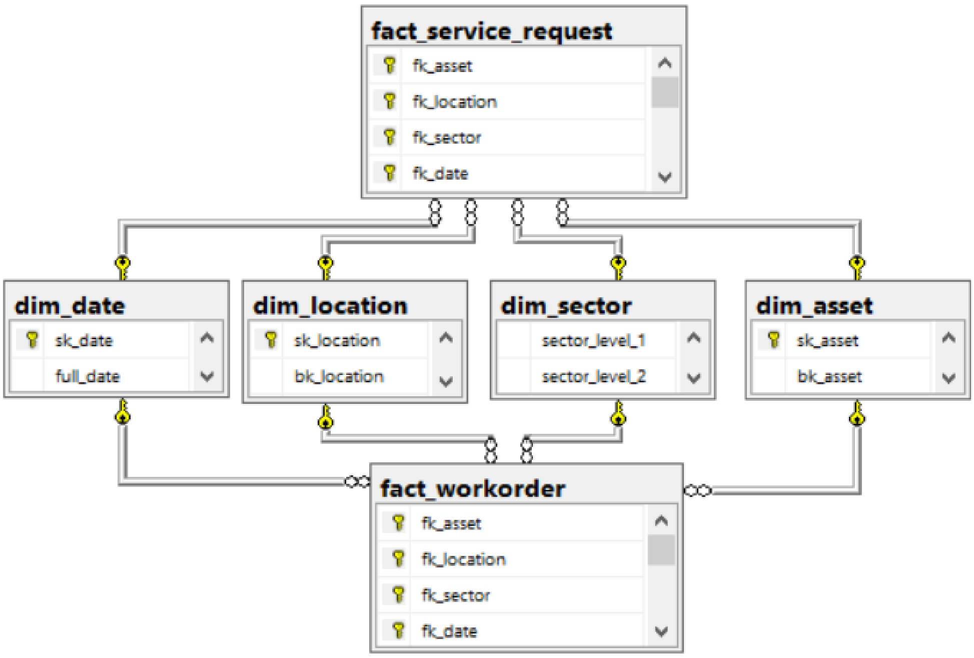


Figure 8 - Star Schema

3.3.4. ETL Process

The Extraction, Transformation, and Loading (ETL) process will oversee the movement of data from the source, which is the EAM's database, to the Asset Management's database located in the area's server. There will be two databases; the first one will be named "Staging Area". This database serves as the repository where the extracted data from the source is stored after applying the necessary transformations to comply with the design of the star schema. Also, the Staging Area does not record any data, instead, it functions as a temporary storage used as a source for the DataMart's database. This implies that every time that the Staging Area is updated, the old data is truncated, and the new information is stored again in the Staging Area´s database, after this process it can be transferred to the DataMart. An important aspect of the Staging Area is that, although a primary key is required in the fact tables, there is no relationship between the dimension tables and the fact tables. The key used in each dimension table is the business key, which is the former organizational identification code for the data within each dimension.

The DataMart database extracts and stores data from the Staging Area database. This data transferred between databases occurs seamlessly because the staging area completes the extraction and transformation of the data to align with the database structure. During this stage, new IDs are assigned to each record in both dimension and fact tables. In the case of dimensions, these IDs are known as surrogate keys and establish the relationships within the star schema. Unlike business keys, which are used for identification within the organization, surrogate keys provide a normalized approach to record identification. In the context of fact tables, a primary key is constructed by concatenating foreign keys, where the foreign keys correspond to the surrogate keys of the dimensions. Leveraging these inter-table relationships within the DataMart's star schema, the loading process takes place, ensuring accomplishment to relationship rules. Following the initial data load, only updated data is subsequently loaded into the database, incremental loading will be employed. Incremental Loading will be used when the dimensions need to be updated using slowly changing dimensions, which allows to add new data to the dimension tables and change specifical columns selected by the developer in the data that is already stored in the table, this happens because the process of the slowly changing dimension uses the key of the row in the table to look for updates in the selected columns. There are different types of slowly changing dimensions uses, for example when the record needs to be overwritten or because of the change in the data the table would create a new row to save the historical data and the new update.

Incremental loading will be employed to update the model´s dimensions through slowly changing dimensions (SCD), allowing for the addition of new data to dimension tables and the alteration of specific columns selected by the developer in the existing table data. This process uses the key to a row in the table to search for updates in the designated columns. Various types of slowly changing dimensions are commonly utilized, such as overwriting the record or, due to data changes, creating a new row to store both the historical and updated data. For this project just overwriting type of slowly changing dimensions are being used, and these are called changing columns.

For the fact tables, the data is more variable since many columns can change when a work order or service request alters its status. As these transactions involve numerous measures and the database aims to reflect the current status of the transactions, the project is implementing full loading. This requires truncating the entire table before reloading it whenever data is updated. This process ensures that the fact table contains the most up-to-date information regarding the transactions.

In figure 9 the flow of data from the source until the self-service BI is shown.

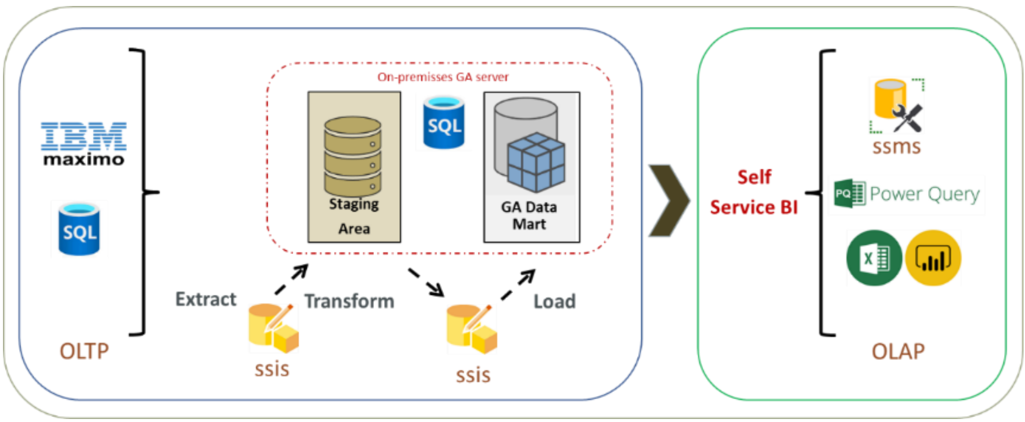


Figure 9 - Data Flow

4. EMPIRICAL STUDY

In the empirical study chapter, we explore the development of the ETL process of the project, which is a vital component for transforming and loading data for analysis. This section provides a detailed, step-by-step description of how the SSIS package was constructed, offering insight into the technical complexities of data integration.

4.1. Staging Area

In line with the methodology outlined in the previous chapter, the staging area serves as the initial repository for data, facilitating the extraction and transformation phases of the whole ETL process. These crucial steps are executed through a SQL Server Integration Services (SSIS) package within the Visual Studio Software environment.

This First SSIS package created for the staging area database is structured into two distinct phases. Firstly, a Sequence container contains the deletion of dimension tables and the truncation of fact tables. This task is accomplished by configuring individual "Execute SQL statements" tasks for each table as shown in figure 10.

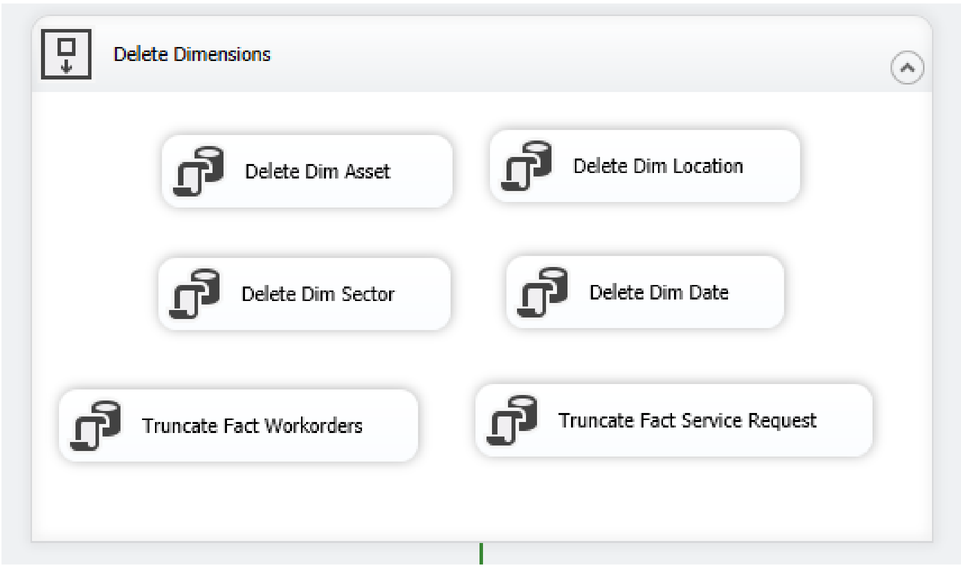


Figure 10 - Staging Area Sequence Container

The initial step of this phase is the deletion and truncation of tables, this activity is essential because the staging area does not retain historical data. Even though extracting and transforming data from the source into the staging area's database, it serves as a temporary repository. Consequently, upon each execution of the ETL process, the tables are emptied to accommodate both older and newer data uploads. The "Execute SQL Statements" task executes actions in the database based on SQL code, Figure 11 is an example of how the task is configured and the SQL code used.

