POLITECHNIKA POZNAŃSKA

WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

Instytut Informatyki

Hurtownie Danych i Przetwarzanie Analityczne

ZASTOSOWANIE STANDARDU OMOP W INTEGRACJI DANYCH MEDYCZNYCH

Wojciech Rzeczycki, wojciech.rzeczycki@student.put.poznan.pl, 140770

Dominik Tomkiewicz, dominik.tomkiewicz@student.put.poznan.pl, 140793

Bartosz Pilarczyk, bartosz.g.pilarczyk@student.put.poznan.pl, 141300

POZNAŃ 2022

Spis treści

Rozdział 1

Cel i zakres projektu

Celem pracy jest zaprojektowanie i zaimplementowanie generycznego rozwiązania pozwalającego na transformację danych systemu szpitalnego do schematu zgodnego ze standardem OMOP. Na rozwiązanie składają się takie elementy jak:

- wybór oraz przygotowanie HISu,
- generacja przykładowych danych do wybranego HISu,
- analiza różnicy pomiędzy schematem danych HISu oraz schematem standardu OMOP,
- przygotowanie i opis środowiska pracy (ETL)
- dokonanie transformacji danych do schematu zgodnego ze standardem OMOP.

Struktura pracy jest następująca. W rozdziale drugim opisano analizę istniejących HISów, porównanie ich oraz wybór najbardziej odpowiedniego HISu do projektu (wraz z uzasadnieniem). W rozdziale trzecim dokonano charakterystyki standardu OMOP. Rozdział czwarty poświęcony jest przedstawieniu schematu źródłowego, docelowego wraz z wymaganymi odwzorowaniami. Rozdział piąty został przeznaczony na analizę i wybór ETLów. Rozdział szósty został przeznaczony na implementację i opis działania rozwiązania transformującego dane źródłowe do standardu OMOP.

Rozdział 2

Analiza istniejących HISów

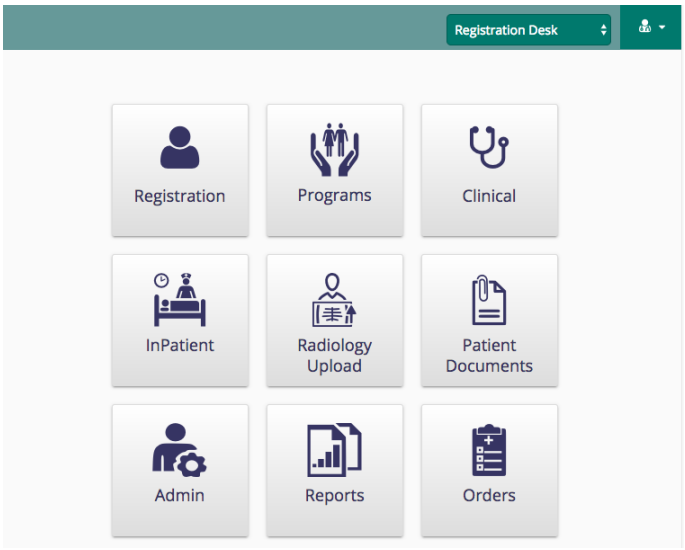
HIS[1] (Hospital Information System) jest elementem informatyki medycznej, który koncentruje się głównie na potrzebach administracyjnych szpitali. W wielu zastosowaniach HIS jest kompleksowym, zintegrowanym systemem informacyjnym przeznaczonym do zarządzania wszystkimi aspektami funkcjonowania szpitala, takimi jak kwestie medyczne, administracyjne, finansowe i prawne oraz związane z nimi przetwarzanie usług.

Szpitalne systemy informacyjne stanowią wspólne źródło informacji o historii zdrowia pacjenta i terminach wizyt lekarskich. System musi przechowywać dane w bezpiecznym miejscu i kontrolować, kto może uzyskać do nich dostęp w określonych okolicznościach. Systemy te zwiększają możliwości koordynowania opieki przez pracowników służby zdrowia poprzez udostępnianie informacji o stanie zdrowia pacjenta i historii wizyt w miejscu i czasie, w którym są one potrzebne.

Na potrzeby pracy ze standardem danych OMOP zespół dokonał analizy open source'owych systemów szpitalnych.

2.1 Bahmni

Bahmni EMR[2] jest rozwiązaniem typu open source, którego celem jest wspieranie opieki zdrowotnej na obszarach wiejskich. Został zbudowany w oparciu o standardy open source i działa na systemie CentOS Linux, który jest stabilną i wydajną dystrybucją systemu Linux klasy korporacyjnej, wspieraną przez społeczność.



Rysunek 2.1: Ekran główny Bahmni.

Bahmni udało się wdrożyć w wielu szpitalach w różnych krajach, takich jak Indie, Bangladesz, Kambodża, Nepal, Sierra Leone, Pakistan, Etiopia i Filipiny ze względu na przeznaczenie dla osób o niskich umiejętnościach informatycznych oraz do pracy w środowiskach o niskich zasobach. Bahmni posiada wbudowane dane przykładowe, lecz ich ilość może okazać się niewystarczająca do przeprowadzania analiz, dodatkowo schemat systemu bliski krajom trzeciego świata może okazać się być zbyt rozbieżny od systemów wykorzystywanych w Europie.

2.2 OpenHospital

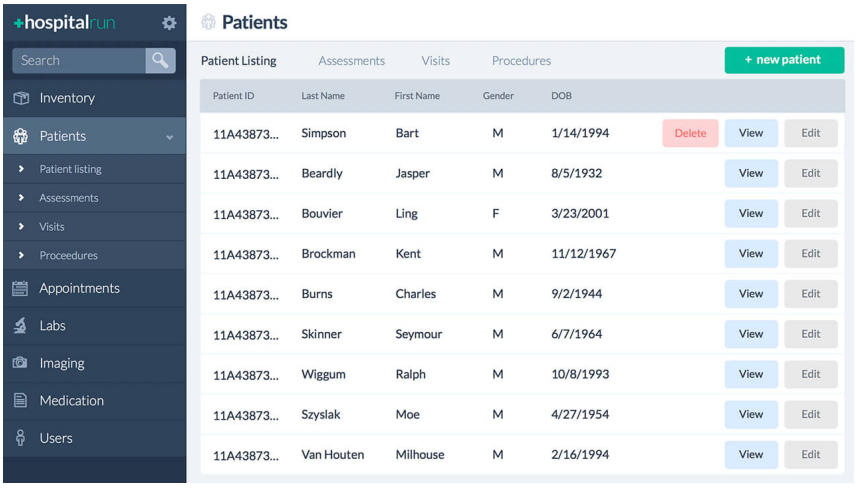
OpenHospital[3] jest wieloplatformowym, opartym na Javie systemem HIS, pochodzącym z Włoch, opracowanym i utrzymywanym przez Informatici Senza Frontiere, organizację non-profit, której celem jest wykorzystanie technologii informatycznych do pomocy krajom rozwijającym się. OpenHospital ma około 23 działających instalacji w 13 krajach, ponieważ obsługuje wiele języków, w tym angielski, włoski, francuski, hiszpański, portugalski, arabski, i holenderski.

Rysunek 2.2: Interfejs systemu OpenHospital.

Interfejs użytkownika jest dość prosty i łatwy w użyciu, biorąc pod uwagę, że jest to szpitalny system informatyczny. Porównując go z innymi systemami HIS, komercyjnymi lub open source, które mają tendencję do zajmowania całego ekranu i demonstrowania pełnej kontroli nad przepływem pracy w interfejsie, co kończy się komplikowaniem go. OpenHealth zapewnia proste menu, które umożliwia dostęp do wszystkich elementów. Archaiczny wygląd systemu może okazać się zaletą ze względu na fakt, iż pracownicy medyczni zwykli pracować na tak wyglądających systemach.

