Knowledge-enriched Schema Mapping: A Preliminary Case Study of e-MedSolution System

This work has been supported by the European Institute of Innovation & Technology (EIT) Project (18346-A2002)

Author: Chuangtao Ma

Advisor: Dr. Bálint Molnár and Ádám Tarcsi



Faculty of Informatics Eötvös Loránd University (ELTE)

Budapest, Hungary 2020

Knowledge-enriched Schema Mapping: A Preliminary Case Study of e-MedSolution System

Abstract

Data integration is a process of accessing and integrating the various data from multi sources database. Generally, this process is composed of three phases, i.e., schema alignment and mapping, record linkage, and data fusion. Schema matching is the critical phase of migrating and integrating the heterogenous databases into a centralized database, in which the semantic correspondences between source schema and target schema are usually identified for creating schema mappings. Typically, there are three essential steps of mapping the source schema onto target schema: vocabulary mapping, data table mapping, data transformation. The terminology and attribute of source schema could be easily mapped to the target schema if there exits the equivalent terminology and attribute in target schema. Contrarily, the domain knowledge and manual mapping are required to determine whether this terminology and attribute should be merged.

Knowledge-based integration are regarded as one of the efficient integration methods due to the excellent semantic interoperability of knowledge base. Ontologies, a representative and formalized knowledge bases, which provides a rich semantic reference for the schema mapping and data integration due to its semantic interoperability and rigorous mathematical foundation. Accordingly, ontology-based data integration(OBDI) has played a critical role in heterogeneous data integration and schema mapping. However, the traditional methods for constructing ontology are manual, in which a lot of effort and experience from domain experts are required.

In recent years, several knowledge bases, i.e., ontologies, linked data, knowledge graph, etc, were constructed accompanied by the knowledge representation learning and natural language processing (NLP), which yield an opportunity for enriching the knowledge and eliminate ambiguous mapping during the schema mapping and the migration of legacy information systems. In particular, ontology learning (OL) was proposed to (semi-) automatically construct ontologies, in which entities and relationship are usually extracted based on the computation and inference. Accordingly, a large number ontologies could be (semi-) automatically constructed to provide the rich semantic reference for schema mapping.

In this work, we investigated the framework of knowledge-enriched schema mapping and ontology learning, and highlighted the feasibility of utilizing these frameworks in heterogenous schema mapping and data integration. In particular, a preliminary case study was conducted to illustrate how the knowledge-enriched schema mapping and ontology leaning could be utilized to migration and integration of heterogenous database. More precisely, we introduced the structure and data tables of e-MedSolution and OMOP CDM, and analyzed the heterogeneity of these two data models initially. Furthermore, we designed a schematic diagram for schema mapping between clinical module of e-MedSolution and OMOP CDM clinical data and health system data. Moreover, we investigated how the existing knowledge bases and ontology learning could provides a semantic reference for vocabulary mapping and data table mapping between e-MedSolution and OMOP CDM. Finally, we summarized the main contribution of this work and gave the directions of future work.

Contents

1	Intr	oduction	4				
2	Rela	Related Work					
	$2.1 \\ 2.2 \\ 2.3$	Schema Matching and Mapping	5 7 9				
3	Pro	Proposed Framework					
	$3.1 \\ 3.2$	Knowledge-enriched Mapping	9 11				
4	Cas	se Study 1					
	$ \begin{array}{c} 4.1 \\ 4.2 \\ 4.3 \\ 4.4 \end{array} $	Problem Statement	13 13 16 18 19 21				
5	Sun	nmary and Future Wrok	21				
	$5.1 \\ 5.2$	Summary	21 22				
A	App	pendix	26				
	A.1 A.2	Concise database schema version of e-MedSolution	26 26				

1 Introduction

In the past few decades, various information systems have been developed in different sub-organizations within an organization to provide business process management and decision-making. Currently, with the expansion of business and the revolution of information technology, most of these information systems are gradually becoming the legacy information systems. In particular, it is an challenge task to integrate these legacy information systems due to the no-standardized data-access protocol and database design paradigm, and diverse naming conventions. As we all know, the most of the data and knowledge of information systems reside in the relational database, thus, the major task of legacy information system integration is to integrate heterogenous database.

Typically, there are two alternative solution for heterogenous database integration: physical integration and logical integration [1]. In physical database integration, the several heterogenous database are migrated and mapped into a global data model, in which the various databases could be stored in a centralized database. In contrast to the physical database integration, the logical database integration just provides a common data access interface by query rewriting, while the data still resides in its original database. Although, it is a cost task to physically integrate the heterogenous database into a centralized data model, in the long term, physical integration could achieve the efficient data access and data management.

However, physical database integration is a challenging work, since the various inconsistencies and conflicts should be resolved during the integrating and migrating of the heterogeneous database into a global data model. In particular, due to the diversity of the database schema and the variety of naming conventions in heterogenous database, there are various inconsistencies, e.g., format inconsistency, structure inconsistency, syntax inconsistency, semantic inconsistency, etc, in which results the different kinds of conflicts and redundancies. Accordingly, the crucial task of physical database integration and is to identify the semantic correspondences between source and target database and handle aforementioned conflicts and redundancies.

In general, there are three critical steps in heterogeneous database integration, i.e., schema alignment and mapping, record linkage, and data fusion [2]. In the phase of schema alignment and mapping, the conflicts are usually resolved based on semantic correspondence of the entities and attributes between target schema and source schema. In the phase of record linkage, the various database instance are linked by measuring the similarity between target record and source record. In the phase of data fusion, the different kinds of conflicts and redundancies are eliminated by the merging the similar records.

Knowledge bases (KBs) is a reference repository of entities, types, and attributes of entities with open-ended scope, in which a rich of taxonomy of types and terminology, attributes of entities, and relationships between entities are included [3]. Knowledge-based integration are regarded as one of the efficient integration methods due to the excellent semantic interoperability of KBs. Typically, constructing knowledge bases are based on manual transformation and mapping, hence the quality of knowledge bases are heavily dependent on the experience from domain experts [4]. Recently, driven by the increasing requirement of dynamic business process, the knowledge bases are required to update and maintain periodically. Hence, it is cost and tedious task to construct and maintain knowledge bases by using of the traditional manual mapping and transformation.

KBs contains the class hierarchy and logical connections of knowledge, the former one is called taxonomy, and the latter one is referred to as ontologies. In addition to the class hierarchy and logical connections, KBs can also contains the logical constraints and rules, in which the consistency of KBs could be checked based on the inference. Ontologies, one of the representative and formalized knowledge bases, which provides a rich semantic reference for the schema mapping and data integration due to its semantic interoperability and rigorous mathematical foundation [5]. Similarity, the traditional methods for constructing ontology are manual, in which a lots of effort and the experience from domain experts are required. Due to the the biases and limitations of human knowledge, some accidental errors and inconsistencies will inevitably occur.

Ontology learning (OL) as one of the knowledge representation learning methods was proposed to (semi-)automatically construct ontologies from various datasources, in which entities and relationship are usually extracted based on the computation and inference. The techniques of ontology learning are classified into four categories: association rule mining (ARM), formal concept analysis(FCA), inductive logic programming (ILP), neural networks(NN) and machine learning [6]. In particular, ontology learning could free human's hands from the tedious mapping and transformation, minimize the negative influence of human knowledge biases. On the basis of knowledge representation learning and natural language process (NLP), several knowledge bases, i.e., ontologies, linked data, knowledge graph, etc, are constructed in recent years. These knowledge bases could provide the knowledge for identifying and enriching the semantic correspondence, by which the conflicts and redundancies could be resolved and eliminated.

The motivation of this work is to investigate how to utilize knowledge representation learning and knowledge base to enrich the semantic correspondence between the target schema and source schema during the schema mapping. More specifically, we investigated the framework of ontology learning, and presented a knowledge-enriched schema mapping method. Additionally, a case study was conducted to map the clnical data model of e-MedSolution system to the data model of OMOP CDM clnical data and the datatbale of health information systems by employing the knowledge enriched schema mapping.

2 Related Work

2.1 Schema Matching and Mapping

Schema mapping is a process of generating the assertions and mappings from the identified semantic correspondence, by which the source schema could be mapped onto the target schema to provide a common interface for accessing and querying of heterogenous data [7]. Schema matching is a process of establishing semantic associations between different schema, in which the semantic correspondences between source schema and target schema are usually identified [8]. In contrast to the schema matching, schema mappings is a kind of formal and intermediate language that are employed to describe the semantic correspondence between source schema and target schema, e.g., XML, R2RML, etc., which provides a semantic reference for the schema mapping.

Schema matching is a preliminary work in several fields, e.g., schema mapping, data integration, data exchange, etc. Accordingly, there are several approaches were proposed to identify the semantic correspondence between source and target schema. Rahm. et al [9] classified the approach of schema matching into the following categorizes based on the matching level: schema-level, instance-level, element-level, structure-level, and constraint-based approach. As shown in Fig. 1, they also class



Figure 1: Taxonomy of the earlier schema matching approaches [9]

sified the matching technique of each matching approach, accordingly, a conclusion could be drawn that the mainstream techniques of schema matching are similaritybased matching and graph-based matching in the early research. However, there is a room for improving the accuracy of schema matching based on similarity and graph matching, since lacking the evidence and knowledge during the matching. To address this issues, a corpus enriched schema matching method was proposed for augmenting the schema matching [10]. In this approach, the similar concepts and their relationships in source and target schema were identified to infer the constrains for pruning the candidate mappings. In addition to improve the method of schema mapping, how to verify the the quality of these generated mappings is a crucial task. To address this issue, the three-layers system was designed to verify the quality of schema mapping, in which the structure analysis was employed to check and select the optimal mapping transformation [11].

In general, the correspondences between two schema were identified based domain experts manually, in which a lot of human effort and experience were required. To achieve the automatic schema mapping, an automatch system was designed to support the automatically schema matching based on machine learning [12]. More specifically, an probability knowledge was obtained from the domain experts based on Bayesian learning, which provides a attribute dictionary to find an optimal mappings. Similarity, a probabilistic and logic-based formal framework sPLMap was proposed to learning schema mapping rules, in which a probabilistic interpretation of predicting weights was given to help selecting the matching with high mappings probability from candidate mappings [13]. Considering the strengths of similaritybased mapping and machine learning based mapping, a hybrid schema mapping model was proposed by combing the lexical and semantic similarity with machine learning [14].

Despite of aforementioned schema matching methods could learn the mappings from example mappings, these expression of mapping was identified and encoded at the lexical level. With the increasing number of matching at meta-level and instance level, how to identify the correspondence between various vocabulary and terminology at the semantic level is a insight direction [15]. In particular, the ontologies and linked data yields an new opportunity of entity resolution, in which the entity and its terminologies could be semantically mapped onto the global database model [16]. Aim to free human's hand from tedious and time-consuming task of mitigating data from multiple legacy systems into a global one, a new semi-automatic schema mapping approach was proposed [17]. In this approach, the domain ontologies and sample instance were reused to identify and determine the semantic correspondences between schema. Similarity, a schema matching method was proposed to match schema based on analysis of attributes values, in which the external knowledge, namely, background ontologies, was utilized to provide the semantic reference based on ontology alignment [18]. In view of the hierarchy of ontologies is conducive to identify the semantic correspondences with various levels, a schema matching methods based on ontology and rule clustering was proposed [19].

2.2 Ontology Learning

Ontology learning (OL) is a kind of ontology construction approach based on the machine learning technique [20], which was proposed to (semi-)automatically extract the knowledge from the text document or database for constructing ontology efficiently [21]. The majority techniques of ontology learning were borrowed from the NLP and data mining. The typical techniques of the terms and entities extraction are originated from NLP, e.g., tagging, syntactic segmentation, parsing, and so forth. The alternative approaches for implementing the NLP including machine learning and statistical inference. Moreover, the representative techniques of the relationship extraction were proposed based on the data mining algorithm, e.g., clustering algorithms, association rule mining, occurrence analysis.

In recent years, there is a great technological advancement in the fields of ontology construction, ontology mapping and semantic integration accompanied by the development of machine learning and computational intelligence [22]. To improve the knowledge representation of ontology, a domain ontology learning method based on LDA (Latent Dirichlet Allocation) model was proposed [23]. Similarity, a partial multi-dividing ontology algorithm was proposed to improve the efficiency of ontology learning by optimizing the partial multi-dividing ontology learning model [24].

