Title: The Generalized Data Model for Clinical Research

Authors: Mark D. Danese, MHS, PhD, Marc Halperin, Jennifer Duryea, MPH, Ryan Duryea

Affiliations:

Outcomes Insights, Inc., Westlake Village, CA

Key words:

Data model, claims data, electronic health records

Disclosures:

Outcomes Insights provides consulting services and software for building studies using observational data. MD, JD, and RD are collaborators in the Observational Data Health Sciences and Informatics organization.

Contact information:

Mark Danese, MHS, PhD

2801 Townsgate Road, Suite 330

Westlake Village, CA 91361

805-498-0034 (office)

805-715-8106 (fax)

Word Count:

4,836 (excluding abstract, tables, and references)

1. Abstract (319 words)

1.1 Background

Healthcare data contains a variety of relationships among data elements that are useful to researchers. However, most data sources store information within their own unique schemas, making it difficult to develop software tools that ensure reliable and reproducible research. One solution to this problem is to create a data model that standardizes the storage of both the data and the relationships among data elements. Developing such a data model was the goal of this research.

1.2 Methods

We defined the Generalized Data Model (GDM) as a set of standard tables in which specific information should be stored. There were two primary aims in constructing the tables and table relationships. The first was to use a single table for storing clinical codes in their original vocabularies to retain the original semantic representation of the data. The second was to retain hierarchical information present in the original data while retaining the provenance of the data. In addition, we borrowed or adapted constructs from other data models where useful. We specifically avoided explicitly requiring the construction of “visits” to store relationships among data elements.

1.3 Results

The resulting data model contains 19 tables, with the Clinical Codes, Contexts, and Collections tables serving as the core of the model, and containing most of the clinical, provenance, and hierarchical information. We loaded synthetic Medicare data as well as Surveillance, Epidemiology, and End Results data linked to Medicare claims into the GDM and tested a subsequent ETL of the synthetic Medicare data from GDM into the Sentinel data model.

1.4 Conclusions

The GDM is designed to retain the relationships among data elements to the extent possible, facilitating ETL and study building as part of a complete data pipeline for researchers. Furthermore, it offers researchers a simpler process of standardizing the location of data in a defined structure and it is intended to make it easier for users to transform their data into other data models.

2. Introduction

Healthcare data contains useful information for researchers across the spectrum of clinical research, including pharmacovigilance, epidemiology, and health services research. However, most data sources store information within their own unique schemas, making it difficult to develop software tools that ensure reliable and reproducible research. One solution to this problem is to create data models that standardize the storage of both the data and the relationships among data elements.

A data model is a set of standard tables, with specific relationships among them, designed to enable standardized retrieval of information from a data source. In healthcare, there are several commonly used data models including those supported by the following organizations: Informatics for Integrating Biology and the Bedside (i2b2)[1–3], Observational Health Data Sciences and Informatics (OHDSI, managing the OMOP data model)[4–6], Sentinel[7–9], and PCORnet[10,11], among others. The first, and biggest, challenge with any data model is the process of migrating the raw (source) data into the data model, referred to as the "extract, transform, and load" (ETL) process. The ETL process is particularly burdensome when one has to support multiple, large data sources, and to update them regularly.

The biggest difficulty with ETL is not the process of standardizing variable names and reorganizing the data; rather, the difficulty is standardizing the relationships among data elements. Since different healthcare data sources encode relationships in different ways, the ETL process can lose information, or create inaccurate information. The best example is creating a visit, a construct which, in most data models, is used to link information (e.g., diagnoses and procedures). Visits are challenging because administrative claims (i.e., billing) data is designed to allow facilities and practitioners to invoice separately for their portions of the same medical encounter, or to allow practitioners to bill for multiple interactions on a single invoice.[12] Even constructing visits from electronic medical records can be challenging when each interaction with a different provider (e.g., nurse, physician, pharmacist, etc.) is recorded separately. And while it may seem like it should be easy to create an unambiguous definition, no data model succeeds in a way that covers all scenarios (see Discussion).

To minimize the need to encode specific relationships that may not exist in the source data, we created a data model with a hierarchical structure that simplifies the ETL process without changing the meaning of the original data. This data model can serve both as a stand-alone data model, as well as a common storage model for later conversion into other data models.

3. Methods

3.1 Design Goals

We defined the Generalized Data Model (GDM) as a set of standard tables in which specific information should be stored. There were two primary aims in constructing the tables and table relationships. Each is described below in a separate section.

3.1.1 Use a single table for clinical codes in their original vocabularies

The first goal was make all clinical codes (e.g., International Classification of Diseases [ICD], Current Procedural Terminology [CPT], National Drug Codes [NDC], etc.) easy to load into the data model and to query because they represent the majority of electronic clinical information. This goal defined the key organizing structure of the GDM: the placement of all clinical codes in a central "fact" table. This is not unlike the i2b2 data model that uses a fact table to store all "observations" from a source data set; however, the GDM was not designed as a star schema despite the similar idea of locating the most important data at the center of the data model.

