TU WIEN Informatics

Generische ETL-Codebasis zur Gesundheitsdatentransformation aus dem EAV-Modell in das OMOP Common Data Model

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Heike Düsseldorf, B.Sc.
Matrikelnummer 12045323

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Projektass. Dipl.-Ing. Dr.techn. Rene Baranyi, Bakk.techn.
Mitwirkung: Univ.-Prof. Dipl.-Ing. Dr. Georg Duftschmid (MedUni Wien)
Univ.-Prof. Dr.med.univ. Elisabeth Presterl, MBA (MedUni Wien)
Dipl.-Ing. Christoph Aigner, Bakk. techn.
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig

Wien, 26. August 2025

Unterschrift Verfasserin | Unterschrift Betreuung

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Design of a generic ETL code base for transforming healthcare data from the EAV model to the OMOP Common Data Model

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Medical Informatics

by

Heike Düsseldorf, B.Sc.
Registration Number 12045323

to the Faculty of Informatics

at the TU Wien

Advisor: Projektass. Dipl.-Ing. Dr.techn. Rene Baranyi, Bakk.techn.
Assistance: Univ.-Prof. Dipl.-Ing. Dr. Georg Duftschmid (MedUni Wien)
Univ.-Prof. Dr.med.univ. Elisabeth Presterl, MBA (MedUni Wien)
Dipl.-Ing. Christoph Aigner, Bakk. techn.
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig

Vienna, 26th August, 2025

Signature Author | Signature Advisor

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Generische ETL-Codebasis zur Gesundheitsdatentransformation aus dem EAV-Modell in das OMOP Common Data Model

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Heike Düsseldorf, B.Sc.
Matrikelnummer 12045323

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Projektass. Dipl.-Ing. Dr.techn. Rene Baranyi, Bakk.techn.

Wien, 26. August 2025

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Heike Düsseldorf, B.Sc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 26. August 2025

Heike Düsseldorf

Danksagung

Ich möchte mich bei allen Personen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt, mir mit Rat und Tat zur Seite gestanden und mich über die Jahre hinweg auf meinem universitären Weg begleitet haben.

Mein besonderer Dank gilt Herrn Prof. Georg Duftschmid vom Institut für Medizinisches Informationsmanagement der Medizinischen Universität Wien. Seine Fachkenntnisse und sein Engagement waren von unschätzbarem Wert für die Entwicklung und erfolgreiche Umsetzung dieses Projekts. Vielen Dank für die fachlichen Diskussionen, die konstruktive Kritik, die hilfreichen Anmerkungen sowie die investierte Zeit. Danke auch an das gesamte Team des Institutes für Medizinisches Informationsmanagement der Medizinischen Universität Wien für die fachliche und organisatorische Unterstützung.

Ebenso danke ich Frau Prof. Elisabeth Presterl von der Universitätsklinik für Krankenhaushygiene und Infektionskontrolle der Medizinischen Universität Wien. Vielen Dank für die Vertiefung meines medizinischen Wissens, die hilfreichen Anmerkungen und die Unterstützung meiner Weiterentwicklung.

Mein Dank gilt darüber hinaus Herrn Dipl.-Ing. Christoph Aigner von der Forschungsgruppe Industrial Software an der Technischen Universität Wien. Vielen Dank für die Zeit, die hilfreichen Anmerkungen und die organisatorische Unterstützung. Ebenso danke ich dem gesamten Team der Forschungsgruppe Industrial Software an der Technischen Universität Wien sowie dessen Leiter, Herrn Prof. Thomas Grechenig, sowie Dr. Rene Baranyi und Dr. Mario Bernhart für die Möglichkeit und Unterstützung, meine Diplomarbeit verfassen zu können.

Herrn Dr. Thomas Wrba von den Research IT Services der IT-Services & Strategisches Informationsmanagement der Medizinischen Universität Wien danke ich herzlich für die fachlichen und technischen Diskussionen sowie für die Unterstützung, die mir nicht nur die erfolgreiche Fertigstellung dieser Arbeit, sondern auch meine persönliche Weiterentwicklung ermöglicht. Außerdem danke ich Herrn Christoph Wild, dem Leiter der IT-Services & Strategisches Informationsmanagement der Medizinischen Universität Wien, sowie der gesamten Abteilung für die Bereitstellung der Infrastruktur und die vielfältige Unterstützung, ohne die diese Arbeit nicht möglich gewesen wäre.

Mein tief empfundener Dank gilt auch meiner Familie, meinen Studienkolleg*innen und Freund*innen, die mich während der letzten Jahre begleitet und unterstützt haben. Eure

Ermutigung und Hilfe waren für mich eine konstante Quelle an Motivation, Inspiration und Zuversicht. Danke, dass ihr mir ermöglicht und geholfen habt, da zu sein, wo ich jetzt bin. Abschließend möchte ich mich bei meinem Partner bedanken. Danke für deine Geduld während der Zeit des Schreibens und für deine ständige Unterstützung. Dein Vertrauen in meine Fähigkeiten hat mir die Kraft gegeben, Herausforderungen zu meistern und meine Ziele zu erreichen.

Kurzfassung

Die Standardisierung von Gesundheitsdaten ist entscheidend, um multizentrische Forschung zu ermöglichen, die klinische Entscheidungsfindung zu verbessern und die Reproduzierbarkeit datenbasierter Erkenntnisse sicherzustellen. Gesundheitsinstitutionen speichern ihre Daten jedoch oft in heterogenen Formaten und institutionsspezifischen Datenmodellen. Ein Beispiel ist das an der Medizinische Universität Wien (MedUni Wien) eingesetzte flexible Entity-Attribute-Value (EAV)-Modell, das auf die Bedürfnisse der Institution zugeschnitten ist. Solche individuellen Modelle erschweren die semantische Interoperabilität und die Integration klinischer Daten.

Zur Bewältigung dieser Herausforderung stellt diese Arbeit einen generischen Extract, Transform, Load (ETL)-Prozess vor, der Gesundheitsdaten aus dem EAV-Modell der MedUni Wien in das Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) überführt. Ziel ist ein wiederverwendbares und erweiterbares ETL-Framework, das sich an verschiedene Datensätze und Anwendungsfälle anpassen lässt.

Die Entwicklung erfolgte iterativ und feedbackbasiert auf Basis domänenspezifischer Anforderungen und verband konzeptionelle Modellierung, Implementierung und Evaluation.

Der Prototyp wurde anhand von zwei Evaluationsszenarien mit realen Daten der MedUni Wien validiert: automatisierte Überwachung von hospital-onset bacteremia and fungemia (HOB) sowie breast cancer benchmarking (BCB). In beiden Fällen gelang die erfolgreiche Transformation heterogener Quelldaten in das OMOP CDM, womit die Anpassungsfähigkeit an unterschiedliche klinische Domänen belegt wurde.

Die Evaluation zeigt, dass das System strukturelle Variabilität bewältigen und in verschiedenen Anwendungsfällen eingesetzt werden kann. Während im HOB-Fall effiziente Laufzeiten erreicht wurden, führten die umfangreicheren BCB-Daten zu längeren Verarbeitungszeiten und verdeutlichten Optimierungspotenzial bei großen Datenmengen. Beide Szenarien bestätigten die korrekte Transformation und die Erweiterbarkeit des Frameworks.

Die Ergebnisse belegen, dass eine flexible und strukturierte ETL-Strategie die zuverlässige Transformation EAV-basierter Gesundheitsdaten in das OMOP CDM ermöglicht und zu Standardisierung und Interoperabilität in klinischen Datenumgebungen beitragen kann.

Keywords: *Gesundheitsdatentransformation, OMOP CDM, Entity-Attribute-Value Modell, ETL-Prozess, Datenharmonisierung, Semantische Interoperabilität*

Abstract

The standardization of healthcare data is crucial for enabling multicenter research, enhancing clinical decision-making, and ensuring reproducibility of data-driven insights. However, healthcare institutions often store their data in heterogeneous formats and institution-specific models. For example, the flexible Entity-Attribute-Value (EAV) model used at Medical University of Vienna (MedUni Vienna) is tailored to the institution's needs. Such individualized data models limit semantic interoperability and complicate clinical data integration.

To address this challenge and realize the benefits of standardized data, this thesis presents a generic Extract, Transform, Load (ETL) process transforming healthcare data from the EAV model used at the MedUni Vienna into the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM). The objective is a reusable, extensible, and high-quality ETL framework adaptable to various datasets and use cases.

The system was designed based on domain-specific requirements and was developed through an iterative, feedback-driven process that integrates conceptual modeling, technical implementation, and evaluation.

The prototype was validated through two evaluation scenarios using real-world datasets from the MedUni Vienna: automated surveillance of hospital-onset bacteremia and fungemia (HOB) and breast cancer benchmarking (BCB) across European hospitals. In both scenarios, the system successfully transformed complex and heterogeneous source data into the OMOP CDM, demonstrating adaptability to different clinical domains.

The evaluation highlights the system's ability to manage structural variability and apply semantic mappings across use cases. The HOB scenario demonstrated efficient runtimes. In contrast, the BCB scenario involved large-scale data, resulting in longer runtimes and highlighting the need for performance optimization in high-volume settings. Nevertheless, both evaluation scenarios confirmed the correctness of the transformation, and the reuse of shared components validated the framework's reusability and adaptability.

These results demonstrate that a flexible yet structured ETL strategy can enable the reliable transformation of EAV-based healthcare data into the OMOP CDM, contributing to broader standardization and interoperability in clinical data environments.

Keywords: *Healthcare Data Transformation, OMOP CDM, Entity-Attribute-Value Model, ETL Process, Data Harmonization, Semantic Interoperability*

Contents

CHAPTER 1

Introduction

This thesis focuses on the development of a generic Extract, Transform, Load (ETL) codebase for transforming healthcare data from the Entity-Attribute-Value (EAV) model used at the Medical University of Vienna (MedUni Vienna) into the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM). It specifically tackles the challenge of converting EAV-modeled data to the OMOP CDM, a critical process for enabling comparable multicenter studies [1].

For instance, in a multicenter study investigating breast cancer treatment outcomes, participating hospitals may store patient records in institution-specific formats using structurally diverse systems. One site might use an EAV-modeled database for flexibility, while another may rely on a traditional horizontal schema. To enable joint analysis of treatment effectiveness or survival metrics across institutions, these heterogeneous datasets must first be harmonized into a common data model such as the OMOP CDM. This transformation ensures consistent representation, facilitating scalable and reproducible analyses across diverse populations and care settings [1].

The research falls within the health informatics domain, particularly addressing the harmonization of heterogeneous data structures in the healthcare sector to achieve semantic interoperability.

This introduction provides a brief overview of the topic of this thesis. It describes the research problem, outlines the motivation and objectives of the study, and provides a short overview of the methodology. Finally, a short overview of the thesis's structure and a breakdown of its chapters will be provided.

1.1 Problem Statement

In healthcare research, the utilization of CDMs, such as the OMOP CDM, aims to standardize data across diverse datasets. This standardization is essential for data

harmonization and semantic interoperability, which can significantly improve healthcare research and decision-making by enabling more efficient and accurate analysis and supporting multicenter studies [1]. Semantic interoperability refers to the ability of disparate systems to exchange data in a way that preserves the meaning of the information and ensures the data is understood consistently across different platforms, organizations, and contexts [2]. Especially in multicenter studies, where data is collected from different institutions, it is essential to standardize healthcare information both structurally and semantically to enable meaningful comparisons and federated analytics [3]. However, healthcare institutions often store data in heterogeneous formats and institution-specific models, which limits semantic interoperability and complicates integration [4].

The data model used in many electronic health record (EHR) systems is the EAV model [5], as it is flexible in adapting to structural differences in incoming data and it allows easy schema updates. It also enables compact storage of sparse data [6]. While this flexibility is beneficial for local requirements, it results in highly heterogeneous implementations [4]. To achieve semantic interoperability and facilitate standardized research, the EAV-modeled data must be transformed into a CDM, like the OMOP CDM.

The OMOP CDM, developed by the Observational Health Data Sciences and Informatics (OHDSI) community, is one of the most widely adopted CDMs on a global scale [7]. It provides a structured framework for integrating data from various sources, optimized explicitly for observational research and large-scale data analytics. By standardizing the data on a structural and semantic level, the OMOP CDM enables semantic interoperability and supports multicenter studies, allowing researchers to perform comparative analyses across patient populations and institutions [1], [8].

Complementing the model, OHDSI provides open-source tools that facilitate diverse data-analytic applications on observational patient-level data. These tools standardize analytical processes across various applications. Researchers can use predefined templates to conduct analyses without building them each time from scratch, thereby improving reproducibility, transparency, and efficiency [9]. By using the OMOP CDM, researchers can overcome the inherent challenges posed by the heterogeneous nature of healthcare data, thereby enabling more robust and reproducible research outcomes [1].

Transforming data from an EAV model to the OMOP CDM can present unique challenges. For instance, transforming data from an EAV model to a horizontal model, such as the OMOP CDM, often involves pivoting the data, which can be a complex operation [10]. Additionally, existing ETL processes are often tailored to data structures or requirements that differ from the EAV model [11], [12], [13]. They are adapted to specific needs, which limits their adaptability to new data structures, such as the EAV model. Therefore, the primary objective of this thesis is to develop a generic ETL code base that transforms healthcare data from the EAV model used at the MedUni Vienna into the OMOP CDM and can be reused across different use cases with minimal adaptations.

1.2 Motivation

Effective and consistent transformation of healthcare data into a unified data model, such as the OMOP CDM, is crucial for advancing research, enhancing clinical practice, and supporting evidence-based decision-making. As the volume and complexity of healthcare data continue to grow, the need for reliable methods to integrate data from multiple sources becomes increasingly urgent [1].

In practice, data is often stored in institution-specific formats, such as the flexible EAV model, which is commonly customized to meet local needs [6]. This variability complicates large-scale, multicenter studies, which require comparable and semantically consistent data [1].

CDMs, such as the OMOP CDM, address these challenges by offering a standardized framework for harmonizing disparate data sources [1]. Existing ETL processes are frequently tailored to specific schemas, limiting reusability and efficiency when applied to EAV-modeled data [10], [14]. This lack of generalizability leads to redundant development efforts and reduces the efficiency of data integration pipelines.

This thesis addresses these limitations by developing a generic ETL framework that supports reusable and adaptable data transformations from the EAV model of the research database at the MedUni Vienna to the OMOP CDM. By providing a flexible and modular solution, the framework streamlines the integration process, reduces the technical burden of preparing data for analysis, and minimizes redundant development efforts. This approach ensures that data can be efficiently transformed for standardized, reproducible analyses across multiple use cases and research projects. It enables efficient transformation of heterogeneous data and supports scalable, reproducible research.

The work is grounded in principles of data harmonization, semantic interoperability, and ETL process design. A thorough understanding of the OMOP CDM's structure and requirements ensures that transformed data remain analytically valid, semantically aligned, and technically interoperable, enabling meaningful comparisons across institutions and datasets.

By addressing these challenges, this thesis contributes to the broader goal of advancing healthcare data integration. It supports collaborative, multicenter research by enabling consistent data processing and harmonization, facilitating robust data-driven insights, and improving the overall quality, accessibility, and reproducibility of healthcare analytics. Ultimately, the framework aims to empower researchers and healthcare institutions to leverage complex, heterogeneous data more effectively, accelerating the translation of data into actionable knowledge for patient care and clinical decision-making.

1.3 Expected Results

This thesis aims to develop a generic ETL code base for transforming healthcare data from the EAV model of the MedUni Vienna into the OMOP CDM. The thesis will cover

the requirements, design, implementation, and evaluation of the developed prototype.

This research addresses a critical need in health informatics for standardized, efficient, and scalable data integration processes, essential for enabling comprehensive and comparable analyses across diverse healthcare datasets. By supporting the transformation of heterogeneous data structures into a standardized format, this thesis contributes to the broader goal of achieving semantic interoperability in healthcare data. Specifically, it tackles the current gap in reusable ETL implementations for transforming MedUni Vienna's EAV-modeled data into the OMOP CDM, a barrier to effective data integration and analysis.

The following research questions guide the work, each targeting a specific aspect of the analysis, development, and evaluation of the developed prototype:

RQ1 *ETL requirements*: **What are the specific requirements for the ETL process to ensure the successful transformation of healthcare data from the EAV model into the OMOP CDM?**

This research question is addressed through a literature review. The goal is to identify functional and non-functional requirements that inform the design of a generic, reusable ETL system.

RQ2 *ETL process design*: **How can an effective and generic transformation of healthcare data from the EAV model into the OMOP CDM be achieved?**

This question focuses on developing a conceptual architecture for the ETL process and implementing it as a prototype using evolutionary prototyping.

RQ3 *Use of the generic ETL code base system*: **To what extent can the developed ETL process for the transformation of healthcare data from the EAV model into the OMOP CDM be extended or adapted to specific use cases?**

This question is explored through two evaluation scenarios involving real-world EAV-modeled datasets. The adaptability of the prototype to different contexts and data characteristics is evaluated, demonstrating its potential for broader applicability.

RQ4 *Evaluation of the ETL process*: **How does the developed generic ETL code base perform regarding data quality and adaptability?**

This research question is investigated by evaluating the implemented prototype and its adaptation to the evaluation scenarios against the defined functional and non-functional requirements.

To answer these research questions, the methodology of this thesis is structured in four phases: Analysis, Design, Implementation, and Evaluation. In the Analysis Phase, a literature review is conducted to identify key requirements and quality characteristics for the ETL process. These insights inform the Design Phase, where a conceptual model

for the ETL pipeline is developed and continuously refined. The Implementation Phase overlaps with the Design Phase, as the ETL prototype is literally developed through iterative refinement cycles using evolutionary prototyping. Finally, in the Evaluation Phase, two evaluation scenarios are performed to examine the prototype's adaptability to different use cases. Additionally, the prototype and its adaptation to the evaluation scenarios are assessed in terms of its functionality, performance, and compliance with the defined requirements. A more detailed description of the methodology is provided in Chapter 4.

1.4 Structure

Chapter 2 lays the groundwork by introducing the key concepts and models essential to this research. It covers the EAV data model and the Research Documentation & Analysis platform (RDA platform) of the MedUni Vienna. The chapter emphasizes the importance of semantic interoperability and introduces the OMOP CDM and ETL processes. It describes JSON Schema as a tool for ensuring data consistency across systems, establishing a solid technical foundation for the approach used in this thesis.

Chapter 3 reviews relevant literature and prior research on designing and implementing an ETL process from the EAV data model to the OMOP CDM. This chapter focuses particularly on existing ETL processes for the OMOP CDM or from the EAV model. It also examines conceptual modeling of ETL processes, quality characteristics of ETL processes, ETL tools, and gaps in current approaches, thereby positioning this research within the broader field of health informatics.

Chapter 4 details the methodology used in this thesis, which is divided into four phases: Analysis, Design, Implementation, and Evaluation. Each phase is described in detail, explaining the approach for developing the ETL code base system and how each step contributes to the overall goal of data transformation and integration.

In Chapter 5, the research outcomes are presented. The results include the requirements for the ETL process, a detailed implementation concept, and the prototypical implementation of the ETL process.

Chapter 6 presents two evaluation scenarios that apply the developed prototype to real-world healthcare data: one focused on automated surveillance of hospital-onset bacteremia and fungemia (HOB) and the other on breast cancer benchmarking (BCB). These evaluation scenarios highlight the practical application of the ETL system, providing insights into its adaptability, the transformation results, and the challenges encountered when applying the generic ETL process to specific use cases. This chapter also evaluates the extent to which the implemented prototype fulfills the defined functional and non-functional requirements. Each requirement is assessed individually, with clear justifications based on implementation details and observed system behavior.

In Chapter 7, the findings are reflected upon, answering the research questions and discussing the limitations of the ETL system, the challenges faced during its development

and implementation, and the broader implications for the field of health informatics.

Finally, Chapter 8 summarizes the key findings and contributions of the thesis, emphasizing the impact on healthcare data integration. The chapter proposes directions for future research and development, identifying areas for further investigation and refinement, and offering suggestions for improving the system and methodology.

CHAPTER 2

Fundamentals

This chapter outlines the foundational concepts essential for the development of a generic ETL code base system aimed at transforming healthcare data from the EAV model to the OMOP CDM.

The chapter begins by examining the EAV data model. This model is a widely used data structure in healthcare databases due to its flexibility in managing complex, variable, and heterogeneous data structures. The EAV model effectively manages data with irregular or evolving attributes [6]. However, its flexibility comes with significant challenges regarding data integration and exchange. These challenges necessitate the use of advanced transformation techniques [10].

Next, the RDA platform of the MedUni Vienna is discussed. The RDA platform is a central repository for clinical and research data, including routine data, laboratory reports, and surgical protocols [15]. This data is stored using the EAV data model and forms the core data sources that will be transformed into the OMOP CDM. The RDA platform ensures that research datasets are accessible, consistent, and ready for analysis, providing a structured environment for managing healthcare data [16].

A critical challenge in this transformation process is achieving semantic interoperability, ensuring that data from different systems can be exchanged and correctly interpreted. Semantic interoperability enables healthcare systems to share data while preserving its meaning, supporting reliable and consistent analyses across various platforms [17].

The chapter then introduces the OMOP CDM, a standardized data model that facilitates the systematic analysis of diverse healthcare datasets. By offering a consistent and structured approach, the OMOP CDM enables semantic interoperability, supports multicenter studies, and ensures the reliability and reproducibility of research findings [1].

After that, ETL processes, which are fundamental to data integration and transformation, are explained. They involve extracting data from multiple sources, transforming it into a

consistent format, and loading it into a target database or data warehouse [18]. An ETL process is a crucial step in ensuring that data is accurately transformed into the OMOP CDM format [1], [19].

Finally, JSON Schema is introduced as a standardized format for specifying the structure, content, and constraints of JSON data, enabling validation and consistent data exchange between systems [20].

Overall, this chapter aims to establish a thorough understanding of these foundational concepts, providing the necessary context for the subsequent development and implementation of the ETL system discussed in the subsequent chapters of this thesis.

2.1 Entity-Attribute-Value Data Model

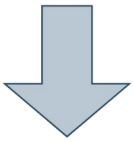
The EAV data model is a generic data model used in many EHR systems [5] due to its flexibility in adapting to structural differences in incoming data, easy schema updates, and compact storage of sparse data. These characteristics make it particularly well-suited for managing the complex and dynamic nature of healthcare data [6].

The EAV data model organizes data into three primary tables: entities, attributes, and values. The entities table lists the subjects of the data, such as patients. The attributes table lists the different properties that can be recorded about each entity, such as physical examinations or lab results. The values table contains the actual data points, linking each value to a specific entity and attribute [6].

Healthcare data is often sparse because not every patient undergoes all possible tests, diagnoses, or treatments. For instance, one patient may have numerous lab results and diagnoses, while another might have only a few recorded visits with minimal information. In a conventional, horizontal data model, where each attribute has its own column, this variability would result in many empty fields, leading to inefficient storage and a wide database schema [21]. The EAV model addresses this issue by storing data in a narrow, flexible format. Instead of having a wide table with many columns, most of which might be empty, the values table in the EAV model only stores existing information. The values table represents each data point as a separate row, with references to the relevant entries in the entities and attributes tables. This approach allows for more compact and efficient storage of sparse data [6].

One of the most significant advantages of the EAV model is its ability to accommodate easy schema updates. In traditional horizontal tables, adding a new property, such as a new lab test or clinical measurement, typically requires altering the existing schema to include new columns. This process can be time-consuming and complex, especially in large systems. In contrast, the EAV model simplifies this process. It is only necessary to insert a new row into the attributes table to add new data variables. The values table can then immediately start storing this new data without requiring changes to the existing schema. This flexibility makes the EAV model ideal for healthcare settings, where new tests, treatments, and other properties are frequently introduced [6].

Name	Albumin	Glucose	Creatin kinase	Hemoglobin	Temperature	Weight	Systolic
Tommie Glover	4.2		42	14	37.6		
Merritt Greene		135			36.4		125
Emery Haden			168	12		79	
Jocelyn Ross	3.1	86			38.9		138



Entity	Name	SSN
1	Tommie Glover	3063 191063
2	Merritt Greene	6209 271175
3	Emery Haden	3194 111203
4	Jocelyn Ross	9219 310872

Attribute	Name	Unit
1	Albumin	g/dl
2	Glucose	mg/dl
3	Creatin kinase	U/l
4	Hemoglobin	g/dl
5	Temperature	° C
6	Weight	kg
7	Systolic	mmHg

Entity	Attribute	Value
1	1	4.2
1	3	42
1	4	14
1	5	37.6
2	2	135
2	5	36.4
2	7	125
3	3	168
3	4	12
3	6	79
4	1	3.1
4	2	86
4	5	38.9
4	7	138

Figure 2.1: This figure is a simplification of possible sample EAV data. The top table shows a sample horizontal table, and the tables below depict the same data in the EAV data model.

In a conventional data model, patient data might be stored in a wide table with many columns representing different attributes. In contrast, the EAV model would store this data in a much narrower and more flexible format. The values table includes rows for each patient-attribute pair, with references to the relevant entries in the entities and attributes tables [6]. Figure 2.1 illustrates how data is represented in both a horizontal and the EAV data model, highlighting the compactness and flexibility of the EAV approach.

2.2 Research Documentation & Analysis Platform of the Medical University of Vienna

The RDA platform is a central component of the scientific infrastructure at the MedUni Vienna. It provides clinically and scientifically relevant data in a central database, which is continuously and automatically updated with routine data from the hospital information system (HIS) of the University Hospital Vienna (German: Allgemeines Krankenhaus der Stadt Wien) (AKH), known as AKH Informationsmanagement (AKIM). These data include laboratory results, surgical protocols, clinical reports, and other forms of clinical documentation. In addition, clinicians can enrich the platform with study-specific documentation or registry data, thereby combining routine clinical information with tailored research content and enhancing the scope of available analyses [15].

The RDA platform stores data in an Oracle relational database using an EAV data model. In this model, patients represent the entities about which information is recorded. The attributes define medical variables, which can be grouped and configured as fields on electronic forms. Data entry occurs by completing such forms, which are then stored as documents. Each document is linked to its underlying form and contains typed values (e.g., numeric, textual, or temporal), each referencing the corresponding variable. Both documents and their values are associated with the corresponding patient record. To promote consistency and standardization, form fields may be constrained to predefined value sets. Units can further be assigned to numerical values [15], [16].

This architecture allows the RDA to expand dynamically as new forms and variables are introduced, without requiring modifications to the underlying database schema. Because all information is stored in a uniform EAV representation, queries can be formulated in a generic, form-independent manner, spanning across projects and studies [15], [16].

2.3 Semantic Interoperability

Semantic interoperability is crucial for data integration in healthcare systems, particularly when working with heterogeneous data sources and diverse terminological frameworks [17]. It refers to the ability of disparate systems to exchange data in a way that ensures that the precise meaning of the information is preserved and consistently understood across diverse platforms, institutions, and contexts [22]. Achieving semantic interoperability enables organizations to share data without losing meaning, regardless of the system that generates the data. It allows the receiving system to automatically process the data, as its semantics are interpretable within that system. This is especially important in healthcare, where diverse records and clinical annotations must be consistently interpreted across different institutions to support patient care, decision-making, and research [17].

As defined in ISO/TR 20514, achieving semantic interoperability requires the use of a standardized set of domain-specific concept models and standardized terminologies. These components form the foundation for enabling systems not only to exchange data in a syntactically correct format but also to ensure that the meaning of the data

remains consistent and interpretable across different systems and healthcare environments. According to the standard, semantic interoperability ensures that exchanged data is interpretable and retains its intended meaning, regardless of the platform or organization receiving the data [2].

Establishing domain-specific concept models provides a shared understanding of the structure and semantics of the data within a given domain. By outlining the formal structure of the data and the rules governing its representation, these models ensure that the data is consistently encoded, facilitating accurate interpretation across different systems. Conceptual models form the foundation upon which healthcare data is structured, enabling interoperability even in diverse system implementations and local adaptations [2].

Equally important is the use of standardized terminologies. Well-established examples include Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT), International Classification of Diseases, 10th Revision (ICD-10), and Logical Observation Identifiers Names and Codes (LOINC). However, organizations can also define individual standardized terminologies to suit their specific use cases.

SNOMED CT is one of the most comprehensive and expressive clinical terminologies. It is multidimensional and structured hierarchically, allowing the representation of complex relationships between clinical concepts, such as symptoms, findings, procedures, and diseases [23].

ICD-10, developed by the World Health Organization (WHO), is a globally used classification system designed primarily for coding diagnoses in administrative and epidemiological contexts. It is particularly important for mortality statistics and public health reporting. Unlike SNOMED CT, ICD-10 is a code-value mapping that provides less semantic detail but is well-suited for statistical aggregation [24].

LOINC is a global standard for identifying laboratory and clinical observations. It is widely used in laboratory diagnostics to ensure consistent coding and interpretation of test results across institutions and systems [25].

These terminologies provide a unified naming system for the concepts represented in healthcare data, ensuring consistency across different systems. Standardized terminologies are essential for accurately interpreting the semantics of data. They eliminate ambiguity and enable information systems to assign precise meaning to data by linking it to well-defined, context-specific concepts. Without such standardized nomenclature, the risk of misinterpretation and loss of meaning increases, which can lead to errors in clinical decision-making, data analysis, and research [2].

2.4 OMOP Common Data Model

In contemporary healthcare, vast amounts of data are generated daily from various sources, including EHRs, claims data, clinical trials, laboratory results, and patient registries [26]. However, this data is often fragmented, heterogeneous, and stored in different formats,

creating significant challenges for researchers and clinicians in integrating and analyzing data across systems. The lack of standardization in medical data and terminology inconsistencies severely limit the potential for large-scale, comparative effectiveness research and public health monitoring [1].

Data heterogeneity is one of the primary barriers to adequate healthcare research. Healthcare information is distributed across various institutions, utilizing proprietary systems and differing coding standards. For example, diagnoses might be recorded using ICD-10 in one institution and SNOMED CT in another. This inconsistency complicates the aggregation of data from multiple sources, which is essential for multicenter studies and population health analysis [1].

In healthcare research, a CDM aims to standardize data across diverse datasets, optimizing the efficiency and accuracy of data analysis and interpretation. Hence, they are essential tools for data harmonization and semantic interoperability, which can significantly improve the quality and reproducibility of healthcare research [1]. Healthcare information needs to be standardized and harmonized on a structural and semantic level, especially when performing multicenter studies, to enable distributed network research and federated analytics. A CDM implements these requirements, enabling semantic interoperability [7]. Section 2.3 explains the requirements for semantic interoperability.

The OMOP CDM, developed by the OHDSI community, is one of the most widely used CDMs globally [7]. It is a patient-centric data model that provides a standardized framework for organizing and harmonizing heterogeneous healthcare data from diverse sources, supporting large-scale observational and multicenter research. By converting data into a consistent structure and utilizing standardized vocabularies, the OMOP CDM promotes semantic interoperability and comparability, enabling reproducible analyses and the generation of meaningful insights from real-world data [1].

A critical component of the OMOP CDM's functionality are the OHDSI standardized vocabularies. These vocabularies provide a unified reference ontology incorporating imported and newly created ontologies, including concepts and their relationships. By mapping various coding standards (e.g., ICD-10 for diagnoses, LOINC for laboratory measurements) to standard concepts within the OHDSI standardized vocabularies, healthcare data converted into the OMOP CDM maintains consistency across diverse datasets. This standardization enables researchers to perform scalable, uniform analyses across regions and healthcare settings [8].

The OMOP CDM is structured into standardized tables, each with predefined fields and relationships. These tables encompass various aspects of healthcare data, including patient demographics, clinical events, drug exposures, procedures, measurements, and healthcare provider information [1]. The current version of the OMOP CDM is CDM v5.4 [27]. A schematic representation of the OMOP CDM v5.4 is shown in Figure 2.2. Key tables in the OMOP CDM include [1], [27]:

- **Person**: Contains demographic information about each patient.

- **Observation Period**: Defines the periods during which data is collected for each patient.
- **Death**: Captures when and how a person died based on available clinical or administrative data.
- **Care site**: Lists institutional (physical or organizational) units where healthcare services are provided (e.g., offices, wards, hospitals, clinics, etc.).
- **Visit Occurrence**: Records details of healthcare encounters.
- **Visit detail**: Represents detailed parts of a visit, such as ward movements or claim lines, linked to a visit occurrence.
- **Condition Occurrence**: Captures diagnoses and medical conditions.
- **Drug Exposure**: Logs medications prescribed and administered to patients.
- **Procedure Occurrence**: Documents medical procedures performed.
- **Measurement**: Includes lab results and other clinical measurements.
- **Observation**: Contains clinical observations not covered by other tables.
- **Specimen**: Stores records of biological samples collected from persons.
- **Fact relationship**: Defines relationships between records across or within CDM tables (e.g., procedure–device, drug–condition)

Figure 2.3 shows how data is structured within the OMOP CDM by illustrating sample entries in three core tables: Person, Measurement, and Concept. This layout demonstrates how patient information, clinical measurements, and standardized concepts are structured and interrelated within the OMOP CDM.

The Person table forms the foundational layer, containing demographic information for each patient, such as Person ID, Gender, and Birth Date. The Person ID serves as the primary key for the Person table and as a foreign key in other related tables, allowing all records to link to an individual patient [27].

The Measurement table captures clinical and laboratory data related to each patient. Each record in this table includes a unique Measurement ID as the primary key. The Person ID functions as a foreign key, linking each measurement to the corresponding patient in the Person table. Additional fields in the Measurement table, such as Measurement Date, Value, and Unit, describe the specific measurement, providing context for clinical values like blood pressure, lab results, and other vital data points [27].

The Concept table standardizes terminology across datasets, ensuring the consistent use of clinical terms within the OMOP CDM. Each concept, represented by a Concept ID as the primary key, includes fields such as Concept Name and Domain to describe the term,

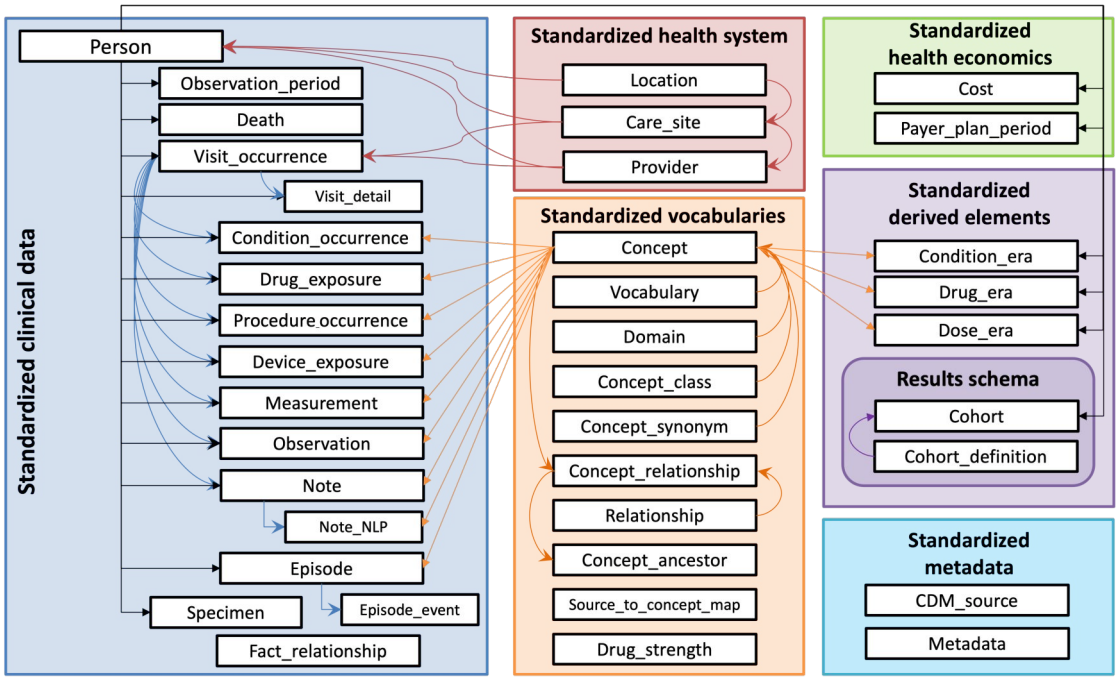


Figure 2.2: This figure is a schematic representation of the OMOP CDM v5.4 [27].

as well as the Vocabulary ID, which identifies the original coding system (e.g., SNOMED CT, LOINC) from which the concept originates. The Concept ID is used as a foreign key within the Measurement table, mapping each measurement to a specific clinical concept in the Concept table [27].

This structure facilitates interoperability and data consistency. Integrating the Concept table ensures that diverse data sources are harmonized under a unified vocabulary, allowing researchers to analyze clinical measurements reliably across different datasets [1], [27].

The OMOP CDM's ability to standardize heterogeneous healthcare data is one of its primary advantages. Researchers can perform comparative effectiveness studies, identify trends, and generate real-world evidence globally by converting disparate data sources into a common format. This standardization also enhances the statistical power and generalizability of research findings by facilitating data pooling across institutions [1].