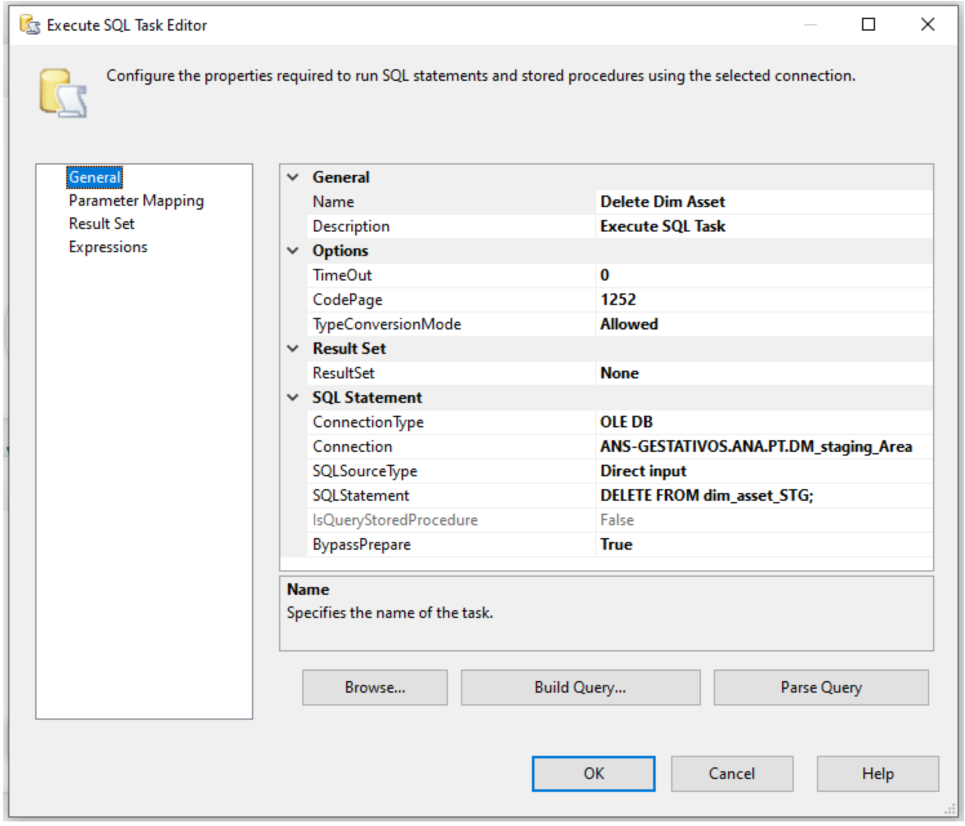


Figure 11 - Execute SQL Task Editor

In the case of a fact table, it makes more sense to use truncation instead of deletion. Truncation deletes everything in the table without creating logs, which means that it is an optimized method to erase information inside a table. Fact tables typically store large quantities of data, so using truncation helps improve performance and efficiency.

After completing the deletion process in the tables, the second phase, called "loading," begins. During this phase, data from the source is extracted, transformed, and then loaded into the tables of the staging area's database. To carry out these tasks within the SSIS package, a sequence container is required. Within this container, multiple "Data Flow Tasks" as shown in figure 12, each of which will be discussed in detail.

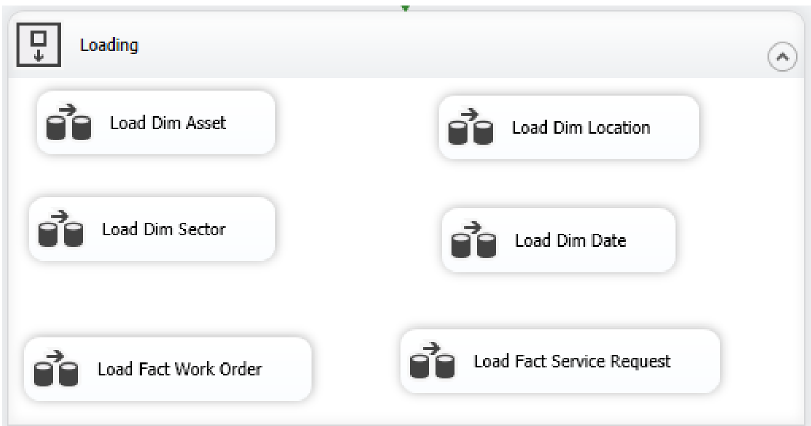


Figure 12 - Loading Sequence Container

4.1.1. Load Dimension Asset

The "Data Flow Task" facilitates the movement of data between databases using various tasks within it. In the case of loading data for the dimension "Asset," the process is straightforward, requiring only two tasks. The first task, "OLE DB Source," connects to the data source and extracts the necessary data. Subsequently, the second task, "OLE DB Destination," transfers the extracted data to the staging area database.

During this phase of the process, data extraction from the source happens. As there is only one source in this scenario, a SQL query was crafted to retrieve the relevant data destined for the specific tables designed within the DataMart, which is also present in the staging area.

The "OLE DB Source" task provides multiple means of accessing the source. For the project, the SQL command option was chosen for data access mode. The configuration of this task is illustrated in Figure 13.

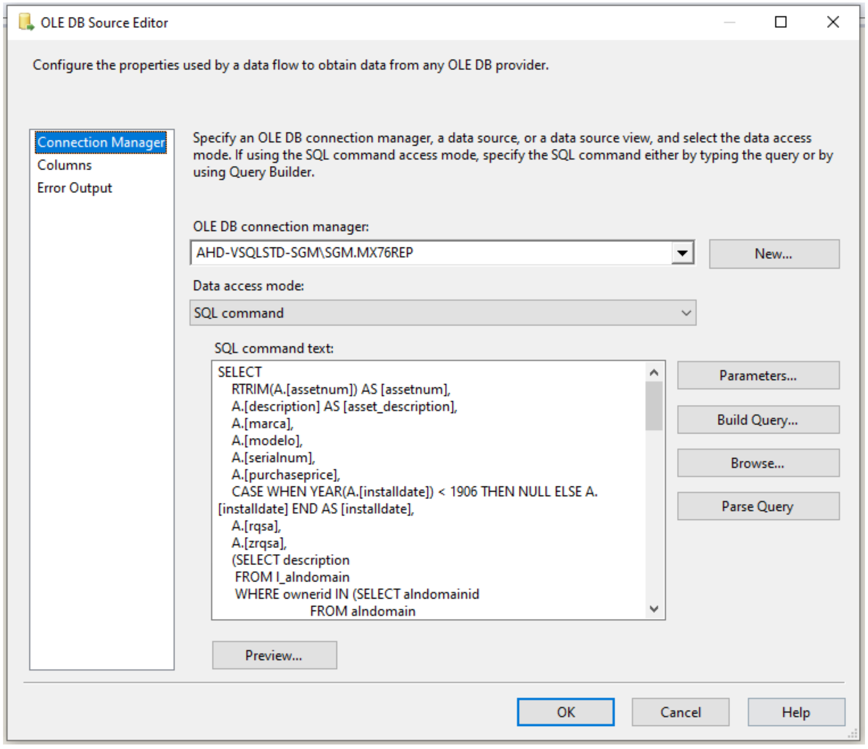


Figure 13 - Dim Asset OLE DB Source Editor

4.1.2. Load Dimension Location

The second dimension loaded into the Staging Area's database is the location dimension. Like the previous dimension, this is also developed within a "Data Flow Task", which contains both "OLE DB Source" and "OLE DB Destination" tasks. Data is retrieved from the source via an SQL code in the first task, and then sent to the destination in the Staging Area's database, the destination task will be further explained below.

The task “OLE DB Destination” allows the retrieved data to be stored in the destination table, but to achieve that, some configurations should be set up. First, it is necessary to choose the database and table where the data will be loaded. This can be done in the Connection Manager part of the "OLE DB Destination Editor". Figure 14 below shows the interface.

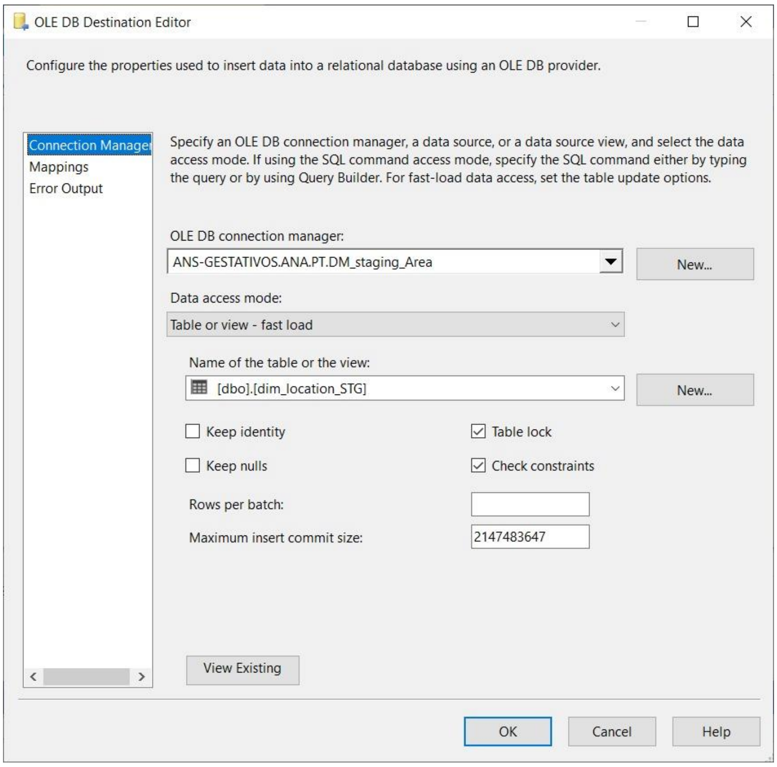


Figure 14 - OLE DB Destination Connection Manager

The second important step in the "OLE DB Destination" task is to configure the Mappings area. Here, the user can determine how the columns from the source will be mapped to the columns in the staging area's location table. This ensures that regardless of how the columns are retrieved from the source, they are appropriately directed to their corresponding destinations in the staging area's location table. This mapping's configurations are depicted in figure 15.