Furthermore, retaining the original (source) vocabulary prevents the loss of the original semantic expression of the underlying clinical information. Because of this decision for our logical data model, we created separate "details" tables to store information not shared among all records for drugs, inpatient hospitalizations, and measured variables. However, there is no requirement that the physical model conform to the logical model (i.e., the details tables could be combined with the central fact table [Clinical Codes] without changing the data model; see Discussion).

3.1.2 Retain hierarchical information with provenance

The second goal was to capture important hierarchical relationships among data elements within a relational data structure. Based on the review of numerous data sources, we decided on a two-level hierarchy for grouping clinical codes, with the lower level table called Contexts and the higher-level table called Collections. This was based on common data structures where many related codes are recorded on a single record in the source data (Contexts), and where these records are often grouped together (Collections) based on clinical reporting or billing considerations. See Figure 1 for a visual depiction of the hierarchical structure.

Our review of data sources suggested that there were only a handful of relationships that were required to support this structure. The primary relationship represents data that is reported together or collected at the same time. One example of this includes a "line", which occurs in claims data when one or more diagnosis codes, a procedure code, and a cost are all reported

together. Another example includes laboratory values assessed at the same time (e.g., systolic and diastolic blood pressure) which could be considered to be co-reported. Also, a set of prescription refills could represent a linked set of records. Even records that contain pre-coordinated expressions (i.e., a linked set of codes used to provide clinical information akin to an English sentence) could also be stored in order as a single Context record.

We also included the provenance for each clinical code as part of Contexts, recording not only the type of relationship among elements within a Context as discussed above, but also the source file from which the data was abstracted. To minimize the loss of information when converting data to another data model, we did not require visits to be created if they did not exist in explicit form. This is important because visits are not consistently defined among other data models, particularly for administrative claims data. However, in EHR data, visits can be explicitly defined and would be recorded in the Collections table. This can become confusing when considering costs, because costs may be present at a Context (line-item) or Collection (invoice) level. Therefore, this led us to align costs with the Contexts table. By evaluating the type of the context record (i.e., “claim” versus “line”) users can determine whether a cost is an aggregated construct or not. We adapted the Payer Reimbursements and the Costs tables from the OMOP version 5.20 cost table because it was the only data model with a cost table, and because we contributed substantially to its design. However, unlike the single OMOP cost table, we created two tables to accommodate both reimbursement-specific information, which has a well-defined structure, and all other kinds of economic information, which requires a very flexible structure.

3.1.3 Other Considerations

There are several other considerations made in building this data model, some of which were borrowed or adapted from other data models. For example, in addition to the cost table, we borrowed the idea to store all codes as “concept ids” (unique numeric identifiers for each code in each vocabulary to avoid conflicts between different vocabularies that use the same code). We also adapted the idea of OMOP “type_concept_ids” to track provenance within our data model. Finally, we wanted to retain enrollment information so that the data could be used for different purposes (e.g., if a study does not require drug data, then enrollment in a drug plan should not be required). Although it was not an explicit goal in building the GDM, we hoped to facilitate a straightforward, subsequent ETL process to other data models, including OMOP, Sentinel, and PCORnet.

While this data model does not require any relationships (mappings) to standard vocabularies, it does include a Mappings table to make it easier to express relationships among data sources to the extent it is needed or desired. In a few very simple cases like sex and race/ethnicity, the model encourages concept mappings to a core set of values to make it easier for users of a study-builder software applications to filter patients by age, gender, and race/ethnicity using a simpler and/or clearer representation of the underlying information. The model also permits an arbitrarily complex set of relationships to be contained in the Mappings table, along the lines of the approach taken with the OMOP model. However, by storing this information in the Mappings table, we reduce the need to re-load the entire dataset when new mappings become available. However, in all cases, the data model still retains the original codes from the raw dataset.

3.2 Test Data

After designing the data model, we constructed extract, transform, and load (ETL) operations on two data sources. The first was the Medicare Synthetic Public Use Files (SynPUF), which is a 2.1-million-person sample of Medicare beneficiaries from 2008 who were followed for three years created to facilitate software development using Medicare data.[13,14] The second was Surveillance, Epidemiology, and End Results (SEER) data linked to Medicare claims data[15] for small cell lung cancer, as part of an ongoing research project to describe patterns of care in that population.

As part of this process, we also built a test conversion to Sentinel. We tested the feasibility of an ETL process to move SynPUF data from the GDM to the Sentinel data model (version 6.0) to ensure that the model did not contain any structural irregularities that would make it difficult to move data into other data model structures.

3.3 Software

The ETL processes were built using R (version 3.2.5) and the data.table package (version 1.10.4). R was selected because it is an open-source, cross-platform software package; because of its flexibility for composing ETL functions; and because of the availability of the data.table package as an in-memory database written in C for speed. While the use of R may be atypical for ETL tasks, it can be readily scaled to larger cloud-based instances, and, with appropriate sharding of the data, to multiple R instances. Furthermore, its data frame orientation lends itself to other distributed approaches than operate on data frames (e.g., a Spark-based adaptation for use with Hadoop). All of the code for the ETL of the data is

available in the following Github repository for the SETLR (Simple ETL in R) package: https://github.com/outcomesinsights/setlr.