In addition to ontology learning from text, there are several works focus on the ontology learning from RDB. There are two critical phases of ontology learning from relational databases. In the first phase, the RDB schema was usually transformed into RDFS (RDF Schema) based on the description logic (DL) and rule mapping. In the second phase, the semantic relationships were extracted, and the ontology instance was generated from RDB data by using semantic measurement and machine learning. The specified process and mainstream techniques of constructing ontology

from RDB are depicted in Fig. 2



Figure 2: Techniques of Ontology Learning from RDB

The mainstream techniques of the OL from RDB could be classified into four categories: reverse engineering, schema mapping, data mining, and machine learning. The corresponding work could be illustrated as follows. Considering the richer semantic of the conceptual model (E-R model), reverse engineering was to analyze and transform the relational model to the conceptual model for building ontology from RDB [25]. This method could recover the lost semantic information and database table during the transformation. There are two alternative solutions for constructing ontology from RDB schema: transform RDB to RDF and mapping RDB to OWl. To implement the transformation from RDB to ontology, a graph-based conceptual was introduced as a intermediate model [26]. This method consists of three steps: extract information (Meta-data) from RDB, build graph middle conceptual model and create the final ontology. Due to the RDB model does not store the semantic relationship among entities directly, there are some limitations of the automatic ontology learning from RDB, i.e., identify the incorrect semantic relationships between entities, ignore the implicit relations. To tackle the above issues, a novel approach of ontology learning from RDB based on semantic enrichment was proposed [27]. in which the meta-model was introduced to augment the semantic of RDB model. The case study shows that this approach could deduce the relationship in various domains.

Given that not only the schema information is implied in RDB SQL, but also the data information is represented in RDB SQL. Hence, a new paradigm of ontology learning from SQL scripts was proposed in recent years. For instance, a method for translating SQL algebra into SPARQL queries based on mapping rules was proposed [28]. In general, ontology learning from RDB SQL consists of three phases: pre-process, semantic enrichment, and transformation mapping. Before the transform and mapping, it is necessary to pre-process the RDB SQL. The majority of techniques of the pre-processing is parsing and lemmatization. To tackle the existing parsing methods that ignore the structure of database schema, there are two parsing methods of Text-to-SQL was proposed based on Graph Neural Network [29] and IRNet [30] respectively, which provide an essential theoretical foundation to construct ontology based on the approach of ontology learning from RDB SQL automatically.

2.3 Brief Summary

To summarize, the prevailing methods of schema matching are similarity based matching[14], graph-based matching [9], corpus enriched matching [10], machine learning [12, 13], hybrid schema mapping [14], and ontology-based mapping [18, 19]. Considering the hierarchy of ontologies is conducive to identify the semantic correspondences, especially, ontology mapping and alignment between global ontology and local ontology could provide an excellent semantic reference for identifying the semantic correspondences between source schema and target schema during schema mapping.

Due to the ontology is domain specific, the corresponding ontologies should be constructed for each domain. In particular, several ontologies should be correctly and efficiently constructed for large-scale schema mapping, [8]. Therefore, how to efficiently construct ontologies is a bottleneck of ontology-based schema mapping. Although, there are several methods on the topic of ontology learning, e.g., reverse engineering [25], graph-based mapping [26], semantic enrichment [27], and machine learning [31, 32]. However, the majority knowledge source of existing ontology learning method is text, and Web resource, accordingly, the most of aforementioned methods has employed to construct domain knowledge for machine translation [33, 34], intelligent question answering system (QAs) [35] and recommendation systems [36].

In contrast to the above fields, using ontology learning in schema mapping and data integration still is a new topic. Moreover, the most of existing works on the topic of ontology-based mapping focus on the technique of ontology mapping and matching, there is a minority number of the existing works [37, 38] on the topic of data integration based on ontology learning. Therefore, it is a meaningful work to investigate the automatic method to construct domain ontologies from RDB, and study how to employ these ontologies and other knowledge bases to enrich the schema mapping for legacy database integration.

3 Proposed Framework

3.1 Knowledge-enriched Mapping

The goal of schema mapping is to establish semantic correspondences between source and target schema. Typically, this kind of semantic correspondences could be formally represented by logic formula [7]. In general, the traditional schema mapping could be formally defined as a \mathcal{M} that is a specification between instance of source and target schema. Given a source schema **S** and its instance I_S , and target schema **T** and its instance I_T , we can say that the instance of source I_S and target schema I_T satisfies the mapping \mathcal{M} , which is formally represented as: $(I_T, I_S) \models \mathcal{M}$.

In general, the schema mapping \mathcal{M} is specified by the matchings and mappings, which are usually represented as a set of attribute-value pairs (AVP). The traditional methods for identifying and constructing mappings is mainly based on the lexical similarity and syntactic inference. Due to the similarity of attributes was measured and correspondence was inferred at the syntactic level, the constructed matchings and mappings are incapable of representing the full semantic correspondence between schema and its instance. For example, the outputs of traditional schema mapping model is a pair of mappings, where only the equivalent relationship is identified and taxonomic relationships is neglected.

Knowledge bases is a encyclopedic knowledge repository, where the open-ended knowledge, i.e., entities, taxonomic classes, attributes of entities, and relationship between entities are contained. Along with the technical advancement of knowledge representation and NLP, a increasing knowledge bases, e.g., WordNet, DBpedia, Linked Data, Onotlogies, and Knowledge Graph (KG), etc., are constructed from the various Web and text data, which provides a reference repository of entities, types, and taxonomic and logical relationship of terminologies. In view of KBs contains the class hierarchy and logical connections of knowledge, a framework of knowledge enriched schema mapping is proposed in Fig. 4, to tackle the aforementioned issue.



Figure 3: Framework of Knowledge Enriched Schema Mapping

We can see from Fig. 3, the different elements, i.e., entity, properties, datatype, relationships, integrity and referential constraints of various source schema could be obtained by using schema analysis, then the different elements of database schema could be mapped respectively based on the semantic reference of the KBs. Accordingly, the elements of the global schema is obtained, and the global schema are generated by using the reverse engineering.

Ontologies, one of the representative and formalized knowledge bases, which

provides a rich semantic reference for the schema mapping and data integration. In particular, not only does the ontology represents the various relationship, e.g., equivalent, inverse, subclass, etc, but also contains the constraints, e.g., type-constraints, disjoint constraints, inclusions and functional dependencies. Considering the excellent semantic interoperability of ontology and formalized representation, the ontology is selected as an example to illustrate how the KBs could provide the semantic reference for schema mapping. Thereby, a enriched ontology-based schema mapping was designed in Fig. 4, through which the heterogenous database with diverse property names could be mapped and integrated.



Figure 4: Enriched ontology-based schema mapping

As shown in Fig.4, a enriched domain ontologies was constructed in the middle part of the figure, which provides a semantic reference of the diversity of properties between source schema and target schema during the schema mapping. As we can see, this tailored ontology not only conveys the equivalent relationship, e.g., *is synonym with, is abbreviation of,* but also comprises the taxonomic relationships, e.g., *is subclass of, is property of,* etc. Accordingly, the properties with diverse name from different schema were connected by the dotted arrow, which represents the semantic correspondences between source and target schema. To conclude, the ontology-based schema mapping could map and integrate the heterogeneous database from legacy information systems at the semantic level.

3.2 Ontology Learning from RDB

As mentioned earlier, the crucial element of ontology-based mapping is ontologies. Due to the ontology is domain specific, the corresponding ontologies should be constructed for each domain. Especially, in some case, a tailored ontologies should be constructed from the heterogenous database schema for providing the precise semantic interoperability. Therefore, how to efficiently construct ontologies from relational database is a bottleneck of ontology-based schema mapping.

In general, there are two critical phases of ontology learning from relational databases: (1) Construct ontology from RDB schema; (2) Generate ontology instance from RDB data. For each critical phase, there are several corresponding sub-phases, e.g., pre-processing, transformation, mapping, enrichment, etc. Considering RDB SQL is a kind of text document, in which all entities, attributes, and their semantic relationships can be inferred. Moreover, SQL scripts can be accessed

easily via the DBMS or database driver, and there is no requirement for the special interface. Therefore, a framework of ontology learning from multiple RDB SQL is designed in Fig. 4, to efficiently construct ontology from RDB.



Figure 5: Framework of ontology learning from RDB

As shown in Fig. 5, the framework of ontology learning from RDB consists of multiple modules, e.g., data access, pre-processing, term extraction, relationship extraction, quality evaluation. Accordingly, the detailed process of learning from SQL scripts could be illustrated as followings:

- Data access from heterogenous database. To integrate and map the heterogeneous database to a global database, it is necessary to access these database. Considering the features of legacy information systems, we intend to directly access data through DBMS, the SQL scripts are exported from heterogenous DBMS and stored in text documents.
- Knowledge extraction from the relational database. Initially, the NLP techniques, i.e., parsing, named entity recognition (NER), and so forth, are utilized to pre-process the RDB SQL scripts and extract the terms. After that, the ontology learning model is employed to extract the relationships between these terms, which is represented as a RDF triples.
- Generate OWL object from RDF schema. On the basis of knowledge extraction, the OWL (Web Ontology Language) objects are generated for representing the entities, terminologies and their relationships between various knowledge rigidly and accurately. In addition, these ontologies was stored in graph database after evaluating its quality.

Given that both the schema information and instance information are implied in RDB SQL, thus, the paradigm of ontology learning from RDB not only construct ontology individuals from RDB schema, but also generate ontology concepts and attributes from RDB data. By employing the above framework of ontology learning from RDB, the tailored ontologies could be efficiently constructed from various database, which could provides a excellent semantic reference for schema matching and mapping.

4 Case Study

In this section, a case study is conducted to illustrate the feasibility of knowledgeenriched schema mapping by mapping the schema of e-MedSolution into the OMOP CDM (Common Data Model).

4.1 Problem Statement

e-MedSolution is a centralized health information system (HIS) that has been deployed and applied in various healthcare institutions, hospital, and medical university in Hungary. This Web-based system has been developed almost twenty years since the released the first version of e-MedSolution, which is conducive to implement the virtual hospital model. By using this systems, not only does the medical staff could access the healthcare data, but also the patient could view their healthcare information. Considering the primary challenge of HIS, namely, integration and globalization, the reversed communication interface of e-MedSolution were considered at the beginning, which provided an opportunity for integrating with sub-applications. Additionally, e-MedSolution system could be easily integrated with the MedSAPSol module, which provides a possibility for user to customized development according to their requirement.

In recent years, there are several modernized HIS have being developed, in which the advanced technology of architecture and the standardized paradigm of database design was utilized. In contrast to these modernized HIS, e-MedSolution system is a kind of legacy HIS, since the non-standardized designing paradigm of the database, and the diversity naming conventions of specified entities and properties. In particular, all of the data of e-MedSolution system resides in a centralized database that was designed two decades ago. Accordingly, the database schema is poor standardization, which has been the obstacle of integrating with other modernized HIS.

Thereby, it is necessary to map the existing database schema of e-MedSolution to a standardized database schema, which could yield a new opportunity to integrate with other modernized HIS. OMOP CDM that is an international, de facto standard for observational medical data, which provides a uniform data representation and standardized analysis of healthcare and clinical data [39].

4.2 Introduction of OMOP CDM

Aim to provide a comprehensive view of clinical and healthcare data for patient and medical staff, a common data standard (OMOP Common Data Model (CDM)) was proposed and employed to access and analysis the heterogenous data from multiple sources. There are several design principles, e.g., data protect, domains, standardized vocabularies, reuses, scalability, backwards compatibility, etc., were considered in OMOP CDM [40]. In particular, the OMOP CDM includes both the standardized vocabularies of terms and the entity domain tables, which provides a potential opportunity to integrate with other healthcare data model.

The Fig.6 depicts the structure of OMOP CDM Version 6.0, in this figure, the data models has been classified into the following categories according to the domain: standardized clinical data, standardized health system data, standardized derived elements, standardized health economics, standardized metadata, standardized vocabularies, and results schema.



