

Common Data Model for Healthcare data

Umair M. Khan, Huzaifa Kothari, Aditya Kuchekar, Prof. Reeta Koshy

Department of Computer Engineering

Bharatiya Vidya Bhavan's Sardar Patel Institute of Technology

Munshi Nagar, Andheri-West, Mumbai, India

umairmujtaba.khan@spit.ac.in, huzaifa.kothari@spit.ac.in,

aditya.kuchekar@spit.ac.in, reeta_koshy@spit.ac.in

Abstract—There is a huge volume of data made openly available to the masses on Government data sites such as the www.data.gov.in. However, it is very difficult to make sense between data from two different data sets or different domains. As a result a lot of time is spent in just collecting and linking this data from various sources before it can be put to use. A potential solution to this problem is using a Common Data Model. A Common Data Model is a data silo or warehouse which can accommodate any data from any source in one pre-designed format. In this paper we discuss an ETL process that allows users to remodel and store data, and a proposed schema for a Common Data Model which will house this transformed data under one roof. We also perform our own analysis on the proposed data model for a particular use-case; state-wise statistics of data related to pregnancy in India. Thus we prove that the common data model is able to integrate data from different sources and facilitate cross domain analysis and linking of data.

Index Terms—Common Data Model, ETL Extract Transform Load, Data Warehouse, Columnar database, Mapping file.

I. INTRODUCTION

Data is one of the most valuable resources of the digital age we live in. In the last few decades, an enormous amount of data has been available globally, owing to the internet. A lot of inferences and information can be obtained from this data via analysis, and hence the field of data analytics is rapidly blooming. However, before data can be analyzed, it has to be collected from the internet, cleaned, and brought into a definite format on which algorithms can be performed.

The challenge is that even though data is available, it is scattered and segregated by domain. Also, datasets from different sectors may differ in structure. The dataset for literacy rate in mumbai may differ in number and type of fields from dataset for crime rate in Mumbai, but by connecting these two datasets, inferences can be made as to how literacy rate can affect crime rate. However, for analysis algorithms to be carried out, we need to integrate both datasets into one common data set. Around 70% of time and effort is spent by data analysts in Wrangling data[6], i.e bringing this data together from different sources, and then connecting it, before analysis can be carried out. The official Indian government repository for data; www.data.gov.in has a huge amount of data available. However, this data is segregated by departments, and

currently no solution exists to house all this interdepartmental data in one common data silo or model.

In this paper we have proposed a schema for a Common Data Model which stores data in a fixed generic format as opposed to a specific format. As a result, we can integrate any kind of data into this CDM. Figure 1 shows how a CDM works. Since the analysis tools are highly dependant on the data they are operating, the CDM must be independent of the structure of its source data files. During the course of this paper we discuss the proposed schema in detail, and how it can be used to accommodate data from different datasets obtained from data.gov.in. We also explain the ETL process and tools used to load data from a source csv file to our CDM. Lastly, we perform an analysis on data in the CDM to obtain state-wise statistics of data from different domains related to pregnancy in India. Thus we prove that our CDM can be used to integrate varying data from data.gov.in, and that some analysis can be performed on the transformed data.

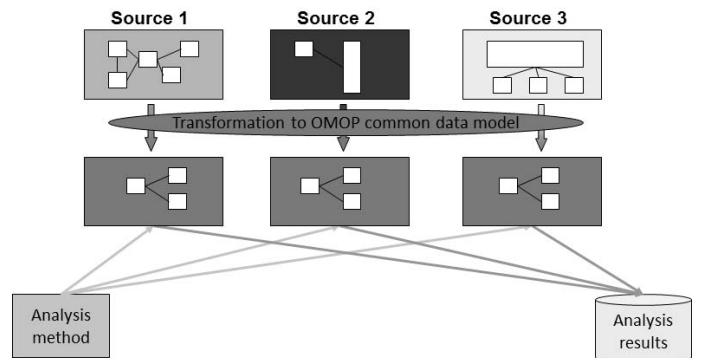


Fig. 1: Common Data Model

II. RELATED WORK

The problem of integrating multiple data sources into one data warehouse has been a long existing one. There has been number of advancements in research in this particular field of building common data models to store data about a particular department or institution, but belonging to multiple sectors. For instance, the Integrated Aircraft Health Management

(IAHM) program has successfully developed a common data model specifically for Aircraft Vehicle Health Management[1]. In their proposed system, they have collected and integrated flight data, maintenance log information and test stand runs into their designed CDM to support the development of the wide variety of aircraft health management analysis tools.

The Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) has been adopted by the Health Data Sciences and Informatics (OHDSI) collaborative[11], a multi-stakeholder, inter-disciplinary effort to create open-source solutions that bring out the value of observational health data through large-scale analytics. The purpose of this CDM is to standardize the format and content of observational data so that common software applications, tools and methods can easily be applied across datasets from multiple healthcare organizations.

The Research on ETL in Land and Resources Star Schema Data Warehouse by Qin, Hanlin, Jin and Xianzhen[9] elaborately explains ETL concepts and their application in Data Wrangling. Our CDM design is inspired greatly by the OHDSI OMOP CDM model. Keeping it as a reference and utilizing the concept of ETL and column-oriented database systems[4], we have designed a more generic CDM that can be used to accommodate data from indian government sites belonging to any domain.

III. PROPOSED MODEL

The entire proposed system is shown in the figure 2 . It is divided into two parts; the CDM that will store multiple data files uploaded by the user and the ETL system designed to transform and load source files into the CDM. In this section we shall discuss each of these subsections in detail.

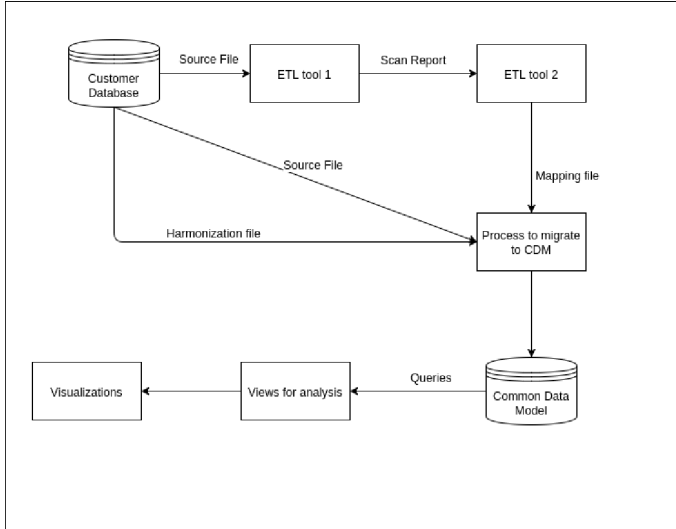


Fig. 2: CDM schema

A. Proposed CDM Schema

Figure 3 shows the CDM Schema we have designed. It consists of 5 tables: **concept**, **domain**, **entity**, **measurement**

and **source**. Each of them is designed for a distinct purpose, but all of them share majority of their data with some or the other table. However, we have reduced the interdependence and interconnectivity of these tables by constructing a master table, i.e concept table, and connecting all tables to this master table, thus giving a topology similar to a star schema. We will now discuss each of these tables in detail.