4. Results

The resulting data model contains 19 tables and is depicted in an overall hierarchical view in Figure 2. Details of the tables are also provided in Supplementary Materials, and the most up-to-date version is available on a GitHub repository.[16] The critical part of the ETL process – moving data into the Clinical Codes, Contexts, and Collections tables – is described in Figure 2 for the SynPUF data. Below we discuss each of the resulting tables in more detail.

4.1 Tables

The Clinical Codes, Contexts, and Collections tables make up the core of the GDM (as shown in Figure 1). All clinical codes are stored in the Clinical Codes table. Each row of the Clinical Codes table contains a single code from the source data. In addition, each row also contains a person id, the associated start and end dates for the record, a provenance concept id, and a sequence number. The sequence number allows codes to retain their order from the source data, as necessary. The most obvious example from billing data is diagnosis codes that are stored in numbered fields (e.g., diagnosis 1, diagnosis 2, etc.). But any set of ordered records could be stored this way. Grouping together ordered records is accomplished by associating them with the same id from the Contexts table. The provenance id allows for the specification of the type of record (e.g., admitting diagnosis, problem list diagnosis, etc.).

The Contexts table allows for grouping clinical codes and storing information about their origin. The record type concept id identifies the type of group that is stored. Examples might include lines from claims data where diagnoses, procedures, and other information are grouped, prescription and refill records that might be in electronic medical record or pharmacy data, or measurements of some kind from electronic health record or laboratory data (e.g., systolic and diastolic blood pressure, or a laboratory panel). In addition, the table stores the file name from the source data, the Centers for Medicare and Medicaid (CMS) place of service values (used for physician records since facility records to not have a place of service in claims data), and foreign keys to the care site and facility tables. The Contexts table also contains a person id and both start and stop dates which could be different from the start and stop dates of the individual records (e.g., a hospitalization may have different dates from the individual records within the hospitalization).

The Collections table represents a higher level of hierarchy for records in the Contexts table. That is, records in the Collections table represent groups of records from the Contexts table. This kind of grouping occurs when multiple billable units ("lines" or "details") are combined into

invoices ("claims"). It also occurs when prescriptions, laboratory measures, diagnoses and/or procedures are all recorded at a single office visit. In short, a Collection is typically a "claim" or a "visit" depending on whether the source data is administrative billing or electronic health record data. In using a hierarchical structure, the model avoids the need to construct "visits" from claims data which often leads to inaccuracy, loss of information, and complicated ETL processing. In the simplest possible case, it is possible to have a single record in the Clinical Codes table which is associated with a single Context record, which is associated with a single Collection record, as shown in Figure 1 for a drug record.

The Details tables capture domain-specific information related to hospitalizations, drugs, and measurements. The Admissions Details table stores admissions and emergency department information that doesn't fit in the Clinical Codes, Contexts, or Collections tables. It is designed to hold one admission per row. Each record in the Collections table for an inpatient admission will link to this table.

There are two tables that store cost, charge, or payment data of some kind. Commonly used information in administrative claims data is stored in the Payer Reimbursements table, with a separate column for each piece of information that might be reported. All other financial information is stored in the Costs table, which is designed to support costs of arbitrary types, and uses a "value_type_concept_id" to indicate the specific type. For both tables, each record is linked to a record in the Contexts table. In administrative claims data, this means that each "line" can have a cost record. For records that have costs only at the claim/header level (e.g., inpatient hospitalizations), only Contexts that refer to "claims" (i.e., a record_type_concept_id for "claim") will have costs. For data with costs at both the line and claim/header level, costs can be distinguished by the Context type. In our experience, the sum of the line costs does not always equal the total cost, so depending on the research question, the researcher will need to determine whether claim, line, or both should be used. It is possible that each item sharing a single Contexts table record could have a different cost; therefore, the two cost-related tables include a column to indicate the specific Clinical Code record to which the cost belongs. This might occur, for example, if multiple laboratory tests have different costs, but are share a common provenance (i.e., Contexts record).

The Facilities table contains unique records for each facility where a patient is seen. The facility_type_concept_id should be used to describe the whole facility (e.g., Academic Medical

Center or Community Medical Center). Specific departments in the facility should be entered in the Contexts table using the care_site_type_concept_id field.

The Addresses table captures address information for patients, practitioners, and facilities.

The Contexts Practitioners table links one or more practitioners with a record in the Contexts table. Each record represents an encounter between a patient and a practitioner in a specific context. This role_type_concept_id in the table captures the role, if any, the practitioner played on the context (e.g., attending physician).

The Deaths table captures mortality information at the patient level, including date and cause(s) of death. This is commonly populated from beneficiary or similar administrative data associated with the medical record. However, it is useful to check discharge status in the Admissions Details table as part of ETL process to ensure completeness. There are also diagnosis codes that indicate death. Deaths that are indicated by diagnosis codes will be in the Clinical Codes table and should not be moved to the Deaths table. If needed, these codes can be identified using an appropriate algorithm (e.g., a set of ICD-9 codes) to identify death as part of the identification of outcomes in an analysis.

