INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Implementing a Data Integration Infrastructure for Healthcare Data: A Case Study

Miguel Pratas Ferreira Mira

Master's in Integrated Business Intelligence Systems

Supervisor:
PhD João Carlos Amaro Ferreira, Assistant Professor with Habilitation,
ISCTE-IUL - Instituto Universitário de Lisboa

Co-supervisor:
PhD Bruno Moisés Teixeira Oliveira, Adjunct Professor,
ESTG-IPP - Escola Superior de Tecnologia e Gestão - Instituto Politécnico do Porto

June, 2023

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

Implementing a Data Integration Infrastructure for Healthcare Data: A Case Study

Miguel Pratas Ferreira Mira

Master's in Integrated Business Intelligence Systems

Supervisor:
PhD João Carlos Amaro Ferreira, Assistant Professor with Habilitation,
ISCTE-IUL - Instituto Universitário de Lisboa

Co-supervisor:
PhD Bruno Moisés Teixeira Oliveira, Adjunct Professor,
ESTG-IPP - Escola Superior de Tecnologia e Gestão - Instituto Politécnico do Porto

June, 2023

ACKNOWLEDGMENTS

I want to express my deepest gratitude to those who have supported and encouraged me along this path. Without your support, reaching this milestone would not be possible.

To my supervisors, Professor João Carlos Ferreira, and Professor Bruno Oliveira, I thank you for all the collaboration, guidance, availability, critical opinions, and knowledge transmitted during the realization of this master's thesis.

To Professor Luís Elvas, I want to express my sincere gratitude for his support and availability while realizing this work, even though he is not officially part of the advisory team. I wish him all the best in his career in the academic teaching field.

To Dr. Luís Rosário and Engª. Diana Mourão, I thank them for being the bridge between ISCTE and the hospital during my research. Your availability and readiness to clarify doubts were essential for developing my work.

To my parents Francisco and Cristina, I express my immense gratitude for all the support and dedication that you have given me throughout my academic journey. I would not be here today, reaching this important milestone without you. Your determination and commitment to strive for my personal and academic success are truly inspiring.

To my grandmother Mimi, I thank you from the bottom of my heart for all the generosity, affection, and unconditional care you have given me.

I express my deepest gratitude to my sisters Rita and Carolina for always being by my side as sisters, friends, and confidants. I cherish every moment we share. Thank you for always being there and making my life more prosperous and meaningful.

To my colleague and girlfriend Rita, I take this opportunity to express my deepest gratitude for being by my side as a colleague and also as my life partner. Your presence fills my days with joy, love, and unconditional support. I thank you for sharing the challenges and achievements of academic and professional life with me and for always encouraging me to achieve my best. I also thank you for understanding my dreams and goals, being by my side in times of difficulty, and celebrating every achievement with me.

To my "brother-in-law" Paulinho, I thank him for being a source of inspiration for academic and professional success. If I chose ISCTE for my studies, much of the responsibility was yours.

To my colleagues and friends, I take a moment to express my deep gratitude for all the support, companionship, and friendship you have provided me over the past years. A special hug to my great friends, Pedro, Francisco, and Sergio.

To my grandfather Tono, who is no longer with us, but I know he would be proud that I did my master's thesis in partnership with the hospital where he worked for over 40 years as a doctor. His legacy and dedication will always inspire me.

RESUMO

Nesta tese de mestrado são abordados os desafios e oportunidades na integração de dados de saúde, com foco em pacientes com histórico de doenças cardiovasculares diagnosticados com COVID-19. O objetivo principal desta pesquisa foi desenvolver uma abordagem abrangente para a integração de dados de saúde de um hospital português num repositório comum compartilhado, no âmbito de um Projeto europeu.

Para o efeito, foram analisadas as características e especificidades dos dados disponibilizados pelo hospital, bem como da base de dados de destino, garantindo a privacidade da informação utilizada. Além disso, foi desenvolvida uma arquitetura ETL para realizar o processo de extração, transformação e carregamento dos dados do hospital para a base de dados de destino.

A metodologia de pesquisa envolveu uma revisão bibliográfica que identificou lacunas nas práticas atuais, incluindo a interoperabilidade entre sistemas de saúde, padronização de formatos e terminologias, questões de privacidade, bem como desafios técnicos e organizacionais.

O sucesso do processo ETL e a completude da base de dados de destino foram avaliados por meio de métricas apropriadas, demonstrando a eficácia desse processo. Este trabalho contribui para o campo de pesquisa da integração de dados de saúde, abordando os desafios e oportunidades na gestão e análise de dados de saúde em pacientes com histórico de doenças cardiovasculares diagnosticados com COVID-19.

A arquitetura ETL desenvolvida e a avaliação do processo ETL confirmam a viabilidade da abordagem proposta. Os resultados destacam a importância da privacidade dos dados, interoperabilidade e técnicas avançadas de análise de dados na área da saúde.

Palavras-chave: Dados de saúde, Integração de dados, ETL, COVID-19, Doenças cardiovasculares

ABSTRACT

In this master thesis, the challenges and opportunities in health data integration are addressed, focusing on data from patients with a history of cardiovascular disease diagnosed with COVID-19. The main objective of this research was to develop a comprehensive approach for integrating health data from a Portuguese hospital into a common shared repository within the scope of a European Project.

For this purpose, the characteristics and specificities of the hospital and target database data were analyzed, ensuring the privacy and security of all the information used. In addition, an ETL architecture was developed to perform the hospital's data extraction, transformation and loading to the target database.

The research methodology involved a literature review that identified gaps in current practices, including interoperability between healthcare systems, standardization of formats and terminologies, privacy issues, and technical and organizational challenges.

The success of the ETL process and the completeness of the target database were evaluated using appropriate metrics, demonstrating the effectiveness of this process. This work contributes to the research field of health data integration by addressing the challenges and opportunities in health data management and analysis in patients with a history of cardiovascular disease diagnosed with COVID-19.

The developed ETL architecture and ETL process evaluation confirm the feasibility of the proposed approach. The results highlight the importance of data privacy, interoperability, and advanced data analysis techniques in healthcare.

Keywords: Healthcare Data, Data Integration, ETL, COVID-19, Cardiovascular Diseases

INDEX

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AI - Artificial Intelligence

EHR - Electronic Health Records

ETL - Extract, Transform, and Loading

HIS - Health Information Systems

PRISMA - Preferred Reporting Items for Systematic Reviews and Meta-Analyses

SSIS - SQL Server Integration Services

WoSCC - Web of Science Core Collection

CHAPTER 1

Introduction

In the ever-evolving healthcare landscape, where data has become the new currency of medical progress, imagine a world where European hospitals collaborate and share vital health information transparently to enable breakthroughs in patient healthcare. What if this unprecedented exchange of knowledge could transcend borders, allowing doctors to access a wealth of collective expertise, accelerating diagnoses, improving treatment outcomes, and ultimately saving countless lives? This compelling vision represents the potential power of harnessing the vast amounts of health data spread across European hospitals and uniting them in a joint mission: revolutionizing healthcare through collaborative data sharing. In this thesis, the complexities of this transformative concept are explored in depth, exploring the opportunities, challenges, and ethical considerations surrounding the creation of a common European shared repository.

1.1 Motivation

Health Data Integration is vital in the modern world, especially during the COVID-19 pandemic. The seamless combination and analysis of health data from multiple sources have become crucial for healthcare systems, researchers, and policymakers, offering numerous benefits that significantly enhance public health efforts and improve patient care [1].

One of the primary reasons for the importance of Health Data Integration is its ability to provide a comprehensive view of an individual's health status. By aggregating data from various sources, healthcare providers can better understand a patient's medical history, current conditions, and treatment outcomes. This comprehensive view enables more accurate diagnoses, personalized treatment plans, and better monitoring of patient progress over time [2], [3].

In COVID-19, Health Data Integration is vital in several critical areas. Firstly, it facilitates effective disease surveillance and monitoring. By integrating data from various sources such as diagnostic test results, hospital admissions, and contact tracing apps, public health authorities can track the spread of the virus, identify hotspots, and make informed decisions regarding resource allocation and targeted interventions [4], [5].

Secondly, Health Data Integration enables rapid research and data-driven decision-making. For example, during the COVID-19 pandemic, researchers worldwide have been tirelessly working to understand the virus, develop treatments, and design vaccination strategies. Researchers can identify patterns, risk factors, and treatment outcomes more efficiently by integrating health data from diverse populations. This knowledge can then be translated into evidence-based guidelines and policies to effectively guide healthcare professionals and policymakers in managing the pandemic [6].

In conclusion, Health Data Integration is paramount in today's healthcare landscape, particularly during COVID-19. It empowers healthcare systems, researchers, and policymakers with a comprehensive understanding of individual and population health, facilitates real-time surveillance and monitoring, supports evidence-based decision-making, enables predictive models and early warning systems, and enhances the delivery of remote healthcare services. Embracing Health Data Integration is essential for optimizing public health efforts and ensuring the well-being of individuals and communities in the face of current and future health challenges [7], [8].

Bearing this scenario in mind, the motivation to develop this thesis is driven by the urgent need to explore and understand the significance of Health Data Integration in the modern healthcare landscape, particularly during the COVID-19 pandemic. Investigating the topic of Health Data Integration aims to uncover its potential benefits and implications, providing valuable insights into how it can significantly enhance public health efforts and improve patient healthcare.

1.2 Problem Overview

The healthcare industry generates about 30% of the existing global data volume, and it is estimated that by 2025 it will reach 36% [9]. The importance of this data can mean the difference between life and death for a patient, and it is, therefore, necessary for healthcare professionals to have access to accurate, consistent, and up-to-date patient data [10], [11]. As a result, healthcare professionals can offer higher quality, more efficient, personalized, and safer healthcare. In addition, they can identify risk factors and speed up the process of diagnosing a disease, increase the quality, efficacy, and safety of treatments, and identify patterns in disease transmission chains. With the availability of this healthcare data, it also becomes possible for the scientific and medical community to conduct health research, thus discovering new drugs and medical treatments [12].

Due to the sensitivity of health-related information, this data is frequently hidden and difficult to access. In addition to privacy and security reasons, the complexity of the domain in question and the several rules, variants, and business metrics result in complex data structures used to store health data. Moreover, hospitals may have multiple departments with their own information systems, resulting in data fragmentation from various sources and data inconsistencies and conflicts. Government restrictions and technical features like these have slowed the adoption of electronic health records (EHR) worldwide. Although it makes sense that health information is restricted, access to this wealth of knowledge, based on legal agreements and trust between the parties, can contribute to saving and improving the quality of life [13].

All over the world, different countries implement different Health Information Systems (HIS) to support managing patient records and monitoring their health [14]. With the massive expansion of technology in healthcare, hundreds of gigabytes of information are stored daily in HIS, forcing this area into new challenges [15]. Although the primary purpose of HIS is to manage patient health, the amount of data storage has driven a paradigm shift, showing that this data can have a secondary use in health research [8].

When discussing health research, the 2019 coronavirus disease pandemic (COVID-19) reaffirms that the most significant challenges and threats to humanity require inter-organizational collaboration, where health data is shared, enabling collaborative decision-making [16]. Furthermore, the pandemic has made it even more evident that large-scale data collection and advances in digital health technologies are beneficial for advancing studies to address major public health concerns. Observational studies show several advantages, including better generalizability across a broader population, faster response time, and reduced costs. In addition, in the context of this pandemic, the ability to conduct these studies would dramatically facilitate surveillance and research into this disease [6]. On the other hand, non-covid patients did not have access to the health care they needed. It is also in these cases where it is most apparent that data sharing and interoperability are critical to health research. International Data Integration is necessary to discover complicated pathways to understand better and prevent diseases, compare causes of disease outcomes in populations, and investigate and compare genetic risk factors [17].

Data integration, enabled by the ETL (Extract, Transform, and Loading) process, represents various models, technologies, and methods for integrating data typically from multiple sources into a single repository. This repository represents a unified data view, providing a single version of the truth. Therefore, this is a costly process and consumes a significant portion of the resources of any data integration project [18]. Furthermore, in healthcare, HIS often have a panoply of disparate data, vocabularies, and models, which creates interoperability problems between different systems.

Thus, these problems need to be addressed by tackling the lack of integrated and comprehensive data and semantic interoperability, facilitating data sharing between various organizations, and allowing data to be used for large-scale research [19], [20].

1.3 Objectives

This research aims to integrate health data from a Portuguese hospital into a common shared repository within the scope of a European project. Throughout this work, the challenges of integrating data from patients with a history of cardiovascular disease, diagnostic information, and the occurrence of cardiovascular complications in patients diagnosed with COVID-19 are addressed. By integrating this information in a standardized manner, a set of research strategies and techniques can be defined and used to identify patterns and promote new treatments and diagnostics. Furthermore, it contributes to studying cardiovascular diseases, one of the world's health priorities [21]. This project has 13815 patients considered, spread across 72 hospitals in 13 countries [22].

Thus, breaking down the primary goal of this thesis into more specific objectives, this work has the following goals:

G1. Analyze the characteristics, specificities, and available information of the data provided by the hospital;

G2. Analyze the target database's characteristics and singularities for storing cardiovascular-related health data;

G3. To develop a robust and efficient ETL architecture that covers data lineage control, incorrect record handling, metadata analysis, and data quality assurance mechanisms for the ETL process of data from the source database to the target database;

G4. Evaluate the ETL process's success and the target database's completeness by validating the populated tables and ensuring they meet the requirements of the European project;

1.4 Outline of the Dissertation

Having defined the motivation, the problem, and the objectives of this thesis, it is now essential to present the remaining chapters. This work consists of 5 chapters (including the Introduction):

Chapter 2: Presents an overview of health data integration processes in the context of cardiovascular diseases and COVID-19. This chapter seeks to gain insight into the advances, challenges, and potential directions in health data integration for research in these diseases through a systematic literature review.

Chapter 3: This chapter aims to show how health data from a Portuguese hospital was integrated into a repository shared by 72 hospitals within a European project. The chapter describes the requirements and the main developments to populate the target database, addressing objectives G1 and G2.

Chapter 4: Presents the implementation of the system, highlighting the software tools used and their importance. It also provides an overview of the ETL flow and demonstrates the system's operation through a specific example. In addition, this chapter presents the evaluation metrics and dashboard used to evaluate the efficiency and effectiveness of the ETL process. Objectives G3 and G4 are covered in this chapter.

Chapter 5: This chapter is devoted to the conclusions of this thesis and future work. It summarizes the main results and contributions achieved throughout the research, highlighting the importance of related work. In addition, the chapter presents suggestions for future work, such as improvements in the data pipeline, application of advanced data analysis techniques, and integration with existing healthcare systems.

CHAPTER 2

State of the art

This chapter examines the current landscape of health data integration processes in the context of cardiovascular disease and COVID-19. It also presents a comprehensive analysis of the existing literature, employing a systematic search strategy and inclusion criteria based on the PRISMA methodology[1]. The search used reputable data repositories, Scopus[2], and the Web of Science Core Collection (WoSCC)[3], resulting in 103 papers. Through a meticulous selection process, 14 articles were identified as relevant to this study. The selected papers, published between 2017 and 2022, cover various topics, including electronic health records and data integration, COVID-19 surveillance and research, health data management and analysis, and data repositories. This chapter aims to gain a comprehensive understanding of the current advancements, challenges, and potential directions in health data integration for cardiovascular disease and COVID-19 research by analyzing these papers.

2.1 Search Strategy and Inclusion Criteria

This section aims to reproduce the strategy used in the literature search related to the topic of this thesis and what criteria were used in the inclusion of documents.

To understand the state of the art, it is intended to answer the question: "What is the state of the art of health data integration processes in cardiovascular disease or COVID-19?". To do this, 2 data repositories were used for the document search: Scopus and the WoSCC. This search was conducted between February and March 2023, followed by the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology.

Then, in the Scopus and WoSCC data repositories, documents were searched, considering the work developed in the scope of the concepts "Healthcare Data" or "Electronic Health Records" with the population "COVID-19" or "Cardiovascular Diseases" and in the context of "ETL" or "Data Integration" or "Data Sharing". In addition, only journal papers, articles, and reviews were considered. Thus, the query used for document search was "("Healthcare Data" OR "Electronic Health Record*") AND ("COVID-19" OR "Cardiovascular Disease*") AND ("ETL" OR "Data Integration" OR "Data Sharing")".

[1] http://www.prisma-statement.org/
[2] https://www.scopus.com/
[3] https://www.webofscience.com/

2.2 Search Results and Document Selection

With the search strategy and inclusion criteria used in section 2.1, 56 documents were obtained from Scopus and 47 from WoSCC, for a total of 103 papers. These documents were then imported to Zotero[4], where 34 duplicate documents were detected between the two data repositories, which were eliminated. Thus, a total of 69 papers were available at this stage.

The title, author, abstract, date, and keywords were selected from these documents, and this information was imported into Microsoft Excel[5]. A first analysis of the papers was then performed, where it was concluded that 42 of these documents did not fit the scope of this study and were, therefore, not considered. The remaining 27 papers were read, of which only 14 made sense to include in the study. The description above was based on the PRISMA methodology, and the respective workflow can be found in Figure 2.1.

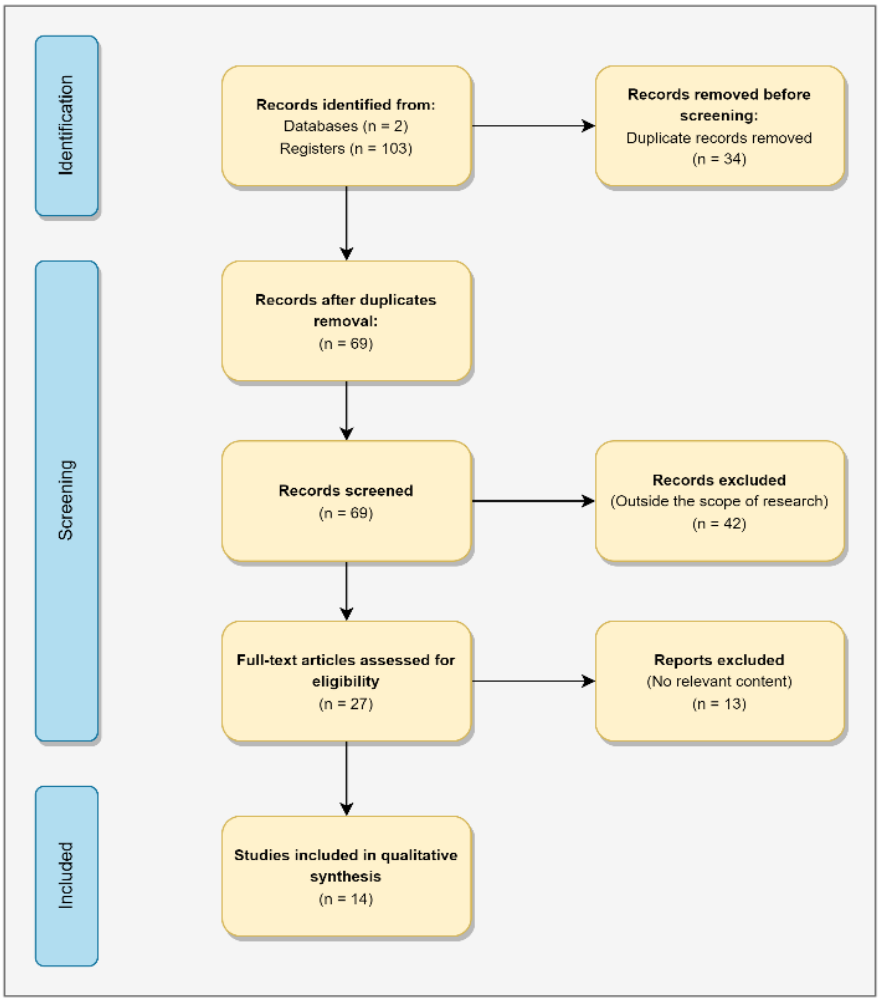

Figure 2.1 - PRISMA Workflow Diagram

[4] https://www.zotero.org/
[5] microsoft.com/pt-pt/microsoft-365/excel

The 14 articles that served as the basis of this literature review were published between 2017 and 2022. By examining Figure 2.2, one can observe a notable increase in the number of articles published related to this theme over this period. This growth highlights the increasing importance and relevance of the study undertaken. The upward trend reflects the recognition of the need to understand and address Health Data Integration processes in the context of cardiovascular disease and COVID-19. The growing interest in this field of research suggests a continuing demand for advances in this area and highlights the importance of the related work in contributing to this development.