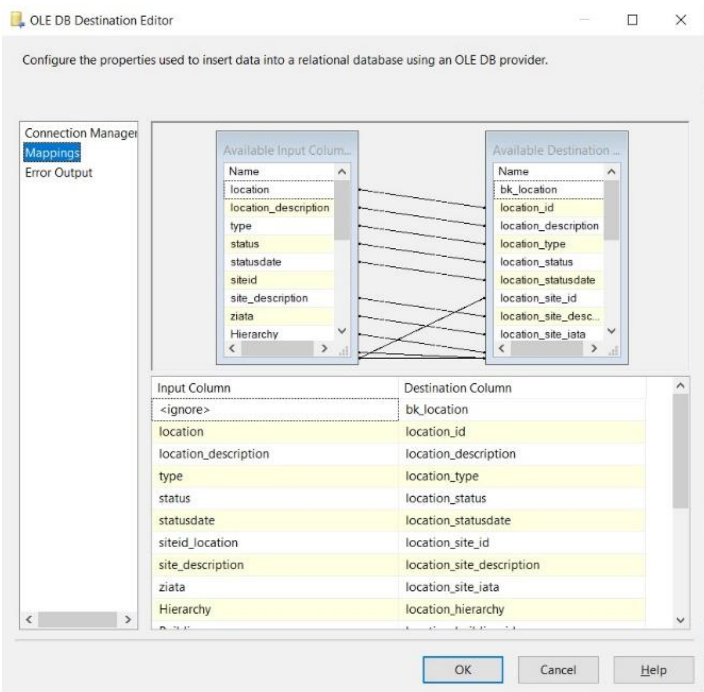


Figure 15 - OLE DB Destination Mappings

4.1.3. Load Dimension Sector

The "Data Flow Task" used for loading the dimension sector employs similar tasks as the preceding dimensions. It starts with the "OLE DB Source," responsible for extracting the data, followed by the "OLE DB Destination" task, which loads the extracted data into the table. Figure 16 illustrates the configuration of tasks within the "Data Flow Task."

Note that the transformation steps in these initial loading processes for all dimensions are minimal. Since the project utilized SQL queries to extract specific data from the source, which were used in the "OLE DB Source" task, the original data types of each column were preserved in the staging area's table design, simplifying the data flow.

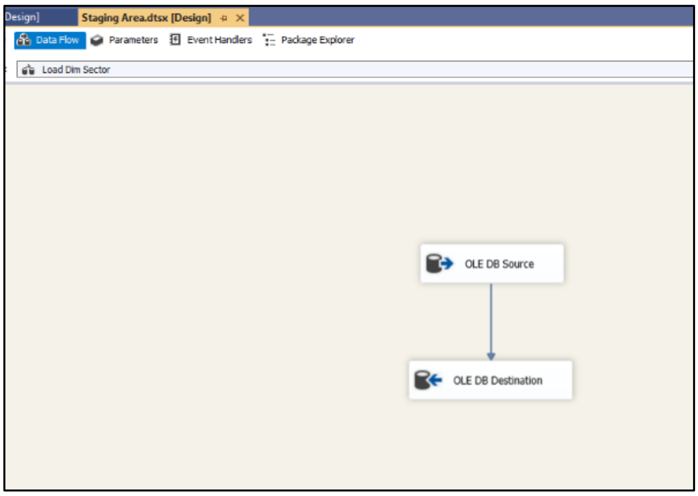


Figure 16 - Dim Sector Data Flow Task

4.1.4. Load Dimension Date

The dimension Date is unique in the fact that it wasn't extracted from the same source as the dimensions before, rather, it was generated using a Python code containing date information from 2009 to 2030, which was then exported to a CSV file. To streamline data access for the database, a table was created within the same SQL server housing the DataMart database. This decision was made because using a source located on a PC is not considered best practice for managing information. Consequently, the dimension was uploaded to a specific table within the database, from which the staging area extracts the data. An ETL package was developed for this purpose, allowing for modifications or additions to the data when necessary. After this process happened the ETL process was the same using and “OLE DB Source” task and a “OLE DB Destination” task. Figure 17 shows the connection to the database source of the dimension date.

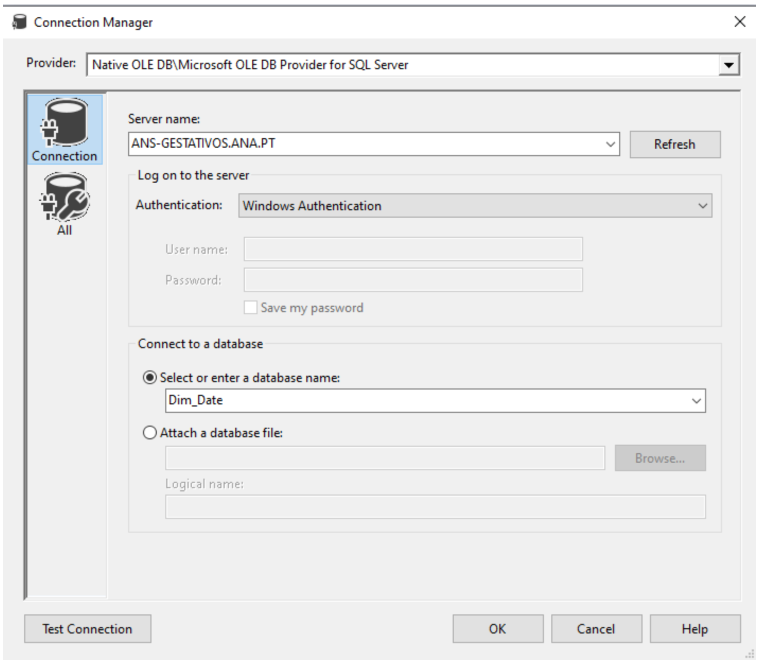


Figure 17 - Connection Manager

4.1.5. Load Fact Work Orders

The loading process for the fact table follows a similar process to the dimension tables. Initially, data is extracted using the "OLE DB Source" task and then sent to the table using the "OLE DB Destination" task. However, an additional task called "Derived Column" is introduced for the fact table loading process. This task can generate new columns based on original data retrieved and perform various transformations using functions, variables, and parameters.

In this specific case, the "Derived Column" task is used to identify "Null" values using the "ISNULL" expression. Subsequently, these "Null" values are replaced with blank spaces, zeros, or a specific column that is more appropriate for storing the data. The setting up of the “Derived Column” task is illustrated in figure 18.

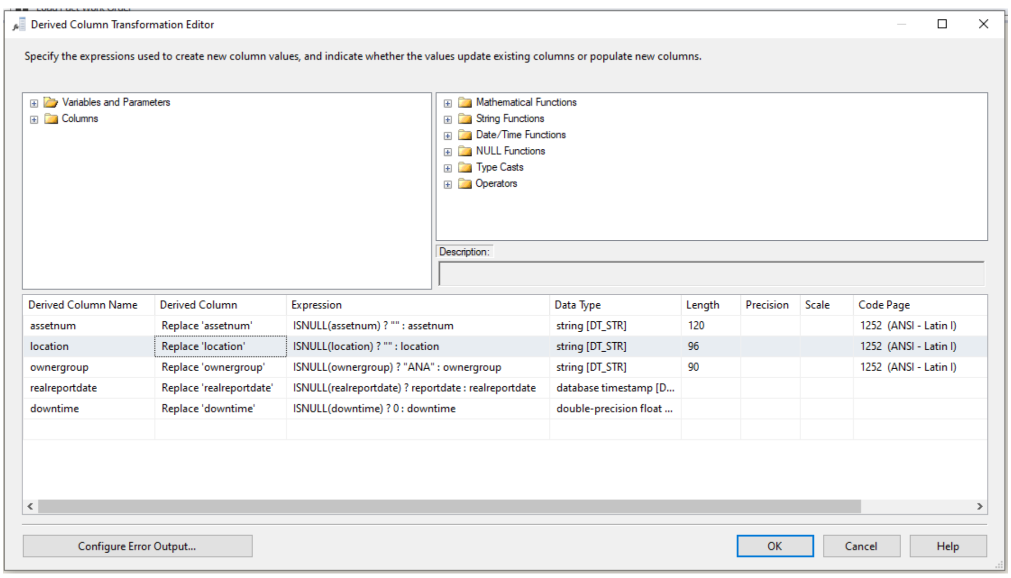


Figure 18 - fact work order Derived column transformation editor

4.1.6. Load Fact Service Request

Loading data into the fact table Service Request follows a similar pattern to the previously loaded tables. In this case, we introduce a new type of task called "Conditional Split." This task allows for setting conditions to filter the retrieved data. The "Conditional Split" task was added because there are some Service Requests that lack a Sector Id. Since this data cannot be stored in the DataMart, it was deemed unnecessary after consultation with area specialists, as incomplete Service Requests do not contribute value to the metrics and evaluations. The configuration of the data flow task is shown in figure 19.

Additionally, a "Derived Column" task was configured to handle "Null" values in the "sr_assetnum" column, replacing them with blank spaces.

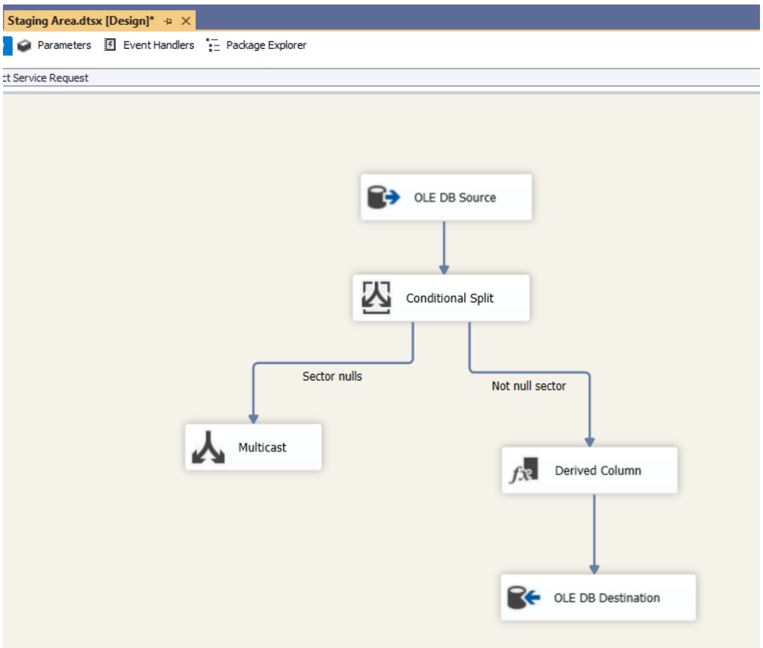


Figure 19 - Load fact service request data flow task

4.2. Data Mart

The final database where the data will be loaded and related between tables is the DataMart database. Since extraction and transformations of the data have already occurred in the staging area, the purpose of this final database is to store historical and accurate information. To achieve this, the loading process needs to assess changes in the data through slowly changing dimensions, which are responsible for loading only the new data.