2.3 HospitalRun

HospitalRun[4] to darmowe, otwarte i łatwe w użyciu oprogramowanie EMR dla służby zdrowia w krajach rozwijających się. Zostało uruchomione w 2014 r., a jego najnowsza wersja to (1.0.0 Beta) wydana w dniu (2017-05-18). Oprogramowanie można wdrożyć w różnych środowiskach opieki zdrowotnej. Dzięki właściwościom technicznym umożliwiającym korzystanie z niego nawet bez łączności, nadaje się ono również do klinik znajdujących się w najbardziej wiejskich rejonach świata.

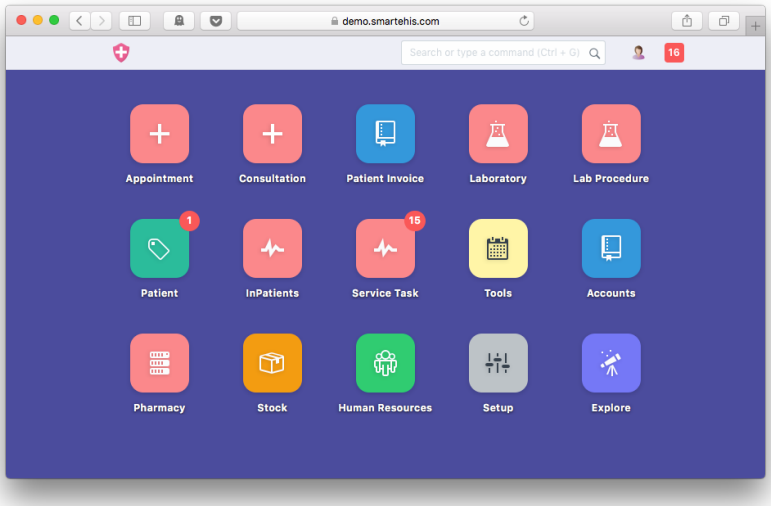


Rysunek 2.3: Przykładowy widok programu HospitalRun.

Interfejs jest zbliżony oprawą do interfejsu Bahmni ze względu na podobne kraje docelowe. Niestety w przeciwieństwie do wielu HISów w HospitalRun istnieje możliwość eksportu danych jedynie do nierelacyjnych plików TypeScript. Istnieje możliwość transformacji, lecz nie jest to celem projektu.

2.4 ERPNext

ERPNext[5] jest rozwiązaniem ERP Open Source, zbudowany przez technologie open source i wydany jako otwarte oprogramowanie na licencji GPLv3, został zbudowany przez kilka ciekawych mieszanek technologii jak Python / JavaScript (NodeJS) przy użyciu MariaDB, może być zainstalowany na Linux, MacOSX i Windows.

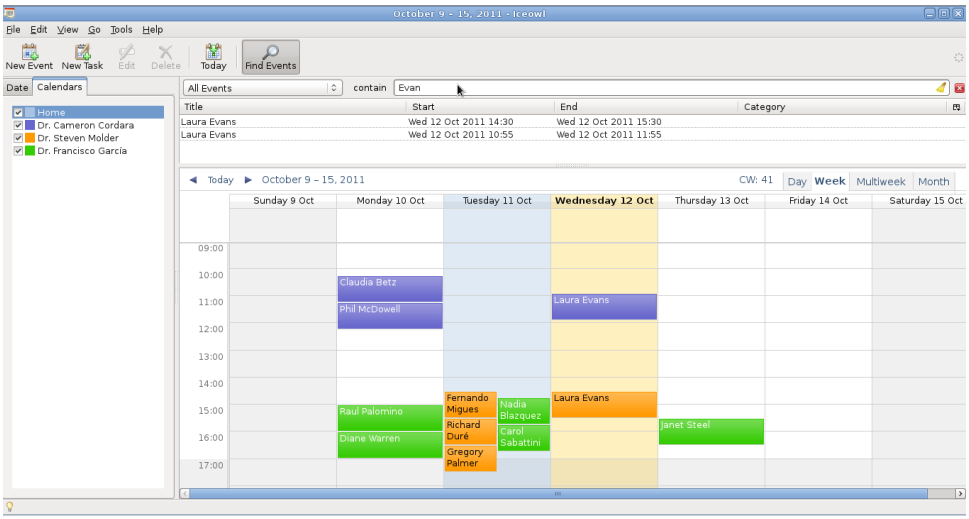


Rysunek 2.4: Demo programu ERPNext.

W opiece zdrowotnej ERPNext jest obecnie wyposażony w zarządzanie rekordów pacjenta, Moduł zarządzania wizytami, zarządzanie rekordów laboratoryjnych, jak również pacjenta i faktur zarządzania. Może to okazać się wystarczające dla wielu zakładów opieki zdrowotnej, które nie wymagają zaawansowanych lub złożonych klinicznych przepływów pracy. Architektura serwerowa jest oparta o Frappe, które jest płatnym rozwiązaniem w związku z czym zaniechano dalszych rozważań nad zastosowaniem ERPNext do projektu.

2.5 GNU Health

GNU Health[6] został wydany wiele lat temu jako w pełni funkcjonalny system EHR, skupiający się nie tylko na przepływie pracy klinicznej, ponieważ zawiera bazę danych chorób, bibliotekę genetyczną, parametry socjoekonomiczne (warunki mieszkaniowe, nadużywanie substancji psychoaktywnych, wykształcenie), standardy chorób i procedur medycznych (ICD-10 / ICD-10-PCS).



Rysunek 2.5: Przykładowy widok programu GNU Health.

Jedną z najciekawszych cech GNU Health jest to, że jest łatwy do zainstalowania, ponieważ został już dodany do repozytoriów wielu dystrybucji Linuksa, więc można go zainstalować bezpośrednio z centrum oprogramowania, a także posiada klientów desktopowych w instalowalnych pakietach dla Windows i Mac OSX. GnuHealth nie działa wyłącznie jako HIS, może pracować jako prosty, jedno-węzłowy system elektronicznego rekordu medycznego EMR. Jego niski próg wejścia oraz podatność na manipulację danymi pozwala wybrać go do pracy w projekcie.

2.6 Oddo

Odoo ERP[7] to darmowe rozwiązanie open source klasy ERP/ CRM wydane po raz pierwszy w 2005 roku, jako kompletny pakiet do zarządzania podstawowymi procesami biznesowymi. Odoo jest rozwiązaniem instalowanym na serwerze i opartym na przeglądarce, ponieważ posiada responsywny design, aby dopasować się i pracować z różnymi rozdzielczościami ekranu i urządzeniami, ma również bardzo potężne wsparcie iOS / Android wdrożenia z udanych historii dla wielu klientów i deweloperów.

Rysunek 2.6: Przykładowy widok programu Odoo ERP.

Większość modułów HIS do Odoo jest płatne, a darmowe funkcjonalności są zbyt mało rozbudowane dla profesjonalnego zastosowania, co doprowadza do rezygnacji z Odoo (podobnie jak z ERPNext).

2.7 Porównanie HISów

Wyżej opisane HISy poddano porównaniu na podstawie następujących elementów:

- nazwa HISu,
- ilość dostępnych danych,
- możliwość eksportu,
- koszt,
- interfejs.

Tabela 2.1. Porównanie analizowanych HISów.