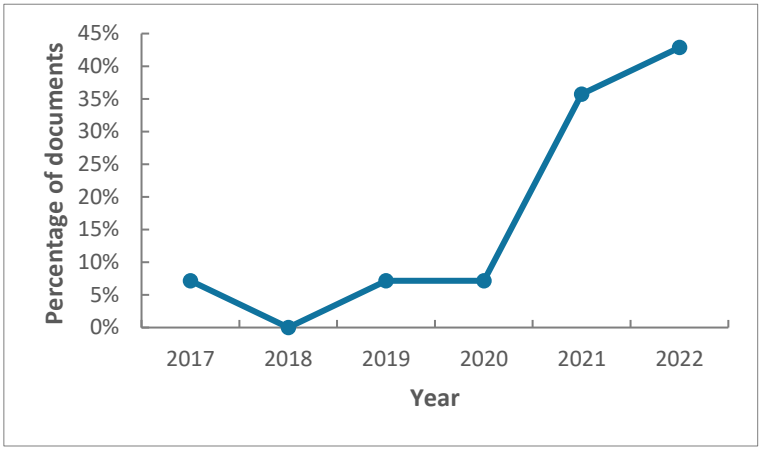

Figure 2.2 - Evolution of the percentage of papers published in the scope of this study, by year

From the analysis of the 14 selected papers, it is concluded that they address 4 main themes. From these themes, the different papers were grouped into 4 distinct groups. Group 1, which focuses on electronic health records and data integration, encompasses 10 selected papers. On the other hand, 7 papers deal more in-depth with the theme of COVID-19 surveillance and research (Group 2). A third central theme was also detected that originated Group 3, whose focus is the management and analysis of health data, and of which 3 papers address it in more detail. Finally, there is 1 paper that addresses the issue of data repositories in more depth (Group 4).

Table 2.1 presents the groups described above and the papers that give them more focus. The fact that, in Table 2.1, a paper is not included in a specific group does not mean that it does not address the respective topic of that group. It just means that the theme addressed in that group is not the central theme addressed in that paper. On the other hand, a paper may be encompassed in more than one group since it may focus on topics presented in different groups.

Table 2.1 - Papers by topic groups

Paper / Group	[1]	[2]	[3]	[4]	[5]	[20]	[21]	[23]	[24]	[25]	[26]	[27]	[28]	[29]
Group 1 - Electronic Health Records and Data Integration	X	X		X		X	X	X	X	X	X	X		
Group 2 - COVID-19 Surveillance and Research	X	X		X	X			X	X		X			
Group 3 - Health Data Management and Analytics			X									X	X	
Group 4 - Data Repositories														X

Once the research strategy and the inclusion criteria have been presented, as well as the search results and the selection of documents for this state of the art, it is now essential to understand what the authors of the selected papers investigated, how they did it, and what conclusions they reached. This analysis is done in section 2.3.

2.3 Related Work

The increasing availability of EHR systems has led to a significant generation of healthcare data, but effectively managing and analyzing this data poses challenges [23]. Optimizing EHR systems is crucial to ensure smooth functioning and usability, as they store comprehensive patient information [24]. Integrating fragmented healthcare data from multiple systems is another challenge, hindering a comprehensive view of patient health information [2], [20]. To address these challenges, various studies propose methodologies and frameworks to enhance EHR systems, facilitate data integration, and leverage analytics for improved patient care, research, and public health initiatives [21], [25].

In the work [21], a method to select patients with cardiovascular diseases from EHR systems is proposed. Problems of lack of data standardization and limitations in information search and retrieval are identified. Proposed solutions include the development of selection algorithms, the use of natural language processing tools, and access to integrated databases.

The study [2] proposes an automated method to create research data repositories from a central healthcare data repository using the *FHIR* (Fast Healthcare Interoperability Resources) standard. The authors recognize problems related to the diversity of healthcare systems and limited access to data. Proposed solutions include using the *FHIR* standard to promote data interoperability, automating the creation of research-specific repositories, and applying this method to COVID-19-related studies. The goal is to facilitate the integration and use of health data for research and analysis.

The study of J. J. Reeves [24] discusses the benefits of integrating student health data into an EHR system. Problems such as fragmentation of student health information and lack of healthcare coordination are identified. Proposed solutions include integrating student data into EHR systems, using communication and collaboration tools, and data privacy and security. The goal is to improve efficiency, continuity of healthcare, and coordination of student healthcare services.

The authors of [20] address EHR integration and harmonization among different institutions. Problems such as the diversity of EHR systems and incomplete data are identified. The authors propose using a multi-view incomplete knowledge graph to represent the data and relate the information across institutions. Algorithms for filling in missing data are proposed to improve the completeness of the integrated data. The importance of data harmonization validation and evaluation is highlighted. These solutions aim to improve Health Data Integration and quality between institutions, allowing more reliable and informed analyses.

In the work of L. M. Fleuren [23], the Dutch Data Warehouse, a comprehensive EHR database for critically ill patients with COVID-19, is presented. The paper's main contribution is to provide a solution for integrating health data from multiple centers, allowing a complete view of the patient's health. The problems identified include data fragmentation and heterogeneity in EHR systems. The Dutch Data Warehouse proposes to solve these problems by creating a centralized database that facilitates data sharing and access. In addition, data standardization and collaboration between health centers are emphasized as essential solutions. These measures aim to improve health data's consistency, comparability, and availability, contributing to the research and care of critically ill patients with COVID-19.

The study [25] presents *TransformEHRs* as a flexible methodology to build transparent ETL processes for EHR reuse. The paper's main contribution is to provide a solution for integrating health data and efficiently reusing EHRs for research and analysis. The problems identified include data heterogeneity and ETL processes' complexity. *TransformEHRs* addresses these problems by providing guidelines and recommended practices for building transparent ETL processes, dealing with data heterogeneity, and facilitating the integration of different sources. Metadata is emphasized to describe the data structure and facilitate proper transformation during the ETL process. In addition, the methodology promotes an iterative and collaborative approach involving clinical domain experts, researchers, and healthcare professionals.

Data integration remains a significant challenge, particularly in the context of COVID-19 surveillance and research [26]. The pandemic has highlighted the need to access and analyze data from various sources to understand the virus's impact and potential interventions comprehensively. COVID-19 surveillance and research require data aggregation from multiple systems, which often have different formats and methodologies, making integration complex. Efficient data integration is crucial for monitoring cases, tracking trends, and supporting evidence-based decision-making [4].

Challenges in data integration include privacy concerns, data quality issues, governance, and interoperability among systems. Ensuring data validation, cleaning, and harmonization is essential to obtain reliable insights [26]. Overall, addressing the challenges of data integration in COVID-19 surveillance and research requires collaborative efforts, privacy considerations, data quality assurance, and the application of appropriate governance frameworks and technological solutions [1], [4], [5] and [26].

The study [4] describes a near-real-time EHR-based COVID-19 surveillance system and addresses the challenges and solutions related to Health Data Integration in developing countries. The paper's main contribution is to provide a system that uses existing data in EHRs to monitor and analyze the spread of COVID-19 in near-real-time. The problems identified include health data's limited availability and quality and the lack of access and interoperability between EHR systems. To overcome these problems, solutions are proposed, such as healthcare infrastructure enhancement, standardization, and interoperability of EHR systems. Healthcare infrastructure improvement involves investments in improving EHR systems and policies that promote data quality and availability. Standardization and interoperability of EHR systems are achieved through the adoption of standards and protocols and clear guidelines for data sharing. The system presented in the article and the proposed solutions aim to facilitate the integration and effective use of health data for more efficient surveillance of COVID-19 in developing countries.

The authors of [26] describe the development of a tool to convert the *PCORnet* (Patient-Centered Outcomes Research Network) data model to the *OMOP* (Observational Medical Outcomes Partnership) data model. The paper's main contribution is to provide a solution to efficiently integrate COVID-19 data from different sources, facilitating data-based analysis and research. The problems identified include the differences between the *PCORnet* and *OMOP* data models, which hinder direct data integration, and the complexity of the ETL processes required to perform the data conversion. To overcome these problems, solutions are proposed, such as developing a specific ETL tool for the conversion, correctly mapping concepts, and terminologies between the data models, and performing validation and verification of the converted data.

These solutions aim to simplify the conversion process, ensure correct data compatibility and integration, and guarantee the accuracy and integrity of the converted data. The article highlights the importance of Health Data Integration and presents a practical approach to overcome the challenges of data model conversion, enabling more efficient analysis of COVID-19 data for research and decision-making.

In the work of J. L. Raisaro [5], the *SCOR* (Secure COVID-19 Research Data Platform), an international secure computing infrastructure developed to research COVID-19, is described. *SCOR* contributes to Health Data Integration by facilitating collaboration and joint data analysis from different sources and countries. Identified problems include data fragmentation related to COVID-19 and security and privacy issues. To overcome these problems, solutions such as developing a secure computing infrastructure, promoting international collaboration and secure data sharing, and implementing appropriate governance and ethics are proposed. These solutions aim to ensure data integrity and confidentiality, promote global cooperation in COVID-19 research, and ensure responsible and ethical use of health data. The article highlights the importance of a secure and collaborative infrastructure for the integration and joint analysis of health data to better understand and respond to the pandemic of COVID-19.

The study [1] presents a comprehensive review of the application of artificial intelligence (AI) and data integration in studies related to COVID-19. This study highlights the importance of these approaches for understanding the pandemic and developing practical solutions. Some problems identified include data diversity and heterogeneity, data quality, and data privacy. To overcome these problems, solutions are proposed, such as using interoperable data standards and structures, artificial intelligence and machine learning algorithms, and ethical data governance. These solutions aim to facilitate the integration and analysis of health data, enabling the identification of valuable patterns, predictions, and insights to address the COVID-19 pandemic. The article highlights the importance of using advanced data analytics approaches and ensuring data ethics and security in research on COVID-19. The role of data integration in enabling comprehensive analysis and deriving insights is emphasized. The paper also discusses current approaches limitations and outlines future research directions. The findings of this review contribute to the understanding of the current state of AI and data integration in COVID-19 research, providing valuable insights for future endeavors in this domain.

Health data management and analytics are crucial in research and healthcare improvement [27]. These processes encompass various activities such as data collection, storage, organization, and retrieval, all aimed at ensuring accurate and accessible health information. EHR are a valuable source of comprehensive patient data, and effective EHR management involves data quality, privacy, and interoperability considerations.

Using statistical and computational techniques, analytics is vital in extracting meaningful insights from health data, enabling evidence-based decision-making and personalized medicine. In addition, it contributes to advancements in patient care, evaluation of healthcare policies, and identification of population health trends. However, challenges persist in data privacy, interoperability, and the availability of skilled professionals [28]. To address these challenges, robust data governance practices, compliance with regulations, adoption of interoperable standards, and investments in data science capabilities are necessary [3].

The authors of [27] propose a health computing architecture to facilitate EHR access and analysis. The article addresses problems such as limited data access, data heterogeneity, and computational complexity. The proposed solutions include health computing infrastructure, data standardization, and analysis and visualization tools. These solutions aim to overcome the challenges of Health Data Integration and support clinical and translational research.

In the work of E. Bacon [28], the creation of a regional distributed data network to improve surveillance of chronic health conditions is addressed. Problems such as data fragmentation and barriers to information sharing are identified. Proposed solutions include developing the network infrastructure, adopting interoperability standards, implementing security and privacy measures, and promoting collaboration among healthcare organizations. These solutions aim to integrate data securely and efficiently to improve the surveillance of chronic health conditions.

In the study [3], *EMR2vec*, a machine learning model, is presented as a solution to integrate health data. Problems such as limited data availability and difficulty interpreting patient data are identified. *EMR2vec* transforms electronic patient data into feature vectors for analysis and decision-making in clinical trials. This model incorporates clinical trial data and machine learning techniques to improve data understanding and utilization in clinical research.

Data repositories are crucial platforms for advancing medical research by storing, sharing, and providing access to research data. These repositories facilitate data sharing among researchers, fostering transparency and reproducibility within the scientific community. Repositories effectively save researchers time and resources by offering pre-existing datasets, allowing them to focus on data analysis and interpretation. Furthermore, repositories enforce data standards and implement quality control measures to ensure the accuracy and consistency of the shared data. Repositories significant advantages are their support for long-term data preservation and accessibility, enabling future analysis and advancements in this field [29].

The study [29] describes the *Acutelines* project as an emergency medicine database which aims to integrate clinical data from patients seen in emergency departments. Although it does not mention specific problems, the project initiative contributes to Health Data Integration by centralizing information, standardizing data, and making it available to researchers. This helps overcome common challenges such as data fragmentation and lack of standardization, facilitating research and advancement in emergency medicine.

Based on the reviewed articles, it is possible to conclude that Health Data Integration is a complex but significant challenge for research, analysis, and patient healthcare. The reviewed studies highlight several problems faced in this context, such as lack of data standardization, information fragmentation, heterogeneity of EHR systems, limited data availability and quality, complexity of ETL processes, and security and privacy issues.

However, the papers also present promising solutions for dealing with these problems. Among the proposed solutions are the development of data selection and completion algorithms, the use of natural language processing tools, the standardization of data using standards such as *FHIR* [2] and *OMOP* [26], the creation of research data repositories, collaboration and communication between systems and institutions, the use of metadata to describe the structure of data, improving healthcare infrastructure, implementing security and privacy measures, promoting interoperability, and establishing appropriate ethical guidelines and governance.

These solutions aim to improve data quality, completeness, availability, and interoperability, facilitating their integration and practical use for research, analysis, and decision-making. In addition, the proposed solutions aim to promote collaboration between researchers and health professionals, enabling a more comprehensive view of patients health.

CHAPTER 3

Design and Architecture

As previously mentioned, the main goal of this work is to integrate health data from a Portuguese hospital into a common shared repository, within the scope of a European project. Therefore, throughout this chapter, the challenges of integrating data from patients with a history of cardiovascular disease, diagnostic information, and the occurrence of cardiovascular complications in patients diagnosed with COVID-19 are addressed.

To meet this challenge, and before any developments, some requirements had to be considered. First, the data provided by the hospital had to be studied to understand its characteristics, specificities, and available information. Then, in the second phase, it was essential to understand the objectives of the target database, namely, its characteristics and singularities, and what data was intended to be obtained. Afterwards, knowing what data was available and what was intended to be achieved, the Data Pipeline to be used for the ETL system had to be defined. This pipeline represents extracting and processing the data from the source database and then loading the processed and unified data into the target database. At the end of this chapter, the main developments that allowed the target database tables to be populated are also presented.

3.1 Analysis and Construction of Source Database

For the development of this work, 138 CSV files were provided by the hospital. These documents were imported into a SQL Server database, resulting in 138 tables. For security and patient privacy reasons, this database was kept on a private SQL server with limited access, where all sensitive data was anonymized. Furthermore, it is also important to mention that this work was approved by the Ethics Committee and the Data Protection Officer of the hospital under study.

Once the legal and ethical issues were assured, an analysis was performed on the hospital's data, stored in a database created for this purpose (Source Database). This study concluded that this database had a panoply of tables with distinct scopes and specificities. Thus, the tables were first grouped into several sets. According to the relations between tables, their particularities, and their domain, 16 sets of tables were defined, as illustrated in Figure 3.1. For each group of tables, it is essential to briefly describe which tables were used in the development of this work.

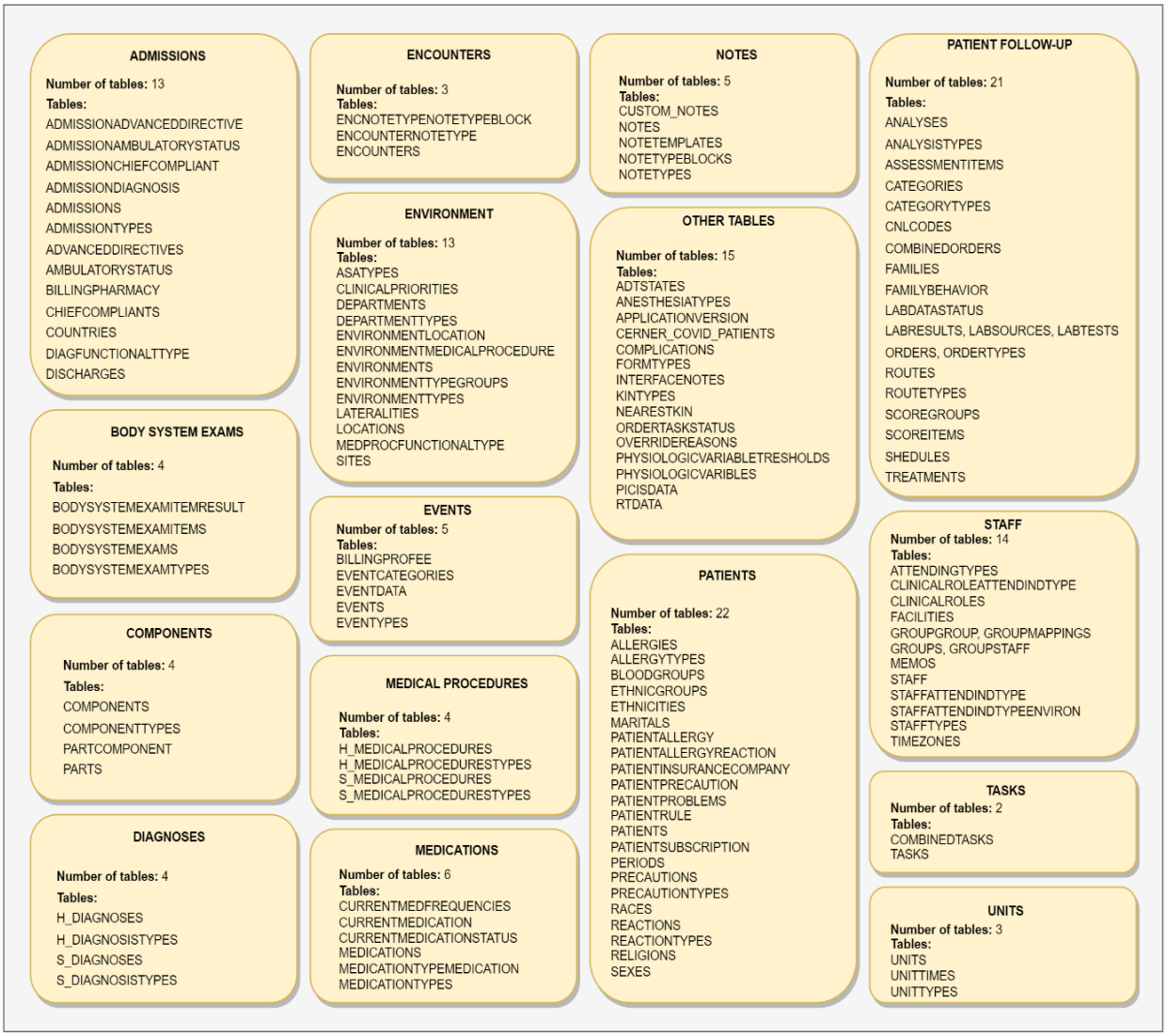

Figure 3.1 - Sets of source database tables

The first set of tables to be analyzed was the *ADMISSIONS*. This group represents 13 tables detailing the patient's hospital admission. From this collection of tables, the *ADMISSIONS* table was used, where the information regarding the patient's admission identifier, the dates of the patient's admission to and departure from the hospital and the PICIS system (the system where COVID-19 patient admissions were recorded), the patient's height and weight on the date of entry to the hospital, and the foreign keys for the *PATIENTS* and *COUNTRIES* tables are stored. Another table used from the *ADMISSIONS* set was the *COUNTRIES* table, which holds the country identifier and the names of the countries where patients may reside.