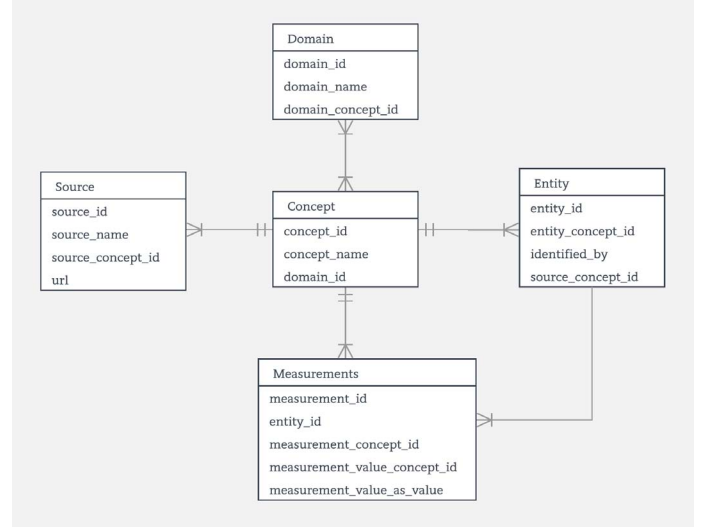


Fig. 3: CDM schema

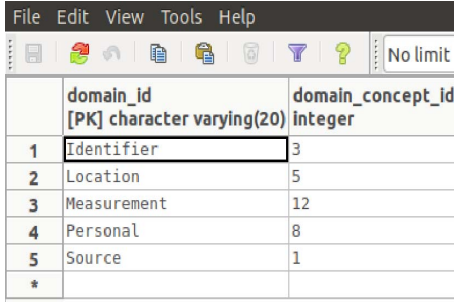
- **Concept:** Figure 4 illustrates the concept table with some data. This table serves as a master table, i.e it stores all data items that are currently present anywhere in the CDM. All entities and measurements become concepts in the concept table. Each concept contains:
 - A Unique identifier **Concept_id** that is used to refer the concept in other tables.
 - A unique **concept_name** that tells us what the concept is.
 - A **domain_id** that tells us what domain the concept falls under. For eg: State comes under domain location.

	concept_name character varying(255)	domain_id character varying(20)	concept_id [PK] bigserial
1	Source	Metadata	1
2	ahs-mort-rajasthan-jhunjhun.csv	Source	2
3	Identifier	Metadata	3
4	id	Identifier	4
5	Location	Metadata	5
6	state	Location	6
7	district	Location	7
8	Personal	Metadata	8
9	year of death	Personal	9
10	168441	Entity	10
11	UTTARAKHAND	Measurement	11
12	Measurement	Metadata	12
13	NAINITAL	Measurement	13

Fig. 4: Concept table

- **Domain:** Figure 5 illustrates the domain table with some data. The purpose of this table is to store all the domains that the data items in the CDM can fall under. Each domain is defined by:

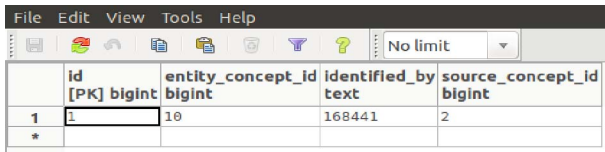
- a unique **domain_id** or name.
- Its reference in the concept table, **domain_concept_id**.



	domain_id	domain_concept_id
	[PK] character varying(20)	integer
1	Identifier	3
2	Location	5
3	Measurement	12
4	Personal	8
5	Source	1
*		

Fig. 5: Domain table

- **Entity:** Figure 6 illustrates the Entity table with some data. In the CDM, each record of the source table is treated as an entity, identified by some value (usually its primary key). All other fields in the source table are treated as measurements to this entity. Each entity has:
 - Its own unique **id**
 - A concept table reference **entity_concept_id**
 - An **identified_by** field which tells us the name of this entity.
 - **Source_concept_id** which tells us which source this entity belongs to.



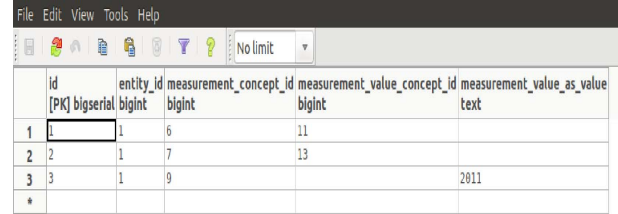
	id	entity_concept_id	identified_by	source_concept_id
	[PK] bigint	bigint	text	bigint
1	1	10	168441	2
*				

Fig. 6: Entity table

- **Measurements:** Figure 7 illustrates the measurement table with some data. This is where the actual data in the source table records is stored. Each record is an entity and each field is a measurement on that entity. Suppose our measurement is capital of maharashtra is mumbai. This table has the following columns to store this measurement:
 - Its own unique **id**
 - **Entity_id** which tells us what entity is the measurement on. In this case: Maharashtra.
 - **Measurement_concept_id** which tells us what is the measurement that is being performed on the entity. It refers to a concept which tells us what the measurement is. Here the measurement is capital.
 - **Measurement_value_concept_id** which tells us what is the value of that measurement. It refers to a concept that is a value of some sort. Here we store the concept id for concept named mumbai
 - **Measurement_value_as_value** which tells us the value of the measurement if it is not a concept.