Porównanie HISów				
Nazwa HISu	Ilość dostępnych danych	Możliwość eksportu	Koszt	Interfejs
Bahmni	mało	CSV, Excel, HTML	darmowy	nowoczesny, prosty w obsłudze
OpenHospital	możliwość generacji	Excel	darmowy	archaiczny
HospitalRun	mało	pliki TypeScript	darmowy	nowoczesny
ERPNext	dużo	CSV, Excel	płatny Framework Frappe	nowoczesny, prosty w obsłudze
GNU Health	możliwość generacji	CSV	darmowy	klasyczny
Oddo	dużo	CSV	płatne moduły	prosty

Z uwagi na studencki charakter zespół był zmuszony odrzucić wszystkie oprogramownia zawierające płatne elementy lub licencje. W celu uniknięcia dodatkowego nakładu pracy wynikającego z przekształceń bazy danych z relacyjnej na nierelacyjną (w przypadku HospitalRun), a także egzotycznego interfejsu (w przypadku Bahmni) do wyboru zostały dwa HISy: OpenHospital oraz GNU Health. Zespół ostatecznie zdecydował się na wybór OpenHospital.

Rozdział 3

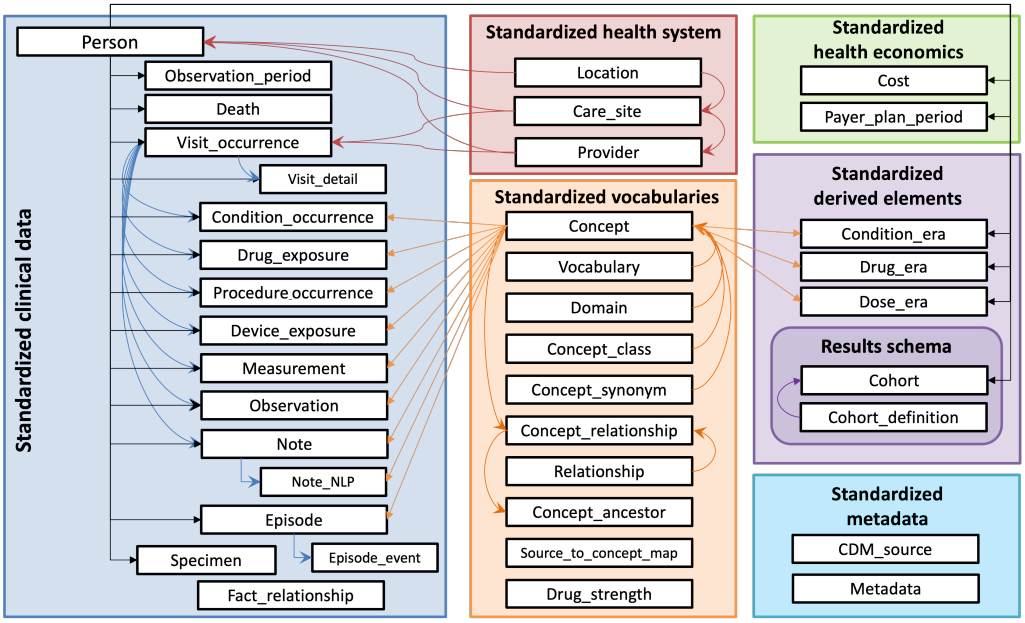
Standard OMOP

W zależności od podstawowych potrzeb żadna z obserwacyjnych baz danych nie rejestruje wszystkich zdarzeń klinicznych równie dobrze. Dlatego też wyniki badań muszą być uzyskane z wielu różnych źródeł danych, a następnie porównane i zestawione ze sobą, aby zrozumieć wpływ potencjalnego błędu przechwytywania danych. Ponadto, aby wyciągnąć wnioski o odpowiedniej mocy statystycznej, potrzebujemy dużej liczby obserwowanych pacjentów. Wyjaśnia to potrzebę jednoczesnej oceny i analizy wielu źródeł danych. Aby to zrobić, dane muszą być zharmonizowane w ramach wspólnego standardu. Ponadto dane pacjentów wymagają wysokiego poziomu ochrony. Pobieranie danych do celów analizy, jak to się robi tradycyjnie, wymaga rygorystycznych umów dotyczących wykorzystania danych i złożonej kontroli dostępu. Wspólny standard danych może złagodzić tę potrzebę, pomijając etap ekstrakcji i umożliwiając wykonywanie standardowych analiz na danych w ich naturalnym środowisku - to analiza przychodzi do danych, a nie dane do analizy. Standard ten zapewnia Wspólny Model Danych (CDM) OMOP.

OMOP[8] (Observational Medical Outcomes Partnership) - ustandaryzowany schemat danych dla systemów szpitalnych. Wspólny Model Danych OMOP umożliwia systematyczną analizę rozbieżnych obserwacyjnych baz danych. Koncepcja tego podejścia polega na przekształceniu danych zawartych w tych bazach do wspólnego formatu (model danych) oraz wspólnej reprezentacji (terminologie, słowniki, schematy kodowania), a następnie przeprowadzaniu systematycznych analiz z wykorzystaniem biblioteki standardowych procedur analitycznych, które zostały napisane w oparciu o wspólny format.

Głównym komponentem są standaryzowane słowniki OHDSI (pozwalają na korzystanie z danych w różnych klinikach korzystających z OMOP)

Relacyjny projekt niezależny od platformy jest zintegrowany z kontrolowaną gramatyką, określany przez domeny, pacjento - centryczny oraz jednolicie integruje dane z różnych źródeł danych.



Rysunek 3.1: Schemat danych OMOP.

OMOP jest uważany za model "osobocentryczny", co oznacza, że wszystkie tabele zdarzeń klinicznych są powiązane z tabelą PERSON. Wraz z datą lub datą rozpoczęcia pozwala to na uzyskanie podłużnego widoku wszystkich zdarzeń istotnych dla opieki zdrowotnej w podziale na osoby.

CDM jest niezależny od platformy. Typy danych są zdefiniowane ogólnie przy użyciu typów danych ANSI SQL (VARCHAR, INTEGER, FLOAT, DATE, DATETIME, CLOB). Precyzja jest zapewniona tylko dla VARCHAR. Odzwierciedla ona minimalną wymaganą długość łańcucha, ale może być rozszerzona w ramach konkretnej instancji CDM.

Schemat podzielony jest na domeny[9]. Domeny to zdefiniowane w OMOP kategorie jednostek klinicznych, które są zdefiniowane dla każdego pojęcia w słowniku standardowym. Lista wszystkich domen jest przechowywana w tabeli DOMENY. W tabeli CONCEPT pole domain id każdego rekordu pojęcia określa dziedzinę, do której należy dany koncept.