Tables 3.1 and 3.2 present the columns used from the *ADMISSIONS* and *COUNTRIES* tables, respectively, as well as their descriptions, data types, and example values.

Table 3.1 - Information used from the *ADMISSIONS* table

ADMISSIONS TABLE			
Column name	**Description**	**Data type**	**Example value**
ADMISSIONDBOID	Admission ID.	numeric (21, 0)	4829474605392769552
HOSPITALSTARTED	Date of patient admission to the hospital.	datetime	2021-09-21 10:35:25.000
STARTED	Date of patient admission to the PICIS system.	datetime	2021-09-21 12:13:45.083
ENDED	Date of patient exit in the PICIS system.	datetime	2021-09-21 12:54:58.700
WEIGHT	Patient weight (kilograms).	numeric (12, 5)	80.00000
HEIGHT	Patient height (meters).	numeric (12, 5)	NULL
PATIENTDBOID	FK to PATIENTS.	numeric (21, 0)	1828774773746769552
COUNTRYDBOID	FK to COUNTRIES.	numeric (21, 0)	43000000000000000000
HOSPITALENDED	Date of patient exit to the hospital.	datetime	NULL

Table 3.2 - Information used from the *COUNTRIES* table

COUNTRIES TABLE			
Column name	**Description**	**Data type**	**Example value**
COUNTRYDBOID	Country ID.	numeric (21, 0)	43000000000003000552
COUNTRYDESC	Country name.	varchar (128)	Angola

Then, for the *COMPONENTS* group, which comprises 4 tables, the *COMPONENTS* table was used, containing the individual items tested as part of a laboratory test (e.g., Troponin) and the *PARTS* table, including the information of the type of laboratory analysis performed (e.g., "SARS-CoV-2 IgG IgM serology"). Tables 3.3 and 3.4 show the data used from the *COMPONENTS* and *PARTS* tables, respectively. The table *PARTCOMPONENT* was also used, but only to perform joins between tables through the identifier *PARTCOMPONENTDBOID*.

Table 3.3 - Information used from the *COMPONENTS* table

COMPONENTS TABLE			
Column name	**Description**	**Data type**	**Example value**
COMPONENTDBOID	Component ID.	numeric (21, 0)	26482683476790119552
COMPONENTDESC	Component description.	varchar (128)	CK-MB massa .....

Table 3.4 - Information used from the *PARTS* table

PARTS TABLE			
Column name	**Description**	**Data type**	**Example value**
PARTDBOID	Parts ID.	numeric (21, 0)	53784553660586769552
PARTDESC	Part description.	varchar (128)	Serologia SARS-CoV-2 IgG IgM

From the third group of tables to be analyzed, *ENCOUNTERS*, which has 3 tables, only the *ENCOUNTERS* table was used to perform joins between tables through the *ENCOUNTERDBOID* attribute. The same happened for the *ENVIRONMENT* collection, which represents 13 tables in the database, of which the tables *DEPARTMENTS*, *ENVIRONMENTLOCATION*, *ENVIRONMENT* and *LOCATIONS* were used, where each table's identifiers were utilized to cross-reference information between different tables.

The *OTHER TABLES* set includes 15 tables, of which the *RTDATA* table was used, containing the data related to physiological variables (e.g., Temperature). In addition, the table *PICISDATA* was also used to perform joins between tables using the identifier *PICISDATADBOID*. Table 3.5 shows the columns used in the *RTDATA* table, their descriptions, data types, and example values.

Table 3.5 - Information used from the *RTDATA* table

RTDATA TABLE			
Column name	**Description**	**Data type**	**Example value**
RTDATADBOID	RTDATA ID.	numeric (21, 0)	63813240000890933552
STARTED	Datetime the physiologic variable was read.	datetime	2021-03-19 12:00:00.000
PICISDATADBOID	FK to PICISDATA.	numeric (21, 0)	55813029751980769552
CD06	Temperature (Manual Registration in °C).	numeric (8, 3)	36.800
C057	Rectal Temperature (°C).	numeric (8, 3)	NULL
C050	Temperature (°C).	numeric (8, 3)	NULL
C980	Systolic Pressure (Art.) 2.	numeric (8, 3)	NULL
C013	Systolic Blood Pressure.	numeric (8, 3)	NULL
C010	Systolic Blood Pressure – Invasive.	numeric (8, 3)	139.000
C981	Diastolic Pressure (Art.) 2.	numeric (8, 3)	NULL
C014	Diastolic Blood Pressure.	numeric (8, 3)	71.000
C011	Diastolic Blood Pressure – Invasive.	numeric (8, 3)	NULL
C100	Respiratory Rate.	numeric (8, 3)	22.000
C080	Respiratory Rate.	numeric (8, 3)	NULL
C120	SpO2 (%).	numeric (8, 3)	97.000
C001	Heart Rate.	numeric (8, 3)	144.000

As for the *PATIENTS* group, which represents 22 tables, the tables *PATIENTS*, *SEXES*, *ETHNICGROUPS* and *ETHNICITIES* were used to obtain the patient's demographic information. Tables 3.6, 3.7, 3.8 and 3.9 show the data used by each of the above tables.

Table 3.6 - Information used from the *PATIENTS* table

PATIENTS TABLE			
Column name	**Description**	**Data type**	**Example value**
PATIENTDBOID	Patient ID.	numeric (21, 0)	1455818944631078552
BIRTHDATE	Patient birth date.	datetime	1999-11-15 00:00:00.000
ETHNICGROUPDBOID	FK to ETHNICGROUPS.	numeric (21, 0)	10000000000000000000
SEXDBOID	FK to SEXES.	numeric (21, 0)	11000000000000000000
ETHNICITYDBOID	FK to ETHNICITIES.	numeric (21, 0)	432737894980934768552

Table 3.7 - Information used from the *SEXES* table

SEXES TABLE			
Column name	**Description**	**Data type**	**Example value**
SEXDBOID	Sex ID.	numeric (21, 0)	11000000000001000000
SEXDESC	Sex description.	varchar (128)	Feminino

Table 3.8 - Information used from the *ETHNICGROUPS* table

ETHNICGROUPS TABLE			
Column name	**Description**	**Data type**	**Example value**
ETHNICGROUPDBOID	Ethnic group ID.	numeric (21, 0)	10000000000002000552
ETHNICGROUPDESC	Ethnic group description.	varchar (128)	Asiático

Table 3.9 - Information used from the *ETHNICITIES* table

ETHNICITIES TABLE			
Column name	**Description**	**Data type**	**Example value**
ETHNICITYDBOID	Ethnicity ID.	numeric (21, 0)	432737894933511768552
ETHNICITYDESC	Ethnicity description.	varchar (128)	Africano

The *PATIENT FOLLOW-UP* set comprises 21 tables, from which the *LABRESULTS* table was used, where the information regarding the laboratory test results is stored, and the *CNLCODES* table, where the descriptions and codes of the physiological variables are represented. In addition, the table *LABTESTS* was also necessary to perform joins between tables through the identifier *LABTESTDBOID*. Tables 3.10 and 3.11 represent the columns used from the *LABRESULTS* and *CNLCODES* tables and their descriptions, data types and example values.

Table 3.10 - Information used from the *LABRESULTS* table

LABRESULTS TABLE			
Column name	**Description**	**Data type**	**Example value**
DBOID	Lab result ID.	numeric (21, 0)	50492515652601119552
TEXTVALUE	Laboratory text value.	varchar (1024)	30
OBSERVATIONTIME	Observation datetime.	datetime	2016-01-02 12:02:00.000

Table 3.11 - Information used from the *CNLCODES* table

CNLCODES TABLE			
Column name	**Description**	**Data type**	**Example value**
CNLCODEDBOID	Cnl code ID.	numeric (21, 0)	317000000000002697000
CODE	CNL (Click'n Link) code.	varchar (6)	000A89
SHORTDESCRIPTION	Short description of the CNL code.	varchar (50)	IT2
LONGDESCRIPTION	Long description of the CNL code.	varchar (255)	Inspiratory time dup

From the last group analyzed, *UNITS*, only the *UNITS* table was used, where the information regarding the units of measurement used in the hospital is present (e.g., Grams), and the relevant data used from that table can be found in Table 3.12.

Table 3.12 - Information used from the *UNITS* table

UNITS TABLE			
Column name	**Description**	**Data type**	**Example value**
UNITDBOID	Unit ID.	numeric (21, 0)	45369590162507097552
UNITSYMBOL	Abbreviation used for the unit of measure.	varchar (16)	g/L

It should be noted that, of the sets presented above, for the tables where the only column used was the table identifier (for example, the *ENCOUNTERS* table with the identifier *ENCOUNTERDBOID*), a tabular format was not presented (as, for example, in Table 3.11) because it was not considered relevant. In any case, it is noted that all table identifiers unambiguously represent a record of the respective table and are of type *numeric (21,0)*. Furthermore, it is also important to mention that of the sets *BODY SYSTEM EXAMS*, *DIAGNOSES*, *EVENTS*, *MEDICAL PROCEDURES*, *MEDICATIONS*, *NOTES*, *STAFF*, and *TASKS*, no tables were used for the development of this work.

Thus, it was concluded that data from 20 tables from the source database were used to develop this work.

3.2 Analysis and Construction of Target Database

The previous section presented the characteristics, specificities, and information available in the source database, built from health data from the hospital under study. In addition, all tables containing relevant information for the development of this work were analyzed in more detail. Once the available data were understood, it was essential to understand the target database's characteristics and singularities.

As discussed earlier, the target database stores the data of patients with a history of cardiovascular disease, diagnostic information, and the occurrence of cardiovascular complications in patients diagnosed with COVID-19. To this end, this project aims to store this information in 14 tables with distinct purposes, as illustrated in Figure 3.2. By analyzing this figure, it is possible to see all the tables of the target database that are intended to be populated, as well as the number of columns in each of the tables, whether the table is mandatory or optional, and a brief description of the contents of each table. There are a total of 812 columns distributed over the 14 tables, 9 of which require completion (Required = "Yes"), and the remaining 5 require optional completion (Required = "No").

The structure of the target database was created based on documentation provided by those responsible for the European project, where information was available about all the tables to be populated, namely, the name of the tables and columns, their type of data and descriptions, and the possible values for each column. It should be noted that, as in the source database, the target database was kept in a private SQL server with limited access, thus respecting patient data privacy.

This work focused on the first 7 tables represented in Figure 3.2: *Participant Identification Number (PIN)*, *Inclusion Criteria*, *Demographics*, *Cardiac Baseline Assessment*, *Cardiac Biomarkers*, *ECG and Echocardiography*. For the study to be considered valid for the European project, the first 5 tables presented above must be filled in, while the remaining two are optional. Only the first 7 tables shown in Figure 3.2 are the focus of this work because the responsibility of filling in the remaining tables has been delegated to another team in the hospital. For that reason, the tables *Cardiac MRI*, *CT*, *Invasive Cardiac Procedures*, *Cardiac and Thromboembolic COVID-19 Complications*, *Cardiac Outcome: 7-day follow-up*, *Cardiac Outcome: 30-day follow-up* and *Discharge* will not be covered in this work. Throughout this section, the constitutions of each table to be filled are presented, focusing on the columns present, their descriptions, types of data and the possible values of that columns.

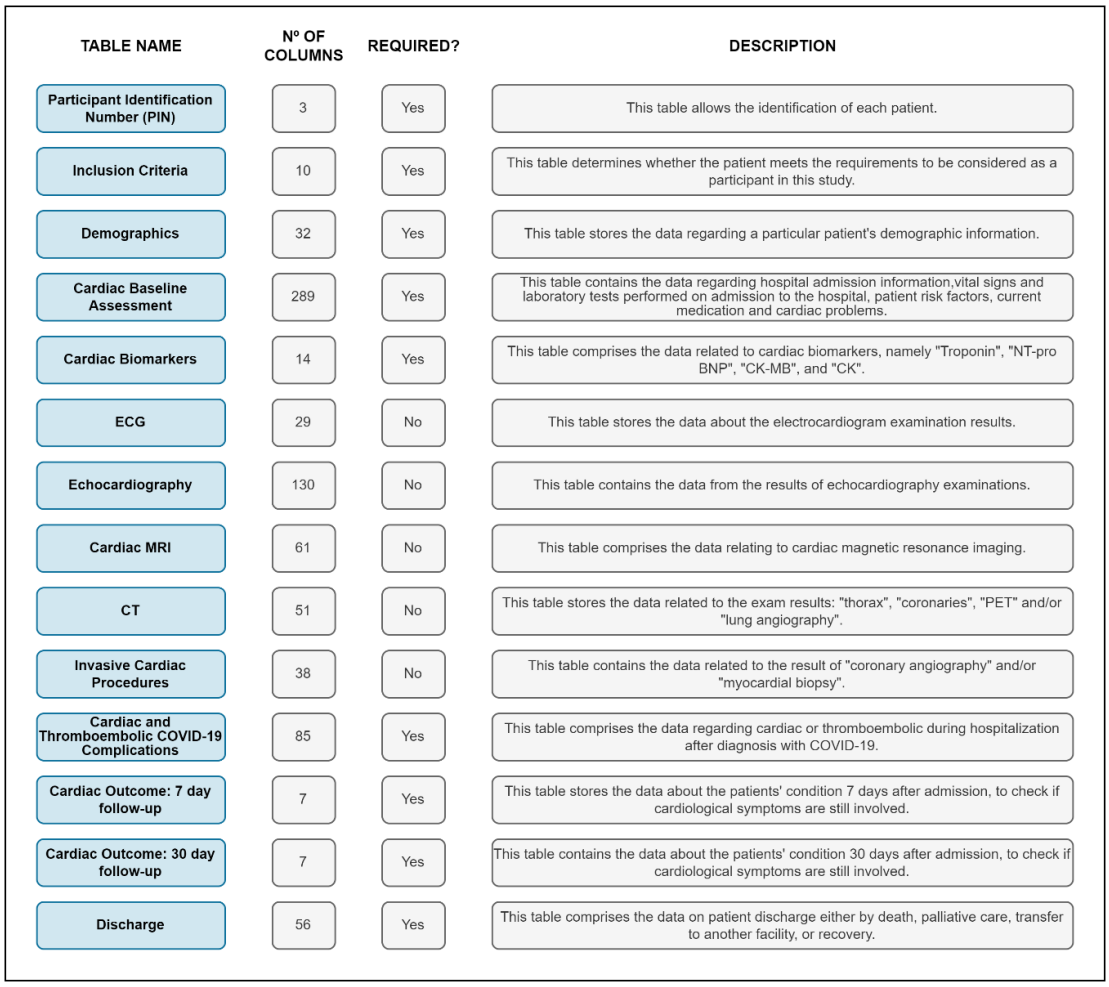

Figure 3.2 - Target database description

The first table to be analyzed was the *Participant Identification Number (PIN)*. This table is intended to store the identification data of each patient and contains the *subjid*, *date_created* and *pin_complete* columns, as seen in Table 3.13. The *subjid* column concerns the identifier of the participant's admission to this study, having the format "<*Hospital Identification Number*>-<*Incremental number*>". Those responsible for the European project provided the Hospital Identification Number with a fixed value 918. Thus, for example, the first patient's admission has the *subjid* equal to 918-1, and the second admission has 918-2. The *date_created* column indicates the date each record was created in the target database. Finally, the *pin_complete* column allows verifying if, for each record in this table, all columns are filled or not (2 - Complete or 0 - Incomplete, respectively) or if this information was not verified (1 - Unverified).

Note also that it is from the *subjid* column that the link between the *Participant Identification Number (PIN)* table and the remaining tables is possible. In other words, the other tables of the target database always have a column that references the *subjid* column, indicating, for each record, which is the patient's admission identifier.

Table 3.13 - *Participant Identification Number (PIN) table constitution*

Participant Identification Number (PIN)			
Column name	**Description**	**Data type**	**Possible values**
subjid	Participant Identification Number (PIN).	nvarchar (25)	918-100, 918-205, …
date_created	Date the record was created in the CAPACITY database.	date	2023-10-04, …
pin_complete	Complete table?	int	0 – Incomplete, 1 – Unverified, 2 - Complete

Subsequently, the *Inclusion Criteria* table was analyzed, which allows for checking whether a given patient meets the requirements to be considered in the study. This table contains 10 columns, as represented in Table 3.14, and the descriptions of each of the columns are shown below:

- *dsstdat*, which indicates the date when the infection caused by COVID-19 was first evaluated as suspected or confirmed by a physician;
- *sitename_nhr*, which identifies the hospital where the patient was admitted as a result of the infection caused by COVID-19;
- *country* and *othcountry*, which specify the patient's country of residence;
- *corona_ieorres* and *symptoms_epi_physical*, which indicate, respectively, whether for each patient there was (1 - Yes) or not (0 - No) a suspected or confirmed coronavirus infection, and whether in the 14 days before the onset of illness, the patient had (1 - Yes) or not (0 - No) any close contact[6], or whether this information is unknown (3 - Unknown);
- *symptoms_epi_healthfac* and *symptoms_epi_lab*, which indicate, respectively, if the patient was (1 - Yes) or not (0 - No) present in any healthcare facility where COVID-19 cases were being managed or if he/she was (1 - Yes) or not (0 - No) present in a laboratory handling suspected or confirmed coronavirus samples, or if this information is unknown (3 - Unknown);
- *inclusion_complete*, which allows verifying if, for each record in this table, all columns are filled in or not (2 - Complete or 0 - Incomplete, respectively) or if such information was not verified (1 - Unverified);
- *subjid*, that for each record in this table, contains the identifier for the patient's admission;

[6] Definition of close contact used in the study: Health care associated exposure, including providing direct care for coronavirus patients; Working together in nearby or sharing the same classroom environment with a coronavirus patient; Travelling together with coronavirus patient in any kind of conveyance; Living in the same household as a coronavirus patient.