Furthermore, the OMOP CDM supports various analytical tools and methods developed by the OHDSI community. These open-source tools include software for data visualization, cohort definition, and statistical analysis, all designed to work seamlessly with the OMOP CDM. Researchers can implement analyses by populating predefined templates, reducing the need to develop analyses from scratch each time [1], [9]. A central example is Atlas, a web-based application that enables researchers to define cohorts, perform characterization studies, and run comparative effectiveness or safety analyses through an intuitive graphical

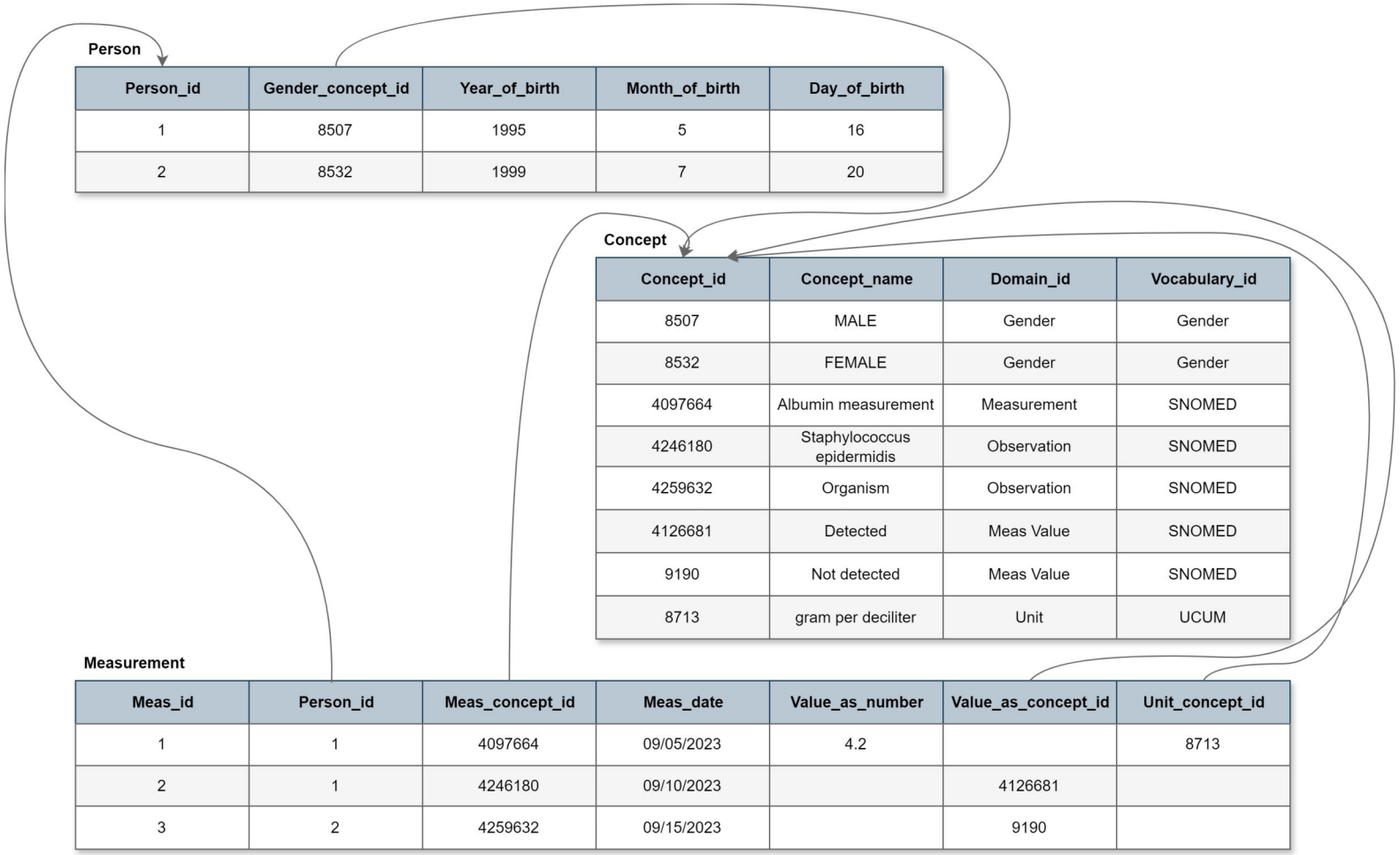


Figure 2.3: Sample data structure in the OMOP CDM, illustrating the Person, Measurement, and Concept tables. This figure shows how Person ID links patient demographics to clinical measurements and how Concept ID standardizes measurement types through a unified coding system.

interface. By providing a standardized environment for study design and execution, Atlas reduces the need for custom programming and ensures that analyses are transparent, reproducible, and easily shareable across research teams [9], [19].

The OMOP CDM has been employed in various fields of healthcare research to address complex questions through standardized and interoperable data. In the following, the main concepts and use cases associated with the OMOP CDM will be explained. These examples illustrate how the OMOP CDM supports large-scale observational studies across diverse healthcare settings:

Multicenter Surveillance

Multicenter surveillance refers to continuously monitoring healthcare trends across multiple healthcare institutions or geographic regions, including disease outbreaks, treatment patterns, and adverse events. The OMOP CDM facilitates multicenter surveillance by standardizing data from diverse healthcare systems, enabling researchers to aggregate and analyze data across different settings. As a result, public health trends can be monitored more comprehensively, emerging health threats identified more rapidly, and coordinated responses implemented across multiple locations. Additionally, the OMOP CDM enables cross-national comparisons of treatment regimens and health outcomes by harmonizing data from different registries and healthcare databases. For example, by analyzing standardized data from multiple regions, researchers can identify variations in disease management strategies and disparities in clinical outcomes across populations. Such insights help inform healthcare policies, optimize resource allocation, and improve patient care through data-driven decision-making [28], [29], [30].

Benchmarking

Benchmarking is the process of comparing healthcare performance across institutions or regions to identify best practices and areas for improvement. The OMOP CDM supports benchmarking by standardizing data from multiple healthcare providers, enabling cross-institutional comparisons on key metrics such as treatment effectiveness, patient outcomes, and healthcare costs. By aggregating data uniformly, researchers and policymakers can assess how different institutions perform in various areas and identify practices that lead to better patient care. A study utilizing the OMOP CDM demonstrated its applicability in benchmarking by analyzing pediatric prescription patterns across multiple countries. By harmonizing data from diverse healthcare systems, the study directly compared drug utilization trends, highlighting variations in prescribing practices and identifying potential areas for standardization and optimization. This supports quality improvement initiatives, provides insight into areas of healthcare inefficiency, and promotes the adoption of best practices across institutions [31].

Comparative Effectiveness Research

Comparative Effectiveness Research (CER) involves evaluating the relative effectiveness of different medical treatments or interventions in real-world settings. One of the primary challenges in CER is data fragmentation across various clinical information systems, including EHRs, insurance claims, and patient registries. This fragmentation often leads to inconsistencies in data formats, terminologies, and practices, making it difficult to conduct meaningful comparisons across diverse datasets. Data fragmentation significantly hinders the ability to compare treatments effectively, and harmonizing data across multiple institutions is essential for improving the quality of CER studies. The OMOP CDM addresses these challenges by integrating data from diverse healthcare systems into a unified format with standardized vocabularies enabling large-scale CER studies that are both reliable and reproducible. For example, researchers can compare the long-term effectiveness of different antihypertensive drugs by analyzing standardized data on medication exposure, clinical outcomes, and patient characteristics. Using standardized vocabularies (e.g., ICD-10 for diagnoses and RxNorm for drug names) ensures that the results are consistent and meaningful across datasets and sites, allowing for accurate and generalizable findings [32].

Pharmacovigilance and Drug Safety

Pharmacovigilance focuses on monitoring the safety of medications and detecting potential adverse drug reactions (ADRs) in the post-marketing phase. The OMOP CDM supports pharmacovigilance by integrating data from disparate real-world healthcare sources, such as hospital records, outpatient visits, and insurance claims. Researchers can use the Drug Exposure table to track medication use and the Condition Occurrence table to identify adverse events like gastrointestinal bleeding or liver toxicity. By linking exposure data with health outcomes, the OMOP CDM allows the identification of safety signals and the evaluation of risk factors associated with medications. This capability is essential for ensuring the ongoing safety of pharmaceuticals once they are on the market [33].

Disease Burden Estimation and Population Health Studies

The OMOP CDM can play a pivotal role in estimating disease burden across populations and understanding the prevalence and incidence of various health conditions. By standardizing data from diverse healthcare systems, the OMOP CDM enables the aggregation of health information, which can then be used to calculate disease prevalence, identify at-risk populations, and assess the broader impact of diseases on public health. For example, one approach for estimating disease burden is integrating geographic data with the OMOP CDM, allowing for spatial analysis of disease distribution. Geographic Information Systems (GIS) can be combined with OMOP CDM data to visually map disease prevalence and incidence at the regional or even sub-regional level. Researchers can identify patterns and disparities in disease burden across different populations and

areas by incorporating spatial dimensions, such as geographic location, healthcare access, and environmental factors [34].

Real-World Effectiveness of Medications and Vaccines

Evaluating the effectiveness of medications and vaccines in real-world settings is a crucial area of research, particularly for monitoring public health interventions and informing clinical decision-making. The OMOP CDM facilitates such analyses by standardizing data on patient exposures, whether to vaccines or medications, and subsequent health outcomes. For instance, the OMOP CDM enables large-scale studies on the effects of medications, such as examining whether certain drug classes influence disease susceptibility or severity. A recent international study utilized standardized data to assess the impact of commonly prescribed medications on COVID-19 outcomes, illustrating how harmonized real-world data can generate robust evidence on treatment effects across diverse populations. By enabling these comprehensive investigations, the OMOP CDM can play a crucial role in supporting public health policies, optimizing treatment strategies, and improving patient outcomes [35].

Health Outcomes Research in Chronic Diseases

Health outcomes research aims to evaluate the long-term impact of healthcare interventions on the health of patients with chronic diseases. The OMOP CDM supports this type of research by providing a standardized framework for tracking patient outcomes over time. For example, the OMOP CDM can be used to examine treatment patterns and health outcomes in patients with multiple chronic conditions, such as cancer and comorbid diseases. The Condition Occurrence and Drug Exposure tables enable researchers to track disease progression and treatment history, while the Measurement table captures clinical outcomes, including laboratory results and symptom assessments. The OMOP CDM's ability to harmonize data from multiple sources facilitates the identification of effective interventions and the improvement of clinical guidelines for chronic disease management [36].

Predictive Modeling for Patient Outcomes

Predictive modeling uses historical data to forecast future patient outcomes, such as the likelihood of disease progression, hospital readmission, or length of stay (LOS). The OMOP CDM enables predictive modeling by providing a structured, standardized dataset with a wealth of clinical, demographic, and treatment-related information. For instance, predictive models can be developed to forecast hospital LOS or the likelihood of hospital readmission. By linking data across various tables in the OMOP CDM, such as the Condition Occurrence (hospitalization), Drug Exposure (medication use), and Person (demographics) tables, models can be trained to identify patients at high risk of adverse outcomes. These predictive models provide clinicians with valuable insights to improve resource management and reduce unnecessary healthcare costs by identifying patients

who are likely to require extended hospital stays or experience complications that may lead to readmission. In this context, the OMOP CDM facilitates the development of tools for personalized care and targeted interventions, thereby enhancing clinical decision-making and improving patient outcomes. Such predictive analytics, made possible by standardized data integration within the OMOP CDM, can also support the development of more accurate and dynamic healthcare policies that better allocate resources across patient populations [37].

Healthcare Utilization and Cost Analysis

Understanding healthcare utilization and associated costs is crucial for effective resource allocation and policy development. The OMOP CDM provides a platform for analyzing healthcare costs by integrating data on medical treatments, hospitalizations, and outpatient visits. For instance, researchers may assess the pharmacological costs of diabetes treatment. In such analyses, researchers examine both direct costs, such as medication usage and hospital admissions, and indirect costs, such as additional healthcare services, long-term management needs, or downstream complications. By utilizing the Visit Occurrence and Drug Exposure tables, the OMOP CDM enables a comprehensive analysis of the economic burden of diseases, shedding light on the contribution of different treatments to healthcare costs. Moreover, by standardizing data from various healthcare systems, the OMOP CDM enables researchers to analyze the cost-effectiveness of different pharmacological therapies across various patient demographics. This approach enables policymakers and healthcare planners to understand better the full financial impact of new treatments, including not only the direct costs of care but also the potential costs arising from side effects and subsequent hospital admissions. It also enables comparisons of the costs associated with different therapeutic approaches, providing insights into the financial sustainability of disease management strategies. Evaluating the economic implications of diseases and treatments, particularly for conditions such as diabetes, significantly contributes to informed healthcare decision-making. It allows stakeholders to make evidence-based decisions on resource allocation, aiming for more cost-effective interventions that improve patient outcomes while reducing unnecessary financial burdens [38].

Longitudinal Cohort Studies

Longitudinal cohort studies track patient outcomes over time to examine the long-term effects of diseases or treatments on individuals. The OMOP CDM is well-suited for such studies because it integrates data from different time points and healthcare settings. For example, researchers can use the OMOP CDM to perform in-depth phenotyping of patients hospitalized with severe conditions, such as COVID-19, and track their clinical outcomes over time. The OMOP CDM enables the monitoring of patients across extended periods, linking treatment exposures with disease progression and long-term health outcomes. This ability to aggregate longitudinal data across diverse healthcare systems enhances the quality and generalizability of cohort studies, allowing for the

identification of key clinical factors that influence patient outcomes and the effectiveness of medical interventions [39].

2.5 ETL Process

An ETL process is a data integration process that combines, cleans, and organizes data from multiple sources into a single, consistent data set for storage in a data warehouse, data lake, or other target system. These data integration approaches involve three phases or tasks: Extract, Transform, Load [18].

During the extraction phase, data is extracted from one or more sources. Each separate system may use a different data organization and format. The extraction phase is followed by the transformation phase, in which transformation rules and techniques are defined and applied to transform the extracted data. This phase involves many subtasks. Typical transformations include applying business rules, cleaning, filtering, splitting, joining, encoding or decoding, deriving new calculated values, aggregating, transposing, or pivoting. In the last phase, the transformed data is transferred or loaded into the target system, which can be any data store, including a simple file or a data warehouse [18].

2.6 JSON Schema

JSON Schema [20] is a standard format that allows the definition and validation of the structure, content, and constraints of JSON data. JSON is widely used for data interchange across web applications and application programming interfaces (APIs) due to its lightweight, human-readable format. However, as the complexity of JSON data has grown, the need to standardize and validate its structure across different systems has become increasingly important, particularly in applications where consistent and structured data exchange is critical. JSON Schema addresses this need by providing a way to formally describe JSON data, enabling both humans and machines to understand and validate data structures efficiently [20].

The core of JSON Schema lies in its ability to define expected data types, properties, and validation rules for JSON objects. It introduces keywords to specify fundamental data types such as string, number, integer, object, array, boolean, and null. This allows for enforcing the type of data each field should contain, ensuring data integrity and reducing the potential for unexpected data-related errors in applications. Beyond types, JSON Schema provides validation mechanisms through keywords like `required`, which mandates that specific fields be present in the JSON data, and `additionalProperties`, which can restrict the presence of unspecified fields. These features give developers fine-grained control over the permissible structure of JSON objects, ensuring that each instance of JSON data aligns with expected formats and contains only relevant information [20].

JSON Schema further enhances validation through type-specific constraints. For example, in the case of strings, developers can enforce constraints such as `minLength` and

`maxLength` to control the allowable length of text data. Similar constraints exist for numbers, where boundaries can be set with minimum and maximum values. For arrays, developers can specify schemas for individual elements and minimum and maximum array lengths, adding flexibility in defining lists and collections within JSON data. JSON Schema also supports complex structures, allowing for nested objects and arrays. This feature is particularly useful when representing hierarchical data, which is common in JSON, as it enables developers to construct intricate, multi-level data schemas that match the complex structures required in many applications [20].

The process of using JSON Schema involves defining a schema document that specifies the structure, types, and constraints for the expected JSON data. A JSON validator then compares a JSON instance against this schema, checking for adherence to each specified constraint and returning detailed error messages if the JSON instance does not conform. This validation process allows developers to identify structural or data type inconsistencies before data is processed or stored, reducing the likelihood of runtime errors due to invalid data [20].

JSON Schema has become integral in various applications, particularly in API development, configuration management, and data quality enforcement. APIs often use JSON to exchange data between clients and servers, and JSON Schema ensures that data sent and received meets a consistent format, reducing errors and enhancing interoperability. In configuration management, JSON Schema can validate configuration files before they are used by software systems, helping prevent misconfigurations that could otherwise lead to system failures. JSON Schema definitions can also be transformed into documentation, clearly representing data requirements, improving developers' communication, and facilitating compliance with data standards [20].

In summary, JSON Schema plays a vital role in modern software development by providing a structured and standardized approach to defining and validating JSON data. By enforcing data integrity, enabling efficient error detection, and supporting documentation, JSON Schema contributes significantly to the reliability and robustness of applications that rely on JSON data, particularly in web and API-based architectures. Its structured approach to data validation has made it a preferred tool for maintaining data consistency and quality across distributed systems [20].

CHAPTER 3

Related Work

The integration of clinical data into standardized formats, such as the OMOP CDM, is a critical step in enabling large-scale, reproducible health data research. This process relies heavily on the design and execution of ETL pipelines that can handle the source schemas and data models [1], [18], [19]. The goal of this chapter is to provide a comprehensive overview of the state of the art in ETL processes, with a particular focus on their application to the EAV data model and the OMOP CDM integration.

The chapter begins by surveying existing approaches for transforming healthcare data into the OMOP CDM. It then focuses specifically on the challenges posed by the EAV data model, which differs structurally from the horizontal schema required by the OMOP CDM. Following this, the chapter reviews literature on conceptual modeling techniques that formalize ETL design. It also discusses quality characteristics essential for designing, evaluating, and refining ETL pipelines in practical settings. Finally, it compares widely used open-source ETL tools suitable for healthcare data integration. Figure 3.1 visualizes the ETL process, with the individual works positioned at the stage of the process to which they primarily contribute.

Together, these sections establish a foundation for understanding the methodological, technical, and practical considerations relevant to this thesis. They also help position the proposed work within the broader research landscape and identify existing gaps that this thesis aims to address.

3.1 ETL Process into the OMOP CDM

Transforming heterogeneous healthcare data into the OMOP CDM presents both conceptual and technical challenges [40], [41]. ETL processes must handle varied source schemas, inconsistent semantics, and performance bottlenecks [18]. Numerous tools and frameworks have emerged to support this process, ranging from Graphical User

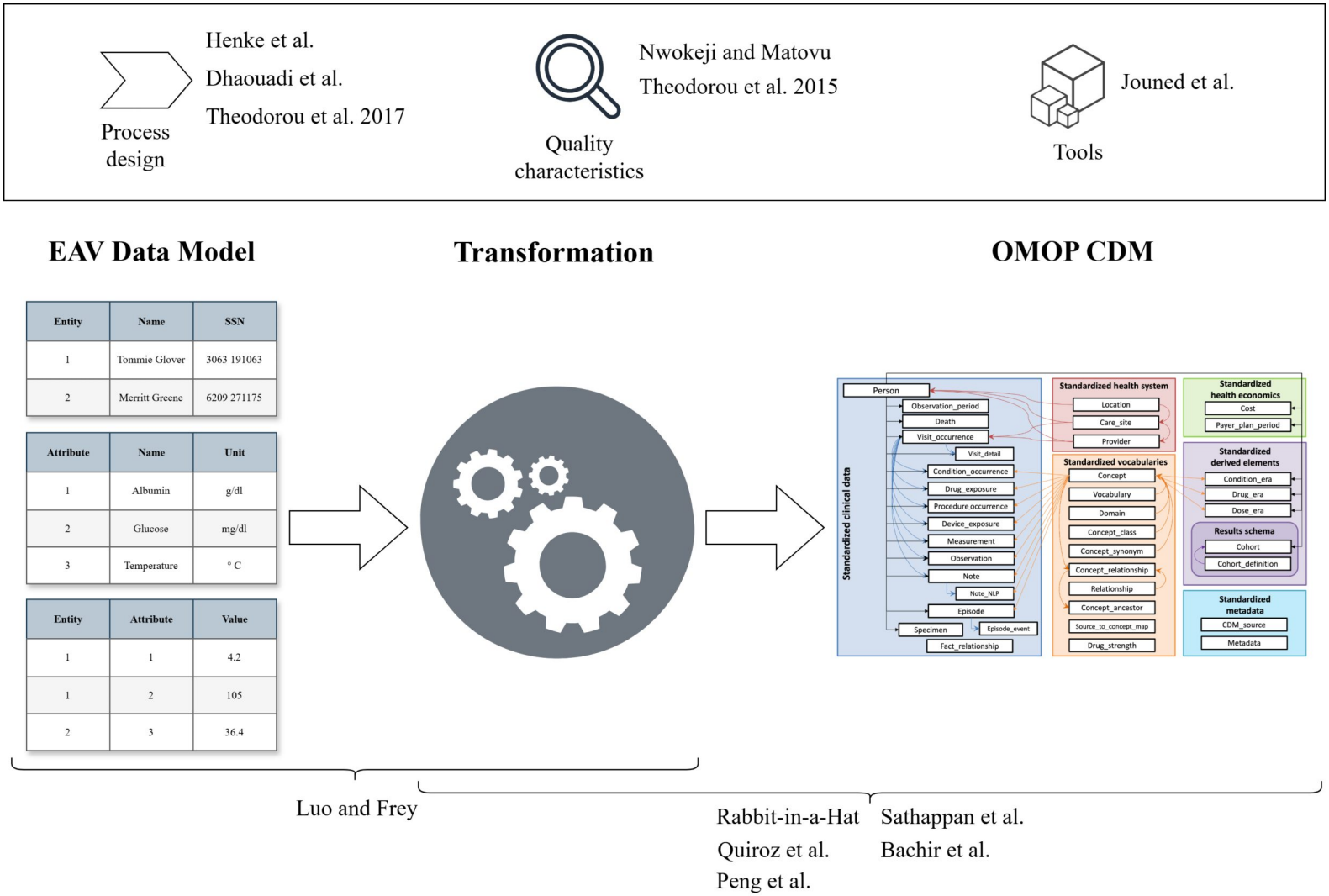


Figure 3.1: Visualization of the ETL process and positioning of related work along the pipeline from source data to the OMOP CDM [27].

Interface (GUI)-based design tools to fully automated, metadata-driven pipelines [42], [43]. However, the diversity of data formats, such as Fast Healthcare Interoperability Resources (FHIR) or EAV data models, limits the universal applicability of many of these approaches [12], [44].

One widely used design tool is Rabbit-in-a-Hat [19], [45], [46], developed by the OHDSI community. It is part of the OHDSI community's suite of open-source tools and is used explicitly for the ETL process of healthcare data into the OMOP CDM. A graphical user interface allows users to define mappings between source data and the OMOP CDM. Through a drag-and-drop interface, users can link source tables and columns to OMOP CDM equivalents. However, the tool is limited to horizontal data models and does not support the EAV data model. Moreover, Rabbit-in-a-Hat is intended for ETL specification rather than execution. It generates documentation but not executable ETL code, and therefore serves primarily as a design aid.

Moving toward executable, configurable ETL pipelines, Quiroz et al. [14] proposed a metadata-driven and generic ETL framework for converting health databases to the OMOP CDM. The framework includes a compiler that converts YAML files into an ETL script. The YAML files contain mapping logic for OMOP CDM tables. The mapping rules are defined on a column-by-column basis. They organize structured query language (SQL) snippets in key-value pairs that define the extract and transform logic to populate the OMOP CDM columns. Each YAML file describes the mapping logic for a target OMOP CDM table. It contains three sections:

1. The name of the OMOP CDM table being mapped (YAML field name)
2. The definition of primary keys used by the ETL framework to manage the load (insert) operations (YAML field primary_key)
3. The mapping rules for each column in the targeted OMOP CDM table (YAML field columns)

The authors developed a Data Manipulation Language (DML) to define the mapping logic. The DML uses the YAML key-value pairs to define the source data, the target OMOP CDM tables and columns, and the extract, transform, and load operations to map from source data to OMOP.

The first step in their ETL framework is to define the primary key of the OMOP CDM table. The framework utilizes definitions of primary keys to manage load operations. Like this, every row in the OMOP CDM table is mapped to all the relevant rows in the source table(s). The primary_key YAML field defines how to construct the primary key of the OMOP CDM table and whether it is composed of one or more sources.

The information needed to define the extract and transform operations from source data to an OMOP column is:

1. The name of the targeted OMOP column (YAML field name)
2. A listing of one or more source tables containing the data needed to populate the target field (YAML field tables)
3. An SQL expression defining how one or more fields from the source table(s) map to the OMOP field (YAML field expression)

A web application provides access to the ETL framework, allowing users to upload and edit YAML files via a web editor and obtain an ETL SQL script for use in development environments. The structure of the DML aims to maximize readability, refactoring, and maintainability while minimizing technical debt and standardizing the writing of ETL operations for mapping to the OMOP CDM. The authors emphasize the need for tools that support transparency of the mapping process and reuse by different institutions.

However, the authors do not specifically address the EAV data model. In the context of the ETL process, the EAV model can present unique challenges. For instance, transforming data from an EAV model to a horizontal, column-based model, such as the OMOP CDM, often involves pivoting the data, which can be a complex operation because pivot operations are commonly computationally inefficient. The EAV model is designed to handle large volumes of sparse data with varying attribute types. Pivoting involves restructuring the data, which requires aggregating and transposing rows into columns. The data is often stored in the same column and needs to be filtered based on the attribute. The pivoting operation typically consumes a significant amount of time and substantial computing resources, especially when the dataset is larger than memory, as it often involves multiple joins and aggregations, particularly when dealing with large datasets [10], [47]. While the paper provides a valuable framework for the ETL process, it does not delve into these EAV-specific issues. Therefore, additional strategies and tools are needed to effectively transform data from an EAV data model to the OMOP CDM.

A related study by Peng et al. [12] presented a comprehensive approach to integrating German real-world health data from FHIR to the OMOP CDM. The data used in this study is a synthetic data set based on the Common Core Data Set (CDS) of the German Medical Informatics Initiative (MII), specified in FHIR, that includes all essential aspects of patient EHR data and can be exchanged among researchers. It contains six basic modules covering patient demographics, hospital visits, diagnoses, procedures, laboratory observations, medications, and other extension modules (e.g., oncology, phenotypes, and biobank). Each module contains several FHIR profiles (e.g., the diagnoses module contains the Condition FHIR profile).

As a first step of the ETL process design, a semantic and syntactic mapping of the MII CDS specification to the OMOP CDM was developed. This mapping provided the basis for further implementation of the ETL process.

For the implementation of the ETL process, a code-based ETL process using Java was chosen due to the existence of the HAPI FHIR Java library that can process FHIR

resources, and the availability of the SpringBatch framework for Java programming, which is designed to process large datasets at once.

While the work of Peng et al. provides a robust framework for ETL processes in the context of FHIR and the OMOP CDM, it is not transferable to the transformation of EAV data into the OMOP CDM. As a standard communication format, FHIR resources are typically presented in XML or JSON format. In contrast, the EAV model is based on a relational database [12].

Sathappan et al. [48] conducted a feasibility study to assess the process of transforming Singaporean healthcare data, including EHRs and questionnaire-based data, into the OMOP CDM. The focus of their research was the SG_T2DM dataset, a rich collection of data from patients with type 2 diabetes mellitus. This dataset integrates structured clinical data (e.g., diagnoses, lab tests, medications) with patient-reported information obtained through standardized questionnaires.

The study aimed to determine whether local healthcare datasets could be effectively harmonized using the OMOP CDM. To that end, the authors evaluated multiple aspects of the transformation process, including data quality assessment, mapping strategies, ETL design, and conformance to the OMOP CDM. A key feature of their dataset was its hybrid nature, which posed unique challenges for harmonization.

The authors leveraged existing OHDSI tools, such as WhiteRabbit and Rabbit-in-a-Hat, to conduct an initial data profiling and mapping exercise. WhiteRabbit was used to scan the source data schema and generate metadata, which in turn informed the design of the ETL process. Rabbit-in-a-Hat helped define mappings between the source data fields and OMOP CDM concepts. These tools facilitated the documentation and validation of the mapping rules, although they did not support the complete execution of the ETL process.

The transformation of EHR data into the OMOP CDM was more straightforward because direct mappings were available between commonly used clinical codes and the OHDSI standardized vocabularies. However, the mapping of questionnaire responses proved more complex. Many questions did not have direct equivalents in the OMOP vocabulary, necessitating the creation of custom concepts or the use of proxy mappings based on expert judgment.

One of the main challenges addressed in the study was the structural transformation of the questionnaire data, which was initially organized in a wide-table format with one row per patient and one column per response. To conform with the OMOP CDM's table structure, the data had to be pivoted to a long format where each observation or response appears as a distinct row. This restructuring required careful consideration to preserve the semantic meaning of each response, particularly when aligning responses with standardized vocabularies.

Despite these challenges, the authors successfully mapped a significant portion of the dataset to the OMOP CDM and demonstrated that the resulting data conformed to the

OMOP CDM's requirements. However, the authors acknowledged that further refinement was needed to ensure semantic alignment, especially in the context of non-clinical data.

Notably, while the paper acknowledges the need to pivot questionnaire data and transform it into OMOP format, it does not provide in-depth details about the implementation of these pivot operations or the technical decisions made during the ETL scripting. When transforming the data from the EAV data model, the data must be pivoted from a long format to a wide format, which is the reverse transformation. As such, the study offers a valuable example of applying the OMOP CDM to a hybrid dataset but leaves some aspects of the transformation process, particularly those related to technical reproducibility, underspecified.

Nevertheless, the work of Sathappan et al. provides important insights into the feasibility of adopting the OMOP CDM in healthcare contexts. It highlights the flexibility of the OMOP CDM in accommodating diverse data sources and underscores the importance of iterative mapping, validation, and expert involvement in the ETL process.

More recently, Bachir et al. [43] presented a metadata-driven approach aimed at generalizing transformation steps in ETL processes, a challenge that remains central in the harmonization of clinical data. Recognizing the limitations of hard-coded, purpose-specific ETL pipelines, the authors explore how Metadata Repositories (MDRs), built on standards such as ISO/TS 21526 and ISO/IEC 11179-3, can externalize transformation logic and improve reuse across use cases.

The core innovation lies in leveraging standardized mappings, defining source-target relationships and transformation rules, and encoding them in a structured and traceable way. The integration of provenance metadata using the W3C PROV model adds transparency and auditability to the data integration workflow, which is crucial in healthcare settings where regulatory compliance and reproducibility are non-negotiable.

A prototype built on the DEHub metadata repository demonstrates the feasibility of this architecture, translating Comma-Separated Values (CSV)-based patient data into OMOP CDM format using rule-based mappings. However, while the extract and load phases benefit from automation, the transformation step remains partially generalized, especially for complex, non-trivial transformations. The prototype supports 1:1 and 1:n mappings effectively, but still requires technical expertise to define advanced transformation logic.

While the work of Bachir et al. makes a valuable contribution to the formalization and externalization of transformation rules using standardized metadata models, it focuses primarily on the storage and representation of these mappings within an MDR. The practical implementation of the ETL process itself is only discussed at a conceptual level. The prototype demonstrates how rules can be defined and stored, but it does not provide a mechanism for automatically executing those rules as part of a runnable ETL workflow. Addressing this implementation gap remains a critical step toward making metadata-driven ETL approaches fully operational and scalable in real-world healthcare data integration scenarios.

3.2 ETL Process from the EAV Data Model

Transforming data from the EAV data model into a horizontal, column-based relational format is a critical step in many clinical data integration workflows, particularly when mapping to target schemas like the OMOP CDM. The OMOP CDM requires data to be structured in a normalized, wide format, which contrasts with the vertical structure of EAV tables. As a result, EAV-based data must undergo pivot operations that transpose rows into columns to align with OMOP CDM's tabular schema [49].

Luo and Frey [10] presented techniques to improve the efficiency of pivot operations in the context of EAV-modeled data. They noted that the process of pivoting can be time-consuming and resource-intensive, particularly when performed regularly, such as for daily content refreshes in a clinical data warehouse.

While the EAV model offers flexibility in handling sparse, heterogeneous, or extensible datasets, such as those found in clinical observations, it introduces significant computational complexity during transformation. Pivoting EAV data can be resource-intensive, particularly when dealing with large volumes of records that include numerous attributes and high levels of sparsity. Many values may be null or missing, and the data is often fragmented across multiple tables. These structural characteristics lead to complex SQL queries involving numerous joins, filters, and aggregations, which can place a heavy burden on memory and processing resources [47].

To address these challenges, Luo and Frey propose a set of optimization techniques aimed at improving the efficiency of pivot operations over EAV-modeled data. Their work targets explicitly scenarios involving frequent or large-scale pivoting, such as daily refreshes of data marts or clinical data warehouses. The proposed techniques include:

1. Filtering out EAV tuples related to unneeded clinical parameters early on: By identifying and removing tuples corresponding to irrelevant clinical parameters before the pivot operation, the volume of data being processed is significantly reduced. This selective filtering minimizes unnecessary operations and intermediate storage requirements.

2. Supporting pivoting across multiple EAV tables: In many systems, clinical data is not stored in a single EAV table but instead partitioned across multiple EAV-like structures. Supporting multi-table pivoting reduces redundancy in transformation workflows and improves the scalability of the ETL process.

3. Conducting multi-query optimization: Pivoting often involves executing a series of interrelated queries. Optimizing these as a group rather than in isolation enables more efficient query planning and execution, particularly when common sub-expressions or filtering conditions can be shared.

These optimizations, while developed for general clinical warehouse use cases, are directly applicable to OMOP CDM transformation workflows.

Despite the contributions of Luo and Frey, a gap remains in translating these performance-focused optimizations into standardized ETL frameworks or OMOP CDM-specific pipelines. Most OMOP CDM ETL tools, such as Rabbit-in-a-Hat or metadata-driven systems like those proposed by Quiroz et al. and Bachir et al., assume horizontally structured input data. They provide little support for the dynamic, schema-less characteristics of EAV models. As a result, institutions using EAV-based systems must either pre-transform their data through manual pivoting scripts or develop custom integration logic, which increases the development burden and hampers reusability.

3.3 Conceptual Modeling of an ETL Process

Conceptual modeling plays a critical role in designing and understanding ETL processes, especially in domains like healthcare data integration, where transformations are often complex, iterative, and semantically rich. Explicit conceptual representations can facilitate communication among stakeholders, improve reusability, and support automation, validation, and optimization of ETL workflows [1], [8], [19], [50].

According to the OHDSI community [19], the ETL process for converting raw clinical data into the OMOP CDM is best understood as a structured, four-step workflow.

1. Design the ETL
2. Create the code mappings
3. Implement the ETL
4. Quality control

In the design phase, data experts and OMOP CDM experts collaborate to align the source schema with the OMOP CDM structure. Profiling tools such as WhiteRabbit and mapping tools such as Rabbit-in-a-Hat are frequently used to support this stage. The code mapping phase focuses on semantic alignment, in which domain experts create mappings between source terminologies and the OHDSI standardized vocabularies, typically supported by OHDSI tools like Usagi and Athena. During the implementation phase, technical staff translate the design and mappings into executable ETL scripts, most often written in SQL, enabling reproducibility and scalability. Finally, the quality control phase involves systematic verification of the ETL output, using validation tools such as Automated Characterization of Health Information at Largescale Longitudinal Evidence Systems (Achilles) and the Data Quality Dashboard (DQD) to ensure correctness, completeness, and conformance.

This four-step structure emphasizes that ETL development is inherently iterative, often requiring cycles of refinement between design, implementation, and validation. It also highlights the interdisciplinary nature of the process, where clinical knowledge,

technical skills, and methodological rigor must come together to achieve reliable data transformation.

This structured framework proposed by OHDSI provides a practical foundation for designing and implementing ETL processes into the OMOP CDM.

Building on this foundation, Henke et al. [40] conducted a literature review to conceptualize a more fine-grained and generic data harmonization process. Their work focuses on publications addressing the harmonization of clinical and claims data into the OMOP CDM and derives a set of nine process steps that extend and refine the four stages suggested by OHDSI.

From 23 publications, they conceptualized a generic data harmonization process for the OMOP CDM, consisting of nine process steps. Based on the literature, the authors determined a chronological order for the data harmonization process, with the most agreement across publications:

1. Dataset specification
2. Data profiling
3. Vocabulary identification
4. Coverage analysis of vocabularies
5. Semantic mapping
6. Structural mapping
7. ETL process
8. Qualitative data quality analysis
9. Quantitative data quality analysis

They assigned the identified steps to those proposed by OHDSI. Steps 1-3 and 6 were assigned to OHDSI's first step, "Design the ETL". Steps 4 and 5 were assigned to OHDSI's second step, "Create the code mappings". Step 7 was assigned to OHDSI's third step, "Implement the ETL". Finally, steps 8 and 9 were assigned to OHDSI's fourth step, "Quality control".

Furthermore, the authors identified seven OHDSI tools that supported five of the process steps.

The tool WhiteRabbit was used for data profiling. OHDSI provides a vocabulary repository called Athena and a tool that supports semantic mapping, called Usagi. The structural mapping was performed using the tool RabbitInAHat. For performing quality checks, OHDSI provides the tools Achilles and the DQD. A quantitative data quality analysis was performed using Atlas to define cohorts.