The Information Periods table captures periods of time during which the information in each table is relevant. This can include multiple records for each person, including records for different enrollment types (e.g., Medicare Part A, Medicare Part B, or Medicare Managed Care) or this can be something as simple as a single date range of “up-to-standard” data as provided by the Clinical Practice Research Datalink. This table includes one row per patient per enrollment type.

The Concepts table provides a unique numeric identifier (“concept_id”) for each source code in each vocabulary used in the data. The Concepts table does not have to be consistent across databases, although there may be efficiencies in using a consistent set of identifiers for all entries from commonly used vocabularies. The specific vocabularies used in the data are provided in the Vocabularies table. The idea of having both Concepts and Vocabularies tables was adapted from the OMOP data models. As mentioned in Methods, the Mappings table allows for the expression of consistent concepts across databases (consistency across databases is not explicitly required).

4.2 ETL

We loaded SynPUF data and SEER Medicare data into the GDM. After downloading the data to a local server, the process of migrating the SynPUF data with 2.1 million patients of data to the GDM took approximately 8 hours on a Windows server with 4 cores and 128 Gb of RAM and conventional hard drives (running two files at a time in parallel). Most of the time was spent loading files into RAM and writing files to disk since the process of ETL with the GDM is primarily about relocating data.

SEER Medicare data for SCLC included approximately 25,000 patients and took less than 1 hour. Selected SEER data was included in the ETL process ignoring recoded versions of existing variables or variables used for consistency of interpretation over time. The ETL process focused on 31 key variables including histology, location, behavior, grade, stage, surgery, radiation, urban/rural status, and poverty indicators. Each SEER variable was included as a new vocabulary in the Concepts table (see Table 1).

The process of moving data from the GDM into the Sentinel data model was straight-forward. Most of the process involved mapping Collections and Context records to visit types from Sentinel, and partitioning the information from the Clinical Codes table into the different domains specified in the data model (e.g., Procedure, Diagnosis, Dispensing, etc.).

5. Discussion

We developed the GDM to support a wide range of clinical research activities. It stores both administrative claims and electronic medical record data while retaining the relationships among data elements as they were represented in the original data. Its main goal is to simplify the location of the most important information for creating analysis data sets and studies, which has the benefit of making ETL easier. It does this by using a hierarchical structure instead of the "visit-centric" approach taken by other data models. It tracks the provenance of the original data elements to enhance the reproducibility of studies. And it allows for a straightforward, subsequent ETL process to other common data models to make it easier for researchers to utilize the analytical tools and frameworks associated with those models.

Part of what makes ETL labor-intensive is that many data models store records in tables intended to hold specific types of information (i.e., domains) to ease analyses. Examples vary by the data model, but typical domains include conditions, procedures, drugs, laboratory values, and observations. Moving raw healthcare data into a data model requires either forcing each vocabulary (e.g., Read) into a single domain, or mapping each code within each vocabulary to its specific domain. Placing codes in domain-specific tables can be particularly challenging when vocabularies cross domains (e.g., Read and International Classification of Diseases [ICD]). The GDM does not use domains; instead, all clinical codes are placed in a single table.

Domains do have advantages. Their main purpose is to facilitate the implementation of automated analyses, which is important for signal detection and other data exploration tasks. However, these benefits may not be relevant to all researchers, particularly those using existing code sets. Importantly, if needed, GDM could support domain information in the Mappings table through the use of a "has_domain" relationship. It could be argued that it is easier to change a domain mapping by updating the Mappings table, rather than re-building the entire data set to move records among the various tables.

Because other data models use visits to connect information within the data model, our emphasis on avoiding visits deserves special comment. Visits are seldom required for clinical research, unless the enumeration of explicit visits is the research topic itself, as might be the case in some health services research. However, for a large portion of research projects, the focus is on identifying the dates of clinically relevant interactions with the healthcare system, which does not require knowledge of a visit, per se. It is a research project in and of itself to define visits, and such definitions are specific to the research question being investigated.[12] A

data model should allow researchers to define visits as appropriate to answer their research questions.

The challenges with visits can best be seen by inspecting the guidelines for creating visits from each data model. In the Sentinel version 6 data model[9], a visit is defined as a unique combination of patient, start date, provider and visit type. Visit types are defined as Ambulatory, Emergency Department, Inpatient Hospital, Non-acute Institutional, and Other. Furthermore, "Multiple visits to the same provider on the same day should be considered one visit and should include all diagnoses and procedures that were recorded during those visits. Visits to different providers on the same day, such as a physician appointment that leads to a hospitalization, should be considered multiple encounters."

PCORnet version 3.0 is very similar to Sentinel.[11] However, PCORnet provides additional guidance on emergency department records, and provides a visit type for emergency department visit that becomes an inpatient stay. It also allows for visit types of Other Ambulatory, No Information, and Unknown.