Table 3.14 - *Inclusion Criteria* table constitution

Inclusion Criteria			
Column name	**Description**	**Data type**	**Possible values**
dsstdat	Enrolment date: date of data entry.	date	2021-04-24, …
sitename_nhr	Site name.	nvarchar (4)	16 - Medisch Centrum Leeuwarden, 12 - Catharina Ziekenhuis, …
country	Country.	int	63 – Georgia, 64 - Germany, …
oth_country	Other country.	nvarchar (50)	Unknown, …
corona_ieorres	Suspected or confirmed novel coronavirus (COVID-19) infection?	tinyint	1 – Yes, 0 – No
symptoms_epi_physical	In the 14 days before onset of illness had any of the following: Close contact with a confirmed or probable case of nCoV infection, while that patient was symptomatic.	int	1 – Yes, 2 – No, 3 - Unknown
symptoms_epi_healthfac	Presence in a healthcare facility where nCoV infections have been managed.	int	1 – Yes, 2 – No, 3 - Unknown
symptoms_epi_lab	Presence in a laboratory handling suspected or confirmed nCoV samples.	int	1 – Yes, 2 – No, 3 - Unknown
inclusion_complete	Complete table?	int	0 – Incomplete, 1 -Unverified, 2 - Complete
subjid	Participant Identification Number (PIN).	nvarchar (25)	918-100, 918-205, …

Then, the *Demographics* table was analyzed, which has 32 columns and contains information regarding each patient's demographic data. Table 3.15 shows the composition of the table mentioned above, and it is essential to describe its columns, presented below:

- *sex*, which indicates the patient's birth sex (1 - Male or 2 - Female) or if this information was not specified or is unknown (-1 Not specified/Unknown);
- *age_estimateyears* and *age_estimateyearsu*, which respectively determine the age of the patient at the time of admission to the hospital and the unit of measurement used (1 - Months, 2 - Years);
- 11 columns for the patient's ethnicity (ethnic__1 - ethnic__10) and *other_ethnic* to specify an ethnic group that is not represented in the picklist;
- *healthwork_erterm* and *labwork_erterm* to assess whether the patient worked (1 - Yes) or not (2 - No) as a health worker and/or in a microbiology laboratory, respectively, or whether this information is unknown (3 - Unknown);
- *pregyn_rptestcd* and *egestage_rptestcd*, which indicate, respectively, whether the patient was (1 - Yes) or not (0 - No) pregnant at the time of admission to the hospital, or if this information is unknown (998 - Unknown) and, in case the patient was pregnant, the number of gestational weeks;

- *postpart_rptestcd*, *pregout_rptestcd*, *dlvrdtc_rptestcd*, *aplb_lbperf* and *aplb_lborres*, whose aim is to check, respectively, if the patient was (1 - Yes) or not (0 - No) in a postpartum situation, what the pregnancy outcome was (1 - Live birth or 2 - Still birth), the delivery date, whether the baby was tested (1 - Yes) or not (2 - No) for COVID-19 infection and if so, whether or not the test result was positive (1 - Positive or 2 - Negative, respectively);
- For patients whose age was less than 1 year, there are 7 variables to fill in:
    - *apdm_age*, which indicates whether the patient was (1 - Yes) or not (0 - No) less than 1 year old at the time of admission to the hospital;
    - *apvs_weight* and *apvs_weightu*, which store respectively the patient's birth weight and the unit of measurement used (1 - kg or 2 - lbs);
    - *apsc_gestout*, which contains the gestational outcome (1 - Term birth or 2 - Preterm birth);
    - *apsc_brfedind* and *apsc_brfedindy*, which check, respectively, if the patient has breastfed (1 - Yes) or not (0 - No) from birth to the date of admission to the hospital and in case the patient has ever breastfed, indicates if at the time of entry to the hospital, the patient was still breastfeeding (1 - Currently breastfed) or if breastfeeding had been discontinued (2 - Breastfeeding discontinued);
    - *apsc_vcageind*, which checks whether or not the patient's vaccinations are age/country appropriate (1 - Yes or 2 - No, respectively) or if this information is unknown (3 - Unknown);
- *demographics_complete*, which allows checking whether or not, for each record in this table, all columns are filled out (2 - Complete or 0 - Incomplete, respectively), or if this information is not verified (1 - Unverified);
- *subjid*, that for each record in this table contains the identifier for the patient's admission;

Table 3.15 - *Demographics* table constitution

Demographics			
Column name	**Description**	**Data type**	**Possible values**
sex	Sex at Birth.	int	1 - Male, 2 - Female, -1 - Not specified/Unknown
age_estimateyears	Age (If patient is a child less than one year age, include age in months).	int	10, 80, …
age_estimateyearsu	Age unit.	int	1 - Months, 2 - Years
etnic__1 etnic__2 etnic__3 etnic__4 etnic__5 etnic__6 etnic__7 etnic__8 etnic__9 etnic__10	Ethnic group: 1 ethnic___1 Arab 2 ethnic___2 Black 3 ethnic___3 East Asian 4 ethnic___4 South Asian 5 ethnic___5 West Asian 6 ethnic___6 Latin American 7 ethnic___7 White 8 ethnic___8 Aboriginal/First Nations 9 ethnic___9 Other 10 ethnic___10 Unknown	tinyint	0 - Unselected ethnicity, 1 - Selected ethnicity
other_ethnic	Other ethic.	nvarchar (50)	Asian, …
healthwork_erterm	Employed as a healthcare worker?	int	1 – Yes, 2 – No, 3 - Unknown
labwork_erterm	Employed in a microbiology laboratory?	int	1 – Yes, 2 – No, 3 - Unknown
pregyn_rptestcd	Pregnant?	int	1 – Yes, 0 – No, 998 - Unknown
egestage_rptestcd	If pregnant: Gestational weeks assessment.	nvarchar (50)	20, …
postpart_rptestcd	post-Partum.	tinyint	1 – Yes, 0 – No
pregout_rptestcd	Pregnancy Outcome.	int	1 - Live birth, 2 - Still birth
dlvrdtc_rptestcd	Delivery date.	date	2021-07-21, …
aplb_lbperf	Baby tested for COVID-19/SARS-CoV-2 infection?	int	1 - Yes, 2 - No
aplb_lborres	If baby was tested for COVID-19/SARS-CoV-2 infection: result of test	int	1 - Positive, 2 - Negative
apdm_age	INFANT - Less than 1 year old?	tinyint	1 – Yes, 0 – No
apvs_weight	Birth weight.	nvarchar (10)	3.45, …
apvs_weightu	Birth weight unit.	int	1 - kg, 2 - lbs
apsc_gestout	Gestational outcome.	int	1 - Term birth (>= 37wk GA), 2 - Preterm birth (< 37wk GA)
apsc_brfedind	Breastfed.	int	1 - Yes, 2 - No
apsc_brfedindy	Breastfeeding status.	int	1 - Currently breastfed, 2 - Breastfeeding discontinued
apsc_vcageind	Vaccinations appropriate for age/country?	int	1 – Yes, 2 – No, 3 - Unknown
demographics_complete	Complete table?	Int	0 – Incomplete, 1 -Unverified, 2 - Complete
subjid	Participant Identification Number (PIN).	nvarchar (25)	918-100, 918-205, …

The fourth table to be studied, *Cardiac Baseline Assessment*, contains a total of 289 columns where data regarding the patient's hospital admission information, the vital signs and laboratory tests performed by the patient on admission to the hospital, risk factors, current medication, and the patient's cardiac problems are stored. Given this table's large number of columns, a different tabular representation is used from those presented previously, as shown in Table 3.16.

To be able to summarize the information contained in the *Cardiac Baseline Assessment* table, 7 sets of columns were created and are presented below.

The first group of columns created, *Hospital Admission Information*, represents 22 columns from the *Cardiac Baseline Assessment* table and contains data regarding the information collected on the patient's admission to the hospital. For example, this group includes information about the date of the appearance of any symptoms suspected of COVID-19, whether or not the patient was transferred from another healthcare facility and the patient's health complaints on admission to the hospital.

Next, the *Vital Signs* group was defined, which encompasses 7 columns and stores information regarding the patient's vital signs on admission to the hospital. This set stores information such as the patient's body temperature (in degrees Celsius), heartbeats per minute, and oxygen saturation (in percent).

The third set of columns, defined as *Laboratory Tests*, contains 22 columns with information regarding the laboratory tests performed on the patient's hospital admission. For each laboratory test, data about the availability of that test, the test result and the unit of measurement used are stored. For example, in the case of the Hemoglobin laboratory test, there are the columns *admission_hb_available*, which indicates whether or not the Hemoglobin values were available until 24 after the patient's admission to the hospital, *admission_hb*, which stores the value of the test relative to the patient's Hemoglobin and *admission_hb_unit*, which comprises the possible measurement units for the Hemoglobin laboratory test result (1 - g/L, 2 - g/dL or 3 - mmol/l).

Then, there is the *Risk Factors* set, which contains 42 columns that store the patient's risk factors information. Information about whether the patient has diabetes and its type, is hypertensive, or is a smoker are some examples of data recorded in this set of columns.

Next, the *Current Medication* group was defined, which presents the columns related to the medications that the patient was taking at the time of admission to the hospital. It is in this group of columns that information, for example, about whether the patient was taking any betablocker (such as Atenolol or Carvedilol), Antiarrhythmic drug (for example, Class I or Class III) or diuretic (such as Bumetanide or Chloortalidon), is stored.

The sixth set of columns defined, Cardiac Problems, contains the columns related to the patient's cardiac issues. For example, the date the patient had a conduction disturbance, heart failure and/or any coronary artery disease.

Finally, the *Additional Information Group*, which as in the other tables above, contains 2 columns. The *cardiac_baseline_assessment_complete* column, which allows verifying if, for each record, all columns are filled or not (2 - Complete or 0 - Incomplete, respectively), or if such information was not verified (1 - Unverified), and the *subjid* column, which, for each record in this table, identifies the associated patient admission.

Table 3.16 - *Cardiac Baseline Assessment* table constitution, by column groups

Cardiac Baseline Assessment		
Column group name	**Nº of columns**	**Description**
Hospital Admission Information	22	This group of tables contains data regarding the information collected on admission of the patient to the hospital. For example: complaints on admission, date of appearance of any suspicious symptoms of COVID-19 and whether the patient was transferred from another hospital.
Vital Signs	7	This collection of tables stores the information regarding the patient's vital signs on admission to the hospital. For example: Temperature, number of heart beats per minute, and oxygen saturation.
Laboratory Tests	22	In this grouping of tables are all the columns that store the information regarding the laboratory tests performed on admission of the patient to the hospital. For example: Hemoglobin, platelet count, and d-dimer.
Risk Factors	42	This set of tables presents the columns that store information regarding the patient's risk factors. For example: Hypertension, Diabetes and smoking.
Current Medication	105	This group of tables presents columns relating to the medications the patient was taking at the time of admission to the hospital. For example: types of betablockers, types of diuretics and types of Antiarrhytmic drugs.
Cardiac Problems	89	This grouping of tables contains the columns that concern the patient's cardiac problems. For example: supraventricular tachycardias Atrial fibrillation, ventricular arrhythmias non-sustained ventricular tachycardia and conduction disorders First degree AV block.
Additional Information	2	This set contains two tables: *capacity_cardiac_baseline_assessment_complete* and *subjid*.

Then, the *Cardiac Biomarkers* table was analyzed, which comprises 14 columns that store the levels of cardiac biomarkers in the blood. It is a requirement of the study to measure these levels at 2 timepoints:

- Up to 24 hours after the patient's admission to the hospital;
- Highest measured value during admission;

Therefore, 2 instances of the *Cardiac Biomarkers* table were created: *Cardiac_biomarkers_24* and *Cardiac_biomarkers_HIGHER*, to store the measurements at the abovementioned timepoints.

Thus, the columns present in the 2 instances of the *Cardiac Biomarkers* table encompass, as represented in Table 3.17, the following columns:

- *date_biomarker*, which stores the date when the cardiac biomarker level in the blood were measured;
- *bio_trop*, *bio_trop_unit* and *bio_trop_value*, which store respectively the type of Troponin that was measured (0 - Not available, 1 - High sensitive Troponin I (hs-cTnI), 2 - High sensitive Troponin T (hs-cTnT), 3 - Troponin I or 4 - Troponin T), the unit of measurement used (1 - ng/L or 2 - µg/l) and the value of the Troponin level result;
- *bio_bnp*, *bio_bnp_unit* and *bio_bnp_value*, which respectively indicate the type of BNP (brain natriuretic peptide) that was measured (0 - Not available, 1 - NT-proBNP or 2 - BNP), the unit of measurement used (1 - pg/mL or 2 - pmol/L) and the value of the BNP level result;
- *bio_ckmb*, *bio_ckmb_unit* and *bio_ckmb_value*, which respectively store whether or not CK-MB (creatine kinase-myocardial band) has been measured (1 - Available or 0 - Not available), the unit of measurement used (1 - U/L, 2 - µg/L) and the result value of the CK-MB level;

- *bio_ck* and *bio_ck_value*, which respectively indicate whether or not the CK (creatine kinase) level was measured (1 - Available or 0 - Not available) and the CK level result value;
- *cardiac_biomarkers_complete*, which allows checking whether or not, for each record in this table, all columns are completed (2 - Complete or 0 - Incomplete, respectively) or if such information was not verified (1 - Unverified);
- *subjid*, that for each record in this table contains the identifier for the patient's admission;

Table 3.17 - *Cardiac Biomarkers* table constitution

Cardiac Biomarkers			
Column name	**Description**	**Data type**	**Possible values**
date_biomarker	Date of examination.	date	2021-05-05, …
bio_trop	Troponin.	int	0 - Not available, 1 - High sensitive Troponin I (hs-cTnI), 2 - High sensitive Troponin T (hs-cTnT), 3 - Troponin I, 4 - Troponin T
bio_trop_unit	Troponin: Unit.	int	1 - ng/L, 2 - µg/l
bio_trop_value	Troponin: value.	nvarchar (10)	22, …
bio_bnp	(NT-pro) BNP.	int	0 - Not available, 1 - NT-proBNP, 2 - BNP
bio_bnp_unit	(NT-pro) BNP: Units.	int	1 - pg/mL, 2 - pmol/L
bio_bnp_value	(NT-pro) BNP: value.	nvarchar (10)	763, …
bio_ckmb	CK-MB.	int	0 - Not available, 1 - Available
bio_ckmb_unit	CK-MB: Units.	int	1 - U/L, 2 - µg/L
bio_ckmb_value	CK-MB: Value.	nvarchar (10)	NULL, …
bio_ck	CK.	Int	0 - Not available, 1 - Available
bio_ck_value	CK: Value U/L.	nvarchar (10)	NULL, …
cardiac_biomarkers_complete	Complete table?	Int	0 – Incomplete, 1 -Unverified, 2 - Complete
subjid	Participant Identification Number (PIN).	nvarchar (25)	918-100, 918-205, …

The constitution of the *ECG* and *Echocardiography* tables, which store data related to Electrocardiograms and Echocardiography performed by patients, will not be presented. This is because, at the time of the preparation of this work, the data that would allow these 2 tables to be populated were not available on the hospital side, nor was there any forecast of when they would be available. Thus, since these are two optional tables for the study, the hospital's responsible for this project indicated that it would not be necessary to populate them. In this way, it was the object of study of this work to fill in the *Participant Identification Number (PIN)*, *Inclusion Criteria*, Demographics, *Cardiac baseline Assessment*, *Cardiac biomarkers_24* and *Cardiac biomarkers_HIGHER* tables.

3.3 Data Pipeline

Up to this point, the characteristics and specificities of the source database have already been presented. Besides that, all the source database tables with relevant information for the development of this work were also presented in more detail. On the other hand, a study of the target database was also performed in order to understand the information that needs to be collected for this project. It is now essential to present the Data Pipeline used to represent the ETL processes necessary for the development of this work.

The development of a Data Pipeline is a complex procedure that needs to consider all the specifics of the most common phases of an ETL process, namely: extracting the data from the source database to support regular data updates, considering the availability and operational changes in the source data systems; transforming the data according to specific business rules and dealing with data quality issues that compromise data integration and consistency; and loading the data considering the requirements of the target database [30].

The data had to be extracted from the previously described source database (section 3.1) to load all the necessary information into the target database. In addition, the data had to be correctly identified to ensure that only data from patient admissions that had COVID-19 were extracted and transformed before being uploaded to the target repository. The data extraction procedure involved fetching information regarding patient identification, study inclusion criteria data and demographics, cardiac assessment, and cardiac biomarkers. Once all the necessary data were available, the transformation phase took place, considering predefined data mappings, cleaning, and enrichment procedures.

One of the first tasks to be performed in the transformation process was the mapping between the columns of the source database and the target database, and this process is covered in more detail in section 3.4. The data mapping process is quite challenging regarding health data, as there is no standardization across all EHR systems. For example, the same parameters may have different names in each hospital, and some may be recorded in one hospital but not in another [23].

Figure 3.3 presents the ETL architecture used in this work, built to address the identified requirements. First, as the input of the Extraction Phase, the source database is specified, created from CSV files made available by the hospital under study. Then, in the transformation phase, a staging area is identified, whose purpose is to support the ETL workflow. Within the staging area, 3 layers are presented. The *Monitoring Layer* provides two components: *Lineage Control* and *Log Handler*. The first is responsible for monitoring the data lineage, and the second allows tracking all the main ETL events to provide information to identify errors or bottlenecks in the process.

Data quality is supported by the *Metadata Layer*, which uses a *Data Dictionary* component to analyze disease and diagnosis terms and metrics that differ from the hospital under study and those used in the European project. The *Metadata Layer* also captures and maintains all ETL metadata, including all domain-specific transformation logic.

Finally, the *Quarantine Layer* provides the mechanisms to handle unexpected records (e.g., values with unknown measurements or unexpected parameterized values) and identify specific rules to handle such cases automatically or to be reintroduced into the ETL stream after human evaluation.

Once the Extraction and Transformation phases are complete, the data must be loaded into the target database, as shown in Figure 3.3.

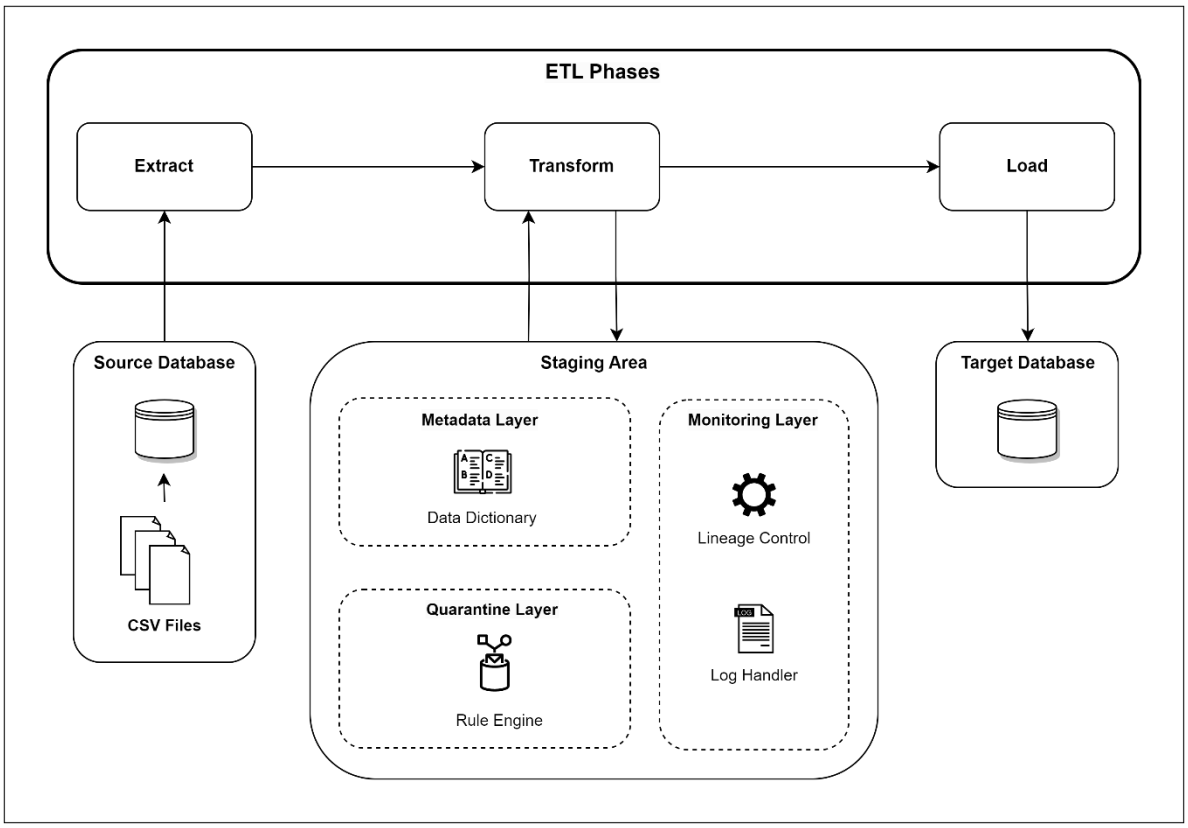

Figure 3.3 - ETL Architecture

3.4 Data Integration and Population of Target Tables

At this stage, the source database tables constitutions and the target database intended to be populated are already known. In addition, the ETL architecture used in this work was also presented, built to deal with the identified requirements. Thus, this section presents the main developments that allowed the population of the target tables. For this purpose, throughout this section, flowcharts of the processes that allowed the different target tables to be populated and examples of table records created to assist those processes and for the final tables of the target database, are presented.