Figure 6: The structure of the OMOP CDM Version 6.0 [39]

Additionally, the OMOP CDM defines several entities in different data model domains, for instance, the various entities, e.g., Person, Visit Occurrence, Drug Exposure, Condition Occurrence, Measurement, Observation, etc, from different sub-domains of healthcare systems were defined in the standardized clinical data. The tables from standardized clinical data were introduced as followings.

- Person. This table serves as the central identity management for all Persons in the database. It contains records that uniquely identify each person or patient, and some demographic information.
- Observation_period. This table contains records which define spans of time during which two conditions are expected to hold: (i) Clinical Events that happened to the Person are recorded in the Event tables, and (ii) absense of records indicate such Events did not occur during this span of time.
- Vist_occurrence. This table table contains Events where Persons engage with the healthcare system for a duration of time. They are often also called "Encounters". Visits are defined by a configuration of circumstances under which they occur, such as (i) whether the patient comes to a healthcare institution, the other way around, or the interaction is remote, (ii) whether and what kind of trained medical staff is delivering the service during the Visit, and (iii) whether the Visit is transient or for a longer period involving a stay in bed.
- Vist_detail. This table is an optional table used to represents details of each record in the parent Vist_occurrence table.

- Condition_occurrence. This table contains records of Events of a Person suggesting the presence of a disease or medical condition stated as a diagnosis, a sign, or a symptom, which is either observed by a Provider or reported by the patient.
- Drug_exposure. This table captures records about the exposure to a Drug ingested or otherwise introduced into the body. Drugs include prescription and over-the-counter medicines, vaccines, and large-molecule biologic therapies.
- **Procedure_occurrence**. This table contains records of activities or processes ordered by, or carried out by, a healthcare provider on the patient with a diagnostic or therapeutic purpose.
- Device_exposure. This table captures information about a person's exposure to a foreign physical object or instrument which is used for diagnostic or therapeutic purposes through a mechanism beyond chemical action. Devices include implantable objects (e.g. pacemakers, stents, artificial joints), medical equipment and supplies (e.g. bandages, crutches, syringes), other instruments used in medical procedures (e.g. sutures, defibrillators) and material used in clinical care (e.g. adhesives, body material, dental material, surgical material).
- Measurement. This table contains records of Measurements, i.e. structured values (numerical or categorical) obtained through systematic and standardized examination or testing of a Person or Person's sample. The Measurement table contains both orders and results of such Measurements as laboratory tests, vital signs, quantitative findings from pathology reports, etc.
- Observation. This table captures clinical facts about a Person obtained in the context of examination, questioning or a procedure. Any data that cannot be represented by any other domains, such as social and lifestyle facts, medical history, family history, etc. are recorded in this table.
- Note. This table captures the unstructured information that was recorded by a provider about a patient in free text (in ASCII, or preferably in UTF8 format) notes on a given date.
- Note_NLP. This table encodes all output of NLP on clinical notes. Each row represents a single extracted term from a note.
- Specimen. This table contains the records identifying biological samples from a person.
- Fact_relationship. This table contains records about the relationships between facts stored as records in any table of the CDM. This relationships can be defined between facts from the same domain, or different domains.
- Survey_conduct. This table is used to store an instance of a completed survey or questionnaire.

In the standardized health system data, there are several data tables were defined as well, which provides the location, healthcare, and the responsible medical worker of each diagnosis and measurement.

- Location. This table represents a generic way to capture physical location or address information of Persons and Care Sites.
- Location_history. This table stores relationships between Persons or Care Sites and geographic locations over time.
- Care_site. This table contains a list of uniquely identified institutional (physical or organizational) units where healthcare delivery is practiced (offices, wards, hospitals, clinics, etc.).

• **Provider**. This table contains a list of uniquely identified healthcare providers. These are individuals providing hands-on healthcare to patients, such as physicians, nurses, midwives, physical therapists etc.

In addition to the aforementioned tables, there are several tables from standardized derived elements, standardized health economics, standardized metadata, and standardized vocabularies were defined. Essentially, the standardized vocabularies, e.g., concept, vocabulary, domain, concept_relationship, etc. were utilized in all of CDM fact tables for specifying the concepts and their relationships from various sub-domains. For example, **concept** represents the clinical information across all of the sub-domains by specifying the codes and associated descriptions. Based on the standardized vocabularies, the various data from multiple data-sources could be unified accessed and retrieved without perverse the diverse concept codes and their relationships of the original data table.

4.3 Database Schema of e-MedSolution

As we mentioned previously, despite of the e-MedSolution is a centralized HIS, the healthcare data of each institutions were still stored and maintained in the local database, which creates an obstacle for integration. In particular, it will poses a challenges when integrating with external HIS due to the no-standardized naming conventions of the database schema. Before mapping the source schema into the target schema, the schema of source and target schema should be analyzed. In previous subsection, we introduced the target schema (OMOP CDM), in this subsection, we will introduce the target schema (e-MedSolution).

The e-MedSolution system covers the several modules of healthcare system, for instance, medical organizations and sub-organizations (e.g. hun_institution, hun_hosp_par, hun_department, hun_nursest, etc.), economics and accounting (e.g. hun_bl_contract, hun_bl_invoice, etc.), and clinical data (e.g. hun_case, hun_diag, hun_patient, hun_medication, hun_drug, etc.).

We address the mapping between the data model of clinical module for e-MedSolution into the OMOP CDM clinical data model in this case study. Thereby, an concise data structure of the clinical module for e-MedSolution system was presented in the Fig. 7.

In this concise data structure ¹, the basic data tables of the clinical module for e-MedSolution system were defined. To construct the schematic diagram for mapping, the basic information of each data tables of the clinical module for e-MedSolution system were described as followings.

- hun_user. This table contains various users (e.g., doctor, nurse, patient, internal physicians, etc.).
- hun_doctor. This table contain the unique number, name, internal ID, and the qualification of doctor.
- hun_workhour. This table contains the available appointment time for each doctor's clinic.
- hun_patient. This table contains the basic data (e.g. name, birth date, birthplace, title, and TAJ number, etc) of patient.
- hun_case. This table contains the basic data of different kinds of case (e.g. outpatient case, inpatient case, and similar cured case) and their status (e.g.

¹The detailed fields of each data tables could be found in the Appendix A.1.



Figure 7: Basic data structure of the clinical module of e-MedSolution

canceled, booked, record, dismissed, and closed), case description, and the corresponding doctor and department, etc.

- hun_case_diag. This table contains the basic data (e.g. diagnoses description, date and time) of the specific diagnoses and indicate the its attending doctor (e.g. external doctor, internal doctor).
- hun_case_i. This table records the basic data of (e.g. patient number, date of admission and discharge, room and bed, interventions, results, payment and reimbursement category, etc.) the inpatient case.
- hun_case_o. This table records the basic data of (e.g. patient number, admission date, attend doctor, reason, results and indicators, payment and reimbursement category, etc.) the outpatient case.
- hun_order. This table contains the basic data (e.g. order status, testing date and time, testing or service items, etc.) of testing order for cases.
- hun_perform_acc. This table contains the performance and results (sending date and time, amount of interventions, results of lab test, and the accident information) of the onsite intervention of the case.
- hun_outp_sched_spec. This table contains the appointment information (e.g. department, date, available days, etc.) for outpatient appointment.

- hun_medication. This tables contains the medication and prescription information (e.g. patient number, drug, dosage, date and time, indication, quantity, alternative drug, contraindications, etc).
- hun_drug. This table contains the basic data (e.g. drug ID, ISOCode, description, brand-name, factor, category, warning, description, deleted information, etc.) of drug.
- hun_drug_package. This table contains the package information (e.g. package type, register, expiration date, etc.) of drug.
- hun_drug_package.This table records the price and tax information of drug.
- hun_drug_ingredient. This tables contains the active ingredients (e.g. classification, strength, unit, factor, valid period, etc.) of drug.
- hun_drug_side_effect. This table records the side effects (e.g. name, description of side effect, valid period, etc.) of drug.
- hun_drug_drug_text. This table records the additional information of drug in text.
- hun_drug_generic. This table records the generic information of drug.
- hun_drug_atc. This table records the ATC information (e.g. code, status, level, valid period. etc.) of each drug.

Despite of this concise schema does not contain and record all of the data of the clinical module for e-MedSolution system, this concise schema is the core and representative module of the e-MedSolution system.

4.4 Mapping e-MedSolution Schema to OMOP CDM

In this subsection, we attempt to map this clinical data model of e-MedSolution system to the OMOP CDM clinical modules by employing the knowledge-enriched schema mapping. Considering the correspondence between clinical module of e-MedSolution and OMOP CDM, a schematic diagram for schema mapping was designed in Fig. 8.

In the proposed schematic diagram of mapping between e-MedSolution and OMOP CDM, the clinical data model will be mapped into the OMOP clinical data and health system data. The preliminary correspondences between e-MedSolution and OMOP CDM were identified, which are represented by the dashed lines.

According to the identified correspondences, there are different kinds of mappings, e.g., one-to-one, one-to-many, and many-to-many. In the one-to-one mapping, one table from the source schema could be exactly mapped into the corresponding table in the target schema. In contrast to the one-to-one mapping, the one-to-many and many-to many mapping require the merging the duplicates fields and splitting the compound fields.

As we mentioned earlier, the semantic correspondences between various vocabulary, terms, and fields is crucial reference for the mapping, merging and splitting. Regardless of which kinds of mapping, the identification of the semantic correspondences based on existing knowledge is the foundation of schema mapping. In general, there are three steps of mapping the source schema onto target schema: vocabulary mapping, data table mapping, data transformation (ETL). Considering the knowledge bases could provide the semantic reference for the mapping and the ELT process could be easily executed by the scripts based on the vocabulary and data table mapping, thereby, we only consider the vocabulary mapping and data table



Figure 8: Schematic diagram of mapping between e-MedSolution and OMOP CDM

mapping in this work.

4.4.1 Vocabulary Mapping

Vocabulary mapping is a mapping process that map the relevant classifications and terminologies of source schema to the OMOP CDM standardized vocabularies. In the vocabulary mapping, the prepared local codes should be added to the OMOP CDM tables Vocabulary, thereby, the Concept and Concept_Relationship should mapped into the OMOP Standard Concepts (as shown in Fig. 9).

		1	
CONCEPT_ID	313217	←	Primary key
CONCEPT_NAME	Atrial fibrillation	←	English description
DOMAIN_ID	Condition	←	Domain
VOCABULARY_ID	SNOMED	←	Vocabulary
CONCEPT_CLASS_ID	Clinical Finding	←	Class in vocabulary
STANDARD_CONCEPT	S	←	Standard, Source
CONCEPT_CODE	49436004	*	of Classification
VALID_START_DATE	01-Jan-1970	K	Code in vocabulary
VALID_END_DATE	31-Dec-2099	$\langle \rangle$	Valid during time interval
INVALID_REASON		4	Interval

Figure 9: Standard representation of vocabulary concepts in the OMOP CDM [39]

In standardized vocabulary, each clinical events in the OMOP CDM are expressed as concepts, which represent the semantic notion of each event. Each concept is assigned a concept ID and Domain_ID to indicate the primary key and the

domain (e.g. Condition, Device, Measurement, Observation, Place of service, and Procedure) of these concept belong. Additionally, the other fields Vocabulary_ID, Concept_Class, Standard_Concpet, and valid periods are defined to specify the classifications, corresponding standard concept and their validity.

The traditional method for mapping the local concept into the standardized vocabulary is mainly based on the measurement of lexical similarity (e.g. Levenshtein distance, Euclidean distance, etc.) between vocabulary, which could achieve the mapping at the lexical level. However, this method will cause some errors (e.g. missing matching, redundancies, etc.), when it meets some complex case (e.g. polysemy vocabulary, synonymy vocabulary, etc.).

Athena² repository already contains the related terms and synonyms of the monitory number of concepts, which resides as a tabular data in the database. Thereby, these concepts and their related terms could be transformed onto the ontology by the ontology learning from tabular data. However, the synonyms and related concepts of the majority terms is missing, in this case, the ontology learning and existing knowledge bases could be utilized to construct and enrich the ontology. As depicted in Fig. 10, the standard vocabulary ontology of OMOP CM provides the hierarchical relationship of object property and data property between different fields in the OMOP CDM standard concept. However, ontology is a general data model,



Figure 10: Visualized ontology of the OMOP CDM Standard Vocabulary

namely, it only model general types of things that share certain properties, but does not contain information of the specific individuals (data). In this case, we may construct the knowledge graphs (KGs) based on the ontology, which will contains the knowledge of both general hierarchical and specific individuals.