The "Data Mart" SSIS package consists of three sequence containers. The first one, named "Dimensions Loading," handles the loading of the dimensions. The second sequence container, "Fact truncation," is responsible for deleting data in the fact tables before loading them with new information. Finally, the third sequence container, "Fact Loading," manages the movement of data into the fact tables. To update the tables with new data, the package employs both incremental loading and full loading. Incremental loading updates only the new changes in the uploaded data, while full loading brings the entire table, including historical and new data as shown in figure 20.

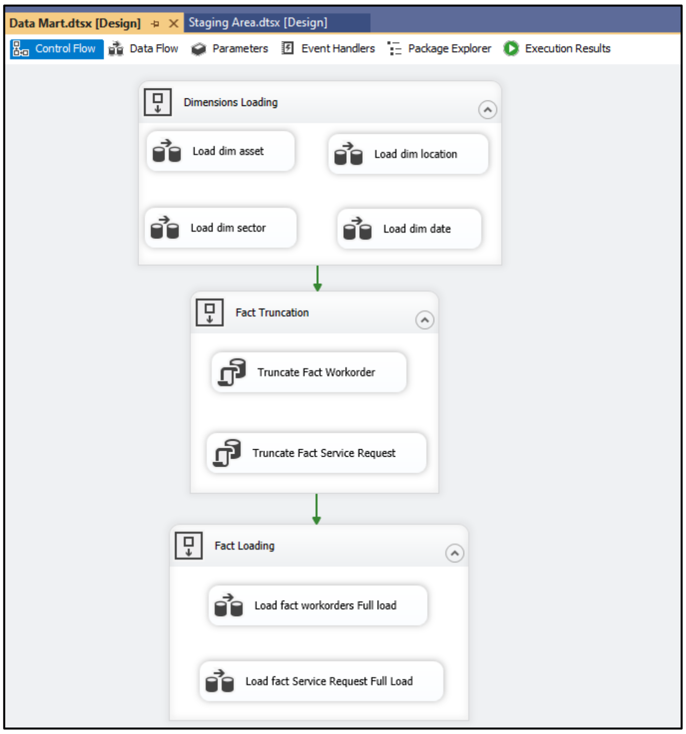


Figure 20 - Data Mart SSIS Package

4.2.1. Load Dimension Asset

In general, loading the various dimensions involves a "Data Flow Task" with an "OLE DB Source" task and a "Slowly Changing Dimension" task. The first task retrieves structured data from the staging area's database, while the "Slowly Changing Dimension" task handles updates and setting up the destination within the same task.

Within the "Slowly Changing Dimension" task, the new inputs that need to be loaded into the table and specific columns that can be updated in an existing row are defined. This task operates based on a business key column, scanning through all the data in the Asset dimension table of the staging area. It checks for new updates in the lines based on the business keys and adds new lines for nonexistent business keys, updating both cases in the table. In figure 21 the whole data flow task is illustrated.

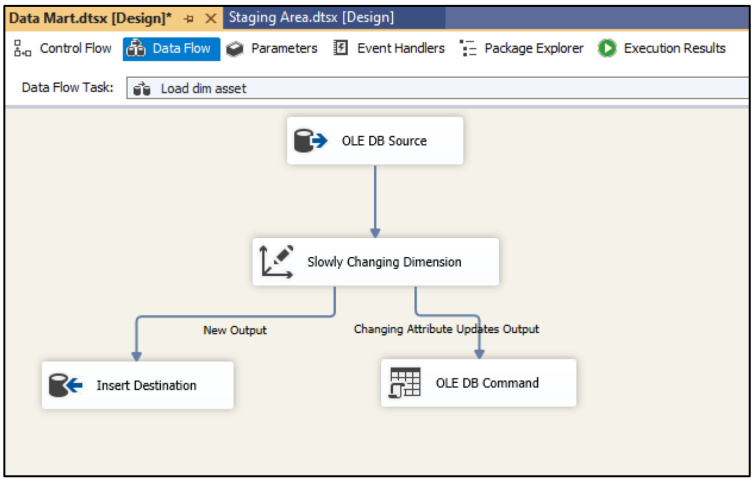


Figure 21 - Dimension Asset Data Flow Task

4.2.2. Load Dimension Location

As the process for loading data into the table is similar across every dimension, in the case of the location dimension, the study will highlight one of the most important aspects of the "Slowly Changing Dimension" task to set up, which is the change type of columns in the table. There are three options for change types for columns in this task of the SSIS package. Fixed attribute, when the value in the column should not change, and if the task detects an error, it will display an error and halt the ETL process. Next, changing attribute, in this case, the values of this column can change, and if that's the case, the values would be overwritten. Finally, there is the Historical attribute, which is used when there is a need to maintain historical changes. Thus, a new entry is created, but information regarding data timestamps should be configured. This project is not using historical attributes within the "Slowly Changing Dimension" tasks. In figure 22 the configuration of type of columns for the slowly changing dimension task is shown.

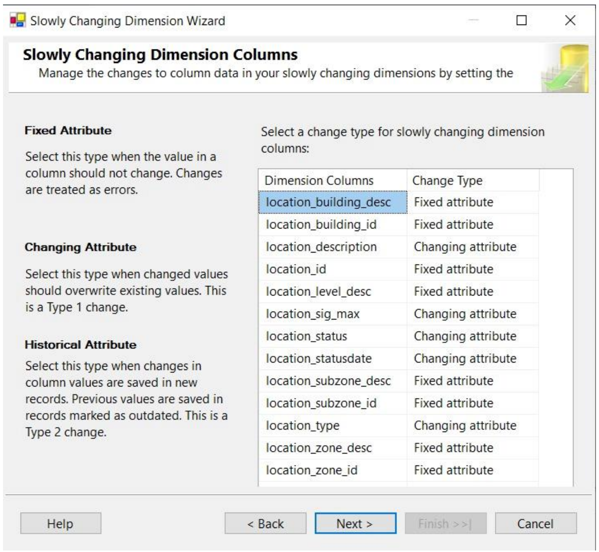


Figure 22 - Location SCD Columns Set Up

4.2.3. Load Dimension Sector

The process of tasks used for loading data into the dimension Sector follows the same pattern as the previous ones: an "OLE DB Source" task followed by a "Slowly Changing Dimension" task. However, one aspect that needs to be addressed in the loading of dimensions is the specific handling of business keys and surrogate keys. As mentioned earlier, surrogate keys are only used in the DataMart database. Therefore, these keys are automatically generated each time a new row is inserted during the data transfer from the staging area to the DataMart's database, based on the SQL code that defines the table in the database. Figure 23 shows the code for the creation of a primary key in a SQL table.

```sql
CREATE TABLE dim_sector (
  sk_sector INT identity(1, 1) PRIMARY KEY,
  bk_sector VARCHAR(90) NOT NULL,
  sector_description VARCHAR(200) NULL,
  sector_level_2 VARCHAR(90) NULL,
  sector_level_3 VARCHAR(90) NULL,
  sector_level_4 VARCHAR(90) NULL,
  sector_level_5 VARCHAR(90) NULL,
  sector_level_6 VARCHAR(90) NULL,
  sector_level_7 VARCHAR(90) NULL,
  sector_level_8 VARCHAR(90) NULL,
  sector_siteid VARCHAR(16) NULL,
  sector_cost_center VARCHAR(20) NULL,
  sector_child SMALLINT NOT NULL,
  sector_active SMALLINT NOT NULL,
  sector_inactive SMALLINT NOT NULL
);
```

Figure 23 - SQL code Dimension Sector

That is why the mapping of surrogate keys in the "OLE DB Destination" task does not require a selected input column. The setup for mapping during the loading of the dimension sector is depicted in figure 24.

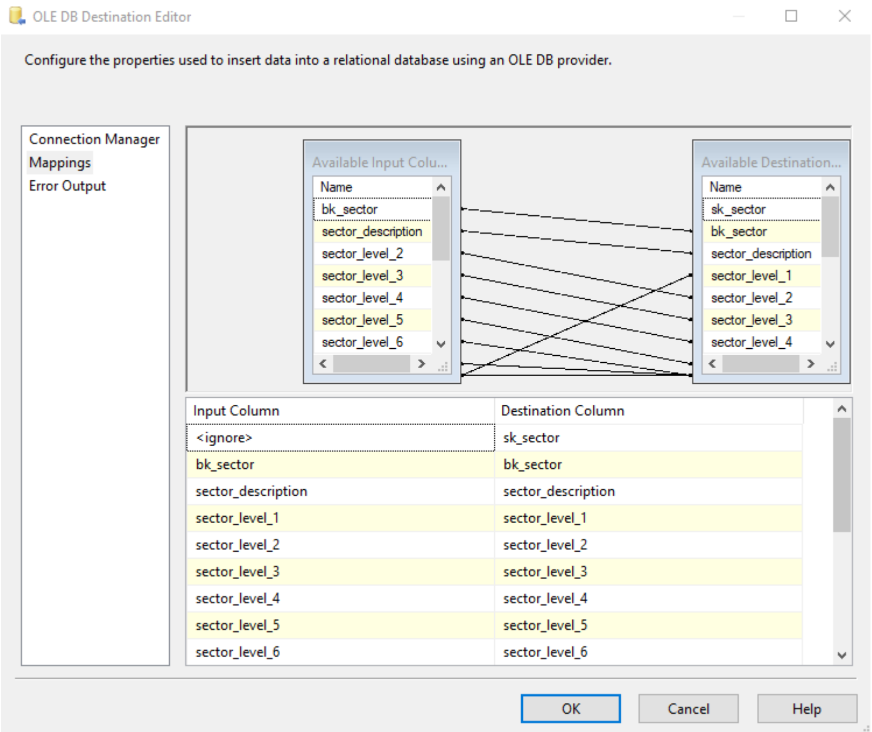


Figure 24 - Dim Sector OLE DB Destination Mappings

4.2.4. Load Dimension Location

The dimension date is the final table uploaded through the slowly changing dimension task, allowing for incremental loading. While this dimension doesn't undergo frequent changes since it's designed to meet the transactional needs regarding dates, incremental loading is configured through the slowly changing dimension in case any date is updated or added.