The first primary objective of the development of this work was to select, from the hospital's total available admissions, only those concerning patients admitted with COVID-19. For this, through the "Get PICIS ADMISSIONS" activity represented in Figure 3.4, only the entries belonging to the PICIS system were filtered, the system where the admissions of patients with COVID-19 were recorded. Thus, from the 1159139 admissions initially available in the source database, 24574 PICIS admissions resulted. Next, only the admissions of patients with COVID-19 had to be selected since the PICIS system stores other types of admissions. For this purpose, 2 activities were responsible for this selection. "Get UNEICS ADMISSIONS" activity, where a join is performed between 7 tables of the source database and 1 of the staging area, selecting only the admissions of patients who were admitted on dates on or after March 2, 2020 (date of the first case of COVID-19 confirmed in Portugal), in UNEICS (units created in the pandemic in which all patients had COVID-19), totaling 553 entries. In addition, the "Get POSITIVE PATIENTS ADMISSIONS" activity was also developed, where a join is performed between 12 tables from the source database and 1 table from the staging area, and only the admissions of patients who were admitted on dates on or after March 2, 2020, and who tested positive for COVID-19 are selected, representing a total of 570 admissions.

Next is the "GET ALL COVID PATIENTS ADMISSIONS" activity, where the admissions of the 2 activities mentioned above are combined (1123 records). If the same admission is part of both activities, only 1 of the entries is considered. Thus, for the development of this work, 815 admissions of patients with COVID-19 and a total of 795 distinct patients are considered. The reason that there are more admissions than patients is explained by the fact that some patients have been admitted to the hospital more than once as a result of the disease caused by COVID-19.

Figure 3.4 shows the flowchart of obtaining patient admissions with COVID-19, where the previously mentioned activities, the tables that were the source for each activity, and the tables originated in each activity can be found. The names of the tables that served as the source of the activities and the tables originated in each activity have a standard format.

That is, not only the table name is displayed, but also the name of the database and the schema where the table is stored, as shown next: "<Database name>.<Database schema>.<Table name>". For example, the table *PICIS_ADMISSIONS* stored in the target database has the following format: capacity_db.Integration.*PICIS_ADMISSIONS*. In this way, it is possible to distinguish between the tables created to support the processes (Schema = Integration) and the final tables of the target database (Schema = dbo).

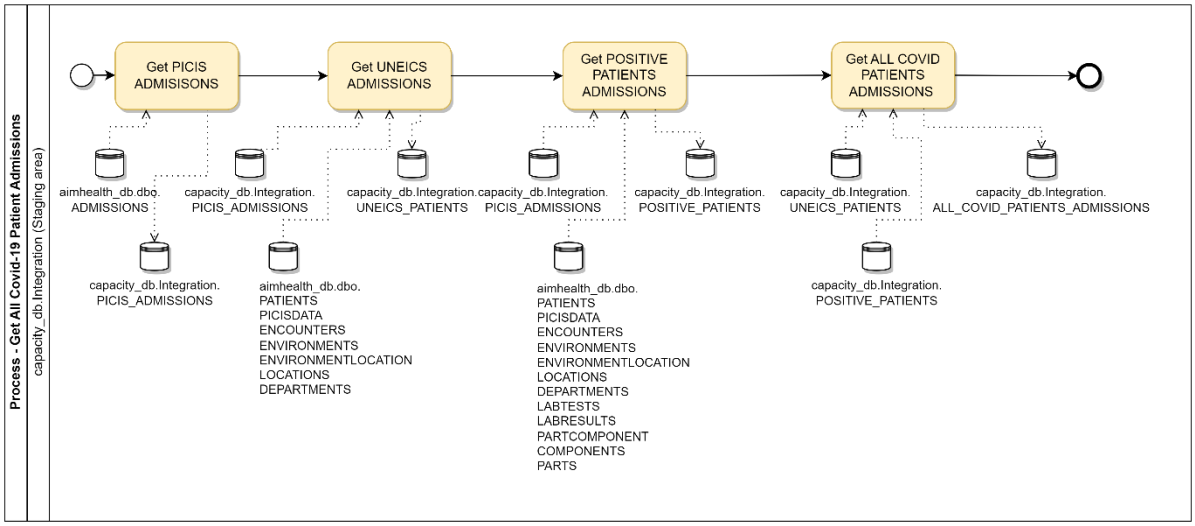

Figure 3.4 - Process Flowchart: "Get All Covid-19 Patient Admission"

In addition to the flowchart presented in Figure 3.4, Table 3.18 shows an example of a final table record from the above process: "Get All Covid-19 Patient Admission".

Table 3.18 - Example of a record from table *ALL_COVID_PATIENTS_ADMISSIONS*

ALL_COVID_PATIENTS_ADMISSIONS			
ADMISSIONDBOID	**PATIENTDBOID**	**COVID_DATE**	**HOSPITAL_ADMISSION**
4715884348813480552	1715884348765480552	2020-04-13	2020-03-24 0:11:00.000

Next, the diagrams with the processes developed for filling each target table considered in this work are presented. In addition, the chapter Appendices presents the template used for each process, called Source-to-Target mappings. This template was developed to document all the processes designed to allow their replication and monitoring. Through this document, it is possible to understand, for each table, which columns were used as data sources and the ETL rules applied, which allow obtaining each of the columns for the target database tables. Additionally, this template contains the name and description of the table to document, the number of variables and records and the name of the package developed in SSIS for this purpose.

Thus, following the logic of section 3.2, the first table to be presented is *Participant Identification Number (PIN)*. Populating this table required the development of 4 activities, as represented in Figure 3.5, which are described next.

First, the "Get PATIENT ADMISSION MAPPINGS" activity was developed to create, for each patient's admission, a new *subjid* column that unambiguously identifies this admission. An example of a record of the result of this activity can be found in Table 3.19.

Table 3.19 - Example of a record from table *PATIENT_ADMISSION_MAPPINGS*

PATIENT_ADMISSION_MAPPINGS		
ADMISSIONDBOID	**PATIENTDBOID**	**subjid**
4715884348813480552	1715884348765480552	918-1

Next, the "Get subjid" activity was created, which selects from the *PATIENT_ADMISSION_MAPPINGS* table, the *subjid* corresponding to each patient admission and also the "Get date_created and pin_complete" activity, which indicates for each record, the date on which it was created in the target database and whether all columns were filled or not or if this information was not verified. Lastly, the "POPULATE PIN TABLE" activity aims to insert in the *Participant Identification Number (PIN)* table the values obtained in the previous activities of this process. Table 3.20 is an example of a record from the *Participant Identification Number (PIN)* target table.

Table 3.20 - Example of a record from table *Participant Identification Number (PIN)*

Participant Identification Number (PIN)		
subjid	**datecreated**	**pin_complete**
918-1	2023-04-27	2

Figure 3.5 shows the process diagram described above ("Get Participant Identification Number Pin Table Data"). As can be seen by analyzing this figure, there are 2 lanes: the *capacity_db. Integration* and the *capacity_db.dbo*. The name of the lanes follows the following format: "<*Database name*>.<*Database schema*>". Thus, one can distinguish between the activities created to support the populating of the target tables (Schema = Integration) and the activities that enabled the populating of the target tables themselves (Schema = dbo). In other words, the activities inside the upper lane concern all the development performed in the Staging Area, that is, the activities necessary to create to support the populating of the target tables. On the other hand, in the lower lane, only 1 activity is presented, which consists of loading into the target table, all the data obtained in the previous activities.

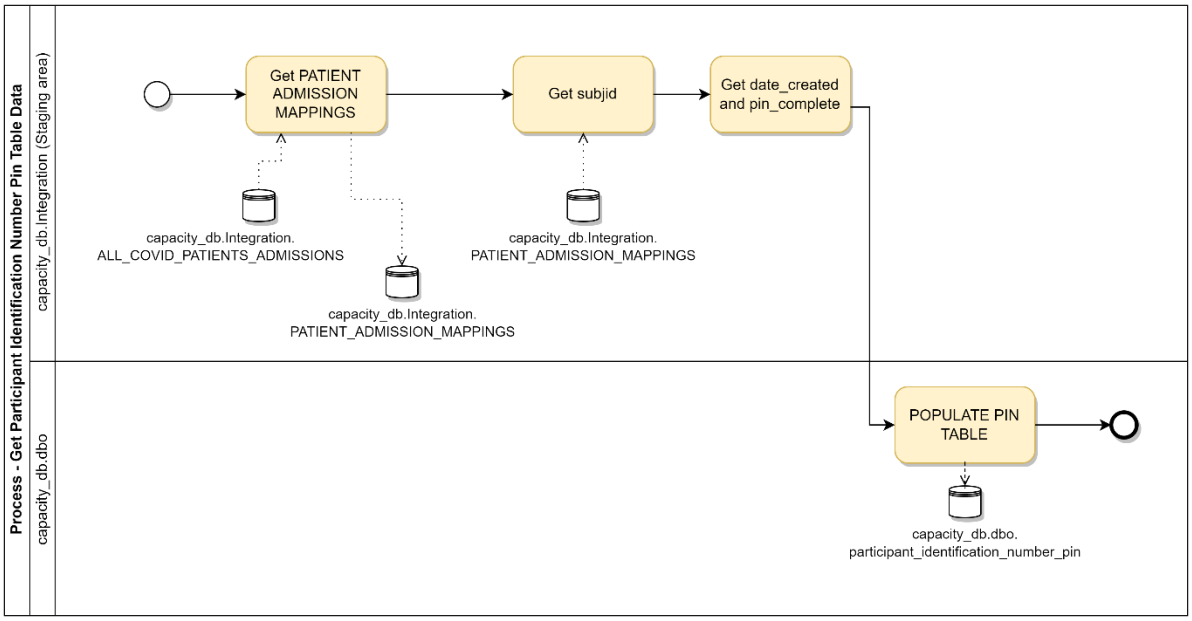

Figure 3.5 - Process Flowchart: “Get Participant Identification Number Pin Table Data”

Next, the *Inclusion Criteria* table is presented. Populating this table implied the creation of 6 activities, as represented in Figure 3.6, which are presented next.

First, the "Get COVID DATE" activity was created, which selects from the *ALL_COVID_PATIENTS_ADMISSIONS* table, the *COVID_DATE* column. This provides the information needed to populate the *dsstdat* and *corona_ieorres* columns of the target table. Then, the "Get COUNTRIES_PT_EN" activity was developed, which aims to create an auxiliary table containing the descriptions of the patient’s countries of residence (in Portuguese), the respective descriptions in English, and the respective country code to be considered in this study. Table 3.21 shows an example of a record from the *COUNTRIES_PT_EN* table.

Table 3.21 - Example of a record from table *COUNTRIES_PT_EN*

COUNTRIES_PT_EN		
COUNTRY_PT	**COUNTRY_EN**	**COUNTRY_EN_COD**
Alemanha	Germany	64

Next is the "Get COUNTRIES INFO" activity, which by joining the table built in the previous activity (*COUNTRIES_PT_EN*) with the *COUNTRIES* and *ADMISSIONS* tables from the source database, allows getting the information for the *country* and *oth_country* columns of the target table. Also, the "Get subjid" activity, involves getting the *subjid* column from the *PATIENT_ADMISSION_MAPPINGS* table. Beyond that, the "Get remaining columns" activity was created, which consists of defining the following columns:

- *sitename_nhr*, which has a fixed value of 918 (identifies the hospital under study);
- *sympt_epi_physical*, *sympt_epi_healthfac* and *sympt_epi_lab*, which have the value 3 (Unknown), since the hospital did not have this information available;
- *inclusion_complete*, which indicates whether or not all columns of the respective table were filled out or if this information was not verified;

Finally, the “POPULATE INCLUSION CRITERIA TABLE" activity was built, whose objective is to insert into the *Inclusion Criteria* target table the values obtained in the previous activities of this process. Table 3.22 is an example of a record from the table mentioned above.

Table 3.22 - Example of a record from table *Inclusion Criteria*

Inclusion Criteria									
dsstdat	**sitename_ nhr**	**country**	**oth_ county**	**corona_ ieorres**	**sympt_epi_ physical**	**sympt_epi_ healthfac**	**sympt_ epi_lab**	**inclusion_ complete**	**subjid**
2021-04-24	918	64	NULL	1	3	3	3	2	918-1

Figure 3.6 shows the diagram of the process described above ("Get Inclusion Criteria Table Data "). The logic behind the table and lane names is the same as in the abovementioned processes. Furthermore, in Appendix A, the corresponding Source-to-Target mappings template can be seen.

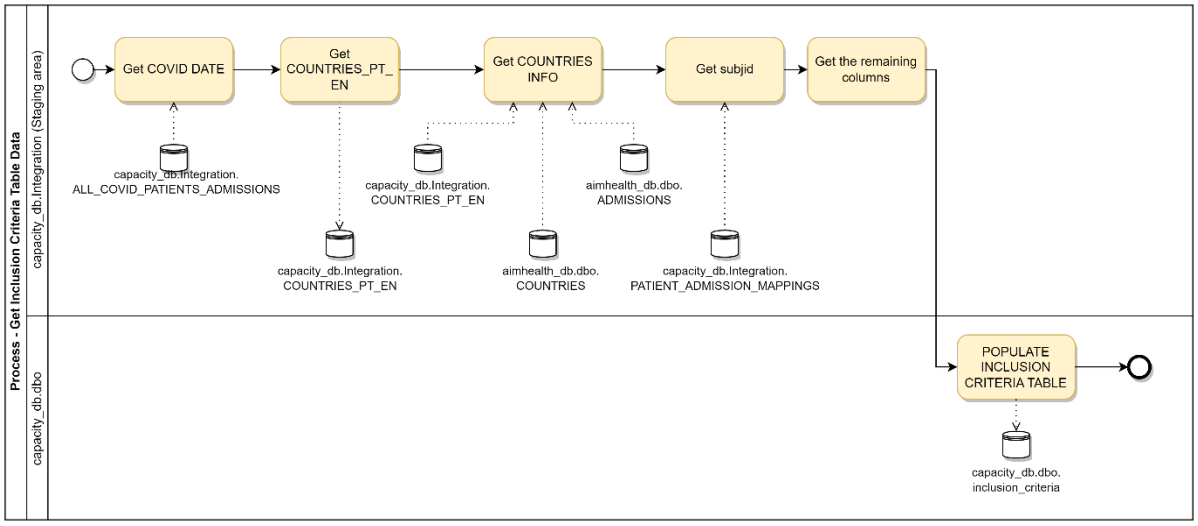

Figure 3.6 - Process Flowchart: “Get Inclusion Criteria Table Data”

As for populating the *Demographics* table, 6 activities were developed, as represented in Figure 3.7, described next. The first activity developed, "Get SEX AND AGE", which by performing a join between the staging area table *ALL_COVID_PATIENTS_ADMISSIONS* as the tables *ADMISSIONS*, *SEXES*, and *PATIENTS* of the source database, allows obtaining the necessary information for populating the columns *sex*, *age_estimateyears* and *age_estimateyearsu* of the target table in question.

Next, the "Get ETHNICITY" activity was created, which by performing a join between the *ALL_COVID_PATIENTS_ADMISSIONS* table and the *ETHNICITIES*, *ETHNICGROUPS* and *PATIENTS* tables of the source database, allows getting the data to populate the *ethnic__1*, *ethnic__2*, *ethnic__3*, *ethnic__4*, *ethnic__5*, *ethnic__6*, *ethnic__7*, *ethnic__8*, *ethnic__9*, *ethnic__10* and *oth_ethic* columns.

The activity "GET PREGNANCY AND POST PARTUM" was also created, that by joining the tables *ALL_COVID_PATIENTS_ADMISSIONS*, *PREGNAT_PATIENTS_ADMISSIONS* and *POST_PARTUM_PATIENTS_ADMISSIONS* from the Staging Area, allows obtaining the necessary information for populating the *pregyn_rptestcd*, *egestage_rptestcd*, *postpart_rptestcd* and *dlvrdtc_rptestcd* columns of the target table in question.

The *PREGNAT_PATIENTS_ADMISSIONS* table stores the data regarding pregnant patient admissions, and an example of a record from this table can be found in Table 3.23. This table contains the admission ID (*ADMISSIONDBOID*) and patient ID (*PATIENTDBOID*), the date when COVID-19 was confirmed (*COVID_DATE*) and the gestational period she was in (*GESTATION_PERIOD*).

The *POST_PARTUM_PATIENTS_ADMISSIONS* table contains the data related to postpartum patient admissions, and in Table 3.24, an example of a record from this table can be found. This table includes the admission ID (*ADMISSIONDBOID*) and patient ID (*PATIENTDBOID*), the date when COVID-19 was confirmed (*COVID_DATE*) and the date of birth of the baby (*DELIVERY_DATE*);

Table 3.23 - Example of a record from table *PREGNAT_PATIENTS_ADMISSIONS*

PREGNAT_PATIENTS_ADMISSIONS			
ADMISSIONDBOID	**PATIENTDBOID**	**COVID_DATE**	**GESTATION_PERIOD**
4788249939277769552	1788240207703769552	2020-06-09	29.00

Table 3.24 - Example of a record from table *POST_PARTUM_PATIENTS_ADMISSIONS*

POST_PARTUM_PATIENTS_ADMISSIONS			
ADMISSIONDBOID	**PATIENTDBOID**	**COVID_DATE**	**DELIVERY_DATE**
4788249939277769552	1788240207703769552	2020-06-09	2021-02-03 00:00:00.000

The activity "Get subjid" was also developed, which consists in getting the *subjid* column from the *PATIENT_ADMISSION_MAPPINGS* table, as well as "Get remaining columns" activity, which consists of defining the following columns:

- *healthwork_erterm* and *labwork_erterm*, which have the value 3 (Unknown), since the hospital did not have this information available;
- *pregout_rptestcd*, *aplb_lbperf*, *aplbb_lborres*, *apdm_age*, *apvs_weight*, *apvs_weightu*, *apsc_gestout*, *apsc_brfedind*, *apsc_brfedindy* and *apsc_vcageind* which have the value NULL, since they depend on the filling in of other columns, which in this case were not filled in due to lack of data by the hospital;
- *demographics_complete*, which indicates whether or not all columns of the respective table were filled out or if this information was not verified;

Lastly, the "POPULATE DEMOGRAPHICS TABLE" activity was also necessary, whose objective is to insert in the *Demographics* target table the values obtained in the previous activities of this process. Table 3.25 is an example of a record from the table mentioned above;

Table 3.25 - Example of a record from table *Demographics*

Demographics							
sex	**age_ estimateyears**	**age_ estimateyearsu**	**etnic__1**	**etnic__2**	**etnic__3**	**etnic__4**	**etnic__5**
2	39	2	0	0	0	0	0
etnic__6	**etnic__7**	**etnic__8**	**etnic__9**	**etnic__10**	**other_ ethnic**	**healthwork_ erterm**	**labwork_ erterm**
0	0	0	0	0	Gypsy	3	3
pregyn_ rptestcd	**egestage_ rptestcd**	**postpart_ rptestcd**	**pregout_ rptestcd**	**dlvrdtc_ rptestcd**	**aplb_ lbperf**	**aplb_ lborres**	**apdm_ age**
1	29.00	0	NULL	NULL	NULL	NULL	NULL
apvs_ weight	**apvs_ weightu**	**apsc_ gestout**	**apsc_ brfedind**	**apsc_ brfedindy**	**apsc_ vcageind**	**demographics_ complete**	**subjid**
NULL	NULL	NULL	NULL	NULL	NULL	2	918-10

The points described above can be seen by analyzing Figure 3.7, which represents the diagram of the "Get Demographics Table Data" process. The logic behind the table and lane names is the same as in the abovementioned processes.