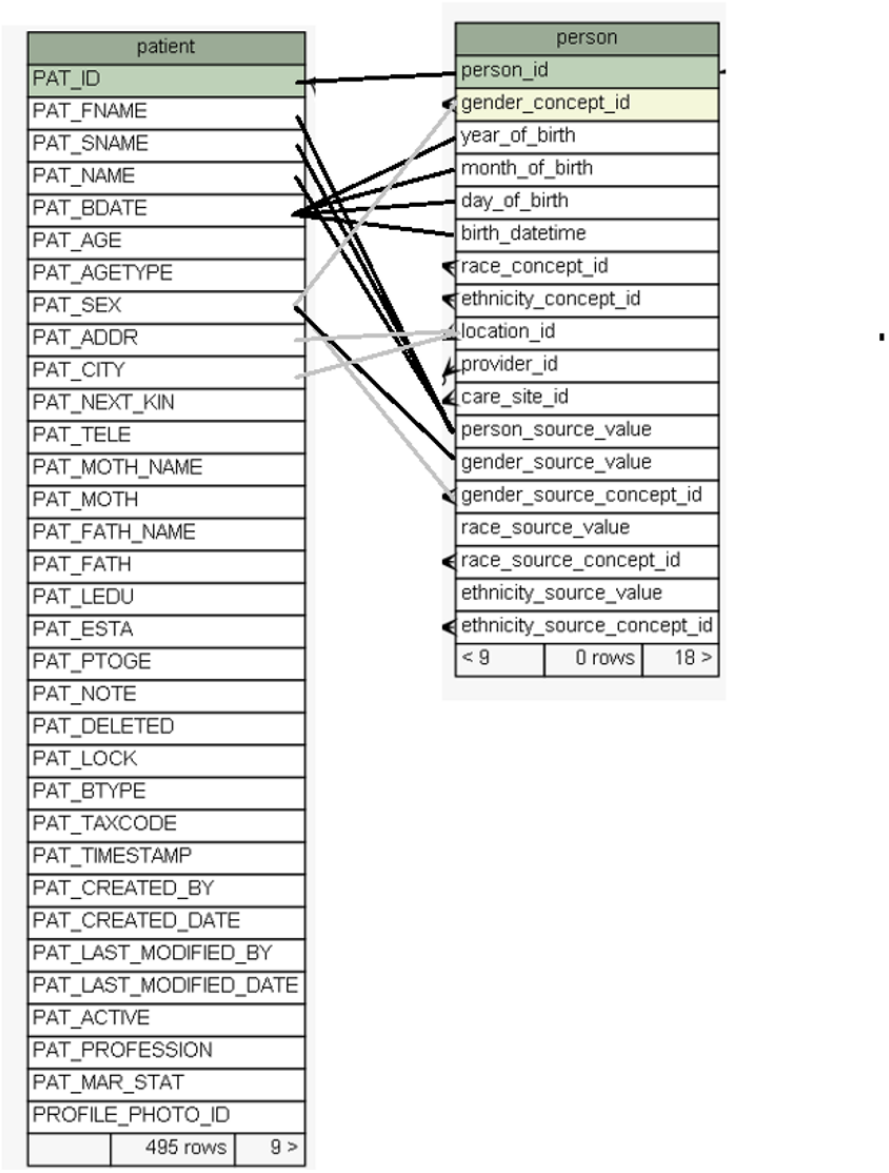
Wyróżniamy następujące domeny w OMOP:

- Condition,
- Drug,
- Device,
- Measurement,
- Procedure,
- Observation.

CDM zawiera 16 tabel zdarzeń klinicznych, 10 tabel słownikowych, 2 tabele metadanych, 4 tabele danych systemu opieki zdrowotnej, 2 tabele danych ekonomiki zdrowia, 3 standardowe elementy pochodne i 2 tabele schematu wyników.

Rozdział 4

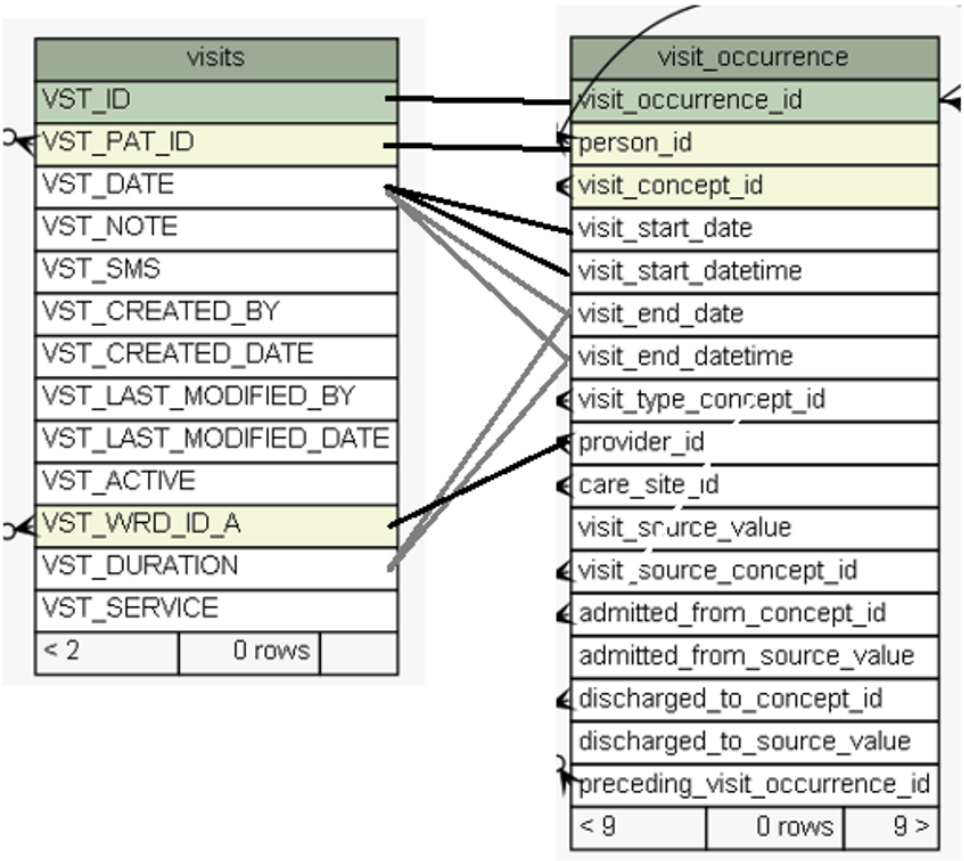
Schemat źródłowy i docelowy



Rysunek 4.1: Odwzorowanie HISowej tabeli Patient do tabeli Person (OMOP).

Podstawową i zarazem najważniejszym odwzorowaniem niezbędnym do zaistnienia korelacji danych jest odwzorowanie tabeli Person (OMOP) z tabelą Patient (HIS). Możliwe jest dopasowanie takich danych jak:

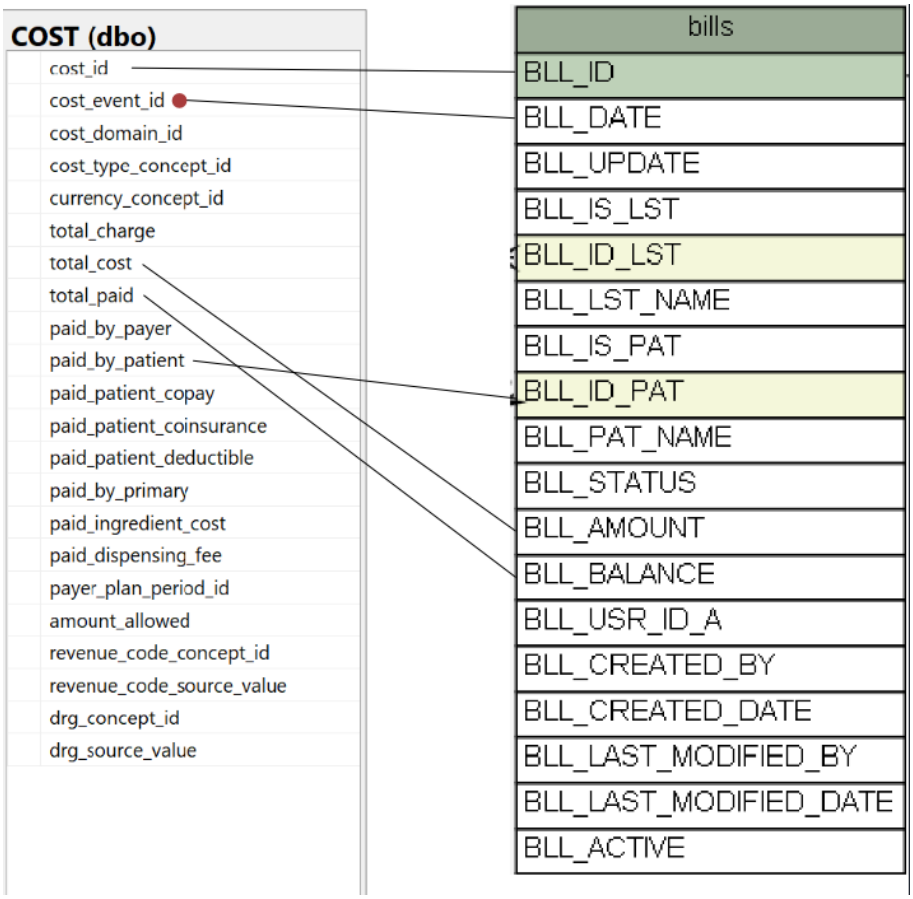
- imię, nazwisko i płeć (HIS) do gender concept (OMOP),
- data urodzenia (HIS) do dnia, miesiąca i roku urodzenia (OMOP),
- adres i miasto (HIS) do lokacji (OMOP).