4.2.5. Fact Truncation

Fact tables use a full loading approach when updating stored information. This means the process resembles that of the staging area, where all data within the table is truncated and reloaded. This method was selected because numerous columns within the table could potentially change, and employing incremental loading would have been excessively resource-intensive. The fact that dimension loading runs without errors during the ETL process ensures that the relationship between dimension and fact tables remains intact. Therefore, employing a full load does not pose any issues.

The first step for the loading of the fact tables is truncating the existent data within the tables, to accomplish that a "Sequence Container" task is set up with two "execute SQL Task", one for each fact table as shown in figure 25.

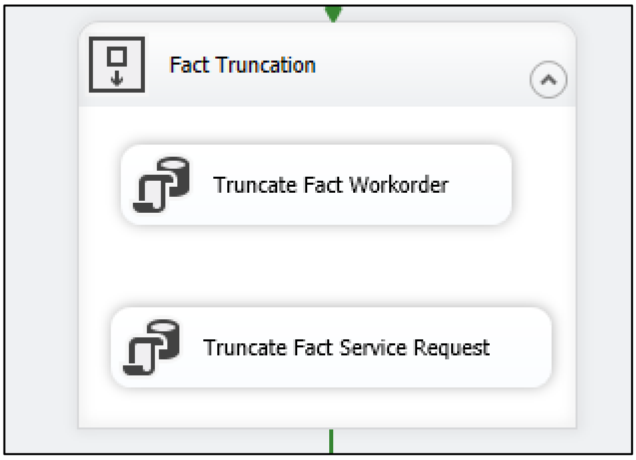


Figure 25 - Fact Truncation Sequence Container

4.2.6. Load Fact Work Orders

The 'Data Flow Task' contains various tasks facilitating the movement of data from the staging area's database to the tables in the DataMart database. Initially, the 'OLE DB Source' task is employed to extract data from the source table. Subsequently, 'Lookup' tasks are utilized for each dimension. These tasks establish relationships between the surrogate key of the dimension and the foreign key of every transaction in the fact table. Additionally, a 'Multicast' task is configured to handle cases where there are no matches between the surrogate keys and foreign keys, redirecting the lookup results accordingly. Furthermore, a 'Data Conversion' task is incorporated to ensure compatibility between the data types of the surrogate key of the dimension date and the foreign key of the fact table. This step precedes the lookup process for the date key. Finally, the 'OLE DB Destination' task is employed to define the column mappings, ensuring that all data is accurately directed to their respective locations in the new table. Figure 26 shows the tasks set up within the data flow task of the fact work orders loading.

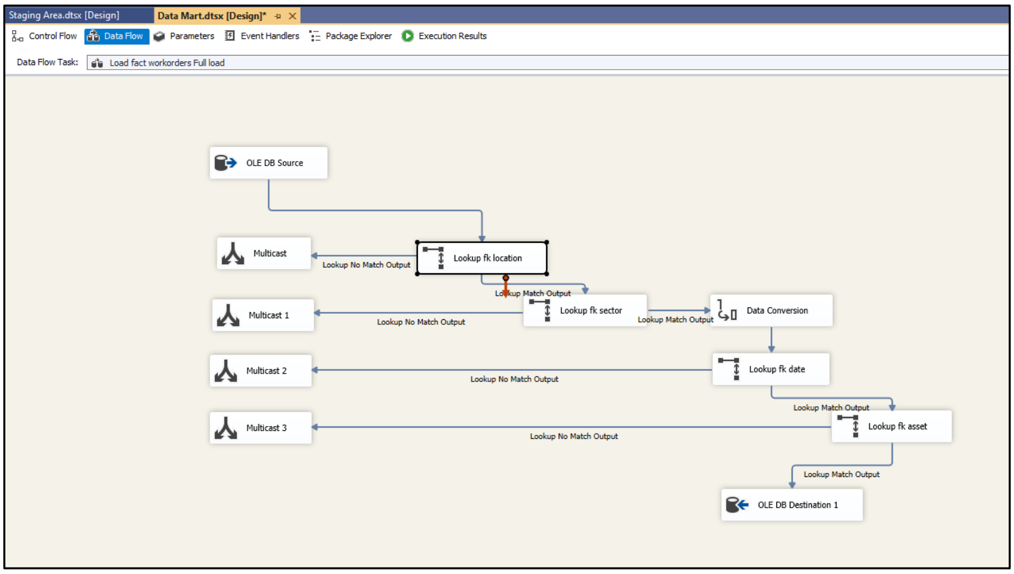


Figure 26 - Fact Workorder Data Flow Task

4.2.7. Load Fact Service Request

The loading process for the Service Request fact tables follows the same approach as the Work Orders loading, given their shared relationship with the dimension tables. Therefore, the "lookup", "OLE DB Source" and "OLE DB Destination" tasks remain consistent across both fact tables.

A specific configuration was applied to the "lookup" task of both fact tables, particularly in the dimension Location lookup. Here, the "Cache Mode" was adjusted to "Partial cache." This adjustment was necessary due to an error encountered during data movement, indicating a violation of primary key constraints, specifically involving the repetition of transaction lines, notably the foreign key of the location. In figure 27 is shown how to choose this configuration within the look up transformation task.

The "Full cache" mode stores the entire table in cache from which data is retrieved, which can lead to various issues such as errors and line duplication. In contrast, the "Partial cache" mode enhances the lookup process efficiency by loading only a portion of the table at a time. This modification resolved the error related to the location´s foreign key, ensuring a smoother lookup process.

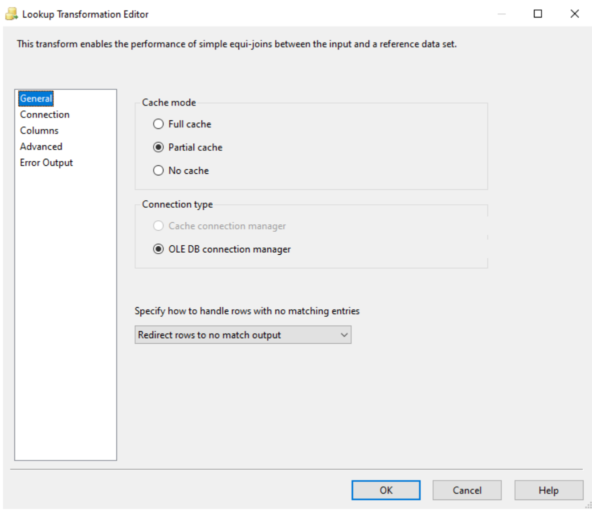


Figure 27 - Lookup Transformation Partial Cache

4.2.8. ETL Performance

The ETL process runs once every day, corresponding to the daily updates in the data source. In terms of performance, there haven't been any major issues with loading new data batches. However, it is worth noting that running the packages while connected to a VPN results in approximately 20 times longer processing times. Additionally, an unstable internet connection can disrupt the ETL process. Care must also be taken with user credentials, if passwords are changed, the package may fail to connect to the source for data extraction. On average, both the "Staging Area" and "Data Mart" packages to run, combined, is around 45 minutes. Figure 28 illustrates the achievement of the ETL process .

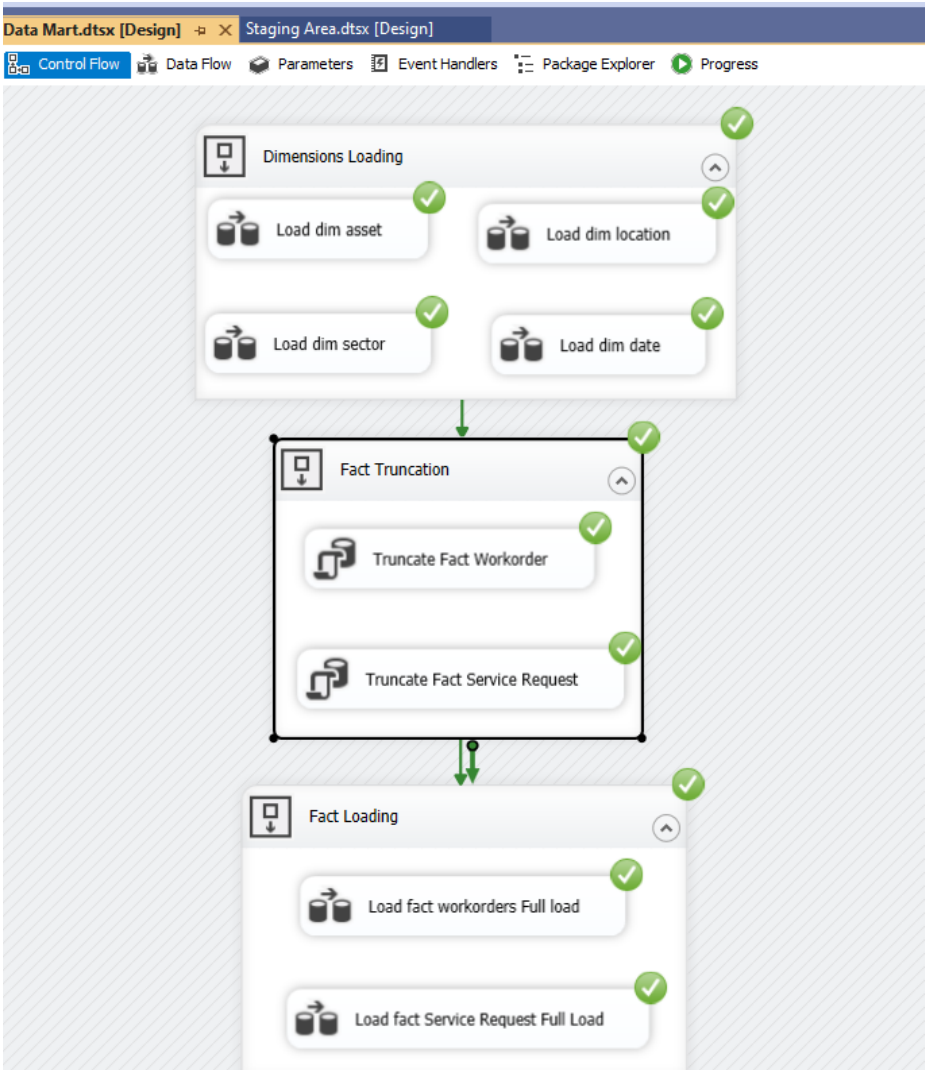


Figure 28 - Loaded DataMart

5. RESULTS & DISCUSSION

The project was conceived to design an efficient data flow that enables the asset management area to leverage their data for enhanced decision-making capabilities and provide dynamic access to information. The objectives were outlined in the initial chapter of the research.