Usually integral or continuous values like age values are mapped to this column.

Thus this table confines any source table to fit into these 5 columns. It converts a horizontal source table with uncertain structure to a vertical table with a fix structure[4].



	id	entity_id	measurement_concept_id	measurement_value_concept_id	measurement_value_as_value
	[PK] bigserial	bigint	bigint	bigint	text
1	1	1	6	11	
2	2	1	7	13	
3	3	1	9		2011
*					

Fig. 7: Measurements table

- **Source:** Figure 8 illustrates the source table with some data. It stores all the source data file locations, so that we know where the data in the CDM has been taken from. Data that is stored about each source file is:
 - **Name** of the source file.
 - **Source_concept_id** that refers to a concept for the source table.
 - **Url** if available, that tells us where the source file was extracted from.
 - **Source_concept_id** which tells us which source this entity belongs to.



	id	source_name	source_concept_id	url
	[PK] bigserial	text	bigint	text
1	1	ahs-mort-rajasthan-jhunjunun.csv	2	
*				

Fig. 8: Source table

B. Proposed ETL System

The ETL system is designed to facilitate the user to load the source file data into the CDM. It consists of two tools, ETL tool 1: WhiteRabbit and ETL tool 2: RabbitInaHat. Both these tools have been developed by OHDSI[11] to help map data to OMOP CDM. We will now discuss the working of both these tools.

- **WhiteRabbit:** This tool, as illustrated in fig 9 is used to perform a scan of the source data, providing detailed information on the tables, fields, and values that appear in a field. This scan will generate a report that can be used by the RabbitInaHat tool. The scan report excel document will be created in the working folder location selected earlier. The document will have multiple tabs, one as an Overview and then one tab for each database table or delimited text files selected for the scan. The Overview tab will tell you about each table selected, what the columns in each table are, the data type of the

columns, the amount of data within the table, the number of rows scanned, and the fraction of data empty.

For a tab that describes a single table, the columns names from the source table (or delimited text file) will be across the columns of the Excel tab. Each source table column will generate two columns in the Excel. One column will list all distinct values. Next to each distinct value will be a second column that contains the frequency, or the number of times that value occurs in the data.

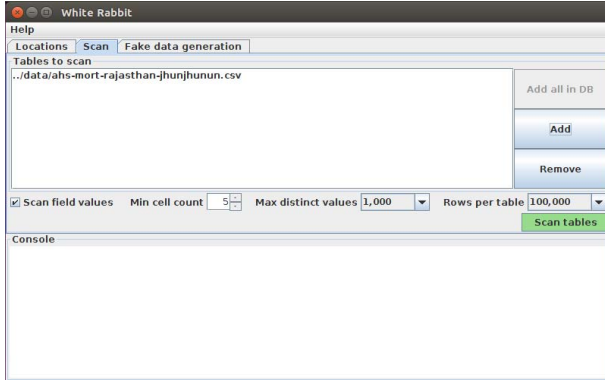


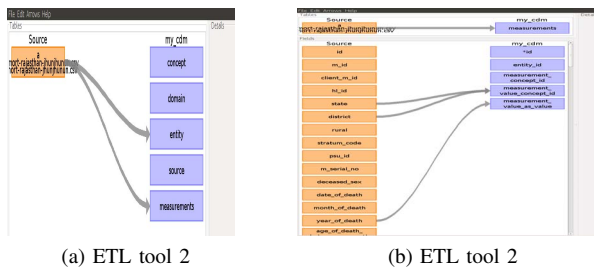
Fig. 9: White rabbit tool

(a) Scan File 1

(b) Scan File 2

Fig. 10: Scan Files

- **RabbitInaHat:** Rabbit-In-a-Hat, as illustrated in fig 9 uses the scan report obtained from WhiteRabbit, and provides a graphical user interface to allow a user to connect source data to tables and columns within the CDM. In our system we have loaded our CDM schema into Rabbit-In-a-Hat as the destination database. The output of this tool is a mapping file that is used by the system to insert values from the source file into the CDM.