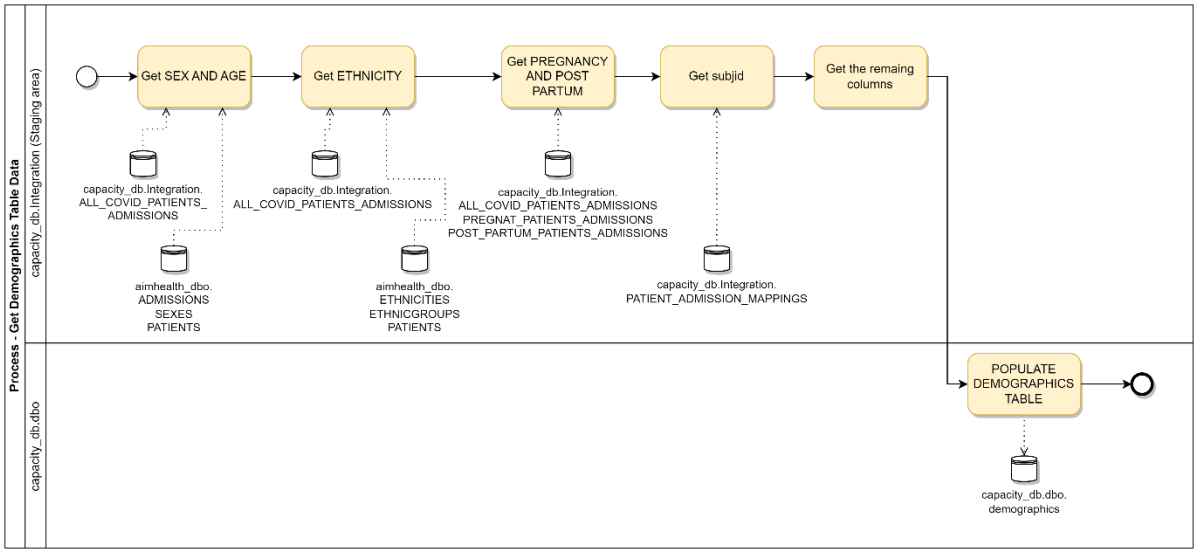

Figure 3.7 - Process Flowchart: “Get Demographics Table Data”

Given the large number of columns and the complexity of filling the *Cardiac Baseline Assessment* table, the strategy adopted to present the population of this table was different. For this process, the idea is to start at a higher abstraction level and go down to the most atomic level possible, presenting a higher level of detail. Thus, in Figure 3.8, this process's highest level of abstraction is presented. Through the analysis of this figure, it is possible to verify that there is 1 subprocess for each group of columns defined in Table 3.16 in section 3.2. Thus, each subprocess is responsible for filling the respective set of columns.

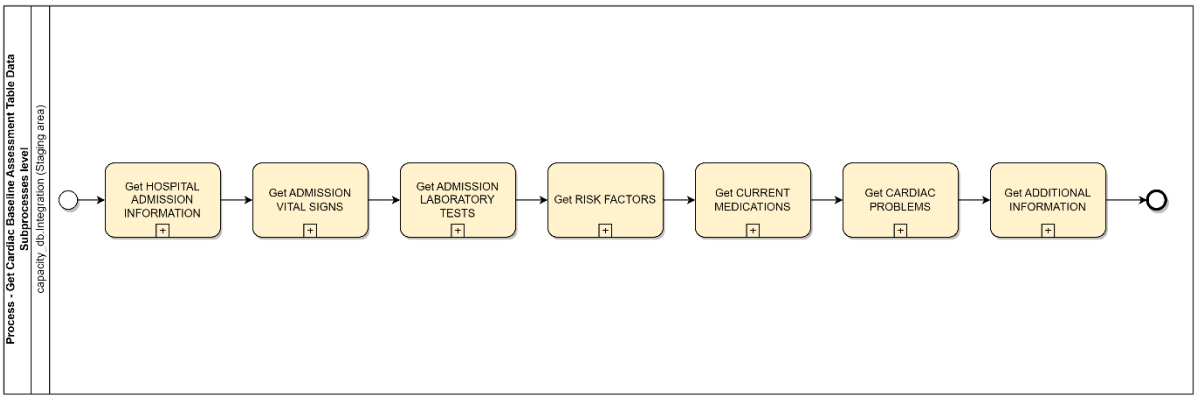

Figure 3.8 - Process Flowchart: “Get Cardiac Baseline Assessment Table Data – Subprocess level”

To demonstrate a lower level of abstraction than that presented in Figure 3.8, was used, as an example, the process: "Get Admission Laboratory Tests - Subprocess level", represented in Figure 3.9. This process is an expansion of the "Get ADMISSION LABORATORY TESTS" subprocess (Figure 3.8), presenting a higher level of detail. Analyzing Figure 3.9, it can be observed that this subprocess comprises 7 other subprocesses. Each of these new subprocesses represents the retrieval of each laboratory test performed on the patient's admission to the hospital. For example, the subprocess "GET ADMISSION PLATELET COUNT" consists of the set of activities required to obtain all the information related to the laboratory test that performs the platelet count in the blood.

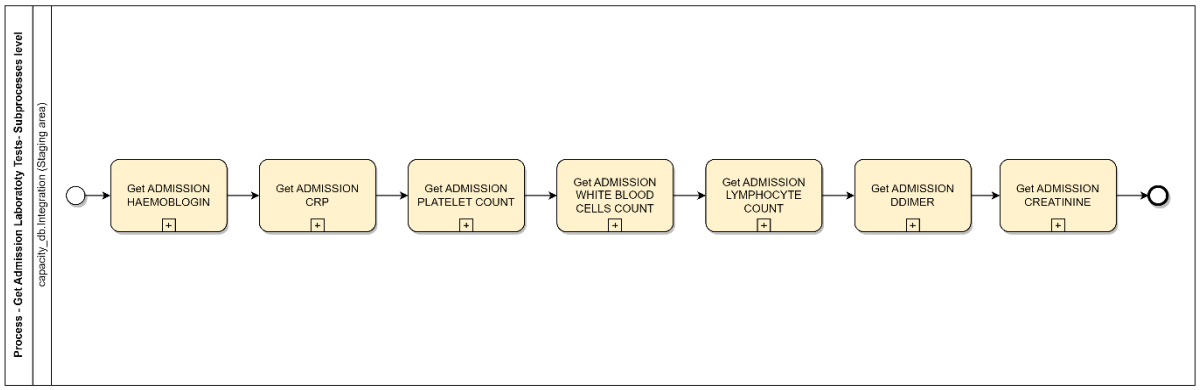

Figure 3.9 - Process Flowchart: "Get Admission Laboratory Tests – Subprocess level"

Taking the example of the "Get ADMISSION PLATELET COUNT" subprocess and drilling down again, ones get the "Get Admission Platelet Count - Subprocess level" process, represented in Figure 3.10. This process represents all the subprocesses comprising the "Get ADMISSION PLATELET COUNT" subprocesses" (Figure 3.9).

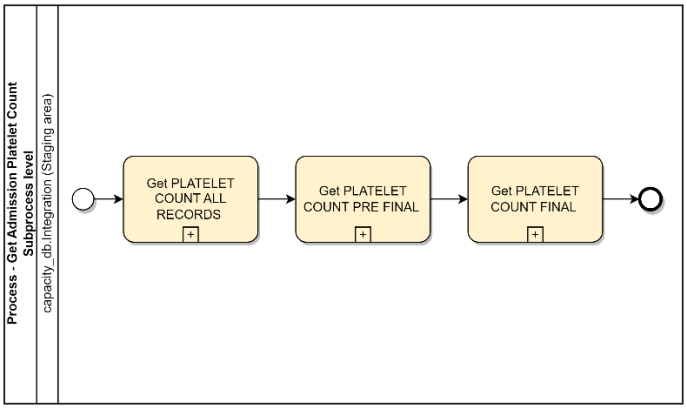

Figure 3.10 - Process Flowchart: "Get Admission Platelet Count – Subprocess level"

It is now essential to present the objective of each subprocess represented in Figure 3.10. The scripts developed under these subprocesses will be presented and analyzed to do this.

Figure 3.11 shows the script composition of the subprocess "Get PLATELET COUNT ALL RECORDS" (Figure 3.10). The block represented by a "2" contains the joins between the 12 tables needed to obtain the desired columns. Represented with a "3" is the application of a necessary filter to select only the component ID (*COMPONENTDBOID*) referring to the platelet count, and also a filter that selects only the information referring to platelet count tests performed after the date of admission, considered in this study, of the patient to the hospital. In the block represented by a "1" are all the columns needed to obtain the relevant information about the platelet count test. This information is loaded into an auxiliary table (*PLATELET_COUNT_ALL_RECORDS*) that contains the information about the date when the platelet count test was performed (*RESULT_DATE*), the value of the test result (*RESULT_PLATELET*), the unit of measurement used (*PLATELET_UNIT*) and also the type of test (*EXAM_TYPE*). In addition, this table comprises the identifier of the admission (*ADMISSIONDBOID*) and the patient who performed the test (*PATIENTDBOID*), the date of the patient's admission to the hospital (*HOSPITAL_ADMISSION*) and the difference (in hours) of the date the test was performed and the admission date (*TEST_ADMISSION_DIFF*). An example of a record from this table is shown in Table 3.26.

Table 3.26 - Example of a record from table *PLATELET_COUNT_ALL_RECORDS*

PLATELET_COUNT_ALL_RECORDS						
ADMISSIONDBOID & PATIENTDBOID	**RESULT_DATE**	**RESULT_ PLATELET**	**HOSPITAL_ ADMISSION**	**TEST_ ADMISSION_ DIFF**	**EXAM_ TYPE**	**PLATELET_ UNIT**
4774022679215769552 1747850744488769552	2020-05-14 18:34:11.000	174	2020-05-14 14:24:11.000	4	Plaquetas, contagem	x10\E^E\9/L

```
CREATE PROCEDURE capacity_db.Integration.Get_PLATELET_COUNT_ALL_RECORDS

AS
BEGIN

INSERT INTO capacity_db.Integration.PLATELET_COUNT_ALL_RECORDS

SELECT  DISTINCT
    A.ADMISSIONDBOID AS ADMISSIONDBOID,
    A.PATIENTDBOID AS PATIENTDBOID ,
    LR.OBSERVATIONTIME AS RESULT_DATE,
    LR.TEXTVALUE AS RESULT_PLATELET,
    A.HOSPITAL_ADMISSION AS HOSPITAL_ADMISSION,
    DATEDIFF(hh,A.HOSPITAL_ADMISSION,LR.OBSERVATIONTIME) AS TEST_ADMISSION_DIFF,
    C.COMPONENTDESC AS EXAM_TYPE,
    U.UNITSYMBOL AS PLATELET_UNIT

FROM
    capacity_db.Integration.ALL_COVID_PATIENTS_ADMISSIONS A
INNER JOIN
    aimhealth_db.dbo.PICISDATA PD ON PD.ADMISSIONDBOID=A.ADMISSIONDBOID
INNER JOIN
    aimhealth_db.dbo.ENCOUNTERS E ON E.PICISDATADBOID=PD.PICISDATADBOID
INNER JOIN
    aimhealth_db.dbo.ENVIRONMENTS ENV ON ENV.ENCOUNTERDBOID=E.ENCOUNTERDBOID AND ENV.PICISDATADBOID=PD.PICISDATADBOID
INNER JOIN
    aimhealth_db.dbo.ENVIRONMENTLOCATION EL ON EL.ENVIRONMENTDBOID=ENV.ENVIRONMENTDBOID
INNER JOIN
    aimhealth_db.dbo.LOCATIONS L ON L.LOCATIONDBOID=EL.LOCATIONDBOID
INNER JOIN
    aimhealth_db.dbo.DEPARTMENTS D ON D.DEPTDBOID=L.DEPTDBOID
INNER JOIN
    aimhealth_db.dbo.LABTESTS LT ON LT.PICISDATADBOID=PD.PICISDATADBOID
INNER JOIN
    aimhealth_db.dbo.LABRESULTS LR ON LR.LABTESTDBOID=LT.LABTESTDBOID
INNER JOIN
    aimhealth_db.dbo.PARTCOMPONENT PC ON PC.DBOID=LR.PARTCOMPONENTDBOID
INNER JOIN
    aimhealth_db.dbo.COMPONENTS C ON C.COMPONENTDBOID=PC.COMPONENTDBOID
INNER JOIN
    aimhealth_db.dbo.UNITS U ON U.UNITDBOID = LR.UNITDBOID

WHERE
    C.COMPONENTDBOID IN (26369601174493097552) AND DATEDIFF(hh,A.HOSPITAL_ADMISSION,LR.OBSERVATIONTIME) >= 0

END
GO
```

Figure 3.11 – “Get PLATELET COUNT ALL RECORDS" subprocess script

By running the script represented in Figure 3.11, it is possible to obtain all the relevant information about all the platelet count tests the patients have had since the hospital admission considered in this study. It is now intended to obtain only the information of the first platelet count test performed since the patient's admission to the hospital. For this purpose, by analyzing Figure 3.12, it is possible to observe the script that allows obtaining this information.

```
CREATE PROCEDURE capacity_db.Integration.GetPLATELET_COUNT_PRE_FINAL

AS
BEGIN

INSERT INTO capacity_db.Integration.PLATELET_COUNT_PRE_FINAL

SELECT
    mt.*
FROM
    capacity_db.Integration.PLATELET_COUNT_ALL_RECORDS mt INNER JOIN (
        SELECT
            ADMISSIONDBOID,PATIENTDBOID, MIN(RESULT_DATE ) AS MinDate
        FROM
            capacity_db.Integration.PLATELET_COUNT_ALL_RECORDS
        GROUP BY
            ADMISSIONDBOID,PATIENTDBOID

    ) t ON (mt.ADMISSIONDBOID = t.ADMISSIONDBOID AND  mt.PATIENTDBOID  = t.PATIENTDBOID) AND mt.RESULT_DATE = t.MinDate

END
GO
```

Figure 3.12 – “Get PLATELET COUNT PRE FINAL" subprocess script

After running the script in Figure 3.12, all relevant information about the first platelet count test for each patient admission to the hospital was stored in the auxiliary table "PLATELET_COUNT_PRE_FINAL". Now, it is necessary to select only those records from that table whose exam data does not exceed the patient's hospital admission date by 24 hours (a project requirement). The script represented in Figure 3.13 ensures that only the admissions of each patient and the platelet count test information corresponding to the first 24 hours of the patient's admission to the hospital are included in the script output table. In this way, it is possible to fill in the *admission_platelet_available*, *admission_platelet* and *admission_platelet_unit* columns that represent, respectively, if the platelet count was available up to 24 hours after the patient's admission to the hospital and, if so, what are the value of the test result and the unit of measurement.

```
CREATE PROCEDURE capacity_db.Integration.GetPLATELET_COUNT_FINAL

AS
BEGIN

INSERT INTO capacity_db.Integration.PLATELET_COUNT_FINAL

SELECT
    ADMISSIONDBOID,
    PATIENTDBOID,
    RESULT_DATE,
    RESULT_PLATELET,
    HOSPITAL_ADMISSION,
    TEST_ADMISSION_DIFF,
    EXAM_TYPE,
    PLATELET_UNIT
FROM
    capacity_db.Integration.PLATELET_COUNT_PRE_FINAL WHERE TEST_ADMISSION_DIFF <= 24

END
GO
```

Figure 3.13 - “Get PLATELET COUNT FINAL" subprocess script

Thus, in the part of the ETL flow that loads the information related to the laboratory test that performs the platelet count in blood, the information is obtained through the auxiliary table *PLATELET_COUNT_FINAL*. That is, for each patient admission, it is verified if there is any record with platelet count information in the auxiliary table mentioned above. This check is performed through a join, as shown in Figure 3.14. If a particular admission has no match, the values of the columns to be created become NULL. Otherwise, a particular record receives the corresponding value from the column stored in the *PLATELET_COUNT_FINAL* table.

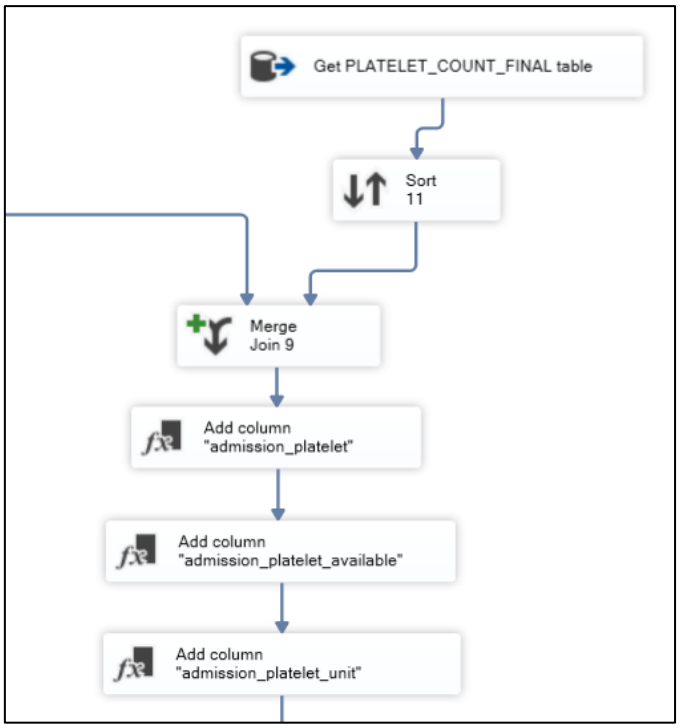

Figure 3.14 - Piece of the ETL flow for creating the columns related to the platelet count in blood

Finally, it remains to present the process for filling in the *Cardiac Biomarkers* table. Although there are 2 distinct instances of this table (*Cardiac_biomarkers_24* and *Cardiac_biomarkers_HIGHER*), only the flowchart of the *Cardiac_biomarkers_24* table is presented since they are similar processes.

Thus, in Figure 3.15 the process "Get Cardiac Biomarkers 24 - Subprocess level" is represented, where the highest level of abstraction of this process is found. That is, the subprocesses necessary to get this table's columns are represented. For example, the subprocess "Get TROPONIN 24" allows to get the columns *bio_trop*, *bio_trop_unit* and *bio_trop_value*, and the subprocess "Get the remaining info" allows to get the columns *date_biomarker*, *cardiac_biomarkers_complete* and *subjid*.

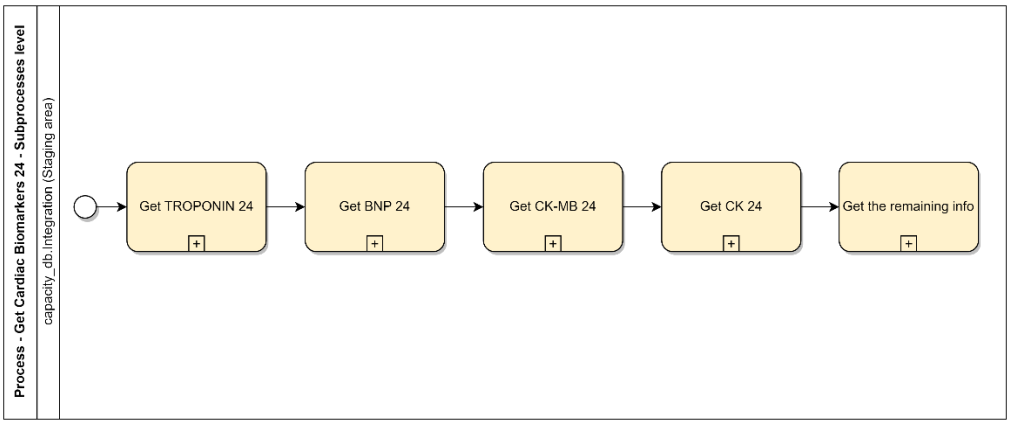

Figure 3.15 - Process Flowchart: “Get Cardiac Biomarkers 24 – Subprocess level”

CHAPTER 4

Implementation and Evaluation

This chapter serves as an introduction to the implementation of the system, focusing on the critical software tools employed and their importance in the overall process. It provides an overview of the ETL flow, highlighting its significance in the system's functionality. Furthermore, this chapter presents a specific example illustrating the system's operation, allowing for a practical understanding of its capabilities. In addition to discussing the system's implementation, this chapter introduces the evaluation metrics used to assess the efficiency and effectiveness of the ETL process. A dedicated Dashboard is also presented as a tool for the evaluation, enabling a comprehensive analysis of the system's performance.