5.1. OBJECTIVES

There are three objectives that were achieved to fulfill the research goals. These objectives outline the sequence of tasks to be followed when designing data storage, specifically a Data Mart.

Objective 1: Integrate an OLTP system into a Data Mart, it was technically achieved by extracting data from the OLTP source. This was based on the business questions identified through the Goal-Question-Metric Approach methodology. Once the metrics necessary for the DataMart were determined, the model tables were developed and created on the server. The extraction process occurred at the outset of the ETL process, using SQL codes to retrieve specific variables and measures for the dimensions and fact tables. Conceptually, this objective aimed to define the data extraction requirements from the OLTP system, known as "Maximo," and to design a relational model capable of storing the data, enabling specialists from the Asset Management area to generate insights and make informed decisions.

Objective 2: Establish the ETL processes. This objective is involved in executing the technical steps to extract, transform, and load data from a source to a designed database using Visual Studio software and an SSIS Package. This process was divided into two main parts: the staging area, where extraction and transformation activities take place, and the final database, where the loading activity occurs to store both historical data and new updates, ensuring clean and up to date information in the DataMart database. Essentially, this objective aimed to achieve the technical aspect of selecting data from a source and storing it in a new database to function as an OLAP system for the Asset Management area.

Objective 3: Delivering the solution to internal clients for further data analysis, this objective marks the final step in achieving the objectives, focusing on how to provide the data product to users, often referred to as "Self-Service BI." As illustrated in Figure 8, users can access the information stored in the DataMart through various means such as SQL Server Management Studio, Power Query, Power BI, and Excel, among others available for accessing a SQL server database. A summary of the results achieved by objective is shown in table 12.

Table 12 - Results

Objective	Result
Integrate an OLTP system into a Data Mart	Generation of business questions Kimball Model development Creation of the databases for the DataMart
Establish the ETL processes	Connection to the EAM system’s database to extract the defined data. Data transformation through staging area. Loading of data into the DataMart’s database
Delivering the solution to internal clients for further data analysis	Creation of a User friendly and dynamic Dashboard that answers the predefined business questions.

During the development of the business questions, it became apparent that a crucial report aiding decision-making in Asset Management is the EAM's life cycle analysis report. Leveraging the DataMart as a source, a "Life Cycle Analysis" Dashboard was created. Subsequently, the following section of the chapter will link the research gap which are the business questions with the developed dashboard to address the identified research gaps.

5.2. RESEARCH GAP

As mentioned in the introduction chapter, defining the scope of the extracted data from the EAM system, which serves as the source for the DataMart, relied on the business questions. These questions were crucial in determining the metrics to be stored in the DataMart. By analyzing the "Life Cycle Analysis" dashboard, the answers to these business questions will be revealed.

The Life Cycle Analysis Dashboard was designed after the report of the same name detailed in Chapter 3.1.4 of the research. Consequently, the dashboard comprises technical, organizational, and economic data concerning assets and maintenance activities. To ensure usability and comprehensibility for internal clients, the dashboard has been segmented into four sections: Technical Interventions, Technical KPIs, Labor, and Costs. These dashboard pages have the crucial information from the EAM report while allowing users the convenience of dynamic analysis options.

5.2.1. Business Questions

How good or bad is the performance of the assets? is the first business question, and this can be addressed by examining various visualizations available in the Life Cycle Analysis Dashboard. For instance, the Technical KPIs page offers insights into the performance of individual or aggregation of assets as shown in table 29.

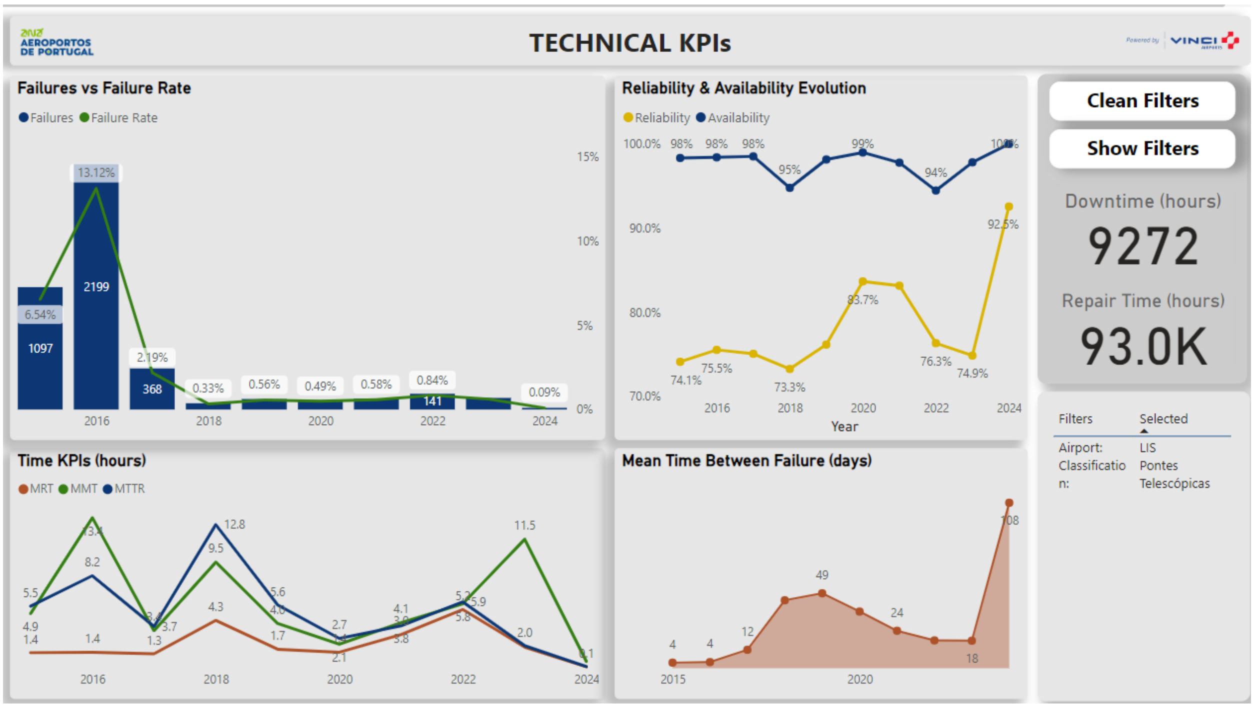


Figure 29 - Technical KPIs

This page of the dashboard displays various key performance indicators (KPIs) commonly used by maintenance specialists to assess asset performance. Firstly, users can view the frequency of asset failures or aggregated failures over time. Additionally, Availability and Reliability KPIs provide insights into the proportion of available time during which assets have not been functioning correctly. The third visualization depicts the trends in KPIs such as mean repair time, mean maintenance time, and mean time to repair. Lastly, the fourth visual presents the evolution of the mean time between failures. By presenting this information to asset management and maintenance specialists, it enables them to evaluate assets' performance and trends over time allowing to understand how failures may impact operations.

The second business question is, *what are the costs related to the maintenance of the assets?* In this case there is a specific page dedicated to analyze how much money was invested in the maintenance of the asset or aggregation of assets which is the "Cost" Page of the Dashboard.

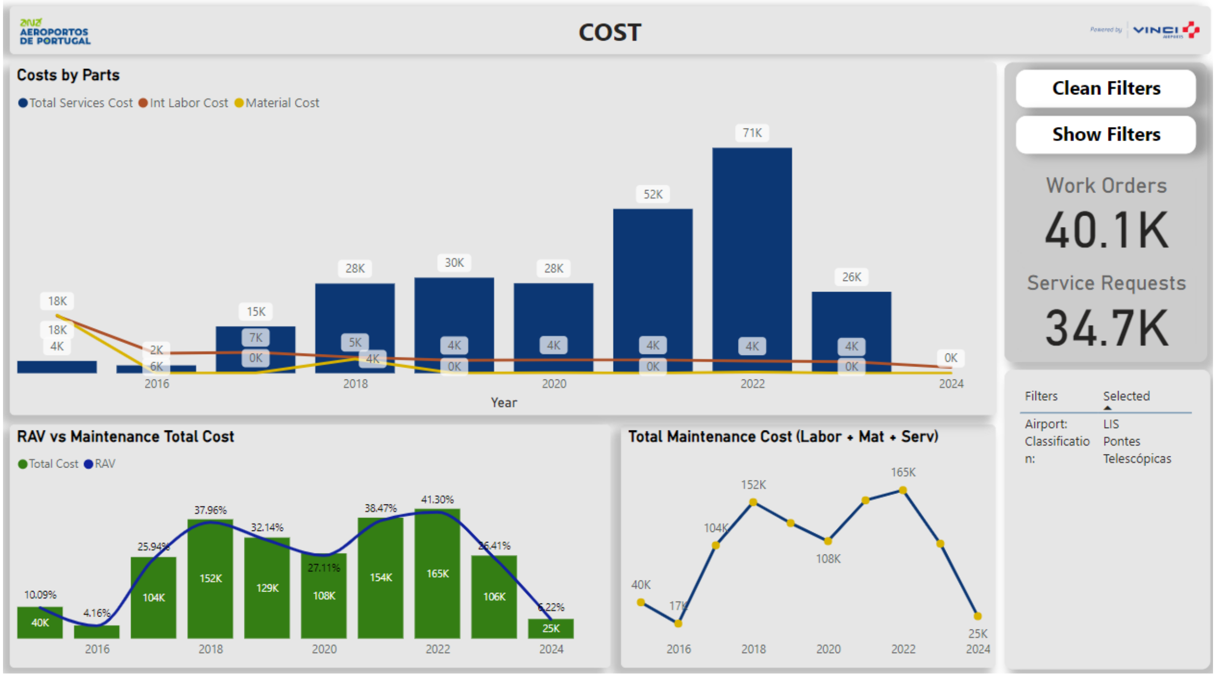


Figure 30 - Dashboard Cost Page

As shown in figure 30, the "cost" page of the dashboard, the visualizations illustrate to users how costs are allocated, encompassing various categories such as expenses and different types of labor. The first bar chart delineates the distribution of costs in terms of monetary value and different types of labor. The second visualization juxtaposes the total expenditure on maintenance with the Replacement Asset Value KPI. Lastly, there is a summary of the total expenditure on maintenance, which encompasses labor, material, and service costs.