The authors noted that the publications have shown that some process steps may not be relevant in a given use case. Additionally, using OHDSI tools is seen as optional. They also mentioned that the process should be considered iterative, so errors identified during the quality analysis may necessitate repeating the process steps. However, depending on the context, some process steps may be skipped in subsequent iterations. The authors suggested that the defined generic data harmonization process can be used as a step-by-step guide to assist other researchers in harmonizing source data in the OMOP CDM.

Dhaouadi et al. [51] provided a comprehensive review of data warehousing modeling approaches. The authors summarized relevant works related to modeling data warehousing approaches, ranging from classical ETL processes to Extract, Load, Transform (ELT) design approaches. They conducted a systematic literature review and defined a detailed set of comparison criteria. They noted that there is no standard model for representing and designing this process, which has led several researchers to propose modeling methods based on different formalisms. These formalisms include Unified Modeling Language (UML), ontology, Model-Driven Architecture (MDA), Model-Driven Development (MDD), and graphical flow, which includes Business Process Model Notation (BPMN), Colored Petri Nets (CPN), YAML, CommonCube, Entity Modeling Diagram (EMD), and more. The paper emphasizes that the success of data warehouse projects is essentially based on correctly modeling the ETL process. These works provide valuable insights into the design and implementation of ETL processes, which might be applied to the transformation of data from the EAV data model to the OMOP CDM.

Theodorou et al. [52] investigate in their work the recurring structures or patterns within ETL workflows. The authors highlight the complexity and importance of ETL processes in data integration and management tasks, and propose a novel approach for identifying frequent patterns in these workflows. ETL workflows, often highly complex, require efficient design and optimization, making the identification of recurring structures crucial for understanding and improving ETL systems.

The approach introduced in this paper models ETL workflows as labeled directed graphs, where nodes represent operations and edges represent the flow of data. By using graph mining techniques, the authors can identify common patterns that frequently appear across multiple ETL workflows. These patterns, once identified, can be used to simplify workflow representation and provide insights into common practices, inefficiencies, or opportunities for optimization.

In their empirical study, the authors apply this methodology to workflows derived from the Transaction Processing Performance Council – Data Integration (TPC-DI) benchmark, a widely used data integration specification. The TPC-DI benchmark is a standardized performance benchmark designed to evaluate the efficiency and scalability of data integration systems. It simulates a realistic data warehousing environment by measuring the ability to ETL data from multiple sources into a central repository. TPC-DI assesses both throughput and data consistency, providing a comprehensive metric for comparing different ETL tools and architectures [53].

Theodorou et al. identify several frequent patterns, such as sequence, parallel split, synchronization, exclusive choice, and simple merge, which are commonly found in ETL processes. These patterns are significant because they encapsulate the core structural components of ETL workflows and can be reused or optimized in future process designs.

Furthermore, the paper explores how these identified patterns can be mapped to conceptual models, facilitating the creation of abstract representations of ETL workflows. This mapping enables the transformation of ETL workflows from logical models to conceptual models, making it easier to analyze and optimize them at a higher level. Additionally, the authors argue that the frequent patterns can be used to generate cost models, which are crucial for evaluating the efficiency of ETL workflows and making decisions about their optimization.

The authors' work provides significant value to the field by offering an empirical and systematic methodology for understanding the common structures within ETL workflows. The identification of frequent patterns not only helps improve the conceptualization and design of ETL processes but also contributes to the optimization of their execution and maintenance in real-world applications.

Together, these works highlight the diversity of conceptual modeling strategies and the critical importance of ETL abstraction in complex integration scenarios. For the specific case of transforming EAV-modeled health data into the OMOP CDM, such conceptualization efforts provide valuable guidance. While tools and standards for the OMOP CDM ETL process are evolving, the integration of conceptual modeling practices, particularly those that capture semantic mappings, process patterns, and quality controls, remains essential for building scalable, maintainable, and transparent ETL systems.

3.4 Quality Characteristics for ETL processes

Theodorou et al. [54] defined a comprehensive model for ETL process quality characteristics based on a systematic literature review. The authors focus on defining specific quality characteristics and metrics for evaluation. The model consists of five process characteristics with construct implications and three process characteristics for design evaluation. They include data quality, performance, upstream overhead, security, auditability and adaptability, usability, and manageability, respectively.

Nwokeji and Matovu [50] performed a comprehensive analysis of the current state and challenges of ETL processes in the context of Big Data. The review aimed to assess existing approaches, identify research gaps, and suggest future directions for improvement. The authors identified eight quality attributes from primary studies. They include performance, interoperability, flexibility, reusability, fault-tolerance, scalability, reliability, and data quality.

Both models overlap significantly and can be summarized by the characteristics defined by Theodorou et al. Upstream overhead was added as a subcategory to Performance. The quality characteristics are defined as follows.

Data quality measures how well the data output of an ETL solution meets the criteria for being accurate, complete, and suitable for its intended use.

Performance refers to the amount of resource utilization and the timeliness of the service delivered.

Security refers to protecting information during data processes and transactions.

Auditability refers to the ability of the ETL process to provide data and business rule transparency.

Adaptability describes how an ETL process can effectively and efficiently be adapted for different operational or usage environments.

Usability describes the ease of use and configuration of the ETL process.

Manageability defines the ease of monitoring, analyzing, testing, and tuning the implemented ETL process.

3.5 ETL Tools

ETL tools are commonly categorized into two categories: tool-based and source-code-based [55], [56]. Tool-based ETL tools are pre-built software solutions designed explicitly for ETL processes, typically featuring a GUI and pre-built functionalities to simplify data transformation tasks. These tools are user-friendly and designed to minimize the need for extensive programming knowledge. Several open-source data integration tools in this category have achieved high levels of maturity and performance, particularly in healthcare use cases [56].

In contrast, source-code-based ETL solutions offer developers greater flexibility. These solutions typically involve writing custom scripts or applications in programming languages like Java or Python. While these solutions offer more control and customization, they require advanced programming expertise and may incur higher maintenance costs, which can increase the complexity of the development process [56].

This thesis focuses on open-source ETL tools primarily due to licensing considerations. Open-source tools eliminate the need for costly licenses associated with proprietary software, making them more accessible for academic research and organizations with limited budgets. Furthermore, open-source software often provides greater control over the software's long-term sustainability and usability, as there are no concerns about the tool being discontinued or locked behind paywalls. These factors make open-source tools an ideal choice for healthcare data integration, where cost-effectiveness, flexibility, and long-term viability are critical. Furthermore, adopting open-source ETL tools facilitates the reuse and extension of the developed prototype, enabling future improvements or adjustments.

Additionally, using an ETL tool rather than building a solution from scratch significantly reduces development overhead. While custom-built solutions may provide more tailored functionality, they demand substantial resources for optimization and ongoing maintenance. In contrast, pre-existing ETL tools are already optimized for performance and have been rigorously tested, offering a stable foundation that can significantly reduce development time and complexity. These characteristics make such tools particularly advantageous for non-technical users, such as healthcare professionals or researchers with limited programming expertise, who may benefit from accessible, user-friendly systems. This capability is crucial for fostering broader adoption and integration of the solution, as it lowers initial hurdles and allows teams without coding expertise to benefit from the developed system. For those reasons, only open-source ETL tools will be covered in this section.

Given the variety of open-source ETL tools available, it is crucial to carefully evaluate which tool best suits the specific needs of healthcare data integration. The selection of an appropriate ETL tool depends on several factors, including performance, ease of use, interoperability, and flexibility. Each tool has its strengths and limitations, making it essential to understand how well they align with the requirements of transforming healthcare data into standardized formats, such as the OMOP CDM.

Widely used open-source ETL tools are Pentaho Data Integration (PDI) and Apache NiFi.

Apache NiFi [57] is a tool designed for scalable data routing and automating data flows between systems. It is designed for real-time data processing and features a web-based user interface that simplifies the design, monitoring, and control of data flows. Apache NiFi supports extensive data integration and transformation capabilities, allowing for the extension of its core functionality through the development of custom processors. This flexibility makes it a powerful solution for complex data workflows [56], [58].

PDI [59], also known as Kettle, is a tool designed to facilitate and streamline data management processes, particularly within Business Intelligence applications. Its primary function is to perform ETL operations. PDI facilitates data extraction from various sources, offering a comprehensive set of transformation tools that enable users to clean, normalize, and aggregate data through functions, filters, and calculations. Once the data is transformed, it can be loaded into various target systems or databases. Highly extensible, PDI supports custom scripting and plugin development, allowing users to create tailored data transformation workflows. The tool's standalone GUI, called Spoon, enables users to visually design data transformations, which are saved in XML format. These transformations, whether stored as XML files or within a database repository, can be executed using the execution engine, known as Kitchen.

In addition to its ETL capabilities, PDI integrates seamlessly with other Business Intelligence tools, providing advanced analytics and reporting functionalities. This integration enables users to create dashboards and visualizations, providing powerful insights into the transformed data [55], [60], [61].

Jouned et al. [49] performed a comparative evaluation of PDI, Apache NiFi, and a custom-built solution in the context of transforming healthcare data to the OMOP CDM. The evaluation of these tools is based on a set of criteria identified through an extensive review of existing literature, which includes connectivity and interoperability, user interface design, performance, and technical flexibility.

The findings indicate that all three tools are capable of performing basic data transformation tasks effectively, although each tool exhibits distinct characteristics in terms of usability and performance. PDI is particularly noted for its user-friendly interface, which simplifies the process of data mapping and transformation. The graphical design environment makes PDI an appealing choice for users with limited technical expertise, as it streamlines the ETL process without requiring extensive programming skills. In contrast, Apache NiFi, while offering a steeper learning curve, provides greater technical flexibility and scalability, making it more suitable for advanced users who require customization and efficiency in handling larger datasets. The custom-built tool, although tailored to meet specific needs, was found to lack the scalability and robustness of the more established tools.

Regarding performance, the study reveals that all tools are competent in handling basic transformation tasks, but their capabilities diverge when working with larger and more complex datasets. Apache NiFi was found to excel in terms of speed and scalability, handling high-volume data processing more effectively than the other tools. While PDI is adequate for smaller-scale transformations, it is less efficient when tasked with more resource-intensive processes. The custom-built solution, while offering specialized functionalities, did not demonstrate the same level of performance optimization as Apache NiFi.

The evaluation of technical flexibility further underscores the differences between the tools. Apache NiFi is identified as the most technically flexible solution, offering extensive customization options that enable it to be tailored to a wide array of use cases. PDI, although less flexible in terms of customization, strikes a balance between ease of use and functional capabilities, making it suitable for a broader range of users. The custom-built tool, although highly specialized, lacked the versatility and adaptability of either PDI or Apache NiFi.

In conclusion, the study suggests that all three ETL tools are adequate for transforming healthcare data into the OMOP CDM, with their applicability largely depending on the user's technical expertise and the specific requirements of the transformation task. PDI is recommended for those seeking an intuitive interface and ease of use, while Apache NiFi is better suited for users who require advanced features, scalability, and customization. The custom-built tool, while adequate for specific needs, lacks the general applicability and robustness of the other tools and may be less suitable for larger-scale or more complex data transformations.

3.6 Conclusion

While existing approaches contribute important tools, frameworks, and conceptual models for OMOP CDM transformation, they often fall short in addressing the practical complexities of integrating EAV-modeled data. This thesis aims to bridge that gap by developing an ETL solution that explicitly supports the transformation of EAV-modeled data from the research database of the MedUni Vienna into the OMOP CDM. Most prior work has focused on ETL processes that assume horizontally structured source data [14], [19], [45], which limits their applicability to healthcare systems that rely on EAV-based representations. The EAV model introduces additional complexity into the transformation process, particularly the need for pivoting, which is both computationally expensive and challenging to generalize [10].

Furthermore, previous studies that addressed pivoting from EAV models did not target the OMOP CDM as the output schema. The OMOP CDM not only imposes a rigid relational structure but also requires alignment with standardized terminologies, such as SNOMED CT and RxNorm. This dual requirement for structural normalization and semantic harmonization significantly increases the complexity of the transformation process. Consequently, this thesis aims to fill this gap by developing an ETL approach that supports EAV-to-OMOP CDM transformation in a way that is both semantically transparent and technically robust.

Table 3.1 presents an overview of the related work and their characteristics regarding the transformation from the EAV data model to the OMOP CDM.

Ref	Author	Year	OMOP	EAV	Terminology	Conceptual	Tool-Based	Code-Based	Metadata
	ETL process into the OMOP CDM								
[45]	OHDSI		✓	×	×	✓	✓	×	✓
[14]	Quiroz et al.	2022	✓	×	×	×	×	Partial	✓
[12]	Peng et al.	2023	✓	×	✓	×	×	✓	×
[48]	Sathappan et al.	2021	✓	Partial	Partial	×	×	×	×
[43]	Bachir et al.	2025	✓	×	✓	✓	✓	Partial	✓
	ETL process from the EAV data model								
[10]	Luo and Frey	2016	×	✓	×	✓	×	×	×
	Conceptual modeling of an ETL process								
[19]	OHDSI	2021	✓	×	✓	✓	✓	×	×
[40]	Henke et al.	2024	✓	×	✓	✓	✓	×	×
[51]	Dhaouadi et al.	2022	×	×	×	✓	×	×	×
[52]	Theodorou et al.	2017	×	×	×	✓	×	×	×
	Quality characteristics for ETL processes								
[54]	Theodorou et al.	2015	×	×	×	✓	×	×	×
[50]	Nwokeji and Matovu	2021	×	×	×	✓	×	×	×
	ETL Tools								
[49]	Jouned et al.	2025	✓	✓	×	✓	✓	✓	×
This work		2025	✓	✓	✓	✓	✓	×	✓

Table 3.1: Overview of the ETL-related work and its characteristics regarding the transformation from the EAV data model to the OMOP CDM.

CHAPTER 4

Methodology

This chapter outlines the methodological framework applied in the development of the generic ETL code base for transforming EAV-modeled data from the research database of the MedUni Vienna to the OMOP CDM. It describes the structured approach to answer the research questions defined in Section 1.3.

The chapter is structured in two parts. The first part provides an overview of the methodology, including the relationship between the individual phases and the research questions. The second part presents a detailed description of each phase, focusing on the objectives, methods, and outputs associated with each phase.

4.1 Overview

The methodological approach of this thesis combines theoretical analysis, iterative system design, prototypical implementation, and systematic evaluation to address the research questions comprehensively. It is grounded in the principles of modern software engineering and is structured into four interrelated phases: Analysis, Design, Implementation, and Evaluation. These phases represent a development lifecycle that is both conceptually robust and practically adaptable, in line with the demands of medium-scale software projects [62]. An overview of the methodological process is shown in Figure 4.1.

The approach follows an evolutionary prototyping model [63], in which the system is developed incrementally. Each phase builds upon the outcomes of the previous one while also enabling feedback loops that allow for continuous adjustment of earlier decisions based on insights gained during implementation and evaluation.

The Analysis Phase includes a literature review [64], focusing on ETL systems in the healthcare domain, particularly regarding the transformation of data from the EAV model to the OMOP CDM. It aims to derive functional and non-functional requirements that serve as the conceptual and technical foundation for subsequent phases. This phase

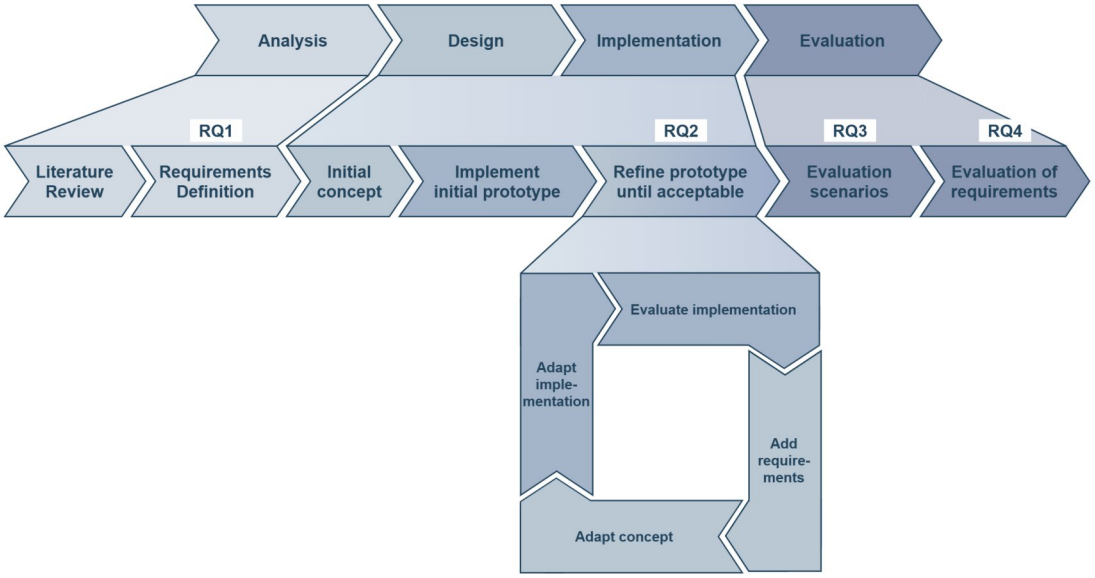


Figure 4.1: Graphical representation of the methodological approach.

primarily addresses research question RQ1 and partially informs research question RQ2 by shaping the system's initial design constraints.

The Design and Implementation Phases follow the evolutionary prototyping methodology [63]. An initial working prototype of the ETL system will be created and refined through four iterative cycles. Each iteration will focus on progressively improving the system, incorporating feedback, and making necessary adjustments based on practical testing and evolving requirements. In the Design Phase, an initial conceptual model for the ETL system is developed, which is refined based on added requirements and in response to feedback from ongoing implementation activities. The Design Phase addresses research question RQ2 by defining the structural and architectural blueprint for the ETL process. In the Implementation Phase, the system design is translated into a working prototype. Iterative development cycles focus on refining technical components, addressing practical constraints, and enhancing system quality based on continuous testing. Each iteration contributes to the progressive realization of a robust ETL system. The Implementation Phase continues to address research question RQ2, particularly from a technical realization perspective.

The process concludes with the Evaluation Phase, where the final prototype is assessed. This phase includes applying the solution to specific use cases to examine its adaptability (research question RQ3) and evaluating how well the system meets the defined requirements (research question RQ4).

This structure ensures that theory and practice are continuously aligned and that design, implementation, and validation are iteratively integrated [62].

The detailed steps of each phase are described in the following sections.

4.2 Analysis Phase

The Analysis Phase lays the foundation for the ETL system by systematically examining the current context and identifying the requirements for transforming healthcare data from the EAV model into the OMOP CDM. It serves as a bridge between understanding the problem and designing a suitable solution and is critical to ensuring that the resulting system meets both technical and user-driven expectations.

This phase is divided into two core activities: a literature review and a requirements analysis. The literature review is structured around three domains: ETL processes targeting the OMOP CDM, ETL processes originating from EAV-based data models, and general ETL methodologies. Each domain is examined with a focus on theoretical foundations, process characteristics, and best practices, ensuring that the resulting system requirements are grounded in current research and practical experience.

The primary objective of the literature review is to provide an in-depth understanding of the landscape of ETL systems, with a particular focus on their application to healthcare data transformation, the EAV data model, and the OMOP CDM. The review will examine state-of-the-art ETL tools and frameworks, identifying how existing systems address the challenges of transforming complex healthcare data. This thorough exploration informs the development of the system's requirements.

The first area explores ETL processes involved in transforming healthcare data into the OMOP CDM. Existing research is reviewed to examine how systems map data from diverse sources into the OMOP CDM. This section identifies data transformation techniques for standardizing and harmonizing healthcare data, including schema mapping and data quality management strategies.

The second area focuses on ETL processes for transforming data from the EAV model. Given the model's flexibility, techniques for handling sparse and dynamic attributes in EAV-based data are explored. Literature is reviewed to identify pivoting methods commonly used to transform EAV data into more structured formats, such as the OMOP CDM. This section focuses on strategies for managing data inconsistencies and ensuring data integrity during the transformation process.

The third area examines general ETL processes, focusing on the theoretical foundations of the extraction, transformation, and loading stages. The core principles of ETL design are reviewed, including data extraction techniques (e.g., batch processing, real-time streaming), data transformation strategies, and best practices for loading data into target models. Quality characteristics critical for ETL processes are also identified, particularly in the context of healthcare data transformation.

From this foundation, a set of requirements is derived. The requirements are grouped into functional requirements, which describe what the system must do (e.g., extract data, apply

mappings, ensure referential integrity), and non-functional requirements, which describe how the system should perform (e.g., performance, maintainability, scalability) [62].

The result of this phase is a baseline catalog of system requirements, which serves as a reference point for the subsequent Design Phase. While iterative refinement is expected during implementation, this structured baseline supports a controlled evolution of the system, reducing the risk of misaligned expectations and costly redesigns.

4.3 Design Phase

The Design Phase focuses on creating both a conceptual and technical blueprint for the ETL process that will transform healthcare data from the EAV model to the OMOP CDM. Following the principles of evolutionary prototyping, this phase is not executed strictly linearly but evolves iteratively in parallel with the Implementation Phase. This parallel structure enables ongoing refinement of the system design based on practical feedback and emerging requirements.

The process begins with the development of an initial conceptual architecture. This architecture defines the primary components of the ETL pipeline: data extraction, transformation, and loading. It outlines their high-level interactions and provides a macroarchitectural view of the system's structure that helps establish system boundaries and responsibilities across components. The initial concept is broad, outlining key components, but these elements are flexible and will evolve as the system is implemented.

Subsequent iterations refine this conceptual model. Each iteration integrates additional requirements identified during implementation and adapts the design to reflect both domain-specific constraints and practical considerations. Design decisions are guided by internal evaluations as well as discussions with a domain expert who specializes in electronic health records, interoperability, and health information management and serves as a key stakeholder in the design validation process. During the discussions, architectural choices and design alternatives were critically reviewed in iterative meetings, and adjustments were made until consensus on the best approach was reached. These design discussions help ensure that architectural decisions align with the project's goals and that evolving constraints are adequately addressed.

The output of the Design Phase is an evolving system specification that includes both the architectural model and the technical strategy for implementation. It provides a solid basis for the Implementation Phase while remaining adaptable to changes introduced during prototyping.

4.4 Implementation Phase

The Implementation Phase focuses on transforming the theoretical and technical designs from the Design Phase into a functional ETL prototype. Following the evolutionary

prototyping methodology, development is carried out in iterative increments that support ongoing testing, validation, and adaptation.

The process begins with the development of an initial prototype that reflects the core structure of the ETL pipeline based on the initial concept established during the Design Phase. This early version includes basic functionality such as data extraction, transformation logic, and loading routines. It primarily validates the feasibility of the conceptual design and the correctness of the end-to-end data flow.

The development progresses incrementally, with each iteration introducing new features, refining system behavior, and addressing issues identified during testing. Over time, the system evolves through continuous integration of additional features, performance improvements, and error-handling mechanisms. Design adaptations are informed by practical experience during development and are discussed with the domain expert, who guides system behavior and alignment with project objectives. This approach enables continuous system refinement, ensuring that issues are addressed promptly and that the system development remains aligned with both functional goals and non-functional quality attributes.

By the end of the Implementation Phase, the prototype is refined into a fully functional system capable of reliably transforming healthcare data from the EAV model into the OMOP CDM. The implementation reflects the defined requirements and incorporates flexibility for future adaptations, thereby providing a stable foundation for the subsequent Evaluation Phase.

4.5 Evaluation Phase

The Evaluation Phase will assess the ETL system's ability to meet the requirements defined in the Analysis Phase. This phase involves two key activities: conducting evaluation scenarios and analyzing the fulfillment of the requirements.

Two evaluation scenarios will be conducted to evaluate the flexibility and adaptability of the ETL system. These evaluation scenarios will involve adapting the generic ETL code base to two specific healthcare datasets that follow the EAV model but have different structures and characteristics. The evaluation of each evaluation scenario will focus on verifying that the system correctly transforms the data into the OMOP CDM format, measuring performance metrics, and documenting any challenges encountered while adapting the ETL process to the specific datasets.

Additionally, the evaluation will systematically examine the implementation against each functional and non-functional requirement defined during the Analysis Phase. Each requirement will be assessed to determine whether it is fully, partially, or not fulfilled. This assessment will be based on the features of the system, including implementation details, runtime behavior, and configuration capabilities. Justifications will be provided for each assessment to ensure transparency and traceability.

CHAPTER 5

Design and Implementation

This chapter presents the design and implementation of the generic ETL code base that transforms healthcare data from the EAV model of the MedUni Vienna into the OMOP CDM.

The chapter is organized into several sections, starting with the section "Requirement Definition", which outlines both the functional and non-functional requirements that guided the design and development of the ETL process.

Next, the section "Implementation Concept" provides a high-level overview of the ETL process, describing the transformations that facilitate the conversion of data from the EAV model to the OMOP CDM. The Semantic Mapping section discusses the inclusion of standardized terminologies in the source schema, a key aspect of data normalization. A detailed description of the structural mappings between the source database and the OMOP CDM is also provided.

The following section, "Prototypical Implementation", explains the chosen technology stack and tools used to implement the ETL process. This section also covers implementation details, including code structure, modularity, and functionality. The Setup of the OMOP CDM section elaborates on how the OMOP CDM was configured to receive the transformed data. Finally, the Data Transformation and Loading section outlines the processes used to extract, transform, and load the data into the OMOP CDM.

5.1 Requirement Definition

Based on the insights from the literature review presented in Chapter 3, the functional and non-functional requirements for the ETL process, designed to transform healthcare data from the RDA platform of the MedUni Vienna, modeled in the EAV format, to the OMOP CDM, are identified. These requirements form the foundation for the development

and implementation of the ETL process, ensuring it meets both technical and business objectives.

The functional requirements define the core capabilities of the ETL system, specifying the processes of data extraction, transformation, and loading, while the non-functional requirements address architectural characteristics. Both sets of requirements are outlined in the following sections.

5.1.1 Functional Requirements

The functional requirements define the core capabilities and behaviors that the ETL system must exhibit in order to successfully transform healthcare data from the RDA platform of the MedUni Vienna, modeled in the EAV format, to the OMOP CDM. These requirements are essential for ensuring that the ETL process operates efficiently, accurately, and in alignment with the standards set by the OMOP CDM.

This section outlines the specific actions the system must perform, including identifying and extracting relevant data, transforming it according to OMOP CDM conventions, and loading it into the target database while preserving data integrity. The functional requirements also address the flexibility of the ETL system, allowing for dynamic configurations, error handling, and robust data transformation rules to accommodate the diverse range of healthcare data.

The functional requirements presented in this section are derived from established guidelines and best practices outlined in the Book of OHDSI [19], which defines the structure and conventions of the OMOP CDM. Additional requirements are informed by recent scientific literature that explores metadata-driven ETL approaches [65] and real-world implementations of OMOP CDM transformations in clinical research settings [66]. These sources provide both theoretical foundations and practical insights to guide the development of a robust and standards-compliant ETL process.

The following functional requirements, as presented in Table 5.1, serve as the foundation for developing the ETL process for transforming healthcare data within the context of the OMOP CDM.

ID	Description	Source
FR01	The ETL process must extract data from source systems. Thus, it must connect to the RDA platform of the MedUni Vienna, which is modeled in the EAV format, and it must support the extraction of data from the RDA platform of the MedUni Vienna.	[19]
FR02	The system must map data from source tables to the appropriate OMOP CDM tables. The attributes of the EAV model must be mapped to the relevant columns of OMOP CDM tables.	[19]

ID	Description	Source
FR03	The system must transform the data from the EAV model to the OMOP CDM ensuring consistency between source data and OMOP CDM schema.	[19]
FR04	The mapping rules for EAV to OMOP CDM transformations must be loaded dynamically from a central repository or configuration file, which supports customizable mappings to allow adaptation to future changes in the data model.	[65]
FR05	The system must transform raw data to adhere to the OMOP CDM's data standards. It must ensure that all raw data values are standardized according to the OMOP CDM conventions.	[19]
FR06	The ETL process must ensure that the data is loaded into the appropriate OMOP CDM tables, maintaining the required schema and table structure. Each OMOP CDM table must be populated with data from the source system in the proper format.	[19]
FR07	The system must handle inconsistent or missing data in the source EAV model according to predefined rules.	[19]
FR08	The system must dynamically load metadata from a central repository or configuration file, which contains detailed mappings of which data should be extracted, transformed, and loaded into the OMOP CDM schema.	[65]
FR09	The system must set up an empty OMOP CDM database with the appropriate schema structure and relationships between tables before starting the ETL process.	[66]

Table 5.1: Functional requirements for the ETL process.

5.1.2 Non-functional Requirements

In developing a comprehensive ETL process to transform healthcare data from an EAV data model to the OMOP CDM, it is essential to define a set of specific goals that guide the overall development. These goals should ensure the ETL system is not only capable of handling complex healthcare data but also flexible, scalable, and efficient in its operations. Each goal aligns with key quality characteristics, as defined by Theodorou et al. [54]. The following goals outline the targeted attributes of the ETL process, with each goal accompanied by its corresponding quality characteristic from Theodorou et al.'s framework (noted in parentheses).

1. **Performance Optimization** (Performance)
   Optimizing the performance of the ETL process is necessary to minimize processing time and resource consumption. Techniques such as parallel and batch processing should be utilized to enhance the efficiency of data transformations and ensure timely results.

2. **Modularity and Reusability** (Maintainability)
   The ETL process should be modular, allowing for individual components to be updated, maintained, or reused without impacting the entire system. This approach facilitates both extensibility and maintainability of the code base over time.

3. **Configurability, Adaptability, and Flexibility** (Adaptability)
   Given the variety and evolving nature of healthcare data, the ETL process must be highly configurable and adaptable. It should allow for easy adjustments to accommodate different data sources, formats, and schema changes without requiring significant modifications to the underlying code. This flexibility is crucial for ensuring that the system can handle diverse datasets and can be reused across various use cases.

4. **Scalability** (Adaptability)
   With the growing volume of healthcare data, scalability is a crucial goal. The ETL process must scale efficiently to handle large datasets, ensuring consistent performance even as data volumes increase.

5. **Error Handling and Logging** (Fault tolerance and Auditability)
   Comprehensive error handling and logging mechanisms are required to ensure the system can recover from failures, trace issues, and maintain a detailed audit trail of ETL operations. These mechanisms will be critical for debugging, monitoring, and ensuring data quality throughout the ETL pipeline.

6. **Integration with the EAV Data Model** (Data quality)
   The ETL process must seamlessly integrate with the EAV data model. Handling the specific characteristics of the EAV model, such as pivoting and managing sparse data, is essential to ensure accurate transformation into the OMOP CDM.

7. **Automation and Scheduling** (Usability)
   Automating the ETL process through scheduling tools is essential for ensuring regular and consistent data updates without manual intervention. This goal ensures that ETL tasks can be executed at scheduled intervals or triggered by specific events.

To achieve these goals, a set of requirements has been defined, as presented in Table 5.2. These requirements translate the overarching goals into specific, actionable criteria that directly inform the ETL system's development and implementation. Each requirement is numbered to reflect its associated goal, with the first number indicating the goal and the second serving as a counter within that goal.

ID	Description
NFR1.1	Implement optimizations like indexing, caching, or efficient memory management to improve the overall performance of the ETL process.

ID	Description
NFR1.2	Leverage parallel and batch processing for data extraction, transformation, and loading to reduce processing time.
NFR2.1	The ETL process must be broken down into independent modules, each responsible for specific subtasks.
NFR2.2	Components like mapping functions and transformation logic should be designed to be reusable across different datasets and transformations.
NFR2.3	Ensure that updates to individual modules do not require changes to the entire ETL process, enabling maintenance and enhancements.
NFR3.1	Parameters, configuration files, or external sources should be used to define ETL settings, such as data mappings, transformations, and schema information, instead of hard-coding them.
NFR3.2	Attribute names and transformation rules should be loaded dynamically from metadata repositories or configuration files to accommodate schema changes and new datasets.
NFR3.3	The system should be capable of adapting to different schema structures without requiring major modifications, ensuring its applicability across various healthcare datasets.
NFR3.4	Implement a flexible, user-defined mapping mechanism for vocabulary mapping, avoiding fixed, hard-coded mappings.
NFR3.5	Utilize a metadata repository to store information about attributes, concepts, and relationships, ensuring this information can be easily updated and reused.
NFR4.1	Ensure that the ETL process can handle large-scale datasets by incorporating batch processing, parallel processing, or distributed computing.
NFR4.2	Incorporate robust error-handling mechanisms that catch errors, log them, and enable the process to resume or restart from failure points.
NFR4.3	Maintain detailed logs of ETL operations, including processing times, errors, successes, and resource usage, for auditing and performance tracking.
NFR4.4	Implement automatic recovery mechanisms to resume operations from the point of failure, reducing the need for manual intervention.
NFR5.1	The ETL process must be capable of processing EAV data, including the ability to pivot and unpivot data as needed to transform it into the OMOP CDM format.
NFR5.2	Handle specific EAV challenges, such as sparse data and multiple attribute types, while ensuring accurate data representation in the target CDM.
NFR6.1	Integrate with scheduling tools to automate the execution of ETL tasks at predefined intervals or in response to events.

Table 5.2: Non-functional requirements for the ETL process.

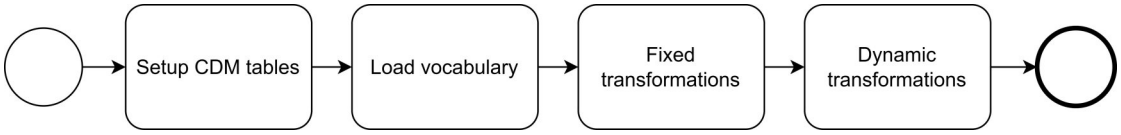


Figure 5.1: Conceptual model of the master ETL job as BPMN model.

5.2 Implementation Concept

The Implementation Concept section outlines the core approach to the ETL process used to transform healthcare data from the EAV model of the MedUni Vienna into the OMOP CDM. This section provides an overview of the key design principles and the underlying architecture that support the ETL pipeline.

This section will detail the specific transformations involved, the process flow, and the architecture that supports the integration of healthcare data into the OMOP CDM, providing a comprehensive understanding of how the generic ETL framework was conceptualized and developed.

Throughout the design of the ETL process, input from a domain expert specializing in electronic health records, interoperability, and health information management informed design decisions. Architectural choices were iteratively reviewed and adjusted based on expert feedback to ensure accuracy, compliance with clinical data standards, and practical feasibility within the research database environment. This collaborative review process helped align the ETL implementation with both functional objectives and domain-specific constraints, ensuring that the prototype reflects best practices in healthcare data integration.

5.2.1 Overview of the ETL process

The ETL process is organized into a single job, which coordinates multiple transformations. This structure enables the entire ETL process to be executed in a single step. Separating the different transformation steps among others by OMOP CDM table ensures that each transformation can be implemented independently, preventing interference between them and making the process easier to maintain and extend.

The master ETL job is the overarching job that manages the entire ETL pipeline. Figure 5.1 shows a schematic representation of the master ETL job. It starts by setting up the schema of the OMOP CDM and loading the vocabulary, followed by invoking the individual transformations for each OMOP CDM table. The ETL job runs sequentially, with each transformation executing in a predefined order.

Before any data transformation begins, the OMOP CDM schema is set up in the target database. This step involves creating all the necessary tables and constraints required by the OMOP CDM specification. A crucial step for OMOP CDM compliance is loading the necessary vocabulary tables that standardize terminologies and codes used in healthcare data. The vocabulary loading includes populating the vocabulary, concept_class, domain,

relationship, concept, concept_ancestor, concept_relationship, concept_synonym, and drug_strength tables with the vocabulary data provided by OHDSI. The vocabulary must be loaded into the OMOP CDM database early in the process to ensure consistency across the subsequent transformations. For the most part, custom vocabulary mappings are not needed, as the required mappings are embedded in the source data. However, a custom mapping is necessary for the gender of a person, as the specific mapping of the RDA platform of the MedUni Vienna does not exist in the OHDSI standardized vocabularies. These custom mappings are stored in the source_to_concept_map table. They are loaded from a CSV file, along with the corresponding entries for the vocabulary and concept tables after the OHDSI vocabularies have been loaded.

Each data transformation in the job handles a specific table within the OMOP CDM. The order of the OMOP CDM tables is as shown in Table 5.3 to ensure the original entry is created before it is referenced in a different table.