4.1 Implementation Details

This work's development involved using several software tools, each with a specific purpose. The following are the tools used and their purposes for developing this work.

- *SQL Server 2019*[7]: a relational database management system (RDBMS) software designed to store data in a structured manner. It was utilized in this work to create and manage both the source and target databases, providing a reliable and organized storage solution;
- *SQL Server Management Studio 19 (SSMS)*[8]: a graphical interface tool to administer and manage SQL Server instances. In this particular work, it served as the primary interface for tasks such as database management, development, and maintenance. In addition, it provided an intuitive environment to handle and oversee the required databases effectively;
- *Visual Studio 2019*[9]: an integrated development environment (IDE) that offers a comprehensive range of tools for software development. In conjunction with *SQL Server Integration Services*[10], it was utilized to construct the entire ETL flow. These tools facilitated the development of custom scripts and provided a powerful platform for designing, debugging, and deploying the required workflows;

[7] https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2019
[8] https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms
[9] https://my.visualstudio.com/Downloads?q=visual%20studio%202019&wt.mc_id=o~msft~vscom~older-downloads
[10] https://marketplace.visualstudio.com/items?itemName=SSIS.SqlServerIntegrationServicesProjects

By employing *SQL Server 2019*, *SQL Server Management Studio 19*, *Visual Studio 2019*, and *SQL Server Integration Services*, this work successfully leveraged these software tools to store and manage data efficiently, administer SQL Server instances effectively, and develop and execute complex data integration and transformation workflows through the ETL process. In addition, it is essential to remind that all data in any of the tools presented was kept in a private SQL server, where access was limited, and all sensitive data was anonymized.

Once the tools used in the development of this work have been presented, as well as its objective, it is now essential to delve into the details of the implementation of this system. First, a project was created in Visual Studio 2019, where the entire ETL flow was developed. This project includes all the necessary connections for the communication between the source and target databases to the project, as well as the variables and packages needed for the ETL development. In total, 32 packages were developed to extract, process and load data into tables, including the final project tables and the auxiliary tables stored in the Staging Area. The only exception was the creation of a package designed to run the remaining 31 packages at once.

The contents of the 31 packages responsible for the ETL flow follow the logic presented in Figure 4.1. Each package has a Control Flow that manages the execution sequence and a Data Flow that focuses on the movement and transformation of the data within the package, which is obtained by expanding the task in Figure 4.1, represented by a "3". The Control Flow always includes the following tasks:

- "Clear Tables": deletes records from all tables that must be empty before running the package;
- "Get Lineage Key": executes a procedure to create a new record associated with the current run in the *Lineage* table;
- "Data Processing and Cleaning": this activity expands, leading to the Data Flow;
- "Update Lineage Table": updates the previously created record in the *Lineage* table if the package is successfully executed;

The Data Flow obtained by expanding the "Data Processing and Cleaning" task in Figure 4.1 represents the typical data movement and transformation process. First, data is usually extracted from several tables and then merged to get information between them. Next, new columns are created based on predefined rules. This process can be repeated as many times as necessary to create new columns. For records suspected of errors, validation rules have been created. Records are sent to the respective quarantine table if these rules are unmet. When all records are ready to be inserted into the final table, the number of records that will be inserted is saved. Finally, there is always an activity to map the newly created columns with those of the respective target table.

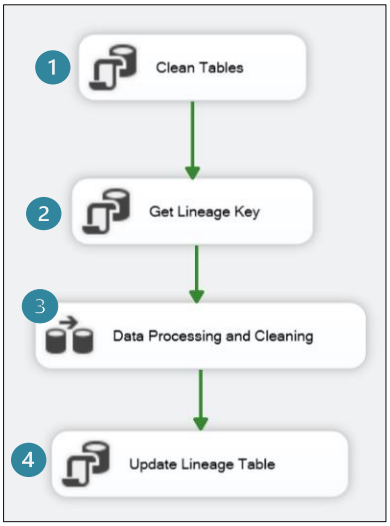

Figure 4.1 - Control Flow

4.2 System Demonstration

To demonstrate the system's functioning, the ETL flow created to populate the table *Cardiac Baseline Assessment* from the target database is used. This flow is stored in a single package, following the logic described in Figure 4.1.

First, the process is initiated by cleaning the records in both the target table and the associated quarantine table (Figure 4.1, represented by a “1”). This step ensures that no duplicate data is loaded. Next, a Stored Procedure stored in the Staging Area (Figure 4.1, represented by a “2”) is executed. This procedure loads the *Lineage* table with the following contents:

- "Lineage_Key": the line ID is entered, which is 1 in this case, as it represents the first execution of this example;
- "Date_Load_Started": the date when the loading started is recorded;
- "Table_Name": the table name where the data will be inserted is entered;
- "N_Record_Uploaded": the value 0 is inserted, as there is no data uploaded yet;
- "Date_Load_Completed": the value NULL is entered as the end date of the upload is not known at the start of the process;
- "Was_Successful": the value 0 (Unsuccessful) is entered as it is unknown whether the upload will be successful or not at the beginning of the process;

Figure 4.2 presents an example of a record created in the *Lineage* Table for the population of the *Cardiac Baseline Assessment* table resulting from the execution of the "Get Lineage Key" task (Figure 4.1, represented by a “2”).

	Lineage_Key	Data_Load_Started	Table_Name	N_Record_Uploaded	Data_Load_Completed	Was_Successful
1	1	2023-05-29 06:24:49.943	cardiac_baseline_assessment	0	NULL	0

Figure 4.2 - Example of a record from table *Lineage:* Beginning of the process

In the task "Data Processing and Cleaning" (Figure 4.1, represented by a “3”) all the data treatment and cleaning are performed to populate the target table correctly. The columns of the *Cardiac Baseline Assessment* table are grouped into 7 distinct groups, as shown in Table 3.16 in section 3.2. The ETL flow for this table considers this grouping, processing the columns from the first group (*Hospital Admission Information*) to the last group (*Additional Information*).

To demonstrate an example of how the process works, the piece of the process that allows the creation of the *admission_temp* column from the *Vital Signs* column group was used. This column stores the patient's body temperature recorded up to 72 hours after admission to the hospital. Figure 4.3 illustrates the Data Flow that allows the creation of this column, which will be described next.

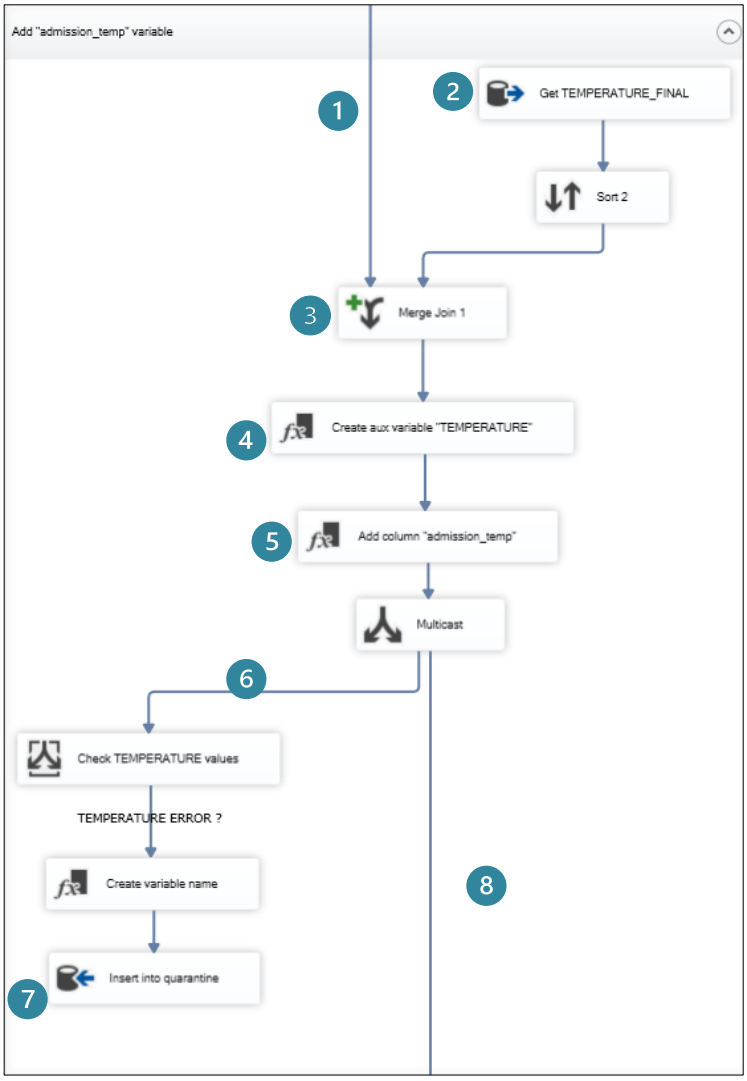

Figure 4.3 – Data Flow: Add *admission_temp* column

First, the flow marked as "1" is derived from the earlier cleaning and transformation processes performed in this table. That is, if, at this moment, the ETL flow for the *admission_temp* column of the *Vital Signs* group is being started, it means that the columns of the previous group (*Hospital Admission Information*) have already been treated.

Then, merge (Figure 4.3, represented by a “3”) is performed between the data from flow marked as "1" and the data from the flow obtained from the table *TEMPERATURE_FINAL*, using the column *ADMISSIONDBOID* (Figure 4.3, represented by a “2”). The *TEMPERATURE_FINAL* table contains the columns related to the patient's body temperature (*CD06*, *C057*, and *C050*). Multiple columns store this information because the hospital staff records it differently. For instance, some staff use the variable *CD06*, while others use variables *C057* or *C050*.

The output of the merge includes all the columns from the previous flow (Figure 4.3, represented by a “1”) along with the three columns from the *TEMPERATURE_FINAL* table.

Consequently, an auxiliary variable, *TEMPERATURE*, is created (Figure 4.3, represented by a “4”). For each record, this variable determines which column (*CD06*, *C057*, or *C050*) contains the patient's body temperature information and stores that information.

In the subsequent task (Figure 4.3, represented by a “5”), validation is performed to check if the value stored in the *TEMPERATURE* variable falls within the acceptable range (30 to 45). If the value is out of this range, the *admission_temp* column is considered NULL; otherwise, it takes the value from the *TEMPERATURE* variable.

The flow then splits into two branches. In branch "6", the records considered as possible errors are inserted into the corresponding quarantine table (Figure 4.3, represented by a “7”) for further evaluation. The records tagged as possible errors are associated with the patient and admission IDs and the column's name where the potential error was detected. An example of the output of this table can be found in Figure 4.4.

	PATIENTDBOID	ADMISSIONDBOID	ERROR_VALUE	VARIABLE
1	1754599056846769552	4787509417893769552	19.740	admission_temp
2	1787922764102000010	4787922764180000010	28.560	admission_temp
3	1789590454121769552	4789591505935769552	67.000	admission_temp
4	1552302202053119552	4799411754454769552	168.000	admission_temp
5	1807905149841769552	4807974769010769552	12.000	admission_temp

Figure 4.4 - Example of records from table *Quarantine Cardiac Baseline Assessment*

The values identified as possible errors were shared with the Cardiologist who assisted in developing this work and, after validation, were confirmed as errors. Therefore, these records were not considered in this work.

Flow "8" serves as the input for the next part of the process, which includes all columns created so far, including the *admission_temp* column described earlier.

To complete the "Data Processing and Cleaning" task (Figure 4.1, represented by a "3"), additional steps are required, depicted in Figure 4.5. These steps are always performed at the end of this process and allow the mapping of the created columns to the columns of the respective target table.

The flow marked with a "1" contains all columns created in the ETL process for the *Cardiac Baseline Assessment* table. Next, a merge is performed (Figure 4.5, represented by a "3") between the data from flow "1" and the data from the flow obtained from the table *PATIENT_ADMISSION_MAPPINGS* (Figure 4.5, represented by a "2"). Then, using the *ADMISSIONDBOID* and *PATIENTDBOID* columns, is obtained the *subjid* column, which represents the unique identifier of the patient admission. Next, in the "Row Count" task (Figure 4.5, represented by a "4"), the number of records produced is stored in the *rowcount* variable. Finally, in task "5", the mapping between the created columns and the final table columns is performed, and these columns are loaded into the *Cardiac Baseline Assessment* table.

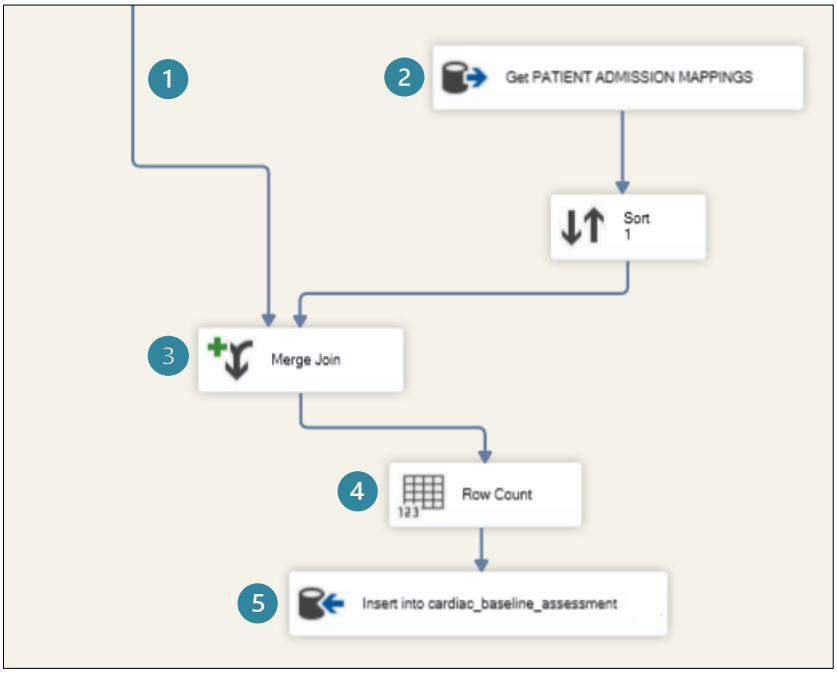

Figure 4.5 - Mapping patient admission

Thus, the "Data Processing and Cleaning" activity in Figure 4.1 is complete, leaving only the "Update Lineage Table" activity in Figure 4.1 to be performed.

To complete the Control Flow as depicted in Figure 4.1, the "Update Lineage Table" task ("4") is executed. This task updates the *Lineage* table. At this point, if the flow has reached this stage, it indicates a successful upload. The following columns of the *Lineage* table are updated:

- "N_Record_Uploaded": The total number of records inserted in the table is entered using the *rowcount* variable;
- "Data_Load_Completed": The date marking the end of the table upload is recorded;
- "Was_Successful": The value 1 (Success) is entered;

If the upload fails, the *Lineage* table remains unchanged with the values entered at the start of the loading process. This enables users of the system to determine whether the upload was successful or not. Figure 4.6 represents the update of the previously created record in the *Lineage* table.

	Lineage_Key	Data_Load_Started	Table_Name	N_Record_Uploaded	Data_Load_Completed	Was_Successful
1	1	2023-05-29 06:24:49.943	cardiac_baseline_assessment	815	2023-05-29 06:25:13.597	1

Figure 4.6 - Example of a record from table *Lineage*: End of the process

4.3 Results Evaluation

This chapter focuses on the analysis of the implemented system by evaluating the results obtained. The evaluation is conducted using three key metrics: the percentage of columns populated per target table, the overall percentage of populated columns, and the percentage of records populated by columns of each target table. These metrics provide a comprehensive understanding of the efficiency and effectiveness of ETL process.

With these metrics, it is possible to perform a global analysis of the implemented system, realizing how the ETL process went. For this, the Dashboard represented in Figure 4.7 was developed, which served as a basis for concluding the results obtained. In this Dashboard are represented 8 graphs distributed in the following way:

- Graph 1 - where the percentages of columns populated by each of the target tables are represented;
- Graph 2 - where the global percentage of populated columns is represented;
- Graphs 3 to 8, where the percentages of records populated by each column of each target table are plotted;

In all charts, a standard color code was used, where green represents the results considered satisfactory (Percentages between 75% and 100%), yellow represents the results considered intermediate (Percentages between 50% and 74%), and red represents the results considered unsatisfactory (Percentages between 0% and 49%).

Furthermore, in Charts 3 to 8, only the columns applicable to the hospital under study are presented. For example, in Graph 4, which represents the percentage of populated records for each column of the *Demographics* table, only 25 of the 32 columns in this table are presented. This is because filling in the 7 columns that are not presented does not apply to the hospital under study.

Analyzing Chart 1 in Figure 4.7, it is possible to observe that for 5 of the 6 target tables, the percentage of populated columns was very satisfactory (with results above 87%). The exception was the *CBA* (*Cardiac Baseline Assessment*) table, where it was only possible to populate 16% of the total columns. As for Charts 3 and 7 in Figure 4.7, which represent the percentage of populated records for the columns of the *PIN* (*Participant Identification Number*) and *Inclusion Criteria* tables, it is possible to observe that 100% of the records in this table were populated. This indicates that for these 2 tables, it was possible to obtain or create all the necessary information for their population.

As for Graph 4, it can be seen that the records in 19 of the 25 columns of the *Demographics* table were fully populated, and there are still 2 columns, *age_estimateyears* and *age_estimateyearsu*, with practically all the records filled in (98%). There is also the column *dlvrdtc_rptestcd*, which stores the date of birth of the baby of the pregnant patient with COVID-19, where only 2 of the 4 possible birth dates could be obtained (50%). Finally, there are 3 columns where no records could be populated (0%). That is, information about whether the baby coming from the pregnant patient with COVID-19 was tested for COVID-19 and its test result (*aplb_lbperf* and *aplb_lborres*, respectively), as well as the pregnancy outcome (*pregout_rptestcd*), could not be obtained from the study hospital.

As for Graph 5 in Figure 4.7, in 46.2% of the columns in the *Cardiac Biomarkers HIGHER* table, it was possible to obtain 100% of the records. On the other hand, information regarding the measurement unit and the Troponin, BNP, and CK-MB values could only be obtained in 74%, 48%, and 2% of the records, respectively. This means some patients either did not perform these tests or the information about these tests was not recorded. About the *date_biomarker* column, which aims to store the date on which the cardiac biomarkers were measured, it was not possible to populate any record. This is because it is intended to obtain a single date for the measurement of the various cardiac biomarkers, and in the hospital under study, the different biomarkers were measured on different dates. Since a single date was unavailable, it was decided not to include any of these dates.

As for Chart 6, which represents the groups of columns of the *CBA* table, only 8.3% of the columns (columns of the *Additional Info* and *Laboratory Tests* groups) had more than 75% of records filled. This is followed by the groups *Vital Signs*, *Risk Factors*, and *Admission Info*, where only 55%, 31%, and 17%, respectively, of the total records could be populated. As for the groups of columns *Cardiac Problems* and *Current Medication*, which represent 67.1% of the total columns of the *CBA* table, no information could be obtained.

As for Chart 8 in Figure 4.7, in 54.5% of the columns in the *Cardiac Biomarkers 24* table, it was possible to obtain 100% of the records. On the other hand, information regarding the unit of measurement and the Troponin and BNP value could only be obtained in 52% and 24% of the records, respectively. Similarly to what happened in the *Cardiac Biomarkers HIGHER* table, it was not possible to obtain the records of the *date_biomarker* column.

Finally, Graph 2 shows the overall percentage of populated columns. By analyzing this graph, it is possible to observe that the results are not very satisfactory, since only 29% of the total possible columns of the target tables were populated. This can be explained by the fact that the tables where good results were obtained have a much smaller number of columns than those with unsatisfactory results. For example, the *PIN* and *Inclusion Criteria* tables, where it was possible to populate all columns, only represents 3.7% of the total number of columns in all target tables. On the other hand, on the *CBA* table, which represents 91.7% of the total columns of all target tables, it was only possible to populate 16% of the total columns of that table. That is, the overall results are hampered by the fact that there is a huge table where the vast majority of columns could not be obtained, either due to unavailability of data from the hospital or the existence of information in clinical notes that can only be obtained with Text Mining techniques. Thus, if the *CBA* table was excluded, the results would be much more satisfactory, where 91.9% of the columns would be populated.