The third question is, *does every asset have a life cycle analysis?* Answering this question can be challenging without supporting data. The only way to address it using the Life Cycle Analysis Dashboard is by referring to the "Technical Interventions" page, shown in figure 31. This allows us to determine if a specific asset has undergone interventions during its "lifetime." If there is no information about interventions when a particular asset is selected, it indicates that no interventions have occurred, and therefore, there is no data available to analyze its performance.

The visualizations on this page provide insight into the frequency of interventions on assets. The first chart displays the number of interventions by type, which could include corrective, preventive, or technical service maintenance. Next, the corrective maintenance ratio KPI assesses the proportion of corrective maintenance work orders compared to the total number of work orders or interventions. This helps analysts gauge the effectiveness of the maintenance plan. Lastly, the "Work Order Type" visualization illustrates the distribution of work order types over time, allowing for trend analysis of interventions.

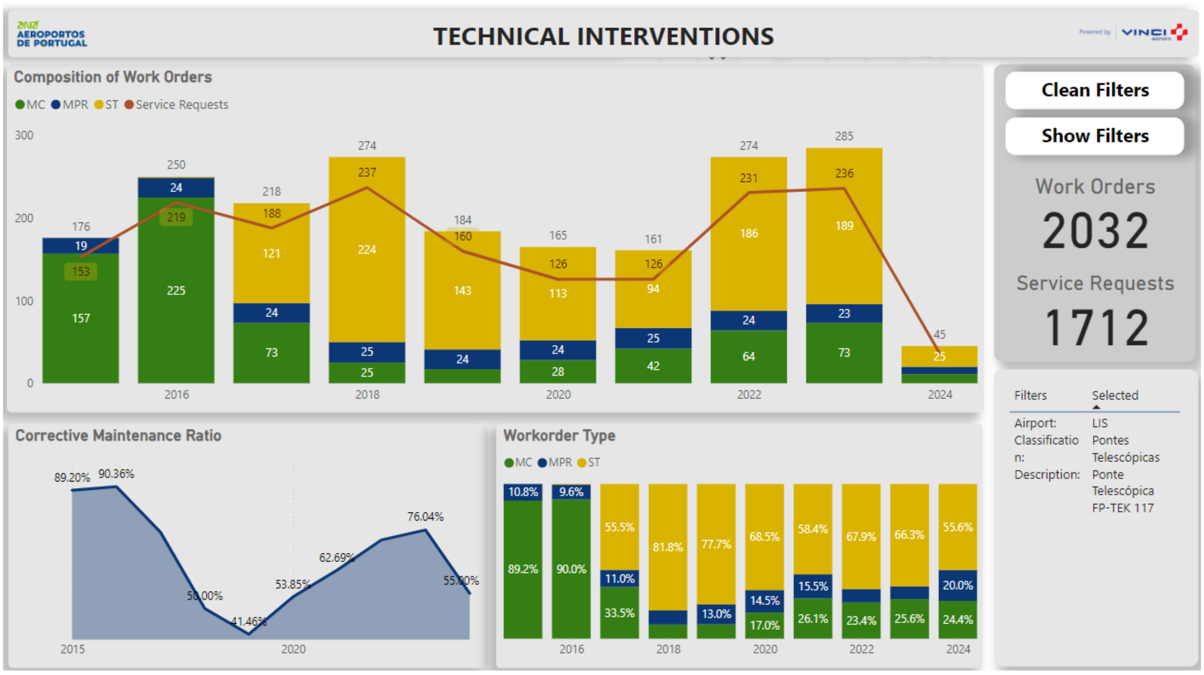


Figure 31 - Dashboard Technical Interventions

The last business question that was found is, *when to choose between replacement and rehabilitation of an asset?* Answering this question requires analyzing the entire life cycle of an asset, considering variables such as the number of interventions, maintenance costs, time spent on maintenance, and asset performance. These factors, combined with the expertise of the analyst interpreting the data, inform the decision to replace or retain the asset. However, a specific KPI proves particularly useful in this context when assessing costs. The Replacement Asset Value (RAV) KPI, shown in figure 32, this metric calculates the percentage of economic resources invested in maintenance relative to the original price of the asset. Thus, this comparison aids in determining whether the funds allocated to maintenance could have instead covered the cost of a new asset.

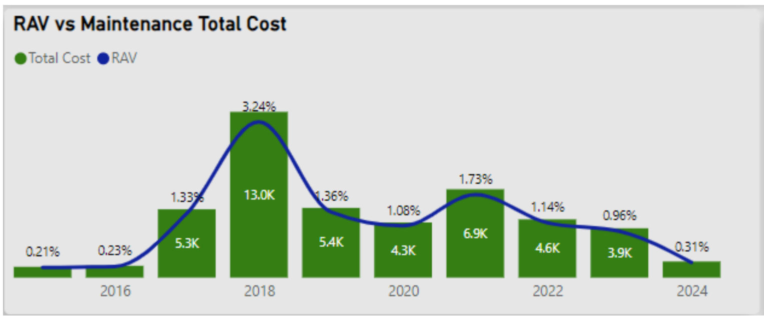


Figure 32 - RAV KPI

To provide insight into whether an asset should be replaced or rehabilitated, it's crucial to understand the amount of time technicians dedicate to maintenance tasks on that asset. To furnish this information, the "Labor" page was developed as part of the Life Cycle Analysis dashboard, illustrated in figure 33.

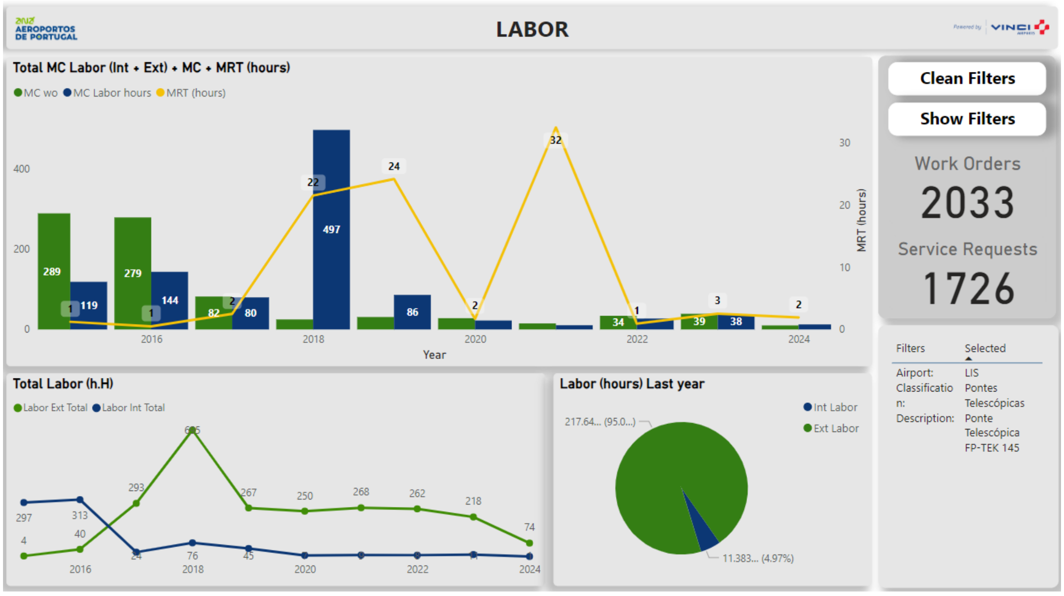


Figure 33 - Dashboard Labor Page

This page initially assesses the quantity of labor hours invested in corrective maintenance tasks in relation to the number of corrective maintenance work orders, along with the mean repair time. By combining these three metrics, users can comprehend the time investment in assets due to failures. Additionally, a line chart illustrates the total labor hours, categorized into internal and external, representing tasks carried out by the maintenance team or external companies hired to maintain assets. Finally, a pie chart provides statistics on labor hours from internal and external interventions over the past year.

5.3. DISCUSSION

This study aims to provide readers with a comprehensive overview of designing a database for the Asset Management area. It delves into the decision-making process that the developer of the Data Base needs to do regarding development approaches, architecture, model representation, and ETL processes. While existing literature offers guidance on creating databases, this study serves as a step-by-step manual for designing, implementing, and using the information repository. The study presents a methodology that could be applied across industries. As there is a lack of specific information regarding the design of a database for the Asset Management area, the research highlights its importance in driving the transition towards data-driven enterprises. Ultimately, this study seeks to emphasize the transformative impact of integrating decision support systems within any industry.

Having a data architecture that enables team members to make data-driven decisions optimizes various processes. Typically, many analysts spend a significant amount of time searching for and cleaning data to perform their work effectively. However, having clean and organized information available streamlines any required tasks and contributes to the organization's transformation into a data-driven entity.

6. CONCLUSIONS & FUTURE WORKS

6.1. CONCLUSIONS

This work has developed a guide on how to design a database to enhance decision-making within an industry sector. While this project focuses on the "asset management" area, the research is intended to be applicable to any area or team.

The study integrates various methodologies to assist in decision-making throughout the entire process. Initially, the type of database suitable for the area's needs was assessed, opting for a data mart in this case. Subsequently, with the team members' input, it was determined what data is going to be retrieved from the EAM system. Following this, the ETL process was developed to transfer the selected data to the DataMart. Finally, recognizing that users or team members may lack SQL database querying skills, a data product was developed in the form of a Power BI Dashboard containing key metrics essential for asset management.