In general, there are three different cases when mapping the local vocabulary into the OMOP CDM standardized vocabulary.

²https://athena.ohdsi.org/

- **Exact Mapping**. The local vocabulary included by the OMOP CDM standardized vocabulary, which could be mapped into the OMOP CDM standardized vocabulary exactly.
- Equivalent Mapping. The local vocabulary does not included by the OMOP CDM standardized vocabulary, while the semantic equivalent concept exists in the OMOP CDM standardized vocabulary.
- Missing Mapping. The local vocabulary does not included by the OMOP CDM standardized vocabulary, and there does not exists the equivalent concepts in the OMOP CDM standardized vocabulary.

The different methods will be employed to tackle above cases, for instance, the local vocabulary will be imported into the OMOP CDM standardized vocabulary in the exact mapping and equivalent mapping.

4.4.2 Data Table Mapping

Data tables mapping is a mapping process that map the names of data tables and columns in the source schema to the corresponding data-tables in the target schema. The column mapping is a essential process of the data tables mapping, in which the correspondences of the columns name and type between data source table and target table plays a crucial roles.

In our case, the relevant fields of data tables in e-MedSolution will be mapped into the corresponding fields of data tables in the OMOP CDM. Considering the correspondence between the hun_case and hun_case_diag, we attempt to map these two tables in e-MedSolution system to the OMOP CDM CONDITION_OCCURRENCE table. The Fig. 13 in the Appendix A.2 depicts the correspondence between different fields. In this schematic diagram of data table mapping, the solid line represents the certain matching, while the dashed line represents the uncertain matching. The main reason behind these uncertain matching is the heterogeneity (e.g. fields name, fields type, etc.), which requires the domain experts to confirm.

As we mentioned in Sec.3.1, the existing knowledge bases could provides a semantic reference between these column name and type. More precisely, an ontology was constructed by considering the correspondence between OMOP CDM Condition_Occurrence and hun_case_diag in Fig. 11.

In this preliminary ontology, we transformed the Condition_Occurrence and its related data table onto the ontology, in which only the referential relationships (object property relationship) between different tables are considered. Nevertheless, it is worth to mention that the relationship (data property relationships) between the column or properties in e-MedSolution data model and the column or properties OMOP CDM determine the semantic correspondence. Therefore, a tailored ontology could be constructed by utilizing the the existing knowledge bases and employing ontology learning, which will enrich the semantic reference for the schema matching.

5 Summary and Future Wrok

5.1 Summary

It is a tedious work to identify the semantic correspondences among multiple schemas while mapping and migrating the large-scale data models from source database



Figure 11: Visualized ontology of the OMOP CDM Condition_Occurrence

into the target database. To address this issues, a general framework of ontology learning and knowledge enriched schema mapping were proposed and a preliminary case study was conducted in this work. Accordingly, the main contribution of this work could be summarized as followings: (i) we proposed a framework of knowledgeenriched schema mapping and ontology learning, (ii) we analyzed and compared the heterogeneity between e-MedSolution data model and OMOP CDM data model, and (iii) we designed a schematic diagram of schema mapping and constructed a tailored ontology for the vocabulary mapping and data table mapping. Additionally, we investigated how the ontology and other knowledge bases (e.g. knowledge graph) could be employed in vocabulary mapping and data table mapping.

5.2 Future Work

Despite of knowledge-enriched schema mapping could greatly harmonize the heterogeneity between source schema and target schema, the bottleneck of this approach is how to (semi-)automatically constructed tailored knowledge bases (e.g., ontology, knowledge graph, etc). In particular, the healthcare information systems (HIS) is a comprehensive information systems, which contains a several sub-modules e.g., clinical module, medication module, occurrence module, etc, and terminology. Accordingly, a lot of tailored knowledge bases should be constructed to provide the excellent semantic reference for vocabulary mapping and data table mapping. This is work a preliminary work currently, there are several works need to be investigated in the future: (i) investigate an ontology learning algorithm from relational data and tabular data, (ii) design an tailored ontology and knowledge graph based on ontology learning to provide the semantic reference for identifying the semantic correspondence between source and target schema mapping, and (iii) develop a (semi-)method for eliminating the duplicated mappings and transforming and loading the data from source database to the target database based on the mappings.

References

- Mohammad Arief Faizal Rachman and Gusti Ayu Putri Saptawati. "Database integration based on combination schema matching approach (case study: Multi-database of district health information system)". In: 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE). IEEE. 2017, pp. 430–435. ISBN: 9781538606582. DOI: 10.1109/ICITISEE.2017.8285544.
- [2] Luna Dong Xin. "Motivation: Challenges and Opportunities for BDI". In: Big Data Integration (Synthesis Lectures on Data Management). Morgan & Claypool, 2015. ISBN: 9781627052238. DOI: 10.2200/S00578ED1V01Y201404DTM040.
- [3] Gerhard Weikum et al. "Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases". In: Foundations and Trends in Databases. 2020. arXiv: 2009.11564v1.
- [4] Julien Fauqueur, Ashok Thillaisundaram, and Theodosia Togia. "Constructing large scale biomedical knowledge bases from scratch with rapid annotation of interpretable patterns". In: *Proceedings of the 18th BioNLP Workshop and Shared Task.* 2019, pp. 142–151.
- [5] Giuseppe De Giacomo et al. "Using Ontologies for Semantic Data Integration". In: *Studies in Big Data*. Springer International Publishing, 2017, pp. 187–202. DOI: 10.1007/978-3-319-61893-7_11.
- [6] Ana Ozaki. "Learning Description Logic Ontologies: Five Approaches. Where Do They Stand?" In: KI - Künstliche Intelligenz (2020). ISSN: 1610-1987. DOI: 10.1007/s13218-020-00656-9.
- [7] Ariel Fuxman and Renée J. Miller. "Schema Mapping". In: Encyclopedia of Database Systems. Springer US, 2009, pp. 2481–2488. DOI: 10.1007/978-0-387-39940-9_964.
- [8] Erhard Rahm. "Towards Large-Scale Schema and Ontology Matching". In: Schema Matching and Mapping. Ed. by Zohra Bellahsene, Angela Bonifati, and Erhard Rahm. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3– 27. ISBN: 978-3-642-16518-4. DOI: 10.1007/978-3-642-16518-4_1.
- [9] Erhard Rahm and Philip A Bernstein. "A survey of approaches to automatic schema matching". In: *the VLDB Journal* 10.4 (2001), pp. 334–350.
- [10] Jayant Madhavan et al. "Corpus-based schema matching". In: 21st International Conference on Data Engineering (ICDE'05). IEEE. 2005, pp. 57–68.
- [11] A. Bonifati et al. "Schema Mapping Verification: The Spicy Way". In: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology. EDBT '08. Nantes, France: Association for Computing Machinery, 2008, pp. 85–96. DOI: 10.1145/1353343.1353358.
- [12] Jacob Berlin and Amihai Motro. "Database schema matching using machine learning with feature selection". In: International Conference on Advanced Information Systems Engineering. Springer. 2002, pp. 452–466.
- [13] Henrik Nottelmann and Umberto Straccia. "Information retrieval and machine learning for probabilistic schema matching". In: *Information processing* & management 43.3 (2007), pp. 552–576.

- [14] Lev Bulygin. "Combining lexical and semantic similarity measures with machine learning approach for ontology and schema matching problem". In: Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2018). 2018, pp. 245–249.
- [15] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. "Semantic Schema Matching". In: On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. Ed. by Robert Meersman and Zahir Tari. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 347–365. ISBN: 978-3-540-32116-3.
- Philip A. Bernstein, Jayant Madhavan, and Erhard Rahm. "Generic Schema Matching, Ten Years Later". In: 4.11 (2011), pp. 695–701. ISSN: 2150-8097.
 DOI: 10.14778/3402707.3402710.
- [17] Christian Drumm et al. "Quickmig: Automatic Schema Matching for Data Migration Projects". In: CIKM '07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 107–116. ISBN: 9781595938039. DOI: 10. 1145/1321440.1321458.
- [18] Abadie Nathalie. "Schema matching based on attribute values and background ontology". In: 12th AGILE International conference on geographic information science. Hannover, Germany, 2009, pp. 1–9.
- [19] Harith Alani and Saidah Saad. "Schema matching for large-scale data based on ontology clustering method". In: International Journal on Advanced Science, Engineering and Information Technology 7.5 (2017), pp. 1790–1797.
- [20] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. "Ontology learning from text: An overview". In: Ontology Learning from Text: Methods, Evaluation and Applications. Ed. by Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. Amsterdam: IOS Press, 2005, pp. 3–12.
- [21] Alexander Maedche and Steffen Staab. "Ontology learning for the semantic web". In: *IEEE Intelligent systems* 16.2 (2001), pp. 72–79.
- [22] Muhammad Nabeel Asim et al. "A survey of ontology learning techniques and applications". In: *Database* 2018 (2018), pp. 1–24.
- [23] Wang Hong, Zhang Hao, and Jinchuan Shi. "Research and Application on Domain Ontology Learning Method Based on LDA." In: JSW 12.4 (2017), pp. 265–273.
- [24] Wei Gao et al. "Partial multi-dividing ontology learning algorithm". In: Information Sciences 467 (2018), pp. 35–58.
- [25] Sara Sbai et al. "Using Reverse Engineering for Building Ontologies with Deeper Taxonomies from Relational Databases". In: *Journal of Software* 14.3 (2019), pp. 138–145. DOI: 10.17706/jsw.14.3.138-145.
- [26] Mona Dadjoo and Esmaeil Kheirkhah. "An Approach For Transforming of Relational Databases to OWL Ontology". In: International Journal of Web & Semantic Technology 6.1 (2015), pp. 19–28.
- [27] Bouchra El Idrissi, Salah Baïna, and Karim Baïna. "Ontology learning from relational database: How to label the relationships between concepts?" In: *Beyond Databases, Architectures and Structures.* Ed. by Stanisław Kozielski et al. Cham: Springer International Publishing, 2015, pp. 235–244. DOI: 10. 1007/978-3-319-18422-7_21.

- [28] Mohamed AG Hazber et al. "An Approach for Generation of SPARQL Query from SQL Algebra based Transformation Rules of RDB to Ontology." In: *Journal of Software* 13.11 (2018), pp. 573–599. DOI: 10.17706/jsw.13.11. 573–599.
- [29] Ben Bogin, Matt Gardner, and Jonathan Berant. "Representing Schema Structure with Graph Neural Networks for Text-to-SQL Parsing". In: 57th Annual Meeting of the Association for Computational Linguistics. 2019, pp. 4560– 4565.
- [30] Jiaqi Guo et al. "Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation". In: 57th Annual Meeting of the Association for Computational Linguistics. 2019, pp. 4524–4535.
- [31] Ondřej Šebek et al. "Semi-automatic Tool for Ontology Learning Tasks". In: *Industrial Applications of Holonic and Multi-Agent Systems*. Ed. by Vladimír Mařík et al. Cham: Springer International Publishing, 2019, pp. 119–129. DOI: 10.1007/978-3-030-27878-6_10.
- [32] Ting Wang et al. "Multi-source knowledge integration based on machine learning algorithms for domain ontology". In: *Neural Computing and Applications* 32.1 (2020), pp. 235–245.
- [33] R. Navigli, P. Velardi, and A. Gangemi. "Ontology learning and its application to automated terminology translation". In: *IEEE Intelligent Systems* 18.1 (2003), pp. 22–31. DOI: 10.1109/mis.2003.1179190.
- [34] Giulio Petrucci, Marco Rospocher, and Chiara Ghidini. "Expressive ontology learning as neural machine translation". In: *Journal of Web Semantics* 52-53 (2018), pp. 66–82. ISSN: 1570-8268. DOI: 10.1016/j.websem.2018.10.002.
- [35] Godandapani Zayaraz et al. "Concept relation extraction using Naïve Bayes classifier for ontology-based question answering systems". In: Journal of King Saud University-Computer and Information Sciences 27.1 (2015), pp. 13–24. DOI: 10.1016/j.jksuci.2014.03.001.
- [36] Edward H. Y. Lim, James N. K. Liu, and Raymond S. T. Lee. "Collaborative Content and User-Based Web Ontology Learning System". In: Knowledge Seeker - Ontology Modelling for Information Search and Management: A Compendium. Springer, Berlin, Heidelberg, 2011, pp. 181–194. ISBN: 978-3-642-17916-7. DOI: 10.1007/978-3-642-17916-7_12.
- [37] Hongsheng Xu and Ruiling Zhang. "Research on Data Integration of the Semantic Web Based on Ontology Learning Technology". In: *TELKOMNIKA Indonesian Journal of Electrical Engineering* 12.1 (2014), pp. 167–178.
- [38] Jingliang Chen et al. "Smart Data Integration by Goal Driven Ontology Learning". In: Advances in Big Data. Ed. by Plamen Angelov et al. Springer. Cham: Springer International Publishing, 2017, pp. 283–292. ISBN: 978-3-319-47898-2. DOI: 10.1007/978-3-319-47898-2_29.
- [39] Observational Health Data Sciences and Informatics. *The Book of OHDSI*. 2020. ISBN: 978-1-088-85519-5.
- [40] Guoqian Jiang et al. "A consensus-based approach for harmonizing the OHDSI common data model with HL7 FHIR". In: Studies in health technology and informatics 245 (2017), p. 887.