Rysunek 4.2: Odwzorowanie HISowej tabeli Visits do tabeli Visit details (OMOP).

Możliwe jest także miękkie odwzorowanie tabeli Visits (HIS) z tabelą Visit Occurence (OMOP). Występują następujące powiązania:

- data wizyty (HIS) z datą rozpoczęcia (OMOP),
- czas trwania wizyty (HIS) z deltą daty rozpoczęcia i zakończenia (OMOP),
- wskazanie na Pacjenta (HIS) oraz Osobę (OMOP) w postaci klucza obcego.



Rysunek 4.3: Odwzorowanie HISowej tabeli Bills do tabeli Cost (OMOP).

Kolejnym możliwym odwzorowaniem jest odwzorowanie tabeli Cost (HIS) do tabeli Bills (OMOP). Możliwe odwzorowania:

- data wydarzenia (HIS) do daty rachunku (OMOP),
- koszt całkowity (HIS) do kwoty rachunku (OMOP),
- wskazanie na Pacjenta (HIS) oraz Osobę (OMOP) w postaci klucza obcego.

Rozdział 5

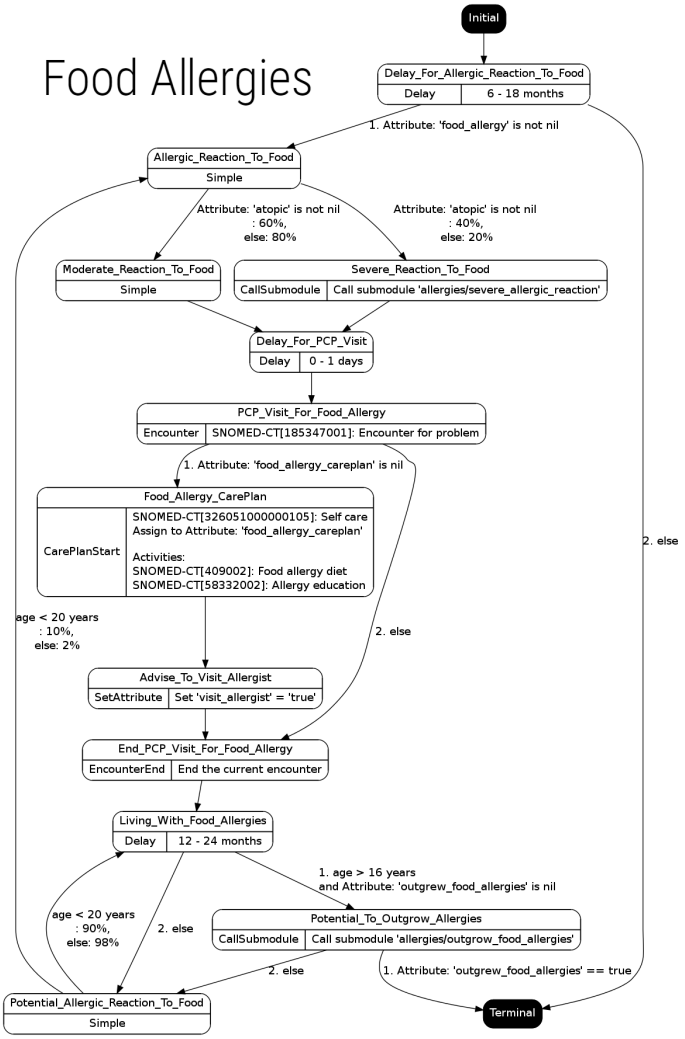
ETL

W celu dokonania odwzorowań opisanych w poprzednim rozdziale niezbędne jest dokonanie przekształceń posługując się jednym z dostępnych narzędzi ETL.

ETL[10] – narzędzia wspomagające proces pozyskania danych dla baz danych, szczególnie dla hurtowni danych. Stanowią nieodzowny element procesów z zakresu Business Intelligence. Zadaniem narzędzi ETL jest: pozyskanie danych ze źródeł zewnętrznych, przekształcenie danych, załadowanie danych do bazy danych.

5.1 Synthea ETL

Synthea[11] to generator syntetycznych pacjentów o otwartym kodzie źródłowym, który modeluje historię medyczną syntetycznych pacjentów. Synthea została uruchomiona w MITRE Corporation jako część Standard Health Record Collaborative (SHRC), projektu open-source, zajmującego się interoperacyjnością danych zdrowotnych. Celem współpracy jest opracowanie Standardowego Rekordu Zdrowotnego (SHR) oraz infrastruktury technologicznej, która napędza innowacje w dziedzinie zdrowia.

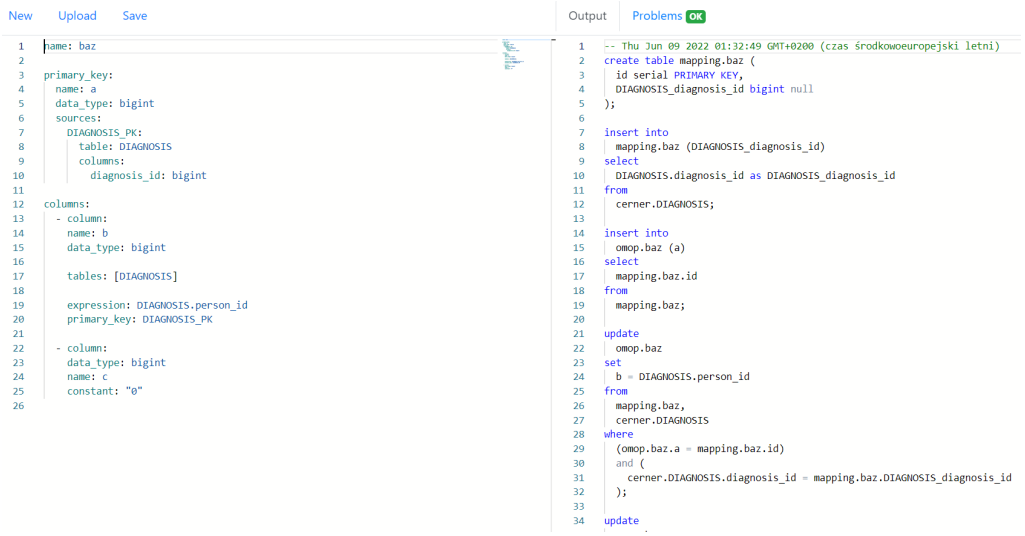


Rysunek 5.1: Przykładowy schemat działania Synthea.

Rozwiązanie niestety jest kompatybilne wyłącznie z plikami CSV tego samego dystrybutora, co jest sprzeczne z założeniem osiągnięcia generycznego rozwiązania.

5.2 OMOP ETL

ETL od Clinical AI[12], który jest oparty na metadanych i generyczny dla różnych źródłowych zbiorów danych. Framework ETL odczytuje logikę mapowania dla tabel OMOP z plików YAML, które organizują fragmenty kodu SQL w pary klucz-wartość definiujące logikę ekstrakcji i transformacji w celu wypełnienia kolumn OMOP.
Wykorzystano język manipulacji danymi (DML) do pisania logiki mapowania ze zbiorów danych dotyczących zdrowia do OMOP, który definiuje operacje mapowania na zasadzie kolumna po kolumnie. Główny potok ETL konwertuje DML w plikach YAML i generuje skrypt ETL.
Struktura DML i operacje odwzorowania zdefiniowane w operacjach kolumna po kolumnie maksymalizują czytelność, refaktoryzację i łatwość konserwacji, minimalizując jednocześnie dług techniczny, oraz standaryzują pisanie operacji ETL w celu odwzorowania na OMOP.