In the OMOP version 5.20 data model, a visit is defined for each "visit to a healthcare facility." According to the specifications[5], in any single day, there can be more than one visit. One visit may involve multiple providers, in which case the ETL must either specify how a single provider is selected or leave it null. One visit may involve multiple care sites, in which case the ETL must either specify how a single site is selected or leave it null. Visits must be given a type, which is limited to Inpatient Visit, Outpatient Visit, Emergency Room Visit and Long Term Care Visit.

For i2b2, the specifications state a visit ". . . can involve a patient directly, such as a visit to a doctor's office, or it can involve the patient indirectly, as in when several tests are run on a tube of the patient's blood. More than one observation can be made during a visit. All visits must have a start date / time associated with them, but they may or may not have an end date. The visit record also contains specifics about the location of the session, such as the hospital or clinic the session occurred and whether the patient was an inpatient or an outpatient at the time of the visit." There are no specified visit types, and the data model allows for an "unlimited number of optional columns but their data types and coding systems are specific to the local implementation."[3]

Clearly, each data model has different perspectives on the definition of a visit. Such ambiguity can lead to differences in how data are loaded in the ETL process. As a result, inconsistencies

within or across data models can lead to differences in how data are analyzed and in the results, an issue that has already been demonstrated.[17] Examples of challenges when creating visits and using them as the primary way to connect information include enabling users to connect records within visits, like procedures with their associated conditions; separating emergency department records from their associated inpatient admissions; and classifying physician records from inpatient hospitalizations to either inpatient or outpatient/ambulatory visits (most data models consider physician records to reflect outpatient or ambulatory care, regardless of the place of service). Laboratory records could be visits as with i2b2, or could be associated with visits as with other data models. Similarly, prescriptions and refills could be considered visits, or associated with visits. And other information, like family history, might not require a visit at all. In short, the most important structural component of other data models cannot be accurately and consistently defined, which creates problems with consistent analyses across the data models.

Every data model has unique characteristics based on its primary design specifications. The GDM is designed to allow researchers to identify the clinical constructs needed for research without side effects from the artificial creation of either visits or domains. This provides flexibility for researchers to study not only clinical encounters like visits, hospitalizations, emergency room visits, or even episodes of care, but also more basic constructs like conditions or medication use. Virtually every study begins by selecting patients with at least one relevant code found in the Clinical Codes table. In a properly designed database, this can be an efficient way to select a superset of all patients that are needed for research. After identifying the superset, the rest of the processing becomes an exercise in filtering, often using information from Contexts or additional information in one of the Details tables. This makes it easy to use vocabulary tools and a broad library of existing processes to operate on the data and create all the variables for study datasets.

We should also note that the GDM describes the logical model, and not necessarily the physical model for the data. Information from other tables (e.g., the “Details” tables, and/or the Contexts and Collections tables, see Figure 2) could easily be combined into a larger Clinical Codes table for a more de-normalized structure without any loss of fidelity. The benefits of de-normalization depend primarily on the database technology being used. In fact, a nested structure may be beneficial on massively parallel systems to prevent scattering of individual patient data across nodes which can slow performance. On a related note, the GDM logical model may be simpler for humans to use as well, because the most important clinical information is in a single table.

The ability for humans to understand and access data using the data model is important for conceptualizing and debugging analyses using the data.

Finally, the GDM can be used as a standardized waypoint in a data pipeline because all necessary information should be available for the subsequent migration of data into other data models. Other data models have substantial user bases and useful analytical tools for conducting research. By starting with the GDM, researchers can do modest transformations to migrate their data to other data models and make use of the analytical tools offered by those models. While this requires another ETL step, or a view to be created on the GDM, it substantially reduces the burden for accessing useful analytical tools available. And it allows data providers to share a single representation of their data to support a variety of end users. It may also allow for a simple update process instead of a complete new ETL process for databases that are regularly updated.

5.1 Conclusion

The GDM is designed to retain the relationships among data elements to the extent possible, facilitating ETL and study building as part of a complete data pipeline for researchers. Furthermore, it offers researchers a simpler process of standardizing the location of data in a defined structure and it is intended to make it easier for users to transform their data into other data models.

6. Acknowledgements

We gratefully acknowledge the influence of the open-source OMOP common data model specifications on our thinking in creating our data model. In addition, we acknowledge the influence of Sentinel, PCORnet, and i2b2 on our approach, although most of our data model was designed prior to reviewing these models in detail. We also thank Chris Adamson for helpful discussions about organizing the data model in different ways. At the time of writing, all references to the concepts table refer to the OMOP version 5.20 vocabulary table maintained by OHDSI. However, there is no reason that a user could not create their own system of codes with unique identifiers across vocabularies, or use the codes from the National Library of Medicine Metathesaurus.[18]

7. Figures

Figure 1: Graphical Depiction of the Relationships Among the Collections, Contexts, and Clinical Codes Tables