A Appendix

- A.1 Concise database schema version of e-MedSolution
- A.2 Table matching between e-MedSolution and OMOP CDM

MRE_JOO numeric RCP_SUPPLY_DAY numeric TYPE var OTY numeric Num_case_J RCP_NUM numeric TYPE var PERFAC_U2N numeric CASE_NO numeric REQ_DISP_GROUP varchar DESCR_ORIG var PERFAC_U3C numeric CASE_TYPE varchar DURA numeric DESCR_ORIG var PERFAC_U4C numeric CASE_TYPE varchar TIME_UNIT varchar DIG_DATE var PERFAC_U5C numeric CASE_STAT varchar REC/DE_REP numeric DIAG_TIME var PERFAC_U6C numeric STAT_SEQ numeric MDD_INT_CHK varchar DOCTOR var PERFAC_U6C numeric REF_DOC varchar MDD_IND_CHK varchar DOCTOR var PERFAC_CCREATE_DATE date ADMIT_TIME numeric LAIM varchar Nun_case_o	hun_patient		hun_oupt_sched_sp	pec	hun_medication		hun_drug	
BETMANE BETMANE <t< th=""><th>IRN</th><th>varchar</th><th>SPEC</th><th>varchar</th><th>MEDICATION_ID</th><th>numeric</th><th>UNIQUE_ID</th><th></th></t<>	IRN	varchar	SPEC	varchar	MEDICATION_ID	numeric	UNIQUE_ID	
RENNARWADEPUT JUTWADEMADE JUDWADE JUD			DESCR				DRUG_ID	
Name Name <th< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>CODED</td><td></td></th<>							CODED	
Norm Norm <th< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></th<>								
Norm Norm <th< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></th<>								
NITALCE Name Norm		varchar	FROM_DATE	date	TYPE	varchar		
No. Cali Nor. Cali <th< td=""><td>OB</td><td>date</td><td>TO_DATE</td><td>date</td><td>DESCR</td><td>varchar</td><td></td><td></td></th<>	OB	date	TO_DATE	date	DESCR	varchar		
ActionAndrewAddrewAndrewAddre	IRTHPLACE	varchar			DOSAGE	varchar		
MATA	PI	varchar	hun_case		FIRST_ADM_DATE	date	BRANDNAME	
AMPL AMPL CASE_TYPE AMPL CASE_MAPL AMPL CASE_TYPE AMPL AMPL CASE_TYPE AMPL AMPL CASE_TYPE AMPL AMPL CASE_TYPE AMPL AMPL AMPL CASE_TYPE AMPL AMPL AMPL CASE_TYPE AMPL AMPL <td>EX</td> <td>varchar</td> <td>CASE_NO</td> <td>numeric</td> <td>FIRST_ADM_TIME</td> <td>time</td> <td>SHORT_DESCR</td> <td></td>	EX	varchar	CASE_NO	numeric	FIRST_ADM_TIME	time	SHORT_DESCR	
NAMECASE_EDTNAMECASE_EDTNAMECASE_EDTNAMECASE_EDTNAMECASE_EDTNAMECASE_EDTNAME	IARTAL	varchar	MRN	varchar	LAST_ADM_DATE	date	RECIPE_DESCR	
ALM, MOwordsCARE, STATWordsCARE, STATMoreACTURNBLTL, MERMarkCARE, STATGalaCARE, MARWardsCOD, FROMNOPESIONWordsADMT, TAREGalaNODESIONCOD, FROMCOD, FROMNOPESIONWordsADMT, TAREGalaNODESIONCOD, FROMCOD, FROMNORDS, NAMEWordsCARE, STATGalaNODESIONCOD, MARTCOD, MARTNODEN, NAMEWordsCARE, STATGalaNOTESIONCOD, MARTNODEN, NAMEWordsCOD, MARTCOD, MARTCOD, MARTNODEN, CAREMARTCODCOD, MARTCOD, MARTCOD, MARTCODCODWordsCOD, MARTNO, MARTMARTCODCODWordsCOD, MARTCOD, MARTCODCODWordsCOD, MARTCOD, MARTMARTWordsCOD, MARTCOD, MARTCOD, MARTCODWordsCOD, MA	IATION	varchar	CASE TYPE	varchar	LAST ADM TIME	time	CLASS	
Athl ATEdataSTAT SEQmatchGRU PHENUDwebsiteATCAthl TakeREP DOCwebsiteGRU DOmatchGRU DOGRU DOGRU DOSCHORESIONwebsiteAMUT TAKEWebsiteGRU CONTONGRU DOGRU DOGRU DOMERELYMMEWebsiteGRU PHENUMEGRU DOMERELYMEGRU DOGRU DOGRU DOMERELYMMEWebsiteGRU PHENUMEGRU PHENUMEGRU PHENUMEGRU PHENUMEGRU PHENUMEMERELYMMEWebsiteGRU PHENUMEGRU PHENUMEGRU PHENUMEGRU PHENUMEGRU PHENUMESCADAWebsiteGRU PHENUMEGRU PHENUMEGRU PHENUMEGRU PHENUM							FACTORY	
RanRanRefCodeRef <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>ATC</td> <td></td>							ATC	
Date of the set o		date						
Non-United<	EATH_TIME	numeric	REF_DOC	varchar	DRUG_ID	numeric		
MADER, MARE MADR. R. R. SON worder PRES., COTV mandel GEN. CODE MADR. MARE worder MADR. R. MARE Worder MADR. N. MARE ALRAN Worder MART. R. MARE Worder GEN. CODE S. NO Worder MART. R. MARE Worder GEN. CODE N. W. WIELEL Worder GEN. L. MARE GEN. L. MARE N. W. WIELEL Worder GEN. L. MARE GEN. L. MARE S. J. M. W. WIELEL Worder GEN. L. MARE GEN. L. MARE S. J. M. W. WIELEL Worder GEN. L. MARE GEN. L. MARE S. J. M. W. WIELEL Worder GEN. L. MARE GEN. L. MARE S. J. M. W. WIELEL Worder GEN. L. MARE GEN. L. MARE S. J. M. W. WIELEL Worder GEN. L. MARE GEN. L. MARE S. J. M. W. WIELEL Worder GEN. L. MARE GEN. L. MARE S. M. M. WIELE MARE MARE MARE GEN. L. MARE S. M. M. WIELE MARE MARE MARE GEN. MARE S. M. M. WIELE MARE MARE MARE GEN. MARE S. M. M. WIELE MARE MARE GEN. MARE GEN. MARE S. M. M. WIELE MARE MARE <td< td=""><td>ROFESSION</td><td>varchar</td><td>ADMIT_DATE</td><td>date</td><td>INDICATION</td><td>varchar</td><td>DOS_FROM</td><td></td></td<>	ROFESSION	varchar	ADMIT_DATE	date	INDICATION	varchar	DOS_FROM	
NAME NAME <th< td=""><td>DUCATION</td><td>varchar</td><td>ADMIT_TIME</td><td>numeric</td><td>INDICATION_DESCR</td><td>varchar</td><td>ROUTE</td><td></td></th<>	DUCATION	varchar	ADMIT_TIME	numeric	INDICATION_DESCR	varchar	ROUTE	
Decker (1990)Decker (1990)Decker (1990)Decker (1990)Decker (1990)Decker (1990)R. perform (2000)marrierSTOMACER (1990)Warder (1990)STOMACER (1990)Warder (1990)Decker (1990)N. M. Warder (1990)CODIDWarder (1990)ODEP (1991)Warder (1990)Decker (1990)VIEZN. D. MARE (1990)Warder (1990)DEP (1991)Warder (1991)Decker (1990)Decker (1990)N. T. M. WARE (1990)Warder (1990)DEP (1991)Warder (1991)Decker (1990)Decker (1990)N. T. M. WARE (1990)Warder (1990)DEP (1990)Warder (1990)Decker (1990)Decker (1990)SE, NOWarder (1990)ODER (1990)Warder (1990)DECKER (1990)DECKER (1990)DECKER (1990)SE, NOWarder (1990)DECKER (1990)Warder (1990)DECKER (1990)DECKER (1990)DECKER (1990)SFAC, LUINWarder (1990)DECKER (1990)Warder (1990)DECKER (1990)DECKER (1990)DECKER (1990)SFAC, LUID (1990)Warder (1990)DECKER (1990)Warder (1990)DECKER (1990)DECKER (1990)DECKER (1990)SFAC, LUID (1990)Warder (1990)DECKER (1990)Warder (1990)DECKER (1990)DECKER (1990)DECKER (1990)SFAC, LUID (1990)Warder (1990)MARE (1990)Warder (1990)DECKER (1990)DECKER (1990)DECKER (1990)SFAC, LUID (1990)Warder (1990)Warder (1990)Warder (1990)DECKER (1990)DECKER (1990)DECKER (1990)SFAC,	OTHER_NAME	varchar	ADMIT_REASON	varchar	PRESC_QTY	numeric	GEN_CODE	
Applorm.JorATTEND, DOCwatherNTERACT, OKwatherDOD, UNITC.MOnumberPAY, CAGwatherNTERACT, PERSwatherDOD, ONNNARwatherCOODEwatherPAY, CAGWatherPAY, CAGDOD, ONNNARwatherCOODEwatherPAY, CAGWatherPAY, CAGWatherDOD, ONNNARwatherCOODEwatherPAY, CAGWatherPAY, CAGWatherDOD, ONNNATEwatherCOODEwatherPAY, CAGWatherPAY, CAGWatherDOD, ONNND, TimemannerPAY, CAGwatherPAY, CAGWatherDOD, NATDOD, UNITC, DATEdasSOUCEwatherDOS, MART, PTPEwatherDOD, MARTDOD, MARTSR, NOmannerCASE, INOmannerPAY, DAGwatherDOD, MARTPARA, MAR	AIDEN_NAME	varchar	DEPT	varchar	REPL_GROUP	varchar	MANU_TYP	
m. perform.sec PAY_CATG without STORAGE UNITERAT_RES Without DO_DAN R2.00 muthout COORDE without COORDE Without DO_DAN SN without COORDE without DOB_DO_UNIC WITZ NO_TABE muthout PM_APR Without DOB_DOU_UNIC WITZ NO_TABE muthout PMADER Without DOB_DOU_UNIC WITZ SOUTOC without DOB_DOU_UNIC WITT DOB_DOU_UNIC WITZ SSIAO muthout CASE_ID Without PLANE WITT SSIAO muthout CASE_ID WITT							DOS_UNIT	
C.M.O. Mathematic Math Mathematic Mathematic	n_perform_acc							
CLOBELWatchCLOBELSWatchCLOBELSWatchCLOBELSWatchCLOBELSWatchVISZNWatchNO, TAGNO, TAGNO, TAGWatchDBP, APP, PP, PP, PP, PP, PP, PP, PP, PP,	C_NO	numeric						
N CLOUD Watched PH, APP Watched DELLER, DESCR NN, TWE data BNS, STAT Watched DSP, DELL, LAG Watched DSP, DELL, LAG Watched DSP, DELL, LAG DSP, DELL,								
NAME No. TATE No. No. TATE No. T								
No. TAME No. TAME No. TAME No. TAME No. TAME DBPREU_CO No. TAME DALE LL_CONV_DECK C_G. TAT No. TAME SOURCE No. TAME DBPREU_CO No. TAME DALE LL_CONV_DECK NATS No. TAME SOURCE No. TAME UPDATE No. TAME DELTE _ DATE Y_G. TAME AGC_STM No. TAME No. CASE_ JO No. No. TAME DELTE _ DATE DELTE _ DATE Y_G. TAME No. TAME AGC_STM No. TAME No. CASE_ JO No. No. TAME CASE_ JO No. No. TAME Y_G. TAME No. TAME No. CASE_ JTYPE No. TAME No. CASE_ JTYPE No. TAME DURA No. CASE_ JTYPE No. TAME DURA DURA DURA DURA_DELTE DECORE_ NO.			NEW_RESULT	varchar	DISP_DEL_FLAG	varchar		
C, STAT windtar PRANCLER windtar Disc. Stat Disc. Stat Disc. Stat C, SATE windtar SOURCE windtar VER, NO minutes DELITE, REASON VER, NO windtar SOURCE windtar UP, DATE windtar VER, NO minutes ADMIT_TYPE windtar TMER_CMIT windtar VER, NO minutes STAT_SEQ minutes MOD_MAT_CRK windtar VERA_UBC minutes ADMIT_TME minutes MOD_MAT_CRK windtar VERA_UBC windtar ADMIT_TME minutes MOD_MAT_CRK windtar VERA_UBC windtar ADMIT_T			INS_STAT	varchar	DISP_MOD_FLAG	varchar	DDD_UNIT	