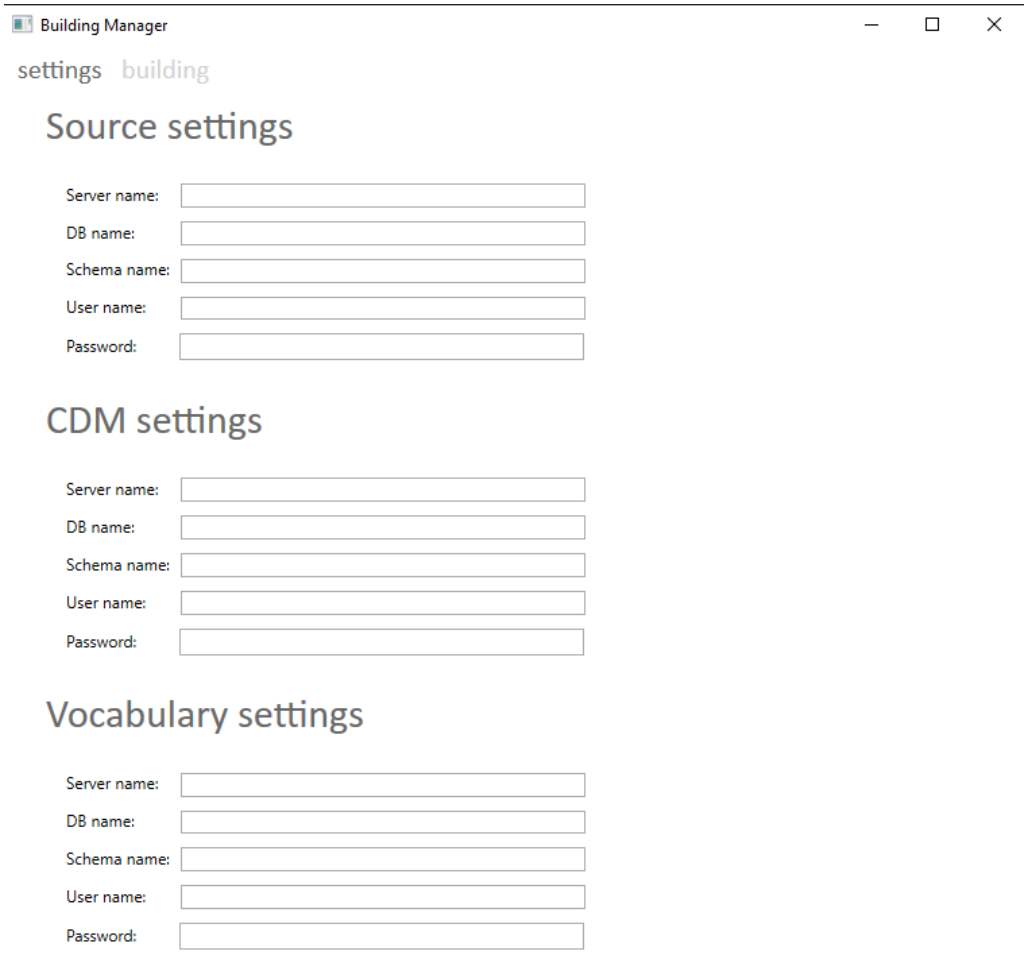


Rysunek 5.2: Aplikacja Webowa OMOP ETL.

Rozwiązanie oferowane jest w wersji webowej, a także istnieje możliwość utworzenia kopii repozytorium z pomocą Dockera bądź Anacondy. Obie wersje oprogramowania nie są intuicyjne w użyciu, a ich podatność na modyfikacje oceniona została na dość niską.

5.3 ETL-CDMBuilder

Narzędzie .Net CDM Builder[13] zostało opracowane przez firmę Janssen Research and Development jako narzędzie do przekształcania obserwacyjnych baz danych w OMOP Common Data Model. Narzędzie to zostało opracowane specjalnie dla środowiska firmy Janssen: MS SQL Server/PostgreSQL/MySQL. Dodatkowo zaprojektowana logika konstruktora opiera się na formacie wejściowym danych źródłowych, które są ładowane w naszym lokalnym środowisku. Narzędzie to nie zostało pierwotnie zaprojektowane do obsługi ETL ogólnego przeznaczenia na różnych platformach, a zastosowanie go w innych systemach wymaga modyfikacji.



Rysunek 5.3: Aplikacja desktopowa CDMBuilder.

Do uruchomienia przekształcenia aplikacja wymaga wskazania trzech źródeł danych:

- bazę danych zawierającą dane źródłowe (pochodzące z HISu),
- bazę danych zawierającą miejsce docelowe i schemat wynikowy (schemat OMOP),
- bazę danych zawierającą mapowania słownikowe.

Do poprawnego działanie wymagane jest wytworzenie mapowania słownikowego (opisane w Rozdziale 6).

Rozdział 6

Proof of concept

Po dokonaniu analiz, wyborze zarówno HISu, jak i ETLów pasujących pod postawione zagadnienie zespół przystąpił do pracy własnej i implementacji składników rozwiązania. Warto zwrócić uwagę na fakt, że wypracowane rozwiązanie musiało mieć charakter generyczny, a wykorzystane oprogramowanie oraz biblioteki powszechnie dostępnie oraz proponowane przez OHDSI.

6.1 Generowanie danych

Wybrany przez zespół HIS posiadał dość niską ilość danych, które mogły podlegać ewentualnym przekształceniom i analizom. W celu pracy na wystarczająco dużym wolumenie danych dokonano wygenerowania syntetycznych danych. Pełen kod wykorzystany do wygenerowania danych znajduje się na poniższych listingach.

Pierwszy skrypt wskazuje na źródła plików sql.

```
#script creazione db
source step_01_create_structure.sql;
source step_02_dump_menu.sql;
source step_03_dump_default_data_en.sql;
source step_04_all_following_steps.sql;
source step_03_dump_vaccine_data_en.sql;
source delete_all_data.sql;
source load_demo_data.sql;
```