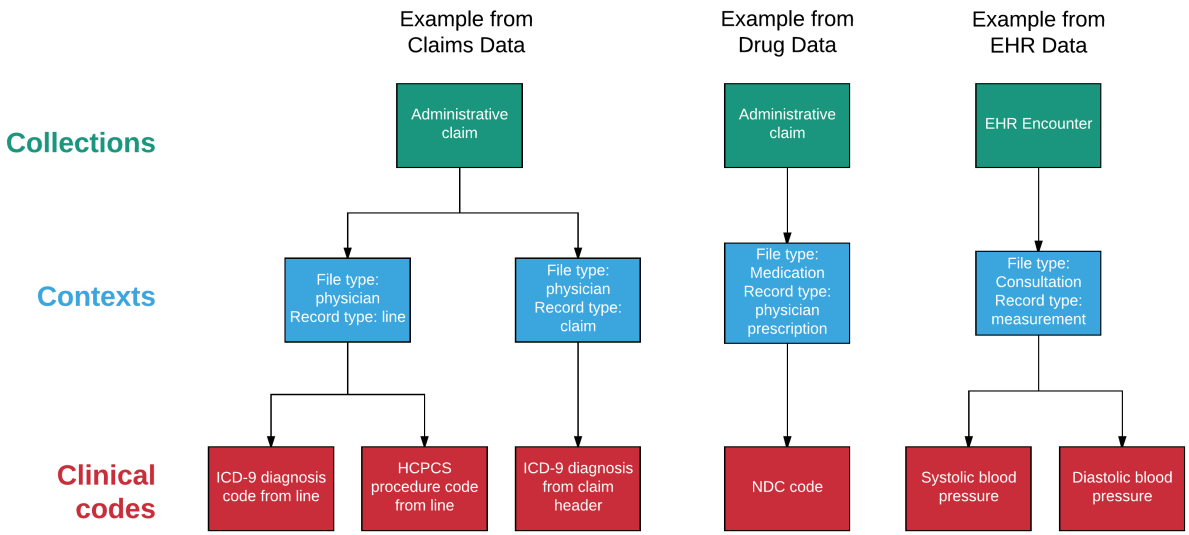
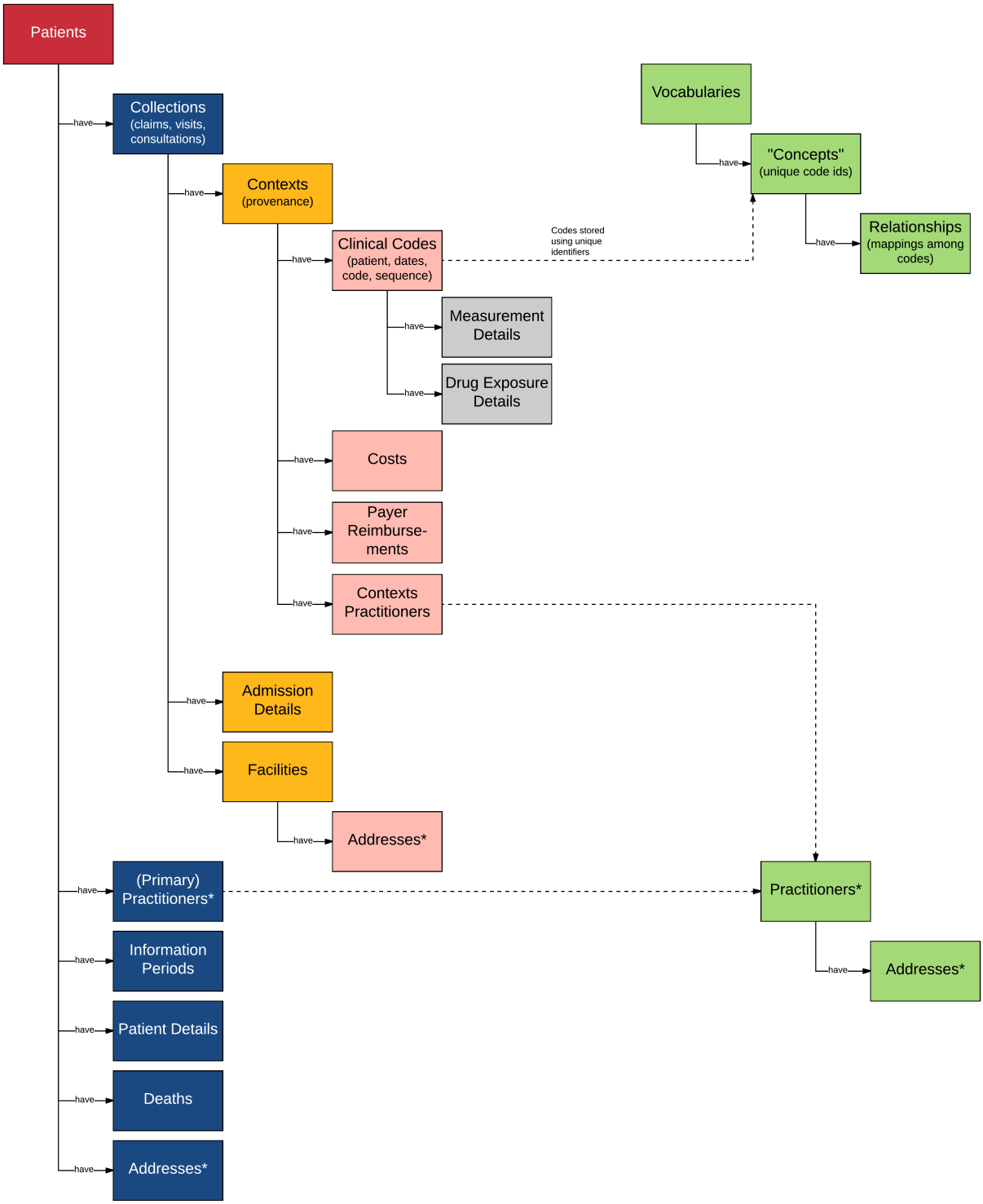