C.DATE data INTERNUTIVE Vacidation VEX.NDT Vex.NDATE DELETE_REASON NINST Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT DELETE_REASON V_CATG Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT DELETE_REASON Y_CATG Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT Y_CATG Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT Vex.NDAT DESCR_VDAT Vex.NDAT DESCR_VDAT Vex.NDAT DESCR_VDAT DESCR_VDAT DESCR_VDAT DESCR_VDAT DESCR_VDAT DESCR_VDAT Vex.NDAT DESCR_VDAT DESCR_VDAT <t< td=""><td>ND_TIME</td><td>numeric</td><td>FINANCIER</td><td>varchar</td><td>DISP_DRUG_ID</td><td>numeric</td><td>DEALER_LC_OWN_DESC</td><td>R</td></t<>	ND_TIME	numeric	FINANCIER	varchar	DISP_DRUG_ID	numeric	DEALER_LC_OWN_DESC	R
LCDATE data SOURCE wathat UP_DATE wathat NINST wathat CASE_ID wathat UP_TME wathat SSE_NO numete DADRT_TYME wathat UP_DATE wathat SSE_NO numete ADMT_TYME wathat NECKE Num_cease_I RFAC_UM numete CASE_NO numete CASE_NO Numete RFAC_USC numete CASE_NO numete DEGR_ORDOP wathat RFAC_USC numete CASE_NO numete DEGR_ORDOP wathat RFAC_USC numete CASE_NO numete DEGR_ORDOP wathat RFAC_USC numete CASE_NO numete NUM_CONC_NK wathat RFAC_USC numete ADMT_TME numete NUM_CONC_NK wathat RFAC_USC numete ADMT_TME numete NUM_CONC_NK wathat RFAC_USC Numete DEGR_ORDON wathat NUM_CONC_NK Wathat RFAC_USC Numete NUM_CONC_NK Wathat NUM_CONC_NK NUM_CONC_NK SFAC_ORDESCON_VARIA NUMEN Wathat NUM_CONC_NK Wathat SFAC_ORDESCON_VARIA NUM_CONC_NK	C_STAT	varchar	INTERN_TYPE	varchar	VER_NO	numeric	DELETE_DATE	
NNST watchar CASE_ID watchar UP_TME watchar Y_CATG watchar ACC_STM watchar UP_TME watchar SENO numeric AMT_TYPE watchar UP_TME watchar Y numeric CASE_NO numeric REO_DSP_GROUP watchar NERA_UNO numeric CASE_NO numeric REO_DSP_GROUP watchar NERA_UNC numeric CASE_NO numeric REO_DSP_GROUP watchar NERA_UNC numeric CASE_NO numeric REO_DSP_GROUP watchar NERA_UNC numeric CASE_TYPE watchar MOD_NT_CMK watchar REA_UNC numeric CASE_TYPE watchar MOD_NT_CMK watchar REA_UNC numeric REF_DCO watchar MOD_NT_CMK watchar REA_UNC watchar AMMT_TME mumeric MOD_NT_CMK watchar SEGOR watchar REF_DCO watchar MOD_NT_CMK watchar SEGOR watchar REG_DS_COMMENT watchar MOD	C_DATE	date	SOURCE				DELETE_REASON	
V. CAG variant ACC_STM variants PURGE_UD variants 15E, NO numetic NUMT_TYPE variants PURGE_UD numetic Y. V. M. numetic NUM_CES_UP numetic PURGE_UD numetic Y. V. M. numetic CASE_NO numetic PUR_C_UD numetic RFAC_USC numetic MRN variants PUR_C_UD numetic RFAC_USC numetic REC_DSS_GROUP variants PUR_C_USC	NINST	varchar						
SENO Number	AY CATG	varchar						
F. JOO Numeric Numeric Numeric Numeric CASE_NO Numeric NPAC_UNN numeric Numeric RPAC_UNC Numeric NUM				varchar		varchar	hun_case_diag	
P. J. O Immedia Immedia <t< td=""><td></td><td></td><td>ADMIT_TYPE</td><td>varchar</td><td>PMGRP_ID</td><td>numeric</td><td>CASE NO</td><td>numeric</td></t<>			ADMIT_TYPE	varchar	PMGRP_ID	numeric	CASE NO	numeric
Image: Market is intermediate inte					RCP_SUPPLY_DAY	numeric		varchar
ANACDN inimite CASENO numeric REQDISPGROUP watchar DESCRORIG watchar RFACUNC numeric CASEYPE watchar DURA numeric DESCRQRUCP watchar RFACLUNC numeric CASEYPE watchar DURA numeric DURA numeric RFACLUNC numeric STATSEO numeric MODN.TOHK watchar RFACCREA.TE_DATE data ADMIT_DATE data MODN.C.HK watchar RFACCREA.TE_DATE data ADMIT_DATE data CASENO NUMCREA LOCOE char EXPSTAY numeric CAMENTNECNT CASENO LCOD.CATG watchar EXPSTAY numeric CASENO NUM LCDCATG watchar DEFT watchar CASENO NUM LCDCATG watchar DISCHYPE watchar STAT ADMITTAE DATE	Υ	numeric	hun_case_i		RCP_NUM	numeric		
Number of the second	RFAC_U1N	numeric	CASE_NO	numeric	REQ_DISP_GROUP	varchar		varcha
PFAC_U3C number CASE_TYPE watchar THE_UNIT watchar DAG_DATE watchar RFAC_USC number CASE_TAT watchar RECUFE_REP number DAG_DATE MD DAG_DATE MD DAG_DATE MD DAG_TIME	RFAC_U2N	numeric	MRN	varchar	DURA	numeric		varcha
RFAC_LUAC numeic CASE_STAT varchar RECPE_REP numeic DAG_DATE NAG_TIME RFAC_USC numeic STAT_SEQ numeic MDD_INT_CHK varchar DOCTOR var RFAC_CREACE_DATE date ADMIT_DATE date MDD_IND_CHK varchar DOCTOR var RFAC_CREATE_DATE date ADMIT_DATE date Numeic CASE_STAT MDD_IND_CHK varchar _CODE data EXP_STAY_UNIT varchar Numeic CASE_STAT MRN _PENAT numeic REC_COMMENT varchar IN CASE_STAT MRN _PENAT numeic STAT_SEQ varchar CASE_STAT MRN _PENAT numeic NAME varchar CASE_STAT MRN _PENAT numeic Varchar CASE_STAT Varchar CASE_STAT _PENAT numeic NAME varchar CASE_STAT MRN _PENAT numeic Varchar CASE_STAT Varchar CASE_STAT _PENAT numeic NAME varchar CASE_STAT Varchar RETTYPE varchar NAME varchar MAMT_DATE STAT_SEQ </td <td>RFAC_U3C</td> <td>numeric</td> <td>CASE TYPE</td> <td>varchar</td> <td>TIME UNIT</td> <td></td> <td>DESCR_USER</td> <td>varchar</td>	RFAC_U3C	numeric	CASE TYPE	varchar	TIME UNIT		DESCR_USER	varchar
RFAC, UBC numeric STAT_SEQ numeric MDD_INT_CHK varbar DDAT_TIME RFAC, CAC, DESCR varbar ADMIT_DATE data MDD_IND_CHK varbar DOCTOR varbar r,CODE data EXP_STAY maneric data CLAM varbar MR. CASE_NO r,CODE data EXP_STAY_UNIT varbar varbar CASE_TYPE CASE_STAT_UNIT Varbar r,PIAC, CREAT varbar MTERN_TYPE varbar DEFT varbar CASE_STAT_UNIT Varbar r,PIAC varbar STAT_SEQ varbar STAT_SEQ Varbar CASE_STAT_UNIT CASE_STAT_UNIT CASE_STAT r,PIAC varbar DEFT varbar Varbar STAT_SEQ Varbar CASE_STAT r,PIAC varbar DESCH_TYPE varbar STAT_Varbar ADMT_TIME ADMT_TRESON r,PIAC varbar DESCH_TYPE varbar DESCH_TYPE varbar STAT_VARC ADMT_TRESON R,NUM varbar DESCH_TYPE varbar DESCH_TYPE varbar DESCH_TYPE Varbar DEST_STAT Varbar ADMT_TRESON R,NUM varbar DESCH_TYPE varbar <t< td=""><td>RFAC_U4C</td><td>numeric</td><td></td><td></td><td></td><td></td><td>DIAG_DATE</td><td>date</td></t<>	RFAC_U4C	numeric					DIAG_DATE	date
RFRAC_USC numeric NUMERAL NUMERAL NUMERAL NUMERAL DOCTOR ust RFRAC_USC varchar ADMT_DATE date ADMT_DATE date NDD_IND_CHK varchar NDD_IND_CHK varchar CASE_NO S_CODE chir EXP_STAY_UNIT varchar numeric EXP_STAY_UNIT varchar CASE_TYPE CASE_TYPE S_PONT numeric SRC_COMMENT varchar NAME varchar CASE_STAT_SEQ SCD_UQULIF varchar NSID varchar NAME varchar CASE_TYPE SCD_UQULIF varchar NSID varchar NAME varchar ADMIT_ITME SCD_UQULIF varchar NSID varchar STAT_SEQ STAME_NO varchar SCD_UQULIF varchar BOSCH_TYPE varchar STAME_NO varchar SFX_V_TYPE SR_N_TTME EED_NO varchar DISCH_TYPE varchar SFX_V_TYPE ADMIT_ITME SR_N_TTME DISCH_TYPE varchar NAME varchar ADMIT_ITME SR_N_TTME DISCH_TYPE varchar SFX_COMAR varchar ADMIT_ITME SR_N_TTPE varchar DISCH_TYPE varchar	RFAC_U5C	numeric					DIAG_TIME	time
NARC, CGU Name, CGU Variant MEP_DOC Variant RFAC, CG, DESCR Variant ADMIT_DATE data CLAM Varianta RFAC, CG, DESCR Variant ADMIT_TIME numeric CLAM Varianta LCODE char EXP_STAY numeric CLAM Varianta LCODE char EXP_STAY numeric CLAM Varianta LCODE char EXP_STAY numeric CLAM Varianta LFLAG Varianta EXP_STAY numeric CASE_TYPE CASE_STAT LFLAG Varianta ST_TYPE Varianta CASE_TAT CASE_STAT CID_QLALIF Varianta ST_TYPE Varianta STAT STAT Varianta ST_TYPE Varianta BED_NO Varianta STAT Varianta STAT Varianta ST_TYPE Varianta BED_NO Varianta STAT Varianta STAT Varianta RNM Varianta DISCH_TYPE Varianta STAT Varianta STAT STAT ADMIT_DATE RN_N Varianta DISCH_TYPE Varianta STAT Varianta STAT ADMIT_REASON RN_N <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>DOCTOR</td> <td>varchar</td>							DOCTOR	varchar
ADMIT_DATE data CLAM Valena ADMIT_TIME mumaric 1:CODE char EXP_STAV mumaric 1:CDESCR varchar EXP_STAV mumaric 1:CDESCR varchar EXP_STAV mumaric 1:CDESCR varchar EXP_STAV_UNIT varchar ID varchar CASE_TVPE 1:POINT numeric SRC_COMMENT varchar ID varchar CASE_STAT 1:CD_OATG varchar DEPT varchar REF_DOC STAT STAT_SEQ CD_OUALIF varchar NOO varchar STAT Varchar ADMIT_TIME ST_TYPE varchar BED_NO varchar STAT Varchar ADMIT_TIME ST_TYPE varchar DISCH_TYPE varchar ADMIT_TIME ADMIT_TIME ST_TYPE varchar DISCH_TYPE varchar DISCH_TYPE Varchar ADMIT_TIME RN_N varchar DISCH_TYPE varchar NAME varchar PS_CATG RN_N varchar DISCH_TYPE varchar STAT_SEQ STATG PS_CATG RN_N varchar DISCH_TYPE varchar DISCH_TYPE Varchar <				varchar	MDD_IND_CHK	varchar		