Listing 6.1: Wskazanie źródeł plików sql

Drugi skrypt tworzy schematy tabel, zakłada niezbędne blokady oraz wypełnia je danymi. Poniższy listing jest tylko fragmentem pełnego skryptu – jego oryginał ma 1000 linii.

```

--
-- Table structure for table `agetype`
--

DROP TABLE IF EXISTS `agetype`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `agetype` (
  `AT_CODE` varchar(4) NOT NULL DEFAULT '',
  `AT_FROM` int(11) NOT NULL DEFAULT 0,
  `AT_TO` int(11) NOT NULL DEFAULT 0,
  `AT_DESC` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL
    DEFAULT '',
```

```
  `AT_CREATED_BY` varchar(50) DEFAULT NULL,
  `AT_CREATED_DATE` datetime DEFAULT NULL,
  `AT_LAST_MODIFIED_BY` varchar(50) DEFAULT NULL,
  `AT_LAST_MODIFIED_DATE` datetime DEFAULT NULL,
  `AT_ACTIVE` tinyint(1) NOT NULL DEFAULT 1,
  PRIMARY KEY (`AT_CODE`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `agetype`
--

LOCK TABLES `agetype` WRITE;
/*!40000 ALTER TABLE `agetype` DISABLE KEYS */;
INSERT INTO `agetype` VALUES ('d0',0,0,'angal.agetype.newborn',NULL,NULL,NULL,NULL
    ,1);
INSERT INTO `agetype` VALUES ('d1',1,5,'angal.agetype.earlychildhood',NULL,NULL,
    NULL,NULL,1);
INSERT INTO `agetype` VALUES ('d2',6,12,'angal.agetype.latechildhood',NULL,NULL,
    NULL,NULL,1);
INSERT INTO `agetype` VALUES ('d3',13,24,'angal.agetype.adolescents',NULL,NULL,NULL
    ,NULL,1);
INSERT INTO `agetype` VALUES ('d4',25,59,'angal.agetype.adult',NULL,NULL,NULL,NULL
    ,1);
INSERT INTO `agetype` VALUES ('d5',60,99,'angal.agetype.elderly',NULL,NULL,NULL,
    NULL,1);
/*!40000 ALTER TABLE `agetype` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `billitems`
--

DROP TABLE IF EXISTS `billitems`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `billitems` (
  `BLI_ID` int(11) NOT NULL AUTO_INCREMENT,
  `BLI_ID_BILL` int(11) DEFAULT NULL,
  `BLI_IS_PRICE` tinyint(1) NOT NULL,
  `BLI_ID_PRICE` varchar(10) DEFAULT NULL,
  `BLI_ITEM_DESC` varchar(100) DEFAULT NULL,
  `BLI_ITEM_AMOUNT` double NOT NULL,
  `BLI_QTY` int(11) NOT NULL,
  `BLI_CREATED_BY` varchar(50) DEFAULT NULL,
  `BLI_CREATED_DATE` datetime DEFAULT NULL,
  `BLI_LAST_MODIFIED_BY` varchar(50) DEFAULT NULL,
  `BLI_LAST_MODIFIED_DATE` datetime DEFAULT NULL,
  `BLI_ACTIVE` tinyint(1) NOT NULL DEFAULT 1,
  PRIMARY KEY (`BLI_ID`),
  KEY `FK_BILLITEMS_BILLS` (`BLI_ID_BILL`),
  CONSTRAINT `FK_BILLITEMS_BILLS` FOREIGN KEY (`BLI_ID_BILL`) REFERENCES `bills` (`
    BLL_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=159 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `billitems`
```

```
--

LOCK TABLES `billitems` WRITE;
/*!40000 ALTER TABLE `billitems` DISABLE KEYS */;
INSERT INTO `billitems` VALUES (50,15,1,'EXAURI','URINALYSIS',10,1,'admin','
    2020-11-21 03:31:42','admin','2020-11-23 15:49:48',1);
INSERT INTO `billitems` VALUES (51,15,1,'EXA03.02','3.2 Blood Slide (OTHERS, E.G.
    TRIUPHANOSOMIAS, MICRIFILARIA, LEISHMANIA, BORRELIA)',12,1,'admin','2020-11-21
    03:31:42','admin','2020-11-23 15:49:48',1);
INSERT INTO `billitems` VALUES (52,16,1,'MED67','Insulin Mixtard 30/70 100IU/ml 5
    x3ml catridges',30,10,'admin','2020-11-21 03:37:53','admin','2020-11-23
    15:49:56',1);
INSERT INTO `billitems` VALUES (53,16,1,'EXA07.02','7.2 SUGAR',8,1,'admin','
    2020-11-21 03:37:53','admin','2020-11-23 15:49:56',1);
INSERT INTO `billitems` VALUES (54,17,1,'MED197','Cloramphenicol 1% Eye Ointment
    3.5g',5,2,'admin','2020-11-21 21:59:57','admin','2020-11-23 15:49:52',1);
INSERT INTO `billitems` VALUES (55,17,1,'EXA07.01','7.1 PROTEIN',8,1,'admin','
    2020-11-21 21:59:57','admin','2020-11-23 15:49:52',1);
INSERT INTO `billitems` VALUES (56,18,1,'MED110','Atenolol 100mg Tab',22,1,'admin',
    '2020-11-21 22:01:55','admin','2020-11-23 15:49:32',1);
INSERT INTO `billitems` VALUES (57,18,1,'MED21','Potassium Oxalate',22,2,'admin','
    2020-11-21 22:01:55','admin','2020-11-23 15:49:32',1);
INSERT INTO `billitems` VALUES (58,18,1,'OTH1','Amount per day',0,2,'admin','
    2020-11-21 22:01:55','admin','2020-11-21 22:01:55',1);
INSERT INTO `billitems` VALUES (59,18,1,'OPE38','Mechanical',80,1,'admin','
    2020-11-21 22:01:55','admin','2020-11-23 15:49:32',1);
INSERT INTO `billitems` VALUES (60,19,1,'OPE37','Intestinal obstruction',85,1,'
    admin','2020-11-21 22:02:53','admin','2020-11-23 15:52:06',1);
INSERT INTO `billitems` VALUES (61,19,1,'EXA03.022','3.22 MICROFILARIA',12,1,'admin
    ','2020-11-21 22:02:53','admin','2020-11-23 15:52:06',1);
INSERT INTO `billitems` VALUES (62,19,1,'MED28','Trisodium Citrate',38,1,'admin','
    2020-11-21 22:02:53','admin','2020-11-23 15:52:06',1);
INSERT INTO `billitems` VALUES (63,20,1,'MED219','Hydrocortisone 1% Ointment 15g'
    ,8,2,'admin','2020-11-21 22:05:25','admin','2020-11-23 15:49:20',1);
INSERT INTO `billitems` VALUES (64,20,1,'EXA03.01','3.1 Blood Slide (Malaria)'
    ,10,1,'admin','2020-11-21 22:05:25','admin','2020-11-23 15:49:20',1);
INSERT INTO `billitems` VALUES (65,21,1,'MED2','Acetic Acid Glacial 1 ltr',18,1,'
    admin','2020-11-21 22:08:16','admin','2020-11-23 15:49:16',1);
```

Listing 6.2: Tworzenie schematu i generowanie danych

6.2 Utworzenie schematu OMOP

Po przygotowaniu danych źródłowych zespół przystąpił do przygotowania schematu OMOP będącym schematem docelowym, do którego dane są migrowane. Zostały wykorzystane do tego DDLe udostępnione przez OHDSI. Poniższy listing,podobnie jak poprzedni, jest tylko fragmentem pełnego skryptu w trosce o zwięzłą treść tej dokumentacji.