Figure 2: Graphical Depiction of the Generalized Data Model



Note: Table names and key contents are listed above showing the key relationships among tables. See Supplementary Materials for more detail on tables. Tables in green serve as lookup

tables across the database, including the vocabularies used. There is a single addresses table for unique addresses, and a single Practitioners table for unique practitioners. The Contexts Practitioners table allows multiple practitioners to be associated with a Context record.

Figure 3: Graphical Depiction of the ETL Process for SynPUF data

SynPUF Record

patient_id	clm_from_dt	clm_thru_dt	icd9_dgns_cd_1	icd9_dgns_cd_2	hcpcs_cd_1	hcpcs_cd_2	line_ icd9_dgns_cd_1	line_ icd9_dgns_cd_2
54916	2009-10-14	2009-10-14	716.84	719.44	73218	73564	719.44	719.44

Clinical Codes

id	collection_id	context_id	patient_id	start_date	end_date	clinical_ concept_id	quantity	seq_num	type_concept_id	clinical_code_ source_value	clinical_code_ vocabulary_id
8632447	10	3132545	54916	2009-10-14	2009-10-14	44831496		1	100000075	716.84	ICD9CM
8987856	10	3132545	54916	2009-10-14	2009-10-14	44829216		2	100000075	719.44	ICD9CM
18754452	10	4741345	54916	2009-10-14	2009-10-14	44829216		1	100000075	719.44	ICD9CM
6642545	10	4741345	54916	2009-10-14	2009-10-14	2211465		1	100000075	73218	CPT4
23495395	10	9482288	54916	2009-10-14	2009-10-14	44829216		2	100000075	719.44	ICD9CM
4740953	10	9482288	54916	2009-10-14	2009-10-14	2211482		2	100000075	73564	CPT4

Contexts

id	collection_id	patient_id	start_date	end_date	facility_id	facility_ type_concept_id	pos_concept_id	file_ type_concept_id	address_id	service_specialty_ type_concept_id	type_concept_id
3132545	10	54916	2009-10-14	2009-10-14				100000062			100000064
4741345	10	54916	2009-10-14	2009-10-14				100000062			100000065
9482288	10	54916	2009-10-14	2009-10-14				100000062			100000065

Collections

id	patient_id	start_date	end_date	facility_id	admission_detail_id
10	54916	2009-10-14	2009-10-14		

Note: Clinical codes are derived from a single row in the source data set (SynPUF record). Colored arrows indicate how each group of codes is used to create records. Each code from the original record gets its own row in the Clinical Codes table. Codes that are grouped together (e.g., line diagnosis 1 and procedure 1 in yellow) share the same context. In the Contexts table, type concept id

ending in “64” indicates a claim level context, and the id ending in “65” indicates a line level context. The three contexts (groups of codes) share the same collection id.

8. Tables

Table 1: Example Concept Table for Variables in the SEER Data

id	vocabulary_id	concept_code	concept_text
100010943	naaccr_grade	1	Grade I; grade i; grade 1; well differentiated; differentiated, NOS
100010944	naaccr_grade	2	Grade II; grade ii; grade 2; moderately differentiated; moderately well differentiated; intermediate differentiation
100010945	naaccr_grade	3	Grade III; grade iii; grade 3; poorly differentiated; differentiated
100010946	naaccr_grade	4	Grade IV; grade iv; grade 4; undifferentiated; anaplastic
100010947	naaccr_grade	5	T-cell; T-precursor
100010948	naaccr_grade	6	B-cell; Pre-B; B-Precursor
100010949	naaccr_grade	7	Null cell; Non T-non B;
100010950	naaccr_grade	8	N K cell (natural killer cell)
100010951	naaccr_grade	9	cell type not determined, not stated or not applicable

Note: SEER = Surveillance, Epidemiology and End Results, a collection of data from cancer registries across the United States. NAACCR = North American Association of Central Cancer Registries.