ADMT_TIME ADMT_TIME Anumaric ADMT_TIME Anumaric L_DESCR Varchar EXP_STAY numaric L_PESCR Varchar EXP_STAY_UNIT varchar CASE_NO L_PESCR Varchar EXP_STAY_UNIT varchar CASE_TYPE L_PESCR Varchar D varchar CASE_TYPE L_PESCR Varchar DEPT varchar CASE_TAT CD_OLQULF Varchar DEPT varchar STAT_SEQ CD_OLQULF Varchar NID Varchar ADMT_DATE CD_OLQULF Varchar NID Varchar ADMT_TIME ST_TYPE Varchar RED_NO Varchar ADMT_TIME SDQ_EXAM Varchar DECH_TYPE Varchar ADMT_REASON SPC Varchar DEPT Varchar ADMT_REASON SEE_NO numeric CODABLE Varchar STAMP_NO Varchar RN varchar DESC_TTFLAG Varchar DEPT Varchar RN_MC numeric CODABLE Varchar DEPT Varchar RR_NO numeric Varchar DEPT Varchar ADMTTRACHTS ASE_NO <td< td=""><td></td><td></td><td>ADMIT_DATE</td><td>date</td><td>CLAIM</td><td>varchar</td><td>hun case e</td><td></td></td<>			ADMIT_DATE	date	CLAIM	varchar	hun case e	
Label and the set of the se			ADMIT_TIME	numeric				
LiesdedvarcharKarlosKarlosKarlosKarlosCASE_TYPESPLAGwarcharNTERN_TYPEwarcharDKCCASE_STATCASE_STATRRE_TYPEvarcharSRC_COMMENTwarcharNAMEwarcharSTAT_SEQCDD_CATGvarcharDEPTwarcharNAMEwarcharREF_DOCCDD_OUALIFvarcharNSIDvarcharSTATwarcharADMIT_DATECDD_OUALIFvarcharROM_NOvarcharSTATvarcharADMIT_TIMEDSCH_TYPEvarcharDISCH_TYPEvarcharSTAMP_NOvarcharARR_CONDPTvarcharDISCH_TYPEvarcharSTAMP_NOvarcharSRV_TYPEPTvarcharDISCH_TATEdateSTAMP_NOvarcharSRV_TYPEPTvarcharDISCH_TATEdateSTAMP_NOvarcharSRV_TYPEPTvarcharDISCH_TTREdateSTAMP_NOvarcharSRV_TYPENAMEvarcharDISCH_TATEdateVarcharSRV_TYPEPTvarcharDISCH_TATEdateVarcharDISCH_TATEPAT_CATGvarcharDISCH_TATEdateVarcharDISTRN_NOnumericCODEDvarcharSPECvarcharDISTRNE_TATATVarcharNEW_RESULTvarcharDISTDISTRAT_TATEdateSER_SEQnumericWEEK_STARTdateNEW_RESULTRAT_TATEMatarNEWRENPLAG<	_CODE	char	EXP_STAY	numeric				num
EfeAGvarcharintrem, TyPevarcharintrem, TyPeintrem, TyPevarcharintrem, TyPeintrem, TyPeintrem, TyPeintrem, TyPeintrem, TyPeintrem, TyPeintrem, TyPeintrem, TyPeintrem, TyPeintrem, TyPe </td <td>_DESCR</td> <td>varchar</td> <td>EXP_STAY UNIT</td> <td>varchar</td> <td>hun_user</td> <td></td> <td>MRN</td> <td>var</td>	_DESCR	varchar	EXP_STAY UNIT	varchar	hun_user		MRN	var
POINTnumericSRC_COMMENTvarcharEXTERNAL_IDvarcharCASE_STATRE_TYPEvarcharDEPTvarcharNAMEvarcharSTAT_SEQCID_CATGvarcharATTEND_DOCvarcharSTATvarcharREF_DOCCID_TYPEvarcharNSIDvarcharSTATvarcharADMIT_DATECID_OULLIFvarcharROOM_NOvarcharSTATvarcharADMIT_REASONST.TYPEvarcharBED_NOvarcharPROFESSIONvarcharADMIT_REASONPTvarcharDISCH_TYPEvarcharSTAMP_NOvarcharSRV_TYPEPTvarcharDISCH_TYPEdateSTAMP_NOvarcharPROFESSIONvarcharPROFESSIONvarcharDISCH_TIMEdatevarcharDEPTARE_CONDSE, NOnumericCODABLEvarcharVarcharPROFESSIONvarcharRDE_STATvarcharNEW_RESULTvarcharDEPTvarcharRDER_STATvarcharNEW_RESULTvarcharDEPTvarcharRDER_STATvarcharNEW_RESULTvarcharDEPTvarcharRDER_STATvarcharNEWRESSEQnumericNEW_RESULTNEWRESSEQRAT_TIMEtimeSER_SEQnumericWEEK_STARTdateRRV_KEYvarcharNEWREN_FLAGvarcharWEEK_DAV_NNRnumericRRV_DESCRIPTIONvarcharNEWREN_FLAGvarcharMER_TTYPERRV_DESCRIPTIONvarchar <td>_FLAG</td> <td>varchar</td> <td></td> <td></td> <td>ID</td> <td>varchar</td> <td>CASE_TYPE</td> <td>vai</td>	_FLAG	varchar			ID	varchar	CASE_TYPE	vai
SKC_UDMENI varchar NAME varchar STAT_SEQ CICD_CATG varchar DEPT varchar TYPE varchar REF_DOC CICD_UQALIF varchar NSID varchar STAT varchar ATTEND_DOC varchar STTYPE varchar ROOM_NO varchar STAMP_NO varchar ADMT_TIME BOR_EXAM varchar DISCH_TYPE varchar BED_NO varchar STAMP_NO varchar POFESSION varchar DISCH_TYPE date STAMP_NO varchar PS_CATG IPT varchar DISCH_TME date STAMP_NO varchar PS_CATG RNUM varchar DISCH_TATE date Varchar DISCH_TATE date PROFESSION varchar DEPT RNUM varchar DISCH_TATE date varchar DEPT AAR_COND PROFESSION varchar DEPT RNUM numeric CODABLE varchar SPEC varchar ADDTIONAL_RIGHTS RNA varchar NEW_RESULT varchar DEPT varchar CODABLE RAT_TIME time SER_SEQ numeric VEEK_NR numeric NEW	POINT	numeric	_		EXTERNAL_ID	varchar	CASE_STAT	var
kmc_m identify warchar identify warchar identify warchar KEF_DOC CiD_CATOG warchar NSID warchar STAT warchar ADMIT_DATE CiD_CULIF warchar ROOM_NO warchar STAT warchar ADMIT_DATE SIST_TYPE warchar BED_NO warchar STAMP_NO warchar ADMIT_TIME BOR_EXAM warchar DISCH_TYPE warchar ISCH_TYPE warchar STAMP_NO warchar IDISCH_TYPE warchar DISCH_TYPE warchar ISCM_TFLAG warchar DISCH_TYPE Warchar				varchar		varchar	STAT_SEQ	nur
Autor ArtEND_DOC varchar NSID varchar ADMIT_DATE CiD_DQULLIF varchar NSID varchar STAT varchar BOR_EXAM varchar BED_NO varchar STAT varchar BOR_EXAM varchar BISCH_TYPE varchar PROFESSION varchar PT varchar DISCH_TYPE date STAMP_NO varchar DISCH_TYME varchar DISCH_TYME date STAMP_NO varchar PROFESSION varchar DISCH_TYME date STAMP_NO varchar RNUM varchar DISCH_TYME date STAMP_NO varchar BCR_EXAM varchar DISCH_TYME date STAMP_NO varchar RNUM varchar DISCH_TYME date STAMP_NO varchar BCR_STAT DISCH_TYME varchar USER_JO varchar DEPT RNN numeric CODED varchar PROFESSION varchar PROFESSION varchar RNN varchar NEW_RESULT varchar DEPT varchar NODITIONAL_REGTS RNA varchar NEW_RESULT varchar DEPT varchar				varchar				var
NSID varchar NSID varchar ADMIT_TIME ST_TYPE varchar ROOM_NO varchar PROOFESSION varchar SOR_EXAM varchar DISCH_TYPE varchar DISCH_TYPE ARR_COND PT varchar DISCH_TYPE varchar STAMP_NO varchar DISCH_TIME diae STAMP_NO varchar SRV_TYPE n.order DISCH_TIME diae STAMP_NO varchar RNUM varchar DISCH_TIME diae STAMP_NO varchar BEC,NO numeric CODABLE varchar DISCH_TIME DISCH_TIME SID Outrettime varchar NAME varchar PROFESSION varchar RN varchar NEW_RESULT varchar SPEC varchar PATIENT_RIGHTS RN varchar NEW_RESULT varchar DEPT varchar RN varchar SER_SEO numeric VE			ATTEND_DOC	varchar				
RCOM_NO varchar ROOM_NO varchar ROOM_NO varchar BD_NO varchar BED_NO varchar PROFESSION varchar BD_NO varchar DISCH_TYPE varchar SRV_TYPE RNM_MO varchar RNM varchar DISCH_TYPE varchar Mun_doctor SRV_TYPE RNM varchar DISCH_TYPE varchar Mun_doctor ARR_COND BD_NO numeric DISCH_TIME date STAMP_NO varchar BS_NO numeric CODABLE varchar USER_JD varchar RN numeric CODED varchar SPEC varchar RDER_STAT varchar NEW_RESULT varchar SPEC varchar RAT_TIME date SER_NUM numeric CODABLE Varchar RAT_TIME time SER_SEQ numeric CODABLE Varchar RAT_TIME time SER_SEQ numeric VEEK_START date NS_STAT RAT_TIME time SER_SEQ varchar WEEK_DAT_NR INS_STAT RAT_TIME time SER_SEQ varchar WEEK_DAT_NR INS_STAT RER_VEY			NSID	varchar				
BCD_NO varchar BED_NO varchar BCR_EXAM varchar DISCH_TYPE varchar PT varchar DISCH_DYPE varchar DISCH_DATE date STAMP_NO varchar PROFESSION varchar DISCH_TIME time PAY_CATG varchar Varchar DISCH_TIME NAME varchar DISCH_TIME time PAY_CATG varchar Varchar PROFESSION RNO numeric CODED varchar RDER_STAT varchar NEW_RESULT varchar RAR_TIME time SER_SEQ numeric RRV_TYPE varchar NEW_RESULT varchar RDER_STAT varchar NEW_RESULT varchar RAR_TIME time SER_SEQ numeric RRV_TYPE varchar NEWRES, STAT varchar RAR_DATE SER_SEQ numeric VEEK_START date RRV_MEVE Varchar NEWBORN_FLAG varchar VEEK_CNAN_NR NUMEric RRV_TIME time SER_SEQ varchar VEEK_CNAN_NR MURER RRV_EIGHT varchar VEEK_CNAN_NR tatre NEMARUER			ROOM_NO	varchar				nur
Inc. default DisCH_TYPE varchar Inu_dector ARR_COND PT varchar DisCH_DYPE date STAMP_NO varchar PS_CATG R_NUM varchar DisCH_DTIE date STAMP_NO varchar DEFT no ordor PAY_CATG varchar USER_JD varchar DEFT SEE_NO numeric CODABLE varchar PROFESSION varchar PRES_MODE RN varchar NEW_RESULT varchar SPEC varchar ADDITIONAL_RIGHTS RAR_T_DATE date SER_NUM numeric DEPT varchar RRV_KEY varchar SER_SEO numeric CODED WEEK_NR numeric RRV_KEY varchar SER_SEO numeric WEEK_START date NKW_RESULT RRV_KEY varchar FINANCIER varchar WEEK_START date INS_STAT RRV_KEY varchar NEWBORN_FLAG varchar WEEK_DAV_NR numeric INS_STAT RRV_CESULT varchar WEIGHT varchar WEEK_DAV_NR numeric INS_STAT RRV_DESCR_FLIPTION varchar NEWBORN_FLAG varchar WEEK_DAV_NR numeric	ST_TYPE	varchar	BED_NO	varchar	PROFESSION	varchar		va