#	Table	Description	Transf.	Section
1	Care site	Lists institutional (physical or organizational) units where healthcare services are provided (e.g., offices, wards, hospitals, clinics, etc.) [27].	Fixed	5.2.2
2	Person	Contains unique records for each person, including key demographic information, and serves as central identity management for all Persons in the database [27].	Fixed	5.2.2
3	Death	Captures when and how a person died, based on available clinical or administrative data [27].	Fixed	5.2.2
4	Visit occurrence	Records high-level healthcare encounters (e.g., outpatient visits, hospital stays) [27].	Fixed	5.2.2
5	Visit detail	Represents detailed parts of a visit, such as ward movements or claim lines, linked to a visit occurrence [27].	Fixed	5.2.2
6	Specimen	Stores records of biological samples collected from persons [27].	Custom	5.2.3
7	Measurement	Contains structured test results or assessments (e.g., lab results, vitals), often with numeric or categorical values [27].	Custom	5.2.3

#	Table	Description	Transf.	Section
8	Observation	Captures clinical facts not stored elsewhere, such as lifestyle factors or family history [27].	Custom	5.2.3
9	Condition occurrence	Records diagnoses or symptoms indicating the presence of a medical condition [27].	Custom	5.2.3
10	Drug exposure	Describes exposure to medications or vaccines, including both prescribed and over-the-counter drugs [27].	Custom	5.2.3
11	Procedure occurrence	Tracks medical procedures performed for diagnostic or therapeutic purposes [27].	Custom	5.2.3
12	Fact relationship	Defines relationships between records across or within CDM tables (e.g., procedure–device, drug–condition) [27].	Fixed	5.2.2
13	Observation period	Defines periods during which a person's clinical events are expected to be recorded [27].	Fixed	5.2.2

Table 5.3: Order of the OMOP CDM tables in the ETL job.

The process begins by querying the source tables to extract the data needed for a specific OMOP CDM table, which is typically stored in the EAV model. Data extracted in this step is then mapped, transformed, and cleaned to align with the data types, constraints, and standards defined by the OMOP CDM. Common transformations include

- Vocabulary lookups,
- Foreign key lookups,
- Date splitting,
- Adding of constants, and
- Data deduplication.

After the data has been transformed, it is loaded into the corresponding OMOP CDM tables.

Load operations are performed in batch mode, ensuring high performance and minimal impact on the target database.

Depending on the target OMOP CDM table, the transformation process can either be fixed or custom. Table 5.3 summarizes for each table of the OMOP CDM if the

transformation is fixed or custom. A fixed ETL process refers to a transformation approach where the transformation logic is hard-coded and remains consistent across all use cases, regardless of their data set. This approach is used for OMOP CDM tables where the transformation is highly tailored to the target table, and the source data does not vary between use cases. The transformation steps are predefined, and the rules do not change based on the data processed or the specific use case.

The following tables allow a fixed ETL process.

- Observation_period
- Person
- Death
- Visit_occurrence
- Visit_detail
- Care_site
- Fact_relationship

In contrast, a dynamic ETL process introduces flexibility by adapting the transformation logic based on metadata associated with the data being processed. The transformation rules are not hard-coded; instead, they are influenced by metadata loaded from the data source, which may vary depending on the specific use case. This metadata-driven approach enables the ETL process to adjust the transformation logic to meet the unique characteristics of each data set. The following tables need a custom ETL process.

- Specimen
- Measurement
- Observation
- Condition_occurrence
- Drug_exposure
- Procedure_occurrence

5.2.2 Fixed Transformations

In this section, the fixed ETL transformations are described in detail. In these transformations, the transformation logic is predefined and consistent across all use cases. These transformations are designed to ensure that data is transformed regardless of the use case. The following sections will provide a detailed description of each fixed transformation, outlining the specific rules, logic, and steps involved in transforming the source data into the target OMOP CDM tables.

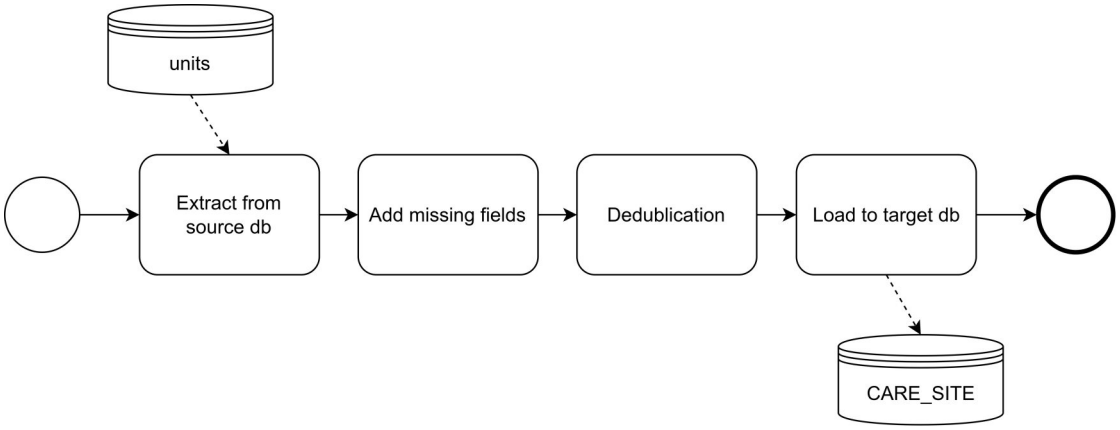


Figure 5.2: Conceptual model of the ETL process for the care_site table as BPMN model.

Care_Site

The source data for the care_site table originates from the units table of the RDA platform. This table contains information about operational sites of the AKH. It contains the following relevant columns for this transformation:

- A unique identifier for each operational site.
- A name of the operational site.
- A code indicating the type of the operational site.

The transformation process involves mapping these source fields to their corresponding target fields in the care_site table within the OMOP CDM.

Only records associated with the AKH, either at the overall level or at the level of specific care units such as wards or operating rooms, are included in the transformation. These categories were chosen because they represent physical locations where patient care is delivered, which matches the intended semantics of the care_site table. Other types of units, such as administrative departments, laboratories, or support services, do not constitute direct care sites and are therefore excluded from the transformation.

The transformation process to populate the care_site table follows a well-defined workflow. Figure 5.2 shows a schematic representation of the workflow.

1. Relevant records are extracted from the units table by filtering the data to include only those representing the AKH, wards, or operating rooms.
2. The place_of_service_concept_id is added to each record.

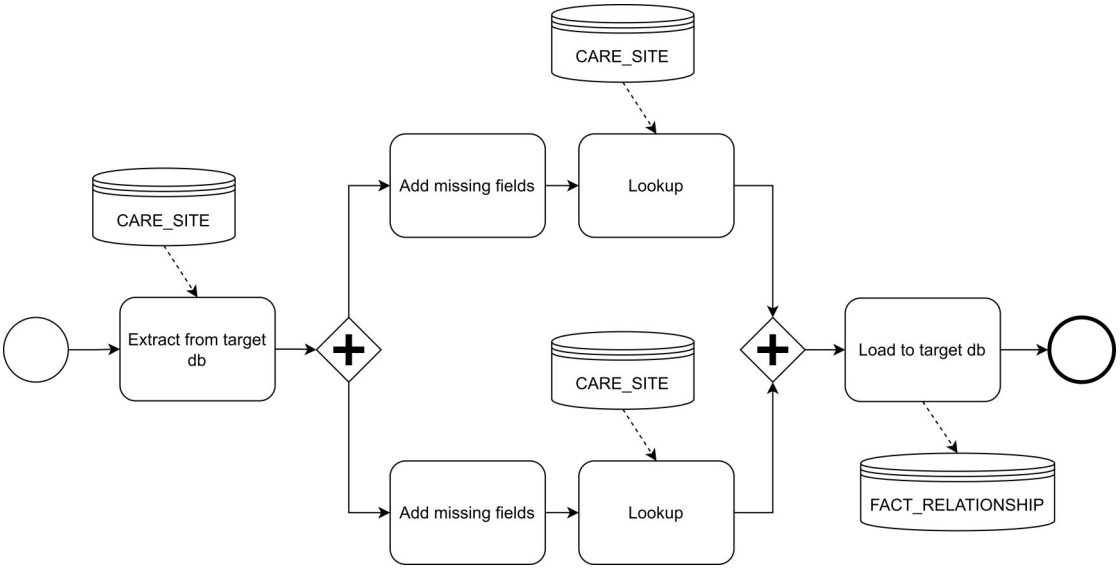


Figure 5.3: Conceptual model of the ETL process for the fact_relationship table as BPMN model.

3. Before the data is inserted into the care_site table, it is validated to ensure that each row is unique. This check ensures that no duplicates exist in the target table and that each unit is correctly represented.

4. After validation, the transformed data is loaded into the care_site table.

The relationship between the wards/operation rooms and the AKH is added in a second step. The records describing this relationship are stored in the fact_relationship table.

Fact_Relationship

The fact_relationship table holds the relationship between the wards/operation rooms, and the AKH. For each record in the care_site table, two relationships are created to describe the bidirectional relationship between the wards/operating rooms and the AKH. The wards and operating rooms are part of the AKH, which in turn contains the wards and operating rooms.

The transformation process to populate the fact_relationship table follows a well-defined workflow. Figure 5.3 shows a schematic representation of the workflow.

1. Relevant records are extracted from the care_site table by filtering the data to include only the wards and operation rooms and not the AKH.

2. For each ward or operating room, two records are created to represent the bidirectional relationship. One record will describe the relationship where the wards and

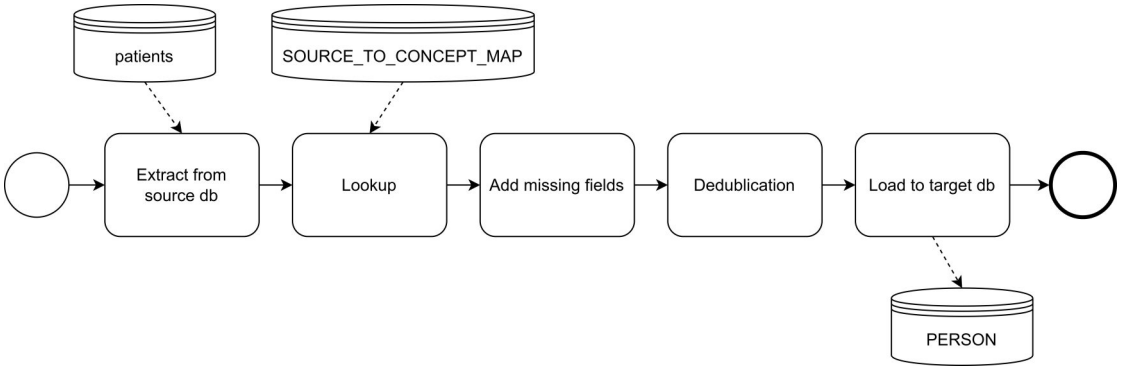


Figure 5.4: Conceptual model of the ETL process for the person table as BPMN model.

operating rooms are part of the AKH, while the second will describe the reverse, where the AKH contains the wards and operating rooms.

3. The care_site_id corresponding to the AKH is assigned to each of the duplicated records, with the correct care site identified through the care_site_source_value.

4. For each relationship record, the relevant domain_concept_id_1, domain_concept_id_2, and relationship_concept_id are added.

5. Finally, the transformed data is loaded into the fact_relationship table.

Person

The source data for the person table is derived from the patient table in the RDA platform. This table contains information about all patients, including demographic details such as sex and date of birth. The relevant columns for this transformation are as follows:

- A unique identifier for each patient.
- The sex of the patient.
- The birth date of the patient.

The transformation process to populate the person table follows a well-defined workflow. Figure 5.4 shows a schematic representation of the workflow.

1. Relevant records are extracted from the patient table, including the necessary fields.

2. The sex field is mapped to the corresponding concept ID in the OMOP CDM using a predefined mapping.

3. The birth date field will be split into separate year, month, and day components.

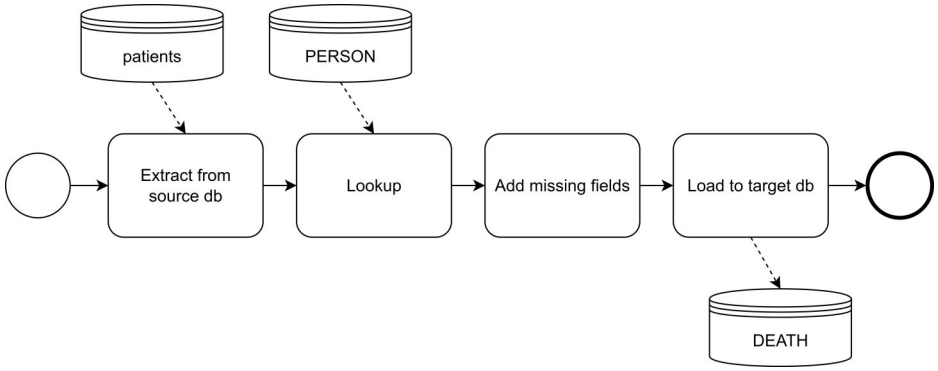


Figure 5.5: Conceptual model of the ETL process for the death table as BPMN model.

4. Because the patient table lacks information on ethnicity and race, while these fields are required in the OMOP CDM, the transformation process assigns constant values that explicitly denote the absence of mapped concepts.

5. Before the transformed data is inserted into the person table, a validation step is performed to ensure that each row is unique. This validation prevents duplicate entries in the target table and ensures that each patient is correctly represented in the OMOP CDM dataset.

6. After validation, the transformed data is loaded into the person table.

Death

The source data for the death table is derived from the patient table in the RDA platform. This table contains patient records, including details about their date of death. The relevant columns for this transformation are as follows:

- A unique identifier for each patient.
- The death date of the patient.

The transformation process to populate the death table follows a well-defined workflow. Figure 5.5 shows a schematic representation of the workflow.

1. Relevant records are extracted from the patient table, including the necessary fields. Only records with a present death date are included.

2. The rows are mapped to the corresponding row from the person table. The reference is achieved by locating the patient's person_id in the person table via the person_source_value field. The person_id is then used to link the patient's death information to the correct patient in the death table.

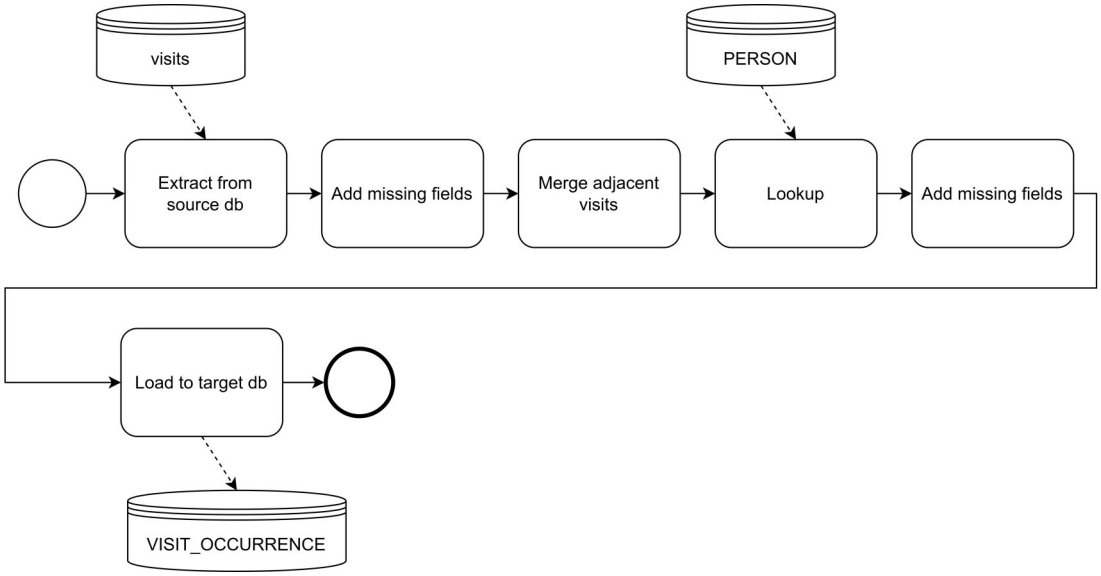


Figure 5.6: Conceptual model of the ETL process for the visit_occurrence table as BPMN model.

3. The death_type_concept_id is added. A predefined constant value will be used to represent the type of death, as no specific categorization is available in the source data.

4. The death_date is derived from the death_datetime.

5. The transformed data is loaded into the death table.

Visit_Occurrence

The source data for the visit_occurrence table is derived from the visits table in the RDA platform. Only inpatient stays are considered. This table contains information about hospital stays, detailing patient admissions and discharges. The relevant columns for this transformation are as follows:

- A unique identifier for each hospital stay.
- A foreign key referencing the patient associated with the hospital stay.
- The admission date of the patient to the hospital.
- The discharge date of the patient from the hospital.

The transformation process to populate the visit_occurrence table follows a well-defined workflow. Figure 5.6 shows a schematic representation of the workflow.

1. Relevant records are extracted from the visits table.
2. The visit_start_date and visit_end_date are derived from the admission date and discharge date, respectively.
3. The row is mapped to the corresponding person_id from the person table. The reference is achieved by locating the patient's person_id in the person table using the person_source_value field. This person_id is then used to link the patient's visit information to the correct entry in the visit_occurrence table.
4. The visit_concept_id and visit_type_concept_id are added based on predefined concepts that categorize the type of visit.
5. The care_site_id for the AKH is added to the records.
6. The transformed data is loaded into the visit_occurrence table.

Visit_Detail

The source data for the visit_detail table is derived from the movement table in the RDA platform. Only inpatient ward movements are considered. This table contains information about patient ward movements, such as admissions, transfers, and discharges. The relevant columns for this transformation are as follows:

- A unique identifier of the ward movement record.
- A foreign key linking to the corresponding hospital stay.
- A classification indicating the type of movement, such as admission, transfer, or discharge.
- The date of the movement.
- The ward where the movement occurred.

The transformation process to populate the visit_detail table follows a well-defined workflow. Figure 5.7 shows a schematic representation of the workflow.

1. Relevant records are extracted from the movement table.
2. Since the source data only contains the dates of the movements and not the entire duration of the stay, each transfer record is duplicated to serve as the end and start date for a stay.
3. After duplication, the records are sorted in chronological order. Adjacent movements (e.g., an admission followed by a transfer or discharge) are merged into a single stay, and the visit_detail_start_date and visit_detail_end_date are derived from the movement dates.

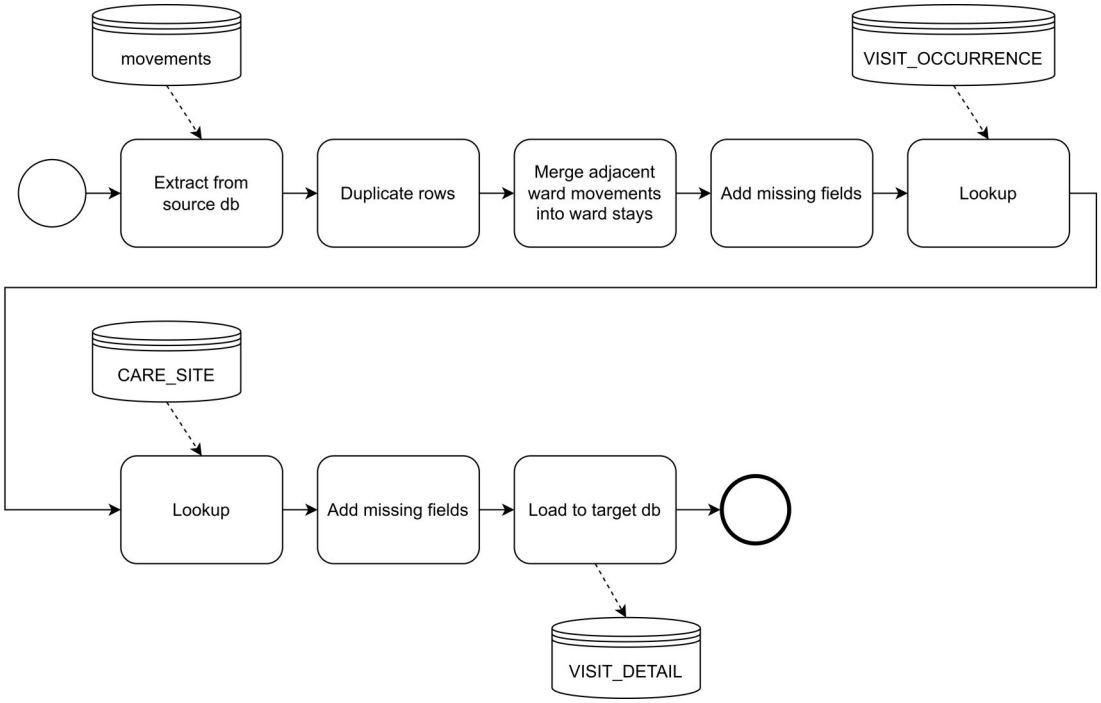


Figure 5.7: Conceptual model of the ETL process for the visit_detail table as BPMN model.

4. The corresponding visit_occurrence is derived.
5. The unit field is used to derive the care_site_id.
6. The visit_detail_concept_id and visit_detail_type_concept_id are added based on predefined concepts that categorize the type of movement.
7. The transformed data is loaded into the visit_detail table.

Observation_Period

Each Person must have at least one Observation_Period record, representing a time interval with a high likelihood of capturing Clinical Events [27].

In many ETL processes, the start date of the first occurrence or first high-quality occurrence of a Clinical Event (such as Condition, Drug, Procedure, Device, Measurement, or Visit) is used as the observation_period_start_date. Similarly, the end date of the last occurrence of a Clinical Event, the last high-quality occurrence, or the end of the database period is assigned as the observation_period_end_date for each Person [27].

Since Observation Periods are often not explicitly defined in source data, they must be inferred. In such cases [27]:

- The observation_period_start_date is set to the earliest available event date for a given Person.
- The observation_period_end_date is set to the latest available event date.

According to the THEMIS convention, the observation_period_end_date should be assigned as the earliest of the following [67]:

- Date of death + 60 days: This allows for post-mortem events (e.g., autopsy reports, final notes).
- Last clinical event + 60 days: Based on the assumption that a patient would return to the same healthcare provider in case of complications or unresolved conditions.
- Date of the data pull from the system

The source data for the observation_period table is derived from multiple clinical tables, including death, visit_occurrence, specimen, measurement, condition_occurrence, drug_exposure, and procedure_occurrence. The relevant attributes are person_id and the associated event dates. The earliest and latest dates are identified and mapped to the observation_period table in the OMOP CDM.

The transformation process to populate the observation_period table follows a well-defined workflow. Figure 5.8 shows a schematic representation of the workflow.

1. All persons are retrieved from the person table.
2. If available, death records are extracted for each person.
3. The minimal and maximal dates from each clinical table are determined for every person.
4. The minimal and maximal dates from the visit_occurrence table are extracted for each person.
5. The final minimal and maximal dates are calculated based on the predefined rules.
6. The data is filtered to include only records where at least one of the date fields, either the start date or the end date, is present. Like this, patients without any recorded clinical events are excluded from the transformation.
7. The period_type_concept_id is assigned based on predefined concepts.
8. The transformed data is loaded into the observation_period table.

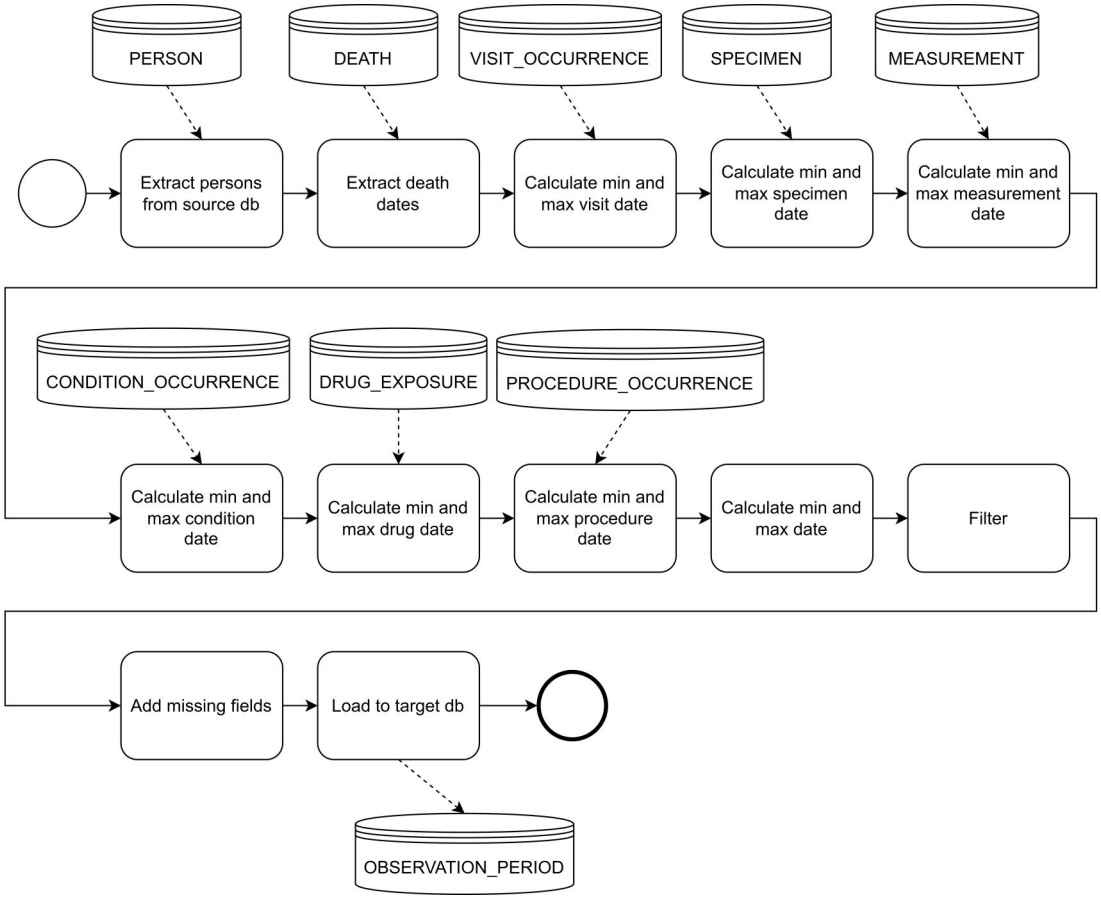


Figure 5.8: Conceptual model of the ETL process for the observation_period table as BPMN model.

5.2.3 Custom transformations

In this section, the custom ETL transformations are described in detail. Unlike the fixed transformations, the logic in the custom transformations is metadata-driven and adaptable across diverse use cases. These transformations enable flexible mapping of source data to OMOP CDM tables using template-specific configurations. The following sections detail each step of the custom transformation process, including how metadata guides the logic, the transformation rules, and the loading of the data into the target OMOP CDM structures.

Central to this approach is the concept of template-based transformation, where each input attribute is associated with a template representing a specific OMOP CDM record type. These templates define the structure of the target record and are stored in a dedicated mapping table. This setup allows the transformation logic to adapt dynamically to different record types at runtime without relying on hardcoded rules.

This metadata-driven strategy significantly reduces implementation complexity and enhances maintainability. New data sources or changes in data structure can be accommodated by simply updating or extending the mapping table, eliminating the need to modify the underlying transformation logic.

The structure and function of the mapping table are described in the following subsection.

Structural Mapping

To facilitate the structural mapping of data from the RDA platform to the OMOP CDM, an additional mapping table is introduced. This table defines the structural relationship between fields in the source data and their corresponding attributes in the OMOP CDM.

The mapping table serves as a central configuration resource for the custom transformations. It specifies, for each field in a source form, where its data should be placed within the OMOP CDM and how it contributes to a target record. This structural mapping includes identifying the correct OMOP CDM table and column, as well as grouping related source fields into coherent target records.

The structure of the mapping table includes the following fields, as shown in Table 5.4.

Template ID	Unique identifier for the template used to process the current part of a form.
Source form ID	Foreign key to the source form that provides the data.
Source attribute ID	Foreign key to the specific form field (attribute) within the source form.
Source field	Identifier of the field in the source data that contains the value. The source fields are possible: • The original value stored in the RDA platform. • The concept ID of the code associated with the value set entry corresponding to the original value stored in the RDA platform. • The standard concept associated with the value set entry corresponding to the original value stored in the RDA platform. • The value associated with the value set entry corresponding to the original value stored in the RDA platform. • The original attribute stored in the RDA platform. • The concept ID of the code associated with the attribute. • The standard concept associated with the attribute. • The original unit stored in the RDA platform.

Source field Contin.	• The concept ID of the code associated with the unit corresponding to the original value stored in the RDA platform. • The standard concept associated with the unit corresponding to the original value stored in the RDA platform. • Use the fixed value specified in the structural mapping table.
Target table	The OMOP CDM table where the data will be inserted.
Target field	The target field in the OMOP CDM table where the data will be inserted.
Fixed value	A constant that should be assigned to the attribute in the target instead of a value-specific semantic mapping based on the source data.
Use default mapping	A flag indicating if the default mapping should be used for this form position.
Referenced template ID	Foreign key to the template ID from which the related record originates and to which this record should be linked (for record linking via event fields).

Table 5.4: Structure of the mapping table for the custom transformations.

The following constraints are enforced to ensure the integrity and clarity of the mappings:

- All rows describing the same target record need to have the same template ID.
- Within a single template, each target field may only occur once. The template ID and target field are unique in combination.
- Within a single template, the target table has to be the same for all records.
- Within a single template, the source form has to be the same for all records.
- The target table has to match an OMOP CDM table name.
- The target field has to be a column of the OMOP CDM table specified in the target table.
- If a fixed value is assigned for the semantic mapping, the column source field needs to indicate that the fixed value should be used. Otherwise, the fixed value column needs to be empty.
- The following columns have to be not null: template id, source form id, source attribute id, and target table.

This structured approach to mapping enables a clear separation of transformation logic and mapping configuration. It enables flexible adaptation to new data structures and supports the automated validation of mapping rules.

In addition to explicitly defined mappings, a set of default mappings is defined for the standard case. These defaults represent common and recurring field mappings that apply across most forms and use cases. The column “use_default_mapping” is used to indicate whether the default mapping should be used for a specific form position. In this case, only the target table needs to be specified.

This mechanism ensures that the mapping table only needs to capture exceptions or custom mappings, significantly reducing its size and complexity. At the same time, it allows flexibility as any default behavior can be overwritten by adding a corresponding entry to the mapping table, giving precedence to explicitly defined mappings when both exist. Additionally, the default mappings can be extended by custom mappings to fit the specific use case.

This layered mapping strategy combines scalability and customizability, streamlining the transformation process while maintaining complete control over specific edge cases.

In the default case, the mapping logic relies on the metadata associated with the form field to derive the necessary target values for the OMOP CDM.

The terminology code linked to the form field is used to determine the appropriate concept_id for the specimen, measurement, observation, condition_occurrence, or procedure_occurrence. The concept id associated with this terminology code is also stored in the source_concept_id column to preserve the original semantic identifier. The human-readable label of the form field, which provides a descriptive name from the source system, is stored in the source_value column.

The document date is used to populate the date and datetime fields for the record. The type_concept_id is set to the standard concept representing “EHR”.

For fields with numeric or quantitative values, the unit is derived using the standardized terminology code associated with the form field’s unit definition. This value is used to determine the unit_concept_id, while the concept id associated with the original unit code and the label are stored in unit_source_concept_id and unit_source_value, respectively.

The person_id is resolved via the patient identifier, which is recorded in the person_source_value.

If the OMOP CDM table includes a value field, the value_as_number is derived from the numerical value column of the document field. The value_source_value is derived from the textual value column of the document field. In cases where a value_as_concept_id is needed, the form field must be linked to a value set. This set contains standardized terminology codes for each possible value of the form field, which are then mapped to concept IDs. Further details on how values are semantically mapped are provided in the following subsection.

To capture visit context, the visit_occurrence_id and visit_detail_id are derived based on the inpatient stay and ward stay associated with the document. If no explicit ward stay is linked, the correct stay is inferred by comparing the document date with the date range of available ward stays.

Semantic Mapping

To enable semantic interoperability between the RDA platform and the OMOP CDM, data elements from the RDA platform must be aligned with standardized terminologies such as SNOMED CT, LOINC, or ICD-10. Since the RDA platform does not natively rely on standardized vocabularies, an additional semantic mapping layer is introduced. This mapping layer defines how various RDA platform elements correspond to OHDSI standard concepts and enables consistent, automated transformation into the target CDM.

A key advantage of this approach is that mappings are stored directly within the source database. As a result, no separate mapping step is required during data integration into the OMOP CDM, which significantly reduces the complexity of the ETL process and improves efficiency by shifting terminology alignment to the source level rather than embedding it in transformation logic.

To store the semantic mapping in a structured and maintainable way, a set of dedicated mapping tables is introduced into the RDA platform. Each table references a specific type of RDA platform database object, such as form fields, value sets, or units, and associates each entry with a standardized terminology code and its corresponding vocabulary.

Instead of storing this semantic information directly in the existing core tables, it is stored in separate mapping tables. This design offers several advantages. First, it ensures a clear separation of concerns: structural definitions remain in the core tables, while semantic information is encapsulated elsewhere. It also supports the association of multiple vocabularies with the same source element, enabling more flexible mappings. Furthermore, semantic mappings can evolve independently of the structural metadata, allowing updates or extensions without requiring modifications to the core schema. The separation also improves validation and maintainability, as constraints, data quality checks, and indexing can be applied more easily.

Separate mapping tables are preferred over a unified table because they offer a more transparent structure and allow for simpler validation, indexing, and extension. By assigning each object type to its table, the mapping remains clear, semantically accurate, and easier to maintain over time.

Each record in the mapping tables contains a foreign key to the corresponding RDA platform object, along with a standardized terminology code and the vocabulary it belongs to. The value and valueVocabulary fields are used together to determine the concept_id from the OHDSI standardized vocabularies. The vocabulary must be a valid OHDSI vocabulary (e.g., SNOMED CT, LOINC, ICD-10). In the context of this ETL

process, the necessary semantic mapping tables are for the form positions, value sets, and units.

For form fields linked to a value set, the value assigned in the document is matched against entries in that set. The matching also includes synonyms, which are defined in the attributes column of the value set item. This design enables robust handling of typos and alternative spellings.

In some cases, semantic meaning is not expressed explicitly but inferred, e.g., the presence of a microorganism may be implied by a single entry without an explicit "positive" flag. Such implicit mappings are handled using the mapping to a second value stored in the attributes of a value set entry. This additional data allows the system to resolve the combined semantic meaning from minimal data.

As part of a proposed extension, the semantic metadata for value sets will be stored using JSON in the attributes column of the value set entries. This approach has not yet been implemented, but is planned to support a wide range of use cases in a flexible and extensible manner. By adopting a structured JSON format, the system will be able to represent rich semantic details, such as synonyms, vocabulary codes, and context-specific mappings, within a single column, without requiring additional schema changes.

To ensure data quality, the proposed JSON structure will be validated using a defined JSON Schema. This validation guarantees that the structure and data types conform to expected formats, that all required fields are present, and that records remain consistent across the dataset. Additionally, the schema serves as self-documentation, supporting both developers and maintainers in understanding and managing the data structure.

A typical JSON object may include the relevant fields “synonyms”, “value”, and “valueVocabulary”, and can be extended by use-case-specific fields that are ignored by the standard ETL logic. The "synonyms" field is optional and may contain a list of alternative terms or spellings. The "value" and "valueVocabulary" fields can be used to represent an associated result concept, such as positive or negative outcomes in the case of a measurement. An exemplary JSON document for the attributes column is provided below in Listing 5.1.

Listing 5.1: Sample data Microorganisms.

```
{
  "synonyms": [
    "E. coli",
    "Colibacillus",
    "Bacterium coli"
  ],
  "value": "260373001",
  "valueVocabulary": "SNOMED"
}
```

Listing 5.1: Sample data Microorganisms.

To validate this structure, the following JSON Schema is applied, as shown in Listing 5.2.

Listing 5.2: Schema Microorganisms.

```
{
  "$id": "http://meduniwien.ac.at/microorganism.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Microorganism",
  "type": "object",
  "properties": {
    "synonyms": {
      "type": "array",
      "description": "A list of synonyms for the organism.",
      "items": {
        "type": "string"
      },
      "minItems": 0,
      "uniqueItems": true
    },
    "value": {
      "type": "string",
      "description": "The unique code for the value concept."
    },
    "valueVocabulary": {
      "type": "string",
      "description": "The vocabulary for the value.",
      "const": "SNOMED"
    }
  },
  "required": [
    "value",
    "valueVocabulary",
    "isCSC"
  ],
  "additionalProperties": false
}
```

Listing 5.2: Schema Microorganisms.

This approach ensures that terminology metadata is not only semantically aligned with the OMOP CDM but also structurally validated and future-proof. By combining JSON and JSON Schema, the RDA platform achieves a balance of flexibility, standardization, and maintainability in its semantic mapping strategy.

In situations where the correct concept depends on the combination of two form values, an external mapping file is used. This file defines valid value combinations and their

fomu	fopo	text	fopo-x	fopo-x-text	target	target-vocab
105	2	Blut	3	Kultur	446131002	SNOMED CT
105	2	Blut	3	Bakterielle Breitspektrum-PCR	119297000	SNOMED CT

Table 5.5: Example for a semantic mapping where the correct concept depends on the combination of two values.

associated standardized codes. During the ETL process, both input values are evaluated together to determine the correct concept_id. An example is provided in Table 5.5.

ETL process

For the ETL process, the following assumptions are made regarding how the relevant data is stored in the RDA platform.

1. If the semantics of a form field are needed in the ETL process, associated standardized terms and their corresponding vocabularies are stored in the semantic mapping tables for the form fields. There are standardized terms for both the semantic meaning and the unit of a numeric value (if applicable).

2. Each form field containing textual values is associated with a value set that defines a set of acceptable values for that field.

3. Each value in the value set is, in turn, associated with standardized terms and vocabularies stored in the mapping table for value sets, analogous to the form field itself.