(a) ETL tool 2

(b) ETL tool 2

Fig. 11: RabbitInaHat tool

The entire system is built like a partnership, similar to OMOP. Only approved collaborators who have knowledge about the CDM and its working can upload data to the CDM. However, the CDM will be openly available for reading and querying, so that anyone can perform analysis on the available data.

IV. IMPLEMENTATION AND RESULTS

This section demonstrates an example of how a record from a source data file (as illustrated in fig 9, fig 12) is mapped and stored in the CDM. Collaborators have to use our webapp interface and go through a series of steps in order to transform and load data from source file to CDM. We shall discuss each step in this example.

	A	B	C	D	E	F	G	H
1	id	m_id	client_m_id	hl_id	state	district	rural	stratum_code
2	163905	182	NA	NA	RAJASTHAN	JHUNJHUNUN	Urban	Urban
3	163897	58	NA	NA	RAJASTHAN	JHUNJHUNUN	Urban	Urban
4	163889	59	NA	NA	RAJASTHAN	JHUNJHUNUN	Urban	Urban
5	163881	141	NA	NA	RAJASTHAN	JHUNJHUNUN	Urban	Urban
6	163873	62	NA	NA	RAJASTHAN	JHUNJHUNUN	Urban	Urban

Fig. 12: Source File

- Generate Scan Report using ETL tool 1: The user uses the first ETL tool to scan the source file and generate a scan report (as figures 10.a and 10.b)
- Using ETL tool 2 and scan report generated to map the source table and columns to destination tables and columns respectively (as illustrated in figures 10 and 11) . This tool generates a mapping file as shown in fig 13.

Fig. 13: Mapping file

- Execute the insertion script via the WebApp using mapping file, source file and domain information. User has to provide which domain each source column in the map file falls under. The data now gets mapped row by row to the CDM. To see how the data is stored in the CDM, refer to figures 4 ,5 ,6 ,7.

AS proof of concept, we have carried out analysis on a particular use-case; state-wise statistical analysis of pregnancy issues in India . We have inserted data files about mortality rate, health care sites and literacy rate obtained from data.gov.in into the CDM, and designed queries to obtain views that gives us statistical data about each state in India. The final output is shown in fig 14. In addition to the implementation and validation of our CDM, we have also looked into its advantages and disadvantages against traditional data storage models. Table I illustrates a comparative study between the two.

	TRADITIONAL DATA STORAGE STRUCTURES	CDM
Flexibility	LOW	VERY HIGH
Schema Complexity	LOW	HIGH
Dependency on source data structure	HIGHLY DEPENDENT	INDEPENDENT
Needs harmonization between synonymous values	NO	YES
Cost of querying 1 logical row	LOW	VERY HIGH
Cost due to changes in source data structures	VERY COSTLY	INCURS NO COST
Utility	FOR DATA SETS THAT DO NOT CHANGE FREQUENTLY OVER TIME	FOR FREQUENTLY UPDATING DATASETS OR CROSS DOMAIN DATASETS