Figure 4.7 - Dashboard for results evaluation

CHAPTER 5

Conclusions

The conclusion of this thesis represents the closing of an intense and enlightening journey in the field of Health Data Integration. The challenges and opportunities related to data analysis, integration and management in the medical context have been explored in the previous chapters. This final chapter summarizes the main results and contributions achieved by this thesis, highlighting how the related work and the defined objectives contributed to the development of this thesis and outlining future perspectives to improve healthcare data management further. This chapter closes not only this work but also opens doors to new possibilities and discoveries in the field of data-driven healthcare.

5.1 Discussion

First, related work played a crucial role in developing this thesis by providing a solid knowledge base on Health Data Integration, especially in the context of cardiovascular disease and COVID-19. The literature review provided an understanding of the existing challenges, solutions, and gaps in the field, providing an overview of concepts and approaches related to Health Data Integration, including data models, interoperability, data extraction, transformation and loading, and quality assurance. It also highlighted the importance of Health Data Integration in the context of COVID-19, justifying the relevance and objectives of the study. The literature review also identified gaps in current knowledge and practices, directing the work and defining the specific goals of this thesis.

Regarding the first objective (G1) of this research study, which was to analyze the characteristics, specificities and information provided by the hospital under study, it can be considered fulfilled. The analysis performed on the source database revealed the existence of 138 tables with distinct scopes and particularities. These tables were grouped into sets according to their relations, particularities, and domain, resulting in 16 sets of tables. The analysis of the tables revealed relevant information about the patient's admission to the hospital, such as the patient's admission identifier, the patient's hospital admission and discharge dates, and the patient's height and weight on the date of admission to the hospital. In addition, other tables were used to analyze laboratory test components, physiological data, patient demographic information, laboratory test results, and physiological variable codes. In total, 20 tables from the source database were analyzed for the development of this thesis. Through this analysis, it was possible to understand the characteristics and information contained in the data provided by the hospital, contributing to the correct integration of these data in the repository shared within the European project.

The second proposed objective (G2), which was to analyze the characteristics and singularities of the target database to store health data related to cardiovascular disease, was achieved. The structure of the target database was created based on the documentation provided, ensuring the privacy of patient data. In total, 14 tables were identified, with 812 columns distributed among them. The focus of this work was on the first 5 tables: Participant Identification Number (PIN), *Inclusion Criteria*, *Demographic*, *Cardiac Baseline Assessment*, and *Cardiac Biomarkers*. Each table was analyzed in detail, describing the columns, their data types, and possible values. The analysis provided a comprehensive understanding of the data to be stored in each table, including identification information, patient assessments, demographic data, cardiac baseline assessment, and cardiac biomarkers. The document successfully describes the structure and information of the target database, meeting the project requirements and respecting patient privacy.

It can also be concluded that the goal of developing a robust and efficient ETL architecture (G3) that covers data lineage control, erroneous record handling, metadata analysis, and data quality assurance mechanisms for the ETL process of data from the source database to the target database has been achieved. The ETL architecture presented in this thesis demonstrates a comprehensive approach to address the identified requirements. By extracting the data from the source database, correctly identifying the relevant data, and applying appropriate transformations, it was possible to ensure data integration and consistency. In addition, mechanisms for lineage control, incorrect record handling, metadata analysis, and data quality assurance were implemented to ensure data traceability, documentation, and compliance. Specific activities were developed for each target table, following a well-defined workflow. The table population processes have been documented and are replicable, allowing reuse and monitoring of the ETL steps.

In summary, the developed ETL architecture comprehensively met the proposed objective, enabling the efficient extraction, transformation, and loading of data from the source database to the target database. Implementing the mechanisms for lineage control, erroneous record handling, metadata analysis, and data quality assurance contributed to the data's reliability and consistency. Thus, it can be stated that the goal of developing a robust and efficient ETL architecture was successfully achieved.

The objective (G4) of evaluating the success of the ETL process and the completeness of the target database has been achieved. The software tools used, including SQL Server 2019, SQL Server Management Studio 19, Visual Studio 2019, and SQL Server Integration Services, played a crucial role in efficiently storing, managing, and integrating data throughout the process. The system implementation involved creating a project in Visual Studio with 32 packages to extract, transform, and load data into the target table. The ETL flow followed a logical sequence, including table cleaning, lineage key generation, data processing and cleaning and lineage table updates. A demonstration of the system showcased the ETL flow used to populate the *Cardiac Baseline Assessment* table, highlighting the various steps involved. The evaluation of results was based on three metrics: percentage of populated columns per target table, overall percentage of populated columns, and percentage of populated records per column in each target table. The results indicated that most target tables had a high percentage of populated columns, demonstrating the effectiveness of the ETL process. However, the *Cardiac Baseline Assessment* table had a lower percentage of populated columns, and some specific columns in other tables were also missing data. Nevertheless, the populated information was considered satisfactory for analysis and achieving the project objectives.

Overall, the successful implementation of the system, demonstration of the ETL flow, and evaluation of results confirmed the achievement of the objective G4. The chosen software tools effectively facilitated data storage, workflow execution, and data quality treatment and cleaning. The evaluation metrics provided valuable insights into the performance and effectiveness of the ETL process, highlighting areas of success and areas that may require further attention.

5.2 Future Work

This thesis has established a solid foundation regarding integrating health data related to cardiovascular disease and COVID-19, paving the way for several opportunities for future work. Advancements in these areas will promote more robust research, better healthcare, and a greater understanding of overall health.

An important direction for future work is the improvement of the data pipeline developed in this thesis. This will involve continued improvements in performance, data quality controls, real-time monitoring mechanisms and the creation of transformation and loading algorithms. The evolution of the pipeline will ensure that it meets the changing needs of healthcare systems and research.

In addition, future work may focus on expanding the data source. This includes new patient data, genetic data, and medical device data. Incorporating multiple data sources will enrich integration and analysis, providing a more comprehensive view of patient health. This may require collaboration with other healthcare institutions and consideration of interoperability issues and data standards.

Another promising direction is the application of advanced data analysis techniques. Future work could explore the use of machine learning algorithms and artificial intelligence (notably Text Mining techniques) to identify patterns, predict risks, personalize treatments, and support clinical decision-making. Advanced data analytics will enable deeper understanding of patient health and significant improvements in healthcare and research.

Integrating the database developed in this thesis with existing healthcare systems is also an important future work area. This will facilitate broader access to health data, enabling collaborative research, monitoring of health indicators, and implementation of evidence-based interventions. Interoperability with electronic medical records, hospital information systems, and other relevant systems will be essential in this process.

Finally, future work should continue to address the ethical and privacy issues associated with Health Data Integration. This includes developing clear policies and guidelines to ensure informed patient consent, the anonymity of sensitive data, and compliance with privacy regulations. Collaboration with ethics and privacy experts will be critical to ensure adequate data protection and compliance with ethical standards.

The future work integrating cardiovascular disease-related health data and COVID-19 presents significant potential. Improving the data pipeline, expanding the data source, advanced data analysis, integration with other healthcare systems, and consideration of ethical and privacy issues are key aspects that will contribute to more advanced research and better cardiovascular healthcare.

In conclusion, this thesis represents a significant step towards a healthier and safer future. Focusing on integrating healthcare data related to cardiovascular disease and COVID-19, this work intends to uncover valuable insights from this information. Behind the data lies a mix of hope and despair, progress, and challenges – touching the lives of many individuals, families, and communities. Through developing a robust and reliable ETL pipeline, this thesis aims to give a voice and significance to each person represented in the data. By conducting a thorough analysis, ensuring quality, and safeguarding patient's data privacy, the goal is to advance scientific knowledge and enhance the well-being of individuals. May every section of this thesis emphasize the importance of this information and encourage positive change in healthcare practices. Ultimately, this work can serve as a guiding light and an urgent call to action for prioritizing healthcare as an indispensable global concern.

REFERENCES

[1] Y. Guo *et al.*, "The application of artificial intelligence and data integration in COVID-19 studies: A scoping review," *J. Am. Med. Inform. Assoc.*, vol. 28, no. 9, pp. 2050–2067, 2021, doi: 10.1093/jamia/ocab098.

[2] L. A. Lenert *et al.*, "Automated production of research data marts from a canonical fast healthcare interoperability resource data repository: Applications to COVID-19 research," *J. Am. Med. Inform. Assoc.*, vol. 28, no. 8, pp. 1605–1611, 2021, doi: 10.1093/jamia/ocab108.

[3] H. Dhayne, R. Kilany, R. Haque, and Y. Taher, "EMR2vec: Bridging the gap between patient data and clinical trial," *Comput. Ind. Eng.*, vol. 156, Jun. 2021, doi: 10.1016/j.cie.2021.107236.

[4] A. Sheikhtaheri, S. M. Tabatabaee Jabali, E. Bitaraf, A. TehraniYazdi, and A. Kabir, "A near real-time electronic health record-based COVID-19 surveillance system: An experience from a developing country," *Health Inf. Manag. J.*, 2022, doi: 10.1177/18333583221104213.

[5] J. L. Raisaro *et al.*, "SCOR: A secure international informatics infrastructure to investigate COVID-19," *J. Am. Med. Inform. Assoc.*, vol. 27, no. 11, pp. 1721–1726, 2020, doi: 10.1093/jamia/ocaa172.

[6] A. Dagliati, A. Malovini, V. Tibollo, and R. Bellazzi, "Health informatics and EHR to support clinical research in the COVID-19 pandemic: An overview," *Brief. Bioinform.*, vol. 22, no. 2, pp. 812–822, 2021, doi: 10.1093/bib/bbaa418.

[7] "Why is health data important? — Data Saves Lives." https://datasaveslives.eu/why-is-health-data-important (accessed Mar. 06, 2023).

[8] T. Sarwar *et al.*, "The Secondary Use of Electronic Health Records for Data Mining: Data Characteristics and Challenges," *ACM Comput. Surv.*, vol. 55, no. 2, 2023, doi: 10.1145/3490234.

[9] "RBC Capital Markets | Navigating the Changing Face of Healthcare Episode." https://www.rbccm.com/en/gib/healthcare/story.page (accessed Mar. 01, 2023).

[10] "Importance of Data Collection in Healthcare," *ForeSee Medical*. https://www.foreseemed.com/importance-of-data-collection-in-healthcare (accessed Mar. 06, 2023).

[11] S. Juumta, "Where Healthcare's Big Data Actually Comes From," *Emerj Artificial Intelligence Research*, Nov. 22, 2019. https://emerj.com/ai-sector-overviews/where-healthcares-big-data-actually-comes-from/ (accessed Feb. 21, 2023).

[12] "mgi-artificial-intelligence-discussion-paper.pdf." Accessed: Mar. 01, 2023. [Online]. Available: https://www.mckinsey.com/~/media/mckinsey/industries/advanced%20electronics/our%20insights/how%20artificial%20intelligence%20can%20deliver%20real%20value%20to%20companies/mgi-artificial-intelligence-discussion-paper.ashx

[13] "Promoting an overdue digital transformation in healthcare | McKinsey." https://www.mckinsey.com/industries/healthcare/our-insights/promoting-an-overdue-digital-transformation-in-healthcare (accessed Mar. 01, 2023).

[14] M.-T. Chen and T.-H. Lin, “A provable and secure patient electronic health record fair exchange scheme for health information systems,” *Appl. Sci. Switz.*, vol. 11, no. 5, 2021, doi: 10.3390/app11052401.

[15] F. Khennou, N. El Houda Chaoui, and Y. I. Khamlichi, “A migration methodology from legacy to new electronic health record based OpenEHR,” *Int. J. E-Health Med. Commun.*, vol. 10, no. 1, pp. 55–75, 2019, doi: 10.4018/IJEHMC.2019010104.

[16] J. A. Aunger, R. Millar, A. M. Rafferty, and R. Mannion, “Collaboration over competition? Regulatory reform and inter-organisational relations in the NHS amidst the COVID-19 pandemic: a qualitative study,” *BMC Health Serv. Res.*, vol. 22, no. 1, p. 640, May 2022, doi: 10.1186/s12913-022-08059-2.

[17] “EUSCIENCEHUBNEWS - Ireland is the country with the highest cancer incidence in the EU.” https://ec.europa.eu/newsroom/eusciencehubnews/items/684847 (accessed Mar. 02, 2023).

[18] “2004 - The Data Warehouse ETL Toolkit (Ralph Kimball).pdf.” Accessed: Mar. 02, 2023. [Online]. Available: https://ia800206.us.archive.org/15/items/2004TheDataWarehouseETLToolkitRalphKimball/2004%20-%20The%20Data%20Warehouse%20ETL%20Toolkit%20%28Ralph%20Kimball%29.pdf

[19] R. Fagin, “Inverting Schema Mappings,” *ACM Trans Database Syst*, vol. 32, no. 4, pp. 25-es, Nov. 2007, doi: 10.1145/1292609.1292615.

[20] D. Zhou *et al.*, “Multiview Incomplete Knowledge Graph Integration with application to cross-institutional EHR data harmonization,” *J. Biomed. Inform.*, vol. 133, 2022, doi: 10.1016/j.jbi.2022.104147.

[21] M. T. F. Abrahão, M. R. C. Nobre, and M. A. Gutierrez, “A method for cohort selection of cardiovascular disease records from an electronic health record system,” *Int. J. Med. Inf.*, vol. 102, pp. 138–149, 2017, doi: 10.1016/j.ijmedinf.2017.03.015.

[22] “Capacity Registry – Cardiac complicAtions in Patients with SARS Corona vIrus 2 regisTrY.” https://capacity-covid.eu/ (accessed Mar. 02, 2023).

[23] L. M. Fleuren *et al.*, “The Dutch Data Warehouse, a multicenter and full-admission electronic health records database for critically ill COVID-19 patients,” *Crit. Care*, vol. 25, no. 1, 2021, doi: 10.1186/s13054-021-03733-z.

[24] J. J. Reeves *et al.*, “Bringing student health and Well-Being onto a health system EHR: the benefits of integration in the COVID-19 era,” *J. Am. Coll. Health*, vol. 70, no. 7, pp. 1968–1974, 2022, doi: 10.1080/07448481.2020.1843468.

[25] M. Pedrera-Jimenez *et al.*, “TransformEHRs: a flexible methodology for building transparent ETL processes for EHR reuse,” *METHODS Inf. Med.*, vol. 61, pp. E89–E102, Dec. 2022, doi: 10.1055/s-0042-1757763.

[26] Y. Yu *et al.*, “Developing an ETL tool for converting the PCORnet CDM into the OMOP CDM to facilitate the COVID-19 data integration,” *J. Biomed. Inform.*, vol. 127, 2022, doi: 10.1016/j.jbi.2022.104002.

[27] T. R. Campion, E. T. Sholle, J. Pathak, S. B. Johnson, J. P. Leonard, and C. L. Cole, “An architecture for research computing in health to support clinical and translational investigators with electronic patient data,” *J. Am. Med. Inform. Assoc.*, vol. 29, no. 4, pp. 677–685, 2022, doi: 10.1093/jamia/ocab266.

[28] E. Bacon *et al.*, “Developing a Regional Distributed Data Network for Surveillance of Chronic Health Conditions: The Colorado Health Observation Regional Data Service,” *J. PUBLIC Health Manag. Pract.*, vol. 25, no. 5, pp. 498–507, Sep. 2019, doi: 10.1097/PHH.0000000000000810.

[29] E. Ter Avest *et al.*, “Cohort profile of Acutelines: A large data/biobank of acute and emergency medicine,” *BMJ Open*, vol. 11, no. 7, 2021, doi: 10.1136/bmjopen-2020-047349.

[30] K. Häyrinen, K. Saranto, and P. Nykänen, “Definition, structure, content, use and impacts of electronic health records: a review of the research literature.,” *Int. J. Med. Inf.*, vol. 77, no. 5, pp. 291–304, May 2008, doi: 10.1016/j.ijmedinf.2007.09.001.

APPENDICES

Appendix A - Source-to-Target Mappings template example

Table Name: Inclusion Criteria
Description: Allows for checking whether a given patient meets the requirements to be considered in the study
Number of variables: 10
Number of records: 815
SSIS Package: 008_INITIAL_LOAD_InclusionCriteria

TARGET	ETL	SOURCE				
Column name	ETL rules	Database. Schema	Table	Column name	Data type	Description
dsstdat	dsstdat = source.COVID_DATE	capacity_db. Integration	ALL_COVID_ PATIENTS_ADMISSIONS	COVID_DATE	datetime	Date of COVID-19 confirmed
sitename_ nhr	sitename_nhr = 918	N/A	N/A	N/A	N/A	N/A
country	country = (SELECT CPE.COUNTRY_EN_COD FROM capacity_db.Integration.ALL_COVID_PATIENTS_ADMISSIONS A INNER JOIN aimhealth_db.dbo.ADMISSIONS AA ON AA.ADMISSIONDBOID = A.ADMISSIONDBOID INNER JOIN aimhealth_db.dbo.COUNTRIES C ON AA.COUNTRYDBOID = C.COUNTRYDBOID INNER JOIN capacity_db.Integration.COUNTRIES_PT_EN CPE ON CPE. COUNTRY_PT = C.COUNTRYDESC)	capacity_db. Integration	COUNTRIES_PT_EN	COUNTRY_EN_COD	int	Country(translated to English) code
oth_country	**Step 1:** SELECT CPE. COUNTRY_PT FROM capacity_db.Integration.ALL_COVID_PATIENTS_ADMISSIONS A INNER JOIN aimhealth_db.dbo.ADMISSIONS AA ON AA.ADMISSIONDBOID = A.ADMISSIONDBOID INNER JOIN aimhealth_db.dbo.COUNTRIES C ON A.COUNTRYDBOID = C.COUNTRYDBOID INNER JOIN capacity_db.Integration.COUNTRIES_PT_EN CPE ON CPE. COUNTRY_PT = C.COUNTRYDESC **Step 2:** oth_country = (country_pt == "Desconhecido" ? "Unknown" : country_pt == "Porto Rico" ? "Puerto Rico" :a country_pt == "Bermuda" ? "Bermuda" : country_pt == "Outro" ? "Unknown" : country_pt == "Macau" ? "Macau" : country_pt == "NULL" ? NULL(DT_WSTR,50) : NULL(DT_WSTR,50))	capacity_db. Integration	COUNTRIES_PT_EN	COUNTRY_PT	nvarchar (100)	Name of the country, in Portuguese
corona_ ieorres	corona_ieorres = (!ISNULL(COVID_DATE) ? "1" : "0")	capacity_db. Integration	ALL_COVID_ PATIENTS_ADMISSIONS	COVID_DATE	datetime	Date of COVID-19 confirmed
sympt _epi_physical	sympt_epi_physical = 3	N/A	N/A	N/A	N/A	N/A
sympt _epi_healthfac	sympt_epi_healthfac = 3	N/A	N/A	N/A	N/A	N/A
sympt _epi_lab	sympt_epi_lab = 3	N/A	N/A	N/A	N/A	N/A
inclusion_ complete	inclusion_complete=(ISNULL(dsstdat) || ISNULL(sitename_nhr) || ISNULL(country) || ISNULL(corona_ieorres) || ISNULL(symptoms_epi_physical) || ISNULL(symptoms_epi_healthfac) || ISNULL(symptoms_epi_lab) ? "0" : "2")	N/A	N/A	N/A	N/A	N/A
subjid	subjid = (SELECT P.subjid FROM capacity_db.Integration.ALL_COVID_PATIENTS_ADMISSIONS A INNER JOIN capacity_db.Integration.PATIENT_ADMISSION_MAPPINGS P ON A.ADMISSIONDBOID = P.ADMISSIONDBOID)	capacity_db. Integration	PATIENT_ADMISSION_ MAPPINGS	subjid	nvarchar (25)	Patient Admission Identifier