4. If applicable, synonyms are stored in the attributes column of the value set entry to cover typos and alternative spellings.

5. If the value in the value set also requires a mapping to a value, for example, for measurements where the value in the value sets describes both the measurement and the result, but the data needs to be split in the OMOP CDM, the value to which this concept is mapped is also stored in the attributes of the value set entry.

6. Each form field references an item representation, which references an item type.

The input to the custom transformation is the template ID, which references a corresponding OMOP CDM record. For each template ID, the same sequence of transformation steps is executed. These steps are configured based on metadata retrieved using the template ID. Figure 5.9 illustrates the transformation steps performed for each template.

Figure 5.9: BPMN model of a custom transformation. The input to the custom transformation is the template ID, which represents a corresponding OMOP CDM record type. For each template ID, the same sequence of transformation steps is executed. These steps are configured based on metadata retrieved using the template ID.

1. The form fields, documents, values, and OMOP_Mapping tables are joined. Relevant records are selected from the joint tables by filtering the data using the template ID.

2. For each form position associated with a value set, the standardized code is extracted from the attributes column of the corresponding value set item. The correct record is identified by comparing the value assigned to the document position to all values in the value set, including any defined synonyms. Once a matching record is identified, the associated standardized code is retrieved. In cases where the mapping depends on a combination of two form positions, the value sets are not used to provide the semantic mapping. The mapping is provided in a separate CSV file. This file specifies combinations of values from both form positions and the corresponding target code. Finally, the concept_id for each assigned standardized code is derived.

3. If the attributes of the value set also include a mapping to a value, its corresponding concept_id is also derived.

4. The concept_id linked to the standardized code associated directly with the form field is derived.

5. The concept_id for the unit associated with the form field is derived.

6. If the derived concept IDs are not standard concepts, the associated standard concepts are derived.

7. The data is pivoted according to the mappings defined in the OMOP_Mapping table, along with the default mappings applicable to the target OMOP CDM table.

8. Identifiers such as person_id, visit_occurrence_id, and visit_detail_id are derived.

9. The date field is extracted from the datetime field by removing the time component.

10. To populate event fields, such as measurement_event_id or observation_event_id, related records are identified using the referenced template ID from the mapping table and the document ID of the current document. During the transformation process, a temporary lookup table is created containing the target OMOP CDM table, template ID, document ID, and primary key of each transformed record. This table is used to find and assign the correct related record to the event field of the current row.

11. A default type_concept_id is added, representing the provenance as “EHR”.

12. Columns required by the OMOP CDM but not present in the data are added and filled with NULL values.

13. The transformed data is loaded into the designated OMOP CDM target table.

The metadata required for the custom transformation process is queried based on the provided template ID and then injected into the respective transformation steps. This metadata provides the necessary configuration, allowing the ETL process implementation to remain generic and adapt the transformation logic to different templates and OMOP CDM record types without requiring hardcoded rules for each OMOP CDM table. The metadata includes:

- Structural mapping of form fields to the corresponding OMOP CDM fields, ensuring each source field is correctly aligned with its target.
- Default mappings for the target OMOP CDM table to fill in values when no custom mapping is defined.
- Source field identifiers, such as form field IDs, to locate the correct values in the source data.
- Data type information of the source data to determine how to interpret and process each field's value.
- The target OMOP CDM table and column names, specifying the schema for loading the transformed data.
- Value set references that link form fields to a predefined list of possible values, which are then semantically mapped to standardized concepts.
- Unit mappings that associate form fields with units and the corresponding standardized codes.

By organizing all this information in metadata, the system supports a highly flexible, reusable, and scalable ETL process that can adapt to new forms, templates, or mappings without manual intervention.

5.3 Prototypical Implementation

The following section presents the prototype developed to demonstrate and evaluate the ETL approach proposed in Section 5.2, which transforms healthcare data from the EAV-based research database of the MedUni Vienna into the OMOP CDM. Workflow sequences were implemented according to the design principles established in collaboration with the domain expert, ensuring alignment with functional objectives, clinical data standards, and domain-specific constraints. The section begins by outlining the selected technology stack, highlighting the rationale for choosing specific tools and frameworks in the context of the project's requirements. Subsequently, a detailed description of the implementation is provided. The prototype serves as a proof of concept to validate the feasibility of the generic ETL solution.

5.3.1 Technology Stack

As described in Section 3.5, this thesis focuses on open-source ETL tools due to licensing considerations facilitating the reuse and extension of the developed prototype. Opting for an existing ETL tool over developing an in-house tool also minimizes development overhead and reduces resources needed for performance optimization and long-term maintenance. Using a pre-existing ETL tool also facilitates the reuse of the developed prototype. The accessible and user-friendly nature of ETL tools makes it easier for non-technical users, such as healthcare professionals or researchers with limited programming skills, to leverage and reuse the prototype. This capability is crucial for fostering broader adoption and integration of the solution, as it lowers the barrier to entry and allows teams without coding expertise to benefit from the developed system.

The choice of PDI [59] as the ETL tool for this prototype was guided by several critical factors that ensured its suitability for the project's objectives. One of the key considerations was user accessibility, as the tool offers a highly intuitive graphical interface, making it especially advantageous for users with limited experience in data integration or ETL processes. This feature was essential because the ETL process implemented in the prototype was intended to be easily customizable and reusable for future projects. PDI's user-friendly design allows new users to quickly design, execute, and troubleshoot ETL workflows, reducing the learning curve typically associated with more complex tools.

In addition to its ease of use, PDI provides a comprehensive set of functionalities that address the full scope of the ETL process. The tool supports various connectors for different data sources and sinks, enabling complex transformations and including robust error-handling capabilities. These features ensured the seamless integration of diverse data formats and sources in the prototype, particularly those based on structured, relational data. Furthermore, PDI's open-source nature and active community provide valuable resources for resolving issues and sharing best practices, which can be particularly beneficial in a development and prototyping environment.

The decision to use an ETL tool rather than manually implementing data integration was driven by the need for an automated and scalable solution to handle data extraction, transformation, and loading. ETL tools offer significant advantages over manual processes, including built-in features for automation, data quality control, logging, and monitoring, all of which are essential for ensuring the integrity and efficiency of the data pipeline. Without such a tool, the ETL processes would have required extensive manual effort, increasing the likelihood of errors and introducing unnecessary complexity. By selecting PDI, the project benefited from a reliable, flexible, and extensible data integration platform that streamlined the entire process, allowing for a greater focus on the core objectives of the prototype.

SQLite [68] was selected as the development database due to several advantages. As a lightweight, serverless, self-contained database engine, it requires minimal setup, making it ideal for the development phase of the prototype. Its small footprint and simplicity allow for rapid testing without the overhead of more complex database systems. Additionally,

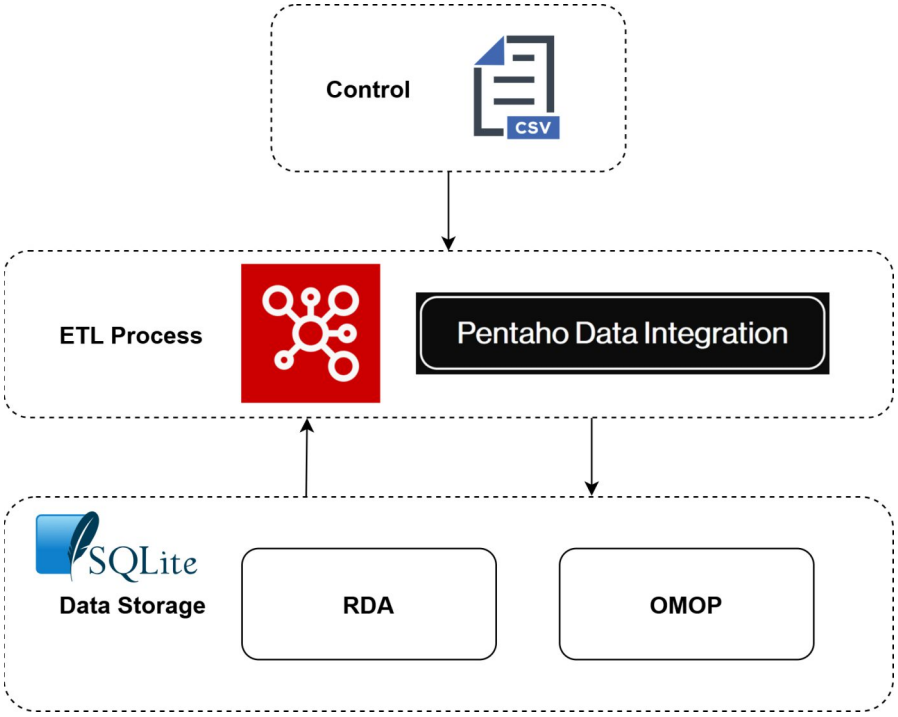


Figure 5.10: Architecture of the Prototype.

SQLite offers fast performance for small-scale datasets, which is well-suited for the scope of the test data. Its file-based nature makes it easy to store, share, and transfer the database while consuming minimal system resources compared to larger database management systems. Furthermore, SQLite fully supports SQL, enabling efficient querying and data manipulation, essential for the ETL process [69]. These factors made SQLite a practical and efficient choice for managing test data and facilitating the development of the prototype.

The technical architecture of the prototype follows a layered architecture model. This model is structured into three main layers: Control Layer, ETL Process Layer, and Data Storage Layer. The Control Layer is responsible for orchestrating the ETL processes by initiating data loads and managing configurations. The ETL Process Layer contains the core logic for data extraction, transformation, and loading. The Data Storage Layer manages the storage of both the source data and the transformed data. Figure 5.10 illustrates the architectural design of the prototype ETL system.

5.3.2 Implementation Details

The Data Integration perspective of PDI allows the creation of two primary file types: transformations and jobs. Transformations define the data flows for the ETL process, including data extraction from a source, applying transformations, and loading the processed data into the target location. Jobs orchestrate these ETL activities by managing

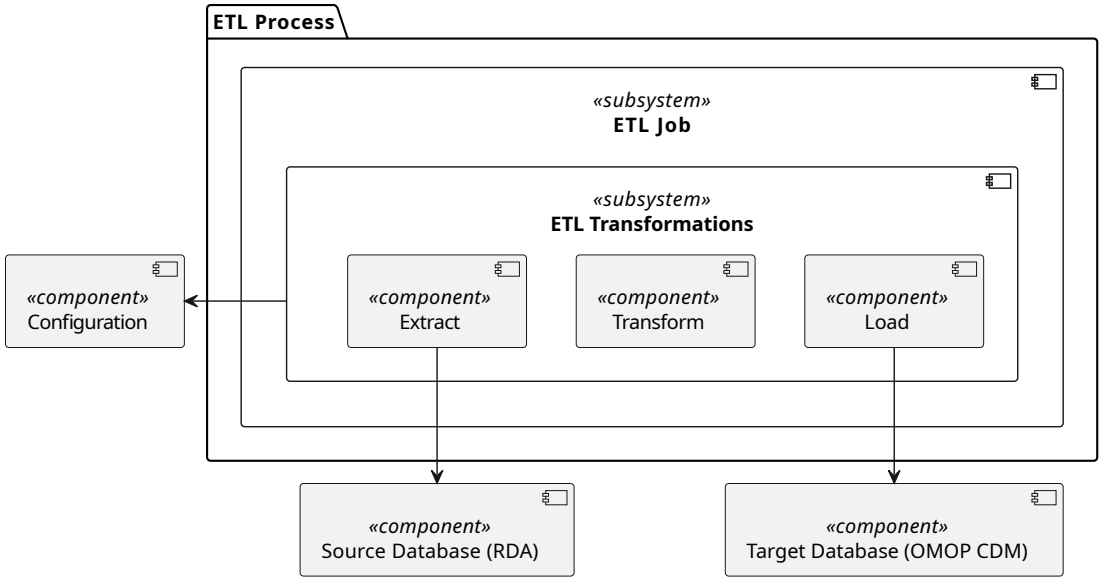


Figure 5.11: UML Component Diagram of the prototype architecture. The ETL Transformations subsystem is displayed as placeholder for all ETL Transformations.

the execution flow, handling dependencies between transformations, and incorporating control logic such as conditional execution, file existence checks, or table validation.

The ETL process in this prototype is governed by a master job, which orchestrates the entire pipeline in a modular and structured manner. This master job coordinates individual transformations, each responsible for specific sub-tasks. Such a modular design enhances maintainability and reusability by encapsulating discrete logic units within separate transformations. Figure 5.11 illustrates the component-based architecture of the prototype.

The master job initiates the process by setting up the OMOP CDM schema in the target database and executing the transformations for each OMOP CDM table in a sequential order. Each table-specific transformation is implemented in an independent transformation invoked in a predefined order to ensure data integrity and correct referencing. For example, core entities, such as Person, are created before dependent entities, like Visit Occurrence or Death. A screenshot of the implemented master job in PDI is provided in Figure 5.12.

To validate the functionality and test the prototype of the system, synthetic test data was generated to simulate real-world scenarios while adhering to the structure of the OMOP CDM. The administrative data was generated with the support of the generative AI chatbot ChatGPT [70], developed by OpenAI. Prompting ChatGPT with the required DDLs and table semantics enabled the creation of semantically consistent datasets that matched the expected data structure of the RDA platform. The clinical data were created in a guided process: candidate values were proposed with ChatGPT, curated, and, where

Figure 5.12: Main PDI job orchestrating the prototype's ETL workflow, including OMOP CDM setup, fixed transformations, and custom transformations.

necessary, manually adjusted to reflect realistic clinical scenarios while ensuring validity against the target OMOP CDM tables. Additionally, clinical data was manually created based on real-world use cases and the structure of the OMOP CDM tables targeted in the transformation process. The data includes representative examples of common clinical scenarios, covering a range of medical observations, procedures, and measurements. Although synthetic, the data was designed to reflect plausible clinical content and to support the end-to-end testing of the ETL pipeline. Mock-ups of the corresponding input forms are provided in Appendix A to illustrate the documentation structure and field definitions. The focus was not on building a statistically representative population but on providing sufficient coverage of typical cases to test the system end-to-end. This approach ensured the creation of synthetic yet semantically valid data, allowing the system to be tested against realistic data structures without using the real-world data intended to evaluate the ETL process.

Data was generated for all key tables in the RDA platform, as outlined below:

Patients: Data was generated for 25 unique patients, each with an ID, sex, birth date, and, where applicable, a death date. This patient data is designed to represent a diverse population, which is necessary for testing the system's performance and handling different patient characteristics. 25 patients were created to ensure variation across characteristics such as sex, presence or absence of a death date, and different combinations of inpatient and outpatient stays.

Wards: Data for 26 wards, including IDs, names, and types (e.g., standard wards labeled "st" and operating wards labeled "op"), were generated. The hospital was also included as a special ward within the data set. 26 wards were included to provide both standard and operating wards as well as the hospital entity itself, allowing transfer and movement data to be meaningfully represented.

Inpatient and Outpatient Stays: For each patient, both inpatient and outpatient stays were generated. Each stay received a unique ID and included attributes such as start and end dates, minimum and maximum dates, and the type (inpatient or outpatient). Inpatient stays were further enriched with an admission type. Each patient was linked to at least one inpatient and one outpatient stay, ensuring both pathways could be validated.

Movement Data: In addition to the stay data, movement data was generated to track patient admissions, discharges, and transfers within the hospital. This movement data includes unique IDs, links to the corresponding stay, the type of movement (e.g., admission, discharge, transfer), the date, and the ward involved in the movement.

Form positions: Data for form positions was created to define the structure and content of various clinical forms used in the test data. Each form position consisted of the form ID, form position ID, item display ID, form position name, and linked value set. The fields were designed to reflect realistic documentation requirements commonly found in clinical practice.

Documents: Clinical documents were generated to simulate filled-out forms for individual patients and cases. Each document includes a unique key, has a date, a reference to the corresponding form, and is associated with a specific patient and stay.

Document positions: For each document, document positions were created to record the values entered into individual form fields. Each entry includes the document reference, the form reference, the field reference, the patient reference, and the captured value.

Value set entries: Value set entries were created to define the accepted values for categorical fields within the form positions. Each value consists of a unique key, a reference to the value set, the value, and an attributes column holding a JSON object.

Item types: The available item types were created.

Item displays: The available item displays were created.

This deliberate coverage created a diverse but not overly complex test population, sufficient to validate data flows. The number of individual records was not driven by statistical requirements but rather by the need to cover a variety of combinations.

Setup of the OMOP Common Data Model

Before any data transformations occur, the PDI master job creates the OMOP CDM schema in the target database. This includes creating all the required tables and constraints according to the OMOP CDM specification. The schema setup is executed conditionally only if a predefined condition variable is set to true. This setup process is encapsulated in a separate job, which is then referenced by the master ETL job to ensure a clean and modular pipeline.

The OMOP CDM schema setup is executed through SQL scripts provided by the OHDSI GitHub repository [71]. These scripts define the database structure, including tables, relationships, and constraints, essential for organizing healthcare data in the OMOP format.

A critical component of the schema setup is loading the OHDSI standardized vocabulary, which is essential for standardizing the terminology and concepts used across the CDM. The vocabulary loading is handled in a separate job, which is called from the setup job. This separation ensures that the process remains clear and manageable. The vocabulary tables are loaded in the following order:

1. Vocabulary
2. Concept class
3. Domain

4. Relationship
5. Concept
6. Concept ancestor
7. Concept relationship
8. Concept synonym
9. Drug strength

For each vocabulary table, a separate transformation exists that extracts the vocabulary data from CSV files exported from Athena [72] and loads them into the OMOP database.

Data Transformation and Loading

After the setup, the tables of the OMOP CDM that do not contain the vocabulary are truncated to ensure that in consecutive runs of the job, no old data is mixed with the new data. The Truncate statements are executed via an SQL script. Next, the transformations for the separate OMOP CDM tables are executed. First, the fixed transformations are executed. The transformations handle the OMOP CDM tables in the following order:

1. Care site
2. Fact relationship
3. Person
4. Death
5. Visit occurrence
6. Visit detail
7. Observation period (after custom transformations)

The custom transformation is executed for each template ID listed in the list of templates to be transformed.

Care site. The implementation of the care site transformation follows the conceptual workflow described in Section 5.2.2. In PDI, relevant records are extracted from the source table of clinical units, assigned the appropriate place_of_service_concept_id, and deduplicated to ensure uniqueness. The resulting dataset is then written into the CARE_SITE table.

Fact relationship. The implementation of this workflow in PDI follows the conceptual design described in Section 5.2.2. The relevant data is extracted from the care_site table and then split into two branches to represent the “is part of” and “contains” relationships. In each branch, the care_site_id is renamed to represent the respective fact id, and the

appropriate concept ID for the hierarchical relationship is assigned as a constant. After that, the care site ID of the AKH is added as the other fact to each row. After adding the AKH's care_site_id to each row, the two branches are merged, and the resulting records are written into the FACT_RELATIONSHIP table.

Person. Patient data is processed according to the mappings introduced in Section 5.2.2. In PDI, the gender attribute is converted into standardized concept IDs, the birth date is split into year, month, and day, and race/ethnicity fields are added as constants. A deduplication step prevents duplicate patient records before the data is inserted into the PERSON table.

Death. The transformation of death-related information is implemented as described conceptually in Section 5.2.2. Patient identifiers are mapped between source and target systems, the date of death is derived from available datetime fields, and a constant value is used to assign the death-type concept ID. The records are then loaded into the DEATH table.

Visit occurrence. Visit information is processed in accordance with the rules described in Section 5.2.2. In practice, visit periods are derived from datetime fields. Concept IDs representing visit type and setting are assigned, and the care site of the AKH is added. The processed records are stored in the VISIT_OCCURRENCE table.

Visit detail. The patient movement data is transformed as outlined conceptually in Section 5.2.2. In PDI, movement timestamps are grouped to construct visit intervals, linked to their corresponding visit occurrence and care site IDs, and enriched with the appropriate concept IDs. Invalid rows without a matching visit occurrence are filtered out before loading the data into the VISIT_DETAIL table.

Observation period. Observation periods are generated following the approach described in Section 5.2.2. In the implementation, start dates are derived from the first available clinical or visit events. In contrast, end dates are determined by either the death date, the last clinical activity, extended by 60 days, or the extraction date. Invalid entries are filtered out, and a constant period type concept ID is added before inserting into the OBSERVATION_PERIOD table.

Custom transformations. The custom transformations of the ETL process are based on a transformation template executed for each template ID. A CSV file stores the list of template IDs that should be processed. Each row in this file corresponds to a distinct transformation run for a given data configuration.

To execute the same transformation logic for each template ID dynamically, the pattern based on PDI "Copy rows to result" step [73] is applied. This step allows transferring rows

in memory from one transformation to another, eliminating the need for intermediate disk or database storage.

The setup consists of a parent job with two sequential transformations. The first transformation reads the template IDs from the CSV file and passes them via "Copy rows to result." The second transformation consumes this in-memory data by enabling "Copy previous results to parameters" and "Execute for every input row." This causes the transformation to be executed once per input row (i.e., once per template ID), effectively implementing loop-like behavior analogous to a "for each loop" in conventional programming.

This pattern enables modular and efficient processing, particularly in scenarios that require repeating the same logic for multiple configurations while maintaining a clear separation between control flow and data transformation logic.

To support custom transformation behavior, the implementation leverages PDI's "ETL Metadata Injection" step [74]. This feature allows transformation steps to be parameterized at runtime using metadata derived from external sources such as files or database queries. Instead of creating static, hardcoded transformations for each input structure, a single template transformation is defined. The actual behavior and configuration of this template are controlled via metadata injected at runtime.

After reading the CSV file with template IDs, the transformation responsible for injecting metadata into the template transformation is executed. This intermediate transformation performs several queries to gather the required metadata from the RDA platform and the OHDSI standardized vocabularies. It contains five distinct branches, each preparing a specific part of the metadata:

- A branch that retrieves and merges default and custom structural mappings, enriches them with data types.
- A branch that determines the name of the datetime column.
- A branch that retrieves all column names for the target OMOP CDM table.
- A branch that determines the appropriate name of the date and type_concept_id columns.
- A branch that combines the name of the date column with the name of the column for the calculated visit_detail_id.

These metadata fragments are injected into the template transformation via the Metadata Injection step. The template transformation then performs the actual transformation of the source data to the OMOP CDM.

Within the template transformation, data is extracted from the clinical data source tables and from the OMOP_Mapping table that specifies the structural mapping.

The relevant value sets are retrieved, and the attributes of the value set items stored in JSON format are decoded into individual columns. The values in the source data are

matched with the value set items. The standardized codes from the value set, form field, and units are resolved to concept IDs using the OHDSI standardized vocabularies. The transformed data is subsequently pivoted according to the structural mapping. Foreign keys are derived for the person_id, visit_occurrence_id, and visit_detail_id. If the data is not directly linked to a visit detail, the appropriate visit_detail_id is calculated based on the record dates and ward movements.

The date is extracted by truncating the time component from the datetime. A type_concept_id is assigned, and missing columns (based on the complete column list of the OMOP CDM target table) are appended with NULL values to ensure schema compatibility.

Finally, the transformed data is written into the respective OMOP CDM table.

This dynamic and metadata-driven approach allows the ETL system to flexibly adapt to new datasets or changes in data structure without requiring changes to the underlying transformation logic.

CHAPTER 6

Evaluation

This chapter presents two evaluation scenarios that demonstrate the practical application of the generic ETL code base developed in this thesis. These evaluation scenarios were selected to highlight the system's flexibility, effectiveness, and performance in transforming heterogeneous healthcare data stored in the EAV model of the RDA platform at the MedUni Vienna into the standardized OMOP CDM.

The first evaluation scenario focuses on the automated surveillance of hospital-onset bacteremia and fungemia (HOB), a clinical quality and infection control use case that relies on timely and accurate data. The second evaluation scenario, breast cancer benchmarking (BCB), is part of a broader initiative aimed at evaluating and comparing treatment quality across institutions.

Each evaluation scenario follows a consistent structure:

- The background section outlines the clinical or research motivation and the intended use of the data-driven system.
- The data section describes the relevant datasets and their sources within the RDA platform of the MedUni Vienna.
- The transformation results section presents the outcome of the ETL process into the OMOP CDM in terms of data accuracy and completeness.
- Performance metrics evaluate the runtime efficiency of the transformation.

Together, these evaluation scenarios illustrate the practical relevance, adaptability, and generalizability of the developed ETL framework across different clinical use cases.

To support the execution of the evaluation scenarios in a production-like setting, the architectural design differs slightly from the development environment. While the

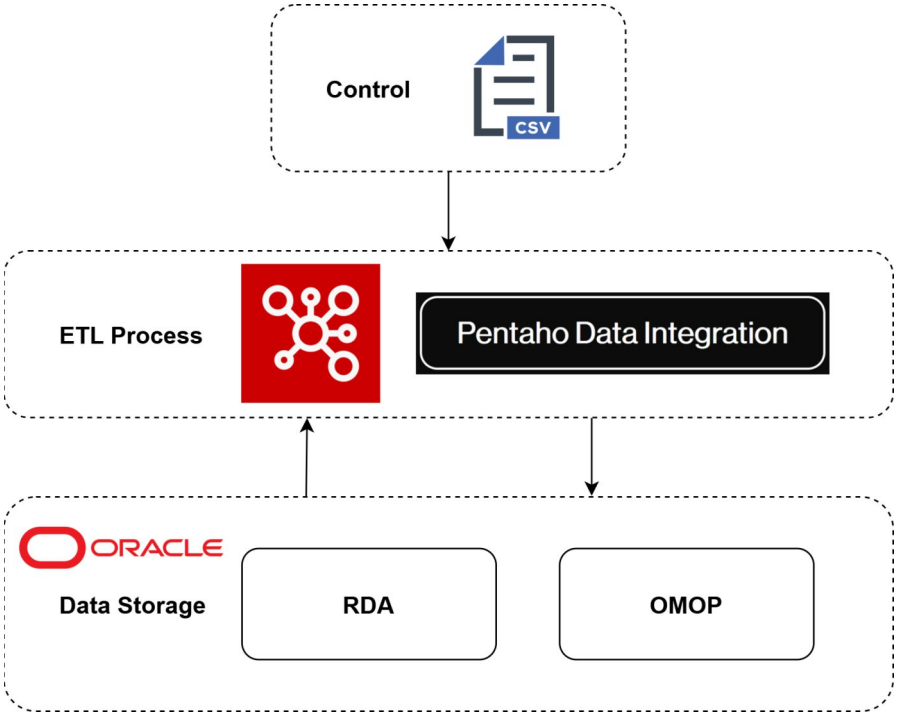


Figure 6.1: Architecture of the Prototype for the Evaluation.

development of the generic ETL code base was carried out using a lightweight SQLite database to ensure portability and rapid prototyping, the evaluation scenarios operate entirely on an Oracle-based infrastructure. Both the source data in the RDA platform and the target OMOP CDM database are hosted on Oracle databases provided by the MedUni Vienna. Oracle was selected for its richer set of features and better alignment with existing clinical systems, ensuring compatibility, performance, and production readiness. Figure 6.1 illustrates the adjusted architecture used for the evaluation scenarios.

After presenting the evaluation scenarios, this chapter discusses the technical, semantic, and practical challenges encountered during the implementation of the ETL process, along with the strategies used to address them. The challenges are organized into three sections: challenges common to both evaluation scenarios, highlighting general implementation challenges, and those specific to each evaluation scenario, reflecting context-dependent challenges and solutions.

The subsequent section evaluates the developed ETL prototype against the functional and non-functional requirements defined in Section 5.1. This evaluation draws on both the implementation results and the insights gained from the two evaluation scenarios. Each requirement is assessed individually and marked as fulfilled, partially fulfilled, or not fulfilled.

6.1 Evaluation Scenario 1: Automated Surveillance of Hospital-onset Bacteremia and Fungemia

This evaluation scenario demonstrates the application of the generic ETL code base to support the automated surveillance of HOB, a critical task in hospital infection control and quality assurance. The focus lies on transforming routine clinical data from the RDA platform of the MedUni Vienna into the OMOP CDM to enable standardized analysis and comparison. The following subsections describe the medical background, underlying data, transformation outcomes, and performance metrics.

6.1.1 Background

Healthcare-associated infections (HAIs) are among the most common adverse events in medical care [75]. Patients acquire these infections while receiving treatment for other conditions in healthcare facilities [76]. The infections are not limited to hospitals but can occur in all healthcare facilities, such as long-term care facilities, rehabilitation centers, or doctors' offices [77]. The most common HAIs include surgical site infections (SSIs), ventilator-associated pneumonias (VAPs) in intensive care units (ICUs), catheter-associated urinary tract infections (CAUTIs), and HOBs [78]. This evaluation scenario will focus on HOBs.

HAIs are a significant cause of prolonged morbidity and increased mortality in patients receiving medical treatment and care in healthcare institutions [79], [80]. They also cause prolonged LOSs and increased medical and nursing workload. Therefore, HAIs not only burden the patients themselves due to the associated pain and discomfort but also lead to increased healthcare costs [81]. For these reasons, surveillance of HAIs is a global standard for infection prevention and control in hospitals. It enables targeted implementation and monitoring of interventions to reduce the number of HAIs [82].

In most institutions, the surveillance is performed through a manual review of medical records. This traditional surveillance method is performed by infection control practitioners (ICPs) or infection control nurses (ICNs), who review the records to determine if the definition of an HAI is met. The problem with this type of surveillance is that it is time-consuming, labor-intensive, and costly. It is also prone to subjectivity. Suppose the ratio of ICPs or ICNs to patients is disproportionate. In that case, the surveillance may be limited in scope to cover only high-risk areas due to resource constraints or to include only some patients, or it may have a delayed response. These problems should be reduced by automated surveillance [83]. The increased availability of data stored in EHRs provides opportunities to (partially) automate the surveillance of HAIs.

The aim is to facilitate surveillance to identify outbreaks, enable early interventions, and reduce the burden on ICPs and ICNs by reusing existing routine care data to detect and document HAIs. It also enhances the efficiency of the surveillance through automated data extraction and analysis. It provides real-time or near-real-time monitoring of HAIs, ensuring timely identification and addressing of clusters of infections. Automated

surveillance ensures consistency by applying standardized criteria for identifying and reporting HAIs. It also offers comprehensive coverage by enabling surveillance of the entire patient population rather than focusing solely on high-risk areas. A distinction is made between semiautomated and fully automated surveillance. In semiautomated surveillance, an algorithm divides the cases into those with a high probability of an HAI and those with a low probability of an HAI. Only cases with a high probability of an HAI are subject to manual chart review. With fully automated surveillance, the system automatically classifies the cases as to whether an HAI is present or not [84].

The EU project PRAISE (Providing a Roadmap for Automated Infection Surveillance in Europe) is leading the way in developing methodologies for automated HAI surveillance to advance automated multicenter HOB surveillance from research to large-scale, real-world implementation. The project focuses on creating standardized definitions and algorithms for HAI detection, ensuring interoperability across different healthcare institutions. Several recommendations for implementing automated surveillance are being developed within this project, resulting in several guidelines, e.g., for governance, technical requirements, etc. [85], [86], [87]. Now, members of this group have reconvened and started to discuss automated surveillance with a focus on automated surveillance of SSIs and HOBs. The PRAISE network developed an algorithm for automated surveillance of HOB. Figure 6.2 shows a schematic representation of the algorithm. In this algorithm, a HOB is defined as a positive blood culture with a recognized pathogen two or more days after hospital admission. For Common Skin Commensals (CSCs), two positive cultures within two days are required [88].

The HOB algorithm was applied retrospectively to data from four European hospitals, demonstrating its feasibility and reproducibility. The results showed consistent HOB rates across different hospitals. The study suggests that automated HOB surveillance can be an actionable tool for infection control [88].

A significant challenge in developing automated surveillance systems is the integration of heterogeneous healthcare data sources and their transformation into a standardized format that ensures semantic interoperability. Such standardization is essential for seamless data integration and enables scalable, reproducible epidemiological research across institutions [89]. The OMOP CDM addresses this challenge by harmonizing data from diverse healthcare systems into a common structure. Moreover, it promotes reusability of analysis workflows, as consistent data structures eliminate the need for site-specific adaptations [30].

6.1.2 Data

The PRAISE network defined a Minimal Data Set (MDS), specifying the minimally needed data elements for the application of the HOB algorithm as well as the required data structure to support validation and further deployment in different settings. The MDS describes the minimum input to achieve the algorithm output and enabling reporting. The following sections describe the essential data elements and their definitions.

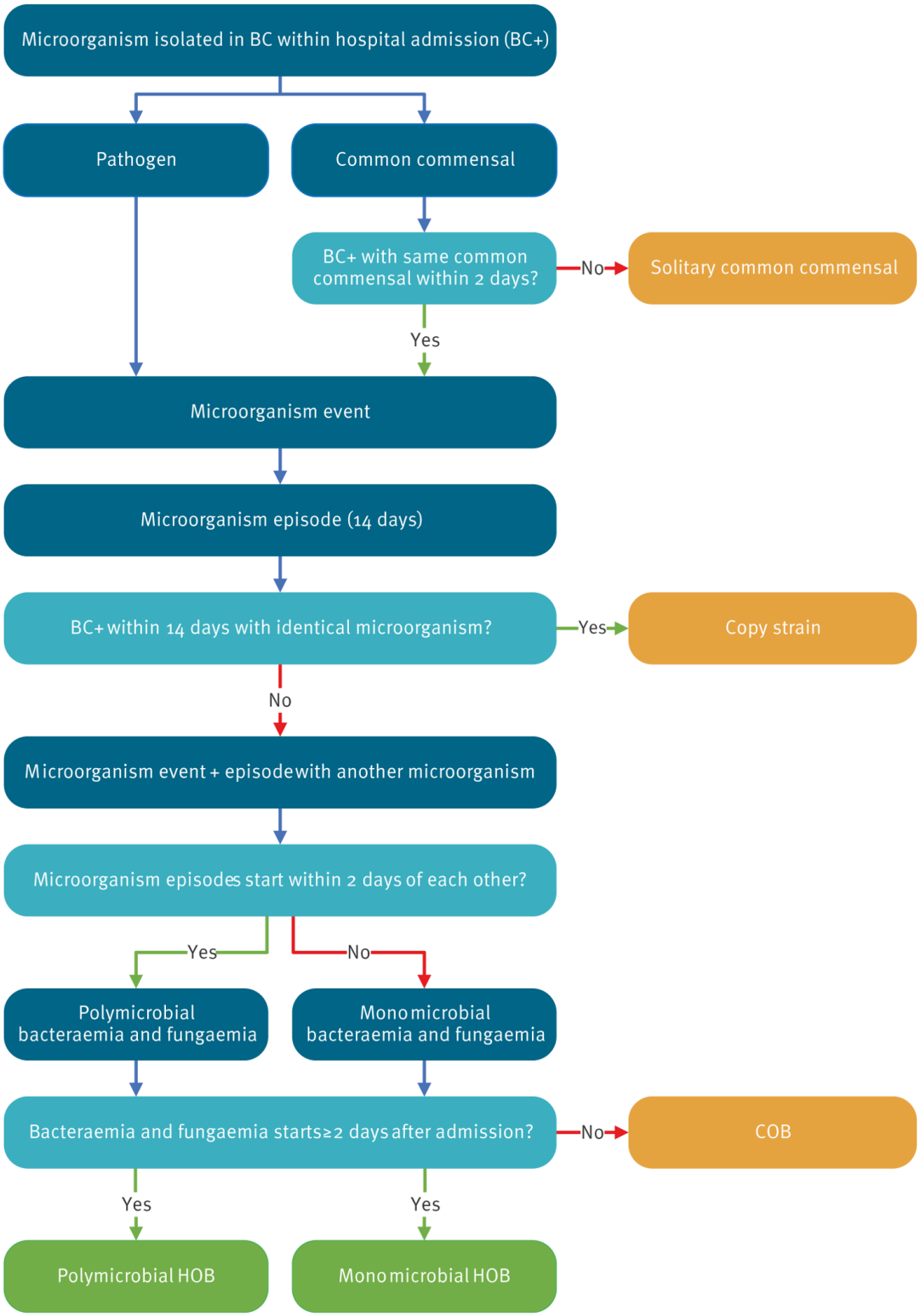


Figure 6.2: Schematic representation of the HOB algorithm [88].

Patient Demographics

1. Patient ID: A unique identifier for each patient to track their medical records without revealing personal information.
2. Sex: The patient's sex.
3. Birth Date: The patient's birth date.
4. Death Date: The patient's death date (only if the patient died in the hospital).

Blood Culture Results

Only positive blood cultures are considered.

1. Blood culture ID: A unique identifier of each microorganism in a blood culture, for example, a combined sample ID and isolate number.
2. Sample ID: A unique identifier for each blood culture sample. One sample can have multiple isolates.
3. Patient ID: Reference to the patient.
4. Sample Date: Date when the blood culture was taken.
5. Sample Ward: The ward where the blood culture was taken.
6. Isolate Number: Sequential identifier of the microorganism in the blood culture.
7. Microorganism: Microorganism identified in the blood culture.
8. Attributable Ward: Ward where the patient was two days before the blood culture was taken.
9. Hospital Admission Date: Date when the patient was admitted to the hospital.