TABLE I: COMPARISON TABLE

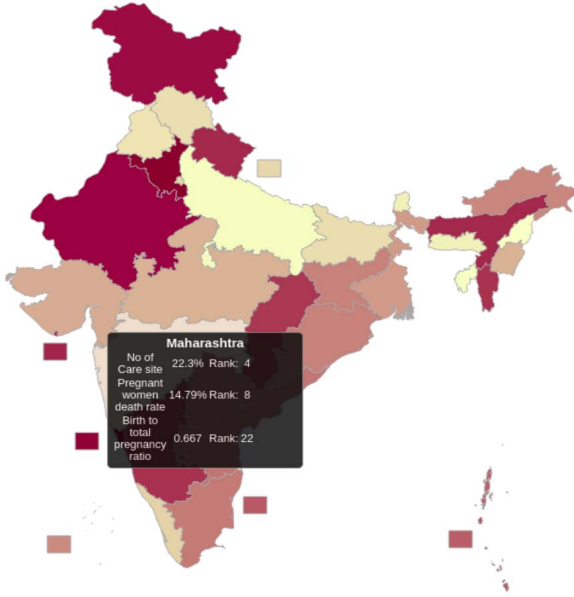


Fig. 14: Final Output

V. CONCLUSION AND FUTURE SCOPE

In this paper we identify the issues regarding the ease of usability of the Indian government data, despite it being openly available on www.data.gov.in. We address the challenge of being able to integrate data from differently structured datasets and store them in one Common Data Model, and propose a system to implement the same. We combine concepts of ETL, Column-oriented databases, and data wrangling in order to design the system and schema to load and store data into the CDM. We thus present a highly flexible CDM that can successfully accommodate datasets of varying structure and sector. Finally, we insert data from different datasets, but all related to pregnancy, into the CDM and perform analysis on the same to prove the validity and usability of our proposed system.

REFERENCES

- [1] Shawver, Matthew A., Geoff J. Hanson, Greg J. Clark, Daniel D. Gilbertson, Aaron A. Kagawa, and Jian Shi Wang. "Design, Implementation, and Utilization of a Common Data Model for Vehicle Health Management." In Aerospace Conference, 2007 IEEE, pp. 1-14. IEEE, 2007.
- [2] Yoon, Dukyong, Eun Kyoung Ahn, Man Young Park, Soo Yeon Cho, Patrick Ryan, Martijn J. Schuemie, Dahye Shin, Hojun Park, and Rae Woong Park. "Conversion and data quality assessment of electronic health record data at a Korean tertiary teaching hospital to a common data model for distributed network research." *Healthcare informatics research* 22, no. 1 (2016): 54-58.
- [3] Stonebraker, Mike, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau et al. "C-store: a column-oriented DBMS." In Proceedings of the 31st international conference on Very large data bases, pp. 553-564. VLDB Endowment, 2005.
- [4] Abadi, Daniel J., Peter A. Boncz, and Stavros Harizopoulos. "Column-oriented database systems." *Proceedings of the VLDB Endowment* 2, no. 2 (2009): 1664-1665.
- [5] Ong, Toan C., Michael G. Kahn, Bethany M. Kwan, Traci Yamashita, Elias Brandt, Patrick Hosokawa, Chris Uhrich, and Lisa M. Schilling. "Dynamic-ETL: a hybrid approach for health data extraction, transformation and loading." *BMC medical informatics and decision making* 17, no. 1 (2017): 134.
- [6] Kandel, Sean, Jeffrey Heer, Catherine Plaisant, Jessie Kennedy, Frank van Ham, Nathalie Henry Riche, Chris Weaver, Bongshin Lee, Dominique Brodbeck, and Paolo Buono. "Research directions in data wrangling: Visualizations and transformations for usable and credible data." *Information Visualization* 10, no. 4 (2011): 271-288.
- [7] Endel, Florian, and Harald Piringer. "Data Wrangling: Making data useful again." *IFAC-PapersOnLine* 48, no. 1 (2015): 111-112.
- [8] Furche, Tim, Georg Gottlob, Leonid Libkin, Giorgio Orsi, and Norman W. Paton. "Data Wrangling for Big Data: Challenges and Opportunities." In *EDBT*, pp. 473-478. 2016.
- [9] Hanlin, Qin, Jin Xianzhen, and Zhang Xianrong. "Research on extract, transform and load (ETL) in land and resources star schema data warehouse." In *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, vol. 1, pp. 120-123. IEEE, 2012.
- [10] Bansal, Srividya K. "Towards a semantic extract-transform-load (ETL) framework for big data integration." In *Big Data (BigData Congress), 2014 IEEE International Congress on*, pp. 522-529. IEEE, 2014.
- [11] Hripcsak, George, Jon D. Duke, Nigam H. Shah, Christian G. Reich, Vojtech Huser, Martijn J. Schuemie, Marc A. Suchard et al. "Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers." *Studies in health technology and informatics* 216 (2015): 574.