This cycle of data flow within an area, from data extraction to user access and insights generation, completes the transformation of data into actionable information. Therefore, it is concluded that we have successfully designed a decision support system, which is the result of effective information management development.

6.2. LIMITATIONS

During the development of the study, limitations were found. Initially, it proved challenging to define the pertinent business questions required to select the data to be retrieved and loaded into the data repository. Designing a database for a specific unit of work presents inherent complexities. However, by adopting a methodology and collaborating with the future users of the information stored in the database, we were able to identify the primary metrics required, After the database was launched, users requested the addition of more metrics, including those in both dimensions and facts. For instance, the Service Request fact table was a later addition to the model. This indicates that the database is in a constant state of growth to accommodate evolving user needs and expanding data requirements.

Another limitation encountered during the project was the selection of the appropriate software for developing the ETL process. Nowadays, there are various applications available to accomplish this task. In this case, several variables were considered when making this decision, including cost for the company, developer experience, the need for robust software, and compatibility with the organization's systems. After seeking assistance from the IT department and the thesis tutor, the decision was made to use Visual Studio with SSIS packages.

Other significant challenges we encountered were obtaining permissions to access the EAM system database for data extraction and acquiring a server to host the database and store the information, whether on-premises or in the cloud. During the initial stages of an organization's digital transformation journey, it can be challenging to recognize the value that a data architecture can bring to the company and its decision-making processes. Therefore, obtaining the necessary tools to build and host the database required collaboration and support from various departments within the organization.

Finally, despite employing a design thinking methodology to identify business questions, not all were used in designing the data mart. During the exercise, team members suggested ideal information without considering the available data collected in the EAM system. Consequently, the study had to limit the number of business questions to just four. However, the data repository and dashboard can dynamically and continuously address these selected questions.

6.3. Future Work

The asset management team became more familiar with utilizing and recognizing the value provided by the database for decision-making based on information, new inquiries emerge. There are constantly evolving metrics that the team seeks to understand. Fortunately, obtaining new information from the source is straightforward, it requires checking the granularity and incorporating the metrics or data into the fact tables.

An example of these emerging data needs is a newly developed dashboard, utilizing data stored in the DataMart but not previously utilized in the "life cycle analysis" dashboard. This new development is called "Root-Cause Analysis", shown in figure 34. Which essentially provides information about predefined symptoms, problems, causes, and actions related to corrective maintenance work orders. This data is used to generate statistics on the root causes of asset´s failures.

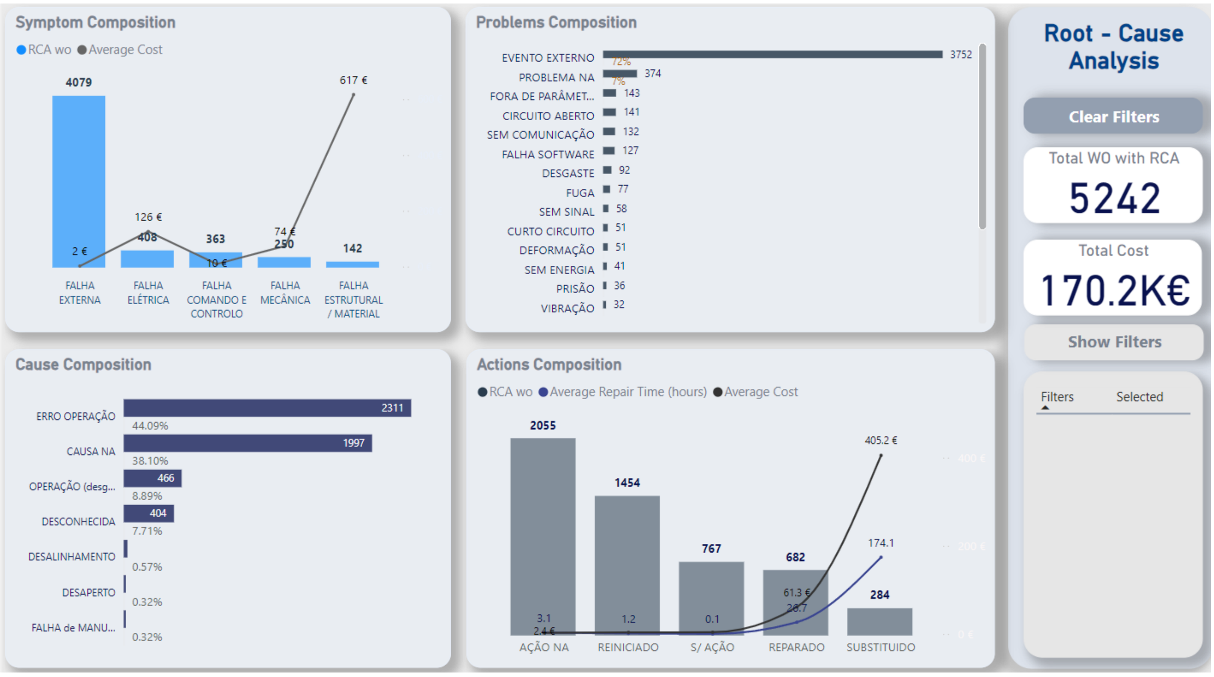


Figure 34 - Root-Cause Analysis Dashboard

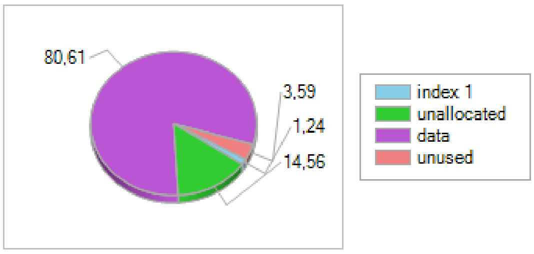
Summarizing potential future tasks for the database includes adding new columns or variables to the fact or dimension tables. Using this new information, knowledge can be generated through dashboards or various self-service BI tools utilizing the database as a source. Additionally, another important consideration is database maintenance and vigilance over disk usage. Insufficient disk space could potentially stop the ETL process when loading new data batches. In figure 35 the disk usage of the staging area database is shown.

Disk Usage
[DM_staging_Area]
on ANS-GESTATIVOS at 03/05/2024 16:39:13

This report provides overview of the utilization of disk space within the Database.

Total Space Reserved	2,08 GB
Data Files Space Reserved	1 096,00 MB
Transaction Log Space Reserved	1 032,00 MB

Data Files Space Usage (%)



Transaction Log Space Usage (%)

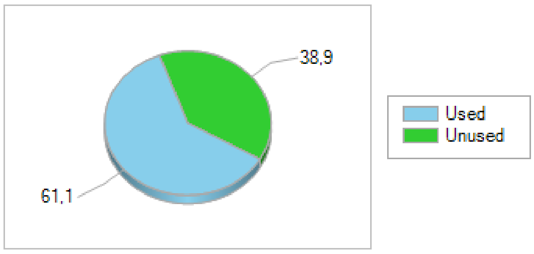


Figure 35 - DataMart Disk Usage

BIBLIOGRAPHICAL REFERENCES

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). *THE GOAL QUESTION METRIC APPROACH*.

Campos, J., Sharma, P., Gabiria, U. G., Jantunen, E., & Baglee, D. (2017). A Big Data Analytical Architecture for the Asset Management. *Procedia CIRP*, *64*, 369–374. https://doi.org/10.1016/j.procir.2017.03.019

De Assis Vilela, F., Times, V. C., De Campos Bernardi, A. C., De Paula Freitas, A., & Ciferri, R. R. (2023). A non-intrusive and reactive architecture to support real-time ETL processes in data warehousing environments. *Heliyon*, *9*(5), e15728. https://doi.org/10.1016/j.heliyon.2023.e15728

Duarte, J. C., Cunha, P. F., & Craveiro, J. T. (2013). Maintenance Database. *Procedia CIRP*, *7*, 551–556. https://doi.org/10.1016/j.procir.2013.06.031

Garani, G., Tolis, D., & Savvas, I. K. (2023). A trajectory data warehouse solution for workforce management decision-making. *Data Science and Management*, *6*(2), 88–97. https://doi.org/10.1016/j.dsm.2023.03.002

Jaleel, R. A., & Abbas, T. M. J. (2020). Design and Implementation of Efficient Decision Support System Using Data Mart Architecture. *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 1–6. https://doi.org/10.1109/ICECCE49384.2020.9179313

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley. https://books.google.pt/books?id=WMEqTf2lK84C

Litcanu, M., Prostean, O., Oros, C., & Mnerie, A. V. (2015). Brain-Writing Vs. Brainstorming Case Study For Power Engineering Education. *Procedia - Social and Behavioral Sciences*, *191*, 387–390. https://doi.org/10.1016/j.sbspro.2015.04.452

Mathew, A., Ma, L., & Hargreaves, D. (2009). *A CONCEPTUAL DATA MODELLING METHODOLOGY FOR ASSET MANAGEMENT DATA WAREHOUSING*.

Moody, D. L. (2000). *From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design*.

Osladil, M., & Kozubik, L. (2015). Smart Asset Management in view of recent analytical technologies. *2015 16th International Scientific Conference on Electric Power Engineering (EPE)*, 60–62. https://doi.org/10.1109/EPE.2015.7161188

Pantelias, A. (2005). Asset Management Data Collection For Supporting Decision Processes. *2005*.

Schneider, J., Gaul, A. J., Neumann, C., Hogräfer, J., Wellßow, W., Schwan, M., & Schnettler, A. (2006). Asset management techniques. *International Journal of Electrical Power & Energy Systems*, *28*(9), 643–654. https://doi.org/10.1016/j.ijepes.2006.03.007

Sharda, R., Delen, D., & Turban, E. (2018). *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*. Pearson. https://books.google.pt/books?id=kIlbvgAACAAJ

Turban, E. (2015). *Business intelligence and analytics: Systems for decision support* (Tenth edition). Pearson.

Xiaofeng He, Gang Wang, & Jiancang Zhao. (2005). Research on the SCADA /EMS System Data Warehouse Technology. *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific*, 1–6. https://doi.org/10.1109/TDC.2005.1547055