Ward Classification

For reporting purposes, the wards are classified according to the European Centre for Disease Prevention and Control (ECDC) ward specialty [90]. Data providers/hospitals may also perform analyses by ECDC patient specialty [90] or a local ward classification system.

1. Ward ID
2. ECDC ward specialty classification
3. ECDC patient specialty classification
4. Local classification

Microorganism Classification

The algorithm runs on the local microorganism representations. The microorganisms may be mapped to the CSC list by SNOMED CT code or manually.

1. Local microorganism ID
2. SNOMED CT Code of the microorganism
3. SNOMED CT label or local label of the microorganism
4. Indicator whether the microorganism is a CSC as per National Healthcare Safety Network (NHSN) classification [91]

OMOP CDM tables

Based on the MDS of the PRAISE network, the following tables of the OMOP CDM are needed:

- Observation_Period
- Person
- Death
- Specimen
- Measurement
- Care_site
- Visit_occurence
- Visit_detail

The structural mapping only needs to be specified for the care_site, specimen, and measurement tables. The other tables follow the default implementation. For the care_site table, the mapping of the care site to the ECDC ward specialty needs to be added.

6.1.3 Transformation Results

Following the execution of the ETL pipeline, the transformed care site and microbiology data were analyzed within the OMOP CDM to evaluate completeness and consistency.

A total of 520 care sites were recorded. Thirteen care sites lacked an assigned ECDC ward specialty due to missing mapping information.

A total of 2,819 specimens were recorded, with the number of specimen records matching the original microbiology documents, confirming a one-to-one correspondence between source records and OMOP entries. Of these, three specimens were assigned a specimen_concept_id of 0. These correspond to documents related to non-blood culture specimens, which fall outside the current mapping scope and therefore lack a defined standard concept.

Regarding temporal completeness, 409 specimens lacked a recorded specimen_datetime. In these cases, only the document creation date was available in the source system. As a fallback, the transformation pipeline preserved the date portion and left the time element null to avoid introducing false precision.

Anatomic site information was partially incomplete. For 106 specimens, the anatomic_site_concept_id was set to 0 due to the absence of a valid concept mapping. In an additional 138 cases, the anatomic site information was not present in the source data, resulting in both the anatomic_site_concept_id and the corresponding anatomic_site_source_value being empty.

A total of 3,213 measurements were recorded, matching the number of original organisms in the microbiology documents and confirming a one-to-one correspondence between source and OMOP records. Two measurements were assigned a measurement_concept_id of 0. These originated from non-blood culture documents outside the mapping scope and were also assigned a value_as_concept_id of 0.

Regarding temporal completeness, 803 measurement records lacked a recorded measurement_datetime. As with specimens, only the document creation date was available in these cases. The transformation pipeline retained the date and omitted the time portion to avoid false precision.

A total of 2,284 measurements lacked an assigned visit_detail_id. Since microbiology documents in the source system were not consistently linked to encounter details, visit matching had to be performed based on the available date information.

6.1.4 Performance

The performance of the ETL pipeline was evaluated based on execution time under different configurations. The core ETL process, excluding setup and with caching enabled, executed in 7.5 minutes, demonstrating efficient runtime for standard use. Caching avoided redundant lookups for vocabulary mappings and concept relationships.

Among the transformation steps, the fixed transformations were executed in 1 minute, while the custom transformations took 2.5 minutes. The setup of custom vocabularies was completed in 4 minutes.

To assess end-to-end deployment time, a second benchmark included the setup phase, such as schema generation, constraint application, and vocabulary loading. In this configuration, the total runtime increased to 29.5 minutes, reflecting the overhead introduced by database configuration and vocabulary loading.

The setup of the OMOP CDM took 22 minutes, with most of this time spent loading the OHDSI standardized vocabularies.

All ETL executions were performed on a Windows 10 Enterprise edition machine with a 64-bit Intel Core i9 3.20 GHz CPU and 32 GB of RAM.

6.2 Evaluation Scenario 2: Breast Cancer Benchmarking

This evaluation scenario demonstrates the application of the generic ETL code base to support the benchmarking of breast cancer care quality indicators. The focus of this evaluation scenario lies in transforming routine clinical data from the RDA platform of the MedUni Vienna into the standardized OMOP CDM, thus enabling the execution of systematic analyses using shared and reproducible analytical routines. The following subsections describe the clinical background, source data characteristics, transformation results, and performance metrics.

6.2.1 Background

Breast cancer is the most frequently diagnosed cancer among women in the European Union (EU). According to Eurostat, in 2021, an estimated 84,800 people died from breast cancer in the EU, of whom the vast majority (approximately 83,900) were women. Breast cancer accounted for 7.4% of all cancer-related deaths in the EU population and 16.5% of all cancer-related deaths among women [92], [93].

In terms of overall mortality, breast cancer was responsible for 1.6% of all deaths in the EU in 2021, and 3.2% of all deaths among women. The age-standardized death rate for breast cancer among women was 30.6 per 100,000 inhabitants, though this rate varied substantially between member states, from a high of 37.4 per 100,000 in Hungary to a low of 22.2 per 100,000 in Spain. These variations indicate potential differences in cancer care delivery, early detection, and treatment outcomes across the EU [92], [93].

Breast cancer also places a substantial burden on health services. In 2021, EU hospitals reported 447,100 inpatient discharges for breast cancer. Austria had the highest discharge rate, exceeding 200 discharges per 100,000 inhabitants, while most countries reported rates below 100 per 100,000. The average length of hospital stay for breast cancer patients ranged from 1.9 days in the Netherlands to over 9 days in Germany and Malta [92], [93].

Given its high prevalence and burden, breast cancer represents a critical focus area for efforts to improve care quality and outcomes through structured data analysis and inter-institutional collaboration. In response to this challenge, the European University Hospital Alliance (EUHA) initiated a collaborative benchmarking project focused on breast cancer care. Nine leading university hospitals across the EU have partnered to develop a minimal yet clinically relevant set of indicators that allow retrospective comparison of care processes and outcomes in female breast cancer patients. These indicators are derived from established international standards, including those developed

by the European Society of Breast Cancer Specialists (EUSOMA) and the International Consortium for Health Outcomes Measurement (ICHOM) [93], [94].

The primary aim of this project is not to produce a definitive benchmarking report, but to demonstrate that clinically meaningful benchmarking across EUHA hospitals is feasible within a short time frame and with sustainable effort. The project serves as a proof-of-concept for federated benchmarking using harmonized electronic health data. Each participating institution maps its local data to the OMOP CDM, enabling decentralized calculation of indicator values using a standardized algorithm developed by University Hospitals Leuven's Management and Information Reporting team. Only anonymized, aggregated metrics are shared centrally to ensure compliance with the General Data Protection Regulation (GDPR) [93], [94].

This approach facilitates trust among institutions and enables benchmarking without transferring patient-level data. By demonstrating the viability of this methodology in breast cancer, a high-burden care program, the project aims to lay the groundwork for broader benchmarking efforts across other diseases and care domains within the EUHA network [93], [94].

Additionally, the project is aligned with broader European initiatives such as the Health Outcomes Observatory (H2O), which seeks to integrate patient-reported outcomes with clinical data across multiple diseases and jurisdictions. Within this context, participating institutions like the MedUni Vienna are contributing data for various disease areas, including breast cancer, further reinforcing the push toward data-driven, patient-centered care across Europe [93], [94].

6.2.2 Data

The BCB project relies on a predefined set of key clinical and process indicators derived from established frameworks such as EUSOMA and ICHOM. The indicators cover aspects such as diagnosis, treatment cycles, and patient outcomes. The following sections describe the essential data elements and their definitions.

Patient Demographics

1. Year of birth: The patient's year of birth.

Diagnosis Information

1. Date of histological diagnosis (in case of metachrone metastasis): The initial date of histological diagnosis of the local regional tumor (in case of metachrone metastasis).

Tumor Characteristics

1. Histological type: Indicate histologic type of the tumor (select all that apply).

2. Tumor grade: Indicate tumor grade of DCIS component of tumor: (Bloom-Richardson classification system).
3. Clinical tumor stage: Clinical tumor stage (per AJCC 8th Ed.).
4. Pathological tumor stage: Pathological tumor stage (per AJCC 8th Ed.).
5. Size of invasive tumor: Indicate the size of the invasive component of the tumor (in mm).
6. Lymph nodes involved: Number of lymph nodes involved according to the TNM stage (per AJCC 7th Ed.).
7. Estrogen receptor status: Indicate if the estrogen receptor status is positive.
8. Progesterone receptor status: Indicate if the progesterone receptor status is positive.
9. HER2 receptor status: Indicate if the HER2 receptor status is positive.

Surgical Interventions

1. Risk-reducing surgery before diagnosis of metastases: Indicate if the patient received surgical removal of organs at high risk of developing cancer (e.g., removal of the breasts) prior to metastases.
2. Surgery: Indicate whether the patient received surgery during the last year.
3. Surgery date: Provide the date of surgery.
4. Surgery on primary site: Whether the patient received surgery on the site of the primary tumor.
5. Surgery on metastatic lesions: Whether the patient received surgery on metastatic lesions.
6. Number of lymph nodes resected: Number of lymph nodes resected.

Treatment

1. Chemotherapy: If the patient received chemotherapy during the last year, please indicate the intent of chemotherapy.
2. Radiotherapy: If the patient received radiotherapy during the last year, please indicate the intent of radiotherapy.
3. Hormonal therapy: If the patient received hormonal therapy during the last year, please indicate the intent of the hormone therapy.
4. Targeted therapy: Indicate what type of targeted therapy.

5. Treatment 0: Indicate whether the patient received no initial treatment for the primary tumor.
6. Start date of new treatment of metastases: Date when new treatment line (in case of chemotherapy or hormonal therapy) or treatment modality (in case of radiotherapy and surgery) was started.
7. Radiotherapy start date: Please provide the start date of radiotherapy.
8. Radiotherapy stop date: Please provide the stop date of radiotherapy.
9. Chemotherapy start date: Please provide the start date of chemotherapy.
10. Chemotherapy stop date: Please provide the stop date of chemotherapy.
11. Treatment status (treatment of metastases): Status of treatment of the metastases.
12. Treatment status (treatment of metastases) change date: Date of change in treatment status.
13. Standard therapy versus experimental/clinical trial therapy: Whether treatment was received according to the guidelines, standard therapy other than guideline, or as part of a clinical trial.
14. Time from diagnosis to treatment: Time between the date of diagnosis of metastasis (based on histological diagnosis on biopsy, and otherwise the date of diagnosis on imaging) and the start date of first treatment.
15. Treatment of metastases: Chemotherapy (with or without targeted therapy): Whether the patient received chemotherapy, neoadjuvant or adjuvant, for the primary tumor.
16. Lines of Chemotherapy (with or without targeted therapy): Indicate the current line of chemotherapy: 1st, 2nd–3rd, or 4th and beyond. Only applicable for systemic therapy.
17. Treatment of metastases: Hormonal therapy (with or without targeted therapy): Whether the patient received hormonal therapy, neoadjuvant or adjuvant, for the primary tumor.
18. Lines of hormonal therapy (with or without targeted therapy): Indicate the current line of hormonal therapy: 1st, 2nd–3rd, or 4th and beyond. Only applicable for systemic therapy.
19. Treatment of metastases: Radiotherapy: Whether the patient received radiotherapy during the last year for the primary tumor.
20. Localisation of (stereotactic) radiotherapy: The localisation of the (stereotactic) radiotherapy.

Outcomes

1. Vital Status: Indicate if the person has deceased, regardless of cause.
2. Date of death: Date of death.
3. Death attributable to breast cancer: Indicate if death is attributable to breast cancer.
4. Progression Free Survival/duration of response: Time between initiation of treatment for metastases and documented progression or change in treatment due to lack of response.
5. Objective response: Response to treatment, categorized as complete response, partial response, or no response.

OMOP CDM tables

Based on the data, the following tables of the OMOP CDM are needed:

- Observation_period
- Person
- Death
- Condition_occurrence
- Measurement
- Procedure_occurrence
- Drug_exposure
- Observation
- Care_site
- Visit_occurence

The structural mapping only needs to be specified for the condition_occurrence, measurement, procedure_occurrence, drug_exposure, and observation tables. The other tables follow the default implementation.

6.2.3 Transformation Results

Following the execution of the ETL pipeline, the transformed care site, visit, and oncology data were analyzed within the OMOP CDM to evaluate completeness and consistency.

A total of 7 care sites were recorded, matching the number of different care sites associated with the source documents.

A total of 10,935 visit occurrences were recorded, matching the original oncology documents, confirming a one-to-one correspondence between source records and OMOP entries. All visit occurrences had an assigned care site.

A total of 25,759 condition occurrences were recorded, matching the number of original records in the RDA platform and confirming a one-to-one correspondence between source and OMOP records. Of these, 365 records were assigned a condition_concept_id of 0. For those records, the mapping was missing because the value was from an outdated version of the value set and was not considered during the semantic mapping.

Regarding temporal completeness, 404 condition occurrences lacked a recorded condition_start_datetime. In these cases, only the document creation date was available in the source system. As a fallback, the transformation pipeline preserved the date portion and left the time element null to avoid introducing false precision.

A total of 42,340 measurements were recorded, matching the number of original records in the RDA platform and confirming a one-to-one correspondence between source and OMOP records.

Regarding temporal completeness, 1,295 measurements lacked a recorded measurement_datetime. In these cases, only the document creation date was available in the source system. As a fallback, the transformation pipeline preserved the date portion and left the time element null to avoid introducing false precision.

A total of 41,582 measurement records were eligible for linkage to a corresponding condition occurrence. Of these, 39,178 include an actual reference. The remaining records could not be linked because the corresponding condition occurrence was missing in the source data, likely due to documentation inconsistencies or omissions.

A total of 47,218 procedure occurrences were recorded, matching the number of original records in the RDA platform and confirming a one-to-one correspondence between source and OMOP records. Of these, 777 records were assigned a procedure_concept_id of 0. For those records, the mapping was missing because the value was from an outdated version of the value set and was not considered during the semantic mapping.

Regarding temporal completeness, 5,197 procedure occurrences lacked a recorded procedure_datetime. In these cases, only the document creation date was available in the source system. As a fallback, the transformation pipeline preserved the date portion and left the time element null to avoid introducing false precision.

A total of 1,036 drug exposures were recorded, matching the number of original records in the RDA platform and confirming a one-to-one correspondence between source and

OMOP records. Of these, 139 records were assigned a drug_concept_id of 0. For those records, the term was mapped to a non-standard concept, which is not mapped to a standard concept. Therefore, no standard concept was assigned during the ETL process.

Regarding temporal completeness, 362 drug exposures lacked a recorded drug_exposure_start_datetime. In these cases, only the document creation date was available in the source system. As a fallback, the transformation pipeline preserved the date portion and left the time element null to avoid introducing false precision.

A total of 30,092 observations were recorded, matching the number of original records in the RDA platform and confirming a one-to-one correspondence between source and OMOP records.

Regarding temporal completeness, 173 observations lacked a recorded observation_datetime. In these cases, only the document creation date was available in the source system. As a fallback, the transformation pipeline preserved the date portion and left the time element null to avoid introducing false precision.

A total of 8,302 observation records were eligible for linkage to a corresponding condition occurrence. Of these, 6,156 include an actual reference. The remaining records could not be linked because the corresponding condition occurrence was missing in the source data, likely due to documentation inconsistencies or omissions.

6.2.4 Performance

The performance of the ETL pipeline was evaluated based on execution time under different configurations. The core ETL process, excluding setup and with caching enabled, executed in 4 hours 29 minutes, reflecting the increased data volume and complexity of the BCB dataset. Caching avoided redundant lookups for vocabulary mappings and concept relationships.

Among the transformation steps, the fixed transformations were executed in 2 minutes, while the custom transformations took 4 hours 23 minutes, indicating that the large number of different templates was the main performance driver. The setup of custom vocabularies was completed in 4 minutes.

To assess end-to-end deployment time, a second benchmark included the setup phase, such as schema generation, constraint application, and vocabulary loading. In this configuration, the total runtime increased to 4 hours 52 minutes, reflecting the overhead introduced by database configuration and vocabulary loading. The setup of the OMOP CDM took 23 minutes, with most of this time spent loading the OHDSI standardized vocabularies.

All ETL executions were performed on a Windows 10 Enterprise edition machine with a 64-bit Intel Core i9 3.20 GHz CPU and 32 GB of RAM.

6.3 Challenges of the Evaluation Scenarios

During the implementation of the evaluation scenarios, several challenges emerged. These challenges arose from different aspects of the ETL process, including data modeling, semantic mapping, configuration management, technical constraints, and performance optimization. Some issues were inherent to the general task of adopting the prototype to a specific use case, while others were specific to the characteristics of the individual evaluation scenarios.

To provide a structured discussion, the following section first highlights challenges that were common across both evaluation scenarios and then examines scenario-specific difficulties in more detail. In each case, the description of the problem is followed by the strategies adopted to mitigate or resolve it.

6.3.1 Common Challenges across both Evaluation Scenarios

Several challenges were not tied to the specifics of a single evaluation scenario but emerged consistently across both evaluation scenarios. These overarching issues relate to fundamental aspects of the ETL process, such as metadata management and technical limitations. This subsection presents the key challenges and the strategies used to address them.

Incomplete Concept Coverage in the RDA platform. Not all required concepts for the transformation of data from the RDA platform to the OMOP CDM were directly available in the existing RDA platform implementation. For instance, the attributes for value sets in JSON, as well as the semantic and structural mapping tables, had not yet been implemented. As a workaround, these tables were added in a personal schema within the RDA platform database, enabling their integration into the transformation pipeline.

Primary Key Handling after Database Migration. Initial development was carried out using SQLite for simplicity, with PDI connecting to the database via JDBC. However, after migrating to Oracle and connecting via OCI, it was not possible to retrieve the automatically generated primary keys of inserted rows. This limitation affected the ability to reference those keys in subsequent transformation steps. To resolve this, a sequence was introduced and used to explicitly generate primary key values for all OMOP CDM tables affected by the custom transformations. Generating the identifiers in advance ensured that identifiers could be set and reused consistently throughout the ETL process.

Constraint Implementation after Database Migration. Initial development was carried out using SQLite for simplicity. Although temporary workarounds were introduced to mitigate real-world testing challenges like file locking, the pipeline was ultimately migrated to an Oracle database to take advantage of its robust performance and concurrency capabilities. This switch also enabled the use of relational constraints such as foreign

keys, which SQLite does not support via ALTER TABLE. As a result, the setup process had to be adapted to define constraints during schema creation and transformation steps explicitly. While this increased the complexity of the deployment process, it significantly improved data integrity and consistency checks across related tables.

Data Type Incompatibilities. The RDA platform uses generic SQL types such as NUMBER for primary and foreign keys. To prevent join errors and imprecision, all such fields were explicitly cast to the appropriate data types during transformation.

Scalability and Database Engine Constraints. Initial development was carried out using SQLite for simplicity. However, real-world testing revealed file locking and concurrency issues under increased data volume. Although temporary workarounds were introduced (e.g., wait until the output step has written all data), the pipeline was ultimately migrated to an Oracle database to take advantage of its robust performance and concurrency capabilities.

6.3.2 Automated surveillance of hospital-onset bacteremia and fungemia

During the implementation of the ETL pipeline for the automated surveillance of HOB, several challenges were encountered. These spanned semantic mapping, filtering logic, technical limitations, and performance concerns. This section outlines the key issues and the strategies used to address them.

Semantic Mapping. Local codes were mapped to the OHDSI standardized vocabularies using Athena as a reference. Microorganism mappings were created using Usagi, covering both organism identifiers and detection status. Where direct matches were unavailable, broader concepts were manually selected. To ensure clinical accuracy, these mappings were reviewed and validated by domain experts with a microbiology background.

Structural Mapping. The lack of established THEMIS conventions for microbiology data necessitated the definition of custom structural mappings. While effective for this study, such institution-specific adaptations may lead to inconsistencies across sites, potentially affecting data comparability in multicenter analyses. The THEMIS convention for microbiology data is still a work in progress in the OHDSI community without a preferred concept yet.

Care Site Mapping to ECDC Ward Specialty. The default implementation of the ETL process for the care_site table did not include a mapping to the ECDC ward specialty, which is needed in this use case for classifying care settings in line with epidemiological surveillance standards. To address this, the transformation logic for the care_site table was extended to include additional mapping metadata. Local unit codes were manually mapped to the corresponding ECDC ward specialties based on clinical

documentation and hospital organizational data. The mapping was stored in a reference file and loaded during the transformation, enabling correct population of the care_site and location tables with standardized specialty classifications.

Selective Document Filtering. Not all microbiology documents were relevant for this transformation. Only blood culture results were needed, while other findings had to be excluded. Two strategies can be used to address this:

- Predefining restricted database views to include only the relevant subset.
- Duplicating the generic transformation definition with injecting use-case specific metadata and adding filters to narrow selection criteria.

In this scenario, the dataset was already based on a database view, which was updated to include only the relevant documents. This approach ensured that the ETL pipeline only processed relevant documents without adding additional filtering logic in the transformation steps. While the second strategy provides greater flexibility, it was not necessary for the HOB scenario, as the database views already provided an effective and efficient filtering mechanism.

Missing Data. Missing values in key fields, such as timestamps, posed additional challenges. When no timestamp was recorded for a microbiology result, the fallback was to use the document creation date. In such cases, only the date portion is preserved, reducing temporal granularity. While this approach ensures completeness, it may introduce temporal imprecision, and including time fields could misleadingly suggest accuracy where none exists.

Internal Bugs and Fixes. An issue was discovered in the handling of attributes recorded multiple times per document. Only the first occurrence correctly received associated data, while subsequent instances lacked shared single-entry attributes. The query logic was revised to ensure all instances were populated with the necessary values.

6.3.3 Breast Cancer Benchmarking

During the implementation of the ETL pipeline for the BCB, several challenges were encountered. These spanned mainly the structural mapping. This section outlines the key issues and the strategies used to address them.

Template-Specific Transformation Logic. Certain templates required transformation behavior beyond what was supported by the default implementation. About half of the templates required filtering the records based on the value of a specific attribute. In other cases, conditional logic dictated whether one or two OMOP records needed to be created based on the value of a single attribute, or the correct datetime had to be selected using a fallback chain: if the primary date attribute was missing, alternative fields were checked in a specific order until a valid date was found. These variations required extending the transformation framework with template-specific logic.

Structural Mapping Deviations for Core Tables. The transformation of the fixed OMOP tables, care_site and visit_occurrence, required adjustments to the structural mapping. The format and relationships in the breast cancer data differed from assumptions in the generic ETL, prompting targeted updates to ensure correct integration.

Handling of Missing Mandatory Attribute Values. In some instances, the source data did not provide values for attributes that were mapped to mandatory fields in the OMOP CDM, such as measurement_concept_id. Without these values, it was not possible to create valid records, as the target schema requires concept identifiers to be present. To maintain data integrity, affected rows were excluded during transformation. The pipeline logged each excluded entry to ensure transparency and allow for further investigation if necessary.

Adapted Drug Exposure Duration Logic. The standard ETL logic assumes identical start and end dates for drug exposures, following the THEMIS convention. However, the breast cancer data modeled treatment duration explicitly as 29 days. To reflect this correctly, the transformation was updated to calculate the drug exposure end date as 29 days after the start date.

6.4 Evaluation of the Requirements

This section evaluates the developed ETL prototype in terms of the functional and non-functional requirements defined in Section 5.1. The objective of this evaluation is to determine to what extent the prototype fulfills the requirements and thus the goals of transforming healthcare data from the EAV-based research database of the MedUni Vienna into the OMOP CDM.

The evaluation is based on the implementation results and the findings from the evaluation scenarios detailed in Chapter 6. Each requirement is assessed individually and marked as fulfilled, partially fulfilled, or not fulfilled. Accompanying comments provide further details on how each requirement is met, highlight any limitations, and indicate areas for future improvement.

By systematically evaluating both functional and non-functional aspects, this section demonstrates the robustness, flexibility, and readiness of the ETL process for real-world applications and future extension.

6.4.1 Evaluation of the Functional Requirements

To assess how well the prototype fulfills the defined functional requirements, each requirement is evaluated based on the implemented functionality. The evaluation distinguishes between fulfilled, partially fulfilled, and not fulfilled features. The results are presented in Table 6.1.

ID	Requirement Summary	Fulfilled	Evaluation Comments
FR01	Extract data from RDA platform	Yes	Connection to the RDA platform and data extraction from EAV tables were successfully implemented and tested in the evaluation scenarios.
FR02	Structural mapping of source attributes to OMOP CDM	Yes	Standard mappings from source EAV attributes to the respective OMOP CDM columns were defined for all relevant tables. These mappings were applied, extended, or replaced as needed during the evaluation scenarios.
FR03	Transform EAV modeled data to OMOP CDM	Yes	The data transformation logic was fully implemented and validated during the evaluation scenarios.
FR04	Load structural mapping dynamically	Yes	Structural mapping rules are loaded from a mapping table at runtime, enabling dynamic and flexible configuration.
FR05	Standardize raw data to follow OMOP CDM conventions	Partially	The ETL process is capable of transforming raw data to the OMOP CDM following the conventions. The process was applied to the data required for the evaluation scenarios. However, not all possible domains or vocabularies are covered yet.
FR06	Load data into OMOP CDM tables	Yes	The loading logic is fully implemented and was verified through the evaluation scenarios. Data is inserted into the correct target tables according to the OMOP CDM.
FR07	Handle missing or inconsistent data	Partially	Missing date values are handled by falling back to the document date. Missing values in optional fields are tolerated, while rows missing required concept IDs are excluded and logged as errors.

ID	Requirement Summary	Fulfilled	Evaluation Comments
FR08	Load metadata for data selection and scope dynamically	Yes	The selection of data elements to be transformed is configurable and loaded at runtime from an external configuration file.
FR09	Set up empty OMOP CDM schema	Yes	Database setup scripts prepare an empty OMOP CDM instance with the correct table structure and vocabularies prior to data transformation and loading.

Table 6.1: Evaluation of the functional requirements for the prototypical implementation of the ETL process.

Overall, the evaluation demonstrates that the implemented prototype fulfills all core functional requirements. The requirements related to standardizing raw data and error handling are partially implemented. They are sufficient for the use cases covered in this thesis, but leave room for future enhancement. The results confirm that the prototype offers a solid and extensible foundation for a robust ETL process tailored to the OMOP CDM.

6.4.2 Evaluation of the Non-functional Requirements

To assess how well the prototype fulfills the defined non-functional requirements, each requirement is evaluated based on the implemented functionality. The evaluation distinguishes between fulfilled, partially fulfilled, and not fulfilled features. The results are presented in Table 6.2.

ID	Requirement Summary	Fulfilled	Evaluation Comments
NFR1.1	Apply optimizations (e.g., indexing, caching, memory management) to improve ETL performance.	Partially	The prototype uses caching to avoid redundant lookups for vocabulary mappings and concept relationships. Additionally, the default indexes of the OMOP CDM and the RDA platform are used. However, additional optimization strategies, such as specific indices and memory management, remain to be implemented.

ID	Requirement Summary	Fulfilled	Evaluation Comments
NFR1.2	Use parallel or batch processing to reduce ETL execution time.	Partially	Batch inserts are implemented, and downstream transformations begin as soon as upstream data becomes available. However, true parallel or distributed execution is not yet in place.
NFR2.1	Structure the ETL as modular components, each handling a distinct subtask.	Yes	The ETL process is composed of modular transformations, each targeting a specific OMOP CDM table or transformation task.
NFR2.2	Ensure transformation and mapping components are reusable across datasets.	Yes	The fixed transformations are reusable and adaptable for standard cases. Custom transformations are adapted based on the configuration and runtime. If this is not sufficient, the transformation with injected metadata can be duplicated and modified to fit the use case.
NFR2.3	Allow updates to individual modules without requiring changes to the entire ETL pipeline.	Yes	Each transformation is independent and can be executed standalone, allowing modular updates without disrupting the full pipeline.
NFR3.1	Use external configuration files or parameters instead of hard-coded ETL settings.	Partially	Some configuration settings, such as the data scope, are stored in external configuration files and loaded at runtime. However, certain parameters, such as the initial flag for rebuilding the OMOP CDM database, are still embedded directly in the implementation.
NFR3.2	Dynamically load attribute names and transformation rules from metadata/config files.	Yes	For custom transformations, attribute names and rules are loaded dynamically from the RDA or mapping tables embedded in the source data.

ID	Requirement Summary	Fulfilled	Evaluation Comments
NFR3.3	Enable the system to adapt to different schema structures with minimal modifications.	Yes	The system handles schema variations by loading necessary metadata at runtime, minimizing required code changes.
NFR3.4	Provide user-defined, flexible vocabulary mapping instead of fixed mappings.	Yes	Vocabulary mappings are stored in the RDA and loaded dynamically, avoiding hard-coded mappings.
NFR3.5	Use a metadata repository for storing and updating attribute and concept information.	Yes	Metadata about attributes is stored in the RDA. OMOP CDM concept data is accessed directly from the OMOP CDM database during execution.
NFR4.1	Ensure the ETL process can scale to large datasets using batching, parallelism, or distributed computing.	Partially	Batch inserts are implemented, and downstream transformations begin as soon as upstream data becomes available. However, full parallel or distributed execution is not implemented, although supported by PDI.
NFR5.1	Implement robust error handling and logging with the ability to resume or restart ETL after failures.	Partially	Logging is implemented, and some error handling is in place (e.g., missing values). However, recovery from intermediate failures may require manual cleanup.
NFR5.2	Maintain detailed logs of ETL performance, processing status, and errors for auditing and monitoring.	Yes	The system logs processing steps, timestamps, and encountered errors.
NFR5.3	Provide automatic recovery mechanisms to resume the ETL process after failure without manual intervention.	Partially	The ETL process resets the OMOP CDM database. However, recovery from intermediate failures may still need manual intervention.
NFR6.1	Enable EAV data processing, including pivoting/unpivoting to fit OMOP CDM format.	Yes	The prototype processes EAV-modeled data through pivoting during the custom transformations.

ID	Requirement Summary	Fulfilled	Evaluation Comments
NFR6.2	Handle EAV-specific challenges such as sparse data and heterogeneous attribute types.	Yes	The ETL dynamically loads attribute metadata at runtime to handle EAV characteristics.
NFR7.1	Integrate with scheduling tools to support automated or event-driven ETL execution.	Partially	PDI supports scheduling, but automated execution was not yet implemented in the prototype.

Table 6.2: Evaluation of the non-functional requirements for the prototypical implementation of the ETL process.

The evaluation demonstrates that the implemented prototype satisfies a broad range of non-functional requirements. Key aspects such as modularity, maintainability, adaptability, and data quality are fully supported. Several performance- and fault-tolerance-related features are partially implemented, providing a foundation for further optimization. Although automation, parallelism, and robust recovery mechanisms are not yet fully realized, the prototype establishes a solid and extensible base for a reliable ETL process to the OMOP CDM.

6.5 Conclusion

The evaluation demonstrated that the developed ETL prototype is capable of transforming heterogeneous EAV-based healthcare data from the research database of the MedUni Vienna into the standardized structure of the OMOP CDM. Across both evaluation scenarios, the transformation framework demonstrated reusability, while dataset-specific adaptations were handled through use case-specific extensions. At the same time, the scenarios highlighted common challenges that are likely to recur in future applications. These included limitations in metadata handling and technical constraints of the execution environment. Such issues emphasize the need for continued refinement of the framework.

The two scenarios also underscored how the characteristics of the source data directly affect performance. The microbiology HOB dataset, with fewer attributes and a narrower clinical scope, could be processed efficiently with little custom logic. In contrast, the oncology BCB dataset involved broader patient trajectories, more heterogeneous attributes, and higher data volumes, which led to substantially longer runtimes and a greater reliance on template-specific transformations. These differences illustrate that the scalability of the prototype is not only a matter of technical performance but also of managing diversity.

Overall, the evaluation confirms that the prototype fulfills its functional and non-functional requirements. At the same time, the findings stress the importance of continued work on performance optimization and metadata support. In sum, the prototype provides a strong foundation for reusable healthcare data transformation, while pointing to clear directions for future enhancement.

CHAPTER 7

Discussion

This chapter discusses the key findings of the thesis in the context of the research objectives and questions outlined earlier. The aim is to reflect on the development, design, and evaluation of a generic ETL framework capable of transforming healthcare data from the EAV data model of the MedUni Vienna into the OMOP CDM.

The analysis is structured around the research questions, focusing on the design decisions that enabled a generic ETL architecture, as well as the practical outcomes observed in the two evaluation scenarios: HOB and BCB. Special attention is given to the system's handling of structural variability, mapping completeness, and performance across different datasets.

Beyond the technical aspects, the discussion also addresses the broader implications of standardizing EAV-modeled healthcare data, the trade-offs inherent in developing generic transformation logic, and the limitations that may inform future improvements.

7.1 Research Questions

7.1.1 Research Question RQ1 – ETL requirements

> What are the specific requirements for the ETL process to ensure the successful transformation of healthcare data from the EAV model into the OMOP CDM?

This research question was addressed through an extensive literature review, presented in Chapter 3, which identified a set of functional and non-functional requirements essential for designing a generic, reusable ETL system tailored to transforming data from the EAV-based source at the MedUni Vienna into the OMOP CDM. These requirements, discussed in detail in Section 5.1, reflect both the technical constraints of the source and

target data models, as well as the quality attributes expected of modern ETL systems used in healthcare data integration.

The functional requirements focus on the core operations the ETL process must support, including:

- Accurate extraction and interpretation of EAV-modeled data,
- Dynamic application of mappings for attribute and vocabulary standardization,
- Support for OMOP CDM-specific constraints (e.g., referential integrity, domain-specific table structures),
- Automation, validation, and logging throughout the transformation process.

These requirements ensure that the ETL system can transform structurally flexible, sparse, and semantically rich EAV data into a standardized, relational format without loss of critical meaning. The emphasis on dynamic mapping and validation further ensures adaptability and data quality, which are prerequisites for downstream use in clinical research and analytics.

The functional requirements were systematically implemented in the prototype and later evaluated against real-world use cases. The evaluation confirmed that all functional requirements were fulfilled or partially fulfilled, with some areas, such as error recovery and advanced mapping logic, offering room for future enhancement. Nevertheless, even in its prototypical form, the ETL system demonstrated the ability to transform EAV-based data into OMOP CDM-compliant outputs reliably.

The non-functional requirements, on the other hand, capture architectural qualities such as modularity, reusability, configurability, scalability, and fault tolerance. These were derived by synthesizing recommendations from existing ETL tools, OHDSI community guidelines (e.g., the Book of OHDSI), and design principles from related research. For instance, requirements such as dynamic configuration loading (NFR 3.1–3.5) and module independence (NFR 2.1–2.3) are critical to achieving a generic system that can be adapted across multiple projects with minimal changes.

The derived non-functional requirements were validated during implementation and systematically evaluated. The evaluation showed that all non-functional requirements were fulfilled or partially fulfilled. Notably, the system's architecture supported key quality attributes such as reusability, configurability, and extensibility, confirming the validity of the requirement set.

The identification and operationalization of these requirements directly influenced the system design and implementation strategy. Rather than building a fixed ETL process for a specific dataset, the prototype was designed to generalize across use cases by supporting metadata-driven transformations, modular architecture, and adaptable configuration. This approach was crucial for ensuring that the system could be reused for multiple

clinical domains, as demonstrated in the evaluation scenarios on HOB surveillance and BCB.

In conclusion, the requirements identified through the literature review provided a robust foundation for designing a generic ETL framework for EAV to OMOP CDM transformation. Their implementation and evaluation confirm the practical relevance and effectiveness of these approaches in guiding the development of interoperable, high-quality data transformation processes in healthcare informatics.

7.1.2 Research Question RQ2 – ETL process design

> How can an effective and generic transformation of healthcare data from the EAV model into the OMOP CDM be achieved?

This research question aimed to explore how a scalable, maintainable, and conceptually sound ETL process can be developed to transform healthcare data stored in the EAV model of the MedUni Vienna into the structure defined by the OMOP CDM.

To address this challenge, a layered ETL architecture was designed that integrates both fixed and dynamic transformation strategies. The dual approach accommodates the heterogeneous nature of the source data and supports consistent, reproducible mapping to the OMOP CDM. The architecture is built around modularity and metadata-driven logic, enabling flexibility and maintainability. The detailed concept of the architecture is discussed in Section 5.2.

The fixed transformations are predefined, rule-based mappings for OMOP CDM tables with predictable and stable structures, such as person, visit_occurrence, care_site, or observation_period. Since these tables reflect structural or administrative data that do not follow the EAV pattern, a hardcoded approach was sufficient and appropriate. The transformation logic here is consistent across use cases and explicitly tailored to the data source. In contrast, the custom transformations were designed to handle clinical data captured in the EAV model (e.g., measurements, observations, procedures). This component uses template-driven mapping and relies on metadata tables that define how each form field or value set in the EAV schema maps to a target OMOP CDM record. This architecture enables the transformation process to be generalized and reused across use cases without requiring new code to be written for each data element. Instead, adding new mappings or supporting new data requires only updates to the metadata tables.

Crucially, this design introduces a semantic mapping layer, where form fields and value sets are linked to standardized terminologies (e.g., SNOMED CT, LOINC). These mappings are maintained in dedicated semantic mapping tables in the source system, enabling a separation of structure and semantics.