IpT varchar DISCH_DATE date Intr_BOCCOV Varchar PS_CATG IR_NUM varchar DISCH_TIME time STAMP_NO Varchar PS_CATG an_order ESCORT_FLAG varchar USER_IO varchar DEPT ASE_NO numeric CODABLE varchar Varchar PROFESSION varchar RD_NO numeric CODED varchar SPEC varchar ADDITIONAL_RIGHTS RRDER_STAT varchar INS_STAT varchar DEPT varchar RAT_TIME date SER_NUM numeric CODED Varchar RRDER_STAT varchar INS_STAT varchar DEPT varchar RRT_TIME date SER_NUM numeric VEEK_NR numeric ERV_KEY varchar NEWDRRN_FLAG varchar VEEK_DAY_NR numeric ERV_KEY varchar NEWBORN_FLAG varchar VEEK_DAY_NR numeric ERV_KEY varchar NEWBORN_FLAG varchar SURCE varchar NUTENT_TYPE SURCE varchar VEEK_TAR numeric SURCE varchar SURCE SURCE varchar SURCE </td <td>BOR_EXAM</td> <td>varchar</td> <td>DISCH_TYPE</td> <td>varchar</td> <td></td> <td></td> <td>SRV_TYPE</td> <td>vai</td>	BOR_EXAM	varchar	DISCH_TYPE	varchar			SRV_TYPE	vai
R_NUM varchar DISCH_TIME time time Varchar PS_CATG un_order ESCORT_FLAG varchar NAME varchar DEPT ASE_NO numeric CODABLE varchar PROFESSION varchar PRES_MODE RN_M varchar CODED varchar PROFESSION varchar PATIENT_RIGHTS RN varchar NEW_RESULT varchar DEPT varchar ADDITIONAL_RIGHTS RN_TAT_DATE date SER_NUM numeric DEPT varchar CODABLE RN_TAT_TIME time SER_SEO numeric Varchar DEPT CODABLE RR_KEY varchar NEW_RESULT varchar DEPT varchar CODABLE RR_TAT_TIME date SER_NUM numeric VEEK_NR numeric CODED RRV_KEY varchar FINANCIER varchar VEEK_START date NEW_RESULT RRV_KEY varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric INA_STAT RRV_KEY varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric IL_IND SOURCE varchar Verchar VEEK_TRM <	PT	varchar					ARR_COND	var
un_ordor ESCORT_FLAG varchar USER_ID varchar DEPT ASE_NO numeric CODABLE varchar USER_ID varchar RN varchar CODED varchar Mur_workhour PROFESSION varchar RN varchar NEW_RESULT varchar SPEC varchar ADDITIONAL_RIGHTS RATE_NDATE date SER_NUM numeric DEPT varchar INS_STAT varchar NS_STAT varchar ODED VerEK_NR numeric RRV_RESULT varchar SER_NUM numeric WEEK_START date CODABLE RRV_KEY varchar SER_SEQ numeric WEEK_START date INS_STAT RRV_KEY varchar NEWBORN_FLAG varchar WEEK_DAV_NR numeric INS_STAT SOURCE varchar WEEK_TRM numeric INTERN_TYPE IL_IND WEIGHT varchar WEIGHT varchar SOURCE SOURCE SOURCE TIME_TO numeric TIME_TO numeric SOURCE SOURCE SOURCE	R_NUM	varchar			STAMP_NO	varchar	PS_CATG	var
in order PAY_CATG varchar USER_JD varchar PROFESSION varchar PAY_CATG varchar PROFESSION varchar PROFESSION varchar RD_NO numeric CODD varchar Varchar PATIENT_RIGHTS RD_STAT varchar NEW_RESULT varchar SPEC varchar RDE_STAT varchar NS_STAT varchar ODEP varchar IART_TATE date SER_NUM numeric WEEK_NR numeric ISRY_KEY varchar FINANCIER varchar WEEK_START date ISRY_KEY varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric ISRY_VES Varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric SOURCE varchar WEIGHT varchar SOURCE SOURCE					NAME	varchar	DEPT	vai
n. order PAY_CATG varchar PROFESSION varchar PROFESSION varchar ASE_NO numeric CODABLE varchar Num_workhour PROFESSION varchar RD_NO numeric CODED varchar Num_workhour ADDITIONAL_RIGHTS RN varchar NEW_RESULT varchar SPEC varchar RDE_STAT varchar INS_STAT varchar DEPT varchar TART_DATE data SER_NUM numeric WEEK_START data IRAT_TIME time SER_SEQ numeric WEEK_START data ERV_KEY varchar FINANCIER varchar WEEK_DAY_NR numeric ERV_VES varchar WEIGHT varchar WEEK_DAY_NR numeric SOURCE varchar TIME_FROM numeric SOURCE					USER_ID	varchar	ATTEND_DOC	var
ASE_NO numeric CODABLE varchar Nur_workhour PATIENT_RIGHTS RD_NO numeric CODED varchar SPEC varchar RN varchar NEW_RESULT varchar SPEC varchar RDE_STAT varchar INS_STAT varchar DEPT varchar TART_DATE data SER_NUM numeric WEEK_NR numeric ERV_KEY varchar FINANCIER varchar WEEK_START data NEW_RESULT ERV_VESV varchar FINANCIER varchar WEEK_DAY_NR numeric INS_STAT ERV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric ILL_IND SOURCE varchar RES_TYPE varchar SOURCE SOURCE CASE_JD	un_order		PAY_CATG	varchar	PROFESSION	varchar		vai
ND numeric CODED varchar Num.workhour ADDITIONAL_RIGHTS NN varchar NEW_RESULT varchar SPEC varchar CODABLE xDER_STAT varchar INS_STAT varchar DEPT varchar CODED ART_DATE date SER_NUM numeric WEEK_NR numeric NEW_RESULT ART_TIME time SER_SEQ numeric WEEK_START date NEW_RESULT IRV_KEY varchar FINANCIER varchar WEEK_CDAY_NR numeric INS_STAT IRV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric ILL_IND VBIGHT varchar Varchar WEIGHT varchar SOURCE sourchar SOURCE TIME_FROM numeric TIME_FROM numeric GASE_JD SOURCE SOURCE SOURCE	SE_NO	numeric	CODABLE	varchar				
NN varchar NEW_RESULT varchar SPEC varchar NDER_STAT varchar INS_STAT varchar DEPT varchar ART_DATE date SER_NUM numeric WEEK_NR numeric ART_DATE date SER_SEQ numeric WEEK_START varchar RV_KEY varchar FINANCIER varchar WEEK_CATART date INS_STAT RV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric FINANCIER VEIGHT varchar varchar WEEK_DAY_NR numeric ILL_IND SOURCE varchar Varchar SOURCE varchar SOURCE TIME_FROM numeric TIME_FROM numeric GASE_JD	RD_NO	numeric	CODED	varchar	hun_workhour			vai
KDER_STAT varchar Nar_STAT varchar DEPT varchar CODABLE ART_DATE date SR_NUM numeric WEEK_NR numeric CODE ART_TIME time SR_SEQ numeric WEEK_START date NEW_RESULT IRV_DESCRIPTION varchar FINANCIER varchar WEEK_END date FINANCIER IRV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric FINANCIER SOURCE varchar KES_TYPE varchar source SOURCE SOURCE SOURCE TIME_FROM numeric TIME_FROM numeric GASE_JD SOURCE SOURCE </td <td>RN</td> <td>varchar</td> <td>NEW_RESULT</td> <td>varchar</td> <td>SPEC</td> <td>varchar</td> <td>ADDITIONAL_RIGHTS</td> <td>vai</td>	RN	varchar	NEW_RESULT	varchar	SPEC	varchar	ADDITIONAL_RIGHTS	vai
ART_DATE date function for the function of the							CODABLE	vai
ART_TIME time SER_SEQ numeric WEEK_START date NEW_RESULT IRV_MEX varchar FINANCIER varchar WEEK_END date INS_STAT IRV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric FINANCIER WEIGHT varchar WH_DATE numeric SOURCE varchar NUMERS_TYPE varchar TIME_FROM numeric TIME_FROM numeric CASE_JD							CODED	vai
ART_TIME time SER_SEQ numeric WEEK_START date SRV_KEY varchar FINANCIER varchar WEEK_DAY_NR date SRV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric WEIGHT varchar WH_DATE numeric ILL_IND SOURCE varchar FINAR-STAFT SOURCE TIME_FROM numeric CASE_JD							NEW_RESULT	vai
ERV_KEY varchar FINANCIER varchar WEEK_END data ERV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric WEIGHT varchar WH_DATE numeric ILL_IND SOURCE varchar RES_TYPE varchar TIME_FROM numeric SOURCE SOURCE	TART_TIME	time						var
ERV_DESCRIPTION varchar NEWBORN_FLAG varchar WEEK_DAY_NR numeric WEIGHT varchar WH_DATE numeric SOURCE varchar RES_TYPE varchar TIME_FROM numeric SOURCE TIME_FROM numeric CASE_ID	ERV_KEY	varchar	FINANCIER	varchar	WEEK_END	date		
WEIGHT varchar WH_DATE numeric SOURCE varchar RES_TYPE varchar TIME_FROM numeric SOURCE TIME_TO numeric CASE_ID	RV_DESCRIPTION	varchar	NEWBORN_FLAG	varchar	WEEK_DAY_NR	numeric		var
SOURCE varchar RES_TYPE varchar SOURCE TIME_FROM numeric CASE_ID TIME_TO numeric CASE_ID			WEIGHT	varchar	WH_DATE	numeric		var
TIME_FROM numeric SOURCE TIME_TO numeric CASE_ID							INTERN_TYPE	var
TIME_TO numeric CASE_ID							SOURCE	var
TIME_TO numeric							CASE_ID	var
					TIME_TO	numeric		var

Figure 12: Concise database schema of the clinical module for e-MedSolution

ADMIT_TYPE

Clinical database schema e-MedSolution (Source Schema)	OMOP CDM Clinical Data Module (Target Schema)			
hun_case_i (o)				CONDITION_OCCURRENCE
PK CASE_NO			PI	_
MRN			•	person_id
ADMIT_DATE			•	condition_start_date
ADMIT_TIME				condition_concept_id
ADMIT_REASON	$\left \right _{1}$		•	condition_start_datetime
DISCH_DATE			•	condition_end_date
DISCH_TIME			•	condition_end_datetime
CASE_STAT	 - -		•	condition_status_concept_id
ATTEND_DOCTOR			•	condition_type_concept_id
INTERN_TYPE			•	stop_reason
hun_case_diag				provider_id
CASE_NO				visit_occurrence_id
CODE		<u>!</u>		
ТҮРЕ				condition_source_value
DESCR_ORIG		۰,		condition_source_concept_id
DOCTOR				condition_status_source_value
		· L	•	visit_detail_id
DESCR_USER DIAG_DATE				
DIAG_DATE				

Figure 13: A schematic diagram of data tables matching between hun_case_i(o), hun_case_diag and CONDITION_OCCURRENCE of OMOP CDM