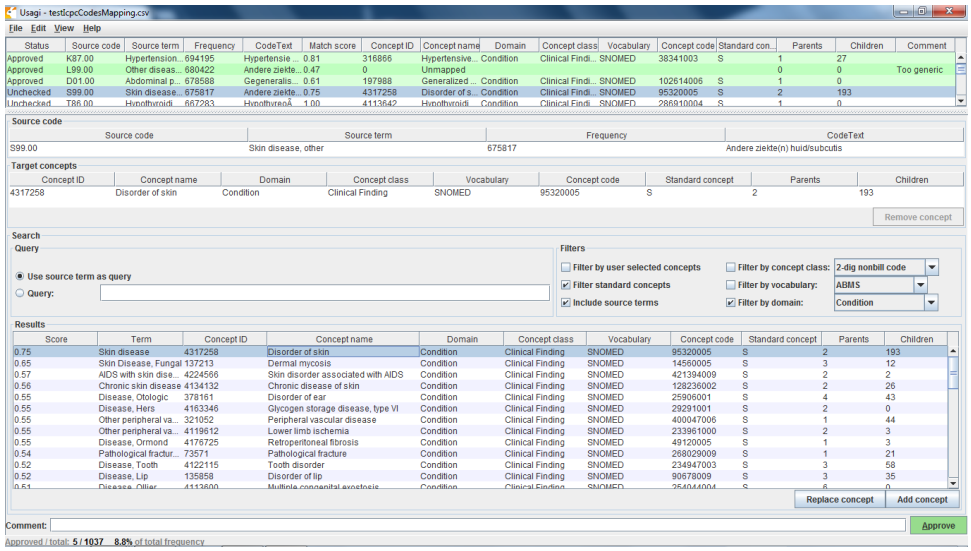
```
--HINT DISTRIBUTE ON KEY (person_id)
CREATE TABLE @cdmDatabaseSchema.PERSON (
      person_id integer NOT NULL,
      gender_concept_id integer NOT NULL,
      year_of_birth integer NOT NULL,
      month_of_birth integer NULL,
      day_of_birth integer NULL,
      birth_datetime datetime NULL,
      race_concept_id integer NOT NULL,
      ethnicity_concept_id integer NOT NULL,
```

```
11      location_id integer NULL,
12      provider_id integer NULL,
13      care_site_id integer NULL,
14      person_source_value varchar(50) NULL,
15      gender_source_value varchar(50) NULL,
16      gender_source_concept_id integer NULL,
17      race_source_value varchar(50) NULL,
18      race_source_concept_id integer NULL,
19      ethnicity_source_value varchar(50) NULL,
20      ethnicity_source_concept_id integer NULL );
21
22
23 --HINT DISTRIBUTE ON KEY (person_id)
24 CREATE TABLE @cdmDatabaseSchema.VISIT_DETAIL (
25      visit_detail_id integer NOT NULL,
26      person_id integer NOT NULL,
27      visit_detail_concept_id integer NOT NULL,
28      visit_detail_start_date date NOT NULL,
29      visit_detail_start_datetime datetime NULL,
30      visit_detail_end_date date NOT NULL,
31      visit_detail_end_datetime datetime NULL,
32      visit_detail_type_concept_id integer NOT NULL,
33      provider_id integer NULL,
34      care_site_id integer NULL,
35      visit_detail_source_value varchar(50) NULL,
36      visit_detail_source_concept_id Integer NULL,
37      admitted_from_concept_id Integer NULL,
38      admitted_from_source_value varchar(50) NULL,
39      discharged_to_source_value varchar(50) NULL,
40      discharged_to_concept_id integer NULL,
41      preceding_visit_detail_id integer NULL,
42      parent_visit_detail_id integer NULL,
43      visit_occurrence_id integer NOT NULL );
```

Listing 6.3: DDL twworzący schemat danych OMOP.

6.3 Mapowanie słownikowe

CDMBuilder oprócz bazy danych wyjściowej oraz docelowej wymaga także bazy danych z mapowaniem słownikowym. Standardowy słownik o nazwie Athena jest podstawowym narzędziem, początkowo opracowanym przez OHDSI w OMOP, które umożliwia tworzenie przejrzystych i spójnych treści w różnych bazach danych obserwacyjnych i służy wsparciu społeczności badawczej OHDSI w prowadzeniu wydajnych i powtarzalnych badań obserwacyjnych. Zespół podjął próbę wykorzystania do tego celu oprogramowania Usagi.
Usagi to oprogramowanie OHDSI wspomagające proces mapowania kodów z systemu źródłowego na terminologie, najlepiej standardowe, przechowywane w słowniku OMOP.



Rysunek 6.1: Przykładowy widok programu Usagi.

Mimo wsparcia OHDSI wszystkich trzech składników rozwiązania (CDMBuilder, Usagi, Athena) wytworzenie słownika do CDMBuildera okazało się niemożliwe ze względu na niezgodność schematów tabel (brakujące atrybuty). Próby naprawieniaśłownika nie przyniosły oczekiwanego rezultatu.

Rozdział 7

Podsumowanie

Celem pracy było zaprojektowanie i zaimplementowanie generycznego rozwiązania pozwalającego na transformację danych systemu szpitalnego do schematu zgodnego ze standardem OMOP.

Pierwszym etapem pracy była analiza oraz wybór HISu wraz z generacją przykładowych danych do niego. Kolejnym etapem była analiza różnicy pomiędzy schematem danych HISu oraz schematem standardu OMOP i wypracowaniem powiązań atrybutów. Następnym etapem był przegląd i przygotowanie środowiska pracy ETL.

Mimo zgodności składników rozwiązania ostatniego etapu (OHDSI jako źródło) nie było możliwe utworzenie mapowania słownikowego, przez co odwzorowanie nie było możliwe. Samo stowarzyszenie OHDSI mimo sporego rozmiaru i profesjonalizmu wydaje się mniej aktywne w ostatnich latach, a sporo wątków na forum pozostało otwarte.

W związku z powyższym zespół doszedł do wniosku, iż migracja danych z HISu do standardu OMOP w sposób w pełni generyczny nie jest możliwe, a implementacja dedykowanego rozwiązania do wybranego przez zespół oprogramowania nie jest istotą projektu. Wynika to najprawdopodobniej z kwestii biznesowych – żadnemu twórcy komercyjnego medycznego oprogramowania nie zależy na migracji oraz standaryzacji, gdyż w ten sposób odcięliby się od źródła dochodu i umożliwiliby na niebezpieczny dla nich rozwój rozwiązań open-source. Niestety cierpi na tym nauka zarówno pod kątem informatycznym, jak i medycznym – wiele zależności medycznych wśród pacjentów tylko czeka na wykrycie, które przy obecnej hermetyzacji każdego z komercyjnych systemów medycznych nie jest możliwe.

Literatura

[1] C. Brook, "What is a health information system." [on-line] https://digitalguardian.com/blog/what-health-information-system. [Data uzyskania dostępu: 21 marca 2022].

[2] B. Coalition, "Bahmni." [on-line] https://www.bahmni.org/. [Data uzyskania dostępu: 30 marca 2022].

[3] I. senza Frontiere, "Open hospital." [on-line] https://www.open-hospital.org/about-open-hospital/. [Data uzyskania dostępu: 30 marca 2022].

[4] O. Foundation, "Hospitalrun." [on-line] https://hospitalrun.io/. [Data uzyskania dostępu: 30 marca 2022].

[5] F. T. P. Ltd., "Erpnext." [on-line] https://erpnext.com/. [Data uzyskania dostępu: 31 marca 2022].

[6] G. Project, "Gnu health." [on-line] https://www.gnuhealth.org/about-us.html. [Data uzyskania dostępu: 31 marca 2022].

[7] O. S.A., "Odoo." [on-line] https://www.odoo.com/. [Data uzyskania dostępu: 31 marca 2022].

[8] OHDSI., "Omop common data model." [on-line] https://www.ohdsi.org/data-standardization/the-common-data-model/. [Data uzyskania dostępu: 10 kwietnia 2022].

[9] ohdsi wiki contributors, "Domains." [on-line] https://www.ohdsi.org/web/wiki/doku.php?id=documentation:vocabulary:domains/. [Data uzyskania dostępu: 13 kwietnia 2022].

[10] W. contributors, "Extract, transform, load." [on-line] https://en.wikipedia.org/wiki/Extract,_transform,_load. [Data uzyskania dostępu: 26 kwietnia 2022].

[11] E.-S. contributors, "Utility to load synthea csv data to omop cdm." [on-line] https://ohdsi.github.io/ETL-Synthea/. [Data uzyskania dostępu: 27 kwietnia 2022].

[12] clinical ai, "Extract, transform, load framework for the conversion of health databases to omop." [on-line] https://github.com/clinical-ai/omop-etl. [Data uzyskania dostępu: 27 kwietnia 2022].

[13] J. Research and Development, "Etl-cdmbuilder." [on-line] https://github.com/OHDSI/ETL-CDMBuilder. [Data uzyskania dostępu: 27 kwietnia 2022].

Instytut Informatyki, Wydział Informatyki i Telekomunikacji
Politechnika Poznańska

Skład przy użyciu systemu LaTeX na platformie Overleaf.