One of the core advantages of this approach is its generality and scalability. The dynamic transformation framework enables the ETL pipeline to support new data structures without requiring modifications to the transformation logic itself. Only updates to the

mapping metadata are required, which reduces implementation effort and enables the system to evolve in line with clinical documentation practices.

This design also significantly improves maintainability. Since transformation rules are externalized into metadata tables, updates can be made without requiring code changes. This separation simplifies governance and allows semantic experts to participate in configuration and mapping, provided they understand the OMOP CDM data model and vocabulary standards.

Another key benefit is semantic interoperability. Through the use of standardized vocabularies such as SNOMED CT, LOINC, and ICD-10, the system ensures that clinical meaning is preserved and aligned with the OHDSI standardized vocabularies.

Lastly, the architecture promotes a clear separation of concerns. Transformation logic, metadata, and terminology mappings are handled in distinct layers. This modular design facilitates easier validation, as well as improved long-term maintainability. Semantic mappings can evolve independently of structural changes.

While the architecture addresses the transformation problem effectively, its success relies heavily on the completeness and correctness of the structural and semantic mappings. Inconsistencies or missing codes in the metadata layer can lead to inaccurate or incomplete OMOP CDM records. Moreover, the flexibility of the EAV model means that edge cases, complex data structures, and non-standard usage of form fields still require careful consideration during the mapping process. As a result, ongoing validation and quality control processes are essential to ensure data reliability.

In summary, research question RQ2 is addressed through the development of a modular ETL architecture that leverages both fixed and dynamic transformation strategies. By combining metadata-driven mapping, template-based logic, and a semantic enrichment layer, the approach offers a flexible yet rigorous solution for transforming healthcare data from the EAV model of the MedUni Vienna into the OMOP CDM. This architecture lays the foundation for scalable and semantically aligned secondary use of clinical data.

7.1.3 Research Question RQ3 – Use of the generic ETL code base system

> To what extent can the developed ETL process for the transformation of healthcare data from the EAV model into the OMOP CDM be extended or adapted to specific use cases?

This research question was addressed through the application of the developed ETL framework to two evaluation scenarios: HOB surveillance and BCB, which are described in detail in Sections 6.1 and 6.2. Both datasets originated from the EAV-based source system used at the MedUni Vienna, but they differed significantly in terms of data structure, complexity, and scope. This diversity provided a robust foundation for assessing the system's generalizability and flexibility.

The ETL framework was designed to be generic and reusable, aiming to decouple use-case-specific logic from the underlying transformation mechanics. The successful application of the same code base across both evaluation scenarios confirms that this objective was largely achieved. Core components of the pipeline, including the interpretation of structural and semantic mappings, the integration of standard vocabularies, and the management of concept relationships, were reused without modification. The shared code base could ingest new templates and configurations defined via metadata tables without requiring changes to the underlying logic, demonstrating a high degree of configurability.

The evaluation scenarios validated that custom transformation rules could be introduced at the template level, enabling localized adaptation without affecting the global transformation logic. For example, both use cases relied on custom template definitions and mapping configurations tailored to the clinical context. However, these were managed as metadata and did not necessitate structural changes to the pipeline itself.

Despite the general success in applying the system generically, several extensions and customizations were required to accommodate case-specific requirements. These adjustments did not undermine the integrity of the core system but instead highlighted areas where domain-specific logic was necessary. Notably, in the BCB use case, several transformation templates required enhancements to support:

- Conditional row generation, where the number of OMOP records created depended on attribute values within a source template.
- Fallback mechanisms for date derivation, where the absence of a primary date field requires the use of secondary fields in a defined cascade.
- Structural mapping adjustments, particularly for fixed transformations, which differed in structure from the assumptions encoded in the default transformation logic.
- Handling of incomplete data, including the exclusion and logging of rows where mandatory fields such as *_concept_id were missing.

These extensions were made possible through the system's modular architecture, which separates reusable components (e.g., caching, data type casting, vocabulary lookup) from configurable transformation steps. Importantly, these adaptations were localized and maintainable, indicating that the framework is not only extensible but also robust to evolving use-case requirements.

The two evaluation scenarios also differed significantly in terms of data volume and transformation runtime. While the HOB use case involved relatively compact and homogeneous data, the BCB use case featured a larger dataset with greater structural heterogeneity. The increased data volume and complexity in the BCB use case resulted in longer transformation times, particularly for custom transformation steps. The increase in runtime, especially in the BCB case, underscores the importance of performance tuning

and highlights potential limitations when scaling to large datasets or when processing a high number of templates. Nevertheless, caching mechanisms and a structured deployment process contributed to maintaining manageable execution times even in the more demanding scenario.

These performance differences do not diminish the adaptability of the framework but rather point to areas for future improvement, such as optimizing transformation logic, supporting parallel execution, and refining caching strategies for large-scale deployments.

Overall, the evaluation demonstrates that the ETL framework provides a solid foundation for transforming EAV-based clinical data into the OMOP CDM. The separation of configuration (e.g., mappings, templates, metadata) from transformation logic allows for flexible reuse and adaptation across multiple projects. Although each use case required some degree of customization, these changes were handled within the framework's extensibility model, indicating that the core system does not need to be rewritten or restructured for new applications. The use of metadata-driven templates, semantic mapping tables, and modular transformation steps makes the system applicable to a wide range of EAV-modeled datasets.

In conclusion, the developed ETL system effectively supports the transformation of heterogeneous healthcare data from an EAV model into the OMOP CDM. The results from both evaluation scenarios demonstrate that the system is not only reusable but also adaptable to a broad spectrum of data contexts. While use-case-specific logic must be expected in real-world deployments, the existing architecture supports such extensions cleanly and maintainably. These features position the framework as a promising candidate for further institutional adoption and broader applications in clinical data standardization initiatives.

7.1.4 Research Question RQ4 – Evaluation of the ETL process

> How does the developed generic ETL code base perform regarding data quality and adaptability?

This research question was addressed during the Evaluation Phase of the thesis (see Chapter 6), where the implemented ETL prototype was systematically assessed against the functional and non-functional requirements defined in the Analysis Phase. In addition to the formal evaluation of the requirements, two evaluation scenarios were conducted to examine the system's ability to transform real-world datasets. This combined approach allowed the evaluation to cover both technical conformance and practical usability.

The evaluation confirms that the prototype is capable of producing high-quality transformations from the flexible, sparse, and semantically rich EAV data model into the structurally strict and standardized OMOP CDM. Several design features contributed to this outcome:

- Vocabulary and concept mappings are dynamically loaded from external metadata sources rather than hard-coded, ensuring flexibility and maintainability and allowing mapping logic to be reused across datasets without changes to the implementation.
- Constraint validation mechanisms are integrated into the OMOP CDM database. These include checks for mandatory fields, correct data types, and referential integrity via foreign keys. Such safeguards prevent invalid or incomplete records from being entered into the target schema.
- Logging and error handling are built into the pipeline. The system records both operational steps and transformation issues (e.g., rows with missing mandatory values), providing transparency and aiding debugging efforts.

These features were evaluated using a combination of implementation review and practical testing. During development, the implementation was continuously checked against the intended architectural principles to ensure compliance with the design goals. In addition, the evaluation scenarios verified that all relevant OMOP CDM tables were populated, concept mappings were applied correctly, and structural constraints were upheld.

Adaptability was demonstrated through the prototype's application to two clinical domains: the HOB evaluation scenario (microbiology data) and the BCB evaluation scenario (oncology data). Both datasets were sourced from the same institutional EAV data model but differed significantly in scope, attribute richness, and semantic structure. The HOB dataset included a limited, well-defined set of attributes, while the BCB dataset featured greater structural variety and heterogeneity. Despite these differences, the ETL system adapted to both scenarios without modifying the core logic. Case-specific requirements, such as specialized date handling, conditional row generation, or structural mapping differences, were addressed through metadata configurations or customized transformations, without affecting the overall pipeline.

Several architectural decisions supported this flexibility:

- A modular design, where each transformation step targets a specific OMOP CDM table or logical unit, which promotes reusability and simplifies maintenance.
- Dynamic configuration loading, allowing mapping rules, attribute definitions, and transformation parameters to be defined externally and injected at runtime.

The evaluation confirmed that the system is reusable across clinical domains with only minimal adaptation. In both evaluation scenarios, most of the ETL logic was reused, requiring only limited modifications. However, some manual effort was still required to prepare mapping metadata, align terms with the OHDSI standardized vocabularies, and verify domain-specific assumptions. While these tasks are manageable within the current framework, additional tools, such as metadata editors or vocabulary alignment aids, could further reduce preparation time and configuration effort for new datasets.

In summary, the evaluation demonstrates that the developed ETL prototype effectively meets its core objectives: high data quality and strong adaptability. Its ability to produce consistent, standards-aligned OMOP CDM transformations across diverse datasets, without duplicating transformation logic, underscores the success of its design. The combination of modular architecture, external configuration, and integrated validation makes the system a practical and extensible solution for transforming EAV-based healthcare data into the OMOP CDM.

7.2 Limitations

While the developed ETL framework demonstrated strong adaptability and reusability across two distinct healthcare datasets, several limitations were identified during the design, implementation, and evaluation phases. These limitations highlight both the current boundaries of the system and opportunities for future refinement.

Manual Effort for Mapping Preparation. A central feature of the framework is its reliance on metadata-driven structural mappings and semantic standardization using controlled vocabularies. However, creating and maintaining these mappings required manual curation and domain knowledge. In both evaluation scenarios, substantial effort was invested in preparing semantic and structural mappings, particularly for the oncology dataset, which included diverse and specialized clinical concepts. The initial configuration effort can be a barrier to adoption, especially in projects with limited informatics or clinical coding resources.

Limitations in Transformation Metadata and Configuration Tooling Although the use of metadata-driven configuration significantly reduced the need for custom code, the process of creating, validating, and maintaining mapping tables was still manual and error-prone. The absence of dedicated tooling for authoring and validating metadata (e.g., template editors, mapping GUIs, or terminology alignment aids) posed challenges, particularly when adapting the system to complex forms such as those in the BCB evaluation scenario. Manual configuration also increased the risk of introducing inconsistencies or omissions.

Incomplete Concept Coverage in the RDA Platform. Although the transformation framework was designed to integrate with the RDA platform, not all required structural components were available in its existing implementation at the time of development. This lack of concept coverage constrained the ability to perform transformations purely within the native RDA platform environment. As a result, necessary mapping tables had to be implemented within a personal schema in the RDA platform database. These additions enabled the integration of required metadata into the transformation pipeline. This workaround introduces additional implementation overhead when the ETL code base is reused, as the tables need to be provided until the RDA platform supports the full range of mapping elements needed for the ETL process to the OMOP CDM.

Distinction Between Generic Logic and Use-Case Specificity. A key design goal of the framework was to maintain a clear separation between generic transformation logic and use-case-specific adaptations. While this goal was largely achieved, in practice it was challenging to define a consistent boundary between what should be handled generically and what should be treated as custom logic. Specific recurring patterns, such as more elaborate fallback rules for missing dates or filtering records based on a specific value, might arguably warrant inclusion in the generic framework. However, due to their tight coupling with clinical context or form structure, they were implemented as template-specific extensions. This tension indicates that further abstraction or pattern generalization may be possible.

Execution Environment and Operational Constraints. The pipeline was executed locally on a developer's machine using ad hoc job scheduling methods. This setup, while sufficient for prototyping and evaluation, limits reproducibility, automation, and scalability. Without integration into a formal scheduling or orchestration framework (e.g., cron-based automation), the system cannot be reliably deployed in a production environment or operated by non-technical users. Moreover, local execution hinders long-term monitoring. Future iterations of the system would benefit from integration into institutional ETL infrastructure.

Scalability and Performance Overhead. While performance was generally acceptable for moderate-sized datasets, the transformation of larger and more complex datasets (as in the BCB case) introduced noticeable overhead. In particular, execution time increased significantly when many custom transformation steps or large template sets were involved. Although caching strategies mitigated some of these effects, further optimization would be needed for high-throughput environments. Additionally, parallelism or distributed processing is not currently supported, which could limit scalability in data-intensive applications.

Dependence on Data Quality and Stability in the Source System. The effectiveness of the ETL process is heavily dependent on the quality and completeness of the source data. Issues such as missing values resulted in the exclusion of certain records. Although logging and validation mechanisms are in place to catch and record such issues, the framework cannot compensate for poor data quality in the source system. Beyond data quality, the stability of the source system's structure is equally critical. Changes in the data model of the source system or in upstream systems from which it ingests data must be communicated promptly. Otherwise, such changes risk breaking parts of the transformation pipeline or, more critically, may lead to the silent production of incorrect results, undermining the reliability of downstream analyses. This limitation is particularly relevant when considering deployment in heterogeneous or evolving clinical databases.

Indirect Access to Source Data. The ETL process developed in this thesis does not extract data directly from the clinical routine system of the AKH. Instead, it operates on the research database of the MedUni Vienna, which is derived from the routine system. Consequently, the data has already undergone at least one transformation before entering the pipeline. Such intermediate processing inevitably carries the risk of information loss or reduced granularity, which cannot be recovered downstream. While direct access to the routine system would mitigate this issue, it is rarely feasible in practice. Routine systems are primarily designed to support clinical workflows rather than research, and introducing additional extraction processes can pose operational and regulatory risks. As a result, the framework inherits potential limitations of the research database, which must be considered when interpreting results or deploying the system in other settings.

Data Not Yet Used in Downstream Analyses A primary motivation for transforming healthcare data into the OMOP CDM is to enable secondary uses such as population health studies, quality improvement, or federated research. In this thesis, however, the transformed data were not yet used for downstream analytical tasks such as cohort characterization or outcome analysis. As a result, the completeness and analytical utility of the transformed data could not be fully assessed. Without empirical validation through real-world research scenarios, it remains uncertain whether the current mappings and transformation rules are sufficient for reliable and reproducible analytics. This limitation is partly mitigated by a preceding project in which patient cohorts for the HOB evaluation scenario were defined and validated in Atlas using data generated by a hardcoded ETL process based on the same source dataset. While the new generic pipeline produces structurally equivalent outputs, cohort definitions have not yet been re-executed on the updated dataset. Full empirical validation in real-world research scenarios, therefore, remains an open area for future work.

CHAPTER 8

Conclusion and Future Work

This thesis presented the design, implementation, and evaluation of a generic ETL framework for transforming healthcare data modeled in the EAV data model of the MedUni Vienna into the standardized OMOP CDM.

The work was motivated by the need to bridge the structural and semantic gap between a flexible research database system and the highly normalized, concept-driven OMOP CDM, a cornerstone for achieving semantic interoperability and large-scale observational research.

To address this challenge, a modular and metadata-driven ETL prototype was developed. The system was designed to be extensible, configurable, and reusable across diverse datasets, allowing for the transformation of heterogeneous EAV-based datasets into the OMOP CDM. The system supports dynamic configuration through external metadata, ensuring extensibility across use cases. These design choices were guided by functional and non-functional requirements identified through literature review, domain expert input, and practical considerations. The domain expert contributed during the design phase through iterative discussions of architectural decisions, data mapping strategies, and workflow organization, ensuring that the resulting system aligns with domain-specific constraints and best practices in healthcare data integration.

The prototype was evaluated through two real-world evaluation scenarios: a HOB surveillance project and a BCB initiative. These studies illustrated the system's ability to handle different types of EAV-modeled data, including microbiology results and oncology care trajectories. Despite substantial differences in data structure, vocabulary use, and volume, the ETL system successfully transformed both datasets into the OMOP CDM with minimal changes to its core logic.

The evaluation demonstrated that the ETL framework fulfills its primary goals: producing high-quality, standards-compliant OMOP CDM data from EAV sources and supporting

reuse across heterogeneous use cases. The findings also underscore the benefits of metadata abstraction and modular architecture in facilitating adaptability and maintainability.

Nonetheless, several limitations were identified, including incomplete integration with the RDA platform, performance bottlenecks during large-scale transformations, and manual mapping work. These limitations point to promising avenues for future work, including improved infrastructure for job scheduling, enhanced support for complex mappings, better metadata tooling, and performance optimization.

In conclusion, this thesis contributes a practical and extensible approach to standardizing healthcare data using the OMOP CDM, offering a foundation for further development toward automated, reliable, and scalable ETL processes. As healthcare institutions increasingly seek to harmonize their data assets for research, the methods and insights presented here can support broader adoption of OMOP CDM-based architectures and foster collaboration within the global OHDSI community.

Future Work

While the developed ETL prototype demonstrates flexibility across two use cases, several areas offer opportunities for further refinement.

Identification and Generalization of Recurring Transformation Patterns. A promising area for future work is the systematic identification of recurring transformation patterns that currently require use-case-specific implementation. During the evaluation scenarios, certain logic, such as conditional filtering based on attribute values, appeared in multiple templates. Currently, this needs to be implemented as a custom extension. By analyzing these repetitions across different use cases, it might be possible to extract generalizable patterns and incorporate them into the core ETL framework. Incorporating such patterns would reduce duplication, simplify future adaptations, and improve maintainability.

Automation of Metadata Preparation and Validation. To reduce manual effort and improve reliability, tools could be developed to assist with the creation, validation, and management of structural and semantic mapping. Features could include automatic extraction of attributes, validation against vocabulary standards, and visual interfaces for configuration.

Performance Optimization and Parallel Execution. The BCB evaluation scenario revealed long execution times, particularly for the custom transformations. Optimization efforts should focus on reducing bottlenecks and carefully investigating whether parallelization can help accelerate transformation times and better utilize available system resources.

Incremental Loading. Currently, each ETL run rewrites the complete dataset in the target schema. Rewriting the entire dataset simplifies implementation and avoids potential inconsistencies, but may become inefficient as data volumes grow. In scenarios where the source database accumulates large amounts of data over time, incremental loading, i.e., transferring only new or modified records since the last run, would be more scalable and resource-efficient. Designing robust incremental strategies requires reliable change tracking in the source system and careful handling of updates and deletions, which were not necessary in the present evaluation setup. Nevertheless, developing such mechanisms would be an important step toward ensuring the framework's scalability in larger deployments.

Increased Fault Tolerance. The ETL pipeline could be made more robust against data inconsistencies, unexpected value formats, or missing entries. Adding mechanisms for effectively handling such issues without halting execution, while still logging warnings, would make the system more suitable for use in diverse and less curated data environments.

Near-Real-Time Data Transformation. An extension of the current framework could involve enabling near-real-time data transformation, where newly generated records are directly ingested into the OMOP CDM. While this approach would ensure that analytical datasets remain continuously up to date, it introduces significant complexity with respect to system integration, monitoring, and error handling. Moreover, in many research and surveillance contexts, near-real-time access to transformed data may not be strictly necessary, as periodic batch loading (e.g., nightly) can provide sufficient timeliness while reducing operational overhead. Future work should therefore assess whether the use cases at hand justify the additional effort and system load required for near-real-time pipelines.

Vocabulary Updates and Mapping Maintenance. As the OHDSI standardized vocabularies are updated regularly, future work should explore strategies for handling vocabulary changes. These strategies include detecting changes that affect existing mappings, identifying obsolete or remapped concepts, and updating transformation metadata accordingly. Tooling to support this process could help ensure long-term compatibility and semantic correctness.

Data Privacy and Pseudonymization. To comply with data protection regulations and facilitate broader data sharing, pseudonymization mechanisms could be integrated into the ETL process. Integrating such mechanisms would involve removing identifiable information while preserving analytical utility.

Data Quality and Plausibility Checks. Additional data quality measures tailored to specific clinical domains could further improve the reliability of the transformed data. For instance, hospitalization records often follow predictable temporal patterns, such as a discharge date not preceding the admission date, which are not yet enforced by

the general ETL logic. Incorporating domain-specific plausibility checks, informed by clinical workflows and medical logic, would help detect anomalies early and prevent the propagation of implausible records into the OMOP CDM. Embedding such checks into the pipeline would not only enhance data quality but also increase trust in the data for downstream analytical use.

User Authentication and Access Control. If the transformed data is exposed through analytical platforms such as Atlas, access control mechanisms must be implemented to ensure security and privacy. Future extensions should support user authentication and fine-grained access control to ensure data privacy and compliance with institutional policies and regulations.

Ultimately, this thesis demonstrated that the transformation of healthcare data from an EAV -based model into the OMOP CDM can be addressed through a generic, reusable ETL framework. By combining modular design with metadata-driven logic, the developed prototype demonstrated adaptability across different datasets while maintaining compliance with technical and semantic requirements. At the same time, the evaluation highlighted limitations and areas where further refinement is needed. Taken together, the work establishes a foundation for future research and development, while contributing a concrete step toward enabling interoperable, large-scale use of healthcare data.

List of Figures

List of Tables

Acronyms

Achilles Automated Characterization of Health Information at Largescale Longitudinal Evidence Systems

ADR adverse drug reaction

AKH University Hospital Vienna (German: Allgemeines Krankenhaus der Stadt Wien)

AKIM AKH Informationsmanagement

API application programming interface

BCB breast cancer benchmarking

BPMN Business Process Model Notation

CAUTI catheter-associated urinary tract infection

CDM Common Data Model

CDS Common Core Data Set

CER Comparative Effectiveness Research

CPN Colored Petri Nets

CSC Common Skin Commensal

CSV Comma-Separated Values

DML Data Manipulation Language

DQD Data Quality Dashboard

EAV Entity-Attribute-Value

ECDC European Centre for Disease Prevention and Control

EHR electronic health record

ELT Extract, Load, Transform

EMD Entity Modeling Diagram

ETL Extract, Transform, Load

EU European Union

EUHA European University Hospital Alliance

EUSOMA European Society of Breast Cancer Specialists

FHIR Fast Healthcare Interoperability Resources

FR Functional Requirement

GDPR General Data Protection Regulation

GIS Geographic Information Systems

GUI Graphical User Interface

H2O Health Outcomes Observatory

HAI healthcare-associated infection

HIS hospital information system

HOB hospital-onset bacteremia and fungemia

ICD-10 International Classification of Diseases, 10th Revision

ICHOM International Consortium for Health Outcomes Measurement

ICN infection control nurse

ICP infection control practitioner

ICU intensive care unit

LOINC Logical Observation Identifiers Names and Codes

LOS length of stay

MDA Model-Driven Architecture

MDD Model-Driven Development

MDR Metadata Repository

MDS Minimal Data Set

MedUni Vienna Medical University of Vienna

MedUni Wien Medizinische Universität Wien

MII Medical Informatics Initiative

NFR Non-functional Requirement

NHSN National Healthcare Safety Network

OHDSI Observational Health Data Sciences and Informatics

OMOP Observational Medical Outcomes Partnership

PDI Pentaho Data Integration

RDA platform Research Documentation & Analysis platform

SNOMED CT Systematized Nomenclature of Medicine Clinical Terms

SQL structured query language

SSI surgical site infection

TPC-DI Transaction Processing Performance Council – Data Integration

UML Unified Modeling Language

UZ Leuven University Hospitals Leuven

VAP ventilator-associated pneumonia

WHO World Health Organization

Bibliography

[1] G. Hripcsak, J. D. Duke, N. H. Shah, C. G. Reich, V. Huser, M. J. Schuemie, M. A. Suchard, R. W. Park, I. C. K. Wong, P. R. Rijnbeek, J. van der Lei, N. Pratt, G. N. Norén, Y.-C. Li, P. E. Stang, D. Madigan, P. B. Ryan, „Observational Health Data Sciences and Informatics (OHDSI): Opportunities for observational researchers“, eng, *Studies in health technology and informatics*, vol. 216, pp. 574–8, 2015. DOI: 10.3233/978-1-61499-564-7-574 (cited on pp. 1–3, 7, 8, 12, 14, 23, 30).

[2] „ISO/TR 20514:2005(E): Health informatics – electronic health record – definition, scope and context“, International Organization for Standardization, ISO/TC, Oct. 15, 2005 (cited on pp. 2, 11).

[3] E. A. Voss, C. Blacketer, S. van Sandijk, M. Moinat, M. Kallfelz, M. van Speybroeck, D. Prieto-Alhambra, M. J. Schuemie, P. R. Rijnbeek, „European health data & evidence network—learnings from building out a standardized international health data network“, *Journal of the American Medical Informatics Association*, vol. 31, no. 1, pp. 209–219, Nov. 2023. DOI: 10.1093/jamia/ocad214 (cited on p. 2).

[4] L. Frank, S. K. Andersen, „Evaluation of different database designs for integration of heterogeneous distributed electronic health records“, in *IEEE/ICME International Conference on Complex Medical Engineering*, IEEE, Jul. 2010, pp. 204–209. DOI: 10.1109/iccme.2010.5558844 (cited on p. 2).

[5] S. Batra, S. Sachdeva, S. Bhalla, „Entity attribute value style modeling approach for archetype based data“, *Information*, vol. 9, no. 1, p. 2, Dec. 2017. DOI: 10.3390/info9010002 (cited on pp. 2, 8).

[6] D. Löper, M. Klettke, I. Bruder, A. Heuer, „Enabling flexible integration of healthcare information using the entity-attribute-value storage model“, *Health Information Science and Systems*, vol. 1, no. 1, Feb. 2013. DOI: 10.1186/2047-2501-1-9 (cited on pp. 2, 3, 7–9).

[7] V. Kalokyri, H. Kondylakis, S. Sfakianakis, K. Nikiforaki, I. Karatzanis, S. Mazzetti, N. Tachos, D. Regge, D. I. Fotiadis, K. Marias, M. Tsiknakis, „MI-Common Data Model: Extending Observational Medical Outcomes Partnership-Common Data Model (OMOP-CDM) for registering medical imaging metadata and subsequent curation processes“, *JCO Clinical Cancer Informatics*, no. 7, Sep. 2023. DOI: 10.1200/cci.23.00101 (cited on pp. 2, 12).

[8] C. Reich, A. Ostropolets, P. Ryan, P. Rijnbeek, M. Schuemie, A. Davydov, D. Dymshyts, G. Hripcsak, „OHDSI standardized vocabularies—a large-scale centralized reference ontology for international data harmonization“, *Journal of the American Medical Informatics Association*, vol. 31, no. 3, pp. 583–590, Jan. 2024. DOI: 10.1093/jamia/ocad247 (cited on pp. 2, 12, 30).

[10] G. Luo, L. J. Frey, „Efficient execution methods of pivoting for bulk extraction of entity-attribute-value-modeled data“, *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 644–654, Mar. 2016. DOI: 10.1109/jbhi.2015.2392553 (cited on pp. 2, 3, 7, 26, 29, 30, 37, 38).

[11] R. Makadia, P. B. Ryan, „Transforming the premier perspective® hospital database to the OMOP common data model“, *eGEMs (Generating Evidence & Methods to improve patient outcomes)*, vol. 2, no. 1, p. 15, Nov. 2014. DOI: 10.13063/2327-9214.1110 (cited on p. 2).

[12] Y. Peng, E. Henke, I. Reinecke, M. Zoch, M. Sedlmayr, F. Bathelt, „An ETL-process design for data harmonization to participate in international research with german real-world data based on FHIR and OMOP CDM“, *International Journal of Medical Informatics*, vol. 169, p. 104 925, Jan. 2023. DOI: 10.1016/j.ijmedinf.2022.104925 (cited on pp. 2, 25–27, 38).

[13] Y. Yu, N. Zong, A. Wen, S. Liu, D. J. Stone, D. Knaack, A. M. Chamberlain, E. Pfaff, D. Gabriel, C. G. Chute, N. Shah, G. Jiang, „Developing an ETL tool for converting the PCORnet CDM into the OMOP CDM to facilitate the COVID-19 data integration“, *Journal of Biomedical Informatics*, vol. 127, p. 104 002, Mar. 2022. DOI: 10.1016/j.jbi.2022.104002 (cited on p. 2).

[14] J. C. Quiroz, T. Chard, Z. Sa, A. Ritchie, L. Jorm, B. Gallego, „Extract, transform, load framework for the conversion of health databases to OMOP“, *PLOS ONE*, vol. 17, no. 4, T. M. Deserno, Ed., e0266911, Apr. 2022. DOI: 10.1371/journal.pone.0266911 (cited on pp. 3, 25, 30, 37, 38).

[16] G. Duftschmid, W. Gall, E. Eigenbauer, W. Dorda, „Management of data from clinical trials using the archimed system“, *Medical Informatics and the Internet in Medicine*, vol. 27, no. 2, pp. 85–98, Jan. 2002. DOI: 10.1080/1463923021000014158 (cited on pp. 7, 10).

[17] B. H. de Mello, S. J. Rigo, C. A. da Costa, R. da Rosa Righi, B. Donida, M. R. Bez, L. C. Schunke, „Semantic interoperability in health records standards: A systematic literature review“, *Health and Technology*, vol. 12, no. 2, pp. 255–272, Jan. 2022. DOI: 10.1007/s12553-022-00639-w (cited on pp. 7, 10).

[18] R. Kimball, *The data warehouse ETL toolkit, Practical techniques for extracting, cleaning, conforming, and delivering data* (Safari Books Online), J. Caserta, Ed. Indianapolis, IN: Wiley, 2004, 491 pp., ISBN: 9780764579233 (cited on pp. 8, 20, 23).

[19] Observational Health Data Sciences and Informatics, *The Book of OHDSI*. Jan. 11, 2021. Accessed: Aug. 23, 2025. [Online]. Available: https://ohdsi.github.io/TheBookOfOhdsi/ (cited on pp. 8, 16, 23, 25, 30, 37, 38, 46, 47).

[21] A. Kamau, W. Mwangi, „An enhanced entity-attribute-value data model for representing high dimensional and sparse healthcare data“, in *2013 IST-Africa Conference & Exhibition*, 2013, pp. 1–7 (cited on p. 8).

[22] P. Wegner, „Interoperability“, *ACM Computing Surveys*, vol. 28, no. 1, pp. 285–287, Mar. 1996. DOI: 10.1145/234313.234424 (cited on p. 10).

[23] K. Donnelly, „SNOMED-CT: The advanced terminology and coding system for ehealth“, en, *Stud. Health Technol. Inform.*, vol. 121, pp. 279–290, 2006 (cited on p. 11).

[26] R. Pastorino, C. D. Vito, G. Migliara, K. Glocker, I. Binenbaum, W. Ricciardi, S. Boccia, „Benefits and challenges of big data in healthcare: An overview of the european initiatives“, *European Journal of Public Health*, vol. 29, no. Supplement_3, pp. 23–27, Oct. 2019. DOI: 10.1093/eurpub/ckz168 (cited on p. 11).

[28] A. Amutha, P. A. Praveen, C. W. Hockett, T. C. Ong, E. T. Jensen, S. P. Isom, R. B. J. D'Agostino, R. F. Hamman, E. J. Mayer-Davis, R. P. Wadwa, J. M. Lawrence, C. Pihoker, M. G. Kahn, D. Dabelea, N. Tandon, V. Mohan, „Treatment regimens and glycosylated hemoglobin levels in youth with type 1 and type 2 diabetes: Data from search (united states) and ydr (india) registries“, *Pediatric Diabetes*, vol. 22, no. 1, pp. 31–39, Mar. 2020. DOI: 10.1111/pedi.13004 (cited on p. 16).

[29] C. W. Hockett, P. A. Praveen, T. C. Ong, A. Amutha, S. P. Isom, E. T. Jensen, R. B. D'Agostino, R. F. Hamman, E. J. Mayer-Davis, J. M. Lawrence, C. Pihoker, M. G. Kahn, V. Mohan, N. Tandon, D. Dabelea, „Clinical profile at diagnosis with youth-onset type 1 and type 2 diabetes in two pediatric diabetes registries: Search (united states) and ydr (india)“, *Pediatric Diabetes*, vol. 22, no. 1, pp. 22–30, Jan. 2020. DOI: 10.1111/pedi.12981 (cited on p. 16).

[30] H. Düsseldorf, G. Duftschmid, E. Presterl, „Transforming hospital-onset bacteraemia and fungaemia data to the OMOP common data model“, *CMI Communications*, May 12, 2025. DOI: 10.1016/j.cmicom.2025.105086 (cited on pp. 16, 86).

[31] R. Brauer, I. C. K. Wong, K. K. C. Man, N. L. Pratt, R. W. Park, S.-Y. Cho, Y.-C. Li, U. Iqbal, P.-A. A. Nguyen, M. Schuemie, „Application of a common data model (CDM) to rank the paediatric user and prescription prevalence of 15 different drug classes in south korea, hong kong, taiwan, japan and australia: An observational, descriptive study“, *BMJ Open*, vol. 10, no. 1, e032426, Jan. 2020. DOI: 10.1136/bmjopen-2019-032426 (cited on p. 16).

[32] O. I. Ogunyemi, D. Meeker, H.-E. Kim, N. Ashish, S. Farzaneh, A. Boxwala, „Identifying appropriate reference data models for comparative effectiveness research (CER) studies based on data from clinical information systems“, *Medical Care*, vol. 51, S45–S52, Aug. 2013. DOI: 10.1097/mlr.0b013e31829b1e0b (cited on p. 17).

[33] H. Shin, S. Lee, „An OMOP-CDM based pharmacovigilance data-processing pipeline (PDP) providing active surveillance for ADR signal detection from real-world data sources“, *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, May 2021. DOI: 10.1186/s12911-021-01520-y (cited on p. 17).

[34] J. Cho, S. C. You, S. Lee, D. Park, B. Park, G. Hripcsak, R. W. Park, „Application of epidemiological geographic information system: An open-source spatial analysis tool based on the OMOP common data model“, *International Journal of Environmental Research and Public Health*, vol. 17, no. 21, p. 7824, Oct. 2020. DOI: 10.3390/ijerph17217824 (cited on p. 18).

[35] D. R. Morales, M. M. Conover, S. C. You, N. Pratt, K. Kostka, T. Duarte-Salles, S. Fernández-Bertolín, M. Aragón, S. L. DuVall, K. Lynch, T. Falconer, K. van Bochove, C. Sung, M. E. Matheny, C. G. Lambert, F. Nyberg, T. M. Alshammari, A. E. Williams, R. W. Park, J. Weaver, A. G. Sena, M. J. Schuemie, P. R. Rijnbeek, R. D. Williams, J. C. E. Lane, A. Prats-Uribe, L. Zhang, C. Areia, H. M. Krumholz, D. Prieto-Alhambra, P. B. Ryan, G. Hripcsak, M. A. Suchard, „Renin–angiotensin system blockers and susceptibility to COVID-19: An international, open science, cohort analysis“, *The Lancet Digital Health*, vol. 3, no. 2, e98–e114, Feb. 2021. DOI: 10.1016/s2589-7500(20)30289-2 (cited on p. 18).

[36] R. Chen, P. Ryan, K. Natarajan, T. Falconer, K. D. Crew, C. G. Reich, R. Vashisht, G. Randhawa, N. H. Shah, G. Hripcsak, „Treatment patterns for chronic comorbid conditions in patients with cancer using a large-scale observational data network“, *JCO Clinical Cancer Informatics*, no. 4, pp. 171–183, Nov. 2020. DOI: 10.1200/cci.19.00107 (cited on p. 18).

[37] H. Lee, S. Kim, H.-W. Moon, H.-Y. Lee, K. Kim, S. Y. Jung, S. Yoo, „Hospital length of stay prediction for planned admissions using observational medical outcomes partnership common data model: Retrospective study“, *Journal of Medical Internet Research*, vol. 26, e59260, Nov. 2024. DOI: 10.2196/59260 (cited on p. 19).

[38] E. Krastev, P. Kovachev, S. Abanos, R. Krasteva, D. Tcharaktchiev, „Assessment of pharmacology costs in diabetes treatment using OMOP CDM: A nationally representative study“, *International Journal on Advances in Life Sciences*, vol. 16, no. 1 & 2, pp. 11–20, 2024 (cited on p. 19).

[39] E. Burn, S. C. You, A. G. Sena, K. Kostka, H. Abedtash, M. T. F. Abrahão, A. Alberga, H. Alghoul, O. Alser, T. M. Alshammari, M. Aragon, C. Areia, J. M. Banda, J. Cho, A. C. Culhane, A. Davydov, F. J. DeFalco, T. Duarte-Salles, S. DuVall, T. Falconer, S. Fernandez-Bertolin, W. Gao, A. Golozar, J. Hardin, G. Hripcsak, V. Huser, H. Jeon, Y. Jing, C. Y. Jung, B. S. Kaas-Hansen, D. Kaduk, S. Kent,

Y. Kim, S. Kolovos, J. C. E. Lane, H. Lee, K. E. Lynch, R. Makadia, M. E. Matheny, P. P. Mehta, D. R. Morales, K. Natarajan, F. Nyberg, A. Ostropolets, R. W. Park, J. Park, J. D. Posada, A. Prats-Uribe, G. Rao, C. Reich, Y. Rho, P. Rijnbeek, L. M. Schilling, M. Schuemie, N. H. Shah, A. Shoaibi, S. Song, M. Spotnitz, M. A. Suchard, J. N. Swerdel, D. Vizcaya, S. Volpe, H. Wen, A. E. Williams, B. B. Yimer, L. Zhang, O. Zhuk, D. Prieto-Alhambra, P. Ryan, „Deep phenotyping of 34,128 adult patients hospitalised with COVID-19 in an international network study“, *Nature Communications*, vol. 11, no. 1, Oct. 2020. DOI: 10.1038/s41467-020-18849-z (cited on p. 20).