Table 2: Example Mapping Table for Conversion to Sentinel Common Data Model

concept_ id _1	relationship _id	concept _id _2	vocabulary _id_1	concept _code_1	concept _text_1	vocabulary _id_2	concept _code_2	concept _text_2
100000002	is_a	100000092	synpuf _sex	2	Female	sentinel _sex	F	Female
100000001	is_a	100000093	synpuf _sex	1	Male	sentinel _sex	M	Male
100000003	is_a	100000095	synpuf _race	1	White	sentinel _hispanic	N	Not Applicable
100000004	is_a	100000095	synpuf _race	2	Black	sentinel _hispanic	N	Not Applicable
100000005	is_a	100000095	synpuf _race	3	Others	sentinel _hispanic	N	Not Applicable
100000006	is_a	100000097	synpuf _race	5	Hispanic	sentinel _hispanic	Y	Yes
100000005	is_a	100000098	synpuf _race	3	Others	sentinel _race	0	Unknown
100000006	is_a	100000098	synpuf _race	5	Hispanic	sentinel _race	0	Unknown
100000004	is_a	100000101	synpuf _race	2	Black	sentinel _race	3	Black or African American
100000003	is_a	100000103	synpuf _race	1	White	sentinel _race	5	White

Note: This table shows how sex and race/ethnicity would be associated if the generalized data model were to use the Sentinel definitions as its defaults. Mappings allow users to query the data based on standardized default values for these variables while retaining the values from the original data. This is most useful for incorporating sex and race/ethnicity into a study builder and

allowing users to filter patients meeting specific inclusion criteria. The generalized data model can support more extensive mapping relationships, but these are not required.

9. References

1. Klann JG, Abend A, Raghavan VA, Mandl KD, Murphy SN. Data interchange using i2b2. *J Am Med Informatics Assoc*. 2016;23(5):909-915. doi:10.1093/jamia/ocv188.

2. Murphy SN, Weber G, Mendis M, et al. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *J Am Med Informatics Assoc*. 2010;17(2):124-130. doi:10.1136/jamia.2009.000893.

3. i2b2 Common Data Model. https://i2b2.org/software/files/PDF/current/CRC_Design.pdf. Accessed April 20, 2017.

4. Overhage JM, Ryan PB, Reich CG, Hartzema AG, Stang PE. Validation of a common data model for active safety surveillance research. *J Am Med Inform Assoc*. 2012;19(1):54-60. doi:10.1136/amiajnl-2011-000376.

5. OHDSI. OMOP Common Data Model. http://www.ohdsi.org/web/wiki/doku.php?id=documentation:overview. Accessed April 20, 2017.

6. Voss EA, Makadia R, Matcho A, et al. Feasibility and utility of applications of the common data model to multiple, disparate observational health databases. *J Am Med Informatics Assoc*. 2015;22(3):553-564. doi:10.1093/jamia/ocu023.

7. Psaty BM, Breckenridge AM. Mini-Sentinel and regulatory science--big data rendered fit and functional. *N Engl J Med*. 2014;370(23):2165. doi:10.1056/NEJMp1401444.

8. Curtis LH, Weiner MG, Boudreau DM, et al. Design considerations, architecture, and use of the Mini-Sentinel distributed data system. *Pharmacoepidemiol Drug Saf*. 2012;21(SUPPL. 1):23-31. doi:10.1002/pds.2336.

9. Sentinel Common Data Model. https://www.sentinelinitiative.org/sentinel/data/distributed-database-common-data-model. Accessed April 20, 2017.

10. Fleurence RL, Curtis LH, Califf RM, Platt R, Selby J V, Brown JS. Launching PCORnet, a national patient-centered clinical research network. *J Am Med Inform Assoc*. 2014;21(4):578-582. doi:10.1136/amiajnl-2014-002747.

11. PCORnet Common Data Model. http://www.pcornet.org/pcornet-common-data-model/. Accessed April 20, 2017.

12. Tyree PT, Lind BK, Lafferty WE. Challenges of using medical insurance claims data for utilization analysis. *Am J Med Qual*. 21(4):269-275. doi:10.1177/1062860606288774.

13. Danese MD, Voss EA, Duryea J, et al. Feasibility of Converting the Medicare Synthetic Public Use Data Into a Standardized Data Model for Clinical Research Informatics. In: *AMIA 2015 Annual Symposium*. San Francisco; 2015. https://knowledge.amia.org/59310-amia-1.2741865/t005-1.2744350?qr=1da.

14. Centers for Medicare and Medicaid Services. Synthetic Public Use File. https://www.cms.gov/research-statistics-data-and-systems/downloadable-public-use-files/synpufs/. Accessed April 20, 2017.

15. Warren JL, Klabunde CN, Schrag D, Bach PB, Riley GF. Overview of the SEER-Medicare data: content, research applications, and generalizability to the United States elderly population. *Med Care*. 2002;40(8 Suppl):IV-3-18. doi:10.1097/01.MLR.0000020942.47004.03.

16. Outcomes Insights Inc. Generalized Data Model. https://github.com/outcomesinsights/generalized_data_model. Accessed April 20, 2017.

17. Xu Y, Zhou X, Suehs BT, et al. A Comparative Assessment of Observational Medical Outcomes Partnership and Mini-Sentinel Common Data Models and Analytics: Implications for Active Drug Safety Surveillance. *Drug Saf*. 2015;38(8):749-765. doi:10.1007/s40264-015-0297-5.

18. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res*. 2004;32(suppl 1):D267-D270. doi:10.1093/nar/gkh061.

10. Supplementary Tables Defining the Generalized Data Model

[see separate document]