[40] E. Henke, M. Zoch, Y. Peng, I. Reinecke, M. Sedlmayr, F. Bathelt, „Conceptual design of a generic data harmonization process for OMOP common data model“, *BMC Medical Informatics and Decision Making*, vol. 24, no. 1, p. 58, Feb. 2024. DOI: 10.1186/s12911-024-02458-7 (cited on pp. 23, 31, 38).

[41] T. C. Ong, R. Pradhananga, E. G. Holve, M. G. Kahn, „A framework for classification of electronic health data extraction-transformation-loading challenges in data network participation“, *eGEMs (Generating Evidence & Methods to improve patient outcomes)*, vol. 5, no. 1, p. 10, Jun. 2017. DOI: 10.5334/egems.222 (cited on p. 23).

[42] M. Patel, D. B. Patel, „Progressive growth of ETL tools: A literature review of past to equip future“, in *Rising Threats in Expert Applications and Solutions*. Springer Singapore, Oct. 2020, pp. 389–398, ISBN: 9789811560149. DOI: 10.1007/978-981-15-6014-9_45 (cited on p. 25).

[43] S. Bachir, A. Vengadeswaran, H. Storf, D. Kadioglu, „Metadata-driven approach to generalisation of transformations in ETL processes“, in *Intelligent Health Systems – From Technology to Data and Knowledge*. IOS Press, May 2025, ISBN: 9781643685960. DOI: 10.3233/shti250640 (cited on pp. 25, 28, 30, 38).

[44] F. Katsch, R. Hussein, G. Duftschmid, „Converting entity-attribute-value data sources to OMOP's CDM: Lessons learned“, in *Digital Health and Informatics Innovations for Sustainable Health Care Systems*. IOS Press, Aug. 2024, ISBN: 9781643685335. DOI: 10.3233/shti240419 (cited on p. 25).

[48] S. M. K. Sathappan, Y. S. Jeon, T. K. Dang, S. C. Lim, Y.-M. Shao, E. S. Tai, M. Feng, „Transformation of electronic health records and questionnaire data to OMOP CDM: A feasibility study using SG_T2DM dataset“, *Applied Clinical Informatics*, vol. 12, no. 04, pp. 757–767, Aug. 2021. DOI: 10.1055/s-0041-1732301 (cited on pp. 27, 28, 38).

[49] A. Jouned, H. Düsseldorf, F. Katsch, M. Jafarpour, G. Duftschmid, „Comparative study of ETL tools for transforming healthcare data to the OMOP common data model (CDM)“, in *Intelligent Health Systems – From Technology to Data and Knowledge*. IOS Press, May 2025, ISBN: 9781643685960. DOI: 10.3233/shti250590 (cited on pp. 29, 36, 38).

[50] J. C. Nwokeji, R. Matovu, „A systematic literature review on big data extraction, transformation and loading (ETL)“, in *Intelligent Computing*. Springer International Publishing, 2021, pp. 308–324, ISBN: 9783030801267. DOI: 10.1007/978-3-030-80126-7_24 (cited on pp. 30, 33, 38).

[51] A. Dhaouadi, K. Bousselmi, M. M. Gammoudi, S. Monnet, S. Hammoudi, „Data warehousing process modeling from classical approaches to new trends: Main features and comparisons“, *Data*, vol. 7, no. 8, p. 113, Aug. 2022. DOI: 10.3390/data7080113 (cited on pp. 32, 38).

[52] V. Theodorou, A. Abelló, M. Thiele, W. Lehner, „Frequent patterns in ETL workflows: An empirical approach“, *Data & Knowledge Engineering*, vol. 112, pp. 1–16, Nov. 2017. DOI: 10.1016/j.datak.2017.08.004 (cited on pp. 32, 33, 38).

[53] M. Poess, T. Rabl, H.-A. Jacobsen, B. Caufield, „TPC-DI: The first industry benchmark for data integration“, *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1367–1378, Aug. 2014. DOI: 10.14778/2733004.2733009 (cited on p. 32).

[54] V. Theodorou, A. Abelló, W. Lehner, M. Thiele, „Quality measures for ETL processes: From goals to implementation“, *Concurrency and Computation: Practice and Experience*, vol. 28, no. 15, pp. 3969–3993, Dec. 2015. DOI: 10.1002/cpe.3729 (cited on pp. 33, 38, 47).

[55] A. S. Pall, J. S. Khaira, „A comparative review of extraction, transformation and loading tools“, *Database Systems Journal*, vol. 4, no. 2, pp. 42–51, 2013 (cited on pp. 34, 35).

[56] K. Y. Cheng, S. Pazmino, B. Schreiweis, „ETL processes for integrating healthcare data – tools and architecture patterns“, in *pHealth 2022*. IOS Press, Nov. 2022, ISBN: 9781643683492. DOI: 10.3233/shti220974 (cited on pp. 34, 35).

[58] S. K. Singu, „ETL process automation: Tools and techniques“, *ESP Journal of Engineering & Technology Advancements*, vol. 2, no. 1, Mar. 2022. DOI: 10.56472/25832646/ESP-V2I1P110 (cited on p. 35).

[60] V. A. Kherdekar, P. S. Metkewar, „A technical comprehensive survey of ETL tools“, *International Journal of Applied Engineering Research*, vol. 11, no. 04, p. 2557, Feb. 2016. DOI: 10.37622/ijaer/11.4.2016.2557-2559 (cited on p. 35).

[61] T. A. Majchrzak, T. Jansen, H. Kuchen, „Efficiency evaluation of open source ETL tools“, in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC'11, ACM, Mar. 2011, pp. 287–294. DOI: 10.1145/1982185.1982251 (cited on p. 35).

[62] T. Grechenig, *Softwaretechnik, Mit Fallbeispielen aus realen Entwicklungsprojekten*, M. Bernhart, R. Breiteneder, K. Kappel, Eds. München: Pearson Studium, 2010, ISBN: 9783868940077 (cited on pp. 39, 40, 42).

[63] T. Wilde, T. Hess, „Methodenspektrum der wirtschaftsinformatik: Überblick und portfoliobildung“, Instituts für Wirtschaftsinformatik und Neue Medien der Ludwig-Maximilians-Universität München, München, Tech. Rep., 2006, Arbeitsbericht Nr. 2/2006 (cited on pp. 39, 40).

[64] H. Snyder, „Literature review as a research methodology: An overview and guidelines“, *Journal of Business Research*, vol. 104, pp. 333–339, Nov. 2019. DOI: 10.1016/j.jbusres.2019.07.039 (cited on p. 39).

[65] Y. Peng, F. Bathelt, R. Gebler, R. Gött, A. Heidenreich, E. Henke, D. Kadioglu, S. Lorenz, A. Vengadeswaran, M. Sedlmayr, „Use of metadata-driven approaches for data harmonization in the medical domain: Scoping review“, *JMIR Medical Informatics*, vol. 12, e52967, Feb. 2024. DOI: 10.2196/52967 (cited on pp. 46, 47).

[66] N. T. Sibert, J. Soff, S. L. Ferla, M. Quaranta, A. Kremer, C. Kowalski, „Transforming a large-scale prostate cancer outcomes dataset to the OMOP common data model—experiences from a scientific data holder's perspective“, *Cancers*, vol. 16, no. 11, p. 2069, May 2024. DOI: 10.3390/cancers16112069 (cited on pp. 46, 47).

[75] World Health Organization, *Report on the burden of endemic health care-associated infection worldwide*. World Health Organization, 2011, ISBN: 9789241501507. Accessed: Aug. 23, 2025. [Online]. Available: https://apps.who.int/iris/handle/10665/80135 (cited on p. 85).

[76] R. G. Hughes, „Patient safety and quality: An evidence-based handbook for nurses“, Apr. 2008, ppublish (cited on p. 85).

[77] C. Suetens, K. Latour, T. Kärki, E. Ricchizzi, P. Kinross, M. L. Moro, B. Jans, S. Hopkins, S. Hansen, O. Lyytikäinen, J. Reilly, A. Deptula, W. Zingg, D. Plachouras, D. L. M. and, „Prevalence of healthcare-associated infections, estimated incidence and composite antimicrobial resistance index in acute care hospitals and long-term care facilities: Results from two european point prevalence surveys, 2016 to 2017“, *Eurosurveillance*, vol. 23, no. 46, Nov. 2018. DOI: 10.2807/1560-7917.es.2018.23.46.1800516 (cited on p. 85).

[78] C. Rock, K. A. Thom, A. D. Harris, S. Li, D. Morgan, A. M. Milstone, B. Caffo, M. Joshi, S. Leekha, „A multicenter longitudinal study of hospital-onset bacteremia: Time for a new quality outcome measure?“, *Infection Control & Hospital Epidemiology*, vol. 37, no. 2, pp. 143–148, Oct. 2015. DOI: 10.1017/ice.2015.261 (cited on p. 85).

[79] N. E. Babady, „Hospital-associated infections“, in *Diagnostic Microbiology of the Immunocompromised Host*. ASM Press, Apr. 2016, pp. 735–758, ISBN: 9781555819033. DOI: 10.1128/9781555819040.ch28 (cited on p. 85).

[80] T. van der Kooi, A. Lepape, P. Astagneau, C. Suetens, M. A. Nicolaie, S. de Greeff, I. Lozoraitiene, J. Czepiel, M. Patyi, D. P. and, „Mortality review as a tool to assess the contribution of healthcare-associated infections to death: Results of a multicentre validity and reproducibility study, 11 european union countries, 2017 to 2018“, *Eurosurveillance*, vol. 26, no. 23, Jun. 2021. DOI: 10.2807/1560-7917.es.2021.26.23.2000052 (cited on p. 85).

[81] M. B. G. Koek, T. I. I. van der Kooi, F. C. A. Stigter, P. T. de Boer, B. de Gier, T. E. M. Hopmans, S. C. de Greeff, J. Entius, J. C. M. Diederen, E. H. Groenendijk, L. Nolles, K. P. Jalink-Olthof, B. J. H. den Broeder, H. G. M. Blaauwgeers, „Burden of surgical site infections in the netherlands: Cost analyses and disability-adjusted life years“, *Journal of Hospital Infection*, vol. 103, no. 3, pp. 293–302, Nov. 2019. DOI: 10.1016/j.jhin.2019.07.010 (cited on p. 85).

[82] H. R. A. Streefkerk, R. P. A. J. Verkooijen, W. M. Bramer, H. A. Verbrugh, „Electronically assisted surveillance systems of healthcare-associated infections: A systematic review“, *Eurosurveillance*, vol. 25, no. 2, Jan. 2020. DOI: 10.2807/1560-7917.es.2020.25.2.1900321 (cited on p. 85).

[83] J. D. M. Verberk, S. M. van Rooden, M. B. G. Koek, D. J. Hetem, A. E. Smilde, W. S. Bril, R. H. R. A. Streefkerk, T. E. M. Hopmans, M. J. M. Bonten, S. C. de Greeff, M. S. M. van Mourik, „Validation of an algorithm for semiautomated surveillance to detect deep surgical site infections after primary total hip or knee arthroplasty—a multicenter study“, *Infection Control & Hospital Epidemiology*, vol. 42, no. 1, pp. 69–74, Aug. 2020. DOI: 10.1017/ice.2020.377 (cited on p. 85).

[84] S. M. van Rooden, E. Tacconelli, M. Pujol, A. Gomila, J. A. J. W. Kluytmans, J. Romme, G. Moen, E. Couvé-Deacon, C. Bataille, J. R. Baño, J. Lanz, M. S. van Mourik, „A framework to develop semiautomated surveillance of surgical site infections: An international multicenter study“, *Infection Control & Hospital Epidemiology*, pp. 1–8, Dec. 2019. DOI: 10.1017/ice.2019.321 (cited on p. 86).

[85] M. Behnke, J. K. Valik, S. Gubbels, D. Teixeira, B. Kristensen, M. Abbas, S. M. van Rooden, P. Gastmeier, M. S. M. van Mourik, O. Aspevall, P. Astagneau, M. J. M. Bonten, E. Carrara, A. Gomila-Grange, S. C. de Greeff, W. Harrison, H. Humphreys, A. Johansson, M. B. G. Koek, A. Lepape, J.-C. Lucet, S. Mookerjee, P. Naucler, Z. R. Palacios-Baena, E. Presterl, M. Pujol, J. Reilly, C. Roberts, E. Tacconelli, T. Tängdén, „Information technology aspects of large-scale implementation of automated surveillance of healthcare-associated infections“, *Clinical Microbiology and Infection*, vol. 27, S29–S39, Jul. 2021, ISSN: 1198-743X. DOI: 10.1016/j.cmi.2021.02.027 (cited on p. 86).

[86] M. S. M. van Mourik, S. M. van Rooden, M. Abbas, O. Aspevall, P. Astagneau, M. J. M. Bonten, E. Carrara, A. Gomila-Grange, S. C. de Greeff, S. Gubbels, W. Harrison, H. Humphreys, A. Johansson, M. B. G. Koek, B. Kristensen, A. Lepape, J.-C. Lucet, S. Mookerjee, P. Naucler, Z. R. Palacios-Baena, E. Presterl, M. Pujol, J. Reilly, C. Roberts, E. Tacconelli, D. Teixeira, T. Tängdén, J. K. Valik, M. Behnke,

P. Gastmeier, „PRAISE: Providing a roadmap for automated infection surveillance in europe“, *Clinical Microbiology and Infection*, vol. 27, S3–S19, Jul. 2021. DOI: 10.1016/j.cmi.2021.02.028 (cited on p. 86).

[87] S. M. van Rooden, O. Aspevall, E. Carrara, S. Gubbels, A. Johansson, J.-C. Lucet, S. Mookerjee, Z. R. Palacios-Baena, E. Presterl, E. Tacconelli, M. Abbas, M. Behnke, P. Gastmeier, M. S. M. van Mourik, „Governance aspects of large-scale implementation of automated surveillance of healthcare-associated infections“, *Clinical Microbiology and Infection*, vol. 27, S20–S28, Jul. 2021. DOI: 10.1016/j.cmi.2021.02.026 (cited on p. 86).

[88] S. J. S. Aghdassi, S. D. van der Werff, G. Catho, M. Brekelmans, L. A. P. Diaz, N. Buetti, F. D. Rüther, D. Teixeira, D. Sjöholm, P. Nauclér, M. Behnke, M. S. M. van Mourik, „Hospital-onset bacteraemia and fungaemia as a novel automated surveillance indicator: Results from four european university hospitals, 2018 to 2022“, *Eurosurveillance*, vol. 30, no. 24, Jun. 2025. DOI: 10.2807/1560-7917.es.2025.30.24.2400613 (cited on pp. 86, 87).

[89] A. Torab-Miandoab, T. Samad-Soltani, A. Jodati, P. Rezaei-Hachesu, „Interoperability of heterogeneous health information systems: A systematic literature review“, *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, Jan. 2023. DOI: 10.1186/s12911-023-02115-5 (cited on p. 86).

[90] European Centre for Disease Prevention and Control, *Point prevalence survey of healthcare- associated infections and antimicrobial use in European acute care hospitals: protocol version 6.1, ECDC PPS 2022 2023.* LU: Publications Office, 2019. DOI: 10.2900/017250 (cited on p. 88).

[93] J. Van Eldere, M. Koual, M. M. Karsten, L. B. Koppert, E. Hedayati, O. D. Gentilini, H. Wildiers, G. Duftschmid, M. D. Lautrup, „EUHA breastcancer benchmarking project – a retrospective data analysis“, Mar. 16, 2023 (cited on pp. 91, 92).

Online References

[9] Observational Health Data Sciences and Informatics. „Software tools“, Accessed: Aug. 23, 2025. [Online]. Available: https://www.ohdsi.org/software-tools/ (cited on pp. 2, 14, 16).

[15] Medizinische Universität Wien. „RDA“, Accessed: Aug. 23, 2025. [Online]. Available: https://www.meduniwien.ac.at/web/mitarbeiterinnen/it-hilfe-support/it4science/plattformen/rda/ (cited on pp. 7, 10).

[20] „JSON schema“, Accessed: Aug. 23, 2025. [Online]. Available: https://json-schema.org/ (cited on pp. 8, 20, 21).

[24] World Health Organization. „International statistical classification of diseases and related health problems (ICD)“, Accessed: Aug. 23, 2025. [Online]. Available: https://www.who.int/standards/classifications/classification-of-diseases (cited on p. 11).

[25] Regenstrief Institute, Inc. „LOINC“, Accessed: Aug. 23, 2025. [Online]. Available: https://loinc.org/ (cited on p. 11).

[27] Observational Health Data Sciences and Informatics. „OMOP Common Data Model“, Accessed: Aug. 23, 2025. [Online]. Available: https://ohdsi.github.io/CommonDataModel/ (cited on pp. 12–14, 24, 51, 52, 60).

[45] Observational Health Data Sciences and Informatics. „Rabbit in a hat“, Accessed: Aug. 23, 2025. [Online]. Available: https://ohdsi.github.io/WhiteRabbit/RabbitInAHat.html (cited on pp. 25, 37, 38).

[46] Observational Health Data Sciences and Informatics. „Whiterabbit“, Accessed: Aug. 23, 2025. [Online]. Available: https://github.com/OHDSI/WhiteRabbit (cited on p. 25).

[47] R. Raszczynski. „Understanding the eav data model and when to use it“, Accessed: Aug. 23, 2025. [Online]. Available: https://inviqa.com/blog/understanding-eav-data-model-and-when-use-it (cited on pp. 26, 29).

[57] The Apache Software Foundation. „Apache nifi“, Accessed: Aug. 23, 2025. [Online]. Available: https://nifi.apache.org/ (cited on p. 35).

[59] Hitachi Vantara LLC. „Pentaho data integration“, Accessed: Aug. 23, 2025. [Online]. Available: https://pentaho.com/products/pentaho-data-integration/ (cited on pp. 35, 73).

[67] Observational Health Data Sciences and Informatics. „Observation periods for ehr data“, Accessed: Aug. 23, 2025. [Online]. Available: https://ohdsi.github.io/Themis/obs_periods_for_ehr.html (cited on p. 61).

[68] SQLite Consortium. „Sqlite“, Accessed: Aug. 23, 2025. [Online]. Available: https://sqlite.org/ (cited on p. 73).

[69] SQLite. „Appropriate uses for sqlite“, Accessed: Aug. 23, 2025. [Online]. Available: https://sqlite.org/whentouse.html (cited on p. 74).

[70] OpenAI. „Chatgpt“, Accessed: Aug. 23, 2025. [Online]. Available: https://chatgpt.com/ (cited on p. 75).

[71] OHDSI. „CommonDataModel“, Accessed: Aug. 23, 2025. [Online]. Available: https://github.com/OHDSI/CommonDataModel (cited on p. 78).

[72] Odysseus Data Services, Inc. „Athena vocabulary download“, Accessed: Aug. 23, 2025. [Online]. Available: https://athena.ohdsi.org/vocabulary/list (cited on p. 79).

[73] Former user (Deleted). „Copy rows to result“, Accessed: Aug. 23, 2025. [Online]. Available: https://pentaho-public.atlassian.net/wiki/spaces/EAI/pages/371558228/Copy+rows+to+result (cited on p. 80).

[74] Hitachi Vantara, LLC. „ETL metadata injection“, Accessed: Oct. 17, 2024. [Online]. Available: https://docs.hitachivantara.com/r/en-us/pentaho-data-integration-and-analytics/9.3.x/mk-95pdia003/pdi-transformation-steps/etl-metadata-injection (cited on p. 81).

[91] National Healthcare Safety Network, *Master organism common commensals list*, 2024. Accessed: Aug. 23, 2025. [Online]. Available: https://www.cdc.gov/nhsn/xls/master-organism-com-commensals-lists.xlsx (cited on p. 89).

[92] Eurostat. „Cancer statistics - specific cancers“, Accessed: Aug. 23, 2025. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Cancer_statistics_-_specific_cancers#Breast_cancer (cited on p. 91).

[94] European University Hospital Alliance. „EUHA gets a step closer to breast cancer benchmarking across europe“, Accessed: Aug. 23, 2025. [Online]. Available: https://www.euhalliance.eu/2022/10/14/euha-gets-a-step-closer-to-breast-cancer-benchmarking-across-europe/ (cited on p. 92).

APPENDIX A

Mock-Ups of Forms

Figures A.1 to A.6 show mock-ups of the clinical documentation forms used to generate synthetic test data for the ETL prototype. The forms were designed to simulate realistic clinical input and are grouped by the OMOP CDM table each form is intended to populate.

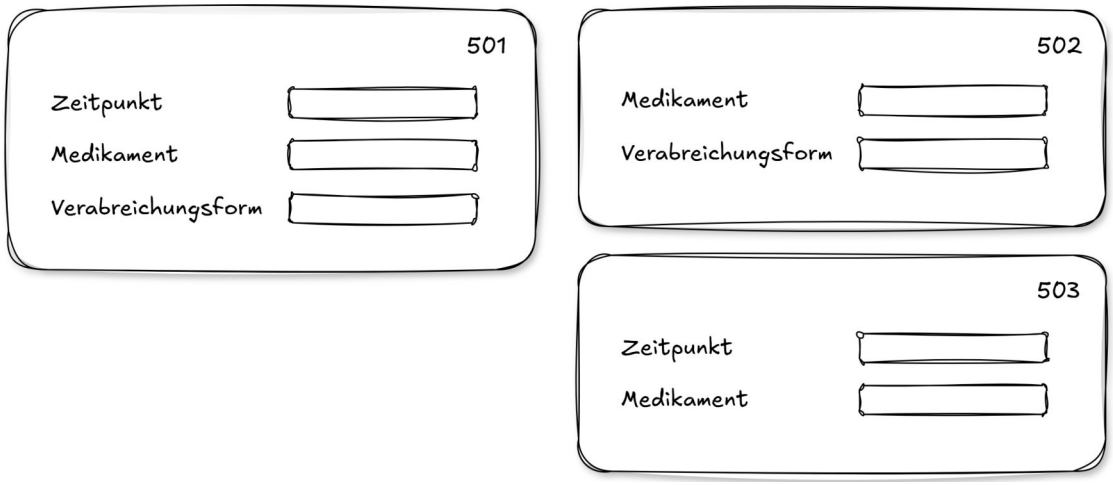


Figure A.1: Mock-up of forms for the prototype development targeting the drug_exposure table.

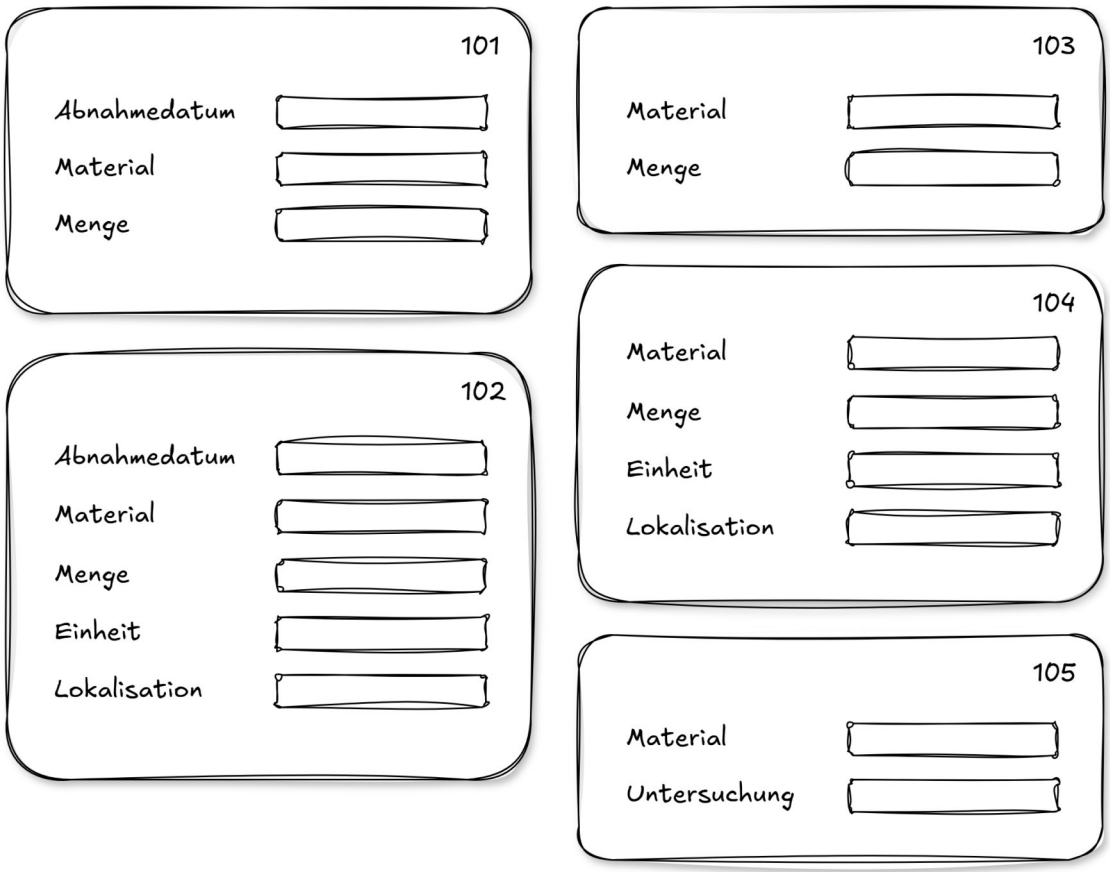


Figure A.2: Mock-up of forms for the prototype development targeting the specimen table.

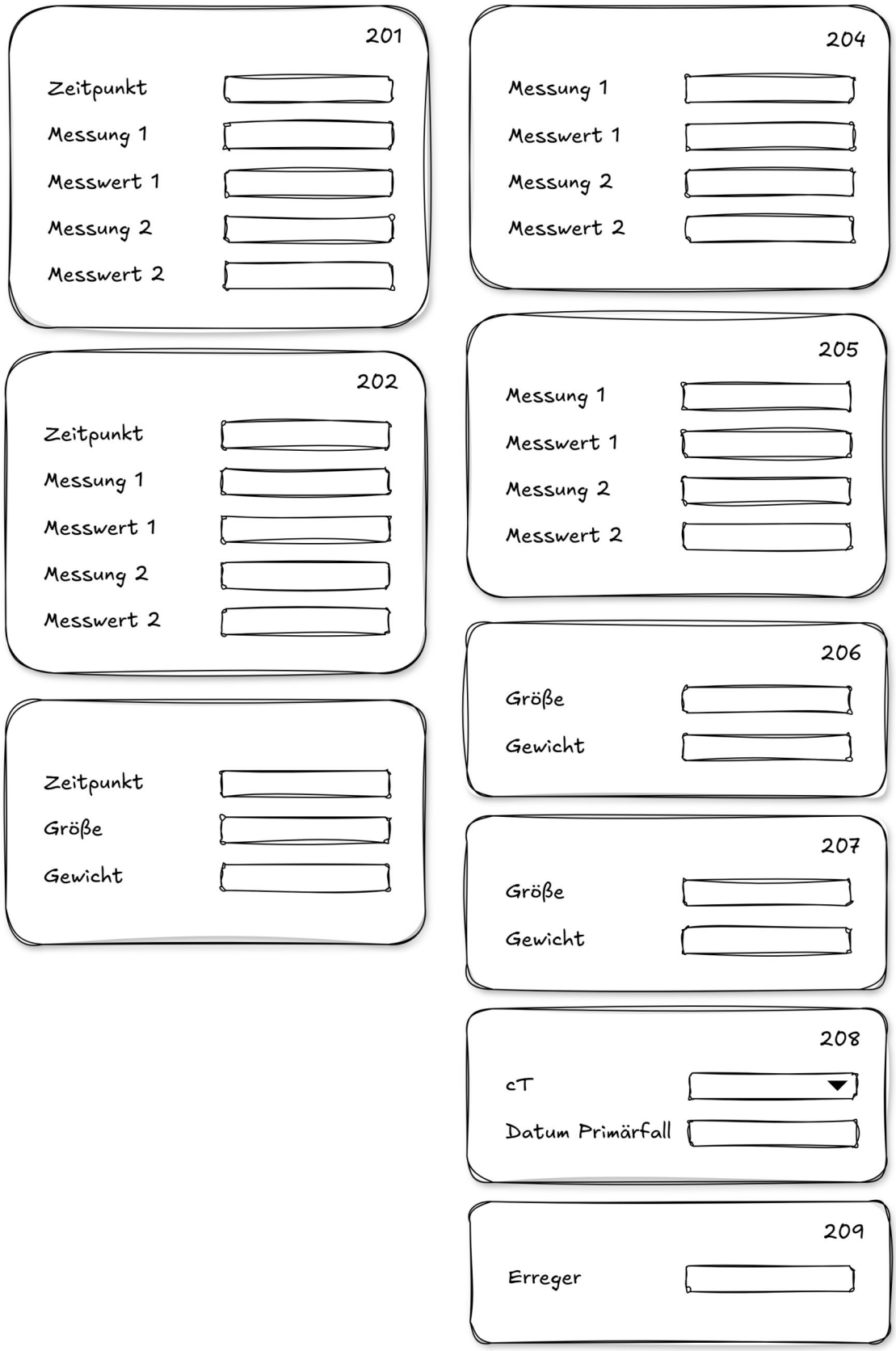


Figure A.3: Mock-up of forms for the prototype development targeting the measurement table.

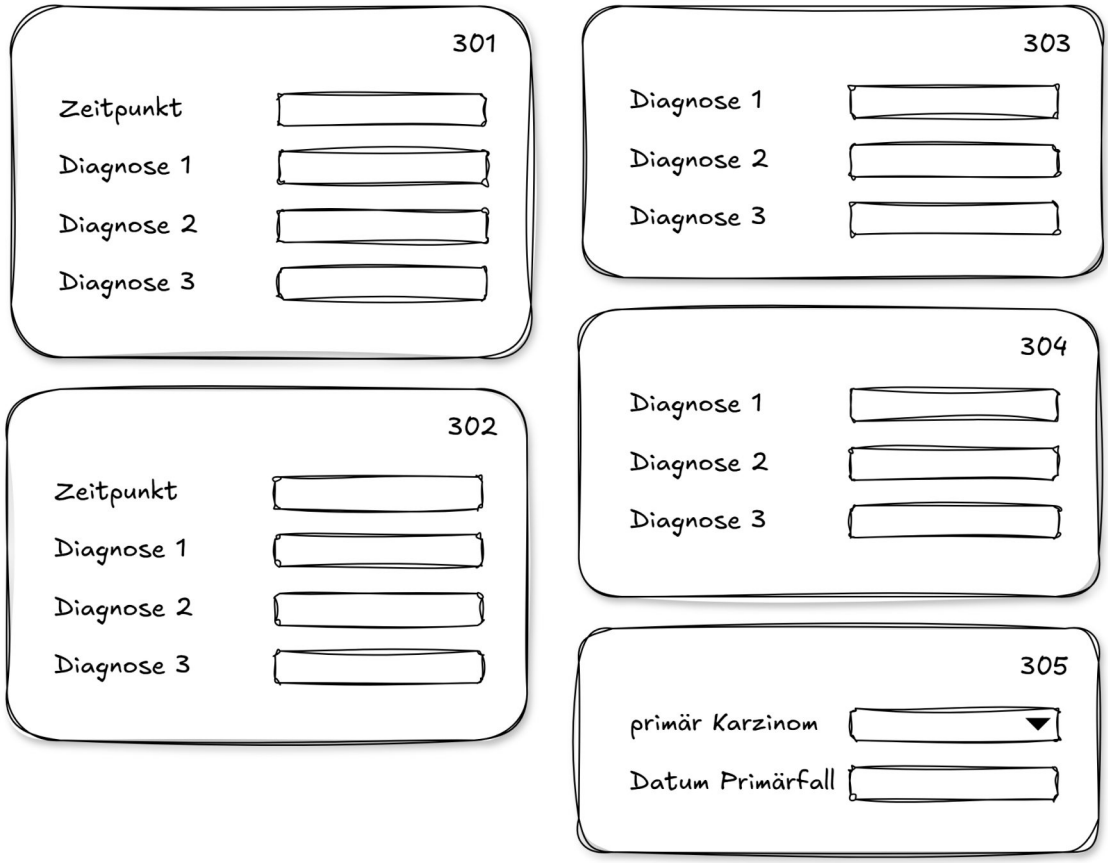


Figure A.4: Mock-up of forms for the prototype development targeting the condition_occurrence table.

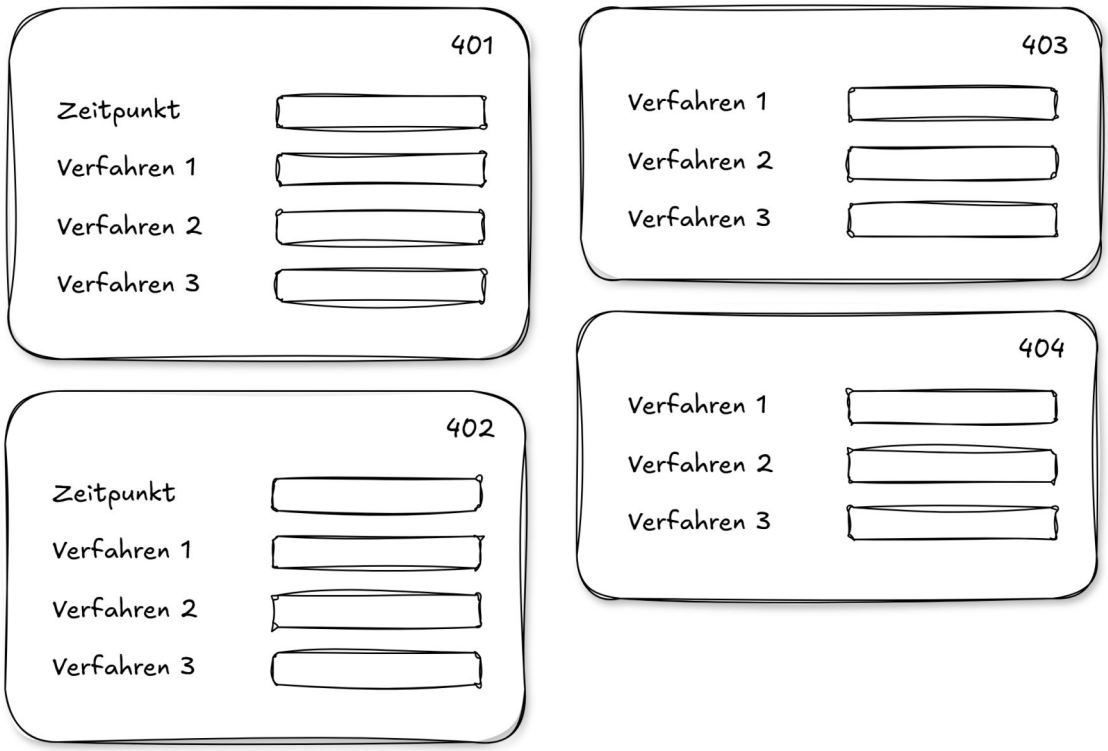


Figure A.5: Mock-up of forms for the prototype development targeting the procedure_occurrence table.

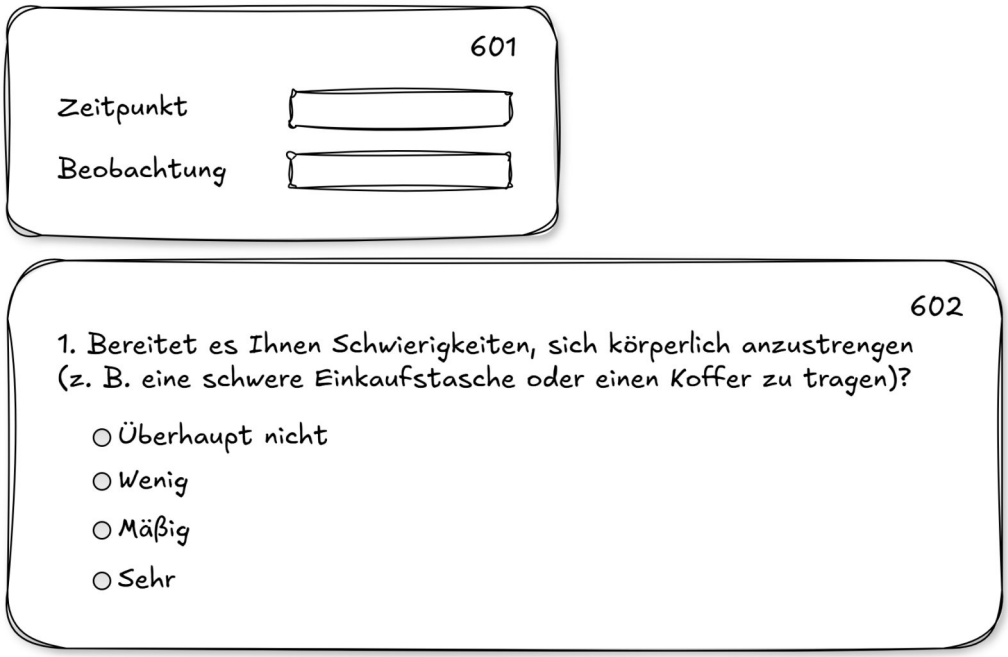


Figure A.6: Mock-up of forms for the prototype development targeting the observation table.